АЛЕКСЕЙ СТАРОВОЙТОВ

bhv®

НАСТРОЙКА АППАРАТНЫХ СРЕДСТВ в Linux

ОБЩИЕ ПРИНЦИПЫ РАБОТЫ LINUX С УСТРОЙСТВАМИ

НАСТРОЙКА ОСНОВНЫХ УСТРОЙСТВ: ВИДЕОКАРТЫ, ЖЕСТКОГО ДИСКА, ОПТИЧЕСКИХ ПРИВОДОВ

НАСТРОЙКА ПЕРИФЕРИЙНЫХ УСТРОЙСТВ: ПРИНТЕРА, СКАНЕРА, МОДЕМА

ОСНОВЫ ПРОТОКОЛА TCP/IP И НАСТРОЙКА СЕТИ

МЕТОДЫ ОПТИМИЗАЦИИ LINUX

ОСОБЫЕ ПРИЕМЫ ПРИ РАБОТЕ С LINUX

АППАРАТНЫЕ СРЕДСТВА

Алексей Старовойтов

НАСТРОЙКА АППАРАТНЫХ СРЕДСТВ В Linux

Санкт-Петербург

«БХВ-Петербург»

2006

УДК 681.3.06
ББК 32.973.26
С77

Старовойтов А. А.

С77 Настройка аппаратных средств в Linux. — СПб.: БХВ-Петербург, 2006. — 304 с.: ил.

ISBN 5-94157-839-3

Описывается установка и настройка аппаратных средств в операционной системе Linux: видеокарт, жестких дисков, оптических приводов и других основных устройств, а также принтеров, сканеров, модемов, flash-накопителей, сетевых карт и прочей периферии. При этом рассматриваются 2 варианта настройки — с использованием конфигурационных файлов и программ с графическим интерфейсом. Даются специальные приемы по работе с Linux: оптимизация операционной системы, восстановления после сбоя, диагностика устройств. Описание производится на примере дистрибутивов Mandrake и ASPLinux.

Для пользователей Linux

УДК 681.3.06
ББК 32.973.26

Группа подготовки издания:

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

Лицензия ИД № 02429 от 24.07.00. Подписано в печать 25.05.06.
Формат 70×100 1/16. Печать офсетная. Усл. печ. л. 24,51.
Тираж 2000 экз. Заказ №
"БХВ-Петербург", 194354, Санкт-Петербург, ул. Есенина, 5Б.

Санитарно-эпидемиологическое заключение на продукцию
№ 77.99.02.953.Д.006421.11.04 от 11.11.2004 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 5-94157-839-3

© Старовойтов А. А., 2006
© Оформление, издательство "БХВ-Петербург", 2006

Оглавление

Введение **1**

Глава 1. Как Linux работает с устройствами **5**

1.1. Ядро Linux 5
1.2. Драйверы устройств 7
1.3. Файлы устройств 13
1.4. Создание ядра и его компиляция 15
1.5. Здоровый консерватизм 21

Глава 2. Настройка видеокарты и монитора **23**

2.1. Видеосистема, эргономика и зрение 23
2.2. Два типа мониторов и принципы их работы 25
2.3. Видеокарта и принцип ее работы 28
2.4. Установка видеокарты и подключение монитора 30
2.5. Настройка графического режима 32
2.6. Установка драйверов от nvidia 36
2.7. Установка драйверов от ATI 39

Глава 3. Linux и жесткие диски **41**

3.1. Принципы работы жестких дисков 41
3.2. Конструкция жесткого диска 42
3.3. Некоторые параметры жестких дисков 45
3.4. RAID-массивы 47
3.5. Механическое подключение жесткого диска 49
3.6. Подключение нового жесткого диска в Linux 51
3.7. Утилита fsck 59
3.8. Утилита hdparm 61
3.9. Технология LVM 64

Глава 4. CD-ROM, DVD-ROM, CD-RW и их настройка в Linux 71

4.1. Основы технологии оптического хранения данных 71
4.2. Устройство и работа привода 72
4.3. Технологии CD-R и CD-RW 74
4.4. Накопители на DVD 76
4.5. Механическая установка привода CD (DVD) 77
4.6. Монтирование привода в ОС Linux 80
4.7. Программы для записи CD-RW-дисков 81

Глава 5. Настройка звука в Linux 93

5.1. Устройство звуковых карт 93
5.2. Механическая установка звуковой карты 99
5.3. Настройка звуковой карты в ОС Linux 102
5.4. Некоторые программы для работы со звуком в ОС Linux 108

Глава 6. Настройка клавиатуры и мыши 113

6.1. Общие сведения о клавиатуре 113
6.2. Настройка клавиатуры в ОС Linux 115
6.3. Мышь и ее конструкция 125
6.4. Настройка мыши в ОС Linux 127

Глава 7. Печать в Linux 135

7.1. Типы принтеров 135
7.2. Механическое подключение принтера 146
7.3. Настройка принтера в Linux 147

Глава 8. Сканеры и Linux 153

8.1. Немного истории 153
8.2. Принцип работы современных сканеров 154
8.3. Выбор сканера 155
8.4. Установка сканера в Linux 157
8.5. Сканирование изображения 159
8.6. Распознавание текста 166

Глава 9. Модемы 167

9.1. Общие сведения о модеме 167
9.2. Подключение модема 173
9.3. Установка модема в Linux и подключение к Интернету 173
9.4. Программа kppp 184

Глава 10. Сеть в Linux ... 191

10.1. Сети ... 191
10.2. Технология Ethernet и ее модификации ... 192
10.3. Физическое подключение компьютера к сети Fast Ethernet ... 201
10.4. Немного о TCP/IP ... 203
10.5. Настройка сети в Linux ... 210

Глава 11. Linux и flash ... 223

11.1. История создания цифрового фотоаппарата ... 224
11.2. Принцип действия цифрового фотоаппарата ... 224
11.3. Использование цифрового фотоаппарата в Linux ... 225

Глава 12. Некоторые вопросы ... 231

12.1. Программы для тестирования оборудования ... 231
12.2. Повышение производительности системы ... 239
12.3. Восстановление системы после сбоя ... 246

Глава 13. Установка ASP Linux ... 257

13.1. Перед установкой ... 257
13.2. Начало установки ... 258
13.3. Разбиение жесткого диска ... 260
13.4. Установка пакетов ... 265
13.5. Установка загрузчика ... 267
13.6. Настройка сети ... 269
13.7. Настройка X Window ... 270
13.8. Параметры клавиатуры и часового пояса ... 272
13.9. Установка пароля root ... 273
13.10. Перезагрузка ... 274

Заключение ... 277

Приложение ... 279

Предметный указатель ... 281

Введение

Операционная система Linux с каждым годом набирает обороты. Об этом свидетельствует многое: многочисленная армия пользователей (согласно Web-статистике популярность Linux возросла в несколько раз), распространение Linux-дистрибутивов почти в каждом компьютерном магазине, продажа компьютеров под управлением Linux, выпускаются даже обычные пакеты с изображением пингвина. Если, скажем, лет пять назад о Linux мало кто знал, то сейчас о ней знает каждый пользователь компьютера. Вполне может быть, что пользователь с Linux и не работал, но слышал о ней — это точно.

Рост популярности Linux не мог не отразиться на книгах, посвященных этой операционной системе. Да, книг по Linux стало больше, значительно больше. Но дело не только в количестве, но и в качестве. Раньше практически каждая книга по Linux представляла собою огромную "библию", в которой была описана практически вся операционная система — от ее установки до описания офисных программ. Постепенно, с ростом популярности Linux, произошло разделение книг на серверную и пользовательскую. Ведь не секрет, что Linux может выступать как в роли сервера, так и в роли рабочей станции. Это разделение произошло благодаря стараниям разработчиков Linux — появились более качественные и удобные графические интерфейсы, а также масса удобных графических программ. Если раньше Linux/UNIX представляли как "многозадачный DOS", то сегодня, благодаря интерфейсам GNOME и KDE, а также многочисленным пользовательским программам, представление о Linux в корне изменилось. Это был первый этап разделения Linux-литературы.

Вскоре после первого этапа начались появляться книги "второй волны". В отличие от первой, они более специализированы, например, взять эту книгу. В ней вы узнаете, как операционная система Linux взаимодействует с аппаратным обеспечением, как настроить ту или иную "железку" в Linux, а также ознакомитесь с основными принципами работы различных аппаратных

компонентов. Как видите, книга довольно узкой специализации. В ней нет ни слова о настройке сервера, ни о пользовательских программах. Поэтому можно сказать, что литература по Linux движется в правильном направлении.

В данной книге основной упор делается на дистрибутив ASPLinux v11 — последней версии ASP Linux на момент выхода книги. На данный момент это один из лучших дистрибутивов семейства Linux. Однако все сказанное в большей части справедливо для всех дистрибутивов семейства Red Hat, в том числе и для Fedora Core 4. Основные вопросы конфигурирования будут рассмотрены также и на примере дистрибутива Mandrake Linux v.10.

Материал каждой главы построен таким образом, что сначала рассматриваются общие теоретические основы функционирования устройства, затем рассказывается о его установке с механической точки зрения и, наконец, о его настройке под операционной системой Linux. Настройка операционной системы производится как путем редактирования конфигурационных файлов, так и при помощи графического конфигуратора.

В *главе 1* вы узнаете, как Linux работает с устройствами. Мы поговорим о ядре, файлах устройств, модулях устройств, то есть о драйверах. Также рассмотрим основные файлы конфигурации операционной системы.

В *главе 2* будет рассмотрена графическая подсистема Linux — система X Window. Вы узнаете, как правильно настроить монитор и видеоплату, а также как выполнить доводку изображения с помощью xvidtune. Основной упор будет сделан на видеокарты производства nVidia и ATI.

Все о настройке жесткого диска в Linux — вот так можно было назвать *главу 3*. В ней вы узнаете как принцип работы SCSI- и IDE-дисков, так и основы работы с файловой системой Linux. Кроме этого, в третьей главе будут рассмотрены такие практические вопросы, как подключение нового жесткого диска, создание на нем разделов и файловых систем. Не останется незатронутым процесс проверки файловой системы.

В *главе 4* мы поговорим об оптических приводах — CD/DVD: рассмотрим принципы их работы, настройку их в Linux, а также поговорим о записи дисков в Linux.

Настройка звуковой платы будет рассмотрена в *главе 5*, а в *главе 6* мы поговорим о настройке устройств ввода — клавиатуры и мыши.

Вопросам печати будет посвящена *глава 7*. В ней мы поговорим о настройке принтера в Linux, а также о типах и принципах работы принтеров.

Если мы рассмотрели принтер, то сканер не может остаться незамеченным. О нем мы поговорим в *главе 8*. Кроме настройки сканера в Linux, вы узнаете, кто изобрел первый сканер, также поговорим о принципах его работы.

Огромное количество пользователей для доступа в Интернет используют модемы, можно сказать, что модем является самым популярным "связным" в мире. Его настройке посвящена *глава 9*. А в *главе 10* мы поговорим о настройке сети в Linux. Вы также узнаете, как самостоятельно подключить свой компьютер к сети, не дожидаясь администраторов, как правильно обжать витую пару, как и куда подключить сетевой кабель.

Глава 11 посвящена вопросам использования flash-устройств, то есть устройств с flash-памятью, в Linux. Мы поговорим о подключении USB-дисков, цифровых фотоаппаратов, а также мобильных телефонов к Linux.

В *главе 12* вы узнаете, как восстановить систему после сбоя, как оптимизировать ее работу.

Глава 13 посвящена установке ASP Linux 11 и решению проблем, связанных с установкой. Во время установки сделаны качественные снимки экрана, поэтому вы без проблем установите ASP Linux, если будете следовать приведенным инструкциям.

В *Приложении* приведен список некоторых русскоязычных интернет-ресурсов, посвященных Linux. Ясно, что Linux-ресурсов достаточно много, поэтому были выделены наиболее интересные из них.

Вот теперь можете приступить к чтению этой книги. Помните, что эта книга не учебник и ее не нужно читать от корки до корки: если вам сейчас нужно настроить модем, читайте сразу девятую главу — вам не нужно читать первые восемь глав. Конечно, настоятельно рекомендую прочитать первую главу, чтобы узнать, как вообще Linux работает с устройствами.

Глава 1

Как Linux работает с устройствами

1.1. Ядро Linux

Ядро любой операционной системы представляет собой наиболее важную ее часть, относится это и к операционной системе Linux. В операционной системе ядро выполняет функции проводника между интерфейсом высокого уровня и аппаратными средствами персонального компьютера. То есть любое действие пользователя, будь то нажатие клавиши на клавиатуре или движение мышью, вначале обрабатывается ядром операционной системы, а уже затем поступает на более высокий уровень. На низком уровне ядро операционной системы выполняет такие функции:

- организация многозадачности, то есть распределение временных ресурсов процессора, и адресного пространства для различных задач;
- организация файловой системы;
- организация виртуальной памяти;
- низкоуровневое взаимодействие с устройствами.

Схему взаимодействия аппаратного обеспечения, ядра операционной системы и оболочки высокого уровня можно представить в следующем виде (рис. 1.1). Перед нами многослойный "колобок". В центре этого колобка располагается аппаратная часть. Доступ к любой аппаратной части возможен только через ядро операционной системы, это центральный слой колобка. Внешнее кольцо — это пользовательское пространство. В пользовательском пространстве работают все остальные процессы, запущенные в системе.

Ядро Linux, как и сама операционная система, находится в постоянном развитии. В ядро постоянно добавляются новые функции, изменяется алгоритм исполнения старых функций. Естественно, что такие изменения могут нести в себе ошибки и, как следствие, нестабильность в работе. По мере тестирования недостатки и недочеты, приводящие к нестабильности, устраняются,

и ядро становится стабильным. Естественно, что использование нестабильного ядра в определенных случаях не желательно. Для того чтобы отличить "стабильную" версию ядра от "нестабильной", была введена специальная система нумерации. Нумерация ядер состоит из трех групп цифр, разделенных точкой. Первая цифра номера ядра представляет собой старший номер ядра. На сегодняшний день это версия 2. Вторая цифра, или младший номер ядра, говорит о том, является ядро стабильным или нет. Если вторая цифра четная: 0, 2, 4, 6, 8 — то ядро является стабильным. Если вторая цифра нечетная: 1, 3, 5, 7, 9 — то это нестабильное ядро, то есть ядро, находящееся в стадии разработки. Третья цифра является номером редакции данного "стабильного" или "нестабильного" ядра (рис. 1.2).

Рис. 1.1. Взаимодействие аппаратной части, ядра и процессов в Linux

Рис. 1.2. Нумерация ядра

В мире аппаратного обеспечения существует большое количество периферийных устройств, иногда попадаются и экзотические. Большим преимуществом Linux является открытый код ядра, что позволяет пользователям, обладающим определенной подготовкой, писать свои драйверы для этих устройств. Конечно, сегодня копаться в исходном коде ядра и писать драйверы среднестатистическому пользователю практически не приходится, но настраивать ядро и добавлять уже существующие модули должен уметь каждый уважающий себя администратор.

Зачем конфигурировать ядро? — спросите вы. Дело в том, что изначально при установке операционной системы создается конфигурация ядра, рассчитанная на выполнение широкого круга задач, в различной аппаратной среде. В созданную при установке конфигурацию попадают различные драйверы устройств, которые вы, возможно, никогда и не будете использовать, значения многих параметров поставлены в "среднее" положение. В то же время, некоторые необходимые драйверы могут отсутствовать. Чтобы добавить поддержку необходимого драйвера, недостаточно просто скопировать файлы, необходимо интегрировать код драйвера и ядро. В конце концов, не оптимально построенное ядро сказывается на производительности системы.

1.2. Драйверы устройств

Драйвер — это программа, которая организует взаимодействие "высшей" программной части операционной системы с аппаратной частью компьютера. Являясь частью ядра, драйвер выполняет функции переводчика между аппаратной частью и остальной системой. Необходимость наличия драйверов продиктована большим разнообразием аппаратной части. С помощью драйверов ядро делается как бы независимым от различного вида аппаратного обеспечения.

Поэтому, когда вы, например, купили новый принтер, вам не надо искать новую версию операционной системы, достаточно подключить к ядру новый драйвер, который идет в комплекте с вашим устройством. Драйверы должны быть для конкретной версии ядра. Недопустимо использование драйверов для других операционных систем.

Драйверы для устройств могут поставляться в двух формах:

- загружаемый модуль;
- заплатка к ядру определенной версии.

1.2.1. Загружаемый модуль

Загружаемый модуль является очень удобным способом подключения или отключения драйверов. (Помимо драйверов при помощи модульной структуры можно подключать и отключать и другие компоненты ядра.) При использовании подключаемых модулей нет необходимости менять код ядра ОС. Благодаря использованию подключаемых модулей уменьшается размер ядра. Подключать и отключать драйвер можно "на лету", то есть без перезагрузки операционной системы.

Как правило, модули хранятся в отдельном каталоге /lib/modules/x.x.x, где x.x.x представляет собой версию ядра, для которой скомпилированы данные модули. В каждом каталоге модули систематизированы по группам, о содержании каталогов можно судить по их наименованию.

О том, какие модули загружены в данный момент, можно узнать, выполнив команду `lsmod` (рис. 1.3).

```
[root@localhost ~]# lsmod
Module                  Size  Used by
md5                     4161  1
ipv6                  237569  10
parport_pc             24577  1
lp                     11565  0
parport                42249  2 parport_pc,lp
autofs4                24261  0
sunrpc                161701  1
ds                     17221  0
yenta_socket           18753  0
pcmcia_core            60937  2 ds,yenta_socket
dm_mod                 55253  0
button                  6609  0
battery                 8645  0
ac                      4933  0
uhci_hcd               31577  0
snd_ens1371            27685  2
```

Рис. 1.3. Выполнение команды **lsmod**

Поскольку подключаемый модуль представляет собой часть ядра, подключение любого нового непроверенного модуля может вызвать нарушение нормальной работы системы, поэтому вначале рекомендуется проверить работу модуля в режиме ручной загрузки, затем уже проверенный модуль перевести в режим автоматической загрузки.

Ручная загрузка

Как правило, она используется при отладке модулей и загрузке редко используемых модулей. Для ручной загрузки и выгрузке модулей используется несколько команд. Одну из них, `lsmod`, позволяющую посмотреть загруженные модули, мы уже рассмотрели. Рассмотрим другие команды.

Команда `insmod` позволяет загрузить модуль в работающее ядро. При загрузке модуля не учитываются существующие зависимости. Команда использует следующий синтаксис:

- `#insmod /lib/modeles/2.4.19/samples.o` — загрузка модуля samples.o без передачи ему параметров;
- `#insmod /lib/modeles/2.4.19/samples.o i=12 o=20` — загрузка модуля samples.o с передачей ему параметров.

Команда `rmmod` используется для удаления загруженных модулей. Команда будет выполнена только в том случае, если число ссылок на модуль в данном случае равно нулю. На рис. 1.4 можно видеть, что выгрузить модуль `raid0` невозможно, так как он используется одним процессом, это можно видеть по фрагменту команды `lsmod`. А выгрузить модуль `lp` оказалось возможным, так как на него нет ссылок.

```
raid0                   7745  1
BusLogic               72157  0
sd_mod                 16449  0
scsi_mod              119825  2 BusLogic,sd_mod
[root@localhost ~]# rmmod raid0
ERROR: Module raid0 is in use
[root@localhost ~]# rmmod lp
[root@localhost ~]#
```

Рис. 1.4. Использование команды **rmmod**

Команда `modprobe` осуществляет проверку зависимостей модулей, просматривая файл /lib/modules/x.x.x/modules.dep, затем с использованием команды `insmod` осуществляет загрузку необходимых модулей.

Файл modules.dep имеет следующий формат (листинг 1.1).

Листинг 1.1. Файл modules.dep

```
/lib/modules/2.6.9-1.667asp/kernel/sound/soundcore.ko:
/lib/modules/2.6.9-1.667asp/kernel/sound/usb/snd-usb-lib.ko:
/lib/modules/2.6.9-1.667asp/kernel/sound/core/snd-rawmidi.ko
```

```
/lib/modules/2.6.9-1.667asp/kernel/sound/core/seq/snd-seq-device.ko
/lib/modules/2.6.9-1.667asp/kernel/sound/core/snd.ko /lib/modules/2.6.9-
1.667asp/kernel/sound/soundcore.ko
```

В приведенном примере модуль

```
/lib/modules/2.6.9-1.667asp/kernel/sound/soundcore.ko:
```

не зависит от других модулей. А модуль

```
/lib/modules/2.6.9-1.667asp/kernel/sound/usb/snd-usb-lib.ko
```

зависит от четырех модулей. Реально в файле каждая строка относится к отдельному модулю. В нашем случае строка получилась длинной, поэтому имеет место перенос строк. `modprobe` имеет следующий синтаксис:

```
modprobe [-adnqv] [-C config] module [symbol=value ...]
modprobe [-adnqv] [-C config] [-t type] pattern
modprobe -l [-C config] [-t type] pattern
modprobe -c [-C config]
modprobe -r [-dnv] [-C config] [module ...]
modprobe -Vh
```

Ключи команды могут быть следующими:

- `-C` — позволяет использовать альтернативный файл конфигурации вместо /etc/modules.conf;
- `-v` — позволяет получить подробный отчет о процессах, происходящих во время загрузки модулей;
- `-c` — просмотр текущей конфигурации модулей.

Автоматическая загрузка

Для проверенных модулей, используемых в работе системы, было бы нецелесообразным использовать ручную загрузку. Поэтому основным методом загрузки модулей остается автоматическая загрузка модулей. В большинстве дистрибутивов для загрузки модулей используется файл /etc/modules.conf. Однако в некоторых версиях может использоваться файл /etc/conf.modules, в некоторых клонах Red Hat может использоваться файл /etc/rc.d/rc.modules. Несмотря на такое разнообразие в файлах, в конечном итоге все дистрибутивы должны перейти на использование файла /etc/modules.conf. Сгенерировать файл /etc/modules.conf можно, выполнив команду

```
modprobe -c.
```

При таком синтаксисе сгенерированный файл выводится на экран, а чтобы его сохранить, можно ввести команду

```
modprobe —c >> /etc/ver1.conf,
```

в этом случае результаты выполнения команды перенаправятся в файл /etc/ver1.conf. Попробуем выполнить данную операцию для ASPLinux, в частности, этот файл может выглядеть примерно так, как показано на листинге 1.2.

Листинг 1.2. Динамически созданный файл /etc/modules.conf

```
# Generated by modprobe -c (2.4.18)
path[boot]=/lib/modules/boot
path[toplevel]=/lib/modules/2.4.18-19.7se
path[toplevel]=/lib/modules/2.4
path[kernel]=/lib/modules/kernel
path[fs]=/lib/modules/fs
path[net]=/lib/modules/net
path[scsi]=/lib/modules/scsi
path[block]=/lib/modules/block
path[cdrom]=/lib/modules/cdrom
path[ipv4]=/lib/modules/ipv4
. . . . .
# Prune
prune modules.dep
prune modules.generic_string
prune modules.pcimap
prune modules.isapnpmap
prune modules.usbmap
prune modules.parportmap
prune modules.ieee1394map
. . . . .
# Below
below r128 agpgart
below radeon agpgart
below mga agpgart
below i810 agpgart
# Aliases
. . . . .
alias block-major-1 rd
alias block-major-2 floppy
alias block-major-3 ide-probe-mod
alias block-major-7 loop
alias block-major-8 sd_mod
```

```
alias block-major-9 md
alias block-major-11 sr_mod
alias block-major-13 xd
. . . .
# Options
options dummy0 -o dummy0
options dummy1 -o dummy1
options sb io=0x220 irq=7 dma=1 dma16=5 mpu_io=0x330
# Commands
post-install binfmt_misc /bin/mount -t binfmt_misc none
/proc/sys/fs/binfmt_misc > /dev/null 2>&1 || :
pre-remove binfmt_misc /bin/umount /proc/sys/fs/binfmt_misc > /dev/null
2>&1 || :
post-install sound-slot-0 /bin/aumix-minimal -f /etc/.aumixrc -L
pre-remove sound-slot-0 /bin/aumix-minimal -f /etc/.aumixrc -S
# Miscellaneous file and directory names
generic_stringfile=/lib/modules/2.4.18-19.7se/modules.generic_string
pcimapfile=/lib/modules/2.4.18-19.7se/modules.pcimap
isapnpmapfile=/lib/modules/2.4.18-19.7se/modules.isapnpmap
usbmapfile=/lib/modules/2.4.18-19.7se/modules.usbmap
parportmapfile=/lib/modules/2.4.18-19.7se/modules.parportmap
ieee1394mapfile=/lib/modules/2.4.18-19.7se/modules.ieee1394map
pnpbiosmapfile=/lib/modules/2.4.18-19.7se/modules.pnpbiosmap
depfile=/lib/modules/2.4.18-19.7se/modules.dep
persistdir=/var/lib/modules/persist
```

Файл разбит на несколько секций, наиболее интересны секции `path`, `alias`, `options`.

В данном случае секция `path` указывает то, где находится конкретный модуль. При необходимости данную запись можно откорректировать, например, в случае если вы используете нестандартный модуль.

Инструкция `alias` обеспечивает привязку номеров устройств к соответствующим модулям.

Инструкция `options` позволяет передать модулю параметры при загрузке. В нашем случае строка:

```
options sb io=0x220 irq=7 dma=1 dma16=5 mpu_io=0x330
```

передает адреса и прерывания звуковой карте.

Кроме этого в секции `#Commands` могут указываться также команды `pre-install`, `post-install`, `pre-remove`, `post-remove`, `install`, `remove`.

При помощи этих строк указываются команды, выполняемые в тот момент, когда модуль подключается к ядру. Момент выполнения команды определяется соответственно в моменты:

- `pre-install` — перед подключением;
- `post-install` — после подключения;
- `pre-remove` — перед удалением;
- `post-remove` — после удаления;
- `install` — во время подключения;
- `remove` — во время удаления.

1.2.2. Заплатка к ядру

Заплатка может быть поставлена в качестве инсталляционного сценария либо в качестве файла. Заплатка устанавливается только на ядро конкретной версии, поэтому, чтобы избежать ошибок, надо быть внимательным к номеру ядра.

Переходим в каталог, где хранятся исходные файлы ядра:

```
cd /usr/src/linux/
```

Затем для того, чтобы проверить корректность заплатки, необходимо воспользоваться следующей командой:

```
patch –p1 –dry-run < имя_файла_заплатки
```

Если эта команда прошла без сбоев и не появлялось строк типа `FILED`, значит, заплатку можно устанавливать:

```
patch –p1 < имя_файла_заплатки
```

1.3. Файлы устройств

С точки зрения пользователя (процесса), все устройства в Linux представляются файлами: хотите получить какую-то информацию из устройства — считываете файл, хотите вывести — наоборот, записываете в файл. Представление устройства в виде файлов облегчает распределение прав доступа к тем или иным аппаратным ресурсам.

По принятой договоренности файлы устройств хранятся в каталоге /dev. В этом каталоге при необходимости они могут группироваться по типам.

С каждым из файлов устройств связаны номера старший и младший. Старший номер определяет тип устройства, младший указывает номер устройства в группе типов. По старшему номеру определяется тип драйвера, который используется для этого устройства.

Большинство устройств являются блочными (хранение данных — дисковод флоппи-дисков, жесткий диск) или символьными (передача и прием данных — модем, мыши, порты).

Некоторые распространенные типы устройств приведены в табл. 1.1.

Таблица 1.1. *Некоторые распространенные символьные и блочные устройства*

Файл	Описание устройства
/dev/fd0	Дисковод флоппи-дисков
/dev/hda	Жесткий диск (IDE) — master на первичном канале
/dev/hdb	Жесткий диск (IDE) — slave на первичном канале
/dev/hdc	Жесткий диск (IDE) — master на втором канале
/dev/hdd	Жесткий диск (IDE) — slave на втором канале
/dev/sda	Первый диск SCSI
/dev/sdb	Второй диск SCSI
/dev/ttys0	Первый последовательный порт (com1)
/dev/ttys1	Второй последовательный порт (com2)
/dev/md0	RAID-массив номер 0
/dev/eth0	Сетевой интерфейс номер 0

Как правило, после установки системы все устройства уже созданы, однако устройство можно создать принудительно, воспользовавшись командой `mknod`. Синтаксис команды таков:

```
mknod [ПАРАМЕТР]... ИМЯ ТИП_устройства [Старший Младший]
```

В качестве параметров могут быть следующие:

- `-m`, `--mode=MODE` — устанавливает атрибуты доступа (как для `chmod`), а не стандартные `=rw`;
- `--help` — выдает эту информацию и заканчивает работу;
- `--version` — выдает информацию о версии и заканчивает работу.

Тип устройства может принимать значение:

- `b` — создать блоковый (буферизованный) файл;
- `c`, `u` — создать символьный (небуферизованный) файл;
- `p` — создать файл FIFO.

Старшие и младшие номера устройства узнать можно, введя команду:

```
ls –l ИМЯ_устройства
```

Например, для дисковода флоппи-дисков это будет выглядеть так:

```
[root@localhost ~]# ls -l /dev/fd0
brw-rw----  1 root floppy 2, 0 Авг 22  2005 /dev/fd0
```

Рис. 1.5. Старший и младший номера дисковода флоппи-дисков

Как видно из рис. 1.5, старший байт равен 2, младший равен 0, соответственно, создание такого устройства будет выполнено командой:

```
# mknod /dev/fd0 b 2 0
```

Вообще надо сказать, что, как правило, создавать устройства вручную приходится довольно редко.

1.4. Создание ядра и его компиляция

Конфигурирование ядра сводится к конфигурированию файла .config в каталоге с исходными кодами ядра. Как правило, это каталог /usr/scr/linux, но для этого у вас должны быть установлены исходные файлы ядра.

Редактировать файл .config можно тремя способами:

- `make xconfig` — этот метод предпочтителен, так как использует графический интерфейс. Соответственно, этот метод может быть использован только в том случае, если в системе инсталлирована оболочка X Window (рис. 1.6);
- `make menuconfig` — тоже достаточно наглядная среда редактирования файла .config, однако использовать ее лучше только в том случае, если нет графического интерфейса (рис. 1.7);
- в самую последнюю очередь стоит использовать утилиту config. При редактировании файла .config она наименее наглядна и удобна. В процессе своей работы эта утилита выдает запрос на изменение каждого параметра и не позволяет перемещаться по диалогу вверх и вниз (рис. 1.8).

Для того чтобы ваша команда была выполнена, необходимо перейти в каталог, где располагаются исходные коды ядра. Это, как правило, /usr/src/linux. Для того чтобы перейти в этот каталог, необходимо выполнить команду

```
cd /usr/src/linux,
```

а затем выполнить одну из трех описанных уже команд.

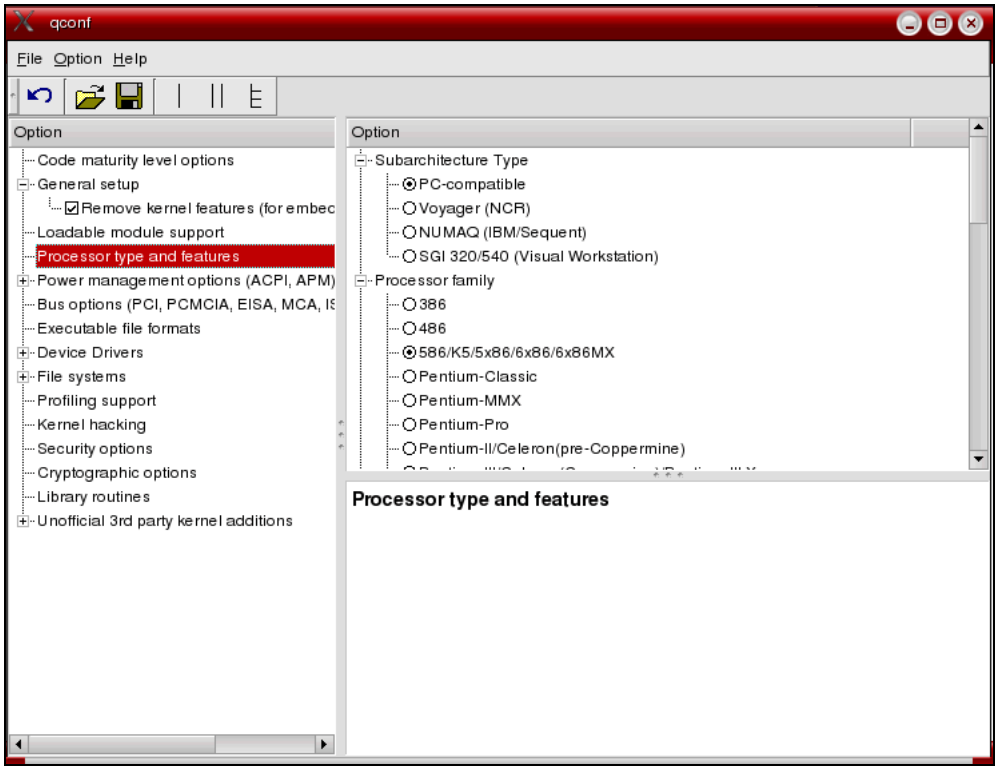

Рис. 1.6. Утилита xconfig

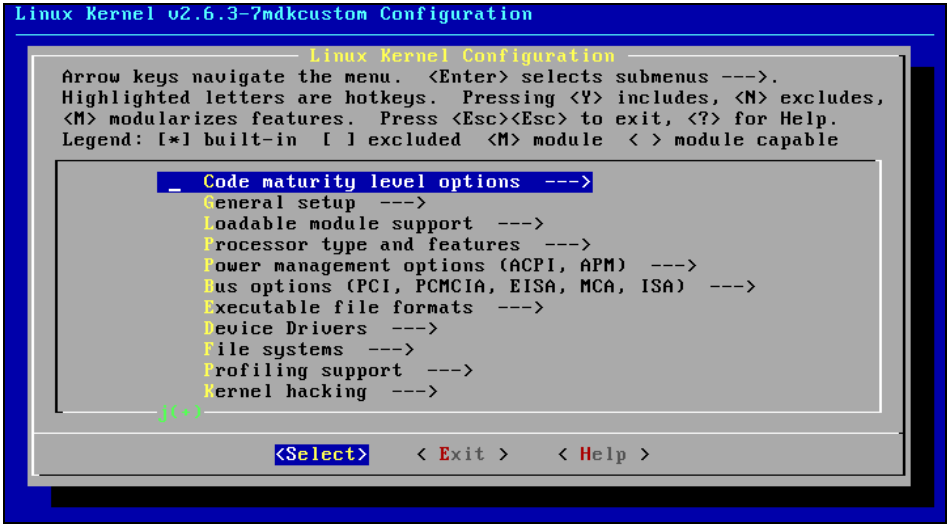

Рис. 1.7. Утилита menuconfig

```
#
# using defaults found in .config
#
*
* Linux Kernel Configuration
*
*
* Code maturity level options
*
Prompt for development and/or incomplete code/drivers (EXPERIMENTAL) [Y/n/?] y
  Select only drivers expected to compile cleanly (CLEAN_COMPILE) [Y/n/?] _
```

Рис. 1.8. Работа утилиты config

Во время конфигурирования ядра пользователь выбирает опции, которые в дальнейшем намерен использовать в ядре.

Если какой-то из параметров вам не знаком, вы можете использовать установки по умолчанию.

Итак, закончив работу с одной из описанных ранее программ, мы создали файл .config, но для компиляции ядра этого недостаточно, нужно выполнить еще несколько операций:

- `make dep` — при этом создаются файлы зависимостей;
- `make clean` — удаление промежуточных файлов от предыдущих операций;
- `make bzImage` — собственно создание ядра, при этом вновь созданное ядро будет располагаться в каталоге /usr/src/linux/arch/i386/boot в файле под именем bzImage;
- `make modules` — эта команда выполняет сборку модулей, соответствующих ядру, созданному командой раньше;
- `make modules_install` — перемещает созданные модули из исходного дерева ядра в каталог /lib/modules.

Теперь необходимо переместить два файла из каталога /usr/src/linux/arch/i386/boot:

- скопировать файл /usr/src/linux/arch/i386/boot/bzImage в файл /boot/bzImage_new;
- скопировать файл /usr/src/linux/arch/i386/boot/System.map в файл /boot/System.map.

Необходимо также отредактировать загрузчик, чтобы он знал, где брать новое ядро.

Поскольку загрузчик LILO, пожалуй, является основным загрузчиком для клонов Red Hat, то начнем с него. Конфигурация загрузчика LILO находится

в файле /etc/lilo.conf, в нашем случае он может выглядеть примерно как в листинге 1.3.

Листинг 1.3 Файл конфигурации загрузчика /etc/lilo.conf

```
boot=/dev/sda
map=/boot/map
default="linux-smp"
keytable=/boot/ru4.klt
prompt
nowarn
timeout=100
message=/boot/message
menu-scheme=wb:bw:wb:bw
image=/boot/vmlinuz
       label="linux"
       root=/dev/sda1
       initrd=/boot/initrd.img
       append="devfs=mount acpi=ht resume=/dev/sda5 splash=silent"
       vga=788
       read-only
image=/boot/vmlinuz
       label="linux-nonfb"
       root=/dev/sda1
       initrd=/boot/initrd.img
       append="devfs=mount acpi=ht resume=/dev/sda5"
       read-only
image=/boot/vmlinuz-2.6.3-7mdk
       label="263-7"
       root=/dev/sda1
       initrd=/boot/initrd-2.6.3-7mdk.img
       append="devfs=mount acpi=ht resume=/dev/sda5 splash=silent"
       read-only
image=/boot/vmlinuz-smp
       label="linux-smp"
       root=/dev/sda1
       initrd=/boot/initrd-smp.img
       append="devfs=mount acpi=ht resume=/dev/sda5 splash=silent"
       read-only
```

```
image=/boot/vmlinuz
       label="failsafe"
       root=/dev/sda1
       initrd=/boot/initrd.img
       append="failsafe acpi=ht resume=/dev/sda5 devfs=nomount"
       read-only
other=/dev/fd0
       label="floppy"
       unsafe
```

Для того чтобы добавить в список выбираемых при загрузке наше ядро, необходимо добавить в конец этого файла следующий фрагмент (листинг 1.4).

Листинг 1.4. Дополнительный фрагмент

```
image=/boot/bzImage_new
    # Ссылка на ядро
       label="linux-new"
    # Метка
       root=/dev/sda1
    # Расположение корневой файловой системы
       append="devfs=mount acpi=ht resume=/dev/sda5 splash=silent"
    # Через append осуществляется передача дополнительных параметров
       read-only
    # Режим монтирования корневой файловой системы
```

Теперь выполните команду

`/sbin/lilo.`

После этих манипуляций ваше новое ядро появится в конце списка выбираемых вариантов загрузки.

В ASPLinux для использования рекомендован ASPLoader. Изменить конфигурацию вариантов загрузки у ASPLoader можно, отредактировав файл /etc/aspldr.conf. По умолчанию этот файл выглядит как в листинге 1.5.

Листинг 1.5. Файл /etc/aspldr.conf

```
[asplinux1@ASPLinux(2.6.9-1.667asp)]
icon linux
kernel /boot/vmlinuz-2.6.9-1.667asp root=/dev/md0 ro rhgb
```

```
initrd /boot/initrd-2.6.9-1.667asp.img

[SEPARATOR]

[floppy@Boot from floppy]
icon floppy
sysboot a:

[BOOTMGR]
video graphics
default asplinux1
timeout 15
clock 24

[ACTIVATOR]
writembr on
writeboot off
biosnum 1
mbrdev /dev/hda
language en
```

Для того чтобы добавить ссылку на новое ядро, необходимо перед секцией `[separator]` вставить строки примерно следующего содержания (листинг 1.6).

Листинг 1.6. Фрагмент вставки

```
[New_Kernel@Menu_New_Kernel]
# New_Kernel — метка
# Menu_New_Kernel — название пункта меню
kernel /boot/bzImage_new root=/dev/md0 ro
# root=/dev/md0 говорит о том, что корневая файловая система
# располагается
# на Рейд-массиве
```

После внесения изменений необходимо выполнить команду

`/sbin/aspldr`,

теперь после перезагрузки ваше ядро появится в списке загружаемых модулей.

1.5. Здоровый консерватизм

Обновления для ядер выходят достаточно часто, и в принципе установка нового ядра или заплаток на старое ядро не вызывает особых сложностей. К вопросу постоянного обновления системы лучше подходить с позиций здорового консерватизма.

Во-первых, установка нового ядра, помимо плюсов, может иметь и отрицательные последствия. Никто не может вам гарантировать, что новое ядрышко будет столь же надежно работать, как и старое, проверенное временем. Конечно, стабильные версии ядер проверены временем и пользователем, но каждый случай индивидуален.

Во-вторых, подготовка нового ядра и его последующая установка занимают некоторое время, а его, как правило, всегда не хватает.

Поэтому, касаясь вопроса обновления, нужно подходить взвешенно и обдуманно, заплатки и обновления лучше группировать и выполнять несколько обновлений за раз. Допустим, ввести правило устанавливать обновления один раз в месяц, специально планируя для этого свое время.

Глава 2

Настройка видеокарты и монитора

2.1. Видеосистема, эргономика и зрение

Обработанную компьютером информацию необходимо выдать человеку. Естественно, что наиболее удобный и продуктивный путь — визуальный.

Преобразованием двоичных нулей и единиц в понятные образы занимается видеосистема компьютера. Видеокарта (видеоадаптер) преобразовывает информацию от процессора в видеосигнал для монитора. Причем на видеокарту перекладывается часть операций по расчету картинки (сцены), что разгружает центральный процессор и повышает производительность системы в целом.

Монитор занимается непосредственно преобразованием видеосигнала, поступившего от видеокарты, в изображение, которое воспринимает глаз человека. В этом процессе восприятия есть некоторые особенности.

Абсолютное большинство объектов материального мира, которые нас окружают (деревья, трава, животные и т. д.), светят отраженным светом. Поэтому в процессе эволюции так сложилось, что наши глаза привыкли к восприятию отраженного света.

Изображение, формируемое дисплеем, светится активным светом. Кроме того, формирование изображения в мониторах на электронно-лучевых трубках (коих у пользователей пока еще большинство) формируется при помощи развертки (см. далее): точки люминофора на мониторе светятся не постоянным светом, а периодически зажигаются и гаснут. Цвета монитора также отличаются от цветов реального мира, они более яркие и насыщенные.

Процесс работы за компьютером — это дополнительная нагрузка для глаз еще и по той причине, что человеку приходится постоянно всматриваться в изображение на мониторе (текст, графика). Это вызывает утомление глаз.

Для комфортной работы необходимо, чтобы излучение монитора было низким. Низкое электромагнитное излучение — это один из главных параметров,

влияющих на зрение. В этой области стандарты появились достаточно давно, а обновляются они в сторону ужесточения почти каждый год. Первым стандартом был MPR I (1987 г.), он ограничивал мощность электростатических, электрических и магнитных полей для компьютерной и офисной техники. Был разработан Национальным департаментом стандартов Швеции (SWEDAC) совместно с Институтом расщепляющихся материалов (SSI). Затем появился стандарт MPR II (1990 г.).

Требования MPR II учитываются при разработке комплексных стандартов TCO, которые действуют в настоящий момент.

Дальнейшие изменения стандартов шли в основном по направлению совершенствования энергосбережения, эргономики, охраны окружающей среды и экологии, пожарной и электрической безопасности. Конечно, при прочих равных условиях стоит обратить внимание на монитор TCO 2003.

Частота обновления экрана должна быть не ниже 85 Гц, а лучше —100 Гц. Этот параметр не так прост, как может показаться. Увеличение частоты обновления требует дополнительных характеристик и от монитора, и от видеокарты. Видеоусилители, находящиеся в мониторе, должны обладать определенной пропускной способностью по частоте. А чем больше пропускная способность видеоусилителя по частоте, тем он дороже, соответственно, и монитор тоже дороже. Кроме того, видеокарта также должна поддерживать эту пропускную способность по частоте.

Пропускную частоту можно вычислить по формуле:

$$W = H \times V \times F .$$

Здесь H — максимальное разрешение по горизонтали, V — максимальное разрешение по вертикали, F — частота кадров.

Действие этой формулы можно заметить в следующем: когда вы поднимаете разрешение, у вас автоматически снижается предельная верхняя частота кадровой развертки, которую можно выбрать.

Если принудительно превысить эти параметры, сработает система защиты, и монитор будет показывать черный экран. Сделано это для защиты от выхода из строя электрической части монитора.

Итак, разрешение экрана и максимальная допустимая частота взаимосвязаны, поэтому лучше задать разрешение поменьше, а частоту сделать 100 Гц.

Выбирайте разрешение так, чтобы можно было работать корректно. Для 17-дюймового монитора это 1024×768. Если вы работаете с чертежами или в какой-нибудь конструкторской программе, то вам понадобится большее разрешение.

Важно правильное расположение монитора относительно источников света. Ясно, что наихудшим вариантом является размещение монитора напротив окна. Монитор надо разместить так, чтобы прямые лучи солнечного света на него не попадали, в противном случае ни о какой работе не может быть и речи.

2.2. Два типа мониторов и принципы их работы

В настоящий момент в продаже присутствуют два типа мониторов. Это мониторы с электронно-лучевыми трубками и жидкокристаллические мониторы. Первые наиболее распространены у потребителей, которые уже купили компьютерную технику, зато вторые уже вытеснили их с прилавков магазинов. Произошло это благодаря существенному снижению цен на жидкокристаллические мониторы. Рассмотрим принципы работы обоих типов.

2.2.1. Мониторы с электронно-лучевыми трубками

Как мы уже говорили, они на сегодняшний день самые распространенные. Их еще называют CRT-мониторы (Cathode Ray Tube, катодно-лучевая трубка). Схема монитора приведена на рис. 2.1.

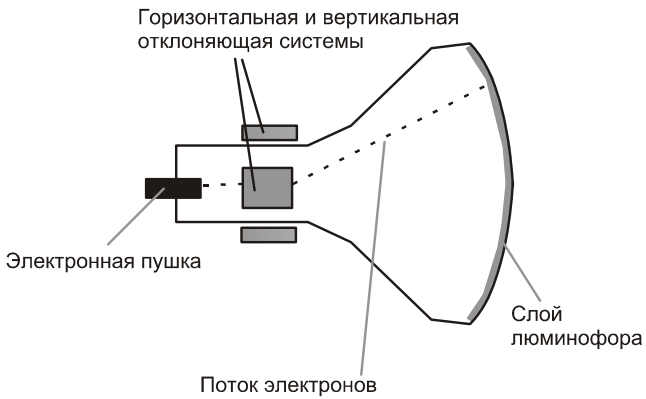

Рис. 2.1. Схема CRT-монитора

Электронно-лучевая трубка представляет собой стеклянную колбу, из которой откачан воздух. С фронтальной стороны внутренняя часть покрыта люминофором. Люминофор — это вещество, испускающее свет при бомбардировке заряженными частицами.

Для формирования потока заряженных частиц используется электронно-лучевая пушка, которая испускает поток электронов. Этот поток электронов разгоняется электромагнитным полем, затем проходит через отклоняющую систему, так, чтобы он попал в нужное место экрана.

Строго говоря, пушек три: для красного (Red), зеленого (Green) и голубого (Blue) цветов. Различная комбинация этих трех основных цветов и дает всю цветовую гамму монитора.

И, пройдя через металлическую маску или решетку, поток электронов попадает на экран, вызывая свечение люминофора.

Отклоняющая система, которая управляет потоком электронов, состоит из двух подсистем. Одна отклоняет луч в вертикальном направлении, другая в горизонтальном.

Луч перемещается по поверхности трубки в определенном порядке (рис. 2.2).

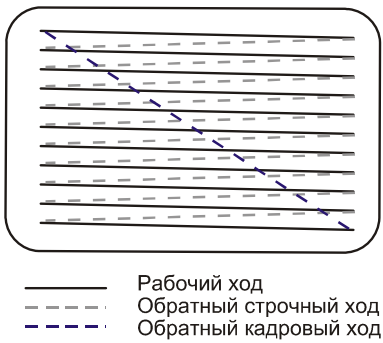

Рис. 2.2. Порядок обхода луча в электронно-лучевой трубке

Сплошными линиями показан рабочий ход луча, пунктирными —возвращение луча на рабочую позицию. Синим цветом обозначен обратный кадровый ход (возвращение в исходную позицию после сформирования кадра).

2.2.2. Жидкокристаллические мониторы

Экраны жидкокристаллических мониторов (LCD, Liquid Crystal Display) сделаны из вещества, которое, находясь в жидком состоянии, обладает свойствами ориентации молекул. Работа таких мониторов основана на поляризации светового потока при прохождении через слой рабочего вещества. Кристаллы вещества пропускают ту часть светового потока, вектор магнитной индукции которой лежит в плоскости, параллельной оптической плоскости поляроида.

Жидкокристаллические мониторы работают либо в отраженном свете, либо в проходящем свете.

Рабочее поле монитора состоит из большого числа сегментов, называемых пикселами. Каждый из пикселов поворачивает вектор светового потока в нужном направлении. Однако поляризация света не дает ни цветовой раскраски, ни регулировки силы света. То есть, несмотря на то, что свет и стал поляризованным, он так и остался немодулированным по яркости. Поэтому добавлены еще два слоя, представляющие поляризационные фильтры. В результате прохождения каждого из поляризаторов пучок света ослабевает, в зависимости от разности угла между поляризацией светового потока и фильтра (рис. 2.3).

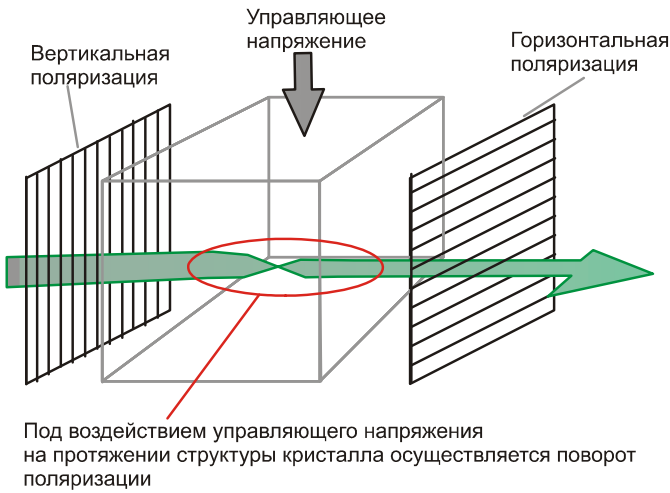

Рис. 2.3. Ячейка жидкокристаллического монитора

Описанная технология представляет собой единичную ячейку жидкокристаллической матрицы. Развитие технологии позволило миниатюризировать эту единичную ячейку, чтоб формировать изображения с достаточной степенью четкости.

Формирование цветного изображения основано на фильтрации света, отраженного от задней подложки тремя светофильтрами (красным, зеленым, голубым). Комбинация этих трех основных цветов, как и в случае с CRT-мониторами, дает необходимый оттенок.

2.3. Видеокарта и принцип ее работы

Как вы знаете, внутри компьютера сигналы представлены в двоичной форме, то есть в форме единиц и нулей. Именно в такой форме они обрабатываются процессором и передаются по внутренней шине материнской платы.

Сигналы в двоичной форме в виде команд и данных поступают на видеокарту, где из них формируется видеосигнал, который впоследствии и формирует изображение.

С ростом объемов графики и сложности графических образов, посылаемых на отображение, увеличилась нагрузка на центральный процессор. С тем, чтобы ее снять, часть функций по обработке изображений (в данном случае более уместно слово сцен, поскольку нагрузка на видеокарту проявляется в основном в играх) и выполняют видеокарты. Можно сказать, что современная видеокарта представляет собой самостоятельный компьютер. Обобщенно схему видеокарты можно представить так (рис. 2.4.).

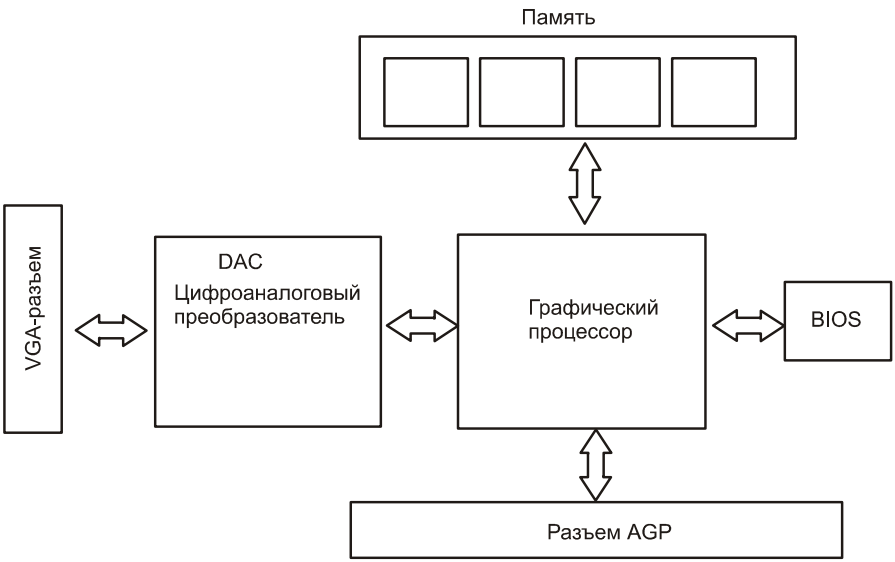

Рис. 2.4. Схема видеокарты

Условно можно разбить современную видеокарту на следующие компоненты:

- BIOS (Basic Input/Output System) — базовая система ввода-вывода. Базовая система вывода видеокарты аналогична BIOS материнской платы. Базовая система ввода-вывода находится в энергонезависимой памяти, то есть в ПЗУ. BIOS видеокарты хранит основные команды интерфейса

видеоадаптера. Благодаря наличию собственного встроенного BIOS, видеокарта начинает работать сразу же после включения питания. Подтверждением этому служит поведение компьютера во время загрузки. Первой после включения появляется надпись о модели и характеристиках видеокарты, а уже потом запускается собственно система, отражается наличие устройств, подключенных к материнской плате, распределение их по прерываниям и т. д. То есть видеокарта начала работать раньше, чем вся остальная система. В большинстве современных видеокарт для хранения BIOS используются электрически перепрограммируемые микросхемы, это позволяет обновлять, по мере необходимости, BIOS видеокарты.

Как и к модернизации BIOS материнской платы, к модернизации BIOS видеоадаптера надо подходить очень осторожно, и без необходимости лучше таких манипуляций не производить.

- Графический процессор, или чипсет. Это центральная часть видеокарты. Именно здесь обсчитывается сцена, и производительность видеокарты во многом определяется именно чипсетом. Наличие графического процессора позволяет разгрузить центральный процессор от обязанностей просчета сцен, что повышает производительность системы.

  Важными параметрами графического процессора, как элемента видеокарты, являются частота, на которой работает ядро процессора, разрядность шины, с которой идет обмен данными с видеокартой, и частота шины памяти. Версия шинного интерфейса с материнской платой (AGP 4x–8x или PCI-E). Быстродействие видеокарты определяется не только процессором, хотя он конечно важен. В процессе работы у процессора возникает необходимость в промежуточном хранении данных. Использовать для этих целей основную память, конечно, можно, но это снизит производительность системы, поскольку придется постоянно перекачивать данные между основной памятью и видеокартой, что накладно. (Такой вариант используется крайне редко.) Поэтому в видеокарте присутствует собственная память, с которой и работает графический процессор. При одинаковой модели графического процессора производительность видеокарты определяется акселератором.

- Память — важный составляющий элемент видеокарты. В памяти хранятся результаты работы графического процессора по просчету сцены. Важнейшими характеристиками видеопамяти являются ее тип, разрядность, объем и частота работы. Современные видеокарты имеют объем памяти 64–256 Мбайт, разрядность шины 128–256 бит и частоты памяти порядка 450–600 МГц.

- DAC (Digital to Analog Converter) — цифроаналоговый преобразователь (ЦАП). Назначение цифроаналогового преобразователя состоит в преобразовании цифровой сцены, просчитанной процессором, в аналоговый сигнал с целью последующей подачи в монитор. Быстродействие цифроаналогового преобразователя измеряется в мегагерцах и на сегодняшний момент достигает нескольких сотен МГц. Чем больше частота преобразования, тем выше частота регенерации по вертикали адаптера. Для современных адаптеров частота вертикальной регенерации может достигать 100 Гц при разрешении 1900×1200 пикселов.

  В современных видеокартах цифроаналоговый преобразователь конструктивно выполнен в одном корпусе с графическим процессором.

Схема работы видеокарты выглядит следующим образом. Данные от центрального процессора, необходимые для просчета графической сцены, через порт (AGP, PCI-E) попадают в графический процессор, где они обрабатываются. Результаты промежуточных вычислений сцены, а также данные об окончательно построенной сцене хранятся в памяти.

Обработанная и построенная сцена изображения хранится в цифровом виде в памяти. Из памяти данные сцены попадают в цифроаналоговый преобразователь, где превращаются в аналоговую форму и выдаются на монитор с последующим отображением.

2.4. Установка видеокарты и подключение монитора

Видеокарты различаются по ряду параметров, таких как чипсет, объем памяти и др. Но в данном разделе нас интересует тип порта, по которому видеокарта подключается к материнской плате. Старые видеокарты имели разъем PCI, современные видеокарты имеют разъем AGP или PCI-E (рис. 2.5).

Причем разъем последнего типа появился совсем недавно и скоро вытеснит шину AGP. В некоторых компьютерах, где не требуется большая производительность видеосистемы, могут встречаться даже видеокарты PCI. Механическая установка видеокарт всех трех типов по своей сути одинакова.

1. Примеряем видеокарту к разъему и определяем слот в корпусе, в который будет устанавливаться.
2. Снимаем заглушку в корпусе. Она может выламываться или снимается (после предварительного откручивания винта) (рис. 2.6).
3. Устанавливаем видеокарту, она должна стать плотно. После установки закрепляем видеокарту винтом (рис. 2.7).

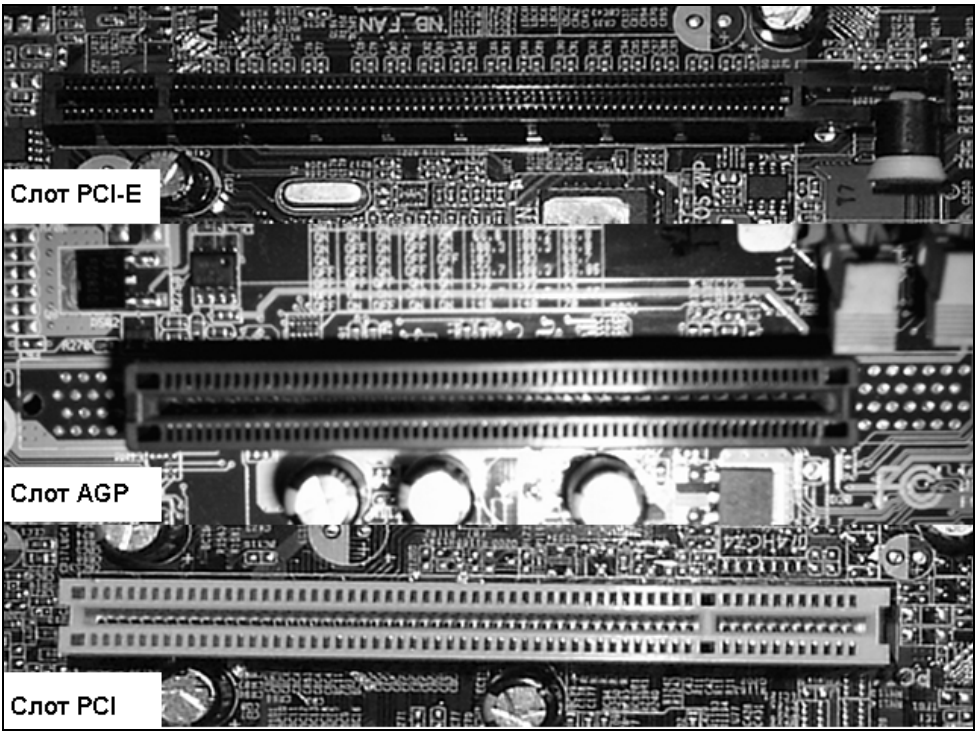

Рис. 2.5. Разъемы PCI, AGP, PCI-E

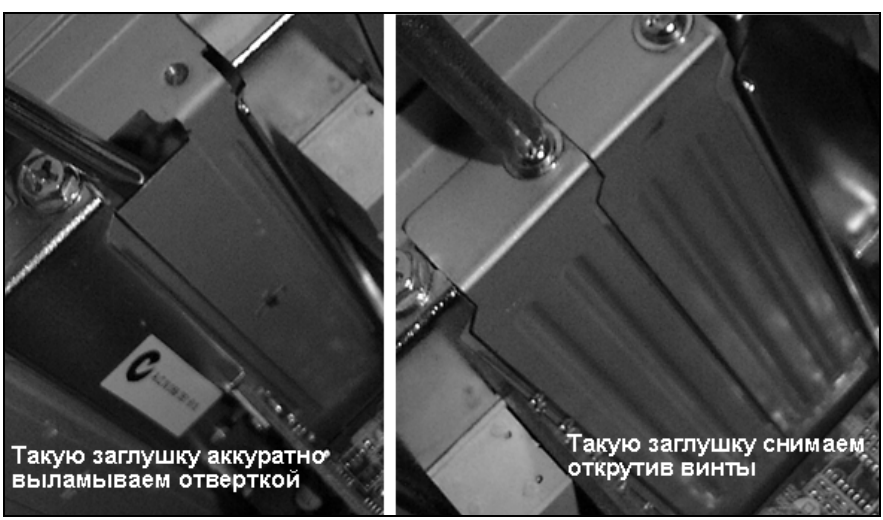

Рис. 2.6. Удаление заглушек

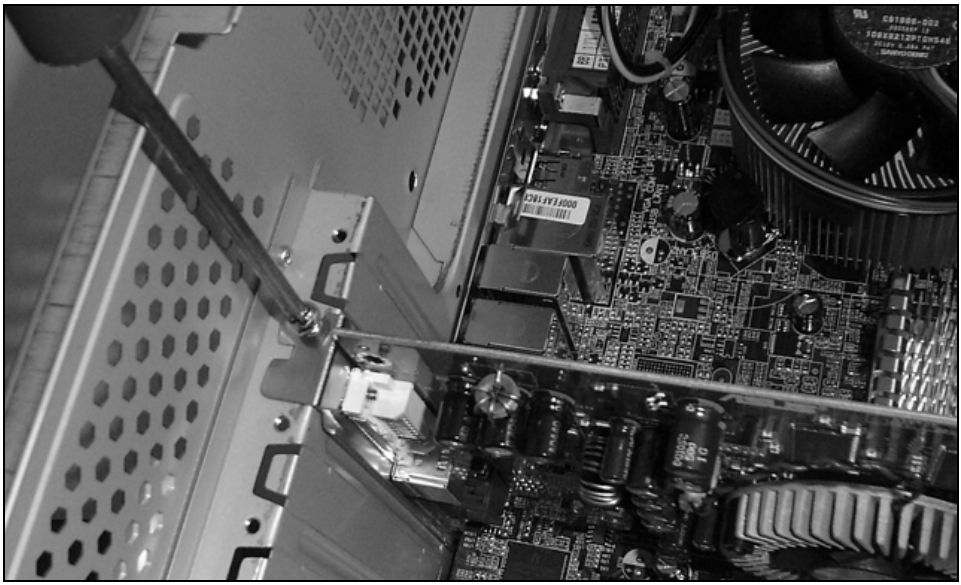

Рис. 2.7. Установка видеокарты

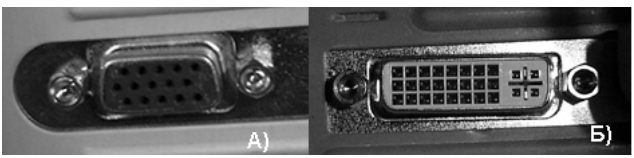

Рис. 2.8. Разъемы VGA и DVI-I

4. Подключение монитора зависит от видеокарты и от монитора. На данный момент наиболее распространены два типа стандартов подключения монитора. Первый — это старый знакомый VGA-разъем на 15 контактов — это аналоговое подключение (рис. 2.8, *а*). Второй — это DVI-I (рис. 2.8, *б*) (реже DVI-D) — предназначен для подключения цифровых мониторов. Однако по нескольким контактам в DVI-стандарте передается аналоговый сигнал, поэтому при использовании специального переходника можно к DVI-разъему подключить и аналоговый монитор.

2.5. Настройка графического режима

Во время установки Linux вы уже должны были столкнуться с настройкой графического режима. Возможно, в процессе установки не удалось правильно настроить графический режим, или он был настроен с ошибками, и у вас

возникли проблемы с запуском графического режима. В этом разделе мы разберемся, как их устранить.

Проблемы с настройкой видеокарты проявляются по-разному. Это может быть и черный экран при переходе в графический режим, и цветной мусор на экране. Возможно, просто графический режим даже не может запуститься. Могут быть и не столь явные неполадки, например, графический режим запускается, но в некоторых случаях начинает давать сбои. Например, запускаете программу, а у вас система выпадает в текстовый режим.

Первоначальная настройка графического режима осуществляется из текстового режима. Если у вас по умолчанию стоит графический вход в систему, то необходимо выйти в текстовый режим. Для этого нажмите одновременно <Ctrl>+<Alt>+<Backspace>, в результате X-сервер будет убит, и вы останетесь в графическом режиме.

Дальнейшая настройка графической среды зависит от вашего дистрибутива, в большинстве случаев помогут утилиты XF86Setup, lizard. В RedHat и ASPLinux это могут быть setup и Xconfigurator, в Mandrake это Xfdrake.

Рассмотрим настройку графического режима для Mandrake 10. Входим в систему как root и набираем в командной строке

`Xfdrake`,

появится заглавная страница (рис. 2.9).

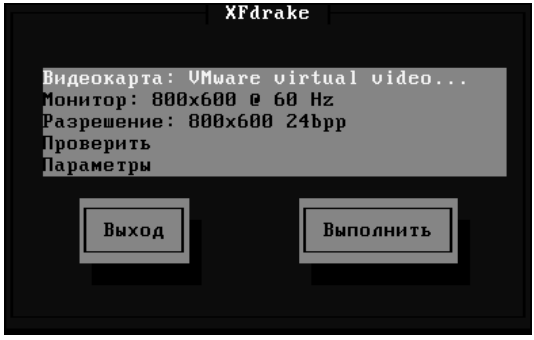

Рис. 2.9. Главное окно XFdrake

Стрелками вверх и вниз выбираем нужный пункт. Затем клавишей <Tab> позиционируемся на кнопке **Выполнить** и нажимаем <Enter>.

Самая верхняя строка позволяет выбрать тип видеокарты (рис. 2.10, *а*). Следующая строка выбирает тип монитора (рис. 2.10, *б*). Третья сверху строка позволяет установить разрешение и глубину цвета (рис. 2.10, *в*).

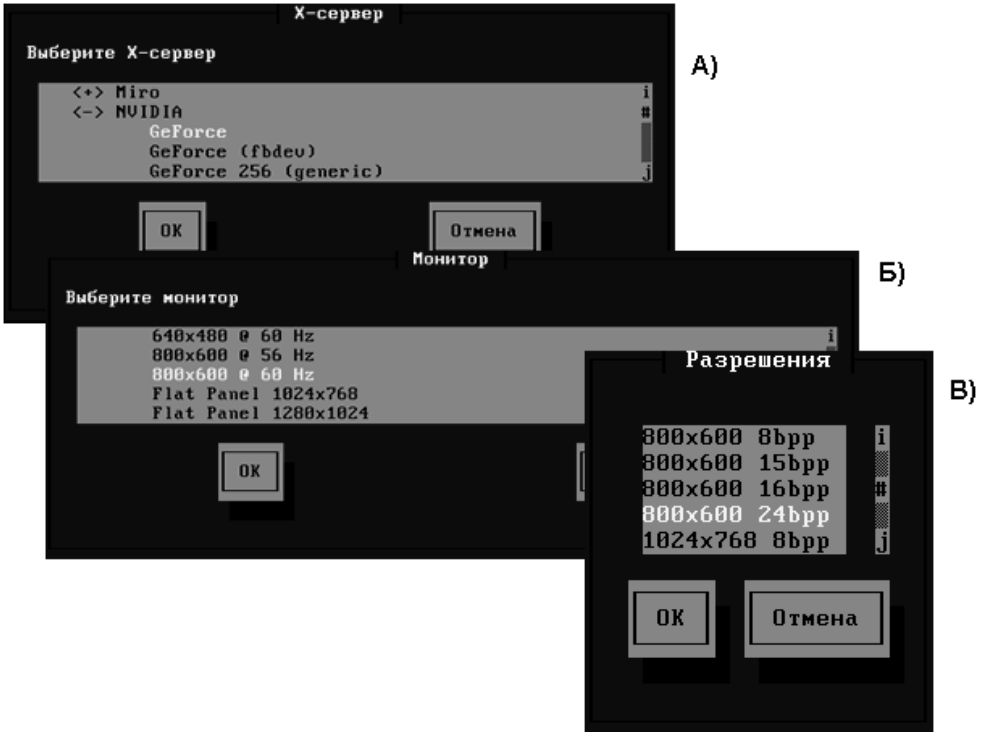

Рис. 2.10. Настройка режимов в XFdrake

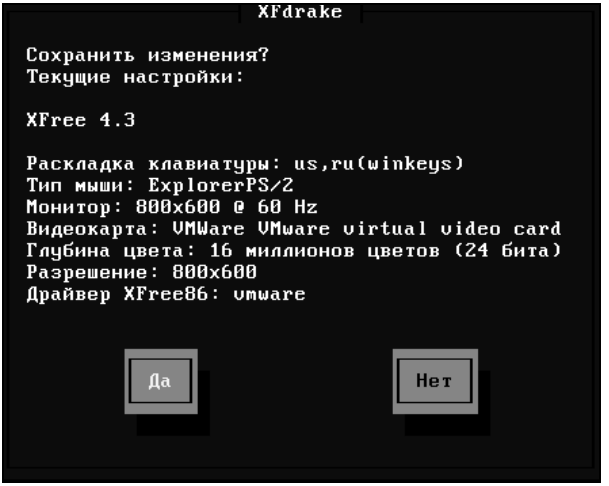

Рис. 2.11. Сохранение конфигурации

После настройки и тестирования нужно сохранить параметры, для этого выбираем кнопку **Выход** и на вопрос о сохранении конфигурации подтверждаем сохранение конфигурации (рис. 2.11).

Воспользовавшись одной из приведенных утилит, вы настроите графический режим, но это еще не все. Необходимо отрегулировать изображение так, чтобы оно развертывалось на полный экран, не было смещения изображения по горизонтали и вертикали. Для этого можно воспользоваться утилитой xvidtune (рис. 2.12).

Рис. 2.12. Вид утилиты xvidtune

Утилита xvidtune призвана облегчить настройку графического режима. К сожалению, xvidtune непосредственно не вносит изменения в файлы конфигурации, она лишь дает строку, которая определяет параметры вашей системы.

Далее описаны действия по настройке графического режима.

1. В командной строке набираем

   `xinit`

   и запускаем Xserver.

2. Теперь набираем `xvidtune`. Появляется предупреждение, которое говорит о том, что вы используете программу на свой страх и риск. Жмем **OK**. Нажмем также кнопку **Auto**, она позволит видеть изменения сразу же. Далее, используя кнопки **Left**, **Right**, **Up**, **Down**, позиционируем изображение по центру экрана. А затем, используя **Wider**, **Narrower**, **Shorter**, **Taller**, развертываем изображение на весь экран. Ну вот, полдела сделано.

3. Теперь нажимаем кнопку **Show**, которая выведет необходимую строку настройки, запомните ее, а лучше запишите. Теперь выходим из программы, нажав **Quit**.
4. Выходим из графического режима, нажав <Ctrl>+<Alt>+<Backspace>.
5. Запускаем файл-менеджер Midnight Commander, для этого в командной строке набираем `mc`.
6. Подсвечиваем курсором файл /etc/x11/xorg.conf (для ASPLinux) (/etc/x11/XF86Config-4 для Mandrake) (/etc/x11/XF86Config для Red Hat) и нажимаем клавишу <F4> на клавиатуре. (В некоторых системах файл может быть в /etc.) В редактируемом файле необходимо найти секцию `monitor` и добавить туда строку

   ```
   modeline
   ```

   и ту строку, которую вы только что получили. Записываем файл.
7. Убиваем графический сервер, нажав <Ctrl>+<Alt>+<Backspace>.
8. Перезагружаем Xserver, набрав `startx` (в этом случае, помимо графического режима, запустится сразу и графическая оболочка). Должно получиться изображение без искажений.

2.6. Установка драйверов от nvidia

Развитие техники идет достаточно быстро, в продаже появляются новые видеокарты, латаются дыры в драйверах. Может статься, если нет новой версии Linux, поддерживающей вашу карту, придется скачивать ее с сайта производителя.

Большим плюсом является то, что разработчики постоянно обновляют версии драйверов. Кроме того, установка драйверов от nvidia становится простым делом.

Для видеокарт на основе чипсета nvidia это можно сделать по адресу **http://www.nvidia.ru/drivers/drivers.shtml**, а ссылка на один из скачиваемых файлов выглядит следующим образом: **http://download.nvidia.com/XFree86/Linux-x86/1.0-7667/NVIDIA-Linux-x86-1.0-7667-pkg1.run**.

1. Итак, первым делом вы загружаете файл с драйверами и сохраняете в специально созданном каталоге.
2. Входим в систему под root. Если до этого вы присутствовали в системе под другим именем, можно выполнить команду `su` без параметров, а затем по запросу ввести пароль (рис. 2.13).

Рис. 2.13. Смена пользователя на root

3. Переходим в каталог, где располагается драйвер, и запускаем файл на исполнение (рис. 2.14).

```
[root@localhost /]# cd /download/driver/nvidia
[root@localhost nvidia]# ./NVIDIA-Linux-x86-1.0-7667-pkg1.run
```

Рис. 2.14. Запуск на исполнение установки драйвера

Запущенный нами на исполнение файл — это самораскрывающийся архив. При запуске он извлекает содержание архива и запускает на исполнение nvidia-инсталлятор (обеспечивает интерактивный интерфейс, которым вы выполните инсталляцию). В принципе после раскрытия архивов от настройки можно отказаться и выполнить ее позднее.

При запуске файла доступно несколько ключей:

- `--info` — вывод информации относительно текущего файла и последующий выход;
- `--check` — проверка целостности архива и выход;
- `--help` — вызов справки;
- `--extract-only` — извлечение файлов без запуска программы инсталлятора;
- `--uninstall` — в течение инсталляции инсталлятор будет делать резервные копии любых файлов и на их место записывать новые файлы. Опция

деинсталляции отменяет установку, восстанавливая систему к ее состоянию до установки.

Если вы читали главу внимательно, то должны были заметить, что в различных версиях Иксов используются различные версии файлов конфигураций. В 2004 г. была выпущена новая версия сервера, который использует файл конфигурации /etc/X11/xogr.conf. В дальнейшем планируется, что все дистрибутивы перейдут на использование этого файла. Старые версии используют файл /etc/X11/XF86Config. Синтаксис файлов одинаков, поэтому все сказанное позже будет справедливо для обеих версий файлов.

Сгенерировать файл конфигурации можно при помощи специальных утилит, о которых рассказывалось ранее. Если у вас уже есть файл конфигураций, но Иксы не хотят работать, необходимо проверить этот файл вручную. Это можно сделать при помощи программы Midnight Commander (mc). Для ее запуска в командной строке введите:

```
#mc
```

Может случиться так, что Midnight Commander у вас не установлен. В этом случае можно воспользоваться любым известным вам текстовым редактором, например vi. При редактировании обратите внимание на следующие строки:

- в секции `Devices` удалите строки типа

  ```
  Driver "nv" (or Driver "vesa") (or Driver "fbdev")
  ```

  и замените ее строкой

  ```
  Driver "nvidia";
  ```

- в секции `Module` удалите строки

  ```
  Load "dri"
  Load "GLCore"
  ```

  и добавьте строку (если ее нет):

  ```
  Load "glx".
  ```

После внесения указанных изменений и перезапуска Иксов система должна обратиться к новым библиотекам.

Вообще файл конфигураций может содержать много опций — полный их список со значениями, выполняемыми данными опциями, вы найдете в документации, поставляемой с драйвером.

Дополнительную информацию по настройке системы можно получить в страницах руководства, введя в командной строке:

```
man XF86Config
```

или

```
man xorg.conf.
```

Некоторые проблемы могут возникнуть в случае, если вы установили драйвер, а затем установили на ядро заплатку (или установили новое ядро). Дело в том, что интерфейс ядра nvidia должен быть скомпилирован с учетом версии ядра операционной системы, в этом случае лучше повторить установку драйвера.

Если какие-то проблемы все же остались, то можно воспользоваться официальным FAQ, взять его можно на сайте **http://www.nvidea.ru** или обсудить вопрос на форуме, расположенном там же.

2.7. Установка драйверов от ATI

Первым делом, конечно, необходимо заполучить самые последние драйверы. Сделать это можно как с сайта производителя видеокарты, так и с официального сайта ATI, в разделе drivers & software (**http://www.ati.com**). Как правило, это будет файл, название которого будет напоминать нечто похожее на fglrx_4_1_0-8.18.6-1.i386.rpm.

1. После того как загрузили файл с сайта производителя, сохраним его в специально созданном каталоге.
2. Входим в систему под root. Если до этого вы присутствовали в системе под другим именем, можно выполнить команду `su` без параметров, а затем по запросу ввести пароль (см. рис. 2.13).
3. Переходим в каталог, где располагается драйвер (rpm-пакет), и даем команду на его установку. Например:

   ```
   rpm – i fglrx_4_1_0-8.18.6-1.i386.rpm
   ```

4. После распаковки пакета запустится скрипт. Этот скрипт будет задавать вопросы. Если этого не произошло, наберите в командной строке принудительную настройку командой `fglrxconfig`.
5. Созданный скриптом модуль будет располагаться в каталоге /lib/modules/fglrx.
6. Необходимо его установить; для этого вводим

   ```
   ./make_install.sh
   ```

   (при этом вы должны располагаться в каталоге /lib/modules/fglrx, то есть там, где находится модуль).
7. Загружаем модуль в ядро:

   ```
   depmod –a
   modprobe fglrx
   ```

Предложенный вариант при загрузке модуля проверит зависимость между модулями, что дополнительно обезопасит от необратимых деструктивных изменений по вине пользователя.

8. Также необходимо в файле /etc/X11/xogr.conf в секцию `Modules` добавить строку:

   ```
   Load "glx"
   ```

Теперь, как и в предыдущем случае, после выполнения указанных действий и перезапуска Иксов система должна обратиться к новым библиотекам.

Вообще файл конфигураций может содержать много опций — полный их список вы найдете в документации, поставляемой с драйвером.

Как и в случае с nvidia, могут возникнуть проблемы, если вы меняли ядро. Как вы уже знаете, интерфейс ядра видеокарты должен быть скомпилирован с учетом версии ядра операционной системы, в этом случае лучше повторить установку драйвера.

В этом случае перейдите в каталог /lib/modules/fglrx/build_mod. Введите команду:

```
/make.sh
```

Остается его только установить, как это было показано несколько раньше. Перейдите в каталог /lib/modules/fglrx и введите

`/make_install.sh`.

Глава 3

Linux и жесткие диски

3.1. Принципы работы жестких дисков

Жесткий диск является, пожалуй, самым интересным и сложным устройством всего компьютера. Жесткий диск на сленге называют еще винчестером (винтом) или хардом (от Hard Disk Drive). Назначение жесткого диска состоит в долговременном хранении данных. Современные жесткие диски — это сложные устройства, однако общие принципы их работы остались неизменными с момента их создания.

Основу жесткого диска составляют пластины со специальным покрытием, на которые осуществляется запись и последующее чтение. Запись осуществляется на обеих сторонах пластины.

Рабочая поверхность диска разбита на дорожки и секторы. Одноименные дорожки, расположенные на различных пластинах, образуют цилиндры (рис. 3.1).

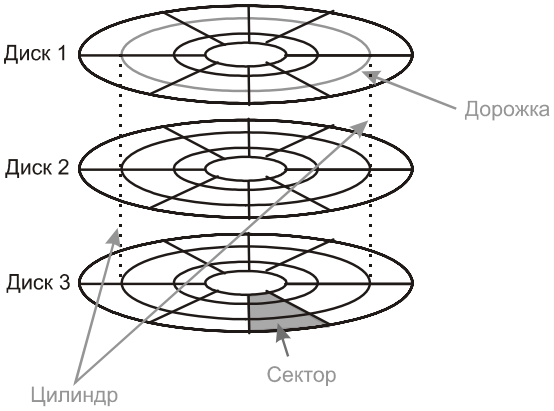

Рис. 3.1. Дорожки сектора – цилиндры

Дорожка представляет собой кольцо данных на поверхности одного из дисков. Дорожки — это относительно большие участки для хранения информации, и поэтому для размещения информации дорожки разбивают на секторы — нумерованные участки дорожки. В начале каждого сектора записывается заголовок, по которому определяется начало сектора и его номер, а в конце записывается контрольная сумма, по которой вычисляется целостность информации, хранящейся в секторе.

3.2. Конструкция жесткого диска

Конструкция жесткого диска приведена на рис. 3.2.

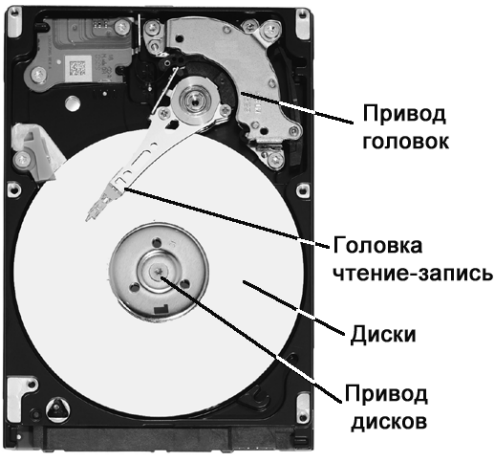

Рис. 3.2. Основные элементы винчестера

Как видно из этой схемы, жесткий диск состоит из следующих элементов:

- собственно диски;
- головки записи-чтения;
- схема управления;
- привод дисков;
- привод головок.

Поскольку плотность записи на диски велика, то поверхность диска должна быть идеальной. Поэтому для защиты поверхностей дисков от повреждений, а также прецизионной механики управления головками, все элементы жесткого диска, кроме платы управления, размещаются в герметичном корпусе.

3.2.1. Пластины и рабочий слой диска

Конструктивно винчестер содержит несколько пластин (блинов), на которых производится запись. Диски закреплены на оси-шпинделе. Ось приводится в движение двигателем привода дисков. Частота вращения может достигать 15 000 об/мин. Однако в большинстве дисков, поступающих в широкую продажу, как правило, используется частота 7 200 об/мин.

Пластина состоит из основы и рабочего слоя (обладающего определенными свойствами).

Первоначально пластины дисков изготавливали из алюминия. Однако рост емкости дисков привел к увеличению плотности записи на одну пластину. Дисковые пластины из алюминия уже мало отвечали этим потребностям, и сегодня производители переходят на пластины из композитных материалов на основе керамики или стекла.

Рабочий слой жесткого диска выполняется из материала с магнитными свойствами, сохраняющего остаточную намагниченность после воздействия на него магнитным полем, созданным головкой чтения/записи.

Рабочий слой жесткого диска может изготавливаться из различных материалов. Однако самыми распространенными типами рабочего слоя являются:

- оксидный слой — полимерное покрытие, рабочим веществом в котором является окись железа. Такие диски имеют желто-коричневый цвет;
- тонкопленочный слой — сплав кобальта толщиной 0,025 мкм. На поверхность дисков рабочий слой наносится методом электролиза или методом напыления в вакууме. Затем сверху наносится защитный слой из углерода;
- AFC — двойной антиферромагнитный слой. Накопители AFC состоят из двух магнитных слоев, разделенных тонкой пленкой. Такая конструкция позволяет использовать более толстые магнитные слои, имеющие более устойчивые кристаллы.

3.2.2. Головки чтения/записи

Принцип работы жесткого диска состоит в намагничивании определенных областей диска. Собственно намагничивание и последующее чтение определенных областей поверхности пластин выполняют головки чтения/записи. Для каждого рабочего слоя (стороны пластины) предусмотрена своя головка чтения/записи. Однако все головки смонтированы в едином каркасе и перемещаются одновременно.

В спокойном состоянии, когда пластины не раскручены, головки прижаты к поверхности диска. Как только двигатель приводит пластины во враща-

тельное движение, воздушный поток отрывает блины от поверхности, и они как бы парят над поверхностью диска.

Перемещение головки относительно рабочего слоя осуществляется в двух направлениях. Первое — за счет вращения дисков вокруг своей оси. Второе перемещение магнитной головки (от центра диска к краям) осуществляется специальными приводами (рис. 3.3).

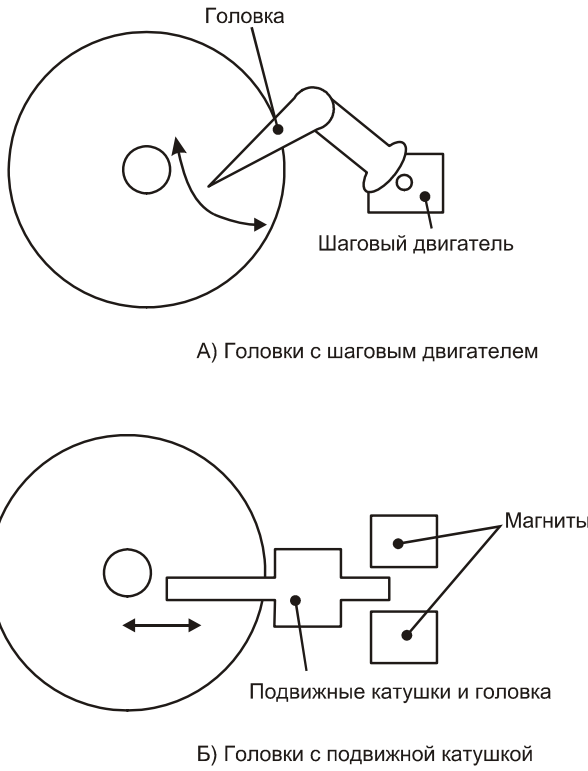

Рис. 3.3. Схема перемещения магнитных головок

Существуют два основных типа конструкций магнитных головок: привод с шаговым двигателем и с подвижной катушкой. Привод с шаговым двигателем сейчас не используется. Принцип работы подвижной катушки похож на принцип работы громкоговорителя. Подвижная катушка находится в поле постоянного магнита, при протекании тока через подвижную катушку образуется электромагнитное поле, которое, взаимодействуя с полем магнита, перемещает катушку относительно этого магнита — притягивает ее или отталкивает. Поскольку, в отличие от шагового двигателя, в такой системе нет

фиксированных положений, то для окончательного позиционирования используется специальная система, называемая сервоприводом. Эта система позиционирует головки на начальную дорожку. Существует несколько систем позиционирования, отличающихся технологией установки на начальную дорожку.

Система перемещения головок может быть линейной, когда головки перемещаются по прямой, от внешних дорожек к внутренним. А может быть и поворотной, когда головки крепятся на концы рычагов. Поворачиваясь, рычаги совершают перемещение от внешних дорожек к внутренним.

3.3. Некоторые параметры жестких дисков

С точки зрения пользователя, наибольший интерес вызывают такие параметры винчестера, как интерфейс и емкость.

3.3.1. Интерфейс

Современные винчестеры выпускаются с интерфейсами нескольких типов: IDE, SATA, SCSI. Причем в настольных компьютерах используются только диски первых двух типов. Диски SCSI используются только на серверах и там, где требуется повышенное быстродействие.

Интерфейсный разъем IDE содержит 40 контактов. Также 40 контактов содержит и его ответная часть на материнской плате. Однако соединяются они кабелем, который содержит 80 жил. Дело в том, что между всеми сигнальными контактами проходит одна дополнительная жила заземления. Это исключает взаимное влияние сигнальных проводников друг на друга.

Интерфейс IDE прошел в своем развитии несколько ступеней, с переходом на каждую ступень скорость передачи данных увеличивалась (табл. 3.1).

Таблица 3.1. *Развитие интерфейса IDE*

Тип интерфейса	Скорость шины	Год внедрения
ATA-2 PIO 3	13,3 Мбит/с	1993
ATA-2 PIO 4	16,6 Мбит/с	1994
UDMA 33	33 Мбит/с	1997
UDMA 66	66,6 Мбит/с	1999
UDMA 100	100 Мбит/с	2000
UDMA 133	133 Мбит/с	2002

В настоящее время более популярным становится интерфейс Serial ATA. В отличие от IDE интерфейса, где данные передаются параллельно по большому числу жил, в интерфейсе SATA данные передаются последовательно. Переход от параллельной передачи данных к последовательной снимает проблемы синхронизации, возникающие на высоких частотах передачи у IDE-кабелей. Эти проблемы обусловлены тем, что при высоких частотах наблюдается разброс по времени прохождения сигналов по различным жилам. В результате данные приходят не одновременно, что усложняет их синхронизацию.

Еще одним преимуществом интерфейса SATA является то, что интерфейсный кабель более узкий, это не загромождает пространство внутри корпуса и способствует лучшей вентиляции. Помимо изменения информационного разъема, изменился и разъем питания. Для того чтобы подключить питание, выпускаются специальные переходники. Кроме того, на некоторых моделях винчестеров установлены два типа разъемов питания. На рис. 3.4 приведен разъем SATA и кабель SATA.

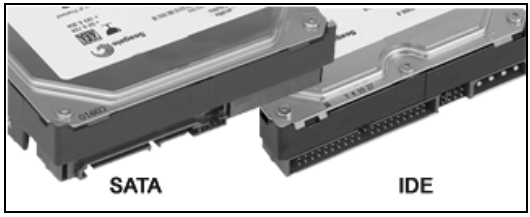

Рис. 3.4. Интерфейсы IDE и SATA

Интерфейс SCSI, как правило, больше используется в серверных решениях. Интерфейс SCSI имеет много преимуществ, по сравнению с традиционным IDE. В частности, это касается более быстрого обмена данными. Длина кабеля может достигать десятка метров, и это не будет мешать нормальной работе.

Основное преимущество SCSI — это большая эффективная пропускная способность шины. Достигнута она увеличением тактовых частот и изменением протоколов. Существует большое количество вариантов SCSI (табл. 3.2).

Главным преимуществом интерфейса SCSI является его высокая скорость, поэтому он используется для серверных решений, а также в тех местах, где требуется высокая производительность. Винчестеры со SCSI-интерфейсом могут иметь скорость оборотов шпинделя до 15 000 об/мин. Для винчестеров IDE в этом нет необходимости, так как такую пропускную способность IDE просто не выдаст.

Таблица 3.2. *Варианты интерфейса SCSI*

Тип SCSI	Скорость шины	Количество устройств
SCSI 1	5 Мбит/с	8
Fast SCSI	10 Мбит/с	8
Fast Wide SCSI	20 Мбит/с	16
Ultra SCSI	20 Мбит/с	8
Wide Ultra SCSI	40 Мбит/с	16
Ultra 2 SCSI	40 Мбит/с	8
Wide Ultra 2 SCSI	80 Мбит/с	16
Ultra 3 SCSI	160 Мбит/с	16
Ultra 320	320 Мбит/с	16

3.3.2. Емкость

Емкость является важной пользовательской характеристикой жесткого диска. Особенно важно это стало с ростом производительности компьютеров в целом. Как только производительность достигла таких высот, что стало возможным обрабатывать мультимедийные данные, к емкости жестких дисков стали предъявляться повышенные требования. Современные винчестеры достигают емкости сотен гигабайт.

При замене старого винчестера на новый, большего объема, необходимо помнить, что материнская плата вашего компьютера может не поддерживать винчестеры такого большого объема.

Характеризуют жесткий диск также объем кеш-памяти, время доступа, время позиционирования и другие, но их рассмотрение мы отложим.

3.4. RAID-массивы

Основные требования, предъявляемые пользователем к жестким дискам:

- скорость обмена данными;
- надежность хранения данных;
- большой объем дискового пространства.

Естественно, что реализация всех этих трех требований в одном устройстве имеет физические ограничения, налагаемые доступной в данный момент технологией. Одним из красивых решений является организация RAID-массивов.

Слово RAID — это сокращение от Redundant Array Independed Disk (избыточный массив независимых дисков). То есть организация RAID-массива предполагает наличие, как минимум, двух дисков. В самом простом случае так и есть.

RAID-массивы могут быть аппаратными и программными. Аппаратный RAID-массив представляет собой контроллер, вставляемый в один из слотов расширения материнской платы. К этому контроллеру уже непосредственно подключаются винчестеры. Аппаратные контроллеры RAID-массива могут быть и интегрированы в материнскую плату.

Альтернативным вариантом аппаратному RAID-массиву является программный RAID-массив. Преимущество — более дешевая реализация, недостатком является более низкая производительность такого массива и дополнительная нагрузка на процессор.

Существует шесть основных типов RAID-массивов (табл. 3.3).

Таблица 3.3. *Типы RAID-массивов*

Тип RAID	Описание
0	Поток данных распределяется на несколько дисков. Контроль четности не производится. Суммарная емкость массива равна сумме емкостей дисков, входящих в массив. Плюсом является увеличение дискового пространства и повышение скорости обмена с диском. Минусом является снижение надежности, так как при выходе из строя одного из жестких дисков вся информация будет утеряна
1	Mirrored Array, или зеркало. При таком массиве данные полностью дублируются на носителях. Емкость массива равна емкости наименьшего диска (но, как правило, используются диски одинаковых моделей, поэтому можно сказать, что емкость массива равна емкости одного диска). Такой тип массива используется, когда предъявляются повышенные требования к надежности
2	Parallel Array with ECC — параллельный массив с коррекцией ошибок. Суть метода заключается в последовательной записи на несколько винчестеров блоков данных. При этом один из винчестеров, входящих в массив, выделяется для записи кодов контроля и исправления ошибок. Для контроля используется метод ECC. Головки винчестеров при записи двигаются синхронно
3	Parallel Array white Parity — параллельный массив с контролем четности. Аналогичен предыдущему, но для контроля четности используется метод исключающего "или". Данные контроля четности пишутся на отдельный диск. В массиве должно быть минимум 3 диска

Таблица 3.3 *(окончание)*

Тип RAID	Описание
4	Striped Array with Parity — полосный массив с контролем четности. Данные контроля четности так же, как и в предыдущем случае, записываются на отдельный диск. Но данные при этом распределяются по байтам, а не по секторам, как в двух предыдущих случаях
5	Stripping Array with Rotating Party — полосный массив со смешанным контролем четности. Данные и контрольные суммы записываются вперемешку по всем дискам. При выходе из строя одного из носителей данные восстанавливаются по контрольным суммам. В случае программной реализации это наиболее трудоемкий метод

3.5. Механическое подключение жесткого диска

Подключение нового жесткого диска осуществляется только при выключенном компьютере. Современные блоки питания и материнские платы устроены так, что питание на материнскую плату подается даже тогда, когда компьютер выключен. Необходимо полностью обесточить компьютер. Для этого выдерните шнур питания из разъема, расположенного в задней части корпуса. Теперь можно приступать к установке.

1. Первым делом устанавливаем винчестер на посадочное место в корпусе и закрепляем его на четыре винта, как показано на рисунке. Винты должны подходить по размеру, то есть не болтаться и в то же время закручиваться без усилий. Прочно закрутите винты (рис. 3.5).

Рис. 3.5. Установка винчестера на посадочное место

2. Подключаем кабель IDE или SATA в зависимости от того, какой винчестер вы купили. Вначале подключаем кабель к материнской плате, затем уже к винчестеру, обратите внимание на ключи (рис. 3.6).

Рис. 3.6. Подключение кабеля SATA

3. Последний этап — подключение питания к винчестеру. В зависимости от модификации винчестера, это может быть как стандартный разъем питания, так и новый, используемый на SATA-винчестерах. Для новых разъемов питания необходимо использовать переходники. При подключении питания обратите внимание на ключи (рис. 3.7).

Рис. 3.7. Подключение кабеля питания

3.6. Подключение нового жесткого диска в Linux

Как бы ни были велики и быстры используемые вашей системой винчестеры, но рано или поздно у вас возникнет необходимость подключить новый жесткий диск. Собственно операцию подключения жесткого диска в ОС Linux можно разбить на несколько стадий:

- подключение жесткого диска к компьютеру (этот вопрос мы с вами уже рассмотрели);
- создание разделов на диске;
- создание файловой системы на каждом из разделов;
- организация автоматического монтирования дисков.

3.6.1. Создание разделов на диске (утилита fdisk)

Разбивка дисков на разделы похожа на подобную операцию в других операционных системах. В Linux для этих целей используется утилита fdisk.

Перед тем как перейти к разбивке жесткого диска на разделы, необходимо вспомнить, как "обзываются" жесткие диски в Linux. Для этого вам предстоит обратиться к табл. 1.1.

Далее мы будем приводить подключение нового SCSI-диска для компьютера, на котором установлен уже один SCSI-диск, на том находится система. Помимо этого, на компьютере имеются CD ROM, подключенный на IDE-канал, и дисковод флоппи-дисков.

1. Итак, входим с правами root в систему. Если вы уже находитесь сейчас под другим именем, то можно либо ввести команду `su` и затем пароль суперпользователя, либо ввести команду `exit` (тем самым вы выйдете из сеанса текущего пользователя). И войти в систему уже под другим именем.
2. Так как у нас второй SCSI-диск, то в командной строке вводим команду:

   `fdisk /dev/sdb`.

   Затем на запрос программы мы вводим букву `m` для просмотра помощи (рис. 3.8):

   - `d` — удалить раздел;
   - `n` — добавить новый раздел или логический диск;
   - `p` — посмотреть таблицу существующих разделов;
   - `q` — выйти без записи сделанных изменений;

- v — проверить таблицу разделов;
- t — сменить идентификатор раздела;
- w — записать таблицу разделов и выйти.

```
[root@localhost root]# fdisk /dev/sdb
Device contains neither a valid DOS partition table, nor Sun, SGI or OSF disklab
el
Building a new DOS disklabel. Changes will remain in memory only,
until you decide to write them. After that, of course, the previous
content won't be recoverable.

Warning: invalid flag 0x0000 of partition table 4 will be corrected by w(rite)

Command (m for help): m
Command action
   a   toggle a bootable flag
   b   edit bsd disklabel
   c   toggle the dos compatibility flag
   d   delete a partition
   l   list known partition types
   m   print this menu
   n   add a new partition
   o   create a new empty DOS partition table
   p   print the partition table
   q   quit without saving changes
   s   create a new empty Sun disklabel
   t   change a partition's system id
   u   change display/entry units
   v   verify the partition table
   w   write table to disk and exit
   x   extra functionality (experts only)
```

Рис. 3.8. Начало работы утилиты fdisk

3. Проверим наличие на диске уже существующих разделов. Это необходимо для того, чтобы избежать потери информации, находящейся на винчестере (быть может, там что-то ценное, а вы забыли). Кроме того, даже если ничего ценного нет, то перед тем, как приступить к созданию нового раздела, надо удалить старый. Для просмотра разделов необходимо ввести команду p. На рис. 3.9, *а* мы видим, что разделов на диске нет. Если бы разделы были, то программа fdisk выдала бы информацию, как на рис. 3.9, *б*.
4. Если диск не был пустым, то при необходимости можно удалить лишний раздел, введя команду d. Допустим, что наш диск был пустым, тогда необходимо создать новый раздел, введя команду n. Однако перед тем как

приступить к созданию нового раздела, необходимо вспомнить немного теории. Разделы в Linux могут быть двух типов:

- первичные (primary partition). Их может быть не более четырех;
- вторичные (extended partition). Их может быть сколько угодно, однако эти разделы не форматируются непосредственно, они предназначены для создания в них логических дисков.

```
Command (m for help): p

Disk /dev/sdb: 4294 MB, 4294967296 bytes
255 heads, 63 sectors/track, 522 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id  System

Command (m for help):
```

а

```
Command (m for help): p

Disk /dev/sda: 10.7 GB, 10737418240 bytes
255 heads, 63 sectors/track, 1305 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1   *           1         764     6136798+  83  Linux
/dev/sda2             765        1305     4345582+   5  Extended
/dev/sda5             765         827      506016   82  Linux swap
/dev/sda6             828        1305     3839503+  83  Linux

Command (m for help):
```

б

Рис. 3.9. *а* — пустой диск, *б* — этот диск не пустой

5. Создадим два раздела, один первичный (рис. 3.10) и один расширенный (рис. 3.11).

```
Command (m for help): n
Command action
   e   extended
   p   primary partition (1-4)
p
Partition number (1-4): 1
First cylinder (1-522, default 1): 1
Last cylinder or +size or +sizeM or +sizeK (1-522, default 522): 256

Command (m for help):
```

Рис. 3.10. Создание первичного раздела

```
Command (m for help): n
Command action
   e   extended
   p   primary partition (1-4)
e
Partition number (1-4): 2
First cylinder (257-522, default 257):
Using default value 257
Last cylinder or +size or +sizeM or +sizeK (257-522, default 522):
Using default value 522
```

Рис. 3.11. Создание расширенного раздела

```
Command (m for help): n
Command action
   l   logical (5 or over)
   p   primary partition (1-4)
l
First cylinder (257-522, default 257):
Using default value 257
Last cylinder or +size or +sizeM or +sizeK (257-522, default 522):
Using default value 522
```

Рис. 3.12. Создание логического диска

```
Command (m for help): t
Partition number (1-5): 1
Hex code (type L to list codes): L

 0  Empty           1c  Hidden W95 FAT3 70  DiskSecure Mult bb  Boot Wizard hid
 1  FAT12           1e  Hidden W95 FAT1 75  PC/IX           be  Solaris boot
 2  XENIX root      24  NEC DOS         80  Old Minix       c1  DRDOS/sec (FAT-
 3  XENIX usr       39  Plan 9          81  Minix / old Lin c4  DRDOS/sec (FAT-
 4  FAT16 <32M      3c  PartitionMagic   82  Linux swap      c6  DRDOS/sec (FAT-
 5  Extended        40  Venix 80286     83  Linux           c7  Syrinx
 6  FAT16           41  PPC PReP Boot   84  OS/2 hidden C:  da  Non-FS data
 7  HPFS/NTFS       42  SFS             85  Linux extended  db  CP/M / CTOS / .
 8  AIX             4d  QNX4.x          86  NTFS volume set de  Dell Utility
 9  AIX bootable    4e  QNX4.x 2nd part 87  NTFS volume set df  BootIt
 a  OS/2 Boot Manag 4f  QNX4.x 3rd part 8e  Linux LVM       e1  DOS access
 b  W95 FAT32       50  OnTrack DM      93  Amoeba          e3  DOS R/O
 c  W95 FAT32 (LBA) 51  OnTrack DM6 Aux 94  Amoeba BBT      e4  SpeedStor
 e  W95 FAT16 (LBA) 52  CP/M            9f  BSD/OS          eb  BeOS fs
 f  W95 Ext'd (LBA) 53  OnTrack DM6 Aux a0  IBM Thinkpad hi ee  EFI GPT
10  OPUS            54  OnTrackDM6      a5  FreeBSD         ef  EFI (FAT-12/16/
11  Hidden FAT12    55  EZ-Drive        a6  OpenBSD         f0  Linux/PA-RISC b
12  Compaq diagnost 56  Golden Bow      a7  NeXTSTEP        f1  SpeedStor
14  Hidden FAT16 <3 5c  Priam Edisk     a8  Darwin UFS      f4  SpeedStor
16  Hidden FAT16    61  SpeedStor       a9  NetBSD          f2  DOS secondary
17  Hidden HPFS/NTF 63  GNU HURD or Sys ab  Darwin boot     fd  Linux raid auto
18  AST SmartSleep  64  Novell Netware  b7  BSDI fs         fe  LANstep
1b  Hidden W95 FAT3 65  Novell Netware  b8  BSDI swap       ff  BBT
```

Рис. 3.13. Изменение идентификатора первичного раздела

```
14  Hidden FAT16 <3 5c  Priam Edisk     a8  Darwin UFS      f4  SpeedStor
16  Hidden FAT16    61  SpeedStor      a9  NetBSD          f2  DOS secondary
17  Hidden HPFS/NTF 63  GNU HURD or Sys ab  Darwin boot     fd  Linux raid auto
18  AST SmartSleep  64  Novell Netware  b7  BSDI fs         fe  LANstep
1b  Hidden W95 FAT3 65  Novell Netware  b8  BSDI swap       ff  BBT
Hex code (type L to list codes): 83
```

Рис. 3.14. Задание нового идентификатора

```
Command (m for help): t
Partition number (1-5): 5
Hex code (type L to list codes): 83

Command (m for help): p

Disk /dev/sdb: 4294 MB, 4294967296 bytes
255 heads, 63 sectors/track, 522 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/sdb1               1         256     2056288+  83  Linux
/dev/sdb2             257         522     2136645    5  Extended
/dev/sdb5             257         522     2136613+  83  Linux
```

Рис. 3.15. Задание нового идентификатора для логического диска и просмотр результатов

6. Теперь в расширенном разделе создаем новый логический диск, используя для него все доступное пространство (рис. 3.12).
7. Теперь для первичного диска и диска в расширенной области сменим тип файловой системы. Для изменения идентификатора раздела необходимо ввести `t`, выбрать номер раздела (в нашем случае это будет 1), затем ввести шестнадцатеричный код идентификатора раздела. Для подсказки можно ввести `L` (рис. 3.13).
8. Затем на запрос системы вводится идентификатор, в нашем случае это `83` (рис. 3.14 и 3.15).
9. Сохраняем сделанные изменения, для этого водим `w` (рис. 3.16).

```
Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.
Syncing disks.
```

Рис. 3.16. Сохранение сделанных изменений

10. После внесенных изменений необходимо перезапустить систему, это приведет к обновлению таблицы разделов.

3.6.2. Создание файловой системы на дисках (утилита mkfs)

Для того чтобы операционная система могла что-либо записать на диск, необходимо не только разбить его на разделы, но и отформатировать его. Для создания файловой системы используется утилита mkfs (сокращение от make filesystem — создать файловую систему). Итак, если посмотреть на рис. 3.17, то станет ясно, что форматировать мы будем диски /dev/sdb1 и /dev/sdb5.

```
[root@localhost root]# mkfs /dev/sdb1
mke2fs 1.34 (25-Jul-2003)
Filesystem label=
OS type: Linux
Block size=4096 (log=2)
Fragment size=4096 (log=2)
257536 inodes, 514072 blocks
25703 blocks (5.00%) reserved for the super user
First data block=0
16 block groups
32768 blocks per group, 32768 fragments per group
16096 inodes per group
Superblock backups stored on blocks:
        32768, 98304, 163840, 229376, 294912

Writing inode tables: done
Writing superblocks and filesystem accounting information: done

This filesystem will be automatically checked every 30 mounts or
180 days, whichever comes first.  Use tune2fs -c or -i to override.
[root@localhost root]#
```

Рис. 3.17. Форматирование диска

Наиболее интересными здесь являются два ключа:

- `c` — производит проверку на наличие плохих блоков. Это полезно, если используется старый диск.
- `t` — определяет тип файловой системы, которая будет сформирована.

3.6.3. Организация автоматического монтирования дисков

На данный этапе мы подключили диски механически, разбили их на разделы и отформатировали эти разделы, однако для того, чтобы данный диск стал доступен в операционной системе, этого недостаточно. Диски нужно еще смонтировать.

Монтирование осуществляется командой `mount`:

Синтаксис команды `mount`:

```
mount -a [-fFnrsvw] [-t Тип_ФС] [-O список_опций]
mount [-fnrsvw] [-o опции [,...]] устройство Точка_монтирования
mount [-fnrsvw] [-t Тип_ФС] [-o опции] устройство Точка_монтирования
```

Пример вызова:

```
#mount /dev/hda2 /home
```

Эта команда монтирует файловую систему диска hda2 (соглашение о наименовании устройств смотри далее) к каталогу /home корневой файловой системы.

При загрузке системы возможно автоматическое монтирование части файловых систем. Как правило, автоматически монтируются постоянно используемые системы. Список автоматически монтируемых файловых систем находится в файле /etc/fstab (листинг 3.1).

Листинг 3.1. Файл /etc/fstab

```
# /etc/fstab: static file system information.
#
# <file system> <mount point> <type> <options>      <dump> <pass>
/dev/sda1 / ext3 defaults 1 1
none /dev/pts devpts mode=0620 0 0
/dev/sda6 /home ext3 defaults 1 2
/dev/hdc /mnt/cdrom auto umask=0,user,iocharset=koi8-
u,codepage=866,noauto,ro,exec 0 0
none /mnt/floppy supermount dev=/dev/fd0,fs=ext2:vfat,--
,umask=0,iocharset=koi8-u,sync,codepage=866 0 0
none /proc proc defaults 0 0
/dev/sda5 swap swap defaults 0 0
```

Путем редактирования этого файла можно добавлять файловые системы, которые теперь будут монтироваться автоматически. Для того чтобы это пра-

вильно сделать, нужно разобраться с форматом файла /etc/fstab. Каждая файловая система представлена в нем отдельной строкой.

Содержимое каждой строки состоит из полей, разделенных пробелами:

- первым в строке идет имя устройства, которое мы монтируем;
- затем идет точка монтирования;
- затем идет тип файловой системы, используемой в этом разделе;
- следующими идут опции монтирования, которые могут определять права доступа, тип кодировки и многое другое.

Для того чтобы монтировать раздел с файловой системой Fat32 (то есть раздел, используемый Windows) к файловой системе Linux, необходимо в файле /etc/fstab добавить строку вида:

```
/dev/hda1 /mnt/disk_c vfat iocharset=koi8-r,codepage=866,auto,user 0 0
```

Обратным монтированию является действие демонтирования. Выполняется демонтирование при помощи команды `unmount`. Если файловая система занята, то есть находится в работе, то демонтировать ее невозможно.

Синтаксис команды `unmont`:

```
umount [-O опции] устройство Точка_монтирования
```

В нашем случае мы имеем диски /dev/sdb1 и /dev/sdb5. Монтировать диски будем к отдельному каталогу, пусть это будет каталог /mnt/sdb1 и /mnt/sdb5. Для этого их нужно создать. По своей сути каталог — это файл, содержащий ссылки на хранящиеся в нем файлы. Читать содержимое каталога может любой процесс, имеющий на это право. Записывать содержимое каталога может только ядро системы. Создается каталог командой `mkdir`. Удаляется командой `rmdir`. Создадим наши каталоги (рис. 3.18).

```
[root@localhost etc]# ls /mnt
cdrom/  floppy/
[root@localhost etc]# mkdir /mnt/sdb1
[root@localhost etc]# mkdir /mnt/sdb5
[root@localhost etc]# ls /mnt
cdrom/  floppy/  sdb1/  sdb5/
[root@localhost etc]#
```

Рис. 3.18. Создание каталогов для дальнейшего монтирования

В приведенном примере мы видим, что перед началом выполнения операции в каталоге /mnt есть два подкаталога /mnt/floppy и /mnt/cdrom. По их названию ясно, что туда монтируется. Мы создаем подкаталоги /mnt/sdb1 и /mnt/sdb5.

Теперь необходимо подправить файл /etc/fstab, для этого в конце файла необходимо добавить две строки:

```
/dev/sdb1 /mnt/sdb1 ext2 defaults 1 2
/dev/sdb5 /mnt/sdb5 ext2 defaults 1 2
```

После внесения изменений в файл /etc/fstab нужно перезагрузить систему, чтобы изменения вступили в силу. Проверить правильность монтирования дисков можно, используя команду `fd`. Эта команда показывает количество свободного дискового пространства в файловых системах, указанных после команды. Если явно не указана файловая система, то показывается свободное место во всех системах, подключенных в данный момент:

```
df [ПАРАМЕТР]... [ФАЙЛ]...
```

Более подробно о синтаксисе этой команды можно узнать, воспользовавшись страницами руководства man.

Как видно из рис. 3.19, монтирование системы прошло нормально.

```
[root@localhost root]# df
Файловая система      Разм  Исп  Дост  Исп% смонтирована на
/dev/scsi/host0/bus0/target0/lun0/part1
                      5,8G  1,8G  3,8G  32% /
/dev/scsi/host0/bus0/target0/lun0/part6
                      3,7G   33M  3,6G   1% /home
/dev/scsi/host0/bus0/target1/lun0/part1
                      2,0G   20K  1,9G   1% /mnt/sdb1
/dev/scsi/host0/bus0/target1/lun0/part5
                      2,1G   20K  2,0G   1% /mnt/sdb5
[root@localhost root]#
```

Рис. 3.19. Проверка правильности монтирования

3.7. Утилита fsck

В принципе файловая система является достаточно надежной и устойчивой. Однако, несмотря на надежность и устойчивость, в ней могут возникать сбои. Причиной может быть аварийная перезагрузка компьютера или перебои в подаче электричества (если у вас нет источника бесперебойного питания).

Аварийное завершение работы или перезагрузка всегда опасны для системы. Смысл в том, что ОС Linux не сразу записывает данные на диск, а буферизует их в оперативной памяти. Поэтому при аварийной перезагрузке данные, которые еще не успели записаться на диск, будут потеряны.

Процесс записи данных на жесткий диск называется синхронизацией. Принудительно синхронизацию можно выполнить, задав команду `sync`. В случае аварийной перезагрузки операция синхронизации дисков не проводится.

При нормальном завершении работы система снимает специальный dirty-бит. Если при загрузке выяснится, что этот бит не снят, то это значит, что завершение работы системы было не нормальным и файловая система может быть повреждена.

Проверка и восстановление файловой системы должны проводиться в демонтированном состоянии. В случае, если это корневая файловая система, то ее монтируют в режиме только для чтения. На первый взгляд, восстановление файловой системы может показаться сложным делом, однако это не так, и в этом вы сейчас убедитесь. Несмотря на наличие утилит по восстановлению файловой системы, всегда лучше иметь резервную копию данных.

Приведем пошаговые инструкции по восстановлению файловой системы.

1. Переходим в однопользовательский режим. Для этого входим с правами суперпользователя и набираем команду:

   ```
   init 1
   ```

2. Следующим шагом демонтируем файловую систему, которую мы будем проверять. Пусть это будет /mnt/sdb1 (файловая система, которая в нашем примере соответствует первичному диску на втором SCSI-диске). Для этого вводим команду:

   ```
   umount /mnt/sdb1
   ```

   В случае, если речь идет о корневой файловой системе, то демонтировать ее невозможно, поэтому перемонтируем ее в режиме только для чтения:

   ```
   mount —n —o remount,ro /
   ```

3. Запускаем проверку файловой системы (рис. 3.20):

   ```
   fsck —f /dev/sdb1
   ```

   В приведенном на рис. 3.19 примере файловая система не содержала ошибок.

```
sh-2.05b# fsck -f /dev/sdb1
fsck 1.34 (25-Jul-2003)
e2fsck 1.34 (25-Jul-2003)
Pass 1: Checking inodes, blocks, and sizes
scsi0: Tagged Queuing now active for Target 1
Pass 2: Checking directory structure
Pass 3: Checking directory connectivity
Pass 4: Checking reference counts
Pass 5: Checking group summary information
/dev/sdb1: 13/257536 files (0.0% non-contiguous), 8102/514072 blocks
sh-2.05b#
```

Рис. 3.20. Работа утилиты fsck

4. Теперь уже проверенные и исправленные файловые системы необходимо смонтировать снова.

   ```
   mount /mnt/sdb1
   ```

Или для корневой файловой системы:

```
mount –n –o remount, rw /
```

5. Возвращаемся к исходному уровню работы, если это был графический режим, то:

```
init 5
```

Если по умолчанию вы запускаетесь в текстовом режиме, то это будет команда:

```
init 3
```

Произойдет смена уровня работы (рис. 3.21).

```
sh-2.05b# init 3
INIT: Switching to runlevel: 3
INIT: Sending processes the TERM signal
INIT: Sending processes the KILL signal
Проверяется наличие новых устройств                          [  OK  ]
Устанавливаются параметры сети:                              [  OK  ]
Поднимается интерфейс loopback:                              [  OK  ]
Поднимается интерфейс eth0:                                  [FAILED]
Запускается portmapper:                                      [  OK  ]
Запускается системный регистратор:                           [  OK  ]
Запускается журнал ядра:                                     [  OK  ]
Запускается partmon:                                         [  OK  ]
Loading sound module (es1371)                                [  OK  ]
Инициализация генератора случайных чисел:                    [  OK  ]
Запускается сервер шрифтов X:                                [  OK  ]
Монтируются остальные файловые системы:                      [  OK  ]
Запускается atd:                                             [  OK  ]
Запускается xinetd:                                          [  OK  ]
Загружается раскладка клавиатуры: ru4                        [  OK  ]
Загружаются клавиши набора: compose.latin.inc                [  OK  ]
Клавиша BackSpace возвращает: ^?                             [  OK  ]
Загружается определение переключения: caps_toggle.inc        [  OK  ]
Запускается numlock:                                         [  OK  ]
Запускается crond:
```

Рис. 3.21. Смена уровня работы с первого на третий

6. В завершение вам придется ввести имя пользователя и пароль.

Ну, вот и все, файловая система смонтирована.

3.8. Утилита hdparm

Одним из немаловажных параметров, характеризующих жесткий диск, является скорость чтения жесткого диска. По своей сути утилита hdparm — это утилита командной строки для настройки параметров жестких дисков типа ATA/IDE. В некоторых случаях для корректной работы утилиты потребуется

обновить ядро (правда, такое бывает редко, если только у вас не совсем древнее ядро).

1. Настройку начнем с изучения текущего состояния жесткого диска. Для этого можно запустить программу с параметром `–t`. В этом случае программа выполняет чтение с устройства и измеряет скорость чтения, после чего выводит данные на экран. Для лучшей точности можно проделать эту операцию 2–3 раза (рис. 3.22).

```
[root@localhost ~]# hdparm -t /dev/hda

/dev/hda:
 Timing buffered disk reads:   14 MB in  3.13 seconds =   4.48 MB/sec
[root@localhost ~]# hdparm /dev/hda
```

Рис. 3.22. Первичный замер параметров

2. Теперь необходимо понять, почему мы получили такие результаты. Для этого можно запустить программу без параметров, просто указав то устройство, о котором нужно вывести информацию.

```
[root@localhost ~]# hdparm /dev/hda

/dev/hda:
 multcount    =  0 (off)
 IO_support   =  0 (default 16-bit)
 unmaskirq    =  0 (off)
 using_dma    =  0 (off)
 keepsettings =  0 (off)
 readonly     =  0 (off)
 readahead    = 256 (on)
 geometry     = 22192/15/63, sectors = 20971520, start = 0
[root@localhost ~]#
```

Рис. 3.23. Анализ параметров винчестера

Дополнительную информацию можно получить, запустив программу с ключом `–I` (рис. 3.24).

На рис. 3.24 мы можем видеть, помимо прочих параметров, список доступных режимов.

3. Как вы можете видеть, режим DMA выключен. Поэтому первым делом необходимо включить режим DMA.

   `hdparm –d1 /dev/had`

```
[root@localhost ~]# hdparm -i /dev/hda

/dev/hda:

 Model=VMware Virtual IDE Hard Drive, FwRev=00000001, SerialNo=00000000000000000
001
 Config={ HardSect NotMFM HdSw>15uSec SpinMotCtl Fixed DTR>5Mbs FmtGapReq }
 RawCHS=16383/15/63, TrkSize=0, SectSize=0, ECCbytes=0
 BuffType=unknown, BuffSize=32kB, MaxMultSect=64, MultSect=off
 CurCHS=17475/15/63, CurSects=15530835, LBA=yes, LBAsects=20971520
 IORDY=on/off, tPIO={min:160,w/IORDY:120}, tDMA={min:120,rec:120}
 PIO modes:  pio0 pio1 pio2 pio3 pio4
 DMA modes:  mdma0 mdma1 mdma2
 UDMA modes: udma0 udma1 *udma2
 AdvancedPM=yes: disabled (255)
 Drive conforms to: ATA/ATAPI-4 T13 1153D revision 17:  1 2 3 4

 * signifies the current active mode
```

Рис. 3.24. Дополнительные параметры винчестера

Теперь необходимо настроить режим DMA. Для этого воспользуемся ключом `–X` (и далее контрольное число). Контрольное число образуется сложением номера режима с постоянной величиной. Эта величина для режимов PIO равна 8, для режимов DMA равна 32 и для режимов UDMA равна 64.

Таким образом, в нашем случае устройство поддерживает udma2, и контрольное число будет 64 + 2 = 66. То есть команда будет иметь вид:

```
hdparm –X66 /dev/hda
```

Будьте осторожны, при установке используйте только доступные для вашего диска режимы.

4. Необходимо также включить 32-разрядный синхронный доступ к диску. Делается это командой

   ```
   hdparm –c3 /dev/hda
   ```

5. Следующим шагом настроим число секторов, которое диск может прочитать за один раз по максимуму. Для этого посмотрим на рис. 3.24 и проанализируем параметры `MaxMultSect` и `MultSect`. Первый показывает максимально допустимое значение, второй — текущее на данный момент. То есть в нашем случае это будет команда

   ```
   hdparm –m64 /dev/had
   ```

Теперь проверим, насколько ускорилась работа винчестера. Для этого снова возьмем и исполним команду (рис. 3.25):

```
hdparm –t /dev/had
```

```
[root@localhost ~]# hdparm -t /dev/hda

/dev/hda:
 Timing buffered disk reads:   80 MB in  3.00 seconds =  26.65 MB/sec
```

Рис. 3.25. Как видим, скорость значительно возросла

Поскольку при перезагрузке системы параметры будут потеряны, то подобранные нами экспериментально параметры необходимо добавить в файл /etc/rc.d/rc.local. Помимо этого, необходимо проверить файл /etc/sysconfig/harddisks, в этом файле задаются параметры, общие для всех дисков.

3.9. Технология LVM

Распределение ресурсов дискового пространства является очень важной задачей. Неправильное распространение дискового пространства может привести к тому, что диски либо переполнятся, либо будет пустовать большой объем дискового пространства. Предсказать заранее объем, который потребуется в дальнейшем, достаточно сложно.

Одним из способов решения этой проблемы является использование менеджера логических дисков Local Volume Manager (LVM). С помощью этой технологии можно управлять разделами популярных файловых систем ext2, ext3, xfs и др.

Создать ее можно как при установленной операционной системе, так и в процессе установки, что значительно проще, особенно для начинающих.

Рассмотрим, как можно создать LVM при установке операционной системы на примере Mandrake v10. Создание LVM идет на этапе разметки жесткого диска и создания разделов. Наш компьютер содержит два SCSI-диска. В первый диск мы поместим корневой раздел (/) и раздел /boot, оставшееся место отведем под LVM. Второй диск полностью отведем под LVM.

1. Вначале необходимо создать корневой раздел. Для этого выделим вкладку **sda** (первый SCSI-диск), щелкнем мышью по полоске, иллюстрирующей свободное пространство, и нажмем кнопку **Создать** (рис. 3.26).
2. После этого откроется окно создания раздела, система сама по умолчанию сделает его корневым и поставит оптимальный размер. Вы можете подкорректировать эти данные или оставить их без изменений. Обратите внимание на поле **Точка монтирования**, в нем должно стоять /, это говорит о том, что монтироваться будет корневой раздел.
3. Тип файловой системы можно выбрать ext2 или ext3. После того как все проверено, нажмем кнопку **OK** (рис. 3.27).

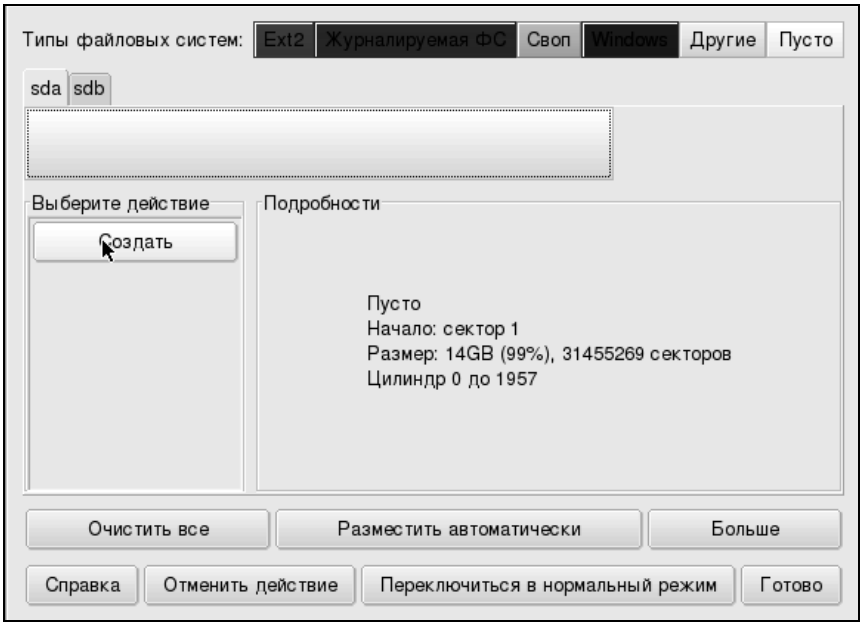

Рис. 3.26. Выбор неразмеченного пространства

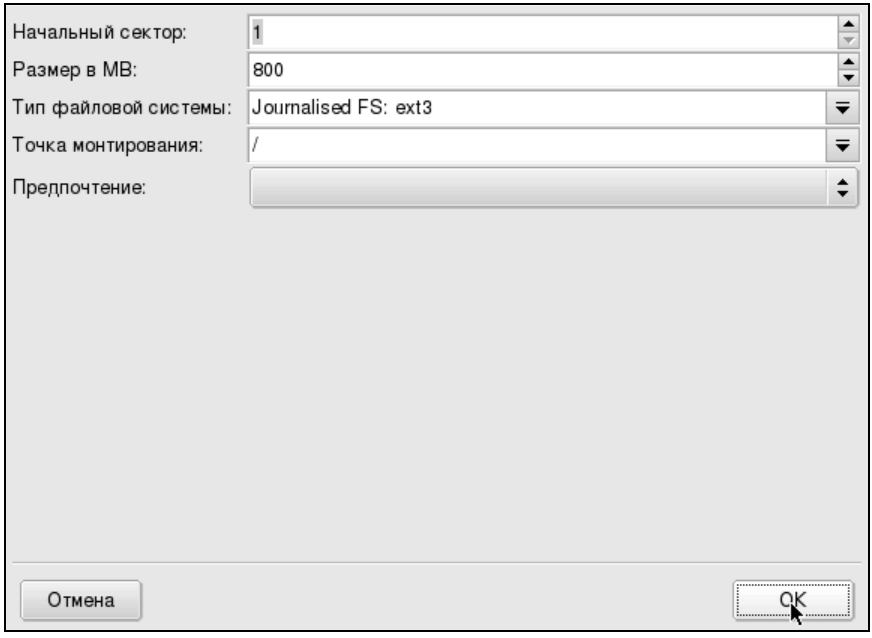

Рис. 3.27. Создание корневого раздела

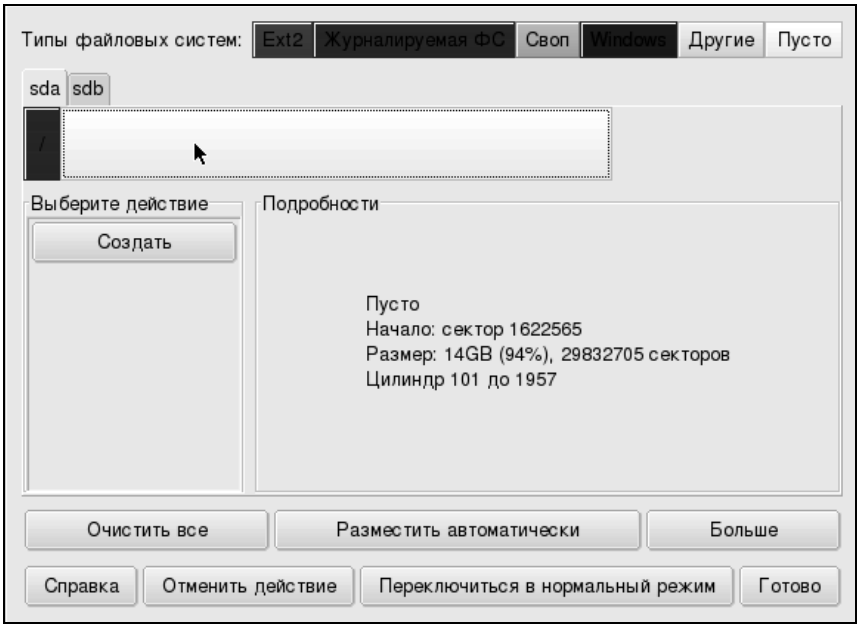

Рис. 3.28. Раздел / создан, создаем раздел /boot

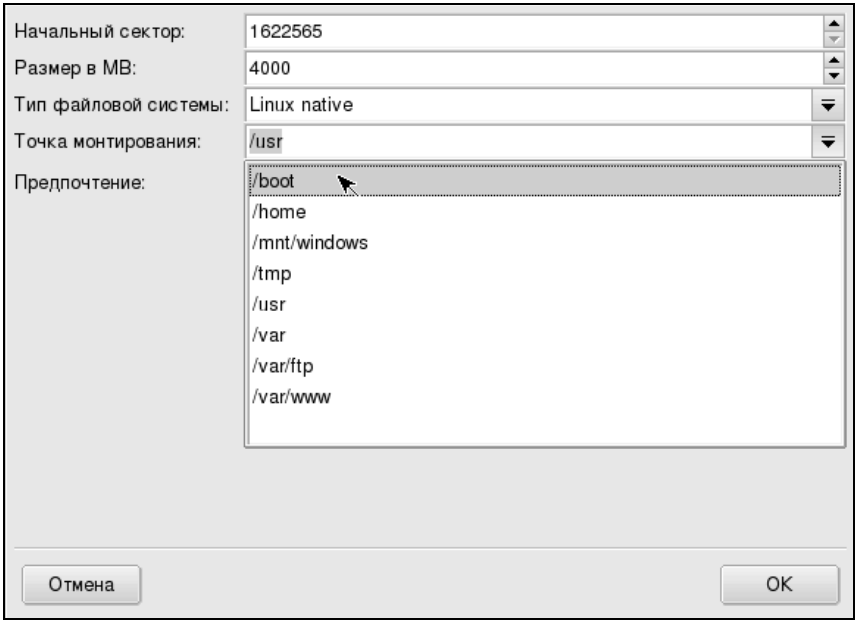

Рис. 3.29. Создание раздела /boot

4. После того как мы нажали **ОК**, вернемся к предыдущему окну создания разделов. Повторим то же самое, с той лишь разницей, что у нас появился новый раздел /.
5. Теперь создадим раздел /boot, для этого снова выделим неразмеченную область и нажмем кнопку **Создать** (рис. 3.28).
6. По нажатию кнопки **Создать** откроется окно создания раздела, в выпадающем списке выберем /boot, при необходимости корректируем емкость раздела и нажимаем кнопку **ОК** (рис. 3.29).
7. Ну, вроде все. Теперь можно перейти к созданию LVM. Для этого на вкладке **sda** щелкнем по свободному пространству, затем по кнопке **Другое** и в последнюю очередь по кнопке **Создать**.
8. Откроется окно выбора типа файловой системы. В окне выберем тип файловой системы **Linux Local Volume Manager partition** и отведем под нее все свободное пространство. Нажимаем кнопку **ОК** (рис. 3.30).

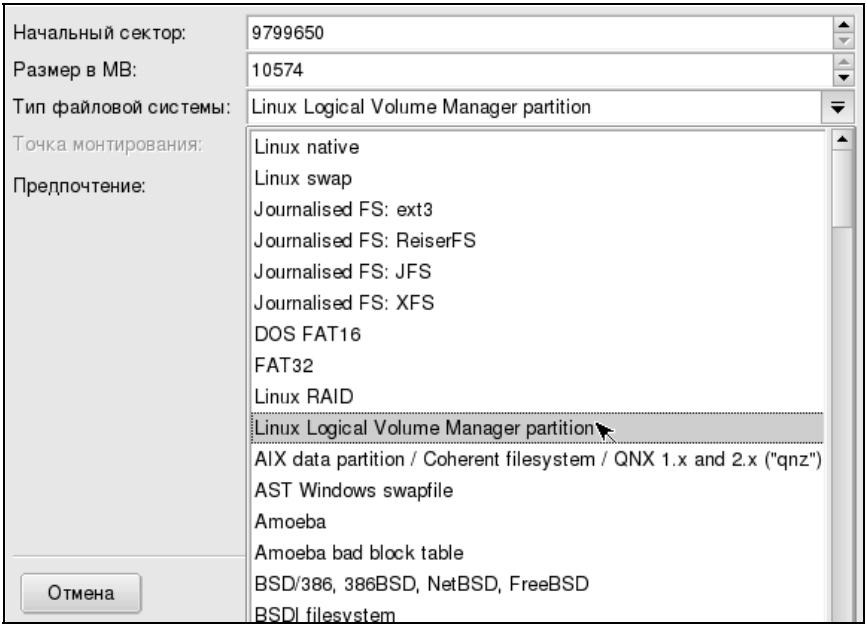

Рис. 3.30. Создание LVM на sda

9. После этого мы снова возвращаемся в окно создания разделов. Теперь необходимо добавить нашу файловую систему в группу томов. Для этого выберем наш LVM-раздел и нажмем **Добавить в LVM** (рис. 3.31). Положительно ответим на вопрос о записи таблицы разделов на диск.

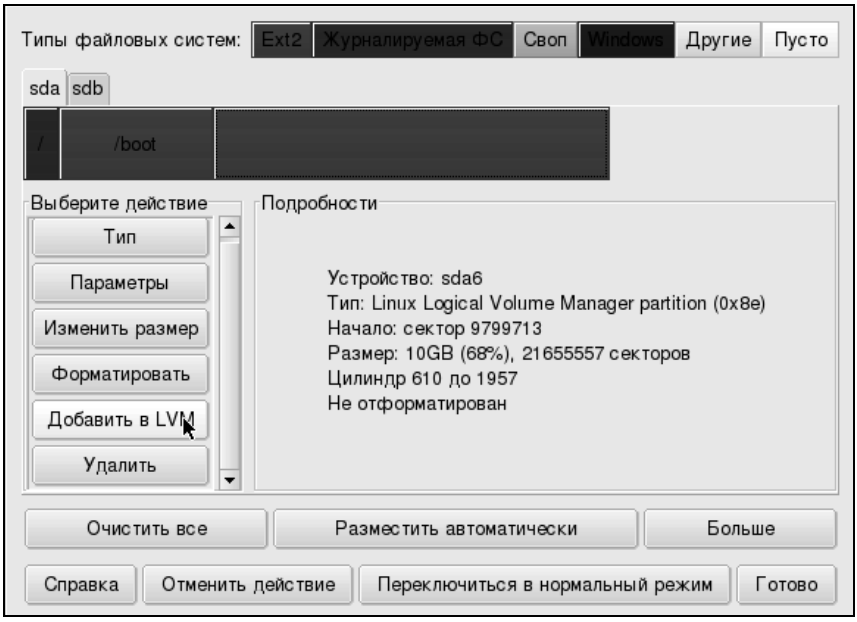

Рис. 3.31. Добавление в LVM

10. На следующем экране будет необходимо ввести имя LVM, пусть это будет: `first_lvm`. Нажимаем кнопку **OK**.
11. После возвращения к окну управления разделами вы обнаружите, что появилась дополнительная вкладка **first_lvm**. Теперь нам необходимо добавить второй диск к созданной нами группе томов.
12. Для этого переходим на вкладку **sdb**, выделим мышью свободное пространство, щелкнем по кнопке **Другое** и по кнопке **Создать**.

    Откроется уже знакомое нам окно создания файловой системы. В разделе **Размер В MB:** зададим все свободное пространство, а в разделе **Тип файловой системы:**

    `Linux Logical Volume Manager partition`

    и нажмем кнопку **OK** (рис. 3.32).
13. После этого мы возвращаемся в окно управления разделами. Находим уже знакомую кнопку **Добавить в LVM** и подтверждаем запись таблицы устройств. На следующем окне у нас появилась возможность добавить уже в существующую или создать новую группу томов, выбираем первый вариант и нажимаем кнопку **OK** (рис. 3.33).

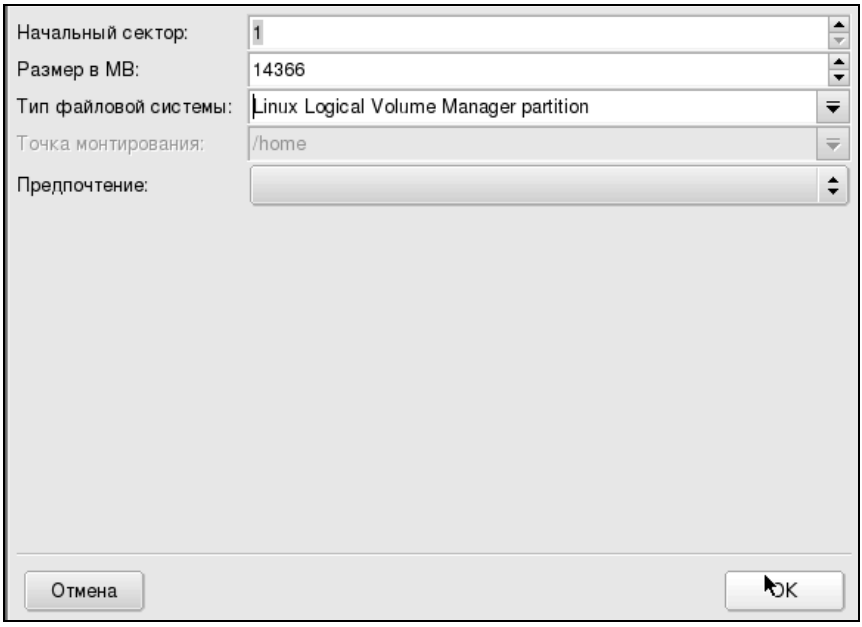

Рис. 3.32. Создание файловой системы

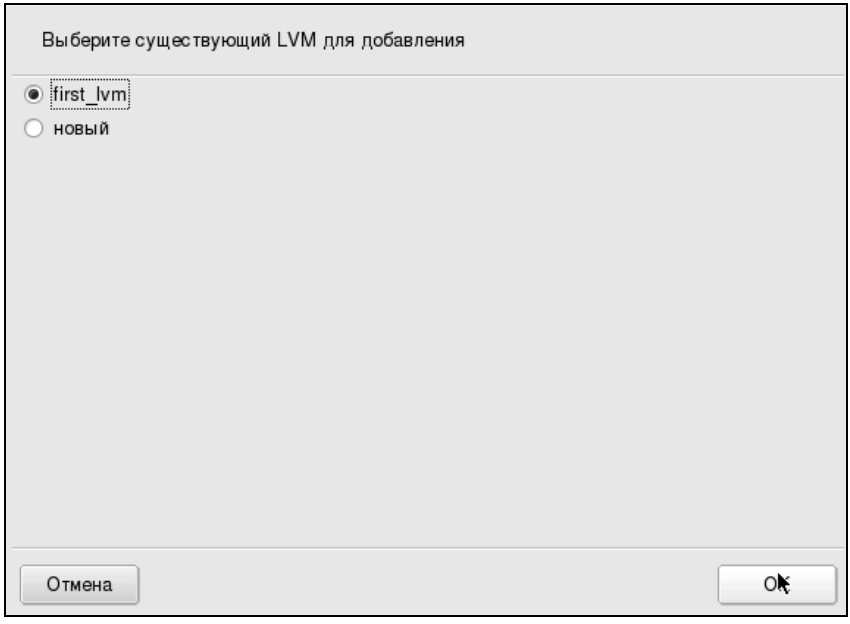

Рис. 3.33. Добавление в группу томов

14. После нажатия кнопки **OK** мы переходим на вкладку **first_lvm** и создаем на этой вкладке недостающие разделы: /usr, /opt, /var, /tmp, /home.

На этом создание группы томов закончено и можно идти по установке дальше. Теперь если вы посмотрите файл /etc/fstab, то увидите, что монтирование происходит следующим образом (рис. 3.34).

```
/dev/first_lvm/2 /usr ext3 defaults 1 2
/dev/first_lvm/3 /var ext3 defaults 1 2
```

Рис. 3.34. Монтирование разделов в /etc/fstab

LVM создана, теперь, при необходимости, можно менять размеры томов.

Глава 4

CD-ROM, DVD-ROM, CD-RW и их настройка в Linux

4.1. Основы технологии оптического хранения данных

Наряду с хранением данных на магнитных носителях (например, винчестерах или дискетах) распространен и другой тип хранения данных — оптические диски. На сегодняшний день технологии оптического хранения данных можно разделить на две большие категории:

- CD;
- DVD.

В принципе, эти две технологи имеют много общего, хотя DVD и имеет более сложную технологическую структуру.

Действие любого устройства оптического хранения данных основано на свойстве материалов, из которых сделана рабочая поверхность диска, отражать или не отражать поток света, идущего от излучателя. Дорожка, несущая информацию, представляет собой комбинацию выпуклых участков и впадинок. Выпуклый участок отражает луч, впадина не отражает луча. Наличие или отсутствие отражения и представляет собой сигнал 0 или 1 (рис. 4.1).

В первой категории можно выделить несколько типов носителей: CD-ROM, CD-R и CD-RW.

CD-ROM (Compact Disk Read-Only Memory) расшифровывается как память на компакт-диске только для чтения. Запись на такие диски производится промышленным способом и не может быть впоследствии изменена. На носителях подобного рода выпускают программное обеспечение, базы данных и другую информацию. Плюсом такой технологии является больший, по сравнению с дискетами, объем информации, явным минусом является то, что отсутствует возможность изменять данные на носителе. Данная технология

получила свое развитие и продолжение, и появились другие форматы оптических дисков CD-R и CD-RW. Благодаря появлению технологии CD-R, появилась возможность однократной записи на диск, а благодаря появлению технологии, появилась возможность не только записывать данные на диск, но и перезаписывать их. То есть появилось полноценное устройство хранения данных. Развитие данной технологии обусловило расширение сферы применения подобных устройств.

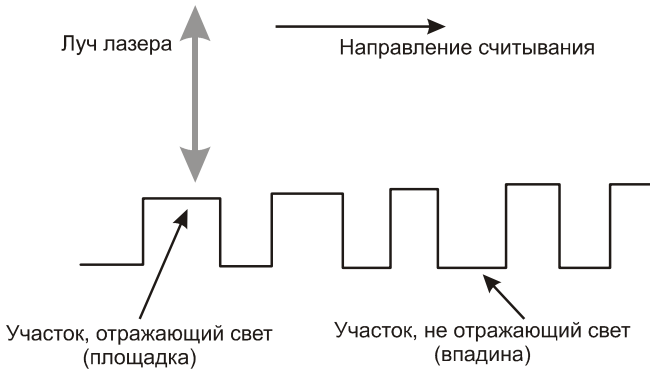

Рис. 4.1. Принцип оптического хранения данных

4.2. Устройство и работа привода

Принцип работы накопителей на оптических дисках основан на считывании при помощи лазерного луча комбинации выпуклостей и впадинок с поверхности компакт-диска. Для того чтобы позиционировать считывающую головку в определенное место на поверхности компакт-диска, используется CD-привод, его схема показана на рис. 4.2.

Логика работы устройства состоит в следующем:

- свет, сгенерированный лазером через систему призм и зеркал, попадает на поверхность компакт-диска;
- в зависимости от того, что в данный момент на поверхности диска, впадина или бугорок, луч лазера либо отражается от поверхности диска, либо не отражается;
- отраженный луч, двигаясь в обратном направлении через систему призм и зеркал, попадает на фотодатчик, который регистрирует отраженный луч и преобразует принятую последовательность световых единиц (наличие луча) и нулей (отсутствие луча) в последовательность электрических сигналов;

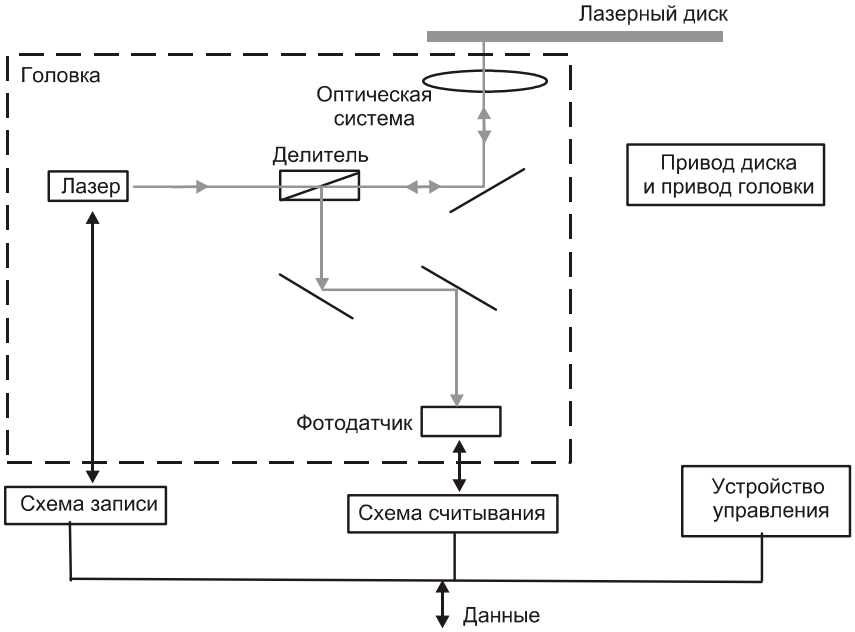

Рис. 4.2. Схема CD-привода

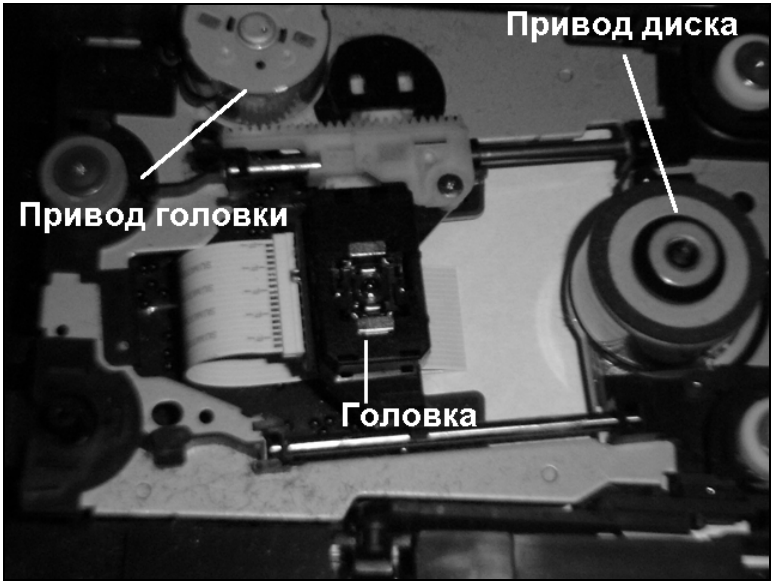

Рис. 4.3. Так выглядит CD-привод

- делитель разделяет луч света, идущий от излучателя на поверхность компакт-диска, и отраженный луч, идущий к фотодатчику;
- система фокусировки (сервоблок фокусировки) фокусирует луч на поверхности компакт-диска;
- позиционирование луча в нужное место осуществляется при помощи двух двигателей. Первый двигатель вращает компакт-диск относительно своей оси. Он служит для доступа к различным участкам в пределах одной дорожки. Называется это устройство сервоблоком вращения диска. Второй двигатель перемещает блок головок перпендикулярно дорожкам на поверхности диска. То есть блок головок двигается вдоль одного из воображаемых радиусов. Служит этот двигатель для выбора дорожки. Называется он сервоблоком дорожки (рис. 4.3).

4.3. Технологии CD-R и CD-RW

Образование участков, не отражающих свет, в технологии CD-R основано на прожигании участков на рабочем слое диска при помощи лазерного луча.

Для того чтобы процесс прожига стал возможным, чистый CD-R диск (или, как его еще называют, болванка) должен быть изготовлен по определенной технологии. В принципе эта технология близка к производству обычных компакт-дисков, но есть и определенные отличия.

В обоих случаях процесс производства диска сводится к прессованию расплавленной поликарбонной массы при помощи формообразующей матрицы. Но если в первом случае идет формирование впадин и бугорков, то в случае с CD-R формируется спиральная бороздка. Границы бороздки определенным образом модулированы по толщине. Эта модуляция несет дополнительную информацию, которая в конечном итоге позволяет распределить информацию на диске и позиционировать луч.

Затем на созданную заготовку наносится рабочий слой, в котором и будет осуществляться прожиг. Далее наносится защитный отражающий слой и лаковое покрытие, защищающее внутренние слои от повреждения (рис. 4.4).

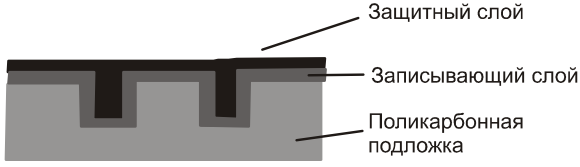

Рис. 4.4. Структура болванки CD-R

Отражающие свойства чистого диска близки к отражающим свойствам площадки, поэтому дорожка чистого диска воспринимается как сплошная отражающая площадка.

При записи на диск модулированный мощный луч лазера нагревает рабочие слои, и краситель выгорает, становясь непрозрачным. То есть, по сути, образуется площадка, не отражающая свет, поскольку тот попросту не доходит до отражающего слоя.

Дальнейшее считывание записанного диска происходит аналогично обычному компакт-диску.

Описанная технология CD-R имеет один существенный недостаток. Процесс прожига красителя необратим, то есть для записи диск можно использовать только один раз, после этого данные остаются там навсегда, и перезаписать их невозможно.

Это не совсем удобно. Поэтому следующим шагом в развитии технологии стало принятие стандарта CD-RW в 1996 г. Преимущество технологии CD-RW состоит в том, что она позволяет не только однократно записывать диски, но и затем стирать их и перезаписывать.

Запись дисков осуществляется точно так же, как и запись CD-R дисков, отличие состоит в том, что образование отражающих и неотражающих участков осуществляется не путем прожига слоя красителя, а путем изменения состояния специального слоя, представляющего собой сплав серебра, индия, сурьмы и теллурия (Ag-In-Sb-Te) (рис. 4.5).

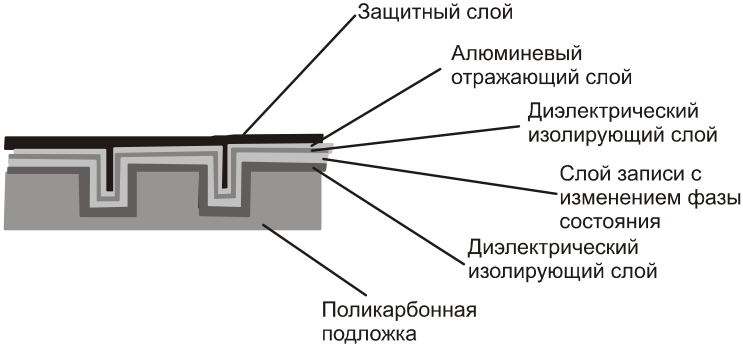

Рис. 4.5. Структура носителя CD-RW

Сплав может менять свою структуру под температурным воздействием. Так при нагреве до 500–700° сплав теряет свою кристаллическую структуру и переходит в аморфное состояние. Это состояние характеризуется меньшей

отражающей способностью. В результате места, где был осуществлен нагрев, воспринимаются как участки, не отражающие свет.

Для перезаписи используется режим нагрева до температуры 200°, при такой температуре не происходит плавления как такового, однако сплав размягчается. Затем происходит медленное охлаждение сплава. При таких манипуляциях сплав переходит из аморфного состояния в кристаллическое.

Однако непосредственно операции стирания как таковой нет. Вместо этого данные перезаписываются напрямую. То есть при перезаписи сразу создаются области, отражающие или не отражающие луч лазера.

К достоинствам CD-RW-дисков можно отнести возможность многократной перезаписи информации. К недостаткам CD-RW можно отнести более низкую скорость записи, более темную поверхность диска и более высокую стоимость.

4.4. Накопители на DVD

Сокращение DVD произошло от Digital Versatile Disk (Цифровой универсальный диск). Некоторые расшифровывают его как Digital Video Disk — цифровой видеодиск, что, строго говоря, неправильно. Диски DVD используют такую же технологию, как и CD. Но благодаря усовершенствованиям в технологии, удалось добиться большей емкости (4,7 Гбайт для однослойного и 8,5 Гбайт для двухслойного). Кроме этого, бывают двухсторонние диски, у которых обе стороны являются рабочими. При производстве диска каждая из сторон штампуется отдельно, затем они объединяются.

При производстве DVD-дисков, как и при производстве обычных компакт-дисков, используются штампованные поликарбонные диски. Однако при производстве DVD-дисков используется более плотная запись, поэтому для считывания диска необходим лазер с более короткой длиной волны. На рис. 4.6 дано представление о разнице в технологиях.

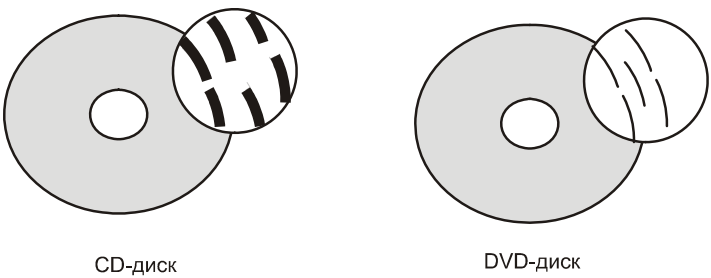

Рис. 4.6. Сравнение технологий записи CD и DVD

4.5. Механическая установка привода CD (DVD)

Первым делом необходимо извлечь привод CD (DVD) из коробки. Варианта тут два: либо у вас технологическая упаковка, то есть просто привод в пакете, либо полная поставка — в этом случае в красочной коробке в комплекте с приводом, как правило, идет IDE-кабель, крепежные винты, руководство пользователя и программный софт.

Так или иначе, какую бы вы ни выбрали упаковку, основное — это привод CD (DVD) (рис. 4.7).

Рис. 4.7. Внешний вид привода CD (DVD)

Для подключения подготовьте отвертку, 8 винтов (по 4 с каждой стороны) с резьбой, совпадающей с резьбой на приводе (на рис. 4.7 эти отверстия показаны стрелками), шлейф IDE.

Может показаться, что здесь все просто, возможно, так оно и есть. В зависимости от конструкции корпуса, модификации материнской платы и геометрических размеров материнской платы может случиться так, что нижние посадочные места будут непригодны для установки. Привод будет просто упираться в материнскую плату. Поэтому лучше ориентируйтесь на верхние гнезда. Далее, идущие в комплекте шлейфы IDE могут оказаться

недостаточно длинными, в этом случае просто приобретите в магазине нужный IDE-шлейф.

1. Для установки привода необходимо снять заглушку корпуса и в освободившееся отверстие с внешней стороны установить привод (рис. 4.8). Привод устанавливается так, чтобы его внешняя панель была вровень с корпусом. Обратите внимание на то, что на рис. 4.8 приведен момент, когда привод еще не до конца утоплен, сделано это для лучшего понимания процесса установки.

Рис. 4.8. Установка привода на посадочное место

2. После установки привода его необходимо закрепить винтами. Для этого потребуется по 4 винта с каждой стороны (рис. 4.9).

Рис. 4.9. Крепеж привода винтами

3. Следующий этап — это подключение привода к материнской плате. Для этого потребуется IDE-кабель (рис. 4.10, *а*) из комплекта материнской платы или специально приобретенный:
   - вначале IDE-кабель подключается к материнской плате (рис. 4.10, *б*). Обратите внимание на совпадение положений ключа на IDE-кабеле и на соответствующем разъеме материнской платы. Дополнительным ориентиром может служить нумерация ножек в разъемах и на кабеле. На кабеле провод, соответствующий первой ноге, маркирован цветом, отличным от цвета кабеля (белый, красный или любой другой);
   - IDE-кабель подключается к приводу, при этом тоже необходимо обратить внимание на ключ. И опять же дополнительным источником информации может служить цветовая маркировка кабеля (рис. 4.10, *в*).

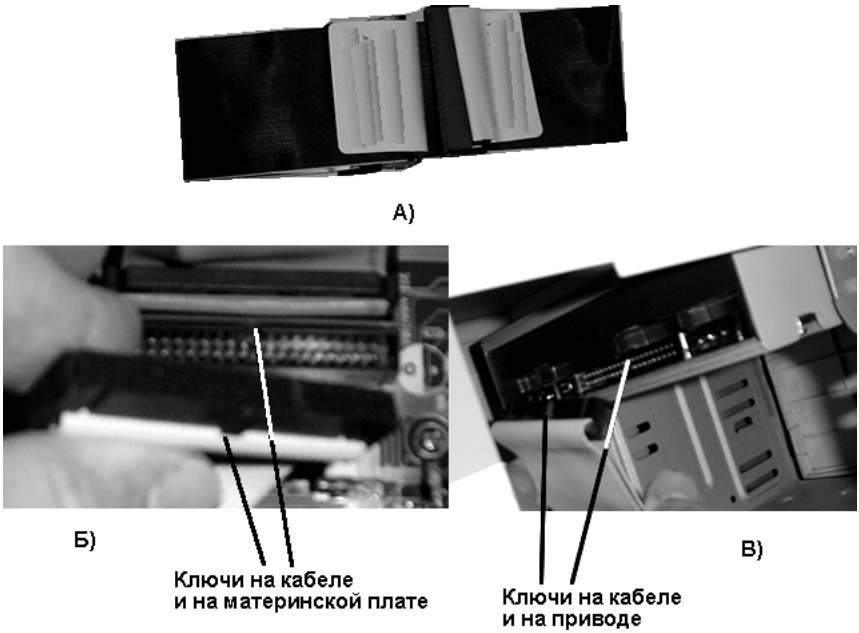

Рис. 4.10. Подключение привода к материнской плате

4. Последняя стадия подключения — это подключение к приводу питания. Обратите внимание на направляющие, которые есть у кабеля питания (рис. 4.11, *а*), такие же есть и во внутренней части разъема питания. Подключение кабеля должно быть произведено в соответствии с этими направляющими (рис. 4.11, *б*).

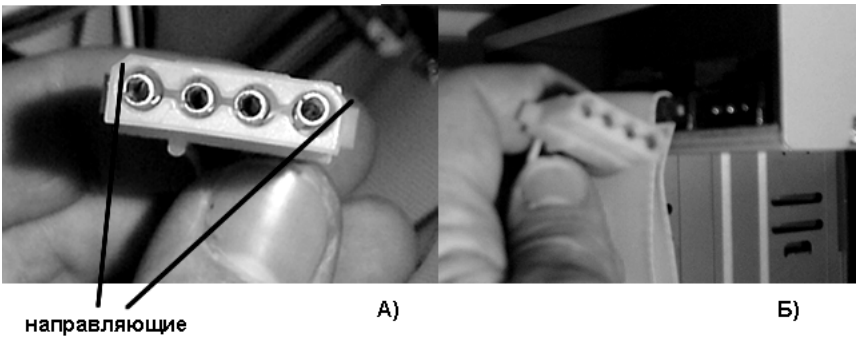

Рис. 4.11. Подключение кабеля питания

4.6. Монтирование привода в ОС Linux

При установке привода, начиная с версии ядра 2.4 и далее, вам ничего не надо делать, устройство будет автоматически найдено и сконфигурировано. Описание правил монтирования любой файловой системы, в том числе и файловой системы приводов CD (DVD), осуществляется в файле /etc/fstab. Фрагмент этого файла после установки приводов приведен на рис. 4.12.

```
/dev/hdc                    /media/cdrecorder       auto
pamconsole,iocharset=cp1251,ro,exec,noauto,managed 0 0
/dev/hdd                    /media/cdrom            auto
pamconsole,iocharset=cp1251,ro,exec,noauto,managed 0 0
```

Рис. 4.12. Фрагмент файла /etc/fstab

В данном случае можно сказать, что в компьютере установлено два привода:

- привод CD-RW мастером (/dev/hdc) на втором канале IDE монтируется в точку /media/cdrecoder;
- обычный CD-ROM зависимым устройством (/dev/hdd) на втором канале IDE монтируется в точку /media/cdrom.

Для обоих устройств установлена опция автоматического монтирования (`auto`), выбрана кодовая страница 1251 (`iocharset=cp1251`), файловая система монтируется с опцией только для чтения (`ro`), разрешено выполнение программ (`exec`).

4.7. Программы для записи CD-RW-дисков

Программы для записи CD-RW-дисков существуют в любом дистрибутиве. Наиболее показательна в этом плане X-CD-Roast. Эта программа по умолчанию присутствует в ASPLinux. Для ее запуска необходимо пройтись по следующему пути: **Программы | Система | CD Writer** (рис. 4.13).

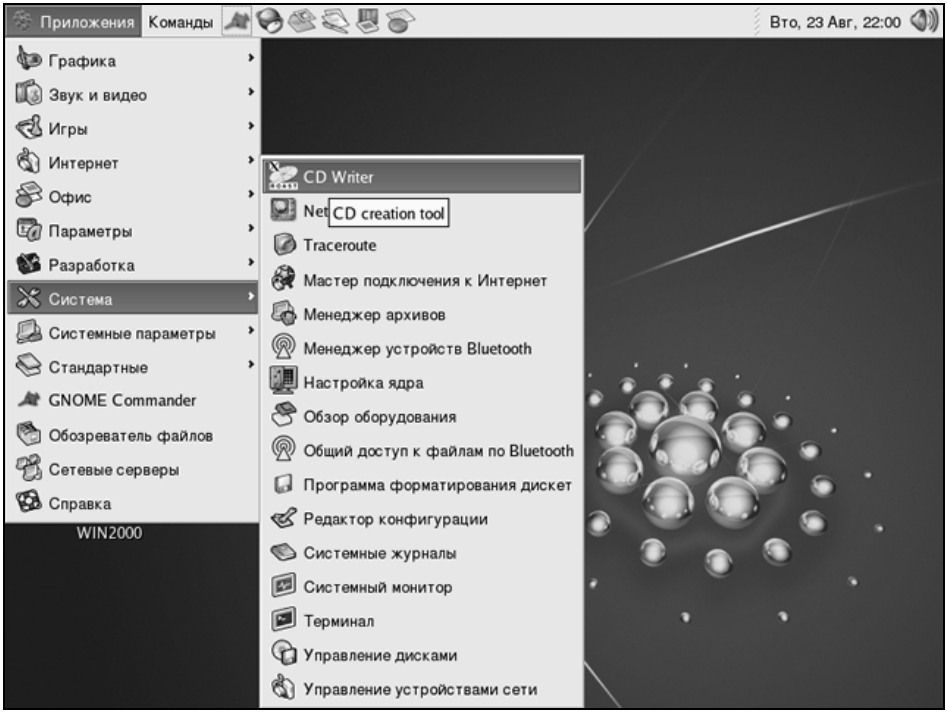

Рис. 4.13. Запуск программы **CD Writer**

После того как вы выберете соответствующий пункт основного меню, запустится программа. Для начала она проведет сканирование присутствующих в системе дисков, о чем будет выдана информация в специальном окне (рис. 4.14):

Принимаем выданные нам данные к сведению и идем далее, для этого нажимаем **ОК**.

Если вы вошли как root, то следующим окном, которое будет вам выдано, будет окно **Режим конфигурации non-root**. В этом окне выбирается, в каком режиме будет работать программа, в режиме root или non-root. Режим non-root является более безопасным, поэтому рекомендуется использовать именно его. Изменить этот режим можно в любой момент, использовав пункт меню **Настроить** (рис. 4.15).

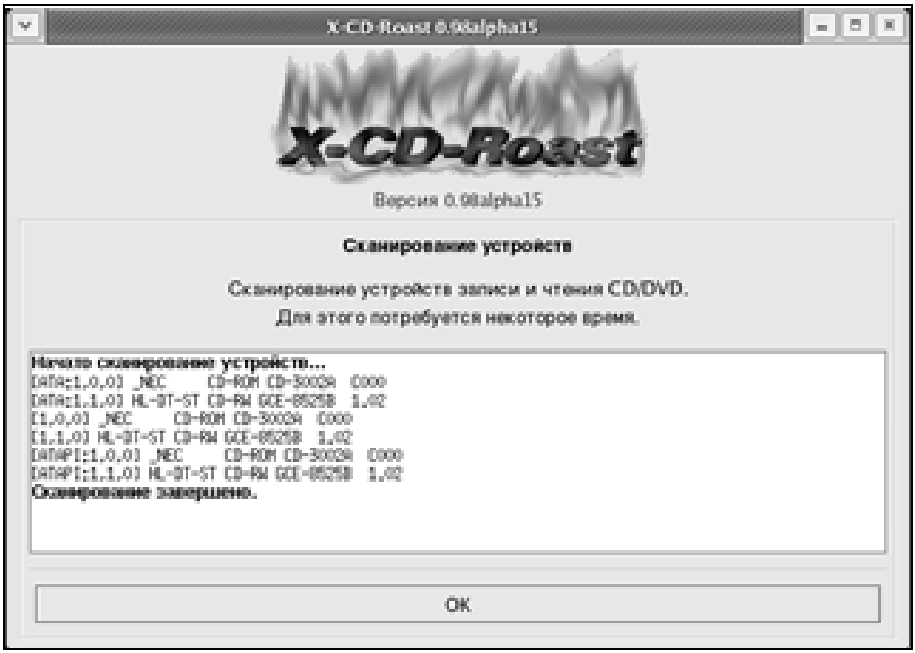

Рис. 4.14. Информация о присутствующих в системе дисках

Рис. 4.15. Режим non-root

В нашем случае лучше оставить режим non-root без изменений. Поэтому нажимаем кнопку **Запустить X-CD-Roast**. Если это первый запуск программы, то появится предупреждающее окно (рис. 4.16).

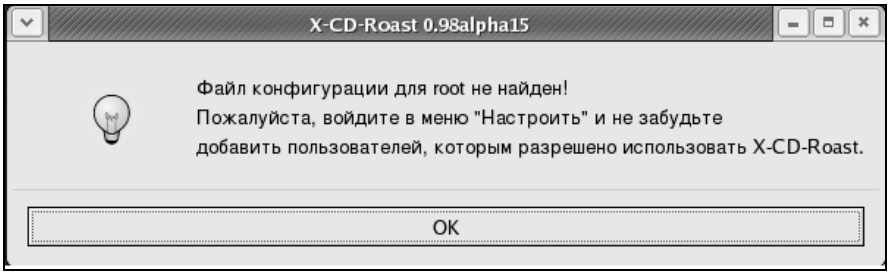

Рис. 4.16. Требуется настройка конфигурации

Нажимаем **ОК** и запоминаем, что необходимо провести предварительную настройку программы. После нажатия кнопки **ОК** появляется основное окно программы (рис. 4.17).

Рис. 4.17. Основное окно программы

Как вы видите, пока активны только две кнопки: **Настроить** и **Выйти**. Так будет до тех пор, пока мы не настроим конфигурацию программы. Поэтому нажимаем кнопку **Настроить**. Открывается окно настроек с вкладками:

- **Список устройств** — выводит список устройств. Здесь можно обновить список устройств, нажав кнопку **Обновить**, или добавить новое устройство, нажав кнопку **Добавить устройство**;
- **CD Приводы** — конфигурируются записывающий и считывающий приводы.

На данном окне остановимся подробнее (рис. 4.18).

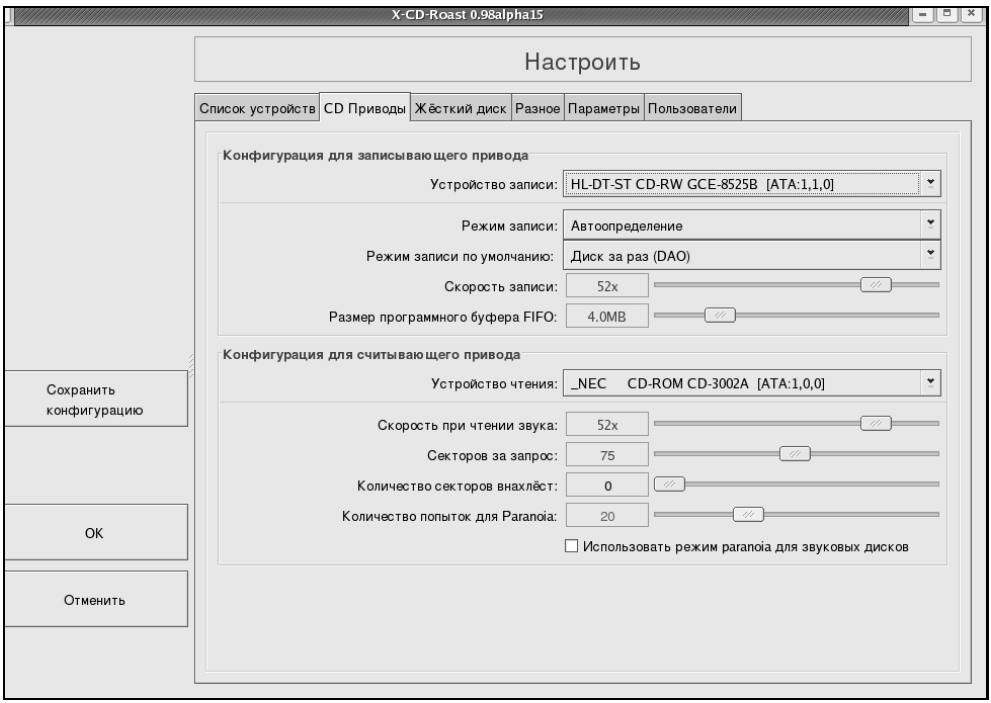

Рис. 4.18. Вкладка **CD Приводы**

На вкладке есть две группы полей: **Конфигурация для записывающего привода** и **Конфигурация для считывающего привода**.

В каждой группе настраивается конфигурация соответствующего устройства. Поля **Устройство записи** и **Устройство чтения** позволяют выбрать устройство, которое будет выступать в соответствующей роли.

- Поле **Режим записи** позволяет выбрать режим записи. Наилучшим вариантом является **Автоопределение**, однако, при необходимости, можно

выставить режим записи и принудительно, например, если при записи возникают какие-либо проблемы. Однако при этом надо знать, поддерживает ли ваше устройство выбранный режим. (Для этого смотрите документацию на устройство.)

- Поле режима записи по умолчанию может принимать одно из следующих значений:
  - **DAO-Disk At Once** (Диск за раз) — предпочтительный режим, однако, может работать не со всеми приводами. Если возникают проблемы, то выбирайте следующий режим;
  - **TAO-Track At Once** (Дорожка за раз) — при записи между дорожками создаются двухсекундные паузы, однако этот режим поддерживается всеми приводами;
  - **Дорожка за раз с нулевой перезаписью** — похож на предыдущий режим, однако отсутствуют паузы между дорожками. Поддерживается только некоторыми приводами Teac и Traxdata;
  - **Запись RAW 96r** — рекомендуется использовать только для продвинутых пользователей;
  - **Запись RAW 96p** — менее желателен, так как не все приводы поддерживают этот режим;
  - **Запись RAW 16** — режим может использоваться для самых дешевых приводов.
- Поле **Скорость записи** говорит само за себя, здесь выставляется скорость, с которой привод будет писать данные на диск.
- Поле **Размер программного буфера FIFO** определяет размер программного буфера, реализуемого для организации буфера данных. Перед записью на диск данные буферизируются, чтобы избежать порчи диска, если поток данных неожиданно прекратится.
- Поле **Скорость при чтении звука** определяет скорость считывания звуковых дорожек. Это поле может быть игнорировано некоторыми приводами.
- Поле **Секторов за запрос** определяет количество звуковых секторов, читаемых при каждом запросе. Чем больше значение, тем больше требуется оперативной памяти.
- Поле **Количество секторов внахлест** разрешает коррекцию против дрожания в дорожках. Чем больше значение, тем медленнее будет идти процесс, 0 — максимальная скорость чтения.

На рис. 4.18 приведены настройки по умолчанию, однако при необходимости их можно заменить.

- **Жесткий диск** — задает каталоги для временного хранения образов. Для добавления каталога в поле путь указываете полный путь, а затем нажмите кнопку **Добавить**. Добавьте в обязательном порядке хотя бы один путь;
- **Разное** — позволяют настроить звуковое устройство, выбрать язык по умолчанию, настроить файл протоколирования и др. Здесь можно все оставить по умолчанию;
- **Параметры** — определяет, как себя будет вести интерфейс программы (рис. 4.19).

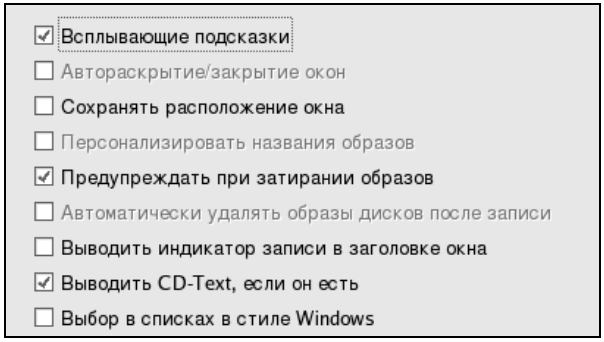

Рис. 4.19. Вкладка **Параметры**

- **Пользователи** — позволяет разрешить или запретить доступ к приводу пользователей. Возможна различная настройка для пользователей и хостов. Доступны следующие опции:
  - разрешить всем;
  - запретить всем;
  - всем из списка;
  - кроме списка.

  Добавить или удалить хост или пользователя из списка возможно при помощи кнопок **Добавить** и **Удалить**.

  Помимо управления пользователями для всех пользователей, которым разрешен доступ, возможно несложное управление правами пользователей (рис. 4.20).

После завершения настроек нажимаем кнопку **Сохранить конфигурацию** (см. рис .4.18), а затем нажимаем кнопку **ОК**.

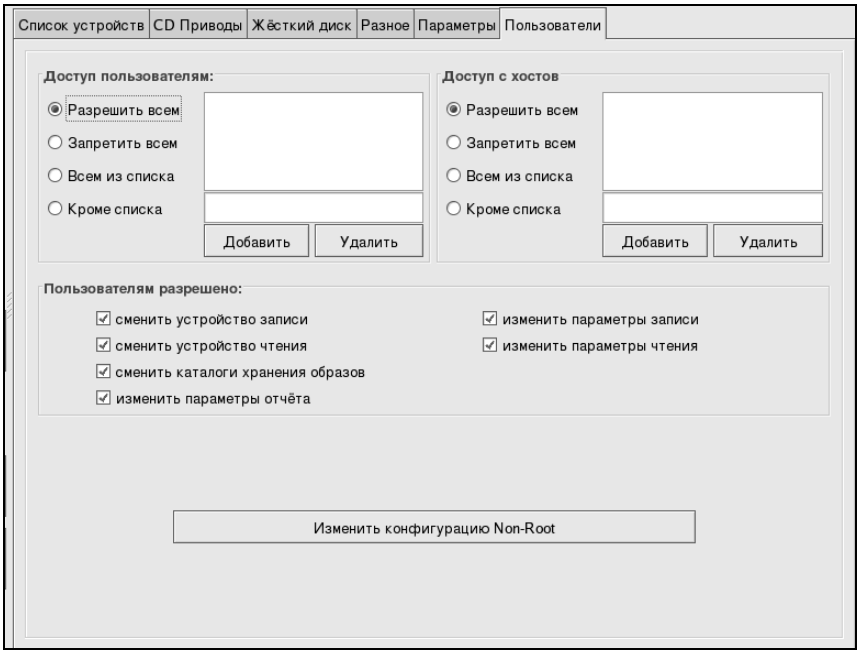

Рис. 4.20. Вкладка **Пользователи**

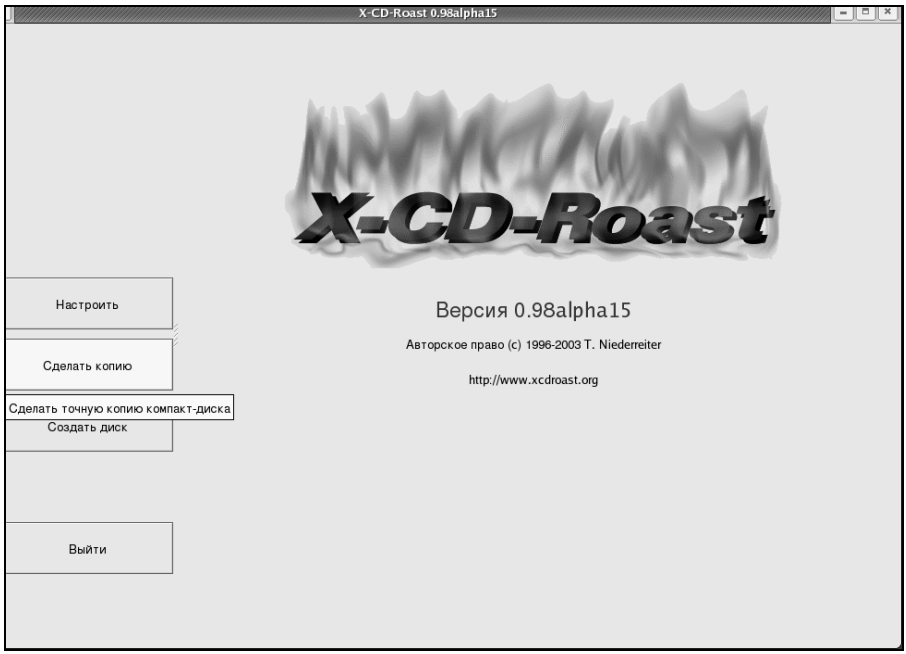

Рис. 4.21. Основное окно. Кнопки теперь доступны

После сохранения конфигурации мы вновь попадаем в основное окно программы, однако теперь кнопки **Создать диск** и **Сделать копию** уже доступны (рис. 4.21).

Рассмотрим, как можно создать новый диск. Для этого нажимаем кнопку **Создать диск** и попадаем в новое окно создания диска (рис. 4.22).

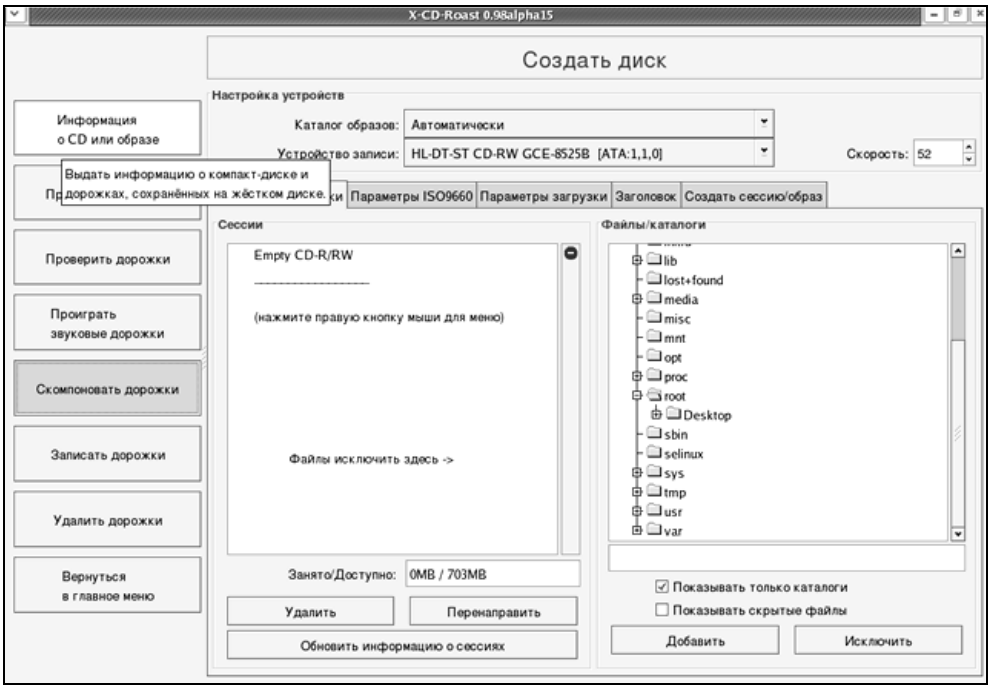

Рис. 4.22. Создание образа диска

В правой части находятся файлы, которые будут записаны на диск. В левой части находятся файлы, находящиеся на жестком диске. Для добавления файлов в образ диска можно либо перетащить их мышкой, либо выделить папку (файл) и нажать кнопку **Добавить** под правым окном. Для того чтобы удалить папку (файл) из образа, необходимо выделить ее в правом окне и нажать кнопку **Удалить**. При этом файл будет удален из образа.

После того как образ диска скомпонован, можно перейти к следующему этапу, а именно к этапу окончательной подготовки и записи диска.

Перейдем на вкладку **Параметры загрузки** (рис. 4.23), здесь можно задать, каким будет диск. В случае если диск будет загрузочным, есть возможность выбора образа загрузки, для этого необходимо воспользоваться кнопкой **Загрузка**.

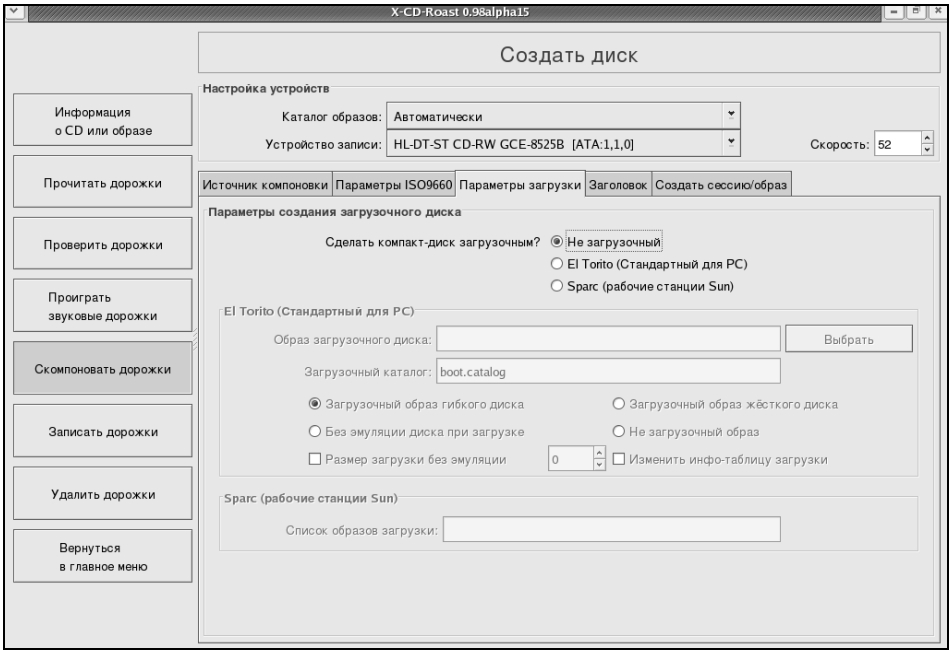

Рис. 4.23. Выбор типа диска

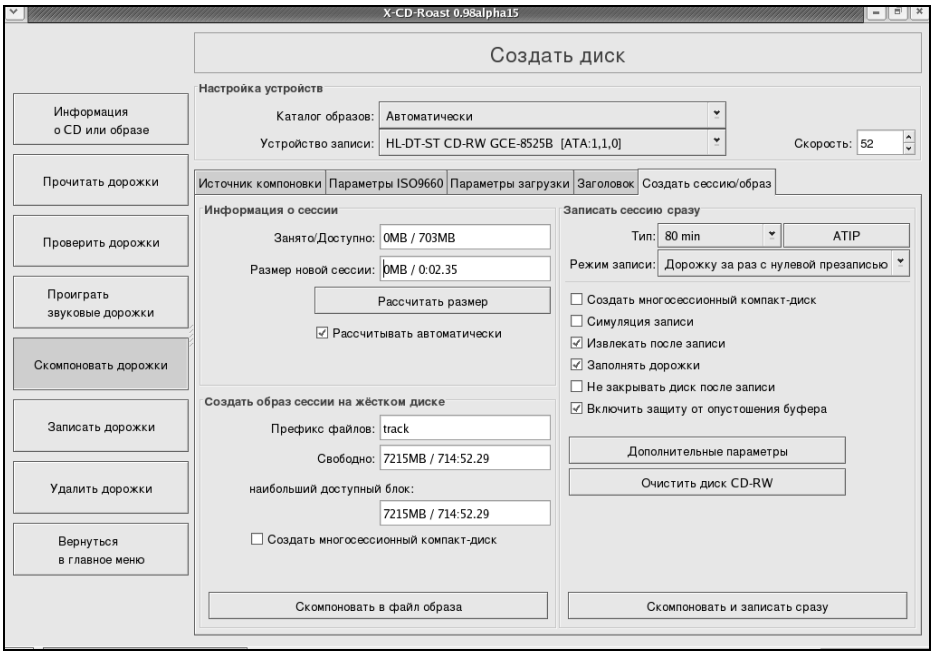

Рис. 4.24. Запись образа диска

На вкладке **Заголовок** можно заполнить информацию о диске:

- **Метка диска**;
- **Кто создал**;
- **Подготовил**;
- **Абстрактная информация**;
- **Библиографическая информация**;
- **Информация об авторских правах**.

После заполнения информации на вкладке **Заголовок** переходим на вкладку **Создать сессию/образ** (рис. 4.24).

Наиболее важны для нас здесь кнопки:

- **Очистить диск CD-RW** позволяет перед записью стереть CD-RW-диск. По нажатию на эту кнопку открывается дополнительное окно **Очистить диск CD-RW** (рис. 4.25).

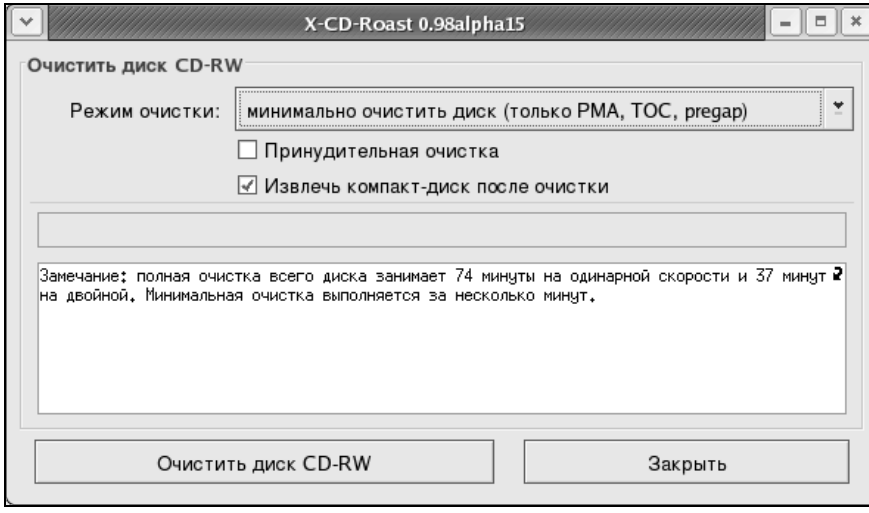

Рис. 4.25. Очистка CD-RW-диска

Здесь вы можете выбрать режим очистки: полная (стирается весь диск) или минимальная (очищается только область с таблицей расположения файлов). И подтвердить очистку, нажав кнопку **Очистить диск CD-RW** или отказаться от очистки, нажав кнопку **Закрыть**.

- Для того чтобы записать диск, нужно нажать кнопку **Скомпоновать и записать сразу**. Появится приглашение вставить диск (рис. 4.26).

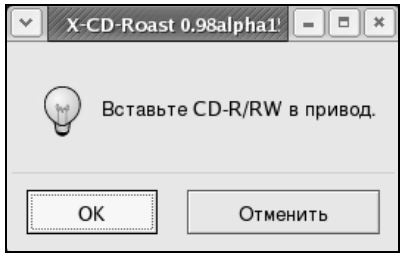

Рис. 4.26. Приглашение вставить диск

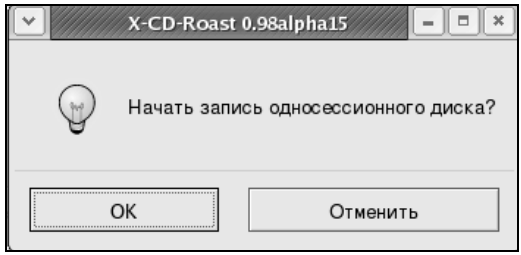

Рис. 4.27. Подтвердите начало записи

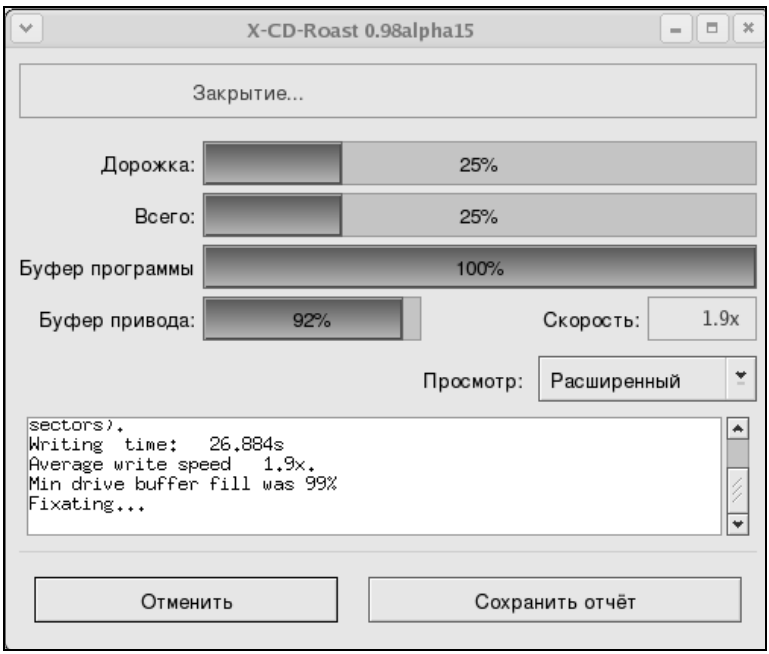

Рис. 4.28. Запись диска

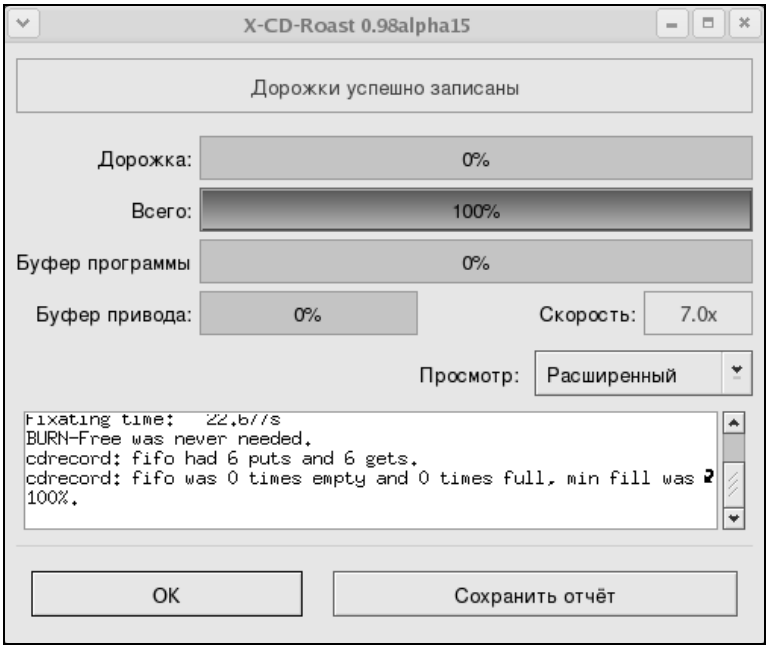

Рис. 4.29. Окончание записи

После того как вы вставите диск и нажмете кнопку **OK**, вам потребуется подтвердить начало записи (рис. 4.27). Начнется запись диска (рис. 4.28).

После окончания записи система выдаст соответствующее предупреждение о том, что файлы записаны на диск (рис. 4.29).

Вам остается только вынуть диск из привода и нажать кнопку **OK**.

Глава 5

Настройка звука в Linux

5.1. Устройство звуковых карт

Самым первым устройством, которое издавало в компьютере звуки, был встроенный динамик. Конструктивно он предназначался для подачи информационных и служебных сигналов в процессе работы компьютера. Естественно, он не предназначался для воспроизведения музыки и звуков. По мере развития программного обеспечения, по большей части игр и мультимедийных приложений, наличие такого устройства, как динамик, перестало устраивать и пользователей, и производителей программного обеспечения.

Появились потребности, а следовательно, появились и предложения. Первым стандартом можно считать Creative Sound Blaster.

Теперь разберемся с тем, что же такое звук и как он формируется в звуковых картах. Звуковые волны представляют собой колебания. Звуковые колебания природного происхождения имеют аналоговый характер, то есть меняются непрерывно. Основными характеристиками для звуковых колебаний является частота амплитуды. Частота — это число колебаний в секунду, на слух изменение частоты сказывается в изменении высоты звука. Чем больше частота, тем выше звук.

Амплитуда звуковых колебаний говорит о громкости звука: чем больше амплитуда звуковых колебаний, тем громче для нас звук (рис. 5.1).

Ухо человека воспринимает очень маленький диапазон частот. Для большинства из нас он укладывается в 20–20000 Гц. Диапазон частот нашего голоса с сохранением узнаваемости можно вогнать в полосу 300–3000 Гц, именно так и поступают на телефонных станциях, ограничивая полосу частот указанным диапазоном.

Теперь вернемся к нашим баранам. Компьютер — цифровое устройство. Это значит, что вся информация внутри него дискретна по своей природе. Дискретные

сигналы отличаются от аналоговых тем, что имеют ограниченное число допустимых состояний. Этих состояний может быть очень много, но их число ограничено. Чем больше этих состояний, тем ближе форма дискретного сигнала приближается к аналоговой. Здесь уместно вспомнить монитор. Допустим, вы смотрите на фотографию, сделанную вашим цифровым фотоаппаратом. (Подразумевается, что фотография хорошего качества.) Так вот, если вы на мониторе установите 16 цветов, то вашу фотографию будет сложно узнать, вряд ли вы на ней что-либо разглядите.

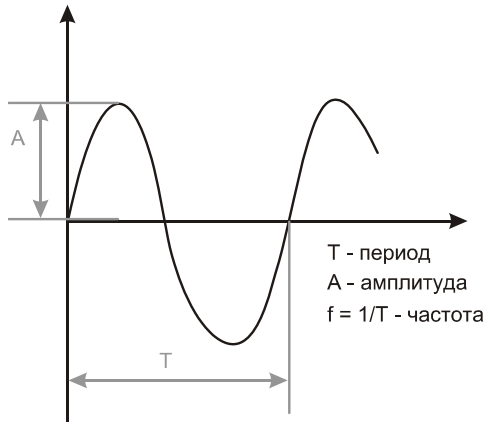

Рис. 5.1. Амплитуда и частота звуковых колебаний

Поставим режим 256 цветов — уже близко к правде, но неправильно передаются полутона, и картинка все же низкого качества.

Поставим разрешение 16 бит, в этом случае картинка выглядит очень реально. Если поставить 32 бит, качество, естественно, возрастет, но на глаз вы этого уже не увидите.

Приведенный пример с монитором призван показать вам, что от качества дискретизации зависит то, насколько сигнал, воспроизводимый цифровым устройством, будет близок к реальному. Вы также должны знать, что при дискретизации необходимо соблюдать отношение качество/затраты, то есть нет надобности задавать дискретизацию сигнала с большей детализацией, чем можно различить на слух.

При вводе аналоговых данных в цифровые (читай дискретные) устройства сигнал оцифровывается (читай, дискретизируется).

При дискретизации сигналов на первое место выходят такие параметры, как частота дискретизации и разрядность. Частота дискретизации определяет частоту, с которой оцифровываемый сигнал оценивается по величине (рис. 5.2).

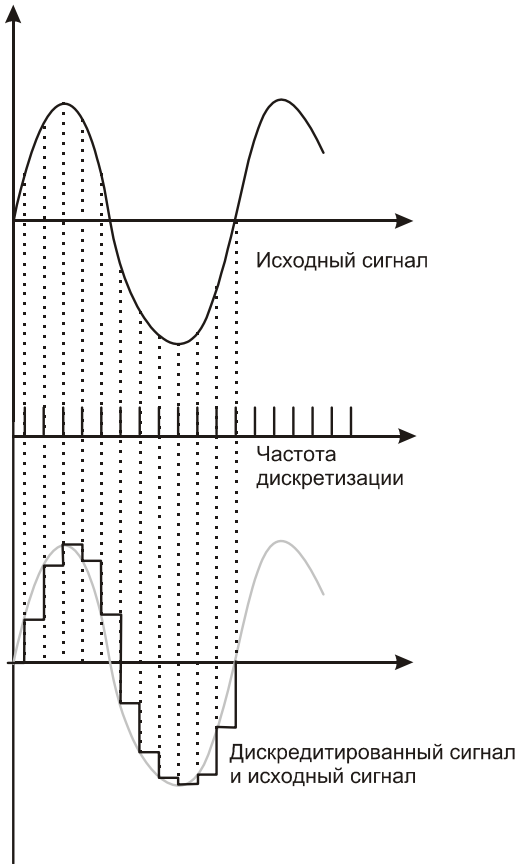

Рис. 5.2. Принцип оцифровки аналоговых сигналов

Разрядность дискретизации в конечном итоге определяет число уровней градаций сигнала. Например, если разрядность дискретизации 8, то таких уровней $2^8 = 256$. Двойка взялась потому, что в основе компьютера лежит двоичная логика, всего два состояния, 0 или 1.

Если разрядность дискретизации 16, то мы уже получим $2^{16} = 65\,536$. Увеличение разрядности дискретизации приводит к тому, что число уровней дискретизации растет в геометрической прогрессии.

При 16-разрядной дискретизации при скачке с одного уровня дискретизации на другой изменение сигнала происходит всего на

$$\frac{1}{65536} \times 100\% = 0{,}00153\%.$$

То есть сигнал достаточно близок к аналоговому.

Современные звуковые карты — это сложные устройства, объединение нескольких устройств в одной микросхеме (рис. 5.3).

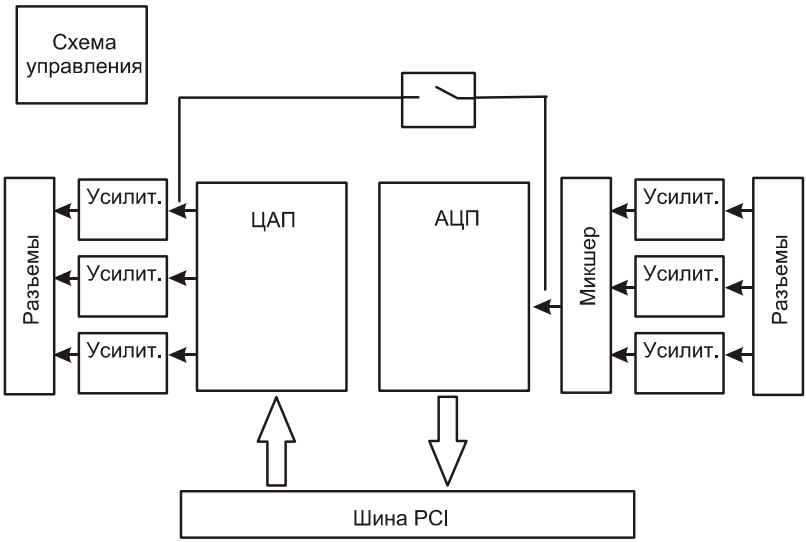

Рис. 5.3. Схематичное устройство звуковой карты

Поясним некоторые обозначения на схеме:

- ЦАП — это цифроаналоговый преобразователь, преобразует данные из цифровой формы в аналоговую. Сигнал через шину PCI поступает на вход ЦАП. Получив информацию в цифровой форме, ЦАП преобразует его в дискретный уровень сигнала на выходе. Вторая порция информации преобразуется во второй дискретный уровень и т. д. В результате на выходе ЦАП образуется ступенчатый сигнал, близкий по форме к аналоговому. Пояснить логику работы ЦАП поможет рис. 5.4.
- АЦП — аналого-цифровое преобразование. Схема осуществляет перевод аналогового сигнала в цифровой методом дискретизации сигналов по частоте и по уровню. Как это делается, было рассмотрено нами ранее.
- Усилители — усиление/ослабление сигнала до необходимого уровня. При необходимости усилители могут быть программно выключены.
- Микшер — осуществляет замешивание сигналов, пришедших с различных входов. На его выходе образуется смесь сигналов, замешанных с различными весовыми коэффициентами. Управление микшером осуществляется

программно через драйвер звуковой карты. Внешний интерфейс управления может быть различным, в зависимости от реализации. Как правило, помимо управления микшером на карту, выводится и управление выходными усилителями (рис. 5.5).

- Схема управления осуществляет управление всеми элементами звуковой карты.

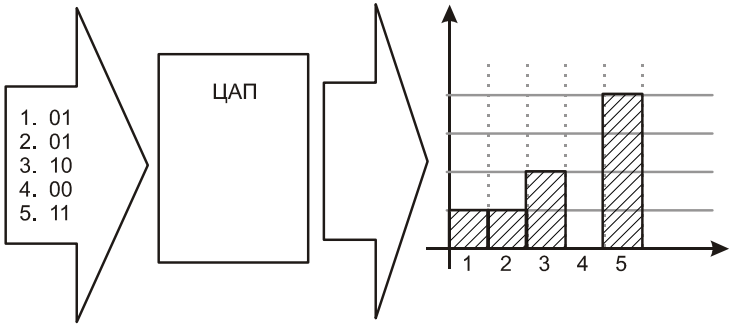

Рис. 5.4. Работа ЦАП при преобразовании сигнала

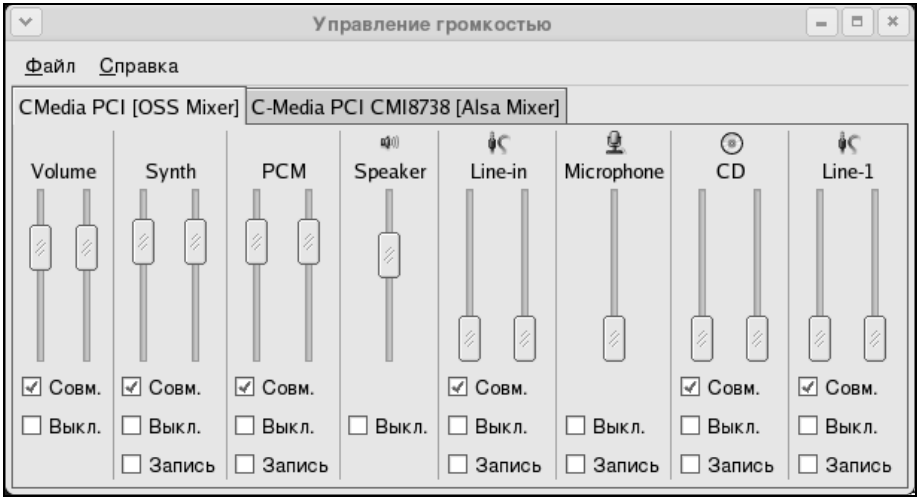

Рис. 5.5. Программа управления звуковой картой

Если внимательнее рассмотреть рис. 5.3, то можно увидеть, что у звуковой карты, есть много входов и выходов (рис. 5.6).

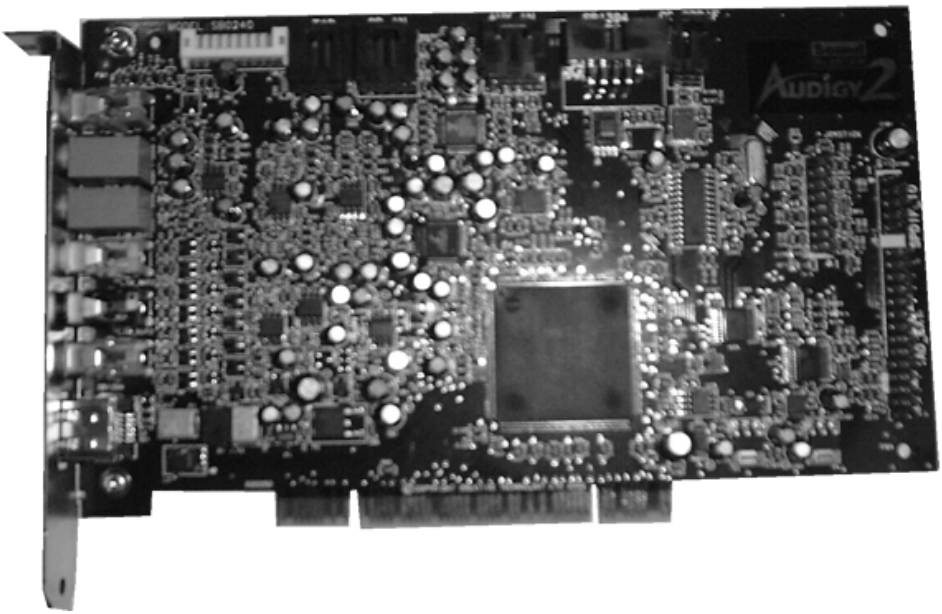

Рис. 5.6. Звуковая карта со множеством входов и выходов

Часть входов/выходов вынесена на внешнюю панель, доступ к ним можно получить с задней стенки корпуса компьютера. Другая часть спрятана на внутренних разъемах, доступ к ним можно получить, только вскрыв корпус компьютера. Все многообразие входов и выходов можно подразделить на типы:

- Линейный выход звуковой карты — это выход звукового сигнала с амплитудой в определенных пределах. Именно с этого выхода сигнал подается на входы активных колонок или усилителей. В современных звуковых картах таких выходов может быть несколько. Например, в системе 5.1 звуковая карта формирует 6 сигналов: передний левый, передний правый, задний левый, задний правый, центральный, сабвуфер.
- Линейный вход — этот вход используется для ввода звуковых сигналов на материнскую плату, их последующего микширования и оцифровки. На внешнюю панель, как правило, выведен только один вход. Однако на внутренних разъемах звуковой карты присутствуют несколько дополнительных входов. В частности, в обязательном порядке есть вход для подключения аудиокабеля от CD-ROM и еще один дополнительный вход (обозначается AUX) и используется для подключения дополнительных устройств, например, для подключения телетюнера.

- Микрофонный вход — название говорит о том, что этот вход предназначен для подключения микрофона. Поскольку сигнал с микрофона ниже сигнала линейного уровня, то этот вход более чувствительный. Кроме того, для нормализации сигнала, как правило, используется схема автоматического регулирования усиления.
- MIDI-порт — этот 15-контактный разъем один в своем роде на звуковой карте. К этому разъему можно подключить джойстик или любую аудиоаппаратуру, имеющую разъем MIDI. Как правило, в большинстве случаев это бывает синтезатор. В профессиональной аппаратуре используется отдельный MIDI-вход, он имеет 5 контактов.
- IEEE-1394 — специально разработанный разъем для подключения источников информации с большим потоком цифровых данных. Это могут быть видеокамеры, жесткие диски или другие устройства. Как правило, такой вход присутствует только в достаточно дорогих устройствах и выносится на заднюю панель.

5.2. Механическая установка звуковой карты

На момент написания книги основная масса звуковых карт имеет разъем PCI (рис. 5.6), поэтому установку рассмотрим именно для этого разъема.

Собственно установку можно разделить на несколько этапов:

1. Подготовительный этап. На этом этапе мы выбираем место, куда будем устанавливать звуковую карту. С точки зрения теплового режима, все устройства, вставляемые в слоты расширения материнской платы, лучше максимально разнести друг от друга. Звуковая карта здесь не является исключением. Наибольшее тепловыделение идет на видеокарте, поэтому звуковую карту лучше опустить как можно ниже, подальше от видеокарты. Если с местом установки определились, то переходим к следующему этапу.
2. Собственно установка звуковой карты. Сначала нужно выломать отверткой (для корпусов подешевле) или отвинтить заглушку (для корпусов подороже) (рис. 5.7).
3. Теперь устанавливаем звуковую карту. Установку осуществляем аккуратно. Необходимо точно совместить разъем звуковой карты с разъемом PCI. Также смотрите, чтобы панель звуковой карты точно вошла в предназначенные для нее пазы. Вставляйте аккуратно, без излишних усилий. Когда все сделано, закрепите звуковую карту винтом (рис. 5.8).

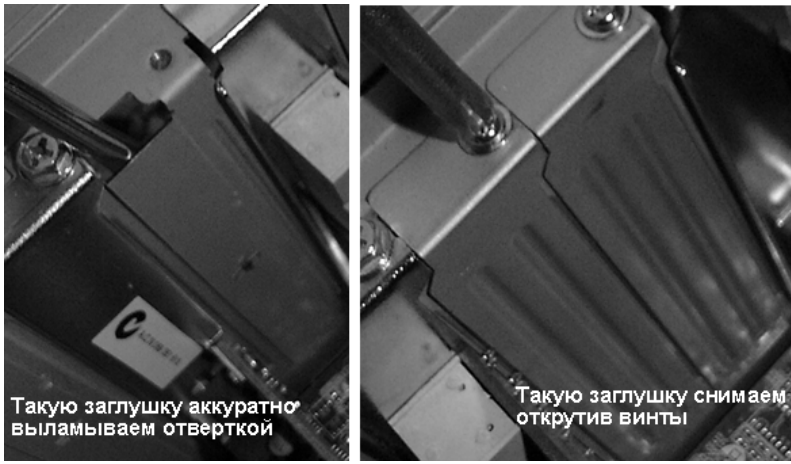

Рис. 5.7. Удаление заглушек

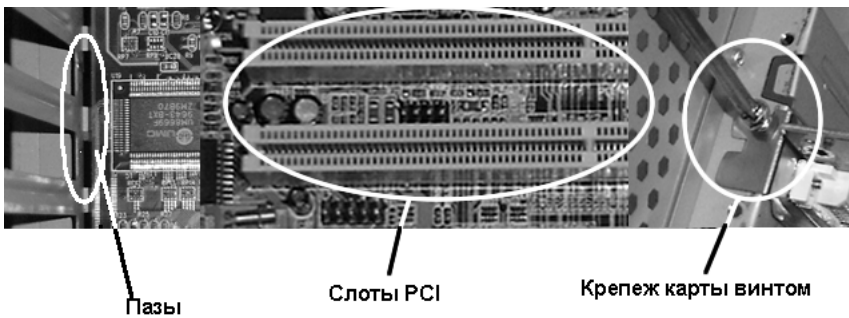

Рис. 5.8. Установка звуковой карты и крепление винтов

Рис. 5.9. Звуковой кабель

4. Последним этапом является подключение аудиовыходов CD-ROM и линейного входа звуковой карты. Для этого используется специальный кабель (рис. 5.9).

   Последовательность подключения кабеля может быть любой. Хотите — вначале подключите к звуковой карте, а затем к CD-ROM, хотите — наоборот. Единственное, на что стоит обратить внимание, так это на правильность подключения. Если вы подключите неправильно, то левый и правый каналы поменяются местами, мелочь, а неприятно. Чтобы этого не произошло, обратите внимание на маркировку левого (L) и правого (R) каналов на разъеме звуковой карты и на разъеме CD-ROM (рис. 5.10).

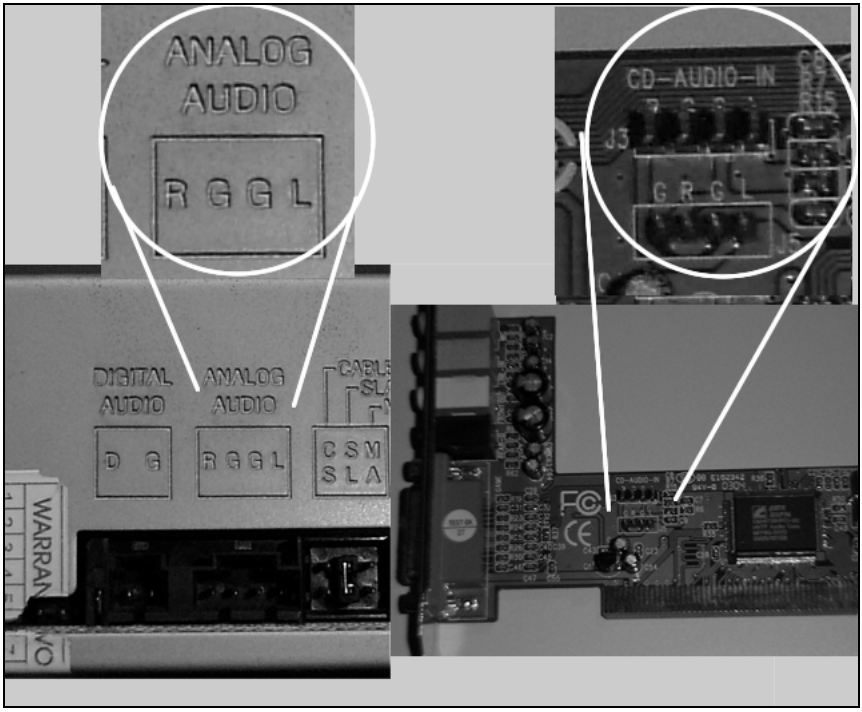

Рис. 5.10. Маркировка левого и правого каналов на CD-ROM и на звуковой карте

5. Если у вас есть TV-тюнер, то подключите его при помощи аудиокабеля к разъему AUX. Здесь, как и при подключении CD-ROM, нужно следить за тем, чтобы не перепутать местами левый и правый каналы.

Теперь, когда все установлено, можно переходить к программной настройке звуковой карты.

5.3. Настройка звуковой карты в ОС Linux

Настройка абсолютного большинства звуковых карт происходит в автоматическом режиме при установке системы. Если вы установили звуковую карту после установки операционной системы, то при загрузке системы будет найдено новое устройство, и для него будет установлен соответствующий драйвер.

При необходимости можно запустить определение звуковой карты и принудительно.

5.3.1. ASPLinux v10

В ASPLinux для этого запускается программа **Обнаружение звуковой платы** (рис. 5.11).

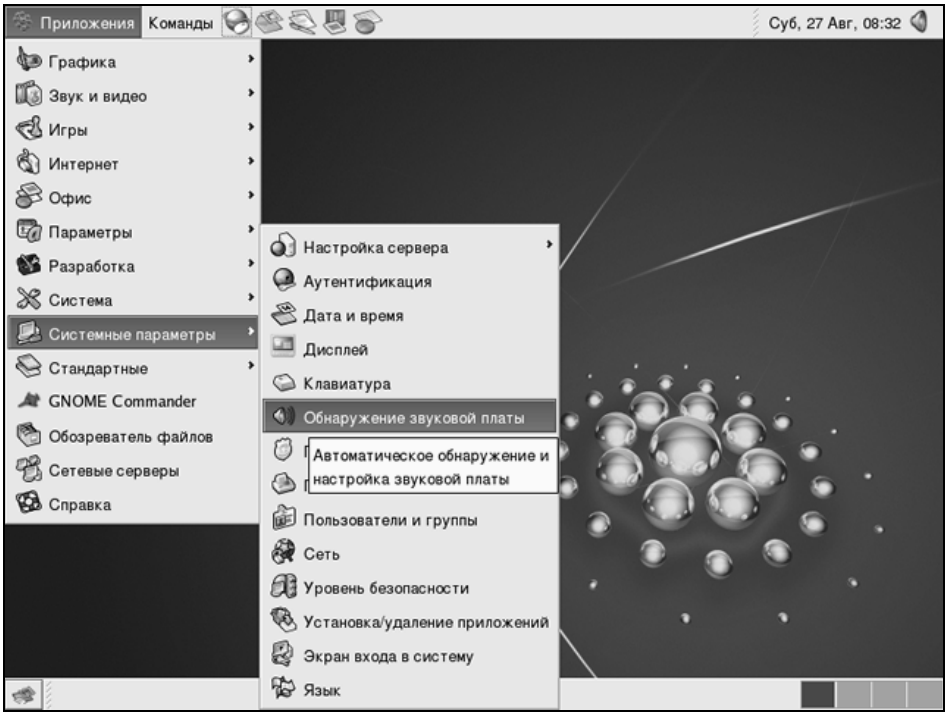

Рис. 5.11. Запуск программы **Обнаружение звуковой платы**

После того как вы нажмете левую кнопку мыши, начнется автоматическое определение звуковой карты, на основании имеющихся у системы сведений. Опрашивая устройство, система подбирает наиболее подходящий драйвер,

устанавливает его. Затем выводится информационное окно (рис. 5.12), в этом окне вы увидите, какое с точки зрения системы у вас находится устройство, и можете проверить правильность настроек, нажав кнопку **Воспроизвести тестовый звук**.

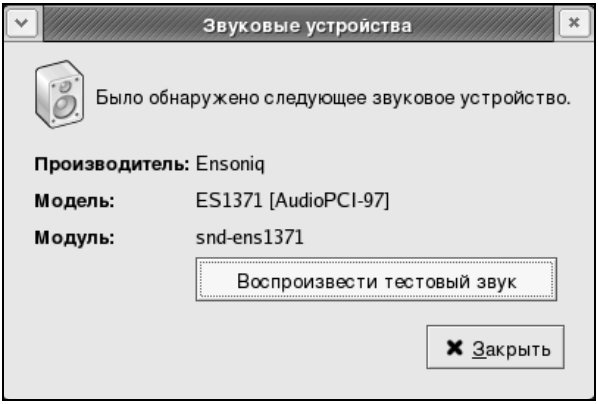

Рис. 5.12. Окно **Звуковые устройства**

Если система правильно подобрала драйверы, то в колонках вы услышите тестовую мелодию. После воспроизведения мелодии, выдается окно с вопросом, слышали ли вы звук (рис. 5.13).

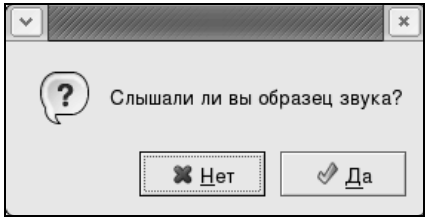

Рис. 5.13. Подтверждение правильности настроек

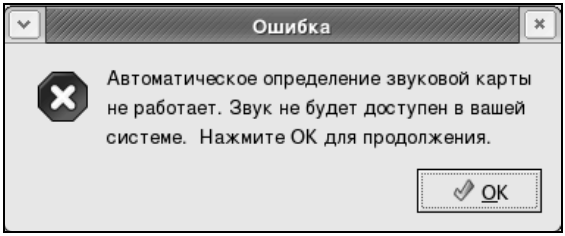

Рис. 5.14. Звуковая плата определена некорректно

Если вы слышали звук, то, естественно, отвечаете — да, в этом случае окно закрывается. И в дальнейшем звуковое устройство будет работать. Если звук вы не слышали, то отвечаете — нет. В этом случае вам выдается информационное сообщение, что устройство работать не будет, и звук вы не услышите (рис. 5.14).

5.3.2. Mandrake v10

В отличие от ASPLinux, где по умолчанию используется графическая оболочка Gnome, в Mandrake по умолчанию используется KDE. KDE — более развитая графическая оболочка (правда, и более тяжелая), поэтому и компоненты по настройке системы здесь серьезнее. Кроме того, Mandrake позиционируется как система, ориентированная на пользователя и простая в настройке. Управление оборудованием осуществляется централизованно, через пункт основного меню **Настройка компьютера** (рис. 5.15).

1. Для его запуска необходимо последовательно выбрать **Система | Настройка | Настройка компьютера**.

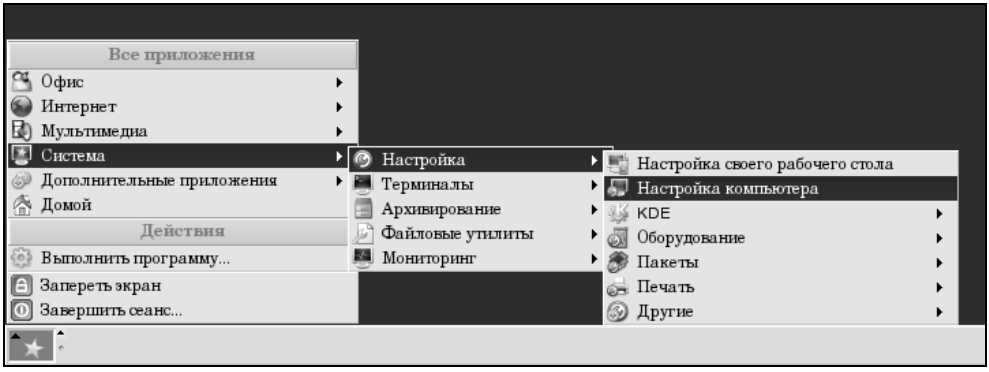

Рис. 5.15. Запуск настройки оборудования

2. После того как вы выберете указанный пункт, перед вами откроется окно **Центр управления Mandrake Linux** (рис. 5.16).
3. Чтобы получить доступ к настройке оборудования, в указанном окне необходимо щелкнуть по значку **Оборудование**. После этого откроется окно управления оборудованием (рис. 5.17).
4. Некоторые устройства представлены в виде отдельных значков. Поскольку нашей звуковой карты здесь нет, то щелкаем пиктограмму **Оборудование**. Открывается список обнаруженного оборудования (рис. 5.18).

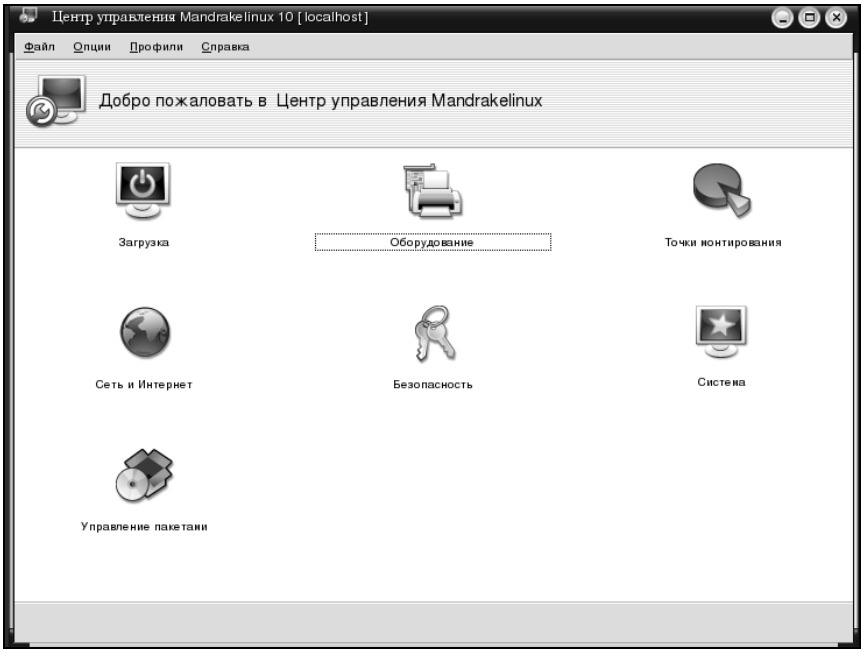

Рис. 5.16. Окно **Центр управления Mandrake Linux**

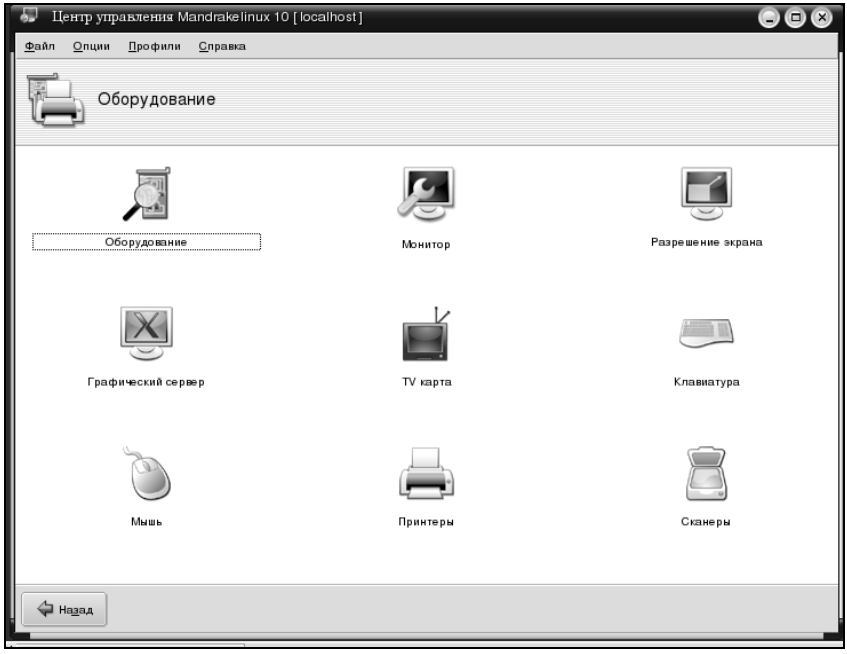

Рис. 5.17. Управление оборудованием

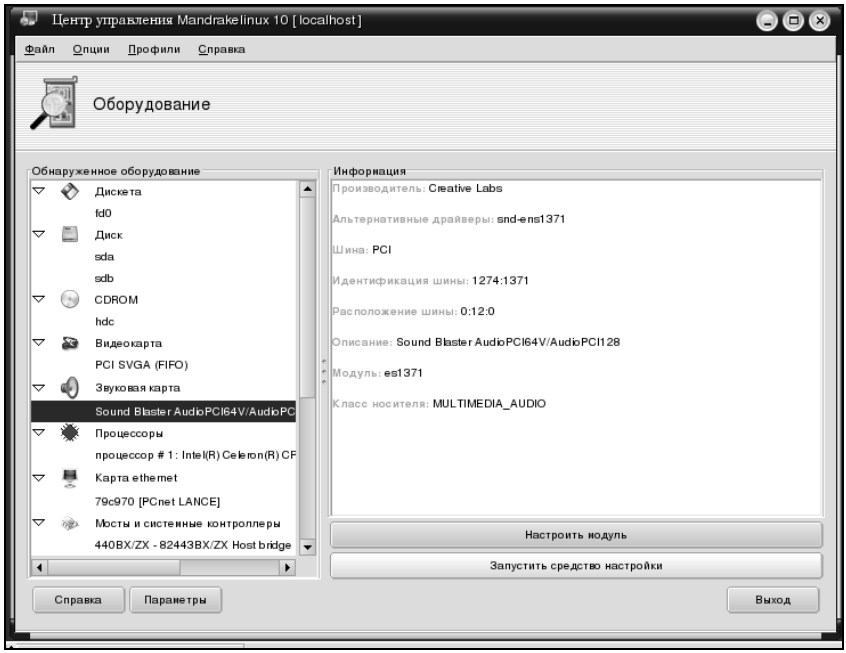

Рис. 5.18. Список обнаруженного оборудования

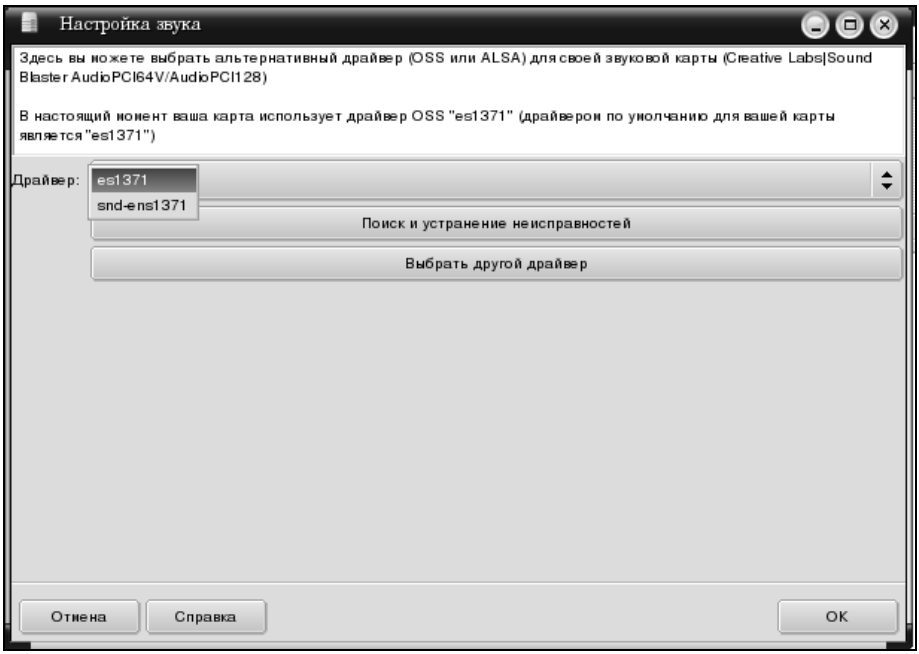

Рис. 5.19. Настройка звука

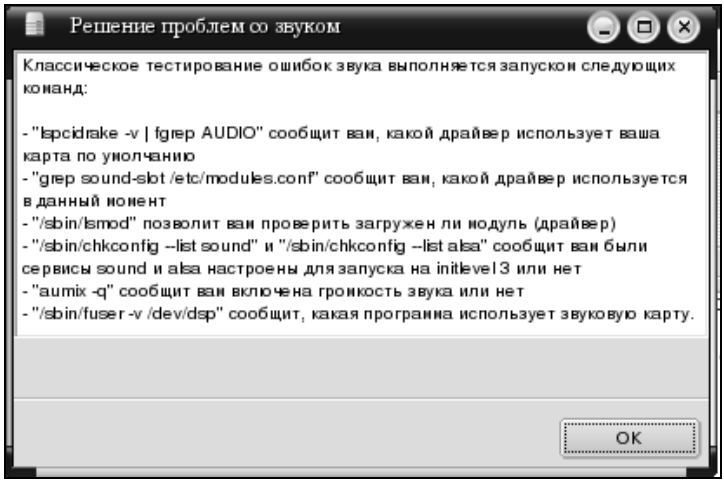

Рис. 5.20. Советы по настройке звука

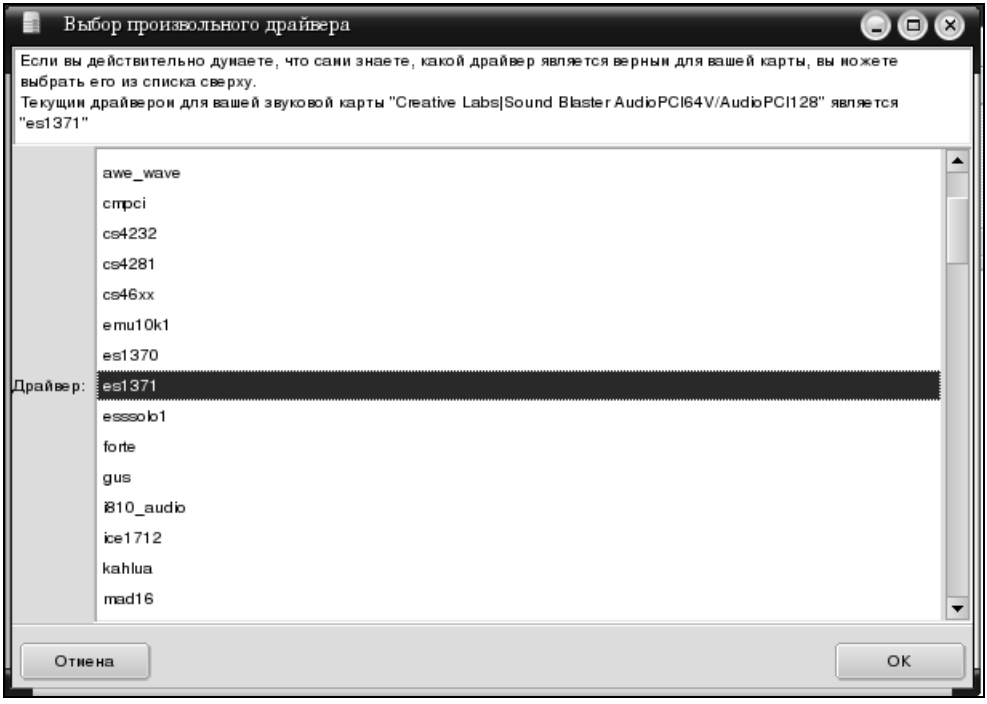

Рис. 5.21. Выбор произвольного драйвера

5. В окне **Обнаруженное оборудование** необходимо мышью выделить звуковую карту. В окне **Информация** вы увидите краткие сведения об обнаруженном оборудовании. Если сведения некорректны или карта не издает звуков, звуковую карту необходимо перенастроить. Для этого нажимаем кнопку **Запустить средство настройки**. Откроется новое диалоговое окно, представленное на рис. 5.19.

   В верхней части этого окна доступны три кнопки. Верхняя из них позволяет заменить драйвер на один из наиболее подходящих, по мнению системы.

   Средняя кнопка **Поиск и устранение неисправностей** открывает обычное информационное окно, в котором даются пошаговые рекомендации по проверке звука. Но эти рекомендации лучше использовать в последнюю очередь, если остальное не помогло (рис. 5.20).

6. Самая нижняя из кнопок открывает для выбора список всех доступных драйверов (рис. 5.21). Выберите из него наиболее подходящий и нажмите **OK**.

5.4. Некоторые программы для работы со звуком в ОС Linux

Спектр программ по воспроизведению и обработке звука в Linux очень широк. Постоянно появляются новые программы, совершенствуются старые. В этом разделе мы рассмотрим лишь некоторые из них, наиболее интересные на взгляд автора.

5.4.1. ASPLinux

Программы для работы со звуком в ASPLinux находятся в разделе **Звук и видео** (рис. 5.22).

Один из пунктов меню называется **Звуковой проигрыватель**, под этим названием скрывается мультимедийный проигрыватель XMMS 1.2.10 (рис. 5.23).

Работа проигрывателя аналогична работе многих подобных программ. Окно мультимедийного плеера состоит из 3 блоков:

- **Audio Player** — собственно окно плеера. Здесь расположено информационное табло, где вы можете увидеть наименование воспроизводимой в данный момент песни, время звучания или время до окончания песни. Используя ползунок 2, можно отрегулировать баланс, используя ползунок 1, можно отрегулировать громкость. Используя ползунок 3, можно регулировать воспроизводимый в данный момент фрагмент песни;

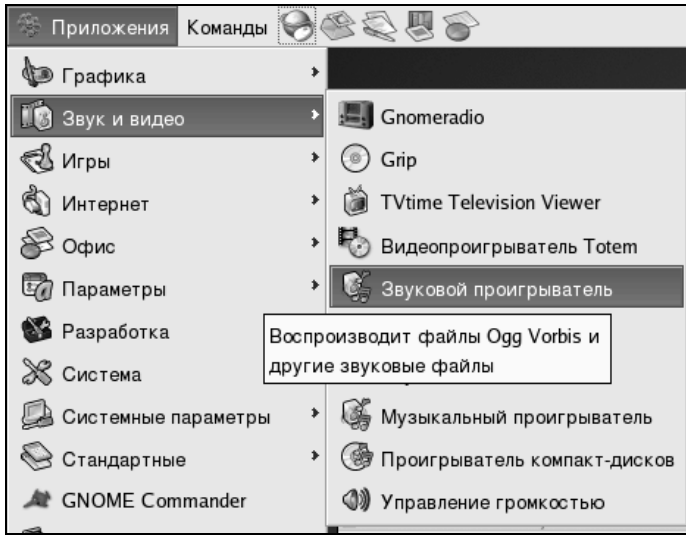

Рис. 5.22. Программы для работы со звуком и видео

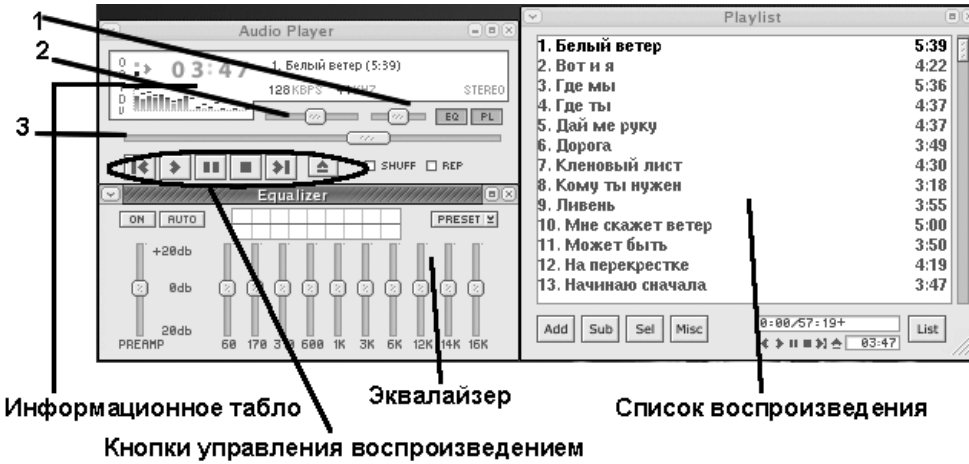

Рис. 5.23. Звуковой проигрыватель XMMS 1.2.10

- **Equalizer** — окно эквалайзера. В этом окне можно отредактировать частотное звучание фонограммы, добавив высоких или низких частот, а также добавить или убрать дополнительное усиление. Допускается регулировка по 10 частотам: 60 Гц, 170 Гц, 300 Гц, 600 Гц, 1КГц, 3 КГц, 6 КГц, 12 КГц, 14 КГц, 16 КГц. Кнопка **Preset** позволяет сохранить или загрузить сохраненные ранее настройки эквалайзера;

- **Playlist** — список песен. В абсолютном большинстве случаев воспроизводится не одна песня, а несколько песен. В окне **Play list** показывается список песен для воспроизведения. Здесь вы можете увидеть наименование песни и ее продолжительность. Используя кнопку **Add**, можно добавить песни к списку воспроизведения. Используя кнопку **List**, можно загрузить или сохранить список песен для воспроизведения. Помимо всего прочего, здесь есть дополнительные кнопки, которые управляют воспроизведением.

5.4.2. Mandrake Linux

По умолчанию в Mandrake список проигрывателей более бедный. Доступ к звуковым проигрывателям осуществляется через пункт меню **Мультимедия | Звук**. Один из интересных проигрывателей — KsCD (рис. 5.24).

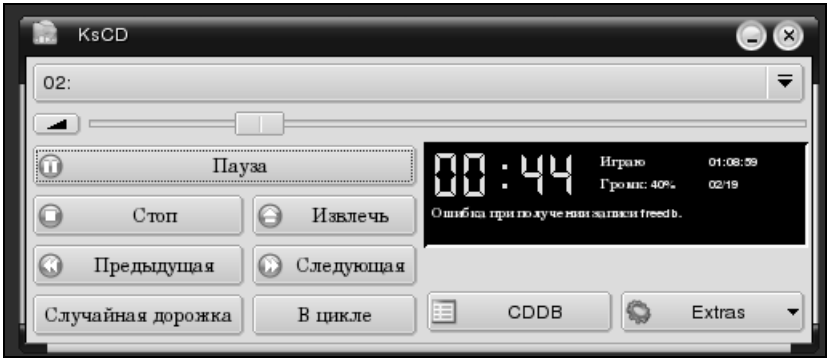

Рис. 5.24. Проигрыватель KsCD

Данный проигрыватель позволяет проигрывать только аудиодиски. На панели программы доступны следующие кнопки:

- **Play / Пауза** — позволяет начать проигрывать компакт-диск или приостановить воспроизведение. Изменение режима воспроизведение (пауза) осуществляется по нажатию кнопки щелчком мыши;
- **Стоп** — останавливает воспроизведение. В дальнейшем при нажатии кнопки **Play** воспроизведение начнется с начала первой дорожки (песни);
- **Извлечь** — по нажатию этой кнопки воспроизведение будет остановлено, а компакт-диск будет выдвинут из привода;
- **Предыдущая** — переход к воспроизведению предыдущей дорожки (песни);

- **Следующая** — переход к воспроизведению следующей дорожки (песни);
- **Случайная дорожка** — воспроизведение дорожек в случайном порядке;
- **В цикле** — после воспроизведения последней дорожки (песни) проигрыватель возвращается к воспроизведению первой песни;
- **CDDB** — при нажатии кнопки откроется окно редактора с информацией о наполнении диска (рис. 5.25);
- **Extras** — расширенное меню, позволяет настроить цветовую гамму, горячие клавиши и др.

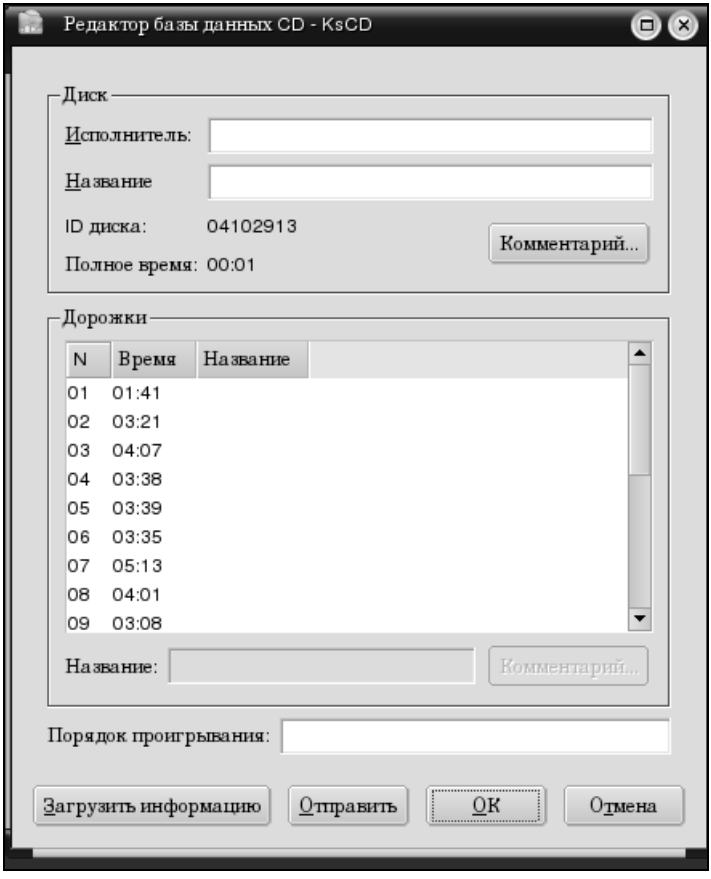

Рис. 5.25. Окно **Редактор базы данных**

Глава 6

Настройка клавиатуры и мыши

6.1. Общие сведения о клавиатуре

Клавиатура представляет собой одно из важнейших устройств ввода (рис. 6.1), которое практически в неизменном виде дошло до наших дней с момента своего создания. Нет, конечно, менялось и число клавиш, и используемые в них механизмы, но общие принципы организации ввода данных с клавиатуры оставались постоянными.

Рис. 6.1. Внешний вид клавиатуры

6.1.1. О клавиатуре и ее подключении

Современная клавиатура, как правило, содержит 104 клавиши, очень редко 101. Подключается клавиатура к компьютеру либо через разъем PS/2, либо через USB. На старых материнских платах может сохраниться пятиштырьковый разъем для подключения клавиатуры (рис. 6.2).

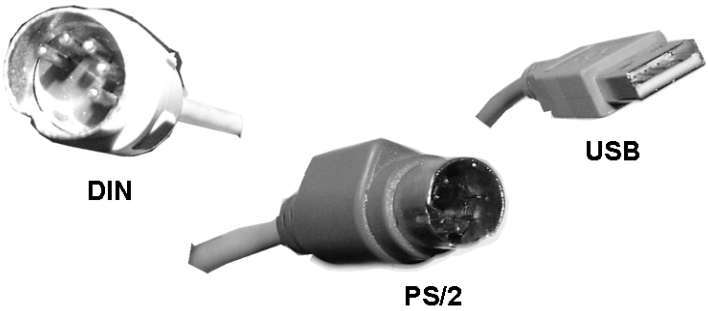

Рис. 6.2. Стандарты подключения клавиатуры

Для того чтобы отличить, куда подключать мышь, а куда клавиатуру, существуют специальные значки. Кроме того, имеется цветовая градация. Разъем для подключения клавиатуры имеет сиреневый цвет, в то время как разъем для подключения мыши имеет зеленый цвет.

При подключении клавиатуры необходимо соблюдать осторожность и следить за правильностью подключения. Для этого обращайте внимание на направляющие, которые присутствуют в разъеме на материнской плате и в клавиатуре.

6.1.2. Конструкция клавиатуры

Если вы помните, то современная клавиатура содержит 104 клавиши, плюс световые индикаторы. Казалось бы, к клавиатуре должен подходить увесистый пучок проводов количеством не менее 100. На самом деле вы этого не видите. Как же так?

Дело в том, что внутри нее спрятан контроллер клавиатуры. Это такая микросхема, которая управляет работой клавиатуры. Она отвечает за обмен данными с компьютером, формирует сигналы на включение или выключение световых индикаторов клавиатуры. Вся информация о нажатых клавишах буферизуется в контроллере клавиатуры и при удобном моменте передается компьютеру (рис. 6.3).

Такая организация работы, когда опросом нажатых клавиш занимается отдельное устройство, обмен с которым идет по инициативе центрального процессора, позволяет разгрузить центральный процессор от несвойственных ему функций.

Матричная схема считывания позволяет сократить число проводников, находящихся внутри клавиатуры. При нажатии одной из клавиш перемыкаются два проводника, контроллер клавиатуры определяет, какие два проводника был перемкнуты, и идентифицирует нажатую клавишу. Номер нажатой клавиши

сохраняется в буфере и затем передается центральному процессору. При этом контроллер определяет как момент нажатия клавиши, так и момент ее отпускания. Благодаря определению момента нажатия и момента отпускания клавиши, вы можете печатать большие буквы, нажав и удерживая клавишу <Shift>.

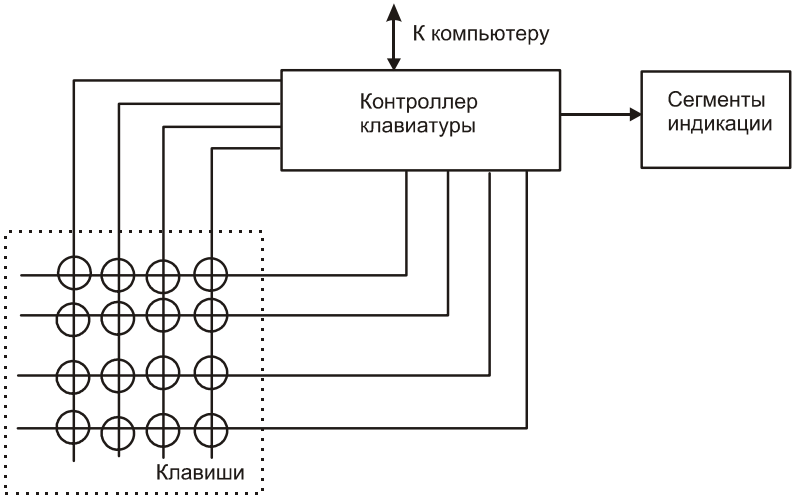

Рис. 6.3. Схема организации клавиатуры

6.2. Настройка клавиатуры в ОС Linux

Как и любая другая операционная система, Linux получает от клавиатуры скан-коды нажатых клавиш и преобразует их в зависимости от выбранной раскладки в соответствующие коды букв.

6.2.1. ASPLinux

Для начала настройки клавиатуры в ASPLinux необходимо пройти по пути **Приложения | Параметры | Клавиатура** (рис. 6.4). При выборе этого пункта откроется диалог настройки параметров клавиатуры (рис. 6.5).

В открывшемся окне доступны четыре вкладки:

- **Клавиатура** — позволяет настроить основные параметры клавиатуры, в частности, такие параметры автоповтора, как задержка и скорость повторения. Для удобства регулировки установка этих параметров оформлена в виде ползунков (рис. 6.5). Также на этой вкладке настраивается скорость мигания курсора. Для удобства настройки параметров на вкладке представлено соответствующее поле;

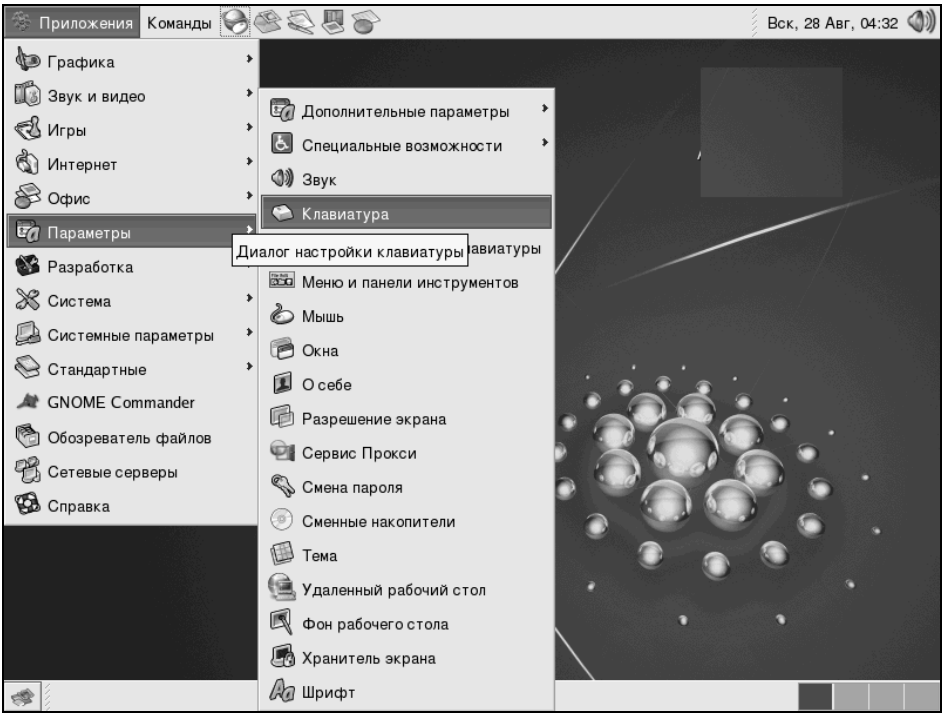

Рис. 6.4. Диалог настройки клавиатуры

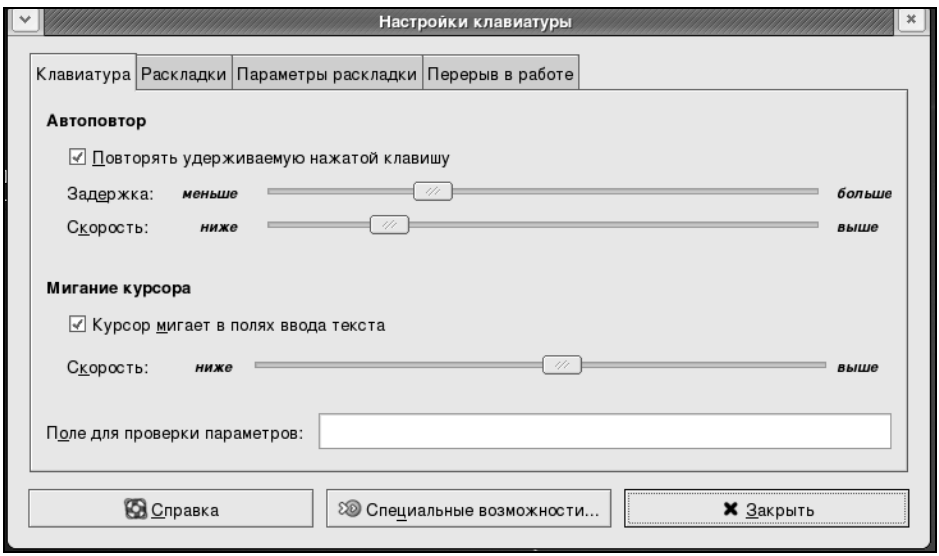

Рис. 6.5. Окно настройки клавиатуры

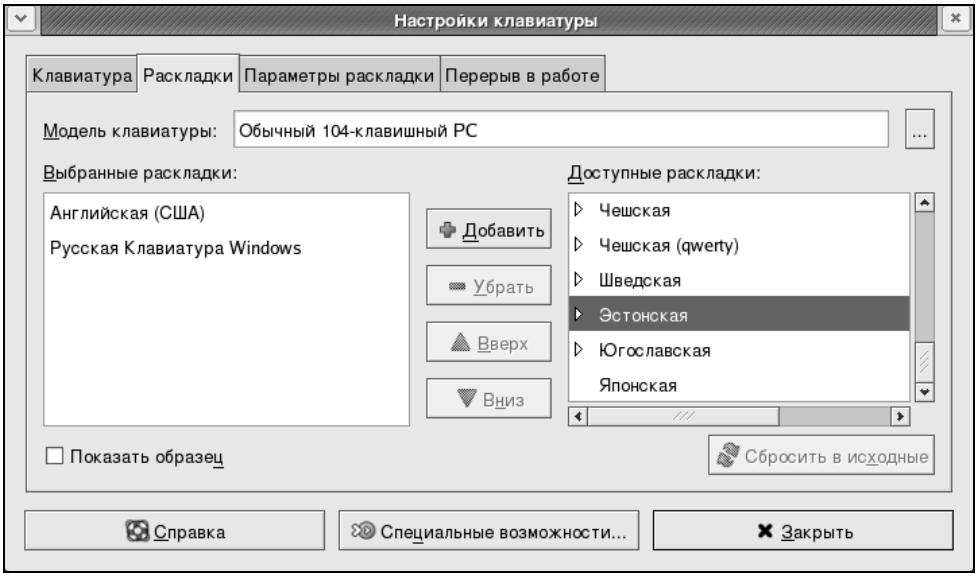

Рис. 6.6. Вкладка **Раскладки**

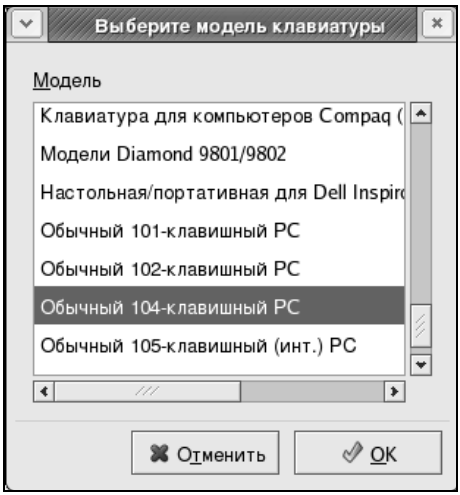

Рис. 6.7. Выбор клавиатуры

- **Раскладки** — позволяет настроить раскладки клавиатуры (рис. 6.6). В окне **Выбранные раскладки** отображены выбранные в данный момент раскладки. В окне **Доступные раскладки** отображаются раскладки, доступные для выбора. Диалоговое окно выбора клавиатуры представлено на рис. 6.7. Для добавления раскладки необходимо выбрать ее в окне

Доступные раскладки и нажать кнопку **Добавить**. Для удаления раскладки необходимо выбрать ее в окне **Выбранные раскладки** и нажать кнопку **Убрать**. Для управления очередностью смены раскладок необходимо выделить соответствующую раскладку и нажатием кнопок переместить ее в соответствующее место;

- **Параметры раскладки** — позволяет настроить более тонкие параметры клавиатуры для переключения между выбранными раскладками, правила использования клавиатурных индикаторов, реакцию на клавиши <Ctrl>, <Alt>, <Caps Lock>, <Win>, параметры совместимости. Логика работы этого окна аналогична предыдущему. Для добавления правила поведения необходимо выбрать его в правом окне и нажать кнопку **Добавить**, для удаления правила из списка необходимо выбрать правило в левом окне и нажать кнопку **Убрать** (рис. 6.8.);
- **Перерыв в работе** — позволяет настроить принудительные перерывы в работе (рис. 6.9). Это может быть полезно для трудоголиков, так как система будет автоматически устраивать вам перерывы.

Помимо настройки клавиатуры как таковой, в ASPLinux можно настроить также и горячие клавиши. Для этого надо щелкнуть мышью пункт меню **Приложения | Параметры | Комбинации клавиш клавиатур** (рис. 6.10). В результате откроется диалог назначения клавиш командам (рис. 6.11).

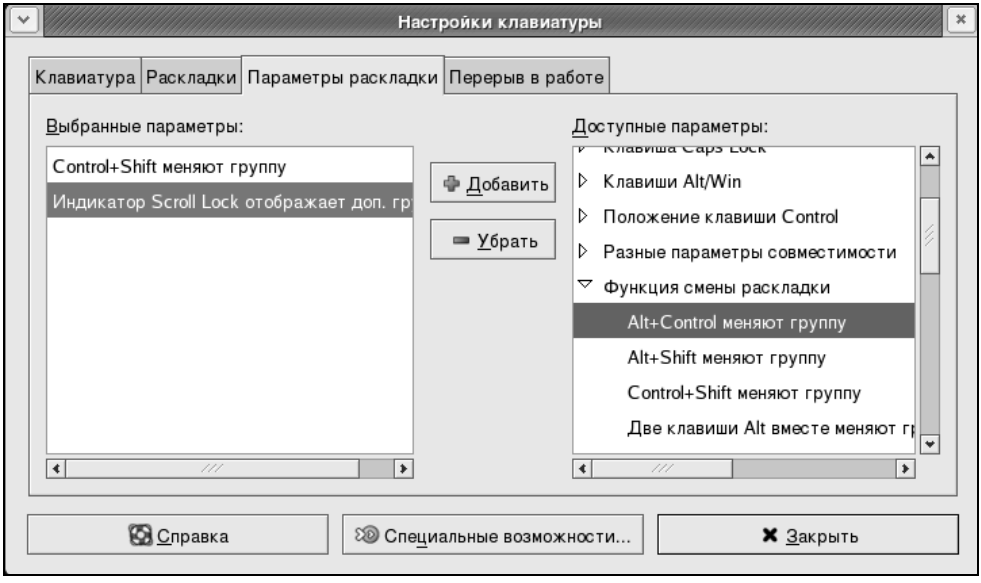

Рис. 6.8. Вкладка **Параметры раскладки**

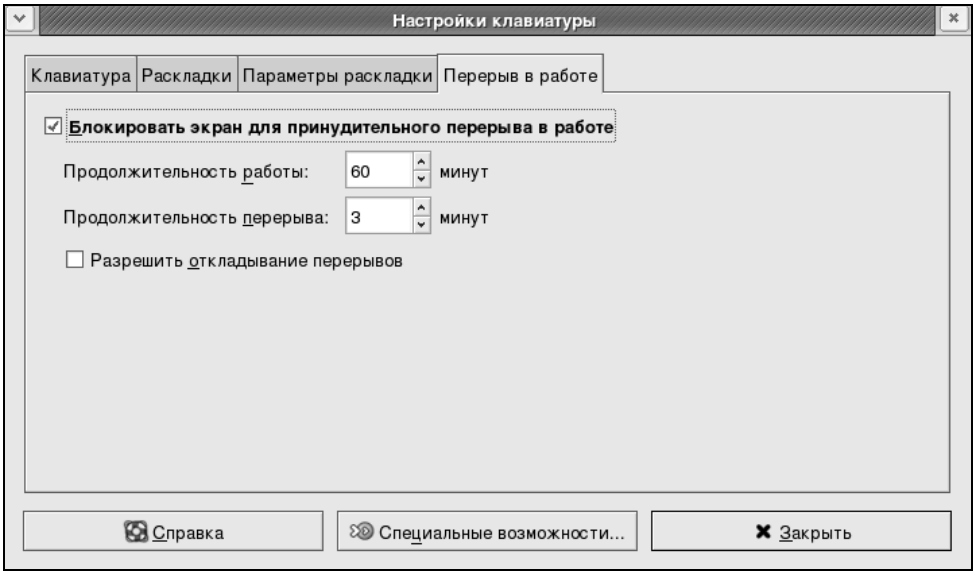

Рис. 6.9. Вкладка **Перерыв в работе**

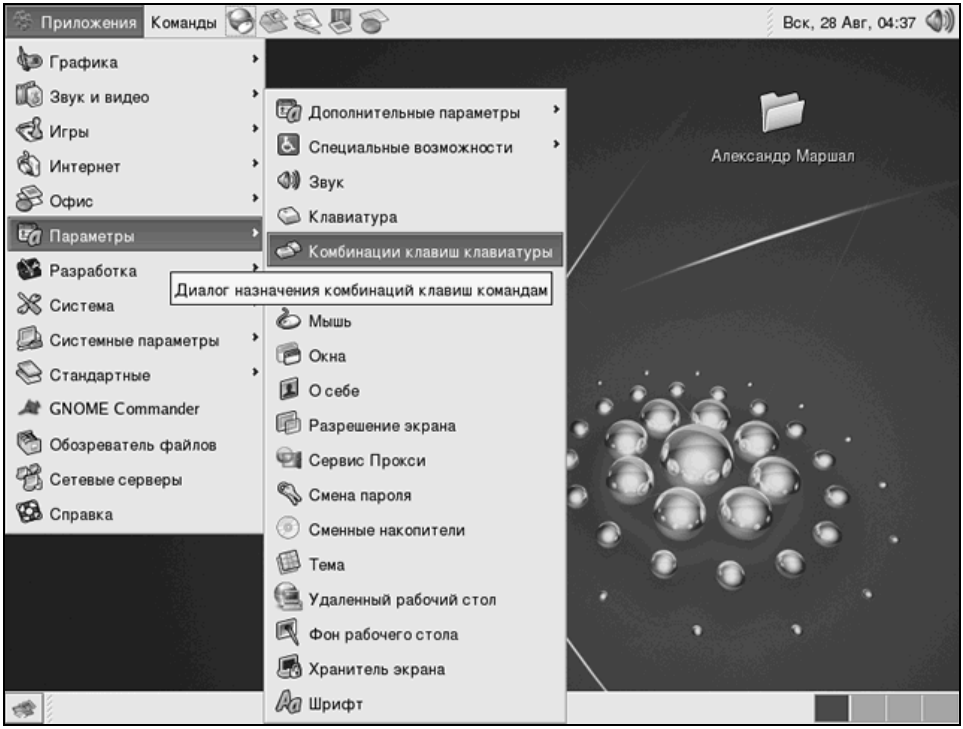

Рис. 6.10. Настройка комбинации клавиш

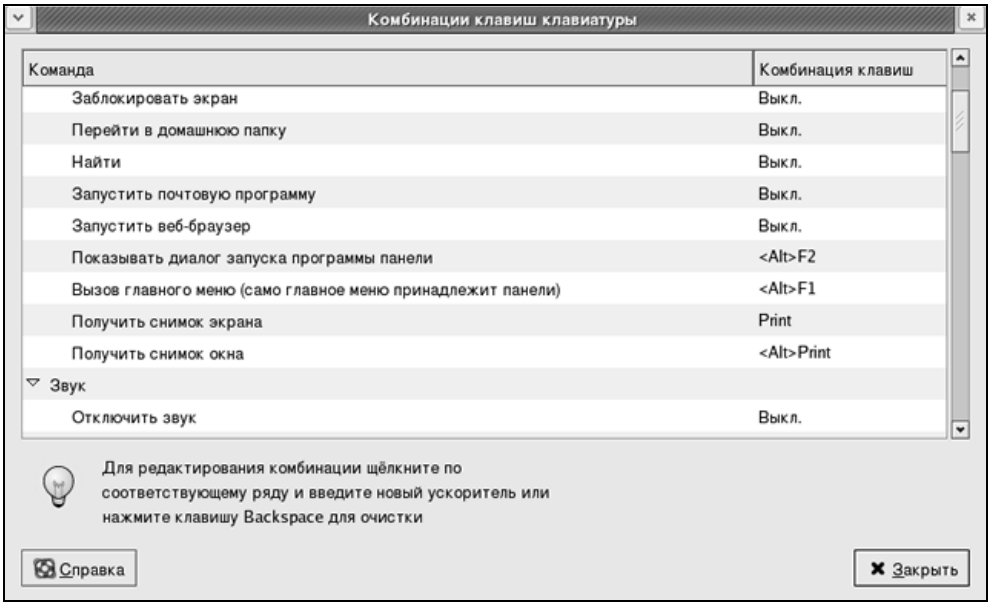

Рис. 6.11. Диалоговое окно настройки комбинаций клавиш

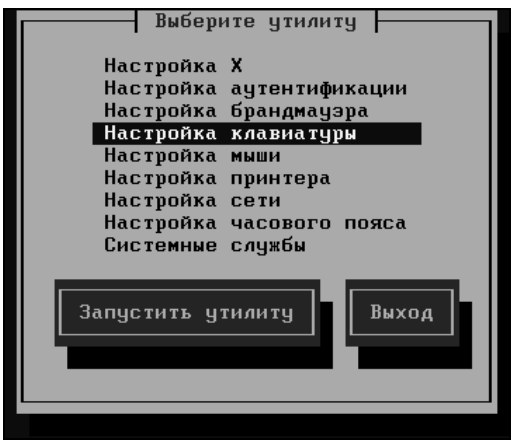

Рис. 6.12. Настройка клавиатуры в текстовом режиме

Для того чтобы настроить горячие клавиши (ускоритель), необходимо при помощи ползунка найти интересующее вас действие и щелкнуть один раз мышью по строке, а затем нажать выбранную вами комбинацию клавиш. При ее нажатии теперь будет выполняться выбранное вами действие.

Справедливости ради стоит сказать, некоторые параметры клавиатуры можно настроить и в текстовом режиме. Для этого в командной строке наберите

```
setup
```

и нажмите <Enter>.

В появившемся окне при помощи стрелок на клавиатуре или при помощи мыши выберите строку **Настройка клавиатуры** и нажмите <Enter> (рис. 6.12). Далее следуйте указаниям мастера.

6.2.2. Mandrake

Точно так же настроить клавиатуру можно и в Mandrake. Вначале настроим раскладку клавиатуры. Для этого пройдем по пути **Главное меню | Система | Настройка | Настройка компьютера** (рис. 6.13).

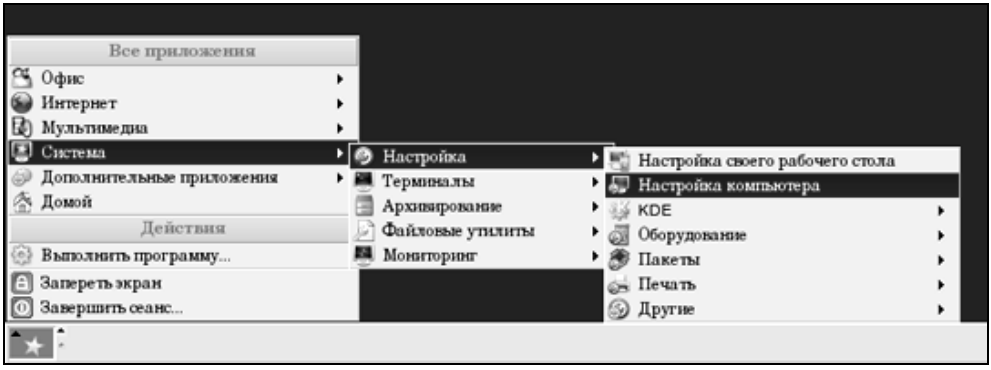

Рис. 6.13. Запуск настройки мыши

Откроется окно **Центр управления Mandrake**. В появившемся окне необходимо щелкнуть по значку **Оборудование**. Теперь вы находитесь в окне настройки оборудования (рис. 6.14).

Теперь в появившемся окне необходимо щелкнуть по значку **Клавиатура**, откроется диалог настройки клавиатуры. Первым делом вам предложат выбрать раскладку клавиатуры, выбор для каждого очевиден (рис. 6.15).

Следующим шагом будет выбор клавиши, которая будет переключать вас между выбранными раскладками (рис. 6.16). После завершения выбора нажимаем кнопку **OK**.

И снова после завершения выбора нажимаем кнопку **OK**. Далее появится информационное окно, которое говорит о том, что сделанные изменения вступят в силу после перезагрузки (рис. 6.17).

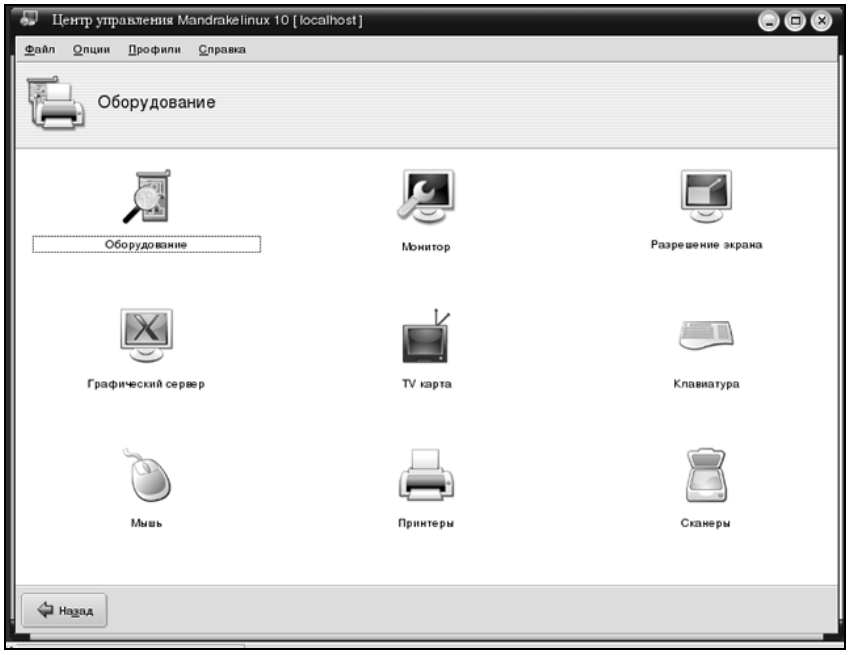

Рис. 6.14. Окно **Оборудование**

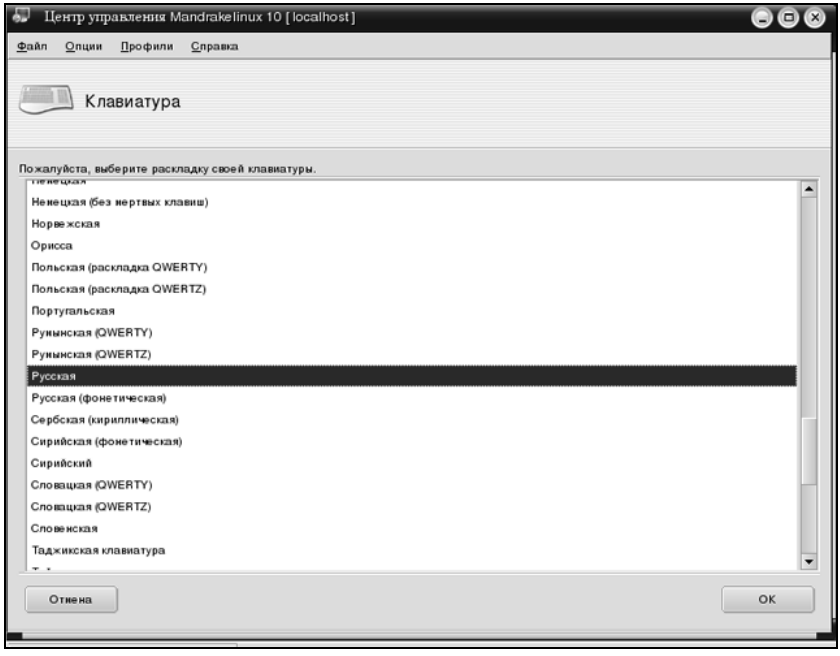

Рис. 6.15. Выбор раскладки клавиатуры

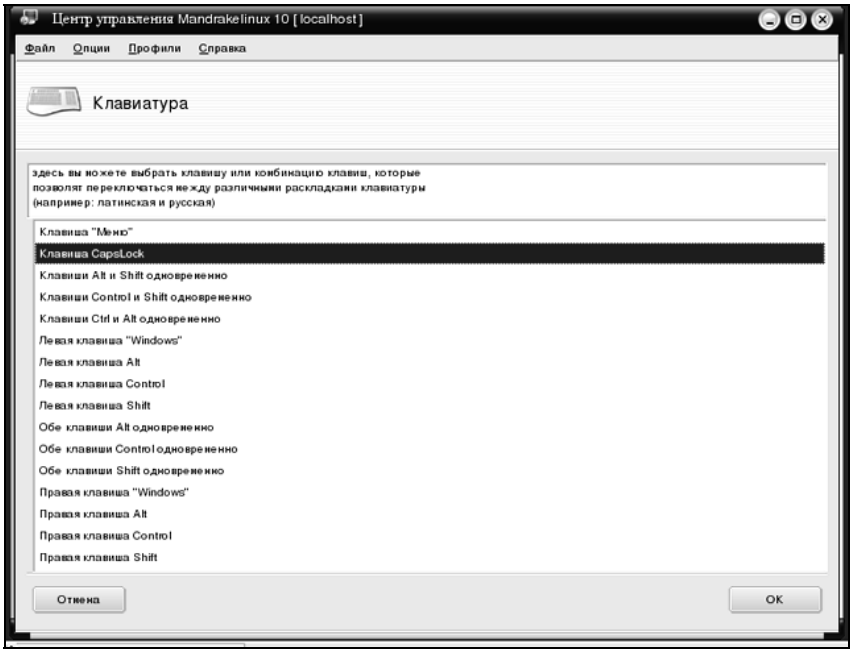

Рис. 6.16. Выбор переключателя раскладок

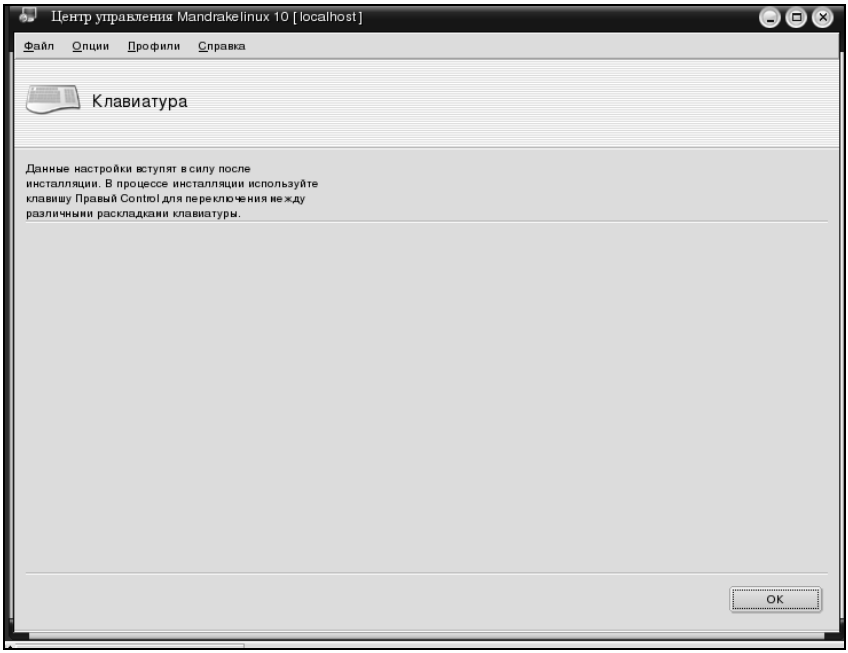

Рис. 6.17. Завершение настройки

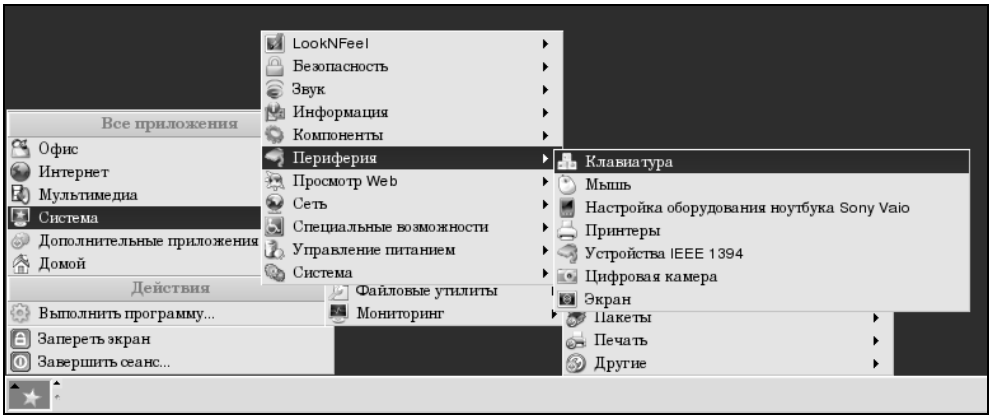

Рис. 6.18. Запуск настроек клавиатуры

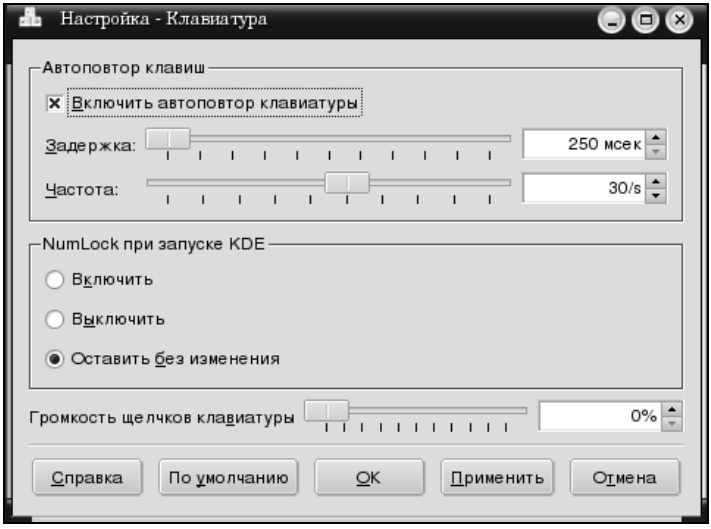

Рис. 6.19. Окно **Настройка - Клавиатура**

Помимо уже изученных нами настроек, в Mandrake, как и в ASPLinux, возможны и другие, правда, эти настройки выполняются уже через оболочку KDE. Запускаем **Главное меню | Система | Настройка | KDE | Периферия | Клавиатура** (рис. 6.18).

Откроется диалог настройки параметров клавиатуры (рис. 6.19).

В указанном окне выбирается скорость повторения клавиш при нажатии, параметры установки клавиши <Numlock>, а также громкость щелчков при нажатии клавиш клавиатуры.

6.3. Мышь и ее конструкция

Наряду с клавиатурой, еще одной неотъемлемой частью современного компьютера является мышь. Мышь представляет собой координатное устройство ввода. При перемещении мыши по столу она фиксирует все ваши манипуляции и передает их компьютеру. В конечном итоге для вас это выливается в перемещение курсора мыши по экрану.

Помимо устройства, фиксирующего перемещение по столу, стандартная мышь может иметь две, три или более кнопок и скрол (дополнительное колесико сверху). На сегодняшний день наиболее распространена трехкнопочная мышь со скролом. Наличие третьей кнопки мыши для Linux особенно важно, так как часть вспомогательных функций возложена именно на среднюю кнопку мыши. Для тех случаев, когда мышь двухкнопочная, можно настроить эмуляцию третьей кнопки. В этом случае вместо нажатия третьей кнопки нажимаются левая и правая кнопки мыши одновременно.

Как и в случае с клавиатурой, мышь может подключаться либо по разъему PS/2, либо по USB. Выделить из двух стандартов более распространенный достаточно сложно. Для того чтобы отличить, куда подключать мышь, а куда клавиатуру, существуют специальные значки. Кроме того, имеется цветовая градация. Разъем для подключения мыши имеет зеленый цвет (рис. 6.20).

Рис. 6.20. Стандарты подключения мыши

При подключении мыши необходимо соблюдать осторожность и следить за правильностью подключения. Для этого обращайте внимание на направляющие, которые присутствуют в разъеме на материнской плате и у мыши.

Взаимодействие мыши с операционной системой осуществляется при помощи специальной программы (драйвера) для стандартных двух- и трехкнопочных мышей. Драйвер, как правило, можно найти в списке устройств, поддерживаемых операционной системой. Для более "реликтовых" мышей с большим количеством дополнительных кнопок драйверы поставляются отдельно.

6.3.1. Оптическая мышь

Несмотря на то, что оптический метод регистрации перемещений в манипуляторах типа мышь стал использоваться недавно, мыши подобной конструкции приобрели большую популярность и полностью вытеснили своих предшественников (шариковые мыши). Конструкция оптической мыши приведена на рис. 6.21.

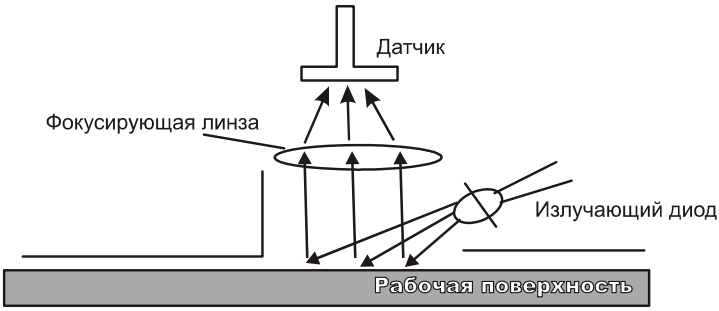

Рис. 6.21. Конструкция оптической мыши

Принцип работы оптической мыши обусловлен датчиком отраженных лучей, модулированных изменением отражающей поверхности. Преимуществом оптической мыши является более легкая конструкция, устойчивость к загрязнению, возможность работы на любой поверхности и практически в любом положении.

6.3.2. Шариковая мышь

Для фиксирования перемещения мыши по рабочей поверхности используется покрытый резиной металлический шарик. При перемещении мыши по рабочей поверхности шарик вращается (рис. 6.22). Вращение шарика передается на два взаимно перпендикулярных валика, один из которых отслеживает перемещение по оси X, а второй по оси Y.

Валики имеют на своих концах диски. В поверхности дисков проделаны отверстия. При вращении валика вращается и диск, в результате в поле датчика чередуются отверстия и сплошные участки, через которые либо проходит, либо не проходит луч света. По количеству и интенсивности чередований этих участков можно судить о скорости перемещения мыши и дистанции, на которую ее переместили.

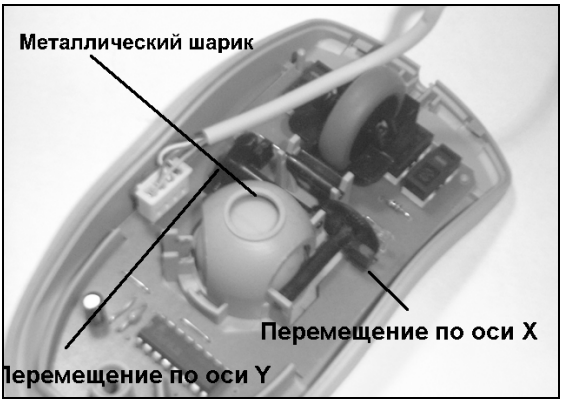

Рис. 6.22. Схема шариковой мыши

6.4. Настройка мыши в ОС Linux

6.4.1. ASPLinux

Начнем с ASPLinux: первичная настройка выполняется в терминальном режиме. В командной строке наберите

```
setup
```

и нажмите <Enter>. Перед вами появится окно, как на рис. 6.23.

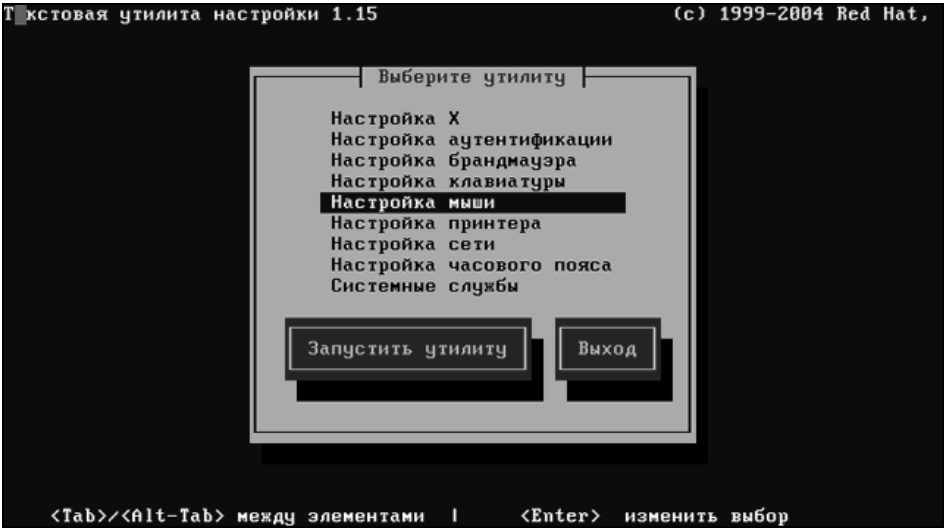

Рис. 6.23. Настройка мыши, шаг 1

При помощи стрелок на клавиатуре выберите строку **Настройка мыши** и нажмите <Enter>. Вы попадете в окно настройки мыши (рис. 6.24).

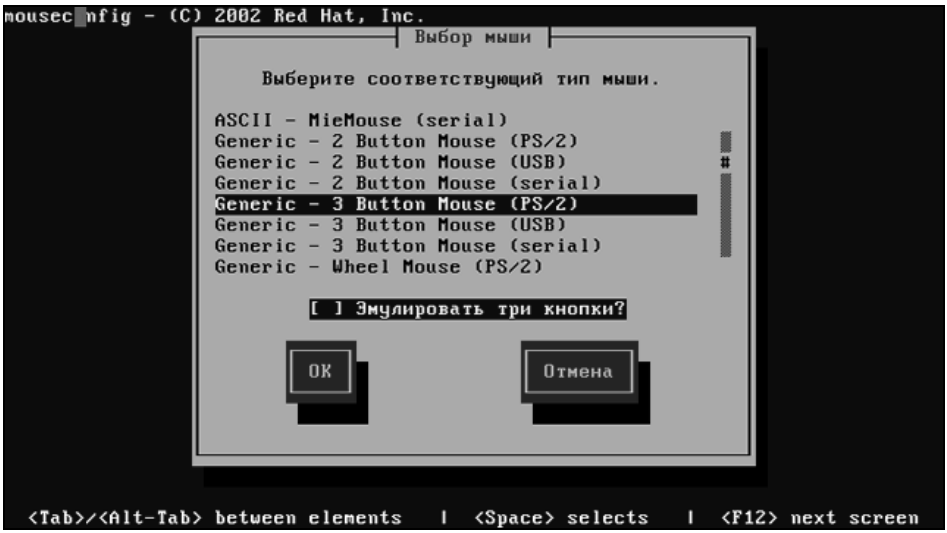

Рис. 6.24. Настройка мыши, шаг 2

В появившемся окне при помощи стрелок необходимо выбрать тип, который соответствует вашей мыши. Если мышь двухкнопочная, то у вас появится возможность эмуляции трех кнопок по нажатию одновременно левой и правой кнопок мыши. Для перемещения между полями экрана используйте клавишу <Tab>. После того как вы все настроили, нажмите кнопку **OK**. В результате остановится и запустится заново служба поддержки мыши, а вы вернетесь в уже знакомое окно, представленное на рис. 6.23. Нажимаем кнопку **Выход**.

В графическом режиме у мыши настраиваются параметры мыши для левши, задержка двойного нажатия, тип курсора и скорость его перемещения. Для доступа к диалогу настройки мыши необходимо выбрать: **Приложения | Параметры | Мышь** (рис. 6.25).

Поле того как вы щелкнете мышью по выбранному пункту, перед вами откроется окно настройки параметров мыши (рис. 6.26).

На первой вкладке можно выбрать мышь под левую руку. При этом правая и левая кнопки мыши поменяются местами. Также здесь можно выбрать скорость двойного щелчка мышью, потренироваться можно на лампочке. Если она загорается в полную силу, значит, все нормально.

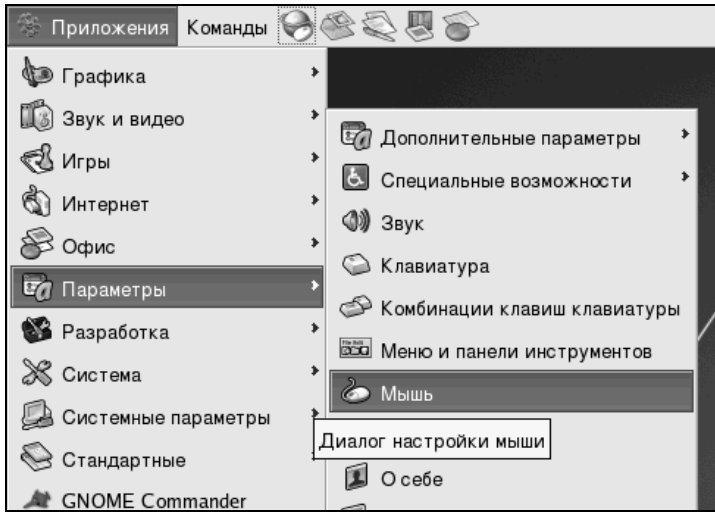

Рис. 6.25. Начало настройки мыши

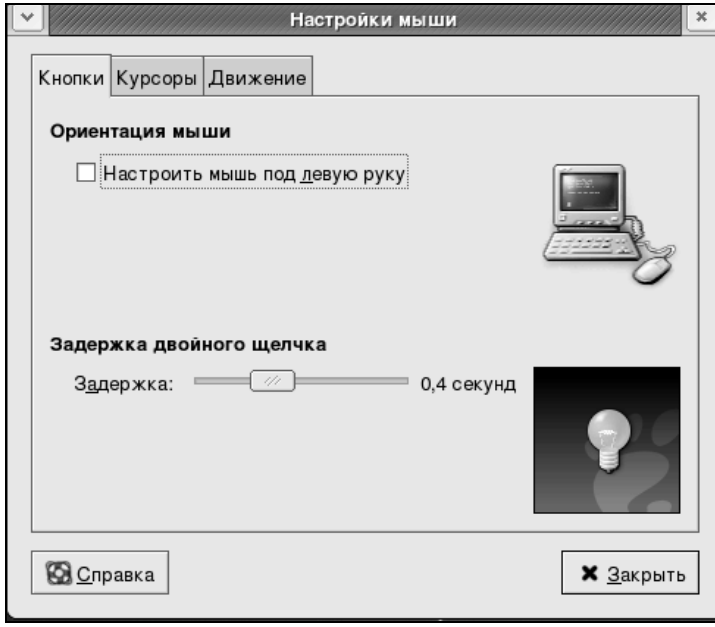

Рис. 6.26. Вкладка **Кнопки**

Следующая вкладка **Курсоры** (рис. 6.27) позволяет выбрать размер курсора, а также подсвечивать курсор при нажатии кнопки <Ctrl>.

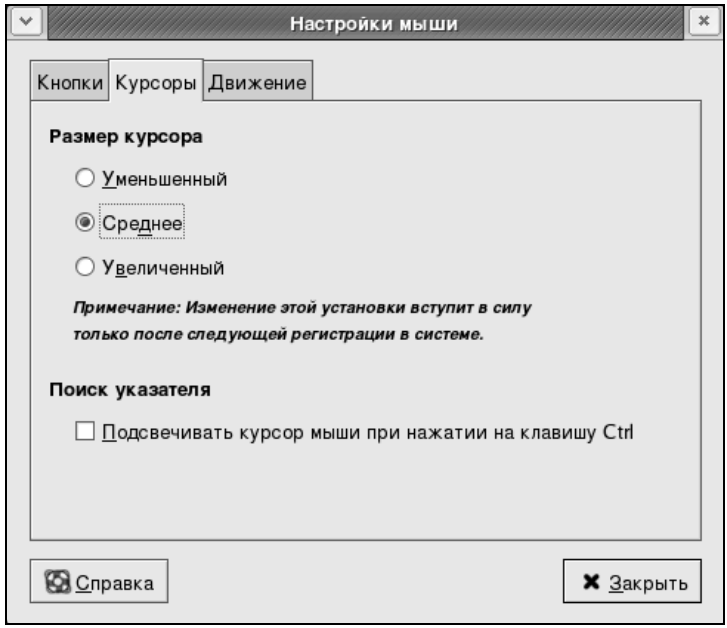

Рис. 6.27. Вкладка **Курсоры**

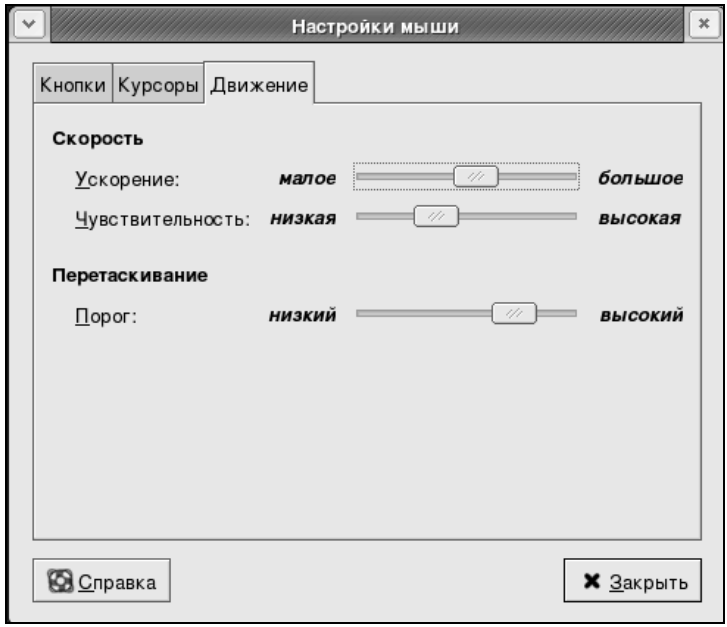

Рис. 6.28. Вкладка **Движение**

Изменения, внесенные в данном окне, вступят в силу только после перезагрузки графического режима.

Вкладка **Движение** на рис. 6.28 позволяет настроить параметры, отвечающие за перемещение мыши.

Доступны такие параметры, как ускорение, чувствительность, порог. Начальные установки достаточно удобны, однако если они вас не устраивают, то можете их сменить.

6.4.2. Mandrake

Настройка в Mandrake совершенно равноправна и для терминального, и для графического режима. В терминальном режиме в командной строке наберем

```
mouseconf
```

и нажмем <Enter>. Появится диалоговое окно, в котором вам необходимо выбрать тип мыши (рис. 6.29).

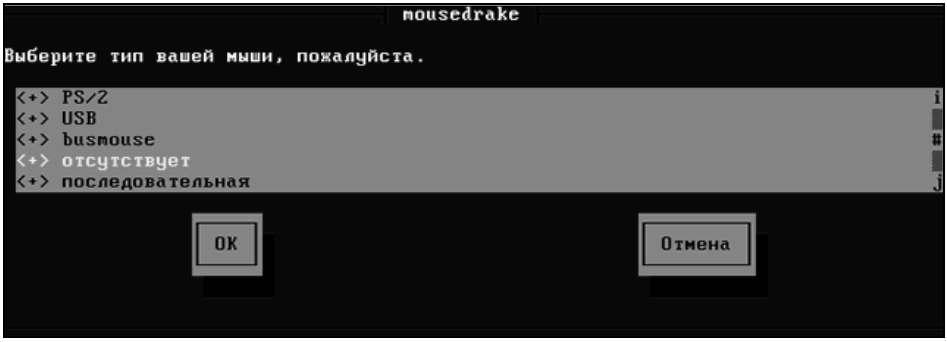

Рис. 6.29. Настройка мыши в терминальном режиме

В графическом режиме выбираем **Главная кнопка | Система | Настройка | KDE | Периферия | Мышь** (рис. 6.30).

Откроется окно, близкое к окну настроек для ASPLinux (рис. 6.31).

Как и в случае с ASPLinux, здесь есть настройка мыши для левши, а также некоторые другие вкусности. Вторая вкладка **Тема курсоров** позволяет выбрать тему для курсора (рис. 6.32).

Следующая вкладка **Дополнительно** на рис. 6.33 регулирует скорость перемещения курсора, скорость двойного щелчка, а также другие параметры, характеризующие перемещение курсора мыши.

Последняя вкладка **Управление курсором при помощи клавиатуры** (рис. 6.34) позволяет настроить перемещение курсора мыши при помощи стрелок.

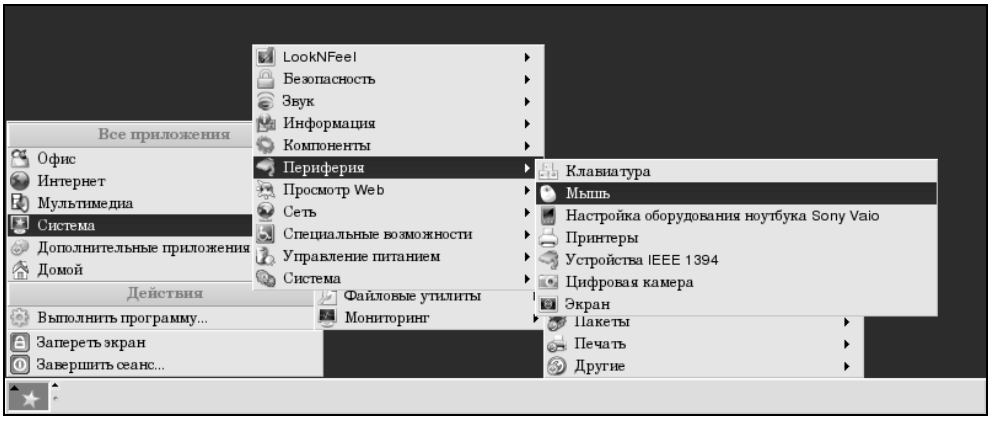

Рис. 6.30. Запуск настройки мыши

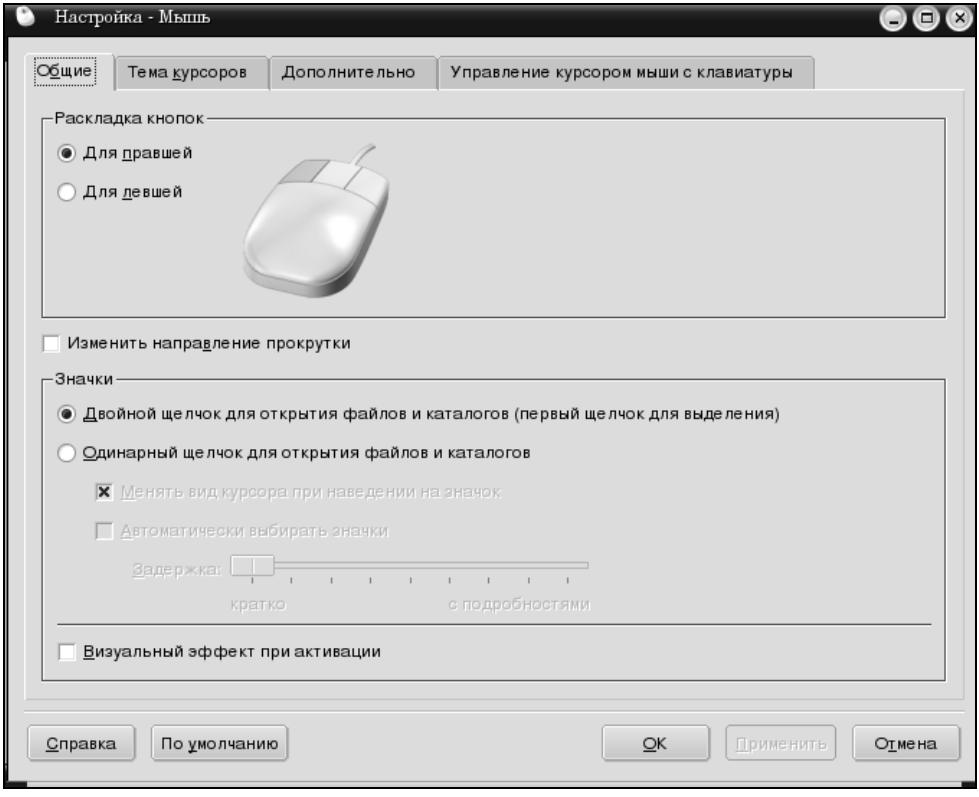

Рис. 6.31. Вкладка **Общие**

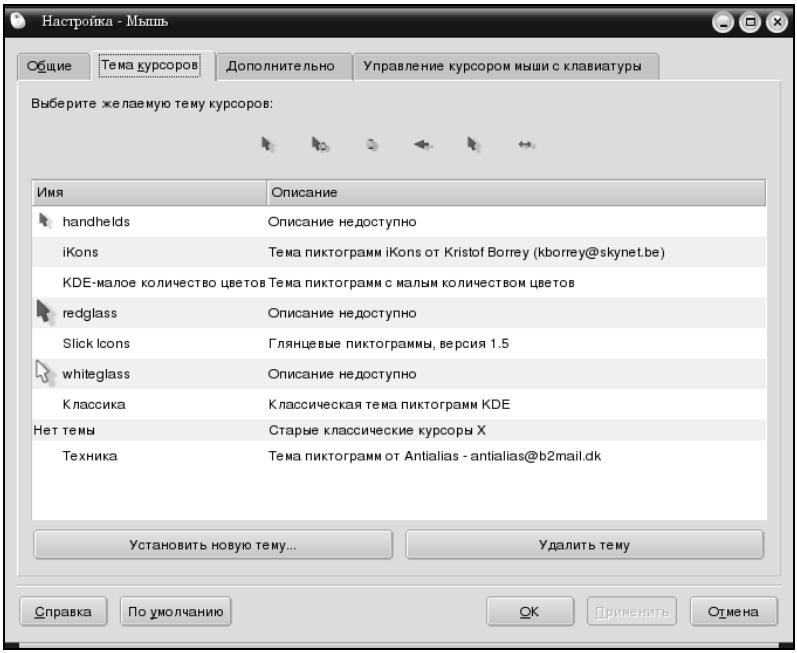

Рис. 6.32. Вкладка **Тема курсоров**

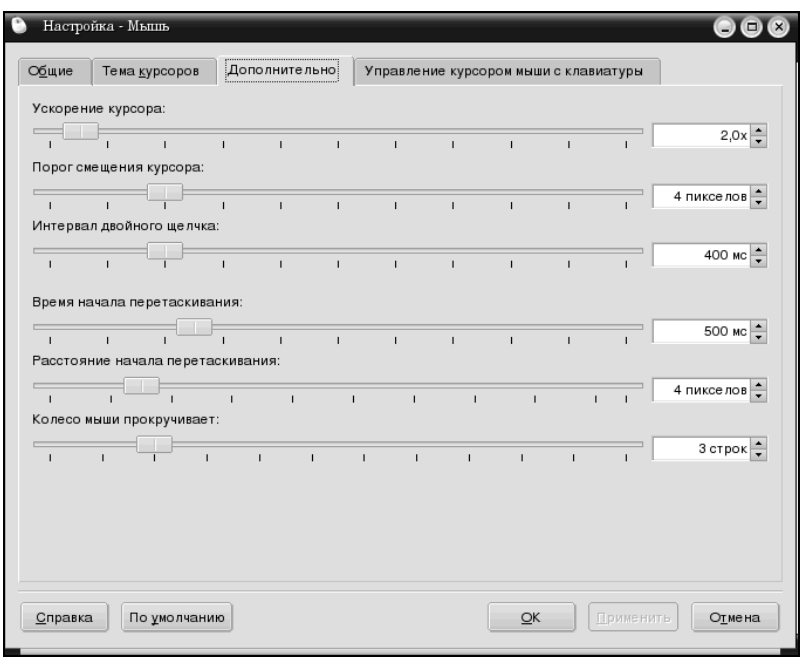

Рис. 6.33. Вкладка **Дополнительно**

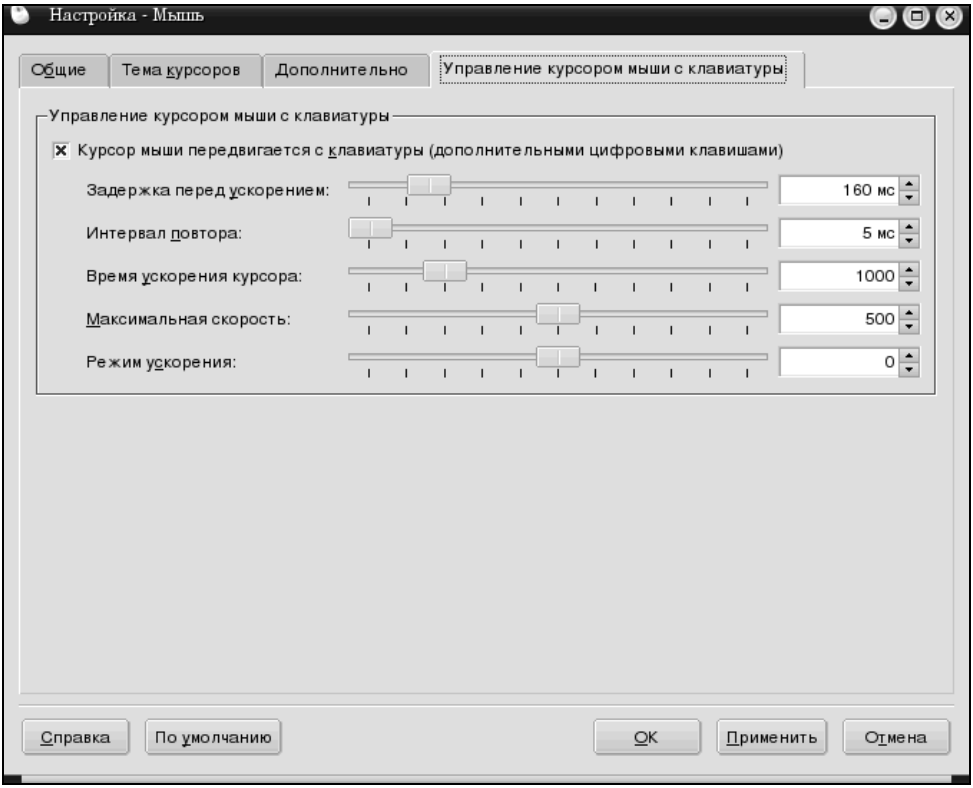

Рис. 6.34. Вкладка **Управление курсором с клавиатуры**

Регулируя такие параметры, как **Ускорение курсора**, **Порог смещения курсора**, **Время начала перетаскивания**, **Расстояние начала перетаскивания**, можно настроить вполне сносное перемещение курсора мыши под действием клавиш на цифровой клавиатуре.

Чтобы сделанные нами изменения вступили в силу, необходимо перезапустить KDE.

Глава 7

Печать в Linux

В настоящее время все мы являемся свидетелями очень важного события, причем сами этого не осознаем. Сейчас происходит стирание грани между материальным и нематериальным. Смотрите сами: цифровое фото, цифровые документы, цифровая подпись, электронные книги, даже есть электронные деньги. Вроде бы документ есть, а вроде бы его и нет. На экране компьютера он есть, его можно передать с компьютера на компьютер, чтобы другие пользователи его просмотрели. Но пощупать-то его нельзя, физически его не существует. Но у цифрового нематериального мира есть один предатель — это принтер. Принтер позволяет воплотить нематериальное в материальное — распечатать ваши любимые фотографии, только что полученный счет, а также недавно загруженную электронную копию книги. Не секрет, что в цифровую эру многим из нас хочется вернуться в прошлое, а именно распечатать фотографии и аккуратно поместить их в альбом, распечатать документ, чтобы, отойдя от компьютера, прочитать его. Причины тому разные: кому-то по-прежнему удобно работать с "твердыми" копиями документов, кто-то хочет взять в дорогу фотографии любимого человека, забыв на время о компьютере, а у кого-то — просто ностальгия…

В этой главе мы поговорим о типах принтеров, о выборе принтера, его подключении, а также настройке вашего принтера в Linux.

7.1. Типы принтеров

Понятно, что принтеры бывают разными. Выбор принтера зависит от цели его использования, а также от наличия определенной суммы, которую вы хотите потратить на его приобретение. Нужно отметить, что цена принтера сильно зависит от типа принтера, например, отличный цветной струйный принтер с возможностью фотопечати будет стоить дешевле, чем простенький черно-белый лазерный принтер.

Итак, принтеры бывают трех типов:

- ударные;
- струйные;
- контактные.

7.1.1. Ударные принтеры

В свою очередь, ударные принтеры делятся на следующие подтипы:

- матричные;
- игольчатые;
- барабанные;
- цепные;
- лепестковые.

Ударная технология печати является самой старой. Самые первые печатающие устройства были именно ударного типа.

Принцип действия ударного принтера следующий: частицы краски переносятся на бумагу с помощью удара ударным устройством по синтетической ленте, пропитанной краской. В роли ударного устройства может выступать, например, набор игл, с помощью которого формируется определенный символ. У каждого типа ударного принтера свое собственное ударное устройство.

По мере развития техники, развивались и требования к представлению информации в документах. Основные направления совершенствования технологии были направлены на устранение указанных выше недостатков. Это привело к созданию технологии игольчатой печати. При реализации технологии игольчатой печати, за основу был взят принцип переноса красящего вещества с носителя на бумагу от ударного воздействия. Нововведением явилось то, что формирование буквы идет не статически, а динамически. Для того чтобы понять, как формируется изображение в матричном принтере, рассмотрим рисунок, раскрывающий принцип работы матричного принтера (рис. 7.1). На рис. 7.1, *а* изображена схема взаимного расположения основных частей матричного принтера. На рис. 7.1, *б* изображена головка матричного принтера. На рис. 7.1, *в* изображена технология получения слова при игольчатой печати.

На рис. 7.1, *а* видно, что бумага расположена так, что непосредственно прилегает к барабану. Это определяет возможность получения качественного оттиска, так как при ударном воздействии чрезвычайно важно иметь жесткую прокладку. Это определяет силу удара, а значит, и качество отпечатков, и наличие четкого контрольного оттиска.

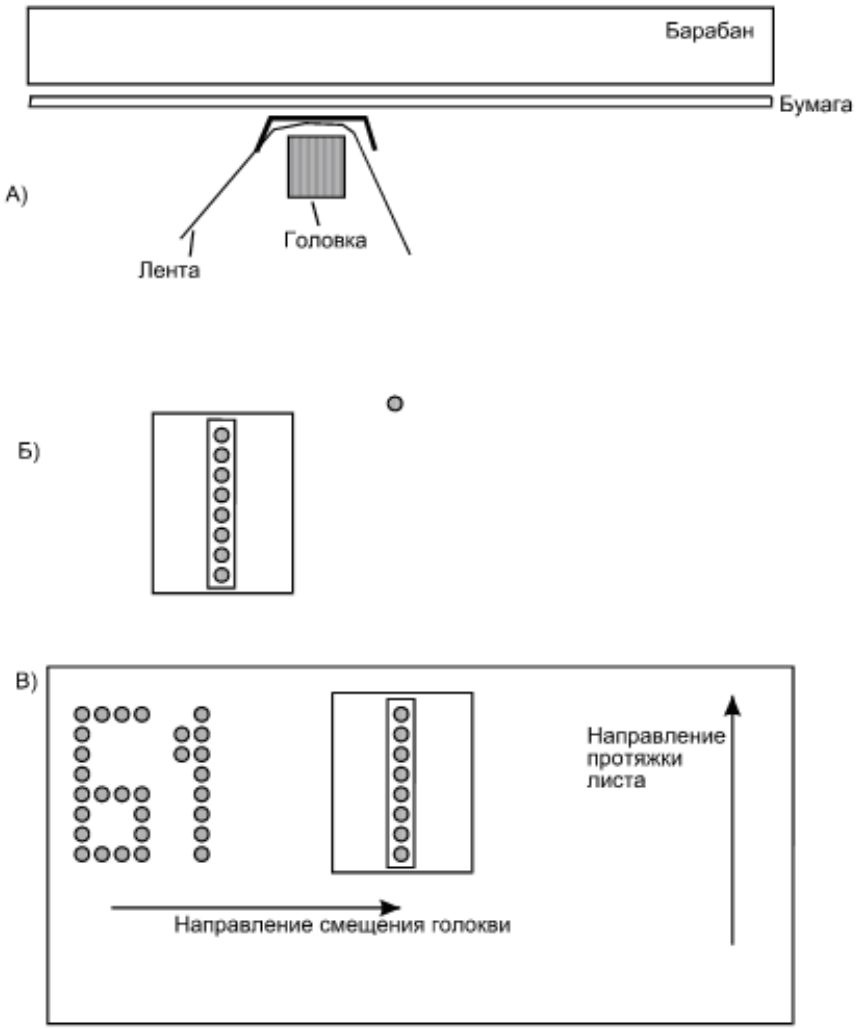

Рис. 7.1. Принцип работы матричного принтера

Если посмотреть на рис. 7.1, *б*, то можно увидеть, что головка представляет собой несколько (как правило, 9 или 24) иголок, расположенных в ряд. Каждая иголка управляется отдельным электромагнитом. При ударе одной иголкой краситель переносится с красящей ленты на бумагу, получается маленькая точка. Совокупность точек, поставленных на бумаге во время перемещения головки поперек листа бумаги и протяжки листа бумаги, образует рисунок или текст.

Собственно, сам принтер состоит из следующих компонентов:

- модуль печати — выполняет печать под "руководством" модуля управления;
- модуль управления и индикации — обеспечивает обмен данными с компьютером, индикацию режимов принтера и управляет модулем печати;
- блок питания — назначение этого компонента, думаю, понятно.

В принципе, можно сказать, что приведенную структуру имеют все принтеры — меняется лишь блок печати, который зависит от выбранной технологии печати, например, струйные или лазерные принтеры используют совершенно другую технологию печати.

Принтеры ударного типа обеспечивают весьма посредственное качество печати текста, а графику на них лучше вообще не печатать. Плохое качество печати обусловлено устаревшей технологией. Спрашивается, почему же до сих пор в некоторых организациях используются ударные принтеры? Причины тому две. Первая — нужна быстрая и дешевая печать текста, при которой качество особой роли не играет. Вторая — у организации просто нет денег на приобретение и обслуживание новых типов принтеров. Ведь ударные принтеры — самые дешевые в обслуживании. Кроме дешевизны, матричные принтеры очень надежны — средняя наработка на отказ составляет, в среднем, около 6 тысяч часов. Но, пожалуй, на этом преимущества матричных принтеров заканчиваются.

А недостатков значительно больше:

- ограниченная возможность печати графики — об этом уже говорилось;
- слишком шумная работа — некоторые принтеры по шумности напоминают механические печатающие машинки "Ятрань" (это, конечно, я преувеличил, но факт остается фактом);
- для каждого языка нужна своя головка — если ваш принтер не печатает символы нужного вам языка, значит, скорее всего, вам нужно будет менять его головку и драйвер.

Нужно отметить, что в последнее время ударные принтеры встречаются довольно редко, в продаже их практически нет. Но не думайте, что ударные принтеры уже не производят, их все еще выпускают.

На рис. 7.2 изображен один из немногих матричных принтеров, который продается до сих пор — это Epson LX-1170, матричный 9-игольчатый принтер формата А3. Данный принтер позволяет печатать со скоростью 337 символов в секунду. Это не самый быстрый принтер, зато довольно экономный и надежный.

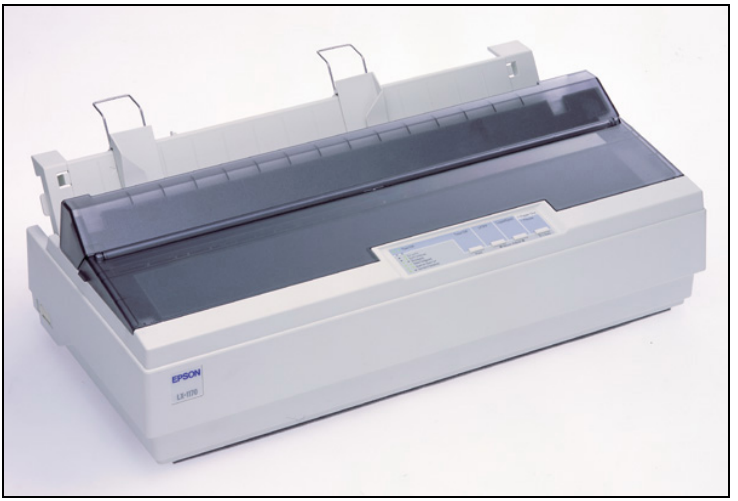

Рис. 7.2. Матричный принтер

7.1.2. Струйные принтеры

Со временем уровень печати, обеспечиваемый матричными принтерами, перестал удовлетворять запросам потребителей: текст они еще печатали, а вот, скажем, графику — уже нет (я молчу о цветных фотографиях). С появлением на рынке струйных принтеров произошла настоящая эволюция (революция связана с разрушением, поэтому лучше использовать все-таки термин "эволюция") в мире печати. В домашних условиях на недорогом струйном принтере можно было распечатать графические изображения и даже цветные фотографии. Качество печати, которое обеспечивали даже недорогие "струйники", было существенно выше качества печати матричных принтеров.

В двух словах принцип действия струйных принтеров можно описать так: чернила на бумагу наносятся с помощью дозирующей головки.

Достаточно распространены "матрично-струйные" принтеры: принцип их действия похож на принцип действия матричного принтера, но вместо иголок в головке находятся сопла (они называются дюзами), распыляющие чернила. Число сопел в головке достаточно велико и может достигать от 16 до 128 на один цвет.

В зависимости от способа формирования струи, струйные принтеры бывают пузырьковые (термические) и пьезоголовочные.

Термические (bubble в англоязычной литературе) принтеры работают так: чернила в дюзе резко нагреваются под действием электротока, в результате

чего образуется струя. Поскольку чернила "закипают", то образуются пузырьки, которые стремятся вырваться наружу. Представьте закипающий чайник — это что-то вроде того. Пузырьки летят в сторону бумаги, лопаются на ней и формируют изображение. Пузырьки настолько маленькие, что для формирования одной точки изображения нужно несколько пузырьков (не нужно думать, что они такого размера, как пузырьки закипающего чайника).

Пьезоголовочные принтеры (рис. 7.3) работают иначе. У них струя образуется при изменении размеров пьезокристалла, которое происходит под действием электротока. Поскольку размер пьезокристалла изменяется, это выталкивает чернила в сторону бумаги. В отличие от пузырьковых принтеров, здесь одна точка изображения формируется всего одним пятном.

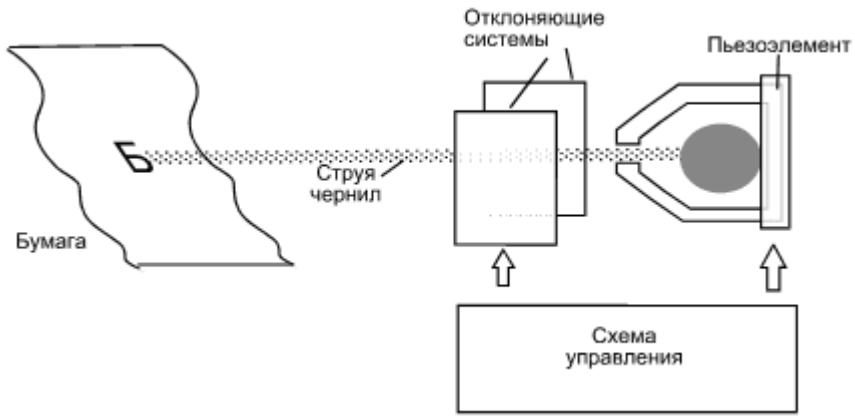

Рис. 7.3. Структурная схема струйного принтера

Да, технологии разные, но результат практически одинаковый: обе технологии позволяют выводить черно-белые и цветные изображения, причем делают это приблизительно с одинаковым качеством.

Струйный принтер (рис. 7.4) — идеальное решение для домашних пользователей:

- Струйные принтеры относительно дешевы (есть и еще дешевле, но стоит ли их покупать?).
- Относительно дешевы расходные материалы — картриджи с чернилами. Хотя при покупке убедитесь, что картридж и головка не интегрированы, иначе придется менять картридж вместе с печатающей головкой, а это в несколько раз дороже, чем просто замена картриджа. В качестве альтернативы можно предложить дозаправку картриджа, если таковая допускается.

- Струйные принтеры работают очень тихо и не опасны для здоровья, как лазерные, что немаловажно для домашнего устройства.
- Возможность фотопечати — большинство современных струйных принтеров не только цветные, но и могут печатать фотографии, что также будет интересно домашним пользователям. Конечно, я сомневаюсь, что стоимость одной такой фотографии будет ниже, чем стоимость ее печати в цифровой лаборатории, но это уже другой вопрос — главное, что вы сами сможете печатать цветные фотографии на фотобумаге.
- Немаловажны хорошее качество печати и приемлемая скорость печати.

Рис. 7.4. Струйный принтер Canon i250

В общем, если вы домашний пользователь — покупайте струйный принтер. Ресурса его картриджа (до дозаправки или замены) вполне хватит для домашнего использования. Вот только у струйных принтеров есть один небольшой недостаток: если вы надолго уезжаете и принтером долго не будут пользоваться, рекомендуется отнести его знакомым, чтобы они печатали хотя бы по одной страничке в неделю, поскольку чернила в картридже могут просто-напросто засохнуть, особенно летом. Конечно, если головка и картридж не интегрированы, то это не сильно ударит по карману. Если же головка и картридж интегрированы, сразу после отпуска вам придется немного потратиться. Помню, приехав из отпуска, обнаружил, что чернила засохли. А у меня была одна из первых моделей Canon с интегрированной в картридж печатающей головкой. С помощью кустарных методов я его привел в чувства, однако это уже было не то, пришлось менять картридж вместе с головкой. Сейчас, конечно, цены значительно ниже, но лучше сразу все предусмотреть, чтобы потом не было неприятных неожиданностей.

7.1.3. Контактные принтеры

Контактные принтеры делятся на пять типов:

- лазерные;
- термические;
- светодиодные;
- офсетные;
- сублимационные.

Наиболее распространены лазерные принтеры, поэтому о них мы и поговорим. Лазерный принтер наносит изображение на бумагу в шесть этапов (рис. 7.5):

1. Очистка барабана перед печатью — с поверхности вращающегося барабана удаляется неиспользованный после последней печати порошок.
2. Нанесение заряда на барабан — заряд наносится заряженным валиком, поверхность валика выполнена из резины.
3. Нанесение изображения на барабан — на поверхность барабана с помощью лазерного луча наносится изображение. С помощью многогранного зеркала лазерный луч развертывается в нужную точку, в которой энергия луча уничтожает электростатический заряд, знак которого (+ или –) совпадает со знаком заряда тонера.
4. Прокат барабана по валу переноса — на вал переноса наносится порошок (состоящий из пластмассы и железа), при этом заряд порошка равен заряду барабана.
5. Прокат барабана по бумаге — на этом этапе тонер наносится на бумагу, ясно, что только в тех участках, которые соответствуют выводимому изображению.
6. Прокат бумаги по термоэлементу — бумага с нанесенным на нее тонером прокатывается по нагретому термоэлементу для закрепления тонера на бумаге. Получается вроде "закалки" тонера. На этом процесс формирования изображения закончен, и бумага извлекается из принтера в лоток.

Качество печати лазерного принтера очень высокое, скорость печати — тоже. Лазерный принтер нужно выбрать, если вам нужно печатать очень много и, причем, с высоким качеством. Отпечатанные на лазерном принтере документы более долговечны, чем документы, отпечатанные на струйном принтере. Дело в том, что когда документ попадает под воду, краска, нанесенная струйным принтером, может "поплыть" — вида у документа уже не будет. А вот документы, отпечатанные на лазерном принтере, благодаря процессу "закалки" тонера, более стойки к воде — после "купания" их достаточно просушить.

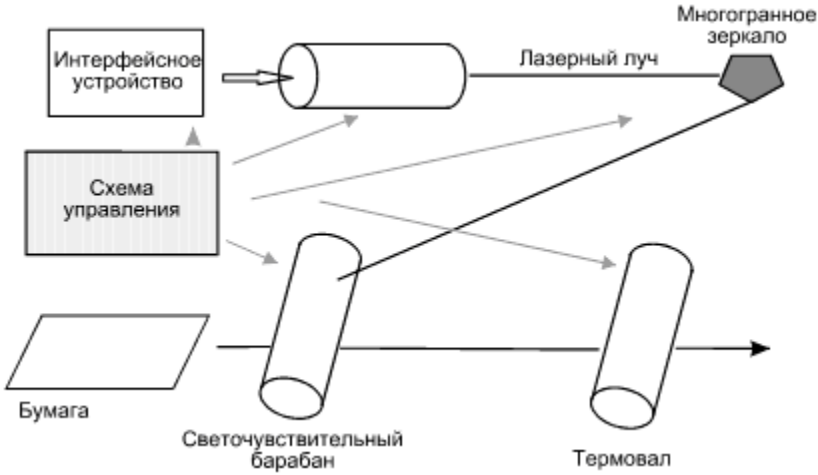

Рис. 7.5. Технология лазерной печати

Рис. 7.6. Лазерный принтер

Преимущественно, встречаются черно-белые лазерные принтеры (рис. 7.6). Но бывают и цветные принтеры — их цена существенно выше.

Нужно отметить, что лазерный принтер не вполне безопасен для здоровья, поэтому помещение, где находится лазерный принтер, должно часто проветриваться. К тому же, лазерный принтер не должен быть установлен в тесном помещении. Заправку тонером нужно производить только в лабораторных

условиях, иначе вы рискуете распрощаться со своим здоровьем — тонер очень неблагоприятно на нем сказывается.

Заправка тонером лазерного принтера стоит дороже, чем заправка краской (или замена картриджа) струйного принтера, но и ресурс у лазерного принтера намного больше.

7.1.4. Выбор принтера

Вы собрались купить принтер? Определить, какой принтер подходит вам, поможет следующая табличка. В ней представлены требования к основным параметрам печати — объему, скорости, качеству, на основании которых вы можете выбрать ваш будущий принтер. Чем выше точек указано в таблице, тем выше значение величины, например, если в колонке Скорость одного принтера стоит только одна точка, то он печатает медленно, две — умеренно, три — быстро. Пустая точка означает, что данная характеристика для вас не важна (табл. 7.1).

Таблица 7.1. *Выбор принтера*

Объем	Скорость	Качество	Цвет	Фото	Тип
●●●	●●	●	○	○	матричный
●●●	●●●	●●●	○	○	лазерный
●●	●●	●●	○	○	струйный
●	●	●●	●	●	цветной струйный
●●●	●●	●●●	●	●	цветной лазерный

Из таблицы видно, что если вам нужно много печатать с приемлемой скоростью, но для вас качество находится не на первом месте, вам нужно выбрать матричный принтер — при этом еще и сэкономите на обслуживании. Если же вам нужно и качество, и скорость, выбирайте лазерный принтер. Он быстро печатает, а себестоимость одной отпечатанной страницы будет меньше, чем у струйного принтера (но больше, чем у матричного). Если вы не часто печатаете, вам нужно хорошее качество и вы не хотите одноразово выкладывать на заправку лазерного принтера, выбирайте струйный. Он недорого стоит, его недорого заправлять, но ресурс его картриджа в несколько меньше, чем

у лазерного принтера. В итоге стоимость печати одной страницы будет выше, чем у лазерного, но это будет не так заметно для вас, поскольку вы печатаете нечасто.

Итак, мы определились с типом принтера. Что же касается меня, то я свой выбор сделал в пользу струйного принтера дома и лазерного — в офисе. Теперь самое время отправиться в магазин. Принтер какой фирмы купить? Если у вас нет определенных предпочтений (например, мне нравится Canon), положитесь на опыт менеджера, он обязательно поможет выбрать оптимальный для вас принтер. Хочу лишь обратить ваше внимание на ресурс службы картриджа (исчисляется в количестве отпечатанных страниц) и на тип картриджа (важно для струйных принтеров — с интегрированной головкой или без нее). Понятно, что лучше покупать такой струйный принтер, у которого печатающая головка не интегрирована в картридж.

GDI-принтеры

Вам нужно уточнить у продавца, сможет ли принтер работать с Linux. Желательно сделать это до покупки. Скорее всего, на компакт-диске с драйверами будут драйверы для всего семейства Windows, а также для MacOS, но не для Linux. Понятно, имея диск с драйверами, в Windows вы настроите принтер без проблем. А вот в Linux иначе. Позже вы ознакомитесь с теми моделями, которые Linux поддерживает явно, но в списке может не быть выбранной вами модели, тогда, как я уже отмечал, лучше всего спросить у продавца, сможет ли данный принтер работать в Linux.

Но тут возникает еще одна проблема: продавец может быть некомпетентен в этом вопросе, попросту говоря — он не знает, будет ли работать принтер в Linux или нет. Конечно, можно оставить деньги и договориться, что вернете принтер, если он откажется работать. Но в целях экономии времени, лучше разузнать все в магазине. Спросите продавца, может ли работать этот принтер в MS DOS? Не в режиме эмуляции, а именно в MS DOS. Если есть рядом где-то компьютер с MS DOS или Windows 9x (который нужно загрузить в режиме командной строки), подключите к нему принтер и введите команду:

```
echo 1111 > PRN
```

Если принтер напечатает четыре единички, значит, можете его покупать. Но такой тест подойдет только для принтеров, которые подключаются к компьютеру с помощью параллельного порта (LPT). Большинство современных принтеров подключаются к компьютеру с помощью USB. Да и где вы компьютер с Windows 9x сейчас найдете?

Поэтому нужно просить у продавца (или прочитать в руководстве по принтеру), является ли этот принтер GDI-принтером или так называемым Win-принтером?

Если да, то такой принтер лучше не покупать — такой принтер вы подключите к Linux только по сети как сетевой принтер. А сам принтер будет при этом подключен к компьютеру под управлением Windows, а в Linux вам придется уже настраивать не принтер, а службу Samba, обеспечивающую подключение Linux к сети Microsoft.

Что же такого страшного в GDI и почему с такими принтерами не работает Linux? На обычный (не GDI) принтер система передачи операционной системы отправляет задание, после этого принтер сам занимается его обработкой и выводом на печать. Обработка информации осуществляется процессором, который есть в любом не GDI-принтере. В GDI-принтере процессора нет, поэтому обработкой информации занимается центральный процессор компьютера, но для того, чтобы он "знал", что и как нужно обрабатывать, используются драйверы принтера. Если нет драйвера, то GDI-принтер не будет работать даже в Windows. Делается это с одной целью — удешевления устройства как такового. Ясно, что на несколько микросхем стало меньше, следовательно, принтер будет стоить дешевле. Как правило, производители GDI-принтеров не утруждаются разработкой драйверов для Linux, ограничиваются только драйверами для Windows и иногда MacOS. Вот именно поэтому данные принтеры еще иногда называют Win-принтерами — они могут работать только в Windows.

Если вам все-таки нужно настроить GDI-принтер в Linux, то, как я уже отмечал, это можно сделать, только лишь подключив его к компьютеру под управлением Windows, а потом настроить его как сетевой принтер. Подобная операция в данной книге рассматриваться не будет. Чтобы немного облегчить вашу задачу, могу порекомендовать статью

http://www.nixp.ru/cgi-bin/go.pl?q=articles;a=win_printing_in_linux.

USB-принтеры

Какой принтер выбрать: LPT или USB? Если есть такая возможность, выбирайте LPT-принтер, то есть принтер, подключающийся к компьютеру через параллельный порт. Если же выбора нет — перед вами USB-принтер, еще раз уточните, что выбранный вами принтер не является GDI-принтером, и… покупайте. Современные дистрибутивы Linux нормально работают с USB-принтерами, но при условии, что они не являются GDI-принтерами.

7.2. Механическое подключение принтера

В большинстве случаев принтеры могут подключаться к компьютеру по LPT-порту (параллельный порт компьютера) или по USB. Как правило, современные принтеры подключаются к компьютеру по USB.

Современные дистрибутивы Linux поддерживают оба типа принтеров, главное, чтобы ваш принтер не был GDI-принтером — а так все будет в порядке.

Если у вас LPT-принтер, перед подключением его к компьютеру выключите питание последнего. Если же вы подключаете USB-принтер, питание компьютера можно не выключать.

7.3. Настройка принтера в Linux

Настройку принтера будем рассматривать на примере двух дистрибутивов — Linux Mandrake 10 и ASP 11. В качестве "подопытного кролика" будет принтер Canon BJC-250ex, подключающийся к компьютеру с помощью параллельного порта.

7.3.1. Настройка принтера в ASP Linux

Для настройки принтера используется конфигуратор system-config-printer (рис. 7.7). Ясно, что перед его запуском нужно подключить принтер к компьютеру и включить его.

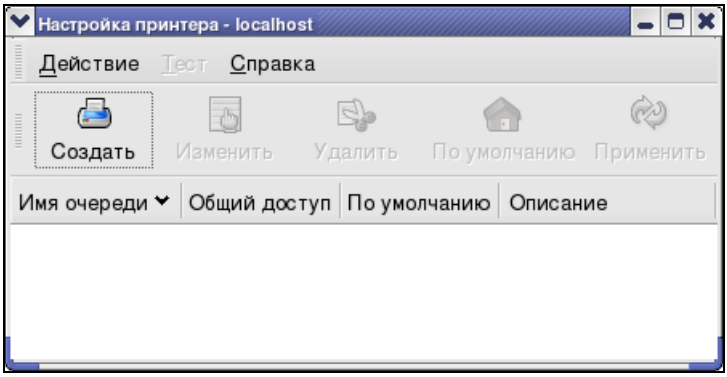

Рис. 7.7. Конфигуратор system-config-printer

Для создания нового принтера нажмите кнопку **Создать**. В появившемся окошке нажмите кнопку **Вперед**, после чего вам будет нужно ввести имя очереди печати (рис. 7.8). Если у вас только один принтер, можете оставить по умолчанию — `printer`, а если у вас несколько, то лучше введите понятные названия для каждого принтера, например, `Canon`, `HP`.

Следующий шаг — это выбор самого принтера (рис. 7.9). В большинстве случаев ASP Linux сам определит принтер и порт, к которому он подключен. В этом случае принтер — Canon-250, а порт — /dev/lp0 (в DOS это LPT1).

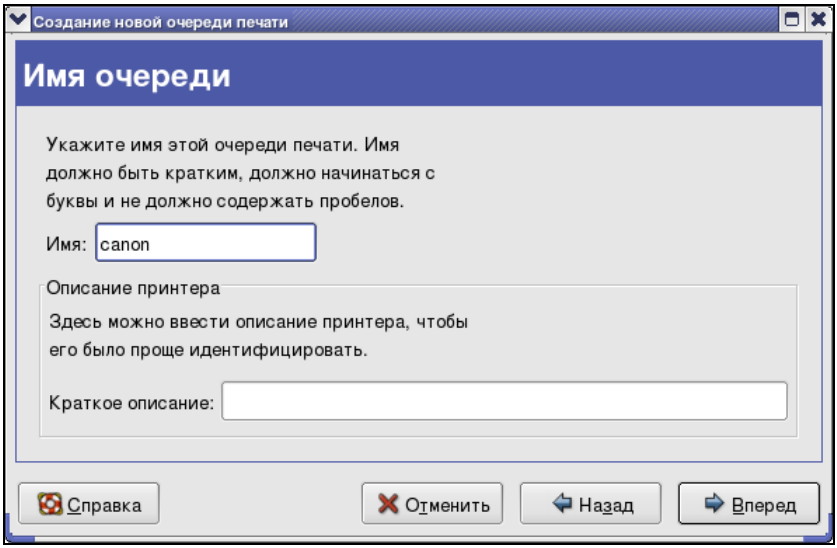

Рис. 7.8. Ввод имени очереди печати

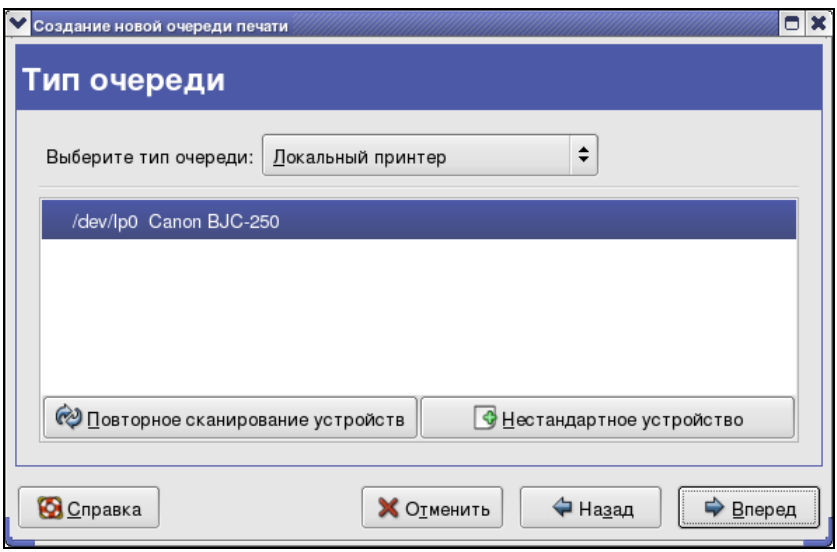

Рис. 7.9. Выбор принтера

После этого вам нужно будет уточнить имя драйвера принтера (рис. 7.10), поскольку конфигуратор может иногда неточно определить модель принтера, например, используется модель Canon-250ex, а конфигуратор определяет его как Canon-250.

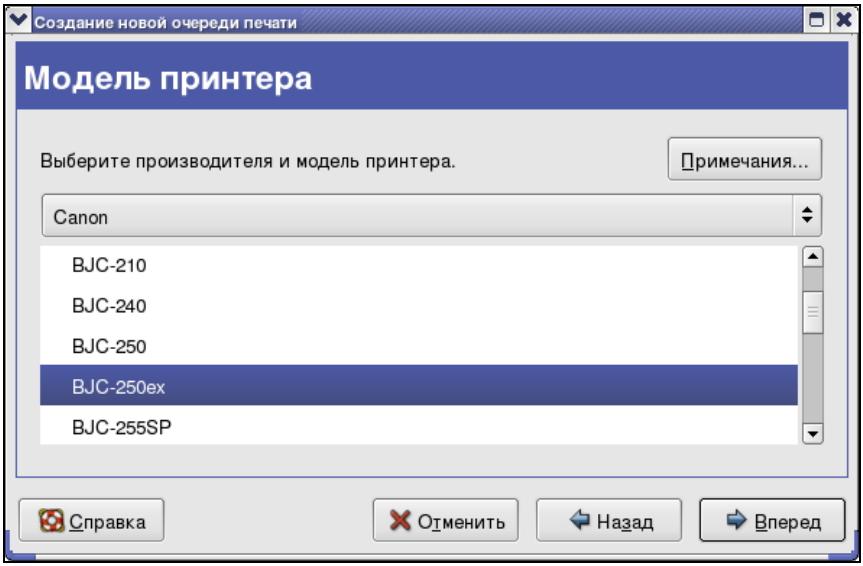

Рис. 7.10. Выбор драйвера принтера

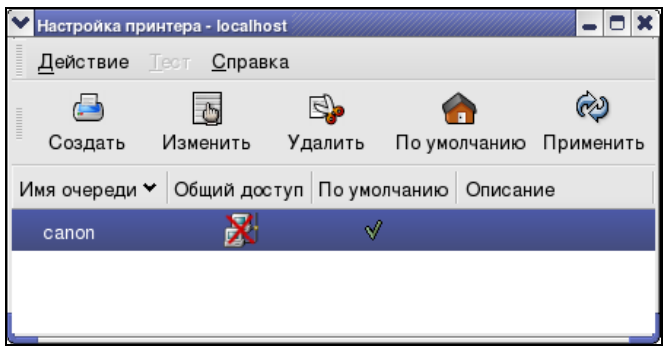

Рис. 7.11. Принтер в окне конфигуратора

На этом настройку принтера можно считать завершенной — ваш принтер появится в окне конфигуратора (рис. 7.11).

Если хотите, можете изменить некоторые параметры принтера. Например, если в вашем принтере черно-белый картридж (для струйных принтеров), вы можете указать это в его опциях, поскольку по умолчанию считается, что если принтер поддерживает цветную печать, то и картридж там цветной (рис. 7.12). Для изменения параметров принтера нажмите кнопку **Изменить**, а затем установите нужные вам параметры.

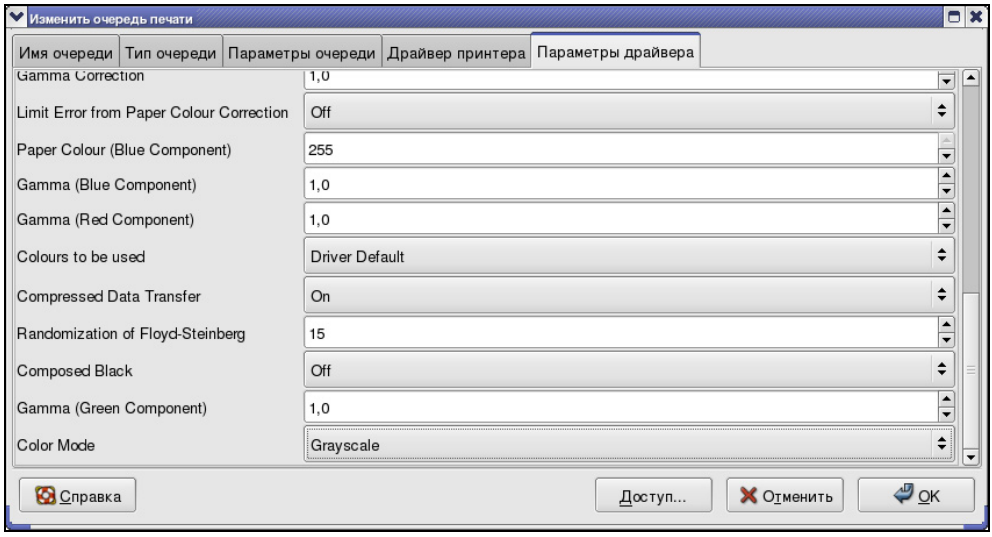

Рис. 7.12. Изменения параметров принтера

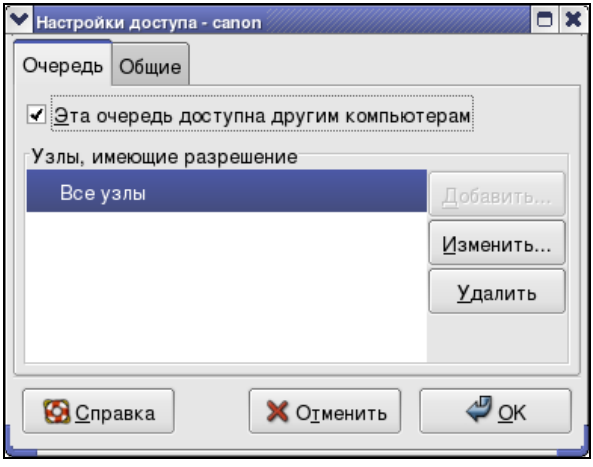

Рис. 7.13. Общий доступ к принтеру

Если вы хотите предоставить другим пользователям общий доступ к этому принтеру, нажмите кнопку **Доступ** и в появившемся окне установите флажок **Эта очередь доступна другим компьютерам** (рис. 7.13). При желании можно указать, каким компьютерам можно получать доступ к принтеру, а каким — нет.

7.3.2. Настройка принтера в Linux Mandrake

Для настройки принтера в Linux Mandrake используется конфигуратор drakprinter. После запуска конфигуратор сам определяет подключенные к компьютеру принтеры (в случае, если они не были еще установлены). Если конфигуратор найдет ваш принтер, вы увидите примерно следующее сообщение (рис. 7.14).

После этого вы увидите запись, соответствующую вашему принтеру, в окне конфигуратора (рис. 7.15).

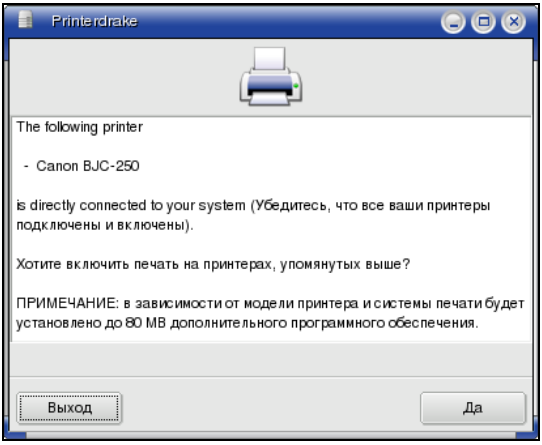

Рис. 7.14. Конфигуратор нашел принтер

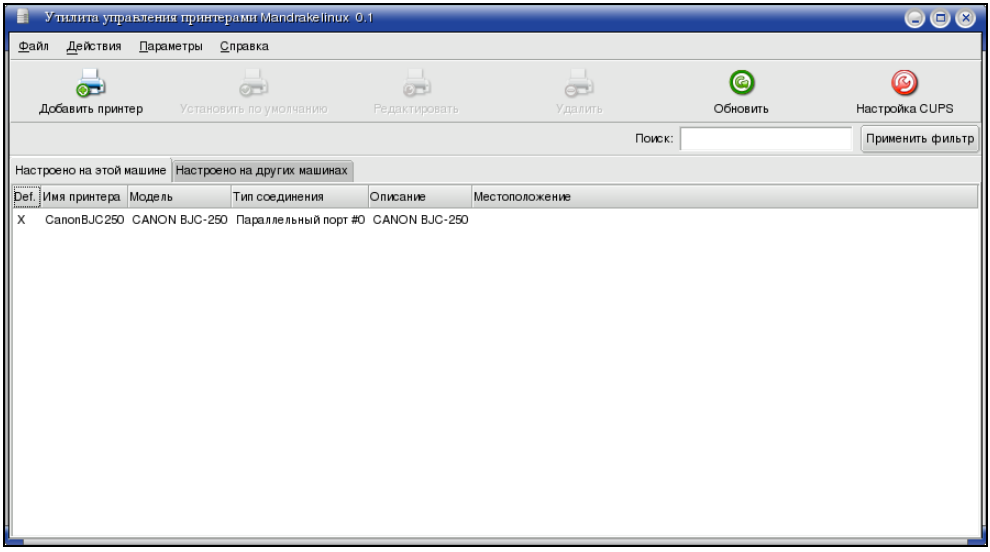

Рис. 7.15. Окно конфигуратора drakprinter

7.3.3. Что делать, если ваш принтер не поддерживается в Linux

Если у вас GDI-принтер, то не следует ожидать его поддержки в ближайшем будущем — просто используйте принтер в Windows или купите другой. Если же у вас нормальный аппаратный принтер, то его поддержка обязательно появится в следующей версии дистрибутива — просто разработчики дистрибутива пока не успели подготовить драйвер. Понимаю, что ждать выхода следующей версии не очень хочется — печатать нужно сейчас, а не через полгода. В этом случае попробуйте поискать информацию о своем принтере в Интернете — уверяю вас, что вы далеко не единственный обладатель такого принтера, и вполне возможно, что ваша проблема уже решена.

Глава 8

Сканеры и Linux

Первые вычислительные машины (компьютером сооружение весом в несколько тонн язык не поворачивается назвать) занимались в основном математическими расчетами. Не каждая фирма могла позволить себе такой компьютер.

С выпуском персоналок от IBM ситуация в корне изменилась: компьютеры появились во многих организациях, а еще через определенное время компьютеры пришли в дома пользователей. Сегодня домашним компьютером уже никого не удивишь, а лет 10 назад "персоналка" дома — это все равно, что сейчас плазменная панель во всю стену.

Современный домашний пользователь довольно избалован, он не представляет себе компьютер без обязательного набора периферии: принтера, сканера и модема. О принтерах мы уже говорили, о модемах еще поговорим, а вот в этой главе мы рассмотрим сканеры.

8.1. Немного истории

В далеком 1855 г. итальянец Казелли создал… первый сканер. Конечно, тогда эта машина называлась иначе — *пантелеграф*, но то, что она делала, весьма похоже на современный процесс сканирования. Работала она так: с помощью специальных электропроводящих чернил создавалось изображение, затем по изображению перемещалась игла и выполняла само сканирование. После этого полученная информация передавалась по телеграфу, где и происходил обратный процесс, т. е. печать принятой информации. Это схоже с тем, как если бы вы отсканировали какой-то документ, отправили своему знакомому по электронной почте, а он после получения распечатал документ. Подробно пантелеграф мы рассматривать не будем, поскольку речь не о нем. Отмечу только, что для работы этой машины нужны были специальные чернила. Если было нужно передать печатный документ (или рукописный), про-

блем никаких не возникало. Но вот как передать изображение, например, портрет? Тогда пантелеграфисту нужно было стать художником, чтобы нарисовать чье-то изображение, а потом отправить его с помощью пантелеграфа.

Со временем ситуация в корне изменилась — был изобретен фотоэлемент, и уже в самом начале XX в. известный ученый-оптик Корн изобрел фототелеграф. Качество сканирования было просто отличным, причем уже в то время Корн добился довольно большой разрешающей способности. Если в пантелеграфе сканирование выполняла игла, то в фототелеграфе — фотоэлемент, улавливающий отраженный свет (фотоприемник). Но, сами понимаете, это еще не предел, сканеры совершенствовались на протяжении всего XX в., и сейчас сканер — это не огромная машина, а довольно компактное устройство с очень низким энергопотреблением.

8.2. Принцип работы современных сканеров

Первые сканеры были ручные: линейку с фотоприемниками приходилось двигать вручную вдоль сканируемого объекта. Также были протяжные сканеры, нужно было двигать не линейку, а сам объект. Чуть позже появились автоматические планшетные сканеры, где планшет с фотоприемниками двигался автоматически. Несмотря на отличающуюся конструкцию, принцип у планшетных сканеров один: набор фотоэлементов двигается вдоль сканируемого изображения. Если изображение меньше области сканирования, то задействуется только часть фотоприемников.

Сегодня ручные и протяжные сканеры уже не выпускаются — их можно увидеть только в музеях вычислительной техники. Хотя история имеет свойство повторяться, поэтому не исключено, что через некоторое время мы вернемся… к ручным сканерам размером со спичечный коробок — достаточно будет провести его по листу бумаги, и изображение сканируемого объекта будет уже во flash-памяти сканера…

Кроме планшетных сканеров, есть еще и проекционные сканеры — они сканируют не объект, а его проекцию, что позволяет получить максимально возможное разрешение. Но такие сканеры — редкость.

Сами понимаете, сначала были черно-белые сканеры. Ясно, что проекционные сканеры только черно-белые, а вот планшетные вскоре "научились" передавать цвет. В этом случае сканирование осуществляется в три этапа. Сначала перед каждым фотоприемником выдвигается фильтр красного цвета. На втором проходе устанавливается фильтр зеленого цвета, а на третьем — синего. В некоторых случаях на планшете расположены сразу три фотоприем-

ника, каждый из которых принимает только свой цвет — красный, зеленый или синий.

Если быть более точным, то процесс сканирования состоит из следующих этапов:

1. Прогрев лампы сканера — на этом этапе прогревается лампа сканера, что есть необходимое условие для получения нужных характеристик сканирования — глубины и оптической плотности, о которых мы поговорим позже.
2. Позиционирование и калибровка сканера.
3. Сканирование изображения, принцип которого был изложен ранее.
4. Передача данных в компьютер.

8.3. Выбор сканера

Прежде чем приступить к выбору сканера, вам нужно знать, что такое *глубина цвета* и *оптическая плотность* — ведь это две самые важные характеристики сканера.

Для нормальной передачи цвета хватает глубины цвета в 24 бита — получается по 8 битов на каждую составляющую RGB. Но в продаже довольно часто встречаются сканеры с большей глубиной цвета, притом такие сканеры стоят относительно недорого. Зачем нужна глубина в 48 битов? Стоит ли переплачивать? Оказывается, что стоит. Повышение глубины цвета позволяет свести до минимума число потерь при аналого-цифровом преобразовании (АЦП), следовательно, качество отсканированной фотографии (для текста глубина цвета не очень важна) будет выше. Иногда встречаются сканеры с меньшей глубиной цвета, например, в 36 битов, но со специальным фильтром, уменьшающим шум АЦП. Такие сканеры обеспечивают иногда качество даже выше, чем их братья с большей глубиной цвета. Так, сканер с глубиной цвета в 36 битов и фильтром может сравняться по качеству с 48-битным сканером.

Чтобы понять, что такое оптическая плотность, нужно разбираться в оптике. Поскольку я не могу быть уверен в этом, то не буду пугать вас разными терминами из курса физики. Скажу только, что оптическая плотность влияет на распознавание оттенков. Запомните: чем выше оптическая плотность, тем лучше сканер различает оттенки.

Данная величина может принимать значения от 0,0 до 4,0. Дешевые сканеры обеспечивают плотность 2,0 (максимум 2,5). Более дорогие — от 3,0.

Вот теперь можно приступить к выбору сканера. Современные сканеры могут подключаться к компьютеру по LPT, USB или SCSI (самые новые модели

могут еще использовать WireFire и USB2) портам. Первый способ — классический, но он не очень удобный, если ваш принтер тоже подключается к LPT. Второй способ обеспечивает максимальный комфорт для домашнего пользователя: просто подключил сканер к USB, установил драйверы и можешь работать. Некоторым USB-сканерам даже не нужен блок питания — они питаются от USB. Это наиболее портативный вид сканеров, их можно подключать к ноутбуку, даже взять с собой в командировку, поскольку, как правило, они очень компактны. К тому же, USB-сканеры работают быстрее, чем LPT.

SCSI-сканеры стоят на порядок дороже USB-сканеров, к тому же, их можно подключить не к каждому компьютеру — для подключения такого сканера вам нужен SCSI-контроллер, который тоже недешево стоит. Поэтому мы ограничимся только LPT- и USB-сканерами.

Выбор сканера — дело непростое, тем более, что всегда есть соблазн купить дешевое "произведение искусства". Но не сильно обольщайтесь. Возьмите в руки прайс-лист и посмотрите, какой величины в характеристиках дешевых сканеров обычно не хватает (а в дорогих ее указывают). Да, я говорю об оптической плотности. В более дорогих сканерах плотность указывается. Например, у сканера Epson Perfection 1670 эта величина равна 3,1 d. Это довольно хорошее значение, такой сканер я бы купил. Но цена его в полтора-два раза дороже, чем самого дешевого. Сейчас объясню почему.

Смотрим дальше в прайс-лист. В характеристиках указанного сканера заявлена разрешающая способность 1600×3200 dpi (точек на дюйм), но там также указано, что это оптическая разрешающая способность. Для некоторых "средних" сканеров указана меньшая разрешающая способность, например, 1200×2400 dpi, но сказана, что она аппаратная (hardware). А с самыми дешевыми сканерами заявлена такая же разрешающая способность, но не сказана, какая она — оптическая или аппаратная. Аппаратную разрешающую способность обеспечивает само устройство, а оптическая "раздувается" оптикой.

Если разрешающая способность большая, например, 1600×3200 dpi, то она, как правило, оптическая. Но 1600×3200 — это довольно хороший показатель, поэтому производителям не стыдно написать на коробке "оптическая". Если у сканера аппаратная разрешающая способность 1200×2400 dpi, то это отлично. При этом производители гордятся своим детищем, и на коробке указывают, что данная способность является аппаратной. А вот если у сканера оптическое разрешение равно всего лишь 1200×2400 dpi, то ничего не пишут — из скромности.

Что же касается Linux, то вы можете покупать популярные сегодня USB-сканеры — главное, чтобы ваше ядро поддерживало USB. Поддержка USB

есть во всех современных дистрибутивах. Ради справедливости, нужно отметить, что поддержка USB была еще в ядрах 2.2 и 2.4, но там она не включалась по умолчанию, а для ее включения нужно было перекомпилировать ядро. Исключение могут составить только USB2-сканеры.

8.4. Установка сканера в Linux

Вы должны быть готовы к тому, что, возможно, ваша версия дистрибутива не поддерживает ваш сканер. Да, современные дистрибутивы поддерживают очень много различных устройств, в том числе и сканеры, как LPT, так и USB. Но при разработке модулей Linux большее внимание уделяется не периферийным устройствам, а основным. Это означает, что в большинстве случаев даже не самая новая версия дистрибутива будет поддерживать вашу видеоплату и сетевой адаптер, а вот сканер может и не работать даже в самой новой версии.

Вам не нужно сетовать на Linux — это проблема не операционной системы. Разработчикам драйверов для Linux нужно написать драйвер устройства зачастую по принципу черного ящика, то есть у них, кроме самого устройства и Linux, больше ничего нет. А это очень сложная задача, требующая много времени на решение. Вот и получается, что покупаешь самый современный дистрибутив, не самую новую версию устройства, а Linux ее не поддерживает.

8.4.1. Установка сканера в Linux Mandrake

Ради эксперимента было решено подключить к Linux не LPT-сканер, а более современный USB-сканер — Microtek ScanMaker 3840 (рис. 8.1). Правда, это тоже не ультрасовременная модель, но исходя из сказанного раньше, можно было предположить, что новинки компьютерного мира не всегда будут поддерживаться самым новым дистрибутивом Linux.

В дистрибутивах Linux Mandrake 10.1 и Fedora Core 3 выбранный сканер работать отказался. А с Mandrake 10 вообще произошла очень смешная ситуация. Для конфигурирования сканера в Mandrake используется конфигуратор scannerdrake (рис. 8.2). После его запуска было получено сообщение, что сканер не может быть определен автоматически. После чего программа предлагает выбрать производителя и модель сканера из списка. Нужная мне модель там есть, но если ее выбрать, scannerdrake сообщает, что выбранная модель не поддерживается данной версией дистрибутива (рис. 8.3).

Как оказалось, это был список "на вырост", то есть в него внесены устройства, поддержка которых будет встроена в будущие версии дистрибутива. Так оно и есть. В дистрибутиве Mandriva 2006 данная модель сканера поддерживается.

Рис. 8.1. Используемый сканер

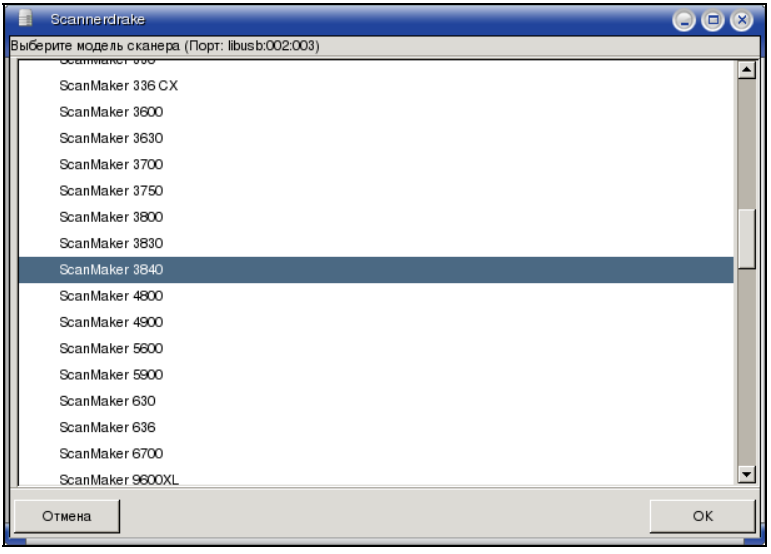

Рис. 8.2. Средство настройки сканера в Mandrake

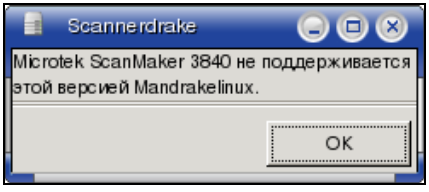

Рис. 8.3. Мой сканер не поддерживается в Mandrake

8.4.2. Установка сканера в ASP Linux 11

Дистрибутив ASP Linux 11 основан на дистрибутиве Fedora Core 4 (можно сказать, что они братья-близнецы), поэтому все, что будет работать в ASP Linux 11, будет работать и в Fedora Core 4 (и наоборот).

Убедитесь, что сканер подключен и включено его питание. После этого запустите конфигуратор hwbrowser. Если в списке системного оборудования найдете свой сканер, значит, он полностью поддерживается Linux (рис. 8.4).

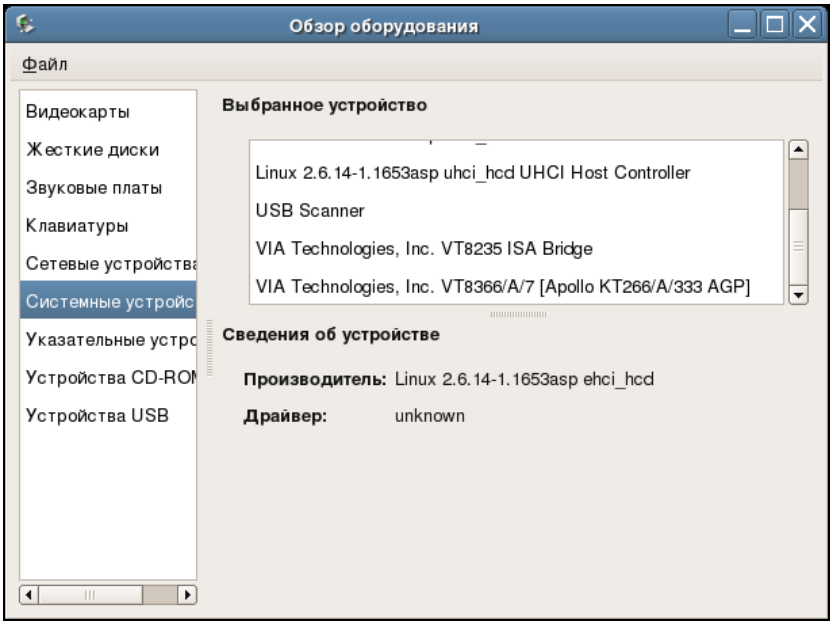

Рис. 8.4. Сканер поддерживается (программа hwbrowser)

8.5. Сканирование изображения

В Linux (ASP/Mandrake) со сканером могут работать следующие программы:

- xsane — используется для сканирования и первичной обработки изображений. Программа устанавливается по умолчанию, если, конечно, при установке вы не отключили группу программ для работы с графикой;
- gimp — сам gimp сканировать изображения не умеет, он для этого вызывает xsane, зато с полученным изображением можно будет сразу работать в gimp;

- kooka — данное приложение входит в состав KDE, поэтому может отсутствовать, если вы не выбрали установку KDE.

Если учитывать, что gimp для сканирования использует xsane, то у нас есть две полноценные программы для работы со сканером: xsane и kooka. Сначала мы рассмотрим xsane, а затем — kooka.

8.5.1. Программа xsane

Для полноценной работы с xsane вам нужно установить следующие пакеты:

- xsane — программа для сканирования изображения;
- sane-frontends — графический интерфейс для xsane;
- xsane-gimp — модуль поддержки сканера для gimp.

Как уже было отмечено, данные пакеты устанавливаются по умолчанию в большинстве случаев. Вам нужно только проверить, установлены ли они у вас. Это можно сделать с помощью команды:

```
# rpm -qa | grep sane
```

Если увидите все три пакета, значит, все нормально. Если чего-то не хватает, установите недостающие пакеты с помощью программы system-config-packages (в Mandrake — rpmdrake).

Для запуска xsane (рис. 8.5) можно выбрать соответствующий пункт меню GNOME/KDE или же ввести команду:

```
# xsane
```

Примечание

Если ваш сканер не поддерживается системой, при запуске xsane непременно сообщит вам об этом.

Если вы работаете как пользователь root, вы увидите сообщение о том, что работа с такими правами небезопасна — в данном случае не обращайте на него внимания.

Для начала сканирования перейдите в основное окно xsane, но не спешите нажимать кнопку **Сканировать**. Сначала нужно положить в сканер какой-нибудь документ и установить минимально необходимые параметры сканирования:

- **Разрешение** — по умолчанию используется разрешение 300 dpi. Такого разрешения хватит для получения довольно хорошего результата, чтобы впоследствии можно было распознать текст с полученного изображения. Если же планируется печать отсканированного документа (например, фо-

тографии), желательно выбрать бóльшее разрешение, например, 600 dpi. А если планируется только экранный просмотр документа, то вполне хватит 100–150 dpi. Помните, что от значения разрешения зависит размер отсканированного изображения: чем выше разрешение, тем больше размер файла.

- **Режим** — вы можете сканировать в цвете или же выбрать черно/белый режим сканирования. От выбора режима также зависит размер файла — черно-белые изображения занимают намного меньше места, чем цветные.

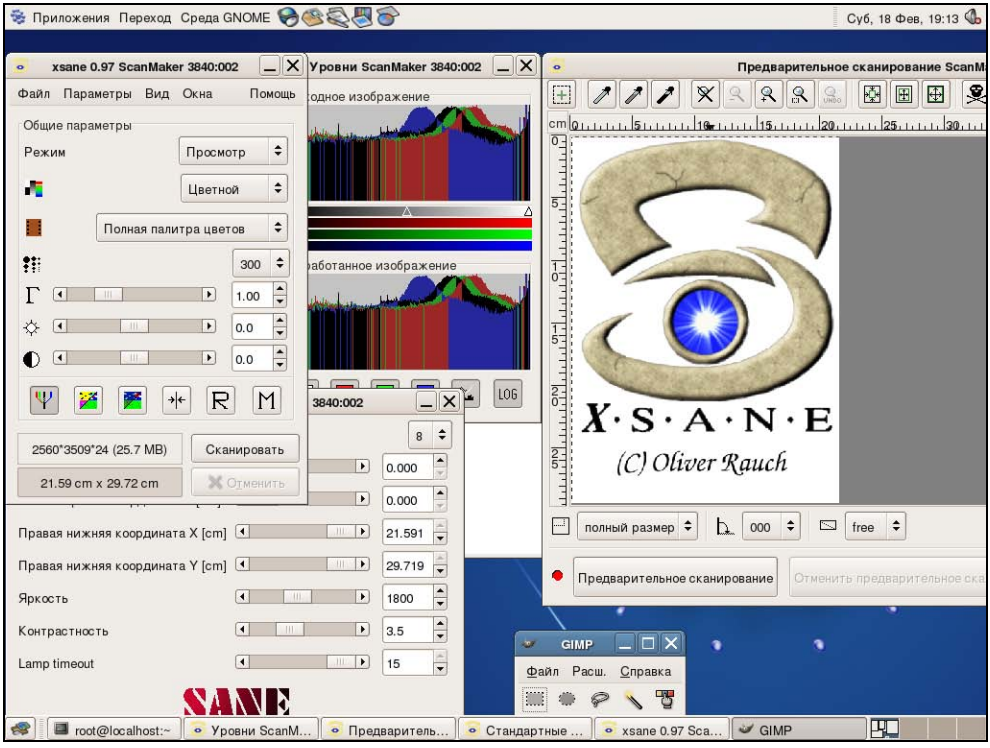

Рис. 8.5. Первый запуск xsane

В зависимости от выбранных вами параметров, xsane пытается вычислить приблизительный размер файла изображения. В данном случае размер будущего файла — 6,4 Мбайт, конечно, если будете использовать формат без сжатия, вроде BMP или TIFF. Для сохранения файла вы можете выбрать формат JPEG, GIF или TIFF со сжатием — размер файла будет значительно меньше, но часть информации о цвете будет потеряна (рис. 8.6).

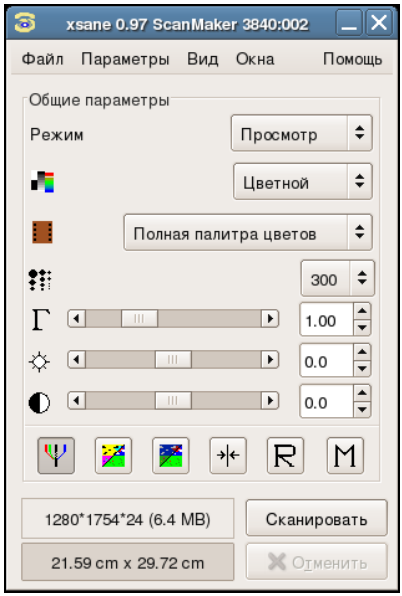

Рис. 8.6. Основное окно xsane

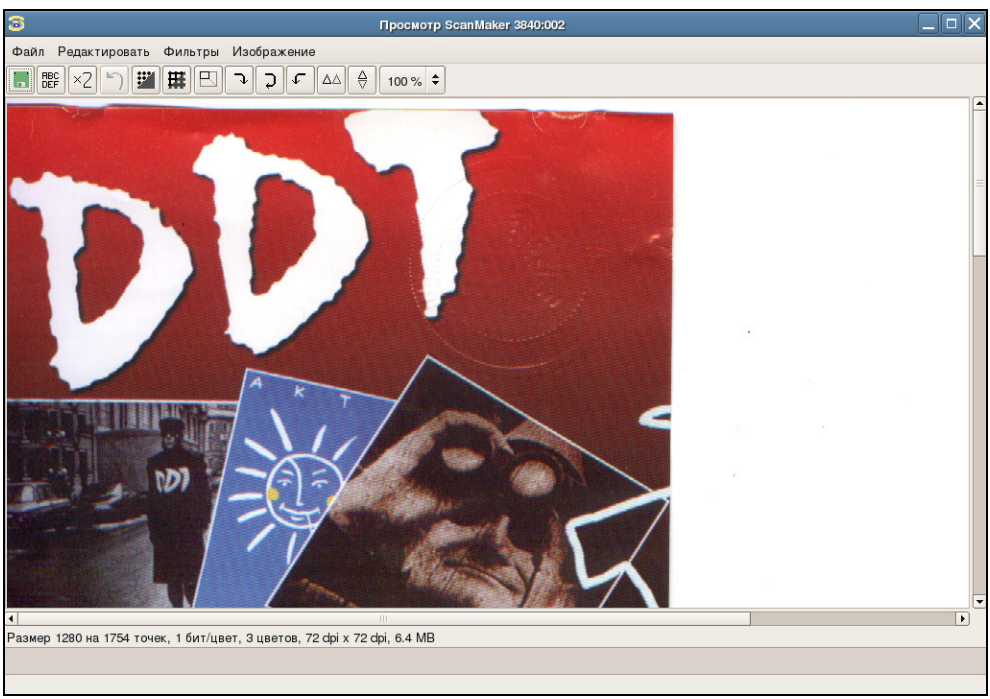

Рис. 8.7. Отсканированное изображение

Отсканированное изображение появится в отдельном окне (рис. 8.7).

Если вас не устраивает качество полученного изображения, можете поэкспериментировать с параметрами в окне параметров xsane (рис. 8.8).

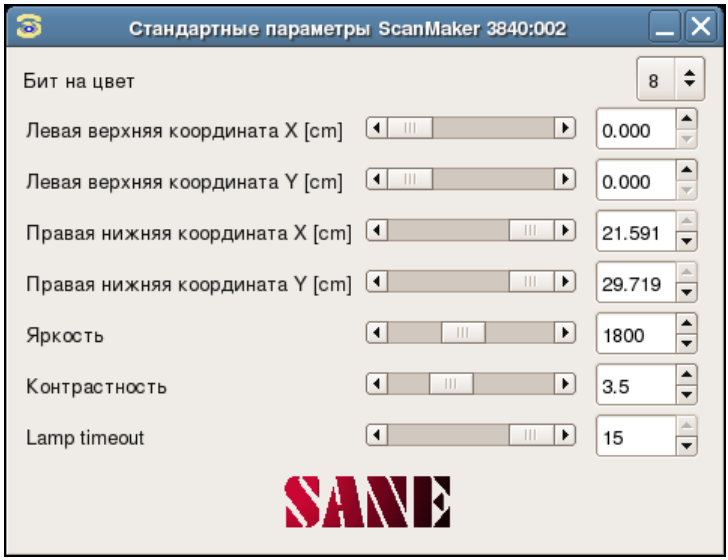

Рис. 8.8. Окно параметров xsane

Полученное изображение с помощью команды **Файл, Сохранить** можно сохранить в следующих форматах:

- JPEG;
- PNG;
- PNM;
- PostScript;
- TIFF.

8.5.2. Сканирование изображения в gimp

Для сканирования изображения в gimp выберите команду **Файл, Захватить, xsane:имя_сканера**. После этого gimp запустит программу xsane, которая отсканирует изображение и передаст его обратно в gimp. После этого с изображением можно работать как обычно.

8.5.3. Программа kooka

При первом запуске kooka попросит вас выбрать используемый вами сканер (рис. 8.9).

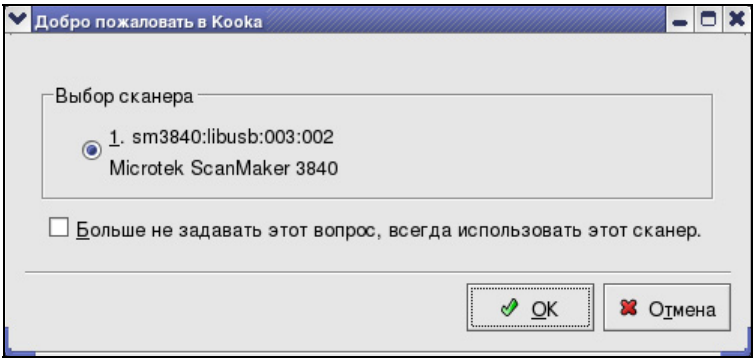

Рис. 8.9. Выбор сканера

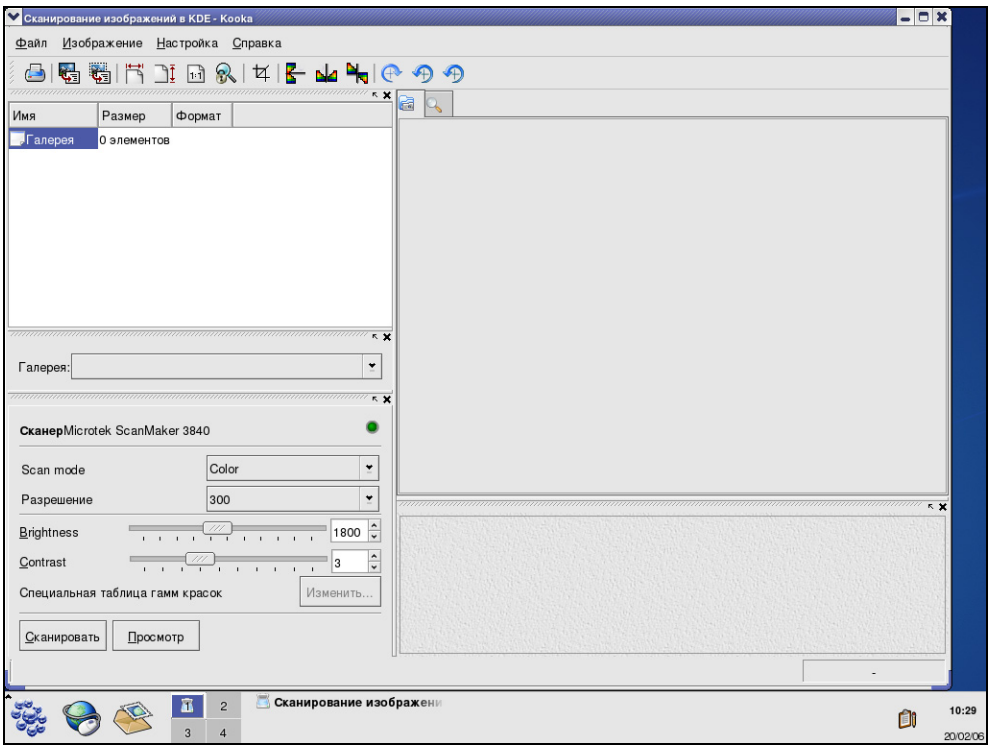

Рис. 8.10. Основное окно kooka

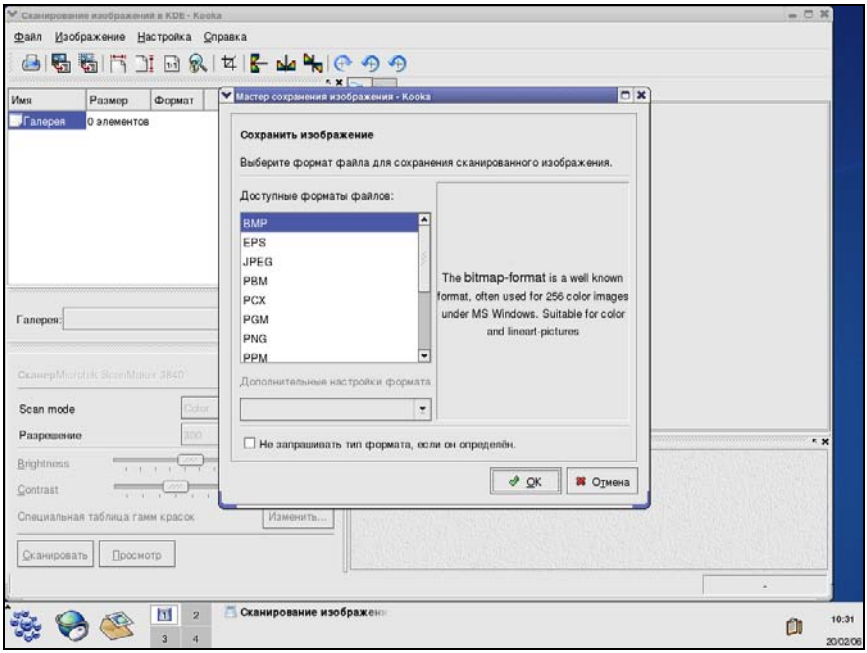

Рис. 8.11. Выбор формата изображения

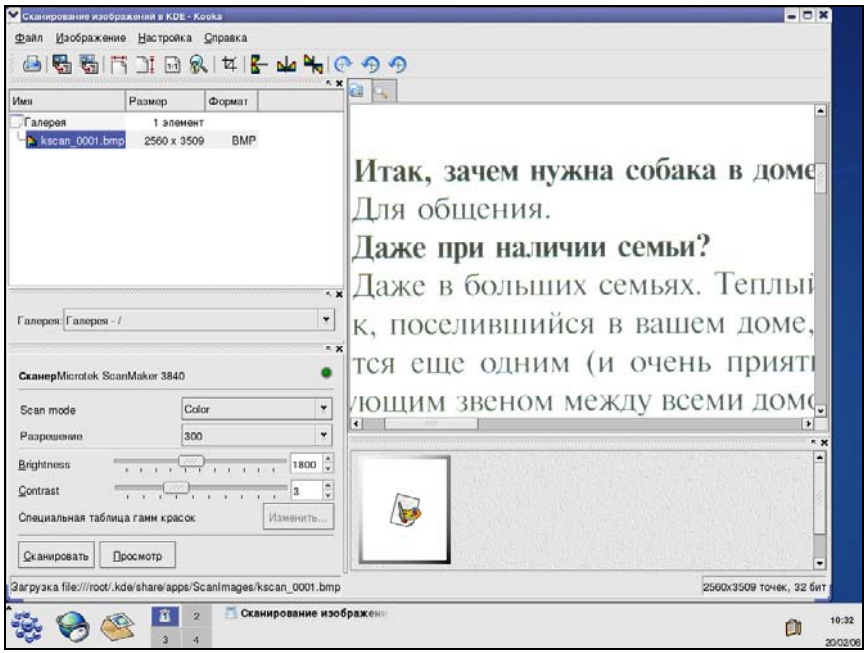

Рис. 8.12. Отсканированное изображение

Данное окно появится, даже если у вас всего один сканер. После этого появится основное окно программы, в нижней части которого можно установить параметры сканирования: режим сканирования, разрешение, яркость, контрастность (рис. 8.10).

Установите нужные вам параметры сканирования и нажмите кнопку **Сканировать**. После того как изображение будет полностью получено от сканера, kooka предоставит вам возможность выбрать формат изображения (рис. 8.11).

После выбора формата вам предстоит ввести имя файла. Все: изображение отсканировано и сохранено. Вы сможете просмотреть его в рабочей области (рис. 8.12).

8.6. Распознавание текста

В домашних целях сканер используется обычно для сканирования фотографий и документов (это громко сказано — как правило, в роли "документов" выступают книги, страницы которых нужно сначала отсканировать, а потом распознать). От фотографий в электронной форме толку мало — их можно и в альбоме посмотреть, а вот сканирование документов — это важно. Электронная копия документа в виде изображения не очень полезна — нужно еще и распознать текст.

В Linux для распознавания текста используются две программы — ocrad и gorc, которые иногда входят в состав дистрибутива. Если этих программ в вашем дистрибутиве не оказалось, вы можете найти RPM-пакеты, содержащие эти программы, с помощью сайта **http://rpmfind.net**. Первая программа используется по умолчанию как OCR-система в программе kooka, а вторая — в программе xsane. Но толку от этих программ будет мало — они умеют распознавать только англоязычные тексты.

Для распознавания текстов на русском языке есть только пока одна программа — ABBY FineReader 6.0 for Linux, но она, к сожалению, не бесплатная. На сайте **http://www.abbyy.ru/** можно скачать демонстрационную версию, работающую 60 дней с момента установки.

Глава 9

Модемы

9.1. Общие сведения о модеме

9.1.1. Что такое модем

Модем — это устройство, выполняющее модуляцию и демодуляцию (отсюда и название — модем) дискретных сигналов. Модем кодирует и декодирует каждый информационный бит, синхронизирует передачу сигналов по линиям связи (обычно — телефонным линиям), выполняет проверку правильности передачи, а также выполняет некоторые другие операции, например, компрессию и декомпрессию передаваемых данных.

Спрашивается, а зачем нужна модуляция и демодуляция сигналов? Дело в том, что модемы предназначены для передачи данных по аналоговым (не цифровым) линиям данных. Ведь компьютер работает с данными в двоичной системе, то есть компьютер оперирует наборами нулей и единичек. В свою очередь, нули и единички кодируются с помощью напряжения, например, есть напряжение — 1, нет напряжения — 0. Если бы линия, по которой мы передаем данные, была цифровой, то можно было бы отправить данные безо всякой модуляции, а просто используя обычную для компьютера схему.

Но что делать, если вам нужно передать сигнал за пределы компьютера? Если расстояние небольшое, то можно было бы соединить компьютеры напрямую, например, используя так называемый нуль-модем (специальный кабель, соединяющий два компьютера по COM или LPT порту) или с помощью перекрестно обжатого кабеля витой пары. А если расстояние довольно велико, скажем, другой компьютер находится в другом конце района или даже в другом городе? Первое, что приходит в голову — купить много кабеля. Но это может оказаться довольно дорогой затеей, к тому же, вам придется бороться не только с затуханием сигнала (кроме кабеля для передачи данных на большие

расстояния, вам нужны будут усилители сигнала), но и с любителями поживиться за ваш счет — кабель могут просто украсть вместе с усилителями и другим оборудованием.

Вот тут действительно понимаешь, что самое дешевое решение — это передача данных по обычным телефонным линиям. Телефон есть у всех (во всяком случае, почти у всех). Модем стоит довольно дешево. Подключаешь модем к компьютеру и к телефонной линии — и все, можешь передавать данные.

Но, как всегда, есть и обратная сторона медали. Достоинство у передачи данных с помощью модема только одно — простота и дешевизна при установке.

На самом деле модемное соединение дешево только при установке. В эксплуатации коммутируемое соединение с Интернетом обходится еще дороже, чем выделенная линия, поскольку вам нужно платить за использование телефона вашему оператору связи, а также интернет-провайдеру — за Интернет. А учитывая небольшую скорость соединения, большое количество данных вам все равно не удастся передать. В то же время вы можете купить комплект оборудования для радио-доступа к провайдеру, обеспечивающее значительно большую скорость соединения, чем модем.

Вернемся к модемам и телефонным линиям. Не секрет, что телефонные линии у нас далеки от совершенства. Даже из очень хорошей (не цифровой) аналоговой телефонной линии вам вряд ли удастся "выжать" скорость более 6–7 Кбайт/с. Я специально не указываю скорость в бодах или в килобитах, поскольку скорость, на которой модем соединился с провайдером (и которую вы видите в информационном окошке), не всегда соответствует реальной скорости передачи данных. Средняя же скорость при использовании модема, работающего на телефонной линии среднего качества, приблизительно равна 2–2,5 Кбайт/с. Кроме того, краски сгущают постоянные обрывы соединения, которые на отечественных телефонных линиях происходят довольно часто. А все это из-за зашумленности телефонной линии. В этой главе (но чуть позже) мы поговорим о том, как настроить модем для относительно устойчивой работы даже на плохой линии. Все происходит потому, что телефонные сети изначально предназначены для передачи голоса, а не цифровых данных.

Основная задача модема — преобразовать поток нулей и единиц в звуковой сигнал, который бы был максимально похож на голос. А это очень непростая задача. Можно сказать, что модем — это маленький компьютер: у модема есть свой процессор, своя память, а также другие микросхемы, необходимые ему для нормальной работы. Кроме этого, современные модемы выполняют функции обычного телефона — "поднимают" трубку, набирают номер и т. д. Конечно, это звучит дико, но первые модемы не были автоматическими, то есть поднять трубку и набрать номер должен был пользователь с помощью обычного телефона.

9.1.2. Стандарты обмена данными

Итак, мы уже знаем, для чего используется модем. Теперь давайте разберемся в популярных стандартах модемов. Ведь руководство по модему (как и его упаковка) пестрит всякими V.*, а что это означает, не сказано. Следующая табл. 9.1 поможет вам сориентироваться в мире модемных стандартов.

Таблица 9.1. *Стандарты обмена данными*

Стандарт	Скорость, бод	Описание
V.21	300	Полнодуплексная (в обоих направлениях) низкоскоростная передача данных
V.22	1200	Полудуплексная (в одном направлении) низкоскоростная передача данных
V.22bis	1200, 2400	Полнодуплексная передача данных. Использовался во всем мире, в отличие от первых двух стандартов, которые не использовались в США
V.29	1200, 2400, 4800, 9600	Полудуплексная передача данных. Используется преимущественно факс-модемами
V.32	4800, 9600	Полнодуплексная передача данных. Модемы, поддерживающие данный стандарт, могут самостоятельно выбирать скорость в зависимости от качества линии
V.32bis	7200, 12000, 14400	Версия V.32, поддерживающая более высокие скорости
V.34	до 28800	Следующее усовершенствование V.32
V.42	–	Стандарт коррекции ошибок для высокоскоростных модемов, например, для V.32bis или V.34-модемов
V.42bis	34000	Стандарт коррекции и сжатия данных. Благодаря сжатию достигается скорость в 34000 бод
V.90	56000/33600	Асинхронный стандарт высокоскоростной передачи данных: данные отправляются со скоростью 56Кбит/с, а принимаются — со скоростью 33,6 Кбит/с. V.90-модемы умеют распознавать цифровую линию, поэтому снижается число преобразования данных (ведь если модем подключен к цифровой линии, модуляция и демодуляция уже вообще не нужна), а это позволяет повысить скорость передачи данных
V.92	56000/48000	Усовершенствованная версия V.90, позволяющая передавать данные со скоростью 48 Кбит/с. Также возросла скорость передачи факса — 33,6 Кбит/с, вместо 14,4 Кбит/с
V.44	–	Стандарт коррекции ошибок, использующийся вместе с V.92

Как вы видите, в таблице приведены стандарты обмена данными и стандарты сжатия данных. Но для модемов определена еще одна группа стандартов — стандарты на команды модема. В данной группе стандартом де-факто является стандарт, разработанный компанией Hayes — Hayes AT.

9.1.3. Стандарт Hayes AT

Модемы, поддерживающие стандарт команд Hayes AT, называют AT-совместимыми (AT-compatible). Каждая команда данного стандарта начинается символами AT, например, ATH1. Вот некоторые самые распространенные AT-команды:

- ATZ — сброс модема. Все параметры модема устанавливаются по умолчанию. Обычно данная команда используется в качестве строки инициализации модема;
- ATH — позволяет поднять/положить трубку (ATH1 — поднять, ATH0 — положить);
- ATD — команда набора номера. Обычно после нее указывает модификатор (например, T — для тонового набора или P — для импульсного) и номер телефона, например, ATDP5551122;
- ATSr=n — команда для записи в регистр r значения n. В регистрах модема хранятся его параметры. Изменяя значения регистров, вы изменяете аппаратные параметры модема. Набор регистров зависит от вашего модема. Как правило, описание регистров модема приводится в документации по нему.

9.1.4. Выбор модема

В компьютерном магазине обычно предлагают довольно хороший ассортимент модемов — от самых дешевых устройств до довольно дорогих от известных брендов, вроде ZyXEL и US.Robotics. Но модем модему рознь. Давайте попробуем выбрать оптимальный для вас модем.

Самые дешевые — внутренние модемы. Они представляют собою плату расширения, устанавливаемую вовнутрь корпуса компьютера (рис. 9.1). Как правило, все современные внутренние модемы выполнены в виде PCI-платы, шансов нарваться на старый внутренний модем, подключаемый к шине ISA, практически нет. Да и у кого до сих пор осталась эта шина?

Настоятельно рекомендуется отказаться от покупки внутреннего модема по следующим причинам:

- В случае "зависания" модема (а такое случается, особенно с дешевыми модемами) вы не сможете сбросить его. Если такое произойдет с внешним

модемом, то для перезапуска достаточно включить/выключить модем. А внутренний модем вы уже не выключите: тут или ждать, пока модем "подумает", или перезагружать компьютер.

- Вы не сможете контролировать состояние модема — у внутреннего модема нет никаких индикаторов, отображающих состояние модема. Помню, раньше выпускались внутренние модемы, к которым подключалась панель с индикаторами состояния. Данную панель можно было установить в отсек корпуса (5,25 или 3,5), сняв заглушку.

Рис. 9.1. Внутренний модем Genius GM56PCI-L

Но приведенные причины, сами понимаете, не являются основными. А теперь поговорим об основной. Помните, я говорил, что у модема есть свой процессор, а также своя собственная память? Так вот, не у каждого модема есть процессор. Модемы с собственным процессором называются аппаратными, а модемы без такового — программными (они Win-модемы). Тут точно такая же ситуация, как и с принтерами, помните, мы рассматривали так называемые GDI-принтеры (Win-принтеры)? В программных модемах вместо модема "думает" центральный процессор компьютера, но чтобы заставить

процессор "думать", нужна программа-драйвер, которую разработчики модема поленились выпустить для операционной системы Linux. Следовательно, Win-модемы работать в Linux не будут. Если вы уж решились на покупку внутреннего модема, то покупайте только аппаратный внутренний модем. Но должен заметить, что хороший аппаратный внутренний модем стоит столько же (а иногда и больше), сколько внешний модем. К тому же, нужно отметить, что Linux вообще не дружит с внутренними модемами. Чтобы заставить заработать аппаратный (внутренний) модем, вам придется немного потрудиться. А вот с внешними проблем никаких нет — подключаете к компьютеру, Linux его определяет, создает ссылку /dev/modem и все — вы можете звонить своему провайдеру.

Хотя опять-таки, даже и с внешними модемами есть небольшие проблемы. В большинстве случаев внешний модем (рис. 9.2) является аппаратным модемом. Внешние модемы могут подключаться к компьютеру через COM-порт или по USB. Лучше выбрать COM-модем — поверьте, у вас не будет никаких проблем с подключением этого устройства и его настройкой в Linux. Не подумайте, что Linux не поддерживает USB-модемы, просто в Linux есть драйверы не для всех USB-модемов, а вот на диске с драйверами модема вряд ли будут драйверы для Linux.

Рис. 9.2. Внешний модем ZyXEL Omni 56K NEO Ext Retail

Итак, мы определились с типом модема — только внешний и, по возможности, COM. Теперь будем определяться с ценой. Как правило, чем дороже модем, тем лучше его "проходимость". Например, относительно дорогие модемы

фирмы ZyXEL могут работать практически на любых линиях. Если же вам нужен настоящий "вездеход" (например, у вас очень плохая телефонная линия), купите ZyXEL U336E — этот модем будет работать на любой телефонной линии. Можете поискать и бывший в употреблении модем (именно ZyXEL 336U).

9.2. Подключение модема

Сначала нужно подключить модем к компьютеру. Если у вас COM-модем, то перед его подключением выключите компьютер. Если же вы купили USB-модем, компьютер можете не выключать.

После этого подключите модем к телефонной сети. Обратите внимание: на задней панели модема (вне зависимости, какой он — внутренний или внешний) два гнезда с надписями LINE и PHONE. Телефонную линию нужно подключать к гнезду LINE — не перепутайте! К гнезду PHONE можно подключить параллельный телефон, если есть такая необходимость.

В Linux файл устройства модема будет называться /dev/ttySn, если вы используете COM-модем, или /dev/ttyUSBn, если у вас USB-модем, n — это порядковый номер устройства. Для COM-модемов он зависит от того, к какому порту подключен модем:

- /dev/ttyS0 — к COM1;
- /dev/ttyS1 — к COM2 и т. д.

Для USB-модемов — это просто порядковый номер модема. Забегая вперед, скажу, что при настройке USB-устройств вы можете использовать утилиту usbview, выводящую информацию обо всех USB-устройствах.

9.3. Установка модема в Linux и подключение к Интернету

Для соединения с Интернетом вам нужно выполнить следующие действия:

1. Программно (ведь физически он уже подключен) установить и настроить модем.
2. Настроить соединение с провайдером.

Для соединения с провайдером по модемной линии могут использоваться два протокола SLIP (Serial Line Internet Protocol) и PPP (Point-to-Point Protocol). Первый уже не используется, поэтому мы будем рассматривать только PPP-соединения.

За установку PPP-соединения и передачу данных отвечает демон pppd. Он запускается не автоматически при запуске системы, а только тогда, когда нужно — когда вы захотите подключиться к Интернету. Все рассматриваемые в этой главе программы представляют собою просто оболочки для этого демона, занимающиеся сбором параметров и позволяющие пользователю удобно редактировать все параметры PPP-соединения. Пользователю намного проще ввести параметры соединения в графическом окошке и нажать кнопку **Соединить**, чем каждый раз для установки соединения вводить команду вроде:

```
/usr/sbin/pppd debug lock modem crtscts /dev/ttyS0 57600 \
      asyncmap 20A0000 escape FF kdebug 0 $LOCAL_IP:$REMOTE_IP \
      noipdefault netmask $NETMASK defaultroute connect $DIALER
```

9.3.1. Установка модема и создание PPP-соединения в ASP Linux

Настроить соединение с Интернетом можно двумя способами:

- Создать интерфейс PPP-соединения (pppN), с которым можно будет работать как с обычным интерфейсом — активизировать, деактивизировать или активизировать при загрузке системы.
- Пойти по пути минимального сопротивления и сразу перейти к настройке kppp — вся информация о соединениях будет доступна только в программе kppp.

Для домашнего пользователя больше подходит второй способ, но в этой книге мы рассмотрим оба способа: сначала первый способ (для общего развития), а второй — в следующем пункте.

Запустите мастер создания новых сетевых подключений командой:

```
# system-config-network-druid
```

В появившемся окне выберите **Модемное соединение** (рис. 9.3).

Если это первый запуск мастера соединений, то он предоставит вам выбор устройства (рис. 9.4) и возможность установки его параметров. При последующих вызовах мастер перейдет сразу к следующему шагу — вводу номера телефона провайдера, имени пользователя и пароля (рис. 9.5). Это связано с тем, что нужно создать ссылку /dev/modem, которая может указывать только на одно устройство. То есть если вы сейчас используете мастер для одного конкретного модема, то с его помощью вы больше не сможете создать модемное соединение для другого модема. С одной стороны, это недостаток, а с другой — преимущество. Ведь обычно у пользователя один модем, а PPP-

соединений может быть несколько (к разным провайдерам), поэтому пользователю при создании новых PPP-соединений уже не придется устанавливать параметры устройства.

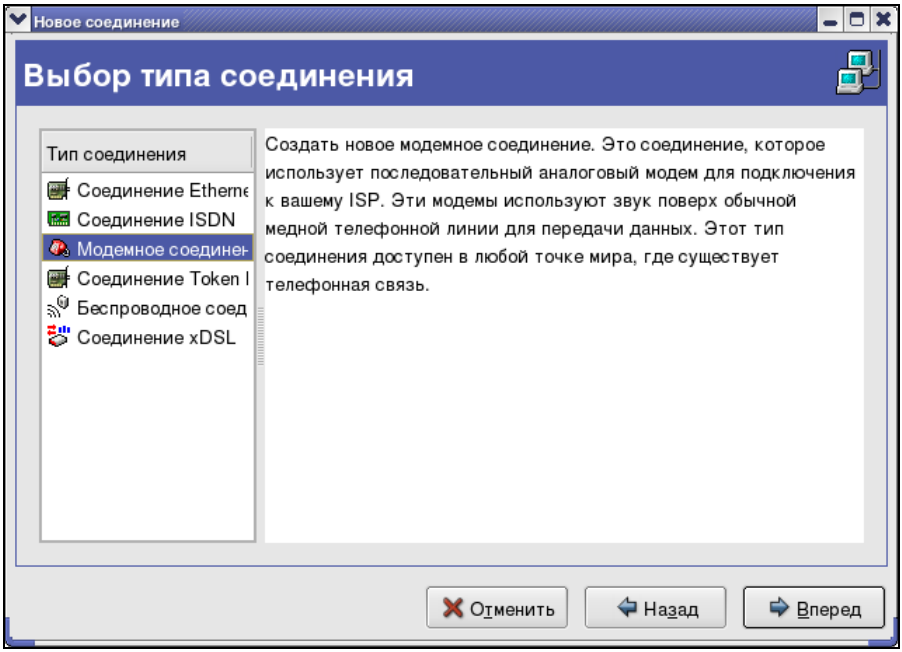

Рис. 9.3. Создание модемного соединения

Если же у вас несколько модемов — не расстраивайтесь. В программе kppp, которая используется для соединения с Интернетом, вы можете указать файл устройства вручную, не используя ссылку /dev/modem. Ссылка /dev/modem используется сугубо для комфорта пользователя. Программ для работы с модемом может быть несколько, и при настройке каждой из них намного проще указать /dev/modem, чем /dev/ttyS0 или /dev/ttyUSB0.

Вы можете установить следующие параметры:

- **Устройство модема** — если у вас COM-модем, подключенный к порту COM1, имя устройства будет /dev/ttyS0, а если у вас USB-модем, то, скорее всего, имя файла будет /dev/ttyUSB0;
- **Скорость порта** — особой роли этот параметр не играет, поскольку скорость соединения может изменяться в процессе самого соединения в зависимости от внешних факторов: скорости порта на другом конце линии, зашумленности линии и т. д;

Новое соединение

Выбор модема

Свойства модема

Устройство модема: /dev/ttyS0

Скорость порта: 115200

Управление потоком: Аппаратное (CRTSCTS)

Громкость модема: Средняя

Использовать тональный набор

Отменить Назад Вперед

Рис. 9.4. Установка параметров устройства

Новое соединение

Выбор провайдера

Провайдер услуг Инт

Austria
Czech Republic
Germany
Slovenia
United Kingdom

Номер телефона

Префикс: Код города: Номер телефона: 5557755

Имя провайдера: ISP

Настройка учетной записи T-Online

Учетное имя: user

Пароль: ********

Отменить Назад Вперед

Рис. 9.5. Параметры дозвона к провайдеру

- **Управление потоком** — как правило, аппаратное;
- **Громкость модема** — по умолчанию динамик выключен, но рекомендуется установить среднее значение громкости — так вы уже по звуку будете определять, соединиться модем или нет;
- **Использовать тональный набор** — выключите этот режим, если у вас импульсная АТС.

Следующий шаг — это параметры дозвона к провайдеру. Вы должны указать номер телефона провайдера, а также имя пользователя и пароль (рис. 9.5).

После этого вам нужно указать параметры протокола IP, а именно ваш IP-адрес, адрес шлюза и маску сети. Обычно настройка протокола IP выполняется автоматически с помощью протокола динамического конфигурирования узла (DHCP, Dynamic Host Configuration Protocol), поэтому вам не нужно ничего вводить, а только убедиться, что установлен автоматический режим настройки (рис. 9.6). Хотя, на всякий случай, параметры IP вам лучше уточнить у своего провайдера.

Последний шаг — вам нужно подтвердить введенную вами информацию. Если все правильно, нажмите кнопку **Применить** (рис. 9.7).

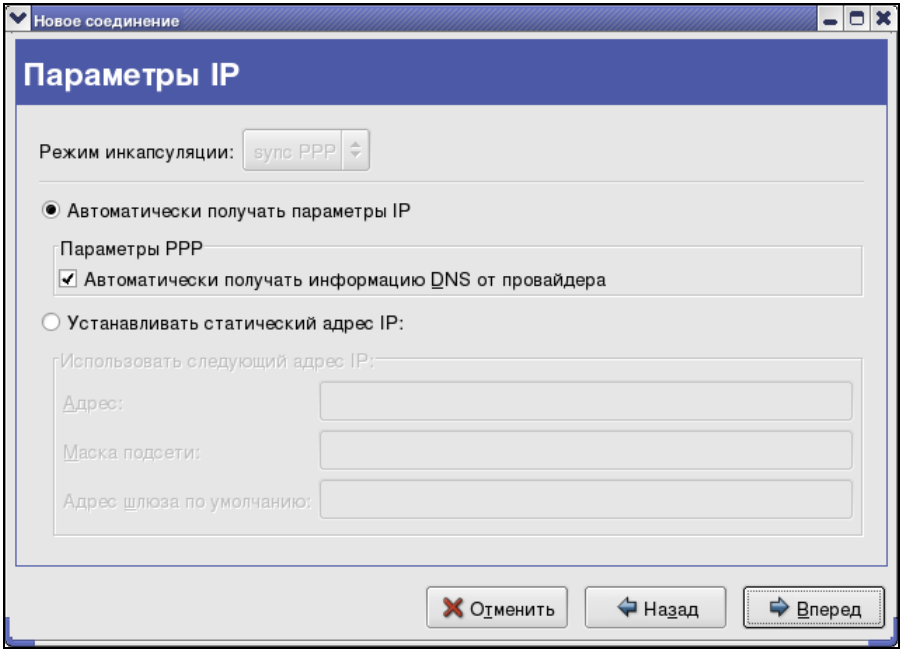

Рис. 9.6. Параметры протокола IP

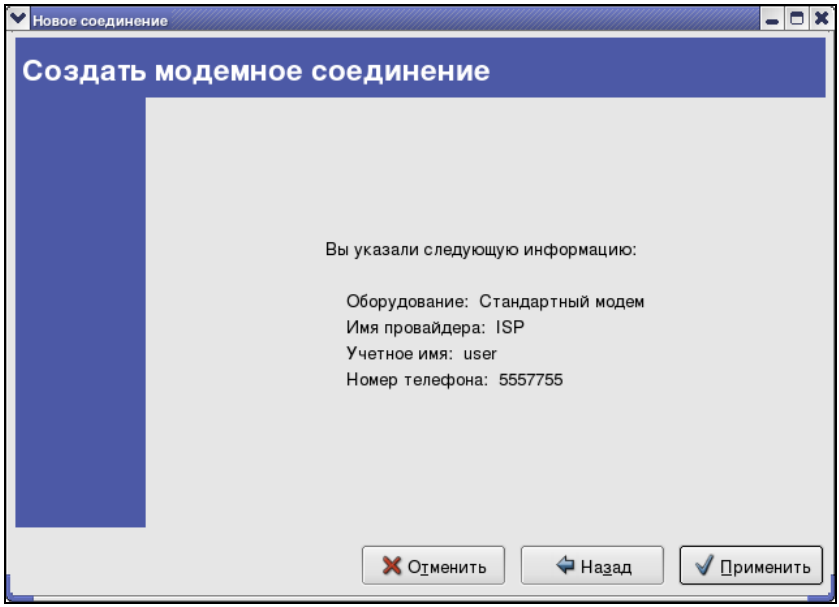

Рис. 9.7. Подтверждение введенной вами информации

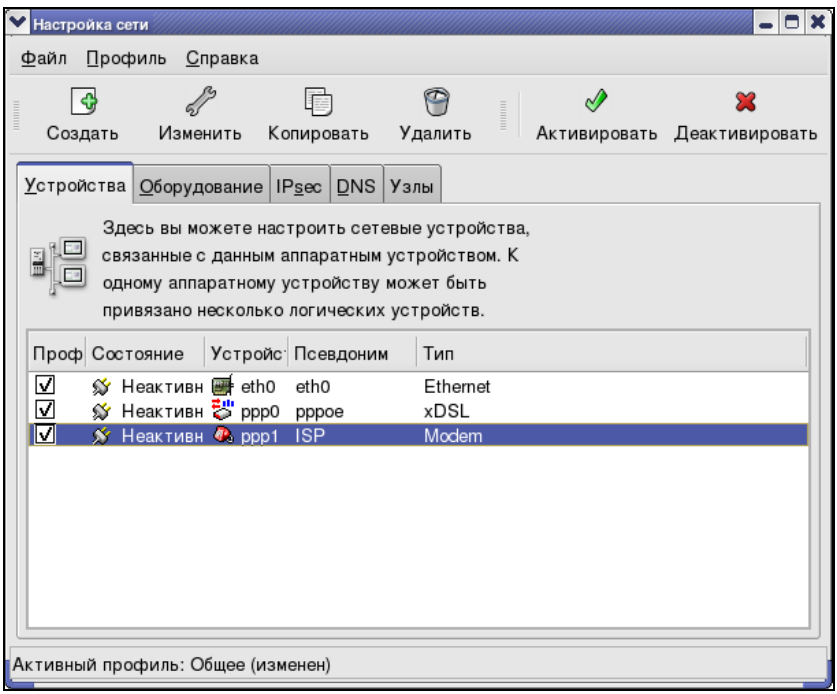

Рис. 9.8. Конфигуратор сети

Как только закроется окно мастера, вы увидите окно конфигуратора сети, в котором будут перечислены все настроенные сетевые интерфейсы (рис. 9.8).

Конфигуратор сети позволяет изменить параметры сетевого интерфейса, активировать и деактивировать любой интерфейс. При активации интерфейса устанавливается соответствующее интерфейсу сетевое соединение, при деактивации выполняется обратное действие — соединение разрывается. В нашем случае для создания PPP-соединения с провайдером (для подключения к Интернету) вам нужно активировать интерфейс ppp1 (это модемное соединение, см. рис. 9.8). Для отключения — деактивировать этот интерфейс.

Если вам нужно изменить параметры какого-либо интерфейса в процессе работы, вы можете вызвать конфигуратор сети без вызова мастера создания соединения. Для этого используется команда:

```
# system-config-network
```

Подробнее о конфигураторе сети мы поговорим в следующей главе.

9.3.2. Установка модема и создание PPP-соединения в Mandrake

В Linux Mandrake для создания нового соединения используется конфигуратор drakconnect. Запустите его и выберите создание нового модемного соединения (рис. 9.9).

После этого вам нужно выбрать ваш модем (рис. 9.10). Если вашего модема нет в списке, убедитесь, что он подключен к компьютеру и включен. Если эти два условия выполнены, а модема в списке нет, можно попытаться использовать опцию **Самостоятельный выбор**, но, скорее всего, ваш модем не поддерживается Linux, и настроить его вы не сможете.

Следующий шаг — это выбор провайдера. Вашего провайдера точно не будет в списке, поэтому, чтобы не тратить время, сразу нажмите кнопку **Далее**. Конфигуратор drakconnect попросит вас ввести следующие параметры соединения (рис. 9.11):

- **Название соединения** — только для вашего удобства, особой роли не играет;
- **Номер телефона** — номер телефона модемного пула вашего провайдера;
- **ID логина** — ваш логин;
- **Пароль** — ваш пароль;
- **Аутентификация** — тип аутентификации, оставьте **PAP/CHAP**, чтобы система сама выбрала используемый метод аутентификации.

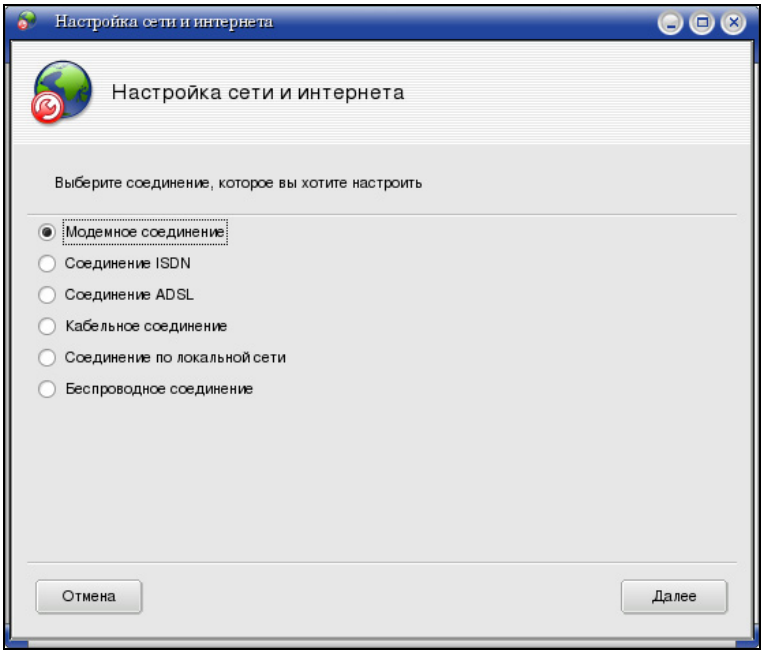

Рис. 9.9. Создание модемного соединения в Linux Mandrake

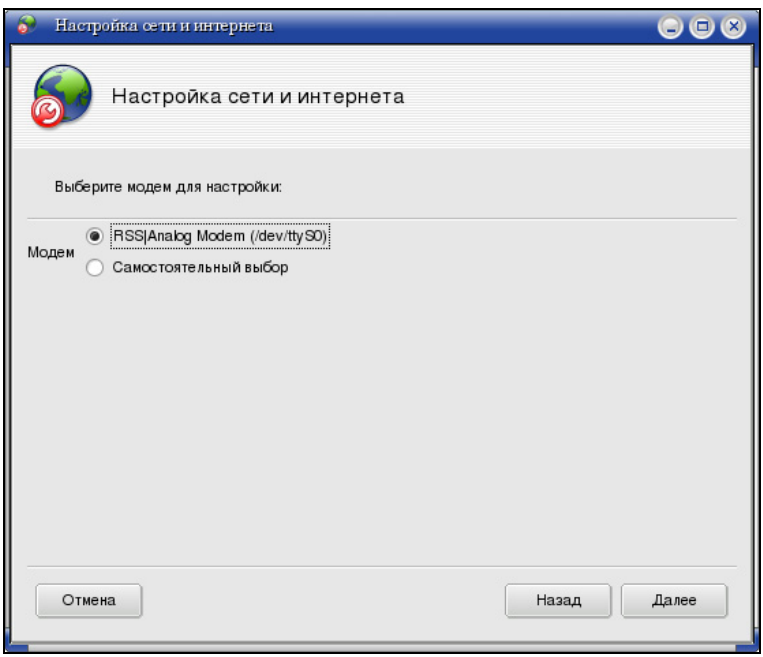

Рис. 9.10. Выбор модема

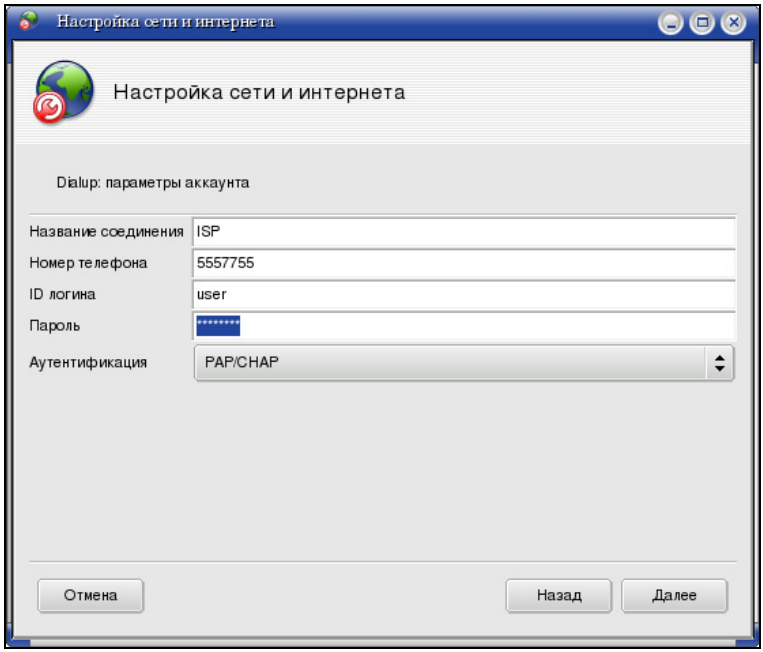

Рис. 9.11. Параметры коммутируемого (dialup) соединения

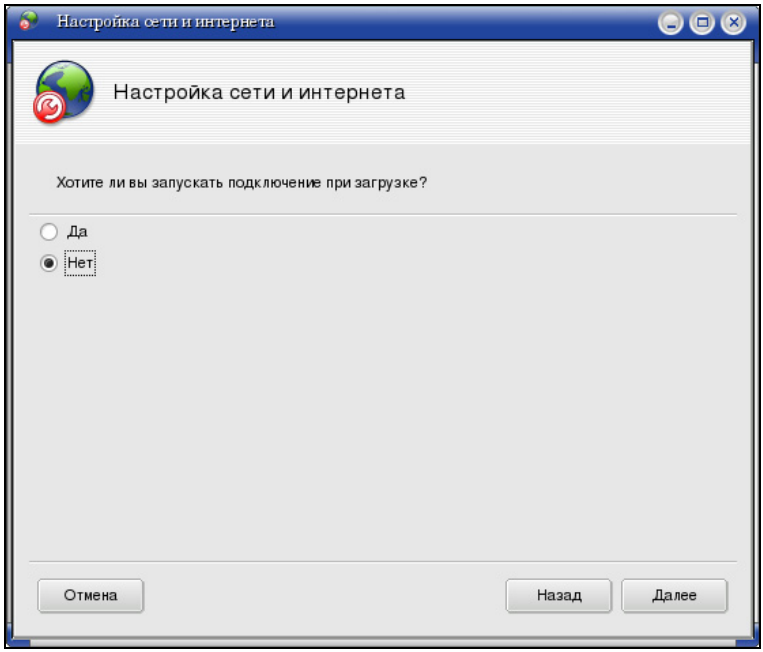

Рис. 9.12. Отказываемся от автоматической установки соединения при запуске

После этого конфигуратор будет последовательно запрашивать у вас различную информацию, связанную с соединением: ваш IP-адрес, IP-адрес DNS-сервер, IP-адрес шлюза. Оставьте все параметры по умолчанию — в большинстве случаев используется протокол DHCP, автоматизирующий настройку соединения, то есть вам ничего не нужно вводить — сервер провайдера самостоятельно настроит вашу систему.

Следующий вопрос мастера — хотите ли вы, чтобы созданное соединение запускалось автоматически при запуске системы (рис. 9.12). Рекомендую отказаться от этой идеи — соединение всегда можно запустить вручную, когда оно будет необходимо.

Если вы захотите изменить параметры соединения, то это можно сделать с помощью конфигуратора DrakConf (рис. 9.13). Для его запуска введите команду:

```
# DrakConf
```

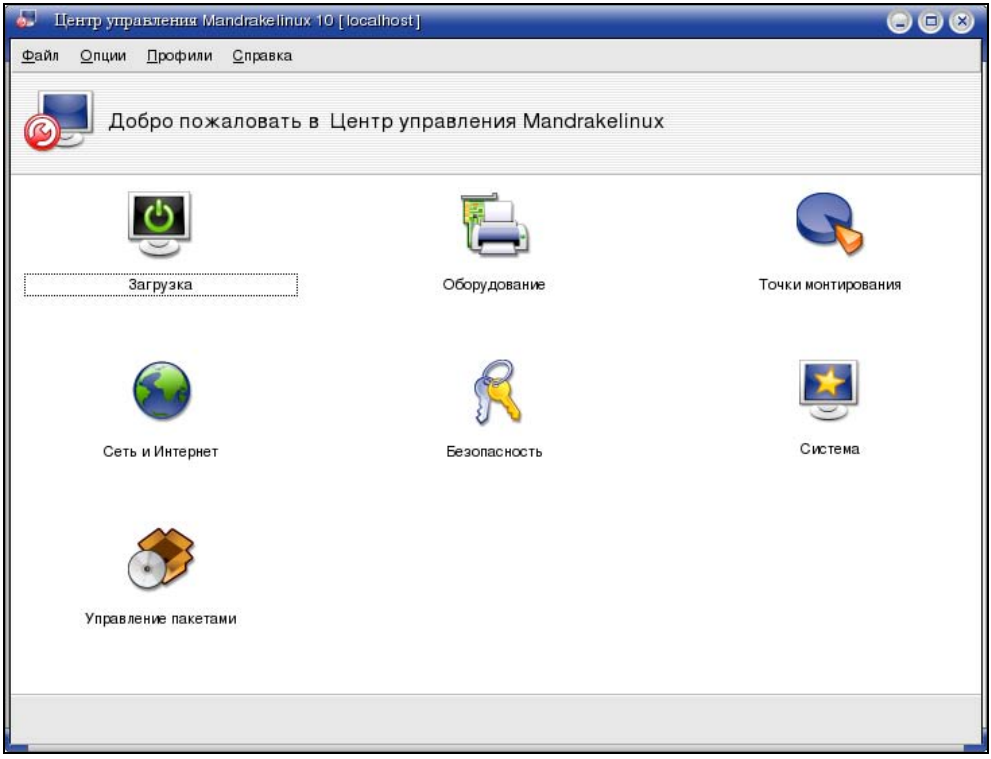

Рис. 9.13. Конфигуратор DrakConf

Для изменения параметров соединения выберите **Сеть и Интернет**, а затем — **Управление соединениями**. В появившемся окне выберите модемное соединение (modem) из списка соединений (рис. 9.14).

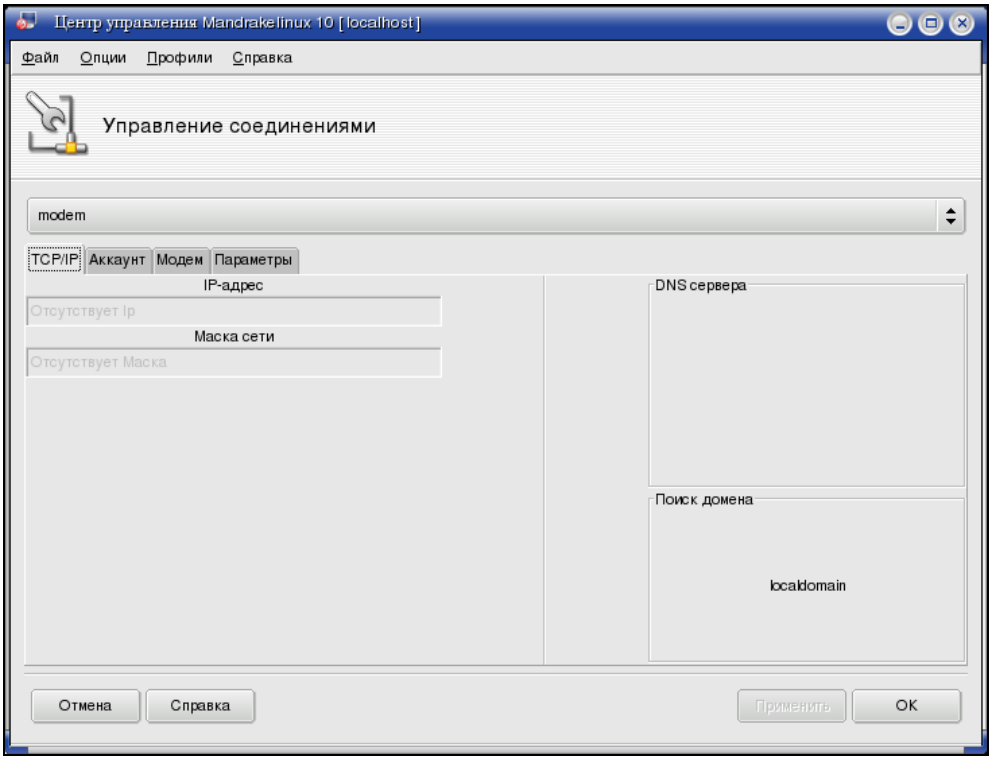

Рис. 9.14. Изменение параметров модемного соединения

На вкладке TCP/IP вы можете изменить параметры протокола TCP/IP, а именно: указать IP-адрес, маску сети. На вкладке **Аккаунт** — параметры вашей учетной записи. Различные параметры модема можно установить на вкладке **Модем**. На последней вкладке вы можете включить или выключить автоматический запуск этого соединения.

В отличие от конфигуратора сети ASP Linux, конфигуратор сети Linux Mandrake не позволяет установить соединение с провайдером — вы можете только создавать соединения, редактировать их параметры и удалять уже созданные соединения. Для установки соединения с провайдером в Linux Mandrake (и многих других дистрибутивах) используется программа kppp.

Нужно отметить, что если вы создаете модемное соединение с помощью конфигуратора сети в Linux Mandrake, оно автоматически "прописывается"

в настройках kppp. А вот в ASP Linux картина иная: если вы уже создали соединение с помощью конфигуратора сети, то вся дальнейшая работа должна происходить через эту программу. Если вы хотите использовать программу kppp, то вам нужно заново создать соединение, но уже в kppp. Я бы рекомендовал в ASP Linux использовать именно kppp, поскольку она предоставляет более удобный интерфейс пользователя и позволяет изменять значительно большее количество параметров соединения, чем штатный конфигуратор сети.

9.4. Программа kppp

Программа kppp соединяет в себе воедино сразу несколько функций. Во-первых, с ее помощью вы можете создавать, удалять и редактировать модемные соединения, причем она позволяет редактировать намного больше параметров соединения, чем конфигураторы сети. Во-вторых, программа kppp может установить соединение с Интернетом, чего не может конфигуратор сети Linux Mandrake. В-третьих, программу kppp может запустить любой пользователь, а конфигуратор сети — только root.

Запустите программу kppp с помощью одноименной команды (можно выбрать ее из меню KDE/GNOME — как кому нравится):

```
$ kppp
```

Если вы запускаете программу впервые и не настраивали модемных соединений с помощью конфигуратора сети, окно программы будет иметь такой вид (рис. 9.15).

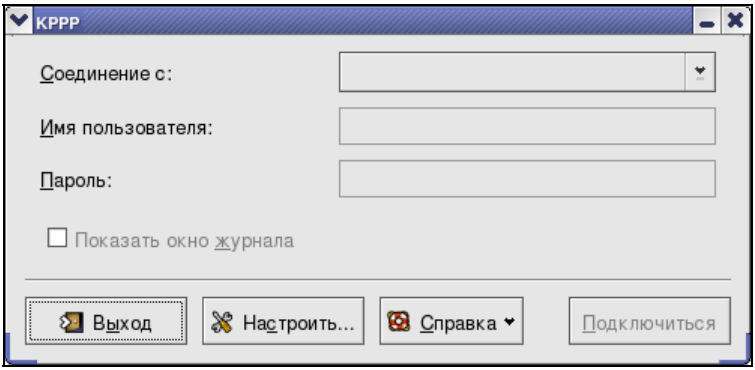

Рис. 9.15. kppp — первый запуск

Для начала настройки программы нажмите кнопку **Настроить**. В появившемся окне нажмите кнопку **Создать** (рис. 9.16).

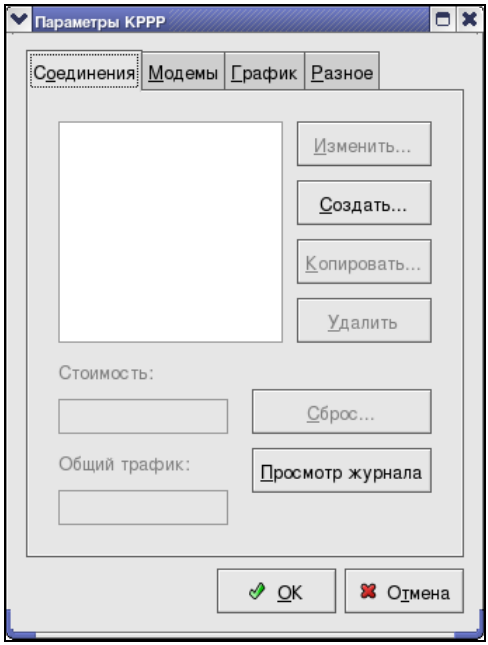

Рис. 9.16. Окно параметров kppp

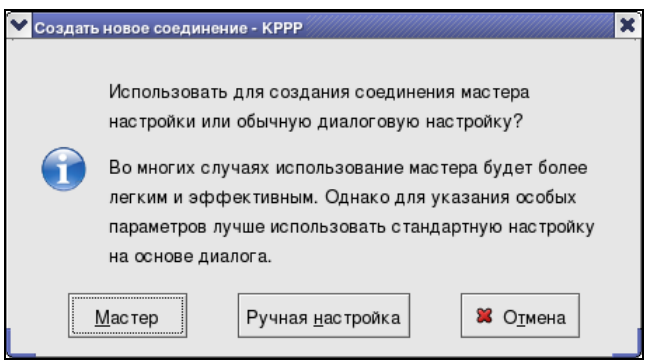

Рис. 9.17. Настраиваем соединение вручную

Программа спросит вас, желаете ли вы настроить соединение с помощью мастера или указать параметры вручную (рис. 9.17). Будет намного быстрее, если вы укажете параметры соединения вручную.

Появится окно нового соединения (рис. 9.18). В нем вам нужно указать название соединения и нажать кнопку **Добавить**, после чего ввести номер телефона модемного пула провайдера.

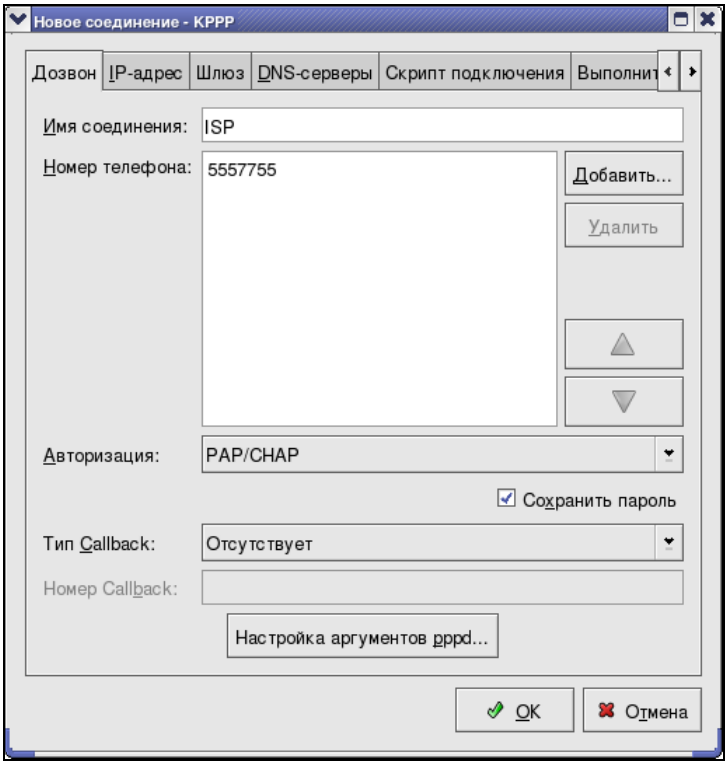

Рис. 9.18. Создание нового соединения

После этого изменять никакие параметры соединения больше не нужно, просто нажмите **OK**. В окне параметров kppp (см. рис. 9.16) перейдите на вкладку **Модем** и нажмите кнопку **Добавить**. Появится окно, в котором нужно будет ввести название модема и указать файл устройства (рис. 9.19). Если у вас COM-модем, подключенный к COM1, укажите файл устройства /dev/ttyS0, а если USB-модем, укажите файл устройства /dev/ttyUSB0.

Теперь перейдите на вкладку **Модем** (рис. 9.20) и укажите громкость и время ожидания при занятой линии. Если вы укажете 0 секунд, то программа автоматически наберет номер сразу же после получения сигнала "Занято".

Нажмите кнопку **Команды модема**. Найдите команду набора номера. По умолчанию используется команда **ATDT**. Если ваша АТС использует импульсный, а не тоновый тип набора номера, измените эту команду на **ATDP** (рис. 9.21). После чего нажмите кнопку **OK**.

Еще раз нажав **OK**, вернитесь в окно параметров kppp и перейдите на вкладку **Разное** (рис. 9.22).

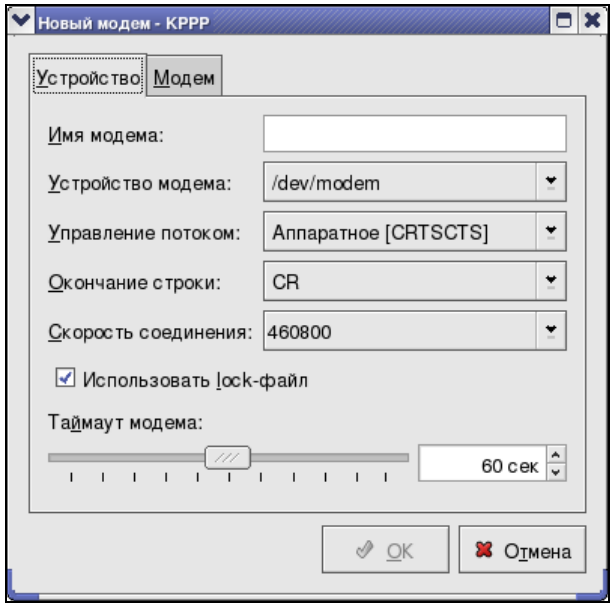

Рис. 9.19. Редактирование основных параметров модема

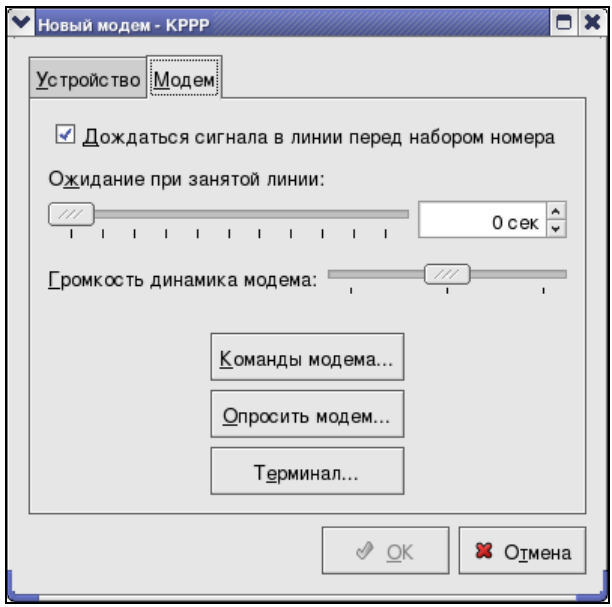

Рис. 9.20. Параметры модема

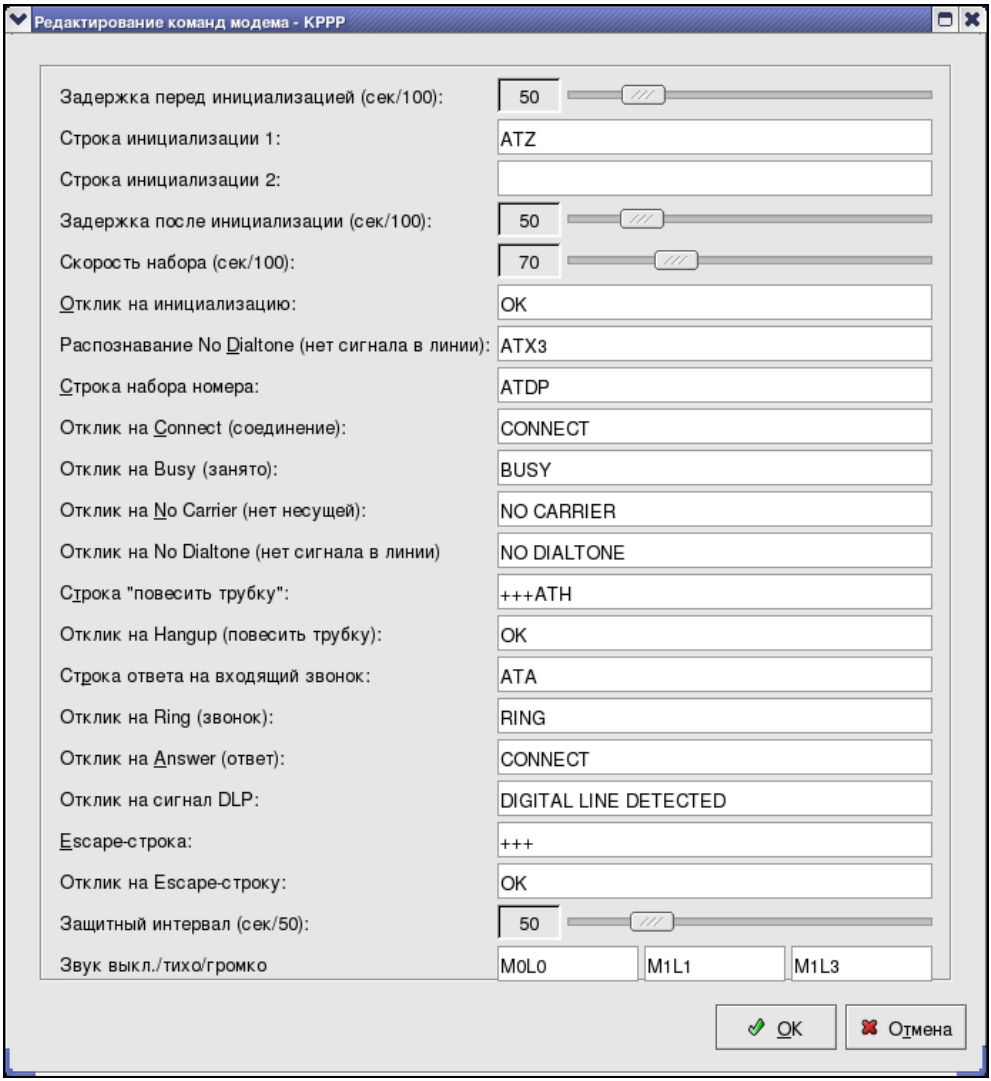

Рис. 9.21. Изменяем строку набора номера

Поскольку kppp — это оболочка для демона pppd, то в этом окне вы сможете узнать версию pppd, а также установить тайм-аут для pppd. Отредактировать параметры демона pppd можно в окне создания/редактирования соединения (см. рис. 9.18).

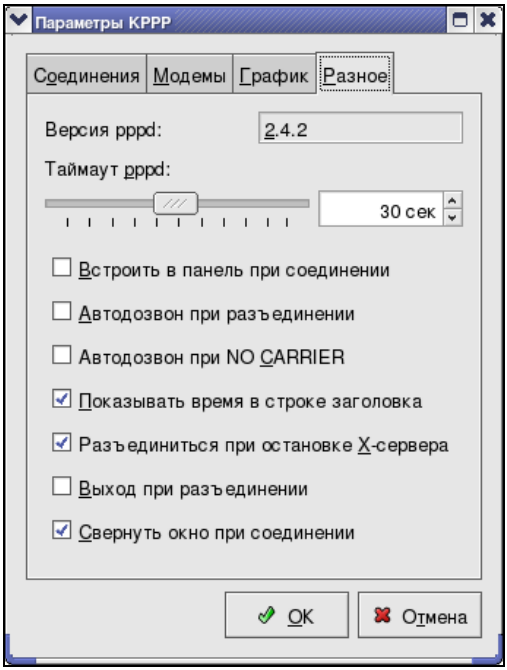

Рис. 9.22. Различные параметры kppp

Кроме этих параметров вы также можете включить/выключить следующие параметры kppp:

- **Встроить в панель при соединении** — по умолчанию kppp отображается в панели задач как обычное приложение, что не очень экономно по отношению к занимаемому на панели месту. Если данный режим включен, при установке соединения программа свернется в небольшой значок в панели задач (как в Windows);
- **Автодозвон при разъединении** — очень полезная опция, если соединение было разорвано, что часто случается, программа автоматически установит соединение;
- **Автодозвон при NO CARRIER** — если модем не может определить несущую, особой надобности в перезвоне нет, скорее всего, что-то случилось с телефонной линией;
- **Показывать время в строке заголовка** — программа будет показывать время соединения в строке заголовка;
- **Разъединить при остановке X-сервера** — если вы остановите X-сервер (или произойдет сбой), программа автоматически разорвет соединение с Интернетом;

- **Выход при разъединении** — абсолютно бесполезная опция, если соединение будет разорвано, программа завершит свою работу;
- **Свернуть окно при соединении** — окно программы будет свернуто на панель задач при установке соединения.

Нажмите кнопку **OK** в последний раз. Теперь в основном окне kppp введите имя пользователя и пароль. Нажмите кнопку **Подключиться** — через некоторое время соединение будет установлено (рис. 9.23).

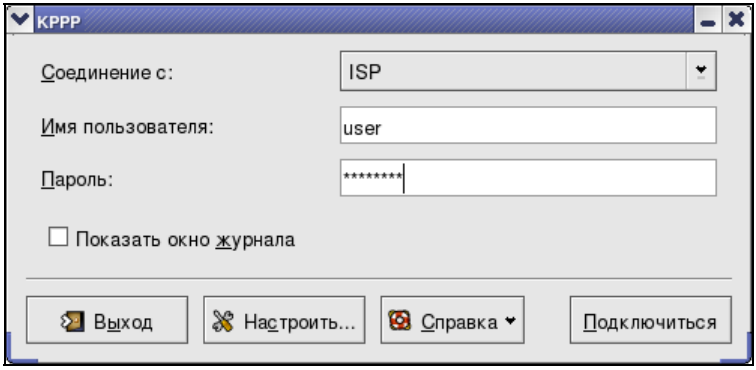

Рис. 9.23. Осталось нажать кнопку **Подключиться**!

Глава 10

Сеть в Linux

В этой главе мы не только поговорим о сетевом адаптере и его настройке в Linux, но и рассмотрим основы технологий Ethernet и Fast Ethernet, а также основы протокола TCP/IP. Ведь недостаточно знать, для чего используется тот или иной конфигуратор, намного важнее знать параметры того интерфейса, которые можно настроить с его помощью.

10.1. Сети

Прежде всего, нужно сказать, что такое сеть. Сеть — это не просто несколько компьютеров, объединенных общим кабелем. Сетью будем считать объединенные коммуникационным оборудованием и средой передачи данных узлы.

Теперь обо всем по порядку. Времена, когда компьютеры объединялись общим коаксиальным кабелем, давно прошли. К тому же, узлом сети может быть не только компьютер, а и другое сетевое устройство, например, принт-сервер. Причем у каждого узла есть свой уникальный (в пределах сети) адрес.

Сейчас для объединения узлов сети, кроме самого кабеля, используется различное коммуникационное оборудование — концентраторы, коммутаторы, мосты и т. д. Подробно обо всем этом мы говорить не будем, но с основными моментами придется разобраться.

10.1.1. Какие бывают сети

Сети бывают трех типов:

- ❒ локальные;
- ❒ муниципальные;
- ❒ глобальные (региональные).

Локальной сетью считается сеть в пределах одного предприятия. Даже если это предприятие не умещается в одном здании — это все равно локальная сеть, пусть и довольно большая. Два домашних компьютера, объединенных воедино — это тоже пример локальной сети, пусть и очень маленькой.

Муниципальная сеть — это сеть в масштабах города. В развитых странах муниципальные сети есть в каждом городе, кроме компьютеров обычных пользователей, они объединяют различные аварийные службы и другие муниципальные структуры. В нашей стране в последнее время популярны "мини-муниципальные" сети, объединяющие несколько микрорайонов (кабельное телевидение). Такие сети, кроме высокоскоростного доступа к Интернету, обеспечивают, как правило, дешевый обмен данными между абонентами сети.

Глобальные сети объединяют весь земной шар. Самым известным примером глобальной сети является Интернет. Существуют и другие глобальные сети, они частично используют каналы Интернета. Объясняется это очень просто: экономически невыгодно создавать отдельные каналы связи между отдельными континентами, если уже такие каналы есть и, тем более, их пропускная способность позволяет передавать нужное организаторам глобальной сети количество данных.

Стоит отметить, что в англоязычной литературе для локальных сетей используется аббревиатура LAN (Local Area Network), для муниципальных — MAN (Municipal Area Network), а для региональных (глобальных) — WAN (Wide Area Network).

В этой главе мы будем говорить только о локальных сетях, да и то не обо всех, а о самых распространенных — локальных сетях, построенных на основе технологии Ethernet и Fast Ethernet.

10.2. Технология Ethernet и ее модификации

Технология Ethernet, а точнее, ее модификации, в настоящее время наиболее часто используются для построения локальных сетей:

- Fast Ethernet — обеспечивает теоретическую скорость передачи данных в 100 Мбит/с, в качестве среды передачи данных использует витую пару пятой категории. Наиболее популярна для построения локальных сетей в пределах одного здания;
- Gigabit Ethernet — скорость передачи данных 1000 Мбит/с. Не очень популярна, поскольку такая скорость передачи данных не всегда бывает востребованной, а оборудование для Gigabit Ethernet все еще довольно дорого, особенно на фоне Fast Ethernet;

- Radio Ethernet — обеспечивает относительно небольшую скорость передачи данных (на практике — несколько Мбит/с), в качестве среды передачи данных использует радиоэфир. Преимущественно используется для подключения к сети удаленных клиентов (в пределах 30 км), которых нельзя подключить к сети обычным образом.

10.2.1. Классический Ethernet

Сейчас мы рассмотрим все эти три модификации более подробно, но сначала нужно сказать, почему нас не устраивает классический Ethernet. Во-первых, нас не устраивает скорость передачи данных — у всех многочисленных вариантов классического Ethernet максимальная скорость передачи составляет 10 Мбит/с, а на практике она достигает в лучшем случае 7 Мбит/с. В остальных многочисленных вариациях Ethernet меняется только среда передачи данных — тонкий коаксиальный кабель, толстый коаксиальный кабель или оптоволокно.

Во-вторых, монтаж витой пары гораздо удобнее, чем коаксиального кабеля. Конечно, толстый коаксиальный кабель лучше подходит для наружного применения, например, его можно, особо не опасаясь, проложить на крыше или протянуть между зданиями. Здесь есть одно "но". Если прокладывается коаксиальный кабель между зданиями, то, скорее всего, с целью подключения не отдельного клиента, а как магистральный кабель, объединяющий локальные сети этих двух зданий. А теперь представим такую ситуацию. Шлюз для доступа к Интернету установлен в здании 1, здания объединены толстым коаксиальным кабелем (рис. 10.1). Через этот шлюз к Интернету подключаются все пользователи обоих зданий.

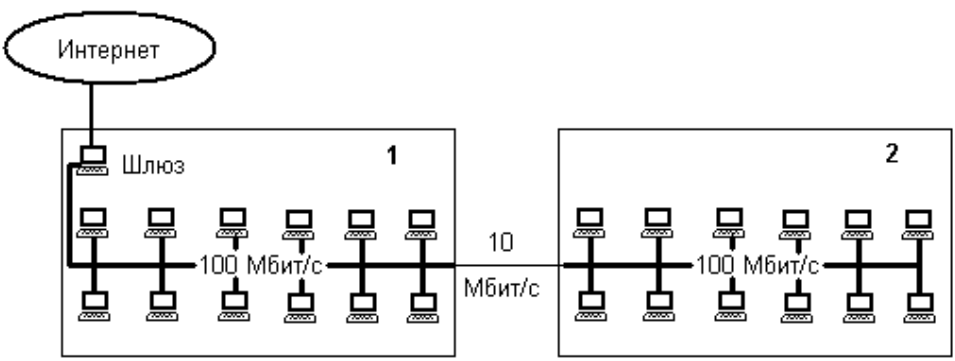

Рис. 10.1. "Тонкая" магистраль

Пользователи первого здания подключаются к серверу по сети 100 Мбит/с (Fast Ethernet), а пропускная способность пользователей второго здания ограничена 10 Мбит/с. Выходит, одним все, а другим — ничего. Другими словами, использование технологии Ethernet 10Base-5 на базе толстого коаксиального кабеля полностью исключено. В качестве магистрали лучше всего использовать Gigabit Ethernet или хотя бы Fast Ethernet на базе экранированной витой пары (STP) — она менее чувствительна к воздействиям окружающей среды, чем неэкранированная витая пара (UTP).

Еще лет 5–7 назад использование Ethernet могла оправдать ее дешевизна: да, тогда сеть Ethernet обходилась дешевле, чем Fast Ethernet. Тогда многие организации довольствовались классической сетью Ethernet стандарта 10Base-2 (рис. 10.2), основанной на тонком коаксиальном кабеле. Для организации такой сети нужно было несколько сетевых плат, поддерживающих Ethernet, несколько метров коаксиального кабеля, который стоил дешевле, чем витая пара третьей категории, использующаяся в стандарте 10Base-T, несколько Т-коннекторов (по количеству компьютеров), а также две заглушки — терминаторы.

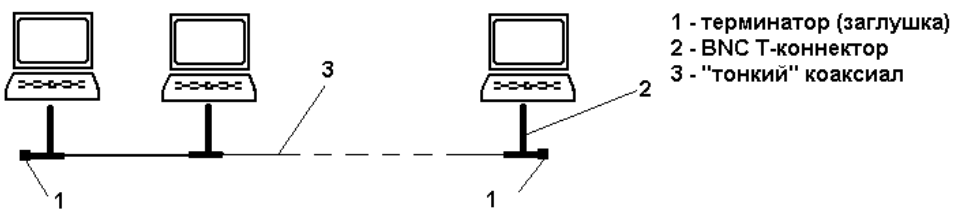

Рис. 10.2. Сеть на основе тонкого коаксиального кабеля (10Base-2)

Немногие себе могли позволить даже сеть стандарта 10Base-T (рис. 10.3). Данный стандарт использовал витую пару третьей категории (а она стоила дороже коаксиального кабеля), а также специальное устройство, объединяющее компьютеры такой сети — концентратор (не говоря уже о коммутаторах). Тем не менее, обслуживание сети 10Base-T было намного проще. Во-первых, монтаж витой пары существенно проще, а во-вторых, если произошел обрыв общего коаксиального кабеля, то останавливалась работа всей сети. В сети 10Base-T все компьютеры объединял концентратор (он же hub), и повреждение одного кабеля не останавливало работу всей сети — просто в сети не был доступен компьютер, кабель которого поврежден, а все остальные компьютеры по-прежнему могли обмениваться данными.

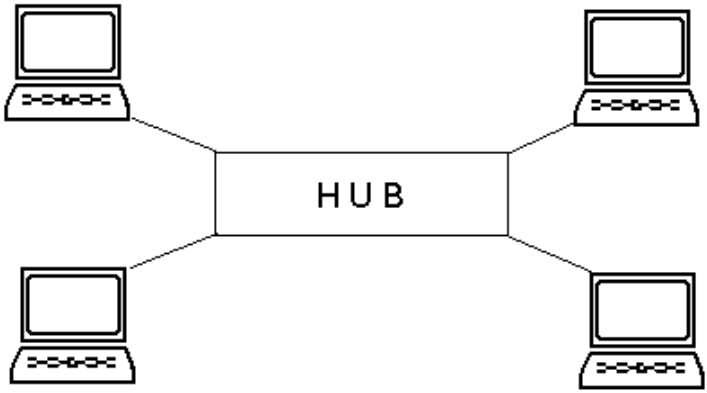

Рис. 10.3. Сеть стандарта 10Base-T

Вот теперь мы вплотную подошли к топологии сети. Существуют три основные топологии сети: общая шина, звезда и кольцо. Кольцо мы рассматривать не будем, а вот на первых двух остановимся. Также нужно различать физическую и логическую топологии. Физическая топология определяет, как компьютеры объединены в сеть, а логическая определяет, как передаются данные в сети.

Начнем с физической. Сеть стандартов 10Base-2 (тонкий коаксиал) и 10Base-5 (толстый коаксиал) основана на топологии общей шины, то есть все компьютеры сети подключаются к общему для всех кабелю. Сеть стандарта 10Base-T относится к топологии "звезда", поскольку каждый компьютер подключается к центральному устройству (концентратору) с помощью собственного кабеля.

А вот логическая топология у всех стандартов классического Ethernet одинаковая — общая шина. Рассмотрим, как передаются данные по сети стандарта 10Base-2/5. Компьютер определяет, занята ли среда передачи данных. Если среда свободна, он отправляет данные, адресованные компьютеру с адресом N. Переданные компьютером данные принимают абсолютно все компьютеры сети. Затем каждый компьютер сравнивает адрес получателя с собственным адресом. Если адреса совпадают, компьютер принимает данные, если нет — игнорирует их.

Точно так же работает и концентратор (стандарт 10Base-T). К его портам подключены все компьютеры сети. Он принимает переданные одним компьютером данные и повторяет их на все свои порты (кроме порта отправителя), то есть, как и в предыдущем случае, переданные одним компьютером данные получают все компьютеры.

Как видите, концентратор не нарушает логической топологии сети Ethernet, другое дело — коммутатор, но о нем мы поговорим в следующем разделе.

10.2.2. Технология Fast Ethernet

Информационные технологии не стоят на месте, в результате чего наблюдается вполне закономерное снижение цен на еще вчера дорогие технологии. Скажем, лет пять тому назад не каждая организация могла себе позволить сеть на основе Fast Ethernet. В лучшем случае данная технология использовалась в качестве магистральной технологии, например, между зданиями. Но сегодня никому в голову уже не придет использование обычного Ethernet, тем более что зачастую сеть, построенная по новой технологии, будет стоить столько же или даже дешевле, чем сеть на основе "старого" Ethernet. Не говоря уж о комфорте обслуживания сети на базе Fast Ethernet.

Также сегодня практически никто не будет использовать концентратор, который нерационально использует ресурсы сети. Его "место под солнцем" занял коммутатор (switch). Коммутатор вносит изменения в логическую топологию сети, а именно: коммутатор "знает", к какому порту подключен компьютер с каким адресом, и передаваемые данные отправляются не всем компьютерам, а только получателю. Сначала коммутатор работает в режиме обычного концентратора — он собирает данные о локальной сети, строит свою таблицу соответствия номера порта и MAC-адреса компьютера. После того как данная таблица построена, коммутатор переходит в "родной" режим и работает, как положено коммутатору:

- получает кадр от компьютера А и выделяет из заголовка MAC-адрес получателя — компьютера Б;
- анализирует свою внутреннюю таблицу и определяет номер порта, к которому подключен компьютер с MAC-адресом получателя;
- отправляет кадр нужному компьютеру.

Нагрузка на сеть снижается ровно в столько раз, сколько компьютеров есть в сети. В небольших сетях, скажем до 5 (максимум до 8) компьютеров, разница между коммутатором и концентратором не ощущается. А вот когда к центральному устройству подключено 16 или 24 узла, а может быть и еще одна подсеть (еще один коммутатор), то результат, как говорится, налицо.

Благодаря развитию научно-технического прогресса цена между концентраторами и коммутаторами почти сравнялась, что привело к снижению спроса на концентраторы. Сейчас концентратор в продаже — редкость.

Технология Fast Ethernet, которая используется сегодня повсеместно, в качестве среды передачи данных использует экранированную или неэкранированную витую пару пятой категории, а также оптоволокно, что позволяет достичь скорости передачи данных в 100 Мбит/с. Конечно, для такой скорости одной витой пары недостаточно — нужны еще специальные 100-мегабитные коммутаторы.

Спрашивается, а чем же отличаются технологии Ethernet и Fast Ethernet, кроме скорости передачи данных? С точки зрения пользователя — больше ничем. Основное отличие пользователь видит лишь в том, что его данные стали передаваться в 10 раз быстрее. Однако с технической точки зрения, отличия большие. Технология Ethernet использует метод доступа CSMA/CD, а Fast Ethernet — метод Demand Priority. Рассмотрим оба метода.

Метод CSMA/CD

CSMA/CD (Carrier-Sense-Multiply-Access with Collision Detection) — метод множественного доступа с обнаружением несущей и коллизий. Данный метод используется в сетях топологии общей шины. Согласно данному методу одновременно обмениваться данными могут только два компьютера: один компьютер передает данные, а другой — принимает.

Если в момент передачи по сети данных (между двумя компьютерами) передачу начнет еще один компьютер, то возникнет коллизия. Почему же возникают коллизии — ведь ранее было сказано, что перед тем, как отправить в сеть данные, компьютер проверяет, свободна ли среда передачи данных? Коллизии очень часто возникают в больших сетях, где крайние компьютеры сети находятся довольно далеко друг от друга. Предположим, что компьютер проверил, что среда свободна и начал передачу данных компьютеру, который находится недалеко от него. Несколько мгновений спустя компьютер на другом конце провода тоже "захотел" передать информацию. Он проверяет среду передачи данных и обнаруживает, что она свободна! Это происходит потому, что сигнал за столь короткий промежуток времени не успел "дойти" до другого конца кабеля. А когда он уже "дошел", то было поздно — еще один компьютер начал передачу данных. В результате на кабель будет передан еще один сигнал — от другого компьютера, и разделить сигналы (тот, который уже был, и тот, который только что отправлен) уже невозможно.

Если в сети обнаружена коллизия (обнаружением коллизий занимается именно метод CSMA/CD), то все компьютеры сети прекращают передачу данных. Сигналом для прекращения передачи является jam-последовательность, которую отправляет в сеть компьютер, первым обнаруживший коллизию. После коллизии каждый компьютер выбирает случайным образом время паузы, по окончанию которой он начинает передачу. Так что после коллизии начать передачу может даже компьютер, сигнал которого как раз и был причиной коллизии.

Если же коллизия повторяется, опять отправляется jam-последовательность в сеть, а всеми компьютерами — выбирается случайная пауза. Так может повторяться до 15 раз. Если после 16 попытки опять возникает коллизия, то ра-

бота сети останавливается, а пользователи видят на экране сообщение вроде "Сеть недоступна".

Спрашивается, а когда же будут передавать данные другие компьютеры, если они вынуждены ждать, пока не прекратится обмен между двумя компьютерами, захватившими среду передачи данных? Ведь компьютер, первым начавший передачу, не сообщает о размере передаваемых данных, он может передавать, скажем, 1 Гбайт информации. Представляете, сколько это займет времени, при условии, что реальная скорость передачи данных в Ethernet-сети составляет всего 6–7 Мбит/с? Неужели все остальные компьютеры будут ждать, пока весь гигабайт не будет передан?

Нет, никто столько времени ждать не будет — иначе можно так ничего и не отправить. Компьютеры не отправляют все данные сразу. Передаваемый объем данных, будь то хоть 10 Кбайт или 10 Гбайт, разбивается на кадры фиксированной длины. По окончании передачи кадра передающий компьютер должен выждать паузу — 9,6 мкс. Данная пауза еще называется *межкадровым интервалом*. А за это время начать передачу может уже другой компьютер. Первый компьютер, выждав паузу, пытается продолжить передачу данных, но сначала он проверит, свободна ли среда. Если среда свободна, он может продолжить передачу. Если нет — то он будет вынужден ждать, пока другой компьютер не передаст свой кадр.

Метод Demand Priority

Благодаря методу Demand Priority (метод доступа по требованию), использующемуся в Fast Ethernet, вероятность коллизии в Fast Ethernet сведена к нулю, поскольку теперь управлением доступом занимается концентратор.

Если компьютеру нужно передать данные, он отправляет концентратору специальный сигнал — запрос на передачу кадра. В этом сигнале также указывается приоритет, низкий или высокий. Низкий приоритет имеют службы, нечувствительные к задержкам (передача файлов, почты и т. д.). Высокий приоритет у мультимедиа-данных, которые чувствительны к задержкам. И в самом деле, когда придет очередная "порция" файла, сейчас или через несколько миллисекунд — разницы нет. А вот с мультимедиа-данными ситуация иная. Например, изображение может начать дергаться или отставать от звука.

Если сеть в данный момент свободна, концентратор сразу же отправляет кадр компьютеру-получателю. Если же сеть занята, кадр ставится в очередь на отправку. Место в очереди зависит от приоритета кадра, а также от времени поступления кадра. Например, если в очереди 3 кадра низкого приоритета и поступает кадр высокого приоритета, то он ставится в начало очереди, что

означает, что он будет отправлен первым. Если же в очереди уже есть кадр с высоким приоритетом (1 с высоким и 3 с низким), то поступивший новый кадр высокого приоритета ставится в очередь вторым — за уже имеющимся в очереди кадром высокого приоритета.

Стандарты технологии Fast Ethernet

Все модификации сети Fast Ethernet передают данные с теоретической скоростью в 100 Мбит/с, но используют разные кабели в качестве среды передачи данных (табл. 10.1).

Таблица 10.1. *Стандарты Fast Ethernet*

Стандарт	Тип кабеля
100Base-TX	неэкранированная витая пара пятой категории (UTP5) или экранированная витая пара первого типа (STP Type 1)
100Base-T4	неэкранированная витая пара категории 3, 4, 5
100Base-FX	оптоволоконный

10.2.3. Технология Gigabit Ethernet

Данная технология была разработана в 1998 г. Gigabit Ethernet позволяет передавать данные со скоростью до 1000 Мбит/с. При этом можно использовать следующие виды кабеля:

- витая пара пятой категории;
- оптоволокно;
- твинаксиальный кабель.

Максимальное расстояние передачи данных достигается при использовании одномодового оптоволоконного кабеля — 5000 м.

Нужно отметить, что оборудование Gigabit Ethernet до сих пор дорого, поэтому данная технология используется редко.

10.2.4. Технология Radio Ethernet

Технология Radio Ethernet в качестве среды передачи данных использует радиоэфир. При этом максимальная скорость передачи данных достигает 11 Мбит/с. Это на 1 Мбит/с больше, чем при использовании обычной технологии Ethernet.

Radio Ethernet используется преимущественно там, где невозможно проложить кабели для доступа компьютеров, например, между отдельными подразделениями компании, находящимися на расстоянии, скажем, 10–20 км друг от друга. Конечно, оптоволоконный кабель обеспечил бы большую скорость передачи данных, но расстояние и стоимость монтажа (не говоря уже о стоимости самого кабеля и коммуникационного оборудования) заставляет задуматься о рациональности его использования. Гораздо дешевле купить комплект оборудования для технологии Fast Ethernet.

Достоинство технологии Radio Ethernet является и ее недостатком — это радиоэфир. Не секрет, что радиоэфир наименее помехоустойчив, не говоря уже о возможности перехвата информации. Отсюда следует:

- Желательна передача данных в зашифрованном виде, иначе данные компании очень быстро могут стать всенародным достоянием.
- Скорость передачи данных может падать из-за плохой погоды, радиопомех и других факторов.

Заявленное максимальное расстояние при использовании технологии Radio Ethernet — 30 км, но это в идеальном случае (в поле, где есть прямая видимость обеих антенн). На практике обычно Radio Ethernet используется для передачи данных на расстояние, не превышающее 15–20 км.

Но в любом случае Radio Ethernet остается заманчивой технологией для беспроводного доступа к Интернету. При условии, что не будет больших радиопомех, то из этой технологии можно будет "выжать" 6–7 Мбит/с, чего вполне хватит для нормального доступа к Интернету, особенно, если подключается один компьютер, например, домашний пользователь, а не целая сеть. Многие провайдеры предоставляют именно такой способ доступа к сети Интернет. Для домашних пользователей это очень выгодно — такую скорость обеспечит не каждый модем (даже ADSL), к тому же, не нужно заботиться о кабеле, который могут просто украсть. Поскольку ретрансляторы устанавливаются провайдером, вам не нужно заботиться о расстоянии от вашего дома до провайдера — в большинстве случае все будет в норме, потому что провайдер старается оптимально распределить "зону покрытия".

Технология Radio Ethernet, кроме всего прочего, позволяет быть более "мобильным". Представьте такую ситуацию: вы потратили несколько месяцев на установку выделенной линии, а через год вы захотели поменять квартиру или офис — выделенную линию нужно будет прокладывать заново. А с антенной все просто — снял и перенес на новое место.

10.3. Физическое подключение компьютера к сети Fast Ethernet

Прежде всего, нужно убедиться, что у вас есть сетевой адаптер, поддерживающий эту технологию. Большинство современных компьютеров оснащены такими сетевыми адаптерами. Как правило, в современных компьютерах сетевые адаптеры являются интегрированными в материнскую плату, и устанавливать их отдельно не нужно. Но встречаются материнские платы и без интегрированных сетевых адаптеров. В этом случае вам нужно купить сетевой адаптер (рис. 10.4).

Рис. 10.4. Сетевой адаптер Fast Ethernet (Intel)

Установка сетевого адаптера проблем не вызывает — просто вставьте ваш сетевой адаптер в свободный разъем шины PCI (все Fast Ethernet-адаптеры выполнены как платы расширения именно для шины PCI). Существуют и USB-сетевые адаптеры (рис. 10.5), позволяющие подключиться к сети, не разбирая компьютер. Но такие адаптеры стоят дорого и встречаются очень редко (тем более, что я не знаю, как будет работать Linux с таким вот чудом научно-технического прогресса).

Рис. 10.5. USB-сетевой адаптер

Ясно, что устанавливать сетевой адаптер нужно при выключенном компьютере — шина PCI пока еще не поддерживает "горячей замены".

После это вам нужно подключить сетевой кабель к сетевому адаптеру. Как правило, кабель обжимается администратором сети.

Вы администратор сети и не знаете, что делать? Не паникуйте, сейчас разберемся. Для создания сети Fast Ethernet вам нужны следующие устройства:

- сетевые адаптеры — с ними мы уже разобрались;
- коммутатор (switch) — его можно купить в любом компьютерном магазине (рис. 10.6);
- витая пара пятой категории — спрашивайте именно такой тип кабеля;
- коннекторы RJ-45 — таких коннекторов вам нужно будет в два раза больше, чем число компьютеров, поскольку кабель нужно будет обжать с двух концов. Но я рекомендую купить еще несколько штук — если вы будете обжимать кабель впервые, думаю, сделаете несколько ошибок;
- инструмент для обжимки витой пары — хороший инструмент стоит относительно дорого (примерно как коммутатор), а плохой лучше не покупать. Если не хотите выкладываться, возьмите у кого-нибудь на пару дней.

Рис. 10.6. Коммутатор (switch)

Теперь приступим к самому процессу обжимки. Внутри кабеля будут 4 витые пары, причем у каждого провода своя цветовая маркировка. Суть процесса обжимки заключается в том, чтобы подключить каждый из проводов к нужному контакту коннектора.

1. Сначала нужно поместить провода в коннектор (зачищать их необязательно — за вас это сделает инструмент).
2. Затем коннектор обратной частью (той, которой он будет вставляться в сетевой адаптер) помещается в инструмент для обжимки и крепко обжимается. Используя табл. 10.2, вы без проблем сможете обжать кабель.

Таблица 10.2. *Обжимка витой пары*

Контакт	Цвет провода
1	Бело-оранжевый
2	Оранжевый
3	Зелено-белый
4	Синий
5	Сине-белый
6	Зеленый
7	Бело-коричневый
8	Коричневый

Обжимать кабель нужно с двух сторон. Один конец подключается к концентратору (или коммутатору), а второй — к сетевому адаптеру. Если вы неправильно (или нечетко) обжали кабель, то ваша сеть работать не будет или же будет работать только на скорости 10 Мбит/с. Проверить, правильно ли вы обжали кабель, очень просто — обратите внимание на коммутатор. Возле каждого порта два индикатора.

- Если горят оба, значит, все нормально.
- Если же горит только один из них, значит, данный порт работает в режиме 10 Мбит/с.
- А если вообще не горит ни один из индикаторов, значит, вам нужно обжать кабель заново.

Как видите, в процессе нет ничего сложного.

10.4. Немного о TCP/IP

В следующем пункте мы будем рассматривать настройку сетевого интерфейса в Linux, при этом программа конфигуратор попросит нас ввести параметры интерфейса: IP-адрес, маску сети, IP-адрес шлюза и т. д. Мы должны понимать, что мы вводим и зачем это нужно. Вот для этого и предназначен данный раздел о TCP/IP.

10.4.1. Основной протокол Интернета

Основным протоколом Интернета является TCP/IP (Transmission Control Protocol/Internet Protocol, Протокол управления передачей/интернет-протокол).

Локальные сети строятся "по образу и подобию" Интернета и также используют TCP/IP. (Конечно, это относится не ко всем локальным сетям, но не будем же мы в Linux настраивать протокол NetBIOS от Microsoft.)

Функции протокола TCP/IP следующие:

- гарантированная передача данных по сети (TCP-часть);
- адресация сети, то есть присваивание каждому узлу сети собственного IP-адреса (IP-часть).

Кроме протокола TCP/IP, вам нужно знать, для чего используются следующие протоколы, чтобы при настройке Linux вы чувствовали себя "в своей тарелке":

- PPP (Point-to-Point Protocol) — протокол точка-точка. Обычно используется для подключения к Интернету с помощью модема (будь то коммутируемое соединение или выделенная линия);
- ICMP (Internet Control Message Protocol) — протокол управляющих сообщений Интернета. Используется для обмена управляющей информации, а также для тестирования сети. Вы знакомы с программой ping? Так вот, эта программа использует именно протокол ICMP — она отправляет ICMP-сообщение echo и ждет ответ от удаленного узла. Если ответ "пришел", значит, все нормально, если нет — возможно, имеется проблема с сетью или ее настройкой;
- HTTP (Hyper Text Transfer Protocol) — протокол передачи гипертекстовой информации. На основе этого протокола работает Всемирная паутина — WWW (World Wide Web);
- FTP (File Transfer Protocol) — протокол передачи файлов. Используется для обмена файлами;
- POP (Post Office Protocol) — протокол почтового отделения. Позволяет хранить на сервере сообщения электронной почты пользователей. С помощью почтового клиента пользователь подключается к POP-серверу и загружает на свой компьютер адресованные ему сообщения;
- SMTP (Simple Mail Transfer Protocol) — простой протокол передачи почты. Используется для отправки почты пользователя;
- RIP (Routing Information Protocol) — протокол обмена информацией для маршрутизации. Используется для маршрутизации в небольших сетях, когда же сеть большая, обычно используется протокол OSPF (Open Shortest Path First).

Все перечисленные протоколы относятся к семейству протоколов TCP/IP. Конечно, мы рассмотрели не все протоколы, но они или редко используются, или вообще не используются (например, SLIP).

10.4.2. IP-адреса

Итак, протокол IP используется для адресации в сети, но при рассмотрении технологии Ethernet мы имели дело с MAC-адресами. Нужно разобраться, для чего используется тот или иной адрес. MAC-адрес — это уникальный (аппаратный) адрес любого сетевого устройства. За каждым производителем сетевых устройств закреплен свой диапазон MAC-адресов, и адреса назначаются производителем именно из этого диапазона. В мире не может быть двух сетевых плат с одинаковым MAC-адресом. Кстати, это относится и к другим сетевым устройствам — тот же мобильный телефон также имеет свой аппаратный адрес, но он называется не MAC, а IME.

Использовать MAC-адреса не совсем удобно, поскольку они группируются по производителю. А как тогда, спрашивается, объединить компьютеры в локальную сеть, если у нас есть платы разных производителей. Да никак. Вот поэтому и используется протокол IP, который назначает каждому устройству сети свой собственный IP-адрес. Нужно отметить, что если у сетевого устройства есть несколько сетевых интерфейсов (обычно у шлюза есть несколько интерфейсов), то IP-адрес назначается каждому из них.

IP-адрес — это 32-разрядное число, которое записывается в десятичном или шестнадцатеричном виде (реже). Как правило, IP-адрес разбивается на четыре части, разделенные точками. Вот примеры адресов:

- 192.168.0.1;
- 127.0.0.1;
- 128.11.99.1.

Легко посчитать, что пространство IP-адресов должно было закончиться еще несколько лет назад, учитывая, сколько компьютеров в мире.

Выход из этого есть: "настоящие" IP-адреса назначаются только определенным сетям, и стоит каждый IP-адрес совсем недешево. Если вашей локальной сети нужен Интернет, то назначается, как правило, один "настоящий" IP-адрес — на шлюзе — компьютере, который будет использоваться для доступа к Интернету. А все остальные компьютеры сети будут использовать "внутренние" IP-адреса. О внутренних IP-адресах мы поговорим в следующем разделе.

10.4.3. Подсети, маски, классы IP-адресов

Сеть (даже небольшую нашу, локальную) можно разделить на подсети. Маска подсети определяет, сколько IP-адресов будет в составе сети. Вот примеры масок:

```
225.225.225.0       маска на 255 адресов (0..255)
225.225.225.224     маска на 32 адреса (224...255)
```

Сети бывают большие и маленькие. Класс сети определяет, сколько IP-адресов будет в сети. Существует пять классов сетей (A, B, C, D, E), но на практике используются только первые три:

- класс A — огромные сети, содержащие 16 777 216 адресов. Маска сети класса A — 255.0.0.0;
- класс B — средние сети на 65 536 адресов. Маска сети класса B — 255.255.0.0;
- класс C — небольшие сети на 254 адреса (на самом деле, адресов 256, но поскольку 0 и 255 зарезервированы, то использовать можно только 254 адреса). Маска сети — 255.255.255.0.

Если у вас небольшая локальная сеть, скорее всего, это будет сеть класса C. Запомните ее маску 255.255.255.0. Именно это значение нужно будет ввести при конфигурировании сетевого интерфейса в Linux. Конечно, я могу и ошибаться, но параметры сети можно всегда уточнить у администратора.

Мы говорили о внутренних адресах, которые не могут использоваться в Интернете, а зарезервированы исключительно для внутреннего использования:

- 10.0.0.0 — сеть класса A, маска 255.0.0.0;
- 172.16.0.0 — 172.31.0.0 — 16 сетей класса B, маска каждой из них — 255.255.0.0;
- 192.168.0.0 — 192.168.255.0 — 256 сетей класса C, маска — 255.255.255.0.

Лучше всего использовать третий вариант — вряд ли в вашей подсети будет более 255 узлов. Даже если это так, тогда ее целесообразно разделить на несколько сетей класса C.

Я говорил, что адреса 0 и 254 в сети класса C зарезервированы? Сейчас мы узнаем, для чего. Рассмотрим IP-адрес 192.168.1.1. Данный IP-адрес принадлежит к сети 192.168.1.0. Да, 192.168.1.0 — это адрес сети. Его также вы будете использовать при настройке сетевого интерфейса. Адрес 192.168.1.255 — это широковещательный адрес (broadcast), необходимый для широковещательной рассылки всем узлам сети.

10.4.4. Маршрутизация

Маршрутизация — это передача данных между сетями. Устройство, выполняющее передачу данных между сетями, называется маршрутизатором.

Кроме термина маршрутизатор, часто используется родственный ему термин — шлюз (gateway). Но, несмотря на то, что эти два устройства выполняют похожие функции, между ними есть разница. Маршрутизатор связывает

две сети одного типа, например, две локальные сети. Шлюз связывает две сети разного типа, например, локальную сеть и глобальную — Интернет.

Маршрутизатор и шлюз могут быть как аппаратными (отдельное устройство), так и программными — компьютер со специальным программным обеспечением.

Операционная система Linux может выступать в роли программного маршрутизатора/шлюза. Для настройки маршрутов прохождения пакетов используется программа route. В нашем случае настраивать ни шлюз, ни маршрутизатор не нужно — пусть этим занимается администратор сети. Наше дело маленькое — узнать у администратора IP-адрес шлюза и указать его при настройке сетевого интерфейса. Наш компьютер будет передавать пакеты, выходящие за пределы сети шлюзы, а тот пусть сам разбирается, что с ними делать.

10.4.5. Система доменных имен DNS

Человеку намного проще работать с символьной информацией, чем с числами, например, проще запомнить имя компьютера alex, чем IP-адрес 192.168.71.159. Разработчикам протокола TCP/IP нужно было выбирать между комфортом пользователей и простотой разработки сетевых приложений. В результате появилось компромиссное решение — система доменных имен (Domain Name System), преобразующая IP-адрес в соответствующее имя.

Предположим, что в строке браузера вы вводите адрес **www.library.ru**. Ваш браузер обращается к операционной системе с просьбой разрешить имя www.library.ru в IP-адрес. Если он получает от системы соответствующий IP-адрес, он устанавливает соединение с удаленным узлом и отображает Web-страницу. Как видите, чтобы установить соединение, компьютеру нужно знать IP-адрес удаленного узла.

Разрешение доменного имени — это процесс преобразования символьного (доменного) имени узла в соответствующий ему IP-адрес.

В каждой сетевой операционной системе есть небольшая программа, занимающаяся разрешением доменных имен — резолвер. По сути, резолвер — это клиент DNS-сервера. Его задача маленькая — передать серверу доменное имя, получить IP-адрес и передать его в качестве результата. Нужно также отметить, что резолвер используется и для обратного преобразования, то есть для разрешения IP-адреса в соответствующее ему имя узла.

Рассмотрим процесс работы Linux-резолвера:

1. Резолвер получает символьное имя узла.
2. Если заданное имя есть в кэше резолвера, резолвер сразу возвращает IP-адрес.

3. Если же имени в кэше резолвера нет, он производит поиск в файле /etc/hosts.
4. Если же имени в файле /etc/hosts нет, резолвер подключается к DNS-серверу, IP-адрес которого указан в конфигурационных файлах резолвера, и просит его разрешить имя.
5. Если сервер возвращает результат, резолвер передает его "наверх". Если же имя разрешить не удалось, резолвер сообщает об этом программе, запросившей преобразование.

Система DNS имеет иерархическую структуру — и это правильно. Представьте, что бы было, если бы в мире был один централизованный DNS-сервер. Нагрузки на этот сервер были бы огромными, и вся система DNS тогда бы работала очень медленно. К тому же, любая неисправность центрального сервера означала бы останов всего Интернета. На практике же каждый DNS-сервер отвечает только за свой домен или небольшую группу доменов. Рассмотрим процесс преобразования доменного имени www.library.ru с того момента, как резолвер передал его серверу DNS:

1. Если имя www.library.ru есть в кэше DNS-сервера, соответствующий IP-адрес сразу же возвращается резолверу.
2. Если заданного имени в кэше нет, тогда сервер обращается к DNS-серверу домена первого уровня — в данном случае к серверу домена .ru.
3. Сервер домена .ru определяет сервер, ответственный за домен library.ru, и передает ему заданное имя.
4. Сервер домена library.ru разрешает доменное имя и возвращает заветный IP-адрес серверу домена .ru.
5. Сервер домена .ru возвращает IP-адрес www.library.ru нашему DNS-серверу, который, в свою очередь, возвращает его резолверу.

Настройки резолвера хранятся в файле /etc/resolv.conf. Обычно в данном файле указываются следующие параметры:

- имена DNS-серверов (директива `nameserver`);
- домен, к которому принадлежит компьютер (директива `domain`);
- список доменов, которые будут использоваться при поиске (директива `search`).

Минимально необходимой является директива `nameserver`. Одна директива может определить один DNS-сервер. Всего можно определить четыре DNS-сервера. Например:

```
domain library.ru
nameserver 113.154.116.1
nameserver 113.154.116.2
```

С помощью директивы `search` можно указать список доменов, которые будут использованы, если пользователь неточно указал имя узла, например:

```
search .ru .net .com
```

Если пользователь введет имя, например, www.library, и система DNS не сможет его разрешить, то будет предпринята попытка сначала разрешить имя www.library.ru, потом, если на предыдущем этапе ничего не получилось — www.library.net и т. д.

На работу резолвера также влияет файл /etc/hosts.conf, точнее его директива `order`:

```
order hosts,bind
```

Данная запись означает, что резолвер должен сначала выполнить поиск IP-адреса в файле /etc/hosts, а потом только (если в файле /etc/hosts не будет найдено нужное имя), обратиться к DNS-серверу. Такой порядок удобен для небольших локальных сетей, у которых нет собственного DNS-сервера. Если у вас обычный домашний компьютер, можете установить директиву `order` так:

```
order bind
```

Данная запись означает, что резолвер сразу будет запрашивать DNS-сервер, то есть действовать "в обход" файла /etc/hosts.

Файл /etc/hosts имеет очень простой формат:

```
IP-адрес полное_доменное_имя псевдоним
```

Например:

```
127.0.0.1     localhost.localdomain            localhost
192.168.1.1   alex.localhdomain          alex
192.168.1.2   igor.localhdomain          igor
```

Как видите, в этом файле просто перечислены IP-адреса и соответствующие им доменные имена. Ясно, что после редактирования этот файл нужно будет скопировать на все компьютеры сети, иначе компьютеры, у которых есть старая версия этого файла, будут работать некорректно.

Использование этого файла имеет смысл в небольшой сети (максимум 10 компьютеров), где есть возможность скопировать этот файл на каждый компьютер в случае его изменения. Если компьютеров в сети много, гораздо проще настроить собственный DNS-сервер, который будет заниматься преобразованием доменных имен в IP-адреса, и наоборот.

В следующем разделе мы поговорим о самом интересном — практической настройке сети в Linux.

10.5. Настройка сети в Linux

Во избежание недоразумений все команды, упомянутые в данном пункте, нужно запускать от имени пользователя root в графическом терминале, команду для запуска которого можно выбрать из меню KDE/GNOME.

10.5.1. Настройка сети в Linux Mandrake

1. Перед началом настройки убедитесь, что сетевой кабель подключен, а также — что запущен сервис network, обеспечивающий поддержку сети. Убедиться в последнем очень просто — достаточно от имени пользователя root выполнить команду `ifconfig`. Если в выводе программы вы увидите информацию об интерфейсе lo, значит, все нормально (рис. 10.7).

   Интерфейс lo — это интерфейс обратной петли, использующийся преимущественно для тестирования поддержки сети. Если интерфейса lo нет в выводе программы, значит, вам нужно запустить сервис network:

   ```
   # service network Start
   ```

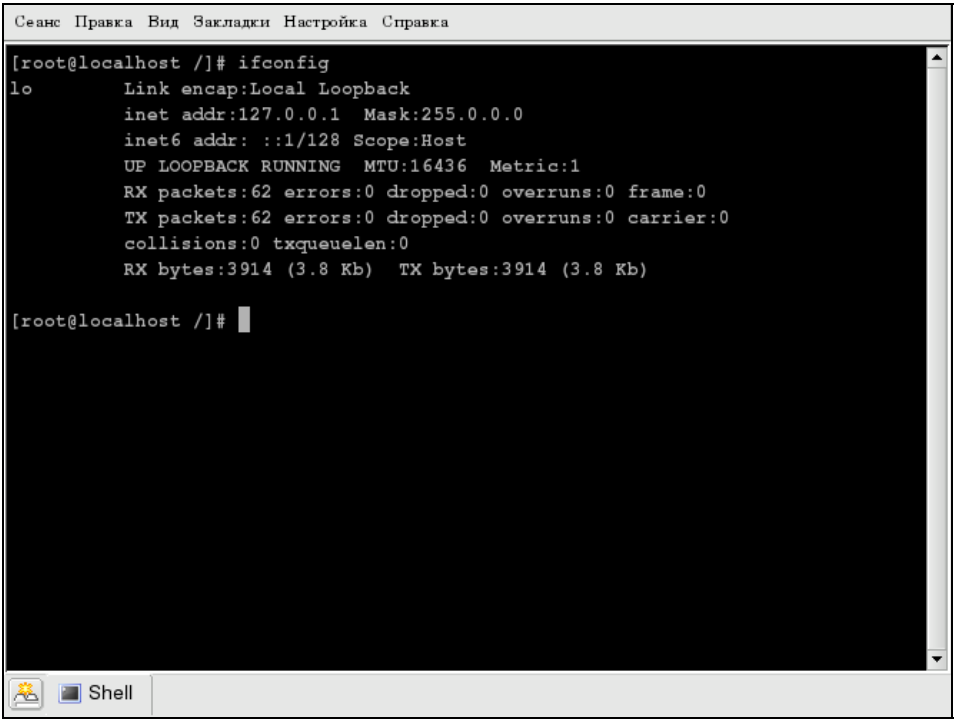

Рис. 10.7. Команда **ifconfig**

2. Для настройки локальной сети запустите конфигуратор drakconnect и выберите тип соединения **Соединение по локальной сети** (рис. 10.8).

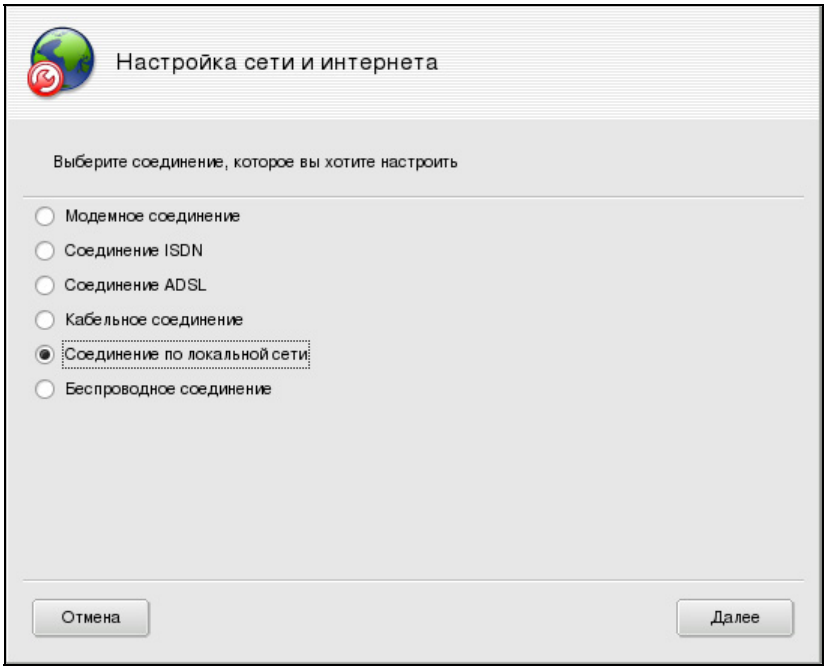

Рис. 10.8. Конфигуратор drakconnect — создание соединения по локальной сети

3. После этого конфигуратор предложит вам выбрать устройство, которое будет использоваться для этого соединения, попросту говоря — сетевую плату (рис. 10.9). Если вы заметили, то до этого момента ничего не было сказано ни о модулях сетевых плат, ни о поддержке сетевых плат операционной системой. А все дело в том, что Linux поддерживает практически все сетевые платы. Во всяком случае, не поддерживаемая сетевая плата мне еще не попадалась.

   Если у вас несколько сетевых плат, нужно выбрать именно ту, к которой подсоединен кабель сети.

4. Следующий этап — выбор типа настройки: автоматический с помощью DHCP или ручной — в этом случае параметры TCP/IP вам нужно будет ввести вручную. Выбирать наугад не нужно — уточните у администратора эти параметры. В случае если в вашей сети развернут DHCP-сервер, вам не нужно будет вводить IP-адрес, маску сети, IP-адрес шлюза, адреса DNS-серверов — в общем, на этом ваша настройка сети и закончится, поэтому далее мы будем рассматривать именно ручное конфигурирование сети (рис. 10.10).

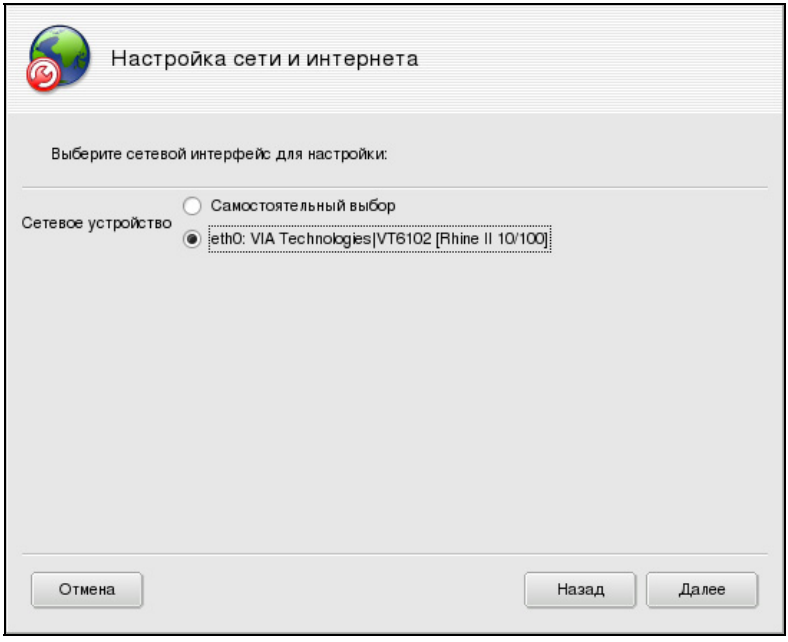

Рис. 10.9. Выбор сетевой платы

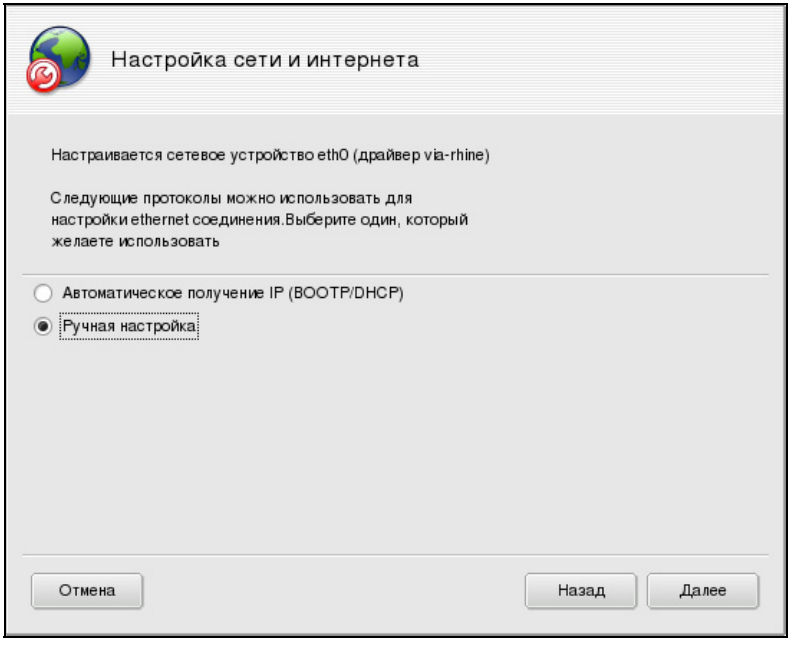

Рис. 10.10. Автоматическая или ручная настройка?

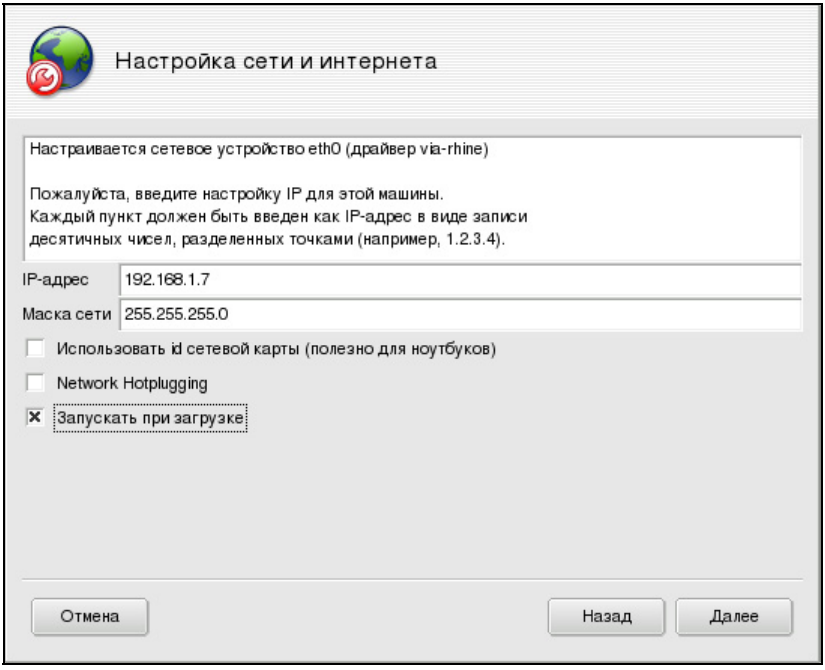

Рис. 10.11. IP-адрес и маска сети

5. Введите IP-адрес сетевого интерфейса и уточните предложенную конфигуратором маску (рис. 10.11). Нужно отметить, что конфигуратор сам пытается вычислить маску сети по введенному IP-адресу, и в большинстве случаев у него это получается.
6. Если вы хотите, чтобы соединение устанавливалось при загрузке системы (в большинстве случаев это так), установите соответствующий флажок. Также можете отметить опцию **Использовать id сетевой карты**, но это только в случае, если у вас ноутбук, иначе она будет бесполезной. Функцию **Network Hotplugging** включать не стоит.
7. После ввода параметров IP вам предстоит ввести параметры DNS (рис. 10.12), а именно:
   - имя узла — введите полное доменное имя вашего компьютера. Если ваша сеть не подключена к Интернету, можете ввести все, что угодно, например, свое имя (естественно, латинскими буквами и без пробелов);
   - IP-адреса DNS-серверов;
   - список поиска домена — это директива `search` файла конфигурации /etc/resolv.conf;
   - IP-адрес шлюза.

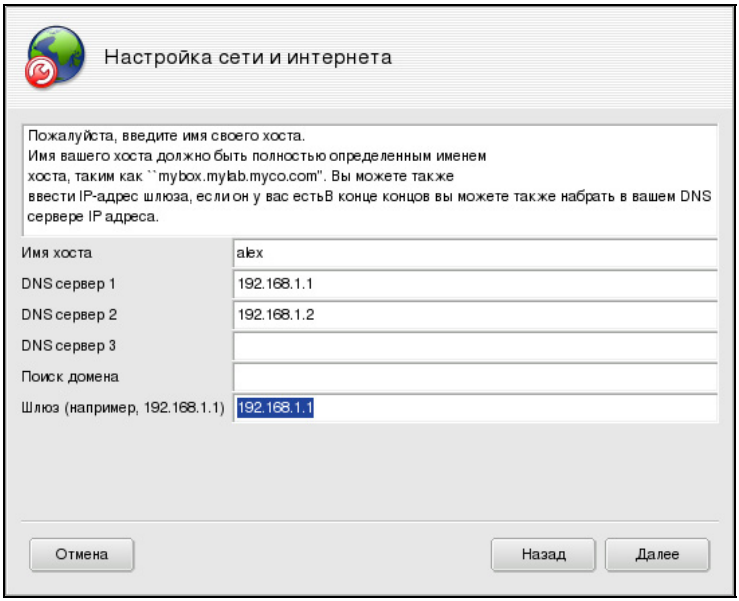

Рис. 10.12. Ввод параметров DNS

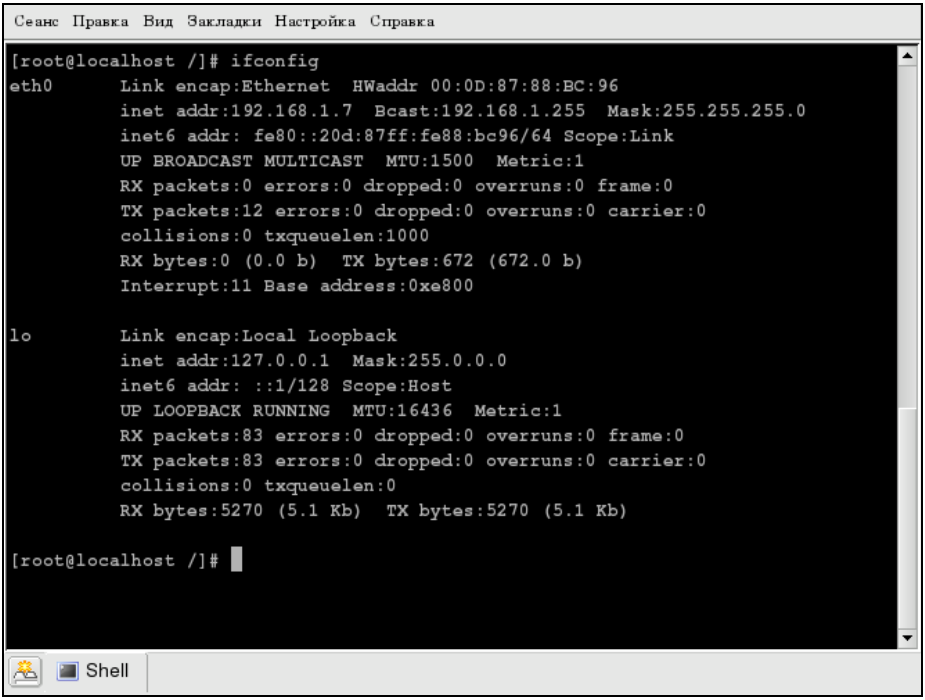

Рис. 10.13. Информация об интерфейсе eth0

8. Следующий шаг — это ввод адреса узла Zeroconf, просто ничего не вводите и нажмите кнопку **Далее**.
9. Последний вопрос конфигуратора — следует ли перезапустить сеть. Согласитесь, если хотите, чтобы изменения вступили в силу сразу.
10. Теперь самое время проверить правильность настроек. Прежде всего, убедимся, что интерфейс eth0 (это ваша первая сетевая плата) "поднят". Введите команду `ifconfig`. Вы должны увидеть информацию об интерфейсе eth0 (а также о других активных интерфейсах). С помощью `ifconfig` (рис. 10.13) вы можете также узнать IP-адрес интерфейса, маску сети, аппаратный MAC-адрес сетевой платы (HWaddr), количество принятых и переданных байтов (RX и TX, соответственно).
11. Интерфейс eth0 поднят, теперь мы пропингуем свой узел по IP-адресу (рис. 10.14):

    # `ping 192.168.1.7`

```
Сеанс  Правка  Вид  Закладки  Настройка  Справка
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:83 errors:0 dropped:0 overruns:0 frame:0
          TX packets:83 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:5270 (5.1 Kb)  TX bytes:5270 (5.1 Kb)

[root@localhost /]# ping 192.168.1.7
PING 192.168.1.7 (192.168.1.7) 56(84) bytes of data.
64 bytes from 192.168.1.7: icmp_seq=1 ttl=64 time=0.048 ms
64 bytes from 192.168.1.7: icmp_seq=2 ttl=64 time=0.044 ms
64 bytes from 192.168.1.7: icmp_seq=3 ttl=64 time=0.043 ms
64 bytes from 192.168.1.7: icmp_seq=4 ttl=64 time=0.042 ms
64 bytes from 192.168.1.7: icmp_seq=5 ttl=64 time=0.044 ms
64 bytes from 192.168.1.7: icmp_seq=6 ttl=64 time=0.043 ms
64 bytes from 192.168.1.7: icmp_seq=7 ttl=64 time=0.042 ms
64 bytes from 192.168.1.7: icmp_seq=8 ttl=64 time=0.043 ms
64 bytes from 192.168.1.7: icmp_seq=9 ttl=64 time=0.043 ms
64 bytes from 192.168.1.7: icmp_seq=10 ttl=64 time=0.041 ms
64 bytes from 192.168.1.7: icmp_seq=11 ttl=64 time=0.043 ms
64 bytes from 192.168.1.7: icmp_seq=12 ttl=64 time=0.042 ms
64 bytes from 192.168.1.7: icmp_seq=13 ttl=64 time=0.042 ms
Shell
```

Рис. 10.14. ping 192.168.1.7

Если ошибок не произошло, можно пропинговать удаленный узел, например, ваш шлюз. Если произошла ошибка при пинговании удаленного узла, это еще не означает, что ошибка — в конфигурации вашего компьютера. Вполне может быть, что удаленный компьютер просто выключен.

12. Напоследок пропингуйте узел, находящийся за пределами вашей сети:

    ```
    # ping www.mail.ru
    ```

    Этим вы убьете сразу двух зайцев. Во-первых, убедитесь, что работает служба DNS — ведь перед тем, как пинговать, нужно получить IP-адрес удаленного узла. Во-вторых, убедитесь, что маршрутизация нормально работает и у вас есть доступ к Интернету. Если же пропинговать удаленный узел не удалось, лучше всего обратиться за помощью к администратору сети — вполне возможно, что пока он не открыл доступ к Интернету для вашего компьютера.

Изменить параметры сетевого интерфейса можно с помощью конфигуратора DrakConf, для запуска которого введите такую же команду (соблюдайте регистр):

```
# DrakConf
```

10.5.2. Настройка сети в ASP Linux

Последовательность действий по настройке сети в ASP Linux такая же, как и в случае с Linux Mandrake, только используются другие конфигураторы.

1. Первым делом, убедитесь, что подключен сетевой кабель и активен интерфейс lo (команда `ifconfig`). Если интерфейс lo не активен, запустите сервис network.
2. После этого введите команду:

    ```
    # system-config-network-druid
    ```

3. В появившемся окне выберите **Соединение Ethernet** и нажмите **Вперед** (рис. 10.15).
4. Следующий шаг — это выбор сетевой платы (рис. 10.16). Выберите сетевую плату, соединяющую вас с нужной сетью. Если у вас всего одна сетевая плата, выбирать вам будет не из чего.
5. Теперь введите параметры IP: IP-адрес, маску сети и IP-адрес шлюза по умолчанию (рис. 10.17).
6. На этом настройка сетевого интерфейса завершена. Проверьте введенные вами данные и, если все правильно, нажмите кнопку **Применить**. Появится основное окно конфигуратора сети system-config-network, в котором будет отображен только что созданный вами интерфейс (рис. 10.18).

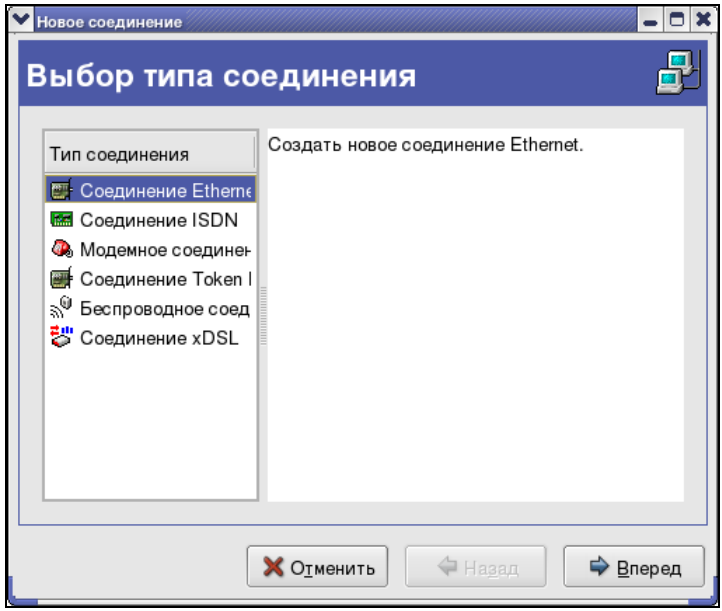

Рис. 10.15. Создание Ethernet-соединения

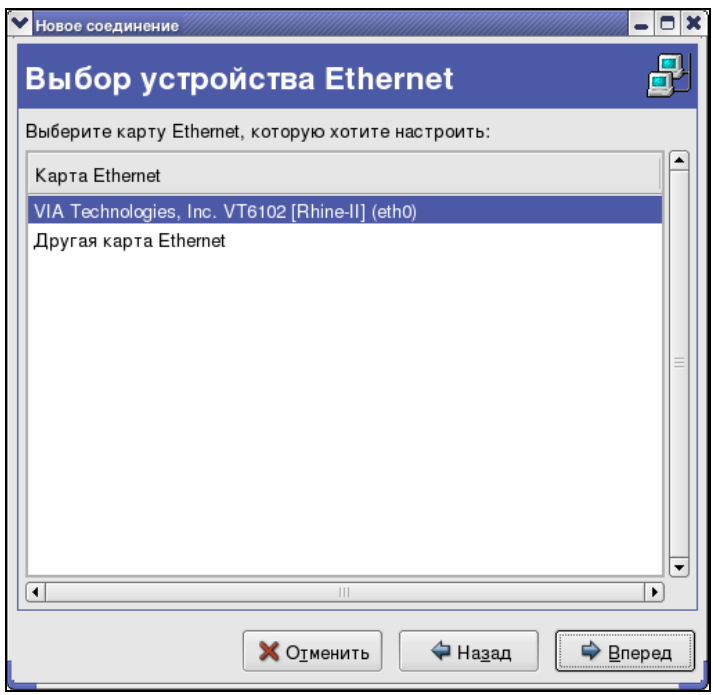

Рис. 10.16. Выбор сетевой платы

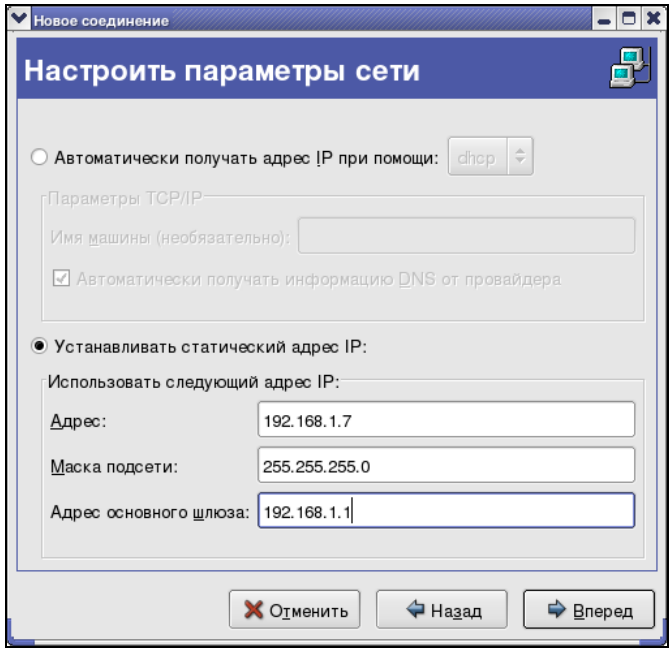

Рис. 10.17. Ввод параметров IP

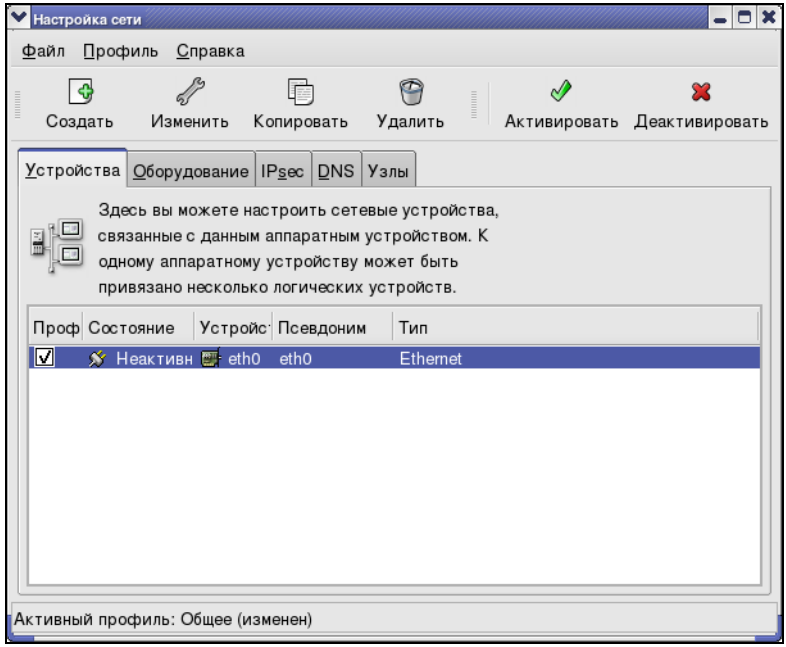

Рис. 10.18. Основное окно конфигуратора сети

7. Сразу после настройки сетевой интерфейс неактивен. Нажмите кнопку **Активировать** для его активации. Изменить параметры интерфейса можно, нажав кнопку **Изменить** (рис. 10.19). В появившемся окне вы сможете изменить различные параметры IP, в том числе выбрать использование протокола DHCP для автоматического конфигурирования интерфейса.

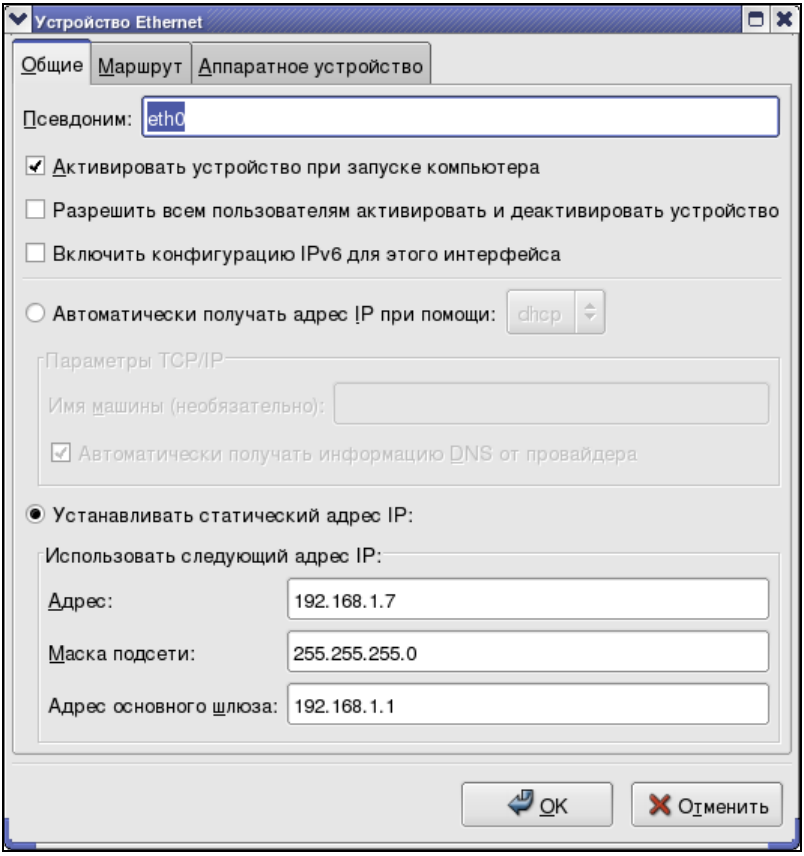

Рис. 10.19. Изменение параметров устройства

Конфигуратор DrakConf позволяет установить параметры DNS сразу при конфигурировании сетевого интерфейса. С одной стороны, это удобно. С другой — неправильно, потому что установки DNS общие для всех интерфейсов. Если вы укажете одни параметры DNS при настройке одного интерфейса и совершенно другие параметры DNS при настройке другого интерфейса, последние указанные параметры заменят параметры, указанные ранее. Разработчики Fedora Core (заслуг разработчиков ASP в конфигу-

раторе system-config-network замечено не было) поступили правильно — они вынесли параметры DNS на отдельную страницу конфигуратора. Теперь ясно, что параметры одни для всех, а не разные для каждого интерфейса, как можно было подумать в Linux Mandrake (рис. 10.20).

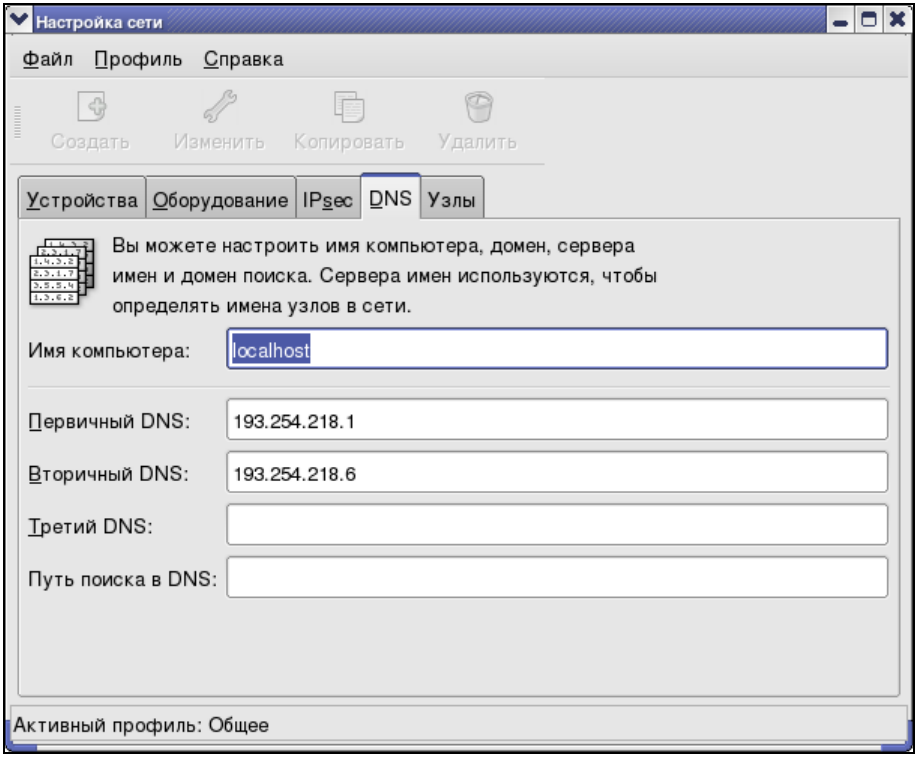

Рис. 10.20. Редактирование параметров DNS

На вкладке DNS (рис. 10.20) вы можете установить имя локального узла, IP-адреса трех серверов DNS (при непосредственной правке файла /etc/resolv.conf можно записать четыре директивы nameserver), а также указать путь поиска домена (это директива search).

Вкладка **Узлы** (рис. 10.21) предоставляет вам возможность редактирования файла /etc/hosts, в котором хранятся соответствия IP-адресов доменным именам. В данный файл для ускорения процесса разрешения доменного имени можно внести IP-адреса, к которым вы обращаетесь чаще всего, например, **www.mail.ru**, **www.google.com** и т. д. Только не забывайте со временем обновлять эту информацию, а то IP-адреса могут со временем измениться.

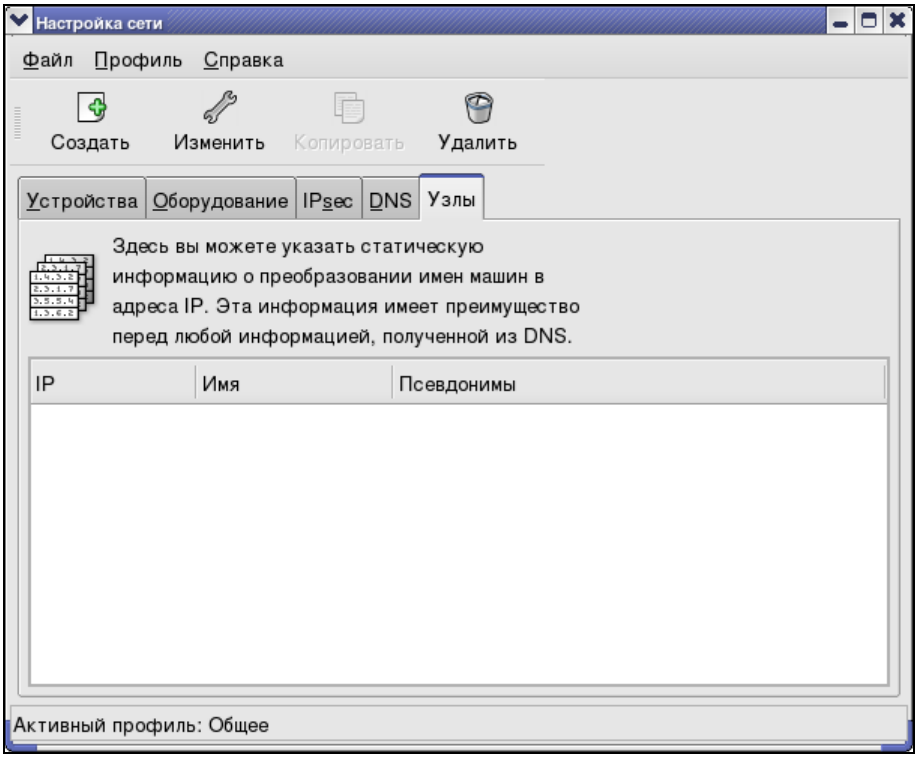

Рис. 10.21. Редактирование файла /etc/hosts

Для добавления записи в файл /etc/hosts нажмите кнопку **Создать**. Появится небольшое окно (рис. 10.22), в котором нужно будет ввести IP-адрес узла, его доменное имя и псевдоним (обычно — сокращенное имя). Например, если имя узла будет **alex.mycompany.com.ru**, то сокращенное имя можно установить как alex.

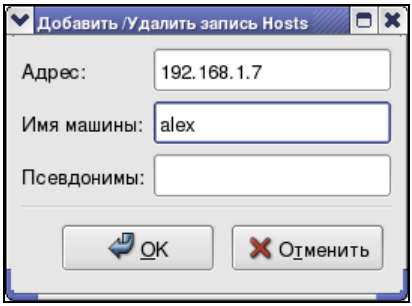

Рис. 10.22. Добавление записи в /etc/hosts

root@localhost:~ - Shell - Konsole

Сеанс Правка Вид Закладки Настройка Справка

```
[root@localhost ~]# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0D:87:88:BC:96
          inet addr:192.168.1.7  Bcast:192.168.1.255  Mask:255.255.255.0
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)
          Interrupt:11 Base address:0xe000

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:2 errors:0 dropped:0 overruns:0 frame:0
          TX packets:2 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:100 (100.0 b)  TX bytes:100 (100.0 b)

[root@localhost ~]# ping 192.168.1.7
PING 192.168.1.7 (192.168.1.7) 56(84) bytes of data.
64 bytes from 192.168.1.7: icmp_seq=0 ttl=64 time=0.068 ms
64 bytes from 192.168.1.7: icmp_seq=1 ttl=64 time=0.049 ms
64 bytes from 192.168.1.7: icmp_seq=2 ttl=64 time=0.047 ms

--- 192.168.1.7 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 1999ms
rtt min/avg/max/mdev = 0.047/0.054/0.068/0.012 ms, pipe 2
[root@localhost ~]#
```

Shell

Рис. 10.23. Тестирование соединения

Настало время проверить работу сетевого интерфейса. Для этого сначала введем команду `ifconfig`, чтобы убедиться, что сетевой интерфейс активен, а затем пропингуем сетевой интерфейс по адресу (рис. 10.23).

Глава 11

Linux и flash

В последнее время очень популярна flash-память. Уже сегодня flash-диски (они же USB-диски), построенные с использованием flash-памяти, практически вытеснили обычные дискеты: они очень компактны и позволяют хранить довольно большие объемы информации. Сегодня никого не удивишь небольшим брелком, позволяющим хранить до 1 Гбайт. Обычно популярны flash-диски емкостью до 256 Мбайт, поскольку flash-диски от 512 Мбайт на фоне братьев своих меньших стоят дороже.

Принцип использования flash-диска очень прост. Достаточно подключить его к шине USB, и через несколько секунд Windows XP определит диск. После этого с ним можно будет работать как с обычным диском. Да, flash-диски не очень шустры, но молниеносной реакции от них никто и не ожидает — во всяком случае, они выглядят настоящими спринтерами на фоне обычных дискет.

А как же дело с flash-дисками в Linux? Вот с этим нам и предстоит разобраться в этой главе. Но прежде чем приступить, нужно отметить, что технология flash-памяти нашла свое применение в различных портативных устройствах — от мобильных телефонов до цифровых фотоаппаратов. Вы можете подключить телефон к компьютеру и записывать на него мелодии и картинки. Аналогичная ситуация и с цифровым фотоаппаратом: когда вы фотографируете, то фотографии и видеоролики записываются на встроенную в фотоаппарат flash-память. Потом вам нужно подключить его к компьютеру и просто скопировать фотографии. Вы также можете записать фотографии (или другие файлы — не имеет значения) на фотоаппарат, используя встроенную flash-память как большую дискету — для переноса своих файлов.

В этой главе мы поговорим о подключении к Linux цифрового фотоаппарата, но принципиально разницы нет, что подключать — ведь если устройство использует flash-память, то алгоритм работы будет одинаковый.

11.1. История создания цифрового фотоаппарата

Вы не поверите, но все началось… с кофеварки. В самом начале 1990-х годов в одной из лабораторий Кембриджа была всего лишь одна кофеварка на 20 человек. Да, это не как у нас — в каждом кабинете свой чайник. Чтобы попить кофе, нужно было перейти с одной части здания в другую. Сами понимаете, за это время кто-нибудь из коллег мог выпить весь кофе. Что же делать? Я бы купил еще одну кофеварку, но ученые на то и ученые, чтобы придумывать нестандартные решения и что-нибудь изобретать.

Один из компьютеров лаборатории был оснащен устройством видеозахвата. Пришлось использовать подручные средства. Ими стали этот компьютер с его полезным устройством, а также видеокамера, которую направили на кофеварку — чтобы посмотреть, есть ли в ней кофе или нет. Камера захватывала изображение (1 кадр) и передавала на сервер. Спустя каждые 3 минуты изображение обновлялось. Удобно? Конечно. По крайней мере, видно, сколько кофе осталось, и можно просчитать, успеешь дойти или нет.

Так родилась первая Web-камера — первая заметка о ней была опубликована в январе 1992 г. в журнале "Comm Week".

Технология цифрового фото не стояла на месте и развивалась, а разработчики делали все возможное, чтобы цифровые фотоаппараты стали такими, какие они сейчас есть.

11.2. Принцип действия цифрового фотоаппарата

Сердцем любого цифрового фотоаппарата является светочувствительный сенсор, позволяющий преобразовывать свет в электрические сигналы, с которыми потом можно работать как с обычными компьютерными сигналами. Существуют два вида сенсоров: CCD-сенсоры и CMOS-сенсоры. Первые обеспечивают более качественную цветовую передачу и стоят дороже. Разница между CDD- и CMOS-сенсорами заключается в способе накопления и передачи заряда, а также в способе преобразования его в аналоговое напряжение.

Кроме светочувствительного сенсора, обязательным является наличие оптического объектива (это должно быть понятным — на то он и фотоаппарат). Но поскольку это цифровой фотоаппарат, нам нужно аналоговые электриче-

ские сигналы, получаемые от сенсора, преобразовать в цифровые данные. Для этого используется аналого-цифровой преобразователь (АЦП).

Полученное цифровое изображение нужно где-то хранить. Для этого, как мы уже знаем, используется flash-память. Но если записывать изображение без компрессии, то никакой flash-памяти не хватит, поэтому все цифровые фотоаппараты обладают платой компрессии изображения и видео. Обычно для компрессии изображений используется формат JPEG, наиболее подходящий для сжатия фото.

А как получить изображение из фотоаппарата? По USB! Следовательно, фотоаппарат должен обладать контроллером USB. Но контроллер обеспечивает не только передачу данных по интерфейсу USB, но и сжатие передаваемых данных "на лету". Сжатие передаваемой информации позволяет значительно повысить скорость передачи данных.

Если подытожить, то любой цифровой фотоаппарат состоит из следующих компонентов:

- светочувствительного сенсора;
- объектива;
- оптического фильтра;
- платы видеозахвата;
- блока компрессии;
- центрального процессора;
- оперативной памяти;
- flash-памяти;
- различных контроллеров (например, USB-контроллер).

11.3. Использование цифрового фотоаппарата в Linux

Цифровой фотоаппарат, как и любое другое USB-устройство, можно подключить к компьютеру, не выключая его. Кабель для подключения к компьютеру входит в комплект поставки. Исключения могут составить лишь мобильные телефоны — USB-кабель для подключения к компьютеру, скорее всего, нужно будет покупать отдельно.

Современные дистрибутивы поддерживают flash-устройства — вам даже не нужно ничего предпринимать. А Linux Mandrake даже автоматически монтирует USB-диски и другие flash-устройства. Но обо всем по порядку.

11.3.1. Использование flash-устройства в Linux Mandrake

Просто подключите flash-устройство (фотоаппарат, мобильный телефон или USB-диск) к USB. Буквально мгновенно создастся каталог /mnt/removable, к которому будет примонтирована flash-память вашего устройства. Через этот каталог можно будет работать с устройством как с обычным диском.

Разберемся, что же произошло:

1. Вы подключили flash-устройство.
2. Демон fam автоматически создал каталог /mnt/removable (рис. 11.1) и примонтировал к нему flash-устройство. На заметку: все falsh-устройства используют систему FAT16 (в Linux это тип файловой системы vfat).

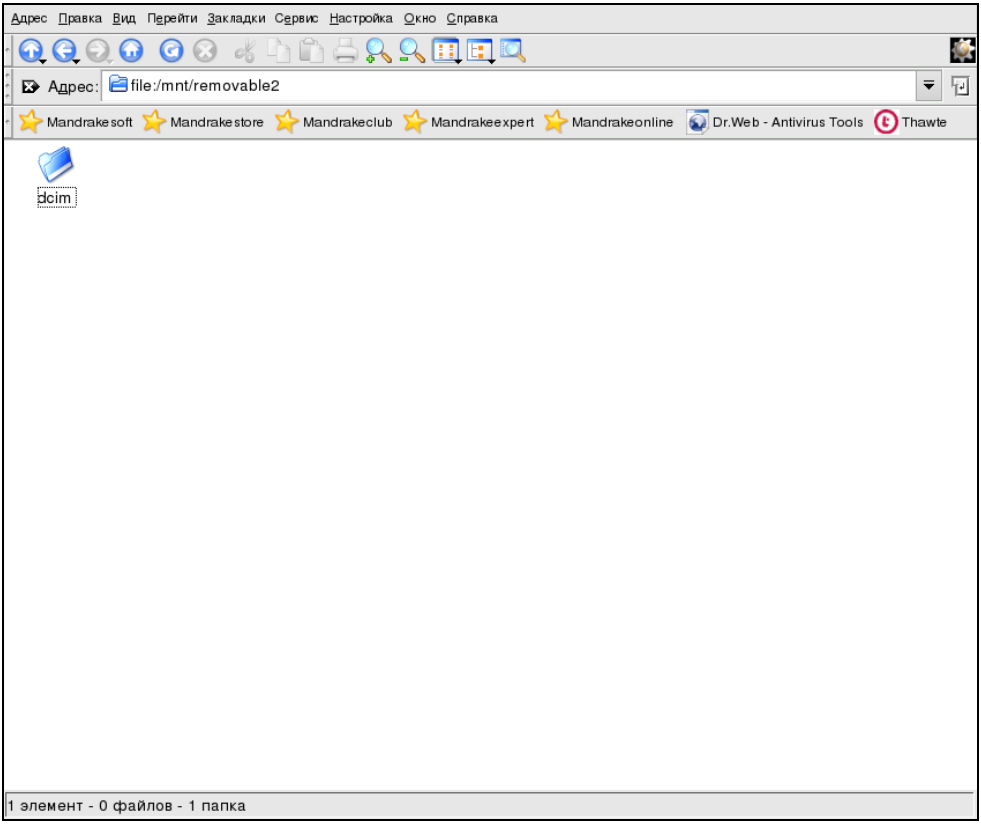

Рис. 11.1. Каталог /mnt/removable

3. Кроме того, fam создал в вашем файле /etc/fstab следующую запись:

```
none /mnt/removable supermount
dev=/dev/scsi/host1/bus0/target0/lun0/part1,fs=ext2:vfat,--
,umask=0,iocharset=koi8-u,kudzu,codepage=866 0 0
```

4. В списке USB-устройств появилось новое устройство (рис. 11.2).

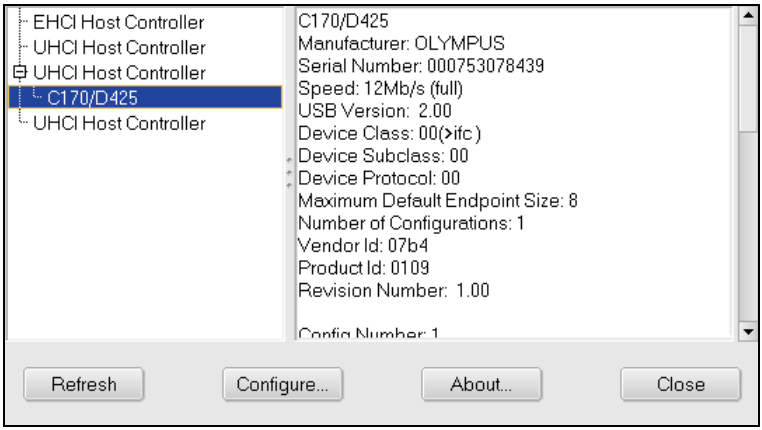

Рис. 11.2. Утилита usbview — список USB-устройств

После этого, как уже было отмечено, вы можете работать с каталогом /mnt/removable как с обычным каталогом — копировать, создавать, переименовывать, удалять файлы и каталоги. Каталог /mnt/removable нельзя размонтировать — это произойдет автоматически, как только вы отключите устройство. Запись из файла /etc/fstab будет также удалена. Она добавляется на случай, если вы подключили устройство и хотите использовать его постоянно, то есть чтобы после перезагрузки системы вы смогли сразу же продолжить работу с устройством.

Если вы, не отключая уже смонтированного flash-устройства, подключите к USB еще одно flash-устройство, то все повторится, но будет создан каталог /mnt/removable2 и т. д.

11.3.2. Использование flash-устройства в ASP Linux

В ASP Linux монтировать flash-устройство придется вручную. Может быть, это и к лучшему — вы сами будете контролировать весь процесс.

Для подключения flash-устройства выполните следующие действия:

1. Создайте каталог /mnt/camera — к нему мы будем монтировать наше устройство (рис. 11.3).

root@localhost:/mnt - Shell - Konsole

Сеанс Правка Вид Закладки Настройка Справка

```
[root@localhost mnt]# mkdir /mnt/camera
[root@localhost mnt]# mount -t vfat /dev/sda1 /mnt/camera
[root@localhost mnt]#
```

Shell

Рис. 11.3. Создание каталога и монтирование USB-диска

Обзор оборудования

Файл

Видеокарты
Жесткие диски
Звуковые платы
Клавиатуры
Сетевые устройства
Системные устройс
Указательные устро
Устройства CD-ROM
Устройства USB

Drive /dev/hda (Geom: 4998/255/63) (Model: Maxtor 6E040L0)

Drive /dev/sda (Geom: 15/255/63) (Model: OLYMPUS C170/D425)

Сведения о диске

Устройство	Старт	Конец	Размер (Мб)	Тип
hda8	1152	2137	7734	fat32
hda9	2138	3154	7978	fat32
hda10	3155	4998	14465	fat32
/dev/sda				
	1	1	< 1	Свободно
sda1	1	500	125	fat16

Рис. 11.4. Утилита hwbrowser — USB-диск подключен

2. Подключите flash-устройство. Обычно flash-устройсва распознаются системой как SCSI-диски, их наличие в системе можно просмотреть с помощью утилиты hwbrowser, она же поможет узнать имя файла устройства (рис. 11.4). В нашем случае имя файла устройства /dev/sda, а файловая система — vfat.
3. Подмонтируйте flash-устройство с помощью команды:

   ```
   # mount -t vfat /dev/sda1 /mnt/camera
   ```

4. После этого вы можете работать с каталогом /mnt/camera как обычно.

По окончании работы с устройством желательно его размонтировать — тут уже нет демона, который следит за вашими действиями и все делает за вас. А если вы еще и записывали что-то на устройство, а не только копировали с него данные, то размонтирование вообще обязательно — чтобы быть уверенными в том, что данные записаны на устройство.

Глава 12

Некоторые вопросы

В этой главе мы поговорим о том, как обнаружить и устранить причину сбоя системы, а также рассмотрим общие способы повышения производительности системы. В качестве дистрибутива мы будем использовать не ASP Linux, а какой-нибудь абстрактный дистрибутив, что сделает материал более универсальным и интересным пользователям других дистрибутивов.

12.1. Программы для тестирования оборудования

12.1.1. Программа hdparm — тестирование жесткого диска

Программа hdparm может использоваться для тонкой настройки винчестера, что иногда позволяет существенно повысить его производительность. Раньше, скажем года три назад, о данной программе говорили очень много, поскольку разработчики дистрибутивов устанавливали отказоустойчивые параметры винчестера, при которых он работал очень медленно. За пару минут можно было "разогнать" винчестер, но это не разгон в прямом смысле слова — это просто использование его всех возможностей. Просто после установки дистрибутива для всех жестких дисков устанавливались параметры, при которых бы любой жесткий диск работал без сбоя. Сейчас ситуация изменилась, и необходимость в hdparm практически отпала. Ее можно использовать в двух случаях:

- когда вы хотите просто узнать, с какой скоростью ваш винчестер обменивается данными;
- когда вам нужно понизить скорость вашего привода CD-ROM. Понижение скорости необходимо, если вам нужно прочитать диск с дефектами

поверхности (например, царапинами) — на скорости 4x вероятность чтения данных с такого диска существенно выше, чем на 24x. Помните, что сомнительные диски не рекомендуется вообще использовать: вы можете просто-напросто распрощаться с приводом.

Для тестирования производительности жесткого диска введите следующую команду (рис. 12.1):

```
# hdparm -t /dev/hda
```

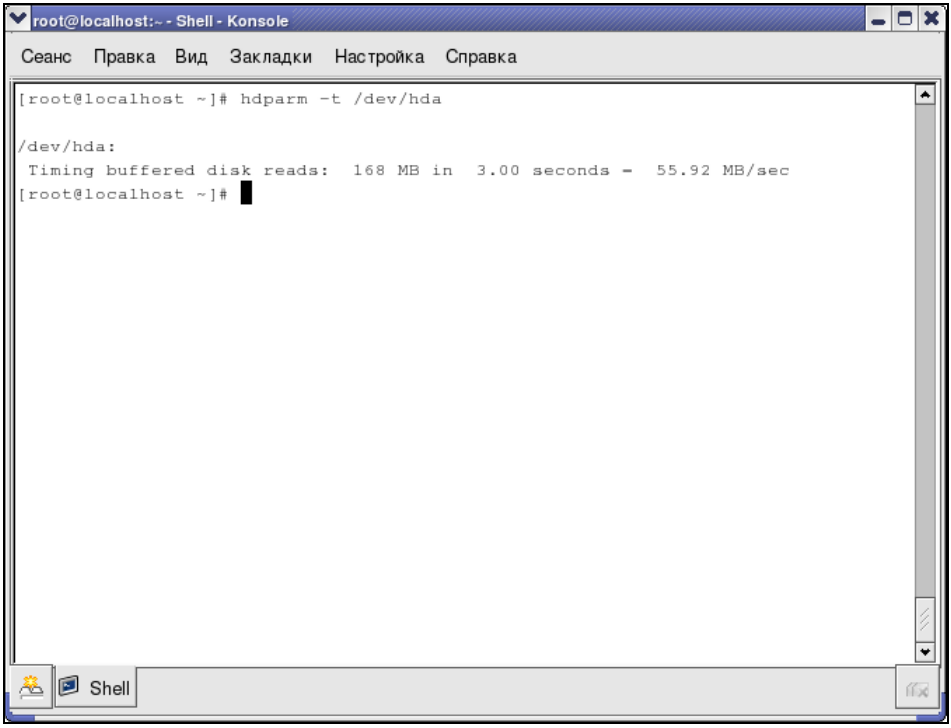

Рис. 12.1. Тестирование производительности жесткого диска

Установка скорости 4x для привода CD-ROM, подключенного как Primary Slave (/dev/hdb):

```
# hdparm -E 4 /dev/hdb
```

12.1.2. Программа fsck — проверка файловых систем

Программа fsck используется для проверки файловых систем и для коррекции ошибок файловой системы, если таковые найдутся. Основное требование для проверки файловой системы: файловая система должна быть размонти-

рована. Запуск fsck для уже смонтированной файловой системы может привести к ее разрушению — тогда уже даже и fsck не поможет. Программа fsck может использоваться для проверки файловых систем, которые поддерживаются ядром Linux.

Формат вызова программы следующий:

```
# fsck [параметры] [файловая_система]
```

Параметры, как и файловую систему, можно не указывать. Если вы не укажете файловую систему, программа начнет проверять все файловые системы, перечисленные в файле /etc/fstab, что крайне нежелательно, поскольку эти файловые системы могут быть смонтированными, и возможно разрушение файловой системы.

Последовательность проверки файловой системы должна быть следующая:

1. Размонтировать файловую систему.
2. Запустить fsck для ее проверки.

Например, для проверки файловой системы раздела /dev/hda5 сначала размонтируем его, а потом запустим fsck (рис. 12.2):

```
# umount /dev/hda5
# fsck /dev/hda5
```

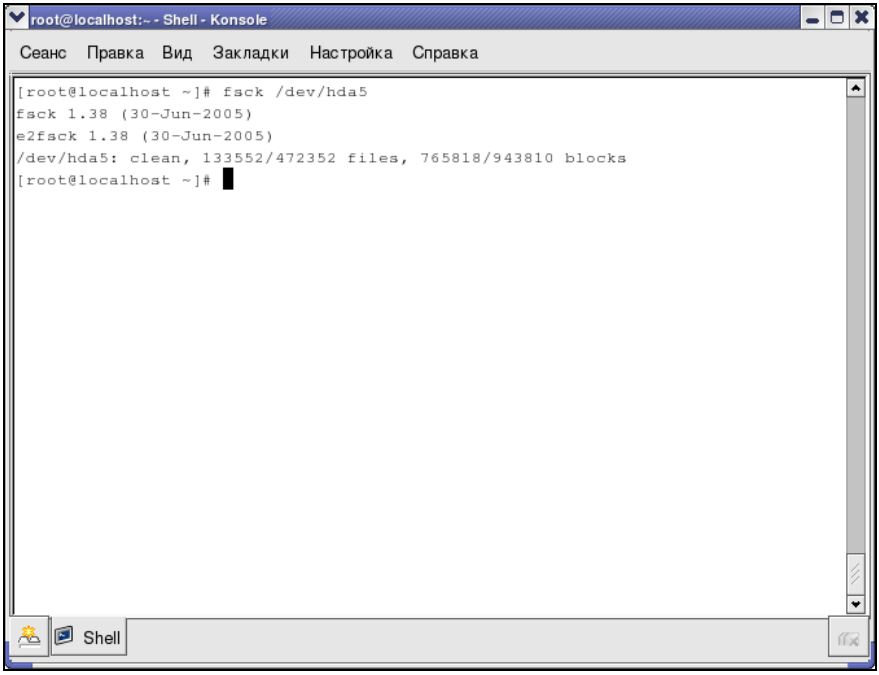

Рис. 12.2. Проверка размонтированного раздела

Но иногда мы не можем размонтировать файловую систему: например, когда нам нужно проверить корневую файловую систему. В этом случае нужно выполнить следующие действия:

1. Перезагрузиться в однопользовательском режиме.
2. Перемонтировать корневую файловую систему в режиме "только чтение".
3. Произвести проверку файловой системы.

Для перезагрузки в однопользовательском режиме перезагрузите систему (команда `reboot`), а при загрузке передайте ядру параметр `single` (рис. 12.3).

```
LILO boot:
linux           memtest          linux-ng
boot: linux single_
```

Рис. 12.3. Перезагрузка в однопользовательском режиме

В однопользовательском режиме, как и следовало ожидать, может работать только один пользователь — root (рис. 12.4).

Все сервисы выключены, так что проверке файловой системы ничто не должно помешать. Для перемонтирования файловой системы введите команду:

```
# mount -o remount ro -t ext3 /
```

Параметр `-o` команды `mount` позволяет указать различные опции. В данном случае мы указываем опции `remount` и `ro`, что означает перемонтировать в режиме "только чтение". Параметр `-t` указывает тип файловой системы — ext3, а последний параметр — это корневая файловая система (`/`). Процесс проверки смонтированной корневой файловой системы приведен на рис. 12.5.

```
Switching to new root
SELinux:  Disabled at runtime.
SELinux:  Unregistering netfilter hooks
INIT: version 2.85 booting
Setting default font (latarcyrheb-sun16):                    [  OK  ]

                Welcome to Fedora Core
                Press 'I' to enter interactive startup.
Запускается udev:                                            [  OK  ]
Инициализируется оборудование...  устройства хранения сеть [  OK  ]тово
Проверяются параметры ядра:                                  [  OK  ]
Устанавливаются часы  (localtime): Срд Фев 22 08:35:34 MSK [  OK  ]
Загружается основная раскладка клавиатуры (ru):              [  OK  ]
Устанавливается имя узла localhost.localdomain:              [  OK  ]
Проверяется корневая файловая система
/: clean, 45871/192000 files, 217782/383544 blocks
                                                             [  OK  ]
Корневая ФС перемонтируется в режиме чтения-записи:          [  OK  ]
Устанавливаются параметры Logical Volume Manager             [  OK  ]
Проверка файловых систем
                                                             [  OK  ]
Монтируются локальные ФС:                                    [  OK  ]
Активируются квоты локальных файловых систем:                [  OK  ]
Подключается пространство свопинга:                          [  OK  ]
sh-3.00# _
```

Рис. 12.4. Однопользовательский режим

```
sh-3.00# mount -o remount ro -t ext3 /
sh-3.00# fsck /
fsck 1.35 (28-Feb-2004)
e2fsck 1.35 (28-Feb-2004)
/dev/hda1 is mounted.

WARNING!!!  Running e2fsck on a mounted filesystem may cause
SEVERE filesystem damage.

Do you really want to continue (y/n)? yes

/: recovering journal
/: clean, 45870/192000 files, 217786/383544 blocks
sh-3.00# _
```

Рис. 12.5. Проверка корневой файловой системы

12.1.3. Программа badblocks — проверка на наличие плохих блоков

Программа badblocks позволяет проверить жесткий диск на наличие сбойных блоков. Проверка выполняется очень просто:

```
# badblocks -v <имя_устройства>
```

Например:

```
# badblocks -v /dev/hda
```

Параметр `-v` включает подробный режим работы — о каждом действии программа badblocks будет выводить отчет (рис. 12.6). Желательно выполнять программу в однопользовательском режиме, чтобы ее работе ничто не мешало.

```
sh-3.00# badblocks -v /dev/hda
Checking blocks 0 to 1782579
Checking for bad blocks (read-only test): done
Pass completed, 0 bad blocks found.
sh-3.00# _
```

Рис. 12.6. Использование программы memtest86

12.1.4. Программа memtest86 — проверка оперативной памяти

Иногда мы произносим слово "глючит", даже не задумываясь о его смысле. Например, "глючит" оперативка. А что же это такое — "глюк"? Глюк — это неисправность устройства, проявляющаяся при определенных условиях или в определенных режимах работы устройства.

Конечно, глюки бывают не только аппаратными, но и программными — это сбои программ при определенных условиях. Довольно часто программные сбои происходят именно из-за мелких неисправностей аппаратуры. Например, причиной неожиданной ошибки при компиляции ядра может стать неисправный модуль оперативной памяти. Ведь не секрет, что модули оперативной памяти имеют модульную структуру. Возможно, что при обычной работе с системой один из модулей (неисправный) не используется, поскольку его ресурсы не востребованы. Но когда система использует свои ресурсы на все 100%, происходит сбой.

Для тестирования оперативной памяти используется программа memtest86. Программа memtest86 работает не как обычная программа. После установки программы устанавливается специальный загрузочный образ, а для запуска программы вы должны перезапустить компьютер и выбрать этот образ из меню вашего загрузчика. Но обо всем по порядку.

1. Установите пакет memtest86 (рис. 12.7).

```
[root@localhost mnt]# [root@localhost RPMS]# rpm -ihv memtest*
предупреждение: memtest86+-1.26-2.i386.rpm: подпись DSA V3: NOKEY, key ID 4f2a6f
d2
Подготовка...      ########################################### [100%]
   1:memtest86+              ########################################### [100%]
```

Рис. 12.7. Установка memtest86

2. Сразу после установки пакета выполните команду `memtest-setup` — данная команда настроит загрузочный модуль (рис. 12.8).

```
[root@localhost etc]# memtest-setup
Setup complete.
```

Рис. 12.8. Установка загрузочного модуля

Теперь проверим изменения, сделанные программой memtest-setup в наших конфигурационных файлах. Программа изменяет конфигурационные файлы загрузчиков /etc/lilo.conf и /boot/grub/grub.conf.

3. Начнем с GRUB. Откройте файл /boot/grub/grub.conf в любом текстовом редакторе и убедитесь, что все нормально.
4. А вот в конфигурационном файле /etc/lilo.conf придется сделать небольшие изменения: программа создала загрузочный образ с меткой 1.26, а мне больше нравится метка memtest (как в случае с GRUB), поэтому пришлось немного изменить файл конфигурации (рис. 12.9).

   Ясно, что если у вас основной загрузчик — GRUB, то в файл /etc/lilo.conf вы можете и не заглядывать, и наоборот. Напомню, что если вы используете LILO, то после изменения файла конфигурации не забудьте ввести команду `lilo` для переустановки загрузчика (команду нужно вводить от имени root):

   ```
   # lilo
   ```

5. Вот теперь начинается самое интересное. Перезагрузите систему и выберите из меню загрузчика только что созданную программой memtest-setup

запись. Запустится программа memtest86 (рис. 12.10), которая сразу начнет тестировать вашу оперативную память, и если произойдет ошибка, программа непременно сообщит вам об этом.

```
grub.conf   [----]  0 L:[  1+13  14/ 21] *(452 / 669b)= t 116 0x74
# grub.conf generated by anaconda
#
# Note that you do not have to rerun grub after making changes to this file
# NOTICE:  You do not have a /boot partition.  This means that
#          all kernel and initrd paths are relative to /, eg.
#          root (hd0,0)
#          kernel /boot/vmlinuz-version ro root=/dev/hda1
#          initrd /boot/initrd-version.img
#boot=/dev/hda
default=1
timeout=5
splashimage=(hd0,0)/boot/grub/splash.xpm.gz
hiddenmenu
title Memtest86+ (1.26)
        root (hd0,0)
        kernel /boot/memtest86+-1.26 ro root=LABEL=/
title Fedora Core (2.6.9-1.667)
        root (hd0,0)
        kernel /boot/vmlinuz-2.6.9-1.667 ro root=LABEL=/
        initrd /boot/initrd-2.6.9-1.667.img

1Помощь 2Запись 3Блок  4Замена 5Копия  6Перемес7Поиск  8Удалить9МенюМС 10Выход
```

Рис. 12.9. Редактирование файла /etc/lilo.conf

```
      Memtest86+ v1.26        | Pass 88% ##################################
AMD Duron (0.13) 1599 Mhz     | Test 80% ################################
L1 Cache:  128K  9809MB/s     | Test #6  [Modulo 20, ones & zeros, cached]
L2 Cache:   64K  4110MB/s     | Testing:   96K -  128M  128M
Memory  :  128M   589MB/s     | Pattern: 00000000-16
Chipset : Intel i440BX

 WallTime   Cached  RsvdMem   MemMap   Cache  ECC  Test  Pass  Errors ECC Errs
 ---------  ------  -------  --------  -----  ---  ----  ----  ------ --------
   0:04:14    128M     212K  e820-Std     on   off   Std     0       0        0
 ---------------------------------------------------------------------------

(ESC)Reboot  (c)configuration  (SP)scroll_lock  (CR)scroll_unlock
```

Рис. 12.10. Программа memtest86 в процессе работы

Да, на рисунке есть определенные несоответствия, например, сказано, что в тестируемой системе процессор AMD, а чипсет Intel 440 BX, но не подумайте, что это "глюк" программы. Просто у меня memtest86 запущен в виртуальной машине vmware, поскольку без этого не сделаешь скриншот окна программы.

12.1.5. Другие программы для диагностики

Кроме программ hdparm, fsck, badblocks и memtest86, вы можете использовать следующие программы (табл. 12.1).

Таблица 12.1. *Утилиты для тестирования оборудования*

Программа	Описание
kudzu/harddrake	Определение новых устройств
hddtemp	Выводит температуру винчестера (нужно заметить, что эта программа работает далеко не со всеми жесткими дисками)
smartmontools	Мониторинг SMART-устройств. Как правило, это жесткие диски с возможностью самодиагностики. Такой жесткий диск сам скажет, сколько ему осталось, чтобы его отказ не стал для вас неприятной неожиданностью — SMART-диски очень дорогие
bonnie	Тестирование производительности винчестера
cpuburn	Тестирование процессора
screentest	Тестирование/настройка монитора

Наиболее интересной является программа cupburn, доступная по адресу:

ftp://rpmfind.net/linux/Mandrake/10.0/contrib/i586/cpuburn-1.4-5mdk.i586.rpm.

Программа cpuburn выполняет так называемый стресс-тест процессора. Суть его заключается в том, что программа загружает ваш процессор на все 100% и продолжает работу в таком режиме. Если у вас в системе что-то не то, например, из-за плохого вентилятора греется процессор, то это обязательно проявится во время стресс-теста в виде сбоя.

12.2. Повышение производительности системы

Иногда Linux кажется эдаким огромным и неповоротливым монстром, особенно после установки. Давайте разберемся, за счет чего мы сможем увеличить производительность системы.

- Первое, что можно сделать — это отключить неиспользуемые сервисы. После этого система должна загружаться намного быстрее.

- Второе — это увеличить размер и изменить настройки виртуальной памяти системы. При правильном подходе можно добиться существенного прироста в производительности.
- Напоследок можно поэкспериментировать с некоторыми параметрами ядра. Обо всем этом мы сейчас и поговорим.

Заранее нужно отметить, что настройка будет производиться на примере домашнего дистрибутива — никаких серверов, все рассчитано на обычного пользователя.

12.2.1. Отключение ненужных сервисов

Каждый запущенный сервис использует не только процессорное время, но и оперативную память. Если сервис вам нужен, то это его оправдывает. Но в большинстве случаев в системе запущены такие сервисы, о которых администратор системы и не знает, не говоря уже об использовании этих сервисов. Кроме того, каждый запущенный с параметрами по умолчанию сервис — это потенциальная "дыра" в системе безопасности. Ведь в большинстве случаев параметры по умолчанию небезопасны — они делаются по принципу "лишь бы работало". Если вам нужен сервис, значит, уделите время и настройте его. Если же сервис вам не нужен, просто выключите его.

В ASL Linux (и в Fedora Core) для настройки сервисов используется конфигуратор system-config-services (рис. 12.11), а в Mandrake — drakxservices (рис. 12.12).

Нужно отметить одну особенность конфигуратора system-config-services: для сохранения изменений вам нужно нажать кнопку **Сохранить**, в то время как конфигуратор drakxservices сохраняет изменения автоматически.

В последней версии ASP Linux 11 можно выделить следующие ненужные сервисы:

- acpid — управляет ACPI-событиями, по большому счету, он просто не нужен;
- apmd — нужен только на ноутбуках;
- anacron, atd, crond — демоны-планировщики, которые запускают указанные пользователем команды в определенное время. Домашнему пользователю они вряд ли нужны, во всяком случае, три сразу. По крайней мере, два можно отключить с чистой совестью;
- Bluetooth — если вы не собираетесь использовать технологию Bluetooth, данный сервис можете просто выключить;
- cpuspeed, haldaemon — не вдаваясь в подробности, просто отключите эти сервисы;

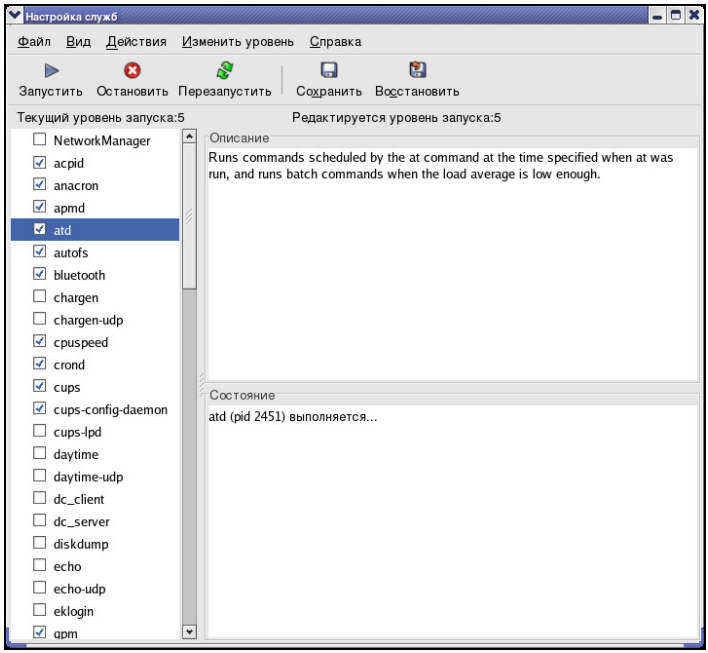

Рис. 12.11. Конфигуратор system-config-services

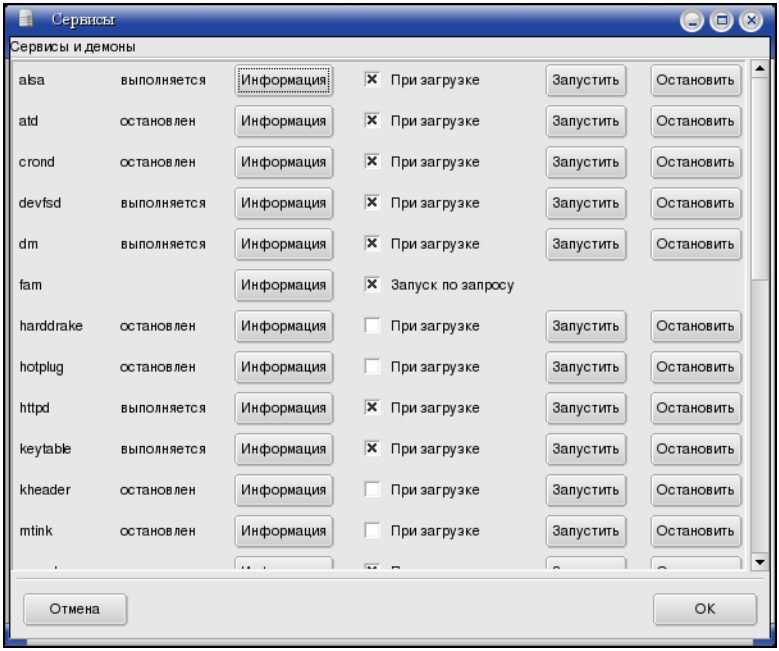

Рис. 12.12. Конфигуратор drakxservices

- cups* — система печати CUPS (Common Unix Printing System). Нужна только, если есть принтер;
- isdn — сервис поддержки ISDN-линий. Если у вас нет ISDN, выключите этот сервис;
- irqbalance — нужен только на SMP-машинах (многопроцессорных машинах);
- kudzu (в Mandrake — harddrake2) — сервис определения новых устройств. В целях экономии времени при загрузке системы, его можно выключить, а запускать вручную после установки нового устройства;
- lm_sensors — используется для мониторинга различных параметров системы (например, температуры процессора). Для правильной настройки этого сервиса нужно потратить много времени (к тому же не все "железо" его поддерживает), поэтому просто выключите его;
- messagebus — "шина" сообщений, выключите ее;
- mDNSResponder, nifd — можете просто выключить эти два сервиса;
- mdmonitor — используется для мониторинга программных RAID-массивов;
- netfs — обеспечивает поддержку различных сетевых файловых систем (в том числе и для поддержки SMB). Нужен в локальной сети, где есть необходимость в использовании сетевых файловых систем (это никак не относится к протоколу FTP);
- pcmcia — нужен для поддержки PCMCIA-карт. Если у вас не ноутбук, можете смело выключить этот сервис;
- portmap — обеспечивает маппинг портов. Домашнему пользователю не нужен, а вот на сервере пригодится;
- rpc* — поддержка удаленного вызова процедур RPC (Remote Procedure Call), в большинстве случаев данный сервис не востребован;
- smartd — нужен для поддержки SMART-устройств. Если у вас нет таких устройств, можете выключить этот сервис;
- sshd — используется для безопасного удаленного к консоли системы. В большинстве случаев не нужен;
- sendmail (или postfix) — используется для организации собственного SMTP-сервера, то есть для сервера отправки сообщений электронной почты. Довольно сложен в настройке, поэтому лучше пока выключить его. Когда у вас будет необходимость в собственном SMTP-сервере, включить сервис особых проблем не составит;
- rhnsd — можно смело выключить.

Чтобы почувствовать, на сколько сократилось время загрузки системы, ее нужно перезагрузить:

```
# reboot
```

12.2.2. Оптимизация виртуальной памяти

Операционная система Linux не очень требовательная к памяти — для нормальной работы даже шлюза небольшой сети вполне хватит 64 Мбайт оперативной памяти. Не верите? Посмотрите на рис. 12.13 — из 128 Мбайт использовано всего 33 Мбайт, а своп вообще не используется.

```
[root@localhost ~]# free
             total       used       free     shared    buffers     cached
Mem:        126284      33244      93040          0       4584      16152
-/+ buffers/cache:      12508     113776
Swap:       240964          0     240964
[root@localhost ~]# _
```

Рис. 12.13. Команда **free** — сведения об использовании оперативной памяти

Но это только в том случае, если не запущена система X Window. После ее запуска Linux превращается в настоящего обжору, съедающего десятки мегабайт памяти. Сама система X Window тоже не особенно требовательна к памяти, чего не скажешь о графических интерфейсах GNOME и KDE. При использовании GNOME или KDE для комфортной работы необходимо минимум 192 Мбайт оперативной памяти.

Ваша система может работать, мягко говоря, не очень быстро только потому, что ей не хватает оперативной памяти.

Сейчас попытаемся определить, хватает ли вам ОЗУ. Запустите те программы, с которыми вы чаще всего работаете — OO Writer, OO Calc, xmms, GIMP. Не все сразу, а только те, которые вы часто используете одновременно. Затем введите команду `free` и посмотрите, сколько оперативной памяти у вас

свободно. Также обратите внимание на "остаток" области подкачки (swap). Возможно, вам пора покупать еще один модуль оперативной памяти. Временно, пока вы его не купили, можно создать файл подкачки, что несколько повысит производительность системы. Хочу обратить ваше внимание на то, что это временная мера — ведь производительность жесткого диска существенно ниже производительности оперативной памяти, следовательно, даже если вы добавите 1 Гбайт к области подкачки, это все равно не сравнится с одним настоящим модулем памяти на 256 Мбайт.

Файл подкачки можно создать с помощью команды:

```
# dd if=/dev/zero of=/swap-file bs=1m count=128
```

Данной командой мы создали файл /swap-file размером 128 Мбайт. После этого нужно отформатировать его как своп-файл:

```
# mkswap /swap-file 131072
```

131 072 — это размер в байтах нашего файла (128×1024). Теперь выполните команду:

```
# swapon /swap-file
```

Данная команда разрешает системе использовать файл /swap-file в качестве файла подкачки. Чтобы не вводить эту команду каждый раз при загрузке системы, добавьте ее вызов в сценарии инициализации системы.

Одного добавления своп-файла мало. Нужно еще оптимизировать работу системы свопинга с помощью коэффициента подкачки. Значение этого коэффициента хранится в файле /proc/sys/vm/swappiness. Минимальное значение коэффициента — 0, максимальное — 100. Значение по умолчанию 70.

Теперь о том, как правильно выбрать оптимальное значение. Если вы в основном работаете с небольшими программами и часто переключаетесь между ними, можно установить значение, меньшее 50, например, 40 или даже 30. В этом случае переключение между приложениями будет мгновенным, но замедлится их работа. Но поскольку эти приложения небольшого размера, то вы этого не заметите.

Если же вы в основном работаете на протяжении дня с громоздкими приложениями, например, Open Office, или занимаетесь обработкой изображений в GIMP, вам лучше установить значение коэффициента, превышающее 70, например, 80 или даже 85. В этом случае переключение между приложениями будет медленное, зато ваше основное приложение будет работать быстро.

Изменить значение коэффициента можно с помощью команды:

```
# echo "значение" > /proc/sys/vm/swappiness
```

Например:

```
# echo "50" > /proc/sys/vm/swappiness
```

12.2.3. Оптимизация ядра

Оптимизация ядра заключается в том, что просматриваются все настройки ядра и отключается все ненужное. После чего ядро нужно перекомпилировать и установить. Данное мероприятие занимает довольно много времени — минимум полчаса, а может быть, и час на изучение всех возможных и невозможных опций ядра, плюс время на перекомпиляцию ядра и модулей. А если что-то пойдет не так и система откажется загружаться с новым ядром, то придется все повторить сначала.

Мы пойдем по другому пути. Сейчас мы попытаемся с помощью параметра ядра elevator установить нужный нам алгоритм работы ядра, что позволит существенно повысить производительность системы. Допустимы следующие значения этого параметра:

- none — значение по умолчанию;
- as — упреждающее планирование;
- cfq — "честная очередь";
- deadline — планирование крайних сроков.

Для домашнего компьютера больше подойдут значения as и cfq. В первом случае ядро будет пытаться "угадать" ход программы, а именно какую операцию ввода/вывода программа "захочет" выполнить в следующий раз. Если ядро будет правильно "угадывать", то производительность системы должна существенно увеличиться. Ясно, что работа данного алгоритма очень зависит от логики программы.

Во втором случае (значение cfq) ядро будет равномерно планировать операции ввода/вывода. Данный алгоритм будет работать лучше первого в случае с запутанной логикой программы, когда невозможно предугадать ее следующую операцию.

Последнее значение (deadline) больше подходит для сервера, чем для рабочей станции, поэтому существенного прироста от него не ждите.

При загрузке передать параметр ядра можно так:

```
linux elevator=значение
```

Чтобы не вводить параметр каждый раз при загрузке, добавьте его в файл конфигурации загрузчика. Если у вас GRUB, то одна из секций конфигурационного файла /etc/grub/grub.conf будет выглядеть так (листинг 12.1).

Листинг 12.1. Фрагмент файла /etc/grub/grub.conf

```
title Linux
root (hd1,0)
kernel /boot/vmlinuz-2.6.9 ro root=/dev/hda1 elevator=as
```

Если у вас LILO, отредактируйте ваш файл /etc/lilo.conf так (листинг 12.2).

Листинг 12.2. Фрагмент файла /etc/lilo.conf

```
image=/boot/vmlinuz-2.6.9
  label=Linux
  root=/dev/hda1
  append="elevator=as"
```

После изменений файла /etc/lilo.conf не забудьте выполнить команду lilo для того, чтобы изменения вступили в силу.

12.3. Восстановление системы после сбоя

Всему есть своя причина. Если произошел сбой, следовательно, была причина сбоя. Причиной может стать или ошибка программного обеспечения или отказ "железа". Исходя из этого, можно выделить две группы сбоев: программные и аппаратные. Последние можно смело назвать аппаратно-программными, поскольку вследствие отказа аппаратуры довольно часто происходят программные сбои. Самый простой пример — это отказ винчестера, вследствие которого программа не может записать или прочитать данные — происходит программный сбой. Или же некорректная работа оперативной памяти, из-за чего очень часто происходят порою сложно объяснимые ошибки программного обеспечения.

В этом разделе мы поговорим об анализе и устранении как программных, так и аппаратных сбоев. Но сначала мы рассмотрим восстановление загрузчика Linux. Не подумайте, что загрузчик Linux ненадежен, отнюдь. Просто после переустановки Windows "затирает" загрузчик Linux и записывает вместо него свой собственный загрузчик. Ясно, что после переустановки Windows переустанавливать Linux не очень хочется.

12.3.1. Восстановления загрузчика Linux

Поскольку Linux больше не загружается, для ее загрузки нам понадобится загрузочная дискета — да, именно та, которую вы создали при установке Linux. Если у вас ее нет, значит, вам нужен загрузочный LiveCD. Иногда возможно восстановление средствами первого дистрибутивного диска, но из-за различий в разных дистрибутивах, мы этот способ рассматривать не бу-

дем. Гораздо проще купить какой-нибудь LiveCD и использовать его. Я рекомендую Gentoo LiveCD — это первый диск дистрибутива Gentoo. Закажите его прямо сейчас — не дожидайтесь очередной переустановки Linux.

Итак, рассмотрим два случая восстановления системы: первый с помощью загрузочной дискеты, а второй — с помощью LiveCD.

Используем загрузочную дискету

Обычно на загрузочной дискете находится небольшой загрузчик и ядро системы, настроенное так, чтобы загрузить именно вашу систему — вот поэтому загрузочная дискета и создается в самом конце установки. Ведь программе установки нужно записать на дискету, как минимум, сведения о корневой файловой системе и о разделе /boot.

Загрузившись с загрузочной дискеты, вы уже можете работать с системой как обычно. Но перед тем как это сделать, вам нужно восстановить загрузчик Linux — вы же не хотите постоянно загружаться с дискеты?

Поскольку все параметры загрузчика уже установлены, вам просто нужно установить сам загрузчик. Если вы использовали LILO, просто введите от имени пользователя root команду:

```
# lilo
```

Если у вас был GRUB, зарегистрируйтесь в системе как пользователь root и введите команду (имя устройства жесткого диска у вас может отличаться):

```
# grub-install /dev/hda
```

После этого введите команду reboot для перезагрузки системы.

Используем LiveCD

Тут ситуация немного сложнее — LiveCD использует свою корневую файловую систему, следовательно, мы должны заменить корневую файловую систему LiveCD своей файловой системой, а уже потом установить загрузчик — когда будет работать на своей файловой системе.
Последовательность действий такая:

1. Загрузка с LiveCD (рис. 12.14).
2. Монтирование корневой файловой системы установленной операционной системы к корневой файловой системе LiveCD.
3. Изменение корневой файловой системы.
4. Установка загрузчика.

Рис. 12.14. Загрузка с LiveCD Gentoo

Рассмотрим процедуру восстановления загрузчика Linux Mandrake, установленного в разделе /dev/hda6. После загрузки с LiveCD введите следующие команды:

```
# mkdir /old_root
# mount /dev/hda6 /old_root
```

Первая команда создает каталог /old_root, который будет использоваться в качестве точки монтирования, а вторая — монтирует устройство /dev/hda6 к указанной точке монтирования.

После этого мы можем изменить корневую файловую систему. Делается это с помощью всего одной команды:

```
# chroot /old_root
```

Теперь осталось установить загрузчик. Если у вас LILO, просто введите команду:

```
# lilo
```

А если GRUB, то команда будет чуть длиннее:

```
# /sbin/grub-install /dev/hda
```

12.3.2. Программный сбой

Прежде всего, нужно выяснить и, по возможности, устранить причину сбоя. Если это сугубо программный сбой, то причины может быть две: неправильная настройка программы (или системы), или же ошибка программы.

Неправильная настройка программы или системы

Как работала система до сбоя? Был ли подобный сбой раньше? Если ничего такого ранее вы не наблюдали и система работала как швейцарские часы, значит, скорее всего, причина в неправильной ее настройке. Вспомните, какие файлы конфигурации вы изменяли (или какие параметры устанавливали с помощью графических конфигураторов). Просто по памяти восстановите исходные значения и перезапустите сервис или службу, ставшую причиной сбоя — скорее всего, причина исчезнет. Рекомендуется перед каким-либо изменением, вносимым в файл конфигурации системы, делать резервную копию этого файла. Потом же вам будет проще восстановить исходные значения. Можно использовать и другой подход: комментировать прежние директивы/значения файла конфигурации, а под ними писать новые директивы/значения. В случае вашей ошибки вы всегда сможете восстановить исходные значения.

Ошибка программы. Журналы системы

Но это был самый простой случай, в котором причиной ошибки стали ваши действия. Но иногда бывает так, что система работала-работала, а на следующий день половина служб не запускается. В чем же причина? А тут вам может помочь только чтение журналов системы. Все журналы находятся в каталоге /var/log:

- /cron/ — журналы демона-планировщика crond;
- /daemons/ — журналы различных демонов;
- /httpd/ — журналы Web-сервера Apache;
- /kernel/ — журналы ядра (только в Linux Mandrake);
- /lpr/ — журналы системы печати LPR;
- /cups/ — журналы системы CUPS (в вашей системе может быть установлена одна из этих систем печати);
- /mail/ — журналы почтовой службы;
- /news/ — журналы NNTP-сервера и NNTP-клиентов;
- auth.log — журнал аутентификации — кто и когда входил в систему;
- boot.log — журнал загрузки системы;
- dmesg — загрузочные сообщения ядра;
- explanations — в этот журнал некоторые программы записывают свои действия, объясняя вам, какие именно изменения они произвели в вашей системе;

- messages — различные сообщения ядра (и в некоторых случаях — обычных программ);
- secure — журнал службы безопасности;
- syslog — журнал демона syslog;
- XFree86.0.log — журнал системы XFree86;
- user.log — различные сообщения программ пользовательского уровня.

Протоколирование сообщений системы и программ выполняется двумя демонами — klogd и syslogd. Первый протоколирует сообщения ядра, а второй — все остальные сообщения, поэтому никогда не отключайте эти демоны. Файл dmesg создается самим ядром при начальной загрузке системы (до запуска init).

Имена файлов журналов могут немного отличаться от приведенных, поскольку имена журналов зависят от настроек системы, в том числе и от настроек syslogd. К тому же у вас могут быть дополнительные файлы протоколов или даже каталоги, содержащие файлы протоколов — повторюсь, все зависит от настроек системы. Чтобы узнать, какие файлы протоколов у вас являются основными, откройте файл конфигурации syslogd — /etc/syslog.conf. Прочитав его, вы узнаете, какие файлы протоколирования есть в вашей системе и для чего они используются.

Но в файле конфигурации /etc/syslogd.conf перечислены далеко не все файлы протоколов. Многие серверы ведут свои журналы, имена файлов которых вы можете узнать в файле конфигурации того или иного сервера.

В каком же журнале искать ошибку? Тут нужно исходить из принципа взаимоисключения: если у вас не работает Web-сервер Apache, то искать причину нужно в каталоге /var/log/httpd/, но никак не в /var/log/mail.

Если ошибка происходит во время загрузки системы, просмотрите файл boot.log:

```
# less boot.log
```

Данную команду нужно вводить от имени root, поскольку журналы системы просматривать может только он.

Сообщения различных программ пользовательского уровня, то есть обычных программ, возможно, запущенных с привилегиями root, протоколируются в файл /var/log/user.log. В некоторых системах этого файла нет, например, он есть в Linux Mandrake, но его нет в ASP Linux (и в Fedora Core). Если данный файл у вас отсутствует, значит, сообщения пользовательских программ протоколируются в другой файл, обычно в /var/log/messages. Рассмотрим фрагмент данного файла (листинг 12.3).

Листинг 12.3. Фрагмент файла /var/log/user.log (/var/log/messages)

```
Jun 20 14:18:28 localhost rpmdrake[2573]: [RPM] libcroco1-0.4.0-1mdk in-
stalled
Jun 20 14:18:31 localhost rpmdrake[2573]: [RPM] librsvg2_2-2.4.0-1mdk
installed
Jun 20 14:18:32 localhost rpmdrake[2573]: [RPM] libpanel-applet-2_0-
2.4.2-6mdk installed
Jun 20 14:18:32 localhost rpmdrake[2573]: [RPM] libmetacity-private0-
2.6.5-2mdk installed
```

В данном случае программа rpmdrake сообщает, что она установила указанные пакеты. Если вас интересуют более подробные сведения об этом процессе, откройте файл /var/log/explanations (листинг 12.4).

Листинг 12.4. Фрагмент файла /var/log/explanations

```
Jun 20 14:15:19 localhost rpmdrake[2573]: Extracting header of gnome2-
2.4.0-3mdk.noarch from /var/lib/urpmi/hdlist.Installation CD 3 (x86)
(cdrom3).cz
Jun 20 14:15:36 localhost rpmdrake[2573]: Extracting header of gnome-
audio-2.0.0-1mdk.noarch from /var/lib/urpmi/hdlist.Installation CD 2
(x86) (cdrom2).cz
Jun 20 14:15:41 localhost rpmdrake[2573]: Extracting header of gnome-
common-2.4.0-1mdk.noarch from /var/lib/urpmi/hdlist.Installation CD 4
(x86) (cdrom4).cz
Jun 20 14:15:57 localhost rpmdrake[2573]: Installing package remova-
ble://mnt/cdrom/Mandrake/RPMS/eel-2.4.2-1mdk.i586.rpm
Jun 20 14:15:57 localhost rpmdrake[2573]: Installing package remova-
ble://mnt/cdrom/Mandrake/RPMS/eog-2.4.1-1mdk.i586.rpm
Jun 20 14:15:57 localhost rpmdrake[2573]: Installing package remova-
ble://mnt/cdrom/Mandrake/RPMS/libapm1-3.1.0-6mdk.i586.rpm
Jun 20 14:15:57 localhost rpmdrake[2573]: Installing package remova-
ble://mnt/cdrom/Mandrake/RPMS/libcroco1-0.4.0-1mdk.i586.
```

В этом файле вы найдете полный отчет о действиях программы — видно даже, откуда был установлен тот или иной пакет. Нужно отметить, что файл explanations существует только в Linux Mandrake — в других дистрибутивах вы его не найдете.

При поиске ошибок вряд ли стоит заглядывать в файл auth.log и secure — это уже в последнюю очередь.

Когда вы определите причину сбоя (она будет записана в один из файлов конфигурации), вы сможете ее устранить.

12.3.3. Аппаратный сбой

Причиной аппаратного сбоя, как мы знаем, может стать или полный отказ устройства, или частичный отказ одного из модулей устройства, что говорит о том, что вскоре нужно будет заменить все устройство. При полном отказе устройства результат виден невооруженным взглядом. Наиболее часто отказываются работать жесткие диски и оптические приводы (поскольку в их конструкции есть механические детали), на втором месте — оперативная память, за ней — видеокарты и прочие карты расширения. Самыми надежными остаются процессор и материнская плата. Хотя это все относительно и зависит от качества устройств, которое напрямую зависит от производителя "железа". Не секрет, что вероятность отказа у "чистокровных" компьютеров от Intel и HP намного меньше, чем у компьютера, собранного в подвале неизвестной компьютерной фирмой.

Отказы жесткого диска

Если у вас отказал жесткий диск, значит, причина кроется или в плохой электронике, или в плохих пластинах (жесткий диск состоит из нескольких магнитных пластин, на которых, собственно, и хранится информация). Но вам от этого не легче — все равно нужно покупать новый, ведь неисправные практически не поддаются ремонту, особенно в кустарных условиях. Иногда можно еще восстановить информацию, но это нужно делать в лабораторных (где есть специальное оборудование) условиях. Фирм, занимающихся восстановлением информации с винчестеров, немного, а их услуги стоят довольно дорого, поэтому, чтобы не пришлось платить двойную цену (за новый жесткий диск и за восстановление информации со старого), периодически делайте резервные копии. Для этого просто записывайте важные для вас данные на CD или DVD (приводы DVD-RW сейчас очень дешевы и позволяют записывать довольно большие объемы данных на один DVD-диск). Потом эти диски лучше всего хранить в безопасном месте.

Жесткий диск может отказать не сразу, а постепенно, как правило, предшественниками полного отказа жесткого диска являются "битые" блоки. Проявляется это так: при записи невозможно записать в определенный сектор жесткого диска или же, наоборот, записать можно, но после чтения данные из "плохого" сектора не читаются. Если система не может прочитать информацию с такого сектора, вы увидите на консоли соответствующее сообщение системы. Если вы подозреваете, что причина именно в наличии "битых" секторов, проверьте ваш жесткий диск с помощью программы badblocks. Если ваши

опасения подтвердились, немедленно сделайте резервную копию всех данных, которые еще можно прочитать с диска, поскольку сейчас ваш жесткий диск непредсказуем — он может еще проработать с полгода и год, а может отказать уже завтра или даже через час. После этого купите новый жесткий диск (именно новый, а не другой б/у) и восстановите информацию с резервной копии.

Отказы памяти

При полном отказе оперативной памяти при запуске системы (именно компьютера, а не Linux, поскольку до загрузки Linux дело не дойдет) вы услышите один длинный гудок системного динамика. Это сигнал к тому, что пора менять модуль оперативной памяти, но пока не спешите этого делать. Возможно, когда вы разбирали компьютер, вы затронули модуль. Вытащите модуль, протрите пыль и установите обратно. Если это не поможет, установите модуль памяти в другое гнездо. Если опять — нет, тогда у вас появился новый брелок (вы уже догадались, что будет в роли брелока). Если вы особо не разбираетесь в "железе", то при покупке нового модуля памяти прихватите с собою старый — чтобы не ошибиться и выбрать новый модуль "по образу и подобию" старого.

В первом разделе этой главы было показано, как можно с помощью memtest86 протестировать имеющиеся модули оперативной памяти — ради интереса протестируйте свои, вдруг их пора менять, а вы об этом и не подозреваете.

Отказ видеокарты

В случае отказа видеокарты звуковой сигнал BIOS будет таким: один длинный и два коротких. Это, конечно, в случае с BIOS фирмы AWARD. Если у вас BIOS другой фирмы, то самое время прочитать о его звуковых сигналах в руководстве по материнской плате. Полный отказ видеоплаты встречается редко, как правило, в этом виноват сам пользователь (нечего было разгонять видео — вследствие этого видеокарта сильнее греется, а если она к тому же не оснащена вентилятором, через определенное время она может выйти из строя). Причем это произойдет намного быстрее, чем вы думаете.

В большинстве случаев же причиной сбоя может быть неполный контакт видеоплаты со слотом шины AGP (или PCI, если у вас до сих пор PCI-видео). Вытащите видеоплату из слота и аккуратно установите ее обратно. То же самое нужно сделать со штекером монитора. Теперь включите компьютер. Думаю, что все будет нормально. В противном случае вам придется покупать новое видео.

Отказ материнской платы и процессора

Обычно если сгорает один из этих компонентов, то следом за ним следует и второй, но бывают исключения, поэтому нужно отнести компьютер в мастерскую для диагностики. Конечно, если у вас есть подобная материнская плата и такой же процессор, можете все проверить и сами. Хотя я бы на вашем месте не рисковал — вдруг у вас неисправна материнская плата, и из-за нее вышел из строя процессор? Тогда вы рискуете "испортить" еще один процессор.

Причиной аппаратного отказа материнской платы и/или процессора может быть скачок напряжения или банальный перегрев, когда электроника не успела выключить питание компьютера.

Но в случае с материнской платой возможен и программный отказ — это выход из строя BIOS. В этом случае поможет перезапись BIOS с помощью программатора. Существуют способы обойтись без него, но они довольно рискованные, чтобы их использовать. Такая процедура стоит недорого — обратитесь в магазин, в котором вы покупали компьютер, вам обязательно помогут. Лечение такой неисправности гораздо дешевле, чем покупка новой материнской платы.

И еще: в большинстве случаев отказа материнской платы или процессора система вообще не подает никаких звуковых сигналов — это верный признак того, что причина именно в материнской плате/процессоре.

Диагностика аппаратного сбоя с помощью ядра

Если во время загрузки или работы Linux произошел серьезный аппаратный сбой и его причина не видеоподсистема, ядро "впадает в панику" (это режим работы ядра — режим паники, когда работа всей системы останавливается), а на дисплее вы увидите сообщение о вероятной причине сбоя.

Если же сбой некритичный, то есть можно продолжать работу, сообщение об ошибке также будет выведено на консоль и записано в журнал /var/log/messages. В некоторых дистрибутивах — в один из файлов в каталоге /var/log/kernel (в зависимости от типа сообщения — предупреждение, ошибка и т. д.).

Во время загрузки системы сообщения ядра так быстро выводятся на экран, что не всегда успеваешь их просмотреть. Однако это можно сделать после загрузки системы. Для этого введите команду:

```
# dmesg | less
```

А можно просто открыть в любом текстовом редакторе файл /var/log/dmesg и просмотреть его. Вот пример сообщений, выводимых при загрузке дистрибутива ASP Linux 11 (рис. 12.15).

root@localhost:~ - Shell - Konsole

Сеанс Правка Вид Закладки Настройка Справка

```
Linux version 2.6.14-1.1653asp (build@amd64.asplinux.com.ua) (gcc version 4.0.2
20051125 (Red Hat 4.0.2-8)) #1 Mon Dec 26 14:31:20 EET 2005
BIOS-provided physical RAM map:
 BIOS-e820: 0000000000000000 - 000000000009fc00 (usable)
 BIOS-e820: 000000000009fc00 - 00000000000a0000 (reserved)
 BIOS-e820: 00000000000f0000 - 0000000000100000 (reserved)
 BIOS-e820: 0000000000100000 - 000000000fff0000 (usable)
 BIOS-e820: 000000000fff0000 - 000000000fff3000 (ACPI NVS)
 BIOS-e820: 000000000fff3000 - 0000000010000000 (ACPI data)
 BIOS-e820: 00000000ffff0000 - 0000000100000000 (reserved)
0MB HIGHMEM available.
255MB LOWMEM available.
Using x86 segment limits to approximate NX protection
On node 0 totalpages: 65520
  DMA zone: 4096 pages, LIFO batch:1
  Normal zone: 61424 pages, LIFO batch:31
  HighMem zone: 0 pages, LIFO batch:1
DMI 2.2 present.
ACPI: RSDP (v000 VIA694                                    ) @ 0x000f6a20
ACPI: RSDT (v001 VIA694 AWRDACPI 0x42302e31 AWRD 0x00000000) @ 0x0fff3000
ACPI: FADT (v001 VIA694 AWRDACPI 0x42302e31 AWRD 0x00000000) @ 0x0fff3040
ACPI: DSDT (v001 VIA694 AWRDACPI 0x00001000 MSFT 0x0100000e) @ 0x00000000
ACPI: PM-Timer IO Port: 0x4008
Allocating PCI resources starting at 20000000 (gap: 10000000:efff0000)
Built 1 zonelists
Kernel command line: BOOT_IMAGE=asp11 ro quiet root=LABEL=/
Initializing CPU#0
CPU 0 irqstacks, hard=c03fd000 soft=c03fc000
PID hash table entries: 1024 (order: 10, 16384 bytes)
Detected 1595.665 MHz processor.
Using pmtmr for high-res timesource
Console: colour VGA+ 80x25
Dentry cache hash table entries: 65536 (order: 6, 262144 bytes)
Inode-cache hash table entries: 32768 (order: 5, 131072 bytes)
Memory: 254540k/262080k available (2125k kernel code, 6972k reserved, 732k data,
```

Shell

Рис. 12.15. Команда **dmesg**

Глава 13

Установка ASP Linux

13.1. Перед установкой

Последние версии дистрибутивов ASP Linux поставляются в двух комплектациях: Standard и Deluxe. Первая поставляется на четырех компакт-дисках, а вторая — на шести (или на одном DVD). На момент написания этих строк была доступна только стандартная комплектация одиннадцатой версии ASP Linux. Она поставляется на четырех компакт-дисках. Примечательно, но если при установке не изменять список пакетов, а оставить все как есть, то дистрибутив установится только с одного компакт-диска — с первого. Это очень удобно: во-первых, не нужно постоянно находиться возле компьютера и ждать, когда система запросит очередной диск, а во-вторых, если нужно установить Linux на другом компьютере, нет необходимости носить с собой все четыре диска — все, что нужно для организации рабочей станции, есть на первом диске. По умолчанию устанавливается много различных программ, в том числе GNOME, OpenOffice, программы для работы с графикой, сканером, звуком и видео.

Что же касается системных требований, то тут ASP Linux ничем не отличается от других дистрибутивов. Для комфортной работы в графической среде GNOME (или KDE — кому что больше нравится), необходимо минимум 192 Мбайт оперативной памяти, а еще лучше — 256 или больше. Такими требованиями сейчас никого не напугаешь. Даже если у вас слабенький компьютер (128 Мбайт ОЗУ), вы все равно можете установить ASP Linux и работать. Конечно, операции, требующие больших объемов оперативной памяти, например, работа с графикой или видео, будут выполняться медленно. Что же касается жесткого диска, то типичная установка (по умолчанию) требует 1,8 Гбайт дискового пространства. Плюс место для ваших файлов (это уже на ваше усмотрение) и на развитие системы (все равно захочется еще что-то установить). Поэтому лучше всего выделить для Linux не менее 3 Гбайт. Но это еще

не все. Вам не нужно забывать о пространстве для своп-раздела. Если у вас достаточно оперативной памяти, скажем 512 Мбайт и более — установите размер своп-памяти 128 Мбайт — этого будет вполне достаточно. Если же у вас 256 Мбайт ОЗУ или даже меньше — установите 256 Мбайт для своп-раздела.

Перед установкой Linux нужно выполнить дефрагментацию всех Windows-разделов, если такие есть. Дефрагментация необходима для корректного изменения размера Windows-раздела. Также не будет лишней резервная копия всех важных данных.

13.2. Начало установки

Первый диск ASP Linux является загрузочным — этим мало кого сейчас удивишь, ведь в основном первые диски всех дистрибутивов являются загрузочными. А было время, когда приходилось создавать загрузочную дискету и загружаться с нее.

Вставьте первый диск (если у вас DVD, то он будет и последним) в ваш привод CD-ROM/DVD-ROM, перезагрузитесь, в опциях SETUP выберите загрузку с CD-ROM, сохраните изменения и еще раз перезагрузитесь.

Первый этап установки — это выбор языка ASP Linux. Думаю, особо не нужно объяснять, какой язык нужно выбрать. В отличие от других дистрибутивов (того же Mandrake или Fedora Core), выбор языков довольно скудноват, но главное, что есть русский (рис. 13.1).

После этого нужно выбрать тип установки (рис. 13.2):

- **Быстрая установка** — подойдет для нового компьютера, на котором еще не созданы разделы. Если вы попытаетесь использовать этот тип установки для компьютера, на котором уже установлены другие операционные системы, например, Windows, вы рискуете навсегда с ними распрощаться;
- **Выборочная установка** — оптимальный вариант для большинства пользователей. Вы можете выбирать не только пакеты, но и вручную разбивать жесткий диск;
- **Обновление существующей системы Linux** — если у вас уже была установлена Linux, совместимая с Red Hat (например, Fedora Core, Mandrake), вы можете попытаться обновить ее до ASP Linux 11.

Последний подготовительный этап: вам нужно уточнить, откуда устанавливать Linux (рис. 13.3). Вы можете ее установить с компакт-диска, с ISO-образа на жестком диске, по NFS или же с сетевого ресурса, например, с FTP-сервера. Пусть этот вопрос программы установки не сбивает вас с толку — выбираем первый вариант и продолжаем установку.

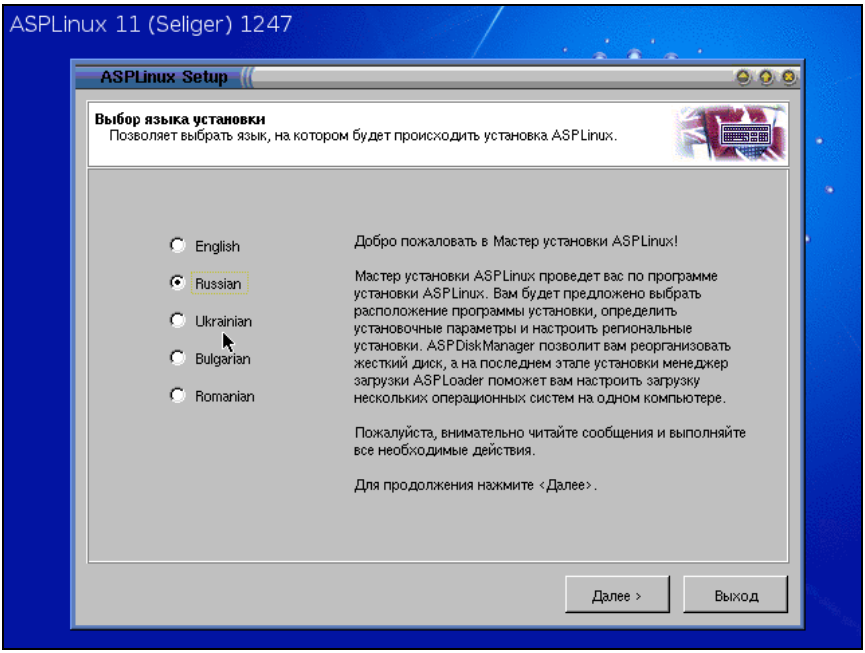

Рис. 13.1. Выбор языка Linux

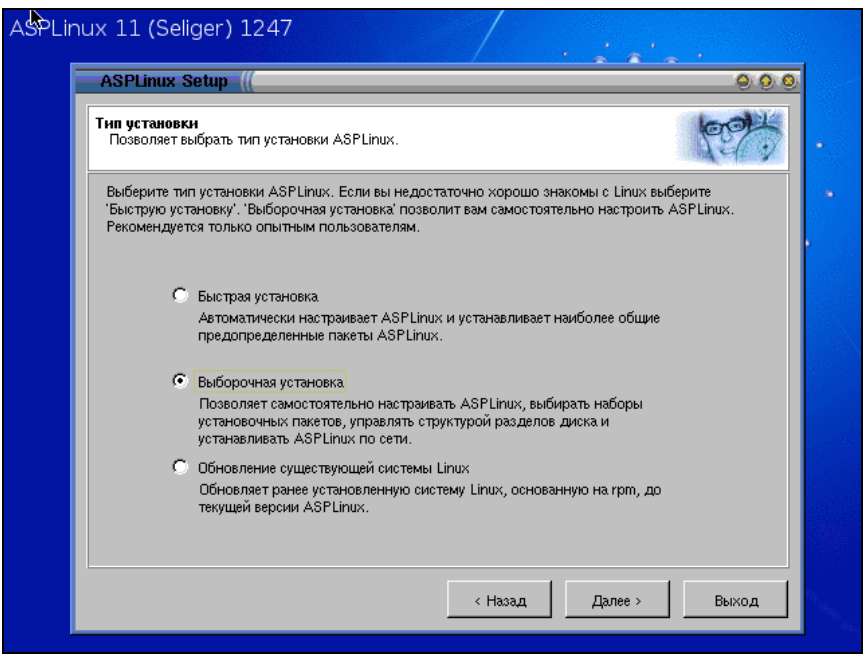

Рис. 13.2. Выбор типа установки

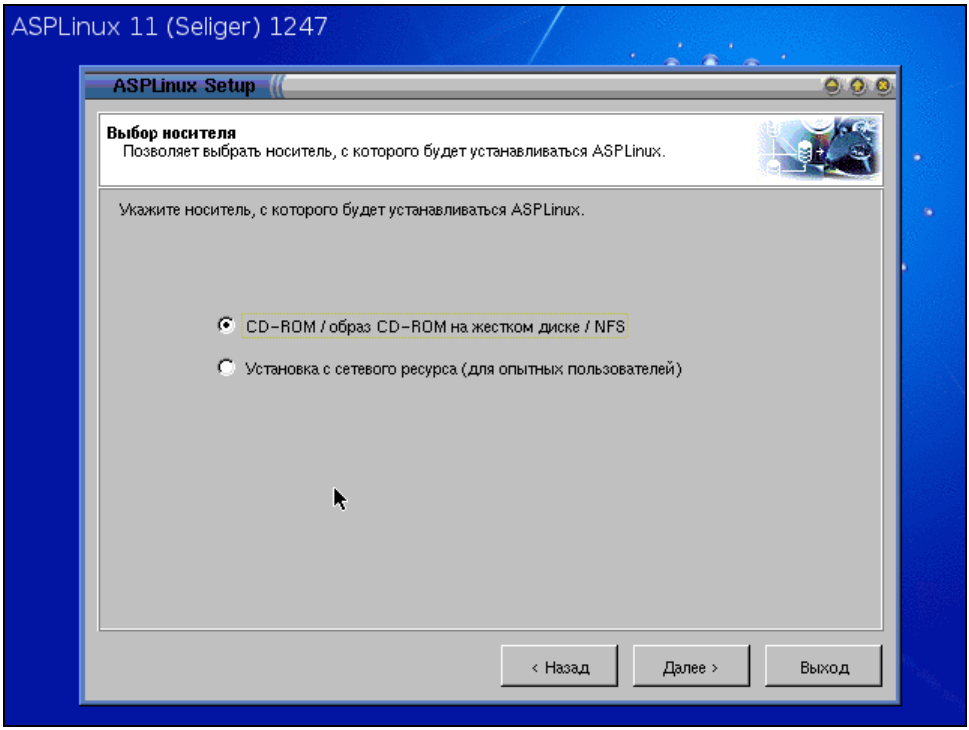

Рис. 13.3. Выбор носителя

13.3. Разбиение жесткого диска

Перед установкой Linux нужно было выполнить дефрагментацию всех Windows-разделов. Если вы этого не сделали — перезагрузитесь и произведите дефрагментацию, иначе при изменении размера Windows-раздела вы можете потерять данные.

Для чего нужно изменение раздела? Linux использует свою файловую систему, для создания файловой системы Linux необходимо создать собственные Linux-разделы (как минимум, два: один для системы и данных, а второй — для свопа). Ясно, что разделы Linux не возьмутся ниоткуда — их нужно создать, следовательно, для них нужно место. Поскольку Windows узурпировала все дисковое пространство (скорее всего, у вас несколько Windows-разделов), вам нужно или удалить Windows-разделы (все или один из них) и на свободном месте создать Linux-раздел, или же изменить размер уже существующего Windows-раздела. В первом случае (удаление раздела) мы потеряем данные, а во втором — нет.

Лучше всего изменять размер диска C, если это возможно. Поскольку диск C обычно является первым разделом (/dev/hda1), то при изменении размера и установке на освободившееся место Linux у вас будет меньше недоразумений, связанных с 1024 цилиндром. Дело в том, что сами Linux-разделы могут начинаться с любого цилиндра, а вот каталог /boot (который обычно является частью корневой файловой системы) должен находиться до 1024 цилиндра. Если это не так, то вы должны создать два Linux-раздела — один небольшой (50–70 Мбайт максимум) для /boot, а остальное — для / (корень). Про своп постарайтесь тоже не забыть!

Иногда размер диска C изменить невозможно — просто на нем нет достаточно места. Например, на диске C у вас может остаться свободным 500–700 Мбайт, а на диске F — места хоть отбавляй. В этом случае можно сделать так: создать небольшой Linux-раздел размером до 100 Мбайт — для каталога /boot. После этого от диска F "отрезать" необходимое для установки и работы Linux место в 3–4 Гбайт.

Примечание

Проблема 1024 цилиндра до сих пор актуальна. Если вы хотите раз и навсегда от нее избавиться, установите загрузчик ASPLoader (об этом мы поговорим позже). Существуют способы заставить загрузчики GRUB и LILO загружать Linux с раздела, находящегося за пределами 1024 цилиндра, но мы их рассматривать не будем, поскольку они работают не всегда и не у всех. Например, в некоторых случаях включение опции LBA в BIOS помогает загрузчику GRUB избавиться от проблемы 1024 цилиндра, а в некоторых — нет.

Теперь, когда стратегия распределения дискового пространства ясна, можем перейти к практике. После выбора носителя программа установки предложит вам выбрать один из трех методов назначения дискового пространства (рис. 13.4):

- **Использовать весь диск** — существующая структура диска будет уничтожена, а новая будет создана с нуля. Можно использовать только для новых компьютеров, иначе вы потеряете все свои данные;
- **Использовать свободное место** — программа установки попытается найти свободное место. Если свободное место будет найдено и его будет достаточно, программа установки создаст Linux-разделы и установит операционную систему;
- **Дополнительно** — вы сами можете редактировать структуру диска.

Если вы предпочитаете самостоятельно контролировать ситуацию, а не полагаться на логику программы, выберите последний вариант — вы сами сможете указать, сколько место вам нужно, насколько вы хотите "потеснить" Windows и т. д.

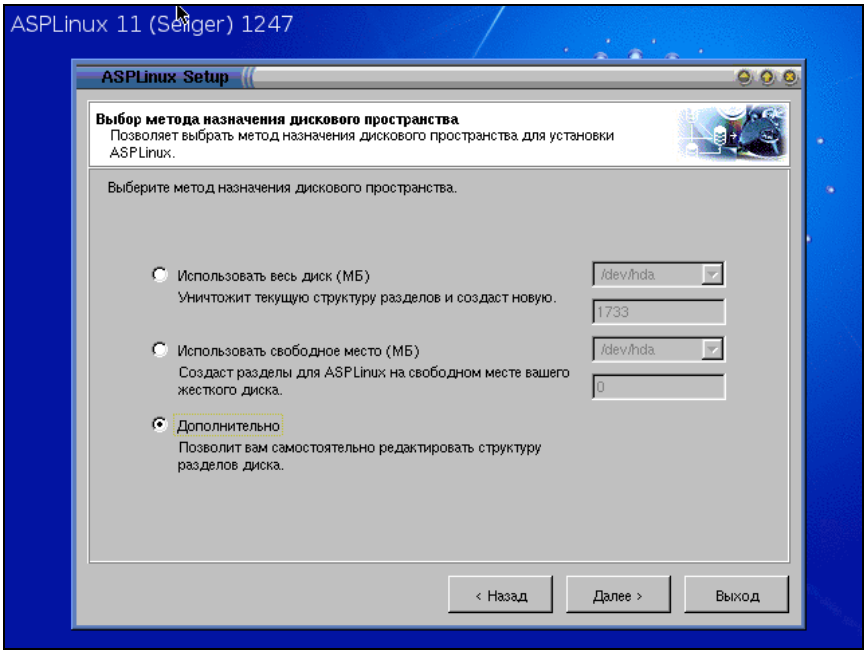

Рис. 13.4. Методы назначения дискового пространства

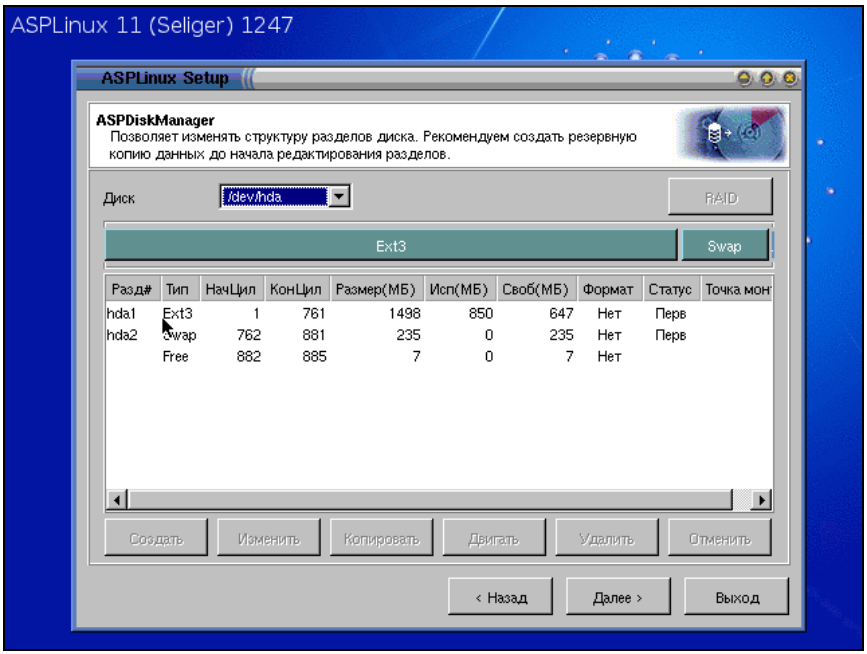

Рис. 13.5. Изменение структуры диска

Если вы — начинающий пользователь, можете выбрать второй вариант, но вы сильно рискуете. Выбирайте третий вариант — там ничего сложного нет.

Итак, сейчас мы будем вручную редактировать структуру диска. Средство для редактирования структуры диска от ASP Linux (рис. 13.5) мало чем отличается от аналогичных средств в других дистрибутивах.

Для изменения размера какого-то раздела нужно его выбрать, а потом нажать кнопку **Изменить**. В появившемся окне нужно будет установить новый размер раздела (рис. 13.6). После этого у вас появится свободное место — его нужно выделить, щелкнув мышью, и нажать кнопку **Создать**. А вот с созданием разделов у ASP Linux немного туговато: если в других дистрибутивах вы можете создать любой раздел, в том числе FAT32, NTFS, то в ASP Linux — сугубо Linux-разделы.

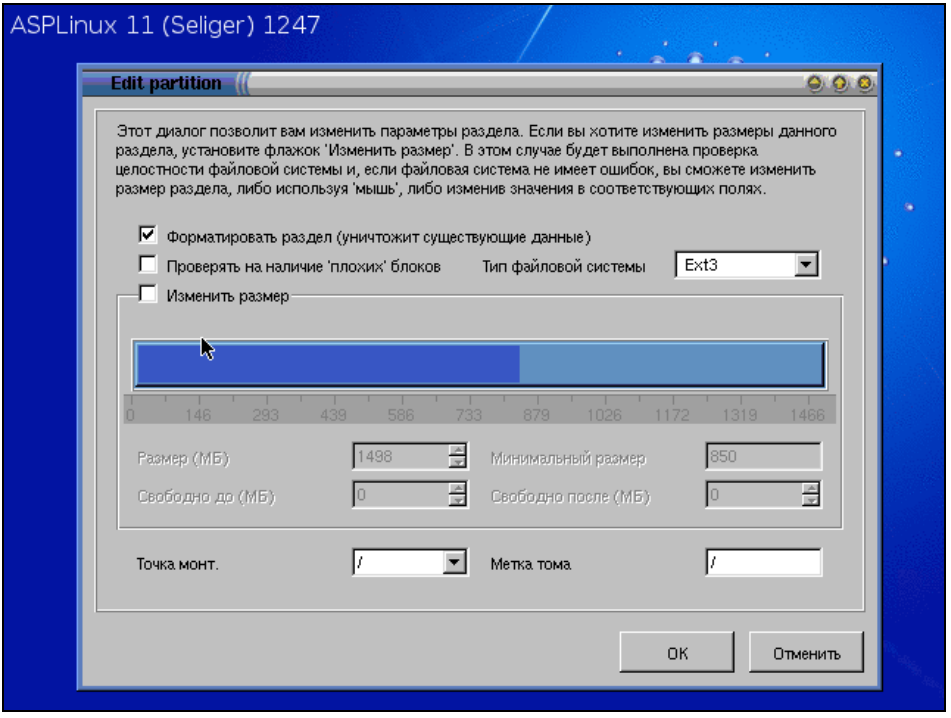

Рис. 13.6. Редактирование раздела

Не забудьте, что вам нужно минимум два раздела — один Linux-раздел, а второй — раздел для свопа.

Если Linux-разделы у вас уже были созданы, то выберите раздел, который вы хотите сделать основным разделом Linux, и нажмите кнопку **Изменить**.

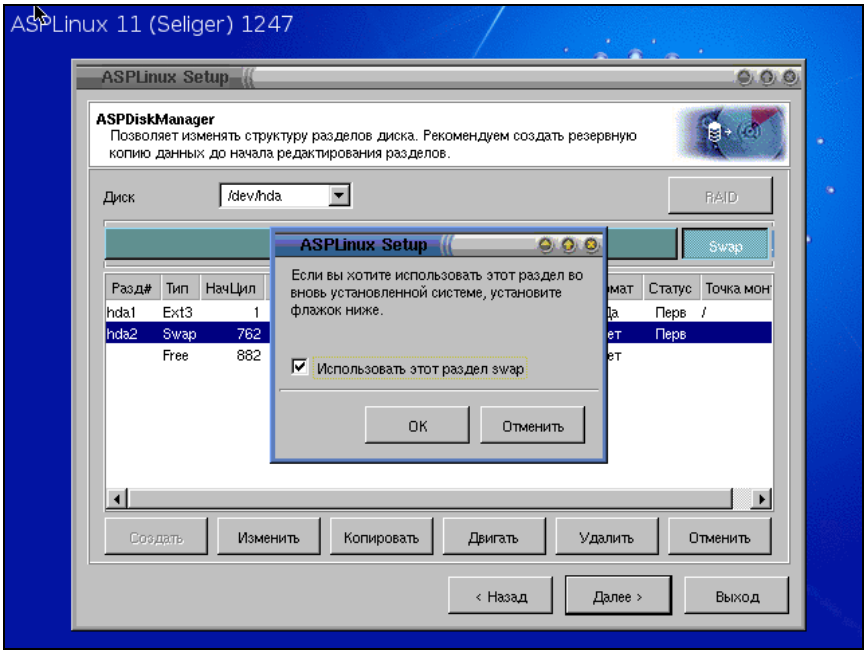

Рис. 13.7. Подключение уже созданного своп-раздела

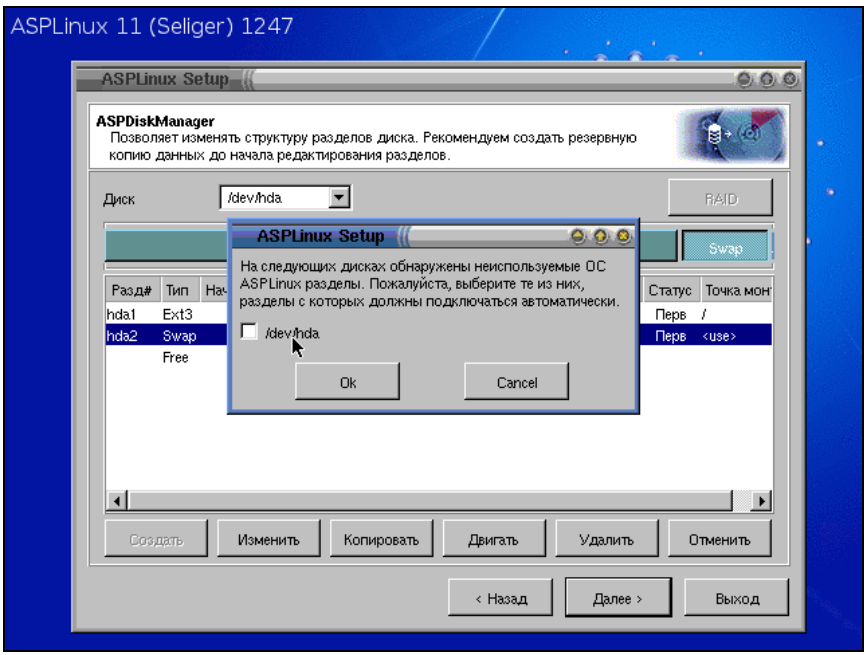

Рис. 13.8. Подмонтировать другие разделы?

В появившемся окне установите точку монтирования и метку корневого раздела. Аналогично выделите своп-раздел и нажмите кнопку **Изменить**. Появится небольшое окошко, в котором нужно установить флажок **Использовать этот раздел swap** (рис. 13.7).

По окончанию редактирования структуры диска нажмите кнопку **Далее**. Программа установки предложит вам подмонтировать к корневой файловой системе найденные разделы (это будут разделы вашего Windows). Не отказывайтесь от этой возможности, если не хотите прописывать их вручную в файле /etc/fstab (рис. 13.8).

13.4. Установка пакетов

После редактирования структуры диска, вам нужно выбрать, какие пакеты вы хотите установить. Чтобы вы не выбирали пакеты вручную, разработчики ASP Linux предусмотрели следующие варианты установки пакетов (рис. 13.9):

- **Типовая установка** — все, что необходимо для рабочей станции или домашнего компьютера. Для этого типа установки нужен только первый диск — очень удобно;
- **Сервер** — выберите этот вариант, если хотите настроить сервер на базе ASP Linux;
- **Разработка** — будут установлены средства разработки программ, а также исходные коды ядра;
- **Минимальная установка** — позволяет сэкономить место на диске, но кому нужен такой Linux? Разве что перед друзьями гордо ударить себя в грудь;
- **Пользовательский** — позволяет вам выбрать пакеты самостоятельно, но по умолчанию не будет отмечено ни одного пакета — вы все будете выбирать сами.

Если вы хотите выбрать пакеты самостоятельно, не устанавливайте пользовательский тип — создание списка пакетов с чистого листа довольно утомительное занятие. Лучше всего выберите один из вариантов установки (**Типовая**, **Сервер**, **Разработка** или **Минимальная установка**) и установите флажок **Выборочно**. После этого нажмите **Далее** — вы увидите список пакетов, в котором уже будут отмечены пакеты для выбранного типа установки (рис. 13.10).

После этого появится окно со сводной информацией — вам нужно только нажать кнопку **Установить**. Во время установки программа будет выводить имена устанавливаемых пакетов (рис. 13.11).

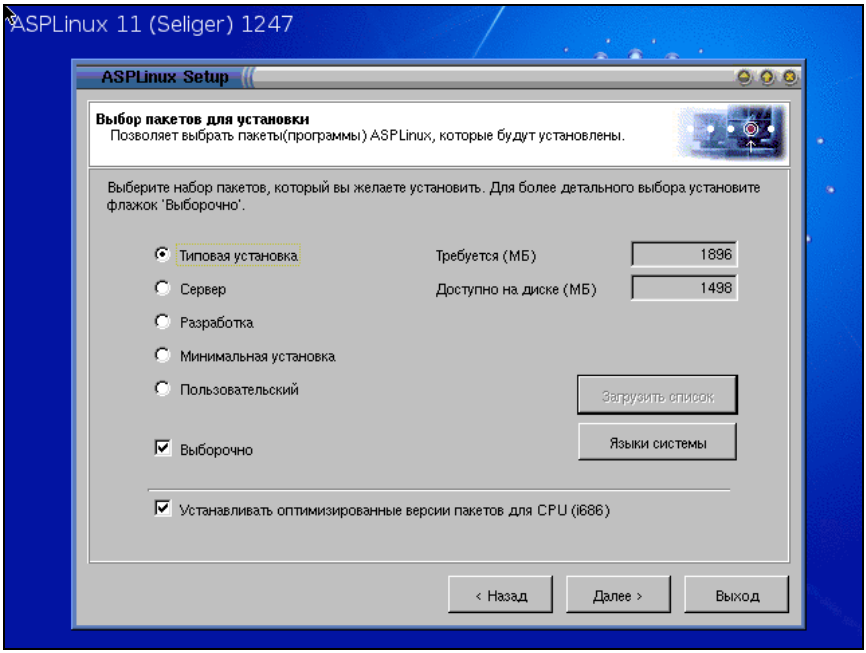

Рис. 13.9. Как устанавливать пакеты

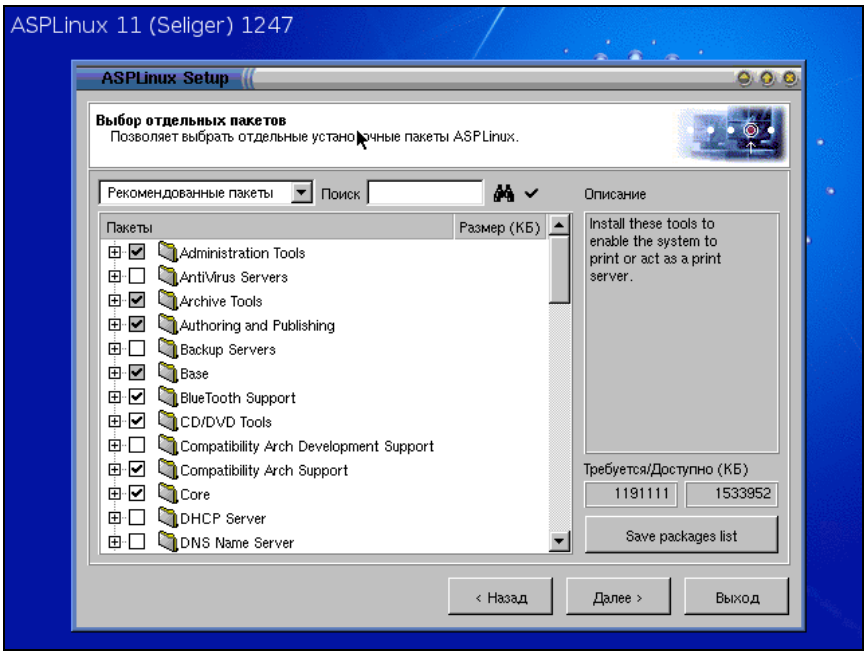

Рис. 13.10. Выбор пакетов вручную

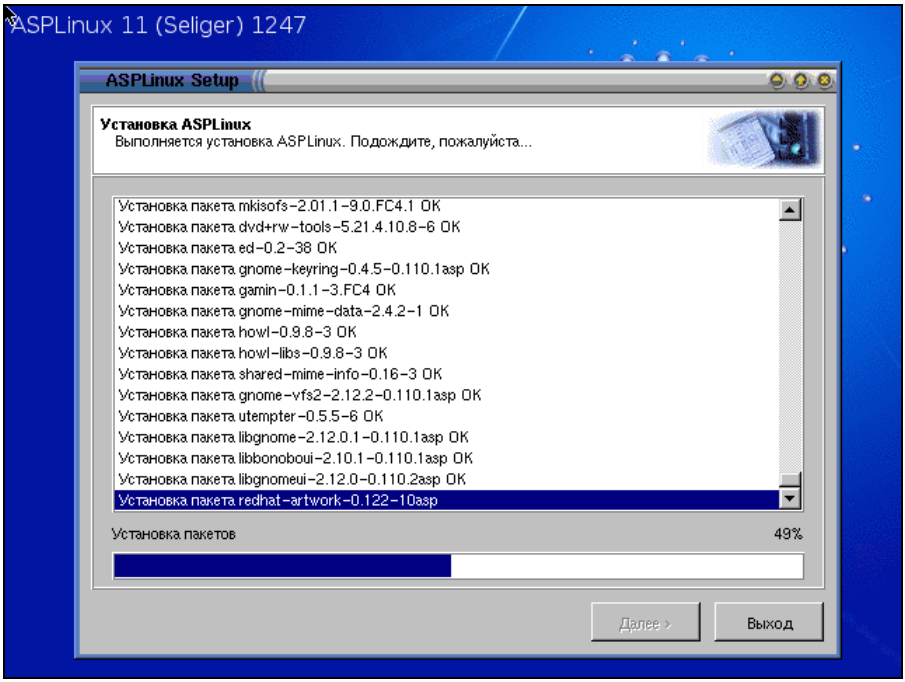

Рис. 13.11. Установка пакетов

13.5. Установка загрузчика

Все пакеты уже установлены. Осталось еще чуть-чуть, и система будет установлена. Установка загрузчика — это очень важный этап, поскольку именно загрузчик позволяет загрузить Linux — без него никак (рис. 13.12).

Программа установки ASP Linux позволяет выбрать один из трех загрузчиков:

- ASPLoader — собственный загрузчик ASP, довольно хороший загрузчик;
- LILO — устаревший загрузчик, сейчас редко используется, но, тем не менее, достаточно удобный. Если вы привыкли работать с LILO, почему бы его не выбрать;
- GRUB — современный стандарт де-факто.

Какой выбрать? Иногда нужно редактировать конфигурационные файлы загрузчика — этого требуют некоторые программы, работающие на очень низком уровне. Как правило, в руководствах по этим программам описано редактирование конфигурационных файлов загрузчиков LILO и GRUB. А вот с ASPLoader вам придется разбираться самостоятельно. Если вы чувствуете в этом деле себя не очень уверенно, лучше выберите GRUB или LILO. Хотя,

с другой стороны, если вы не экспериментатор, а Linux вам нужен как рабочая лошадка — для выполнения повседневных задач, можно выбрать ASPLoader — к его конфигурационным файлам вы и не прикоснетесь. Есть еще одна причина, по которой нужно выбрать ASPLoader: он не имеет ограничения 1024 цилиндра, то есть загрузочный раздел (/boot) может быть расположен в любом месте диска.

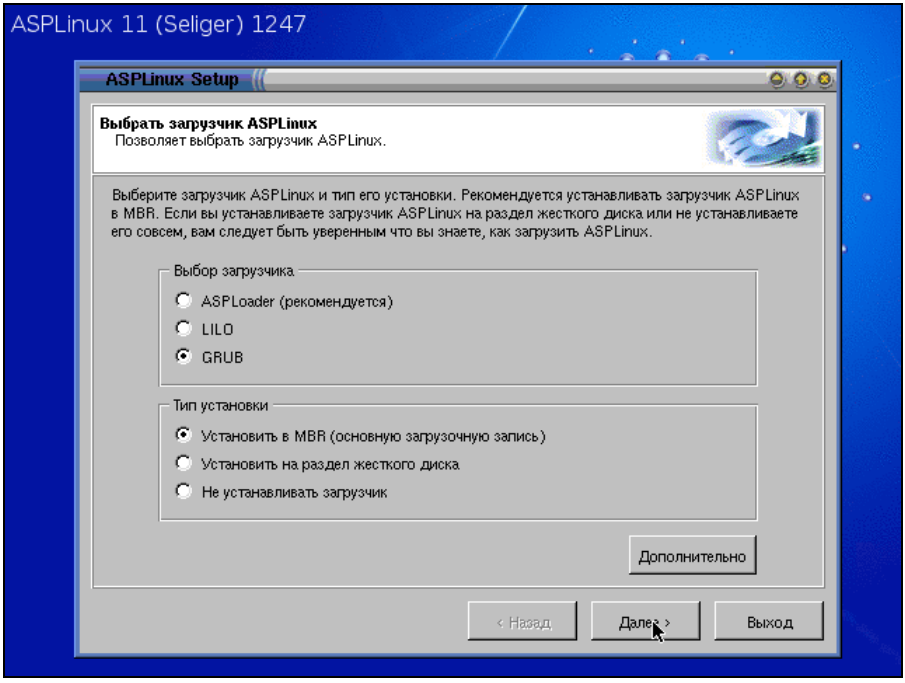

Рис. 13.12. Установка загрузчика Linux

Надеюсь, что загрузчик вы уже выбрали, осталось решить, куда его устанавливать. ASP Linux предлагает три варианта:

- в MBR;
- на раздел жесткого диска;
- не устанавливать.

Первый вариант подойдет для большинства пользователей. Второй подойдет, если вы хотите для загрузки Linux использовать не-Linux загрузчик, например, NTLoader, использующийся для загрузки Windows NT/2000/XP/2003. Второй вариант требует дополнительной настройки загрузчика NTLoader, поэтому мы его рассматривать не будем.

Третий вариант можно выбрать, если у вас уже установлен загрузчик Linux, например, когда у вас уже установлен какой-то дистрибутив Linux, а устанавливаемый ASP Linux — это второй (или третий, четвертый — неважно) дистрибутив Linux на вашем компьютере. В этом случае после перезагрузки системы вам нужно загрузить ваш основной дистрибутив и в его загрузчике "прописать" загрузку ASP Linux.

13.6. Настройка сети

После настройки загрузчика вам будет предложено настроить сеть. Сначала нужно выбрать сетевую плату (рис. 13.13), а потом — установить параметры сетевого интерфейса (рис. 13.14).

Во время установки Linux вы уже сможете настроить сети — установить нужный IP-адрес, маску сети, имя узла, шлюз, а также IP-адреса серверов DNS. Конечно, можно все это возложить на плечи DHCP-сервера, если такой работает в вашей сети. Если вы не хотите сейчас настраивать сети, выключите флажок **Активировать при загрузке** и нажмите кнопку **Далее**.

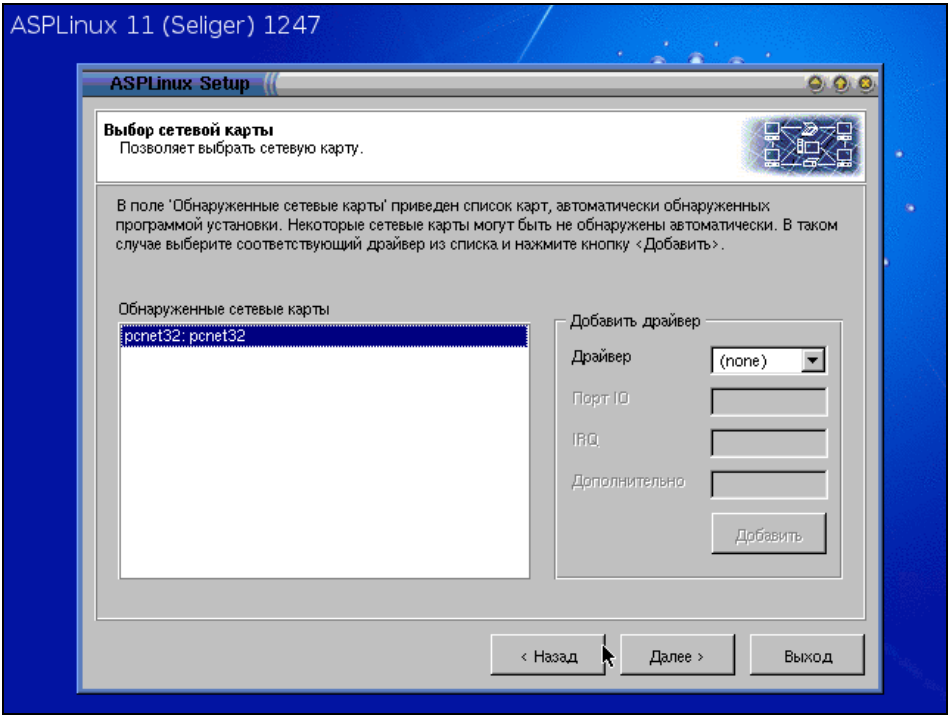

Рис. 13.13. Выбор сетевой платы

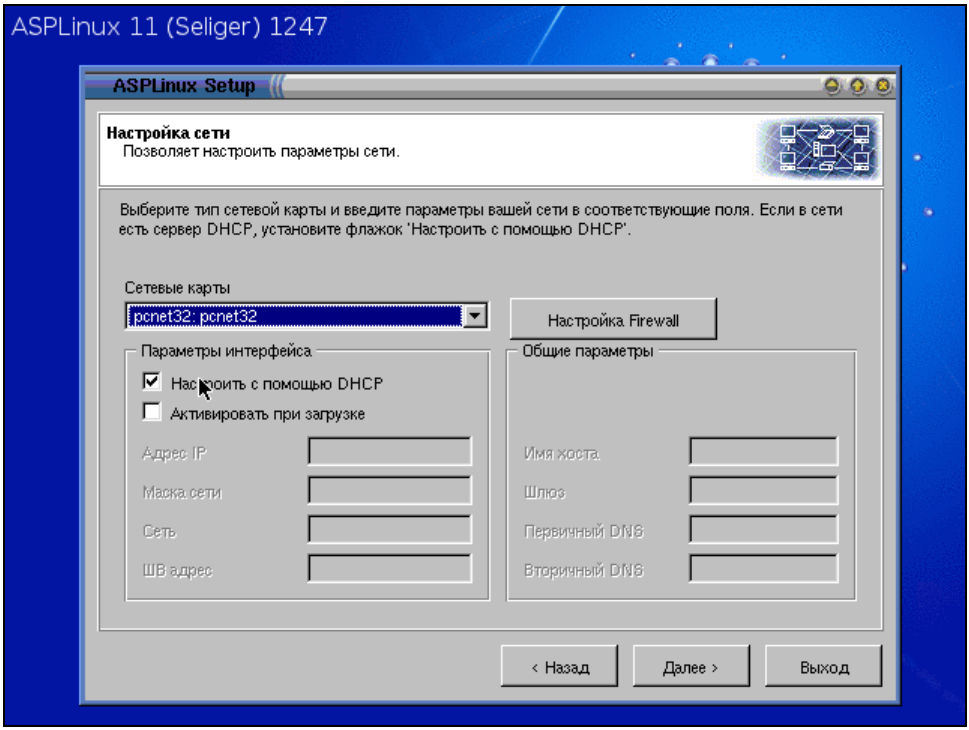

Рис. 13.14. Установка параметров сетевого интерфейса

13.7. Настройка X Window

Конфигурация системы X Window (или графической системы Unix) состоит из двух основных этапов: настройка монитора и настройка видеоплаты. В большинстве случаев Linux правильно определяет используемые монитор и видеоплату, поэтому вам не о чем беспокоиться. Но, как обычно, есть свои "но". Если программа установки неправильно определила ваш монитор, вам нужно или выбрать его из списка, или ввести его параметры (горизонтальную и вертикальную развертку) вручную (рис. 13.15). Данные параметры можно узнать в руководстве по монитору.

С видеоплатами тоже есть один неприятный момент, касающийся видеоплат от nVidia GeForce FX. По умолчанию используется драйвер nVidia GeForce generic, который вызывает зависания при работе в X Window. Для нормальной работы лучше выбрать драйвер nVidia GeForce (generic) (рис. 13.16).

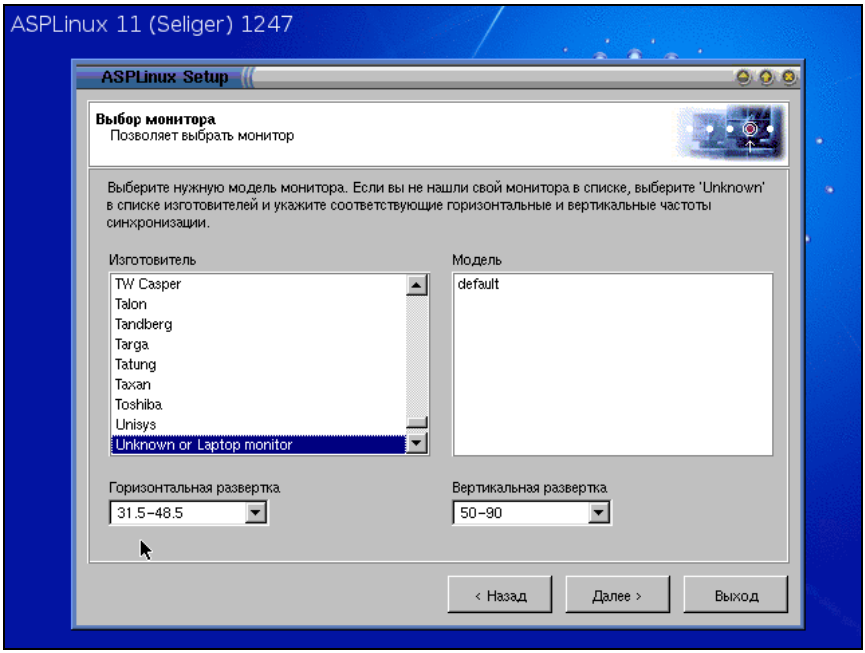

Рис. 13.15. Выбор монитора

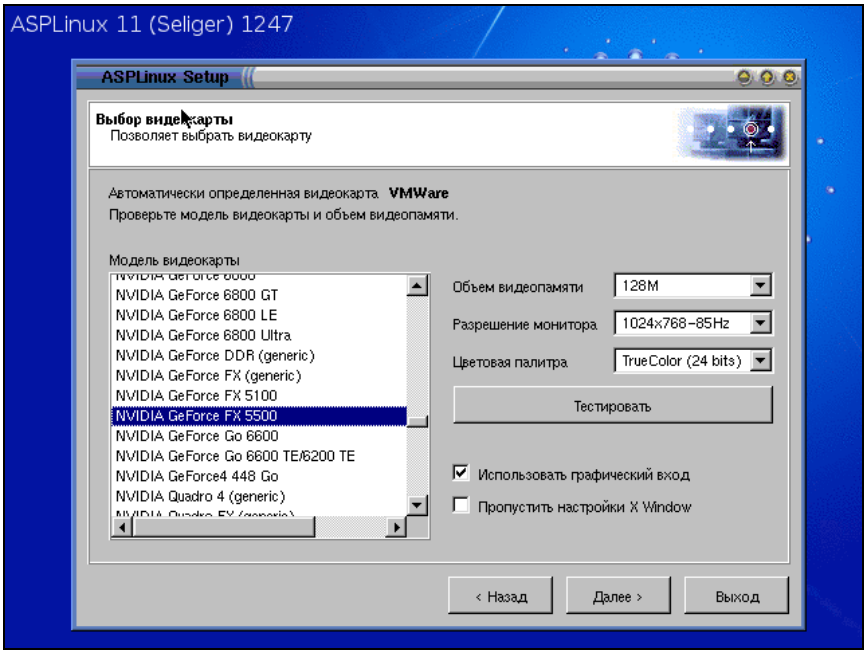

Рис. 13.16. Выбор драйвера видеоплаты

Установив параметры видеоплаты, вы можете нажать кнопку **Тестировать**, однако лучше этого не делать, поскольку при тестировании компьютер может зависнуть, и вам придется повторить установку системы с самого начала — ведь вы пока не установили пароль root.

13.8. Параметры клавиатуры и часового пояса

Сразу после настройки X Window вам нужно будет выбрать модель клавиатуры, кодировку ввода (рекомендую KOI8-R, а не UTF), раскладку клавиатуры и комбинацию клавиш для переключения языков ввода (рис. 13.17).

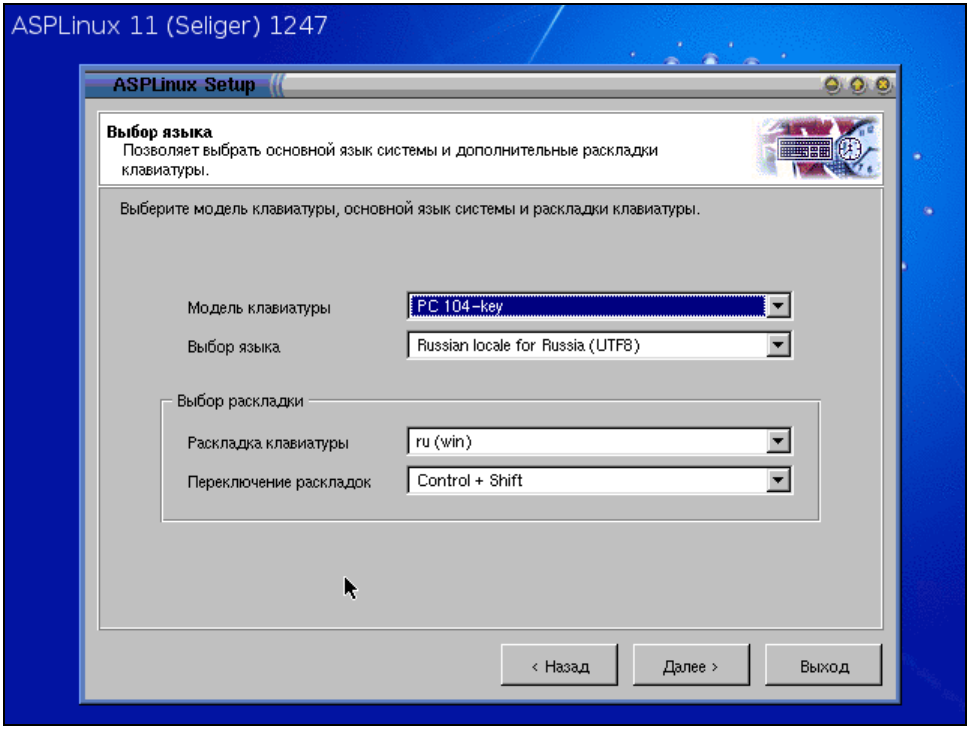

Рис. 13.17. Параметры клавиатуры

Параметры часового пояса, думаю, в особых комментариях не нуждаются (рис. 13.18).

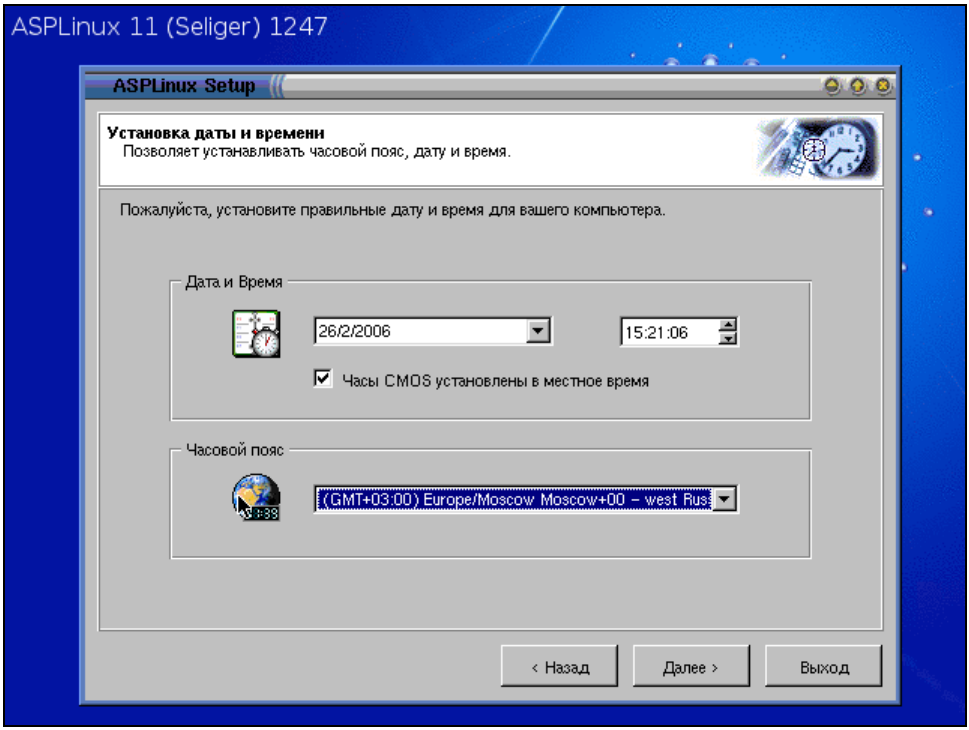

Рис. 13.18. Параметры часового пояса

13.9. Установка пароля root

Это очень важный этап. Пароль пользователя root должен быть одновременно легким в запоминании и трудным в подборе. В качестве пароля root нельзя использовать имена и инициалы близких (это же относится и к собственному имени), комбинации вроде qwerty или 123456, а также другие легко подбираемые слова. Лучше всего комбинировать большие буквы с маленькими (можно использовать только английские буквы) и цифрами. Оптимальная длина пароля — от шести до десяти символов.

Также на этом этапе нужно создать обычного пользователя, поскольку постоянно работать от имени root не рекомендуется из соображений безопасности (рис. 13.19).

Я вас поздравляю — установка завершена. Осталось перезагрузить систему. Перед этим извлеките компакт-диск из привода.

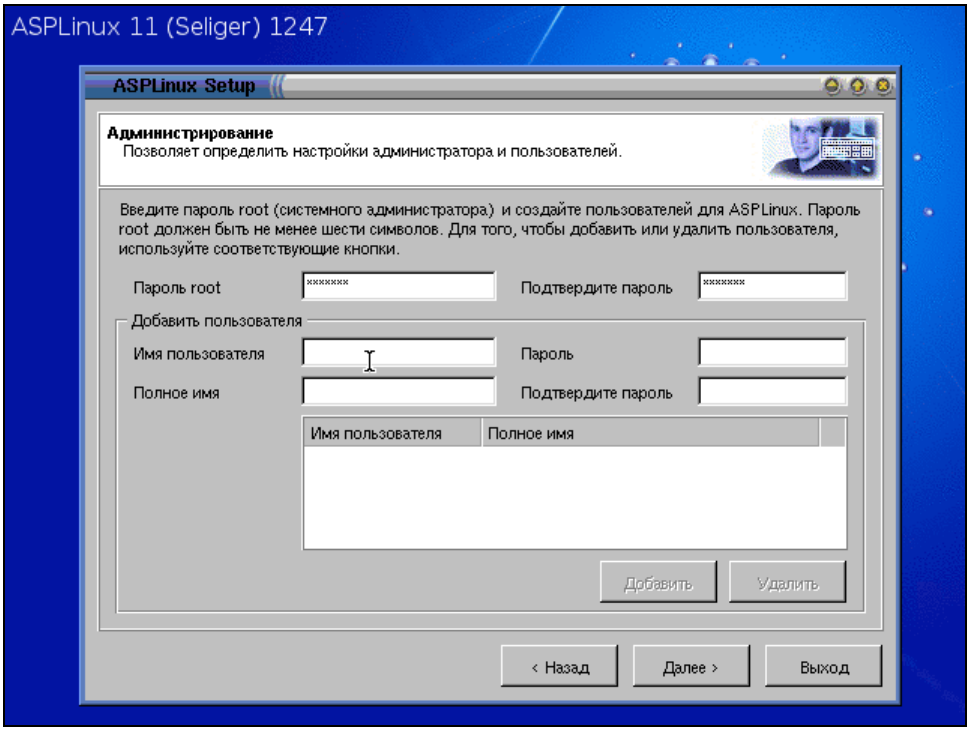

Рис. 13.19. Установка пароля root

13.10. Перезагрузка

Загрузчик Linux позволит вам выбрать одну из установленных на компьютере операционных систем и загрузить ее (рис. 13.20).

Для загрузки операционной системы сначала ее нужно выбрать с помощью стрелок из списка и нажать <Enter>. Если вы хотите изменить параметры ядра, выберите нужную вам операционную систему и нажмите клавишу <a>. Если вам нужно изменить команды перед загрузкой — нажмите <e>, а если хотите увидеть командную строку (как в старом добром LILO), нажмите <c>.

По умолчанию подробности загрузки ASP Linux скрыты (рис. 13.21), если вы хотите увидеть подробные сведения о загрузке ASP Linux, нажмите кнопку **Показать сведения**.

Если во время загрузки произойдет ошибка, вы увидите соответствующее сообщение — автоматически будет включен режим отображения подробных сведений.

Все, что вам осталось — это зарегистрироваться в системе (рис. 13.22).

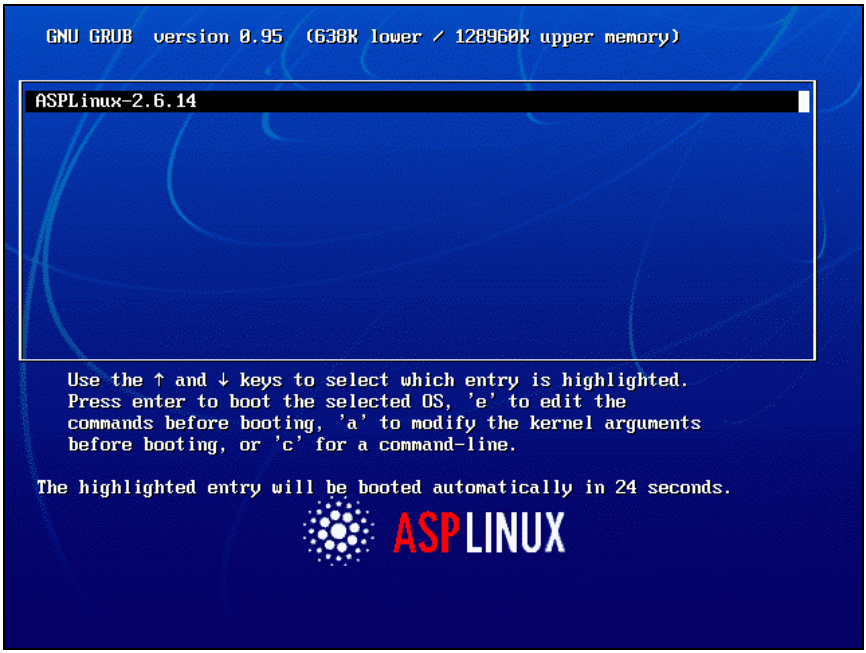

Рис. 13.20. Загрузчик ASPLoader

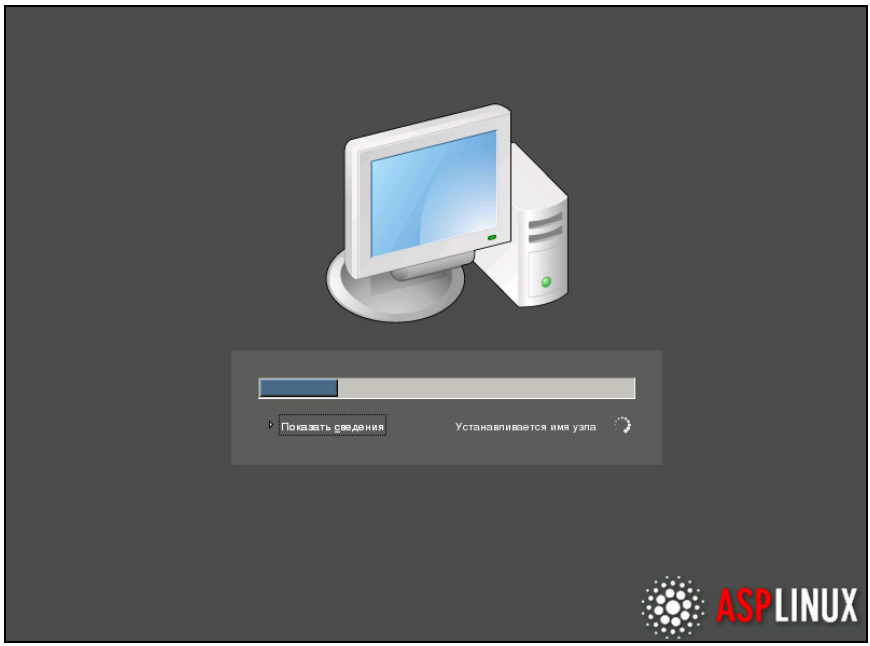

Рис. 13.21. Загрузка ASP Linux

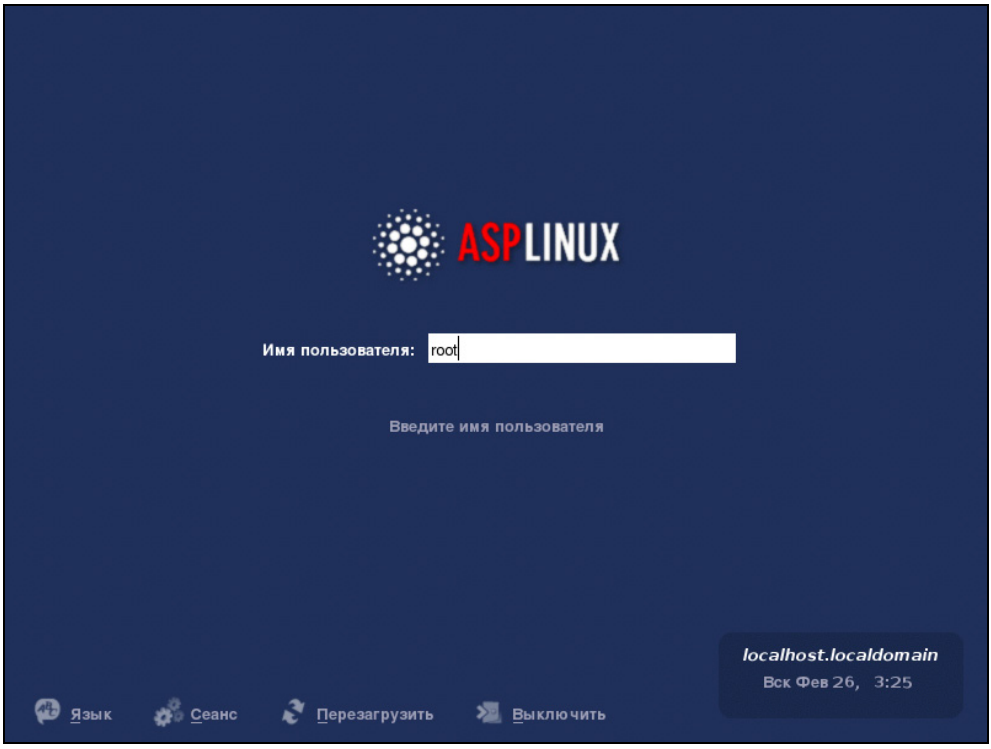

Рис. 13.22. Регистрация в системе

Заключение

Прочитав эту книгу, вы узнали, как настраивается то или иное устройство в Linux. Вполне вероятно, что у вас что-то не получилось — на данный момент Linux поддерживает много различных устройств, но далеко не все. Должен еще раз отметить, что это не проблема операционной системы, а проблема производителей устройств, которые поленились создать драйвер для операционной системы Linux.

Если у вас не получилось настроить то или иное устройство, не отчаивайтесь. Просто подождите до выхода новой версии дистрибутива или попробуйте установить другой дистрибутив. Вполне вероятно, что в следующей версии (при условии, что вы сейчас используете последнюю), появится поддержка вашего устройства. Конечно, не нужно надеяться, что появится поддержка так называемых Windows-устройств. Если же устройство вам необходимо заставить работать прямо сейчас, попробуйте поискать информацию о его настройке в Интернете — скорее всего, вашу проблему уже кто-то решил, ведь вы не единственный обладатель такого устройства. В приложении вы найдете список наиболее информативных Интернет-ресурсов, из которых вы сможете извлечь нужную для себя информацию.

Приложение

Вот список некоторых интересных интернет-ресурсов, посвященных Linux:

- **http://www.linux.ru.net/** — огромный Linux-портал, где вы сможете найти статьи, документацию по Linux, форум, а также каталог ссылок на другие Linux-ресурсы;
- **http://www.linuxcenter.ru/** — на этом сайте, кроме различной информации о Linux, вы еще сможете заказать любой дистрибутив Linux;
- **http://www.linuxforum.ru/** — форум о Linux, рассмотрение различных вопросов, связанных с Linux;
- **http://www.asplinux.ru/** — сайт дистрибутива ASP Linux;
- **http://www.openoffice.ru/** — сайт, посвященный офисному пакету OpenOffice;
- **http://gimp.ru/** — сайт, посвященный профессиональному использованию программы The GIMP — Linux-аналога Photoshop;
- **http://linuxtech.pp.ru/** — содержит очень неплохую коллекцию ссылок по Linux;
- **http://linux.spab.ru/** — подборка ссылок по Linux;
- **http://www.asmodeus.com.ua/** — сайт о создании и администрировании Web-сервера, а также о Web-программировании;
- **http://www.linuxrsp.ru/** — проект Linux Russian Security Project;
- **http://www.dkws.org.ua/** — на этом сайте вы найдете много различной информации по Linux — статей, howto, а также сможете посетить форум;
- **http://www.opennet.ru/** — проект OpenNet, все об открытых системах;

- **http://linux-ve.net** — виртуальная энциклопедия "Linux по-русски";
- **http://bioforce.net.ru/** — сайт о Linux;
- **http://unixlib.org.ru/** — библиотека информации о Unix;
- **http://linuxmusic.ru/news.php** — сайт о музыке в Linux;
- **http://linuxmobile.lrn.ru/** — сайт о подключении мобильных телефонов к Linux;
- **http://linuxnews.ru/** — новости о Linux;
- **http://www.root-ua.info/** — интернет-журнал "Root-UA.info".

Предметный указатель

/

/boot/grub/grub.conf 237
/dev/modem 174
/dev/sda 229
/dev/ttyS0 173
/etc/aspldr.conf 19
/etc/fstab 57, 227
/etc/hosts 208
/etc/hosts.conf 209
/etc/lilo.conf 18, 237
/etc/modules.conf 11
/etc/resolv.conf 213
/etc/x11/XF86Config 36
/etc/x11/XF86Config-4 36
/etc/x11/xorg.conf 36
/proc/sys/vm/swappiness 244

1

100Base-FX 199
100Base-T4 199
100Base-TX 199
10Base-2 194
10Base-5 194
10Base-T 194

A

Acpid 240
AFC 43
AGP 29, 30, 31
Anacron 240
Apache 250
Apmd 240
ASPLoader 261
ATDP 186
AT-команды 170
AUX 98
AWARD 253

B

Badblocks 235
BIOS 28, 29, 253
Bluetooth 240
Bonnie 239
Broadcast 206
Bubble 139

C

CCD-сенсоры 224
CDDB 111
CD-ROM 71, 101

Chroot 248
CMOS-сенсоры 224
Cpuburn 239
Cpuspeed 240
Crond 240
CRT-мониторы 25
CSMA/CD 197
Cups 242

D

DAO 85
Demand Priority 198
Depmod 39
Dirty-бит 60
DMA 62, 63
Dmesg 249
Domain Name System 207
DrakConf 182, 216
Drakconnect 211
Drakprinter 151
Drakxservices 240
DVD 76
DVI-I 32
DAC 30

E

Elevator 245

F

Fam 226
Fast Ethernet 192
FAT32 263
Fd 59
Fdisk 51
FIFO 85
Fsck 60, 232
FTP 204

G

Gateway 206
GDI-принтеры 145
Gentoo 247
Gigabit Ethernet 192
Gimp 159
Gorc 166
Grub-install 247

H

Haldaemon 240
Harddrake 239
Hayes AT 170
Hddtem 239
Hdparm 61, 231
HTTP 204
Hub 194
Hwbrowser 159, 228

I

ICMP 204
IDE 45, 46, 50, 51, 61
IEEE-1394 99
Ifconfig 210
Init 60
Insmod 9
Install 13
IP-адрес 205
Irqbalance 242
Isdn 242

K

KOI8-R 272
Kooka 160
KsCD 110
Kudzu 239, 242

L

LAN 192
LCD 26
Less 254
Lilo 247
LiveCD 246
Lizard 33
Lm_sensors 242
Local Volume Manager 64
LPT 145
Lsmod 8

M

MacOS 145
MAC-адрес 196
Make 15
MAN 192
MBR 268
Mdmonitor 242
Memtest86 237
Menuconfig 15
Messagebus 242
MIDI порт 99
Mkdir 58
Mkfs 56
Mknod 14
Mkswap 244
Modeline 36
Modprobe 10, 39
Modules.dep 9
Monitor 36
Mount 57
Mouseconf 131
MS DOS 145

N

NetBIOS 204
Netfs 242
Nifd 242
NO CARRIER 189
NTFS 263
NTLoader 268
NVidia 270

O

Ocrad 166
OSPF 204

P

PCI 99
PCI-E 29, 30, 31
Pcmcia 242
Ping 215
POP 204
Portmap 242
Postfix 242
Post-install 13
Post-remove 13
PPP 173, 204
Pppd 174
Pre-install 12
Pre-remove 13

R

Radio Ethernet 193
RAW 85
Reboot 234
Redundant Array
Independed Disk 48
Remove — во время удаления 13
Rhnsd 242
RIP 204
Rmdir 58
Rmmod 9
Rpc 242
Rpmdrake 160

S

Samba 146
SATA 45, 46, 50
Scannerdrake 157
Screentest 239
SCSI 45, 46, 47, 51, 60, 64
Sendmail 242
Setup 33
SLIP 173, 204
Smartd 242
Smartmontools 239
SMTP 204
Sshd 242
Startx 36
STP 194
Supermount 227
Swapon 244
SWEDAC 24
Switch 196
Sync 59
Syslogd 250
System-config-network 179
System-config-network-druid 174, 216
System-config-packages 160
System-config-printer 147
System-config-services 240

T

TAO 85
TCO 24
Transmission Control Protocol/Internet Protocol 203

U

Unmount 58
Usbview 227
USB-модемы 172
USB-сканеры 156
UTF 272
UTP 194

V

V.42bis 169
V.44 169
V.90 169
V.92 169
VGA 32
Vmware 239

W

WAN 192

X

X Window 270
X-CD-Roast 83
Xconfig 15
Xconfigurator 33
XF86Setup 33
Xfdrake 33
Xinit 35
XMMS 108
Xsane 159, 160
Xserver 35
Xvidtune 35

А

Амплитуда 93
Аналого-цифровой
преобразователь 225
АЦП 96, 155

Б, Г

Барабан 136
Глубина цвета 155

Д

Директива
domain 208
nameserver 208
search 208
Дорожка 42
Драйвер 7

И

Инструкция
alias 12
options 12
Интерфейс lo 210

К

Каталог /boot 261
Коммутатор 196
Коннектор 202
Контроллер клавиатуры 114
Концентратор 194

М

Маршрутизатор 206
Маска подсети 205
Микшер 96
Модем 167
Модуль 7, 8, 9, 10, 12, 13

Н, О

Нуль-модем 167
Оксидный слой 43
Оптическая плотность 155

П

Пантелеграф 153
Параметр single 234
Пластина 43
Пьезоголовочные
принтеры 140
Пьезокристалл 140

Р, С

Резолвер 207
Сервис network 210
Сервоблок 74
Скан-коды 115

Т, Ф, Ц, Ч, Ш, Я

Терминатор 194
Тонкопленочный слой 43
Фототелеграф 154
ЦАП 96
Частота амплитуды 93
Шлюз 206
Ядро 5