Алексей Стахнов

Linux

4-е издание

Санкт-Петербург
«БХВ-Петербург»
2011

УДК 681.3.06
ББК 32.973.26-018.2
С78

Стахнов А. А.

С78 Linux: 4-е изд., перераб. и доп. — СПб.: БХВ-Петербург, 2011. — 752 с.: ил. — (В подлиннике)

ISBN 978-5-9775-0712-7

Приведены подробные сведения об особенностях и возможностях операционной системы Linux, идеологии файловой системы, инсталляции и основных командах, компиляции ядра, настройках и сервисах. Рассмотрены вопросы организации на базе Linux различных серверов и служб: электронной почты, WWW, FTP, INN, Proxy, NTP, а также проблемы администрирования сети, обеспечения безопасной работы и др. Описаны способы настройки под Linux рабочих станций, в том числе и бездисковых, установки и эксплуатации на них графических сред типа X Window, а также конфигурирование модемных соединений, принтеров и сканеров. Уделено внимание отладке взаимодействия с Linux-машинами современной периферии, такой как карманные компьютеры, мобильные телефоны, TV-тюнеры и т. п. Рассматриваемые в книге конфигурационные файлы и структура каталогов соответствуют дистрибутиву Fedora Core 14, однако при минимальной адаптации все упоминаемые в книге пакеты устанавливаются в любом дистрибутиве Linux. В четвертом издании добавлена информация о 3G-модемах, остальной текст обновлен и доработан.

Для пользователей и начинающих администраторов Linux

УДК 681.3.06
ББК 32.973.26-018.2

Группа подготовки издания:

Главный редактор	*Екатерина Кондукова*
Зам. главного редактора	*Евгений Рыбаков*
Зав. редакцией	*Григорий Добин*
Редактор	*Леонид Кочин*
Компьютерная верстка	*Натальи Караваевой*
Корректор	*Наталия Першакова*
Дизайн серии	*Инны Тачиной*
Оформление обложки	*Елены Беляевой*
Зав. производством	*Николай Тверских*

Лицензия ИД № 02429 от 24.07.00. Подписано в печать 10.06.11.
Формат 70×100 1/16. Печать офсетная. Усл. печ. л. 95,46.
Тираж 1500 экз. Заказ №
"БХВ-Петербург", 190005, Санкт-Петербург, Измайловский пр., 29.

Санитарно-эпидемиологическое заключение на продукцию
№ 77.99.60.953.Д.005770.05.09 от 26.05.2009 г. выдано Федеральной службой
по надзору в сфере защиты прав потребителей и благополучия человека.

Отпечатано с готовых диапозитивов
в ГУП "Типография "Наука"
199034, Санкт-Петербург, 9 линия, 12.

ISBN 978-5-9775-0712-7

© Стахнов А. А., 2011
© Оформление, издательство "БХВ-Петербург", 2011

Оглавление

Введение **1**

Почему написана эта книга 1
Для кого написана эта книга 1
Структура книги 2
Как со мной связаться 3

ЧАСТЬ I. ВВЕДЕНИЕ В LINUX **5**

Глава 1. Особенности ОС Linux **7**

FreeDOS 8
Windows NT (Windows 2000, Windows XP, Windows 2003 Server, Windows 2008 Server, Windows Vista, Windows 7) 9
Embedded Windows 9
Windows CE (Windows mobile) 10
Mac OS X 10
iOS 10
Семейство UNIX 10
FreeBSD, OpenBSD, NetBSD 11
Linux 11
Embedded Linux 12
Android 12
MeeGo 12
QNX 12
Symbian 13
Почему выбирают Linux 19
Ссылки 20

Глава 2. Возможности Linux **21**

Сеть 21
  Сетевые протоколы и аппаратура 21
  Сетевые сервисы 22
Файловые менеджеры 23

Текстовые редакторы ... 24
Графические оболочки ... 24
Графические редакторы ... 25
Web-инструментарий ... 25
Офисные пакеты ... 25
Oracle OpenOffice ... 26
OpenOffice.org ... 27
Koffice ... 27
GNOME Workshop ... 27
Базы данных ... 28
Эмуляторы Windows ... 28
Средства разработки программ ... 28
Kylix ... 28
KDevelop ... 29
Glade ... 30
Rhide ... 30
Eclipse ... 30
Lazarus ... 32
Мультимедиа-приложения ... 32
Аудио ... 32
Видео ... 33
Игры ... 33
Итоги ... 34
Ссылки ... 34

Глава 3. Дистрибутивы ... 35

Группа Debian ... 36
Группа Red Hat ... 37
Группа Slackware ... 38
Группа Gentoo ... 38
Дистрибутивы LiveCD ... 39
Дистрибутивы USB Flash ... 40
Дискетные дистрибутивы ... 40
Ссылки ... 41

ЧАСТЬ II. БАЗОВАЯ ИНФОРМАЦИЯ О LINUX ... 43

Глава 4. Работа в сети. Основные понятия ... 45

Модели сетевых взаимодействий ... 45
Терминология ... 45
Модель взаимодействия открытых систем (OSI) ... 46
Модель сетевого взаимодействия TCP/IP ... 48
Сопоставление сетевых моделей OSI и TCP/IP ... 48

Сетевые протоколы ... 49
Семейство протоколов TCP/IP ... 49
Протоколы межсетевого уровня (интернет) ... 49
Протокол IP ... 49
Формат пакета IPv4 ... 50
Протокол IPv6 ... 51
Адресация в IPv6 ... 52
Сетевые пакеты ... 53
Маршрутизация пакетов ... 53
Протоколы маршрутизации ... 53
Адресация в TCP/IP ... 54
Протокол адресации ARP/RARP ... 56
Протокол ICMP ... 56
Протоколы транспортного уровня ... 59
Протокол TCP ... 59
Протокол UDP ... 60
Протоколы уровня приложений ... 60
Протокол FTP ... 61
Протокол SMTP ... 61
Протокол Telnet ... 61
Сетевая файловая система NFS ... 61
Протокол IPX ... 61
Протокол NetBIOS ... 62
Стандарты в Интернете ... 62
Ссылки ... 63

Глава 5. Идеология файловой системы ... 64

История развития файловых систем Linux ... 64
Файл ... 64
Типы файлов ... 65
Файл ... 65
Каталог ... 65
Файл устройства ... 65
Канал ... 65
Ссылки ... 65
Сокет ... 66
Владельцы файлов ... 66
Права доступа к файлам ... 66
Модификаторы прав доступа ... 67
Файловые системы ... 68
Типы файловых систем ... 69
Установка файловой системы ... 70
Монтирование и демонтирование файловой системы ... 70
Поддержка работоспособности файловых систем ... 72

Виртуальная файловая система (VFS) ... 73
Принцип функционирования ... 73
Структура VFS ... 73
Файловая система Ext2 ... 74
Стандартные возможности Ext2 ... 74
Дополнительные возможности Ext2 ... 74
Физическая структура Ext2 ... 75
Оптимизация производительности ... 77
Средства управления файловой системы Ext2 ... 77
Журналируемые файловые системы ... 78
Файловая система Ext3 ... 79
Файловая система Ext4 ... 79
Файловая система ReiserFS ... 80
Ссылки ... 81

Глава 6. Дерево каталогов Linux ... 82

Иерархия каталогов Linux ... 83
Корневой каталог (Root) ... 83
Каталог /bin ... 84
Каталог /boot ... 85
Каталог /dev ... 86
Каталог /etc ... 86
etc/rc.d — инициализационные скрипты системы ... 93
S99windows/etc/sysconfig — конфигурационные файлы для процессов ... 94
etc/X11 — конфигурационные файлы для X Window System ... 101
etc/sgml — конфигурационные файлы для SGML и XML ... 101
Каталог /home — пользовательские домашние каталоги ... 102
Каталог /lib — важные разделяемые библиотеки и модули ядра ... 102
Каталог /lib64 — важные разделяемые библиотеки и модули ядра ... 102
Каталог /lost+found ... 102
Каталог /media — точка монтирования автоматически монтируемых устройств ... 103
Каталог /mnt — точка монтирования для временно монтируемой файловой системы ... 103
Каталог /opt — дополнительные программные пакеты ... 103
Каталог /proc — точка монтирования виртуальной файловой системы procfs ... 103
proc/№процесса_PID-процесса ... 105
proc/ide — IDE-устройства, установленные в системе ... 105
proc/net — сетевая информация ... 105
proc/parport — параллельные порты ... 106
proc/scsi — SCSI-устройства, установленные в системе ... 106
proc/sys — системная информация ... 107
proc/tty — терминалы ... 107
Каталог /root — домашний каталог для пользователя root (администратора) ... 108

Каталог /sbin — системные исполняемые файлы ... 108
Каталог /sys — точка монтирования файловой системы sysfs ... 109
Каталог /tmp — временные файлы ... 109
Каталог /usr — иерархия ... 110
usr/bin — пользовательские программы ... 110
usr/include — каталог для стандартных include-файлов ... 110
usr/lib — библиотеки для программирования и пакетов ... 110
usr/local — локальная иерархия ... 110
usr/sbin — не жизненно необходимые стандартные системные программы ... 111
usr/share — архитектурно-независимые данные ... 111
usr/src — исходные тексты программ ... 113
Каталог /var ... 114
var/cache — кэш программ ... 115
var/games — файлы для игровых программ ... 115
var/lib — библиотеки ... 115
var/lock — lock-файлы (файлы-защелки) ... 116
var/log — файлы и каталоги журналов (log-файлов) ... 116
var/mail — пользовательские почтовые ящики ... 116
var/opt — изменяемые данные для каталога /opt ... 117
var/run — переменные файлы времени исполнения ... 117
var/spool — spool-данные приложений ... 117
var/tmp — временные файлы, сохраняемые между перезагрузками ... 117
var/yp — файлы баз данных Network Information Service (NIS) (опционально) ... 118
Ссылки ... 118

Глава 7. Процесс загрузки Linux ... 119

Программы-загрузчики ... 120
LILO — LInux LOader ... 120
GRUB ... 120
LoadLin ... 120
Параметры ядра ... 121
Обзор параметров строки загрузки ... 121
Утилита rdev ... 121
Разбор параметров ядром Linux ... 121
Общие неаппаратные параметры загрузки ... 122
Опции корневой файловой системы ... 122
Параметр *root* ... 122
Параметры *ro* и *rw* ... 122
Опции управления RAM-диском ... 123
Параметр *ramdisk_start* ... 123
Параметр *load_ramdisk* ... 123
Параметр *prompt_ramdisk* ... 123

Параметр *ramdisk_size* ... 123
Параметр *noinitrd* (начальный RAM-диск) ... 123
Параметры загрузки для управления памятью ... 124
Параметр *mem* ... 124
Параметр *swap* ... 124
Параметр *buff* ... 124
Параметры загрузки для файловой системы NFS ... 125
Параметр *nfsroot* ... 125
Параметр *nfsaddrs* ... 125
Дополнительные параметры загрузки ... 126
Параметр *debug* ... 126
Параметр *init* ... 126
Параметр *kbd-reset* ... 127
Параметр *maxcpus* ... 127
Параметр *md* ... 127
Параметр *no-hlt* ... 127
Параметр *no-scroll* ... 127
Параметр *noapic* ... 127
Параметр *nosmp* ... 127
Параметр *panic* ... 128
Параметр *pirq* ... 128
Параметр *profile* ... 128
Параметр *reboot* ... 128
Параметр *reserve* ... 128
Загрузочные параметры, определяющие поведение шины PCI ... 129
Аргументы *pci=bios* и *pci=nobios* ... 129
Аргументы *pci=conf1* и *pci=conf2* ... 129
Аргумент *pci=io=XXX* ... 129
Аргумент *pci=nopeer* ... 129
Аргумент *pci=nosort* ... 129
Аргумент *pci=off* ... 129
Аргумент *pci=reverse* ... 129
Аргументы загрузки для драйверов буфера видеофреймов ... 130
Аргумент *video=map:...* ... 130
Аргумент *video=scrollback:...* ... 130
Аргумент *video=vc:...* ... 130
Аргументы загрузки для SCSI-периферии ... 130
Аргументы для драйверов Mid-level ... 130
Максимальный LUN (*max_scsi_luns=*) ... 130
Регистрация SCSI (*scsi_logging=*) ... 131
Параметры для ленточного накопителя SCSI (*st=*) ... 131
Аргументы для контроллеров SCSI ... 131
Жесткие диски ... 132
Параметры драйвера IDE — винчестера/CD-ROM ... 132

Последовательные и ISDN-драйверы ... 132
Драйвер PCBIT ISDN (*pcbit*) ... 133
Драйвер Teles ISDN (*teles*) ... 133
Драйвер DigiBoard (*digi*) ... 133
Последовательный/параллельный радиомодем Baycom (*baycom*) ... 133
Драйверы других устройств ... 133
Устройства Ethernet (*ether*) ... 134
Драйвер звуковой карты (*sound*) ... 134
Драйвер принтера (*lp*) ... 135
Процесс init ... 135
Конфигурационный файл init — /etc/inittab ... 136
Основные конфигурационные файлы ... 140
rc.sysinit ... 140
Скрипт rc ... 142
rc.local ... 145
Другие файлы, влияющие на процесс загрузки ... 145
События, происходящие при регистрации пользователя ... 146
Основные файлы, участвующие в регистрации пользователя ... 146
Загрузка в однопользовательском режиме ... 147
Утилиты ... 148
Ссылки ... 148

Глава 8. Безопасная работа в Linux ... 149

Основные положения ... 149
Зачем вам безопасность? ... 149
Надежность защиты системы ... 149
Определение приоритетов защиты ... 150
Политика безопасности ... 150
Основные направления защиты ... 150
Физическая безопасность ... 150
Замки ... 151
Охрана жесткого диска ... 151
BIOS ... 151
Загрузочные устройства ... 152
Безопасность загрузчика операционной системы ... 152
Программы xlock и vlock ... 152
Определение нарушений физической безопасности ... 152
Локальная безопасность ... 153
Регистрация новых пользователей ... 153
Безопасность пользователя root ... 153
Безопасность файлов и файловой системы ... 154
Проверка целостности файлов ... 155
Особенности безопасности файловой системы Ext2(3,4) ... 155
Команды для установки и чтения атрибутов в Ext2 ... 156

Пароли и шифрование ... 156
Протоколы шифрования трафика ... 157
SSH ... 157
PAM ... 157
CIPE ... 157
Kerberos ... 158
CFS и TCFS ... 158
Безопасность ядра ... 158
Устройства ядра ... 158
Сетевая безопасность ... 158
Packet Sniffers ... 159
Системные сервисы ... 159
DNS ... 159
identd ... 159
Сетевые сканеры ... 159
Электронная почта ... 160
"Отказ в предоставлении доступа" ... 160
SELinux/AppArmor ... 160
Безопасность NFS ... 161
Firewall ... 162
Антивирусная защита ... 162
Администрирование системы ... 163
Резервная копия системы ... 163
Режим резервирования ... 163
Резервирование RPM-базы ... 164
Файлы регистрации ... 164
Обновляйте операционную систему ... 165
Действия во время и после взлома системы ... 165
Нарушение безопасности ... 165
Взлом системы произошел ... 165
Закрытие бреши ... 165
Оценка повреждений ... 166
Выслеживание взломщика ... 166
Ссылки ... 166

Глава 9. RPM- и DEB-пакеты ... 168

Система поддержки пакетов RPM ... 169
Принципы наименования пакетов ... 170
Достоинства RPM ... 170
Недостатки RPM ... 171
Информация, содержащаяся в пакете ... 171
Категории пакетов ... 172
Команды консольного менеджера RPM ... 174
Общие опции ... 174

Опции установки и обновления ... 175
Опции удаления (деинсталляции) ... 176
Опции запроса ... 177
Опции выбора пакетов ... 177
Опции выбора информации ... 177
Опции проверки ... 178
Проверка подписи ... 179
Опции сборки пакетов ... 179
Опции пересборки и перекомпиляции ... 180
Подпись существующего RPM ... 180
Подписи PGP ... 181
Опции пересборки базы данных ... 181
Опции FTP/HTTP ... 181
Используемые файлы ... 182
Примеры использования консольного менеджера пакетов RPM ... 182
Система обновлений пакетов Yum ... 186
Поиск в репозиториях ... 187
Установка пакетов с помощью Yum ... 187
Обновление системы ... 187
Удаление пакетов ... 187
Информация о пакетах ... 187
Очистка кэша Yum ... 187
Midnight Commander ... 188
purp ... 189
Kpackage ... 190
GnoRPM ... 190
Yumex ... 192
DEB-пакеты ... 192
Достоинства DEB ... 193
Недостатки DEB ... 193
Информация, содержащаяся в пакете ... 193
Менеджеры DEB-пакетов ... 193
Работа с APT ... 194
Обновление локального кэша пакетов ... 194
Просмотр существующих пакетов ... 194
Просмотр информации об отдельном пакете ... 195
Установка пакета ... 196
Удаление пакета ... 196
Обновление системы ... 196
Aptitude ... 196
Поиск пакетов ... 196
Информация о пакете ... 197
Установка пакетов ... 197
Обновление пакетов ... 197

Переустановка пакета ... 197
Удаление пакета ... 197
Очистка кэша ... 198
Ссылки ... 198

ЧАСТЬ III. ИНСТАЛЛЯЦИЯ LINUX ... 199

Глава 10. Подготовка к инсталляции ... 201

Перед инсталляцией ... 202
До начала работы ... 202
Список оборудования ... 202
Дополнительная информация ... 203
Предполагаемый объем инсталляции ... 203
Разбиение жесткого диска ... 204
Каталог / ... 204
Каталог /bin ... 204
Каталог /boot ... 204
Каталог /dev ... 205
Каталог /etc ... 205
Каталог /home ... 205
Каталог /lib ... 205
Каталог /lost+found ... 206
Каталог /mnt ... 206
Каталог /opt ... 206
Каталог /proc ... 206
Каталог /root ... 206
Каталог /sbin ... 207
Каталог /tmp ... 207
Каталог /usr ... 207
Каталог /var ... 207
Создание разделов на клиентских машинах ... 207
Создание разделов на сервере ... 207
Применение рекомендаций ... 210
Проблемы с оборудованием ... 211
Ссылки ... 212

Глава 11. Инсталляция ... 213

Графическая инсталляция ... 213
Начало инсталляции ... 214
Первые этапы ... 216
Разбиение жесткого диска ... 219
Выбор устанавливаемых пакетов ... 219

Процесс инсталляции ... 221
Конфигурирование системы ... 222
Текстовая инсталляция ... 222
Инсталляция с жесткого диска ... 226
Сетевая инсталляция ... 226
Ссылки ... 226

Глава 12. После инсталляции ... 227

Домашний компьютер ... 227
Офисный компьютер ... 228
Компьютер программиста, администратора ... 229
Сервер ... 230
Ссылки ... 232

ЧАСТЬ IV. ОСНОВНЫЕ КОМАНДЫ LINUX ... 233

Глава 13. Помощь ... 235

apropos ... 235
Man-справка ... 235
whatis ... 235
HOWTO — как сделать ... 236
Мини-HOWTO ... 236
Руководства пользователя Fedora ... 236
Документация Slackware ... 236
Руководство пользователя Alt Linux ... 236
Документация Debian ... 236
Ссылки ... 237

Глава 14. Справочник наиболее часто употребляемых команд ... 238

Стандартный ввод/вывод, перенаправление ... 239
Конвейер (поток) ... 239
Команды ... 240
Дата, время ... 240
cal ... 240
date ... 240
Файлы и каталоги ... 241
Административные команды ... 241
Общие команды ... 243
Сеть ... 249
dig ... 249
elm ... 249
finger ... 249

ftp ... 249
getty (*mgetty*) ... 249
host ... 249
hostname ... 250
ipchains ... 250
iptables ... 250
kppp ... 250
lynx ... 250
mail ... 250
mimencode ... 250
minicom ... 250
netcfg ... 251
netstat ... 251
nslookup ... 251
pine ... 251
ping ... 251
procmail ... 251
ssh ... 252
telnet ... 252
traceroute ... 252
uudecode ... 252
uuencode ... 252
wget ... 252
Администрирование ... 252
at ... 252
atq ... 253
atrm ... 253
batch ... 253
cksum ... 253
crond ... 253
crontab ... 253
getkeycodes ... 253
ifconfig ... 253
insmod ... 254
Isapnp ... 254
kill ... 254
killall ... 254
lilo ... 254
linuxconf ... 255
md5sum ... 255
modprobe ... 256
mount ... 256
nice ... 256
passwd ... 256
pnpdump ... 256

renice ... 256
rpm ... 256
rmmod ... 257
setserial ... 257
setterm ... 257
skill ... 257
snice ... 257
strace ... 258
stty ... 258
umount ... 258
useradd ... 258
xf86config ... 258
xvidtune ... 258
zic ... 258
Состояние системы ... 259
df ... 259
du ... 259
dumpkey ... 260
free ... 260
ftpcount ... 260
ftpwho ... 260
kdb_mode ... 260
last ... 260
ps ... 261
quota ... 261
tload ... 261
top ... 261
uptime ... 262
users ... 262
who ... 262
w ... 263
Создание файловой системы ... 263
fdisk ... 263
fdformat ... 263
mkfs ... 263
Диагностика файловой системы ... 263
fsck ... 263
Архивация ... 263
gzip ... 263
tar ... 264
Работа с текстовыми файлами ... 264
joe ... 264
sort ... 264
uniq ... 264
vi ... 264
vim ... 264

Помощь ... 264
apropos ... 264
man ... 264
whatis ... 264
Разное ... 265
banner ... 265
bash ... 265
bc ... 265
chvt ... 265
clear ... 265
cpp ... 265
csh ... 265
echo ... 265
env ... 265
g77 ... 266
gawk ... 266
gcc ... 266
id ... 266
login ... 266
logname ... 266
make ... 266
nohup ... 267
openvt ... 267
perl ... 267
printenv ... 267
reset ... 267
resizecons ... 267
startx ... 267
strings ... 267
strip ... 268
subst ... 268
su ... 268
true ... 268
yes ... 268
Ссылки ... 268

ЧАСТЬ V. НАСТРОЙКА И СЕРВИСЫ LINUX ... 269

Глава 15. Локализация ... 271

Теоретическая часть ... 272
Стандарты кодировки ... 272
Стандарт ASCII ... 272

Альтернативная кодировка (CP866) ... 273
Кодировка Microsoft CP1251 ... 273
Стандарт КОИ8 ... 273
Unicode ... 274
Украинский язык ... 274
Кириллизация консоли ... 274
Консольный драйвер ... 274
Схема функционирования консольного драйвера ... 274
console-tools ... 275
Cyrillic console tools ... 275
kbd ... 275
Настройка консольных приложений ... 276
bash ... 276
csh/tcsh ... 276
zsh ... 276
less ... 277
mc (The Midnight Commander) ... 277
nroff ... 277
man ... 277
ls ... 277
Samba ... 277
telnet ... 278
Локализация и интернационализация ... 278
Локаль ... 278
Настройка локали ... 278
Интернационализация ... 279
Кириллизация X Window ... 279
Работа с текстом ... 280
Проверка правописания ... 280
Словарь Александра Лебедева ... 280
Словарь Константина Книжника ... 280
Редактор *vim* ... 280
Редактор *joe* ... 280
Кириллица в программах электронной почты и чтения новостей ... 280
elm ... 281
pine ... 281
mutt ... 281
tin ... 281
Кириллические имена файлов ... 281
Поддержка кириллицы в Perl ... 282
Перекодировщики ... 282
Ссылки ... 282

Глава 16. Обновление и компиляция ядра ... 283

Обновление ядра операционной системы Linux ... 283
Подготовка к обновлению ядра операционной системы ... 283
Обновление ядра операционной системы ... 284
Конфигурирование загрузчика ... 284
GRUB ... 284
Компиляция ядра операционной системы Linux ... 286
"За" компиляцию ядра операционной системы ... 286
"Против" компиляции ядра операционной системы ... 286
Утилиты конфигурирования ядра операционной системы Linux ... 287
Процесс компиляции ядра ... 288
Ядро с поддержкой загружаемых модулей (модульное) ... 288
Монолитное ядро ... 290
Параметры настройки ядра ... 291
Параметры настройки ядра (комментарии) ... 291
Ссылки ... 292

Глава 17. DNS ... 293

Настройка сетевых параметров ... 294
Файл host.conf ... 294
Файл /etc/hosts ... 294
Файл /etc/resolv.conf ... 294
Настройка кэширующего сервера ... 295
Файл /etc/named.conf ... 295
Файл /etc/127.0.0 ... 296
Запуск named ... 297
Настройка полнофункционального DNS-сервера ... 298
Файл /etc/named.conf ... 299
Файл /etc/named/ivan.petrov ... 300
Файл /etc/192.168.0 ... 301
Некоторые тонкости ... 301
Записи ресурсов (RR) службы DNS ... 302
Реверсная зона ... 303
Два сервера DNS ... 303
Иерархические поддомены ... 304
Вторичные DNS-серверы ... 304
Используйте серверы кэширования ... 304
Инструменты ... 304
Ссылки ... 304

Глава 18. DHCP ... 305

DHCP-протокол ... 305
Архитектура и формат сообщений ... 305

Режимы выдачи IP-адресов ... 306
Параметры конфигурации (поле *options*) ... 307
Недостатки DHCP ... 308
DHCP-сервер ... 308
Файл dhcpd.conf ... 308
Файл dhcpd.leases ... 311
Пример файла dhcpd.conf ... 312
DHCP-клиент ... 313
Файл dhclient.conf ... 313
Файл dhclient.leases ... 314
Ссылки ... 315

Глава 19. Почта ... 316

Протокол SMTP ... 317
Протокол POP3 ... 317
Протокол IMAP ... 317
Формат почтового сообщения ... 318
Спецификация MIME ... 319
MIME-Version ... 319
Content-Type ... 319
Content-Transfer-Encoding ... 320
Спецификация S/MIME ... 320
PGP, GPG ... 321
Программное обеспечение ... 321
Программа sendmail ... 321
Принцип работы программы sendmail ... 321
Настройка программы sendmail ... 322
Тестирование отправки почты sendmail ... 323
Тестирование обслуживания по протоколу SMTP ... 323
Тестирование обслуживания по протоколу POP3 ... 326
Программа Postfix ... 329
Конфигурационные файлы ... 329
Почтовые клиенты ... 330
Mail ... 330
Pine ... 331
Thunderbird mozilla ... 331
Sylpheed ... 331
Evolution ... 331
Kmail ... 334
Ссылки ... 334

Глава 20. Web-сервер Apache ... 335

Конфигурация ... 335
Раздел "Глобальные переменные" ... 340

Раздел "Конфигурация «основного» сервера" ... 341
Раздел "Виртуальные серверы" ... 342
Файл access.conf ... 342
Ссылки ... 344

Глава 21. FTP ... 345

Протокол FTP ... 345
Представление данных ... 345
Тип файла ... 345
Управление форматом ... 346
Структура ... 346
Режим передачи ... 346
Управляющие команды FTP ... 346
Ответы на управляющие FTP-команды ... 347
Управление соединением ... 348
Программное обеспечение ... 349
Пакет wu-ftp ... 349
Команды ... 349
Конфигурирование сервера ... 351
Файл ftpaccess ... 351
Файл ftpservers ... 356
Файл ftpconversions ... 356
Файл ftpgroups ... 356
Файл ftphosts ... 357
Файл ftpusers ... 357
Параметры запуска программ, входящих в пакет ... 357
Программа ftpd ... 357
Программа ftpwho ... 358
Программа ftpcount ... 358
Программа ftpshut ... 358
Программа ftprestart ... 358
Программа ckconfig ... 358
Формат файла журнала xferlog ... 358
Безопасность ... 359
Ссылки ... 360

Глава 22. NNTP. Сервер новостей INN ... 361

Протокол NNTP ... 361
Основные команды протокола NNTP ... 363
ARTICLE ... 363
BODY ... 364
HEAD ... 364
STAT ... 364

GROUP ggg ... 364
HELP ... 364
IHAVE <message-id> ... 364
LAST ... 364
LIST ... 365
NEWGROUPS date time [GMT] [<distributions>] ... 365
NEWNEWS newsgroups date time [GMT] [<distribution>] ... 365
NEXT ... 365
POST ... 366
QUIT ... 366
SLAVE ... 366
Сервер новостей INN ... 366
Работа пакета INN ... 366
Управляющие сообщения ... 366
Настройка системы INN ... 367
Файл active ... 375
Файлы базы данных и журналы ... 376
Настройка списка получаемых групп новостей ... 376
Журналирование пакета INN ... 379
Программы пакета INN ... 380
Ссылки ... 381

Глава 23. Proxy-сервер ... 382

Squid ... 383
Протокол ICP ... 383
Cache digest ... 383
Иерархия кэшей ... 383
Алгоритм получения запрошенного объекта пакетом Squid ... 384
Конфигурирование пакета Squid ... 384
Сетевые параметры ... 384
Соседи ... 385
Размер кэша ... 385
Имена и размеры файлов ... 386
Параметры внешних программ ... 386
Тонкая настройка кэша ... 387
Время ожидания ... 388
ACL — Access Control List ... 388
Права доступа ... 389
Параметры администрирования ... 389
Параметры для работы в режиме ускорителя HTTP-сервера ... 389
Разное ... 390
Пример конфигурации Squid ... 391
Создание иерархии proxy-серверов ... 393

Transparent proxy ... 393
Ключи запуска Squid ... 394
Файлы журналов Squid ... 395
Файл access.log ... 395
Файл store.log ... 396
Файл useragent.log ... 396
Нестандартные применения ... 396
Борьба с баннерами ... 396
Разделение внешнего канала ... 397
Обработка статистики ... 399
Программа Squid Cache and Web Utilities (SARG) ... 399
Программа MRTG ... 399
Программа RRDtool ... 399
Ссылки ... 400

Глава 24. Синхронизация времени через сеть, настройка временной зоны ... 401

Сетевой протокол времени ... 401
Классы обслуживания ... 402
Обеспечение достоверности данных ... 402
Формат NTP-пакета ... 402
Рекомендуемая конфигурация ... 403
Стандарты ... 403
Сервер xntpd ... 404
Конфигурация сервера ... 404
Класс *symmetric* ... 404
Класс *procedure-call* ... 404
Класс *multicast* ... 404
Общие параметры ... 405
Обеспечение безопасности сервера ... 407
Программы и утилиты, относящиеся к службе точного времени ... 407
ntpdate ... 407
ntpq ... 407
ntptrace ... 408
xntpd ... 408
xntpdc ... 408
Публичные NTP-серверы ... 408
Клиентские программы для синхронизации времени ... 408
UNIX/Linux ... 409
Apple ... 409
Windows ... 409
Ссылки ... 409

Глава 25. Сетевая информационная система NIS (NIS+) и ее конфигурирование. LDAP 410

NIS 410
Как работает NIS 410
Программа-сервер ypserv 411
NIS+ 411
Как работает NIS+ 411
LDAP 412
Установка LDAP-сервера 412
Настройка LDAP-сервера 413
Формат конфигурационного файла 413
Ключи командной строки 417
База данных LDAP 418
Механизмы баз данных LDAP, объекты и атрибуты 418
Создание и поддержание базы данных 420
Утилиты 422
Slapindex 422
Slapcat 422
Ldapsearch 422
Ldapdelete 422
Ldapmodify 423
Ldapadd 423
Kldap 423
GQ 423
Взаимодействие программ с LDAP 423
Ссылки 424

Глава 26. NFS — сетевая файловая система 425

Установка и настройка NFS-сервера 425
Установка и настройка NFS-клиента 426
Опции монтирования 427
rsize 427
wsize 427
soft 427
hard 427
timeo=n 427
retrans=n 428
Безопасность NFS 428
Безопасность клиента 428
Безопасность сервера 428
Ссылки 428

Глава 27. Сервер Samba для клиентов Windows ... 429

Файл конфигурации smb.conf ... 430
Секция *[global]* ... 436
Секция *[homes]* ... 437
Секция *[comm]* ... 438
Секция *[tmp]* ... 438
Пароли пользователей ... 438
Добавление пользователей Samba ... 439
Принтеры ... 440
Использование ресурсов Samba ... 440
Конфигурирование Samba в качестве первичного контроллера домена ... 442
Утилиты ... 443
SWAT ... 444
Webmin ... 444
Ksamba ... 445
SambaSentinel ... 445
Ссылки ... 445

Глава 28. Виртуальные частные сети ... 446

Протокол IPSec ... 447
VPN-сервер FreeS/WAN ... 447
Ipsec.conf ... 448
Ipsec.secrets ... 450
MS Windows NT VPN (PPTP) ... 450
Linux PPTP-сервер ... 451
Linux PPTP-клиент ... 452
OpenVPN ... 452
Ссылки ... 452

Глава 29. Управление процессами ... 453

Выполнение процесса на переднем плане и в фоновом режиме ... 453
Остановка и возобновление процесса ... 455
Завершение работы процесса ... 455
Программы для управления процессами ... 456
nohup ... 457
ps ... 457
top ... 461
kill ... 462
killall ... 463
Изменение приоритета выполнения процессов ... 463
nice ... 464
renice ... 464

Выполнение процессов в заданное время ... 464
at ... 465
batch ... 465
cron ... 465
Ссылки ... 467

Глава 30. Администрирование сети ... 468

Расширенное управление доступом к файлам ... 468
Установка и изменение прав доступа ... 469
Дополнительные возможности ... 470
Шифрование трафика ... 471
Stunnel ... 471
Установка ... 471
Организация шифрованного туннеля ... 471
Stunnel и приложения, поддерживающие SSL ... 472
Сертификаты ... 473
Утилиты сканирования и защиты сети ... 473
SATAN ... 473
Portsentry ... 473
Установка и настройка ... 474
Запуск ... 475
Сетевая статистика ... 476
NeTraMet ... 476
Ключи запуска NeTraMet ... 476
Ключи запуска NeMaC ... 476
Протоколирование ... 477
Демон syslogd ... 477
Параметры запуска ... 477
Файл конфигурации ... 477
Сетевое протоколирование ... 479
Демон klogd ... 479
Защита системы после взлома ... 480
Rootkit ... 480
Обнаружение rootkit ... 481
Сканирование портов ... 482
Использование RPM ... 482
Сканер для rootkit ... 482
После обнаружения ... 483
LIDS ... 483
Установка ... 483
Конфигурирование LIDS ... 485
Способности ... 485
Правила доступа ... 487

Portsentry ... 489
LogSentry ... 489
Tripwire ... 489
AIDE ... 490
RSBAC ... 490
Security-Enhanced Linux ... 490
Ссылки ... 491

Глава 31. Доступ к удаленным компьютерам ... 492

Telnet ... 492
Протокол Telnet ... 492
Команды Telnet ... 493
Программа-клиент telnet ... 494
Программа-сервер telnetd ... 495
Применение Telnet и безопасность ... 495
Семейство r-команд ... 496
rlogin ... 496
rsh ... 496
rcp ... 496
rsync ... 496
rdist ... 496
Применение r-команд и безопасность ... 496
SSH и OpenSSH ... 497
Принцип работы SSH ... 497
OpenSSH ... 497
Конфигурирование OpenSSH ... 497
Ключи запуска сервера SSH ... 502
Ключи запуска клиента SSH ... 503
Программы, входящие в пакет OpenSSH ... 504
Программа ssh-keygen ... 504
Программа ssh-agent ... 504
Программа ssh-add ... 505
Программа sftp ... 505
Программа scp ... 506
Программа ssh-keyscan ... 506
Ссылки ... 507

Глава 32. Firewall ... 508

Типы брандмауэров ... 508
Брандмауэр с фильтрацией пакетов ... 510
Политика организации брандмауэра ... 511
Фильтрация сетевых пакетов ... 512
Фильтрация входящих пакетов ... 512

Фильтрация исходящих пакетов ... 514
Защита локальных служб ... 515
Программа ipchains ... 515
Опции ipchains ... 516
Символьные константы ... 517
Создание правил фильтрации ... 518
Удаление существующих правил ... 518
Определение политики по умолчанию ... 519
Разрешение прохождения пакетов через интерфейс обратной петли ... 519
Запрет прохождения пакетов с фальсифицированными адресами ... 520
Фильтрация ICMP-сообщений ... 522
Сообщения об ошибках и управляющие сообщения ... 522
Противодействие smurf-атакам ... 525
Разрешение функционирования служб ... 525
Запрет доступа с "неблагонадежных" узлов ... 530
Поддержка обмена в локальной сети ... 530
Разрешение доступа к внутреннему сетевому интерфейсу брандмауэра ... 530
Выбор конфигурации для пользующейся доверием локальной сети ... 531
Организация доступа из локальной сети к брандмауэру бастионного типа ... 531
Перенаправление трафика ... 531
Разрешение доступа в Интернет из локальной сети: IP-перенаправление и маскировка ... 532
Организация демилитаризованной зоны ... 533
Защита подсетей с помощью брандмауэров ... 534
Отладка брандмауэра ... 535
Общие рекомендации по отладке брандмауэра ... 535
Отображение списка правил брандмауэра ... 536
Утилиты ... 536
Iptables ... 536
Порядок движения транзитных пакетов ... 538
Порядок движения пакетов для локальной программы ... 539
Порядок движения пакетов от локальной программы ... 539
Таблица *mangle* ... 540
Таблица *nat* ... 540
Таблица *filter* ... 541
Построение правил для iptables ... 541
Команды ipchains ... 541
Критерии проверки пакетов ... 542
Общие критерии ... 543
TCP-критерии ... 544
UDP-критерии ... 544
ICMP-критерии ... 545
Специальные критерии ... 545

Действия и переходы ... 547
Действие *ACCEPT* ... 547
Действие *DNAT* ... 547
Действие *DROP* ... 547
Действие *LOG* ... 547
Действие *MARK* ... 548
Действие *MASQUERADE* ... 548
Действие *MIRROR* ... 548
Действие *QUEUE* ... 548
Действие *REDIRECT* ... 548
Действие *REJECT* ... 548
Действие *RETURN* ... 548
Действие *SNAT* ... 548
Действие *TOS* ... 549
Действие *TTL* ... 549
Действие *ULOG* ... 549
Утилиты iptables ... 549
Iptables-save ... 549
Iptables-restore ... 549
Ссылки ... 550

Глава 33. Организация шлюза в Интернет для локальной сети ... 551

Начальные установки ... 551
Связь с провайдером ... 552
Схема организации подключения локальной сети ... 552
Организация связи через коммутируемое соединение ... 552
Настройка программ ... 553
Настройка связи с провайдером ... 553
Команды pppd ... 555
Настройка diald ... 557
Создание скрипта соединения: /etc/diald/connect ... 558
Настройка основной конфигурации: /etc/diald.conf ... 560
Настройка правил тайм-аутов: /etc/diald/standard.filter ... 561
Комплексное тестирование ... 561
Организация связи по выделенному каналу ... 562
Настройка связи с провайдером ... 562
Комплексное тестирование ... 563
Защита локальной сети ... 563
Установка proxy-сервера ... 563
Transparent proxy ... 564
Борьба с баннерами ... 564
Разделение внешнего канала (ограничение трафика) ... 564
Мониторинг загрузки каналов ... 565

Программа MRTG ... 565
Конфигурирование MRTG ... 566
Программа RRDtool (Round Robin Database) ... 569
Подсчет трафика ... 569
Ссылки ... 570

Глава 34. Настройка модемного соединения ... 571

Протокол PPP ... 571
Общая информация ... 571
Свойства протокола PPP ... 571
Составляющие PPP ... 572
Функционирование протокола PPP ... 573
Поддерживаемое оборудование ... 573
Структура пакета протокола PPP ... 573
PPP-протокол управления соединением (LCP) ... 574
Сокращения, принятые при описании протокола PPP ... 575
Стандарты, описывающие протокол PPP ... 576
Настройка сервера входящих звонков (dial-in) ... 577
Настройка mgetty ... 577
Настройка pppd ... 578
Настройка callback-сервера ... 579
Конфигурация callback-сервера ... 579
Конфигурация клиентов ... 580
Конфигурирование Linux-клиента ... 580
Конфигурирование клиента MS Windows ... 581
Настройка модемного соединения для пользователя ... 581
Настройка модема в текстовом режиме ... 582
Настройка модема в X Window ... 583
Настройка 3G-модема в X Window ... 583
Ссылки ... 588

Глава 35. Резервное копирование и хранение данных ... 589

Планирование резервного копирования ... 589
Что такое резервное копирование ... 592
Носители данных ... 592
Жесткий диск ... 592
Внешний жесткий диск ... 593
CD-RW ... 593
DVD-RW ... 593
Blue Ray-привод ... 593
USB Flash-накопители ... 593
Магнитооптические диски ... 593
Стримеры ... 594
NAS ... 594

Тестирование архивов ... 594
Риск при тестировании архивов ... 595
Утилиты резервного копирования ... 595
Создание резервной копии утилитой tar ... 595
Использование утилиты cpio ... 596
Восстановление с локального ленточного устройства ... 597
Восстановление с удаленного ленточного устройства ... 597
Программа резервного копирования dump ... 597
Создание резервных копий с помощью программы dump ... 598
Восстановление файлов, созданных dump ... 598
Пакет AMANDA ... 599
Команды *mt* и *mtx* ... 599
Команда *buffer* ... 599
Многотомные резервные копии ... 599
Ссылки ... 600

Глава 36. X Window и другие графические оболочки ... 601

Конфигурирование X Window (X Org) ... 601
Конфигурирование X-сервера ... 601
Секция *Files* ... 603
Секция *Keyboard* ... 603
Секция *Pointer* ... 603
Секция *Device* ... 604
Секция *Screen* ... 604
Настройка параметров монитора ... 604
Последовательность запуска X Window ... 605
Конфигурация Window Manager ... 605
Графическая интегрированная среда ... 606
Графическая среда GNOME ... 606
KDE — K Desktop Environment ... 607
Ссылки ... 608

Глава 37. Печать ... 609

Способы вывода на принтер ... 609
Система печати CUPS ... 610
Программный пакет LPD ... 610
Настройка LPD ... 611
Учет ресурсов ... 613
Программа печати LPRng ... 613
Программный пакет netcat ... 613
Система печати PDQ ... 613
Настройка PDQ ... 614
Система буферизации печати PPR ... 614

Печать на сетевой принтер ... 615
Использование принт-сервера ... 615
Печать на Ethernet-принтер ... 617
Графические утилиты конфигурирования принтера ... 617
Ссылки ... 621

Глава 38. Сканер ... 625

Настройка Linux для подключения сканера ... 630
Программный пакет SANE ... 631
Программное обеспечение (frontend) для пакета SANE ... 632
Xsane ... 632
xscanimage ... 633
QuiteInsane ... 633
FlScan ... 633
scanimage ... 633
TkScan ... 633
saned ... 633
scanadf ... 633
scanlite ... 633
xcam ... 633
Staroffice v7/ OpenOffice 1.1 ... 633
NSane ... 634
Программа VueScan ... 634
Ссылки ... 634

ЧАСТЬ VI. РАЗНОЕ ... 623

Глава 39. Различная "экзотическая" периферия и внешние устройства ... 635

Linux и телефоны ... 635
Linux и КПК ... 636
Linux и Palm ... 636
pilot-xfer ... 637
Программы под X Window ... 637
Linux и PocketPC ... 637
Linux и TV-тюнер ... 639
wmtv ... 642
kWinTV ... 642
LIRC ... 643
Создание Real Video под Linux ... 643
Пакет SANE ... 643
Видеокарта с TV-out ... 643
Цифровые фотокамеры ... 645
USB Flash-накопители, картридеры ... 645

Спутниковый Интернет ... 646
UPS (источники бесперебойного питания) ... 646
Ссылки ... 647

Глава 40. Эмуляторы ... 649

Эмуляторы ... 650
DOSEmu ... 650
Конфигурирование DOSEmu ... 650
Wine ... 655
Cedega ... 655
CrossOwer Office ... 656
WINE@Etersoft ... 656
Виртуальные машины ... 656
VMWare ... 656
Установка ... 656
Win4Lin ... 657
VirtualBox ... 657
XEN ... 658
KVM ... 658
Ссылки ... 658

Глава 41. Мультимедиа ... 659

Настройка звуковой карты ... 659
Консольные утилиты для работы со звуком ... 659
Звук в X Window ... 661
Видео в Linux ... 665
Программа XMPS ... 665
Программа avifile-player ... 666
Программа xmms ... 667
Программа XMMP — LinuX MultiMedia Player ... 667
Программа MPlayer ... 667
Программа XINE ... 668
Запись CD-R/CD-RW-дисков ... 669
Создание образа CD-ROM ... 670
Проверка образа CD ... 670
Запись образа диска на CD ... 670
Запись Audio-CD ... 671
Копирование дисков ... 671
Перезаписываемые диски ... 671
Оболочки для записи дисков ... 671
K3b ... 671
Eroaster ... 673
CD Bake Oven ... 673
Ссылки ... 673

Глава 42. Программы для работы с файлообменными сетями, менеджеры закачки 675

Wget 675
Команды Wget 676
Скачивание одного файла 676
Скачивание нескольких файлов 677
Конфигурационный файл .wgetrc 677
Графические интерфейсы для Wget 679
MLdonkey 681
Графические клиенты для MLdonkey 682
Transmission 682
Vuze 683
Ссылки 683

Глава 43. Действия в нештатных ситуациях 684

Утрата пароля root 684
Восстановление без перезагрузки 684
Перезагрузка в однопользовательском режиме 684
Восстановление пароля root после перезагрузки 686
Устранение последствий атак хакеров 686
Проблемы с загрузкой операционной системы 687
Останов загрузки в процессе выполнения LILO 688
Программа LILO выводит последовательность 01010101010 688
Программа LILO останавливается, выдав L 688
Программа LILO останавливается, выдав LI 688
Программа LILO останавливается, выдав LIL? 688
Программа LILO останавливается, выдав LIL 689
Программа LILO останавливается, выдав LIL- 689
Проблемы с выполнением программы LILO 689
Неверная сигнатура LILO 689
BIOS не имеет доступа к жесткому диску 689
Повреждение главной загрузочной записи (MBR) 690
Новое ядро операционной системы не загружается 691
Новое ядро выдает сообщение о превышении размера ядра 691
Ядро выдает сообщение о невозможности монтирования корневого каталога 691
Экран мерцает, и на нем отсутствует приглашение к регистрации в системе 692
Проблемы с запуском программ 692
Повреждение или удаление разделяемых библиотек 692
Сообщение "getcwd: cannot access parent directories" 693
Программа вызывает SIG11 693
Превышение максимального числа открытых файлов 694

Проблемы с файловыми системами ... 694
Ошибка "unable to find swap-space signature" ... 694
Переполнение файловой системы ... 694
Переполнение числа блоков индекса файловой системы ... 695
Подозрение на наличие сбойного кластера или сектора ... 695
При выполнении команды *mount* доступ к системе блокируется ... 695
Случайное удаление файла ... 696
Разрушение данных ... 696
Проблемы с сетью ... 696
К системе нет доступа из сети ... 696
Проблемы ввода/вывода данных ... 697
Любой текст воспроизводится в виде двоичных символов ... 697
Система не реагирует на команды, вводимые с клавиатуры ... 697
Переназначение клавиш ... 697
Окно сеанса X Window не воспринимает команд с клавиатуры и сигналов мыши ... 697
Прочие аварийные ситуации ... 698
Не работает устройство, подключенное к параллельному порту ... 698
После увеличения объема оперативной памяти система работает нестабильно ... 698
После увеличения объема оперативной памяти система не видит добавленную память ... 698
Ссылки ... 698

ПРИЛОЖЕНИЯ ... 699

Приложение 1. Дополнительная литература ... 701

Приложение 2. Ссылки ... 703

Дистрибутивы ... 703
Документация ... 704
Программное обеспечение ... 705
Безопасность ... 706

Предметный указатель ... 707

Введение

Это уже четвертое издание книги и, я надеюсь, не последнее. Книга несколько "похудела", немного изменилась структура и достаточно сильно поменялось содержание. Кое-что обновилось — все течет, все изменяется, одно добавлено, а от другого пришлось отказаться — слишком устарело. По мере возможностей я старался писать о самом современном программном обеспечении, исправлять неточности и ошибки. Но человек несовершенен, поэтому пожелания и замечания, описания интересных программ и особенностей Linux прошу присылать мне.

Почему написана эта книга

Это достаточно сложный вопрос. Здесь переплелись и меркантильный интерес, и честолюбие, желание попробовать себя в другой области, попытка побороть свою неуверенность и лень, и не в последнюю очередь хотелось сделать книгу для наших реалий и нашей специфики. Не секрет, что большинство переводной литературы неадекватно нашей полунищей действительности. Часто можно встретить несколько "раздражающие" для администратора бюджетной организации советы типа "... в качестве маршрутизатора мы рекомендуем использовать устройство фирмы Cisco со следующими параметрами… ". Конечно, с точки зрения надежности, простоты в обслуживании и тому подобных вещей такой совет верен. А для скромного бюджета какой-нибудь государственной конторы трата 4–5 тысяч американских долларов за "железку" размером с кирпич — полный абсурд. Поэтому для наших реалий нужна книга, описывающая построение сетевой и программной инфраструктуры, позволяющей решать большинство типовых задач. Помимо этого, одной из причин для написания книги явилось желание систематизировать и углубить свои собственные знания об операционной системе Linux и ее приложениях.

Для кого написана эта книга

Прежде чем создавать какое-то произведение, автор всегда должен определить своего потенциального читателя. Каким же я его вижу? Это должен быть человек, увлекающийся информационными технологиями, обладающий достаточно приличным багажом знаний в области программного обеспечения (как правило, хорошо

знающий ОС Windows), интересующийся настройкой программ и собирающийся перейти к использованию операционной среды Linux (или недавно проделавший это). При этом уровень книг серии "для чайников" или "сделай все за 21 день" его заведомо не устраивает, поскольку ему необходимо четко представлять себе возможности операционной системы, ее структуру, решаемые с ее помощью прикладные задачи, наиболее популярное программное обеспечение, его установка, настройка и применение. При этом я подразумеваю, что читатель хочет попробовать "выжать" из Linux максимум, поэтому в книге рассматривается множество вопросов — и настройка сервисов, и мультимедиа, и различная "экзотика".

Вот таким мне представляется читатель этой книги.

Структура книги

Материал книги условно разбит на семь частей. Рассмотрим, что в них описывается и для кого они предназначены.

Часть I представляет интерес для новичков в мире Linux. В ней содержится обзор операционных систем, достаточно подробно освещены их особенности, надеюсь, объективно показаны конкуренты Linux и описаны их достоинства и недостатки. Также дан краткий обзор программного обеспечения по основным направлениям: компиляторы и средства разработки, офисные пакеты и мультимедиа, игры и графические редакторы. И главное, представлена классификация дистрибутивов и их краткое описание. Эти сведения будут интересны в первую очередь начинающим и чуть более опытным пользователям, поскольку администраторы со стажем должны знать данную тему назубок.

Часть II важна как для новичков, так и для "продвинутых" пользователей операционной системы Linux, поскольку именно здесь рассматриваются основополагающие вопросы, базис, на основе которого приходит понимание логики системы. В этой части вы узнаете о сетях, их основных понятиях и протоколах; об идеологии файловой системы, дереве каталогов Linux и загрузчиках ядра операционной системы; о процессе загрузки Linux, что за чем происходит, на каких этапах осуществляется инициализация оборудования, загрузка сервисов и сетевых служб, загрузка текстовой консоли или графической оболочки. Также вы узнаете о принципах распространения программного обеспечения, о том, какими бывают инсталляционные пакеты, и о менеджерах пакетов. Здесь же пойдет речь о безопасности операционной системы.

В *части III* рассматриваются вопросы, связанные с инсталляцией операционной системы — разбиение диска, выбор пакетов, постинсталляционная "уборка территории". Поскольку, как говорил Козьма Прутков, "нельзя объять необъятное", инсталляцию операционной системы я буду рассматривать на примере достаточно простого дистрибутива Fedora Core.

Часть IV посвящена основным командам. Это утилиты справочной системы (помощи) и наиболее часто используемые консольные команды. Конечно, для пользо-

вателя вроде бы необязательно знать консольные команды, если он постоянно "сидит" в графической оболочке, но идеология Linux/UNIX подразумевает наличие небольших утилит, что плохо совмещается с графическими приложениями.

Предыдущие главы книги были подготовительным этапом для освоения *части V*. Она предназначена больше для опытных пользователей, т. к. я хотел, чтобы моя книга служила вам верой и правдой в качестве справочного пособия долгое время, и вы периодически возвращались к ней для решения специфических задач, возникающих в работе. Здесь вы найдете описание основных приложений для организации надежного функционирования сети организации, подключенной к Интернету. В этой части рассмотрена защита сети от нежелательного воздействия, создание виртуальных частных сетей, учет сетевого трафика, настройка сетевых принтеров, применение бездисковых компьютеров и организация шлюза в Интернет. Здесь описывается процесс компиляции ядра Linux, установка и настройка всех основных сетевых сервисов: почты, FTP, HTTP, DNS, NFS, Proxy, удаленного доступа и т. п.

Часть VI посвящена отдельным вопросам, не вошедшим в предыдущие разделы. Это настройка и использование сканеров, устройств бесперебойного питания, TV-тюнеров и видеовыходов видеокарт, фотоаппаратов и КПК. Последняя глава описывает действия в нештатных ситуациях.

Заключительная часть книги содержит приложения, в которых приведен список рекомендуемой литературы, небольшая коллекция ссылок, тем или иным образом касающихся Linux и программ для этой операционной системы.

Как со мной связаться

Читатели, которые захотят внести свои предложения или уточнения по содержанию данной книги, поделиться интересными идеями, темами и тому подобным, могут воспользоваться электронным адресом **alexey_stahnov@ukr.net**. Я постараюсь ответить на все письма.



Часть I

Введение в Linux

Глава 1



Особенности ОС Linux

Операционных систем существует очень много, они созданы для любого мыслимого и немыслимого применения. Компьютеры и стиральные машины, суперсерверы и MP3-плееры, мобильные телефоны и автоматические кофеварки, — везде есть операционные системы. Не буду подробно останавливаться на истории и особенностях каждой из них — для этого понадобится чрезмерно много места и времени. Да и не всем это интересно: раз вы читаете эту книгу, значит, вас заинтересовала Linux. Особо любопытным рекомендуем обратиться к соответствующей литературе и Интернету — там можно найти много интересной информации по данной теме.

Условно операционные системы можно разделить на несколько групп:

- серверные и пользовательские;
- обслуживаемые и необслуживаемые;
- встраиваемые (Embedded) и загружаемые;
- ОС реального времени с гарантированным временем отклика на события и прочие.

Операционные системы можно также классифицировать по платформе, для которой они предназначены, по назначению и по занимаемому объему.

Какие же операционные системы используются в настоящее время на компьютерах общего назначения? Вот некоторые из них:

- Windows NT (Windows 2000, Windows XP, Windows 2003 Server, Windows 2008 Server, Windows Vista, Windows 7);
- Embedded Windows;
- Windows CE;
- Mac OS X;
- семейство UNIX;
- FreeBSD, OpenBSD, NetBSD;
- Linux;
- Embedded Linux;
- QNX;
- Symbian;
- iOS;
- MeeGo;
- Android.

Конечно, приведенный список далеко не полон, но мы сознательно ограничим его наиболее распространенными современными операционными системами.

Некоторые из них сейчас мало используются, тем не менее, для полноты картины они представлены в перечне. Рассмотрим этот список с точки зрения человека, которому необходимо иметь на компьютере операционную систему, удовлетворяющую нескольким, порой противоречивым, требованиям.

Как уже упоминалось, операционные системы можно классифицировать по многим параметрам:

- платные и бесплатные (условно-бесплатные);
- с открытым исходным кодом (с правом вносить изменения или без права внесения изменений) и с закрытым исходным кодом;
- одноплатформенные (способные функционировать только на одной платформе, например PC-совместимой) и многоплатформенные (способные функционировать на нескольких платформах, например PC-совместимой, Macintosh, Sun, PowerPC);
- однозадачные и многозадачные;
- однопользовательские и многопользовательские;
- серверные, клиентские или универсальные;
- с текстовым, графическим или комбинированным интерфейсом;
- ориентированные на работу с сетью и Интернетом;
- различные по потребляемым ресурсам и т. д.

А ведь это только малая часть параметров классификации. Критериев для выбора намного больше. Имеют право на существование и такие критерии, как "Она мне нравится" или "Друг себе поставил, чем я хуже?" А ведь выбор операционной системы определяет, как вы будете жить и работать в ближайшие несколько лет (или десятилетий). Поэтому к выбору ОС следует относиться с большой тщательностью и достаточной долей скептицизма. Сжато охарактеризуем наиболее популярные операционные системы.

> ***Замечание***
>
> В 1980 году была организована инициативная группа под названием /usr/group с целью стандартизации программного интерфейса UNIX. Стандарт был разработан к 1984 году и использовался комитетом ANSI при описании библиотек языка C. В 1985 году был создан Portable Operating System Interface for Computing Environment, сокращенно POSIX (переносимый интерфейс операционной системы для вычислительной среды). На сегодняшний день большинство операционных систем удовлетворяют (полностью или частично) стандарту POSIX.

FreeDOS

Свободная операционная система, совместимая с MS-DOS. Выпускается под лицензией GNU. Чаще всего применяется для загрузки утилиты перепрошивки BIOS или различных утилит для проверки дисков, а также для предустановки на бюджетные компьютеры.

Windows NT (Windows 2000, Windows XP, Windows 2003 Server, Windows 2008 Server, Windows Vista, Windows 7)

Ощущая бесперспективность развития ветки "DOS — Windows 3.1*x*", Microsoft разработала новую операционную систему — Windows NT (New Technology), базирующуюся на стандарте POSIX и новой файловой системе NTFS (New Technology File System). Разработчики Windows NT серьезно взялись за проектирование операционной системы с учетом ее дальнейшего развития (последняя в этой линейке операционная система — Windows 7 для настольных компьютеров, последняя версия серверной ОС — Windows 2008). Актуальны две ее ветки: Windows 7 (Windows XP) и Windows 2008 Server. Первый вариант предназначен для настольных систем, второй представляет собой серверную платформу. В обеих версиях имеется графический интерфейс, что для сервера, в общем-то, излишество и нерациональная трата ресурсов. Это многозадачная, многопользовательская, одноплатформенная (PC) и весьма устойчивая операционная система. Имеет встроенную поддержку многопроцессорных и кластерных систем.

Windows для настольных систем производитель делит на две группы: для дома и для офиса. В домашней версии нет некоторых возможностей работы в домене, администрирования и разграничения ресурсов, которые присутствуют в версии для офиса. Также есть сверхдешевая версия для дома, способная работать на маломощных компьютерах и ограничивающая пользователя некоторым количеством одновременно запущенных программ.

Выпускается как в 32-, так и в 64-разрядном вариантах. Обеспечивает хорошую поддержку мультимедийных устройств и игр. Полного набора серверных приложений в составе инсталляции не содержит (приходится докупать и устанавливать отдельно). Исходный код недоступен, система платная. Помимо NTFS, поддерживает файловые системы FAT. Требования к аппаратному обеспечению Windows XP: процессор не хуже Celeron 800 МГц, не менее 128 Мбайт оперативной памяти, не менее 500 Мбайт места на жестком диске. Для Windows 7 процессор с тактовой частотой не менее 1,6 ГГц, не менее 1 Гбайт оперативной памяти, не менее 3 Гбайт места на жестком диске. Windows 2008 Server: процессор не менее 1,6 ГГц, не менее 1 Гбайт оперативной памяти, не менее 1500 Мбайт места на жестком диске.

В ближайшие несколько лет Windows XP еще останется актуальной и будет использоваться наравне с Windows 7.

Embedded Windows

"Встраиваемая" версия Windows. Разделяется на два семейства: Embedded Windows и Windows CE. Embedded Windows — Windows 2000, Windows XP — специальные облегченные, "урезанные" версии, оптимизированные под определенную аппаратуру, но практически не отличающиеся по функциональным возможностям от своих настольных "собратьев". Встречаются в основном в банкоматах. Система платная, исходный код частично доступен для разработчиков аппаратуры.

Windows CE (Windows mobile)

Семейство ОС, предназначенное для мобильных телефонов, смартфонов и КПК. Microsoft пытается также продвигать эту ОС на рынок бытовой аппаратуры. Windows CE 4.*x* и последующие версии предназначены только для ARM.

Помимо цифрового обозначения версий встречаются и такие названия: PocketPC 2000, PocketPC 2002, Pocket Mobile 2003, Windows CE .NET.

Microsoft активно продвигает это семейство на рынок КПК и смартфонов.

Достоинства этой ОС — неплохая совместимость по API со своими "настольными" версиями, что упрощает перенос программного обеспечения, а также хорошая графика и поддержка различной периферии. Недостатки — не очень продуманный интерфейс, не всегда стабильное функционирование (отмечаются непонятные задержки в работе), "раздутый" размер операционной системы и приложений. Система платная, исходный код закрытый.

Mac OS X

Производитель — Apple. Новая операционная система для компьютеров Macintosh, базирующихся на процессоре X86. Это UNIX-подобная, POSIX-совместимая, многозадачная ОС с графическим интерфейсом. Она способна выступать в качестве как сервера, так и клиентской операционной системы. Поддерживает сетевую файловую систему NFS (Network File System). Система платная, исходный код частично закрыт, предназначена для работы на компьютерах Macintosh, однако энтузиасты с успехом устанавливают ее и на стандартные PC-совместимые компьютеры. Требования: не менее 512 Мбайт оперативной памяти, рекомендуемый объем жесткого диска — 5 Гбайт.

iOS

Apple iOS — операционная система, разработанная специально для телефонов, планшетов и проигрывателей фирмы Apple. Входит в семейство операционных систем OS X. Великолепная эргономика, продуманность системы. Достаточно разумные требования к аппаратуре. Система платная, исходный код закрыт, работает только на аппаратуре фирмы Apple.

Семейство UNIX

Группа операционных систем, имеющих общего предка и традиционно носящих название UNIX. Фирмы-производители: AT&T, DEC, Sun, Hewlett-Packard, IBM, SCO и многие другие. Несмотря на то, что первая версия UNIX была выпущена более 30 лет назад, UNIX до сих пор считается наиболее современной, надежной и динамично развивающейся операционной системой для массового пользователя. Большой вклад в успех UNIX внесли специалисты AT&T, студенты и преподаватели университета Беркли. UNIX той или иной фирмы-производителя установлена

практически на каждом сервере уровня предприятия, больших кластерах и мультипроцессорных системах, а также на многих рабочих и графических станциях. Это многоплатформенная, мультизадачная, многопользовательская ОС. Поддерживает кластеризацию, мультипроцессорные системы, распределенные вычислительные среды, массивы накопителей огромной емкости и многое другое. На сегодняшний день трудно найти компьютер, на котором не смогла бы работать одна из версий UNIX. Как правило, UNIX, выпускаемая фирмами, платная, с закрытым исходным кодом. Тем не менее существует довольно много бесплатных UNIX-совместимых операционных систем с открытым исходным кодом (например, семейство BSD, Linux). Благодаря POSIX и многим другим стандартам и соглашениям практически любое приложение можно перенести из одного представителя семейства UNIX в другой. Поэтому для UNIX имеется огромное количество как бесплатных, так и коммерческих программ. Как правило, для каждой разновидности UNIX разработана своя файловая система, но все они могут работать с распространенными файловыми системами.

Существуют и так называемые *журналируемые* файловые системы, в которых для решения проблемы повреждения структуры файлов или данных применяют транзакции. Транзакция считается незавершенной до тех пор, пока все изменения не сохранены на диске. А для того чтобы сбои, происходящие до завершения всех операций, входящих в транзакцию, не приводили к необратимым последствиям, все действия и все изменяемые данные протоколируются. В том случае, если сбой все-таки произойдет, по протоколу можно вернуть систему в рабочее состояние. Требования к аппаратной платформе самые разнообразные. Как уже упоминалось, трудно найти компьютер, на котором не смогла бы работать одна из версий UNIX.

FreeBSD, OpenBSD, NetBSD

POSIX-совместимые операционные системы семейства UNIX на основе кода университета Беркли. Принципиальные различия:

- FreeBSD — очень надежная, достаточно консервативная система (в хорошем смысле этого слова). Аппаратная платформа — Intel;
- NetBSD — переносимая на большое количество аппаратных платформ;
- OpenBSD — попытка объединить достоинства FreeBSD и NetBSD в одном дистрибутиве с упором на безопасность системы.

Все эти системы бесплатные, с открытым исходным кодом. На сегодняшний день наибольшее распространение из-за своей особой надежности получила FreeBSD. Системы обеспечивают двоичную совместимость со многими программами, построенными под SCO, BSD/OS, Net/Free/OpenBSD, 386BSD и Linux.

Linux

POSIX-совместимая UNIX-подобная операционная система. Самая распространенная бесплатная ОС с открытым исходным кодом. При ее разработке из мира семейства UNIX старались взять все лучшее. Благодаря участию десятков тысяч разра-

ботчиков и координации их действий через Интернет Linux и программное обеспечение для нее развивается очень динамично, ошибки и различные проблемы в программах, как правило, исправляются в считанные часы после их обнаружения. Большую помощь в развитии и распространении Linux и сопутствующего ПО оказали Фонд свободного программного обеспечения (Free Software Foundation, USA) и лицензия GNU для программного обеспечения (The GNU General Public License, Универсальная общественная лицензия GNU). На сегодняшний день существует основное ядро Linux, разработку которого координирует его создатель Линус Торвальдс, и множество дистрибутивов (не менее 5–7 десятков), отличающихся как функциональным назначением, так и составом программ, входящих в дистрибутив. Существуют дистрибутивы, занимающие десяток компакт-дисков или несколько DVD, и дистрибутивы, умещающиеся на одной-двух дискетах. Все, что относится к семейству UNIX, справедливо и для Linux. Широчайшая поддержка аппаратных платформ, мультипроцессорных систем и распределенных вычислений, малая требовательность к аппаратным ресурсам, масштабируемость, кластеризация, множество графических оболочек — и это далеко не все. Поддерживаются десятки файловых систем, "родная" файловая система Ext2 (Ext3, Ext4). И при всей мощи это довольно дружественная операционная система, способная работать как на высокопроизводительном сервере, так и на стареньком "пентиуме" где-нибудь в офисе или на КПК.

Embedded Linux

Этот вариант Linux специально предназначен для различной бытовой аппаратуры, промышленных мини-компьютеров, КПК, смартфонов и телефонов.

Популярна среди азиатских разработчиков, поскольку бесплатна и имеет открытый исходный код.

Android

Данный вариант Linux создан специально для мобильных телефонов, планшетов и смартбуков. Производится фирмой Google.

MeeGo

Еще один вариант Linux, предназначенный для мобильных телефонов и планшетов. Разрабатывается совместно фирмами Intel и Nokia.

QNX

Производитель QNX — QNX Software Systems. Это UNIX-подобная, POSIX-совместимая, многозадачная, многопользовательская и микроядерная ОС реального времени. Первоначальное предназначение — промышленная операционная система для работы в режиме 99,999% надежности ("пять девяток"). Применяется для управления технологическими процессами, начиная от атомных электростанций

и заканчивая производством мороженого. Исходный код до последнего времени был закрыт. Планируется открыть исходный код для некоммерческого использования. Есть проблемы с драйверами (пока драйверов немного). Минимальные требования для промышленного дистрибутива: 386-й процессор, 8 Мбайт ОЗУ. Помимо дорогостоящих промышленных дистрибутивов QNX существует бесплатный вариант "QNX Real Time Platform", который загружается с сайта производителя (**www.qnx.com**). Минимальные требования для бесплатного дистрибутива: процессор Pentium-200, 32 Мбайт ОЗУ, 100 Мбайт на жестком диске.

Symbian

Операционная система для КПК и смартфонов от Symbian Inc. Предыдущие версии ОС назывались EPOC и выпускались английской фирмой Psion, которая производила одноименные КПК. В дальнейшем эта фирма отказалась от производства КПК и создала Symbian Inc, которая занимается разработкой ОС и акциями которой владеют крупнейшие производители мобильных телефонов. Одна из наиболее продуманных и надежных ОС для КПК. Система платная, исходные коды недоступны.

Ознакомившись с приведенным кратким обзором операционных систем, можно представить в общих чертах их области применения, достоинства и недостатки. Поскольку наша книга посвящена Linux, а операционные системы Windows 9*x* или Windows NT/2000/XP установлены приблизительно на 90% PC-совместимых персональных компьютерах, то в дальнейшем мы будем сравнивать эти три операционные системы, не забывая, впрочем, и об остальных. Что же касается КПК — это отдельный класс "компьютеров" и рассматриваться в данной книге он не будет (за исключением главы, посвященной взаимодействию Linux и мобильных телефонов, КПК и фотоаппаратов).

НЕБОЛЬШОЕ ОТСТУПЛЕНИЕ

Что такое пользователь? Никогда не задумывались? "Пользователя" сложно охарактеризовать "среднестатистическим" понятием. Он многолик и разнообразен. Единственное, что есть общего у всех пользователей компьютера — они сидят за компьютером. С точки зрения системного администратора это все, кто входит в систему в качестве пользователя, "юзера". С точки зрения системного программиста — все, кто запускает программы на компьютере. Для разработчика прикладного программного обеспечения — пользователи его программы. Для авторов книг "… для чайников" — это люди, знающие о компьютере только то, что у него есть шнур питания и какая-то доска с кнопками. И так далее. Если попытаться обобщить, основной пользователь — это человек, который не разбирается в устройстве компьютеров, не знает, как настроить модем, не обязан знать тонкости операционной системы и т. п. Пользователь решает с помощью компьютера свои профессиональные задачи, зачастую не имеющие с компьютерами ничего общего. На практике все, конечно, не так мрачно. Для успешной работы пользователь просто обязан знать, что такое файл, как настроить рабочий стол, установить программу, что такое вирусы и как с ними бороться и т. д.

Пользователей условно можно разделить на три группы: не знающий о компьютере ничего, знающий кое-что и знающий многое. В соответствии с уровнем пользователей можно выделить три категории операционных систем.

К первой категории операционных систем можно отнести Mac OS X и Windows 7, ко второй — Windows XP, к третьей — DOS, Windows 2008 Server, UNIX-семейство, BSD-семейство, Linux и QNX. Такое разбиение не всегда соответствует официальному позиционированию фирм-разработчиков (например, Microsoft рекламирует Windows XP как систему для домохозяек — включил и работай). Однако с точки зрения коллективного разума (по крайней мере, так считают авторы новостных конференций, посвященных сравнительному обзору операционных систем) данная классификация достаточно верна. Впрочем, жизнь, как всегда, не стоит на месте. Сейчас уже можно говорить, что Linux с ее графическими менеджерами окон KDE и GNOME и им подобным постепенно переходит, если уже не перешла, во вторую категорию (т. е. для пользователей, знающих об операционной системе кое-что), при этом не теряя ни мощности, ни возможности настройки всего и вся. Семейство Windows постепенно сдвигается к группе пользователей, не знающих об операционной системе ничего, вызывая при этом заметное раздражение "продвинутых" (Advanced Users) своей уверенностью, что пользователь системе приносит только вред, а посему ничего настраивать он не должен, а если очень хочет, то пусть платит за поддержку или специальное ПО.

В идеале операционная система должна удовлетворять, по меньшей мере, семи противоречивым требованиям:

1. Легкость в освоении и дружественность к пользователю (User Friendly).
2. Мощность и универсальность (способность работать на любом оборудовании).
3. Достаточно простая настройка.
4. Надежность (в идеале — сверхнадежность).
5. Малая потребность в памяти и других ресурсах.
6. Быстрая реакция разработчиков на проблемы, обнаруженные в процессе эксплуатации.
7. Широкий выбор совместимого программного обеспечения.

Рассмотрим эти пункты подробнее. Пункт *первый*. Тут, собственно, и так все ясно. От того, как быстро человек освоится с операционной системой и насколько удобно ему в ней работать, напрямую зависит производительность труда, да и просто хорошее настроение. Пункт *второй*. Можно, конечно, возразить, что чем более универсальный инструмент, тем слабее он для какого-нибудь специфического применения, и в теории это действительно так. Но давайте посмотрим на универсальность с другой стороны. Теоретические принципы построения операционной среды, по большому счету, одинаковы, что для старенькой 386-й, что для новейших мультипроцессорных систем. Специфику платформы (тип процессора, мультипроцессорность, кластеризацию и т. п.) всегда можно учесть при разработке специфического ядра ОС или драйверов. Некоторая потеря в производительности с лихвой окупается тем, что пользователю, поработавшему на мощнейшем сервере и перешедшему на офисный компьютер, графическую станцию или домашний ПК, не придется осваивать другую операционную систему, поскольку его ОС может функционировать на любом компьютере. А способность работать на любом компьютере автоматически подразумевает, что операционная система должна занимать как можно меньше места и потреблять мало аппаратных ресурсов. Пункт *третий*.

И здесь все понятно без пространных пояснений. Пользователь должен иметь возможность настроить ОС под свои нужды, не прибегая к стороннему (не входящему в поставку операционной системы) программному обеспечению. Пункт *четвертый*. Правда, больное место? Ведь каждый пользователь хочет, чтобы зависания никогда не происходили на его компьютере. Пункт *пятый*. Здесь тоже все понятно. Пункт *шестой*. И это требование очевидно. Пользователь должен получить исправления к своей операционной системе при обнаружении просчетов ее разработчиков. Причем как можно скорее, если ОС удовлетворяет п. 4, и абсолютно бесплатно, поскольку это просчет разработчика. Пункт *седьмой*. Пусть операционная система будет самой распрекрасной, но если для нее нет программ, она останется невостребованной.

Теперь оценим операционные системы на соответствие перечисленным требованиям:

- DOS — не удовлетворяет ни одному пункту, кроме п. 7;
- Windows 3.1*x* — удовлетворяет п. 1 с оговорками, частично пп. 3 и 5, удовлетворяет п. 7;
- OS/2 — удовлетворяет пп. 1 и 3, п. 2 (с учетом одноплатформенности), частично удовлетворяет пп. 4, 5 и 7;
- Windows 9*x* — удовлетворяет п. 1, частично п. 3, безусловно удовлетворяет п. 7;
- Windows NT (Windows XP, Windows Vista, Windows 2003 server) — удовлетворяет п. 1, п. 2 (с учетом одноплатформенности и непомерных требований к аппаратному обеспечению), пп. 3 и 4 с оговорками, безусловно удовлетворяет п. 7;
- Mac ОС — полностью удовлетворяет п. 1, п. 2 (с учетом одноплатформенности), частично пп. 3–6, удовлетворяет п. 7;
- Mac ОС X — безусловно удовлетворяет п. 1, п. 2 (с учетом частичной многоплатформенности и завышенных требований к аппаратному обеспечению), пп. 3–7;
- UNIX-семейство — безусловно удовлетворяет всем пунктам, кроме п. 1, да и то, в последнее время легкость освоения и дружественность у UNIX-разработчиков стоят на первом месте;
- FreeBSD, OpenBSD, NetBSD — все сказанное о UNIX-семействе справедливо и для этих операционных систем;
- Linux — безусловно удовлетворяет всем пунктам, особенно пп. 2, 3, 6 и 7;
- BeOS — удовлетворяет всем пунктам, кроме п. 7;
- QNX — удовлетворяет всем пунктам.

Попробуем выбрать операционную систему, исходя из перечисленных пунктов. Mac OS X — неплохая операционная среда с точки зрения как пользователя, так и администратора, но она подходит *только* для компьютеров фирмы Apple (или специально собранных компьютеров, более-менее способных соответствовать требованиям операционной системы). А в нашей стране подобных компьютеров не наберется и одного процента от общего количества персональных ЭВМ. QNX — достаточно специфичная система, рассчитанная для применения в сверхнадежных системах реального времени. Что остается: семейство Windows, семейство UNIX, а также представители "свободного мира" UNIX: FreeBSD, OpenBSD, NetBSD и Linux.

Теперь попытаемся максимально корректно сопоставить Windows-семейство и Linux.

Во-первых, что очень выгодно отличает Linux от Windows — ее *бесплатность*. За Windows 7 Home Basic по сегодняшним ценам придется уплатить около 90 долл., а за Windows 2008 Server порядка 740 долл. Кроме того, для офисной работы обычно нужен и Microsoft Office, за стандартный вариант которого придется уплатить около 200 долл. и, если потребуется еще что-то, придется продолжать платить и платить. Сегодня никого не удивляет, когда затраты на ПО превышают стоимость самого компьютера. А если у вас несколько компьютеров, то во столько же раз возрастет требуемая сумма. Вот и получается, что маленькая фирма с пятью компьютерами потратит 5–7 тыс. долл. только на программное обеспечение. Но это только начало. Политика Microsoft очень проста и действенна — раз в полгода-год выходит новая версия программного продукта, который все вольно или невольно вынуждены покупать, потому что партнеры присылают вам файлы в формате Excel 2010, а ваш Excel 2000 отказывается их понимать. В результате за всю жизнь компьютера (3–5 лет) только на ПО придется потратить порядка 1–3 тыс. долл. С другой стороны, Linux обойдется в 5–15 долл., за которые можно купить 2–3 диска, заполненные бесплатным программным обеспечением с открытым исходным кодом. Даже если скачивать дистрибутив Linux через Интернет, все равно не потратить больше 20 долл. (приблизительно столько стоит месяц неограниченного подключения к Интернету). И что характерно, с этого дистрибутива можно сколько угодно раз инсталлировать Linux *на абсолютно законных основаниях*. Можно возразить, что за деньги, потраченные на продукты Microsoft, пользователи получают поддержку сервис-центра Microsoft. У некоторых локальных сборщиков дистрибутивов тоже есть телефонные сервис-центры, но для Linux поддержка в основном осуществляется через Интернет. Поскольку Linux — дитя Интернета, решение проблем следует искать там. Помимо Интернета, где находятся тысячи Web-сайтов, посвященных как Linux в целом, так и конкретному программному продукту для нее, существуют десятки групп новостей, а также в дистрибутив входит более 15 тыс. страниц документации, описывающих всё и вся. Есть, правда, одно неудобство — поскольку Linux разрабатывается и сопровождается людьми всех стран мира, то и документация, в основном, на английском языке. Впрочем, это небольшая плата за обладание практически бесплатным программным обеспечением. Тем не менее существует большой пласт литературы и на русском языке. Если же вы решили купить дистрибутив, обычно в его цену входит печатная документация и консультации по телефону в течение 90 дней после покупки.

Во-вторых, Linux *способна функционировать* на множестве аппаратных платформ и *с минимальными требованиями к аппаратуре*. С Windows сложнее. Она совместима только с процессорами Intel или их клонами, а по требованиям к аппаратуре превосходит Linux. И если Windows 9*x*/ME сносно работает на Pentium-166 с 64 Мбайт оперативной памяти, то для Windows NT/2000/XP требуется хотя бы Pentium II 350 МГц и 128, а лучше 256 Мбайт оперативной памяти и процессор раза в два помощнее.

По поводу дружественности, легкости в освоении и инсталляции. Установить Linux на абсолютно чистый диск сможет любой пользователь, для этого нужно только взять соответствующий дистрибутив. Практически все современные дистрибутивы, ориентированные на конечного пользователя, все сделают сами (если, конечно, это нужно) — самостоятельно разобьют и отформатируют жесткий диск, настроят требуемую раскладку языка и интерфейс (богатейший выбор из более чем ста языков: русский, украинский и белорусский в том числе), определят аппаратное обеспечение компьютера и настроят его на максимальную производительность. Установят необходимое программное обеспечение в зависимости от выбранного профиля компьютера (сервер, рабочая станция, ноутбук или выборочная установка), при этом ни в коей мере не ограничивая владельца в самостоятельной конфигурации. Что примечательно, устанавливать Linux можно в текстовом (обычно так поступают опытные пользователи, или если машина слабая) или в графическом интерфейсе, с CD-ROM, жесткого диска, с Flash-накопителя или даже по сети, загрузив компьютер со специально изготовленного диска. В процессе инсталляции можно указать Linux при старте загружать графическую оболочку или работать в текстовом режиме. Единственная проблема — экзотическое аппаратное обеспечение или так называемые WIN-устройства, с ними возникают трудности, поскольку производители подобных изделий всю функциональность вынесли в драйверы, а исходные коды драйверов не предоставляют. Однако и это решаемо. В большинстве случаев драйверы можно найти в Интернете. Поэтому миф о сложности инсталляции не соответствует действительности. Набирают популярность так называемые LiveCD и USB Flash-накопители с предустановленной ОС, позволяющие работать с Linux на компьютере, просто загрузившись с этого диска.

С легкостью освоения, несомненно, похуже. Для грамотной работы в Linux необходимо иметь представление об операционной системе. К сожалению, Windows приучила пользователя щелкать мышью и не думать. Мешает и наш менталитет — "сами с усами", метод "тыка". В UNIX это не проходит. Там подход другой — прочитай, разберись и можешь быть уверен, что это одинаково функционирует в любой UNIX-подобной системе. Еще нюанс — документация для Linux пишется в расчете на грамотного, способного размышлять человека. Это, разумеется, отпугивает пользователя, привыкшего руководствоваться инструкцией-комиксом, и порождает очередной миф о недружелюбности Linux. Однако приятно, что творцы документации считают тебя умным человеком, а не семилетним ребенком.

Относительно настройки операционной системы. Microsoft внедрила в свою ОС непродуманную идею — системный реестр. В результате получился монстрообразный (зачастую в несколько десятков мегабайт) файл двоичного формата, от целостности которого зависит жизнеспособность операционной системы. Очевидно, разработчики совсем забыли старое изречение "Не клади все яйца в одну корзину". Очень часто (по меньшей мере, в 30–40 % случаев) ошибки функционирования операционной системы связаны с повреждением файла реестра. Еще одна проблема — очень многие настройки Windows не описаны в документации, и необходимо перерыть горы литературы, чтобы по крохам насобирать информацию о тонкой настройке системы. Есть, конечно, программное обеспечение, позволяющее

настроить Windows, но, как правило, оно не бесплатно. В Linux все более надежно и доступно. Практически все о настройке системы или программного обеспечения можно узнать из документации. Конфигурационные файлы обычно для каждой программы отдельные, и практически все имеют понятный текстовый формат с подробными комментариями. А настроить в Linux можно все, причем отдельно для каждого пользователя в системе.

О надежности. Конечно, семейство Windows — это не Windows 3.1*x* и даже не Windows 95, для которых ни дня не проходило без сбоя, но до надежности и живучести Linux (не говоря уже о проверенных десятилетиями UNIX) Windows еще далеко, хотя стабильная работа без перезагрузки в течение одной-двух недель для настольного компьютера под Windows уже давно стала нормальным явлением.
Потребность в ресурсах — в настоящее время, наверное, уже не так актуально, сколько операционная система занимает места на жестком диске: 500 Мбайт или 5 Гбайт, но, все равно, чем меньше система, тем она быстрее и надежнее. Тут опять в лидерах Linux — ее можно установить на одну дискету 1,44 Мбайт. Вполне функциональный интернет-сервер можно уместить в 80–150 Мбайт. С Windows XP, а уж тем более с Windows 2008, такого сделать не удастся.

Реакция разработчиков на ошибки. Скорость исправления ошибок в Windows в большинстве случаев достаточно мала (порядка нескольких недель). Надежность программ пропорциональна количеству человек, которое участвовало в тестировании. У производителей закрытого коммерческого ПО процесс тестирования является, по большей части, внутренним. С открытыми программами, в частности с Linux и программным обеспечением для нее, дело обстоит гораздо проще. Практически у каждого проекта есть две ветки — стабильная и текущая. В стабильную входит код, который был проверен многими пользователями в течение некоторого разумного времени. Текущая ветка содержит рабочую версию, которая может изменяться ежедневно, включает все последние нововведения, но при этом не гарантирована от ошибок. Каждый для себя решает, чем пользоваться — стабильной веткой или нестабильной, но содержащей все нововведения. Поскольку процесс тестирования открытого ПО не имеет ограничений по времени, он продолжается все то время, что существует конкретное программное обеспечение. Более того, программист, имея на руках исходные тексты, может сам исправить ошибку, не дожидаясь, пока это сделают за него. Благодаря интернет-сообществу практически всегда ошибки, обнаруженные в ПО для Linux, исправляются в течение суток и тут же становятся доступными для скачивания из Интернета.

Единственное, в чем Windows превосходит Linux, — это *в количестве и разнообразии прикладного программного обеспечения*. Тем не менее в последние полтора-два года очень бурно пошел процесс переноса под Linux коммерческого программного обеспечения. Пожалуй, сейчас осталось мало направлений, для которых в Linux нет бесплатного или, на худой конец, платного ПО. Появились офисные программные комплексы, совместимые по форматам файлов с Microsoft Office, разнообразные интернет-приложения, базы данных, мультимедиа-приложения и т. п. Конечно, остались и незанятые ниши, например, нет того изобилия программ для многоцветной полиграфии, трехмерного моделирования, видеомонтажа или игр.

Но давайте себя спросим, много ли людей занимаются видеомонтажом или анимацией? Наверное, даже не сотая часть процента компьютерных пользователей. А если на компьютере только играть, зачем вообще ПК? Есть ведь Sony Play Station и Microsoft Xbox.

Отдельного упоминания заслуживает *безопасность*. Нехорошо, когда чуть ли не каждую неделю по всему офису прокатывается эпидемия компьютерного вируса, который, ко всему, портит данные на жестком диске. Или кто-то удалил на вашем компьютере данные случайно. Или произошло еще что-то похуже. На сегодняшний день для Windows существует более 1 800 000 (!!!) вирусов или троянских программ, причем многие несут в себе деструктивные функции. Конечно, с безопасностью в Windows NT намного лучше, но, тем не менее, ее гораздо чаще взламывают через сеть, чем UNIX. Для Linux в настоящее время существует около сотни вирусов или троянских коней, причем реально опасных из них всего десяток. И программы, через которые происходило проникновение троянских коней, давно уже избавлены от этого недостатка.

Подведем итог, почему выбирают Linux.

Почему выбирают Linux

Приведем ряд аргументов в пользу Linux.

- Лучшая операционная система — UNIX. Linux — это современный вариант UNIX, работающий практически на всех платформах.
- В отличие от большинства операционных систем, дистрибутивы Linux бесплатны, их можно скачивать из Интернета.
- В стандартный дистрибутив Linux входят сотни программ, позволяющие выполнить 95% задач, решаемых с помощью компьютера.
- Исходный код всех программ под Linux открыт, при желании его можно модифицировать так, как нужно.
- На базе Linux легко создать очень надежные (99,99%) центры данных с поддержкой кластерных конфигураций и высокой степенью масштабирования.
- Корпоративная intranet-сеть "из коробки", элементарная установка интернет-сервисов и серверов, практически сразу настроенных для стандартного применения.
- Высокая степень безопасности и ограничения доступа к ресурсам и данным системы.
- Большое количество аппаратных платформ, поддерживаемых Linux.
- Графический интерфейс с десятками оконных менеджеров, позволяющих создать эксклюзивную графическую среду, точно настроенную для нужд пользователя под имеющиеся аппаратные ресурсы.
- Относительно малые требования к аппаратным ресурсам, достаточно новый дистрибутив вполне можно установить на старших 486-х компьютерах.
- Обширная библиотека документации, ежедневно улучшающаяся и дополняющаяся.
- Великолепная поддержка программного обеспечения, ответы практически на любой вопрос можно найти в Интернете, а также у самих разработчиков, которые не скрываются за копирайтом большой фирмы.

- В Linux можно настроить всё и вся. Простота конфигурации и подробное описание конфигурационных файлов выгодно отличают Linux от большинства коммерческих ОС.
- Можно инсталлировать Linux на одну дискету, и при этом она способна выполнять функции маршрутизатора или отправлять электронную почту.
- Постоянное обновление и улучшение как ядра Linux, так и большинства программных продуктов для нее.
- Независимость от патентов и лицензий.

Ссылки

Сайты, посвященные QNX:

- **www.qnx.com** — сайт фирмы QNX Software Systems, разработчика QNX.

Сайты, посвященные FreeBSD:

- **www.freebsd.org** — сайт FreeBSD;
- **www.freebsd.ru** — русскоязычный сайт.

Сайты, посвященные Linux:

- **www.linux.org.ru** — отличный сайт о Linux;
- **www.linux.org** — сайт о Linux;
- **www.linuxdocs.org** — много литературы о Linux;
- **www.linuxrsp.ru** — русскоязычный сайт;
- **www.redhat.com** — сайт версии Red Hat;
- **fedoraproject.org** — сайт Fedora;
- **www.debian.org** — сайт Debian;
- **www.slackware.com** — сайт Slackware.

Сайты, посвященные Windows:

- **www.microsoft.com** — официальный сайт фирмы Microsoft;
- **www.winfiles.com** — обширная коллекция программ для Windows.

Сайты, посвященные Apple:

- **www.apple.com** — официальный сайт Apple;
- **www.apple.ru** — русскоязычный сайт Apple.

Глава 2



Возможности Linux

В этой главе пойдет разговор об администраторах, офисном и домашнем применении Linux. Автор достаточно долго занимается сопровождением как программ, так и локальных сетей и компьютеров, поэтому не понаслышке знает проблемы администратора. Администратор — это человек, который во время рабочего дня ничего не делает, пьет кофе и играет в компьютерные игры, шляется по офису и болтает с сотрудниками. В идеале администратору платят зарплату за то, что он бездельничает. В том смысле, что надежное и не слишком требовательное к сопровождению оборудование и программное обеспечение (включая, разумеется, и операционную систему), будучи один раз правильно отлажено, должно потом долго работать, не требуя постоянного вмешательства администратора для дополнительных перенастроек, переналадок и инсталляций. Если в вашей организации это не так, можно сделать вывод, что у вас неудачное программное обеспечение либо плохой администратор. Поэтому всех, кто отвечает за бесперебойную работу компьютеров и периферии, всегда интересует, как реализовано администрирование в той или иной операционной системе. И если эта задача решена недостаточно хорошо, заставляя делать изо дня в день одно и то же, такая операционная система вызывает раздражение и желание сменить ее на более "дружелюбную". Офисное использование Linux интересует нас с точки зрения применимости ее на рабочем месте, в фирме, на предприятии. Домашнее применение тоже, разумеется, будет рассмотрено. О серверном функционировании Linux в этой главе мы подробно говорить не будем, потому что этому посвящена добрая половина книги. Кроме того, практически все знают, что UNIX (Linux) и сервер — "близнецы-братья", а об установке ее в офисе или дома еще мало кто задумывается. Но начнем с азов. Как выразился один из грандов компьютерного бизнеса, "Компьютер — это сеть".

Сеть

Сетевые протоколы и аппаратура

Linux по умолчанию работает со своим "родным" протоколом TCP/IP, на котором функционирует и Интернет. Но это вовсе не означает, что она, кроме этого протокола, ничего не понимает. При установке соответствующего программного обеспечения, входящего в дистрибутив, Linux способна также взаимодействовать

с протоколами IPX/SPX фирмы Novell Netware, NetBIOS (только в варианте "NetBIOS поверх TCP/IP"; Microsoft Windows 3.1*x*, Windows 9*x*/ME, Windows NT/2000/XP) и AppleTalk (Apple Mac OS). И это еще не все, что она понимает и поддерживает, хотя перечисленные четыре сетевых протокола сегодня используются, наверное, более чем в 95% случаев.

Из аппаратных сетевых средств Linux способна работать практически с любым оборудованием, предназначенным в том или ином виде для сетевых соединений: сетевыми картами Ethernet, Radio Ethernet, Wi-Fi, ArcNet, аппаратурой для спутникового Интернета, ISDN, ATM, обычными модемами и др. Правда, с аппаратным обеспечением не все так гладко, как хотелось бы.

Как правило, для всех распространенных устройств существуют драйверы для Linux. Можно сказать, что при наличии соответствующего программного обеспечения и драйверов сетевые протоколы и аппаратура под Linux очень хорошо настраиваются с помощью текстовых конфигурационных файлов или специальными программами, например netconf.

Замечание

С написанием названий программ ситуация двойственная — в UNIX и, соответственно, в Linux регистр символов имеет значение, и поэтому названия программ в командной строке необходимо набирать правильно. Традиционно системные утилиты пишут исключительно строчными, "маленькими" буквами, хотя в документации к ним же можно увидеть, что некоторые имена содержат и прописные, "большие" буквы. Такая двойственность в ряде случаев имеет место и в этой книге.

Сетевые сервисы

О сетевых сервисах более подробно будет рассказано в последующих главах книги, а сейчас приведем краткий обзор возможностей. Начнем с Интернета, где зарождалась и развивалась Linux. Было бы удивительно, если бы дитя Интернета и представитель семейства UNIX не предоставлял всей полноты интернет-сервисов. Что интересует пользователя в Интернете? На первый, поверхностный, взгляд Web-сайты, FTP, электронная почта и новости. Но для нормального (и комфортного) функционирования Интернета необходимо множество других сервисов — это и DNS, и proxy-серверы, и серверы точного времени, и многое другое. Все это для Linux есть, и не в единственном экземпляре, нужно только решить, какой "тяжести" инструмент необходим. Сказанное касается и серверного, и клиентского ПО. Так, например, Web-браузеров существует более десяти: Links, Lynx, w3m — текстовые браузеры, Mozilla, Opera, Konqueror, Galeon, Firefox, Nautilus и др.

Почтовых клиентов также известно несколько десятков, как текстовых, так и графических: Pine, Mutt, Elm, Thunderbird, Kmail, Evolution, Sylpheed, Balsa, Gnus, Aethera и т. д.

Можно рассматривать любой интернет-сервис, и всегда в списке клиентских приложений для этого сервиса будет не менее десятка программ. Если необходим файл-сервер, тоже есть большой выбор. Можно пользоваться "родным" NFS, можно Mars — файл-сервером для сетей Netware, можно Samba — файл-сервером для сетей Microsoft. Для всех упомянутых файловых серверов, конечно же, есть

и клиентское программное обеспечение. При желании можно создать тонкий клиент — компьютер без жесткого диска и каких-либо накопителей, загружающийся через сеть и нормально функционирующий (причем с графической оболочкой). Решены для Linux и вопросы статистики. Множество пакетов могут собрать, обработать, представить в текстовом и графическом виде информацию о любой стороне функционирования Linux, в частности о загрузке сети, входящем и исходящем трафике, построить диаграммы, отобразить их на Web-странице и, если необходимо, адекватно отреагировать на какое-то отклонение в функционировании сети. Настроить множество сервисов можно с помощью специальных программ, например linuxconf, Webmin, средствами администрирования, входящими в GNOME и KDE, или отредактировав конфигурационные файлы. У большинства сервисов есть еще одна возможность — настройка через Web-интерфейс. Существуют и совместимые с ICQ интернет-пейджеры: licq, kicq, GNOMEICQ, micq, в том числе и для текстовой консоли.

Файловые менеджеры

Для пользователей "старой закалки", знакомых еще с DOS, непременным атрибутом работы за компьютером был файловый менеджер, который подменял собой скуку командной строки и черноту экрана. Хотя адепты Linux упорно твердят о полной ненужности файлового менеджера для Linux, тем не менее, спрос порождает предложение. Существуют файловые менеджеры и для нашей операционной системы. Как обычно, есть они и для текстовой консоли, и для X Window. Самый известный и, наверное, один из старейших текстовых файловых менеджеров — Midnight Commander (почти полный эквивалент Norton Commander), изображенный на рис. 2.1.

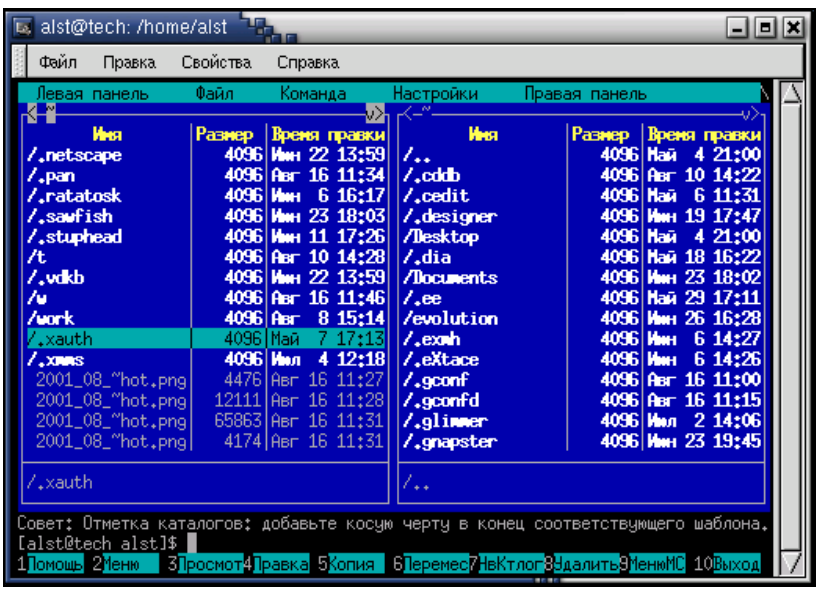


Рис. 2.1. Файловый менеджер Midnight Commander

Он позволяет работать с файлами, редактировать их, осуществлять поиск, взаимодействовать с архивами и т. д. Однако есть и другие достоинства — способность устанавливать пакеты RPM, работать с FTP, просматривать HTML-документы, подключаться к сетевым дискам. Приблизительно такими же возможностями обладает текстовый файловый менеджер XNC. Помимо текстовых, создано довольно много графических файловых менеджеров для X Window, например Nautilus, Kcommander или Kruiser.

Текстовые редакторы

Тут выбор широчайший — от примитивного строчного текстового редактора до мощных пакетов, которые текстовым редактором и назвать трудно. И такое разнообразие наблюдается как для текстовой консоли, так и для X Window. Конечно, в век торжества графики многие удивляются наличию большого числа текстовых консольных редакторов. Однако не следует забывать о широкой распространенности Linux, в том числе и на слабых машинах, куда нет смысла устанавливать объемную графическую оболочку и ресурсоемкий графический текстовый редактор только для того, чтобы откорректировать несколько конфигурационных файлов. Не исключена также необходимость отредактировать тот или иной Web-скрипт на удаленной машине через Интернет. Возможен и совсем неприятный вариант — сбой системы, не позволяющий загрузиться в графическом режиме. Поэтому до сих пор существуют текстовый редактор vi, появившийся в самом начале становления UNIX, и его более функциональные потомки vim, joe, pico, jed, встроенный Midnight Commander и редактор EMACS.

Под X Window еще больше редакторов. Очень много простых, типа Gnotepad, и, конечно, множество мощных текстовых процессоров, часть из которых входит в офисные пакеты. В качестве примера можно привести Kedit, Gedit, Kwrite, Kword, Ted, Abiword, OpenOffice и др. Более подробно о редакторах будет рассказано далее.

Графические оболочки

Неоднократно опровергаемое утверждение, что Linux — чисто текстовая среда, почему-то очень живуче. Хотя по разнообразию графических оболочек (или менеджеров окон) она оставляет далеко позади семейство Windows, да и большинство UNIX-собратьев. В отличие от Windows, в Linux (UNIX) графическая оболочка (X Window) разделена на два приложения: X-сервер и менеджер окон. Сервер в какой-то мере специфичен для аппаратных средств (зависит от видеокарты, шины данных и т. д.) и играет роль "рабочей лошадки", а менеджер окон обеспечивает внешний вид приложений, отрисовку окон, меню и прочих элементов графического интерфейса. Благодаря такой независимости пользователь получает богатейший выбор средств для персонализации своего рабочего места. Можно поставить IceWM или AfterStep и получить легкую и мощную графическую среду (вполне

нормально функционирующую на старших 486-х процессорах), для тех, кому хочется "как в Windows, но лучше" — KDE или GNOME.

И это далеко не предел: менеджеров окон (только самых известных) существует десятка полтора, и каждый из них легко настраивается. Конечно, неопытного пользователя очень смущает текстовая консоль, но можно при инсталляции Linux (или позже) установить загрузку X Window сразу при старте системы. Тем более что практически все текстовые программы или дублируются графическими, или имеют графический интерфейс.

Графические редакторы

В этой категории тоже достаточно много программ от самых простых до очень сложных, практически ничем не уступающих по возможностям CorelDRAW и Photoshop. Как обычно, есть векторные и растровые редакторы. Для примера, упомянем Gimp — мощнейший редактор, портированный, в частности, под Windows, StarDRAW — программа создания рисунков на основе векторной графики, StarImage — программа создания рисунков на основе битовых образов, KimageShop и множество других.

Web-инструментарий

Традиционно лучшим для Web-дизайнера считается простой текстовый редактор, однако многие люди работают в специализированных HTML-редакторах. Для Linux, однако, выбор HTML-редакторов не очень большой. К ним относятся, например, такие программы подготовки HTML-файлов, как OpenOffice, Amaya, GINF, WebMaker (разработка Алексея Дець, Россия) или Quanta Plus (разработка Дмитрия Поплавского и Александра Яковлева, Украина).

Офисные пакеты

Исторически сложилось так, что разработкой полноценного офисного пакета для Linux сообщество озаботилось относительно недавно. По всей видимости, это связано с тем, что только сейчас Linux стала продвигаться на офисные рабочие места, оставаясь до последнего времени серверной или "домашней" ОС. Конечно, и до этого существовали текстовые редакторы, электронные таблицы, органайзеры и программы презентаций. Однако в полноценный офисный пакет они не складывались из-за ряда нерешенных проблем: несовместимости с Microsoft Office, отсутствия единого разработчика, способного создать все составные части пакета, недостаточной интеграции программ от разных разработчиков, нехватки полноценной поддержки русского языка.

Под офисным пакетом будем понимать набор программ, включающих в себя:

- текстовый редактор (процессор);
- программу для работы с электронными таблицами;
- программу обработки электронной почты (в принципе необязательно);
- программу подготовки презентаций;

- одну или несколько программ для работы с изображениями;
- персональный органайзер;
- программу для организации работы в группе и т. д.

В офисный пакет могут входить и другие программы или, наоборот, некоторые из них могут отсутствовать. Но комплект программ можно назвать пакетом только тогда, когда все входящие в него составляющие обладают единым стилем интерфейса и позволяют обмениваться информацией между собой.

При оценке офисных пакетов мы вынуждены сравнивать их с Microsoft Office, поскольку подавляющая часть пользователей, так или иначе, работает именно с ним. Поэтому при выборе программ, которые можно отнести к разряду офисных, обязательно подразумевается совместимость по форматам файлов с Microsoft Office. Даже если пакет полностью работает под Linux, рано или поздно возникнет необходимость отправить партнерам документ в формате Microsoft Office или, наоборот, получить от них такого рода файл. И никому не будет дела до того, что в вашей фирме не признают программное обеспечение от Microsoft. Поэтому рассмотрим офисные пакеты под Linux с учетом приведенных требований.

В настоящее время существует много офисных пакетов как платных, так и с открытым исходным кодом. Начнем с коммерческих пакетов.

Oracle OpenOffice

Пакет начинал свою жизнь как StarOffice. Изначально был разработан немецкой фирмой Star Division, в дальнейшем куплен фирмой Sun Microsystems, а затем его исходные коды были открыты под лицензией GPL для Linux-общественности. В результате появился некоммерческий OpenOffice и остался платный StarOffice. После покупки Sun фирмой Oracle пакет переименовали.

Составляющие OpenOffice (при инсталляции можно отказаться от установки некоторых частей пакета):

- Writer — текстовый процессор;
- Calc — программа работы с электронными таблицами;
- Impress — программа подготовки презентаций;
- Draw — программа создания рисунков на основе векторной графики;
- Base — система управления базами данных.

Совместимость с Microsoft Office удовлетворительная, однако могут возникать проблемы с таблицами и со связанными файлами (например, с внедренными в файл Word документами Excel).

Отличия Oracle OpenOffice от OpenOffice.org:

- шрифты Unicode оптимизированы для отрисовки мелких кеглей;
- набор латинских шрифтов;
- база данных Adabas D;
- шаблоны и типовые документы;
- расширенный набор клипартов;
- расширенная проверка орфографии;
- конвертор макросов Microsoft Office в StarBasic.

OpenOffice.org

Проект, базирующийся на исходном коде StarOffice. Различия в проектах в основном косметические. Поскольку Oracle OpenOffice — коммерческий продукт, в него включается коммерческое ПО (шрифты, средства проверки правописания, различные утилиты) и больше внимания уделяется внешнему виду приложений (дизайн иконок, кнопок и т. п.). В русской редакции OpenOffice большую роль играет команда дистрибутива Alt Linux, на сайте которой всегда можно получить последнюю версию пакета как для Linux, так и для Windows. Пакет OpenOffice Pro (**www.i-rs.ru** — сборка "Инфра-ресурс") имеет углубленную русскую локализацию, улучшенный словарь и проверку правописания, шаблоны и клипарты.

Koffice

Очень динамично развивающийся пакет, являющийся частью проекта KDE.

Составляющие Koffice:

- KSpread — электронные таблицы;
- KPresenter — создание презентаций;
- KChart — создание диаграмм;
- Krita — растровый графический редактор;
- Karbon14 — векторный графический редактор;
- KFormula — математический пакет;
- KWord — WYSIWYG-текстовый редактор;
- Kivio — программа создания диаграмм;
- Kexi — аналог Access;
- Kplato — программа для планирования и управления проектами.

Помимо выдержанного в стиле KDE-интерфейса, отличной интеграции с другими KDE-приложениями и нормальной поддержкой русского языка, что немаловажно, заявлена совместимость с файлами Microsoft Office, а также возможность обработки файлов в форматах CSV и RTF. При этом пакет характеризуется очень простой инсталляцией и достаточно скромными требованиями к ресурсам.

GNOME Workshop

Программы, разработанные под GNOME, составляющие полноценный офис:

- AbiWord — популярный мультиплатформенный текстовый редактор;
- Achtung — программа презентаций;
- Balsa — мощный почтовый клиент;
- Dia — отличное приложение для создания различных диаграмм, аналог Microsoft Visio;
- Eye of GNOME — программа просмотра графических изображений;
- Evolution — мощная программа, аналог Microsoft Outlook;
- Gfax — программа для приема и отправки факсов;
- GIMP — великолепный графический редактор;
- GNOME-DB — средство для работы с БД;

- Gnucash — персональный финансовый менеджер;
- Gnumeric — электронные таблицы;
- Guppi — программа для рисования;
- MrProject — инструмент управления проектами;
- Sketch — редактор векторной графики;
- Sodipodi — редактор векторной графики;
- Toutdoux — инструмент управления проектами.

Базы данных

Вопреки распространенному мнению, под Linux разработано и перенесено большое количество серверов баз данных различного уровня: IBM DB2, Informix, Oracle, Sybase SQL Anywhere, Interbase, FireBird, PostgreeSQL, MySQL, SAP DB. Часть из них некоммерческие и с открытым исходным кодом, остальные, при определенных условиях, можно бесплатно получить или использовать некоторое время.

Эмуляторы Windows

Существует немало эмуляторов Windows (и виртуальных машин) в среде Linux: Citrix MetaFrame, Mainsoft's MainWin, Win4Lin, VMWare, BOSCH, Wine, Cedega, CrossOver и др. Они различаются по функциональным возможностям: одни обеспечивают работу приложений для Windows 9*x*, другие способны запускать еще и продукты для Windows. Есть здесь и сложность — отсутствие полноценной поддержки DirectX. (Wine много функций уже нормально поддерживает. Правда, OpenGL делает это лучше.) Прекрасно запускаются под Linux игры, разработанные в расчете на OpenGL, но большинство игр, для которых требуется DirectX, пока не работают под эмуляторами. Однако, по заявлениям разработчиков, эта проблема будет вскоре преодолена.

Средства разработки программ

Для Linux, как и для UNIX, "родным" языком является, естественно, C/C++, но это совершенно не означает, что кроме них, никаких других компиляторов (или интерпретаторов) языков не существует. Большего разнообразия языков на одной платформе встретить невозможно. Настоящее вавилонское смешение! Трудно найти какой-либо язык, компилятор или интерпретатор, которого не существует для Linux: C/C++, Pascal, Perl, Java, Lisp, Rexx, Fortran и т. д. Не обойдены стороной и интегрированные среды разработки. Событием стал выпуск фирмой Borland интегрированной среды Kylix — Linux-аналога Delphi (Windows).

Kylix

Приложения, написанные в Delphi с использованием специальной библиотеки, можно практически без переделок перенести в Linux. Наряду с коммерческим пакетом Kylix существует и версия для разработки программного обеспечения

с открытым исходным кодом, скачать которую можно бесплатно с Web-сайта фирмы Borland. Впрочем, и здесь есть своя ложка дегтя. Во-первых, при работе Kylix требуется эмулятор Windows — Wine. Это понятно, программисты из Borland облегчили себе перенос Delphi в Linux, но поскольку Wine не до конца реализует Windows-совместимость и постоянно модернизируется, Kylix временами работает нестабильно. И, во-вторых, совместно с вновь созданным в Kylix приложением необходимо распространять некоторые специфические библиотеки. К сожалению, пакет не поддерживается разработчиком (Kylix 3 был выпущен давно и с тех пор ничего нового не появилось). Однако есть похожее решение — связка Free Pascal и Lazarus.

KDevelop

Программа предназначена для разработки приложений под KDE с библиотекой Qt. Позволяет создавать консольные приложения. Обладает интерфейсом, похожим на MS Visual C++ (рис. 2.2). Требует много сторонних приложений типа a2ps, Khexedit, KTranslator и т. д. Встроен довольно удобный интерактивный отладчик. В качестве компилятора использует GNU Compiler Collection, поэтому может работать с языками C,C++, Pasccal, Fortran, Perl, Python, PHP, Java, Ruby и Ada.

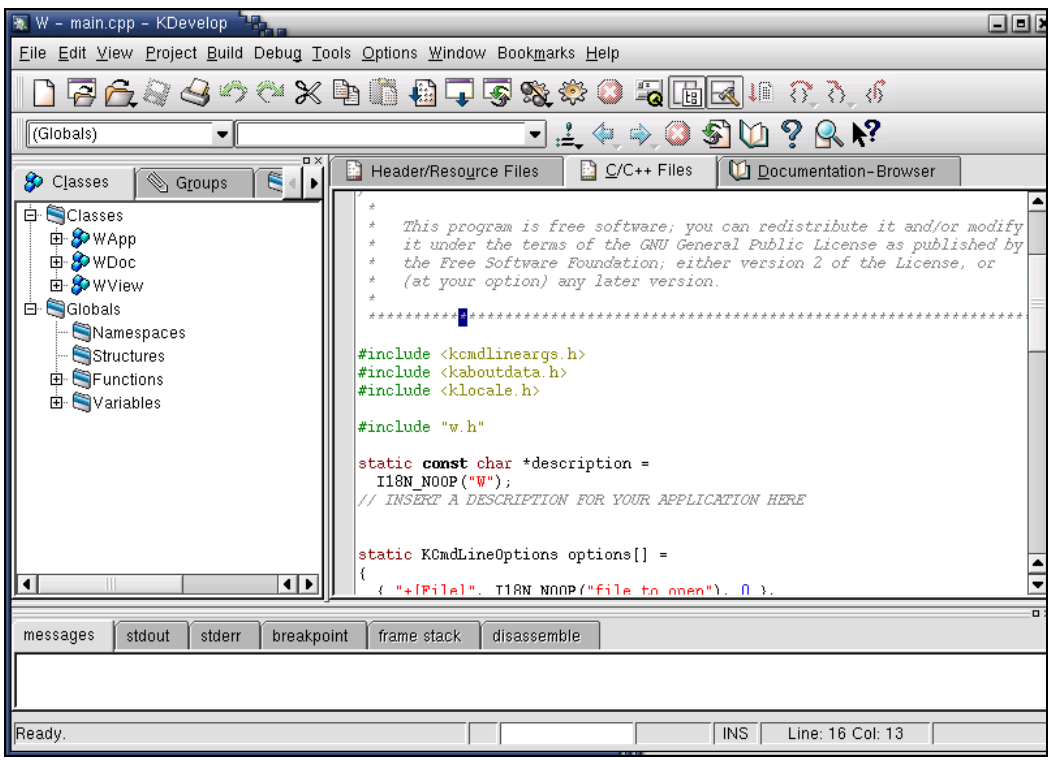


Рис. 2.2. Интегрированная оболочка разработки программного обеспечения KDevelop

Glade

Приложение для визуального создания графических интерфейсов на основе GTK+ (рис. 2.3).

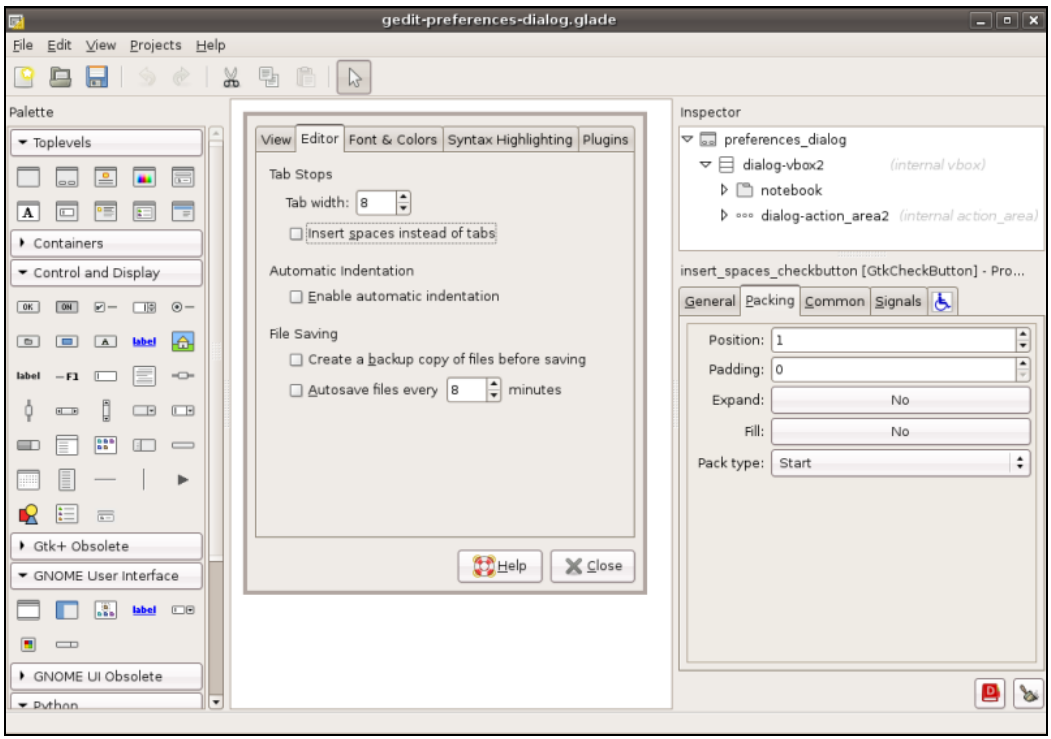


Рис. 2.3. Программа Glade

Rhide

Программа написана на перенесенной из DOS библиотеке Turbo Vision. Поддерживает C, C++, Assembler, Pascal и Fortran. Оболочка для gdb выделена в отдельное приложение, благодаря чему ее можно использовать как Turbo Debugger. Предусмотрена возможность настраивать цвета, компилятор и его опции, языки.

Eclipse

Свободная интегрированная среда разработки модульных кроссплатформенных приложений. Поддерживает несколько языков программирования, ориентирована на групповую разработку приложений. Написана на Java, мультиплатформенна.

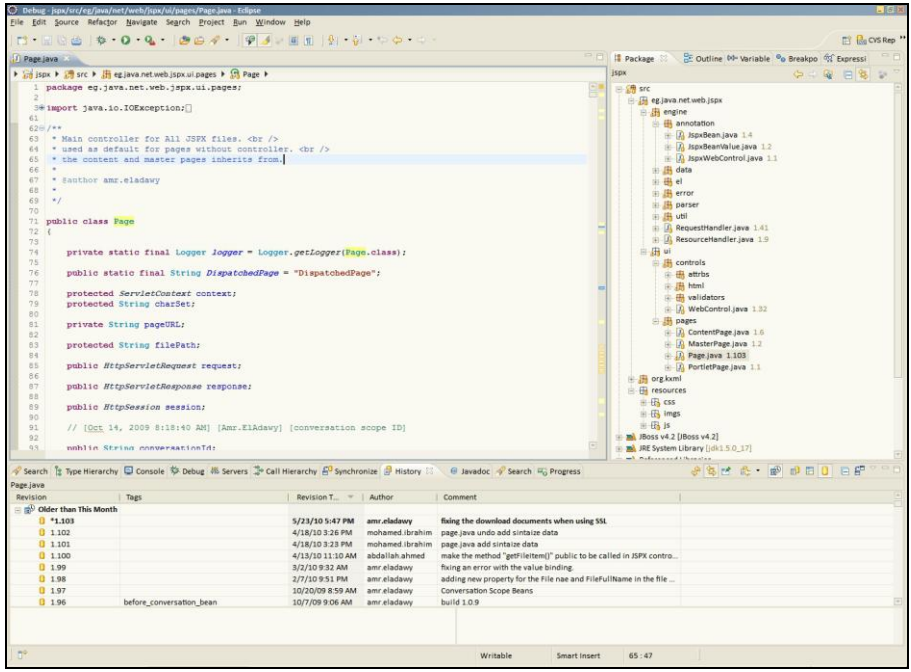


Рис. 2.4. Среда разработки Eclipse

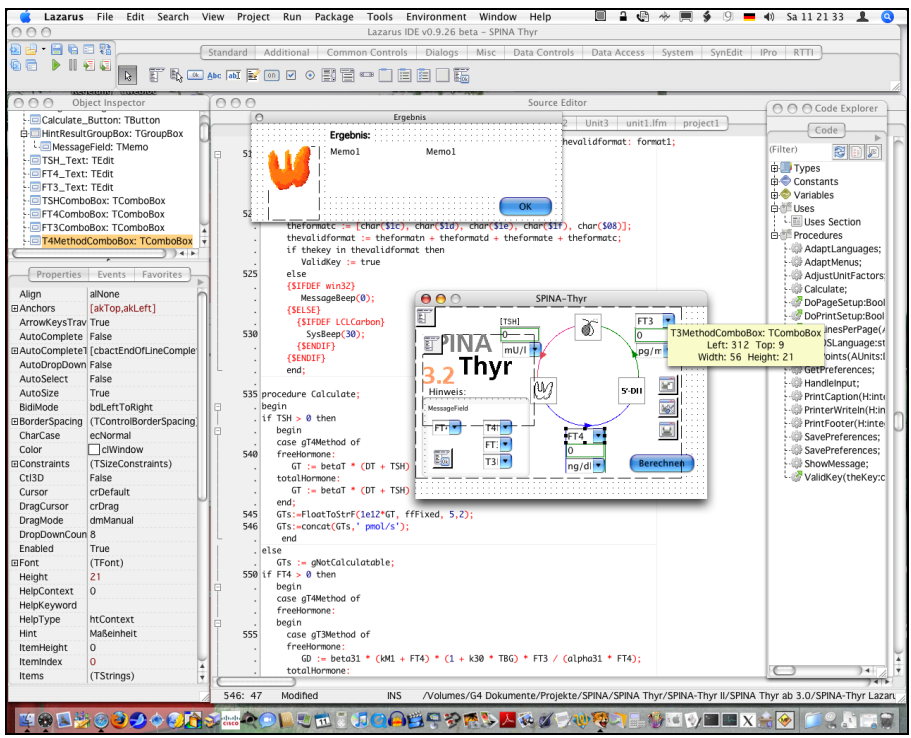


Рис. 2.5. Программа Lazarus

Lazarus

Некоммерческая среда разработки приложений для компилятора Free Pascal. Поддерживает кросс-платформенность. Частично совместима по исходным кодам с Delphi.

Как видите, выбор обширен, и всегда можно найти продукт, удовлетворяющий самому требовательному вкусу.

Мультимедиа-приложения

Аудио

Звуковые средства должны воспроизводить, как минимум, WAV- и MIDI-файлы, MPEG-3, а также обычные аудио-CD.

Времена сложного ручного конфигурирования этих устройств (подробно описанные в литературе), похоже, закончились. По крайней мере, в RedHat и его клонах поддержка звука предполагается по умолчанию. Поддерживаются почти все мало-мальски распространенные устройства. Обычно после инсталляции дистрибутива звуковая карта уже сконфигурирована и вполне работоспособна. Настраивать ее приходится лишь в редких случаях.

С аудиодисками тоже все просто. В состав KDE входит несложный и вполне удобный CD-плеер, аналогичный таковому из комплекта Windows. Кроме того, имеется еще несколько похожих средств как графических, так и консольных, например несколько проигрывателей входят в состав GNOME.

Для управления звуком, как и в других операционных средах, используется микшер. Микшеров под Linux разработано очень много, как консольных, так и графических (рис. 2.6). Например, в составе KDE и GNOME имеется микшер, позволяющий регулировать громкость и баланс при воспроизведении звуков разного типа.

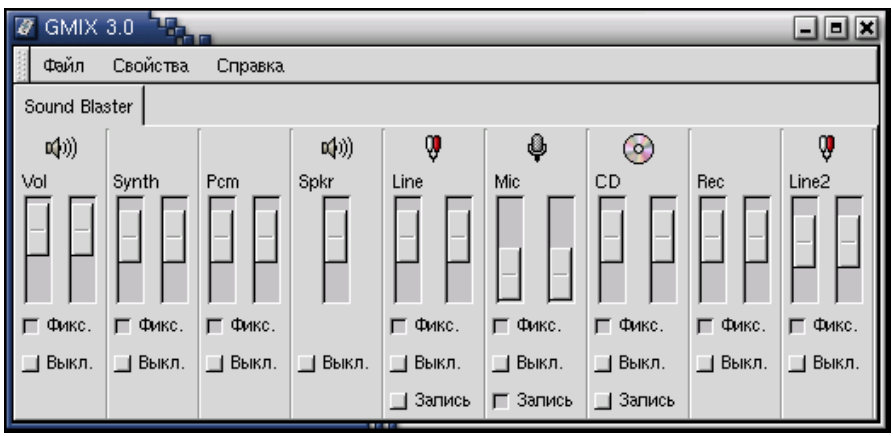


Рис. 2.6. Микшер GMIX, один из многих

KDE и GNOME также включают штатный проигрыватель для воспроизведения WAV-файлов. Существует несколько программ для проигрывания MPEG-3, самая распространенная — mpg123 — консольный проигрыватель, потребляющий очень мало ресурсов. Несколько простеньких MP3-проигрывателей входят в KDE и GNOME, имеются также XMMS (полный функциональный аналог известного Winamp для Windows) и KJukeBox. А если требуется преобразовать аудиодиск в MP3-файлы, есть программы, которые сделают и это (Grip, например), причем можно выбрать различные кодеки, качество оцифровки, получить названия треков из базы данных аудиодисков (при подключении к Интернету). Динамично развивается Vorbis — неплохой кодек для музыки.

Видео

Начнем этот обзор с телевидения, которое принимают на компьютер с помощью TV-тюнера. Наиболее распространенные их модели поддерживаются Linux, правда, не совсем понятна ситуация с USB TV-тюнерами. (Как обычно, не все производители выпускают драйверы для Linux, а разработка сторонними энтузиастами продвигается медленно, из-за отсутствия спецификаций.) А для воспроизведения видеофайлов выбор достаточно большой. Для KDE есть универсальный штатный видеоплеер — aKtion, который считывает форматы AVI, MPEG, Quick Time, а также FLI/FLC-анимации. Он обеспечивает масштабирование (оригинальный размер, удвоенный, максимальный) и полноэкранное воспроизведение (любой из вариантов можно установить по умолчанию) и имеет регулятор громкости. Есть возможность захвата кадров (в формате BMP и еще нескольких). Довольно много всяких настроек.

Для воспроизведения видео, записанного в формате MPEG-4 (DivX), можно воспользоваться программой Mplayer или Xine.

Однако поддержкой специальных плат нелинейного монтажа Linux похвастаться пока не может. Есть драйверы, написанные для плат серии Miro, однако они постоянно совершенствуются и поэтому не до конца стабильны. Маловато программного обеспечения для нелинейного видеомонтажа. Остается надеяться, что положение постепенно исправится.

Игры

С играми ситуация хуже всего. Одно из основных применений домашнего компьютера, помимо мультимедиа, — это игры. Наверное, только из-за них приобретается треть всех домашних компьютеров. Проблема с разнообразием и качеством игр напрямую вытекает из технологии их создания. Для Linux разработано множество небольших игр типа тетриса, пасьянсов, шахмат, нардов, го и реверси, т. е. таких, которые не требуют огромной работы по программированию, написанию сценария, разработке трехмерных моделей и невообразимого количества текстур и рисунков. Как только дело доходит до серьезных игр, сразу образуется вакуум.

Фирмы-разработчики игр не считают сегмент Linux перспективным. Игры разрабатываются в расчете на Sony PlayStation, Nintendo или Windows. А на рынок Linux с играми они выходить не спешат. Тем не менее (вот парадокс) программ-серверов, рассчитанных на Linux для игры через Интернет, достаточно много

(те же Quake, Unreal). До недавнего времени только фирма Id Software выпускала Linux-версии своих игр. Кроме коммерческих, есть несколько игр с открытым исходным кодом, хотя, в основном, это Linux-реализации давно известных коммерческих игр мира Windows. Самые яркие представители здесь — FreeCiv и FreeCraft. Так что, если вы требовательны к разнообразию и качеству игр, к сожалению, Linux пока не для вас. Хотя при помощи эмуляторов можно запускать часть игр, написанных для Windows и некоторых игровых консолей.

Итоги

Как следует из материалов этой главы, Linux отлично справится с всякими серверными приложениями и сервисами. С точки зрения администрирования тоже особых проблем нет. Мнение, что Linux — чисто серверная платформа, и решать на ней офисные задачи невозможно, — только миф. Помимо этого, динамично развиваются офисы KDE, GNOME и OpenOffice. Для домашнего применения картина складывается противоречивая. С одной стороны, налицо отличная поддержка мультимедиа, с другой стороны, — практически полное отсутствие современных игр под Linux. Остается надеяться, что с увеличением числа пользователей Linux производители будут выпускать для них версии своих игр.

Ссылки

- **www.freshmeat.net** — сайт, содержащий большое количество программ для Linux и не только.
- **www.openoffice.org** — официальный сайт OpenOffice.
- **koffice.kde.org** — официальный сайт Koffice.
- **www.kdevelop.org** — официальный сайт KDevelop.
- **www.borland.com** — официальный сайт фирмы Borland, разработчика Kylix.
- **www.gnome.org** — официальный сайт GNOME.
- **www.kde.org** — официальный сайт KDE.
- **www.mozilla.org** — официальный сайт Mozilla.
- **www.opera.com** — сайт фирмы-разработчика Opera.
- **www.linuxdocs.org** — одно из наиболее полных собраний документации о Linux.
- **www.linux-ve.chat.ru** — электронная библиотека, посвященная Linux.
- **www.citforum.ru** — большое собрание русскоязычной документации и книг, в том числе посвященных Linux.
- **www.linux.org.ru** — один из основных русскоязычных сайтов, посвященных Linux.
- **www.applix.com** — сайт фирмы-разработчика Applixware.
- **www.mysql.org** — официальный сайт MySQL.
- **www.interbase.com** — официальный сайт Interbase.
- **www.idsoftware.com** — разработчик игр Doom, Quake, Quake II, Quake III.
- **www.lokigames.com** — сайт фирмы, которая переносит Windows-игры для Linux.
- **www.linuxgames.org.ru** — сайт, посвященный играм для Linux.



Глава 3

Дистрибутивы

Дистрибутивы. Что это такое? Какие они бывают? Чем один дистрибутив лучше другого?

Дистрибутив — это определенный набор программ, утилит и документации, объединенный логичной системой установки и сопровождения программных пакетов, ориентированный на определенную группу пользователей и конкретный тип задач. По большому счету, обладая достаточными знаниями, можно скачать из Интернета ядро операционной системы, загрузчик, драйверы, программное обеспечение и все это установить вручную, а потом долго подгонять и настраивать. Но в следующий раз, когда возникнет необходимость установить систему у другого пользователя, вы дважды подумаете — ставить все это самостоятельно и снова мучиться с настройкой или взять какой-либо дистрибутив и за полчаса установить систему (о настройке мы пока деликатно умолчим, случаи бывают разные).

О пользователях. Условно их можно разделить на начинающих, "продвинутых" и специалистов. Соответственно было бы неплохо иметь для каждой группы свой тип дистрибутива. И дистрибутивы, действительно, в некоторой степени ориентируют на такое разделение пользователей. Есть дистрибутивы, где инсталляция проходит буквально за десять щелчков мышью, а существуют и такие, где очень многое необходимо настраивать вручную. В некоторых дистрибутивах сделана попытка совместить легкость в установке и возможность настройки всего и вся. Кстати, с переходом пользователей из одной группы в другую тяга к тотальной настраиваемости системы возрастает.

Помимо деления по простоте установки и сопровождения, дистрибутивы подразделяются и по назначению. Обычно это офисный (домашний) дистрибутив, сервер для маленького офиса, мощный сервер и, конечно, дистрибутивы специального назначения. Существует ряд многофункциональных дистрибутивов.

Попробуем теперь определить дистрибутив, приемлемый для большинства пользователей. Такой подход весьма субъективен, но все же некоторые требования для большинства дистрибутивов должны быть общими:

- ❒ логичный и удобный набор пакетов;
- ❒ основная часть дистрибутива — стабильные пакеты, все остальные пакеты помечены в документации как экспериментальные; для увеличения стабильности эти пакеты желательно "пропатчить" (внести в них имеющиеся исправления, патчи, от англ. *patch* — исправление, заплатка);

- загрузочный диск с опцией аварийной загрузки;
- возможность устанавливаться с компакт-диска, с жесткого диска и по сети;
- набор программного обеспечения в пакете соответствует целям дистрибутива;
- все программное обеспечение надежно функционирует на любом оборудовании, выпущенном в пределах 2–3 ближайших лет;
- локализация и интернационализация во всех программах;
- полный контроль над системой при установке и настройке пакета администратором;
- все необходимое для инсталляции системы находится на одном компакт-диске, все остальное — на дополнительных (опциональных) дисках.

В свете этих требований более пристально взглянем на проблему локализации и интернационализации. Исторически сложилось так, что языком международного общения является английский. Поэтому большинство дистрибутивов "говорят" на хорошем английском языке. Существуют "французский" и "немецкий" дистрибутивы, и, конечно, парочка русскоязычных дистрибутивов.

По моему мнению, нормально локализованный дистрибутив должен обладать следующими свойствами:

- поддерживать все распространенные языки;
- обеспечивать ввод и вывод символов национальных алфавитов как в текстовом, так и в графическом режиме во всех официальных кодировках или, по крайней мере, в наиболее распространенных;
- иметь локализованные версии всех сопутствующих программ, документации, проверки орфографии и т. п.;
- включать толковое, достаточно обширное локализованное руководство пользователя.

Существует несколько сайтов, которые постоянно отслеживают дистрибутивы, следят за версиями, выкладывают описания и особенности дистрибутивов. Один из них — **distrowatch.com**.

На сегодняшний день есть четыре базовых дистрибутива Linux: Debian, Red Hat, Slackware, Gentoo (Source Based distributive) и множество их потомков, причем некоторые из них уже имеют крайне мало общего с "родителями".

Группа Debian

В этой группе представлены дистрибутивы, исторически и идеологически родственные дистрибутиву Debian, программы распространяются в пакетах формата DEB.

Организация-разработчик — Debian. Web-сайт: **www.debian.org**. Весьма неплохой дистрибутив, существуют три ветви в основном дереве — "stable", "testing" и "unstable". При появлении новой версии программы она помещается в нестабильную ("unstable") ветвь для начального тестирования. Если начальный тест пройден, то программа переходит в тестовую ("testing") ветвь, где подвергается строгому тестированию, продолжающемуся несколько месяцев. Эта ветвь объявляется устойчивой ("stable") после очень продолжительного тестирования. Стабильная ветвь идеальна для серверов, выполняющих критически важные задачи, а на своих пер-

сональных компьютерах многие пользователи предпочитают устанавливать более продвинутые тестовые и нестабильные ветви.

Кратко охарактеризуем некоторые из дистрибутивов группы.

- Adamantix — дистрибутив, уделяющий большое внимание безопасности.
- Amber Linux — дистрибутив, разработанный и поддерживаемый латвийцами.
- Damn Small Linux — LiveCD-дистрибутив размером 50 Мбайт.
- StormLinux — относительно маленький, пригодный для не очень мощных машин. Web-сайт: **www.stormlinux.com.**
- Symphony OS — относительно новый дистрибутив, отличающийся дружелюбным пользовательским интерфейсом.
- Knoppix — разработчик Klaus Knopper. Дистрибутив, распространяемый на LiveCD, также может устанавливаться на жесткий диск. Пожалуй, самый популярный LiveCD-дистрибутив в мире. Web-сайт: **www.knoppix.org.**
- Bonzai — разработчик Marcus Moeller. Наиболее свежий Debian с самой современной версией ядра и KDE, умещающийся на диск объемом 180 Мбайт. Web-сайт: **http://www.gnulinux.de.**
- Ubuntu — дистрибутив, базирующийся на Debian и субсидируемый фирмой Canonical Ltd. Использует собственные репозитории. "Восходящая звезда" среди дистрибутивов на базе Debian. Существуют несколько основных разновидностей дистрибутива, которые отличаются набором программ, устанавливаемых по умолчанию:
  - Kubuntu — использует KDE;
  - Xubuntu — Xfce;
  - Edubuntu — предназначен для учебных заведений;
  - nUbuntu — предназначен для анализа сетевой безопасности;
  - Ubuntu Studio — ориентирован для использования в качестве медиа-студии.

Группа Red Hat

В этой группе представлены дистрибутивы, исторически и идеологически родственные дистрибутиву Red Hat системой пакетов RPM и идеологией и схожестью инсталляции.

- Red Hat — разработан фирмой Red Hat. На сегодня это самый популярный дистрибутив для коммерческого использования. Обладает приемлемой русификацией и неплохой поддержкой. В данный момент все внимание направленно на серверную версию. Web-сайт: **www.redhat.com.**
- Fedora — это настольная версия Red Hat, отправленная в "свободное плавание". Red Hat осуществляет хостинг, консультацию сборщиков дистрибутива, частичное финансирование. Web-сайт: **www.fedoraproject.org**.
- ASP Linux (ASP, Advanced Server Platform) — разработан фирмой SWsoft. Практически первый коробочный российский дистрибутив, отличается легкостью установки и настройки. Web-сайт: **www.asplinux.ru.**
- AltLinux — разработан фирмой Alt Linux Team. На сегодняшний день, пожалуй, самый популярный в России дистрибутив. Существует несколько его вариантов,

предназначенных для обычного офисного применения, для разработчика и в качестве защищенного сервера. Фирма поддерживает огромный репозиторий ПО, называемый Sisyphus. Web-сайт: **www.altlinux.ru**.

- CentOS — дистрибутив, целью которого является полная совместимость с Red Hat Enterprise Linux без использования проприетарных программ.
- Mandriva — разработан фирмой Mandriva Linux. Обладает простой инсталляцией, неплохой русификацией, ориентируется на KDE и мультимедиа. Web-сайт: **www.linuxmandriva.com/ru**.
- BestLinux — разработан фирмой SOT Finish Software Engineering. Отличается хорошей поддержкой русского языка и удобным графическим инсталлятором. Web-сайты: **www.bestlinux.net/ru**, **www.bestlinux.net**.
- TurboLinux — разработан фирмой TurboLinux Inc. Средний дистрибутив, ничем особо не отмечен. Web-сайт: **www.turbolinux.com**.
- Lycoris — разработчик Joseph Cheek. Цель дистрибутива — легкость использования, что позволит сделать переход от Windows к Linux настолько безболезненным, насколько это возможно. Web-сайт: **www.lycoris.com**.
- OpenWall Linux — серверный дистрибутив с упором на высокую безопасность системы.

Группа Slackware

В этой группе представлены дистрибутивы, исторически и идеологически родственные дистрибутиву Slackware, формат пакетов — TGZ.

- Slackware — разработка Patrick Volkerding, Walnut Creek CDROM. Один из старейших дистрибутивов. Сегодня относительно мало распространен в связи с тем, что представляет собой конструктор для опытного пользователя Linux. В результате настраивается всё и вся, ставится только то, что указано (можно получить очень компактную систему). Web-сайт: **www.slackware.com**.
- SuSE Linux — разработан фирмой Novell. Основное его преимущество — огромное количество включенных в дистрибутив программ и поддержка влиятельной фирмы. Web-сайт: **www.novell.com/linux**.
- OpenSUSE — после покупки SuSE почитатели дистрибутива создали ветку, OpenSUSE, независимую от Nowell. Качественный дистрибутив с продуманной и удобной системой управления Yast2.
- VectorLinux — в отличие от родителя ориентирован на рядового пользователя. Помимо инсталляции есть и LiveCD-версия. Web-сайт: **www.vectorlinux.org**.

Группа Gentoo

Это группа (я ее так назвал, поскольку именно Gentoo получил популярность, а вслед стали появляться похожие дистрибутивы) дистрибутивов, основанных на исходном коде. Идея, заложенная в основе Gentoo, заключается в компилировании всех программ на компьютере пользователя с учетом аппаратной конфигурации системы. Основное преимущество заключается в том, что все программное обеспе-

чение оказывается хорошо оптимизированным под конкретную архитектуру. Однако установка Gentoo — это утомительный и долгий процесс.

- Gentoo Linux — разработан Daniel Robbins, бывшим создателем Stampede Linux и FreeBSD. Система "портов" FreeBSD вдохновила автора включить их в Gentoo под именем "portage". Web-сайты: **www.gentoo.org**, **www.gentoo.ru**.
- Calculate Linux — идеально подходит администраторам, обслуживающим десятки или сотни компьютеров под управлением ОС Linux/UNIX, а также продвинутым пользователям. Попытка средствами Linux сделать Active Directory. Web-сайт: **www.calculate-linux.ru**.
- Linux From Scratch ("линукс с самого начала") — это не дистрибутив в общепринятом смысле слова, а книга Герарда Бикманса [и др.], описывающая процесс сборки своего дистрибутива из исходных кодов. Книга доступна по адресу **www.linuxfromscratch.org**. Для сборки необходимо иметь готовую систему с компилятором и соответствующими библиотеками. (К книге официально прилагается LiveCD с комплектом исходных файлов и средой для сборки пакетов.) В дополнение к этому есть проекты-спутники:
  - Beyond Linux From Scratch — расширение функциональности LFS;
  - Hardened Linux From Scratch — доработка LFS для увеличения безопасности;
  - Automated Linux From Scratch — инструменты для автоматизации и управления установкой LFS и BLFS;
  - Cross Linux From Scratch — утилиты и настройки для кросс-компиляции. Все эти проекты можно найти на том же Web-сайте, где и книга.

Дистрибутивы LiveCD

В последнее время получили широкое распространение так называемые LiveCD дистрибутивы. Это дистрибутивы Linux, которые поставляются на загружаемом компакт-диске, не требуют установки на жесткий диск и работают прямо с компакт-диска.

Такого типа дистрибутивы есть как офисной направленности (графическая среда, поддержка сети, Интернета, почты, офисные пакеты), так и специализированные (набор диагностических и административных программ, системы-маршрутизаторы, и даже системы для просмотра фильмов и прослушивания музыки). Среди этого типа выделяются "супермодные" дистрибутивы, помещающиеся на мини-компакт-диски (185 Мбайт) и даже на диски-визитки (50 Мбайт).

Практически все современные дистрибутивы имеют и версию LiveCD, как официальную, так и неофициальную, собранную энтузиастами. Тем не менее я позволю себе упомянуть некоторые дистрибутивы особо.

- Knoppix — разработчик Klaus Knopper. Дистрибутив, распространяемый на LiveCD, который можно установить и на жесткий диск. Пожалуй, это самый популярный дистрибутив LiveCD в мире. Отличная поддержка оборудования, возможность использования в качестве "спасательного" диска. Помимо всего, в комплекте идет KNOPPIX Remastering HOWTO, что позволяет самостоятельно видоизменять дистрибутив, чем и пользуются многочисленные "отпрыски". Web-сайт: **www.knopper.org**.

- Cool Linux — разработчик Андрей Великоредчанин. Основная цель дистрибутива — предоставить большой набор диагностического и восстановительного ПО. Отличительная особенность — возможность записи компакт-дисков при условии, что в системе помимо CD-ROM установлен CD-RW. Web-сайт: **www.coollinux.sourceforge.net**.
- Slackware LiveCD — система базируется на Slackware. В общем, в ней нет ничего принципиально нового. Web-сайт: **http://www.slackware-live.org**.
- Fedora LiveCD — диск идет в комплекте дистрибутива Fedora.
- Eagle Linux — ISO-образ занимает всего чуть более 22 Мбайт. Содержит базовый набор консольных утилит, достаточный для первоначального знакомства с системой, в том числе для настройки и работы в сети. Web-сайт: **http://eaglelinux.w32.net**.
- LNX-BBC — разработан Linux Bootable Business Card. Система представляет собой маленький дистрибутив, который можно записать на компакт-визитку. Можно использовать как спасательный или ознакомительный, он включает в себя много диагностических утилит и инструментальных средств конфигурации. Web-сайт: **http://www.lnx-bbc.org**.
- MoviX — дистрибутив предназначен для просмотра видео и прослушивания музыки. Поддерживаются все форматы, о которых "знает" mplayer: avi, DivX, mpeg, mp3, ogg/vorbis и некоторые другие. Web-сайт: **http://movix.sourceforge.net**.
- GeeXboX — подобие MoviX с поддержкой множества графических плат, TV-out, средств проигрывания файлов по сети.

Дистрибутивы USB Flash

По большей части это разновидности дистрибутивов LiveCD, хотя есть варианты, родственные дискетным дистрибутивам. Практически все компьютеры допускают загрузку с внешних USB-устройств. В качестве примера можно упомянуть продаваемый Mandriva 2 GB Flash-накопитель с установленным Mandriva Linux. Установлено KDE, Open Office, Mozilla Amarok, K3b, Gimp и многое другое. Другой пример — Domn Small Linux — занимает всего 50 Мбайт. На самом деле, если есть LiveCD, то не составляет труда перенести файлы с компакт-диска на Flash-носитель.

Дискетные дистрибутивы

В основном этот тип дистрибутивов необходим в двух случаях: для создания спасательных дистрибутивов и специализированных систем, например, маршрутизаторов. Причем подобные дистрибутивы ориентированы на старое "железо", не поддерживающее загрузку с компакт-дисков. Поскольку такого "добра" становится все меньше и меньше, большинство проектов в настоящий момент не развиваются.

- 2–Disk Xwindow Linux — дистрибутив базируется на Debian 2.2 и состоит из двух дискет. Идея проекта — дать пользователю минимальные инструментальные средства, требуемые для UNIX. Содержит X Window, PPP для дозвона к провайдеру, оконный менеджер alloywm, Web-браузер chimera, chat, файловый менеджер xfm, iptables, демоны inetd и crond, DHCP — клиент и сервер, простой текстовый редактор xedit и, наконец, xpaint. При необходимости можно скачать

и загрузить дополнительные модули ядра для поддержки сетевых протоколов и устройств, т. к. дисковод после загрузки освобождается. Минимальные системные требования: 486DX с 32 Мбайт ОЗУ. Web-сайт: **http://www.thepub.nildram.co.uk/mirrors/2diskxwin/**.

- Alfalinux — разработан Giancarlo Erra. Дистрибутив состоит из двух дискет и базируется на Slackware, содержит только стандартные утилиты и ядро ОС. Web-сайт: **http://alfalinux.sourceforge.net**.
- Embedded Linux — однодискетный, легко конфигурируемый дистрибутив. Содержит HTTP+CGI-, FTP-, Telet- и TFTP-серверы, а также lynx и snarf. Поддерживает PPP dialup и Ethernet, имеется небольшой SNMP-агент. Системные требования: i386 CPU и 16 Мбайт ОЗУ. Возможна установка на раздел жесткого диска. Web-сайт: **http://www.dobit.com/emblin**.
- Ulric's Router Construction Kit — дистрибутив позволяет собрать свой роутер на дискете, используя ядра серии 2.4.*x*. Web-сайт: **http://siag.nu/urck**.
- Linux Router Project — разработчик Dave Cinege. Дистрибутив предназначен для построения маршрутизатора (проект не развивается с 2003 года, есть ответвление — LEAF Project **http://leaf.sourceforge.net/**). Web-сайт: **http://linuxrouter.org**.
- Freesco — очень активный проект маршрутизатора с широким набором функций и легкой настройкой. Web-сайты: **http://www.freesco.org/**, **http://freesco.linux.kiev.ua/**.

Это далеко не полный список дистрибутивов. Выбор здесь обширный, и он за вами. Конечно, у автора есть свои предпочтения — Fedora Linux. Многие с этим не согласятся, и это их право. Как написано в FAQ по Linux, нужно выбирать тот дистрибутив, с которым работает ваш знакомый специалист. Намного проще решить любой вопрос с помощью знающего человека, чем просматривать горы литературы или рыться в Интернете (хотя книги и Интернет тоже никто не отменял). В плане простоты и удобства инсталляции и обновления системы можно посоветовать дистрибутивы, основанные на Red Hat. Локализация (русификация) достаточно хорошо выполнена у российских и украинских дистрибутивов.

Ссылки

- **distrowatch.com** — один из сайтов, отслеживающих дистрибутивы.
- **www.debian.org** — сайт фирмы Debian.
- **www.redhat.com** — сайт фирмы Red Hat.
- **www.slackware.com** — сайт с описанием дистрибутива Slackware.
- **www.gentoo.org**, **www.gentoo.ru** — сайты с описанием дистрибутива Gentoo.
- **www.knoppix.org** — официальный сайт дистрибутива Knoppix.
- **www.ubuntu.com** — сайт производителя группы дистрибутивов ubuntu.
- **fedoraproject.org** — сайт дистрибутива Fedora.
- **www.asplinux.ru** — сайт российского дистрибутива Asplinux.
- **www.altlinux.ru** — сайт российского дистрибутива ALT Linux.
- **opensuse.org** — сайт дистрибутива OpenSUSE.
- **www.linuxfromscratch.org** — проект, позволяющий построить шаг за шагом из исходных кодов собственную Linux-систему, оптимизированную под вашу аппаратуру.



Часть II

Базовая информация о Linux



Глава 4

Работа в сети. Основные понятия

В этой главе будут рассмотрены базовые понятия, лежащие в основе всего последующего изложения. Как уже упоминалось, "компьютер — это сеть". С рассмотрения основных сведений о работе в сети мы и начнем.

Модели сетевых взаимодействий

Как и любая сложная система, сеть должна опираться на стандарты, без которых невозможно ее нормальное функционирование. За последние двадцать лет было создано множество концепций сетевых взаимодействий, однако наибольшее распространение получили всего две:

- ❒ модель взаимодействия открытых систем (OSI);
- ❒ модель сетевого взаимодействия TCP/IP.

Терминология

Чтобы облегчить понимание содержимого этой главы, приведем основные термины (табл. 4.1).

Таблица 4.1. *Базовые сетевые термины*

Термин	Определение
Датаграмма	Пакет данных. Обозначает единицу информации при сетевом обмене
DNS (Domain Name Service, служба доменных имен)	Специально выделенные компьютеры, которые осуществляют поиск соответствия символического имени хоста и цифрового адреса хоста
Интернет	Глобальная компьютерная сеть, основанная на семействе протоколов TCP/IP
FTP (File Transfer Protocol, протокол передачи файлов)	Используется для приема и передачи файлов между двумя компьютерами
IP (Internet Protocol, протокол Интернета)	Основа основ семейства протоколов TCP/IP. Практически любой протокол из этого семейства базируется на протоколе IP
NFS (Network File System, сетевая файловая система)	Система виртуальных дисков, позволяющая клиентским компьютерам использовать каталоги сервера

Таблица 4.1 *(окончание)*

Термин	Определение
NIC (Network Information Center, Сетевой информационный центр)	Организация, которая отвечает за администрирование и раздачу сетевых адресов и имен
Узел (Node, Host)	Компьютер в сети. Название применимо как к клиенту, так и к серверу
OSI (Open System Interconnection, взаимодействие открытых систем)	Модель взаимодействия открытых систем
RFC (Request For Comments, запрос для пояснений)	Стандарты протоколов Интернета и их взаимодействия
RIP (Routing Information Protocol, протокол маршрутизации информации)	Протокол, используемый для обмена информацией между маршрутизаторами
SMTP (Simple Mail Transfer Protocol, простой протокол передачи электронной почты)	Используется для обмена электронной почтой
SNMP (Simple Network Management Protocol, простой протокол управления сетью)	Служит для управления сетевыми устройствами
TCP (Transmission Control Protocol, протокол управления передачей)	Протокол управления передачей. Предназначен для надежной передачи данных
Telnet	Протокол, осуществляющий удаленное сетевое подключение к компьютеру, эмулирующее терминал
UDP (User Datagram Protocol, протокол пользовательских датаграмм)	Протокол пользовательских датаграмм для обмена блоками информации без установления соединения

Модель взаимодействия открытых систем (OSI)

Еще в 1983 году Международная организация по стандартизации (International Organization for Standardization, ISO) разработала стандарт взаимодействия открытых систем (Open System Interconnection, OSI).

В результате получилась семиуровневая модель.

1. Физический уровень (Physical Level).
2. Уровень данных (Data Link Level) (в литературе часто называется "Канальный уровень").
3. Сетевой уровень (Network Level).
4. Транспортный уровень (Transport Level).
5. Уровень сессии (Session Level).
6. Уровень представления (Presentation Level).
7. Уровень приложения (Application Level).

Первый уровень связан напрямую с аппаратурой, последующие — все более и более абстрагируются от особенностей физической среды передачи информации.

Каждый уровень модели OSI решает свои задачи, использует сервисы предыдущего уровня и, в свою очередь, предоставляет сервисы следующему уровню. Согласно этой модели, уровни не могут "перескакивать" через соседей, например транспортный уровень не может непосредственно пользоваться сервисом физического уровня, он обязан пройти по цепочке: Сетевой уровень → Уровень данных → Физический уровень. В табл. 4.2 приведено описание уровней сетевой модели OSI.

На каждом уровне блоки информации имеют собственное название (табл. 4.3).

Таблица 4.2. *Уровни сетевой модели OSI*

Уровень	Название	Описание
1	Физический уровень	Отвечает за физическое подключение компьютера к сети. Определяет уровни напряжения, параметры кабеля, разъемы, распайку проводов и т. п.
2	Уровень данных	Физически подготавливает данные для передачи (разбивая их на кадры определенной структуры) и преобразует обратно во время приема (восстанавливая из кадров)
3	Сетевой уровень	Маршрутизирует данные в сети
4	Транспортный уровень	Обеспечивает последовательность и целостность передачи данных
5	Уровень сессии	Устанавливает и завершает коммуникационные сессии
6	Уровень представления	Преобразует данные и обеспечивает их передачу в универсальном формате
7	Уровень приложения	Осуществляет связь между приложением и процессом сетевого взаимодействия

Таблица 4.3. *Название блока информации в модели*

Уровень	Название уровня	Название блока информации
1	Физический уровень	Бит
2	Уровень данных	Кадр (пакет)
3	Сетевой уровень	Датаграмма
4	Транспортный уровень	Сегмент
5, 6, 7	Уровень приложения	Сообщение

Несмотря на то, что OSI является международным стандартом и на его основе правительство США выпустило спецификации GOSIP (Government Open Systems Interconnection Profile, Государственный регламент взаимодействия открытых систем),

у производителей программного обеспечения стандарт OSI широкой поддержки не получил. Это объясняется несколькими причинами:

- ❒ длительное время процедуры принятия стандарта;
- ❒ его "оторванность" от реалий;
- ❒ наличие большого числа уровней трудно для реализации и приводит к потере производительности;
- ❒ широчайшее распространение протокола TCP/IP и нежелание потребителей отказываться от него.

В результате спецификации OSI сегодня — это, в основном, страницы в учебнике, в реальной жизни они не применяются.

Модель сетевого взаимодействия TCP/IP

Архитектура семейства протоколов TCP/IP (Transmission Control Protocol / Internet Protocol, протокол управления передачей / интернет-протокол) основана на представлении, что коммуникационная инфраструктура содержит три вида объектов: процессы, хосты и сети.

Основываясь на этих трех объектах, разработчики выбрали четырехуровневую модель:

1. Уровень сетевого интерфейса (Network interface layer).
2. Уровень межсетевого интерфейса — интернета* (Internet layer).
3. Транспортный уровень (Host-to-host Layer).
4. Уровень приложений/процессов (Application/process layer).

Сопоставление сетевых моделей OSI и TCP/IP

Нетрудно заметить, что модели TCP/IP и OSI различаются (табл. 4.4).

Таблица 4.4. *Соответствие модели TCP/IP и модели OSI*

TCP/IP	OSI
Уровень приложений	Уровень приложений
	Уровень представления
	Уровень сеанса
Транспортный уровень	Транспортный уровень
Межсетевой уровень (интернет)	Сетевой уровень
Уровень сетевого интерфейса	Уровень канала данных
	Физический уровень

Как видно из таблицы, уровень сетевого интерфейса сетевой модели TCP/IP соответствует сразу двум уровням сетевой модели OSI, а уровень приложений сетевой модели TCP/IP — трем уровням сетевой модели OSI.

* Здесь "интернет" — термин, указывающий на межсетевой характер взаимодействия, а отнюдь не Глобальная сеть Интернет. — *Ред.*

Сетевые протоколы

В этом разделе мы рассмотрим различные сетевые протоколы, используемые в современной компьютерной индустрии.

Семейство протоколов TCP/IP

Семейство TCP/IP включает следующие протоколы:

- межсетевой протокол (Internet Protocol, IP — протокол интернета) — соответствует уровню интернет-модели TCP/IP. Отвечает за передачу данных с одного хоста на другой;
- межсетевой протокол управления сообщениями (Internet Control Message Protocol, ICMP) — отвечает за низкоуровневую поддержку протокола IP, включая подтверждение получения сообщения, сообщения об ошибках и многое другое;
- протокол преобразования адресов (Address Resolution Protocol, ARP) — выполняет преобразование логических сетевых адресов в аппаратные MAC-адреса (Media Access Control). Соответствует уровню сетевого интерфейса;
- протокол пользовательских датаграмм (User Datagram Protocol, UDP) — обеспечивает пересылку данных без проверки с помощью протокола IP;
- протокол управления передачей (Transmission Control Protocol, TCP) — обеспечивает пересылку данных (с созданием сессии и проверкой передачи данных) с помощью протокола IP;
- множество протоколов уровня приложений (FTP, Telnet, IMAP, SMTP и др.).

Схема протоколов семейства TCP/IP представлена в табл. 4.5.

Таблица 4.5. *Схема семейства протоколов TCP/IP*

<table>
<tr><td>Уровень приложений</td><td>FTP</td><td>SMTP</td><td>NFS</td><td>SNMP</td></tr>
<tr><td>Транспортный уровень</td><td colspan="2">TCP</td><td colspan="2">UDP</td></tr>
<tr><td>Межсетевой уровень (интернет)</td><td colspan="2">IP</td><td>ARP/RARP</td><td>ICMP</td></tr>
<tr><td rowspan="2">Уровень сетевого интерфейса</td><td colspan="4">Ethernet, FDDI, ATM</td></tr>
<tr><td colspan="4">Витая пара, коаксиальный кабель, оптический кабель и т. п.</td></tr>
</table>

Протоколы межсетевого уровня (интернет)

Протоколы межсетевого уровня (интернет) — базовые в семействе протоколов TCP/IP — TCP/IP, ARP/RARP и ICMP.

Протокол IP

Первоначальный стандарт IP разработан в конце 1970-х годов и не был рассчитан на огромное количество хостов, которое сейчас находится в Интернете. Поэтому в настоящее время утвержден новый стандарт IP (в литературе часто старый

стандарт упоминается как IPv4, а новый — как IPv6). Однако массового применения он пока не нашел из-за того, что многие программные и аппаратные средства не способны работать с IPv6, поэтому мы здесь будем рассматривать, в основном, протокол IPv4.

Формат пакета IPv4

Пакет IP состоит из заголовка и поля данных. *Заголовок* пакета имеет следующие поля:

- *Номер версии* (VERS) — указывает версию протокола IP. Сейчас повсеместно используется версия 4 и готовится переход на версию 6;
- *Длина заголовка* (HLEN) пакета IP — занимает 4 бита и указывает значение длины заголовка, измеренное в 32-битовых словах. Обычно длина заголовка 20 байтов (пять 32-битовых слов), но при возрастании объема служебной информации она может быть увеличена за счет дополнительных байтов в поле *Резерв* (IP OPTIONS);
- *Тип сервиса* (SERVICE TYPE) — занимает 1 байт и задает приоритетность пакета и вид критерия выбора маршрута. Первые три бита этого поля образуют подполе приоритета пакета (PRECEDENCE). Приоритет может иметь значения от 0 (нормальный пакет) до 7 (пакет управляющей информации). Поле *Тип сервиса* содержит также три бита, определяющие критерий выбора маршрута. Установленный бит D (delay) говорит о том, что маршрут должен выбираться для минимизации задержки доставки данного пакета, бит T — для максимизации пропускной способности, а бит R — для максимизации надежности доставки;
- *Общая длина* (TOTAL LENGTH) — занимает 2 байта и указывает общую длину пакета с учетом заголовка и поля данных;
- *Идентификатор пакета* (IDENTIFICATION) — занимает 2 байта и используется для распознавания пакетов, образовавшихся путем фрагментации исходного пакета. Все фрагменты должны иметь одинаковое значение этого поля;
- *Флаги* (FLAGS) — занимает 3 бита и указывает на возможность фрагментации пакета (установленный бит Do not Fragment, DF запрещает маршрутизатору фрагментировать данный пакет), а также на то, является ли данный пакет промежуточным или последним фрагментом исходного пакета (установленный бит More Fragments, MF говорит о том, что пакет переносит промежуточный фрагмент);
- *Смещение фрагмента* (FRAGMENT OFFSET) — занимает 13 битов и служит для указания в байтах смещения поля данных этого пакета от начала общего поля данных исходного пакета, подвергнутого фрагментации. Используется при сборке/разборке фрагментов пакетов при передачах их между сетями с различными величинами максимальной длины пакета;
- *Время жизни* (TIME TO LIVE) — занимает 1 байт и указывает предельный срок, в течение которого пакет может перемещаться по сети. Время жизни данного пакета измеряется в секундах и задается источником передачи средствами протокола IP. На шлюзах и в других узлах сети по истечении каждой секунды из текущего времени жизни вычитается единица, единица вычитается также при

каждой транзитной передаче (даже если не прошла секунда). По истечении времени жизни пакет аннулируется;

- *Идентификатор протокола верхнего уровня* (PROTOCOL) — занимает 1 байт и указывает, какому протоколу верхнего уровня принадлежит пакет (например, это могут быть протоколы TCP, UDP или RIP);
- *Контрольная сумма* (HEADER CHECKSUM) — занимает 2 байта, она рассчитывается по всему заголовку;
- *Адрес источника* (SOURCE IP ADDRESS) и *Адрес назначения* (DESTINATION IP ADDRESS) — имеют одинаковую длину (32 бита) и структуру;
- *Резерв* (IP OPTIONS) — необязательное поле, необходимое только при отладке сети. Это поле состоит из нескольких подполей, имеющих один из восьми предопределенных типов. Так как число подполей может быть произвольным, то в конце поля *Резерв* должно быть добавлено несколько байтов для выравнивания заголовка пакета по 32-битовой границе.

Максимальная длина поля данных пакета ограничена разрядностью поля, определяющего эту величину, и составляет 65 535 байтов, однако при передаче по сетям различного типа длина выбирается с учетом максимальной длины пакета протокола нижнего уровня, несущего IP-пакеты. В большинстве локальных и глобальных сетей определяется такое понятие, как максимальный размер поля данных кадра, в который должен разместить свой пакет протокол IP. Эту величину обычно называют максимальной единицей транспортировки — MTU (Maximum Transfer Unit). К примеру, сети Ethernet имеют значение MTU, равное 1500 байтов, сети FDDI — 4096 байтов.

IP-маршрутизаторы не собирают фрагменты в более крупные пакеты, даже если на пути встречается сеть, допускающая такое укрупнение. Это связано с тем, что отдельные фрагменты сообщения могут перемещаться в интерсети по различным маршрутам.

При приходе первого фрагмента пакета узел назначения запускает таймер, который определяет максимально допустимое время ожидания прихода остальных фрагментов этого пакета. Если время истекает раньше прибытия последнего фрагмента, то все полученные к этому моменту фрагменты пакета отбрасываются, а в узел, пославший исходный пакет, с помощью протокола ICMP направляется сообщение об ошибке.

Протокол IPv6

Основные причины создания IPv6:

- протокол IPv4 разрабатывался в конце 1970-х годов с учетом существовавшей на тот момент сетевой инфраструктуры и аппаратного обеспечения. С того времени производительность массовых компьютеров увеличилась в десятки раз, и во столько же увеличилась пропускная способность сетей;
- появление приложений, передающих через Интернет данные в реальном времени (звук, видео), чувствительных к задержкам передачи пакетов, поскольку при задержках или пропадании пакетов звук и видеоизображения искажаются. Особенность подобных приложений — передача очень больших объемов информации.

Однако в IPv4 не было предусмотрено специального механизма резервирования полосы пропускания или механизма приоритетов;

- бурное развитие сети Интернет. Наиболее очевидное следствие — почти полное истощение адресного пространства Интернета, определяемого полем адреса IP в четыре байта. Конечно, были разработаны механизмы компенсации нехватки адресов, однако это не решает проблему.

Основное предложение по модернизации протокола IP — разработка группы IETF (Internet Engineering Task Force, группа решения задач межсетевого взаимодействия). В предложении IETF протокол IPv6 оставляет неизменными основные принципы IPv4. К ним относятся датаграммный метод работы, фрагментация пакетов, разрешение отправителю задавать максимальное число хопов (*хоп* — количество пересылок пакета от одного сетевого интерфейса к другому, иногда называется временем жизни пакета) для своих пакетов.

Однако в деталях реализации протокола IPv6 имеются существенные отличия от IPv4:

- более длинные 128-битовые адреса (16 байтов);
- гибкий формат заголовка. Вместо заголовка с полями фиксированного размера (за исключением поля Резерв) IPv6 использует базовый заголовок фиксированного формата плюс набор необязательных заголовков различного формата;
- поддержка резервирования пропускной способности;
- поддержка расширяемости протокола. Это одно из наиболее значительных изменений в подходе к построению протокола — от полностью детализированного описания к протоколу, который разрешает поддержку дополнительных функций.

Адресация в IPv6

Адреса в IPv6 имеют длину 128 битов или 16 байтов. Версия 6 обобщает специальные типы адресов версии 4 в следующих типах адресов:

- Unicast — индивидуальный адрес. Определяет отдельный узел — компьютер или порт маршрутизатора. Пакет должен быть доставлен узлу по кратчайшему маршруту;
- Cluster — адрес кластера. Обозначает группу узлов, которые имеют общий адресный префикс (например, присоединенных к одной физической сети). Пакет должен быть маршрутизирован группе узлов по кратчайшему пути, а затем доставлен только одному из членов группы (например, ближайшему узлу);
- Multicast — адрес набора узлов, находящихся в том числе в различных физических сетях. Копии пакета должны быть доставлены каждому узлу набора с использованием аппаратных возможностей групповой или широковещательной доставки, если это возможно.

Как и в версии IPv4, адреса в версии IPv6 делятся на классы, в зависимости от значения нескольких старших битов адреса.

Большая часть классов зарезервирована для будущего применения. Наиболее интересным с практической точки зрения является класс, предназначенный для провайдеров услуг Интернета, названный Provider-Assigned Unicast.

Для обеспечения совместимости со схемой адресации версии IPv4 в версии IPv6 есть класс адресов, имеющих 0000 0000 в старших битах адреса. Младшие 4 байта

адреса этого класса должны содержать адрес IPv4. Маршрутизаторы, поддерживающие обе версии адресов, должны обеспечивать трансляцию при передаче пакета из сети с адресацией IPv4 в сеть, поддерживающую адресацию IPv6, и наоборот.

Сетевые пакеты

Как уже упоминалось, информация по сети передается определенными порциями — пакетами. Причем на каждом уровне пакет имеет свой размер и структуру. В результате в пакет нижнего уровня вкладывается пакет следующего уровня и т. д. Чем более высокого уровня пакет, тем меньше информации он может содержать в себе. Размеры пакетов ограничиваются как особенностями аппаратуры, так и требованиями протоколов.

Маршрутизация пакетов

Маршрутизация — механизм передачи пакетов между сетями. При маршрутизации пакетов решается задача, как за наименьшее время, по кратчайшему пути, с минимальной стоимостью доставить пакет. Как правило, в совокупности решить эту задачу невозможно. Поэтому протоколы маршрутизации пакетов должны обеспечивать различные правила и стратегии маршрутизации. К примеру, доставить пакет с максимальной скоростью или с минимальной стоимостью.

Протоколы маршрутизации

Протоколы маршрутизации подразделяются на протоколы внутреннего (Interior Gateway Protocol, IGP) и внешнего шлюза (Exterior Gateway Protocol, EGP). Протокол внутреннего шлюза управляет маршрутизацией в пределах сети или группы сетей одного владельца, носящей название "автономная система". Внутри автономных систем имеется только список сетей, входящих в систему, и известны точки взаимодействия с внешним миром. Протокол внешнего шлюза отвечает за маршрутизацию между автономными системами.

На сегодняшний день широко применяются следующие протоколы маршрутизации:

- RIP (Routing Information Protocol) — протокол данных маршрутизации. Устаревший протокол. Тем не менее он достаточно широко распространен благодаря сервису routed, который является стандартной программой для операционных систем UNIX-семейства;
- OSPF (Open Shortest Path First) — протокол выбора кратчайшего пути. Протокол промышленного уровня. Он рассчитан на крупные сети со сложной топологией. Более гибок, чем протокол RIP, однако по сравнению с ним сложнее в администрировании; сервис gated;
- IGRP (Interior Gateway Routing Protocol) — протокол маршрутизации внутреннего шлюза. Используется маршрутизаторами CISCO;
- EGP (Exterior Gateway Protocol) — протокол внешнего шлюза. Старый протокол времен зарождения Интернета. Практически вытеснен протоколом BGP;
- BGP (Border Gateway Protocol) — протокол граничного шлюза. В отличие от EGP, поддерживает сложную топологию сети и имеет возможность настройки стратегии маршрутизации;

- DVMRP (Vector Multicast Routing Protocol) — протокол групповой маршрутизации по вектору расстояния;
- RIP, OSPF и IGRP — внутренние протоколы; EGP и BGP — внешние.

Адресация в TCP/IP

Каждый компьютер в сети IP имеет адреса трех уровней:

- *Локальный адрес узла*, определяемый технологией (например, Ethernet), с помощью которой построена отдельная сеть, в которую входит данный узел. Для узлов, входящих в локальные сети, — это MAC-адрес (Media Access Control) сетевого адаптера. MAC-адреса назначаются производителями оборудования и являются (теоретически) уникальными адресами, т. к. управляются централизованно, однако большинство производителей Ethernet-карт предоставляют утилиту для переназначения MAC-адреса. Для всех существующих технологий локальных сетей MAC-адрес имеет 6-байтовый формат: старшие 3 байта — идентификатор фирмы-производителя, а младшие 3 байта — уникальный код производителя;
- *IP-адрес*, состоящий из 4 байтов (стандарт IPv4) или 16 байтов (стандарт IPv6). Этот адрес используется на сетевом уровне. Его назначает администратор во время конфигурирования сети. IP-адрес состоит из двух частей: номера сети и номера узла. Номер сети администратор может выбрать произвольно либо назначить по рекомендации специального подразделения Интернета (Network Information Center, NIC), если сеть должна работать как составная часть Интернета;
- Символьный *идентификатор-имя*, например tosser.mail.ru. Этот адрес назначается администратором и состоит из нескольких частей, например, имени машины, имени фирмы, имени домена. Такой идентификатор-имя используется на прикладном уровне, например в протоколе FTP.

IP-адрес состоит из двух частей: сетевой части и адреса хоста. На основании сетевой части адреса принимается решение о сетевой маршрутизации. Адрес хоста однозначно определяет сетевое устройство, которое, в большинстве случаев, совпадает с хостом (как обычно, не обойтись без исключений — некоторые компьютеры имеют несколько IP-адресов). IP-адреса записывают в виде десятичных чисел (по одному на каждый байт адреса), разделенных точками, например 192.168.44.2. Однако не все сетевые адреса можно назначить компьютерам. Исключения — адреса 0.0.0.0, 127.0.0.1, 255.255.255.255 и некоторые другие. Существует несколько классов сетевых адресов (табл. 4.6). (Правда, на сегодняшний день это разделение устарело, однако его все равно стараются придерживаться).

Выделением IP-адресов занимается служба регистрации информационного центра InterNIC, но если необходимо получить 4–5 IP-адресов, то их вполне может предоставить любой интернет-провайдер. Однако не все адреса предназначены для доступа из Интернета. Существует группа адресов только для локальных сетей:

- 10.0.0.0—10.255.255.255
- 172.16.0.0—172.31.255.255
- 192.168.0.0—192.168.255.255

Таблица 4.6. *Распределение сетевых адресов по классам сетей*

Класс	Первый байт	Формат*	Комментарии
A	1–26	C.M.M.M	Очень крупные сети, как правило — корпорации или большие государственные учреждения
B	128–191	C.C.M.M	Крупные сети — крупные фирмы, большие интернет-провайдеры
C	192–223	C.C.C.M	Обычная сеть на 254 компьютера
D	224–239	—	Как правило, подсети, выдаваемые провайдерами клиентам
E	240–254	—	Экспериментальные адреса

* В колонке "Формат" буква C обозначает сетевую, а M — компьютерную часть адреса.

Как можно видеть, это адреса классов A, B и C соответственно. Тут нужно иметь в виду, что последний диапазон — это не одна сеть класса B, а 256 сетей класса C.

Несколько IP-адресов имеют специальное значение:

- адрес, в котором сетевая часть содержит нули, соответствует хосту в локальной сети. Например, 0.0.0.145 соответствует рабочей станции 145 в локальной сети, а адрес 0.0.0.0 — текущему хосту;
- сеть с адресом 127.X.X.X фиктивная, не имеющая никаких аппаратных сетевых интерфейсов и состоящая только из локального компьютера. Адрес 127.0.0.1 всегда обозначает текущую машину и имеет символическое имя localhost;
- адрес, содержащий в какой-либо части число 255, широковещательный. Например, пакет, посланный по адресу 192.168.3.255, будет отослан всем компьютерам в сети 192.168.3, а пакет, посланный по адресу 255.255.255.255, отправится по *всем* компьютерам Интернета (правда, на уровне маршрутизатора обычно такие широковещательные пакеты режутся).

Символьные имена Интернета имеют следующую структуру:

Имя_компьютера.домен3уровня.домен2уровня.домен1уровня

Пример: **www.rambler.ru**, **www.yahoo.com**.

Домены первого уровня стандартизированы и состоят из двух или трех букв латинского алфавита. В последнее время вводятся домены первого уровня, состоящие из более чем трех букв, например, .info, .shop, .name. Как правило, именем домена первого уровня может быть com, org, net, mil или двухсимвольное название страны, за которой закреплен домен: ru — Россия, ua — Украина, uk — Великобритания. Относительно имени домена второго уровня строгих правил нет. Для доменов первого уровня типа com домен второго уровня имеет имя компании или фирмы. Для домена страны правило именования несколько другие. В частности для России имя домена второго уровня определяется покупателем — к примеру, lib.ru, а для Украины имя домена второго уровня — это либо название областного центра (odessa.ua), либо имя типа com, org, net, mil, правда, можно уже получить и домен второго

уровня. Похожая ситуация наблюдается и в других странах: Швеция, Франция и Германия имена доменов второго уровня жестко не закрепляют, а Великобритания, Тайвань и Япония — закрепляют.

Протокол адресации ARP/RARP

Несмотря на то, что адресация IP-пакетов осуществляется при помощи IP-адресов, при передаче данных с компьютера на компьютер необходимы аппаратные MAC-адреса (конечно, кроме соединений типа "точка-точка"). Для определения соответствия аппаратных MAC-адресов IP-адресам служит протокол ARP (Address Resolution Protocol) — протокол преобразования адресов. Он применяется в сетях любых типов, использующих широковещательный режим. ARP можно применять только в пределах одной сети. Однако это не мешает передавать пакет через несколько сетей, просто при прохождении пакетом маршрутизатора он определяет новый MAC-адрес приемника. Каждый компьютер в сети создает кэш ARP, содержащий последние запросы.

Иногда аппаратные адреса необходимо транслировать в IP-адреса. Для этого предназначен протокол RARP (Reverse Address Resolution Protocol, обратный протокол преобразования адресов).

Протокол ICMP

Протокол ICMP — межсетевой протокол управления сообщениями (Internet Control Message Protocol) — отвечает за низкоуровневую поддержку протокола IP, включая подтверждение получения сообщения, сообщения об ошибках и многое другое. ICMP-сообщения посылаются с помощью стандартного IP заголовка. Первый октет в поле данных датаграммы — это поле типа ICMP-сообщения. Значение этого поля определяет формат всех остальных данных в датаграмме. Затем идет октет, содержащий код, поясняющий сообщение, и двухбайтовая контрольная сумма.

В табл. 4.7 приведены типы и коды ICMP-сообщений.

Таблица 4.7. *Типы и коды ICMP-сообщений*

Тип	Код	Описание кода	Описание типа
0	0		Ответ на эхо-сообщение
3	0	Невозможно передать датаграмму на локальную сеть, где находится адресат	Сообщение о проблемах при передаче пакетов. Шлюз может послать сообщения с кодами 0, 1, 4 и 5. Хост может послать сообщения с кодами 2 и 3
	1	Невозможно передать датаграмму на хост, являющийся адресатом	
	2	Нельзя воспользоваться указанным протоколом	
	3	Нельзя передать данные на указанный порт	

Таблица 4.7 *(продолжение)*

Тип	Код	Описание кода	Описание типа
3	4	Для передачи датаграммы по сети требуется фрагментация, но выставлен флаг DF (запрет фрагментации пакетов)	Сообщение о проблемах при передаче пакетов. Шлюз может послать сообщения с кодами 0, 1, 4 и 5. Хост может послать сообщения с кодами 2 и 3
	5	Сбой в маршрутизации при отправлении	
4	0		Сообщение для приостановки отправителя. Шлюз может удалить датаграммы, если у него нет места в буфере для постановки этих датаграмм в очередь на отправление по маршруту следования к адресату. Если шлюз удаляет датаграмму, то он должен послать сообщение для приостановки хосту, отправившему данную датаграмму. Сообщение о приостановке может послать также сам адресат, если датаграммы приходят слишком быстро, чтобы успеть их обработать. Сообщение о приостановке является запросом для хоста уменьшить скорость посылки данных на этот адрес
4	0		Шлюз или хост может посылать сообщение о приостановке еще до достижения предельной пропускной способности и не ждать, пока этот предел будет пройден. И шлюз, и хост могут отправить сообщение с кодом 0
5	0	Переадресация датаграмм для сети	Сообщение о переадресации
	1	Переадресация датаграмм для хоста	
	2	Переадресация датаграмм для типа услуг и сети	
	3	Переадресация датаграмм для типа услуг и хоста	
8	0		Эхо-сообщение и сообщение в ответ на эхо. Данные из эхо-сообщения должны быть переданы в ответ на это сообщение. Идентификатор и номер очереди может использоваться отправителем эхо-сообщения для идентификации приходящих пакетов. Компьютер, отозвавшийся на это сообщение, возвращает в своем ответе те же значения для идентификатора и номера очереди, что были в исходном эхо-сообщении. Как шлюз, так и хост могут возвращать сообщение с кодом 0

Таблица 4.7 *(окончание)*

Тип	Код	Описание кода	Описание типа
11	0	При передаче превышено время жизни	Сообщение о превышении контрольного времени. Шлюз может послать сообщение с кодом 0, а хост — с кодом 1
	1	Превышено контрольное время при сборке фрагментов датаграммы	
12	0	Указатель показывает ошибку	Сообщение о проблемах с параметром. Если шлюз или хост, обрабатывающий датаграмму, обнаруживает проблему с обработкой параметров заголовка, и это не позволяет завершить ее обработку, то он должен удалить датаграмму. Одной из причин этого могут быть неправильные аргументы в опции. Указатель определяет октет в заголовке исходной датаграммы, где была обнаружена ошибка. Код 0 сообщения может приходить как от шлюза, так и от хоста
13	0		Сообщение со штампом времени. Данные из сообщения возвращаются вместе с ответом, при этом в них добавляется еще один штамп времени. *Штамп времени* — это 32 бита, где записано время в миллисекундах, прошедшее после полуночи по единому времени (UT). *Штамп времени отправления* — это время, которое отправитель фиксировал последний раз перед посылкой сообщения. *Штамп времени получения* — это время, когда исходное сообщение впервые увидел получатель первоначального сообщения. *Штамп времени передачи* — это время, которое фиксировал в последний раз компьютер, отправляющий ответное сообщение. И шлюз, и хост могут возвращать сообщения с кодом 0
14	0		Сообщение с ответом на штамп времени
15	0		Запрос информации. Данное сообщение может быть послано, когда в IP-заголовке в полях отправителя и получателя записаны нули. В ответ должен быть послан IP-модуль с полностью заданными адресами. Данное сообщение является способом, с помощью которого хост может определить номер сети, куда он подключен. И хост, и шлюз могут возвращать сообщения с кодом 0
16	0		Ответное сообщение с информацией

Протоколы транспортного уровня

Протоколы транспортного уровня TCP и UDP базируются на протоколе IP и обеспечивают передачу данных с заданными характеристиками между источником и приемником. Эти протоколы вводят новый уровень адресации, называемый номером порта (port number), который определяет, какому процессу на хосте передаются данные. Номера портов занимают 2 байта. Существует список соответствия номеров портов приложениям, определенный в RFC1700 (Request For Comments, запрос для пояснений, описывающий стандарты протоколов Интернета и их взаимодействие). Некоторые зарезервированные порты приведены в табл. 4.8.

Таблица 4.8. *Сервисы и закрепленные за ними порты*

№ порта	Сервис	Описание
7	Echo	Эхо
20	FTP-data	Передача данных
21	FTP	Управляющие команды
23	Telnet	Удаленный доступ в систему
25	SMTP	Протокол электронной почты
53	Domain	Сервер доменных имен DNS
80	HTTP	Сервер WWW
110	POP3	Протокол электронной почты
119	NNTP	Телеконференции
123	NTP	Синхронизация времени
161	SNMP	Протокол управления сетевыми устройствами
179	BGP	Маршрутизация

Протокол TCP

Протокол TCP поддерживает надежную передачу данных с предварительной установкой связи между источником и приемником информации. На его базе реализована бóльшая часть протоколов уровня приложений.

Характеристики протокола TCP:

- перед началом передачи данных протокол создает канал между источником и приемником информации путем передачи запроса на начало сеанса и получения ответа. По окончании передачи данных сеанс должен быть явно завершен передачей соответствующего запроса;
- доставка данных является надежной. Перед отправкой следующего пакета источник информации должен получить подтверждение о приеме предыдущего пакета от приемника информации;

Примечание

Для ускорения передачи отправитель обычно держит "окно" определенного размера с отправленными, но неподтвержденными пакетами, которое сдвигается по мере их подтверждения. Таким образом, внутри "окна" может оказаться несколько неподтвержденных пакетов.

- возможность управления потоком данных;
- возможность доставки экстренных данных.

Эти возможности позволяют программам, использующим протокол TCP, не заботиться об организации надежной передачи данных. С другой стороны, этот протокол ограничивает скорость передачи данных.

Протокол UDP

Протокол UDP обеспечивает логический канал между источником и приемником данных без предварительного установления связи. Иными словами, пакеты, передаваемые по протоколу UDP, не зависят друг от друга, и никакого подтверждения доставки пакета протоколом не предусматривается. Это сильно напоминает бросание бутылки с запиской в море — авось дойдет. Поэтому программы, использующие этот протокол, должны сами проверять факт доставки информации. Однако благодаря своей простоте протокол UDP может при нормальных условиях передать гораздо больше информации, чем протокол TCP.

В качестве примера приведем несколько приложений, использующих протокол UDP:

- сервер DNS;
- программы, задействующие протокол синхронизации времени NTP;
- программы, применяющие протокол удаленной загрузки BOOTP.

Для всех перечисленных программ предполагается, что при утере пакета необходимые действия (повторная посылка пакета, выдача сообщения и т. п.) осуществляют сами программы. Гарантированную доставку данных реализует протокол TCP.

Протоколы уровня приложений

Последний, четвертый уровень — уровень приложений. К сожалению, почти каждый разработчик программ, имеющий дело с протоколом уровня приложения, изобретает свой протокол или модифицирует уже существующие. Однако существует некий "костяк" протоколов, описанный в соответствующих RFC. В зависимости от используемого протокола транспортного уровня протоколы уровня приложений либо полагаются на надежную доставку данных (протокол TCP), либо обеспечивают свой способ контроля достоверности данных (протокол UDP). В большинстве протоколов уровня приложений командами служат обычные английские слова (например, в протоколах SMTP, HTTP), что значительно упрощает отладку приложений.

Протокол FTP

Протокол передачи файлов. Предназначен для организации и приема файлов. Позволяет просматривать каталоги и файлы, переименовывать их, удалять и т. п. При пересылке файлов контролирует их целостность. Существует "младший брат" протокола FTP — TFTP, который намного проще в реализации и, в основном, применяется для загрузки информации на бездисковые рабочие станции.

Протокол SMTP

Простой протокол передачи почтовых сообщений. Позволяет работать с электронной почтой. Благодаря тому, что все команды — обычные английские слова, можно с помощью программы telnet подключиться на 25-й порт (SMTP) и передавать соответствующие команды с консоли.

Протокол Telnet

Протокол предназначен для удаленного доступа в систему. К примеру, можно с домашнего компьютера через Интернет зайти на рабочий компьютер и выполнять на нем любые команды (запускать программы, редактировать файлы и т. п.). Применяется в основном для удаленного администрирования системы. Считается небезопасным для операционной системы, т. к. при входе в систему логин (имя пользователя) и пароль передаются в открытом виде. Повсеместно заменен на протокол SSH.

Сетевая файловая система NFS

Протокол, разработанный фирмой Sun, для использования дисков и каталогов сервера рабочими станциями в качестве "псевдодисков". Возник очень давно, когда винчестер в 100 Мбайт стоил весьма дорого и один диск, распределяемый через сеть, давал существенную экономию денежных средств. Сегодня протокол NFS постепенно уходит, одно из немногих мест его применения — бездисковые рабочие станции с NFS в качестве "своей" файловой системы.

Протокол IPX

IPX (Internet Packet Exchange) — протокол обмена пакетами между сетями, разработанный фирмой Novell для своего программного продукта NetWare. Однако, начиная с четвертой версии своей операционной системы, фирма Novell стала внедрять поддержку протокола TCP/IP, а в пятой версии протокол TCP/IP стал практически "родным" для NetWare. Тем не менее протокол IPX еще кое-где встречается.
IPX произошел от протокола межсетевых датаграмм IDP (Internet Datagram Protocol), разработанного в научно-исследовательском центре Xerox. Протокол IPX реализует механизм сокетов с негарантированной доставкой датаграмм. Поверх IPX могут функционировать многие протоколы, в том числе:

- протокол данных маршрутизации RIP;
- протокол обмена нумерованными пакетами SPX (Sequenced Packet Exchange), гарантированная доставка;

- протокол Echo;
- протокол сообщений об ошибках;
- протокол обмена пакетами PEP (Packet Exchange Protocol);
- протокол сервисных объявлений SAP (Service Advertisement Protocol).

Существует программное обеспечение под Linux (Mars), выполняющее функции сервера NetWare, и ПО, выступающее клиентом для серверов NetWare. Также есть программы под Linux, позволяющие маршрутизировать пакеты IPX.

Протокол NetBIOS

Протокол для организации одноранговых сетей в продуктах фирмы Microsoft. В последнее время продукты Microsoft по умолчанию используют протоколы TCP/IP.

Стандарты в Интернете

Стандарты Интернета описаны в документах, известных как RFC (Request For Comments). В табл. 4.9 приведены некоторые стандарты.

Таблица 4.9. *Список основных стандартов Интернета*

Номер стандарта	Комментарий
RFC768	Описание протокола UDP
RFC791	Описание протокола IP
RFC792	Описание протокола ICMP
RFC793	Описание протокола TCP
RFC821	Описание протокола SMTP
RFC826, RFC903	Описание протокола ARP/RARP
RFC827, RFC904, RFC911	Описание протокола маршрутизации EGP
RFC854	Описание протокола Telnet
RFC950	Описание процедуры выделения подсетей
RFC959	Описание протокола FTP
RFC1058	Описание протокола RIP
RFC1094	Описание протокола NFS
RFC1157	Описание протокола SNMP
RFC1178	Рекомендации по выбору сетевого имени компьютера
RFC1180	Введение в TCP/IP
RFC1208	Сетевые термины
RFC1219	Порядок присвоения номеров подсетей

Таблица 4.9 *(окончание)*

Номер стандарта	Комментарий
RFC1234	Спецификация по прохождению IPX-пакетов по сетям IP
RFC1245, RFC1246, RFC1247, RFC1583	Описание протокола маршрутизации OSPF
RFC1267	Описание протокола BGP
RFC1597	Распределение локальных IP-адресов
RFC1700	Зарезервированные номера портов

Ссылки

- **www.rfc-editor.org** — сайт, посвященный RFC.

Глава 5



Идеология файловой системы

Один из столпов операционной системы, определяющий ее работоспособность, — файловая система. От ее архитектуры, возможностей и надежности во многом зависит работоспособность операционной системы. Помимо надежной "родной" файловой системы с продуманной архитектурой крайне желательно иметь возможность работать с другими наиболее распространенными системами (например, FAT 16/ FAT 32). В этой главе мы подробно рассмотрим, что нам предлагает Linux.

История развития файловых систем Linux

Первоначально Linux разрабатывалась как расширение ОС Minix, и было вполне логично взять от предшественника все, что можно, поскольку такое решение позволяло довольно быстро пройти этап проектирования (ведь все уже и так разработано, нужно было только создать соответствующий программный код). На тот момент (начало 1990-х, компьютеры на базе 386-го процессора считались мощными, в порядке вещей был жесткий диск емкостью 120 Мбайт) файловая система Minix была достаточно эффективна. Однако ее архитектурные ограничения (адреса блоков 16-битовые, что ставит предел максимального объема файловой системы в 64 Мбайт, каталоги содержат записи с ограниченным размером, имя файла не должно превышать 14 символов) очень скоро вынудили разработчиков задуматься об альтернативной файловой системе. Была разработана "Extended File System" (Ext FS — расширенная файловая система), затем ее сменила в качестве стандартной "Second Extended File System" (Ext2FS — вторая расширенная файловая система). Сегодня в качестве основной файловой системы практически во всех дистрибутивах используется Ext3 — с поддержкой журналирования, совместимая с Ext2, при необходимости может монтироваться как Ext2. В настоящее время осуществляется переход на Ext4. Существуют также и другие журналируемые файловые системы: ReiserFS и JFS от фирмы IBM, XFS от SGI.

Файл

Концепция файла — ключевая в операционной системе Linux. Практически все моменты, связанные с данными, в том или ином виде представляются в виде файла или операций с файлами. Для Linux по большому счету все равно, с каким устройством или процессом взаимодействовать, — система работает с *файлом*. В результате получается весьма унифицированный интерфейс.

Типы файлов

Поскольку понятие файла применяется к достаточно разнородным вещам (файл как таковой, физические устройства, каталоги и т. п.), поневоле возникает разделение файлов на типы. В Linux существует шесть типов файлов:

- файл;
- каталог;
- файл устройства;
- канал (FIFO, PIPE);
- символьная или мягкая ссылка (soft link);
- сокет (Socket).

Файл

Содержит информацию в некотором формате. Для операционной системы это просто набор байтов. Вся интерпретация содержимого файла осуществляется прикладной программой.

Каталог

Каталоги являются элементами иерархического дерева. Любой каталог может содержать файлы и подкаталоги. По существу, каталог — это файл, содержащий список записей, состоящих из номера индексного дескриптора и имени файла. Структуру записи *см. в разд. "Физическая структура Ext2"*.

Файл устройства

В операционной системе Linux доступ к устройствам осуществляется через специальные файлы. Такой файл является точкой доступа к драйверу устройства. Существует два типа файлов устройств: *символьные* и *блочные*.

Символьный файл устройства служит для небуферизированного обмена данными с устройством (байт за байтом), *блочный* — для обмена с устройством блоками данных. Некоторые устройства имеют как символьный, так и блочный интерфейс.

Канал

Файлы этого типа предназначены для связи между процессами для передачи данных.

Ссылки

Индексный дескриптор может быть связан с несколькими именами файлов. Дескриптор содержит поле, хранящее число, с которым ассоциируется файл. Добавление ссылки заключается в создании записи каталога, где номер индексного дескриптора указывает на другой дескриптор, и увеличении счетчика ссылок в дескрипторе. При удалении ссылки ядро уменьшает счетчик ссылок и удаляет дескриптор, если этот счетчик станет равным нулю. Такие ссылки называются *жесткими* и могут использоваться только внутри одной файловой системы (например, внутри одного раздела жесткого диска).

Описанные жесткие ссылки не являются особым типом файла, хотя бы потому, что они не отличаются от "оригинальных" файлов даже теоретически: команда `ls -l` никогда не выведет, что файл — жесткая ссылка.

Еще один тип ссылок — *символическая* ссылка — содержит только имя файла. Символические ссылки — специальный объект файловой системы, сильно отличающийся от остальных типов файлов. Так как символическая ссылка не указывает на индексный дескриптор, то возможно создание ссылок на файлы, расположенные в другой файловой системе. Эти ссылки могут указывать на файл любого типа, даже на несуществующий.

Сокет

Сокеты обеспечивают взаимодействие между процессами при доступе к сети TCP/IP.

Владельцы файлов

Файлы в Linux имеют одного владельца, а доступ к файлу определяется отдельно для трех групп пользователей: собственно владельца, группы владельца и прочих пользователей. Существует только один владелец, в группе может быть любое количество членов, все остальные в группу не входят. Привилегия владения — одно из ключевых понятий в системе защиты операционной системы Linux.

Каждый пользователь может (или не может) иметь право на чтение (запись и/или исполнение) файла, владельцем которого он является. На основе указанных трех групп пользователей можно построить политику прав доступа к файлам и каталогам, позволяющую достаточно надежно и непротиворечиво обезопасить операционную систему.

Обычно права доступа к файлу изменяются от максимальных у владельца файла до минимальных (вплоть до полного отсутствия) у всех остальных. Устанавливать и изменять права доступа к файлу или каталогу могут только два пользователя владелец файла и администратор системы (пользователь root). Изменить права доступа к файлу можно утилитой chmod.

Права доступа к файлам

Права доступа к файлу или к каталогу описываются тремя (вообще-то больше, но эти три основные и самые важные) восьмеричными цифрами: самая левая из этой тройки — права владельца, средняя — права группы, правая — права всех остальных. Каждая восьмеричная цифра представляет собой трехбитовую маску. Эти биты отвечают за права на (слева направо) чтение, запись и исполнение файла или каталога. Если установлена единица — доступ разрешен, если ноль — запрещен. Таким образом, права доступа к файлу, описанные цифрой 644, означают, что владелец может писать и читать файл, группа и остальные пользователи — только читать.

Посмотрим, что означают чтение, запись и выполнение файла с точки зрения функциональных возможностей.

- Чтение:
  - просмотр содержимого файла;
  - чтение каталога.
- Запись:
  - добавление или изменение файла;
  - удаление или перемещение файлов в каталоге.
- Выполнение:
  - запуск программы;
  - возможность поиска в каталоге в комбинации с правом чтения.

Узнать о том, какие права доступа установлены к файлам и каталогам, можно, используя команду `ls`. Результат выполнения команды `ls -l` приведен в листинге 5.1.

Листинг 5.1

```
lrwxrwxrwx    1 root     root      4 Авг 31  10:15 [ -> test
-rwxr-xr-x    1 root     root     93 Янв 22  2010  4odb_clean
-rwxr-xr-x    1 root     root     93 Янв 22  2010  4odb_clear
-rwxr-xr-x    1 root     root     95 Янв 22  2010  4odb_create
-rwxr-xr-x    1 root     root     97 Янв 22  2010  4odb_destroy
-rwxr-xr-x    1 root     root     89 Янв 22  2010  4odb_dig
-rwxr-xr-x    1 root     root     93 Янв 22  2010  4odb_grant
-rwxr-xr-x    1 root     root      97 Янв 22  2010 4odb_metadig
-rwxr-xr-x    1 root     root      99 Янв 22  2010 4odb_odmsdump
drwxr-xr-x    1 root     root      99 Янв 22  2010 t
```

В первой колонке представлены права доступа к файлу, во второй — число жестких ссылок, в третьей — имя владельца файла, в четвертой — название группы владельца файла, в пятой — дата создания и в шестой — имя файла или каталога. Первая строка листинга содержит ссылку на `test` (буква `l` в правах доступа обозначает, что это не файл, а ссылка). В последней строке листинга указан каталог `t` (буква `d` в правах доступа обозначает, что это каталог (directory), а не файл). Остальные строки листинга — файлы. В правах доступа вы видите десять символов. Первый слева — тип файла (файл, ссылка, каталог и т. п.). Следующие три символа — права доступа владельца файла: `rwx` (чтение, запись, исполняемость файла). Далее идут символы, задающие права доступа членов группы и всех остальных.

Модификаторы прав доступа

Как у любого правила, в жесткой системе прав доступа существуют свои исключения. Это так называемые дополнительные атрибуты файла:

- Sticky bit (Save Text Attribute) — "липкий" бит;
- SUID (Set User ID) — установка идентификатора пользователя;
- SGID — установка идентификатора группы.

Рассмотрим эти атрибуты подробнее.

- Sticky bit для файлов — в современных операционных системах потерял свое значение.
- Sticky bit для каталогов — если установлен для каталога, то пользователь, несмотря на то, что ему разрешена запись в этот каталог, может удалять только те файлы, владельцем которых он является или к которым ему явно заданы права записи.
- SUID для файлов — если установлены права доступа SUID и файл исполняемый, то файл при запуске на выполнение получает не права пользователя, запустившего его, а права владельца файла. Такие "фокусы" необходимы для того, чтобы пользователь мог работать с некоторыми системными файлами, владельцем которых является некий привилегированный пользователь. К примеру, для того, чтобы пользователь мог самостоятельно изменить свой пароль при помощи утилиты passwd, у этой утилиты (владелец которой — пользователь root) должен быть установлен бит SUID, поскольку она работает с файлами (/etc/passwd), модифицировать которые имеет право только пользователь root.
- SGID для файлов — аналогично установке бита SUID, только вместо владельца файла используется группа владельца.
- SGID для каталогов — файлы, создающиеся в этом каталоге, будут иметь установки группы такие же, как у каталога.

Узнать о том, какие дополнительные права доступа установлены к файлам и каталогам, позволяет команда `ls`. Вот пример выполнения команды `ls -l`:

```
-r-s--x--x    1 root     root        13536 Июл 12  2010 passwd
```

Как видно из прав доступа, у этого файла установлен SUID-бит (буква `s` в списке прав доступа).

Файловые системы

Файловая система — это методы и структуры данных, которые используются операционной системой для хранения файлов на диске или в его разделе.

Перед размещением файловой системы в разделе или на диске она должна быть инициализирована, а требуемые служебные данные перенесены на этот раздел или диск. Этот процесс называется созданием файловой системы (иногда его еще называют форматированием, что в принципе неверно).

Основные понятия в файловой структуре Linux (и в большинстве ОС UNIX-семейства):

- суперблок;
- индексный дескриптор (inode);
- блок данных;
- блок каталога (прямой блок);
- косвенный блок;
- файл.

Типы файловых систем

Linux поддерживает много типов файловых систем. Перечислим наиболее важные из них.

- Minix — старейшая файловая система, ограниченная в своих возможностях (у файлов отсутствуют некоторые временные параметры, длина имени файла не превышает 30 символов) и доступных объемах (максимум 64 Мбайт на одну файловую систему).
- Xia — модифицированная версия системы Minix, в которой увеличена максимальная длина имени файла и размер файловой системы.
- Ext — предыдущая версия системы Ext2. В настоящее время практически не встречается.
- Ext2 — наиболее богатая функциональными возможностями файловая система Linux. До последнего времени была самой популярной системой.
- Ext3 — модернизация системы Ext2. Помимо некоторых функциональных расширений является журналируемой. Получила широкое распространение.
- Ext4 — модернизация файловой системы Ext3. Основная особенность — увеличение максимального объема одного раздела диска до 1 эксабайта (2^{60} байт). Кроме того, в Ext4 представлен механизм пространственной записи файлов (новая информация добавляется в конец заранее выделенной по соседству области файла), уменьшающий фрагментацию и повышающий производительность.
- VFS — виртуальная файловая система. По сути — эмулятор-прослойка между реальной файловой системой (MS-DOS, Ext2, xia и т. д.) и ядром операционной системы Linux.
- Proc — псевдофайловая система, в которой посредством обычных файловых операций предоставляется доступ к некоторым параметрам и функциям ядра операционной системы.
- Sysfs — аналогична предыдущей по назначению и использованию.
- ReiserFS — наиболее популярная журналируемая файловая система для Linux.
- Devfs — псевдофайловая система, в которой посредством обычных файловых операций предоставляется доступ к устройствам компьютера. Позволяет очень гибко работать и конфигурировать устройства и взаимодействие с ОС.

В Linux для обеспечения обмена файлами с другими операционными системами включена поддержка некоторых сторонних файловых систем. Однако их функциональные возможности могут оказаться значительно ограниченными по сравнению с возможностями, обычно предоставляемыми файловыми системами UNIX.

- msdos — обеспечивает совместимость с системой MS-DOS.
- umsdos — расширяет возможности драйвера файловой системы MS-DOS для Linux так, что в Linux появляется возможность работы с именами файлов нестандартной длины, просмотра прав доступа к файлу, ссылок, имени пользователя, которому принадлежит файл, а также оперирования с файлами устройств. Это позволяет использовать (эмулировать) файловую систему Linux на файловой системе MS-DOS.
- iso9660 — стандартная файловая система для CD-ROM.
- xenix — файловая система Xenix.

- sysv — файловая система System V (версия для x86).
- hpfs — доступ "только для чтения" к разделам HPFS.
- nfs — сетевая файловая система, обеспечивающая разделение одной файловой системы между несколькими компьютерами для предоставления доступа к ее файлам со всех машин по сети.
- NTFS — обеспечивает доступ к разделам, созданным Windows Nt/2000/XP.

В табл. 5.1 приведена общая информация о функциональных возможностях, предоставляемых различными файловыми системами.

Таблица 5.1. *Сравнение файловых систем*

Критерий сравнения	Файловая система				
	Minix FS	**Xia FS**	**Ext FS**	**Ext2 FS**	**Ext4 FS**
Максимальный объем файловой системы	64 Мбайт	2 Гбайт	2 Гбайт	4 Тбайт	1 Эксабайт
Максимальная длина файла	64 Мбайт	64 Мбайт	2 Гбайт	2 Гбайт	16 Тбайт
Максимальная длина имени файла	30 символов	248 символов	255 символов	255 символов	256 символов
Поддержка трех ячеек времени изменения файла	Нет	Да	Нет	Да	Да
Возможность расширения	Нет	Нет	Нет	Да	Да
Изменяемый размер блока	Нет	Нет	Нет	Да	Да
Защита информации	Да	Да	Нет	Да	Да

Установка файловой системы

Файловая система устанавливается при помощи команды `mkfs`. Для каждого типа файловой системы существует своя версия этой программы. Команда `mkfs` запускает требуемую программу в зависимости от типа файловой системы.

Параметры командной строки, передаваемые `mkfs`, слегка различаются для разных типов файловых систем. Полное описание параметров командной строки `mkfs` можно найти в соответствующем разделе man (справочной системы программы). С помощью параметров командной строки можно задать тип создаваемой файловой системы, произвести верификацию диска и маркировку сбойных блоков или получить список сбойных блоков из текстового файла.

Монтирование и демонтирование файловой системы

Для нормальной работы операционной системы ядро должно получить параметры файловых систем, используемых во время работы, и определенным образом настроить специальные таблицы. Для этого существует, по крайней мере, два способа:

1. Каким-то образом один раз получить тип и параметры файловой системы и работать с ними все время.
2. Получать их каждый раз при обращении к файловой системе.

У обоих вариантов имеются свои плюсы и минусы. Плюсы первого варианта — уменьшаются затраты времени на определение файловой системы и инициализацию таблиц ядра ОС. Минусы — невозможно "на ходу" заменить одно устройство (носитель информации) на другое (к примеру, диск Zip100 на Zip250), поскольку в таблицах ядра зафиксированы емкость носителя, емкость кластеров, используемые блоки и тому подобная информация. Достоинства и недостатки второго варианта прямо противоположны первому — возможность "на ходу" заменить устройство (носитель информации), но большие затраты времени на определение файловой системы и инициализацию таблиц ядра ОС. К тому же, во втором варианте намного труднее достичь надежности хранения данных.

Поэтому большинство операционных систем (не только UNIX) в явной или неявной форме реализуют первый вариант взаимодействия с файловой системой. Для этого в Linux предусмотрены операции *монтирования* и *демонтирования* файловой системы.

Поскольку в Linux реализовано единое связанное дерево каталогов, то, в отличие от DOS/Windows, не существует такого понятия файловой системы, как диск. Все дисковые устройства (файловые системы) интегрируются в дереве каталогов в так называемые точки монтирования, в качестве которых выступают обычные каталоги. Причем если до монтирования в этом каталоге содержались какие-то файлы, то они становятся недоступными до тех пор, пока вы не демонтируете эту файловую систему. Операции монтирования/демонтирования осуществляют две команды: `mount` и `umount`.

Команда `mount` принимает несколько параметров, причем обязательными являются всего два. Первый их них — файл устройства, соответствующий диску или разделу, на котором расположена файловая система, или его псевдоним (к примеру, CD-ROM, floppy). Второй параметр — имя каталога, к которому будет монтироваться система.

Пример:

```
mount /dev/hda1 /mnt.
```

Помимо обязательных параметров можно задавать тип монтируемой файловой системы (при его отсутствии команда пытается самостоятельно определить ее тип), режим доступа, кодировку имен файлов и некоторые другие параметры.

Существует специальный файл /etc/fstab, содержащий список файловых систем и их параметры монтирования. Этот файл требуется ядру операционной системы при ее старте. Ядро пытается смонтировать файловые системы, описанные в этом файле, с соответствующими параметрами монтирования.

После того как отпала необходимость в файловой системе, ее можно демонтировать. Чаще всего это бывает при работе с Flash-накопителями и дисками CD/DVD (один диск необходимо заменить на другой). Демонтирование выполняет команда `umount`. В качестве параметра указывают файл устройства или точку монтирования.

Примеры:

```
umount /dev/hda1
umount /mnt/floppy.
```

По окончании работы со сменным носителем информации его обязательно необходимо отмонтировать. Поскольку ядро Linux осуществляет "отложенную" запись на диск, то к тому моменту, когда вы извлечете из компьютера Flash-накопитель без отмонтирования, информация еще может быть не записана на диск из системного буфера.

Для выполнения операций монтирования и демонтирования требуется наличие прав доступа root. Но при работе на своем персональном компьютере это усложняет процедуру. Есть несколько вариантов решения такой проблемы:

- в KDE или GNOME обычному пользователю можно монтировать CD/DVD-ROM и Flash-устройства;
- осуществить временный вход в систему пользователем root, монтировать/демонтировать диск и немедленно выйти;
- применить программу sudo, разрешающую выполнение команды `mount` отдельным пользователям;
- задействовать пакет mtools, предназначенный для работы с файловой системой MS-DOS;
- поместить список файлов устройств, необходимых при работе со сменными носителями, и доступных узлов монтирования вместе с нужными опциями (разрешением монтирования пользователем) в файл /etc/fstab;
- назначить динамически устанавливаемые права на некоторые устройства для пользователей, работающих за консолью, с помощью системы PAM и модуля pam_console.

Поддержка работоспособности файловых систем

Даже лучшая файловая система не обладает стопроцентной надежностью. Рано или поздно целостность файловой системы нарушается. Это может произойти от некорректного завершения работы системы (нажатие кнопки Reset, перебои в электропитании) или повреждения носителя информации. Проверку и восстановление целостности файловой системы осуществляет команда `fsck`. Она при загрузке системы запускается автоматически, поэтому возможные неполадки будут обнаружены (и может быть исправлены) перед работой с файловой системой.

Полная проверка файловой системы на современных жестких дисках может занять достаточно большое время, поэтому существуют некоторые способы избежать таких проверок. В файловой системе Ext2 предусмотрен специальный флаг, расположенный в суперблоке, который служит для выявления корректности демонтирования файловой системы при последнем выключении компьютера. Можно принудительно отключить проверку файловой системы, создав файл /etc/fastboot.

Автоматическая проверка целесообразна только для файловых систем, монтируемых во время загрузки. Для проверки других систем команду `fsck` нужно выполнять вручную.

Если `fsck` находит неисправность, которую не может исправить, то для восстановления структуры файловой системы или потерянной информации могут потребоваться глубокие знания и понимание работы файловых систем и их типов.

Команда `fsck` должна использоваться только для демонтированных систем (за исключением корневой файловой системы, которая проверяется смонтированной в режиме read-only), т. к. при ее работе осуществляется прямой доступ к диску, и информация о внесении каких-либо изменений в файловую систему может быть недоступна операционной системе, что обычно приводит к нарушению ее работы.

Также рекомендуется утилита `badblocks`. При ее выполнении выводится список номеров поврежденных блоков, который можно передать программе `fsck` для внесения изменений в структуру файловой системы.

Виртуальная файловая система (VFS)

VFS — это база, на которой основывается все многообразие поддерживаемых файловых систем.

Принцип функционирования

Ядро системы Linux содержит в себе программный код-посредник, выполняющий функции виртуальной файловой системы. Этот код обрабатывает запросы к файлам и вызывает необходимые функции соответствующей файловой системы для выполнения операции ввода/вывода. Такой механизм работы с файлами позволяет упростить объединение и совместное использование нескольких типов файловых систем.

Пусть программа записывает информацию в файл (или считывает ее, не суть важно). Программа вызывает библиотечную функцию, отвечающую за запись (или чтение) информации в файл. Эта функция определенным образом подготавливает информацию, которая затем передается в ядро системы. Ядро, в свою очередь, вызывает соответствующую функцию виртуальной файловой системы. Эта функция определяет, с каким типом файловой системы будут производиться манипуляции, подготавливает данные и вызывает необходимую функцию соответствующей файловой системы, с которой производится операция. Такая многоуровневая процедура позволяет максимально абстрагироваться от особенностей операционной системы и при необходимости безболезненно эмулировать недостающие атрибуты файла.

Структура VFS

Виртуальная файловая система содержит набор функций, которые должна поддерживать любая файловая система (создание, удаление, модификация файла, каталога и тому подобные действия). Интерфейс состоит из функций, оперирующих тремя типами объектов: файловые системы, индексные дескрипторы и открытые файлы.

Виртуальная файловая система использует таблицу, в которой во время компиляции ядра сохраняется информация обо всех типах поддерживаемых файловых систем. Запись в таблице содержит тип файловой системы и указатель на соответствующую функцию монтирования файловой системы. При монтировании файловой системы эта функция возвращает виртуальной файловой системе дескриптор, используемый в дальнейшем при операциях ввода/вывода.

Дескриптор смонтированной файловой системы содержит определенный набор информации: указатели на функции, служащие для выполнения операций файловой системы, и данные, используемые этой системой. Указатели на функции, расположенные в дескрипторе файловой системы, позволяют виртуальной файловой системе получить доступ к функциям, специфичным для данной файловой системы.

В виртуальной файловой системе применяются еще два типа дескрипторов: индексный дескриптор и дескриптор открытого файла. Каждый из них содержит информацию, связанную с обрабатываемыми файлами и набором операций, необходимых файловой системе. Индексный дескриптор содержит указатели к функциям, применяемым к любому файлу, а дескриптор открытого файла содержит указатели к функциям, оперирующим только с открытыми файлами.

Файловая система Ext2

Файловая система Ext2 (The Second Extended File System, вторая расширенная файловая система) была разработана с целью устранения ошибок, обнаруженных в предыдущей системе Ext (Extended File System), и снятия некоторых ее ограничений.

Стандартные возможности Ext2

Файловая система Ext2 поддерживает стандартные типы файлов UNIX:

- файлы;
- каталоги;
- файлы устройств;
- символические ссылки.

Ext2 может управлять файловыми системами, установленными на очень больших дисковых разделах. Система поддерживает длинные имена файлов (до 255 символов). Ext2 резервирует некоторое количество блоков для пользователя root, что позволяет системному администратору избежать нехватки объема жесткого диска при его заполнении другими пользователями.

Дополнительные возможности Ext2

В файловой системе Ext2 допустима синхронная модификация данных, которая повышает плотность записи информации, но одновременно ухудшает производительность.

Ext2 позволяет при создании файловой системы выбрать размер логического блока: 1024, 2048 или 4096 байтов. Организация блоков большого объема приводит к ускорению операций чтения/записи, но при этом дисковое пространство используется нерационально.

Ext2 позволяет применять ускоренные символические ссылки. В этом случае блоки данных файловой системы не задействованы. Имя файла назначения хранится не в блоке данных, а в самом индексном дескрипторе. Такая структура позволяет сохранить дисковое пространство и ускорить обработку символических ссылок. Максимальная длина имени файла в ускоренной ссылке равна 60 символам.

В Ext2 для индикации состояния файловой системы выделено отдельное поле в суперблоке. Если файловая система смонтирована в режиме read/write, то ее

состояние устанавливается как Not Clean (не "чистая"). Если же она демонтирована или смонтирована заново в режиме read-only, то ее состояние устанавливается в Clean ("чистая"). При загрузке операционной системы и анализе состояния файловой системы эта информация указывает на необходимость тестирования. Ядро также помещает в это поле некоторые ошибки. При определении ядром какого-либо несоответствия файловая система помечается как Erroneous ("ошибочная").

Длительное отсутствие проверки может привести к проблемам функционирования файловой системы, поэтому Ext2 включает в себя два метода для организации принудительного тестирования. В суперблоке содержится счетчик монтирования системы, который увеличивается каждый раз, когда система монтируется в режиме read/write. Если его значение достигает максимально возможного (оно также хранится в суперблоке), то запускается программа проверки файловой системы (только при старте операционной системы), даже если ее состояние Clean. В суперблоке также хранится последнее время тестирования и максимальный интервал между проверками. При превышении этого интервала также запускается программа проверки файловой системы.

В системе Ext2 имеются утилиты для ее настройки. Так, программа tune2fs служит для определения порядка действий при обнаружении ошибки. Возможно одно из трех действий:

- продолжение выполнения;
- монтирование файловой системы заново в режиме read-only;
- перезагрузка системы для проверки файловой системы.

Кроме того, эта программа позволяет задать:

- максимальное значение числа монтирований файловой системы;
- максимальный интервал между проверками файловой системы;
- количество логических блоков, зарезервированных для пользователя root.

Физическая структура Ext2

Как и во многих файловых системах, в Ext2 существует загрузочная область. На первичном разделе (primary, в терминологии программы Fdisk фирмы Microsoft) она содержит загрузочную запись — фрагмент кода, который инициирует процесс загрузки операционной системы при запуске. Все остальное пространство раздела делится на блоки стандартного размера (1, 2 или 4 Кбайт). Блок — минимальная логическая единица дискового пространства (в других операционных системах такой блок называют кластером). Выделение места файлам осуществляется целыми блоками.

Блоки, в свою очередь, объединяются в группы. Каждая группа блоков имеет одинаковое строение. Рассмотрим подробнее их структуру (рис. 5.1).

Суперблок одинаков для всех групп, все остальные поля индивидуальны для каждой группы. Суперблок хранится в первом блоке каждой группы блоков, является начальной точкой файловой системы, имеет размер 1024 байта и располагается со смещением в 1024 байта от начала файловой системы. Копии суперблока используются при восстановлении файловой системы после сбоев и тоже хранятся в файловой системе.



Рис. 5.1. Структура группы блоков

Информация в суперблоке служит для доступа к остальным данным на диске. В суперблоке определяется размер файловой системы, максимальное число файлов в разделе, объем свободного пространства. При старте операционной системы суперблок считывается в память, и все изменения файловой системы сначала записываются в копию суперблока, находящуюся в оперативной памяти, и только затем сохраняются на диске. Значения при описании структуры суперблока:

- `SHORT` — короткое целое (1 байт);
- `USHORT` — беззнаковое короткое целое (1 байт);
- `LONG` — длинное целое (4 байта);
- `ULONG` — беззнаковое длинное целое (4 байта).

~~Структура суперблока приведена в *приложении 1 (табл. П1.1).*~~

После суперблока следует описание группы блоков (Group Descriptors). ~~Структура этого массива приведена в *приложении 1 (табл. П1.2).*~~

Битовая карта блоков (Block Bitmap) — это структура, каждый бит которой показывает, отведен ли соответствующий ему блок какому-либо файлу. Если бит равен единице, то блок занят. Эта карта служит для поиска свободных блоков в тех случаях, когда нужно выделить место под файл.

Битовая карта индексных дескрипторов (Inode Bitmap) выполняет аналогичную функцию по отношению к таблице индексных дескрипторов — показывает, какие дескрипторы заняты.

Индексные дескрипторы файлов

Индексные дескрипторы файлов содержат информацию о файлах группы блоков. Каждому файлу на диске соответствует один и только один индексный дескриптор файла, который однозначно идентифицируется своим порядковым номером — индексом файла. Отсюда следует, что число файлов, которые могут быть созданы в файловой системе, ограничено числом индексных дескрипторов.

Поле типа и прав доступа к файлу (`i_mode`) представляет собой слово, каждый бит которого служит флагом.

Некоторые индексные дескрипторы используются файловой системой в специальных целях.

Каталог, по сути, является специальным файлом, содержимое которого состоит из записей определенной структуры.

Система адресации данных

Система адресации данных позволяет находить нужный файл среди блоков на диске. В Ext2 система адресации реализуется полем `i_block` индексного дескриптора файла.

Поле `i_block` в индексном дескрипторе файла представляет собой массив из 15 адресов блоков. Первые 12 адресов в этом массиве (`EXT2_NDIR_BLOCKS [12]`) представляют собой прямые ссылки на номера блоков, в которых хранятся данные из файла. 13-й адрес является косвенной ссылкой (адресом блока), в которой хранится список адресов следующих блоков с данными из этого файла. 14-й адрес в поле `i_block` индексного дескриптора указывает на блок двойной косвенной адресации (double indirect block), содержащий список адресов блоков, которые, в свою очередь, содержат списки адресов следующих блоков данных того файла, который задается индексным дескриптором.

Последний адрес в поле `i_block` индексного дескриптора задает адрес блока тройной косвенной адресации, т. е. блока со списком адресов блоков, которые являются блоками двойной косвенной адресации.

Оптимизация производительности

Файловая система Ext2 при операциях ввода/вывода использует буферизацию данных. При считывании блока информации ядро выдает запрос операции ввода/вывода на несколько расположенных рядом блоков, что заметно ускоряет извлечение данных при последовательном считывании файлов.

При занесении данных в файл файловая система Ext2, записывая новый блок, заранее размещает рядом до 8 смежных блоков. Подобный метод позволяет размещать файлы в смежных блоках, что ускоряет их чтение и дает возможность достичь высокой производительности системы.

Средства управления файловой системы Ext2

Средства управления файловой системы служат для создания, модификации и коррекции любых искажений файловой структуры:

- mke2fs — применяется для установки дискового раздела, содержащего пустую файловую систему Ext2;
- tune2fs — позволяет настроить параметры файловой системы;
- e2fsck — предназначена для устранения несоответствий в файловой системе;
- ext2ed — применяется для правки файловой системы;
- debugfs — определяет и устанавливает состояния файловой системы.

Программа e2fsck спроектирована так, что выполняет проверку с максимально возможной скоростью. В первом проходе e2fsck просматривает все индексные дескрипторы файловой системы и проверяет их как отдельные элементы системы. Также проверяются карты битов, указывающие использование блоков и дескрипторов.

Если e2fsck находит блоки данных, номера которых содержатся более чем в одном дескрипторе, то запускаются проходы с 1B по 1D для устранения несоответствия либо путем увеличения разделяемых блоков, либо удалением одного или более дескрипторов.

Во втором проходе проверяются каталоги как отдельные элементы файловой системы. Блок каждого каталога тестируется отдельно, без ссылки на другие блоки каталогов. Для первого блока каталога в каждом дескрипторе каталога проверяется существование записей "." (ссылка на себя) и ".." (ссылка на родительский каталог), а также соответствие номера дескриптора для записи "." текущему каталогу.

В третьем проходе анализируются связи каталогов. Программа e2fsck тестирует пути каждого каталога по направлению к корневому. В этом же проходе проверяется запись ".." для каждого каталога. Все каталоги, не имеющие связи с корневым каталогом, помещаются в каталог /lost+found.

В четвертом проходе e2fsck проверяет счетчики ссылок для каждого индексного дескриптора. Все неудаленные файлы с нулевым счетчиком ссылок также помещаются в каталог /lost+found.

В пятом проходе e2fsck анализирует правильность всей информации о файловой системе. При этом сравниваются карты битов блоков и дескрипторов, записанных на носителе информации, со значениями, полученными во время проверки файловой системы, и при необходимости информация на диске корректируется.

Журналируемые файловые системы

Основная цель, которая преследуется при создании журналируемых файловых систем, состоит в том, чтобы обеспечить как можно бóльшую вероятность быстрого восстановления системы после сбоев (например, после потери питания). Дело в том, что если происходит сбой, то часть информации о расположении файлов теряется, поскольку система не успевает записать все изменения из буфера на диск. После сбоя утилита fsck должна проверить все диски, которые не были корректно демонтированы, чтобы восстановить потерянную информацию. При современных объемах жестких дисков, исчисляемых десятками гигабайтов, на проверку двух-трех таких дисков может уйти слишком много времени. Кроме того, нет гарантии, что все данные удастся восстановить.

В журналируемых файловых системах для решения этой проблемы применяют транзакции, которые хорошо известны всем программистам баз данных. Идея транзакции достаточно проста — существует набор связанных операций, называемых транзакцией, и эта группа операций является атомарной (неделимой). Таким образом, транзакция признается успешной (завершенной) в том случае, если все операции, составляющие транзакцию, завершились успешно. Но это еще не все. Система ведет журнал, в котором отражаются все действия с данными и каждое изменение данных протоколируется. При сбое на основании журнала можно вернуть систему в безошибочное состояние.

Основное отличие транзакций баз данных от транзакций, применяемых в журналируемых файловых системах, состоит в том, что в базах данных в журнале

сохраняются изменяемые данные и вся управляющая информация, а в файловых системах — только метаданные: индексные дескрипторы изменяемых файлов, битовые карты распределения свободных блоков и свободных индексных дескрипторов.

Файловая система Ext3

По большому счету, файловая система Ext3 не является новой файловой системой. Это похоже на ситуацию с файловой системой FAT 16/FAT 32 — они совместимы, но проблема решена экстенсивным путем. Было необходимо срочно создать журналируемую файловую систему. Если начинать с нуля, то это долго и накладно, тогда сделали для Ext2 несколько десятков специальных функций и назвали все это Ext3 — получился некий гибрид. Вроде бы добавились журналирующие функции, но не в том объеме, в каком хотелось. И узкие места Ext2 остались: отсутствие оптимизации дискового пространства, ограничение на размер файла и т. п.

Файловая система Ext4

Файловая система Ext4 — результат эксплуатации и переосмысления Ext3. В систему были внесены серьезные изменения, рассмотренные далее.

В файловой системе Ext3 адресация данных выполнялась поблочно. С ростом размеров файла такой способ адресации становится менее эффективным. Экстенты позволяют адресовать большое количество (до 128 Мбайт) последовательно идущих блоков одним дескриптором. До четырех указателей на экстенты может размещаться непосредственно в inode, что достаточно для файлов маленького и среднего размера.

48-битовые номера блоков. При размере блока 4K это позволяет адресовать до одного эксабайта пространства.

Выделение блоков группами (multiblock allocation). Файловая система хранит не только информацию о местоположении свободных блоков, но и количество свободных блоков, идущих друг за другом. При выделении места файловая система находит такой фрагмент, в который данные можно записать без фрагментации. Это сделано с целью уменьшения фрагментации файловой системы.

Отложенное выделение блоков (delayed allocation). Блоки для хранения данных файла выделяются непосредственно перед физической записью на диск. Поэтому операции выделения блоков можно делать группами, что в свою очередь минимизирует фрагментацию и ускоряет процесс выделения блоков. Однако это увеличивает риск потери данных при внезапном сбое питания.

Отсутствует лимит в 32000 каталогов. В Ext3 без использования специальных патчей в одном каталоге можно было создать не более 32 000 подкаталогов.

Резервирование inode при создании каталога (directory inodes reservation). При создании каталога резервируется несколько inode. Впоследствии, при создании

файлов в этом каталоге сначала используются зарезервированные inode, и если таких не осталось, выполняется обычная процедура.

Размер inode. Размер inode (по умолчанию) увеличен с 128 до 256 байтов.

Временные метки с наносекундной точностью (nanosecond timestamps). Более высокая точность и расширенный диапазон времен, хранящихся в inode: верхней границей хранимого времени стало 25 апреля 2514 года.

Версия inode. В inode появился номер, который увеличивается при каждом изменении inode файла. Это позволяет, например, в NFSv4 узнавать, изменился ли файл.

Хранение расширенных атрибутов в inode (EA in inode). Хранение расширенных атрибутов, таких как ACL, атрибутов SELinux и прочих, позволяет повысить производительность. Атрибуты, для которых недостаточно места в inode, хранятся в отдельном блоке размером 4 килобайта.

Контрольное суммирование в журнале (Journal checksumming). Контрольные суммы журнальных транзакций позволяют обнаруживать и исправлять ошибки при проверке целостности системы после сбоя.

Предварительное выделение (persistent preallocation). В Ext4 появилась возможность зарезервировать множество блоков для записи и не тратить на инициализацию лишнее время. Если приложение попробует прочитать несуществующие данные, оно получит сообщение о том, что они не проинициализированы. Таким образом, несанкционированно прочитать удаленные данные не получится.

Дефрагментация без размонтирования (online Defragmentation). Реализовано в самой последней версии e2fsprogs.

Неинициализированные блоки (uninitialised groups). Пока не реализовано. Позволяет ускорить проверку файловой системы с помощью `fsck`. Блоки, отмеченные как неиспользуемые, проверяются группами, и детальное тестирование проводится, только если проверка группы показала, что внутри есть повреждения. Предполагается, что эта возможность позволит в 2–10 раз ускорить процесс тестирования целостности файловой системы.

Как видите, внесены значительные изменения, призванные увеличить производительность и устранить узкие места предыдущих версий файловой системы.

Файловая система ReiserFS

Кроме проблемы быстрого восстановления после сбоев, в файловой системе Ext2 имеется еще несколько нерешенных задач: нерациональное использование дискового пространства, ограничение на размер файла и неоптимальный поиск.

Поскольку файловая система содержит простой связный список, то время поиска информации линейно зависит от длины списка. Таким образом, чем длиннее список (к примеру, файлов в каталоге), тем дольше идет поиск необходимого элемента.

В системе ReiserFS применяются так называемые "сбалансированные деревья" или "B+Trees", время поиска в которых пропорционально не числу объектов, а логарифму этого числа. В сбалансированном дереве все ветви имеют одинаковую

длину. ReiserFS использует сбалансированные деревья для хранения всех объектов файловой системы: файлов в каталогах, данных о свободных блоках и т. д. Это позволяет существенно повысить производительность обращения к дискам.

Кроме того, система ReiserFS журналируемая, т. е. в ней решена задача быстрого восстановления после сбоев. Ограничения на размер файла в ReiserFS тоже сняты.

Ссылки

- **e2fsprogs.sourceforge.net** — утилиты и документация файловой системы Ext2.
- **www.nongnu.org/ext2-doc/** — документация файловой системы Ext2.
- **ftp.uk.linux.org/pub/linux/sct/fs/jfs/** — код и документация Ext3.
- **xgu.ru/wiki/Ext4** — обзор файловой системы Ext4.
- **www.atnf.csiro.au/~rgooch/linux/docs/vfs.txt** — обзор виртуальной файловой системы.
- **www.osp.ru/pcworld/2000/02/064.htm** — Виктор Хименко. Файлы, файлы, файлы. Обзор файловых систем.
- **www.opennet.ru/docs/RUS/fs/** — руководство по файловым системам ReiserFS, tmpfs, devfs, Ext3 и XFS (перевод).
- **http://xgu.ru/wiki/Ext4** — руководство по файловой системе Ext4.

Глава 6



Дерево каталогов Linux

Эта глава полностью посвящена структуре и размещению каталогов и файлов в Linux. Поскольку структура различных дистрибутивов может слегка отличаться, для определенности будем рассматривать дистрибутив Red Hat.

Для того чтобы ориентироваться в Linux, необходимо хорошо представлять себе структуру и размещение каталогов и файлов. Эти параметры для UNIX и Linux описаны в документе "Filesystem Hierarchy Standard — Version 2.3 final", Filesystem Hierarchy Standard Group, edited by Rusty Russell, Daniel Quinlan and Christopher Yeoh, редакция от January 28, 2004. Дальнейший текст в основном базируется на этом документе.

Все файлы можно классифицировать по двум признакам — доступность (shareable, разделяемость) на сетевом уровне и изменяемость/неизменность содержимого.

Соответственно, для каждого признака можно ввести свои понятия:

- *разделяемые* данные — те, которые могут использовать несколько хостов одновременно, т. е. данные, доступные для других хостов через сеть;
- *неразделяемые* данные — как правило, специфичные для каждого хоста, недоступные через сеть для других хостов;
- *статические* данные — включают системные файлы, библиотеки, документацию и другое, что не изменяется без вмешательства администратора;
- *динамические (переменные)* данные — все то, что может изменять пользователь.

Эти признаки взаимно ортогональны, в табл. 6.1 приведены некоторые каталоги, соответствующие этим признакам.

Таблица 6.1. *Признаки данных и каталоги*

	Разделяемые данные	**Неразделяемые данные**
Статические данные	/usr/opt	/etc/boot
Динамические данные	/var/mail /var/spool/news	/var/lock/var/run

Как видно из табл. 6.1, каталог /usr — статический разделяемый, а /var/lock — динамический неразделяемый. По этим признакам можно распределить все каталоги в файловой системе, о чем и будет упоминаться в соответствующих разделах. Однако такое четкое распределение не всегда наблюдается в современных UNIX-системах.

Как правило, эта проблема возникает из-за поддержки совместимости со старым программным обеспечением. Каталоги, не удовлетворяющие четкому разделению, будут упомянуты особо.

Иерархия каталогов Linux

В табл. 6.2 приведена иерархия каталогов первого уровня.

Таблица 6.2. *Каталоги первого уровня операционной системы Linux*

Имя	Содержимое
/	Корневой (Root) каталог. Является родительским для всех остальных каталогов в системе
/bin	Содержит важные для функционирования системы файлы
/boot	Содержит файлы для загрузчика ядра
/dev	Хранит файлы устройств
/etc	Содержит Host-специфичные файлы системной конфигурации
/home	Пользовательские домашние каталоги
/lib	Важные разделяемые библиотеки и модули ядра
/lost+found	Содержит файлы, восстановленные при ремонте утилитами восстановления файловых систем
/media	Каталог для автоматически монтируемых устройств (дисковод, CD-ROM)
/mnt	Точка монтирования временных разделов
/opt	Дополнительные пакеты приложений
/proc	Точка монтирования псевдофайловой системы proc, которая является интерфейсом ядра операционной системы
/root	Домашний каталог для пользователя root
/sbin	Содержит важные системные исполняемые файлы
/srv	Данные, специфичные для окружения системы
/sys	Точка монтирования файловой системы sysfs, частично заменяющей /proc
/tmp	Хранит временные файлы
/usr	Вторичная иерархия
/var	Содержит переменные данные

Рассмотрим подробнее иерархию каталогов.

Корневой каталог (Root)

Точка монтирования всей файловой системы. Играет исключительно важную роль в процессе "жизнедеятельности" операционной системы. Для загрузки системы необходимо, чтобы в корневом разделе (корневой раздел в Linux — это аналог

диска C: для DOS/Windows — только на него можно установить операционную систему, корневой раздел является точкой монтирования корневого каталога) находились утилиты и конфигурационные файлы, необходимые для монтирования других файловых систем. Кроме того, в корневой файловой системе должны присутствовать утилиты, необходимые для создания, восстановления или ремонта файловых систем, а также для административного восстановления (backup) системы. Каталоги /usr, /opt, /var спроектированы так, что они могут размещаться на файловых системах, отличных от корневой. В дистрибутиве Slackware в корневом каталоге по умолчанию находится ядро операционной системы (что на больших винчестерах иногда вызывало определенные проблемы), в дистрибутиве Red Hat ядро операционной системы перенесено в каталог /boot.

Имеется несколько причин, по которым размер корневой файловой системы рекомендуется делать минимально возможным:

- это позволяет монтировать файловую систему с очень маленьких носителей информации (например, дискет);
- корневая файловая система не может быть разделяемой, потому что содержит много системно-зависимых конфигурационных файлов. Создание малой по объему корневой файловой системы позволяет сохранить на серверах больше места для разделяемых ресурсов;
- у маленького по объему корневого каталога меньше вероятность пострадать при крахе системы.

Каталог /bin

Содержит важные исполняемые файлы, которые используются всеми пользователями, в том числе и администратором системы. Кроме того, в каталоге /bin должны находиться исполняемые файлы, необходимые для функционирования системы в однопользовательском режиме (single mode). Он также может содержать исполняемые файлы, которые напрямую задействованы в скриптах. Каталог /bin не должен содержать подкаталогов. Исполняемые файлы, от которых напрямую не зависит функционирование системы, рекомендуется размещать во вторичной иерархии — в каталоге /usr/bin.

Таким образом, в каталоге /bin должны находиться следующие утилиты (файлы или символические ссылки на команды):

- cat — выдает на стандартное устройство вывода объединенные файлы;
- chgrp — позволяет изменить группу владельца файла;
- chmod — изменяет права доступа к файлу;
- chown — изменяет владельца и группу файла;
- cp — позволяет копировать файлы и каталоги;
- date — позволяет вывести или установить системные дату и время;
- dd — конвертирует и копирует файлы;
- df — показывает использование дискового пространства;
- dmesg — выводит или управляет буфером сообщения ядра;
- echo — отображает строку текста;
- false — возвращает значение "Не успешно" (unsuccessfully);

- hostname — показывает или устанавливает имя хоста;
- kill — посылает управляющие сигналы процессам;
- ln — создает линки (связи, ссылки) между файлами;
- login — начинает сессию в системе;
- ls — показывает содержимое каталога;
- mkdir — позволяет создавать каталог;
- mknod — создает блочные или символьные специальные файлы;
- more — позволяет просматривать текстовые файлы постранично;
- mount — монтирует файловую систему;
- mv — перемещает или переименовывает файлы;
- ps — показывает статус процессов;
- pwd — выводит имя текущего рабочего каталога;
- rm — удаляет файлы или каталоги;
- rmdir — удаляет пустой каталог;
- sed — редактор;
- setserial — настраивает последовательные порты;
- sh — командная оболочка Bourne;
- sty — изменяет и выводит установки терминальной линии;
- su — изменяет пользовательский идентификатор (user ID);
- sync — сбрасывает (flush) буферы файловой системы;
- true — возвращает значение "успешно" (successfully);
- umount — размонтирует файловые системы;
- uname — выводит системную информацию.

Если в системе отсутствует утилита sh, то sh должна быть ссылкой на используемую системой командную оболочку.

Если установлены соответствующие пакеты, в каталоге /bin могут присутствовать следующие программы или символические ссылки:

- csh — командная оболочка C shell;
- ed — редактор;
- tar — архивная утилита;
- cpio — архивная утилита;
- gzip — утилита архивации файлов GNU;
- gunzip — утилита разархивации файлов GNU;
- zcat — утилита разархивации файлов GNU;
- netstat — утилита сетевой статистики;
- ping — ICMP-сетевая утилита.

Каталог /boot

Содержит все, что требуется для процесса загрузки, исключая файлы конфигурации. В каталоге /boot находятся данные, используемые ядром до того, как оно начинает исполнять программы пользовательского режима (user-mode). В этом же каталоге может находиться сохраненный сектор master boot и другие специфичные данные. Конфигурационные файлы загрузчика находятся в каталоге /etc. Ядро операционной системы, как было сказано ранее, должно находиться или в корневом

каталоге (дистрибутив Slackware), или в каталоге /boot (дистрибутив Red Hat). В некоторых случаях приходится создавать отдельный раздел /boot, находящийся до 1024 цилиндра. Это зависит от версии загрузчика и от BIOS компьютера. В этом каталоге, как правило, находится подкаталог /grub, содержащий конфигурационные файлы загрузчика системы. Таким образом, в каталоге /boot версии Linux Red Hat должны находиться следующие файлы или символические ссылки на команды:

boot.0300	kernel.h-2.6.3	module-info-2.6.2-2 os2_d.b	vmlinuz@
boot.b	map	System.map@	vmlinuz-2.6.2-2
chain.b	message	System.map-2.6.2-2	
kernel.h@	module-info@	vmlinux-2.6.2-2*	

Каталог /dev

Содержит файлы устройств или специальные файлы. Создание в каталоге /dev файлов устройств осуществляется с помощью утилиты makedev, находящейся в нем же. В этом каталоге может также присутствовать утилита makedev.local, предназначенная для создания локальных устройств. Все устройства и специальные файлы описываются в документе Linux Allocated Devices, который поставляется вместе с исходным кодом ядра.

Каталог /etc

Каталог содержит конфигурационные файлы и каталоги, которые специфичны для данной системы. В этом каталоге не должно находиться никаких исполняемых модулей. В каталоге /etc обязательно должен присутствовать каталог /opt, содержащий конфигурационные файлы для программ, установленных в каталоге /opt.

> ***Замечание***
>
> Везде, где далее упоминается "… должны присутствовать в каталоге /etc", нужно учитывать, что необходимые файлы и каталоги появляются в /etc только в том случае, если соответствующие программы установлены в системе.

В каталоге /etc также должны присутствовать следующие каталоги:

- /cron.d — конфигурация cron;
- /cron.daily — ежедневно выполняемые операции cron и anacron;
- /cron.hourly — ежечасно выполняемые операции cron и anacron;
- /cron.monthly — ежемесячно выполняемые операции cron и anacron;
- /cron.weekly — еженедельно выполняемые операции cron и anacron;
- /default — файлы, используемые пакетом shadow при создании новой учетной записи пользователя в системе;
- /gnome — разнообразная конфигурационная информация, касающаяся графической системы GNOME и ее приложений (информацию о конфигурации GNOME и ее приложений смотрите в руководстве пользователя GNOME);
- /kde — разнообразная конфигурационная информация, касающаяся графической системы KDE и ее приложений (информацию о конфигурации KDE и ее приложений смотрите в руководстве пользователя KDE);

- /locale — настройки локали;
- /opt — конфигурационные файлы для пакетов, устанавливаемых в каталоге /opt. Для каждого пакета создается (точно так же, как и в /opt) свой каталог, с точно таким же именем, как и в /opt, в котором содержатся конфигурационные файлы для этого пакета;
- /ppp — конфигурационные файлы и скрипты, необходимые для функционирования демона pppd. В частности здесь находятся скрипты, поднимающие и опускающие PPP-интерфейс с поддержкой IPv4 и IPv6, скрипты аутентификации и конфигурационные файлы;
- /rc.d — каталог скриптов, используемых при старте системы;
- samba — конфигурационные файлы для сервера Samba. Вот список файлов, которые обычно содержатся в этом каталоге:
  - lmhosts — список хостов и соответствующих им адресов;
  - smbpasswd — пароли пользователей сервера Samba;
  - smbusers — файл, предназначенный для хранения конфигурационных файлов пользователей, которым разрешен доступ к ресурсам Samba;
  - smb.conf — главный конфигурационный файл сервера;
- /sgml — конфигурации для SGML и XML;
- /skel — конфигурационные файлы для вновь создаваемых пользователей. В этом каталоге хранятся конфигурационные файлы пользователя, которые при создании нового пользователя в системе копируются в его домашний каталог. Это очень удобно с точки зрения системного администратора — один раз настроив окружение пользователя, мы для вновь созданных пользователей получаем уже готовую рабочую среду. Мы можем определить язык, раскладку клавиатуры, палитру, редактор по умолчанию, графическую оболочку и многое-многое другое. Не следует думать, что этим мы ограничиваем пользователя, наоборот, он получает настроенное рабочее место. Если ему что-то не подходит — он может внести необходимые ему изменения в *свои* конфигурационные файлы. Таким образом, мы получаем, с одной стороны, единообразие, а с другой — возможности для индивидуализации рабочего места.

  Обычно в этом каталоге находятся следующие файлы:
  - .bashrc
  - .bash_logout
  - .less
  - .Xdefaults
  - .bash_profile
  - .inputrc
  - .xinitrc

  Однако ничто не мешает удалить или, наоборот, добавить файлы в этот каталог;
- /sysconfig — каталог с файлами системной конфигурации;
- /X11 — конфигурационные файлы для X Window System.

Кроме перечисленных каталогов в /etc должны находиться следующие файлы:

- aliases — файл определяет для программы доставки почтовых сообщений, куда посылать письма, приходящие на адрес псевдопользователей. Большей частью они перенаправляются пользователю root;
- anacrontab — конфигурационный файл для программы anacron. В этом файле задаются периодичность выполнения команд (ежедневно, еженедельно, ежемесячно) и каталоги, в которых содержатся исполняемые модули (как правило — скрипты).

Программа anacron использует те же каталоги с исполняемыми модулями, что и cron. Однако anacron применяется в системах, которые не предназначены для постоянного функционирования (24 часа в сутки). Программа просматривает список задач и запускает текущие в списке или *просроченные*;

- at.allow — задает список пользователей, которым разрешена команда `at`;
- at.deny — задает список пользователей, которым запрещена команда `at`;
- bashrc — конфигурационный файл, определяющий поведение bash. Как правило, не требует ручного вмешательства;
- cron.allow — задает список пользователей, которым разрешено пользоваться демоном cron;
- cron.deny — задает список пользователей, которым запрещено пользоваться демоном cron;
- crontab — конфигурационный файл для программы cron. В этом файле задаются периодичность выполнения команд (ежечасно, ежедневно, еженедельно, ежемесячно) и каталоги, в которых содержатся исполняемые модули (как правило — скрипты).

Программа cron рассчитана на постоянно функционирующие системы. Если во время, когда компьютер отключен, необходимо было выполнить какую-то операцию, то программа cron не поможет. Для выполнения просроченных операций необходима программа anacron;

- cron.allow — программа cron может разрешать или запрещать доступ конкретным пользователям. Имена пользователей, которым разрешен доступ к cron, необходимо вписать в файл cron.allow;
- cron.deny — конфигурационный файл для программы cron, с помощью которого можно запретить использование программы cron конкретным пользователям или всем пользователям кроме тех, которые записаны в файле cron.allow;
- dir_colors — определяет, каким цветом команда `ls` будет отображать файлы на экране. Для разных типов файлов можно определить свой цвет;
- exports — содержит управление доступом к файловой системе NFS;
- fstab — содержит таблицу, в которой определены монтируемые устройства (файлы драйверов), соответствующие им точки монтирования, тип файловой системы и параметры монтирования. Пример файла fstab приведен в листинге 6.1.

Листинг 6.1

```
LABEL=/          /           ext3      defaults              1 1
LABEL=/boot      /boot       ext2      defaults              1 2
none             /dev/pts    devpts    gid=5,mode=620        0 0
none             /proc       proc      defaults              0 0
none             /dev/shm    tmpfs     defaults              0 0
/dev/hda8        swap        swap      defaults              0 0
/dev/cdrom       /mnt/cdrom  iso9660   noauto,owner,kudzu,ro 0 0
/dev/fd0         /mnt/floppy auto      noauto,owner,kudzu    0 0
```

- ftpusers — конфигурационный файл FTP-демона, содержащий список пользователей FTP с их правами доступа;
- gateways — содержит список шлюзов (gateways) для демона маршрутизации routed;
- gettydefs — содержит терминальные установки, используемые getty;
- group — содержит пользователей и группы, членами которых они являются. Файл состоит из строк с четырьмя полями в каждой:
  - имя пользователя;
  - пароль;
  - GUID — числовой идентификатор группы;
  - список имен групп, к которым принадлежит пользователь.

  Пример файла group приведен в листинге 6.2.

Листинг 6.2

```
root:x:0:root
bin:x:1:root,bin,daemon
daemon:x:2:root,bin,daemon
sys:x:3:root,bin,adm
adm:x:4:root,adm,daemon
wheel:x:10:root
mail:x:12:mail
news:x:13:news
nobody:x:99:
users:x:100:
slocate:x:21:
floppy:x:19:
utmp:x:22:
mysql:x:27:
alst:x:500:
```

- gshadow — содержимое этого файла (листинг 6.3) напоминает содержимое файла group.

Листинг 6.3

```
root:::root
bin:::root,bin,daemon
daemon:::root,bin,daemon
sys:::root,bin,adm
adm:::root,adm,daemon
disk:::root
lp:::daemon,lp
mem:::
```

```
kmem:::
wheel:::root
utmp:x::
mailnull:x::
mysql:x::
alst:!::
```

- host.conf — конфигурационный файл, который определяет порядок разрешения символического имени хоста в IP-адресе. Обычно содержимое этого файла имеет вид:
  ```
  order hosts,bind
  ```
- hostname — обычно содержит имя хоста. Текущее имя хоста можно посмотреть с помощью команды `hostname`;
- hosts — содержимое этого файла используется для определения пары "IP-адрес — символическое имя хоста". Очень рекомендуется, чтобы в этом файле была следующая запись:
  ```
  127.0.0.1 localhost.localdomain localhost
  ```

  Замечание

  Если эта строчка отсутствует, возникнут проблемы, связанные с сетью (в частности возможно зависание программы sendmail).

- hosts.allow — определяет, каким хостам разрешено подключаться к системе;
- hosts.deny — определяет, каким хостам запрещено подключаться к системе;
- hosts.equiv — содержит список доверенных хостов для rlogin, rsh, rcp;
- hosts.lpd — содержит список доверенных хостов для lpd;
- inetd.conf — конфигурационный файл для демона inetd;
- inittab — конфигурационный файл для процесса init. Описывает, как процесс init должен настроить операционную систему в соответствующем уровне исполнения;
- issue — содержит сообщение, выдаваемое системой до приглашения "login:".
- ld.so.conf — содержит список каталогов для поиска разделяемых библиотек;
- localtime — бинарный файл, определяющий временную зону компьютера, правила перехода на летнее/зимнее время и другую информацию, связанную с местной временной зоной. Обычно берется один из файлов, находящихся в каталоге /usr/share/zoneinfo/, и копируется в каталог /etc с именем localtime. В том случае, если для вас не существует готового файла localtime, его можно создать с помощью утилиты zic;
- man.config — конфигурационный файл, содержащий настройки для справочных страниц man;
- modules.conf — файл, используемый операционной системой для загрузки по требованию программ некоторых модулей ядра. Обычно необходим для модулей звуковых карт и плат TV-тюнеров или в том случае, если в системе установлено несколько сетевых плат;

- motd — сообщение, выдаваемое системой после входа пользователя в систему;
- mtab — содержит динамическую информацию о файловых системах;
- mtools.conf — конфигурационный файл для mtools;
- networks — содержит статическую информацию о сетевых именах;
- passwd — содержит информацию обо всех пользователях системы, в том числе и псевдопользователях, которые необходимы для правильного функционирования некоторых сервисов. Типичный файл passwd приведен в листинге 6.4.

Листинг 6.4

```
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
news:x:9:13:news:/var/spool/news:
uucp:x:10:14:uucp:/var/spool/uucp:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
gopher:x:13:30:gopher:/var/gopher:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
nobody:x:99:99:Nobody:/:/sbin/nologin
mailnull:x:47:47::/var/spool/mqueue:/dev/null
rpm:x:37:37::/var/lib/rpm:/bin/bash
xfs:x:43:43:X Font Server:/etc/X11/fs:/bin/false
ntp:x:38:38::/etc/ntp:/sbin/nologin
rpc:x:32:32:Portmapper RPC user:/:/bin/false
gdm:x:42:42::/var/gdm:/sbin/nologin
rpcuser:x:29:29:RPC Service User:/var/lib/nfs:/sbin/nologin
nfsnobody:x:65534:65534:Anonymous NFS User:/var/lib/nfs:/sbin/nologin
nscd:x:28:28:NSCD Daemon:/:/bin/false
ident:x:98:98:pident user:/:/sbin/nologin
radvd:x:75:75:radvd user:/:/bin/false
apache:x:48:48:Apache:/var/www:/bin/false
squid:x:23:23::/var/spool/squid:/dev/null
pcap:x:77:77::/var/arpwatch:/bin/nologin
mysql:x:27:27:MySQL Server:/var/lib/mysql:/bin/bash
vasya:x:500:500::/home/vasya:/bin/bash
```

Каждая строка листинга 6.4 определяет одного пользователя. В строке есть семь полей, отделенных друг от друга двоеточием. Рассмотрим более подробно эти поля:

- имя пользователя;
- пароль пользователя; в современных системах не используется (указан символ `x`). Пароль хранится в файле shadow;
- UID — числовой идентификатор пользователя;
- GID — числовой идентификатор группы пользователя;
- поле описания пользователя (телефон, адрес и т. п.). Обычно не используется;
- домашний каталог пользователя;
- командный интерпретатор, применяемый пользователем.

Никто, кроме пользователя root, не имеет доступа на запись в файл passwd. Таким образом, если вам необходимо запретить пользователю регистрироваться в системе, можно назначить ему командный интерпретатор /sbin/nologin или /dev/null. Другой способ — отредактировать пароль (смотри shadow);

- printcap — база совместимых принтеров для lpd;
- profile — общесистемный конфигурационный файл для оболочки sh. Все настройки, добавленные в этот файл, влияют на переменные окружения вашей системы;
- protocols — содержит список IP-протоколов;
- resolv.conf — конфигурационный файл Resolver;
- rpc — содержит RPC-список протоколов;
- securetty — управление доступом к консоли для администратора (root);
- sendmail.cf — конфигурационный файл программы sendmail для передачи почтовых сообщений;
- services — содержит имена портов для сетевых сервисов. Описание сервиса представляет собой строку, состоящую из четырех полей:
  - имя сервиса;
  - номер порта/протокол;
  - псевдонимы;
  - комментарии.

  Пример записи в файле services:

```
tcpmux          1/tcp           # TCP port service multiplexer
tcpmux          1/udp           # TCP port service multiplexer
```

- shadow — файл паролей, по структуре напоминающий passwd. Содержит полный список пользователей системы. Интересны первые два поля — имя пользователя и хэш пароля. Алгоритм создания хэша пароля работает так, что первым символом хэша *никогда* не может быть символ `*`. Чтобы запретить пользователю вход в систему, достаточно первым символом пароля поставить `*`.

  Пример файла shadow содержит листинг 6.5.

Листинг 6.5

```
root:$1$є¦zA¦+NЎ$ZL.87fvylY.:11689:0:99999:7:::
bin:*:11689:0:99999:7:::
daemon:*:11689:0:99999:7:::
adm:*:11689:0:99999:7:::
lp:*:11689:0:99999:7:::
```

- shells — содержит пути для установленных командных оболочек;
- sysctl.conf — файл конфигурации ядра операционной системы. Позволяет настраивать различные свойства ядра — сетевые и не только.

Пример файла sysctl.conf приведен в листинге 6.6.

Листинг 6.6

```
# Disables packet forwarding
net.ipv4.ip_forward = 0
# Enables source route verification
net.ipv4.conf.default.rp_filter = 1
# Disables the magic-sysrq key
kernel.sysrq = 0
```

- syslog.conf — конфигурационный файл для демона syslogd.

/etc/rc.d — инициализационные скрипты системы

В каталоге содержатся следующие файлы и каталоги, необходимые для загрузки операционной системы Linux и запуска требуемых сервисов:

- /init.d — каталог с управляющими скриптами для загружаемых при старте операционной системы сервисов;
- /rc0.d — каталог со скриптами, отвечающими за запуск и остановку сервисов при переходе на нулевой уровень выполнения;
- /rc1.d — каталог со скриптами, отвечающими за запуск и остановку сервисов при переходе на первый уровень выполнения;
- /rc2.d — каталог со скриптами, отвечающими за запуск и остановку сервисов при переходе на второй уровень выполнения;
- /rc3.d — каталог со скриптами, отвечающими за запуск и остановку сервисов при переходе на третий уровень выполнения;
- /rc4.d — каталог со скриптами, отвечающими за запуск и остановку сервисов при переходе на четвертый уровень выполнения;
- /rc5.d — каталог со скриптами, отвечающими за запуск и остановку сервисов при переходе на пятый уровень выполнения;
- /rc6.d — каталог со скриптами, отвечающими за запуск и остановку сервисов при переходе на шестой уровень выполнения;

- rc — файл для запуска и останова сервисов при переходе на указанный уровень выполнения;
- rc.local — файл для команд, добавляемых администратором для запуска в процессе начальной загрузки;
- rc.sysinit — файл для выполнения начальных действий, необходимых для корректного функционирования операционной системы.

/etc/rc.d/init.d — управляющие скрипты для сервисов

Каталог содержит управляющие скрипты для сервисов, которые выполняются (или могут выполняться) при старте системы или при переходе с одного уровня выполнения на другой.

Если соответствующие сервисы установлены, в этом каталоге находятся следующие файлы:

anacron	ipchains	nfslock	sendmail
apmd	iptables	nscd	single
arpwatch	isdn	portmap	snmpd
atd	kdcrotate	pppoe	sshd
autofs	keytable	random	syslog
crond	killall	awdevices	tux
functions	kudzu	rhnsd	windows
gpm	lpd	rstatd	xfs
halt	netfs	rusersd	xinetd
httpd	network	rwalld	ypbind
identd	nfs	rwhod	ppasswdd
			ypserv

/etc/rc.d/rc0.d … rc6.d — каталоги для соответствующего уровня выполнения

Эти каталоги содержат стартовые и стоповые скрипты сервисов, используемых операционной системой при переходе на нужный уровень выполнения. К примеру, каталог /rc3.d конкретного компьютера может содержать следующие файлы:

K03rhnsd	K50tux	S13portmap	S56xinetd
K15httpd	K65identd	S14nfslock	S60lpd
K20nfs	K73ypbind	S17keytable	S80isdn
K20rstatd	K74nscd	S20random	S80pppoe
K20rusersd	K74ypserv	S25netfs	S80sendmail
K20rwalld	S05kudzu	S26apmd	S85gpm
K20rwhod	S08ipchains	S28autofs	S90crond
K34yppasswdd	S08iptables	S40atd	S90xfs
K45arpwatch	S10network	S55sshd	S95anacron
K50snmp	S12syslog	S56rawdevices	S99local

S99windows/etc/sysconfig — конфигурационные файлы для процессов

Каталог содержит различные конфигурационные файлы и скрипты, необходимые операционной системе во время загрузки и останова сервисов.

В частности в нем находятся следующие файлы и каталоги:

- /etc/sysconfig/apm-scripts — каталог со скриптами, относящимися к демону apmd, предназначенному для управления питанием системы. Чаще всего встречается в системах, установленных на ноутбуках;
- /etc/sysconfig/cbq — каталог для конфигурирования программы cbq (так называемого трафик-шейпера). Принцип действия — искусственно ограничивает полосу пропускания сетевого устройства с заданной шириной канала;
- /etc/sysconfig/console — каталог для конфигурирования консоли. В частности /etc/sysconfig/console/default.kmap — файл раскладки клавиатуры по умолчанию;
- /etc/sysconfig/network/ — каталог хранит различные настройки сети, а также скрипты, отвечающие за старт и останов сетевой подсистемы;
- /etc/sysconfig/clock — файл служит для конфигурирования системных часов (временная зона, формат хранения времени, переход на летнее/зимнее время и т. п.);
- /etc/sysconfig/i18n/ — каталог содержит файлы, связанные с локализацией системы, в частности шрифты.

В самом каталоге /sysconfig находятся следующие файлы:

- apmd — конфигурация демона управления электропитанием;
- arpwatch — конфигурация программы arpwatch;
- clock — конфигурация часовой зоны и некоторые другие параметры.

Пример:

```
ZONE="Europe/Kiev"
UTC=false
ARC=false
```

Как видно из примера, системные часы не используют универсальное представление времени, а система находится в часовом поясе Киева (Гринвич + 2 часа);

- gpm — предназначен для конфигурирования gpm — программы, осуществляющей поддержку мыши в консоли;
- harddisks — предназначен для тонкой настройки производительности жестких дисков. Смотрите также описание программы hdparm;
- hwconf — содержит базу обнаруженных и сконфигурированных устройств программой kudzu. Пример приведен в листинге 6.7.

Листинг 6.7

```
-
class: OTHER
bus: PCI
detached: 0
driver: agpgart
desc: "Intel Corporation|82815 815 Chipset Host Bridge and Memory Con-
troller Hub"
vendorId: 8086
deviceId: 1130
subVendorId: 8086
```

```
subDeviceId: 1130
pciType: 1
-
class: OTHER
bus: PCI
detached: 0
driver: unknown
desc: "Intel Corporation|unknown device 8086:1131"
vendorId: 8086
deviceId: 1131
subVendorId: 0000
subDeviceId: 0000
pciType: 1
-
class: OTHER
bus: PCI
detached: 0
driver: unknown
desc: "Intel Corporation|82820 820 (Camino 2) Chipset PCI"
vendorId: 8086
deviceId: 244e
subVendorId: 0000
subDeviceId: 0000
pciType: 1
-
class: OTHER
bus: PCI
detached: 0
driver: i810-tco
desc: "Intel Corporation|82820 820 (Camino 2) Chipset ISA Bridge
(ICH2)"
vendorId: 8086
deviceId: 2440
subVendorId: 0000
subDeviceId: 0000
pciType: 1
-
class: OTHER
bus: PCI
detached: 0
driver: unknown
desc: "Intel Corporation|82820 820 (Camino 2) Chipset IDE U100"
```

```
vendorId: 8086
deviceId: 244b
subVendorId: 8086
subDeviceId: 244b
pciType: 1
-
class: OTHER
bus: PCI
detached: 0
driver: unknown
desc: "Intel Corporation|82820 820 (Camino 2) Chipset SMBus"
vendorId: 8086
deviceId: 2443
subVendorId: 8086
subDeviceId: 244b
pciType: 1
-
class: OTHER
bus: PCI
detached: 0
driver: btaudio
desc: "Brooktree Corporation|Bt878"
vendorId: 109e
deviceId: 0878
subVendorId: 0000
subDeviceId: 0000
pciType: 1
-
class: OTHER
bus: USB
detached: 0
driver: unknown
desc: "USB UHCI Root Hub"
usbclass: 9
usbsubclass: 0
usbprotocol: 0
usbbus: 1
usblevel: 0
usbport: 0
vendorId: 0000
deviceId: 0000
productrevision: unknown
```

```
-
class: OTHER
bus: USB
detached: 0
driver: unknown
desc: "USB UHCI Root Hub"
usbclass: 9
usbsubclass: 0
usbprotocol: 0
usbbus: 2
usblevel: 0
usbport: 0
vendorId: 0000
deviceId: 0000
productrevision: unknown
-
class: MOUSE
bus: PSAUX
detached: 0
device: psaux
driver: generic3ps/2
desc: "Generic 3 Button Mouse (PS/2)"
-
class: AUDIO
bus: PCI
detached: 0
driver: i810_audio
desc: "Intel Corporation|82801BA/BAM (ICH2) AC'97 Audio Controller"
vendorId: 8086
deviceId: 2445
subVendorId: 11d4
subDeviceId: 5360
pciType: 1
-
class: CDROM
bus: IDE
detached: 0
device: hdc
driver: ignore
desc: "SONY CD-ROM CDU4821"
-
class: VIDEO
```

```
bus: PCI
detached: 0
driver: Card:NVIDIA GeForce 2 MX (generic)
desc: "nVidia Corporation|NV11"
vendorId: 10de
deviceId: 0110
subVendorId: 0000
subDeviceId: 0000
pciType: 1
-
class: FLOPPY
bus: MISC
detached: 0
device: fd0
driver: unknown
desc: "3.5" 1.44MB floppy drive"
-
class: HD
bus: IDE
detached: 0
device: hda
driver: ignore
desc: "FUJITSU MPG3409AT E"
physical: 79428/16/63
logical: 4983/255/63
-
class: CAPTURE
bus: PCI
detached: 0
driver: bttv
desc: "Brooktree Corporation|Bt878"
vendorId: 109e
deviceId: 036e
subVendorId: 0000
subDeviceId: 0000
pciType: 1
-
class: USB
bus: PCI
detached: 0
driver: usb-uhci
desc: "Intel Corporation|82820 820 (Camino 2) Chipset USB (Hub A)"
```

```
vendorId: 8086
deviceId: 2442
subVendorId: 8086
subDeviceId: 244b
pciType: 1
-
class: USB
bus: PCI
detached: 0
driver: usb-uhci
desc: "Intel Corporation|82820 820 (Camino 2) Chipset USB (Hub B)"
vendorId: 8086
deviceId: 2444
subVendorId: 8086
subDeviceId: 244b
pciType: 1
-
class: MODEM
bus: SERIAL
detached: 1
device: ttyS0
driver: ignore
desc: "IDC|5620 IDC 5614BXL VR PnP"
pnpmfr: IDC
pnpmodel: 5620
pnpcompat: *PNPC107
pnpdesc: IDC 5614BXL VR PnP
```

- i18n — файл, отвечающий за локализацию системы. Название расшифровывается как internationalization (между i и n восемнадцать букв).

  Пример:

```
LANG="ru_RU.koi8r"
SUPPORTED="ru_RU.koi8r:ru_RU:ru"
SYSFONT="cyr-sun16"
SYSFONTACM="koi8-u"
```

  В примере определено, что у нас русская локаль с кодировкой KOI8-R и шрифтом cyr-sun16;
- identd — конфигурационный файл демона identd, реализующего поддержку протокола идентификации пользователя;
- keyboard — этот файл, находящийся в каталоге /etc/sysconfig, отвечает за конфигурирование клавиатуры.

Для настройки клавиатуры (раскладки и скорости повтора) необходимы следующие операции:

- выбрать раскладку клавиатуры (описания раскладки клавиатуры находятся в каталоге /usr/lib/kbd/keytables/ в файлах с расширением map);
- внести изменения в файл keyboard так, чтобы он содержал строку `KEYTABLE="/usr/lib/kbd/keytables/zzz.map"`, где `zzz` — имя раскладки клавиатуры;
- для настройки скорости повтора нажатия и времени задержки необходимо добавить в файл следующую строку: `/etc/rc.d/rc.sysinit` или, если у вас Caldera, к /etc/rc.d/rc.boot: `/sbin/kbdrate -s -r 16 -d 500` — где `-r 16` — число символов, а `-d 500` — задержка в миллисекундах.

  Пример стандартного файла keyboard:

```
KEYBOARDTYPE="pc"
KEYTABLE="ru"
```

- ❒ kudzu — отвечает за поведение программы kudzu при перезапуске системы — как она будет себя вести при обнаружении нового оборудования;
- ❒ mouse — определяет параметры мыши, эмуляцию нажатия третьей кнопки и файл устройства. Листинг 6.8 содержит пример файла mouse.

Листинг 6.8

```
MOUSETYPE="imps2"
XMOUSETYPE="IMPS/2"
FULLNAME="Microsoft IntelliMouse (PS/2)"
XEMU3=no
DEVICE=/dev/mouse
```

- ❒ sendmail — определяет, как стартует программа sendmail и через какой интервал времени отправляется почтовая очередь;
- ❒ squid — настройки программы squid;
- ❒ syslog — настройка демона syslog;
- ❒ xinetd — настройка демона xinetd, который в Linux заменяет inetd.

/etc/X11 — конфигурационные файлы для X Window System

Каталог содержит конфигурационные файлы X11, специфичные для данного хоста. Если соответствующие пакеты установлены, здесь должны находиться следующие файлы или символические связи:

- ❒ Xconfig — конфигурационный файл для ранних версий XFree86;
- ❒ XF86Config — конфигурационный файл для XFree86 версии 3 и 4;
- ❒ Xmodmap — глобальный файл клавиатурных раскладок X11.

/etc/sgml — конфигурационные файлы для SGML и XML

Каталог содержит базовые конфигурационные файлы для определения параметров высокого уровня для SGML или XML. Имена *.conf соответствуют базовым конфигурационным файлам. Файлы с именами *.cat — DTD-специфичные централизированные каталоги, содержащие руководства по всем остальным каталогам.

Каталог /home — пользовательские домашние каталоги

В каталоге находятся домашние каталоги пользователей. Как правило, каждый пользователь в небольшой системе имеет свой домашний каталог, и его имя совпадает с именем (login) пользователя. Например, у пользователя frozzy домашний каталог — /home/frozzy. Типичное содержимое каталога пользователя, только что зарегистрированного в системе, включает следующие файлы и каталоги:

/cedit	/.gnome-desktop	/.netscape	.bash_profile
/Desktop	/.gnome_private	/nsmail	.bashrc
/.dia	/.gnp	/.sawfish	.ICEauthority
/.gimp-1.2	/kde	.bash_history	.screenrc
/.gnome	/.mc	.bash_logout	

Как можно видеть, это, в основном, конфигурационные файлы программ, установленных в операционной системе.

Каталог /lib — важные разделяемые библиотеки и модули ядра

Каталог содержит разделяемые библиотеки, необходимые для загрузки системы и запуска команд в корневой файловой системе, т. е. только для файлов, находящихся в каталоге /bin и /sbin. По меньшей мере одна из групп файлов должна находиться в каталоге /lib:

- libc.so.* — динамически подключаемая (линкуемая) библиотека C;
- ld* — линкер/загрузчик (linker/loader) времени выполнения.

Следующие каталоги также должны находиться в /lib:

- /modules — загружаемые модули ядра;
- /security — модули PAM.

Каталог /lib64 — важные разделяемые библиотеки и модули ядра

Этот каталог появляется в системах, построенных на 64-разрядных процессорах, например, PPC 64, AMD 64, IBM s390x и sparc64. Структуры /lib64 и /lib (для 32-разрядных систем) полностью совпадают. Хранит 64-разрядные версии библиотек.

Каталог /lost+found

Каталог, который обязательно должен присутствовать в каждом разделе. Если, к примеру, винчестер разбит на три раздела, которые монтируются в /, /home, /var, то в корневой файловой системе, в каталогах /home и /var будет присутствовать /lost+found. Назначение этого каталога достаточно очевидно — при аварийных ситуациях возможна потеря информации. Специальная утилита chkfsk восстанавливает (если, конечно, это возможно) утерянную информацию. Однако иногда нельзя достоверно определить принадлежность данных какому-нибудь определенному файлу. В этом случае восстановленные данные помещаются в каталог /lost+found.

Каталог /media — точка монтирования автоматически монтируемых устройств

Каталог служит в качестве точки монтирования дискет и CD-ROM программой automount. Может содержать каталоги:

- floppy — точка монтирования дискет;
- cdrom — точка монтирования CD-ROM;
- cdrecorder — точка монтирования записываемых дисков;
- zip — точка монтирования ZIP-накопителей.

Каталог /mnt — точка монтирования для временно монтируемой файловой системы

Каталог предназначен для того, чтобы системный администратор мог временно монтировать файловую систему (например, дискету или CD-ROM). В различных дистрибутивах Linux в каталоге /mnt могут находиться каталоги, являющиеся точками монтирования дискет, разделов жесткого диска, CD-ROM и т. п. Если в каталоге /mnt есть какие-то файлы, и к нему монтируется некий раздел, то файлы, находящиеся в каталоге /mnt, становятся недоступными до тех пор, пока не размонтируют раздел, подмонтированный к /mnt.

Каталог /opt — дополнительные программные пакеты

Каталог зарезервирован для инсталляции дополнительного программного обеспечения. Пакет, который устанавливается в каталог /opt, должен хранить свои неизменяемые файлы в каталоге /opt/<имя_пакета>, где <имя_пакета> — имя устанавливаемого пакета. Структура поддерева каталогов в каталоге <имя_пакета>:

/bin; /doc; /lib; /man и т. д.

Исполняемые модули нужно размещать в каталоге /bin, а если пакет включает в себя документацию, ее следует сохранить в каталоге /doc. При наличии страниц справочной системы надо разместить их в /opt/<имя_пакета>/man и использовать подструктуру каталогов, как в /usr/share/man. Специфичные для конкретного пакета библиотеки размещают в /opt/<имя_пакета>/lib и т. д. Файлы пакета, которые могут изменяться, должны быть установлены в каталоге /var/opt, хост-специфичные конфигурационные файлы — в /etc/opt.

Каталог /proc — точка монтирования виртуальной файловой системы procfs

Procfs — псевдофайловая система, обеспечивающая интерфейс с ядром Linux, — позволяет получить доступ к определенным структурам данных ядра, в частности к списку процессов (отсюда и название). Все эти структуры выглядят как файловая система, и ими можно оперировать обычными средствами работы с файловой системой.

Структура каталогов в /proc:

- /1 — подкаталог процесса, имя каталога соответствует номеру PID-процесса;
- /2;
- /3;
- /4;
- /5;
- /6;
- /7;
- …/384;
- /389;
- /403;
- /418;
- /490;
- /5196;
- /bus — каталог содержит специфичную информацию, касающуюся шин (PCI, ISA);
- /driver — здесь сгруппированы различные драйверы;
- /fs — параметры файловых систем;
- /ide — информация о IDE-подсистеме;
- /irq — маски для управления аппаратными прерываниями;
- /net — сетевая информация;
- /sys — системная информация;
- /sysvipc — информация о SysVIPC-ресурсах (msg, sem, shm);
- /tty — информация о TTY-драйверах;
- apm — расширенная информация управлением питанием;
- cmdline — командная строка ядра операционной системы;
- cpuinfo — информация о микропроцессоре;
- devices — доступные устройства (блочные и символьные);
- dma — используемые каналы DMA;
- execdomains — используемые домены;
- fb — Frame Buffer-устройства;
- filesystems — поддерживаемые файловые системы;
- interrupts — задействованные прерывания;
- iomem — карта памяти;
- ioports — используемые порты ввода/вывода;
- isapnp — информация о ISA-устройствах;
- kcore — образ ядра операционной системы;
- kmsg — сообщения ядра;
- ksyms — таблица символов ядра;
- loadavg — средняя загрузка за последние 1, 5 и 15 минут;
- locks — "защелки" ядра;
- mdstat — файл, сообщающий о конфигурации RAID-массива системы;
- meminfo — информация о памяти;
- misc — различная информация, не попавшая ни в одну из категорий;

- modules — список загруженных модулей;
- mounts — смонтированные файловые системы;
- mtrr — управление использованием памяти;
- partitions — список разделов, известных системе;
- pci — устаревшая информация о PCI-шине (см. /proc/bus/pci/);
- rts — часы реального времени;
- scsi — информация о SCSI-устройствах;
- self — символическая ссылка к каталогу процесса, пытающегося получить информацию из /proc;
- slabinfo — информация о Slab;
- stat — разнообразная статистика;
- swaps — использование разделов и файлов подкачки;
- uptime — время работы системы без перезагрузки;
- version — версия ядра;
- video — BTTV-информация о видеоресурсах.

/proc/№процесса_PID-процесса

Имя каталога соответствует номеру PID-процесса. Каждый процесс в системе имеет соответствующий ему каталог в /proc. В этом каталоге обязательно находятся следующие файлы:

- cmdline — файл, содержащий аргументы командной строки процесса;
- cpu — текущий и последний использовавшиеся микропроцессоры (только для мультипроцессорных систем);
- /cwd — ссылка на текущий рабочий каталог;
- environ — значения переменных окружения;
- exe — ссылка на исполняемый файл этого процесса;
- /fd — каталог, содержащий все файловые дескрипторы данного процесса;
- maps — карты памяти исполняемых и библиотечных файлов;
- mem — память, занятая этим процессом;
- /root — ссылка на корневой каталог этого процесса;
- stat — статус процесса;
- statm — информация об использовании процессом памяти;
- status — статус процесса в форме, воспринимаемой человеком.

/proc/ide — IDE-устройства, установленные в системе

В каталоге содержится информация обо всех установленных в системе IDE-устройствах, в том числе о драйверах.

/proc/net — сетевая информация

В этом каталоге содержится информация, относящаяся к сети. Следующие файлы общие как для протокола IPv4, так и IPv6:

- arp — ARP-таблица ядра;
- dev — сетевые устройства со своей статистикой;
- dev_stat — статус сетевого устройства;

- ip_fwchains — связи цепочки Firewall;
- ip_fwnames — имена цепочек Firewall;
- /ip_masq — каталог содержит таблицы маскарадинга*;
- ip_masquerade — главная таблица маскарадинга;
- netstat — сетевая статистика;
- raw — статистика сетевых устройств;
- route — таблица маршрутизации ядра;
- /rpc — каталог содержит RPC-информацию;
- rt_cache — кэш маршрутизации;
- snmp — данные SNMP;
- sockstat — статистика сокетов;
- tcp — TCP-сокеты;
- tr_rif — таблица маршрутизации Token ring RIF;
- udp — UDP-сокеты;
- unix — UNIX-сокеты;
- wireless — данные беспроводного интерфейса (Wavelan и т. п.);
- igmp — IP-адреса, которые хост принимает;
- psched — параметры глобального администратора пакетов;
- netlink — список PF_NETLINK-сокетов;
- ip_mr_vifs — список виртуальных интерфейсов;
- ip_mr_cache — список кэша маршрутизации.

Перечислим файлы, используемые протоколом IPv6:

- udp6 — UDP-сокеты (IPv6);
- tcp6 — TCP-сокеты (IPv6);
- raw6 — статистика устройств (IPv6);
- igmp6 — IP-адреса, принимаемые хостом (IPv6);
- if_inet6 — список IPv6-интерфейсных адресов;
- ipv6_route — таблица маршрутизации для IPv6;
- rt6_stats — общая статистика IPv6-таблиц маршрутизации;
- sockstat6 — статистика сокетов (IPv6);
- snmp6 — SNMP-данные (IPv6).

/proc/parport — параллельные порты

Каталог содержит информацию обо всех параллельных портах, установленных в системе.

/proc/scsi — SCSI-устройства, установленные в системе

Если в компьютере установлены SCSI-устройства, то должен существовать каталог /proc/scsi, содержащий информацию обо всех установленных в системе SCSI-устройствах, в том числе о драйверах.

* Маскарадинг — подмена реального IP-адреса любого исходящего пакета на другой (специальный), а для входящего пакета — замена IP-адреса (специального) с помощью таблицы соответствия на реальный адрес компьютера, которому адресован сетевой пакет.

/proc/sys — системная информация

Каталог содержит файлы, изменением которых можно, не перегружая системы, менять различные параметры ядра.

/proc/sys/dev — информация, специфическая для устройств

На сегодняшний день поддерживаются только устройства CD-ROM.

/proc/sys/fs — данные файловой системы

Каталог содержит различную информацию о файловой системе.

/proc/sys/kernel — основные параметры ядра операционной системы

В этом каталоге находятся файлы, с помощью которых можно изменять настройки ядра операционной системы.

/proc/sys/net — сетевая "начинка"

Каталог содержит интерфейс по управлению различными сетевыми протоколами. Здесь могут находиться следующие подкаталоги:

- /802 — протокол E802;
- /appletalk — Appletalk-протокол;
- /ax25 — AX25;
- /bridge — Bridging;
- /core — основные параметры;
- /decnet — DEC-net;
- /ethernet — Ethernet-протокол;
- /ipv4 — IP версии 4;
- /ipv6 — IP версии 6;
- /ipx — IPX;
- /netrom — NET/ROM;
- /rose — X.25 PLP layer;
- /token-ring — IBM token ring;
- /unix — UNIX domain sockets;
- x25 — протокол X.25.

/proc/sys/sunrpc — удаленные вызовы процедур

Каталог содержит файлы, которые разрешают или запрещают отладку удаленно вызываемых процедур.

/proc/sys/vm — виртуальная подсистема памяти

Файлы в этом каталоге необходимы для настройки виртуальной подсистемы памяти ядра Linux.

/proc/tty — терминалы

Здесь содержится информация о доступных и используемых терминалах.

Каталог /root — домашний каталог для пользователя root (администратора)

Существенных причин для вынесения домашнего каталога /root в корневой уровень нет. Однако существует практика выделения отдельного раздела для каталога /home, который при аварийных ситуациях может не подмонтироваться. Видимо, по этой причине каталог /root вынесли на корневой уровень.

Каталог /sbin — системные исполняемые файлы

Утилиты для системного администрирования и другие, предназначенные только для администратора (пользователя root), хранятся в каталогах /sbin, /usr/sbin и /usr/local/sbin. Каталог /sbin содержит исполняемые файлы, необходимые для загрузки, восстановления, починки системы в добавление к файлам, находящимся в каталоге /bin. Программы, используемые после монтирования файловых систем, в основном помещают в каталог /usr/sbin. Административные программы, работающие только на локальной системе, помещают в каталог /usr/local/sbin.

Обычные пользователи не должны иметь доступа в каталоги /sbin. Если обычный пользователь (не администратор) может запускать команду, она должна находиться в одном из каталогов /bin. В каталоге /sbin должны присутствовать следующие файлы:

- badblocks — утилита для проверки жестких дисков;
- ctrlaltdel — программа для перезагрузки операционной системы;
- dumpe2fs — утилита для работы с файловой системой;
- e2fsck — утилита для проверки файловой системы;
- fastboot — утилита, перезагружающая систему без проверки дисков;
- fasthalt — утилита, останавливающая систему без проверки дисков;
- fdisk — утилита, позволяющая производить различные действия с таблицей разделов (создавать, редактировать, удалять раздел и т. д.);
- fsck — утилита, проверяющая и восстанавливающая файловую систему;
- fsck.* — утилита, проверяющая и восстанавливающая специфичную файловую систему (например, Ext2);
- getty — программа getty;
- halt — команда, останавливающая систему;
- ifconfig — утилита конфигурации сетевого интерфейса;
- init — Init-процесс;
- kbdrate — утилита для настройки клавиатуры;
- lilo — загрузчик операционной системы;
- mke2fs — утилита создания файловой системы;
- mkfs — команда, создающая файловую систему;
- mkfs.* — команда, создающая специфичную файловую систему;
- mkswap — команда, устанавливающая своп-область;
- reboot — команда, перезагружающая систему;
- route — утилита для таблицы IP-маршрутизации;

- swapon — утилита, разрешающая свопирование;
- swapoff — утилита, запрещающая свопирование;
- tune2fs — утилита тонкой настройки файловой системы;
- update — демон, периодически сбрасывающий буферы файловой системы.

Каталог /sys — точка монтирования файловой системы sysfs

Каталог используется системой подобно каталогу /proc. В ядро Linux относительно недавно начали внедрять Unified Device Model Of Kernel (унифицированная модель устройств для ядра системы). Подобно старой системе /proc имеем обобщенную модель взаимодействия устройств и драйверов в ядре Linux — механизм взаимодействия между ядром, устройствами, драйверами и пользователем.

Применение этой модели связано с файловой системой sysfs, которая также появилась в новом ядре для частичной замены файловой системы /proc.

Основные объекты модели:

- device — устройство;
- device_driver — драйвер устройства;
- bus_type — шина взаимодействия устройства и драйвера;
- device_attribute — атрибуты устройства;
- driver_attribute — атрибуты драйвера устройства;
- bus_attribute — атрибуты шины.

Объект bus_type содержит список всех устройств и драйверов, "подключенных" к ней. Основная задача — установить однозначное соответствие подключенного устройства и драйвера. При регистрации шины в системе каталог с ее названием появляется в sysfs, например /sys/bus/flash_bus.

Объект device_driver — абстракция драйвера устройства. Драйвер регистрируется в объекте bus_type. Его задача — выполнять необходимые действия при подключении и отключении устройства. После подключения к шине создаются каталоги, подобные /sys/bus/drivers/flash_driver, /sys/drivers/flash_driver.

Объект device характеризует подключаемое устройство. После подключения создаются каталоги /sys/bus/devices/flash_device, /sys/devices/flash_device.

Объекты device_attribute, driver_attribute, bus_attribute обеспечивают взаимодействие ядра и пользовательского уровня. Каждый атрибут представляет собой файл в sysfs. При помощи атрибутов пользователь может настраивать параметры устройства и драйвера.

Каталог /tmp — временные файлы

Каталог должен быть доступен для программ, которые нуждаются во временных файлах. При загрузке системы файлы, находящиеся в /tmp, должны удаляться (по крайней мере, это рекомендуется).

Каталог /usr — иерархия

Каталог /usr — это вторая основная секция файловой системы, разделяемая, только для чтения. Здесь должны находиться следующие каталоги:

- /bin — содержит бóльшую часть утилит, используемых пользователем;
- /include — файлы заголовков, включаемых в C-программы;
- /lib — библиотеки;
- /local — локальная иерархия;
- /sbin — содержит не жизненно необходимые системные исполняемые файлы;
- /share — архитектурно-независимые данные;
- /X11R6 — X Window System, версия 11, выпуск 6;
- /games — игры и образовательные программы;
- /src — исходные коды.

/usr/bin — пользовательские программы

В каталоге содержится большинство программ, предназначенных для пользователей. В частности здесь должны находиться следующие программы (если установлены соответствующие пакеты):

- perl — интерпретатор языка Perl;
- python — интерпретатор языка Python;
- tclsh — интерпретатор Tcl;
- wish — простая оконная оболочка Tcl/Tk;
- expect — программа для интерактивного диалога.

/usr/include — каталог для стандартных include-файлов

В этом каталоге хранится большинство включаемых файлов для компилятора C/C++.

/usr/lib — библиотеки для программирования и пакетов

Каталог содержит объектные файлы, библиотеки и другие файлы, которые не используются напрямую пользователем или скриптами командных оболочек. Если программа создает подкаталог в /usr/lib, все архитектурно-зависимые данные должны помещаться в этот каталог. Например, подкаталог /perl5 содержит в себе модули и библиотеки для Perl 5.

/usr/local — локальная иерархия

Каталог предназначен для системного администратора под установку локального ПО. Это необходимо для предотвращения перезаписи ПО при обновлении системы. Содержит следующие каталоги:

- /bin — локальные исполняемые файлы;
- /games — локальные исполняемые файлы игр;
- /include — локальные файлы C-заголовков;
- /lib — локальные библиотеки;
- /sbin — локальные системные исполняемые файлы;
- /share — локальная архитектурно-независимая иерархия;
- /src — локальный исходный код.

/usr/sbin — не жизненно необходимые стандартные системные программы

Каталог содержит любые не жизненно необходимые для функционирования системы исполняемые файлы, предназначенные исключительно для системного администратора. Программы и утилиты, применяемые при восстановлении работоспособности системы, должны находиться в каталоге /sbin.

/usr/share — архитектурно-независимые данные

Каталог предназначен для всех архитектурно-независимых файлов данных только для чтения (неизменяемых). Содержит следующие каталоги:

- /dict — списки слов (словари);
- /doc — разнообразная документация;
- /games — неизменяемые файлы данных для /usr/games;
- /info — основной каталог информационной системы GNU;
- /locale — информация для локализации системы;
- /man — файлы справочной системы;
- /misc — разнообразные архитектурно-независимые данные;
- /terminfo — каталог для базы данных terminfo;
- /zoneinfo — информация и конфигурация временной зоны (Timezone).

Любая программа или пакет, который содержит или требует данных, не нуждающихся в модификации, должны храниться в /usr/share (или /usr/local/share, если программное обеспечение установлено локально).

/usr/share/dict — списки слов (словари)

Каталог содержит словари, находящиеся в системе. Традиционно здесь находится только файл с английскими словами, которые используются программой look и многими программами проверки правописания. В этот каталог можно установить свои файлы, например с русскими словами.

/usr/share/man — страницы справочной системы

Каталог предназначен для хранения данных справочной системы. Вся справочная информация разделена на восемь больших тем, для каждой существует свой отдельный каталог — от /man1 до /man8. Содержит следующие каталоги:

- /man1 — справочные страницы, описывающие доступные пользователям программы;
- /man2 — раздел, описывающий все системные вызовы (для взаимодействия с ядром);
- /man3 — библиотечные функции и подпрограммы. Описывает программные библиотеки, напрямую не взаимодействующие с ядром операционной системы. Этот и второй разделы справочной системы представляют интерес только для программистов;

- /man4 — описывает специальные файлы, осуществляющие функции драйверов и сетевой поддержки в системе. В основном эти файлы находятся в каталоге /dev;
- /man5 — документация по множеству файловых форматов;
- /man6 — документация по разнообразным играм;
- /man7 — разное. Содержит документацию, которую трудно классифицировать;
- /man8 — системное администрирование. Программы для администрирования и сопровождения системы.

Система справочной информации должна поддерживать несколько языков одновременно, поэтому для исключения конфликтов в каталоге /usr/share/man файлы справочной системы принято хранить следующим образом:

- для каждого языка, установленного в системе (locale, локаль), в каталоге /usr/share/man создается подкаталог, носящий имя своей локали;
- в этом подкаталоге создаются каталоги /man<раздел>, причем только те, в которых есть справочная информация;
- в каталоге /man<раздел> хранятся справочные файлы, отдельные для каждой установленной программы, причем стандартом де-факто является то, что справочные файлы заархивированы (никто, однако, не запрещает хранить их в распакованном виде, но для экономии места на жестком диске их упаковывают).

При обращении к man для получения справочной информации по какой-то программе сначала делается попытка найти справочную информацию на языке, соответствующем текущей локали. Если это не удается, то берется информация, хранящаяся в /usr/share/man/man<раздел>. По умолчанию в этих каталогах содержится англоязычная справочная информация.

Наименование языковых подкаталогов в /usr/share/man основывается на приложении E стандарта POSIX 1003.1, описывающем строку-идентификатор локали в виде

```
<язык>[_<территория>][.<кодовая страница символов>][,<версия>]
```

- `<язык>` — берется из стандарта ISO 639. Это должны быть два символа исключительно в нижнем регистре;
- `<территория>` — двухсимвольный код только в верхнем регистре (согласно стандарту ISO 3166);
- `<кодовая страница символов>` — стандартное описание кодовой страницы. Если это поле содержит числовую спецификацию, она соответствует интернациональному стандарту, описывающему эту страницу;
- `<версия>` — рекомендуется не использовать без крайней необходимости. Требуется, например, для страны, имеющей один язык и кодировку, но разные диалекты.

Пример формирования каталогов локализованной справочной системы приведен в табл. 6.3.

Таблица 6.3. *Пример формирования каталогов локализованной справочной системы*

Язык	Территория	Кодовая страница символов	Каталог
Английский	—	ASCII	/usr/share/man/en
Английский	Великобритания	ASCII	/usr/share/man/en_GB
Английский	США	ASCII	/usr/share/man/en_US
Французский	Франция	ISO 8859-1	/usr/share/man/fr_FR
Французский	Канада	ISO 8859-1	/usr/share/man/fr_CA
Русский	СНГ	KOI8-R	/usr/share/man/ru_RU

Архитектурно-зависимые справочные файлы можно помещать в отдельные каталоги, соответствующие архитектуре. Например, /usr/share/man/<locale>/man8/i386/ctrlaltdel.8. Однако проще написать общее справочное руководство, в котором особо отметить архитектурно-зависимые случаи, чем разрабатывать справочные файлы для каждой архитектуры.

Справочная информация для программ и данных, находящихся в /usr/local, размещается в каталоге /usr/local/man. Справочная информация, касающаяся X11R6, находится в каталоге /usr/X11R6/man.

Правило размещения справочных руководств на различных языках в отдельные подкаталоги также распространяется на каталоги /usr/local/man и /usr/X11R6/man.

/usr/share/misc — различные архитектурно-независимые данные

Каталог содержит различные архитектурно-независимые файлы, которые не требуют отдельного каталога в /usr/share/. Если соответствующие пакеты установлены в системе, то здесь должны находиться следующие файлы:

- ascii — ASCII-таблица символов;
- magic — список "магических" цифр;
- termcap — база данных совместимости терминалов.

/usr/src — исходные тексты программ

Любой исходный код нелокальной программы должен помещаться в этот каталог.

/usr/src/Linux-x.y.z — каталог исходного кода ядра Linux

Здесь хранятся файлы и каталоги, содержащие исходный код ядра Linux, модулей, различная документация. Имя каталога меняется в зависимости от версии ядра Linux, исходный код которого находится в каталоге.

/usr/src/Linux-x.y.z/Documentation — документация к ядру и модулям операционной системы Linux

В каталоге содержится документация, которая тем или иным образом касается ядра операционной системы Linux или загружаемых модулей. Типичное содержимое каталога:

/arm	/networking	cachetlb.txt	floppy.txt
/cdrom	/parisc	cciss.txt	ftape.txt
/cris	/powerpc	Changes	hayes-esp.txt
/DocBook	/s390	CodingStyle	highuid.txt
/fb	/sound	computone.txt	ide.txt
/filesystems	/sparc/sysctl	Configure.help	initrd.txt
/i2c	/telephony	cpqarray.txt	ioctl-number.txt
/i386	/video4linux	devices.txt	IO-mapping.txt
/ia64	/vm	digiboard.txt	IRQ-affinity.txt
/isdn	/usb	digiepca.txt	isapnp.txt
/kbuild	00-INDEX	DMA-mapping.txt	java.txt
/m68k	binfmt_misc.txt	dnotify.txt	joystick-api.txt
/mips	BUG-HUNTING	exception.txt	joystick-parport.txt
joystick.txt		nbd.txt	serial-console.txt
kernel-doc-nano-HOWTO.txt		nfsroot.txt	sgi-visws.txt
kernel-docs.txt		nmi_watchdog.txt	smart-config.txt
kernel-parameters.txt		oops-tracing.txt	smp.tex
kmod.txt		paride.txt	smp.txt
locks.txt		parport-lowlevel.txt	specialix.txt
logo.gif		parport.txt	spinlocks.txt
logo.txt		pci.txt	stallion.txt
LVM-HOWTO		pcwd-watchdog.txt	SubmittingDrivers
magic-number.txt		pm.txt	SubmittingPatches
mandatory.txt		ramdisk.txt	svga.txt
mca.txt		README.DAC960	sx.txt
md.txt		README.moxa	sysrq.txt
memory.txt		README.nsp_cs.eng	unicode.txt
mkdev.cciss		riscom8.txt	VGA-softcursor.txt
mkdev.ida		rtc.txt	watchdog.txt
modules.txt		SAK.txt	xterm-linux.xpm
moxa-smartio		scsi-generic.txt	zorro.txt
mtrr.txt		scsi.txt	

Каталог /var

Каталог содержит изменяемые файлы. Сюда входят spool-каталоги и файлы, административные и журнальные данные, временные файлы. Некоторые каталоги, входящие в иерархию /var, такие как /var/log, /var/lock и /var/run, не должны быть разделяемыми между различными системами. Другие каталоги, такие как /var/mail, /var/cache/man, /var/cache/fonts и /var/spool/news, могут быть разделяемыми.

Рекомендуется для каталога /var выделить отдельный раздел на жестком диске. В том случае, если это невозможно, не следует размещать его в корневой файловой системе. Это позволит избежать некоторых проблем, возникающих при переполнении диска. Приложения не должны создавать каталоги в верхнем уровне иерархии /var. В каталоге /var должны присутствовать следующие каталоги:

- /cache — каталог кэша программ;
- /db — каталог для файлов баз данных;
- /games — файлы для игровых программ;
- /lib — библиотеки;
- /local — изменяемые данные для /usr/local;
- /lock — Lock-файлы (файлы-защелки);
- /log — Log-файлы и каталоги (файлы журналов);
- /lost+found — каталог для файлов, восстановленных после краха системы;
- /mail — почтовые ящики пользователей;
- /named — файлы DNS-сервера;
- /opt — переменные данные для /opt;
- /run — данные о запущенных процессах;
- /spool — spool-данные приложений;
- /state — состояние приложений;
- /tmp — временные файлы, сохраняемые между перезагрузками системы.

/var/cache — кэш программ

Каталог служит для хранения временных "короткоживущих" данных, создаваемых программами. Это могут быть буферы ввода/вывода или файлы, содержащие какие-нибудь промежуточные данные. Подкаталоги в /var/cache создаются при установке пакетов и обычно носят имя соответствующей программы.

Если соответствующие пакеты установлены в системе, в каталоге должны находиться следующие файлы:

- /fonts — динамически создаваемые шрифты;
- /man — сформатированные страницы руководств. Справочные страницы в /usr/man хранятся в специальном виде, и перед тем, как показать справочное руководство пользователю, страницы необходимо сформатировать;
- /www — файлы или кэш-данные proxy-сервера WWW;
- /<пакет> — кэш соответствующего пакета.

/var/games — файлы для игровых программ

В этом каталоге должны храниться файлы, которые могут изменяться, например файлы, содержащие таблицы результатов, файлы сохраненных игр и т. п.

/var/lib — библиотеки

Немного неверное наименование раздела. В этом каталоге содержатся различные файлы, входящие в какие-либо пакеты, которые можно отнести к системным. Обычно каждый пакет, который сохраняет какие-то файлы в каталог /var/lib, созда-

ет свой каталог, имеющий вид /var/lib<имя_пакета>. Если соответствующие пакеты установлены в системе, то здесь должны находиться следующие файлы:

- /misc — разные несистематизированные файлы;
- /<редактор> — каталог соответствующего редактора, в котором хранятся резервные копии файлов и файлы состояния;
- /rpm — каталог для менеджера пакетов RPM. В нем содержатся базы установленных в системе пакетов и другая служебная информация;
- /<пакет> — файлы соответствующего пакета;
- /xdm — данные X-менеджера.

/var/lock — lock-файлы (файлы-защелки)

Lock-файлы (файлы-защелки) — это файлы, которые "закрепляют" какое-либо оборудование или файлы для использования только программой, создающей файл-защелку. Обычно уничтожаются по окончании работы программы, а также если файл или оборудование не нужны в данный момент программе. В каталоге /var/lock могут находиться, например, следующие подкаталоги:

- /console — данные, относящиеся к консоли системы;
- /samba — данные, связанные с программой Samba.

/var/log — файлы и каталоги журналов (log-файлов)

Каталог содержит разнообразные файлы журналов. Для некоторых пакетов тоже предусмотрены каталоги, в которых хранятся соответствующие файлы журналов. Если соответствующие пакеты установлены в системе, в каталоге должны присутствовать следующие файлы:

- /httpd — каталог для журнальных файлов Web-сервера;
- /samba — каталог для журнальных файлов сервера Samba;
- /squid — каталог для журнальных файлов SQUID;
- /uucp — каталог для журнальных файлов UUCP.

Также в каталоге /var/log должны находиться следующие файлы:

- cron — события демона cron;
- dmesg — сообщения в течение дня;
- lastlog — записи о последней регистрации в системе каждого пользователя;
- maillog — регистрация событий, связанных с почтовыми сообщениями;
- messages — системные сообщения от syslogd;
- secure — сообщения, связанные с безопасностью;
- statistics — файл статистики;
- usracct — файл активности пользователей;
- wtmp — записи всех logins и logouts;
- boot.log — журнал загрузки системы;
- htmlaccess.log — журнал доступа к Web-серверу;
- XFree86.0.log — журнал XFree86.

/var/mail — пользовательские почтовые ящики

Этот каталог хранит пользовательские почтовые ящики, сохраненные в стандартном формате UNIX mailbox.

/var/opt — изменяемые данные для каталога /opt

Здесь должны храниться изменяемые данные пакетов, устанавливаемые в каталог /opt. Рекомендуется для каждого пакета создать свой каталог вида /opt/<имя_пакета>.

/var/run — переменные файлы времени исполнения

Каталог содержит системную информацию, описывающую состояние системы. Файлы в этом каталоге при загрузке системы должны быть удалены или усечены до нулевого размера. Программы при необходимости могут иметь подкаталоги, если во время функционирования создается более чем один файл (однако, например, демон FTP создает следующие файлы: ftp.pids-all, ftp.pids-local, ftp.pids-other, а отдельного каталога не имеет).

Здесь, в основном, содержатся файлы-идентификаторы процессов (PID, Process identifie file), имеющие имя <имя_программы>.pid, например, /var/run/named.pid. Pid-файл должен содержать символы, соответствующие номеру PID, и символ перевода строки.

Каталог /var/run должен быть недоступен для непривилегированных пользователей, поскольку запись информации или ее удаление из каталога /var/run может привести к печальным последствиям, вплоть до краха системы.

/var/spool — spool-данные приложений

Каталог /var/spool содержит данные, которые ожидают какой-либо обработки. После обработки (программой, пользователем, администратором) они должны быть удалены из каталога. Если соответствующие пакеты установлены в системе, то здесь должны находиться следующие файлы:

- /at — spool-каталог программы at;
- /cron — spool-каталог программы cron;
- /lpd — spool-каталог программы печати;
- /mail — каталог входящей почты;
- /mqueue — исходящая почтовая очередь;
- /news — spool-каталог сервера новостей;
- /samba — spool-каталог сервера Samba;
- /squid — spool-каталог SQUID;
- /uucp — spool-каталог для UUCP.

/var/tmp — временные файлы, сохраняемые между перезагрузками

Каталог /var/tmp используется для того, чтобы временные файлы, необходимые для программ, сохранялись при перезагрузке системы. Файлы, находящиеся в /tmp, при перезагрузке системы могут быть удалены.

/var/yp — файлы баз данных Network Information Service (NIS) (опционально)

Если в системе установлена сетевая информационная служба (Network Information Service, NIS), так же известная, как "Желтые страницы" (Sun Yellow Pages, YP), то в этом каталоге хранятся ее базы данных.

Ссылки

- **http://www.pathname.com/fhs/** — Filesystem Hierarchy Standard в различных текстовых форматах.
- **http://www.kernel.org/pub/linux/docs/device-list/devices.txt** — список устройств и специальных файлов.
- proc.txt — документация по файловой системе procfs. Входит в состав документации к ядру Linux.
- Соответствующие man-страницы.
- Соответствующие HOWTO:
  - Networking-HOWTO;
  - SMB-HOWTO;
  - DNS-HOWTO;
  - LILO-HOWTO.

Глава 7



Процесс загрузки Linux

Для того чтобы достичь полного контроля над операционной системой, крайне важно представлять себе, как происходит ее загрузка. При включении компьютера специальная программа, записанная в ПЗУ материнской платы, тестирует установленное в компьютере оборудование. В случае неудачи вы либо услышите из встроенного динамика компьютера серию гудков, либо программа тестирования оборудования выведет на дисплей предупреждающее сообщение.

Если система успешно прошла тестирование, на дисплее можно будет увидеть перечень установленного оборудования, емкость оперативной памяти и жесткого диска. После этого программа BIOS (Basic Input/Output System — базовая система ввода/вывода), хранящаяся в ПЗУ материнской платы, определит, с какого устройства будет происходить загрузка (например, с жесткого диска C:), и считает из первого сектора загрузочного диска короткую программу — *загрузчик*. Эта программа (GRUB) загружает с жесткого диска ядро Linux, которое имеет имя `vmlinuz-x.y.z-a` (где `x.y.z` — это номер версии ядра, например, 2.6.19, строка `a` — признак сборки, может быть каким-то числом или словом) и находится в каталоге /boot (для Red Hat-подобных дистрибутивов) или в корне файловой системы (для дистрибутива Slackware). Во время загрузки ядру можно передать различные параметры, позволяющие более тонко настроить систему (об этом немного позже). Сразу после загрузки ядро инициализирует устройства, установленные в компьютере, а затем пробует загрузить и смонтировать корневую (root) файловую систему. Ядру необходимо тем или иным образом сообщить, где искать корневую файловую систему. Если ядро Linux не может ее найти, оно выдает соответствующее сообщение и останавливается.

Во многих дистрибутивах и практически всегда при загрузке с дискеты в оперативной памяти создается псевдодиск (RAM-disk, виртуальный диск), который выступает в роли корневой файловой системы. На это есть две причины. Во-первых, оперативная память на несколько порядков быстрее, чем дискета, во-вторых, на виртуальный диск ядро может загрузить с дискеты и распаковать сжатую файловую систему, что позволяет поместить на дискете намного больше файлов.

После того как ядро Linux успешно смонтирует корневую файловую систему, оно запускает процесс init — программу, которая, собственно, и осуществляет переход от начального состояния системы в стандартный многопользовательский режим (или тот, который установлен администратором по умолчанию). Кроме того,

процесс init выполняет множество различных операций, необходимых для корректной работы системы: проверку и монтирование файловых систем, запуск различных сервисов, запуск системы входа пользователя и т. п. А теперь подробнее разберемся с каждым шагом загрузки системы.

Программы-загрузчики

Загружают ядро операционной системы Linux, передают ему параметры и при необходимости организуют загрузку нескольких операционных систем, установленных на компьютере.

LILO — LInux LOader

Программа-загрузчик, которая на сегодняшний день устарела и практически не используется. LILO (LInux Loader — загрузчик Linux) без проблем может загружать DOS, OS/2, Linux, FreeBSD, Windows и множество других операционных систем.

Стандартно сконфигурированная программа LILO после запуска приостановит свое выполнение и выведет на экран графическое изображение с меню, пунктами которого являются варианты загрузки. Несколько секунд LILO ожидает ввода пользователем варианта загрузки (или специальных команд) и при их отсутствии запускает вариант, заданный по умолчанию. Обычные варианты загрузки в LILO носят название linux и dos (если в системе установлены одновременно операционная система Windows 9*x* и Linux).

Конфигурационный файл LILO — /etc/lilo.conf, формат его можно найти в соответствующей справочной документации.

GRUB

GRand Unified Bootloader (Главный унифицированный загрузчик) — универсальный загрузчик, разработан в Фонде свободного программного обеспечения. По сравнению с LILO имеет больше возможностей, а также избавлен от некоторых ограничений.

LoadLin

Еще одна программа запуска Linux, которая требуется не при старте компьютера, а для загрузки ядра Linux из командной строки DOS (с параметрами загрузки). Применение LoadLin оправдано в том случае, если мы не хотим устанавливать загрузчик типа LILO в MBR (Master Boot Record) винчестера. Этот загрузчик также необходим, если у нас имеется оборудование, использующее драйвер DOS для установки в определенное состояние.

Есть еще несколько других программ, пригодных для загрузки Linux, однако они не получили широкого распространения.

Параметры ядра

Обзор параметров строки загрузки

Описанные программы-загрузчики способны также, помимо загрузки самого ядра, передавать ему необходимые параметры.

В параметрах загрузки пробелы допустимы только между отдельными аргументами. Значения в списке для одного аргумента разделяют запятыми:

```
ether=9,0x300,0xd0000,0xd4000,eth0  root=/dev/hda1
```

Посмотреть параметры командной строки, заданные при загрузке, можно, набрав `/proc/cmdline`.

Утилита rdev

Есть несколько параметров загрузки ядра Linux, хранящих свои значения по умолчанию в его образе. Эти параметры задаются при компиляции ядра, и для того, чтобы не перекомпилировать каждый раз ядро, применяется утилита rdev.

Утилита rdev может изменять следующие параметры:

- rdev — устройство, с которого производится загрузка;
- swapdev — устройство, содержащее раздел подкачки (swap);
- ramsize — параметры RAM-диска;
- vidmode — видеорежим по умолчанию;
- rootflags — установка режима монтирования корневого устройства ("только для чтения" или "чтение/запись").

Более подробную информацию по rdev можно найти в соответствующей справочной документации.

Разбор параметров ядром Linux

Многие параметры загрузки имеют вид:

```
имя[=значение_1][,значение_2]...[,значение_11],
```

где `имя` — уникальное ключевое слово, идентифицирующее часть ядра, которому передаются связанные значения, но не более одиннадцати параметров. Бóльшая часть разбора параметров загрузки происходит в linux/init/main.c. Сначала ядро проверяет, не являются ли параметры одним из специальных параметров `root=`, `ro`, `rw` или `debug`, затем просматривает список функций установки (находящийся в массиве bootsetups) в поиске совпадения заданной строки параметра с функцией установки конкретного устройства или части ядра. Если мы передаем ядру строку `foo=3,4,5,6,bar`, то оно будет искать, присутствует ли `foo` в массиве bootsetups. Если присутствует, то ядро вызовет функцию установки, связанную с `foo` (`foo_setup()`), и передаст ей целочисленные значения `3`, `4`, `5` и `6`, указанные в командной строке ядра, и также строковый параметр `bar`.

Если строка не подходит ни для одной функции установки, то этот случай считается установкой переменной окружения. Примером может служить указание пе-

ременных окружения `TERM=vt100` или `BOOT_IMAGE=vmlinuz.bak` в качестве параметров загрузки. Как правило, переменные окружения проверяются скриптами инициализации для разрешения или запрещения большого диапазона параметров.

Любые оставшиеся параметры, не выбранные ядром и не интерпретированные в качестве переменных окружения, будут переданы в дальнейшую обработку, которую обычно выполняет программа init. Чаще всего процессу init в качестве параметра передается слово single, которое сообщает init о необходимости загрузить компьютер в однопользовательском режиме. Список параметров программы init можно найти в соответствующей справочной документации.

Общие неаппаратные параметры загрузки

В этом разделе рассматриваются параметры загрузки, связанные не с каким-либо оборудованием или периферией, а с параметрами ядра, такими как управление памятью, RAM-диском, корневой системой и т. п.

Опции корневой файловой системы

Параметр *root*

Этот параметр сообщает ядру, какое устройство будет использовано в качестве корневой файловой системы во время загрузки. По умолчанию эта установка имеет значение корневого устройства системы, на котором было скомпилировано ядро. Например, на одном компьютере корневая файловая система находится на /dev/hda2, а на другом — на /dev/hda6. Если скомпилировать ядро на втором компьютере, перенести его на первый и не указать в параметре `root=/dev/hda2`, то ядро будет думать, что оно загружается с /dev/hda6. А такого устройства на этом компьютере нет!

Перечислим допустимые корневые устройства:

- /dev/hdaN, /dev/hdbN, /dev/hdcN, /dev/hddN, которые являются разделами N на IDE-диске;
- /dev/sdaN, /dev/sdbN, /dev/sdcN, /dev/sddN, /dev/sdeN, которые являются разделами N на SCSI-диске;
- /dev/fd0, /dev/fd1 — привод флоппи-диска с номером N;
- /dev/nfs, не являющееся флагом, заставляющим ядро получить корневую файловую систему по сети.

`root` — один из немногих параметров загрузки ядра, которые хранятся в его образе и могут быть изменены утилитой rdev.

Параметры *ro* и *rw*

Параметр `ro` сообщает ядру о необходимости монтирования корневой файловой системы в режиме "только для чтения". Парный ему параметр `rw` указывает ядру монтировать корневую файловую систему в режиме "чтение/запись". Сразу после загрузки ядра и запуска процесса init система должна проверить подмонтированные файловые системы на отсутствие ошибок. Однако если корневая файловая

система смонтирована в режиме "чтение/запись", надежно проверить целостность файловой системы невозможно. Существуют два способа решения этой проблемы:

- изначально смонтировать корневую файловую систему в режиме "только для чтения";
- изначально смонтировать корневую файловую систему в режиме "чтение/запись", а перед проверкой перемонтировать ее в режим "только для чтения".

Это одни из немногих параметров загрузки ядра, которые хранят значение в образе ядра и могут быть изменены утилитой rdev.

Опции управления RAM-диском

Все следующие опции сообщают ядру, как управлять устройством RAM-диска, обычно используемым для загрузки машины.

Параметр *ramdisk_start*

Чтобы разрешить образу ядра находиться на флоппи-диске со сжатым образом RAM-диска, необходимо добавить команду `ramdisk_start=<смещение>`.

Параметр *load_ramdisk*

Этот параметр сообщает ядру, нужно загружать образ RAM-диска или нет. При `load_ramdisk=1` ядро будет загружать RAM-диск. По умолчанию значение параметра равно нулю, т. е. ядро не должно загружать RAM-диск.

Параметр *prompt_ramdisk*

Этот параметр сообщает ядру о необходимости вывести пользователю приглашение вставить флоппи-диск с образом RAM-диска. В однодисковой конфигурации образ RAM-диска находится на той же дискете, с которой только что закончилась загрузка ядра, поэтому приглашение не нужно. В этом случае необходима команда `prompt_ramdisk=0`. В двухдисковой конфигурации может потребоваться заменить диски, поэтому следует указать команду `prompt_ramdisk=1`. По умолчанию значение равно единице.

Параметр *ramdisk_size*

Поскольку RAM-диск размещается в оперативной памяти, необходимо каким-то способом указать занимаемый им объем оперативной памяти. По умолчанию это 4096 Кбайт.

Параметр *noinitrd* (начальный RAM-диск)

В ядрах, начиная с версии 2.*x*, корневой файловой системой изначально может быть RAM-диск. Эта возможность обычно требуется для загрузки модулей, необходимых для монтирования реальной корневой файловой системы (например, загрузка модулей драйвера SCSI, хранящихся в образе RAM-диска, а затем монтирование реальной файловой системы на SCSI-диске).

Параметр `noinitrd` определяет, что будет происходить с данными initrd после загрузки ядра.

Параметры загрузки для управления памятью

Следующие параметры определяют действия ядра Linux по обнаружению или управлению физической и виртуальной памятью системы.

Параметр *mem*

Служит для указания объема установленной памяти (или меньшего значения, если требуется ограничить объем памяти, доступный Linux). Старые версии BIOS не могли корректно возвращать объем оперативной памяти, если он превышал 64 Мбайт. Поэтому приходилось вручную передавать в ядро реально установленную емкость памяти.

Ядро воспримет любое значение параметра `mem=xx`, которое будет указано, однако если задать больший размер памяти, чем физически установлено в компьютере, то при определенном количестве процессов система попытается обратиться к несуществующему участку памяти и тогда могут возникнуть проблемы, и что самое неприятное, такое может произойти и через месяц-другой после конфигурирования системы.

Объем памяти можно указывать как в шестнадцатеричном представлении, так и в десятичном. Например, если в компьютере установлено 96 Мбайт оперативной памяти, можно указать `mem=0x6000000` или `mem=96M`.

Параметр *swap*

Позволяет пользователю настраивать некоторые параметры виртуальной памяти (Virtual Memory), относящиеся к разделу подкачки. Возможные значения:

- MAX_PAGE_AGE;
- PAGE_ADVANCE;
- PAGE_DECLINE;
- PAGE_INITIAL_AGE;
- AGE_CLUSTER_FRACT;
- AGE_CLUSTER_MIN;
- PAGEOUT_WEIGHT;
- BUFFEROUT_WEIGHT.

В каталоге /usr/src/Linux-x.y.z/Documentation/vm/ содержится полезная документация по этой теме, поставляемая с ядром операционной системы.

Параметр *buff*

Параметр, похожий на `swap`, позволяет пользователю настроить некоторые опции, связанные с управлением буферной памятью. Возможные значения:

- MAX_BUFF_AGE;
- BUFF_ADVANCE;
- BUFF_DECLINE;
- BUFF_INITIAL_AGE;
- BUFFEROUT_WEIGHT;
- BUFFERMEM_GRACE.

Параметры загрузки для файловой системы NFS

Linux поддерживает и бездисковые рабочие станции, загружаемые по локальной сети. Для этого необходимо настроить корневую файловую систему бездисковой станции как NFS (Network File System, сетевая файловая система). Сообщить бездисковой рабочей станции, с какой машины она будет получать операционную систему, можно с помощью параметров, указанных в этом разделе. Также необходимо установить параметр `root=/dev/nfs`. Подробная информация по использованию NFS в качестве корневой файловой системы содержится в файле /usr/src/Linux-x.y.z/Documentation/nfsroot.txt.

Параметр *nfsroot*

Параметр сообщает ядру, какую машину, какой каталог и с какими опциями NFS выбрать в качестве корневой файловой системы. Формат этого параметра следующий:

```
nfsroot=[<server-ip>:]<root-dir>[,<nfs-options>]
```

Если параметр `nfsroot` не был задан в командной строке, то значением по умолчанию будет `/tftpboot/%s`.

Другие опции:

- `<server-ip>` — задает IP-адрес сервера NFS. Если это поле не задано, по умолчанию адрес будет определен переменной `nfsaddrs`;
- `<root-dir>` — имя каталога на сервере, монтируемого как корневой. Если в строке имеется фраза `"%s"`, она будет заменена на ASCII-представление IP-адреса клиента;
- `<nfs-options>` — стандартные опции NFS. Все опции разделены запятыми. Если поле опций не задано, будут назначены следующие параметры:
  - `port =` указывается демоном portmap-сервера;
  - `rsize = 1024`;
  - `wsize = 1024`;
  - `timeo = 7`;
  - `retrans = 3`;
  - `acregmin = 3`;
  - `acregmax = 60`;
  - `acdirmin = 30`;
  - `acdirmax = 60`;
  - `flags = hard, nointr, noposix, cto, ac`.

Параметр *nfsaddrs*

`nfsaddrs` устанавливает параметры сетевого интерфейса. Если параметр опущен, то для выяснения этих значений ядро попытается использовать RARP и/или BOOTP. Формат параметра:

```
nfsaddrs=<my-ip>:<serv-ip>:<gw-ip>:<netmask>:<name>:<dev>:<auto>
```

Здесь:

- `<my-ip>` — IP-адрес клиента. Если параметр опущен, адрес определяется с помощью RARP или BOOTP. Выбор протокола будет зависеть от того, как было сконфигурировано ядро, и от параметра `<auto>`. Если параметр указан, ни RARP, ни BOOTP задействованы не будут;
- `<serv-ip>` — IP-адрес сервера NFS. Если это поле опущено, будет выбран адрес сервера, ответившего на запрос RARP или BOOTP;
- `<gw-ip>` — IP-адрес шлюза. Если поле опущено, шлюзы будут проигнорированы;
- `<netmask>` — маска сети для сетевого интерфейса;
- `<name>` — имя клиента;
- `<dev>` — имя применяемого сетевого устройства. Если поле опущено, для RARP-запросов будут использованы все устройства, а для BOOTP — первое найденное. Для NFS будет выбрано устройство, на котором были получены ответы RARP или BOOTP;
- `<auto>` — автоконфигурирование. Возможны следующие значения:
  - `rarp` — протокол RARP;
  - `bootp` — протокол BOOTP;
  - `both` — будут применены оба протокола;
  - `none` — отсутствие автоконфигурирования. В этом случае следует указать все необходимые значения в предыдущих полях.

Дополнительные параметры загрузки

Эти параметры начальной загрузки позволяют пользователю настраивать некоторые внутренние параметры ядра.

Параметр *debug*

Ядро Linux имеет возможность выводить важные сообщения на консоль (ошибки ввода/вывода, проблемы с оборудованием и т. п.). Пороговое значение важности сообщения задается переменной `console_loglevel`. По умолчанию на консоль отправляется практически все, кроме отладочной информации. Указание параметра `debug` позволит всем сообщениям ядра попадать на консоль.

Параметр *init*

Во время загрузки ядро Linux запускает программу init, которая затем подготавливает операционную систему для полноценной работы. Сначала ядро Linux ищет программу init в каталоге /sbin, а при неудаче пробует запустить ее из каталога /bin/sh. Если программа init повреждена, и загрузить операционную систему не удается, можно ввести командную строку загрузки init=/bin/sh, которая даст возможность заменить поврежденную программу или выполнить какие-то другие программы.

Параметр *kbd-reset*

Обычно на компьютерах семейства *x*86 ядро Linux не сбрасывает при загрузке контроллер клавиатуры, предполагая, что это делает BIOS. Однако такое предположение не всегда соответствует действительности. Применение этой опции заставляет во время загрузки Linux выполнять сброс контроллера клавиатуры.

Параметр *maxcpus*

Параметр ограничивает максимальное количество процессоров в режиме SMP. Указание в параметре `0` эквивалентно опции `nosmp`.

Параметр *md*

Если корневая система компьютера расположена на составном (Multiple) устройстве (как правило, это RAID-массив дисков), то сообщить ядру конфигурацию составного устройства можно через параметр `md`. Подробная информация по этой теме содержится в файле /usr/src/Linux-x.y.z/Documentation/md.txt.

Параметр *no-hlt*

Параметр актуален только для старых компьютеров на базе процессора i486. У процессоров Intel есть инструкция `hlt`, заставляющая процессор ничего не делать, пока внешнее устройство (клавиатура, винчестер и т. п.) не вызовет его для выполнения задачи. Некоторые чипы i486 имели проблемы с командой `hlt`, после которой они не могли вернуться в рабочий режим. С помощью параметра `no-hlt` можно заставить ядро Linux при отсутствии активности вместо остановки процессора выполнять бесконечный цикл.

Параметр *no-scroll*

Параметр запрещает при загрузке функцию прокрутки. Актуально только для некоторых устаревших терминалов.

Параметр *noapic*

Параметр позволяет ядру Linux с поддержкой мультипроцессорности не использовать расширенные возможности контроллера прерываний в многопроцессорных машинах. Подробную информацию можно найти в файле /usr/src/Linux-x.y.z/Documentation/IO-APIC.txt.

Параметр *nosmp*

Позволяет ядру Linux с поддержкой мультипроцессорности на SMP-машинах работать только с одним процессором. Обычно служит для отладки.

Параметр *panic*

В крайне редком случае "паники" ядра (обнаруженная внутренняя ошибка, которую ядро считает достаточно серьезной, что приводит к выдаче сообщения `kernel panic` и полной остановке системы) по умолчанию компьютер остается в этом состоянии, пока администратор его не перезагрузит. Однако иногда для восстановления нормальной работы системы необходимо, чтобы машина автоматически перезагрузила себя. Данный параметр позволяет установить время (в секундах), по прошествии которого система попытается перезагрузиться. Например, при установке параметра `panic=20` ядро Linux попытается перезагрузиться через 20 с после выдачи сообщения `kernel panic`. Нулевое значение соответствует стандартному поведению — ждать вмешательства администратора.

Время тайм-аута можно прочитать и изменить через интерфейс /proc/sys/kernel/panic.

Параметр *pirq*

Эта опция передает мультипроцессорному ядру информацию об установках IRQ-слота PCI для некоторых материнских плат SMP. Подробную информацию можно найти в файле /usr/src/Linux-x.y.z/Documentation/IO-APIC.txt.

Параметр *profile*

Разработчики ядер могут разрешать опции, позволяющие им для оптимизации быстродействия ядра определять, как и где ядро может использовать циклы процессора. Эта опция позволяет указать номер конфигурационного файла при загрузке. Можно также скомпилировать ядро с конфигурацией, разрешенной по умолчанию.

Параметр *reboot*

Параметр задает тип перезагрузки, выполняемой ядром Linux. Стандартно ядро Linux выполняет так называемую "холодную" перезагрузку (полная инициализация аппаратного обеспечения, BIOS проверяет память и т. д.). Существует также "теплая" перезагрузка, при которой не происходит первоначального тестирования оборудования, что несколько убыстряет загрузку операционной системы.

Параметр *reserve*

Служит для защиты диапазона портов ввода/вывода от тестирования (I/O probe). Формат команды:

```
reserve=iobase,extent[,iobase,extent]...
```

В некоторых машинах бывает необходимо защитить драйверы устройств от поиска устройств (auto-probing) в определенном диапазоне. Причиной могут послужить устройства, идентифицирующиеся ошибочно, или устройства, инициализация которых ядром нежелательна.

Параметр загрузки `reserve` устраняет проблему, указывая диапазон портов ввода/вывода, который необходимо исключить из тестирования. При этом диапазон резервируется в таблице ядра регистрации портов как уже определенный. Такой механизм необходим только при наличии проблем или в специальных случаях.

Загрузочные параметры, определяющие поведение шины PCI

Параметр `pci` дает возможность изменить способ поиска устройств на шине PCI и поведение этих устройств. Как правило, это необходимо либо для старого оборудования, не совсем корректно поддерживающего технологию Plug and Play, либо для специфических PCI-устройств.

Аргументы *pci=bios* и *pci=nobios*

Устанавливают или сбрасывают флаг индикации тестирования (probing) PCI через PCI BIOS. По умолчанию используется флаг bios.

Аргументы *pci=conf1* и *pci=conf2*

Разрешают тип конфигурации 1 или 2. Также они неявно сбрасывают флаг PCI BIOS probe (т. е. `pci=nobios`).

Аргумент *pci=io=XXX*

Если получено сообщение типа

```
Unassigned IO space for.../
```

то может потребоваться указать значение ввода/вывода этой опцией.

Аргумент *pci=nopeer*

Специфический аргумент, исправляющий погрешности некоторых версий BIOS.

Аргумент *pci=nosort*

Указание этого аргумента заставляет ядро не сортировать PCI-устройства в процессе проверки.

Аргумент *pci=off*

Эта опция запрещает все проверки PCI-шины. Любые драйверы устройств, задействующие функции PCI для поиска и инициализации оборудования, скорее всего, потеряют работоспособность.

Аргумент *pci=reverse*

Эта опция меняет на обратный порядок PCI-устройств на шине PCI.

Аргументы загрузки для драйверов буфера видеофреймов

Аргумент `video=` используется, когда уровень абстракции устройства буфера фреймов встроен в ядро. Это означает, что вместо отдельных программ для каждого семейства видеокарт (Intel, AMD,nVidia и др.) ядро имеет встроенный драйвер для каждой видеокарты и экспортирует единственный (единый) интерфейс для видеопрограмм. Типичный формат этого аргумента:

```
video=name:option1,option2,...
```

Здесь `name` — название универсальной опции или драйвера буфера фреймов. Как только найдено совпадающее имя драйвера, список параметров, разделенных запятыми, передается в этот конкретный драйвер для окончательной обработки.

Информацию по опциям, поддерживаемым каждым драйвером, можно найти в файле /usr/src/Linux-x.y.z/Documentation/fb/.

Аргумент *video=map:...*

Эта опция служит для установки консоли отображения устройства буфера фреймов.

Аргумент *video=scrollback:...*

Число после двоеточия устанавливает размер памяти, выделенной для буфера прокрутки. Суффикс `k` или `K` после числа указывает, что число представляет собой килобайты.

Аргумент *video=vc:...*

Число или диапазон чисел определяют первую или первую и последнюю виртуальные консоли буфера фреймов.

Аргументы загрузки для SCSI-периферии

Этот раздел содержит описание аргументов загрузки для передачи информации об установленных SCSI-контроллерах и устройствах.

Аргументы для драйверов Mid-level

Драйверы уровня Mid управляют такими устройствами, как винчестеры, CD-ROM и стримеры без учета специфики SCSI-контроллера.

Максимальный LUN (*max_scsi_luns=*)

Каждое SCSI-устройство внутри себя может иметь несколько псевдоустройств. Например, SCSI CD-ROM, обслуживающий более чем один диск одновременно. Каждый CD-ROM адресуется номером логического устройства (Logical Unit Number,

LUN). Но большинство SCSI-устройств являются одним устройством, и им назначается нулевой LUN.

Старые SCSI-устройства не могут обработать запросы поиска с LUN, не равным нулю. Зачастую это приводит к зависанию устройства. Чтобы избежать указанной проблемы, по умолчанию пробуется только нулевой LUN.

Для определения количества пробуемых LUN при загрузке в качестве аргумента загрузки вводится `max_scsi_luns=n`, где *n* — номер от 1 до 8.

Регистрация SCSI (*scsi_logging*=)

Ненулевое значение этого загрузочного аргумента включает регистрацию всех SCSI-событий.

Параметры для ленточного накопителя SCSI (*st*=)

При загрузке ядра Linux можно изменить конфигурацию ленточного накопителя SCSI, используя

```
st=buf_size[,write_threshold[,max_bufs]]
```

Первые два числа указываются в килобайтах. По умолчанию `buf_size` равен 32 Кбайт. `write_threshold` — значение, при котором буфер сбрасывается на ленту, по умолчанию 30 Кбайт. Максимальное число буферов зависит от количества обнаруженных ленточных накопителей, по умолчанию равно 2.

Аргументы для контроллеров SCSI

Понятия, рассматриваемые в данном разделе:

- `iobase` — первый порт ввода/вывода, занимаемый контроллером SCSI. Указывается в шестнадцатеричной нотации и обычно находится в диапазоне от `0x200` до `0x3ff`;
- `irq` — аппаратное прерывание, установленное на карте. Допустимые значения зависят от конкретного контроллера, но обычно это 5, 7, 9, 10, 11, 12 и 15;
- `dma` — используемый картой канал DMA (Direct Memory Access — прямой доступ к памяти). Обычно применяется только для карт с управлением шиной (bus-mastering);
- `scsi-id` — идентификатор, необходимый контроллеру для идентификации себя на SCSI-шине. Только некоторые контроллеры позволят изменить это значение. Типичное значение по умолчанию — 7.
- `parity` — ожидает ли SCSI-контроллер поддержку всеми подсоединенными устройствами четности при всех информационных обменах. Единица разрешает проверку четности, ноль — запрещает.

К сожалению, большей неразберихи, чем в настройках SCSI-контроллеров и устройств, наверное, не существует. До недавнего времени любая попытка улучшить поддержку SCSI-устройств в Linux оборачивалась тем, что какие-то новые контроллеры работали, а старые (казалось, уже давно отлаженные) теряли свою работоспособность.

Подробную информацию по настройкам следует искать в документации на конкретные контроллеры.

Жесткие диски

В этом разделе приводится список аргументов загрузки для стандартных жестких дисков (винчестеров) и устройств IDE.

Параметры драйвера IDE — винчестера/CD-ROM

Драйвер IDE допускает множество параметров, от определения геометрии диска до поддержки расширенных или дефектных микросхем контроллера:

- `hdx=` — распознается от `a` до `h`, например HDD;
- `idex=` — распознается от 0 до 3, например IDE1;
- `hdx=noprobe` — привод может присутствовать, но он не тестируется;
- `hdx=none` — жесткий диск отсутствует, CMOS игнорируется и тестирование не выполняется;
- `hdx=nowerr` — игнорируется бит `WRERR_STAT` на этом приводе;
- `hdx=cdrom` — привод присутствует и является приводом CD-ROM;
- `hdx=cyl,head,sect` — принудительное указание геометрии жесткого диска;
- `hdx=autotune` — привод попытается настроить скорость интерфейса на самый быстрый поддерживаемый режим PIO, который только возможен для этого привода. На старых материнских платах не гарантируется полная поддержка такого режима;
- `idex=noprobe` — не тестировать данный интерфейс;
- `idex=base` — задать адрес указанному интерфейсу, где `base` обычно `0x1f0` или `0x170`, а `ctl` подразумевается `base+0x206`;
- `idex=base,ctl` — указывает как `base`, так и `ctl`;
- `idex=base,ctl,irq` — указывает `base`, `ctl` и номер IRQ;
- `idex=autotune` — будет произведена попытка настроить скорость интерфейса на самый быстрый поддерживаемый режим PIO для всех приводов на этом интерфейсе. На старых материнских платах не гарантируется полная поддержка такого режима;
- `idex=noautotune` — привод не будет пытаться настроить скорость интерфейса;
- `idex=serialize` — не выполнять операции `overlap` на `idex`.

Подробная информация по конфигурации драйвера содержится в файле /usr/src/Linux-x.y.z/Documentation/ide.txt.

Последовательные и ISDN-драйверы

В разделе приведены параметры для некоторых ISDN-карт и так называемых мультипортовых последовательных контроллеров. Как обычно, первоначально единых стандартов не существовало, и из-за этого приходится иногда использовать параметры, передаваемые при загрузке ядра.

Драйвер PCBIT ISDN (*pcbit*)

Параметры:

```
pcbit=membase1,irq1[,membase2,irq2],
```

где `membaseN` — база разделяемой памяти для N-й карты; `irqN` — установленное прерывание для N-й карты. По умолчанию `IRQ=5` и `membase=0xD0000`.

Драйвер Teles ISDN (*teles*)

ISDN-драйвер требует аргументы загрузки в следующем виде:

```
teles=iobase,irq,membase,protocol,teles_id,
```

где `iobase` — адрес порта ввода/вывода карты; `membase` — базовый адрес разделяемой памяти карты; `irq` — прерывание, используемое картой; `teles_id` — уникальная строка идентификатора.

Драйвер DigiBoard (*digi*)

Драйвер мультипортового последовательного контроллера DigiBoard принимает строку из шести идентификаторов или целых чисел, разделенных запятыми. Значения по порядку:

- `Enable/Disable` — разрешить/запретить использование контроллера;
- тип карты — `PC/Xi(0)`, `PC/Xe(1)`, `PC/Xeve(2)`, `PC/Xem(3)`;
- `Enable/Disable` — разрешить/запретить альтернативное расположение контактов;
- количество портов на этой карте;
- порт ввода/вывода, на который сконфигурирована карта;
- база окна памяти.

Пример аргумента загрузки:

```
digi=E,PC/Xi,D,16,200,D0000
```

Более подробную информацию можно найти в файле /usr/src/Linux-2.4.3/Documentation/digiboard.txt.

Последовательный/параллельный радиомодем Baycom (*baycom*)

Формат аргумента загрузки для этого устройства:

```
baycom=modem,io,irq,options[,modem,io,irq,options]
```

`modem=1` означает, что у вас устройство `ser12`; `modem=2` — устройство `par96`. Значение `options=0` предписывает использование аппаратного DCD, а `opton=1` — программного DCD. Параметры `io` и `irq` — базовый порт ввода/вывода и прерывание.

Драйверы других устройств

В разделе приведены параметры загрузки других устройств, не вошедших ни в одну из упомянутых ранее категорий.

Устройства Ethernet (*ether*)

Драйверы для различных видов сетевых контроллеров поддерживают разные параметры, но им всем требуются значения прерывания, базовый адрес порта ввода/вывода и имя. В наиболее общей форме это выглядит так:

```
ether=irq,iobase[,param_1[,param_2,…]]],name
```

Первый нецифровой аргумент воспринимается как имя. Обычно значения `param_n` имеют различные назначения для разных сетевых контроллеров. Чаще всего этот параметр задают для второй сетевой карты, поскольку по умолчанию автоматически определяется только одна сетевая карта. Это можно сделать, указав

```
ether=0,0,eth1
```

Обратите внимание, что нулевые значения IRQ и базы ввода/вывода в примере заставляют драйвер сделать автоопределение параметров сетевой карты.

Данный пример не будет автоматически определять параметры второй сетевой карты в случае загружаемых модулей вместо вкомпилированных в ядро. Большинство современных дистрибутивов Linux используют ядро операционной системы в комбинации с загружаемыми модулями. Параметр `ether=` применяется только для драйверов, вкомпилированных непосредственно в ядро.

Полная информация по конфигурации и работе с несколькими сетевыми картами, а также описание особенностей настройки конкретных типов сетевых карт содержится в Ethernet-HOWTO.

Драйвер звуковой карты (*sound*)

Драйвер звуковой карты также может принимать аргументы загрузки для изменения вкомпилированных в ядро значений. Так делать не рекомендуется, поскольку в связи с отсутствием внятной документации подобные действия сильно смахивают на шаманство. Загружаемые модули намного надежнее. Тем более что за последнее время существенно улучшилось качество драйверов для звуковых карт и заметно увеличился ассортимент поддерживаемых драйверами устройств. Принимается аргумент загрузки в следующем виде:

```
sound=device1[,device2[,device3...]],
```

где каждое значение `deviceN` имеет формат `0xDTaaaId`. Расшифруем формат `deviceN`:

- `D` — второй канал DMA (ноль не применяется);
- `T` — тип устройства (список звуковых карт до типа 26 находится в файле /usr/src/Linux-x.y.z/include/linux/soundcard.h, а от 27 до 999 — в файле /usr/src/Linux-x.y.z/drivers/sound/dev_table.h.):
  - `1=FM`
  - `2=SB`
  - `3=PAS`
  - `4=GUS`
  - `5=MPU401`
  - `6=SB16`
  - `7=SB16-MIDI` и т. д.;

- `aaa` — адрес ввода/вывода в шестнадцатеричном представлении;
- `I` — номер прерывания в шестнадцатеричном представлении;
- `d` — первый канал DMA.

Параметр загрузки `sound=0` полностью запрещает драйвер звуковой карты.

Драйвер принтера (*lp*)

Этот аргумент загрузки позволяет сообщить драйверу принтера, какие порты доступны, а какие — нет. Параметр удобен для запрета захвата драйвером принтера всех доступных параллельных портов.

Формат аргумента — несколько пар адресов ввода/вывода, прерываний, например,
`lp=0x3bc,0,0x378,7`

Драйвер принтера будет использовать порт на `0x3bc` без прерывания и порт `0x378` с седьмым прерыванием. Порт `0x278` (если он присутствует в компьютере) не будет использоваться, поскольку автоопределение выполняется при отсутствии аргумента `lp=`. Параметр `lp=0` полностью отключает драйвер принтера.

Процесс init

После того как ядро Linux полностью загрузилось, считало конфигурационные параметры и настроило оборудование (по крайней мере, то, которое упоминалось в конфигурационных параметрах, и то, драйверы которого присутствуют в ядре), оно приступает к монтированию разделов жесткого диска. Монтирование всегда начинается с корневой файловой системы. Как только корневая файловая система будет загружена и смонтирована, появится сообщение:

```
VFS: Mounted root (ext2 filesystem) readonly
```

В этой точке система находит на корневой файловой системе программу init и выполняет ее.

Процесс init — это программа, ответственная за продолжение процедуры загрузки и перевод операционной системы из начального состояния, возникающего после загрузки ядра, в стандартное состояние. В это время init выполняет множество операций, необходимых для нормального функционирования операционной системы: монтирование и проверку файловых систем, запуск различных служб и т. п. Список производимых действий помимо конфигурации системы зависит от так называемого уровня выполнения (run level).

Достаточно простой аналогией уровня выполнения является обычный распорядок дня человека — пробуждение, приведение себя в порядок, завтрак, "выход в свет" — общение с окружающим миром, ужин, приведение себя в порядок, сон. Так изо дня в день, одни и те же операции, в одной и той же последовательности. Не умывшись, вы на работу не пойдете, завтрак обязательно идет перед ужином и т. д.

Точно так же разбиты уровни выполнения. Каждый уровень однозначно (по крайней мере, в пределах дистрибутива) определяет перечень действий, выполняемых процессом init, и конфигурацию запущенных процессов. К сожалению (а может, и к счастью), четкого разделения на уровни выполнения, их количество,

действия, выполняемые на каждом уровне, нет. Так, в некоторых UNIX-системах всего два уровня. Так же конфигурируют свою операционную систему некоторые дистрибутивы Linux (например, в дистрибутиве Slackware два уровня выполнения). В других дистрибутивах (Red Hat Linux) уровней выполнения восемь. Поскольку очень многие дистрибутивы принадлежат семейству Red Hat, на нем и остановимся.

В операционной системе Linux существует восемь уровней выполнения:

- 0 — останов системы;
- 1 — однопользовательский режим для специальных случаев администрирования. Отсутствует поддержка сети, практически нет сервисов;
- 2 — многопользовательский режим без поддержки сети;
- 3 — многопользовательский режим с поддержкой сети;
- 4 — использование не регламентировано;
- 5 — обычно по умолчанию стартует X Window System;
- 6 — перезагрузка системы;
- S или s — практически то же, что и однопользовательский режим, но уровень выполнения S используется, в основном, в скриптах.

Как можно заметить, существует определенное логическое нарушение в следовании уровней выполнения. Логичнее нулевой уровень выполнения вставить перед шестым. Однако здесь проявили себя исторические традиции — как повелось много лет назад в UNIX, так ради совместимости и остается.

К сожалению, не существует единого мнения, как использовать уровни со второго по пятый. В основном это определяется идеологами дистрибутива или пристрастиями системного администратора. Приведенная схема уровней выполнения оптимальна, и, в конечном итоге, только вы сами решаете, какие уровни выполнения вам нужны.

Конфигурационный файл init — /etc/inittab

Как всякая программа, после старта init сразу считывает свой конфигурационный файл /etc/inittab. Это обычный текстовый файл, состоящий из отдельных строк. Если строка начинается со знака `#` (стандартный признак комментария в конфигурационных файлах и скриптах) или пуста, она игнорируется. Все остальные строки состоят из четырех полей, разделенных двоеточиями:

```
id:runlevels:action:process
```

Здесь:

- `id` — идентификатор строки. Выбирается произвольно, но в файле не может быть двух строк с одинаковыми идентификаторами. Если конфигурационный файл модифицируют достаточно часто, целесообразно придерживаться неписанного правила нумерации строк в BASIC — номера строкам назначать кратно пяти или десяти;
- `runlevels` — уровни выполнения, на которых эта строка будет задействована. Уровни задают цифрами (без разделителей);
- `process` — команда, которая должна быть запущена;

- `action` — действие. В этом поле стоит ключевое слово, которое определяет, что должен делать процесс init, пока выполняется (или после выполнения) команда, заданная полем `process`:
  - `wait` — ожидать завершения процесса. Соответственно, пока не закончится данный процесс, init не запускает никаких других процессов. Как правило, такого типа процессы нужны для выполнения разнообразных проверочных действий (проверка и восстановление файловых систем), а также для запуска различных служб (демонов);
  - `once` — выполнить процесс только один раз;
  - `respawn` — перезапустить процесс в случае его "смерти". Актуально для некоторых служб, которые должны постоянно присутствовать в системе;
  - `off` — игнорировать данный элемент. Можно использовать при отладке конфигурационного файла;
  - `boot` — процесс должен быть выполнен при загрузке операционной системы, поле `runlevels` (уровни выполнения) при этом игнорируется;
  - `bootwait` — то же, что и предыдущая опция, но init должен ожидать окончания работы процесса;
  - `initdefault` — указывает init, в какой уровень выполнения необходимо перейти системе после загрузки;
  - `sysinit` — процесс должен быть выполнен во время загрузки операционной системы до выполнения любой строки с `boot` или `bootwait`;
  - `powerwait` — позволяет процессу init остановить систему при пропадании электроэнергии. Применение этого ключевого слова предполагает, что в наличии есть источник бесперебойного питания (UPS) со специальным интерфейсом, позволяющим посылать в компьютер и принимать из него различные управляющие сигналы (например "нет питания", "выключить источник бесперебойного питания", "аккумуляторы разряжены" и т. п.), а также программное обеспечение, которое отслеживает состояние источника бесперебойного питания и информирует init о том, что питание отключилось;
  - `ctrlaltdel` — разрешает init перезагрузить систему, когда пользователь нажимает комбинацию <Ctrl>+<Alt>+<Del> на клавиатуре. Однако системный администратор может определить действия по <Ctrl>++<Alt>+<Del>, например игнорировать нажатие этой комбинации.

Приведенный список не является исчерпывающим. Подробную информацию о файле inittab можно узнать из man-страниц init, inittab.

В качестве примера (листинг 7.1) рассмотрим файл inittab, который находится в только что установленной системе.

Листинг 7.1

```
# inittab    Этот файл описывает, как процесс INIT должен настроить
#  операционную систему на соответствующем уровне выполнения
#
# Author:  Miquel van Smoorenburg, <miquels@drinkel.nl.mugnet.org>
```

```
#       Modified for RHS Linux by Marc Ewing and Donnie Barnes
#

# Default runlevel. The runlevels used by RHS are:
#   0 — halt (Do NOT set initdefault to this)
#   1 — Single user mode
#   2 — Multiuser, without NFS (The same as 3, if you do not have
#       networking)
#   3 — Full multiuser mode
#   4 — unused
#   5 — X11
#   6 — reboot (Do NOT set initdefault to this)
#
id:3:initdefault:

# System initialization.
si::sysinit:/etc/rc.d/rc.sysinit

l0:0:wait:/etc/rc.d/rc 0
l1:1:wait:/etc/rc.d/rc 1
l2:2:wait:/etc/rc.d/rc 2
l3:3:wait:/etc/rc.d/rc 3
l4:4:wait:/etc/rc.d/rc 4
l5:5:wait:/etc/rc.d/rc 5
l6:6:wait:/etc/rc.d/rc 6

# Things to run in every runlevel.
ud::once:/sbin/update
# Trap CTRL-ALT-DELETE
ca::ctrlaltdel:/sbin/shutdown -t3 -r now

# When our UPS tells us power has failed, assume we have a few
# minutes
# of power left.  Schedule a shutdown for 2 minutes from now.
# This does, of course, assume you have powerd installed and your
# UPS connected and working correctly.
pf::powerfail:/sbin/shutdown -f -h +2 "Power Failure; System Shutting Down"

# If power was restored before the shutdown kicked in, cancel it.
pr:12345:powerokwait:/sbin/shutdown -c "Power Restored; Shutdown Cancelled"


# Run gettys in standard runlevels
```

```
1:2345:respawn:/sbin/mingetty tty1
2:2345:respawn:/sbin/mingetty tty2
3:2345:respawn:/sbin/mingetty tty3
4:2345:respawn:/sbin/mingetty tty4
5:2345:respawn:/sbin/mingetty tty5
6:2345:respawn:/sbin/mingetty tty6

# Run xdm in runlevel 5
# xdm is now a separate service
x:5:respawn:/etc/X11/prefdm -nodaemon
```

Сразу после запуска процесс init считывает свой конфигурационный файл /etc/inittab и производит его разбор. Сначала он определяет, какой уровень по умолчанию установлен в системе. Как видно из приведенного конфигурационного файла id:3:initdefault, уровень выполнения, на котором будет функционировать операционная система после загрузки, равен трем (то есть предполагается многопользовательский режим с поддержкой сетевых функций). Дистрибутив Fedora Core по умолчанию предлагает установить вход в систему в графическом режиме — пятый уровень выполнения.

Затем процесс init принимает к сведению строки, содержащие специальные команды:

```
ca::ctrlaltdel:/sbin/shutdown -t3 -r now
pf::powerfail:/sbin/shutdown -f -h +2 "Power Failure; System Shutting Down"
pr:12345:powerokwait:/sbin/shutdown -c "Power Restored; Shutdown Cancelled"
```

После этого процесс init инициирует команду, которую необходимо запустить при старте системы, но перед тем как перейти к какому-нибудь уровню выполнения. Эта команда содержится в строке с ключевым словом `sysinit`:

```
si::sysinit:/etc/rc.d/rc.sysinit
```

Далее процесс init запускает скрипты, которые должны действовать в любом уровне выполнения:

```
ud::once:/sbin/update
```

а затем команды, соответствующие уровню, заданному по умолчанию:

```
l3:3:wait:/etc/rc.d/rc 3
1:2345:respawn:/sbin/mingetty tty1
2:2345:respawn:/sbin/mingetty tty2
3:2345:respawn:/sbin/mingetty tty3
4:2345:respawn:/sbin/mingetty tty4
5:2345:respawn:/sbin/mingetty tty5
6:2345:respawn:/sbin/mingetty tty6
```

Как можно заметить, есть несколько строк, запускающих скрипт rc, которые отличаются только уровнем выполнения и аргументом командной строки, передаваемой в этот скрипт. Функции, выполняемые скриптами rc.sysinit и rc, будут рассмотрены в *разд. "Основные конфигурационные файлы"* данной главы.

После скрипта rc процесс init запускает шесть виртуальных консолей (процессов mingetty или, в более старом варианте, — gettty), что дает пользователям возможность регистрироваться в системе с терминалов (или виртуальных консолей, поскольку терминал вы вряд ли где-нибудь встретите). Для переключения между виртуальными консолями необходимо нажимать комбинацию одной из русифицированных клавиш верхнего ряда клавиатуры: клавиши <Alt> с номером, соответствующим номеру виртуальной консоли. После инициализации виртуальных консолей можно считать, что система полностью перешла на соответствующий уровень выполнения, загрузка завершилась, операционная система ожидает регистрации пользователя.

После окончания загрузки init продолжает функционировать в фоновом режиме. Поэтому с помощью команды `telinit`, которая взаимодействует с процессом init, можно перевести систему с одного уровня выполнения на другой или указать init перечитать свой конфигурационный файл.

Когда пользователь останавливает систему (командой `shutdown`, `halt`, `poweroff` или `reboot`), процесс init завершает все исполняющиеся процессы, размонтирует все файловые системы и останавливает процессор или перезагружает систему.

Основные конфигурационные файлы

Таким образом, из предыдущего раздела ясно, что процесс init выполняет три основных действия:

- запускает скрипт rc.sysinit из каталога /etc/rc.d;
- запускает скрипт rc из того же каталога /etc/rc.d с опцией, равной уровню выполнения (обычно третий или пятый уровни выполнения);
- запускает процессы getty.

Как следует из материала *главы 6*, в каталоге /etc находится каталог rc.d, содержимое которого непосредственно касается процесса загрузки системы. Вот оно:

/init.d	/rc2.d	/rc5.d	rc.local
/rc0.d	/rc3.d	/rc6.d	rc.sysinit
/rc1.d	/rc4.d	rc	

Опираясь на предыдущую информацию, нетрудно заметить, что существует семь каталогов для каждого уровня выполнения, какой-то каталог /init.d и три исполняемых файла, два из которых нам уже знакомы — rc и rc.sysinit. Третий файл — rc.local — вызывается по окончании исполнения файла rc и предназначен для команд, добавляемых администратором для запуска в процессе начальной загрузки. Редактировать файл rc не возбраняется, однако вероятность ошибки в файле, содержащем сотню-другую строк, очень велика, поэтому настоятельно рекомендуется использовать только файл rc.local.

rc.sysinit

Вернемся к процессу загрузки. Файл rc.sysinit предназначен для выполнения начальных действий, необходимых для корректного функционирования операционной системы. Далее приведен список действий, выполняемых скриптом rc.sysinit.

Конечно, этот перечень зависит от дистрибутива и от конфигурации системы, но бóльшая его часть неизменна.

Действия скрипта:

- установка путей;
- установка имени хоста;
- чтение конфигурационных данных из /etc/sysconfig/network;
- вывод баннера;
- монтирование файловой системы /proc;
- конфигурирование параметров ядра системы, используя файл /etc/sysctl.conf;
- установка системных часов с помощью конфигурации из /etc/sysconfig/clock;
- установка параметров клавиатуры консоли программой loadkeys в соответствии с файлами /etc/sysconfig/console/default.kmap или /etc/sysconfig/keyboard;
- загрузка системного шрифта из /etc/sysconfig/i18n и файлов с расширением pcf.gz или gz из каталогов /etc/sysconfig/console, /usr/lib/kbd/consolefonts или /lib/kbd/consolefonts;
- активация области подкачки;
- инициализация USB-контроллера;
- запуск программы fsck для корневой системы, при обнаружении серьезных проблем выполняется немедленная перезагрузка;
- старт PNP-устройств в соответствии с /etc/isapnp.conf;
- перемонтирование корневой файловой системы в режим чтения/записи;
- перенастройка таблицы монтирования /etc/mtab;
- проверка квот для корневой файловой системы;
- проверка необходимости загрузки модулей, нахождение зависимостей, загрузка и конфигурирование модулей;
- подключение RAID-устройств;
- запуск fsck для других систем;
- монтирование локальных файловых систем;
- включение механизма квот;
- удаление триггерных файлов загрузки;
- очистка каталогов /var/lock и /var/run;
- очистка файлов /var/run/utmp и /var/run/utmpx;
- удаление файлов-защелок из /tmp;
- включение подкачки;
- инициализация последовательных устройств, используя скрипт /etc/rc.d/rc.serial;
- загрузка модулей для SCSI-стримера;
- генерация файла заголовка для определения загружаемого ядра командой /sbin/mkkerneldoth;
- установка ссылки /boot/System.map;
- проверка использования интерактивного режима загрузки и, в случае необходимости, создание файла /var/run/confirm.

Запуск проверки файловой системы командой `fsck` можно принудительно отключить при наличии файла /fastboot, а также включить при наличии /forcefsck.

Создать эти файлы можно командой `shutdown` с соответствующими ключами. Однако злоупотреблять этими возможностями не рекомендуется.

Sysctl позволяет зафиксировать ряд параметров и обеспечить (через /etc/sysctl.conf) их установку после перезагрузки. Листинг 7.2 иллюстрирует /etc/sysctl.conf сразу после инсталляции системы.

Листинг 7.2

```
# Disables packet forwarding
net.ipv4.ip_forward = 0
# Enables source route verification
net.ipv4.conf.all.rp_filter = 1
# Disables the magic-sysrq key
kernel.sysrq = 0
```

Скрипт rc

Прежде чем приступить к разбору скрипта rc, необходимо упомянуть о каталогах /rcX.d и /init.d. Уточним еще раз — иерархия /rcX.d характерна для дистрибутивов Red Hat и базирующихся на нем, в других дистрибутивах и в UNIX-системах они могут отсутствовать. Эти каталоги играют исключительную роль в процессе загрузки, поскольку они содержат основные скрипты, необходимые для организации процесса загрузки.

Подкаталог /init.d содержит по одному скрипту для каждой из служб, установленных в системе (sendmail, HTTP, Samba, FTP и т. п.). Этот скрипт отвечает за запуск, остановку или перезагрузку соответствующей службы. В каталоге /rcX.d находятся ссылки на файлы скриптов, как правило, расположенные в каталоге /etc/rc.d/init.d. Названия этих ссылок начинаются либо с буквы K, либо с буквы S, после которой идет двузначное число и имя соответствующей службы. Буквы S и K — первые буквы слов start и kill соответственно. Из этого следует, что файл, начинающийся с буквы S, отвечает за старт соответствующего процесса, а файл, начинающийся с буквы K, отвечает за остановку соответствующего процесса. Цифры, идущие после S или K в именах ссылок, задают порядок запуска скриптов.

Заглянем в файл rc. Первым делом скрипт пытается определить текущий уровень выполнения и уровень, на который необходимо перевести систему. После этого он проверяет, нажимал ли пользователь клавишу <I> для перехода в режим пошаговой загрузки процессов. Затем скрипт останавливает запущенные на предыдущем уровне выполнения процессы, отсутствующие на новом уровне выполнения, а потом запускает необходимые службы для нового уровня выполнения. Как правило, одна и та же служба нужна на нескольких уровнях. Поэтому не имеет смысла эту службу при переходе с одного уровня выполнения на другой останавливать и тут же запускать. В Linux для этой цели предусмотрены специальные флаги.

В качестве флагов служат файлы в каталоге /var/lock/subsys/${subsys} или /var/lock/subsys/${subsys}.init, где subsys — имя соответствующей службы. Если

файлов нет, то данный процесс считается незапущенным (запуск S-файла имеет смысл), а если есть — запущенным (запуск K-файла имеет смысл). Для программы linuxconf создается специальный флаг /var/run/runlevel.dir, из которого можно узнать текущий уровень выполнения системы.

Для управления набором доступных служб в текущем уровне выполнения можно использовать программу конфигурирования linuxconf, программу ntsysv, /usr/sbin/setup или графическую программу Control-panel.

Можно сконфигурировать набор доступных сервисов и вручную. Для запрета старта какого-либо сервиса достаточно просто удалить соответствующую ссылку (SXXlalala) из необходимого каталога /rcX.d, а для разрешения — создать соответствующую ссылку в нужном каталоге /rcX.d. Однако не следует забывать помимо стартовой ссылки создавать стоповую, иначе возможны проблемы, когда система некорректно завершит функционирование сервиса, для которого забыли создать стоповую ссылку. А как же корректно установить порядковый номер у соответствующей ссылки? Конечно, можно чисто эмпирически подобрать номер, исходя из функций, выполняемых сервисом. Но давайте заглянем в любой файл в каталоге /etc/rc.d/init.d/, к примеру, в файл anacron (листинг 7.3).

Листинг 7.3

```
#!/bin/sh
# Startup script for anacron
# chkconfig: 2345 95 05
# description: Run cron jobs that were left out due to downtime

# Source function library.
. /etc/rc.d/init.d/functions
[ -f /usr/sbin/anacron ] || exit 0
prog="anacron"
start() {
    echo -n $"Starting $prog: "
    daemon anacron
    RETVAL=$?
    [ $RETVAL -eq 0 ] && touch /var/lock/subsys/anacron
    echo
    return $RETVAL
}
stop() {
    if test "x"pidof anacron"" != x; then
      echo -n $"Stopping $prog: "
      killproc anacron
      echo
    fi
```

```
    RETVAL=$?
    [ $RETVAL -eq 0 ] && rm -f /var/lock/subsys/anacron
    return $RETVAL
}
case "$1" in
      start)
      start
      ;;
      stop)
      stop
      ;;
      status)
      status anacron
      ;;
      restart)
      stop
      start
      ;;
      condrestart)
      if test "x"pidof anacron"" != x; then
        stop
        start
      fi
      ;;
      *)
      echo $"Usage: $0 {start|stop|restart|condrestart|status}"
      exit 1
esac
exit 0
##############################################################################
```

Обратите внимание на часть заголовка файла:

```
#!/bin/sh
# Startup script for anacron
# chkconfig: 2345 95 05
# description: Run cron jobs that were left out due to downtime
```

Помимо указания, какой командной оболочкой необходимо воспользоваться, там есть строчка:

```
# chkconfig: 2345 95 05
```

из которой следует, что этот скрипт может использоваться на втором, третьем, четвертом и пятом уровнях выполнения, а цифры `95` и `05` — порядковый номер для стартового (95) и стопового (05) скриптов. Обратите внимание — в сумме эти две цифры

составляют 100. Таким образом, достаточно добиться того, чтобы порядок останова сервисов был в точности обратный стартовому. `Description` в приведенном файле — это комментарий, который linuxconf выдает на экран для объяснения роли данного сервиса.

Если внимательно посмотреть скрипт, то сразу видно, что опций у него больше, чем стандартные `start` и `stop`. Имеются еще `restart`, `condrestart` и `status`. Старт, останов и проверка состояния демона выполняются рядом функций типа `daemon`, `killproc`, `status`. Процедуры `daemon`, `killproc`, `status` определяются в файле /etc/rc.d/init.d/functions (а тот пользуется определениями из /etc/sysconfig/init). Они предназначены для старта, останова и проверки статуса демона (сервиса).

Функция `daemon` обеспечивает старт сервиса. При этом можно учесть особенности поведения демона. Перед стартом сервиса всегда проверяется его наличие в системе. Так как появление *дампа* памяти (дамп — моментальный снимок памяти, используемой зависшей программой на момент ее краха) сервиса может привести к проблемам с безопасностью, то все демоны запускаются в режиме без создания дампа памяти.

Процедура `killproc` останавливает сервис. Данная функция предполагает один аргумент в виде имени демона и, при необходимости, еще один для указания сигнала, который будет послан сервису. Сигнал `SIGKILL` часто может быть нежелателен для останова сервиса, поэтому если сигнал назначен, то используется только он, в противном случае сперва посылается `SIGTERM`, и если данный сигнал не произвел на процесс впечатления, то посылается сигнал `SIGKILL`. В последнюю очередь скрипт подчищает различные блокировочные файлы.

Функция `status` позволяет проверить текущее состояние сервиса. Если сервис нормально функционирует, то просто сообщается об этом факте. В противном случае проверяется наличие флаговых файлов (/var/run/подсистема.pid и /var/lock/subsys/подсистема), которые должны блокировать повторный запуск сервиса. Таким образом, несложно самому создать скрипт для управления сервисом.

rc.local

Файл /etc/rc.d/rc.local выполняется после скрипта rc. В него рекомендуется помещать дополнительные сервисы или персональные настройки. Однако обычный пользователь редко прибегает к этой возможности.

Другие файлы, влияющие на процесс загрузки

Все файлы конфигурации, задействованные при загрузке системы, находятся в каталоге /etc:

- /etc/fstab — содержит информацию об автоматически монтируемых при старте файловых системах;
- /etc/skel — образцы файлов конфигурации, используются при создании учетных записей новых пользователей;
- /etc/bashrc — общесистемный файл для командной оболочки;

- /etc/initscript — файл, позволяющий задать специфические действия для каждой команды из файла /etc/inittab (подробную информацию следует искать в справочной системе).

Второстепенные файлы конфигурации:

- /etc/issue — сообщение, выдаваемое системой до приглашения "login:";
- /etc/motd — сообщение, выдаваемое системой после регистрации пользователя.

События, происходящие при регистрации пользователя

Последовательность событий при регистрации пользователя такова:

1. Пользователь вводит свое регистрационное имя (login) по приглашению "login:" процесса getty.
2. Процесс getty выполняет программу login, передавая программе login в качестве аргумента регистрационное имя пользователя.
3. Программа login запрашивает пароль и сверяет регистрационное имя и пароль пользователя с записанными в файле /etc/passwd (login) и /etc/shadow (пароль). При этом введенный пароль пользователя шифруется по специальному алгоритму (в последнее время чаще всего используется алгоритм MD5), и полученный результат сравнивается с зашифрованным паролем, хранящимся в /etc/shadow.
4. В случае если регистрационное имя пользователя или пароль не совпали с хранящимся в системе, то после паузы (около 3 с, задержка настраивается) выводится сообщение `Password incorrect`. Программа login завершает свою работу, а процесс getty снова выводит приглашение "login:".
5. Если проверка регистрационного имени и пароля пользователя прошла успешно, login выводит на экран из файла /etc/motd так называемое "сообщение дня".
6. Затем login запускает командную оболочку (shell), указанную в бюджете пользователя, и устанавливает переменную среды TERM.
7. Оболочка shell выполняет файлы, исполняемые при регистрации пользователя в системе, сперва общесистемные, а затем пользовательские (если это Bourne-shell, выполняется файл .profile, если C-shell — .login и .cshrc, если Korn-shell — .profile и .kshrc). В этих файлах можно указать специфические настройки пользователя, переменные окружения, запустить какие-то приложения. После этого shell выводит на экран приглашение и ожидает ввода информации.

Основные файлы, участвующие в регистрации пользователя

В регистрации пользователя участвуют следующие файлы:

- /etc/profile — общесистемный профильный файл, устанавливает пути и другие важнейшие переменные;
- /etc/passwd — различная регистрационная информация, такая как имя пользователя, группа пользователя, домашний каталог и командный интерпретатор;

- /etc/shadow — в определенной мере дублирует файл passwd, но его основное назначение — хранить пароли пользователей;
- /etc/bashrc — общесистемный файл конфигурации bash;
- /домашний каталог/.* — пользовательские файлы конфигурации.

Если требуется, чтобы при регистрации пользователя выполнялся какой-то скрипт или устанавливались переменные окружения, то для этого вызов данного скрипта нужно поместить в ~/.profile или в .bash_profile.

Если необходимо, чтобы пользователь не мог отменить выполнение какого-то скрипта или команды при его регистрации в системе, то следует вписать в ./etc/profile дополнительный код:

```
if test $USER = petya; then
        echo Hello Petya!
#здесь запускается ваш скрипт
fi
```

Эти команды будут исполняться только при регистрации в системе пользователя petya.

Загрузка в однопользовательском режиме

Помимо стандартной загрузки, описанной ранее, можно заставить процесс init загрузить систему на уровне выполнения, отличном от загружаемого по умолчанию. Данная возможность крайне необходима в случае, когда у операционной системы есть проблемы. Это могут быть чьи-то неудачные эксперименты с файлом inittab или неправильно сконфигурированный процесс (например, если сеть нормально не настроена, sendmail пытается найти хост, которого нет, что может привести к задержке при старте системы в 10 минут и более). При возникновении такой ситуации необходимо перевести систему в однопользовательский режим и решить проблему. Для этого в строку приглашения `LILO (boot:)` нужно ввести аргументы `single` или `emergency`. Это позволит загрузить систему в однопользовательском режиме (уровень выполнения 1), при котором в системе работает только суперпользователь (root) и запускается очень небольшое число самых необходимых системных служб, включая login. Другой способ перевода системы в однопользовательский режим — применение команды `telinit`, которая, собственно, является символической ссылкой на сам init. Команда `telinit`, помимо перевода системы с одного уровня выполнения на другой, может заставить процесс init перечитать файл inittab без перезагрузки операционной системы.

Однопользовательский режим необходим для выполнения административных задач, таких как проверка и восстановление файловой системы. Например, запуск fsck в разделе /usr. Как правило, необходимость перехода в однопользовательский режим возникает тогда, когда fsck не может автоматически восстановить файловую систему при загрузке. Такое случается редко, обычно при пропадании электроэнергии во время работы компьютера или выходе из строя жесткого диска.

Однако бывают ситуации, когда в однопользовательском режиме загрузиться с жесткого диска невозможно. В этом случае можно загрузиться в однопользователь-

ском режиме с дискеты или с CD-ROM. Такое может произойти при серьезном крахе системы (что иногда бывает смертельно для информации на жестком диске) или когда, к примеру, при установке Windows переписывает MBR, и в результате уничтожается загрузчик (разделы Linux и система жизнеспособны, только загрузить ее нечем).
Из соображений безопасности правильно сконфигурированная система при загрузке в однопользовательском режиме запросит пароль пользователя root.

Существует еще один способ вмешаться в процесс загрузки. При загрузке системы выдается сообщение `Press "I" to enter interactive startup`. Если вы нажмете клавишу <I>, система перейдет в режим пошаговой загрузки сервисов (подобно нажатию клавиши <F8> в Windows и выбора режима загрузки step by step).

Это позволит отказаться от загрузки подозрительных (с вашей точки зрения) процессов и определить, при запуске какого из них возникают неприятности.

Утилиты

Подводя итог, перечислим утилиты, участвующие в процессе загрузки системы:

- init — программа, управляющая загрузкой операционной системы;
- telinit — утилита для управления процессом init;
- runlevell — выводит текущий уровень выполнения;
- linuxconf — утилита конфигурации операционной системы Linux. В том числе позволяет редактировать список сервисов, запускаемых в текущем уровне выполнения;
- ntsysv — консольная утилита для редактирования списка сервисов, запускаемых в текущем уровне выполнения;
- /usr/sbin/setup — консольная утилита для конфигурирования операционной системы;
- control-panel — графическая утилита для конфигурирования операционной системы.

Ссылки

- **www.osp.ru/os/2001/02/073.htm** — И. Облаков. Восход солнца вручную.
- **/usr/src/Linux-x.y.z/Documentation/** — много информации, так или иначе связанной с ядром операционной системы, драйверами, файловыми системами и т. п.
- Справочные страницы man — init, inittab, telinit, initscript.
- Соответствующие HOWTO:
  - Ethernet-HOWTO — различные тонкости настройки сетевых адаптеров;
  - The Linux BootPromt HOWTO — справочник по аргументам начальной загрузки, передаваемым ядру Linux во время загрузки системы;
  - The Linux Bootdisk HOWTO — создание загрузочной дискеты.

Глава 8



Безопасная работа в Linux

В этой главе мы обзорно, особо не вдаваясь в детали, рассмотрим вопросы повышения безопасности операционной системы Linux. Подробности вы всегда найдете в соответствующей литературе и главах, описывающих конкретное программное обеспечение.

Основные положения

Зачем вам безопасность?

Безопасность компьютеров и компьютерных сетей сейчас, в связи с повсеместным распространением Интернета и электронной коммерции, все больше выходит на первый план. В любой серьезной организации при приеме на работу системного администратора одним из основных требований к нему является умение организовать безопасность системы и сетей.

Даже если вы устанавливаете один-единственный сервер, не исключена возможность, что кто-то попробует взломать его просто от скуки. Поэтому нельзя особо надеяться, что на ваш сервер или Web-страницу никто не покусится.

Надежность защиты системы

Следует помнить: "все, что один человек построил, другой всегда сможет поломать". Надежность защиты, в идеале, должна соответствовать следующему правилу: затраты взломщика на преодоление защиты должны существенно превышать стоимость поврежденных или украденных данных. Это, конечно, не означает, что домашнюю систему, на которой ничего важного нет, защищать не надо. Просто необходимо соразмерять затраченные усилия — для домашнего пользователя их потребуется гораздо меньше, чем для банковской сети.

У защиты есть одна особенность — чем более безопасна система, тем больше усилий необходимо затрачивать на поддержание ее в рабочем состоянии, и тем более навязчивой становится сама система безопасности. Всегда необходимо соблюдать золотую середину между безопасностью системы и неудобствами пользователей, связанными с режимом безопасности.

Защита не бывает идеальной. Вы должны представлять, сколько времени и усилий необходимо потратить на восстановление или воссоздание данных при их потере.

Определение приоритетов защиты

До того как начать настраивать безопасность системы, необходимо определить, от чего и как будет защищена ваша система, какие службы должны быть защищены в первую очередь. Следует четко представлять, какая информация или какое оборудование наиболее ценно, что ни при каких обстоятельствах не должно пропасть или попасть к постороннему человеку, а что имеет минимальную ценность (было бы нелепо выстроить супермощную систему для защиты сочинений вашего ребенка и установить минимальную безопасность для банковских документов, стоящих миллионы).

Кроме того, всегда нужно помнить — никаких поблажек для пользователей вне зависимости от того, насколько высокое положение они занимают. Пользователь, который имеет привилегии, неизбежно является брешью в безопасности.

Политика безопасности

Если ваша задача — администрирование средней или большой сети, необходимо разработать документ, который называется "Политика безопасности" и определяет права и обязанности пользователей, меры по защите системы и действия, применяемые в случае нарушения безопасности системы.

Этот документ должен быть утвержден руководством фирмы, и с ним следует ознакомить каждого сотрудника, причем желательно под расписку. Чем проще и понятнее составлен документ, тем больше вероятность, что его поймут и будут им руководствоваться. Главное правило, которое должно быть четко зафиксировано: "То, что не разрешено — запрещено".

Основные направления защиты

Прежде чем что-то защищать, необходимо знать, что, от кого и от чего.

- ❒ Физическая защита:
  - от физического проникновения в компьютер;
  - от аппаратных сбоев.
- ❒ Защита рабочей станции (локальная безопасность).
- ❒ Защита сервера:
  - от пользователей;
  - от внешнего мира.
- ❒ Защита сети:
  - от проникновения извне;
  - от взломов изнутри.

Физическая безопасность

Первое, что следует сделать, — позаботиться о физической безопасности системы. Нет никакого прока от программной защиты, если из сервера можно просто изъять жесткий диск и переписать данные. Необходимо выяснить, кто имеет пря-

мой физический доступ к системе, а также можно (и нужно ли) защитить систему от их потенциально вредного воздействия.

Степень физической безопасности напрямую зависит от ситуации и финансовых возможностей. Домашнему пользователю, скорее всего, не нужна сильная физическая защита. Офисный компьютер желательно обезопасить на время отсутствия пользователя. А если это сервер, то его физическая защищенность должна быть максимальной, в идеальном случае он должен находиться в бронированной комнате без окон, с сейфовым замком на стальной двери.

Замки

Практически любой серверный корпус имеет специальный замок или отверстия для установки навесного замка. Как правило, наружные отсеки для винчестеров тоже снабжены замками.

С корпусом обычного компьютера дело обстоит похуже. Сейчас не встретишь замок блокировки клавиатуры, хотя несколько лет назад он был обязательным элементом любого компьютерного корпуса. Но даже наличие такого замка — крайне слабая защита, поскольку к такому замку подходит любой ключ от аналогичных корпусов, а с помощью обыкновенной канцелярской скрепки можно открыть его за несколько секунд. Тем не менее для неопытного взломщика даже простейший замок является преградой.

Еще хуже дело обстоит с ноутбуками — его можно просто взять со стола. Правда, любой современный ноутбук имеет специальное отверстие, куда крепится замок, похожий на велосипедный.

Охрана жесткого диска

Если на жестком диске хранится информация, которая ни в коем случае не должна попасть в чужие руки, и лучше ее уничтожить, чем допустить пропажу, то для этого необходимо предпринять дополнительные меры. Если у вас небольшой сервер или рабочая станция, то жесткий диск, на котором хранится информация, нужно установить в специальную съемную корзину, так называемый Rack Mount. Это позволит по окончании работы извлекать жесткий диск из компьютера и прятать в сейф. Для больших серверов так сделать несколько затруднительно, но иногда имеет смысл.

BIOS

BIOS — самое первое звено, от которого зависят настройки компьютера, и в 99% случаев взломщик попытается проникнуть в BIOS. Практически любая BIOS позволяет установить два типа паролей — на вход в систему и на вход в BIOS. Это не дает стопроцентной защиты (установки BIOS можно обнулить, если кто-то имеет доступ вовнутрь корпуса), но может быть хорошим сдерживающим фактором. Однако следует помнить, что при установке загрузочного пароля на сервере система не сможет загрузиться без вмешательства администратора.

(Очень неприятно, когда в три часа ночи вас выдергивают из постели для того, чтобы вы ввели пароль в сервер.) Многие современные ноутбуки содержат специальный модуль шифрования данных на жестком диске, управление которым находится в BIOS.

Загрузочные устройства

Если не установлен пароль в BIOS, велика вероятность, что взломщик попытается загрузить систему с помощью системной дискеты, DVD-ROM-диска, Zip-дисковода, USB Flash-устройства и т. п. Это дает ему полный доступ к жестким дискам компьютера. Поэтому, помимо установки пароля на вход в BIOS, необходимо запретить в BIOS загрузку со всех устройств, кроме жесткого диска. Также рекомендуется либо отключить в BIOS сервера дисководы съемных устройств, либо (при наличии такой возможности) физически отключить флоппи-, Zip- и тому подобные дисководы и DVD-ROM, а иногда даже вообще не устанавливать их в сервер.

Безопасность загрузчика операционной системы

Загрузчики Linux также имеют возможность установки стартового пароля. Это позволяет предотвратить несанкционированную загрузку операционной системы.

Программы xlock и vlock

Если вы отходите на какое-то время от своего рабочего компьютера и не хотите выключать его, используйте программы xlock и vlock. Эти программы заблокируют доступ к системе, причем как визуально (черный экран или какая-то надпись типа "Консоль заблокирована"), так и с помощью клавиатуры:

- xlock — предназначена для X Window. Она "запирает" дисплей и для продолжения работы запрашивает пароль;
- vlock — консольная программа, которая позволяет "запереть" часть или все виртуальные консоли системы.

Конечно, "запирание" консоли не позволит нанести прямой вред работе, однако не помешает перезагрузить машину (кнопку Reset или отключение питания еще никто не отменял).

Определение нарушений физической безопасности

При подозрении на попытку нарушения физической безопасности системы первым делом проверьте то, что сразу бросается в глаза — наличие физических повреждений. Второе — определите, перезагружалась ли система. Операционная система Linux очень надежна, и непроизвольные перезагрузки по ее вине исключены. Если ОС непроизвольно перезагружается, причина, скорее всего, в аппаратной части. Во всех остальных случаях система перезагружается администратором (пользователем) или по команде источника бесперебойного питания. Если компьютер перезагружен без вас — это проблема, требующая изучения.

Вот часть того, что следует проверить:

- короткие или незаконченные системные журналы;
- системные журналы, которые содержат неверные права доступа или права собственности;
- системные журналы, в которых присутствуют записи перезагрузки или перезапуска сервисов;
- отсутствие системных журналов;
- подключение пользователя с нетипичного для него места или использование программы su пользователем, который никогда этого не делал.

Также крайне желательно вести журнал с указанием даты, времени и причины перезагрузки системы. Время, прошедшее с момента перезагрузки, можно узнать из системного журнала или командой `uptime`.

Локальная безопасность

Сразу после установки системы необходимо позаботиться о защите от локальных пользователей. Достаточно много методов взлома основано на различных недочетах или ошибках программного обеспечения, доступного только им. Локальный пользователь и сам по себе потенциально опасен. Предоставьте ему чуть больше прав, чем следует — и одной простой командой `rm` он может "снести" половину операционной системы.

Регистрация новых пользователей

Следует разработать и неукоснительно соблюдать правила регистрации новых пользователей в системе. Существует несколько правил, которых необходимо придерживаться при работе с пользователями:

- предоставлять минимальное количество привилегий;
- отслеживать, когда и откуда происходит регистрация пользователя;
- не забывать удалить пользователя, если он больше не работает в фирме;
- стараться максимально ограничить число пользователей, имеющих нестандартную конфигурацию или привилегии.

Безопасность пользователя root

Наиболее желанное приобретение для взломщика — пароль суперпользователя (root). Поскольку этот пользователь в системе "царь и бог", проблемы, связанные с ним, для системы могут быть катастрофическими. Следует заходить в систему под именем пользователя root как можно реже и, желательно, не через сеть. Существует несколько правил, которых необходимо придерживаться при работе в системе в качестве пользователя root:

- старайтесь избегать использования сложных комплексных команд или длинных одиночных команд. Велика вероятность того, что вы ошибетесь и выполните не то, что требуется;

- вводите потенциально опасные команды (удаление, переименование, перенос файлов) со специальным ключом, который заставляет команду спрашивать, действительно ли вы хотите совершить эту операцию с файлами. Помните, удаленные файлы в операционной системе Linux восстановить невозможно;
- регистрируйтесь в системе как пользователь root только в экстраординарных случаях. Для выполнения отдельных команд вполне можно воспользоваться программами su и sudo;
- исключите из своего и пользовательского обихода r-утилиты — rlogin, rsh, rexec и тому подобные, а также программу telnet. Лучше просто удалите их из всех систем. Эти программы были хороши лет двадцать пять назад. Теперь же каждый второй взломщик пытается ими воспользоваться в корыстных целях. В качестве замены используйте пакет SSH;
- сначала хорошенько подумайте, что собираетесь делать, потом проверьте, правильно ли ввели команду, и только затем нажмите клавишу <Enter>.

Безопасность файлов и файловой системы

Множество взломов операционной системы увенчались успехом из-за неправильной установки прав доступа к файлам или проблем с файловой системой. Существует несколько правил, которых необходимо придерживаться при работе с установкой прав:

- ограничьте возможность запуска пользователем специальных команд или файлов. Для разделов, на которые может записывать данные любой пользователь, в файле /etc/fstab поставьте опцию `nosuid`. Можно использовать `nodev` (запрещает создание символьных и блочных устройств), `noexec` (запрет выполнения программ) и `ro` (монтировать раздел только для чтения);
- рекомендуется отказаться от NFS. Если все же NFS установлена, применяйте максимальные ограничения;
- настройте маску для создания пользователями файлов в максимально ограничивающем режиме. Идеальный вариант — маска 077;
- установите квоты на использование файловой системы для пользователей. Также желательно запретить приложениям пользователя создавать дамп памяти программы на диске;
- старайтесь свести к минимуму количество SUID- и SGID-файлов в системе. Поскольку эти программы предоставляют пользователям, которые их запускают, специальные привилегии, необходимо убедиться, что небезопасные программы не установлены;
- обнаруживайте и удаляйте файлы .rhosts;
- прежде чем изменить права доступа для системных файлов, убедитесь, что понимаете, что делаете. Никогда не изменяйте права доступа файла только потому, что это простой способ заставить что-то работать;
- периодически проверяйте права доступа ко всем важнейшим файлам системы. Изменение прав доступа к файлам — один из основных признаков взлома системы.

Проверка целостности файлов

Хороший способ обнаружения атаки на операционную систему — проверка целостности файлов. Существует несколько пакетов, позволяющих проверить систему. Простейший случай — программа rpm, с помощью которой можно сверить все файлы установленных в системе пакетов. Однако эта программа не может сверить файлы, попавшие в систему, минуя `rpm`.

Например, программа Tripwire вычисляет контрольные суммы для всех важных бинарных и конфигурационных файлов в системе и сравнивает их с предыдущими записями, хранящимися в базе данных. Никто не мешает написать скрипт, автоматизирующий эту процедуру. Хранить базу данных во избежание подмены ее взломщиком рекомендуется либо на дискете, либо на другом компьютере.

Особенности безопасности файловой системы Ext2(3,4)

В файловой системе Ext2 присутствует поддержка дополнительных флагов для файлов, повышающих безопасность системы. Ядро Linux позволяет работать со следующим набором атрибутов:

- `A` — `Atime`. Система не модифицирует поле `access time` для данного файла;
- `S` — `Sync`. Система фиксирует все изменения, происходящие в данном файле на физическом диске, синхронно с приложением, изменяющим данный файл;
- `a` — `append`. Система позволяет открывать данный файл только для его дополнения и не позволяет никаким процессам перезаписывать или усекать его. Если данный атрибут применяется к каталогу — процесс может создавать или модифицировать файлы в этом каталоге, но не удалять их;
- `i` — `immutable`. Система запрещает любые изменения данного файла. Если данный атрибут применяется к каталогу, то процессы могут модифицировать файлы, уже содержащиеся в данном каталоге, но не могут удалять их или создавать новые;
- `d` — `no dump`. Программе, создающей дампы системы, дается указание игнорировать данный файл во время создания резервной копии;
- `c` — `compress`. Система использует прозрачную компрессию для данного файла;
- `s` — `secure deletion`. Удаление такого файла сопровождается записью блоков диска, на которых он располагался, нулями;
- `u` — `undelete`. Когда приложение запрашивает файл на удаление, система должна сохранить его блоки на диске, чтобы потом его можно было восстановить.

Несмотря на то, что файловая система поддерживает приведенный набор атрибутов, у ядра и различных приложений остается выбор, учитывать или не учитывать их.

Флаг `A` или `Atime` для определенных файлов может дать некоторую прибавку производительности, т. к. избавляет систему от необходимости постоянно обновлять поле `access time` для этих файлов каждый раз, когда их открывают на чтение. Атрибут `S` или `Sync` увеличивает надежность сохранения данных ценой некоторой потери производительности системы.

Команды для установки и чтения атрибутов в Ext2

Есть две утилиты, специально предназначенные для установки и чтения данных атрибутов: `chattr` и `lsattr`.

Команда `chattr` устанавливает и снимает флаги:

- `chattr +Si test.txt` — установить флаги `sync` и `immutable` для файла test.txt;
- `chattr -ai test.txt` — убрать флаги `append-only` и `immutable` у test.txt;
- `chattr =aiA test.txt` — установить ограничение на использование только флагов `a`, `i` и `A`.

Команда `lsattr` выводит список файлов и каталогов с атрибутами и функционально напоминает команду `ls`.

Команда `lsattr -a test*`, например, выдаст на экран:

```
---i-------- test.conf
----a------- test.log
------------ test.txt
```

Защита файлов с помощью атрибутов файловой системы не обеспечивает стопроцентной гарантии защищенности системы. Конечно, атрибуты `a` и `i` запрещают изменение защищенных файлов даже процессами, владельцем которых является root, однако в обычных обстоятельствах ничто не мешает пользователю root снять эти флаги. Тем не менее есть возможность решить эту проблему.

Утилита lcap позволяет конфигурировать параметры ядра, в том числе те, которые определяют работу файловой системы Ext2 с расширенными атрибутами. Вот наиболее важные вызовы lcap, которые нас интересуют:

- `lcap CAP_LINUX_IMMUTABLE` — запрещает процессам root изменять флаги `a` и `i`;
- `lcap CAP_SYS_RAWIO` — запрещает низкоуровневый доступ к блочным устройствам, таким как диски, чтобы предотвратить изменение флагов через прямой доступ к файлам.

Пароли и шифрование

Стандартным атрибутом безопасности системы в наше время является пароль. Вот наиболее общие рекомендации по выбору паролей:

- длина пароля должна быть не менее 8 символов;
- пароль должен состоять из букв, набираемых в разных регистрах, символов типа `# $ @ / . ,` и цифр;
- не рекомендуется использовать что-либо обозначающие слова;
- желательно периодически изменять пароли.

Шифрование паролей в семействе Linux осуществляется по одностороннему алгоритму DES (Data Encryption Standard, стандарт шифрования данных). Хэши паролей затем сохраняются в файле /etc/shadow. Алгоритм DES практически исключает возможность расшифровки /etc/shadow для получения паролей. Однако наличие у взломщика файла /etc/shadow значительно облегчает подбор пароля пользователя программами типа John the Ripper. На современных машинах пароль из шести символов эта программа подбирает за пару часов. PAM-модули (такие как MD5 или подобные) обеспечивают различные алгоритмы шифрования паролей.

Протоколы шифрования трафика

Перечислим методы и протоколы для безопасной передачи данных в Интернете:

- SSL — Secure Sockets Layer, метод шифрования, разработанный Netscape для обеспечения безопасности в сети. Он поддерживает несколько различных протоколов шифрования и обеспечивает идентификацию на уровне как клиента, так и сервера. SSL работает на транспортном уровне и создает безопасный шифрованный канал данных. Чаще всего он используется при посещении пользователем защищенного Web-узла;
- S-HTTP — интернет-протокол, реализующий сервис безопасности;
- S/MIME — Secure Multipurpose Internet Mail Extension, стандарт шифрования, применяемый в Интернете для электронной почты и сообщений других типов.

SSH

SSH (Secure Shell) — программа, позволяющая зарегистрироваться на удаленном сервере и иметь шифрованное соединение. SSH заменяет устаревшие и небезопасные утилиты rlogin, rsh и rcp. Применяемый протокол реализует шифрование с помощью открытого ключа как для соединения между двумя машинами, так и для опознавания пользователей. Существует также несколько бесплатных реализаций SSH-клиентов для Windows.

PAM

PAM (Pluggable Authentication Modules) — унифицированный метод идентификации. Практически все современные приложения, которые осуществляют идентификацию пользователя, имеют соответствующий модуль PAM. Это позволяет пользователю "на лету" изменять методы идентификации, требования, инкапсулировать все локальные методы идентификации без перекомпиляции программ.

Вот что можно делать с PAM:

- использовать различные алгоритмы шифрования для своих паролей;
- устанавливать лимиты на ресурсы для пользователей;
- "на лету" активизировать теневые пароли (shadow password);
- разрешать определенным пользователям регистрироваться только в указанное время и/или из заданного места.

CIPE

CIPE — криптографическая IP-инкапсуляция, шифрует данные на сетевом уровне. Шифруются пакеты, которые передаются между компьютерами в сети. CIPE можно также использовать при туннелировании (tunnelling) для создания виртуальных частных сетей (VPN, Virtual Private Networks). Преимущество низкоуровневого шифрования состоит в том, что оно разрешает прозрачную работу между двумя сетями, соединенными в VPN, без каких-либо изменений в программном обеспечении.

Kerberos

Kerberos — идентификационная система, разработанная по проекту Athena в Массачусетском технологическом институте (MIT). Kerberos представляет собой сервер идентификации, услугами которого могут пользоваться компьютеры, подключенные к сети. Таким образом, нет необходимости заводить на всех компьютерах учетную запись пользователя. Очень часто применяется при модемном соединении провайдерами, имеющими несколько удаленных площадок.

CFS и TCFS

CFS — криптографическая файловая система. Это метод шифрования всей файловой системы, позволяющий пользователям сохранять в ней зашифрованные файлы. Метод использует NFS-сервер, запущенный на локальной машине.

TCFS — прозрачная криптографическая файловая система — улучшенный вариант CFS, поскольку более интегрирована с файловой системой и, таким образом, прозрачна для всех пользователей, работающих с зашифрованной файловой системой.

Безопасность ядра

Поскольку ядро контролирует поведение компьютера в сети, очень важно, чтобы оно было максимально защищено от взломов. С этих позиций крайне желательно иметь последние стабильные версии ядер, а при компиляции ядра максимально исключить ненужные вам опции и драйверы устройств.

Устройства ядра

Устройства /dev/random и /dev/urandom служат для получения случайных чисел в любой момент времени. Эти числа используются в генераторах PGP-ключей (Pretty Good Privacy — общедоступная система кодирования информации с открытым ключом), SSH-вызовах и других аналогичных приложениях.

Устройство /dev/random — высококачественный генератор случайных чисел, основанный на временно-зависимых параметрах системы.

Устройство /dev/urandom работает аналогично, но быстрее и менее надежно, поэтому если скорость некритична, то лучше генерировать псевдослучайные числа с помощью /dev/random.

Сетевая безопасность

Все наверняка слышали, как кто-то по сети взломал Web-сервер и испортил его содержимое или украл номера кредиток. Таких случаев становится все больше, поэтому исключительно важно заботиться о сетевой безопасности. Не следует, однако, забывать, что атака может проистекать в равной мере как из Интернета, так и из внутренней сети фирмы, поэтому крайне неразумно защищать сеть от атак снаружи и ничего не предпринимать для защиты от взлома изнутри.

Packet Sniffers

Один из наиболее общих методов взлома сетевых машин — применение *снифферов* (Packet Sniffer — в дальнейшем просто сниффер — программа, позволяющая перехватывать сетевые пакеты, предназначенные для других компьютеров. Первоначально использовалась для анализа сетевого трафика) с уже взломанного компьютера вашей сети. Эта программа перехватывает все Ethernet-пакеты сети и сканирует их на наличие слов Password, Login или su. Таким образом, без особых усилий взломщик получает множество паролей для систем, которые он даже и не пробовал пока взламывать. Поэтому крайне нежелательны сетевые сервисы, передающие пароли в незашифрованном виде. Этот способ взлома напрямую связан с обеспечением физической безопасности, т. к. посторонний может просто принести с собой ноутбук и подключиться с его помощью к внутренней сети фирмы.

SSH или другие методы шифрования паролей сводят к нулю эффективность этого способа взлома.

Системные сервисы

Прежде чем подключить систему к сети, следует подумать, какие сервисы будут предоставляться системой. Чем меньше запущенных сервисов, тем меньше вероятность взлома системы. Можно также ограничить список компьютеров, для которых разрешен доступ к сервисам вашего компьютера, прописав эти компьютеры в файле /etc/hosts.allow. Для запрещения доступа "подозрительных" систем следует использовать файл /etc/hosts.deny. Также проверьте ваши каталоги /etc/rc.d/rcN.d на наличие запуска сервисов, которые вам не нужны.

Однако нельзя огульно выбросить все, что, как вам кажется, не используется. К примеру, удаление сервиса в файле /etc/services приводит к тому, что локальный клиент также не сможет работать с этим сервисом.

DNS

Поддержка достоверной DNS-информации обо всех компьютерах сети также помогает повысить безопасность. Зачастую при атаке осуществляется подмена DNS-информации, что облегчает взломщику проникновение в систему.

identd

Программа identd фиксирует информацию о том, какой пользователь какой TCP-сервис запускает. С точки зрения повседневной жизни задача, вроде бы, бесполезная, однако подобная информация может пригодиться при анализе взлома системы.

Сетевые сканеры

Существует целый класс программных пакетов, которые выполняют сканирование портов и сервисов в компьютерных сетях. Сетевые сканеры нужны администратору системы для определения уязвимых мест системы, однако никто не мешает вос-

пользоваться ими и злоумышленнику. Наиболее известными, хотя уже и устаревшими, представителями этого класса программ являются SATAN и ISS. SATAN (инструмент администратора безопасности для анализа сетей) — это сканер портов с Web-интерфейсом. Он может быть полезен для выполнения легкой, средней или тщательной проверки машины или сети машин. ISS (сканер безопасности Интернета) также является сканером портов. Он быстрее, чем SATAN, но предоставляет меньше информации.

Электронная почта

Один из наиболее важных сетевых сервисов — сервер электронной почты. К сожалению, это наиболее часто атакуемый взломщиками сервис, поскольку он уязвим просто из-за огромного числа выполняемых задач и необходимых привилегий. Поэтому крайне желательно обновлять свой сервер электронной почты.

"Отказ в предоставлении доступа"

Очень популярный вид атаки. Смысл ее напоминает пословицу — "Сам не гам, и другому не дам". Взломщик пытается искусственно загрузить некоторые сервисы настолько, чтобы они не могли отвечать на запросы или запрещали доступ к вашей машине законным пользователям. Имеется несколько разновидностей такой атаки:

- SYN flooding — сетевая атака "отказ в предоставлении доступа". Использует преимущества "лазейки" (loophole) в методе создания TCP-соединения. Последние версии ядер Linux имеют несколько конфигурационных настроек для предотвращения SYN Flooding-атак;
- Ping flooding — простая грубая реализация атаки "отказ в предоставлении сервиса". Взломщик посылает компьютеру "поток" ICMP-пакетов. Если атака происходит с компьютера с большей полосой пропускания, чем у вашего компьютера, или с нескольких компьютеров одновременно, то ваша машина будет лишена возможности посылать что-либо в сеть. При вариации этой атаки, называемой "smurfing", на определенный сервер посылается поток ICMP-пакетов с обратным IP-адресом вашей машины;
- Ping of Death — атака, учитывающая тот факт, что поступающие ICMP-пакеты ECHO REQUEST могут быть больше, чем может вместить структура данных ядра, которая сохраняет эту информацию. Из-за приема единичного большого (65 510 байтов) ping-пакета многие системы зависали, отсюда и название атаки;
- Teardrop/New Tear — атака, основанная на ошибке, присутствующей в коде фрагментации IP в Linux- и Windows-платформах. Она была исправлена еще в ядре версии 2.0.33.

SELinux/AppArmor

SELinux (*Security-Enhanced Linux* — Linux с улучшенной безопасностью) — реализация системы принудительного контроля доступа, которая работает параллельно с классической системой. Входит в стандартное ядро Linux. Для функ-

ционирования SELinux требуются модифицированные версии некоторых утилит (ps, ls и др.), которые обеспечивают поддержку новых функций ядра, и поддержка файловой системы.

В отличие от классической системы контроля доступа, где предоставление ресурсов основывается на правах доступа пользователя, в SELinux права доступа определяются системой при помощи специально определенных политик. Политики работают на уровне системных вызовов и применяются самим ядром. SELinux действует после отработки классической модели безопасности. Таким образом, при помощи SELinux нельзя разрешить то, что запрещено через права доступа пользователей/групп. Политики описываются при помощи специального языка описания правил доступа. В большинстве случаев правила SELinux "прозрачны" для приложений, и не требуется никакая их модификация.

При помощи SELinux можно организовать несколько типов политик. Самый простой с точки зрения поддержки тип политики — так называемая "целевая" политика, разработанная в Fedora. В рамках политики описано более 200 процессов, которые могут выполняться в операционной системе. Все, что не описано "целевой" политикой, выполняется с типом unconfined_t. SELinux не защищает процессы, работающие с этим типом. Таким образом, все сторонние пользовательские приложения будут без всяких проблем работать в системе с "целевой" политикой в рамках классической системы контроля доступа.

Также в состав некоторых дистрибутивов входит политика с многоуровневой моделью безопасности (поддержка модели Bell LaPadula).

Третий вариант политики — "строгий". Тут действует принцип "что не разрешено, то запрещено".

SELinux был разработан Агентством национальной безопасности США и затем передан разработчикам открытого кода.

AppArmor — защита, основанная на политиках безопасности (профилях), которые определяют, к каким системным ресурсам и с какими привилегиями может получить доступ то или иное приложение. В AppArmor включен набор стандартных профилей, а также средства статического анализа и инструменты, основанные на обучении, позволяющие ускорить и упростить построение новых профилей. Для работы требует модификации ядра Linux.

Изначально программа была разработана фирмой Immunix. После ее приобретения компанией Novell инструмент был открыт под лицензией GNU GPL и включен в openSUSE и Ubuntu.

Безопасность NFS

Система NFS позволяет серверам предоставлять целые файловые системы для других машин со встроенной в ядро поддержкой NFS. В экспортируемых файловых системах существуют довольно ограниченные возможности реализации безопасности. Если вы вынуждены использовать NFS, прежде всего убедитесь, что предоставляете доступ только тем машинам, которым это действительно нужно. Никогда не экспортируйте полностью ваш корневой каталог.

Firewall

Firewall (брандмауэр, сетевой экран) с помощью определенного набора правил ограничивает прохождение информации как внутрь, так и за пределы вашей локальной сети. Обычно компьютер, выполняющий роль брандмауэра, соединен с Интернетом и вашей локальной сетью, и доступ к Интернету из локальной сети выполняется только через него. Брандмауэр является полезным инструментом в обеспечении безопасности вашей сети. Однако не следует забывать о безопасности только из-за того, что у вас установлен брандмауэр. Если нарушитель прорвался через брандмауэр или действует изнутри сети, у него будет огромное поле деятельности.

Существует много типов и методов организации брандмауэра. Более подробную информацию вы получите, прочитав великолепную книгу "Брандмауэры в Linux".

Антивирусная защита

Хотя защищенность Linux в плане вирусных атак, троянских программ и закладок на несколько порядков выше, чем Windows, тем не менее антивирусная защита все же необходима.

Необходима она не столько самому Linux, сколько файлам пользователей и почтовым сообщениям. Не секрет, что основная ниша Linux-машин в офисах — это Internet-сервер и файловый сервер, а клиенты — это компьютеры с операционной системой Windows. На момент написания книги Антивирус Dr Web знает 1 861 304 вируса.

Большое распространение получили вирусы, передающиеся через почту. Для борьбы с этой заразой необходимо установить антивирус, который:

- проверяет "на лету" всю входящую-исходящую почту;
- проверяет файлы (желательно "на лету").

Помимо нескольких коммерческих антивирусов, например Dr Web, Антивирус Касперского, которые хороши, но и стоят прилично, уже существуют антивирусы, распространяемые под лицензией GPL. По адресу **cvs.souceforge.net/viewcvs.py/openantivirus/mini-faq/av-unix_e.txt** находится список антивирусов, работающих под UNIX/Linux. Пожалуй, самый известный из этих антивирусов — Clam AntiVirus — ClamAV.

Чем же интересен этот антивирус?

- Распространяется под лицензией GPL.
- Мультиплатформенный (UNIX/Linux, Windows).
- Проверяет файлы "на лету".
- Проверяет почтовые сообщения и ящики.
- Поддерживает проверку архивных файлов (RAR, ZIP, GZIP).
- Онлайн-обновление баз с поддержкой цифровой подписи.

Из личного опыта: базы вирусов обновляются 3–4 раза в сутки. Практически все "свежие" вирусы вносятся в базу в течение 2–3 часов. Проверка файлов и почты не тормозит компьютер.

Администрирование системы

После установки и настройки системы вы сделали ее настолько безопасной, насколько это было возможно. Теперь необходимо выполнить несколько действий, чтобы быть подготовленным на случай ее взлома или аварии.

Резервная копия системы

Администрированию и резервированию операционной системы посвящено много интересных публикаций, так что особо останавливаться на этом мы не будем. Однако рассмотрим несколько общих соображений.

В настоящее время существует несколько малобюджетных вариантов резервирования системы.

- Жесткие IDE-диски — сегодня приличный жесткий диск емкостью 1 Тбайт стоит порядка 90 долл. Как представляется, это небольшая плата за систему, которую отлаживали долго и упорно, а тем более, если в ней хранятся важные данные. Еще одно преимущество резервного копирования на жесткий диск — его можно сконфигурировать так, чтобы было достаточно подключить винчестер в сервер, и система практически сразу стала полностью работоспособной.
- Привод CD-RW — очень экономичный способ. При стоимости одного диска CD-R менее 30 центов, а диска CD-RW около доллара это весьма выгодный вариант, но все же предпочтительнее DVD-диски.
- Привод DVD-RW — позволяет резервировать большие объемы данных при малых затратах — однослойный DVD стоит 40 центов.
- Привод Zip — удобен тем, что существует внешний вариант исполнения. В настоящее время морально устарел.
- Привод Jazz — емкость диска порядка одного гигабайта, морально устарел.
- Накопители на магнитооптике — существуют различные модели с разной емкостью дисков. Довольно дороги.
- Ленточные накопители — модели с разной емкостью. Достаточно дороги.
- USB Flash-накопители — удобны для переноски, малогабаритны, скорость записи 5–15 Мбайт/с. Стоимость Flash-диска на 16 Гбайт порядка 30 долл. Целесообразны для сбора резервных копий в нескольких несвязанных сетях.

Сразу после создания резервных копий на лентах и других перезаписываемых носителях необходимо поставить защиту от записи. Сохраняйте ваши резервные копии в надежных недоступных местах. Периодически проверяйте восстанавливаемость резервной копии. Время от времени устраивайте "боевые учения" по восстановлению системы.

Режим резервирования

Существует несколько стратегий резервирования, и только вам решать, какая из них вам подходит. Достаточно универсальна следующая стратегия:

- в конце рабочей недели создают полную резервную копию системы;
- в течение недели проводят нарастающее резервирование системы, т. е. резервируют изменение данных по сравнению с прошлым днем;
- при особо важных изменениях в системе резервную копию делают немедленно.

Однако довольно часто выполняется резервирование не всей системы, а только особо важных данных (например, базы данных), но с совершенно другим интервалом (к примеру, каждый час).

Существуют специальные программные пакеты, позволяющие писать сценарии автоматизированного сохранения и восстановления данных. Для простых схем резервирования достаточно скриптов, написанных администратором системы.

Резервирование RPM-базы

При взломе системы нарушитель обычно модифицирует для своих нужд несколько файлов, устанавливаемых при инсталляции из пакетов RPM. Если есть подозрение на взлом системы, одним из первых действий будет проверка целостности этих файлов. Однако если работал опытный взломщик, велика вероятность того, что он подправит нужным ему образом базу установленных RPM-пакетов или, вообще, уничтожит ее. На функциональность системы отсутствие или повреждение базы установленных RPM-пакетов не влияет, но при этом теряется возможность проверки целостности установленных пакетов. Поэтому крайне желательно периодически копировать базу RPM (/var/lib/rpm/*) на съемный носитель и хранить его отдельно.

Для проверки целостности установленных пакетов можно воспользоваться командой

```
rpm -Va
```

Но не забывайте после установки или удаления пакетов обновлять резервную копию базы RPM.

Файлы регистрации

Первое, что делает опытный взломщик системы после успешного проникновения в нее, — это заметает следы. А поскольку ведутся специальные файлы регистрации пользователей в системе, подключений к сетевым сервисам и тому подобных событий, вполне очевидным действием нарушителя является уничтожение или модификация файлов журналов. Поэтому крайне важно сохранить эти файлы в неприкосновенности. Начинать нужно с ограничения списка пользователей, способных читать и писать в каталог /var/log.

Регулярно инспектируйте свои журнальные файлы. Большое число неудачных попыток регистрации или сканирование портов с одного и того же компьютера может свидетельствовать о попытке вторжения. Чтобы узнать, где ваш дистрибутив ведет системные журналы, нужно посмотреть в файл /etc/syslog.conf, который указывает syslog, куда записывать различные сообщения. Если вы заметили, что в журнальных файлах кто-то похозяйничал, необходимо определить, когда это началось и каких процессов касалось. Лучше всего в такой ситуации восстановить журналы с резервных копий и определить момент взлома.

Можно настроить syslog так, чтобы он отсылал копию наиболее важных данных на безопасную систему. Это не даст возможность взломщику скрыть свою деятель-

ность путем удаления информации о его действиях в системе. Более подробную информацию по syslog.conf можно найти на соответствующей странице помощи (man-странице).

Обновляйте операционную систему

Большинство систем Linux устанавливаются с компакт-дисков. Но жизнь не стоит на месте, выходят различные обновления и исправления программ. Например, к дистрибутиву Fedora только официальных обновлений за четыре месяца набралось более 700 Мбайт. И это далеко не все обновления! Однако в погоне за номером версии не стоит устанавливать самое свежее программное обеспечение. Есть одно неплохое правило: "Работает — не трогай!" Нашли в программе ошибку — обновите. Хотите получить самую свежую версию программы — подождите неделю-другую после ее выхода. Пусть на грабли наступают другие. За это время наверняка найдутся прорехи в безопасности системы или ошибки в программе. Однако после обнаружения ошибки в безопасности программы не затягивайте с ее обновлением — велик шанс, что вашу систему попытаются взломать, используя именно эту брешь.

Действия во время и после взлома системы

Если вы обнаружили, что ваша система взломана — не паникуйте и внимательно проанализируйте ситуацию. Поспешные действия еще никого до добра не доводили.

Нарушение безопасности

Проще всего обнаружить физический взлом системы или подключение к вашей сети. В каждой фирме есть служба безопасности, воспользуйтесь ее услугами или вызовите представителей правоохранительных органов.

Если обнаружено вторжение в сеть, первым делом отсоедините вашу сеть. Если это невозможно, запретите доступ из сети взломщика или заблокируйте пользователей в системе. После того как будет сделано что-либо из перечисленного (отсоединена сеть, запрещен доступ из сети взломщика или заблокированы его учетные записи), следует ликвидировать все его пользовательские процессы.

Некоторое время после этого необходимо отслеживать состояние системы, поскольку попытка взлома может повториться, причем необязательно от имени этого же пользователя или с того же сетевого адреса.

Взлом системы произошел

Взлом обнаружен. Что дальше?

Закрытие бреши

Если вы четко знаете, каким образом взломщик проник в систему, постарайтесь сразу же закрыть эту брешь. Предположим, взлом произошел через сервер Samba. Самый простой выход — завершить процесс и отправиться в Интернет искать ре-

шение проблемы. Как правило, существует обновленная версия программы или какой-либо список исправлений известных ошибок.

Но это еще не означает, что вы в безопасности. Проверьте сомнительные события во всех ваших журнальных файлах. Поищите более свежие версии ключевого программного обеспечения и обновите его.

Оценка повреждений

Оцените повреждения. Выясните, что было нарушено. Не исключено, что в результате взлома проще переустановить систему, чем пытаться восстановить ее. Правда, такие тяжелые повреждения встречаются редко.

Так как Linux достаточно легко инсталлировать, рекомендуется создать специальный конфигурационный файл kickstart, содержащий список установленных в системе пакетов, который затем используется при инсталляции системы. А конфигурационные файлы следует заранее переписать. Рекомендуется также восстановить систему из резервной копии, которая наверняка содержит важные данные. Однако нужно очень точно определить момент взлома системы, чтобы не случилось так, что восстановленная из резервной копии система уже содержит "закладки" взломщика.

Выслеживание взломщика

Допустим, нарушитель заблокирован, система восстановлена, откуда пришел взломщик определено. Однако не забывайте, что раз взломали вас, могут взломать и кого-то другого. Поэтому следует сообщить об атаке администратору системы, из которой была взломана ваша система (этого администратора можно найти с помощью базы internic). Пошлите ему описание процесса взлома, перечень нанесенных повреждений и приложите содержимое системных журналов с датой и временем событий. Как правило, система, откуда была произведена атака, оказывается тоже взломанной, но администратор об этом даже не подозревает.

Опытные взломщики используют большое количество промежуточных, посреднических систем. Поэтому не стоит сразу предъявлять претензии администратору системы, откуда произошел взлом. Как говорят китайцы: "Не теряйте лицо". Будьте очень вежливы с администраторами других систем при выслеживании взломщика, и они сделают все возможное для его поимки.

Ссылки

В Интернете существует очень много узлов, посвященных безопасности систем UNIX и Linux. Обязательно подпишитесь на списки рассылки по вопросам безопасности и анонсы свежих выпусков программ. В частности большой список русскоязычных рассылок, в том числе и по Linux, можно найти на сайте **www.subscribe.ru.**

Вот несколько полезных адресов:

- **www.rootshell.com** — сайт, полезный для изучения современных методов взлома;
- **www.netspace.org/lsv-archive/bugtraq.html/** — содержит советы в области безопасности;

- **www.aoy.com/Linux/Security/** — хороший узел по безопасности в Linux.

Вопросы безопасности затрагивают документы, находящиеся и по нижеприведенным ссылкам:

- **www.linuxdocs.org** — Network Administrators Guide (Руководство сетевого администратора);
- **linux.webclub.ru/books/linuxsos/index.html** — безопасность и оптимизация Linux. Редакция для Red Hat — русский перевод;
- **dc.internic.net/rfc/rfc2196.txt** — документ, посвященный политике безопасности системы;
- **www.consensus.com/security/ssl-talk-faq.html** — часто задаваемые вопросы по протоколу SSL;
- **www.kernel.org/pub/linux/libs/pam/index.html** — PAM-модули;
- **linux.webclub.ru/adm/attr_ext2.html** — Michael Shaffer. Безопасность файловой системы Ext2;
- **pw1.netcom.com/~spoon/lcap/** — Linux Kernel Capabilities Bounding Set Editor;
- **cvs.souceforge.net/viewcvs.py/openantivirus/mini-faq/av-unix_e.txt** — список антивирусов, работающих под UNIX/Linux;
- **www.clamav.org** — адрес официального сайта антивируса ClamAV;
- **www.linuxdocs.org** — содержит соответствующие HOWTO:
  - security-HOWTO — документ, посвященный безопасности операционной системы;
  - hacker-HOWTO — документ, посвященный взлому и защите ОС от взлома;
  - NFS-HOWTO — документ о настройке и использовании NFS — сетевой файловой системы;
  - Firewall-HOWTO — документ, посвященный настройке брандмауэра;
  - IP-Masquerade mini-HOWTO — организация маскарадинга.

Глава 9



RPM- и DEB-пакеты

Фирма Microsoft и Windows уже приучили нас, что установка любого ПО начинается с запуска программ Setup или Install. Затем, после вывода лицензионного соглашения (по которому фирма-производитель обязывает вас установить программное обеспечение только на один компьютер и, в свою очередь, сообщает, что не несет никакой ответственности за функционирование этого ПО), задается пара вопросов (куда и какие модули программного обеспечения установить) и все: "Программа установлена, перезагрузите, пожалуйста, компьютер". Достаточно быстро и просто.

Остаются, конечно, некоторые неясности: "Почему я не могу убрать лишнюю функциональность, зачем эта игра перезаписала мой DirectX 9 своим DirectX 7, для чего в системе столько DLL для разных версий Visual Basic". Но в общем, это намного проще, чем вручную копировать какие-то архивы, распаковывать их, править конфигурационные файлы, искать конфликты библиотек и версий, а в самом неприятном случае — получать ошибки компиляции или линковки и просматривать исходные тексты программ. Это еще один упрек Linux со стороны пользователей Windows. И как часто бывает, они не совсем правы. Да, в большинстве своем программы Linux поставляются в виде исходных кодов, упакованных в архив. Но так наиболее просто удовлетворить требование лицензии GNU, которая обязывает дистрибьютора программы в обязательном порядке предоставить потребителю ее исходный код.

Не следует также забывать, что программы разрабатываются не только для Linux, обычно их можно откомпилировать на многих UNIX-платформах, а в UNIX-мире стандарт де-факто для пакетов — так называемый "tarballs" — архивы, которые распаковываются утилитой tar (файлы с расширением tar) или gzip (файлы с расширением tar.gz). Поскольку большинство Linux-программ распространяется через Интернет, проще выложить на FTP-сервер и скачать оттуда архив только с исходным кодом программы, чем архив и с исходными кодами, и с откомпилированной программой. Кроме того, энтузиасты Linux, как правило, имеют привычку смотреть исходные коды программ, изменять их и компилировать так, как им нравится (включать поддержку команд определенного процессора, добиваться максимального уровня оптимизации, различной степени выдачи отладочной информации и т. д.).

Однако с приходом в мир Linux пользователей, которые не желают учить опции компилятора, помнить, какие библиотеки установлены в системе, ждать по полчаса, пока откомпилируется программа и т. п., остро возник вопрос о стандартизации процесса установки программ в Linux. На сегодняшний день есть, по меньшей ме-

ре, три или, если быть совсем точным, "три с половиной" способа установки программ.

- *Способ первый*, "старейшина" — программы распространяются в виде архивов исходных кодов *.tar.gz, которые необходимо распаковать и, в простейшем случае, откомпилировать командами `make`, `make install`.
- *Способ второй* — воспользоваться программой RPM (Red Hat Linux package management; менеджер пакетов Red Hat Linux) и, соответственно, пакетами RPM, содержащими уже откомпилированный код программ.
- *Способ "два с половиной"* — воспользоваться программой RPM и пакетами RPM с исходным кодом. Здесь два варианта: или получать исходный код пакетов, или делать из пакета с исходным кодом пакет с исполняемым кодом и устанавливать.
- *Способ третий* (разновидность второго) — менеджер пакетов, входящий в дистрибутив Linux Debian.

Возможно, найдется и какой-то другой способ инсталляции или менеджер пакетов. Мир Linux и Интернет настолько велики, что узнать или охватить все невозможно. Как уже упоминалось, значительная часть современных дистрибутивов тем или иным образом основаны на Red Hat Linux или Debian. По крайней мере, имеются утилиты, способные работать с пакетами формата RPM или DEB. Поэтому эта глава полностью посвящена RPM-, DEB-пакетам и менеджерам.

Система поддержки пакетов RPM

Во многом благодаря RPM, а также удобной программе инсталляции Linux, дистрибутив Red Hat Linux завоевал огромнейшую популярность. Рассмотрим вкратце основные особенности RPM.

Для системного администратора RPM предоставляет следующие возможности:

- модернизировать отдельные компоненты системы или набора пакетов, сохраняя их конфигурацию;
- получать информацию об используемых пакетом файлах;
- получать информацию о зависимостях пакетов (необходимых библиотеках и т. д.);
- проверять пакеты;
- выдавать раздельно пакеты в авторском виде и сделанные к ним добавки;
- автоматически обновлять пакеты (например, получать обновления с FTP-сервера).

Благодаря этому с помощью RPM можно устанавливать, обновлять и удалять пакеты единственной командой в текстовом режиме или несколькими щелчками мышью в графическом менеджере пакетов. Информация о пакете RPM содержится в его заголовке. Эти сведения при установке пакета добавляются в базу данных установленных пакетов, где содержится информация о том, где находится пакет, какие дополнительные (supporting) пакеты ему необходимы и установлены ли они. Знатоки Windows могут заметить, что централизованная база данных установленных пакетов очень сильно напоминает часто критикуемый реестр Windows. Сравнение, однако, поверхностно. На самом деле реестр Windows помимо списка установленных про-

грамм содержит в себе многочисленные системные настройки, без которых (повреждение или отсутствие реестра) не будет функционировать система в целом. Для Linux отсутствие или повреждение базы данных установленных пакетов вовсе не фатально. Как будет показано далее, базу данных всегда можно попытаться создать заново. Но не это главное. Отсутствие или повреждение базы данных никоим образом не сказывается на работоспособности системы — она полноценно функционирует. Могут возникнуть проблемы с обновлением или установкой пакетов, но их можно обойти с помощью специальных ключей программы RPM (принудительная инсталляция, отказ проверять зависимости и т. п.).

Принципы наименования пакетов

Имя пакета характеризует сам пакет, его версию, версию сборки исполняемых файлов (релиз) и архитектуру и задается в виде "имя_программы-версия-релиз.платформа" или "src.rpm".

Рассмотрим для примера пакет telnet-server-0.17-18.i386.rpm. По названию файла можно определить, что пакет содержит telnet-сервер версии 0.17, версия сборки файлов (релиз) 18 для данной версии пакета Red Hat Linux собрана для процессора Intel 80386 и выше, формат файла — RPM. Файл пакета, у которого вместо архитектуры (например, i586) стоит src, содержит в себе исходные тексты программы. Иногда встречается немного другая структура именования пакета, например apache-1.3.3-1.src.rpm. Здесь версия пакета состоит из трех цифр (1.3.3). На инсталляционных дисках Red Hat и на FTP скомпилированные пакеты хранятся в каталоге RPMS, а пакеты, содержащие исходный код, — в каталоге SRPMS.

В действительности структура пакета RPM сложнее. Вкратце можно сказать, что в пакете содержатся исполняемые и конфигурационные файлы, документация, все дополнительные файлы, напрямую связанные с пакетом, а также информация о том, куда должны устанавливаться файлы пакета, и какие другие пакеты необходимы для его функционирования. После успешной установки пакета информация о нем заносится в базу данных системы RPM.

Достоинства RPM

Достоинства RPM:

- удобная установка программ;
- возможность инсталляции по FTP;
- проверка системы на наличие компонентов, необходимых устанавливаемому пакету;
- простое удаление пакетов из системы. При этом осуществляется проверка зависимостей пакетов системы от удаляемого пакета;
- обновление (Upgrade) пакетов с контролем версии, запрет установки пакета с более ранней версией, чем установленный в системе (Degrade);
- просмотр информации о пакете: что делает, кто сделал, где взять, файлы, содержащиеся в пакете, и т. д.;

- наличие общей иерархии пакетов, с помощью которой просто определить, к какой категории программ относится пакет;
- обеспечение возможности определения принадлежности файла или каталога к пакету;
- комплексная проверка состояния пакетов в системе: что изменялось, что испортилось, что случайно удалили и т. д.;
- отсутствие необходимости перезагружать систему после инсталляции нового пакета. Пакет готов к эксплуатации сразу после установки.

Недостатки RPM

Пакет RPM имеет и недостатки:

- многие программы пакета обновляются позже, чем официально выходят версии программного обеспечения;
- отсутствие RPM для некоторых программ;
- централизованная база установленных пакетов.

Информация, содержащаяся в пакете

Каждый пакет RPM содержит в себе стандартный набор полей, которые характеризуют содержание пакета. Вот перечень полей, наиболее интересных для пользователя:

- Build Host — имя хоста, на котором собран пакет;
- Build Date — время сборки пакета;
- Change Log — краткий список изменений в программе по сравнению с предыдущими версиями;
- Copyright — копирайт владельца;
- Description — описание пакета, обычно 1–2 Кбайт текста;
- Group — группа/подгруппа программного обеспечения, к которому относится пакет, например Development/Languages;
- License — лицензия, по которой распространяется пакет. Для большинства программ, поставляемых в дистрибутиве, лицензия — GPL. Для большинства библиотек — LGPL;
- Name — имя программы, например apache;
- Version — версия программы;
- Release — релиз (версия сборки);
- RPM version — версия пакета RPM: для Red Hat Linux 7.*x* версия 4, для более ранних — версия 3;
- Size — размер в байтах;
- Source RPM — пакет с исходными кодами, на базе которого собирался бинарный пакет, например gcc-2.96-85.src.rpm;
- Summary — краткое, в одно-два предложения описание пакета, например The C Preprocessor;
- URL — Web-адрес разработчика программы;
- Vendor — сборщик пакета, например Red Hat, Inc.

Категории пакетов

Для удобства пользователей пакеты содержат в себе признак, указывающий, к какой категории программного обеспечения относится пакет (поле Group). Стандартная иерархия пакетов приведена на рис. 9.1.

Кратко расшифруем категории пакетов.

- Amusements — развлечения. К этому разделу обычно относятся игры и всякие бесполезные, но веселые программки (глаза, которые следят за курсором, котенок, бегающий по экрану, и т. п.):
  - Games — подраздел предназначен для игр;
  - Graphics — всякие забавные графические программы, в том числе хранители экрана (Screensavers).
- Applications — приложения. Раздел предназначен для пользовательских (в широком смысле) программ. Как правило, сюда помещаются программы общего назначения (редакторы, инженерные пакеты, средства мультимедиа):
  - Archiving — программы и утилиты архивации;
  - Communications — подраздел, содержащий все, что относится к связи. Здесь собраны разнообразные программы и утилиты для работы с модемами, факсами, ISDN, ATM, радиосвязью и многое другое;

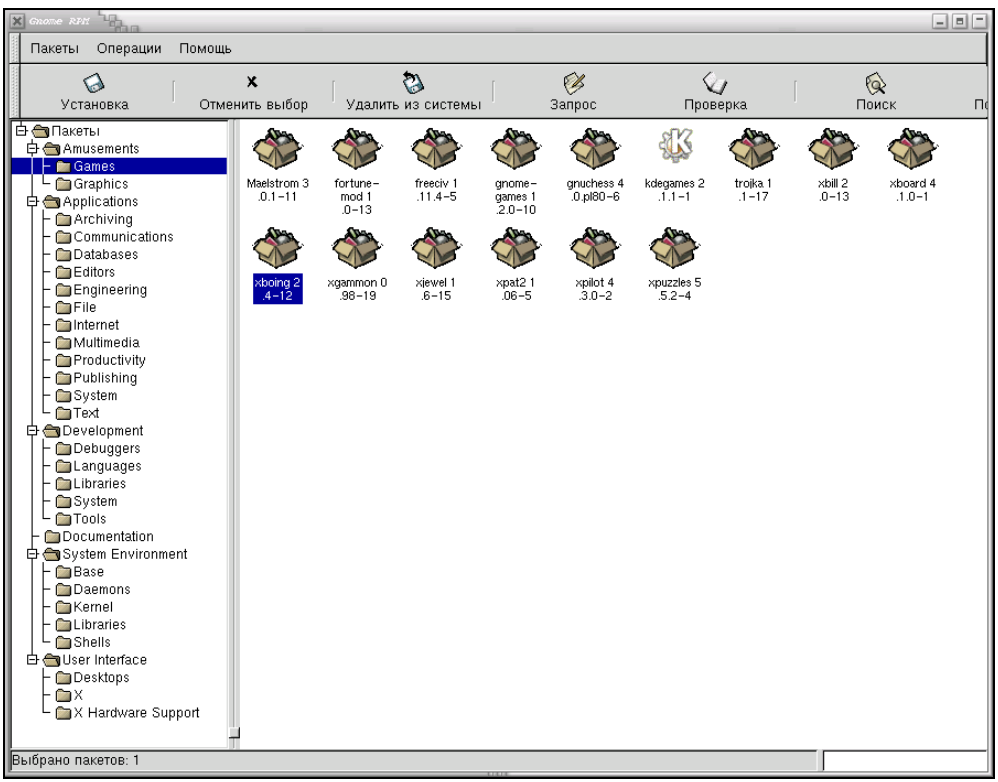


Рис. 9.1. Стандартная иерархия пакетов

- Databases — базы данных и разнообразные утилиты для взаимодействия с базами данных;
- Editors — редакторы. В этом разделе хранятся разнообразные редакторы, от очень простых консольных редакторов до графических монстров;
- Engineering — инженерные пакеты: редакторы схем, формул, химических соединений, чертежные пакеты и тому подобные приложения;
- File — утилиты для работы с файлами;
- Internet — программы, предназначенные для работы в Интернете: Web-браузеры, почтовые клиенты, клиенты ICQ и новостей, чатов и FTP;
- Multimedia — все для мультимедиа: проигрыватели CD, MP3-файлов, программы для просмотра телепередач и приема радиостанций, микшеры и т. д.;
- Productivity — подраздел для программ, позволяющих увеличить производительность труда: органайзеры, "напоминалки", картотеки и т. п.;
- Publishing — программы подготовки документов к печати: программы верстки, разметки и т. п.;
- System — системные программы. Некоторые из них предназначены только для администратора, другие представляют интерес только для пользователя;
- Text — программы и утилиты работы с текстом: поиск слов и фраз, замены и т. п.

❒ Development — раздел, полностью посвященный программированию и программистам (отладчики, компиляторы, библиотеки разработчика, различные утилиты):
- Debuggers — программы-отладчики;
- Languages — подраздел, посвященный языкам программирования, компиляторам, интерпретаторам;
- Libraries — библиотеки (по большей части библиотеки разработчика, не системные);
- System — системные утилиты;
- Tools — различный инструментарий программиста, не попавший в предыдущие подразделы.

❒ Documentation — раздел для документации, поставляемой отдельно от программ.

❒ System Environment — раздел системного окружения, наиболее ориентированный на ядро системы:
- Base — базовые пакеты;
- Daemons — подраздел исключительно для демонов (daemon, демон — программа, выполняющая некоторые системные функции или являющаяся сервером каких-то услуг, сервисов);
- Kernel — подраздел, предназначенный исключительно для ядра Linux как в двоичном виде, так и в исходных кодах;
- Libraries — системные библиотеки;
- Shells — разнообразные командные оболочки.

❒ User Interface — раздел пользовательского интерфейса. Вернее было бы назвать его разделом, посвященным X Window:
- Desktops — различные оконные менеджеры;
- X — пакеты, относящиеся к X Window;
- X Hardware Support — пакеты, ориентированные на конкретный тип видеокарт.

Команды консольного менеджера RPM

Раздел посвящен консольному менеджеру RPM и основан на содержимом man-страницы RPM. Понятно желание пользоваться графическими менеджерами пакетов — красиво, наглядно, удобно, просто, в конце концов. Но не следует забывать, всегда может случиться так, что у вас не будет возможности загрузить X Window (например, необходимо установить новую версию X Window), да и возможностей у RPM побольше, а ресурсов он потребляет несравненно меньше. Тем более что еще никто не отменял дистанционное администрирование, при котором вообще невозможно воспользоваться графическими пакетами.

Итак, остановимся на использовании RPM, менеджера пакетов от Red Hat. Можно выбрать один из следующих основных режимов:

- инициализация базы данных;
- пересборка базы данных;
- сборка пакетов;
- рекомпиляция пакетов;
- сборка пакетов из tar-архивов;
- запрос;
- показ полей запроса;
- установка;
- обновление;
- удаление;
- верификация;
- проверка подписи;
- повторная подпись;
- добавление подписи;
- установка владельцев и групп;
- показ конфигурации.

Общие опции

Общие опции, пригодные для всех режимов работы:

- `-vv` — выводить много отладочной информации;
- `-quiet` — выводить как можно меньше сообщений (как правило, выводятся только сообщения об ошибках);
- `-help` — вывести более детальную, чем обычно, справку о работе с RPM;
- `-version` — вывести одну строку, содержащую номер версии используемого RPM;
- `-rcfile <список_файлов>` — каждый из файлов из разделенного двоеточиями `<списка_файлов>` последовательно читается RPM на предмет конфигурационной информации. По умолчанию `<список_файлов>` выглядит как /usr/lib/rpm/rpmrc:/etc/rpmrc:~/.rpmrc. В этом списке обязательна только первая строка; все тильды будут заменены значением `$HOME`;
- `-root <каталог>` — файловая система с корнем в `<каталог>` для всех операций. Обратите внимание, это значит, что база данных также будет читаться и модифицироваться под `<каталог>` и все pre- и post-скрипты будут исполняться после `chroot()` в `<каталог>`;
- `-dbpath <путь>` — база данных RPM в `<путь>`;
- `-justdb` — обновить только базу данных, не файловую систему;
- `-ftpproxy <host>` — `<host>` как FTP-прокси (*см. разд. "Опции FTP/HTTP"*);
- `-httpproxy <host>` — `<host>` как HTTP-прокси (*см. разд. "Опции FTP/HTTP"*);
- `-ftpport <порт>` — `<порт>` как FTP-порт proxy-сервера (*см. разд. "Опции FTP/HTTP"*);

- `-httpport <порт>` — `<порт>` как HTTP-порт proxy-сервера (*см. разд. "Опции FTP/HTTP"*);
- `-pipe <cmd>` — перенаправляет вывод RPM на вход команды `<cmd>`.

Опции установки и обновления

Общая форма команды установки новых RPM:

```
rpm -i [опции-установки] <файл_пакета>
```

Общая форма команды обновления установленных RPM:

```
rpm -U [опции-установки] <файл_пакета>
```

Команда обновления установленных пакетов аналогична команде установки за исключением того, что `rpm` проверяет версию уже установленного пакета и, если она меньше версии нового пакета, происходит удаление старого пакета и установка нового. Иначе говоря, если пакет отсутствовал, эта команда устанавливает его, а если был установлен и имеет более раннюю версию, то версия пакета обновляется.

Примеры:

```
rpm -F [опции-установки] <файл_пакета>
```

или

```
rpm --freshen [опции-установки] <файл_пакета>
```

Эта команда обновляет пакеты, но только если в системе существуют их более ранние версии.

Допускается задание `<файл_пакета>` в виде FTP- или HTTP-адресов (например, http://www.freshmeat.net/Linux/ww-1.11-5.src.rpm). В этом случае перед установкой пакет будет получен с сервера, указанного в адресе. Подробную информацию о встроенной поддержке FTP/HTTP *см. в разд. "Опции FTP/HTTP" данной главы*.

Опции:

- `-force` — то же, что и комбинация `-replacepkgs`, `-replace-ffiilleess` и `-oldpackage`. Принудительная установка пакета, невзирая на наличие неудовлетворенных зависимостей или уже установленных пакетов, имеющих более позднюю версию;
- `-h`, `-hash` — выводить 50 раз знак `#` по мере распаковки архива с пакетом. В комбинации с `-v` придает читабельный вид. Подходит для автоматической установки пакетов, когда результат инсталляции выводится в журнальный лог-файл (log);
- `-oldpackage` — позволяет заменить новый пакет на более старый при обновлении (откатиться назад). Как правило, необходимость отката (roll-back) возникает в двух случаях: 1) при смене версий программного обеспечения (например, компилятор gcc поменял версию с 2.9*x* на 3.0), а новая версия имеет недостатки в функционировании ("подвисает", исчезли необходимые вам свойства программы и т. д.); 2) новая версия программного обеспечения конфликтует с уже установленными пакетами (не те версии библиотек, другой формат вызова модулей и т. п.);
- `-percent` — выводить процент готовности по мере распаковки архива с пакетом. Задумано для облегчения использования RPM из других утилит;

- `-replacefiles` — устанавливать пакеты, даже если они перепишут файлы из других, уже установленных пакетов;
- `-replacepkgs` — устанавливать пакеты, даже если некоторые из них уже присутствуют в системе;
- `-allfiles` — устанавливать или обновлять все файлы, определенные как missingok (согласно базе RPM — отсутствующие файлы в системе для данного пакета), даже если они уже существуют;
- `-nodeps` — не проверять зависимости перед установкой или обновлением пакета;
- `-noscripts` — не исполнять pre- и post-установочных скриптов;
- -notriggers — не исполнять триггер-скриптов, взведенных на установку данного пакета;
- `-ignoresize` — не проверять файловую систему на наличие достаточного свободного места перед установкой этого пакета;
- `-excludepath <путь>` — не устанавливать файлы, имена которых начинаются с `<путь>`;
- `-excludedocs` — не устанавливать никаких файлов, отмеченных как файлы документации (включает man-документацию и документы texinfo);
- `-includedocs` — устанавливать файлы документации. Это поведение по умолчанию;
- `-test` — не устанавливать пакет, просто проверить возможность установки и сообщить о потенциальных проблемах;
- `-ignorearch` — выполнить установку или обновление, даже если архитектуры бинарного RPM и машины не совпадают;
- `-ignoreos` — провести установку или обновление, даже если операционные системы бинарного RPM и машины не совпадают;
- `-prefix <путь>` — назначить префикс установки `<путь>` для переместимых пакетов;
- `-relocate <старый_путь>=<новый_путь>` — для переместимых пакетов преобразовывает в `<новый_путь>` файлы, которые должны были бы устанавливаться в `<старый_путь>`;
- `-badreloc` — для использования вместе с `-relocate`. Перемещает, даже если пакет неперемещимый;
- `-noorder` — не переупорядочивать список устанавливаемых пакетов. Обычно список переупорядочивается для удовлетворения зависимостей.

Опции удаления (деинсталляции)

Общая форма команды удаления пакета:

```
rpm -e <название_пакета>
```

Опции:

- `-allmatches` — удалить все версии пакета, отвечающие `<название_пакета>`. Обычно если `<название_пакета>` соответствует нескольким пакетам, то они не удаляются и выдается сообщение об ошибке;
- `-noscripts` — не исполнять pre- и post-установочные скрипты;

- `-notriggers` — не исполнять триггер-скрипты, взведенные на удаление данного пакета;
- `-nodeps` — не проверять зависимости перед удалением пакетов;
- `-test` — не удалять, а только протестировать возможность удаления. Полезна в сочетании с опцией `-vv`.

Опции запроса

Общая форма команды запроса RPM:

```
rpm -q [опции-запроса]
```

Можно задать формат, в котором будет выводиться информация о пакете. Для этого служит опция `-queryformat` с последующей строкой формата. Форматы запроса представляют собой модифицированную версию стандартного форматирования `printf()`. Формат состоит из статических строк (которые могут включать стандартные escape-последовательности языка программирования C для переводов строки, табуляций и других специальных символов) и форматов по типу используемых в `printf()`.

Есть два набора опций для запроса — выбор пакетов и выбор информации.

Опции выбора пакетов

Запрос установленного пакета с именем `<название_пакета>`:

```
-q <название_пакета>
```

Опции:

- `-a`, `-all` — запрос всех установленных пакетов;
- `-whatrequires <capability>` — запрос всех пакетов, требующих `<capability>` для правильного функционирования;
- `-whatprovides <virtual>` — запрос всех пакетов, предоставляющих сервис `<virtual>`;
- `-f <файл>`, `-file <файл>` — запрос пакета, которому принадлежит файл `<файл>`;
- `-g <группа>`, `-group <группа>` — запрос пакетов из группы `<группа>`;
- `-p <файл_пакета>` — запрос (неустановленного) пакета `<файл_пакета>`. Файл `<файл_пакета>` может быть задан как FTP- или HTTP-адрес;
- `-specfile <spec_file>` — разбор и запрос `<spec_file>` так, как если бы это был пакет. Хотя не вся информация (например, списки файлов) доступна, этот тип запроса позволяет использовать RPM для извлечения информации из spec-файлов;
- `-querybynumber <num>` — запросить непосредственно запись базы данных номер `<num>`. Полезна для отладочных целей;
- `-triggeredby <имя_пакета>` — запрос всех пакетов, содержащих триггер-скрипты, активизируемые пакетом `<имя_пакета>`.

Опции выбора информации

Опции выбора информации выглядят так:

- `-i` — выводит информацию о пакете, включая название, версию и описание. Использует `-queryformat`, если таковой задан;
- `-R`, `-requires` — выводит список пакетов, от которых зависит данный пакет;

- `-provides` — выводит список сервисов и библиотек, предоставляемых данным пакетом;
- `-changelog` — выводит протокол изменений данного пакета;
- `-l`, `-list` — выводит список файлов, входящих в данный пакет;
- `-s`, `-state` — выводит состояние файлов в пакете (подразумевает `-l`). Каждый файл может находиться в одном из следующих состояний: нормальный, не установлен или заменен;
- `-d`, `-docfiles` — выводит список только файлов документации (подразумевает `-l`);
- `-c`, `-configfiles` — выводит список только конфигурационных файлов (подразумевает `-l`);
- `-scripts` — выводит специфические для данного пакета скрипты, используемые как часть процессов инсталляции/деинсталляции, если таковые есть;
- `-triggers`, `-triggerscripts` — показать все триггер-скрипты, содержащиеся в пакете, если таковые имеются;
- `-dump` — выводит информацию о файлах следующим образом: `path size mtime md5sum mode owner group isconfig isdoc rdev symlink`. Эта опция должна использоваться в сочетании, по меньшей мере, с одной из опций `-l`, `-c` или `-d`;
- `-last` — упорядочивает список пакетов по времени установки так, что наиболее свежие пакеты находятся в начале списка;
- `-filesbypkg` — показывает все файлы в каждом пакете;
- `-triggerscripts` — показывает все триггер-скрипты для выбранных пакетов.

Опции проверки

Общая форма команды проверки RPM:

```
rpm –V [опции-верификации]
```

или

```
rpm -y [опции-верификации]
```

или

```
rpm -verify [опции-верификации]
```

В процессе проверки пакета сведения об установленных файлах пакета сравниваются с информацией из оригинального пакета и из базы данных RPM. В числе прочих верификация проверяет размер, контрольную сумму MD5, права доступа, тип, хозяина и группу каждого файла. Обо всех несоответствиях сообщается. Опции выбора пакетов такие же, как и для инспекции пакетов.

Файлы, которые не устанавливались из пакета (например, файлы документации, исключенные из процесса инсталляции при помощи опции `-excludedocs`), молча игнорируются.

Крайне полезная опция для администратора, т. к. позволяет при сбое в системе обнаружить поврежденные файлы (конечно, не все, например, конфигурационные файлы или файлы, созданные пользователем, так проверить не удастся). В случае взлома системы можно вычислить, какие файлы взломщик модифицировал (например, login).

Опции в процессе верификации позволяют игнорировать:

- `-nofiles` — отсутствующие файлы;
- `-nomd5` — ошибки контрольной суммы MD5;
- `-nopgp` — ошибки подписи PGP.

Формат вывода — строка из восьми символов. Каждый из них показывает результат сравнения одного из атрибутов файла со значением, записанным в базе данных RPM. Точка обозначает, что тест прошел. Следующие символы говорят об ошибках некоторых тестов:

- 5 — контрольная сумма MD5;
- S — размер файла;
- L — ссылка (симлинк);
- T — время модификации;
- D — устройство;
- U — владелец;
- G — группа;
- M — права доступа (включает права доступа и тип файла).

Проверка подписи

Общая форма команды проверки подписи RPM:

```
rpm -checksig <файл_с_пакетом>
```

Эта команда проверяет встроенную в пакет PGP-подпись для подтверждения целостности и источника происхождения пакета. Информация о конфигурации PGP читается из конфигурационных файлов. Подробнее *см. в разд. "Подписи PGP"* данной главы.

Опции сборки пакетов

Общая форма команды построения пакета RPM:

```
rpm -bO [опции-сборки] <spec_файл>
```

или

```
rpm -tO [опции-сборки] <arc_файл>
```

Аргумент `-b` подходит в том случае, если для сборки пакета используется spec-файл. Если же команда `rpm` должна извлечь этот файл из архива gzip, задают аргумент `-t`. За первым аргументом следует `O`, указывающий, какие этапы сборки и упаковки должны быть выполнены:

- `-bp` — исполнить стадию `%prep` spec-файла. Обычно это включает в себя распаковку исходного кода и прикладывание к нему патчей (от англ. *patch* — патч, заплатка, исправление);
- `-bl` — проверить список. В секции `%files` spec-файла производится расширение макросов и проверка перечисленных файлов на существование;
- `-bc` — исполнить стадию `%build` spec-файла (предварительно исполнив стадию `%prep`). Обычно это сводится к исполнению некоего эквивалента `make`;
- `-bi` — исполнить стадию `%install` spec-файла (предварительно исполнив стадии `%prep` и `%build`). Обычно это сводится к исполнению некоего эквивалента `make install`;

- `-bb` — собрать бинарный пакет (предварительно исполнив стадии `%prep`, `%build` и `%install`);
- `-bs` — собрать только исходный пакет (предварительно исполнив стадии `%prep`, `%build` и `%install`);
- `-ba` — собрать бинарный (RPM) и исходный (SRPM) пакеты (предварительно исполнив стадии `%prep`, `%build` и `%install`).

Также можно задать следующие опции:

- `-short-circuit` — исполнить непосредственно указанную стадию, пропустив предшествующие. Допустима только с `-bc` и `-bi`;
- `-timecheck` — установить возраст для `timecheck` (0 — чтобы запретить). Это значение также можно установить, определив макрос `_timecheck`. Значение `timecheck` — максимальный возраст файлов (в секундах), пакуемых в пакет. Для всех файлов, которые старше этого возраста, будет выводиться предупреждение;
- `-clean` — удалить дерево, использованное для сборки, после того, как построены пакеты;
- `-rmsource` — удалить исходный код и spec-файл после сборки (можно использовать отдельно, например `rpm -rmsource foo.spec`);
- `-test` — не исполнять никаких стадий сборки. Полезно для тестирования spec-файлов;
- `-sign` — встроить в пакет PGP-подпись. Эта подпись может служить для проверки целостности и источника происхождения пакета. Подробную информацию *см. в разд. "Подписи PGP"* данной главы;
- `-builroot <каталог>` — назначить каталог `<каталог>` как корневой для сборки пакетов;
- `-target <платформа>` — при сборке пакета интерпретировать `<платформа>` как `arch-vendor-os` и соответственно установить макросы `_target`, `_target_arch` и `_target_os`.

Опции пересборки и перекомпиляции

Существуют два способа запуска RPM:

- `rpm -recompile <файл_исходного_пакета>`;
- `rpm -rebuild <файл_исходного_пакета>`.

При вызове любым из способов RPM устанавливает указанный исходный пакет и исполняет стадии `%prep`, `%build` и `%install`. Кроме того, `-rebuild` собирает новый бинарный пакет. После того как сборка закончена, удаляется дерево, использованное для сборки (как с опцией `-clean`), исходный код и spec-файл.

Подпись существующего RPM

Подпись RPM выполняется следующими командами:

```
rpm -resign <файл_бинарного_пакета>
```

Опция `resign` генерирует и вставляет новые подписи в указанные пакеты. Все существующие подписи из пакетов удаляются.

```
rpm -addsign <файл_бинарного_пакета>
```

Опция `addsign` генерирует и добавляет новые подписи в указанные пакеты. Все существующие подписи пакетов при этом сохраняются.

Подписи PGP

Чтобы реализовать возможность подписи, RPM должен быть настроен для запуска PGP. Для этого необходимо создать свою собственную пару из публичного и секретного ключей и настроить следующие макросы:

- `_signature` — тип подписи. В настоящее время поддерживается только PGP;
- `_pgp_name` — имя "пользователя", ключи которого вы хотите использовать для подписи ваших пакетов.

При сборке пакетов к командной строке добавляется опция `-sign`. У вас спросят пароль, и ваш пакет будет собран и подписан.

Опции пересборки базы данных

Общая форма команды перестроения базы данных RPM:

```
rpm -rebuilddb
```

Команда для построения новой базы данных:

```
rpm -initdb
```

Этот режим поддерживает только две опции: `-dbpath` и `-root`.

Опции FTP/HTTP

RPM содержит несложные клиенты FTP и HTTP для упрощения установки и изучения пакетов, доступных через Интернет. Файлы пакетов для установки, обновления и запроса можно указать в виде FTP- или HTTP-адреса:

```
ftp://<user>:<password>@hostname:<port>/path/to/packagee.rpm.
```

Если `<password>` опущен, пароль будет запрошен (по одному разу для каждой пары user/hostname). Если ни `<user>`, ни `<password>` не указаны, будет использован anonymous ftp. Во всех случаях осуществляется пассивная (PASV) пересылка по FTP.

Опции RPM с адресом FTP:

- `-ftpproxy <hostname>` — система `<hostname>` будет организована как proxy-сервер для всех пересылок, что позволяет выполнять FTP-соединения через firewall, использующий proxy для выхода во внешний мир. Эту опцию можно задать также настройкой макроса `_ftpproxy`;
- `-ftpport <port>` — задает номер TCP-порта, открываемого для FTP-соединений вместо порта по умолчанию. Эту опцию можно также задать настройкой макроса `_ftpport`.

Опции RPM с адресом HTTP:

- `-httpproxy <hostname>` — система `<hostname>` будет организована как proxy-сервер для всех пересылок, что позволяет осуществлять HTTP-соединения через firewall, использующий proxy для выхода во внешний мир. Эту опцию можно задать также настройкой макроса `_httpproxy`;

- `-httpport <port>` — задает номер TCP-порта, открываемого для HTTP-соединений вместо порта по умолчанию. Эту опцию можно также задать настройкой макроса `_httpport`.

Используемые файлы

Файлы, необходимые при работе с пакетом RPM:

- /usr/lib/rpm/rpmrc;
- /etc/rpmrc;
- ~/.rpmrc;
- /var/state/rpm/packages;
- /var/state/rpm/pathidx;
- /var/state/rpm/nameidx;
- /tmp/rpm*.

Примеры использования консольного менеджера пакетов RPM

В предыдущем разделе мы познакомились с опциями менеджера RPM. С легкостью установки программ в Windows не сравнить. Впрочем, пользователи вряд ли применяют даже десятую часть имеющихся опций, поэтому и не следует запоминать их все. Рассмотрим, что требуется на практике при работе с пакетами.

Установка пакетов осуществляется с помощью команды:

```
rpm -i <полное_имя_пакета>
```

или

```
rpm -i <полное_имя_пакета> <полное_имя_пакета> <полное_имя_пакета> …
```

Пример:

```
rpm -i cpp-2.96-85.i386.rpm
```

Таким образом, можно установить сразу несколько пакетов. Помимо удобства (сразу указывается список пакетов, и они устанавливаются сами) указание нескольких пакетов необходимо в том случае, если возникают неудовлетворенные зависимости. Попадаются пакеты, зависящие друг от друга. Без второго пакета не установить первый, а второй не устанавливается, т. к. требует установки первого. Простейшее решение — поставить пакеты командой:

```
rpm -i <полное_имя_пакета_1> <полное_имя_пакета_2>
```

Команда простая, работает хорошо, но если в системе уже установлен пакет, пусть и более ранней версии, вы получите предупреждение, а сам пакет не установится. Чтобы обновить пакет, используем команду

```
rpm -U <полное_имя_пакета>
```

которая обновляет пакет, если он уже установлен, или устанавливает, если его нет. Однако не всегда при обновлении необходимо устанавливать отсутствующий пакет. В таких случаях можно воспользоваться командой

```
rpm -F <полное_имя_пакета>
```

которая проверит, есть ли в системе соответствующий пакет, и если есть — обновит его. При желании устанавливать или обновлять пакеты можно прямо с сервера FTP.

Например, в локальной сети есть FTP-сервер с именем bluewater. Ваш администратор регулярно скачивает с FTP-сервера Red Hat обновления RPM и выкладывает их на FTP-сервер локальной сети. Вот команда, с помощью которой можно обновить свои пакеты (для определенности возьмем компилятор C++):

```
rpm -F ftp://bluewater/pub/linux/updates/redhat-7.1/cpp-2.96-85.i386.rpm
```

Однако у консольного менеджера RPM есть одна неприятная особенность: при успешном завершении операции он ничего не сообщает на консоль. В принципе, это не страшно, можно задать ключ `-h`, который выводит процент выполнения процедуры.

Пример:

```
rpm -ih cpp-2.96-85.i386.rpm
```

или

```
rpm -ivh cpp-2.96-85.i386.rpm
```

Если при работе с пакетом возникнут проблемы, RPM выдаст причину, по которой невозможно выполнить какое-то действие. При установке пакета это, как правило, уже упомянутые неудовлетворенные зависимости либо отсутствие необходимых библиотек или установленных пакетов (или они в системе есть, но не той версии). Подобные проблемы решаются просто — установите соответствующие пакеты или обновите их до необходимой версии. Впрочем, бывают и здесь свои трудности.

Рассмотрим еще один пример. У автора на компьютере стоял в свое время Red Hat 7.1, а в нем удобная система GNOME с менеджером окон Sawfish. Все хорошо функционирует, только есть одна проблема: при сборке пакета Sawfish сборщики (американцы, им простительно) напутали с кириллическими шрифтами (системное меню вместо кириллицы отображает знаки вопроса). Был найден пакет посвежее, в котором эта оплошность убрана, да еще и функциональности добавлено. Пакет содержал исходные коды, поэтому пришлось сначала собрать его в бинарном виде командой

```
rpm -rebuild Sawfish-1.0-1.src.rpm
```

После приблизительно пяти минут компиляции в каталоге /usr/src/redhat/RPMS/i386/ образовался пакет Sawfish, который был запущен на обновление командой

```
rpm -F Sawfish
```

А в результате получено сообщение: "Обновление пакета не произведено, поскольку в системе уже установлен пакет версии 0.36, которая больше, чем версия 1.02". По всей видимости, сборщики пакета что-то перепутали в его описании. Пришлось воспользоваться командой

```
rpm -U -force Sawfish-1.0-1.i386.rpm
```

которая принудительно обновляет пакет, не проверяя зависимостей. Ключом `-force`, однако, следует пользоваться осторожно, т. к. можно ненароком развалить всю систему.

Иногда встречаются сообщения другого рода. При обновлении, например, пакета с исходным кодом ядра Linux версии 2.4.2 пакетом, содержащим исходный код ядра Linux версии 2.4.3, было выдано сообщение, которое в переводе на русский

язык звучит так: "Не могу удалить каталог такой-то, потому что он не пуст". Однако пакет успешно обновился, а каталог, фигурирующий в сообщении, на самом деле был пуст. Так что не стоит сразу расстраиваться, достаточно часто сообщения, выдаваемые RPM, весьма безобидны.

Удаляются пакеты из системы элементарно, с помощью команды:

```
rpm -e <имя_пакета>
```

Обратите внимание — указывается только имя пакета. Если написать полное имя пакета, то RPM выдаст сообщение: "Такой пакет в системе не установлен". Немного нелогично, но так уж исторически сложилось: при установке необходимо указывать полное имя пакета, при удалении — только имя пакета без упоминания версии, релиза и т. п.

При удалении сперва проверяются зависимости, и пакет удаляется, если от него не зависит никакой другой установленный в системе пакет. В противном случае на экран выдаются имена пакетов, для функционирования которых нужен удаляемый пакет. Конечно, если вы все-таки решили удалить пакет, можно воспользоваться ключами `-nodeps` или `-force`, однако рекомендуется применять их с большой осторожностью.

Узнать версию пакета, установленного в системе, позволяет команда:

```
rpm -q <имя_пакета>
```

Например, на запрос `rpm -q cpp` может быть получен такой ответ: `cpp-2.96-85`.

Для получения расширенной информации о пакете необходимо выполнить команду:

```
rpm -qi <полное_имя_пакета>
```

Результат выполнения команды `rpm -qi cpp-2.96-85` на конкретном компьютере иллюстрирует листинг 9.1.

Листинг 9.1

```
Name    : cpp             Relocations: (not relocateable)
Version   : 2.96             Vendor: Red Hat, Inc.
Release   : 85              Build Date: Срд 09 Май 2001 21:04:50
Install date: Птн 31 Авг 2001 07:38:10  Build Host:
porky.devel.redhat.com
Group    : Development/Languages   Source RPM: gcc-2.96-85.src.rpm
Size    : 292618           License: GPL
Packager  : Red Hat, Inc. <http://bugzilla.redhat.com/bugzilla>
URL     : http://gcc.gnu.org
Summary   : The C Preprocessor.
Description :
Cpp is the GNU C-Compatible Compiler Preprocessor. Cpp is a macro proces-
sor which is used automatically by the C compiler to transform your pro-
gram before actual compilation. It is called a macro processor because it
allows you to define macros, abbreviations for longer constructs.
```

```
The C preprocessor provides four separate functionalities: the inclusion
of header files (files of declarations that can be substituted into your
program); macro expansion (you can define macros, and the C preprocessor
will replace the macros with their definitions throughout the program);
conditional compilation (using special preprocessing directives, you can
include or exclude parts of the program according to various conditions);
and line control (if you use a program to combine or rearrange source
files into an intermediate file which is then compiled, you can use line
control to inform the compiler about where each source line originated).

You should install this package if you are a C programmer and you use ma-
cros.
```

Команда

```
rpm -ql <полное_имя_пакета>
```

выдает список файлов пакета (и каталогов, куда они будут установлены).

Например, `rpm -ql cpp-2.96-85` выведет на экран список файлов, приведенный в листинге 9.2.

Листинг 9.2

```
/lib/cpp
/usr/bin/cpp
/usr/lib/gcc-lib
/usr/lib/gcc-lib/i386-redhat-linux
/usr/lib/gcc-lib/i386-redhat-linux/2.96
/usr/lib/gcc-lib/i386-redhat-linux/2.96/cpp0
/usr/lib/gcc-lib/i386-redhat-linux/2.96/tradcpp0
/usr/share/info/cpp.info-1.gz
/usr/share/info/cpp.info-2.gz
/usr/share/info/cpp.info-3.gz
/usr/share/info/cpp.info.gz
/usr/share/man/man1/cpp.1.gz
```

А что, если нужна обратная операция: по имени файла узнать, к какому пакету он принадлежит? Выполним следующую команду:

```
rpm -qf /usr/bin/mc
```

В результате получим имя пакета: `mc-4.5.51-32`.

Теперь о безопасности. Прежде чем устанавливать пакет, полученный через Интернет, крайне желательно его проверить, вдруг он поврежден?

Проверить PGP-подпись пакета позволяет команда:

```
rpm -checksig <имя_пакета>
```

Если ваша система — сервер или к компьютеру имеет доступ кто-то, в чьих действиях вы не уверены, необходимо регулярно проверять целостность установленных пакетов и зависимостей командой:

```
rpm -V gimp
```

В ответ можно получить, например, следующее:

```
.M......  /usr/lib/gimp/1.2/modules/libcolorsel_gtk.a
.M......  /usr/lib/gimp/1.2/modules/libcolorsel_triangle.a
.M......  /usr/lib/gimp/1.2/modules/libcolorsel_water.a
```

Результат говорит, что права доступа на эти файлы были модифицированы.

Проверить все пакеты, установленные в системе, можно командой:

```
rpm -Va
```

Результат работы команды иллюстрирует листинг 9.3.

Листинг 9.3

```
S.5....T c /etc/printcap
.M......  /var/spool/at/.SEQ
отсутствует /etc/rpm/macros.db1
.......T  /usr/share/pixmaps/netscape.png
SM5....T  /usr/X11R6/lib/X11/fonts/Speedo/encodings.dir
отсутствует /var/cache/ssl_gcache_data.dir
.M....G.  /dev/jsfd
......G.  /dev/tty0
.....U..  /dev/vcs3
.....U..  /dev/vcsa3
S.5....T c /etc/X11/fs/config
отсутствует /usr/share/ssl/certs/stunnel.pem
S.5....T c /etc/openldap/ldap.conf
```

Совет

Если вы применяете дистрибутив, использующий пакеты RPM, избегайте установки программ компиляцией из исходного кода (не из пакетов RPM). Поскольку программа компилируется и устанавливается вручную, информация в базу данных установленных RPM не попадает. Следовательно, велика вероятность, что при установке или обновлении какого-нибудь пакета вы нарушите зависимости для скомпилированной вами программы, и она не будет работать.

Помимо консольного менеджера RPM, существуют еще несколько утилит, предоставляющих текстовый интерфейс и позволяющих работать с пакетами формата RPM. Однако они имеют обычно значительно меньшую функциональность.

Система обновлений пакетов Yum

Yum — надстройка над RPM для решения следующих задач:

- поиск пакетов в репозиториях;
- установка пакетов из репозиториев (при необходимости — с разрешением зависимостей);
- обновление системы и удаление ненужных пакетов.

С помощью Yum можно автоматизировать установку и обновление систем в ваших сетях. Например, на сервере Yum настраивают так, чтобы он обновлялся из Интернета, а на остальных компьютерах в вашей сети в качестве репозитория выступает ваш сервер. Запускать Yum можно при помощи cron.

Поиск в репозиториях

Для поиска в репозиториях с помощью Yum можно воспользоваться одной из трех команд:

- `list` — просматривает названия пакетов и их версии в поисках совпадений. Например, для просмотра пакетов с названием mc выполните команду `yum list mc`;
- `search` — ищет в названии пакета и его описании указанную строку. Например, для поиска пакетов, связанных с KDE, выполните команду `yum search KDE`;
- `provides` — ищет пакеты, содержащие указанный файл. Например, для поиска пакетов с файлом libc выполните команду `yum provides libc`.

Все команды поиска поддерживают символы `?` и `*` (экранируйте их обратным слешем `\` для корректной обработки bash). К примеру, для поиска пакетов, начинающихся на lib, введите: `yum list lib\*`, а для поиска пакетов, содержащих файлы, которые находятся в каталоге /etc/httpd, выполните команду `yum provides /etc/httpd\*`.

Установка пакетов с помощью Yum

Для установки пакетов предназначена команда `install`. Например, `yum install mc` установит Midnight Commander.

Yum автоматически разрешает зависимости и предлагает устанавливать необходимые пакеты.

Обновление системы

Обновление системы осуществляется с помощью команды `update`. Команда `yum update` обновит всю систему, а команда `yum update mc` обновит *только* пакет mc.

Удаление пакетов

Удалить пакеты можно командой `remove`. Например, `yum remove mc` удалит пакет mc.

Информация о пакетах

Информацию о пакетах (их описание) в репозитории можно получить при помощи команды `info`.

Очистка кэша Yum

Со временем в кэше Yum накапливается всякий "мусор". Очистить кэш можно при помощи команды `clean`.

Помимо рассмотренных простых применений возможны и более сложные. Вот некоторые из них.

- `yum list updates` — выводит все пакеты, для которых в репозиториях Yum доступно обновление.
- `yum list updates after [date]` — выводит все пакеты, для которых в репозиториях Yum доступно обновление, собранное позднее чем `[date]`. Формат даты: ГОД-МЕСЯЦ-ДЕНЬ.
- `yum list updates last [days]` — выводит все пакеты, для которых в репозиториях Yum доступно обновление, собранное за последние `[days]` дней.
- `yum list installed` — выводит список всех установленных пакетов.
- `yum list extras` — выводит все установленные пакеты, которые недоступны в репозиториях.
- `yum clean packages` — удаляет все скачанные пакеты из кэша системы. После скачивания пакеты автоматически из кэша не удаляются.
- `yum clean headers` — удаляет все файлы, используемые Yum для определения доступности пакетов в репозиториях. Выполнение этой команды приводит к тому, что при следующем запуске Yum заново загрузит все списки пакетов из репозиториев.
- `yum clean oldheaders` — удаляет старые заголовки, которые Yum более не использует для определения доступности пакетов в репозиториях.
- `yum clean [all]` — выполняет `yum clean packages` и `yum clean oldheaders`, как описано ранее.

Midnight Commander

Помимо функций файлового менеджера, работы с архивами и множества других возможностей, Midnight Commander способен получить информацию из пакетов форматов RPM и DEB, установить или обновить пакет. Конечно, это не заменит полноценного менеджера пакетов, но быстро поставить или обновить несколько пакетов или посмотреть информацию о пакете также иногда бывает полезно. На рис. 9.2 изображено содержимое RPM-пакета, требуется только нажать клавишу <Enter> в нужном месте.

Для нас интересны виртуальные файлы и каталоги (они все пишутся прописными буквами):

- HEADER — содержит заголовок пакета — то, что можно получить командой `rpm -qi <имя_пакета>`;
- *INSTALL, *UPGRADE — если запустить на выполнение, Midnight Commander проинсталлирует или обновит этот пакет;
- /INFO — каталог с данными о пакете. Содержит виртуальные файлы с информацией, описывающей пакет.

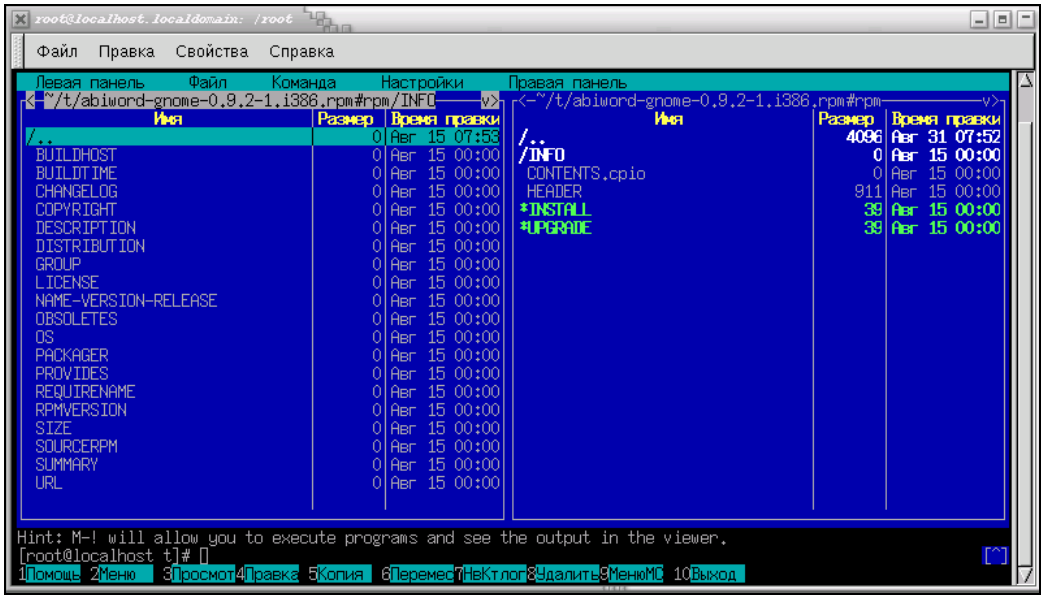


Рис. 9.2. Midnight Commander, работа с пакетами RPM

purp

Программа purp удобна для просмотра установленных пакетов, получения разнообразной информации, установки, удаления пакетов. Весьма полезная программа, по функциональности близка к RPM. На рис. 9.3 показано ее основное окно.

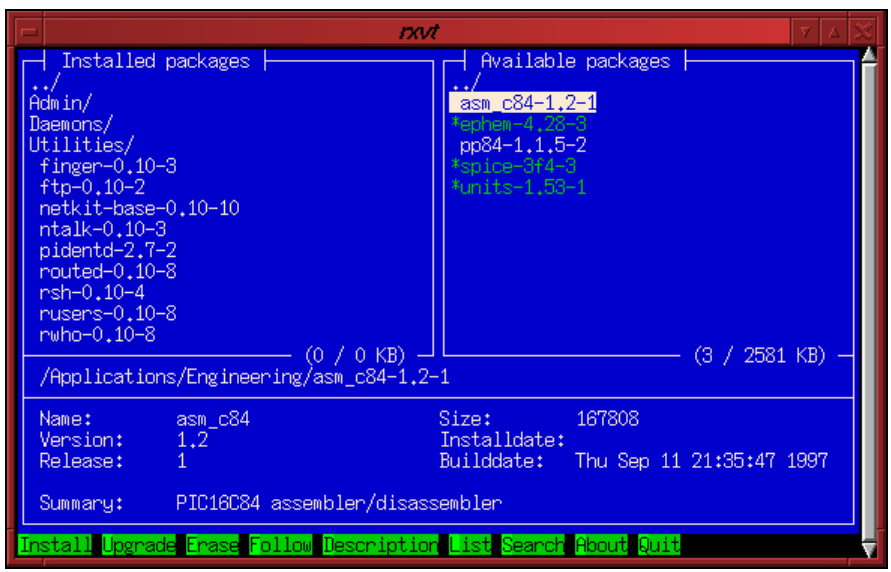


Рис. 9.3. Основное окно purp

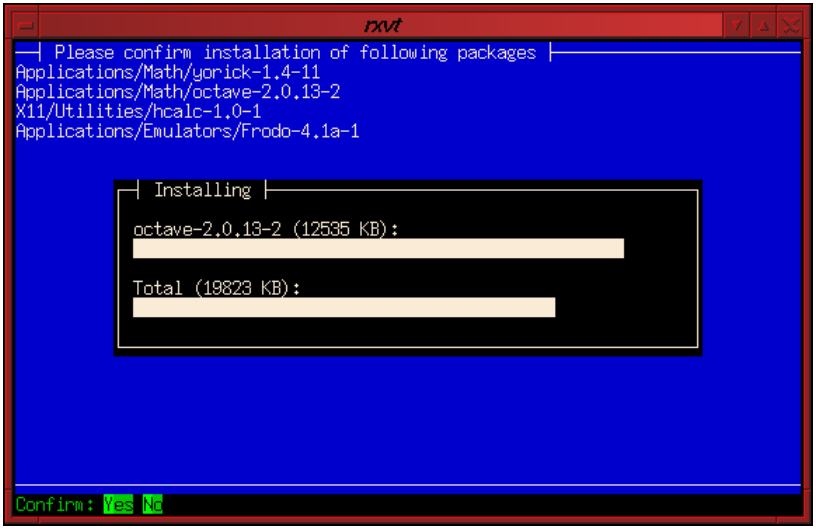


Рис. 9.4. Установка пакетов

Предназначена для тех, кто не хочет (или не может) работать в X Window, а пользоваться RPM по каким-то причинам не желает (типичный представитель — бывший пользователь DOS/Windows, для которого привычен и удобен Norton Commander). Процесс установки пакетов иллюстрирует рис. 9.4.

Помимо текстовых менеджеров пакетов RPM, существует несколько графических менеджеров.

Kpackage

Kpackage — это полнофункциональный графический интерфейс для менеджеров пакетов RPM, Debian, Slackware, BSD и KISS. Kpackage является частью рабочей среды K Desktop Environment и тесно интегрирован с файл-менеджером KDE (KFM). Практически все, что можно делать в консольном менеджере RPM, реализовано в Kpackage. Окно менеджера пакетов Kpackage приведено на рис. 9.5.

GnoRPM

Полнофункциональный, в целом довольно удобный менеджер пакетов, входящий в состав GNOME. Однако есть несколько неприятных моментов:

- при установке необходимо отметить соответствующие пакеты. Однако после установки отметки автоматически не убираются;
- если при установке обнаружены неудовлетворенные зависимости, то менеджер не предлагает их автоматического удовлетворения.

Окно менеджера пакетов GnoRPM приведено на рис. 9.6.

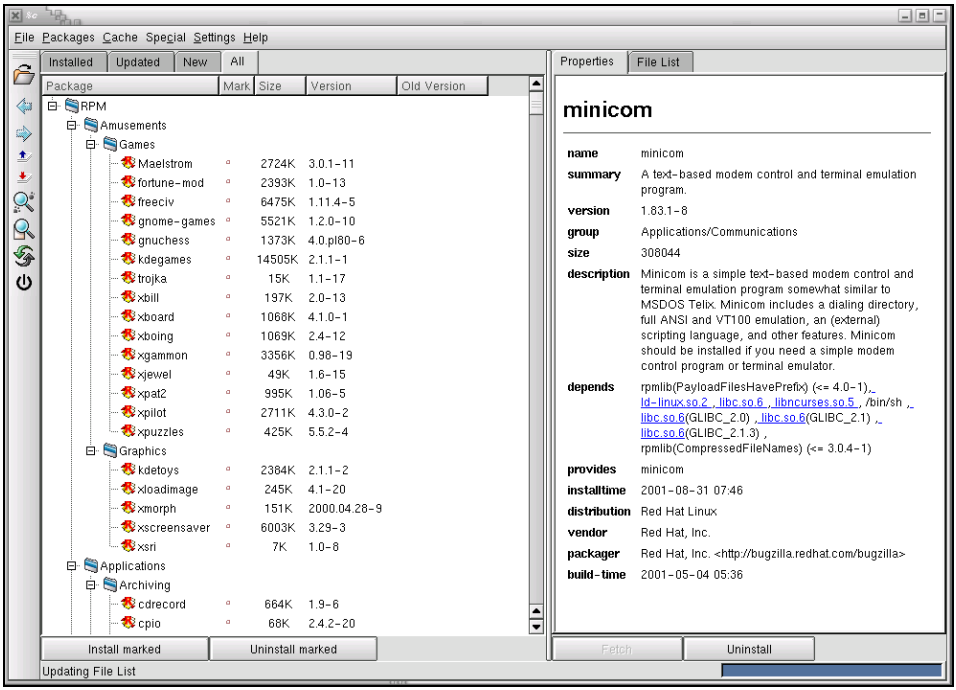


Рис. 9.5. Менеджер пакетов Kpackage

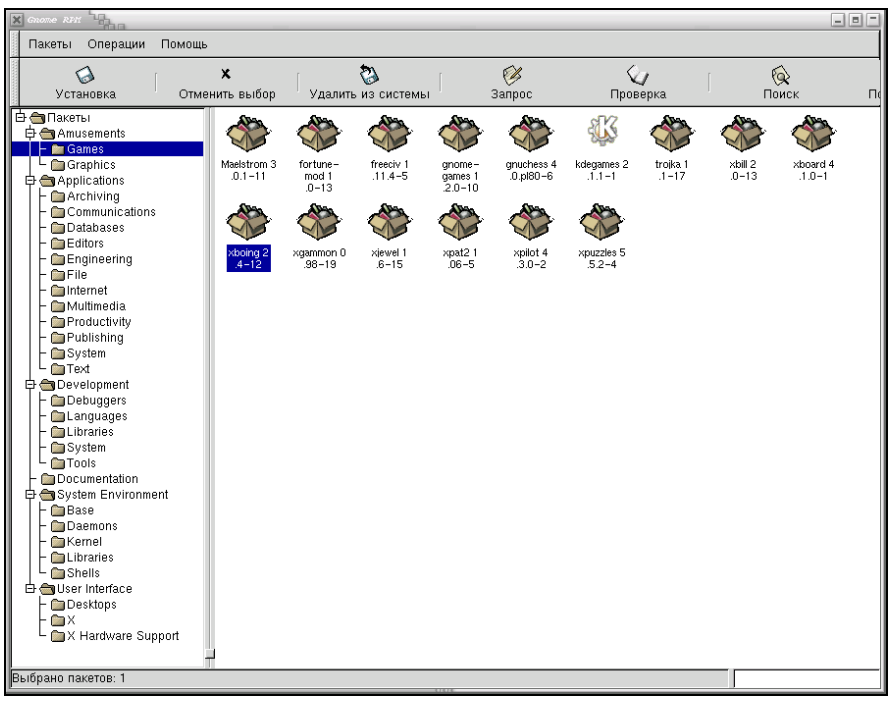


Рис. 9.6. Менеджер пакетов GnoRPM

Yumex

Yumex (Yum extender) — графическая надстройка над Yum. Полнофункциональный, удобный менеджер пакетов, входящий в состав многих дистрибутивов. Написан на языке Python. Домашняя страница программы — **fedorahosted.org/yumex/**.

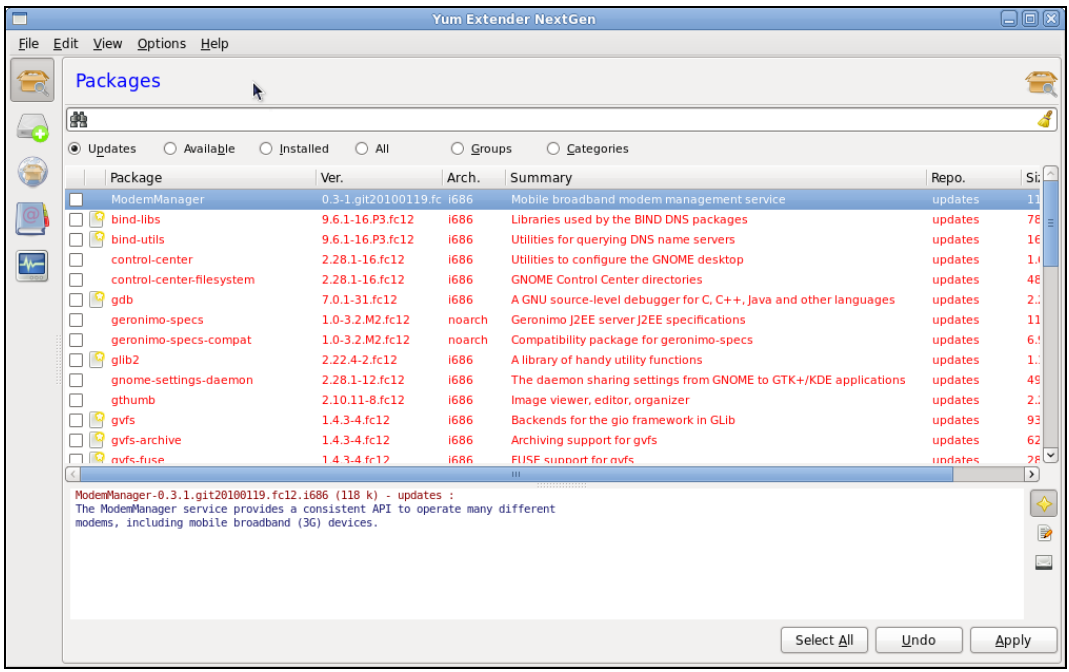


Рис. 9.7. Менеджер пакетов Yumex

Существуют также менеджеры glint, grpm, gtkrpm и много других. Однако наиболее распространенные — Kpackage и GnoRPM — входят в KDE и GNOME.

DEB-пакеты

Помимо RPM-пакетов, характерных для Red Hat Based дистрибутивов, достаточно широко распространены DEB-пакеты, используемые в дистрибутивах Debian. Пакет — это файл с расширением deb, содержащий программу и некоторую дополнительную информацию. Пакет представляет собой архив, в котором находятся три файла:

- ❒ debian-binary — специфичная информация;
- ❒ control.tar.gz — служебная информация (зависимости, скрипты);
- ❒ data.tar.gz — собственно файлы пакета.

Таким образом, DEB-пакет можно очень просто распаковать на любой системе.

Достоинства DEB

Достоинства DEB-пакетов:

- удобная установка программ;
- возможность инсталляции по FTP;
- проверка системы на наличие компонентов, необходимых устанавливаемому пакету;
- простое удаление пакетов из системы. При этом осуществляется проверка зависимостей пакетов системы от удаляемого пакета;
- обновление (Upgrade) пакетов с контролем версии, запрет установки пакета с более ранней версией, чем установленный в системе (Degrade);
- наличие общей иерархии пакетов, с помощью которой легко определить, к какой категории программ относится пакет;
- обеспечение возможности определения принадлежности файла или каталога к пакету;
- комплексная проверка состояния пакетов в системе (что изменялось, что испортилось, что случайно удалили и т. д.);
- отсутствие необходимости перезагружать систему после инсталляции нового пакета. Пакет готов к эксплуатации сразу после установки;
- DEB-пакеты при установке могут динамически настраиваться и при необходимости задавать вопросы пользователю, чего нет в RPM-пакетах.

Недостатки DEB

Пакет DEB имеет и недостатки:

- многие пакеты обновляются позже, чем официально выходят версии программного обеспечения;
- отсутствие DEB для некоторых программ.

Информация, содержащаяся в пакете

Каждый пакет DEB содержит в себе стандартный набор данных, которые характеризуют содержание пакета. Компоненты пакета:

- само приложение;
- описание приложения;
- список зависимостей приложения;
- скрипты для установки приложения;
- пользовательская документация.

Менеджеры DEB-пакетов

С DEB-пакетами могут работать многие программы, самые основные собраны в пакет APT (Advanced Package Tools). APT позволяет устанавливать пакеты из различных источников: CD-ROM, FTP, HTTP-сервера или из локального репозитория пакетов. Debian также предоставляет свободный доступ к архивам своих приложений на публичных FTP- и HTTP-серверах. Процесс установки пакетов сводится

к указанию адреса одного из этих серверов в настройках APT и запуску последнего с соответствующими опциями. APT сам отслеживает зависимости и выбирает для установки дополнительные пакеты.

Работа с APT

Перед началом работы с APT необходимо указать, где искать файлы пакетов. Каждое такое местоположение называется *источником*. Источником может быть CD-ROM, HTTP-, FTP-сервер или архив на вашем жестком диске.

APT просматривает список источников в файле /etc/apt/sources.list. Пример sources.list приведен в листинге 9.4.

Листинг 9.4

```
deb http://security.debian.org/ stable/updates main
deb http://http.us.debian.org/debian stable main contrib non-free
deb cdrom:[Debian GNU/Linux 3.0 r0 _Woody_ - Official i386 Binary-3 (20020718)]/
unstable contrib main non-US/contrib non-US/main
deb cdrom:[Debian GNU/Linux 3.0 r0 _Woody_ - Official i386 Binary-2 (20020718)]/
unstable contrib main non-US/contrib non-US/main
deb cdrom:[Debian GNU/Linux 3.0 r0 _Woody_ - Official i386 Binary-1 (20020718)]/
unstable contrib main non-US/contrib non-US/main
```

Добавить ссылку в sources.list можно в текстовом редакторе. Добавить ссылки на CD-ROM позволяет команда `apt-cdrom`:

```
apt-cdrom add
```

Обновление локального кэша пакетов

Кэш пакетов (package cache) — это полный список пакетов, имеющихся в дистрибутиве Debian. Он хранится на компьютере, и вам нужно следить за его своевременным обновлением. Каждый раз, когда вы хотите установить или обновить программное обеспечение, то должны в первую очередь обновить локальный кэш пакетов. Это гарантирует, что информация об устанавливаемых пакетах будет актуальной.

Для обновления кэша выполните следующую команду:

```
apt-get update
```

Просмотр существующих пакетов

Утилита apt-cache позволяет искать пакеты в локальном кэше по тексту, содержащемуся в описании пакета.

Например, команда

```
apt-cache search perl
```

выдаст информацию обо всех пакетах, связанных с perl.

Просмотр информации об отдельном пакете

Для получения детальной информации о пакете можно воспользоваться утилитой `apt-cache` с опцией `show`:

```
apt-cache show packagename
```

Например, если вы хотите просмотреть информацию по пакету apache, то должны ввести команду:

```
prompt$ apt-cache show apache
```

В моем случае она выдала текст, приведенный в листинге 9.5.

Листинг 9.5

```
Package: apache
Priority: optional
Section: web
Installed-Size: 748
Maintainer: Matthew Wilcox <willy@debian.org>
Architecture: i386
Version: 1.3.26-0woody1
Replaces: apache-modules
Provides: httpd

Depends: libc6 (>= 2.2.4-4), libdb2 (>= 2:2.7.7.0-7), libexpat1
(>= 1.95.2-6), mime-support, apache-common (>= 1.3.26-0),
apache-common (<< 1.3.27-0), perl5 | perl, logrotate (>=
3.5.4-1), dpkg (>> 1.9.0)

Suggests: apache-doc
Conflicts: apache-modules, libapache-mod-perl (<= 1.17-1), jserv (<= 1.1-3)
Filename: pool/main/a/apache/apache_1.3.26-0woody1_i386.deb
Size: 352814
MD5Sum: 728257f5de8d71e0d00701bdca9d452d
Description: Универсальный, высокопроизводительный HTTP-сервер. Один из
самых популярных HTTP-серверов в мире, Apache характеризуется модульной
структурой и возможностью динамической загрузки внешних модулей в процес-
се работы (runtime). Некоторые особенности сервера, являющиеся его силь-
ными сторонами, это: возможность гибкой настройки сервера; изменение ко-
личества процессов "на лету"; большое количество внешних модулей,
включающих в себя механизмы авторизации; грамматический разбор HTML; кон-
троль доступа со стороны сервера; эмуляция CERN httpd-метафайлов; возмож-
ность кэширования (proxy) и пр. Кроме этого, Apache поддерживает множест-
венные виртуальные подключения (multiple virtual homing). Отдельные
пакеты предоставляют Apache возможность работы с PHP3, mod_perl, Java
Servlet, Apache-SSL и пр. За детальной информацией обращайтесь на
http://www.apache.org/.
Task: web-server
```

Установка пакета

Пакет устанавливают при помощи утилиты `apt-get`:

```
apt-get install packagename
```

Удаление пакета

Для удаления пакета необходимо воспользоваться командой

```
prompt$ apt-get remove packagename
```

Обновление системы

Периодически возникает желание обновить систему последними стабильными версиями программ. Сделать это при помощи APT очень просто — достаточно выполнить следующие команды:

```
apt-get update
apt-get upgrade
```

Первая команда обновляет локальный кэш пакетов, вторая обновляет любые пакеты, уже установленные на вашей машине.

Aptitude

Aptitude — оболочка для Advanced Packaging Tool, имеющая командную строку и псевдографический интерфейс. Ее функции сводятся к установке и удалению пакетов и получению информации о них, отслеживании зависимостей и разрешении связанных с ними коллизий. Разработчики Debian рекомендуют ее в качестве основного средства для управления пакетами.

Функции Aptitude похожи на `apt-get`, но есть некоторые преимущества.

Поиск пакетов

Для поиска пакета достаточно выполнить команду `aptitude search keyword`, выдающую список всех пакетов, содержащих в названии ключевое слово. Результат работы команды — некая таблица, в столбцах которой приведена следующая информация: буква-признак (одна или две), имя пакета, краткое описание. Рассмотрим расшифровку букв-признаков:

- i (installed) — пакет установлен в системе;
- p (purge) — пакет не установлен или был полностью удален;
- c (clean) — пакет удален с сохранением конфигурационных файлов;
- v (virtual) — виртуальные пакеты.

Дополнительная буква-признак расшифровывается следующим образом:

- A (Auto) — пакет был установлен автоматически для разрешения зависимостей другого пакета;
- h (hold) — пакет "заморожен", т. е. он не будет обновляться при выполнении команд `upgrade` и `dist-upgrade`;
- u (unpacked) — пакет получен, распакован, но не установлен и не сконфигурирован;
- C (half-Configured) — установка пакета прекратилась на стадии конфигурирования;

- H (Half-installed) — установка пакета прекратилась на стадии инсталляции;
- B (Broken) — "поврежденный" пакет, содержащий ошибки или утративший свои зависимости.

Информация о пакете

Информацию о пакете выводит команда `aptitude show имя-пакета`.

В результате вы получите следующие сведения: имя пакета, его состояние (установлен или нет), устанавливался пакет автоматически или нет, версия пакета, приоритет, категория пакета (к какому типу пакетов отнесли его разработчики дистрибутива), кто собирал этот пакет, объем пакета в распакованном виде, список зависимостей пакета (обязательные и рекомендуемые к установке), с какими пакетами конфликтует, текстовое описание пакета.

Установка пакетов

Установка выбранных пакетов осуществляется командой

```
aptitude install имя_пакета
```

При этом происходит просмотр репозиториев, описанных в конфигурационном файле (/etc/apt/source.list), скачивание пакета из сети, помещение его в локальный кэш пакетов, распаковка архива, установка, при необходимости, выполнение действий по настройке, автоматически или, если требуется, в интерактивном режиме.

Команда `aptitude`, в отличие от `apt-get`, скачивает из репозитория, устанавливает и настраивает не только "жесткие" зависимости пакета, но и часть "мягких" (относящихся к категории рекомендованных). На усмотрение пользователя остается только установка "мягких" зависимостей из категории suggest.

Обновление пакетов

Пакеты обновляются при помощи команды `aptitude upgrade` или `aptitude dist-upgrade`. Последняя принудительно обновляет дистрибутив. Пакеты, имеющие статус h (фиксированная версия), не обновляются ни при выполнении `upgrade`, ни при `dist-upgrade`. Для установки статуса нужно выполнить команду `aptitude hold имя_пакета`, а для снятия — `aptitude keep имя_пакета`.

Переустановка пакета

Если по какой-либо причине необходимо переустановить пакет, то можно выполнить команду `aptitude reinstall имя_пакета`.

Удаление пакета

Для удаления пакета достаточно выполнить команду

```
aptitude remove имя_пакета
```

которая удалит пакет, но не затронет его конфигурационные файлы. Полностью очистить систему от всех остатков пакета можно командой

```
aptitude purge имя_пакета
```

Однако оператор `purge` не удаляет конфигурационные файлы пакета из домашнего каталога пользователя.

Обе команды удаления (`remove` и `purge`) деинсталлируют не только пакет, указанный в качестве аргумента, но и все пакеты, установленные автоматически для удовлетворения зависимостей, правда только в том случае, если в системе от этих пакетов больше ничто не зависит.

Очистка кэша

Для очистки кэша необходимо воспользоваться командами `aptitude clean` или `aptitude autoclean`. Первая просто удалит из локального кэша все пакеты, скачанные в процессе работы Aptitude. `Autoclean` удалит те пакеты из кэша, которые сейчас не установлены.

Помимо рассмотренных основных команд существует много редко встречающихся команд и ключей, описание которых можно почитать в документации на программу.

Ссылки

- **www.linuxdocs.org** — одно из собраний документации о Linux.
- **www.rpm.org/maximum-rpm.ps.gz** — источник сведений о RPM: "Maximum RPM" в формате PostScript.
- **www.redhat.com/support/docs/rpm/RPM-HOWTO/RPM-HOWTO.html** — RPM-HOWTO — описание RPM, тонкости работы (на английском языке).
- **www.linux.org.ru** — один из основных русскоязычных сайтов, посвященных Linux, в разделе документации есть RPM-HOWTO на русском языке.
- **www.rpm.org** — сайт, полностью посвященный RPM.
- **rpmfind.net** — репозиторий и поисковая система RPM.
- **rufus.w3.org/linux/RPM** — репозиторий RPM.
- **www.freshmeat.net** — большая коллекция программ, в том числе и в RPM-пакетах.
- **www.debian.org/doc/** — документация по Debian.
- man-страницы по apt-get, apt-cache и sources.list.
- **gazette.linux.ru.net/lg84/tougher.html** — Debian APT. Часть 1: Основные команды. Автор: Rob Tougher. Перевод: Александр Куприн.



Часть III

Инсталляция Linux

Глава 10



Подготовка к инсталляции

Рассмотрим процесс подготовки к установке операционной системы Linux. С Linux (если вы специально не ищете трудностей) проблем при инсталляции, скорее всего, не будет — вы поставите систему и все. И при выходе следующей версии дистрибутива ничего не придется переустанавливать заново. Даже если полностью поменять всю аппаратуру (кроме винчестера), в большинстве случаев Linux сама определит новое оборудование и перенастроит систему.

Впрочем, если вы не экспериментатор — кардинально менять систему часто не придется. Существуют серверы, на которых ОС Linux функционирует годами без внесения существенных изменений. Администратор к ним подходит раз в два месяца, чтобы сделать профилактику системного блока (пыль и т. п.). На этих машинах лишь периодически обновлялись некоторые прикладные пакеты: одни из-за проблем безопасности (ошибки есть в любой программе), другие для установки новых версий. Опытные пользователи Windows 9*x* (особенно те, кто много и часто ставят разнообразное программное обеспечение) помнят — систему приходится периодически переустанавливать. В последних версиях Windows эту проблему решили, однако при частой установке/деинсталляции программ система начинает медленней работать и приходится сторонними утилитами проводить профилактику системы.

С операционной системой Linux все несколько иначе. Во-первых, ее крайне сложно штатными способами довести до необходимости переустановки. Во-вторых, и это особенность любого программного обеспечения, как правило, новые версии программ весьма "сырые". К примеру, тяжело дался переход с версии ядра 2.2 на 2.4. Были времена, когда исправления к ядру выпускались буквально ежедневно. Есть хорошее правило: "Работает — не трогай". Поэтому опытные администраторы и пользователи выдерживают некоторую паузу после выхода очередного обновления, изучают отзывы и только после этого устанавливают обновление на систему. В-третьих, для перехода с одной версии дистрибутива на другую иногда приходится выводить систему из "общего пользования" на день-два, а то и больше. Вот, собственно, почему не следует без особых причин менять одну версию дистрибутива на другую.

Перед инсталляцией

В первую очередь необходимо где-то взять сам дистрибутив. Существует несколько путей:

- скачать с сайта производителя или с одного из "зеркал" (дублирующих сайтов);
- купить на сайте производителя или дистрибьютора с доставкой по почте;
- приобрести в магазине или на рынке;
- взять у знакомых.

Желательно также посмотреть в Интернете список обновленных программ и скачать необходимые пакеты.

До начала работы

Прежде чем начать работу, ознакомьтесь с инструкцией. В коробочный вариант входит брошюра с инструкцией по установке дистрибутива. Если вы не имеете коробочного варианта дистрибутива — не беда. На сайте производителя дистрибутива либо на одном из инсталляционных дисков есть документация по установке. Процесс подготовки к инсталляции и сама инсталляция подробнейшим образом описаны (на английском языке, если это не дистрибутив национального сборщика).

Список оборудования

При инсталляции система проверяет установленное оборудование и пытается самостоятельно определить его тип. В большинстве случаев это получается неплохо. Однако не всегда определение происходит корректно. Некоторые устройства система вообще может не найти. И тогда вам придется самостоятельно указать их параметры. Вот что вам необходимо знать об аппаратном обеспечении компьютера:

- число установленных жестких дисков;
- наличие RAID-контроллера, его чипсет и производителя;
- объем оперативной памяти;
- тип мыши;
- тип видеокарты (объем памяти и марку чипсета);
- тип SCSI-контроллера (если он есть);
- тип монитора, его кадровую и строчную частоты (для ЭЛТ-монитора), максимальное разрешение;
- сетевая карта (ее тип и чипсет);
- тип различных плат расширения (если они присутствуют).

Как правило, достаточно современное распространенное оборудование определяется нормально. Однако могут быть некоторые проблемы со звуком или с видеокартой (неверное определение размера оперативной памяти или типа процессора). Не всегда корректно происходит инсталляция на ноутбуках, т. к. производители ставят мало распространенные видеокамеры или картридеры.

Дополнительная информация

Если в компьютере установлена сетевая карта, и он подключен к локальной сети, необходимо также знать следующую информацию о сетевых настройках:

- IP-адрес;
- сетевую маску;
- адрес шлюза по умолчанию;
- IP-адрес DNS-сервера;
- доменное имя;
- имя компьютера в сети.

> ***Примечание***
>
> При динамическом распределении адресов в сети некоторые пункты из этого списка не обязательны.

Предполагаемый объем инсталляции

Весьма желательно представлять себе, сколько места займет установленный дистрибутив Linux. А для этого необходимо четко знать, что за систему вы устанавливаете. Вряд ли следует весь диск отвести под один раздел Linux. Тем более что на компьютере вполне может быть установлено несколько различных операционных систем.

Немного забегая вперед, посмотрим, что предлагает дистрибутив Fedora в качестве стандартного решения (установка всех пакетов): процессор *x*86 с частотой не менее 400 МГц, объем оперативной памяти не менее 512 Мбайт (лучше 1 Гбайт) и 10 Гбайт свободного места на жестком диске.

> ***Замечание***
>
> На первый взгляд — достаточно внушительные требования. Однако не следует забывать, что в инсталляционный комплект входит много разнообразных программ. Конечно, если вам требуется установить только роутер (маршрутизатор, от англ. *route* — маршрут, программное, аппаратное или программно-аппаратное решение, обеспечивающее передачу сетевых пакетов из одной сети в другую) или что-то подобное, нет необходимости занимать на жестком диске так много места. Существуют специальные дистрибутивы, предлагающие объем инсталляции в одну-две дискеты. Здесь и далее мы будем рассматривать среднестатистическую инсталляцию.

Теперь вы представляете, какие ресурсы необходимы для инсталляции дистрибутива. Не следует забывать, что помимо самой операционной системы вы будете устанавливать и свое программное обеспечение, записывать собственную информацию. Для функционирования ОС необходимо будет также создать так называемый Swap-раздел (своп-раздел, раздел подкачки), в котором временно хранится часть информации из оперативной памяти. Такая необходимость возникает, когда какому-то процессу срочно понадобилось много оперативной памяти. Если в системе для выделения этому процессу оперативной памяти не хватает, ядро операционной системы переносит неиспользуемые в данный момент процессы из оперативной памяти на своп-раздел. А когда необходимость в большом объеме оперативной памяти

отпадет, возвращает эти процессы из своп-раздела в оперативную память. Кроме того, из оперативной памяти на своп-раздел могут быть перенесены процессы, которые не используются длительное время.

Разбиение жесткого диска

Теперь, когда у нас есть представление об объеме, необходимом дистрибутиву, рассмотрим варианты разбиения жесткого диска. В зависимости от того, в каком качестве будет выступать ваша система (сервер, рабочая станция, экспериментальная система и т. п.), изменяются и требования по разбиению жесткого диска на разделы. Для начинающих практически любой дистрибутив Linux предлагает автоматическое разбиение жесткого диска. Для более опытных существуют рекомендации, учитывающие особенности установки. Но всегда найдется система, которая предъявит специфические требования.

Далее приведен список каталогов и рекомендации по вынесению их на отдельные разделы.

Каталог /

Каталог / — корень файловой системы. Все остальные каталоги являются его подкаталогами. Поскольку каталог / нельзя смонтировать в другом каталоге, обязательно создается корневой раздел.

Каждый каталог файловой системы Linux, не имеющий своего собственного раздела, представляет собой часть корневого раздела.

Обычно каталоги, размещаемые в корневом каталоге, не занимают много места и кардинально не увеличиваются во время эксплуатации системы. Каталоги такого типа (/bin, /dev, /etc, /mnt и т. п.), как правило, не помещают на отдельные разделы, а хранят в корневой файловой системе.

Каталог /bin

Каталог /bin содержит только исполняемые файлы, предназначенные в основном для администратора. Список файлов, содержащихся в этом каталоге, уже долгое время не претерпевает изменений, поэтому размер каталога /bin увеличивается только тогда, когда системный администратор устанавливает новые административные пакеты. Поскольку это происходит крайне редко, размер каталога /bin можно считать неизменным и поместить его в корневой раздел. Каталог /bin не зависит от других каталогов и не нуждается в свободном дисковом пространстве для выполнения своих задач.

Каталог /boot

Каталог /boot содержит все компоненты, необходимые для загрузки ядра ОС. Это могут быть несколько образов ядер операционной системы, карты модулей, конфигурационный файл, содержащий информацию о необходимых компонентах для запуска операционной системы. В процессе эксплуатации эти файлы изменяются только тогда, когда производится компиляция ядра. Перекомпиляция ядра,

как правило, не увеличивает занимаемое каталогом /boot дисковое пространство. Если каталог /boot находится в разделе, полностью заполненном информацией, это никоим образом не влияет на нормальную загрузку ОС. Следовательно, каталог /boot можно размещать в корневом разделе.

Каталог /dev

В каталоге /dev находятся специальные файлы устройств, предоставляющие интерфейс для доступа к различному оборудованию компьютера. Единственный исполняемый файл здесь — makedev — предназначен для создания файлов новых устройств.

Файлы устройств создаются только при установке нового оборудования. Но если каталог /dev находится на переполненном разделе, то создать новый файл устройства не удастся, что, как минимум, приведет к невозможности функционирования этого устройства. Каталог /dev не занимает много дискового пространства, однако обычно содержит более тысячи файлов устройств. Поскольку каталог /dev не увеличивается в размерах, его обычно размещают в корневой файловой системе.

Каталог /etc

Вся информация о настройках файловой системы содержится в файлах и подкаталогах, находящихся в каталоге /etc. Каталог /etc обычно не увеличивается в размерах, поскольку конфигурационные файлы программ редко занимают более чем 15–20 Кбайт. По этой причине каталог /etc обычно размещают в корневой файловой системе. Однако в каталоге /etc есть несколько файлов, изменяемых в процессе эксплуатации операционной системы. В частности, это файлы, содержащие список доступных дисковых разделов и смонтированных разделов. Поэтому, если каталог /etc окажется в переполненном дисковом разделе, нормальное функционирование операционной системы нарушается.

Каталог /home

В каталоге /home находятся каталоги пользователей системы. Для систем, в которых существует только несколько пользователей, для каталога /home обычно отдельный раздел не выделяется. Если в системе более десяти пользователей, имеет смысл создать для каталога /home отдельный дисковый раздел. Это позволит избежать проблем с переполнением диска. Для каталога /home рекомендуется также использовать программу quota, ограничивающую доступное для каждого пользователя место на жестком диске. В больших локальных сетях существует практика размещения каталога /home на сетевой файловой системе (NFS). Это дает пользователю возможность, помимо легкости администрирования и резервного копирования, получать доступ к своему домашнему каталогу с любого компьютера локальной сети.

Каталог /lib

В этом каталоге содержатся основные библиотеки операционной системы. Обновление или установка библиотек обычно производятся только при модернизации системы. Поскольку состав библиотек давно устоялся, резкого изменения занимаемого

дискового места в случае их модернизации не происходит. Даже переполнение дискового раздела не оказывает влияния на нормальное функционирование библиотек из каталога /lib. Поэтому каталог /lib помещают в корневую файловую систему. Еще одним аргументом в пользу размещения каталога /lib в корневой файловой системе является то, что практически все исполняемые файлы (в частности системные утилиты из каталогов /lib и /sbin) используют библиотеки из этого каталога.

Каталог /lost+found

В каждой файловой системе (разделе) автоматически создается каталог /lost+found. В нем утилита fsck размещает записи о файлах этой файловой системы, структура которых оказалась нарушенной. Поскольку каталог создается автоматически, нет необходимости заботиться о его размещении.

Каталог /mnt

Предназначен для размещения точек монтирования. Обычно в этом каталоге находятся каталоги /floppy и /CDROM, являющиеся точками монтирования дискет и компакт-дисков. Помимо этого, здесь могут монтироваться разнообразные файловые системы, в том числе и NFS. Каталог /mnt никогда не расходует дисковое пространство, поэтому обычно он размещается в корневой файловой системе. Как уже упоминалось ранее, если дисковый раздел, в котором располагается каталог /etc, будет переполнен, автоматическое монтирование файловых систем станет невозможным.

Каталог /opt

Каталог /opt предназначен для установки программного обеспечения, не входящего в стандартный состав операционной системы, к примеру, сервера баз данных Interbase. Размеры каталога /opt сильно зависят от устанавливаемого программного обеспечения. Поэтому для каталога /opt рекомендуется создать отдельный дисковый раздел. Переполнение этого дискового раздела влияет на функционирование программного обеспечения, находящегося в каталоге /opt, и практически не затрагивает работу операционной системы.

Каталог /proc

Каталог /proc — это точка монтирования псевдофайловой системы. Файлы и каталоги, находящиеся в каталоге /proc, на самом деле являются интерфейсом к некоторым параметрам операционной системы и не занимают место на жестком диске. Поэтому нет необходимости выделения для каталога /proc отдельного раздела. Каталог /proc создается в корневой файловой системе.

Каталог /root

Это личный каталог пользователя root. В нем находятся конфигурационные файлы этого пользователя. Хранить другие файлы в данном каталоге не рекомендуется, поскольку он находится в корневой файловой системе, переполнение которой приведет к неправильному функционированию операционной системы.

Каталог /sbin

Каталог /sbin по функциональному назначению подобен каталогу /bin. Поскольку изменения в каталоге /sbin маловероятны, его размещают в корневой файловой системе.

Каталог /tmp

Здесь хранятся временные файлы, создаваемые приложениями операционной системы. Поскольку такие файлы могут иметь размеры, исчисляемые десятками мегабайт, и не все приложения удаляют по окончании работы свои временные файлы, размер каталога /tmp за короткое время может существенно увеличиться. Поэтому крайне желательно выделять для каталога /tmp отдельный дисковый раздел.

Переполнение каталога /tmp не приводит к краху операционной системы, однако процессы, использующие его, перестают нормально функционировать. По этой причине каталог /tmp при перезагрузке ОС очищается от всех файлов.

Каталог /usr

Большинство приложений операционной системы устанавливается в каталог /usr, в связи с чем он требует много места на диске. Поэтому для каталога /usr практически всегда выделяется дисковый раздел. Также достаточно часто выделяют дисковый раздел и для каталога /usr/local.

Каталог /var

В каталоге /var приложения размещают свои рабочие файлы (почтовые файлы, файлы групп новостей, буфер печати, файлы для сервера FTP и т. п.). Здесь же находятся файлы журналов различных служб операционной системы. Отсюда следует, что в процессе жизнедеятельности ОС объем каталога /var не остается постоянным. При переполнении каталога /var бóльшая часть процессов операционной системы перестает корректно функционировать. Поэтому рекомендуется создать для каталога /var отдельный дисковый раздел. В зависимости от назначения сервера отдельные дисковые разделы могут создаваться и для хранения файлов серверов FTP, HTTP, каталогов /var/log и /var/spool.

Создание разделов на клиентских машинах

Требования к рабочим станциям менее жесткие, чем к серверам. Это связано с тем, что на клиентских машинах выполняется значительно меньше процессов, способных вызвать переполнение раздела. Кроме того, последствия переполнения раздела на клиентской машине не столь опасны, как на сервере.

На клиентских машинах можно создавать два или три раздела.

Создание разделов на сервере

Обычно на серверах требуется создавать больше разделов, чем на клиентских машинах, поскольку очень важно обеспечить максимальную устойчивость работы каждого сервера. Помимо своп-раздела на серверах создают отдельные разделы

для каталогов /, /boot, /tmp, /usr, /var. На многих серверах в зависимости от их назначения предусматривают и другие дополнительные разделы.

Сервер DNS

Сервер DNS предоставляет данные всем остальным компьютерам сети. Чтобы повысить надежность, обычно создают несколько дисковых разделов. Для каталога /var обязательно должен быть выделен отдельный раздел объемом не менее 100 Мбайт, поскольку при работе служба DNS генерирует большие файлы журналов и резервных копий.

Сервер NIS

К дисковым разделам ведомого сервера NIS предъявляют те же требования, что и к серверу DNS, за исключением того, что рост журналов не должен вызывать заполнения раздела, содержащего файлы карт YP. (Обычно они размещаются в каталоге /var/yp/maps.) Поэтому рекомендуется выделить отдельные дисковые разделы для каталогов /var и /var/log. Раздел /var/log должен иметь размер не менее 50 Мбайт, а размер раздела /var по крайней мере вдвое превосходить ожидаемый максимальный размер файлов карт YP.

К ведущим серверам NIS предъявляются те же требования, что и к ведомым, причем разделы /var и /var/log должны быть еще больше.

Почтовый сервер

На почтовых серверах для каталогов /var и /var/spool необходимо создать отдельные дисковые разделы довольно большого объема. Каталог /var/log должен располагаться в разделе размером не менее 100 Мбайт. Требуемый объем этого раздела пропорционален нагрузке на сервер. Каталогу /var/spool следует выделить объем, достаточный для размещения всех почтовых сообщений для всех пользователей, обслуживаемых данным почтовым сервером. Если почтовый сервер обслуживает достаточно много пользователей, размер этого раздела может составлять несколько гигабайт.

Серверы FTP и HTTP

Все FTP- и HTTP-серверы в своих разделах /var или /var/log должны иметь не менее 100 Мбайт свободной дисковой памяти, предназначенной для размещения файлов журналов. Требуемый объем свободного пространства пропорционален нагрузке на сервер.

Кроме того, вся структура каталогов, доступная FTP- или HTTP-демонам, должна располагаться в своем собственном разделе. Это позволит монтировать в системе данный раздел с особыми параметрами (например, с разрешением доступа "только для чтения"). Если службы FTP и HTTP функционируют на одном и том же сервере, для сохранения их данных следует предусмотреть отдельные разделы. Это упростит настройку параметров вычислительной среды.

Сервер NFS

Каталоги, предоставляемые в качестве ресурсов NFS, должны располагаться в отдельных разделах. Это упрощает их резервное копирование, а также предохра-

няет сервер от переполнения системных разделов. Величина NFS-раздела зависит только от администратора и политики фирмы. У одних это сотня мегабайт, у других — сотни гигабайт.

Сервер Samba

Как и в случае NFS, экспортируемые средствами Samba каталоги должны располагаться в отдельных разделах. Требования по объему раздела аналогичны серверу NFS.

Серверы новостей

Серверы групп новостей обрабатывают большой объем временных данных, имеющих сравнительно невысокую ценность. Спулы (под этим термином понимается временное хранилище информации, в частности очередь печати, почтовый файл пользователя и т. п.) файлов групп новостей могут иметь очень большой размер и должны обеспечивать быстрый доступ к данным.

Для /var/spool/news необходимо выделить отдельный раздел, который, в идеале, должен располагаться на отдельном жестком диске. Это способствует повышению производительности системы. Кроме того, при отказе диска со спулом новостей на новом устройстве потребуется лишь создать пустые разделы. Поскольку спулы новостей обычно содержат множество мелких файлов, отношение числа индексных блоков к общему объему дискового пространства должно быть в 3–4 раза больше, чем для обычных файловых систем.

Серверы баз данных

Планировать разделы на серверах баз данных следует при участии администраторов баз данных. Большинство крупных СУБД устанавливают в нескольких файловых системах, размещенных на нескольких дисковых устройствах. Кроме того, часто используют один или более разделов без файловых систем (raw, "сырой раздел"), предназначенных для хранения данных.

Достаточный объем свободного дискового пространства нужно зарезервировать и для создания файлов журналов, размещаемых в каталоге /var или /var/log.

Серверы приложений

На серверах приложений выполняется программное обеспечение, работа которого обычно жизненно важна для организации. Во многих случаях отказ любого из таких серверов вызывает остановку работы части или даже всей компании.

Выполняемые файлы программ, функционирующих на сервере приложений, обычно размещают в каталогах /opt или /usr/local. В любом случае, этот каталог должен располагаться в собственном разделе, поскольку объем его будет увеличиваться при каждой модернизации эксплуатируемых программ. Кроме того, следует обеспечить объем дискового пространства, достаточный для размещения нескольких копий приложения. Это упростит модернизацию приложений и обеспечит при необходимости возможность быстрого отката. Идеальный вариант — размещение раздела на RAID-массиве.

Сервер общего назначения

При создании разделов на сервере общего назначения необходимо учитывать два требования. Во-первых, система должна предоставлять пользователям множество различных служб. Во-вторых, нужно обеспечить возможность быстрого запуска сервера.

При принятии схемы разделения жесткого диска системные администраторы любого уровня должны учитывать приведенные рекомендации, собственный опыт и даже результаты применения метода проб и ошибок.

Применение рекомендаций

На практике реализация указанных рекомендаций может выглядеть следующим образом.

Создание своп-раздела. Общее правило для оценки его объема — RAM × 2 — корректно для 80% случаев. Но в специфических ситуациях размеры своп-раздела необходимо подбирать экспериментально. Впрочем, никто не мешает создать несколько своп-разделов и подключить к системе или создать специальные своп-файлы.

Для систем, у которых мало памяти (менее 1 Гбайт), рекомендуется выделять под своп-раздел не менее 2 Гбайт. Сейчас крайне редко можно встретить компьютер с таким объемом оперативной памяти. Поэтому стандартный объем своп-раздела на сегодняшний день — 1 Гбайт.

Более опытным пользователям можно рекомендовать в процессе работы следить за использованием своп-раздела командой `free` или `top`. Если обращение к своп-разделу систематически превышает 50% от времени функционирования системы, то желательно увеличить размер раздела или создать своп-файл.

В зависимости от назначения можно выделить три категории систем:

- домашний (офисный) компьютер, испытательный сервер, сервер небольшой локальной сети;
- удаленный сервер, сервер приложений (обобщенный);
- специальные серверы.

Первый тип систем простой (мгновенное обслуживание, практически нет угрозы взлома и большой нагрузки), поэтому диск можно разбить всего на 2–3 раздела:

- / — корневой;
- /boot — загрузочный (если нужен);
- /swap — раздел подкачки (своп-раздел).

Для второго и третьего типа систем общепринятая практика разбиения диска — создание отдельных разделов для каждого (или для группы) основных каталогов файловой системы. Это увеличивает безопасность и отказоустойчивость системы и, кроме того, удобно для выдачи пользователям дисковых квот. Лучший вариант: размещение каждого раздела на отдельном винчестере.

Достигаемые цели: защита от атак, гибкое управление дисковыми квотами, более быстрая загрузка (впрочем, для серверов это не актуально), легкое резервирование и восстановление системы, лучшая контролируемость файловой системы в целом.

Для систем второго и третьего типа рекомендуется такая разбивка:

- раздел / — 512 Мбайт, здесь находятся каталоги /bin, /sbin и т. п.;
- раздел /boot — 256 Мбайт, все образы ядер должны находиться здесь;
- раздел /usr — более 256 Мбайт, поскольку бóльшая часть исполняемых файлов Linux устанавливается в этот раздел;
- раздел /home — N Мбайт пропорционально числу пользователей + размерность квоты на каждого пользователя + небольшой запас. Например, 100 Мбайт на пользователя × на число пользователей;
- раздел /var — более 512 Мбайт, содержит файлы, которые могут изменяться (например, log-файлы, почтовые ящики);
- раздел /tmp — более 256 Мбайт, раздел для временных файлов. Сильно зависит от типа приложений.

Системы третьего типа отличаются особыми требованиями к определенным разделам. К примеру, серверу FTP необходимо выделить отдельный раздел для хранения файлов.

И в заключение еще одна рекомендация. Если в эксплуатации находятся несколько однотипных систем, старайтесь сделать максимально похожие конфигурации дисковых разделов и операционной системы — будет намного проще сопровождать и администрировать эти компьютеры.

Проблемы с оборудованием

Если у вас нетривиальная конфигурация компьютера, вполне может случиться, что какое-то устройство не установится. В этом случае остается через Интернет обращаться к FAQ, HOWTO, форумам и службам рассылки. На сайтах производителей дистрибутивов обычно существуют форумы поддержки и списки аппаратного обеспечения, которое нормально не функционирует под Linux.

Обычно проблемы с оборудованием возникают в следующих случаях.

- Очень новая видеокарта. Раньше приходилось ждать по полгода, пока энтузиасты напишут драйвер. Сейчас ситуация с драйверами исправляется. По крайней мере, лидеры на рынке видеокарт nVIDIA и AMD(ATI) выпускают драйверы под Linux.
- Принтеры. Можно подобрать драйвер похожего принтера или ждать выхода Linux-драйверов.
- Модемы. Для обычных модемов проблем нет. С так называемыми Win-модемами сложнее. На сайте **www.linmodems.org** можно найти драйверы для некоторых типов модемов. В частности хорошо работают Win-модемы на чипсете Lucent.
- Некоторые сетевые карты. По этому поводу существует специальный HOWTO, в котором подробно описывается решение проблем.
- RAID-контроллеры. Поищите драйверы на сайте производителя, почитайте соответствующий HOWTO.

- SCSI-контроллеры. Читайте FAQ и HOWTO.
- Манипулятор "мышь". Не всегда удается задействовать колесо прокрутки или дополнительные кнопки.
- Экзотическая периферия. Тут уж как повезет…

Ссылки

- **www.redhat.com/support/manuals** — руководства и документация.
- The Official Red Hat Linux x86 Installation Guide — руководство по установке Red Hat Linux x86.
- **linuxiso.org** — специальный сайт, содержащий ISO-образы.
- **www.linuxlinks.com** — почти полный список существующих дистрибутивов.
- **http://rus-linux.net** — виртуальная энциклопедия Linux.
- **www.debian.org** — сайт дистрибутива Debian.
- **www.redhat.com** — сайт дистрибутива Red Hat.
- **www.asplinux.ru** — сайт дистрибутива ASP Linux.
- **www.mandriva.ru** — русская версия дистрибутива Madriva.
- **www.bestlinux.net** — сайт дистрибутива Best Linux.
- **www.turbolinux.com** — сайт дистрибутива Turbo Linux.
- **www.slackware.com** — сайт дистрибутива Slackware.
- **www.novell.com/linux/** — сайт дистрибутива SuSE.

Глава 11



Инсталляция

Установка операционной системы сильно зависит от выбранного дистрибутива и требуемого результата: сервер, офисная система или домашний компьютер. Соответственно, имеется несколько вариантов инсталляции.

Во-первых, инсталляция возможна в графическом или текстовом режиме. Большинство современных дистрибутивов по умолчанию пытаются запустить инсталляцию операционной системы в графическом режиме, однако, если на компьютере недостаточно оперативной памяти, инсталляция происходит в текстовом режиме. Во-вторых, для семейства дистрибутивов Red Hat есть вариант установки в так называемом режиме kickstart, который позволяет провести инсталляцию по заранее созданному профилю. Это целесообразно для создания большого количества идентичных систем. И, в-третьих, можно обновить установленную ранее версию дистрибутива до текущей с минимальным вмешательством пользователя (если дистрибутивы однотипны). При этом сохраняются все данные пользователей, выполняется обновление ядра ОС, модулей и пакетов программ.

Для начала процесса инсталляции необходимо загрузить компьютер с носителя, содержащего специальную программу инсталляции. Таким носителем может быть компакт-диск дистрибутива, загрузочная дискета (обычная или с поддержкой PCMCIA) или Flash-накопитель.

Графическая инсталляция

В процессе работы программа инсталляции использует пять виртуальных консолей (1–5 — для текстового режима, 2–5 и 7 — для графического), на которые выводятся различные сообщения, позволяющие решать проблемы, возникающие при установке операционной системы. В табл. 11.1 приведена информация об этих консолях.

Таблица 11.1. *Доступные консоли при инсталляции Fedora Core*

№ консоли	Комбинация клавиш	Содержание
1	<Ctrl>+<Alt>+<F1>	Диалог инсталляции
2	<Ctrl>+<Alt>+<F2>	Командная строка

Таблица 11.1 *(окончание)*

№ консоли	Комбинация клавиш	Содержание
3	<Ctrl>+<Alt>+<F3>	Сообщения от программы инсталляции
4	<Ctrl>+<Alt>+<F4>	Системные сообщения
5	<Ctrl>+<Alt>+<F5>	Другие сообщения
7	<Ctrl>+<Alt>+<F7>	Графический дисплей X Window

Начало инсталляции

Особенности инсталляции таковы, что процесс начинается в текстовом режиме (рис. 11.1).

В самом начале программа инсталляции выводит меню, в котором предлагает установить, обновить или восстановить ОС, загрузиться с жесткого диска или протестировать оперативную память. Выбираем установку операционной системы (нажимаем <Enter>). Далее нам предлагают выбрать язык, который будет использоваться при инсталляции системы (рис. 11.2), затем следует выбор типа клавиатуры (рис. 11.3). После этого мы должны указать источник, откуда будет производиться установка: сменный носитель, жесткий диск, сетевой каталог либо установка через Интернет (рис. 11.4). Перед инсталляцией программа установки предлагает проверить носитель на ошибки (рис. 11.5). И, наконец, происходит переход в графический режим инсталляции (рис. 11.6).



Рис. 11.1. Начало инсталляции

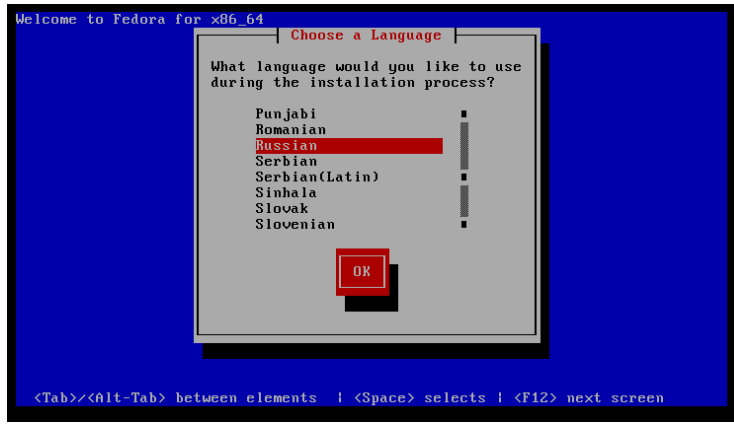


Рис. 11.2. Выбор языка инсталляции

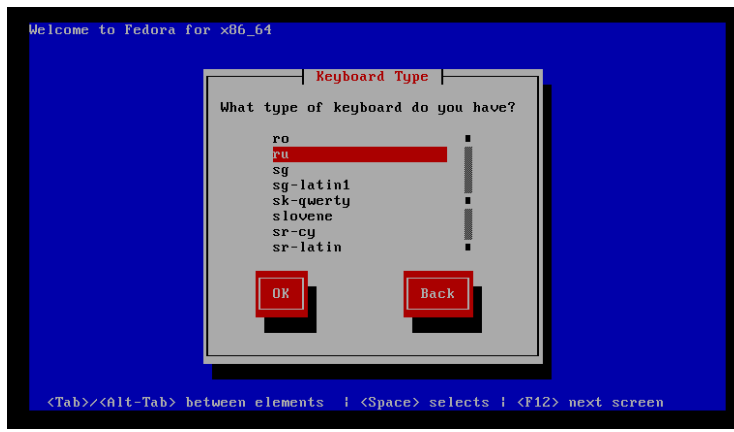


Рис. 11.3. Выбор раскладки клавиатуры

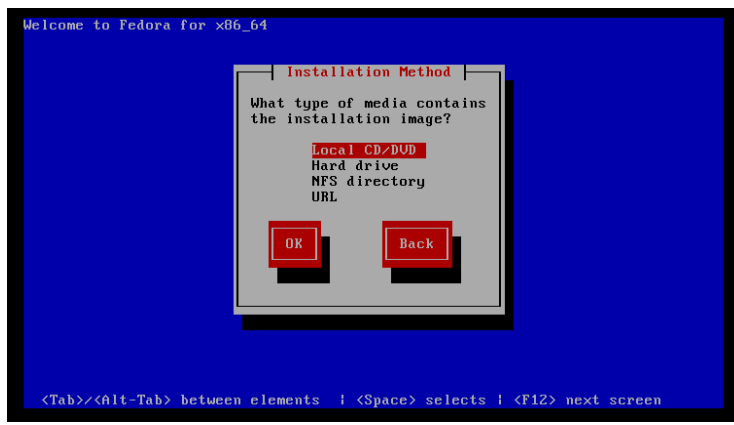


Рис. 11.4. Выбор источника инсталляции



Рис. 11.5. Проверка диска перед инсталляцией

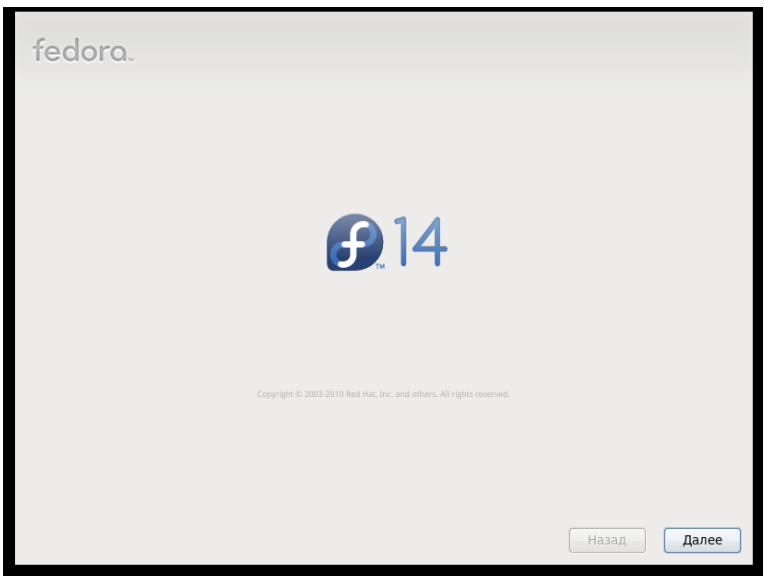


Рис. 11.6. Начало инсталляции в графическом режиме

Первые этапы

Далее следует определить, куда будет устанавливаться система (рис. 11.7). После этого нужно задать имя компьютера (рис. 11.8) и определить часовой пояс (рис. 11.9). Здесь можно ориентироваться по карте или по списку крупнейших городов.

На следующем этапе необходимо ввести и подтвердить пароль root (рис. 11.10). При этом система проверяет пароль на устойчивость и простоту и при необходимости сообщает о ненадежном пароле.

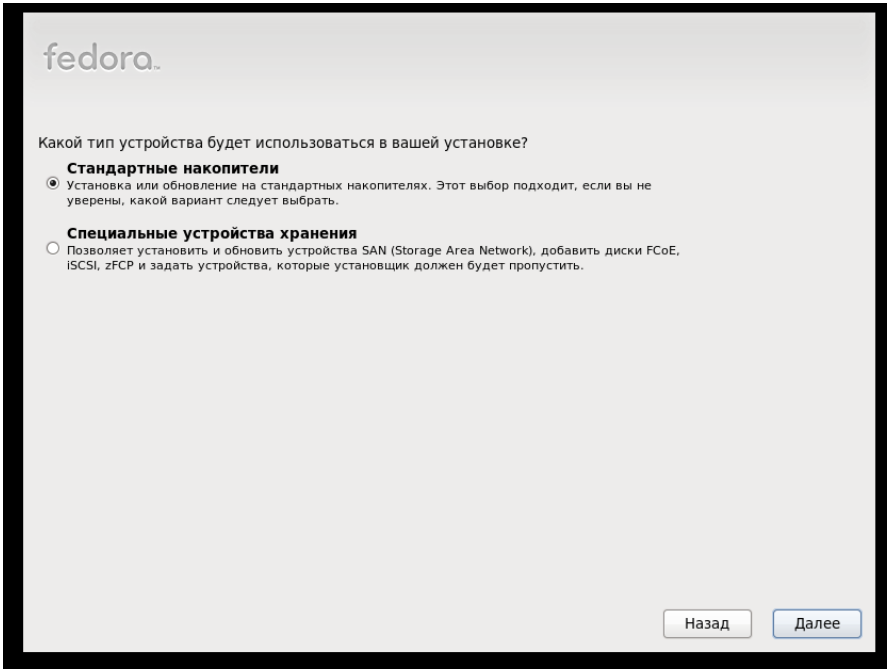


Рис. 11.7. Выбор жесткого диска

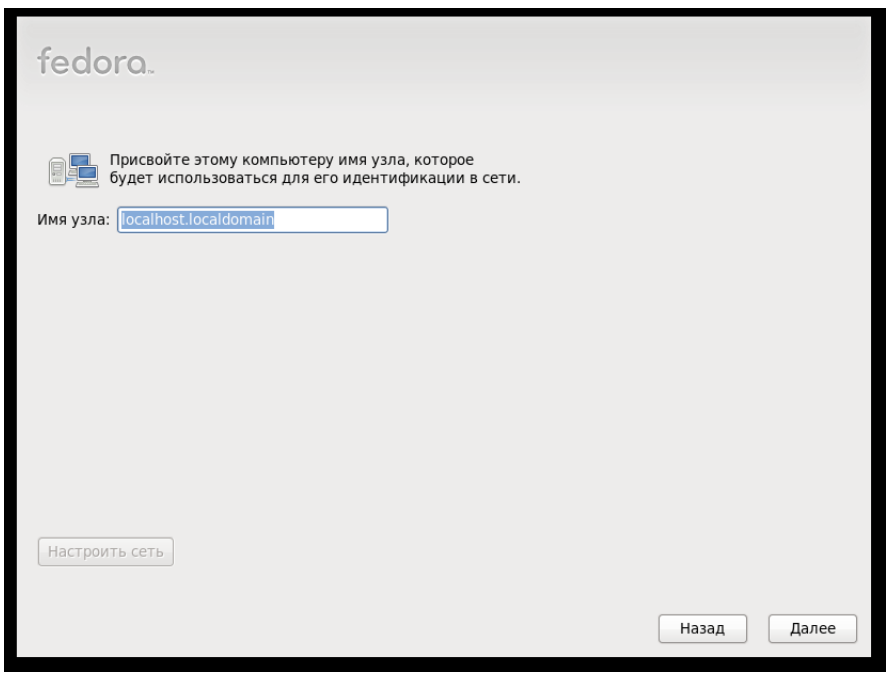


Рис. 11.8. Выбор имени компьютера

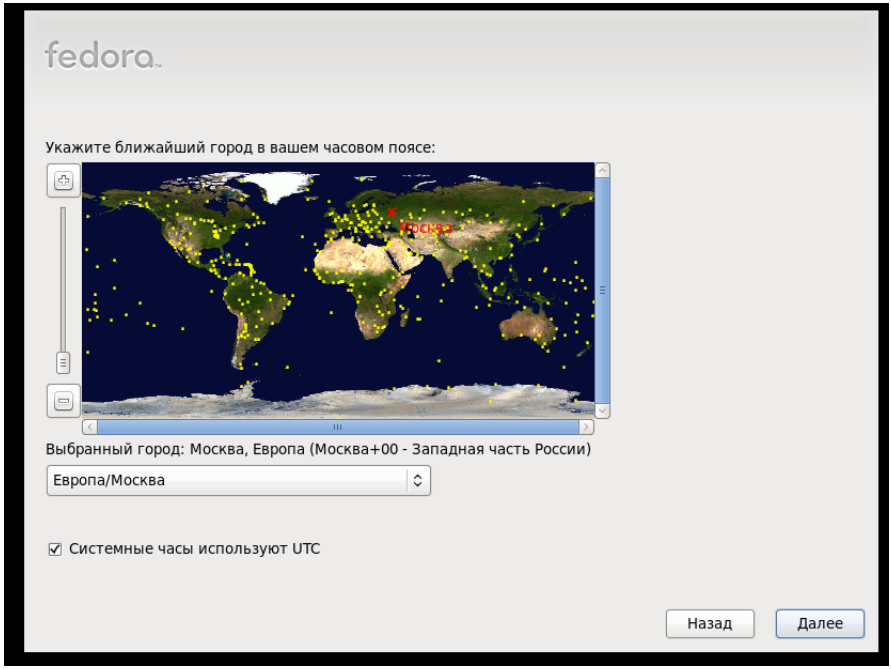


Рис. 11.9. Определение часового пояса

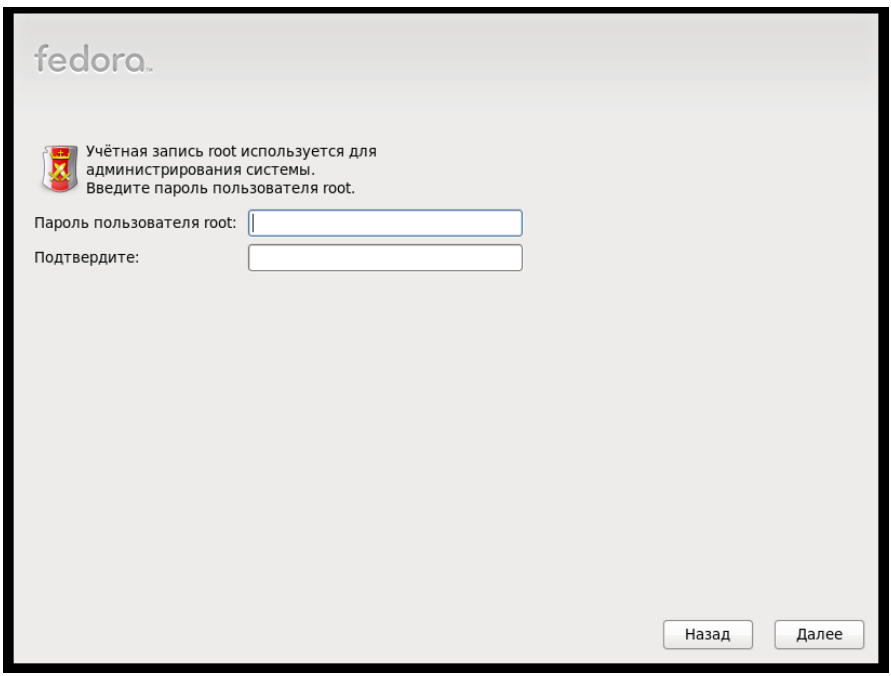


Рис. 11.10. Определение пароля root

Разбиение жесткого диска

На этом этапе следует определить, как будет разделено дисковое пространство (рис. 11.11). Необходимо выбрать физические жесткие диски, на которых будут создаваться разделы. Именно в этот момент можно перераспределить нужным образом дисковое пространство.

Можно зашифровать создаваемые разделы, это увеличит безопасность системы при хищении, но уменьшит скорость и затруднит восстановление данных при повреждении носителя. Если установим галочку возле пункта **Просмотр и изменение структуры разделов**, то сможем самостоятельно вносить изменения в конфигурацию разделов.

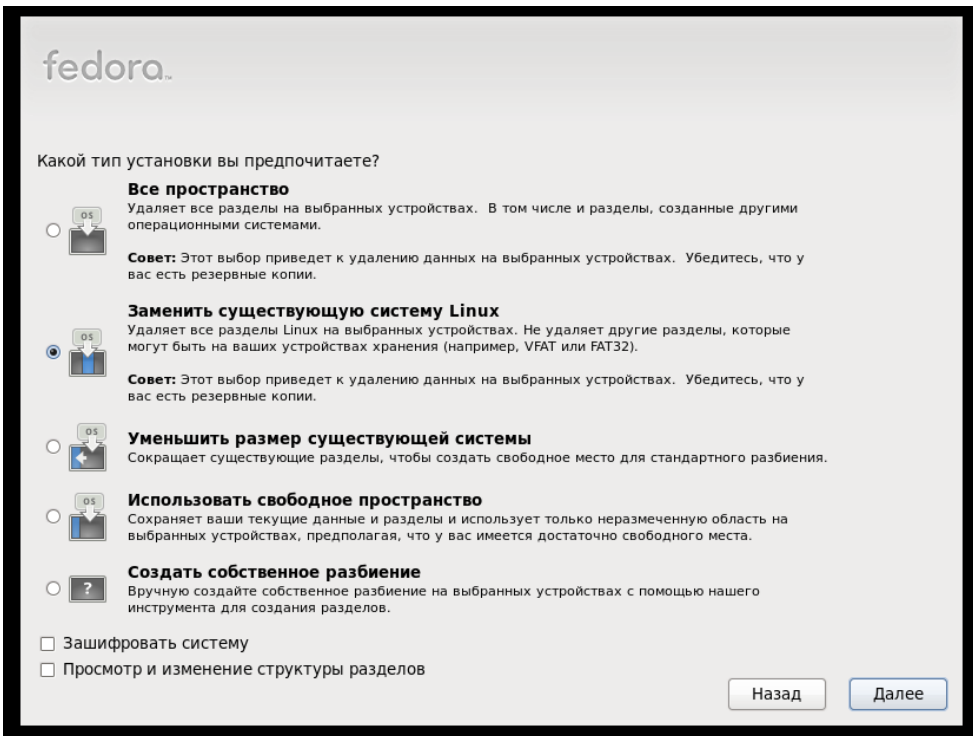


Рис. 11.11. Создание разделов диска

Выбор устанавливаемых пакетов

На этом этапе система предложит вам выбрать большие блоки программ, которые будут установлены при инсталляции. Обычно нам нужно более детально "покопаться" в системе, чтобы точнее подобрать необходимое ПО. Поэтому следует определить необходимые блоки программ и внизу окна выбрать опцию **Настроить сейчас**. Здесь можно также задать дополнительные репозитории программ, которые система будет проверять через Интернет для поиска обновлений (рис. 11.12).

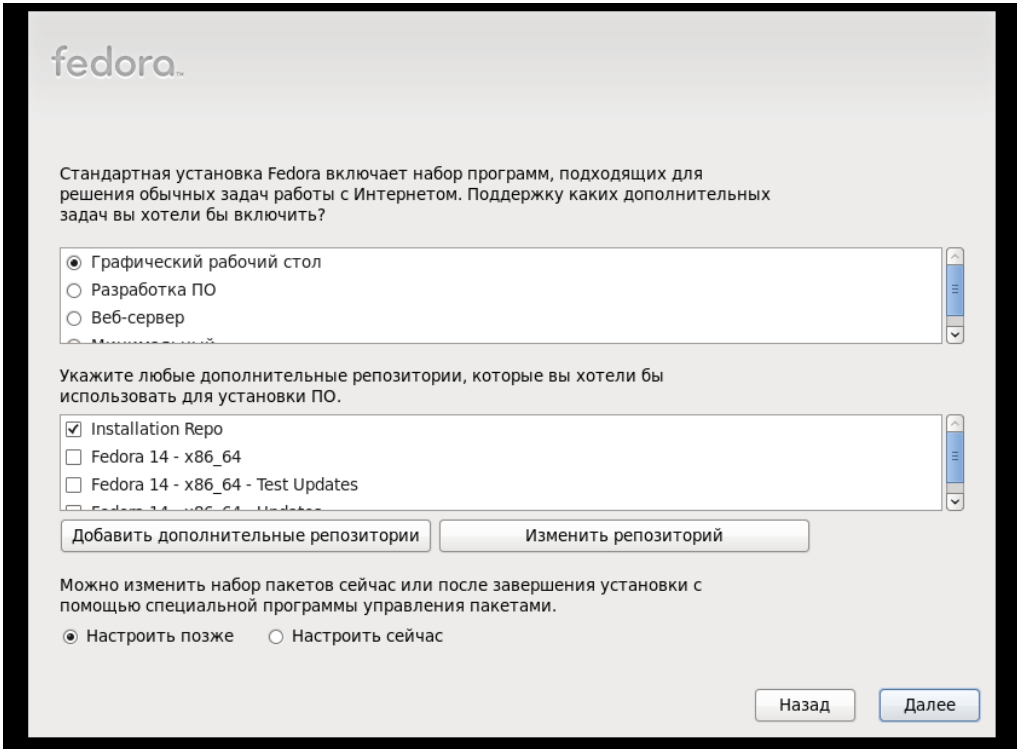


Рис. 11.12. Определение устанавливаемых пакетов

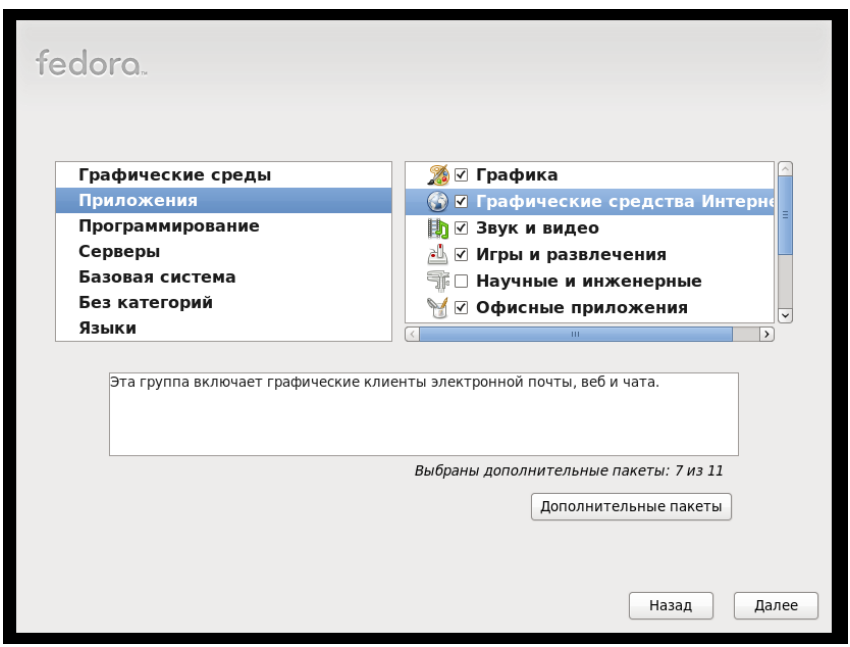


Рис. 11.13. Выбор устанавливаемых пакетов

Если мы выбрали опцию **Настроить сейчас**, то в следующем окне нам предложат поточнее определиться, какие пакеты мы желаем установить (рис. 11.13).

Все пакеты сгруппированы в шесть разделов:

- ❒ Графические среды
- ❒ Приложения
- ❒ Базовая система
- ❒ Программирование
- ❒ Серверы
- ❒ Языки

В этих разделах выбираем необходимые пакеты и жмем кнопку **Далее**.

Процесс инсталляции

Сначала система сообщит вам, что необходимо немножко подождать (рис. 11.14).

В этот момент система запускает модуль инсталляции, форматирует разделы, проверяет зависимости пакетов. Далее идет установка пакетов (рис. 11.15).

По окончании установки система перезагрузится и мы займемся ее конфигурацией.

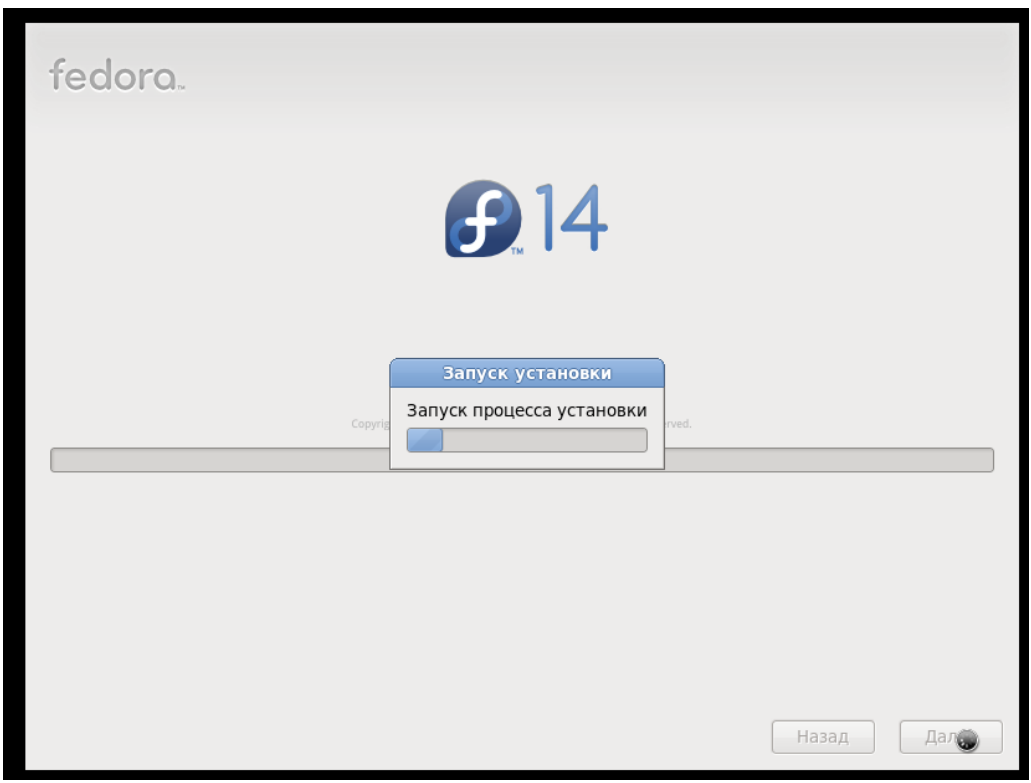


Рис. 11.14. Начало процесса установки



Рис. 11.15. Установка пакетов

Конфигурирование системы

После перезагрузки система выдаст нам следующее окно (рис. 11.16).

Затем нам предложат прочитать лицензию GPL (рис. 11.17). Далее необходимо создать пользователя для нормальной (не привилегированной) работы в системе (рис. 11.18). Следующий шаг — ввод даты и времени, а также указание источников синхронизации часов.

На следующем этапе нам показывают информацию о нашем оборудовании и предлагают отправить ее разработчикам для учета статистики и совершенствования дистрибутива (рис. 11.19). Но можно и не отправлять. Затем требуется выбрать пользователя и ввести пароль (рис. 11.20). И наконец, отображается наш рабочий стол (рис. 11.21).

Текстовая инсталляция

Текстовая инсталляция (поддерживаемая только старыми дистрибутивами) ничем принципиально не отличается от установки в графическом режиме, просто информация выводится в текстовой консоли.

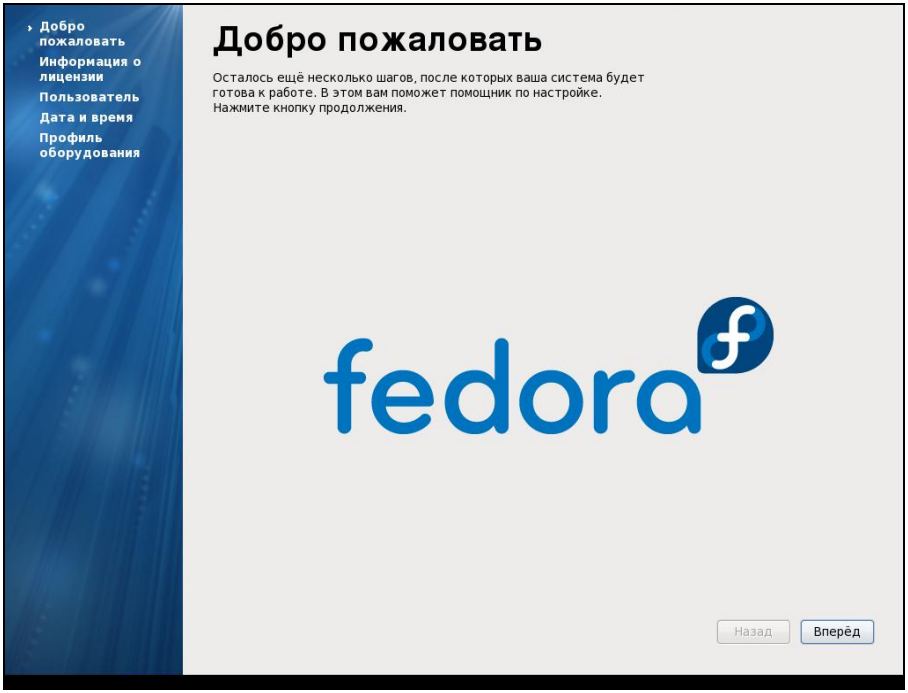


Рис. 11.16. Информационное окно

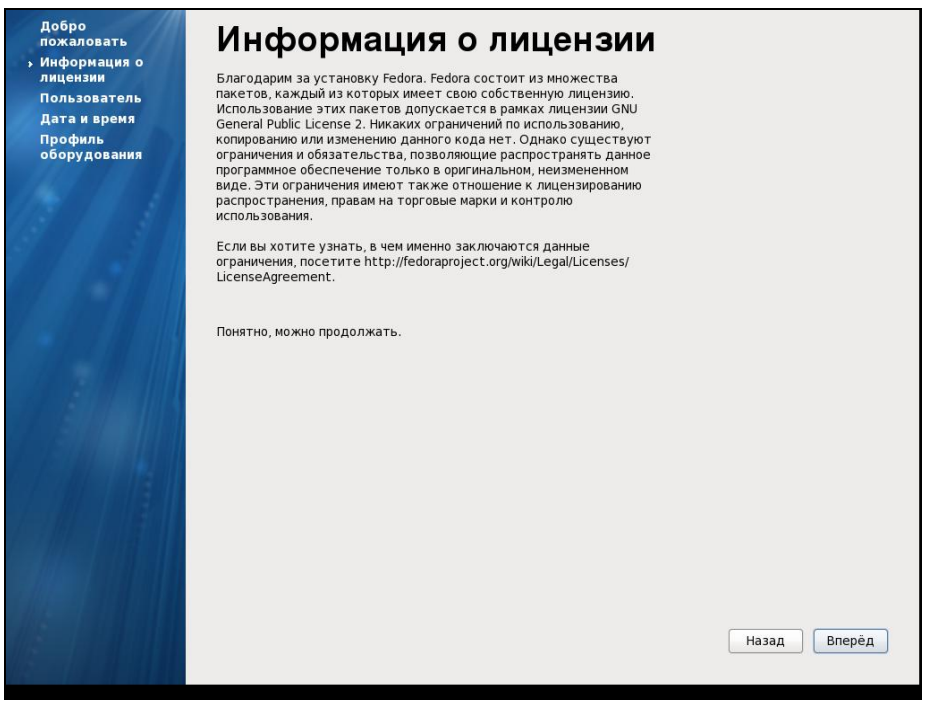


Рис. 11.17. Лицензия

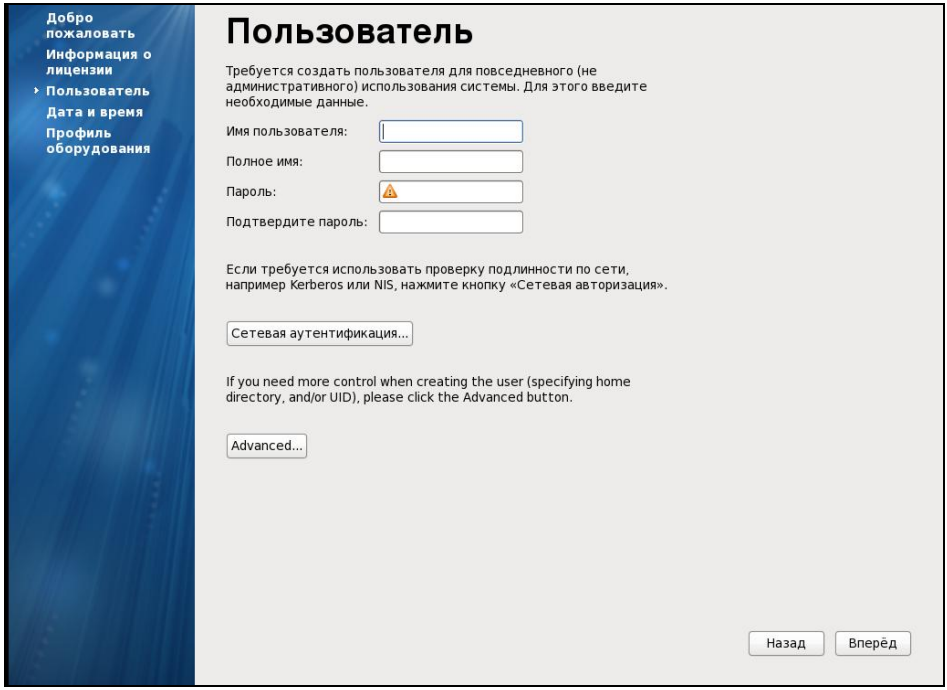


Рис. 11.18. Создание пользователя

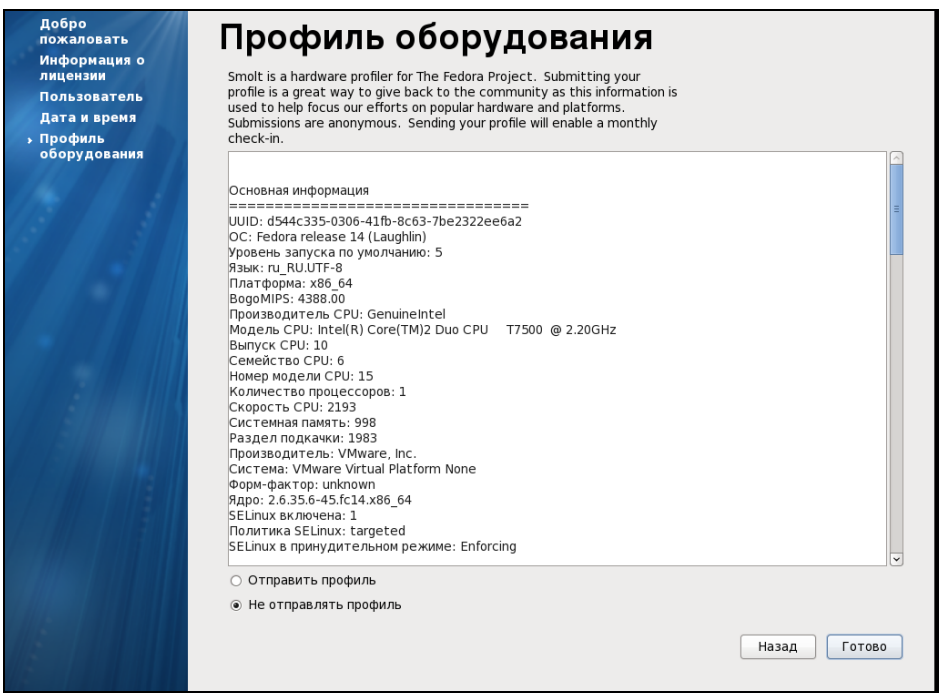


Рис. 11.19. Окно профиля оборудования

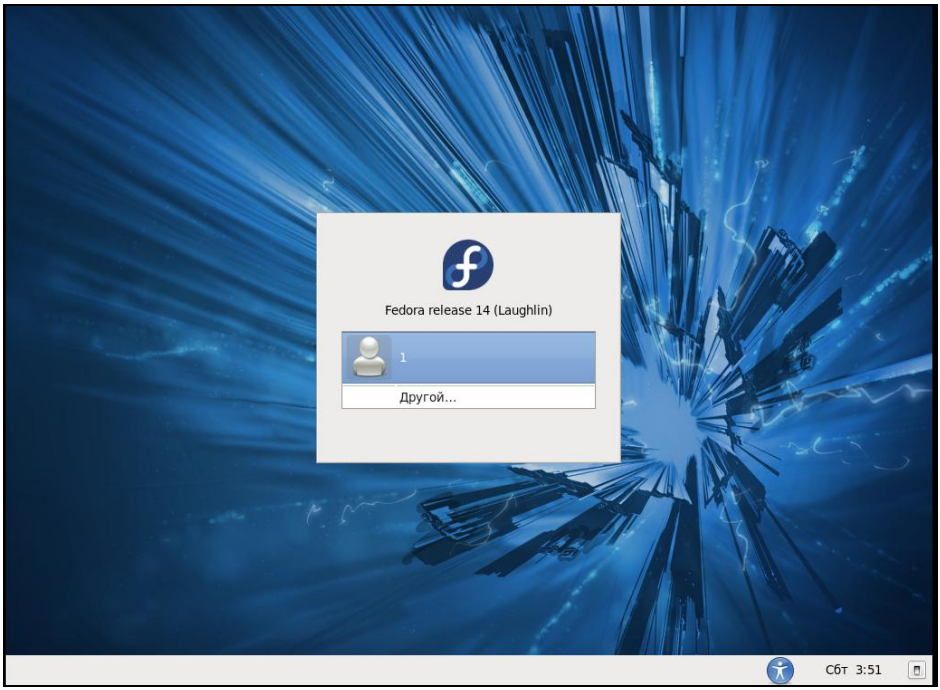


Рис. 11.20. Вход в систему

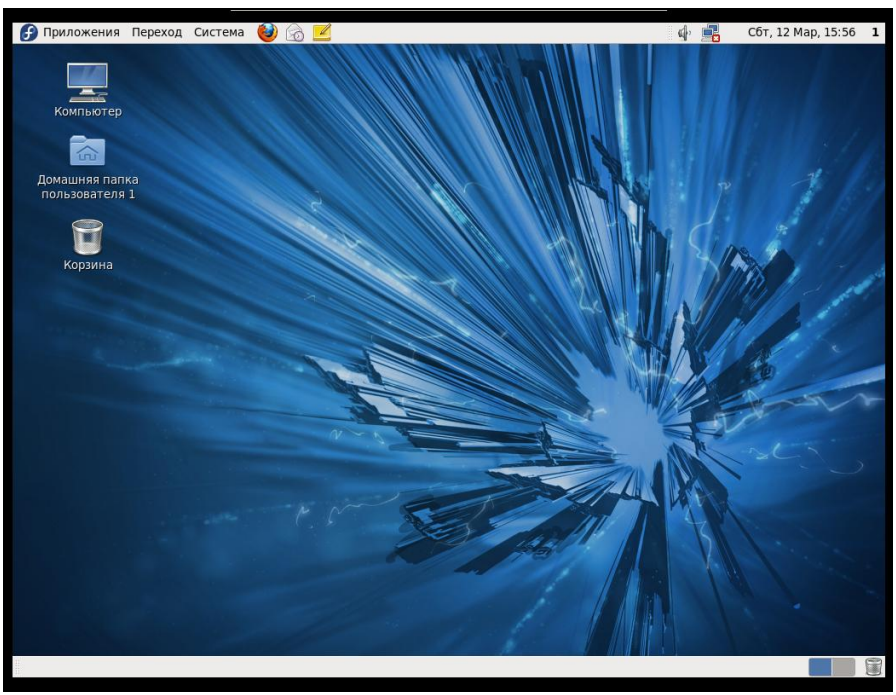


Рис. 11.21. Вход в систему выполнен

Инсталляция с жесткого диска

У многих дистрибутивов существует возможность инсталляции не только с компакт-диска, но и с жесткого диска. Для этого необходимо выбрать раздел жесткого диска и каталог, в котором находится ISO-образ инсталляционного диска.

Сетевая инсталляция

Процессы сетевой инсталляции с использованием NFS- и HTTP-сервера во многом схожи. Главное — выбрать тип инсталляции и адрес, откуда будет осуществляться установка.

Ссылки

- **http://www.redhat.com/support/manuals** — руководства и документация.
- **http://docs.fedoraproject.org/en-US/Fedora/14/html/Installation_Guide/index.html** — официальное руководство по инсталляции Fedora.



Глава 12

После инсталляции

По окончании процесса инсталляции, прежде всего, необходимо заняться конфигурированием операционной системы и удалением лишних пакетов. Затем нужно создать пользователей системы. Даже если у компьютера только один пользователь, и этот человек — вы, все равно следует создать обычного пользователя и работать только от его имени. Безопасность превыше всего. Работать от имени пользователя root крайне нежелательно. Малейшая ошибка — и операционной системы как не бывало. А восстановить удаленные файлы практически невозможно. В большинстве дистрибутивов в процессе инсталляции вам предложат создать пользователей. Не стоит пренебрегать этой возможностью.

Следующий шаг — установка всех обновлений пакетов, которые следует получить из Интернета, по крайней мере, для сервера и для рабочих мест, подключенных тем или иным способом к сети. Как правило, в большинстве современных дистрибутивов предусмотрена автоматическая проверка наличия обновлений при входе пользователя в систему.

Дальнейшие действия зависят от варианта инсталляции операционной системы Linux и от назначения компьютера.

Домашний компьютер

Для таких компьютеров характерно наличие разнообразного программного обеспечения. Поэтому, если у вас полный дистрибутив, просмотрите все имеющиеся пакеты программ. Это удобно делать из X Window, поскольку в ней (в частности, GNOME) производится автомонтирование компакт-дисков, а также автоматический запуск менеджера пакетов RPM (конечно, только в том случае, если эти пакеты присутствуют на компакт-диске). С менеджерами пакетов мы уже знакомились, поэтому здесь на их описании останавливаться не будем. В каждом менеджере пакетов можно посмотреть краткое описание выбранного пакета и решить, нужен он или нет. Поскольку домашний компьютер обычно предназначен для экспериментов, смело устанавливайте пакеты: не понравится — деинсталлируете. Как правило, в течение одной-двух недель вы определитесь, что подходит и будет использоваться, а что вряд ли понадобится. Тогда, настроив Интернет, следует обновить понравившиеся вам программные пакеты.

Как уже говорилось ранее, за решением проблем и советами по настройке различного программного и аппаратного обеспечения лучше всего обращаться на профильные форумы или сайты — люди там грамотные и достаточно доброжелательные.

Офисный компьютер

Подобный компьютер отличает, в первую очередь, использование офисных приложений и однотипность настроек. Если необходимые офисные приложения не были установлены во время инсталляции, то следует это сделать сразу после нее. Причем крайне желательно, чтобы это были последние версии, поскольку сейчас Linux-сообщество очень серьезно занялось доработкой офисных приложений, и полезные изменения и дополнения появляются буквально каждую неделю. Окончательный выбор программ остается за вами, но есть и общие рекомендации. Если у вашей фирмы велик объем документации, приходящей извне или отправляемой во "внешний" мир в электронном виде, то вам не обойтись без офисных приложений, способных работать с документами формата Microsoft Office. В настоящее время из бесплатного ПО наиболее корректно работает с такими документами OpenOffice. Однако за функциональность все же приходится платить, в данном случае местом на жестком диске и требованиями к процессору и оперативной памяти.

Если сеть не была настроена при инсталляции (офис подразумевает наличие локальной сети), это нужно сделать сейчас. Затем, с помощью менеджера пакетов следует отсеять все лишнее программное обеспечение. Как правило, это службы типа finger, r- (rlogin, rcopy и т. п.), telnet (клиент и сервер) и довольно много другого ПО, представляющего потенциальную брешь в защите системы. В идеале, на офисном компьютере не должно быть никаких сетевых служб, кроме SSH. По инерции в дистрибутивы входят (и по умолчанию устанавливаются на компьютер) многие устаревшие пакеты, разработанные лет двадцать-тридцать назад. В те времена о мощи современных персональных компьютеров даже и не мечтали, Интернета не было как такового, а локальные сети только начали появляться. Что такое взлом, подбор паролей, троянские программы и вирусы, никто не знал. Поэтому пароли в сетевых приложениях типа telnet передавались в открытом виде, а порой вообще отсутствовали. В современном мире такое недопустимо. Поэтому лучше все потенциально опасное убрать от греха подальше.

После установки необходимого программного обеспечения, настройки сети, электронной почты и службы новостей рекомендуется установить и настроить клиент NTP — службу точного времени. Благодаря ей вы навсегда забудете о проблеме синхронизации системных часов компьютера. В большинстве дистрибутивов в конце инсталляции вам предложат разрешить компьютеру при запуске синхронизироваться с серверами точного времени.

И теперь, когда операционная система полностью настроена, желательно сделать ее резервную копию. Вариантов может быть несколько:

- каждый компьютер предприятия имеет свою резервную копию;
- группа компьютеров имеет одну резервную копию, а для каждого компьютера резервируются его конфигурационные файлы;
- все компьютеры имеют одну резервную копию, а для каждого компьютера резервируются его конфигурационные файлы.

Первый вариант представляется несколько излишним. Нет смысла организовывать запись и хранение нескольких десятков DVD только из-за того, что у компьютеров разные имена и IP-адреса. С другой стороны, в случае проблемы на компьютере просто разворачивают резервную копию.

Вариант номер два оптимальный. В фирме выделяют группы компьютеров, абсолютно идентичных по установленному программному обеспечению. Делают одну резервную копию. Дополнительно резервируют конфигурационные файлы каждого компьютера. При возникновении проблем нужно развернуть резервную копию и переписать соответствующие конфигурационные файлы (и информацию пользователя).

Третий вариант самый неудачный. В случае проблем придется развернуть резервную копию, переписать соответствующие конфигурационные файлы, информацию пользователя и добавить отсутствующее программное обеспечение (либо наоборот, убрать ненужное).

И, конечно, необходима процедура ежедневного резервного копирования. Для ее облегчения рекомендуется выделить сервер, на котором должны храниться файлы и почтовые сообщения пользователей, и проводить резервирование содержимого этого сервера.

Компьютер программиста, администратора

Компьютер программиста или системного администратора несколько отличается по установленному программному обеспечению от обычной офисной машины.

Как правило, на такие компьютеры устанавливается много специфичного ПО, которое потенциально может ухудшить безопасность системы. Поэтому доступ к таким машинам должен быть ограничен как в физическом смысле, так и посредством сетевых соединений.

На компьютер программиста помимо разнообразных сред программирования, отладочных средств, компиляторов и интерпретаторов зачастую устанавливают сервисы баз данных, FTP-, HTTP-демоны и т. п. Это необходимо для того, чтобы в процессе отладки программы (или скрипта) программист (или администратор) ставил эксперименты на тренировочной, тестовой базе данных, Web-сайте или сервере приложений, а не на рабочем сервере фирмы.

Поскольку компьютер программиста часто содержит достаточно ценную информацию, а из-за особенности программистской деятельности подвергается повышенному риску, его данные желательно резервировать два-три раза в день.

Приблизительно такой же спецификой обладает компьютер системного администратора. Помимо различных компиляторов и интерпретаторов (C/C++ для компиляции ядра и программ, Perl и Pyton для скриптов и т. п.), на него устанавливается специальное программное обеспечение для мониторинга сети и администрирования. Все вновь разработанные или модифицированные скрипты системный администратор должен попробовать сначала "на себе", и только после этого устанавливать на другие компьютеры.

Сервер

Самым специфическим компьютером, как правило, является сервер. Специфика эта возникает в зависимости от конфигурации, выполняемых задач, количества обслуживаемых пользователей и требований надежности. Поэтому я дам здесь лишь весьма общие рекомендации.

Сразу по окончании процедуры инсталляции следует проверить, все ли аппаратное обеспечение сконфигурировано и работает правильно. Особое внимание нужно обратить на SCSI-устройства и сетевые карты (если их более одной). По умолчанию обычно конфигурируется только одна сетевая карта, все остальные, установленные в системе, приходится настраивать самостоятельно. Устройства SCSI также прибавят хлопот. Если на сервере установлены и USB-устройства (правда, для чего они на сервере, представить трудно), необходимо проверить и их функционирование.

Затем следует установить новое программное обеспечение (и в дальнейшем отслеживать выход новых версий программ). Если потребуется — обновить и/или скомпилировать ядро операционной системы. Здесь нужно быть особенно внимательным — для выполнения некоторых функций (например, firewall) при компиляции придется включить свойства, обычно отключенные по умолчанию. После этого настраивают сервисы и удаляют все лишние для сервера программы.

Сервер — самая чувствительная к взлому система в локальной сети. От потери функциональности или замедления его работы страдают все работники фирмы, а если это почтовый или Web-сервер, то проблемы появляются и у людей, желающих отправить вам почту или посмотреть ваш Web-сайт. Именно поэтому, как и с простого офисного компьютера, с сервера необходимо удалить все потенциально опасные службы типа finger, r- (rlogin, rcopy и т. п.), telnet (клиент и сервер), NFS и т. п. Установленное ПО должно точно соответствовать назначению сервера. К примеру, на сервере баз данных должно стоять только программное обеспечение баз данных. И ничего другого. Никаких Web-серверов, игр, X Window и компиляторов.

Далее приведен небольшой (далеко не полный) список пакетов, которые на серверах общего назначения не нужны:

- BOOTP (Boot Protocol) — служит для загрузки бездисковых рабочих станций. Если сервер не предназначен для удаленной загрузки, нет необходимости оставлять этот пакет;
- DHCP (Dynamic Host Configuration Protocol) — протокол, который позволяет отдельным устройствам в IP-сетях получать от сервера конфигурационную информацию (IP-адрес, сетевую маску, широковещательный адрес и т. д.). Пакет необходим исключительно для DHCP-сервера;
- mt-st — включает программное обеспечение для управления устройствами чтения с магнитных лент: mt (для устройств magnetic tape devices) и st (для SCSI tape devices). При отсутствии стримера эти пакеты лишние;
- eject — позволяет пользователям извлекать диски (обычно это CD-ROM, Iomega Jazz и Zip), используя программные средства. Данная программа тоже не понадобится;

- apmd — демон расширенного управления питанием и сопутствующие ему утилиты. Такое программное обеспечение предназначено для ноутбуков, на сервере ему делать нечего;
- linuxconf — удобная утилита для настройки системы. По умолчанию она не устанавливается. Если она все же установлена, следует знать, что эта программа занимает довольно много места и к тому же содержит ошибки;
- isapnptools — включает утилиты для настройки карт ISA Plug and Play (PnP) и плат, которые совместимы со спецификацией ISA Plug and Play. Поскольку в современном компьютере вот уже на протяжении нескольких лет не устанавливаются ISA-устройства, наличие этого пакета нецелесообразно;
- setserial — системная утилита для просмотра и установки информации о последовательных портах. Необходима на сервере модемного доступа и сервере управления кассовыми аппаратами с последовательным интерфейсом. Может потребоваться для маршрутизаторов, имеющих модемные соединения. Утилита позволяет отлаживать соединения и управлять источником бесперебойного питания (UPS). Для всех остальных типов серверов наличие ее нецелесообразно;
- kudzu — утилита для автоматического определения аппаратного обеспечения. Во время загрузки она может определить, какие устройства были добавлены или удалены из системы. Однозначного мнения, нужен или нет данный пакет на сервере, не существует;
- raidtools — включает утилиты, которые нужны для установки и управления программными RAID-массивами. Если программные RAID-массивы не используются, не устанавливайте;
- redhat-logos — файлы логотипов — лишняя трата дискового пространства;
- redhat-release — на сервере не нужен;
- rmt — предоставляет удаленный доступ для резервного копирования. Как и все r-команды — потенциальная брешь в безопасности операционной системы;
- tux — встроенный в ядро HTTP-сервер. Позволяет ускорить обработку HTTP-запросов. Ни для каких серверов, кроме Web-сервера, не нужен. К тому же, на Web-сервере традиционно устанавливается сервер Apache.

После инсталляции и компиляции всего необходимого программного обеспечения рекомендуется удалить с сервера все компиляторы и подобные им программы. Это делается для того, чтобы злоумышленник, проникший на сервер, не смог скомпилировать или модифицировать необходимые ему утилиты. Как известно, знаменитый "червь Морриса" пересылал свой исходный код на компьютер жертвы, там себя компилировал и запускал на выполнение.

Создание различных серверов для разных задач упрощает процесс администрирования и управления ими и увеличивает контроль и настраиваемость для каждого из них.

Дальнейшие действия большей частью административные.

1. Создайте список файлов с их правами и владельцами. Регулярно проверяйте, не изменились ли атрибуты, права и владельцы файлов. Изменение этих параметров — один из признаков взлома или некорректной работы пользователей.
2. Заведите на сервере пользователя, которому делегируйте некоторые права — выключение сервера, монтирование дисковых разделов, административные задачи.

3. Проверьте, чтобы пользователь root не мог зайти через сеть или способом, отличным от входа с консоли.
4. Организуйте перенаправление почты пользователя root на обычного пользователя, отвечающего за администрирование. По умолчанию многие сервисы отсылают электронной почтой сообщения о проблемах, возникающих во время их работы.
5. Не работайте пользователем root — этот пользователь случайно может уничтожить операционную систему.
6. Настройте службу logrotate: на серьезных серверах log-файлы в 100–200 Мбайт — в порядке вещей.
7. Разработайте стратегию резервного копирования сервера.
8. Проводите учебное восстановление сервера из резервных копий раз в квартал. Это позволит и хорошо отработать процедуру восстановления, и проверять целостность резервных копий.
9. Обязательно приобретите источник бесперебойного питания с управлением по последовательному порту и установите соответствующее программное обеспечение.
10. Разработайте и утвердите у руководства правила, в которых четко и кратко описано, что может делать пользователь, а что ему запрещено. Ознакомьте с ними всех сотрудников.
11. Периодически проверяйте компьютеры на наличие постороннего ПО и правильное функционирование штатного программного и аппаратного обеспечения.

Более подробные рекомендации можно прочитать в книгах "UNIX: руководство системного администратора" и "Системное администрирование Linux".

Ссылки

- **www.linuxdocs.org** — Network Administrator's Guide.
- **www.linuxdocs.org** — по этому же адресу расположены и соответствующие HOWTO:
  - Security-HOWTO;
  - Hacker-HOWTO;
  - NFS-HOWTO;
  - Firewall-HOWTO.



Часть IV

Основные команды Linux

Глава 13



Помощь

Ни одна мало-мальски приличная программа не обходится без справочного руководства. UNIX-системы обладают, пожалуй, самой обширной и объемной документацией, касающейся функционирования операционной системы и программ.

Традиционно информацию о системе или программе можно получить несколькими путями.

apropos

Команда `apropos` ищет заданное ключевое слово (команду) в базе `whatis` (см. далее) и выводит на экран краткое его описание.

Man-справка

В операционной системе Linux справочная система глобальна и работать с ней очень просто — в командной строке следует набрать:

```
man имя_программы
```

В результате на текущую консоль будет выведена справочная страница по использованию указанной программы. В принципе, параметром команды `man` не обязательно должно быть имя программы, это может быть любой файл операционной системы. В том случае, если для указанного файла не существует справочной страницы, на экран будет выдано соответствующее сообщение.

Если система инсталлирована с языком, отличным от английского, то и справочные страницы (если, конечно, они есть) установятся на языке, который был выбран при инсталляции. К сожалению, для русского языка справочных страниц крайне мало, однако со временем такая ситуация будет исправлена.

Программа `man` сначала пытается найти справочную страницу на языке текущей локали. Если же такой страницы нет, она выдаст на консоль справочную страницу на английском языке.

whatis

Команда `whatis` представляет собой мини-справочную систему. В качестве аргумента указывают имя файла, на выходе получают строку информации об этом файле.

HOWTO — как сделать

С помощью программы man можно узнать многое, единственное, что она не дает — это алгоритмов решения сложных проблем. Для подобных вопросов существуют так называемые HOWTO — "как сделать что-то". Это довольно большой набор файлов, предназначенных для решения разнообразных проблем: настройки сети, почтовых программ, Web-серверов, установки различного аппаратного обеспечения и многого другого. Как правило, эти файлы написаны на английском языке, но существуют версии некоторых HOWTO на других языках, в частности на русском. Найти их в Интернете нетрудно — достаточно ввести наименование искомого HOWTO в любую поисковую систему, например **www.rambler.ru**.

Мини-HOWTO

В отличие от HOWTO, решающих глобальные задачи и имеющих объем порядка 50–100 страниц, мини-HOWTO посвящены узкоспециальным проблемам и имеют небольшой размер. Их также легко найти в Интернете.

Руководства пользователя Fedora

На сайте дистрибутива **http://docs.fedoraproject.org/ru-RU/index.html** существует большой выбор документации, посвященной инсталляции, работе и настройке операционной системы Linux.

В этих документах подробно рассказывается об инсталляции, работе, решениях возникающих проблем в дистрибутиве, настройке безопасности, работе с системами хранения данных и многое другое. К сожалению, бóльшая часть документации на английском языке.

Документация Slackware

На сайте **www.slackware.ru** есть несколько русскоязычных статей и документации. На **www.slackware.org** находится англоязычная документация.

Руководство пользователя Alt Linux

В комплекте с дистрибутивом Alt Linux поставляется печатная документация на русском языке, а также и в электронном виде в составе дистрибутива. На сайте **www.altlinux.ru** можно скачать последнюю версию документации.

Документация Debian

На сайте **www.debian.org/doc/** находится много документов как на русском, так и на английском языке.

Ссылки

- **www.linuxdocs.org** — одно из наиболее полных собраний документации о Linux. Ресурс англоязычный, зато почти все, что касается Linux, здесь можно тем или иным образом найти.
- **www.redhat.com** — сайт фирмы Red Hat, производителя одноименного дистрибутива. Одно из достоинств данного дистрибутива — его хорошая поддержка, начиная с версии 4.*x* (создан в 1995 году) и заканчивая текущим. Дистрибутив Red Hat получил очень широкое распространение, его уже начали сравнивать с Windows. В настоящее время Red Hat стал стандартом де-факто для производителей коммерческого программного обеспечения и компьютерного оборудования. Кроме собственно дистрибутива на сайте присутствует и достаточно большой объем качественно написанной документации (на английском языке).
- **www.fedoraproject.org** — сайт проекта Fedora. Как известно, Red Hat переключился полностью на коммерческий дистрибутив Linux, однако был создан проект Fedora, в котором обкатываются новые решения, внедряемые затем в дистрибутив Red Hat.
- **www.debian.org** — сайт разработчиков дистрибутива Debian.
- **www.altlinux.ru** — сайт дистрибутива Alt Linux.
- **www.slackware.ru** — русское зеркало дистрибутива Slackware.

Глава 14



Справочник наиболее часто употребляемых команд

Эта глава посвящена консольным командам и утилитам. Конечно, после продолжительной работы в графической среде нет особого желания возвращаться в текстовую консоль. Однако не всегда разумно пользоваться X Window там, где достаточно набрать всего три буквы. Да и не везде будет возможность (желание, необходимость) устанавливать и использовать X Window. Нецелесообразно занимать на сервере лишние десятки мегабайтов, расходовать драгоценную оперативную память и время процессора (камень в огород Windows NT Server, Windows 2000, Windows 2003) на обслуживание графической оболочки. Действительно, зачем серверу (конечно кроме сервера приложений X Window) иметь графическую оболочку, если на нем выполняется сервер баз данных, Web-сервер, почтовый сервер или сервер новостей? Этого не требуется и для администрирования. Практически любое приложение имеет понятный, самодокументированный (или описанный в документации) текстовый файл конфигурации. Отредактировать конфигурационные файлы можно либо с текстовой консоли, либо даже удаленно (подключиться к компьютеру, находясь хоть на другом континенте). Для целого ряда приложений существуют также утилиты конфигурирования, имеющие текстовый (а иногда и графический) интерфейс. Многие приложения снабжены инструментами удаленного администрирования с Web-интерфейсом. Так что наличие консольных команд и утилит для сервера вполне обоснованно.

То же и для обычных клиентских машин. Казалось бы, поскольку большинство пользователей безвылазно сидят в X Window, им вообще не нужно знать о консоли. Однако это совершенно не так. Как нам уже известно, ОС Linux загружается в текстовом (консольном) режиме и лишь на последнем этапе (и то, если это было определено при конфигурации системы) происходит переключение в X Window. Даже из-за этого уже стоит ознакомиться с консольными утилитами.

Идеология утилит Linux подразумевает модульность, доведенную до совершенства. В распоряжении пользователя множество утилит, идеально выполняющих какую-то одну конкретную операцию. Из-за узкой специализации утилиты получаются очень маленькими по размеру и, как следствие, — идеально отлаженными.

> ***Замечание***
>
> Конечно, пользователю Windows это кажется неудобным. Для утилит Windows принят другой подход — "все в одном" (all in one). В результате получается объемный комплекс, трудный для анализа и отладки. Он потребляет также значительный объем оперативной памяти и процессорного времени.

Кроме того, все утилиты Linux способны взаимодействовать друг с другом. Это значит, что с помощью утилит можно организовать цепочку взаимосвязанных операций. А скрипты командной оболочки позволяют создать инструментарий для выполнения часто встречающихся последовательностей операций. Linux требует от пользователя достаточно обширных знаний и некоторого размышления (планирования) перед выполнением нетривиальных действий. С одной стороны, после Windows с ее бездумным нажиманием кнопок мыши это несколько напрягает. Но с другой стороны, это позволяет точно, шаг за шагом, понять процесс получения нужного результата, заодно и память потренировать.

Стандартный ввод/вывод, перенаправление

Во многих операционных системах существует понятие стандартного устройства (устройства ввода, вывода и отображения ошибок). Эти устройства можно задать самостоятельно. По умолчанию используется клавиатура и терминал. Концепция стандартного ввода/вывода очень удобна для его автоматизации.

Перенаправить стандартный ввод позволяет символ перенаправления <.

Например, команда

```
mysql <2.sql
```

передает программе `mysql` данные, содержащиеся в файле 2.sql.

Для перенаправления стандартного вывода предназначены символы > и >>. В чем их отличие? Символ > не проверяет наличие файла, в который сохраняется стандартный вывод программы. Если такой файл существует, то его содержимое полностью заменяется выводом программы. Символ >> проверяет наличие файла, и если он существует, то вывод программы дописывается в конец существующего файла.

Можно перенаправить вывод на другой дескриптор:

```
command 2>&1
```

Эта команда перенаправит вывод ошибок в стандартный вывод.

Примеры:

```
df > 1.txt
ls –A >>1.txt
```

Операции перенаправления ввода/вывода можно применять одновременно.

Конвейер (поток)

Конвейер (поток, pipe) объединяет несколько команд в одну операцию и обозначается символом |. Применяется для передачи стандартного вывода одной программы на стандартный ввод другой программы. В одной командной строке допустимы несколько операций конвейера.

Пример:

```
ls | grep
```

Команды

Операционная система Linux очень многое наследует от UNIX, в том числе и бóльшую часть команд и утилит. Конечно, эти команды адаптированы и усовершенствованы, но, тем не менее, в целом они сохранили синтаксис соответствующих команд UNIX. Поэтому, по большому счету, не важно, в чем вы работаете: в UNIX или в Linux — система команд на 98% совпадает. Далее мы рассмотрим наиболее часто используемые команды и утилиты. Обратите внимание, что в данной главе упоминаются далеко не все команды и утилиты. Никакая самая объемистая книга не отменяет команду `man`, документацию и файлы HOWTO. За время написания любой книги выходят новые версии программ, и зачастую их возможности кардинально изменяются. Поскольку глава обзорная, то в описании команд не всегда (или не в полном объеме) приводятся ключи и параметры вызова.

Дата, время

cal

Команда `cal` выводит на консоль календарь, содержание которого зависит от параметров.

Календарь выводится:

- на текущий месяц — при отсутствии параметров;
- на заданный месяц — при указании месяца и года;
- на заданный год — при указании года.

Пример:

```
cal
    Апрель 2011
Вс Пн Вт Ср Чт Пт Сб
                 1  2
 3  4  5  6  7  8  9
10 11 12 13 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29 30
```

date

Команда `date` выводит текущие дату и время в указанном формате, а также позволяет изменять системные дату и время.

Параметры:

- `+` — формат отображения времени и даты в указанном формате;
- `-s` — установка времени и даты;
- `-u` — вывод времени и даты по Гринвичу.

При установке даты и времени можно указывать их как в числовом, так и в нечисловом формате. При числовом формате строка данных записывается в следующем виде:

```
MMddhhmmyy
```

где:

- `MM` — месяц;
- `dd` — день;
- `hh` — часы;
- `mm` — минуты;
- `yy` — две последние цифры года.

Пример:

```
date
Сбт Апр  9 19:57:30 EEST 2011
```

Более подробную информацию можно получить по команде `man date`.

Файлы и каталоги

В этом разделе представлены команды и утилиты, которые напрямую взаимодействуют с файлами и каталогами.

Административные команды

Здесь собраны команды, которые отвечают за "административную работу" с файлами и каталогами.

chgrp

Команда `chgrp` изменяет одну группу каждого заданного файла на другую, которая может быть представлена как именем группы, так и ее числовым идентификатором (GID). Более подробную информацию можно получить по команде `man chgrp`.

chmod

Команда `chmod` изменяет права доступа файла так, как указано в параметре, который может быть представлен восьмеричным числом или в символьном виде.

Формат символьного режима:

```
[ugoa...][[+-=][rwxXstugo...]...][,...]
```

Здесь каждый аргумент — это список символьных команд изменения прав доступа, разделенных запятыми. Каждая такая команда начинается с какой-нибудь из букв `u`, `g`, `o` или `a` (впрочем, букв может вообще не быть) или их комбинации, которая указывает, чьи права доступа к файлу будут изменены:

- `u` — владельца;
- `g` — группы;
- `o` — других пользователей, не входящих в данную группу;
- `a` — всех пользователей. Буква `a` эквивалентна `ugo` и действует по умолчанию;
- `+` — добавляет выбранные права доступа к уже имеющимся;
- `-` — удаляет эти права;
- `=` — присваивает только эти права файлу.

Буквы `rwxXstugo` выбирают новые права доступа для пользователя, заданного одной из букв `ugoa`:

- `r` — чтение;
- `w` — запись;

- `x` — выполнение;
- `X` — выполнение, если файл является каталогом или уже имеет право на выполнение для какого-нибудь пользователя;
- `S` — `setuid` или `setgid`-бит;
- `t` — `sticky`-бит;
- `u` — установка для остальных таких же прав доступа, какие имеет пользователь, владеющий этим файлом;
- `g` — установка для остальных таких же прав доступа, какие имеет группа файла;
- `o` — установка для остальных таких же прав доступа, какие имеют остальные пользователи.

Установка `sticky`-бита для каталога приводит к тому, что только владелец файла и владелец этого каталога могут удалять файл из каталога.

В операционной среде Linux, если на файле установлен бит `setgid`, но не установлен бит выполнения группой, то блокировки этого файла становятся жесткими (mandatory), в отличие от обычных — информационных (advisory). Подробная информация по этому вопросу находится в файле /usr/src/linux/Documentation/mandatory.txt.

Числовой режим состоит из четырех восьмеричных цифр, которые складываются из битовых масок 4, 2 и 1. Любые пропущенные разряды дополняются лидирующими нулями:

- первая цифра выбирает установку идентификатора пользователя — `setuid` (4), идентификатора группы — `setgid` (2) или `sticky`-бита (1);
- вторая цифра выбирает права доступа для пользователя, владеющего данным файлом: чтение (4), запись (2) и выполнение (1);
- третья цифра выбирает права доступа для пользователей, входящих в группу;
- четвертая цифра выбирает права доступа для остальных пользователей, не входящих в группу.

Эту команду может применять либо владелец файла, либо пользователь root.

Более подробную информацию можно получить по команде `man chmod`.

chown

Команда `chown` изменяет владельца и/или группу для заданного файла.

В качестве имени владельца/группы берется первый аргумент, не являющийся опцией. Возможные варианты:

- задано только имя пользователя (или его числовой идентификатор) — данный пользователь становится владельцем каждого из указанных файлов, а группа этих файлов не изменяется;
- за именем пользователя через двоеточие следует имя группы (или ее числовой идентификатор) без пробелов между ними — изменяется также и группа файла;
- за именем пользователя следует двоеточие, но группа не задана — данный пользователь становится владельцем указанных файлов, а группа указанных файлов изменяется на основную группу пользователя;
- имя пользователя опущено, а двоеточие или точка вместе с группой заданы — будет изменена только группа указанных файлов.

Как и предыдущие команды, ее может применять либо владелец файла, либо пользователь root.

chroot

Команда `chroot` доступна только пользователю root, который с помощью команды

```
chroot имя_каталога
```

делает каталог корневым. Администратор использует эту команду для повышения безопасности системы.

Более подробную информацию можно получить по команде `man chroot`.

lockfile

Команда `lockfile` создает специальные семафорные файлы.

mknod

Команда `mknod` создает именованный канал (FIFO), специальный символьный или специальный блочный файл (файл устройства).

Специальный файл именуется с помощью тройки параметров: один логический и два целых. Логический параметр говорит о том, является ли специальный файл символьным или блочным. Два целых параметра задают старший и младший номера устройства.

Специальный файл практически не занимает места на диске и необходим только для общения с операционной системой, а не для хранения данных. Часто специальные файлы указывают на аппаратные устройства или на службы операционной системы.

Специальные блочные файлы обычно являются устройствами, подобными диску. Все другие устройства — это специальные символьные файлы.

Аргумент, следующий за именем, задает тип файла, который нужно создать:

- `p` — для FIFO;
- `b` — для блочного специального файла;
- `c` — для символьного специального файла.

В файле /usr/src/linux/Documentation/devices.tex находится список устройств, где есть имена устройства, тип, старший и младший номер.

Более подробную информацию можно получить по команде `man mknod`.

Общие команды

В этом разделе собраны команды, тем или иным способом воздействующие на файлы и каталоги.

cat

Команда выводит на экран содержимое файла, начиная с первой строки.

cd

Команда `cd` встроена в `bash` и предназначена для смены текущего каталога.

Пример:

```
cd /var/log
```

Приведенная команда делает текущим каталог `/var/log`.

cp

Команда `cp` копирует файлы или каталоги. Если последний аргумент является существующим каталогом, то команда `cp` копирует каждый файл в этот каталог. Если задано только два имени файла, то команда `cp` копирует первый файл во второй.

Права доступа к файлам и каталогам станут равны тем, что были на оригинальных файлах, но биты `sticky`, `setuid` и `setgid` будут сброшены.

Пример:

```
cp /home/user1/test /home/user2/1.txt
```

Команда копирует файл /home/user1/test в файл /home/user2/1.txt.

Более подробную информацию можно получить по команде `man cp`.

dir

См. команду `ls`.

file

Команда `file` определяет тип (или принадлежность к определенному процессу) файла. Иногда для этих целей используется файл /usr/share/magic.

Листинг 14.1 содержит пример.

Листинг 14.1

```
file file.c
       file.c:  C program text

file -s /dev/hda{,1,2,3,4,5,6,7,8,9,10}
       /dev/hda:   x86 boot sector
       /dev/hda1:  Linux/i386 ext3 filesystem
       /dev/hda2:  x86 boot sector
       /dev/hda3:  x86 boot sector, extended partition table
       /dev/hda4:  Linux/i386 ext3 filesystem
       /dev/hda5:  Linux/i386 swap file
       /dev/hda6:  Linux/i386 swap file
       /dev/hda7:  Linux/i386 swap file
       /dev/hda8:  Linux/i386 swap file
       /dev/hda9:  empty
       /dev/hda10: empty
```

Более подробную информацию можно получить по команде `man file`.

find

Команда `find` осуществляет поиск файлов. Имеет много параметров, позволяющих проводить как простой поиск, так и поиск со многими условиями.

Более подробную информацию можно получить по команде `man find`.

head

Команда `head` выводит на экран первые 10 строк файла. С помощью параметров можно изменить размер выводимой части.

ln

Команда `ln` создает ссылки на файлы. По умолчанию создаются жесткие ссылки, а при указании опции `-s` делаются символические ссылки.

Если задан только один файл, то для него делается ссылка в текущем каталоге с таким же именем, как у этого файла. В противном случае, если последний аргумент является именем существующего каталога, то команда `ln` создаст ссылки в этом каталоге для каждого из файлов с такими же именами, как и у исходных файлов. В случае если задано два имени, то создается ссылка (второе имя) на файл (первое имя).

По умолчанию команда `ln` не удаляет существующие файлы или символьные ссылки.

Пример:

```
ln make test
```

Создает жесткую ссылку с именем test на файл make.

Более подробную информацию можно получить по команде `man ln`.

locate

Команда `locate` выполняет быстрый поиск в базе данных имен файлов системы.

ls

Команда `ls` выводит содержимое каталога. Эта команда сначала выдает список всех файлов, перечисленных в командной строке, а затем — список всех файлов, находящихся в каталогах, перечисленных в командной строке. Если не указано ни одного файла, то по умолчанию аргументом назначается текущий каталог.

Каждый список файлов сортируется отдельно в алфавитной последовательности текущих региональных настроек (locale).

Результат направляется в стандартный вывод, по одному файлу на строку.

При наличии ключа `-l` информация выдается в следующем виде:

- тип файла;
- права доступа к файлу;
- число ссылок на файл;
- имя владельца;
- имя группы;
- размер файла;
- временной штамп;
- имя файла.

Типы файлов могут принимать следующие значения:

- `-` — для обычного файла;
- `d` — для каталога;
- `b` — для блочного устройства;
- `c` — для символьного устройства;

- `l` — для символической ссылки;
- `p` — для FIFO;
- `s` — для сокета.

Листинг 14.2 иллюстрирует пример.

Листинг 14.2

```
ls -l
итого 124
-rw-rw-r--    1 alst     alst          665 Окт  6 16:09 cd
-rw-rw-r--    1 alst     alst          665 Окт  6 16:09 cdd
-rw-rw-r--    1 alst     alst         4005 Окт  6 16:08 chgrp
-rw-rw-r--    1 alst     alst         6909 Окт  6 16:08 chmod
-rw-rw-r--    1 alst     alst         3668 Окт  6 16:08 chown
-rw-rw-r--    1 alst     alst         1126 Окт  6 16:08 chroot
-rw-rw-r--    1 alst     alst        12508 Окт  6 16:10 cp
drwxr-xr-x    2 alst     alst         4096 Авг 31 10:29 Desktop
-rw-rw-r--    1 alst     alst        16011 Окт  6 16:10 file
-rw-rw-r--    1 alst     alst        17248 Окт  6 16:10 find
-rw-rw-r--    1 alst     alst         8497 Окт  6 16:10 ln
-rw-rw-r--    1 alst     alst         2550 Окт  6 16:11 locate
-rw-rw-r--    1 alst     alst         7228 Окт  6 16:09 locfile
-rw-rw-r--    1 alst     alst            0 Окт  6 16:11 lss
-rw-rw-r--    1 alst     alst         3917 Окт  6 16:09 mknod
drwx------    2 alst     alst         4096 Сен  8 16:03 nsmail
-rw-rw-r--    1 alst     alst          978 Окт  6 16:11 uptime
-rw-rw-r--    1 alst     alst           62 Окт  6 16:11 uptm
```

Более подробную информацию можно получить по команде `man ls`.

mc

Команда `mc` запускает на выполнение файловый менеджер Midnight Commander, который позволяет выполнять множество операций с файлами и каталогами и имеет огромное число команд и настроек. Исчерпывающую информацию о Midnight Commander можно получить из его справочной системы, вызываемой нажатием клавиши <F1>.

mkdir

Команда создает каталоги с заданными именами. По умолчанию права доступа к каталогам устанавливаются в 0777 за вычетом битов, установленных в `umask`.

Пример:

```
mkdir test
```

Более подробную информацию можно получить по команде `man mkdir`.

mkfifo

Команда `mkfifo` создает именованные каналы (FIFO) с указанными именами. FIFO — это специальный тип файла, который позволяет общаться независимым процессам. Один процесс открывает FIFO-файл для записи, а второй для чтения, после чего данные могут передаваться как в обычных неименованных каналах в `shell`.

Более подробную информацию можно получить по команде `man mkfifo`.

mv

Команда `mv` перемещает или переименовывает файлы или каталоги.

Если последний аргумент является именем существующего каталога, то команда `mv` перемещает все указанные файлы в этот каталог. Если задано два файла, то имя первого файла будет изменено на имя второго.

Пример:

```
mv /tmp/test /home/user1
```

Команда перемещает файл test из каталога /tmp в каталог /home/user1.

Более подробную информацию можно получить по команде `man mv`.

pwd

Команда `pwd` выводит имя текущего каталога.

Пример:

```
pwd
/home/alst
```

rm

Команда `rm` удаляет файлы или каталоги. По умолчанию каталоги не удаляются, но если заданы опции `-r` или `-R`, то будет удаляться все дерево вложенных каталогов.

Пример:

```
rm *.tmp
```

Удаляет все TMP-файлы из текущего каталога.

Более подробную информацию можно получить по команде `man rm`.

rmdir

Команда `rmdir` удаляет пустые каталоги. Если каталог не пуст, то будет выдано сообщение об ошибке. Для удаления непустых каталогов используйте команду

```
rmdir –r
```

size

Команда `size` выводит размеры сегментов программы, указанной в командной строке.

Пример:

```
size /sbin/agetty
 text      data      bss      dec      hex   filename
 10819      844    10336    21999      55ef  agetty
```

slocate

Команда `slocate` — это более защищенный вариант команды `locate`.

Листинг 14.3 иллюстрирует пример.

Листинг 14.3

```
locate dir

/var/run/runlevel.dir
/var/www/icons/dir.gif
/var/www/icons/small/dir.gif
/var/www/icons/small/dir2.gif
/etc/X11/applnk/Games/xpuzzles/.directory
/etc/X11/xdm/authdir
...
/usr/src/linux-2.4.3/net/tux/redirect.c
/bin/mkdir
/bin/rmdir
/home/alst/.kde/Autostart/.directory
/home/alst/Desktop/.directory
/lib/security/pam_mkhomedir.so
/root/.kpackage/dir
```

Более подробную информацию можно получить по команде `man slocate`.

split

Команда `split` предназначена для разбиения файла на несколько частей. По умолчанию создаются части размером в 1000 строк.

stat

Команда `stat` показывает информацию о файле или файлах, заданных в командной строке.

Пример иллюстрирует листинг 14.4.

Листинг 14.4

```
stat /sbin/agetty

  File: "agetty"
  Size: 13148      Blocks: 32         Regular File
Access: (0755/-rwxr-xr-x)     Uid: ( 0/ root)  Gid: ( 0/ root)
Device: 302        Inode: 350883     Links: 1
Access: Sat Oct  6 20:10:19 2010
Modify: Fri Jul 13 01:22:17 2010
Change: Fri Aug 31 07:44:08 2010
```

Более подробную информацию можно получить по команде `man stat`.

tac

Команда `tac` выводит содержимое файла в обратном порядке, от первой строки к последней.

tail

Команда `tail` выводит на экран последние 10 строк файла. С помощью параметров можно изменить размер выводимой части.

vdir

См. описание команды `ls`.

Сеть

dig

Эта команда служит для формирования запросов о доменах DNS-серверам. Имеет много управляющих параметров, информация о которых содержится в соответствующей документации.

elm

Команда `elm` — это интерактивная почтовая программа, имеющая больше возможностей, чем `mail`.

finger

Команда `finger` получает информацию о пользователе, а также содержимое его файлов .plan и .project. Можно указать пользователя, задав его системный идентификатор, имя или фамилию. Обычно для увеличения безопасности системы администраторы не устанавливают на своих системах finger-серверы.

ftp

Команда `ftp` позволяет соединиться с удаленной системой посредством протокола FTP. После установки соединения можно копировать файлы между локальной и удаленной системами, удалять файлы и просматривать каталоги, если имеются соответствующие права доступа к удаленной системе.

Эта команда — самый простой FTP-клиент. Для эффективной работы необходимо знать команды FTP-протокола. В современных системах используют либо графические FTP-клиенты, либо текстовые с удобным интерфейсом (например, встроенный в `mc`).

getty (mgetty)

Команда позволяет модему осуществлять исходящие звонки и принимать входящие. Имеет гибкую конфигурацию. Более подробную информацию можно получить по соответствующей команде `man`.

host

Команда `host` выводит IP-адрес указанной системы, задействуя службу DNS. Можно указать IP-адрес, и он будет преобразован в имя системы.

hostname

Команда `hostname` выводит имя локальной системы. Привилегированный пользователь может с ее помощью задать системе новое имя.

ipchains

Команда `ipchains` предназначена для создания, сопровождения и проверки правил IP-брандмауэра (firewall) ядра операционной системы Linux.

Существует четыре вида правил:

- входящие;
- исходящие;
- правила IP-маршрутизации (forwarding);
- пользовательские.

Для каждого из этих правил создается отдельная таблица.

Команда несколько устарела, сейчас заменяется на `iptables`.

Более подробную информацию можно получить по команде `man ipchains` и из соответствующей литературы.

iptables

Команда `iptables` служит для создания, сопровождения и проверки правил IP-брандмауэра (firewall) ядра операционной системы Linux. Эта команда сегодня заменила команду `ipchains`.

Более подробную информацию можно получить по команде `man iptables` и из соответствующей литературы.

kppp

Программа, входящая в состав KDE. Позволяет максимально легко настроить модемное PPP-соединение с провайдером. Обладает понятным и удобным графическим интерфейсом.

lynx

Команда `lynx` запускает текстовый браузер Интернета. Он не позволяет выводить графические изображения или различные шрифты. Обычно применяется для просмотра Web-документов (в формате HTML), находящихся локально на компьютере.

mail

Команда `mail` реализует чтение и отправку электронной почты. Имеет простой аскетичный интерфейс, занимает мало места на жестком диске. Используется редко.

mimencode

Команда `mimencode` позволяет производить кодировку и раскодировку MIME — формата сообщений электронной почты.

minicom

Программа `minicom` предназначена для интерактивной работы с модемами — организации соединения, настройки последовательного порта, чата и т. п. Полезна для отладки модемного соединения. Имеет богатые возможности по автоматизации.

netcfg

Утилита `netcfg` входит в пакет linuxconf. Эта программа с помощью текстового меню позволяет быстро и просто сконфигурировать параметры системы, тем или иным образом относящиеся к сетевым настройкам.

netstat

Выводит информацию о сетевых соединениях, таблицы маршрутизации, статистику по сетевым интерфейсам и т. п. Используется для отладки и мониторинга сети.

nslookup

Утилита для получения различной информации от DNS-серверов. Обычно применяется для получения расширенной информации о хосте либо для обнаружения и устранения неполадок в конфигурации сети.

pine

Текстовая программа `pine` для чтения и отправки электронной почты и новостей Usenet. Она поддерживает MIME-кодирование и позволяет отправлять письма с MIME-содержанием. Имеет много функциональных возможностей, удобный интерфейс.

ping

Команда `ping` отправляет ICMP-пакеты ECHO_REQUEST на указанную систему (IP-адрес или символическое имя системы) для определения прохождения пакетов. Простое, но незаменимое средство диагностики сети. Пример иллюстрирует листинг 14.5.

Листинг 14.5

```
ping
PING 127.0.0.1 (127.0.0.1) from 127.0.0.1 : 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=0 ttl=255 time=214 usec
64 bytes from 127.0.0.1: icmp_seq=1 ttl=255 time=69 usec
64 bytes from 127.0.0.1: icmp_seq=2 ttl=255 time=29 usec
64 bytes from 127.0.0.1: icmp_seq=3 ttl=255 time=30 usec

--- 127.0.0.1 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max/mdev = 0.029/0.085/0.214/0.076 ms
```

Более подробную информацию можно получить по команде `man ping`.

procmail

Команда `procmail` осуществляет сортировку приходящей почты. Для ее вызова обычно применяется механизм перенаправления почты при помощи файла .forward. Кроме того, `procmail` можно настроить для работы в сочетании с почтовой программой.

ssh

Команда `ssh` (secure shall) призвана заменить устаревшие и небезопасные команды `telnet` и r-команды. При работе создает шифрованный туннель между системой, к которой подключаются, и подключаемой системой.

telnet

Команда `telnet` позволяет установить соединение с удаленной системой с использованием протокола Telnet. Считается небезопасной, поскольку пересылает по сети логин и пароль в незашифрованном виде. Сегодня рекомендуется применение SSH или OpenSSH.

traceroute

Команда `traceroute` определяет маршрут следования пакетов от вашего хоста до указанного вами хоста. Применяется как в целях отладки маршрутизации (если в обслуживании большая группа сетей), так и в познавательных целях, например, чтобы определить, почему внутри одного города между провайдерами так долго проходят пакеты. Иногда оказывается, что пакеты передаются не через внутреннюю точку обмена трафиком, а через город на другом континенте.

Более подробную информацию можно получить по команде `man traceroute`.

uudecode

Команда `uudecode` выполняет UU-декодирование файла из формата, подходящего для пересылки по электронной почте (для кодирования допустимы только цифры и латинские символы).

uuencode

Команда `uuencode` выполняет UU-кодирование файла в формат, подходящий для пересылки по электронной почте (для кодирования допустимы только цифры и латинские символы).

wget

Программа `wget` предназначена для загрузки файлов по протоколу HTTP и включена во все основные дистрибутивы. Также может обрабатывать FTP, временные метки (date stamps), рекурсивно отражать полное дерево каталогов Web-сайта и др.

Кроме того, `wget` позволяет возобновлять прерванное задание, если задан незавершенный файл, к которому добавляются оставшиеся данные.

Очень мощная программа, полную информацию по которой можно получить из соответствующей документации.

Администрирование

at

Команда `at` позволяет однократно запустить на выполнение команду или группу команд в назначенное время. Эти команды не должны требовать ввода информации

с консоли. Как правило, такие команды удобны для архивации данных, создания резервной копии данных и т. п.

Более подробную информацию можно получить по команде `man at`. Рекомендуется также `man crontab`.

atq

Команда `atq` выводит список всех заданий, поставленных на выполнение командой `at`.

atrm

Команда `atrm` позволяет удалить задания из очереди команды `at`.

batch

Команда `batch`, подобно команде `at`, также позволяет выполнять задания, но, в отличие от нее, не определяет четкого времени выполнения заданий. Вместо этого критерием запуска команд является минимальная загрузка операционной системы.

cksum

Команда `cksum` вычисляет контрольную сумму (CRC) указанных файлов.

crond

Автоматически запускает программы в указанное время. Использует файлы crontab.

crontab

Программа, позволяющая просматривать и редактировать файлы crontab.

getkeycodes

Команда `getkeycodes` выводит таблицу соответствия скан-кодов кодам клавиш.

ifconfig

Необходима для конфигурации сетевых интерфейсов при отладке сетевых соединений. При вызове без параметров команда отображает статус активных интерфейсов.

Пример иллюстрирует листинг 14.6.

Листинг 14.6

```
lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:14 errors:0 dropped:0 overruns:0 frame:0
          TX packets:14 errors:0 dropped:0 overruns:0 carrier:0
```

```
          collisions:0 txqueuelen:0

ppp0      Link encap:Point-to-Point Protocol
          inet addr:195.114.131.239  P-t-P:195.114.128.4  Mask:255.255.255.255
          UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1500  Metric:1
          RX packets:174301 errors:31 dropped:0 overruns:0 frame:0
          TX packets:98860 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:3
```

insmod

Утилита `insmod` позволяет добавлять модули в ядро без его перекомпиляции. При подключении нового модуля рекомендуется предварительно воспользоваться программой `modprobe`.

lsapnp

Утилита считывает конфигурационный файл, в котором определены настройки устройств ISA PnP, и конфигурирует их в ядре операционной системы Linux.

Поскольку в современных чипсетах шина ISA отсутствует, команда интересна только владельцам устаревших компьютеров.

kill

Команда `kill` отправляет процессу с указанным идентификатором сигнал. Часто используется для завершения работы процессов. Команда доступна только владельцу процесса или привилегированному пользователю.

killall

Команда `killall` завершает работу всех процессов с указанным именем.

lilo

`lilo` — системный загрузчик, который загружает операционные системы, в частности Linux. Так же применяется для собственного администрирования.

Параметры загрузчика:

- `-v` — повысить уровень информативности;
- `-q` — показать текущую карту загрузки. `lilo` сохраняет в файле /boot/map имена и расположение на диске ядер, предназначенных для загрузки;
- `-m файл-карты` — использовать карту загрузки, находящуюся в указанном файле;
- `-C файл-настроек` — `lilo` читает настройки из файла настроек /etc/lilo.conf. Этот параметр позволяет указать другой файл в качестве файла настроек;
- `-d` — задержка. Если в настройках `lilo` указано несколько ядер, и во время загрузки нажата клавиша <Shift>, загрузчик спросит, какое из ядер нужно загрузить. После указанной задержки будет загружено первое ядро из списка. Этот параметр указывает время задержки в десятых долях секунды;
- `-D` — метка. Использовать по умолчанию ядро с указанной меткой, а не первое ядро, указанное в файле настроек;

- `-r каталог` — сделать указанный каталог корневым. Служит для восстановления записи в MBR;
- `-t` — тестирование конфигурации. Не записывает на диск новый загрузочный сектор и файл карты. Применяется вместе с параметром `-v`, чтобы узнать, что собирается сделать `lilo`;
- `-c` — разрешает уплотнение карты. Запросы на чтение к смежным секторам будут объединяться. Ускоряет загрузку;
- `-f disk-tab` — определяет, в каком файле будет храниться геометрия диска (по умолчанию /etc/disktab);
- `-i загрузочный_сектор` — определяет файл, который будет использован как новый загрузочный сектор. (По умолчанию /boot/boot.b.);
- `-l` — вместо адресов типа "сектор/головка/цилиндр" `lilo` будет генерировать линейные адреса секторов;
- `-P {fix|ignore}` — исправлять (`fix`) или игнорировать (`ignore`) поврежденные таблицы разделов;
- `-s резервный_файл` — когда `lilo` переписывает загрузочный сектор, его старое содержимое помещается в специальный резервный файл — /boot/boot.NNNN, где NNNN зависит от выбранного устройства. Этот параметр позволяет задать другой резервный файл для загрузочного сектора или в комбинации с параметром `-u`, определяет, откуда восстанавливать загрузочный сектор;
- `-S резервный_файл` — обычно `lilo` не переписывает существующий резервный файл. Этот параметр разрешает перезапись;
- `-u имя_устройства` — деинсталлировать `lilo` путем копирования резервного файла назад в загрузочный сектор;
- `-U имя_устройства` — то же самое, но без проверки времени сохранения;
- `-R командная_строка` — этот параметр устанавливает загрузчику команду, которая выполнится при следующей загрузке. Сразу после загрузки операционной системы загрузчик стирает эту строку. Обычно она встречается в сценариях при перезагрузке;
- `-I метка` — имя используемого ядра после загрузки помещается в переменную окружения `BOOT_IMAGE`. Эта команда выведет путь к файлу, соответствующему указанной метке, на стандартное устройство;
- `-V` — напечатать номер версии.

Более подробную информацию можно получить по команде `man lilo.conf`.

linuxconf

В этой программе для конфигурирования и администрирования системы собраны практически все системные настройки. Обладает удобным текстовым (и графическим) интерфейсом. Однако использование ее нежелательно, т. к. она не всегда корректно выставляет права доступа на важные файлы.

md5sum

Команда `md5sum` рассчитывает контрольную сумму файла или строки по алгоритму MD5. Рекомендуется для проверки целостности файлов, полученных через

Интернет, и для организации безопасности файловой системы (как один из элементов контроля целостности системы).

Более подробную информацию можно получить по команде `man md5sum`.

modprobe

Команда `modprobe` загружает и тестирует модули ядра. Полезна при установке и конфигурировании нестандартного оборудования.

Более подробную информацию можно получить по команде `man modprobe`.

mount

Эта команда предназначена для монтирования файловых систем пользователем root. В качестве аргументов принимает файл устройства и точку монтирования. Имеет много дополнительных параметров.

Пример:

```
mount /dev/fd0 /mnt/floppy
```

Более подробную информацию можно получить по команде `man mount`.

nice

Команда `nice` позволяет изменить приоритет запускаемой команды. Значение приоритета может быть выбрано от 15 (низший) до –20 (высший). Значения меньше нуля может устанавливать только пользователь root.

При запуске без параметров команда `nice` выводит значение приоритета по умолчанию.

passwd

Эта команда позволяет изменить пароль, который установлен для входа в систему. Если запускается `passwd` без параметров, то система запросит сначала старый, а затем новый пароль.

Пользователь root может изменить пароль любого пользователя системы, указав его системный идентификатор (логин) и новый пароль в командной строке.

pnpdump

Команда `pnpdump` совместно с `isapnp` позволяет сконфигурировать устройства ISA PnP. Она сканирует все ISA-платы и выводит занятые ими ресурсы на стандартный вывод. На основании полученных данных можно правильно настроить устройства ISA PnP.

renice

Эта команда изменяет приоритет одного или нескольких запущенных процессов.

rpm

`rpm` — менеджер пакетов, который используется для сборки, установки, инспекции, проверки, обновления и удаления программных пакетов.

Можно выбрать один из следующих основных режимов:

- инициализация базы данных;
- перестроение базы данных;
- сборка пакетов;
- рекомпиляция пакетов;
- сборка пакетов из TAR-архивов;
- запрос;
- показ полей запроса;
- установка;
- обновление;
- удаление;
- верификация;
- проверка подписи;
- повторная подпись;
- добавление подписи;
- установка владельцев и групп;
- показ конфигурации.

Подробную информацию по `rpm` *см. в главе 8.*

rmmod

Команда `rmmod` удаляет загружаемые модули из ядра, если они не используются ядром или другими модулями.

Более подробную информацию можно получить по команде `man rmmod`.

setserial

Команда `setserial` позволяет получить или установить настройки последовательного порта.

Пример:

```
setserial ttyS0
```

Более подробную информацию можно получить по команде `man setserial`.

setterm

Команда `setterm` позволяет установить атрибуты терминала. Как правило, этой командой пользоваться не придется.

Более подробную информацию можно получить по команде `man setterm`.

skill

Эта команда отправляет сигналы или изменяет приоритет указанного процесса. По умолчанию отправляется сигнал `TERM`.

snice

Команда `snice` позволяет изменить приоритет запущенного процесса. По умолчанию новый приоритет равен +4. Приоритет можно задать явно в виде параметров `+приоритет` или `–приоритет`.

strace

Команда `strace` осуществляет трассировку системных вызовов и сигналов. Служит для запуска определенной программы. После этого `strace` будет выполнять трассировку системных вызовов и сигналов, соответствующих программе процессов. Информация выводится на экран или сохраняется в файле. Команда `strace` обычно удобна при отладке программ.

stty

Команда позволяет выводить и изменять настройки терминала. Без параметров выводит установки терминала, на котором она запущена.

Пример:

```
stty

speed 0 baud; line = 0;
-brkint -imaxbel
```

umount

Команда `umount` предназначена для размонтирования файловых систем. Аргументом выступает точка монтирования файловой системы или файл устройства.

Пример:

```
umount /mnt/floppy
```

Более подробную информацию можно получить по команде `man umount`.

useradd

Утилита позволяет создавать в операционной системе нового пользователя. При вводе пользователя ему можно задать группу, пароль и некоторые другие параметры.

xf86config

Команда `xf86config` создает файл конфигурации Xf86config, используемый X-сервером.

xvidtune

Команда `xvidtune` позволяет точно настроить видеорежимы X-сервера. При задании параметров в командной строке команда `xvidtune` может переключать видеорежимы, а также устанавливать время отключения мониторов, имеющих расширенное управление питанием. Команда `xvidtune` без параметров открывает диалоговое окно, дающее возможность выполнять настройку видеорежима. Полученные режимы могут быть описаны в виде, позволяющем вставить их в файл Xf86Config.

zic

Утилита, позволяющая компилировать бинарный файл временной зоны. Источником информации служит текстовый файл описания временной зоны. Практиче-

ски никогда не используется администратором системы, поскольку в каталоге /usr/share/zoneinfo/ находятся скомпилированные файлы временных зон на все случаи жизни.

Состояние системы

df

Команда `df` выдает информацию о доступном и занятом дисковом пространстве на файловых системах.

Вызов команды без аргументов выдает отчет для всех файловых систем, которые смонтированы в данный момент. По умолчанию все размеры выдаются в блоках по 1024 байта, за исключением случая, когда установлена переменная `POSIXLY_CORRECT` (тогда размер блока соответствует POSIX-версии).

Пример:

```
df
Filesystem           1k-blocks      Used Available Use% Mounted on
/dev/hda2              4134932   1607188   2317696  41% /
/dev/hda1              4008372   1085892   2922480  28% /mnt/floppy
```

du

Команда `du` выдает отчет об использовании дискового пространства указанными файлами или каталогами. Под "использованным дисковым пространством" понимается пространство, занятое под всю иерархию каталогов указанного каталога.

Запущенная без аргументов, команда `du` выдает отчет о дисковом пространстве для текущего каталога.

Размеры дискового пространства указываются в блоках по 1024 байта (если размер не задан посредством опций), за исключением случая, когда задана переменная окружения `POSIXLY_CORRECT`.

Пример выполнения команды в каталоге /root иллюстрирует листинг 14.7.

Листинг 14.7

```
du
16  ./.gnome/accels
4  ./.gnome/apps
20  ./.gnome/panel.d/default/launchers
52  ./.gnome/panel.d/default
56  ./.gnome/panel.d
4  ./.gnome/nautilus-scripts
8  ./.gnome/gnome-vfs
4  ./.gnome/application-info
168  ./.gnome
.........
```

```
16  ./.ee/minis/root
20  ./.ee/minis
32  ./.ee
612  .
```

dumpkey

Команда `dumpkey` выводит информацию о драйвере клавиатуры.

free

Команда `free` выдает информацию об использовании оперативной памяти.

Пример иллюстрирует листинг 14.8.

Листинг 14.8

```
free
             total       used       free     shared    buffers     cached
Mem:        255532     227600      27932          0      66140      74568
-/+ buffers/cache:      86892     168640
Swap:       257000          0     257000
```

ftpcount

Команда `ftpcount` выдает текущее число пользователей в каждом классе, подключенных к FTP-серверу, используя определение классов из файла ftpaccess.

ftpwho

Команда `ftpwho` выводит информацию о пользователях, подключенных к FTP-серверу в данный момент.

kdb_mode

Эта команда выводит текущий режим драйвера клавиатуры и позволяет изменить его.

last

Команда `last` выводит список последних зарегистрировавшихся в системе пользователей.

Пример иллюстрирует листинг 14.9.

Листинг 14.9

```
last
alst     tty4                         Sun Nov  4 12:55   still logged in
alst     tty3                         Sun Nov  4 12:55 - 12:56  (00:00)
alst     tty2                         Sun Nov  4 12:54   still logged in

wtmp begins Sun Nov  4 12:54:36 2001
```

ps

Команда `ps` выводит разнообразную информацию о процессах операционной системы (листинг 14.10).

Листинг 14.10

```
ps -A
  PID TTY          TIME CMD
    1 ?        00:00:04 init
    2 ?        00:00:00 keventd
    3 ?        00:00:00 kapm-idled
    4 ?        00:00:00 kswapd
    5 ?        00:00:00 kreclaimd
    6 ?        00:00:00 bdflush
    7 ?        00:00:00 kupdated
    8 ?        00:00:00 mdrecoveryd
    . . . . . . . . . ..
  741 tty1     00:00:00 login
  742 tty1     00:00:00 bash
  781 tty1     00:00:00 mc
  782 ?        00:00:00 cons.saver
  783 pts/0    00:00:00 bash
  802 tty2     00:00:00 bash
  837 tty2     00:00:00 mc
  838 ?        00:00:00 cons.saver
  839 pts/1    00:00:00 bash
 1292 pts/1    00:00:00 ps
```

Более подробную информацию можно получить по команде `man ps`.

quota

Команда `quota` отображает ограничения на использование дискового пространства пользователями.

Более подробную информацию можно получить по команде `man quota`.

tload

Команда выводит график загрузки системы.

top

Команда `top` выводит список процессов в системе, отсортированных в порядке убывания используемого процессорного времени. Неплохой инструмент для определения подозрительных процессов.

Пример иллюстрирует листинг 14.11.

Листинг 14.11

```
top

4:19pm up 13 min, 2 users, load average: 0,01, 0,02, 0,00
37 processes: 36 sleeping, 1 running, 0 zombie, 0 stopped
CPU states: 1,0% user, 1,0% system, 0,0% nice, 97,8% idle
Mem:319968K av, 50468K used, 269500K free, 0K shrd, 4164K buff
Swap: 216868K av, 0K used, 216868K free, 29524K cached
PID USER     PRI  NI  SIZE  RSS SHARE STAT %CPU %MEM  TIME COMMAND
 1  root       8   0   544  544   472 S     0,0  0,1   0:04 init
 2  root       8   0     0    0     0 SW    0,0  0,0   0:00 keventd
 3  root       9   0     0    0     0 SW    0,0  0,0   0:00 kapm-idled
 4  root       9   0     0    0     0 SW    0,0  0,0   0:00 kswapd
 5  root       9   0     0    0     0 SW    0,0  0,0   0:00 kreclaimd
 6  root       9   0     0    0     0 SW    0,0  0,0   0:00 bdflush
 7  root       9   0     0    0     0 SW    0,0  0,0   0:00 kupdated
61  root       9   0     0    0     0 SW    0,0  0,0   0:00 khubd
364 root       9   0   600  600   500 S     0,0  0,1   0:00 syslogd
369 root       9   0  1060 1060   460 S     0,0  0,3   0:00 klogd
383 rpc        9   0   596  596   504 S     0,0  0,1   0:00 portmap
398 rpcuser    9   0   772  772   668 S     0,0  0,2   0:00 rpc.statd
470 root       8   0   532  532   464 S     0,0  0,1   0:00 apmd
519 root       9   0   648  648   544 S     0,0  0,2   0:00 automount
531 daemon     9   0   584  584   508 S     0,0  0,1   0:00 atd
546 root       9   0  1136 1136   948 S     0,0  0,3   0:00 sshd
566 root       9   0   992  992   788 S     0,0  0,3   0:00 xinetd
```

uptime

Команда выводит информацию о системе: число работающих пользователей, среднюю загрузку системы, время, прошедшее с момента запуска операционной системы.

Пример:

```
uptime
  4:11pm  up 5 min,  2 users,  load average: 0.04, 0.04, 0.01
```

users

Команда `users` выводит данные о пользователях, подключенных в настоящий момент к системе. Для получения этой информации используется файл /etc/utmp.

who

Эта команда выводит сведения о системе или о пользователе. Если параметры не указаны, то выводится информация о пользователях, зарегистрированных в системе.

Пример:

```
who
alst     tty1     Oct  6 14:13
root     tty2     Oct  6 14:18
```

Более подробную информацию можно получить по команде `man who`.

w

Команда `w` выводит информацию о системе: список пользователей, подключенных к системе, статистику работы системы, а также выполняемые пользователями задачи. Эта команда является комбинацией команд `who`, `ps`, `-a` и `uptime`.

Создание файловой системы

fdisk

Утилита для создания, изменения и удаления дисковых разделов. Обычно применяется во время инсталляции операционной системы или при подключении нового диска.

fdformat

Команда `fdformat` осуществляет низкоуровневое форматирование дискеты в формате FAT.

mkfs

С помощью утилиты `mkfs` создается файловая система. Обычно работает совместно с утилитой `fdisk`. При использовании этой утилиты необходимо определить тип файловой системы и количество необходимых блоков. Более подробную информацию можно получить по соответствующей команде `man`.

Диагностика файловой системы

fsck

Утилита `fsck` обычно нужна при загрузке операционной системы для проверки и восстановления файловых систем. Более подробную информацию смотрите в документации.

Архивация

gzip

Программа, осуществляющая сжатие файла по алгоритму Лемпела—Зиффа. В отличие от аналогов в MS-DOS или Windows, может сжимать только один файл. Для архивации нескольких файлов в один архив необходимо воспользоваться утилитой `tar`.

tar

Утилита, предназначенная для изготовления из нескольких файлов/каталогов одного файла архива. При этом компрессия файлов не производится. Первоначально применялась для записи файлов на ленточный накопитель.

Работа с текстовыми файлами

joe

Команда `joe` запускает простой, гибкий и удобный текстовый редактор.

sort

Команда `sort` сортирует, объединяет или сравнивает строки текстовых файлов. Результат выводится на экран.

uniq

Команда `uniq` удаляет повторяющиеся строки из первого файла и выводит результат во второй файл.

vi

Команда `vi` запускает один из старейших текстовых редакторов, установленный практически на всех UNIX-системах. В настоящее время вместо оригинального редактора `vi` обычно вызывается `vim` или `elvis`.

vim

Текстовый редактор, запускаемый по команде `vim`, — это `vi`-совместимый редактор для обработки текстовых файлов. Более "продвинутый", с большей функциональностью и меньшими ограничениями.

Помощь

apropos

Команда `apropos` ищет заданное ключевое слово в базе `whatis`.

man

Команда `man` форматирует и выводит справочные страницы для команд, функций и тому подобных вещей. Справочные страницы `man` являются официальным руководством и имеют жестко заданный формат.

Более подробную информацию можно получить по команде `man man`.

whatis

Команда `whatis` представляет собой мини-справочную систему. В качестве аргумента указывается имя файла, на выходе — строка информации об этом файле.

Пример:

```
whatis du
du                    (1)  — estimate file space usage
```

Разное

banner

Команда `banner` выводит слева направо строку, рисуя буквы при помощи символа звездочки `*`.

bash

Эта команда запускает интерпретатор командной строки Bourne Again Shell (модификацию интерпретатора командной строки `sh`). Является интерпретатором командной строки по умолчанию.

bc

Команда `bc` представляет собой калькулятор, позволяющий проводить вычисления с произвольной точностью. Также имеется возможность преобразования чисел из одной системы счисления в другую.

chvt

Команда обеспечивает переключение на указанную виртуальную консоль. Имеет смысл, если в системе более двенадцати виртуальных консолей.

clear

Команда `clear` очищает экран в текстовом режиме.

cpp

Команда `cpp` запускает препроцессор, используемый C-компилятором для преобразования программы перед началом компиляции.

csh

Эта команда запускает C shell — один из интерпретаторов командной строки в Linux.

echo

Команда `echo` выводит текст или значения переменных на стандартное устройство (обычно на экран). Существуют три практически одинаковых варианта: команда Linux /bin/echo, а также команды echo-интерпретаторов командной строки C shell и Bourne Again Shell.

env

Команда `env` устанавливает значения переменных окружения на время выполнения указанной команды или выводит значения переменных окружения на экран.

Операционная система Linux имеет набор переменных окружения для различных ситуаций. Например, большинство программ, которым для работы нужен текстовый редактор, используют утилиту, заданную в переменной окружения `EDITOR`. Другие переменные определяют установленный по умолчанию интерпретатор командной строки, тип терминала, путь, домашний каталог пользователя и т. д.

g77

`g77` — это компилятор программ на языке Fortran. Язык был разработан фирмой IBM специально для математических расчетов. Современные программисты редко его используют, но осталось обширное "наследие" программного обеспечения от прошлых времен (по крайней мере, на Западе), которое необходимо сопровождать.

gawk

Программа `gawk` представляет собой GNU-версию языка программирования AWK.

gcc

Программа `gcc` — компилятор языков программирования C и C ++, предназначенный для Linux. Существует для большинства версий UNIX и для других операционных систем, что облегчает перенос программного обеспечения (и экономит деньги, поскольку программа бесплатная).

id

Команда `id` выводит информацию об указанном пользователе: системный идентификатор пользователя, его номер, идентификаторы и номера групп, к которым принадлежит пользователь.

login

Команда `login` осуществляет вход в операционную систему, выполняет некоторые административные задачи, такие как установка UID- и GID-терминала, а также уведомляет пользователя о наличии почты. Кроме того, данная команда позволяет пользователю root вход в систему только с определенных терминалов, список которых находится в файле /etc/securetty.

logname

Эта команда выводит имя пользователя, которому принадлежит вызывающий ее процесс. Для его определения используется файл /etc/utmp.

make

Команда `make` управляет группой файлов, из которых создается программа.

Для определения зависимостей между файлами и командами команда `make` использует созданный пользователем файл правил. По умолчанию это файл Makefile.

nohup

Программа `nohup` позволяет продолжить выполнение указанной в той же строке команды после выхода пользователя из операционной системы.

Обычно необходима для программ, которые качают большие объемы информации из Интернета или выполняют длительные расчеты.

openvt

Утилита, позволяющая создавать текстовую консоль (до 64). Целесообразна в том случае, если окажется недостаточно стандартных шести виртуальных консолей. Совместно указывают опции и выполняемую команду, для которой создается консоль.

perl

PERL — это сокращение от Practical Extraction and Report Language, интерпретируемого языка программирования, обычно применяемого для написания системными администраторами различных скриптов, призванных автоматизировать и упростить ежедневные операции администратора. Очень популярный инструмент создания CGI-скриптов для Web-сайтов.

printenv

Эта команда выводит значения переменных окружения. Если в командной строке указана переменная, то выдается ее значение, в противном случае выводятся значения всех переменных окружения.

reset

Эта команда выполняет начальную инициализацию терминала.

resizecons

Утилита позволяет изменить разрешение текстовой консоли (стандартное — 80 символов в строке, 25 строк на экране) в достаточно большом диапазоне.

startx

Команда `startx` предназначена для запуска X Window из командной строки.

После запуска `startx` осуществляется поиск файла .xinitrc в домашнем каталоге пользователя. Этот файл содержит информацию о настройках системы X Window, а также о том, какие X-клиенты должны быть запущены. Большинство этих клиентов запускаются как фоновые процессы, за исключением последнего клиента в списке, который обычно является диспетчером окон.

strings

Эта команда выполняет поиск текстовых строк в файле. По умолчанию выводятся только строки, длина которых составляет не менее четырех символов.

strip

Команда `strip` удаляет таблицы символов из объектных файлов. Список объектных файлов может включать библиотеки, но должен быть указан, по крайней мере, один объектный файл. Позволяет уменьшить размеры исполняемых файлов и библиотек.

subst

Команда `subst` производит в файлах указанные подстановки. Обычно она применяется для настройки программного обеспечения под конкретную систему. Каждый из указанных файлов изменяется в соответствии с содержимым файла подстановок.

Файл подстановок содержит по одной подстановке на строке. Строка подстановки состоит из двух полей, разделенных одним или несколькими символами табуляции. Первое поле строки представляет подстановку, второе — значение. Ни одно из полей не должно содержать символа `#`. Строки, начинающиеся с `#`, считаются комментариями и игнорируются.

su

Команда `su` запускает интерпретатор командной строки с правами указанного пользователя. Обычно она применяется в административных целях для временного входа под именем пользователя root. Запускаемый интерпретатор командной строки берется из файла /etc/passwd для указанного пользователя. Если этот пользователь имеет пароль, то команда `su` запросит его.

true

Эта команда возвращает код возврата, равный нулю, что означает успешное выполнение.

yes

Команда непрерывно выводит указанную строку, разделяя две выводимые строки символом новой строки.

Если строка не указана, то выводится символ `y`. Данная команда обычно используется для того, чтобы передать ее стандартный вывод программе, на все вопросы которой следует ответить утвердительно.

Ссылки

- Соответствующие страницы руководства man.
- **www.linuxdocs.org** — разнообразная документация, включая HOWTO.
- Соответствующие HOWTO:
  - iptables-HOWTO;
  - NAT-HOWTO.



Часть V

Настройка и сервисы Linux

Глава 15



Локализация

Еще лет пятнадцать назад нормальным явлением в компьютерном мире было почти полное отсутствие русского, украинского, белорусского и тому подобных языков в большинстве операционных сред и программ. Знание пользователем английского технического (правильнее сказать — "компьютерного") языка считалось само собой разумеющимся. Такой порядок вещей обусловливался множеством факторов, и в первую очередь тем, что популярные операционные системы производились американскими компаниями и были рассчитаны на англоговорящую аудиторию. С той поры компьютер стал массовым явлением, а наш среднестатистический компьютерный пользователь в большинстве своем английским языком так и не овладел.

> ***Замечание***
>
> Чтобы постоянно не перечислять здесь все множество языков, основанных на кириллице, в дальнейшем мы станем упоминать в этом контексте лишь русский язык, но подразумевать будем, разумеется, и все остальные.

Большинство коммерческих программ и операционных систем в той или иной мере русифицированы. Что же в этом плане может предложить Linux? Как известно, "спасение утопающих — дело рук самих утопающих", и поскольку Linux некоммерческая операционная система, ее локализацию выполняют сами пользователи. В последние годы усилиями наших русскоговорящих разработчиков, а также фирм-производителей дистрибутивов и многочисленных энтузиастов, большинство коробочных иностранных дистрибутивов (не говоря уже о русских и украинских) непосредственно после инсталляции могут корректно работать с кириллицей, вплоть до того, что на русский язык переведен и интерфейс большинства программ и значительная часть справочной информации.

Тем не менее администратору необходимо знать, как можно локализировать операционную систему Linux.

Поскольку Linux-сообщество велико и разнородно, а программы портировались с различных операционных систем, привести их к одному знаменателю для нормальной локализации, к сожалению, весьма затруднительно. Однако современные тенденции таковы, что в мире Linux назревает осознание необходимости принятия стандартов на ключевые технологии, в частности, написания программ, локализуемых с минимальными усилиями. Но об этом позже.

А сейчас определим, что нам прежде всего потребуется от хорошо локализованной системы:

- ❒ корректная текстовая консоль (правильные шрифты, ввод и вывод кириллицы);
- ❒ должным образом настроенная система X Window;
- ❒ правильный вывод кириллицы на принтер;
- ❒ настроенная на кириллицу система проверки правописания;
- ❒ локализованные основные программы (редакторы, офисные пакеты, словари и т. п.).

Теоретическая часть

Стандарты кодировки

Как известно, символ (минимальный элемент алфавита) и представление этого символа в компьютере (кодировка) — две разные вещи.

Кодировкой называется совокупность уникальных символов, которые система способна распознать как самостоятельную сущность.

Поскольку первоначально компьютеры были разработаны за рубежом (в Великобритании и США) и не предназначались для экспорта, производители не озаботились поддержкой языков, отличных от английского. Со временем это вызвало определенные проблемы, однако они решались хаотично и без учета перспектив дальнейшего развития программ и аппаратуры. В результате русскоязычное компьютерное сообщество получило несколько различных кодировок, в той или иной степени учитывающих национальную специфику.

Стандарт ASCII

Наиболее популярной кодировкой была (и фактически ей и остается) кодировка ASCII (Американский стандартный код для Информационного обмена).

Стандарт ASCII, иногда называемый 7-битовый ASCII, включает в себя 128 уникальных символов. Они подразделяются на отображаемые символы и символы управления, бóльшая часть которых использовалась в старых протоколах связи. Каждому элементу набора соответствует целочисленный символьный код от 0 до 127.

Со временем 7-битовый стандарт ASCII был расширен до 8-битового ASCII (расширенный ASCII). Этот стандарт подразумевает наличие 256 символов, которые соответствуют кодам от 0 до 255. Первая часть таблицы — от 0 до 127 не претерпела по сравнению с предыдущим стандартом никаких изменений, а во второй половине таблицы… пусто. Дело в том, что 8-битовый стандарт ASCII не определяет содержание второй половины таблицы кодировки. Для этого Международная организация по стандартизации (ISO) выпустила серию стандартов (известных как семейство ISO 8859-*x*), устанавливающих кодировку второй половины таблицы для различных языков. Нас как пользователей кириллицы интересуют следующие кодовые страницы:

- ❒ 8859-0 — новый европейский стандарт (Latin 0);
- ❒ 8859-1 — Европа, Латинская Америка (Latin 1);

- 8859-2 — Восточная Европа;
- 8859-5 — кириллица.

В кодовой странице 8859-1 (Latin 1) старшая половина таблицы определяет символы, которые не входят в английский алфавит, но присутствуют в различных европейских языках. Соответственно, в остальных кодировках в старшей половине таблицы находятся специфические национальные символы, входящие в алфавит указанного региона.

Есть еще одна реализация расширенного ASCII — *кодовая страница IBM*. Эта кодировка в старшей половине содержит псевдографические символы.

Казалось бы, вполне достаточный набор стандартов. Однако есть несколько отрицательных моментов:

- ограничение набора модифицируемых символов (128);
- недопустимость совмещения в одном чисто текстовом документе нескольких кодировок.

Альтернативная кодировка (CP866)

Альтернативная кодировка (CP866) — это кодовая страница IBM, где все специфические европейские символы заменили на буквы из кириллицы, оставив нетронутыми псевдографические символы.

Кодировка Microsoft CP1251

Кодовая страница Microsoft CP1251 — это попытка Microsoft облегчить труд программиста. Предназначена для кодировки кириллицы в Windows. Устраняет проблему с сортировкой по алфавиту, связанную с тем, что в странице CP866 буквы русского алфавита располагались не подряд.

Стандарт КОИ8

Стандарт разработан достаточно давно, когда была распространена 7-битовая кодировка символов ASCII. Разработчики КОИ8 поместили символы русской кириллицы в верхней части расширенной таблицы ASCII таким образом, чтобы позиции кириллических символов соответствовали их фонетическим аналогам в английском алфавите в нижней части таблицы. Так, если в тексте, хранящемся в кодировке КОИ8, убрать старший (восьмой) бит каждого символа, то получится текст, написанный английскими символами в русской транскрипции. Например, предложение "Мама мыла раму" после удаления старшего бита будет выглядеть так: "Mama myla ramu".

Существует несколько реализаций стандарта КОИ8, в частности KOI8-R — для русского языка, KOI8-U — для украинского.

Стандарт RFC 1489 Registration of a Cyrillic Character Set, созданный Андреем Черновым, регламентирует представление русскоязычных документов в Интернете, где KOI8-R давно уже стал фактическим стандартом для русской кириллицы.

Unicode

Unicode — частичная реализация стандарта ISO 10646, в котором первые 256 символов соответствуют кодировке Latin-1 (ISO 8859-1). Основная идея этого стандарта — кодирование символа с переменным числом байтов (до 8). На данном этапе применяется двухбайтовое кодирование символа, позволяющее определить 65 535 символов. В настоящее время позиции зарезервированы за буквами практически всех известных алфавитов, включая древнеегипетские иероглифы, благодаря чему можно, имея всего один шрифт, писать одновременно на русском и греческом, английском и иврите и делать еще вставки на японском. Используется в Windows 98 и более поздних версиях. В UNIX-системах поддержка Unicode реализована практически полностью.

Украинский язык

Специфика локализации для Украины состоит в том, что зачастую нужны и украинский, и русский языки одновременно (бóльшая часть жителей городов Центральной, Южной и Восточной Украины — русскоговорящие, а официальный язык — украинский).

Кириллизация консоли

В большинстве современных дистрибутивов "кириллизация" консоли происходит при инсталляции. Однако необходимо рассмотреть способы кириллизации текстового режима как фундамента, на котором держится локализация операционной системы в целом.

Консольный драйвер

Для настройки консоли можно воспользоваться пакетами console-tools, Cyrillic console-tools (модификация console-tools с расширенным набором шрифтов и дополнительными свойствами) или kbd.

Схема функционирования консольного драйвера

Для понимания дальнейших действий необходимо четко представлять, как функционирует консольный драйвер.

В Linux применяются две таблицы символов — таблица символов приложения (Application Charset Map, ACM) и таблица экранных шрифтов (Screen Font Map, SFM).

Когда программа предлагает консольному драйверу вывести на экран символ, имеющий код, например A, то консольный драйвер прежде ищет код A в таблице символов приложения. Из нее он узнает, какой код B согласно кодировке Unicode соответствует коду A. Далее консольный драйвер ищет код B в таблице экранных шрифтов. Из нее он узнает, какой символ активного шрифта имеет код B, и выводит его на экран. А используемую операционной системой кодировку посредством таблицы символов приложения определяет пользователь.

Аппаратные ограничения видеокарт VGA не допускают в текстовом режиме шрифты, имеющие более 512 символов. Поэтому иногда консольный драйвер не может найти код B в таблице экранных шрифтов. В этом случае используется так называемая fallback-таблица. Она определяет для кода B возможные его аппроксимации B1, B2 и т. д. Например, если B является кодом символа "левая двойная угловая кавычка", то, возможно, B1 будет кодом символа "левая одинарная угловая кавычка", а B2 будет просто кодом символа <.

Настройка поддержки кириллицы с помощью пакетов `console-tools` и `kbd` состоит из:

- настройки экранного шрифта и таблицы экранных шрифтов. Это делается с помощью программы `consolechars` (для console-tools) или `setfont` и `mapscrn` (для kbd);
- настройки таблицы символов приложения и fallback-таблицы;
- загрузки соответствующей раскладки клавиатуры с помощью программы `loadkeys`.

Файлы шрифтов обычно размещаются в каталогах /usr/share/consolefonts или /usr/lib/kbd/consolefonts, символьные таблицы — в каталоге /usr/share/consoletrans, клавиатурные раскладки — в /usr/share/keymap/i386/qwerty.

console-tools

Если на компьютере установлен пакет console-tools, то необходимо выполнить следующие команды:

```
loadkeys ru.map
consolechars -v -f Cyr_a8x16 -m $foo/koi2alt
```

Раскладку клавиатуры переключают нажатием правой клавиши <Ctrl> (иногда это можно сделать нажатием клавиши <Alt> или <Caps Lock>).

Cyrillic console tools

Все шрифты в этом пакете основаны на альтернативной кодировке (CP866). Это сделано потому, что в текстовом режиме VGA использование другой кодировки приводит к разрывам в отображении горизонтальной псевдографики. Все шрифты содержат в себе таблицу отображения в Unicode.

В пакет также включены таблицы перекодировки в Unicode из распространенных кодировок русского, белорусского, болгарского, сербского и украинского языков.

Для настройки консоли нужно выполнить следующие команды:

```
consolechars -f UniCyr_8x16.psf -m koi8-r.acm
loadkeys console_russian.map
```

Для украинизации вместо `koi8-r` необходимо подставить `koi8-u`.

kbd

Настройку кириллицы с помощью `kbd` обычно осуществляют следующие команды:

```
loadkeys /usr/lib/kbd/keytables/ru.map
setfont /usr/lib/kbd/consolefonts/Cyr_a8x16
mapscrn /usr/lib/kbd/consoletrans/koi2alt
```

```
# ниже идет "магическая" последовательность
echo -ne "\033(K"
```

Во время загрузки системы для русификации всех виртуальных текстовых консолей необходимо выполнить команду

```
echo -ne "\033(K"
```

семь раз. Это можно сделать с помощью следующей строки:

```
for i in 1 2 3 4 5 6 7; do echo -ne "\033(K" > /dev/tty$i; done
```

"Магическая" последовательность необходима для перекодировки вывода символов на экран при наличии шрифтов, основанных на кодовой странице CP866.

Настройка консольных приложений

После настройки консоли необходимо настроить и консольные программы, которые работают с символами. Основная проблема большинства этих программ — они "считают", что задана 7-битовая кодировка символа.

bash

В файле .inputrc, находящемся в домашнем каталоге пользователя, необходимо установить следующие три переменные:

```
set meta-flag on
set convert-meta off
set output-meta on
```

Эти строки указывают, что символ кодируется 8-битовой последовательностью.

Поскольку .inputrc является конфигурационным файлом библиотеки `GNU readline`, внесенные исправления кириллизируют не только `bash`, но и другие программы, использующие `GNU readline`.

csh/tcsh

Те же действия в отношении программ `csh/tcsh` будут выглядеть следующим образом.

В файле .inputrc, находящемся в домашнем каталоге пользователя, нужно установить три переменные:

```
set meta-flag on
set convert-meta off
set output-meta on
```

А в файл .cshrc необходимо добавить две строки:

```
setenv LC_CTYPE iso_8859_5
stty pass8
```

zsh

В файле .zshrc необходимо добавить следующие строки:

```
setenv LC_CTYPE iso_8859_5
stty pass8
```

less

Для нормального функционирования программы `less` в файл ~/.lesskey необходимо добавить:

```
LESSCHARSET=
```

Это позволяет программе игнорировать установку переменной `LESSCHARSET=` другими программами. Затем нужно запустить `lesskey` для получения бинарного файла ~/.less.

mc (The Midnight Commander)

Чтобы установить кириллицу в `mc`, нажатием клавиши <F9> зайдите в системное меню, выберите пункт меню **Options | Display** и установите опцию **full 8 bits**.

nroff

Для корректной работы `nroff` с кириллицей необходимо запускать его с ключом `Tlatin1`.

man

Если программа `man` не желает корректно отображать кириллицу на экране, необходимо правильно настроить `less`.

Также измените в файле /usr/lib/man.conf строку:

```
NROFF        /usr/bin/groff -S -Tascii -mandoc
```

на

```
NROFF       /usr/bin/groff -S -Tlatin1 -mandoc
```

ls

При неправильно настроенной локали `ls` не будет выводить кириллические символы. В этом случае поможет одна из следующих команд:

```
ls -N;
ls --show-control-chars;
```

Samba

Чтобы увидеть кириллические символы в именах файлов, в файл /etc/smb.conf нужно добавить следующие строчки:

```
[global]
character set = koi8-r
client code page = 866
preserve case = yes
short preserve case = yes
```

Первые две строки указывают кодировку пользователя (`character set = koi8-r`) и кодировку имен файловой системы (`client code page = 866`).

Третья и четвертая строки определяют, что необходимо сохранять регистр длинных и коротких имен файлов.

telnet

При возникновении проблемы ввода русских символов необходимо создать файл ~/.telnetrc, содержащий следующую строку:

```
DEFAULT set outbinary
```

Локализация и интернационализация

Как можно заметить, каждая из перечисленных программ требует своего особого подхода, выражающегося в том или ином изменении индивидуальных конфигурационных файлов. Это сильно раздражает. А проблема возникла потому, что при проектировании программ не учитывались какие-либо национальные особенности.

Решение таких проблем основывается на двух базисных концепциях: локализации (Localization, l1on) и интернационализации (Internationalization, i18n).

Под *локализацией* подразумевается написание программного кода, способного адекватно воспринимать и обрабатывать различающиеся стандарты представления данных для разных стран. Например, формат записи даты в США имеет вид ММ/ДД/ГГ, в странах СНГ — ДД.ММ.ГГ, а в Японии — ГГ.ММ.ДД. Помимо даты необходимо обеспечить правильное представление форматов времени, чисел, валюты и т. п. Кроме того, базовый аспект локализации — определение соответствующих классов символов.

Интернационализация должна решать проблемы, связанные со способностью программ взаимодействовать с пользователем на его родном языке.

Обе эти концепции должны быть стандартизованы, давая программистам непротиворечивый путь создания программ, работающих в национальной среде.

Локаль

Одно из основных понятий локализации — локаль (locale). Под локалью подразумевается набор соглашений, специфичных для конкретного языка в отдельно взятой стране.

Каждая локаль определяет, по меньшей мере, следующие соглашения:

- классификация символов и преобразований;
- представление валюты;
- представление чисел;
- формат даты/времени.

Настройка локали

Локализация включается путем задания переменной окружения `LANG` строкой:

```
export LANG={язык}
```

В том случае, если такой строки не существует, используется значение локализации по умолчанию: `LANG="C"` или `LANG="POSIX"`.

По стандарту POSIX.2 язык локализации записывается в форме:

```
language_TERRITORY.Codeset
```

где:

- `language` — двухсимвольный код, обозначающий язык (ru, fr и т. д.);
- `TERRITORY` — двухсимвольный код, обозначающий страну (RU, UA и т. д.);
- `Codeset` — определяет кодировку символов.

Стандарт ISO 639 определяет коды языков, ISO 3166 — коды стран.

Для русского языка переменная `LANG` устанавливается, как правило, равной `LANG="ru_RU.KOI8-R"` или `LANG="ru_RU.ISO_8859-5"`.

По стандарту допустимы также короткие именования значений локали, которые часто выступают в качестве псевдонимов. Наиболее известная пара псевдоним—наименование: "C" — "POSIX".

Локализацию можно провести частично либо определить для отдельных ее категорий значения, отличные от общесистемной локализации. Посмотреть текущие значения категорий локализации можно утилитой `locale` (без параметров). В табл. 15.1 приведены опции локали.

Таблица 15.1. *Опции локали*

Опция	Описание
LC_ALL	Задает значение для всех опций (не рекомендуется использовать)
LC_CTYPE	Определяет одиночные символы
LC_NUMERIC	Задает формат чисел
LC_TIME	Определяет формат времени
LC_COLLATE	Используется для сравнения строк
LC_MONETARY	Задает формат валюты
LC_MESSAGES	Системные сообщения
LC_PAPER	Задает формат бумаги
LC_NAME	Задает формат имен
LC_ADDRESS	Задает формат адресов
LC_TELEPHONE	Задает формат телефонов

Интернационализация

Интернационализация детализирует способы общения программы с неанглоговорящим пользователем. Для этого при разработке ПО используют функции, специально предназначенные для создания интернационализированных программ. Эти функции и особенности их применения описываются документом LI18NUX 2000 Globalization Specification Version 1.0 with Amendment 4 Linux Internationalization Initiative (Li18nux).

Кириллизация X Window

X Window в современных дистрибутивах кириллизованы "из коробки". Поэтому необходимости в настройках нет.

Работа с текстом

Этот раздел посвящен программам, тем или иным способом обрабатывающим текст.

Проверка правописания

Одна из лучших программ проверки правописания для операционных систем UNIX — `ispell`. Добавив новые словари, с ее помощью можно проверять правописание текстов, написанных на языках, отличных от английского.

Для правильной работы `ispell` необходимо скомпилировать с поддержкой 8-битовых символов и установить словарь русских слов. О некоторых таких словарях рассказано далее.

Словарь Александра Лебедева

Словарь постоянно совершенствуется и дополняется и корректируется. Его отличительная черта — полноценная поддержка буквы ё. Последнюю версию словаря можно найти по адресу **ftp://mch5.chem.msu.su/pub/russian/ispell/rus-ispell.tar.gz**.

Словарь Константина Книжника

В поставку словаря включен скрипт, обеспечивающий инкрементный режим проверки правописания слов для emacs. Найти словарь можно по адресу **www.ispras.ru/~knizhnik**.

Редактор *vim*

После корректной настройки редактор `vim` нормально работает с кириллическими символами. Единственное неудобство: редактор понимает управляющие команды, набранные только в английской раскладке. Такое ограничение можно обойти, выполнив (для командного режима) отображение кириллических символов на английские с помощью опции `langmap`. Для этого достаточно добавить в файл .vimrc две строки:

```
set langmap=ж;;
set langmap=е`,йq,цw,уе,кr,еt,ну,гu,ші,що,зр,х[,ъ],фа,ыs,вd,аf,пg,рh,оj,лk,
дl,э',яz,чх,сс,мv,иb,тn,ьm,б\,,ю.,Е~,ЙQ,ЦW,УЕ,КR,ЕТ,НУ,ГU,ШІ,ЩО,ЗР,Х{,Ъ},ФА,
ЫS,ВD,АF,ПG,РН,ОJ,ЛК,ДL,Ж:,Э\",ЯZ,ЧХ,СС,МV,ИВ,ТN,ЬМ,Б<,Ю>
```

Редактор *joe*

Для того чтобы распознавать 8-битовые символы, `joe` требуется опция `-asis`. Ее можно указать в командной строке или вставить в файл .joerc.

Кириллица в программах электронной почты и чтения новостей

Для настройки программы электронной почты необходимо указать:

- что письма будут содержать 8-битовые символы;
- кодировку, в которой вы работаете;
- кодировку, в которой отсылаются письма.

elm

Добавьте следующую запись в файл ~/.elm/elmrc:

```
CHARSET=koi8-r
```

pine

Добавьте следующую запись в файл pine.conf для настройки всей системы:

```
character-set=koi8-r
```

Можно также изменить настройку pine, чтобы предотвратить посылку письма в кодировке quoted-printable:

```
enable-8bit-nntp-posting
enable-8bit-esmtp-negotiation
```

Чтобы настроить перекодировку win в koi в программе pine, в файле .pinerc следует прописать:

```
  display-filters=_CHARSET(iso8859-5)_ /usr/local/bin/icat,
          _CHARSET(utf-8)_ /usr/local/bin/ucat,
          _CHARSET(windows-1251)_ /usr/local/bin/wcat
```

Вместо программ icat, wcat и ucat можно воспользоваться другими, например iconv.

mutt

Добавьте следующую запись в файл .muttrc:

```
set charset=koi8-r
set allow_8bit
```

Для перекодировки посылаемых писем из одной кодировки в другую (символы отображаются в кодировке КОИ8, а сообщение посылается в кодировке CP1251) необходимо добавить следующие строки в файл .muttrc:

```
set charset= koi8-r
set send_charset= windows-1251
set allow_8bit
```

tin

Для включения отображения кириллицы в файл конфигурации .tin/tinrc добавьте следующие строки:

```
  post_mime_encoding=8bit
  mail_mime_encoding=8bit
```

Кириллические имена файлов

При стандартном монтировании разделов FAT32 созданные в Windows имена файлов с кириллическими символами видны как набор вопросительных знаков. Для решения этой проблемы необходимо при монтировании раздела указать коди-

ровку символов, в которой хранятся имена файлов, и кодировку, в которой необходимо эти имена файлов отображать.

Так, монтируя раздел FAT32, при вызове команды `mount` добавьте опции: `codepage=866`, `iocharset=koi8-r`.

Если компакт-диск содержит файлы с кириллическими шрифтами, укажите следующую команду монтирования:

```
mount -t iso9660 -o iocharset=koi8-r  /dev/cdrom  /mnt/cdrom
```

Поддержка кириллицы в Perl

Для того чтобы можно было правильно применять регулярные выражения в кириллических текстах, а также работать со стандартными функциями преобразования текста, в программу на Perl необходимо добавить следующие строки:

```
use locale;
use POSIX qw (locale_h);
setlocale(LC_CTYPE, 'ru_RU.KOI8-R');
```

Перекодировщики

Наиболее широко распространены перекодировщики `iconv` и `recode`. Для использования `iconv` следует указать в командной строке кодировку файла и кодировку, в которой необходимо сохранить файл. Например, при перекодировке из CP866 в КОИ8-R:

```
iconv -f866 -tKOI8-R -o<outfile> infile
```

Похожим образом вызывают и программу `recode`:

```
recode CP1251..KOI8-R  winfile.txt
```

Ссылки

- RFC 1489 — стандарт, описывающий кодировку KOI8-R.
- RFC 2319 — стандарт, описывающий кодировку KOI8-U.
- **www.unicode.org** — сайт, посвященный Unicode.
- **charts.unicode.org** — на этом сайте можно посмотреть набор символов Unicode.
- **www.sensi.org/~alec/** — сайт, посвященный локализации.
- **www.tsu.ru/~pascal/x_locale/** — сайт Ивана Паскаля: локаль и X Window.
- **www.inp.nsk.su/~baldin** — Балдин Евгений. The Linux Cyrillic HOWTO (rus). Здесь же расположен Cyrillic HOWTO (old rus), перевод устаревшего англоязычного документа.



Глава 16

Обновление и компиляция ядра

Системный администратор зачастую сталкивается с необходимостью обновления ядра операционной системы Linux. И возникает дилемма — искать новое ядро ОС в виде инсталляционного пакета или самостоятельно скомпилировать его из исходных текстов.

Рассмотрим более простой вариант — обновление ядра операционной системы Linux из пакета RPM, созданного специалистами Fedora (Red Hat).

Обновление ядра операционной системы Linux

Как правило, почти все производители дистрибутивов Linux выпускают обновленные пакеты программ, в том числе и ядра ОС. Это, правда, происходит с некоторой временной задержкой, да и не всегда в инсталляционных пакетах выходят все версии ядра операционной системы.

Дальнейшее описание процесса обновления ядра ОС будет основываться на компиляции ядра 2.6.*x*.

Подготовка к обновлению ядра операционной системы

Обязательно перед любыми действиями, затрагивающими жизнедеятельность системы, необходимо провести ряд мероприятий, позволяющих восстановить систему в случае краха. Для этого следует создать резервную копию, содержащую образ работоспособного ядра Linux.

Следующим этапом подготовки будет определение всех установленных пакетов, относящихся к ядру операционной системы. Для этого выполним команду:

```
rpm -qa | grep kernel
```

В результате вы получите что-то подобное:

```
kernel-headers-2.6.21
kernel-2.6.21
kernel-source-2.6.21
kernel-doc-2.6.21
```

На основании этого списка определим пакеты, которые необходимо получить из Интернета. Если у вас хороший канал, желательно загрузить и обновить все относящиеся к ядру установленные пакеты. Если же нет — загружаемые пакеты зависят от ваших намерений:

- для обновления ядра ОС — нужен только `kernel-2.6.xx`.;
- для перекомпилирования ядра ОС — потребуются пакеты `kernel-headers-2.6.xx`, `kernel-source-2.6.xx`.

Загрузить пакеты можно напрямую с FTP-сервера.

Обновление ядра операционной системы

Теперь, когда все необходимые пакеты получены, можно приступать к обновлению ядра операционной системы. Существуют два способа:

- команда `rpm -Uvh kernel-2.6.XX.i386.rpm` — обновить ядро ОС;
- команда `rpm -ivh kernel-2.6.xx.i386.rpm` — установить новое ядро ОС.

Второй способ позволит в случае, если новое ядро вызывает проблемы, безболезненно "откатиться" (roll back, downgrade) на старое ядро операционной системы.

Аналогично обновляются пакеты с исходными текстами ядра Linux.

Для проверки обновления ядра выполните команду:

```
ls -l /boot
```

Вы должны увидеть файл vmlinuz-2.6.*xx*.

После обновления ядра ОС следует сконфигурировать загрузчик (boot loader).

Помимо этого можно обновиться при помощи утилиты Yum. А если вы ничего не изменяли, то при старте графического интерфейса и при наличии подключения к Интернету запускается утилита автоматического обновления пакетов, в том числе и ядра системы.

Конфигурирование загрузчика

После установки нового ядра ОС необходимо сконфигурировать загрузчик таким образом, чтобы при последующих стартах операционной системы загружалось ее обновленное ядро (при автоматическом обновлении ядра это не обязательно).

> ***Предупреждение***
>
> Будьте предельно внимательны во время конфигурирования загрузчика — если вы ошибетесь, операционная система Linux не сможет загрузиться. В этом случае придется воспользоваться заблаговременно созданным загрузочным диском и внимательно переконфигурировать загрузчик.

GRUB

Если у вас установлен загрузчик GRUB, то вы должны отредактировать файл /boot/grub/grub.conf.

Типичный конфигурационный файл GRUB приведен в листинге 16.1.

Листинг 16.1

```
# NOTICE: You have a /boot partition. This means that
# all kernel paths are relative to /boot/
default=0
timeout=30
splashimage=(hd0,0)/grub/splash.xpm.gz
title Fedora Core Linux (2.6.21)
root (hd0,0)
kernel /vmlinuz-2.6.21 ro root=/dev/sda3
initrd /initrd-2.6.21.img
```

Добавлять новое ядро в список загрузчика рекомендуется в два этапа:

1. Сначала добавить новую секцию для нового ядра и убедиться, что загрузка происходит нормально. Проще всего скопировать существующую секцию и подправить ее в нужном месте. В результате получим текст листинга 16.2 (добавленная секция выделена полужирным шрифтом).

Листинг 16.2

```
# NOTICE: You have a /boot partition. This means that
# all kernel paths are relative to /boot/
default=0
timeout=30
splashimage=(hd0,0)/grub/splash.xpm.gz
title My new kernel (2.6.22)
root (hd0,0)
kernel /vmlinuz-2.6.22 ro root=/dev/hda3
initrd /initrd-2.6.22.img
title Fedora Core Linux (2.6.21)
root (hd0,0)
kernel /vmlinuz-2.6.21 ro root=/dev/sda3
initrd /initrd-2.6.21.img
```

После редактирования конфигурационного файла следует перезагрузить операционную систему и выбрать новое ядро.

2. Убедившись, что загрузка происходит без эксцессов и система функционирует нормально, удалите из конфигурационного файла описание старой версии ядра.

Компиляция ядра операционной системы Linux

Ядро операционной системы Linux может быть двух типов: монолитное и модульное.

Если драйверы почти всех устройств содержатся во внешних модулях, загружаемых по мере необходимости, то такое ядро *модульное*. С одной стороны, это несколько замедляет работу ядра, а с другой — нет необходимости заново собирать ядро при установке какого-нибудь нового аппаратного устройства.

Ядро, в которое вкомпилированы все необходимые драйверы, называют *монолитным*. Недостаток такого ядра — необходимость повторно собирать ядро ОС при появлении новых аппаратных устройств, а достоинство — сравнительно небольшой объем ядра и повышенная производительность операционной системы. Мы рассмотрим компиляцию и модульного, и монолитного ядра.

Для чего нужна повторная сборка ядра? Каковы аргументы "за" и "против"?

"За" компиляцию ядра операционной системы

Рассмотрим аргументы "за".

- Основная причина для самостоятельной компиляции ядра — *выход новых версий ядра* операционной системы Linux. Как правило, в них добавляют дополнительные функциональные возможности и исправляют замеченные ошибки. К сожалению, большинство сборщиков дистрибутивов отстают в выпусках "фирменных" ядер ОС, а иногда даже делают доступным новую версию ядра только с выходом новой версии дистрибутива.
- Пересборка ядра используется *для оптимизации ядра* Linux конкретно под имеющийся набор аппаратных средств (процессора, чипсета материнской платы, контроллеров, сетевых плат, видеокарт и т. п.).
- Компиляцию ядра осуществляют также *для включения специфических свойств ядра*, которые появляются в нем после наложения специальных "заплаток" (патчей), разрабатываемых отдельными программистами или группами. Обычно эти свойства связаны с безопасностью системы или с функционированием экзотических аппаратных средств. Наиболее известен вариант "альтернативного" ядра от Алана Кокса (Alan Kox).
- Самостоятельно компилировать *экспериментальные ядра с новыми возможностями* приходится потому, что разработчики дистрибутивов их не тестируют и не выпускают с ними инсталляционные пакеты. Следует помнить, что экспериментальные ядра не всегда стабильны, и с новыми возможностями можно получить набор ошибок, правда зачастую не критичных для функционирования системы.

"Против" компиляции ядра операционной системы

Аргументов "против" компиляции ядра ОС столько же (если не больше), сколько и "за".

- Чтобы скомпилировать ядро операционной системы, необходимо много сведений из различных областей администрирования: особенности настройки и функцио-

нирования сетей, поддержка аппаратуры, периферии, файловых систем, специфического программного обеспечения и др.

- При неправильно сконфигурированном или неверно установленном новом ядре ОС (это вытекает из предыдущего пункта) получаем проблемы вплоть до полной потери работоспособности операционной системы.
- Это противоречит принципу "работает — не трогай". Если ядро ОС работает устойчиво и его функционирование всех удовлетворяет — зачем его компилировать?
- Если вы решили уменьшить объем ядра ОС, изъяв все лишнее, не забудьте: почти все свойства ядра операционной системы вынесены в загружаемые модули, поэтому сэкономить удастся только 50–100 Кбайт.

Утилиты конфигурирования ядра операционной системы Linux

Если вы все-таки решились скомпилировать ядро Linux, перед вами встанет вопрос — как сконфигурировать его параметры? Можно, конечно, изучить исходные тексты ядра операционной системы, но это займет не менее одной-двух недель. Гораздо проще воспользоваться специальными утилитами:

- xconfig — простая, понятная и удобная в использовании утилита для конфигурирования параметров ядра в графической системе X Window;
- menuconfig — простая текстовая утилита с системой меню для конфигурации ядра (рис. 16.1). Передвигаясь по пунктам меню, достаточно удобно настраивать ядро ОС;

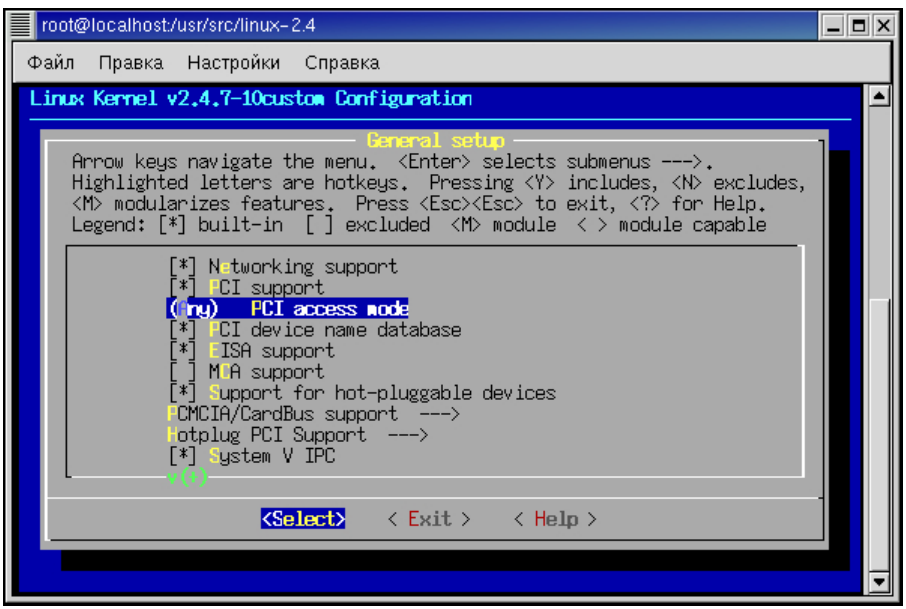


Рис. 16.1. Утилита menuconfig

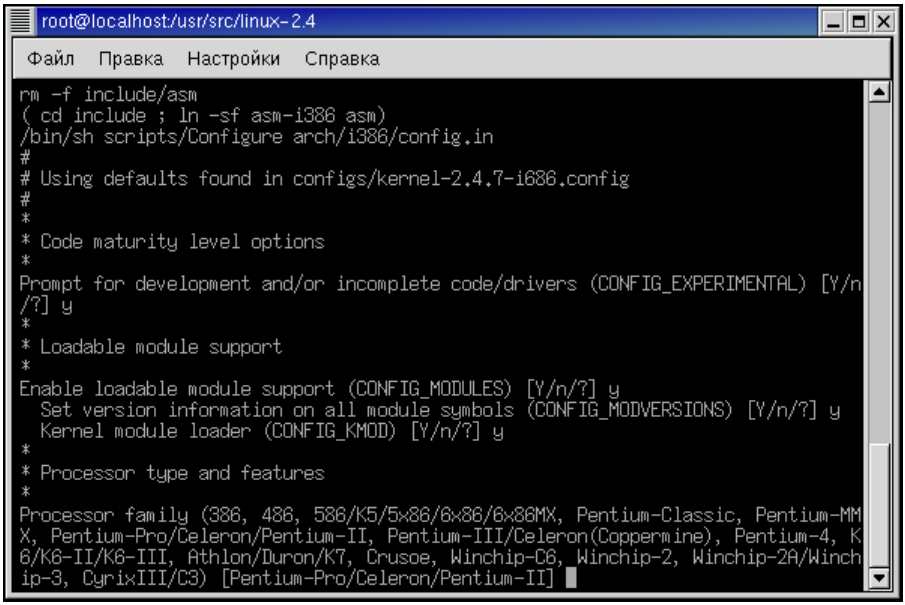


Рис. 16.2. Утилита config

- config — старейшая текстовая утилита для конфигурирования ядра операционной системы (рис. 16.2). Требует минимального количества ресурсов и библиотек, однако неудобна в эксплуатации. Нельзя вернуться назад и отредактировать предыдущие пункты, необходимо закончить конфигурирование (или прервать) и начать его заново. На сегодняшний день практически не применяется.

Процесс компиляции ядра

Как обычно, к компиляции ядра операционной системы следует предварительно подготовиться — создать загрузочную дискету. Для этого зайти в систему пользователем root и выполнить команду:

```
/sbin/mkbootdisk kernelversion
```

где `kernelversion` — версия ядра, полученная с помощью команды `uname`.

Далее необходимо получить из Интернета (или с инсталляционного компакт-диска) и установить пакеты kernel-headers и kernel-source.

Затем перейти в каталог /usr/src/kernels/2.6.27.XXXXXX и из него выполнять все дальнейшие команды.

Ядро с поддержкой загружаемых модулей (модульное)

Компиляция ядра Linux происходит в несколько этапов.

1. Необходимо быть уверенным, что после предыдущих компиляций ядра ОС в дереве исходных кодов не осталось каких-либо несуразностей типа исходных текстов ядра младшей версии или неубранных объектных файлов. Поэтому компиляцию ядра рекомендуется начать с команды `make mrproper`, которая не только

удаляет "мусор" после предыдущих компиляций, но и уничтожает конфигурационный файл ядра, находящийся в /usr/src/linux-2.6/.config. Если у вас уже есть рабочий конфигурационный файл (/usr/src/linux-2.6/.config), который вы хотите взять за основу, перед выполнением этой команды скопируйте его в свой домашний каталог, а после выполнения команды `make mrproper` верните на прежнее место.

2. Теперь следует сконфигурировать ядро операционной системы. Если у вас уже есть готовый конфигурационный файл, пропустите этот шаг. Для конфигурирования ядра, как мы уже указывали ранее, можно воспользоваться четырьмя разными утилитами, приводящими к одному результату:
   - `make xconfig` — для конфигурирования ядра операционной системы в среде X Window;
   - `make config` — простая текстовая утилита конфигурации ядра операционной системы;
   - `make menuconfig` — текстовая утилита с системой меню для конфигурации ядра Linux;
   - `make oldconfig` — неинтерактивный скрипт, который устанавливает в конфигурационном файле ядра значения по умолчанию.
3. После создания конфигурационного файла /usr/src/kernels/2.6.27.XXXXX/.config для корректной установки всех зависимостей выполняем команду make dep.
4. Для подготовки исходных текстов для компиляции выполняем команду make clean.
5. Теперь необходимо отредактировать файл /usr/src/kernels/2.6.27.XXXX/Makefile так, чтобы полученное новое ядро ОС не перекрыло старое (более подробную информацию смотрите в Kernel-HOWTO). Редактируем /usr/src/kernels/2.6.27.XXXXX/Makefile и исправляем строку, начинающуюся с `EXTRAVERSION=`, таким образом, чтобы создать уникальное имя. Самый простой вариант — добавить дату компиляции ядра. К примеру, `EXTRAVERSION= -0.1.6-jul2009`. Это позволит одновременно иметь старую и новую версии ядра операционной системы.
6. Компилируем ядро ОС командой `make bzImage`.
7. Компилируем модули ядра ОС командой `make modules`.
8. Устанавливаем модули операционной системы командой `make modules_install`. Эта команда должна установить модули ядра в каталог /lib/modules/KERNELVERSION/kernel/drivers, где KERNELVERSION — версия, описанная в файле Makefile. В нашем примере это /lib/modules/2.6.27-jul2009/kernel/drivers/.
9. Если в вашей системе установлен SCSI-контроллер, и вы сделали SCSI-драйвер модульным, необходимо создать новый файл initrd (см. далее).
10. Выполняем команду make install, чтобы скопировать наше новое ядро ОС и необходимые файлы в соответствующие каталоги.
11. Ядро успешно скомпилировано и установлено. Далее необходимо сконфигурировать загрузчик (*см. разд. "Конфигурирование загрузчика"*).

Создание образа initrd

Файл initrd необходим для загрузки SCSI-модуля во время старта операционной системы. Скрипт /sbin/mkinitrd создает соответствующий образ initrd для вашего компьютера, если выполнены следующие условия:

- loopback block device доступно;
- файл /etc/modules.conf содержит описание вашего SCSI-контроллера.

Для построения образа initrd необходимо выполнить команду `/sbin/mkinitrd` с параметрами

```
/sbin/mkinitrd /boot/initrd-2.6.27-jul2008.img 2.6.28-jul2009
```

Здесь `/boot/initrd-2.6.28-jul2009.img` — имя файла для нового образа initrd, а `2.6.27-jul2008` — ядро, чьи модули (из /lib/modules) должны быть использованы при создании образа initrd.

Этапы компиляции модульного ядра

Подведем итог. Укажем последовательность действий при компиляции и инсталляции модульного ядра операционной системы Linux:

1. `make mrproper`.
2. `make menuconfig`.
3. `make dep`.
4. `make clean`.
5. Отредактировать /usr/src/kernels/2.6.27.XXXX/Makefile.
6. `make bzImage`.
7. `make modules`.
8. `make modules_install`.
9. `/sbin/mkinitrd /boot/initrd-2.6.xx.img 2.6.xx` (если в вашей системе установлен SCSI-контроллер).
10. `make install`.
11. Сконфигурировать загрузчик.

Монолитное ядро

Компиляция монолитного ядра ОС в основном повторяет компиляцию модульного ядра за некоторыми небольшими исключениями:

- когда конфигурируется ядро, не должны использоваться модули, т. е. на любой вопрос нужно отвечать только `Yes` или `No`. Также необходимо ответить `No` для пунктов kmod support и module version (CONFIG_MODVERSIONS) support;
- нужно пропустить следующие команды:
  - `make modules`;
  - `make modules_install`;
- для загрузчика LILO в файл lilo.conf следует добавить строчку `append=nomodules`.

Этапы компиляции монолитного ядра

Подведем итог. Укажем последовательность действий при компиляции и инсталляции монолитного ядра ОС Linux:

1. `make mrproper`.
2. `make menuconfig`.

3. `make dep`.
4. `make clean`.
5. Отредактировать /usr/src/kernels/2.6.27.xxxx/Makefile.
6. `make bzImage`.
7. `/sbin/mkinitrd /boot/initrd-2.6.xx.img 2.6.xx` (если в вашей системе установлен SCSI-контроллер).
8. `make install`.
9. Сконфигурировать загрузчик (*см. разд. "Конфигурирование загрузчика"*).

Параметры настройки ядра

Этот раздел полностью посвящен параметрам настройки ядра операционной системы Linux. Сначала на основе утилиты menuconfig покажем дерево параметров настройки ядра Linux, а затем кратко поясним основные параметры.

Параметры настройки ядра (комментарии)

Как вы уже заметили, параметров настройки ядра много и, чтобы правильно их сконфигурировать, необходимо иметь обширные знания о функциях операционной системы Linux. Еще одна особенность — почти 90% всех настроек и свойств ядра вынесены в модули. Если вы не уверены в том или ином свойстве — поставьте "использовать модуль".

Далее кратко прокомментируем основные пункты меню конфигурации ядра ОС.

- General setup — определяет основные свойства ядра: что оно сможет делать и какие типы устройств будет поддерживать. Здесь выбирается поддержка сети, шин PCI, EISA, MCA, PCMCIA-устройств, различных форматов исполняемых файлов и т. п.
- Enable loadable module support — отвечает за вид скомпилированного ядра операционной системы (модульное или монолитное).
- Infrastructure for tracing and debugging user processes — используется в основном программистами для углубленной отладки и трассировки программ.
- Enable block layer — раздел, отвечающий за большие блоковые устройства.
- Processor type and features — определяет тип процессора, для которого компилируется ядро операционной системы, поддержку набора процессорных команд, поддержку мультипроцессорной системы, объем поддерживаемой оперативной и виртуальной памяти и некоторые другие параметры. Для максимальной производительности системы рекомендуется выбрать именно тот тип процессора, который установлен в вашей системе, однако с целью совместимости в дистрибутиве ядро компилируется так, чтобы оно работало на любом процессоре — от Pentium до I7, AMD и Cyrix.
- Power management options — подсистема управления питанием. Важна для владельцев ноутбуков.
- Bus options (PCI etc.) — определяет настройки для различных шин периферийных устройств и систем ввода-вывода.

- Executable file formats / emulations — поддержка различных форматов исполняемых файлов и эмуляторов.
- Networking support — поскольку одним из важнейших назначений операционной среды Linux является работа с сетью, то, наверное, четверть параметров ядра тем или иным образом касается сети. Здесь определяются различные сетевые параметры: используемые сетевые протоколы, сетевая маршрутизация, конфигурирование виртуального сервера и многое другое.
- Device drivers — отвечает за поддержку ядром драйверов устройств.
- Firmware drivers — поддержка ядром различных BIOS, интеллектуальных контроллеров и т. п.
- File systems — поддержка различных файловых систем (VFAT, Ext3, ISO 9660 и т. п.), сетевых файловых систем, квотирования дискового пространства для пользователей, типов разделов жесткого диска. Сюда же вынесена поддержка различных языковых кодировок.
- Kernel hacking — всякие "хаки" ядра, в частности возможность отладки файловых систем и ядра системы.
- Security options — отвечает за систему безопасности ОС.
- Cryptographic API — поддержка систем шифрования.
- Virtualization — поддержка виртуализации в ядре. В основном необходима в различных серверах.
- Library routines — поддержка библиотек проверки и создания контрольных сумм.

Ссылки

- The Linux Kernel on Red Hat Linux Systems — обновление ядра операционной системы.
- **www.gnu.org/software/grub/** — домашняя страница GRUB.
- **www.redhat.com/support/docs/howto/kernel-upgrade/kernel-upgrade.html** — описание, как правильно обновлять операционную систему.
- /usr/src/linux-2.6/Documentation — большой объем документации, посвященный ядру Linux и ее модулям.
- Kernel-HOWTO (The Linux Kernel HOWTO) — описание конфигурирования и компиляции ядра.

Глава 17



DNS

DNS — это доменная система имен (Domain Name System). DNS преобразует символическое имя хоста в IP-адрес и наоборот — из IP-адреса в символическое имя. Для чего это нужно? Человеку легче запомнить осмысленное имя — типа www.lazzycat.com, чем 213.162.145.242, а для компьютера проще передать 4 байта адреса, чем 50–60 байтов имени.

На заре эры глобальных сетей все пары "имя–IP-адрес" хранились в файле /etc/hosts. Со временем, когда число компьютеров в сети достигло тысяч и десятков тысяч, этот механизм стал крайне неэффективен, и на смену пришли DNS-серверы.

DNS-сервер представляет собой базу данных, в которой хранится соответствие символического имени хоста IP-адресу. В сети существуют десятки тысяч серверов DNS, которые обмениваются информацией с другими серверами DNS.

DNS — это иерархическая система. Вершина записывается как . (точка) и произносится как root (корень). В корне существует несколько доменов верхнего уровня (Top Level Domains, TLDs). Некоторые наиболее известные из них: ORG, COM, EDU, GOV, MIL, NET, RU, UA и т. п.

При поиске машины запрос обрабатывается рекурсивно, начиная с корня. Если нужно найти адрес машины moshkin.bins.ru, то ваш сервер имен сперва проверяет свою базу. Если там не оказалось нужной нам записи, ваш сервер должен найти сервер имен, который обслуживает ru. Он запрашивает корневой сервер (.), который выдает список серверов ru. Из полученного списка выясняется, какие серверы имен обслуживают зону ru. Затем запрашивается сервер (выбирается по определенному алгоритму или берется первый в полученном списке), чтобы узнать, какие серверы обслуживают зону bins.ru. Далее берется сервер из полученного списка и узнается IP-адрес хоста moshkin.bins.ru. А чтобы в следующий раз не повторять этот поиск, полученную пару "имя–IP-адрес" ваш сервер DNS сохраняет в своей базе данных.

При обратной задаче — по IP-адресу узнать имя хоста — опять используется DNS-сервер. Для этих целей существует псевдодомен in-addr.arpa и в нем точно так же прописываются адреса, только порядок следования цифр обратный. Например, для адреса 213.162.145.242 запрос получится как 242.145.162.213.in-addr.arpa, а схема поиска ответа остается такая же.

По своим функциональным обязанностям различают два вида DNS-серверов:

- *кэширующий сервер DNS* — хранить запрошенные пользователем пары "имя–IP-адрес" локально, что при интенсивном общении со многими Web-серверами позволяет экономить время на DNS-запросах. Кэширующий сервер не отвечает на внешние DNS-запросы;
- *обычный сервер DNS* — это полнофункциональный сервер, позволяющий получать, передавать и синхронизировать DNS-данные с другими DNS-серверами.

Настройка сетевых параметров

Поскольку DNS-сервер очень сильно завязан на всю сетевую инфраструктуру, его работоспособность зависит от правильной конфигурации сети. В современных дистрибутивах, если вы выбрали опцию "устанавливать DNS-сервер", то он конфигурируется автоматически. Однако разработчик дистрибутива рассчитывает на абстрактную среднестатистическую систему, которой, как показывает практика, не существует. Прежде всего, следует убедиться, что с сетевыми настройками у вас все в порядке, поэтому рассмотрим основные файлы, отвечающие за конфигурацию сети.

Файл host.conf

Этот файл предназначен для того, чтобы система могла определить, каким образом она будет получать информацию об именах и IP-адресах.

Следующая запись в файле host.conf означает, что при поиске хостов система сначала посмотрит в /etc/hosts, а потом только обратится к серверу DNS:

```
order hosts,bind
```

Запись должна быть именно такая, поскольку может оказаться, что по какой-либо причине нет доступа к серверу DNS (простейший случай — он просто еще не запущен), а уже есть необходимость воспользоваться сетью.

Файл /etc/hosts

В этом файле находятся пары "IP-адрес–имя":

```
127.0.0.1    localhost localhost.localdomain
192.168.0.1  user
192.168.0.2  user2
```

Причем обязательно должна присутствовать следующая строка:

```
127.0.0.1    localhost localhost.localdomain
```

Этот файл позволяет выполнять преобразование "Имя хоста–IP-адрес" без обращений к DNS-серверу. Обычно используется для небольшой локальной сети, когда нет необходимости в установке и настройке DNS-сервера.

Файл /etc/resolv.conf

В этом файле должны находиться строки, подобные приведенным:

```
   search bins.ru
nameserver 213.166.195.22
```

В строке, которая начинается со слова `search`, указывают, какое доменное имя будет принято по умолчанию. Так, если вы напишете `user`, то система сразу попытается обратиться к компьютеру user.bins.ru. После `search` можно указывать несколько имен. В следующей строчке указывают адреса DNS-серверов, к которым будет обращаться ваша машина.

Настройка кэширующего сервера

Кэширующий сервер предназначен для нахождения пары "Имя хоста–IP-адрес" в своей базе или на других DNS-серверах и сохранения пары у себя в базе. При этом он сконфигурирован таким образом, что только принимает данные извне сети, но не отдает информацию наружу. Такого типа DNS-серверы позволяют уменьшить время ожидания ответа от DNS-сервера и рекомендуются при медленном соединении или в том случае, когда внешний DNS-сервер перегружен.

Рассмотрим конфигурационные файлы такого сервера.

Файл /etc/named.conf

Это основной конфигурационный файл DNS-сервера. Листинг 17.1 иллюстрирует содержимое данного файла для кэширующего сервера.

Листинг 17.1

```
options {
        directory "/var/named";
};

zone "." {
        type hint;
        file "root.hints";
};

zone "0.0.127.in-addr.arpa" {
        type master;
        file "127.0.0";
};
```

Строка `directory` указывает `bind`, в каком каталоге искать файлы. Все файлы, используемые впоследствии, будут иметь путь относительно этого каталога.

Строка `zone "0.0.127.in-addr.arpa"` показывает, что `bind` также отвечает за обратную зону для подсети 127.*.*.*, является в ней мастером, и файл описания зоны — 127.0.0.

Секция `zone "."` самая важная. Она описывает, в каком файле лежат адреса корневых DNS-серверов, которые отвечают за зоны первого уровня.

Файл, названный /var/named/root.hints, должен находиться в указанном каталоге и содержать информацию, пример которой приведен в листинге 17.2.

Листинг 17.2

```
.                       6D IN NS        G.ROOT-SERVERS.NET.
.                       6D IN NS        J.ROOT-SERVERS.NET.
.                       6D IN NS        K.ROOT-SERVERS.NET.
.                       6D IN NS        L.ROOT-SERVERS.NET.
.                       6D IN NS        M.ROOT-SERVERS.NET.
.                       6D IN NS        A.ROOT-SERVERS.NET.
.                       6D IN NS        H.ROOT-SERVERS.NET.
.                       6D IN NS        B.ROOT-SERVERS.NET.
.                       6D IN NS        C.ROOT-SERVERS.NET.
.                       6D IN NS        D.ROOT-SERVERS.NET.
.                       6D IN NS        E.ROOT-SERVERS.NET.
.                       6D IN NS        I.ROOT-SERVERS.NET.
.                       6D IN NS        F.ROOT-SERVERS.NET.

G.ROOT-SERVERS.NET.     5w6d16h IN A    192.112.36.4
J.ROOT-SERVERS.NET.     5w6d16h IN A    198.41.0.10
K.ROOT-SERVERS.NET.     5w6d16h IN A    193.0.14.129
L.ROOT-SERVERS.NET.     5w6d16h IN A    198.32.64.12
M.ROOT-SERVERS.NET.     5w6d16h IN A    202.12.27.33
A.ROOT-SERVERS.NET.     5w6d16h IN A    198.41.0.4
H.ROOT-SERVERS.NET.     5w6d16h IN A    128.63.2.53
B.ROOT-SERVERS.NET.     5w6d16h IN A    128.9.0.107
C.ROOT-SERVERS.NET.     5w6d16h IN A    192.33.4.12
D.ROOT-SERVERS.NET.     5w6d16h IN A    128.8.10.90
E.ROOT-SERVERS.NET.     5w6d16h IN A    192.203.230.10
I.ROOT-SERVERS.NET.     5w6d16h IN A    192.36.148.17
F.ROOT-SERVERS.NET.     5w6d16h IN A    192.5.5.241
```

Этот файл описывает имена корневых серверов имен по всему миру. Их список периодически изменяется. Поэтому данный файл необходимо время от времени корректировать.

Для получения файла root.hints существует, по меньшей мере, два пути: либо забрать его по FTP с сервера internic, либо выполнить команду

```
dig @rs.internic.net . ns >root.hints
```

Файл /etc/127.0.0

127.0.0 — это файл, который отвечает за преобразование IP-адресов в символические имена.

В листинге 17.3 показано, как должен выглядеть файл 127.0.0.

Листинг 17.3

```
@            IN     SOA   ns.bins.ru. hostmaster.bins.ru. (
                                1       ; Serial
                                8H      ; Refresh
                                2H      ; Retry
                                1W      ; Expire
                                1D)     ; Minimum TTL
IN                      NS      ns.bins.ru.
1                       PTR     localhost.
```

Эта запись обозначает следующее:

- `@` указывает, что описываем сами себя;
- описываемая зона поддерживается сервером с именем `ns.bins.ru`;
- отвечает за нее администратор, доступный по адресу `hostmaster@bins.ru` (первая точка заменяет `@`);
- у зоны серийный номер равен `1` (обычно для него указана дата последней правки зоны — на него опираются другие серверы, которые получают информацию с вашего сервера);
- другие серверы будут обновлять информацию о вашем сервере с периодичностью в `8` часов;
- при неудачном обновлении следующая попытка будет произведена через `2` часа;
- зона будет считаться содержащей недостоверную информацию на кэширующих серверах через `1` неделю;
- но не менее чем через `1` день;
- строка `IN NS ns.bins.ru.` показывает, что авторитетным сервером для этой зоны является `ns.bins.ru.`, и именно ему нужно рассылать обновления зоны `ns.bins.ru`;
- строка `1 PTR localhost.` показывает, что хост с адресом `1` в зоне `127.0.0.` имеет имя `localhost`.

Запуск named

После правки конфигурационных файлов можно запускать сервер. Наберите `nds start` без опций и нажмите клавишу <Enter>.

Затем запускаем программу nslookup:

```
$ nslookup
Default Server:  localhost
Address:  127.0.0.1
```

Если вы увидели `>_` на мониторе — система работает. Каждый раз, когда вы изменяете файл named.conf, необходимо перезапустить named, вводя команду `nds restart`.

Теперь проверим, как функционирует ваш кэширующий сервер — введем `user.bins.ru` (листинг 17.4).

Листинг 17.4

```
> user7.bins.ru
Server:  localhost
Address:  127.0.0.1

Name:    user7.bins.ru
Address:  213.166.195.55
```

При этом программа nslookup попросила ваш сервер DNS посмотреть информацию о данном хосте. Если вы повторно запросите адрес компьютера `user7.bins.ru`, то получите ответ, приведенный в листинге 17.5.

Листинг 17.5

```
> user7.bins.ru
Server:  localhost
Address:  127.0.0.1

Non-authoritative answer:
Name:    user7.bins.ru
Address:  213.166.195.55
```

Теперь вы получили сообщение "Non-authoritative answer". Это значит, что DNS во второй раз не запрашивал внешние серверы имен, а провел поиск в своем кэше и нашел нужную запись. Поскольку вы увидели это сообщение, ваш кэширующий DNS-сервер нормально функционирует. Получив положительный результат, можно завершить работу программы nslookup, дав команду `exit`.

Настройка полнофункционального DNS-сервера

Настройка полнофункционального DNS-сервера несколько сложнее, чем кэширующего, но, в основном, файлы и записи те же самые. Для чистоты эксперимента рекомендуется выполнить настройку для несуществующего домена. У нас он будет называться `ivan.petrov`.

Файл /etc/named.conf

В листинге 17.6 приведено содержимое данного файла для нашего сервера DNS.

Листинг 17.6

```
options {
        directory "/var/named";
};

zone "." {
        type hint;
        file "root.hints";
};

zone "0.0.127.in-addr.arpa" {
        type master;
        file "127.0.0";
};
zone "ivan.petrov" {
        notify no;
        type master;
        file "ivan.petrov";
};
zone "0.168.192.in-addr.arpa" {
        notify no;
        type master;
        file "192.168.0";
};
```

Как можно видеть, по сравнению с кэширующим сервером добавились только секции `zone "ivan.petrov"` и `zone "0.168.192.in-addr.arpa"`.

Секция `zone "ivan.petrov"` определяет, что наш DNS-сервер предназначен для зоны `ivan.petrov` и является в ней мастером (т. е. другие серверы лишь синхронизируют по нему свои записи по зоне `ivan.petrov`), при изменении записей в зоне не извещает другие серверы и использует для описания зоны файл ivan.petrov.

Секция zone `"0.168.192.in-addr.arpa"` описывает, что наш DNS-сервер поддерживает реверсную зону `0.168.192.in-addr.arpa`, является в ней мастером, при изменении записей в зоне не извещает другие серверы и использует для описания зоны файл 192.168.0.

Файл /etc/named/ivan.petrov

В файле зоны `ivan.petrov` поместим данные из листинга 17.7.

Листинг 17.7

```
@   IN   SOA   ns.ivan.petrov. hostmaster.ivan.petrov. (
               199802151       ; serial, todays date + todays serial #
               8H              ; refresh, seconds
               2H              ; retry, seconds
               1W              ; expire, seconds
               1D )            ; minimum, seconds
;
    NS      ns                      ; Интернет-адрес сервера имен
    MX      10 mail.ivan.petrov.    ; Основной почтовый сервер
    MX      20 mail2.ivan.petrov.   ; Дополнительный почтовый сервер
;
localhost       A       127.0.0.1
ns              A       192.168.0.1
mail            A       192.168.0.40
```

Этот файл зоны содержит четыре записи ресурсов (Resource Records, RR):

- `SOA RR` — запись SOA (Start Of Authority, начало полномочий) находится в преамбуле каждого из файлов зон, и она должна быть первой записью в файле. Описывает зону, откуда ее берут (машина, названная `ns.ivan.petrov`), кто отвечает за содержимое зоны (`hostmaster@ivan.petrov`), какая версия файла зоны текущая (`serial: 1`) и другие сведения, необходимые для кэширующих и вторичных серверов DNS;
- `NS RR` — это RR для сервера имен (Name Server, NS);
- `MX RR` — запись MX (Mail eXchanger, почтовый сервер) сообщает почтовой системе, куда посылать почту, адресованную любому адресату в домене `ivan.petrov`, в нашем случае — серверам `mail.ivan.petrov` или `mail2.ivan.petrov`. Число перед каждым именем системы — это приоритет записи MX RR. Запись ресурса с наименьшим номером (10) — хост, куда почта должна отправляться в первую очередь. Если происходит ошибка, то почта может быть послана на машину с бóльшим номером. И так далее. Таким образом, можно указать несколько почтовых серверов, что поможет вам при форс-мажорных обстоятельствах не потерять ваши почтовые сообщения;
- `A RR` — A (Address, адрес) — IP-адрес:

```
localhost       A       127.0.0.1
ns              A       192.168.0.1
mail            A       192.168.0.40
```

Эти строки описывают соответствие имен `mail` и `ns` в зоне `ivan.petrov` их IP-адресам.

Файл /etc/192.168.0

Для нормального функционирования DNS-сервера требуется обратная (реверсная) зона, которая дает возможность DNS преобразовывать IP-адреса в имена хостов. Поскольку эти имена используются серверами различного рода (FTP, IRC, WWW и т. п.), то обратная зона требуется для полного доступа к различным сервисам в Интернете.

Листинг 17.8 иллюстрирует содержимое файла /etc/192.168.0.

Листинг 17.8

```
@       IN      SOA     ns.ivan.petrov. hostmaster. ivan.petrov. (
                        199802151 ; Serial, todays date + todays serial
                        8H      ; Refresh
                        2H      ; Retry
                        1W      ; Expire
                        1D)     ; Minimum TTL
                NS      ns.linux.bogus.

2               PTR     gw.ivan.petrov.
1               PTR     ns.ivan.petrov.
3               PTR     petya.ivan.petrov.
40              PTR     mail.ivan.petrov.
5               PTR     ftp.ivan.petrov.
```

Приведенный файл в принципе мало чем отличается от файла описания прямой зоны строки, выделенные полужирным шрифтом, описывают то, что машина с адресом `2` в зоне `192.68.0.` имеет имя `gw.ivan.petrov`, а компьютер с адресом `40` — `mail.ivan.petrov`.

Вот, собственно, и все. Перезапускаем DNS-сервер и проверяем правильность функционирования нашей системы.

Некоторые тонкости

Как вы видите, глубоко в тонкости функционирования DNS-сервера мы не погружались. Во-первых, этого вполне достаточно для настройки небольшого DNS-сервера, а во-вторых, решать проблемы следует по мере их возникновения и для этих целей незаменима документация, поставляющаяся вместе с DNS-сервером. Тем не менее некоторые нюансы необходимо знать.

Записи ресурсов (RR) службы DNS

Давайте рассмотрим несколько расширенный файл описания зоны (листинг 17.9).

Листинг 17.9

```
gw              A       192.168.0.2
                HINFO   "i586" "RH 6.2"
                TXT     "The router"

ns              A       192.168.0.1
                MX      10 mail
                HINFO   "Pentium4" "Fedora 9"
www             CNAME   ns
User            A       192.168.0.3
                MX      10 mail
                HINFO   "p4"  "WindowsXP"
                TXT     "Developer computer home tel 223344"
```

Помимо знакомых вам строчек появились строки, содержащие `HINFO`, `CNAME` и `TXT`.

- `HINFO` — информация о компьютере (Host INFOrmation); состоит из двух частей: первая часть — это информация об оборудовании, а вторая — о программном обеспечении и операционной системе данной машины. Помимо этой информации не рекомендуется вносить ничего другого. Пример:

  ```
  HINFO   "Pentium4" "Fedora 9"
  ```

  Из данной строки видно, что наш DNS-сервер собран на базе процессора Pentium IV и на нем установлена ОС Linux Fedora 9;
- `CNAME` — каноническое имя (Canonical NAME) — это способ присвоить каждой машине несколько имен. При использовании `CNAME` необходимо следовать правилу, что записи `MX`, `CNAME` или `SOA` *никогда* не должны ссылаться на имя, указанное как запись `CNAME`;
- `TXT` — произвольная текстовая информация. Обычно расширенный комментарий для описания хоста. Пример:

  ```
  TXT     "Developer computer home tel 223344"
  ```

  Из содержимого строчки понятно, что это компьютер разработчика, а его домашний телефон — 223344.

Существует еще один тип записи — RP (Responsible Party, группа ответственных). В принципе та же информация может храниться и в записях `TXT`, однако применение записи RP ускоряет поиск данных об ответственных лицах. Список основных записей ресурсов службы DNS приведен в табл. 17.1.

Таблица 17.1. *Основные записи ресурсов (RR) службы DNS*

Обозначение записи	Содержание записи	Номер RFC или автор проекта
A	IP-адрес хоста	RFC1035
AAAA	Адрес IPv6	Проект, автор Thomson
CNAME	Каноническое имя домена	RFC1035
GPOS	Географическое положение	RFC1712
HINFO	Информация о хосте (процессор и ОС)	RFC1035
ISDN	Адрес ISDN	RFC1183
KEY	Ключ шифрования	Проект, автор Eastlake
LOC	Расположение	Проект, автор Vixie
MX	Имя хоста или домена для переадресации почты	RFC1035
NSAP	SAP-адрес (адрес A в формате NSAP)	RFC1706
NSAP-PTR	Аналог записи PTR для адреса NSAP	RFC1706
NULL	Пустая запись ресурса	RFC1035
NXT	Следующий домен	Проект, автор Eastlake
PTR	Указатель на имя домена	RFC1035
RP	Ответственные лица	RFC1183
SIG	Цифровая подпись	Проект, автор Eastlake
SRV	Выбор сервера	Проект, автор Vixie
TXT	Произвольный текст	RFC1035
WKS	Описание подключенных сервисов	RFC1035
X25	Адрес X.25	RFC1183

Реверсная зона

Не забывайте об обратной (реверсной) зоне! Очень неприятно, когда по этой причине вы не сможете воспользоваться FTP-сервером или получите сообщения о нарушениях системы защиты. Не поленитесь — потратьте час на описание реверсной зоны.

Два сервера DNS

Существует множество причин, по которым нежелательно раскрывать всю информацию о вашей сети через службу DNS. Поэтому рекомендуется создать два сервера DNS: один для внутренних пользователей, другой — для внешних. Для этого необходимо обеспечить два различных набора IP-адресов: для внутренних клиентов и для внешнего мира.

Иерархические поддомены

Если в вашей организации имеется более одной подсети, то вам придется задать несколько доменов `in-addr.arpa`. Создание поддоменов, подчиненных первичному домену, целесообразно также при наличии в вашей организации нескольких отделов или подразделений. Это облегчит мониторинг сети, а также упростит организацию доступа в сеть и установку защитных фильтров. Конечно, если ваша сеть состоит всего из нескольких машин, смысла в создании иерархии доменов просто нет.

Вторичные DNS-серверы

Если у вас большая сеть или если вы занимаетесь хостингом сайтов, то обязательно должны помимо первичного сервера DNS иметь еще и вторичный сервер DNS. Это позволит уменьшить время отклика на запрос, а также повысить отказоустойчивость сети.

Используйте серверы кэширования

Если вы занимаетесь обслуживанием сети, рекомендуется установить кэширующие DNS-серверы, пусть не на каждый компьютер, но в каждой подсети. Быстродействие, которое обеспечивает такой подход, становится заметным уже в сетях средней сложности.

Инструменты

Для тех, кто не хочет подробно изучать настройку DNS с помощью конфигурационных файлов, существуют доступные инструменты, позволяющие вносить изменения, особо не задумываясь. Поищите, и вы наверняка найдете десяток-другой программ для удаленного администрирования DNS-сервера, в том числе и имеющих графическую "дружественную" оболочку. В частности исходный код на языке HTML для создания инструментария по управлению службой DNS можно найти в Интернете по адресу **webdns.lcs.mit.edu/cgi-bin/webdns**/. Существует также универсальная программа для администрирования множеством сервисов через Интернет — `webmin`.

Ссылки

- DNS-HOWTO.
- **linux.webclub.ru/bind/pers_dns.html** — Водолазкий В. Мой личный сервер DNS.
- **www.biblioteka.agava.ru/nastroyka_dns.htm** — Калошин В. Настройка DNS.
- **www.4com.ru/support/DNSAdvanSetup.html** — Холл Э. Тонкая настройка DNS.
- **www.webmin.com/webmin/** — сайт программы `webmin`.

Глава 18



DHCP

Как вы знаете, без IP-адреса компьютер нельзя включить в сеть с протоколом TCP/IP. В малой сети назначить IP-адреса каждому компьютеру и прописать их в соответствующих местах нетрудно. Все намного хуже, если под вашим "крылом" сеть из 40–50 компьютеров или даже больше. А если в вашей сети у пользователей "очумелые ручки", и установлены компьютеры с операционной системой Windows — наверняка не пройдет и недели, как вы получите вызов к "пациенту", у которого пользователь удалил сетевые настройки, неправильно выставил DNS или начудил с IP-адресом компьютера.

Похожие проблемы возникают при подключении нового или "гостевого" компьютера, особенно если IP-адреса ранее выдавались бессистемно.

Для решения проблем с корректным автоматизированным назначением IP-адресов и предназначен протокол DHCP.

DHCP-протокол

DHCP — протокол динамического конфигурирования хостов (Dynamic Host Configuration Protocol) — это клиент-серверный протокол, предназначенный для управления сетевыми параметрами хостов. Описание протокола содержится в RFC 2131, RFC 2132, которые сменили устаревшие RFC 1531 и RFC 1541.

Архитектура и формат сообщений

DHCP — классический клиент-серверный протокол. Клиентами выступают компьютеры сети, пытающиеся получить IP-адрес, адрес сетевого шлюза, имя хоста и другие параметры, о которых вы узнаете чуть позже. Сервер DHCP выдает в ответ на запрос клиентов назначаемые им сетевые параметры (IP-адрес, адрес шлюза), контролирует использование IP-адресов, поддерживает пул свободных адресов и ведет собственную базу клиентов.

В роли транспортного протокола для обмена DHCP-сообщениями выступает протокол UDP. При отправке сообщения с клиента на сервер используется 67-й порт DHCP-сервера, при передаче в обратном направлении — 68-й. Эти номера портов, как и схожая структура сообщений, обеспечивают обратную совместимость протоколов DHCP с BOOTP.

Структура DHCP-пакета приведена в табл. 18.1.

Таблица 18.1. *Структура DHCP-пакета*

Название поля	Величина в байтах	Описание
op	1	Тип сообщения (1 = BOOTREQUEST (запрос), 2 = BOOTREPLY (ответ))
htype	1	Тип адреса оборудования
hlen	1	Длина адреса оборудования
hops	1	Используется ретранслирующим агентом
xid	4	Идентификатор транзакции между клиентом и сервером
Secs	2	Время с момента выдачи запроса или начала обновления конфигурации
Flags	2	Флаги
Ciaddr	4	IP-адрес клиента
Yiaddr	4	IP-адрес, предлагаемый сервером хосту в качестве клиентского
Siaddr	4	IP-адрес следующего сервера, участвующего в загрузке
Giaddr	4	IP-адрес ретранслирующего агента
Chaddr	16	MAC-адрес клиента
Sname	64	Хост-имя сервера (опционально)
File	128	Имя загрузочного файла (опционально)
options	312–576	Используется для передачи параметров конфигурации

Режимы выдачи IP-адресов

Казалось бы, раз протокол предназначен для динамической выдачи адресов, то и режим один — динамический. Но нет! Чтобы не лишать администратора гибкости при назначении IP-адресов, предусмотрено три режима: статический, динамический и ручной. Рассмотрим их отличия:

- *статический* — DHCP-сервер конфигурируется так, что хостам назначаются неизменные со временем IP-адреса;
- *динамический* — хосты получают IP-адреса, которые могут меняться с течением времени;
- *ручной* — DHCP-сервер уведомляет клиента об адресе, присвоенном ему администратором сети вручную.

Как видите, первый и последний варианты достаточно тривиальны, и особо на них останавливаться не будем. Нас интересует второй случай — динамическое распределение адресов.

IP-адрес выдается в аренду по инициативе (запросу) клиента. DHCP-сервер гарантирует, что до истечения срока аренды этот IP-адрес не будет выдан в аренду

другому клиенту. Сервер обычно настраивают так, что при повторных обращениях клиента в течение определенного срока (зависит от администратора, обычно неделя-две) он старается выдать клиенту IP-адрес, использовавшийся им ранее. Клиент может (при соответствующей настройке) запросить продление сроков аренды IP-адреса либо досрочно от него отказаться.

Давайте рассмотрим процесс получения IP-адреса клиентом.

1. Клиент посылает широковещательный запрос, в котором может указываться устраивающий клиента IP-адрес и срок его аренды. Если в физическом сегменте сети клиента DHCP-сервер отсутствует, сообщение будет передано в другие сегменты сети ретранслирующими агентами протокола BOOTP.
2. DHCP-сервер посылает в ответ пакет, содержащий доступный IP-адрес (поле `yiaddr`) и, возможно, параметры конфигурации клиента. На этой стадии сервер не обязан резервировать указанный в поле `yiaddr` адрес, однако должен проверить посредством ICMP то, что этот IP-адрес свободен.
3. Клиент не обязан реагировать на первое поступившее предложение. Возможен вариант, что клиент получил отклики от нескольких DHCP-серверов, выбрал понравившийся ему адрес и отправил в сеть широковещательное сообщение, в котором содержатся идентификатор выбранного DHCP-сервера и желательные значения запрашиваемых параметров конфигурации. Именно поэтому сервер не резервирует сразу IP-адрес, переданный в первом ответе сервера.
4. Сервер посылает пакет-подтверждение, в котором содержатся значения параметров конфигурации, и резервирует IP-адрес за клиентом. Если к моменту поступления ответа от клиента предложенный ранее адрес уже закреплен за другим клиентом, сервер сообщает клиенту о невозможности получения именно этого IP-адреса.
5. Клиент, получив сетевые параметры от DHCP-сервера, должен средствами протокола ARP убедиться в уникальности IP-адреса и зафиксировать суммарный срок его аренды. Если IP-адрес уже занят другим хостом, то клиент обязан отправить серверу уведомляющее сообщение и начать всю процедуру снова не ранее чем через 10 с.

Параметры конфигурации (поле *options*)

В качестве параметров конфигурации DHCP-сервер может выдать клиенту помимо IP-адреса и имени хоста довольно большой объем информации. Перечислим основные данные, которые может получить клиент от DHCP-сервера.

- Маска подсети.
- MTU (максимальный размер передаваемого пакета).
- TTL (время жизни пакета).
- Адреса COOKIE-серверов.
- Адреса DNS-серверов.
- Адреса FINGER-серверов.
- Адреса IRC-серверов.
- Адреса LOG-серверов.
- Адреса LPR-серверов.

- Адреса WINS-серверов.
- Адреса NIS-серверов.
- Адреса NNTP-серверов.
- Адреса NTP-серверов.
- Адреса POP-серверов.
- Адреса SMTP-серверов.
- Адреса TFTP-серверов.
- Адреса WWW-серверов.

Предусмотрено еще много второстепенных параметров, полное описание которых можно найти в документации на DHCP-сервер.

Недостатки DHCP

К недостаткам протокола, прежде всего, следует отнести крайне низкий уровень информационной безопасности, что обусловлено непосредственным использованием протокола UDP. Также нет защиты от появления в сети несанкционированных DHCP-серверов, способных рассылать клиентам ошибочную или потенциально опасную информацию: некорректные или уже задействованные IP-адреса, неверные сведения о маршрутизации и т. д.

Помимо этого, существует проблема согласования информационной адресной базы в службах DHCP и DNS.

И, наконец, централизация процедуры назначения адресов снижает надежность системы при отказе DHCP-сервера, поскольку все его клиенты оказываются не в состоянии получить IP-адрес и другую информацию о сетевой конфигурации.

DHCP-сервер

Начнем описание программного обеспечения с серверной части, как наиболее трудоемкой в настройке и более ответственной. Программное обеспечение DHCP-сервера входит в практически любой современный дистрибутив. При отсутствии на дисках дистрибутива это ПО можно загрузить с сайта разработчика DHCP — Internet Software Consortium — **http://www.isc.org**.

Установка пакета не вызывает никаких сложностей. После нее нужно убедиться, что демон dhcpd будет автоматически стартовать при загрузке операционной системы.

За конфигурацию dhcpd отвечают два файла:

- /etc/dhcpd.conf;
- /var/lib/dhcp/dhcpd.leases.

Файл dhcpd.conf

В этом файле содержатся все настройки DHCP-сервера. Сначала опишем настройки, а после рассмотрим парочку типичных конфигурационных файлов.

Текстовый файл ASCII dhcpd.conf содержит конфигурационную информацию для демона dhcpd. Комментарий — строка, начинающаяся с символа #. Конфигура-

ционные переменные состоят из двух частей: параметров и значений, заканчивающихся точкой с запятой.

Глобальные параметры, действие которых распространяется на все группы, размещаются в начале данных, до описания групп.

> ***Замечание***
>
> Большинство параметров можно безболезненно размещать как в глобальной области, так и в группах.

В начале файла можно определить параметры, действие которых будет распространяться на все группы данных:

- `ddns-update-style none;` — разрешает либо запрещает динамическое обновление DNS;
- `option domain-name "test.org";` — задает имя домена, в котором функционирует DHSP-сервер. В дальнейшем можно не указывать в переменной `host` полное имя хоста;
- `option domain-name-servers` *имена DNS-серверов*`;` — определяет список DNS-серверов, используемых сервером DHCP при разрешении символических имен;
- `option netbios-name-servers` *список IP-адресов*`;` — если клиент применяет протокол NetBIOS, определяет список WINS-серверов;
- `option netbios-node-type` *цифровое значение*`;` — определяет порядок использования параметра `netbios-name-servers`:
  - `1` — широковещательные запросы вместо WINS-сервера;
  - `2` — только WINS-сервер;
  - `3` — сначала широковещательные запросы, затем WINS-сервер;
  - `4` — сначала WINS-сервер, затем широковещательные запросы;
- `option nis-domain "test.org";` — если присутствует поддержка NIS, можно задать домен подсети;
- `max-lease-time` *секунды*`;` — максимальное время аренды IP-адреса клиентом (в секундах). Если за это время клиент не запросил о подтверждении аренды адреса, клиент считается отсутствующим в сети, а его IP-адрес свободным для аренды;
- `default-lease-time` *секунды*`;` — время аренды IP-адреса клиентом, заданное по умолчанию (в секундах). Если за это время клиент не запросил о подтверждении аренды адреса, клиент считается отсутствующим в сети. Обычно данные переменные устанавливаются одинаково;
- `min-lease-time` *секунды*`;` — минимальное время аренды IP-адреса клиентом (в секундах). Если за это время клиент не запросил о подтверждении аренды адреса, клиент считается отсутствующим в сети. Обычно данные переменные устанавливаются одинаково.

Следующие три параметра позволяют задать способ взаимодействия с клиентами, которые не определены в списке хостов DHCP-сервера (отсутствуют MAC-адреса):

- `allow unknown-clients;` — разрешает выдачу IP-адреса для неизвестного клиента;
- `deny unknown-clients;` — отклоняет выдачу IP-адреса для неизвестного клиента;
- `ignore unknown-clients;` — игнорирует запросы неизвестного клиента на получение IP-адреса.

Следующие параметры позволяют определить стратегию сервера по взаимодействию с клиентами, посылающими запрос по bootp-протоколу:

- `allow bootp;` — разрешать получение IP-адреса по протоколу bootp;
- `deny bootp;` — отклонять запросы по протоколу bootp;
- `ignore bootp;` — игнорировать запросы по протоколу bootp.

Логически связанные параметры могут находиться между фигурными скобками `{}`, перед ними ставят конфигурационную переменную, к которой применяются эти параметры (листинг 18.1).

В качестве таких переменных могут выступать:

- `subnet`;
- `group`;
- `host`.

Листинг 18.1

```
subnet 204.254.239.64 netmask 255.255.255.224 {
          параметры подсети ...
          range 204.254.239.74 204.254.239.94;
 }
group {
          параметры группы ...
          host vasya.test.org {
          параметры хоста ...
           }
          host petya.test.org {
              параметры хоста ...
           }
 }
```

Здесь:

- `subnet` *IP-адрес* `netmask` *маска сети* — этот параметр определяет адрес подсети и маску, для которой сервер DHCP будет выдавать динамические IP-адреса. В конфигурационном файле может быть несколько записей `subnet`:
  - `range` *IP-адрес начала диапазона IP-адрес конца диапазона* — задает диапазон IP-адресов, из которых сервер для данной подсети может выдавать адреса. Параметр `range` необязательный, при его отсутствии диапазон динамически выдаваемых IP-адресов определяется исходя из подсети и ее маски;
  - `option domain-name "test.org";` — задает имя домена, в котором функционирует DHSP-сервер. В дальнейшем можно не указывать в переменной `host` полное имя хоста;
  - `option nis-domain "test.org";` — если присутствует поддержка NIS, можно задать домен подсети;
  - `option routers` *IP-адрес;* — список IP-адресов маршрутизаторов;

- `option subnet-mask` *маска подсети*`;` — позволяет задать маску подсети;
- `option domain-name-servers` *имена DNS-серверов*`;` — список DNS-серверов, используемых сервером DHCP при разрешении символических имен;
- `range dynamic-bootp` *IP-адрес начала диапазона IP-адрес конца диапазона*`;` — для клиентов, которые производят загрузку по протоколу bootp, этот параметр задает диапазон адресов, откуда берутся IP-адреса;
- `option broadcast-address` *IP-адрес*`;` — адрес для посылки широковещательных сообщений;

- `group` — параметр, с помощью которого для некоторого набора хостов можно задавать некоторые общие параметры. Эти параметры уже упоминались в описании глобальных параметров и в описании подсети;
- `host` *имя хоста* — позволяет задать для хоста *имя хоста* и некоторые параметры, специфичные именно для него:
  - `hardware ethernet` *xx*`:`*xx*`:`*xx*`:`*xx*`:`*xx*`:`*xx*`;` — определяет тип сетевого интерфейса и его MAC-адрес, который записывается в виде *xx*`:`*xx*`:`*xx*`:`*xx*`:`*xx*`:`*xx* (для Ethernet-карты), где *xx* — восьмибитовое число в шестнадцатеричном представлении;
  - `fixed-address` *IP-адрес*`;` — позволяет зафиксировать IP-адрес за определенным хостом;
  - `filename "`*filename*`";` — задает имя файла, который загружает хост после получения IP-адреса. Используется для загрузки бездисковых клиентов.

Помимо этого существует еще десятка два конфигурационных параметров, однако они достаточно специфичны и встречаются редко. Для ознакомления с ними рекомендую воспользоваться ссылками, приведенными в конце главы.

Файл dhcpd.leases

Файл dhcpd.leases — это база данных, в которой хранятся записи о клиентах и арендованных ими IP-адресах. Запись представляет собой несколько строк (листинг 18.2).

Листинг 18.2

```
lease 192.168.10.27 {
       starts 5 200806/20 09:14:54;
       ends 5 2008/06/27 09:14:54;
       hardware ethernet 00:60:67:75:40:37;
       uid 01:00:60:67:75:40:37;
       client-hostname "Oscar";
}
```

Здесь:

- `lease 192.168.10.27` — показывает, какой IP-адрес взят в аренду;
- `starts 5 2008/06/20 09:14:54;` — начало срока аренды (в данном случае 20 июня 2008 года в 9 часов 14 минут и 54 секунды);

- `ends 5 2008/06/27 09:14:54;` — предполагаемый конец аренды (если клиент не запросит продления аренды). Легко заметить — время аренды равно 7 суткам;
- `hardware ethernet 00:60:67:75:40:37;` — показывает, что сетевой интерфейс, которому назначен IP-адрес, это Ethernet-карта с MAC-адресом 00:60:67:75:40:37;
- `uid 01:00:60:67:75:40:37;` — необязательный параметр. Идентификатор клиента, основывается на протоколе ARP и для Ethernet-карты представляет собой MAC-адрес, впереди которого добавлена единица;
- `client-hostname "Oscar";` — имя хоста клиента.

Помимо этого есть еще десяток параметров, с которыми вы можете ознакомиться в справке к программе dhcpd.

Пример файла dhcpd.conf

Теперь, когда мы имеем представление о структуре файла /usr/local/etc/dhcpd.conf, рассмотрим конфигурационный файл (листинг 18.3), в котором описывается простая локальная сеть со следующими характеристиками:

- адрес сети 192.168.1.0;
- маска сети 255.255.255.0;
- домен test.org;
- один DHSP-сервер с адресом 192.168.0.2;
- один DNS-сервер с адресом 192.168.0.3;
- один шлюз с адресом 192.168.0.1;
- 11 клиентов, получающих адреса, причем один из них всегда постоянен.

Листинг 18.3

```
#global options
ddns-update-style none;
option domain-name "test.org";
option domain-name-servers 192.168.10.3;

# 7 X 24 hours - lease time
default-lease-time 604800;
max-lease-time 604800;
# my subnet
subnet 192.168.1.0 netmask 255.255.255.0 {
    range 192.168.10.5 192.168.10.20;
    option routers 192.168.1.1;
}
host vasya {
hardware ethernet 00:70:58:bc:10:03;
fixed-address 192.168.1.17;
}
```

Что же мы видим в листинге 18.3? Первая строка — комментарий — показывает, что дальше идет описание глобальных параметров. В следующей строке мы запрещаем использовать динамический DNS. Затем мы назначаем имя домена и адрес DNS-сервера. После этого устанавливаем время аренды IP-адреса клиентами в семь суток. Потом определяется наша подсеть, в описании которой мы задаем диапазон динамических IP-адресов и адрес маршрутизатора. Наконец, последняя секция — определяем хост со статическим IP-адресом с именем `vasya`.

DHCP-клиент

Установка DHCP-клиента и DHCP-сервера ничем не различаются. Если пакет отсутствует в вашем дистрибутиве, его можно получить на сайте **http://www.isc.org**.

После установки клиента нам необходимо его сконфигурировать. Для этого нужно разобраться со следующими файлами:

- etc/dhclient.conf;
- /var/lib/dhcp/dhclient.leases.

Файл dhclient.conf

Файл dhclient.conf отвечает за конфигурацию DHCP-клиента. Может содержать следующие параметры:

- `timeout` *`секунды;`* — время (в секундах), которое клиент ожидает ответа от сервера (по умолчанию 60 с);
- `retry` *`секунды;`* — время ожидания клиента перед повторением запроса к серверу, если предыдущий запрос окончился неудачей;
- `reboot` *`секунды;`* — время ожидания клиента выделения того же самого IP-адреса при перезагрузке. Если за это время DHCP-сервер не предоставляет старый IP-адрес, клиент делает попытку получить новый IP-адрес;
- `initial-interval` *`секунды;`* — приблизительный интервал времени между первой попыткой послать сообщение серверу и второй. При каждой попытке этот параметр увеличивается на некоторую величину;
- `select-timeout` *`секунды;`* — этот параметр используется в том случае, если в сети находятся несколько DHCP-серверов. Он задает время ожидания ответов от DHCP-серверов, по истечении которого клиент определяет, с каким сервером он будет сотрудничать;
- `reject` *`IP-адрес;`* — указывает, с каким DHCP-сервером клиент не должен сотрудничать.

Следующие три параметра отвечают за временные настройки аренды IP-адреса. Обычно эти параметры выдают DHCP-серверы, но их можно явно указать в конфигурационном файле:

- `renew` *`дата;`* — момент времени, когда необходимо попытаться продлить время аренды у сервера, выдавшего IP-адрес;
- `rebind` *`дата;`* — момент времени, когда клиент должен произвести попытку продлить аренду IP-адреса у любого доступного DHCP-сервера;
- `expire` *`дата;`* — время истечения аренды IP-адреса.

В файле dhclient.conf, подобно dhcpd.conf, можно определять групповые параметры.

Например, `interface "Имя интерфейса"`. В этом групповом параметре можно задать настройки, специфичные для данного интерфейса.

Давайте рассмотрим описание параметров сетевой карты DHCP-клиента (листинг 18.4).

Листинг 18.4

```
interface "eth0" {
   send host-name "andare.fugue.com";
   send dhcp-client-identifier 1:0:a0:24:ab:fb:9c;
   send dhcp-lease-time 3600;
   request subnet-mask, broadcast-address, routers,
   domain-name, domain-name-servers;
   }
```

В первой строке содержится определение, к какому интерфейсу относятся параметры. Затем идет группа строк, начинающихся с `send`. В этих параметрах задаются те переменные, которые клиент отсылает на сервер для своей идентификации. В данном случае это имя хоста, идентификатор хоста (MAC-адрес сетевой карты, в начале которого добавлена единица) и время аренды IP-адреса в секундах. Затем идет строка, начинающаяся с `request`. Она определяет переменные, которые запрашивает клиент от сервера. В нашем примере это маска подсети, широковещательный адрес, адрес маршрутизаторов, имя домена и адреса DNS-серверов.

Существуют еще несколько малоиспользуемых параметров, описание которых вы можете посмотреть в литературе.

Но самое интересное, что для функционирования DHCP-клиента в простейшей локальной сети нет необходимости заполнять файл dhclient.conf! Достаточно просто факта его существования.

Файл dhclient.leases

Файл dhclient.leases подобно файлу dhcpd.leases представляет собой базу данных, в которой хранятся параметры DHCP-клиента, полученные от DHCP-сервера (листинг 18.5).

Листинг 18.5

```
lease {
  interface "eth0";
  fixed-address 192.168.1.15;
  option subnet-mask 255.255.240.0;
  option routers 192.168.1.1;
  option domain-name-servers 192.168.1.3;
```

```
option broadcast-address 255.255.255.255;
option dhcp-server-identifier 192.168.1.2
option host-name "vasya";
option domain-name "test.org";
renew 3 2008/4/2 00:22:38;
rebind 6 2008/4/5 02:50:06;
expire 6 2008/4/5 23:50:06;
}
```

Из листинга видно, что параметры относятся к сетевому интерфейсу `eth0`. Этой сетевой карте выдан IP-адрес, равный 192.168.1.15, маска подсети 255.255.240.0, адрес маршрутизатора, DHCP-сервера и DNS-сервера соответственно 192.168.1.1, 192.168.1.2 и 192.168.1.3. Имя хоста — `vasya`, домен — `test.org`. Последние три записи определяют временные параметры аренды IP-адреса.

Ссылки

- Журнал "Сети" 1999 г. № 10. Иванов П. DHCP: искусство управления IP-адресами.
- **www.asmodeus.com.ua/library/net/dhcp_linux.htm** — Калошин В. Настраиваем DHCP.
- **ezine.daemonnews.org/200207/dhcp.html** — Pham Linh. HOWTO — Setting Up ISC-DHCP 3.x Under FreeBSD.
- **www.dhcp.org** — сервер, полностью посвященный протоколу DHCP.
- **mvd.h1.ru/tr/** — DHCP mini-HOWTO, русский перевод.
- **www.isc.org** — Internet Software Consortium (разработчик DHCP).
- **www.nominum.com/resources/faqs/dhcp-faq.html** — Nominum's DHCP FAQ.
- **www.onlamp.com/pub/a/bsd/2003/04/17/** — Lavigne Dru. Introducing DHCP.
- **www.onlamp.com/pub/a/bsd/2003/05/01/FreeBSD_Basics.html** — Lavigne Dru. Configuring a DHCP Server.
- **www.onlamp.com/lpt/a/3689** — Lavigne Dru. DHCP on a Multi-Segment Network.
- man dhcpd.conf.
- man dhcpd.leases.
- man dhcp-options.
- man dhclient.leases.

Глава 19



Почта

Эта глава посвящена электронной почте, тому, с чего и начинался Интернет. Функционирование электронной почты очень похоже на свой "бумажный" прототип. Давайте представим, как работает обычная почта. Человек пишет письмо, подписывается, указывает на конверте адрес отправителя и получателя, запечатывает конверт и опускает его в почтовый ящик. Специальная служба, условно назовем ее курьерской, периодически объезжает почтовые ящики, собирает письма и отвозит их на почтамт. Там их сортируют и отправляют на почтамты в города назначения. Оттуда после сортировки по районам и адресатам почтальоны доставляют письма по индивидуальным почтовым ящикам.

Приблизительно по такому же принципу работает и электронная почта. Есть программа — почтовый клиент, в которой пользователь подготавливает к отправке и получает письма. Есть программа — транспортный агент, которая отвечает за доставку электронной почты от компьютера к компьютеру, и есть программы, выполняющие роль почтамтов — они получают почту, сортируют ее по адресатам и раскладывают по почтовым ящикам.

В качестве почтового клиента выступают десятки программ: mail, Pine, Kmail, Evolution, Sylpheed, Mutt и многие другие. Для транспортировки почты используется MTA, Mail Transport Agent — почтовый транспортный агент. Старейший и наиболее распространенный транспортный агент — программа sendmail. Также получили популярность программы Qmail, postfix, exim.

Почтовая служба основана на системе адресов. В Интернете принята система адресов, которая базируется на доменном адресе машины, подключенной к сети. Например, для пользователя ivan машины с адресом ogpu.odessa.ua почтовый адрес будет выглядеть так: **ivan@ogpu.odessa.ua.**

Почтовый адрес состоит из двух частей: идентификатора пользователя, который записывается перед знаком "коммерческого at" — "@", и доменного адреса компьютера после этого знака. Существует еще один вариант задания почтового адреса — адрес UUCP (UNIX to UNIX Copy Program), который записывается в виде: **odessa.ua!ogpu!ivan**. Правда, протокол UUCP сейчас почти не встречается.

Для работы электронной почты разработан специальный протокол Simple Mail Transfer Protocol (SMTP) — простой почтовый протокол, который является протоколом прикладного уровня и использует транспортный протокол TCP.

Протокол SMTP

Simple Mail Transfer Protocol был разработан для обмена почтовыми сообщениями в сети Интернет. SMTP не зависит от транспортной среды и может применяться для доставки почты в сетях с протоколами, отличными от TCP/IP.

Взаимодействие в рамках SMTP строится по принципу двусторонней связи, которая устанавливается между отправителем и получателем почтового сообщения. При этом отправитель инициирует соединение и посылает запросы на обслуживание, а получатель на эти запросы отвечает.

Как и множество других протоколов, команды и ответы протокола SMTP передаются в ASCII-кодах и представляют собой небольшой набор английских слов.

В листинге 19.1 приведен простой пример отправки почтового сообщения по протоколу SMTP.

Листинг 19.1

```
Отправитель: MAIL FROM: <ivan@ogpu.odessa.ua>
Получатель: 250 Ok
Отправитель: RCPT TO: <vano@mail.ru>
Получатель: 250 Ok
Отправитель: DATA
Получатель: 354 Start mail input; end with <CRLF>.<CRLF>
Текст почтового сообщения
Отправитель:
Получатель: 250
```

Протокол, помимо отправки почты, поддерживает переадресацию, прямую посылку сообщения на терминал, обработку ошибок и некоторые другие возможности.

Протокол POP3

Протокол обмена почтовой информацией POP3 (Post Office Protocol) предназначен для получения почты из почтовых ящиков пользователей на их рабочие места при помощи программ-клиентов. Таким образом, по протоколу SMTP пользователи отправляют корреспонденцию, а по протоколу POP3 — получают ее из своих почтовых ящиков на почтовом сервере в локальные файлы. Этот протокол также основан на установлении двусторонней связи, команды и ответы протокола передаются в ASCII-кодах и представляют собой небольшой набор английских слов.

Протокол IMAP

Еще один протокол разбора почты — IMAP — почтовый протокол интерактивного доступа (Interactive Mail Access Protocol) по своим возможностям похож на POP3, но разрабатывался как более надежная альтернатива последнему. Он обладает

более широкими возможностями по управлению процессом обмена сообщениями с сервером.

Главное отличие его от POP3 — возможность поиска нужного сообщения и разбор заголовков сообщения непосредственно на почтовом сервере.

Формат почтового сообщения

Формат почтового сообщения определен в документе RFC-822. Почтовое сообщение состоит из трех частей: *конверта*, *заголовка* и *тела сообщения*. Конверт используется программами доставки почтового сообщения, а заголовок и тело сообщения предназначены для его получателя. Заголовок находится перед телом сообщения, отделен от него пустой строкой и состоит из определенных стандартом полей. Поля включают имя и содержание. Имя поля отделяется от содержания символом ":". Для доставки сообщения можно воспользоваться только такими частями полей заголовка, как `Date`, `From`, `Cc` или `To`, например:

```
Date:  26 Aug 76 1429 EDT
From:  1@mail.ru
To:    Sm2@chat.ru
```

Поле `Date` определяет дату отправки сообщения, поле `From` — отправителя, а поля `Cc` и `To` — получателей. Однако если следовать установленным правилам, необходимо определять все поля заголовка, описанные в стандарте (листинг 19.2).

Листинг 19.2

```
Date:          27 Aug 76 0932
From:          Motya <1@mail.ru>
Subject:       Re: Ответ на письмо
Sender:        K@Other-host
Reply-To:      Sam.Irving@R.org.ru
To:            Geo <J@chat.ru>
Cc:            Sm3@chat.ru
Comment:       Sam is away on bisiness
In-Reply-To:   <some.string@DBM.Group>, George's message
Message-ID:    <4331.629.XYzi-What@Other-Host
```

Поле `Subject` определяет тему сообщения, `Reply-To` — пользователя, которому отвечают, `Comment` — комментарий, `In-Reply-To` — показывает, что сообщение относится к типу "В ответ на Ваше сообщение, отвечающее на сообщение, отвечающее ...", `Message-ID` — уникальный идентификатор письма, используемый почтовыми программами.

Формат сообщения постоянно дополняется и совершенствуется. В частности в RFC-1327 введены дополнительные поля для совместимости с почтой X.400.

Спецификация MIME

Спецификация MIME (Multipurpose Internet Mail Extension), приведенная в стандарте RFC-1341, предназначена для описания тела почтового сообщения Интернета. Необходимость в этом документе возникла в силу того, что по стандарту RFC-822 в тело почтового сообщения не могут быть включены некоторые специальные и восьмибитовые символы.

Стандарт RFC-822 подробно описывает в заголовке почтового сообщения текстовое тело письма и механизм его рассылки, а MIME сориентирован на описание в заголовке письма структуры тела почтового сообщения и возможности составления письма из информационных единиц различных типов.

В стандарте зарезервировано несколько способов представления разнородной информации. Для этого предусмотрены специальные поля заголовка почтового сообщения:

- поле версии MIME — для идентификации сообщения, подготовленного в новом стандарте;
- поле описания типа информации в теле сообщения — обеспечивает правильную интерпретацию данных;
- поле типа кодировки информации в теле сообщения — указывает на тип процедуры декодирования;
- два дополнительных поля — зарезервированы для более детального описания тела сообщения.

Стандарт MIME разработан как расширяемая спецификация, в которой подразумевается, что число типов данных будет расти по мере развития их форм представления.

Рассмотрим некоторые из полей MIME.

MIME-Version

Поле версии относится ко всему сообщению целиком, указывается в его заголовке и позволяет определить, что сообщение подготовлено в стандарте MIME. Формат поля:

```
MIME-Version: 1.0
```

Content-Type

Поле типа служит для описания типа данных, которые содержатся в теле почтового сообщения. Это поле сообщает программе чтения почты, какие преобразования необходимы для того, чтобы правильно проинтерпретировать сообщение. Та же информация используется и программой рассылки при кодировании/декодировании почты. Стандарт MIME определяет семь типов данных, которые можно передавать в теле письма. Перечислим и кратко опишем важнейшие из них.

- Текст (text) — указывает на то, что в теле сообщения содержится текст. Основным подтипом типа text является plain — плоский (неразмеченный) текст. Размеченному тексту соответствует подтип richtext, а гипертексту — подтип html.

- Смешанный тип (multipart) — определяет смешанный документ, который может состоять из фрагментов данных разного типа. Имеет ряд подтипов.
- Сообщение (message) — предназначен для работы с обычными почтовыми сообщениями, которые напрямую нельзя передать по почте. Существует несколько подтипов:
  - partial — предназначен для передачи одного большого сообщения по частям для последующей автоматической сборки у получателя;
  - External-Body — позволяет ссылаться на внешние информационные источники;
  - rfc822 — стандартный подтип типа message. Определяет сообщения стандарта RFC-822.
- Графический образ (image).
- Аудиоинформация (audio).
- Видеоинформация (video).
- Приложение (application).

Content-Transfer-Encoding

Тип кодирования сообщения. Поскольку сообщения передаются в неоднородной среде, неизбежны их перекодировки. Данное поле используется для правильной распаковки данных при получении.

Спецификация S/MIME

S/MIME (Secure/Multipurpose Internet Mail Extensions) — безопасная реализация MIME, описана в RFC2630, RFC2632, RFC2633, RFC2634. S/MIME представляет собой реализацию криптографической системы с асимметричным ключом. Система подходит как для внедрения цифровой подписи в почтовые сообщения, так и для их шифрования. Поддерживается также комбинация двух перечисленных ранее методов. Системы с симметричным и асимметричным ключом различаются тем, что во второй отправителю не придется сообщать пароль, который был задействован для пересылки секретных сведений. Все, что необходимо иметь — публичную (несекретную) часть S/MIME-сертификата вашего адресата. Корреспондент должен иметь и секретную часть своего S/MIME-сертификата, иначе он не сможет дешифровать полученный текст.

В случае применения электронной подписи подписанное сообщение передается в незашифрованном виде, однако получатель письма имеет возможность убедиться в подлинности отправителя и в целостности (не искаженности) письма. При этом получателю не требуется никакой дополнительной информации, т. к. публичная часть S/MIME-сертификата отправителя подписанного письма передается вместе с этим письмом.

Для использования S/MIME (а также PGP или GPG) клиентское программное обеспечение должно уметь работать с данными протоколами. Большинство современных программ поддерживают такую возможность.

PGP, GPG

Подобно S/MIME, PGP (Pretty Good Privacy) и ее аналог из мира GNU GPG (GnuPG, GNU Privacy Guard) применяются для надежного шифрования почтовых сообщений и организации электронной подписи. Однако помимо электронной почты PGP служит (по крайней мере, достаточно широко в мире Windows) для шифрования файлов и организации шифрованных псевдодисков.

Программное обеспечение

Как и многое другое, взаимодействие между участниками обмена почтового сообщения основано на технологии "клиент-сервер". Можно выделить три независимых этапа:

- взаимодействие по протоколу SMTP между почтовым клиентом и почтовым транспортным агентом;
- взаимодействие между транспортными агентами в процессе доставки почты;
- получение сообщения из почтового ящика пользователя почтовым клиентом по протоколу POP3 или IMAP.

Программа sendmail

Основное средство рассылки почты — программа sendmail — одна из старейших и сложных в конфигурации. Sendmail позволяет организовать почтовую службу локальной сети и обмениваться почтой с другими серверами почтовых служб через специальные шлюзы. Sendmail можно сконфигурировать для работы с различными почтовыми протоколами (обычно UUCP и SMTP).

Sendmail может интерпретировать почтовые адреса SMTP и UUCP.

Sendmail можно настроить для поддержки:

- списка адресов-синонимов;
- списка адресов рассылки пользователя;
- автоматической рассылки почты через шлюзы;
- очередей сообщений для повторной рассылки почты в случае отказов при рассылке;
- работы в качестве SMTP-сервера;
- доступа к адресам машин через сервер доменных имен BIND;
- доступа к внешним серверам имен и др.

Принцип работы программы sendmail

Принцип работы sendmail аналогичен обычной почтовой службе — почта отправляется с заданной периодичностью, перед этим сообщения собираются в очереди и только затем отсылаются.

Как уже упоминалось ранее, каждое сообщение состоит из трех частей: конверта, заголовка и тела сообщения:

- *конверт* включает адреса отправителя и получателя, а также специфическую информацию, которая необходима программам подготовки, рассылки и получе-

ния почты. Конверт остается невидимым для отправителя и получателя почтового сообщения;

- *заголовок* состоит из стандартных текстовых строк с адресами, информацией о рассылке и данными. Данные из заголовка могут использоваться для оформления конверта сообщения;
- *тело сообщения* следует после первой пустой строки вслед за заголовком сообщения. Все, что расположено после этой строки, передается по почте без изменений.

После постановки почтовых сообщений в очередь начинается ее рассылка. При этом выполняются следующие действия:

- адреса отправителя и получателя преобразуются в формат сети — получателя почты;
- если необходимо, то в заголовок сообщения добавляются отсутствующие данные;
- почта передается одной из программ рассылки почты.

Настройка программы sendmail

Программу sendmail настраивают при помощи конфигурационного файла /etc/sendmail.cf, который состоит из нескольких частей:

- описания компьютера (local information) — в данной секции описывается имя компьютера и т. п.;
- описания макроопределений sendmail, отвечающих за работу в локальной сети;
- групп имен, которые используются программой для рассылки почты;
- номера версии файла конфигурации;
- опций команды sendmail, определяющих режимы работы программы;
- доверенных пользователей;
- описания формата заголовка почтового сообщения — в данной секции определяются поля и их формат, которые отображаются в заголовке;
- правил преобразования адресов;
- описания программ рассылки;
- общего набора правил преобразования адресов;
- машиннозависимой части общего набора правил преобразования адресов.

Обычно после инсталляции sendmail изменения, которые вносятся в файл конфигурации, касаются только имени хоста, домена и шлюзов. В современных дистрибутивах (таких как Red Hat) иногда не приходится делать даже этого.

Подробно на конфигурировании sendmail здесь останавливаться не будем — разобраться в конфигурационном файле, который имеет около 100 Кбайт текста, весьма не просто. Для детального ознакомления с конфигурацией sendmail рекомендуется почитать книгу "UNIX — руководство системного администратора", а также документацию, идущую в комплекте с sendmail.

Для примера приведем небольшую секцию локальной конфигурации программы sendmail (листинг 19.3).

Листинг 19.3

```
##################
#   local info   #
##################
Cwlocalhost
CP.
# UUCP relay host
DYucbvax.Berkeley.EDU
CPUUCP
#  BITNET relay host
#DBmailhost.Berkeley.EDU
DBrelay.kiae.su
CPBITNET
# "Smart" relay host (may be null)
DSrelay.kiae.su
# who I send unqualified names to (null means deliver locally)
DR
# who gets all local email traffic ($R has precedence for unqualified names)
DH
# who I masquerade as (null for no masquerading)
DM
# class L: names that should be delivered locally, even if we have a relay
# class E: names that should be exposed as from this host, even if we masquerade
#CLroot
CEroot
# operators that cannot be in local usernames (i.e., network indicators)
CO @ % !
# a class with just dot (for identifying canonical names)
C..
# dequoting map Kdequote dequote
```

Тестирование отправки почты sendmail

Для проверки правильности функционирования программы sendmail можно запустить ее с ключом `-v` (режим verbose). При этом процесс обмена между транспортными почтовыми агентами выводится на консоль или записывается в файл. Так можно исключить большую часть ошибок в настройке sendmail.

Тестирование обслуживания по протоколу SMTP

Сервис SMTP проверяют с помощью программы telnet, подключаемой к порту 25:

```
telnet ivan.petrov 25
```

Если на компьютере установлен SMTP-сервер, в ответ получим строку приглашения протокола SMTP, после чего можно вводить команды SMTP (листинг 19.4).

Листинг 19.4

```
MAIL FROM: user
250 user... Sender ok
RCPT TO: user
250 user... Recipient ok
DATA
354 Enter mail, end with "." on a line by itself
This is a test message!!!
...
250 JAA24856 Message accepted for delivery
quit
221 ivan.petrov closing connection
Connection closed by foreign host.
You have new mail.
#
```

В приведенном примере мы отправили сами себе сообщение. Команда `MAIL FROM:` указывает адрес отправителя почтового сообщения. Затем вводится команда `RCPT TO:` для указания адреса получателя почтового сообщения. Команда `DATA` разрешает ввод почтового сообщения. Конец режима редактирования обозначается символом "." в первой позиции строки. Более подробную информацию следует искать в документации по sendmail, а также в табл. 19.1, где приведены команды протокола SMTP, и в табл. 19.2, содержащей коды возврата протокола SMTP.

Команды и коды возврата протокола SMTP

Для тестирования работы SMTP-сервера необходимо знать команды протокола SMTP (табл. 19.1) и его коды возврата (табл. 19.2) и воспользоваться программой telnet.

Таблица 19.1. *Команды протокола SMTP*

Команда	Описание
`HELO <SP> <domain> <CRLF>`	Открыть сессию взаимодействия по протоколу SMTP. `<domain>` — доменное имя машины
`MAIL <SP> FROM:<reverse-path> <CRLF>`	Сообщить адрес отправителя `<reverse-path>`. Обязательная команда, которую нужно выдать перед отправкой сообщения
`RCPT <SP> TO:<forward-path> <CRLF>`	Сообщить адрес получателя <`forward-path`>. Обязательная команда, которую выдают после `MAIL FROM`, но перед `DATA`

Таблица 19.1 *(окончание)*

Команда	Описание
DATA <CRLF>	Начать передачу тела почтового сообщения. Тело сообщения должно завершаться точкой (.) в первой позиции строки
RSET <CRLF>	Конец операции
SEND <SP> FROM:<reverse-path> <CRLF>	Послать сообщение на терминал пользователя, который определяется командой RCPT
SOML <SP> FROM:<reverse-path> <CRLF>	SEND OR MAIL. Послать в почтовый ящик или на терминал пользователя
SAML <SP> FROM:<reverse-path> <CRLF>	SEND AND MAIL. Послать в почтовый ящик и на терминал пользователя
VRFY <SP> <string> <CRLF>	Получить информацию о пользователе, имя которого указывается в качестве аргумента команды <string>
EXPN <SP> <string> <CRLF>	Получить информацию о пользователях, зарегистрированных в качестве получателей корреспонденции
HELP [<SP> <string>] <CRLF>	Краткая справка по командам протокола
NOOP <CRLF>	Нет операции
QUIT <CRLF>	Завершить сессию
TURN <CRLF>	Поменять местами сервер и клиент

Таблица 19.2. *Коды возврата протокола SMTP*

Код возврата	Текстовое пояснение сервера	Описание
211	System status, or system help reply	Статус системы или помощь
214	Help message. [Information on how to use the receiver or the meaning of a particular non-standard command; this reply is useful only to the human user]	Краткая справка
220	<domain> Service ready	SMTP-сервис готов к работе
221	<domain> Service closing transmission channel	Сервис закрыл канал передачи данных
250	Requested mail action okay, completed	Соединение установлено
251	User not local; will forward to <forward-path>	Пользователь не местный. Выполнить перенаправление запроса

Таблица 19.2 *(окончание)*

Код возврата	Текстовое пояснение сервера	Описание
354	`Start mail input; end with <CRLF>.<CRLF>`	Начать ввод почтового сообщения
421	`<domain> Service not available, closing transmission channel [This may be a reply to any command if the service knows it must shut down]`	Сервис отсутствует. Канал передачи данных закрыт
450	`Requested mail action not taken: mailbox unavailable [E.g., mailbox busy]`	Нет возможности записать данные в почтовый ящик
451	`Requested action aborted: local error in processing`	Ошибка при обработке запроса
452	`Requested action not taken: insufficient system storage`	Запрос не выполнен — недостаточно памяти
500	`Syntax error, command unrecognized [This may include errors such as command line too long]`	Синтаксическая ошибка — нет такой команды
501	`Syntax error in parameters or arguments`	Синтаксическая ошибка в аргументах команды
502	`Command not implemented`	Данная команда не может быть выполнена
503	`Bad sequence of commands`	Неправильная последовательность команд
504	`Command parameter not implemented`	Параметр команды не может быть использован в данном контексте
550	`Requested action not taken: mailbox unavailable [E.g., mailbox not found, no access]`	Не найден соответствующий почтовый ящик
551	`User not local; please try <forward-path>`	Пользователь не найден; можно попробовать отправить почту по другому адресу
552	`Requested mail action aborted: exceeded storage allocation`	Превышены квоты на использование ресурсов памяти
553	`Requested action not taken: mailbox name not allowed [E.g., mailbox syntax incorrect]`	Неправильное имя почтового ящика
554	`Transaction failed`	Аварийное завершение

Тестирование обслуживания по протоколу POP3

Аналогично тестированию обслуживания по протоколу SMTP с помощью программы telnet можно проверить функционирование и протокола POP3. Для этого необходимо подключиться к нашему серверу по порту 110 (листинг 19.5).

Листинг 19.5

```
telnet ivan.petrov 110
  user user
  +OK Password required for user.
  pass 12345623432
  +OK user has 3 messages (33276 octets).
  list
  +OK 3 messages (33276 octets)
  1 11276
  2 11000
  3 11000
  .
  dele 3
  +OK Message 3 has been deleted.
  quit
  +OK
  Connection closed by foreign host.
```

Очень похоже на протокол SMTP. Подключились к порту 110. Выполняем "опознание" пользователя с помощью команд `user` и `pass`. Затем командой `list` узнаем количество сообщений в почтовом ящике и их размер. Командой `dele` отмечаем сообщение к удалению, которое произойдет по окончании сеанса. Команда `quit` завершает сеанс работы с сервером. Все просто.

Команды протокола POP3

Для тестирования работы POP3-сервера необходимо знать его команды (табл. 19.3) и воспользоваться программой telnet.

Успешное выполнение команды заканчивается выводом сообщения `+OK`, а неуспешное `-ERR` соответственно.

Таблица 19.3. *Команды протокола POP3*

Команда	Назначение	Возможные возвращаемые значения (кроме `+OK` или `-ERR`)
`USER <имя пользователя>`	Посылка имени пользователя серверу	
`PASS <пароль>`	Посылка пароля серверу	
`QUIT`	Окончание сеанса работы	
`STAT`	Получить состояние почтового ящика	`+OK <кол-во сообщений> <общий размер всех сообщений>`

Таблица 19.3 *(продолжение)*

Команда	Назначение	Возможные возвращаемые значения (кроме +OK или –ERR)
UST [<номер сообщения>]	Получить параметры всех сообщений в ящике пользователя. Если задан номер сообщения, то будут получены только его параметры	+OK <параметры сообщений> Возвращаемые параметры сообщений зависят от того, был ли задан номер сообщения. Если да, то сразу после +OK следует сообщение сервера. Затем строка за строкой передаются параметры всех сообщений в формате <номер сообщения> <размер сообщения>
RETR <номер сообщения>	Получить сообщение с сервера	+OK <тест запрошенного сообщения> — если команда прошла удачно -ERR <комментарий сервера> — если запрошенное сообщение отсутствует на сервере
DELE <номер сообщения>	Пометить сообщение на сервере как удаленное. Реально оно будет удалено после команды QUIT	+OK <комментарий сервера> — если сообщение было помечено на удаление -ERR <комментарий сервера> — если сообщение не существует или уже отмечено как удаленное
NOOP	Пустая операция	+OK
RSET	Отменить удаление сообщений, помеченных как удаленные	
TOP <номер сообщения> <кол-во строк>	Считать заголовок сообщения и первые строки в количестве, заданном параметром <кол-во строк>	+OK Далее строка за строкой передается заголовок сообщения. За ним следует пустая строка и, если имеется второй параметр, передаются начальные строки сообщения
UIDL [<номер сообщения>]	Получить уникальные идентификаторы всех сообщений в ящике пользователя. Если задан номер сообщения, то будет получен только его идентификатор	+OK <параметры сообщений> Возвращаемые параметры сообщений зависят от того, был ли задан номер сообщения. Если да, то сразу после +OK идут номер запрошенного сообщения и его идентификатор. Если команда вызвана без параметра, то после статуса +OK следует сообщение сервера. Затем строка за строкой передаются параметры всех сообщений в формате <номер сообщения> <идентификатор>

Таблица 19.3 *(окончание)*

Команда	Назначение	Возможные возвращаемые значения (кроме `+OK` или `-ERR`)
`APOP <имя пользователя> <дайджест>`	Осуществляет подключение к почтовому серверу по закодированной алгоритмом MD5 строке, защищая транзакцию от разглашения пароля пользователя	`+OK <комментарий сервера>` — если имя пользователя или дайджест соответствуют имеющемуся почтовому ящику пользователя

Программа Postfix

В последнее время Postfix становится популярным MTA-агентом. Sendmail сложна в настройке и проигрывает Postfix по ряду параметров. Для работы Postfix создает группы postfix и postdrop. В группе postfix добавляется пользователь postfix. Почтовая система работает с правами пользователя postfix, а группа postdrop владеет очередью сообщений.

Конфигурационные файлы

Файлы настройки располагаются в каталоге /etc/postfix. Основные параметры определяются в файле main.cf.

В первоначальном виде этот файл содержит конфигурацию, позволяющую серверу работать в пределах машины, а также развернутые комментарии с примерами.

Далее рассмотрим кратко конфигурирование Postfix. Опишем только те опции, которые необходимо изменить.

- `myhostname=tech.test.ru` — имя хоста. Лучше всего устанавливать в соответствии с результатами выполнения `hostname`.
- `mydomain=test.ru` — имя домена. Если переменная не определена, то Postfix использует подкорректированное значение `myhostname`.
- `inet_interfaces=192.168.0.2, 195.80.10.26` — адреса интерфейсов, на которых нужно ждать SMTP-соединений. Можно указать слово `all` для установки прослушивания всех интерфейсов.
- `mydestination=$myhostname, $mydomain` — список доменов, которые обслуживает программа.
- `mynetworks=192.168.0.0/24, 127.0.0.0/8` — задает список доверенных сетей. Клиенты с адресами, принадлежащими этим сетям, смогут рассылать через нас почту. Если этот параметр не определен, то доверенной сетью считается IP-подсеть, к которой принадлежит машина с Postfix.
- `alias_database=dbm:/etc/postfix/aliases` — путь к файлу почтовых псевдонимов.

Вот вкратце и все. Само собой, параметров намного больше, что позволяет очень тонко настроить программу. Однако для небольшой сети перечисленных операций достаточно.

Проверить корректность конфигурационного файла можно, выполнив команду `postfix check`.

После редактирования конфигурации при работе Postfix ее нужно активизировать командой `postfix reload`.

Почтовые клиенты

Сегодня существует несколько десятков почтовых клиентов — простейшие текстовые, сложные текстовые, графические и даже Web-клиенты. Трудно охватить все разнообразие, поэтому приведем примеры нескольких почтовых клиентов. Для настройки всех почтовых клиентов необходимо знать ряд параметров:

- ❒ логин пользователя;
- ❒ пароль пользователя;
- ❒ адрес SMTP-сервера;
- ❒ порт SMTP-сервера;
- ❒ адрес POP3- или IMAP-сервера;
- ❒ порт POP3- или IMAP-сервера.

Если сеть на вашем компьютере функционирует правильно, то, зная эти параметры, не составляет труда настроить практически любой почтовый клиент.

Mail

Один из первых почтовых клиентов. Программа не отличается красотой интерфейса (его просто нет), достаточно примитивна, зато не занимает много места и поэтому присутствует практически на любом компьютере. Представление о программе Mail можно получить из рис. 19.1.

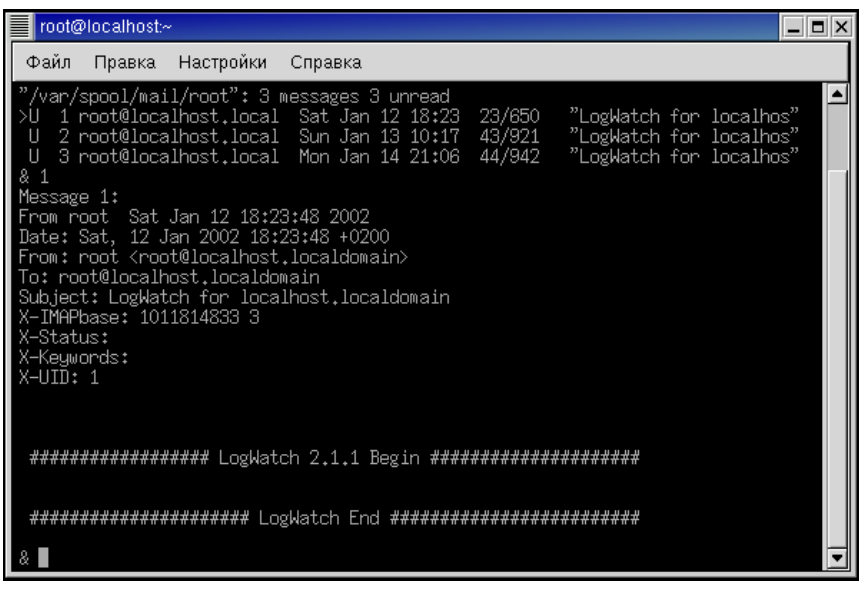


Рис. 19.1. Почтовый клиент Mail

Pine

Один из самых "навороченных" текстовых почтовых клиентов, который также позволяет работать с сообщениями новостей (news). Обладает удобным дружественным интерфейсом. Внешний вид программы Pine изображен на рис. 19.2.

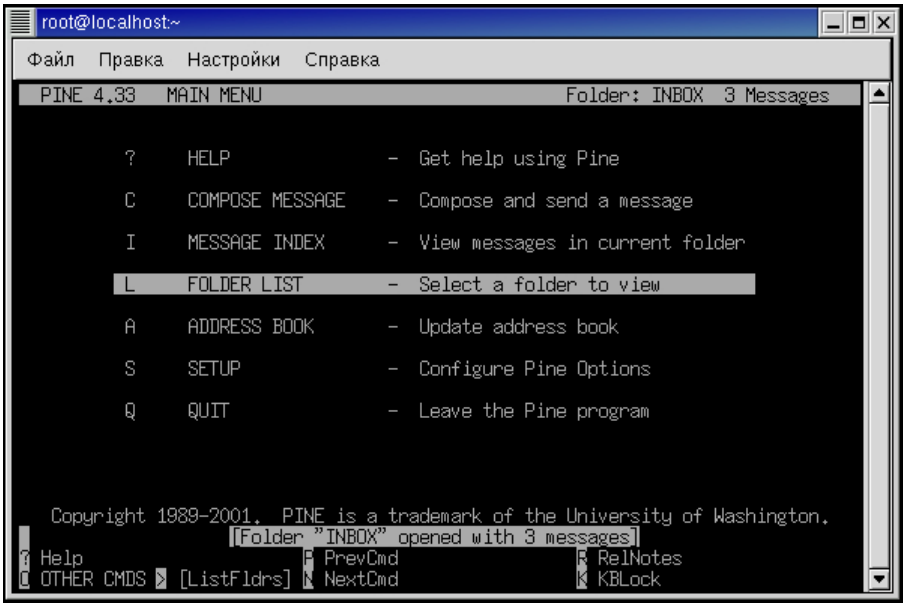


Рис. 19.2. Почтовый клиент Pine

Thunderbird mozilla

Устойчивая и надежная почтовая программа. Является частью программного комплекса "Mozilla — Web-браузер", почтовый клиент, клиент чата. Существует версия для Windows.

Sylpheed

Хороший почтовый клиент. Существует версия для Windows (рис. 19.3).

Evolution

Попытка программистов создать нечто подобное Microsoft Outlook — почтовый клиент (рис. 19.4), органайзер (рис. 19.5), дневник и записная книжка в одном комплекте. Получился довольно "увесистый" программный пакет.

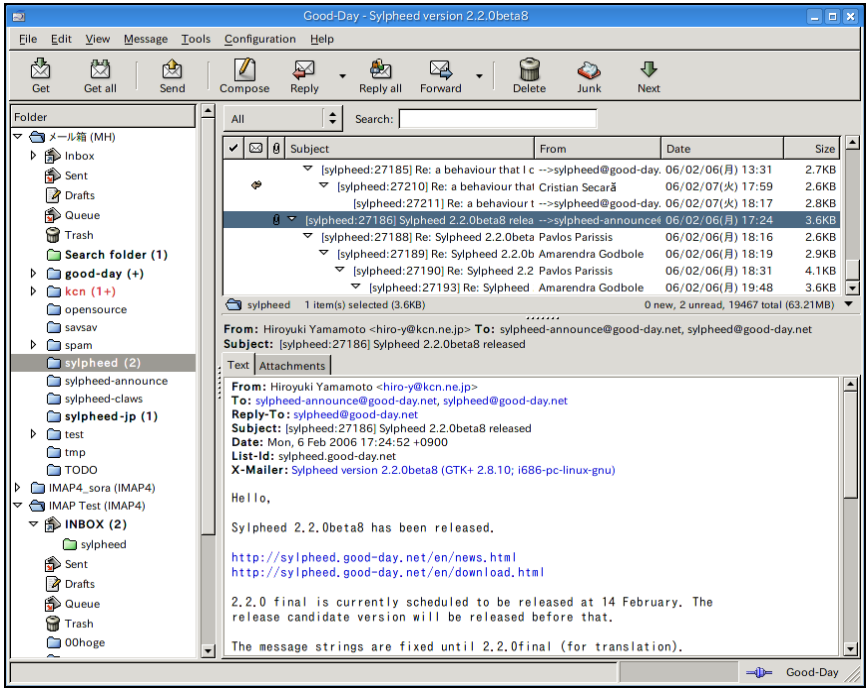


Рис. 19.3. Почтовый клиент Sylpheed

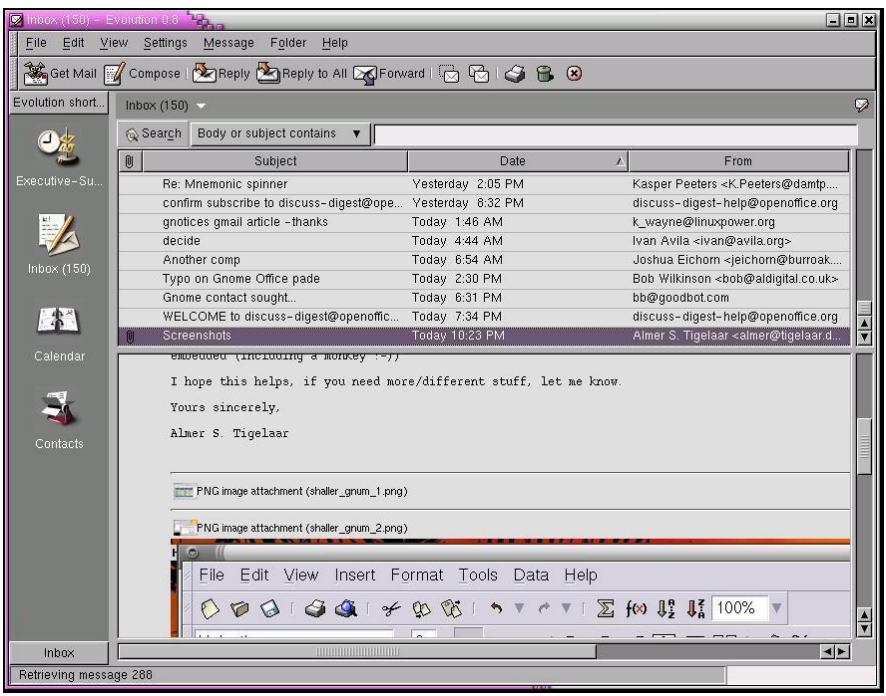


Рис. 19.4. Почтовый клиент Evolution

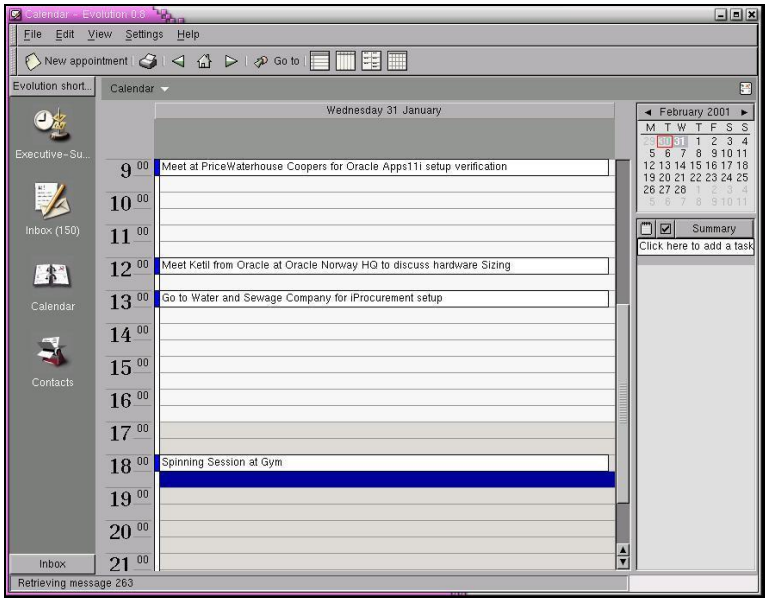


Рис. 19.5. Почтовый клиент Evolution — планировщик встреч

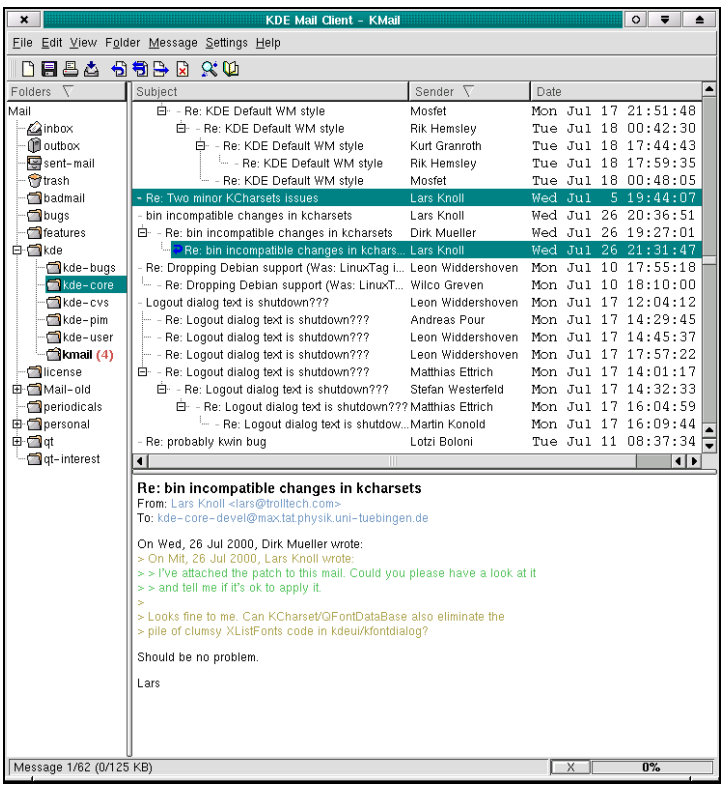


Рис. 19.6. Почтовый клиент Kmail

Kmail

Очень хороший почтовый клиент (рис. 19.6). Работает с различными кодировками, отличается удобным и понятным интерфейсом. Стандартный почтовый клиент для KDE.

Ссылки

- **www.citforum.ru/internet/servers/** — Павел Храмцов. Организация и администрирование почтовых и файловых серверов Internet. Центр информационных технологий.
- Соответствующие HOWTO:
  - Linux Mail-Queue mini-HOWTO;
  - Sendmail+UUCP HOWTO;
  - Sendmail address rewriting mini-HOWTO.

Глава 20



Web-сервер Apache

В качестве HTTP-сервера в UNIX-сообществе в основном используется Web-сервер Apache, который распространяется по лицензии GNU. По статистическим данным более половины всех Web-серверов в Сети созданы на базе Apache.

Чем же привлекателен этот сервер? Во-первых, большое количество возможностей: CGI-скрипты, шифрование, доступ по паролю, перекодирование страниц "на лету", поддержка виртуальных хостов и многое другое. Во-вторых, малая требовательность к ресурсам и большая производительность. В-третьих, многоплатформенность — Apache есть для Linux, для различных клонов UNIX и для Windows. В-четвертых, он бесплатный и с открытым исходным кодом. Список можно продолжать. Конечно, есть и недостатки, к примеру, некоторые сложности с конфигурированием. Но в целом этот сервер не зря получил столь большую популярность.

Второе место по количеству установок занимает Web-сервер Microsoft IIS, который входит в стандартную поставку Windows-сервера.

Microsoft IIS — мощный сервер, который по основным параметрам находится на уровне сервера Apache. Однако у Microsoft IIS также есть недостатки и основной из них — платформозависимость. Он функционирует только под управлением операционной системы семейства Windows.

В качестве альтернативы для Linux-платформы можно использовать Web-сервер TUX, который тесно интегрирован с ядром Linux, что позволило резко увеличить количество обрабатываемых запросов за единицу времени. Однако у этого сервера есть несколько минусов, в том числе:

- платформозависимость;
- неустоявшийся код;
- мало дополнительных возможностей по сравнению с Apache.

Конфигурация

Установка сервера для дистрибутивов с RPM-пакетами стандартна — необходимо скачать нужный пакет и задать команду

```
rpm -I <имя_пакета>
```

Конфигурирование сервера довольно сложно — несколько сотен команд и параметров, некоторые из них нужны крайне редко. Поэтому далее рассматриваются наиболее распространенные директивы и их параметры на примере типового конфигурационного файла.

Примечание

В том случае, если проводится переконфигурирование на рабочем сервере Apache, вам необходимо дать указание серверу перечитать конфигурационные файлы. Сервер перечитывает конфигурационные файлы при запуске либо при получении сигнала `-HUP` или `-USR1`. Если сервер Apache находится в работе, то при изменении конфигурации его рекомендуется перезапустить командой `kill -USR1`, поскольку при этом текущие соединения завершаются обычным образом, а следующие клиенты работают уже с новыми конфигурационными файлами.

Конфигурация сервера задается в файлах httpd.conf, srm.conf, access.conf и .htaccess. Файл httpd.conf предназначен для общей конфигурации сервера, srm.conf содержит описание доступных ресурсов, а access.conf — права доступа к ресурсам. Однако в современных версиях сервера любая директива конфигурации может находиться в любом из этих файлов. Сейчас де-факто все директивы конфигурации содержатся в файле httpd.conf.

Некоторые модули могут иметь свои отдельные файлы конфигурации (например, mod_charset требует файлы, хранящие таблицы перекодировки).

Большинство конфигурационных файлов находятся в каталогах /etc/httpd/conf и /etc/httpd/conf.d.

Листинг 20.1 иллюстрирует структуру конфигурационного файла httpd.conf (основные моменты).

Листинг 20.1

```
...
### Section 1: Global Environment
...
ServerTokens OS
ServerRoot "/etc/httpd"
PidFile run/httpd.pid
Timeout 120
KeepAlive Off
MaxKeepAliveRequests 100
KeepAliveTimeout 15
<IfModule prefork.c>
StartServers       8
MinSpareServers    5
MaxSpareServers   20
ServerLimit      256
MaxClients       256
MaxRequestsPerChild  4000
</IfModule>

<IfModule worker.c>
StartServers         2
```

```
MaxClients         150
MinSpareThreads     25
MaxSpareThreads     75
ThreadsPerChild     25
MaxRequestsPerChild  0
</IfModule>

Listen 80
...
Include conf.d/*.conf
User apache
Group apache

### Section 2: 'Main' server configuration
...
ServerAdmin root@localhost
UseCanonicalName Off
DocumentRoot "/var/www/html"

<Directory />
    Options FollowSymLinks
    AllowOverride None
</Directory>

<Directory "/var/www/html">
    Options Indexes FollowSymLinks
    AllowOverride None
    Order allow,deny
    Allow from all
</Directory>

<IfModule mod_userdir.c>
    UserDir disabled
</IfModule>

DirectoryIndex index.html index.html.var
AccessFileName .htaccess
<Files ~ "^\.ht">
    Order allow,deny
    Deny from all
</Files>
TypesConfig /etc/mime.types
```

```
DefaultType text/plain
<IfModule mod_mime_magic.c>
#   MIMEMagicFile /usr/share/magic.mime
    MIMEMagicFile conf/magic
</IfModule>
HostnameLookups Off
#EnableSendfile off
ErrorLog logs/error_log
LogLevel warn
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\"" combined
LogFormat "%h %l %u %t \"%r\" %>s %b" common
LogFormat "%{Referer}i -> %U" referer
LogFormat "%{User-agent}i" agent
CustomLog logs/access_log combined
ServerSignature On
Alias /icons/ "/var/www/icons/"

<Directory "/var/www/icons">
    Options Indexes MultiViews FollowSymLinks
    AllowOverride None
    Order allow,deny
    Allow from all
</Directory>
<IfModule mod_dav_fs.c>
    # Location of the WebDAV lock database.
    DAVLockDB /var/lib/dav/lockdb
</IfModule>

ScriptAlias /cgi-bin/ "/var/www/cgi-bin/"

<Directory "/var/www/cgi-bin">
    AllowOverride None
    Options None
    Order allow,deny
    Allow from all
</Directory>

IndexOptions FancyIndexing VersionSort NameWidth=* HTMLTable Charset=UTF-8
AddIconByEncoding (CMP,/icons/compressed.gif) x-compress x-gzip
...
AddIcon /icons/binary.gif .bin .exe
...
```

```
DefaultIcon /icons/unknown.gif

ReadmeName README.html
HeaderName HEADER.html
IndexIgnore .??* *~ *# HEADER* README* RCS CVS *,v *,t

AddLanguage ca .ca
...
LanguagePriority en ca cs da de el eo es et fr he hr it ja ko ltz nl nn no pl
pt pt-BR ru sv zh-CN zh-TW

ForceLanguagePriority Prefer Fallback
AddDefaultCharset UTF-8
AddType application/x-compress .Z
AddType application/x-gzip .gz .tgz
AddType application/x-x509-ca-cert .crt
AddType application/x-pkcs7-crl    .crl

AddHandler type-map var
AddType text/html .shtml
AddOutputFilter INCLUDES .shtml
Alias /error/ "/var/www/error/"

<IfModule mod_negotiation.c>
<IfModule mod_include.c>
    <Directory "/var/www/error">
        AllowOverride None
        Options IncludesNoExec
        AddOutputFilter Includes html
        AddHandler type-map var
        Order allow,deny
        Allow from all
        LanguagePriority en es de fr
        ForceLanguagePriority Prefer Fallback
    </Directory>

</IfModule>
</IfModule>

BrowserMatch "Mozilla/2" nokeepalive
...
```

```
# enable the proxy server:
...
# End of proxy directives.

### Section 3: Virtual Hosts

#<VirtualHost *:80>
#    ServerAdmin webmaster@dummy-host.example.com
#    DocumentRoot /www/docs/dummy-host.example.com
#    ServerName dummy-host.example.com
#    ErrorLog logs/dummy-host.example.com-error_log
#    CustomLog logs/dummy-host.example.com-access_log common
#</VirtualHost>
```

Рассмотрим подробнее этот файл. Разработчики Apache разбили его на три раздела.

1. Директивы, отвечающие за "глобальное" поведение Apache.
2. Параметры, определяющие поведение "основного" хоста и всех виртуальных хостов (за исключением особенностей, описанных при конфигурировании виртуальных хостов).
3. Конфигурирование виртуальных хостов. Если вы хотите использовать какие-то настройки по умолчанию, закомментируйте ненужные директивы. В противном случае явно пишите `On` — для включения директивы, `Off` — для ее выключения, `allow` — для разрешения и `deny` — для запрещения действия.

Раздел "Глобальные переменные"

В этом разделе определяются свойства Apache, влияющие на всю его инфраструктуру, например общее количество одновременных подключений к серверу или месторасположение конфигурационных файлов. Пройдемся по основным параметрам этого раздела.

- `ServerTokens OS` — разрешает получение дополнительной информации о сервере, версию, установленные модули. Если вы не хотите ее предоставлять, закомментируйте параметр.
- `ServerRoot "/etc/httpd"` — указывает, где находится дерево каталогов, содержащее конфигурационные файлы.
- `PidFile run/httpd.pid` — указывает серверу, где находится файл, в который сервер записывает идентификационный номер процесса при старте.
- `Timeout 120` — интервал времени (в секундах), по прошествии которого сервер разрывает соединение по тайм-ауту.
- `KeepAlive Off`, `MaxKeepAliveRequests 100`, `KeepAliveTimeout 15` — эти параметры разрешают несколько конкурентных запросов с одного соединения, ограничивают число конкурентных запросов и разрыв их по тайм-ауту (15 с). Обычно нет необходимости трогать эти параметры.

- `Listen 12.34.56.78:80` — важный параметр. Указывает серверу, какие IP-адреса и порты нужно слушать. Обычно используется в том случае, если в компьютере несколько сетевых интерфейсов и нужно разрешить работать серверу с одним или несколькими интерфейсами. Можно также переопределить этот параметр при создании виртуальных хостов.

Далее идет блок директив вида `LoadModule foo_module modules/mod_foo.so`. Apache — очень гибкая система, и в данном случае он посредством директив `LoadModule` подгружает различные функциональные модули, например несколько модулей аутентификации, модули поддержки различных типов данных и т. п.

- `Include conf.d/*.conf` — показывает серверу, что ему необходимо подключить все конфигурационные файлы из каталога conf.d.
- `User apache`, `Group apache` — эти директивы указывают серверу, от чьего имени и какой рабочей группы необходимо запускать исполняемые модули.

Раздел "Конфигурация «основного» сервера"

Директивы этого раздела служат для конфигурирования "основного" сервера. В том случае, если в системе есть еще виртуальные серверы, то все, что не подлежит "юрисдикции" раздела виртуальных серверов, обрабатывается по правилам основного сервера. При описании виртуального сервера можно переопределить все параметры, заданные в этом разделе.

- `ServerAdmin root@localhost` — адрес электронной почты, куда сервер будет отправлять сообщения о возникших проблемах.
- `ServerName www.example.com:80` — имя и порт, с которым будет себя отождествлять сервер. При отсутствии имени можно указать IP-адрес.
- `DocumentRoot "/var/www/html"` — каталог, в котором хранятся файлы, отображаемые сервером. Данные берутся из этого каталога, но при помощи символических ссылок их можно получить и из другого места.
- `<Directory Имя_каталога>…</Directory>` — позволяет задать различное поведение сервера для разных каталогов, например разрешить или запретить индексацию, модификацию правил при помощи файла .htaccess и т. п.
- `DirectoryIndex index.html index.html.var` — определяет, какой файл будет отображать сервер при попытке просмотра каталога.
- `AccessFileName .htaccess` — имя файла, в котором сервер ищет правила модификации доступа и прав для текущего каталога.
- `ErrorLog logs/error_log` — каталог и имя файла для ведения журнала ошибок.
- `LogLevel warn` — определяет, события какого уровня будут записываться в журнал ошибок. Может принимать значения `debug`, `info`, `notice`, `warn`, `error`, `crit`, `alert`, `emerg`.
- `CustomLog logs/access_log combined` — журналы событий доступа могут записываться раздельно (отдельно адрес, отдельно браузер и т. п.), но по умолчанию они комбинируются и пишутся в один файл access_log.
- `AddDefaultCharset UTF-8` — определяет, в какой кодировке сервер отдает текстовые файлы.

Раздел "Виртуальные серверы"

В этом разделе мы описываем виртуальные серверы. Виртуальные серверы — интересный способ оптимизации нагрузки на аппаратуру и экономии адресного пространства — в последнее время повсеместно распространены для Web-сайтов с относительно небольшой нагрузкой (десятки запросов в секунду). Идея очень проста — на мощном сервере Apache конфигурируется таким образом, что на одном IP-адресе находятся несколько Web-серверов с различными символическими именами и разным физическим пространством для организации структуры Web-страниц.

Виртуальный хост задается блоком `<VirtualHost 192.168.33.4:80> … </VirtualHost>`, где `192.168.33.4` — адрес сетевой карты, например `www.gost.ru`, а `80` — порт, по которому будет отвечать сервер на обращение по этому адресу.

Внутри блока мы можем задавать различные директивы, относящиеся к работе конкретно этого виртуального хоста (листинг 20.2).

Листинг 20.2

```
<VirtualHost www.gost.ru:80>
#    ServerAdmin webmaster@gost.ru
#    DocumentRoot /www/docs/gost.ru
#    ServerName www.gost.ru
#    ErrorLog logs/www.gost.ru-error_log
#    CustomLog logs/www.gost.ru-access_log common
#</VirtualHost>
```

Здесь мы определили электронный адрес администратора сайта, имя сервера и имена журналов событий для сервера.

Файл access.conf

В access.conf содержатся директивы, описывающие права доступа к каталогам и файлам Web-сервера. Обычно создается каталог /www/<*имя_сервера*>/, потому что при такой организации проще ориентироваться в структуре файлов.

Файл access.conf содержит секции `Directory`, `Location` и `Files`, которые ограничены одноименными директивами. В параметрах этих директив допустимы символы `?` и `*`, а также регулярные выражения, предваряемые тильдой `~`. В секции `Directory` помещаются инструкции, относящиеся к определенному каталогу на диске, в секции `Location` — относящиеся к виртуальному пути, в секции `Files` — относящиеся к файлу или группе файлов (листинг 20.3).

Листинг 20.3

```
<Directory /www/lazzycat.com>
# директивы, относящиеся ко всем документам, хранящимся в
каталоге /www/lazzycat.com и вложенным в него
```

```
</Directory>

<Location /cgi-bin>
# директивы, относящиеся ко всем документам, доступным по
адресу http://<имя_сервера>/cgi-bin/ <путь_к_файлу>
</Location>

<Files /www/lazzycat.com/form.htm>
# директивы, относящиеся к файлу form.htm из каталога
 /www/ lazzycat.com
</Files>
```

Различие между секциями `Directory` и `Location` состоит в том, что первая относится к каталогам на диске, вторая — к виртуальному пути (URL), который браузер запрашивает у Web-сервера. И в той, и в другой могут присутствовать директивы `order`, `allow` и `deny`, которые позволяют ограничить доступ к каталогу или URL с различных машин.

При отсутствии специальных требований к безопасности можно указать `Options All` в секции `<Directory /www>`, иначе нужно описать параметры каждого каталога отдельно.

В листинге 20.4 приведен пример файла `access.conf`.

Листинг 20.4

```
## access.conf — Apache HTTP server configuration file

<Directory />
Options FollowSymLinks
AllowOverride None
</Directory>

<Directory /www>
Options All
AllowOverride All
order allow, deny
allow from all
</Directory>
```

Ссылки

- Apache-Overview-HOWTO — обзор Web-сервера Apache.
- Building a Secure Red Hat Apache Server HOWTO — настройка безопасного Apache.
- _fastcgi mini-HOWTO — инсталляция Apache Web-сервера с поддержкой Apache+DSO+mod_ssl+mod_perl+php+mod_auth_nds+mod_auth_mysql+modмодулей mod_perl, mod_ssl и php.
- Linux Apache SSL PHP/FI frontpage mini-HOWTO — настройка Web-сервера, поддерживающего динамически изменяемые страницы и данные.
- **http://apache.lexa.ru** — сервер группы разработчиков русского модуля Apache.
- **http://bog.pp.ru/work/apache.html** — Apache: HTTP-сервер. Установка, настройка и русификация.
- **http://www.apache.org** — официальный сайт Apache.
- **http://www.cs.ifmo.ru/education/documentation/rapacheman/index.shtml** — Подстрешный А. Работа с Web-сервером Russian Apache.

Глава 21



FTP

FTP — протокол передачи файлов, возникший очень давно, но до сих пор популярный. В этой главе мы поговорим о самом протоколе, программах, его использующих, а также о настройке сервера FTP.

Протокол FTP

Протокол FTP (File Transfer Protocol, протокол передачи файлов) предназначен для передачи файлов в сети Интернет. Он несколько устарел, частично его функции взял на себя Web-протокол HTTP, но, несмотря на это, протокол FTP, похоже, будет востребован еще долгое время.

FTP, в отличие от большинства других протоколов, для пересылки файла использует два TCP-соединения. Одно соединение для передачи файла, а второе — для управления этим процессом. Порт 20 предназначен для пересылки данных, а порт 21 для управляющего соединения. Протокол FTP может задействовать как TCP-, так и UDP-соединение.

Представление данных

Протокол передачи файлов допускает различные способы представления файлов и управления передачей. Далее приведены критерии, от выбора которых зависит корректность передачи файлов по протоколу FTP.

Тип файла

Протокол должен знать, каков тип передаваемого файла. От этого зависит корректное представление его на компьютере (и в операционной системе) получателя.

- *ASCII-файлы* — текстовый файл передается как NVT ASCII (в формате виртуального сетевого терминала — NVT). При этом требуется, чтобы программа-отправитель конвертировала текстовый файл в NVT ASCII, а программа-получатель производила обратное преобразование. Конец каждой строки передается в виде NVT ASCII-символа возврата каретки (CR), после которого следует перевод строки (LF). Если отправитель текстового файла установит тип файла как бинарный, программы не будут преобразовывать передаваемый файл. Это вызывает проблемы несовместимости между текстовыми файлами DOS/Windows и текстовыми файлами UNIX. В DOS/Windows принято конец

текстовой строки обозначать парой символов возврат каретки/перевод строки (CR/LF), а в UNIX — перевод строки (LF).

- *EBCDIC-файлы* — альтернативный способ передачи текстовых файлов.
- *Бинарные файлы* — данные между FTP-сервером и клиентом передаются как непрерывный поток битов.
- *Локальный тип файлов* — способ передачи бинарных файлов между компьютерами, которые имеют различный размер байта. Число битов в байте определяется отправителем. Обычно не используется.

Управление форматом

Применяется только для передачи ASCII- и EBCDIC-файлов.

- *Nonprint* — файл не содержит информацию вертикального форматирования.
- *Telnet format control* — файл содержит управляющие символы вертикального форматирования Telnet, которые интерпретируются принтером.
- *Fortran carriage control* — первым символом каждой строки служит Fortran-символ управления форматированием.

Структура

Существует несколько способов передачи структуры данных.

- *Структура файла* — пересылаемый файл воспринимается в виде непрерывного потока байтов.
- *Структура записи* — используется только для текстовых файлов.
- *Структура страницы* — каждая страница передается с номером страницы. (Не рекомендуется к применению.)

Режим передачи

Определяет способ передачи файла по соединению данных.

- *Потоковый режим* — файл передается как поток байтов.
- *Блочный режим* — файл передается как последовательность блоков. Блок имеет управляющий заголовок и собственно пересылаемые данные.
- *Режим сжатия* — при передаче заменяют неоднократно встречающиеся повторяющиеся байты на байт и число его повторений.

Как видите, протокол FTP поддерживает много различных представлений данных. Однако в реальной жизни наиболее часто встречается ограниченное подмножество представлений:

- тип файла — ASCII или двоичный;
- управление форматом — только nonprint;
- структура — только структура файла;
- режим передачи — только потоковый режим.

Управляющие команды FTP

Управляющие команды и ответы передаются по управляющему соединению между клиентом и сервером в формате NVT ASCII. В конце каждой строки присутствует пара символов возврат каретки/перевод строки (CR/LF).

В полном наборе насчитывается более 30 команд. В табл. 21.1 приведены наиболее распространенные команды. Полный список команд можно посмотреть в соответствующем RFC.

Таблица 21.1. *Управляющие команды протокола FTP*

Команда	Описание
ABOR	Прервать последнюю команду FTP и любую передачу данных
LIST *список файлов*-	Список файлов или каталогов
PASS *пароль*	Передача пароля пользователя
PORT a, b, c, d, e, f	IP-адрес клиента (a.b.c.d) и порт (e×256 + f)
QUIT	Разорвать соединение
RETR *имя файла*	Получить файл
STOR *имя файла*	Выгрузить файл
SYST	Сервер возвращает тип системы
TYPE *тип*	Указать тип файла: A — для ASCII, I — для бинарного
USER *имя пользователя*	Передача имени пользователя (логин)

Ответы на управляющие FTP-команды

Ответы на управляющие FTP-команды состоят из трехзначного числа в формате ASCII и необязательного текстового сообщения, которое следует за числом.

Каждая из трех цифр в коде ответа имеет собственное значение. Расшифровка первой и второй цифр кода приведена в табл. 21.2.

Таблица 21.2. *Значения первой и второй цифр в коде ответа на управляющие команды*

Ответ	Описание
1*xx*	Положительный предварительный отклик. Действие началось, однако необходимо дождаться еще одного отклика перед отправкой следующей команды
2*xx*	Положительный отклик о завершении. Может быть отправлена новая команда
3*xx*	Положительный промежуточный отклик. Команда принята, однако необходимо отправить еще одну команду
4*xx*	Временный отрицательный отклик о завершении. Требуемое действие не произошло, однако ошибка временная, поэтому команду необходимо повторить позже
5*xx*	Постоянный отрицательный отклик о завершении. Команда не была воспринята, и повторять ее не стоит
*x*0*x*	Синтаксическая ошибка
*x*1*x*	Информация

Таблица 21.2 *(окончание)*

Ответ	Описание
x2x	Соединения. Отклики имеют отношение либо к управляющему соединению, либо к соединению данных
x3x	Аутентификация и бюджет. Отклик имеет отношение к регистрации пользователя в системе или командам, связанным с бюджетом
x4x	Не определено
x5x	Состояние файловой системы

Третья цифра дает уточняющее определение сообщению об ошибке. В табл. 21.3 приведены коды и пояснения.

Обычно каждая FTP-команда генерирует однострочный ответ. Если необходим ответ, состоящий из нескольких строк, то первая строка содержит дефис вместо пробела после трехзначного кода отклика, а последняя строка содержит тот же самый трехзначный код отклика, за которым следует пробел.

Таблица 21.3. *Значения третьей цифры в коде ответа на управляющие команды*

Ответ	Описание
125	Соединение данных уже открыто; начало передачи
200	Команда выполнена
214	Сообщение о помощи
331	Имя пользователя принято, необходимо ввести пароль
425	Невозможно открыть соединение данных
452	Ошибка записи файла
500	Неизвестная команда
502	Нереализованный тип MODE

Управление соединением

Использовать соединение (канал) данных можно тремя способами:

- ❒ отправка файлов от клиента к серверу;
- ❒ отправка файлов от сервера к клиенту;
- ❒ отправка списка файлов или каталогов от сервера к клиенту.

Третий способ необходимо пояснить. FTP-сервер посылает список файлов по соединению данных — при таком функционировании канала данных появляется возможность избежать любых ограничений в строках, накладывающихся на размер списка каталога, и несколько упрощается обмен информацией.

Управляющее соединение остается в активизированном состоянии все время, пока установлено соединение клиент-сервер, но соединение данных может уста-

навливаться и отключаться по необходимости. Рассмотрим, как выбирают номера портов для соединения данных и кто осуществляет активное открытие, а кто пассивное.

Основной режим передачи — потоковый. В этом режиме конец файла обозначает закрытие соединения данных. Следовательно, для передачи каждого файла требуется новое соединение данных. Обычная процедура выглядит следующим образом:

1. Создание соединения данных осуществляется клиентом.
2. Клиент выбирает динамически назначаемый номер порта на компьютере клиента для своего конца соединения данных и осуществляет пассивное открытие с этого порта.
3. Клиент посылает номер порта на сервер по управляющему соединению с использованием команды `PORT`.
4. Сервер принимает номер порта с управляющего соединения и осуществляет активное открытие на этот порт компьютера клиента. Сервер всегда задействует порт 20 для соединения данных.

Сервер всегда осуществляет *активное открытие* соединения данных. Обычно сервер также выполняет *активное закрытие* соединения данных, за исключением тех случаев, когда клиент отправляет файл на сервер в потоковом режиме, который требует, чтобы клиент закрыл соединение.

Если клиент не выдает команду `PORT`, сервер осуществляет активное открытие на тот же самый номер порта, который был у клиента для управляющего соединения.

Программное обеспечение

Для работы по протоколу FTP необходимо две программы: сервер и клиент. Клиентских программ очень много: от простейших, работающих в командной строке, до имеющих весьма развитый графический интерфейс. Любой современный Web-браузер способен выступать в роли FTP-клиента. Поэтому на клиентских программах останавливаться не будем, а перейдем сразу к программному обеспечению сервера FTP.

Сегодня стандартом де-факто для множества дистрибутивов является пакет программ wu-ftp (Washington University at Saint Louis FTP daemon).

Пакет wu-ftp

Программный пакет wu-ftp написан в Вашингтонском университете. Обычно поставляется вместе с дистрибутивом, поэтому установка его не представляет сложности.

Команды

Как уже упоминалось ранее, FTP-серверы имеют свои наборы команд, иногда несколько отличающиеся друг от друга. В табл. 21.4 приведен список команд сервера wu-ftp.

Помимо перечисленных ранее команд, сервер wu-ftp имеет несколько специфичных (табл. 21.5).

Таблица 21.4. *Стандартные команды FTP-сервера wu-ftp*

Команда	Описание
ABOR	Прервать предыдущую команду
APPE	Добавить к файлу
CDUP/XCUP	Подняться на каталог вверх
CWD /XCWD	Поменять текущий каталог
DELE	Удалить файл
HELP	Получить справочную информацию
LIST	Получить список файлов и каталогов в текущем каталоге
MKD /XMKD	Создать каталог
MDTM	Показать время последнего изменения файла
MODE	Задать режим пересылки файла
NLST	Получить список файлов
PASS	Передать пароль пользователя
PASV	Вход в "пассивный" режим передачи
PORT	Задает порт для последующей передачи данных
QUIT	Окончание сеанса
REST	Продолжить прерванную передачу данных
RETR	Получить файл
RMD/XRMD	Удалить каталог
RNFR	Исходное имя переименовываемого файла
RNTO	Новое имя переименовываемого файла
SIZE	Получить размер файла
STAT	Показать состояние сервера
STOR	Сохранить файл
STOU	Сохранить файл с уникальным именем
STRU	Задает структуру передачи
SYST	Вывести тип операционной системы, на которой работает сервер
TYPE	Задать тип передачи
USER	Передать имя пользователя

Таблица 21.5. *Нестандартные команды FTP-сервера wu-ftp*

Команда	Описание
SITE EXEC	Запустить программу на выполнение
SITE GROUP	Сменить группу
SITE GPASS	Передать пароль группы
SITE IDLE	Задать время неактивности пользователя, по истечении которого соединение разрывается
SITE MINFO	Показать список файлов более новых, чем указанная дата. Команда выдает более расширенную информацию, чем NEWER
SITE NEWER	Показать список файлов более новых, чем указанная дата
SITE UMASK	Задать umask для файлов, сохраняемых пользователем на сервере

Конфигурирование сервера

Сервер wu-ftp конфигурируют в два этапа. Первый — компилирование сервера со специфическими для вашего случая свойствами. Этот вариант мы описывать не будем, поскольку для создания простого сервера достаточно пакета rpm, входящего в дистрибутив. Второй этап — редактирование конфигурационных файлов сервера.

Как вы уже знаете, конфигурационные файлы находятся в каталоге /etc. Сервер wu-ftp использует следующие конфигурационные файлы:

- ftpaccess;
- ftpusers;
- ftpgroups;
- ftpservers;
- ftphosts;
- ftpconversion.

Рассмотрим подробно каждый конфигурационный файл.

Файл ftpaccess

Этот конфигурационный файл служит для определения прав доступа к серверу. Здесь определяется, какие и сколько пользователей могут получить доступ к серверу, а также важные элементы настройки безопасности сервера.

Рассмотрим подробно конфигурационные параметры этого файла.

Управление правами доступа:

- `autogroup` *<имя_группы>* *<класс>* `...` — в том случае, если анонимный пользователь является членом указанного класса, то сервер использует заданную группу, что позволяет анонимным пользователям из разных классов получать доступ к различным наборам каталогов;
- `class` *<класс>* `typelist` *<шаблон_адресов>* `...` — позволяет закрепить клиента за указанным классом, исходя из IP-адреса и типа клиента, где:
  - `typelist` — список из ключевых слов, обычно `anonymous`, `guest` и `real` (зарегистрированные на локальном хосте — /etc/passwd), через запятую;
  - *<шаблон_адресов>* — шаблон имени или адреса хоста клиента или адрес: маска или имя файла (имя файла должно начинаться с /, а файл — содержать шаблоны адресов);

- `deny <шаблон_адресов> <файл_с_текстом_сообщения>` — запретить доступ клиентов с указанного адреса с выдачей текста сообщения;
- `guestgroup <имя_группы> ...` — если реальный пользователь является членом указанной группы, то с ним поступают так же, как с анонимным. Имя группы можно заменить ее номером, перед которым нужно поставить знак процента, или интервалом номеров, или звездочкой для всех групп;
- `guestuser <имя_пользователя> ...` — аналогично `guestgroup`, но указано имя реального пользователя;
- `realgroup <имя_группы> ...` — инвертирует действие `guestgroup` и `guestuser`;
- `realuser <имя_пользователя> ...` — инвертирует действие `guestgroup` и `guestuser`;
- `defumask umask [ <класс> ]` — задание `umask`, применяемой при создании файлов;
- `keepalive { yes | no }` — установить `TCP SO_KEEPALIVE`;
- `timeout accept <секунд>` — интервал ожидания входного соединения для передачи данных (PASV);
- `timeout connect <секунд>` — сколько ожидать установления выходного соединения для передачи данных (PORT);
- `timeout data <секунд>` — максимальный период неактивности пользователя при передаче данных;
- `timeout idle <секунд>` — время ожидания следующей команды;
- `timeout maxidle <секунд>` — поскольку клиент имеет возможность установить `idle` самостоятельно, параметр `maxidle` позволяет установить верхний предел для клиента;
- `timeout RFC931 <секунд>` — максимальное время ожидания ответа для протокола `ident`;
- `file-limit [ raw ] { in | out | total } <число> [ <класс> ]` — ограничивает число передаваемых файлов;
- `byte-limit [ raw ] { in | out | total } <число> [ <класс> ]` — ограничивает число передаваемых байтов;
- `limit-time { * | anonymous | guest } <минут>` — ограничение времени сессии. Реальные пользователи не ограничиваются никогда;
- `guestserver [ <имя_серверного_хоста> ]` — гостевой и анонимный доступ предоставляется только к указанному хосту. Имеет смысл, если сервер обслуживает несколько виртуальных доменов;
- `limit <класс> <число> <временной_интервал> <имя_файла_с_сообщением>` — ограничение на число одновременно работающих клиентов из данного класса. Проверка производится только в момент входа. Если к сеансу применимо несколько команд `limit`, то используется первая;
- `noretrieve [ absolute | relative ] { class=<класс> } <имя_файла> ...` — запретить клиенту читать указанные файлы. Если имя начинается с /, то только этот файл, иначе — любой файл с соответствующим именем. Если указан каталог, то любой файл из этого каталога;

- `allowretrieve [ absolute | relative ] { class=<класс> } <имя_файла> ...` — отменить действие директивы `noretrieve`;
- `loginfails <число>` — после указанного числа неудачных попыток зайти на сервер, сделать запись в журнале и разорвать соединение;

Выдача сообщений клиенту:

- `greeting { full | brief | terse | text <строка> }` — определяет, какой текст будет выдаваться в строке приветствия:
  - `full` — имя хоста и версия сервера;
  - `brief` — имя хоста;
  - `terse` — ничего, кроме факта готовности к обслуживанию;
  - `text` — произвольная строка текста;
- `banner <имя_файла>` — текст сообщения, выдаваемого клиенту до ввода имени/пароля;
- `hostname <имя_хоста>` — имя хоста по умолчанию (имя локального хоста);
- `email <адрес>` — адрес администратора;
- `message <имя_файла> { LOGIN | CWD=<имя_каталога> { <класс> } }` — содержимое файла выдается клиенту при входе или смене каталога;
- `readme <имя_файла> { LOGIN | CWD=<имя_каталога> { <класс> } }` — при входе или смене каталога сервер информирует клиента о наличии указанного файла и дате создания/последней модификации.

Журнализация:

- `log commands список_типов` — выводить в журнал все команды клиента, где `список_типов` — список через запятую слов `real`, `guest` и `anonymous`;
- `log transfers список_типов список_направлений` — выводить в журнал пересылки файлов, где `список_типов` — список через запятую слов `real`, `guest` и `anonymous`; `список_направлений` — список через запятую слов `incoming` и `outbound`;
- `log security список_типов` — выводить в журнал нарушения правил безопасности, где `список_типов` — список через запятую слов `real`, `guest` и `anonymous`;
- `log syslog` — перенаправлять сообщения о пересылках в syslog вместо файла xferlog;
- `log syslog+xferlog` — направлять сообщения о пересылках в syslog и файл xferlog.

Виртуальные серверы:

- `daemonaddress <IP-адрес>` — использовать для соединения только указанный адрес;
- `virtual <IP-адрес> { root | banner | logfile } <имя_файла>` — определить соответственно корень файловой системы, файл, содержащий баннер приветствия, и журнал для указанного виртуального сервера;
- `virtual <IP-адрес> { hostname | email } <строка>` — определить имя хоста (отображаемое в приветствии) и адрес администратора для указанного виртуального сервера;
- `virtual <IP-адрес> private` — закрыть анонимный доступ по указанному адресу;

- `virtual <IP-адрес> incmail <email-адрес>` — кого извещать в случае анонимной загрузки файлов;
- `virtual <IP-адрес> mailfrom <email-адрес>` — какой обратный адрес подставлять при рассылке сообщений об анонимной загрузке файлов;
- `defaultserver { deny | allow } <имя_пользователя> ...` — по умолчанию доступ разрешен всем;
- `defaultserver private` — закрыть анонимный доступ;
- `defaultserver incmail <email-адрес>` — кого извещать при анонимной загрузке файлов;
- `defaultserver mailfrom <email-адрес>` — какой обратный адрес подставлять при рассылке сообщений об анонимной загрузке файлов.

Права доступа:

- `{ chmod | delete | overwrite | rename | umask } { yes | no } <список_типов>` — разрешить/запретить пользователям выполнять соответствующее действие. По умолчанию — все разрешено. `<Список_типов>` — список слов через запятую: `anonymous`, `guest`, `real` или `class=<имя_класса>`;
- `passwd-check { none | trivial | rfc822 } ( { enforce | warn } )` — уровень проверки правильности вводимых анонимными пользователями в качестве пароля e-mail-адресов и реакция сервера на ошибку:
  - `none` — никакой проверки;
  - `trivial` — строка должна содержать символ @;
  - `rfc822` — полная проверка согласно стандарту RFC-822;
  - `warn` — если обнаружена ошибка, то выдавать предупреждение;
  - `enforce` — если обнаружена ошибка, то не впускать пользователя;
- `deny-email <email-адрес>` — считать данный адрес неправильным;
- `path-filter <список-типов> <имя_файла_сообщения> <шаблон_допустимых_имен> <шаблон_недопустимых> ...` — когда пользователь из списка типов пытается загрузить файл на сервер, то сервер проверяет имя файла на соответствие регулярному выражению допустимых имен, указанному в шаблоне, и на несоответствие ни одному из регулярных выражений в шаблонах недопустимых имен;
- `upload [ absolute | relative ] [ class=<имя-класса> ]... [ – ] <корень шаблон_каталога> { yes | no } owner group mode [ dirs | nodirs ] [ dir_mode ]` — определяет каталоги, в которые разрешено/запрещено записывать файлы пользователям из указанного класса. Все создаваемые файлы будут иметь соответствующие права доступа и принадлежность;
- `throughput` — позволяет задать скорость передачи определенных файлов на определенные хосты;
- `anonymous-root <корень> [ <класс> ] ...` — определяет корневой каталог (chroot) для анонимных пользователей указанного класса и их домашний каталог;
- `guest-root <корень> [ <интервал-uid> ] ... <корень>` — определяет аргумент chroot для гостевых пользователей и их домашний каталог. Можно задавать отдельные `uid` или интервалы через дефис;
- `deny-uid <интервал> ...` — запрещает доступ к серверу определенным пользователям и может использоваться вместо файла ftpusers;

- deny-gid *<интервал>* ... — запрещает доступ к серверу определенным группам пользователей и может использоваться вместо файла ftpusers;
- allow-uid *<интервал>* ... — разрешает доступ к серверу определенным пользователям и может использоваться вместо файла ftpusers;
- allow-gid *<интервал>* ... — разрешает доступ к серверу определенным группам пользователей и может использоваться вместо файла ftpusers;
- restricted-uid *<интервал>* ... — разрешает реальному или гостевому пользователю доступ вовне его домашнего каталога;
- restricted-gid *<интервал>* ... — разрешает группе пользователей доступ вовне его домашнего каталога;
- unrestricted-uid *<интервал>* ... — запрещает реальному или гостевому пользователю доступ вовне его домашнего каталога;
- unrestricted-gid *<интервал>* ... — запрещает группе пользователей доступ вовне его домашнего каталога;
- site-exec-max-lines *<число>* [*<класс>*] ... — ограничивает число строк, посылаемых командой SITE EXEC;
- dns refuse_mismatch *<файл_с_сообщением>* [override] — выдавать сообщение, если прямой и обратный адреса клиента не совпадают. Если не указано override, то прекращать сеанс;
- dns refuse_no_reverse *<файл_с_сообщением>* [override] — выдавать сообщение, если клиент не имеет обратного адреса. Если не указано override, то прекращать сеанс.

Разное:

- alias *<строка>* *<имя_каталога>* — позволяет переходить в указанный каталог по команде cd *<строка>* из любого каталога;
- cdpath *<имя_каталога>* — добавляет каталог к переменной cdpath, которая является списком поиска для команды cd;
- compress { yes | no } *<шаблон_классов>* ... — разрешает/запрещает компрессию/декомпрессию для классов, подпадающих под шаблон;
- tar { yes | no } *<шаблон_классов>* ... — разрешает/запрещает использование tar для классов, подпадающих под шаблон;
- shutdown *<имя_управляющего_файла>* — файл содержит описание для остановки сервера;
- passive address *<возвращаемый_IP-адрес>* *<CIDR_шаблон>* — если клиент выдает команду PASS, то сервер определяет возвращаемый адрес, исходя из соответствия IP-адреса клиента CIDR-шаблону;
- pasive ports *<CIDR_шаблон>* min max — определяет интервал портов, из которых сервер выбирает порт для прослушивания случайным образом и передает его номер клиенту;
- pasv-allow *<класс>* *<шаблон_адресов>* — позволяет пользователям указанного класса соединяться не только с исходного адреса, но и с заданных шаблоном адресов;
- port-allow *<класс>* *<шаблон_адресов>* — позволяет пользователям данного класса указывать в команде PORT адрес, подходящий под шаблон;

- `lslong <команда> [ <параметры> ]` — определяет команду и параметры для генерации расширенного списка файлов в каталоге;
- `lsshort <команда> [ <параметры> ]` — определяет команду и параметры для генерации списка файлов в каталоге;
- `lsplain <команда> [ <параметры> ]` — определяет, какую команду и параметры использовать для генерации списка файлов в каталоге;
- `incmail <email-адрес>` — определяет, кого извещать в случае анонимной загрузки файлов;
- `mailserver <имя-хоста>` — определяет почтовый сервер для рассылки сообщений об анонимной загрузке файлов;
- `mailfrom <email-адрес>` — какой обратный адрес подставлять при рассылке сообщений об анонимной загрузке файлов.

Файл ftpservers

Этот файл определяет набор файлов конфигурации для каждого виртуального сервера. Каждая строка в данном конфигурационном файле описывает виртуальный сервер и состоит из двух полей:

- имя и IP-адрес виртуального сервера;
- имя каталога, содержащего конфигурационные файлы. Имена файлов фиксированы: ftpaccess, ftpusers, ftpgroups, ftphosts, ftpconversions. Если какой-либо конфигурационный файл отсутствует, то вместо него используется конфигурационный файл основного сервера.

Файл ftpconversions

В этом файле каждая строка описывает возможное преобразование файлов "на лету" и состоит из восьми полей, разделенных двоеточиями:

- удаляемый префикс;
- удаляемый суффикс;
- добавляемый префикс;
- добавляемый суффикс;
- используемая для преобразования внешняя программа и ее параметры;
- типы преобразуемого файла: `T_REG` — обычный файл, `T_ASCII` — текстовый, `T_DIR` — каталог или сочетание перечисленных типов;
- опции: `O_COMPRESS`, `O_UNCOMPRESS`, `O_TAR` или их сочетание;
- комментарий к строке преобразования.

Файл ftpgroups

Этот файл необходим для поддержки функционирования нестандартных команд типа `SITE GROUP` и `SITE GPASS`. В файле ftpgroups находятся строки, состоящие из трех полей, разделенных двоеточием:

- задаваемое клиентом имя группы;
- зашифрованный пароль группы;
- реальное имя группы.

Файл ftphosts

Этот файл предназначен для ограничения доступа к FTP-серверу с определенных хостов. Используются всего две команды:

- `allow <имя_пользователя> <шаблон_IP-адреса> ...` — разрешить доступ;
- `deny <имя_пользователя> <шаблон_IP-адреса> ...` — запретить доступ.

Файл ftpusers

Этот файл служит для запрета доступа к FTP-серверу некоторым реальным пользователям. Обычно применяется для повышения безопасности системы, чтобы исключить доступ пользователей типа `root`, `news` и т. п.

Параметры запуска программ, входящих в пакет

Помимо демона ftpd в пакет входит несколько программ, выполняющих различные действия. Они могут быть полезны для отладки FTP-сервера, получения статистической информации, управления сервером и т. д.

Программа ftpd

Эта программа — сервер FTP. При запуске возможно указание следующих ключей (приведены только основные):

- `-d` — выдавать отладочную информацию;
- `-l` — вести протокол по каждой сессии;
- `-t <число_секунд>` — время бездействия клиента, после которого сервер автоматически разрывает соединение (может быть изменен клиентом);
- `-T <число_секунд>` — время бездействия клиента, после которого сервер автоматически разрывает соединение;
- `-a` — использовать файл ftpaccess;
- `-A` — не использовать ftpaccess;
- `-i` — вести протокол о полученных файлах в файле xferlog;
- `-I` — запретить протокол IDENT;
- `-o` — записывать имена переданных файлов в xferlog;
- `-X` — делать записи о полученных и переданных файлах в файле syslog;
- `-u umask` — маска файла по умолчанию;
- `-w` — записывать заходы в wtmp;
- `-W` — не записывать заходы в wtmp;
- `-s` — самостоятельный запуск без INETD;
- `-S` — самостоятельный запуск, минуя INETD, отсоединиться от терминала;
- `-p <порт>` — управляющий порт, по умолчанию берется FTP-порт из файла /etc/services, при использовании INETD не применяется;
- `-P <порт>` — порт данных, по умолчанию берется значение `ftp-data` из файла /etc/services;
- `-q` — использовать файлы для хранения номеров процессов;
- `-Q` — не использовать файлы для хранения номеров процессов; при указании этого параметра не будет работать ограничение на число пользователей в классе;

- `-r rootdir` — сделать chroot (определение корневого каталога для программы) немедленно после запуска, не дожидаясь ввода имени пользователя; служит для построения защищенной системы.

Программа ftpwho

Эта утилита показывает информацию о каждом подключенном в данный момент клиенте.

Программа ftpcount

Утилита показывает текущее и максимальное число пользователей для каждого класса пользователей.

Программа ftpshut

Утилита предназначена для безаварийного завершения работы FTP-сервера. Представляют интерес следующие ключи запуска:

- `-l <минуты>` — позволяет задать время, за сколько минут до завершения работы сервера запрещать установку новых соединений;
- `-d <минуты>` — задает время, за сколько минут до завершения работы сервера разрывать текущие соединения;
- `<время_завершения>` — время завершения работы сервера. Может быть задано в следующем виде:
  - `now` — немедленно завершить работу сервера;
  - `+минут` — через сколько минут завершить работу сервера;
  - `ЧЧММ` — время завершения работы сервера.

Программа ftprestart

Утилита запускает FTP-сервер, если он был завершен командой `stop`.

Программа ckconfig

Утилита проверяет конфигурацию FTP-сервера. Позволяет выявить случаи явных ошибок в конфигурационных файлах, однако не способна обнаружить логические ошибки.

Формат файла журнала xferlog

Как и положено, FTP-сервер ведет журнал событий. Файл журнала событий называется xferlog и в нем протоколируется любой прием или передача файла. Информация о событии записывается строкой, состоящей из более чем десятка полей. Далее приведено описание полей записи.

1. Название дня недели, например `Sat`.
2. Название месяца.
3. День.
4. Часы:минуты:секунды.
5. Год.

6. Продолжительность передачи в секундах.
7. Имя удаленного хоста.
8. Размер файла в байтах.
9. Имя файла.
10. Тип передачи:
    - `a` — текстовый;
    - `b` — бинарный.
11. Действие над файлом в процессе передачи:
    - `C` — сжат;
    - `U` — разархивирован;
    - `T` — обработан программой tar;
    - _ (символ подчеркивания) — не было произведено никаких действий.
12. Направление передачи:
    - `o` — с сервера;
    - `i` — на сервер.
13. Тип пользователя:
    - `a` — анонимный;
    - `g` — `guest` (гость);
    - `r` — `real` (зарегистрированный).
14. Имя реального пользователя либо идентификационная строка для анонимного или гостевого пользователя.
15. Имя сервиса.
16. Способ аутентификации:
    - `0` — отсутствует;
    - `1` — ident (RFC931).
17. Аутентифицированный идентификатор пользователя. Если аутентификация не использовалась — *.
18. Состояние передачи:
    - `c` — передача была закончена;
    - `i` — не закончена.

Безопасность

Во время конфигурации FTP-сервера очень желательно подумать о его безопасности. Зачастую неправильно сконфигурированный FTP-сервер становится тем слабым местом, через которое осуществляется прорыв безопасности вашей операционной системы.

Чрезвычайно важно, чтобы ваши анонимные и гостевые пользователи FTP не имели доступа к реальному командному процессору. Тогда даже если они по каким-либо причинам покинут окружение FTP, то не смогут выполнить никаких посторонних задач. Для обеспечения этого требования убедитесь, что в файле /etc/passw у пользователей `guest` и `anonymous` в поле, где расположена командная оболочка пользователя, находится что-то типа /dev/null.

В листинге 21.1 приведен список псевдопользователей, которым будет отказано в подключении к FTP-серверу (содержимое файла ftpusers).

Листинг 21.1

```
root
bin
daemon
adm
lp
sync
shutdown
halt
mail
news
uucp
operator
games
nobody
```

Обычно FTP-сервер разрешает загрузку файлов на сервер (upload) всем пользователям. Однако необходимо запретить пользователям загружать свои файлы в некоторые каталоги (а иногда и во все). Для этого в файле ftpaccess необходимо прописать опцию `upload` с ключом `no` и указать каталог, на который налагается запрет.

Иногда желательно запретить пользователям получение с FTP-сервера некоторых каталогов и файлов. Для этого в файле ftpaccess добавляем строку `noretrieve` с каталогом, к которому необходимо запретить доступ.

Ссылки

- RFC959 — RFC, описывающий FTP-протокол.
- **ftp.fni.com/pub/wu-ftpd/guest-howto** — HOWTO по настройке анонимного доступа на FTP-сервер.
- **ftp.wu-ftpd.org** — исходный текст пакета wu-ftp.
- **www.bog.pp.ru/work/ftpd.html** — описание конфигурирования сервера wu-ftp.
- **www.westnet.com/providers/multi-wu-ftpd.txt** — описание настройки виртуальных FTP-серверов.

Глава 22



NNTP. Сервер новостей INN

Один из популярных сервисов, доступных в Интернете, — Usenet (новости, телеконференции, эхо-конференции в FIDO). Это похоже на смесь электронной почты и Web-форума — те же темы и сообщения, как в Web-форуме, только пользователь получает и отправляет сообщения, как электронную почту. В Usenet минимальной единицей информации является статья. Статья помещается в конференцию. Конференций множество — несколько десятков тысяч. Каждая конференция имеет свою тему и иерархическую структуру. Полное имя образуется из имени родительской иерархии, к которому через точку добавляется имя конференции. К примеру — **fido7.ru.linux**, где **fido7** — корень иерархии, показывающий, что группа новостей импортирована из эхо-конференций FIDO, **ru** — русскоязычная, **linux** — конференция посвящена Linux. Прием и передачу статей обеспечивают News-серверы (Usenet-серверы), которые осуществляют синхронизацию (обмен статьями) между собой. Для передачи и приема статей предусмотрен протокол NNTP (Network News Transfer Protocol, сетевой протокол передачи новостей).

Протокол NNTP

Протокол NNTP описан в документе RFC977, а стандарт обмена сообщениями в USENET — в документе RFC1036.

Протокол NNTP предназначен для рассылки, подписки, поиска и доставки новостей на основе TCP по технологии клиент-сервер. Протокол NNTP использует стандартные сообщения, описанные в RFC850. Единица хранения на сервере — статья.

Рассмотрим стандартный сценарий обмена информацией по протоколу NNTP. Пусть есть два или более хоста (один из них выступает в роли клиента, остальные — серверы). Процедура обмена начинается с запроса на получение списка новых групп новостей, для чего выдается команда `NEWGROUPS`. Затем клиент делает запрос командой `NEWNEWS` о наличии новых статей из групп, представляющих интерес. Сервер высылает список статей, клиент обрабатывает список и запрашивает о получении отсутствующих статей. И, наконец, клиент может сообщить серверу, какие новые статьи он получил в последнее время.

NNTP задействует протокол TCP и порт 119. На команды, отправляемые клиентом, предусмотрены текстовые и статусные отклики. Всякая сессия начинается с процедуры установления соединения между клиентом и сервером по инициативе клиента.

Данные могут пересылаться только после цифрового статусного отклика. Они представляют собой последовательности строк, каждая из которых завершается парой символов CR/LF. В конце текста всегда посылается строка, содержащая один символ ".", за которым следует CR/LF. Если исходный текст содержит точку в начале строки, то она перед посылкой дублируется. Статусный отклик представляет собой реакцию сервера на команду, полученную от клиента. Строки статусного отклика начинаются с позиционного трехзначного десятичного кода, в котором закодировано определенное состояние системы.

Первая (слева направо) цифра кода определяет состояние команды: успешно, в процессе выполнения, ошибка (табл. 22.1). Вторая цифра характеризует категорию команды (табл. 22.2). Третья цифра — уточняющее сообщение.

Некоторые коды являются предшественниками последующего текстового отклика.

Таблица 22.1. *Расшифровка первой цифры кода статусного отклика*

Код	Описание
1*xx*	Информационное сообщение
2*xx*	Команда `ok`
3*xx*	Команда корректна, можно продолжать обмен
4*xx*	Команда корректна, но не может быть выполнена по какой-то причине
5*xx*	Команда неприменима, неверна или произошла ошибка

Таблица 22.2. *Расшифровка второй цифры кода статусного отклика*

Код	Описание
*x*0*x*	Соединение, установка режима, прочие сообщения
*x*1*x*	Выбор группы новостей
*x*2*x*	Выбор статьи
*x*3*x*	Функции распределения
*x*4*x*	Отправка адресату
*x*8*x*	Нестандартное расширение
*x*9*x*	Отладочный вывод

Некоторые статусные отклики могут иметь параметры (числа или имена). Число и тип параметров фиксированы для каждого конкретного отклика. Параметры отделяются от кода отклика и друг от друга одиночным пробелом. Все цифровые параметры имеют десятичное представление и могут начинаться с нулей. Все строковые параметры начинаются и завершаются пробелом или символами CR/LF. Любой текст, который не является параметром отклика, должен отделяться от последнего параметра, если таковой имеется, пробелом и завершаться пробелом.

Коды категории *x9x* предназначены для отладочных целей. Так как большинство отладочных откликов можно рассматривать как информационные сообщения, для отладочных выдач зарезервирован диапазон кодов 190–199.

В табл. 22.3 приведен список сообщений общего назначения, которые может послать NNTP-сервер. Эти отклики не привязаны к каким-то конкретным командам и могут быть присланы в результате сбоя или каких-то других необычных обстоятельств.

Таблица 22.3. *Коды сообщений общего назначения*

Код	Описание
100	Поясняющий текст
190–199	Отладочный вывод
200	Сервер готов, отправка разрешена
201	Сервер готов, отправка запрещена
400	Обслуживание прерывается
500	Команда не распознана
501	Синтаксическая ошибка в команде
502	Доступ ограничен или нет разрешения
503	Ошибка в программе, команда не выполнена

Основные команды протокола NNTP

В этом разделе мы рассмотрим основные команды протокола NNTP. Команды можно записывать в любом регистре, поскольку NNTP-сервер регистронезависим. Длина команды не должна превышать 512 байт.

ARTICLE

Существует две формы команды ARTICLE с различными методами спецификации извлекаемой статьи. Когда за командой ARTICLE следует идентификатор сообщения в угловых скобках ("<" и ">"), используется первая форма команды; если же в команде указывается цифровой параметр или нет параметра совсем, реализуется вторая форма.

ARTICLE <message-id>

Команда отображает заголовок, пустую строку и текст заданной статьи. Идентификатор сообщения (message-id) содержится в заголовке статьи. Предполагается, что клиент извлек идентификатор сообщения из списка, полученного командой NEWNEWS. Команда не изменяет указателя текущей статьи.

ARTICLE [nnn]

Отображает заголовок, пустую строку и текст текущей или указанной в цифровом параметре статьи. Опционный параметр nnn представляет собой числовой идентификатор статьи в текущей группе новостей. Он выбирается из диапазона,

который был выдан при выборе группы. Если этого параметра нет, предполагается текущая статья. Эта команда устанавливает указатель текущей статьи, если номер статьи указан корректно.

В ответ сервер выдает номер текущей статьи, строку (идентификатор сообщения) и текст статьи. Присылаемая строка идентификатора сообщения представляет собой заключенную в угловые скобки последовательность символов, которая извлечена из заголовка статьи.

BODY

Команда `BODY` в отличие от `ARTICLE` возвращает только основной текст статьи.

HEAD

Команда `HEAD` в отличие от `ARTICLE` возвращает только строки заголовка статьи.

STAT

Команда `STAT` похожа на команду `ARTICLE` за исключением того, что в ответ не присылается никакого текста. Команда `STAT` служит для установки указателя статьи без пересылки какого-либо текста. Возвращаемый отклик содержит идентификатор сообщения.

GROUP ggg

Команда предназначена для выбора группы сообщений. Обязательный параметр `ggg` — имя группы новостей, которая должна быть выбрана. Отклик на успешный выбор группы возвращает номера первой и последней статьи в группе и оценку общего числа статей в группе.

После выбора группы внутренний указатель статьи устанавливается на первую запись в ней. Если выбрана несуществующая группа, остается в силе выбор предыдущей группы.

HELP

Дает краткое описание команд, которые может воспринять сервер. Отклик на команду имеет текстовую форму и завершается строкой с одиночной точкой в начале.

IHAVE <message-id>

Команда `IHAVE` информирует сервер о том, что клиент владеет статьей с идентификационным кодом `<message-id>`. Если сервер хочет скопировать статью, он выдаст отклик, предлагающий клиенту прислать ее.

Если запрошена передача статьи, клиент должен выслать полный текст статьи, включая заголовок. Сервер в этом случае пришлет отклик, уведомляющий об успехе или неудаче данной операции.

LAST

По команде `LAST` указатель на статью устанавливается на предшествующую запись в текущей группе. Если указатель уже установлен на первую статью, отправляется сообщение об ошибке, а указатель остается неизменным.

LIST

Присылает список доступных групп новостей. Каждой группе соответствует строка в следующем формате:

```
<group> <last> <first> <p>
```

где `<group>` — название группы новостей, `<last>` — номер последней известной статьи в данной группе, `<first>` — номер первой статьи в группе, `<p>` — может быть либо '`y`' либо '`n`', указывая на наличие или отсутствие разрешения на рассылку.

Поля `<first>` и `<last>` числовые. Если код поля `<last>` превосходит код поля `<first>`, в файле данной группы новостей нет ни одной статьи.

NEWGROUPS date time [GMT] [<distributions>]

Список групп новостей, созданных начиная с даты `<date>` и времени `<time>`, будет представлен в том же формате, что и в случае команды `LIST`.

Дата посылается в виде шести цифр в формате ГГММДД, где ГГ — последние две цифры года, ММ — номер месяца (с нулем в начале, если необходимо), ДД — номер дня в заданном месяце. Дополнение для года берется из предположения о ближайшем тысячелетии, так 86 предполагает 1986, 30 — 2030, 99 — 1999, а 00 — 2000 годы.

Время должно характеризоваться шестью цифрами в формате ЧЧММСС, где ЧЧ — часы с начала суток в 24-часовом исчислении, ММ — минуты 00–59, а СС — секунды 00–59. Временная зона определяется сервером, в противном случае появляется символьная комбинация "GMT" (время указанно по Гринвичу).

Заключенный в угловые скобки опциональный параметр `distributions` представляет собой список групп рассылки. Если параметр задан, рассылаемая часть новых групп новостей будет сравниваться с данным списком, и только при совпадении включается в список.

NEWNEWS newsgroups date time [GMT] [<distribution>]

Команда формирует список идентификаторов статей для заданной группы новостей, с датой после указанной. Для каждого идентификатора сообщения в списке выделяется одна строка. Список завершается строкой с одиночным символом точки, за которым следует CR/LF. Дата и время задаются в том же формате, что и для команды `NEWGROUPS`.

Для расширения зоны поиска в имени группы новостей можно использовать символ "*". Программа может подставить вместо звездочки любую комбинацию символов. Если вместо имени группы подставлен символ звездочка, поиск будет проведен по всем группам новостей.

Имя группы должно быть взято из списка доступных групп. Допускается задание нескольких групп (имена разделяются запятыми). После последнего имени группы не должно быть запятой.

NEXT

Команда устанавливает указатель текущей статьи на следующую запись в текущей группе новостей. Если в группе нет больше статей, посылается сообщение об ошибке, а указатель текущей статьи остается неизмененным.

В качестве отклика на команду возвращается номер текущей статьи и идентификатор сообщения. Никаких текстовых сообщений не посылается.

POST

Команда позволяет отправить сообщение в текущую новостную группу.

QUIT

Сервер подтверждает получение команды `QUIT` и затем закрывает канал связи с клиентом. Эта команда предоставляет клиенту корректную возможность сообщить NNTP-серверу, что все операции завершены и сессия закончена.

SLAVE

Команда сообщает серверу, что он связывается не с пользователем, а с обслуживающим сервером (slave). Эта команда позволяет разделить случаи соединения сервера с отдельным пользователем и промежуточными обслуживающими серверами.

Сервер новостей INN

Пакет INN (InterNetNews) — один из старейших пакетов программного обеспечения, предназначенного для создания сервера новостей. Использует стандартный протокол NNTP. Новости хранятся на сервере в дереве каталогов, имена которых формируются из имен телеконференций и повторяют их иерархическую структуру.

Работа пакета INN

Основной процесс — innd — постоянно запущен в системе. Он ожидает и принимает поток статей по протоколу NNTP от серверов новостей, прослушивает порт 119 на наличие входящих соединений, ведет список активных групп, список статей, статьи, базу заголовков статей, пакеты статей для рассылки по серверам новостей, журналы.

При соединении клиентов для чтения новостей программа `innd` передает управление демону nnrpd, который просматривает файл nnrp.access для определения прав доступа к локальной базе статей.

Управляет работой innd (добавляет, удаляет группы, статьи, серверы новостей, изменяет параметры работы) программа ctlinnd.

Удалением старых статей с истекшим сроком хранения занимаются программы expire и expireover, которые удаляют устаревшие файлы, не останавливая innd.

Для автоматического обновления списка новостей используются управляющие сообщения.

Управляющие сообщения

Представляют собой обычные статьи в группе новостей, имеющие заголовок `Control:`. Встретив такую статью, innd обрабатывает записанную в ней команду и сохраняет статью.

Сообщения запоминаются в псевдогруппе `control`. Если создать подгруппу `control.`*`имя_команды`*, то в нее будут помещаться все соответствующие статьи.

Настройка системы INN

Сервер новостей INN по возможностям и сложности настройки весьма напоминает пакет sendmail. Перечислим его конфигурационные файлы, расположенные в каталоге /etc/news.

- /etc/news/actsync.cfg — для конфигурации автоматического изменения списка групп новостей. Обычно добавление новых групп новостей возлагается на администратора системы;
- /etc/news/actsync.ign — для конфигурации автоматического изменения списка групп новостей;
- /etc/news/control.ctl — в этом файле описывается, как обрабатывать управляющие сообщения. Каждая его строка задает действие. Строки состоят из четырех полей, разделенных двоеточием. Первое поле задает команду, к которой применяется действие (можно указать ключевое слово `all`), последнее поле — действие. Строки просматриваются по порядку. Используется последняя подошедшая строка. Возможные действия:
  - `doit`
  - `doifarg`
  - `doit=`*`отдельный_журнал`*
  - `doit=mail`
  - `doit=` (без журнализации)
  - `drop`
  - `log` (запись в журнал — `errlog`)
  - `log=`*`отдельный_журнал`*
  - `mail`

  Для увеличения безопасности и устойчивости системы рекомендуется отказаться от управляющих сообщений, а в файл control.ctl записать единственную строку `all:*:*:drop` — не делать никакой обработки вообще;
- /etc/news/cycbuff.conf — содержит конфигурацию метода хранения CNFS, обычно не используется;
- /etc/news/distrib.pats — файл необходим программам посылки статей, в частности, inews — для определения области распространения статьи. Область распространения определяется по шаблону группы новостей и приоритету. Обычно файл не используется;
- /etc/news/expire.ctl — определяет, через какое время статьи в базе устаревают. Применение файла зависит от метода хранения статей. В частности, метод хранения CNFS самостоятельно удаляет старые статьи. В этом же файле определяется, сколько времени хранить в "истории" информацию об удаленных или отвергнутых статьях.

  В начале файла обязательно должна находиться строка, определяющая срок хранения записи об идентификаторах статей в файле history после удаления тела статьи. Это позволяет отклонить статью, если поставщик новостей вновь предложит ее в определенный промежуток времени. Эта строка имеет следующий формат: `/remember/:`*`время`*, где *`время`* — срок хранения в днях, по истечении которого из системы удаляются идентификаторы старых статей.

Здесь можно определить для различных групп (или для набора иерархий групп) разные сроки хранения статей. Правила хранения статей задаются следующей строкой, состоящей из пяти полей, разделенных двоеточием:

```
<Шаблоны_имени_группы_через_запятую>:<флаг>:min:default:max
```

- Первое поле в строке задает группу или иерархию, удовлетворяющую шаблону;
- Второе поле содержит флаг, который определяет, к какому типу групп применять данное условие:
  - ◊ `A` — все группы;
  - ◊ `M` — только модерируемые;
  - ◊ `U` — только немодерируемые;
  - ◊ `X` — все группы. Если статья была послана в несколько групп и удовлетворяет данному шаблону, то она удалится не только из данной группы, но и из всех остальных групп, в которые была отослана.
- Третье поле задает минимальное число дней хранения. Также можно указать ключевое слово `never`;
- Четвертое поле определяет число дней хранения по умолчанию. Допустимо также ключевое слово `never`;
- Пятое поле определяет максимальное число дней хранения статьи в базе. Возможно ключевое слово `never`;

❐ /etc/news/incoming.conf — в файле определяется, кто может служить для нашего сервера поставщиком новостей. Определяющая строка имеет вид: имя, двоеточие, пробел, значение. Перечислим имена и возможные значения:

- `hostname` — список полных доменных имен хостов или десятичных IP-адресов через запятую;
- `streaming` — `true` или `false`; параметр определяет, разрешен ли потоковый режим;
- `max-connections` — параметр определяет максимальное число параллельных соединений;
- `password` — если сервер новостей требует авторизации, здесь прописывается пароль, обычно не используется;
- `patterns` — шаблон групп, принимаемых с указанного хоста;
- `noresendid` — `true` или `false`; параметр определяет, должен ли сервер новостей посылать ответ `431 RESENDID` в потоковом режиме и `436 Retry later` в непотоковом режиме в ответ на попытку послать статью, которая уже была принята;

❐ /etc/news/inn.conf — файл содержит глобальные параметры сервера новостей и параметры для формирования заголовков статей, создаваемых на этом сервере. Все изменения, сделанные в этом файле, считываются демоном innd только после перезагрузки сервера новостей. Формат строк конфигурационного файла:

```
<имя>: <значение>
```

Далее описываются имена параметров и их значения:

- `fromhost` — параметр необходим при формировании заголовка `From:`, если его нет. Переменная окружения `FROMHOST` переопределяет это значение. По умолчанию, это полное доменное имя локальной машины;

- `moderatormailer` — имя хоста, содержащего псевдонимы для всех модерируемых групп. Рекомендуется использовать файл moderators;
- `organization` — содержимое заголовка `Organization:`, если таковой отсутствует. Если определена переменная окружения `ORGANIZATION`, то она переопределяет это значение;
- `pathhost` — имя локального узла в заголовоке `Path:`. По умолчанию, это полное доменное имя локальной машины;
- `server` — имя NNTP-сервера, на котором должны публиковаться созданные статьи. В том случае, если определена переменная окружения `NNTPSERVER`, то она изменяет это значение;
- `domain` — имя домена, к которому принадлежит локальная машина;
- `overviewmmap` — определяет, будут ли программы expire, nnrpd и makehistory использовать mmap для доступа к файлу overview;
- `storageapi` — способ хранения статей:
  - ◊ `false` — традиционный метод (каждая статья в отдельном файле; каждая группа в каталоге с соответствующим именем);
  - ◊ `true` — хешированные имена (каждая статья хранится в отдельном файле, но имена выбираются исходя из ускорения доступа к файлам);
  - ◊ `cnfs` — кольцевые буферы (все статьи хранятся в кольцевых буферах; есть возможность группировки статей по определенным критериям);
- `maxforks` — максимально возможное количество одновременно запущенных демонов innd;
- `maxartsize` — максимально возможный размер статьи;
- `nicekids` — приоритет процессов, порождаемых программой nnrpd;
- `nicenewnews` — определяет еще более низший приоритет программе nnrpd, обрабатывающей команду `NEWNEWS`;
- `mta` — задает программу для посылки почтой модерируемых статей;
- `mailcmd` — задает программу для посылки отчетов;
- `logcancelcomm` — определяет, сбрасывать ли в стандартную систему журнализации событий (syslog) сообщения о выполнении команды `cancel`;
- `wanttrash` — определяет, сохранять ли статьи для несуществующей группы в группе junk;
- `remembertrash` — определяет, запоминать ли отвергнутые статьи в файле history;
- `linecountfuzz` — определяет, исправлять ли заголовок `Lines`;
- `logartsize` — указывает серверу запоминать в журнале размер статьи;
- `logipaddr` — определяет, записывать ли в журнал событий IP-адрес вместо значения из заголовка `Path`;
- `logsitename` — определяет, сохранять ли имя хоста в журнале полученных статей;
- `overviewname` — задает имя файла для хранения истории сообщений; для каждой группы — свой; по умолчанию имя файла — .overview;
- `extendeddbz` — ускоряет работу с overview за счет увеличения DBZ-файла; требует определенного параметра `storageapi`;

- `nnrpdoverstats` — позволяет сохранять в стандартную систему журнализации событий syslog статистку истории сообщений для nnrpd;
- `storeonxref` — указывает использовать `Xref:` вместо `Newsgroup:` при нестандартном методе хранения;
- `nnrpdcheckart` — благодаря этому значению nnrpd будет не только читать overview, но и проверять реальное наличие статьи;
- `storemsgid` — разрешает хранить идентификатор сообщения (Message-ID);
- `usecontrolchan` — позволяет использовать канал для обработки управляющих статей;
- `refusecybercancel` — указывает серверу отвергать статьи, идентификатор сообщения (Message-ID) которых начинается с `cancel`;
- `activedenable`, `activedupdate`, `activedport` — указывает использовать вспомогательный процесс для буферизации доступа nnrpd к файлу active;
- `pathnews`, `pathbin`, `pathfilter`, `pathcontrol`, `pathdb`, `pathetc`, `pathrun`, `pathlog`, `pathhttp`, `pathtmp`, `pathspool`, `patharticles`, `pathoverview`, `pathoutgoing`, `pathincoming`, `patharchive`, `pathuniover` — указывают серверу пути к различным составляющим сервера новостей: исполняемым файлам, базам сообщений, журналам событий и т. п.;
- `backoff` — задает ограничение на количество статей, посылаемых локальными клиентами с помощью nnrpd;
- `strippostcc` — указывает nnrpd удалять поля `To:`, `Cc:` и `Bcc:`;
- `nnrpperlauth` — указывает серверу аутентифицировать читателя nnrpd с помощью внешней программы на языке программирования скриптов perl;
- `pathalias` — указывает, какую строку добавлять перед `pathhost`;
- `nnrpdposthost`, `nnrpdpostport` — программы nnrpd и rnews будут отправлять статьи на этот сервер;
- `wireformat` — указывает серверу хранить статьи в том же формате, что и при передаче CR LF в конце каждой строки и удвоении точки в начале строки;
- `status` — позволяет регулярно выдавать статистику на стандартную систему журнализации событий syslog;
- `timer` — позволяет регулярно выводить информацию о загруженности сервера на стандартную систему журнализации событий syslog;
- `peertimeout` — определяет, сколько секунд входной канал может оставаться неактивным, прежде чем innd его закроет;
- `chaninacttime`, `chanretrytime` — определяют, сколько секунд канал может быть неактивным, прежде чем innd его закроет;
- `maxconnections` — число одновременных NNTP-соединений;
- `artcutoff` — количество дней для хранения статей (статьи, старше указанного числа дней, удаляются);
- `nntplinklog` — разрешает записывать в журнал сообщения nntplink;
- `nntpactsync` — задает, сколько статей обрабатывать между записями в журнал;
- `badiocount` — определяет, сколько ошибок ввода/вывода допускать, не закрывая канал;
- `pauseretrytime` — задает паузу между проверками канала на неактивность;

- `sourceaddress` — определяет, какой адрес будут иметь исходящие пакеты; если указано `any` — параметр будет выбран операционной системой;
- `port` — задает порт, который будет прослушиваться;
- `localmaxartsize` — максимальный размер статей, посылаемых через nnrpd;
- `mimeversion` — разрешает nnrpd добавлять MIME-заголовки;
- `mimecontenttype` — если добавляются MIME-заголовки, то здесь определяется значение заголовка `Content-Type`;
- `mimeencoding` — если добавляются MIME-заголовки, то здесь определяется значение заголовка `Content-Transfer-Encoding`;
- `spoolfirst` — если задано `true`, то nnrpd помещает статью от клиента в спул, даже не пытаясь обратиться к innd; если `false` — помещает ее в спул только при получении сообщения об ошибке при посылке;
- `articlemmap` — разрешает использовать mmap при доступе к статье в спуле;
- `clienttimeout` — определяет, сколько секунд клиент nnrpd может не проявлять активность до разрыва соединения;
- `innflags` — задает флаги, передаваемые innd при запуске;
- `doinnwatch` — определяет, запускать ли программу innwatch;
- `innwatchsleeptime` — промежуток между проверками innwatch в секундах;
- `controlfailnotice` — определяет, посылать ли администратору письма об ошибках обработки управляющих сообщений;
- `logcycles` — число копий старых журналов, которые нужно сохранять;
- `innwatchpauseload` — средняя загрузка, умноженная на 100, при которой innwatch будет переводить innd в режим ожидания;
- `innwatchhiload` — средняя загрузка, умноженная на 100, при которой innwatch будет переводить innd в режим throttle (отключение сервера);
- `innwatchloload` — средняя загрузка, умноженная на 100, при которой innwatch будет возвращать innd в нормальный режим;
- `innwatchspoolspace` — размер свободного места на устройстве, хранящем articles и overview, в единицах inndf, при достижении которого innwatch переводит innd в режим throttle;
- `innwatchbatchspace` — размер свободного места на устройстве, хранящем исходящие сообщения, в единицах inndf, при достижении которого innwatch переводит innd в режим throttle;
- `innwatchlibspace` — размер свободного места на устройстве, хранящем файлы db-history, active, в единицах inndf, при достижении которого innwatch переводит innd в режим throttle;
- `docnfsstat` — определяет, запускать ли cnfsstat (данный параметр нужен только при использовании метода хранения статей CNFS);

❒ /etc/news/innfeed.conf — конфигурационный файл для программы innfeed. Более подробную информацию следует искать в документации к серверу новостей;

❒ /etc/news/innreport.conf — конфигурационный файл для программы innreport. Более подробную информацию следует искать в документации к серверу новостей;

❒ /etc/news/innwatch.ctl — конфигурационный файл для программы innwatch. Каждая строка определяет одну проверку, состоит из семи полей, разделенных

одним символом, и начинается с того же символа. Разделитель полей един для всей строки и выбирается из списка: восклицательный знак, запятая, двоеточие, @, точка с запятой или вопросительный знак; в зависимости от того, какой знак из перечисленных ранее не встречался внутри полей в этой строке;

- /etc/news/moderators — файл, который хранит имя модерируемой группы и электронный адрес модератора. Когда nnrpd или inews получает статью от клиента и выясняется, что она послана в модерируемую группу, то вместо того, чтобы послать ее innd, он отправляет ее по электронной почте модератору этой группы. В данном файле задаются шаблоны для определения адреса модератора по имени группы. Каждая строка состоит из двух полей, разделенных двоеточием. В первом поле указывается шаблон имени группы. Во втором поле указывается электронный адрес модератора конференции;
- /etc/news/news2mail.cf — конфигурационный файл для программы news2mail;
- /etc/news/newsfeeds — файл содержит информацию о том, какие статьи и каким образом необходимо пересылать на соседние NNTP-узлы. Для каждого узла, с которым вы обмениваетесь новостями, должно быть соответствующее описание в этом файле.

  Каждая строка представляет собой отдельное правило, состоящее из четырех полей, разделенных двоеточиями:

  - `<имя_сайта>/<список_исключений_через_запятую>` — первым сайтом в файле должен быть сайт с именем ME. Если он имеет список шаблонов групп, то этот список добавляется в начало списков остальных сайтов:
    - ◊ `<Имя_сайта>` получателя записывается в журнал; если имя сайта уже встречается в `Path:`, то статья на него не посылается; для локальных имен (программ обработки типа overchan, archive и т. д.) рекомендуется добавлять восклицательный знак в конце, чтобы не пересечься с реальным именем сайта; в качестве имени сайта получателя обычно выбирается то имя, которое этот сайт вставляет в `Path:` при обработке статьи;
    - ◊ `<Список_исключений>` — список имен сайтов через запятую; для каждого имени делается аналогичная проверка — не встречается ли он в `Path:`. Часто встречаются имена генераторов управляющих сообщений: cyberspam, spewcancel, bincancel;
  - `<список_шаблонов_имен_групп_через_запятую>/<список_областей_распределения_через_запятую>`
    - ◊ `<Список_шаблонов>` — определяет, какие группы будут посылаться на сайт получателя. Восклицательный знак в начале шаблона означает отрицание. Наибольший приоритет имеет последнее соответствие. Если вместо `!` использовать `@`, то статья из соответствующей группы не будет посылаться на данный сайт, даже если она отсылается в группу, подлежащую посылке;
    - ◊ область распределения — дополнительно ограничивает список рассылаемых статей: если статья имеет заголовок `Distribution:` и определен список областей распространения для данного сайта получателя, то они должны соответствовать друг другу. Правила записи аналогичны правилам записи шаблонов. Если статья имеет несколько областей распространения, то применяется логическое "ИЛИ";

- `<список_флагов>`
  - ◊ `<size` — статья посылается, если ее размер меньше указанного числа байтов;
  - ◊ `>size` — статья посылается, если ее размер больше указанного числа байтов;
  - ◊ `Ac` — не посылать управляющие сообщения;
  - ◊ `AC` — посылать только управляющие сообщения;
  - ◊ `Ad` — только статьи с заголовком `Distribution:`;
  - ◊ `Ae` — если заголовок статьи `Newsgroups:` содержит только те группы, которые имеются в списке активных групп;
  - ◊ `Ap` — не проверять наличие имени сайта получателя в `Path:` до отсылки сообщения;
  - ◊ `F`*`<имя_файла>`* — задает имя файла для спула;
  - ◊ `G`*`<число>`* — посылать статью, если она послана не более чем в указанное число групп;
  - ◊ `H`*`<число>`* — посылать статью, только если в `Path:` накопилось не более указанного числа хостов;
  - ◊ `I`*`<размер>`* — величина внутреннего буфера, после которого данные начинают сбрасываться в файл;
  - ◊ `Nm` — только модерируемые группы;
  - ◊ `Nu` — только немодерируемые группы;
  - ◊ `P`*`<приоритет>`* — число от 0 до 20, которое будет назначено программе или каналу;
  - ◊ `O`*`<шаблон>`* — требуется наличие заголовка `X-Trace`, и первое поле в нем должно соответствовать шаблону;
  - ◊ `S`*`<размер>`* — если в очереди к данному сайту находится больше указанного размера байтов, то innd переходит в режим спулинга — сбрасывает во временный файл;
  - ◊ `T`*`<тип>`* — способ передачи статей на сайт: `c` — канал, `f` — файл, `l` — только запись в журнал (очень удобно собирать статистику), `p` — программа;
  - ◊ `W`*`<поле>`* — если передача происходит через файл или канал, то здесь указывается, какую информацию туда записывать. Можно использовать несколько флагов. Поля будут записаны в указанном порядке и разделены пробелами. Программы понимают только поле `*` (`b` — размер статьи в байтах, `f` — полное имя файла статьи, `g` — имя первой группы, `h` — hash-ключ Message-ID, `m` — Message-ID, `n` — имя файла статьи относительно спула, `p` — время посылки статьи, `s` — откуда пришла статья, `t` — время получения статьи, `*` — имена всех сайтов, получающих данную статью, `D` — значение заголовка `Distribution:` ("`?`" если не было), `H` — все заголовки, `N` — заголовок `Newsgroups:`, `P` — заголовок `Path:`, `R` — данные для репликации);
- `<параметры>` — формат зависит от способа посылки статей на сайт. Перечислим способы посылки статей:
  - ◊ журнал — делается только запись в журнале /var/log/news/news;
  - ◊ файл — для каждой статьи в файл, определяемый полем `<параметры>`, записывается одна строка. По умолчанию, имя файла — outgoing/*имя_сайта*;

- ◊ программа — для каждой статьи запускается новый экземпляр программы;
- ◊ канал — в поле `<параметры>` задается полное имя программы, которая запускается при старте innd. На каждую статью запущенный процесс получает одну строку на стандартный ввод. Стандартный вывод, ошибки, UID и GID — как для случая программы. Если процесс уже запущен, он перезапускается. Если процесс не удается запустить, то образуется спул в outgoing/*имя_сайта*;
- ◊ exploder — особый подтип канала, кроме обычных статей на него могут быть посланы команды. Команда предваряется восклицательным знаком. Автоматически генерируются следующие команды: `newgroup` *<имя_группы>*, `rmgroup` *<имя_группы>*, `flush`, `flush` *<имя_сайта>*;
- ◊ `funnel` — слияние нескольких потоков в один. Поле *<параметры>* определяет реального получателя;

❒ /etc/news/nnrp.access — файл определяет права доступа к данному NNTP-узлу. Все строки состоят из пяти полей, разделенных двоеточием, и имеют следующий формат:

```
<шаблон_хостов>:<права_доступа>:<имя_пользователя>:<пароль>:↳
    <шаблон_имен_групп>
```

- `<шаблон_хостов>` — задает шаблон для сравнения с хостом клиента и может использовать как имена, так и адреса с сетевой маской;
- `<права_доступа>` — перечень букв, которые определяют права клиента, зашедшего с соответствующего адреса:
  - ◊ `R` — клиент имеет право на чтение;
  - ◊ `P` — клиент имеет право на посылку;
  - ◊ `N` — клиент может использовать команду `NEWNEWS`, несмотря на глобальный запрет;
  - ◊ `L` — клиент может посылать статьи в группы с запретом на локальную посылку;
  - ◊ `<полное_имя_файла>` — формат файла такой же, как и основного, права доступа уточняются, исходя из него;
- `<имя_пользователя>` — пустое, если аутентификация клиента не нужна;
- `<пароль>` — пустой, если аутентификация клиента не нужна;
- `<шаблон_имен_групп>` — список шаблонов имен групп через запятую, к которым клиент должен иметь доступ;

❒ /etc/news/nnrpd.track — файл позволяет nnrpd записывать в журнал доступа определенную строку текста вместо имени или адреса хоста клиента. Состоит из строк вида:

```
<шаблон_имен_или_адресов_хостов>:<строка_идентифицирующая_пользователя>
```

❒ /etc/news/nntpsend.ctl — файл определяет список хостов, на которые nntpsend будет рассылать статьи, если имя хоста не указано явно при запуске. Каждая строка задает отдельный хост и имеет вид:

```
<сайт>:fqdn:размер<параметры>
```

- `<сайт>` — имя, указанное в newsfeeds;
- `fqdn` — полное доменное имя хоста, на который должны быть посланы статьи;

- размер — размер для обрезания пакета заданий, если он станет слишком большим;
- <параметры> — параметры для innxmit;

❒ /etc/news/overview.ctl — файл служит для создания файла истории сообщений overview при использовании новых способов хранения статей;
❒ /etc/news/overview.fmt — файл определяет, какие заголовки будут храниться в файле истории сообщений overview;
❒ /etc/news/passwd.nntp — в этом файле хранятся пароли для доступа к NNTP-серверам;
❒ /etc/news/storage.conf — файл определяет параметры для нестандартных методов хранения статей. Для каждого класса определяется своя структура хранения.

Файл active

Этот файл содержит список групп новостей, которые принимает локальный сервер. Все статьи, опубликованные в группы новостей, которые не указаны в файле active, игнорируются локальным сервером новостей. Строки в этом файле имеют следующий формат:

```
<имя> <старшая_метка> <младшая_метка> <флаги>
```

где:

❒ <имя> — имя группы новостей;
❒ <старшая_метка> — номер самой новой статьи в данной группе новостей на локальном сервере. Это число увеличивается при получении новых статей;
❒ <младшая_метка> — номер самой старой статьи в данной группе новостей на локальном сервере. Это число изменяется в результате удаления старых статей на диске;
❒ <флаги> — это поле определяет один из шести возможных флагов:
  - y — для данной группы новостей разрешена локальная публикация;
  - n — для данной группы новостей не разрешена локальная публикация;
  - m — данная группа модерируемая, и все публикации должны быть одобрены модератором;
  - j — статьи из данной группы новостей не хранятся на локальном сервере, а только передаются через него;
  - x — статьи не могут посылаться в данную группу новостей;
  - =news.group — статьи для данной группы новостей помещаются локально в группу news.group.

Основные операции, которые должен время от времени выполнять администратор, включают в себя добавление новых групп, удаление ненужных групп, изменение флагов текущих групп новостей. Все эти операции должны находить свое отображение в файле active. Существует два основных подхода к выполнению указанных ранее операций с группами новостей.

❒ *Первый подход* — использование соответствующих подкоманд команды ctlinnd — newgroup, rmgroup и changegroup;
❒ *Второй подход* — непосредственное редактирование файла active — удобен для операций с большим количеством групп.

Файлы базы данных и журналы

Список файлов базы данных и их стандартное размещение:

- /var/lib/news/.news.daily;
- /var/lib/news/active;
- /var/lib/news/active.times;
- /var/lib/news/distributions;
- /var/lib/news/history;
- /var/lib/news/newsgroups;
- /var/lib/news/subscriptions.

Список файлов журналов и их стандартное размещение:

- /var/log/news;
- /var/log/news/OLD;
- /var/log/news/news.crit;
- /var/log/news/news.err;
- /var/log/news/news.notice.

Сами статьи находятся в следующих файлах:

- /var/spool/news/archive;
- /var/spool/news/articles;
- /var/spool/news/incoming;
- /var/spool/news/incoming/bad;
- /var/spool/news/innfeed;
- /var/spool/news/outgoing;
- /var/spool/news/overview;
- /var/spool/news/uniover.

Настройка списка получаемых групп новостей

Для настройки получаемых групп новостей необходимо получить список новостей, которые провайдер предоставляет клиентам. Приведем один из способов получения списка:

```
getlist -h newsserver.our.pro > active.provider
```

Созданный этой командой файл active.provider содержит список групп новостей, которые предоставляет провайдер. Выберем из списка те группы, на которые мы действительно хотим подписаться, и пропишем их в нашем файле active. Например, если вы хотите подписаться на конференцию relcom.humor, добавьте в этот файл примерно следующее:

```
relcom.humor 0000000000 0000000001 y
```

Если вы хотите принимать все (или почти все) группы новостей, которые предоставляет провайдер, то файл active можно получить из active.provider, выполнив для него следующие команды (обнуляются два средних поля каждой строки):

```
#!/bin/sh
sed < active.provider > active \
-e 's/^\([^ ]*\) [0-9]* [0-9]* \([^ ]*\)$/\1 0000000000 0000000000 \2/'
```

Нужный файл active готов (он содержит строки для всех групп, которые поддерживает наш сервер), но нужно сообщить и провайдеру о нашем выборе (чтобы он знал, какие группы новостей ему нужно пересылать на наш хост).

Даже если провайдер пропишет нас в своей конфигурации сервера новостей, он не сможет пересылать нам новости по NNTP. Мы должны дать ему разрешение на это. Для этого добавим строчку в файл hosts.nntp:

```
newsserver.our.provider:
```

Здесь нужно заметить, что мы полагаемся на провайдера — знаем, что он будет снабжать нас только теми конференциями, о которых мы его попросили. Если же вы не доверяете своим NNTP-соседям, то можно указать конкретно шаблон конференций, которые вы принимаете на локальный диск от конкретного NNTP-соседа. Например, мы хотим принимать от newsserver.our.badprovider только relcom-группы новостей:

```
newsserver.our.badprovider::relcom.*
```

Отредактируем файл newsfeeds, перечислив всех NNTP-соседей, которых мы хотим снабжать статьями. Не забудем указать в этом файле своего провайдера. Далее приведены два примера этого файла.

- В первом случае мы планируем снабжать статьями хост newsserver.our.provider по NNTP:

```
ME:*, !junk, !control*, !local*/!local::
newsserver.our.provider:*, !junk, !control*, !local*:Tf,
Wnm:newsserver.our.provider
```

- Во втором случае мы хотим снабжать этот же хост по UUCP (имя этой UUCP-системы `provider`), используя программу sendbatch:

```
ME:*, !junk, !control*, !local*/!local::
provider/newsserver.our.provider:*, !junk, !control*, !local*:Tf, Wnb:
```

Затем назначим различные глобальные параметры сервера новостей (имя сервера, имя домена) и параметры, используемые при формировании заголовков статей, публикуемых у нас. Эта информация хранится в файле inn.conf.

Определимся теперь с клиентами нашего сервера новостей (хосты, которые через программу чтения новостей общаются с нашим сервером). Например, мы хотим ограничить пространство пользования ресурсами нашего сервера новостей своей интранет-сетью (192.168.1.0/255.255.255.0) и нашей внешней сетью (домен our.domain), причем пользователям этих сетей мы разрешаем и читать новости, и публиковать их на нашем сервере. При этом нужно помнить о партнерах из домена partner.domain (правда, им нечего делать в наших локальных конференциях). Ну, а для остальных поместим первым правило, запрещающее любой доступ. Для этого добавим в файл nnrp.access строки:

```
*:: -no- : -no- :!*
192.168.1.*:Read Post:::*
*.our.domain:Read Post:::*
*.partner.domain:Read Post:::*, !local*
```

Как только мы начнем получать статьи на локальный диск, необходимо следить за сроком их хранения на диске и удалять старые. К счастью, за нас это будет

делать программа expire, а от нас требуется только дать ей соответствующие указания в файле expire.ctl (ну и, конечно, запускать механизм очистки). В этом файле следует указать сроки хранения:

- идентификаторов статей в файле history (это делается для того, чтобы не принимать заново удаленные статьи);
- самих тел статей.

Приведенный далее пример показывает, что запись об идентификаторе статей хранится в файле history 14 дней после удаления тела этих статей, тела статей из локальных телеконференций хранятся в системе от 5 до 7 дней (по умолчанию — 6), а для всех остальных телеконференций тела хранятся от 3 до 5 дней (по умолчанию — 4 дня).

```
/remember/:14
*:A:3:4:5
local*:A:5:6:7
```

Заметим, что значение по умолчанию (образец `*`) должно фигурировать раньше, чем строки для отдельных групп, поскольку применяется последнее соответствие образцу в первом поле.

Важный шаг после редактирования конфигурационных файлов — проверка корректности сделанных нами изменений. Система INN имеет ряд средств, помогающих нам в решении этой задачи. Укажем некоторые из них.

- Для поиска ошибок в файле newsfeeds можно дать следующую команду:

  ```
  innd -s
  ```

  Например, если вы получили в ответ следующее:

  ```
  Found 1 errors --see syslog
  ```

  значит, командой обнаружена одна ошибка, о которой сообщается через syslog в файлах news.err и news.notice.
- Проверить файл active на наличие неверных строк можно следующей командой:

  ```
  expire -n -x -t
  ```

  Например, если получен ответ

  ```
  /var/news/etc/active: line 5 wrong number of fields
  ```

  значит, вы ошиблись с количеством полей в 5-й строке данного файла (их должно быть 4). Однако это не лучший способ проверки файла active. В частности expire не замечает отсутствие флага для группы новостей (в отличие от inncheck).

Итак, обратим внимание на inncheck — Perl-сценарий, предназначенный для проверки всех рассматриваемых нами конфигурационных файлов. Помимо тестирования файлов на наличие синтаксических ошибок, он может проверять права доступа к файлам и их владельцев. Возвращаясь к примеру, рассмотренному ранее (отсутствие флага в конце строки файла active), inncheck сообщит вам об этой ошибке:

```
/var/news/etc/active:5: ends with whitespace
```

Запущенный без параметров, inncheck проверит синтаксис всех файлов (которые может проверить), с выводом на экран сообщений об ошибках. Если мы укажем опцию `-v` (режим verbose), то inncheck расскажет нам о том, что он просматривает.

Мы можем ограничить работу inncheck проверкой синтаксиса конкретного файла, дав команду `inncheck <имя_файла>`. Для того чтобы выяснить корректность прав доступа к файлам и корректность владельцев и групп файлов, можно дать команду `inncheck -perm`. Аналогичную информацию, да еще и с указанием того, что нужно выполнить, чтобы устранить ошибки, дает команда

```
inncheck -f  -perm
```

Последний шаг настройки — периодически запускать программу отправки статей с нашей машины, программу чистки каталога статей и обобщения log-файлов. Для этого отредактируем таблицу заданий пользователя news для демона cron:

```
crontab -u news -e
```

Ваш редактор (определенный переменной окружения `EDITOR`) откроет файл /var/cron/tabs/news. Ежедневно в 4 часа утра мы будем запускать сценарий news.daily, в функции которого входит обобщение и ротация файлов регистрации, прогон программы expire и др. Далее, в 1-ю и 28-ю минуту каждого часа мы будем запускать программу nntpsend для отправки потоков статей по NNTP нашим соседям:

```
0    4  *  *  *   /usr/news/bin/news.daily > /dev/null 2>&1 &
1, 28 *  *  *  *  /usr/news/bin/nntpsend > /dev/null 2>&1 &
```

Наконец, если мы планируем отправлять потоки новостей по UUCP на UUCP-систему provider, то в 37-ю минуту каждого часа из cron будем вызывать программу sendbatch:

```
37  *  *  *  *  /usr/news/bin/sendbatch -c provider > /dev/null 2>&1 &
```

Журналирование пакета INN

Пакет INN использует стандартный способ — систему журнализации событий syslog. Помимо этого, возможны дополнительные журналы сообщений, в частности:

- news.crit — содержит сообщения о критических ошибках, требующих внимания от администратора сервера новостей;
- news.err — содержит сообщения о фатальных ошибках сервера;
- news.notice — записывает информацию о соединении удаленных NNTP-хостов, активности клиентов, в этом же файле информируют о своей работе программы ctlinnd, innxmit, rnews.

Система INN имеет помимо log-файлов, поддерживаемых системой syslog, встроенные log-файлы — errlog и news (по умолчанию они расположены в каталоге /var/log/news):

- errlog — содержит стандартный вывод и стандартные ошибки любых программ, порождаемых демоном innd;
- news — регистрирует все статьи, поступающие к innd для обработки.

Помимо перечисленных ранее файлов, несколько программ системы INN ведут собственные файлы регистрации (expire.log, send-uucp.log, nntpsend.log и др.).

Программы пакета INN

Поскольку пакет INN очень велик, то в этом разделе приведены лишь некоторые программы, имеющие отношение к пакету, с небольшими комментариями.

- /usr/bin/actived — вспомогательный демон для nnrpd, хранит в памяти проиндексированный файл active;
- /usr/bin/actmerge — выполняет слияние двух файлов active;
- /usr/bin/actsync — утилита для синхронизации, сравнения или слияния файлов active;
- /usr/bin/archive — создает архивную копию части статей;
- /usr/bin/batcher — разбивает на пакеты указанного размера список статей, подготовленных для посылки на хост;
- /usr/bin/controlchan — позволяет передать обработку управляющих сообщений из innd внешней программе;
- /usr/bin/convdate — преобразует формат времени;
- /usr/bin/ctlinnd — интерфейс для управления работающим innd;
- /usr/bin/cvtbatch — преобразует Usenet-пакеты в формат INN;
- /usr/bin/expire — удаляет старые статьи, не прерывая работы innd;
- /usr/bin/expireindex — удаляет старые статьи из списка заголовков статей группы;
- /usr/bin/expireover — удаляет старые статьи из списка статей группы;
- /usr/bin/fastrm — быстро удаляет группы файлов;
- /usr/bin/getlist — получает списки от NNTP-сервера;
- /usr/bin/grephistory — быстро извлекает статьи по ее индексу;
- /usr/bin/inncheck — проверяет конфигурационные файлы;
- /usr/bin/innd — основной сервер, принимающий данные и изменяющий базу данных;
- /usr/bin/inndstart — пусковая программа для innd;
- /usr/bin/innreport — обработка журналов;
- /usr/bin/innstat — выдать состояние сервера;
- /usr/bin/innwatch — мониторинг сервера INN;
- /usr/bin/innxbatch — послать статьи в формате Usenet другому NNTP-серверу;
- /usr/bin/innxmit — пересылка пакета статей другому NNTP-серверу;
- /usr/bin/mailpost — поместить письмо в группу news;
- /usr/bin/makeactive — восстановление файла active по спулу;
- /usr/bin/news.daily — подготовка ежедневного отчета;
- /usr/bin/news2mail — превращение статей в письма;
- /usr/bin/nnrpd — отдельный процесс, предоставляющий клиентам доступ к статьям;
- /usr/bin/nntpsend — оболочка для innxmit;
- /usr/bin/overchan — заполнение данных списка заголовков статей группы;
- /usr/bin/parsecontrol — анализ управляющих сообщений;
- /usr/bin/pgpverify — проверка управляющих сообщений;
- /usr/bin/scanlogs — обработка журналов;
- /usr/bin/send-nntp — подготовка и рассылка пакетов с помощью innxmit;
- /usr/bin/sendxbatches — подготовка и рассылка пакетов с помощью innxbatch;
- /usr/bin/writelog — запись в журнал INN.

Ссылки

- RFC977 — Network News Transfer Protocol — Протокол обмена сетевыми новостями.
- RFC1036 — Standard for interchange of USENET — Стандарт обмена сообщениями в USENET.
- **antonio.mccinet.ru/net/nntp.html** — протокол новостей (NNTP).
- **ief.tup.km.ua/docs/Linux/NAG/nag19.html** — описание NNTP.
- **malik.bishkek.su/doc/UNIX/innd/inn.htm** — Савин Ю. Сервер новостей InterNetNews (INN).
- **www.bog.pp.ru/work/inn.html** — конфигурирование сервера INN.
- **www.isc.org/products/INN** — официальный сайт INN.
- **www.logic.ru/Russian/soft/ligs/node382.html** — система электронных новостей и Usenet.
- **www.mibsoftware.com/userkt/inn/0346.htm** — утилиты для пакета INN.
- **www.switch.ch/switch/netnews/wg/newstools.html** — утилиты для пакета INN.

Глава 23



Proxy-сервер

При подключении к любому провайдеру вам выдаются параметры настройки: адреса сервера DNS, почтового, сервера новостей и proxy-сервера.

Что собой представляет proxy-сервер? Если вы настроите свой браузер для работы через proxy-сервер, то при запросе некоторого документа из Интернета, если некоторое время назад кто-то уже обращался с подобным запросом, вы получите документ незамедлительно, с максимальной скоростью, на которую способно ваше сетевое подключение, потому что вы получите копию документа, взятую из кэша proxy-сервера. Если же proxy-сервер не имеет в своем кэше данного документа, то proxy-сервер запросит удаленный WWW-сервер, хранящий оригинал, и выдаст документ вам, одновременно положив его копию в свой кэш.

Чем больше пользователей пользуются proxy-сервером, тем более существенной становится его помощь. Согласно статистике, число обращений пользователей к одним и тем же документам в сети Интернет приближается к 60%.

Многие proxy-серверы обладают еще одним интересным свойством — они могут обмениваться информацией с соседними proxy-серверами, что существенно ускоряет доступ к данным, хранящимся на удаленных или сильно загруженных серверах.

Proxy-сервер предоставляет следующие возможности:

- централизованный выход в Интернет через один сервер в сети;
- локальное хранение часто просматриваемых документов для увеличения скорости загрузки страниц;
- возможность регулировать пропускную способность канала в зависимости от его нагрузки;
- авторизованный доступ в Интернет;
- возможность обмена данными кэша с соседними proxy-серверами.

Не все данные могут быть корректно получены через proxy-серверы. Это касается, прежде всего, динамически формируемой информации. Однако современные proxy-серверы имеют достаточно много настроек и обладают множеством интеллектуальных алгоритмов, позволяющих, в большинстве случаев, корректно получать самую свежую информацию.

Наиболее распространенный proxy-сервер, доступный под лицензией GNU, — Squid.

Squid

Squid — это высокопроизводительный кэширующий proxy-сервер, поддерживающий протоколы FTP, gopher и HTTP. Squid сохраняет часто запрашиваемые данные в оперативной памяти компьютера, что позволяет резко увеличить производительность proxy-сервера, кэширует DNS-запросы (это свойство интересно тем, у кого нет своего DNS-сервера). Помимо перечисленного, этот сервер поддерживает SSL, расширенный контроль доступа и полную регистрацию запросов.

Одна из ключевых возможностей пакета Squid — использование протокола Internet Cache Protocol (ICP, протокол интернет-кэширования), что позволяет создать иерархию proxy-серверов Squid для дополнительной экономии пропускной способности канала.

Поддерживаемые функции Squid:

- proxy и кэширование HTTP, FTP;
- proxy для SSL;
- иерархия кэшей;
- ICP, HTCP, CARP, Cache digests;
- прозрачный proxy;
- WCCP;
- гибкий контроль доступа;
- HTTP-серверное ускорение;
- SNMP;
- кэширование DNS-запросов;
- возможность ограничения трафика.

Рассмотрим некоторые из этих функций подробнее.

Протокол ICP

Протокол ICP используется в иерархии кэшей для поиска объектов в дереве кэшей Squid-серверов. Если ваш Squid не находит нужного документа, то посылает ICP-запрос другим Squid-серверам, входящим в вашу иерархию proxy-серверов. Эти серверы отвечают ICP ответами `HIT` (попадание) или `MISS` (промах). После получения ответов ваш сервер решает, при помощи какого кэша proxy-сервера получить необходимые ему данные.

Cache digest

Компактная форма представления списка содержимого кэша proxy-сервера. Proxy-серверы могут обмениваться этой информацией с соседями для экономии трафика (нет необходимости делать ICP-запросы). Ключи объектов шифруются по протоколу MD5.

Иерархия кэшей

Иерархия кэшей — это структура кэширующих proxy-серверов, расположенных логически как родительский/дочерний и братский узлы таким образом, что кэши,

ближайшие к интернет-каналу, являются родителями тем proxy-серверам, которые находятся дальше от точки доступа к Интернету. В случае, когда кэш запрашивает объект от родителя, и у того в кэше необходимый объект отсутствует, родительский proxy-сервер получает объект из Интернета, кэширует его и передает дочернему. Таким образом, при помощи иерархии достигается максимальная разгрузка канала.

Кроме родительских/дочерних отношений, Squid поддерживает понятие "братских кэшей" — находящихся на одном уровне иерархии. Каждый proxy-сервер в иерархии независимо ни от кого решает, откуда получать необходимый объект — напрямую из Интернета, от родительского или братского кэша.

Алгоритм получения запрошенного объекта пакетом Squid

Алгоритм таков:

1. Разослать ICP-запросы всем братским кэшам.
2. Дождаться всех ответов, пришедших в течение заданного времени:
   - получив первый ответ `HIT` (попадание), извлечь объект;
   - или взять объект от первого родительского кэша, ответившего `MISS` (зависит от настройки);
   - или получить объект из Интернета при отсутствии объекта в братских кэшах.

Конфигурирование пакета Squid

Основное место конфигурирования пакета Squid — файл /etc/Squid.conf. Размер этого файла достаточно велик, поскольку он содержит множество конфигурируемых параметров, начиная с номера порта для ICP-запросов и заканчивая правилами доступа к информации. Далее перечислены параметры конфигурации Squid-сервера, разбитые на типы. Однако приведенный список не исчерпывающий, поскольку он содержит только наиболее интересные (с нашей точки зрения) параметры конфигурации.

Сетевые параметры

Сетевые параметры proxy-сервера имеют несколько настроек.

- `http_port 3128` — порт для запросов клиентов proxy-сервера;
- `icp_port 3130` — порт для ICP-запросов. В том случае, если не предполагается использовать иерархию proxy-серверов — необходимо указать нулевой порт;
- `htcp_port 4827` — порт для общения с соседями ICP — через TCP-протокол;
- `mcast_groups 239.128.16.128 224.0.1.20` — определяет, к каким multicast-группам (соседи-серверы squid) подсоединяться для получения ICP, если используется multicast;
- `passive_ftp on | off` — выключает режим пассивного FTP (заданный по умолчанию), если Squid находится за брандмауэром.

Соседи

Как уже упоминалось ранее, Squid может обмениваться информацией с другими squid-серверами, которых принято называть соседями.

- Каждый сосед описывается отдельной строкой:
  `cache_peer hostname type proxy-port icp-port options`
  - значения параметра `type`:
    - ◊ `parent` — старший в иерархии;
    - ◊ `sibling` — одного уровня;
  - значения параметра `options`:
    - ◊ `proxy-only` — объекты, взятые с указанного узла, не хранить у себя в кэше;
    - ◊ `weight=число` — указывает приоритет хоста, чем значение больше, тем больше приоритет;
    - ◊ `ttl=число` — время жизни пакета; служит для настройки multicast;
    - ◊ `no-query` — не посылать ICP-запросы;
    - ◊ `default` — самый старший в иерархии;
    - ◊ `round-robin` — определяет родительские кэши, используемые по очереди;
    - ◊ `multicast-responder` — данный сосед является членом multicast-группы;
    - ◊ `no-digest` — не запрашивать от этого соседа cashe digest;
    - ◊ `login=user:password` — определение имени и пароля для случая, если старший в иерархии proxy-сервер требует аутентификации;
    - ◊ `connect-timeout=число` — время ожидания ответа от соседей;
- `cache_peer_domain host domain [domain...]` — ограничить запросы к данному соседу указанным списком доменов;
- `icp_query_timeout milisec` — время ожидания ответа в миллисекундах;
- `mcast_icp_query_timeout milisec` — ожидание ответа на регулярные multicast-опросы;
- `dead_peer_timeout seconds` — время ожидания ответа от соседа, по истечении которого считается, что сосед отсутствует в сети;
- `hierarchy_stoplist` — список строк (через пробел), при встрече которых в URL запрос не будет кэшироваться; по умолчанию cgi-bin;
- `no_cache deny имя-ACL` — определяет список объектов, которые не будут кэшироваться.

Размер кэша

Раздел предназначен для определения параметров кэша — размера, использования, времени хранения информации и т. п.

- `cache_mem 8 MB` — объем оперативной памяти для хранения обрабатываемых объектов;
- `cache_swap_high 95` — при достижении данного уровня заполнения кэша (в процентах) начинается ускоренный процесс очистки кэша от устаревших объектов;
- `cache_swap_low 90` — процесс удаления старых объектов заканчивается, если достигнут данный уровень (в процентах);
- `maximum_object_size 4096 KB` — максимальный размер кэшируемого объекта;

- `minimum_object_size 0 KB` — минимальный размер кэшируемого объекта; файлы меньшего размера не сохраняются;
- `ipcache_size 1024` — размер кэша для IP-адресов;
- `ipcache_high 95` — верхний уровень заполнения IP-кэша для алгоритма удаления старых объектов;
- `ipcache_low 90` — нижний уровень заполнения IP-кэша для алгоритма удаления старых объектов.

Имена и размеры файлов

В этом разделе определяются имена и размеры используемых файлов:

- `cache_dir тип Directory-Name Mbytes Level-1 Level2` — определяет имя, размер и число подкаталогов на первом и втором уровне кэша на диске (каждый кэшируемый объект кладется в отдельный файл, файлы хранятся в двухуровневой иерархии каталогов);
- `cache_access_log /usr/local/squid/logs/access.log` — место хранения журнала обращений к кэшу;
- `cache_log /usr/local/squid/logs/cache.log` — место хранения журнала запусков процессов;
- `cache_store_log /usr/local/squid/logs/store.log` — место хранения журнала записи объектов в дисковый кэш;
- `emulate_httpd_log on|off` — производить ли эмуляцию формата журнала HTTPD;
- `mime_table /usr/local/squid/etc/mime.conf` — таблица типов MIME;
- `log_mime_hdrs off` — в журнал access записываются полученные HTTP-заголовки;
- `useragent_log имя-файла` — в этот файл будут записываться строки User-agent из HTTP-заголовков;
- `debug_options раздел, уровень` — уровень отладки; `ALL` — для всех разделов; по умолчанию `ALL,1`;
- `log_fqdn off` — позволяет определять и записывать в журнал полные доменные имена источника запроса.

Параметры внешних программ

Как и большинство серьезных программ, Squid позволяет воспользоваться внешними программами для выполнения некоторых действий (например, для сбора статистики или обработки трафика).

- `ftp_user email-адрес` — будет подставляться вместо пароля при анонимном доступе к FTP-серверам; по умолчанию — `Squid@`, вызывает проблемы с серверами, которые проверяют синтаксис адреса;
- `cache_dns_program /usr/local/squid/bin/dnsserver` — местоположение программы, кэширующей DNS-запросы;
- `dns_children 5` — число процессов, которые делают DNS lookup (получение по IP-адресу доменного имени и наоборот);

- `dns_nameservers список-IP-адресов` — заменяет список DNS-серверов, определенный в /etc/resolv.conf;
- `redirect_program none` — позволяет подключить программу преобразования URL при каждом запросе;
- `redirect_children 5` — определяет, сколько процессов преобразования URL запускать параллельно;
- `redirect_rewrites_host_header on` — разрешает или запрещает изменение поля `Host:` в заголовке запроса (по умолчанию Squid переписывает поле `Host:` в заголовках преобразованных запросов);
- `redirector_access acl` — какие запросы направлять через редиректор (по умолчанию — все);
- `authenticate_program none` — позволяет производить аутентификацию клиентов, делающих запросы; программа должна в цикле читать строку "имя пароль" и выдавать `OK` или `ERR`; должен быть определен параметр `ACL proxy_auth`;
- `authenticate_children 5` — определяет, сколько параллельных процессов будут заниматься аутентификацией;
- `authenticate_ttl 3600` — определяет, сколько секунд кэшировать результаты работ программы аутентификации;
- `authenticate_ip_ttl число` — необходимо установить 0, чтобы с нескольких адресов не смогли воспользоваться одним именем.

Тонкая настройка кэша

Параметры для тонкой настройки кэша:

- `wais_relay_host localhost` — куда перенаправлять WAIS-запросы;
- `wais_relay_port 8000` — куда перенаправлять WAIS-запросы;
- `request_header_max_size 10KB` — максимальный размер заголовка;
- `request_body_max_size 1 MB` — максимальный размер объекта;
- `refresh_pattern [-i] regex MIN_AGE percent MAX_AGE[options]` — определяет, не устарел ли объект в кэше.

  Имя объекта сравнивается по очереди с регулярными выражениями в строках `refresh_pattern` до первого совпадения, параметры из соответствующей строки используются в алгоритме проверки. По умолчанию регулярные выражения различают прописные/строчные буквы; чтобы игнорировать это различие, применяется ключ `-i`. `MIN_AGE` и `MAX_AGE` — время жизни объекта в минутах. По умолчанию:
  - `refresh_pattern ^ftp: 1440 20% 10080`
  - `refresh_pattern ^gopher: 1440 0% 1440`
  - `refresh_pattern. 0 20% 4320`

  Более подробную информацию смотрите в документации на Squid;
- `reference_age 1 month` — максимальное время хранения неиспользуемого объекта до его удаления;
- `quick_abort_min 16 KB` — если клиент оборвал запрос, а осталось докачать всего `min KB`, то Squid загрузит объект;
- `quick_abort_max 16 KB` — если клиент оборвал запрос и осталось качать больше `max KB`, то Squid прекратит получение объекта;

- `quick_abort_pct число` — если клиент оборвал запрос и уже получено больше чем `число` процентов объекта, то Squid докачает объект;
- `negative_ttl 5 minutes` — время кэширования негативных ответов (например "connection refused ", "404 not found") — число задает их время жизни в кэше;
- `positive_dns_ttl 6 hours` — время кэширования положительных DNS-ответов — число задает их время жизни в кэше;
- `negative_dns_ttl 5 minutes` — время кэширования негативных DNS-ответов — число задает их время жизни в кэше;
- `range_offset_limit 0 KB` — если клиент делает запрос с середины объекта, то:
  - `1` — вынуждает Squid загрузить весь объект в кэш до того, как начать передачу клиенту;
  - `0` — означает, что Squid никогда не будет загружать больше, чем клиент запросил;
  - число, отличное от `1` — начало запроса меньше этого числа — Squid будет грузить весь объект.

Время ожидания

В этом разделе задаются различные временные параметры Squid:

- `connect_timeout 120 seconds` — время ожидания соединения с сервером;
- `siteselect_timeout 4 seconds` — максимальное время на выбор URL;
- `read_timeout 15 minutes` — сколько времени ждать следующего байта от сервера;
- `request_timeout 30 seconds` — сколько ждать запроса после установления соединения;
- `client_lifetime 1 day` — сколько времени разрешать клиенту быть присоединенным к Squid; соединение обрывается, даже если происходит передача данных;
- `half_closed_clients on` — разрешать наполовину закрытые соединения, например чтение есть, а запись уже закрыта;
- `shutdown_lifetime 30 seconds` — сколько времени продолжать обслуживание после получения сигнала `SIGTERM` или `SIGHUP`.

ACL — Access Control List

Этот раздел определяет правила доступа пользователей к группам файлов и хостов. С помощью ACL (Access Control List, список контроля доступа) можно очень гибко настроить доступ к различным сайтам. Список доступа определяют с помощью команды

`acl <имя> <тип> <строка>`

где *<имя>* — имя правила, *<тип>* — тип объекта, *<строка>* — регулярное выражение (шаблон для сравнения), по умолчанию чувствительное к регистру букв.

Значения параметра `тип`:

- `src ip-address/netmask...` — IP-адреса клиентов;
- `src addr1-addr2/netmask...` — диапазон адресов;
- `srcdomain foo.com...` — получение IP-адреса по URL;

- `dstdomainвn foo.com...` — попытка определить имя домена (при неудаче подставляется слово `none`);
- `srcdom_regex [-i] строка...` — получение IP-адреса клиента по URL с использованием регулярных выражений;
- `dstdom_regex [-i] строка...` — попытка определить имя домена, используя регулярные выражения;
- `url_regex [-i] строка` — регулярное выражение для всего URL;
- `urlpath_regex [-i] строка` — регулярное выражение для path-части URL;
- `port порт...` — безопасные порты;
- `browser [-i] regexp` — сопоставляется заголовок User-Agent;
- `maxconn число` — ограничивает число соединений с одного и того же IP.

Права доступа

Права доступа определяются следующими строками:

- `http_access allow|deny [!]aclname...` — кому разрешать доступ к proxy по HTTP;
- `icp_access allow|deny [!]aclname...` — кому разрешать доступ к proxy по ICP;
- `miss_access allow|deny [!]aclname...` — кому разрешить получать ответ MISS;
- `cache_peer_access cache-host allow|deny [!]aclname...` — ограничить запросы к данному соседу;
- `proxy_auth_realm Squid proxy-caching web server` — строка текста, которая будет выдана на экран клиента при запросе имени/пароля доступа к кэшу.

Параметры администрирования

Параметры администрирования определяются следующими строками:

- `cache_mgr email` — почтовый адрес, на который будет послано письмо, если у Squid возникнут проблемы;
- `cache_effective_user nobody` — если запускается Squid от имени root, то заменить UID на указанный;
- `cache_effective_group nogroup` — если запускается Squid от группы root, то заменить GID на указанный;
- `visible_hostname имя-хоста` — это имя будет упоминаться в сообщениях об ошибках;
- `unique_hostname уникальное-имя` — если нескольким кэшам дали одно и то же `visible_hostname`, необходимо определить каждому из них уникальное имя;
- `hostname_aliases имя...` — список синонимов для имени хоста.

Параметры для работы в режиме ускорителя HTTP-сервера

Параметры для работы в режиме ускорителя HTTP-сервера определяются следующими строками:

- `httpd_accel_host hostname` — если нужна поддержка виртуальных хостов, в частности для transparent proxy (прозрачное кэширование), то вместо имени указать `virtual`;
- `httpd_accel_port port` — порт для HTTP-сервера;

- `httpd_accel_with_proxy on|off` — кэширование для ускоряемого сервера;
- `httpd_accel_uses_host_header on|off` — для работы в прозрачном режиме требуется включить, иначе виртуальные серверы не будут правильно кэшироваться.

Разное

Здесь содержатся параметры Squid, не вошедшие в предыдущие разделы:

- `dns_testnames netscape.com internic.net microsoft.com` — список имен хостов, на примере которых проверяется работоспособность DNS;
- `logfile_rotate 10` — данный параметр задает количество старых копий при ротации (раз в сутки архивируется лог-файл и создается новый, пустой);
- `append_domain.vasya.ru` — добавляется к имени хоста, если в нем нет ни одной точки;
- `tcp_recv_bufsize 0 bytes` — `0` означает, что нужно задать размер буфера по умолчанию;
- `err_html_text строка` — подставляется в шаблоны текстов сообщений об ошибках;
- `deny_info err_page_name acl` — запросы, не прошедшие проверку в http_access, проверяются на соответствие ACL, выдается соответствующее сообщение об ошибке из файла page_name;
- `memory_pools on|off` — эта переменная определяет политику работы с захваченной памятью:
  - `on` — однажды захваченная, но не используемая память не отдается обратно в систему;
  - `off` — позволяет освобождать захваченную память;
- `memory_pools_limit байт` — максимальный объем неиспользуемой памяти, которую Squid будет удерживать, если `0` — то удерживать все, что было захвачено;
- `forwarded_for on|off` — если включено, то Squid будет вставлять IP-адрес или имя в заголовки перенаправляемых HTTP-запросов: `X-Forwarded-For: 192.1.2.3`; если выключено, то `X-Forwarded-For: unknown`;
- `log_icp_queries on|off` — записываются ли в журнал ICP-запросы;
- `icp_hit_stale on|off` — возвращать ли ответ `ICP_HIT` для устаревших объектов;
- `cachemgr_passwd password action action...` — задание пароля для действий по администрированию Squid; чтобы запретить действие — поставьте пароль `disable`; чтоб разрешить действие без проверки пароля — поставьте пароль `none`, кроме действий config и shutdown; полную информацию смотрите в документации на Squid;
- `store_avg_object_size 13 KB` — предполагаемый средний размер объекта, используемый для расчетов;
- `store_objects_per_bucket 20` — число объектов на хэш-корзину;
- `client_db on|off` — сбор статистики о клиентах;
- `netdb_low 900` — нижняя граница для базы данных измерения ICMP;
- `netdb_high 1000` — верхняя граница для базы данных измерения ICMP;
- `netdb_ping_period 5 minutes` — минимальное время между посылками ping-пакетов в одну и ту же сеть;
- `query_icmp on|off` — должны ли соседи в ICP-ответы включать ICMP-данные;

- `test_reachability on|off` — при включении ответ `ICP_MISS` будет заменяться на `ICP_MISS_NOFETCH`, если сервер отсутствует в базе данных ICMP или RTT равен нулю;
- `buffered_logs on|off` — при включении запись в журнал буферизуется;
- `always_direct allow|deny [!]aclname...` — запросы, удовлетворяющие данным ACL, не кэшировать, а всегда направлять к первоисточнику;
- `never_direct allow|deny [!]aclname...` — запросы, удовлетворяющие данным ACL, всегда кэшировать;
- `anonymize_headers allow|deny header_name...` — перечень заголовков, которые нуждаются в анонимизации;
- `fake_user_agent none` — если заголовок `User-Agent` фильтруется с помощью анонимизатора, то подставляется эта строка;
- `minimum_retry_timeout 5 seconds` — если сервер имеет несколько IP-адресов, то тайм-аут соединения делится на их количество;
- `maximum_single_addr_tries 3` — сколько раз пытаться соединиться с сервером, имеющим один IP-адрес; если у сервера несколько IP-адресов, то каждый из них будет опробован один раз;
- `snmp_port 3401` — порт, который слушает Squid для SNMP-запросов;
- `snmp_access allow|deny [!]aclname...` — определяет, кто будет допущен к SNMP-порту;
- `offline_mode on|off` — если включить, то Squid будет брать объекты только из кэша и не станет обращаться к первоисточникам;
- `uri_whitespace strip` — что делать с запросами, имеющими пробелы в URI; возможные варианты:
  - `strip` — удалять пробелы;
  - `deny` — сообщать Invalid Request (ошибочный запрос);
  - `allow` — передавать как есть;
  - `encode` — кодировать в соответствии с RFC1738, передавать дальше;
  - `chop` — остаток после первого же пробела отбрасывать;
- `mcast_miss_addr адрес` — по этому multicast-адресу посылается сообщение при каждом "непопадании" в кэш;
- `mcast_miss_port порт` — порт для посылки сообщения;
- `strip_query_terms on` — удалять параметры запроса перед записью в журнал;
- `ignore_unknown_nameservers on` — игнорировать сообщения от DNS-серверов, с которыми Squid не работает.

Пример конфигурации Squid

Как вы уже заметили, опций для конфигурации Squid очень много. Для быстрой настройки proxy-сервера можно воспользоваться приведенными далее параметрами. Конечно, они не идеальны, тонкая настройка поможет вам лучше оптимизировать сервер с точки зрения повышения как производительности, так и безопасности.

Возьмем стандартный файл Squid.conf и отредактируем только нижеприведенные строки:

- `http_port 3128` — номер порта, на котором Squid будет слушать команды от клиентов;

- `hierarchy_stoplist cgi-bin, chat` — слова в URL, при обнаружении которых proxy-сервер будет не кэшировать объекты, а напрямую перенаправлять запрос серверу;
- `cache_mem 16 MB` — сколько оперативной памяти Squid может забрать под свои нужды. Чем больше выделить памяти, тем быстрее будут обрабатываться запросы. Сильно зависит от количества клиентов;
- `maximum_object_size 16384 KB` — максимальный размер объектов, которые будут сохранены в кэше. Размер специфичен для ваших задач и объема жесткого диска;
- `cache_dir /usr/local/Squid/cache 2048 16 256` — указывает proxy-серверу, где сохранять кэшируемые файлы. Под кэш выделяется два гигабайта и создается 16 и 256 каталогов 1-го и 2-го уровня;
- `ftp_user anonymous@vasya.ru` — задает proxy-серверу, под каким паролем регистрироваться на анонимных FTP-серверах;
- `negative_ttl 1 minutes` — время жизни страничек с ошибкой;
- `positive_dns_ttl 6 hours` — время жизни удачного преобразования DNS-имен в IP-адреса;
- `negative_dns_ttl 5 minutes` — время жизни неудачного преобразования DNS-имен в IP-адреса.

Дальнейшие наши действия касаются разграничения прав пользователей.

Прежде всего, необходимо определить ACL (Access Control List, список управления доступом). Сначала закомментируем все строчки в файле Squid.conf, начинающиеся на `acl`. Затем пишем свои правила, например:

- `acl users proxy_auth vasya tolik petya nina` — этой строчкой мы указываем proxy-серверу правило, по которому разрешаем пускать вышеперечисленных пользователей через Squid с помощью авторизующей программы;
- `acl BANNER url_regex banner reklama linkexch banpics us\.yimg\.com [\./]ad[s]?[\./]` — это правило определяет адреса, содержащие рекламу. Интересно для тех, кто хочет отказаться от получения разнообразных баннеров. Позволяет экономить сетевой трафик;
- `http_access deny !users` — эта строка запрещает доступ всем пользователям, кроме перечисленных в группе `users`;
- `http_access deny BANNER` — запрещаем доступ к URL, удовлетворяющим правилу `BANNER` (убираем рекламу);
- `proxy_auth_realm Vasy Pupkina proxy-caching web server` — строка, которая выводится в окно с логином/паролем;
- `cache_mgr vasya@pupkin.ru` — если у клиента возникает проблема, то выводится HTML-страница с сообщением и адресом электронной почты администратора, в нашем случае **vasya@pupkin.ru**;
- `cache_effective_user nobody` — с правами какого пользователя выполняется proxy-сервер;
- `cache_effective_group nogroup` — с правами какой группы выполняется proxy-сервер;
- `client_db on` — параметр разрешает собирать статистику по клиентам.

Поскольку стандартной настройки в такой сфере, как использование канала, места на винчестере, оперативной памяти просто не может быть, более тонкие настройки и ограничения вы должны обдумать и реализовать самостоятельно.

Создание иерархии proxy-серверов

Чтобы разместить кэш в иерархии, нужно воспользоваться директивой `cache_host`.

Приведенной далее частью конфигурационного файла Squid.conf сервер pupkin.ru сконфигурирован так, что его кэш получает данные с одного родительского и с двух братских кэшей:

```
cache_host petya.com parent 3128 3130
cache_host monya.ru sibling 3128 3130
cache_host gesha.ru sibling 3128 3130
```

Директива `cache_host_domain` позволяет задавать для каждого определенного домена или группы доменов как братский, так и родительский кэш. Приведенный в листинге 23.1 пример показывает, что kesha.ru получает данные из доменов .ru, .au, .aq, .fj, .nz, а gesha.ru — из доменов .uk, .de, .fr, .no, .se, .it.

Листинг 23.1

```
cache_host kesha.ru parent 3128 3130
cache_host gesha.ru parent 3128 3130
cache_host uc.cache.nlanr.net sibling 3128 3130
cache_host bo.cache.nlanr.net sibling 3128 3130
cache_host_domain kesha.ru.ru.au.aq.fj.nz
cache_host_domain gesha.ru.uk.de.fr.no.se.it
```

Transparent proxy

Transparent proxy — это таким образом настроенный proxy-сервер, что его использование "прозрачно" для пользователей. Иными словами, пользователям не придется что-либо настраивать в своих браузерах. Для этого необходимо добиться того, чтобы:

1. Все HTTP-запросы пользователей попали на компьютер, где работает ваш HTTP proxy-сервер.
2. Запросы попадали собственно к proxy-серверу.
3. Ваш proxy-сервер их правильно обработал.

Выполнить первый пункт можно разными способами. Самый простой путь — поставить proxy-сервер и маршрутизатор на один сервер, через который проходит весь трафик.

HTTP-запросы пользователей будут попадать к HTTP proxy-серверу, если маршрутизатор (брандмауэр) настроить так, что транзитные пакеты, предназначенные для порта 80, поступают на вход proxy-сервера. Если proxy-сервер должным обра-

зом настроен, он правильно обработает полученные запросы. В Squid.conf добавляются следующие строчки:

```
httpd_accel www.your.domain 80
httpd_accel_with_proxy on
httpd_accel_uses_host_header on
```

Ключи запуска Squid

Помимо конфигурационного файла, поведением программы Squid можно управлять с помощью ключей командной строки. Далее приведены некоторые из них с пояснениями:

- `-a` — указывает порт для входных HTTP-запросов;
- `-d` — выводит отладочную информацию на устройство stderr (обычно — текущая консоль);
- `-f имя_файла_конфигурации` — позволяет использовать альтернативный конфигурационный файл (удобно для отладки сервера);
- `-h` — выводит краткую справку по программе Squid;
- `-k` — этот ключ позволяет посылать Squid следующие управляющие сигналы:
  - `reconfigure` — посылка сигнала `HUP`. Служит для чтения измененного конфигурационного файла;
  - `rotate` — позволяет провести ротацию журналов (сигнал `USR1`);
  - `shutdown` — прервать выполнение программы с корректным завершением (сигнал `TERM`);
  - `interrupt` — немедленно завершить работу программы (сигнал `INT`);
  - `kill` — "убить" приложение (`KILL`);
  - `debug` — начать/закончить полную трассировку (сигнал `USR2`);
  - `check` — проверка (сигнал `ZERO`);
- `-u` — задает порт для входных ICP-запросов;
- `-v` — выводит версию программы;
- `-z` — создает дисковый кэш при первом запуске (Важно!);
- `-D` — предписывает не выполнять DNS-тест при запуске;
- `-F` — восстанавливает после сбоя не в фоновом режиме (ускорение восстановления);
- `-N` — предписывает не становиться фоновым процессом;
- `-V` — включает поддержку виртуальных хостов для режима акселерации;
- `-X` — включает отладку при разборе конфигурационного файла;
- `-Y` — включает быстрое восстановление после сбоев.

Первый раз Squid нужно запускать с ключом `-z`:

```
Squid -z
```

При этом программа создаст дерево кэшей. Этой же командой следует пользоваться в том случае, если вам необходимо очистить кэш proxy-сервера.

Команда `Squid – k rotate` закрывает текущие файлы журналов и создает новые (чистые) файлы.

Файлы журналов Squid

Файл access.log

В файле access.log хранится информация обо всех подключениях к proxy-серверу. Запись добавляется, когда клиент закрывает соединение. Для сервера с большим трафиком файл может за день увеличиваться на десятки мегабайт. К примеру, при трафике в 10 тыс. запросов за сутки объем журнала увеличивается примерно на 2 Мбайт.

Единицей информации о соединении является строка, состоящая из десяти полей. Далее приведено описание полей с пояснениями:

- `timestamp` — время в UNIX-формате (время начиная с 1 января 1970 года в миллисекундах);
- `elapsed` — затраченное время в миллисекундах;
- `client IP address` — IP-адрес клиента, пославшего запрос;
- `type/HTTP` — результат запроса, где `type`:
  - `TCP_HIT` — верная копия объекта нашлась в кэше;
  - `TCP_MISS` — запрашиваемый объект не был в кэше;
  - `TCP_EXPIRED` — объект есть в кэше, но он устарел;
  - `TCP_CLIENT_REFRESH` — клиент запросил принудительное обновление объекта;
  - `TCP_REFRESH_HIT` — объект в кэше был старым, сделан запрос к источнику и источник ответил "объект не изменился";
  - `TCP_REFRESH_MISS` — объект в кэше был старым, сделан запрос к источнику, и тот вернул обновленное содержание;
  - `TCP_IMS_HIT` — клиент выдал запрос, объект оказался в кэше и свежим;
  - `TCP_IMS_MISS` — клиент выдал запрос для просроченного объекта;
  - `TCP_REF_FAIL_HIT` — объект в кэше устарел, но запросить новую копию не удалось;
  - `TCP_SWAPFAIL` — объект должен находиться в кэше, но его не смогли извлечь;
  - `TCP_DENIED` — отказ;
- `size` — число байтов, переданных клиенту;
- `method` — метод передачи информации; `GET`, `HEAD`, `POST` для TCP-запросов или `ICP_QUERY` для UDP-запросов;
- `URL` — адрес запрашиваемого объекта;
- `ident "-"`, если объект недоступен;
- `hierarchy data/Hostname` — результат запросов к братским/родительским кэшам:
  - `PARENT_HIT` — UDP-запрос к родительскому кэшу (parent) вернулся с подтверждением;
  - `PARENT_UDP_HIT_OBJECT` — объект оказался в родительском кэше и поместился в UDP-ответе;
  - `DIRECT` — объект был запрошен с оригинального сервера;
- тип содержимого (MIME-тип/подтип).

Файл store.log

В файле store.log хранится информация обо всех кэшируемых объектах proxy-сервера. Единицей информации о соединении является строка, состоящая из одиннадцати полей. Далее приведены поля с пояснениями:

- `Time` — время в UNIX-формате (время начиная с 1 января 1970 года в миллисекундах);
- `action` — действие:
  - `RELEASE` — удален из кэша;
  - `SWAPOUT` — сохранен на диск;
  - `SWAPIN` — был на диске, загружен в память;
- `HTTP reply code` — код ответа HTTP-сервера;
- `HTTP Date` — дата создания объекта;
- `HTTP Last-Modified` — время последней модификации объекта;
- `HTTP Expires` — срок жизни объекта;
- `HTTP Content-Type` — тип объекта;
- `HTTP Content-Length` — размер объекта;
- реально полученное число байтов. В том случае, если не совпадает с предыдущим полем, объект не сохраняется;
- `HTTP metod` — метод передачи информации (`GET`, `HEAD`, `POST`);
- `Access key` — ключ доступа (обычно URL).

Файл useragent.log

Предназначен для хранения информации о том, с какими пользовательскими агентами (Web-браузерами) работают клиенты. Малоинтересен в практическом плане. Разве что для получения статистики по частоте обращения тех или иных Web-браузеров.

Нестандартные применения

Возможности программы Squid не ограничиваются только функцией proxy-сервера. У нее есть достаточно много других интересных применений. В этом разделе мы рассмотрим только некоторые из них.

Борьба с баннерами

Наверняка вам встречались Web-страницы, на которых нужной информации было от силы на килобайт, а рекламных баннеров (зачастую анимированных) — пять-шесть. Хорошо, когда канал широкий и практически бесплатный. Когда же пользуешься мобильным GPRS соединением, каждый килобайт начинаешь считать. В этом случае можно настроить локальный сервер Squid таким образом, чтобы не закачивались ненужные баннеры. Этого можно добиться несколькими способами.

Вариант 1

Простой. На месте баннеров показываются разорванные картинки или перекрещенные прямоугольники (неполученные файлы).

- Определяем сайты баннерных сетей и создаем для них регулярные выражения.

- Создаем в каталоге /usr/local/Squid/etc следующие файлы:
  - banners_path_regex — содержит по одному регулярному выражению на строку;
  - banners_regex — содержит по одному регулярному выражению на строку;
  - banners_exclusion — это строки, трактуемые в предыдущих файлах как баннеры, но изменять которые не рекомендуется.
- В Squid.conf добавляем правила из листинга 23.2.

Листинг 23.2

```
acl banners_path_regex urlpath_regex
"/usr/local/Squid/etc/banners_path_regex"
acl banners_regex url_regex "/usr/local/Squid/etc/banners_regex"
acl banners_exclusion url_regex "/usr/local/Squid/etc/banners_exclusion"
http_access deny banners_path_regex !banners_exclusion
http_access deny banners_regex !banners_exclusion
```

Вариант 2

Замена рекламных баннеров на свою картинку, которая находится на локальном для proxy-сервера компьютере.

- Определяем сайты баннерных сетей и создаем для них регулярные выражения.
- На своем сервере создаем "заменитель" рекламных картинок — файл mybanner.gif.
- Настраиваем перенаправление (редиректор) в файле Squid.conf — redirect_program /usr/local/Squid/bin/banner.pl.
- Создаем прстой скрипт на Perl — banner.pl (листинг 23.3).

Листинг 23.3

```
#!/usr/bin/perl
$|=1;
while (<>)
{
 s@регулярное-выражение@http://www.myhost.org/mybanner.gif@;
 print;
}
```

Конечно, все это можно сделать более элегантно, однако данный метод работает и позволяет настроить proxy-сервер за пару минут.

Разделение внешнего канала

Часто бывает так, что у вас есть внешний канал — скажем, 1024 Кбит, и несколько групп пользователей с определенным приоритетом. Требуется, чтобы группа 1 имела одну фиксированную ширину наружного канала (скажем, 512 Кбит), а группы 2

и 3 — ширину наружного канала по 256 Кбит. Для решения этой непростой задачи мы также можем воспользоваться Squid.

Немного терминологии:

- пул — набор групп "емкостей" определенного класса;
- группа "емкостей" — часть пула, привязанная к хосту, сети или общая для всех;
- "емкость" ограниченного объема — та, в которую с определенной скоростью вливается внешний трафик и из которой он раздается клиенту.

Определены три класса пулов:

- одна "емкость" на всех из этого класса;
- одна общая "емкость" и 255 отдельных для каждого хоста из сети класса C;
- 255 "емкостей" для каждой сети класса B и отдельная "емкость" для каждого хоста.

Пример конфигурации Squid для трех классов пулов приведен в листинге 23.4.

Листинг 23.4

```
delay_pools 3  # 3 poolы
delay_class 1 1 # 1 pool 1 класса
delay_class 2 1 # 2 pool 1 класса
delay_class 3 3 # 3 pool 3 класса
delay_access 1 allow staff
delay_access 1 deny all
delay_access 2 allow students
delay_access 2 deny all
delay_access 3 allow college
delay_access 3 deny all
delay_parameters 1 512000/512000
delay_parameters 2 512000/512000
delay_parameters 3 512000/512000 256000/512000 12800/256000
```

Строка, определяющая максимальную ширину виртуального канала, имеет вид:

```
delay_parameters pool total_rest/total_max net_rest/net_max ind_rest/ind_max
```

где:

- `pool` — номер пула, для которого определяются каналы;
- `total` — ширина канала на всех;
- `net` — ширина канала на подсеть;
- `ind` — ширина канала на отдельный адрес;
- `rest` — скорость заполнения (байт/с);
- `max` — объем "емкости" (в байтах).

Обработка статистики

В стандартную поставку пакета Squid входят скрипты, написанные на Perl, позволяющие создавать отчеты о работе программы Squid:

- access-extract.pl — получает на стандартный ввод журнал access.log и выдает на стандартный вывод промежуточный результат;
- access-summary.pl — получает на вход результат работы access-extract.pl и делает из него красивый отчет.

Программа Squid Cache and Web Utilities (SARG)

Эта программа обрабатывает журналы Squid и составляет на их базе отчеты.

С помощью нее можно получить следующую информацию:

- число работавших пользователей;
- время их работы;
- трафик по каждому пользователю;
- использование кэша каждым пользователем;
- список Web-серверов, посещаемых пользователем;
- итоговые цифры по трафику и времени.

Существуют также дополнительные отчеты: Top sites и Useragents.

Отчет генерируется за период, интервал которого берется из log-файла Squid. В отчете, кроме зарегистрированных пользователей Squid, отражается информация о попытках незаконного вхождения в сеть и неправильных наборах пароля.

Если не проводится ротация журналов Squid, то SARG генерирует отчеты нарастающим итогом. Отчеты хранятся в стандартном формате HTML-страниц, поэтому их можно просматривать через браузер, копировать и распечатывать. Также отчеты можно генерировать в определенный каталог или получать по почте.

Программа MRTG

Еще одна программа, позволяющая получать при соответствующей настройке отчеты о работе Squid. Вывод осуществляется в виде HTML-страниц.

Программа RRDtool

Программа, позволяющая получать отчеты о работе Squid. Вывод осуществляется в виде HTML-страниц.

Ссылки

- **http://www.Squid-cache.org** — официальный сайт Squid.
- **http://karjagin.narod.ru/solaris/Squid-faq-rus.html** — русский перевод Squid-faq.
- **http://www.nlanr.net/Cache/ICP/ICP-id.txt** — протокол Internet Cache Protocol.
- **http://Squid.org.ua** — зона особого внимания: сайт, полностью посвященный программе Squid.
- **http://linux.webclub.ru/security/proxy/Squid.html** — Иван Паскаль. Настройка Squid.
- **http://www.bog.pp.ru/work/Squid.html** — Bog BOS: Squid (кэширующий proxy для HTTP): установка, настройка и использование.
- **http://www.nitek.ru/~igor/Squid** — борьба с баннерами с помощью Squid.

Глава 24



Синхронизация времени через сеть, настройка временной зоны

Для грамотно настроенной сети предприятия характерен учет всяких нюансов, особенно когда эти "мелочи" таковыми не являются. В частности немаловажна правильная установка системных даты/времени компьютера. При разнообразных "разборах полетов" приводят различные журналы действий пользователя, и при неправильной настройке системного времени грош цена таким данным. Можно, конечно, изменять дату и время вручную, однако через пару-тройку недель вам это наскучит. Особенно неприятно, когда компьютеров несколько десятков, и время у всех должно быть синхронизировано. Для синхронизации системного времени создатели Интернета предусмотрели специальный сервис — сетевой протокол времени (Network Time Protocol, NTP).

Сетевой протокол времени

Протокол NTP предназначен для синхронизации клиента или сервера точного времени с другим сервером точного времени или эталоном времени (радио, атомные часы и тому подобные устройства). Для локальной сети служба NTP способна обеспечить точность до одной миллисекунды, а для распределенной сети (в частности Интернета) достижима точность синхронизации порядка нескольких десятков миллисекунд. Последний стандарт этого протокола предусматривает криптографическую защиту передаваемых данных, одновременное подключение к нескольким серверам точного времени для более точной синхронизации времени и повышения отказоустойчивости системы и многое другое.

Структура сети серверов точного времени многоуровневая. Главные серверы точного времени, напрямую подключенные к источнику эталонного времени, образуют первый уровень, серверы точного времени, присоединенные непосредственно к главным серверам, образуют второй уровень и т. д.

В качестве сетевого протокола используется протокол UDP, порт 123. Для увеличения надежности и точности получаемых данных применяется фильтрация, селекция и комбинация пакетов на принципах максимальной вероятности, а также несколько резервных серверов и путей передачи.

Для передачи и хранения времени предназначено беззнаковое 64-битовое число с фиксированной точкой, которое содержит число секунд в формате UTC. Старшие 32 бита — число секунд, младшие 32 бита — дробная часть секунд. Достижимая точность — 232 пикосекунды. Ноль означает неопределенное время.

Классы обслуживания

Служба точного времени имеет несколько классов обслуживания клиентов:

- `multicast` — для быстрой локальной сети с множеством клиентов, где отсутствует необходимость в высокой точности. Принцип действия — один или более NTP-серверов рассылают широковещательное сообщение, клиенты определяют время, предполагая, что задержка составляет несколько миллисекунд. Сервер не принимает ответных NTP-сообщений;
- `procedure-call` — для получения высокоточного времени. NTP-клиент посылает запрос на сервер точного времени, который обрабатывает запрос и немедленно посылает ответ. Сервер не синхронизируется с клиентом;
- `symmetric` — для серверов точного времени. Представляет собой динамически реконфигурируемую иерархию серверов точного времени. Каждый сервер точного времени синхронизирует своих соседей и синхронизируется своими соседями в соответствии с правилами выбора соседей. Активный режим применяют серверы точного времени низшего уровня с заранее определенными адресами соседей, пассивный режим — серверы точного времени, близкие к первому уровню, и взаимодействующие с соседями с заранее неизвестными адресами.

Обеспечение достоверности данных

Алгоритм функционирования сервера точного времени подразумевает несколько способов для обеспечения достоверности данных.

- Если в течение восьми последовательных интервалов опроса от соседнего сервера точного времени не было сообщений, то этот сервер считается недостижимым.
- Проверка времени осуществляется:
  - если время передачи совпадает со временем предыдущего сообщения — дублированный пакет;
  - если время отправки сообщения не совпадает со временем, содержащимся в пакете, сервер считает, что он получил фальшивый пакет.
- Используется алгоритм защиты от очень старых сообщений.
- Аутентификатор состоит из ключа и контрольной суммы, которая шифруется по алгоритму DES.

Формат NTP-пакета

Пакет NTP включает следующие поля:

- `LI` (leap indicator) — в конце суток должна быть вставлена секунда для синхронизации атомных и астрономических часов;
- `VN` — номер версии протокола;
- `mode` — режим работы сервера точного времени;
- `stratum` — уровень сервера;
- `precision` — точность часов сервера;
- `poll interval` — интервал запросов. Выбирается наименьший интервал из своего и сервера, отвечающего на запросы;

- `synchronization distance` — полный цикл обмена сообщениями до первичного источника;
- `synchronization dispersion` — дисперсия задержек синхронизации;
- `reference clock identifier` — тип источника времени;
- `reference timestamp` — время последнего изменения источника времени;
- `originate timestamp` — время соседа, когда было отправлено последнее NTP-сообщение;
- `receive timestamp` — местное время получения последнего NTP-сообщения;
- `transmit timestamp` — местное время отправки текущего сообщения;
- `authenticator` (96 бит) — ключ и шифрованная контрольная сумма сообщения.

Рекомендуемая конфигурация

Рекомендуемая конфигурация подразумевает наличие трех местных серверов точного времени, соединенных между собой, каждый из которых подключен к двум различным внешним серверам. Клиенты службы точного времени подключаются к каждому местному серверу точного времени.

Стандарты

Стандарты протокола NTP приведены в табл. 24.1.

Таблица 24.1. *Стандарты протокола NTP*

Стандарт	Название	Примечание
RFC1128	Measured performance of the Network Time Protocol in the Internet system (Измерение производительности сетевого протокола времени в Интернете)	
RFC1129	Internet time synchronization: The Network Protocol (Синхронизация времени через Интернет: сетевой протокол времени)	Описывает процесс синхронизации времени
RFC1165	Network Time Protocol (NTP) over the OSI Remote Operations Service (Сетевой протокол времени и взаимодействие с моделью OSI)	
RFC1305	Network Time Protocol (v3) (Сетевой протокол времени, третья версия)	Отменил стандарты RFC1119, RFC1059, RFC958
RFC2030	Simple Network Time Protocol (SNTP) Version 4 for IPv4, IPv6 and OSI (Простой сетевой протокол времени, четвертая версия для IPv4, IPv6 и OSI)	Упрощенный по сравнению с NTP протокол, рекомендуется к использованию там, где нет необходимости для прецизионной синхронизации времени

Сервер xntpd

Для UNIX-платформы, в том числе и Linux, существует сервер точного времени, носящий название xntpd. Этот сервер полностью реализует стандарт RFC1305 и имеет расширенные возможности, которые планируется включить в следующую версию стандарта. Входит в стандартную поставку большинства дистрибутивов Linux. Установка тривиальна. Файл конфигурации — /etc/ntp.conf.

Конфигурация сервера

Поскольку варианты конфигурирования сервера зависят от класса обслуживания, сервер имеет достаточно много настроек, которые в основном содержатся в конфигурационном файле /etc/ntp.conf.

Класс *symmetric*

Этот класс предназначен для конфигурирования сервера точного времени в режиме `symmetric`.

```
peer <address> [key <key>] [version <version>] [prefer] [minpoll <minpoll>]
[maxpoll <maxpoll>]
```

Здесь:

- `<address>` — адрес симметричного сервера;
- `<key>` — 32-битовый ключ для поля аутентификации (по умолчанию отсутствует);
- `prefer` — предпочитать данный сервер при прочих равных условиях;
- `<minpoll>` — минимальный интервал запросов (секунды, 2 в степени `<minpoll>` в диапазоне от 4 (16 с) до 14 (16 384 с), по умолчанию 6 (64 с));
- `<maxpoll>` — максимальный интервал запросов (секунды, 2 в степени `<maxpoll>`, по умолчанию 10 (1024 с)).

Класс *procedure-call*

Этот класс предназначен для конфигурирования сервера точного времени в режиме `procedure-call`.

- `server address [key <key>] [version <version>] [prefer] [mode <mode>];`
- `<address>` — адрес сервера;
- `<key>` — 32-битовый ключ для поля аутентификации (по умолчанию отсутствует);
- `<mode>` — режим.

Класс *multicast*

Предназначен для настройки режима `multicast`. Распространен в локальных сетях.

- `broadcast <address> [key <key>] [version <version>] [ttl <ttl>]`
  - `<address>` — адрес симметричного сервера;
  - `<key>` — 32-битовый ключ для поля аутентификации (по умолчанию отсутствует);
  - `<version>` — версия протокола;
  - `<ttl>` — время жизни пакета;

- `broadcastclient [<address>] <address>` — адрес клиента, получающего информацию;
- `broadcastdelay <секунд>` — позволяет самостоятельно указать задержку в распространении пакета.

Общие параметры

Опишем общие параметры настройки сервера xntpd:

- `driftfile <driftfile>` — определяет файл, в котором хранится и извлекается при запуске сдвиг частоты местных часов;
- `enable/disable auth/monitor/pll/pps/stats` — включить/выключить режим работы:
  - `auth` — с неупомянутыми соседями общаться только в режиме аутентификации;
  - `monitor` — разрешить мониторинг запросов;
  - `pll` — разрешать настраивать частоту местных часов по NTP;
  - `stats` — разрешить сбор статистики;
- `statistics loopstats` — при каждой модификации локальных часов записывает строчку в файл `loopstats`, формат которого имеет вид:
  - номер модифицированного Юлианского дня;
  - секунды с полуночи (UTC);
  - смещение в секундах;
  - смещение частоты в миллионных долях;
  - временная константа алгоритма дисциплинирования часов;
- `statistics peerstats` — каждое общение с соседом записывается в журнал, хранящийся в файле `peerstats`, формат которого имеет вид:
  - номер модифицированного Юлианского дня;
  - секунды с полуночи (UTC);
  - IP-адрес соседа;
  - статус соседа, шестнадцатеричное число;
  - смещение, с;
  - задержка, с;
  - дисперсия, с;
- `statistics clockstats` — каждое сообщение от драйвера локальных часов записывается в журнал, хранящийся в файле `clockstats`;
- `statsdir <имя-каталога-со-статистикой>` — задает имя каталога, в котором будут находиться файлы со статистикой сервера;
- `filegen [file <filename>] [type <typename>] [flag <flagval>] [link | nolink] [enable | disable]` — определяет алгоритм генерации имен файлов, которые включают:
  - префикс — постоянная часть имени файла, задается либо при компиляции, либо специальными командами конфигурации;
  - имя файла — добавляется к префиксу без косой черты, две точки запрещены, может быть изменена ключом `file`;
  - суффикс — генерируется в зависимости от `<typename>`;
  - `none` — обычный файл;

- `pid` — при каждом запуске xntpd создается новый файл (к префиксу и имени файла добавляются точка и номер процесса);
- `day` — каждый день создается новый файл (к префиксу и имени файла добавляется .yyyymmdd);
- `week` — каждую неделю создается новый файл (к префиксу и имени файла добавляется .yyyywww);
- `month` — каждый месяц создается новый файл (к префиксу и имени файла добавляется .yyyymm);
- `year` — каждый год создается новый файл (к префиксу и имени файла добавляется .yyyy);
- `age` — новый файл создается каждые 24 часа (к префиксу и имени файла добавляются `.a` и 8-значное число секунд на момент создания файла от момента запуска xntpd);
- `link`/`nolink` — по умолчанию создается жесткая ссылка от файла без суффикса к текущему элементу набора (это позволяет обратиться к текущему файлу из набора, используя постоянное имя);
- `enable/disable` — разрешает/запрещает запись в соответствующий набор файлов;

❒ `restrict numeric-address [ mask <numeric-mask> ] [flag] ...` — задает ограничение доступа: пакеты сортируются по адресам и маскам, выбирается исходный адрес и последовательно сравнивается, от последнего удачного сравнения берется флаг доступа:

- нет флагов — дать доступ;
- `ignore` — игнорировать все пакеты;
- `noquery` — игнорировать пакеты NTP 6 и 7 (запрос и модификация состояния);
- `nomodify` — игнорировать пакеты NTP 6 и 7 (модификация состояния);
- `notrap` — отказать в обеспечении mode 6 trap-сервиса (удаленная журнализация событий);
- `lowpriotrap` — обслуживать ловушки, но прекращать обслуживание, если более приоритетный клиент потребует этого;
- `noserve` — обслуживать только запросы mode 6 и 7;
- `nopeer` — обслуживать хост, но не синхронизироваться с ним;
- `notrust` — не рассматривать как источник синхронизации;
- `limited` — обслуживать только ограниченное количество клиентов из данной сети;
- `ntpport/non-ntpport` — модификатор алгоритма сравнения адресов (сравнение успешно, если исходный порт равен/не равен 123), алгоритм сортировки ставит эту строку в конец списка;

❒ `clientlimit limit` — для флага `limited` определяет максимальное число обслуживаемых клиентов (по умолчанию 3);

❒ `clientperiod <секунд>` — сколько секунд считать клиента активным и учитывать при определении числа обслуживаемых клиентов;

- `trap host-address [port <port-number>] [interface <interface-address>]` — задать хост и порт, которые будут вести журнал;
- `setvar <variable>` — установка дополнительных переменных;
- `logfile <имя-файла>` — использовать файл `<имя-файла>` для ведения журнала вместо syslog;
- `logconfig <keyword>` — управление числом сообщений, сбрасываемых в журнал. Ключевое слово может быть предварено символами равно (установка маски), минус (удаление класса сообщений), плюс (добавление); ключевое слово образуется слиянием класса сообщений (`clock`, `peer`, `sys`, `sync`) и класса событий (`info`, `event`, `statistics`, `status`); в качестве суффикса или префикса возможно слово `all`.

Обеспечение безопасности сервера

Если сервер точного времени используется только внутренней локальной сетью, желательно закрыть порт 123 для доступа извне, чтобы избежать возможной атаки типа denial of service (отказ в обслуживании), поскольку это грозит неправильным функционированием сервера. Помимо этого необходимо шифрование трафика.

Программы и утилиты, относящиеся к службе точного времени

В этом разделе приведены несколько утилит синхронизации с серверами точного времени для различных операционных систем. Таких программ много, особенно для Windows, поэтому приведены только несколько примеров.

ntpdate

Эта утилита позволяет установить время на компьютере с помощью списка NTP-серверов.

Ключи:

- `-B` — только плавный сдвиг, даже если смещение больше 128 мс;
- `-b` — всегда использовать settimeofday;
- `-d` — отладка;
- `-p <число>` — число запросов к каждому серверу (от 1 до 8, по умолчанию 4);
- `-q` — только запрос времени;
- `-s` — syslog вместо stdout;
- `-t <timeout>` — время ожидания ответа (по умолчанию 1 с);
- `-u` — задействовать непривилегированный порт.

ntpq

Утилита для получения состояния NTP-сервера и его изменения (использует NTP mode 6).

ntptrace

Утилита для поиска серверов первого уровня.

Ключи:

- `-r <число>` — число запросов (по умолчанию 5);
- `-t <секунд>` — время ожидания ответа (по умолчанию 2).

xntpd

Собственно демон точного времени. Параметры при запуске:

```
xntpd [-aAbdm ] [-c <config-file>] [-f <drift-file>] [-k <key-file>]
   [-l <log-file>] [-p <pid-file>] [-r <broadcast-delay>] [-s <stats-dir]
   [-t <key>] [-v <variable>] [-V <variable>]
```

Здесь:

- `-a` — разрешить аутентификацию;
- `-A` — запретить аутентификацию;
- `-b` — широковещательные сообщения;
- `-c <config-file>` — конфигурационный файл (по умолчанию /etc/ntp.conf);
- `-d` — отладка;
- `-f <drift-file>` — файл, хранящий смещение часов (по умолчанию /etc/ntp.drift);
- `-k <key-file>` — файл ключей (по умолчанию /etc/ntp.keys);
- `-l <log-file>` — файл протокола (по умолчанию syslog).

xntpdc

Утилита для запроса состояния NTP-сервера и его изменения. Применяется только для Xntpd-серверов. Использует NTP mode 7.

Публичные NTP-серверы

Список публичных серверов точного времени можно найти в Интернете. В любом случае вам придется протестировать серверы из этого списка, чтобы определить задержки и качество соединения. Попробуйте сначала получить список серверов точного времени вашего провайдера (провайдеров).

Клиентские программы для синхронизации времени

Сам по себе сервер точного времени бесполезен, если у пользователей отсутствует программное обеспечение для синхронизации даты/времени. Сейчас практически для всех операционных систем есть программы получения времени с серверов NTP. Некоторые из них приведены далее.

UNIX/Linux

Для этих операционных систем можно на компьютере установить сервер xntpd и настроить его для получения точного времени. У такого решения есть как достоинства, так и недостатки. Положительным моментом является то, что мы можем максимально точно синхронизировать время и построить отказоустойчивую конфигурацию. Отрицательный момент — довольно сложное конфигурирование сервера и относительно большой объем занимаемой оперативной памяти компьютера.

Более простой вариант — воспользоваться утилитой ntpdate. Она небольшая по размерам и простая в конфигурировании. С ее помощью можно получить весьма точное время — расхождение порядка 100 мс. Для синхронизации времени достаточно выполнить команду

```
ntpdate -B <ntp> <ntp2> <ntp3>
```

где *<ntp>*, *<ntp2>*, *<ntp3>* — адреса серверов точного времени. Добавив эту строчку в таблицу заданий crontab, вы всегда будете иметь на компьютере точное время.

Apple

Для старых компьютеров фирмы Apple есть клиент NTP, называющийся macntp. В новых компьютерах с операционной системой Mac Os X клиент встроен в систему.

Windows

Для операционной системы Windows существует несколько десятков клиентов службы точного времени. В частности программы AboutTime, AnalogX и Atomic TimeSync, которые можно получить по адресу **http://www.listsoft.ru/programs/536/** и **www.analogx.com/contents/download/network/ats.htm**. А можно воспользоваться программой Dimension 4 — **www.thinkman.com/~thinkman**. Все это справедливо для старых версий Windows. Начиная с Windows XP, клиент точного времени уже встроен в систему.

Ссылки

- **www.bog.pp.ru/work/ntp.html** — Богомолов С. Bog BOS: Network Time Protocol.
- **www.bog.pp.ru/work/xntpd.html** — Богомолов С. Bog BOS: xntpd (UNIX-сервер NTP — Network Time Protocol).
- **www.ntp.org** — страница, посвященная xntp.
- **cisco.opennet.ru/docs/RUS/lasg/time.html** — сетевые сервисы: NTP.
- **www.psn.ru/net/servis/ntp.shtml** — статья "Как пользоваться службой NTP?".
- **www.tomsknet.ru/ftp/docs/rfc/rfc1305.txt** — Network Time Protocol (Version 3) Specification, Implementation and Analysis (RFC1305).

Глава 25



Сетевая информационная система NIS (NIS+) и ее конфигурирование. LDAP

В этой главе описаны две службы NIS (NIS+) и LDAP, предназначенные для оптимизации настроек "сетевых" пользователей (т. е. тех пользователей, которые могут часто менять свое местоположение и работать на разных компьютерах), а также для облегчения жизни сетевых администраторов. Как обычно, применяется технология клиент-сервер, поэтому будет рассмотрена настройка как клиентской части, так и серверной.

NIS

NIS (Network Information Service, служба сетевой информации) — служба, предоставляющая данные для компьютеров, подключенных к сети. Основная информация, которую предоставляет NIS:

- имена для входа в систему/пароли/домашние каталоги (/etc/passwd);
- сведения о группах (/etc/group).

Первоначально NIS была разработана фирмой Sun Microsystems, Inc. и носила название Yellow Pages. Однако из-за того, что Yellow Pages является торговой маркой, принадлежащей British Telecom, пришлось переименовать этот протокол.

Как работает NIS

Идеальная структура серверной части NIS — наличие нескольких серверов, среди которых один главный (мастер-сервер), а все остальные подчиненные серверы (для определенного домена NIS). Вырожденный случай — один NIS-сервер.

Подчиненные серверы хранят только копии баз данных NIS и получают эти копии от мастер-сервера в тот момент, когда в базы данных на мастер-сервере вносятся изменения. Такая структура подразумевает, что в случае, когда NIS-сервер станет недоступен или будет перегружен, клиент NIS, подключенный к этому серверу, будет искать дублирующий NIS-сервер (рабочий или менее загруженный).

При изменении баз данных мастер-сервера подчиненные серверы будут извещены об этом посредством программы yppush и автоматически получат необходимые изменения. Клиенты NIS не нуждаются в синхронизации, поскольку не кэшируют данные.

Программа-сервер ypserv

Если вы используете ваш сервер как мастер, определите, какие файлы нужны вам для доступа через NIS, а затем добавьте или удалите соответствующие записи в правиле `all` в /var/yp/Makefile.

Теперь отредактируйте /var/yp/securenets и /etc/ypserv.conf. Убедитесь, что portmapper (portmap(8)) запущен, и запустите сервер ypserv. Команда

```
rpcinfo -u localhost ypserv
```

должна выдать примерно следующее

```
 program 100004 version 1 ready and waiting
 program 100004 version 2 ready and waiting
```

Строка `version 1` может отсутствовать в зависимости от версии ypserv и ваших настроек.

Далее необходимо сгенерировать базу данных NIS (YP). На мастер-сервере запустите команду

```
ypinit -m
```

На подчиненном сервере убедитесь, что `ypwhich -m` работает. Это означает, что ваш подчиненный сервер должен быть настроен как клиент NIS перед запуском команды

```
ypinit -s masterhost
```

для установки этого узла как подчиненного сервера NIS.

Для обновления базы данных NIS запустите `make` в каталоге /var/yp на вашем мастер-сервере. Это приведет к ее обновлению и загрузке на подчиненные серверы, если ее исходный файл имеет более свежую дату.

NIS+

NIS+ (Network Information Service Plus, Расширенная служба сетевой информации) — этот протокол предназначен для замены NIS и обратно совместим с ним. Однако, как и NIS, NIS+ является коммерческим и бесплатной реализации для Linux не имеет. Под Linux реализован только клиент NIS+. Поэтому NIS+ в среде Linux не получил большого распространения и вместо него используется LDAP.

Как работает NIS+

NIS+ — это новая версия службы имен сетевой информации от Sun. Самое большое различие между NIS и NIS+ состоит в том, что NIS+ имеет поддержку шифрования данных и авторизацию через безопасный RPC.

Модель имен в NIS+ основывается на структуре дерева. Каждый узел в дереве соответствует одному объекту NIS+ из шести типов: каталог, запись, группа, ссылка, таблица и личное.

Каталог NIS+, формирующий главное пространство имен NIS+, называется корневым каталогом. Имеется два специальных каталога NIS+: org_dir и groups_dir. Каталог org_dir содержит все административные таблицы, такие как passwd (пароли),

hosts (узлы) и mail_aliases (почтовые псевдонимы). Каталог groups_dir содержит объекты групп NIS+, которые используются для управления доступом. Совокупность org_dir, groups_dir и их родительского каталога называется *доменом* NIS+.

LDAP

LDAP (Lightweight Directory Access Protocol, облегченный протокол службы каталогов) основан на клиент-серверной модели.

Один или более серверов LDAP содержат данные, составляющие дерево каталога LDAP, или базу данных LDAP. Клиент LDAP подключается к LDAP-серверу и задает ему вопросы. Сервер выдает ответ или указатель места, где клиент может получить более подробную информацию (обычно другой LDAP-сервер). Не имеет значения, к какому LDAP-серверу подключается клиент, он видит один и тот же каталог.

Протокол LDAPv3 описывается в RFC2251-2256, 2829-2831.

Далее рассматривается реализация LDAP, известная как OpenLDAP. Основные особенности OpenLDAP:

- поддержка LDAPv2 и LDAPv3;
- поддержка взаимодействия с существующими клиентами;
- поддержка IPv4 и IPv6;
- Strong Authentication (SASL — безопасная система авторизации) (RFC2829);
- Start TLS (RFC2830);
- Language Tags (языковые метки) (RFC2596);
- служба расположения, основанная на DNS (RFC2247);
- усовершенствованный автономный сервер;
- Named References/ManageDsaIT (справочники наименований);
- усовершенствованная подсистема контроля доступа;
- Threads pool, пул нитей;
- поддержка приоритетов нитей;
- поддержка множества слушателей;
- LDIFv1 (RFC2849);
- усовершенствованное определение платформы/подсистемы.

Установка LDAP-сервера

Для установки LDAP-сервера помимо самого пакета OpenLDAP необходимо установить еще несколько пакетов.

- Библиотеки OpenSSL TLS — обычно входят в состав системы или являются опциональным программным компонентом.
- Kerberos — клиенты и серверы OpenLDAP поддерживают службы аутентификации на основе Kerberos. В частности, OpenLDAP поддерживает механизм аутентификации SASL/GSSAPI на основе пакетов Heimdal либо MIT Kerberos V.
- Sleepycat Software BerkeleyDB или с Free Software Foundation's GNU Database Manager (GDBM) — если на время конфигурирования ни один из этих пакетов

недоступен, вы не сможете построить slapd с поддержкой главного механизма работы с базой данных.

При наличии перечисленных пакетов установка OpenLDAP не вызовет никаких трудностей.

Имя исполняемого файла сервера OpenLDAP — slapd.

Настройка LDAP-сервера

Для конфигурирования сервера необходимо в файле slapd.conf определить соответствующие переменные. Файл может располагаться в каталоге /etc либо /usr/local/etc/openldap. Далее приводятся наиболее общеупотребительные директивы файла slapd.conf. Полный список директив можно найти в соответствующем файле справки.

Формат конфигурационного файла

Файл slapd.conf состоит их трех типов конфигурационной информации: глобальной, специфичной для механизма базы данных и специфичной для базы данных. Глобальная информация указывается первой, за ней следует информация, связанная с механикой базы данных, за которой следует информация, связанная с отдельным экземпляром базы данных.

Глобальные директивы могут переопределяться в описании механизмов баз данных и/или директивах базы данных. Директивы механизма базы данных могут переопределяться директивами базы данных.

Пустые строки и строки с комментариями (начинаются с символа #) игнорируются. Если строка начинается с пробела, она рассматривается как продолжение предыдущей строки. Общий формат файла slapd.conf приведен в листинге 25.1.

Листинг 25.1

```
# глобальные конфигурационные директивы
<глобальные конфигурационные директивы>

# определение механизма базы данных
backend <typeA>
<специфичные для механизма базы данных директивы>

# определение базы данных и конфигурационные директивы
database <typeA>
<директивы, специфичные для базы данных>

# определение второй базы данных и конфигурационные директивы
database <typeB>
<директивы, специфичные для базы данных>

# определение третьей базы данных и конфигурационные директивы
```

```
database <typeA>
<директивы, специфичные для базы данных>

# последующие механизмы баз данных, определения баз данных и
# конфигурационные директивы
...
```

Конфигурационные директивы могут иметь аргументы, разделяемые пробелами. Если аргумент содержит пробелы, он должен заключаться в кавычки. Если аргумент содержит двойную кавычку или обратный слеш (\), то символ должен предваряться обратным слешем.

Глобальные директивы

Директивы, описанные в этой секции, применяются ко всем механизмам баз данных и базам данных, если они не переопределяются в специфических секциях конфигурации. Аргументы, которые необходимо заменить соответствующим текстом, приведены в скобках <>.

- `access to <что> [ by <кому> <уровень доступа> <control> ]+` — предоставляет доступ (указанный в *<уровень доступа>*) к набору записей и/или атрибутов (указанных в *<что>*) одному или более запрашивающих (указанных в *<кому>*).
- `attributetype <RFC2252 описание типа атрибута>` — определяет тип атрибута.
- `defaultaccess { none | compare | search | read | write }` — указывает доступ по умолчанию для всех запрашивающих, если не указана директива `access`. Любой назначенный уровень доступа включает в себя все нижележащие уровни доступа (например, доступ `read` предполагает также `search` и `compare`, но не `write`).
- `idletimeout <целое число>` — период ожидания в секундах перед принудительным закрытием соединения с клиентом, находящимся в состоянии ожидания.
- `include <имя файла>` — указывает slapd перед чтением следующей строчки конфигурационного файла читать дополнительную конфигурационную информацию. Включаемый файл должен иметь формат конфигурационного файла slapd. Используется для включения файлов, содержащих спецификации схем.
- `loglevel <целое число>` — указывает уровень, на котором должны регистрироваться в syslog отладочные сообщения и статистика работы. Чтобы данная функция работала (за исключением двух уровней статистики, которые всегда включены), вам следует сконфигурировать OpenLDAP с опцией `enable-debug` (по умолчанию). Уровни доступа аддитивны. Для отображения номеров, соответствующих определенному типу отладочной информации, вызовите slapd с ключом `-?`.
- Для *<целое число>* возможны значения:
  - `-1` — включает всю отладочную информацию;
  - `0` — без отладочной информации;
  - `1` — трассировать вызовы функций;
  - `2` — отладка обработки пакетов;

- 4 — тщательная отладочная трассировка;
- 8 — управление соединением;
- 16 — печать принятых и отправленных пакетов;
- 32 — обработка фильтра поиска;
- 64 — обработка конфигурационного файла;
- 128 — обработка списка контроля доступа;
- 256 — регистрировать статистику соединения/обработки/результатов;
- 512 — регистрировать статистику отправленных элементов;
- 1024 — печать коммуникаций с shell-механизмом баз данных;
- 2048 — печать отладки анализа элемента.

- objectclass *<RFC2252 описание класса объектов>* — определяет класс объектов.
- referral *<URL>* — определяет возвращаемую ссылку в случае, если slapd не может найти в локальной базе данных информацию для обработки запроса.
- sizelimit *<целое число>* — максимальное число элементов, возвращаемых одной операцией поиска.
- timelimit *<целое число>* — максимальное число секунд, которые slapd затрачивает, отвечая на запрос. Если запрос не завершился за это время, будет выдан результат, свидетельствующий об истечении времени.

Общие директивы механизмов баз данных

Директива в этой секции применима только к механизму базы данных, в котором она определена. Она поддерживается каждым типом механизма базы данных. Эта директива применима ко всем экземплярам баз данных одного типа и может переопределяться директивами баз данных.

- backend *<тип>* — отмечает начало определения механизма базы данных. *<тип>* должен быть одним из ldbm, shell, passwd или других поддерживаемых типов механизмов базы данных.

Общие директивы базы данных

Директивы этой секции применимы только к базе данных, в которой они определены. Они поддерживаются всеми типами баз данных.

- database *<тип>* — отмечает начало нового экземпляра базы данных. *<тип>* должен быть одним из ldbm, shell, passwd или другим поддерживаемым типом базы данных.
- readonly { on | off } — переводит базу данных в режим "только чтение". Любые попытки модифицировать базу данных в режиме "только чтение" вызовут ошибку "unwilling to perform".
- replica host=*<имя хоста>*[:*<порт>*] [bindmethod={ simple | kerberos | sasl }] ["binddn=*<отличительное имя>*"] [mech=*<механизм>*] [authcid=<identity>] [authzid=<identity>] [credentials=*<пароль>*] [srvtab=*<имя файла>*] — указывает место для репликации базы данных. Параметр host= определяет хост и опционально порт подчиненного сервера LDAP. В качестве *<имя хоста>* может быть либо имя домена, либо IP-адрес. Если *<порт>* не указан, то используется стандартный для LDAP номер порта. Параметр binddn= указывает имя

для привязки обновлений в подчиненном сервере. Атрибут `bindmethod` может иметь значение `simple`, `kerberos` или `sasl`, в зависимости от типа аутентификации при подключении к подчиненному slapd: простая парольная аутентификация, Kerberos-аутентификация или SASL-аутентификация. Простая аутентификация требует указания параметров `binddn` и `credentials`. Kerberos-аутентификация требует задания параметров `binddn` и `srvtab`. Аутентификация SASL требует указания используемого механизма в параметре `mech`. В зависимости от механизма, может устанавливаться особенность аутентификации и/или удостоверение, параметрами `authcid` и `credentials` соответственно. Параметр `authcid` позволяет указать особенности аутентификации.

- `replogfile` *<имя файла>* — имя регистрационного файла репликации, в котором сервер регистрирует изменения. Регистрационный файл репликации обычно записывается slapd и читается slurpd.
- `rootdn` *<отличительное имя>* — отличительное имя, которое не подлежит контролю доступа или административным ограничениям при операциях в этой базе данных. Не требуется, чтобы отличительное имя ссылалось на элемент каталога. Отличительное имя может ссылаться на SASL.

  Пример с элементом:

```
rootdn "cn=Manager, dc=example, dc=com"
Пример с SASL:
rootdn "uid=root@EXAMPLE.COM"
```

- `rootpw` *<пароль>* — пароль приведенного ранее отличительного имени, который будет всегда работать, независимо от того, существует ли элемент с данным отличительным именем и есть ли у него пароль.
- `suffix` *<суффикс отличительного имени>* — суффикс отличительного имени при запросах к этому механизму баз данных. Можно указывать несколько строк. Для каждой базы данных требуется хотя бы одна строка.
- `updatedn` *<отличительное имя>* — эта директива применима только к подчиненному slapd. Она указывает отличительное имя, которому разрешено внесение изменений в реплику. Это может быть отличительное имя, к которому привязывается slurpd при внесении изменений в реплику, или отличительное имя, ассоциированное с SASL.
- `updateref` *<URL>* — эта директива применима только к подчиненному серверу. Она указывает URL, возвращаемый клиентам, которые получают запросы обновления на реплику. Если она указана несколько раз, то предоставляется каждый URL.

Директивы, специфичные для LDBM-механизма базы данных

Директивы этой категории применимы только к механизму базы данных LDBM. Они должны следовать за строкой `database ldbm` и перед любой другой database-строкой.

- `cachesize` *<целое число>* — размер поддерживаемого экземпляром механизма базы данных LDBM кэша элементов в памяти.

- `dbcachesize <целое число>` — размер кэша в памяти (в байтах), связанного с каждым открытым индексным файлом. Если она не поддерживается методом нижележащей базы данных, то игнорируется без комментариев. Увеличение этого числа приводит к росту потребляемой памяти, но и к резкому повышению производительности, особенно при модификации или перестроении индексов.
- `dbnolocking` — если присутствует эта опция, то она отменяет блокировку базы данных. Включение такой опции может привести к увеличению производительности при снижении безопасности данных.
- `dbnosync` — эта опция приводит к отложенной синхронизации изменений в памяти с содержимым базы данных на диске. Ее включение может увеличить производительность и снизить безопасность данных.
- `directory <каталог>` — каталог, в котором находятся файлы, содержащие базу данных LDBM, и связанные с ними индексные файлы.
- `index {<список атрибутов> | default} [pres, eq, approx, sub, none]` — индексы для хранения данных атрибутов. Если указан только *<список атрибутов>*, то поддерживаются только индексы по умолчанию.
- `mode <целое число>` — режим защиты вновь создаваемых индексных файлов базы данных.

Прочие механизмы баз данных

Slapd поддерживает и другие типы механизмов баз данных (кроме LDBM):

- `ldbm` — Berkeley или GNU DBM-совместимый механизм;
- `passwd` — обеспечивает доступ к /etc/passwd в режиме "только для чтения";
- `shell` — shell-механизм (доступ к внешней программе);
- `sql` — механизм базы данных на основе SQL.

Ключи командной строки

Slapd поддерживает несколько ключей командной строки:

- `-f <имя файла>` — альтернативный файл конфигурации slapd;
- `-h <URL>` — конфигурация слушателя. По умолчанию — ldap:///, что подразумевает использование протокола LDAP поверх TCP на всех интерфейсах, определенных при конфигурировании. Вы можете задать пару хост-порт или другие схемы протокола. Хост можно указывать в форме IPv4-чисел, разделенных точками, или в форме имени хоста. Значение номера порта должно быть числом;
- `-n <имя службы>` — имя службы, используемой для регистрации и других целей. Служба по умолчанию — slapd;
- `-l <локальный пользователь syslog>` — определяет локального пользователя для функции syslog;
- `-u <пользователь> -g <группа>` — определяет соответственно пользователя и группу, от чьего имени происходит запуск программы. Пользователь может быть задан либо именем, либо uid; группа — либо именем группы, либо gid;
- `-r <каталог>` — каталог запуска. После открытия слушателей, но до чтения каких-либо конфигурационных файлов или инициализации механизмов баз данных slapd выполнит для этого каталога chroot;

- `-d <уровень> | ?` — устанавливает slapd уровень отладки на значение `<уровень>`. Если уровень — символ `?`, то выводятся различные уровни отладки, и slapd завершается, вне зависимости от любых других приложенных ключей.

  Текущие уровни отладки:
  - `-1` — включает всю отладочную информацию;
  - `0` — без отладки;
  - `1` — трассировать вызовы функций;
  - `2` — отладка обработки пакетов;
  - `4` — тщательная отладочная трассировка;
  - `8` — управление соединением;
  - `16` — печать принятых и отправленных пакетов;
  - `32` — обработка фильтров поиска;
  - `64` — обработка конфигурационного файла;
  - `128` — обработка списка контроля доступа;
  - `256` — регистрировать статистику соединения/обработки/результатов;
  - `512` — регистрировать статистику отправленных элементов;
  - `1024` — печать коммуникаций с shell-механизмом базы данных;
  - `2048` — печать отладки анализа элемента.

  Вы можете активизировать несколько уровней, указав по одному отладочному уровню в каждом ключе, или вычислить число самостоятельно.

База данных LDAP

Как уже упоминалось ранее, LDAP имеет свою базу данных. Далее мы рассмотрим основные особенности структуры баз данных.

Механизмы баз данных LDAP, объекты и атрибуты

Slapd может работать с тремя различными механизмами баз данных.

- LDBM — высокоскоростная дисковая база данных.
- SHELL — интерфейс базы данных к обычным UNIX-командам и shell-скриптам.
- PASSWD — простая база данных на основе файла паролей.
- LDAP по умолчанию использует LDBM-базу данных, что подразумевает повышенную производительность.

База данных LDBM назначает уникальный четырехбайтовый идентификатор каждому элементу базы данных. Она задействует этот идентификатор для ссылок на записи из индексов. База данных состоит из одного главного файла индексов, который устанавливает соответствие уникального индекса элемента (EID) текстовому представлению самого элемента. Также поддерживаются другие индексные файлы.

Для импортирования или экспортирования информации каталога между серверами каталогов, основанных на LDAP, или для описания набора вносимых в каталог изменений обычно применяется формат LDIF (LDAP Data Interchange Format, LDAP-формат обмена данных). Файл LDIF хранит информацию об объектно-ориентированной иерархии элементов. Пакет LDAP, который вы собираетесь использовать, поставляется с утилитой конвертирования LDIF-файлов в LDBM-формат.

В листинге 25.2 приведен типичный LDIF-файл.

Листинг 25.2

```
dn: o=Home, c=UA
o: Home
objectclass: organization
dn: cn=Vasya Pupkin, o=Home, c=UA
cn: Vasya Pupkin
sn: Pupkin
mail: vasya@yahoo.com
objectclass: person
```

Вы можете заметить, что каждый элемент уникально идентифицируется отличительным именем `dn`, которое включает имя элемента и путь имен, ведущих к вершине иерархии каталога.

В LDAP класс объектов определяет набор атрибутов для определения элемента. Стандарт LDAP определяет такие основные виды классов объектов, как:

- группа каталогов, включая неупорядоченный список индивидуальных объектов или групп объектов;
- местоположение, такое как имя страны и описание;
- организации в каталоге;
- люди в каталоге.

Элемент может принадлежать более чем одному классу. Например, элемент человека определен классом объектов `person`, но также могут быть заданы атрибуты в классах объектов `inetOrgPerson`, `groupOfNames` и `organization`. Структура классов объектов сервера (его схема) определяет общий список требуемых и разрешенных атрибутов отдельного элемента.

Данные каталога представлены в виде пар "атрибут–значение". Любая определенная часть информации ассоциируется с этим описательным атрибутом.

Например, атрибут `cn` (commonName) служит для размещения имени человека.

Каждый человек, введенный в каталог, определяется набором атрибутов в классе объектов `person`.

Требуемые атрибуты включают атрибуты, которые могут представляться в элементах, используя класс объектов. Все элементы требуют наличия атрибута `objectClass`. В нем перечислены классы объектов, к которым принадлежит элемент.

Разрешенные атрибуты включают атрибуты, которые могут быть представлены в элементах класса объектов. Например, в классе объектов `person` обязательны атрибуты `cn` и `sn`. Атрибуты `description`, `telephoneNumber`, `seeAlso` и `userpassword` разрешены, но не обязательны.

Каждый атрибут имеет соответствующее синтаксическое определение. Синтаксическое определение описывает тип предоставляемой атрибутом информации:

- `bin` (binary) — двоичный;
- `ces` (case exact string) — строка с соответствием регистра (регистр символов должен совпадать при сравнении);

- `cis` (case ignore string) — строка с игнорированием регистра (регистр символов игнорируется при сравнении);
- `tel` — строка с номером телефона (подобен `cis`, но пробелы и тире '–' игнорируются при сравнении);
- `dn` (distinguished name) — отличительное имя.

Создание и поддержание базы данных

Есть два способа создания базы данных. Во-первых, вы можете создать базу данных в реальном времени, используя LDAP. Таким образом, вы просто запускаете slapd и добавляете элементы с помощью любого LDAP-клиента на ваш выбор. Этот способ хорош для относительно небольших баз данных. Во-вторых, можно сформировать базу данных автономно средствами генерации индексов. Это наилучший метод для случая, когда вам нужно создать тысячи элементов.

Создание базы данных в реальном времени

Программный пакет OpenLDAP поставляется с утилитой ldapadd, которая добавляет записи при запущенном LDAP-сервере. Если вы выбрали создание базы данных в реальном времени, то можете добавить элементы посредством ldapadd. После первичного добавления элементов вы все еще можете использовать ldapadd для добавления элементов. Перед запуском sldap следует проверить, что в вашем файле sldap.conf установлены определенные конфигурационные опции.

```
suffix <отличительное имя>
```

Настоящая опция указывает, какие элементы помещены в этой базе данных. Вы должны ее установить в значение отличительного имени корня поддерева, которое собираетесь создать, например:

```
suffix "o=Home, c=UA"
```

Нужно проверить, указали ли вы каталог, где должны быть созданы индексные файлы:

```
directory <каталог>
```

Например:

```
directory /usr/local/home
```

В заключение вы должны удостовериться, что определение базы данных содержит определение необходимых вам индексов:

```
index {<attrlist> | default} [pres,eq,approx,sub,none]
```

Например, для индексации атрибутов `cn`, `sn`, `uid` и `objectclass` можно использовать следующие строки конфигурации индексов:

```
index cn,sn,uid
index objectclass pres,eq
index default none
```

После конфигурирования сервера запустите slapd, подключитесь к нему с помощью LDAP-клиента и добавляйте элементы.

Автономное создание базы данных

Второй метод формирования базы данных — вызов утилиты генерирования индексов — предпочтителен при создании большого объема индексов. Утилиты считывают конфигурационный файл slapd, содержащий текстовое представление добавляемых элементов, и входной LDIF-файл. Они создают непосредственно индексный файл LDBM. Есть несколько важных конфигурационных опций, которые необходимо добавить при определении базы данных в конфигурационном файле:

```
suffix <отличительное имя>
```

Как описано в предыдущем разделе, эта опция указывает, какие элементы содержатся в базе данных. Вы должны установить ее в значение отличительного имени корня создаваемого вами поддерева, например:

```
suffix "o=Home, c=UA"
```

Нужно проверить, указали ли вы каталог, где должны быть созданы индексные файлы:

```
directory <каталог>
```

Пример:

```
directory /usr/local/home
```

Далее увеличим размер внутреннего кэша, используемого каждым открытым файлом. Для наилучшей производительности при создании индекса весь индекс должен поместиться в память. Этот размер устанавливает опция

```
dbcachesize <целое число>
```

Например, посредством следующей опции будет создан кэш размером 50 Мбайт:

```
dbcachesize 50000000
```

Далее нужно указать, какие индексы вы хотите создать. Это делается одной или более опциями

```
index {<attrlist> | default} [pres,eq,approx,sub,none]
```

Пример:

```
index cn,sn,uid pres,eq,approx
index default none
```

Будут созданы индексы `presence`, `equality` и `approximate` для атрибутов `cn`, `sn` и `uid`, а для других атрибутов не будет создано никаких индексов.

Как только вы настроили все, как хотели, вы создаете первичную базу данных и ассоциированные индексы программой slapadd:

```
slapadd -l <входной файл> -f <конфигурационный файл slapd> [-d <уровень отладки>] [-n <целое число>|-b <суффикс>]
```

Значения аргументов:

- `-l <входной файл>` — определяет входной LDIF-файл, содержащий добавляемые элементы в текстовой форме;
- `-f <конфигурационный файл slapd>` — указывает конфигурационный файл slapd, в котором определено, где создавать индексы, какие индексы создавать и т. д.;

- `-d <уровень отладки>` — включает отладку в соответствии с `<уровень отладки>`. Уровни отладки такие же, как и для slapd;
- `-n <номер базы данных>` — необязательный аргумент. Указывает, какую базу данных модифицировать. Первая по списку база данных в конфигурационном файле считается `1`, вторая — `2` и т. д. Не должен использоваться вместе с ключом `-b`;
- `-b <суффикс>` — необязательный аргумент, указывающий, какую базу данных модифицировать. Представленный суффикс сопоставляется с суффиксом базы данных, и при совпадении определяется номер модифицируемой базы данных. Не должен использоваться вместе с ключом `-n`.

Утилиты

В этом разделе кратко описаны утилиты для добавления, удаления и модификации записей в LDAP.

Slapindex

Иногда возникает необходимость перегенерировать индексы (например, после модификации slapd.conf). Это можно сделать программой slapindex.

Slapcat

Программа slapcat записывает дамп базы данных в LDIF-файл. Она может быть полезна, когда вы хотите создать читаемую резервную копию вашей базы данных или отредактировать вашу базу данных автономно.

Ldapsearch

Ldapsearch — это интерфейс shell к библиотечному вызову ldap_search. Используйте утилиту для поиска элементов в вашем LDAP-механизме базы данных.

Ldapsearch открывает соединение с сервером LDAP, присоединяется и выполняет поиск с фильтром, заданным пользователем. Фильтр должен соответствовать строковому представлению фильтров LDAP, как определено в RFC1558. Если ldapsearch находит один или более элементов, то извлекаются атрибуты, элементы и их значения печатаются в стандартный вывод.

Ldapdelete

Ldapdelete — это shell-интерфейс к библиотечному вызову ldap_delete. Утилита удаляет элементы из вашего LDAP-механизма базы данных.

Ldapdelete открывает соединение к LDAP-серверу, подключается и удаляет один или более элементов. Если приложен один или более аргументов `dn`, элементы с этими отличительными именами будут удалены. Каждое отличительное имя должно быть строковым представлением отличительного имени в соответствии с RFC1779. Если вызов был сделан без аргументов, список отличительных имен читается со стандартного устройства ввода.

Ldapmodify

Ldapmodify — это shell-интерфейс к библиотечным вызовам ldap_modify и ldap_add. Утилита изменяет содержимое элементов вашего LDAP-механизма баз данных.

Ldapadd

Ldapadd — это просто жесткая ссылка на утилиту ldapmodify. При вызове ldapadd автоматически включается ключ `-a` (добавить элемент) утилиты ldapmodify.

Kldap

Kldap — графический LDAP-клиент для KDE. Отображает информационное дерево каталога.

GQ

GQ — графический LDAP-клиент для GNOME с простым интерфейсом.

Взаимодействие программ с LDAP

Наиболее универсальный способ взаимодействия программ с LDAP — проведение аутентификации через модули PAM (Pluggable Authentication Module — подключаемый модуль аутентификации). Модуль PAM называется pam_ldap и входит в большинство современных дистрибутивов. Модуль pam_ldap использует конфигурационный файл ldap.conf. В простейшем случае при настройке клиента /etc/ldap.conf содержит три строки:

```
BASE dc=home,dc=ua
HOST 192.168.0.1
pam_password clear
```

Здесь:

- `BASE` — база для поиска в дереве LDAP;
- `HOST` — IP-адрес или имя хоста, на котором работает LDAP-сервер;
- `pam_password` — тип шифрования паролей.

Основной конфигурационный файл PAM — pam.conf — находится в каталоге /etc и состоит из правил, описывающих методы проведения аутентификации для различных сервисов. В этом файле необходимо прописать записи для соответствующих программ, которым при аутентификации необходим pam_ldap.

> ***Замечание***
>
> Некоторые программы, например SQUID, напрямую взаимодействуют с LDAP без помощи PAM.

Ссылки

- Немет Э., Снайдер Г., Сибасс С., Хейн Т. Р. UNIX: руководство системного администратора. Для профессионалов: Пер. с англ. — СПб.: Питер; К.: Издательская группа BHV, 2002.
- LDAP Linux HOWTO.
- The Linux NIS(YP)/NYS/NIS+ HOWTO.
- Man-страница lapd.conf.
- RFC 1558: A String Representation of LDAP Search Filters.
- RFC 1777: Lightweight Directory Access Protocol.
- RFC 1778: The String Representation of Standard Attribute Syntaxes.
- RFC 1779: A String Representation of Distinguished Names.
- RFC 1781: Using the OSI Directory to Achieve User Friendly Naming.
- RFC 1798: Connectionless LDAP.
- RFC 1823: The LDAP Application Programming Interface.
- RFC 1959: An LDAP URL Format.
- RFC 1960: A String Representation of LDAP Search Filters.
- RFC 2251: Lightweight Directory Access Protocol (v3).
- RFC 2307: LDAP as a Network Information Service.
- **http://www.keldysh.ru/metacomputing/ism99.html** — Валиев М. К., Китаев Е. Л., Слепенков М. И. Использование службы директорий LDAP для представления метаинформации в глобальных вычислительных системах. ИПМ им. М. В. Келдыша РАН.
- **http://www.openldap.org** — официальный сайт OpenLDAP.
- **http://www.opennet.ru/docs/RUS/ldap/index.html** — Захаров М. Руководство по настройке аутентификации пользователей посредством LDAP.

Глава 26



NFS — сетевая файловая система

NFS (Network File System, сетевая файловая система) — это программное обеспечение, позволяющее предоставлять в общее сетевое использование дисковое пространство хоста и включать в локальное дерево каталогов общие сетевые ресурсы. NFS позволяет работать с удаленной файловой системой через разные типы сетевых соединений: локальные и беспроводные сети, а также через модемные соединения. NFS разработана и используется давно — еще с начала 80-х годов. Ее реализация и поддержка присутствует в любой версии UNIX и Windows.

Не будем заострять внимание на безопасности и надежности NFS, а перейдем сразу к процессу ее установки и настройки на Linux-компьютере.

Установка и настройка NFS-сервера

Как и большинство сетевых служб, NFS — клиент-серверная система. И как обычно, настройка серверной части сложнее, чем клиентской. С нее мы и начнем.

Установить NFS не составляет проблем, поскольку в большинстве дистрибутивов она входит в стандартную поставку. После установки пакета необходимо его настроить и сконфигурировать.

К сожалению (для администратора системы), для функционирования NFS необходимо наличие службы RPC (Remote Procedure Called, служба вызова удаленных процедур), в Linux это пакет portmap.

Хосты, которые имеют право доступа к службе RPC, определяют в конфигурационном файле /etc/hosts.allow, а запрет доступа к службе — в файле /etc/hosts.deny.

Для указания, какой каталог файловой системы можно предоставить в пользование удаленным компьютерам (экспортировать) и каким компьютерам можно монтировать эти каталоги, используется файл /etc/exports.

Этот файл содержит строки в следующем формате:

```
Полный_путь_к_каталогу↵
    Имя_хоста_которому_разрешено_монтирование_каталога (Права_доступа)
```

Пример:

```
/home/boss/documents  zam(ro)
```

Здесь хост с именем `zam` может читать файлы (`ro`) из каталога `/home/boss/documents`, если вам нужна возможность записи, воспользуйтесь опцией (`rw`). Помимо этого способа управления доступом можно воспользоваться NIS или LDAP.

После того как определены хосты, которым можно иметь доступ к экспортируемым ресурсам, приступим к тестированию NFS. Для нормального функционирования NFS необходимы следующие демоны: portmapper, mountd и nfsd. Проверить наличие этих демонов несложно:

```
rpcinfo -p.
```

Результат работы программы приведен в листинге 26.1.

Листинг 26.1

```
program vers proto   port
  100000    2   tcp    111  portmapper
  100000    2   udp    111  portmapper
  100005    1   udp    745  mountd
  100005    1   tcp    747  mountd
  100003    2   udp   2049  nfs
  100003    2   tcp   2049  nfs
```

Если выполнение программы завершилось ошибкой, необходимо убедиться, что у вас установлены и запущены ранее перечисленные программы и правильно заполнены файлы hosts.allow и hosts.deny.

При изменении файла exports необходимо заставить демон nfsd перечитать файл exports. Это можно сделать, выполнив следующие команды:

```
killall -HUP /usr/sbin/mountd
killall -HUP /usr/sbin/nfsd
```

Установка и настройка NFS-клиента

Установка NFS-клиента не вызывает проблем, в большинстве дистрибутивов он входит в стандартную поставку. Перед установкой клиента необходимо убедиться, что ядро собрано с поддержкой файловой системы NFS.

После установки клиента попробуем подмонтировать удаленную файловую систему.

Пусть мы хотим смонтировать раздел /home/boss/documents, расположенный на хосте boss. Это делается с помощью команды

```
mount -o rsize=1024,wsize=1024 boss:/home/boss/documents /mnt/docs
```

Если вместо монтирования файловой системы команда `mount` выдаст сообщение об ошибке:

```
mount: boss:/home/boss/documents failed, reason given by server:
Permission denied
```

то в файле exports не разрешен доступ к этому ресурсу для хоста, на котором пытаются смонтировать удаленный каталог.

Чтобы прекратить использование файловой системы, нужно выполнить команду

```
umount /mnt
```

Для автоматического монтирования файловой системы NFS при загрузке системы в файл /etc/fstab необходимо добавить строку

```
boss:/home/bosss/documents /mnt/docs nfs rsize=1024, wsize=1024,
↳hard, intr 0 0
```

Опции монтирования

В этом разделе мы более подробно рассмотрим некоторые опции монтирования удаленной файловой системы.

Поскольку работа с удаленными источниками данных априори на несколько порядков менее надежна, необходимо тщательно подходить к опциям монтирования удаленной файловой системы, чтобы не возникало проблем (вплоть до потери реакции системы) при обрывах сетевого соединения. Грамотно настроенные параметры монтирования помогут также увеличить производительность сетевого доступа к удаленной файловой системе.

rsize

Опция `rsize` позволяет задавать размер блока чтения (в байтах). По умолчанию — 8192 байта.

wsize

Опция `wsize` позволяет задавать размер блока записи (в байтах). По умолчанию — 8192 байта.

Применяя эти параметры, можно добиться максимальной производительности на медленных и неустойчивых соединениях (например, модемных).

soft

NFS-клиент будет сообщать об ошибке программе, которая попытается получить доступ к файлу, расположенному на файловой системе, смонтированной через NFS. Этой опции рекомендуется избегать, поскольку она может привести к появлению испорченных файлов и потере данных.

hard

При использовании опции монтирования `hard` программа, осуществляющая доступ к файлу на смонтированной по NFS файловой системе, просто приостановит выполнение при разрыве связи с сервером. Процесс не может быть прерван или "убит" до тех пор, пока вы явно не укажете опцию `intr`.

timeo=n

Параметр `n` задает задержку в десятых долях секунды до посылки первой ретрансляции после тайм-аута RPC. По умолчанию эта величина равна 0,7 с. После первого тайм-аута время ожидания удваивается после каждого тайм-аута, пока не будет достигнута величина 60 с, или произойдет ретрансляция, заданная параметром `retrans`, вызвав главный тайм-аут.

retrans=n

Величина `n` задает число неосновных тайм-аутов и ретрансляций, которые должны произойти до возникновения главного тайм-аута. По умолчанию эта величина равна трем. Когда возникает главный тайм-аут, то файловые операции либо прерываются, либо выдается сообщение "server not responding".

Безопасность NFS

Основная проблема NFS заключается в том, что клиент, если не принять специальных мер, будет доверять серверу и наоборот. Это значит, что если запись администратора сервера NFS взломана, то так же легко может быть взломана запись администратора клиентской машины. И наоборот.

Безопасность клиента

Для обеспечения безопасности на клиентской стороне необходимо запретить выполнение программ с установленным битом `suid` в файловой системе NFS. Для этого в опциях монтирования следует добавить опцию `nosuid`. Также рекомендуется запретить выполнение файлов на смонтированной файловой системе с помощью опции `noexec`, однако это не всегда возможно, поскольку файловая система может содержать программы, которые необходимо выполнять.

Безопасность сервера

Для исключения возможности доступа с удаленного хоста к файлам, владельцем которого является пользователь root, необходимо указать опцию `root_squash` в файле exports:

```
/home/boss/documents zam(rw,root_squash)
```

Таким образом, если пользователь с UID 0 на стороне клиента попытается получить доступ, то файловый сервер выполнит подстановку UID пользователя nobody на сервере. Это означает, что администратор клиента не сможет получить доступ к файлам, к которым имеет доступ только администратор сервера. Аналогичная ситуация с изменением файлов.

Помимо этого, вы можете запретить доступ определенным пользователям или группе пользователей удаленного компьютера. За более подробной информацией обратитесь к руководству по NFS.

Также необходимо с помощью файлов hosts.allow и hosts.deny разрешить доступ только тем хостам, которые будут использовать ваши экспортируемые ресурсы.

И, наконец, защитите порты nfs, mountd и portmap с помощью firewall на вашем маршрутизаторе. Nfsd задействует порт с номером 2049 и протоколы UDP и TCP. Portmapper — порт номер 111 и протоколы TCP и UDP. Mountd — порты 745 и 747 и протоколы TCP и UDP.

Ссылки

- Справочные страницы man: NFS, portmap, mountd, nfsd, exports.
- NFS-HOWTO.



Глава 27

Сервер Samba для клиентов Windows

В современном мире, как бы ни хотелось некоторым фирмам и личностям, невозможно продуктивно работать и существовать без взаимодействия и сотрудничества с конкурентами/соперниками. Вряд ли в вашей фирме есть только компьютеры под управлением Linux. Почти наверняка найдется парочка и с Windows, причем разных семейств. Само собой, хочется все это хозяйство "подружить", сделать более-менее прозрачный доступ к различным сетевым ресурсам.

Наверняка читатель знает, что *сетевое окружение* в понятиях Microsoft Windows — сложная система, позволяющая обеспечивать совместный доступ к дискам, каталогам, принтерам, организовывать домен, использовать Active Directory и некоторые другие "прелести".

При этом достаточно много всяких тонкостей и ограничений возникает из-за различной реализации поддержки сети в разных версиях Windows. Кое-что исправляется патчами (patch), а некоторые ограничения невозможно устранить из-за особенностей архитектуры Windows различных версий.

Для корректного сосуществования Linux и Windows в мире UNIX существует пакет Samba, предназначенный для взаимодействия с клиентами сети Microsoft Windows.

Этот пакет дает возможность Linux-системе выступать в качестве файлового и принт-сервера в сети Microsoft Windows, а также позволяет компьютеру под управлением Linux функционировать в качестве первичного контроллера домена (Primary Domain Controller, PDC) сети Windows. Помимо этого есть Samba-клиент для операционной системы Linux, обеспечивающий подключение Linux-клиента к ресурсам, предоставляемым серверами сети Microsoft Windows.

Такая объединенная схема дает ряд преимуществ:

- поскольку в целом ОС Linux зачастую устойчивей Windows XP/Vista, повышается надежность функционирования системы;
- отпадает необходимость приобретать лицензионную Windows для организации сервера печати и доступа к данным;
- если у вас уже есть Linux-сервер, представляется рациональным нагрузить его дополнительной работой;
- сервер Samba имеет возможность мониторинга и удаленного управления как через SSH, так и через Web-интерфейс, предоставляемый пакетом SWAT (Samba Web-based Administrative Tool);
- обычно на небольшом сервере пакет Samba 3 работает быстрее, чем Windows Server.

Установка сервера Samba проблем не вызывает — достаточно при инсталляции Linux выбрать установку Samba. Если вы не сделали этого, то сервер можно установить на вашем компьютере с помощью команды `rpm -i sambaXXX.rpm`.

Если вы хотите установить самую последнюю версию пакета, и она досталась вам в виде TGZ-архива, содержащего исходный текст, процесс установки несколько растянется.

1. Сначала необходимо распаковать архив, содержащий исходные коды Samba. Для этого нужно выполнить следующую команду:

   `tar zxvf samba-X.X.X.tar.gz`,

   где *X.X.X* — версия пакета.
2. После этого следует перейти в каталог, где находятся исходные коды. Там есть и файл Readme, в котором подробно рассказано, как сконфигурировать, откомпилировать и установить пакет Samba.
3. Далее выполнить команду

   `configure --with-smbmount --prefix=/opt/samba --with-msdfs`,

   которая конфигурирует файл Makefile.

   Примечание

   Команда указывает компилировать утилиту smbmount, которая служит для монтирования SMB-ресурсов в файловую структуру Linux, включает поддержку Microsoft DFS и указывает устанавливаться после компиляции в каталог /opt/samba. Конечно, есть еще много параметров, которые можно назначить. Подробную информацию о них следует смотреть в документации к пакету Samba или вызывать командой `configure --help`.

4. Следующее действие — набрать в командной строке `make` и нажать <Enter>. Этой командой запускается процесс компиляции программного пакета.
5. Если в ходе работы программы make не появились сообщения об ошибках, то далее необходимо выполнить команду `make install`, которая установит пакет Samba в ее родной каталог (если действовать в точности по инструкции, то файлы попадут в каталог /opt/samba).

   Теперь на очереди конфигурирование сервера Samba.

Файл конфигурации smb.conf

Самое трудное, с чем можно столкнуться при настройке сервера Samba — это создание (или редактирование) файла конфигурации. Все файлы конфигурации Samba находятся в каталоге /etc/samba. Вот список этих файлов:

- lmhosts — содержит список хостов и соответствующих им адресов;
- smbpasswd — содержит пароли пользователей сервера Samba (в зашифрованном виде);
- smbusers — файл, предназначенный для хранения списка пользователей, которым разрешен доступ к ресурсам Samba;
- smb.conf — главный конфигурационный файл сервера.

Примеры конфигурационных файлов, поставляемых с пакетом, находятся в каталоге examples (точнее будет сказать: каталог examples в исходном коде Samba). В большинстве случаев их можно использовать в качестве базы.

В листинге 27.1 приведен пример файла smb.conf сервера Samba, который успешно функционирует на одном из серверов.

Листинг 27.1

```
# This is the main Samba configuration file. You should read the
# smb.conf(5) manual page in order to understand the options listed
# here. Samba has a huge number of configurable options (perhaps too
# many!) most of which are not shown in this example
# Any line which starts with a ; (semi-colon) or a # (hash)
# is a comment and is ignored. In this example we will use a #
# for commentary and a ; for parts of the config file that you
# may wish to enable
# NOTE: Whenever you modify this file you should run the command "testparm"
# to check that you have not many any basic syntactic errors.
#
#======================= Global Settings ===================================
[global]

# workgroup = NT-Domain-Name or Workgroup-Name
   workgroup = Kontora

# server string is the equivalent of the NT Description field
   server string = Kontora Samba Server

# This option is important for security. It allows you to restrict
# connections to machines which are on your local network. The
# following example restricts access to two C class networks and
# the "loopback" interface. For more examples of the syntax see
# the smb.conf man page
hosts allow = 192.168.10.
# if you want to automatically load your printer list rather
# than setting them up individually then you'll need this
#   printcap name = /etc/printcap
#   load printers = yes

# It should not be necessary to spell out the print system type unless
# yours is non-standard. Currently supported print systems include:
# bsd, sysv, plp, lprng, aix, hpux, qnx
   printing = lprng

# Uncomment this if you want a guest account, you must add this to /etc/passwd
```

```
# otherwise the user "nobody" is used
;  guest account = pcguest

# this tells Samba to use a separate log file for each machine
# that connects
   log file = /var/log/samba/%m.log

# Put a capping on the size of the log files (in Kb).
   max log size = 0

# Security mode. Most people will want user level security. See
# security_level.txt for details.
   security = user
# Use password server option only with security = server or
# security = domain
;   password server = <NT-Server-Name>

# Password Level allows matching of _n_ characters of the password for
# all combinations of upper and lower case.
;  password level = 8
;  username level = 8

# You may wish to use password encryption. Please read
# ENCRYPTION.txt, Win95.txt and WinNT.txt in the Samba documentation.
# Do not enable this option unless you have read those documents
encrypt passwords = yes
smb passwd file = /etc/samba/smbpasswd

# The following are needed to allow password changing from Windows to
# update the Linux system password also.
# NOTE: Use these with 'encrypt passwords' and 'smb passwd file' above.
# NOTE2: You do NOT need these to allow workstations to change only
#        the encrypted SMB passwords. They allow the Unix password
#        to be kept in sync with the SMB password.
;  unix password sync = Yes
;  passwd program = /usr/bin/passwd %u

# Unix users can map to different SMB User names
;  username map = /etc/samba/smbusers

# Using the following line enables you to customise your configuration
# on a per machine basis. The %m gets replaced with the netbios name
```

```
# of the machine that is connecting
;   include = /etc/samba/smb.conf.%m

# Most people will find that this option gives better performance.
# See speed.txt and the manual pages for details
   socket options = TCP_NODELAY SO_RCVBUF=8192 SO_SNDBUF=8192

# Configure Samba to use multiple interfaces
# If you have multiple network interfaces then you must list them
# here. See the man page for details.
interfaces = 192.168.10.0/24

# Configure remote browse list synchronization here
#  request announcement to, or browse list sync from:
#  a specific host or from / to a whole subnet (see below)
;   remote browse sync = 192.168.3.25 192.168.5.255
# Cause this host to announce itself to local subnets here
;   remote announce = 192.168.1.255 192.168.2.44

# Browser Control Options:
# set local master to no if you don't want Samba to become a master
# browser on your network. Otherwise the normal election rules apply
;   local master = no

# OS Level determines the precedence of this server in master browser
# elections. The default value should be reasonable
;   os level = 33

# Domain Master specifies Samba to be the Domain Master Browser. This
# allows Samba to collate browse lists between subnets. Don't use this
# if you already have a Windows NT domain controller doing this job
;   domain master = yes

# Preferred Master causes Samba to force a local browser election on #startup
and gives it a slightly higher chance of winning the election
;   preferred master = yes

# Enable this if you want Samba to be a domain logon server for
# Windows95 workstations.
;   domain logons = yes

# if you enable domain logons then you may want a per-machine or
```

```
# per user logon script
# run a specific logon batch file per workstation (machine)
;   logon script = %m.bat
# run a specific logon batch file per username
;   logon script = %U.bat

# All NetBIOS names must be resolved to IP Addresses
# 'Name Resolve Order' allows the named resolution mechanism to be
# specified the default order is "host lmhosts wins bcast".
name resolve order = wins lmhosts bcast

# Windows Internet Name Serving Support Section:
# WINS Support — Tells the NMBD component of Samba to enable it's WINS Server
wins support = yes

# WINS Server — Tells the NMBD components of Samba to be a WINS Client
#  Note: Samba can be either a WINS Server, or a WINS Client, but NOT both
;   wins server = w.x.y.z

# WINS Proxy — Tells Samba to answer name resolution queries on
# behalf of a non WINS capable client, for this to work there must be
# at least one  WINS Server on the network. The default is NO.
;   wins proxy = yes

# DNS Proxy — tells Samba whether or not to try to resolve NetBIOS names
# via DNS nslookups. The built-in default for versions 1.9.17 is yes,
# this has been changed in version 1.9.18 to no.
   dns proxy = no

# Case Preservation can be handy — system default is _no_
# NOTE: These can be set on a per share basis
;  preserve case = no
;  short preserve case = no
# Default case is normally upper case for all DOS files
default case = lower
# Be very careful with case sensitivity — it can break things!
case sensitive = no

client code page = 866
character set = koi8-r
printer driver file=/home/samba/hplj1200/printers.def
#============================ Share Definitions ========================
```

```
[homes]
   comment = Home Directories
   browseable = no
   writable = yes
   valid users = yura katya lena alst

[comm]
        comment = Common place
        path = /home/samba/comm
        valid users = root yura katya lena alst
        public = no
        writable = yes
        printable = no
        create mask = 0775
        directory mask= 0775
        force group = office
[hp]
   comment = HP LaserJet 1200 Series PCL6
   path = /var/spool/samba
   printer = lp
   public = no
   printable = yes
   printer driver=HP LaserJet 1200 Series PCL6
   printer driver location=\\%h\printer$

[printer$]
  path=/home/samba/hplj1200
  public=yes
  browseable=yes

# This one is useful for people to share files
[tmp]
   comment = Temporary file space
   path = /tmp
   read only = no
   public = yes
```

Как видно из примера, конфигурационный файл разбит на разделы. Каждый раздел начинается с заголовка, такого как `[global]`, `[homes]` и т. д. По структуре конфигурационный файл сильно напоминает INI-файлы операционной системы Windows. Символы `#` и `;` служат признаками комментария.

Секция *[global]*

Секция `[global]` назначает переменные, которые Samba будет использовать для определения доступа ко всем ресурсам. Рассмотрим переменные секции `[global]`.

- `workgroup = Kontora` — имя NT-домена или имя рабочей группы, к которой будет принадлежать сервер Samba.
- `netbios name = Serwer` — имя сервера для отклика по протоколу NetBIOS. Не делайте его таким же, как и имя рабочей группы.
- `server string = Kontora Samba Server` — описание сервера (комментарий).
- `hosts allow = 192.168.10` — список IP-адресов компьютеров и сетей, разделенных пробелом, которые имеют право подключаться к ресурсам вашего сервера Samba.
- `printing = lprng` — тип системы печати; поддерживается `bsd`, `sysv`, `plp`, `lprng`, `aix`, `hpux`, `qnx`.
- `guest account = pcguest` — переменная нужна, если вы хотите разрешить гостевой вход на Samba-сервер. Соответствующего пользователя также придется завести в Linux-системе (или отобразить на реального пользователя типа nobody через файл /etc/samba/smbusers) Однако по соображениям безопасности не рекомендуется разрешать гостевой вход.
- `log file = /var/log/samba/%m.log` — указывает серверу создавать log-файлы отдельно для каждого пользователя; заодно указывает каталог, где будут создаваться файлы.
- `max log size = 0` — максимальный размер log-файла (ноль — без ограничений).
- `security = user` — уровень безопасности системы; обычно это уровень `user`, могут быть также уровни `share`, `server` и `domain`.
- `password server = <NT-Server-Name>` — используется только совместно с `security = server` или `security = domain`; задает имя сервера паролей.
- `password level` и `username level` — позволяют задать число символов пароля и имени пользователя.
- `encrypt passwords = yes` — разрешает пересылку паролей пользователей в зашифрованном виде; если задать `encrypt passwords = no`, то пароли пользователей будут пересылаться в незашифрованном виде, что очень плохо с точки зрения безопасности.
- `smb passwd file = /etc/samba/smbpasswd` — путь и имя файла, содержащего пароли пользователей; поскольку принципы хранения пароля в Linux не позволяют его расшифровать, приходится создавать отдельный файл паролей для пользователей Samba.
- `local master = yes` — позволяет серверу Samba стать мастер-браузером для сети, в которой он находится.
- `preferred master = yes` — позволяет серверу Samba сразу же при запуске устроить перевыборы мастера с наибольшим шансом для себя.

> ***Замечание***
>
> Протокол NetBIOS в принципе предназначен для одноранговой локальной сети, т. е. такой сети, где все компьютеры равноправны. Тем не менее в NetBIOS предусмотрен специальный компьютер, называемый мастером (master), который ведет список ком-

пьютеров, подключенных к сети, их разделяемые ресурсы и вновь подключаемые компьютеры. Именно от мастера вновь подключающиеся компьютеры получают список компьютеров в сети и их доступные ресурсы.

- `dns proxy = yes` — разрешает серверу сопоставлять NetBIOS-имена с IP-адресом при помощи DNS.
- `username map = /etc/samba/smbusers` — позволяет задать файл пользователей Samba, в котором ставится соответствие имя Linux-пользователя имени Samba-пользователя; обычно в качестве имени пользователя Samba используется имя Linux-пользователя.
- `socket options = TCP_NODELAY SO_RCVBUF=8192 SO_SNDBUF=8192` — используется для тонкой настройки сетевых параметров, позволяющих несколько улучшить производительность сервера.
- `interfaces = 192.168.10.0/24` — указывает серверу, с какой сетевой картой (сетью) он имеет дело; необходима в том случае, если на сервере установлено несколько сетевых карт из разных локальных сетей.
- `name resolve order = wins lmhosts bcast` — определяет порядок получения имен.
- `wins support = yes` — указывает, что сервер Samba выступает в роли WINS-сервера.
- `wins server = w.x.y.z` — IP-адрес WINS-сервера; если установлено `wins support = yes`, то переменная `wins server` запрещена.
- `default case = lower` — регистр имен файлов, создаваемых на ресурсах Samba.
- `case sensitive = no` — чувствительность к регистру символов.
- `client code page = 866` — кодовая страница клиента; для DOS-клиента — `866`.
- `character set = koi8-r` — набор символов, используемых сервером.
- `printer driver file=/home/samba/hplj1200/printers.def` — имя драйвера принтера.
- `time server = true` — предписывает серверу показывать клиентам Windows, что он выступает для них в роли сервера точного времени.

Секция *[homes]*

Секция `[homes]` позволяет удаленным пользователям получить доступ к своим домашним каталогам на Linux-машине. Для этого пользователь должен быть зарегистрирован в Linux-системе. Рассмотрим переменные секции `[homes]`.

- `comment = Home Directories` — эта переменная просто комментирует содержимое данной секции.
- `browseable = no` — запрещает просматривать каталог посторонним пользователям.
- `writable = yes` — разрешает записывать в домашний каталог.
- `valid users = yura katya alst` — список пользователей, для которых разрешен доступ к своим домашним каталогам; в принципе параметр необязательный.

Секция *[comm]*

Секция `[comm]` отвечает за каталог, доступный всем пользователям Samba. Это своего рода аналог FTP, куда могут записывать и откуда могут читать пользователи. Разберем подробнее данную секцию.

- `comment = Common place` — просто комментирует содержимое данной секции.
- `path = /home/samba/comm` — определяет каталог, который используется для совместного доступа.
- `valid users = root yura katya alst` — список пользователей, которым разрешен доступ к общему ресурсу.
- `public = no` — запрещает остальным пользователям получать доступ к данному ресурсу.
- `writable = yes` — разрешает запись в общий ресурс.
- `printable = no` — указывает, что разделяемый ресурс не является печатающим устройством.
- `create mask = 0775` — маска для создания файлов на разделяемом ресурсе.
- `directory mask = 0775` — маска для создания каталогов на разделяемом ресурсе.
- `force group = office` — определяет, что файлу, создаваемому или копируемому на общий ресурс, принудительно задается принадлежность к группе `office`, для того чтобы любой пользователь, который входит в группу office, мог изменить или удалить файл.

Секция *[tmp]*

Секция `[tmp]` предназначена для создания разделяемого ресурса, в который могли бы записывать все пользователи. Она отличается от секции `[comm]` отсутствием списка пользователей и значением переменной `public`:

```
comment = Temporary file space
path = /tmp
read only = no
public = yes
```

Пароли пользователей

Сервер Samba имеет несколько типов безопасности. В частности переменная `encrypt password` определяет, какой механизм авторизации будет использован. Если переменной `encrypt password` присвоено значение `no`, то авторизация пользователей производится исходя из учетных записей Linux, хранящихся в файлах /etc/passwd и /etc/shadow. При таком типе авторизации пользователя пароли передаются по сети в незашифрованном виде, что несколько упрощает настройку, но резко снижает безопасность системы. В дополнение к этому, такой тип авторизации требует изменений в системном реестре в Windows 95, Windows 98 и Windows NT. Далее приведены изменения, которые необходимо внести в системный реестр:

- Windows 95

```
[HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\VxD\VNETSUP]
"EnablePlainTextPassword"=dword:00000001
```

- Windows 98

```
[HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\VxD\VNETSUP]
"EnablePlainTextPassword"=dword:00000001
```

- Windows NT

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Rdr\Parameters]
"EnablePlainTextPassword"=dword:00000001
```

- Windows 2000

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\LanmanWorkStation\Par
ameters]
"EnablePlainTextPassword"=Data: 0x01
```

В том случае, если переменной `encrypt password` присвоено значение `yes`, авторизация пользователя происходит с использованием файла /etc/samba/smbpasswd, и передача паролей происходит в зашифрованном виде.

Почему для шифрованных паролей необходимо создавать отдельную базу паролей пользователей Samba? Все дело в методе хранения пароля. Windows-системы хранят зашифрованный пароль и при аутентификации пользователя сверяют пароли. Linux *не хранит* пароль как таковой. В файле shadow хранится так называемый хэш (hash) пароля, а в последних версиях Linux — контрольная сумма пароля, рассчитанная по алгоритму MD5. При аутентификации пользователя сравниваются хэши паролей. Особенность хэша — он необратим, т. е., зная хэш, невозможно по нему восстановить пароль. Поэтому приходится отдельно для Samba заводить базу паролей пользователей. Для администратора системы это представляет некоторое неудобство — еще один повод забыть прописать пользователя, а с другой стороны — за все нужно платить.

Добавление пользователей Samba

Для добавления пользователей в файл /etc/samba/smbpasswd необходимо наличие самого файла /etc/samba/smbpasswd. Должна также существовать учетная запись пользователя в Linux-системе. Если эти условия соблюдены, следует:

- воспользоваться программой smbpasswd для создания учетной записи: `smbpasswd -a user_name`;
- активизировать учетную запись: `smbpasswd -e user_name`.

Эта операция потребуется для каждого пользователя. Существуют скрипты, позволяющие "перебросить" пользователей из файла passwd в файл smbpasswd. Но пароли все равно придется заводить вручную. Еще один недостаток этих скриптов — после них нужно удалять пользователей типа nobody, root, news и т. п.

Команды `smbclient` и `smbmount` монтируют ресурсы, предоставляемые сервером Samba. Обо всех возможностях этих команд можно узнать из соответствующих man-страниц, краткие сведения о команде `smbclient` вы получите в этой главе.

Принтеры

Принтер, установленный в системе с сервером Samba, предоставить в общее пользование Samba-клиентам очень просто. Все принтеры, которые определены в файле /etc/printcap, становятся доступными после того, как вы добавите следующую секцию в конфигурационный файл smb.conf (листинг 27.2).

Листинг 27.2

```
[printers]
path = /var/spool/lpd
writeable = no
guest ok = no
printable = yes
```

Использование ресурсов Samba

Хотя сервер Samba позиционируется как средство доступа Windows-клиентов к ресурсам Linux-систем, тем не менее, в пакете есть средства для того, чтобы Linux-компьютеры могли также просматривать и монтировать SMB-ресурсы. И что особенно приятно, доступ к ресурсам Windows-сети можно получить и в том случае, когда сервером является машина с Windows.

Программа клиента SMB для Linux включена в дистрибутив Samba и называется smbclient. Она обеспечивает FTP-подобный интерфейс командной строки. Также существует пакет samba-client, который позволяет монтировать и размонтировать SMB-ресурсы с помощью стандартной команды `mount/umount` или `smbmount/smbumount`. Для `mount/umount` нужно указывать параметр `-t smbfs`.

Для того чтобы увидеть доступные SMB-ресурсы, выполните команду

```
/usr/bin/smbclient -L host,
```

где `host` — это имя машины, ресурсы которой вас интересуют. Данная команда вернет список имен доступных сервисов.

Листинг 27.3 содержит пример команды `smbclient`.

Листинг 27.3

```
  smbclient -L ziga
Server time is Sat Aug 17 19:58:27 2010
Timezone is UTC+2.0
Password:
Domain=[WORKGROUP] OS=[Windows NT 4.5] Server=[NT LAN Manager 4.5]

Server=[ZIGA] User=[] Workgroup=[WORKGROUP] Domain=[]

        Sharename       Type       Comment
```

```
        ---------       ----      -------
        ADMIN$          Disk      Remote Admin
        public          Disk      Public
        C$              Disk      Default share
        HP              Printer   HP6L


This machine has a browse list:

        Server               Comment
        ---------            -------
        HOP                  Samba 3.4.10p8
        ZIGA
```

Для использования сервиса выполните следующую команду:

`/usr/bin/smbclient service <password>`,

где `service` — имя хоста и сервиса. Например, если вы пытаетесь обратиться к каталогу, который доступен под именем `public` на машине, названной `ziga`, то имя сервиса должно представлять собой `\\ziga\public`. Поскольку в языке C (а точнее в языке shell) обратный слеш является спецсимволом, то необходимо ввести такую строку:

`/usr/bin/smbclient \\\\ziga\\public <mypasswd>`,

где *<mypasswd>* — ваш пароль.

В результате вы должны получить приглашение smbclient:

```
smb: \>
```

Для получения справки необходимо ввести `h` и нажать <Enter> (листинг 27.4).

Листинг 27.4

```
smb: \> h
ls             dir            lcd            cd             pwd
get            mget           put            mput           rename
more           mask           del            rm             mkdir
md             rmdir          rd             prompt         recurse
translate      lowercase      print          printmode      queue
cancel         stat           quit           q              exit
newer          archive        tar            blocksize      tarmode
setmode        help           ?              !
smb: \>
```

Как видите, практически все команды дублируют команды FTP-клиента.

Утилита smbclient многое позволяет, однако она неудобна. Если от Windows-сети нужен только доступ к дисковым ресурсам, рекомендуется воспользоваться пакетом Smbfs.

В пакет samba-client входят утилиты `smbmount` и `smbumount`, которые работают подобно `mount` и `umount`. Также есть графическая утилита `gnomba`, подобная программе **Сеть** Windows.

Конфигурирование Samba в качестве первичного контроллера домена

Ранее мы рассмотрели основные моменты конфигурирования сервера в качестве простого сервера, предоставляющего в пользование свои ресурсы. В этом разделе мы будем конфигурировать Samba так, чтобы сервер выступал в качестве первичного контроллера домена сети Windows.

Конфигурирование Samba в качестве контроллера домена можно разделить на два больших шага:

- Настройка Samba PDC.
- Создание доверенных бюджетов машин и подключение клиентов к домену.

Есть несколько моментов, на которые следует обратить ваше внимание:

- необходимо включить шифрование паролей;
- сервер должен поддерживать `domain logons` и ресурс `[netlogon]`;
- для того чтобы клиенты Windows корректно определяли сервер как контроллер домена, сервер должен быть главным обозревателем сети (domain master browser).

В листинге 27.5 приведена часть файла конфигурации smb.conf, в которой прописаны параметры, позволяющие серверу Samba стать контроллером домена.

Листинг 27.5

```
[global]
    ; основные настройки сервера
    netbios name = domain_pdc
    workgroup = test

    ; сервер должен выступать в роли domain и local master browser
    os level = 64
    preferred master = yes
    domain master = yes
    local master = yes

   ; название сервера-контроллера домена
    password server = domain_pdc
    ; разрешить работу с доверенными доменами
        allow trusted domains = yes
        ; поддержка правил доступа и ограничений в стиле NT
          nt acl support = yes
```

```
    ; настройки безопасности
    security = user

    ; для PDC требуется шифрование паролей
    encrypt passwords = yes

    ; поддержка domain logons
    domain logons = yes

    ; место, куда помещать профили пользователей
    logon path = \\%N\profiles\%u
    ; место нахождения домашних каталогов пользователей
    ; и где они должны быть смонтированы
    logon drive = H:
    logon home = \\homeserver\%u

    ; указываем общий скрипт подключения для всех пользователей
    ; это относительный путь к [netlogon] ресурсу
    logon script = logon.cmd

; необходимый ресурс для контроллера домена
[netlogon]
    path = /usr/local/samba/lib/netlogon
    writeable = no
    write list = ntadmin

; ресурс для размещения профилей пользователей
[profiles]
    path = /export/smb/ntprofile
    writeable = yes
    create mask = 0600
    directory mask = 0700
```

Утилиты

Как и у других подобных проектов, для пакета Samba существует достаточно много сторонних утилит, позволяющих упростить конфигурирование и доступ к ресурсам.

Вот список утилит и программ, в той или иной мере относящихся к пакету Samba:

- smbstatus — утилита для мониторинга Samba;
- SWAT — инструмент для конфигурирования Samba через Web-интерфейс;
- smbpasswd — управление паролями Samba;

- testparm — проверка конфигурационного файла;
- testprns — проверка конфигурации принтера;
- smbtar — SMB-утилита резервного копирования;
- smbclient — клиент командной строки;
- Ksamba — KDE-программа, предназначенная для конфигурации;
- Smbedit — Win32-приложение для правки конфигурационного файла Samba;
- Webmin — универсальная программа конфигурации через Web-интерфейс, в том числе и Samba;
- GSMB — графический интерфейс для утилиты smbpasswd;
- SambaSentinel — графический интерфейс для утилиты smbstatus.

SWAT

SWAT (Samba Web Administration Tool) — одна из наиболее известных утилит для работы с сервером Samba через Web-интерфейс (рис. 27.1). Для доступа к SWAT в браузере необходимо набрать `localhost:901`. Далее, после ввода логина и пароля вы получаете доступ к программе SWAT.

SWAT охватывает практически все настройки Samba, которые доступны администратору через Web-интерфейс.

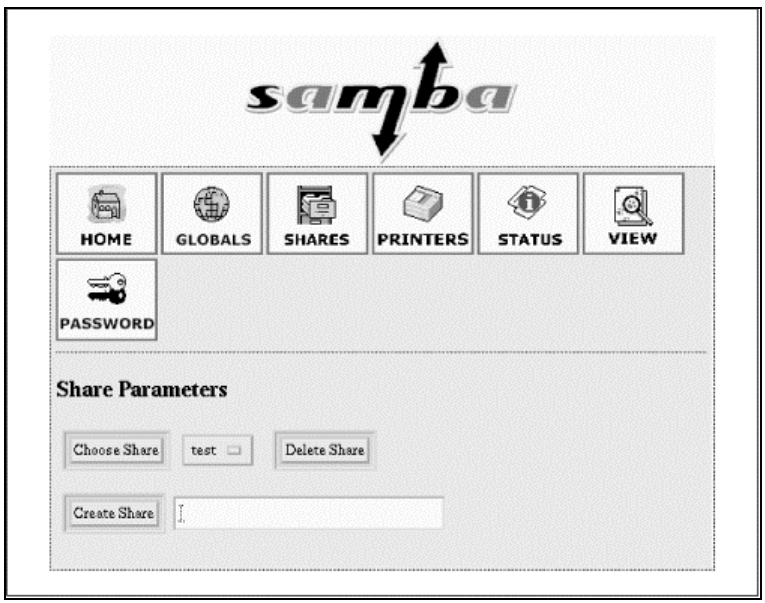


Рис. 27.1. SWAT — страница глобальных переменных

Webmin

Webmin — программа с Web-интерфейсом, позволяющая конфигурировать множество служб и сервисов через Web (рис. 27.2). В частности есть возможность настраивать сервер Samba.

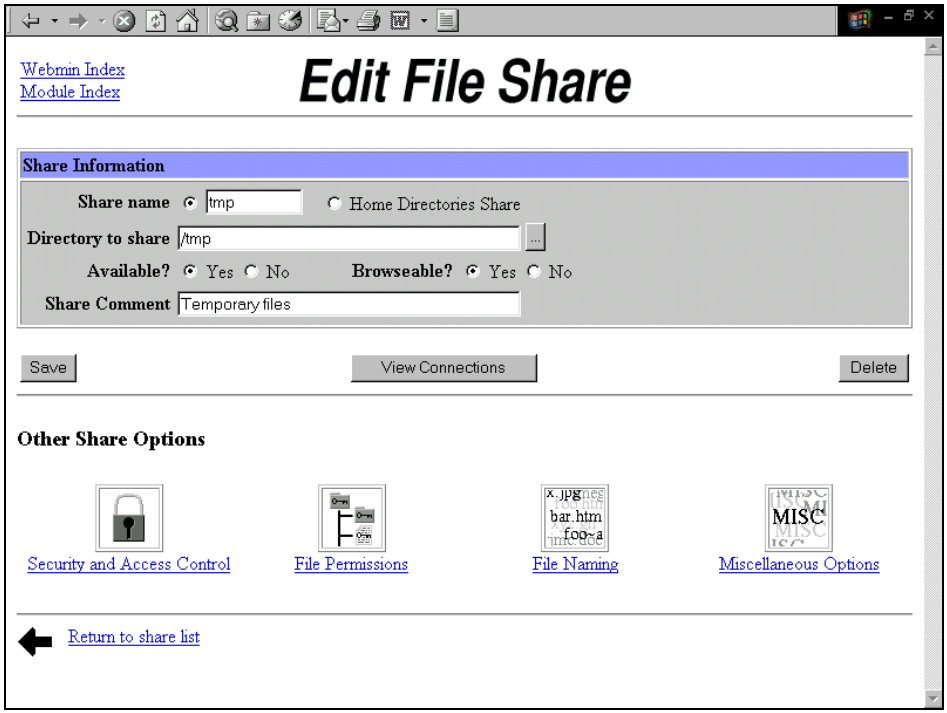


Рис. 27.2. Webmin — управление Samba

Ksamba

Ksamba — удобная и понятная программа для KDE-оболочки, предназначенная для конфигурации Samba.

SambaSentinel

Графический интерфейс к утилите smbstatus. Позволяет проводить мониторинг, удалять зависшие задачи и т. п.

Ссылки

- **boombox.campus.luth.se/sambasentinel.php** — сайт проекта sambasentinel.
- **www.culte.org/projets/developpement/gsmb/** — официальный сайт проекта GSMB.
- **www.linuxcenter.ru/lib/soft/samba_pdc.phtml** — как настроить Samba 2.2 в качестве основного контроллера домена (Primary Domain Controller, PDC).
- **www.linuxoid.ru/how_to/samba5.html** — Басин Илья. Samba за пять минут.
- **www.linux.org.ru/books/HOWTO/SMB-HOWTO.html** — SMB-HOWTO (русский перевод).
- **www.samba.org** — официальный сайт проекта Samba.
- **www.webmin.com** — официальный сайт проекта Webmin.



Глава 28

Виртуальные частные сети

Виртуальные частные сети (Virtual Private Network, VPN) — общее наименование группы протоколов, программного, а иногда и аппаратного обеспечения для соединения независимых сетей через небезопасные (Интернет) сети общего пользования, прозрачно для пользователей сети. В связке с VPN часто употребляют термин "туннель" (туннелирование, tunneling), подразумевающий создание между двумя точками (обычно из разных локальных сетей, которые объединяют в VPN) шифрованного канала для безопасной пересылки данных из одного сегмента сети в другой.

Программное обеспечение может быть самым разнообразным, начиная от стандартных программ маршрутизации, PPP, SSH и заканчивая специализированными программно-аппаратными комплексами с аппаратурой шифрования и многоуровневыми протоколами проверки целостности и безопасности соединения.

В этой главе будут рассмотрены несколько вариантов организации VPN, но поскольку VPN характерно для корпоративного мира, то реализации предусматривают шифрование трафика между точками VPN и взаимодействие с операционными средами семейства Windows.

Протоколы, применяемые в реализациях VPN:

- SSH — позволяет быстро организовать защищенный канал для доступа к удаленным системам, отправки и получения файлов;
- IPSec (Internet Protocol Security) — реализован в проекте FreeS/WAN. Его обычно применяют для объединения фрагментов корпоративных сетей с помощью общедоступных сетей. При этом данные, передаваемые из одной сети в другую, шифруются исходящим шлюзом при передаче через сети общего пользования и преобразуются в исходный вид принимающим шлюзом;
- PPTP (Point-To-Point Tunneling Protocol) — считается устаревшим и не совсем безопасным. Тем не менее он используется Microsoft, поэтому распространен в корпоративной среде;
- CIPE — входит в большинство дистрибутивов и имеет поддержку на уровне ядра.

Организацию туннеля с помощью SSH в этой главе мы рассматривать не будем, поскольку SSH был описан ранее, а настройка маршрутизации не составляет особых проблем.

Протокол IPSec

IPSec — это расширение протокола IP, которое предоставляет не только шифрование, но и аутентификацию в транспортном слое. IPv6 поддерживает протокол IPSec изначально, т. к. IPSec является требованием спецификации IETF для протокола IPv6.

IPSec по сути — это набор из трех протоколов для инкапсуляции, шифрования и аутентификации: AH (Authentication Header, заголовок аутентификации), ESP (Encapsulating Security Payload, инкапсулированные защищенные данные) и IKE (Internet Key Exchange, обмен ключами через Интернет). IPSec прозрачен для пользователей и приложений, поэтому переделка программного обеспечения под него не требуется.

Протоколы AH и ESP отвечают за шифрование и аутентификацию. AH добавляется после заголовка IP, но перед данными. AH содержит аутентификационную информацию, обычно это ключи в MD5 (Message Digest, профиль сообщения) или SHA (Secure Hash Algorithm, алгоритм безопасного хэша). Протокол AH предназначен только для аутентификации и не предусматривает шифрования.

Протокол ESP может использоваться как для шифрования, так и для аутентификации. Он может применяться как с AH, так и без него. Ключи шифрования распределяются с помощью IKE. IKE устанавливает параметры соединения, включая инициализацию, обработку и обновление ключей шифрования. Аутентификация проводится на основе общих секретных фраз или криптографических RSA-ключей, которые гарантируют идентичность обеих частей. IKE основан на методе Diffie-Hellman обмена идентификационными метками (tokens). Шифрование данных осуществляют алгоритмы группового шифрования, такие как тройной DES (Data Encryption Standard, стандарт шифрования данных). Хэш-алгоритмы, такие как MD5 и SHA, обеспечивают аутентификацию каждого пакета. Для дополнительной безопасности соединения через малые промежутки времени происходит обновление аутентификационных ключей.

IPSec вставляет заголовок и зашифрованные полезные данные в обычный IP-пакет. Это позволяет данным IPSec проходить через любые IP-сети. IPSec реализован как в программном, так и в аппаратном виде. Хотя практически все реализации IPSec соответствуют существующим RFC, они совсем не обязательно смогут взаимодействовать. Необходимо тестировать реализации IPSec от различных производителей, чтобы убедиться, что они полностью совместимы.

VPN-сервер FreeS/WAN

Как уже упоминалось ранее, FreeS/WAN использует протокол IPSec, описанный в предыдущем разделе.

FreeS/WAN состоит из двух частей:

- KLIPS (KerneL IP Security) — дополнение стандартного ядра Linux;
- демон pluto — обрабатывает запросы аутентификации протокола IKE и взаимодействует в ядре с компонентом KLIPS, который отвечает за инкапсуляцию и шифрование.

Программа ipsec осуществляет работу протокола IPSec: активизирует и отключает туннели, а также опрашивает состояние каждого туннеля IPSec.

IPSec позволяет создавать защищенные туннели "хост–хост", "подсеть–подсеть" и "подсеть–хост". У IPSec есть два режима: *транспортный* и *туннельный*. Первый обеспечивает защиту только дейтаграмм (datagram) между источником и адресатом. Он производит аутентификацию, инкапсуляцию и шифрование только IP-данных, но оставляет неизменными транспортные заголовки. Поэтому транспортный режим обычно целесообразен для создания зашифрованных туннелей "хост–хост". В туннельном режиме создается новый IP-заголовок и вся оригинальная дейтаграмма инкапсулируется, скрывая информацию об оригинальном отправителе. Инкапсуляция позволяет пакетам из одной сети туннелировать через другие сети. Одно из применений — IPSec-шлюзы, соединяющие доверительные частные сети через общедоступные.

Конфигурирование FreeS/WAN осуществляется с помощью файлов ipsec.conf и ipsec.secrets.

Ipsec.conf

Файл ipsec.conf, как и все конфигурационные файлы, находится в каталоге /etc. Это текстовый файл, содержащий одну или более секций. Строка, которая начинается с символа #, — комментарий. Пример конфигурационного файла ipsec.conf приведен в листинге 28.1.

Листинг 28.1

```
#Определяем секцию для конфигурирования сетевых настроек
coning setup
# Сетевой интерфейс, используемый для организации
# VPN-соединения
interfaces="IPSEC0=eth0"
#Разрешение выдачи отладочных сообщений - в нашем случае -
# запрещаем выдачу сообщений для части ядра и для pluto
klipsdebug=none
plutodebug=none
# Автоматическая установка соединений и аутентификация при
# запуске IPSec

plutoload=%search
plutostart=%search
# Параметры соединения между локальными сетями
# Название соединения

conn Test

# Данные для 1-го шлюза
# IP адрес
```

```
left=193.1.1.2
# Описание локальной сети
leftsubnet=192.168.1.0/24
# IP ближайшего к 1-му шлюзу маршрутизатора
leftnexthop=194.17.2.5
# Исходные данные для 2-го шлюза
# IP адрес
right=197.11.0.213
# Описание локальной сети
rightsubnet=192.168.1.0/24
# IP ближайшего к 2-му шлюзу маршрутизатора
rightnexthop=202.22.8.24
# Количество попыток проверки ключей
# 0 - до достижения положительного результата
keyingtries=0
# Тип аутентификации (AH или ESP)
auth=ah
# Устанавливать соединение при запуске IPSec
auto=start
```

Разберем конфигурационный файл, приведенный в листинге 29.1.

- `config setup` — слово `config` задает тип секции, а `setup` — метка секции. Обычно тип секции — это `config`, определяющий системные настройки FreeS/WAN, или `conn`, задающий параметры каждого VPN-туннеля. Каждый VPN-туннель должен иметь собственную секцию `conn`.
- `interfaces="IPSEC0=eth0"` — задает интерфейсы, используемые для организации VPN-соединений.
- `klipsdebug=none` — разрешает/запрещает выдачу отладочных сообщений.

> ***Примечание***
>
> Значения параметров, начинающиеся с `%`, обозначают системную переменную, загружаемую FreeS/WAN при образовании туннеля.

Следующие параметры позволяют автоматически искать и устанавливать соединения при загрузке и старте демона:

- `plutoload=%search`;
- `plutostart=%search`.

Pluto сканирует файл ipsec.conf в поисках соединений, которые будут загружены, и выдает доступ к VPN-туннелям. Явно можно задать установку соединения следующим образом: `plutoload="Test"`, где `Test` — имя секции, в которой описан нужный нам туннель.

Далее идет секция `Test`, описывающая параметры нашей виртуальной сети, — ее левую и правую части. Тут все просто: IP-адрес шлюзов, маски подсети, адреса ближайших для данной "половинки" маршрутизаторов.

- `keyingtries=0` — говорит IKE бесконечно продолжать попытки обмена ключами при потере соединения.
- `keyexchange=ike` — устанавливает IKE как механизм по умолчанию для обмена ключами.
- `keylife=24h` — время между сменой ключей.
- `auth=ah` — тип аутентификации.
- `authby=secret` — метод аутентификации.

Ipsec.secrets

FreeS/WAN поддерживает два формата ключей, используемых демоном pluto для проверки подлинности соединений длиной до 256 бит. Каждый из этих форматов требует некоторых дополнительных действий по созданию ключей и изменению конфигурационных файлов. В случае открытых зашифрованных ключей для создания нового ключа на одном из шлюзов выполните команду

```
ipsec ranbits 256 > /root/key
```

Теперь файл /root/key содержит новый ключ, который необходимо прописать в файлы /etc/ipsec.secrets на обоих шлюзах, для чего добавьте в них следующую строку:

```
194.17.2.5 197.11.0.213↳
"0xaf4a2a4c_f58a444f_5a55d31e_55555ac4_555a58e2_b6ea25a3_0ee661d4_daf155"
```

Она представляет собой IP-адреса левой и правой части шлюза и сгенерированный нами ранее ключ.

В случае RSA-ключей необходимо выполнить следующие операции для файлов ipsec.conf и ipsec.secrets:

1. Создайте ключи RSA для каждого шлюза командой
   ```
   ipsec rsasigkey --verbose 1024 > /root/leftey
   ```
2. В файлах /etc/ipsec.conf на обоих шлюзах для обеспечения возможности работы RSA-ключей необходимо добавить следующие строки:
   ```
   authby=rsasig
   leftrsasigkey= ""
   rightrsasigkey= ""
   ```
   В опциях `leftrsasigkey` и `rightsasigkey` задают открытые RSA-ключи. А закрытые ключи прописывают в файле /etc/ipsec.secrets.

Более подробную информацию по алгоритмам шифрования и их применению в FreeS/WAN смотрите в документации к программе.

MS Windows NT VPN (PPTP)

Фирма Microsoft в своих продуктах для организации VPN использует протокол PPTP (Point-to-Point Tunneling Protocol). Особенности этого протокола мы не будем рассматривать, отметим лишь, что PPTP позволяет шифровать передаваемую информацию с ключом длиной 128 бит.

Итак, у нас возникают две задачи:

- организация PPTP-сервера на Linux для подключения Windows-клиентов;
- подключение Linux-клиентов к Windows NT VPN-серверу.

Linux PPTP-сервер

Устанавливаем pptpd, который входит в большинство современных дистрибутивов. Проверяем по ntsysv, что сервис pptpd запускается по умолчанию.

Вид конфигурационного файла /etc/pptpd.conf для большинства случаев приведен в листинге 28.2.

Листинг 28.2

```
speed 115200
option /etc/ppp/options.pptpd
debug
localip 192.168.0.1
remoteip 192.168.0.100-150
```

В этом файле мы задаем скорость соединения, файл, в котором хранятся настройки PPP для PPTP, разрешаем выдачу отладочной информации, определяем локальный адрес нашей VPN и диапазон адресов, который будет назначаться VPN-клиентам.

Содержимое файла /etc/ppp/options.pptpd иллюстрирует листинг 28.3.

Листинг 28.3

```
lock
mtu 1490
mru 1490
ipcp-accept-local
ipcp-accept-remote
lcp-echo-failure 3
lcp-echo-interval 5
deflate 0
auth
+chap
-pap
proxyarp
ms-dns 192.168.0.1
+chapms
+chapms-v2
nobsdcomp
nodeflate
nodefaultroute
+mppe-128
+mppe-stateless
```

В данном файле мы задаем размер пакета, тайм-ауты, типы авторизации и т. п. Особо следует отметить строку `+mppe-128`. Она разрешает 128-битовое шифрование.

Поле этого в файл /etc/ppp/chap-secrets заносим логины и пароли клиентов. Вот и все, можно работать.

Linux PPTP-клиент

PPTP-клиент реализует протокол PPTP, предназначенный для подключения систем с Linux к VPN-сетям через MS Windows NT VPN-сервер. Для этого необходимо установить PPTP Client (обычно в дистрибутивах пакет называется pptp-linux) и программное обеспечение, реализующее протокол MPPE (Microsoft Point-To-Point Encryption) и находящееся в пакете ppp-mppe.

После установки ppp-mppe в файле /etc/modules.conf должны быть следующие строки:

```
alias char-major-108 ppp_generic
alias ppp-compress-18 mppe
```

Далее запускаем файл /usr/sbin/pptp-command, выбираем пункт меню **Конфигурирование** и заполняем его поля (**IP-адрес**, **DNS**, **Имя системы**, **Тип авторизации** и т. п.).

Для запуска и останова VPN-соединения используется тот же файл pptp-command.

OpenVPN

Бесплатный проект. Официальный сайт — **openvpn.net**. Реализован под большинство UNIX-подобных операционных систем, включая MAC OS X и Windows.

Достаточно простой в настройке. Использует для шифрования пакет OpenSSL, для сжатия данных — LZO.

Ссылки

- **securitylab.ru/34649.html** — Разумов М. По материалам Sys Admin Magazine. Администрирование IPSec VPN под Linux. Часть первая.
- **securitylab.ru/34764.html** — Разумов М. По материалам Sys Admin Magazine. Администрирование IPSec VPN под Linux. Примеры конфигураций.
- **www.bruy.info/vpn.html** — Бруй В., Карлов С. Linux-сервер: пошаговые инструкции инсталляции и настройки.
- **www.freeswan.org** — официальный сайт FreeS/WAN.
- **www.multik.ru/linux/linuxvpn/** — Калошин Вячеслав. Установка VPN Linux-сервера.
- **www.opennet.ru/base/net/vpn_pptp.txt.html** — Коптев Д. Настройка VPN (PPTP) сервера под Linux.

Глава 29



Управление процессами

Данная глава посвящена процессам ОС Linux. Поскольку администрирование операционной системы в конечном счете сводится к управлению процессами, вполне логично выделить отдельную главу на описание столь важной темы.

Каждый раз, когда вы запускаете на выполнение программу, вы начинаете то, что в литературе именуется термином *процесс*. Или другими словами — процессом называется выполняемая в данный момент программа или ее потомки. Всякий процесс запускается от имени какого-то пользователя. Процессы, которые стартовали при загрузке, обычно выполняются от имени пользователей root или nobody.

Любой пользователь может управлять поведением процессов, им запущенных. При этом пользователь root может управлять всеми процессами — как запущенными от его имени, так и процессами, порожденными другими пользователями операционной системы. Управление процессами осуществляется с помощью утилит, а также при помощи некоторых команд командной оболочки (shell).

Каждый процесс в системе имеет уникальный номер — идентификационный номер процесса (Process Identification, PID). Этот номер используется ядром операционной системы, а также некоторыми утилитами для управления процессами.

Выполнение процесса на переднем плане и в фоновом режиме

Процессы могут выполняться на *переднем плане* (foreground) — режим по умолчанию и в *фоновом режиме* (background). На переднем плане в каждый момент для текущего терминала допускается только один процесс. Однако пользователь может перейти в другой виртуальный терминал и запустить на выполнение еще один процесс, а на другом терминале еще один и т. д. Процесс переднего плана — это процесс, с которым вы взаимодействуете, он получает информацию с клавиатуры (стандартный ввод) и посылает результаты на ваш экран (стандартный вывод).

Фоновый процесс после своего запуска благодаря специальной команде командной оболочки отключается от клавиатуры и экрана, т. е. не ожидает ввода данных со стандартного ввода и не выводит информацию на стандартный вывод, а командная оболочка не ожидает окончания запущенного процесса, что позволяет пользователю немедленно запустить еще один процесс.

Обычно фоновые процессы выполняются в течение длительного времени и не требуют вмешательства пользователя. К примеру, компиляция программ или архивирование большого объема информации — кандидаты номер один для перевода процесса в фоновый режим.

Процессы могут быть *отложенными*. Отложенный процесс в данный момент не выполняется и временно остановлен. После того как вы остановили процесс, в дальнейшем вы можете его продолжить как на переднем плане, так и в фоновом режиме. Возобновление приостановленного процесса не изменит его состояния — процесс начнется с того места, на котором был приостановлен.

Для выполнения программы в режиме переднего плана достаточно набрать ее имя в командной строке и запустить на выполнение. После этого вы можете работать с программой.

Для запуска программы в качестве фонового процесса следует набрать в командной строке ее имя и в конце добавить знак амперсанта (`&`), отделенный пробелом от имени программы, а также параметры командной строки, если таковые имеются. Затем программу запускают на выполнение. В отличие от запуска программы в режиме переднего плана мы получим приблизительно следующее сообщение:

```
/home/vasya# yes > /dev/null &;
[1] 123
/home/vasya#
```

Оно состоит из двух чисел и приглашения командной строки. Таким образом, программа работает в фоновом режиме и есть возможность запустить с той же самой консоли на выполнение еще какую-то программу.

Число `[1]` означает номер запущенного нами фонового процесса. Как вы узнаете несколько позже, с его помощью можно манипулировать нашим фоновым процессом. Значение `123` показывает идентификационный номер (PID) нашего процесса. Отличия этих двух чисел довольно существенные. Номер фонового процесса уникален *только* для пользователя, запускающего данный процесс. Иными словами, если у нас в системе три пользователя решили запустить фоновый процесс (первый для текущего сеанса) — в результате у каждого пользователя появится фоновый процесс с номером `[1]`. Напротив, идентификационный номер процесса (PID) уникален для всей операционной системы и однозначно идентифицирует в ней каждый процесс. Спрашивается, для чего тогда вводить нумерацию фонового процесса для пользователя? Для удобства. Номер фонового процесса хранится в переменных командной оболочки пользователя и позволяет не забивать голову цифрами типа 2693 или 1294, а использовать переменные вида `%1`, `%2`. Однако можно пользоваться и идентификационным номером процесса.

Для проверки состояния фоновых процессов можно воспользоваться командой командной оболочки — `jobs`:

```
/home/vasya# jobs
[1]+  Running                 yes >/dev/null  &
/home/vasya#
```

Из приведенного примера видно, что у пользователя в данный момент запущен один фоновый процесс, и он выполняется.

Остановка и возобновление процесса

Помимо прямого указания выполнять программу в фоновом режиме, существует еще один способ перевести процесс в фоновый режим. Для этого мы должны выполнить следующие действия:

1. Запустить процесс выполняться на переднем плане.
2. Остановить выполнение процесса.
3. Продолжить процесс в фоновом режиме.

Для выполнения программы введем ее имя в командной строке и запустим на выполнение. Для остановки выполнения программы необходимо нажать на клавиатуре следующую комбинацию клавиш — <Ctrl>+<Z>. После этого вы увидите на экране следующее:

```
/home/vasya# yes > /dev/null
ctrl+Z
[1]+  Stopped                 yes >/dev/null
/home/vasya#
```

Мы получили приглашение командной строки. Чтобы перевести процесс в фоновый режим, необходимо выполнить команду

```
bg %1
```

Причем необязательно делать это сразу после остановки процесса, главное правильно указать номер остановленного процесса.

Возврат процесса из фонового режима на передний план осуществляет команда

```
fg %1
```

Если вы хотите перевести программу в фоновый режим или, наоборот, на передний план выполнения сразу после остановки процесса, можно выполнить соответствующую программу *без* указания номера остановленного процесса.

Существует большая разница между фоновым и остановленным процессом. Остановленный процесс не выполняется и не потребляет ресурсы процесса, однако занимает оперативную память или пространство свопинга. В фоновом же режиме процесс продолжает выполняться.

Как остановить выполнение фонового процесса? Комбинация клавиш <Ctrl>+<Z> не поможет, поскольку процесс находится в фоновом режиме и не реагирует на ввод данных с консоли. Для решения этой проблемы следует переместить процесс на передний план, а затем остановить.

Завершение работы процесса

Ну вот, вы научились запускать и останавливать выполнение процессов, а также переводить исполняемый процесс в фоновый режим и в режим переднего плана. Однако вы, возможно, не умеете завершать работу процесса. Существуют три варианта.

- Если процесс интерактивный, как правило, в документации или прямо на экране написано, как корректно завершить программу.
- Если вы не знаете, как завершить текущий процесс (не фоновый), можно воспользоваться клавиатурной комбинацией <Ctrl>+<C>. Попробуйте также ком-

бинацию клавиш <Ctrl>+<Break>. А для остановки фонового процесса можно перевести его на передний план, а затем уже воспользоваться приведенными клавиатурными комбинациями.

- Самый действенный способ. Если вам не удалось прекратить выполнение процесса предыдущими способами — например, программа "зависла" или "слетел" терминал, то для завершения процесса можно воспользоваться командами `kill` и `killall`.

Команда `kill` может получать в качестве аргумента как номер процесса, так и его идентификационный номер (PID).

Таким образом, команда:

```
/home/vasya# kill 123
```

эквивалентна команде:

```
/home/vasya# kill %1
```

Очевидно, что при обращении к работе по идентификационному номеру (PID) процесса символ `%` не нужен.

С помощью команды `killall` можно прекратить выполнение нескольких процессов сразу, имеющих одно и то же имя. Например, команда `killall mc` прекратит работу всех программ mc, запущенных от имени данного пользователя.

Чтобы завершить работу процесса, вам нужно быть его владельцем. Это сделано в целях безопасности. Если бы одни пользователи могли завершать процессы других пользователей, открылась бы возможность исполнения в системе множества злонамеренных действий. Пользователь root может завершить работу любого процесса в операционной системе.

Программы для управления процессами

Существует достаточно много утилит, так или иначе предназначенных для управления процессами, исполняемыми в операционной системе. Здесь мы рассмотрим только основные утилиты (табл. 29.1).

Таблица 29.1. *Программы управления процессами*

Программа	Описание
at	Выполняет команды в определенное время
batch	Выполняет команды тогда, когда это позволяет загрузка системы
cron	Выполняет команды по заранее заданному расписанию
crontab	Позволяет работать с файлами crontab отдельных пользователей
kill	Прекращает выполнение процесса
nice	Изменяет приоритет процесса перед его запуском
nohup	Позволяет работать процессу после выхода пользователя из системы
ps	Выводит информацию о процессах
renice	Изменяет приоритет работающего процесса
w	Показывает, кто в настоящий момент работает в системе и с какими программами

nohup

Эта утилита дает возможность организовать фоновый процесс, продолжающий свою работу даже тогда, когда пользователь отключился от терминала, в отличие от команды &, которая этого не позволяет. Для организации фонового процесса необходимо выполнить команду

```
nohup выполняемая_фоновая_команда &
```

ps

Программа ps предназначена для получения информации о существующих в операционной системе процессах. У этой утилиты есть множество различных опций, но мы остановимся на самых необходимых. Для получения подробной информации смотрите man-страницу этой программы.

Простой запуск ps без параметров выдаст перечень программ, выполняемых на терминале. Обычно этот список очень мал:

```
PID TTY      TIME CMD
885 tty1     00:00:00 login
893 tty1     00:00:00 bash
955 tty1     00:00:00 ps
```

Что означает полученная информация?

- Первый столбец — PID (идентификационный номер процесса). Как уже упоминалось, каждый выполняющийся процесс в системе получает уникальный идентификатор, с помощью которого осуществляется управление процессом. Каждому вновь запускаемому на выполнение процессу присваивается последующий свободный PID. Когда процесс завершается, его номер освобождается. Когда достигнут максимальный PID, следующий свободный номер будет взят из наименьшего освобожденного.
- Следующий столбец — TTY — терминал, на котором выполняется процесс. Запуск команды без параметров ps отобразит процессы, выполняемые на текущем терминале.
- Столбец TIME — показывает, сколько процессорного времени выполняется процесс. Оно не является фактическим временем с момента запуска процесса, поскольку Linux — это многозадачная операционная система. Информация, указанная в столбце TIME, соответствует времени, реально потраченному процессором на выполнение процесса.
- Столбец CMD — показывает, что же это за программа. Отображается только имя программы, опции командной строки не выводятся.

Для получения расширенного списка процессов, выполняемых в системе, используется команда

```
ps -ax
```

Листинг 29.1 иллюстрирует результат.

Листинг 29.1

```
  PID TTY      STAT   TIME COMMAND
    1 ?        S      0:04 init
    2 ?        SW     0:00 [keventd]
    3 ?        SW     0:00 [kapm-idled]
    4 ?        SWN    0:00 [ksoftirqd_CPU0]
    5 ?        SW     0:00 [kswapd]
    6 ?        SW     0:00 [kreclaimd]
    7 ?        SW     0:00 [bdflush]
    8 ?        SW     0:00 [kupdated]
    9 ?        SW<    0:00 [mdrecoveryd]
   13 ?        SW     0:00 [kjournald]
  437 ?        S      0:00 syslogd -m 0
  442 ?        S      0:00 klogd -2
  462 ?        S      0:00 portmap
  490 ?        S      0:00 rpc.statd
  647 ?        S      0:00 /usr/sbin/sshd
  704 ?        S      0:00 lpd Waiting
  732 ?        S      0:00 sendmail: accepting connections
  751 ?        S      0:00 gpm -t ps/2 -m /dev/mouse
  769 ?        S      0:00 crond
  835 ?        S      0:00 xfs -droppriv -daemon
  853 ?        S      0:00 anacron
  871 ?        S      0:00 /usr/sbin/atd
  885 tty1     S      0:00 login -- root
  886 tty2     S      0:00 /sbin/mingetty tty2
  887 tty3     S      0:00 /sbin/mingetty tty3
  888 tty4     S      0:00 /sbin/mingetty tty4
  889 tty5     S      0:00 /sbin/mingetty tty5
  890 tty6     S      0:00 /sbin/mingetty tty6
  893 tty1     S      0:00 -bash
 1037 tty1     R      0:00 /usr/bin/mc -P
 1038 ?        S      0:00 cons.saver /dev/tty1
 1039 pts/0    S      0:00 bash -rcfile .bashrc
 1067 pts/0    R      0:00 ps -ax
```

Как можно видеть, список запущенных процессов в системе велик и сильно зависит от конфигурации операционной системы. Параметры, заданные программе в этом примере, заставляют ее выводить не только имена программ, но и список опций, с которыми были запущены программы.

Появился новый столбец — STAT. В нем отображается состояние (status) процесса. Полный список состояний вы можете прочитать в описании программы ps, а здесь перечислим самые важные состояния:

- R — запущенный процесс, исполняющийся в данный момент времени;
- S — спящий (sleeping) процесс; ожидает какое-то событие, необходимое для его активизации;
- Z — "зомбированные" процессы (zombied), родительский процесс которых прекратил свое существование, оставив дочерние процессы рабочими.

Помимо этого позвольте обратить ваше внимание на столбец TTY. Как вы, наверное, заметили, многие процессы, расположенные в верхней части таблицы, содержат знак ? вместо терминала. Так обозначаются процессы, запущенные с более не активного терминала. Как правило, это всякие системные сервисы.

Если вы хотите увидеть еще больше информации о выполняемых процессах, попробуйте выполнить команду

```
ps -aux
```

Листинг 29.2 иллюстрирует результат.

Листинг 29.2

```
USER     PID %CPU %MEM    VSZ   RSS TTY   STAT START TIME COMMAND
root       1  1.2  0.2   1412   520 ?     S    14:51 0:04 init
root       2  0.0  0.0      0     0 ?     SW   14:51 0:00 [keventd]
root       3  0.0  0.0      0     0 ?     SW   14:51 0:00 [kapm-idled]
root       4  0.0  0.0      0     0 ?     SWN  14:51 0:00 [ksoftirqd_CPU0]
root       5  0.0  0.0      0     0 ?     SW   14:51 0:00 [kswapd]
root       6  0.0  0.0      0     0 ?     SW   14:51 0:00 [kreclaimd]
root       7  0.0  0.0      0     0 ?     SW   14:51 0:00 [bdflush]
root       8  0.0  0.0      0     0 ?     SW   14:51 0:00 [kupdated]
root       9  0.0  0.0      0     0 ?     SW<  14:51 0:00 [mdrecoveryd]
root      13  0.0  0.0      0     0 ?     SW   14:51 0:00 [kjournald]
root     437  0.0  0.2   1472   592 ?     S    14:52 0:00 syslogd -m 0
root     442  0.0  0.4   1928  1040 ?     S    14:52 0:00 klogd -2
rpc      462  0.0  0.2   1552   588 ?     S    14:52 0:00 portmap
rpcuser  490  0.0  0.2   1596   756 ?     S    14:52 0:00 rpc.statd
root     590  0.0  0.2   1396   524 ?     S    14:52 0:00 /usr/sbin/apmd -p
root     647  0.0  0.4   2676  1268 ?     S    14:52 0:00 /usr/sbin/sshd
root     680  0.0  0.3   2264   992 ?     S    14:52 0:00 xinetd -stayalive
lp       704  0.0  0.3   2600  1020 ?     S    14:52 0:00 lpd Waiting
root     732  0.0  0.7   5296  1984 ?     S    14:52 0:00 sendmail: accepti
root     751  0.0  0.1   1440   492 ?     S    14:52 0:00 gpm -t ps/2 -m /d
root     769  0.0  0.2   1584   660 ?     S    14:52 0:00 crond
xfs      835  0.0  1.4   4988  3612 ?     S    14:52 0:00 xfs -droppriv -da
root     853  0.0  0.2   1416   600 ?     S    14:52 0:00 anacron
```

```
daemon   871  0.0  0.2  1444   568 ?     S    14:52 0:00 /usr/sbin/atd
root     885  0.0  0.4  2320 1076 tty1  S    14:52 0:00 login -- root
root     886  0.0  0.1  1384   448 tty2  S    14:52 0:00 /sbin/mingetty tt
root     887  0.0  0.1  1384   448 tty3  S    14:52 0:00 /sbin/mingetty tt
root     893  0.0  0.5  2464 1312 tty1  S    14:52 0:00 -bash
root    1037  0.0  0.7  3284 1804 tty1  R    14:56 0:00 /usr/bin/mc -P
root    1038  0.0  0.1  1380   348 ?     S    14:56 0:00 cons.saver /dev/t
root    1039  0.0  0.5  2552 1392 pts/0 S    14:56 0:00 bash -rcfile .bas
root    1068  0.0  0.3  2780   824 pts/0 R    14:57 0:00 ps -aux
```

Как вы видите — информации прибавилось. Появились дополнительные столбцы:

- `USER` — показывает, от имени какого пользователя был запущен данный процесс;
- `%CPU`, `%MEM` — процессорное время и оперативная память, выделенные данному процессу;
- `TIME` — время запуска программы.

В табл. 29.2 приведены некоторые параметры командной строки программы ps.

Таблица 29.2. *Параметры командной строки программы ps*

Ключ	Описание
a	Показать процессы всех пользователей
c	Имя команды из переменной среды
e	Показать окружение
f	Показать процессы и подпроцессы
h	Вывод без заголовка
j	Формат заданий
l	"Длинный" формат вывода
m	Вывод информации о памяти
n	Числовой вывод информации
r	Только работающие процессы
s	Формат сигналов
S	Добавить время использования процессора порожденными процессами
txx	Только процессы, связанные с терминалом `xx`
u	Формат вывода с указанием пользователя
v	Формат виртуальной памяти
w	Вывод без обрезки информации для размещения в одной строке
x	Показать процессы без контролирующего терминала

Программа ps обладает большим списком возможностей, имеет много ключей запуска и выводит много информации, однако для обычной работы будет достаточно приведенных здесь сведений.

top

Еще одна утилита, с помощью которой можно получать информацию о запущенных в операционной системе процессах. Для активизации нужно просто выполнить команду `top`. Эта утилита выводит на экран список процессов в системе, отсортированных в порядке убывания значений используемых ресурсов (листинг 29.3).

Листинг 29.3

```
2:55pm  up 3 min,  1 user,  load average: 0,06, 0,09, 0,03
32 processes: 31 sleeping, 1 running, 0 zombie, 0 stopped
CPU states:  1,1% user,  2,9% system,  0,0% nice, 95,8% idle
Mem:   255532K av,   42856K used,  212676K free,    0K shrd,   8560K buff
Swap:  257000K av,       0K used,  257000K free              19920K cached

  PID USER   PRI  NI  SIZE  RSS SHARE STAT %CPU %MEM  TIME COMMAND
    1 root     8   0   520  520   452 S     0,0  0,2  0:04 init
    2 root     9   0     0    0     0 SW    0,0  0,0  0:00 keventd
    3 root     9   0     0    0     0 SW    0,0  0,0  0:00 kapm-idled
    4 root    19  19     0    0     0 SWN   0,0  0,0  0:00 ksoftirqd_CPU0
    5 root     9   0     0    0     0 SW    0,0  0,0  0:00 kswapd
    6 root     9   0     0    0     0 SW    0,0  0,0  0:00 kreclaimd
    7 root     9   0     0    0     0 SW    0,0  0,0  0:00 bdflush
    8 root     9   0     0    0     0 SW    0,0  0,0  0:00 kupdated
    9 root    -1 -20     0    0     0 SW<   0,0  0,0  0:00 mdrecoveryd
   13 root     9   0     0    0     0 SW    0,0  0,0  0:00 kjournald
  437 root     9   0   592  592   496 S     0,0  0,2  0:00 syslogd
  442 root     9   0  1040 1040   448 S     0,0  0,4  0:00 klogd
  462 rpc      9   0   588  588   504 S     0,0  0,2  0:00 portmap
  490 rpcuser  9   0   756  756   660 S     0,0  0,2  0:00 rpc.statd
  590 root     8   0   524  524   464 S     0,0  0,2  0:00 apmd
  647 root     9   0  1268 1268  1076 S     0,0  0,4  0:00 sshd
  680 root     9   0  1008  992   816 S     0,0  0,3  0:00 xinetd
  704 lp       9   0  1020 1020   872 S     0,0  0,3  0:00 lpd
```

Сначала идет общесистемная информация: время запуска ОС, время ее работы от момента последнего перезапуска системы, количество зарегистрированных в данный момент в операционной системе пользователей, а также минимальная, максимальная и средняя загрузка ОС. Помимо этого, отображается общее число

процессов и их состояние, сколько процентов ресурсов системы занимают пользовательские процессы и системные процессы, использование оперативной памяти и свопа.

Далее идет таблица, во многом напоминающая вывод программы ps.

Здесь для процесса указаны:

- идентификационный номер;
- имя пользователя (владельца процесса);
- приоритет;
- размер;
- состояние процесса;
- потребляемая оперативная память;
- ресурс центрального процесса,
- время выполнения;
- имя процесса.

Утилита top после запуска периодически обновляет информацию о состоянии процессов в операционной системе, что позволяет нам динамически получать информацию о загрузке системы.

kill

Программа kill (в переводе с английского — убить) предназначена для посылки соответствующих сигналов указанному нами процессу. Как правило, это бывает тогда, когда некоторые процессы начинают вести себя неадекватно. Наиболее часто программа применяется, чтобы прекратить выполнение процессов.

Для того чтобы прекратить работу процесса, необходимо знать PID процесса либо его имя. Например, чтобы "убить" процесс 123, достаточно выполнить команду

```
kill 123
```

Как обычно, чтобы прекратить работу процесса, вам необходимо быть его владельцем. Само собой, пользователь root может прекратить работу любого процесса в системе.

Иногда обычное выполнение программы kill не справляется с поставленной задачей. Как правило, это объясняется тем, что данный процесс завис либо выполняет операцию, которую с его точки зрения нельзя прервать немедленно. Для прерывания такого процесса можно воспользоваться командой

```
kill -9 123
```

Что это означает? Вообще-то программа kill предназначена для посылки процессам управляющих сигналов, в том числе сигнала `SIGTERM` (terminate, завершиться). Этот сигнал посылается процессу при выполнении программы kill по умолчанию. Процесс, получивший данный сигнал, должен корректно завершить свою работу (закрыть используемые файлы, сбросить буферы ввода/вывода и т. п.). Ключ `-9` указывает программе kill посылать процессу другой тип сигнала — `SIGKILL`. Это приводит не к корректному завершению, а к немедленному прекращению жизнедеятельности процесса. Полный перечень сигналов (листинг 29.4) можно получить, выполнив команду

```
kill -l
```

Листинг 29.4

```
 1) SIGHUP       2) SIGINT       3) SIGQUIT      4) SIGILL
 5) SIGTRAP      6) SIGABRT      7) SIGBUS       8) SIGFPE
 9) SIGKILL     10) SIGUSR1     11) SIGSEGV     12) SIGUSR2
13) SIGPIPE     14) SIGALRM     15) SIGTERM     17) SIGCHLD
18) SIGCONT     19) SIGSTOP     20) SIGTSTP     21) SIGTTIN
22) SIGTTOU     23) SIGURG      24) SIGXCPU     25) SIGXFSZ
26) SIGVTALRM   27) SIGPROF     28) SIGWINCH    29) SIGIO
30) SIGPWR      31) SIGSYS      32) SIGRTMIN    33) SIGRTMIN+1
34) SIGRTMIN+2  35) SIGRTMIN+3  36) SIGRTMIN+4  37) SIGRTMIN+5
38) SIGRTMIN+6  39) SIGRTMIN+7  40) SIGRTMIN+8  41) SIGRTMIN+9
42) SIGRTMIN+10 43) SIGRTMIN+11 44) SIGRTMIN+12 45) SIGRTMIN+13
46) SIGRTMIN+14 47) SIGRTMIN+15 48) SIGRTMAX-15 49) SIGRTMAX-14
50) SIGRTMAX-13 51) SIGRTMAX-12 52) SIGRTMAX-11 53) SIGRTMAX-10
54) SIGRTMAX-9  55) SIGRTMAX-8  56) SIGRTMAX-7  57) SIGRTMAX-6
58) SIGRTMAX-5  59) SIGRTMAX-4  60) SIGRTMAX-3  61) SIGRTMAX-2
62) SIGRTMAX-1  63) SIGRTMAX
```

Как видите, список внушительный. Подробную информацию о сигналах вы найдете в документации на программу kill.

killall

Еще один вариант программы kill. Используется для того, чтобы завершить работу процессов, носящих одно и то же имя. К примеру, в нашей системе запущено несколько программ mc. Для того чтобы одновременно завершить работу этих программ, достаточно выполнить команду

```
killall mc
```

Конечно, этим не ограничиваются возможности данной команды. С ее помощью можно отсылать сигналы группе одноименных процессов. Для получения более подробной информации по этой команде обращайтесь к ее man-странице.

Изменение приоритета выполнения процессов

В ОС Linux у каждого процесса есть свой приоритет исполнения. Это очень удобно. Поскольку операционная система многозадачная, то для выполнения каждого процесса выделяется определенный интервал времени. Для одних задач необходимо выделить побольше, для других можно поменьше. Для этого и предназначен приоритет процесса. Приоритетом процесса управляют программы nice и renice.

nice

Программа nice позволяет запустить команду с предопределенным приоритетом выполнения, который задается в командной строке. При обычном запуске все задачи имеют один и тот же приоритет, и операционная система равномерно распределяет между ними процессорное время. Однако с помощью утилиты nice можно понизить приоритет какой-либо задачи, таким образом предоставляя другим процессам больше процессорного времени. Повысить приоритет той или иной задачи имеет право только пользователь root. Синтаксис nice следующий:

```
nice -number command
```

Уровень приоритета процесса определяется параметром *number*, при этом большее его значение означает меньший приоритет процесса. Значение по умолчанию — 10, и *number* представляет собой число, на которое должен быть уменьшен приоритет.

К примеру, процесс `top` имеет приоритет, равный –5. Для того чтобы понизить приоритет выполнения процесса на десять, мы должны выполнить следующую команду:

```
nice 10 top
```

В результате процесс `top` получит приоритет, равный 5.

Только пользователь root может поднять приоритет того или иного процесса, задав *отрицательное* значение параметра *number*.

renice

Программа renice, в отличие от nice, позволяет изменить приоритет уже работающего процесса. Формат запуска программы:

```
renice -number PID
```

В общем, renice и nice работают аналогично. Уровень приоритета процесса определяется параметром *number*, при этом большее его значение соответствует меньшему приоритету. Значение по умолчанию — 10, и *number* представляет собой число, на которое должен быть уменьшен приоритет процесса.

Только пользователь root может поднять приоритет того или иного процесса, указав *отрицательное* значение параметра *number*.

Выполнение процессов в заданное время

Одна из основных задач автоматизации администрирования операционной системы — выполнение программ в определенное время или с заданной периодичностью. Конечно, можно запускать программы самостоятельно, но проводить 24 часа на работе или постоянно удаленно запускать программы в самое неподходящее время (часа в три ночи) — безумие. Для решения этих проблем существует несколько утилит, позволяющих запускать процессы в нужное время.

at

Запустить одну или более команд в заранее определенное время позволяет команда `at`, которой вы можете определить время и дату запуска той или иной программы. Команда `at` требует задания, по меньшей мере, двух параметров: времени выполнения и имени запускаемой программы с ее параметрами запуска.

Приведенный далее пример запустит команду на выполнение в 01:01. Для этого введите все указанные строки с терминала, завершая ввод каждой из них нажатием клавиши <Enter> и по окончании ввода всей команды — <Ctrl>+<D> для ее завершения:

```
at 1:01
ls
echo "Time is 1:01"
```

Помимо времени, в команде `at` можно также определить и дату запуска программы на выполнение.

Пользователь root может без ограничения применять практически любые команды. Для обычных пользователей права доступа к команде `at` определяются файлами /etc/at.allow и /etc/at.deny. В файле /etc/at.allow содержится список тех, кому разрешена команда `at`, а в файле /etc/at.deny находится список тех, кому она запрещена.

batch

Команда `batch` в принципе аналогична `at`. Более того, `batch` представляет собой псевдоним команды `at -b`. Для чего необходима эта команда? Представьте, вы хотите запустить резервное копирование вечером. Однако в это время система очень занята, и выполнение резервирования системы практически парализует ее работу. Для этого и существует команда `batch` — она позволяет операционной системе самой решить, когда наступает подходящий момент для запуска задачи в то время, когда система не сильно загружена.

Формат команды `batch` представляет собой просто список команд для выполнения, следующих в строках за командой; заканчивается список комбинацией клавиш <Ctrl>+<D>. Можно также поместить список команд в файл и перенаправить его на стандартный ввод команды `batch`.

cron

Cron — это программа, выполняющая задания по расписанию, но, в отличие от команды `at`, она позволяет повторять задания неоднократно. Вы определяете времена и даты, когда должна запускаться та или иная программа. Времена и даты могут определяться в минутах, часах, днях месяца, месяцах года и днях недели.

Программа cron запускается один раз при загрузке системы. При запуске cron проверяет очередь заданий `at` и задания пользователей в файлах crontab. Если для запуска не было найдено заданий, то следующую проверку cron проведет через минуту.

Список задач для программы cron формирует команда `crontab`. Для каждого пользователя с помощью этой команды создается его собственный crontab-файл со списком заданий, имеющий то же имя, что и имя пользователя.

Каждая строка в файле crontab содержит шаблон времени и команду. Команда выполняется тогда, когда текущее время соответствует приведенному шаблону. Шаблон состоит из пяти частей, разделенных пробелами или символами табуляции, и имеет вид:

```
минуты часы день_месяца месяц день_недели задание
```

Первые пять полей представляют собой шаблон времени и обязательно должны присутствовать в файле. Для того чтобы программа cron игнорировала поле шаблона времени, поставьте в нем символ звездочки (`*`).

Например, шаблон `10 01 01 * *` говорит о том, что команда должна быть запущена в десять минут второго каждого первого числа любого (`*`) месяца, каким бы днем недели оно ни было. В табл. 29.3 приведены поля таблицы задания cron.

Таблица 29.3. *Параметры таблицы заданий программы cron*

Поле	Описание
`минуты`	Минуты в течение часа. Значения от 0 до 59
`часы`	Час запуска задания. Значения от 0 до 23, где 0 — полночь
`день_месяца`	День месяца, в который должна исполняться команда
`месяц`	Месяц, в который необходимо запускать задание. Значения лежат в пределах от 1 до 12, где 1 — январь
`день_недели`	День недели в виде цифр от 0 до 7 (0 и 7 означают воскресенье) или первых трех букв, например Mon
`задание`	Командная строка для запуска задания

Приведем несколько примеров команд, запускаемых программой cron:

- в первую минуту каждого часа:
  ```
  01 * * * * /usr/bin/script
  ```
- каждый день в 8:20:
  ```
  20 8 * * *  /usr/bin/script
  ```
- в 6 часов каждое воскресенье:
  ```
  00 6 * * 0 /usr/bin/script
  ```
- в 7:40 каждое первое число месяца:
  ```
  40 7 1 * * /usr/bin/script
  ```

Для создания и редактирования файла заданий для программы cron служит команда `crontab`. Прямое редактирование файла заданий не допускается.

Параметры командной строки `crontab`:

- `-e` — позволяет редактировать компоненты файла (при этом вызывается редактор, определенный в переменной `EDITOR`);
- `-r` — удаляет текущий crontab-файл из каталога;
- `-l` — выводит список текущих заданий.

Можно разрешать или запрещать конкретным пользователям использование cron. Для этого существуют файлы /etc/cron.allow и /etc/cron.deny, которые аналогичны описанным ранее /etc/at.allow и /etc/at.deny.

Ссылки

- **www.tts.esoo.ru/~lesenka/linux/slack_book.html** — Дэвид Кэнтрелл, Логэн Джонсон, Крис Люменс. Основы Slackware Linux. Официальный учебник.

Глава 30



Администрирование сети

Пожалуй, одна из самых сложных и трудоемких задач системного администратора — администрирование сети. Эта задача настолько комплексная, что можно практически все, о чем писалось ранее, отнести к подготовке администрирования сети. Слишком много параметров, программ, настроек могут прямо или косвенно отражаться на функционировании сети и сетевых сервисов.

В этой главе все, так или иначе, будет касаться администрирования и управления сетью, хотя некоторые вещи с первого взгляда никоим образом не относятся к сети или ее настройке.

В той части главы, где будет говориться об инструментах, предназначенных для обнаружения уязвимости системы, опишем несколько программных пакетов, которые с одинаковым успехом можно применить как для взлома системы, так и для ее защиты.

Расширенное управление доступом к файлам

К сожалению, стандартные средства организации прав доступа к файлам в UNIX-подобных операционных системах зачастую не удовлетворяют требованиям некоторых системных администраторов. Проблема заключается в том, что определение прав доступа к файлам сводится к установке девяти битов, с помощью которых можно задать права доступа для владельца файла, группы, к которой принадлежит владелец файла, а также для всех остальных. Часто необходима достаточно сложная настройка доступа к файлу: допустим, три человека из трех разных групп имеют право делать с файлом все, что угодно, десять человек из других групп могут открывать файл для чтения, а еще десять — только выполнять. Для всех других пользователей доступ к этому файлу необходимо запретить. Устроить нечто подобное стандартными средствами Linux — весьма нетривиальная задача. В такой ситуации для решения данной проблемы можно воспользоваться Linux ACLs (Access Control Lists, списки контроля доступа) — версией POSIX ACLs для Linux. Linux ACLs — это набор патчей для ядра операционной системы и программ для работы с файловой системой, а также несколько утилит, дающих возможность устанавливать права доступа к файлам не только для пользователя-владельца и группы-владельца файла, но и для любого пользователя или группы.

Linux ACLs использует расширенные атрибуты (Extended Attributes) для хранения данных о правах доступа к файлам пользователей и групп. Расширенные атрибуты — это пара имя/значение, привязанная к определенному файлу.

Список расширенного контроля доступа существует для каждого inode и состоит из шести компонентов. Первые три являются копией стандартных прав доступа к файлу. Они содержатся в единственном экземпляре в ACL и есть у каждого файла в системе:

- ACL_USER_OBJ — режим доступа к файлу пользователя-владельца;
- ACL_GROUP_OBJ — режим доступа к файлу группы-владельца;
- ACL_OTHER — режим доступа к файлу остальных пользователей.

Следующие два компонента устанавливаются для каждого файла в отдельности и могут присутствовать в ACL в нескольких экземплярах:

- ACL_USER — содержит UID и режим доступа к файлу пользователя, которому установлены права, отличные от основных. На каждого пользователя со своими правами на данный файл хранится отдельная запись. Не может существовать более одной записи на одного и того же пользователя;
- ACL_GROUP — то же самое, что и ACL_USER, но для группы пользователей;
- ACL_MASK — маска действующих прав доступа для расширенного режима.

При установке дополнительных прав доступа присваивается значение и элементу ACL_MASK.

Каталоги также могут иметь список контроля доступа по умолчанию. В отличие от основного ACL, он действует на создаваемые внутри данного каталога файлы и каталоги. При создании файла внутри такого каталога файл получает ACL, равный ACL, по умолчанию этого каталога.

Установка и изменение прав доступа

Две утилиты — getfacl и setfacl — управляют списками контроля доступа.

С помощью getfacl можно просмотреть текущие параметры доступа любого файла. Листинг 30.1 иллюстрирует результат вызова getfacl для домашнего каталога пользователя vasya.

Листинг 30.1

```
getfacl  /home/vasya
file: home/vasya
owner: vasya
group: users
user::rwx
group::---
other:---
```

Как можно видеть, каталог /home/vasya принадлежит пользователю vasya, группе users и значение прав доступа к каталогу — 0700. Каталог имеет только основ-

ные параметры доступа, поскольку изначально дополнительные права не устанавливаются.

Дополнительные права доступа к файлу устанавливаются и изменяются при помощи утилиты setfacl. Формат вызова следующий:

```
setfacl -опции ACL_структура, ACL_структура,...,ACL_структура имя_файла
имя_файла ...
```

ACL-структура представляет собой одну из конструкций:

- `[d:][u:][пользователь] [:[+|^]режимы_доступа]` — режим доступа к файлу или каталогу пользователя. Если пользователь не указан, определяет режим доступа пользователя-владельца;
- `[d:] g:[группа] [:[+|^]режимы_доступа]` — то же, что и предыдущая конструкция, но для группы;
- `[d:] m [:[+|^] режимы_доступа]` — действующие права доступа;
- `[d:] o [:[+|^] режимы_доступа]` — режим доступа для остальных пользователей.

Опции для установки и изменения ACL:

- `-s` — заменяет полностью ACL-файл на указанный в командной строке;
- `-m` — изменяет режимы доступа к файлу (каталогу);
- `-x` — убирает правила доступа из ACL.

Листинг 30.2 содержит результат применения setfacl к каталогу vasya.

Листинг 30.2

```
setfacl -s u::rwx,g::---,o:---,u:us1:rwx,g:usrs2:rx,u:us2:--- /home/vasya
getfacl /home/dh

file: home/vasya
owner: vasya
group: users
user::rwx
user:us1:rwx
user:us2:---
group::---
group:usrs2:r-x
mask:rwx
other:---
```

Дополнительные возможности

Кроме основных опций запуска, обе команды имеют много дополнительных. Мы не будем останавливаться на этих возможностях, поскольку пакет динамично изменяется, и вполне вероятно, что он уже обладает существенно бóльшими возможностями по сравнению с теми, которые присутствовали на момент написания книги. Мы не будем этого делать и потому, что пакет управления правами доступа

вряд ли понадобится обычному пользователю или администратору небольшой локальной сети, а администратор большой фирмы должен быть в состоянии самостоятельно разобраться в возможностях любого программного пакета.

Шифрование трафика

Традиции — вещь очень неоднозначная. Иногда они помогают жить и успешно развиваться, иногда они просто странные или бесполезные, а иногда — весьма вредны. То же самое можно сказать и о сетевых протоколах — традициях компьютерного мира.

Бóльшая часть существующих сетевых протоколов разрабатывалась по компьютерным меркам в чуть ли не в доисторическую эпоху рыцарских традиций, когда о сетевых взломах и сетевом шпионаже можно было прочитать только в научной фантастике. В результате подавляющее большинство данных в Интернете передаются в открытом виде. Существует и обратная сторона медали — множество утилит для прослушивания сетевого трафика, умеющих анализировать перехватываемые данные. С помощью таких утилит можно получить пароли пользователей для различных сетевых служб, тексты электронных писем, файлы, сообщения, переданные по ICQ, и т. д. Защитить себя от такого прослушивания можно с помощью шифрования трафика.

Наиболее распространенный протокол шифрования — SSL (Secure Sockets Layer). Чаще всего он используется для шифровки протокола HTTP (HTTPS), но также может применяться для создания защищенных соединений с SMTP, POP3, IMAP и другими высокоуровневыми сетевыми протоколами.

Программа, осуществляющая поддержку протокола SSL почти для любых серверных и клиентских приложений под Linux и Windows, называется Stunnel. Основное ее применение состоит в создании надежного шифрованного канала между двумя и более хостами в сетях, где существует угроза прослушивания трафика.

Stunnel

Как обычно, рекомендуется получить с сайта разработчика последнюю версию программного пакета.

Установка

Для работы Stunnel необходим OpenSSL. Обычно OpenSSL устанавливается при инсталляции операционной системы Linux (по крайней мере, в дистрибутиве Red Hat Linux), поэтому проблем с установкой OpenSSL возникнуть не должно. Пакет Stunnel также входит в состав дистрибутива в виде RPM-пакета.

Организация шифрованного туннеля

Stunnel может работать в двух режимах — сервера и клиента. В качестве сервера Stunnel открывает указанный порт, дешифрует все поступившие данные и передает их либо в указанную в параметрах запуска программу, либо на заданный порт

указанного хоста. В качестве клиента Stunnel открывает указанный порт, шифрует все поступившие на него данные и передает их в определенную программу или на определенный порт на заданном хосте.

Давайте организуем защищенное telnet-соединение (хотя это и не имеет практической пользы, поскольку есть SSH) между двумя компьютерами (А и Б).

На компьютере Б запускаем Stunnel в режиме сервера:

```
stunnel -d 999 -r 23
```

Опция `-d` указывает Stunnel работать в режиме отдельного демона, ждущего соединения по порту 999. Все данные, полученные в шифрованном виде на порт 999, в открытом виде передаются на порт 23 на локальной машине.

Затем на компьютере А запускаем Stunnel в режиме клиента:

```
stunnel -c -d 1055 -r B:999
```

Опция `-c` указывает на работу в режиме клиента, все данные, полученные в открытом виде на порт 1055, передаются в шифрованном виде на порт 999 на хосте Б.

После проделанных манипуляций можно устанавливать telnet-соединение с компьютером Б. Команда запуска telnet на компьютере А будет выглядеть следующим образом:

```
telnet localhost 1055
```

Несколько непривычно, зато трафик полностью шифруется. Точно по такому же принципу организовывается шифрованный туннель и для других сетевых протоколов.

Stunnel и приложения, поддерживающие SSL

Достаточно часто возникает ситуация, когда одно из приложений поддерживает протокол SSL, а приложение с другой стороны не поддерживает протокол SSL. В этом случае Stunnel можно запускать только с одной стороны — там, где приложение не способно поддерживать протокол SSL. Но тогда возникает проблема — какие порты используются приложением, поддерживающим протокол SSL.

Листинг 30.3 содержит официальный список SSL-портов.

Листинг 30.3

```
https          443/tcp   # http protocol over TLS/SSL
smtps          465/tcp   # smtp protocol over TLS/SSL (was ssmtp)
nntps          563/tcp   # nntp protocol over TLS/SSL (was snntp)
imap4-ssl      585/tcp   # IMAP4+SSL (use 993 instead)
sshell         614/tcp   # SSLshell
ldaps          636/tcp   # ldap protocol over TLS/SSL (was sldap)
ftps-data      989/tcp   # ftp protocol, data, over TLS/SSL
ftps           990/tcp   # ftp protocol, control, over TLS/SSL
telnets        992/tcp   # telnet protocol over TLS/SSL
imaps          993/tcp   # imap4 protocol over TLS/SSL
ircs           994/tcp   # irc protocol over TLS/SSL
pop3s          995/tcp   # pop3 protocol over TLS/SSL (was spop3)
```

Сертификаты

Программа Stunnel позволяет проверять подлинность сертификатов тех хостов, к которым (или с которых) идет подключение. Для этого предназначена опция командной строки `-v`. После `-v` необходимо указать уровень проверки сертификата:

- `0` — наличие и подлинность сертификата никак не проверяются;
- `1` — сертификат проверяется на подлинность, если присутствует. Если сертификат не являются подлинным, то соединение не устанавливается;
- `2` — проверяются присутствие сертификата и его подлинность. Если сертификат отсутствует или не является подлинным, то соединение не устанавливается;
- `3` — проверяются присутствие сертификата и его наличие в списке проверенных сертификатов. Если сертификат отсутствует или его нет в списке проверенных сертификатов, то соединение не устанавливается.

Сертификат создается при сборке пакета и помещается вместе с секретным ключом, используемым при расшифровке входящего трафика, в файл stunnel.pem.

Более полную информацию по этому программному обеспечению смотрите в документации, идущей в комплекте с Stunnel.

Утилиты сканирования и защиты сети

Утилиты сканирования — это класс программного обеспечения, предназначенный для нахождения уязвимостей в конфигурации компьютера или сети. Они могут быть и средством для улучшения безопасности, и инструментом для взлома системы.

SATAN

Одна из старейших утилит сканирования. SATAN может работать на нескольких операционных системах. Считается устаревшим, но, тем не менее, для проверки правильности основных сетевых настроек вполне пригоден. Работает от пользователя root, требует наличия Perl.

После запуска SATAN становится Web-сервером и запускает браузер, поскольку интерфейс у него Web-ориентированный. Для начала работы необходимо указать сканируемый хост или диапазон адресов и "уровень нападения", который может быть слабым, нормальным и тяжелым. После этого кнопкой **Start the scan** запускается сканирование.

По окончании сканирования следует перейти в раздел **Reporting & Data Analysis**, где можно ознакомиться с найденными проблемами, которые необходимо устранить.

Portsentry

Еще один программный продукт, предназначенный для обнаружения сканирования сетевых портов. Основные возможности программы Portsentry:

- обнаруживает практически все известные виды сканирования компьютеров;
- в реальном времени блокирует компьютер, производящий сканирование, посредством установленного на атакуемом компьютере брандмауэра, команду запуска которого можно задать в файле конфигурации;

- записывает в журнал операционной системы посредством syslogd информацию об атаке;
- может вызывать любую указанную в файле конфигурации программу в ответ на сканирование или подключение к защищенному сетевому порту.

Установка и настройка

Процесс установки подробно описан в документации на программу и не вызывает трудностей, поэтому сразу перейдем к настройке программы.

Основной конфигурационный файл программы Portsentry называется portsentry.conf. Содержимое файла portsentry.conf представляет собой несколько строк, каждая из которых имеет вид:

```
ОПЦИЯ = "значение"
```

Основные поддерживаемые опции:

- `TCP_PORTS` — в этой опции через запятую перечисляются TCP-порты, которые проверяются программой Portsentry. При обнаружении подключения к перечисленным портам Portsentry записывает информацию об этом в системный журнал и выполняет команду, заданную пользователем, а после этого блокирует хост посредством брандмауэра. TCP-порты, открытые на защищаемом компьютере другими программами, в этот список включаться не должны;
- `UDP_PORTS` — то же, что и `TCP_PORTS`, но для UDP-портов;
- `ADVANCED_PORTS_TCP` — верхняя граница множества TCP-портов, которые проверяются Portsentry при работе в режиме Advanced Stealth Scan Detection Mode. Нижней границей является 1, т. е. при значении `ADVANCED_PORTS_TCP`, равном 2048, проверяется подключение к любому порту в промежутке от 1 до 2048;
- `ADVANCED_PORTS_UDP` — то же, что и `ADVANCED_PORTS_TCP`, но для UDP-портов;
- `ADVANCED_EXCLUDE_TCP` — TCP-порты, которые исключаются из промежутка проверяемых портов, заданного параметром `ADVANCED_PORTS_TCP`. Здесь обязательно нужно перечислить TCP-порты, открытые работающими на защищаемом компьютере программами;
- `ADVANCED_EXCLUDE_UDP` — то же, что и `ADVANCED_EXCLUDE_TCP`, но для UDP-портов;
- `IGNORE_FILE` — имя и путь к файлу с IP-адресами хостов, которые не блокируются при подключении к портам, проверяемым программой Portsentry;
- `HISTORY_FILE` — имя и путь к файлу с историей работы программы Portsentry. В файл записывается время блокирования, имя и IP хоста, атакованный порт, протокол;
- `BLOCKED_FILE` — строка, из которой формируется имя и путь к файлам, куда записывается информация о блокированных хостах;
- `BLOCK_TCP` — эта опция в зависимости от значения задает ответную реакцию Portsentry на сканирование портов:
  - `0` — не блокировать хост, не запускать заданную пользователем команду;
  - `1` — блокировать хост и запустить команду;
  - `2` — только запустить заданную команду.

  Команда задается при помощи опции `KILL_RUN_CMD`;

- BLOCK_UDP — то же, что и BLOCK_TCP, но для UDP;
- KILL_ROUTE — эта опция задает команду, которую нужно выполнить для блокирования атакующего хоста. IP-адрес задает переменная $TARGET$. Переменная $PORT$ указывает порт, к которому было подключение;
- KILL_HOSTS_DENY — задает строку, которая записывается в /etc/hosts.deny для блокирования доступа к сервисам, запускаемым через inetd;
- KILL_RUN_CMD — с помощью этой опции можно задать команду, запускаемую до блокирования хоста;
- SCAN_TRIGGER — данная опция задает число разрешенных подключений к проверяемым программой Portsentry портам одного и того же хоста, прежде чем Portsentry начнет действовать. 0 определяет немедленную реакцию;
- PORT_BANNER — задает сообщение, которое будет выводиться при подключении к проверяемому Portsentry порту.

В файле portsentry.ignore необходимо перечислить IP-адреса компьютеров, которые не должны быть блокированы программой при подключении к проверяемому порту.

Запуск

Portsentry можно запускать в трех различных режимах. Режимы задаются в командной строке при вызове Portsentry. Одновременно можно задать только один режим работы для одного протокола:

- Classic — при работе в этом режиме Portsentry открывает порты, указанные в TCP_PORTS или UDP_PORTS, и ждет соединения. При попытке подключиться к такому порту происходит блокировка удаленного хоста. Этот режим работы задается опциями командной строки -tcp — для TCP-портов и -udp — для UDP-портов;
- Enhanced Stealth Scan Detection — этот режим используется для проверки перечисленных в TCP_PORTS или UDP_PORTS портов на предмет подключения или сканирования. Выявляет почти все виды Stealth-сканирования, а не только сканирование подключением. В отличие от режима Classic, не держит открытыми порты, поэтому сканировщик получает достоверную информацию об открытых портах. Задается опциями командной строки -stcp — для TCP-портов и -sudp — для UDP-портов;
- Advanced Stealth Scan Detection — этот режим служит для проверки всех портов в промежутке от 1 до ADVANCED_PORT_TCP или ADVANCED_PORT_UDP. Порты, открытые другими программами и перечисленные в ADVANCED_EXLUDE_TCP или ADVANCED_EXCLUDE_UDP, исключаются из проверки. Любой компьютер, попытавшийся подключиться к порту в этом промежутке, тут же блокируется. Задается опциями командной строки -atcp — для TCP-портов и -audp — для UDP-портов.

Сетевая статистика

Очень часто администратору необходимо получить развернутую информацию по сетевому трафику: кто, когда, сколько и по какому протоколу отправлял/принимал информацию. Конечно, все это можно выяснить из различных log-файлов, однако незачем тратить время на изготовление анализаторов log-файлов, когда уже есть готовые программные решения.

NeTraMet

Этот программный пакет позволяет подсчитывать трафик по IP-адресам в локальной сети раздельно по типам трафика: SMTP, ICMP, HTTP, FTP, UDP, TCP и т. п. Также существует возможность подробной регистрации трафика.

Программный пакет состоит из:

- NeTraMet — программы-сборщика трафика. Собирает и хранит в оперативной памяти статистику с сетевых интерфейсов сервера;
- NeMaC — программы-менеджера сборщика NeTraMet. Собирает статистику и записывает ее в журнал;
- srl — компилятора правил для NeMaC;
- fd_filter — программы обработки журналов NeMaC;
- fd_extract — программы обработки результатов fd_filter.

Ключи запуска NeTraMet

Программа запускается со следующими ключами:

- `-i network_interface` — сетевой интерфейс, трафик которого будет считать NeTraMet;
- `-l` — предписывает использовать размер пакета из заголовка, а не аппаратный размер;
- `-m 614` — UDP-порт, на котором будет соединяться NeTraMet с NeMaC;
- `-r password_for_read` — пароль на чтение;
- `-w password_for_write_and_read` — пароль на чтение/запись;
- `-f 60000` — максимальное количество сетевых потоков в NeTraMet. Чем больше клиентов, трафика и степень детализации статистики, тем больше сетевых потоков.

Ключи запуска NeMaC

Программа запускается со следующими ключами:

- `-k 120` — каждые 120 секунд NeMaC будет проверять, не перезагрузился ли NeTraMet;
- `-F /var/ntm.log/$DATER.flows` — записывает статистику в заданный файл;
- `-m 614` — порт для управления NeTraMet;
- `-c 900` — предписывает забирать статистику с NeTraMet каждые 15 минут;
- `-p` — предписывает после записи в файл статистики данных закрывать его. Если файл не найден, то он создается заново;
- `-L /var/ntm.log/$DATER.nemac` — журнал работы NeMaC;
- `-r /root/ntm.sh/short.3.rules` — файл с правилами.

Протоколирование

Нет смысла тратить много времени на защиту компьютера от взлома и не обращать внимания на систему протоколирования событий. Как можно узнать о попытке и способе взлома, не используя инструментов для ведения log-файлов? В этом разделе мы познакомимся со стандартной системой записи log-файлов — демоном syslogd.

Демон syslogd является частью пакета sysklogd, в который входят две программы: syslogd и klogd. Syslogd отвечает за протоколирование сообщений системы, а klogd — ядра.

Демон syslogd

Демон syslogd запускается автоматически при старте системы и обеспечивает протоколирование событий, которое используется большинством программ. Демон syslogd пишет сообщения в файлы /var/log/* в зависимости от настроек. Обычно записи в log-файле, создаваемом syslogd, содержат следующие поля: дата и время, имя компьютера, программа, сообщение.

Параметры запуска

В табл. 30.1 приведены основные параметры командной строки демона syslogd.

Таблица 30.1. *Основные параметры командной строки syslogd*

Параметр	Описание
`-d`	Включает режим отладки
`-f file`	Определяет альтернативный файл конфигурации
`-h`	По умолчанию демон не перенаправляет сообщения, которые он получает от других узлов. Этот параметр позволяет перенаправить сообщения другим хостам
`-n`	Этот параметр нужен, если syslogd запускается и контролируется программой init
`-p socket`	Позволяет задать другой сокет UNIX вместо /dev/log
`-r`	Позволяет принимать сообщения из сети
`-s socket`	Этот параметр позволяет указать дополнительный сокет, который syslog должен прослушивать
`-v`	Выводит версию syslogd

Файл конфигурации

Файл конфигурации по умолчанию — /etc/syslog.conf. Вы можете указать другой файл конфигурации с помощью опции `-f`. В листинге 30.4 приведен типичный файл конфигурации.

Листинг 30.4

```
# Все сообщения ядра операционной системы выводить на консоль
#kern.*                                          /dev/console

# Все сообщения уровня info или выше протоколировать в файл
# /var/log/messages
# Кроме почтовых сообщений и сообщений аутентификации
*.info;mail.none;authpriv.none;cron.none      /var/log/messages

# Протоколирование аутентификации.
# Файл протокола /var/log/secure
authpriv.*                                       /var/log/secure

# Все log-сообщения почтовой системы сохранять в файле /var/log/maillog.
mail.*                                           /var/log/maillog
# Все сообщения демона cron сохранять в файле /var/log/cron
cron.*                                           /var/log/cron

# Everybody gets emergency messages
*.emerg                                          *

# Сообщения системы новостей уровня crit и выше сохранять в файле
# /var/log/spooler
uucp,news.crit                                   /var/log/spooler

# Все загрузочные сообщения хранить в файле /var/log/boot.log
local7.*                                         /var/log/boot.log
```

Файл конфигурации состоит из двух полей: объект протоколирования и файл, в который будут записываться сообщения, порождаемые этим объектом. Для каждого объекта можно указать один из уровней протоколирования:

- `debug` — отладочная информация;
- `info` — просто информация;
- `notice` — уведомление;
- `warn` — предупреждение;
- `err` — ошибка;
- `emerg` — критический уровень.

Первые три уровня протоколирования относятся к информационным сообщениям. Уровень `warn` — это предупреждения, а `err` — ошибки. Помимо этого, существуют критические сообщения, которые выводятся прямо на консоль. Для обозначения объектов и уровней протоколирования допускается символ `*`, который обозначает все объекты или все уровни.

Сетевое протоколирование

Для обеспечения повышенной защищенности сети все сообщения можно хранить не на локальном компьютере, а передавать по сети на специальный сервер, на котором будет находиться база log-файлов компьютеров, подключенных к сети.

Сообщения передаются по протоколу UDP. Для нормального функционирования необходимо в файле /etc/service снять комментарий со строки `syslog 514/udp`.

После этого нужно внести изменения в файл конфигурации /etc/syslog.conf: файлы протоколов замените параметром `@hostname`, где `hostname` — это имя компьютера, на который будут перенаправлены сообщения.

Имя узла желательно указать в файле /etc/hosts, поскольку демон syslogd обычно стартует раньше, чем сервер DNS.

Демон klogd

Демон klogd предназначен для перехвата и протоколирования сообщений ядра Linux. В табл. 30.2 приведены основные параметры командной строки демона klogd.

Таблица 30.2. *Основные параметры командной строки klogd*

Параметр	Описание
`-c n`	Устанавливает уровень сообщений, которые будут выводиться на экран
`-d`	Режим отладки
`-f file`	Записывать сообщения в указанный файл раньше демона syslogd
`-i`	Позволяет перезагрузить символьную информацию ядра о модулях
`-I`	Перезагружает статическую символьную информацию и информацию о модулях ядра
`-k file`	Использует указанный файл в качестве файла, содержащего символьную информацию ядра
`-n`	Не переходить в фоновый режим. Этот параметр используется, когда демон управляется программой init
`-o`	Демон читает и протоколирует все сообщения, которые он найдет в буферах сообщений ядра. После одного цикла чтения/протоколирования демон завершает работу
`-s`	Заставляет демон klogd использовать системные вызовы для обращений к буферам сообщений ядра
`-v`	Выводит версию klogd

По умолчанию демон klogd активизируется системным вызовом для того, чтобы препятствовать отображению всех сообщений на консоль. Это не распространяется на критические сообщения ядра (kernel panic). Такие сообщения в любом случае будут отображены на консоли.

Защита системы после взлома

Правда, несколько странное название раздела? Как это — защита системы *после* взлома? Если вы помните, вопросы обеспечения безопасности компьютера в целом уже рассматривались нами ранее. Теперь остановимся на сетевой безопасности, а именно на том моменте, когда взлом уже произошел. После обнаружения факта взлома стандартным решением является отключение взломанного компьютера от сети и полная переустановка операционной системы с последующим обновлением всего программного обеспечения, установленного на компьютере. А что делать, если нет возможности вывести из работы взломанный компьютер, а защитить его все равно необходимо? Именно такую ситуацию мы и рассмотрим в этом разделе.

Взломы операционной системы бывают разные. Самый простой вариант — какой-то подросток начитался литературы или нашел в Интернете программку-взламыватель, например sendmail, применил свои псевдознания к вашей системе, пошалил — удалил что-то с вашего компьютера или, наоборот, оставил послание — "ваш компьютер взломан суперхакером Васей" и ушел, причем, зачастую, не уничтожив следы своего воздействия на компьютер. На такой простой случай вам рассчитывать не стоит. Как правило, серьезный взлом подготавливается долгое время, и о нем вы узнаете, например, от администратора какого-нибудь сервера на другом конце земного шара, да и то потому, что от вас на его сервер идет очень большой трафик или еще по каким-либо косвенным признакам. Такой взлом преследует чисто прагматические цели — воспользоваться вашей системой для дальнейшего взлома других компьютеров, устроить на вашем сервере хранилище файлов или что-нибудь в подобном роде. Причем взломщик после себя всегда оставляет на вашем компьютере набор специальных утилит, называемых rootkit.

Rootkit

Rootkit (набор инструментов администратора) — это набор утилит, которые взломщик устанавливает на взломанном компьютере после получения первоначального доступа. Rootkit обычно содержит сетевой сниффер (утилиту, способную получать и обрабатывать *весь* сетевой трафик вашей локальной сети, независимо от того, какому компьютеру адресованы сетевые пакеты) для прослушивания сетевого трафика, программы для модификации log-файлов, позволяющие скрыть присутствие взломщика на вашем компьютере, и специально модифицированные системные утилиты, замещающие основные утилиты системы, например ps, netstat, ifconfig, killall, login.

Основное назначение rootkit — позволить взломщику возвращаться во взломанную систему и получать доступ к ней, не будучи при этом обнаруженным системным администратором. Обычно для этого применяется модифицированная версия telnetd или sshd. Модифицированный сервис будет использовать сетевой порт, отличный от того, который этот демон по умолчанию прослушивает. Большинство версий rootkit снабжены модифицированными системными программами, которые замещают существующие во взломанной системе. Конкретный набор модифицируемых системных

утилит зависит от версии rootkit и нужд и квалификации взломщика, но, как правило, заменяются программы ps, w, who, netstat, ls, find, login и другие, которые позволяют контролировать работу взломанной системы.

Для усложнения обнаружения подмены системных утилит большинство rootkit, производя замену системных утилит на модифицированные версии, устанавливают точно такие же даты их создания и размеры файлов, поэтому простой список файлов с датой их создания и модификации и размером никакой пользы в обнаружении подмены системных утилит не принесет. Лучший способ обнаружения подмены системных утилит — получить контрольную сумму файлов в системе и сохранить этот список в надежном месте — на другом компьютере или на компакт-диске.

В принципе, можно воспользоваться возможностями, предоставляемыми менеджером пакетов RPM — контрольной суммой пакета, рассчитанной по алгоритму MD5. При этом RPM учитывает контрольные суммы пакетов, хранящиеся в базе данных установленных RPM. Как легко заметить, подобный способ не подходит для обнаружения опытных взломщиков. Причин тому две.

- В вашей системе могут быть установлены программы не из RPM, а скомпилированные из исходных кодов — совершенно очевидно, что ваш менеджер пакетов абсолютно ничего не знает о программах, устанавливаемых без помощи RPM.
- База данных RPM находится на взломанном компьютере, и взломщику не составляет труда модифицировать ее нужным образом или вообще повредить.

Эту проблему обычно решают специализированные программные пакеты, например Tripwire или AIDE, о которых мы поговорим несколько позже.

Помимо перечисленного, некоторые rootkit содержат сетевой анализатор пакетов и утилиты для записи нажатий клавиатурных кнопок, что позволяет взломщику с целью получения необходимой информации организовать сбор паролей и анализ сетевого трафика.

Наибольшую угрозу для безопасности вашей системы представляют rootkit, использующие загружаемые модули ядра (Loadable Kernel Module, LKM), что позволяет не подменять системные утилиты, а нарушать их правильное функционирование через ядро операционной системы.

Обнаружение rootkit

Мрачная картина — получается, что после взлома системы сделать для ее излечения ничего не возможно? К счастью, не все так плохо.

Сначала необходимо определить сам факт взлома системы. Последствием взлома вашего компьютера и установки на нем rootkit может стать изменение в поведении системных утилит. Например, некоторые утилиты отказываются запускаться от имени пользователя, которому было разрешено их выполнение. Или ваша любимая утилита top стала выглядеть несколько иначе. Другие очень настораживающие признаки — изменение показателей сетевого трафика, а также резкое уменьшение свободного места на жестком диске.

Сканирование портов

После обнаружения взлома первое, что необходимо сделать вслед за сменой паролей, — лишить взломщика возможности проникновения в систему через сетевые порты. Поскольку взломанный компьютер не вызывает доверия, просканировать сетевые порты нужно с другого компьютера.

Проще всего просканировать порты с помощью программы nmap, выполнив команду

```
nmap -p 1-65535 192.168.0.1
```

Указываем диапазон сканируемых портов от 1 до 65 535, а также адрес сканируемого компьютера. После этого на консоль будет выдан список портов, протокол, используемый для каждого порта, и сервис, который использует этот порт. Обычно всякие "специальные" программы обращаются к портам выше 1023, причем зачастую это порты с номером более 10 000.

Помимо nmap, можно воспользоваться программой lsof. Она позволяет получить список открытых на вашем компьютере сетевых портов. Для этого достаточно выполнить команду

```
lsof -i
```

Использование RPM

Хотя чуть ранее утверждалось, что обнаружение rootkit с помощью RPM — дело бесперспективное, это не совсем так. RPM можно применить для быстрой проверки. Если он не найдет ничего подозрительного — воспользуемся другими средствами, если найдет — и на том спасибо — будем знать, чтó у нас не так в системе.

RPM записывает и проверяет контрольную сумму всех файлов в пакете, включая те файлы, которые должны изменяться с течением времени. О проверке контрольных сумм пакетов RPM см. в *главе 9*.

Сканер для rootkit

Пакет chkrootkit — набор утилит для выявления присутствия в системе уже известных rootkit. Chkrootkit удобен тем, что способен отыскать большое количество rootkit с помощью единственного приложения. Вдобавок к выявлению известных rootkit, он также включает несколько тестов, помогающих обнаружить новые rootkit.

Пакет chkrootkit состоит из следующих утилит:

- chkrootkit — выявление сигнатур известных rootkit;
- ifpromisc — обнаружение прослушивания сетевого трафика взломанным компьютером;
- chklastlog, chkwtmp, check_wtmpx — проверка log-файлов;
- chkproc — обнаружение "посторонних" загружаемых модулей ядра операционной системы.

Об особенностях применения chkrootkit можно узнать в документации, идущей в комплекте с пакетом.

После обнаружения

Что делать после обнаружения rootkit? Единственно верный способ избавиться от последствий взлома — заново полностью переустановить операционную систему и установить все обновления пакетов для вашего дистрибутива. Однако не всегда есть возможность проделать такие действия сразу — квартальный отчет, непрерывное производство, болезнь администратора — да мало ли что еще.

В дистрибутивах на основе RPM-пакетов вы можете определить поврежденные пакеты. После этого необходимо переустановить их командой

```
rpm -U --force rpm_package_name.rpm
```

Затем вы должны удалить файлы, установленные в вашу систему взломщиком. Данные, полученные chkrootkit, помогут вам определить местонахождение файлов. После удаления всех обнаруженных "чужих" файлов запустите top и ps для выявления и уничтожения оставшихся нежелательных процессов. Помимо этого, необходимо проверить стартовые скрипты операционной системы и убедиться, что эти скрипты не используются никакими посторонними программами.

LIDS

LIDS (Linux Intrusion Detection/Defense System) — система обнаружения и защиты от вторжения. Представляет собой дополнение к ядру операционной системы Linux, добавляющее возможности для увеличения безопасности операционной системы. LIDS позволяет запретить или ограничить доступ к файлам, памяти, устройствам, сетевым интерфейсам, запущенным приложениям и т. п. пользователю root, что дает возможность надежно оградить даже взломанную операционную систему от дальнейшего вмешательства.

В отличие от других средств защиты Linux, эту систему невозможно отключить, не зная пароля администратора LIDS, который в зашифрованном виде хранится в специальном файле, видимом только программой администрирования LIDS. Точно так же защищены и конфигурационные файлы LIDS. Даже узнав каким-то образом пароль администратора LIDS, отключить систему можно, только находясь за консолью компьютера.

LIDS позволяет распределять права доступа к файлам на уровне программ, а не на уровне пользователей, а также запретить перезапуск операционной системы, загрузку/выгрузку модулей ядра и многое другое.

Информация обо всех действиях, имеющих отношение к защищаемым объектам, помимо записи в log-файлах может немедленно отправляться по электронной почте.

Кроме всего прочего, в LIDS присутствует встроенный детектор сканирования сетевых портов.

Установка

Получив пакет LIDS, необходимо разархивировать его и наложить патч на исходники ядра операционной системы Linux. После этого следуйте инструкции — там все понятно — компилируем, устанавливаем.

Далее, нам необходимо перекомпилировать ядро Linux с поддержкой LIDS. Для этого в пункте меню конфигурации ядра **Code maturity level options** необходимо включить опцию **Prompt for development and/or incomplete code/drivers**.

Затем в пункте меню **General setup** необходимо включить опцию **Sysctl support**.

Далее необходимо зайти в меню **Linux Intrusion Detection System**. Это меню полностью относится к конфигурированию LIDS. Первым идет включение поддержки LIDS в ядре:

```
[*] Linux Intrusion Detection System support (EXPERIMENTAL)
```

После включения поддержки LIDS станет доступным список опций настройки LIDS:

- **Maximum protected objects to manage** — этот пункт позволяет установить максимальное количество защищаемых объектов;
- **Maximum ACL subjects to manage** — позволяет установить максимальное количество субъектов правил доступа LIDS;
- **Maximum ACL objects to manage** — позволяет установить максимальное число объектов правил доступа LIDS;
- **Maximum protected proceeds** — позволяет установить максимальное количество защищаемых процессов;
- **Hang up console when raising securit alert** — разрешает закрытие консоли, с которой произошло нарушение безопасности;
- **Security alert when execing unprotected programs before sealing LIDS** — разрешает вывод сообщения о нарушении безопасности при запуске незащищенных программ;
- **Do not execute unprotected programs before sealing LIDS** — включает запрет на запуск незащищенных программ до установки способностей;
- **Try not to flood logs** — при включении этой опции LIDS не будет записывать в log-файлы дублирующиеся сообщения об одном и том же нарушении защиты;
- **Authorized time between two identic logs (seconds)** — устанавливает время в секундах, в течение которых проверяется появление двух идентичных сообщений, чтобы не записывать одинаковые сообщения в log-файлы;
- **Allow switching LIDS protections** — включает возможность отключения и включения LIDS в процессе работы системы после ввода пароля. При включении данной опции появляется возможность поменять любые параметры работы без перезагрузки операционной системы;
- **Numbers of attempts to submit password** — определяет число попыток ввода пароля, по истечении которых отключение LIDS становится невозможным на заданный далее промежуток времени;
- **Time to wait after fail (seconds)** — время в секундах, в течение которого после ввода неправильного пароля указанное количество раз отключение LIDS становится невозможным;
- **Allow remote users to switch LIDS protections** — дает возможность удаленным пользователям отключать LIDS. С целью увеличения безопасности вашей операционной системы не включайте эту опцию;

- **Allow any program to switch LIDS protections** — позволяет любой программе отключать LIDS. Не включайте эту опцию;
- **Allow reloading config. File** — разрешает переконфигурирование LIDS без перезагрузки компьютера;
- **Port Scanner Detector in kernel** — позволяет в ядро операционной системы добавить детектор сканирования портов;
- **Send security alerts through network** — разрешает отправку электронной почты при нарушении безопасности на указанный электронный адрес с информацией о нарушении. Письмо отправляется незамедлительно при попытке совершения несанкционированных действий;
- **Hide klids network threads** — позволяет скрывать сетевые соединения LIDS;
- **Number of connection tries before giving up** — задает число попыток соединения с SMTP-сервером;
- **Sleep time after a failed connection** — задает время в секундах между попытками соединения с почтовым сервером;
- **Message queue size** — определяет максимальное количество почтовых сообщений в очереди. При превышении данного количества самое старое неотправленное сообщение удаляется из очереди;
- **LIDS debug** — включает вывод отладочных сообщений LIDS.

После конфигурирования можно компилировать и устанавливать ядро операционной системы.

Конфигурирование LIDS

После установки LIDS в каталоге /etc появляется каталог lids, содержащий следующие конфигурационные файлы:

- lids.cap — хранит текущие значения установок способностей;
- lids.net — предназначен для настройки отправки электронных сообщений системой LIDS;
- lids.pw — содержит в зашифрованном виде пароль администратора. Изменять этот файл можно только с помощью lidsadm;
- lids.conf — содержит текущие установки правил доступа. Изменять этот файл можно только с помощью lidsadm.

Способности

Способности (capabilities) — определяют возможность программ совершать какие-либо действия. LIDS позволяет использовать по отношению к программам большое количество способностей. В частности LIDS поддерживает способность перезагружать компьютер, изменять владельца файла, загружать или выгружать модули ядра и многое другое.

Текущие установки способностей хранятся в файле lids.cap в формате:

```
[+|-] Номер:Способность
```

Здесь:

- `+` — включает способность;
- `-` — отключает способность.

Редактировать файл lids.cap можно с помощью любого текстового редактора. Включение способности влияет на все программы без исключения, а выключение влияет на все программы, кроме тех, которым напрямую указана данная способность с помощью правил доступа lidsadm.

Сразу после установки LIDS файл lids.cap содержит включенными следующие способности:

- `CAP_CHOWN` — устанавливает способность программ изменять владельца и группу владельца файла;
- `CAP_DAC_OVERRIDE` — разрешает программам, запускаемым пользователем root, не принимать во внимание режимы доступа к файлам. При отключении этой способности пользователь root теряет возможность изменять любые файлы, невзирая на права доступа;
- `CAP_DAC_READ_SEARCH` — то же самое, что и предыдущая способность, только по отношению к каталогам;
- `CAP_FOWNER` — разрешает операции с файлами, когда владелец файла должен совпадать с пользователем, совершающим операцию;
- `CAP_FSETID` — разрешает установку SUID- или SGID-бита на файлах, не принадлежащих пользователю root;
- `CAP_KILL` — разрешает процессам пользователя root "убивать" чужие процессы;
- `CAP_SETGID` — управляет способностью программ пользователя root изменять группу, под которой работает программа;
- `CAP_SETUID` — управляет способностью программ пользователя root изменять пользователя, под которым работает программа;
- `CAP_SETPCAP` — разрешает программам менять способности;
- `CAP_LINUX_IMMUTABLE` — управляет способностью снимать атрибуты `S_IMMUTABLE` и `S_APPEND` с файлов;
- `CAP_NET_BIND_SERVICE` — разрешает программам использовать сетевой порт, меньший, чем 1024;
- `CAP_NET_BROADCAST` — управляет способностью программ рассылать широковещательные пакеты;
- `CAP_NET_ADMIN` — управляет большим количеством различных способностей: конфигурирование сетевых интерфейсов, изменение правил брандмауэра, изменение таблиц маршрутизации и многих других, связанных с сетевыми настройками Linux;
- `CAP_NET_RAW` — управляет способностью программ использовать сокеты;
- `CAP_IPC_LOCK` — управляет способностью процессов пользователя root блокировать сегменты разделяемой памяти;
- `CAP_IPC_OWNER` — управляет доступом программ пользователя root к ресурсам межпроцессорного взаимодействия процессов, не принадлежащих пользователю root;
- `CAP_SYS_MODULE` — управляет способностью загружать модули ядра;

- CAP_SYS_RAWIO — управляет доступом на чтение/запись к таким устройствам, как /dev/mem, /dev/kmem, /dev/port, /dev/hdXX, /dev/sdXX;
- CAP_SYS_CHROOT — управляет способностью устанавливать корневой каталог для текущей командной оболочки;
- CAP_SYS_PTRACE — включает способность программ использовать вызов функции ptrace(), которая позволяет управлять выполнением процессов-потомков процессу-родителю;
- CAP_SYS_PACCT — управляет способностью конфигурировать учет процессов;
- CAP_SYS_ADMIN — управляет множеством способностей: управление устройством /dev/random, создание новых устройств, конфигурирование дисковых квот, настройка работы klogd, установка имени домена, установка имени хоста, сброс кэша, монтирование и размонтирование дисков, включение/отключение swap-раздела, установка параметров последовательных портов и многое другое;
- CAP_SYS_BOOT — управляет способностью перезагружать систему;
- CAP_SYS_NICE — управляет способностью изменять приоритет процессов, не принадлежащих пользователю root;
- CAP_SYS_RESOURCE — управляет способностью изменять лимиты использования ресурсов системы: дисковые квоты, зарезервированное пространство на разделах, максимальное количество консолей и т. п.;
- CAP_SYS_TIME — управляет способностью изменять системное время;
- CAP_SYS_TTY_CONFIG — управляет способностью изменять настройки tty-устройств;
- CAP_HIDDEN — управляет способностью программ делаться невидимыми в списке процессов. Не влияет на все программы;
- CAP_INIT_KILL — управляет способностью "убивать" процессы-потомки процесса init.

Как видите, впечатляющий набор возможностей. Самое время разобраться, что из этого нужно включить, а что выключить для вашей операционной системы.

Для инициализации параметров способностей в процессе загрузки используется команда

```
lidsadm -I
```

Обычно ее ставят в конце /etc/rc.d/rc.local, что позволяет отключить способности только после запуска всех программ, необходимых для работы сервера.

Правила доступа

Все управление LIDS осуществляется с помощью программы — lidsadm. Lidsadm работает в двух режимах — настройки правил доступа или ввода команд администрирования. Установки правил доступа находятся в файле /etc/lids/lids.conf. Для просмотра текущих установок правил доступа необходимо выполнить команду

```
lidsadm -L
```

Результат приведен в листинге 30.5.

Листинг 30.5

```
LIST
Subject ACCESS TYPE Object
-------------------------------------------------
Any File READ /sbin
Any File READ /bin
Any File READ /boot
Any File READ /lib
Any File READ /usr
Any File DENY /etc/shadow
/bin/login READ /etc/shadow
/bin/su READ /etc/shadow
Any File APPEND /var/log
Any File WRITE /var/log/wtmp
/sbin/fsck.ext2 WRITE /etc/mtab
Any File WRITE /etc/mtab
Any File WRITE /etc
/usr/sbin/sendmail WRITE /var/log/sendmail.st
/bin/login WRITE /var/log/lastlog
/bin/cat READ /home/xhg
Any File DENY /home/httpd
/usr/sbin/httpd READ /home/httpd
Any File DENY /etc/httpd/conf
/usr/sbin/httpd READ /etc/httpd/conf
/usr/sbin/sendmail WRITE /var/log/sendmail.st
/usr/X11R6/bin/XF86_SVGA NO_INHERIT RAWIO
/usr/sbin/in.ftpd READ /etc/shadow
/usr/sbin/httpd NO_INHERIT HIDDEN
```

Правила доступа состоят из трех элементов: субъекта, объекта и цели. Объектом является любой файл или каталог, на который и должны действовать правила доступа и защита LIDS. Если в качестве объекта указывается каталог, то все файлы в нем и вложенные каталоги с их файлами автоматически становятся объектами. Субъект — любая защищенная программа, которой дают доступ к защищаемому объекту, поэтому прежде чем использовать программу в качестве субъекта, ее саму надо защитить средствами LIDS, применив к ней правила доступа как к объекту. Если субъект не указан, то субъектом будет любая программа. Целью является тип доступа:

- READ — доступ на чтение;
- WRITE — запись;
- DENY — запрет на какой-либо доступ;
- APPEND — открытие только для записи в конец файла;
- IGNORE — игнорирование защиты.

Построение прав доступа подробно описано в документации на пакет LIDS, поэтому мы на этом здесь не останавливаемся.

После настройки LIDS необходимо перезагрузить операционную систему. В том случае, если с функционированием LIDS возникли проблемы, можно загрузить Linux с выключенным LIDS, для чего при загрузке следует передать ядру операционной системы параметр `security=0`. Например, для LILO это будет выглядеть так:

```
LILO boot: linux security=0
```

Portsentry

Программа предназначена для обнаружения попыток сканирования портов и организации адекватного (с точки зрения администратора) ответа. Основные возможности Portsentry:

- обнаруживает практически все известные виды сканирования UNIX-машин;
- в реальном времени блокирует хост, с которого происходит сканирование портов, посредством установленного на атакуемом компьютере брандмауэра;
- записывает в log-файлы посредством syslogd информацию об атаке;
- в ответ на сканирование или подключение к защищенному порту вызывает программу, указанную администратором при конфигурировании.

Пакет Portsentry прост в установке, конфигурировании и использовании. Его можно получить как в исходных кодах, так и в виде RPM-пакета.

LogSentry

Программа LogSentry предназначена для автоматического мониторинга log-файлов и уведомления системного администратора с помощью электронной почты о подозрительных происшествиях в системе. Гибко настраивается.

Tripwire

Программный пакет tripwire предназначен для обнаружения изменения файлов, позволяя выявлять порчу данных и взломы. База данных контрольных сумм файлов шифруется, что предотвращает ее подделку взломщиками.

Непосредственно после установки операционной системы необходимо инсталлировать tripwire, которая на основе правил, определенных политикой безопасности, создает базу данных, содержащую информацию обо всех файлах в системе (список файлов может задаваться администратором) — размер, контрольная сумма, дата модификации и т. п. После создания базы данных она ежедневно сравнивается с текущим состоянием файловой системы, позволяя обнаружить добавленные, измененные и удаленные файлы. Получаемые при этом отчеты можно просмотреть с различной степенью детализации.

Пакет tripwire входит в состав практически всех современных дистрибутивов Linux.

AIDE

Пакет AIDE — система обнаружения вторжений, основанная на мониторинге изменения контрольных сумм защищаемых файлов операционной системы. Система AIDE разработана так, что полная инсталляция ее помещается на одной дискете, что позволяет избежать вмешательства взломщика в функционирование программы.

Функционально программа аналогична tripwire, только имеет более простые конфигурационные файлы и интерфейс.

RSBAC

RSBAC — это надстройка над ядром Linux и комплект утилит управления, позволяющие создать на базе Linux защищенную систему. Механизмы защиты реализованы на уровне ядра системы, системные вызовы, затрагивающие безопасность, дополняются специальным кодом, выполняющим обращение к центральному компоненту RSBAC. Этот компонент принимает решение о допустимости данного системного вызова на основе многих параметров:

- типа запрашиваемого доступа (чтение, запись, исполнение);
- субъекта доступа;
- атрибутов субъекта доступа;
- объекта доступа;
- атрибутов объекта доступа.

Функционально RSBAC состоит из нескольких модулей, а центральный компонент принимает комплексное решение, основываясь на результатах, возвращаемых каждым из активных в данный момент модулей (какие модули задействовать и в каком объеме определяется на этапе настройки системы).

Вся информация, используемая RSBAC, хранится в дополнительном каталоге, который доступен только ядру системы.

RSBAC — мощная система защиты и разграничений прав доступа, но она несколько тяжеловата в настройке.

Security-Enhanced Linux

Security-Enhanced Linux имеет аналогичное с RSBAC назначение и представляет собой дополнения к ядру, а также набор утилит. Разработка Security Enhanced Linux продвигается Агентством национальной безопасности США (National Security Agency, NSA). Security-Enhanced Linux обеспечивает гибкую мандатную архитектуру управления доступом, использующую язык описания конфигураций политики безопасности.

По сравнению с RSBAC система Security-Enhanced Linux менее гибкая, зато имеет очень хорошую предопределенную политику безопасности. Ее настройка достаточно сложна и невозможна без изучения специального языка конфигурации. Входит практически во все современные дистрибутивы.

Ссылки

- **acl.bestbits.at** — официальная страница проекта Linux ACLs (Access Control Lists).
- **bog.pp.ru/work/tripwire.html** — Bog BOS: Tripwire: принципы работы, установка и настройка.
- **freshmeat.net/projects/netramet/** — страница проекта NeTraMet.
- **gazette.linux.ru.net/lg75/articles/rus-maiorano.html** — Ariel Maiorano. Инсталляция и использование AIDE. Перевод А. Куприна.
- **linuxrsp.ru/artic/portsentry.html** — Ерижоков А. А. Portsentry.
- **linuxrsp.ru/artic/posixacls.html** — Ерижоков А. А. Списки контроля доступа.
- **linuxrsp.ru/artic/stunnel.html** — Ерижоков А. А. Stunnel: Шифрование трафика.
- **linuxsecurity.com** — сайт, посвященный безопасности операционной системы Linux.
- **rootshell.com** — сайт, посвященный безопасности операционных систем.
- **stunnel.mirt.net** — официальный сайт пакета Stunnel.
- **www.chkrootkit.org** — официальный сайт chkrootkit.
- **www.cs.tut.fi/~rammer/aide.html** — страница разработчика AIDE.
- **www.false.com/security/linux/** — Secure Linux patches by Solar Designer — дополнения к ядру Linux, повышающие безопасность операционной системы.
- **www.insecure.org** — местонахождение программы nmap — сканера сетевых портов.
- **www.lids.org** — сайт проекта LIDS.
- **www.linuxrsp.ru/artic/lids.html** — Ерижоков А. А. LIDS — система обнаружения и защиты от вторжения.
- **www.monkey.org/~dugsong/dsniff** — страничка программы-сниффера Dsniff.
- **www.psionic.com** — сайт Psionic Software, разработчика программы Portsentry.
- **www.softerra.ru/freeos/16901/** — Oktay Altunergil. Понятие Rootkit. Перевод Инги Захаровой.
- **www.softerra.ru/freeos/16999/** — Oktay Altunergil. Сканирование для обнаружения Rootkit. Перевод Инги Захаровой.
- **www.softerra.ru/freeos/17032/** — Денис Колисниченко. Протоколирование.
- **www.tripwire.org** — сайт разработчиков Tripwire.
- Мяснянкин В. В. Linux на защите информации. Открытые системы 2001. № 4.
- **linux.ru.net/~inger/RSBAC-DOC-ru.html** — начала RSBAC.
- **www.opennet.ru/docs/RUS/netramet/index.html** — Денис Матыцын. Сбор статистики по TCP/IP на базе NeTraMet.
- REFERENCE MANUAL NeTraMet & NeMaC. Nevil Brownlee.

Глава 31



Доступ к удаленным компьютерам

Любая UNIX-подобная ОС может предоставлять удаленный доступ, начиная от простейшего консольного режима и заканчивая работой системы X Window, от простого редактирования текста (сидя за сотни и тысячи километров от хоста) до полного администрирования удаленной системы. В мире UNIX в порядке вещей ситуация, когда администратор месяцами не имеет физического контакта с сервером и, тем не менее, он ежедневно удаленно проводит мониторинг, обновление программного обеспечения и администрирование сервера.

Для этих целей разработано несколько программных пакетов и протоколов: Telnet, SSH, r-команды и некоторые другие. Будем рассматривать эти программы по старшинству.

Telnet

Под Telnet понимают трехкомпонентную систему, состоящую из:

- Telnet-клиента;
- Telnet-сервера;
- Telnet-протокола.

Протокол Telnet

Протокол Telnet описан в документе RFC854. Авторы стандарта говорят, что назначение Telnet — дать общее описание, насколько это только возможно, двунаправленного, восьмибитового взаимодействия, главная цель которого — обеспечение стандартного метода взаимодействия терминального устройства и терминал-ориентированного процесса. Кроме того, протокол позволяет организовать взаимодействие "терминал–терминал" (связь) и "процесс–процесс" (распределенные вычисления).

Telnet — протокол уровня приложения, использующий транспортный протокол TCP. Базовые концепции протокола Telnet:

- сетевой виртуальный терминал (Network Virtual Terminal, NVT);
- согласование параметров взаимодействия;
- симметрия связи "терминал–процесс".

Рассмотрим эти концепции подробнее.

Сетевой виртуальный терминал позволяет абстрагироваться от реалий жизни. Это стандартное описание типовых свойств реальных физических терминальных устройств. Сетевой виртуальный терминал дает возможность описать и преобразовать в стандартную форму способы отображения и ввода информации. Telnet-клиент и Telnet-сервер преобразовывают характеристики физических устройств в спецификацию сетевого виртуального терминала, что позволяет унифицировать параметры различных физических устройств и обеспечить их совместимость. Характеристики диалога диктуются устройством с меньшими возможностями.

В протоколе Telnet сетевой виртуальный терминал определен как "двунаправленное символьное устройство, состоящее из принтера и клавиатуры". Принтер предназначен для отображения информации, приходящей по сети, а клавиатура — для ввода данных, передаваемых по сети. По умолчанию предполагается, что для обмена информацией применяется 7-битовый код ASCII, каждый символ которого закодирован в 8-битовое поле.

Согласование параметров взаимодействия позволяет унифицировать представление информации на терминальных устройствах. Благодаря этой концепции, доступно большинство возможностей современных терминалов. Обычно для этого существует специальная таблица соответствия, которая обеспечивает замену нестандартных команд терминала стандартными. Как правило, согласование форм представления информации происходит в начальный момент организации Telnet-соединения. Каждый из процессов старается установить максимальные параметры сеанса. В UNIX-подобных системах параметры терминалов содержатся в базе данных описания терминалов termcap. При инициировании Telnet-соединения обычно именно эти параметры учитываются в процессе согласования формы представления данных. При этом из одной системы в другую передается значение переменной окружения `TERM`. В процессе договора останутся только те функции, которые поддерживаются на обоих концах соединения.

Симметрия взаимодействия позволяет клиенту и серверу в течение одной сессии меняться ролями.

Команды Telnet

В табл. 31.1 приведены некоторые команды протокола Telnet с кратким пояснением.

Таблица 31.1. *Команды протокола Telnet*

Команда	Десятичное значение	Описание
`EOF`	236	Конец файла
`SUSP`	237	Подавить текущий процесс
`ABORT`	238	Прервать процесс
`EOR`	239	Конец записи
`SE`	240	Конец вспомогательной процедуры согласования

Таблица 31.1 *(окончание)*

Команда	Десятичное значение	Описание
NOP	241	Нет операции
Data Mark	242	Часть потока данных для синхронизации
Break	243	Символ BRK виртуального терминала сети
Interrupt Process	244	Прервать текущий процесс
Abort Output	245	Отменить вывод без прекращения процесса
Are You There	246	Показывает, что связь не разорвана
Erase Character	247	Удалить предшествующий символ
Erase Line	248	Удалить текущую строку
Go Ahead	249	Запрос на ввод (для полудуплексных соединений)
SB	250	Начало вспомогательной процедуры согласования
WILL	251	Предложение начать выполнение факультативной команды (или подтверждение, что вы ее выполняете)
WON'T	252	Отказ выполнить (или продолжить выполнение)
DO	253	Требование к партнеру выполнить факультативную команду (или подтверждение ожидания выполнения другой стороной)
DON'T	254	Требование к партнеру прекратить выполнение факультативной команды (или подтверждение того, что больше не ожидается выполнение операции другой стороной)
IAC	255	Интерпретировать как команду

Протокол Telnet предусматривает единый TCP-канал и для данных пользователя, и для управления. Поскольку по протоколу Telnet команды управления чередуются с данными, командам должен предшествовать специальный символ, называемый IAC (Interpret as Command, интерпретировать как команду) с кодом 255. Если необходимо передать символ данных с десятичным кодом 255, то его следует продублировать.

Таким образом, каждая команда протокола Telnet состоит не менее чем из двух байтов. Первый из них всегда символ перехода в командный режим — IAC. Второй — собственно код команды.

Программа-клиент telnet

Программа telnet — стандартный Telnet-клиент, входящий во все операционные системы UNIX-семейства и практически во все ОС Windows.

Подключение к удаленной системе обычно осуществляет команда вида:

```
telnet <имя_хоста>
```

Основные команды программы telnet приведены в табл. 31.2.

Таблица 31.2. *Команды программы telnet*

Команда	Описание
`Open <host> [<port>]`	Начать Telnet-сессию с машиной `<host>` по порту `<port>`. Адрес машины можно задавать как в форме IP-адреса, так и в форме доменного адреса
`close`	Завершить Telnet-сессию и вернуться в командный режим
`Quit`	Завершить работу программы telnet
`Z`	"Заморозить" Telnet-сессию и перейти в режим интерпретатора команд локальной системы. Из этого режима можно выйти по команде `Exit`
`Mode <type>`	Если значение `<type>` равно `line`, то обмен данными буферизованный, если `character` — обмен не буферизованный
`? [<command>] help [<command>]`	Список команд или описание конкретной команды
`Send <argument>`	Данная команда используется для ввода команд и сигналов протокола Telnet, которые указываются в качестве аргумента

Программа-сервер telnetd

Программа telnetd — это сервер, обслуживающий протокол Telnet. Программа telnetd задействует TCP-порт 23, но ее можно сконфигурировать на работу с другим портом.

При установке взаимодействия с удаленным клиентом telnetd обменивается командами настройки: включение режима эха, обмен двоичной информацией, определение типа терминала, скорости обмена и установка переменных окружения.

Применение Telnet и безопасность

Протокол Telnet долгие годы был единственной универсальной возможностью удаленно работать с различными консольными программами. По своей простоте, нетребовательности к ресурсам и полосе пропускания он до сих пор не имеет себе равных. Помимо этого, клиентская программа telnet позволяет устанавливать соединения и с другими сервисами (например, с почтовым сервером SMTP или POP3), что дает возможность проводить различные манипуляции (например, просмотреть без почтового клиента содержимое своего почтового ящика или отправить письмо). Однако при всех достоинствах протокол Telnet имеет один огромный недостаток — весь трафик пересылается в открытом виде. Из-за этого любому злоумышленнику не составляет труда перехватить логин и пароль пользователя, а также другую информацию. Альтернатива протоколу Telnet — программные пакеты SSH или OpenSSH.

Семейство r-команд

Приблизительно в то же время, что и протокол Telnet, были созданы программы, предназначенные для удаленного администрирования и работы, так называемые r-команды (remote-команды).

rlogin

Команда `rlogin` (remote login) предназначена для входа на UNIX-хост через удаленный терминал. Стандарт RFC1282 содержит спецификацию протокола Rlogin. Программа rlogin устанавливает одно TCP-соединение между клиентом и сервером. Для нормального функционирования необходимо создать файл .rhosts, содержащий список хостов и пользователей, которым разрешено удаленно регистрироваться в системе. Каждая строка представляет собой пару "хост–пользователь", разделенную пробелом.

rsh

Команда `rsh` (remote shell) запускает командную оболочку на удаленном компьютере, после чего на нем возможно выполнение различных программ.

rcp

Команда `rcp` (remote copy) копирует файлы между компьютерами, причем эта операция может осуществляться между двумя удаленными компьютерами. Данная команда может копировать как один файл, так и группу файлов или каталогов.

rsync

Команда `rsync` аналогично команде `rcp` позволяет копировать файлы между удаленными компьютерами. Однако, в отличие от `rcp`, может значительно ускорить процесс копирования часто изменяемых пользователем файлов, поскольку благодаря реализованным алгоритмам передает только измененные части файлов. Также способна копировать ссылки (links), специальные устройства (device), владельца и группу файла, права доступа.

rdist

Эта команда позволяет осуществить массовую автоматическую рассылку файлов с локального хоста на большую группу хостов с проверкой наличия места, рассылкой извещений о проблемах, исполнением завершающих процедур и т. п. Сохраняет имя владельца, имя группы, права доступа и время модификации файла.

Применение r-команд и безопасность

Как и в ситуации с Telnet, r-команды имеют ту же проблему с безопасностью — абсолютно не защищенную передачу информации, поэтому применять r-команды категорически не рекомендуется.

SSH и OpenSSH

Протокол SSH дает возможность удаленного выполнения команд и копирования файлов с аутентификацией клиента и сервера, а также с шифрованием передаваемых данных, в том числе имени и пароля пользователя. Дополнительно обеспечивается шифрование данных X Window и перенаправление любых TCP-соединений. Существует несколько программных реализаций, в частности, коммерческий SSH и бесплатный пакет с открытым исходным кодом OpenSSH.

Принцип работы SSH

SSH представляет собой протокол транспортного уровня, аутентификации и соединения, а также программные средства безопасного доступа к компьютерам по небезопасным каналам связи (Telnet, X11, RSH, FTP). При аутентификации осуществляется асимметричное шифрование с открытым ключом (SSH1 — RSA, SSH2 — RSA/DSA), при обмене данными — симметричное шифрование. Целостность переданных данных проверяется с помощью специальных контрольных сумм. Протокол транспортного уровня работает поверх TCP и использует порт 22. В качестве ключа берется случайная строка, которую генерирует клиент, шифрует с помощью открытого ключа сервера и передает серверу. Протокол аутентификации работает поверх протокола транспортного уровня и обеспечивает аутентификацию клиента для сервера. Шифрование трафика начинается после аутентификации сервера, но до аутентификации клиента, таким образом, пароли в открытом виде не передаются. Возможно соединение произвольных портов TCP по защищенным каналам. Предусматривается возможность сжатия.

Существуют две версии протокола: SSH1 и SSH2. По своей реализации это совершенно разные протоколы. Протокол SSH2 был разработан с учетом уязвимостей, найденных в первом варианте. Однако не стоит уповать на абсолютную надежность и защищенность SSH2. Желательно отслеживать сообщения о найденных проблемах в пакетах, используемых на сервере, и обновлять их по мере выхода новых версий.

OpenSSH

Это некоммерческая реализация протокола SSH с открытым кодом. Программный пакет способен работать с протоколом SSH1 и SSH2. Имеется также поддержка r-команд.

Для большинства дистрибутивов установка пакета OpenSSH включена в стандартную инсталляцию.

Конфигурирование OpenSSH

Конфигурирование OpenSSH очень сильно зависит от вашей концепции обеспечения безопасности и необходимости поддержки старого типа протокола. Поскольку протокол SSH1 оказался уязвимым, при конфигурировании его нужно запретить. Также рекомендуется запретить r-команды и все, что с ними связано. При конфигурировании OpenSSH следует настроить сервер и клиент. Конфигурационный файл сервера называется sshd_config, а клиента — ssh_config.

Файл sshd_config

Файл sshd_config задает параметры SSH-серверу и может содержать внушительный список различных опций. Далее приведены его основные конфигурационные параметры:

- `AllowGroups <список-имен-групп-через-пробел>` — разрешает вход только пользователям из группы, указанной в списке;
- `AllowTcpForwarding yes/no` — разрешает или запрещает TCP Forwarding;
- `AllowUsers <список-имен-через-пробел>` — разрешает вход только перечисленным пользователям;
- `AuthorizedKeysFile <имя-файла-с-публичным-ключом>` — задает имя файла, содержащего публичный ключ;
- `Banner <сообщение-перед-аутентификацией>` — текст сообщения, выводимого сервером перед аутентификацией клиента;
- `Ciphers` — список алгоритмов симметричного шифрования для SSH2: `aes128-cbc`, `3des-cbc`, `blowfish-cbc`, `cast128-cbc`, `arcfour`;
- `ClientAliveInterval <секунд>` — определяет интервал в секундах, через который сервер будет проверять, произошло или нет отключение клиента;
- `ClientAliveCountMax <число>` — число неудачных проверок существования связи с пользователем до разрыва сессии;
- `DenyGroups <список-имен-групп-через-пробел>` — список групп пользователей, которым запрещено устанавливать соединение с сервером;
- `DenyUsers <список-имен-через-пробел>` — список пользователей, которым запрещено устанавливать соединение с сервером;
- `GatewayPorts no/yes` — разрешать или нет удаленным хостам доступ к перенаправленным портам;
- `HostbasedAuthentication no/yes` — разрешить или запретить аутентификацию по имени хоста (только для SSH2);
- `HostKey <имя-файла-содержащего-приватный-ключ>` — с помощью данного параметра можно указать серверу, где расположен файл, содержащий секретный ключ шифрования;
- `IgnoreRhosts yes/no` — использовать или нет файлы .rhosts и .shosts для аутентификации. Для увеличения безопасности системы рекомендуется запретить эти файлы;
- `IgnoreUserKnownHosts no/yes` — позволяет запретить использование файла ~/.ssh/known_hosts во время аутентификации rhosts+RSA;
- `KeepAlive yes/no` — позволяет задействовать механизм регулярных сообщений для проверки разрыва связи;
- `KerberosAuthentication yes/no` — позволяет запретить применение Kerberos при аутентификации;
- `KerberosOrLocalPasswd yes/no` — в том случае, если аутентификация через Kerberos не прошла, данный параметр разрешает аутентификацию на основании /etc/passwd;
- `KeyRegenerationInterval 3600` — задает интервал регенерации ключа сервера;

- `ListenAddress 0.0.0.0` — определяет, к каким адресам прислушиваться; при использовании необходимо также определить параметр `Port`;
- `LoginGraceTime <секунд>` — определяет, через сколько секунд произойдет разрыв соединения, если при аутентификации пользователь за это время не введет пароль;
- `LogLevel INFO` — заданный уровень при создании сообщений в журнал системы. Возможны следующие уровни: `QUIET`, `FATAL`, `ERROR`, `INFO`, `VERBOSE`, `DEBUG`;
- `MACs <алгоритмы-проверки-целостности-данных>` — определяет, какой алгоритм будет назначен для проверки целостности данных: `hmac-md5`, `hmac-sha1`, `hmac-ripemd160`, `hmac-sha1-96`, `hmac-md5-96`;
- `MaxStartups 10` — максимально возможное число соединений, ожидающих аутентификации;
- `PasswordAuthentication yes/no` — разрешает аутентификацию по паролю;
- `PermitEmptyPasswords no/yes` — допускает пустые пароли;
- `PermitRootLogin yes/no/without-password/forced-commands-only` — разре-шает пользователю root подключаться к серверу;
- `PidFile <имя-файла>` — задает имя файла, в котором будет храниться PID процесса сервера;
- `Port 22` — определяет, какой порт слушает сервер;
- `PrintMotd yes/no` — разрешает использование /etc/motd при входе пользователя в систему для выдачи сообщения;
- `Protocol 2` — определяет, с какой версией протокола работает сервер;
- `PubkeyAuthentication yes/no` — разрешает применение публичного ключа при аутентификации;
- `ReverseMappingCheck no/yes` — разрешает после определения адреса по имени хоста производить проверку того, что обратная зона для этого адреса указывает на тот же самый хост;
- `RhostsAuthentication no/yes` — разрешает аутентификацию только на основании файлов .rhosts или /etc/hosts.equiv;
- `RhostsRSAAuthentication no/yes` — разрешает аутентификацию на основе .rhosts- и RSA-аутентификации;
- `RSAAuthentication yes/no` — данный параметр необходим только для протокола SSH1;
- `ServerKeyBits 768` — определяет длину ключа;
- `StrictModes yes/no` — разрешает проверять права доступа к файлам с частными паролями при запуске;
- `SyslogFacility AUTH` — задает тип сообщений, передаваемых на syslog: `DAEMON`, `USER`, `AUTH`, `LOCAL0`, `LOCAL1`, `LOCAL2`, `LOCAL3`, `LOCAL4`, `LOCAL5`, `LOCAL6`, `LOCAL7`;
- `UseLogin no/yes` — разрешает использовать login для интерактивных сессий;
- `X11DisplayOffset 10` — определяет первый доступный номер дисплея при передаче X11.

Файлы на сервере, используемые при входе SSH

При входе SSH на сервере используются следующие файлы:

- /etc/nologin — при наличии этого файла запрещается вход пользователей, кроме root. Содержимое файла выдается в качестве сообщения о причине;
- /etc/hosts.allow — при компиляции с libwrap разрешает доступ;
- /etc/hosts.deny — при компиляции с libwrap запрещает доступ;
- ~/.rhosts — файл содержит пары "хост – пользователь", разделенные пробелом. Для указанного пользователя с указанного хоста разрешается заходить без ввода пароля при задании `RhostsAuthentication` и `RhostsRSAAuthentication`. Также необходим для семейства r-команд;
- ~/.shosts — аналогично файлу .rhosts, но не используется семейством r-команд;
- /etc/hosts.equiv — список хостов, с которых пользователи могут заходить, не указывая паролей, под теми же самыми именами. За именем хоста можно указывать имя конкретного пользователя. Также необходим для семейства r-команд;
- /etc/shosts.equiv — аналогично файлу hosts.equiv, но не используется семейством r-команд;
- ~/.ssh/environment — содержит пары вида "имя–значение", которые помещаются в окружение при входе.

Файлы ключей сервера

Перечислим файлы ключей сервера:

- /usr/local/etc/ssh_host_key — приватный ключ хоста;
- /usr/local/etc/ssh_host_rsa_key — приватный ключ хоста, алгоритм шифрования RSA;
- /usr/local/etc/ssh_host_dsa_key — приватный ключ хоста, алгоритм шифрования DSA;
- /usr/local/etc/ssh_host_key.pub — публичный ключ хоста;
- /usr/local/etc/ssh_host_rsa_key.pub — публичный ключ хоста, алгоритм шифрования RSA;
- /usr/local/etc/ssh_host_dsa_key.pub — публичный ключ хоста, алгоритм шифрования DSA.

Файл ssh_config

Данный файл предназначен для конфигурации SSH-клиента и разделен на секции директивами `Host`. Секция применяется при работе с хостом, удовлетворяющим шаблону секции:

- `Host` *<шаблоны>* — следующие опции применимы к хостам, подходящим под один из шаблонов; имя хоста берется из командной строки, в шаблонах возможны символы `*` и `?`;
- `BatchMode no|yes` — разрешает не запрашивать пароль/парольную фразу;
- `CheckHostIP yes|no` — позволяет дополнительно проверять адрес сервера в known_hosts;
- `Cipher 3des|blowfish` — определяет алгоритм шифрования данных;

- `Ciphers aes128-cbc, 3des-cbc, blowfish-cbc, cast128-cbc, arcfour, aes192-cbc, aes256-cbc` — определяют алгоритм шифрования данных;
- `ClearAllForwardings no|yes` — позволяет сбросить все перенаправления портов;
- `Compression no|yes` — разрешает сжатие передаваемых данных;
- `CompressionLevel <уровень-сжатия>` — определяет уровень компрессии для протокола SSH1;
- `ConnectionAttempts <число-попыток-соединения>` — задает число попыток установления соединения;
- `EscapeChar <символ>|<^символ>|none` — позволяет определить символ для замены тильды;
- `FallBackToRsh no|yes` — разрешает использовать RSH в том случае, если сервер не имеет SSH-сервера;
- `ForwardAgent no|yes` — определяет, передавать ли запрос к агенту аутентификации на удаленный хост;
- `GatewayPorts no|yes` — разрешает удаленным хостам соединяться на перенаправленные локальные порты;
- `GlobalKnownHostsFile <имя-файла>` — разрешает заменять /usr/local/etc/ssh_known_hosts указанным файлом;
- `HostKeyAlgorithms ssh-rsa,ssh-dss` — назначает алгоритмы шифрования (SSH2);
- `IdentityFile <имя-файла>` — определяет файл, хранящий RSA- или DSA-приватный ключ;
- `KeepAlive yes|no` — позволяет заметить разрыв связи или аварийное завершение на удаленном конце;
- `KerberosAuthentication yes|no` — разрешает Kerberos-аутентификацию;
- `LogLevel INFO` — определяет, какой уровень использовать при создании сообщений в журнал системы. Возможны следующие уровни: `QUIET`, `FATAL`, `ERROR`, `INFO`, `VERBOSE`, `DEBUG`;
- `MACs hmac-md5, hmac-sha1, hmac-ripemd160, hmac-sha1-96, hmac-md5-96` — устанавливает алгоритмы для создания контрольной суммы;
- `NumberOfPasswordPrompts 3` — число попыток ввода пароля пользователя;
- `PasswordAuthentication yes/no` — разрешает аутентификацию по паролю;
- `Port 22` — определяет, к какому порту будет подключаться клиент;
- `PreferredAuthentications publickey, password, keyboard-interactive` — определяет приоритеты аутентификации (SSH2);
- `Protocol <список-версий-протокола>` — задает список версий протокола в порядке предпочтительности;
- `ProxyCommand` — позволяет использовать дополнительную команду для соединения с сервером;
- `PubkeyAuthentication yes|no` — разрешает применять при аутентификации публичный ключ (SSH2);
- `RhostsAuthentication yes|no` — разрешает аутентификацию на основании файла .rhosts (SSH1);
- `StrictHostKeyChecking ask|no|yes` — разрешает не добавлять незнакомые или изменившиеся хосты в know_hosts;

- `UsePrivilegedPort yes|no` — разрешает работу с привилегированными портами для установления соединения;
- `User <имя-пользователя>` — задает имя пользователя;
- `UserKnownHostsFile <файл-known_hosts>` — определяет местоположение файла known_hosts;
- `UseRsh no|yes` — разрешает RSH в том случае, если SSH на хосте отсутствует.

Файлы ключей клиента

Перечислим файлы ключей клиента:

- ~/.ssh/identity — приватный RSA1-ключ пользователя;
- ~/.ssh/id_dsa — приватный DSA2-ключ пользователя;
- ~/.ssh/id_rsa — приватный RSA2-ключ пользователя;
- ~/.ssh/identity.pub — публичный RSA1-ключ пользователя;
- ~/.ssh/id_dsa.pub — публичный DSA2-ключ пользователя;
- ~/.ssh/id_rsa.pub — публичный RSA2-ключ пользователя.

Ключи запуска сервера SSH

Помимо конфигурационного файла, некоторые особенности функционирования сервера SSH можно задать посредством ключей запуска. Далее приведены основные ключи запуска:

- `-D` — не отсоединяться от терминала при запуске;
- `-b <бит>` — число битов ключа сервера (SSH1), по умолчанию 768;
- `-d` — переводит сервер в отладочный режим, указание нескольких ключей увеличивает объем отладочной информации;
- `-e` — разрешает выводить сообщения на stderr вместо syslog (т. е. не журналировать сообщения, а сразу выводить на стандартное устройство для вывода ошибок — обычно это текстовый терминал);
- `-f <имя-конфигурационного-файла>` — определяет положение конфигурационного файла, удобен при отладке;
- `-g <время-ожидания>` — время ожидания между вводом логина и пароля пользователя;
- `-h <файл-ключей-хоста>` — местоположение файла ключей;
- `-k <интервал>` — интервал регенерации ключа сервера;
- `-p <порт>` — назначает порт, который будет слушать сервер;
- `-q` — запрещает выдачу информации на syslog (т. е. запрещает журналирование событий);
- `-t` — с помощью этого параметра можно проверить конфигурационный файл и ключи на отсутствие ошибок;
- `-u <число>` — вместо имен хостов, превышающих `<число>`, в журнале utmp будет записываться IP-адрес: `-u0` вызывает безусловную запись IP-адресов;
- `-4` — задает протокол IPv4;
- `-6` — задает протокол IPv6.

Ключи запуска клиента SSH

Как и для сервера, для изменения некоторых параметров клиента можно воспользоваться ключами запуска:

- `-a` — запрещает перенаправление агента аутентификации;
- `-A` — разрешает перенаправление агента аутентификации;
- `-b <адрес>` — позволяет для хоста с несколькими интерфейсами указать конкретный адрес;
- `-c blowfish|3des` — задает алгоритм шифрования (SSH1);
- `-c <список-алгоритмов-шифрования-через-запятую>` — алгоритм в начале списка имеет наибольший приоритет; по умолчанию: `aes128-cbc`, `3des-cbc`, `blowfish-cbc`, `cast128-cbc`, `arcfour`, `aes192-cbc`, `aes256-cbc` (SSH2);
- `-D <локальный-порт>` — эмуляция SOCKS4-сервера по защищенному каналу;
- `-e <символ> | <^символ> | none` — определяет Escape-символ вместо тильды; `none` обеспечивает прозрачную передачу данных;
- `-f` — перейти в фоновый режим после запроса пароля или парольной фразы;
- `-F <имя-конфигурационного-файла>` — использовать указанный файл в качестве конфигурационного;
- `-g` — разрешает удаленному хосту подсоединяться к локальным перенаправленным портам;
- `-i <имя-файла>` — файл, хранящий RSA/DSA-приватный ключ;
- `-k` — запрещает перенаправление Kerberos;
- `-l <имя-пользователя>` — определяет, от имени какого пользователя устанавливается соединение;
- `-m <список-алгоритмов-обеспечения-целостности-соединения>` — задает алгоритмы подсчета контрольной суммы;
- `-n` — направить /dev/null на stdin и перейти в фоновый режим;
- `-p <порт>` — соединиться с указанным хостом на удаленном хосте;
- `-P` — использовать непривилегированный порт для исходящего соединения, чтобы обойти ограничения сетевого экрана;
- `-R <локальный-порт>:<хост>:<удаленный-порт>` — если происходит соединение на удаленный порт, то оно перенаправляется по защищенному каналу на локальный порт;
- `-s` — запуск подсистемы на сервере — например, sftp; имя подсистемы задается последним параметром;
- `-t` — требовать выделения псевдо-tty;
- `-T` — не выделять псевдо-tty;
- `-x` — запретить перенаправление X11;
- `-X` — разрешить перенаправление X11;
- `-1` — только SSH1-протокол;
- `-2` — только SSH2-протокол;
- `-4` — использовать IPv4;
- `-6` — использовать IPv6.

Программы, входящие в пакет OpenSSH

Помимо клиента и сервера, в пакет OpenSSH входят программы, предназначенные для генерации ключей, аутентификации, а также программы, призванные заменить набор r-команд.

Программа ssh-keygen

Программа ssh-keygen предназначена для генерации, преобразования и управления ключами. По умолчанию генерирует RSA-ключ. При генерации запрашивается парольная фраза. Забытую парольную фразу восстановить невозможно. Число битов по умолчанию — 1024. Имя файла для хранения публичного ключа образуется из имени файла для частного ключа добавлением суффикса .pub. Ключ хоста должен иметь пустую парольную фразу.

Возможные строки запуска:

- генерирует ключ по указанному пользователем алгоритму:

```
ssh-keygen [-t rsa1|dsa|rsa] [-b <бит>] [-N <парольная-фраза>]
[-C <комментарий>] [-f <имя-файла-записи>] [-q]
```

- изменяет комментарий:

```
ssh-keygen -c [-P <парольная-фраза>] [-C <комментарий>]
[-f <файл-с-ключами>]
```

- читает приватный или публичный ключ в формате OpenSSH и преобразует его в формат SECSH для экспорта в другие реализации SSH:

```
ssh-keygen -e [-f <файл-с-ключами>]
```

- читает приватный или публичный ключ в формате SSH2 или SECSH и преобразует его в формат OpenSSH:

```
ssh-keygen -i [-f <файл-с-ключами>]
```

- позволяет изменить парольную фразу:

```
ssh-keygen -p [-P <старая-парольная-фраза>] [-N <новая-парольная-фраза>]
[-f <файл-с-ключами>]
```

- читает приватный OpenSSH DSA ключ и выдает OpenSSH DSA публичный ключ:

```
ssh-keygen -y [-f <файл-с-ключами>]
```

Программа ssh-agent

Программа ssh-agent позволяет проводить RSA/DSA-аутентификацию. Она запускается в начале сессии и устанавливает переменные окружения, с помощью которых остальные программы могут использовать ее для автоматической аутентификации SSH. Параметром является имя команды и ее аргументы, выполнение ssh-agent останавливается при завершении команды. Если имя команды не указано, то ssh-agent запускается в фоновом режиме, а на stdout выдаются команды экспортирования необходимых переменных окружения.

Опции командной строки ssh-agent:

- `-c` — позволяет выдавать на stdout команды в стиле csh;
- `-s` — позволяет выдавать на stdout команды в стиле sh;
- `-k` — завершает работу агента — по переменной `SSH_AGENT_PID`.

Программа ssh-add

Эта программа служит для добавления приватных ключей. Она запрашивает парольную фразу, расшифровывает приватный ключ и посылает его ssh-agent. Если терминал недоступен, но определена переменная `DISPLAY`, то для ввода парольной фразы используется программа, определенная переменной `SSH_ASKPASS`. Таким образом, парольная фраза запрашивается только один раз за сеанс, а не при каждом вызове ssh/scp/sftp.

Опции командной строки ssh-add:

- имя файла — имя файла с приватным ключом (по умолчанию ~/.ssh/identity);
- `-L` — выдает публичные ключи, хранящиеся в ssh-add;
- `-d` — удаляет приватный ключ;
- `-D` — удаляет все ключи.

Программа sftp

Программа sftp (secure FTP) является клиентом для SFTP-сервера, который должен быть описан в опции `Subsystem` в конфигурационном файле sshd.

Программа sftp позволяет пересылать файлы в режиме, подобном FTP-протоколу, однако она осуществляет все операции поверх защищенного транспорта SSH. К сожалению, данный вариант FTP пока не получил широкого распространения.

Опции командной строки:

- `[`*`<пользователь>`*`@]`*`<имя-хоста>`*`[:`*`<каталог>`*`/]` — задает аналогично FTP имя пользователя, хост, к которому производится подключение, и каталог подключения;
- `-b` *`<имя-файла>`* — позволяет читать команды из файла вместо стандартного устройства ввода;
- `-C` — разрешает сжатие пересылаемых файлов;
- `-F` *`<имя-конфигурационного-файла-ssh>`* — указывает, какой конфигурационный файл использовать;
- `-o` *`<опция>`* — передается SSH.

Интерактивные команды sftp аналогичны FTP-командам:

- `bye` — разорвать соединение;
- `cd` *`<путь>`* — сменить каталог;
- `lcd` *`<путь>`* — сменить каталог;
- `chgrp gid` *`<имя-файла>`* — изменить групповой идентификатор файла на указанный в команде;
- `chmod mode` *`<имя-файла>`* — изменить атрибуты файла;
- `chown uid` *`<имя-файла>`* — изменить владельца файла;
- `exit` — выйти;
- `get [-P]` *`<имя-удаленного-файла>`* `[`*`<имя-локального-файла>`*`]` — команда для получения файла, ключ `-P` позволяет сохранить права и время создания и модификации получаемого файла;
- `help` — получить справку по командам;
- `lls [`*`<опции-ls>`* `[`*`<имя-файла>`*`]]` — получить список файлов;

- `lpwd` — пароль;
- `mkdir <имя>` — создать каталог;
- `put [-P] <имя-локального-файла> [<имя-удаленного-файла>]` — выгрузить на сервер файл, ключ `-P` позволяет сохранить права, время создания и модификации передаваемого файла;
- `pwd` — пароль;
- `quit` — выйти;
- `rename <старое-имя> <новое-имя>` — переименовать файл;
- `rmdir <имя>` — удалить каталог;
- `rm <имя-файла>` — удалить файл;
- `symlink <старое-имя> <новое-имя>` — создать символическую ссылку.

Программа scp

Программа scp, как и rcp, копирует файлы между хостами, причем оба могут быть удаленными. Способы аутентификации аналогичны SSH. Вызывает SSH для организации канала передачи данных. Имя файла записывается в виде:

`[[<пользователь>@]<хост>:]<файл>`

Опции командной строки:

- `-c <алгоритм-шифрования>` — передается SSH;
- `-i <имя-файла>` — файл с приватным ключом, передается в SSH;
- `-o <опция>` — передается SSH;
- `-p` — сохраняет время модификации, использования и права доступа к файлу;
- `-r` — позволяет рекурсивно копировать весь каталог;
- `-B` — пакетный режим — не запрашивать пароль или парольную фразу;
- `-C` — разрешает сжатие при передаче файла;
- `-F <конфигурационный-файл>` — определяет альтернативный конфигурационный файл;
- `-P <порт>` — задает порт сервера;
- `-S <программа>` — разрешает использовать указанную программу вместо SSH;
- `-4` — назначает IPv4;
- `-6` — назначает IPv6.

Программа ssh-keyscan

Программа ssh-keyscan позволяет собрать публичные ключи хостов, имена хостов задаются в качестве параметров или в файле. Опрос производится параллельно. Опции командной строки:

- `-t <тип-ключа>` — задает тип шифрования ключа (RSA1, RSA, DSA);
- `-T <секунд>` — определяет тайм-аут;
- `-f <имя-файла>` — определяет файл, в котором каждая строка содержит имя или адрес хоста;
- `-4` — задает IPv4;
- `-6` — задает IPv6;
- `-p <удаленный-порт>` — назначает порт.

Ссылки

- RFC854 — описание протокола Telnet.
- **lib.ru/LABIRINT/telnet.htm** — доступ к ресурсам Интернета в режиме удаленного терминала.
- **www.bog.pp.ru/work/ssh.html** — Bog BOS: SSH и OpenSSH: принципы работы, установка и настройка.
- **www.mnet.uz/citforum/internet/services/index.shtml** — Храмцов П. Б. Администрирование сети и сервисов Internet. Учебное пособие.
- **www.openssh.com** — сайт некоммерческой реализации SSH.
- **www.ssh.com** — сайт коммерческой реализации SSH.
- **www.tigerlair.com/ssh/faq/** — SSH FAQ.

Глава 32



Firewall

Эта глава посвящена защите сети от вторжения как извне, так и изнутри. Локальную сеть защищает комплекс программного обеспечения, известный под названием Firewall (брандмауэр, межсетевой экран). Брандмауэр позволяет "отгородить" систему (или сеть) от внешнего мира. Он предотвращает получение посторонними данных (или ресурсов) защищаемой сети, а также обеспечивает контроль внешних ресурсов, к которым имеют доступ пользователи вашей сети.

Чаще всего брандмауэр — это набор программ маршрутизации и фильтрации сетевых пакетов. Такие программы позволяют определить, можно ли пропустить данный пакет, и если можно, то отправят его точно по назначению. Для того чтобы брандмауэр мог сделать это, ему необходимо задать набор правил фильтрации. Главная цель брандмауэра — контроль удаленного доступа извне или изнутри защищаемой сети или компьютера.

Брандмауэр позволяет лишь частично решить проблемы, связанные с обеспечением безопасного функционирования вашей сети. Как бы хорошо он ни был настроен, если вы вовремя не обновили программный пакет, в котором найдена уязвимость, или кто-то узнал ваши логин и пароль — ждите больших неприятностей. Основная задача брандмауэра — допускать функционирование только тех служб, которым было явно разрешено работать в вашей сети или защищаемом компьютере. В результате мы получаем маленькую "дверцу", через которую в уютный внутренний мирок смогут попасть только те гости, которых пропустила ваша охрана, а список этих гостей рекомендуется по возможности сузить.

Основные компоненты брандмауэра:

- ❒ политика безопасности сети;
- ❒ механизм аутентификации;
- ❒ механизм фильтрации пакетов.

О практической реализации этих компонентов мы поговорим несколько позже, а пока разберемся, какие бывают брандмауэры.

Типы брандмауэров

При построении брандмауэра обычно используется компьютер (компьютеры), непосредственно подключенный к Интернету и содержащий базовый набор средств, реализующих брандмауэр. Такой компьютер иногда называют *бастионом*.

Термин "брандмауэр" может приобретать различные значения в зависимости от принципа, положенного в основу работы средств защиты, сетевой архитектуры и схемы маршрутизации. Брандмауэры обычно подразделяют на три типа:

- брандмауэр с фильтрацией пакетов;
- прикладной шлюз;
- универсальный proxy-сервер.

Брандмауэр с фильтрацией пакетов, как правило, действует на сетевом и транспортном уровнях и входит в состав операционной системы. Источник информации для фильтрации — содержимое заголовков IP-пакетов, на основе которого брандмауэр принимает решение, по какому маршруту следует направить пакет.

Прикладной шлюз реализуется посредством выбора сетевой архитектуры и конфигурации системы. Сетевой трафик никогда не проходит через компьютер, на котором выполняется прикладной шлюз. Чтобы получить доступ в Интернет, локальный пользователь должен зарегистрироваться на прикладном шлюзе. Компьютер, содержащий прикладной шлюз, может быть защищен брандмауэрами с фильтрацией пакетов как извне, так и из локальной сети.

Proxy-сервер обычно представляет собой независимое приложение, управляющее доступом к различным типам сетевых служб. Для клиентов proxy-сервер выполняет функцию сервера, предоставляющего информацию. Вместо того чтобы непосредственно обращаться к удаленным серверам, клиентские программы взаимодействуют с proxy-сервером. Получив обращение клиента, proxy-сервер устанавливает связь с удаленным узлом от своего имени, при этом он заменяет в пакете адрес клиента своим адресом. Подобный сервер может контролировать целостность данных, проверять наличие вирусов и обеспечивать выполнение правил системной политики, определяющих обмен высокоуровневыми данными.

Помимо этого, брандмауэры можно разделить по типу построения защиты:

- пороговый и его разновидность — бастионного типа;
- организующий так называемую демилитаризованную зону.

Брандмауэр порогового типа призван защитить локальную сеть от несанкционированного проникновения извне, а при соответствующей настройке и от атак изнутри. Такие брандмауэры обычно предназначены для защиты небольшой сети или даже одного компьютера. Как правило, сетевые службы, предоставляющие услуги вне локальной сети (HTTP, FTP и т. п.), размещаются на том же компьютере, что и брандмауэр.

Организация демилитаризованной зоны оправдана, когда в сети выделено несколько специальных компьютеров для интернет-сервисов, предоставляемых "большому миру", а также при отсутствии уверенности в благонадежности собственных сотрудников. Для создания демилитаризованной зоны необходимы, по меньшей мере, два брандмауэра: один для защиты демилитаризованной зоны от проникновения извне, а второй — от проникновения из вашей собственной локальной сети. Демилитаризованная зона сложнее, чем брандмауэр бастионного типа, но взамен вы получаете более высокую степень защиты ваших данных.

Брандмауэр с фильтрацией пакетов

Брандмауэр с фильтрацией пакетов представляет собой "сито" для входящих, проходящих и исходящих пакетов. В операционной системе Linux реализован брандмауэр, позволяющий контролировать ICMP-, UDP- и TCP-пакеты. Брандмауэр с фильтрацией пакетов использует набор разрешающих и запрещающих правил для входящих и исходящих пакетов. Этот набор правил определяет, какие пакеты могут проходить через конкретный сетевой интерфейс.

Брандмауэр с фильтрацией пакетов может производить с проходящим пакетом всего три действия:

- переслать пакет в узел назначения;
- удалить пакет без уведомления посылающей пакет стороны;
- вернуть передающему компьютеру сообщение об ошибке.

Несмотря на простоту таких действий, в большинстве случаев их достаточно для организации эффективной защиты. Как правило, брандмауэр устанавливается для того, чтобы контролировать данные, которыми компьютеры обмениваются с Интернетом. В результате работы фильтрующего брандмауэра отсеиваются недопустимые обращения к узлам внутренней сети и запрещается передача из внутренней сети в Интернет для пакетов, определенных правилами фильтрации.

Чтобы получить более гибкую систему, правила фильтрации пакетов составляют для каждого сетевого интерфейса, в них учитывают IP-адреса источника и получателя, номера портов TCP и UDP, флаги TCP-соединений и ICMP-сообщений. Причем правила для входящих и исходящих пакетов различаются. Это значит, что при настройке фильтрующего брандмауэра для конкретного сетевого интерфейса предусматривают отдельные правила для входящей и исходящей информации, поскольку входящие и исходящие пакеты брандмауэр обрабатывает независимо друг от друга. Списки правил, которые управляют фильтрацией сетевых пакетов, поступающих извне в локальную сеть и отправляемых из локальной сети в Интернет, принято называть *цепочками* (chains). Термин "цепочка" употребляют потому, что при проверке пакета правила применяются последовательно одно за другим, пока не обнаружится подходящее правило для сетевого пакета или список правил не будет исчерпан.

Описанный механизм фильтрующего брандмауэра довольно эффективен, однако он не обеспечивает полной безопасности локальной сети. Брандмауэр — это всего лишь один из элементов общей схемы защиты. Анализ заголовков сетевых пакетов — операция слишком низкого уровня, для того чтобы реально выполнять аутентификацию и контролировать доступ. В процессе фильтрации пакетов практически невозможно распознать отправителя сообщения и проанализировать смысл передаваемой информации. Из всего набора данных, пригодных для аутентификации, на рассматриваемом уровне доступен только IP-адрес отправителя, который очень легко подделать, на чем и базируется множество способов сетевых атак. Несмотря на то, что средства фильтрации пакетов позволяют эффективно контролировать обращение к портам, использование протоколов обмена и содержимое пакетов, проверку данных необходимо продолжить на более высоком уровне.

Политика организации брандмауэра

Существуют два основных подхода при построении брандмауэров:

- запрещается прохождение всех пакетов, пропускаются лишь те, которые удовлетворяют явно определенным правилам;
- разрешается прохождение всех пакетов, за исключением пакетов, удовлетворяющих определенным правилам.

Или образно выражаясь, запрещено все, что не разрешено, и разрешено все, что не запрещено.

С практической точки зрения предпочтительнее подход, при котором поступающий пакет по умолчанию отвергается (запрещено все, что не разрешено). В этом случае организация безопасности сети достигается достаточно просто, но с другой стороны, приходится предусматривать возможность обращения к каждой сетевой службе и использование каждого конкретного протокола. Это означает, что администратор сети, занимающийся настройкой брандмауэра, должен точно знать, какие протоколы применяются в его локальной сети. При реализации подхода, предусматривающего запрет по умолчанию, приходится предпринимать специальные меры всякий раз, когда необходимо разрешить доступ к какому-то ресурсу, однако эта модель более надежна, чем противоположный вариант.

Политика разрешения по умолчанию позволяет добиться функционирования системы малыми усилиями, но при этом необходимо предусмотреть каждую конкретную ситуацию, требующую запрета доступа. Может случиться так, что необходимость внесения запретов станет ясна лишь тогда, когда в результате несанкционированного доступа сети будет нанесен значительный ущерб.

В обоих случаях конфигурация брандмауэра основана на цепочках правил. Каждая цепочка представляет собой набор правил, заданных явным образом, и политику по умолчанию. Пакет проверяется на соответствие каждому из правил, а правила выбираются из списка последовательно до тех пор, пока не будет обнаружено соответствие сетевого пакета одному из них. Если пакет не удовлетворяет ни одному из заданных правил, с сетевым пакетом производятся действия, определенные политикой по умолчанию.

В процессе работы брандмауэр может *пропустить* сетевой пакет (ACCEPT), *запретить* прохождение сетевого пакета (DENY) либо отказать сетевому пакету в прохождении, т. е. *отклонить* его (REJECT). С прохождением сетевого пакета все ясно, а чем же различаются запрет и отклонение пакета? При отклонении сетевого пакета (REJECT) сам пакет удаляется, а его отправителю возвращается ICMP-сообщение об ошибке. При запрете прохождения сетевого пакета (DENY) сам пакет удаляется, но отправитель об этом не оповещается.

В большинстве случаев запрет сетевого пакета считается лучшим решением, чем отказ в прохождении. Во-первых, отправка сообщения об ошибке увеличивает сетевой трафик, а во-вторых, сообщения об ошибке могут быть использованы для организации атаки с целью вывода из строя сервера. Помимо этого, любое ответное действие на "неправильные" пакеты предоставляет взломщику дополнительную информацию о конфигурации вашей системы.

Фильтрация сетевых пакетов

Рассмотрим, на основании каких данных можно фильтровать входящие и исходящие сетевые пакеты, а также как определять "неправильные" сетевые пакеты.

Фильтрация входящих пакетов

Анализ построения брандмауэра логично начать с входящих пакетов, поскольку именно извне обычно происходит проникновение в сеть.

Фальсификация исходящего адреса и недопустимые адреса

Рассмотрим признаки, по которым можно однозначно судить о поддельности сетевого пакета, поступающего из Интернета, или о проблемах прикладного программного обеспечения. На основании этих признаков нужно будет задать соответствующие правила фильтрации, чтобы ваш брандмауэр, обнаружив такой "неправильный" исходящий адрес в пакете, мог запретить прохождение сетевого пакета.

1. В заголовке сетевого пакета в качестве исходного указан адрес вашего компьютера. В процессе сетевого обмена невозможна ситуация, при которой сетевой пакет, отправленный с вашего компьютера, вернулся бы через внешний интерфейс. Следовательно, такой сетевой пакет — поддельный.
2. Исходящим указан IP-адрес, попадающий в зарезервированный диапазон адресов, предназначенных для внутреннего применения. Согласно правилам распределения IP-адресов в каждом из классов IP-адресов A, B и C существуют группы IP-адресов, выделенных для организации внутренних локальных сетей. В Интернете эти адреса не используются. При правильной конфигурации программного обеспечения через внешний порт не может прийти пакет с адресом источника, попадающий в один из перечисленных далее диапазонов:
   - класс A — в диапазоне от 10.0.0.0 до 10.255.255.255;
   - класс B — в диапазоне от 172.16.0.0 до 172.31.255.255;
   - класс C — в диапазоне от 192.168.0.0 до 192.168.255.255.
3. Исходящим указан IP-адрес класса D, предназначенный для группового вещания. Адреса класса D, специально выделенные для организации группового вещания, находятся в диапазоне от 224.0.0.0 до 239.255.255.255 и ни при каких обстоятельствах не могут выступать в качестве адреса источника.
4. Исходящим указан зарезервированный IP-адрес класса E с адресами от 240.0.0.0 до 247.255.255.255. Если брандмауэр встретит пакет с исходным адресом класса E, он должен предотвратить попадание такого пакета в локальную сеть.
5. Исходящий IP-адрес принадлежит интерфейсу обратной петли, предназначенному для локальных сетевых служб. Как правило, для обращения к интерфейсу обратной петли служит адрес 127.0.0.1, а вообще за интерфейсом локальной сети зарезервирована целая подсеть 127.*x.x.x*. Адрес интерфейса обратной петли не может присутствовать в заголовке пакета, полученного через внешний сетевой интерфейс.
6. Исходящий IP-адрес — некорректный широковещательный адрес. Широковещательный адрес — это специальный тип адреса, определяющий передачу сетевого пакета на все компьютеры в сети. В качестве исходного адреса при широковещательной передаче может выступать обычный IP-адрес или адрес 0.0.0.0.

Фильтрация на основе адреса источника

При фильтрации пакетов единственный способ идентификации отправителя сетевого пакета — проверка IP-адреса источника в заголовке пакета. Один из самых распространенных приемов при организации сетевых атак — фальсификация сетевых пакетов, при которой отправитель заменяет свой IP-адрес в заголовке сетевого пакета другим значением. Для подмены может быть выбран несуществующий или реальный IP-адрес, принадлежащий другому узлу.

Блокирование ненадежных узлов

Еще одна схема фильтрации, основанная на анализе IP-адресов источников, — блокирование доступа с компьютеров, IP-адреса которых попадают в определенный диапазон. Как правило, так отсекают "подозрительные" компьютеры и целые сети, в частности обычно это происходит с сетями различных учебных заведений или разнообразных интернет-клубов, поскольку именно там молодежь любит "пошалить" в сети.

Работа с ограниченным набором удаленных узлов

Если вы организуете корпоративную сеть, то не исключена такая настройка брандмауэра, что некоторые типы пакетов принимаются только в том случае, если они были отправлены с компьютеров с определенными адресами. Например, для организации системы передачи приватной информации.

Фильтрация на основе адреса назначения

В большинстве случаев фильтрация на основе адреса назначения выполняется автоматически. Сетевой интерфейс игнорирует пакеты, не адресованные непосредственно ему. Исключение — широковещательные пакеты, адресованные всем узлам сети.

Фильтрация на основе порта источника

Номер порта источника, содержащийся в заголовке пакета, предназначен для идентификации программы-отправителя сетевого пакета, выполняющейся на удаленном узле. В запросах удаленных клиентов к вашему серверу содержатся различные номера портов, а в ответах сервера клиентам — один и тот же порт.

Фильтрация на основе порта назначения

Порт назначения определяет программу на вашем компьютере, которой предназначен пакет. В запросах удаленных клиентов, передаваемых на сервер, содержится один и тот же порт назначения, а в ответах сервера клиентам — различные номера портов.

Фильтрация на основе информации о состоянии TCP-соединения

В правилах обработки сетевых пакетов возможны флаги, определяющие состояние TCP-соединения, поскольку любое сетевое соединение проходит через определенные состояния. Состояния клиента и сервера различаются между собой.

В первом пакете, отправленном удаленным клиентом, установлен флаг `SYN`, а флаг `ACK` сброшен. Передача такого пакета является началом в установлении TCP-

соединения. Во всех последующих сетевых пакетах, предаваемых клиентом, установлен флаг ACK, а флаг SYN сброшен.

Удаленные серверы передают пакеты только в ответ на предыдущие обращения клиентов. В каждом пакете, поступившем от удаленного сервера, должен быть установлен флаг ACK, поскольку TCP-соединение никогда не устанавливается по инициативе сервера.

Анализ флагов позволяет отсеивать "неправильные" сетевые пакеты, которые могут являться признаком сетевой атаки.

Фильтрация исходящих пакетов

Фильтрация исходящих сетевых пакетов позволит исключить попадание в Интернет сетевых пакетов, передаваемых по локальной сети, а также избежать нежелательных обращений к серверам с узлов локальной сети. Источником таких обращений могут быть неверно сконфигурированные или вредоносные программы, запускаемые пользователями на своих компьютерах.

Фильтрация на основе адреса источника

При этом необходимо сформировать правила фильтрации так, чтобы пакет, в котором указан адрес источника, не совпадающий ни с одним из адресов компьютеров вашей локальной сети, не был пропущен брандмауэром. Это может вызвать некоторые затруднения, если в вашей организации разветвленная локальная сеть или IP-адреса выдаются динамически. Однако такие проблемы решаемы.

Фильтрация на основе адреса назначения

Как уже упоминалось ранее, возможна ситуация, при которой вам потребуется ограничить передачу сетевых пакетов за пределы локальной сети адресами отдельных сетей или отдельных компьютеров. Эти адреса или диапазоны адресов можно указать в правилах, задаваемых брандмауэру.

Фильтрация на основе порта источника

Проверять порты, указанные в заголовках сетевых пакетов, можно как для клиентов, запущенных в локальной сети, так и для серверов. Такая проверка позволяет убедиться в том, что программы работают корректно и защищают Интернет от попадания в него внутреннего трафика локальной сети.

Пакеты, передаваемые сервером, обязательно должны содержать в заголовке порт источника, совпадающий с номером порта, выделенным для службы данного типа. Проверка номера порта представляет собой анализ конфигурации сетевых протоколов.

Фильтрация на основе порта назначения

Поскольку локальные клиенты могут обращаться к удаленным серверам лишь по конкретным номерам портов, фильтрация исходящих пакетов является одновременно средством контроля за использованием протоколов. Запрет прохождения сетевых пакетов на основе порта назначения не дает возможности пользователям локальной сети сканировать порты удаленных компьютеров, ведь обычно сканирование портов — предвестник сетевой атаки.

Защита локальных служб

Как правило, локальные сервисы функционируют только внутри вашей сети, и предоставление доступа к этим службам извне нецелесообразно, а зачастую и вредно. Самый простой способ уберечься от проникновения в систему через один из сервисов — запретить доступ к сервису извне. Однако существуют службы, которые могут вызвать серьезные проблемы при запрете доступа, например ICQ.

Один из способов защитить службы, предназначенные для внутреннего использования, — отказаться от размещения соответствующих серверов на компьютерах, доступных из глобальной сети. Однако в небольших сетях обычно существует один-единственный сервер, зачастую выполняющий роль брандмауэра, поэтому в некоторых случаях компромиссы неизбежны.

Для защиты сервера от обращений из Интернета можно применить брандмауэр, выполняющий фильтрацию пакетов по порту назначения. Наличие такого брандмауэра позволяет запускать в локальной сети большое количество служб, не подвергая серьезной опасности сетевые ресурсы.

Программа ipchains

Как уже упоминалось ранее, брандмауэр — это набор программных средств для организации защиты вашей сети. Бóльшая часть функциональности брандмауэра интегрирована в ядре операционной системы Linux, но для создания и управления цепочками правил созданы внешние программы. Для ядер версий 2.0–2.2 была разработана программа ipchains. В ядрах версий 2.4 можно использовать как ipchains, так и iptables, в ядрах версий 2.6 — только iptables.

Правила фильтрации пакетов, составляющих цепочки input, output и forward (входящие, исходящие и переадресация), содержатся во внутренних таблицах ядра операционной системы Linux. Каждое правило можно включить в начало цепочки или добавить в ее конец. Для определенности будем считать, что все правила, определяемые в данной главе, добавляются в конец цепочки. Порядок задания правил определяет последовательность их включения в цепочку и применения к каждому пакету.

При поступлении информационного пакета на сетевой интерфейс извне анализируется содержимое заголовка этого пакета. Правила, принадлежащие цепочке input данного сетевого интерфейса, применяются последовательно одно за другим до тех пор, пока не будет найдено такое, которому удовлетворяет данный сетевой пакет. Соответственно, каждый сетевой пакет, отправляемый вовне, проверяется на соответствие правилам, содержащимся в цепочке output сетевого интерфейса. При обнаружении первого соответствия правилу проверка прекращается и к пакету применяется действие, указанное в составе правила: ACCEPT, REJECT или DENY. Если пакет не удовлетворяет ни одному из правил, содержащихся в цепочке, вступает в действие политика по умолчанию. Таким образом, при работе брандмауэра пакет обрабатывается по первому из правил, которому он удовлетворяет.

Программе ipchains параметры передаются при вызове в командной строке. Формат командной строки:

```
ipchains  -A|I [<цепочка>] [-i <интерфейс>] [-p <протокол>] [  [!]  -y]
[-s <адрес> [<порт> [: <порт>]]]          [-d <адрес> [<порт> [: <порт>]]] — j
<действие> [l]
```

В правилах, управляющих работой брандмауэра, предусматривается проверка адреса источника и адреса назначения. Для сравнения могут использоваться IP-адрес узла, диапазон IP-адресов, символьное имя узла и имя домена.

Программа ipchains позволяет задавать после IP-адреса *дескриптор маски* — целое число от 0 до 32, определяющее число битов в маске. Дескриптор маски указывает, сколько старших битов адреса узла должны в точности совпадать с адресом, заданным в составе правила. Дескриптор маски, равный 32, означает, что адрес узла должен полностью совпадать с адресом, указанным в правиле. Если дескриптор маски отсутствует, считается, что он равен 32. Так, адрес 192.168.0.45 означает то же самое, что и выражение 192.168.0.45/32.

Опции ipchains

В табл. 32.1 приведены наиболее часто применяемые опции программы ipchains.

Таблица 32.1. *Опции программы ipchains*

Опция	Описание
`-A [<цепочка>]`	Добавляет правило к концу цепочки. Используются встроенные цепочки input, output и forward. Если цепочка не указана, правило добавляется ко всем цепочкам
`-I [<цепочка>]`	Включает правило в начало цепочки. Используются встроенные цепочки input, output и forward. Если цепочка не указана, правило включается во все цепочки
`-i <интерфейс>`	Определяет сетевой интерфейс, к которому должно применяться данное правило. Если сетевой интерфейс не указан, правило применяется ко всем сетевым интерфейсам
`-p <протокол>`	Определяет протокол семейства TCP/IP, к которому должно применяться правило. Если опция `-p` не указана, правило применяется ко всем протоколам. В качестве имен протоколов могут быть заданы `tcp`, `udp`, `icmp` и `all`. Разрешается использование имен и числовых значений, указанных в файле /etc/protocols
`-y`	В пакете, содержащем запрос на установление TCP-соединения, флаг `SYN` должен быть установлен, а флаг `ACK` — сброшен. Если данная опция не указана, состояния флагов `SYN` и `ACK` не проверяются
`! -y`	В пакете, который передается в ответ на запрос на установление TCP-соединения, а также во всех последующих пакетах флаг `ACK` должен быть установлен. Если опция `! -y` не указана, состояние флага `ACK` не проверяется

Таблица 32.1 *(окончание)*

Опция	Описание
`-s <адрес> [<порт>]`	Определяет исходящий адрес пакета. Если исходящий адрес не указан, обрабатываются пакеты, переданные с любого узла. Если указан порт или диапазон портов, правило применяется только к пакетам, содержащим заданный номер порта. Если порт не указан, правило применяется ко всем пакетам, независимо от номера порта источника. При указании диапазона портов задают начальный и конечный номера, разделенные двоеточием (например, `1024:65535`). Если в опции `-s` задается порт, адрес также должен быть указан
`-d <адрес> [<порт>]`	Определяет адрес назначения пакета. Если адрес назначения не указан, обрабатываются все пакеты, передаваемые любому узлу. Если указан порт или диапазон портов, правило применяется только к пакетам, содержащим заданный номер порта. Если порт не указан, правило применяется ко всем пакетам, независимо от номера порта назначения. При указании диапазона портов задают начальный и конечный номера, разделенные двоеточием (например, `1024:65535`). Если в опции `-d` задается порт, адрес также должен быть указан
`-j <действие>`	Определяет действие, которое должно быть выполнено над пакетом (ACCEPT, REJECT или DENY). Для цепочки forward данный параметр также может принимать значение `MASQ` (masquerade — маскировка)
`-l`	Если пакет удовлетворяет правилу, в файл протоколов (по умолчанию /var/log/messages) должно быть записано информационное сообщение ядра

Символьные константы

Чтобы улучшить удобство составления и сопровождения правил фильтрации в сценариях брандмауэра, рекомендуется указывать символьные имена. В табл. 32.2 приведены некоторые символьные константы, используемые при конфигурировании брандмауэра.

Таблица 32.2. *Символьные константы, используемые при описании правил фильтрации*

Константа	Описание
`EXTERNAL_INTERFACE = "eth0"`	Сетевой интерфейс, подключенный к Интернету
`Internal_ INTERFACE = "eth1"`	Сетевой интерфейс, подключенный к локальной сети (для брандмауэра бастионного типа)
`LAN_1="192.168.1.0/24"`	Диапазон адресов внутренней сети
`LAN_IPADDR_1="192.168.1.1"`	Адрес внутреннего интерфейса
`LOOPBACK_INTERFACE = "lo"`	Интерфейс локальной петли
`IPADDR = "ipaddress"`	Адрес вашего компьютера

Таблица 32.2 *(окончание)*

Константа	Описание
ANYWHERE ="any/0"	Произвольный адрес
MY_ISP = " ip range"	Диапазон адресов провайдера
LOOPBACK="127.0.0.0/8"	Диапазон адресов обратной петли
CLASS_A ="10.0.0.0/8"	Адреса класса A для локальных сетей
CLASS_B ="172.16.0.0/12"	Адреса класса B для локальных сетей
CLASS_C ="192.168.0.0/16"	Адреса класса C для локальных сетей
CLASS_D_MULTICAST ="224.0.0.0/4"	Адреса класса D для группового вещания
Class_E_Reserved_Net ="240.0.0.0/5"	Адреса класса E. Зарезервировано
BROADCAST_SRC ="0.0.0.0"	Исходящий широковещательный адрес
BROADCAST_DEST ="255.255.255.255"	Широковещательный адрес
NAMESERVER = "mydns"	Адрес DNS-сервера
SMTP_GATEWAY="isp.server"	Адрес почтового шлюза провайдера
POP_SERVER="isp.server"	Адрес POP-сервера провайдера
NEWS_SERVER="isp.server"	Адрес NEWS-сервера провайдера
IMAP_SERVER="isp.server"	Адрес IMAP-сервера провайдера
PRIVPPORTS="0:1023"	Номера привилегированных портов
UNPRIVPORTS="1024:65535"	Номера непривилегированных портов
SSH_PORTS="1000:1023"	Номера привилегированных портов для протокола SSH — не более 24-х одновременно возможных соединений

Создание правил фильтрации

Теперь рассмотрим создание правил фильтрации для нашего брандмауэра. Допустим, что у нас среднестатистический брандмауэр, который полностью удовлетворяет потребностям одного компьютера или малой локальной сети. Для более серьезных случаев вы сможете оформить свой набор правил на основе приведенных далее рекомендаций.

Удаление существующих правил

Начиная создание своих правил фильтрации, обязательно удалите уже существующие правила. Это необходимо с точки зрения элементарной предосторожности — вдруг в системе уже определены некоторые правила фильтрации, идущие вразрез с вашей политикой безопасности.

Удаление правил фильтрации называется *сбросом цепочки*. Для сброса встроенных цепочек не обязательно обращаться к ним явно. Все три цепочки — input, output и forward — можно сбросить с помощью одной команды:

```
ipchains -F
```

Определение политики по умолчанию

После удаления правил фильтрации автоматически устанавливается политика фильтрации сетевых пакетов по умолчанию, согласно которой разрешается прохождение всех сетевых пакетов. Таким образом, до тех пор, пока вы не внесете изменения в политику фильтрации сетевых пакетов, непосредственно фильтрация производиться не будет.

Для обеспечения безопасности вашей операционной системы нужно так выбрать политику по умолчанию, чтобы входящие сетевые пакеты удалялись без передачи сообщений на хосты, посылающие сетевые пакеты. Исходящим сетевым пакетам необходимо отказать в прохождении, а компьютеры, с которых эти сетевые пакеты были отправлены, должны получать ICMP-сообщения об ошибке. Обратившись к компьютеру вашей сети, программа, выполняющаяся на одном из компьютеров, размещенных в Интернете, не получит никакой информации о том, существует ли сервер, указанный в запросе. Такая же программа на локальном компьютере сразу получит сообщение о том, что операция, которую она собиралась выполнить, недопустима.

В следующих трех строках мы задаем уничтожение сетевых пакетов для входящей цепочки и их отклонение для исходящей цепочки и цепочки маршрутизации; компьютеры, посылающие сетевые пакеты, получат уведомление об ошибке:

```
ipchains -P input   DENY
ipchains -P output  REJECT
ipchains -P forward REJECT
```

После ввода в действие приведенных ранее правил весь сетевой трафик оказывается блокированным, в том числе и весь трафик, проходящий через интерфейс обратной петли.

Разрешение прохождения пакетов через интерфейс обратной петли

Поскольку в предыдущем разделе мы заблокировали весь сетевой трафик, автоматически возникнут проблемы с рядом программ, исполняемых на вашем компьютере, т. к. многим из них для нормального функционирования нужен интерфейс обратной петли. Поэтому мы должны обеспечить прохождение всего сетевого трафика через интерфейс обратной петли. Поскольку этот интерфейс недоступен из-за пределов системы, подобные установки не могут повлечь за собой нежелательных последствий.

Правила, разрешающие прохождение сетевых пакетов без ограничений, очень просты. В данном случае нужно нейтрализовать влияние политики по умолчанию на интерфейс обратной петли. Для этого введем следующие правила:

```
ipchains — A  input    -i $LOOPBACK_INTERFACE -j ACCEPT
ipchains — A  output  -i $LOOPBACK_INTERFACE -j ACCEPT
```

Таким простым способом прохождение трафика через интерфейс обратной петли будет восстановлено.

Запрет прохождения пакетов с фальсифицированными адресами

Как уже упоминалось ранее, фальсификация адресов — один из признаков сетевых атак, поэтому следует бороться с сетевыми пакетами, имеющими фальшивый адрес. Первое и очевидное правило — запретить прием сетевых пакетов, якобы отправленных с внешнего интерфейса вашего узла:

```
ipchains -A input  -i  $EXTERNAL_INTERFACE  -s  $IPADDR  -j  DENY  -1
```

Это правило отсекает входящие сетевые пакеты, содержащие в качестве адреса источника сетевой адрес вашего внешнего интерфейса. В том случае, если вы посылаете сетевой пакет на свой компьютер, он пройдет не через внешний сетевой интерфейс, а через интерфейс обратной петли. Если система настроена нормально, сетевой пакет, направленный на локальный компьютер, никогда не попадет на внешний интерфейс. В противном случае — либо у вас проблемы с сетевыми настройками, либо кто-то пытается пробраться к вам в систему, поскольку весь сетевой трафик, идущий через интерфейс обратной петли, проходит внутри системы, и любой сетевой пакет, содержащий такой адрес, является поддельным.

Следующие два правила запрещают пакеты, содержащие в качестве исходящего адреса интерфейс обратной петли:

```
ipchains -A input    -i $EXTERNAL_INTERFACE  -s  $LOOPBACK  -j DENY
ipchains -A output  -i  $EXTERNAL_INTERFACE  -s  $LOOPBACK  -j DENY  -l
```

Далее необходимо отсечь сетевые пакеты, исходящие адреса которых попадают в диапазон IP-адресов, выделенных для внутренних сетей. Маршрутизаторы не должны обрабатывать пакеты с исходящими адресами, принадлежащими внутренним сетям.

Тем же самым ограничениям должны подвергаться и исходящие сетевые пакеты, у которых адреса назначения попадают в диапазон IP-адресов, выделенных для использования во внутренних сетях, поскольку в Интернете не могут существовать адреса, предназначенные исключительно для локальных сетей.

Приведенные в листинге 32.1 наборы правил запрещают прохождение входящих и исходящих сетевых пакетов в случае, если адрес источника или адрес назначения принадлежит диапазонам сетевых адресов классов A, B и C, выделенных для локальных сетей.

Листинг 32.1

```
# Запретить прохождение сетевых пакетов,
# которые содержат адрес источника,
# принадлежащий диапазону адресов класса A,
# предназначенных для внутреннего использования.
ipchains -A input    -i  $EXTERNAL_INTERFACE  -s  $CLASS_A  -j DENY
ipchains -A input    -i  $EXTERNAL_INTERFACE  -d  $CLASS_A  -j DENY
ipchains -A output  -i  $EXTERNAL_INTERFACE  -s  $CLASS_A  -j DENY  -l
ipchains -A output  -i  $EXTERNAL_INTERFACE  -d  $CLASS_A  -j DENY  -l

# Запретить прохождение сетевых пакетов,
# которые содержат адрес источника,
# принадлежащий диапазону адресов класса B,
# предназначенных для внутреннего использования.
ipchains -A input    -i  $EXTERNAL_INTERFACE  -s  $CLASS_B  -j DENY
ipchains -A input    -i  $EXTERNAL_INTERFACE  -d  $CLASS_B  -j DENY
ipchains -A output  -i  $EXTERNAL_INTERFACE  -s  $CLASS_B  -j DENY  -l
ipchains -A output  -i  $EXTERNAL_INTERFACE  -d  $CLASS_B  -j DENY  -l

# Запретить прохождение сетевых пакетов,
# которые содержат адрес источника,
# принадлежащий диапазону адресов класса C,
# предназначенных для внутреннего использования.
ipchains -A input    -i  $EXTERNAL_INTERFACE  -s  $CLASS_C  -j DENY
ipchains -A input    -i  $EXTERNAL_INTERFACE  -d  $CLASS_C  -j DENY
ipchains -A output  -i  $EXTERNAL_INTERFACE  -s  $CLASS_C  -j DENY  -l
ipchains -A output  -i  $EXTERNAL_INTERFACE  -d  $CLASS_C  -j DENY  -l
```

Правила, необходимые для блокирования сетевых пакетов, содержащих недопустимые широковещательные адреса:

```
ipchains -A input  -i $EXTERNAL_INTERFACE -s  $BROADCAST_DEST  -j DENY -l
ipchains -A input  -i $EXTERNAL_INTERFACE -d  $BROADCAST_SRC  -j DENY  -l
```

Первое из приведенных правил запрещает пакеты с исходящим адресом 255.255.255.255. Второе правило запрещает пакеты с адресом назначения 0.0.0.0. Подобные пакеты появляются не в результате ошибки, а свидетельствуют о попытке атаки на вашу сеть.

Адреса группового вещания могут применяться лишь в качестве адреса назначения. Следующие правила выявляют фальсифицированные сетевые пакеты и фиксируют случаи их появления в файлах протоколов:

```
# Запретить пакеты, содержащие адреса класса D.
ipchains -A input -i $EXTERNAL_INTERFACE -s $CLASS_D_MULTICAST -j DENY -l
ipchains -A output -i $EXTERNAL_INTERFACE -s $CLASS_D_MULTICAST -j REJECT -l
```

Групповое вещание реализуется на основе протокола UDP. В сетевых пакетах, передаваемых посредством группового вещания, адрес назначения отличается от пакетов, которые передаются в процессе обычного обмена между двумя узлами. Правило, запрещающее передачу сетевых пакетов группового вещания с локального узла:

```
ipchains -A output -i $EXTERNAL_INTERFACE -d $CLASS_D_MULTICAST -j REJECT  -l
```

Фильтрация ICMP-сообщений

Сообщения ICMP передаются при возникновении различных ситуаций, в том числе ошибочных. Они генерируются программами, анализирующими состояние сети, например ping или traceroute. В табл. 32.3 приведены ICMP-сообщения, которые могут представлять интерес для администратора сети.

Таблица 32.3. *Часто встречающиеся ICMP-сообщения*

Тип сообщения	Символьное имя	Описание
0	`Echo Reply`	Отклик программы ping
3	`Destination Unreachable`	Сообщение об ошибке: один из маршрутизаторов по пути следования сетевого пакета не может доставить данные на следующий узел
4	`Source Quench`	Сообщение, предназначенное для управления потоком между двумя маршрутизаторами или между маршрутизатором и обычным узлом
5	`Redirect`	Если маршрутизатор знает о наличии более короткого пути, данное сообщение возвращается узлу, с которого был передан сетевой пакет
8	`Echo Request`	Запрос программы ping
11	`Time Exceeded`	Данное сообщение передается, когда количество узлов, через которые прошел сетевой пакет, превышает максимально допустимое
12	`Parameter Problem`	В заголовке сетевого пакета была обнаружена недопустимая запись

Сообщения об ошибках и управляющие сообщения

При настройке брандмауэра необходимо обеспечить прохождение четырех типов сообщений:

- `Source Quench` — подавление источника;
- `Parameter Problem` — некорректный параметр;
- `Destination Unreachable` (подтип `Fragmentation Needed`) — узел назначения недоступен (для входящих сообщений);
- `Destination Unreachable` (подтип `Fragmentation Needed`) — узел назначения недоступен (для исходящих сообщений).

Еще четыре типа ICMP-сообщений также можно пропустить через брандмауэр. Это `Echo Request` (эхо-запрос), `Echo Reply` (эхо-ответ), различные подтипы исходящих сообщений `Destination Unreachable`, а также сообщение `Time Exceeded` (превышение времени). Остальные сообщения желательно игнорировать, чтобы они были удалены в соответствии с политикой по умолчанию.

Из всех сообщений, которые следует игнорировать, в табл. 32.3 приведено только `Redirect` (перенаправление). Данное сообщение позволяет организовать атаку с целью вывода из строя сервисных средств. Остальные типы сообщений в основном предназначены для организации взаимодействия маршрутизаторов.

Далее описываются типы сообщений, поддержка которых необходима для обеспечения работы хостов локальной сети.

Управляющее сообщение *Source Quench*

ICMP-сообщение типа `Source Quench` (подавление источника) передается в тех случаях, когда маршрутизатор отправляет пакеты быстрее, чем принимающий узел может их обработать. `Source Quench` — одно из простейших средств контроля обмена данными на сетевом уровне. Обычно такими сообщениями обмениваются компьютеры, непосредственно связанные между собой.

Правила, разрешающие прохождение входящих и исходящих ICMP-сообщений `Source Quench`:

```
ipchains -A input -i $EXTERNAL_INTERFACE -p icmp -s $ANYWHERE 4 -d $IPADDR -j ACCEPT
ipchains -A output -i $EXTERNAL_INTERFACE -p icmp -s $ANYWHERE 4 -d $IPADDR -j ACCEPT
```

Если компьютер, которому предназначены пакеты, возвращает сообщение `Source Quench`, передающий узел должен снизить скорость обмена. Со временем он начинает повышать скорость передачи данных и делает это до тех пор, пока не получает следующее сообщение `Source Quench`.

Сообщение *Parameter Problem*

ICMP-сообщение типа `Parameter Problem` (некорректный параметр) возвращается в том случае, когда в заголовке сетевого пакета содержится недопустимая запись либо если контрольная сумма заголовка сетевого пакета не соответствует контрольной сумме, указанной передающим хостом.

Следующие правила разрешают прохождение входящих и исходящих ICMP-сообщений `Parameter Problem`:

```
ipchains -A input -i $EXTERNAL_INTERFACE -p icmp -s $ANYWHERE 12 -d $IPADDR -j ACCEPT
pchains -A output -i $EXTERNAL_INTERFACE -p icmp -s $ANYWHERE 12 -d $IPADDR -j ACCEPT
```

Сообщение об ошибке *Destination Unreachable*

ICMP-сообщение типа `Destination Unreachable` (узел назначения недоступен) представляет собой сообщение об ошибке. В заголовке пакета данного типа содержится код, обозначающий ошибку, которая имела место.

Следующие правила разрешают прохождение входящих и исходящих ICMP-сообщений `Destination Unreachable`:

```
ipchains -A input -i $EXTERNAL_INTERFACE -p icmp -s $ANYWHERE 3 -d $IPADDR -j ACCEPT
ipchains -A output -i $EXTERNAL_INTERFACE -p icmp -s $ANYWHERE 3 -d $IPADDR -j ACCEPT
```

С точки зрения безопасности это весьма неоднозначный пакет, поскольку сообщения такого типа позволяют собирать информацию об адресах узлов и портах. Кроме того, сообщения `Destination Unreachable` можно использовать для организации атаки с целью вывода узла из строя.

Тем не менее подтип `Fragmentation Needed` сообщения `Destination Unreachable` необходим для нормальной работы сетевых средств. С его помощью взаимодействующие узлы договариваются об особенностях разбиения передаваемых сетевых пакетов на фрагменты.

Если требуется, чтобы компьютеры вашей локальной сети отвечали на входящие запросы программы traceroute, нужно разрешить передачу исходящих пакетов, содержащих подтип `Port Unreachable` сообщения `Destination Unreachable`.

Сообщение *Time Exceeded*

ICMP-сообщение типа `Time Exceeded` (превышение времени) передает сведения о том, что число узлов, через которые проходил сетевой пакет по пути его следования, превысило максимально допустимое значение. В настоящее время сообщение `Time Exceeded` обычно передается в ответ на UDP-запрос программы traceroute.

Если требуется, чтобы ваша система отвечала на входящие запросы traceroute, необходимо разрешить передачу исходящих ICMP-сообщений `Time Exceeded`:

```
ipchains -A input -i $EXTERNAL_INTERFACE -p icmp -s $ANYWHERE 11 -d $IPADDR -j ACCEPT
ipchains -A output -i $EXTERNAL_INTERFACE -p icmp -s $IPADDR 11 -d $MY_ISP -j ACCEPT
```

Приведенные ранее правила допускают обращения traceroute лишь с компьютера провайдера. Если вам необходима traceroute на локальном узле, вы должны разрешить входящие сообщения `Time Exceeded`. Так как описываемая конфигурация брандмауэра не является маршрутизатором общего назначения, сообщения `Time Exceeded` используются только с описанной ранее целью.

Программа ping: сообщения *Echo Request* и *Echo Reply*

Программа ping проверяет связь с конкретными узлами сети с помощью ICMP-сообщений двух типов: `Echo Request` (эхо-запрос) и `Echo Reply` (эхо-ответ). Следующие два правила дают возможность передавать пакеты ping по любому адресу:

```
ipchains -A output -i $EXTERNAL_INTERFACE -p icmp -s $IPADDR 11 -d $MY_ISP -j ACCEPT
ipchains -A input -i $EXTERNAL_INTERFACE -p icmp -s $ANYWHERE 11-d $IPADDR -j ACCEPT
```

Приведенные далее правила позволяют принимать пакеты ping только с определенных узлов, а конкретно — из сети вашего провайдера:

```
ipchains -A input -i $EXTERNAL_INTERFACE -p icmp -s $MY_ISP 8 -d $IPADDR -j ACCEPT
ipchains -A output -i $EXTERNAL_INTERFACE -p icmp -s $IPADDR 0 -d $MY_ISP -j ACCEPT
```

В данном примере набор внешних узлов, которым разрешена передача вашей системе пакетов ping, ограничен компьютерами провайдера. Сделано это для того, чтобы администратор провайдера мог в любой момент проверить, как происходит обмен данными с внешним интерфейсом вашего компьютера. Прием ping с остальных хостов запрещен.

Противодействие smurf-атакам

При организации атаки типа smurf пакеты ping, содержащие сообщения `Echo Request`, передаются в широковещательном режиме. Исходный IP-адрес в составе пакета подменяется IP-адресом "жертвы" — IP-адресом того узла, против которого направлена атака. В результате все узлы сети, получившие сообщения `Echo Request`, передают ответы по адресу "жертвы", загружая линии связи ICMP-пакетами. В результате, если у вас наружный канал не очень широкий, вы лишаетесь доступа в Интернет.

Правила, приведенные в листинге 32.2, предназначены для протоколирования попыток smurf-атаки. Поскольку прохождение широковещательных ICMP-пакетов явно не разрешено ни одним из правил, эти пакеты будут удалены по умолчанию. Обратите внимание, что в правилах указаны не только `Echo Request`, но и другие типы сообщений. Дело в том, что возможности для атаки не ограничиваются сетевыми пакетами ping.

Листинг 32.2

```
# Противодействие smurf-атаке
ipchains –A input –i $EXTERNAL_INTERFACE –p icmp –d $BROADCAST_DEST –j DENY –l
ipchains –A output –i $EXTERNAL_INTERFACE –p icmp –d $BROADCAST_DEST –j REJECT –l
# Маска сети
ipchains –A input –i $EXTERNAL_INTERFACE –p icmp –d $NETMASK –j DENY –l
ipchains –A output –i $EXTERNAL_INTERFACE –p icmp –d $NETMASK –j REJECT –l

# Адрес сети
ipchains –A input –i $EXTERNAL_INTERFACE  –p icmp –d $NETWORK –j DENY –l
ipchains –A output –i $EXTERNAL_INTERFACE –p icmp –d $NETWORK –j REJECT –l
```

Разрешение функционирования служб

Ранее мы определили правила, позволяющие отклонять сетевые пакеты с сомнительными адресами, а также разрешили локальному компьютеру работать через интерфейс обратной петли. В результате мы получили нормально функционирующий локальный компьютер с полностью отсутствующим доступом в Интернет. Наша дальнейшая задача — обеспечить нормальное функционирование локального компьютера (сети) в Интернете. Для того чтобы ваш компьютер мог принимать и отправлять почту, работать по FTP, HTTP и т. п., необходимо разрешить прохождение сетевых пакетов с определенными портами. На первый взгляд, задача объемная, впрочем, необходимо обеспечить прохождение пакетов всего от десятка служб, что не так уж и много.

Служба DNS

Служба DNS работает с портом 53 и протоколами UDP и TCP. Соединение может устанавливаться как между клиентом и сервером, так и между двумя серверами.

Для разрешения взаимодействия между клиентом и сервером нужно добавить такие правила:

```
ipchains –A input –i $EXTERNAL_INTERFACE –p udp –s $IPADDR $UNPRIVPORTS –d
$NAMESERVER 53 –j ACCEPT
ipchains –A output –i $EXTERNAL_INTERFACE –p udp –s $IPADDR $UNPRIVPORTS –d
$NAMESERVER 53 –j ACCEPT
```

В том случае, если ответ сервера не помещается в одной UDP-датаграмме, между клиентом и сервером устанавливается TCP-соединение. Обычно это происходит при передаче данных зоны между первичным и вторичным DNS-серверами. Для этого случая необходимо в цепочку правил добавить правила:

```
ipchains –A input –i $EXTERNAL_INTERFACE –p tcp –s $IPADDR $UNPRIVPORTS –d
$NAMESERVER 53 –j ACCEPT
ipchains –A output –i $EXTERNAL_INTERFACE –p tcp –s $IPADDR $UNPRIVPORTS –d
$NAMESERVER 53 –j ACCEPT
```

Если у вас есть локальный DNS-сервер, и вы предоставляете его услуги каким-либо клиентам (например, компьютерам вашей локальной сети), доступ к вашему локальному DNS-серверу желательно ограничить конкретным списком компьютеров. Для этого воспользуйтесь правилами, приведенными в листинге 32.3.

Листинг 32.3

```
# Разрешение обмена между клиентом и DNS-сервером
ipchains –A input –i $EXTERNAL_INTERFACE –p udp –s <clients.addr> $UNPRIVPORTS
–d $IPADDR 53 –j ACCEPT
ipchains –A output –i $EXTERNAL_INTERFACE –p udp –s <clients.addr>
$UNPRIVPORTS –d $IPADDR 53 –j ACCEPT


# Разрешение обмена между DNS-серверами
ipchains –A input –i $EXTERNAL_INTERFACE –p udp –s <clients.addr> 53 –d
$IPADDR –j ACCEPT
ipchains –A output –i $EXTERNAL_INTERFACE –p udp –s <clients.addr> 53 –d
$IPADDR –j ACCEPT
```

Следующие правила обеспечивают передачу по протоколу TCP:

```
ipchains –A input –i $EXTERNAL_INTERFACE –p tcp –s <dns.sec> $UNIPRIVPORTS –d
$IPADDR 53 –j ACCEPT
ipchains –A output –i $EXTERNAL_INTERFACE –p tcp ! –y –s $IPADDR 53 –d
<dns.sec> $UNIPRIVPORTS –j ACCEPT
```

E-mail

Протоколы для приема и пересылки электронной почты:

- SMTP порт 25 TCP;
- POP3 порт 110 TCP;
- IMAP порт 143 TCP.

При создании правил, разрешающих функционирование SMTP-протокола, будем считать, что наша почта отправляется через провайдера.

Для передачи почты по SMTP-протоколу на почтовый сервер провайдера необходимо добавить следующие правила:

```
ipchains –A input –i $EXTERNAL_INTERFACE –p tcp –s $IPADDR $UNIPRIVPORTS –d
$SMTP_GATEWAY 25 –j ACCEPT
ipchains –A output –i $EXTERNAL_INTERFACE –p tcp ! –y –s $SMTP_GATEWAY 25 –d
$IPADDR $UNIPRIVPORTS –j ACCEPT
```

В том случае, если у вас в локальной сети присутствует свой собственный SMTP-сервер, правила фильтрации принимают такой вид:

```
ipchains –A input –i $EXTERNAL_INTERFACE –p tcp –s $IPADDR $UNIPRIVPORTS –d
$ANYWHERE 25 –j ACCEPT
ipchains –A output –i $EXTERNAL_INTERFACE –p tcp ! –y –s $ANYWHERE 25 –d
$IPADDR $UNIPRIVPORTS –j ACCEPT
```

Для получения электронной почты требуется протокол POP3 или IMAP. Для нормального функционирования POP3-протокола необходимо для нашего брандмауэра добавить правила:

```
ipchains –A input –i $EXTERNAL_INTERFACE –p tcp –s $IPADDR $UNIPRIVPORTS –d
$POP_SERVER 110 –j ACCEPT
ipchains –A output –i $EXTERNAL_INTERFACE –p tcp ! –y –s $POP_SERVER 110 –d
$IPADDR $UNIPRIVPORTS –j ACCEPT
```

Правила для предоставления некоторым внешним хостам доступа к вашему POP3-серверу:

```
ipchains –A input –i $EXTERNAL_INTERFACE –p tcp –s <pop.clients> $UNIPRIVPORTS
–d $IPADDR 110 –j ACCEPT
ipchains –A output –i $EXTERNAL_INTERFACE –p tcp ! –y –s $IPADDR 110 –d
<pop.clients> $UNIPRIVPORTS –j ACCEPT
```

Если у вас IMAP-протокол, добавьте следующие правила:

```
ipchains –A input –i $EXTERNAL_INTERFACE –p tcp –s $IPADDR $UNIPRIVPORTS –d
$IMAP_SERVER 143 –j ACCEPT
ipchains –A output –i $EXTERNAL_INTERFACE –p tcp ! –y –s $IMAP_SERVER 143 –d
$IPADDR $UNIPRIVPORTS –j ACCEPT
```

Правила, позволяющие предоставить некоторым внешним хостам доступ к вашему IMAP-серверу:

```
ipchains –A input –i $EXTERNAL_INTERFACE –p tcp –s <pop.clients> $UNIPRIVPORTS
–d $IPADDR 143 –j ACCEPT
ipchains –A output –i $EXTERNAL_INTERFACE –p tcp ! –y –s $IPADDR 143 –d
<pop.clients> $UNIPRIVPORTS –j ACCEPT
```

NNTP

Сервер новостей задействует порт 119 и протокол TCP. Для обеспечения нормального функционирования сервера новостей необходимо добавить три набора правил.

Если вы используете сервер новостей вашего провайдера, то для получения и отправки статей в группы новостей следует добавить правила:

```
ipchains –A input –i $EXTERNAL_INTERFACE –p tcp –s $IPADDR $UNIPRIVPORTS –d
$NEWS_SERVER 119 –j ACCEPT
ipchains –A output –i $EXTERNAL_INTERFACE –p tcp ! –y –s $NEWS_SERVER 119 –d
$IPADDR $UNIPRIVPORTS –j ACCEPT
```

Если у вас в локальной сети есть свой собственный сервер новостей, и вы хотите разрешить извне доступ определенным хостам, воспользуйтесь правилами:

```
ipchains –A output –i $EXTERNAL_INTERFACE –p tcp –s <ip.clients> $UNIPRIVPORTS
–d $NEWS_SERVER 119 –j ACCEPT
ipchains –A input –i $EXTERNAL_INTERFACE –p tcp ! –y –s $NEWS_SERVER 119 –d
<ip.clients> $UNIPRIVPORTS –j ACCEPT
```

А поскольку вашему локальному серверу новостей необходимо получать и передавать статьи от сервера новостей провайдера, назначьте такие правила:

```
ipchains –A output –i $EXTERNAL_INTERFACE –p tcp –s $IPADDR $UNIPRIVPORTS –d
$NEWS_SERVER 119 –j ACCEPT
ipchains –A input –i $EXTERNAL_INTERFACE –p tcp ! –y –s $NEWS_SERVER 119 –d
$IPADDR $UNIPRIVPORTS –j ACCEPT
```

Telnet

Telnet работает с портом 23 по протоколу TCP. Применение протокола удаленного доступа Telnet было очень популярно еще три-четыре года назад, однако из-за того, что этот протокол абсолютно не защищен, а также вследствие появления альтернативы в виде протокола SSH, в настоящее время категорически не рекомендуется разрешать доступ извне через Telnet.

SSH

Протокол SSH использует порт 22 и TCP. Защищенная замена Telnet и r-командам. При функционировании занимает привилегированные порты 513–1023.

В листинге 32.4 приведены правила, позволяющие применять протокол SSH для доступа из локальной сети к Интернету SSH-серверам.

Листинг 32.4

```
ipchains –A output –i $EXTERNAL_INTERFACE –p tcp –s $IPADDR $UNIPRIVPORTS –d
$ANYWHERE 22 –j ACCEPT
ipchains –A input –i $EXTERNAL_INTERFACE –p tcp ! –y –s $ANYWHERE 22 –d
$IPADDR $UNIPRIVPORTS –j ACCEPT
ipchains –A input –i $EXTERNAL_INTERFACE –p tcp –s $ANYWHERE $SSH_PORTS –d
$IPADDR 22 –j ACCEPT
ipchains –A output –i $EXTERNAL_INTERFACE –p tcp ! –y –s $IPADDR 22 –d
$ANYWHERE $SSH_PORTS –j ACCEPT
```

Правила, приведенные в листинге 32.5, разрешают доступ удаленным клиентам к вашим локальным SSH-серверам.

Листинг 32.5

```
ipchains –A input –i $EXTERNAL_INTERFACE –p tcp –s $ANYWERE $UNIPRIVPORTS –d
$IPADDR 22 –j ACCEPT
ipchains –A output –i $EXTERNAL_INTERFACE –p tcp ! –y –s $IPADDR 22 –d
$ANYWHERE $UNIPRIVPORTS –j ACCEPT
```

```
ipchains –A -i $EXTERNAL_INTERFACE –p tcp -s $IPADDR $SSH_PORTS -d $ANYWERE 22
–j ACCEPT
ipchains –A input -i $EXTERNAL_INTERFACE –p tcp ! –y -s $ANYWHERE 22 -d
$IPADDR $SSH_PORTS –j ACCEPT
```

FTP

Использует несколько портов (TCP 21, 20). Протокол предусматривает два режима передачи — активный и пассивный канал передачи, что несколько осложняет конфигурацию брандмауэра.

Следующие правила разрешают доступ к удаленным FTP-серверам:

```
ipchains –A output -i $EXTERNAL_INTERFACE –p tcp -s $IPADDR $UNPRIVPORTS -d
$ANYWERE 21 –j ACCEPT
ipchains –A input -i $EXTERNAL_INTERFACE –p tcp ! –y -s $ANYWHERE 21 -d
$IPADDR $UNPRIVPORTS –j ACCEPT
```

В листинге 32.6 приведены правила, разрешающие устанавливать соединения в режиме активного канала передачи данных.

Листинг 32.6

```
ipchains –A input -i $EXTERNAL_INTERFACE –p tcp ! –y -s $ANYWHERE 20 -d
$IPADDR $UNPRIVPORTS –j ACCEPT
ipchains –A output -i $EXTERNAL_INTERFACE –p tcp -s $IPADDR $UNPRIVPORTS -d
$ANYWERE 20 –j ACCEPT
ipchains –A output -i $EXTERNAL_INTERFACE –p tcp -s $IPADDR $UNPRIVPORTS -d
$ANYWERE $UNPRIVPORTS –j ACCEPT
ipchains –A input -i $EXTERNAL_INTERFACE –p tcp ! –y -s $ANYWHERE $UNPRIVPORTS
-d $IPADDR $UNPRIVPORTS –j ACCEPT
```

Для того чтобы к вашему локальному FTP-серверу был разрешен доступ извне, необходимо ввести правила, приведенные в листинге 32.7.

Листинг 32.7

```
ipchains –A input -i $EXTERNAL_INTERFACE –p tcp -s $ ANYWERE $UNPRIVPORTS -d
$IPADDR 21 –j ACCEPT
ipchains –A output -i $EXTERNAL_INTERFACE –p tcp ! –y -s $IPADDR 21 -d
$ANYWERE $UNPRIVPORTS –j ACCEPT
ipchains –A output -i $EXTERNAL_INTERFACE –p tcp -s $IPADDR 20 -d $ANYWERE
$UNPRIVPORTS –j ACCEPT
ipchains –A input -i $EXTERNAL_INTERFACE –p tcp ! –y -s $ANYWHERE $UNPRIVPORTS
-d $IPADDR 20 –j ACCEPT
ipchains –A input -i $EXTERNAL_INTERFACE –p tcp -s $ANYWHERE $UNPRIVPORTS -d
$IPADDR $UNPRIVPORTS –j ACCEPT
ipchains –A output -i $EXTERNAL_INTERFACE –p tcp ! –y -s $IPADDR $UNPRIVPORTS
-d $ANYWHERE $UNPRIVPORTS –j ACCEPT
```

HTTP

Протокол HTTP использует порт 80 TCP. Для того чтобы локальные клиенты могли получить доступ к Web-серверам Интернета, необходимо ввести следующие правила:

```
ipchains –A output -i $EXTERNAL_INTERFACE –p tcp -s $IPADDR $UNPRIVPORT -d
$ANYWHERE 80 –j ACCEPT
ipchains –A input -i $EXTERNAL_INTERFACE –p tcp !-y -s $ANYWHERE 80 -d $IPADDR
$UNIPRIVPORTS –j ACCEPT
```

Если у вас в локальной сети есть свой собственный Web-сервер, и вы хотите разрешить извне доступ, воспользуйтесь такими правилами:

```
ipchains –A input -i $EXTERNAL_INTERFACE –p tcp -s $ANYWHERE $UNIPRIVPORTS -d
$IPADDR 80 –j ACCEPT
ipchains –A output -i $EXTERNAL_INTERFACE –p tcp ! –y -s $IPADDR 80 -d
$ANYWHERE $UNIPRIVPORTS –j ACCEPT
```

Помимо перечисленных сервисов у вас могут встретиться и другие, однако, зная протокол, используемые порты и опираясь на ранее приведенные примеры, не составит труда добавить соответствующие правила для нормального функционирования ваших сервисов.

Запрет доступа с "неблагонадежных" узлов

Если вы обнаружите попытки сканирования портов или другие сомнительные действия, которые периодически предпринимаются с одного и того же хоста, желательно вовсе запретить обращение к системе с этого адреса.

Пример запрещающего правила:

```
ipchains –I input  -i $EXTERNAL_INTERFACE  -s <адрес/маска> –j DENY
```

Согласно этому правилу удаляется любой пакет, независимо от протокола и номера исходного порта или порта назначения.

Поддержка обмена в локальной сети

Для поддержки локальной сети, стоящей за брандмауэром, следует добавить некоторые правила. Они необходимы для того, чтобы разрешить доступ к внутреннему сетевому интерфейсу брандмауэра и направить трафик в глобальную сеть. Как только на компьютере-брандмауэре будет реализована поддержка двух или более интерфейсов, он превратится в бастион.

Разрешение доступа к внутреннему сетевому интерфейсу брандмауэра

При работе с небольшими сетями вряд ли нужно ограничивать доступ к брандмауэру из локальной сети. Следующие правила разрешают все виды взаимодействия между брандмауэром и локальной сетью:

```
ipchains –A input –i $INTERNAL_INTERFACE -s LAN –j ACCEPT
ipchains –A output –i $Internal_INTERFACE -s LAN_1 –j ACCEPT
```

Обратите внимание, что данные правила разрешают обмен лишь с брандмауэром. Доступ к Интернету отсутствует, поскольку по умолчанию компьютер, выполняющий роль брандмауэра, не проводит динамической маршрутизации пакетов и не поддерживает статических маршрутов. Чтобы обеспечить маршрутизацию пакетов через брандмауэр, нужно задать дополнительные правила.

Выбор конфигурации для пользующейся доверием локальной сети

Пакеты, передаваемые компьютерами локальной сети, можно условно разбить на две категории:

- данные, которыми локальные узлы обмениваются с брандмауэром;
- данные, направляемые через внешний интерфейс брандмауэра в Интернет.

При работе небольшой сети вряд ли может возникнуть потребность в фильтрации пакетов, проходящих через внутренний интерфейс брандмауэра, однако некоторая обработка все-таки необходима. Речь идет о маскировке.

Если компьютер, выполняющий роль брандмауэра, имеет реальный IP-адрес, а всем остальным машинам, подключенным к локальной сети, присвоены адреса, предназначенные для внутреннего использования, то для обеспечения доступа локальных машин в Интернет брандмауэр должен взять на себя функции proxy-сервера.

По сути, компьютер, выполняющий маскировку пакетов, представляет собой низкоуровневый proxy-сервер, который обслуживает клиентов, устанавливая соединения с удаленными узлами от своего имени. Поскольку исходный адрес в пакете, передаваемом в Интернет, заменяется адресом компьютера, выполняющего маскировку, то с точки зрения удаленного узла обмен данными с ним проводит proxy-сервер. В пакетах, передаваемых удаленным узлом, адрес назначения заменяется на адрес локального компьютера.

Организация доступа из локальной сети к брандмауэру бастионного типа

Если вы занимаетесь администрированием небольшой сети, то, скорее всего, не захотите ограничивать доступ локальных компьютеров к брандмауэру бастионного типа. Следующее правило организует неограниченный доступ к внутреннему сетевому интерфейсу брандмауэра:

```
ipchains -A input -i $INTERNAL_INTERFACE -S LAN_1 -j ACCEPT ipchains -A output
-i $INTERNAL_INTERFACE -d LAN_1 -j ACCEPT
```

Перенаправление трафика

Если несколько локальных сетей должны обмениваться информацией, вам необходимо разрешить передачу пакетов между соответствующими интерфейсами. Конечно, делать это нужно лишь в том случае, если маршрутизация не выполняется какими-либо другими средствами.

Чтобы брандмауэр можно было использовать в качестве маршрутизатора, объединяющего две локальные сети, необходимо добавить правила:

```
#Приведенные правила разрешают доступ к брандмауэру
ipchains -A input -i $LAN_INTERFACE_1 -s LAN_1 -j ACCEPT ipchains -A output -i
$LAN_INTERFACE_1 -d LAN_1 -j ACCEPT
ipchains -A input -i $LAN_INTERFACE_2 -s LAN_2 -j ACCEPT ipchains -A output -i
$LAN_INTERFACE_2 -d LAN_2 -j ACCEPT
```

Правила, обеспечивающие передачу трафика между локальными сетями в двух направлениях без маскировки:

```
ipchains -A forward -i $LAN_INTERFACE_2 -s LAN_1 -d LAN_2 -j ACCEPT
ipchains -A forward -i $LAN_INTERFACE_1 -s LAN_2 -d LAN_1 -j ACCEPT
```

Разрешение доступа в Интернет из локальной сети: IP-перенаправление и маскировка

На данном этапе на обмен данными между машинами локальной сети и внутренним интерфейсом брандмауэра не накладывается никаких ограничений. Однако локальным компьютерам Интернет недоступен. Для обеспечения такого доступа необходимо реализовать перенаправление и маскировку пакетов.

Механизм перенаправления реализуется на уровне ядра системы и позволяет компьютеру под управлением Linux выступать в роли маршрутизатора, перенаправляя трафик из одной сети в другую. Однако даже если IP-перенаправление осуществляется путем выбора конфигурации сети, пакеты не будут передаваться между интерфейсами до тех пор, пока не созданы правила, разрешающие такую передачу.

Перенаправления пакетов, адресованных различным узлам глобальной сети, не всегда достаточно для нормального взаимодействия локальных компьютеров с Интернетом. Если компьютерам локальной сети присвоены IP-адреса классов A, B и C, предназначенные для внутреннего использования, необходимо выполнить их маскировку, т. е. заменить исходящий адрес локального узла в пакете IP-адресом внешнего интерфейса брандмауэра. Эта возможность также реализована на уровне ядра операционной системы. Но даже если компьютеры локальной сети имеют обычные IP-адреса, допустимые в Интернете, маскировка остается одним из самых эффективных средств защиты внутренней сети.

Несмотря на то, что перенаправление и маскировка — это совершенно различные механизмы, на уровне программы ipchains они представляют одну процедуру. Пакеты, поступившие на внутренний интерфейс брандмауэра, передаются на его внешний интерфейс. Перед помещением пакета в очередь внешнего интерфейса средства маскировки заменяют адрес источника IP-адресом внешнего интерфейса брандмауэра. Наличие средств перенаправления и маскировки превращает брандмауэр в proxy-сервер с возможностями фильтрации.

Приведенное далее правило позволяет перенаправить трафик с внутреннего интерфейса на внешний, попутно выполняя маскировку пакетов:

```
ipchains –A forward –I $EXTERNAL_INTERFACE –s LAN_1 –j MASQ
```

Действия ACCEPT и DENY, указанные в цепочке output внешнего интерфейса, производятся после того, как перенаправление будет выполнено. Таким образом, несмотря на то, что передача от внутреннего к внешнему интерфейсу разрешена для всех пакетов, в Интернет попадут лишь те из них, для которых существуют разрешающие правила, связанные с внешним интерфейсом.

Правила маскировки позволяют задавать адреса источника и назначения, а также номера портов.

При перенаправлении трафик передается между сетевыми интерфейсами без изменений. Если компьютеры внутренней сети имеют IP-адреса, допустимые в Интернете, и брандмауэр перенаправляет трафик, то с точки зрения стороннего наблюдателя между локальной машиной и хостом Интернета устанавливается непосредственное соединение. При обращении удаленного компьютера к локальному узлу пакеты перенаправляются в локальную сеть.

При наличии маскировки передача трафика перестает быть симметричной. В этом случае разрешены лишь обращения из локальной сети к внешним серверам. При передаче пакета в Интернет исходный адрес, принадлежащий локальному компьютеру, заменяется IP-адресом внешнего интерфейса брандмауэра. При получении ответа от сервера осуществляется обратное преобразование пакета: IP-адрес брандмауэра заменяется адресом локального компьютера, которому адресован пакет.

Как перенаправление, так и маскировка пакетов выполняются на уровне ядра операционной системы, поэтому оно должно быть скомпилировано с поддержкой маскировки и перенаправления пакетов.

Разрешить маскировку можно с помощью программы ipchains. Для маскировки всего трафика, направленного из локальной сети к удаленным узлам, необходимо задать следующее правило:

```
ipchains -A forward -i $EXTERNAL_INTERFACE \ -s LAN_1 -j MASQ
```

Независимо от того, выделены ли для компьютеров локальной сети допустимые IP-адреса или им присвоены адреса, предназначенные для внутреннего использования, при настройке брандмауэра рекомендуется отказаться от прямого перенаправления пакетов и задействовать маскировку. Маскировка локальных сетей — мощное средство защиты. При маскировке хосты Интернета не могут обращаться к компьютерам вашей локальной сети. Более того, локальные машины не видны извне. С точки зрения Интернета вся ваша локальная сеть состоит из одного хоста — компьютера, на котором реализован брандмауэр.

Дополнительной мерой защиты могут стать proxy-фильтры прикладного уровня, такие как SOCKS. И в этом случае при обмене с удаленным узлом создается впечатление, что запросы генерируются брандмауэром. Преимущество фильтров прикладного уровня также состоит в том, что с их помощью можно организовать специальную обработку трафика, учитывающую специфику обмена с конкретными службами.

Организация демилитаризованной зоны

Конфигурация брандмауэра, описанная в начале главы, вполне подходит для защиты одного компьютера от нежелательных воздействий извне. Брандмауэр с двумя сетевыми интерфейсами способен защищать локальную сеть. Брандмауэр бас-

тионного типа защищает локальную сеть до тех пор, пока система, на которой установлен брандмауэр, не будет взломана. Даже если в процессе фильтрации участвует не только внешний, но и внутренний интерфейс, это не спасет систему. Если злоумышленнику удастся взломать компьютер, выполняющий роль брандмауэра, то ваша локальная сеть остается беззащитной перед лицом взломщика. Поэтому брандмауэр бастионного типа представляет собой единственную линию обороны. Такой тип защиты распространен в небольших организациях.

В средних и крупных организациях обычно предусмотрены proxy-серверы либо система из двух брандмауэров, между которыми располагается *демилитаризованная зона*, или граничная сеть. Внешний интерфейс первого брандмауэра осуществляет соединение с Интернетом, а внутренний принадлежит демилитаризованной зоне, как правило, использующей свою локальную сеть. Второй брандмауэр, который обычно называется *заглушкой* (choke), также имеет два интерфейса. Внешний интерфейс подключен к демилитаризованной зоне, а внутренний — к внутренней сети предприятия. Обычно в демилитаризованной зоне размещаются серверы, которые должны быть доступны из Интернета. Описанная архитектура требует намного больше компьютеров и обслуживающего персонала, чем брандмауэр бастионного типа.

Защита подсетей с помощью брандмауэров

Демилитаризованную зону обычно организуют одним из двух способов. Первый способ предполагает применение брандмауэра бастионного типа с тремя сетевыми интерфейсами. Один сетевой интерфейс подключается к Интернету, а два остальных — к двум изолированным локальным сетям. Одна из сетей играет роль демилитаризованной зоны, в ней размещают общедоступные серверы. Во второй сети располагают службы, предназначенные для внутреннего использования, и компьютеры пользователей.

Другой способ состоит в создании второго брандмауэра, называемого заглушкой (choke). Компьютер, на котором реализован брандмауэр-заглушка, выполняет функцию шлюза между демилитаризованной зоной и локальной сетью. Внутренний сетевой интерфейс брандмауэра-заглушки подключен к локальной сети, а внешний — к демилитаризованной зоне. Данные, передаваемые компьютерами локальной сети, маскируются. Таким образом, с точки зрения брандмауэра-бастиона и машин, принадлежащих демилитаризованной зоне, вся внутренняя сеть представлена одним брандмауэром-заглушкой.

Бастион маскирует трафик внутренней сети, поэтому, на первый взгляд, брандмауэр-заглушка не должен выполнять маскировку. Однако если вся внутренняя сеть имеет один адрес, принадлежащий брандмауэру-заглушке, набор правил бастиона упрощается.

Подобная структура реализует две линии обороны локальной сети. Локальная сеть расположена за заглушкой и полностью изолирована от бастиона и, тем более, от Интернета. При реализации описанной системы не обязательно задавать полный набор правил для внутреннего интерфейса бастиона, достаточно, если правила, осуществляющие фильтрацию пакетов, будут связаны с внешним интерфейсом заглушки.

Таким образом, система защиты внутренней сети содержит как минимум четыре набора правил — по одному для внутреннего и внешнего интерфейса каждого из брандмауэров. Правила для внешнего интерфейса бастиона практически совпадают с правилами брандмауэра, описанного ранее.

Реально описанная здесь система отличается от ранее рассмотренной наличием демилитаризованной зоны, а также новыми правилами, заданными для внутреннего интерфейса бастиона и для внешнего интерфейса заглушки. Указанные два набора правил, по сути, представляют собой зеркальное отражение друг друга.

Отладка брандмауэра

Предположим, брандмауэр установлен, настроен и активизирован, но действует не так, как хотелось. Даже если брандмауэр работает, рекомендуется сразу после установки и настройки проверить правильность функционирования брандмауэра.

Общие рекомендации по отладке брандмауэра

Приведем рекомендации, позволяющие облегчить отладку брандмауэра.

- Перед запуском сценария убедитесь, что в первой строке находится команда удаления существующих правил, а последующая команда устанавливает политику по умолчанию.
- Проводите отладку с текстовой консоли. Не отлаживайте брандмауэр с удаленной машины. При обрыве связи или неправильном конфигурировании вы рискуете остаться без доступа в Интернет.
- По возможности добавляйте правила по одному. В этом случае гораздо проще выявить причину неисправности. Сразу после добавления правил рекомендуется проверить их работоспособность.
- Обработка сетевого пакета определяется первым правилом, которому удовлетворяет этот сетевой пакет, поэтому порядок следования правил имеет большое значение.
- Помните, что существуют как минимум две не зависящие друг от друга цепочки: input и output. Если правила, содержащиеся в одной цепочке, обрабатывают пакет корректно, причина неисправности, очевидно, находится в другой цепочке.
- Если сценарий "зависает", возможно, что правило, в котором содержится доменное имя узла, вступает в действие раньше, чем правило, разрешающее доступ к DNS. Если какое-либо правило предшествует правилам, определяющим взаимодействие с DNS, в нем должны быть указаны IP-адреса. Доменные имена в таких правилах недопустимы, поскольку у вас еще нет доступа к серверу DNS.
- Проверяйте синтаксис команд программы ipchains. При составлении правил легко перепутать адрес или порт источника с адресом или портом назначения либо неверно задать регистр опции.
- При наличии синтаксической ошибки выполнение сценария брандмауэра завершается, и последующие правила не устанавливаются. Чтобы определить неверно составленное правило, запускайте сценарий с опциями `-x` или `-v`. Если указана опция `-v`, строки сценария выводятся в тот момент, когда они читаются

интерпретатором команд. Опция `-x` задает вывод строк по мере выполнения команд оболочкой.

- Если какой-либо из серверов не работает, включите протоколирование удаляемых пакетов, указав опцию `-l` программы ipchains. Проанализируйте записи в файле /var/log/messages.
- Если вы обмениваетесь данными с Интернетом, работая на компьютере-брандмауэре, но не можете сделать этого с узла локальной сети, проверьте установки в /etc/sysconfig/network, связанные с перенаправлением пакетов.
- Если сервер доступен в пределах локальной сети, но попытка обратиться к нему извне оканчивается неудачей, включите протоколирование пакетов, проходящих через внутренний интерфейс. Постарайтесь выполнить всю проверку как можно быстрее, в противном случае в файле /var/log/messages появятся сотни записей.
- При неработоспособности одной из служб временно включите в начало сценария брандмауэра правила, разрешающие прохождение пакетов в обоих направлениях, и задайте протоколирование, указав опцию `-l`. Проверьте, доступен ли сервер. Если это так, просмотрите записи в файле /var/log/messages и определите, какие порты используются при его работе.

Отображение списка правил брандмауэра

Чтобы убедиться, что правила брандмауэра инсталлированы именно так, как вы это планировали при составлении сценария, можно вывести содержимое цепочек. Сделать это позволяет опция `-L` программы ipchains. Если опция `-L` задана, ipchains выводит содержащиеся в соответствующей таблице ядра правила в той последовательности, в которой они применяются при обработке пакета. Содержимое цепочек выводят следующие команды:

```
ipchains -L input
ipchains -L output
ipchains -L forward
```

Различные опции программы ipchains позволяют выдавать содержимое одной и той же цепочки с различной степенью детализации. Форматы вывода для цепочек input, output и forward совпадают.

Утилиты

В состав пакета ipchains входит утилита ipchains-save, с помощью которой вы можете получить текущую конфигурацию брандмауэра на стандартный вывод, а затем перенаправить его в файл. Также есть утилита-близнец ipchains-restore, которая может получать информацию из стандартного ввода.

Iptables

Iptables — это логическое развитие ipchains. Взяли все лучшее, модифицировали, увеличили гибкость, надежность и производительность. В современных дистрибутивах iptables является неотъемлемой частью системы и ядро скомпилировано

с учетом требований iptables. В том случае, если вы предпочитаете лично пересобрать ядро операционной системы, при его конфигурации необходимо сделать такие настройки:

- CONFIG_PACKET — опция необходима для приложений, работающих непосредственно с сетевыми устройствами, например: tcpdump или snort;
- CONFIG_NETFILTER — опция необходима, если вы собираетесь использовать компьютер в качестве сетевого экрана или шлюза;
- CONFIG_IP_NF_CONNTRACK — трассировка соединений (среди всего прочего, задействована при трансляции сетевых адресов и маскарадинге (masquerading));
- CONFIG_IP_NF_FTP — трассировка FTP-соединений;
- CONFIG_IP_NF_IPTABLES — опция необходима для выполнения операций фильтрации, преобразования сетевых адресов (NAT) и маскарадинга;
- CONFIG_IP_NF_MATCH_LIMIT — модуль предоставляет возможность ограничения количества проверок для некоторого правила. Например, `-m limit --limit 3/minute` указывает, что заданное правило может пропустить не более трех пакетов в минуту. Таким образом, данный модуль может применяться для защиты от нападений типа "Отказ в обслуживании";
- CONFIG_IP_NF_MATCH_MAC — модуль позволяет строить правила, основанные на MAC-адресации;
- CONFIG_IP_NF_MATCH_MARK — функция маркировки пакетов MARK, позволяющая пометить требуемые пакеты, а затем, в других таблицах, в зависимости от значения метки, принимать решение о маршрутизации помеченного пакета;
- CONFIG_IP_NF_MATCH_MULTIPORT — модуль позволяет строить правила с проверкой на принадлежность пакета к диапазону номеров портов источника/приемника;
- CONFIG_IP_NF_MATCH_TOS — модуль позволяет строить правила, отталкиваясь от состояния поля TOS в пакете. Поле TOS устанавливается для Type Of Service;
- CONFIG_IP_NF_MATCH_TCPMSS — опция добавляет возможность проверки поля MSS в TCP-пакетах;
- CONFIG_IP_NF_MATCH_STATE — это самое серьезное усовершенствование по сравнению с ipchains. Данный модуль предоставляет возможность управления TCP-пакетами, основываясь на их состоянии (state);
- CONFIG_IP_NF_MATCH_UNCLEAN — модуль реализует возможность дополнительной проверки IP-, TCP-, UDP- и ICMP-пакетов на предмет наличия в них несоответствий и ошибок;
- CONFIG_IP_NF_MATCH_OWNER — проверка "владельца" соединения (socket). Для примера, мы можем позволить только пользователю root выходить в Интернет;
- CONFIG_IP_NF_FILTER — реализация таблицы filter, в которой в основном и осуществляется фильтрация. В данной таблице находятся цепочки input, forward и output;
- CONFIG_IP_NF_TARGET_REJECT — добавляется действие REJECT, которое передает ICMP-сообщение об ошибке в ответ на входящий пакет, который отвергается заданным правилом;
- CONFIG_IP_NF_TARGET_MIRROR — возможность отправки полученного пакета обратно;

- CONFIG_IP_NF_NAT — трансляция сетевых адресов. С помощью этой опции вы сможете дать выход в Интернет всем компьютерам вашей локальной сети, имея лишь один уникальный IP-адрес;
- CONFIG_IP_NF_TARGET_MASQUERADE — маскарадинг. В отличие от NAT, маскарадинг целесообразен в тех случаях, когда заранее неизвестен наш IP-адрес в Интернете. Маскарадинг сильнее нагружает компьютер, по сравнению с NAT, однако он работает в ситуациях, когда невозможно заранее указать собственный внешний IP-адрес;
- CONFIG_IP_NF_TARGET_REDIRECT — перенаправление. Вместо того чтобы просто пропустить пакет дальше, это действие перенаправляет пакет на другой порт сетевого экрана;
- CONFIG_IP_NF_TARGET_LOG — фиксирует отдельные пакеты в системном журнале (syslog);
- CONFIG_IP_NF_TARGET_TCPMSS — эта опция позволяет преодолеть ограничения, накладываемые некоторыми провайдерами, которые блокируют ICMP-пакеты Fragmentation Needed;
- CONFIG_IP_NF_COMPAT_IPCHAINS — добавляет совместимость с ipchains;
- CONFIG_IP_NF_COMPAT_IPFWADM — добавляет совместимость с ipfwadm.

Порядок движения транзитных пакетов

В табл. 32.4 приведен порядок движения транзитных пакетов, соответствующие таблицы фильтрации и описание.

Таблица 32.4. *Прохождение транзитных пакетов*

Таблица	Цепочка	Примечание
mangle	prerouting	Входящий сетевой интерфейс Цепочка используется для внесения изменений в заголовок пакета
nat	prerouting	Цепочка служит для трансляции сетевых адресов. Любого рода фильтрация в этой цепочке может производиться только в исключительных случаях. В этой точке решается, куда пойдет пакет: локальному приложению или на другой хост
mangle	forward	Цепочка forward таблицы mangle должна использоваться только в исключительных случаях, когда необходимо внести некоторые изменения в заголовок пакета между двумя точками принятия решения о маршрутизации
filter	forward	В цепочку forward попадают только те пакеты, которые идут на другой хост. Вся фильтрация транзитного трафика должна выполняться здесь. Через эту цепочку проходит трафик в обоих направлениях, обязательно учитывайте это обстоятельство при написании правил фильтрации

Таблица 32.4 *(окончание)*

Таблица	Цепочка	Примечание
mangle	postrouting	Эта цепочка предназначена для внесения изменений в заголовок пакета после того, как принято последнее решение о маршрутизации
nat	postrouting	Эта цепочка предназначена в первую очередь для NAT и маскарадинга. Исходящий сетевой интерфейс

Порядок движения пакетов для локальной программы

Существует определенный порядок движения пакетов для локальной программы (табл. 32.5).

Таблица 32.5. *Движение пакетов для локальных программ*

Таблица	Цепочка	Примечание
		Входной сетевой интерфейс
mangle	prerouting	Используется для внесения изменений в заголовок пакета
nat	prerouting	Преобразование адресов. Фильтрация пакетов здесь допускается только в исключительных случаях
		Принятие решения о маршрутизации
mangle	input	Здесь вносятся изменения в заголовок пакета перед тем, как он будет передан локальной программе
filter	input	Фильтруется входящий трафик
		Локальная программа

Порядок движения пакетов от локальной программы

Существует определенный порядок движения пакетов от локальной программы (табл. 32.6).

Таблица 32.6. *Движение пакетов, созданных локальными программами*

Таблица	Цепочка	Примечание
		Локальный процесс
		Принятие решения о маршрутизации
mangle	output	Используется для внесения изменений в заголовок пакета
nat	output	Предназначена для трансляции сетевых адресов (NAT) в пакетах, исходящих от локальных процессов брандмауэра

Таблица 32.6 *(окончание)*

Таблица	Цепочка	Примечание
filter	output	Фильтруется исходящий трафик
mangle	postrouting	Используется для правил, которые должны вносить изменения в заголовок пакета перед тем, как он покинет брандмауэр, но уже после принятия решения о маршрутизации. В эту цепочку попадают все пакеты, как транзитные, так и созданные локальными процессами брандмауэра
nat	postrouting	Нежелательно фильтровать пакеты во избежание побочных эффектов. Однако и здесь можно останавливать пакеты, применяя политику по умолчанию DROP
		Сетевой интерфейс

Таблица *mangle*

В таблице mangle, предназначенной для внесения изменений в заголовки пакетов, допускается выполнять только три действия:

- TOS;
- TTL;
- MARK.

Действие TOS устанавливает биты поля `Type of Service` в пакете. Это поле используется для назначения сетевой политики обслуживания пакета.

Действие TTL задает значение поля `TTL` (Time To Live) пакета.

Действие MARK устанавливает специальную метку на пакет, которая затем может быть проверена другими правилами в iptables или другими программами. С помощью "меток" можно управлять маршрутизацией пакетов, ограничивать трафик и т. п.

Таблица *nat*

Таблица служит для преобразования сетевых адресов (Network Address Translation, NAT) и предусматривает следующие действия:

- DNAT (Destination Network Address Translation) — преобразует адреса назначения в заголовках пакетов (перенаправляет пакеты);
- SNAT (Source Network Address Translation) — изменяет исходные адреса пакетов. С помощью этого действия можно скрыть структуру локальной сети, разделить единственный внешний IP-адрес между компьютерами локальной сети для выхода в Интернет;
- MASQUERADE (маскировка) — применяется в тех же целях, что и SNAT, но в отличие от последнего, MASQUERADE сильнее нагружает систему. Происходит это потому, что каждый раз, когда требуется выполнение этого действия, производится запрос IP-адреса для указанного в действии сетевого интерфейса, в то время как для SNAT IP-адрес указывается непосредственно. Однако благодаря такому отличию MASQUERADE может работать с динамическим IP-адресом.

Таблица *filter*

В этой таблице должны содержаться наборы правил для фильтрации пакетов. Пакеты могут пропускаться далее либо отвергаться (действия ACCEPT и DROP соответственно) в зависимости от их содержимого.

Построение правил для iptables

Теперь рассмотрим порядок построения правил для iptables. В целом, он мало отличается от построения правил для ipchains.

Каждое правило — это строка, содержащая в себе правила, с помощью которых определяется, подпадает ли пакет под заданное правило, и действие, которое необходимо выполнить при выполнении критерия. В общем виде правила записываются так:

```
iptables [-t table] command [match] [target/jump]
```

Если в правило не включается спецификатор `[-t table]`, то по умолчанию предполагается использование таблицы filter, если же необходима другая таблица, то это требуется указать явно.

Непосредственно за именем таблицы должна стоять команда, определяющая действие iptables. Если спецификатора таблицы нет, то команда всегда должна стоять первой.

Раздел `[match]` задает критерии проверки, по которым определяется, подпадает ли пакет под действие этого правила или нет. Здесь можно задать самые разные критерии: IP-адрес источника пакета или сети, IP-адрес места назначения, порт, протокол, сетевой интерфейс и т. п.

И, наконец, `[target]` указывает действие при условии выполнения критериев в правиле.

Команды ipchains

В табл. 32.7 приведен список команд и правила их использования. Обычно предполагается одно из двух действий: добавление нового правила в цепочку или удаление существующего правила из таблицы.

Таблица 32.7. *Команды iptables*

Команда	Пример использования	Описание
`-A, --append`	`iptables -A INPUT ...`	Добавляет новое правило в конец цепочки
`-D, --delete`	`iptables -D INPUT --dport 80 -j DROP`, `iptables -D INPUT 1`	Удаление правила из цепочки. Команда имеет два формата записи, первый — когда задается критерий сравнения с опцией `-D`, второй — порядковый номер правила. Если задается критерий сравнения, то удаляется правило, которое имеет в себе этот критерий, если задается номер правила, то будет удалено правило с заданным номером. Счет правил в цепочках начинается с 1

Таблица 32.7 *(окончание)*

Команда	Пример использования	Описание
`-R, --replace`	`iptables -R INPUT 1 -s 192.168.0.1 -j DROP`	Команда заменяет одно правило другим
`-I, --insert`	`iptables -I INPUT 1 --dport 80 -j ACCEPT`	Вставляет новое правило в цепочку. Число, следующее за именем цепочки, указывает номер правила, перед которым нужно вставить новое правило
`-L, --list`	`iptables -L INPUT`	Вывод списка правил в заданной цепочке. Если имя цепочки не указывается, то выводится список правил для всех цепочек
`-F, --flush`	`iptables -F INPUT`	Удаление всех правил из заданной таблицы. Если имя цепочки и таблицы не указывается, то удаляются все правила, во всех цепочках
`-Z, --zero`	`iptables -Z INPUT`	Обнуление всех счетчиков в заданной цепочке. Если имя цепочки не указывается, то подразумеваются все цепочки
`-N, --new-chain`	`iptables -N allowed`	Создается новая цепочка с заданным именем в заданной таблице. Имя цепочки должно быть уникальным и не совпадать с зарезервированными именами цепочек и действий
`-X, --delete-chain`	`iptables -X allowed`	Удаление заданной цепочки из заданной таблицы. Удаляемая цепочка не должна иметь правил и не должно быть ссылок из других цепочек на удаляемую цепочку
`-P, --policy`	`iptables -P INPUT DROP`	Задает политику по умолчанию для заданной цепочки. Политика определяет действие, применяемое к пакетам, не попавшим ни под одно из правил в цепочке
`-E, --rename-chain`	`iptables -E allowed disallowed`	Команда переименовывает пользовательскую цепочку

Критерии проверки пакетов

Критерии проверки пакетов можно разделить на пять групп.

- Общие критерии — для любых правил.
- TCP-критерии — только для TCP-пакетов.
- UDP-критерии — только для UDP-пакетов.
- ICMP-критерии — для работы с ICMP-пакетами.
- Специальные критерии: state, owner, limit и пр.

Общие критерии

Общие критерии (табл. 32.8) допустимы в любых правилах, они не зависят от типа протокола и не требуют модулей расширения.

Таблица 32.8. *Общие критерии проверки пакетов*

Критерий	Пример использования	Описание
`-p, --protocol`	`iptables -A INPUT -p tcp`	Этот критерий указывает тип протокола. Допустимые значения: `TCP`, `UDP` и `ICMP` или ключевое слово `ALL`. Для логической инверсии критерия перед именем протокола указывают символ `!`
`-s, --src, --source`	`iptables -A INPUT -s 192.168.1.1`	IP-адрес источника пакета. Адрес источника может указываться так, как показано в примере, тогда подразумевается единственный IP-адрес. А можно указать адрес в виде *<address>/<mask>*, например как 192.168.0.0/255.255.255.0, или более современным способом 192.168.0.0/24. Символ `!`, установленный перед адресом, означает логическое отрицание
`-d, --dst, --destination`	`iptables -A INPUT -d 192.168.1.1`	IP-адрес получателя. Можно определять как единственный IP-адрес, так и диапазон адресов. Символ `!` означает логическую инверсию критерия
`-i, --in-interface`	`iptables -A INPUT -i eth0`	Интерфейс, с которого был получен пакет. Критерий допустим только в цепочках input, forward и prerouting. При отсутствии этого критерия предполагается любой интерфейс. Символ `!` инвертирует результат совпадения. Если имя интерфейса завершается символом `+`, то критерий задает все интерфейсы, начинающиеся с заданной строки
`-o, --out-interface`	`iptables -A FORWARD -o eth0`	Задает имя выходного интерфейса. Этот критерий допустим только в цепочках output, forward и postrouting. При отсутствии этого критерия предполагается любой интерфейс, что равносильно использованию критерия `-o +`. Символ `!` инвертирует результат совпадения. Если имя интерфейса завершается символом `+`, то критерий задает все интерфейсы, начинающиеся с заданной строки
`-f, --fragment`	`iptables -A INPUT -f`	Правило распространяется на все части фрагментированного пакета, кроме первого, сделано это потому, что нет возможности определить исходящий/входящий порт для фрагмента пакета, а для ICMP-пакетов определить их тип. Допускается использование символа `!` для инверсии результата сравнения, только в данном случае символ `!` должен предшествовать критерию `-f` (т. е. `! -f`)

TCP-критерии

Этот набор критериев работает только с TCP-пакетами (табл. 32.9). В правилах вам потребуется указать тип протокола `--protocol tcp`.

Таблица 32.9. *TCP-критерии проверки пакетов*

Критерий	Пример использования	Описание
`--sport`, `--source-port`	`iptables -A INPUT -p tcp --sport 22`	Исходный порт, с которого был отправлен пакет. В качестве параметра может указываться номер порта или название сетевой службы. Номера портов могут задаваться в виде интервала из минимального и максимального номеров, например, `--source-port 22:80`. Символ `!` служит для инверсии
`--dport`, `--destination -port`	`iptables -A INPUT -p tcp --dport 22`	Порт или диапазон портов, на который адресован пакет
`--tcp-flags`	`iptables -p tcp --tcp-flags SYN,FIN,ACK SYN`	Определяет маску и флаги TCP-пакета. Пакет считается удовлетворяющим критерию, если из перечисленных флагов в первом списке в единичное состояние установлены флаги из второго списка. Аргументами критерия могут быть флаги `SYN`, `ACK`, `FIN`, `RST`, `URG`, `PSH`, а также зарезервированные идентификаторы `ALL` и `NONE`. Символ `!` означает инверсию критерия. Имена флагов в каждом списке разделяют запятыми, пробелы служат для разделения списков
`--syn`	`iptables -p tcp --syn`	Этот критерий аналогичен критерию `--tcp-flags SYN,ACK,FIN SYN`. Такие пакеты открывают соединения TCP. Заблокировав такие пакеты, вы надежно запретите все входящие запросы на соединение, однако этот критерий не способен заблокировать исходящие запросы на соединение
`--tcp-option`	`iptables -p tcp --tcp-option 16`	Удовлетворяющим условию данного критерия будет считаться пакет, TCP-параметр которого равен заданному числу. Допускается использование флага инверсии условия `!`

UDP-критерии

Этот набор критериев работает только с UDP-пакетами (табл. 32.10). В правилах вам следует указать тип протокола `--protocol udp`.

Таблица 32.10. *UDP-критерии проверки пакетов*

Критерий	Пример использования	Описание
`--sport`, `--source-port`	`iptables -A INPUT -p udp --sport 53`	Исходный порт, с которого был отправлен пакет. В качестве параметра может указываться номер порта или название сетевой службы. Номера портов могут задаваться в виде интервала из минимального и максимального номеров. Символ `!` служит для инверсии
`--dport`, `--destination -port`	`iptables -A INPUT -p udp --dport 53`	Порт, на который адресован пакет. Формат аргументов полностью аналогичен `--source-port`

ICMP-критерии

Этот набор критериев работает только с ICMP-пакетами (табл. 32.11). В правилах вам необходимо указать тип протокола `--protocol icmp`.

Таблица 32.11. *ICMP-критерии проверки пакетов*

Критерий	Пример использования	Описание
`--icmp-type`	`iptables -A INPUT -p icmp --icmp-type 8`	Тип сообщения ICMP определяется номером или именем

Специальные критерии

Перед использованием эти расширения нужно загрузить явно, с помощью ключа `-m` или `--match`. Если мы собираемся задействовать критерии `state`, то должны явно указать это в строке правила `-m state` слева от критерия (табл. 32.12).

Таблица 32.12. *Специальные критерии проверки пакетов*

Критерий	Пример использования	Описание
`--limit`	`iptables -A INPUT -m limit --limit 3/hour`	Средняя скорость "освобождения емкости" за единицу времени. В качестве аргумента указывается число пакетов и время. Допустимые единицы измерения времени: `/second`, `/minute`, `/hour`, `/day`
`--limit-burst`	`iptables -A INPUT -m limit --limit-burst 5`	Максимальное значение числа `burst limit` для критерия `limit`. Это число увеличивается на единицу, если получен пакет, подпадающий под действие данного правила, и при этом средняя скорость поступления пакетов (задаваемая ключом `--limit`) уже достигнута. Так происходит до тех пор, пока число `burst limit` не достигнет максимального значения, устанавливаемого ключом `--limit-burst`. После этого правило начинает пропускать пакеты со скоростью, задаваемой ключом `--limit`

Таблица 32.12 *(окончание)*

Критерий	Пример использования	Описание
`--mac-source`	`iptables -A INPUT -m mac --mac-source 00:00:00:00:00:01`	MAC-адрес сетевого узла, передавшего пакет. MAC-адрес должен указываться в форме *XX*:*XX*:*XX*:*XX*:*XX*:*XX*. Этот критерий имеет смысл только в цепочках prerouting, forward и input
`--mark`	`iptables -t mangle -A INPUT -m mark --mark 1`	Проверяет пакеты, которые были предварительно "помечены". Метки устанавливаются действием `MARK`
`--source-port`	`iptables -A INPUT -p tcp -m multiport --source-port 22,53,80,110`	Список исходящих портов. С помощью данного критерия можно указать до 15 различных портов. Названия портов в списке должны отделяться друг от друга запятыми, пробелы в списке недопустимы
`--destination-port`	`iptables -A INPUT -p tcp -m multiport --destination-port 22,53,80,110`	Список входных портов
`--port`	`iptables -A INPUT -p tcp -m multiport --port 22,53,80,110`	Проверка как исходящего, так и входящего портов пакета
`--uid-owner`	`iptables -A OUTPUT -m owner --uid-owner 500`	Проверка "владельца" по User ID (UID), которая может использоваться для блокировки выхода в Интернет отдельных пользователей
`--gid-owner`	`iptables -A OUTPUT -m owner --gid-owner 0`	Проверка "владельца" пакета по Group ID (GID)
`--pid-owner`	`iptables -A OUTPUT -m owner --pid-owner 78`	Проверка "владельца" пакета по Process ID (PID)
`--sid-owner`	`iptables -A OUTPUT -m owner --sid-owner 100`	Проверка Session ID пакета
`--state`	`iptables -A INPUT -m state --state RELATED,ESTABLISHED`	Проверка признака состояния соединения (state). На сегодняшний день можно указывать четыре состояния: `INVALID` — пакет связан с неизвестным потоком или соединением и, возможно, содержит ошибку в данных или в заголовке; `ESTABLISHED` — пакет принадлежит уже установленному соединению, через которое пакеты идут в обоих направлениях; `NEW` — пакет открывает новое соединение или пакет принадлежит однонаправленному потоку; `RELATED` — пакет принадлежит уже существующему соединению, но при этом он открывает новое соединение
`--tos`	`iptables -A INPUT -p tcp -m tos --tos 0x16`	Проверка установленных битов `TOS`. Аргументом может быть десятичное или шестнадцатеричное число
`--ttl`	`iptables -A OUTPUT -m ttl --ttl 60`	Проверка поля `TTL` на равенство заданному значению

Действия и переходы

Действия и переходы сообщают правилу, что необходимо выполнить, если пакет соответствует заданному критерию.

Действие — это предопределенная команда, описывающая действие, которое необходимо выполнить, если пакет совпал с заданным критерием.

Переходы в правилах описывают точно так же, как и действия, т. е. ставят ключ `-j` и указывают название цепочки правил, на которую выполняется переход. На переходы есть ограничения: цепочка, на которую выполняется переход, должна находиться в той же таблице, что и цепочка, из которой этот переход выполняется. Цепочка, являющаяся целью перехода, должна быть создана до того, как на нее будут выполняться переходы.

При выполнении перехода iptables продолжит движение пакета по цепочке, в которую был произведен переход. Если пакет достиг конца цепочки, то он будет возвращен в вызывающую цепочку и движение пакета продолжится с правила, следующего за правилом, вызвавшим переход. Если к пакету во вложенной цепочке будет применено действие `ACCEPT`, то пакет будет считаться принятым и в вызывающей цепочке и его движение по вызывающим цепочкам прекратится.

Действие *ACCEPT*

Если над пакетом выполняется действие `ACCEPT`, то пакет прекращает движение по цепочке и считается принятым. Однако пакет продолжит движение по цепочкам в других таблицах и может быть отвергнут там. Действие задается с помощью ключа`-j ACCEPT`.

Действие *DNAT*

`DNAT` (Destination Network Address Translation) преобразует адрес места назначения в IP-заголовке пакета. Если пакет подпадает под критерий правила, выполняющего `DNAT`, то пакет и все последующие пакеты из этого же потока будут подвергнуты преобразованию адреса назначения и переданы на требуемый адрес. Действие `DNAT` может выполняться только в цепочках prerouting и output таблицы nat и во вложенных цепочках.

Действие *DROP*

Данное действие удаляет пакет. Пакеты прекращают свое движение полностью — они не передаются в другие таблицы, как это происходит при действии `ACCEPT`.

Действие *LOG*

LOG-действие служит для журналирования отдельных пакетов и событий. В журнал могут заноситься IP-заголовки пакетов и другая интересующая вас информация. Информация из журнала затем может быть прочитана с помощью dmesg или syslogd.

Действие *MARK*

Устанавливает метки для определенных пакетов. Это действие может выполняться лишь в пределах таблицы mangle. Метка пакета существует только в период времени, пока пакет не покинул брандмауэр, таким образом, метка не передается по сети.

Действие *MASQUERADE*

Маскарадинг подразумевает получение IP-адреса от заданного сетевого интерфейса, вместо прямого его указания. Действие `MASQUERADE` имеет хорошее свойство "забывать" соединения при остановке сетевого интерфейса.

`MASQUERADE` допускается указывать только в цепочке postrouting таблицы nat.

Действие *MIRROR*

`MIRROR` заменяет в пакете поле `source` на `destination` и `destination` на `source`. Данное действие допустимо только в цепочках input, forward и prerouting, а также в цепочках, вызываемых из этих трех.

Действие *QUEUE*

`QUEUE` ставит пакет в очередь на обработку пользовательскому процессу. Оно может быть использовано для нужд учета или дополнительной фильтрации пакетов.

Действие *REDIRECT*

`REDIRECT` перенаправляет пакеты и потоки на другой порт того же самого хоста. Действие `REDIRECT` очень удобно для выполнения "прозрачного" проксирования (transparent proxying), когда машины в локальной сети даже не подозревают о существовании proxy.

Действие *REJECT*

Используется в тех же самых ситуациях, что и `DROP`, но в отличие от него, `REJECT` выдает сообщение об ошибке на хост, передавший пакет.

Действие *RETURN*

`RETURN` прекращает движение пакета по текущей цепочке правил и производит возврат в вызывающую цепочку, если текущая цепочка была вложенной, или если текущая цепочка лежит на самом верхнем уровне (например, input), то к пакету будет применена политика по умолчанию.

Действие *SNAT*

Преобразует сетевые адреса, т. е. изменяет исходящий IP-адрес в IP-заголовке пакета. `SNAT` допускается только в таблице nat, в цепочке postrouting. Если первый

пакет в соединении подвергся преобразованию исходящего адреса, то все последующие пакеты из этого же соединения будут преобразованы автоматически и не пойдут через эту цепочку правил.

Действие *TOS*

Команда `TOS` устанавливает биты в поле `Type of Service` IP-заголовка пакета.

Действие *TTL*

Изменяет содержимое поля `Time To Live` в IP-заголовке пакета. Действие `TTL` можно указывать только в таблице mangle и нигде больше.

Действие *ULOG*

Действие `ULOG` предоставляет возможность журналирования пакетов в пользовательское пространство. Оно заменяет традиционное действие `LOG`, базирующееся на системном журнале. При использовании этого действия пакет передается специальному демону, который может выполнять детальное журналирование в различных форматах (обычный текстовый файл, база данных MySQL и т. д.) и поддерживает возможность добавления надстроек для формирования разных выходных форматов и обработки сетевых протоколов.

Утилиты iptables

В этом разделе приведены некоторые утилиты для управления iptables.

Iptables-save

Утилита iptables-save предназначена для сохранения текущего набора правил в файл, который затем может быть использован утилитой iptables-restore. Эта команда очень проста и имеет всего два аргумента:

```
iptables-save [-c] [-t table]
```

Первый аргумент `-c` (или `--counters`) заставляет iptables-save сохранить значения счетчиков байтов и пакетов.

С помощью ключа `-t` (или `--table`) можно указать имя таблицы для сохранения. Если ключ `-t` не задан, то сохраняются все таблицы.

Iptables-restore

Утилита iptables-restore служит для восстановления набора правил, которые ранее были сохранены с помощью iptables-save. Iptables-restore получает набор правил со стандартного ввода и не может загружать его из файла напрямую. Синтаксис команды:

```
iptables-restore [-c] [-n]
```

Ключ `-c` (или `--counters`) позволяет восстановить значения счетчиков.

Ключ `-n` (или `--noflush`) сообщает iptables-restore о том, что правила должны быть добавлены к имеющимся. По умолчанию iptables-restore очищает содержимое таблиц и цепочек перед загрузкой нового набора правил.

Ссылки

- Ipchains-HOWTO.
- **bog.pp.ru/work/ipchains.html** — Bog BOS: ipchains: фильтрация пакетов в Linux: принципы работы, установка и настройка.
- **gazette.linux.ru.net/rus/articles/iptables-tutorial.html** — Andreasson O. Iptables Tutorial 1.1.19. Перевод А. Киселева.
- Зиглер Р. Брандмауэры в Linux: учебн. пособие; пер. с англ. — М.: Издательский дом "Вильямс", 2000.



Глава 33

Организация шлюза в Интернет для локальной сети

В этой главе мы займемся созданием точки доступа в Интернет для локальной сети. Раньше для этого по концам выделенной линии устанавливали модемы, подключаемые к последовательному порту. В последнее время используются технологии *x*DSL или оптико-волоконные сети, позволяющие подключать специальные достаточно дорогие модемы по интерфейсу Ethernet напрямую к сетевой карте, причем эти устройства сами зачастую являются маршрутизаторами. Для определенности будем считать, что у нас есть модем, подключенный к последовательному порту.

Обычно в небольших локальных сетях выделяется один компьютер, который и выполняет функции маршрутизатора между локальной сетью и Интернетом, а также счетчика трафика, брандмауэра, ограничителя скорости Web-сервера и т. п. Почти все, что необходимо для создания такой многопрофильной системы, мы уже описывали ранее, поэтому в данной главе остановимся только на тех проблемах, которые еще не рассматривались.

Начальные установки

Как правило, во всех современных дистрибутивах Linux ядро собрано так, что оно работает как маршрутизатор пакетов между разными сетями и поддерживает механизм защиты маршрутизируемых пакетов и подсчет статистики.

Однако не будет лишним убедиться перед началом настройки системы, что в ядре вашей операционной системы присутствуют следующие необходимые для построения маршрутизатора элементы (функции):

- Networking support (поддержка сетевых свойств);
- TCP/IP networking (поддержка TCP/IP);
- IP forwarding/gatewaying (поддержка IP-маршрутизации);
- IP multicasting (поддержка специфических свойств IP-протокола);
- IP firewalling (поддержка брандмауэров);
- IP accounting (поддержка управления IP);
- Network device support (поддержка сетевых устройств).

Помимо этого, ядро операционной системы должно уметь работать с сетевыми картами, установленными на вашем компьютере, и поддерживать протокол PPP (Point-to-Point Protocol).

Разумеется, следует правильно настроить сетевое оборудование, IP-адреса и т. п.

Связь с провайдером

Возможны два варианта подключения локальной сети к Интернету при помощи модема. Первый из них предназначен для тех, кто платит за трафик, а второй используется теми, кто оплачивает время, проведенное в Интернете.

В первом случае выход в Интернет осуществляется при помощи стандартного для Linux набора программ — pppd, chat и, возможно, еще нескольких скриптов. Происходит это следующим образом: вначале маршрутизатор дозванивается до провайдера и устанавливает с ним связь по протоколу PPP (или SLIP, который сейчас встречается крайне редко). После установления соединения полученным каналом может пользоваться любой компьютер в вашей локальной сети (при соответствующей настройке). Канал удерживается до тех пор, пока не выключится ваш маршрутизатор или администратор явным образом не разорвет соединение.

Второй вариант — модификация первого, в англоязычной литературе он носит название dial on demand (звонок по требованию). Для этого дополнительно требуется программа diald, с помощью которой можно организовать работу так, что если в течение заранее обусловленного времени не происходит обмена данными между локальной сетью и Интернетом, то diald разрывает соединение. При первой же попытке пользователя подключиться к Интернету diald снова дозванивается и устанавливает связь.

Поскольку второй вариант более сложный, будем рассматривать его как основной для организации нашего маршрутизатора.

Схема организации подключения локальной сети

Перечислим требования, выполнение которых необходимо для подключения локальной сети к Интернету:

- возможность доступа в Интернет — модем, телефонный номер и подключение к провайдеру;
- набор программ для организации связи — pppd, chat и diald;
- средство для управления брандмауэром — утилиты ipchains или iptables;
- средство для ограничения трафика (если необходимо);
- программное обеспечение для организации proxy-сервера;
- программное обеспечение для учета и просмотра статистики.

Теперь, когда цели и средства известны, можно приступать к настройке программ.

Организация связи через коммутируемое соединение

Старейший вариант соединения с провайдером, который, к сожалению, еще кое-где встречается в нашей стране. По сравнению с организацией связи по выделенному каналу представляет собой более сложную схему, поэтому рассмотрим ее первой.

Настройка программ

Допустим, что на компьютере, который будет выходить в Интернет, правильно настроены сетевые параметры, и вы убедились в работоспособности локальной сети. Следующий шаг — добиться устойчивой связи с провайдером на вашем компьютере-маршрутизаторе.

Настройка связи с провайдером

Настроим подсистему дозвона и соединения с провайдером. Для удобства разобьем работу на два этапа:

1. Настройка PPP-соединения.
2. Установка и конфигурирование демона дозвона по требованию (diald).

Настройку модемного соединения мы здесь описывать не будем, поскольку эта достаточно простая задача подробно рассмотрена в работе одного из отечественных патриархов Linux — В. Водолазкого "Установка PPP-соединения в Linux".

Почему мы выбираем протокол PPP? Основные преимущества протокола PPP по сравнению с протоколом SLIP:

- IP-адреса в PPP назначают с помощью демона pppd, что значительно упрощает процесс конфигурирования при наличии динамических IP-адресов;
- ошибки, возникающие при передаче данных, корректируют между компьютером провайдера и клиента, а не между удаленным компьютером, откуда берутся данные, и потребителем, как в протоколе SLIP.

Для организации связи между провайдером и клиентом необходимо получить данные, приведенные в табл. 33.1.

Таблица 33.1. *Необходимые данные для настройки модемного соединения*

Необходимые данные	Значения в примере
Имя пользователя (login)	Myname
Пароль пользователя (password)	Vasya
IP-адрес пользователя (если есть)	192.168.0.100
IP-адрес сервера DNS	192.168.10.1

Процесс установления связи между вами и провайдером состоит из трех этапов:

- соединение с компьютером провайдера с помощью модема;
- регистрация пользователя в удаленной системе;
- установка PPP-соединения.

Для решения этих задач в Linux есть несколько скриптов, каждый из которых выполняет какую-то небольшую функцию. А поскольку это набор скриптов, никто не мешает на их базе определить именно те действия, которые необходимы вам при установлении или обрыве PPP-соединения.

Размещение скриптов зависит от настройки вашего дистрибутива. В современных версиях дистрибутивов семейства Red Hat предусмотрено два места: каталоги /etc/ppp и /etc/sysconfig/network-scripts. Наименования скриптов могут быть произ-

вольными и очень часто зависят от предпочтений сборщика дистрибутива или системного администратора.

Будем считать, что у нас есть следующие файлы:

- /etc/ppp/chap-secrets — для аутентификации пользователя провайдером по протоколу chap. Обычно содержит имя и пароль пользователя для входа к провайдеру. В нашем случае это будет выглядеть следующим образом:

```
myname * vasya;
```

- /etc/ppp/pap-secrets — для аутентификации пользователя провайдером по протоколу pap. Обычно содержит имя и пароль пользователя для входа к провайдеру. В нашем случае это будет выглядеть так:

```
myname * vasya;
```

- /etc/ppp/ip-up — для соединения с провайдером. Зачастую этот файл содержит только одну строку:

```
/usr/sbin/pppd
```

  Здесь можно настроить установление модемом соединения с провайдером или вызвать необходимый скрипт или программу;
- /etc/ppp/ip-down — для разрыва соединения с провайдером;
- /etc/ppp/options — это, пожалуй, самый сложный и ответственный файл. Он определяет параметры нашего модема, скорость передачи по последовательному интерфейсу данных, настройки программы pppd и некоторые другие параметры. Обычно файл /etc/ppp/options оставляют неизменным, а для конфигурирования параметров соединения создают копию с именем /etc/ppp/options.ttySX, где ttySX — имя последовательного порта, к которому подключен наш модем. Пусть для определенности модем подключен к ttyS0 (COM1).

  Листинг 33.1 содержит пример файла /etc/ppp/options.ttySX.

Листинг 33.1

```
# Устройство
/dev/ttyS0
# Скорость
115200
mru 1500
# наш интерфейс : удаленный интерфейс
192.168.0.100:192.168.0.101
# маска подсети
netmask 255.255.255.0
bsdcomp 0
chap-interval 15
debug
crtscts
defaultroute
```

Первые две строки определяют последовательный порт, к которому подключен наш модем, и скорость, на которой будет происходить обмен между модемом и последовательным портом. Далее обратите внимание на строку

```
192.168.0.100:192.168.0.101,
```

которая определяет IP-адреса нашего последовательного интерфейса и провайдера. Такую строку необходимо добавить, если провайдер выдал нам постоянный IP-адрес. Как правило, в современном мире с коммутируемыми соединениями такого не происходит. Для статического IP-адреса также необходимо задать маску подсети.

Поскольку наш компьютер является маршрутизатором для локальной сети, необходимо настроить маршрутизацию с помощью программы route и документации, прилагаемой к ней. В том случае, если у вас одно подключение к провайдеру (а мы предположили, что точка подключения к провайдеру у нас одна), можно в конец файла вписать команду `defaultroute`, что позволит вам добавить маршрут в системную таблицу маршрутизации, используя удаленную сторону как шлюз.

Команды pppd

Далее мы рассмотрим основные команды программы pppd (табл. 33.2).

***Таблица 33.2.** Основные команды программы pppd*

Команда	Описание
`asyncmap 0`	Async-карта символов — 32-bit hex; каждый бит — символ, который нужно представить в виде escape-последовательности, чтобы pppd мог его принять
`auth`	Требует от удаленной стороны назвать себя перед тем, как начнется обмен пакетами
`bsdcomp 0`	Определяет сжатие передаваемого трафика. На обычном модемном соединении не используется, позволяет в некоторых случаях почти в два раза увеличить объем передаваемых данных за единицу времени
`chap-interval` *интервал*	Определяет, что pppd будет заново вызывать удаленную сторону каждые *интервал* секунд
`chap-restart` *интервал*	Устанавливает интервал рестарта `chap` (пауза возобновления передач challenges) в *интервал* секунд
`chap-max-challenge` *значение*	Устанавливает максимальное число передач `chap challenge`
`connect` *<программа>*	Определяет программу для установки соединения
`Crtscts`	Задает аппаратное управление потоком данных на последовательном порту
`Debug`	Предписывает увеличить уровень отладки. Если эта опция есть, pppd будет записывать в журнал все прибывшие и отправленные пакеты в понятной для человека форме. Пакеты регистрируются в log-файлах через syslog. Эту информацию можно перенаправить в файл соответствующей установкой /etc/syslog.conf

Таблица 33.2 *(продолжение)*

Команда	Описание
`disconnect <программа>`	Предписывает запустить данную программу после того, как программа pppd завершила связь
`domain имя_домена`	Добавляет имя домена к имени машины
`ipcp-max-configure значение`	Устанавливает максимальное число передач IPCP configure-request
`ipcp-max-terminate значение`	Устанавливает максимальное число передач IPCP terminate-request
`ipcp-max-failure значение`	Устанавливает максимальное число IPCP configure-NAK, возвращенных перед началом отправки вместо configure-Rejects
`ipcp-restart интервал`	Устанавливает интервал перезапуска IPCP в *интервал* секунд
`local`	Предписывает не задействовать линии управления модемом
`lock`	Предписывает, что pppd должна использовать `lock` в стиле UUCP для последовательного устройства
`login`	Предписывает осуществлять идентификацию удаленной стороны на основе базы данных паролей
`modem`	Предписывает использовать линии управления модемом
`mru число`	Устанавливает значение MRU (Maximum Receive Unit, максимально принимаемый пакет) в *число*. При договоренности pppd запросит удаленную сторону отправлять пакеты не более чем по `число` байтов. Минимальное значение MRU — 128. Значение MRU по умолчанию — 1500. Для медленных соединений рекомендуется 296 (40 байтов для заголовка TCP/IP плюс 256 байтов данных)
`mtu число`	Устанавливает значение MTU (Maximum Transmit Unit, максимально передаваемый пакет) в *число*. Пока другая сторона не попросит меньшее значение при договоре о MRU, pppd будет требовать у сетевого кода ядра отправлять пакеты данных не более чем по `число` байтов через сетевой интерфейс PPP
`name имя_машины`	Устанавливает имя машины (для аутентификации)
`noauth`	Не требует удаленную сторону назвать себя перед тем, как начнется обмен пакетами
`noipdefalut`	Запрещает поведение по умолчанию, когда не указан локальный IP-адрес, которое определяет локальный IP-адрес по имени хоста. С этой опцией удаленная сторона должна обеспечить локальный IP-адрес в течение IPCP-переговоров (если она не определена явно в командной строке или в файле options)
`pap-restart интервал`	Устанавливает интервал возобновления передачи PAP в *интервал* секунд
`pap-max-authreq значение`	Устанавливает максимальное число передач PAP authenticate-request (запросов на аутентификацию по протоколу PAP)

Таблица 33.2 *(окончание)*

Команда	Описание
`passive`	Разрешает опцию `passive` в LCP. С этой опцией pppd будет пытаться инициировать соединение, а если ответ от другой стороны не принят, то pppd будет пассивно ожидать правильный LCP-пакет от другой стороны вместо выхода
`silent`	С этой опцией pppd не будет передавать LCP-пакеты для инициации соединения, пока не придет правильный LCP-пакет от другой стороны
`user` *имя*	Устанавливает имя пользователя для аутентификации этой машины на другой стороне, используя PAP. Нельзя указывать вместе с `name`
`xonxoff`	Задает программное управление потоком данных на последовательном порту
`+chap`	Двусторонняя chap-аутентификация
`+pap`	Двусторонняя pap-аутентификация
`-all`	Не разрешает договариваться о любых опциях LCP и IPCP
`-am`	Запрещает договариваться о `asyncmap`
`-chap`	Предписывает отказаться от chap-аутентификации
`-d`	Устанавливает уровень отладки. Если эта опция есть, pppd будет записывать в журнал все прибывшие и отправленные пакеты в понятной для человека форме. Пакеты регистрируются в log-файлах через syslog. Эта информация может быть перенаправлена в файл соответствующей установкой /etc/syslog.conf
`-detach`	Предписывает не переходить в фоновый режим
`-ip`	Предписывает не договариваться об IP-адресе
`-mru`	Запрещает договариваться о mru
`-pap`	Предписывает отказаться от pap-аутентификации
`-pc`	Запрещает сжатие полей протокола

Как видите, параметров много и для полного понимания вопроса необходимо изучить соответствующую документацию.

Настройка diald

Обычно программа diald входит в стандартный дистрибутив, и установка ее с помощью менеджера пакетов rpm занимает немного времени. После инсталляции необходимо привести стандартную конфигурацию программы diald в соответствие с нашими реалиями.

Чтобы лучше понять то, что мы будем делать дальше, поговорим немного о принципе работы программы diald. Программа создает соединение на псевдотерминале и устанавливает маршрутизацию на получившийся интерфейс. После этого

она начинает отслеживать пакеты, проходящие по виртуальному каналу. Если кто-то пытается выйти в Интернет, diald перехватывает данные, анализирует их и на основе правил, определяемых администратором, присваивает им определенные тайм-ауты. Далее пакеты отправляются по назначению, а тайм-ауты заносятся в так называемый набор соединения. Как только в наборе появляется первый тайм-аут, diald начинает дозваниваться до провайдера и пытается установить соединение. Организовав сеанс связи, демон переустанавливает маршрутизацию на реальный канал. Таким образом, связь с внешним миром оказывается установленной.

На протяжении всего времени соединения продолжает обновляться набор соединения. Истекшие тайм-ауты удаляются, новые поступают. И так продолжается, пока по какой-либо причине трафик не прекратится. Тайм-аутов в наборе становится все меньше и меньше, и когда последний из них оканчивается, diald разрывает связь.

Теперь перейдем непосредственно к конфигурированию. Этот процесс состоит из трех этапов:

- создание скрипта соединения — файл /etc/diald/connect;
- настройка основной конфигурации — файл /etc/diald.conf;
- настройка правил тайм-аутов — файл /etc/diald/standard.filter.

Создание скрипта соединения: /etc/diald/connect

Как мы уже знаем, для организации сеанса связи необходимо выполнить несколько действий: дозвониться по телефону до поставщика услуг, пройти процедуру авторизации и запустить PPP-соединение. Поскольку у разных провайдеров данный процесс может коренным образом отличаться, то нет смысла встраивать такую процедуру в программу, а следует задействовать внешний скрипт. Для этого достаточно подправить скрипт, входящий в стандартную поставку diald.

В листинге 33.2 приведен вариант файла /etc/diald/connect.

Листинг 33.2

```
#!/bin/sh

NIT="ATZ"     # Строка инициализации модема
PHONE="223322"     # Телефон провайдера
ACCOUNT="myname"     # логин
PASSWORD="vasya"     # пароль
# Определяем функцию для посылки
# сообщений в системный журнал
# и в FIFO-канал diald
function message ()
 {
   [ $FIFO ] && echo "message $*" >$FIFO
   logger -p local2.info -t connect "$*"
 }
```

```
# Начинаем процедуру связи
# Инициализируем модем
message "*** Initializing Modem ***"
chat "" $INIT OK ""
if [ $? != 0 ]
   then
      message "!!! Failed to initialize modem !!!"
      exit 1
fi
# Пытаемся дозвониться
message "*** Dialing system  ***"
chat \
      ABORT "NO CARRIER" \
      ABORT BUSY \
      ABORT "NO DIALTONE" \
      ABORT ERROR \
      "" ATDT$PHONE \
      CONNECT ""
case $? in
      0) message "*** Connected ***";;
      1) message "!!! Chat Error !!!"; exit 1;;
      2) message "!!! Chat Script Error !!!"; exit 1;;
      3) message "!!! Chat Timeout !!!"; exit 1;;
      4) message "!!! No Carrier !!!"; exit 1;;
      5) message "!!! Busy !!!"; exit 1;;
      6) message "!!! No DialTone !!!"; exit 1;;
      7) message "!!! Modem Error !!!"; exit 1;;
      *) esac
# Проходим авторизацию
message "*** Send login and password ***"
chat \
      login: $ACCOUNT \
      password: $PASSWORD        TIMEOUT 5 ""
if [ $? != 0 ] then
      message "!!! Failed to send !!!"
      exit 1
fi
# Все прошло удачно!
message "*** Protocol started *** "
```

Код, приведенный в листинге 33.2, — это просто сценарий на языке командной оболочки, который необходимо немного адаптировать для ваших параметров.

Настройка основной конфигурации: /etc/diald.conf

/etc/diald.conf — основной конфигурационный файл программы diald, в котором задаются параметры устанавливаемого соединения и определяется поведение программы. Набор команд конфигурации у diald весьма обширен, поэтому в приведенном далее примере оставлены только необходимые, а подробную информацию по конфигурационным командам можно посмотреть в документации на программу diald.

Листинг 33.3 иллюстрирует содержимое файла diald.conf.

Листинг 33.3

```
# Протокол для связи с провайдером
mode ppp
# Вести журнал сеансов связи diald.log
accounting-log /var/log/diald.log
# Для управления демоном из внешних программ
# организовать канал FIFO — diald.ctl.
fifo /etc/diald/diald.ctl
# Для дозвона использовать файл /etc/diald/connect
connect /etc/diald/connect
# Далее несколько команд, описывающих применяемый модем.
# Поскольку мы уже определили параметры в /etc/ppp/options,
# то нижеприведенные команды необходимо закомментировать во избежание
# конфликтов в файле /etc/ppp/options
# device /dev/modem
# speed 115200
# modem
# lock
# crtscts
# Назначаем локальный и удаленный адреса нашего
# соединения. Если при связи с провайдером IP-адрес
# для вас выделяется динамически, то здесь можно
# поставить любые свободные адреса из диапазона,
# оговоренного при настройке нашей TCP/IP-сети.
# При запуске PPP diald сам выставит корректные значения
local 192.168.0.100
remote 192.168.0.101
# Провайдер дает нам динамический IP
dynamic
# Установить маршрут по умолчанию
# на виртуальное соединение
defaultroute
# Максимальное количество неудачных попыток дозвона
dial-fail-limit 10
```

```
# Задержка между попытками дозвона
redial-timeout 5
# время ожидания завершения скрипта connect
connect-timeout 120
# Файл с правилами для тайм-аутов
include /etc/diald/standard.filter
```

Настройка правил тайм-аутов: /etc/diald/standard.filter

Следующее, что вы должны сделать — настроить правила тайм-аутов. Это самый сложный момент конфигурирования diald, т. к. требует знания внутренней структуры IP-пакетов. Однако разработчики diald — люди добрые, и стандартный файл standard.filter имеет вполне приемлемые для большинства случаев настройки. Оставив в нем все как есть, мы получим набор правил, рассчитанный на трехминутную паузу между окончанием активности в Интернете и разрывом связи с провайдером.

Комплексное тестирование

После проделанных манипуляций настало время проверить, правильно ли настроены наши программы. Для этого на компьютере желательно временно отключить все настройки брандмауэра (если вы, конечно, установили его). Затем необходимо запустить программу diald и попытаться выйти в "большой мир". Можно использовать браузер lynx (и зайти, например, на сайт **http://www.bhv.ru**), можно — программу ping.

Если все было настроено корректно, то после ввода предыдущей команды модем должен начать дозваниваться до провайдера. Через некоторое время связь будет установлена. Однако практически всегда lynx выдает сообщение о том, что не может соединиться с удаленным сервером! В данном случае — это нормальное явление. Дело в том, что при PPP-соединении с динамическими IP-адресами в силу определенных особенностей первый пакет обычно бывает утерян и не доходит до адресата. В результате мы ждем ответа от сервера, а он об этом и не подозревает. Достаточно повторить введенную ранее команду, чтобы все заработало.

Далее нам необходимо убедиться, что модем аккуратно разорвет соединение по прошествии трех минут. Дождавшись конца загрузки Web-страницы, засечем время. Примерно через три минуты diald должен дать команду на разрыв соединения.

Если у вас все прошло именно таким образом, значит, система работает правильно. В противном случае проанализируйте последние строки системного журнала (/var/log/messages).

Пока мы проверили корректную работу только с нашего компьютера-маршрутизатора. Однако нам нужно сделать то же самое и с любого компьютера в локальной сети, поэтому попробуем повторить описанную процедуру и на произвольном компьютере. Реакция diald должна быть аналогичной. Если что-то пошло не так, проверьте корректность настройки протокола TCP/IP на этой машине, в частности — настройки сетевого шлюза, которые должны указывать на наш компьютер-маршрутизатор.

Организация связи по выделенному каналу

В отличие от настройки связи по коммутируемому соединению, организация соединения по выделенному каналу гораздо проще. В большинстве случаев все это настраивается через Web-интерфейс в самом модеме, в котором, кстати, установлен Linux. И нам по большому счету на компьютере ничего настраивать не нужно. Но еще встречается старое оборудование на выделенных линиях. Вот для него и рассмотрим процесс конфигурирования.

Настройка связи с провайдером

Как и в предыдущем случае, нам необходимо правильно настроить программу pppd. Поскольку параметры программы pppd мы уже рассматривали, просто приведем файл options (листинг 33.4) и прокомментируем его содержание.

Листинг 33.4

```
# Устройство
/dev/ttyS0
# Скорость
115200
mru 1500
noauth
# наш интерфейс : удаленный интерфейс
192.168.0.100:192.168.0.101
# маска подсети
netmask 255.255.255.0
bsdcomp 0
chap-interval 15
debug
crtscts
-detach
defaultroute
```

Первые две строки определяют последовательный порт, к которому подключен наш модем, и скорость обмена между модемом и последовательным портом. Далее обратите внимание на строку

```
192.168.0.100:192.168.0.101
```

Она определяет IP-адреса нашего последовательного интерфейса и провайдера. Такую строку необходимо добавить, если провайдер выдал нам постоянный IP-адрес. Для статического IP-адреса также следует задать маску подсети.

Если у вас одно подключение к провайдеру, то можно в конец файла вписать команду `defaultroute`, что позволит вам добавить маршрут в системную таблицу маршрутизации, используя удаленную сторону как шлюз.

Вот и все, что требовалось для конфигурации программы pppd для соединения по выделенному каналу. Правда, намного проще, чем с коммутируемым?

Осталось только отредактировать файл inittab, чтобы pppd автоматически стартовала. Для этого необходимо добавить следующую строчку:

```
7 : 2345 : respawn: /usr/sbin/pppd file /etc/ppp/options.ttyS0 >
/var/log/pppS0.log
```

Комплексное тестирование

Теперь настало время проверить, правильно ли настроено наше соединение по выделенному каналу. Для этого перезагрузите компьютер-шлюз для вступления в силу изменений, внесенных в файл inittab, и временно отключите все настройки брандмауэра (если вы, конечно, установили его). Затем необходимо попытаться выйти в Интернет, например, с помощью программы ping:

```
ping lazzycat.com
```

Если все было настроено корректно, то вы увидите отклик от сайта.
Если у вас все прошло именно таким образом, значит, система работает правильно. В противном случае проанализируйте последние строки системного журнала (/var/log/messages).

Этим действием мы проверили корректную работу только с нашего компьютера-маршрутизатора. Однако нам нужно сделать то же самое и с любого компьютера в локальной сети. Если что-то пошло не так, проверьте настройки протокола TCP/IP на этой машине, в частности настройки сетевого шлюза, которые должны указывать на наш компьютер-маршрутизатор.

Итак, вы получили вполне работоспособный шлюз в Интернет для вашей локальной сети. Однако это далеко не все. Наша система открыта для любого постороннего вмешательства, а шлюз должен обеспечить защиту локальной сети извне и изнутри, вести учет потребленного трафика (зачастую на каждом компьютере), ограничить нас от нежелательной или сомнительной информации (например, баннеров), обработать статистику и красиво ее подать (лучше всего графически). Как видите, задач много, и мы будем их решать постепенно.

Защита локальной сети

Защита локальной сети — понятие комплексное и многогранное. В данном случае мы имеем в виду правильную настройку брандмауэра на нашем компьютере-шлюзе. Процедура настройки брандмауэра была описана ранее, и к этому вопросу добавить больше нечего.

Установка proxy-сервера

Следующее, что мы должны сделать для нашей локальной сети, — минимизировать расходы на потребляемый Интернетом трафик и увеличить скорость получения информации. Для решения этой проблемы есть стандартный рецепт — proxy-сервер. Что собой представляет proxy-сервер? Если с помощью браузера, настроен-

ного для работы через proxy-сервер, вы запросите из Интернета какой-либо документ, и при этом окажется, что некоторое время назад кто-то уже обращался с подобным запросом, вы получите документ незамедлительно, с максимальной скоростью, на которую способно ваше сетевое подключение, потому что направлена вам будет копия документа, взятая из кэша proxy-сервера. Если же в кэше proxy-сервера данный документ отсутствует, то proxy-сервер запросит удаленный Web-сервер, хранящий оригинал, выдаст документ вам, и одновременно положит копию документа в свой кэш на случай такого же запроса. Чем больше пользователей пользуются proxy-сервером, тем более существенной становится его помощь.

Наиболее известная программа proxy — Squid — высокопроизводительный кэширующий proxy-сервер, поддерживающий протоколы FTP, Gopher и HTTP. Squid сохраняет часто запрашиваемые данные в оперативной памяти компьютера, что позволяет резко увеличить производительность proxy-сервера, кэширует DNS-запросы (это свойство интересно тем, кто не имеет своего DNS-сервера), а также поддерживает SSL, расширенный контроль доступа и полную регистрацию запросов.

Программа Squid описана ранее, однако мы позволим себе напомнить некоторые интересные моменты по работе с ней.

Transparent proxy

Transparent proxy — proxy-сервер, настроенный так, что его использование прозрачно для пользователей. Это имеет как хорошую, так и плохую стороны. С одной стороны, пользователям не придется настраивать соединение через proxy-сервер в своей системе, а трафик гарантированно проходит через proxy-сервер. С другой стороны, теряется свобода выбора — пользоваться или нет proxy-сервером. Кроме того, некоторые сайты некорректно обрабатываются proxy.

Для организации transparent proxy необходимо так настроить маршрутизатор (брандмауэр), чтобы транзитные пакеты, предназначенные для порта 80, попадали на вход proxy-сервера.

Борьба с баннерами

Наверняка вам встречались Web-страницы, на которых рекламных баннеров было больше, чем нужной информации. В этом случае можно настроить локальный сервер Squid так, чтобы баннеры не закачивались. Бороться с баннерами можно разными методами:

- настроить отдельный proxy-сервер с ограничением баннеров: хочешь — используй, не хочешь — откажись;
- совместить ограничение баннеров с transparent proxy;
- организовать proxy на локальной системе для ограничения баннеров.

Разделение внешнего канала (ограничение трафика)

Часто бывает так, что у вас есть внешний канал, скажем, 1024 Кбит, и несколько групп пользователей с определенным приоритетом.

И нужно, чтобы одна группа имела фиксированную ширину наружного канала (скажем, 512 Кбит), а две другие — ширину наружного канала по 256 Кбит. Для решения этой непростой задачи мы также можем воспользоваться Squid. Соответствующие настройки программы Squid для разделения внешнего канала приведены в *главе 23*.

Помимо Squid, для этого можно воспользоваться специализированными программами, называемыми traffic shaper. Существует несколько программ такого типа с различной функциональностью. В частности есть traffic shaper, которая позволяет ограничить канал не по пропускной способности, а по полученным мегабайтам. Принцип действия ее оригинален и прост. Допустим, у вас выделенный канал, причем в арендную плату входит один гигабайт входящего трафика. В программе traffic shaper выставляется ограничение один гигабайт в месяц. Далее происходит следующее. Сначала информация качается с той скоростью, с какой способен передавать информацию выделенный канал, но при приближении к "заветной" цифре пропускная способность канала, ограниченного traffic shaper, все уменьшается и уменьшается, не позволяя вам выйти за рамки ограничения в один гигабайт в месяц. В результате, в последние дни месяца скорость канала может упасть до десятков байтов в секунду.

В качестве стабильной и хорошо конфигурируемой программы типа traffic shaper можно порекомендовать пакет CBQ. Ограничивать трафик можно и с помощью утилиты tc, входящей в пакет iproute2.

Мониторинг загрузки каналов

Для анализа загрузки интернет-канала необходим дополнительный пакет, поскольку разбираться самим в log-файлах системы — задача неблагодарная. Чтобы обеспечить требуемую наглядность, такой пакет должен выдавать информацию в графической форме, причем, желательно, с помощью Web-интерфейса. Все эти условия реализованы в программах MRTG (Multi Router Traffic Grapher) и RRDtool (Round Robin Database).

Программа MRTG

MRTG создает HTML-страницу с отображением загрузки канала за сутки, неделю, месяц и год. Для этого используется скрипт на Perl, опрашивающий маршрутизатор через SNMP, а программа, написанная на C, обрабатывает получившийся результат и создает встроенные в HTML-страницу изображения в формате GIF/PNG. Помимо самостоятельно собранных сведений пакет MRTG может обрабатывать информацию и из других источников (cpuinfo, df, squid и т. п.) и строить графики по полученным данным.

Большое преимущество MRTG — постоянный размер журналов, в которых более старая информация хранится с меньшими подробностями.

Внешний вид получаемых графиков приведен на рис. 33.1.

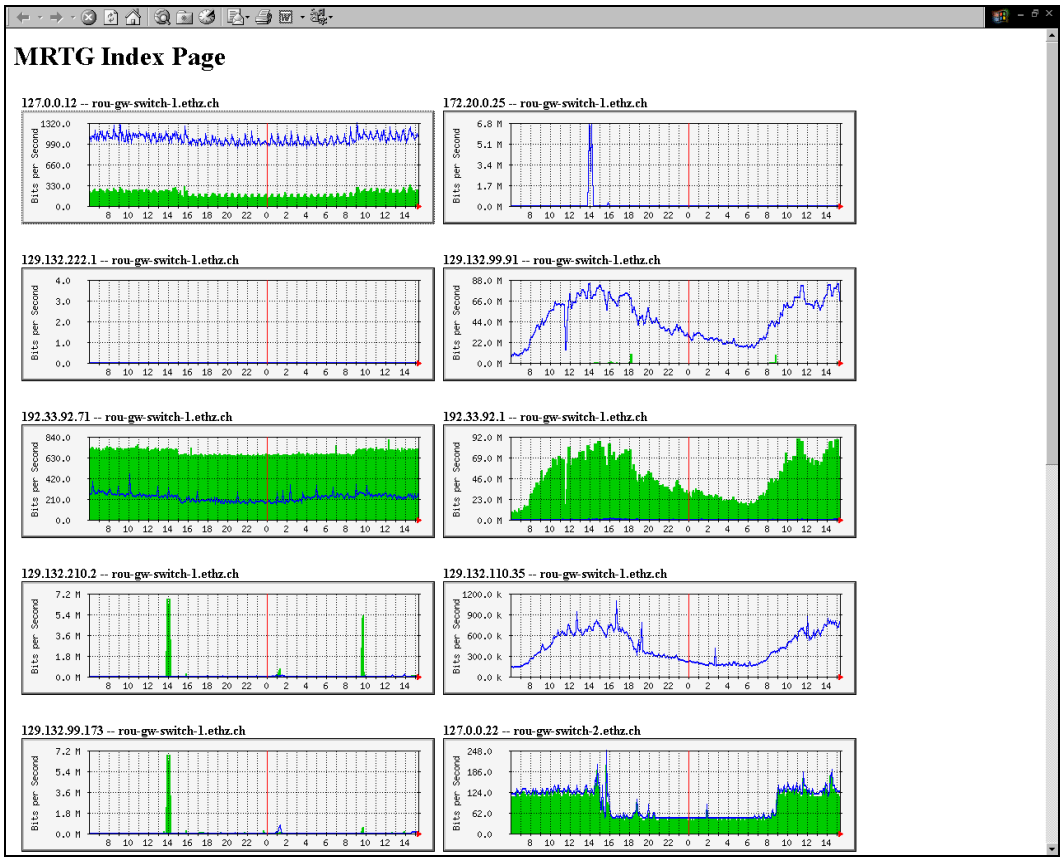


Рис. 33.1. Результат работы программы MRTG

Конфигурирование MRTG

Для конфигурирования программы MRTG используется файл mrtg.cfg, параметры которого и будут рассматриваться в данном разделе. Как обычно, будут приведены только ключевые параметры, с полным списком можно ознакомиться в документации, прилагаемой к этому программному пакету.

Правила записи параметров в конфигурационном файле:

- ключевое слово — в начале строки;
- двоеточие — разделитель, идущий сразу за ключевым словом;
- строка продолжения — начинается с пробела;
- строки комментария — начинаются с символа `#`.

Итак, файл mrtg.cfg может содержать следующие команды:

- `Include: имя_файла` — подключаемый файл;
- `WorkDir: имя_каталога` — задает размещение журнала, рабочих файлов и генерируемых страниц, имеет приоритет над `HtmlDir`, `ImageDir` и `LogDir`;
- `HtmlDir: имя_каталога` — размещение генерируемых страниц;

- `ImageDir: имя_каталога` — размещение генерируемых изображений; обязательно находится под `HtmlDir` — страницы генерируются в этом предположении;
- `LogDir: имя_каталога` — задает размещение журнала;
- `Refresh:` — частота перерисовки графиков в браузере;
- `RunAsDaemon: no | yes` — запуск MRTG в режиме демона;
- `Interval:` — предполагаемый интервал запуска MRTG;
- `IconDir:` — каталог, где хранятся значки;
- `Forks: число` — определяет, сколько параллельных процессов опроса запускать;
- `WriteExpire: no | yes` — создавать файлы .meta для apache;
- `NoMib2: no | yes` — не запрашивать `sysUptime`, `sysName`;
- `Language: язык_отчетов` — определяет язык отчетов (есть поддержка русского языка);
- `LogFormat: rrdtool` — формат журналов для rrdtool — динамическое создание отчетов;
- `LibAdd: адрес-библиотеки-rrdtool RRDs.pm` — адрес библиотеки rrdtool;
- `PathAdd: адрес-rrdtool` — адрес rrdtool.

Для каждого контролируемого устройства (обозначается как target) буквы преобразуются к строчным, создается отдельная секция. При работе MRTG каждый target порождает файлы журнала (target.log и target.old), картинки с графиками (target-day.gif, target-week.gif, target-month.gif, target-year.gif) и HTML-страницу (target.html).

- `Target[target]: порт:community@маршрутизатор [:port[:timeout[:retries[:backoff[:2]]]]]`

  где:
  - `порт` — номер интерфейса на маршрутизаторе;
  - `community` — пароль на чтение;
  - `маршрутизатор` — имя или IP-адрес;
  - `port` — по умолчанию стандартный порт SNMP;
  - `timeout` — время ожидания;
  - `retries` — число попыток;
  - `backoff` — во сколько раз увеличивать timeout при каждом повторе;
  - `2` — означает использование 64-битовых счетчиков;
- `Target[target]: внешняя-программа-с-параметрами-в-обратных-кавычках`

  Программа должна возвращать на стандартный вывод четыре строки:
  - значение счетчика входных байтов;
  - значение счетчика выходных байтов;
  - текстовую строку, содержащую информацию о времени работы объекта после включения;
  - строку, указывающую имя объекта;
- `RouterUptime[target]: community@маршрутизатор` — откуда брать информацию об имени маршрутизатора и его времени работы для составных target;
- `MaxBytes[target]:число` — значения переменных, которые больше этого числа, игнорируются;

- `Title[target]:` — заголовок для HTML-страницы;
- `PageTop[target]:` — текст, выдаваемый в верхней части HTML-страницы;
- `PageFoot[target]:` — текст, выдаваемый в нижней части HTML-страницы;
- `AddHead[target]:` — HTML-текст, вставляемый после `TITLE` внутри `HEAD`;
- `MaxAbs[target]: число` — при наличии сжатия возвращаемое значение может превосходить `MaxByte`;
- `Unscaled[target]: [d][w][m][y]` — подавить масштабирование по вертикали для соответствующего графика (`d` — день, `w` — неделя, `m` — месяц, `y` — год);
- `WithPeak[target]: [w][m][y]` — показывать в недельном, месячном и годовом графиках не только средние, но и пиковые значения (`w` — неделя, `m` — месяц, `y` — год);
- `Supress[target]: [d][w][m][y]` — подавить генерацию части графиков (`d` — день, `w` — неделя, `m` — месяц, `y` — год);
- `Directory[target]: имя-каталога` — размещать в данном каталоге все файлы, относящиеся к указанному `target`;
- `XSize[target]: число` — число пикселов в графике по горизонтали;
- `YSize[target]: число` — число пикселов в графике по вертикали;
- `YTics[target]: число` — число вертикальных делений;
- `Step[target]: секунд` — определяет шаг отображения в секундах;
- `Options[target]: список-опций-через-запятую`:
  - `growright` — время движется вправо;
  - `bits` — все числа умножать на 8 (измерять в битах);
  - `perminute` — все числа умножать на 60 (измерять в единицах за минуту);
  - `perhour` — все числа умножать на 3600 (измерять в единицах за час);
  - `transparent` — генерировать прозрачный фон картинки;
  - `gauge` — интерпретировать полученные данные в виде абсолютных значений. Полезно для отображения таких параметров, как загрузка процессора, занятый объем дискового пространства;
  - `unknaszero` — трактовать неверные значения как 0, а не как повторение предыдущего значения;
- `kilo[target]: число` — что понимается под `kilo`. По умолчанию — 1000, можно установить 1024;
- `kMG[target]: список-префиксов-множителей` — какими буквами обозначать kilo, mega и т. п. По умолчанию: "K, M, G, T, P";
- `Colours[target]`:
  `Colouri#RRGGBB,Colouri#RRGGBB,Colouri#RRGGBB,Colouri#RRGGBB` — определение цветовой схемы, где `Colour` — текстовое имя цвета, помещаемое в легенду графика, `i` = 1, 2, 3, 4 — номера цвета, `RRGGBB` — шестнадцатеричные значения, определяющие RGB-цвет;
- `Background[target]: #RRGGBB` — задает цвет фона;
- `YLegend[target]: текстовая-строка` — по умолчанию: "Bits per second";
- `ShortLegend[target]: текстовая-строка` — по умолчанию: "b/s".

Помимо MRTG, существует еще один пакет аналогичного назначения — RRDtool.

Программа RRDtool (Round Robin Database)

Этот программный пакет хранит и отображает результаты мониторинга: загрузку каналов, температуру и любую другую последовательность данных, зависящую от времени. Программа задумывалась как более правильная реализация MRTG. Объем хранимых данных не увеличивается со временем, т. к. ячейки хранения используются циклически. В отличие от MRTG, программа не упаковывает старые данные самостоятельно, сбор информации и генерация HTML-кода также производятся с помощью внешних средств. Параметры передаются в командной строке или через утилиту stdin.

Подсчет трафика

Иногда необходимо подсчитать трафик по клиентам, особенно когда организуется подключение домовой локальной сети или несколько небольших фирм совместно покупают выделенную линию для подключения к провайдеру. К сожалению, стопроцентного совпадения вычисленного трафика с данными провайдера добиться вряд ли удастся, поскольку приведенные далее способы подсчета дают *разные* результаты. Правда, погрешность обычно не превышает 5%.

Есть несколько вариантов подсчета трафика:

- по данным, взятым из SNMP (OutOctets на интерфейсе);
- по данным из Cisco;
- по данным из /proc/tty/driver/serial;
- по данным из radacct (radius-accounting/ OutOctets);
- по ipchains;
- с помощью nacctd.

Рассмотрим довольно простой способ подсчета трафика с использованием ipchains.

Смысл метода такой — ставим разрешительную цепочку для интересующего нас IP-адреса, например:

```
ipchains -A output -d AA.BB.CC.DD -j ACCEPT
```

Теперь можно посчитать байты (листинг 33.5).

Листинг 33.5

```
ipchains -L -v
Chain input (policy ACCEPT: 4195746 packets, 1765818402 bytes):
Chain forward (policy ACCEPT: 142999 packets, 29941516 bytes):
Chain output (policy ACCEPT: 4182597 packets, 1309541595 bytes):
pkts bytes target prot opt tosa tosx  ifname mark outsize source destination
ports
4  308  ACCEPT all  --  0xFF 0x00  any    anywhere  AA.BB.CC.DD  n/a
```

Из примера видно, что клиенту ушло 308 байтов. Со временем в столбике `bytes` будет накапливаться статистика по байтам. Далее необходимо как-то обрабатывать

эти данные и выводить себе и клиенту. Для этого можно воспользоваться программой на Perl, расположенной по адресу **linux.uatel.net.ua/ipcount.perl**.

Существует также пакет, предназначенный для подсчета IP-трафика через протокол SNMP. Он так и называется — "Универсальный счетчик IP-трафика через SNMP". Адрес пакета приведен в списке ссылок.

Помимо этих двух простых способов, есть еще много программ для подсчета трафика, в частности IpTraf, useripacct, netacct, ipacct.

Ссылки

- **www.linux.org.ru/books/gateway/** — Костарев А. Ф. ОС Linux как мост между локальной сетью и Internet.
- **lin-omts.airport.sakhalin.ru/departs/ccito/guide1.htm** — как установить, настроить и запустить Web-узел UNIX, не тратя лишних денег, сил и здоровья.
- **people.ee.ethz.ch/~oetiker/webtools/mrtg/mrtg.html** — описание MRTG.
- **www.mrtg.org** — официальный сайт пакета MRTG.
- **rrdtool.eu.org** — официальный сайт пакета rrdtool.
- **www.geocities.com/SiliconValley/Pines/7895/PPP.DOC** — В. Водолазкий. Установка PPP-соединения в Linux.
- **http://linux.perm.ru/doc/net/mrtg.html** — С. Богомолов. Мониторинг загрузки каналов (и не только) MRTG.
- **www.bog.pp.ru/work/rrdtool.html** — С. Богомолов. RRDtool — хранение и отображение данных мониторинга.
- **linux.uatel.net.ua/ipcount.phtml** — как оперативно подсчитать IP-трафик клиента.
- **ftp://ftp.kiev.farlep.net/pub/os/linux/soft/trafficcounter-snmp** — универсальный счетчик IP-трафика через SNMP.
- **http://www.tux.in.ua/articles/1098** — RRDtool: удобный инструмент мониторинга сети.
- Соответствующие HOWTO:
  - ISP-Hookup-HOWTO;
  - FIREWALLING_AND_PROXY_SERVER_HOWTO;
  - THE_LINUX_KERNEL_HOWTO.

Глава 34



Настройка модемного соединения

В этой главе мы рассмотрим настройку модемного соединения. Казалось бы, модемы сейчас уже практически не применяются. Тем не менее иногда приходится ими пользоваться. А массовое появление 3G-модемов опять вернуло актуальность данной теме. Начнем с теории.

Протокол PPP

Последние несколько лет протокол PPP стал стандартом де-факто для организации соединения по коммутируемым каналам и выделенным линиям. Поэтому для организации нормальной работы модемного соединения и извлечения из него максимальной пользы необходимо иметь понятие об этом протоколе. Итак, PPP — интернет-стандарт по передаче IP-пакетов по последовательным линиям — поддерживает синхронный и асинхронный режимы.

Общая информация

Point-to-Point Protocol (PPP, протокол "точка-точка") разработан для инкапсуляции протоколов вида "point-to-point IP". При создании протокола PPP стремились упростить выдачу и управление IP-адресами, асинхронную и синхронную инкапсуляцию, смешивание сетевых протоколов (Network Protocol Multiplexing), конфигурирование и тестирование качества связи, обнаружение ошибок, а также опции настройки адресов и сжатия данных. Для поддержки перечисленных качеств PPP должен предоставлять управление по расширенному протоколу Link Control Protocol (LCP, протокол управления соединением) и семейству протоколов Network Control Protocols (NCPs, протоколы управления сетью), которые используются для установления параметров связи. На сегодняшний день PPP поддерживает не только IP, но и другие протоколы, включая IPX и DECNet.

Свойства протокола PPP

В табл. 34.1 приведены основные возможности протокола PPP. Однако следует учитывать, что программное обеспечение может не в полной мере воплощать эти функции, а зачастую и привносит что-то свое, поэтому прежде чем пытаться реализовать то или иное свойство, заявленное в стандарте протокола PPP, рекомендуется

предварительно ознакомиться с описанием используемых программ, особенно в гетерогенной среде.

Таблица 34.1. *Основные возможности, реализуемые протоколом PPP*

Свойство	Описание
Demand on dial (дозвон по запросу)	Подключение PPP-интерфейса и набор телефонного номера по приходу пакета. Отключение интерфейса PPP после некоторого периода отсутствия активности
Redial	Подключение PPP-интерфейса, который потом не отключается и будет всегда сохранять в своем распоряжении подключенный канал
Campling	См. Redial
Scripting	Настройки через серию сообщений или промежуточных соединений для установления PPP-соединения — больше похоже на последовательности, используемые для установления связи по UUCP
Parallel	Конфигурирование нескольких PPP-линий для одного и того же подключения к хосту в целях равномерного разделения трафика между ними (в процессе стандартизации)
Filtering	Выбор, при каких пакетах имеет смысл начинать прозвон по линии, а при каких нет, отталкиваясь в принятии решения от IP- или TCP-типа пакета или TOS (Type of Service). К примеру, игнорировать все ICMP-пакеты
Header Compression (сжатие заголовка)	Сжатие TCP-заголовка в соответствии с RFC1144
Server	Прием входящих PPP-соединений, которые могут также требовать дополнительной маршрутизации
Tunneling	Построение виртуальных сетей по PPP-соединению, через TCP-поток и существующую IP-сеть. (Build a virtual network over a PPP link across a TCP stream through an existing IP network.)
Extra escaping	Байт-ориентированные символы, не входящие в стандартный набор символов, используемый при установлении связи, они могут быть сконфигурированы отдельно, но также не пересекаться с теми, что заданы при установлении связи

Как видите, возможности протокола богатые, и не удивительно, что некоторые из них не реализованы в полной мере.

Составляющие PPP

PPP предоставляет возможность передачи датаграмм по последовательным point-to-point-линиям и имеет три составляющие:

- метод предоставления инкапсуляции датаграмм по последовательным PPP-линиям с использованием HDLC (High-Level Data Link Control, высокоуровневого управления данными соединения) — протокол для упаковки датаграмм по PPP средствами связи;

- расширенный протокол LCP для установления, конфигурирования и тестирования физического соединения;
- семейство протоколов NCP для установки и управления другими сетевыми протоколами, что позволяет протоколу PPP поддерживать одновременно несколько сетевых протоколов.

Функционирование протокола PPP

В момент установления связи через PPP-соединение PPP-демон вначале шлет пакеты LCP для конфигурирования и тестирования линии связи. После того как связь и дополнительные возможности будут установлены посредством протокола LCP, PPP-демон посылает NCP-фреймы для изменения и настройки одного или более сетевых протоколов. По окончании процесса настройки сетевые пакеты могут передаваться через установленное соединение. Оно будет оставаться активным, пока специальные LCP- или NCP-пакеты не закроют соединение или пока не произойдет какое-нибудь внешнее событие, которое приведет к потере соединения, например сработает таймер отсутствия активности или разорвется модемное соединение.

Поддерживаемое оборудование

Протокол PPP адаптирован для работы с любым интерфейсом DTE/DCE, включая RS-232, RS-422, RS-423, CITT V.35. Помимо этих интерфейсов протокол может работать практически на любом оборудовании, единственное требование — наличие дуплексного режима.

Структура пакета протокола PPP

Принципы, терминология и структура пакетов протокола PPP описаны в стандартах ISO, касающихся HDLC:

- ISO 3309-1984/PDAD1 "Addendum 1: Start/stop transmission" (приложение 1 начало/конец передачи);
- ISO 3309-1979 — описывает структуру пакетов HDLC для синхронных систем;
- ISO 3309:1984/PDAD1 — описывает предложения по изменениям в ISO 3309-1979 для асинхронных систем.

На рис. 34.1 изображен формат пакета протокола PPP.

Величина поля, байты	1	1	1	2	Переменный	2 или 4
Назначение	Флаг	Адрес	Управление	Протокол	Данные	Контрольная сумма

Рис. 34.1. Структура пакета протокола PPP

Рассмотрим значения полей пакета протокола PPP:

- `Флаг` — один байт, обозначающий начало или конец пакета. Поле флага содержит двоичную последовательность 01111110;

- `Адрес` — один байт, содержащий двоичную последовательность 11111111, стандартный широковещательный адрес. PPP не поддерживает индивидуальную адресацию станций;
- `Управление` — один байт, содержащий двоичную последовательность 00000011, который посылается для передачи пользовательских данных в неразделенных пакетах;
- `Протокол` — два байта определяют протокол, упакованный в пакете протокола PPP. Значения протоколов можно узнать в соответствующем RFC;
- `Данные` — ноль или больше байтов, составляющих датаграмму протокола, указанного в поле `Протокол`. Конец информационного поля определяется нахождением заканчивающей последовательности и двухбайтовой последовательности в поле контрольной суммы. По умолчанию максимальная длина поля данных 1500 байтов. Однако во время установления сеанса программы pppd могут договориться выбрать другое значение поля данных;
- `Контрольная сумма` — обычно 16 бит. Однако при установлении соединения pppd могут договориться об использовании 32-битовой контрольной суммы.

PPP-протокол управления соединением (LCP)

PPP-протокол управления соединением (LCP) предоставляет методы для установления, конфигурирования, поддержания и тестирования PPP-соединения. Функции протокола LCP:

- Конфигурирование и установление связи. Перед передачей какой-либо информации (к примеру, пакета IP) протокол LCP должен открыть соединение и провести начальный обмен параметрами настройки. Этот этап заканчивается, когда пакет о подтверждении произведенной настройки будет послан и принят обратно.
- Определение качества связи. Протокол LCP позволяет (но этой возможностью зачастую пренебрегают) добавить фазу тестирования канала связи. Тестирование канала связи должно происходить сразу же за конфигурированием и установлением связи. Во время проверки качества связи определяется, способно ли соединение с достаточным качеством транспортировать какой-либо сетевой протокол.
- Установление настроек сетевого протокола. После того как протокол LCP закончит определение параметров связи, сетевые протоколы должны быть независимо друг от друга настроены соответствующими протоколами NCP, которыми могут в любой момент времени начать или прекратить пользоваться.
- Окончание связи. Протокол LCP может в любое время прервать установленную связь. Это может произойти по требованию пользователя или из-за какого-нибудь события, к примеру, потери несущей или истечению допустимого периода времени неиспользования канала.

Существуют три типа LCP-пакетов:

- пакеты установления — для установления и настройки связи;
- пакеты прерывания — для прерывания установленной связи;
- пакеты сохранения связи — для управления и диагностики связи.

Сокращения, принятые при описании протокола PPP

В табл. 34.2 приведены некоторые аббревиатуры, используемые при описании протокола PPP. Расшифровка содержит как английское значение, так и русский перевод.

Таблица 34.2. *Аббревиатуры, используемые при описании протокола PPP*

Аббревиатура	Расшифровка
ack	Acknowledgement — подтверждение о получении данных
AO	Active Open [state diagram] — соединение активно
C	Close [state diagram] — соединение закрыто
CHAP	Challenge-Handshake Authentication Protocol (RFC1334) — протокол аутентификации
D	Lower layer down [state diagram] — нижний уровень отсутствует
DES	Data Encryption Protocol — протокол шифрования данных
DNA	Digital Network Architecture — архитектура цифровых сетей
IETF	Internet Engineering Task Force — организация, непосредственно отвечающая за разработку протоколов и архитектуры сети Интернет
FCS	Frame Check Sequence [X.25] — проверочная последовательность кадра
LCP	Link Control Protocol — протокол управления соединением
LQR	Link Quality Report — отчет о качестве соединения
MD4	MD4 digital signature algorithm — протокол цифровой подписи
MD5	MD5 digital signature algorithm — протокол цифровой подписи
MRU	Maximum Receive Unit — максимальная величина принимаемого кадра
MTU	Maximum Transmission Unit — максимальная величина передаваемого кадра
NAK	Negative Acknowledgement — негативный ответ
NCP	Network Control Protocol — протокол сетевого управления
PAP	Password Authentication Protocol (RFC1334) — протокол аутентификации
PDU	Protocol Data Unit — пакет
PO	Passive open — пассивное соединение
PPP	Point to Point Protocol — протокол "точка-точка"
RCA	Receive Configure-Ack — принят конфигурационный запрос
RCJ	Receive Code-Reject — принят код отклонения
RCN	Receive Configure-Nak or -Reject — принят код отклонения

Таблица 34.2 *(окончание)*

Аббревиатура	Расшифровка
RCR+	Receive good Configure-Request [state diagram] — принят нормальный запрос конфигурации
RER	Receive Echo-Request — принят эхо-запрос
RTA	Receive Terminate-Ack [state diagram] — принят запрос на разрыв соединения
RUC	Receive unknown code [state diagram] — принят неизвестный код
SCA	Send Configure-Ack [state diagram] — послан конфигурационный запрос
SCJ	Send Code-Reject [state diagram] — послан код отклонения
SCN	Send Configure-Nak or -Reject [state diagram] — послан код отклонения
ST-II	Stream Protocol — потоковый протокол
TO+	Timeout with counter > 0 [state diagram] — счетчик тайм-аута больше чем ноль
TO–	Timeout with counter expired [state diagram] — счетчик тайм-аута превысил предел
VJ	Van Jacobson (RFC1144 header compression algorithm) — алгоритм компрессии заголовка пакетов PPP
XNS	Xerox Network Services — сетевые службы Xerox

Стандарты, описывающие протокол PPP

В табл. 34.3 приведены стандарты (RFC) протокола PPP.

Таблица 34.3. *Стандарты протокола PPP*

Номер RFC	Название
1144	Compressing TCP/IP headers for low-speed serial links — Сжатие заголовков пакетов для низкоскоростных последовательных соединений
1220	Point-to-Point Protocol extensions for bridging — Расширение протокола PPP
1332	PPP Internet Protocol Control Protocol (IPCP) — Управляющий протокол IP
1333	PPP link quality monitoring — Контроль качества соединения PPP
1334	PPP authentication protocols — Протоколы аутентификации PPP
1547	Requirements for an Internet Standard Point-to-Point Protocol — Требования для интернет-стандарта PPP
1552	The PPP Internetwork Packet Exchange Control Protocol (IPXCP) — Управляющий протокол обмена пакетами для разнородных сетей
1570	PPP LCP Extensions — Расширения протокола LCP
1598	PPP in X.25 — Использование протокола PPP в сетях X.25

Таблица 34.3 *(окончание)*

Номер RFC	Название
1618	PPP over ISDN — Использование протокола PPP поверх протокола ISDN
1619	PPP over SONET/SDH — Использование протокола PPP поверх протокола SONET/SDH
1638	PPP Bridging Control Protocol (BCP) — Протокол управления PPP
1661	The Point-to-Point Protocol (PPP) — Протокол "точка-точка"
1662	PPP in HDLC-like Framing — PPP в HDLC-подобных кадрах
1663	PPP Reliable Transmission — Надежная передача PPP
1717	The PPP Multilink Protocol (MP) — Многопоточный PPP-протокол

Настройка сервера входящих звонков (dial-in)

В этом разделе мы перейдем к сугубо практическим действиям — настроим наш сервер для приема входящих звонков. Мы уже умеем настраивать систему так, чтобы она выступала в роли шлюза для вашей локальной сети. Но бывает нужно получить доступ к локальной сети организации, например из дома, или с домашнего компьютера выйти в Интернет через корпоративную локальную сеть. Нет ничего проще. Настраивать PPP-соединение мы уже научились, осталось только установить программу, которая будет "поднимать трубку" модема по входящему звонку и совершать некоторые дополнительные действия. Такая программа — mgetty — признанный фаворит в своей области, умеющий помимо всего прочего посылать и принимать факсы, а также с помощью голосового модема принимать и отправлять voice mail — голосовую почту.

Настройка mgetty

Обычно mgetty, как и ppp, входит в стандартную поставку дистрибутива. Единственное, что необходимо проверить, был ли пакет mgetty скомпилирован с опцией `-DAUTO_PPP`, и если нет, то пакет следует перекомпилировать с этой опцией.

После установки mgetty нам нужно отредактировать конфигурационные файлы. В файле /etc/mgetty+sendfax/login.config мы должны написать:

```
/AutoPPP/ – a_ppp /usr/sbin/pppd auth refuse-chap require-pap login
- – /bin/login @
```

Эта строка говорит mgetty следующее:

- после установления входного соединения необходимо вызвать программу pppd;
- для пользователя требуется авторизация;
- аутентификацию по протоколу CHAP отклонять и требовать авторизацию по протоколу PAP.

После установления соединения mgetty анализирует данные, приходящие с модема, и в случае, когда приходит запрос на авторизацию по протоколу PAP, программа сразу же запускает pppd, который и проводит аутентификацию по протоколу PAP.

Далее нам необходимо отредактировать файл /etc/mgetty+sendfax/mgetty.config так, как показано в листинге 34.1.

Листинг 34.1

```
port ttyS1
speed 115200
data-only y
debug 3
init-chat "" ATZ OK
     answer-chat "" ATA CONNECT \c \r
```

Как видите, модем подключен ко второму последовательному порту, скорость обмена 115 200, строка инициализации `ATZ`.

Далее нужно добавить mgetty в файл inittab. Для этого достаточно дописать лишь одну строку:

```
S4:2345:respawn:/sbin/mgetty /dev/ttyS1
```

Перезагрузив операционную систему, можно приступать к испытаниям: попробуйте позвонить на телефонный номер, где установлен ваш модем. Если все настроено нормально, модем должен "поднять трубку".

Настройка pppd

С настройкой pppd вы уже ознакомились в *главе 33*. Чтобы не повторяться, просто приведем соответствующие конфигурационные файлы с небольшими комментариями.

Файл options.tty должен содержать данные, приведенные в листинге 34.2.

Листинг 34.2

```
# Устройство
lock
login
auth
modem
crtscts
-chap
+pap
# наш интерфейс : удаленный интерфейс
192.168.10.100:192.168.10.101
# маска подсети
netmask 255.255.255.0
# адрес сервера DNS для клиента Windows
ms-dns 192.168.10.100
```

Файл /etc/ppp/pap-secrets должен содержать следующие данные:

```
user1  сервер.домен  ""      *
user2  сервер.домен  ""  *
```

где:

- `user1` — имя пользователя, причем он должен существовать в вашей системе, где установлен модем;
- `user2` — сервер, на котором будет проводиться аутентификация; в нашем случае вместо `сервер.домен` необходимо поставить имя компьютера, где расположен модем;
- `""` — отсутствие пароля указывает на то, что пароли необходимо брать из файла /etc/shadow;
- `*` — абонент может проводить аутентификацию с любого IP-адреса.

Вот и все — вы стали микропровайдером, причем пользователям Windows сильно облегчили жизнь, поскольку IP-адрес и адрес DNS-сервера вы выдаете автоматически, кроме того, отпадает потребность в использовании скрипта для соединения.

Настройка callback-сервера

Итак, вы настроили свой dial-in-сервер, попользовались им какое-то время и захотели чего-то другого. Например, в вашем городе повременная оплата и часами работать в Интернете из дома не получается, но руководство вашей организации не возражает против того, чтобы вы работали за ее счет. Дело за малым — организовать ваш dial-in-сервер так, чтобы не вы ему звонили, а он вам. В компьютерных документах такой сервер зовется callback-сервером. Функционирует он следующим образом.

Сначала клиент дозванивается через модем к callback-серверу. Модем на сервере настроен на прием входящих звонков (установку и настройку dial-in-сервера мы только что рассмотрели). После установки соединения сервер предлагает клиенту пройти аутентификацию. Клиент подключается к нему как особый callback-пользователь. После этого модем на сервере обрывает связь и сам звонит клиенту по номеру, который закреплен за компьютером клиента. Модем на клиентском компьютере готов принять обратный звонок, и после установления соединения происходит повторная авторизация. По окончании аутентификации устанавливается PPP-соединение. Далее клиент работает как обычно.

Конфигурация callback-сервера

После того как настройка dial-in-сервера завершена, необходимо настроить callback. Для этого нужно выполнить следующие действия:

1. Создать нового пользователя back.
2. Создать пустой файл с именем callback.conf в /etc/mgetty/.
3. В файл /etc/mgetty/login.config добавить строку

```
back – – /usr/sbin/callback -S 1234567
```

После ключа `-S` указывается номер, по которому сервер должен сделать обратный звонок клиенту.

Конфигурация клиентов

Поскольку сервер мы уже сконфигурировали, необходимо сконфигурировать клиента и проверить, как же работает callback. Начнем с операционной системы Linux.

Конфигурирование Linux-клиента

Для конфигурирования клиента Linux необходимо выполнить следующее:

1. Создать файл /etc/ppp/options (листинг 34.3).

Листинг 34.3

```
lock
defaultroute
noipdefault
modem
115200
crtscts
debug
passive
```

2. Создать файл ppp-callback в /etc/ppp/peers/, в котором должны быть такие строки:

```
ttyS1 33600 crtscts
connect '/usr/sbin/chat -v -f /etc/ppp/chat-callback'
noauth
```

3. Создать файл /etc/ppp/chat-callback (листинг 34.4).

Листинг 34.4

```
ABORT BUSY
ABORT VOICE
ABORT "NO DIALTONE"
ABORT "NO ANSWER"
"" ATZ
OK ATDP7654321            # Телефонный номер сервера
CONNECT \d\d
ogin: \q\dback
TIMEOUT 90
RING AT&C0S0=1
ogin: \q\dvasya
assword: \q\dpasswordforvasya
```

В файл chat-callback необходимо вписать телефон callback-сервера, имя и пароль пользователя.

4. Создать файл /usr/bin/pppcall, в котором должны быть такие строки:

```
#!/bin/bash
/usr/sbin/pppd -detach call ppp-callback &
```

Сделать этот файл исполняемым.

Теперь для того, чтобы позвонить на ваш сервер, достаточно запустить скрипт pppcall.

Конфигурирование клиента MS Windows

Для Windows конфигурация проводится по-другому. Создайте новое соединение. Укажите данные, необходимые для дозвона к серверу. Помимо этого, в настройках модема на вкладке **Подключения** нажмите кнопку **Дополнительно** и в строке инициализации модема укажите следующее:

```
&c0s0=1
```

Теперь пробуем дозвониться до нашего сервера. После дозвона в открывшемся окне терминала вы увидите приглашения для аутентификации.

Зарегистрируйтесь в системе как back. После этого модем со стороны сервера оборвет связь, подождет несколько секунд и перезвонит вам. После установки callback-соединения вам предложат пройти повторно авторизацию. Введите ваш нормальный логин и пароль и нажмите кнопку **Продолжить** в окне терминала. Все.

Настройка модемного соединения для пользователя

Настала пора приступить к настройке модемного соединения клиента. Но предварительно поговорим о модемах.

Модемы бывают трех классов.

- Наружный модем, подключаемый к последовательному порту (нормальный, аппаратный модем).
- Внутренний модем (нормальный, аппаратный модем).
- Win-модем (наружный модем, подключаемый к USB-порту, или внутренний модем с интерфейсом PCI).

С первыми двумя понятно — поставил, настроил, работай.

С Win-модемом все несколько сложнее. Идея этого модема заключается в том, чтобы упростить и удешевить конструкцию за счет возложения всей обработки сигнала после преобразования из аналогового вида в цифровой на процессор компьютера и драйвер модема. А Win-модемом такие устройства назвали потому, что драйверы первоначально были написаны только для Windows. Сказать, что Win-модем работает хорошо, особенно на наших телефонных линиях, нельзя. На нормальной телефонной линии и цифровой АТС Win-модем может устойчиво работать, правда скорости выше 44 000 бит/с вы никогда не получите, а реальная скорость будет в пределах 28 800–33 600 бит/с. Причем, по опыту работы, Win-модем на чипе от Lucent более послушный в настройке и несколько лучше себя ведет, чем модемы на чипе Conexant или Pctel.

Предположим, вы купили Win-модем и хотите его настроить под операционной системой Linux. Некоторое время назад это бы не удалось, поскольку производители модемов драйверы под Linux не выпускали, а спецификаций на модем сторонним разработчикам не давали. Но в последнее время индустрия разворачивается к Linux лицом: выпускают драйверы, некоторые даже с исходным кодом.

Первое, что следует сделать — найти на сайте производителя модема (или модемного чипа) драйвер под Linux. Сходите также по ссылкам, приведенным в конце главы, например на **www.linmodems.org**, — наверняка это вам поможет. Далее действуйте по инструкции, прилагаемой к драйверу.

Настройка модема в текстовом режиме

Все просто, идем по пунктам:

1. Создаем файл /etc/ppp/options (листинг 34.5).

Листинг 34.5

```
lock
defaultroute
noipdefault
modem
115200
crtscts
debug
```

2. Создаем файл ppp-call в /etc/ppp/peers/, в котором содержатся следующие строки:

```
ttyS1 115200 crtscts
connect '/usr/sbin/chat -v -f /etc/ppp/chat-call'
noauth
```

3. Создаем файл /etc/ppp/chat-call (листинг 34.6).

Листинг 34.6

```
ABORT BUSY
ABORT VOICE
ABORT "NO DIALTONE"
ABORT "NO ANSWER"
"" ATZ
OK ATDP7654321            # Телефонный номер провайдера
CONNECT \d\d
ogin: \q\dvasya
assword: \q\dpasswordforvasya
```

В файл chat-call необходимо вписать телефон дозвона провайдера, имя и пароль пользователя.

4. Создаем файл /usr/bin/pppcall, в котором содержатся строки:

```
#!/bin/bash
/usr/sbin/pppd -detach call ppp-call &
```

И делаем его исполняемым.

Теперь для того, чтобы позвонить на ваш сервер, достаточно запустить скрипт pppcall.

Настройка модема в X Window

Самый простой путь настройки модема — с помощью графических утилит. Во многих дистрибутивах в среде GNOME есть удобная и простая программа — сетевой менеджер. Ее значок находится в верхнем правом углу рабочего стола. Открываем его, выбираем нужный вид соединения, создаем новое подключение, вводим нужный логин и пароль. Все. Более подробно рассмотрим настройку на примере 3G-модема.

Настройка 3G-модема в X Window

3G-модем представляет собой небольшой брелок, подключаемый к компьютеру по USB-интерфейсу. Выпускают их около полутора десятков производителей, и наверное есть десяток чипсетов 3G-модемов. Тем не менее в стандартную поставку большинства дистрибутивов входят пакеты для поддержки таких модемов и практически все они нормально могут функционировать. 3G-модем можно настроить и руками, посредством конфигурационных файлов. Но намного проще воспользоваться сетевым менеджером, значок которого находится в правом верхнем углу рабочего стола (рис. 34.2).

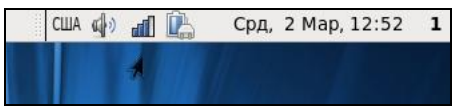


Рис. 34.2. Значок сетевого менеджера

Щелкаем на нем и получаем следующее окно (рис. 34.3), в котором можно увидеть все существующие сетевые соединения. Как видите, мобильных соединений пока нет. Подключаем наш модем и жмем кнопку **Добавить.**

Внизу окна есть список (рис. 34.4). Выбираем соответствующий модем и нажимаем **Вперед.**

Выбираем страну (рис. 34.5). В моем случае это Украина.

Далее выбираем из списка подходящего оператора (рис. 34.6). Благодаря монополизации операторов мало, и почти все они уже внесены в список создателями дистрибутива. Но если его нет в списке, нужно добавить самостоятельно.

Теперь необходимо задать точку доступа (рис. 34.7). Для стандартных тарифов она уже прописана сборщиком дистрибутива. Предварительно нужно узнать у оператора эти данные для вашего тарифного плана.

Нам покажут введенные настройки, которые можно при необходимости изменить (рис. 34.8).

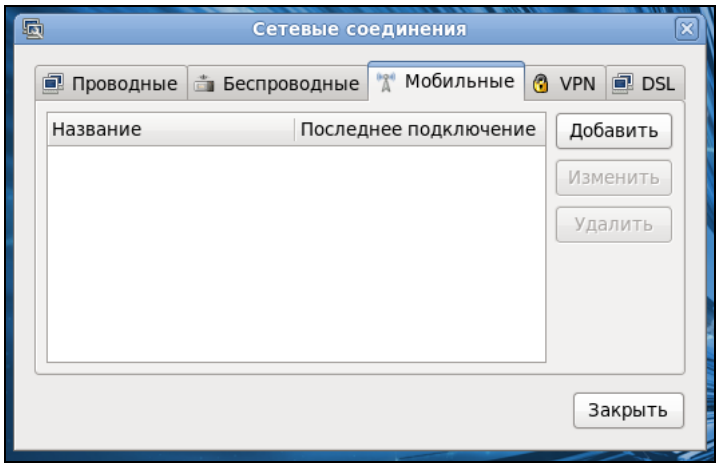


Рис. 34.3. Сетевой менеджер

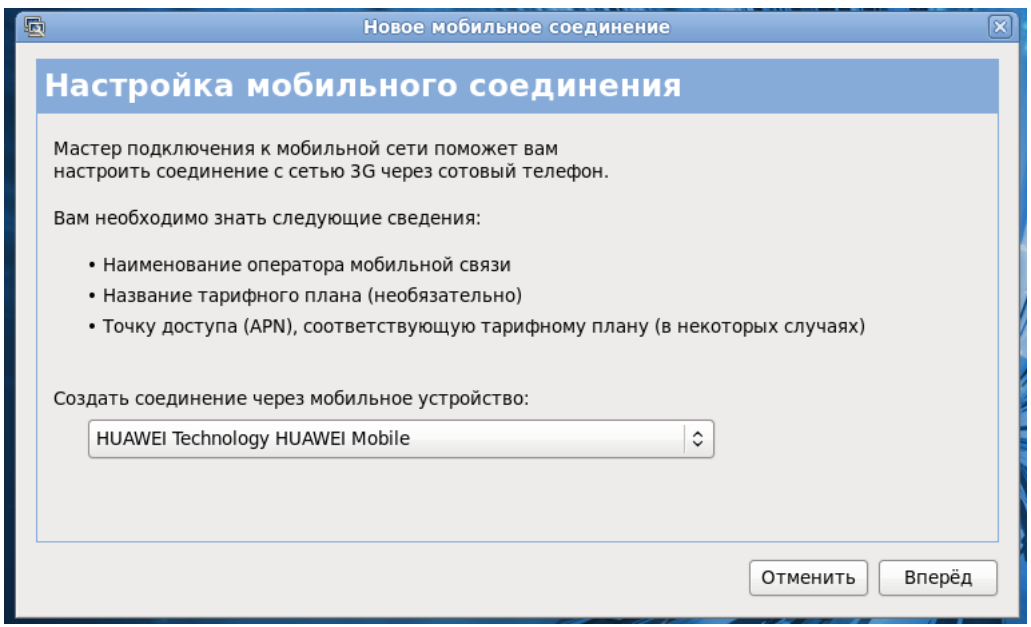


Рис. 34.4. Окно выбора модема

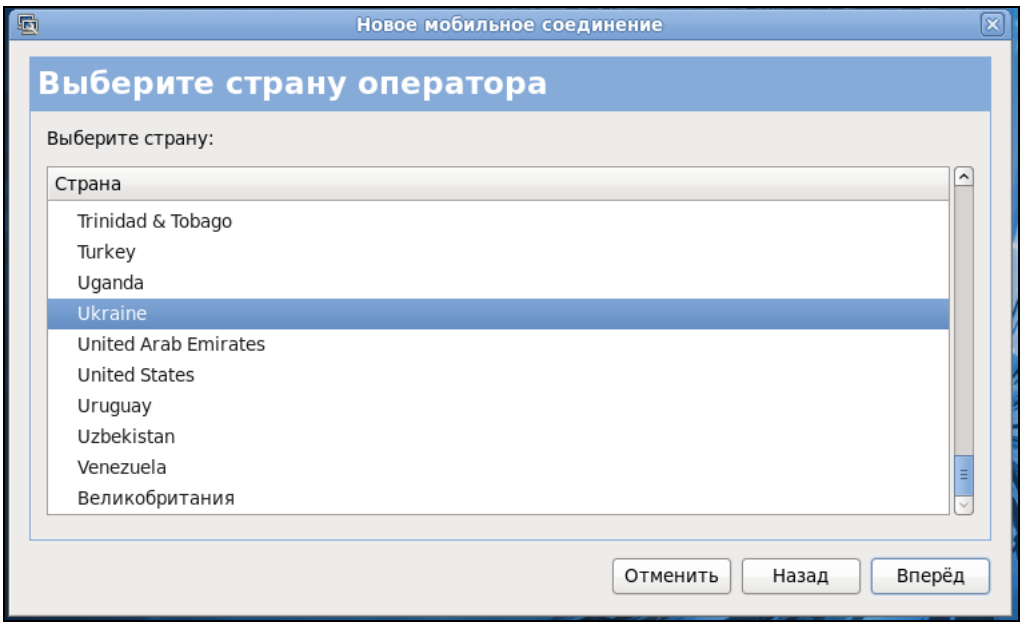


Рис. 34.5. Окно выбора страны

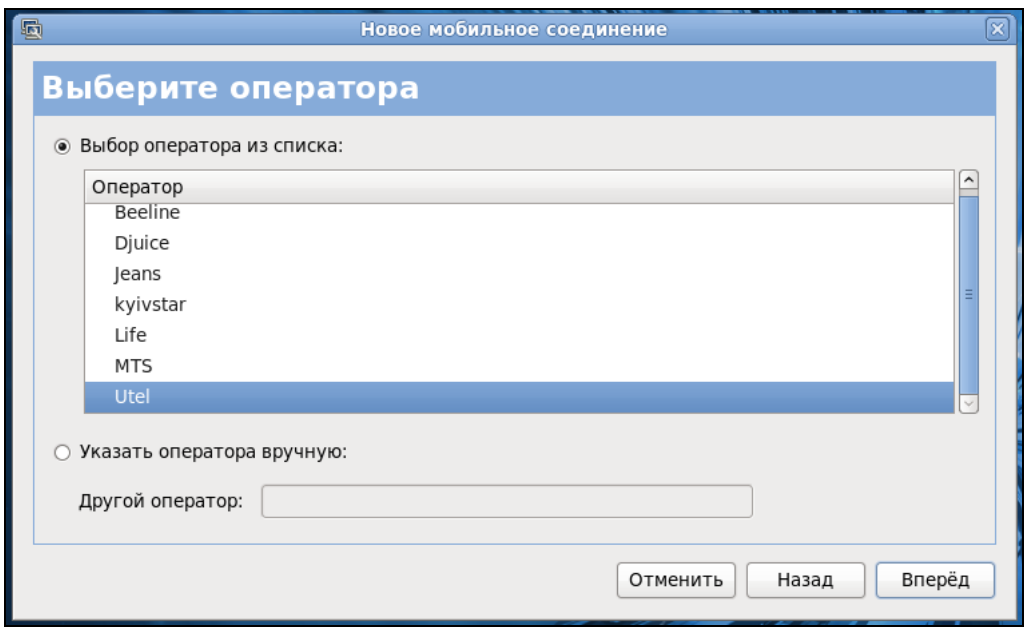


Рис. 34.6. Окно выбора оператора мобильной связи

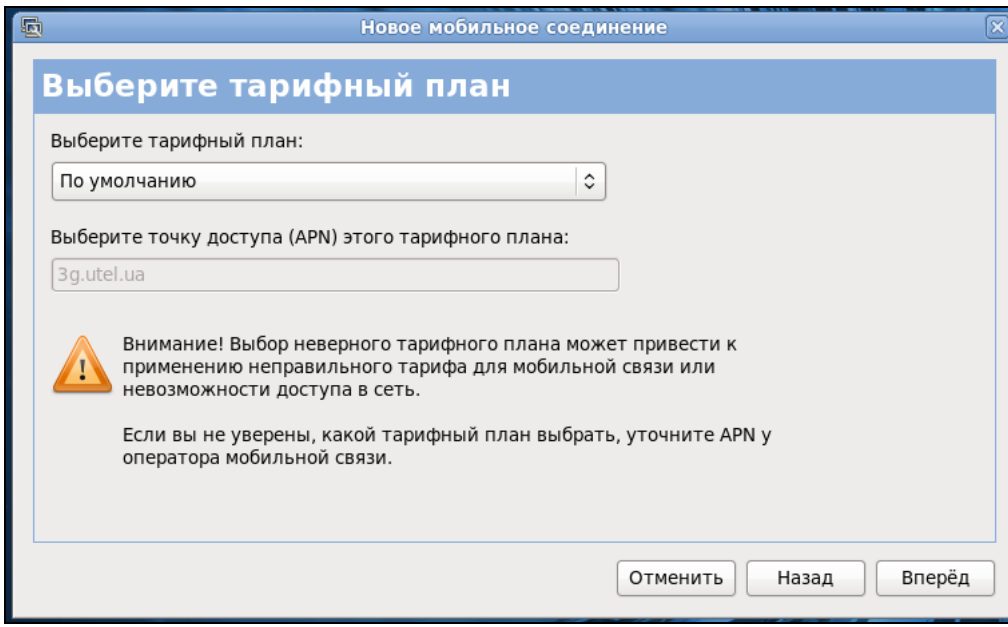


Рис. 34.7. Окно выбора точки доступа

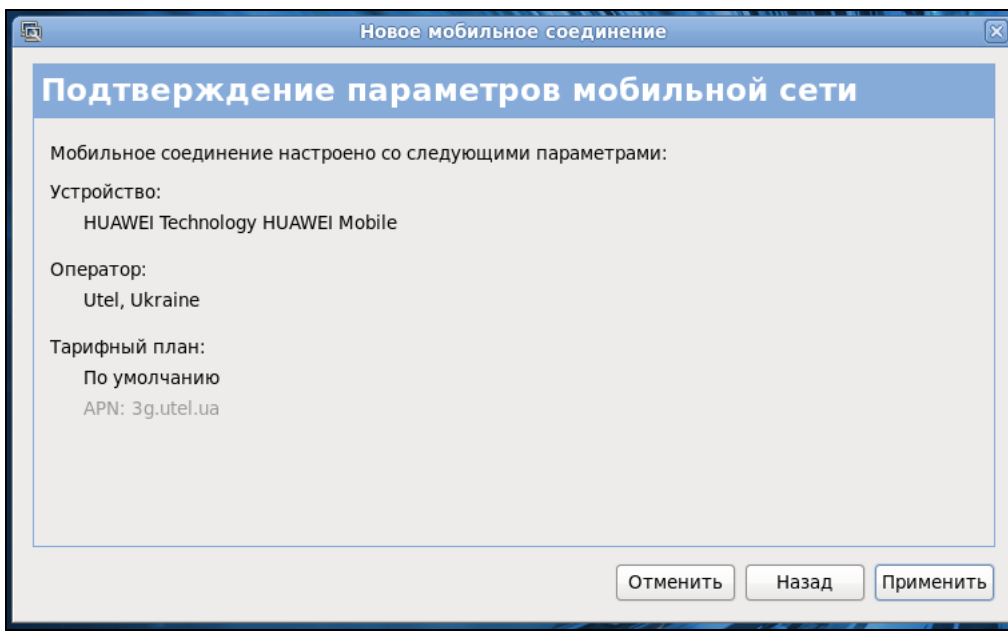


Рис. 34.8. Данные создаваемого соединения

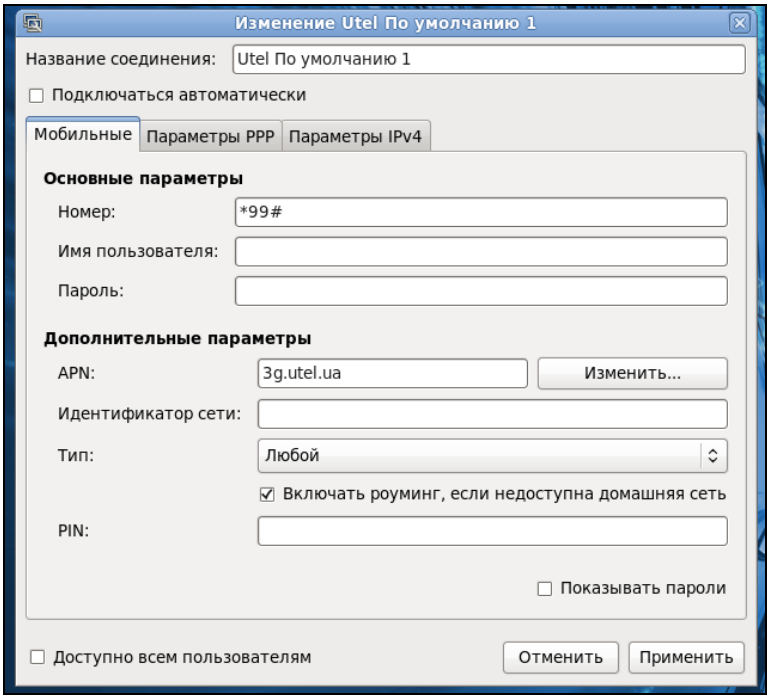


Рис. 34.9. Окно настройки параметров соединения

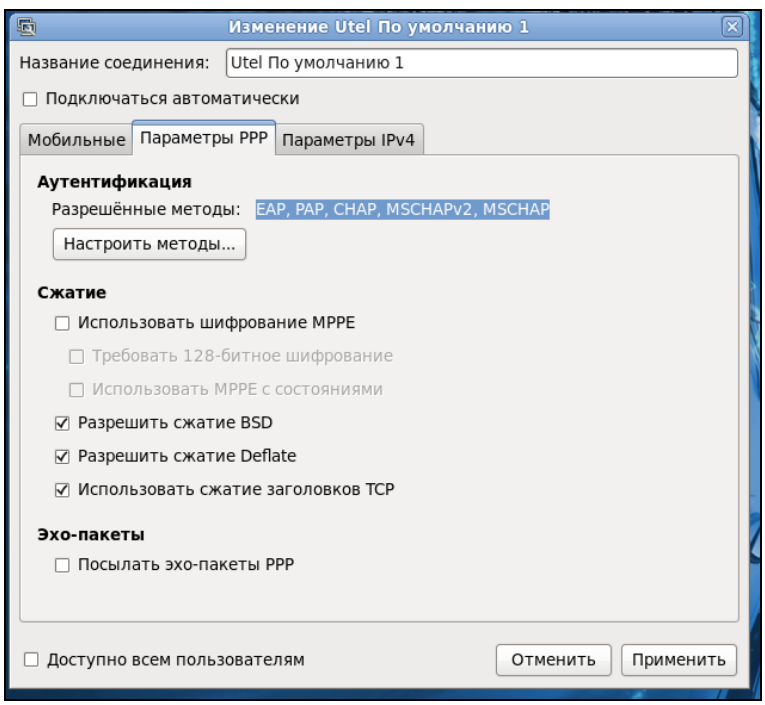


Рис. 34.10. Настройка параметров PPP

Теперь внимательно читаем инструкцию к модему и настройки, выдаваемые оператором. Здесь нужно заполнить номер дозвона, логин, пароль и дать название соединению (рис. 34.9). При необходимости на соответствующих вкладках следует задать параметры PPP и IP соединения (рис. 34.10).

Сохраняем настройки. Теперь достаточно выбрать и активировать созданное соединение в сетевом менеджере.

Кроме обычных 3G-модемов, бывают комбинированные устройства, содержащие в одном корпусе модем и Flasch-память или картридер. Для того чтобы система корректно обнаруживала модем, иногда его нужно принудительно переключить из режима флешки/картридера в режим модема. Можно воспользоваться программным обеспечением, поставляемым в комплекте с устройством или установить пакеты usb-modeswitch и usb-modeswitch-data. В документации на пакеты подробно написано, как переключать модем из режима Flasch-накопителя в режим модема.

Ссылки

- **cs.uni-bonn.de/ppp/part1.html, netware.nwsoft.ru** — John Wobus. Протокол PPP. Перевод Виталия Горохова.
- **www.linmodems.org** — драйверы и настройки для Win-модемов на базе чипов различных производителей — Lucent, Connexant (Rockwell), Pctel.
- **www.o2.net/~gromitkc/winmodem.html** — драйверы и настройки для Win-модемов на базе чипов различных производителей — Lucent, Connexant (Rockwell), Pctel.
- **www.idir.net/~gromitkc/winmodem.html** — драйверы и настройки для Win-модемов на базе чипов различных производителей — Lucent, Connexant (Rockwell), Pctel.
- **www.olitec.com/pci56kv2.html** — драйверы для Win-модемов на базе чипа Connexant (Rockwell).
- **www.heby.de/ltmodem/** — драйверы и настройки для Win-модемов на базе чипа Lucent.
- **www.sfu.ca/~cth/ltmodem/** — драйверы и настройки для Win-модемов на базе чипа Lucent.
- **linux.uatel.net.ua/ppp-dialin.phtml** — настройка PPP dial-in-сервера (PAP-аутентификация).
- **www.softerra.ru/freeos/12279/** — Денис Колисниченко. Пошаговая настройка dial-in-сервера.
- **www.linuxgazette.com** — Sunil Thomas Thonikuzhiyil. Настройка callback-сервера на базе Linux. Перевод Александра Куприна.
- **www.bdcol.ee/linux/callback.shtml** — Linux-callback.
- **www.leo.org/~doering/mgetty/** — документация по Mgetty+Sendfax.
- **http://koi.citforum.tula.ru/operating_systems/articles/ppp.shtml** — В. Водолазкий. Установка PPP-соединения в Linux.
- Документация к программе pppd.
- **linux.yaroslavl.ru/Howto/Howto-mini/call-back-mini-HOWTO.html** — Callback mini-HOWTO (русский перевод).
- PPP-HOWTO.



Глава 35

Резервное копирование и хранение данных

Резервное копирование выполняется для сохранения данных на случай их потери или разрушения. Подобные копии нужно создавать периодически, в соответствии с заранее установленным графиком. Схемы резервного копирования зависят от размеров и степени охвата резервированием операционной системы, а также от выдвигаемых требований к сохранению жизнеспособности системы. Элементы системы резервного копирования должны включать необходимое оборудование, носители резервных копий и специальное программное обеспечение. Оборудование может представлять собой достаточно широкий набор аппаратных средств, начиная от обычного дисковода и заканчивая библиотекой ленточных устройств. Тип и количество носителей определяются имеющимся оборудованием, объемами обрабатываемых данных и выбранной схемой резервирования данных. Программное обеспечение может быть очень разнородным, начиная от бесплатных утилит типа tar, cpio, gzip и заканчивая распределенными системами управления хранилищами данных.

Цели резервного копирования информации:

- восстановление файлов, случайно удаленных пользователями или утерянных из-за отказов дисковых устройств;
- получение периодически создаваемых моментальных снимков (snapshots) состояния данных организации. Эта информация может иметь большое техническое или коммерческое значение;
- получение данных для восстановления после аварий. Система резервного копирования обязательно является составной частью любого продуманного плана восстановления системы. В случае широкомасштабных катастроф данные доставляются из архивов, сохраняемых в отдельном помещении.

Планирование резервного копирования

При разработке системы резервного копирования очень важно правильно сформулировать исходные требования. Постарайтесь учесть все аспекты резервного копирования, сделайте два варианта сметы — минимально необходимый и желательный, и обоснуйте руководству организации необходимость дополнительного финансирования, связанного с резервным копированием данных, критичных для существования организации. Попытайтесь реализовать хотя бы минимальный вариант резервного копирования. Если вы работаете в банке или в другом подобном

учреждении, где данные стоят очень дорого, то обычно удается реализовать схему резервного копирования по максимуму. Добейтесь письменного одобрения ваших действий (лучше всего приказа по организации), чтобы не было отступлений от утвержденной сметы.

Основной фактор, определяющий стоимость системы резервного копирования, — объем архивируемых данных и время, выделяемое на эту процедуру. Чем больше объем данных, обрабатываемых системой резервного копирования в единицу времени, тем дороже получается создаваемая система. Если на проведение резервного копирования ваше серверное хозяйство может выделить ограниченный интервал времени, или если оно функционирует круглосуточно, при планировании системы резервного копирования следует учитывать это обстоятельство, поскольку изменение резервируемых данных до завершения копирования приводит к некорректным результатам. Если вы можете остановить работу сервера либо по окончании рабочего дня сервер не используется, процесс резервирования становится тривиальным и, как правило, достаточно дешевым.

Ежедневные процедуры копирования должны выполняться в то время, когда данные находятся в некотором стабильном состоянии, например, после окончания рабочего дня. Если невозможно исключить сервер на время создания резервной копии из производственного процесса, руководство фирмы должно знать о потенциальных осложнениях — обычно это не совсем корректное сохранение информации. В серьезных серверах баз данных об этой проблеме знают, и существуют методы получения точной резервной копии баз данных.

Большинство систем резервного копирования строится на использовании либо команды `cron`, либо собственных утилит автоматического вызова программ по установленному расписанию. Как правило, они позволяют разрабатывать и поддерживать относительно сложные графики проведения работ. Кроме того, системы резервного копирования могут предусматривать прямое взаимодействие с приложениями для запуска их собственных механизмов резервного копирования. Зачастую резервное копирование осуществляют программы и скрипты собственной разработки, учитывающие особенности функционирования вычислительной среды организации.

Выбор схемы хранения резервных данных — еще один фактор, существенно влияющий на стоимость создаваемой системы резервного копирования. Схема хранения должна соответствовать специфике организации, а также требованиям контролирующих органов (например, для банков требования Центрального банка в отношении резервирования данных очень высоки). Кроме того, при выборе схемы хранения нужно учитывать и требования к восстановлению после аварий.

Большинство систем резервного копирования обеспечивают эффективное использование носителей информации за счет организации, по крайней мере, двух независимых уровней хранения данных. Так, *полная* копия содержит все без исключения файлы системы. При *инкрементном* копировании в архив помещают только те файлы, которые были изменены с момента создания последней полной или инкрементной копии. Применяя различные алгоритмы резервного копирования, можно разработать стратегию, которая сбалансирует требования к эффективности и надежности.

Приведем пример схемы резервного копирования, которую можно реализовать в достаточно крупной фирме. Все данные копируют по субботам. С воскресенья по пятницу создают инкрементные копии. Носители информации с еженедельными и ежедневными копиями возвращают на перезапись через месяц.

Формат хранения резервных копий должен быть таким, чтобы резервную копию при желании можно было развернуть на другой операционной системе (например, Windows). Рекомендуется пользоваться программами tar и gzip, аналоги которых существуют практически в любой операционной системе. Это позволит в случае надобности извлечь нужные файлы практически где угодно.

Чтобы не превратить библиотеку резервных копий в ненужную "свалку данных", необходимо составить каталоги данных резервного копирования. Обычно в более или менее серьезных пакетах резервного копирования присутствуют функции ведения каталогов.

Базовые утилиты, в том числе tar и cpio, не позволяют создавать подобные каталоги данных резервного копирования. Если для копирования применяются именно они, то каталоги придется вести либо с помощью специально созданного программного обеспечения, либо вручную.

Некоторые приложения для корректного функционирования требуют абсолютной согласованности наборов данных. Так, системы управления базами данных обычно имеют собственные средства резервного копирования. Поскольку принадлежащие такого рода системам данные непрерывно изменяются, задача фиксирования их согласованного состояния выходит за рамки возможностей программ типа tar, cpio и dump.

Для разрешения указанной проблемы разработчики обычно включают в подобные системы специальное программное обеспечение, способное зафиксировать в копии согласованное состояние их данных. В API системы могут включаться необходимые вызовы, или администраторам предоставляются специализированные сценарии, вызываемые из приложения. Поскольку такие приложения и системы копирования не имеют единого интерфейса, потребуется самостоятельно создать связующие программы промежуточного уровня.

В тех случаях, когда деловой процесс позволяет останавливать приложения, работающие с базами данных, а сами базы данных периодически закрываются, можно применять и обычную схему резервного копирования.

Те серверы баз данных, которые поддерживают репликацию, также нуждаются в создании резервных копий. Репликация не защищает от случайного или преднамеренного удаления данных. Кроме того, сетевые соединения между реплицируемыми системами могут отказывать, что вызывает нарушение согласованности данных.

Стратегию резервного копирования баз данных следует разрабатывать совместно с администраторами баз данных. В одних случаях может оказаться приемлемым разрешить администраторам баз данных выполнять процедуры резервного копирования и восстановления вручную. В других случаях эти действия должны выполняться при участии системного администратора.

Что такое резервное копирование

Основная идея резервного копирования — создание копий всего, что установлено на вашей системе, с некоторыми исключениями. Основные исключения, не входящие в резервные копии:

- файловая система /proc — содержит только данные, которые ядро генерирует во время работы операционной системы, и нет никакого смысла сохранять их;
- файловая система /sys — аналогично /proc;
- файловая система /mnt — поскольку в нее монтируются сменные носители (CD-ROM, флешки и т. п.);
- /tmp — тут находятся временные файлы;
- /lost+found — здесь хранятся "потерянные" и восстановленные файлы, обычно не представляющие ценности для резервирования;
- сетевые каталоги — смонтированная файловая система NFS, Samba и прочие виды сетевых данных;
- программное обеспечение, которое можно легко установить повторно. Здесь нужно иметь в виду, что в состав ПО могут входить конфигурационные файлы, которые необходимо копировать, чтобы не выполнять работы по их настройке позже.

Носители данных

Тип носителей для резервного копирования сильно зависит от ваших финансовых возможностей и объема сохраняемой информации. Совершенно нелогично покупать дорогую магнитооптику, если объем резервируемой информации не превышает одного-двух мегабайт за неделю. Рассмотрим носители информации и приводы, которые можно использовать в целях резервного копирования.

Жесткий диск

Резервное копирование на жесткий диск, установленный в системе, — неплохой бюджетный вариант резервного копирования и достаточно надежный. Существуют разные способы резервного копирования на жесткий диск: использование жесткого диска как хранилища данных, создание RAID-массивов различного уровня (стопроцентное резервное копирование "на лету" — "зеркалирование" жесткого диска) и др. Однако у этого варианта есть свои недостатки — при выходе из строя контроллера жестких дисков существует большая вероятность, что он за собой "потянет" и сами жесткие диски или какая-то программа начнет бесконтрольно писать (или стирать) данные. Возможное решение данной проблемы — резервный диск держать размонтированным и монтировать его только на время создания резервной копии. Можно также реализовать резервное копирование по сети на жесткий диск, расположенный на другом компьютере, однако в этом случае могут сказаться сетевые ошибки.

Внешний жесткий диск

Внешний жесткий диск с интерфейсом FireWire или USB 2.0 — недорогое (порядка 100 долл.) и надежное решение. Благодаря мобильности очень просто переносить резервные данные, развертывать систему из запасной копии. Однако внешнему жесткому диску противопоказаны удары и сильные сотрясения, особенно во время работы.

CD-RW

Благодаря невысокой стоимости приводов и носителей информации устройства CD-RW для резервного копирования в последнее время становятся очень популярными. Действительно, при стоимости устройства от 20 долл и чистого диска CD-R от 30 центов мы имеем достаточно дешевый вариант для хранения резервных копий средних размеров. Достоинства этого способа резервного копирования — дешевизна, большой срок хранения информации (некоторые производители дисков обещают двадцать лет сохранности данных) и доступность считывающих устройств. Недостаток — ограниченный объем диска (700 Мбайт). В настоящее время CD-диски устарели.

DVD-RW

DVD-RW лишены основных недостатков CD-RW — их емкость доходит до 8,5 Гбайт и на сегодня это оптимальный способ резервирования.

Blue Ray-привод

Большая емкость диска (порядка 25 Гбайт) весьма привлекательна для резервирования, но пока это устройство дороговато. Массовое использование начнется в ближайшем будущем.

USB Flash-накопители

При стоимости Flash-устройства объемом 32 Гбайт порядка 60 долл вариант целесообразен для мобильного резервирования. Достоинства — достаточно высокая скорость записи и чтения, устойчивость к механическим и магнитным воздействиям.

Магнитооптические диски

Существуют разные модификации магнитооптических дисков емкостью от 640 Мбайт и до 4,7 Гбайт, а в дальнейшем производители обещают еще бóльшие емкости носителей. Наряду с ленточными накопителями (стримерами), считаются основными устройствами для резервирования данных в серьезных проектах. Для магнитооптики разработаны специальные библиотеки, благодаря чему можно сохранять терабайты информации. К сожалению, магнитооптические накопители дороги и пока недоступны небольшим фирмам.

Стримеры

Пожалуй, одно из старейших устройств резервного копирования. За свою долгую жизнь получило широкое распространение. Достоинства — отработанная десятилетиями технология, неплохая надежность и средняя себестоимость хранения информации. Недостатки — привод довольно дорогой, чехарда с форматами кассет и хранения данных, ограниченный срок службы ленты (несколько лет).

На базе ленточных накопителей создают библиотеки и роботизированные системы для хранения огромных объемов данных, однако вам вряд ли доведется столкнуться с такими устройствами.

NAS

Network Attached Storage — подключаемое сетевое хранилище. По сути, это узкоспециализированный сервер, предоставляющий дисковое пространство. Диски объединены в RAID-массив, в некоторых версиях состоящий из десятков и даже сотен дисков. Таким образом, NAS — некая разновидность внешнего диска. Минимальная стоимость — 200 долл.

Тестирование архивов

Обязательный элемент резервного копирования — тестирование полученных архивов. Хранение непроверенной резервной копии создает ложное ощущение защищенности. Исходя из этого, каждую резервную копию перед хранением следует проверить на целостность.

Пробное развертывание системы из резервной копии необходимо, во-первых, для проверки процедуры восстановления данных, во-вторых, чтобы убедиться в корректности самих резервированных данных.

Еще одна проблема, связанная с восстановлением систем из резервных копий, заключается в установлении права владения файлами. Когда данные извлекает из файла копии пользователь с правами root, утилита GNU tar предпринимает попытки восстановить существовавшие права владения (пользователя и группы) каждым файлом, но только если при ее вызове был установлен переключатель (`--preserve-permission`). В противном случае утилита tar будет использовать текущие установки UMASK. Однако если перечисленных в файле tar пользователей и групп не существует, то право владения не может быть корректно установлено!

Следовательно, прежде чем восстанавливать любые некорневые файлы системы, следует восстановить файлы /etc/passwd и /etc/group.

Даже в том случае, когда компьютер под Linux является рабочей станцией единственным пользователем, корректное функционирование многих программ и подсистем будет зависеть от установленных прав владения и разрешений на доступ к системным файлам.

Риск при тестировании архивов

Тестирование процедур восстановления может быть достаточно рискованным, особенно когда вы только отрабатываете резервное копирование. Возможно случайное разрушение файлов, задействованных другим приложением в момент восстановления файлов.

Самый безопасный способ выполнить контрольное восстановление системы — провести его на резервной рабочей станции, а не на сервере, находящемся в промышленной эксплуатации.

Утилиты резервного копирования

В этом разделе приводятся примеры подготовки и создания полных, выборочных и инкрементных резервных копий с помощью утилит tar, cpio и dump/restore.

Для определенности будем считать, что в нашей системе установлен стример. Аналогично можно работать и с другим оборудованием.

Создание резервной копии утилитой tar

Самый простой вариант использования утилиты tar — создать архив всех каталогов, начиная с корневого. В этом случае простейшая команда для вызова утилиты tar с целью создания копии будет иметь следующий вид:

```
tar c /
```

Однако при выполнении указанной команды возникнет несколько проблем. Во-первых, по умолчанию утилита может выбрать не тот тип ленточного устройства, который установлен на данном компьютере, и даже вообще осуществлять вывод не на магнитную ленту. Во-вторых, в этом примере будет считано все дерево файловой системы. Это значит, что будет обработана файловая система /proc, любые установленные CD-ROM, файловые системы NFS и Samba, а также другие разделяемые сетевые файловые системы.

Приведенный пример может вызвать и несколько других проблем. Например, при обработке подобной команды GNU tar никогда не будет обеспечивать специальной поддержки sparse-файлов (файлов, имеющих реальный размер меньше, чем место, зарезервированное под них в файловой системе) и выполнять сжатие выходной информации.

Вот пример более корректного вызова утилиты:

```
tar cSlzf – $(backdirs) | buffer -o /dev/st0
```

Здесь создается (`c`) архив с поддержкой sparse-файлов (`S`), ограниченный локальными файловыми системами (`l`). Выполняется сжатие данных (`z`) и их запись в файл (`f`) stdout, в архив включаются только указанные каталоги (`backdirs`).

Созданный архив по каналу передается программе buffer, которая записывает его (`-o`) на первое ленточное устройство с интерфейсом SCSI. Подобный подход целесообразен и при получении резервной копии от программы сжатия или через сеть.

Параметр `bakdirs` — это сценарий, в котором указаны каталоги и файлы, включаемые в создаваемую резервную копию. Сценарий `backdirs` может состоять просто из команд `echo`, которые перечисляют все точки входа локальных файловых систем (за исключением каталога /proc, любых каталогов /temp, установленных CD-ROM, каталогов NFS и других сетевых ресурсов). Назначение `bakdirs` состоит в просмотре и фильтрации выходных данных команд `mount`, что позволяет динамически включать в копию только требуемые файловые системы. Неудобство обычного статического списка — он не может автоматически обновляться при добавлении новых файловых систем.

Избегайте дублирования ссылок в командной строке, содержащей вызов утилиты tar. Если одновременно будут копироваться каталоги /some/moutpoint и /some/mountpoint/somedir, расположенные в одной и той же файловой системе, утилита tar дважды поместит в архив все содержимое каталога /some/mountpoint/somedir.

Использование утилиты cpio

Утилита cpio представляет собой еще один традиционный инструмент создания резервных копий и архивирования файловых систем. В сравнении с утилитой tar ее работа организована иначе.

Во многих случаях принимаемый в утилите cpio подход к указанию подлежащих копированию или восстановлению файлов и каталогов прямо противоположен подходу, применяемому в утилите tar. При создании архива утилите tar передается список файлов и каталогов, указываемых как параметры командной строки. Любой указанный каталог просматривается рекурсивно. При создании архива с помощью утилиты cpio ей предоставляется список объектов (имена файлов и каталогов, символические имена любых устройств, гнезда доменов UNIX, поименованные каналы и т. п.). Этот список помещается в стандартный поток stdin утилиты cpio с помощью канала и обычно генерируется командой `find`.

Простая команда, выполняющая копирование всей файловой системы, выглядит следующим образом:

```
find / -print0 I cpio –o0B > /dev/st0
```

Результаты выполнения команды `find` будут включать каталог /proc и тому подобные нежелательные для резервного копирования каталоги. Уточнив параметры команды `find`, можно исправить ситуацию:

```
find /* -fstype ext2 –print0 I cpio –o0B > /dev/st0
```

В этом примере копируемые объекты ограничены только файловыми системами типа Ext2. Также будут пропущены все скрытые файлы и каталоги.

Устройства вывода информации на магнитную ленту стоят дорого. Получить *доступ к удаленным устройствам* не намного сложнее, чем к локальным:

```
find /* -fstype ext2 –print0 |  ssh $TAPEHOST "cpio –o0B I buffer -o /dev/st0"
```

Обратите внимание, что для обращения к удаленным ленточным устройствам указана команда `buffer`.

Восстановление с локального ленточного устройства

Еще одно принципиальное различие между утилитами tar и cpio состоит в способе сохранения и восстановления абсолютных путей. В случае с утилитой tar ведущая косая черта в абсолютных именах файлов при создании копии удаляется. Утилита cpio в процессе восстановления принудительно превращает все пути в относительные.

Как правило, файлы должны восстанавливаться в тех каталогах, которые будут задаваться относительно текущего каталога (в некоторых случаях относительно каталога root). По умолчанию утилита cpio не восстанавливает каталоги, поэтому при ее вызове следует указывать параметр `-d`.

Восстановление с удаленного ленточного устройства

Восстановление с удаленных ленточных устройств осуществляется так же просто, как и копирование, например:

```
ssh $OTHERHOST 'buffer -i /devst0', I 'find /* -fstype ext2 –print0 |    cpio -id'
```

Если необходимо восстановить только некоторые файлы, добавьте в конец команды `cpio` список глобальных шаблонов.

Здесь обнаруживается еще одно различие между утилитами cpio и tar, связанное с выполнением частичного восстановления. При использовании утилиты tar список требуемых файлов и каталогов можно поместить прямо в команду ее вызова. Однако утилита tar не поддерживает глобальных шаблонов.

При работе с утилитой tar типичный способ обойти это ограничение состоит в том, чтобы извлечь индекс архива в файл путем простого перенаправления вывода. Данные полученного файла фильтруются с помощью команды `grep`, после чего полученный список передается команде вызова утилиты tar для извлечения данных. Например, подготовив файл restorelist, содержащий имена требуемых файлов и каталогов, помещенные в отдельные строки, можно ввести команду

```
ssh $OTHERHOST 'buffer -i /dev/st0' I 'tar xTf /tmp/restorelist -'
```

Программа резервного копирования dump

Программа dump в корне отличается от tar — она предназначена для резервного копирования и восстановления файловой системы и создает резервные копии элементов файловой системы. Эта утилита позволяет получить копию одной файловой системы быстро и эффективно. К сожалению, ее нельзя применить к отдельным каталогам. Программа restore выполняет функцию, обратную dump, она восстанавливает полную резервную копию файловой системы.

Утилита dump имеет несколько уровней резервного копирования от 0 до 9, где уровень 0 — полная резервная копия системы — гарантирует, что все элементы файловой системы будут скопированы. Уровни выше 0 — добавочные резервные

копии, которые указывают dump копировать все новые или модифицированные после последнего копирования файлы. Чтобы быть более точным, на каждом уровне добавочного резервного копирования вы сохраняете все изменения, произошедшие после создания последней резервной копии на том же или предыдущем уровне. Так можно осуществлять инкрементное резервирование системы.

Создание резервных копий с помощью программы dump

Использование программы dump очень простое:

```
dump -0u -f /dev/st0 /home
```

Таким образом, мы создали полную копию каталога /home. Для получения инкрементной копии выполним следующую команду:

```
dump -3u -f /dev/st0 /home
```

Восстановление файлов, созданных dump

Для восстановления резервных копий, созданных утилитой dump, служит программа restore, которая восстанавливает файлы и каталоги. При диалоговом восстановлении файлов из копии программа restore предоставляет интерфейс, который позволяет пользователю перемещаться по дереву каталогов, выбирая файлы для извлечения, после чтения информации о каталогах из копии.

При восстановлении резервной копии мы должны перейти в раздел файловой системы, где хотим восстанавливать нашу резервную копию. Это требуется, т. к. в диалоговом режиме программа restore восстанавливает все файлы раздела файловой системы, из которой она была запущена.

Для восстановления файлов из копии в диалоговом режиме задайте команду

```
restore -i -f /dev/st0
restore >
```

На вашем терминале вы увидите командную строку, для получения списка файлов текущего или заданного каталога выполните команду `ls`:

```
restore > ls
admin/ lost+found/ named/ quota.group quota.user wahib/
restore >
```

Чтобы внести текущий каталог или файл в список файлов для извлечения, укажите команду `add`:

```
restore > add Personal/ restore >
```

Удалить текущий каталог или заданный файл из списка файлов для извлечения можно командой `delete`.

Восстановить все файлы из списка для извлечения позволяет команда `extract`.

Для выхода из интерактивного режима программы restore после завершения восстановления файлов введите команду `quit`.

Пакет AMANDA

Многие системные администраторы создают свои собственные сценарии и выполняют бóльшую часть работы по контролю за использованием томов вручную.

Пакет AMANDA (Advanced Maryland Automatic Network Disk Archiver) контролирует процесс проведения серий полных и инкрементных резервирований данных на промежуточное дисковое хранилище хоста ленточных устройств для некоторого набора сетевых клиентов. Затем полученные наборы данных переносят на ленточные носители. При корректной установке пакет AMANDA работает полностью автоматически. Процесс резервирования обычно запускается ночью и осуществляет все операции копирования, последовательно соединяя хост копирования с каждым из клиентов. Существует возможность устанавливать эти соединения параллельно, а также контролировать и регулировать создаваемую нагрузку на сеть. Модуль планировщика определяет, какой уровень инкрементного копирования должен быть выполнен для каждой файловой системы каждого из хостов.

Программы пакета AMANDA создают архивные файлы посредством утилит dump или tar. Поэтому извлекать созданные ими архивы можно с помощью обычных инструментов. Кроме того, для восстановления предусмотрена команда `amrecover` пакета AMANDA.

Команды *mt* и *mtx*

Команды `mt` и `mtx` предназначены для управления устройствами вывода на магнитную ленту. Команда `mt` служит для получения сведений о состоянии устройства, определения или установки абсолютной и/или логической позиции головки над носителем, перемотки носителя, извлечения носителя и поиска "по направлению вперед" конца последнего блока записанных данных.

Команда `mtx` включает расширения для большинства распространенных типов устройств автоматической смены носителей.

Команда *buffer*

Команда `buffer` осуществляет вывод непрерывной последовательности данных на носитель даже в тех случаях, когда предоставляющие эти данные команды выводят их неравномерно. Описанная ситуация типична для работы программ сжатия данных, считывающих информацию через сетевые соединения. Подобное возможно даже при выполнении нормальных операций архивирования в среде перегруженной многозадачной операционной системы.

Многотомные резервные копии

Приведенные ранее примеры не предназначены для создания резервных копий, занимающих несколько магнитных лент. Утилиты tar и cpio предоставляют параметры, позволяющие разместить любой архивный файл на нескольких томах

носителей. Хранение частей архива на нескольких лентах снижает надежность, поскольку порча любой из лент делает бесполезным весь остальной набор носителей. Тем не менее иногда это необходимо, а при соответствующей адаптации подхода и полезно, например для резервирования на компакт-дисках.

Ссылки

- **www.veter.sky.net.ua/docs/linux/LINUXSOS/index.html** — Gerhard Mourani. Безопасность и оптимизация Linux. Редакция для Red Hat.
- **www.amanda.org** — сайт программы AMANDA.
- Соответствующие страницы man.



Глава 36

X Window и другие графические оболочки

Операционная система Linux давно уже немыслима без графической оболочки X Window (X Org), по крайней мере, на рабочих местах пользователей, поэтому необходимо иметь хотя бы общее представление о ее конфигурировании. В принципе, в большинстве современных дистрибутивов во время инсталляции система корректно распознает вашу аппаратуру и настраивает X Window, однако некоторые аспекты конфигурирования иногда желательно подправить. История X Window достаточно сложна, как с правовой точки зрения, так и с технической. В результате очередных правовых споров от X Window отпочковался проект X Org, который входит в большинство дистрибутивов.

Конфигурирование X Window (X Org)

Конфигурирование включает в себя четыре основных компонента:

- конфигурирование X-сервера;
- конфигурирование диспетчеров окон Window Manager;
- конфигурирование прикладных программ;
- русификацию.

Эти действия может выполнить как администратор (для всей системы сразу), так и пользователь (только для себя). Исключениями являются лишь X-сервер, конфигурацию которого может модифицировать только root, и, частично, русификация.

Конфигурирование X-сервера

Система X Window базируется на X-сервере, выполняющем основную работу системы. Все настройки X-сервера располагаются в файле /etc/X11/xorg.conf.

Этот файл состоит из нескольких секций, каждая из которых содержит настройки для определенной подсистемы: шрифтов, мыши, клавиатуры, монитора, видеоадаптера.

Общий вид секции такой:

```
Section "имя-секции"
  данные
  ...
EndSection
```

Внутри секций могут быть подсекции — они определяются парой ключевых слов `SubSection`/`EndSubsection`.

В табл. 36.1 приведены основные секции конфигурационного файла xorg.conf.

Таблица 36.1. *Основные секции файла xorg.conf*

Секция	Содержимое
`Files`	Пути к используемым файлам — в основном, это каталоги со шрифтами либо используемый сокет
`InputDevice`	Описание устройств ввода — клавиатуры, мыши
`Device`	Описание видеокарты
`Screen`	Описание экрана — разрешение и глубина цвета
`ServerLayout`	Описание используемых в настоящий момент секций, описывающих нашу конфигурацию
`Module`	Описание загружаемых сервером модулей

В листинге 36.1 приведен пример конфигурационного файла xorg.conf для ноутбука.

Листинг 36.1

```
# Xorg configuration created by pyxf86config

Section "ServerLayout"
      Identifier     "Default Layout"
      Screen      0  "Screen0" 0 0
      InputDevice    "Keyboard0" "CoreKeyboard"
      InputDevice    "Synaptics" "CorePointer"
EndSection
Section "InputDevice"
      Identifier  "Keyboard0"
      Driver      "kbd"
      Option      "XkbModel" "pc105"
      Option     "XkbLayout" "us,ru"
      Option     "XkbOptions" "grp:shifts_toggle,grp_led:scroll"
EndSection

Section "InputDevice"
      Identifier  "Synaptics"
      Driver      "synaptics"
      Option      "Device" "/dev/input/mice"
      Option      "Protocol" "auto-dev"
```

```
      Option      "Emulate3Buttons" "yes"
EndSection

Section "Device"
      Identifier  "Videocard0"
      Driver      "intel"
EndSection

Section "Screen"
      Identifier "Screen0"
      Device     "Videocard0"
      DefaultDepth     24
      SubSection "Display"
            Viewport   0 0
            Depth     24
      EndSubSection
EndSection

Section "InputDevice"
    Identifier    "Mouse0"
    # Modified by mouseconfig
    Driver        "mouse"
    Option        "Device"             "/dev/mouse"
    Option        "Protocol"           "IMPS/2"
    Option        "Emulate3Buttons"    "no"
    Option        "ZAxisMapping"       "4 5"
EndSection
```

Секция *Files*

В этой секции задается местоположение файла с перечнем цветов и содержится список каталогов, в которых X-сервер должен искать шрифты.

Порядок директив, задающих пути к шрифтам, имеет значение — при подборе шрифтов по псевдонимам они будут искаться в указанном порядке. Таким образом, если поставить каталог со шрифтами koi8-r в начало списка, то во многих случаях вместо европейских будут использоваться кириллические шрифты.

Секция *Keyboard*

В этом разделе определяются параметры и поведение клавиатуры, в частности переключатель раскладок клавиатуры.

Секция *Pointer*

Здесь задаются параметры мыши: тип устройства, эмуляция третьей кнопки, число кнопок и т. п.

Секция *Device*

В этой секции указываются параметры видеокарты. Обычно все параметры X-сервер определяет сам, считывая их непосредственно из видеокарты.

Когда требуется изменить какие-либо настройки, следует посмотреть man-страницу по используемому X-серверу.

Секция *Screen*

Здесь указывается, какую конфигурацию видеокарты и какой монитор следует выбрать, а также параметры видеорежимов — разрешение и глубина цвета.

Настройка параметров монитора

Впрочем, вам, скорее всего, не понадобится ручное вмешательство в конфигурацию X Window. Для настройки монитора, видеокарты, мыши, клавиатуры можно воспользоваться утилитами конфигурирования, например достаточно выполнить команду `Xorg -configure`, и вы получите файл xorg.conf. Но в нестандартной ситуации все равно придется править конфигурационный файл. Можно вызвать утилиту xorgcfg, она сперва выполняет `Xorg -configure`, а затем запускает X-сервер для тонкой настройки. Можно воспользоваться утилитой xorgconfig — она потребует сведения о мониторе, клавиатуре, мыши и другой аппаратуре.

Однако для старых электронно-лучевых дисплеев каждый режим работы монитора можно описать отдельно, что позволяет "выжать" из вашего монитора все, на что он способен. Для этого необходимо в конфигурационном файле задать для нужного видеорежима строку в таком формате:

```
Modeline  "mode_name"  D  H1 H2 H3 H4  V1 V2 V3 V4  Flags
```

где:

- `Modeline` — ключевое слово, определяющее строку с описанием видеорежима;
- `"mode_name"` — название нашего видеорежима. Написать можно что угодно, традиционно имя записывают в виде "разрешение_по_горизонтали × разрешение_по_вертикали", например "1024×768". `"mode_name"` используется в качестве ссылки на имя режима в `Section "Screen"`, `Subsection "Display"`, `Modes "mode_name"`. Режимы устанавливаются в порядке перечисления;
- `D` — частота тактового генератора в мегагерцах;
- `H1`, `H2`, `H3`, `H4` — числа, отвечающие за строчную синхронизацию;
- `V1`, `V2`, `V3`, `V4` — числа, отвечающие за кадровую синхронизацию;
- `Flags` — параметры для тонкой подстройки синхронизации.

Давайте разбираться с этими параметрами. У нас есть тактовая частота генератора, к которому привязываются все параметры видеокадра. Видеокадр состоит из видеострок. Строка имеет следующие параметры:

- `A` — число пикселов в строке (временной интервал, затрачиваемый на вывод строки пикселов);

- `B` — время между окончанием вывода строки и появлением строчного синхроимпульса;
- `C` — время, за которое выводится синхроимпульс;
- `D` — время обратного хода развертки.

Таким образом, для строчной развертки получаем:

```
H1 = A
H2 = A+B
H3 = A+B+C
H4 = A+B+C+D
```

Для кадровой развертки в качестве единицы измерения используется частота строк. Поэтому:

- `V1` — число строк, отображаемых в одном кадре;
- `V2` — число строк от начала кадра до начала кадрового синхроимпульса;
- `V3` — число строк от начала кадра до конца кадрового синхроимпульса;
- `V4` — общее число строк в кадре.

На современном оборудовании при установке операционной системы программа инсталляции выставляет частоты монитора по максимуму, поэтому ручное вмешательство в настройки монитора вам, скорее всего, не понадобится.

Последовательность запуска X Window

Чтобы лучше понять функционирование системы X Window, рассмотрим процесс ее запуска.

Стандартный процесс запуска состоит из 5–6 уровней:

1. Запуск пользователем программы startx.
2. Запуск программой startx программы xinit.
3. Запуск X Window и обработка файлов /etc/X11/xinit/xinitrc или ~/.xinitrc.
4. Обработка файлов /etc/X11/xinit/Xclients или ~/.Xclients.
5. Запуск разных программ.
6. Запуск Window Manager.

Бóльшая часть из перечисленного — скрипты, и при необходимости можно внести коррективы в процесс запуска.

Конфигурация Window Manager

Файлы конфигурации диспетчеров окон (Window Manager) располагаются в каталоге /etc/X11/ и находятся в подкаталоге, совпадающем с названием диспетчера окон. Синтаксис файлов конфигурации у каждого диспетчера свой, так что наилучший вариант настройки — почитать документацию и посмотреть примеры файлов.

Большинство современных диспетчеров окон имеют программу конфигурации, с помощью которой можно полностью их настроить.

В современных дистрибутивах отказываются от непосредственного использования диспетчеров окон и заменяют их графическими средами KDE или GNOME.

Графическая интегрированная среда

По большому счету — это операционная среда над операционной средой (кажется парадоксом), организующая единый стилистический интерфейс для приложений, написанных для графической среды, предоставляющая стандартизированные методы взаимодействия процессов и стандартные библиотеки. Приложения, входящие в графическую среду, отлично друг с другом взаимодействуют и представляют практически законченный интерфейс для офисного и домашнего применения. Сегодня наибольшее распространение получили две таких интегрированных среды — KDE и GNOME. И кстати, графическая среда может использовать различные оконные менеджеры (по крайней мере, GNOME).

Достоинство этого решения — набор программного обеспечения и стандарты взаимодействия и интерфейса. Недостаток — некоторая тяжеловесность, не позволяющая комфортно работать на слабых компьютерах. Более или менее современный компьютер вполне нормально работает в графической интегрированной среде, а те, у кого машина послабее, могут установить оконный менеджер попроще, например twm.

Графическая среда GNOME

GNOME (GNU Network Object Model Environment, Среда GNU, основанная на модели сетевых объектов) базируется на библиотеке GTK+ и реализована для разных платформ, что позволяет запускать ее в операционных средах Linux, BSD и Solaris. Система очень гибкая, использует внешний менеджер окон, в качестве которых можно применять наиболее распространенные оконные менеджеры.

Различные приложения GNOME взаимодействуют друг с другом с помощью CORBA (Common Object Request Broker Architecture), что обеспечивает независимость приложений от того, на каком языке они были написаны, или от того, на каком компьютере они работают.

Для настройки GNOME (и не только) не нужны никакие сторонние средства. На рис. 36.1 представлено меню администрирования системы.

Все прозрачно — список названия конфигурируемых служб, устройств и параметров. Все очень просто, на русском языке и с неплохой справочной системой. Единственное ограничение состоит в том, что настройки, затрагивающие систему в целом (например сетевые), требует пароля root, при настройке текущего пользователя пароль не нужен.

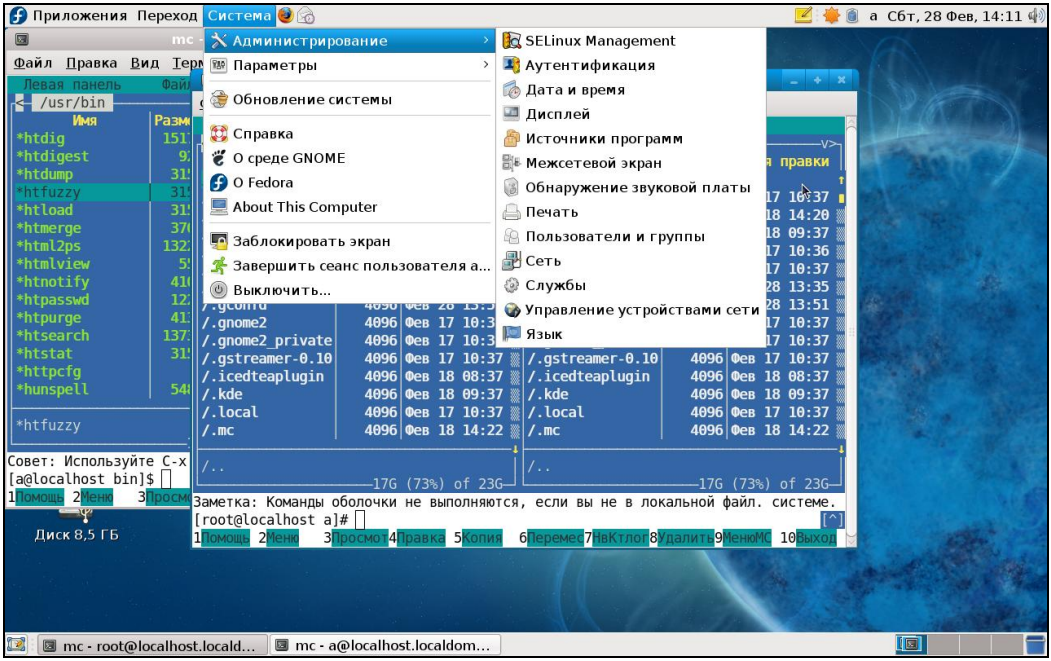


Рис. 36.1. Настройка GNOME

KDE — K Desktop Environment

Еще одна графическая интегрированная среда. По своим возможностям очень напоминает GNOME. Использует библиотеку Qt. Идеология KDE несколько отличается от GNOME — в ней невозможно применять независимые менеджеры окон, у нее несколько более строгий интерфейс. В отличие от GNOME, сильнее централизовано управление разработкой как KDE, так и программ для нее. Одна из основных задач, которые ставили разработчики, — сделать систему, напоминающую Windows, но гораздо лучше. Оболочка более эргономична и привычна для пользователей Windows. Благодаря координированной разработке дизайн практически всех программ решен в едином стиле, а благодаря популярности KDE, специально для нее был написан большой объем программного обеспечения. Как обычно, если система в чем-то хороша, у нее должны быть и недостатки. Недостатков, по большому счету, два:

- требовательность к ресурсам;
- библиотека QT (лицензионные ньюансы).

По существу конфигурирование KDE и GNOME мало различается. Та же система меню, то же разделение средств конфигурирования KDE и операционной системы.

Ссылки

- **knot.pu.ru/faq/xfaq** — XFAQ по настройке X Window.
- **www.linux.org.ru/books/gnome-ug/ug/** — руководство пользователя GNOME.
- **www.linux.org.ru/books/kde/general/userguide/index.html** — K Desktop Environment. Руководство пользователя.
- **sky.inp.nsk.su/~bolkhov/teach/inpunix/xsetup_simple.ru.html** — Дмитрий Болховитянов. Настройка X Window.
- **gazette.linux.ru.net/lg67/articles/rus-adam.html** — Thomas Adam. Колонка The Weekend Mechanic: настройка X Window. Перевод Владимира Меренкова.
- **gazette.linux.ru.net/lg64/articles/rus-sipos.html** — настройка режима монитора в XFree86.
- **linux.net.kg/articles/x.html** — настройка X Window.
- **www.gnome.org** — официальный сайт проекта GNOME.
- **www.kde.org** — официальный сайт проекта KDE.

Глава 37



Печать

Вот система настроена, кириллические шрифты есть, и X Window работает с заданными частотами и разрешениями, по Интернету путешествуем с приемлемой скоростью, документы разные находим, просматриваем… А распечатать ничего нельзя, т. к. принтеры не установлены.

В данной главе мы постараемся исправить это упущение. Принтеры бывают разные — матричные, струйные, лазерные и сублимационные, цветные лазерные и даже объемные. Они могут подключаться к различным интерфейсам: последовательному и параллельному портам, USB и даже Ethernet. Производители принтеров продолжают увеличивать набор проблем — то протокол свой придумают, то с целью удешевления создадут Win-принтер, для которого драйвер не достать. И во всем этом приходится разбираться.

Сначала опишем настройку принтера "правильным" способом, с применением стандартных UNIX-средств в консольном режиме. Затем рассмотрим конфигурацию с помощью графической утилиты, которая не идет ни в какое сравнение со стандартным способом — буквально за минуту можно сконфигурировать любой принтер.

Способы вывода на принтер

Существует несколько способов добиться вывода данных на принтер. Разумеется, в конце концов, все завершается простой передачей последовательности байтов в порт, к которому подключен принтер, однако с данными по пути следования от документа до распечатки могут проводиться различные манипуляции.

Самый простой путь — прямой вывод информации без всякой предварительной обработки на порт принтера. Для этого достаточно выполнить всего одну команду:

```
cat mytext.txt > /dev/lp
```

Вот аналогичная команда для DOS:

```
copy mytext.txt > prn
```

Как обычно, простота лишь кажущаяся. Во-первых, чтобы таким образом что-то отправить на печать, необходимо быть пользователем root — для остальных пользователей невозможно напрямую работать с файлами устройств. Во-вторых, зачастую вы получите на распечатке сплошную "кашу" из символов. Так произойдет потому, что любой принтер имеет свой специальный язык управления, причем

существует более десятка разновидностей этих языков. Один из вариантов — использовать специальные утилиты, на вход которых подают текстовые файлы, а на выходе получают текст, преобразованный с учетом языка управления принтера. Однако это крайне неудобно. Поэтому применяют специальные программные пакеты, предназначенные для управления печатью. Именно о них и пойдет далее речь.

Система печати CUPS

CUPS (Common UNIX Printing System, общая система печати для UNIX) отличается богатыми возможностями. В ней реализован протокол печати, сходный с HTTP, заменяющий морально устаревший протокол LPD.

Поддерживает форматы Adobe PostScript, PDF, HP-GL/2, TIFF, JPEG, PNG, PBM, PGM, PPM, GIF, SGI, RGB, Sun Raster, Kodak Photo CDTM. Интересные для администратора особенности системы:

- правила управления доступом;
- наличие системы квот;
- авторизация пользователя;
- ведение log-журналов.

Программный пакет LPD

LPD (Line Printer Daemon, демон линейной печати) — пожалуй, старейший программный пакет для печати в мире UNIX. Структура стандартна для UNIX: программы-утилиты для управления процессом печати и программа-демон, обеспечивающая вывод на несколько принтеров. Благодаря такому построению программного пакета вы имеете возможность одновременно работать с несколькими принтерами и настроить сетевую печать. В пакет входят следующие программы:

- lpd — демон системы печати;
- lpr — пользовательская команда печати. lpr выдает новое задание в очередь печати lpd. Синтаксис lpr очень прост:

  ```
  lpr [ опции ] [ имя_файла ... ]
  ```

  Если `имя_файла` не задано, lpr ожидает ввод данных со стандартного ввода. Это позволяет пользователям перенаправлять вывод команд в очередь печати;
- lpq — утилита для просмотра очереди печати. Команда, запущенная без аргументов, возвращает содержимое очереди печати принтера по умолчанию;
- lpc — утилита контроля lpd. С ее помощью можно производить любые манипуляции с очередью печати — добавлять и удалять задания, останавливать печать, переупорядочивать задания в очереди печати и т. д. lpc чаще всего используется в системах, где несколько принтеров установлено на один компьютер.

  Команду `lpc` обычно вводят в интерактивном режиме, однако никто вам не мешает указать опции. Вот некоторые из них:
  - `disable` — запрещает добавление любых новых заданий печати;
  - `down` — запрещает все задания на принтере;
  - `enable` — разрешает ввод новых заданий в очередь печати;

- `quit` (or exit) — покинуть lpc;
- `restart` — перезагрузить lpd для данного принтера;
- `status` — статус печати принтера;
- `up` — разрешить все и запустить новый демон lpd;

❒ lprm — утилита для удаления задания из очереди печати. Команда `lprm` удаляет из очереди все задания печати, владельцем которых является пользователь, выполнивший эту команду. Для того чтобы отменить одиночное задание печати, нужно сначала получить номер задания с помощью команды `lpq`, а затем сообщить полученный номер команде `lprm`.

Функционирует система следующим образом. При старте операционной системы запускается демон lpd. Из файла /etc/printcap он узнает, какие принтеры будет обслуживать. При запуске (пользователь что-то выводит на печать) lpr взаимодействует с lpd через именованный сокет /dev/printer и передает lpd-файл для печати и некоторую информацию о том, кто печатает и как печатать файл. Затем lpd печатает файл на соответствующем принтере в порядке очереди.

Настройка LPD

Начнем с простого: настроим обычный струйный принтер фирмы Hewlett-Packard HP DeskJet 400. Будем считать, что LPD уже установлен в вашей операционной системе, поскольку этот пакет входит во множество дистрибутивов как стандартная система печати.

Для добавления очереди печати к lpd вы должны внести запись в файл /etc/printcap и создать новый буферный каталог в каталоге /var/spool/lpd. В листинге 37.1 показана запись в файле /etc/printcap.

Листинг 37.1

```
# ЛОКАЛЬНЫЙ deskjet400
lp|dj|deskjet:\
        :sd=/var/spool/lpd/dj:\
        :mx#0:\
        :lp=/dev/lp0:\
        :sh:
```

Приведенная запись определяет принтер с псевдонимами lp, dj или deskjet, его спул печати размещается в каталоге /var/spool/lpd/dj. Отсутствует ограничение максимального размера задания. Печать производится на устройство /dev/lp0 и не сопровождается выводом страницы с именем человека, который печатает, добавленной в начало задания печати. Как вы видите — все очень просто. Но, во-первых, извечная проблема текстовых файлов UNIX и Windows состоит в том, что для UNIX в конце текстовой строки достаточно символа перевода строки, для Windows необходимо наличие символов возврата каретки и перевода строки. Большинство современных принтеров рассчитаны для Windows, и поэтому для нормальной печати

текста им также необходимо в конце текстовой строки наличие символов возврата каретки и перевода строки. Если не учесть данную особенность, при распечатке текста на принтере получится приблизительно следующее:

```
Строка номер один
                Строка номер два
                                Строка номер три
                                                Строка номер четыре
```

Это называется "лестничным эффектом", и с ним необходимо бороться. Существует много способов, самый простой — написать небольшой фильтр, через который перед печатью будет пропускаться наш текстовый файл, а результат уходить на печать.

Поправим нашу запись в файле /etc/printcap (листинг 37.2).

Листинг 37.2

```
# ЛОКАЛЬНЫЙ deskjet400
lp|dj|deskjet:\
        :sd=/var/spool/lpd/dj:\
        :mx#0:\
        :lp=/dev/lp0:\
        :if=/var/spool/lpd/dj/filter:\
        :sh:
```

В документации к printcap описаны атрибуты принтера `if` — входной фильтр и `of` — выходной фильтр. Как видите, мы определили входной фильтр, расположенный в каталоге /var/spool/lpd/dj/ и носящий имя `filter`. Этот файл представляет собой две строчки, написанные на Perl:

```
#!/usr/bin/perl
while(<STDIN>){chop $_; print "$_\r\n";};
print "\f";
```

В результате мы получаем принтер, на котором можно корректно распечатать текстовые файлы, используя встроенные шрифты принтера. Для современного мира это не актуально — практически всегда применяется графическая печать. Обычно печатают документы PostScript или графические файлы. На первый взгляд — нетривиальная задача, на самом деле все довольно просто. Вспомните еще раз идеологию UNIX: сколь угодно сложные задачи решать посредством последовательности небольших утилит.

Решение этой проблемы опять основано на свойстве файла printcap — использование входных и выходных фильтров. Если у нас будет фильтр, который сможет воспринимать произвольные типы файлов на входе, обрабатывать их в зависимости от формата файла и выводить на принтер, — мы решим нашу задачу.

Такой фильтр называется *магическим фильтром* (magic-filter). Существует много разных магических фильтров, причем наверняка какие-то из них находятся

в вашем дистрибутиве операционной системы. Приведем для примера два магических фильтра печати:

- APSFILTER — фильтр печати для стандартного lpd;
- lprMagic — фильтр печати с неплохими возможностями. Автоматически определяет тип входного документа, поддерживает печать через Samba.

Учет ресурсов

Обычно в больших фирмах принято хранить информацию о том, кто, когда и сколько печатал. Возможности стандартного LPD для учета ресурсов весьма ограничены. Вы можете указать имя файла для учета ресурсов через атрибут `af=` в printcap, но, по большому счету, это не решение проблемы. Пожалуй, лучший вариант — писать данные в файл учета ресурсов с помощью магического фильтра, а обрабатывать этот файл позднее каким-нибудь скриптом обработки статистики.

Программа печати LPRng

Доработанная версия LPD, по всей видимости, скоро станет стандартной во всех дистрибутивах Linux. LPRng легче для администрирования и имеет значительно лучшие возможности по сравнению с LPD при наличии большого количества принтеров (в том числе и сетевых). Программа безопаснее с точки зрения администратора, т. к. поддерживает аутентификацию через PGP или Kerberos.

Программный пакет netcat

Netcat — несложный программный пакет для работы с принтерами. Удобен и прост в настройке, имеет проблемы с сетевой печатью, однако для домашнего пользователя, которому не нужна сеть, — очень неплохой вариант.

Система печати PDQ

PDQ (Print Don't Queue, печатать не буферизуя). Это система печати без центрального демона. Она включает возможность объявления настроек печати, а также графическую утилиту и утилиту командной строки для настройки и вывода на печать.

Для управления печатью предусмотрены следующие программы:

- Xpdq — приложение для X Window, которое показывает список доступных принтеров и данные об очереди печати. Вы можете настроить ваш драйвер принтера в диалоговом окне **Driver Options**; обычно можно установить параметры двунаправленного соединения, плотность печати, размер и тип бумаги и т. д.;
- Pdq — утилита командной строки. Она может заменять команду `lpr` в большинстве случаев. Подобно lpr, она печатает либо перечисленные файлы, либо данные со стандартного ввода.

Функционирует PDQ следующим образом:

- запускается pdq или xpdq с указанием файла, который необходимо распечатать;
- выбирается принтер;
- определяются параметры печати — двусторонняя печать, число копий, качество печати и т. д.;

- программа анализирует содержимое файла, который вы печатаете, и следует инструкциям, записанным в файле драйвера PDQ, которые описывают, как обрабатывать ваши данные для печати на данном принтере с заданными параметрами;
- программа посылает обработанные данные на принтер через указанный интерфейс — прямо на /dev/lp0, или сетевому демону LPD, или на факс-гейт (специальную программу, на вход которой поступают документы, предназначенные для отправки по факсу. Эта программа при помощи факс-модема дозванивается до нужного абонента и автоматически отправляет факс);
- если PDQ не может послать данные на принтер указанным способом, то она запускает в фоновом режиме процесс, который пытается осуществить печать.

Настройка PDQ

PDQ может настроить либо администратор, либо обычный пользователь. Администратор для настройки PDQ редактирует файл /etc/printrc, а обычный пользователь может изменять только свой персональный файл .printrc.

PDQ позволяет пользователям выбрать принтер, на который будет выводиться печать. Принтеры в PDQ определяются как комбинации драйвера и интерфейса и являются текстовыми описаниями в файле настройки PDQ.

Интерфейс PDQ описывает то, как данные посылаются на принтер. Приведем некоторые параметры интерфейса:

- `local-port` — интерфейс локального порта работает с параллельным или последовательным портом на той машине, на которой запущен PDQ. Этот интерфейс обеспечивает вывод прямо в параллельный порт;
- `bsd-lpd` — интерфейс `bsd-lpd` общается по сети с демоном LPD или с работающим по протоколу LPD сетевым принтером. PDQ поддерживает постановку, отмену заданий и запросы к интерфейсу LPD.

Драйвер PDQ описывает, как перевести выводимые на принтер данные в формат, воспринимаемый принтером. Для принтеров, понимающих PostScript, он включает преобразование из ASCII в PostScript; для остальных принтеров — описывает преобразования из PostScript в язык принтера, используя GhostScript.

Для того чтобы определить принтер в PDQ, необходимо запустить xpdq и выбрать команду меню **Printer | Add printer**. Этот мастер настройки проведет вас через выбор нужного драйвера и интерфейса.

Вот, собственно, и все по настройке PDQ. Если вашего принтера нет в списке принтеров, поддерживаемых программой PDQ, — почитайте документацию, там описано, как можно самостоятельно добавить ваш принтер.

Система буферизации печати PPR

PPR — система буферизации печати, ориентированная на PostScript. Она обладает хорошими возможностями учета, поддержки клиентов Appletalk, SMB и LPD. Система PPR, как и другие системы буферизации, может вызывать Ghostscript для работы с принтерами, не понимающими PostScript.

Печать на сетевой принтер

Одно из важных свойств пакетов PDQ и LPD — возможность печати по сети на принтер, физически подключенный к другому компьютеру, принт-сервер или просто сетевой принтер.

Для того чтобы разрешить удаленным компьютерам печатать на ваш принтер, используя протокол LPD, вы должны перечислить эти компьютеры в файле /etc/hosts.lpd. Помимо этого, вы можете разрешить только определенным пользователям с других компьютеров печатать на ваш принтер.

Для печати на другой компьютер вы должны в /etc/printcap сделать запись, приведенную в листинге 37.3.

Листинг 37.3

```
# Удаленный deskjet400
lp|dj|deskjet:\
        :sd=/var/spool/lpd/dj:\
        :rm=machine.out.there.com:\
        :rp=printername:\
        :lp=/dev/null:\
        :sh:
```

Как видно из листинга, на нашем компьютере существует каталог очереди печати, обслуживаемой lpd. Это позволяет сохранить и распечатать задание позднее, если удаленная машина занята или отключена. Также мы определяем имя компьютера, который предоставляет нам свой принтер (`machine.out.there.com`), имя принтера на удаленном компьютере (`printername`) и показываем, что сетевой принтер не подключен ни к какому ресурсу на нашем компьютере (`lp=/dev/null`).

Использование принт-сервера

Сегодня уже нет необходимости для небольшой локальной сети покупать достаточно дорогой Ethernet-принтер или выделять отдельный компьютер для организации принт-сервера. Множество фирм выпускают специализированные принт-серверы, причем стоимость некоторых моделей уже менее 50 долл США.

В качестве примера конфигурации и использования такого принт-сервера в сетях UNIX рассмотрим принт-сервер фирмы Surecom.

Принт-сервер поддерживает следующие сетевые протоколы:

- Novell NetWare IPX/SPX и NDS;
- TCP/IP;
- DHCP для автоматического получения IP-адреса;
- BOOTP для автоматического получения IP-адреса;
- RARP для автоматического получения IP-адреса.

Принт-сервер Surecom взаимодействует с UNIX-хостами через протокол LPD, описанный ранее. Для конфигурирования принт-сервера необходимо выполнить следующие операции:

- ❒ включить поддержку протокола TCP/IP;
- ❒ назначить IP-адрес для принт-сервера;
- ❒ сконфигурировать удаленную LPD-печать на хостах;
- ❒ проверить корректность печати.

Опишем все по порядку.

Включить поддержку протокола TCP/IP и назначить IP-адрес для принт-сервера можно с помощью утилиты psetup, поставляемой с принт-сервером. После запуска программы необходимо:

- ❒ выбрать пункт **TCP/IP Configuration**;
- ❒ в пункте **TCP/IP Support** установить опцию ENABLE;
- ❒ в пункте **IP Address** указать IP-адрес, присваиваемый принт-серверу, либо установить все нули, если адрес назначается динамически;
- ❒ **DHCP server** — использовать динамическое назначение IP-адреса;
- ❒ **Gateway IP** — в этом поле задается IP-адрес шлюза;
- ❒ **Netmask** — сетевая маска;
- ❒ **Name server** — адрес DNS-сервера.

После того как вы ввели необходимые параметры, проверьте программой ping прохождение пакетов на ваш принт-сервер. Теперь перейдем к конфигурированию на хостах системы LPD.

На хосте создаем каталог, в котором будет находиться спул печати (листинг 37.4).

Листинг 37.4

```
mkdir /var/spool/lpd/pserverd
chown daemon /var/spool/lpd/pserverd
chgrp daemon /var/spool/lpd/pserverd
chmod 775 /var/spool/lpd/pserverd
```

Затем в файл /etc/printcap добавляем записи, приведенные в листинге 37.5.

Листинг 37.5

```
printer-name:\
:lp=:\
:rm=203.66.191.186:\
:rp=lpt1:\
:lf=/var/spool/lpd/pserverd.log:\
:sd=/var/spool/lpd/pserverd:\
:mx#0:
```

Здесь `rm` — IP-адрес принт-сервера, `sd` — спул принтера, `rp` — имя порта на принт-сервере.

После этого можно выполнить пробную печать командой

```
lpr -P<printer-name> <file> ...
```

Печать на Ethernet-принтер

Обычно высокоскоростные принтеры, позиционируемые производителем как устройства для совместной печати, имеют встроенный сетевой интерфейс, на который вы можете печатать по протоколу LPD. Как правило, в инструкции, идущей в комплекте с принтером, описывается, как необходимо настроить клиентский компьютер для печати на сетевой принтер. Например, следующая запись в файле printcap обеспечивает работу с сетевым принтером фирмы Hewlett-Packard:

```
lj-6|remote-hplj:\
        :lp=/dev/null:sh:\
        :sd=/var/spool/lpd/lj-6:\
        :rm=printer.name.com:rp=raw:
```

Принтеры HP LaserJet с интерфейсами Jet Direct поддерживают две встроенных очереди LPD: "raw", которая принимает PCL (или PostScript), и "text", которая принимает "чистые" файлы ASCII и автоматически устраняет лестничный эффект.

Графические утилиты конфигурирования принтера

Перейдем к визуализации информации — все-таки некоторые вещи проще делать в X Window.

В дистрибутивах Linux есть много удобных утилит, и одна из них — printconf-gui. С помощью этой простой утилиты установим принтер HP DeskJet 400.

На рис. 37.1 вы видите меню **Администрирование**, а в нем пункт **Печать.**

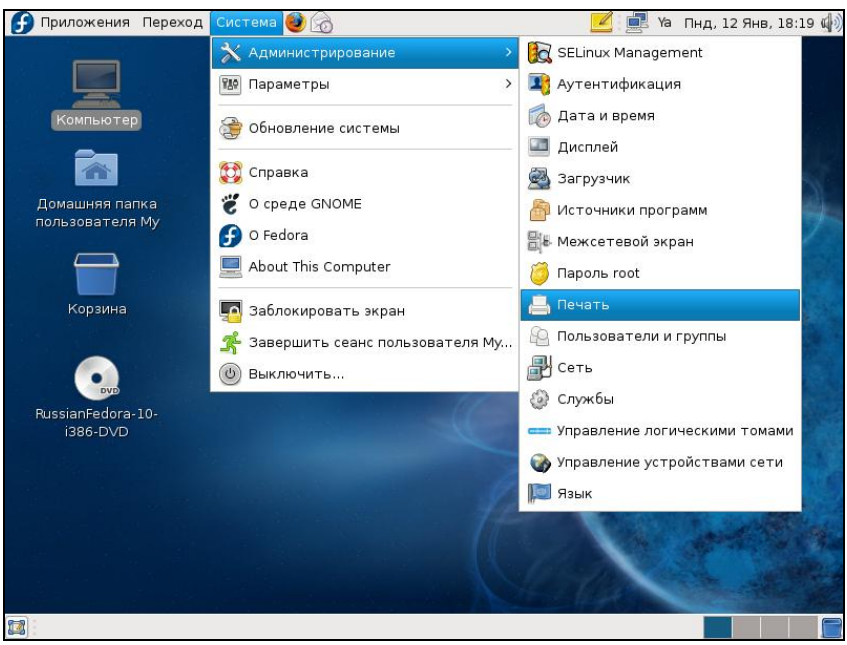


Рис. 37.1. Меню **Администрирование**

После запуска утилиты **Печать** мы увидим небольшое окно (рис. 37.2) всего лишь с четырьмя пунктами меню.

Перейдем к собственно конфигурации принтера. Процедура занимает мало времени и в большинстве случаев проходит без осложнений. Как видно из рис. 37.2, в верхней части окна утилиты присутствует кнопка **Создать**. Нажимаем. Если принтер новый и драйверы есть, система автоматически распознает и установит принтер. Если же этого не произошло, то появится окно, изображенное на рис. 37.3.

Нам необходимо выбрать тип подключения принтера (в данном случае по параллельному порту). Нажимаем кнопку **Далее**. Получаем следующее окно (рис. 37.4).

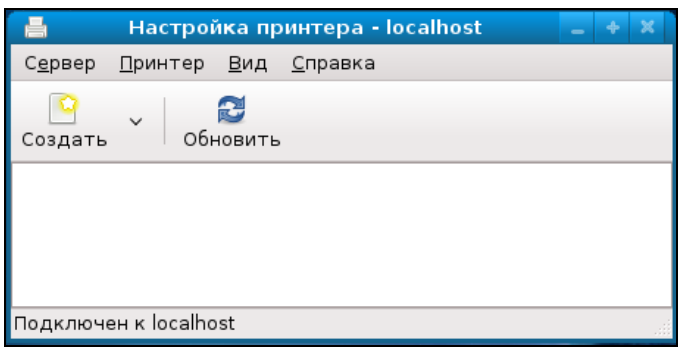


Рис. 37.2. Внешний вид утилиты printconf-gui

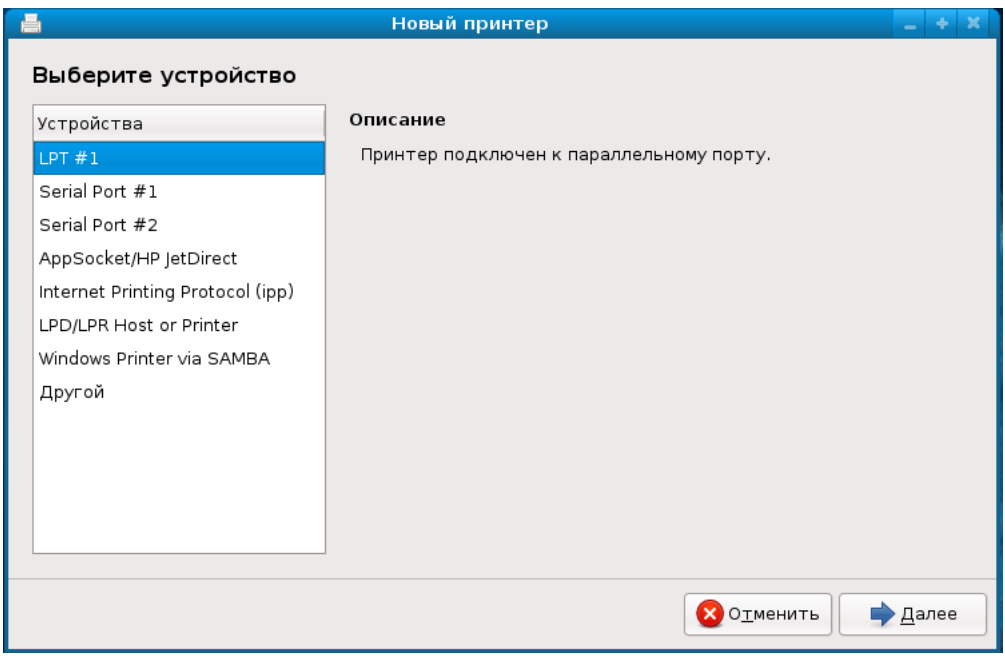


Рис. 37.3. Начало установки нового принтера

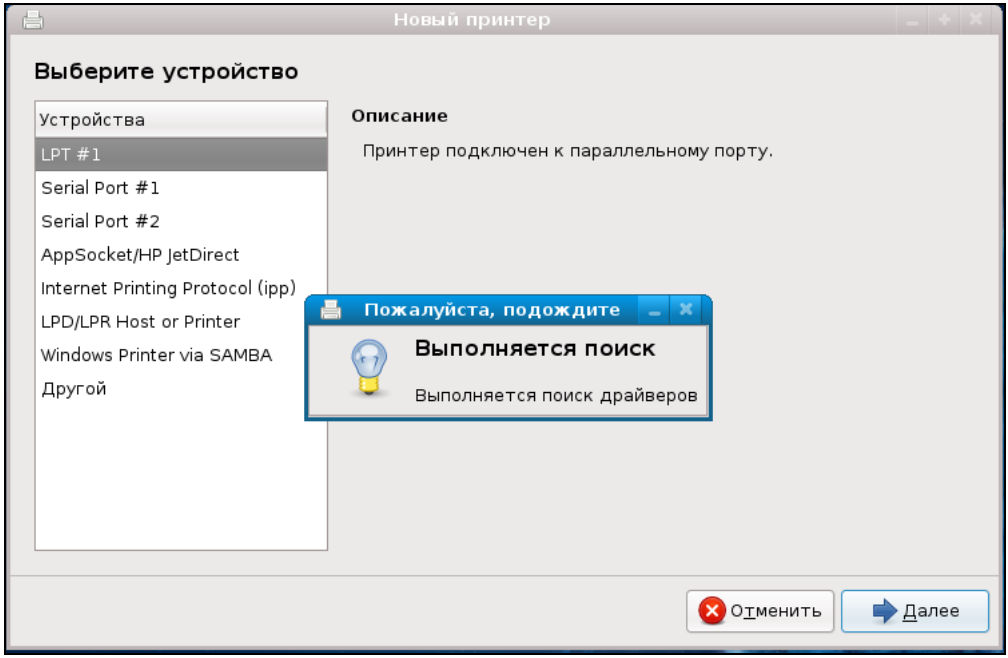


Рис. 37.4. Поиск драйверов системой

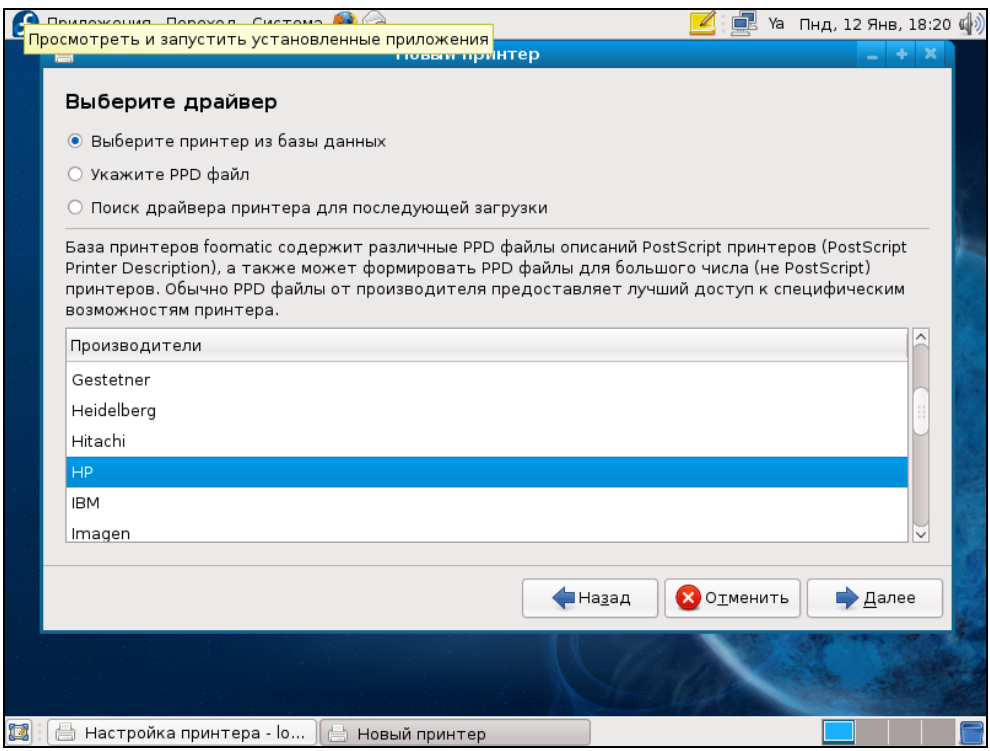


Рис. 37.5. Выбор семейства принтеров

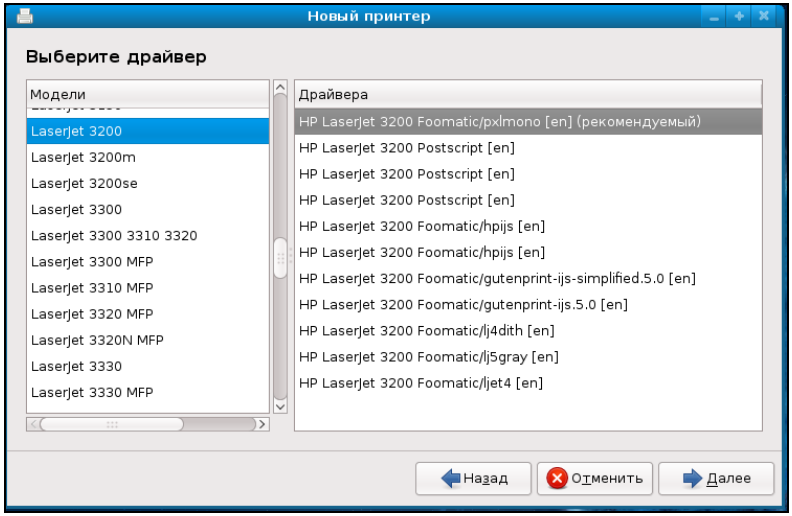


Рис. 37.6. Выбор модели принтера и драйвера

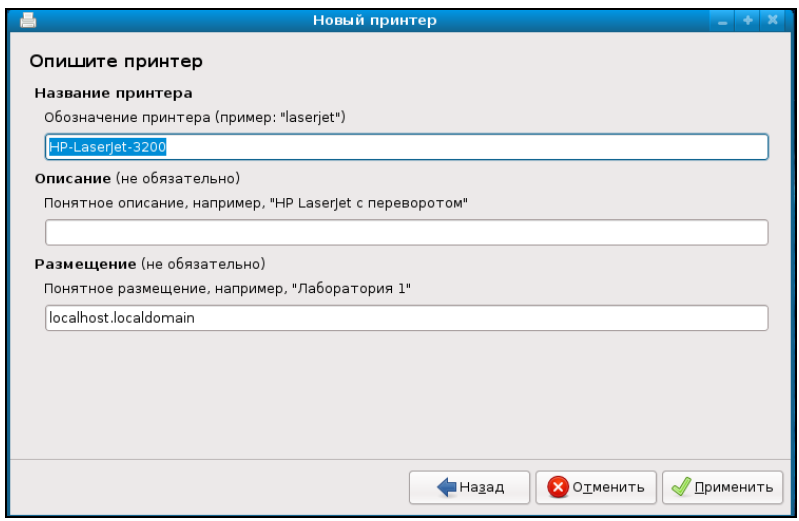


Рис. 37.7. Описание принтера

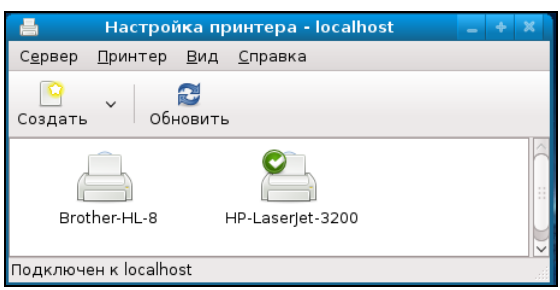


Рис. 37.8. Новый принтер установлен в системе

Система пытается определить принтер и подходящие для него драйверы. Допустим, автоматически система не может определиться с драйвером, и мы тогда получим следующее окно (рис. 37.5), где необходимо выбрать принтер из базы данных.

Сначала выберем семейство, потом модель. При наличии драйверов можно напрямую "подсунуть" их системе или взять какие-то PostScript-драйверы.

В нашем случае выбираем пункт **HP** и получаем следующее окно (рис. 37.6).

Выбираем принтер, указываем наиболее подходящий по нашему мнению драйвер и нажимаем кнопку **Далее.**

В следующем окне (рис. 37.7) мы должны задать название и описание принтера, а также компьютер, к которому он подключен. Нажимаем кнопку **Применить,** и в системе установлен новый принтер (рис. 37.8).

Ссылки

- **hpinkjet.sourceforge.net** — драйверы для принтеров Hewlett-Packard непосредственно от фирмы-производителя (поддерживается более 60 моделей).
- **61.251.162.120:8080** — драйверы для принтеров Samsung от фирмы-производителя. Поддерживаются все принтеры серии ML.
- Документация на принт-сервер Surecom.
- **http://www.linuxrsp.ru/artic/print_server.html** — Юрий Лушня. Печать в Linux с железными нервами.
- **linuxcenter.ru/lib/hardware/usbprinter.phtml** — Юрий Лушня. Настраиваем USB-принтер под Linux.
- **linux.yaroslavl.ru/Docum/Rus/print.html** — В. Толпекин. Настройка сетевого принтера для печати русского текста.
- **www.astart.com/lprng/LPRng.html** — страница проекта LPRng.
- **www.freebsd.org/~andreas/#apsfilter** — страница APSFILTER: Магический фильтр для печати.
- **metalab.unc.edu/pub/Linux/system/printing/** — lprMagic: Фильтр печати с неплохими возможностями.
- **feynman.tam.uiuc.edu/pdq/** — страница PDQ.
- **ftp://ppr-dist.trincoll.edu/pub/ppr/** — местонахождение PPR — системы буферизации печати, ориентированной на PostScript.
- **www.Linux-USB.org** — сайт, посвященный USB-устройствам и их применимости с точки зрения Linux.
- **http://www.linuxdoc.org/** — сайт, содержащий много интересной документации по Linux на английском языке.
- **www.citycat.ru/linux/docs/index.html** — сайт, содержащий много интересной документации по Linux на русском языке.
- **www.l0pht.com/~weld/netcat/** — страница netcat-пакета для работы с принтером.
- **www.penguincomputing.com/prtools/npadmin.html** — страница npadmin — программы для управления сетевыми принтерами. Управление осуществляется через SNMP.
- Linux Printing HOWTO — Mark Komarinski. Использование печати в Linux. Перевод Alex Ott.



Часть VI

Разное



Глава 38

Сканер

Поскольку в среде неискушенных пользователей бытует мнение, что операционная система Linux предназначена только для организации различных серверов, то эти пользователи и не помышляют о мультимедийном компьютере или графической станции с Linux. В следующих главах мы постараемся убедить вас, что графика, мультимедиа и Linux вполне совместимы.

Начнем, пожалуй, с организации рабочего места дизайнера. Для этого необходимо иметь следующее:

- ❒ большой, хороший монитор;
- ❒ современную видеокарту;
- ❒ сканер;
- ❒ принтер;
- ❒ графический редактор с мощными возможностями.

С монитором вообще проблем практически нет, настроить его можно именно так, как вам хочется и как позволит ваша аппаратура. То же относится и к видеокарте.

О принтерах и их настройке мы тоже знаем.

Графический редактор Gimp мало в чем уступает Photoshop, а кое в чем и превосходит его; кроме того, он дешевле.

Остается один существенный компонент — сканер. Именно поддержке сканеров в Linux и посвящена данная глава.

До последнего времени производители аппаратного обеспечения, мягко говоря, не баловали наличием драйверов для своих устройств под Linux, поэтому приходилось выходить из положения собственными силами. Если драйверы для сетевых карт, большинства видеокарт и принтеров энтузиасты всеми правдами и неправдами разрабатывали, портировали или приспосабливали уже существующие, то с драйверами для "экзотической" периферии (с точки зрения пользователя офисного компьютера или разработчика программ) — сканеров, фотокамер, плат видеозахвата — дела обстояли совсем печально.

Отголоски этих времен и до сих пор чувствительно отзываются для обычного домашнего пользователя, т. к. для многих периферийных устройств, особенно выпущенных два-три года назад, не существует драйверов или программ, способных полностью реализовать их возможности. К большому сожалению, это касается и сканеров. Для того чтобы заставить работать сканер в операционной системе Linux в настоящее время, по большому счету, существует только один программный

пакет — SANE. И, как уже упоминалось ранее, далеко не для всех сканеров есть драйверы. Такое положение объясняется не только просчетами производителей, но и разнообразием типов интерфейсов, применяемых в сканерах.

Как известно, многие современные сканеры снабжены одним из четырех (а иногда двумя из четырех) интерфейсов:

- SCSI;
- параллельный (подключаемый к порту принтера);
- USB;
- IEEE-1394.

Помимо этого, существуют сканеры, которые имеют свой оригинальный интерфейс и, соответственно, специальную интерфейсную плату, устанавливаемую в компьютер, а также сканеры, подключаемые к последовательному порту.

Неудивительно, что при отсутствии спецификаций (которые в бизнес-мире составляют коммерческую тайну) Linux-сообщество не смогло в полной мере самостоятельно создать необходимые драйверы. Еще одним тормозом в расширении применения сканеров для Linux явилось то, что еще года три назад наиболее массовым на рынке был сканер с SCSI-интерфейсом, причем для его удешевления производитель обычно комплектовал сканер SCSI-контроллером с урезанными функциями либо не совсем отвечающий SCSI-стандарту.

Впрочем, с приходом параллельного и USB-интерфейса, а также из-за того, что электронику сканеров сейчас производят пять-семь фирм, положение со сканерами в операционной системе Linux постепенно улучшается.

Начинать необходимо с выбора сканера. К сожалению, в отличие от Windows, где работает практически любой сканер, существует не так уж много моделей, поддержка которых реализована в Linux и пакетом SANE *полностью*. Значительно больше моделей сканеров, поддерживаемых лишь частично. Списки поддерживаемых Linux сканеров вы можете посмотреть на сайтах, перечень которых находится в конце главы.

В табл. 38.1 приведен список некоторых сканеров, полностью поддерживаемых Linux, причем производства только тех фирм, сканеры которых реально могут приобрести наши пользователи.

Таблица 38.1. *Список сканеров, полностью поддерживаемых Linux*

Фирма-производитель	Модель сканера	Интерфейс
Acer/Benq	Prisa 620U	USB
	Prisa 640U	
	Prisa 640BU	
	AcerScan 1240	
	AcerScan 3300	
	AcerScan 4300	
	AcerScan 5300	

Таблица 38.1 *(продолжение)*

Фирма-производитель	Модель сканера	Интерфейс
Agfa	Snapscan 1212U Snapscan 1236U Snapscan e20 Snapscan e25 Snapscan e26 Snapscan e40 Snapscan e42 Snapscan e50 Snapscan e52	USB
Canon	CanoScan FB630U CanoScan FB636U CanoScan N650U CanoScan N656U CanoScan N670U CanoScan N676U	USB
Epson	GT-7000 Perfection 610U Perfection 636U Perfection 640U Perfection 1200U/Photo Perfection 1240U/Photo Perfection 1640SU Perfection 1650/Photo Perfection 1660 Perfection 2400 Perfection 2450 Perfection 3200	USB
	Actionscanner II GT-5000 GT-6500 ES-300C ES-600C ES-1200C	Параллельный

Таблица 38.1 *(продолжение)*

Фирма-производитель	Модель сканера	Интерфейс
Epson	GT-5500 Perfection 636S ES-8500 GT-8000 GT-7000 Expression 1600 Expression 1680	SCSI
Hewlett-Packard	ScanJet 4100C ScanJet 5200C ScanJet 5300C ScanJet 6200C ScanJet 6250C	USB
Hewlett-Packard	ScanJet 6300C ScanJet 6350C ScanJet 6390C ScanJet 7400c ScanJet 7450c ScanJet 7490c	USB
Microtek	Scanmaker X6 Scanmaker 3600 Scanmaker V6 USB Scanmaker X12 USB	USB
Minolta	Scan Dual II Plug-a-Scan 600CU Plug-a-Scan 1200UB Plug-a-Scan 1200CU Plug-a-Scan 1200CU Plus	USB
Mustek Umax	600 IIIEP Plus Paragon 600 II N	Параллельный
	AstraSlim SE	USB

Таблица 38.1 *(продолжение)*

Фирма-производитель	Модель сканера	Интерфейс
Mustek Umax	Paragon MFS-6000CX Paragon MFS-12000CX Paragon MFC-600S Paragon 600 II CD ScanMagic 600 II SP Paragon MFC-800S Paragon 800 II SP Paragon MFS-6000SP Paragon MFS-8000SP Paragon MFS-1200SP Paragon MFS-12000SP ScanExpress 6000SP ScanExpress 12000SP ScanExpress 12000SP Plus	SCSI
Mustek Umax	Vista S6 Vista S6E UMAX S-6E UMAX S-6EG Vista-S8 Supervista S-12 UMAX S-12 UMAX S-12G Astra 600S Astra 610S Astra 1200S Astra 1220S Astra 2200 (SU) Astra 2400S Astra MX3 Mirage D-16L Mirage II Mirage Iise PowerLook	SCSI

Таблица 38.1 *(окончание)*

Фирма-производитель	Модель сканера	Интерфейс
Mustek Umax	PowerLook II PowerLook III PowerLook 270 PowerLook 270plus PowerLook 2000	SCSI
	Astra 6400 Astra 6450 PowerLook 1100	IEEE-1394

Настройка Linux для подключения сканера

Зачастую ядро операционной системы Linux не опознает новое USB-устройство и, соответственно, не представляет, что с ним делать. Поэтому необходимо самостоятельно определить наше устройство. Для этого в файл /etc/modules.conf следует добавить строку

```
options scanner vendor=0x04b0 product=0x100 read_timeout=8000
```

Для конкретного USB-сканера, вполне вероятно, необходимо будет подставить свои значения.

Могут возникнуть сложности из-за слишком маленького времени ожидания подтверждения в драйвере. Для решения этой проблемы придется поэкспериментировать с параметром `read_timeout`, значение которого задается в сотых долях секунды.

Параметры вашего USB-сканера можно посмотреть в log-файлах операционной системы:

```
hub.c: USB new device connect on bus1/1, assigned device number 5
usb.c: USB device 5 (vend/prod 0x4b0/0x100) is not claimed by any active driver.
/etc/hotplug/usb.agent: ... no drivers for USB product 4b8/110/110
```

Как видно из сообщения, ядро операционной системы ничего не знает о данном сканере. Чтобы решить эту проблему, в файле /etc/hotplug/usb.distmap нужно взять подходящую строчку от другого сканера этого же производителя:

```
scanner 0x0003 0x04b0 0x0107 0x000 0x000 0x00 0x00 0x00 0x00 0x00 0x00 0x00000000
```

и скопировать ее в файл /etc/hotplug/usb.handmap, заменив идентификатор устройства на `0x100`. Затем следует заново подключить сканер, и в log-файлах системы вы увидите код, приведенный в листинге 38.1.

Листинг 38.1

```
usb.c: USB disconnect on device 5
hub.c: USB new device connect on bus1/1, assigned device number 6
usb.c: USB device 6 (vend/prod 0x4b0/0x100) is not claimed by any active driv-
er.
```

```
usb.c: registered new driver usbscanner
scanner.c: probe_scanner: User specified USB scanner -- Vendor:Product —
4b0:100
scanner.c: USB Scanner support registered.
```

Есть еще один небольшой нюанс — если сканер долго не использовать, то он отключается, а модуль выгружается из памяти. В результате автоматический поиск устройства не работает. Для решения этой проблемы необходимо отключить и заново включить сканер.

Программный пакет SANE

Установленный нами для сканера драйвер ядра Linux обеспечивает только транспортный уровень протокола — передает и принимает байты, но не более того. Для работы потребуется программа, умеющая общаться именно с данной моделью сканера. Наиболее популярный комплект таких программ — пакет SANE.

SANE представляет собой интерфейс, который обеспечивает доступ к сканирующему оборудованию стандартным образом, а также библиотеку модулей для многих моделей сканеров. Поддерживаются USB- и SCSI-сканеры, сканеры, подключаемые к параллельному порту, и даже сканеры с интерфейсом FireWire (IEE-1394), а также некоторые цифровые камеры.

В дополнение к библиотеке модулей, в состав пакета входят программы для сканирования (frontends), а также ПО от других разработчиков.

> ***Замечание***
>
> Есть два понятия — frontend и backend. Frontend — программа, с которой непосредственно "общается" пользователь, обычно она имеет графический интерфейс и никогда не взаимодействует напрямую с аппаратными средствами. Backend — программа, с которой обычно работает не пользователь, а программа frontend — она передает какую-то информацию, а backend управляет аппаратурой.

Обычно практически любой дистрибутив содержит пакет SANE, однако лучше всего взять его на сайте разработчиков, поскольку пакет динамично развивается и дополняется. После установки пакета желательно отредактировать список устройств в файле /etc/sane.d/dll.conf — все лишние устройства "закомментировать".

Добавим наше устройство в файл /etc/sane.d/scanner.conf:

```
usb /dev/usb/scanner0
```

После этого протестируем список доступных устройств командой:

```
scanimage -L -v
```

Среди распознанных должно быть и наше устройство. Теперь можно посмотреть, на что оно способно:

```
scanimage --help -v --device scanner:/dev/usb/scanner0
```

Вы должны увидеть нечто, подобное приведенному в листинге 38.2.

Листинг 38.2

```
--mode Binary|Gray|Color
--depth 8|16
--halftoning
--dropout None|Red|Green|Blue
--brightness -4..3
--sharpness -2..2
--gamma-correction
--color-correction --resolution
50|60|72|75|80|90|100|120|133|144|150|160|175|180|200|216|240|266|300|320|350|
360|400|480|600|720|800|900|1200|1600|1800|2400|
--threshold 0..255
--mirror[=(yes|no)]
--speed[=(yes|no)]
--auto-area-segmentation[=(yes|no)]
--zoom 50..250
--preview[=(yes|no)]
--preview-speed[=(yes|no)]
--source Flatbed|Transparency Unit
--film-type Positive Film|Negative Film
--focus-position Focus on glass|Focus 2.5mm above glass
```

Программное обеспечение (frontend) для пакета SANE

На сайте SANE заявлено о наличии в данный момент ряда программ для сканирования с помощью SANE. Рассмотрим их поподробнее.

Xsane

Графическая программа для сканирования под X Window. Поддерживает следующие возможности:

- сканирование и просмотр изображения в формате JPEG, PNG, PNM, PS, RAW, TIFF;
- отправку отсканированного изображения по факсу с помощью специальной утилиты;
- отправку отсканированного изображения по электронной почте с помощью специальной утилиты;
- управление гамма-коррекцией;
- встраивается в качестве plug-in в GIMP;
- работает в следующих операционных системах:
  - UNIX (Linux);
  - OS/2 с X11;
  - Windows 9*x*/NT/2000/XP.

xscanimage

Программа для сканирования в среде X Window. По сравнению с Xsane имеет слишком мало возможностей:

- сохраняет сканированное изображение в файл в формате PNM;
- встраивается в качестве plug-in в GIMP.

QuiteInsane

Программа работает в среде X Window и позволяет сканировать и сохранять изображения. Базируется на библиотеке Qt.

FlScan

Программа работает в среде X Window и позволяет сканировать и сохранять изображения. Базируется на библиотеке FLTK.

scanimage

Утилита командной строки для сканирования изображений. Неудобна в использовании, зато работает в текстовом режиме.

TkScan

Как написано на сайте SANE, TkScan обладает очень приятным графическим интерфейсом, поддерживает сканеры Mustek, используя утилиту scanimage, входящую в состав SANE.

saned

Сетевой демон для удаленного сканирования. Существуют же сканеры с автоподачей оригиналов…

scanadf

Утилита командной строки, позволяющая задействовать дополнительные возможности сканеров с автоподачей оригиналов.

scanlite

Утилита для сканирования изображений, написанная на Java. В настоящее время находится в стадии тестирования.

xcam

Графическая утилита для фотокамер. Немного не по теме данной главы, но эта программа входит в SANE.

Staroffice v7/ OpenOffice 1.1

Этот офисный пакет содержит простой интерфейс для сканирования, который использует SANE.

NSane

Графическая программа по взаимодействию с SANE в NeXTStep.

Программа VueScan

Разработчики позиционируют VueScan как альтернативу SANE. Объединяет в одной программе библиотеку драйверов сканеров и графическую оболочку. Исходные тексты программы не публикуются. Распространяется как Shareware, но без оплаты не сохраняет сканированные изображения. Ориентирована на слайд-сканеры: поддерживает инфракрасный канал, фокусировку, установку времени экспозиции, пакетную обработку, многократное сканирование. Содержит специальные фильтры обработки изображений для пленки: удаление зерна, восстановление блеклых цветов.

Ссылки

- **www.bog.pp.ru** — Сергей Богомолов. Hardware: Использование USB-сканера в Linux.
- **www.digitalware.ru/static/dwscanners/** — обзор сайтов, посвященных сканерам и сканированию.
- **www.hamrick.com/vsm.html** — официальный сайт VueScan — программы для сканирования, содержащий набор драйверов для сканеров.
- **www.scaner.ru** — сайт, посвященный сканерам.
- **www.scaners.ru** — сайт, посвященный сканерам.
- **www.buzzard.me.uk/jonathan/scanners-usb.html** — список USB-сканеров, поддерживаемых SANE.
- **www.mostang.com/sane** — официальная страница пакета SANE.
- **panda.mostang.com/sane/sane-backends.html** — поддерживаемые сканеры.
- **www.qbik.ch/usb/devices/devices.php** — список USB-устройств, более или менее поддерживаемых Linux, с отзывами владельцев.
- **www.epsondevelopers.com/lscan.jsp** — страница на сайте Epson о драйверах сканеров для Linux.
- **www.xsane.org** — официальный сайт Xsane.
- **www.hamrick.com/vsm.html** — сайт программы VueScan.
- **sunsite.unc.edu/pub/Linux/apps/graphics/capture/** — месторасположение программы TkScan.
- **www.bible-mda.ru/soft/scanning/scanner-linux.html** — установка и настройка сканера в GNU/Linux на примере Epson Perfection 1270 для Debian 4 и OpenSUSE 10.2

Глава 39



Различная "экзотическая" периферия и внешние устройства

В этой главе пойдет речь о таких устройствах, которые совместно используются с компьютером относительно редко или вызывают трудности при работе с Linux. Например, карманный персональный компьютер (КПК, смартфон) или мобильный телефон с инфракрасным портом, цифровой фотоаппарат или карточки Flash-памяти. Одним словом, "экзотика", которая, тем не менее, встречается все чаще. И основная проблема — как компьютеру обмениваться информацией с этими приборами? Как обычно, производители всевозможных электронных устройств позаботились о ПО для Windows, а для альтернативных операционных систем практически ничего нет. Попробуем устранить этот недостаток и рассказать о программном обеспечении для синхронизации информации между Linux и вашими электронными новинками.

Linux и телефоны

Пожалуй, добрая треть мобильных телефонов, находящихся в эксплуатации у нашего населения, — это аппараты финской фирмы Nokia. Вы не замечали, что обыкновенная записная книжка, по крайней мере, для записи телефонных номеров, совсем исчезла? Теперь все номера находятся либо в памяти вашего мобильного телефона, либо на его же SIM-карте? А не задумывались ли вы о перспективе потери мобильного телефона или выходе его из строя? Ведь в таком случае вы потеряете все сведения, которые собирали на протяжении, наверное, целого года. Перспектива не радужная... Руками переписывать всю информацию с дисплея телефона на бумажку? Многие, наверное, уже забыли, как авторучку держать, все время на компьютере да на компьютере. Надо бы для этого компьютер и приспособить. К слову, некоторые модели мобильных телефонов (Siemens, Sony Ericsson, Motorola) могут напрямую обмениваться записями телефонов по инфракрасному порту или Bluetooth. Но вернемся к телефонам Nokia.

Здесь нам поможет замечательная программа Gammu (Gammu+) — "наследница" программы Gnokii.

Программа предназначена для работы с мобильными телефонами и смартфонами. Обеспечивает обмен данными по инфракрасному порту, Bluetooth-соединению и USB-кабелю. Поддерживает большинство современных телефонов. Позволяет синхронизировать записную книжку, использовать телефон в качестве GSM-

модема, отправлять SMS. Настройка проста и описана на сайте разработчика **http://www.mwiacek.com/www/?q=gammu**.

Еще один вариант — программа Kandy, предназначенная для синхронизации телефонной книги и адресной книги KDE.

Linux и КПК

Существует еще один класс устройств, которому не менее, а пожалуй, и в большей степени необходима синхронизация с компьютером — карманные персональные компьютеры (КПК). Большинство из них работает под управлением трех ОС:

- Palm OS (устарели);
- Symbian (Epoc OS, устарело);
- Windows CE.

Синхронизацию КПК с этими операционными средами и компьютера под управлением операционной системы Linux мы и рассмотрим далее.

Linux и Palm

КПК под управлением операционной системы Palm OS очень много — это и собственно КПК производства фирм Palm, Sony и Handspring, и множество устройств менее именитых производителей.

Чтобы соединить КПК с операционной системой Palm OS и компьютер под управлением Linux, ничего сверхординарного не нужно: два устройства, так называемый *кредл* (от англ. *cradle*, колыбель — специальная подставка с разъемом для подключения к компьютеру и подзарядки) для синхронизации или инфракрасный порт на компьютере (в КПК он уже присутствует) и программа для синхронизации компьютера и КПК.

Для комфортной работы с КПК под управлением операционной системы Palm OS есть множество программ, но все они основаны на программном пакете Pilot-Link, в котором есть все необходимое для работы с подобными КПК. Однако в большинстве случаев вам не понадобятся все возможности данного пакета, поскольку значительная часть утилит с успехом заменяется более удобной и красивой программой, работающей в X Window.

После установки программы Pilot-Link необходимо указать, к какому последовательному порту и на какой скорости подключен ваш КПК. Проще всего добавить следующие строки в файл /etc/profile:

```
export PILOTRATE=115200
export PILOTPORT=/dev/ttyS1
```

Здесь:

- `PILOTRATE` — скорость передачи данных от КПК к компьютеру. Эту скорость желательно установить как можно больше, в идеале — 115 200 бит/с. Однако, если для связи вы пользуетесь инфракрасным портом, могут возникнуть проблемы, особенно когда ваш стол с компьютером стоит возле окна, и на улице вовсю

светит солнце. Тут, как обычно, два выхода: или зашторить окно, или понизить скорость передачи информации;

- `PILOTPORT` — эта переменная указывает, к какому порту подключен кредл синхронизации с КПК.

pilot-xfer

Утилита для синхронизации КПК и компьютера в консольном режиме. Вот основные опции командной строки этой программы:

- `-b [`*`каталог`*`]` — копирует все содержимое памяти КПК в указанный каталог;
- `-u [`*`каталог`*`]` — обновляет копию памяти КПК в каталоге;
- `-s [`*`каталог`*`]` — синхронизирует каталог и память КПК;
- `-r [`*`каталог`*`]` — переносит содержимое каталога в память КПК;
- `-i` *`файлы`* — инсталлирует в КПК указанные файлы;
- `-m` *`файлы`* — инсталлирует в КПК те файлы, которых в нем нет;
- `-f` *`база`* — забирает соответствующую базу из КПК;
- `-d` *`база`* — удаляет из памяти КПК соответствующую базу.

Программы под X Window

Консольный режим хорош своим минимализмом, но иногда хочется красоты и удобства. Пойдем за ними на сайт **www.freshmeat.net**. В поле ввода поисковой системы сайта укажем слово `pilot` и получим достаточно длинный список, в котором найдется десятка полтора программ, предназначенных для работы с КПК. Рассмотрим некоторые из них.

gnome-pilot

Программа, являющаяся частью проекта GNOME, позволяет синхронизировать КПК с компьютером, устанавливать и удалять приложения, править записную книжку и т. п.

J-Pilot

Все, что написано о предыдущей программе, можно смело сказать и о программе J-Pilot. Ее внешний вид изображен на рис. 39.1.

KPilot

Программа для синхронизации КПК и компьютера, является частью проекта KDE. Внешний вид программы представлен на рис. 39.2.

Linux и PocketPC

Наиболее популярны PocketPC и смартфоны, имеющие операционную систему WindowsCE. Характерное отличие этих устройств — однообразие как в программной части, так и в аппаратной.

Программа SynCE, находящаяся на сайте **synce.sourceforge.net**, предназначена для взаимодействия с КПК на базе WindowsCE. Поддерживает синхронизацию с КПК посредством последовательного, инфракрасного и USB-соединений.

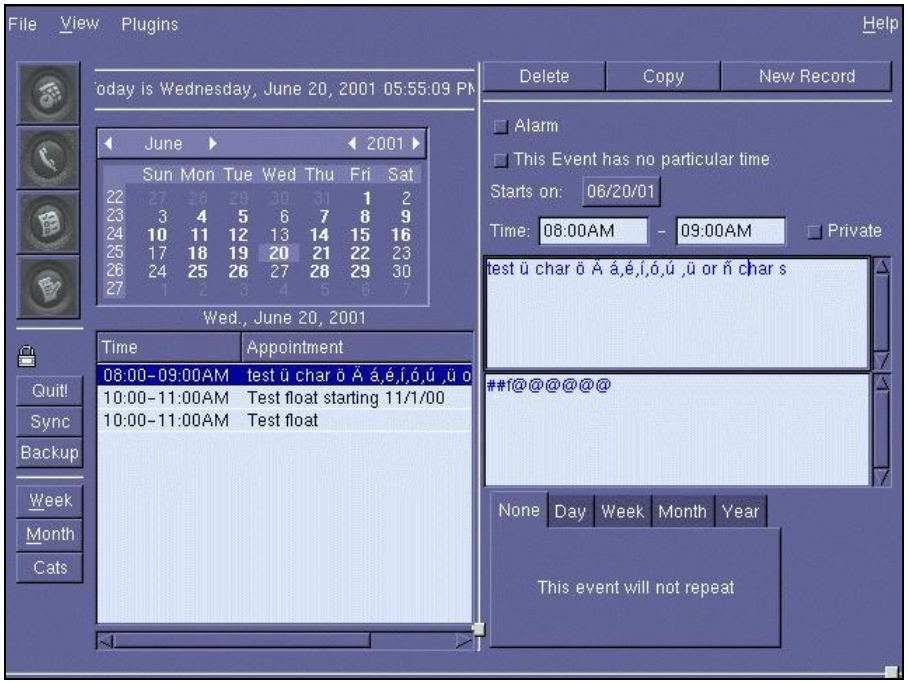


Рис. 39.1. Программа J-Pilot

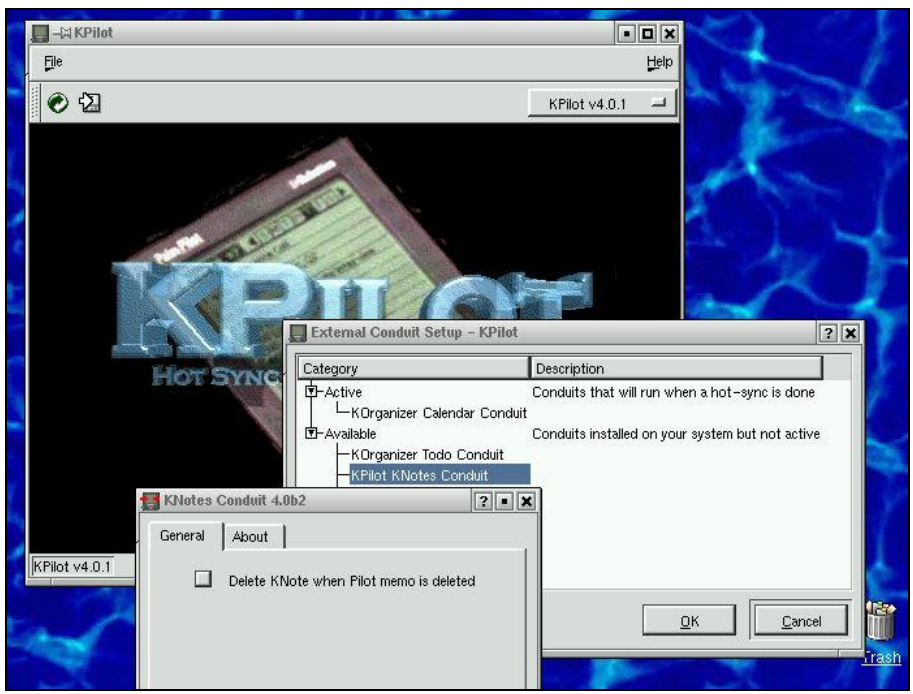


Рис. 39.2. Программа KPilot

После установки требуемых пакетов необходимо сконфигурировать соединение с устройством, а затем можно производить дальнейшие действия с данными. Подключить КПК по последовательному соединению можно командой

```
synce-serial-config ttySX
```

где `ttySX` — последовательный порт, к которому подключено устройство.

Для подключения КПК по инфракрасному соединению необходимо выполнить команду

```
synce-serial-config irdaX
```

где `irdaX` — инфракрасный порт, к которому подключено устройство.

Подключить КПК по USB-соединению можно командой

```
synce-serial-config ttyUSBX
```

где `ttyUSBX` — USB-порт, к которому подключено устройство.

После того как соединение с КПК сконфигурировано, можно подключаться к устройству, выполнив команду

```
dccm
```

Если на вход в КПК установлен пароль, например qwerty, то даем команду

```
dccm -p qwerty
```

Далее запускаем программу synce-serial-start и получаем консоль, в которой можно выполнять следующие команды:

- **pcp** — копирование файлов;
- **pls** — показывает содержимое каталога;
- **pmkdir** — создает каталог;
- **pmv** — перемещение/переименование файла;
- **prm** — удаление файла;
- **prmdir** — удаление каталога;
- **prun** — запуск программы;
- **pstatus** — отображает статус устройства;
- **synce-install-cab** — инсталлирует .cab-файл.

Также у SyncCE есть графические оболочки SyncCE-KDE и SyncCE-GNOME. Для синхронизации календаря и записей почты и прочего нужно воспользоваться программой Multisync (**http://multisync.sourceforge.net**).

Linux и TV-тюнер

Помимо различных устройств, предназначенных для работы, к компьютеру можно подключить и средства для развлечений. Одно из них — плата телевизионного приемника (TV-тюнер), зачастую совмещенная с радиоприемником. Для нормального функционирования таких устройств необходимы две вещи — драйверы (и корректная их настройка) и соответствующие программы для просмотра телевизионных передач. Начнем с настройки драйверов.

Ключевое слово для нас при поиске информации — video4linux или video4linux2 — набор драйверов и документации для обеспечения поддержки видео под Linux.

Как правило, практически все платы TV-тюнеров основаны на одной из микросхем:

- ❒ Bt848;
- ❒ Bt848a;
- ❒ Bt849/Bt878;
- ❒ Bt879.

Для обеспечения работы TV-тюнера необходимо выполнить несколько условий.

1. Иметь настроенную звуковую карту, поскольку звук с TV-тюнера передается по кабелю на вход звуковой карты.
2. Скомпилировать ядро операционной системы Linux с поддержкой следующих функций:
   - установить поддержку ядром операционной системы драйвера bttv;
   - установить поддержку I2C-интерфейса;
   - установить поддержку I2C bit-banging-интерфейса;
   - в секции Multimedia Devices включить поддержку Video For Linux и установить BT8XX Video For Linux как модуль.
3. Проверить наличие устройства /dev/video* и, если таковое отсутствует, создать его следующими командами:
   - `cd /dev`;
   - `./MAKEDEV video`.
4. После компиляции и установки ядра операционной системы Linux и модулей перезагрузить компьютер и выполнить команду:
   ```
   /sbin/insmod bttv
   ```
5. В документации на вашу плату найдите точное ее название, а в описании драйвера bttv выясните, поддерживает ли драйвер эту карту.

После выполнения этих условий можно переходить к настройке платы TV-тюнера.

Для проверки работоспособности платы TV-тюнера, а также для подборки параметров для драйвера потребуется какое-то программное обеспечение, способное работать с video4linux. Пожалуй, оптимальный вариант — применить программу xawtv, которая примечательна тем, что использует библиотеку libXaw и не требует никаких дополнительных специфических библиотек для компиляции.

Самый простой вариант заставить вашу плату TV-тюнера функционировать — попытаться, чтобы программное обеспечение самостоятельно определило ее тип. К сожалению, такая удача бывает не часто, поэтому нам ничего не остается, как внимательно изучить документацию, сопровождающую плату TV-тюнера. Находим точное название платы и фирму-производителя и ищем это устройство в списке поддерживаемых драйвером bttv TV-тюнеров, где и определяется необходимый нам номер карты TV-тюнера. Если же ваше устройство в указанном списке отсутствует, остается только одно — настройка методом подбора.

Для этих целей воспользуемся программой xawtv. После установки программы нам необходимо подправить конфигурационный файл .xawtv.

Небольшое отступление — у нас принят стандарт телевизионного вещания SECAM D/K, поэтому при редактировании конфигурационного файла .xawtv задан тип кодировки SECAM. Помимо этого, при настройке драйвера bttv необходимо выставить переменную `tuner type`. Для большинства плат TV-тюнеров и стандарта SECAM под-

ходит `tuner type=3`. Однако для некоторых разновидностей плат переменной `tuner type` нужно присвоить значение 1 или 5.

Конфигурационный файл .xawtv приведен в листинге 39.1.

Листинг 39.1

```
[global]
 fullscreen = 800 x 600
 freqtab = europe-east
 pixsize = 128 x 96
 pixcols = 1
 jpeg-quality = 75

 [defaults]
 norm = SECAM
 capture = over
 source = Television
```

В этом файле мы определили размер изображения, частотную таблицу каналов, качество jpeg-сжатия, стандарт телевизионного изображения и источник сигнала.

Теперь необходимо подобрать для драйвера bttv номер типа TV-тюнера, при котором наша плата будет нормально функционировать.

Алгоритм подбора следующий:

1. Устанавливаем модуль ядра операционной системы, поддерживающий I2C:
   ```
   modprobe i2c
   ```
2. Устанавливаем модуль ядра операционной системы, поддерживающий стандарт SECAM:
   ```
   modprobe tuner type=3
   ```
3. Устанавливаем модуль ядра операционной системы с драйвером bttv и типом карты TV-тюнера, равным 1:
   ```
   modprobe bttv card=1
   ```
4. Затем запускаем программу xawtv:
   ```
   xawtv &
   ```
5. Далее, с помощью клавиш со стрелками вверх-вниз находим телевизионный канал, а с помощью клавиш со стрелками вправо-влево выполняем точную подстройку.

   Проверяем изображение и звук. Если телепередача не выводится нормально — черно-белое изображение, нет звука или вообще ничего не видно и не слышно, то выполняем команду `q` в окне xawtv и следующие действия:
   - выгружаем драйвер bttv:
     ```
     rmmod bttv
     ```
   - меняем тип карты TV-тюнера:
     ```
     modprobe bttv card=2
     ```
6. Повторяем пп. 4–5 до тех пор, пока не добьемся результата.

Однако в этот простой алгоритм могут добавиться еще кое-какие действия. Некоторые платы TV-тюнеров имеют в своем составе *отдельный* декодер звука (обычно микросхемы msp34xx, tda8425, tea6300). В этом случае необходимо дополнительно загружать соответствующие модули (предварительно их нужно скомпилировать).

Предположим, все прошло успешно, и вы определили параметры, с которыми нужно загружать модули ядра, относящиеся к плате TV-тюнера. Теперь нам необходимо сделать так, чтобы эти модули автоматически загружались при старте операционной системы. Для этого в файл /etc/conf.modules следует добавить строки листинга 39.2.

Листинг 39.2

```
alias char-major-81-0 bttv
alias char-major-81 videodev
options tuner type=3
options bttv card=8
pre-install bttv modprobe -k tuner
```

Перезагружаем компьютер, запускаем опять программу xawtv и проверяем функционирование платы TV-тюнера. В случае успеха можно переходить к программам, функционирующим под управлением X Window.

wmtv

Программа интересна тем, что может в свернутом виде выводить изображение. Много места не занимает, а в тот момент, когда идет что-то интересное — просто делаем на минимизированной программе двойной щелчок мышью и получаем увеличенное изображение. Программу можно настроить так, что по двойному щелчку она будет вызывать внешнее приложение, например тот же xawtv.

kWinTV

Удобная, красивая и функциональная программа для просмотра телепередач под KDE (рис. 39.3).



Рис. 39.3. Программа kWinTV

LIRC

LIRC (Linux Infrared Remote Control, программное обеспечение для управления устройством с помощью дистанционного пульта). Поскольку проект развивающийся, рекомендуется перед установкой скачать самую последнюю версию программного обеспечения с сайта разработчиков. Ознакомьтесь со списком поддерживаемых устройств и документацией, поскольку вполне вероятно, что вам придется вносить изменения в драйвер bttv. В дистрибутиве LIRC содержатся примеры конфигурационных файлов для поддерживаемых устройств.

Поддержка управления с дистанционного пульта есть, например, в том же kWinTV.

Создание Real Video под Linux

Имея в составе компьютера плату TV-тюнера, можно получить с нее видеоизображение и закодировать его в формате Real Video, а при желании даже организовать видеовещание через локальную сеть или в Интернете.

Для организации видеозахвата и кодирования видеоинформации в формате Real Video необходимо выполнить следующие действия:

1. Получить Real Producer Basic с сайта **www.real.com**.
2. После инсталляции зайти в систему как пользователь root, перейти в каталог, где установлен real producer, и выполнить команды

```
realproducer -o /tmp/testing.rm -t 7 -a 3 -v 0 -f 0 -b "Testing Video" -h
"localhost" -c "Personal" -vc video -l 2:1,8:1
```

Таким образом, вы захватили видеопоток с TV-тюнера, перекодировали его в Real Player 8 и записали в каталог /tmp как testing.rm.

Этот простой пример показывает, как несложно получить телепередачу, записанную в формате Real Video. Чтобы узнать обо всех возможностях Real Producer Basic, ознакомьтесь с документацией, поставляемой в комплекте с этим программным обеспечением.

Для организации трансляции видеопотока по сети необходимо на том же сайте получить Real Server и установить его в системе. К сожалению, Real Server — коммерческая программа, бесплатно ею можно пользоваться только ограниченное время.

Возможная альтернатива — программный пакет ffmpeg — очень быстрый audio/video кодировщик/преобразователь, а также потоковый сервер (видео, аудио).

Пакет SANE

В состав пакета SANE, который предназначен для работы со сканерами, входит модуль для захвата изображений с video4linux, который работает с платами TV-тюнера.

Видеокарта с TV-out

Теперь давайте разберемся с выводом видеоизображения, например на видеомагнитофон. В последнее время стандартной видеокартой для домашнего компьютера стали изделия на базе чипов от nVidia. Существует много моделей видеокарт,

которые, помимо своих прямых обязанностей по выводу изображения на монитор, также осуществляют вывод изображения на телевизор, а иногда и захват видеоизображения. Поэтому дальнейший рассказ будет касаться видеокарт, основанных на чипах nVidia. Всю нужную информацию по настройке видеокарты можно получить из документации, идущей в комплекте с фирменными драйверами от nVidia.

Для настройки TV-out необходимо выполнить следующие действия:

1. С сайта nVidia берем последние драйверы и устанавливаем их.
2. В файле /etc/X11/XF86Config-4 приводим данные к виду, показанному в листинге 39.3.

Листинг 39.3

```
Section "Module"
    Load  "dbe"
    Load  "glx"
    Load  "extmod"
    Load  "type1"
    Load  "freetype"
EndSection

Section "Device"
    Identifier  "NVIDIA GeForce2 DDR"
    VendorName  "nvidia"
    BoardName   "ABIT"
    Driver      "nvidia"
    VideoRam    32768
    Option      "DPMS"
    # запустите 'lspci' чтобы узнать BusID
    BusID       "PCI:1:0:0"

    # Если при переключении из консоли в X Window случаются
    # падения X'ов,
    # измените 3 на 1.

    Option "NvAGP" 3
    Option "ConnectedMonitor" "TV"

    # SVIDEO или COMPOSITE — в зависимости от того, каким образом
    # подключен TV к видеокарте, через svideo-разъем
    # или разъемом типа "тюльпан" (COMPOSITE)
    Option  "TVOutFormat" "COMPOSITE"

    # Описываем частотные характеристики телевизора.
```

```
    Option  "SecondMonitorHorizSync" "30-50"
    Option  "SecondMonitorVertRefresh" "60"

    # Какой телевизионный стандарт использовать для вывода
    # изображения, как следует выбирать либо PAL-I, либо NTSC-J
    Option  "TVStandard" "PAL-I"
    # Включаем режим TwinView
    Option  "TwinView"

    # Clone — дублирование на TV изображения с экрана монитора
    # можно использовать также "RightOf" "LeftOf" "Above" "Below",
    Option  "TwinViewOrientation" "Clone"
    # Сопоставляем частоту монитора и телевизора.
    Option "MetaModes" "1024x768,640x480; 1024x769,640x480; 800x600,
640x480; 640x480,640x480"

    # Показываем, что подключили TV, а не второй монитор.
    Option "ConnectedMonitor"  "crt,tv"
EndSection
```

3. Перезапускаем X Window.

Вот, собственно, и все, можно смотреть фильмы в формате AVI или MPEG4 прямо на экране телевизора.

Цифровые фотокамеры

Несомненным преимуществом этой технологии является быстрое перемещение фотографий на компьютер, минуя стадии печати фотоизображений и их сканирования. Для работы с цифровыми фотоаппаратами существует программа xcam, входящая в комплект пакета SANE. Кроме того, практически все современные фотоаппараты имеют USB-интерфейс и для операционной системы представляют собой устройство стандарта Mass Storage. О работе с такого типа устройствами — в следующем разделе.

USB Flash-накопители, картридеры

В последние годы получили массовое распространение так называемые USB Flash-накопители — относительно дешевые устройства небольшого размера с объемом перезаписываемой памяти от 1 до 32 Гбайт. Вот уже несколько лет BIOS материнских плат допускает загрузку с таких устройств. Существует даже несколько дистрибутивов Linux, рассчитанных на загрузку и работу именно с такими накопителями. USB-накопители Linux определяет как SCSI-устройства, правда, странно? Точно так же, как и пишущий CD-ROM. Причина в том, что SCSI-идеология с мини-

мальными доработками подходит и для этих устройств, поэтому разработчики ядра себе просто упростили жизнь.

Итак, подключаем USB-накопитель в порт. Определяем, как называется подключенное устройство. Это можно сделать несколькими способами, простейший — заглянуть в log-файл и прочитать, какое именно устройство только что было подключено. Еще один вариант — воспользоваться программой hwbrowser. Для обычного компьютера безо всякой экзотики Flash-накопитель (картридер) определяется как sda1, который обычно отформатирован с файловой системой FAT16. Что дальше? Создаем точку монтирования (если ее еще нет), например /mnt/flash, и с помощью команды `mount` монтируем устройство. Если у устройства тип файловой системы FAT, команда будет иметь следующий вид:

```
Mount -t vfat /dev/sda1 /mnt/flash -o iocharset=koi8-r,codepage=866
```

Современные дистрибутивы автоматически монтируют подключенные USB-устройства без дополнительных хлопот.

Спутниковый Интернет

Нужно купить антенну, конвертер, немного кабеля и карту типа SkyStar1(2). И, конечно, установить драйверы для этой карты. Давайте посмотрим, реально ли это? Оказывается, вполне реально. Чтобы ознакомиться с этой темой — зайдите на сайт **www.gs.ru** в раздел спутникового Интернета и походите по ссылкам. Там же есть несколько статей, поясняющих, как установить и настроить карту типа SkyStar1.

UPS (источники бесперебойного питания)

Для людей, впервые столкнувшихся с такой нужной "железкой", рекомендую обратить внимание на комплектацию и документацию — в комплект поставки входит специальный кабель для подключения к компьютеру (более дорогие модели могут иметь полноценное подключение к локальной сети) и компакт-диск с программным обеспечением для Windows. Для чего необходимо подключение к компьютеру? Простейший бесперебойник выдает компьютеру следующие сигналы:

- On Battery — электропитание не в норме, UPS перешел на работу от аккумулятора;
- Low Battery — аккумулятор почти разряжен, и через 1–2 минуты UPS прекратит подачу энергии;
- Kill power — UPS прекращает подачу энергии.

Более дорогие устройства позволяют собирать и получать подробнейшую информацию о состоянии электропитания, потребляемой мощности, заряде аккумуляторов, температурном режиме и др.

Существуют несколько программ для работы с источниками бесперебойного питания: apcupsd, smartups, smartupstools, smupsd, upsd, NUT.

Для простейшего прибора фирмы APC Back UPS 500 проще всего воспользоваться пакетом apcupsd. Установка пакета тривиальна. После инсталляции необходимо отредактировать конфигурационный файл /etc/apcupsd/apcupsd.conf.

В листинге 39.4 приведен пример конфигурационного файла.

Листинг 39.4

```
UPSNAME APC Back 500
UPSCABLE 940-0020C
UPSTYPE dumb
DEVICE /dev/ttyS1
TIMEOUT 900
NETSERVER off
EVENTSFILE /var/log/apcupsd.events
UPSCLASS standalone
UPSMODE disable
STATTIME 1
STATFILE /var/log/apcupsd.status
```

Перечисленные программы позволяют записывать информацию о событиях, связанных с электропитанием, и управлять компьютером; при получении сигнала Low Battery программа выдает команду на выключение компьютера.

Для большой локальной сети интересен пакет NUT (Network UPS Tools), который позволяет собирать информацию по сети от разных компьютеров, строить графики и дистанционно управлять бесперебойниками.

Ссылки

- **fero.koli.kando.hu/rivatv/** — описание настройки TV-out для видеокарт на чипах nVidia.
- **ftp://ftp.cs.unm.edu/mirrors/kde/unstable/apps/utils/** — утилита kpsion для связи с КПК Psion.
- **ftp://ftp.to.com/pub/psion/** — утилиты plptools для связи с КПК Psion.
- **ftp://ryeham.ee.ryerson.ca/pub/PalmOS/** — местонахождение утилиты Pilot-Link.
- **gazette.linux.ru.net/lg62/articles/rus-silva.html** — Anderson Silva. Видео-приложения на вашем Linux. Перевод Дмитрия Попкова.
- **huizen.dds.nl/~frodol/psiconv/** — официальная страница утилиты Psiconv.
- **jpilot.org** — сайт проекта J-Pilot.
- **linuxtv.org** — сайт, посвященный телевидению и Linux.
- **palm.opennet.ru/base/X/tv_out.txt.html** — пример настройки видеокарты nVidia с TV-out (linux tv video).
- **www.cadsoft.de/people/kls/vdr/index.htm** — организация Video Disk Recorder на базе компьютера, платы SkyStar1 и Linux.
- **www.mwiacek.com/www/?q=gammu** — официальная страница проекта Gammu.

- **www.gnome.org/projects/gnome-pilot/** — официальная страница пакета gnome-pilot.
- **www.in-berlin.de/User/kraxel/xawtv.html** — официальная страница программы xawtv.
- **www.linuxphone.ru** — сайт, посвященный мобильным телефонам под управлением Linux.
- **www.linux.opennet.ru/base/X/video_out.txt.html** — пример TwinView для nVidia-карт, редактирование и запись видео (linux tv video).
- **www.linuxdvb.tv** — сайт, посвященный драйверам для карт спутникового телевидения.
- **www.lirc.org** — страничка проекта LIRC (Linux Infrared Remote Control).
- **www.mainconcept.com** — сайт программного обеспечения для редактирования видео.
- **www.mathematik.uni-kl.de/~wenk/kwintv** — официальная страница программы kWinTV.
- **www.medsyn.fr/perso/g.delafond/psilin/psiolinu.htm** — страница пакета PsiLin — программы для связи с КПК Psion.
- **synce.sourceforge.net** — страница проекта The SynCE Project — программы для связи с КПК PocketPC.
- **www.nvidia.com** — сайт фирмы nVidia.
- **www.real.com** — Real Producer Basic.
- **www.slac.com/pilone/kpilot_home/** — официальная страница пакета KPilot.
- **www.strusel007.de/Linux/bttv/** — драйверы для чипов BT8XX.
- **www.stud.uni-hamburg.de/users/lennart/projects/atitvout/** — описание настройки TV-out и программного обеспечения для видеокарт на чипах ATI.
- **www.student.uwa.edu.au/~wliang** — официальная страница программы wmtv.
- **www.thp.uni-koeln.de/~rjkm/linux/bttv.html** — драйверы bttv.
- **http://linux.webclub.ru/adm/palm_pilot.html** — Вячеслав Калошин. Линукс и PalmPilot.
- Соответствующие HOWTO:
  - bttv mini-HOWTO — Владимир Бормотов, Алексей Дец;
  - Linux and Psion HOWTO.

Глава 40



Эмуляторы

Как бы мы ни старались, а полностью жить в операционной системе Linux в современном мире не получается. Так сложилось, что множество программ написаны под операционные системы MS Windows или даже DOS. И зачастую по тем или иным причинам эти программы незаменимы. Поэтому данную проблему необходимо решать — вариант "или-или" нам не подходит.

Существует тривиальное решение этого вопроса — установить две или три операционные системы и перезапускать компьютер, когда необходимо поработать в Windows. Но наверняка такой вариант мало кому понравится — неоптимальное расходование дискового пространства, постоянные перезагрузки компьютера и, как следствие, непроизводительные затраты времени. Поэтому желательно решить эту проблему по-другому.

Сосуществование двух или более операционных систем необходимо рассматривать комплексно. При взаимодействии разных ОС приходится решать несколько задач.

1. *Передача файлов из одной операционной системы в другую.* Задача решается созданием протоколов передачи файлов и информации, которыми на данном этапе являются протоколы Интернета.
2. *Работа с дисками и данными другой операционной системы*. Это условие неплохо решено для операционной системы Linux — она поддерживает файловые системы FAT, VFAT, NTFS и т. п. Со стороны операционных систем производства Microsoft все выглядит намного хуже — поддерживаются только файловые системы, разработанные Microsoft.
3. *Выполнение программ, созданных для другой операционной системы*. Для решения этой задачи есть два подхода — разработка виртуальных машин и создание эмуляторов операционной системы:
   - *виртуальные машины* позволяют создать внутри операционной системы "виртуальный" компьютер, на котором и выполняется альтернативная операционная система и ее приложения. Для виртуальной машины все равно, какая ОС будет установлена внутри нее, поскольку она обеспечивает псевдокомпьютер, на который и устанавливается операционная система. У этого подхода есть недостатки — мы вынуждены устанавливать на виртуальную машину альтернативную операционную систему и программное обеспечение, что не всегда возможно с точки зрения ресурсов и финансов. Несомненное достоинство виртуальной машины — полная эмуляция компьютера, и как следствие — возможность полноценной работы альтернативной ОС;

- *эмуляторы* призваны сформировать для программ альтернативной операционной системы "нормальную среду обитания". Но для сложной ОС практически невозможно создать нормально функционирующий эмулятор, особенно для закрытых коммерческих систем.

4. *Работа с файлами, созданными в форматах программ другой операционной системы*. Это условие выполняется при наличии либо эквивалентного программного обеспечения для другой операционной системы, либо доступной документации на формат необходимых файлов.

   Четвертое условие можно решить, выполнив третье.

Как видите, список небольшой, но охватывающий множество проблем сосуществования операционных систем.

Эмуляторы

Начнем описание с эмуляторов, поскольку исторически в операционной системе Linux они появились раньше, чем виртуальные машины. Так что, пойдем от простого к сложному.

DOSEmu

Эмулятор однозадачной, однопользовательской операционной системы MS-DOS. Вы скажете, что в эпоху развитой Windows эмулятор MS-DOS не актуален, и будете неправы. Еще много программ, написанных под MS-DOS, находится в эксплуатации. Различные учетные, складские программы, программы отделов кадров и т. п. спокойно трудятся на своих рабочих местах. Достаточно много есть и хороших игр, написанных под MS-DOS, к примеру, WarCraft II, Doom и Dune II. В свое время много специфических аппаратно-программных комплексов было разработано под MS-DOS, устройства эти эксплуатируются и по сегодняшний день.

Установка пакета DOSEmu несложна, поскольку данный пакет обычно входит в состав дистрибутива операционной системы. Перейдем сразу к конфигурированию этого эмулятора DOS.

Конфигурирование DOSEmu

Пакет DOSEmu не отличается особой оригинальностью — конфигурационный файл называется dosemu.conf и находится в каталоге /etc. Помимо этого, каждый пользователь может создать в своем домашнем каталоге файл .dosrc, в котором можно откорректировать некоторые настройки DOSEmu для данного пользователя. Поведение эмулятора также можно изменить, используя параметры запуска.

На самом деле все записи в файле в dosemu.conf — это просто переменные, которые в последующем используются в /var/lib/dosemu/global.conf и имеют вид:

```
$_xxx = (n)
```

или

```
$_zzz = "s"
```

Описание параметров конфигурации сгруппировано по исполняемым функциям.

Управление отладочной информацией

Для включения вывода отладочной информации DOSEmu необходимо в конфигурационный файл добавить строку

```
$_debug = "-a"
```

которая содержит то, что обычно передается через ключ командной строки `'-a'`.

Отладочная информация будет выводиться в файл, определенный опциями`'-o file'` либо `'-O'` (в последнем случае выводит в stderr).

Основные параметры

- `$_timint = (on|off)` — разрешает или запрещает прерывание таймера INT08.
- `$_mathco = (on|off)` — позволяет либо запрещает задачам DOS использовать математический сопроцессор.
- `$_cpu = (80386)` — определяет, какой тип процессора эмулировать. Можно установить тип процессора не выше существующего в компьютере. Разрешенные значения: `80386`, `80486` и `80586`.
- `$_rdtsc = (on)` — разрешает или запрещает DOSEmu задействовать счетчик циклов Pentium для лучшей обработки временных интервалов.

Для использования `'rdtsc'` DOSEmu необходимо выставить точную тактовую частоту процессора. Обычно она определяется автоматически, но в случае ошибок можно задать ее явно:

```
$_cpuspeed = (166.666).
```

- `$_pci = (on)` — разрешает DOSEmu доступ к конфигурированию PCI-устройств.

Параметры, приведенные в листинге 40.1, позволяют задать распределение оперативной памяти, которая доступна для DOS.

Листинг 40.1

```
$_xms = (1024)
$_ems = (1024)
$_ems_frame = (0xe000)
$_dpmi = (off)
$_dosmem = (640)
```

Параметр, приведенный в листинге 40.2, определяет стиль поведения DOSEmu по отношению к процессорному времени, используемому DOSEmu.

Листинг 40.2

```
$_hogthreshold = (1) # 0 — максимум процессорного времени для
                       DOSEMU
                     # 1 — максимум процессорного времени для
                       Linux
                     # >1 чем больше, тем меньше процессорного
                       времени
                     #    для DOSEMU
```

Если на вашем компьютере установлено нестандартное оборудование, для которого отсутствует Linux-драйвер, но существует DOS-драйвер, часто необходимо разрешить соответствующие IRQ в DOS:

```
$_irqpassing = ""  # список номеров IRQ (2-15) для передачи DOS
```

Параметр

```
$_speaker = ""     # or "native" or "emulated"
```

определяет, как будет работать встроенный динамик.

При помощи следующей строки параметров можно получить управление реальными портами компьютера, но с точки зрения безопасности этого делать ни в коем случае нельзя:

```
$_ports = ""  # список портов, например "0x1ce 0x1cf 0x238"
```

Терминалы

Этот раздел предназначен для DOSEmu, выполняемой на удаленном компьютере или в графическом терминале xterm.

- `$_term_char_set = ""` — определяет набор шрифтов.
- `$_term_color = (on)` — разрешает цвет.
- `$_term_updfreq = (4)` — задает интервал между обновлениями экрана в 1/20 секунды.

Установки клавиатуры

При запуске DOSEmu из консоли или X Window может понадобиться задать подходящую раскладку клавиатуры. Это делается либо выбором одной из внутренних таблиц клавиатуры, либо загрузкой внешней таблицы.

- `$_layout = "name"` — определяет внутреннюю таблицу клавиатуры.
- `$_X_keycode = (on)` — используется для сосуществования с X Window, поскольку по умолчанию устанавливается нейтральная (US) клавиатура.
- `$_rawkeyboard = (1)` — позволяет получить прямой доступ к клавиатуре для DOS-программ. Обычно это необходимо для игр.
- `$_keybint = (on)` — улучшает обработку прерывания клавиатуры.
- `$_escchar = (30)` — определяет символ ESC.

Поддержка X Window

Для запуска DOSEmu в собственном окне X Window необходимо установить некоторые приведенные далее переменные.

- `$_X_updfreq = (5)` — интервал обновления изображения в 1/20 секунды.
- `$_X_title = "DOS in a BOX"` — заголовок окна программы.
- `$_X_icon_name = "xdos"` — текст значка.
- `$_X_keycode = (off)` — разрешает трансляцию клавиатурных кодов через таблицы DOSEmu.
- `$_X_blinkrate = (8)` — частота мерцания курсора.
- `$_X_font = ""` — тип шрифта для DOS-программы.
- `$_X_mitshm = (on)` — разрешает использование разделяемой памяти.
- `$_X_sharecmap = (off)` — задает системную палитру.
- `$_X_fixed_aspect = (on)` — разрешает пропорциональное изменение размеров окна.

- `$_X_aspect_43 = (on)` — назначает отношение сторон окна 4:3 в графике.
- `$_X_winsize = ""` — начальные размеры окна.
- `$_X_gamma = (1.0)` — коэффициент гамма-коррекции.
- `$_X_vgaemu_memsize = (1024)` — размер фрейм-буфера для эмуляции VGA в килобайтах.
- `$_X_lfb = (on)` — разрешает линейный фрейм-буфер для VESA-режимов.

Видеоустановки для консоли

За конфигурирование DOSEmu для работы в консольном режиме отвечают следующие параметры.

- `$_video = "vga"` — позволяет выбрать тип видеокарты.
- `$_console = (0)` — разрешает или запрещает использование видео на консоли.
- `$_graphics = (0)` — разрешает применять BIOS видеокарты для установки видеорежима.
- `$_videoportaccess = (1)` — разрешает доступ к видеопорту в графических режимах.
- `$_vbios_seg = (0xc000)` — адрес видео-BIOS.
- `$_vbios_size = (0x10000)` — размер видео-BIOS.
- `$_vmemsize = (1024)` — размер буфера регенерации.
- `$_chipset = ""` — чипсет видеокарты для лучшего взаимодействия с ней.

Диски и дискеты

Следующие переменные определяют наличие дисковода и параметры жесткого диска.

- `$_vbootfloppy = ""` — имя файла виртуальной дискеты, с которой будет производиться загрузка.
- `$_floppy_a ="threeinch"` — тип и наличие дисковода A.
- `$_floppy_b = ""` — тип и наличие дисковода B.
- `$_hdimage = "hdimage.first"` — имя файла, содержащего список образов жесткого диска в /var/lib/dosemu.

При установке DOSEmu в файл /var/lib/dosemu/hdimage.first записывается образ загрузочного диска. Это файл, содержащий виртуальный образ файловой системы DOS — FAT.

Вместо загрузки с виртуального диска можно загрузиться с виртуальной дискеты командой

```
'dd if=/dev/fd0 of=floppy_image'
```

Если это загрузочная дискета DOS, то при установке параметра

```
$_vbootfloppy = "floppy_image"
```

система будет загружаться с этой виртуальной дискеты.

COM-порты

Приведенные далее параметры определяют COM-порты и работающие с ними устройства.

- `$_com1 = "/dev/mouse"` — определяет, какое устройство Linux соответствует порту COM1.

- `$_com2 = "/dev/modem"` — определяет, какое устройство Linux соответствует порту COM2.
- `$_mouse = "microsoft"` — тип используемой мыши.
- `$_mouse_dev = "/dev/mouse"` — драйвер мыши.
- `$_mouse_flags = ""` — с помощью этого параметра можно установить специальные управляющие флаги.
- `$_mouse_baud = (0)` — скорость обмена информацией с мышью, 0 — не устанавливать.

Принтеры

Принтер эмулируется передачей печатаемых данных на обычный Linux-принтер. Параметры указывают DOSEmu, какой из принтеров задействовать.

- `$_printer = "lp"` — имя Linux-принтера, который будет называться LPT1.
- `$_printer_timeout = (20)` — задержка перед началом печати.

Работа с сетью IPX/SPX

Следующие два параметра служат для поддержки сетевого протокола IPX/SPX, при этом ядро операционной системы также должно быть сконфигурировано с поддержкой протокола IPX.

- `$_ipxsupport = (on)` — разрешает протокол IPX/SPX.
- `$_vnet = (on)` — параметр необходим при установленном драйвере dosnet.

Звук

Для поддержки звуковой карты DOSEmu средствами звуковой подсистемы Linux необходимо установить следующие параметры.

- `$_sound = (off)` — разрешает или запрещает поддержку звука.
- `$_sb_base = (0x220)` — базовый адрес портов ввода/вывода звуковой карты.
- `$_sb_irq = (5)` — назначает прерывание звуковой карте.
- `$_sb_dma = (1)` — канал DMA для звуковой карты.
- `$_sb_dsp = "/dev/dsp"` — используемое звуковое устройство.
- `$_sb_mixer = "/dev/mixer"` — назначает микшер.
- `$_mpu_base = "0x330"` — базовый адрес MPU-401.

Приложения DEXE

Непосредственно исполняемые DOS-приложения DOSEmu (DEXE) — достаточно оригинальная концепция. На самом деле — это загружаемый образ диска, содержащий одно DOS-приложение. Достоинства такого типа приложений — они имеют доступ только к образу диска, и следовательно, порождают меньше проблем с безопасностью. Помимо этого, вам не придется инсталлировать и настраивать DOS-приложения.

Для создания приложения формата DEXE нужно иметь:

- пакет mtools;
- скомпилированный DOSEmu;
- zip-архив, содержащий все файлы, относящиеся к DOS-приложению;

- следующую информацию перед запуском mkdexe:
  - размер раздела для образа диска;
  - версию DOS, которую следует поместить на этот образ;
  - содержимое файлов Config.sys и Autoexec.bat.

После этого можно приступать к созданию приложения. Необходимо зайти в систему как пользователь root и выполнить команду

```
mkdexe myapp.zip -x myapp.exe -o confirm
```

Если все прошло нормально, то у вас появится файл myapp.exe, который можно запустить на выполнение командой

```
dos -L myapp.exe [ dosemu-options ]
```

либо

```
dosexec myapp.exe [ dosemu-options ]
```

Wine

Wine (Wine Is Not an Emulator) — эмулятор операционной системы Windows разных версий. Обеспечивает запуск некоторых Windows-приложений под X Window.

На сегодняшний день выпущена версия 1.2.2, которая позволяет запускать большинство Windows-программ. Программа интенсивно развивается, поэтому рекомендуется перед установкой получить самую свежую версию Wine с сайта разработчиков. Процесс установки несложен и подробно описан в документации.

Для работы с приложением Windows необходимо в Xterm запустить Wine с параметрами командной строки. При запуске Wine без параметров появится строка формата запуска.

Самый простой вариант вызова программы, написанной для Windows, — набрать следующую строку:

```
wine имя_программы.exe
```

Можно указать, для какой версии Windows написана запускаемая программа, например:

```
wine winver win98 имя_программы.exe
```

Если программа требует каких-либо библиотек, их подключение также можно задать в строке запуска, например:

```
wine winver win95 dll a.dll b.dll c.dll имя_программы.exe
```

На базе Wine выпущено несколько продуктов, в частности Cedega, CrossOwer Office и WINE@Etersoft.

Cedega

Cedega (WineX) — проект, основанный на коде Wine. Коммерческая попытка "довести до ума" программу Wine, причем основная цель разработчиков — запуск игр, написанных для Windows. Проект коммерческий, но для домашнего использования его можно загрузить бесплатно. В инсталляции и функционировании мало чем отличается от Wine.

CrossOwer Office

CrossOwer Office — проект, основанный на коде Wine. Специально оптимизирован для запуска "тяжелых" приложений вроде Microsoft Office, Photoshop и т. п. Проект коммерческий, но разработчики активно обмениваются информацией с командой WINE.

WINE@Etersoft

WINE@Etersoft — проект, основанный на коде Wine. Разрабатывается российской командой. Специально предназначен для запуска "тяжелых" приложений: 1С:Предприятие, БЭСТ, Консультант Плюс, Гарант.

Виртуальные машины

Те, кто в компьютерной индустрии давно, наверняка помнят Систему виртуальных машин (СВМ), которая была очень распространена на больших ЭВМ серии ЕС (ЕС 1033/1066 — советский аналог IBM 360/370). Идеи живучи, и для Linux также была создана СВМ, которая получила достаточно широкое распространение и с успехом эксплуатируется.

VMWare

VMWare — это коммерческий продукт, позволяющий запускать на одной машине одновременно несколько операционных систем. Программу можно скачать с сайта производителя и пользоваться ею в тестовых целях в течение месяца.

Установка

Для установки VMWare необходимо скачать rpm-пакет для вашего дистрибутива с сайта разработчика. Установить VMWare можно только от имени пользователя root. После установки нужно запустить vmware-config.pl — скрипт, помогающий настроить VMWare.

Для каждой операционной системы, запускаемой под VMWare, нужно создавать свою конфигурацию. Для этого следует запустить на выполнение файл /usr/bin/vmware. После проверки видеорежима появится окно выбора конфигурации VMWare.

Назначение режимов:

- **Run Configuration Wizard** — для создания быстрой и простой настройки новой виртуальной машины;
- **Run Configuration Editor** — для создания и детальной настройки новой виртуальной машины;
- **Open An Existing Configuration** — для выбора уже созданной виртуальной машины.

При создании виртуальной машины сначала необходимо выбрать тип устанавливаемой на ней операционной системы и каталог, где будут располагаться все

файлы новой виртуальной машины. После этого нужно задать тип жесткого диска (виртуальный или физический диск), установленного на вашем компьютере.

Далее следует разрешить использование дисковода и CD-ROM для виртуальной машины.

Затем нужно настроить поддержку сети для виртуальной машины: полное ее отсутствие, настройки реальной сети или эмуляция сети средствами VMWare.

С помощью Configuration Editor можно провести тонкую настройку уже созданной виртуальной машины.

Пожалуй, это все о VMWare — система очень надежна, позволяет устанавливать множество операционных систем на одном компьютере и, что самое интересное, — эти операционные системы могут быть одновременно запущены и даже обмениваться информацией.

Win4Lin

Еще один эмулятор виртуального компьютера, но, в отличие от VMWare, он создан и оптимизирован специально для запуска Windows в Linux. Для своей работы требует внесения изменений в код операционной системы Linux. Благодаря этому он быстрее и более надежен, чем VMWare. Кроме того, Win4Lin позволяет организовать полнофункциональную DOS-сессию. Единственный недостаток — отсутствие нормальной поддержки DirectX.

Сама Windows запускается из-под X Window в окне. Возможен доступ к любому разделу на винчестере и даже к каталогам операционной системы Linux.

Получить Win4Lin следует с сайта производителя. Для этого необходимо зайти в раздел Members, где нужно бесплатно зарегистрироваться, после чего на ваш электронный адрес будет выслано письмо с вашим логином и паролем. Только после получения пароля вы сможете скачать с сайта нужную программу. Программа-инсталлятор определяет версию дистрибутива, библиотек, установленного ядра операционной системы и предлагает загрузить требуемые для вашей системы подправленное ядро Linux и сам пакет Win4Lin.

На том же сайте нужно получить пробную лицензию на Win4Lin сроком на 30 дней. Сначала устанавливаем новое ядро операционной системы, затем — пакет Win4Lin. После этого необходимо инсталлировать Windows . В каталоге /var/win4lin/publicbin есть утилита installwindows, которую следует запустить и указать ей, где брать инсталляцию Windows. Далее с помощью программы winsetup необходимо настроить устройства и разделы жесткого диска для работы с Windows. Помимо этого, можно указать каталоги, которые будут видны в Windows как диски.

VirtualBox

VirtualBox — еще один эмулятор виртуального компьютера наподобие VMWare. Существуют варианты под все основные операционные системы. Базируется на коде Qemu. Есть две версии: свободная и коммерческая, различающиеся по функциональности. Сейчас принадлежит Sun Microsystems. По отзывам на стандартных конфигурациях работает немного быстрее VmWare, однако были замечены некоторые проблемы на нестандартных конфигурациях персональных компьютеров.

XEN

XEN — монитор виртуальных машин (гипервизор). Способен поддерживать одновременную работу большого числа виртуальных машин на одной физической. Достоинство — очень высокая скорость работы виртуальных машин, не сильно отличающаяся от работы на физической машине. Возможна миграция виртуальной машины между физическими машинами. Отличная поддержка оборудования. Один крупный недостаток — для работы в виртуальной машине необходимо модифицировать операционную систему. На сегодня полностью поддерживается Linux и NetBSD. Поскольку сейчас начали выпускать процессоры с аппаратной поддержкой виртуализации, появилась возможность запускать Windows. На базе XEN существуют коммерческие реализации, например Virtual Iron, XenSource Server, Oracle VM.

KVM

KVM (Kernel-based Virtual Machine) — еще одна система виртуализации. В отличие от XEN более легковесен, поскольку значительная часть функций переложена на ядро Linux. Пока работает только с Linux и Windows, но вскоре обещают поддержку и других операционных систем.

Ссылки

- **linuxbegin.by.ru/articles/article17.shtml** — запуск Windows-программ в Linux.
- **www.linux-ve.chat.ru** — виртуальная библиотека Linux.
- **linux.yaroslavl.ru/Docum/Other/dosemu/README.html** — документация по DOSEmu v. 0.97 pl. 3.0. Перевод Валерия Груздева.
- **www.suse.com/~dosemu/** — домашняя страница DOSEmu.
- **www.osp.ru/os/2001/07-08/023.htm** — Виктор Костромин. Две системы на одном компьютере.
- **www.winehq.org** — официальный сайт проекта Wine.
- **www.vmware.org** — официальный сайт проекта VMWare.
- **dhls.agava.ru/vmware.html** — Ерижоков А. А. Использование VMWare.
- **www.softerra.ru/freeos/16294/print.html** — Александр Куприн. VMWare Workstation — песочница для взрослых.
- **www.netraverse.com** — сайт производителя Win4Lin.
- **www.linux.hitech.by X-Stranger** — Win4Lin — Windows из-под Linux.
- **t37.nevod.perm.su/linux/tune/dosemu.html** — В. Вислобоков. Как установить и настроить DOSEmu.
- **www.mgul.ac.ru/~t-alex/Linux/howto.mine/howto.mine.2.htm** — эмуляция других сред. MINI-NOWTO.
- **www.etersoft.ru** — сайт разработчиков WINE@Etersoft.
- **itc.ua/node/27014/** — ядро Linux как hypervisor.
- **kvm.sourceforge.net** — страница KVM.



Глава 41

Мультимедиа

О программах и устройствах, необходимых для работы в операционной системе Linux, мы говорили на протяжении всей книги, теперь настала пора немного развлечься. Для комфортной работы никогда не мешает слегка отдохнуть. А отдых при помощи компьютера можно обозначить одним емким словом — мультимедиа.

Точного определения мультимедиа так никто и не сформулировал, мы же под этим подразумеваем звук и видео во всех их проявлениях.

Настройка звуковой карты

Начнем со звука. Современные дистрибутивы распознают большинство звуковых карт и при инсталляции дистрибутива практически всегда корректно их устанавливают. Если у вас, все же, возникли трудности со звуковой картой, не огорчайтесь — они преодолимы. Во-первых, зайдите на сайт производителя дистрибутива — возможно, там о проблеме с вашим типом звуковой карты уже известно, предложено решение или есть обновленные драйверы. Во-вторых, можно сходить на сайт разработчиков драйверов звуковых карт **www.alsa-project.org**. Почти наверняка для вашей звуковой карты там есть свежий драйвер. В документации на драйвер описан процесс компиляции, установки и настройки драйвера.

Консольные утилиты для работы со звуком

Чтобы не обижать тех, кто привык работать с консольным режимом, начнем с них. Первое, что нам необходимо сделать — добраться до утилит регулирования громкости звука. По традиции эти программы называются *микшерами* и зачастую в своем названии содержат это слово, например aumix (рис. 41.1).

Помимо этой утилиты есть еще с ряд других микшеров, к примеру, alsamixer, xmix.

Управлять громкостью звука мы уже умеем. Как правило, первое, что приходит в голову — попытаться воспроизвести Audio-CD. Нет ничего легче. Если на вашем CD-ROM есть кнопка воспроизведения — вставляем диск и нажимаем ее. Если такой кнопки нет, воспользуемся какой-нибудь подходящей программой, например cdplay. Простая утилита, минимум функциональности, но для прослушивания компакт-дисков в консольном режиме вполне подходит. Если же вас не удовлетворяют ее возможности, воспользуйтесь программой cdp. В отличие от предыдущей утилиты

вы можете более комфортно прослушивать музыку: управлять громкостью, последовательностью воспроизведения музыкальных треков, менять компакт-диски.

Следующая идея — каким-то образом воспроизвести четырехгигабайтный архив музыки в формате MP3.

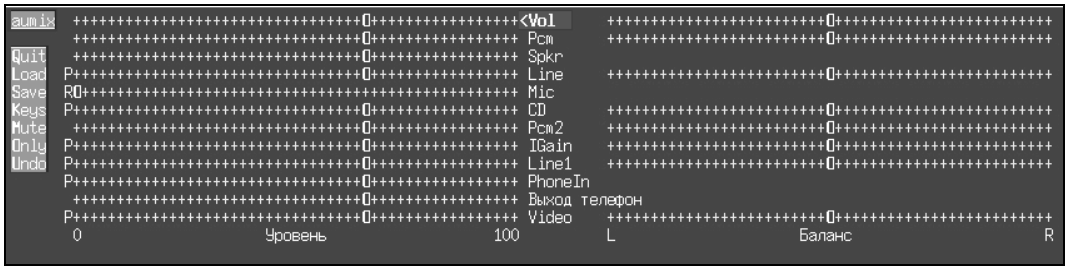


Рис. 41.1. Утилита управления громкостью звука aumix

В связи с тем, что за использование кодера/декодера MP3 держатель патента требует лицензионных отчислений, практически все сборщики дистрибутивов не включают в состав программы/кодеки MP3. Однако ничто не мешает вам скачать недостающее из Интернета.

Для воспроизведения файлов формата MP3 существует отличная программа — mpg123. За полным описанием возможностей программы mpg123 мы рекомендуем обратиться к документации на нее. Вкратце поясним, что в командной строке необходимо задать путь к воспроизводимому файлу или к каталогу с маской файла *, и все, что находится в этом файле, будет воспроизведено. Можно также создать свой список воспроизведения (play-list) или включить режим воспроизведения музыкальных файлов в произвольном порядке. Для воспроизведения файлов MP3 предназначены также программы blaster и splay.

Помимо файлов MP3 вам может понадобиться воспроизвести музыкальные файлы и других форматов. Утилита wavplay предназначена для воспроизведения музыкальных файлов формата WAV, программа playmidi — для музыкальных файлов MIDI, а утилита tracker — формата MOD. В том случае, если вам захочется преобразовать музыкальный файл из одного формата в другой, воспользуйтесь программой Sox.

Все, о чем писалось ранее, касалось готовых музыкальных файлов. Но вот вам принесли новый музыкальный компакт-диск, и вы хотите сохранить его содержимое в файлах формата MP3. Для этого сначала воспользуемся программой cdparanoia. Программа проста в использовании — вставляем компакт-диск и запускаем cdparanoia, причем первым аргументом указываем номер музыкального трека, а вторым — имя выходного файла формата WAV.

Программа cdparanoia имеет много опций командной строки, о которых можно узнать из документации. Для нас наиболее интересна опция `-B`, позволяющая оцифровать сразу все дорожки компакт-диска.

Однако после работы программы cdparanoia мы имеем просто wav-файл. А нам необходимо было получить файлы формата MP3. Различных программ преобразования полученных wav-файлов в формат MP3 довольно много, поэтому обратим ваше внимание только на одну — lame. Эта утилита очень популярна среди пользователей, позволяет выбирать алгоритм сжатия файла, частоту дискретизации и многое другое. Однако в самом простом случае достаточно указать только два аргумента — имена исходного wav-файла и целевого файла формата MP3. Процесс оцифровки компакт-диска и создания файла формата MP3 можно совместить, выполнив команду

```
cdparanoia 1 | lame – my_music.mp3
```

Если вас не устраивает такой процесс получения MP3-файлов, можно воспользоваться программой mp3c (рис. 41.2).

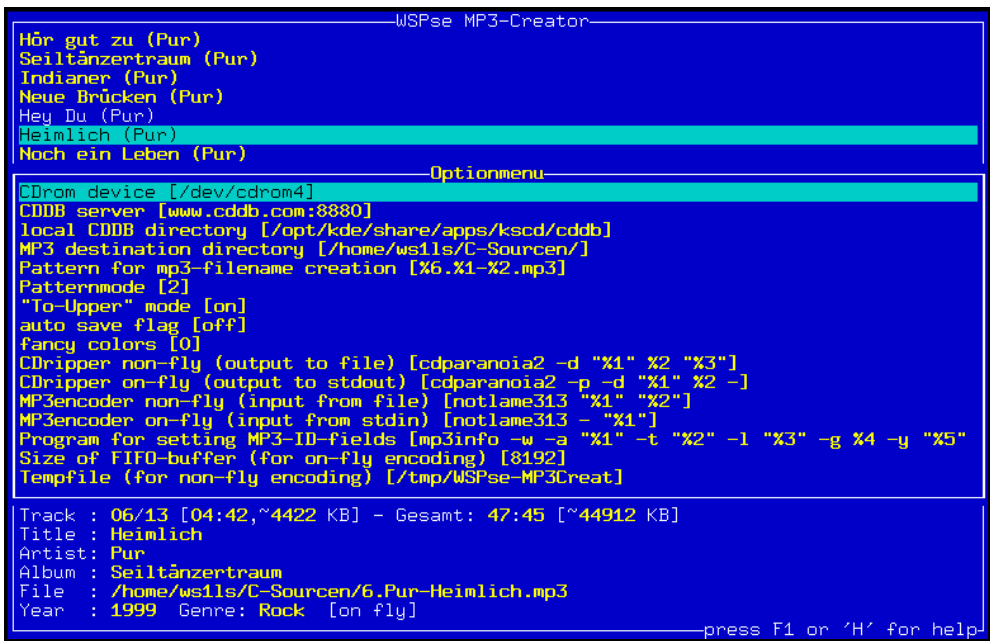


Рис. 41.2. Программа оцифровки компакт-дисков в формат MP3

Это просто оболочка, которая для выполнения своих функций вызывает различные программы оцифровки компакт-дисков и утилиты-конвертеры.

Программа mp3c имеет ряд дополнительных возможностей — заполнение полей автора, названия песни, имени альбома и другой информации, которая может содержаться в файле формата MP3.

Звук в X Window

Рассмотрев консольный режим, перейдем в графический, в X Window. Здесь еще больше разнообразия. Возьмем для примера KDE и GNOME. В этих оболочках

программы мультимедиа сгруппированы в пункте меню мультимедиа. Посмотрим, что же есть в этом разделе в среде GNOME.

Выбор достаточный. На рис. 41.3 вы видите окно программы, очень похожей на программу **Звукозапись** в Windows. То же назначение — запись с микрофона. Простой интерфейс: начать запись, остановить, продолжить и сохранить файл.

Пойдем далее. На рис. 41.4 изображен микшер звуковой платы и проигрыватель компакт-дисков.

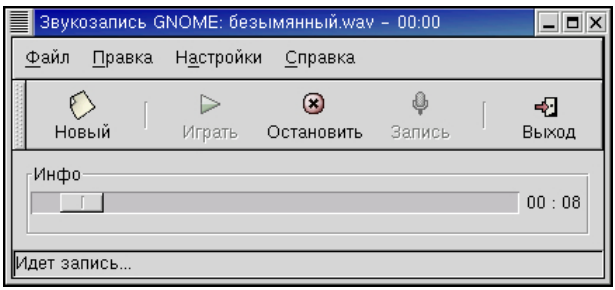


Рис. 41.3. Программа звукозаписи с микрофона

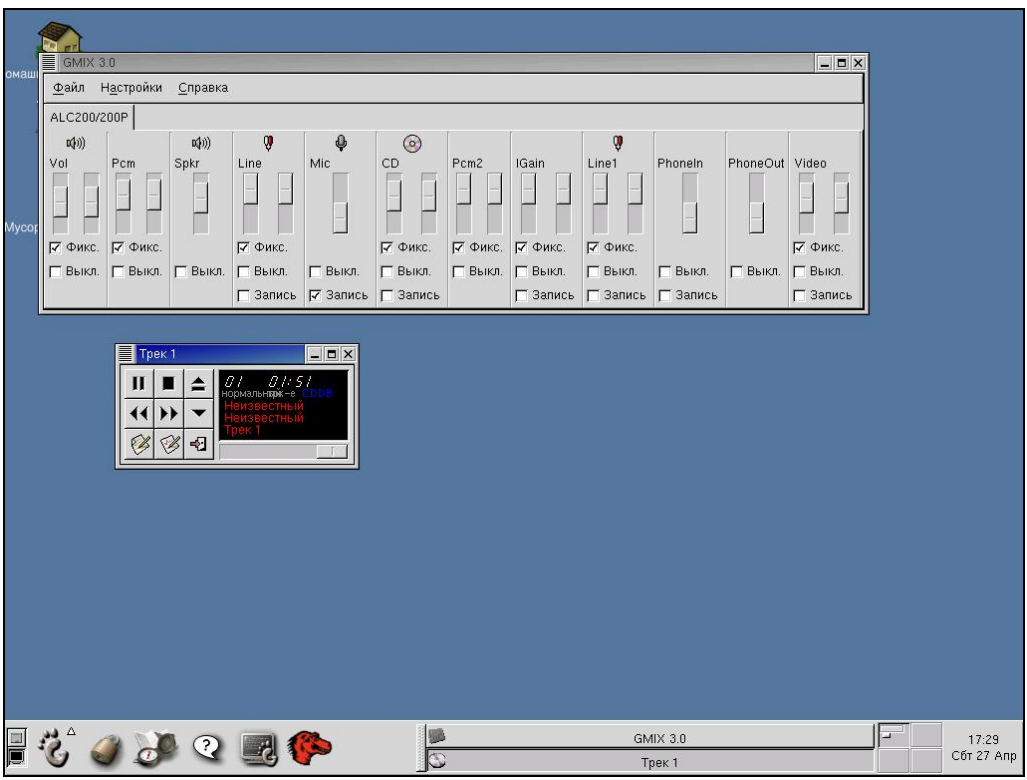


Рис. 41.4. Микшер звуковой платы и программа для воспроизведения компакт-дисков

Программа xmms (рис. 41.5) — практически полный аналог winamp. Великолепно воспроизводит MP3-файлы, позволяет создавать play-list. Помимо всего прочего, существует возможность самостоятельно изменять внешний вид программы в широких пределах.

А теперь перейдем к программам мультимедиа, входящим в стандартную поставку KDE. Набор программ близок по составу к GNOME, однако он несколько шире. Кратко рассмотрим имеющиеся программы.

Программа KMid (рис. 41.6) представляет собой проигрыватель midi-файлов. Позволяет выбирать инструменты, редактировать файл, менять темп воспроизведения музыки и некоторые другие параметры.

Аналогичная по назначению программа, только имеющая больше возможностей, называется KMidi. Внешний вид программы представлен на рис. 41.7.



Рис. 41.5. Программа xmms

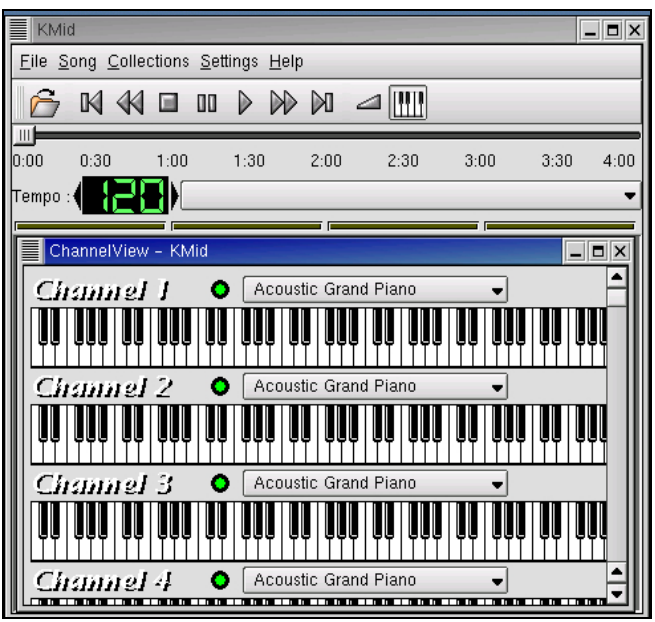


Рис. 41.6. Программа KMid

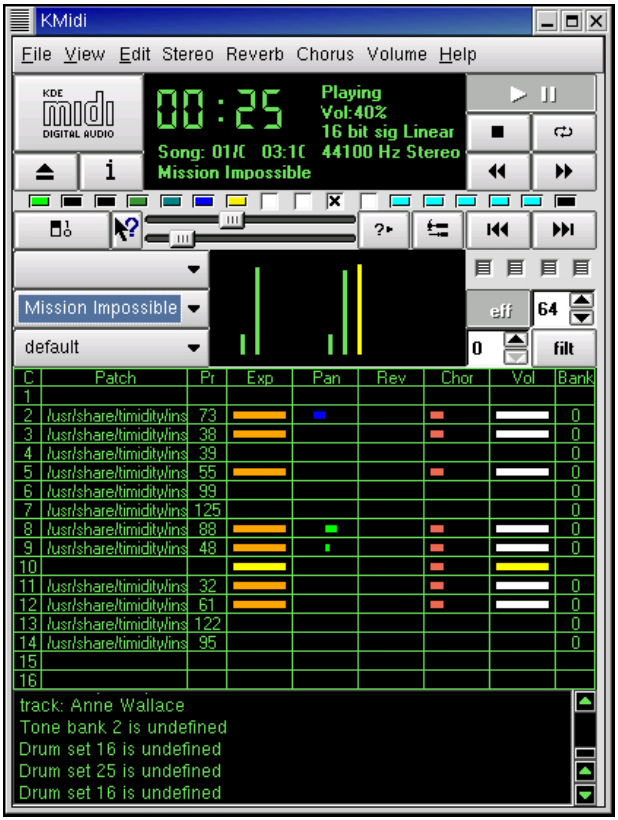


Рис. 41.7. Программа Kmidi



Рис. 41.8. Программа CD-проигрыватель

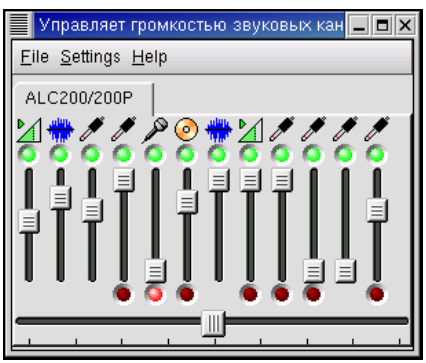


Рис. 41.9. Микшер звуковой платы

Программа для воспроизведения компакт-дисков также входит в мультимедийный набор KDE (рис. 41.8).

И, наконец, есть микшер для регулировки громкости воспроизведения звука (рис. 41.9).

Помимо рассмотренных программ, можно найти и установить много дополнительного программного обеспечения для работы со звуком на все случаи жизни. Вот пример — программный пакет festival. Если у вас бессонница, а самому читать себе сказки на ночь не хочется, воспользуйтесь этой программой — она вам вслух почитает сказки. Правда, полноценной поддержки русского языка пока нет, но в качестве альтернативы можно воспроизводить тексты на испанском — звучание получается вполне приемлемого качества. А если вам вдруг захотелось попытаться заставить свой компьютер понимать человеческую речь — фирма IBM распространяет бесплатную систему распознавания речи ViaVoice.

Видео в Linux

На компьютере под управлением операционной системы Linux можно и кино посмотреть. Для этой цели существует несколько программ, позволяющих воспроизводить видеофайлы различных форматов. На некоторых из них мы остановимся подробней.

Пожалуй, один из старейших форматов хранения видео на компьютере — MPEG. Хорошая программа для воспроизведения файлов такого типа — smpeg — обладает понятным интерфейсом, проста в управлении, нетребовательна к ресурсам компьютера. На базе программы smpeg созданы следующие программные пакеты:

- Enjoympeg — MPEG-проигрыватель;
- Dumpmpeg — простая программа для захвата кадров из mpeg-фильмов;
- XMPS — полнофункциональный MPEG-проигрыватель с поддержкой play-list и изменением внешнего вида;
- ZZPlayer — MPEG-проигрыватель для KDE;
- Xtheater — программа для воспроизведения Video CD.

Но главное не это. Почти все диски с видеофильмами для воспроизведения на компьютере, которые сейчас доступны у нас, закодированы с использованием формата MPEG4 (DivX). Соответственно, нам нужна программа, которая могла бы воспроизводить эти файлы. Давайте посмотрим, что нам могут предложить разработчики программного обеспечения.

Большинство программ работают с кодеком сжатия, предназначенным для Windows, поэтому желательно иметь самый свежий кодек. Получить его можно на сайте DivX (**www.divx.com**). Помимо этого, возьмите библиотеку avifile (**avifile.sourceforge.net**), которая позволяет использовать Windows AVI-кодеки (Indeo, Video, DivX) в операционной системе Linux.

Программа XMPS

Программа XMPS — полнофункциональный MPEG-проигрыватель с поддержкой play-list и изменением внешнего вида (рис. 41.10).



Рис. 41.10. Программа XMPS

Программа avifile-player

Внешний вид avifile-player приведен на рис. 41.11. Поскольку программа задействует Win32-библиотеки, то можно смотреть не только DivX, но и AVI-файлы, кодированные другими Windows-кодеками.

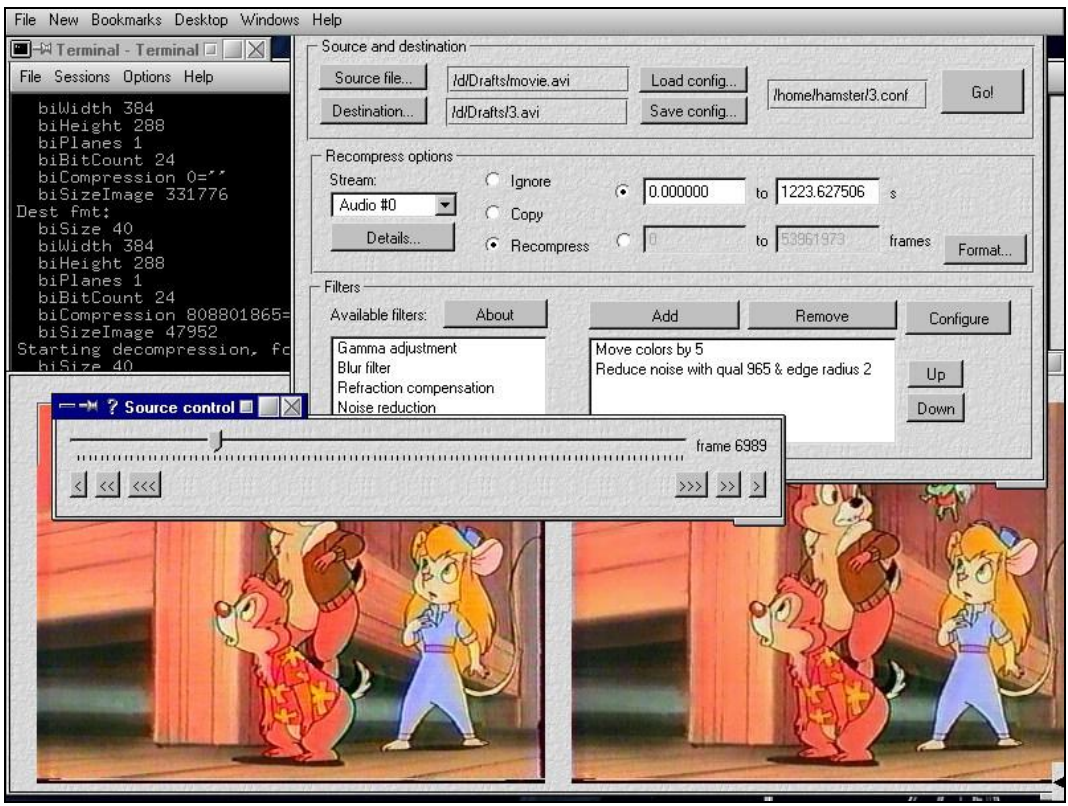


Рис. 41.11. Проигрыватель avifile-player

Программа xmms

Об этой программе мы уже упоминали, когда рассматривали воспроизведение MP3-файлов. Однако при установке соответствующих plug-ins программа может работать и с видеофайлами. Для этого необходимо загрузить xmms-avi и установить avifile-библиотеку и AVI-кодеки Win32. В результате получается хорошая программа для воспроизведения видеофильмов, правда достаточно сильно загружающая центральный процессор.

Программа XMMP — LinuX MultiMedia Player

Программа XMMP — LinuX MultiMedia Player. Использует avifile-библиотеку и AVI-кодеки Win32. Проект новый, очень динамично развивающийся, поэтому многие возможности заявлены, но до конца не реализованы.

По замыслу автора проекта XMMP (рис. 41.12) должен представлять собой центр по обработке видео. Это программное обеспечение должно проигрывать видео, создавать, редактировать и конвертировать файлы мультимедиа. Обеспечивает хорошее качество воспроизведения.



Рис. 43.11. Проигрыватель XMMP

Программа MPlayer

Программа MPlayer — Movie Player for Linux (рис. 41.13).

Эта программа на сегодняшний день — безусловный лидер в мире Linux. Она очень удобна, многофункциональна, имеет множество настраиваемых параметров и обеспечивает очень качественное воспроизведение. Внешний вид программы

можно изменять по своему усмотрению (рис. 41.14). И при этом центральный процессор при воспроизведении загружен всего лишь на 15–20%. Но главное, программу можно скомпилировать для работы и с графическим интерфейсом, и в консольном режиме. Кроме того, если у вас есть соответствующее оборудование, можно управлять проигрывателем при помощи дистанционного пульта управления. Программа очень динамично развивается, поэтому заходите на сайт разработчика хоть раз в месяц — наверняка найдете что-то новое.

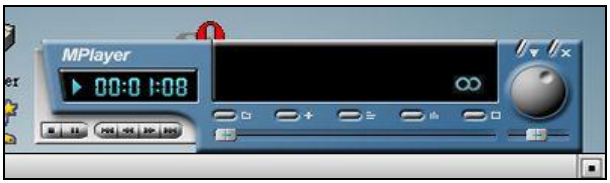


Рис. 41.13. Внешний вид проигрывателя MPlayer

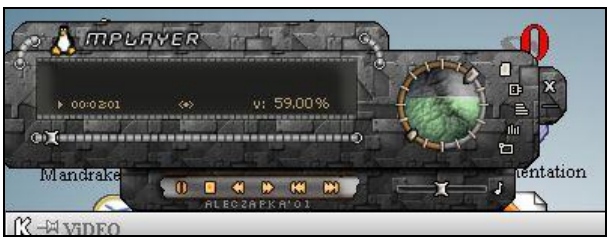


Рис. 41.14. Проигрыватель MPlayer в другом обличье

Программа XINE

Это еще один проигрыватель видеофайлов (рис. 41.15).

Проигрыватель поддерживает большое число видеокодеков:

- MPEG1;
- MPEG2;



Рис. 41.15. Проигрыватель XINE

- MPEG4;
- DivX;
- motion JPEG;
- AVI (использует Win32-кодеки: Indeo 3.1-5.0, cinepak, Window Media 7/8).

Помимо воспроизведения видео, XINE умеет работать и с аудиофайлами:

- MPEG audio layer 1;
- MPEG audio layer 2;
- MPEG audio layer 3;
- a/52 (ac3, dolby digital);
- dts;
- vorbis;
- pcm;
- DivX audio.

Отличается несколько нестандартным управлением, умеет изменять внешний вид, не сильно загружает центральный процессор.

Запись CD-R/CD-RW-дисков

Запись CD-R/RW-дисков давно уже стала рутиной и насущной необходимостью. Поэтому логично рассмотреть этот процесс именно в главе, посвященной мультимедиа.

Итак, что следует знать о записи дисков? Во-первых, приводы бывают внешние и внутренние. С внутренними приводами проблем нет, все давно отработано. С внешними многое зависит от его интерфейса. Внутренние приводы последние несколько лет поставляются интерфейсом IDE, однако встречаются еще устройства с интерфейсом SCSI.

До последнего времени ядро операционной системы Linux работало с пишущими устройствами через эмуляцию интерфейса SCSI, однако в последних версиях ядра включена поддержка пишущих приводов через ATAPI-интерфейс. В результате сейчас мы имеем переходный период, когда не все программы для записи дисков понимают новый интерфейс. На этом моменте я остановлюсь особо, а пока перейдем к практике.

Запись дисков основывается на наборе консольных утилит в cdrtools, куда входят программы cdda2wav, cdrecord, isoinfo, mkisoft, readcd.

При инсталляции дистрибутива он автоматически определит тип вашего записывающего устройства и настроит систему для его нормальной работы. Теперь можно переходить к записи. Как и для большинства других программ в Linux, существуют консольные (пакет cdrtools) и графические программы. Однако графические программы по сути дела являются front-end (надстройка над базовой функциональностью, программа для выполнения основных задач использует более низкоуровневую программу). Далее вкратце я расскажу о записи в консольном режиме, а основной упор буду делать на графические приложения.

Обычно под Linux компакт-диск записывают в два шага:

- упаковка желаемых данных (файлы, музыка или и то, и другое) в файлы в специальном формате;
- запись файлов на CD-R с помощью `cdrecord`.

Для CD-RW может добавиться этап стирания диска.

Создание образа CD-ROM

Перед использованием любого носителя необходимо создать файловую систему, которая ответственна за организацию и хранение файлов на нем. Но поскольку диски CD-R однократно записываемые, то необходимо создать виртуальный диск, отформатировать его, разместить на нем данные и затем полученный образ диска записать на CD-R-диск. Для этого предусмотрена утилита `mkisofs`. Типичный запуск выглядит так:

```
mkisofs  -r   -o cd_image   game/
```

Опция `-r` устанавливает права всех файлов на компакт-диске на чтение всем и разрешает расширение Rock Ridge. `mkisofs` пробует отобразить все имена файлов в формате 8.3, используемом DOS для максимальной совместимости.

Как вы заметили, выход `mkisofs` непосредственно не послан на устройство записи компакт-дисков и вот почему:

- `mkisofs` ничего не знает об устройствах записи компакт-дисков;
- вы можете захотеть протестировать образ перед записью.

Проверка образа CD

Linux может монтировать файлы в виде разделов диска. Это позволяет нам подмонтировать созданный образ диска, проверить его содержимое, при необходимости извлечь некоторую информацию с диска. Чтобы монтировать созданный ранее файл cd_image в каталог /cdrom, дайте команду

```
mount -t iso9660 -o ro,loop=/dev/loop0 cd_image /cdrom
```

Теперь вы можете проверить файлы в /cdrom — они появляются точно так, как будут на реальном компакт-диске. Для демонтирования CD-изображения просто введите команду:

```
umount /cdrom
```

Запись образа диска на CD

Теперь необходимо узнать, где находится ваше записывающее устройство:

```
cdrecord -scanbus
```

Данная команда покажет, как именуется ваш привод в системе. После этого можно провести запись, задав команду

```
cdrecord -v speed=32 dev=0,6,0  -data  cd_image
```

Запись Audio-CD

Запись audio-CD очень похожа на запись обычного диска. Основные отличия заключаются в том, что Audio-CD состоит из аудиотреков, которые необходимо организовать как отдельные образы. Второе отличие — формат образов (16 бит стерео в PCM-кодировании с частотой оцифровки 44 100 Гц).

Одна из программ для конвертирования звуковых файлов в требуемый формат — sox. Для создания образа трека достаточно выполнить команду:

```
sox  myMusic.wav  track1.cdr
```

cdrecord записывает образы как аудиотреки, если указана опция `-audio`:

```
cdrecord -v speed=4 dev=0,6,0  -audio  track1.cdr track2.cdr...
```

Для конвертации MP3-файлов можно воспользоваться командой:

```
mpg123 -s track1.mp3>track.cdr
```

Копирование дисков

Копирование дисков не составляет труда. К примеру, у нас есть пишущий привод и отдельный привод CD-ROM. Воспользуемся командой

```
cdrecord -v dev=0,6,0 speed=32 -isosize /dev/hdc
```

Программа читает поток данных из привода CD-ROM, присоединенного как /dev/hdc, и записывает его непосредственно на CD-R.

Если в системе есть только пишущий привод, то необходимо считать образ CD на жесткий диск, а затем его записать.

Перезаписываемые диски

Для перезаписи CD-RW-диска необходимо сперва его очистить от старой информации, указав при записи параметр `blank=fast` для cdrecord.

Оболочки для записи дисков

Для записи дисков можно и нужно применять программы, облегчающие жизнь. Обычно их называют front-end или оболочками. Их назначение — предоставить пользователю удобный графический интерфейс и выполнить все необходимые манипуляции и настройки для программы cdrecord.

K3b

Программа K3b удобна в работе и проста в освоении. Внешний вид программы приведен на рис. 41.16.

K3b обычно поставляется вместе с KDE. Программа самостоятельно находит рабочие параметры cdrecord и сопутствующих утилит.

Программа при создании Audio-CD из готовых файлов может конвертировать файлы форматов Ogg Vorbis, MP3, WAV и FLAC в звуковые дорожки аудиодиска. Для создания проекта достаточно перетащить в него файлы или каталоги. Проект можно записать на CD, DVD или сохранить как ISO-образ.

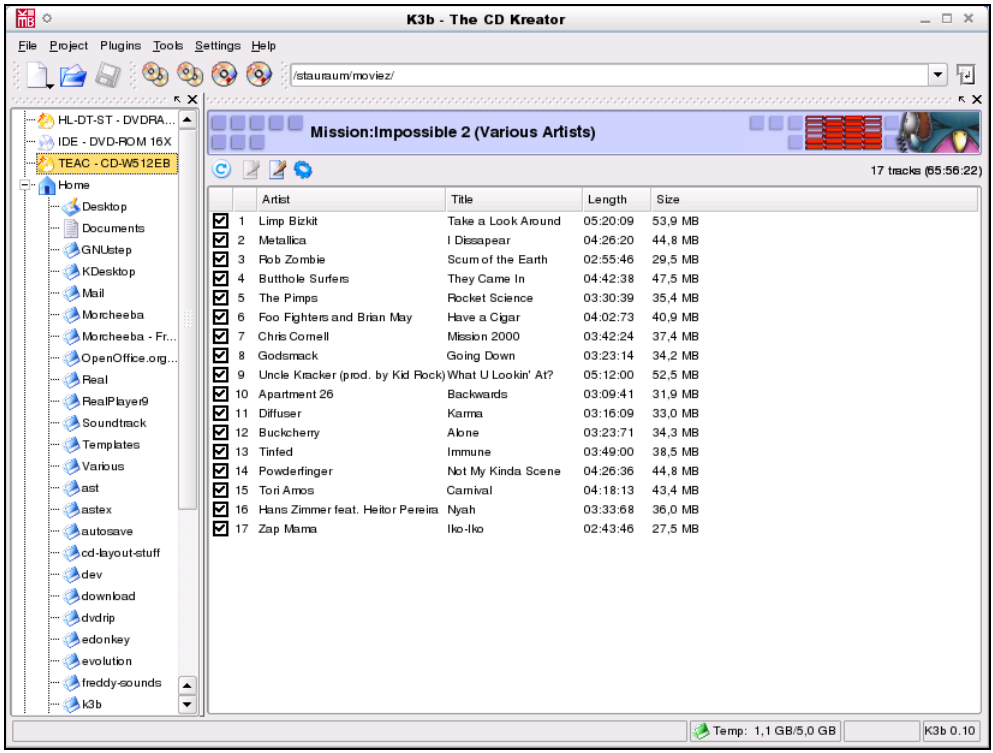


Рис. 41.16. Главное окно K3b

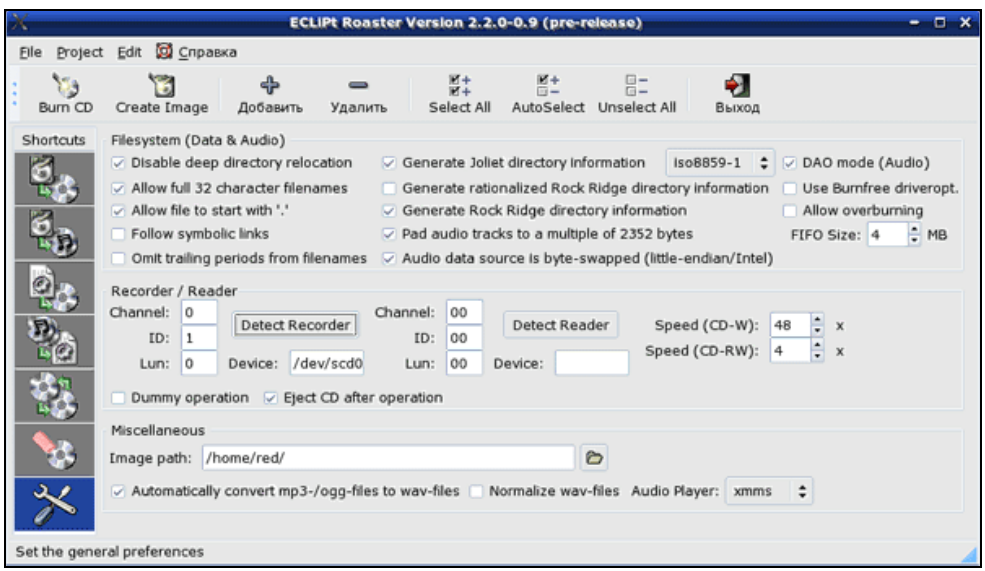


Рис. 41.17. Страница настроек в Eroaster

Помимо этого, K3b может конвертировать музыкальный компакт-диск в нужный вам формат файлов.

Eroaster

Пакет Eroaster предназначен для продвинутых пользователей. Он не определяет автоматически настройки привода, параметры записи. Делается это на странице настроек, в секции **Recorder/Reader** (рис. 41.17).

CD Bake Oven

Эта программа — аналог Nero. Она позволяет копировать аудиодиски и диски с данными, записывать диск с образа, создавать музыкальный диск, стирать диски CD-RW. Поддерживает Drag'n'Drop, мультисессионную запись, запись дисков без создания образа, автоматическое определение устройства записи, возможность просмотра созданного и имеющегося ISO-образа, прослушивание аудиодиска, сохранение проекта, автоматический подсчет суммарного размера файлов в проекте, создание загрузочного диска и т. п. Программа развивается, поэтому не забывайте отслеживать изменения на ее сайте.

Помимо упомянутых ранее программ существуют десятка три менее известных, однако перечислять их названия и особенности я не буду. При желании вы самостоятельно найдете ту, которая наиболее полно подойдет для ваших нужд.

Вот, пожалуй, и все о мультимедиа.

Ссылки

- **avifile.sourceforge.net** — сайт avifile; программа позволяет использовать AVI-кодеки Windows (Indeo, Video, DivX).
- **divx.euro.ru** — сайт видеопроигрывателя avifile-player.
- **mplayer.sourceforge.net** — сайт программы Mplayer (Movie Player for Linux).
- **people.freenet.de/for_Ki/** — сайт проигрывателя Enjoympeg.
- **sourceforge.net/projects/dumpmpeg** — сайт программы захвата видеокадров dumpmpeg.
- **www.chez.com/tsc/zzplayer/zzplayer.html** — сайт KDE MPEG-проигрывателя ZZPlayer.
- **www.divx.com** — сайт оригинального DivX-кодека.
- **www.frozenproductions.com/xmmp** — сайт проекта XMMP (LinuX MultiMedia Player).
- **www.linuxjournal.com/article.php?sid=4382** — Adam Williams. Issue 81: Movie Making on a Linux Box? No Way!
- **www.linuxoid.ru/how_to/DivX.html** — Гвоздев Андрей. DivX в Linux.
- **www.lokigames.com/development/smpeg.php3** — официальная страница проигрывателя smpeg.
- **www.opendivx.org** — сайт кодека DivX с открытым исходным кодом.
- **www.softerra.ru/freeos/13036/** — Алексей Федорчук. Консольное мультимедиа.

- **www.softerra.ru/freeos/14906/** — Алексей Федорчук. Как граббить на-грабленное.
- **xine.sourceforge.net** — сайт программы XINE.
- **xmms.org** — сайт программы XMMS (воспроизведение аудио и видео).
- **xmps.sourceforge.net** — сайт видеопроигрывателя XMPS.
- **xtheater.sourceforge.net** — сайт программы для воспроизведения Video-CD Xtheater.
- **http://gazette.linux.ru.net/lg63/articles/rus-andreiana.html** — Marius Andreiana. Линукс на вашем рабочем столе: Мультимедиа. Перевод Александра Михайлова.
- **http://soft.mail.ru/article_page.php?id=147** — Семилетов П. В. Записываем CD-R/RW в Linux.
- **http://k3b.sourceforge.net** — официальный сайт программы K3b.
- **http://cdbakeoven.sourceforge.net** — официальный сайт программы CD Bake Oven.
- **http://eclipt.uni-klu.ac.at** — официальный сайт программы Eroaster.
- Журнал "Мой компьютер" № 236 — Сергей Яремчук. Пингвин печет блины.
- CD-Writing-HOWTO.



Глава 42

Программы для работы с файлообменными сетями, менеджеры закачки

Современные скорости Интернета, пожалуй, даже несколько развращают: просмотр видео по запросу в реальном времени, онлайн-радио, чтение объемных PDF-файлов и многое другое.

Однако не все имеют скоростной канал, некоторые используют "хитрые" тарифы, например ночью скорость больше в два раза. С другой стороны, есть такие вещи (например, дистрибутивы или HD-видео), которые даже на канале в 10 Мбит придется скачивать сутками.

Поэтому нужны некие программы, позволяющие получать данные по разным протоколам (HTTP, FTP, Torrent, Edonkey) с возможностью приостановки, докачки, регулируемой скорости скачивания и отдачи и т. п. Под Windows этого "добра" много, что же есть для Linux?

Существует множество программ — как клиентов, так и серверов, как консольных, так и с графическим и даже Web-интерфейсом, начиная от встроенных в браузеры Mozilla и Opera, различных Torrent-клиентов и Edonkey/Emule для p2p-сетей файлообмена и просто отдельных "качалок". В этой главе подробно рассмотрим всего две программы — Wget и MLdonkey. Возможно, вы посчитаете это недостаточным, но благодаря данным программам мы перекрываем весь спектр протоколов для скачивания; кроме того, описанные программы функционируют в консольном режиме, могут работать демоном и главное — у каждой много разных графических оболочек, в том числе и с Web-интерфейсом. Поэтому можно очень гибко расходовать машинное время, запускать программы по расписанию, в конце концов, установить и использовать на сервере или удаленном компьютере.

Wget

Консольная программа, стандартная для большинства дистрибутивов уже порядка 10 лет. Неудобство командной строки позволяют скрыть специально разработанные графические интерфейсы.

Wget поддерживает протоколы HTTP, HTTPS, FTP и может работать через HTTP-proxy. Поддерживается автоматическая докачка файлов с момента обрыва. При зеркалировании Wget автоматически сравнивает локальные файлы и каталоги

с файлами и каталогами на скачиваемом сайте, докачивая только недостающие или измененные данные. Поддерживается эмуляция конкретно заданного браузера, что позволяет "прикинуться" Wget для WWW-сервера, например, Mozilla или Internet Explorer.

Wget не является интерактивной программой: после запуска пользователь не может повлиять на ее работу иначе, как прервав выполнение.

Файлы можно скачивать рекурсивно по ссылкам в HTML-страницах как с одного сайта с определенной глубиной следования по ссылкам, так и с нескольких. При загрузке с FTP файлы можно скачивать по маске имени. На рис. 42.1 приведен пример работы программы Wget.

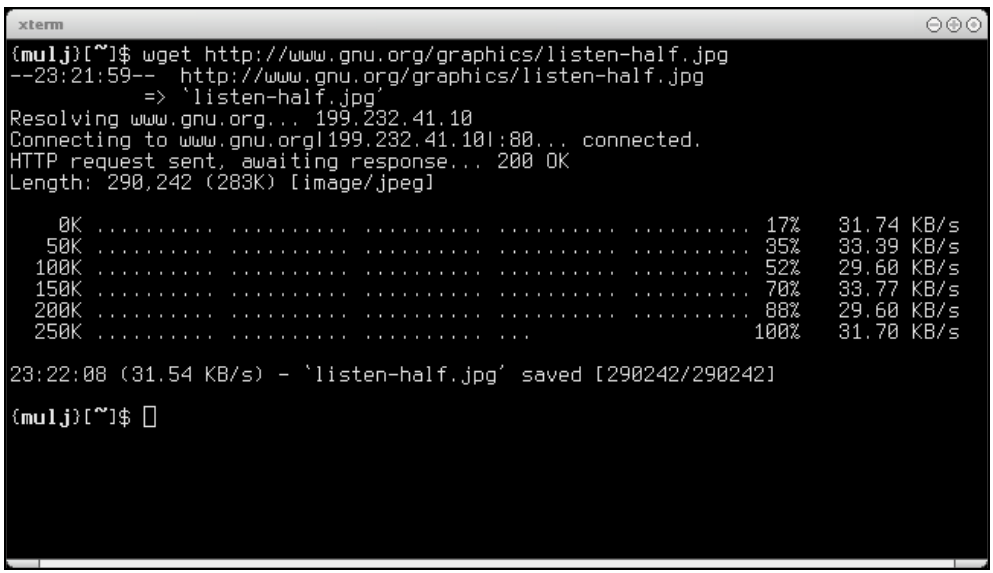


Рис. 42.1. Пример работы программы Wget

Команды Wget

Команд достаточно много, с полным списком можно ознакомиться в соответствующей документации. Здесь приведем только самые основные.

Скачивание одного файла

- `wget http://www.sample.org/graphics/sample.png` — скачать файл в текущий каталог.
- `wget -P /path/for/save ftp://ftp.example.org/some_file` — запись файла в определенный каталог.
- `wget -b ftp://ftp.example.org/some_file` — скачивание в фоновом режиме.
- `wget -c ftp://ftp.example.org/some_file.iso` — продолжить получение недокачанного файла (после обрыва связи, останова скачивания).

Скачивание нескольких файлов

- `wget -i имя_файла` — скачать несколько файлов; здесь `имя_файла` — некий файл, каждая строчка которого содержит полный путь скачиваемого файла.
- `wget -r -l5 http://sample.org/ -o log` — скачать страницу с глубиной вложенности ссылок 5 и записью протокола в log-файл.

> ***Замечание***
>
> Для преобразования ссылок в скачанных страницах в локальные достаточно в командную строку добавить ключ `-k`.

- `wget -r --no-parent http://example.org/my-archive/` — скачать каталог и все подкаталоги.
- `wget -r -l 0 -k http://example.org` — скопировать весь сайт целиком.
- `wget -m -np http://example.org` — зеркалирование сайта.

Конфигурационный файл .wgetrc

Постоянно используемые Wget ключи командной строки и другие данные удобно определить в конфигурационном файле, называемом .wgetrc. Он хранится в домашнем каталоге пользователя и состоит из текстовых строчек вида `имя_переменной = значение`.

Вот список переменных и значений.

- `accept/reject = STRING` — скачивать или нет файлы, заданные `STRING`.
- `add_hostdir = on/off` — включить/выключить префикс имени хоста в именах файлов.
- `continue = on/off` — включить/выключить продолжение загрузки.
- `background = on/off` — включить/выключить выполнение в фоновом режиме.
- `base = STRING` — установить базу для относительных URL.
- `cache = on/off` — при установке в `off` запрещает кэширование на уровне сервера.
- `convert links = on/off` — конвертировать локально абсолютные ссылки.
- `cut_dirs = N` — игнорировать `N` компонентов удаленного каталога.
- `debug = on/off` — включить/выключить режим отладки.
- `delete_after = on/off` — удалять после загрузки.
- `dir_prefix = STRING` — вершина дерева каталогов.
- `dirstruct = on/off` — создание структуры каталогов.
- `dot_bytes = N` — число байтов, "содержащихся" в точках, которые отображаются при выгрузке (по умолчанию 1024). Вы можете заканчивать значение суффиксами `k` или `m` для килобайтов и мегабайтов соответственно. При помощи настройки точек можно подстраивать отображение по своим нуждам или пользоваться предопределенными стилями.
- `dots_in_line = N` — число точек, выводимых в одной строке во время выгрузки (по умолчанию 50).
- `dot_spacing = N` — число точек в одном кластере (по умолчанию 10).

- `exclude_directories = STRING` — разделенный запятыми список каталогов, которые необходимо исключить из процесса загрузки.
- `exclude_domains = STRING` — исключаемые из загрузки домены.
- `follow_ftp = on/off` — следование FTP-ссылкам из HTML-документов.
- `force_html = on/off` — при установке в `on` принуждает рассматривать входной файл как документ формата HTML.
- `ftp_proxy = STRING` — использовать `STRING` как FTP proxy-сервер вместо значения, определенного в окружении.
- `header = STRING` — определить дополнительный HTTP-заголовок.
- `http_passwd = STRING` — установить HTTP-пароль.
- `http_proxy = STRING` — использовать `STRING` как HTTP proxy-сервер вместо того, что определен в окружении.
- `http_user = STRING` — устанавливает имя HTTP пользователя.
- `ignore_length = on/off` — установка в `on` приводит к игнорированию заголовка "Content-Length".
- `include_directories = STRING` — задает разделенный запятыми список каталогов, которые необходимо обработать во время загрузки.
- `input = STRING` — имя файла для получения списка обрабатываемых URL.
- `kill_longer = on/off` — расценивать данные, превышающие по длине значение, определенное в заголовке "Content-Length", как некорректные (и повторить попытку их получения). Основное назначение — сохранить так много данных, насколько это вообще возможно исходя из того, что размер удаленных данных равен значению"Content-Length" или больше него.
- `logfile = STRING` — имя файла отчета.
- `login = STRING` — имя пользователя для доступа по FTP на дистанционную машину. Значение по умолчанию — "anonymous".
- `mirror = on/off` — управление режимом зеркалирования.
- `netrc = on/off` — включить/выключить обработку файла `netrc`.
- `no_parent = on/off` — запретить загрузку за пределами указанной иерархии каталогов.
- `no_proxy = STRING` — назначает `STRING` как разделенный запятыми список доменов, для которых не нужна загрузка через прокси-сервер, вместо того, что определен в окружении.
- `output_document = STRING` — имя выходного файла.
- `passive_ftp = on/off` — устанавливает пассивный режим FTP.
- `passwd = STRING` — пароль для доступа к FTP-сервису. По умолчанию установлен ваш адрес электронной почты.
- `proxy_user = STRING` — имя пользователя для авторизации на proxy-сервере.
- `proxy_passwd = STRING` — пароль для авторизации на proxy-сервере.
- `quota = QUOTA` — определяет квоту на загрузку. Квота может быть определена в байтах (по умолчанию), килобайтах (с добавлением `k`) и мегабайтах (с добавлением `m`). Например, `quota = 5m` устанавливает квоту 5 Мбайт. Команда целесообразна в глобальном wgetrc. Когда она задана, Wget остановит выгрузку в момент, когда суммарный размер загруженных данных станет равным либо больше квоты. Глобальные установки могут быть перекрыты пользовательскими.

- `reclevel = N` — глубина рекурсии.
- `recursive = on/off` — разрешение рекурсии.
- `relative_only = on/off` — следовать только относительным ссылкам.
- `remove_listing = on/off` — при установке в `on` Wget будет удалять файлы полученных FTP-листингов.
- `retr_symlinks = on/off` — при установке в `on` Wget будет загружать символьные ссылки как обычные файлы.
- `robots = on/off` — задействовать (или нет) файл "/robots.txt".
- `server_response = on/off` — печатать или нет HTTP- и FTP-ответы серверов.
- `timeout = N` — устанавливает время тайм-аута.
- `timestamping = on/off` — управление временными штампами.
- `tries = N` — установить количество попыток на URL.
- `use_proxy = on/off` — использование proxy-серверов.
- `verbose = on/off` — включить/выключить подробный отчет.
- `wait = N` — ждать `N` секунд между запросами.

Графические интерфейсы для Wget

Приведем список некоторых графических интерфейсов для Wget:

- Gwget (рис. 42.2).
- GGet (рис. 42.3).
- Kiwi.
- Web.GET.
- KWebGet (рис. 42.4).
- KMAGO.

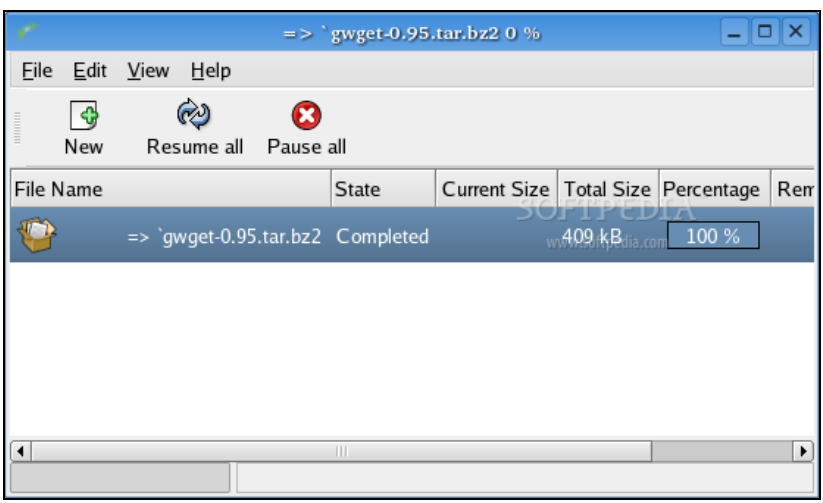


Рис. 42.2. Главное окно Gwget

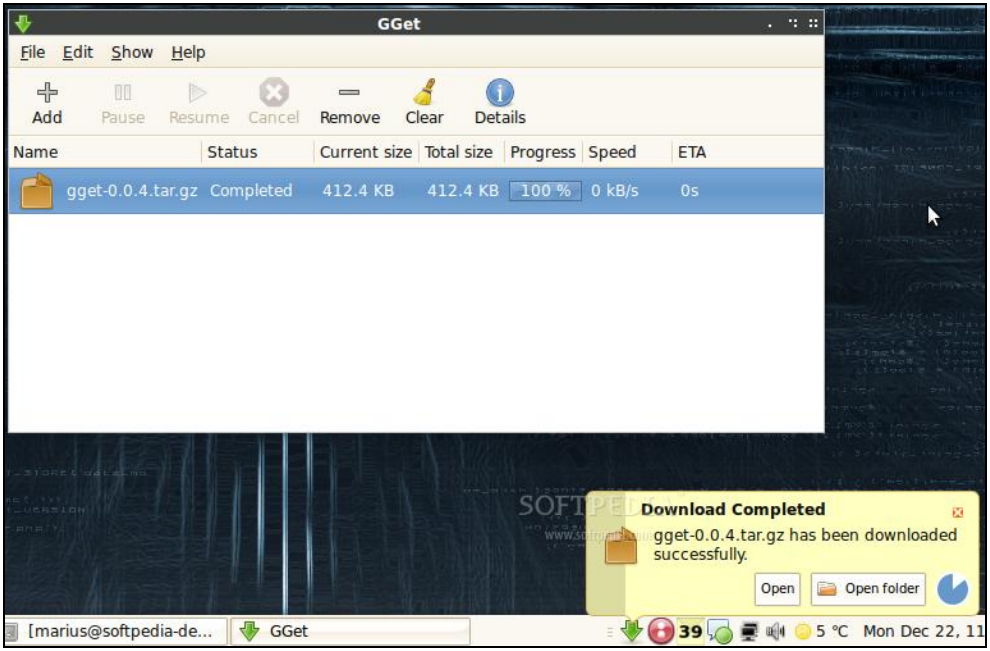


Рис. 42.3. Главное окно GGet

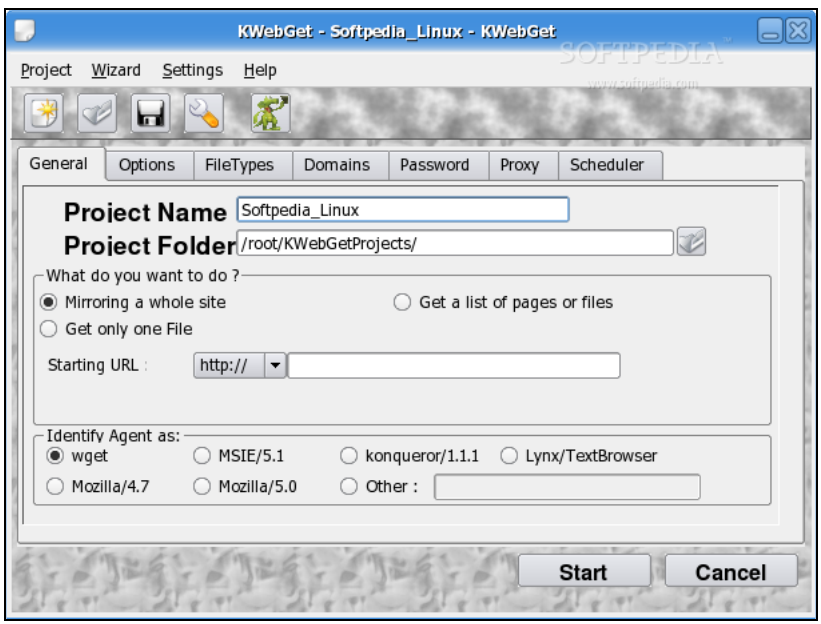


Рис. 42.4. KWebGet — создаем новую задачу

MLdonkey

MLdonkey — пакет, способный работать на компьютере в качестве демона, имеющий большое количество графических оболочек-клиентов, в том числе и с управлением через Web. Поддерживает одновременно много разных пиринговых сетей: ED2K (и Kademlia, и Overnet), BitTorrent, DC++, FastTrack, SoulSeek, Gnutella и G2. Управляется через интерфейс командной строки и при помощи скриптов.

Установка стандартная, либо из пакета, либо компиляция из исходных кодов. Для запуска достаточно в командной строке набрать `mlnet`.

Запуск в качестве демона осуществляет команда `daemon`. После запуска в браузере набираем адрес **http://localhost:4080/** и попадаем в Web-интерфейс, который позволяет вам сконфигурировать работу MLdonkey (рис. 42.5).

Как видно из рис. 42.5, нажав кнопку **Options**, мы вызываем окно настройки параметров программы. Здесь мы можем задать имя клиента, его IP-адрес, максимальное число скачиваемых и отдаваемых файлов, максимальное количество соединений, скорость отдачи и приема данных и т. п.

Нажав кнопку **Transfers**, мы увидим, какие файлы закачиваются и отдаются в данный момент. Само собой, в этом окне можно управлять закачками. Список всех настроек и конфигурационных файлов можно найти в документации к MLdonkey.

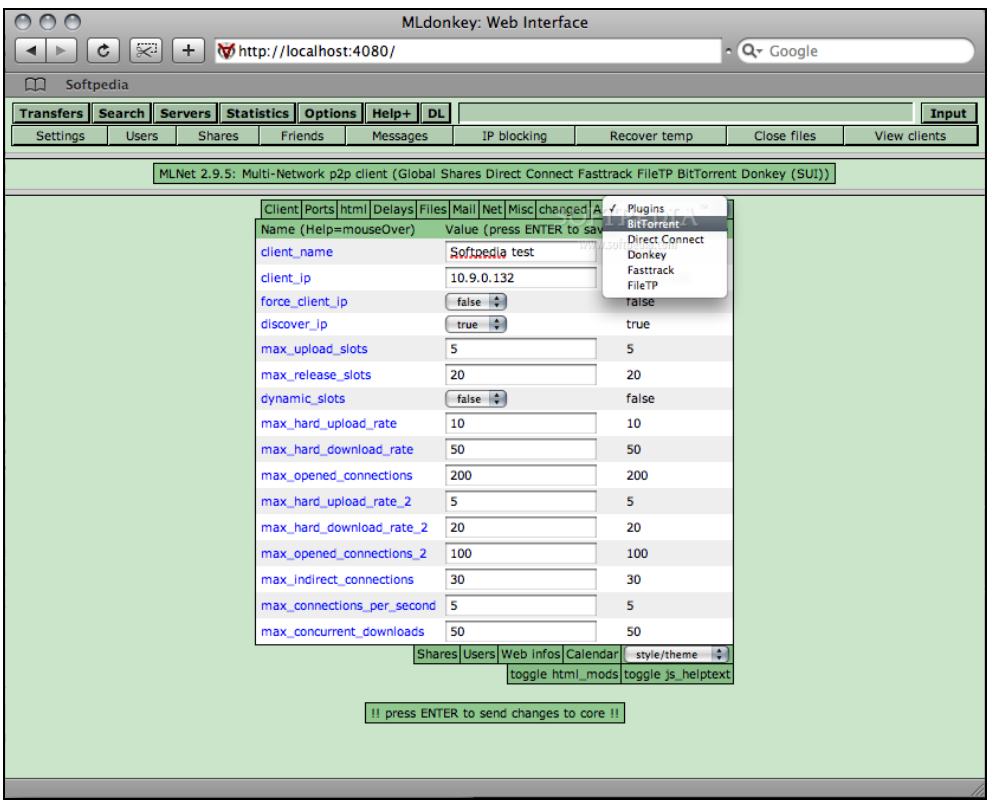


Рис. 42.5. Web-интерфейс MLdonkey; настройка параметров

Графические клиенты для MLdonkey

Существует пара десятков графических интерфейсов для MLdonkey.

Вот некоторые из них:

- Sancho — мультиплатформенный клиент, написанный на Java (рис. 42.6).
- KMLdonkey — графический KDE-клиент, написанный на C++.
- MLdonkeyGtkUi — "родной" графический интерфейс под GTK.
- CocoDonkey, xDonkey, mlMac — клиенты для Mac OS X, написанные на Cocoa.
- MLdonkeyWatch — клиент для MS Windows.
- Alemula, phpEselGui, saman, Zuul — клиенты, написанные на PHP.
- Platero — клиент для KDE.
- JMoule — очень простой клиент, написанный на Java.
- Web-GMUI — Web-клиент, включающий в себя простой Web-сервер.



Рис. 42.6. Sancho — просмотр ссылки

Существует еще один проект, подобный MLdonkey, — Hydranode, но, к сожалению, его развитие практически остановилось.

Transmission

Простой Torrent-клиент (рис. 42.7). Работает на нескольких Unix ОС, в том числе и на Mac OS X. Входит в некоторые дистрибутивы, например Debian, Fedora и Ubuntu.

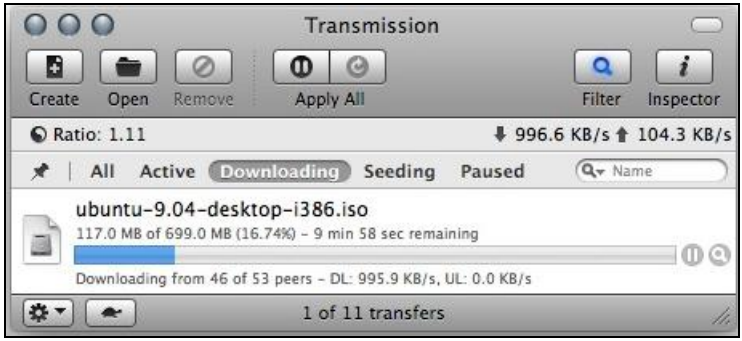


Рис. 42.7. Transmission — главное окно

Vuze

Кроссплатформенный Torrent-клиент, написанный на Java (рис. 42.8). Поддерживает несколько протоколов анонимного обмена данными. Есть возможность просмотра недокачанного видеофайла.

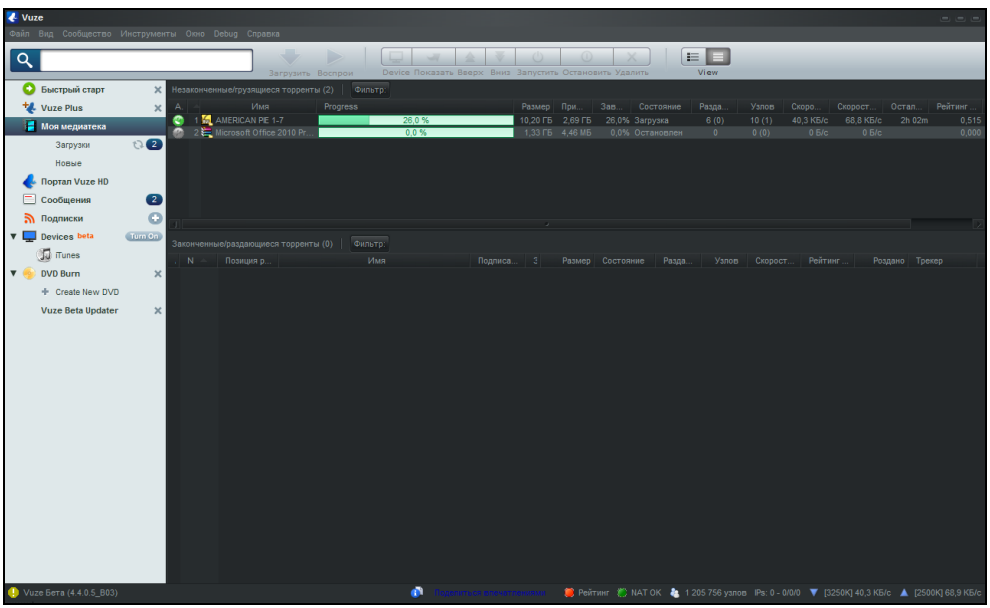


Рис. 42.8. Vuze — главное окно

Ссылки

vap.org.ru/storeforever/wget/ — статья "Wget — насос для Интернета".
http://mldonkey.sourceforge.net/ — страница проекта MLdonkey.

Глава 43



Действия в нештатных ситуациях

Эта глава посвящена решению проблем, возникающих с операционной системой либо с жестким диском. Большинство таких ситуаций необходимо учитывать при составлении планов восстановления. Как правило, подобные проблемы появляются неожиданно, и иметь соответствующие навыки по их решению необходимо, поскольку по известным законам неприятность возникает именно тогда, когда "упавшая" система нужна всем и немедленно. Бóльшая часть материала этой главы основывается на книге "Системное администрирование Linux".

Утрата пароля root

Обычно эта ситуация встречается при смене администратора либо когда в фирме работают несколько администраторов, а сопровождение системы поставлено плохо. Иногда такая неприятность случается, когда кто-то редактирует файл /etc/passwd вручную, а в это время отключают электроэнергию. Наконец, пароль можно просто забыть.

Существуют несколько вариантов восстановления пароля, которые мы сейчас и рассмотрим.

Восстановление без перезагрузки

Если на одной из виртуальных консолей остался открытым сеанс пользователя root (работать под пользователем root, вообще-то, вопиющее нарушение трудовой дисциплины и безопасности системы), восстановить пароль root проще простого — воспользоваться командой `passwd`. Можно отредактировать файлы passwd и shadow так, чтобы пользователя root пускали без пароля, и опять-таки воспользоваться командой `passwd`.

Перезагрузка в однопользовательском режиме

Для доступа к системным файлам можно перезагрузить систему в режиме одиночного пользователя.

Безопасную перезагрузку с системной консоли обеспечивает нажатие комбинации клавиш <Ctrl>+<Alt>+<Delete>. Однако не везде операционная система позволяет любому пользователю перезагрузить ее. Можно также попробовать команды

`reboot`, `poweroff`, `halt`, однако на серверах обычно операционную систему настраивают так, что перезагрузить ее может либо пользователь root, либо специально назначенный пользователь.

При компиляции ядра в него иногда включают средства поддержки функции Magic SysRq, предназначенной для принудительной перезагрузки компьютера или ввода команд `sync` и `unmount`.

Вызывать данную функцию удобно с помощью следующих комбинаций клавиш:

- <Alt>+<SysRq>+<s> — предпринимается попытка выполнить команду `sync` для всех смонтированных файловых систем;
- <Alt>+<SysRq>+<u> — последовательно выполняются команды `unmount` и `remount` в режиме "только для чтения";
- <Alt>+<SysRq>+<b> — выполняется перезагрузка системы.

Если перезагрузить систему с помощью упомянутых корректных методов все же не удастся, постарайтесь минимизировать разрушения файловых систем, которые могут возникнуть в связи с аварийным выключением системы. Лучше всего такие действия проводить тогда, когда для восстановления операционной системы у вас будет достаточно много времени, например по окончании рабочего дня или в выходные. При аварийном выключении компьютера файловые системы не будут корректно демонтированы, и после перезагрузки системы автоматически запустится программа fsck. На сервере с большим объемом дискового пространства выполнение программы может занять несколько часов.

После перезагрузки компьютера нам необходимо загрузить операционную систему в однопользовательском режиме. Как правило, для этого достаточно указать параметр `single`. При загрузке операционной системы с помощью программы LILO в командную строку нужно ввести команду

```
LILO: linux single
```

После этого ОС должна загрузиться в однопользовательском режиме. Однако иногда операционную систему настраивают так, что для ее перевода в однопользовательский режим необходимо указывать пароль пользователя root. При такой настройке для выполнения команды стартовых скриптов потребуется ввести соответствующий пароль. Чтобы решить эту проблему, можно воспользоваться следующим вариантом запуска операционной системы Linux:

```
LILO: linux init=/bin/sh
```

В результате нормальная загрузка ОС будет отменена, запустится только командная оболочка.

В приведенных примерах подразумевается, что указан пункт меню программы LILO, имеющий название linux. Узнать, какие, собственно, варианты загрузки существуют на вашем компьютере, можно в тот момент, когда программа LILO ожидает ввода команды. Список доступных для загрузки ядер можно вывести с помощью клавиши <Tab>.

Помимо описанных сложностей, вполне может оказаться, что ваш загрузчик операционной системы (LILO в данном случае) настроен на использование паролей, значения которых указываются в файле /etc/lilo.conf. Этот режим применяется для запроса специальных паролей при загрузке определенных конфигураций системы.

В этом случае без так называемой rescue-дискеты делать нечего. В большинстве дистрибутивов инсталляционный диск содержит вариант применения компакт-диска в качестве rescue-дискеты или есть LiveCD-диск.

Восстановление пароля root после перезагрузки

После успешной загрузки выполните команду `mount` для корневой файловой системы (при загрузке операционной системы с внешнего носителя или CD-ROM) или заново смонтируйте ее в режиме доступа "для чтения/записи" (при обычной загрузке в режиме single или init=/bin/sh).

Для монтирования корневой файловой системы в режиме доступа "для чтения/записи" можно воспользоваться командой

```
mount -o remount,rw /
```

Если схема размещения файловых систем вам незнакома, можно с помощью команды `fdisk -l` вывести список существующих дисковых разделов. Монтируя по очереди дисковые разделы, выясните, на каком из них находится каталог /etc.

Теперь можно удалить прежний пароль пользователя root путем редактирования файлов /etc/passwd или /etc/shadow. Однако иногда удобней изменить пароль, просто скопировав его из записи пользователя, пароль которого достоверно известен.

Если в системе применяется PAM или подобный ему пакет, запрещающий пустые пароли, скопируйте хэшированное значение некоторого известного вам пароля.

Если корневая файловая система жесткого диска была смонтирована в каталоге /mnt (при загрузке системы с аварийной дискеты), то с помощью команды `chroot` можно указать, с какой именно копией файла passwd должны работать такие команды, как `passwd`.

После того как корневая файловая система будет смонтирована в режиме доступа "для чтения/записи", с помощью команды `chroot` можно сделать ее корневой для всех процессов оболочки. В этом случае для установки нового значения пароля можно просто воспользоваться командой `passwd`. Данный вариант обеспечивает автоматическое корректное обновление любых файлов, используемых средствами синхронизированной аутентификации.

При повреждении файла /etc/passwd скопируйте его с любой машины, имеющей сходные параметры настройки, или с аварийной дискеты, а еще лучше — извлеките этот файл из резервной копии системы.

Устранение последствий атак хакеров

Это большая и сложная тема. Действия, которые следует предпринимать после различных нарушений защиты системы, описывались в *главе 8*, посвященной безопасности системы.

Приведем краткий алгоритм решения проблемы, вызванной атакой.

1. Отключить компьютер от Интернета и от локальной сети, поскольку зачастую целью атаки является либо получение информации, находящейся на атакованном компьютере, либо использование взломанной машины в качестве плацдарма для организации атак на другие компьютеры сети.

2. Переустановить "с нуля" операционную систему, безжалостно уничтожив всю информацию на жестком диске. В принципе, если вам позволяют время и возможности — желательно установить, как был произведен взлом операционной системы. Для этого необходимо действовать двумя путями:
   - на основании log-файлов операционной системы попытаться выяснить, когда и каким образом произошел взлом;
   - просмотреть на сайте производителя дистрибутива (в разделах errata, support, updates и т. д.) или в рассылках по безопасности (bugtraq, на сайте **www.securityfocus.org**) информацию обо всех найденных со времени выхода дистрибутива (или обновления вами соответствующих пакетов операционной системы) брешах в безопасности используемых вами пакетов. При обнаружении в пакетах различных "дыр", как правило, патчи этих пакетов выходят весьма оперативно — обычно это занимает день-два.

   Замечание

   Как правило, если взломщик достаточно квалифицированный, самое большее, что вы сможете понять на основании log-файлов (да и то приблизительно) — время взлома системы, поскольку "нормальные" взломщики подчищают log-файлы или вообще их удаляют. Если вы хотите все-таки докопаться до сути, то необходимо настроить ведение log-файлов так, чтобы их копии передавались по сети на отдельный компьютер, на котором будут храниться все log-файлы вашей сети.

3. Восстановить файлы с данными и параметрами настройки взломанной машины. Крайне не рекомендуется восстанавливать систему либо исполняемые файлы из резервной копии, поскольку вполне вероятно, что в последних резервных копиях находится уже взломанная система, а взломщики обычно "подсаживают" во взломанную операционную систему исполняемые программы, модифицированные специальным образом.

   Файлы настройки, содержимое которых должно отличаться от полученных во время установки операционной системы, необходимо тщательно проверить, поскольку многие из них содержат команды сценариев и различных программ, с помощью которых хакеры могут организовать атаки для получения доступа к системе. Лучший вариант — делать копии всех файлов настройки сразу же после их создания, а также после внесения в них любых изменений, причем копировать их не только на носитель информации, но и распечатывать на бумаге.
4. После переустановки и восстановления файлов настройки обязательно установить все обновления, касающиеся критически важных пакетов операционной системы.

Проблемы с загрузкой операционной системы

В процессе загрузки операционной системы принимают участие различные программные компоненты, поэтому существует несколько этапов, на каждом их которых могут возникать разнообразные проблемы. Будем последовательно решать каждую проблему.

Останов загрузки в процессе выполнения LILO

Во избежание появления большинства проблем, связанных с программой LILO после изменения конфигурационного файла lilo.conf, а также после установки нового ядра операционной системы или изменения конфигурации жестких дисков, всегда выполняйте команду `lilo`. Обязательно обращайте внимание на сообщения об ошибках или предупреждениях, которые программа LILO генерирует в процессе обновления главной загрузочной записи или карты расположения используемых при загрузке файлов.

Создайте загрузочную дискету на случай разрушения загрузочных карт LILO.

Сохраняйте копию файла /etc/lilo.conf, копии главных загрузочных записей (MBR) на специальной аварийной дискете, а лучше — на компакт-диске. Постоянно поддерживайте эти копии в актуальном состоянии, отражающем все вносимые в систему изменения.

Далее описаны ситуации, которые могут возникнуть при ненормальном функционировании программы LILO.

Программа LILO выводит последовательность 01010101010

Ядро и карты загрузки находятся на устройстве, которое не поддерживается средствами BIOS.

Программа LILO останавливается, выдав L

Первичный загрузчик LILO не может найти вторичный загрузчик этой программы. Кроме того, могут возвращаться коды ошибок жесткого диска, описанные в документации к `lilo`. Обычно эта ситуация означает, что программы BIOS и LILO по-разному определяют геометрию жесткого диска.

При возникновении такой проблемы проверьте, чтобы в BIOS был установлен режим LBA (если у вас жесткий диск достаточно больших размеров) и поместите директиву `linear` или `lba32` (для новых версий BIOS и больших дисков) в файл /etc/lilo.conf, после чего выполните команду `lilo` до повторной перезагрузки системы.

Проверьте, не превосходят ли размеры дисковых разделов максимальный размер раздела, поддерживаемый программами BIOS.

Обычно такого рода проблемы возникали на материнских платах для процессоров i486, Pentium и ранних материнских платах с поддержкой Pentium II.

Программа LILO останавливается, выдав LI

Вторичный загрузчик программы LILO был найден, но его не удалось корректно загрузить. Такое обычно случается при наличии проблемы с геометрией диска. Если реальный файл boot.b и его описание в карте загрузки не соответствуют друг другу, выполните команду `lilo` и перезагрузите компьютер.

Программа LILO останавливается, выдав LIL?

Вторичный загрузчик программы LILO не смог получить доступ к требуемому ему адресу. Проблема устраняется так же, как и в предыдущем случае.

Программа LILO останавливается, выдав LIL

Вторичный загрузчик программы LILO не смог прочитать системную карту. Выполните команду `lilo` и перезагрузите компьютер.

Программа LILO останавливается, выдав LIL-

Некорректная таблица дескрипторов системной карты. Обычно это означает, что файл /boot/map был разрушен или перемещен.

Проблемы с выполнением программы LILO

В процессе выполнения программы LILO могут встречаться самые различные проблемы. Рассмотрим наиболее распространенные из них.

Неверная сигнатура LILO

```
First boot sector doesn't have a valid LILO signature
```

Приведенное сообщение об ошибке обычно означает, что разрушен файл /boot/boot. Другой вариант — в файле /etc/lilo.conf директива `install` указывает на объект, который программа LILO не воспринимает как программу первоначальной загрузки.

```
Chain loader doesn't have a valid LILO signature
```

Приведенное сообщение об ошибке означает, что разрушен файл /boot/chain. Другой вариант — в файле /etc/lilo.conf директива `loader=` указывает на объект, который программа LILO не воспринимает как загрузчик цепочки.

Загрузка ядра Linux выполняется программой вторичного загрузчика boot.b, которая загружается в память программой первичного загрузчика, расположенного в главной загрузочной записи жесткого диска. Все имеющиеся в файле lilo.conf директивы `image=` обрабатываются исключительно вторичным загрузчиком.

Программа загрузки chain используется для загрузки MS-DOS и других операционных систем подобного типа. Она вызывается при загрузке версий операционных систем, задаваемых в файле lilo.conf директивами `other=`. Кроме того, специфические варианты загрузки машины могут осуществлять особые программы первоначальной загрузки, описываемые директивой программы `LILO loader=`.

BIOS не имеет доступа к жесткому диску

Иногда ваш компьютер может выдать следующее сообщение:

```
Warning: BIOS drive 0x82 may not be accessible
```

Оно появляется в тех случаях, когда некоторые из указанных в файле lilo.conf вариантов загрузки системы ссылаются на программы и операционные системы, расположенные на жестком диске, отличном от первых двух устройств, подключенных к первичному контроллеру. Иначе говоря, параметры в файле lilo.conf требуют выполнить загрузку с третьего устройства первичного контроллера или с устройства, подключенного к вторичному контроллеру.

Приведенное ранее сообщение программы LILO является просто предупреждением. Его появление не свидетельствует о нанесении вреда устанавливаемой системе.

Повреждение главной загрузочной записи (MBR)

Если главная загрузочная запись жесткого диска или таблица разделов повреждена, ее можно восстановить. Как правило, это не вызывает никаких нарушений в файловых системах или данных, размещенных на этом жестком диске.

Предварительно необходимо сохранить копию главной загрузочной записи и таблицы разделов на резервную дискету. Копию главной загрузочной записи и таблицы разделов можно сделать следующей командой:

```
dd if=/dev/hda of= hda-mbr.bin bs=512 count=1
```

Здесь:

- `/dev/hda` — ссылка на первый жесткий диск с интерфейсом IDE;
- файл `hda-mbr.bin` — файл, содержимое которого будет главной загрузочной записью жесткого диска;
- размер блока устанавливается равным 512 байтам;
- параметру `count` присвоено значение 1, поскольку требуется скопировать только один сектор данных.

Программа первичного загрузчика содержится в главной загрузочной записи только первичного жесткого диска (/dev/hda или /dev/sda). На всех остальных дисковых устройствах в этой записи будет содержаться только таблица разделов, а оставшаяся ее часть останется пустой. В любом случае полезно сохранять сведения о таблицах разделов всех жестких дисков компьютера.

Для восстановления главной загрузочной записи жесткого диска достаточно ввести команду

```
dd of=/dev/hda if=$BACKUP_FILE bs=512 count=1
```

Помимо использования команды `dd` и резервной копии главной загрузочной записи жесткого диска, таблицу разделов можно восстановить вручную, воспользовавшись информацией, предварительно сохраненной или распечатанной с помощью команды `fdisk -l`. Полезно сохранить копию этих данных непосредственно в процессе исходной установки системы. В результате вы всегда будете знать местонахождение этих данных, независимо от последующих перемещений системы.

Понятно, что когда потребуется воспользоваться созданной копией главной загрузочной записи жесткого диска, саму операционную систему загрузить не удастся. Поэтому копии всех резервных файлов главной загрузочной записи жесткого диска необходимо поместить на аварийные дискеты и, может быть, в корневую файловую систему вашего компьютера.

Поскольку полученный с помощью команды `dd` файл двоичный, вы не можете без особых ухищрений извлечь из него информацию о разбиении жесткого диска. Поэтому желательно хранить также текстовый вариант списка всех разделов диска (с указанием их размеров и расположения), причем еще и в распечатанном виде, предназначенном для чтения человеком. Этот список легко получить с помощью команд

```
fdisk -1
```

или

```
mount
```

В результате даже при отсутствии резервной копии главной загрузочной записи жесткого диска всегда можно будет восстановить таблицу разделов вручную. Восстановить вручную текст программы первоначальной загрузки можно с помощью программы LILO, выполнив команду `/sbin/lilo` с указанием корректного файла параметров /etc/lilo.conf.

Новое ядро операционной системы не загружается

Для обновления карты размещения файлов, используемых при загрузке, необходимо выполнить программу LILO. В этой карте содержится информация о точном расположении на жестком диске каждого из файлов, которые программа LILO задействует в процессе загрузки системы. В их число входит и файл ядра операционной системы.

Если до остановки процесса загрузки операционной системы система выводит сообщение `Loading Kernel...`, то, возможно, были неверно выбраны параметры компиляции ядра. В этом случае для определения и последующего устранения проблемы попробуйте воспользоваться параметрами ядра `reserve=` и `exclude=`, которые можно задавать в командной строке программы LILO.

Новое ядро выдает сообщение о превышении размера ядра

Некоторое время назад ядро ОС было компактным само по себе. В последние годы из-за серьезного увеличения функциональности ядра операционной системы загрузить несжатое ядро программа-загрузчик не в состоянии. Если вы пользуетесь устаревшими рекомендациями по компиляции — ваше новое ядро получится больше, чем может загрузить программа-загрузчик. Поэтому почитайте рекомендации по компиляции ядра на сайте фирмы-производителя дистрибутива. Обычно после компиляции из полученного ядра ОС создают его сжатый образ, который после загрузки распаковывается в оперативной памяти компьютера.

Кроме того, можно создать ядро меньших размеров за счет перемещения большего количества необходимых функций в отдельные загрузочные модули и отказа от вкомпилирования в ядро поддержки устройств, не установленных на вашем компьютере.

Ядро выдает сообщение о невозможности монтирования корневого каталога

В ядре Linux определено устанавливаемое по умолчанию устройство и раздел, на котором располагается корневая файловая система. Это значение, задаваемое прямо в исходном тексте ядра, можно изменять командой `rdev`. Существует еще несколько подобных значений по умолчанию, которые жестко записываются в ядро в процессе его компиляции. Их также можно менять с помощью различных параметров команды `rdev`, что позволяет избежать перекомпиляции ядра.

Если существующее имя корневой файловой системы не соответствует значению, установленному в ядре, то при попытке ее монтирования будет выдано упомянутое ранее сообщение. Самый простой способ изменить записанное в ядре и принимаемое по умолчанию значение — указать требуемое имя в параметре `root=`.

После успешной загрузки системы выполните команду `rdev` и/или модифицируйте файл /etc/lilo.conf, чтобы добавить в него директиву, например `append="root=hda2"`.

Помимо вышеприведенного случая, такое сообщение можно получить, если при компиляции ядра операционной системы драйверы устройства, на котором размещается корневая файловая система, не были вкомпилированы в ядро или были вынесены в загружаемый модуль, а поскольку загружаемые модули ядра грузятся с подмонтированного жесткого диска, возникает аналогичная ошибка.

Экран мерцает, и на нем отсутствует приглашение к регистрации в системе

Если рабочая станция настроена на использование при загрузке графического приглашения для регистрации пользователя, и на экране монитора заметны повторяющиеся безрезультатные попытки системы начать процедуру регистрации, проверьте состояние мыши.

Вначале выясните, подключена ли мышь к компьютеру. Затем вручную перезагрузите систему в режиме одного пользователя и убедитесь, что в каталоге /dev имеется соответствующий файл устройства. Затем попробуйте выполнить команду `gpm` — это позволит убедиться, что система знает о существовании мыши и может с ней работать. В противном случае вручную запустите команду `startx` и проанализируйте выводимые сообщения об ошибках. Проверьте состояние файлов настройки системы X Window.

Другая проблема, не позволяющая системе X Window нормально начать работу, может заключаться в отсутствии доступа к каталогу со шрифтами — локальному или расположенному на некотором сервере.

Если все упомянутые условия выполнены, то ошибка может заключаться в неверной настройке X Window — либо установлен не тот тип видеокарты, либо завышены частоты монитора.

Проблемы с запуском программ

В этом разделе рассматривается устранение проблем, возникающих при попытке запуска различных программ. Обычно такие проблемы возникают при неверно установленных правах доступа или отсутствующих системных библиотеках, необходимых данной программе.

Повреждение или удаление разделяемых библиотек

При повреждении разделяемых библиотек операционную систему, как правило, можно будет перезагрузить только с помощью аварийной загрузочной дискеты.

Поскольку работа всех компонентов операционной системы Linux полностью зависит от разделяемых библиотек, то при их отсутствии или повреждении ни одну из обычных команд и утилит выполнить невозможно. В последних версиях Linux лишь очень немногие программы связаны с библиотеками статически. Именно по этой причине стандарт File Hierarchy Standard (Стандарт иерархии размещения

файлов) требует, чтобы каталог /lib находился непосредственно в корневом каталоге, а также рекомендует избегать его использования в качестве точки монтирования.

Поскольку программы, задействованные в нормальном процессе остановки системы, также могут быть динамически связаны с системными библиотеками, то лучший способ безопасной перезагрузки систем — метод Magic SysRq, описанный ранее.

В противном случае потребуется перезагрузить машину с аварийной дискеты, после чего восстановить в системе корректные копии разделяемых библиотек.

Сообщение "getcwd: cannot access parent directories"

Такое сообщение выводится при переходе некоторого процесса в каталог с ограниченным доступом. Здесь этот процесс отменяет свои привилегии или вызывает функции `setuidO` или `setgidO` для объекта, который не имеет права доступа к одному из родительских каталогов, входящих в путь, ведущий в текущий рабочий каталог.

Как правило, в этом случае дочерний процесс, не имеющий необходимых привилегий, не может использовать команду `is` или даже команду `echo *`.

Чаще всего подобная ситуация возникает, когда некоторым пользователям присвоены неверные права по отношению к каталогу, ведущему к их основному каталогу.

Программа вызывает SIG11

Если программа сообщила, что было вызвано прерывание SIG11 и получен дамп ядра, это обычно означает, что в вашей системе проблемы с оборудованием.

Обычно такие ошибки возникают из-за модулей памяти, отдельные ячейки микросхем которых некорректно работают, причем эта проблема может не проявляться неделями. Реже подобную ошибку вызывает нестабильно работающая материнская плата.

Народное средство проверки нестабильной памяти — несколько раз подряд откомпилировать ядро ОС. Если попытка откомпилировать ядро операционной системы Linux завершится выдачей сообщения `Internal compiler error` со ссылкой на прерывание SIG11, причина, вероятнее всего, в ненадежной работе оперативной памяти.

В дистрибутиве Fedora есть один из вариантов загрузки установочного диска — запуск memtest86 — очень качественной программы для тестирования оперативной памяти. Необходимо поставить на 4–5 прогонов, чтобы гарантированно выявить нестабильность работы модулей памяти.

К сожалению, в современных микросхемах оперативной памяти чрезвычайно трудно обнаружить непостоянные отказы. Компьютеры и операционные системы настолько сложны, что простая последовательность операций "запись, чтение, проверка" в оперативной памяти едва ли будет пригодна для выявления проблем с оборудованием.

Если предполагается, что ошибка связана с оборудованием, попробуйте установить в компьютер другие модули памяти.

Превышение максимального числа открытых файлов

Ядро имеет ограничение, связанное с максимальным числом одновременно открытых файлов, которое задается при компиляции ядра ОС. Достижение этого предела приводит к тому, что операционная система отказывает в открытии файла.

Изменить текущее значение этого параметра можно, отредактировав псевдофайлы /proc/sys/kernel/file-max и /proc/ sys/kernel/inode-max.

Пример:

```
inode-max = 32768 file-max .=5.120
```

Два параметра системы — максимальное число задач в системе и предельное число задач для одного пользователя — переопределяются при компиляции ядра. Эти значения задаются в файле параметров ядра.

Проблемы с файловыми системами

Далее речь пойдет об устранении различных проблем, которые возникают при работе с файловыми системами.

Ошибка "unable to find swap-space signature"

Подобная ошибка может возникнуть, когда одно и то же дисковое пространство страниц виртуальной памяти используется одновременно несколькими операционными системами, либо была повреждена таблица swap-раздела.

При появлении такой ошибки необходимо воспользоваться командой `fdisk` для повторной проверки типов разделов, описанных в таблице разделов диска. Убедившись, что все выполненные для разделов назначения корректны, введите команду `mkswap`.

Переполнение файловой системы

При заполнении всего дискового пространства, выделенного файловой системе, за пользователем root резервируется некоторый свободный объем дискового пространства. Как разумно предусмотрели разработчики файловой системы, администратору и некоторым утилитам необходимо наличие некоторого пустого дискового пространства для нормальной работы с переполненным разделом.

Разрешение на работу с этим резервным пространством можно предоставить отдельному пользователю или группе пользователей при помощи утилиты tune2fs.

Очевидное решение этой проблемы — удаление некоторых файлов либо архивирование редко используемых файлов.

Если пользователь root или процесс, запущенный с правами пользователя root, вызовет переполнение диска, начнется заполнение резервного пространства диска. Поэтому почту для пользователя root всегда нужно посылать на учетную запись, не имеющую особых привилегий, а ротацию файлов журналов тщательно контролировать.

Для предупреждения случаев переполнения файловых систем целесообразно предусмотреть какую-либо программу мониторинга состояния операционной системы.

Переполнение числа блоков индекса файловой системы

Переполнение числа блоков индекса файловой системы возможно даже в том случае, когда основное пространство файловой системы еще не заполнено. Этот показатель не имеет отношения к параметру ядра, описывающему максимальное число одновременно открытых блоков индексов. Если файловая система содержит много файлов размером менее 4 Кбайт, то все блоки индекса такой файловой системы могут оказаться заполненными раньше, чем ее основное пространство.

Отношение числа блоков индекса к числу блоков данных любой заданной файловой системы устанавливается при ее создании (параметр `-i` команды `mke2fs`). Файловые системы, предназначенные для размещения спула групп новостей, всегда должны иметь увеличенное количество индексных блоков.

Подозрение на наличие сбойного кластера или сектора

Если вы заподозрили, что на вашем жестком диске появились сбойные кластеры, можно запустить утилиту для проверки жесткого диска на наличие сбойных секторов. Эту операцию необходимо проводить в то время, когда никто не работает с компьютером, поскольку она может занять много времени.

Выявить сбойные блоки и поместить сведения о них в соответствующий список файловой системы типа Ext2 позволяет команда `e2fsck -c`.

При выполнении команды *mount* доступ к системе блокируется

В некоторых случаях выполняемый процесс может "зависнуть", если команда `mount` применяется к файловой системе на устройстве, не отвечающем на запросы системы. Кроме того, подобная ситуация иногда возникает при обращении к устройствам активной SCSI-цепочки, которые отсоединены или выключены.

Похожие ситуации наблюдаются и при переключении на другие виртуальные консоли, регистрации через последовательные терминалы или соединения telnet и т. п. Если запустить утилиту ps, то подобные "подвешенные" процессы отмечаются как находящиеся в состоянии D. Выполнение для такого процесса команды `kill -9` не оказывает на него никакого влияния, поскольку обработка сигналов блокируется на все время, пока процесс ожидает завершения выполнения подпрограммы системного вызова ядра ОС.

В подобном состоянии операционная система может находиться сколь угодно долго, причем она будет нормально функционировать до тех пор, пока не возник-

нет попытка обращения к "подвешенному" процессу или устройству. Чтобы выйти из этого положения, необходимо корректно завершить все процессы операционной системы (которые не находятся в "подвешенном" состоянии), после чего компьютер можно будет перезагрузить.

Случайное удаление файла

Если все ссылки на файл и все связанные с ним блоки обработки уже удалены, то после закрытия последнего открытого для него дескриптора занятое файлом пространство становится доступным для системного драйвера сборки мусора. Как только этот драйвер очистит занимаемое ранее файлом пространство, файл окажется утраченным навсегда.

В состав Linux включен документ "Undelete HOWTO" и несколько редакторов шестнадцатеричных данных. В частности программы ext2ed и debugfs предоставляют некоторые инструменты, которые могут оказаться полезными при устранении проблем подобного рода.

Можно воспользоваться программой mc (Midnight Commander). Для этого запускаем `mc` и в командной строке набираем `cd /#undel:/hda`. В результате получаем панель, в которой находится список удаленных файлов, причем имя файла — номер inode. Эти файлы можно просмотреть и, выбрав нужный, восстановить.

Разрушение данных

Команда `fsck` предназначена для проверки и восстановления файловых систем. Восстановленные блоки индекса помещаются в зарезервированный каталог lost+found, который существует в каждом физическом разделе Ext2(3,4).

Если резервной копии данных не существует, можно попробовать разобраться в каталоге lost+found и попытаться вручную восстановить данные.

Проблемы с сетью

В этом разделе рассматривается устранение проблем, которые возникают при некорректной настройке, неправильном функционировании или повреждении сети.

К системе нет доступа из сети

Проверьте значения параметров TCP, содержащихся в файлах /etc/hosts.allow и /etc/hosts.deny. Кроме того, проверьте все другие аспекты организации работы брандмауэра, которые применимы к данной машине. Проконтролируйте состояние сетевого кабеля в тех точках, в которых он подключается к машине и к остальной части сети.

Используйте утилиту ping для проверки функционирования сети.

Проблемы ввода/вывода данных

Во многих приложениях можно устанавливать комбинации клавиш, предназначенные для вызова специальных функций. Если проблема с вводом возникает только в одной программе (например, emacs), то назначить комбинации клавиш можно с помощью команд этого же приложения.

Любой текст воспроизводится в виде двоичных символов

Чаще всего подобная ситуация возникает при использовании простых утилит, предназначенных для чтения двоичных файлов. Терминал воспринимает одну или более двоичных комбинаций как команду изменения символического шрифта. В результате прочесть выводимые на экран сообщения невозможно. Введите команду `reset`, не обращая внимания на то, что будет выведено на экран. В результате все параметры терминала будут приведены к значениям, принимаемым по умолчанию.

Система не реагирует на команды, вводимые с клавиатуры

Убедитесь, что клавиатура подключена к компьютеру правильно, а не, скажем, к порту мыши. Если доступ к машине через сеть все еще возможен, то с помощью команды `loadkeys -d` восстановите карту ключей клавиатуры, заданную по умолчанию. В противном случае не избежать перезагрузки системы со всеми вытекающими последствиями.

Переназначение клавиш

Утилита xmodmap предоставляет средства переназначения клавиш клавиатуры. Однако внесенные изменения остаются в силе только на время сеанса X Window. Для изменения раскладки клавиатуры при работе с текстовой консолью следует вызвать утилиту loadkeys.

Окно сеанса X Window не воспринимает команд с клавиатуры и сигналов мыши

В среде X Window был выдан запрос, захвативший фокус ввода. Если выдавшее его приложение или задача "зависнет", менеджер окон окажется заблокированным, и любой ввод, направленный в среду X Window, будет игнорироваться.

Для решения этой проблемы необходимо получить доступ к компьютеру по сети либо через последовательный терминал и после этого выполнить команду `kill -9` для заблокированного задания. Если этого окажется недостаточно, продолжайте указанную процедуру, поднимаясь по соответствующему дереву процессов. В худшем случае остановка процесса X-сервера вынудит процесс init "собрать мусор" в ресурсах всех порожденных им процессов. Как правило, "убиения" заблокированного процесса или его родителей бывает достаточно для разблокирования устройства ввода.

Прочие аварийные ситуации

Некоторые аварийные ситуации нельзя отнести к какой-нибудь конкретной категории. О них мы и поговорим в данном разделе.

Не работает устройство, подключенное к параллельному порту

К параллельному порту в настоящее время подключают различные периферийные устройства: принтеры, сканеры, CD-RW, ZIP Drive и многие другие. Если в вашей операционной системе для устройств, подключаемых к параллельному порту, применяются загружаемые модули, то вы должны с помощью команды `lsmod` проверить, соответствует ли загруженный модуль тому типу устройства, которое в данный момент подключено к параллельному порту.

Настоятельно рекомендуется не предпринимать попыток выгрузить модули до тех пор, пока не будет демонтирована файловая система, связанная с данным устройством

После увеличения объема оперативной памяти система работает нестабильно

Некоторые старые материнские платы занимают для собственных нужд небольшой блок ячеек у верхней границы оперативной памяти. Попробуйте указать параметр ядра `mem=xxxM`, где значение `xxx` — на один мегабайт меньше полного объема установленной в компьютере оперативной памяти.

После увеличения объема оперативной памяти система не видит добавленную память

Некоторые старые материнские платы (в основном для Pentium и ранние платы для Pentium II) страдают подобным недостатком. Для исправления ситуации можно указать параметр ядра `mem=xxxM`, где значение `xxx` — полный объем установленной оперативной памяти. Если операционная система покажет вам полный объем оперативной памяти, но будет вести себя нестабильно, воспользуйтесь советом из предыдущего раздела.

Ссылки

- **http://www.bitwizard.nl/sig11** — "SIG11 Problem". Описание проблемы SIG11 и пути ее решения.
- Соответствующие HOWTO:
  - Multiboot Using LILO mini-HOWTO;
  - LILO mini-HOWTO.



Приложения



Приложение 1

Дополнительная литература

Здесь приведен список рекомендуемой литературы. За последнее время число книг, заслуживающих внимания, увеличилось. Что отрадно, появились книги отечественных авторов. Одни работы интересны глубоким освещением теоретической части, другие более привлекательны для практиков. Бóльшая часть книг, как нам представляется, предназначена для специалистов среднего и высокого уровня.

- Немет Э., Снайдер Г., Сибасс С., Хейн Т. Р. UNIX: руководство системного администратора. Для профессионалов: пер. с англ. — 3-е изд., перераб. и доп. — СПб.: Питер; Киев: Издательская группа BHV, 2007.
  Третье существенно переработанное издание книги. Уменьшено число рассматриваемых операционных систем, и, что очень важно, среди них появилась Linux. Эта книга должна быть на столе у каждого администратора.
- Карлинг М., Деглер С., Деннис Дж. Системное администрирование Linux: учеб. пособие; пер. с англ. — М.: Издательский дом "Вильямс", 2000.
  Хорошая книга по администрированию системы. Некоторые вопросы изложены неглубоко, однако направление поиска задают верно. Очень рекомендуется для системных администраторов и специалистов IT.
- Зиглер Р. Брандмауэры в Linux: учеб. пособие; пер. с англ. — М.: Издательский дом "Вильямс", 2000.
  Книга полностью посвящена построению защищенной сети. Рекомендуется для системных администраторов.
- Максвел С. Ядро Linux в комментариях: пер. с англ. — Киев: ДиаСофт, 2000.
  Содержание этой книги понятно из названия. С выходом ядер версии 2.4 информация несколько устарела, однако книга весьма полезна для ознакомления с общей идеологией ядра операционной системы Linux.
- Робачевский А. М. Операционная система UNIX. — СПб.: БХВ — Санкт-Петербург, 2002.
  Хорошая теоретическая книга. В ней рассмотрены принципы функционирования операционной системы, основные понятия, протокол TCP/IP и многое другое.
- Шевель А. Linux. Обработка текстов. Специальный справочник. — СПб.: Питер, 2001.
  Книга посвящена текстовым редакторам и системе CVS-управления и контроля версий текстов (исходных кодов программного обеспечения).

- Блам Р. Система электронной почты на основе Linux: учеб. пособие; пер. с англ. — М.: Издательский дом "Вильямс", 2001.
  В книге рассматриваются настройка почтового сервера и конфигурирование почтовых клиентов. Хорошая книга для освоения начал работы с почтовыми сообщениями и построения простых почтовых серверов.
- Стахнов А. Сетевое администрирование Linux (+ CD) — СПб.: БХВ — Санкт-Петербург, 2004.
  Книга предназначена администраторам и посвящена вопросам сетевого администрирования.
- Колесников О., Хетч. Б. Linux. Создание виртуальных частных сетей (VPN). — Кудиц-образ, 2004.
  В книге освещаются наиболее популярные из существующих VPN-технологий для платформы Linux. В начальных главах обсуждаются теоретические аспекты VPN, включая требования и области применения. Также охватываются конфигурации обычных сетей и хостов. В последующих главах более детально рассматриваются реализации и конфигурирование конкретных программных пакетов.
- Костромин В. А. Самоучитель Linux для пользователя. — СПб.: БХВ — Санкт-Петербург, 2003.
  Книга от создателя "Виртуальная энциклопедия LINUX по-русски" (**http://rus-linux.net**).
- Птицын К. Серверы Linux. Самоучитель. — М.: Издательский дом "Вильямс", 2003.
  Небольшая книга, предназначенная для начинающих. В краткой форме рассмотрены основные моменты установки и настройки сервера на базе дистрибутива Red Hat.
- Манн С., Митчелл Э., Крелл М. Безопасность Linux. — 2-е изд.; пер. с англ. — М.: Издательский дом "Вильямс", 2003.
  В книге рассказывается об инсталляции, конфигурировании и сопровождении Linux-систем с точки зрения безопасности. Это руководство администратора по реализации стратегии защиты Linux, а также по утилитам защиты, существующим в Linux. Книга предназначена для пользователей средней и высокой квалификации.
- Немет Э., Снайдер Г., Хейн Т. Руководство администратора Linux; пер. с англ. — М.: Издательский дом "Вильямс", 2007.
  Хит от известных авторов, рекомендую. В книге рассмотрены три основных дистрибутива Linux: Red Hat, SuSE и Debian. Это одна из немногих книг, предназначенных не для широкого круга пользователей, а для системных администраторов, работающих в среде Linux.
- Далхаймер М., Уэлш М. Запускаем Linux. — 5-е изд.; пер. с англ., Символ, 2008.
  Хорошая книжка для начинающих, доступным языком описаны основные проблемы и решения по установке и настройке системы.
- Водолазкий В. Путь к Linux: учебный курс. — 3-е изд. — СПб.: Питер, 2002.
  Книга от мэтра! Впервые была прочитана автором в 1996–97 гг. Несмотря на такой почтенный возраст, книга остается современной и полезной для начинающих пользователей



Приложение 2

Ссылки

Здесь собраны тематические ссылки на Web-ресурсы, так или иначе связанные с операционной системой Linux. Приведенный список далеко не полон, вряд ли он охватывает даже малую часть таких ресурсов. Создателей Web-ресурсов прошу не обижаться — всех удовлетворить невозможно, да и Интернет необозримо велик. Если кто-либо знает отсутствующий в списке полезный ресурс — сообщите нам.

Дистрибутивы

- **www.altlinux.ru** — сайт дистрибутива AltLinux. Популярный российский дистрибутив.
- **www.asplinux.ru** — сайт дистрибутива ASPLinux. Популярный российский дистрибутив.
- **www.debian.org** — сайт дистрибутива Debian. Один из распространенных дистрибутивов Linux. В его состав входит огромный набор программных пакетов. Одно из несомненных достоинств — постоянная поддержка дистрибутива разработчиками. Несомненным плюсом для русскоговорящих пользователей является Web-сайт на русском языке.
- **iso.linuxquestions.org** — специальный сайт, содержащий ISO-образы дистрибутивов.
- **www.redhat.com** — сайт фирмы Red Hat, производителя одноименного дистрибутива. Одно из достоинств данного дистрибутива — его хорошая поддержка, начиная с версии 4.*x* (создан в 1995 году) и заканчивая текущим. Дистрибутив Red Hat очень широко распространен, его уже начали сравнивать с Windows. В настоящее время Red Hat стал стандартом де-факто для производителей коммерческого программного обеспечения и компьютерного оборудования. Дистрибутив существует в трех вариантах: базовый (доступен для скачивания через Интернет), Professional и Advanced Server. Кроме собственно дистрибутива на сайте присутствует и достаточно большой объем качественно написанной документации (на английском языке).
- **fedoraproject.org/ru/** — сайт дистрибутива Fedora Core — реинкарнация пользовательской версии Red Hat-дистрибутива.

- **www.slackware.com** — сайт дистрибутива Slackware. Один из старейших дистрибутивов. Считается, что он достаточно сложен в инсталляции и настройке. Однако его несомненное достоинство — действительно понятный и логичный набор пакетов, а также возможность создать малую по размеру установленную операционную систему.
- **www.novell.com/linux/** — сайт дистрибутива SuSE. Несомненное достоинство — большой набор программных пакетов, входящих в состав дистрибутива.
- **www.ubuntu.com** — сайт дистрибутива Ubuntu. Ориентирован на конечных пользователей. Существует несколько вариантов дистрибутива с различной функциональностью.

Документация

- **dc.internic.net/rfc/rfc2196.txt** — документ, посвященный политике безопасности системы.
- **www.bog.pp.ru** — великолепный сайт Сергея Богомолова. Статьи по установке и настройке множества программ.
- **www.citforum.ru** — огромное собрание русскоязычной документации и книг, в том числе и посвященных Linux. Ресурс интересен разнообразием предоставляемой информации: программирование, базы данных, описание протоколов, обзоры программ и многое другое.
- **dc.internic.net/rfc** — стандарты RFC.
- **www.lib.ru** — знаменитая библиотека Мошкова.
- **www.linuxfocus.org** — электронный журнал "LinuxFocus", есть русский перевод.
- **www.linux.org.ru** — один из основных русскоязычных сайтов, посвященных Linux. На сайте собрана различная информация, тем или иным образом касающаяся Linux и программного обеспечения для этой операционной среды. Рекомендуется всем пользователям Linux.
- **www.linuxrsp.ru** — неплохой русскоязычный сайт.
- **linuxtv.org** — сайт, посвященный телевидению и Linux.
- **www.Linux-USB.org** — сайт, посвященный USB-устройствам и их применимости с точки зрения Linux.
- **www.opennet.ru** — очень хороший сайт, посвященный операционным системам и сетям.
- **www.pathname.com/fhs/** — Filesystem Hierarchy Standard в различных текстовых форматах.
- **www.redhat.com** — в разделе "Документация" содержится документация на дистрибутив и некоторый объем дополнительной информации.
- **www.rfc-editor.org** — сайт, посвященный стандартам RFC.
- **www.rpm.org** — сайт, полностью посвященный RPM.
- **www.tldp.org** — Linux Documentation project. Англоязычный сайт, содержащий в структурированном виде документацию по Linux.

Программное обеспечение

- **acl.bestbits.at** — официальная страница проекта Linux ACLs (Access Control Lists).
- **www.amanda.ocg** — сайт программы AMANDA.
- **www.apache.org** — официальный сайт Apache.
- **www.eecis.udel.edu/~ntp** — страница, посвященная серверу точного времени XNTP.
- **www.freshmeat.net** — сайт, содержащий очень много программ для Linux.
- **www.gnokii.org** — официальная страница проекта Gnokii.
- **www.gnome.org** — официальный сайт GNOME.
- **www.gnu.org/software/grub/** — домашняя страница программы-загрузчика GRUB.
- **www.idsoftware.com** — сайт компании-разработчика игр Doom, Quake, Quake II, Quake III. Эта коммерческая фирма одна из первых (если не первая) выпустила почти все свои игры для Linux, причем исходные коды нескольких игр фирма открыла для свободного доступа.
- **www.isc.org/products/INN** — официальный сайт сервера новостей INN.
- **www.kde.org** — официальный сайт KDE.
- **www.kdevelop.org** — официальный сайт среды программирования kdevelop.
- **koffice.kde.org** — официальный сайт офисного пакета Koffice.
- **www.kernel.org** — сайт ядра операционной системы Linux.
- **www.lids.org** — сайт проекта LIDS.
- **www.linmodems.org** — сайт, посвященный Win-модемам и драйверам для них под Linux.
- **linux.freeware.ru** — программное обеспечение для Linux.
- **www.lirc.org** — страничка проекта Linux Infrared Remote Control, LIRC.
- **www.mostang.com/sane** — официальная страница пакета SANE.
- **www.mozilla.org** — официальный сайт Mozilla. Кросс-платформенный Web-браузер с открытым исходным кодом Mozilla, основан на движке Gecko фирмы Netscape. Динамично развивающийся проект. На сегодняшний день код практически очищен от ошибок. На основе кода Mozilla разрабатываются несколько альтернативных Web-браузеров.
- **www.mrtg.org** — официальный сайт пакета MRTG.
- **www.mysql.org** — официальный сайт SQL-сервера MySQL. Основными задачами разработчики поставили быстродействие, простоту реализации и нетребовательность к ресурсам. К сожалению, в этом сервере не реализовано много возможностей языка SQL, в частности хранимые процедуры. Основной нишей MySQL являются простые проекты баз данных и хранилище информации для Web-сайтов.
- **www.opendivx.org** — сайт кодека DivX с открытым исходным кодом.
- **www.openoffice.org** — официальный сайт офисного пакета Open Office.
- **www.openssh.com** — сайт некоммерческой реализации SSH.
- **www.opera.com** — сайт фирмы-разработчика Web-браузера Opera. Очень неплохая кросс-платформенная альтернатива Web-браузеру Mozilla: легкий

и удобный. Единственное "но" — это коммерческий продукт с закрытым исходным кодом.

- **www.psionic.com** — сайт Psionic Software — разработчика программы Portsentry.
- **rpmfind.net** — репозиторий и поисковая система RPM.
- **rrdtool.eu.org** — официальный сайт пакета rrdtool.
- **rufus.w3.org/linux/RPM** — репозиторий RPM.
- **www.samba.org** — официальный сайт проекта Samba.
- **www.slug.org.au/etherboot/** — страница пакета Etherboot, предназначенного для загрузки бездисковых станций.
- **www.squid-cache.org** — официальный сайт программы Squid.
- **stunnel.mirt.net** — официальный сайт пакета Stunnel.
- **www.tripwire.org** — сайт разработчиков Tripwire.
- **www.vmware.org** — официальный сайт проекта VMWare.
- **www.webmin.com** — официальный сайт проекта Webmin.
- **www.winehq.org** — официальный сайт проекта Wine.
- **xmms.org** — сайт программы XMMS для воспроизведения аудио и видео.
- **www.xsane.org** — официальный сайт Xsane.

Безопасность

- **linuxsecurity.com** — сайт, посвященный безопасности операционной системы Linux.
- **www.security.nnov.ru** — сайт, посвященный компьютерной безопасности.
- **www.rootshell.com** — англоязычный сайт, посвященный компьютерной безопасности.

Предметный указатель

/

/ 83
/bin 83, 84
/boot 83, 85
/dev 83, 86
/etc 83, 86
/etc/bashrc 145, 147
/etc/fstab 145
/etc/initscript 146
/etc/inittab 136
/etc/issue 146
/etc/motd 146
/etc/profile 146
/etc/rc.d 93
/etc/rc.d/init.d 94
/etc/skel 145
/etc/sysconfig 94
/home 83, 102
/init.d 93
/lib 83, 102
/lib64 102
/lost+found 83, 102
/media 83, 103
/mnt 83, 103
/opt 83, 103
/proc 83, 103
/root 83, 108
/sbin 83, 108
/sys 83, 109
/tmp 83, 109
/usr 83, 110
/usr/bin 110
/usr/local 110
/usr/share/man 111
/usr/src 113
/usr/src/Linux-x.y.z 113
/var 83, 114
/var/cache 115
/var/lock 116
/var/log 116
/var/mail 116
/var/run 117
/var/spool 117
/var/tmp 117

A

AboutTime 409
ACL (Access Control List) 388, 468
ACM 274
Active Directory 429
AIDE 490
AltLinux 37
Apropos 235, 264
APT 193
ASCII 272
ASP Linux 37
at 252, 465
atq 253
atrm 253

B

background 453
badblocks 73
banner 265
bash 265
batch 253, 465
bc 265
BestLinux 38
BGP (Border Gateway Protocol) 53
Bonzai 37
BOOTP 230

C

cal 240
callback-сервер 579
Canonical NAME 302
cat 243

cd 243
Cedega 655
chat 552
chattr 156
chgrp 241
chkrootkit 482
chmod 241
chown 242
chroot 243
chvt 265
CIPE 157
Ckconfig 358
cksum 253
Clam AntiVirus (ClamAV) 162
clear 265
CNAME 302
control-panel 143, 148
Cool Linux 40
cp 244
CP1251 273
CP866 273
cpp 265
cron 465
crond 253
crontab 253
CrossOwer Office 656
csh 265
ctlinnd 366
CUPS (Common UNIX Printing System, общая система печати для UNIX) 610

D

date 240
DEB 188
DEB-пакеты 192
Debian 36, 41
debugfs 77
dhclient.conf 313
dhclient.leases 314
DHCP (Dynamic Host Configuration Protocol) 230, 305
dhcpd.conf 308
dhcpd.leases 311
DHCP-клиент 313
DHCP-сервер 308
dial on demand 552
diald 552, 557
dial-in-сервер 579
dig 249
dir 244
DivX 665
DNS (Domain Name System, доменная система имен) 293
Domain Name Service 45
DOSEmu 650
du 259
dumpkey 260

E

e2fsck 77
Eagle Linux 40
echo 265
EGP (Exterior Gateway Protocol) 53
eject 230
elm 249
Embedded Linux 10, 12
Embedded Windows 9
env 265
Ext 69
Ext2 69
ext2ed 77
Ext3 69
Extended Attributes 469

F

fdformat 263
fdisk 263
file 244
Filesystem Hierarchy Standard 82
find 244
finger 249
Firewall 162, 508
FlScan 633
foreground 453
free 260
FreeBSD 11
FreeS/WAN 446
fsck 72, 263
fstab 71
FTP (File Transfer Protocol) 45, 345
ftp 249
Ftpaccess 351
Ftpconversions 356
ftpcount 260
Ftpcount 358

Ftpd 357
Ftpgroups 356
Ftphosts 357
Ftprestart 358
Ftpservers 356
Ftpshut 358
Ftpusers 357
ftpwho 260
Ftpwho 358

G

g77 266
gawk 266
gcc 266
Gentoo 39
getkeycodes 253
getty 146, 249
Gnokii 635
GNOME 606
gnome-pilot 637
GnoRPM 190, 192
GPG (GNU Privacy Guard) 321
GQ 423
Group Descriptors 76
GRUB 120
gzip 263

H

head 245
HINFO (Host INFOrmation) 302
host 249
hostname 250
HOWTO 236

I

ICMP (Internet Control Message Protocol) 56
id 266
ifconfig 253
IGRP (Interior Gateway Routing Protocol) 53
init 135
INN 366
innd 366
innwatch 371
insmod 254
Internet Cache Protocol 383
IP (Internet Protocol) 45
ipchains 250
Ipchains 515
IPSec (Internet Protocol Security) 446
iptables 250, 515
IPv4 50
IPv6 51
Isapnp 254
isapnptools 231
ISO 8859-*x* 272

J

joe 264
J-Pilot 637

K

kdb_mode 260
KDE 607
Kerberos 158
kill 254, 456, 462
killall 254, 456, 463
killproc 145
Kldap 423
klogd 479
Knoppix 37, 39
KOI8-R 273
KOI8-U 273
Kpackage 190
Kpilot 637
kppp 250
Ksamba 444
kudzu 231
KVM 658
kWinTV 642

L

last 260
Latin 0 272
Latin 1 272
LCP (Link Control Protocol, протокол управления соединением) 571
LDAP (Lightweight Directory Access Protocol) 412
LDAP Data Interchange Format 418
Ldapadd 423
Ldapdelete 422
Ldapmodify 423

Ldapsearch 422
LIDS (Linux Intrusion Detection/Defense System) 483
lilo 254
LILO 120
Linux 11
linuxconf 143, 231, 255
LIRC (Linux Infrared Remote Control) 643
ln 245
LNX-BBC 40
LoadLin 120
locate 245
lockfile 243
login 146, 266
logname 266
logrotate 232
LPD (Line Printer Daemon, демон линейной печати) 610
LPRng 613
ls 245
lsattr 156
Lycoris 38
lynx 250

M

Mac OS X 10
macntp 409
mail 250
make 266
man 235, 264
Mandrake 38
mc 246
md5sum 255
mgetty 249, 577
Midnight Commander 188
mimencode 250
minicom 250
minix 69
mkdir 246
mke2fs 77
mkfifo 247
mkfs 263
mknod 243
modprobe 256
mount 71, 256
MoviX 40
MPPE (Microsoft Point-To-Point Encryption) 452
MRTG (Multi Router Traffic Grapher) 565
mv 247

N

NCPs (Network Control Protocols, протоколы управления сетью) 571
NetBSD 11
netcat 613
netcfg 251
NeTraMet 476
netstat 251
News-серверы 361
NFS (Network File System) 45, 425
nfs 70
NIC (Network Information Center) 46
nice 256, 464
NIS (Network Information Service) 410
NIS+ 411
nnrp.access 366
NNTP (Network News Transfer Protocol) 361
Node 46
nohup 267, 457
nslookup 251
NTP (Network Time Protocol) 401
  пакет 402
ntp.conf 404
ntpdate 407
ntpq 407
ntptrace 408
ntsysv 143, 148
NVT (Network Virtual Terminal) 492

O

OpenBSD 11
OpenSSH 497
openvt 267
OSI (Open System Interconnection) 46
OSPF (Open Shortest Path First) 53

P

PAM 157
passwd 146, 256
Pdq 613
perl 267
PGP (Pretty Good Privacy) 321

PID (Process Identification) 453
Pilot-Link 636
pine 251
ping 251
pipe 239
pnpdump 256
Portsentry 473, 489
PPP (Point-to-Point Protocol, протокол "точка-точка") 571
pppd 552, 578
PPTP (Point-To-Point Tunneling Protocol) 446, 450
Primary Domain Controller 429
printenv 267
Proc 69
procmail 251
proxy-сервер 382, 509
ps 261, 457
purp 189
pwd 247

Q

QNX 12
QuiteInsane 633
quota 261

R

r-команды (remote-команды) 496
raidtools 231
rc 142
rc.local 145
rc.sysinit 140
Rcp 496
rdev 121
Rdist 496
Red Hat 37
ReiserFS 69
renice 256, 464
reset 267
resizecons 267
Resource Records 300
Responsible Party 302
RFC (Request For Comments) 46
1128 403
1129 403
1165 403
1305 403
1489 273
2030 403
2131 305
2132 305
RIP (Routing Information Protocol) 46, 53
Rlogin 496
rm 247
rmdir 247
rmmod 257
Rootkit 480
rpm 174, 256
RPM 169
RPMS 170
RRDtool (Round Robin Database) 565
RSBAC 490
Rsh 496
Rsync 496
run level 135
runlevel 148

S

S/MIME (Secure/Multipurpose Internet Mail Extensions) 157, 320
Samba 429
SambaSentinel 444
SANE 631
saned 633
SATAN 473
scanadf 633
scanimage 633
scanlite 633
Scp 506
Secure Sockets Layer 157
Security-Enhanced Linux 490
setserial 231, 257
setterm 257
setup 148
SFM 274
Sftp 505
SGID 68
shadow 146
size 247
skill 257
Slackware 38
Slackware-LiveCD 40
slocate 248
Smb.conf 430
smbclient 444

smbpasswd 443
smbstatus 443
smbtar 444
SMTP (Simple Mail Transfer Protocol) 46
snice 257
Sniffer 159, 480
SNMP (Simple Network Management Protocol) 46
sort 264
split 248
Squid 383
Squid.conf 384
SRPMS 170
ssh 252
SSH 157, 497
ssh_config 500
Ssh-add 505
Ssh-agent 504
sshd_config 497
Ssh-keygen 504
Ssh-keyscan 506
Start Of Authority 300
startx 267
stat 248
sticky bit 68
StormLinux 37
strace 258
strings 267
strip 268
stty 258
Stunnel 471
su 268
subst 268
SUID 68
SuSE Linux 38
SWAT 444
Symbian 13
syslogd 477

T

tac 249
tail 249
tar 264
Tarballs 168
tc 565
TCP (Transmission Control Protocol) 46
telinit 140, 147
telnet 252
Telnet 46, 492
TkScan 633
tload 261
top 261, 461
Top Level Domains 293
traceroute 252
traffic shaper 565
Transparent proxy 564
tripwire 489
Tripwire 155
true 268
tune2fs 75, 77
tunneling 446
TurboLinux 38
tyx 231

U

UDP (User Datagram Protocol) 46
umount 71, 258
umsdos 69
Unicode 274
uniq 264
Usenet 361
useradd 258
users 262
uudecode 252

V

vdir 249
Vector Multicast Routing Protocol 54
VectorLinux 38
VFS 69
vi 264
vim 264
VirtualBox 657
vlock 152
VMWare 656
VPN (Virtual Private Networks) 157, 446
VueScan 634

W

w 263
Webmin 444
wget 252
whatis 235, 264

who 262
Win4Lin 657
Window Manager 605
Windows CE 10
Windows NT 9
Wine (Wine Is Not an Emulator) 655
WINE@Etersoft 656
wmtv 642
Wu-ftp 349

X

X Window 601
xawtv 640
xcam 633
XEN 658
xf86config 258
Xferlog 358
xia 69
xlock 152
xntpd 404, 408
xntpdc 408
xorg.conf 601
Xpdq 613
Xsane 632
xscanimage 633
xvidtune 258

Y

yes 268

Z

zic 258

Б

Брандмауэр 508
  с фильтрацией 510

В

Виртуальная частная сеть. *См.* VPN
Владельцы файлов 66

Д

Датаграмма 45
Демилитаризованная зона 534
Динамический режим 306
Дистрибутив 35
Доменная система имен. *См.* DNS
Домены верхнего уровня 293

Ж

Журналируемые файловые системы 78

З

Зомбированный процесс 459

И

Идентификационный номер
  процесса. *См.* PID
Индексные дескрипторы 76

К

Канал 65
Кодовая страница:
  8859-2 273
  8859-5 273
  Latin 0 272
  Latin 1 272
  КОИ8 273
Конвейер 239
Конференция 361
Кэш пакетов 194

М

Магический фильтр (magic-filter) 612
Микшер 659
Модификаторы прав доступа 67
Монтирование 70

О

Отложенный процесс 454

П

Пакет NTP. *См.* NTP
Передний план 453
Перенаправление стандартного ввода 239
Права доступа 66
Протокол ICP 383

Протокол PPP 571
Протокол динамического конфигурирования хостов. *См.* DHCP
Протокол передачи файлов. *См.* FTP
Процесс 453

Р

Расширенные атрибуты 469
Ручной режим 306

С

Сервер точного времени 401
Сетевая файловая система. *См.* NFS
Сетевой протокол времени. *См.* NTP
Сигнал SIGKILL 145
Служба сетевой информации. *См.* NIS
Сниффер 159
Сокет 66
Списки контроля доступа. *См.* ACL
Спящий процесс 459
Ссылки 65
Статический режим 306
Суперблок 75, 76

Т

Туннель 446

У

Уровень выполнения 135

Ф

Файл 65
Файл устройства 65
Файловая система 68
Фоновый режим 453

Х

Хоп 52

Э

Эхо-конференции 361