

Денис Колисниченко

СЕРВЕРНОЕ ПРИМЕНЕНИЕ Linux

3-е издание

Санкт-Петербург

«БХВ-Петербург»

2011

УДК 681.3.06
ББК 32.973.26-018.2
К60

Колисниченко Д. Н.

К60 Серверное применение Linux. — 3-е изд., перераб и доп. — СПб.: БХВ-Петербург, 2011. — 528 с.: ил. — (Системный администратор)

ISBN 978-5-9775-0652-6

Описана настройка различных типов серверов: Web-, FTP-, DNS-, DHCP-, почтового сервера, сервера баз данных. Подробно рассмотрена установка и базовая настройка операционной системы, настройка связки Apache + MySQL + PHP, дано общее устройство Linux и разобраны основные принципы работы с этой операционной системой. Отдельное внимание уделено защите сервера на базе Linux: настройка брандмауэра, защита маршрутизатора и точки доступа и т. д. Описана работа системы контроля доступа Tomoyo, прокси-серверов Squid и SquidGuard. Изложение основано на последних на момент написания книги версиях популярных дистрибутивов Fedora, Mandriva, Ubuntu, openSUSE.

Третье издание существенно дополнено новым материалом: рассматривается дистрибутив openSUSE, приводится расширенное описание брандмауэра iptables, настройка сети производится не только с помощью графических конфигураторов, но и с помощью конфигурационных файлов системы, рассмотрены средства резервного копирования remastersys, Clonezilla, Linux Live.

Для администраторов Linux и опытных пользователей

УДК 681.3.06
ББК 32.973.26-018.2

Группа подготовки издания:

Главный редактор	<i>Екатерина Кондукова</i>
Зам. главного редактора	<i>Евгений Рыбаков</i>
Зав. редакцией	<i>Григорий Добин</i>
Редактор	<i>Анна Кузьмина</i>
Компьютерная верстка	<i>Натальи Смирновой</i>
Корректор	<i>Наталья Першакова</i>
Дизайн серии	<i>Инны Тачиной</i>
Оформление обложки	<i>Елены Беляевой</i>
Зав. производством	<i>Николай Тверских</i>

Лицензия ИД № 02429 от 24.07.00. Подписано в печать 28.03.11.

Формат 70×100¹/₁₆. Печать офсетная. Усл. печ. л. 42,57.

Тираж 1500 экз. Заказ №

"БХВ-Петербург", 190005, Санкт-Петербург, Измайловский пр., 29.

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

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

ISBN 978-5-9775-0652-6

© Колисниченко Д. Н., 2011
© Оформление, издательство "БХВ-Петербург", 2011

Оглавление

ВВЕДЕНИЕ	1
О чем эта книга	1
Как следует читать эту книгу	1
Что нового в третьем издании?	1
Выбор дистрибутива	2
Поддержка читателей	3
 ЧАСТЬ I. УСТАНОВКА LINUX	5
 Глава 1. Особенности установки LINUX	7
1.1. Системные требования.....	7
1.2. Первоначальная загрузка	8
1.2.1. POST и загрузчики.....	8
1.2.2. Ядро Linux и его параметры	9
1.3. Проверка носителей	13
1.4. Изменение таблицы разделов	14
1.5. Выбор групп пакетов.....	19
1.6. Выбор графической среды.....	19
1.7. Установка пароля root	21
1.8. Создание учетных записей пользователей	22
1.9. Порядок установки операционных систем	22
1.10. Проблемы при установке.....	24
1.10.1. Ошибка: <i>kernel panic:VFS: Unable to mount root fs</i>	24
1.10.2. Проблемы с некоторыми LCD-мониторами	24
1.10.3. Список известных проблем в Mandriva Linux 2010.....	24
1.10.4. Не переключается раскладка в Fedora 13	24
1.10.5. Установка Linux на HP Mini 2133 (проблема с ACPI)	25
1.10.6. Проблема с ACPI на Fujitsu Siemens esprimο mobile u9200	25
1.10.7. Писк при выключении или перезагрузке компьютера в Mandriva	25
1.10.8. Mandriva One не запускается на компьютерах с видеокартой nvidia	26
 Глава 2. ВХОД В СИСТЕМУ.....	27
2.1. Графический и текстовый вход в систему. Завершение работы.....	27
2.2. Переключение в консоль из графического интерфейса.....	29
2.3. Вход в систему как root.....	29

ГЛАВА 3. РЕШЕНИЕ ПРОБЛЕМ ПРИ УСТАНОВКЕ И ПОСЛЕ НЕЕ	30
3.1. Небольшие проблемы с LCD-мониторами	30
3.2. Проблема с показом времени в Ubuntu	30
3.3. Зависание графического интерфейса в процессе работы	31
3.4. Медленная загрузка системы.....	31
3.5. Забыт пароль root.....	31
 ЧАСТЬ II. ФАЙЛОВАЯ СИСТЕМА.....	35
ГЛАВА 4. ПОДДЕРЖИВАЕМЫЕ ФАЙЛОВЫЕ СИСТЕМЫ	37
4.1. Какие файловые системы поддерживает Linux?	37
4.1.1. Файловые системы ext2, ext3 и ext4.....	37
4.1.2. Файловые системы XFS, ReiserFS и JFS.....	38
4.1.3. Особенности файловой системы ext4	39
Сравнение ext3 и ext4	39
Совместимость с ext3.....	40
Переход на ext4	41
4.1.4. Выбор файловой системы	41
4.2. Файловые системы Windows	42
4.3. Сменные носители.....	43
 ГЛАВА 5. ОСОБЕННОСТИ ФАЙЛОВОЙ СИСТЕМЫ LINUX.....	44
5.1. Имена файлов в Linux	44
5.2. Файлы и устройства.....	44
5.3. Корневая файловая система и монтирование	45
5.4. Стандартные каталоги Linux	49
 ГЛАВА 6. КОМАНДЫ ДЛЯ РАБОТЫ С ФАЙЛАМИ И КАТАЛОГАМИ.	
ПРАВА ДОСТУПА	50
6.1. Работа с файлами.....	50
6.2. Работа с каталогами	52
6.3. Команды для работы со ссылками.....	54
6.4. Права доступа. Команды <i>chown</i> , <i>chmod</i> и <i>chattr</i>	55
6.4.1. Права доступа к файлам и каталогам	55
6.4.2. Смена владельца файла	57
6.4.3. Специальные права доступа	57
6.4.4. Атрибуты файла. Запрет изменения файла	58
6.5. Команды поиска файлов	58

ГЛАВА 7. МОНТИРОВАНИЕ ФАЙЛОВЫХ СИСТЕМ.....	60
7.1. Команда <i>mount</i>	60
7.2. Файлы устройств и монтирование	61
7.3. Типы файловых систем	62
7.4. Монтирование разделов при загрузке	63
7.4.1. Формат файла <i>/etc/fstab</i>	63
7.4.2. Подробно о <i>UUID</i> и <i>/etc/fstab</i>	65
7.5. Flash-диски	67
ГЛАВА 8. ОСОБЫЕ ОПЕРАЦИИ ПРИ РАБОТЕ С ФАЙЛОВОЙ СИСТЕМОЙ	70
8.1. Создание и монтирование образов <i>CD/DVD</i>	70
8.2. Запись образов на болванку.....	71
8.3. Программа <i>mkisofs</i>	72
8.4. Преобразование образов дисков	73
8.5. Создание и монтирование файлов с файловой системой	73
8.6. Создание файла подкачки.....	74
8.7. Создание файловой системы	74
8.8. Проверка и восстановление файловой системы	75
8.9. Смена корневой файловой системы. Команда <i>chroot</i>	75
8.10. Работа с журналом файловой системы.....	76
8.11. Монтирование <i>NTFS</i>	77
8.12. Установка скорости <i>CD/DVD</i>	77
8.13. Псевдофайловая система <i>/proc</i>	78
8.13.1. Информационные файлы	78
8.13.2. Файлы, позволяющие изменять параметры ядра	79
8.13.3. Файлы, изменяющие параметры сети.....	80
8.13.4. Файлы, изменяющие параметры виртуальной памяти	80
8.13.5. Файлы, позволяющие изменить параметры файловых систем	81
8.13.6. Как сохранить изменения?.....	81
ГЛАВА 9. ПОДДЕРЖКА RAID В LINUX.....	82
9.1. Что такое <i>RAID</i>	82
9.2. Программные <i>RAID</i> -массивы.....	84
9.3. Создание программных массивов	85
9.4. Использование <i>RAID</i> -массива	88
9.5. Сбой и его имитация	88
ГЛАВА 10. ЗАПИСЬ CD/DVD В LINUX	90
10.1. <i>CD/DVD</i> — оптимальное решение для резервных копий	90
10.2. Форматы и маркировка <i>DVD</i> -дисков	90
10.3. Программа <i>k3b</i>	93

10.4. Использование стандартных средств записи CD/DVD в Ubuntu.....	99
10.5. Программа Nero для Linux.....	101
10.6. Чтение "битых" компакт-дисков	102
ГЛАВА 11. РЕЗЕРВНОЕ КОПИРОВАНИЕ	103
11.1. Зачем нужно делать резервные копии	103
11.2. Выбор носителя для резервной копии	104
11.3. Правила хранения носителей с резервными копиями	105
11.4. Стратегии создания резервной копии.....	106
11.5. Программа tar.....	107
11.6. Сетевое резервное копирование.....	109
ГЛАВА 12. РЕДАКТИРОВАНИЕ ТАБЛИЦЫ РАЗДЕЛОВ ЖЕСТКОГО ДИСКА	110
12.1. Когда и зачем нужно редактировать таблицу разделов	110
12.2. Использование fdisk	111
12.3. Утилита parted — изменение размера разделов и восстановление таблицы разделов.....	114
12.4. Графические редакторы таблицы разделов diskdrake и gparted.....	115
12.5. Программа testdisk — восстановление случайно удаленных разделов.....	115
ЧАСТЬ III. ПОЛЬЗОВАТЕЛИ И ГРУППЫ	117
ГЛАВА 13. ПОЛЬЗОВАТЕЛИ И ГРУППЫ	119
13.1. Многопользовательская система	119
13.2. Создание, удаление и модификация пользователей стандартными средствами.....	120
13.3. Группы пользователей	122
13.3.1. Управление пользователями и группами с помощью графических конфигураторов.....	123
13.3.2. Конфигуратор system-config-users в Fedora	123
13.3.3. Конфигуратор drakuser в Linux Mandriva	124
13.3.4. Пользователи и группы в Ubuntu	124
13.3.5. Графический конфигуратор в openSUSE	129
ГЛАВА 14. ПОЛЬЗОВАТЕЛЬ ROOT	135
14.1. Максимальные полномочия	135
14.2. Как работать без root.....	136
14.2.1. Команда <i>sudo</i>	136
Команда <i>su</i>	137
14.2.2. Проблемы с <i>sudo</i> в Ubuntu и Kubuntu	138
14.2.3. Ввод серии команд <i>sudo</i>	139

14.3. Переход к традиционной учетной записи root.....	139
14.3.1. Преимущества и недостатки <i>sudo</i>	139
14.3.2. Традиционная учетная запись root в Ubuntu	140
14.3.3. Традиционная учетная запись root в Mandriva.....	141
14.3.4. Вход в качестве root в Fedora.....	142
ГЛАВА 15. ОГРАНИЧЕНИЕ ДИСКОВОГО ПРОСТРАНСТВА	144
15.1. Квотирование — это полезно!.....	144
15.2. Включение квот	144
15.3. Задание и просмотр квот.....	147
15.4. Прототипы.....	149
ЧАСТЬ VI. ЗАГРУЗКА И ИНИЦИАЛИЗАЦИЯ LINUX	151
ГЛАВА 16. ЗАГРУЗЧИКИ LINUX.....	153
16.1. Основные загрузчики	153
16.2. Конфигурационные файлы GRUB и GRUB2.....	154
16.2.1. Конфигурационный файл GRUB	154
16.2.2. Конфигурационный файл GRUB2	157
16.3. Команды установки загрузчиков	161
16.4. Установка тайм-аута выбора операционной системы. Редактирование параметров ядра	162
16.5. Установка собственного фона загрузчика GRUB и GRUB2	164
16.6. Постоянные имена и GRUB.....	165
16.7. Восстановление загрузчика GRUB/GRUB2	166
16.8. Две и более ОС Linux на одном компьютере.....	167
16.9. Загрузка с ISO-образов.....	169
16.10. Установка пароля загрузчика	169
16.10.1. Загрузчик GRUB	170
16.10.2. Загрузчик GRUB2	170
ГЛАВА 17. СИСТЕМЫ ИНИЦИАЛИЗАЦИИ LINUX	173
17.1. Начальная загрузка Linux	173
17.2. Система инициализации init	174
17.2.1. Файл <i>/etc/inittab</i>	174
17.2.2. Команда <i>init</i>	176
17.2.3. Команда <i>service</i>	176
17.2.4. Редакторы уровней запуска	176
17.3. Система инициализации upstart.....	179
17.3.1. Как работает upstart	179
17.3.2. Конфигурационные файлы upstart	180
17.4. Система инициализации Slackware.....	181

Глава 18. ПРОЦЕССЫ. УПРАВЛЕНИЕ ПРОЦЕССАМИ. СЕРВИСЫ	184
18.1. Управление процессами.....	184
18.2. Управление сервисами	187
18.3. Отключение неиспользуемых сервисов	188
 ЧАСТЬ V. КОМАНДНАЯ СТРОКА.....	 191
Глава 19. КОНСОЛЬ LINUX	193
19.1. Что такое консоль	193
19.2. Правильная работа в консоли.....	194
19.3. Служебные команды. Псевдонимы команд	195
19.4. Приглашение командной строки и права пользователя	196
19.5. Эмуляторы консоли.....	196
19.6. Перенаправление ввода/вывода	196
 Глава 20. ПОЛЕЗНЫЕ КОМАНДЫ.....	 198
20.1. Команды, о которых нужно знать каждому администратору	198
20.2. Общие команды	198
20.2.1. Команда <i>arch</i> — вывод архитектуры компьютера	198
20.2.2. Команда <i>clear</i> — очистка экрана	198
20.2.3. Команда <i>date</i>	199
20.2.4. Команда <i>echo</i>	199
20.2.5. Команда <i>exit</i> — выход из системы.....	199
20.2.6. Команда <i>man</i> — вывод справки	199
20.2.7. Команда <i>passwd</i> — изменение пароля.....	199
20.2.8. Команда <i>startx</i> — запуск графического интерфейса X Org	200
20.2.9. Команда <i>uptime</i> — информация о работе системы	200
20.2.10. Команда <i>users</i> — информация о пользователях	200
20.2.11. Команды <i>w</i> , <i>who</i> и <i>whoami</i> — информация о пользователях	201
20.2.12. Команда <i>xf86config</i> — настройка графической подсистемы	201
20.3. Команды для работы с текстом	202
20.3.1. Команда <i>diff</i> — сравнение файлов	202
20.3.2. Команда <i>grep</i> — текстовый фильтр	202
20.3.3. Команды <i>more</i> и <i>less</i> — страничный вывод.....	203
20.3.4. Команды <i>head</i> и <i>tail</i> — вывод начала и хвоста файла.....	203
20.3.5. Команда <i>wc</i> — подсчет слов в файле.....	203
20.4. Команды системного администратора	203
20.4.1. Команды <i>free</i> и <i>df</i> — информация о системных ресурсах.....	203
20.4.2. Команда <i>md5sum</i> — вычисление контрольного кода MD5	204
20.4.3. Команды <i>ssh</i> и <i>telnet</i> — удаленный вход в систему	204
20.5. Команды <i>vi</i> , <i>nano</i> , <i>ee</i> , <i>mcedit</i> , <i>pico</i> : текстовые редакторы	204

ГЛАВА 21. КОМАНДНЫЙ ИНТЕРПРЕТАТОР BASH	209
21.1. Автоматизация задач с помощью bash	209
21.2. Привет, мир!	210
21.3. Использование переменных в собственных сценариях	210
21.4. Передача параметров сценарию	211
21.5. Массивы и bash	212
21.6. Циклы	212
21.7. Условные операторы	213
 ЧАСТЬ VI. УСТАНОВКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ	213
 ГЛАВА 22. ПЛАНИРОВЩИКИ ЗАДАЧ	215
22.1. Зачем нужен планировщик задач	215
22.2. Планировщик cron	215
22.3. Планировщик at	217
22.4. Разовое выполнение команд — демон atd	218
 ГЛАВА 23. ПАКЕТ. МЕНЕДЖЕР ПАКЕТОВ RPM	221
23.1. Что такое пакет?	221
23.2. Программы для управления пакетами	223
23.3. Программа RPM (все RH-совместимые дистрибутивы)	225
23.4. Графический менеджер пакетов rpm-drake (Mandrake и Mandriva)	225
23.5. Программа urpmi	228
23.5.1. Установка пакетов. Управление источниками пакетов	228
23.5.2. Обновление и удаление пакетов	233
23.5.3. Поиск пакета. Получение информации о пакете	233
 ГЛАВА 24. ПРОГРАММЫ DPKG И APT: УСТАНОВКА ПАКЕТОВ В DEBIAN/UBUNTU	234
24.1. Программа dpkg	234
24.2. Программа apt	235
24.3. Установка RPM-пакетов в Debian/Ubuntu	237
24.4. Графический менеджер Synaptic в Ubuntu	237
 ГЛАВА 25. ПРОГРАММЫ YUM И GPG-APPLICATION	239
25.1. Программа yum	239
25.1.1. Общая информация о программе	239
25.1.2. Установка пакетов	240
25.1.3. Управление источниками пакетов	241

25.1.4. Установка пакетов через прокси-сервер.....	243
25.1.5. Плагины для yum	243
25.2. Графический менеджер пакетов в Fedora: gpk-application.....	243
ГЛАВА 26. УПРАВЛЕНИЕ ПАКЕТАМИ В OPENSUSE.....	245
26.1. Источники пакетов <i>zypper</i>	245
26.2. YMP-файлы	246
26.3. Использование <i>zypper</i>	247
ЧАСТЬ VII. СЕТЬ И ИНТЕРНЕТ	249
ГЛАВА 27. НАСТРОЙКА ЛОКАЛЬНОЙ СЕТИ	251
27.1. Локальная сеть с использованием технологии Fast Ethernet	251
27.2. Файлы конфигурации сети в Linux	253
27.3. Настройка сети с помощью конфигуратора.....	255
27.3.1. Настройка сети в Linux Mandriva	256
27.3.2. Настройка сети в Fedora	263
27.3.3. Настройка сети в Debian, Ubuntu и Denix Конфигураторы nm-connection-editor (NetworkManager) и network-admin	268
27.3.4. Конфигуратор netconfig в Slackware	271
27.4. Проблемы с ноутбуком Acer eMachines E525	271
27.5. Утилиты для диагностики соединения	272
27.6. Для фанатов, или как настроить сеть вручную	276
27.6.1. Конфигурационные файлы Fedora	277
27.6.2. Конфигурационные файлы openSUSE.....	279
27.6.3. Конфигурационные файлы Debian/Ubuntu.....	280
27.7. Команда <i>mii-tool</i>	281
27.8. Еще несколько слов о настройке сети	282
ГЛАВА 28. НАСТРОЙКА ADSL-ДОСТУПА К ИНТЕРНЕТУ	283
28.1. Причина популярности DSL-соединений	283
28.2. Физическое подключение ADSL-модема.....	284
28.3. Настройка DSL-соединения.....	284
28.3.1. В Fedora.....	284
28.3.2. В openSUSE	286
28.3.3. В Ubuntu.....	291
28.3.4. В Mandriva	294
ГЛАВА 29. КОМАНДЫ ДЛЯ РАБОТЫ С СЕТЬЮ И ИНТЕРНЕТОМ.....	295
29.1. Команда <i>ifconfig</i> : управление сетевыми интерфейсами.....	295
29.2. Текстовые браузеры	296

29.3. Команда <i>ftp</i> : FTP-клиент	297
29.4. Команда <i>wget</i> : загрузка файлов	298
29.5. Команда <i>mail</i> — чтение почты и отправка сообщений.....	300
 ЧАСТЬ VIII. LINUX-СЕРВЕР	191
 ГЛАВА 30. СУПЕРСЕРВЕР XINETD	303
30.1. Сетевые сервисы и суперсервер.....	303
30.2. Конфигурационный файл суперсервера	303
 ГЛАВА 31. WEB-СЕРВЕР. СВЯЗКА APACHE + PHP + MYSQL.....	305
31.1. Самый популярный Web-сервер	305
31.2. Установка Web-сервера и интерпретатора PHP. Выбор версии	305
31.3. Тестирование настроек	307
31.4. Файл конфигурации Web-сервера.....	309
31.4.1. Базовая настройка.....	309
31.4.2. Самые полезные директивы файла конфигурации.....	310
31.4.3. Директивы <i>Directory</i> , <i>Limit</i> , <i>Location</i> , <i>Files</i>	311
31.5. Управление запуском сервера Apache.....	314
31.6. Пользовательские каталоги	314
31.7. Установка сервера баз данных MySQL	315
31.7.1. Установка сервера	315
31.7.2. Изменение пароля root и добавление пользователей	315
31.7.3. Запуск и останов сервера	317
31.7.4. Программа MySQL Administrator.....	317
 ГЛАВА 32. FTP-СЕРВЕР	319
32.1. Зачем нужен FTP	319
32.2. Установка FTP-сервера	320
32.3. Конфигурационный файл	320
32.4. Настройка реального сервера	325
32.5. Программы ftpwho и ftpcount	326
32.6. Конфигуратор gproftpd.....	327
 ГЛАВА 33. ПОЧТОВЫЙ СЕРВЕР.....	328
33.1. Что такое Qmail.....	328
33.2. Подготовка к установке Qmail	328
33.3. Установка Qmail и необходимых дополнений	330
33.3.1. Загрузка и установка Qmail.....	330
33.3.2. Установка ucspi-tcp и daemontools	331

33.3.3. Установка EZmlm — средства для создания рассылки	331
33.3.4. Установка Autoresponder — автоответчика	332
33.3.5. Установка MailDrop — фильтра для сообщений.....	332
33.3.6. Установка QmailAdmin — Web-интерфейса для настройки Qmail	332
33.4. Настройка после установки и запуск Qmail	333
33.5. Настройка почтовых клиентов	335
33.6. Дополнительная информация.....	336
ГЛАВА 34. DNS-СЕРВЕР	337
34.1. Еще раз о том, что такое DNS	337
34.2. Кэширующий сервер DNS.....	338
34.3. Полноценный DNS-сервер.....	344
34.4. Вторичный DNS-сервер	348
34.5. Обновление базы данных корневых серверов.....	349
ГЛАВА 35. DHCP-СЕРВЕР	352
35.1. Протокол динамической конфигурации узла	352
35.2. Конфигурационный файл DHCP-сервера	352
35.3. База данных аренды.....	354
35.4. Полный листинг конфигурационного файла	354
35.5. Управление сервером DHCP	355
35.6. Настройка клиентов	355
ГЛАВА 36. ПРОКСИ-СЕРВЕР SQUID	356
36.1. Зачем нужен прокси-сервер в локальной сети?	356
36.2. Базовая настройка Squid	356
36.3. Практические примеры	358
36.3.1. Управление доступом.....	358
36.3.2. Создание черного списка URL	359
36.3.3. Отказ от баннеров.....	359
36.4. Управление прокси-сервером.....	359
36.5. Настройка клиентов	359
36.6. Прозрачный прокси-сервер	360
36.7. Расширение squidGuard.....	361
ГЛАВА 37. МАРШРУТИЗАЦИЯ И НАСТРОЙКА БРАНДМАУЭРА.....	364
37.1. Краткое введение в маршрутизацию	364
37.2. Таблица маршрутизации ядра. Установка маршрута по умолчанию	365
37.3. Изменение таблицы маршрутизации. Команда <i>route</i>	369
37.4. Включение IPv4-переадресации или превращение компьютера в шлюз.....	371
37.5. Настройка брандмауэра	372

37.5.1. Что такое брандмауэр	372
37.5.2. Цепочки и правила	373
37.5.3. Использование iptables	376
37.5.4. Шлюз своими руками	379
ГЛАВА 38. СЕРВЕР ВРЕМЕНИ	385
38.1. Проблема синхронизации времени	385
38.2. Настройка сервера и Linux-клиентов	385
38.3. Настройка Windows-клиентов	387
ГЛАВА 39. ВИРТУАЛЬНЫЕ ЧАСТНЫЕ СЕТИ	389
39.1. Для чего нужна виртуальная частная сеть	389
39.2. Необходимое программное обеспечение	390
39.3. Канал для передачи данных VPN	390
39.3.1. Соединение "сеть—сеть"	390
39.3.2. Соединение "клиент—сеть"	391
39.4. Настройка соединения "сеть—сеть"	391
39.4.1. Установка OpenS/WAN	391
39.4.2. Немного терминологии	392
39.4.3. Генерирование ключей	392
39.4.4. Конфигурационный файл	392
39.4.5. Установка VPN-соединения	396
39.4.6. Настройка iptables	396
39.5. Настройка соединения "клиент—сеть"	396
39.5.1. Редактирование конфигурационных файлов	397
39.5.2. Настройка Linux-клиента	399
39.5.3. Настройка Windows-клиента	402
В Windows 2000/XP	402
В Windows Vista/Windows 7	405
ГЛАВА 40. СЕРВИС SAMBA	407
40.1. Установка Samba	407
40.2. Базовая настройка Samba	407
40.3. Настройка общих ресурсов	408
40.4. Просмотр ресурсов Windows-сети	410
40.5. Оптимизация Samba	411
ГЛАВА 41. УДАЛЕННЫЙ ДОСТУП	412
41.1. Зачем нужен удаленный доступ	412
41.2. Протокол SSH	412
41.3. X-терминалы	418

ГЛАВА 42. ОПТИМИЗАЦИЯ СЕРВЕРА И РАБОЧЕЙ СТАНЦИИ.....	422
42.1. Общая оптимизация Linux	422
42.1.1. Оптимизация подкачки	422
42.1.2. Изменение планировщика ввода/вывода	424
42.2. Оптимизация сетевых сервисов	425
42.2.1. Секреты оптимизации Samba	425
42.2.2. Оптимизация ProFTPD	426
42.2.3. Оптимизация Apache	428
ГЛАВА 43. СЕРВЕР MySQL	430
43.1. Сервер баз данных MySQL	430
43.2. Установка сервера	430
43.3. Изменение пароля root и добавление пользователей	431
43.4. Базовые MySQL-операторы	433
43.5. Запуск и останов сервера	436
ГЛАВА 44. СЕТЕВАЯ ФАЙЛОВАЯ СИСТЕМА NFS.....	437
44.1. Установка сервера и клиента	437
44.2. Настройка сервера	438
44.3. Монтирование удаленных файловых систем	439
ЧАСТЬ IX. ЗАЩИТА LINUX-СЕРВЕРА	441
ГЛАВА 45. АНТИВИРУС CLAMAV	443
45.1. Зачем нужен антивирус в Linux	443
45.2. Установка ClamAV	444
45.3. Проверка файловой системы	444
45.4. Прозрачная проверка почты	445
45.5. Проверка Web-трафика	446
45.6. Клиентский антивирус	448
ГЛАВА 46. ЗАЩИТА ПОПУЛЯРНЫХ СЕТЕВЫХ СЕРВИСОВ	449
46.1. Защита Apache	449
46.2. Защита FTP	449
46.3. Защита DNS	450
46.4. Защита Samba	451
46.5. DHCP: привязка к MAC-адресу	452
46.6. Защита от спама: greylisting и qmail	454

ГЛАВА 47. CHROOT-ОКРУЖЕНИЯ	456
47.1. Песочница.....	456
47.2. Пример создания chroot-окружения.....	457
ГЛАВА 48. УПРАВЛЕНИЕ ДОСТУПОМ	459
48.1. Что такое Tomyo	459
48.2. Установка Tomyo. Готовые LiveCD.....	459
48.3. Инициализация системы	460
ГЛАВА 49. ЗАЩИТА ТОЧКИ ДОСТУПА.....	465
49.1. Изменение параметров по умолчанию	465
49.2. Отключение широковещания SSID	466
49.3. Используйте WPA	466
49.4. Фильтрация MAC-адресов.....	467
49.5. Обновление прошивки оборудования	468
49.6. Использование аутентификации	468
49.7. Понижение мощности передачи	469
49.8. Отключение точки доступа, когда вы не работаете	470
49.9. Защита портов управления	470
49.10. Защита от внешних угроз. Общая защита сети	470
ГЛАВА 50. ЗАЩИТА МАРШРУТИЗАТОРА	471
50.1. О маршрутизаторе	471
50.2. Установка пароля.....	472
50.3. Ограничение доступа по сети.....	472
50.4. Только локальный доступ.....	472
50.5. Защита SNMP.....	473
50.6. Ведение журналов	473
50.7. Отключение ненужных сервисов.....	473
50.8. Ограничение ICMP	474
50.9. Отключение потенциально опасных опций.....	474
50.10. Анти-spoofing и защита от DoS-атак	474
50.11. Отключение CDP.....	475
ГЛАВА 51. СРЕДСТВА РЕЗЕРВНОГО КОПИРОВАНИЯ.	
СОЗДАНИЕ ISO-ДИСКА.....	476
51.1. Необходимость в "живой" резервной копии.....	476
51.2. Какие средства мы будем рассматривать.....	477
51.3. Clonezilla.....	478
51.4. Remastersys Backup.....	484
51.5. Linux Live.....	486

ГЛАВА 52. ЧТО ДЕЛАТЬ В СЛУЧАЕ ВЗЛОМА?	487
52.1. 100% безопасности не гарантируется	487
52.2. Ваши действия в наихудшем варианте развития событий	488
52.2.1. Своя учетная запись	489
52.2.2. Файлы <code>hosts.allow</code> и <code>hosts.deny</code>	489
52.2.3. Сетевая файловая система	489
52.2.4. Руткиты	489
52.2.5. Модули ядра	490
52.2.6. Удаленный командный интерпретатор	491
52.2.7. Настройка PHP и CGI	491
52.2.8. SSH — огромная дыра	492
ЗАКЛЮЧЕНИЕ	493
ПРИЛОЖЕНИЯ	495
Приложение 1. Настройка принтера в Linux	497
Приложение 2. Параметры ядра	499
Приложение 3. "Горячее" администрирование с помощью /proc	501
ПЗ.1. Информационные файлы	501
ПЗ.2. Файлы, позволяющие изменять параметры ядра	502
ПЗ.3. Файлы, изменяющие параметры сети	503
ПЗ.4. Файлы, изменяющие параметры виртуальной памяти	504
ПЗ.5. Файлы, позволяющие изменить параметры файловых систем	504
ПЗ.6. Как сохранить изменения?	505
ПРЕДМЕТНЫЙ УКАЗАТЕЛЬ	506

Введение

О чем эта книга

Данная книга посвящена настройке сервера на базе операционной системы Linux. Наверное, у многих читателей, знакомых с другими моими книгами, сразу возникнет вопрос: что лучше, эта книга или "Linux-сервер своими руками"?

Данная книга — это не следующее развитие "Linux-сервера". Это полностью самостоятельная книга. Но две "серверные" книги — не конкуренты. Они предназначены дополнять друг друга. Например, в этой книге описываются дистрибутивы Ubuntu, openSUSE, Fedora и Mandriva, а в пятом издании "Linux-сервера" — Mandriva и Fedora — я решил не изменять традиции. Зато в "Linux-сервере" описана настройка некоторых сетевых служб, которые не затронуты в этой книге.

Как следует читать эту книгу

Данная книга не учебник, поэтому совсем не обязательно ее читать последовательно. Однако если вы новичок в Linux, то настоятельно рекомендую читать книгу, не "перепрыгивая" сразу через десять глав.

Что нового в третьем издании?

По сравнению со вторым изданием этой книги изменений очень много: из 52 глав без изменений остались лишь 12. Из оставшихся 40 глав половина переписана полностью, а в остальных существенные изменения. Подробно описывать изменения в каждой главе не стану: если у вас есть первое или второе издание этой книги, нововведения вы сами заметите. А если вы купили эту книгу впервые, то и говорить о них незначем.

Вкратце опишу лишь самые значительные изменения. Во-первых, удален неактуальный материал: настройка модемного соединения, GPRS-соединения, информация о компиляции ядра, ликбез о типах DVD-дисков и прочий ненужный мате-

риал, за который вы платили деньги, покупая эту книгу. Модемные соединения уже давно "канули в Лету", GPRS/EDGE-соединение на сервере, по большому счету, не нужно: в случае обрыва высокоскоростного канала медленное беспроводное соединение не спасет всю сеть. Информация об истории создания DVD не имеет никакого отношения к резервному копированию, а перекомпилировать ядро приходится все реже и реже. Чаще администратор загружает и устанавливает пакет с уже откомпилированным ядром, содержащим все необходимые ему функции.

Вместо всего этого добавлена информация о новой файловой системе ext4, новом загрузчике GRUB2, подмене MAC-адреса сетевой платы, менеджере пакетов zypper (используется в openSUSE). Появилось нормальное описание брандмауэра iptables вместо неполноценной программы Firestarter. Вместо SELinux описывается система управления доступом Tomoyo — мне она показалась более простой и логичной, чем SELinux, но при этом не менее надежной. А в предпоследней главе описываются утилиты для создания резервных копий системы в виде LiveDVD: Clonezilla, Remastersys и др.

Выбор дистрибутива

В этой книге рассматривается несколько дистрибутивов Linux: Ubuntu 10.04 (Debian), Fedora 13, Mandriva 2010.1 Spring, частично рассмотрен дистрибутив openSUSE 11.3. На момент написания данных строк это самые последние версии упомянутых дистрибутивов, хотя все сказанное в книге применимо к предыдущим (Ubuntu 9.x, Fedora 11-12, Mandriva 2009 и 2010.0) и, надеюсь, к следующим версиям.

Дистрибутив Debian затронут в книге косвенно — ведь Ubuntu основан на Debian и является его близким родственником, но основной упор в книге сделан именно на Ubuntu.

Какой дистрибутив лучше использовать для сервера? Любой дистрибутив можно настроить и использовать в качестве сервера. Все зависит от того, насколько правильно вы его настроите. Раньше о пригодности применения того или иного дистрибутива можно было судить по наличию пакетов с серверными службами, которые включены в состав дистрибутива. Сейчас в большинстве случаев загрузка необходимых пакетов выполняется из интернет-репозитариев, а в них есть все необходимые пакеты. Так что такой критерий, как наличие дополнительного программного обеспечения, во внимание не берется.

Хочется отметить дистрибутив Fedora: неплохой дистрибутив, если не считать небольших проблем с переводом (до сих пор не все элементы интерфейса переведены на русский, хотя, такая проблема может появиться в любом дистрибутиве), не очень эффективная система установки пакетов. Но раньше многие администраторы предпочитали устанавливать на сервер именно Red Hat — это предок Fedora.

Ubuntu можно использовать как в качестве дистрибутива для рабочей станции, так и для сервера. Вообще-то Ubuntu — это десктопный дистрибутив, т. е. по умолчанию (если не устанавливать дополнительных пакетов) он ориентирован на рабо-

чую станцию. Но в этой книге я полностью развеял миф о том, что Ubuntu нельзя использовать в качестве сервера. Можно и даже нужно, поскольку Ubuntu — это простой и надежный дистрибутив. Кстати, получить дистрибутивные диски с Ubuntu вы можете на сайте **www.ubuntu.com**, причем совершенно бесплатно.

openSUSE — стабильный и относительно простой дистрибутив, который можно использовать и в качестве сервера, и в качестве рабочей станции. Последние два настроенных мною сервера были как раз на базе openSUSE. Каких-либо особых недостатков у этого дистрибутива мною замечено не было.

Mandriva 2010 — очень хороший дистрибутив, хотя версии пакетов, входящие в его состав, не самые современные, но таков выбор разработчиков. Субъективно: он работает не так быстро, как Ubuntu, но мне кажется, что это не проблема дистрибутива, а KDE, поскольку если я работаю в Mandriva, то предпочитаю KDE. Но графический интерфейс для сервера не имеет никакого значения — его можно вообще отключить.

Итак, выбирайте тот дистрибутив, который вам больше нравится, а данная книга поможет его настроить.

Поддержка читателей

Каждый читатель этой книги может рассчитывать на *посильную* помощь автора в настройке Linux на форуме **www.dkws.org.ua**. Поддержка читателей осуществляется только на форуме. Найти мой электронный адрес в Интернете не составит труда, но я не гарантирую, что отвечу на ваше письмо.



ЧАСТЬ I

УСТАНОВКА LINUX

ГЛАВА 1



Особенности установки Linux

Установка Linux совсем не похожа на установку привычной многим операционной системы Windows. В этой главе мы поговорим об особенностях установки Linux, которые вы просто обязаны знать до начала работы с ней. Зная эти особенности, установить Linux сможет даже новичок — ведь вся установка проходит в графическом режиме, да еще и на русском языке, что существенно облегчает весь процесс.

Забегая вперед (об этом мы еще поговорим позже), хочу сразу предупредить, что Windows нужно устанавливать до Linux, потому что загрузчик Linux без проблем загружает все имеющиеся версии Windows, а вот заставить загрузчик Windows загружать Linux довольно сложно. Поэтому, дабы не усложнять себе жизнь, сначала установите все нужные версии Windows, а затем — все необходимые дистрибутивы Linux.

1.1. Системные требования

Современные дистрибутивы Linux, с одной стороны, не очень требовательны к системным ресурсам, а, с другой стороны, 256 Мбайт для запуска графической программы установки — это слишком! Да, некоторые дистрибутивы требуют для запуска программы установки в графическом режиме 256 Мбайт (а некоторые даже больше — см. примечание далее) оперативной памяти. Если у вас меньше оперативки (например, вы хотите создать шлюз из запывлившегося в углу старенького компьютера), установка будет происходить в текстовом режиме.

FEDORA 13

Fedora 13 вообще меня неприятно удивила. Попытался запустить инсталлятор на стареньком компьютере (256 Мбайт ОЗУ), но инсталлятор запустился только в текстовом режиме. Ради интереса я попробовал запустить установку в виртуальной машине с 384 Мбайт ОЗУ (иногда встречаются компьютеры, где установлены два модуля памяти — $256 + 128$, но найти такой сейчас мне сложно, поэтому пришлось тестировать в VMWare). И что вы думаете? Инсталлятор тоже запустился в текстовом режиме. А ведь на сайте Fedora сказано, что для запуска инсталляции в графическом режиме нужно ми-

нимум 384 Мбайт! А вот "графика" пошла, когда было установлено 512 честных мегабайтов. Fedora 12 я не пытался запустить на старом компьютере, вполне возможно, что такая же "особенность" есть и у двенадцатой версии дистрибутива. Ради справедливости нужно отметить, что последние версии других дистрибутивов (в частности Ubuntu 10.04 и openSUSE 11.3) запускаются на компьютере с 256 Мбайт ОЗУ в графическом режиме.

Что же касается дискового пространства, то ориентируйтесь минимум на 4—5 Гбайт (это с небольшим запасом — ведь еще нужно оставить место для своих данных), что вполне приемлемо по нынешним меркам, учитывая, что после установки вы получаете не "голую" систему, а уже практически готовую к работе — с офисными пакетами и мультимедиа-программами. Был неприятно удивлен, когда Ubuntu 8.04 (да, речь идет о старой версии Ubuntu, не говоря уже о новых) сообщила, что ей для установки нужно как минимум 5 Гбайт. Да, после установки она заняла менее 3 Гбайт, но на время установки ей понадобилось дополнительное место для хранения временных файлов. Особо неприятно было то, что я по привычке "отрезал" раздел в 4 Гбайт...

Если вы настраиваете сервер, то все офисные и мультимедиа-программы, понятно, можно не устанавливать. Тогда для самой системы понадобится максимум 2 Гбайт (с графическим интерфейсом и необходимыми пакетами, содержащими программы-серверы), но не нужно забывать, что само слово "сервер" подразумевает достаточное количество дискового пространства. Получается, что потребуется 2 Гбайт — для самой системы и еще N Гбайт для данных, которые будет обрабатывать сервер.

1.2. Первоначальная загрузка

1.2.1. POST и загрузчики

После включения питания компьютера запускается процедура самотестирования (Power On Self Test, POST), проверяющая основные компоненты системы: видеокарту, оперативную память, жесткие диски и т. д. Затем начинается загрузка операционной системы. Компьютер ищет на жестком диске (и других носителях) программу-загрузчик операционной системы. Если такая программа найдена, то ей передается управление, если же такая программа не найдена ни на одном из носителей, то выдается сообщение с просьбой вставить загрузочный диск.

В настоящее время популярны два загрузчика Linux: LILO и GRUB. GRUB является более современным и используется по умолчанию в большинстве дистрибутивов. Так что после установки Linux начальным загрузчиком будет именно GRUB (если вы самостоятельно не выберете другой загрузчик). Некоторые дистрибутивы имеют собственные загрузчики — например, ASPLinux использует загрузчик ASPLoader.

Относительно недавно появилась новая версия GRUB — GRUB-PC, или GRUB-2. Особенность этой версии — возможность загружать Linux с раздела ext4 и другой,

более гибкий, файл конфигурации. Новая версия GRUB также будет рассмотрена в этой книге.

Задача загрузчика — предоставить пользователю возможность выбрать нужную операционную систему (ведь кроме Linux на компьютере может стоять и другая операционная система) и передать ей управление. В случае с Linux загрузчик загружает ядро операционной системы и передает управление ему. Все последующие действия по загрузке системы (монтирование корневой файловой системы, запуск программы инициализации) выполняет ядро Linux.

1.2.2. Ядро Linux и его параметры

Ядро — это святая святых операционной системы Linux. Ядро управляет всем: файловой системой, процессами, распределением памяти, устройствами и т. п. Если программе нужно выполнить какую-либо операцию, она обращается к ядру Linux. Например, если программа хочет прочитать данные из файла, то она сначала открывает файл, используя системный вызов `open()`, а затем читает данные из файла с помощью системного вызова `read()`. Для закрытия файла используется системный вызов `close()`. Конечно, на практике все выглядит сложнее, поскольку Linux — многопользовательская и многозадачная система. Это значит, что с системой могут работать одновременно несколько пользователей, и каждый из пользователей может запустить несколько процессов. Ясно, что программе нужно учитывать "поправку на совместный доступ", т. е. во время работы с файлом одного из пользователей программа должна установить блокировку доступа к этому файлу других пользователей. Впрочем, в такие нюансы мы здесь вникать не будем.

Итак, ядро — это программа, пусть и самая главная программа в Linux. Как и любой другой программе, ядру Linux можно передать параметры, влияющие на его работу. Передать параметры ядру Linux можно с помощью любого загрузчика Linux. При установке Linux, особенно если операционная система отказывается устанавливаться с параметрами по умолчанию, полезно передать ядру особые параметры. Например, на некоторых ноутбуках для установки Linux требуется передать ядру параметры `noauto` и `nomscia`. Первый параметр запрещает автоматическое определение устройств, а второй — проверку PCMCIA-карт.

На компьютере с процессором AMD64 ядро переходило в режим паники, и установить Linux было невозможно. При этом на экране красовалось следующее сообщение:

```
kernel panic - not syncing: IO-APIC + timer doesn't work! Boot with apic=debug
and send report. Then try booting with the 'noapic' option
```

Помог решить проблему параметр ядра `noapic`, позволяющий SMP-ядру не использовать расширенные возможности контроллера прерываний в многопроцессорных машинах. Обратите внимание: ядро само подсказало, чего ему не хватает! Но также ради справедливости нужно отметить, что указанная проблема характерна для первых версий ядра линейки 2.6.x. Новые версии нормально работают с процессорами AMD64.

Также мною была замечена еще одна проблема, относящаяся к современным дистрибутивам. Современные версии ядра Linux поддерживают механизм Enhanced Disk Device (EDD) polling, позволяющий собирать информацию обо всех дисковых устройствах, с которых возможна загрузка. Вся собранная информация потом сохраняется в каталоге `/sys`. Иногда возникает проблема с EDD, при загрузке Linux пользователь видит сообщение "Updating EDD...", и компьютер как бы зависает. В некоторых случаях загрузка продолжается секунд через 30—40, а в некоторых вообще не начинается. На проблему с EDD указывает тот факт, что при загрузке система "обнаруживает" лишние загрузочные устройства. В этом случае вам поможет параметр ядра `edd=skipmbr`. Если он не поможет решить проблему (длительная загрузка или лишние устройства), то попробуйте параметр `edd=off`, вообще отключающий механизм EDD.

Кроме передачи параметров ядру, при установке можно передать параметры программе установки — например, параметр `vga` при установке Linux Mandriva означает, что программа установки должна работать при разрешении 640×480, что позволяет запустить установку на самых "древних" компьютерах или когда видеокарта не полностью совместима с Linux (такое редко, но бывает). Передать параметры программе установки можно так же, как и параметры ядру.

В различных дистрибутивах редактирование параметров ядра, естественно, осуществляется по-разному. Так, в Fedora 13 нужно выбрать необходимый вариант установки (обычно выбирается первый, предлагающий установить или обновить существующую систему) и нажать клавишу `<Tab>` (рис. 1.1). В результате мы получим текстовую строку, в которой можно отредактировать параметры ядра (рис. 1.2).



Рис. 1.1. Начальное меню при установке Fedora 13

В Ubuntu 10 для редактирования параметров ядра нужно выбрать необходимый вариант установки и нажать клавишу <F6> (рис. 1.3).

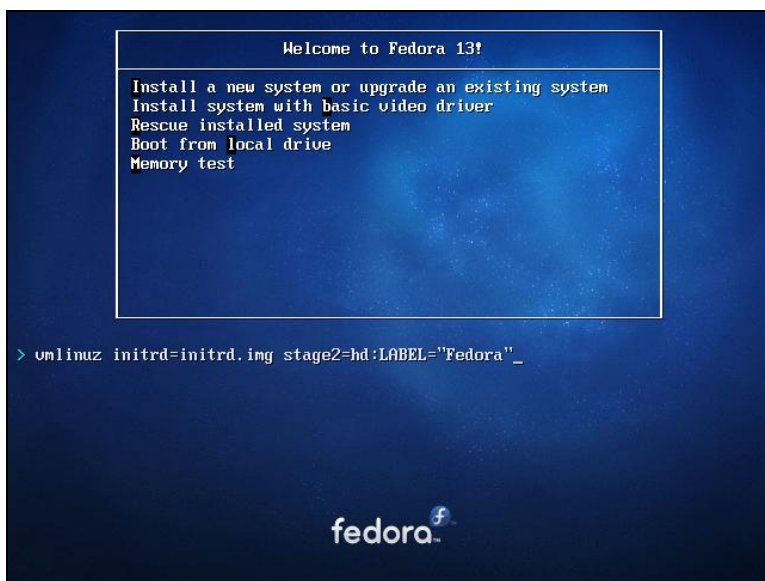


Рис. 1.2. Редактирование параметров ядра в Fedora 13

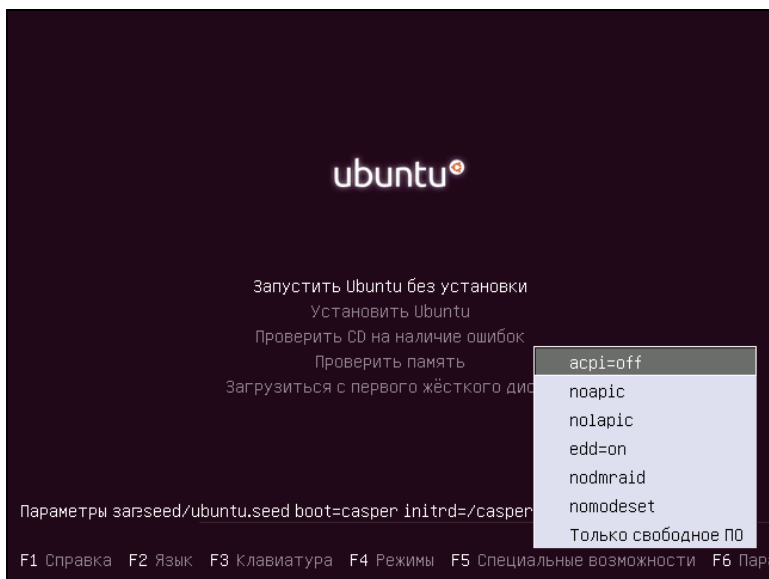


Рис. 1.3. Начальное меню при установке Ubuntu 10

В Mandriva 2010 для ввода параметров ядра нужно нажать <F6>, а потом ввести параметры ядра в поле **Опция ядра** (Boot options) (рис. 1.4).



Рис. 1.4. Начальное меню при установке Mandriva 2010.1 Spring

ПРИМЕЧАНИЕ

Обратите внимание на рис. 1.3 и 1.4: меню загрузчика GRUB русифицированное. Однако сразу после загрузки с DVD меню будет на английском языке. Для выбора языка нужно нажать клавишу <F2> и выбрать русский язык из списка. Такая возможность есть не у всех дистрибутивов. Например, в Fedora выбор языка возможен только после запуска программы установки.

Подробнее о параметрах ядра вы сможете прочитать в *приложении 2*. Здесь же, в табл. 1.1, представлены некоторые полезные параметры программы установки Fedora.

Таблица 1.1. Дополнительные параметры программы установки Fedora

Параметр	Описание
<code>linux noprobe</code>	Запретить исследования "железа" вашего компьютера. Очень полезно, например, на ноутбуках, когда не хочет правильно определяться та или иная PCMCIA-карта
<code>linux mediacheck</code>	Проверка носителя перед установкой. Бессмысленный параметр — ведь при установке программа все равно спросит вас, хотите ли вы проверить носитель
<code>linux rescue</code>	Запуск режима восстановления Linux
<code>linux askmethod</code>	Позволяет выбрать другой метод установки, например установку по сети

Таблица 1.1 (окончание)

Параметр	Описание
memtest86	Запускает программу memtest86, если у вас есть подозрение на неисправность оперативной памяти, что проявляется в непредсказуемых зависаниях и перезагрузках компьютера. Программа протестирует вашу оперативную память и сообщит о возможных ошибках
linux resolution=XxY	Устанавливает разрешение экрана для программы установки, например <code>linux resolution=1024x768</code>

В стандартных условиях ни один из этих параметров вводить не нужно — все и так пройдет успешно.

1.3. Проверка носителей

Некоторые дистрибутивы, в частности Fedora (и дистрибутивы, основанные на этом дистрибутиве), предлагают выполнить проверку установочного DVD-диска перед установкой (рис. 1.5).



Рис. 1.5. Проверка носителя

Если поверхность DVD вызывает у вас сомнения, можно его проверить — зачем тратить время на установку, если на 99-м проценте программа установки сообщит вам, что ей не удастся прочитать какой-то очень важный пакет, и система не может быть установлена? Если же DVD новый (только что купленный), можно отказаться от проверки носителя — вы сэкономите немного времени.

1.4. Изменение таблицы разделов

Система Linux не может быть установлена в Windows-разделы (FAT32, NTFS). Для ее установки нужно создать Linux-разделы (файловая система ext3). Понятно, что для этого на жестком диске должно быть неразмеченное пространство. Если его нет, то придется или удалить один из Windows-разделов и на его месте создать Linux-раздел, или же уменьшить размер одного из Windows-разделов и на освободившемся месте создать разделы Linux.

Понятно, что удалять раздел не хочется — ведь вы можете потерять данные. Поэтому обычно уменьшают размер Windows-раздела. Перед началом установки убедитесь, что в каком-либо разделе у вас есть 4—6 Гбайт свободного пространства (лучше не жадничать и рассчитывать на 6 Гбайт, даже если после установки системы останутся свободными несколько гигабайтов, они вам не мешают).

ПРИМЕЧАНИЕ

Если вы устанавливаете старый дистрибутив Linux, в котором все еще используется загрузчик LILO, то основной раздел Linux должен находиться ближе к началу диска. Linux может загружаться с разделов, которые начинаются до 1024-го цилиндра, т. е. первый блок раздела должен находиться до 1024-го цилиндра. Это не проблема самой операционной системы, это требования загрузчика Linux. В некоторых случаях эту проблему удастся обойти, а в некоторых — нет. Лучше лишний раз не тратить время зря и создать Linux-раздел так, чтобы он начинался как можно ближе к "началу" диска. После установки Linux сможет использовать (читать и записывать данные) любые разделы вне зависимости от начального номера цилиндра раздела.

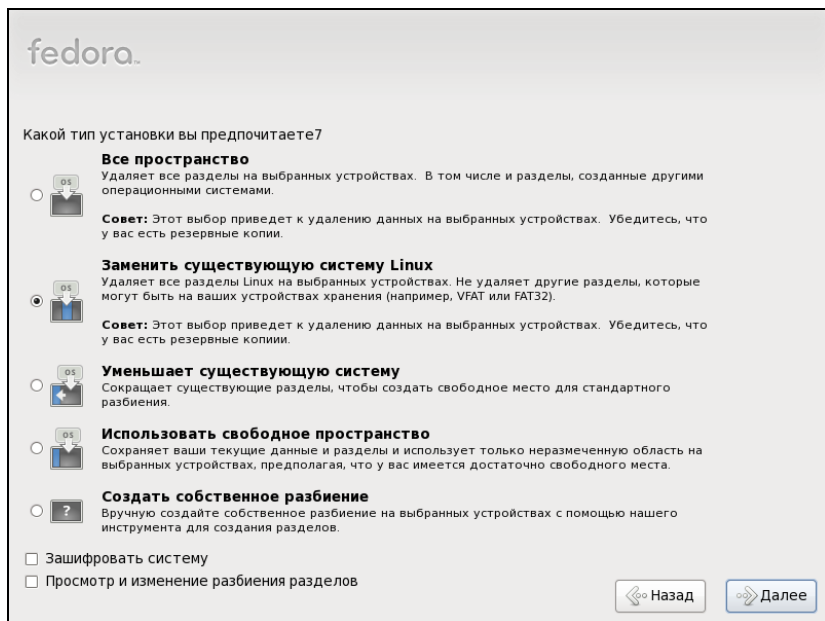


Рис. 1.6. Выбор типа разметки диска в Fedora 13

Перед установкой Linux нужно произвести дефрагментацию того Windows-раздела, который вы собрались уменьшать, чтобы упростить задачу программе установки по переносу ваших файлов.

В любом дистрибутиве программа установки системы Linux умеет автоматически разбивать жесткий диск — она сама создаст Linux-разделы без вашего участия. Например, в Fedora 13 вам доступны следующие варианты разметки диска (рис. 1.6):

- ❖ **Все пространство** — будет использован весь жесткий диск. Используйте этот вариант, если вы устанавливаете Linux на новый компьютер;
- ❖ **Заменить существующую систему Linux** — если на вашем компьютере уже была установлена операционная система Linux, то выбор этого варианта уничтожит эту систему, а на ее место будет установлена Fedora 13;
- ❖ **Уменьшает существующую систему** — существующая система будет сжата и в освободившееся после сжатия пространство будет установлена Fedora. По своим последствиям этот вариант непредсказуем. На своей системе я его не проверял и вам не советую. Если все-таки спортивный интерес победит здравый смысл, сделайте резервную копию всех важных данных перед выбором этого варианта;
- ❖ **Использовать свободное пространство** — инсталлятор будет использовать свободное, т. е. неразмеченное пространство для установки Linux. Этот вариант я протестировал и обнаружил, что он работает не так, как нужно. Система почему-то пытается использовать неразмеченное пространство, которое я зарезервировал для первичного раздела (куда я планировал установить FreeBSD), при этом она совсем не хочет видеть свободное дисковое пространство в расширенном разделе;
- ❖ **Создать собственное разбиение** — этот вариант подходит для пользователей, которые понимают, что делают и которым не все равно, что случится с их данными после установки Linux.

Mandriva предлагает два варианта: либо использовать весь жесткий диск (когда еще ни одна ОС не установлена) и ручную разметку, либо удалить Windows и ручную разметку. Каких-либо вариантов сосуществования двух ОС нет (Mandriva и Windows), поэтому если вы хотите сохранить свою Windows, то придется выбрать ручную разметку и собственноручно настроить разделы (рис. 1.7).

Лично я люблю контролировать процесс разметки (не без преувеличения скажу, что это один из самых важных процессов), поэтому всегда выбираю ручную разметку (рис. 1.8).

Если вы выбрали ручную разметку, тогда вам нужно изменить размер одного из существующих Windows-разделов и создать два Linux-раздела. Первый — корневой, его точка монтирования обозначается слэшем — /. Второй — раздел подкачки (тип swap).

ПРИМЕЧАНИЕ

Изменение размера Windows-раздела — довольно медленная операция. Помню, устанавливал Linux на компьютер, на котором был всего лишь один Windows-раздел размером в 38 Гбайт, 30 из которых было занято. Операция по изменению размера заняла

около получаса. Желательно на время изменения размера раздела исключить возможность отключения питания компьютера (т. е. подключить компьютер через мощный ИБП), а то результаты отключения питания во время этой операции предсказать сложно. Скорее всего, будут существенные потери данных.

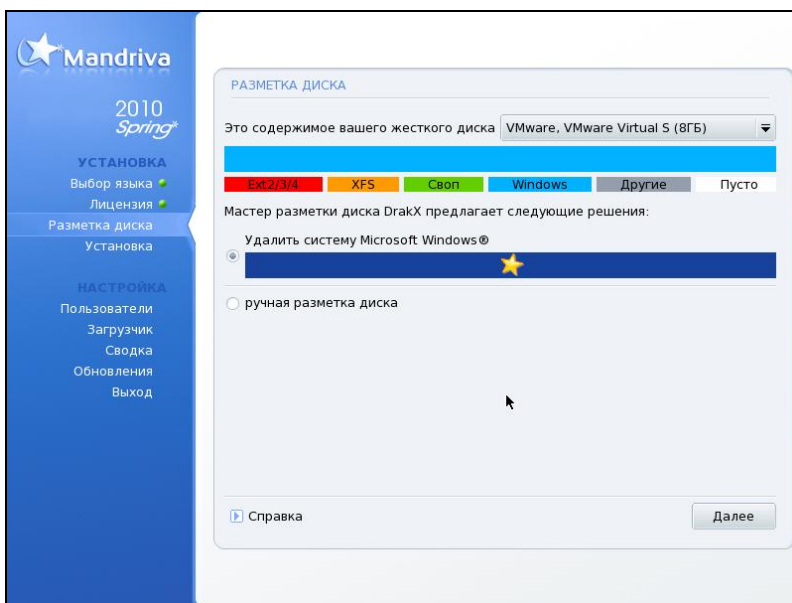


Рис. 1.7. Выбор типа разметки диска в Mandriva 2010.1 Spring

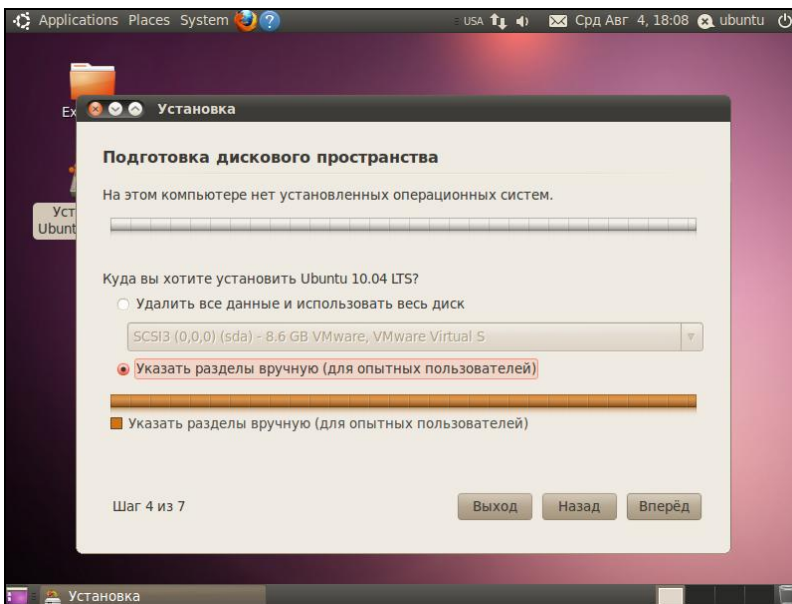


Рис. 1.8. Выбор типа разметки диска в Ubuntu 10

Размер раздела подкачки зависит от объема оперативной памяти:

- ❖ если ваш компьютер имеет менее 256 Мбайт оперативной памяти, то можно установить 512 Мбайт или больше для раздела подкачки;
- ❖ если у вас 256—1024 Мбайт, можно установить размер раздела подкачки в пределах 256—512 Мбайт;
- ❖ если у вас 1 Гбайт или более, можно вообще отказаться от раздела подкачки или установить чисто символический размер — 256 Мбайт. Даже если вам и не хватит виртуальной памяти (оперативной + подкачка), вы всегда сможете создать файл подкачки.

ПРИМЕЧАНИЕ

В Linux можно создать специальный файл подкачки, который тоже будет использоваться в процессе свопинга. Как правило, файл подкачки создается, если размера раздела подкачки оказалось недостаточно, а заново переразбивать жесткий диск (с целью увеличения размера раздела подкачки) не хочется.

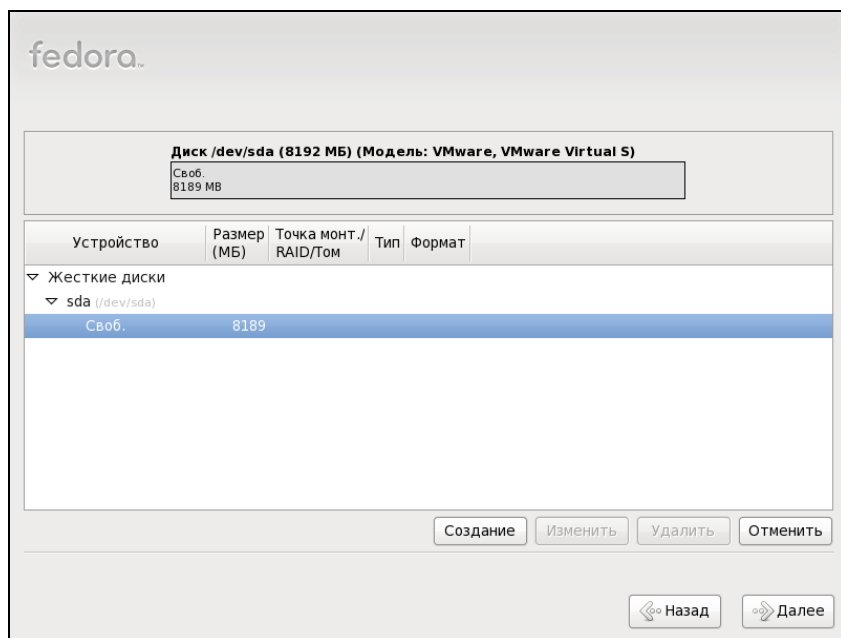


Рис. 1.9. Программа разметки диска в Fedora 13

Работать с программой разметки очень просто. Нужно выделить свободное пространство и нажать кнопку **Создать**. Далее нужно выбрать тип раздела (стандартный, программный RAID или том LVM) и установить параметры раздела (рис. 1.9), затем в открывшемся окне указать новый размер Windows-раздела. Для изменения параметров уже существующих разделов используется кнопка **Изменить**. Как получить свободное пространство, если жесткий диск уже размечен? Правильно, можно изменить его размер.

ПРИМЕЧАНИЕ

В Mandriva кнопка **Изменить** называется **Изменить размер**.

После освобождения места вы увидите, что на диаграмме диска появилось свободное (неразмеченное) место. Нужно его выделить, нажать кнопку **Создать**, в открывшемся окне (рис. 1.10) определить параметры создаваемого раздела (файловая система должна быть `ext3` или `ext4` — в зависимости от дистрибутива) и нажать кнопку **ОК** — этим вы создадите Linux-раздел.

ПРИМЕЧАНИЕ

Хотя инсталлятор Mandriva и не предлагает автоматического решения вопроса размещения Windows и Linux на одном жестком диске, нужно отметить, что сама программа работы с разделами (`diskdrake`) работает лучше и корректнее, нежели другие программы (в том числе и `grated`, которая так популярна в Debian и Ubuntu). Поэтому если вам нужна надежная программа разметки диска, можете смело использовать DVD с Mandriva (в режиме ручной разметки) вместо платных программ вроде Partition Magic. Во всяком случае, `diskdrake` наиболее корректно изменяет размер раздела, что немаловажно.

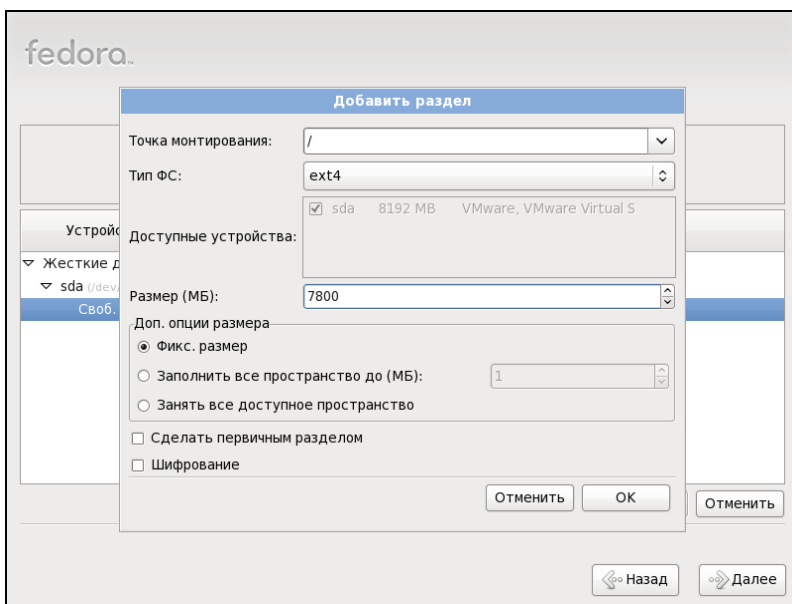


Рис. 1.10. Создание раздела в Fedora 13

ПРИМЕЧАНИЕ

Практически все современные дистрибутивы поддерживают шифрование файловой системы. При создании раздела вы можете включить шифрование (например, включить параметр **Зашифровать** или **Шифрование** (см. рис. 1.10) — в зависимости от дистрибутива он называется по-разному). Но нужно ли вам это? Если вы агент 007 — бесспорно — это очень полезная опция. А вот во всех остальных случаях в случае сбоя системы при попытке восстановления данных опция шифрования создаст только дополнительные проблемы.

Аналогично осуществляется разметка диска и в программе `diskdrake` из состава дистрибутива Mandriva 2010.1 Spring (рис. 1.11).

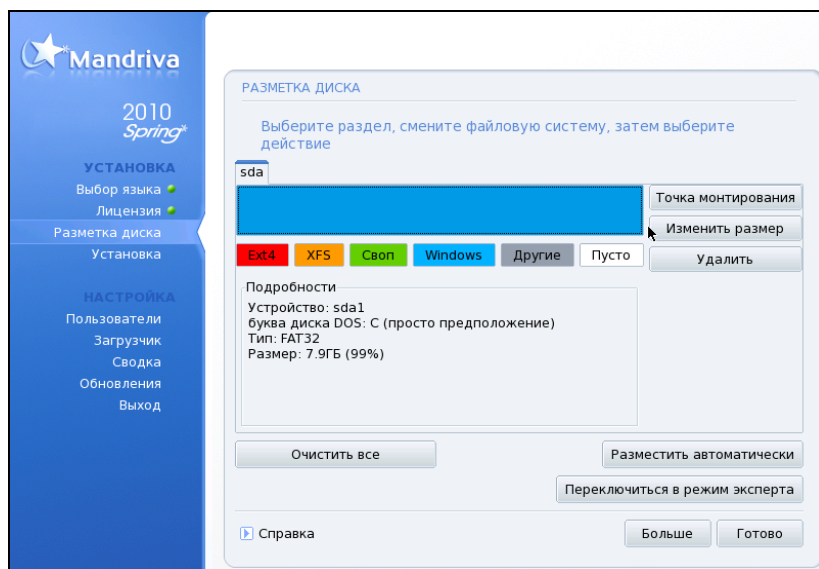


Рис. 1.11. `diskdrake` — программа разметки диска в Mandriva 2010

1.5. Выбор групп пакетов

Некоторые дистрибутивы, например Mandriva и Fedora, разрешают пользователю самому выбирать, какие группы пакетов нужно устанавливать, а какие — нет. Другие — например Ubuntu и его клоны, не имеют такой функции.

Если в вашем дистрибутиве можно выбирать пакеты самому, главное, о чем нужно заботиться, — это дисковое пространство. У меня как-то раз произошла анекдотическая ситуация: программа установки установила почти все пакеты, а потом лишь сообщила, что не хватает места на диске, и предложила... перезагрузку.

Серьезная недоработка программы установки Fedora (я уже номер версии даже не указываю — видимо это "фирменная особенность" дистрибутива) — она не сообщает полный объем выбранных пакетов (рис. 1.12). В других дистрибутивах (openSUSE, Mandriva) с этим проще — ведь вы видите, сколько у вас доступно места на диске и какой объем пакетов вы выбрали (рис. 1.13).

1.6. Выбор графической среды

При установке Linux часто существует возможность выбрать графическую среду — GNOME или KDE (есть также возможность установить обе, если имеется

достаточно дискового пространства). В Windows мы привыкли к тому, что у нас один-единственный графический интерфейс.

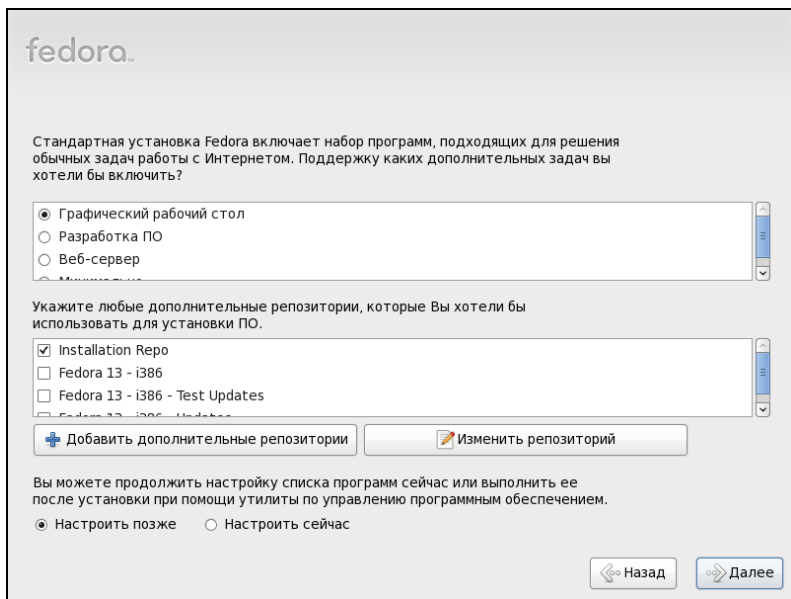


Рис. 1.12. Выбор групп пакетов в Fedora 13: сколько места на диске займет система неизвестно

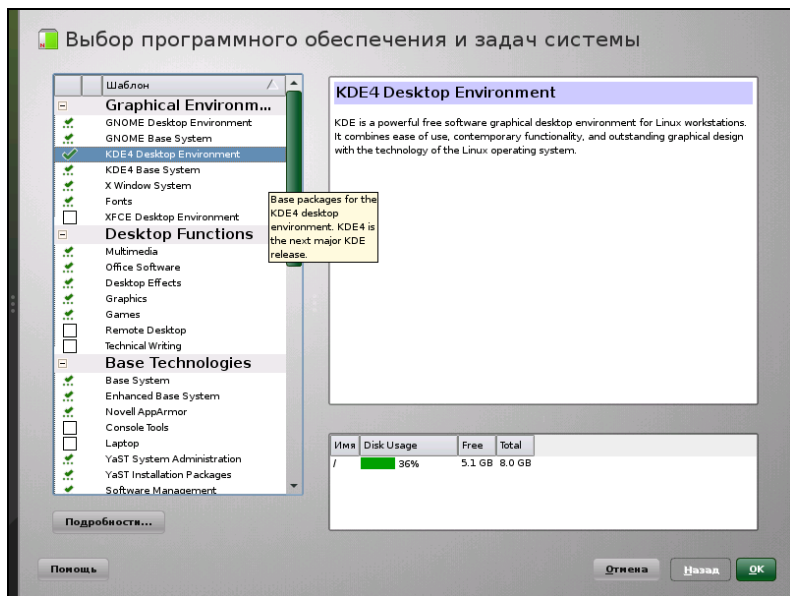


Рис. 1.13. Выбор групп пакетов в openSUSE 11.2: система займет 5,1 Гбайт дискового пространства (выбраны и KDE, и GNOME)

Мы можем менять графическую тему, изменять настройки отдельных графических элементов, но чтобы мы ни делали (установку программ вроде Talisman не учитываем — это от лукавого!), графический интерфейс пользователя останется тем же.

В Linux все немного иначе. Есть графическая подсистема — сервер X (X.Org), который предоставляет фундамент для построения графического интерфейса. А вот построением самого интерфейса пользователя занимаются графические среды, т. е. графическая среда определяет, как будет выглядеть интерфейс пользователя.

Первой графической средой для Linux, способной тягаться по комфорту использования с графическим интерфейсом Windows, стала KDE (1996 год). В 1999 году появилась графическая среда GNOME. С тех пор они конкурируют между собой.

Назначая графическую среду, вы выбираете не только, как станет выглядеть интерфейс пользователя, — вы определяете набор программ, с которыми будете работать. Дело в том, что среда KDE использует библиотеку Qt, а в основе GNOME лежит библиотека GTK. Следовательно, если вы выбрали KDE, то будут установлены программы, которые основаны на этой библиотеке. Если же вы выберете GNOME, то будут установлены приложения, основанные на GTK. Простейший пример: в качестве файлового менеджера при выборе KDE будет установлена программа Dolphin, а если выбрать GNOME, то — Nautilus.

Какую графическую среду выбрать? Раньше я советовал выбирать KDE, потому что эта графическая среда была лучше русифицирована и более удобна в использовании для бывших Windows-пользователей. Сейчас у GNOME нет никаких проблем с русским языком, и в то же время GNOME так же удобна, как и KDE. Во всяком случае, в последний год я использую GNOME.

Текущая версия KDE — KDE4, KDE3 уже окончательно удалена из состава некоторых дистрибутивов как устаревшая среда. Хотя некоторые дистрибутивы все еще позволяют выбрать версию KDE. Текущая версия GNOME — 2.29, но уже не за горами третья версия — GNOME3.

1.7. Установка пароля root

Пользователь root — это главный пользователь в системе (как Администратор в Windows). Постарайтесь не забыть его пароль (рис. 1.14)! В некоторых дистрибутивах окно для ввода пароля root совмещено с окном добавления пользователя (например, в Mandriva), некоторые дистрибутивы выводят отдельно окно для задания пароля root (Fedora), а openSUSE предлагает создать обычного пользователя, а его пароль будет использоваться в качестве пароля root (рис. 1.15). Это довольно удобно, но с точки зрения безопасности лучше, чтобы пароль root не совпадал с пользовательским паролем.

ПРИМЕЧАНИЕ

В Fedora 13 при установке сразу же активируется русская раскладка (или другая — какую вы выберете), поэтому при вводе имени и пароля могут возникнуть проблемы — переключу-

читься на английскую раскладку нельзя, потому что ее вообще не существует. Поэтому либо нужно выбрать английскую раскладку (при установке), либо вводить пароль цифрами.

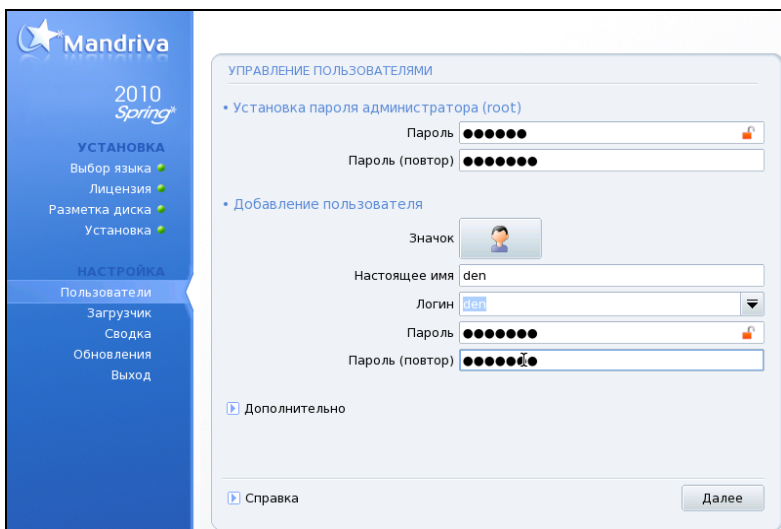


Рис. 1.14. Установка пароля root (Mandriva 2010)

1.8. Создание учетных записей пользователей

При установке системы вам нужно создать хотя бы одну пользовательскую учетную запись — ее вы будете использовать для входа в систему. Многие современные дистрибутивы запрещают вход в систему от имени root, поэтому вы будете использовать именно созданную при установке учетную запись пользователя.

Кстати, openSUSE, Mandriva и Ubuntu предлагают создать учетную запись во время установки ОС (рис. 1.15), а Fedora — при первом запуске (рис. 1.16).

ПРИМЕЧАНИЕ

На страничке <http://www.dkws.org.ua/index.php?page=show&file=video-lessons/index> вы найдете несколько полезных видеоуроков, в том числе и урок по установке Fedora 13.

1.9. Порядок установки операционных систем

Как уже отмечалось ранее, сначала нужно устанавливать Windows, а уже затем — Linux. Дело в том, что при установке Windows узурпирует главную загрузочную запись, и после ее установки Linux вы уже не запустите.

The screenshot shows the 'Создать нового пользователя' (Create new user) window in openSUSE. It contains the following fields and options:

- Full name: den
- Username: den
- Password: masked with dots
- Confirm password: masked with dots
- Checkboxes:
 - ☒ Use this password for the system administrator
 - ☐ Receive system messages
 - ☒ Automatic login to the system
- Summary section:
 - Authentication method: local /etc/passwd
 - Password encryption method: Blowfish
- Buttons: 'Изменить...' (Change...), 'Помощь' (Help), 'Прервать' (Cancel), 'Назад' (Back), and 'Далее' (Next).

Рис. 1.15. Создание пользователя в openSUSE (при установке системы)

The screenshot shows the 'Пользователь' (User) screen in Fedora. It includes a sidebar on the left with navigation links: 'Добро пожаловать' (Welcome), 'Информация о лицензии' (License information), 'Пользователь' (User), 'Дата и время' (Date and time), 'Профиль оборудования' (Hardware profile), and 'Оборудован' (Equipped). The main area contains:

- Title: 'Пользователь' (User)
- Text: 'Требуется создать пользователя для повседневного (не административного) использования системы. Для этого введите необходимые данные.' (A user must be created for everyday (non-administrative) use of the system. Enter the necessary data.)
- Fields:
 - Username: den
 - Full name: empty
 - Password: masked with dots
 - Repeat password: masked with dots
- Text: 'Если требуется использовать проверку подлинности по сети, например Kerberos или NIS, нажмите кнопку «Сетевая аутентификация».' (If you need to use network authentication, such as Kerberos or NIS, click the 'Network authentication' button.)
- Button: 'Сетевая аутентификация...' (Network authentication...)
- Buttons at the bottom: 'Назад' (Back) and 'Вперёд' (Forward).

Рис. 1.16. Создание пользователя в Fedora (при первом запуске системы)

При установке Linux такого не происходит — загрузчик Linux настраивается так, чтобы вы могли запускать как Linux, так и Windows.

Если вы планируете установить несколько версий Windows, например, XP и Windows 7, то сначала установите все необходимые вам версии Windows, а потом установите Linux.

1.10. Проблемы при установке

1.10.1. Ошибка: *kernel panic: VFS: Unable to mount root fs*

Появление такого сообщения означает, что ядро не может подмонтировать корневую файловую систему. Понятно, что дальнейшее продолжение работы невозможно. Наиболее вероятная причина — повреждение установочного диска. Если с поверхностью диска все в порядке (она не поцарапана, отсутствуют следы грязи и/или жира), тогда причина в ошибке при записи DVD. Выход один — раздобыть другой DVD и загрузиться с него.

1.10.2. Проблемы с некоторыми LCD-мониторами

Если ваш LCD-монитор подключен к DVI-разъему видеокарты и с ним возникают проблемы (не поддерживается максимальное разрешение, низкое качество изображения, самопроизвольное выключение питания монитора), попробуйте передать ядру параметр `nofb`. Если это поможет решить проблему, "пропишите" данный параметр в конфигурационном файле загрузчика (об этом мы также поговорим далее).

Что делать, если параметр `nofb` не помог? Просто подключите монитор к аналоговому разъему видеокарты — все должно заработать нормально.

1.10.3. Список известных проблем в Mandriva Linux 2010

Вы можете ознакомиться со списком известных проблем, обнаруженных в Mandriva 2010 (для версий 2010.0 и 2010.1): возможно, ваша проблема уже решена:

http://wiki.mandriva.com/ru/2010.0_Errata

http://wiki.mandriva.com/ru/2010.1_Errata

Поскольку этот список постоянно обновляется, не вижу смысла приводить его в книге — вы всегда сможете прочитать его обновленную версию в Интернете.

1.10.4. Не переключается раскладка в Fedora 13

После входа в свежее установленную систему вам наверняка захочется запустить терминал и ввести пару команд. Но активна только русская раскладка (или только та, которую вы выбрали при установке). Какие бы комбинации клавиш вы ни на-

жимали (<Ctrl>+<Shift>, <Alt>+<Shift>, <Shift>+<Shift> и др.) — ничего не помогает. Оказывается, в пользовательский профиль устанавливается только выбранная раскладка. Чтобы вводить латиницу, нужно добавить соответствующую раскладку. Выполните команду **Система | Параметры | Клавиатура** и на вкладке **Раскладки** добавьте раскладку **Соединенные штаты/США**.

1.10.5. Установка Linux на HP Mini 2133 (проблема с ACPI)

При установке Linux на этот нетбук может возникнуть проблема с ACPI (Advanced Configuration and Power Interface). В этом случае нужно или отключить ACPI (параметр ядра `acpi=off`), или обновить BIOS нетбука до версии F.06. Отключение ACPI на нетбуке можно воспринимать только как временную меру — пока не обновите BIOS. Отключить интерфейс управления питанием на нетбуке — это все равно, что не включать кондиционер в жару (при условии наличия самого кондиционера!). Пока вы не обновите BIOS (при отсутствии опыта эту операцию лучше производить в сервисном центре), можно выключить ACPI. А после обновления BIOS ваша система сможет работать нормально.

1.10.6. Проблема с ACPI на Fujitsu Siemens esprime mobile u9200

Проблема с ACPI есть еще у одного ноутбука. На ноутбуке esprime mobile u9200 неправильно работает подсветка. Чтобы все 8 уровней подсветки были доступны, вам нужно передать ядру параметр `acpi_backlight=vendor`. Понятно, что этот параметр первый раз нужно просто передать ядру, дабы проверить, правильно ли работает подсветка, а затем его нужно добавить в конфигурационный файл загрузчика, чтобы не вводить его каждый раз при запуске Linux.

1.10.7. Писк при выключении или перезагрузке компьютера в Mandriva

Собственно, писк при перезагрузке компьютера — это не проблема, но иногда он раздражает, особенно на домашнем компьютере, когда ночью он может разбудить окружающих. Mandriva настраивает встроенный динамик компьютера чуть ли не на максимальную громкость. Чтобы избавиться от писка, нужно добавить в файл `/etc/modprobe.d/disable-pcspkr.conf` (если он не существует, его нужно создать) строку:

```
blacklist pcspkr
```

1.10.8. Mandriva One не запускается на компьютерах с видеокартой nvidia

На некоторых компьютерах Mandriva One не запускается с собственным драйвером от nvidia, входящим в состав Mandriva One. Чтобы исправить данную проблему, нужно передать ядру параметр `xdriver=free`. Можно также войти как root и ввести следующие команды:

```
rmmod nvidia  
modprobe nvidia  
/etc/init.d/dm restart
```

После этого графическая система X.Org запустится без ошибок.

ГЛАВА 2



Вход в систему

2.1. Графический и текстовый вход в систему. Завершение работы

Как только система загрузится, вы увидите окно графического менеджера регистрации (рис. 2.1).

Здесь все просто: нужно сначала ввести имя пользователя, созданного при установке системы (потому что явно пользователей мы еще не добавляли), и его пароль.

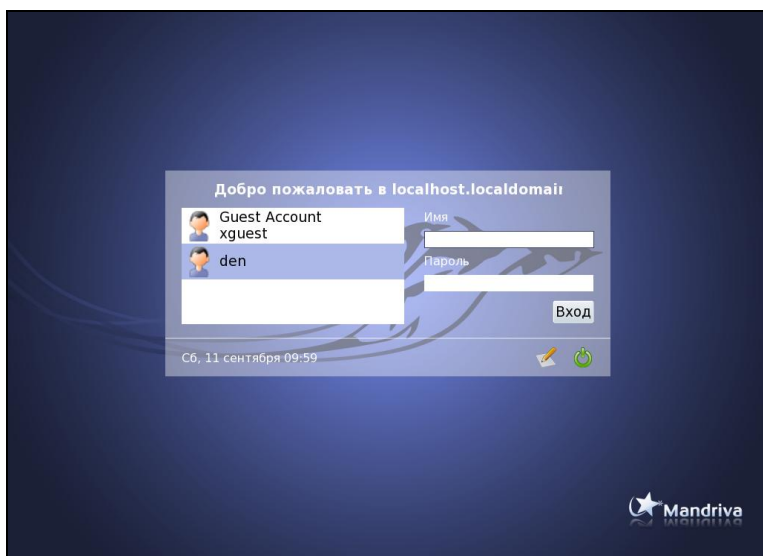


Рис. 2.1. Графический вход в систему

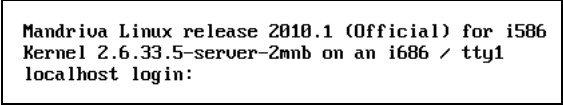
Обычно при входе в систему вы можете выбрать язык графического интерфейса и установить дополнительные параметры, например, выбрать графическую среду. Также можно выключить компьютер, если вы передумали входить в систему.

На рис. 2.1 изображен графический менеджер регистрации из дистрибутива 2010,1. Под кнопкой **Вход** находятся две графические кнопки: первая позволяет выбрать графическую среду, а вторая — выключить компьютер.

В Fedora 13 менеджер регистрации подобен этому: тоже есть возможность выбора сеанса и выключения компьютера, а также возможность выбора языка. А вот в Ubuntu выбрать сеанс вы не можете, поскольку в системе установлена только одна графическая среда — GNOME.

Теперь немного поговорим о графическом интерфейсе. По умолчанию современные дистрибутивы используют пятый уровень запуска (об этом мы поговорим в *главе 17*). Это означает, что будет загружен графический интерфейс, соответственно, и регистрация в системе будет выполнена средствами графического менеджера регистрации. Но графический интерфейс на сервере — непозволительная роскошь. Почему? Ведь вы же не будете работать на сервере постоянно, как на обычной рабочей станции. Вы его один раз настроите, затем будете работать на обычной рабочей станции, а сервер будет себе спокойно стоять в углу вашего кабинета. Выходит, что загруженный графический интерфейс только даром занимает оперативную память (а занимает он действительно много памяти) и отнимает драгоценное процессорное время. Многие администраторы используют третий уровень запуска — это нормальный многопользовательский режим с поддержкой сети, но без загрузки графического интерфейса. Такой режим позволяет сэкономить системные ресурсы сервера. О том, как выбрать третий уровень запуска, мы поговорим в *главе 17*.

Если система загрузилась на третьем уровне запуска, то при входе вы увидите текстовое приглашение ввести логин и пароль (рис. 2.2).



```
Mandriva Linux release 2010.1 (Official) for i586
Kernel 2.6.33.5-server-2mnb on an i686 / tty1
localhost login:
```

Рис. 2.2. Текстовый вход в систему

При вводе логин отображается на экране, а пароль — нет (даже не отображаются "звездочки"), поэтому будьте внимательны.

Поработав в консоли, для завершения работы вы можете использовать следующие команды:

- ◆ `exit` (или `logout`) — завершение сеанса без завершения работы системы;
- ◆ `poweroff` — завершение работы системы с выключением питания;
- ◆ `reboot` — перезагрузка.

Самой "продвинутой" командой является команда `shutdown` — она позволяет завершить работу системы, перезагрузить систему, указать время завершения работы и причину останова системы. Предположим, что вы хотите уйти пораньше, но компьютер нужно выключить в 19:30 — вдруг некоторые пользователи задержались на работе, а вы выключите сервер — некрасиво получится. Вам поможет `shutdown`:

```
# shutdown -h 19:30 [сообщение]
```

Сообщение можно и не указывать — все равно Windows-пользователи его не увидят. Если нужно завершить работу системы прямо сейчас, вместо времени укажите `now`:

```
# shutdown -h now
```

Для перезагрузки системы используется опция `-r`:

```
# shutdown -r now
```

ПРИМЕЧАНИЕ

Здесь и далее, если приглашение командной строки выглядит как `#`, это означает, что команда должна быть введена с привилегиями `root` (т. е. после ввода команды `su`) или же от имени пользователя `root` (когда вы вошли в систему как `root`), что, собственно, одно и то же.

2.2. Переключение в консоль из графического интерфейса

Если вы хотите стать настоящим линуксоидом, тогда вы просто обязаны знать, как работать в консоли. Для переключения в консоль нажмите комбинацию клавиш `<Ctrl>+<Alt>+<F1>` — вы попадаете на первый виртуальный терминал (на первую консоль). Для возврата нажмите `<Alt>+<F7>`. Подробно о работе в консоли мы поговорим в *главе 19*, поэтому сейчас не будем останавливаться на этом подробно.

2.3. Вход в систему как root

Пользователь `root` обладает максимальными полномочиями: одна его команда — и система уничтожит сама себя, поэтому в современных дистрибутивах разработчики стараются защитить систему от непродуманных действий пользователя. Причем в каждом дистрибутиве защита от `root` реализована различными способами, поэтому нет одного общего для всех способа, позволяющего войти в систему как `root`, когда это нужно.

Нужно отметить, что из соображений безопасности действительно не рекомендуется работать в системе как `root`. Рекомендуется все операции выполнять от имени обычного пользователя, а действия, требующие прав `root` — через команды `su` и `sudo`, которые мы рассмотрим в *главе 14*. В этой же главе мы поговорим о том, как войти в систему в качестве пользователя `root`. Однако сразу вынужден вас предупредить: все действия, выполняемые от имени `root`, вы выполняете на свой страх и риск — я не виноват, если в результате непродуманной команды вы разрушите всю систему.

ГЛАВА 3



Решение проблем при установке и после нее

3.1. Небольшие проблемы с LCD-мониторами

Если ваш LCD-монитор подключен к DVI-разъему видеокарты и с ним возникают проблемы (не поддерживается максимальное разрешение, проблемы с качеством изображения, самопроизвольное выключение питания монитора), попробуйте передать ядру параметр `nofb`. Если это поможет решить проблему, "пропишите" данный параметр в конфигурационном файле загрузчика (*см. главу 16*).

Что делать, если параметр `nofb` не помог? Просто подключите монитор к аналоговому разъему видеокарты — все должно работать нормально.

3.2. Проблема с показом времени в Ubuntu

По умолчанию Ubuntu считает, что системные часы компьютера установлены по UTC. Но обычно это не так, и системные часы компьютера установлены по локальному времени. В результате при загрузке Ubuntu происходит сдвиг времени (обычно на 2—3 часа, но это зависит от вашего часового пояса).

Чтобы изменить поведение Ubuntu, откройте файл конфигурации `/etc/default/rcS`:

```
gksu gedit /etc/default/rcS
```

Найдите в нем строку:

```
UTC=yes
```

Замените ее строкой:

```
UTC=no
```

Сохраните изменения, перезагрузите компьютер и заново установите время.

3.3. Зависание графического интерфейса в процессе работы

Перезапустить графический интерфейс можно с помощью комбинации клавиш `<Ctrl>+<Alt>+`. После этого, как правило, все будет нормально. Если же проблема повторится, нужно заново перенастроить графический интерфейс, выбрав другой модуль видеокарты.

3.4. Медленная загрузка системы

В большинстве случаев медленная загрузка системы — это следствие многочисленных служб, запускаемых автоматически при старте системы. Большая часть этих служб вам не нужна, поэтому их можно смело отключить. Отключение неиспользуемых служб не только повысит производительность системы, но и сделает ее более защищенной. Подробно об отключении служб мы поговорим в *главе 18*.

Кроме сервисов, тормозить запуск системы может проверка файловых систем. Наверняка в вашем компьютере есть несколько Windows-разделов (файловые системы VFAT и NTFS). При загрузке время от времени производится проверка всех файловых систем, а это, во-первых, замедляет запуск Linux, а, во-вторых, никому не нужно — пусть Windows сама проверяет свои разделы. Поэтому откройте файл `/etc/fstab` в любом текстовом редакторе. Обратите внимание на последнее поле — обычно оно содержит 1:

```
/dev/sda6 /media/sda8 vfat defaults,utf8,umask=007,gid=46 0 1
```

Измените значение последнего поля на 0 для всех Windows-разделов — так вы сможете немного ускорить запуск Linux.

3.5. Забыт пароль root

Восстановить пароль root довольно просто. Для этого достаточно загрузиться в однопользовательском режиме (при перезагрузке компьютера передать ядру Linux параметр `single`) и изменить пароль root (команда `passwd root`). Как правило, при перезагрузке в однопользовательском режиме пароль не запрашивается, но не всегда. Тут все зависит от настроек вашей системы. Ваша система может быть настроена так, что даже при перезагрузке в однопользовательском режиме будет запрошен пароль. Поэтому сейчас, пока вы еще помните пароль root, вам нужно выбрать между своим комфортом и безопасностью системы. Ведь в ваше отсутствие кто угодно может перезагрузить компьютер и, загрузившись в однопользовательском режиме, изменить ваш пароль. Если вы хотите, чтобы ваша система при

перезагрузке в single-режиме запрашивала пароль, добавьте в ваш файл /etc/inittab следующие строки:

```
ls:S:wait:/etc/rc.d/init.d/rc S
~~:S:respawn:/sbin/sulogin
```

```
plymouthd: ply-keyboard.c:384: ply_keyboard_watch_for_input: Assertion 'keyboard
!= ((void *)0)' failed.
sd 2:0:0:0: [sdal] Assuming drive cache: write through
sd 2:0:0:0: [sdal] Assuming drive cache: write through
sd 2:0:0:0: [sdal] Assuming drive cache: write through
bash: cannot set terminal process group (-1): Inappropriate ioctl for device
bash: no job control in this shell
bash-4.1# _
```

Рис. 3.1. Консоль после передачи параметра `init=/bin/bash` (Fedora 13)

```
mptbase: ioc0: Initiating bringup
ioc0: LSI53C1030 B0: Capabilities={Initiator}
scsi0 : ioc0: LSI53C1030 B0, FwRev=00000000h, Ports=1, MaxQ=128, IRQ=17
scsi 0:0:0:0: Direct-Access VMware, VMware Virtual S 1.0 PQ: 0 ANSI: 2
scsi target0:0:0: Beginning Domain Validation
scsi target0:0:0: Domain Validation skipping write tests
scsi target0:0:0: Ending Domain Validation
scsi target0:0:0: FAST-40 WIDE SCSI 80.0 MB/s ST (25 ns, offset 127)
sd 0:0:0:0: [sdal] 16777216 512-byte logical blocks: (8.58 GB/8.00 GiB)
sd 0:0:0:0: [sdal] Write Protect is off
sd 0:0:0:0: [sdal] Cache data unavailable
sd 0:0:0:0: [sdal] Assuming drive cache: write through
sd 0:0:0:0: [sdal] Cache data unavailable
sd 0:0:0:0: [sdal] Assuming drive cache: write through
sda: sda1 sda2 < sda5 sda6 >
sd 0:0:0:0: [sdal] Cache data unavailable
sd 0:0:0:0: [sdal] Assuming drive cache: write through
sd 0:0:0:0: [sdal] Attached SCSI disk
Loading ata_piix module
scsil : ata_piix
scsi2 : ata_piix
ata1: PATA max UDMA/33 cmd 0x1f0 ctl 0x3f6 bmdma 0x10c0 irq 14
ata2: PATA max UDMA/33 cmd 0x170 ctl 0x376 bmdma 0x10c0 irq 15
ata2.00: ATAPI: VMware Virtual IDE CDR0M Drive, 00000001, max UDMA/33
ata2.00: configured for UDMA/33
scsi 2:0:0:0: CD-ROM NECUMWar VMware IDE CDR10 1.00 PQ: 0 ANSI: 5
Loading ahci module
waiting for device sda5 to appear (timeout 1min)
waiting for device sda6 to appear (timeout 1min)
Creating root device.
Mounting root filesystem.
EXT4-fs (sda5): mounted filesystem with ordered data mode
Setting up other filesystems.
Switching to new root and running init.
bash: cannot set terminal process group (-1): Inappropriate ioctl for device
bash: no job control in this shell
bash-4.1#
```

Рис. 3.2. Консоль после передачи параметра `init=/bin/bash` (Mandriva 2010.1)

А если комфорт для вас на первом месте (или это ваш домашний компьютер, за которым работаете только вы), наоборот, прокомментируйте данные строки, если они есть в `/etc/inittab`.

Если войти в систему с помощью параметра `single` не получается, тогда передайте ядру параметры `rw init=/bin/bash`. Будут загружены ядро и командный интерпретатор. Все, что вам нужно будет сделать, — это ввести команду `passwd root` для изменения пароля `root`.

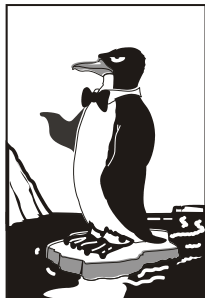
Ради интереса я передал Fedora 13 параметр `init=/bin/bash`. В результате увидел "картину", изображенную на рис. 3.1. Как видите, система никак не воспрепятствовала действиям "злоумышленника". Та же ситуация и с Mandriva 2010.1 (рис. 3.2). Остальные дистрибутивы проверять было лень. По умолчанию ваша система открыта для "локального" злоумышленника как двери в квартиру без замка. Оставлять систему так это все равно, что выйти из квартиры в магазин на полчаса и не закрыть дверь, наивно полагая, что в нее никто не войдет. Единственный способ бороться с таким поведением системы — это запретить редактировать меню GRUB/GRUB2. Тогда, если злоумышленник захочет изменить параметры ядра, загрузчик попросит его ввести пароль.



ЧАСТЬ II

ФАЙЛОВАЯ СИСТЕМА

ГЛАВА 4



Поддерживаемые файловые системы

4.1. Какие файловые системы поддерживает Linux?

Linux поддерживает много файловых систем. Иногда пользователь просто теряется, когда видит такое многообразие выбора — в качестве корневой файловой системы можно выбрать следующие файловые системы: ext2, ext3, XFS, ReiserFS, JFS.

ПРИМЕЧАНИЕ

О корневой файловой системе мы поговорим в следующей главе.

4.1.1. Файловые системы ext2, ext3 и ext4

"Родной" файловой системой Linux является журналируемая файловая система ext4, но вы можете выбрать предыдущую версию — ext3 или даже ext2. Linux также до сих пор поддерживает файловую систему ext — самую первую свою файловую систему, но при установке системы вы не можете выбрать ext. Поддержка ext добавлена в ядро только на тот случай, если вам попадетесь носитель информации, отформатированный в этой файловой системе.

Основное отличие ext3 от ext2 — наличие журнала, что делает файловую систему менее чувствительной к всевозможным сбоям. Новая версия — ext4 построена на базе ext3, но изменений гораздо больше, чем просто наличие журнала, поэтому о них мы поговорим отдельно.

Таким образом, в качестве корневой файловой системы и файловой системы других Linux-разделов используются файловые системы ext3, ext4, XFS, ReiserFS, JFS. Все перечисленные файловые системы (кроме ext2) ведут журналы своей работы, что позволяет восстановить данные в случае сбоя. Осуществляется это сле-

дующим образом — перед тем как выполнить операцию, *журналируемая* файловая система записывает эту операцию в журнал, а после выполнения операции удаляет запись из журнала. Представим, что после занесения операции в журнал произошел сбой (например, выключили свет). Позже, когда сбой будет устранен, файловая система по журналу выполнит все действия, которые в него занесены. Конечно, и это не всегда позволяет уберечься от последствий сбоя — стопроцентной гарантии никто не дает, но все же такая схема работы лучше, чем вообще ничего.

4.1.2. Файловые системы XFS, ReiserFS и JFS

Файловые системы ext2 и ext3 совместимы. По сути, ext3 — та же ext2, только с журналом. Раздел ext3 могут читать программы (например, Total Commander и Ext2Fsd в Windows), рассчитанные на ext2. В современных дистрибутивах по умолчанию задана файловая система ext3. При необходимости можно выбрать другие файловые системы. Далее мы рассмотрим их особенности, чтобы понять, нужно ли их использовать или же остановить свой выбор на стандартной ext3.

❖ **Файловая система XFS** была разработана компанией Silicon Graphics в 2001 году. Основная особенность данной системы — высокая производительность (до 7 Гбайт/с). XFS может работать с блоками размером от 512 байтов до 64 Кбайт. Ясно, что если у вас много маленьких файлов, то в целях экономии места можно установить самый маленький размер блока. А если вы работаете с файлами большого размера (например, мультимедиа), то нужно выбрать самый большой размер блока — так файловая система обеспечит максимальную производительность (конечно, если "железо" позволяет). Учитывая высокую производительность этой файловой системы, ее нет смысла устанавливать на домашнем компьютере, поскольку все ее преимущества будут сведены на нет. А вот если вы будете работать с файлами очень большого размера, XFS проявит себя с лучшей стороны.

❖ **Файловая система ReiserFS** считается самой экономной, поскольку позволяет хранить несколько файлов в одном блоке (другие файловые системы могут хранить в одном блоке только один файл или одну его часть). Например, если размер блока равен 4 Кбайт, а файл занимает всего 512 байт (а таких файлов очень много в разных каталогах), то 3,5 Кбайт просто не будут использоваться. А вот ReiserFS позволяет задействовать буквально каждый байт вашего жесткого диска!

Но у этой файловой системы есть два больших недостатка: она неустойчива к сбоям, и ее производительность сильно снижается при фрагментации. Поэтому, если вы выбираете данную файловую систему, покупайте UPS (источник бесперебойного питания) и почаще дефрагментируйте жесткий диск.

❖ **Файловая система JFS** (разработка IBM) сначала появилась в операционной системе AIX, а потом была модифицирована под Linux. Основные достоинства этой файловой системы — надежность и высокая производительность (выше, чем у XFS). Но у нее маленький размер блока (от 512 байтов до 4 Кбайт). Следо-

вательно, она хороша на сервере баз данных, но не при работе с данными мультимедиа, поскольку блок в 4 Кбайт для работы, например, с видео в реальном времени, будет маловат.

4.1.3. Особенности файловой системы ext4

Файловая система ext4 заслуживает отдельного разговора. Все, что было сказано о файловых системах ранее, справедливо и для ext4, но у новой файловой системы есть ряд особенностей, о которых мы сейчас и поговорим.

Поддержка ext4 как стабильной файловой системы появилась в ядре Linux версии 2.6.28. Если сравнивать эту файловую систему с ext3, то производительность и надежность новой файловой системы существенно увеличена, а максимальный размер раздела доведен до 1024 петабайт (1 эксбибайт). Максимальный размер файла — более 2 Тбайт. Ресурс Phoronix (www.phoronix.com) произвел тестирование новой файловой системы на SSD-накопителе (такие накопители устанавливаются на современные нетбуки) — результат, как говорится, налицо: ext4 почти в два раза превзошла файловые системы ext3, XFS, JFS и ReiserFS.

Впрочем, когда я установил Fedora 11 на рабочую станцию, прироста производительности при работе с файлами мне почувствовать не удалось. Однако производительность — это не основной конек ext4. Но обо всем по порядку.

Сравнение ext3 и ext4

Описание особенностей файловой системы ext4 и ее преимуществ по сравнению с ext3 сведено в табл. 4.1.

Таблица 4.1. Особенности ext4

Особенность	Комментарий
Увеличенный размер файла и файловой системы	<p>Для ext3 максимальный размер файловой системы составляет 32 Тбайт, а файла — 2 Тбайт, но на практике ограничения были более жесткими. Так, в зависимости от архитектуры, максимальный размер тома составлял до 2 Тбайт, а максимальный размер файла — до 16 Гбайт.</p> <p>В случае с ext4 максимальный размер тома составляет 1 эксбибайт (EiB) — это 2^{60} байт. Максимальный размер файла составляет 16 Тбайт. Такие объемы информации пока не нужны обычным пользователям, однако весьма пригодятся на серверах, работающих с большими дисковыми массивами</p>
Экстененты	<p>Основной недостаток ext3 — ее метод выделения места на диске. Дисковые ресурсы выделялись с помощью битовых карт свободного места, а такой способ не отличается ни скоростью, ни масштабируемостью. Получилось, что ext3 более эффективна для небольших файлов, но совсем не подходит для хранения больших файлов.</p>

Таблица 4.1 (окончание)

Особенность	Комментарий
	<p>Для улучшения выделения ресурсов и более эффективной организации данных в ext4 были введены <i>экстенты</i>. Экстент — это способ представления непрерывной последовательности блоков памяти.</p> <p>Благодаря использованию экстентов сокращается количество метаданных (служебных данных файловой системы), поскольку вместо информации о том, где находится каждый блок памяти, экстент содержит информацию о том, где находится большой список непрерывных блоков памяти.</p> <p>Для эффективного представления маленьких файлов в экстентах применяется уровневый подход, а для больших файлов используются деревья экстентов. Например, один индексный дескриптор может ссылаться на четыре экстента, каждый из которых может ссылаться на другие индексные дескрипторы и т. д. Такая структура является мощным механизмом представления больших файлов, а также более защищена и устойчива к сбоям</p>
Отложенное выделение пространства	Файловая система ext4 может отложить выделение дискового пространства до последнего момента, что увеличивает производительность системы
Контрольные суммы журналов	Контрольные суммы журналов повышают надежность файловой системы
Большее количество каталогов	В ext3 могло быть максимум 32 000 каталогов, в ext4 количество каталогов не ограничивается
Дефрагментация "на лету"	Файловая система ext3 не особо склонна к фрагментации, но все же такое неприятное явление имеется. В ext4 производится дефрагментация "на лету", что позволяет повысить производительность системы в целом
Наносекундные временные метки	В большинстве файловых систем временные метки (timestamp) устанавливаются с точностью до секунды, в ext4 точность повышена до наносекунды. Также ext4 поддерживает временные метки до 25 апреля 2514 года, в отличие от ext3 (18 января 2038 г.)

Совместимость с ext3

Файловая система ext4 является прямо и обратно совместимой с ext3, однако все же существуют некоторые ограничения. Предположим, что у нас на диске имеется файловая система ext4. Ее можно смонтировать и как ext3, и как ext4 (это и есть прямая совместимость) — и тут ограничений никаких нет. А вот с обратной совместимостью не все так безоблачно — если файловую систему ext4 смонтировать как ext3, то она будет работать без экстентов, что снизит ее производительность.

Переход на ext4

Если вы при установке системы выбрали файловую систему ext3, то перейти на ext4 можно без потери данных и в любой удобный для вас момент. Откройте терминал и введите команду:

```
sudo tune2fs -O extents,uninit_bg,dir_index /dev/имя_устройства
```

На момент ввода этой команды устройство должно быть размонтировано.

ВНИМАНИЕ!

Если нужно преобразовать в ext4 корневую файловую систему, то данную команду нужно вводить с LiveCD, поддерживающего ext4.

После этого проверим файловую систему:

```
sudo fsck -pf /dev/имя_устройства
```

Затем смонтируем файловую систему так:

```
mount -t ext4 /dev/имя_устройства /точка_монтирования
```

```
mount -t ext4 /dev/disk/by-uuid/UUID-устройства /точка_монтирования
```

Если раздел автоматически монтируется через /etc/fstab, не забудьте исправить файловую систему на ext4:

```
UUID=UUID-раздела    /точка ext4    defaults,errors=remount-ro,relatime
0 1
```

Если вы изменили тип файловой системы корневого раздела, тогда необходимо отредактировать файл /boot/grub/menu.lst и добавить опцию `rootfstype=ext4` в список параметров ядра, например:

```
title                Linux
root                 (hd0,1)
kernel /boot/vmlinuz-2.6.28.1 root=UUID=879f797c-944d-4c28-a720-249730705714
ro quiet splash rootfstype=ext4
initrd              /boot/initrd.img-2.6.28.1
quiet
```

СОВЕТ

Рекомендую прочитать статью Тима Джонса "Анатомия ext4":

<http://www.ibm.com/developerworks/ru/library/l-anatomy-ext4/index.html>.

4.1.4. Выбор файловой системы

С точки зрения производительности рассматриваемых файловых систем напрашиваются следующие рекомендации:

- ❖ для рабочей станции и сервера общего назначения оптимальной файловой системой являются ext4/ext3 или ReiserFS (в крайнем случае);

- ❖ на сервере баз данных можно использовать JFS — в этом случае (особенно, если база данных огромная) будет наблюдаться определенный прирост производительности;
- ❖ файловая система XFS — это удел мультимедиа-станции, на обычной рабочей станции или обычном сервере ее использовать не следует.

Но производительность — это не единственный критерий выбора файловой системы, особенно для сервера. Да, производительность учитывать нужно, но, кроме того, нельзя пренебрегать и следующими факторами:

- ❖ надежностью — все-таки мы выбираем файловую систему для сервера, а не для домашнего компьютера;
- ❖ наличием программ для восстановления файловой системы в случае сбоя — сбой может произойти даже в случае использования самой надежной файловой системы, поэтому наличие программного комплекса для восстановления файловой системы не будет лишним;
- ❖ максимальным размером файла — сервер обрабатывает огромные объемы информации, поэтому данный критерий для нас также важен.

Файловые системы ext4/ext3, ReiserFS и XFS одинаково надежны, а вот надежность JFS иногда оставляет желать лучшего. Учитывая это, а также то, что программы для восстановления файловой системы имеются только для ext*, на сервере лучше использовать все-таки ext4/ext3.

Если вы уже интересовались характеристиками файловых систем, то могли в некоторых источниках встретить неправильную информацию о максимальном размере файла для файловой системы ext3. Так, иногда сообщается, что максимальный размер файла для ext3 равен 2 Гбайт, что делает ее непригодной для использования на сервере. Это не так. Раньше, во времена ext2 и ядер 2.2 и 2.4, действительно, существовало такое ограничение, но для ext2. Файловая система ext3 поддерживает файлы размером до 1 Тбайт, а максимальный размер тома (раздела) равен 4 Тбайт, что вполне достаточно даже для сервера. Если же вам нужна поддержка больших объемов данных, тогда рекомендую обратить внимание на другие файловые системы, например, на ReiserFS (максимальный размер файла 16 Тбайт) или на XFS/JFS (размер файла вообще исчисляется в петабайтах).

4.2. Файловые системы Windows

Linux почти без всяких ограничений поддерживает файловые системы FAT12 (DOS), FAT16, или просто FAT (Windows 95), и FAT32 (Windows 98 и все последующие версии). Вы можете читать файлы и каталоги, изменять, создавать новые файлы и каталоги, удалять их — в общем, все, что и с файловой системой в Windows.

Однако файловые системы Windows не поддерживают установку прав доступа, поэтому можете даже не пытаться установить в Linux права доступа к файлу, который находится на Windows-разделе — у вас ничего не получится.

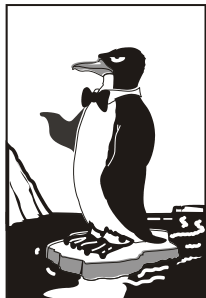
О файловой системе NTFS — отдельный разговор. По умолчанию (без перекомпиляции ядра) Linux умеет только читать данные, расположенные в NTFS-разделе. Однако даже после перекомпиляции ядра существует ряд ограничений на запись в NTFS-раздел: например, вы не можете создавать новые файлы, а можете только редактировать уже имеющиеся. Кстати, поддержка NTFS современным ядром до сих пор экспериментальна, т. е. в один не совсем прекрасный момент при попытке записи вы можете потерять данные в вашем NTFS-разделе.

Я вас напугал? Существуют решения (мы их рассмотрим в этой книге), позволяющие снять бóльшую часть ограничений на запись в NTFS-разделы. Конечно, все эти решения не идеальные: что-то работает, но ужасно медленно, что-то снимает далеко не все ограничения на запись, но, тем не менее, все же есть возможность записывать данные в NTFS-раздел без потери данных.

4.3. Сменные носители

Linux превосходно работает со сменными CD/DVD- и USB-дисками, даже в большинстве случаев выполняется автоматическое монтирование и размонтирование сменных носителей (хотя эта функция доступна не во всех дистрибутивах). С другой стороны, автоматическое монтирование сменных носителей на сервере — это от лукавого, на домашнем компьютере — да, но не на сервере. О монтировании, в том числе автоматическом, мы поговорим в *главе 7*.

ГЛАВА 5



Особенности файловой системы Linux

5.1. Имена файлов в Linux

В Linux несколько другие правила построения имен файлов, поэтому вам придется с этим смириться. Начнем с того, что в Linux нет такого понятия, как расширение файла. Например, возьмем имя файла `Document1.doc`. В Windows именем файла является "`Document1`", а "`.doc`" — это расширение. В Linux "`Document1.doc`" — это имя файла, расширения нет.

Максимальная длина имени файла — 254 символа. Имя может содержать любые символы (в том числе и кириллицу), кроме `/ \ ? < > * " |`. Кириллицу в именах файлов я бы не рекомендовал вообще использовать. Если вы уверены, что не будете эти файлы передавать Windows-пользователям (на сменном носителе, по электронной почте) — используйте на здоровье. А если же нужно будет отправить файл по электронной почте (кодировка-то у всех разная, поэтому вместо русскоязычного имени пользователь увидит абракадабру), лучше использовать латиницу в имени файла.

Также вам придется привыкнуть к тому, что система Linux чувствительна к регистру в имени файла: `FILE.txt` и `FiLe.Txt` — это два разных файла.

Для разделения элементов пути используется символ `/` (прямой слэш), а не `\` (обратный слэш), как в Windows.

5.2. Файлы и устройства

Пользователи Windows привыкли к тому, что файл — это именованная область данных на диске. Отчасти так оно и есть. Отчасти потому, что приведенное определение файла было верно для DOS (Disk Operating System) и Windows.

В Linux же понятие файла значительно шире. Сейчас Windows-пользователи будут очень удивлены: в Linux есть файлы устройств, позволяющие обращаться с устройством, как с обычным файлом. Файлы устройств находятся в каталоге `/dev`

(от англ. *devices*). Да, через файл устройства вы можем обратиться к устройству! Если вы работали в DOS, то, наверное, помните: что-то подобное было и в DOS. Ведь вы же еще не забыли зарезервированные имена файлов PRN (принтер), CON (клавиатура при вводе, дисплей при выводе), LPT n (параллельный порт, n — номер порта), COM n (последовательный порт). Кому-то может показаться, что разработчики Linux "украли" идею специальных файлов у Microsoft: ведь Linux появился в начале 90-х годов, а DOS — в начале 80-х годов прошлого века. Но на самом деле это не так. Наоборот, это Microsoft позаимствовала идею файлов устройств из операционной системы UNIX, которая появилась еще до создания операционных систем Microsoft. Но сейчас не время говорить об истории развития операционных систем, поэтому лучше вернемся к файлам устройств. Файлы устройств хранятся в каталоге `/dev`.

Вот некоторые примеры файлов устройств:

- ◆ `/dev/sdx` — файл жесткого диска;
- ◆ `/dev/sdxN` — файл устройства раздела на жестком диске, N — это номер раздела;
- ◆ `/dev/scdN` — файл устройства CD/DVD-привода;
- ◆ `/dev/mouse` — файл устройства мыши;
- ◆ `/dev/modem` — файл устройства модема (на самом деле является ссылкой на файл устройства `ttySn`);
- ◆ `/dev/ttySn` — файл последовательного порта, n — номер порта (`ttyS0` соответствует COM1, `ttyS1` — COM2 и т. д.).

В свою очередь, файлы устройств бывают двух типов: блочные и символьные. Обмен информацией с блочными устройствами, например, с жестким диском, осуществляется блоками информации, а с символьными — отдельными символами. Пример символьного устройства — последовательный порт.

5.3. Корневая файловая система и монтирование

Наверняка на вашем компьютере установлена система Windows. Откройте **Мой компьютер**. Скорее всего, вы увидите пиктограмму гибкого диска (имя устройства A:), пиктограммы разделов жесткого диска (пусть будет три раздела C:, D: и E:), пиктограмму привода CD/DVD (F:).

В Linux носители данных называются иначе. Первый дисковод для гибких дисков (в Windows это диск A:) называется `/dev/fd0`, второй (это диск B:) — `/dev/fd1`.

С жесткими дисками сложнее всего, поскольку одно и то же устройство может в различных версиях одного и того же дистрибутива называться по-разному. Например, мой IDE-диск, подключенный как первичный мастер, в Fedora 5 все еще назывался `/dev/hda`, а сейчас, в Fedora 8, он называется `/dev/sda`. Раньше накопители, подключающиеся к интерфейсу IDE (PATA), назывались `/dev/hdx`, а SCSI/SATA-накопители — `/dev/sdx` (где в обоих случаях x — буква).

После принятия `udev` и глобального уникального идентификатора устройств (UUID) все дисковые устройства, вне зависимости от интерфейса подключения (PATA, SATA, SCSI), называются `/dev/sdx`, где *x* — буква. Все современные дистрибутивы поддерживают `udev` и UUID. Так что не удивляйтесь, если вдруг ваш старенький IDE-винчестер будет назван `/dev/sda`. С одной стороны, это вносит некоторую путаницу (см. разд. 7.4.2). С другой стороны, все современные компьютеры оснащены именно SATA-дисками (т. к. PATA-диски уже устарели, а SCSI — дорогие), а на современных материнских платах только один контроллер IDE (PATA), потому многие пользователи даже ничего не заметят.

ПРИМЕЧАНИЕ

`udev` — это менеджер устройств, используемый в ядрах Linux версии 2.6. Пришел на смену более громоздкой псевдофайловой системы `devfs`. Управляет всеми манипуляциями файлами из каталога `./dev`.

Рассмотрим ситуацию с жесткими дисками чуть подробнее. Пусть у нас есть устройство `/dev/sda`. На жестком диске, понятное дело, может быть несколько разделов (логических дисков). В нашем случае на диске имеются три раздела, которые в Windows называются C:, D: и E:. Диск C: обычно является загрузочным (активным), поэтому он будет записан в самом начале диска. Нумерация разделов жесткого диска в Linux начинается с 1, поэтому в большинстве случаев диску C: будет соответствовать имя `/dev/sda1` — первый раздел на первом жестком диске.

Резонно предположить, что двум оставшимся разделам (D: и E:) были присвоены имена `/dev/sda2` и `/dev/sda3`. Это может быть так — и не так. Сейчас поясню. Раздел может быть первичным (primary partition), расширенным (extended partition) или логическим (logical partition). Всего на диске может быть или четыре первичных раздела, или три первичных и один расширенный.

Пусть на жестком диске есть четыре первичных раздела, для которых зарезервированы номера 1, 2, 3, 4. Если разделы D: и E: — первичные, то им будут присвоены имена `/dev/sda2` и `/dev/sda3`. Но в большинстве случаев данные разделы являются логическими, а логические разделы содержатся в расширенном разделе (там может быть максимум 11 логических разделов). При этом в Windows расширенному разделу не присваивается буква, потому что этот раздел не содержит данных пользователя, а только информацию о логических разделах. Логические разделы именуются, начиная с 5, т. е. если разделы D: и E: — логические, то им будут присвоены имена `/dev/sda5` и `/dev/sda6` соответственно.

Узнать номер раздела очень просто: достаточно запустить утилиту, работающую с таблицей разделов диска. В Mandriva это `diskdrake`, а в Fedora Core (ASPLinux) придется использовать стандартный `fdisk` или `cfdisk` (он немного удобнее). В Debian — `gparted` (кстати, очень удобное средство разметки диска). В openSUSE нужно выполнить команду **Компьютер | Центр управления | YaST**, а в открывшемся окне нажать кнопку **Средство разметки**. В большинстве случаев удобнее всего запустить (от имени `root`) утилиту `fdisk` — она есть в любом дистрибутиве Linux.

```
den@den-desktop:~$ sudo fdisk /dev/hda
Password:

Количество цилиндров для этого диска установлено в 19457.
С этим все в порядке, но значение больше, чем 1024,
и в отдельных установках могут возникнуть проблемы с:
1) программами, запускаемым при загрузке (напр., старые версии LILO)
2) загрузкой и программами разметки из других ОС
(напр., DOS FDISK, OS/2 FDISK)

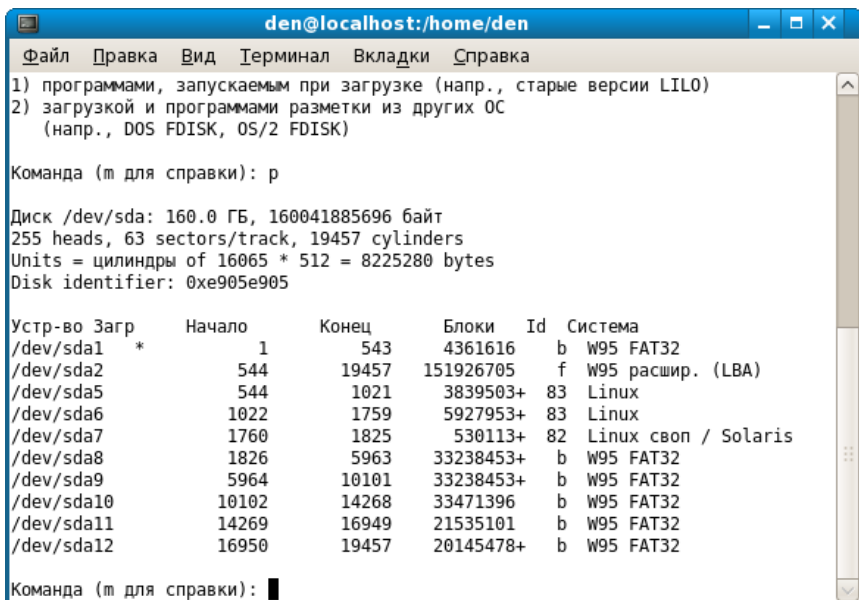
Команда (m для справки): p

Диск /dev/hda: 160.0 ГБ, 160041885696 байт
255 головок, 63 секторов/дорожку, 19457 цилиндров
Единицы = цилиндры по 16065 * 512 = 8225280 байт

Устр-во Загр   Начало      Конец      Блоки   Id Система
/dev/hda1  *         1          543      4361616   b  W95 FAT32
/dev/hda2             544      19457    151926705   f  W95 расшир. (LBA)
/dev/hda5             544      1021     3839503+   83  Linux
/dev/hda6            1022      1759     5927953+   83  Linux
/dev/hda7            1760      1825     530113+   82  Linux swap / Solaris
/dev/hda8            1826      5963     33238453+   b  W95 FAT32
/dev/hda9             5964     10101     33238453+   b  W95 FAT32
/dev/hda10           10102     14268     33471396   b  W95 FAT32
/dev/hda11           14269     16949     21535101   b  W95 FAT32
/dev/hda12          16950     19457     20145478+   b  W95 FAT32

Команда (m для справки): █
```

а



```
den@localhost:/home/den
Файл  Правка  Вид  Терминал  Вкладки  Справка
1) программами, запускаемым при загрузке (напр., старые версии LILO)
2) загрузкой и программами разметки из других ОС
(напр., DOS FDISK, OS/2 FDISK)

Команда (m для справки): p

Диск /dev/sda: 160.0 ГБ, 160041885696 байт
255 heads, 63 sectors/track, 19457 cylinders
Units = цилиндры of 16065 * 512 = 8225280 bytes
Disk identifier: 0xe905e905

Устр-во Загр   Начало      Конец      Блоки   Id Система
/dev/sda1  *         1          543      4361616   b  W95 FAT32
/dev/sda2             544      19457    151926705   f  W95 расшир. (LBA)
/dev/sda5             544      1021     3839503+   83  Linux
/dev/sda6            1022      1759     5927953+   83  Linux
/dev/sda7            1760      1825     530113+   82  Linux swap / Solaris
/dev/sda8            1826      5963     33238453+   b  W95 FAT32
/dev/sda9             5964     10101     33238453+   b  W95 FAT32
/dev/sda10           10102     14268     33471396   b  W95 FAT32
/dev/sda11           14269     16949     21535101   b  W95 FAT32
/dev/sda12          16950     19457     20145478+   b  W95 FAT32

Команда (m для справки): █
```

б

Рис. 5.1. Таблица разделов жесткого диска:
а — в старом дистрибутиве; б — в новом дистрибутиве

Чтобы узнать номера разделов первого жесткого диска (/dev/hda), введите команду:

```
# /sbin/fdisk /dev/sda
```

После этого вы увидите приглашение fdisk. В ответ на приглашение нужно ввести `p` и нажать клавишу `<Enter>`. Вы увидите таблицу разделов (рис. 5.1, б). После этого для выхода из программы введите `q` и нажмите клавишу `<Enter>`. Обратите внимание, что на рис. 5.1, а изображена таблица разделов моего IDE-диска в старом дистрибутиве, а на рис. 5.1, б — таблица разделов этого же диска, но в новом дистрибутиве.

На рис. 5.1 изображена таблица разделов моего первого жесткого диска. Первый раздел (это мой диск C:, где установлена система Windows) — первичный. Сразу после него расположен расширенный раздел (его номер — 2). Следующий за ним — логический раздел (номер 5). Разделы с номерами 3 и 4 пропущены, потому что их нет на моем жестком диске. Это те самые первичные разделы, которые я не создал — они мне не нужны.

Теперь, когда мы разобрались с именами жестких дисков и разделов, самое время вернуться в окно **Мой компьютер**. На рис. 5.2 изображено окно **Мой компьютер** моего домашнего компьютера. В этом окне отображаются все носители данных, доступные в системе.

В Linux есть понятие *корневой файловой системы*. Допустим, вы установили Linux в раздел с именем `/dev/sda3`. В этом разделе будет развернута корневая файловая система вашей Linux-системы. Корневой каталог обозначается как `/`, т. е. для перехода в корневой каталог в терминале (или консоли) нужно ввести команду `cd /`.

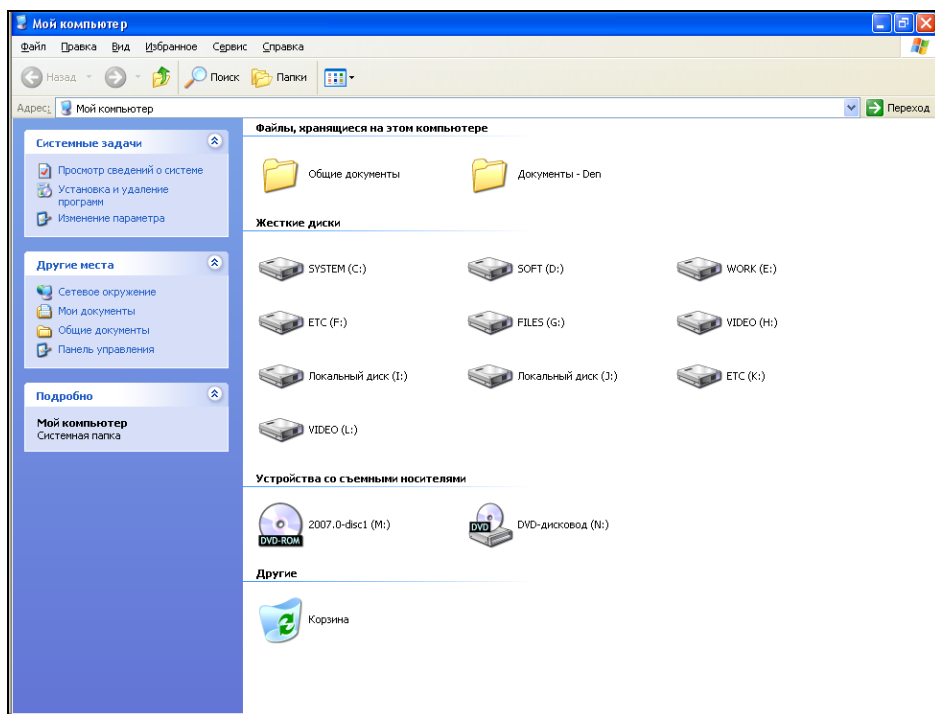


Рис. 5.2. Окно **Мой компьютер**

Понятно, что на вашем жестком диске есть еще разделы. Чтобы получить доступ к этим разделам, вам нужно подмонтировать их к корневой файловой системе. После монтирования вы можете обратиться к содержимому разделов через *точку монтирования* — назначенный вами при монтировании специальный каталог, например `/mnt/cdrom`. Монтированию файловых систем посвящена целая глава, поэтому сейчас не будем говорить об этом процессе подробнее.

5.4. Стандартные каталоги Linux

Файловая система любого дистрибутива Linux содержит следующие каталоги:

- ◆ `/` — корневой каталог;
- ◆ `/bin` — содержит стандартные программы Linux (cat, cp, ls, login и т. д.);
- ◆ `/boot` — каталог загрузчика, содержит образы ядра и Initrd, может содержать конфигурационные и вспомогательные файлы загрузчика;
- ◆ `/dev` — содержит файлы устройств;
- ◆ `/etc` — содержит конфигурационные файлы системы;
- ◆ `/home` — содержит домашние каталоги пользователей;
- ◆ `/lib` — библиотеки и модули;
- ◆ `/lost+found` — восстановленные после некорректного размонтирования файловой системы файлы и каталоги;
- ◆ `/misc` — может содержать все, что угодно, равно как и каталог `/opt`;
- ◆ `/mnt` — обычно содержит точки монтирования;
- ◆ `/proc` — каталог псевдофайловой системы procfs, предоставляющей информацию о процессах;
- ◆ `/root` — каталог суперпользователя root;
- ◆ `/sbin` — каталог системных утилит, выполнять которые имеет право пользователь root;
- ◆ `/tmp` — каталог для временных файлов;
- ◆ `/usr` — содержит пользовательские программы, документацию, исходные коды программ и ядра;
- ◆ `/var` — постоянно изменяющиеся данные системы, например очереди системы печати, почтовые ящики, протоколы, замки и т. д.

ГЛАВА 6



Команды для работы с файлами и каталогами. Права доступа

6.1. Работа с файлами

В данном разделе мы рассмотрим основные команды для работы с файлами в Linux (табл. 6.1), а в последующих разделах этой главы изучим команды для работы с каталогами, ссылками и поговорим о правах доступа к файлам и каталогам.

Таблица 6.1. Основные команды Linux, предназначенные для работы с файлами

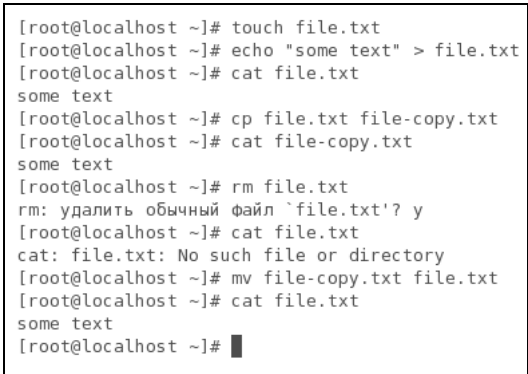
Команда	Назначение
<code>touch <файл></code>	Создает пустой файл
<code>cat <файл></code>	Просмотр текстового файла
<code>tac <файла></code>	Вывод содержимого текстового файла в обратном порядке, т. е. сначала выводится последняя строка, потом предпоследняя и т. д.
<code>cp <файл1> <файл2></code>	Копирует файл <файл1> в файл <файл2>. Если <файл2> существует, программа попросит разрешение на его перезапись
<code>mv <файл1> <файл2></code>	Перемещает файл <файл1> в файл <файл2>. Эту же команду можно использовать и для переименования файла
<code>rm <файл></code>	Удаляет файл
<code>locate <файл></code>	Производит быстрый поиск файла
<code>which <программа></code>	Выводит каталог, в котором находится программа, если она вообще установлена. Поиск производится в каталогах, указанных в переменной окружения <code>PATH</code> (это путь поиска программ)
<code>less <файл></code>	Используется для удобного просмотра файла с возможностью скроллинга (постраничной прокрутки)

ПРИМЕЧАНИЕ

Все представленные команды предназначены для работы в консоли, т. е. в текстовом режиме. Понятно, что большинство современных дистрибутивов запускаются в графическом режиме, поэтому некоторые пользователи Linux даже не подозревают о том, что существует консоль. Да, такое новое поколение Linux-пользователей, которым проще использовать графический файловый менеджер, чем вводить команды. Но если вы хотите стать квалифицированным Linux-пользователем, вы просто обязаны знать, как работать в консоли, иначе уподобитесь Windows-пользователям, которые при каждом сбое переустанавливают операционную систему... О том, как работать с консолью, вы узнаете в *главе 23*.

Рассмотрим небольшую серию команд (рис. 6.1):

```
touch file.txt
echo "some text" > file.txt
cat file.txt
cp file.txt file-copy.txt
cat file-copy.txt
rm file.txt
cat file.txt
mv file-copy.txt file.txt
cat file.txt
```



```
[root@localhost ~]# touch file.txt
[root@localhost ~]# echo "some text" > file.txt
[root@localhost ~]# cat file.txt
some text
[root@localhost ~]# cp file.txt file-copy.txt
[root@localhost ~]# cat file-copy.txt
some text
[root@localhost ~]# rm file.txt
rm: удалить обычный файл `file.txt'? y
[root@localhost ~]# cat file.txt
cat: file.txt: No such file or directory
[root@localhost ~]# mv file-copy.txt file.txt
[root@localhost ~]# cat file.txt
some text
[root@localhost ~]# █
```

Рис. 6.1. Операции с файлом

Первая команда (`touch`) создает в текущем каталоге файл `file.txt`. Вторая команда (`echo`) записывает в файл строку `"some text"` в этот же файл. Обратите внимание на `>` — это символ перенаправления ввода/вывода, о котором мы поговорим чуть позже.

Третья команда (`cat`) выводит содержимое файла — в файле записанная нами строка `"some text"`. Четвертая команда (`cp`) копирует файл `file.txt` в файл с именем `file-copy.txt`. После этого мы опять используем команду `cat`, чтобы вывести содержимое файла `file-copy.txt` — надо же убедиться, что файл действительно скопировался.

Шестая команда (`rm`) удаляет файл `file.txt`. При удалении система спрашивает, хотите ли вы удалить файл. Если хотите, то нужно нажать `<Y>`, а если нет, то `<N>`. Точно ли файл удален? Убедимся в этом: введите команду `cat file.txt`. Система нам сообщает, что нет такого файла.

Восьмая команда (`mv`) переименовывает файл `file-copy.txt` в файл `file.txt`. Последняя команда выводит исходный файл `file.txt`. Думаю, особых проблем с этими командами у вас не возникло, тем более, принцип действия этих команд вам должен быть знаком по командам DOS, которые, как квалифицированный пользователь Windows, вы должны знать наизусть.

Вместо имени файла иногда очень удобно указать маску имени файла. Например, у нас есть много временных файлов, которые заканчиваются строкой `"tmp"`, для их удаления нужно воспользоваться командой:

```
rm *tmp
```

Если же нужно удалить все файлы в текущем каталоге, можно просто указать звездочку:

```
rm *
```

Аналогично, можно использовать символ `?`, который в отличие от звездочки, заменяющей последовательность символов произвольной длины, заменяет всего один символ. Например, нам нужно удалить все файлы, имена которых состоят из трех букв и начинаются с `"d"`:

```
rm d??
```

Будут удалены файлы `d11`, `dbm`, `d78` и т. д., но не будут тронуты файлы, имена которых состоят из более чем трех букв и которые не начинаются на `"d"`.

Маски имен можно также использовать и при работе с каталогами.

6.2. Работа с каталогами

Основные команды для работы с каталогами приведены в табл. 6.2.

При указании имени каталога можно использовать следующие символы:

- ◆ `.` — означает текущий каталог, если вы введете команду `cat ./file`, то она выведет файл `file`, который находится в текущем каталоге;
- ◆ `..` — родительский каталог, например, команда `cd ..` перейдет на один уровень "вверх" по дереву файловой системы;
- ◆ `~` — домашний каталог пользователя (об этом мы поговорим позже).

Таблица 6.2. Основные команды для работы с каталогами

Команда	Описание
<code>mkdir <каталог></code>	Создание каталога
<code>cd <каталог></code>	Изменение каталога

Таблица 6.2 (окончание)

Команда	Описание
<code>ls <каталог></code>	Вывод содержимого каталога
<code>rmdir <каталог></code>	Удаление пустого каталога
<code>rm -r <каталог></code>	Рекурсивное удаление каталога

Теперь рассмотрим команды для работы с файлами на практике. Выполните следующие команды:

```
mkdir directory
cd directory
touch file1.txt
touch file2.txt
ls
cd ..
ls directory
rm directory
rmdir directory
rm -r directory
```

Первая команда (`mkdir`) создает каталог `directory` в текущем каталоге. Вторая команда (`cd`) переходит (изменяет каталог) в только что созданный каталог. Следующие две команды `touch` создают в новом каталоге два файла — `file1.txt` и `file2.txt`.

Команда `ls` без указания каталога выводит содержимое текущего каталога. Команда `cd ..` переходит в родительский каталог. Как уже было отмечено, в Linux родительский каталог обозначается как `..`, а текущий как `.`. То есть, находясь в каталоге `directory`, мы можем обращаться к файлам `file1.txt` и `file2.txt` без указания каталога или же как `./file1.txt` и `./file2.txt`.

Еще раз обратите внимание: в Linux в отличие от Windows для разделения элементов пути используется прямой слэш (`/`), а не обратный (`\`). Запомните это!

Кроме обозначений `..` и `.` в Linux часто используется обозначение `~` — это домашний каталог. Предположим, что наш домашний каталог `/home/den`. В нем мы создали подкаталог `dir` и поместили в него файл `file1.txt`. Полный путь к файлу можно записать так:

```
/home/den/dir/file1.txt
```

или так:

```
~/dir/file1.txt
```

Как видите, тильда (`~`) заменяет часть пути. Удобно? Конечно!

Поскольку мы находимся в родительском для каталога `directory` каталоге, для того чтобы вывести содержимое только что созданного каталога, в команде `ls` нам нужно четко указать имя каталога:

```
ls directory
```

Команда `rm` используется для удаления каталога. Но что мы видим: система отказывается удалять каталог! Пробуем удалить его командой `rmdir`, но и тут отказ. Система сообщает нам, что каталог не пустой, т. е. содержит файлы. Для удаления каталога нужно удалить все файлы. Конечно, делать это не очень хочется, поэтому проще указать опцию `-r` команды `rm` для рекурсивного удаления каталога. В этом случае сначала будут удалены все подкаталоги (и все файлы в этих подкаталогах), а затем будет удален сам каталог (рис. 6.2).

```
[root@localhost ~]# mkdir directory
[root@localhost ~]# cd directory
[root@localhost directory]# touch file.txt
[root@localhost directory]# touch file2.txt
[root@localhost directory]# ls
file2.txt  file.txt
[root@localhost directory]# cd ..
[root@localhost ~]# ls directory
file2.txt  file.txt
[root@localhost ~]# rm directory
rm: невозможно удалить каталог `directory': Is a directory
[root@localhost ~]# rmdir directory
rmdir: `directory': Directory not empty
[root@localhost ~]# rm -r directory
rm: спуститься в каталог `directory'? y
rm: удалить пустой обычный файл `directory/file.txt'? y
rm: удалить пустой обычный файл `directory/file2.txt'? y
rm: удалить каталог `directory'? y
[root@localhost ~]#
```

Рис. 6.2. Операции с каталогами

Команды `cp` и `mv` работают аналогично: для копирования (перемещения/переименования) сначала указывается каталог-источник, а потом каталог-назначение. Для каталогов желательно указывать параметр `-r`, чтобы копирование (перемещение) производилось рекурсивно.

6.3. Команды для работы со ссылками

В Linux допускается, чтобы один и тот же файл был в системе под разными именами. Для этого используются ссылки. Ссылки бывают двух типов: жесткие и символические. Жесткие ссылки жестко привязываются к файлу: вы не можете удалить файл, пока на него указывает хотя бы одна жесткая ссылка. А вот если на файл указывают символические ссылки, его удалению ничто не мешает.

Жесткие ссылки не могут указывать на файл, который находится за пределами файловой системы. Предположим, у вас два Linux-раздела. Один корневой, а второй используется для домашних файлов пользователей и монтируется к каталогу `/home` корневой файловой системы. Так вот, вы не можете создать в корневой файловой системе ссылку, которая ссылается на файл в файловой системе, подмонтированной к каталогу `/home`. Это очень важная особенность жестких ссылок. Если

вам нужно создать ссылку на файл, который находится за пределами файловой системы, вам нужно использовать символические ссылки.

Для создания ссылок используется команда `ln`:

```
ln file.txt link1
ln -s file.txt link2
```

Первая команда создает *жесткую ссылку* `link1`, ссылающуюся на текстовый файл `file1.txt`. Вторая команда создает *символическую ссылку* `link2`, которая ссылается на текстовый файл `file1.txt`.

Модифицируя ссылку (все равно какую — `link1` или `link2`), вы автоматически модифицируете исходный файл — `file1.txt`.

Особого внимания заслуживает операция удаления. По идее, если вы удаляете ссылку `link2`, файл `file.txt` также должен быть удален, но не тут-то было. Вы не можете его удалить до тех пор, пока на него указывает хотя одна жесткая ссылка. При удалении ссылки `link2` просто будет удалена символическая ссылка, но жесткая ссылка и сам файл останутся. Если же вы удалите ссылку `link1`, будет удален и файл `file.txt`, поскольку на него больше не ссылается ни одна жесткая ссылка.

6.4. Права доступа.

Команды *chown*, *chmod* и *chattr*

6.4.1. Права доступа к файлам и каталогам

Для каждого каталога и файла вы можете задать права доступа. Точнее, права доступа автоматически задаются при создании каталога/файла, а вам при необходимости нужно их изменить. Какая может быть необходимость? Например, вам нужно, чтобы к вашему файлу-отчету смогли получить доступ пользователи — члены вашей группы. Или вы создали обычный текстовый файл, содержащий инструкции командного интерпретатора. Чтобы этот файл стал сценарием, вам нужно установить право на выполнение для этого файла.

Существуют три права доступа: чтение (`r`), запись (`w`), выполнение (`x`). Для каталога право на выполнение означает право на просмотр содержимого каталога.

Вы можете установить разные права доступа для владельца (т. е. для себя), для группы владельца (т. е. для всех пользователей, входящих в одну с владельцем группу) и для прочих пользователей. Пользователь `root` может получить доступ к любому файлу или каталогу вне зависимости от прав, которые вы установили.

Чтобы просмотреть текущие права доступа, введите команду:

```
ls -l <имя файла/каталога>
```

Например,

```
ls -l video.txt
-r--r----- 1 den group 300 Apr 11 11:11 video.txt
```

`-r--r-----` — это права доступа. Первый символ — это признак каталога. Сейчас перед нами файл. Если бы перед нами был каталог, то первый символ был бы символом `d` (от *directory*).

Последующие три символа (`r--`) определяют права доступа владельца файла или каталога. Первый символ — это чтение, второй — запись, третий — выполнение. Как видно, владельцу разрешено только чтение этого файла, запись и выполнение запрещены, поскольку в правах доступа режимы `w` и `x` не определены.

Следующие три символа (`r--`) задают права доступа для членов группы владельца. Права такие же, как и у владельца: можно читать файл, но нельзя изменять или запускать.

Последние три символа (`---`) задают права доступа для прочих пользователей. Прочие пользователи не имеют право ни читать, ни изменять, ни выполнять файл. При попытке получить доступ к файлу они увидят сообщение "Access denied".

ПРИМЕЧАНИЕ

После прав доступа программа `ls` выводит имя владельца файла, имя группы владельца, размер файла, дату и время создания, а также имя файла.

Права доступа задаются командой `chmod`. Существуют два способа указания прав доступа: символьный (когда указываются символы, задающие право доступа — `r`, `w`, `x`) и абсолютный. Так уже заведено, что в мире UNIX чаще пользуются абсолютным методом.

Разберемся, в чем заключается этот метод. Рассмотрим следующий набор прав доступа:

`rw-r-----`

Данный набор прав доступа предоставляет владельцу право чтения и модификации файла (`rw-`), запускать файл владелец не может. Члены группы владельца могут только просматривать файл (`r--`), а все остальные пользователи не имеют вообще никакого доступа к файлу.

Возьмем отдельный набор прав, например для владельца:

`rw-`

Чтение разрешено, значит, мысленно записываем 1, запись разрешена, значит, запоминаем еще 1, а вот выполнение запрещено, поэтому запоминаем 0. Получается число 110. Если из двоичной системы перевести число 110 в восьмеричную, получится число 6. Для перевода можно воспользоваться табл. 6.3.

Таблица 6.3. Преобразование чисел из восьмеричной системы в двоичную

Восьмеричная система	Двоичная
0	000
1	001
2	010
3	011

Таблица 6.3 (окончание)

Восьмеричная система	Двоичная
4	100
5	101
6	110
7	111

Аналогично произведем разбор прав для членов группы владельца. Получится 100, т. е. 4. С третьим набором (---) все вообще просто — это 000, т. е. 0.

Записываем полученные числа в восьмеричной системе в порядке "владелец—группа—остальные". Получится число 640 — это и есть права доступа. Для того чтобы установить эти права доступа, выполните команду:

```
chmod 640 <имя_файла>
```

Наиболее популярные права доступа:

- ❖ 644 — владельцу можно читать и изменять файл, остальным пользователем — только читать;
- ❖ 666 — читать и изменять файл можно всем пользователям;
- ❖ 777 — всем можно читать, изменять и выполнять файл. Напомню, что для каталога право выполнения — это право просмотра оглавления каталога.

Иногда проще воспользоваться символьным методом. Например, у нас есть файл `script`, который нужно сделать исполнимым, для этого используется команда:

```
chmod +x script
```

Для того чтобы снять право выполнения, используется параметр `-x`:

```
chmod -x script
```

Подробнее о символьном методе вы сможете прочитать в руководстве по команде `chmod` (`man chmod`).

6.4.2. Смена владельца файла

Если вы хотите "подарить" кому-то файл, т. е. сделать какого-то пользователя владельцем файла, то вам нужно использовать команду `chown`:

```
chown пользователь файл
```

Учтите, что, возможно, после изменения владельца файла вы сами не сможете получить к нему доступ, ведь владелец уже не вы.

6.4.3. Специальные права доступа

Ранее мы рассмотрели обычные права доступа к файлам, но в Linux есть еще так называемые специальные права доступа: SUID (Set User ID root) и SGID (Set Group ID root).

Данные права доступа позволяют обычным пользователям запускать программы, требующие для своего запуска привилегий пользователя root. Например, демон rppd требует привилегий root, но чтобы каждый раз при установке PPP-соединения (модемное, ADSL-соединение) не входить в систему под именем root, достаточно установить специальные права доступа для демона rppd. Делается это так:

```
chmod u+s /usr/sbin/pppd
```

Однако не нужно увлекаться таким решением, поскольку каждая программа, для которой установлен бит SUID, является потенциальной "дырой" в безопасности вашей системы. Для выполнения программ, требующих прав root, намного рациональнее использовать программы sudo и su (описание которых можно получить по командам `man sudo` и `man su`).

6.4.4. Атрибуты файла. Запрет изменения файла

С помощью команды `chattr` можно изменить атрибуты файла. Параметр `+` устанавливает атрибут, а параметр `-` — атрибут снимает. Например:

```
# chattr +i /boot/grub/menu.lst
```

Данная команда устанавливает атрибут `i`, запрещающий любое изменение, переименование и удаление файла. Установить этот атрибут, равно как и снять его, имеет право только суперпользователь или процесс с возможностью `CAP_LINUX_IMMUTABLE`. Чтобы изменить файл, нужно очистить атрибут с помощью команды:

```
# chattr -i /boot/grub/menu.lst
```

Если установить атрибут `j`, то все данные прежде, чем они будут записаны непосредственно в файл, будут сохранены в журнал `ext3`. Данный атрибут имеет смысл только, если файловая система смонтирована с опциями `data=ordered` или `data=writeback` (см. главу 8). Когда файловая система смонтирована с опцией `data=journal`, данный атрибут не имеет значения, поскольку все данные файла и так уже журналируются. Об остальных атрибутах вы сможете прочитать в справочной системе:

```
man chattr
```

6.5. Команды поиска файлов

Для поиска файлов в Linux используется команда `find`. Это довольно мощная утилита со сложным синтаксисом и далеко не всегда нужна обычному пользователю. Обычному пользователю намного проще будет установить файловый менеджер `mc` и использовать встроенную функцию поиска.

Но команду `find` мы все же рассмотрим, по крайней мере, ее основы. Синтаксис команды следующий:

```
find список_поиска выражение
```

Мощность программы `find` заключается в множестве самых разных параметров поиска, которые не так легко запомнить — их просто много. К тому же `find` может выполнять команды для найденных файлов. Например, вы можете найти временные файлы и сразу удалить их.

Подробно опции команды `find` мы рассматривать не будем — это вы можете сделать самостоятельно с помощью команды `man find`. Зато мы рассмотрим несколько примеров использования этой команды.

Попытаемся найти файлы с именем `a.out` (точнее, в имени которых содержится строка `"a.out"`), поиск начнем с корневого каталога (`/`):

```
find / -name a.out
```

Найдем файлы по маске `*.txt`:

```
find / -name '*.txt'
```

Найдем файлы нулевого размера, поиск начнем с текущего каталога (`.`):

```
find . -size 0c
```

Хотя для поиска пустых файлов намного проще использовать параметр `-empty`:

```
find . -empty
```

Найдем файлы, размер которых от 100 до 150 Мбайт, поиск произведем в домашнем каталоге и всех его подкаталогах:

```
find ~ -size +100M -size -150M
```

Найдем все временные файлы и удалим их (для каждого найденного файла будет запущена команда `rm`):

```
# find / -name *.tmp -ok rm {} \;
```

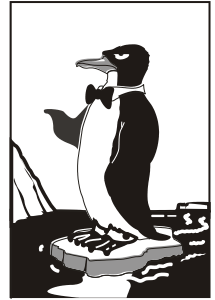
Вместо параметра `-ok` можно использовать параметр `-exec`, который также запускает указанную после него команду, но не запрашивает подтверждение выполнения этой команды для каждого файла.

Кроме команды `find` можно использовать команды `which` и `locate`. Первая выводит полный путь к программе или к сценарию, если программа или сценарий находятся в списке каталогов, заданном в переменной окружения `PATH`:

```
which sendmail
```

Программа `locate` ищет в базе данных демона `located` файлы, соответствующие заданному образцу. Недостаток этой команды в том, что `located` имеется далеко не во всех дистрибутивах, поэтому команды `locate` у вас может и не быть. Зато если `located` имеется и запущен, поиск файлов будет осуществляться быстрее, чем с помощью `find`.

ГЛАВА 7



Монтирование файловых систем

7.1. Команда *mount*

Чтобы использовать какую-либо файловую систему, нужно подмонтировать ее к корневой файловой системе. Например, вы вставляете в дисковод дискету. После этого нужно подмонтировать файловую систему дискеты к корневой файловой системе. Только так мы сможем получить доступ к файлам и каталогам, которые записаны на дискете. Аналогичная ситуация с жесткими, оптическими дисками и другими носителями данных.

Если вы хотите заменить сменный носитель данных (дискету, компакт-диск), вам нужно сначала размонтировать файловую систему, затем извлечь носитель данных, сменить его новым — и заново смонтировать файловую систему.

Размонтирование файловой системы также необходимо для извлечения сменного носителя. В случае с дискетой о размонтировании должны помнить вы сами, поскольку при размонтировании выполняется синхронизация буферов ввода/вывода и файловой системы, т. е. данные физически записываются на диск, если это еще не было сделано. А в случае с компакт-диском система не разрешит вам извлечь диск, если он не размонтирован. В свою очередь, размонтировать смонтированную файловую систему можно, если ни один процесс не использует эту файловую систему.

При завершении работы системы (перезагрузке, выключении компьютера) размонтирование всех файловых систем выполняется автоматически.

Команда монтирования выглядит так (команду нужно выполнять с привилегиями root):

```
# mount [опции] <устройство> <точка монтирования>
```

Точка монтирования — это каталог, через который будет осуществляться доступ к монтируемой файловой системе. Например, если вы подмонтировали компакт-диск к каталогу `/mnt/cdrom`, то получить доступ к файлам и каталогам, записанным на компакт-диске, можно будет через точку монтирования — каталог `/mnt/cdrom`. Точка монтирования — это любой каталог корневой файловой систе-

мы, хоть, /aaa-111. Главное, чтобы этот каталог существовал на момент монтирования файловой системы.

ПРИМЕЧАНИЕ

В современных дистрибутивах обычно запрещен вход в систему под именем суперпользователя — `root`. Поэтому для выполнения команд с привилегиями `root` вам нужно использовать команду `sudo`. Например, чтобы выполнить команду монтирования привода компакт-диска, вам надо ввести команду:

```
sudo mount /dev/sdc /mnt/cdrom
```

Перед выполнением команды `mount` команда `sudo` попросит вас ввести пароль текущего пользователя (под учетной записью которого вы в данный момент работаете). Если введенный пароль правильный, то будет выполнена команда `mount`. Но для выполнения команды `sudo` у пользователя должны быть определенные полномочия, а именно пользователь должен быть "прописан" в файле `/etc/sudoers`, редактировать который имеет право только администратор. Если же вам лень изменять этот файл, можно ввести команду `su`. Она запросит у вас пароль пользователя `root`. После этого все команды можно выполнить от имени пользователя `root`.

Для размонтирования файловой системы используется команда `umount`:

```
# umount <устройство или точка монтирования>
```

7.2. Файлы устройств и монтирование

В *главе 5* мы уже говорили о файлах устройств. В этом разделе мы поговорим о них снова, но в контексте монтирования файловой системы.

Как уже было отмечено в *главе 5*, для Linux нет разницы между устройством и файлом. Все устройства системы представлены в корневой файловой системе как обычные файлы. Например, `/dev/fd0` — это ваш дисковод для гибких дисков, `/dev/sda` — жесткий диск. Файлы устройств хранятся в каталоге `/dev`.

Также было уже отмечено, что все жесткие диски вне зависимости от интерфейса (ATA, SATA, SCSI) теперь называются `/dev/sdx`. На жестком диске, понятное дело, будет не один раздел, поэтому нам нужно знать номер монтируемого раздела, например, файл устройства `/dev/sda1` в большинстве случаев соответствует Windows-диску C: на рабочей станции. Сервер же работает под управлением Linux, и вторая операционная система на нем не предусмотрена, поэтому с большой уверенностью можно сказать, что этот файл будет соответствовать корневому разделу Linux. Узнать номер раздела можно с помощью программы разметки диска. Практически во всех дистрибутивах есть программа `fdisk`. С ней мы уже успели познакомиться в *главе 5*. Чуть позже в этой главе мы продолжим наше знакомство с этой программой.

Приводы для чтения/записи CD/DVD-дисков называются `/dev/scdN`, где `N` — номер устройства. Если у вас только один привод CD/DVD-дисков, то его имя будет `/dev/scd0`.

Для монтирования привода для чтения оптических дисков нужно ввести команду:

```
# mount /dev/scd0 /mnt/cdrom
```

После этого обратиться к файлам, записанным на диске, можно будет через каталог `/mnt/cdrom`. Напомню, что каталог `/mnt/cdrom` должен существовать. Каталог `/mnt/cdrom` называется *точкой монтирования*.

Аналогичная ситуация и с дискетами. В системе могут быть установлены два дисководов для дискет — первый (`/dev/fd0`) и второй (`/dev/fd1`). Для их монтирования можно использовать команды:

```
# mount /dev/fd0 /mnt/floppy
```

```
# mount /dev/fd1 /mnt/floppy
```

В Windows-терминологии устройство `/dev/fd0` — это диск A:, а устройство `/dev/fd1` — диск B:.

7.3. Типы файловых систем

У команды `mount` довольно много опций, но на практике, сами понимаете, наиболее часто используются только некоторые из них. Наиболее востребованными являются опции `-t`, `-r`, `-w` и `-a`.

Опция `-t` позволяет задать тип файловой системы. Обычно программа сама определяет файловую систему, но иногда это у нее не получается. Тогда мы должны ей помочь. Формат использования этой опции следующий:

```
# mount -t <файловая система> <устройство> <точка монтирования>
```

Например,

```
# mount -t iso9660 /dev/scd0 /mnt/cdrom
```

Вот наиболее популярные файловые системы:

- ❖ `ext2` или `ext3` — файловая система Linux;
- ❖ `iso9660` — указывается при монтировании CD-ROM;
- ❖ `vfat` — FAT, FAT32 (поддерживается Windows 9x, ME, XP, Vista);
- ❖ `ntfs` — NT File System (поддерживается NT, XP), в этом случае файловая система NTFS будет доступна в режиме "только чтение";
- ❖ `ntfs-3g` — будет использован модуль `ntfs-3g`, входящий в большинство современных дистрибутивов. Данный модуль позволяет производить запись информации на NTFS-разделы.

ПРИМЕЧАНИЕ

Если в вашем дистрибутиве нет модуля `ntfs-3g`, т. е. при попытке указания данной файловой системы вы увидите сообщение об ошибке, тогда можете скачать его с сайта www.ntfs-3g.org. На данном сайте доступны как исходные коды, так и уже откомпилированные для разных дистрибутивов пакеты.

Параметр `-r` монтирует указанную файловую систему в режиме "только чтение". А параметр `-w` монтирует файловую систему в режиме "чтение/запись". Данный параметр используется по умолчанию для файловых систем, поддерживающих запись (например, NTFS по умолчанию запись не поддерживает, как и файловые системы CD/DVD-дисков).

Последний параметр, параметр `-a`, применяется для монтирования всех файловых систем, указанных в файле `/etc/fstab` (кроме тех, для которых указано `noauto` — такие файловые системы нужно монтировать вручную). При загрузке системы вызывается программа `mount` с параметром `-a`.

7.4. Монтирование разделов при загрузке

7.4.1. Формат файла `/etc/fstab`

Если вы не хотите при каждой загрузке монтировать постоянные файловые системы (например, ваши Windows-разделы), то вам нужно прописать их в файле `/etc/fstab`. Обратите внимание — в этом файле не нужно прописывать файловые системы сменных носителей (дисковода, CD/DVD-привода, Flash-диска). Следует отметить, что программы установки некоторых дистрибутивов, например Mandriva, читают таблицу разделов и автоматически заполняют файл `/etc/fstab`. В результате все ваши Windows-разделы доступны сразу после установки системы. К сожалению, не все дистрибутивы могут похвастаться такой интеллектуальностью, поэтому вам нужно знать формат файла `fstab`:

устройство точка_монтирования тип_ФС опции флаг_РК флаг_проверки

Здесь: *тип_ФС* — это тип файловой системы, а *флаг_РК* — флаг резервного копирования. Если он установлен (1), то программа `dump` заархивирует данную файловую систему при создании резервной копии. Если не установлен (0), то резервная копия этой файловой системы создаваться не будет. *флаг_проверки* устанавливает, будет ли данная файловая система проверяться на наличие ошибок программой `fsck`. Проверка производится в двух случаях:

- ❖ если файловая система размонтирована некорректно;
- ❖ если достигнуто максимальное число операций монтирования для этой файловой системы.

Поле опций содержит важные параметры файловой системы. Некоторые из них представлены в табл. 7.1.

ПРИМЕЧАНИЕ

Редактировать файл `/etc/fstab`, как и любой другой файл из каталога `/etc`, можно в любом текстовом редакторе (например, `gedit`, `kate`), но перед этим нужно получить права `root` (команды `su` или `sudo`).

Таблица 7.1. Опции монтирования файловой системы в файле `/etc/fstab`

Опция	Описание
<code>auto</code>	Файловая система должна монтироваться автоматически при загрузке. Опция используется по умолчанию, поэтому ее указывать не обязательно
<code>noauto</code>	Файловая система не монтируется при загрузке системы (при выполнении команды <code>mount -a</code>), но ее можно смонтировать вручную с помощью все той же команды <code>mount</code>
<code>defaults</code>	Используется стандартный набор опций, установленных по умолчанию
<code>exec</code>	Разрешает запуск выполняемых файлов для данной файловой системы. Эта опция используется по умолчанию
<code>noexec</code>	Запрещает запуск выполняемых файлов для данной файловой системы
<code>ro</code>	Монтирование в режиме "только чтение"
<code>rw</code>	Монтирование в режиме "чтение/запись". Используется по умолчанию для файловых систем, поддерживающих запись
<code>user</code>	Данную файловую систему разрешается монтировать/размонтировать обычному пользователю (не <code>root</code>)
<code>nouser</code>	Файловую систему может монтировать только пользователь <code>root</code> . Используется по умолчанию
<code>umask</code>	Определяет маску прав доступа при создании файлов. Для файловых систем не на базе Linux маску нужно установить так: <code>umask=0</code>
<code>utf8</code>	Применяется только на дистрибутивах, которые используют кодировку UTF8 в качестве кодировки локали. В старых дистрибутивах (где используется KOI8-R) для корректного отображения русских имен файлов на Windows-разделах нужно указывать параметры <code>iocharset=koi8-u</code> , <code>codepage=866</code>

Рассмотрим небольшой пример:

```
/dev/scd0 /mnt/cdrom auto umask=0,user,noauto,ro,exec 0 0
/dev/sda1 /mnt/win_c vfat umask=0,utf8 0 0
```

Первая строка — это строка монтирования файловой системы компакт-диска, а вторая — строка монтирования диска C:.

♦ Начнем с первой строки. `/dev/scd0` — это имя устройства CD-ROM. Точка монтирования — `/mnt/cdrom`. Понятно, что этот каталог должен существовать. Обратите внимание — в качестве файловой системы не указывается жестко `iso9660`, поскольку компакт-диск может быть записан в другой файловой системе, поэтому в качестве типа файловой системы задано `auto`, т. е. автоматическое определение. Теперь идет довольно длинный набор опций. Ясно, что `umask` установлен в ноль, поскольку файловая система компакт-диска не под-

держивает права доступа Linux. Параметр `user` говорит о том, что данную файловую систему можно монтировать обычному пользователю. Параметр `noauto` запрещает автоматическое монтирование этой файловой системы, что правильно — ведь на момент монтирования в приводе может и не быть компакт-диска. Опция `ro` разрешает монтирование в режиме "только чтение", а `exec` разрешает запускать исполняемые файлы. Понятно, что компакт-диск не нуждается ни в проверке, ни в создании резервной копии, поэтому два последних флага равны нулю.

- ❖ Вторая строка проще. Первые два поля — это устройство и точка монтирования. Третье — тип файловой системы. Файловая система постоянна, поэтому можно явно указать тип файловой системы (`vfat`), а не `auto`. Опция `umask`, как и в предыдущем случае, равна нулю. Параметр `utf8` нужен для корректного отображения имен файлов

7.4.2. Подробно о UUID и /etc/fstab

Пока вы еще не успели забыть формат файла `/etc/fstab`, нужно поговорить о UUID (Universally Unique Identifier), или о *длинных именах* дисков. В некоторых дистрибутивах, например, в Ubuntu, вместо имени носителя (первое поле файла `fstab`) указывается его ID, поэтому `fstab` выглядит устрашающее, например вот так:

```
# /dev/hda6
UUID=1f049af9-2bdd-43bf-a16c-ff5859a4116a / ext3 defaults 0 1
# /dev/hda1
UUID=45AE-84D9 /media/hda1 vfat defaults,utf8,umask=007 0 0
```

В SUSE 10.3/11 идентификаторы устройств указываются немного иначе:

```
/dev/disk/by-id/scsi-SATA_WDC_WD1600JB-00_WD-WCANM7959048-part5 / ext3
acl,user_xattr 1 1
/dev/disk/by-id/scsi-SATA_WDC_WD1600JB-00_WD-WCANM7959048-part7 swap swap de-
faults 0 0
```

Понятно, что использовать короткие имена вроде `/dev/sda1` намного проще, чем идентификаторы в стиле `1f049af9-2bdd-43bf-a16c-ff5859a4116a`. Использование имен дисков еще никто не отменял, поэтому вместо идентификатора носителя можете смело указывать его файл устройства — так вам будет значительно проще!

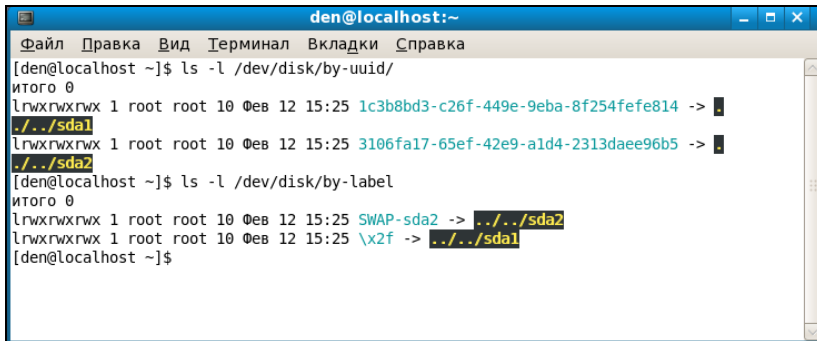
Но все же вам нужно знать соответствие длинных имен коротким именам устройств. Ведь система использует именно эти имена, а в файле `/etc/fstab` не всегда указывается, какой идентификатор принадлежит какому короткому имени устройства (или указывается, но не для всех разделов).

Узнать "длинные имена" устройства можно с помощью простой команды:

```
ls -l /dev/disk/by-uuid/
```

Результат выполнения этой команды приведен на рис. 7.1.

Спрашивается, зачем введены длинные имена, если короткие имена были удобнее, во всяком случае для пользователей? Оказывается, разработчики Linux в первую очередь и заботились как раз о пользователях. Возьмем обычный IDE-диск. Как известно, данный диск можно подключить либо к первичному (primary), либо к вторичному (secondary), если он есть, контроллеру. В зависимости от положения перемычки выбора режима, винчестер может быть либо главным устройством (master), либо подчиненным (slave). Таким образом, в зависимости от контроллера, к которому подключается диск, изменяется его короткое имя — hda (primary master), hdb (primary slave), hdc (secondary master), hdd (secondary slave). То же самое происходит с SATA/SCSI-винчестерами — при изменении параметров подключения изменяется и короткое имя устройства.



```

den@localhost:~
Файл Правка Вид Терминал Вкладки Справка
[den@localhost ~]$ ls -l /dev/disk/by-uuid/
итого 0
lrwxrwxrwx 1 root root 10 Фев 12 15:25 1c3b8bd3-c26f-449e-9eba-8f254fefe814 -> ../../sda1
lrwxrwxrwx 1 root root 10 Фев 12 15:25 3106fa17-65ef-42e9-aid4-2313daee96b5 -> 
[den@localhost ~]$ ls -l /dev/disk/by-label
итого 0
lrwxrwxrwx 1 root root 10 Фев 12 15:25 SWAP-sda2 -> ../../sda2
lrwxrwxrwx 1 root root 10 Фев 12 15:25 \x2f -> ../../sda1
[den@localhost ~]$
  
```

Рис. 7.1. Соответствие длинных имен дисков коротким

При использовании же длинных имен идентификатор дискового устройства остается постоянным вне зависимости от типа подключения устройства к контроллеру. Именно поэтому длинные имена дисков часто также называются *постоянными именами* (persistent name). Получается, что раньше вы могли ошибочно подключить жесткий диск немного иначе, и разделы, которые назывались, скажем, /dev/hdaN, стали называться /dev/hdbN. Понятно, что загрузить Linux с такого диска не получится, поскольку везде указаны другие имена устройств. Если же используются длинные имена дисков, система загрузится в любом случае, как бы вы ни подключили жесткий диск. Удобно? Конечно.

Но это еще не все. Постоянные имена — это только первая причина. Вторая причина заключается в обновлении библиотеки libata. В новой версии libata все PATA-устройства именуются не как hdx, а как sdx, что (как отмечалось ранее в этой главе) вносит некую путаницу. Длинные имена дисков от этого не изменяются, поэтому они избавляют пользователя от беспокойства по поводу того, что его старый IDE-диск вдруг превратился в SATA/SCSI-диск.

При использовании UUID однозначно идентифицировать раздел диска можно несколькими способами:

- ◆ UUID=45AE-84D9 /media/hda1 vfat defaults,utf8,umask=007,gid=46 0 0 — здесь с помощью параметра UUID указывается идентификатор диска;

- ❖ `/dev/disk/by-id/scsi-SATA_WDC_WD1600JB-00_WD-WCANM7959048-part7 swap`
`swap defaults 0 0` — здесь указывается длинное имя устройства диска;
- ❖ `LABEL=/ / ext3 defaults 1 1` — самый компактный третий способ, позволяющий идентифицировать устройства по их метке.

ПРИМЕЧАНИЕ

Первый способ получения длинного имени в англоязычной литературе называется "by-uuid", т. е. длинное имя составляется по UUID. Второй способ называется "by-id", т. е. по аппаратному идентификатору устройства. Третий способ называется "by-label" — по метке. Просмотреть соответствие длинных имен коротким можно с помощью команд:

```
ls -l /dev/disk/by-uuid
ls -l /dev/disk/by-id
ls -l /dev/disk/by-label
```

Но есть еще и четвертый способ, который называется "by-path". В этом случае имя генерируется по sysfs. Данный способ является наименее используемым, поэтому вы редко столкнетесь с ним.

Узнать метки разделов можно с помощью команды:

```
ls -lF /dev/disk/by-label
```

Установить метку можно с помощью команд, указанных в табл. 7.2.

Таблица 7.2. Команды для установки меток разделов

Файловая система	Команда
ext2/ext3	# <code>e2label /dev/XXX <метка></code>
ReiserFS	# <code>reiserfstune -l <метка> /dev/XXX</code>
JFS	# <code>jfs_tune -L <метка> /dev/XXX</code>
XFS	# <code>xfs_admin -L <метка> /dev/XXX</code>
FAT/FAT32	Только средствами Windows
NTFS	# <code>ntfslabel /dev/XXX <метка></code>

В файле `/etc/fstab` вы можете использовать длинные имена в любом формате. Можно указывать имена устройств в виде: `/dev/disk/by-uuid/*`, `/dev/disk/by-id/*` или `/dev/disk/by-label/*`, можно использовать параметры `UUID=идентификатор` или `LABEL=метка`. Используйте тот способ, который вам больше нравится.

7.5. Flash-диски

В последнее время очень популярна Flash-память. Уже сегодня Flash-память, точнее 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 удивит небольшим брелоком, позволяющим хранить до 32 Гбайт информации.

Принцип использования Flash-диска очень прост — достаточно подключить его к шине USB, и через несколько секунд система 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.

Технология Flash-памяти нашла свое применение в различных портативных устройствах — от мобильных телефонов до цифровых фотоаппаратов. Вы можете подключить мобильник к компьютеру и работать с ним как с обычным диском — записывать на него мелодии и картинки. Аналогичная ситуация и с цифровым фотоаппаратом: когда вы фотографируете, то фотографии и видеоролики записываются на его 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.

Все современные дистрибутивы умеют автоматически монтировать Flash-диски. После монтирования открывается окно (рис. 7.2) с предложением просмотреть содержимое диска или же импортировать фотографии (в зависимости от типа подключенного устройства — обычный USB-диск или фотоаппарат).

Понятно, что нам, как настоящим линуксоидам, интересно, как самостоятельно смонтировать Flash-диск. Оказывается, тут все просто. USB-диск — это обычный накопитель, и его можно увидеть в каталоге `/dev/disk/by-id`. Напомню, что способ "by-id" подразумевает получение длинного имени по аппаратному идентификатору устройства, а поэтому с помощью каталога `/dev/disk/by-id` проще всего найти длинное имя USB-диска среди имен других накопителей: в его начале будет префикс `usb_`. Введите команду:

```
ls -l /dev/disk/by-id | grep usb
```

Результат выполнения этой команды представлен на рис. 7.3.

Исходя из рис. 7.3, для монтирования Flash-диска нужно выполнить команду:

```
# mount /dev/sdb1 /mnt/flash
```

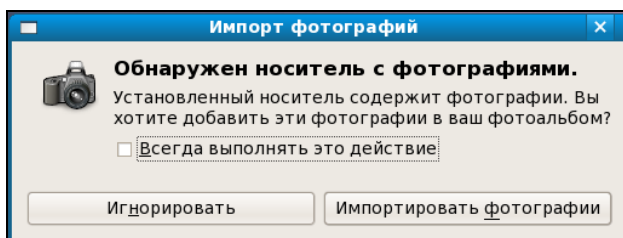
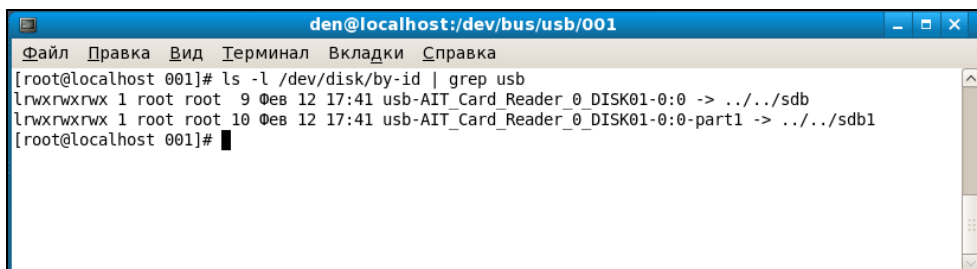


Рис. 7.2. Подключение Flash-диска

ПРИМЕЧАНИЕ

О файловой системе Linux можно говорить бесконечно долго, поэтому часть интересной информации вынесена в *главу 8* (и так эта глава получилась настолько большой, что вряд ли вы ее прочитали за один раз). В *главе 8* мы рассмотрим разные интересные трюки, которые позволяет проделывать файловая система Linux.



```
den@localhost:/dev/bus/usb/001
Файл  Правка  Вид  Терминал  Вкладки  Справка
[root@localhost 001]# ls -l /dev/disk/by-id | grep usb
lrwxrwxrwx 1 root root 9 0ев 12 17:41 usb-AIT_Card_Reader_0_DISK01-0:0 -> ../../sdb
lrwxrwxrwx 1 root root 10 0ев 12 17:41 usb-AIT_Card_Reader_0_DISK01-0:0-part1 -> ../../sdb1
[root@localhost 001]#
```

Рис. 7.3. USB-диск найден

ГЛАВА 8



Особые операции при работе с файловой системой

8.1. Создание и монтирование образов CD/DVD

Довольно часто нужно создать образ оптического диска (не знаю, как у вас, но у меня такая потребность возникает примерно один раз в неделю). Причина проста: или под рукой нет чистой "болванки", или же надо поработать с диском, который нужно отдать, но при этом нет никакого желания записывать его на болванку.

В Windows для создания образа диска мы используем посторонние программы, например Nero или WinImage. В Linux мы будем применять только средства операционной системы.

С помощью команды `dd` можно создать образ CD/DVD-диска. Делается это так:

```
dd if=/dev/cdrom of=~/cd.iso
```

Вместо `/dev/cdrom` нужно подставить файл устройства вашего привода CD/DVD, хотя обычно этого делать не нужно, поскольку ссылка `/dev/cdrom` устанавливается самой системой на ваш привод CD/DVD.

Данная команда создаст образ `cd.iso`, который будет записан в ваш домашний каталог. Аналогично с помощью данной команды можно создать образ дискеты: только вместо `/dev/cdrom` нужно указать имя файла устройства `/dev/fd0`.

Что можно сделать в Windows с ISO-образом? Его можно записать на чистую болванку или же открыть в специальной программе для изменения (для этого я использую программы `ISOpen` и `UltraISO`). В Linux открыть образ можно с помощью средств самой операционной системы. Для этого его нужно просто подмонтировать к корневой файловой системе.

Смонтировать образ диска можно с помощью команды:

```
# mount -o loop -t iso9660 образ точка_монтирования
```

Опция `-o loop` означает, что будет монтироваться не файл устройства, а образ диска, который записан на жесткий диск. Следующий параметр `-t 9660` задает тип файловой системы образа: `iso9660` — стандартная файловая система для CD/DVD.

После файловой системы указывается файл образа, например `~/cd.iso`. Последний параметр — это точка монтирования, каталог, к которому будет подмонтирован образ (напомню, что каталог должен существовать).

ПРИМЕЧАНИЕ

В большинстве случаев команду `mount` нужно выполнять от имени пользователя `root` или через команды `sudo/su`.

В нашем случае для монтирования образа `~/cd.iso` к каталогу `/mnt/image` нужно выполнить команду:

```
# mount -o loop -t iso9660 ~/cd.iso /mnt/image
```

После этого можно обращаться к образу как к обычному каталогу:

```
ls /mnt/image
```

8.2. Запись образов на болванку

Предположим, у вас есть образ `cd.iso`, и вы хотите записать его на компакт-диск, но не хотите (или не имеете возможности) использовать графические программы вроде Nero или k3b. В этом случае вам нужно использовать программу `cdrecord` (пакет называется аналогично). Команда для записи образа на болванку CD-R очень проста и выглядит так:

```
# cdrecord dev=0,0,0 -dao speed=16 файл_образа
```

Для записи DVD-R используется аналогичная команда:

```
# dvdrecord dev=0,0,0 -dao speed=4 файл_образа
```

В этой команде вам нужно изменить параметр `dev` — это идентификатор устройства CD/DVD. Если в вашей системе установлен только один привод CD/DVD, и он же является пишущим, тогда, скорее всего, у него будет идентификатор `0,0,0`. Но если у вас несколько приводов CD/DVD (например, обычный и пишущий), вы должны ввести следующую команду:

```
# cdrecord -scanbus
```

Команда выведет список CD/DVD, установленных в вашей системе (рис. 8.1). Вам надо запомнить идентификатор нужного привода и использовать его при записи образа диска.

Для очистки DVD-RW диска используется команда:

```
# dvd+rw-format -f имя_устройства_DVD-RW
```

Для быстрой очистки CD-RW введите команду:

```
# cdrecord -v blank=fast dev=0,0,0
```

```

den@localhost: /home/den - Shell - Konsole
Сеанс  Правка  Вид  Закладки  Настройка  Справка
[den@localhost ~]$ cdrecord -scanbus
Cdrecord-ProDVD-Clone 2.01.01a11 (i686-pc-linux-gnu) Copyright (C) 1995-2006 Jörg
Schilling
Linux sg driver version: 3.5.33
Using libscg version 'schily-0.8'.
scsibus0:
  0,0,0 0) 'VMware, ' 'VMware Virtual S' '1.0 ' Disk
  0,1,0 1) *
  0,2,0 2) *
  0,3,0 3) *
  0,4,0 4) *
  0,5,0 5) *
  0,6,0 6) *
  0,7,0 7) *
[den@localhost ~]$

```

Рис. 8.1. Идентификаторы приводов CD/DVD

Если нужно произвести полную, а не быструю очистку, замените `blank=fast` на `blank=all`.

8.3. Программа mkisofs

Иногда нужно создать образ CD/DVD не с оригинального диска, а с каталогов файловой системы. Другими словами, у вас есть файлы и каталоги, которые вам нужно записать на CD/DVD. Технология CD/DVD не позволяет записывать файлы и каталоги непосредственно на носитель. Вам нужно создать каталог, в который поместить все файлы и каталоги, которые вы хотите записать на оптический диск, затем создать по этому каталогу ISO-образ, а потом записать его на болванку.

Скопируйте все необходимые вам файлы в каталог `~/cd`. Затем выполните команду:

```
mkisofs -r -jcharset koi8-r -o ~/cd.iso ~/cd
```

Данная команда создаст по каталогу `~/cd` файл образа `cd.iso` и поместит в ваш домашний каталог. Обратите внимание на кодировку локализованной версии: сейчас используется `koi8-r`. Если у вас другая кодировка, например `utf8`, вы должны указать ее:

```
mkisofs -r -jcharset utf8 -o ~/cd.iso ~/cd
```


Указание кодировки необходимо для правильного отображения русскоязычных имен файлов и каталогов под управлением MS Windows.

После создания ISO-образа его нужно записать на носитель с помощью `cdrecord`, как было показано ранее. После записи не забудьте удалить образ, чтобы он не занимал места на диске.

Существует способ записи каталога на CD/DVD без создания промежуточного ISO-образа. Для этого используется команда:

```
mkisofs -jcharset кодировка /каталог | cdrecord -опции
```

8.4. Преобразование образов дисков

Иногда нужно записать образ диска, созданный в другой программе, формат которого отличается от ISO9660. Наиболее часто встречаются образы дисков в форматах `.img`, `.bin`, `.cue`, `.nrg`, `.ccd`.

Если у файла образа "расширение" (в Linux нет понятия "расширение", поэтому данное слово взято в кавычки) `img`, то это еще не означает, что формат образа ISO9660. Одни программы, например `k3b`, действительно создают образ в формате ISO9660 и записывают его в файл с "расширением" `img`, а другие программы могут записывать в файл с таким же расширением образы диска в собственных форматах.

Файлы `.bin/.cue` можно записать на диск с помощью программы `cdrdao` или преобразовать в ISO с помощью `bchunk`.

`Nero` записывает образы диска в формате `.nrg`, который можно преобразовать в ISO с помощью программы `nrg2iso`. Если вам нужно открыть NRG-образ, дабы просмотреть, что в нем, вы это можете сделать с помощью команды:

```
mount -t udf,iso9660 -o loop,ro,offset=307200 файл.nrg  
точка_монтирования
```

Образ в формате CloneCD (`ccd`) можно преобразовать в ISO с помощью программы `ccd2iso`.

8.5. Создание и монтирование файлов с файловой системой

Иногда (например, для создания собственного мини-дистрибутива) нужно создать файл, содержащий собственную файловую систему. Первым делом нужно создать пустой файл, потом создать в нем файловую систему, а затем подмонтировать данный файл к корневой файловой системе. Все это можно сделать с помощью трех команд:

```
# dd if=/dev/zero of=/file.fs bs=1k count=100000  
# mkfs.ext2 -F /file.fs  
# mount -t ext2 -o loop file.fs /mnt/disk
```

Первая команда создает пустой файл размером почти 100 Мбайт (100 000 Кбайт), вторая команда создает файловую систему типа ext2 в этом файле, третья монтирует файл к каталогу /mnt/disk.

8.6. Создание файла подкачки

Оперативная память — это очень критичный для Linux ресурс. Даже более критичный, чем частота процессора, поэтому нехватка оперативной памяти очень остро ощущается в Linux. Иногда работать становится просто невыносимо.

При установке Linux создается раздел подкачки, который используется, если системе не хватает оперативной памяти — на него сгружается неиспользуемая в данный момент информация, а в оперативную память с жесткого диска подгружаются необходимые процессору данные. Ясно, что система с разделом подкачки работает медленнее, чем с модулем оперативной памяти, но все же она работает быстрее и стабильнее, нежели вообще без раздела подкачки.

Если вы пожадничали и при установке Linux создали маленький раздел подкачки, делу можно помочь даже без переразметки жесткого диска. Мы можем создать файл подкачки, который будет использоваться в паре с разделом подкачки.

Сейчас мы создадим файл /swap_file размером 128 Мбайт:

```
# dd if=/dev/zero of=/swap_file bs=1k count=131072
```

Файл /swap_file пока еще нельзя назвать файлом подкачки, поскольку мы его не отформатировали как файл подкачки. Сделаем это:

```
# mkswap /swap_file 131072
```

Теперь осталось активировать только что созданный файл подкачки:

```
# swapon /swap_file
```

Последнюю команду нужно добавить в файл /etc/rc.d/rc.sysinit (или в /etc/rc.local в Debian/Ubuntu) для того, чтобы не вводить ее при каждом запуске системы.

8.7. Создание файловой системы

В разд. 8.5 с помощью команды `mkfs.ext2` мы создали файловую систему в файле. С помощью этой команды мы можем создать файловую систему на разделе жесткого диска, например: `mkfs.ext2 /dev/sda1`.

Вообще, создать файловую систему нужного типа (если эта файловая система поддерживается ядром вашей системы) можно с помощью команды `mkfs.<имя_файловой_системы>`, например:

```
mkfs.ext3
```

```
mkfs.vfat
```

```
mkfs.reiserfs
```

Подробнее прочитать об этом можно, введя команду

```
man mkfs.<ИМЯ_ФАЙЛОВОЙ_СИСТЕМЫ>
```

8.8. Проверка и восстановление файловой системы

Для проверки файловой системы используется программа `fsck`. Использовать ее нужно так:

```
fsck <раздел>
```

Например,

```
fsck /dev/sda5
```

Перед использованием этой команды нужно размонтировать проверяемую файловую систему. Если нужно проверить корневую файловую систему, то следует загрузиться с LiveCD и запустить `fsck` для проверки нужного раздела.

Если же жесткий диск "посыпался", т. е. появились "плохие" блоки, нужно, не дожидаясь полной потери данных, выполнить следующие действия:

1. Выполнить команду `fsck -с <раздел>` (данная команда пометит "плохие" блоки).
2. Сделать резервную копию всех важных данных.
3. Отправиться в магазин за новым жестким диском и перенести данные со старого жесткого диска на новый. Проверить жесткий диск на наличие плохих секторов можно программой `badblocks`.

ПРИМЕЧАНИЕ

Программа `fsck` может проверять не только файловые системы `ext2/ext3`. Для проверки, например `vfat`, можно использовать команду `fsck.vfat <раздел>`.

Для восстановления "упавшей" таблицы разделов можно использовать программу `gpart`. Только используйте ее осторожно и внимательно читайте все сообщения, выводимые программой.

8.9. Смена корневой файловой системы. Команда *chroot*

Предположим, мы установили Windows после установки Linux, и программа установки Windows перезаписала начальный загрузчик. Теперь Windows загружается, а Linux — нет. Что делать? Нужно загрузиться с LiveCD (подробнее о LiveCD мы поговорим в последней части книги) и выполнить команду:

```
# chroot <раздел, содержащий корневую файловую систему>
```

Например, если Linux был установлен в раздел `/dev/hda5`, то нужно ввести команду:

```
# chroot /dev/hda5
```

Данная команда сменит корневую файловую систему, т. е. вы загрузите ядро Linux с LiveCD, а затем сделаете подмену корневой файловой системы. Вам останется только ввести команду записи загрузчика (например, `lilo`) для восстановления начального загрузчика.

8.10. Работа с журналом файловой системы

Журналируемая файловая система имеет три режима работы: `journal`, `ordered` и `writeback`. Первый режим является наиболее медленным, но он позволяет минимизировать потери ваших данных в случае сбоя системы (или отключения питания). В этом режиме в системный журнал записывается все, что только можно, что позволяет максимально восстановить файловую систему в случае сбоя.

В последовательном режиме (`ordered`) в журнал заносится информация только об изменении метаданных (служебных данных файловой системы). Данный режим используется по умолчанию и является компромиссным вариантом между производительностью и отказоустойчивостью.

Самым быстрым является режим обратной записи (`writeback`). Но использовать его я вам не рекомендую, поскольку особого толку от него не будет. Проще тогда уже при установке Linux выбрать файловую систему `ext2` вместо `ext3`.

Если отказоустойчивость для вас на первом месте — выбирайте режим `journal`, во всех остальных случаях лучше выбрать `ordered`. Выбор режима осуществляется редактированием файла `/etc/fstab`. Например,

```
# режим ordered используется по умолчанию,  
# поэтому ничего указывать не нужно  
/dev/sda1 / ext3 defaults 1 0  
# на этом разделе важные данные, используем режим journal  
/dev/sda2 /var ext3 data=journal 1 0  
# здесь ничего важного нет, режим outback  
/dev/sda3 /opt ext3 data=writeback 0 0
```

После изменения этого файла выполните команду:

```
# mount -a
```

Данная команда заново смонтирует все файловые системы, чтобы изменения вступили в силу.

8.11. Монтирование NTFS

Как уже было отмечено в этой книге, Linux умеет только читать информацию из NTFS-разделов, но не умеет записывать информацию в такие разделы. Если при перекомпиляции ядра включить опцию записи информации в NTFS-разделы, то поддержка записи будет, но очень ограниченная.

Существуют средства, позволяющие добавить поддержку записи NTFS-разделов. Самым оптимальным подобным средством является модуль `ntfs-3g`. Модуль `ntfs-3g` доступен по интернет-адресу <http://www.ntfs-3g.org/>. На этом сайте доступны как исходные коды, так и уже собранные пакеты для разных дистрибутивов. Все, что вам нужно, — это установить скачанный пакет:

```
# rpm -i ntfs-3g*
```

или

```
sudo dpkg -i ntfs-3g*
```

После того как пакет установлен, можно примонтировать NTFS-раздел:

```
# mount -t ntfs-3g раздел точка_монтирования
```

Понятно, что вам нужно указать ваш раздел и вашу точку монтирования. Если нужно обеспечить автоматическое монтирование NTFS-раздела, тогда в `/etc/fstab` нужно добавить запись, подобную этой:

```
раздел каталог ntfs-3g defaults,nls=utf8,umask=007,gid=46
```

У модуля `ntfs-3g` все равно есть ограничения, например вы не можете изменять сжатые файлы (хотя можно их читать), и нельзя читать зашифрованные файлы.

Если вы не можете смонтировать NTFS-раздел с помощью опции `ntfs-3g`, то, вероятнее всего, он был неправильно размонтирован (например, работа Windows не была завершена корректно). В этом случае для монтирования раздела нужно использовать опцию `-o force`, например:

```
sudo mount -t ntfs-3g /dev/sdb1 /media/usb -o force
```

После этого раздел должен монтироваться нормально. Чтобы убедиться в этом, нужно размонтировать раздел и смонтировать его заново, но без опции `-o force`.

8.12. Установка скорости CD/DVD

Программа `hdparm` позволяет ограничить скорость оптического привода (CDROM/DVDROM). Иногда нужно ограничить скорость, чтобы информация была считана без ошибок (как правило, если поверхность носителя информации немного повреждена). Рассмотрим команду ограничения скорости:

```
# hdparm -q -E<множитель> <устройство>
```

Множитель — это и есть скорость, например 1 соответствует скорости 150 Кбит/с для CD, 1385 Кбит/с для DVD.

Чтобы установить вторую (2, 300 Кбит/с) скорость чтения для CD, используется команда:

```
# hdparm -q -E2 /dev/cdrom
```

Для ограничения скорости DVD можно использовать следующую команду:

```
# hdparm -q -E1 /dev/dvd
```

8.13. Псевдофайловая система /proc

Виртуальная (псевдофайловая) система /proc — это специальный механизм, позволяющий посылать информацию ядру, модулям и процессам (кстати, потому данная файловая система так и называется: *proc* — это сокращение от англ. *process*). Также, используя /proc, вы можете получать информацию о процессах и изменять параметры ядра и его модулей "на лету". Для этого в /proc есть файлы, позволяющие получать информацию о системе, ядре или процессе, и есть файлы, с помощью которых можно изменять некоторые параметры системы. Первые файлы мы можем только просмотреть, а вторые — просмотреть и, если нужно, изменить.

Просмотреть информационный файл можно командой `cat`:

```
cat /proc/путь/⟨название_файла⟩
```

Записать значение в один из файлов *proc* можно так:

```
echo "данные" > /proc/путь/⟨название_файла⟩
```

8.13.1. Информационные файлы

В табл. 8.1 представлены некоторые (самые полезные) информационные *proc*-файлы: с их помощью вы можете получить информацию о системе.

Таблица 8.1. Информационные *proc*-файлы

Файл	Описание
/proc/version	Содержит версию ядра
/proc/cmdline	Список параметров, переданных ядру при загрузке
/proc/cpuinfo	Информация о процессоре
/proc/meminfo	Информация об использовании оперативной памяти (почти то же, что и команда <code>free</code>)
/proc/devices	Список устройств
/proc/filesystems	Файловые системы, которые поддерживаются вашей системой
/proc/mounts	Список подмонтированных файловых систем

Таблица 8.1 (окончание)

Файл	Описание
/proc/modules	Список загруженных модулей
/proc/swaps	Список разделов и файлов подкачки, которые активны в данный момент

8.13.2. Файлы, позволяющие изменять параметры ядра

Каталог /proc/sys/kernel содержит файлы, с помощью которых вы можете изменять важные параметры ядра. Конечно, все файлы мы обсуждать не будем, а рассмотрим лишь те, которые используются на практике (табл. 8.2).

Таблица 8.2. Файлы каталога /proc/sys/kernel

Файл	Каталог
/proc/sys/kernel/ctrl-alt-del	Если данный файл содержит значение 0, то при нажатии комбинации клавиш <Ctrl>+<Alt>+ будет выполнена так называемая "мягкая перезагрузка", когда управление передается программе init, и последняя "разгружает" систему, как при вводе команды <code>reboot</code> . Если этот файл содержит значение 1, то нажатие комбинации клавиш <Ctrl>+<Alt>+ равносильно нажатию кнопки Reset. Сами понимаете, значение 1 устанавливать не рекомендуется
/proc/sys/kernel/domainname	Здесь находится имя домена, например dkws.org.ua
/proc/sys/kernel/hostname	Содержит имя компьютера, например <code>den</code>
/proc/sys/kernel/panic	При критической ошибке ядро "впадает в панику" — работа системы останавливается, а на экране красуется надпись "kernel panic" и выводится текст ошибки. Данный файл содержит значение в секундах, которое система будет ждать, пока пользователь прочитает это сообщение, после чего компьютер будет перезагружен. Значение 0 (по умолчанию) означает, что перезагружать компьютер вообще не нужно
/proc/sys/kernel/printk	Данный файл позволяет определить важность сообщения об ошибках. По умолчанию файл содержит значения 6 4 1 7. Это означает, что сообщения с уровнем приоритета 6 и ниже (чем ниже уровень, тем выше важность сообщения) будут выводиться на консоль. Для некоторых сообщений об ошибках уровень приоритета не задается. Тогда нужно установить уровень по умолчанию. Это как раз и есть второе значение — 4.

Таблица 8.2 (окончание)

Файл	Каталог
	Третье значение — это номер самого максимального приоритета, а последнее число — значение по умолчанию для первого значения. Обычно изменяют только первое значение, дабы определить, какие значения должны быть выведены на консоль, а какие — попасть в журнал демона syslog

8.13.3. Файлы, изменяющие параметры сети

В каталоге `/proc/sys/net` вы найдете файлы, изменяющие параметры сети (табл. 8.3).

Таблица 8.3. Файлы каталога `/proc/sys/net`

Файл	Описание
<code>/proc/sys/net/core/message_burst</code>	Опытные системные администраторы используют этот файл для защиты от атак на отказ (DoS). Один из примеров DoS-атаки — когда система заваливается сообщениями атакующего, а полезные сообщения системой игнорируются, потому что она не успевает реагировать на сообщения злоумышленника. В данном файле содержится значение времени (в десятых долях секунды), необходимое для принятия следующего сообщения. Значение по умолчанию — 50 (5 секунд). Сообщение, попавшее в "перерыв" (в эти 5 секунд), будет проигнорировано
<code>/proc/sys/net/core/message_cost</code>	Чем выше значение в этом файле, тем больше сообщений будет проигнорировано в перерыв, заданный файлом <code>message_burst</code>
<code>/proc/sys/net/core/netdev_max_backlog</code>	Задаёт максимальное число пакетов в очереди. По умолчанию 300. Используется, если сетевой интерфейс передает пакеты быстрее, чем система может их обработать
<code>/proc/sys/net/core/optmem_max</code>	Задаёт максимальный размер буфера для одного сокета

8.13.4. Файлы, изменяющие параметры виртуальной памяти

В каталоге `/proc/sys/vm` вы найдете файлы, с помощью которых можно изменить параметры виртуальной памяти:

- ♦ в файле `buffermem` находятся три значения (разделяются пробелами): минимальный, средний и максимальный объем памяти, которую система может использовать для буфера. Значения по умолчанию: 2 10 60;

- ◆ в файле `kswapd` тоже есть три значения, которые можно использовать для управления подкачкой:
 - ◆ первое значение задает максимальное количество страниц, которые ядро будет пытаться переместить на жесткий диск за один раз;
 - ◆ второе значение — минимальное количество попыток освобождения той или иной страницы памяти;
 - ◆ третье значение задает количество страниц, которые можно записать за один раз. Значения по умолчанию: 512 32 8.

8.13.5. Файлы, позволяющие изменить параметры файловых систем

Каталог `/proc/sys/fs` содержит файлы, изменяющие параметры файловых систем. В частности:

- ◆ файл `file-max` задает максимальное количество одновременно открытых файлов (по умолчанию 4096);
- ◆ в файле `inode-max` содержится максимальное количество одновременно открытых индексных дескрипторов (максимальное значение также равно 4096);
- ◆ в файле `super-max` находится максимальное количество используемых суперблоков;

ПРИМЕЧАНИЕ

Поскольку каждая файловая система имеет свой суперблок, легко догадаться, что количество подмонтируемых файловых систем не может превысить значение из файла `super-max`, которое по умолчанию равно 256, чего в большинстве случаев вполне достаточно. Наоборот, можно уменьшить это значение, дабы никто не мог подмонтировать больше файловых систем, чем нужно (если монтирование файловых систем разрешено обычным пользователям).

- ◆ в файле `super-ng` находится количество открытых суперблоков в текущий момент. Данный файл нельзя записывать, его можно только читать.

8.13.6. Как сохранить изменения?

Итак, вы изменили некоторые параметры системы с помощью `/proc`, и теперь вам нужно их сохранить. Для этого их надо прописать в файле `/etc/sysctl.conf`. Вот только формат этого файла следующий: надо отбросить `/proc/sys/` в начале имени файла, а все, что останется, записать через точку, а затем через знак равенства указать значение параметра. Например, для изменения параметра `/proc/sys/vm/buffermem` нужно в файле `etc/sysctl.conf` прописать строку:

```
vm.buffermem = 2 11 60
```

Если в вашем дистрибутиве нет файла `/etc/sysctl.conf`, тогда пропишите команды вида `echo "значение" > файл` в сценарий инициализации системы.

ГЛАВА 9



Поддержка RAID в Linux

9.1. Что такое RAID

RAID (Redundant Array of Independent Disk) — матрица независимых дисков с избыточностью. Массивы RAID обеспечивают более надежное хранение ваших данных. Как? Например, у нас есть два винчестера. Мы объединим эти два винчестера в один RAID-массив. Все, что будет записано на первый винчестер, автоматически продублируется на второй. Если с первым винчестером что-то случится (у жестких дисков есть свойство периодически выходить со строя, это может произойти раз в 5 лет, но все равно терять данные не хочется), то мы сможем восстановить свои данные со второго винчестера. Описанный способ является далеко не единственным способом организации RAID-массивов. Алгоритм работы RAID-массива зависит от уровня RAID (табл. 9.1).

Таблица 9.1. Уровни RAID-массивов

Уровень	Алгоритм работы
0	Предназначен не для обеспечения надежности, а для увеличения суммарного объема диска. Предположим, у нас есть два винчестера по 200 Гбайт. Объединив их в RAID-массив, мы получим один диск на 400 Гбайт. Очень удобно, если мы работаем с видео (имеется в виду профессиональный видеомонтаж, а не просто просмотр фильмов)
1	Простое зеркальное копирование, как было описано ранее. Все что записано на первый жесткий диск, будет продублировано на второй. Желательно, чтобы диски были одного размера. Если это не так, то размер RAID-массива будет равен размеру меньшего диска
2	Используется метод битового чередования блоков данных, при этом добавляются коды коррекции ошибок
3	Усовершенствованный уровень 2: коды коррекции ошибок записываются на другой диск
4	Усовершенствованный уровень 3: практически то же самое, но изменен метод записи контрольных кодов

Таблица 9.1 (окончание)

Уровень	Алгоритм работы
5	Самый оптимальный уровень по соотношению "производительность/надежность". Использует контрольные суммы, и данные записываются вместе с контрольными кодами на все диски. Если с одним из дисков что-то случилось, то данные можно восстановить с помощью контрольной суммы. Общий размер массива вычисляется по формуле $M \times (N - 1)$, N — это количество дисков в массиве, а M — размер наименьшего диска. Минимальное значение $N = 3$
6	Представляет собой усовершенствованный уровень 5. Он надежнее, чем RAID 5, но менее производительный. Скорость чтения информации примерно такая же, как и в случае с RAID 5, но скорость записи обычно ниже на 40—50%, не говоря уже о медленном восстановлении данных. Однако RAID 6 позволяет восстановить данные даже в случае выхода из строя двух жестких дисков. Довольно дорог в реализации, поскольку требует как минимум четыре жестких диска, а полезная емкость равна $N - 2$, где N — это количество дисков (два жестких диска отводятся для хранения контрольных сумм). Если у вас в массиве четыре жестких диска по 200 Гбайт каждый, то полезная емкость составит только 400 Гбайт из 800. Кратко RAID 6 можно охарактеризовать так: дорогой, медленный, но надежный. Из-за своей дороговизны и низкой производительности применяется редко. Однако его можно использовать, если надежность превыше всего
RAID 10 (он же RAID 1+0)	Диски массива парами объединяются в "зеркала" (уровень RAID 1), далее зеркала объединяются в общий массив с чередованием данных (RAID 0). Отсюда и название уровня — RAID 10 или RAID 1+0. В массив RAID 10 можно объединить только парное количество дисков — от 4 до 16. Довольно надежен, поскольку используются зеркала, но в то же время быстр, поскольку от RAID 0 унаследована производительность. Полезная емкость — в два раза меньше от общей емкости массива. Более предпочтителен, чем RAID 6 там, где нужна производительность и надежность
RAID 1E	Расширенная (E — Enhanced) версия RAID 1. Данные чередуются блоками по всем дискам массива, а затем еще раз чередуются со сдвигом на один диск. Данный уровень позволяет объединять от 3 до 16 дисков. По надежности примерно такой, как RAID 10, но имеет большую полезную емкость и еще большую производительность
RAID 1E0	Позволяет объединять в нулевой массив массивы уровня RAID 1E. Можно объединить от 3 до 60 (!) дисков. Преимуществ в скорости нет — наоборот, данный массив работает медленнее, чем RAID 1E, преимуществ в надежности тоже нет — из-за сложной реализации менее надежный, чем RAID 1E, но зато этот массив позволяет объединить в один большой массив до 60 дисков

На практике обычно используются уровни 0, 1, 5. Некоторые материнские платы поддерживают RAID-массивы на аппаратном уровне. Раньше поддержкой RAID-массивов обладали только дорогие серверные материнские платы. Сейчас поддержку RAID можно встретить в относительно недорогих материнских платах

среднего ценового диапазона. О создании и поддержке аппаратных RAID-массивов вы можете прочитать в документации по материнской плате вашего компьютера.

Кроме уровней RAID 1—RAID 6, описанных в стандарте, некоторые производители создают комбинированные уровни: RAID 10 (1+0), RAID 15 (1+5), RAID 50 (5+0) и т. д. Суть таких комбинаций заключается в следующем. RAID 10 — это комбинация уровней 1 и 0, 15 — это уровни 1+5, т. е. зеркало "пятерок" и т. д. Такие комбинированные уровни сочетают в себе преимущества и недостатки своих "родителей". Например, уровень RAID 50 — практически то же, что и RAID 5, но зато быстрее, чем RAID 5.

Кроме обычных уровней, есть еще и расширенные уровни RAID, к наименованию уровня добавляется буква E, например RAID 1E, RAID 5E и т. д. Это усовершенствованные версии базовых уровней. Чтобы не описывать каждый такой уровень, лучше рассмотрим таблицу с общими характеристиками самых часто используемых уровней RAID (табл. 9.2), т. е. именно с характеристиками тех уровней, которые вы будете использовать на практике.

Таблица 9.2. Характеристики уровней RAID

Уровень	Избыточность	Мин.	Макс.	Чтение	Запись	Емкость
0	—	1	16	10	10	100
1	+	2	2	8	8	50
5	+	3	16	10	7	67—94
6	+	4	16	10	7	50—88
10 (1+0)	+	4	16	9	9	50
15	+	6	60	10	7	33—48
1E	+	3	16	8	8	50
1E0	+	2	60	8	8	50
50	+	6	60	10	7	67—94
5E	+	4	16	10	7	50—88
5EE	+	4	16	10	7	50—88
00	—	2	60	10	10	100

9.2. Программные RAID-массивы

В Linux можно создавать программные RAID-массивы, даже если материнская плата вашего компьютера не поддерживает их на аппаратном уровне. У программных массивов есть один маленький недостаток — они работают немного медленнее

аппаратных, но у программных RAID-массивов есть и одно неоспоримое преимущество — поскольку обработка данных происходит на программном уровне, совсем необязательно, чтобы жесткие диски, входящие в состав массива, были совместимы между собой. Например, можно создать массив уровня 5, который будет состоять из дисков EIDE, SATA и SCSI — эти три разных интерфейса объединить в аппаратный массив просто невозможно.

Поддержка RAID-массивов встроена в ядро по умолчанию, поэтому вам даже не придется его перекомпилировать. При загрузке Linux вы должны увидеть следующие строки:

```
md: md driver 0.90.2 MAX_MD_DEVS=256, MD_SB_DISKS=27
md: bitmap version 3.39
...
md: Autodetecting RAID arrays.
md: autorun ...
md: ... autorun DONE.
```

Появление этих строк (если при загрузке вы не успели их заметить, введите команду `dmesg`) означает, что ядро поддерживает RAID. Не поддерживать RAID могут лишь компактные ядра некоторых дистрибутивов, которые мы здесь рассматривать не будем. Fedora, ASPLinux, Mandriva, ALT Linux поддерживают RAID-массивы по умолчанию.

Если же поддержки RAID почему-то в вашем дистрибутиве не оказалось, то включить ее можно в разделе **Block device** конфигулятора `make menuconfig`. После этого нужно будет перекомпилировать ядро. После загрузки с новым ядром следует установить пакет `raidtools`, содержащий необходимые нам команды `raidhotadd`, `raidhotremove`, `mkraid`. Последняя команда создает RAID-массив, первая добавляет в него диск, а вторая — удаляет диск из массива.

9.3. Создание программных массивов

Как уже было отмечено, на практике чаще всего используются уровни 0, 1, 5. Уровни 1 и 5 пригодятся на серверах, где нужно обеспечить надежность хранения данных, а уровень 0 — на рабочей станции или домашнем компьютере, если надо создать один большой раздел для хранения данных — например, при видеомонтаже раздел в несколько сотен гигабайт будет совсем не лишним.

Начнем с массива уровня 1. Создайте два раздела типа **Linux raid autodetect** (тип `fd` в программе `fdisk`). Разделы можно создать как на одном, так и на разных дисках. Лучше, если вы создадите разделы на разных дисках — так будет надежнее. Забегая наперед, отмечу, что в массив RAID можно добавить только разделы типа `fd`. Обычные Linux-разделы вы не сможете добавить в массив.

После этого отредактируйте файл `/etc/raidtab` (листинг 9.1).

Листинг 9.1. Файл /etc/raidtab для уровня 1

```
# Имя устройства RAID-массива
raiddev /dev/md0
# Указываем уровень
raid-level 1
# Число дисков в RAID-массиве
nr-raid-disk 2
# Число дисков "на подхвате" — они будут использованы, если один из дисков
# выйдет из строя
nr-spare-disk 0
# Другие параметры
chunk-size 8
# Всегда указывайте эту опцию, иначе массив сразу не запустится
# после его остановки командой raidstop.
persistent-superblock 1

# Первый диск RAID
device /dev/sdc3
raid-disk 0
# Второй диск RAID
device /dev/sda7
raid-disk 1
```

Теперь нужно создать устройство /dev/md0, которое мы упомянули в конфигурационном файле. Для этого используем команду:

```
# mkraid /dev/md0
```

После этого вы можете использовать устройство /dev/md0 как самый обычный жесткий диск — его можно форматировать, монтировать и т. д.

Конфигурационный файл для уровня 5 выглядит немного иначе (листинг 9.2).

Листинг 9.2. Файл /etc/raidtab для уровня 5

```
raiddev /dev/md0
raid-level 5
nr-raid-disk 3
nr-spare-disk 0
persistent-superblock 1
parity-algorithm left-symmetric
chunk-size 64
device /dev/sdc1
raid-disk 0
```

```
device /dev/sda7
raid-disk 1
device /dev/sdd3
raid-disk 2
```

Если один из дисков вышел из строя, то нужно использовать команду `raidhotremove`, чтобы извлечь его из массива. Затем на другом жестком диске создать разделы для RAID-массива (размер и количество разделов должны быть такими же, как у извлеченного диска), а затем добавить новый диск командой `raidhotadd` (см. далее).

ПРИМЕЧАНИЕ

Получить информацию о работе RAID можно командой `/proc/mdstat`.

Теперь рассмотрим реализацию уровня RAID 0. Для реализации этого уровня вам понадобятся как минимум два устройства. Предположим, что у нас есть два раздела на разных жестких дисках, которые мы хотим объединить воедино — `/dev/sdc1` и `/dev/sda7`. Файл `/etc/raidtab` для уровня RAID 0 представлен в листинге 9.3.

Листинг 9.3. Файл `/etc/raidtab` для уровня 0

```
raiddev /dev/md0
raid-level      0
nr-raid-disks   2
persistent-superblock 1
chunk-size      4
device          /dev/sdc1
raid-disk       0
device          /dev/sda7
raid-disk       1
```

После корректировки файла выполните команду:

```
# mkraid /dev/md0
```

Все, теперь устройство `/dev/md0` готово к форматированию (созданию файловой системы), монтированию и использованию. Напомню, что RAID 0 не поддерживает избыточности, поэтому если "умрет" один из дисков, входящий в его состав, весь массив падет смертью храбрых.

ПРИМЕЧАНИЕ

Никогда не изменяйте разметку жестких дисков, входящих в состав RAID. При этом может измениться нумерация разделов, и тогда весь массив RAID перестанет функционировать!

9.4. Использование RAID-массива

Спрашивается, что делать после создания RAID-массива? Итак, у нас есть устройство `/dev/md0`. Начнем с создания файловой системы.

Для создания файловой системы `ext2` (`ext3` использовать нет смысла, поскольку RAID надежнее журнала `ext3`) на `/dev/md0` можно использовать команду:

```
# mke2fs -b 4096 -R stride=8 /dev/md0
```

ПРИМЕЧАНИЕ

Опцию `stride=8` нужно использовать только для RAID уровней 4 и 5 (она позволяет повысить производительность). Для остальных уровней ее лучше не указывать. Опция `-b` задает размер блока (в данном случае 4096 байтов).

Однако файловая система `ext2` не является ультрасовременной, поэтому вместо нее лучше создать `ReiserFS`:

```
# mkreiserfs /dev/md0
```

После того как файловая система создана, устройство `/dev/md0` можно подмонтировать, например:

```
# mkdir /mnt/raid
```

```
# mount /dev/md0 /mnt/raid
```

Для автоматического монтирования (при каждой загрузке системы) нашего массива в файл `/etc/fstab` нужно добавить строки:

```
/dev/md0 /mnt/raid reiserfs defaults 0 0
```

ПРИМЕЧАНИЕ

Если вы используете файловую систему `ext2`, то приведенную здесь строку нужно изменить так:

```
/dev/md0 /mnt/raid ext2 defaults 0 0
```

Для "горячего" добавления в массив еще одного раздела нужно использовать команду `raidhotadd`:

```
# raidhotadd /dev/md0 /dev/sdXn
```

Остановить массив можно командой:

```
# raidstop /dev/md0
```

Для запуска массива используется команда:

```
# raidstart /dev/md0
```

9.5. Сбой и его имитация

Предположим, один из жестких дисков, входящих в состав массива, вышел из строя. Нужно завершить работу системы и выключить питание компьютера. Затем извлечь из системы вышедший из строя жесткий диск и установить новый.

Следующий шаг — создание раздела типа fd. Размер раздела должен быть таким же, как и на старом жестком диске. В заключение надо ввести команду:

```
# raidhotadd /dev/mdX /dev/sdX
```

Выполнение этой команды займет некоторое время, поскольку будет запущена реконструкция данных.

Вам интересно, как RAID справится с выходом диска из строя? Тогда выключите компьютер и отключите один из жестких дисков, входящих в состав массива. Так вы имитировали сбой. Теперь вам остается действовать, как при сбое.

ГЛАВА 10



Запись CD/DVD в Linux

10.1. CD/DVD — оптимальное решение для резервных копий

Ранее на предприятиях для создания резервных копий использовались стримеры, поскольку на дискеты много информации не запишешь, а использовать жесткие диски для создания резервных копий — дорого.

Стримеры позволяли записывать на магнитную ленту (которая стоила копейки) довольно большие объемы информации. С распространением и удешевлением CD/DVD стримеры постепенно отошли в прошлое. Диски CD/DVD намного надежнее магнитной ленты, стоят дешево и позволяют записывать большие объемы информации (от 4,5 до 18 Гбайт для DVD). В этой главе мы рассмотрим процесс записи CD/DVD, а о стратегии резервного копирования мы поговорим позже.

10.2. Форматы и маркировка DVD-дисков

DVD — это более "продвинутая" версия CD. При разработке DVD решили пойти не по качественному пути, а по количественному — просто "раздули" показатели. Ведь большая емкость DVD достигается за счет более плотной записи. Конечно, были разработаны более совершенные методы коррекции ошибок, дополнительные методы оптимизации дискового пространства, но суть от этого не меняется.

DVD-диски бывают:

- ◆ односторонними и однослойными (4,7 Гбайт, маркировка DVD-5);
- ◆ односторонними и двухслойными (8,54 Гбайт, маркировка DVD-9);
- ◆ двухсторонними и однослойными (9,4 Гбайт, маркировка DVD-10);
- ◆ двухсторонними и двухслойными (17 Гбайт, маркировка DVD-18).

Наиболее распространены диски DVD-5 и DVD-10. Диски DVD-9 встречаются реже, а DVD-18 — вообще редко, их сложно найти в продаже. Большинство проигрывателей умеют читать только односторонние диски. Для чтения второй стороны нужно перевернуть диск. Встречаются особо древние аппараты, которые даже и не подозревают о существовании двухслойных дисков — они предназначены для чтения самых первых DVD, которые вы сейчас практически не найдете.

Теперь поговорим о форматах DVD.

- ◆ DVD-ROM — базовый формат DVD, используется для массового производства дисков, например дисков с фильмами. DVD-ROM — это фундаментальный формат для остальных типов DVD, но в большей мере он относится к DVD-Video и DVD-Audio. Данный формат поддерживает файловые системы UDF и ISO9660 (как у обычных CD), однако не задает, как должны физически размещаться файлы. Порядок физического размещения задается спецификацией DVD-Audio и DVD-Video.
- ◆ DVD-Video — является логической надстройкой на DVD-ROM. Формат задает, как будут расположены файлы на диске. Как ясно из названия формата, DVD-Video предназначен для хранения фильмов. Кроме записи фильмов и звуковых потоков, сопровождающих фильмы, на такие диски можно записывать картинки (которые потом можно будет просматривать с помощью средств навигации DVD-проигрывателя), субтитры на разных языках, диалоговые окна. Конечно, на данный диск можно записать и любые другие файлы — они будут проигнорированы домашним DVD-проигрывателем, но зато к ним можно будет получить доступ, если вставить диск в компьютер. Вот что нужно помнить о DVD-Video:
 - ◆ на такой диск можно записать 133 минуты фильма со звуком (имеется в виду однослойный односторонний DVD). Если фильм не помещается на однослойный односторонний DVD, можно использовать DVD большей емкости;
 - ◆ имеется поддержка многоканального звука (до 8 каналов) — об этом мы говорили;
 - ◆ есть поддержка surround-звука — это отдельный канал для баса;
 - ◆ поддержка экранных форматов 4:3 (обычное телевидение) и 16:9 (широкоформатное видео);
 - ◆ защита от нелегального копирования;
 - ◆ поддержка регионов распространения (об этом мы уже говорили);
 - ◆ поддержка субтитров на 32 языках;
 - ◆ поддержка интерактивного управления.
- ◆ DVD-Audio — используется для записи очень качественного звука. Все мы знаем, что звук в формате MP3 при воспроизведении на профессиональном проигрывателе хуже, чем звук AudioCD. Так вот, звуковой поток формата DVD-Video намного лучше, чем AudioCD, а звучание DVD-Audio намного лучше, чем звучание звукового потока DVD-Video. Понимаете, насколько хорош этот формат? Появился он не так уж и давно — в 1999 г., правда, один год он существовал в лаборатории, ведь первые проигрыватели, поддерживающие этот

формат, появились только в 2000 г., а еще через год мир увидел первый коммерческий диск DVD-Audio. На сегодняшний день DVD-Audio является лучшим аудиоформатом. Высокое качество звучания достигается благодаря сжатию без потерь (алгоритм LPCM), т. е. все 4,7 Гбайт используются исключительно для звука, что позволяет сохранить оригинальное качество звучания.

- ◆ DVD-R — это записываемый диск DVD. Такой диск можно купить в любом магазине (не обязательно компьютерном) и записать на него свои данные. Записывать можете все, что хотите: музыку, документы, фильмы, картинки или все и сразу. Лишь бы у вас был привод, поддерживающий запись DVD. Существуют два типа DVD-R: для общего использования (который продается на каждом углу) и для продюсеров (DVD-Authoring). Отличие первого от второго заключается в том, что с помощью первого вы не сможете создать так называемый мастер-диск, который используется для тиражирования фильмов, а также не сможете использовать некоторые схемы защиты от нелегального копирования. В общем, обычный DVD-R предназначен сугубо для личного, а не коммерческого использования. Конечно, есть еще отличия, например DVD-Authoring стоит существенно дороже обычных DVD-R. С технической стороны, разница заключается в различной длине волны лазера. Для обычных DVD-R используется волна длиной 635 нм, а для "продюсерского" диска длина волны составляет 650 нм. Кроме того, для записи DVD-Authoring нужен специальный привод. Как правило, из-за разницы в длине волны обычные приводы DVD-RW не умеют записывать DVD-Authoring, а приводы, рассчитанные на запись DVD-Authoring, не могут записывать обычные DVD-R. Но оба типа приводов могут читать оба типа дисков. Данный факт нужно учитывать при покупке привода DVD-RW или при покупке "болванок" (если привод вы уже купили): нет смысла покупать более дорогой DVD-Authoring — все равно вы не сможете его записать. Не нужно думать, что на DVD-R можно записать только файлы. С помощью чистого DVD-R вы можете создать диск любого формата — DVD-Video, DVD-Audio, DVD-ROM, просто вы не сможете, еще раз повторюсь, использовать схемы защиты диска от нелегального копирования.
- ◆ DVD-RW и DVD-RAM — перезаписываемые DVD-диски. На такой диск вы можете записать информацию, потом стереть все, потом заново записать — принцип такой же, как и в случае с дисками CD-RW. Перезаписываемые диски маркируются "DVD-RW". Маркировка "DVD-RAM" применяется намного реже. Но если найдете диск с маркировкой "DVD-RAM" — покупайте. В чем отличие? А в том, что DVD-RW можно перезаписывать много раз, а DVD-RAM — очень много (сотни тысяч). DVD-RAM намного надежнее, чем DVD-RW. Но у DVD-RAM есть один недостаток: перезапись диска является очень медленным процессом. В среднем на запись диска нужно 1 час. Зато у DVD-RAM есть неоспоримое преимущество: для записи этих дисков не нужно создавать образ на жестком диске, можно сразу писать прямо на диск. Это очень важно — ведь не всегда на жестком диске есть 5 Гбайт (или более) свободного места. Если у вас есть хотя бы 200 Мбайт свободного места, вы сможете запи-

сать диск DVD-RAM полностью. С другой стороны, DVD-RAM можно использовать только на компьютере: нет DVD-проигрывателей, которые читают диски данного формата. Перезаписываемые диски не такие надежные, как DVD-R. Если вам нужно записать диск для многократного использования, например фильм, который вы потом одолжите всем своим друзьям, и они по несколько раз его посмотрят, то лучше записать его на DVD-R: есть вероятность, что, когда он вернется к вам, он все еще будет читаться. А вот если вам нужен диск для того, чтобы перенести файлы из офиса домой или наоборот, DVD-RW — лучшее решение. Не DVD-RAM, а именно DVD-RW. В случае с DVD-RAM вам нужно будет уходить с работы на час позже — пока запишется диск.

- ❖ DVD+R/+RW — новый формат получил "+" в своей маркировке, чтобы подчеркнуть свое отличие и превосходство над старыми форматами. Помните, что устаревшие приводы (и DVD-проигрыватели) не умеют читать диски этого формата, поэтому если вы покупали свой DVD до 2003 г. (или даже в 2003 г.), скорее всего, он не будет читать диски этого формата. Ведь этот формат появился в 2003 г. Что же касается старых приводов для записи DVD, то они могли записывать диски или только с "минусом" или только с "плюсом". Современные приводы "умеют" записывать оба формата. Преимущество данного формата заключается в более высокой скорости записи, например для DVD+R скорость записи на момент появления этого формата составляла 4×, в то время как обычные диски записывались максимум со скоростью 2×. Сейчас можно смело покупать диски и с "плюсом", и с "минусом". Современные проигрыватели и приводы нормально читают и записывают оба формата. Если же вам больше нравится "классика", покупайте диски DVD-R: они стоят немного дешевле DVD+R. Что же касается скорости, то сейчас передо мной на столе лежит диск DVD-R со скоростью записи 16×. А на крышке привода DVD-RW красуется надпись, что он может записывать диски как с "плюсом", так и с "минусом".

10.3. Программа k3b

В состав многих дистрибутивов входит программа k3b, предназначенная для записи CD- и DVD-дисков. Программа очень удобная и простая. При этом по возможностям ее можно сравнить с популярной Windows-программой Nero. Рекомендую использовать последние версии k3b, поскольку первые версии работали не очень стабильно. Я использую версию k3b, входящую в состав Mandriva.

ПРИМЕЧАНИЕ

В этой главе будут рассмотрены графические программы для прожига "болванок". Но в Linux есть программы без графического интерфейса, позволяющие записывать CD/DVD-диски. Эти программы были описаны в *главе 8*.

Сразу нужно отметить, что для поддержки DVD-дисков следует установить пакет k3b-dvd — без него вы не сможете работать с DVD.

Запустите программу k3b. В нижней части окна выберите предполагаемое действие (рис. 10.1):

- ◆ **Новый проект звукового CD;**
- ◆ **Новый проект CD с данными;**
- ◆ **Новый проект DVD с данными;**
- ◆ **Копировать компакт-диск.**

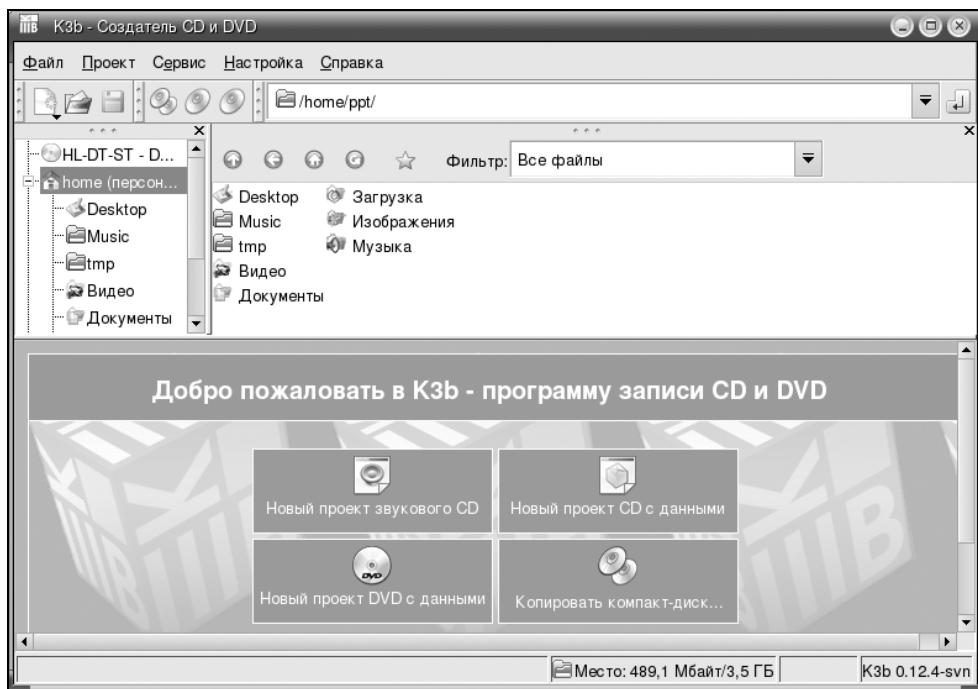


Рис. 10.1. Программа k3b — выбор действия

Сейчас мы попробуем записать CD с данными. Настоятельно рекомендую в первый раз использовать CD-RW, а не CD-R: если вы сделаете ошибку, CD-RW можно всегда стереть. А вот CD-R — только выбросить.

После выбора действия откроется рабочая область программы. В случае с Data CD она будет выглядеть, как на рис. 10.2.

В верхней части окна вы видите файловый менеджер, позволяющий выбрать файлы для записи на CD. Чтобы записать нужный вам файл на CD, просто перетащите его мышью в нижнюю область.

Теперь нажмите кнопку записи (рис. 10.3). Откроется окно, позволяющее установить параметры записи (рис. 10.4). Обычно нужно выбрать только скорость записи, не полагаясь на значение **Auto**.

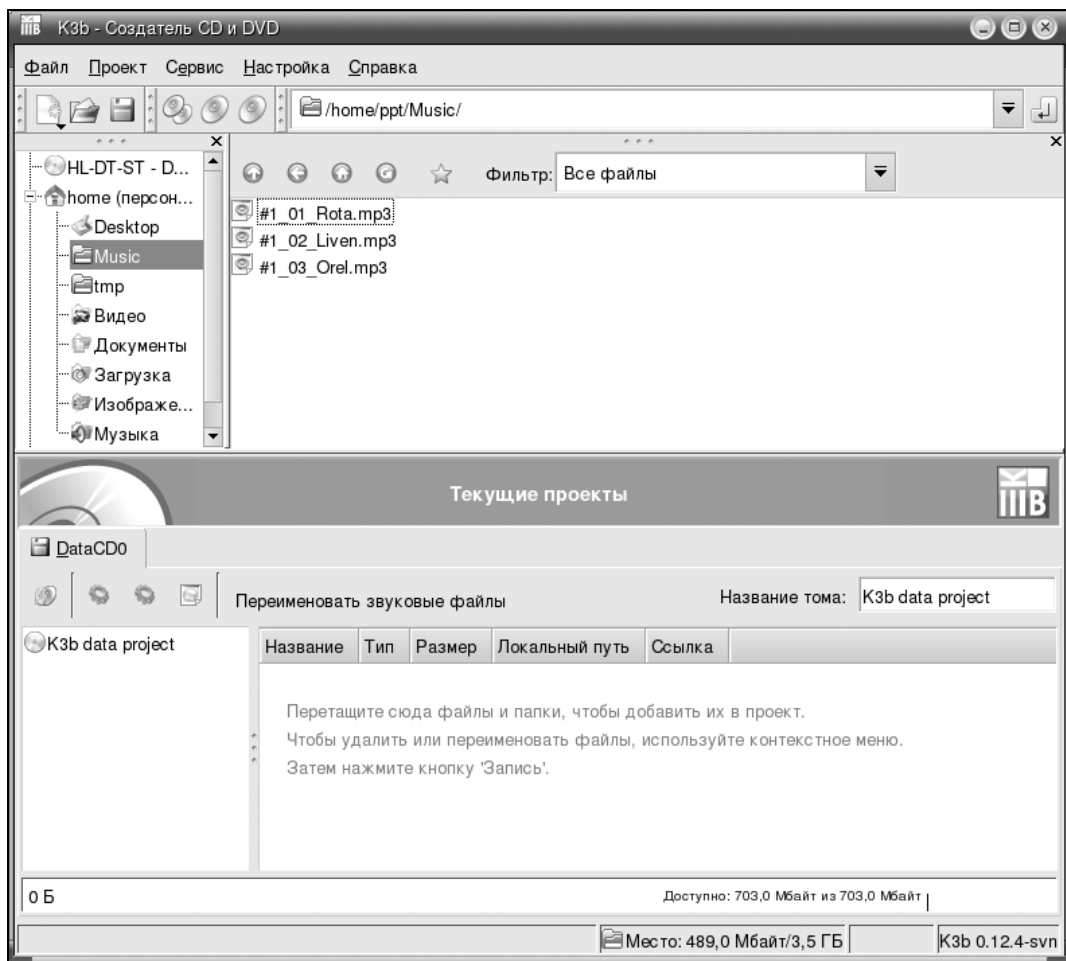


Рис. 10.2. Рабочая область k3b

Ошибочка вышла. Оказывается, в спешке вставил не тот диск (рис. 10.5). K3b успешно прошла тест: определила тип вставленного носителя и указала правильный целевой диск.

Если у вас есть уже записанный CD-RW с закрытой сессией (т. е. без возможности дозаписи), то с помощью команды **Сервис | Очистить CD-RW** можно очистить диск (рис. 10.6). Помните, что в случае, если вы закроете сессию, дописать информацию на диск CD-RW вы уже не сможете — вам придется его стирать, а если у вас CD-R, тогда это означает, что вы больше вообще не сможете на него записать. Закрытие сессии имеет смысл, если вы записали диск полностью или далее не планируете его использовать.

Теперь, когда установлен чистый CD-RW, можно продолжить прерванный процесс записи. Опять откройте диалог записи и нажмите кнопку **Записать** (рис. 10.7).

После окончания процесса записи вы увидите примерно такое окно (рис. 10.8). Обратите внимание на то, что k3b закрыл сессию, поэтому записать еще что-то на диск невозможно. Нужно только стереть его, а потом заново записать.

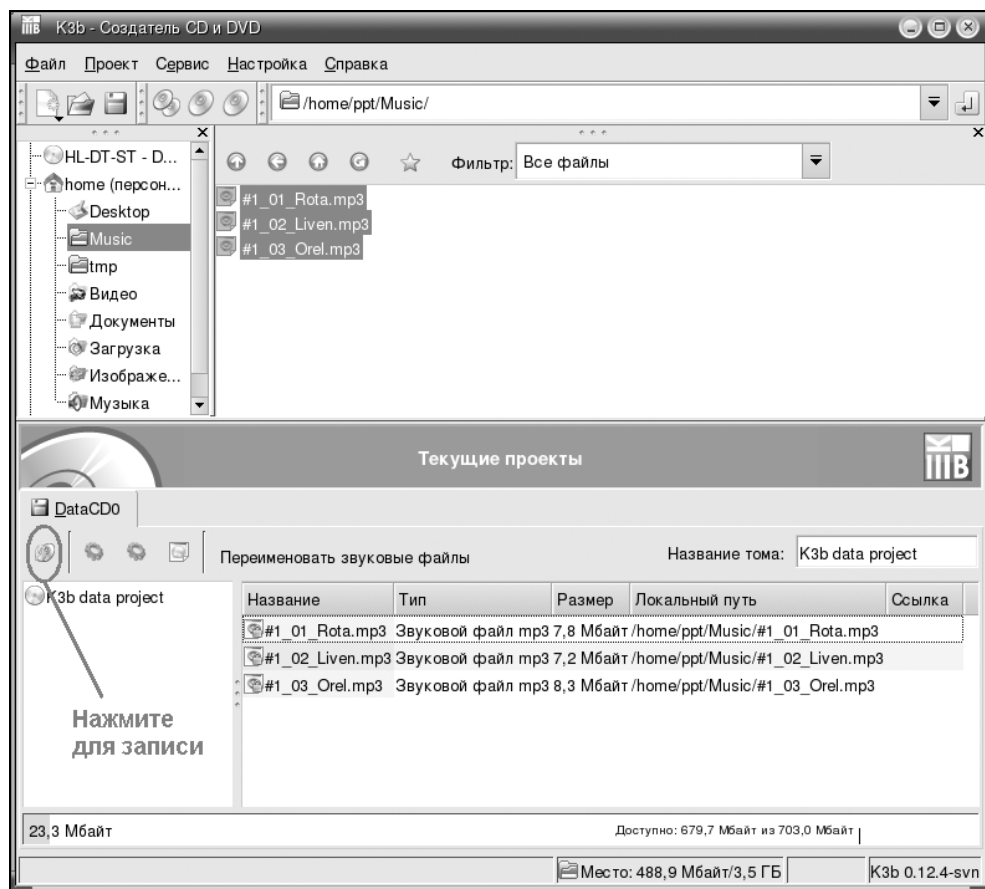


Рис. 10.3. Все готово для начала прожига диска

Если вы хотите, чтобы k3b не закрывал сессию, в диалоге записи перейдите на страницу **Настройки** и выберите один из режимов (рис. 10.9):

- ❖ **Начать многосессионную запись** — если у вас чистый CD-R/RW;
- ❖ **Продолжить многосессионную запись** — если у вас CD-R/RW с открытой сессией.

На вкладке **Файловая система** будут параметры **Создать расширение RockRidge** и **Создать расширение Joliet**. Не нужно выключать эти параметры, иначе русскоязычные имена файлов будут неправильно отображаться в MS Windows.

Действия по созданию DVD такие же, но в самом начале нужно выбрать действие **Новый проект DVD с данными**.

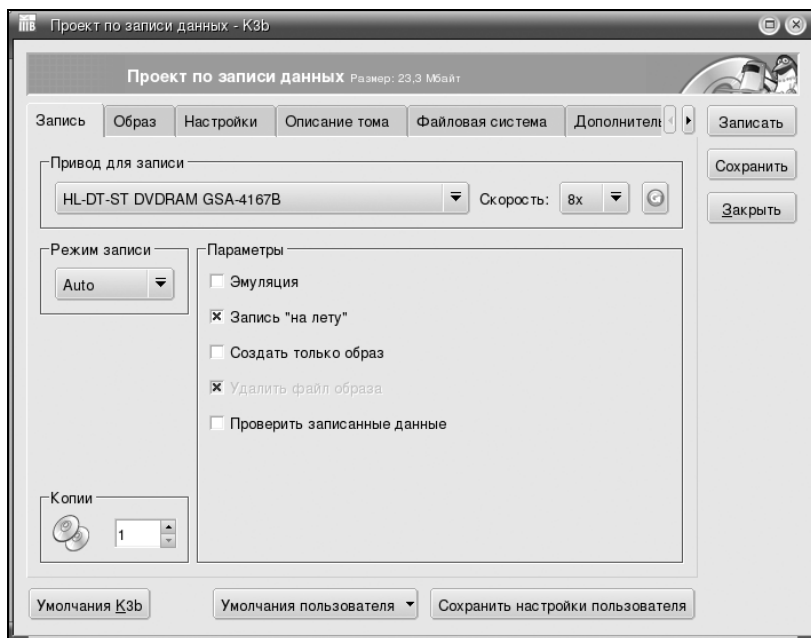
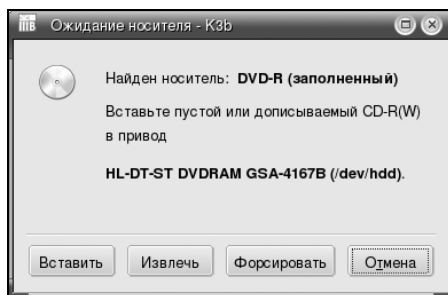
Рис. 10.4. Осталось нажать кнопку **Записать**

Рис. 10.5. Просьба вставить подходящий для записи проекта диск

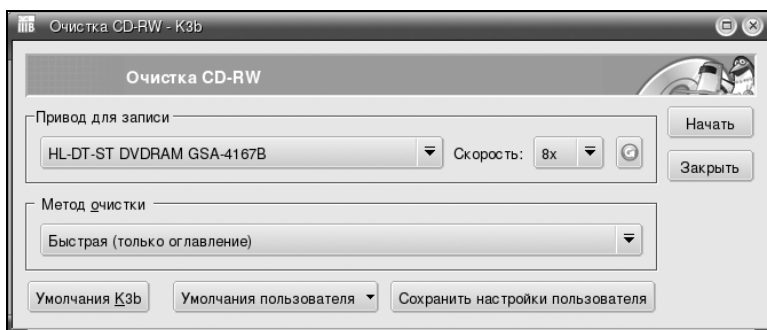


Рис. 10.6. Диалог очистки CD-RW

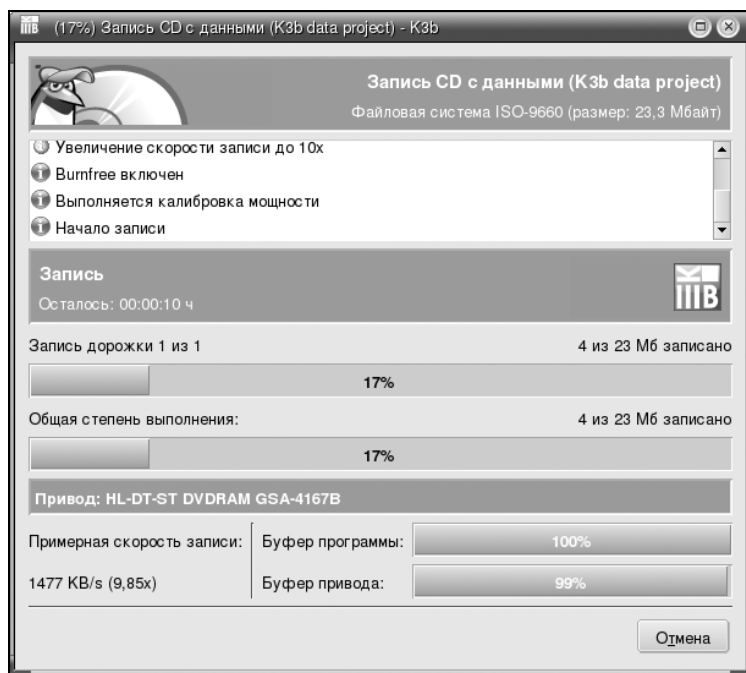


Рис. 10.7. Процесс записи

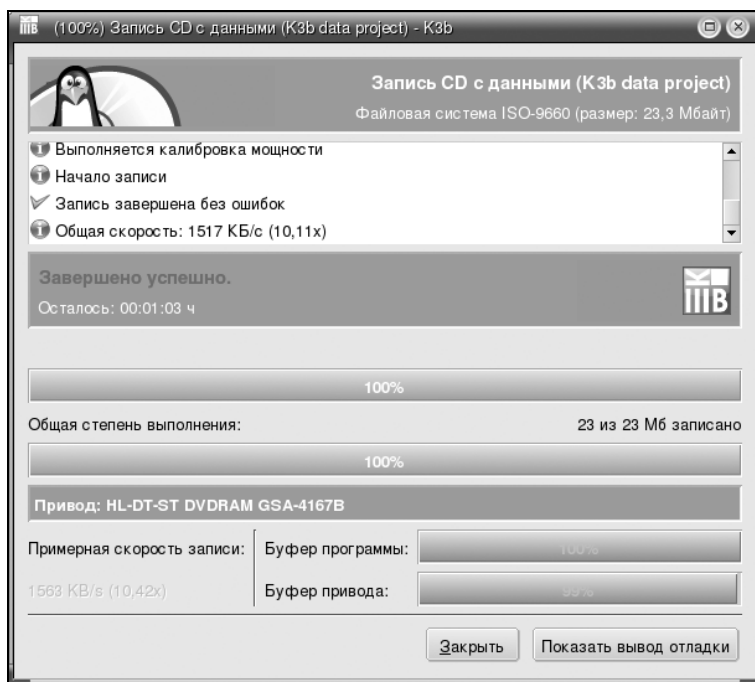


Рис. 10.8. Запись завершена

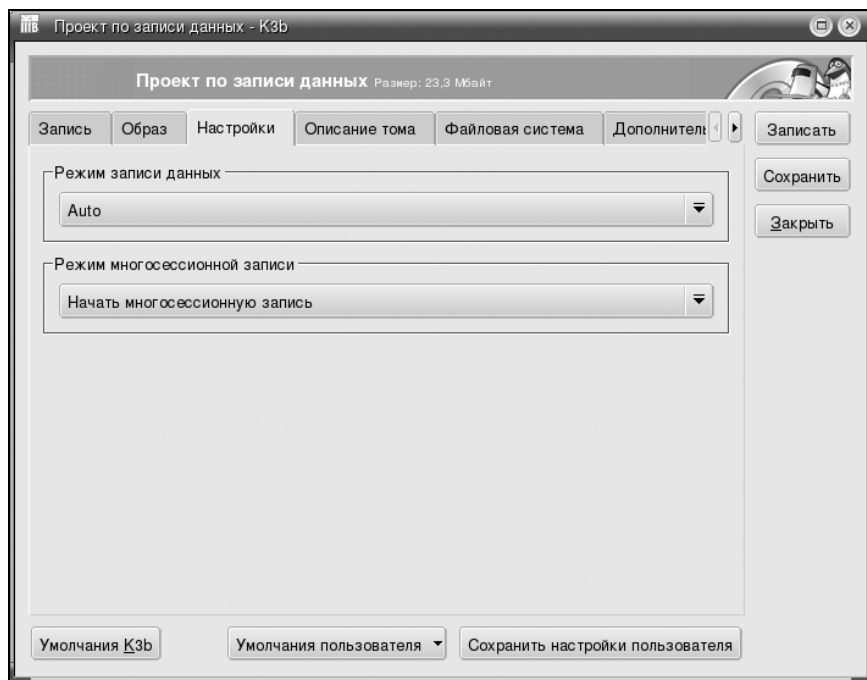


Рис. 10.9. Параметры мультисессии

10.4. Использование стандартных средств записи CD/DVD в Ubuntu

Записать CD/DVD-диск в Ubuntu очень просто. Даже проще, чем в других дистрибутивах. Для записи CD/DVD-диска (причем тип самого диска программа определит самостоятельно) вставьте чистый CD/DVD-диск в привод и выберите команду меню **Переход | Создать CD/DVD**. Появится окно, в которое нужно перетащить файлы, предполагаемые для записи на CD/DVD (рис. 10.10).

Для начала записи нажмите кнопку **Записать на диск**. В появившемся окне вам нужно выбрать привод для записи (если у вас их несколько) и максимальную скорость записи. Ради эксперимента я решил записать DVD-диск в виртуальной машине, т. е. Ubuntu запускался не на реальном компьютере, а в эмуляторе. Диск был создан без ошибок, что не всегда получается с другими дистрибутивами. Вот так вот (рис. 10.11).

Если вы вставили диск, на который невозможно записать данные (диск уже записан и сессия закрыта, т. е. уже нельзя дописать информацию на этот диск), то вы увидите соответствующее сообщение (рис. 10.12).

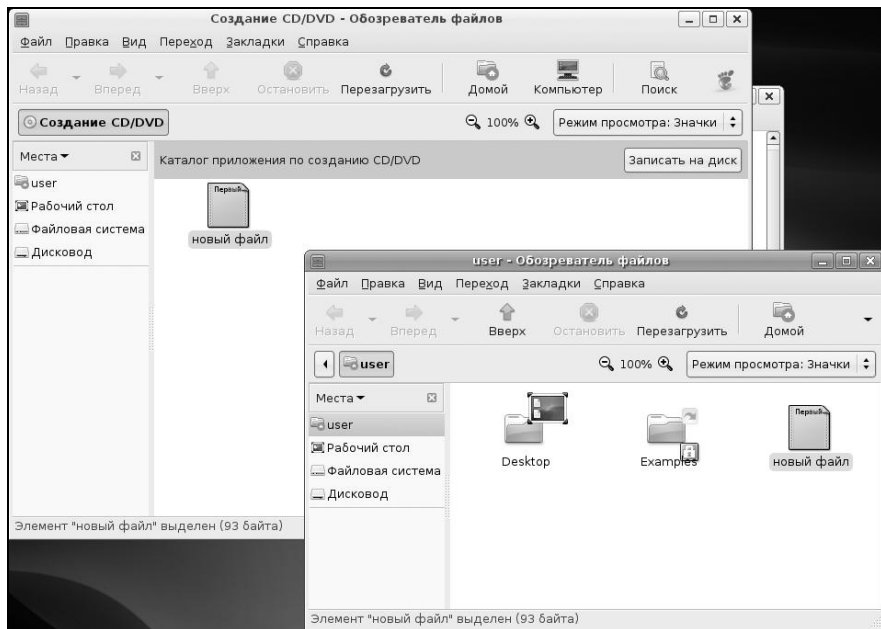


Рис. 10.10. Просто перетащите файлы, которые вы хотите записать

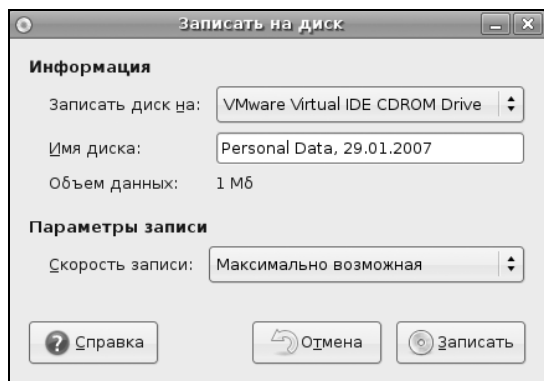


Рис. 10.11. Выберите привод и скорость записи

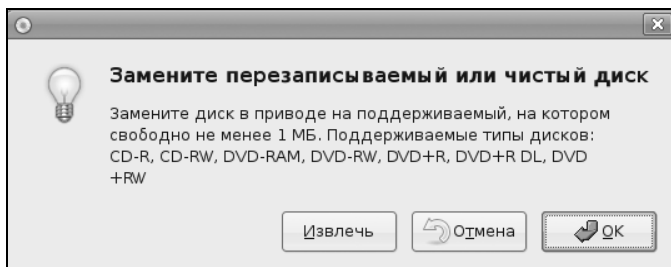


Рис. 10.12. Вставьте другой диск

ПРИМЕЧАНИЕ

Для очистки CD-RW/DVD-RW используются команды (предполагается, что ваш привод называется `/dev/cdrom`):

```
sudo umount /dev/cdrom
```

```
cdrecord dev=/dev/cdrom blank=fast
```

10.5. Программа Nero для Linux

В Windows, несмотря на наличие стандартных средств для записи CD/DVD, большинство пользователей используют программу Nero. Относительно недавно появилась ее Linux-версия, скачать которую можно на сайте <http://www.nero.com>.

Nero для Linux — *не бесплатная* программа. Вы можете скачать только trial-версию, которую можете некоторое время использовать бесплатно, после чего придется или удалить программу, или купить лицензию.

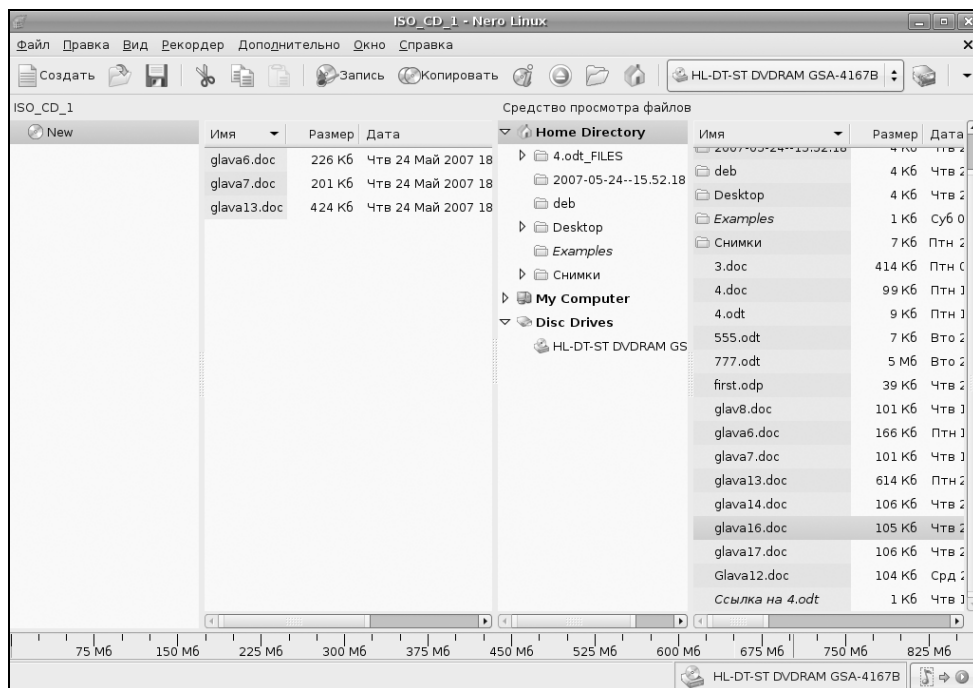


Рис. 10.13. Nero для Linux

При запуске от имени обычного пользователя (а не от имени суперпользователя root) Nero сообщит вам, что некоторые из устройств (а именно — ваш пишущий привод) недоступны. Все закономерно — просто у вашего пользователя не хватает прав для полного доступа к этим устройствам. Nero также сообщит имена этих устройств. У меня это были устройства `/dev/sg0`, `/dev/sg1` и `/dev/sg2`.

Если вы откроете список рекордеров (команда меню **Рекордер | Выбрать рекордер**), то обнаружите, что ваш рекордер не доступен, а вместо него используется виртуальный рекордер — Image Recorder, используемый для записи образов дисков.

Проблему решить достаточно просто (пусть это и не самое лучшее решение, но оно работает). Нужно от имени root ввести следующую команду:

```
# chmod 666 /dev/sg0 /dev/sg1 /dev/sg2
```

После этого нужно запустить Nero, выполнить команду **Рекордер | Выбрать рекордер** и выбрать ваш рекордер, который к тому времени уже появится в списке. Вот теперь можно приступить к полноценному использованию программы.

Программа Nero для Linux (рис. 10.13) полностью аналогична Windows-версии, поэтому подробно мы ее рассматривать не будем — все и так знают, как с ней работать.

10.6. Чтение "битых" компакт-дисков

Компакт-диски иногда портятся. Чаще всего причиной является сугубо механическое повреждение диска, например царапина. Считать данные с такого диска уже нельзя, но если потеря данных не критична (например, это диск с фильмом), можно попытаться считать те данные, которые остались.

Сначала нужно создать образ компакт-диска как есть (ясно, с пропуском ошибок):

```
# dd if=/dev/cdrom of=~cd.iso conv=noerror,sync
```

Затем нужно подмонтировать созданный образ к каталогу /mnt/iso (если данный каталог не существует, создайте его):

```
# mount -o loop ~/cd.iso /mnt/iso
```

Затем скопируем фильм из этого каталога в домашний каталог:

```
cp /mnt/iso/film.avi ~
```

После этого нужно размонтировать образ и удалить его:

```
# umount /mnt/iso
```

```
# del ~/cd.iso
```

ГЛАВА 11



Резервное копирование

11.1. Зачем нужно делать резервные копии

К сожалению, даже самые новые компьютеры не совершенны. Они иногда ломаются. Причиной сбоя может быть все, что угодно, например банальный перепад напряжения, из-за которого выходит из строя жесткий диск. Или же программный сбой — обычный вирус, который уничтожил таблицу разделов жесткого диска. Да, вирусов под Linux очень мало. Но в большинстве случаев на компьютере с Linux установлена еще и система Windows, а таблица разделов, как и винчестер, общая для обеих операционных систем. Поэтому нет никакой гарантии, что вирус, уничтожающий данные на винчестере, оставит данные с Linux-разделов в целости и сохранности.

Делать резервные копии полезно не только на сервере, но и на обычной рабочей станции (или домашнем компьютере). Представьте, что вы нечаянно удалили какой-то важный файл или изменили его не так, как нужно, а после этого выполнили команду **Файл | Сохранить**. В первом случае (удаление) файл еще можно восстановить, но только в том случае, если вы обнаружили пропажу файла сразу после восстановления. Если прошло некоторое время, скажем, неделя, то можно восстановить только часть файла или вообще ничего не восстановить, поскольку блоки, в которых размещался файл, были физически перезаписаны другими данными. Восстановление части файла имеет смысл только в случае с текстовыми файлами (не двоичными). Но все современные текстовые процессоры хранят данные не в текстовом формате, а в двоичном. Это связано, в первую очередь, с тем, что в документы часто внедряются двоичные данные — те же рисунки. Поэтому восстановление части файла ничего вам не даст — следовательно, можете считать, что файл потерян навсегда.

В случае, если у вас есть резервная копия, восстановить файл не составляет большого труда — вы просто скопируете его из копии. Правда, это хороший выход из положения?

Если же вы изменили файл и сохранили изменения, вам тоже поможет резервная копия. Ведь оригинальный файл уже не вернуть: как правило, после сохране-

ния изменений функция отмены последнего действия не работает. Конечно, можно заново все сделать (скажем, перепечатать несколько страниц), но намного быстрее и удобнее восстановить файл из резервной копии.

11.2. Выбор носителя для резервной копии

Раньше для создания резервных копий использовались стримеры — это устройства, записывающие данные на магнитную ленту. Конечно, не на обычную магнитофонную. В стример устанавливалась специальная кассета с магнитной лентой, на которую и записывалась информация. Преимущество такого решения заключалось в его дешевизне. На кассету можно было записать несколько гигабайт информации (это я рассказываю о том, как было раньше — сейчас на один DVD можно записать 18 Гбайт), а сами стримеры появились давно и были проверенным решением. Но недостатков тоже хватало: процесс создания резервной копии мог длиться часами — стримеры довольно медленные устройства. Кассеты с резервными копиями нужно было очень бережно хранить, поскольку они имели свойство, как и обычные магнитофонные кассеты, размагничиваться. Поэтому со временем актуальные резервные копии нужно было перезаписывать — обновлять. Это уже не говоря о "диверсии": испортить весь архив резервных копий мог один небольшой магнит.

С появлением CD все изменилось. Конечно, сначала приводы CD-RW стоили довольно дорого, поэтому для создания резервных копий использовались стримеры или обычные дискеты (в домашних условиях). Да, на дискету много не запишешь, но ведь и объемы данных были не такими, как сейчас. Скажем, в 1995 г. на несколько дискет в сжатом виде можно было записать практически все документы небольшой фирмы — текстовая информация очень хорошо сжимается.

Спустя несколько лет приводы CD-RW, как и сами диски ("болванки"), существенно подешевели и стали доступны обычным пользователям. На CD можно было записать до 700 Мбайт информации в несжатом виде. Если сжать информацию, то можно было записать до 1 Гбайт (все зависит от архиватора и от сжимаемой информации). Домашние пользователи, которые вообще раньше не делали резервных копий, начали активно использовать CD. А некоторые организации до сих пор использовали стримеры — им так было проще.

С появлением и удешевлением DVD стримеры вымерли как вид. Может, они где-то и используются, но намного проще сделать резервную копию на DVD. Преимущество такого решения заключается в следующем:

- ❖ на обычный DVD-диск можно записать до 4,5 Гбайт информации, на двухслойный и двусторонний можно записать до 17 Гбайт ваших данных;
- ❖ скорость записи и чтения DVD-диска не сравнится со скоростью чтения/записи стримера;
- ❖ DVD-диски намного надежнее кассет стримера.

Совершенно нет смысла делать резервную копию на другом жестком диске (или в другом разделе жесткого диска). В случае выхода жесткого диска из строя вы не сможете прочитать не только свои данные, но и резервную копию. Поэтому резервные копии нужно хранить на съемных носителях. Идеально подходят CD и DVD. Конечно, лучше записывать на DVD — на них больше помещается данных. Можно использовать и другие съемные носители, которые есть под рукой — Flash-диски, магнитооптические носители. Хотя по емкости пока не знаю ни один съемный носитель, доступный обычному пользователю, который смог бы превзойти DVD.

11.3. Правила хранения носителей с резервными копиями

Ваша резервная копия будет "жить долго", если вы будете придерживаться следующих простых правил.

1. На носитель с резервной копией не нужно записывать посторонние данные — предположим, что вы решили записать на DVD-диск свои документы общим объемом 1 Гбайт, но ведь 3,5 Гбайт вообще никак не используются! Существует соблазн использовать свободное место по прямому назначению и дописать диск до конца. Можно записать на диск, например, фильм или музыку. Но делать этого не стоит. Рано или поздно вы захотите послушать музыку или посмотреть фильм или, возможно, одолжите диск приятелю (чтобы он посмотрел фильм). В результате этого диск может быть утерян или поврежден. Помните, диском с резервной копией нужно пользоваться только тогда, когда эта копия вам необходима.
2. Не нужно дописывать на диск вторую резервную копию — опять-таки на диске осталось еще много свободного места, и вы хотите дописать на него следующую резервную копию (спустя некоторое время после записи первой). Не нужно этого делать: чем меньше мы используем диск, тем меньше он изнашивается, следовательно, тем лучше (правило 1). Хотя из этого правила есть исключения, диктуемые здравым смыслом и стратегией копирования. Об этом мы еще поговорим.
3. Никогда не доверяйте диски с важными данными посторонним людям — во-первых, это не желательно с точки зрения конфиденциальности данных, во-вторых, важные резервные копии могут быть просто утеряны.
4. Хранить диски нужно в темном сухом помещении и обязательно в отдельном боксе — на CD/DVD-диски, как и на магнитооптические носители (не говоря уже о лентах стримеров), негативно влияют солнечные лучи. Поэтому диски с резервными копиями нужно убрать подальше от прямого попадания солнечных лучей. Лучше положить их подальше от солнечных лучей и взглядов посторонних. В случае с магнитооптическими носителями их нужно держать подальше

от источников магнитного излучения, чтобы избежать размагничивания диска. Каждый диск нужно хранить в отдельном боксе — не храните диски в круглых коробках, в которых диски нанизываются на шкив, размещенный по центру коробки.

5. Подписывайте ваши носители — указывайте дату и время копирования, а также, по возможности, что находится на диске (это нужно писать на бумажной обложке бокса). Для надписи на поверхности диска можно использовать маркер. Все это облегчит поиск нужной резервной копии.

11.4. Стратегии создания резервной копии

Существует несколько стратегий создания резервных копий. Одна из них предполагает создание копии всего жесткого диска, а потом на отдельные носители записываются только изменившиеся данные. Спустя некоторое время (зависит от количества новых данных, время — от недели до месяца) опять делается копия всего жесткого диска. Такая стратегия идеально подходит для сервера предприятия, но не для обычного домашнего пользователя. К сожалению, хороших стратегий мало, поэтому пришлось разрабатывать собственную.

Начнем с первого положения — копирование всего винчестера. Делать этого не нужно — в этом просто нет смысла. Сейчас никого не удивит винчестером в 200 Гбайт. Даже при условии, что на DVD помещается до 17 Гбайт информации, для создания полной копии вам понадобится от 1 до 12 DVD-дисков. А теперь посчитаем, сколько времени понадобится на запись такой резервной копии? Учитывая даже восьмикратную скорость записи, на данное мероприятие уйдет целый день. Оно того не стоит. Ведь в случае с сервером предприятия все эти 200 Гбайт могут быть "забиты" важной информацией, например базой данных. А в случае с домашним компьютером большая часть дискового пространства занята фильмами и музыкой. Даже если произойдет потеря информации, то большую часть фильмов и музыки можно будет взять или во внутренней сети, или в ближайшем прокате. Поэтому данную информацию можно вообще исключить из резервной копии.

Делать резервную копию всех программных файлов тоже не вижу смысла — все это можно легко восстановить с дистрибутивных дисков. В резервную копию следует поместить только RPM-пакеты, загруженные из Интернета (т. е. пакеты, которых нет на других съемных носителях). Так, в случае потери данных вам не придется заново загружать нужные вам пакеты из Интернета.

На носитель резервной копии следует записывать:

- ❖ конфигурационные файлы системы — просто каждый раз при создании резервной копии записывайте на диск весь каталог `/etc`;
- ❖ измененные пользовательские данные — если не помните, что вы изменяли (с какими документами работали), можно записать весь каталог `/home` (не забудьте о каталоге `/root`). Учитывая объем DVD-диска, места вам хватит;

- ❖ RPM-пакеты, загруженные из Интернета — чтобы потом не пришлось заново их загружать;
- ❖ каталог `/var/www/html` — это корневой каталог Web-сервера, если, конечно, вы его используете. Как правило, Web-программисты тестируют на домашнем компьютере свои сценарии, поэтому копию данного каталога нужно сделать обязательно;
- ❖ каталог `/var/named` — в этом каталоге хранятся настройки кэширующего сервера DNS, если, конечно, вы его используете;
- ❖ каталог `/var/lib/mysql` — содержит базы данных сервера MySQL, если вы его используете;
- ❖ файл `/usr/src/linux-<версия_ядра>/config` — это конфигурационный файл вашего ядра. Его нужно записывать, если вы не используете стандартную версию ядра, а перекомпилировали ядро после установки системы.

Самое главное — создать первую резервную копию. Как правило, она будет самая большая. Потом нужно делать резервные копии, в среднем, раз в неделю. На диски нужно записывать только изменившиеся данные. Если вы знаете, что не изменяли конфигурацию системы, записывать каталог `/etc` уже не нужно. Если вы только работали с документами, запишите просто свой домашний каталог.

Для данной схемы вам понадобится два диска. Первый диск назовем диском месяца. Он будет содержать полную копию (по приведенным пунктам). На диск недели вы будете записывать каждую неделю (возможно, чаще — все зависит от важности изменившейся информации) изменившуюся информацию (в лучшем случае — просто каталог `/home`). На диск недели нужно информацию дописывать, т. е. в конце месяца у вас на этом диске будет минимум 4 каталога `/home` (названные по-разному, естественно, например, `home-1`, `home-2`, `home-3` и `home-4` — по номеру недели).

После этого в начале следующего месяца вы на новый диск записываете полную копию — создаете диск месяца. Ну, а потом схема повторяется. Здесь все просто, думаю, не запутаетесь.

11.5. Программа tar

Программу `tar` можно использовать для создания архива резервной копии: с одним архивом работать проще, чем тысячей файлов, которые нужно поместить в резервную копию, да и место на DVD сэкономим.

Мы не будем рассматривать все опции `tar` — их достаточно много (о них вы можете прочитать в руководстве по команде `man tar`), а рассмотрим только команду, позволяющую заархивировать нужный нам каталог:

```
tar -cvjf имя_архива.tar.bz2 каталог
```

Например,

```
tar -cvjf homes.tar.bz2 /home
```

На рис. 11.1 приведен процесс архивации.

```
[root@localhost etc]# tar -cvjf homes.tar.bz2 /home
tar: Удаляется начальный '/' из имен объектов
/home/
/home/den/
/home/den/.gnome/
/home/den/.gnome/gnome-vfs/
/home/den/.gnome/gnome-vfs/.trash_entry_cache
/home/den/.nautilus/
/home/den/.nautilus/metafiles/
/home/den/.nautilus/metafiles/file:%2F%2F%2Fmedia%2F2007.0-disc1.xml
/home/den/.nautilus/metafiles/x-nautilus-desktop:%2F%2F%2F.xml
/home/den/.nautilus/metafiles/file:%2F%2F%2Fmedia.xml
/home/den/.nautilus/metafiles/file:%2F%2F%2Fmedia%2F2007.0-disc1%2Fi586.xml
/home/den/.nautilus/metafiles/file:%2F%2F%2Fmedia%2F2007.0-disc1%2Fi586%2Fmedia.xml
/home/den/.nautilus/saved-session-S6PXXT
/home/den/.nautilus/saved-session-WBWRXT
/home/den/.nautilus/saved-session-LEG0XT
/home/den/Public/
/home/den/.gconfd/
/home/den/.gconfd/saved_state
/home/den/Music/
/home/den/.gnome2_private/
/home/den/.gstreamer-0.10/
/home/den/.gstreamer-0.10/registry.i686.xml
/home/den/.metacity/
/home/den/.metacity/sessions/
/home/den/.metacity/sessions/1188925986-2855-3760837421.ms
/home/den/.metacity/sessions/1181203896-2407-3925888756.ms
/home/den/.metacity/sessions/1188925847-2586-4103640012.ms
```

Рис. 11.1. Архивирование

Совет: M-t быстро изменяет формат списка панели.

```
[root@localhost etc]#
```

Помощь 2Меню 3Просмотр 4Правка 5Копия 6Переместить 7Настроить 8Удалить 9Меню 10Выход

Рис. 11.2. Работа с архивом в mc

Чтобы разархивировать архив, перейдите в каталог, в который вы хотите его распаковать, и введите команду:

```
tar -xvjf имя_архива.tar.bz2
```

Если вам нужно извлечь всего пару файлов, тогда проще использовать файловый менеджер `mc` (пакет тоже называется `mc`) (рис. 11.2).

11.6. Сетевое резервное копирование

Задача проста: у вас в сети есть компьютеры, например Web-сервер и почтовый сервер. Вам нужно сделать резервные копии данных, хранимых на этих компьютерах. Понятно, что отлучаться от своего рабочего места очень не хочется. Поэтому вы можете создать резервную копию по сети с помощью команды `scp`:

```
scp -r имя_каталога компьютер:каталог
```

Например:

```
scp -r web-cp web-server:/var/www
```

Команда `scp` (`secure copy`) используется для безопасного копирования файлов по сети. Для того чтобы она работала, нужно, чтобы на удаленном компьютере был установлен сервис `sshd`, о котором мы поговорим в *главе 41*.

Вернемся к нашей команде. Параметр `-r` означает, что нужно скопировать подкаталоги удаленного каталога, т. е. рекурсивное копирование. После него задается имя локального каталога, куда будут записаны скопированные файлы и каталоги. `web-sever` — это имя удаленного компьютера (можно задать IP-адрес), а через двоеточие указан удаленный каталог, который вы хотите скопировать.

Вам осталось лишь заархивировать каталог `web-cp`:

```
tar -cvjf web-cp.tar.bz2 web-cp
```

ГЛАВА 12



Редактирование таблицы разделов жесткого диска

12.1. Когда и зачем нужно редактировать таблицу разделов

При установке Linux вы в любом случае явно (когда задаете разметку диска вручную) или неявно (когда поручаете разметку диска программе установки Linux) изменяете таблицу разделов. Зачем же нужно изменять таблицу разделов после установки Linux?

Редактирование таблицы жесткого диска может понадобиться в следующих случаях:

- ♦ вы хотите изменить размер раздела — это нужно, например, если вы исчерпали размер Linux-раздела и хотите немного "подвинуть" Windows-раздел за счет освободившегося места;
- ♦ вы купили новый жесткий диск и хотите разметить его, т. е. создать разделы.

Для разметки диска в различных программах используются разные графические программы, например: `diskdrake` в Linux Mandriva, `YaST` в SUSE, `Disk Druid` в Fedora (Red Hat), `ASPDiskManager` в ASPLinux.

Использовать все эти программы очень просто — до такой степени просто, что даже новичок без особых проблем разметит диск. Собственно, для новичков эти программы и разрабатывались. Но, к сожалению, данные программы есть не во всех дистрибутивах и далеко не всегда под рукой. Например, программа `Disk Druid` входит в состав `Anaconda` — программы установки Fedora. Если дистрибутивных дисков нет под рукой, то и данная программа недоступна.

В любом дистрибутиве Linux есть стандартная программа `fdisk`, которая и будет рассмотрена в этой главе. Основной недостаток этой программы заключается в том, что она не умеет изменять размер раздела без потери данных. Это означает, что если вам нужно изменить размер раздела, то вам придется сначала удалить раздел, а потом на его месте создать новый — большего или меньшего размера (в зависимости от того, что вам нужно), а это невозможно выполнить без потери дан-

ных. Что делать, если вам нужно изменить размер раздела? Тогда лучшее решение — использовать одну из графических программ, о которых мы говорили ранее. Если же они не входят в состав дистрибутива, то у вас есть выбор:

- ♦ или достать дистрибутив Mandriva и загрузиться с загрузочного диска, после чего выбрать ручную разметку диска, изменить размер раздела, а после записи таблицы разделов нажать кнопку компьютера Reset, дабы отказаться от установки дистрибутива;
- ♦ или же использовать другую стандартную утилиту (правда, она не всегда устанавливается по умолчанию) — parted, но эта утилита пока умеет изменять разделы не всех типов.

В любом случае, если вы хотите стать квалифицированным администратором, вам нужно знать, как использовать стандартный fdisk.

12.2. Использование fdisk

Для разметки диска мы будем использовать стандартную программу fdisk, которая имеется во всех дистрибутивах Linux.

Введите команду (можно использовать короткие имена):

```
# fdisk <ИМЯ_устройства>
```

Например, если вы подключили винчестер как вторичный мастер, то команда будет следующей:

```
# fdisk /dev/sda
```

Чтобы убедиться, что диск не размечен, введите команду `p`. Программа выведет пустую таблицу разделов (рис. 12.1).

```
Command (m for help): p

Disk /dev/sda: 1825 MB, 1825360896 bytes
64 heads, 63 sectors/track, 884 cylinders
Units = cylinders of 4032 * 512 = 2064384 bytes

   Device Boot      Start         End      Blocks   Id  System
Command (m for help): _
```

Рис. 12.1. Таблица разделов пуста

Самое время создать раздел. Для этого используется команда `n` (рис. 12.2). Кстати, для справки можете ввести команду `m`, которая выведет список доступных команд fdisk (рис. 12.3).

После ввода команды `n` программа попросит вас уточнить, какого типа должен быть раздел. Можно выбрать первичный или расширенный раздел. В нашем случае больше подойдет первичный, поэтому вводим букву `p`. Затем нужно ввести номер раздела. Поскольку это первый раздел, то вводим 1. Потом fdisk попросит ввести

номер первого цилиндра. Это первый раздел, поэтому вводим номер 1. После ввода первого цилиндра нужно ввести номер последнего цилиндра. Чтобы не высчитывать на калькуляторе номер цилиндра, намного проще ввести размер раздела. Делается это так: +<размер>М. После числа должна идти именно буква м, иначе размер будет воспринят в байтах, а этого нам не нужно. Например, если вы хотите создать раздел размером 10 Гбайт, то введите +10240М.

```
Command (m for help): n
Command action
  e   extended
  p   primary partition (1-4)
p
Partition number (1-4): 1
First cylinder (1-884, default 1): 1
Last cylinder or +size or +sizeM or +sizeK (1-884, default 884): +700M
```

Рис. 12.2. Создание нового раздела

```
64 heads, 63 sectors/track, 884 cylinders
Units = cylinders of 4032 * 512 = 2064384 bytes

   Device Boot      Start         End      Blocks   Id  System
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)
Command (m for help): _
```

Рис. 12.3. Список команд программы fdisk

```
Command (m for help): p

Disk /dev/sda: 1825 MB, 1825360896 bytes
64 heads, 63 sectors/track, 884 cylinders
Units = cylinders of 4032 * 512 = 2064384 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1             1           340       685408+   83  Linux
/dev/sda2           341           884       1096704   83  Linux
Command (m for help): _
```

Рис. 12.4. Создание второго раздела, вывод таблицы разделов

Для создания второго раздела опять введите команду `n`. Программа вновь попросит тип раздела, номер первого цилиндра (это будет номер последнего цилиндра первого раздела плюс 1) и размер раздела. Если вы хотите создать раздел до "конца" диска, то просто введите номер последнего цилиндра.

Теперь посмотрим на таблицу разделов. Для этого опять введите команду `p` (рис. 12.4).

По умолчанию программа `fdisk` создает Linux-разделы. Если вы собираетесь работать только в Linux, можно оставить и так, но ведь не у всех есть Linux. Если вы снимете этот винчестер, чтобы, например, переписать у товарища большие файлы, то вряд ли сможете комфортно с ним работать. Прочитать данные (например, с помощью `Total Commander`) вам удастся, а что-либо записать — уже нет. Поэтому давайте изменим тип разделов. Для этого используется команда `t`. Введите эту команду. Программа запросит у вас номер раздела и тип файловой системы. С номером раздела все ясно, а вот с кодом файловой системы сложнее. Введите `L`, чтобы просмотреть доступные файловые системы (рис. 12.5).

Код `FAT32` — `b`. Введите его, и вы увидите сообщение программы, что тип файловой системы изменен (рис. 12.6).

0	Empty	1e	Hidden W95 FAT1	80	Old Minix	be	Solaris boot
1	FAT12	24	NEC DOS	81	Minix / old Lin	bf	Solaris
2	XENIX root	39	Plan 9	82	Linux swap / So	c1	DRDOS/sec (FAT-
3	XENIX usr	3c	PartitionMagic	83	Linux	c4	DRDOS/sec (FAT-
4	FAT16 <32M	40	Unix 80286	84	OS/2 hidden C:	c6	DRDOS/sec (FAT-
5	Extended	41	PPC PreP Boot	85	Linux extended	c7	Syrinx
6	FAT16	42	SFS	86	NTFS volume set	da	Non-FS data
7	HPFS/NTFS	4d	QNX4.x	87	NTFS volume set	db	CP/M / CTOS / .
8	AIX	4e	QNX4.x 2nd part	88	Linux plaintext	de	Dell Utility
9	AIX bootable	4f	QNX4.x 3rd part	8e	Linux LVM	df	BootIt
a	OS/2 Boot Manag	50	OnTrack DM	93	Amoeba	e1	DOS access
b	W95 FAT32	51	OnTrack DM6 Aux	94	Amoeba BBT	e3	DOS R/O
c	W95 FAT32 (LBA)	52	CP/M	9f	BSD/OS	e4	SpeedStor
e	W95 FAT16 (LBA)	53	OnTrack DM6 Aux	a0	IBM Thinkpad hi	eb	BeOS fs
f	W95 Ext'd (LBA)	54	OnTrackDM6	a5	FreeBSD	ee	EFI GPT
10	OPUS	55	EZ-Drive	a6	OpenBSD	ef	EFI (FAT-12/16/
11	Hidden FAT12	56	Golden Bow	a7	NeXTSTEP	f0	Linux/PA-RISC b
12	Compaq diagnost	5c	Priam Edisk	a8	Darwin UFS	f1	SpeedStor
14	Hidden FAT16 <3	61	SpeedStor	a9	NetBSD	f4	SpeedStor
16	Hidden FAT16	63	GNU HURD or Sys	ab	Darwin boot	f2	DOS secondary
17	Hidden HPFS/NTF	64	Novell Netware	b7	BSDI fs	fd	Linux raid auto
18	AST SmartSleep	65	Novell Netware	b8	BSDI swap	fe	LANstep
1b	Hidden W95 FAT3	70	DiskSecure Mult	bb	Boot Wizard hid	ff	BBT
1c	Hidden W95 FAT3	75	PC/IX				

Hex code (type L to list codes): _

Рис. 12.5. Коды файловых систем

```
Command (m for help): t
Partition number (1-4): 2
Hex code (type L to list codes): b
Changed system type of partition 2 to b (W95 FAT32)

Command (m for help): _
```

Рис. 12.6. Тип файловой системы изменен

Еще раз введите команду `p`, чтобы убедиться, что все нормально. Для сохранения таблицы разделов введите `w`, а для выхода без сохранения изменений — `q`.

12.3. Утилита `parted` — изменение размера разделов и восстановление таблицы разделов

Утилита `parted`, по сути, является аналогом `fdisk`, но умеет немного больше, чем `fdisk`: данная утилита умеет изменять размеры уже созданного раздела без потери данных, причем делает это, не требуя дефрагментации раздела, т. е. перед изменением размера раздела вам не нужно дефрагментировать его.

ПРИМЕЧАНИЕ

Утилита `parted` умеет изменять размеры разделов `ext2`, `ext3`, `fat16`, `fat32`, `linux-swaps` и `reiserfs`. Изменять размеры разделов других типов она пока не умеет.

Рассмотрим, как можно уменьшить размер определенного раздела. Запустим `parted`:

```
# parted имя_устройства
GNU Parted 1.6.4
Copyright (C) 1998, 1999, 2000, 2001, 2002 Free Software Foundation, Inc.
This program is free software, covered by the GNU General Public License.
This program is distributed in the hope that it will be useful, but WITHOUT
ANY
WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR
A PARTICULAR PURPOSE. See the GNU General Public License for more details.
Using /dev/ide/host0/bus0/target0/lun0/disc
...
```

После этого введем команду `print`, которая выведет таблицу разделов:

```
(parted) print
Disk geometry for /dev/ide/host0/bus0/target0/lun0/disc: 0.000-
38639,672 megabytes
Disk label type: msdos
```

Minor	Start	End	Type	Filesystem	Flags
2	0,031	1898,305	primary	fat32	boot
4	1898,306	2063,034	primary	linux-swaps	
3	2063,035	6063,596	primary	ext3	

```
...
```

Уменьшим третий раздел на 1 Гбайт:

```
resize 3 2063,035 5063,596
```

Команда `resize` используется для изменения размера раздела. Данной команде нужно передать три параметра: номер раздела, "начало" и "конец" раздела. В нашем случае мы изменяем размер третьего раздела: второй параметр соответствует значению `start` для выбранного раздела (как в выводе команды `print`) — его изменять не нужно, а вот в качестве третьего параметра следует установить нужное значение. Мы установили значение `5063,596`, что сократит размер нашего раздела на 1 Гбайт.

Но изменение разделов — это не единственная полезная функция `parted`. Кроме всего прочего, `parted` можно использовать еще и для восстановления таблицы разделов, если она по какой-то причине оказалась разрушенной. Для этого используется команда `rescue`. Ей нужно передать два параметра — начальное и конечное значения поиска разделов. Начальное значение — 0, конечное лучше указывать с запасом, например:

```
rescue 0 1000
```

Еще `parted` умеет перемещать и копировать разделы в пределах жесткого диска. Для этого используются команды `cp` и `move`, познакомиться с которыми вы можете, введя команду:

```
man parted
```

12.4. Графические редакторы таблицы разделов `diskdrake` и `gparted`

Если говорить о графических программах для разметки диска, то мне нравятся всего две программы — `diskdrake` для Linux Mandriva (рис. 12.7) и `gparted` для Debian/Ubuntu (рис. 12.8). Обе программы поддерживают все актуальные на сегодняшний день файловые системы и умеют изменять размер раздела.

Остальные программы не заслуживают внимания, уж лучше использовать текстовые программы `fdisk` или `parted`.

12.5. Программа `testdisk` — восстановление случайно удаленных разделов

Если вы нечаянно удалили раздел, то можете его восстановить с помощью программы `testdisk`. Данная программа сканирует дисковое пространство на предмет первого сектора раздела (первый сектор раздела в двух последних байтах содержит значения `0x55 0xAA`).

В результате сканирования `testdisk` выводит предполагаемые разделы (которые вы удалили), и вы можете выбрать раздел для восстановления.

После восстановления рекомендуется проверить восстановленный раздел программой `fsck`.

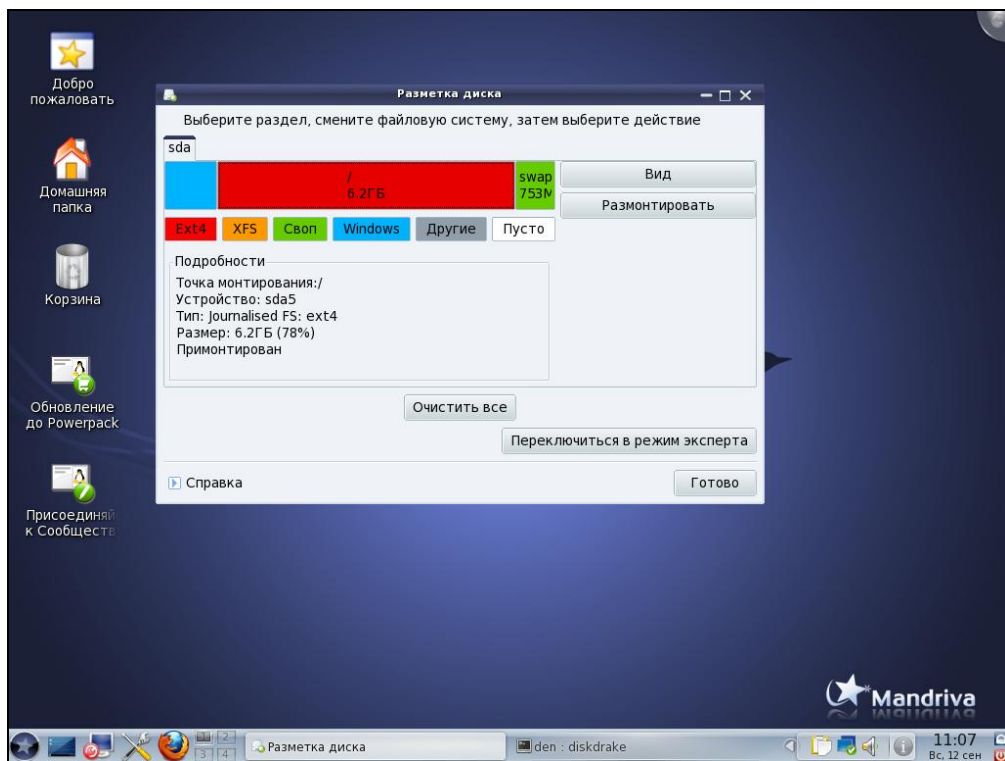


Рис. 12.7. Программа diskdrake (Linux Mandriva)

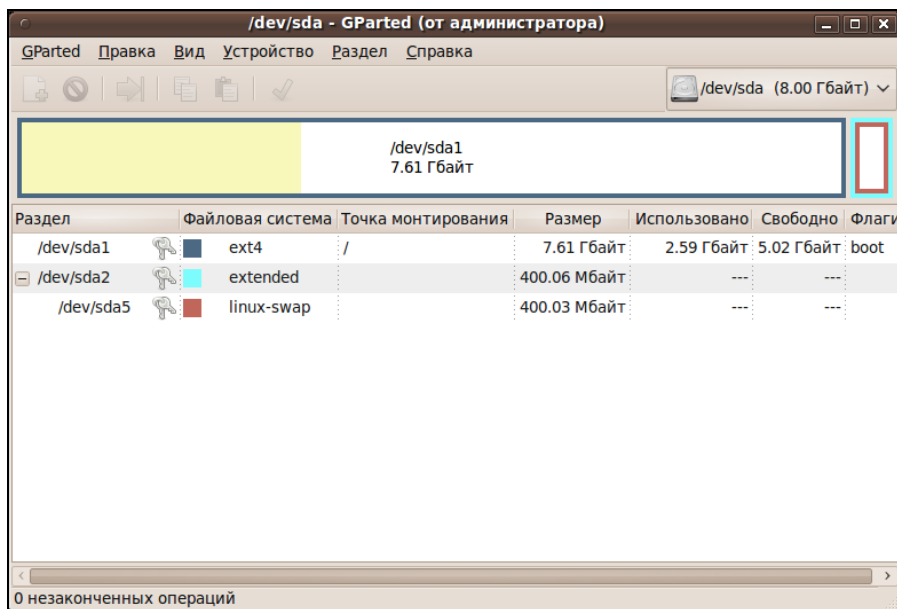


Рис. 12.8. Программа gparted



ЧАСТЬ III

ПОЛЬЗОВАТЕЛИ И ГРУППЫ

ГЛАВА 13



Пользователи и группы

13.1. Многопользовательская система

Linux, как и UNIX, является многозадачной многопользовательской операционной системой. Это означает, что в один момент с системой могут работать несколько пользователей, и каждый пользователь может запустить несколько приложений. При этом вы можете зайти в систему локально, а кто-то — удаленно, используя один из протоколов удаленного доступа (telnet, ssh) или по FTP. Согласитесь, очень удобно. Предположим, что вы забыли распечатать очень важный документ, а возвращаться домой уже нет времени. Если ваш компьютер должным образом настроен и подключен к Интернету, вы можете получить к нему доступ (даже если компьютер выключен, достаточно позвонить домой и попросить кого-то включить его, а к Интернету компьютер подключится автоматически). После чего зайдете в систему по ssh (или подключитесь к графическому интерфейсу, если вы предпочитаете работать в графическом режиме) и скопируете нужный вам файл. Даже если кто-то в момент вашего подключения уже работает с системой, вы не будете мешать друг другу.

Вы можете обвинить меня в рекламе Linux: мол, эта возможность была и в Windows 98, если установить соответствующее программное обеспечение вроде Remote Administrator. Должен отметить, что в Windows все иначе. Да, Remote Administrator предоставляет удаленный доступ к рабочему столу, но если за компьютером уже работает пользователь, то вы вместе работать не сможете — вы будете мешать ему, а он вам. Ведь все, что будете делать вы, будет видеть он, а все, что будет делать он, вы увидите у себя на экране, т. е. рабочий стол получится как бы общий. Если вы предварительно не предупредите пользователя о своем удаленном входе, он даже может подумать, что с системой что-то не то. Помню, со мной так и было — пользователь, работавший за компьютером, закрывал окна, которые я открывал, работая в удаленном режиме. Пришлось мне самому пойти к компьютеру того пользователя и попросить его не мешать.

В Linux же все так, как и должно быть. Несколько пользователей могут работать с системой и даже не подозревать о существовании друг друга, пока не введут соответствующую команду (`who`).

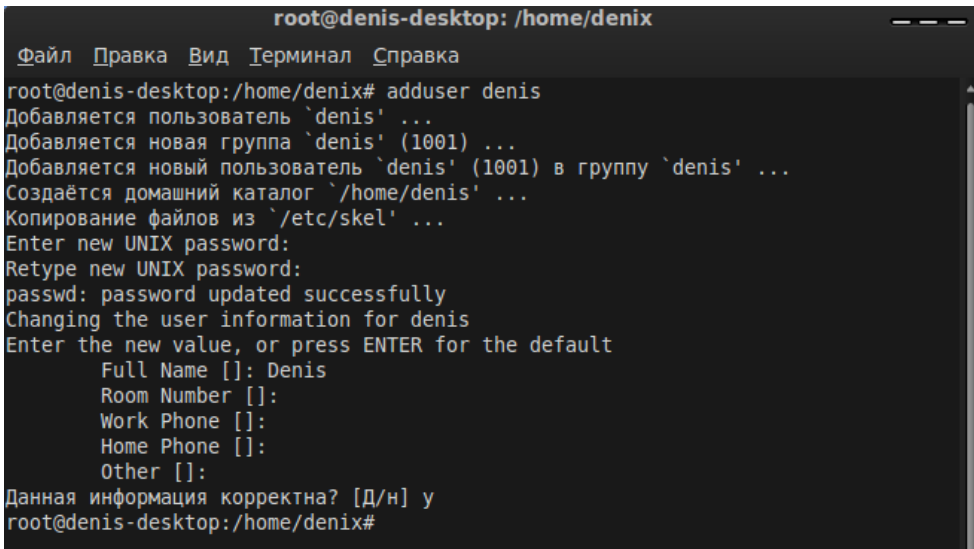
13.2. Создание, удаление и модификация пользователей стандартными средствами

Для добавления нового пользователя выполните следующие команды (от имени root):

```
# adduser <ИМЯ ПОЛЬЗОВАТЕЛЯ>
# passwd <ИМЯ ПОЛЬЗОВАТЕЛЯ>
```

Первая команда (`adduser`) добавляет пользователя, а вторая (`passwd`) изменяет его пароль. Ясно, что и в первом, и во втором случае вы должны указать одно и то же имя пользователя.

В некоторых дистрибутивах, например в Ubuntu и Debian, сценарий `adduser` не только добавляет пользователя, но позволяет указать дополнительную информацию о пользователе и сразу же задать пароль пользователя (рис. 13.1).



```
root@denis-desktop: /home/denix
Файл Правка Вид Терминал Справка
root@denis-desktop:/home/denix# adduser denis
Добавляется пользователь `denis' ...
Добавляется новая группа `denis' (1001) ...
Добавляется новый пользователь `denis' (1001) в группу `denis' ...
Создаётся домашний каталог `/home/denis' ...
Копирование файлов из `/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for denis
Enter the new value, or press ENTER for the default
  Full Name []: Denis
  Room Number []:
  Work Phone []:
  Home Phone []:
  Other []:
Данная информация корректна? [Д/н] у
root@denis-desktop:/home/denix#
```

Рис. 13.1. Добавление нового пользователя в Ubuntu

ПРИМЕЧАНИЕ

В некоторых дистрибутивах (например, в openSUSE) вместо команды `adduser` используется команда `useradd`. Программы `adduser` и `useradd` обычно находятся в каталоге `/usr/sbin`.

Обратите внимание — если пароль слишком прост для подбора, программа `passwd` выдаст соответствующее предупреждение — **BAD PASSWORD** и сообщит, чем же наш пароль плох (например, в основе пароля лежит словарное слово, что делает пароль легким для подбора).

Для модифицирования учетной записи пользователя можно использовать команду `usermod`. О ней вы прочитаете в руководстве `man`, вызвав его командой:

```
man usermod
```

Особого смысла рассматривать эту команду я не вижу, ведь обычно нужно менять только пароль пользователя, а это можно сделать с помощью команды `passwd`. А если вам требуется изменить саму учетную запись (например, указать другой домашний каталог), то это гораздо удобнее сделать с помощью графического конфигуратора (об этом позже) или обычного текстового редактора.

ПРИМЕЧАНИЕ

Команду `passwd` может использовать не только администратор, но и сам пользователь для изменения собственного пароля.

Для удаления пользователя служит команда `userdel`:

```
# userdel <ИМЯ_ПОЛЬЗОВАТЕЛЯ>
```

Давайте разберемся, что же происходит при создании новой учетной записи пользователя.

Во-первых, создается запись в файле `/etc/passwd`. Формат записи следующий:

```
ИМЯ_ПОЛЬЗОВАТЕЛЯ:пароль:UID:GID:полное_имя:домашний_каталог:оболочка
```

Рассмотрим фрагмент этого файла (две строки):

```
root:x:0:0:root:/root:/bin/bash
den:x:500:500:Denis:/home/den:/bin/bash
```

- ❖ Первое поле — это логин пользователя, который он вводит для регистрации в системе. Пароль в современных системах в этом файле не указывается, а второе поле осталось просто для совместимости со старыми системами. Пароли хранятся в файле `/etc/shadow`, о котором мы поговорим чуть позже.
- ❖ Третье и четвертое поля — это UID (User ID) и GID (Group ID), идентификаторы пользователя и группы соответственно. Идентификатор пользователя `root` всегда равен 0, как и идентификатор группы `root`. Список групп вы найдете в файле `/etc/groups`.
- ❖ Пятое поле — это настоящее имя пользователя. Может быть не заполнено, а может содержать фамилию, имя и отчество пользователя — все зависит от педантичности администратора системы, т. е. от вас. Если вы работаете за компьютером в гордом одиночестве, то, думаю, свою фамилию вы не забудете. А вот если ваш компьютер — сервер сети, тогда просто необходимо указать Ф.И.О. каждого пользователя, а то, когда придет время обратиться к пользователю по имени, вы его знать не будете. (Попробуйте запомнить 500 фамилий и имен!)
- ❖ Шестое поле содержит имя домашнего каталога. Обычно это каталог `/home/<ИМЯ_ПОЛЬЗОВАТЕЛЯ>`.
- ❖ Последнее поле — это имя командного интерпретатора, который будет обрабатывать введенные вами команды, когда вы зарегистрируетесь в консоли.

В целях безопасности пароли были перенесены в файл `/etc/shadow` (доступен для чтения/записи только пользователю `root`), где они и хранятся в закодированном виде (используется алгоритм MD5 или Blowfish в некоторых системах). Узнать, с помощью какого алгоритма зашифрован пароль, очень просто: посмотрите на шифр — если он достаточно короткий и не начинается с символа `$`, то применен алгоритм DES (самый слабый и ненадежный — как правило, используется в старых дистрибутивах). Если же шифр начинается с символов `1`, то это MD5, а если в начале шифра имеются символы `$2a$`, то это Blowfish.

Во-вторых, при создании пользователя создается каталог `/home/<имя пользователя>`, в который копируется содержимое каталога `/etc/skel`. Каталог `/etc/skel` содержит "джентльменский набор" — файлы конфигурации по умолчанию, которые должны быть в любом пользовательском каталоге. Название каталога `skel` (от англ. *skeleton*) полностью оправдывает себя — он действительно содержит "скелет" домашнего каталога пользователя.

ПРИМЕЧАНИЕ

Файл `/etc/passwd` можно редактировать с помощью обычного текстового редактора. То есть вы можете очень легко, не прибегая к помощи ни графического конфигуратора, ни команды `usermod`, изменить параметры учетной записи любого пользователя, например, задать для него другую оболочку или прописать его настоящую фамилию. Однако нужно быть осторожным при изменении домашнего каталога пользователя! Если вы это сделали, то, чтобы у пользователя не возникло проблем с правами доступа для нового каталога, нужно выполнить команду:

```
chown -R <пользователь> <каталог>
```

13.3. Группы пользователей

Иногда пользователей объединяют в *группы*. Группы позволяют более эффективно управлять правами пользователей. Например, у нас есть три пользователя: `igor`, `ravel`, `alex`, которые должны совместно работать над проектом. Их достаточно объединить в одну группу — тогда пользователи будут иметь доступ к домашним каталогам друг друга (по умолчанию один пользователь не имеет доступ к домашнему каталогу другого пользователя, поскольку пользователи находятся в разных группах).

Создать группу, а также поместить пользователя в группу позволяют графические конфигураторы. Вы можете использовать их — они очень удобные, но если вы хотите стать настоящим линуксоидом, то должны знать, что доступные в системе группы указываются в файле `/etc/group`. Добавить новую группу в систему можно с помощью команды `groupadd`, но, как правило, проще добавить в текстовом редакторе еще одну запись в файл `/etc/group`, а изменить группу пользователя еще проще — для этого достаточно отредактировать файл `/etc/passwd`.

13.3.1. Управление пользователями и группами с помощью графических конфигураторов

Обычно добавлять/изменять учетные записи пользователей принято в командной строке. Но сейчас мы поговорим о *графических конфигураторах* — они пригодятся любителям графического интерфейса, а также начинающим пользователям, которые еще не уверены в своих силах. Понятно, что в каждом дистрибутиве будут свои конфигураторы, поэтому мы остановимся лишь на четырех наиболее популярных дистрибутивах: Fedora, Mandriva, openSUSE и Ubuntu.

13.3.2. Конфигуратор system-config-users в Fedora

В Fedora (ASPLinux, CentOS) для редактирования учетных записей пользователей и групп пользователей служит конфигуратор system-config-users (рис. 13.2).

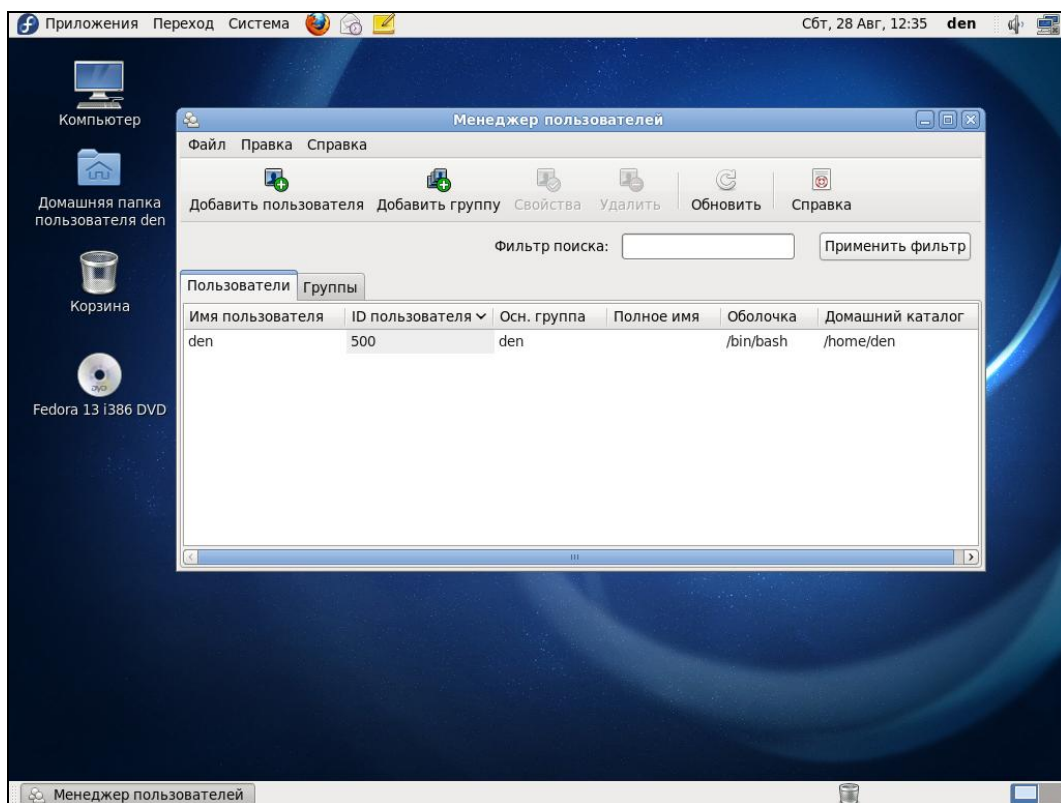


Рис. 13.2. Конфигуратор system-config-users (Менеджер пользователей)

Для добавления пользователя используется кнопка **Добавить пользователя**, для удаления — **Удалить**, для добавления группы — **Добавить группу**, а для ре-

дактирования пользователя или группы — кнопка **Свойства**. Разобраться с конфигуратором очень просто, поэтому вы справитесь и без моих комментариев (рис. 13.3).

Кроме этой программы, встречается программа `userpasswd`, позволяющая изменить пароль текущего пользователя. Опять-таки, это графическая программа, представляющая собой окно с полем ввода, двумя кнопками (**Ок** и **Отмена**) и приглашением изменить свой пароль.

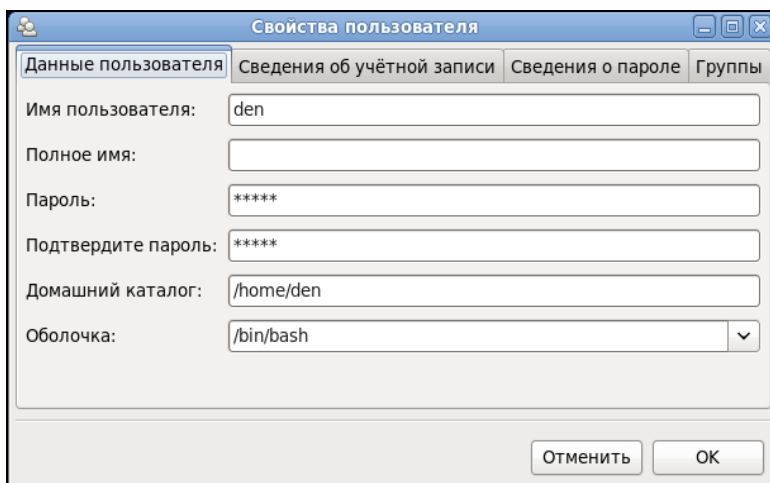


Рис. 13.3. Конфигуратор `system-config-users`: редактирование учетной записи пользователя

13.3.3. Конфигуратор `drakuser` в Linux Mandriva

В Mandriva для редактирования пользователей и групп используется конфигуратор `drakuser` (рис. 13.4).

На панели инструментов `drakuser` всего пять кнопок:

- ◆ **Добавить пользователя в систему** — добавляет пользователя;
- ◆ **Добавить группу** — добавляет группу;
- ◆ **Редактировать** — редактирует учетную запись пользователя или группы в зависимости от того, какая запись выделена;
- ◆ **Удалить** — удаляет выделенную учетную запись пользователя или группы;
- ◆ **Обновить** — обновляет список (список создается при запуске программы) на тот случай, если вы добавили пользователя с помощью `adduser` уже после запуска конфигуратора.

13.3.4. Пользователи и группы в Ubuntu

Для создания учетной записи пользователя в Ubuntu выполните команду меню **Система | Администрирование | Пользователи и группы**.

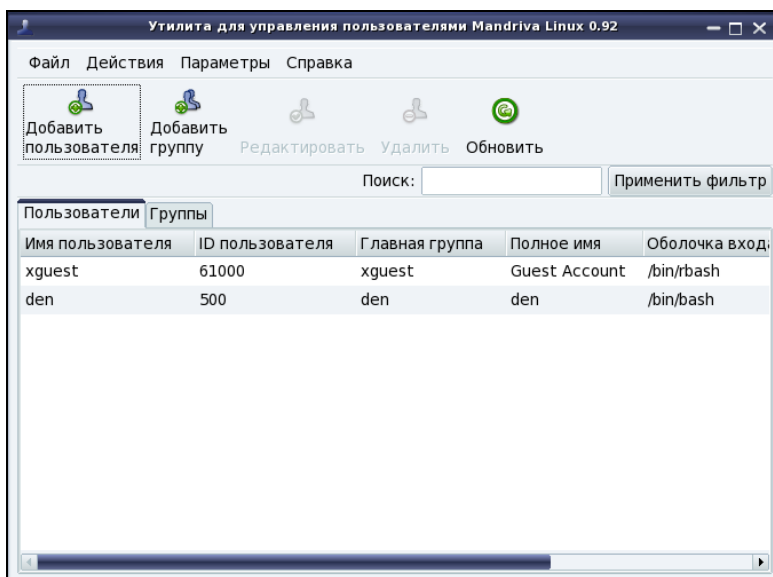


Рис. 13.4. Конфигуратор drakuser

СОВЕТ

В Ubuntu 9.10 заметил небольшой "глюк" — при запуске конфигуратора **Пользователи и группы** кнопка **Добавить пользователя** может быть недоступна, ничего не происходит и при нажатии кнопки разблокирования...

Для выхода из ситуации запустите конфигуратор users-admin напрямую из терминала:
`sudo users-admin.`

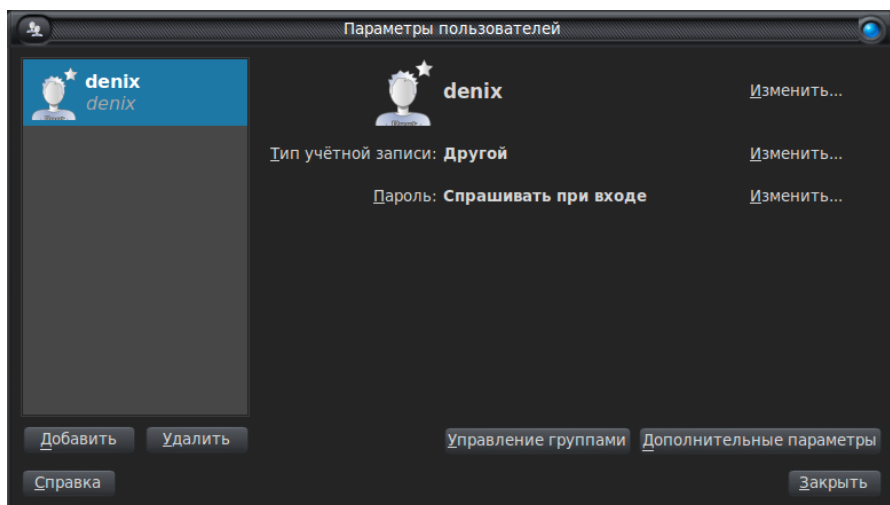


Рис. 13.5. Окно Параметры пользователей

В открывшемся окне (рис. 13.5) нажмите кнопку **Добавить** (в предыдущих версиях — **Добавить пользователя**). Откроется следующее окно (рис. 13.6), в котором нужно ввести имя пользователя и логин (**Короткое имя**) — имя, которое будет использоваться для входа в систему.

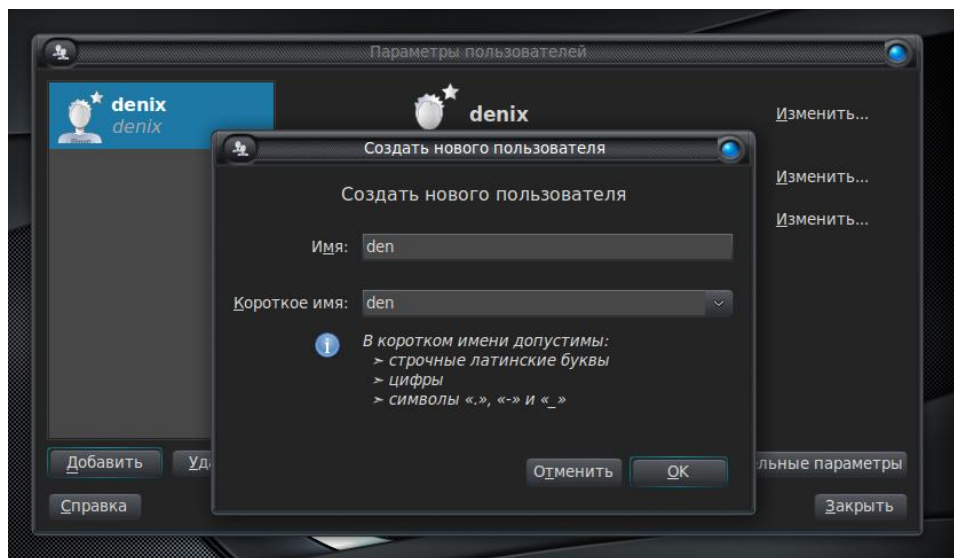


Рис. 13.6. Ввод имени и логина пользователя

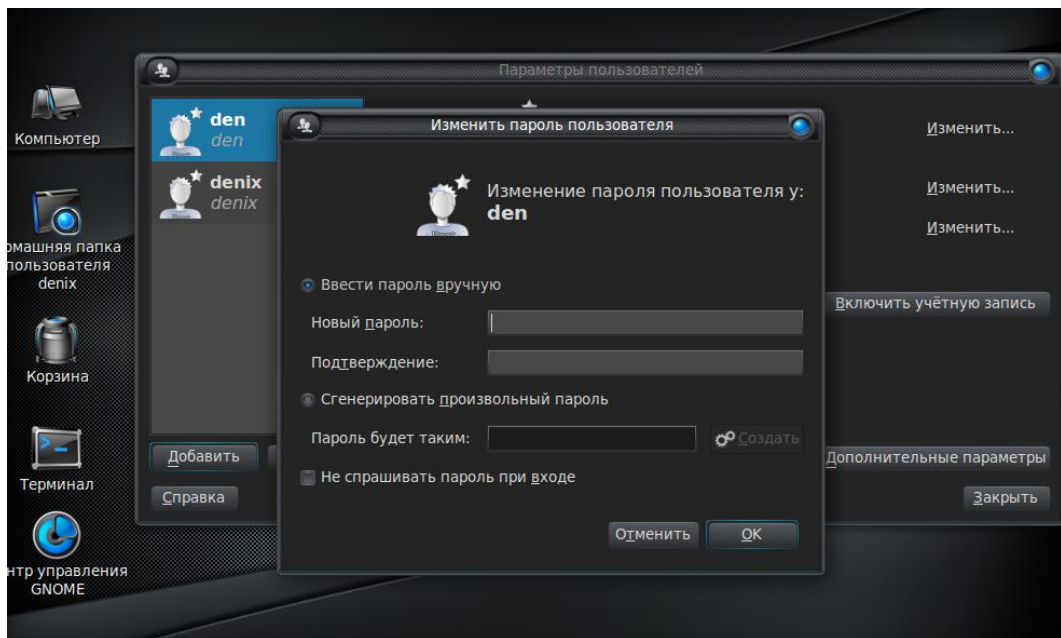


Рис. 13.7. Ввод пароля пользователя

После этого вы увидите окно изменения пароля пользователя (рис. 13.7). Вы можете или ввести пароль вручную, или сгенерировать его автоматически. Произвольно сгенерированный пароль будет сложнее для подбора, но и сложнее для запоминания. Поэтому решайте сами, что для вас важнее — безопасность или комфорт.

Для гостевых учетных записей можно включить параметр **Не спрашивать пароль при входе**.

Если щелкнуть на кнопке **Изменить** напротив поля **Тип учетной записи**, можно выбрать тип учетной записи (рис. 13.8):

- ❖ **Администратор** — пользователь может администрировать систему;
- ❖ **Пользователь** — пользователь может работать в системе, но не может администрировать ее (использовать команду `sudo`, устанавливать программы, управлять пользователями и т. д.);
- ❖ **Другое** — учетная запись с особыми параметрами доступа, которые устанавливаются вручную. Для установки особых параметров доступа нужно щелкнуть на кнопке **Дополнительные параметры**. На вкладке **Права пользователя** (рис. 13.9) можно определить, какие операции может выполнять пользователь, а какие — нет. На вкладке **Дополнительно** можно выключить учетную запись, изменить группу, ID пользователя, командный интерпретатор и даже домашний каталог (рис. 13.10).

В Ubuntu вход в систему под именем `root` запрещен, поэтому команда `sudo`, позволяющая запускать программы с привилегиями `root`, очень важна. Пользователи-администраторы имеют право отдавать эту команду, а обычные пользователи — нет.

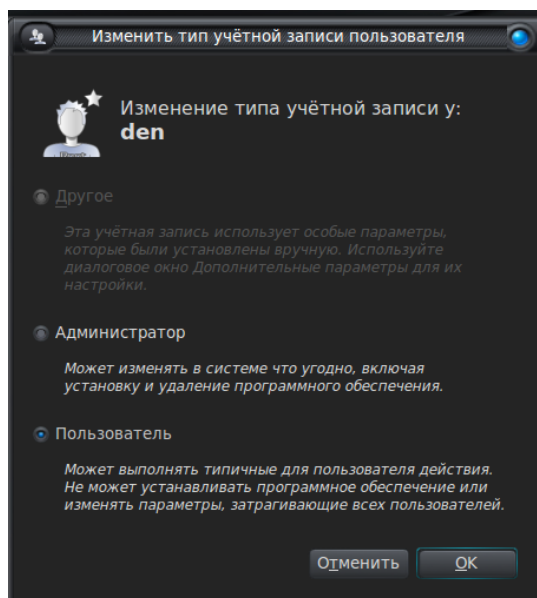


Рис. 13.8. Тип учетной записи

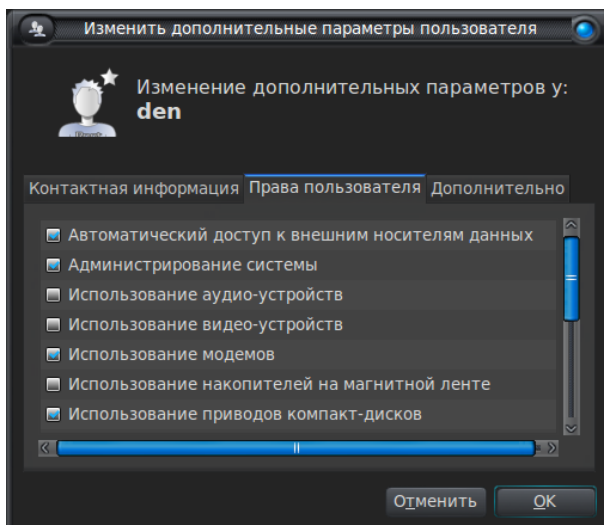


Рис. 13.9. Дополнительные параметры учетной записи

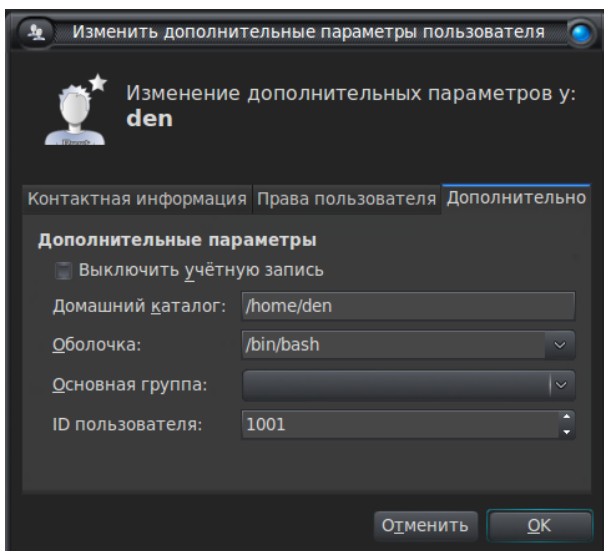


Рис. 13.10. Вкладка Дополнительно

Если пользователь уже создан, но вам понадобилось разрешить ему использовать `sudo`, в окне **Параметры пользователей** (ранее **Пользователи и группы**) выделите учетную запись пользователя, нажмите кнопку **Дополнительные параметры** и на вкладке **Права пользователя** (см. рис. 13.9) разрешите **Администрирование системы**.

Кнопка **Управление группами** открывает окно управления группами. Вы увидите список групп и кнопки **Добавить** (добавляет новую группу), **Удалить** (ис-

пользуется для удаления группы) и **Свойства** (позволяет легко добавить/исключить членов группы). Конфигуратор групп очень прост, поэтому вы разберетесь с ним без моих комментариев.

13.3.5. Графический конфигуратор в openSUSE

Для запуска конфигуратора **Управление пользователями и группами** (рис. 13.11) выполните команду меню **Компьютер | YaST | Управление пользователями и группами**.

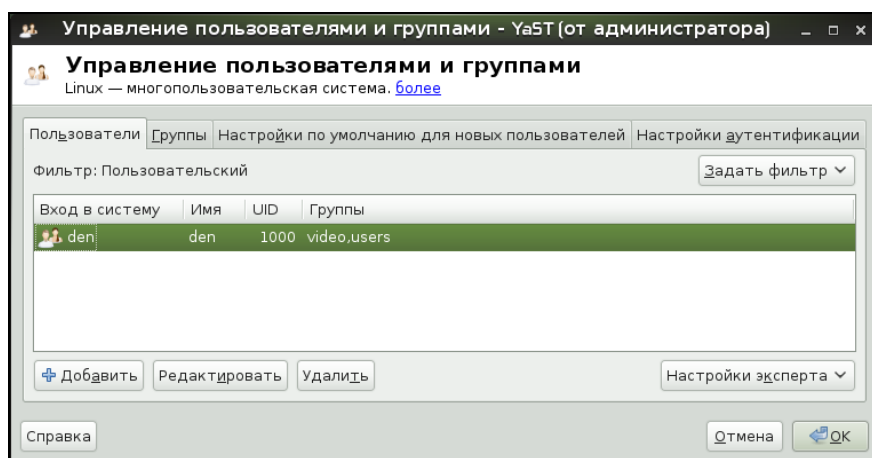


Рис. 13.11. Окно **Управление пользователями и группами**, вкладка **Пользователи**

Использовать конфигуратор очень просто: кнопка **Добавить** служит для создания нового пользователя, а кнопки **Редактировать** и **Удалить** — для изменения и удаления, соответственно, уже созданного.

При создании пользователя (рис. 13.12) у вас есть возможность (на вкладке **Подробности**) выбрать, к каким группам должен принадлежать данный пользователь. Если пользователю не нужен доступ к Интернету, не следует пометить его принадлежность к группе **dialout**.

Даже если при создании пользователя вы забыли определить группы, к которым должен принадлежать пользователь, то всегда сможете сделать это позже — при изменении его учетной записи (кнопка **Редактировать**).

СОВЕТ

Если вы не хотите удалять пользователя, а нужно временно запретить ему вход в систему, выделите его, нажмите кнопку **Редактировать** и установите флажок **Отключить вход пользователя в систему**.

Для редактирования групп (создания, удаления, изменения списка членов группы) следует перейти на вкладку **Группы** (рис. 13.13).

Новый локальный пользователь - YaST (от администратора)

Новый локальный пользователь

Введите Полное имя пользователя, Имя пользователя и Пароль, присваиваемые этой учётно... [более](#)

Информация о пользователе | Подробности | Настройки пароля | Дополнения

Полное имя пользователя:
suse

Имя пользователя:
suse

Пароль:
●●●●

Подтвердить пароль:
●●●●

☐ Получать системные сообщения

☐ Отключить вход пользователя в систему

Справка Отмена OK

Рис. 13.12. Создание нового пользователя

Управление пользователями и группами - YaST (от администратора)

Управление пользователями и группами

Linux — многопользовательская система. [более](#)

Пользователи | Группы | Настройки по умолчанию для новых пользователей | Настройки аутентификации

Фильтр: Пользовательский [Задать фильтр](#)

Имя группы	ID группы	Члены группы
users	100	den, games, suse

[+](#) Добавить Редактировать Удалить

Настройки эксперта

Справка Отмена OK

Рис. 13.13. Окно Управление пользователями и группами, вкладка Группы

Нажав кнопку **Редактировать**, вы можете изменить параметры группы (рис. 13.14), например добавить в ее состав новых пользователей. А вот чтобы удалить пользователя из группы, вам придется перейти на вкладку **Пользователи**, выбрать нужного пользователя, нажать кнопку **Редактировать**, затем перейти на вкладку **Подробности** и уже там отключить группы, членом которых не должен быть пользователь. Да, неудобно, но другого способа нет.

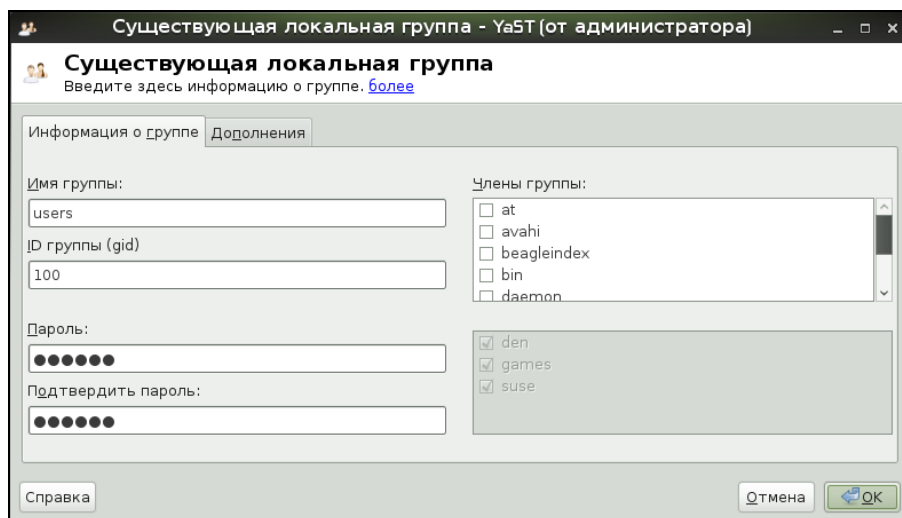


Рис. 13.14. Изменение группы

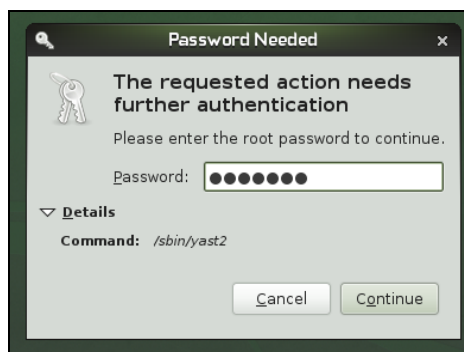


Рис. 13.15. Введите пароль root

Когда вы запускаете какой-нибудь конфигуратор, система просит вас ввести пароль root (рис. 13.15). Вы его вводите, запускается конфигуратор с правами root, и вы успешно производите настройку системы.

А что делать, если вам нужно отредактировать вручную какой-нибудь файл конфигурации, например `/boot/grub/menu.lst`? Если вы его откроете в текстовом редакторе, например в `gedit`, то не сможете потом сохранить изменения, поскольку у вас нет прав доступа к каталогу `/boot` (точнее, нет права изменять файлы в этом каталоге). Короче, вам нужны права root.

Чтобы их получить, откройте **Терминал** (щелчок правой кнопки мыши на рабочем столе, команда **Open in Terminal**). Затем введите команду:

```
su
```

После этого программа `su` запросит у вас пароль пользователя root. При вводе пароля в терминале он не отображается на экране — просто введите пароль и нажмите

клавишу <Enter>. Теперь вы можете вводить команды от имени пользователя root. В нашем случае для редактирования файла /boot/grub/menu.lst нужно ввести команду:

```
gedit /boot/grub/menu.lst
```

Если вы работаете за компьютером один, то можете смело использовать команду `su`. Но бывают ситуации, когда нужно предоставить возможность настройки компьютера другому пользователю, но вы не хотите сообщать ему пароль root. В этом случае на помощь приходит команда `sudo`. После ввода команды `sudo` нужно ввести *свой пароль*, а не пароль root. Понятно, что право использовать `sudo` имеет не каждый пользователь, а только указанные в файле /etc/sudoers (файл редактируется не вручную, а с помощью конфигуратора **YaST | Sudo**). Но по умолчанию в openSUSE в данном файле установлена политика, разрешающая использовать `sudo` всем пользователям системы (рис. 13.16). Да, это неправильно с точки зрения безопасности, но вполне приемлемо для домашнего компьютера.

Выполнять команду `sudo` нужно так:

```
sudo команда_которую_нужно_выполнить_с_правами_root
```

Например,

```
sudo gedit /boot/grub/menu.lst
```

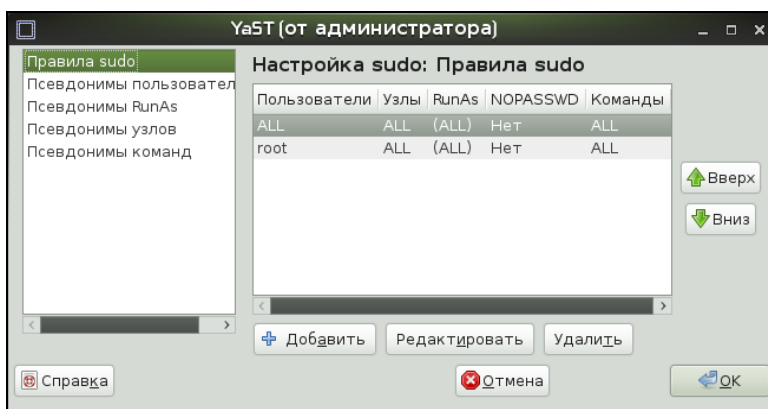


Рис. 13.16. Использовать команду `sudo` могут все пользователи

В группе **Пользователи и безопасность** конфигуратора YaST имеется конфигуратор **Локальная безопасность**. В разделе **Обзор безопасности** (рис. 13.17) содержится список настроек, касающихся безопасности системы. Для максимальной безопасности выберите следующие установки:

- ◆ **Использовать безопасные разрешения файлов** — в файлах /etc/permissions.* содержатся разрешения файлов. Самые жесткие разрешения находятся в файлах `secure` или `paranoid`;
- ◆ **Запускать демон DHCP в chroot** — демон DHCP будет запускаться в `chroot`-окружении. Даже если его взломают, злоумышленник не сможет добраться до основной файловой системы компьютера;

- ❖ **Отключить удаленный доступ к X-серверу** — не выбирайте эту опцию, если планируете предоставить удаленный доступ к своему компьютеру (см. приложение 2);
- ❖ **Отключить IPv6-переедресацию** — IPv6 пока не используется, поэтому переедресация IPv6 не нужна.

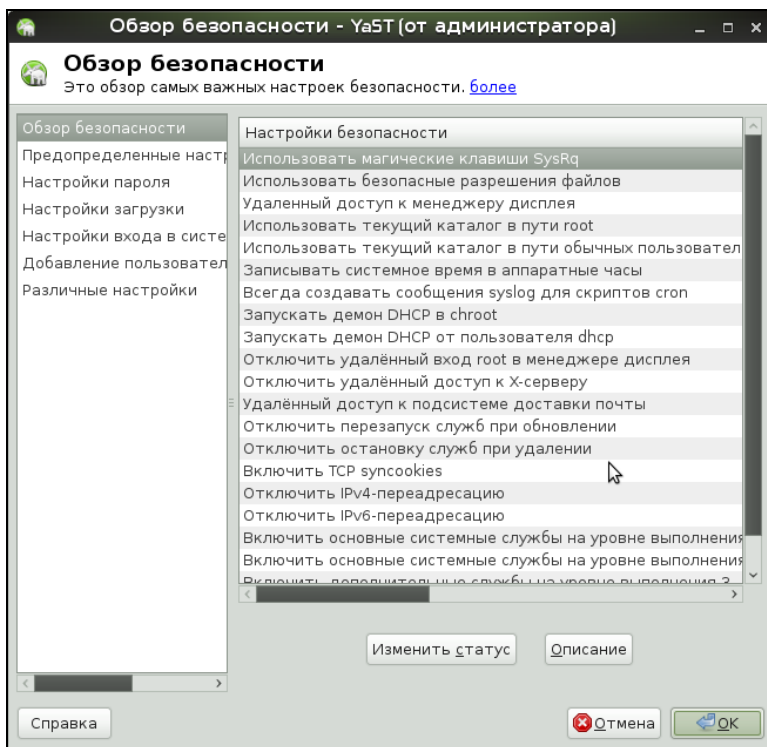


Рис. 13.17. Локальная безопасность

В разделе **Предопределенные настройки** вы можете выбрать параметры безопасности для домашнего компьютера, для рабочей станции и для сервера сети. По умолчанию используются пользовательские настройки, определенные ранее.

Раздел **Настройки пароля** (рис. 13.18) позволяет изменить параметры паролей — например, выбрать другой метод шифрования (хотя используемый по умолчанию Blowfish является самым безопасным), установить "возраст" пароля.

В разделе **Настройки загрузки** вы можете установить реакцию на нажатие комбинации клавиш <Ctrl>+<Alt>+. Выключить реакцию на нажатие этой комбинации клавиш целесообразно на сервере, чтобы никто случайно его не перезагрузил.

Параметры из разделов **Настройки входа в систему** и **Добавление пользователей** вы вряд ли будете изменять, а вот в разделе **Различные настройки** имеется параметр **Разрешить магические клавиши SysRq** — включите его, если ваша

система часто зависает, и вам нужно контролировать процесс ее "разгрузки", когда система находится в "полузависшем" состоянии. С "магическими" клавишами можно познакомиться в книге "Linux. От новичка к профессионалу"¹ или по адресу <http://www.dkws.org.ua/index.php?page=show&file=soveti/s21>.

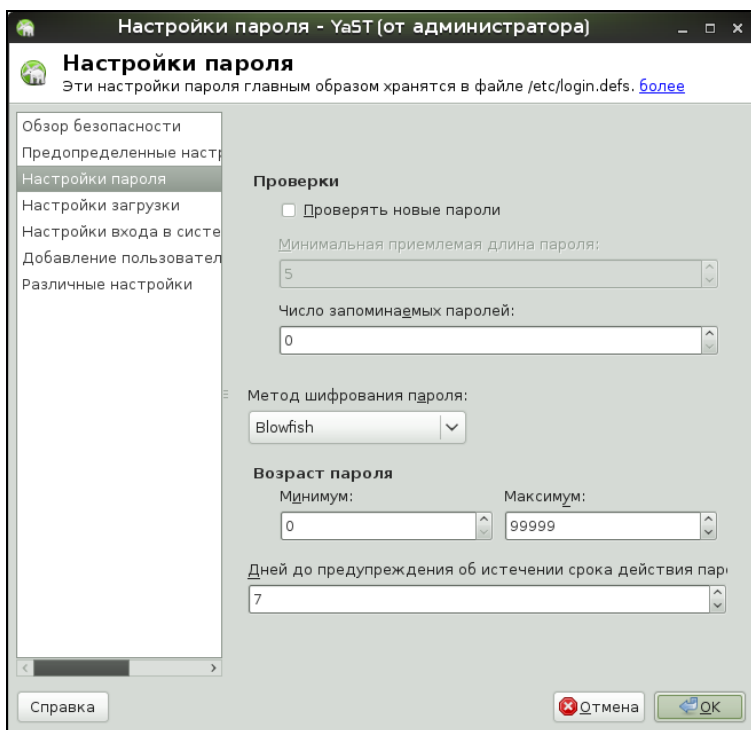


Рис. 13.18. Параметры пароля

ПРИМЕЧАНИЕ

На рис. 13.17 виден параметр **Использовать магические клавиши SysRq** в разделе **Обзор безопасности**. Это то же самое, что и параметр **Разрешить магические клавиши SysRq** в разделе **Различные настройки**. Почему один и тот же параметр называется по-разному, мне не понятно. Разработчикам виднее.

¹ Колисниченко Д. Linux. От новичка к профессионалу. — СПб.: БХВ-Петербург, 2010.

ГЛАВА 14



Пользователь root

14.1. Максимальные полномочия

Пользователь root обладает максимальными полномочиями в системе. Система полностью подвластна этому пользователю. Любая команда будет безоговорочно выполнена системой. Поэтому работать под именем пользователя root нужно с осторожностью. Всегда думайте над тем, что собираетесь сделать. Если вы дадите команду на удаление корневой файловой системы, система ее выполнит. Если же вы попытаетесь выполнить определенную команду, зарегистрировавшись под именем обычного пользователя, система сообщит вам, что у вас нет полномочий.

Представим, что кто-то решил пошутить и выложил в Интернете (записал на диск или прислал по электронной почте — не важно) вредоносную программу. Если вы ее запустите от имени пользователя root, система может быть уничтожена. Запуск этой же программы от имени обычного пользователя ничего страшного не произведет — система просто откажется ее выполнять. Или же все может быть намного проще — вы ошибочно введете команду, которая разрушит вашу систему. Или просто отойдете ненадолго от своего компьютера, а тут сразу же появится доброжелатель, — имея полномочия пользователя root, уничтожить систему можно одной командой.

Именно поэтому практически во всех современных дистрибутивах вход под именем пользователь root запрещен. В одних дистрибутивах вы не можете войти как root в графическом режиме (но можете войти в консоли, переключившись на первую консоль с помощью комбинации клавиш <Ctrl>+<Alt>+<F1>), а в других вообще не можете войти в систему как root — ни в графическом режиме, ни в консоли (пример такого дистрибутива — Ubuntu).

Отсюда можно сделать следующие выводы:

- ❖ старайтесь реже работать пользователем root;
- ❖ всегда думайте, какие программы вы запускаете под именем root;
- ❖ если программа, полученная из постороннего источника, требует root-полномочий, это должно насторожить;

- ❖ создайте обычного пользователя (даже если вы сами являетесь единственным пользователем компьютера) и рутинные операции (с документами, использование Интернета и т. д.) производите от имени этого пользователя;
- ❖ если полномочия root все же нужны, совсем необязательно заходить в систему под этим пользователем, достаточно запустить терминал и выполнить команду `sudo` или `su` (см. разд. 14.2). После этого в терминале можно выполнять команды с правами root. Если вы закроете терминал, то больше не сможете работать с правами root. Очень удобно — ведь обычно права root нужны для одной-двух операций (например, выполнить команду установки программы или создать/удалить пользователя).

14.2. Как работать без root

Некоторые операции, например, установка программного обеспечения, изменение конфигурационных файлов, требуют полномочий root. Чтобы их временно получить, нужно использовать команды `sudo` или `su` (эти команды, скорее всего, вы будете запускать в терминале).

14.2.1. Команда `sudo`

Команда `sudo` позволяет запустить любую команду с привилегиями root. Использовать ее нужно так:

```
sudo <команда_которую_нужно_выполнить_с_правами_root>
```

Например, вам необходимо изменить файл `/etc/apt/sources.list`. Для этого используется команда:

```
sudo gedit /etc/apt/sources.list
```

ПРИМЕЧАНИЕ

Программа `gedit` — это текстовый редактор, мы ему передаем один параметр — имя файла, который нужно открыть.

Если ввести эту же команду, но без `sudo` (просто `gedit /etc/apt/sources.list`), текстовый редактор тоже запустится и откроет файл, но сохранить изменения вы не сможете, поскольку у вас не хватит полномочий.

Программа `sudo` перед выполнением указанной вами команды запросит у вас пароль:

```
sudo gedit /etc/apt/sources.list
```

Password:

Вы должны ввести свой *пользовательский пароль* — тот, который применяете для входа в систему, но не пароль пользователя root (кстати, мы его и не знаем).

Использовать команду `sudo` имеют право не все пользователи, а только те, которые внесены в файл `/etc/sudoers`. Администратор системы (пользователь `root`) может редактировать этот файл с помощью команды `visudo`. Если у вас дистрибутив, который запрещает вход под учетной записью `root` (следовательно, у вас нет возможности отредактировать файл `sudoers`), то в файл `sudoers` вносятся пользователи, которых вы добавили при установке системы.

Чтобы команду `sudo` могли применять все пользователи, в файл `/etc/sudoers` нужно добавить строку:

```
username      ALL=(ALL)      ALL
```

Ради справедливости нужно отметить, что файл `/etc/sudoers` можно редактировать любым другим редактором, например для редактирования этого файла графическим текстовым редактором `gedit` нужно ввести команды:

```
su
gedit /etc/sudoers
```

Но команда `visudo` предотвращает всевозможные коллизии при редактировании этого файла. Например, во время редактирования этого файла другим редактором (любым другим, кроме `visudo`) система может внести какие-либо изменения в этот файл. Как будет работать система после внесения в `sudoers` изменения с учетом того, что вы добавите в обычном текстовом редакторе — одному Богу известно. Так что лишний раз лучше не рисковать и использовать команду `visudo`. Команда `visudo` для редактирования `/etc/sudoers` вызывает текстовый редактор, установленный в переменной окружения `EDITOR`. По умолчанию это стандартный и очень неудобный редактор `vi`. О том, как его использовать, см. в главе 20. Но вы можете изменить значение переменной `EDITOR` (как это сделать, см. в главе 21) и выбрать любой другой редактор, например `nano` — компактный и очень удобный редактор.

Команда `su`

Команда `su` позволяет получить доступ к консоли `root` любому пользователю (даже если пользователь не внесен в файл `/etc/sudoers`) при условии, что он знает пароль `root`. Понятно, что в большинстве случаев этим пользователем будет сам пользователь `root` — не будете же вы всем пользователям доверять свой пароль? Поэтому команда `su` предназначена, в первую очередь, для администратора системы, а `sudo` — для остальных пользователей, которым иногда нужны права `root` (чтобы они меньше отвлекали администратора от своей работы).

Использовать команду `su` просто:

```
su
```

После этого нужно ввести пароль пользователя `root`, и вы сможете работать в консоли, как обычно. Использовать `su` удобнее, чем `sudo`, потому что вам не нужно вводить `su` перед каждой командой, которая должна быть выполнена с правами `root`.

Чтобы закрыть сессию `su`, нужно или ввести команду `exit`, или просто закрыть окно терминала.

В случае, если вы запускаете какую-нибудь графическую программу, требующую привилегий `root`, тогда вы увидите окно с требованием ввести свой пароль, подобное изображенному на рис. 14.1.

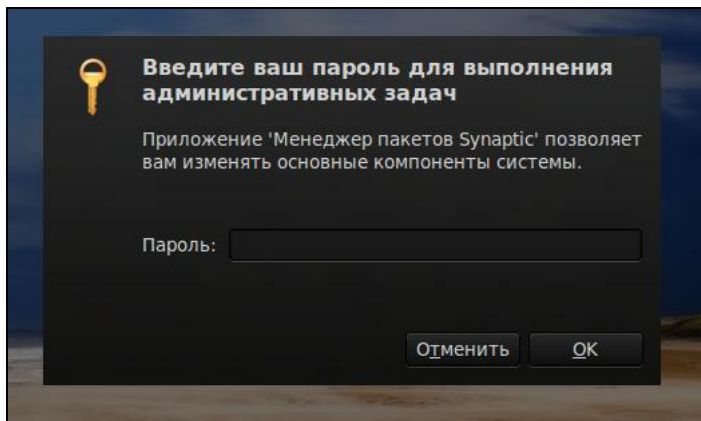


Рис. 14.1. Требование ввести пароль

14.2.2. Проблемы с `sudo` в Ubuntu и Kubuntu

Если вы в терминале хотите запустить графическую программу с правами `root` (например, `gedit`), желательно использовать не программу `sudo`, а программу `gksudo` (`gksu` — для Ubuntu или `kdesu` — для Kubuntu). Программа `sudo` не всегда корректно работает с графическими приложениями, поэтому рано или поздно вы можете получить сообщение "Unable to read ICE authority file", и после этого вообще станет невозможным запуск графических программ с правами `root`. Если это все же произошло, поправить ситуацию можно, удалив файл `.{ICE,X}authority` из вашего домашнего каталога:

```
rm ~/.{ICE,X}authority
```

Напомню, что тильда здесь означает домашний каталог текущего пользователя.

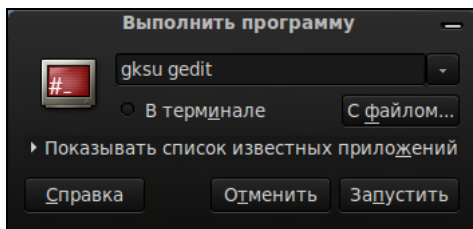


Рис. 14.2. Быстрое выполнение программы

Графические приложения с правами root проще запускать, используя главное меню. Но не все приложения есть в главном меню, или не все приложения вызываются с правами root — например, в главном меню есть команда вызова текстового редактора, но нет команды для вызова текстового редактора с правами root. Поэтому намного проще нажать клавиатурную комбинацию `<Alt>+<F2>` и в открывшемся диалоговом окне **Выполнить программу** ввести команду в соответствующее поле (рис. 14.2).

`gksu <команда>`

14.2.3. Ввод серии команд *sudo*

Вам надоело каждый раз вводить `sudo` в начале команд? Тогда выполните команду:

```
sudo -i
```

Данная команда запустит оболочку root, т. е. вы сможете вводить любые команды, и они будут выполнены с правами root. Обратите внимание, что изменился приглашение командной строки (рис. 14.3). До этого приглашение имело вид `$`, что означало, что вы работаете от имени обычного пользователя, а после выполнения программы приглашение изменилось на `#` — это верный признак того, что каждая введенная команда будет выполнена с правами root.

Опция `-i` позволяет так же удобно вводить команды, как если бы вы использовали команду `sudo`.

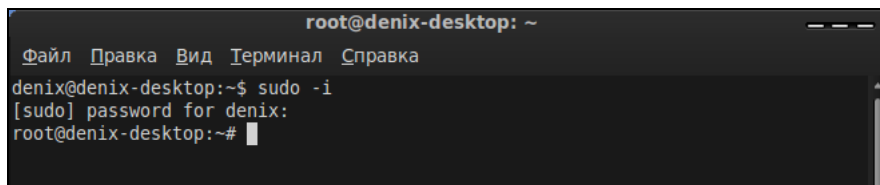


Рис. 14.3. Оболочка root

14.3. Переход к традиционной учетной записи root

14.3.1. Преимущества и недостатки *sudo*

Как уже было отмечено, во многих дистрибутивах учетная запись root немного ограничена. В одних дистрибутивах она отключена, и для получения необходимых полномочий нужно использовать команду `sudo`, в других ограничивается

использование учетной записи, например невозможно войти в графическом режиме как `root`.

Тем не менее, возможность перейти к традиционной учетной записи `root`, т. е. заходить в систему под именем `root`, как вы заходите под именем обычного пользователя, имеется всегда. Чуть позже мы поговорим о том, как это сделать, но сначала рассмотрим преимущества (и недостатки) использования команды `sudo`.

К преимуществам `sudo` можно отнести следующие соображения:

- ❖ вам не нужно помнить несколько паролей (т. е. ваш пароль и пароль пользователя `root`) — вы помните только свой пароль и вводите его, когда нужно;
- ❖ с помощью `sudo` вы можете выполнять практически те же действия, что и под именем `root`, но перед каждым действием у вас будет запрошен пароль, что позволит еще раз подумать о правильности своих действий;
- ❖ каждая команда, введенная с помощью `sudo`, записывается в журнал `/var/log/auth.log`, поэтому в случае чего вы хотя бы будете знать, что случилось, прочитав этот журнал. У вас также будет храниться история введенных команд с полномочиями `root`, в то время как при работе под именем `root` никакой журнал не ведется;
- ❖ предположим, некто захотел взломать вашу систему. Этот некто не знает, какие учетные записи есть в вашем компьютере, зато уверен, что учетная запись `root` есть всегда. Знает он также, что, завладев паролем к этой учетной записи, можно получить неограниченный доступ к системе. Но не к вашей системе — у вас учетная запись `root` отключена!
- ❖ вы можете разрешать и запрещать другим пользователям использовать полномочия `root` (позже мы разберемся, как это сделать), не предоставляя пароль `root`, т. е. практически нет риска скомпрометировать учетную запись `root` (впрочем, риск есть всегда, ведь при неправильно настроенной системе с помощью команды `sudo` можно легко изменить пароль `root`).

Но у `sudo` есть и недостатки.

- ❖ Неудобно использовать перенаправление ввода/вывода, например команда:

```
sudo ls /etc > /root/somefile
```

работать не будет, вместо нее нужно использовать команду:

```
sudo bash -c "ls /etc > /root/somefile"
```

Длинновато, правда?

- ❖ Имеются и неудобства, связанные с технологией NSS. К счастью, она используется не очень часто, поэтому основной недостаток `sudo` будет связан только с перенаправлением ввода/вывода.

14.3.2. Традиционная учетная запись `root` в Ubuntu

Вы все-таки хотите использовать обычную учетную запись `root`? Для этого достаточно задать пароль для пользователя `root`. Делается это командой:

```
sudo passwd root
```

Сначала программа запросит ваш пользовательский пароль, затем новый пароль root и его подтверждение:

Enter your existing password:

Enter password for root:

Confirm password for root:

После этого вы сможете входить в систему под учетной записью root.

Для отключения учетной записи root используется команда:

```
sudo passwd -l root
```

Помните, что после закрытия учетной записи root у вас могут быть проблемы с входом в систему в режиме восстановления, поскольку пароль root уже установлен (т. е. он не пустой, как по умолчанию), но в то же время учетная запись закрыта. Поэтому, если вы уже включили учетную запись root, то будьте внимательны и осторожны. А вообще лучше ее не включать, а пользоваться командой `sudo -i`.

14.3.3. Традиционная учетная запись root в Mandriva

В Ubuntu учетная запись root отключена честно. В Linux Mandriva 2010 отключена лишь возможность графического входа в систему под именем root. Другими словами, вы можете переключиться в консоль, нажав клавиатурную комбинацию `<Ctrl>++<Alt>+<F1>`, и войти в систему под именем root.

Тем не менее, и в Mandriva 2010 можно войти под именем root в графическом режиме. За регистрацию пользователей в системе в графическом режиме отвечает KDM (KDE Display Manager, дисплейный менеджер KDE), он-то и не пускает пользователя root в систему. Для изменения поведения KDM нужно открыть его конфигурационный файл. Это следует сделать с привилегиями root:

```
su
```

```
kwrite /etc/kde/kdm/kdmrc (для Mandriva 2008)
```

```
kwrite /etc/alternatives/kdm4-config (для Mandriva 2009/2010)
```

В этом файле найдите строку:

```
AllowRootLogin=false
```

Значение директивы `AllowRootLogin` измените на `true`:

```
AllowRootLogin=true
```

После этого можно будет войти в систему под именем root (рис. 14.4).

ПРИМЕЧАНИЕ

В ранних версиях Mandriva при входе с правами root вы получали предупреждение, а фон графического стола становился красным. В Mandriva 2010 ни предупреждения, ни каких-либо других визуальных изменений не будет. На рис. 14.4 изображен домашний каталог текущего пользователя — это каталог root, принадлежащий пользователю root. Следовательно, вход с учетной записью root выполнен успешно (но другие пользователи просмотреть этот каталог не могут). Также приводится вывод команды `whoami`, сообщающей имя текущего пользователя — root.

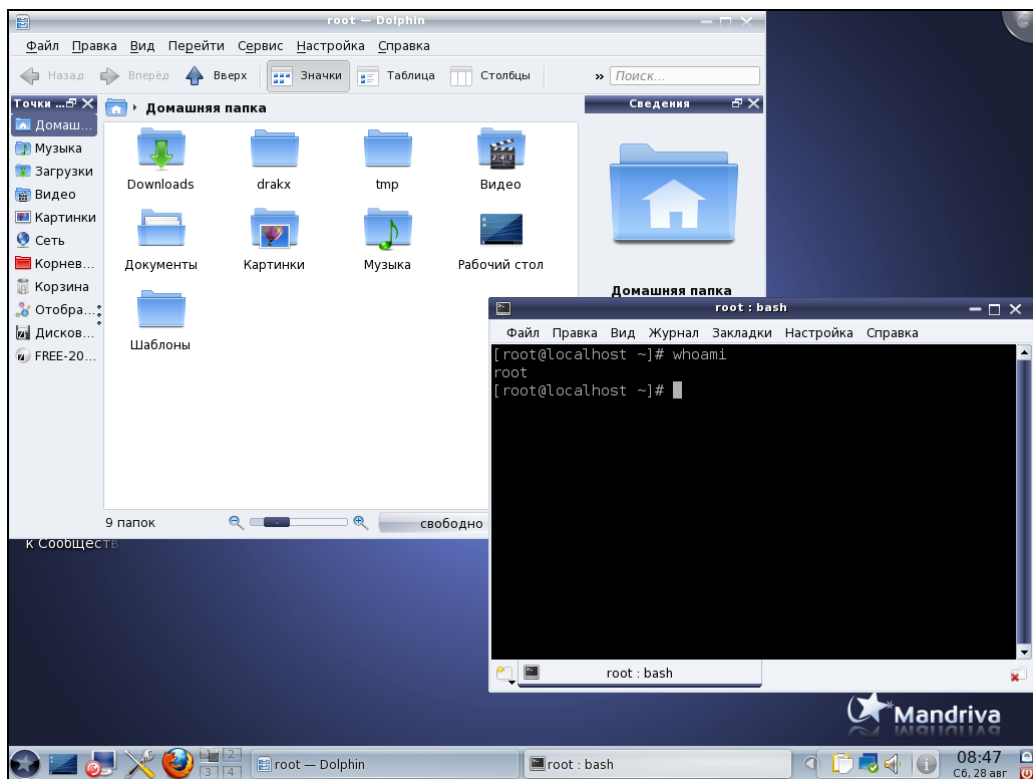


Рис. 14.4. Вход в графическом режиме под именем root в Mandriva 2010.1

14.3.4. Вход в качестве root в Fedora

Как и в Mandriva, в Fedora 9 и 10 вход пользователя root ограничен менеджером рабочего стола. Введите команду:

```
su -c 'gedit /etc/pam.d/gdm'
```

Вы запустите с правами root текстовый редактор gedit для редактирования файла /etc/pam.d/gdm. Найдите в этом файле следующую строку

```
auth required pam_succeed_if.so user != root quiet
```

Закомментируйте ее (поставьте знак # перед ней) или вообще удалите эту строку.

В Fedora 11—13 дополнительно нужно открыть файл /etc/pam.d/gdm-password и найти следующую строку (рис. 14.5):

```
pam_succeed_if.so user != root quiet
```

Эту строку тоже нужно или закомментировать, или удалить.

Если вы используете вход в систему по отпечатку пальца, тогда откройте файл gdm-fingerprint и закомментируйте в нем следующую строку:

```
pam_succeed_if.so user != root quiet
```

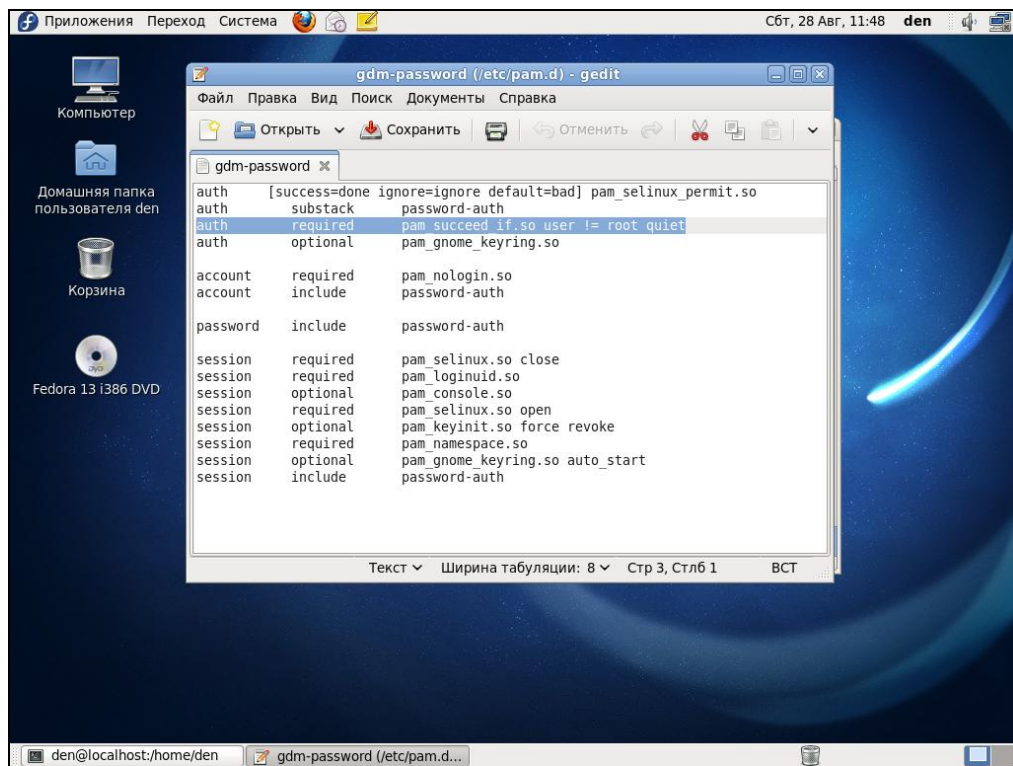


Рис. 14.5. Редактирование файла /etc/pam.d/gdm-password

После этого сохраните файлы и завершите сеанс пользователя. После перезагрузки GDM вы сможете войти в систему как root.

Просмотреть видеоурок, демонстрирующий разрешение входа root в графическом режиме, можно по адресу: http://dkws.org.ua/video-lessons/login_as_root_in_fedora_12.avi.

ПРИМЕЧАНИЕ

Для просмотра видео необходим кодек VMWare Movie Decoder, скачать который можно по адресу: <http://download3.vmware.com/software/wkst/VMware-moviedecoder-5.5.0-18463.exe>

ГЛАВА 15



Ограничение дискового пространства

15.1. Квотирование — это полезно!

На практике довольно часто возникает потребность ограничить дисковое пространство пользователей. Предположим, что вы — администратор сервера хостинг-провайдера, и вам нужно ограничить дисковое пространство, доступное пользователю, согласно его тарифному плану. Кто-то купил хостинг на 50 Мбайт, а кто-то — на 1 Гбайт. Вам нужно задать соответствующие квоты.

Механизм квотирования в Linux работает на уровне ядра. Современные ядра собраны с поддержкой квот, поэтому вам не придется перекомпилировать ядро вашего сервера, чтобы включить квотирование. Однако в процессе настройки пара перезагрузок все же понадобится. Об этом вы должны знать: одно дело, когда настраиваешь "голый" сервер, и совсем другое, когда работы ведутся на уже настроенном сервере, и пользователи то и дело обращаются к его ресурсам. Поэтому не забудьте перед настройкой сервера оповестить ваших пользователей о том, что некоторое время сервер будет недоступен — чтобы они не волновались и лишний раз не звонили вам, узнавая причину "сбоя" сервера.

15.2. Включение квот

Для включения квотирования достаточно установить пакет `quota` (рис. 15.1) и включить квотирование в файле `/etc/fstab`. Для установки пакетов в Mandriva используется программа `rpm-drake`, в Fedora — `gpk-application` (в старых версиях — `system-config-packages`), а в Ubuntu — Synaptic (**Система | Администрирование | Менеджер пакетов Synaptic**). С установкой пакета все просто — несмотря на то, что мы еще не рассматривали установку пакетов (*см. часть VII*), не думаю, что у вас возникнут какие-нибудь сложности с этим.

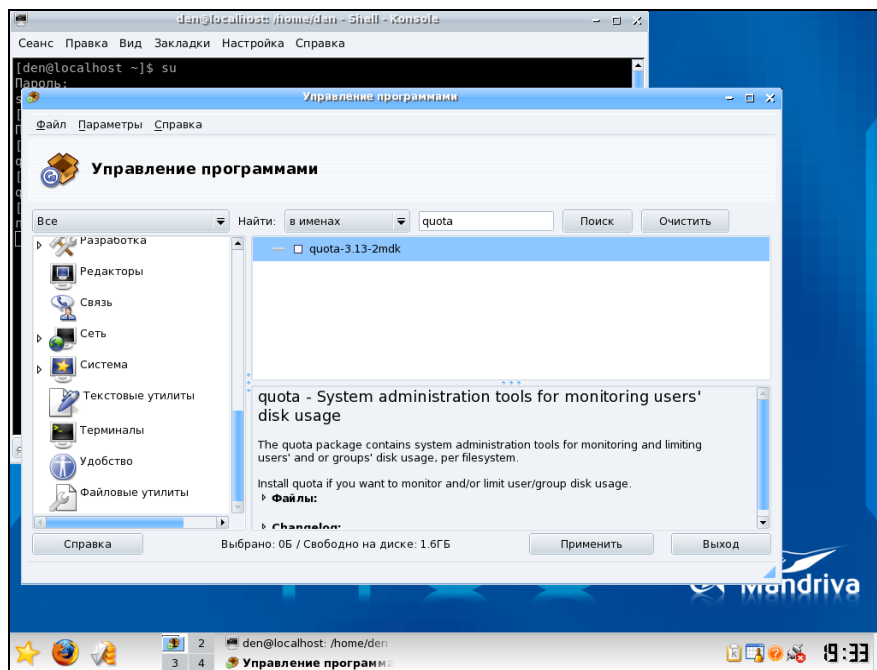


Рис. 15.1. Установка пакета quota в Mandriva

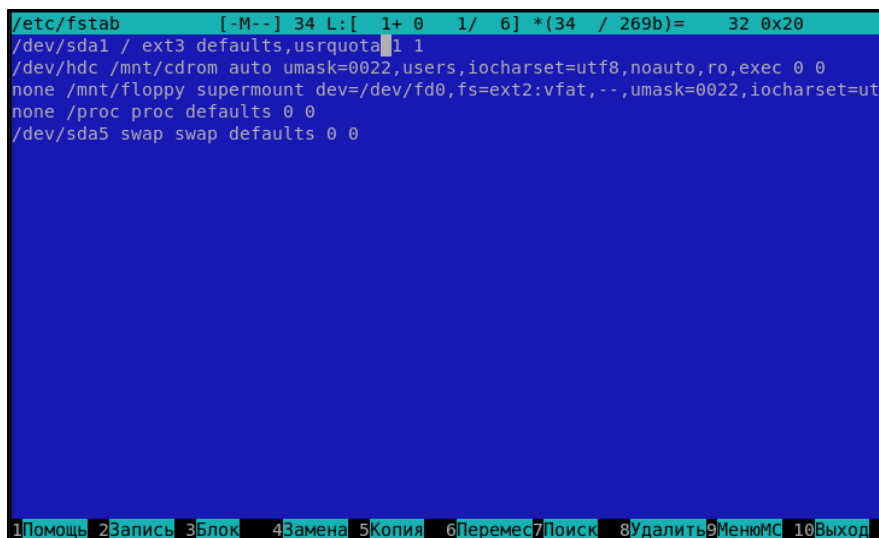


Рис. 15.2. Редактирование /etc/fstab в mcedit

А теперь поговорим о редактировании файла `/etc/fstab`. Предположим, что у вас есть несколько файловых систем: например, одна — корневая, а вторая содержит домашние каталоги пользователей (точка монтирования `/home`). Понятно,

что включить квотирование нужно только для второй файловой системы, поэтому откройте файл `/etc/fstab` (для записи в этот файл нужны права root):

```
$ su
Password:
# текстовый_редактор /etc/fstab
```

В качестве текстового редактора вы можете использовать `kedit` (если у вас KDE), `gedit` (если у вас GNOME) или `mcedit` (если установлен пакет `mc`). В терминале я рекомендую использовать простенький текстовый редактор `mcedit` (рис. 15.2).

Для файловой системы `/home` вы должны указать параметр `usrquota`, например, `/dev/sda2 /home ext4 defaults,usrquota 0 1`

Параметр `usrquota` включает поддержку квот для отдельных пользователей, если вам нужна поддержка квот групп пользователей, тогда добавьте параметр `grpquota`.

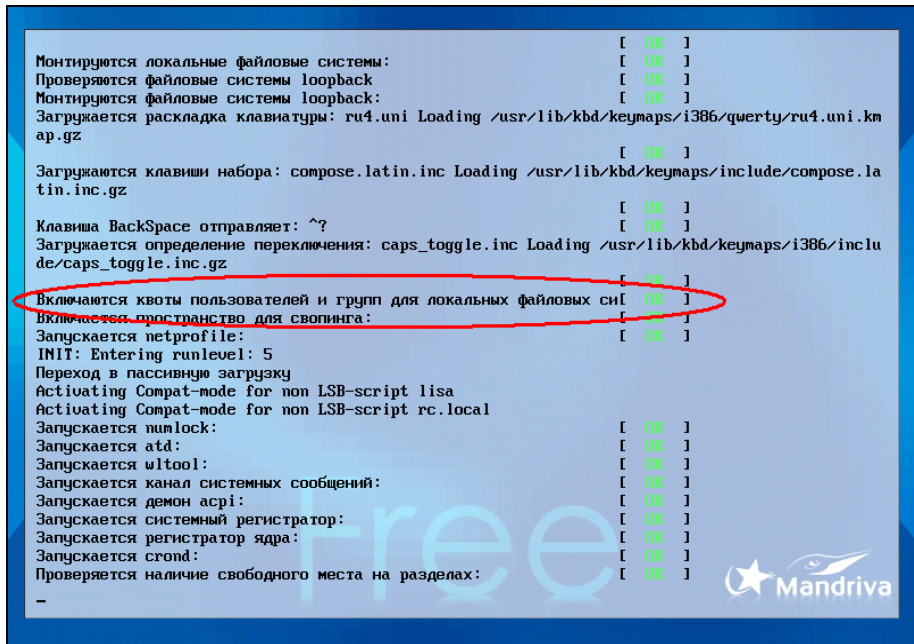


Рис. 15.3. Квоты включены

Если у вас всего одна файловая система — корневая, тогда параметр `usrquota` нужно добавить для нее, как это показано на рис. 15.2.

После этого включим квотирование для наших файловых систем:

```
# quotaon файловая_система
```

Например,

```
# quotaon /
```

После этого перезагружаем систему:

```
# reboot
```

При загрузке вы увидите сообщение: "Включаются квоты пользователей и групп для локальных файловых систем" ("Turning on user and group quotas for local filesystems"). Это означает, что механизм квотирования правильно работает (рис. 15.3).

15.3. Задание и просмотр квот

Теперь приступим к самому интересному — заданию квот для отдельных пользователей и групп пользователей. Но сначала введите команду `repquota -ua` для просмотра текущих квот (рис. 15.4).

```
[root@localhost /]# repquota -ua
*** Report for user quotas on device /dev/sda1
Block grace time: 7days; Inode grace time: 7days
```

User		used	Block limits		grace	File limits		
			soft	hard		used	soft	hard
root	--	1113796	0	0		84713	0	0
daemon	--	3	0	0		3	0	0
rpm	--	39078	0	0		59	0	0
avahi	--	2	0	0		3	0	0
xfs	--	2	0	0		3	0	0
clamav	--	5812	0	0		11	0	0
den	--	1528	0	0		258	0	0
max	--	9	0	0		6	0	0

```
[root@localhost /]#
```

Рис. 15.4. Просмотр квот

Как видно из рис. 15.4, квоты не заданы, поэтому самое время их задать. Для задания квоты отдельного пользователя введите команду:

```
# edquota -u ИМЯ_ПОЛЬЗОВАТЕЛЯ
```

Если нужно задать квоту для группы, введите команду:

```
# edquota -g ИМЯ_ГРУППЫ
```

Откроется текстовый редактор `vi` (рис. 15.5). Об этом редакторе вы должны знать следующее: это самый "древний" текстовый редактор для UNIX и вдобавок самый неудобный. В книге мы его рассматривать не будем: я вам расскажу лишь, как отредактировать квоты пользователя, а если вам захочется узнать об этом редакторе больше (в чем я глубоко сомневаюсь), вы всегда сможете прочитать в Интернете вот этот документ: <http://www.gentoo.org/doc/ru/vi-guide.xml>.

Итак, для задания квоты пользователя/группы с помощью клавиш управления курсором подведите курсор к нулю, стоящему под словом "soft". Нажмите клавишу `<I>`. В нижней части окна появится надпись **ВСТАВКА**. Задайте новое значение,


```

Отчеты об ошибках отправляйте по адресу: bonzini@gnu.org .
Убедитесь, что включили где-либо в поле ``Тема:'' слово ``sed''.
[root@localhost bin]# mc /etc/fstab

[root@localhost bin]#
[root@localhost bin]# edquota -u max
[root@localhost bin]# repquota -ua
*** Report for user quotas on device /dev/sda1
Block grace time: 7days; Inode grace time: 7days

```

User		used	Block limits		grace	File limits			
			soft	hard		used	soft	hard	grace
root	--	1116168	0	0		84719	0	0	
daemon	--	3	0	0		3	0	0	
rpm	--	39078	0	0		59	0	0	
avahi	--	2	0	0		3	0	0	
xfs	--	2	0	0		3	0	0	
clamav	--	5812	0	0		11	0	0	
den	--	1529	0	0		258	0	0	
max	--	9	100	150		6	0	0	

```

[root@localhost bin]#

```

Рис. 15.6. Просмотр квот после редактирования

Установив ограничения, введите еще раз команду `repquota -ua` — нужно убедиться, что изменения задействованы (рис. 15.6).

15.4. Прототипы

Наверняка использовать `vi` вам не очень нравится. Значительно упрощают задание квот так называемые прототипы. Например, вы задали ограничение для пользователя `max`. Но у вас есть еще несколько пользователей, для которых нужно задать такие же ограничения. Вы можете использовать квоту пользователя `max` в качестве прототипа:

```

# edquota -p max user1
# edquota -p max user2
...

```



ЧАСТЬ IV

ЗАГРУЗКА И ИНИЦИАЛИЗАЦИЯ LINUX

ГЛАВА 16



Глава 16. Загрузчики Linux

16.1. Основные загрузчики

Основное назначение загрузчика — запуск выбранной пользователем операционной системы. Наиболее популярным загрузчиком сегодня является GRUB, который мы здесь подробно рассмотрим. В более старых дистрибутивах по умолчанию применялся загрузчик LILO. Списывать со счета LILO пока нельзя, поскольку еще много систем используют именно его, да и в современных дистрибутивах присутствует возможность установить старый добрый LILO. Многие администраторы по привычке ставят LILO вместо более современного GRUB. Однако в данной книге загрузчик LILO рассмотрен не будет. Если он вам нужен, тогда рекомендую прочитать мою книгу "Linux. От новичка к профессионалу" (<http://bhv.ru/books/book.php?id=186944>)¹.

Кроме LILO и GRUB некоторые дистрибутивы могут включать собственные загрузчики — например в ASPLinux таковым является ASPLoader. Подобные загрузчики мы рассматривать не будем, поскольку в большинстве случаев в дистрибутивах, использующих собственные загрузчики, имеется возможность установки GRUB или LILO.

Загрузчик GRUB (GRand Unified Bootloader) считается более гибким и современным, чем LILO. Благодаря иной схеме загрузки операционных систем GRUB "понимает" больше файловых систем, нежели LILO, а именно: FAT/FAT32, ext2, ext3, ReiserFS, XFS, BSDFS и др.

Время не стоит на месте. В свое время загрузчик GRUB пришел на смену LILO, поскольку последний не поддерживал загрузки с разделов, начинающихся после 1024-го цилиндра. Об этой проблеме знает, наверное, каждый Linux-пользователь — ведь всего несколько лет назад она была актуальной (пока все дистрибутивы не перешли на GRUB). Точно такая же участь постигла и GRUB — на его ме-

¹ Колисниченко Д. Linux. От новичка к профессионалу (+DVD-ROM). — СПб.: БХВ-Петербург, 2010.

сто пришел GRUB2, умеющий загружаться с файловой системы ext4. А загрузка с ext4-разделов просто необходима современному дистрибутиву.

GRUB2 — это не просто набор патчей для GRUB, а полностью новая разработка, созданная с "нуля". Именно поэтому у GRUB2 совершенно другой формат конфигурационного файла.

Разработка "обычного" GRUB полностью прекращена, к нему выпускаются лишь патчи. Да, можно скачать патч, добавляющий к GRUB загрузку с разделов ext4. Так, в Ubuntu 9.10, где по умолчанию впервые был установлен GRUB2, я его удалил (с сохранением конфигурационных файлов), затем установил GRUB (имеющаяся в составе версии 9.10 версия GRUB поддерживает ext4), создал вручную его конфигурационный файл и перезагрузил систему — она загрузилась без ошибок. Но, учитывая, что будущее все-таки за GRUB2, я вернул его обратно на заслуженное место.

ПРИМЕЧАНИЕ

В Ubuntu GRUB2 используется, начиная с версии 9.10 — не зря я упомянул ее ранее. И в этой версии Ubuntu, и в новой — 10.04 — имеется один небольшой "глюк", связанный с установкой тайм-аута выбора операционной системы. Чуть позже мы решим эту проблему, а пока приступим к рассмотрению конфигурационных файлов GRUB2.

ПРИМЕЧАНИЕ

На самом деле то, что называется GRUB2 — это GRUB v1.98. То есть почти вторая версия, а когда выйдет вторая версия (в смысле 2.0), пока никто не знает. Хотя ведущие разработчики дистрибутивов уже включили GRUB2 в состав дистрибутивов, что говорит о его надежности.

16.2. Конфигурационные файлы GRUB и GRUB2

16.2.1. Конфигурационный файл GRUB

Конфигурационным файлом GRUB служит файл `/boot/grub/grub.conf` (в старых версиях — `/boot/grub/menu.lst`; впрочем, `menu.lst` в новых версиях — это ссылка на `grub.conf`). Рассмотрим пример этого файла (листинг 16.1).

Думаю, не стоит говорить о том, что конфигурационные файлы загрузчика нужно редактировать только с правами `root`. В некоторых дистрибутивах, например в Fedora 13, конфигурационный файл `grub.conf` обычный пользователь не может даже просмотреть. И это правильно, поскольку в этом файле могут содержаться незашифрованные (если администратор поленился их зашифровать) пароли загрузчика.

Листинг 16.1. Файл /boot/grub/grub.conf

```
# Следующие параметры будут описаны далее:
boot=/dev/hda
default=0
timeout=10
fallback=1
splashimage=(hd0,1)/grub/mysplash.xpm.gz

# по умолчанию скрывает меню (для того чтобы увидеть меню,
# нужно нажать клавишу Esc)
#hiddenmenu

# Главное загрузочное устройство GRUB (можно не указывать)
#groot=(hd0,1)

# Опции загрузчика по умолчанию (более подробно см. man menu.lst)
# defoptions=quiet splash

# опции ядра по умолчанию
# kopt=root=/dev/hda2 ro

# Предпочитаемые цвета
#color cyan/blue white/blue

title MDK
    root (hd0,1)
    kernel /vmlinuz-2.6.14-1.1263 ro root=/dev/hda2
    initrd /initrd-2.6.14-1.1263.img

title WinXP
    rootnoverify (hd0,0)
    makeactive
    chainloader+1
```

ПРИМЕЧАНИЕ

Как вы уже успели заметить, в листинге 16.1 (параметр `boot`) до сих пор используются устаревшие номера устройств — `/dev/hd*`. Но в современных дистрибутивах даже IDE-диски именуются как `/dev/sd*`. Все правильно: во времена использования GRUB еще применялась старая схема именования жестких дисков. Поэтому если вам попался дистрибутив с загрузчиком GRUB, то в большинстве случаев IDE-диски будут называться `/dev/hd*`. Исключение могут составить разве что некоторые дистрибутивы, например openSUSE, где даже в современных версиях используется обычный GRUB, а не GRUB2.

Во всех остальных современных дистрибутивах IDE-диски называются `/dev/sd*` и используется загрузчик GRUB2, поэтому никаких ошибок в листинге 16.1 нет.

Параметр `boot` указывает загрузочное устройство, а параметр `default` — загрузочную метку по умолчанию. Метка начинается параметром `title` и продолжается до следующего `title`. Нумерация меток начинается с 0. Параметр `timeout` задает количество секунд, по истечении которых будет загружена операционная система по умолчанию.

Параметр `default` полезно использовать с параметром `fallback`. Первый задает операционную систему по умолчанию, а второй — операционную систему, которая будет загружена в случае, если с загрузкой операционной системы по умолчанию произошла ошибка.

Задать графическое изображение позволяет параметр `splashimage`. Чуть позже мы разберемся, как самостоятельно создать такое изображение.

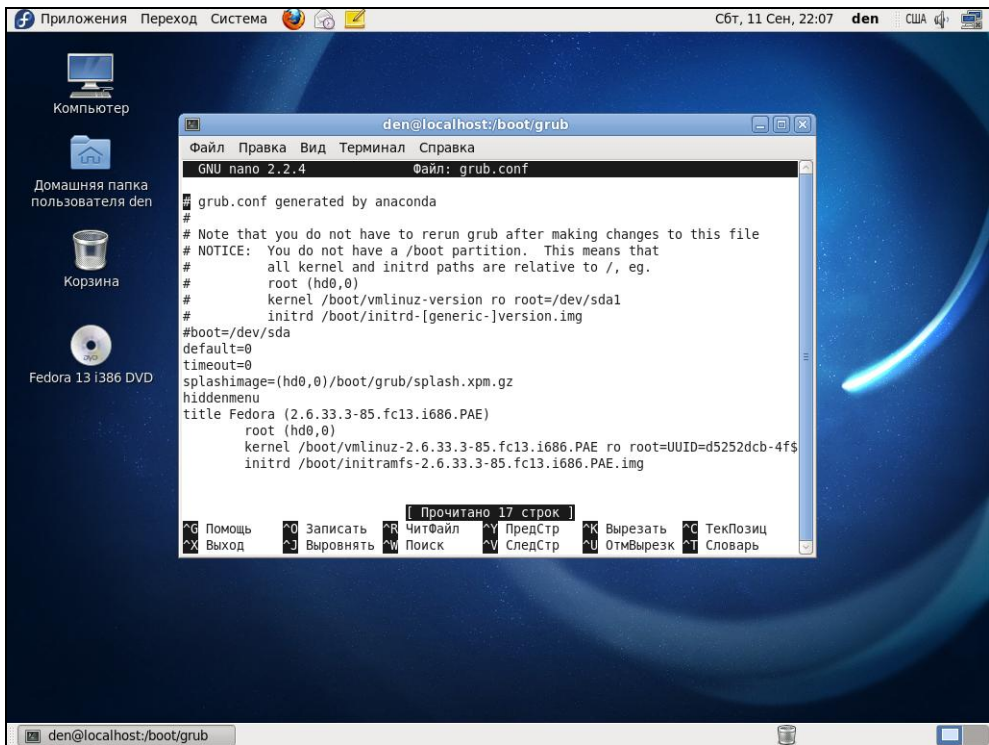


Рис. 16.1. Редактирование файла `grub.conf` в Fedora 13

При работе с GRUB вам поначалу будет трудно разобраться с именами разделов. GRUB вместо привычных `/dev/hd*` (или `/dev/sd*` для SCSI-дисков) использует свои имена. Перевести имя `/dev/hd*` в имя в формате GRUB просто. Во-первых, опускается `/dev/`. Во-вторых, устройства отсчитываются не с буквы "a", как в Linux, а с нуля. Разделы на дисках отсчитываются не с единицы, а тоже с нуля, причем

номер раздела указывается через запятую. Потом все имя берется в скобки. Например, раздел `/dev/hda1` в GRUB будет выглядеть как `(hd0,0)`, а раздел `/dev/hdb2` как `(hd1,1)`. Впрочем, об именах разделов в GRUB мы еще поговорим, но чуть позже.

Параметр `rootnoverify` указывается для Windows (точнее, для всех операционных систем не типа Linux). Параметр `chainloader` указывается для операционных систем, поддерживающих цепочечную загрузку. Если Windows на вашем компьютере установлен в неактивном разделе, с которого Windows загружаться не может, перед параметром `chainloader` нужно указать параметр `makeactive`.

ПРИМЕЧАНИЕ

К своему превеликому удивлению я обнаружил, что в Fedora 13 (а это последняя версия Fedora на момент написания этих строк) до сих пор используется обычный GRUB, а не GRUB2. С одной стороны — это хорошо, т. к. конфигурационный файл GRUB проще, чем у GRUB2, да и GRUB более привычный. С другой стороны, GRUB2 позволяет более гибко настроить процесс загрузки. На рис. 16.1 изображен процесс редактирования конфигурационного файла загрузчика в Fedora 13: как видите, используется обычный GRUB.

16.2.2. Конфигурационный файл GRUB2

В листинге 16.2 приведен основной конфигурационный файл GRUB2 — `/boot/grub/grub.cfg`. Этот конфигурационный файл не редактируется вручную. Для его создания используется утилита `/usr/sbin/grub-mkconfig`, которая генерирует этот конфигурационный файл на основе шаблонов, хранящихся в каталоге `/etc/grub.d`, и настроек из файла `/etc/default/grub`.

Листинг 16.2. Конфигурационный файл `grub.cfg`

```
#
# DO NOT EDIT THIS FILE
#
# It is automatically generated by /usr/sbin/grub-mkconfig using templates
# from /etc/grub.d and settings from /etc/default/grub
#

### BEGIN /etc/grub.d/00_header ###
if [ -s /boot/grub/grubenv ]; then
    have_grubenv=true
    load_env
fi
set default="0"
if [ ${prev_saved_entry} ]; then
    saved_entry=${prev_saved_entry}
    save_env saved_entry
```

```

    prev_saved_entry=
    save_env prev_saved_entry
fi
insmod ext2
set root=(hd0,1)
search --no-floppy --fs-uuid --set 34eaa635-ef0e-4d5c-8b61-3c22c767834b
if loadfont /usr/share/grub/unicode.pf2 ; then
    set gfxmode=640x480
    insmod gfxterm
    insmod vbe
    if terminal_output gfxterm ; then true ; else
        # For backward compatibility with versions of terminal.mod that don't
        # understand terminal_output
        terminal gfxterm
    fi
fi
fi
if [ ${recordfail} = 1 ]; then
    set timeout=-1
else
    set timeout=10
fi
### END /etc/grub.d/00_header ###

### BEGIN /etc/grub.d/05_debian_theme ###
set menu_color_normal=white/black
set menu_color_highlight=black/white
### END /etc/grub.d/05_debian_theme ###

### BEGIN /etc/grub.d/10_linux ###
menuentry "Denix, Linux 2.6.31-14-generic" {
    recordfail=1
    if [ -n ${have_grubenv} ]; then save_env recordfail; fi
    set quiet=1
    insmod ext2
    set root=(hd0,1)
    search --no-floppy --fs-uuid --set 34eaa635-ef0e-4d5c-8b61-3c22c767834b
    linux /boot/vmlinuz-2.6.31-14-generic root=UUID=34eaa635-ef0e-4d5c-8b61-
3c22c767834b ro    quiet splash
    initrd /boot/initrd.img-2.6.31-14-generic
}
menuentry "Denix, Linux 2.6.31-14-generic (recovery mode)" {
    recordfail=1
    if [ -n ${have_grubenv} ]; then save_env recordfail; fi

```

```
insmod ext2
set root=(hd0,1)
search --no-floppy --fs-uuid --set 34eaa635-ef0e-4d5c-8b61-3c22c767834b
linux /boot/vmlinuz-2.6.31-14-generic root=UUID=34eaa635-ef0e-4d5c-8b61-
3c22c767834b ro single
initrd /boot/initrd.img-2.6.31-14-generic
}
### END /etc/grub.d/10_linux ###

### BEGIN /etc/grub.d/20_memtest86+ ###
menuentry "Memory test (memtest86+)" {
    linux16 /boot/memtest86+.bin
}
menuentry "Memory test (memtest86+, serial console 115200)" {
    linux16 /boot/memtest86+.bin console=ttyS0,115200n8
}
### END /etc/grub.d/20_memtest86+ ###

### BEGIN /etc/grub.d/30_os-prober ###
if [ ${timeout} != -1 ]; then
    if keystatus; then
        if keystatus --shift; then
            set timeout=-1
        else
            set timeout=0
        fi
    else
        if sleep --interruptible 3 ; then
            set timeout=0
        fi
    fi
fi
### END /etc/grub.d/30_os-prober ###

### BEGIN /etc/grub.d/40_custom ###
# This file provides an easy way to add custom menu entries.  Simply type the
# menu entries you want to add after this comment.  Be careful not to change
# the 'exec tail' line above.
### END /etc/grub.d/40_custom ###
```

Вы наверняка заметили, что синтаксис `grub.cfg` весьма напоминает синтаксис `bash`-сценариев. Параметры GRUB2 задаются в файле `/etc/default/grub`, а в файле `grub.cfg` описываются элементы меню загрузчика.

Рассмотрим описание элемента меню:

```
menuentry "Denix, Linux 2.6.31-14-generic" {
    recordfail=1
    if [ -n ${have_grubenv} ]; then save_env recordfail; fi
    set quiet=1
    insmod ext2
    set root=(hd0,1)
    search --no-floppy --fs-uuid --set 34eaa635-ef0e-4d5c-8b61-3c22c767834b
    linux /boot/vmlinuz-2.6.31-14-generic root=UUID=34eaa635-ef0e-4d5c-8b61-
3c22c767834b ro quiet splash
    initrd /boot/initrd.img-2.6.31-14-generic
}
```

В кавычках после `menuentry` находится описание элемента меню (выделено полужирным) — можете заменить этот текст на все, что вам больше нравится. Далее следуют команды GRUB. Например, команда `insmod ext2` загружает модуль `ext2`. Это не модуль ядра Linux! Это модуль GRUB2 — файл `ext2.mod`, находящийся в каталоге `/boot/grub`.

Команда `set root` устанавливает загрузочное устройство. Формат имени устройства такой же, как в случае с GRUB2.

После служебного слова `linux` задается ядро (файл ядра) и параметры, которые будут переданы ядру. Служебное слово `initrd` указывает файл `initrd`.

Теперь рассмотрим файл `/etc/default/grub`, содержащий параметры GRUB2 (листинг 16.3). Поскольку этот файл вы будете редактировать чаще, чем `grub.cfg`, то комментарии для большего удобства я перевел на русский язык.

Листинг 16.3. Файл `/etc/default/grub`

```
# Если вы измените этот файл, введите команду 'update-grub'
# для обновления вашего файла /boot/grub/grub.cfg.

# Элемент по умолчанию, нумерация начинается с 0
GRUB_DEFAULT=0

# Чтобы увидеть меню GRUB, нужно или закомментировать следующую опцию,
# или установить значение больше 0, но в этом случае
# нужно изменить значение GRUB_HIDDEN_TIMEOUT_QUIET на false
GRUB_HIDDEN_TIMEOUT=0
GRUB_HIDDEN_TIMEOUT_QUIET=true

# Тайм-аут (в секундах)
GRUB_TIMEOUT="10"

# Название дистрибутива — вывод команды lsb_release или просто Debian
GRUB_DISTRIBUTOR=`lsb_release -i -s 2> /dev/null || echo Debian`
```

```
# Параметры ядра по умолчанию
GRUB_CMDLINE_LINUX_DEFAULT="quiet splash"
GRUB_CMDLINE_LINUX=""
# Раскомментируйте для отключения графического терминала
# (только для grub-pc)
#GRUB_TERMINAL=console

# Разрешение графического терминала
#GRUB_GFXMODE=640x480

# Раскомментируйте следующую опцию, если вы не хотите передавать
# параметр "root=UUID=xxx" ядру Linux
#GRUB_DISABLE_LINUX_UUID=true

# Раскомментируйте, если нужно отключить генерацию элемента меню
# режима восстановления
#GRUB_DISABLE_LINUX_RECOVERY="true"
```

После изменения файла `/etc/default/grub` не забудьте запустить команду `update-grub` для обновления вашего `/boot/grub/grub.cfg`.

При редактировании конфигурации GRUB2 нужно придерживаться одной стратегии из двух возможных. Первая заключается в ручном редактировании файла `grub.cfg` — вы редактируете его вручную и больше не используете других программ вроде `grub-mkconfig` или `update-grub`. Вторая стратегия заключается в использовании вспомогательных программ, но тогда не нужно редактировать файл `grub.cfg` вручную, иначе при последующем изменении файла `grub.cfg` программами `grub-mkconfig` или `update-grub` все изменения, внесенные вручную, будут уничтожены.

По умолчанию команда `grub-mkconfig` генерирует конфигурационный файл на консоль, поэтому вызывать ее нужно так:

```
sudo grub-mkconfig > /boot/grub/grub.cfg
```

16.3. Команды установки загрузчиков

Установить GRUB/GRUB2, если вы это еще не сделали, можно командой:

```
/sbin/grub-install <устройство>
```

Например:

```
/sbin/grub-install /dev/sda
```

После изменения конфигурационного файла переустанавливать загрузчик, как в случае с устаревшим LILO, не нужно.

16.4. Установка тайм-аута выбора операционной системы.

Редактирование параметров ядра

По умолчанию GRUB2 не отображает меню выбора операционной системы. Следовательно, вы не можете ни выбрать другую операционную систему (в том числе и Windows), ни изменить параметры ядра Linux, ни выбрать режим восстановления или режим тестирования памяти. Одним словом, такое поведение загрузчика создает определенные неудобства.

Чуть ранее было сказано, что для установки тайм-аута загрузчика нужно отредактировать следующие параметры:

```
GRUB_HIDDEN_TIMEOUT=0
GRUB_HIDDEN_TIMEOUT_QUIET=true
# Тайм-аут (в секундах)
GRUB_TIMEOUT="10"
```

Все правильно, но если бы GRUB2 в Ubuntu был без "глюка". Вообще, "глюки" — это хорошо. Чем корявее будет Canonical делать свои дистрибутивы, тем больше будет работы у авторов книг и дистрибутивов на базе Ubuntu. Вы думаете, почему я создал свой дистрибутив Denix (denix.dkws.org.ua)? Нет, не для того, чтобы гордо ткнуть себя в грудь (мол, я тоже могу сделать свой дистрибутив!). А для того, чтобы после каждой установки Ubuntu пользователи могли не тратить свое личное время часами, настраивая операционную систему.

Например, чтобы побороть такое неадекватное поведение (а каким его еще называть, если программа не реагирует на установку параметров из конфигурационного файла?) загрузчика, мне пришлось потратить минут 15—20. К своему решению я пришел методом эксперимента, поэтому я не удивлюсь, если на каком-то форуме в Интернете вы найдете другое решение (не исключая, может быть даже лучшее).

Итак, откройте ваш файл `/etc/grub.d/30_os-prober`:

```
sudo nano /etc/grub.d/30_os-prober
```

Найдите в нем строку:

```
if [ "x${GRUB_HIDDEN_TIMEOUT}" = "x0" ]
```

Далее все значения `-1` во фрагменте кода, представленном в листинге 16.4, замените на `1`. Строки, которые нуждаются в редактировании, выделены жирным. Изменять значение `-1` в остальном коде, выходящем за рамки листинга 16.4, не нужно!

Листинг 16.4. Фрагмент файла `/etc/grub.d/30_os-prober`

```
if [ "x${GRUB_HIDDEN_TIMEOUT}" = "x0" ] ; then
    cat <
    if [ \${timeout} != 1 ]; then
        if keystatus; then
            if keystatus --shift; then
```

```
        set timeout=1
    else
        set timeout=0
    fi
else
    if sleep$verbose --interruptible 3 ; then
        set timeout=0
    fi
fi
fi
EOF
else
    cat << EOF
    if [ ${timeout} != 1 ]; then
        if sleep$verbose --interruptible ${GRUB_HIDDEN_TIMEOUT} ; then
            set timeout=0
        fi
    fi
fi
EOF
```

После этого сохраните файл и введите команды:

```
sudo grub-mkconfig > /boot/grub/grub.cfg
sudo update-grub
sudo reboot
```

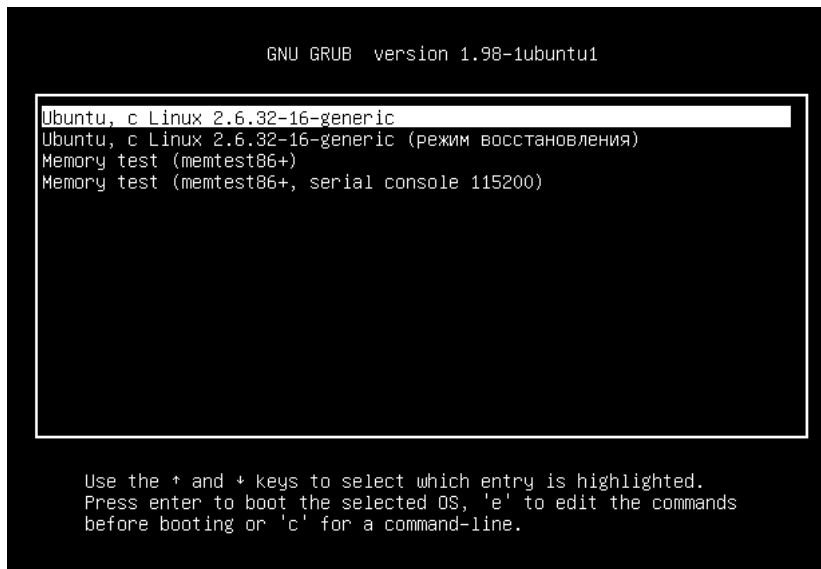


Рис. 16.2. Меню загрузчика

Да, после перезагрузки вы увидите меню GRUB2 (рис. 16.2). Для редактирования параметров ядра (см. приложение 3), которые передаются Linux, выделите загрузочную метку Linux и нажмите клавишу <e>. Если вы защитили GRUB от редактирования параметров ядра, то увидите требование ввести имя пользователя и пароль (рис. 16.3). Если они правильные, вы сможете отредактировать загрузочную метку (рис. 16.4). В данном случае дополнительные параметры нужно вводить после параметра `splash` (строка параметров начинается после служебного слова `linux`). Кстати, если у вас проблемы с запуском Linux, то, чтобы увидеть больше диагностических сообщений, параметры `quiet` и `splash` лучше вообще удалить. Для возврата обратно в меню GRUB2 нажмите клавишу <Esc>, а для загрузки выбранной операционной системы — комбинацию клавиш <Ctrl>+<X>.

```
Enter username:
den
Enter password:
_
```

Рис. 16.3. Ввод имени пользователя и пароля

```
GNU GRUB version 1.98-1ubuntu1

recordfail
insmod ext2
set root='(hd0,1)'
search --no-floppy --fs-uuid --set 4ae4fcbc-2672-400c-9f50-555f496ae\
bd8
linux /boot/vmlinuz-2.6.32-16-generic root=UUID=4ae4fcbc-2672-400c-9\
f50-555f496aebd8 ro quiet splash
initrd /boot/initrd.img-2.6.32-16-generic

Minimum Emacs-like screen editing is supported. TAB lists
completions. Press Ctrl-x to boot, Ctrl-c for a command-line or
ESC to return menu.
```

Рис. 16.4. Редактирование загрузочной метки

16.5. Установка собственного фона загрузчика GRUB и GRUB2

Вы хотите создать собственный фон для загрузчика GRUB? Это очень просто. Создайте или найдите в Интернете понравившуюся вам картинку. Уменьшите ее до размера 640×480 и конвертируйте в формат XPM.

Все это можно сделать одной командой:

```
# convert image.jpg -colors 14 -resize 640x480 image.xpm
```

Затем сожмите картинку с помощью команды `gzip`:

```
# gzip image.xpm
```

Скопируйте сжатую картинку в каталог `/boot/grub` и пропишите в конфигурационном файле `/boot/grub/grub.conf`:

```
splashimage=(hd0,1)/grub/image.xpm.gz
```

Теперь разберемся, как установить графический фон в GRUB2. Убедитесь, что установлен пакет `grub2-splashimages`. Этот пакет содержит графические заставки для GRUB2, которые будут установлены в каталог `/usr/share/images/grub`. Если вам не нравятся стандартные картинки, тогда множество фонов для GRUB2 вы можете скачать с сайта <http://www.gnome-look.org/> или создать вручную, как было показано ранее. Вот только GRUB2 уже поддерживает форматы PNG и TGA, поэтому конвертировать в формат XPM уже не нужно. Будем считать, что картинка у нас уже выбрана. Осталось только установить ее как фон.

Откройте файл темы GRUB2. Он находится в каталоге `/etc/grub.d`. В Ubuntu и Debian он называется `/etc/grub.d/05_debian_theme`. В других дистрибутивах (учитывая, что далеко не все современные дистрибутивы перешли на GRUB2, точное название не могу сказать) он может называться иначе.

Найдите в файле темы следующую строку:

```
for i in {/boot/grub,/usr/share/images/desktop-base}/moreblue-orbit-  
grub.{png,tga} ; do
```

Замените ее следующей строкой:

```
for i in {/boot/grub,/usr/share/images/desktop-  
base,/usr/share/images/grub}/имя_файла.{png,tga} ; do
```

Как видите, мы просто прописали выбранную вами картинку. Далее нужно обновить GRUB2:

```
sudo update-grub
```

16.6. Постоянные имена и GRUB

Как было отмечено ранее, все современные дистрибутивы перешли на так называемые постоянные ("длинные") имена. Раньше, когда еще никто не знал о длинных именах, запись в файле `grub.conf` могла выглядеть так:

```
kernel /boot/vmlinuz26 root=/dev/hda1 vga=0x318 ro
```

Эта запись указывает имя ядра (`/boot/vmlinuz26`). Все, что после него — параметры, которые будут переданы ядру. Один из них (параметр `root`) — указывает имя корневой файловой системы. Здесь оно приведено еще в старом формате. Сейчас вы такие имена в `grub.conf` не увидите (если, конечно, сами не пропишете).

Варианты указания длинных имен выглядят так:

```
root=/dev/disk/by-uuid/2d781b26-0285-421a-b9d0-d4a0d3b55680
root=/dev/disk/by-id/scsi-SATA_WDC_WD1600JB-00_WD-WCANM7959048-part5
root=LABEL=
```

Какой вариант будет использоваться у вас, зависит от дистрибутива. Например, в Fedora применяют третий способ, а в openSUSE — второй.

16.7. Восстановление загрузчика GRUB/GRUB2

Что делать, если вы переустановили Windows, а она установила в MBR свой загрузчик, и теперь вы не можете загрузить Linux? Не переустанавливать же еще и Linux из-за такой мелочи!

Для восстановления загрузчика GRUB нужно загрузиться с LiveCD (подойдет любой LiveCD с любым дистрибутивом Linux) и ввести следующие команды:

```
mkdir /old
mkdir /old/dev
mount /dev/sdaN /old
```

ПРИМЕЧАНИЕ

Все команды нужно вводить от имени root. Для этого следует использовать команды `su` или `sudo`. В частности, в LiveCD Ubuntu нужно вводить все команды с использованием команды `sudo`, например так:

```
sudo mkdir /old
sudo mkdir /old/dev
...
```

Разберемся, что означают эти команды:

- ❖ первая из них создает каталог `/old`, который будет использоваться в качестве точки монтирования;
- ❖ вторая создает в этом каталоге подкаталог `dev`, который пригодится для монтирования `devfs` — псевдофайловой системы;
- ❖ третья используется для монтирования корневой файловой системы дистрибутива Linux, установленного на жестком диске в разделе `/dev/sdaN` (где *N* — номер раздела), к каталогу `/old`. Предположим, что на вашем компьютере дистрибутив Linux был установлен в раздел `/dev/sda5`. Тогда вам нужно ввести следующую команду:

```
mount /dev/sda5 /old
```

После этого нужно подмонтировать каталог `/dev` к каталогу `/old/dev`. Это делается с помощью все той же команды `mount`, но с параметром `--bind`:

```
mount --bind /dev /old/dev
chroot /old
```

Команда `chroot` заменяет корневую систему нашего LiveCD корневой системой дистрибутива, установленного на винчестере. Вам остается лишь ввести команду:

```
/sbin/grub-install /dev/sda
```

Эта команда установит загрузчик GRUB так, как он был установлен до переустановки Windows. После установки GRUB нужно перезагрузить компьютер командой `reboot`.

ПРИМЕЧАНИЕ

Дополнительную информацию о восстановлении загрузчика GRUB вы можете получить на моем форуме: <http://www.dkws.org.ua/phpbb2/viewtopic.php?t=3275>.

16.8. Две и более ОС Linux на одном компьютере

Рассмотрим другую ситуацию, часто возникающую на практике. Вы решили установить на свой компьютер (на котором уже была установлена Windows) операционную систему Linux. Все прошло гладко, и теперь вы с помощью GRUB можете запустить две системы — Windows и Linux. Но потом вы решили установить еще один дистрибутив Linux, однако старый удалять пока не хотите. Поэтому вы создали еще один Linux-раздел и установили в него новый дистрибутив, но после перезагрузки обнаружили небольшую проблему:

- ❖ в меню GRUB отображается только последний установленный дистрибутив и Windows, т. е. вы не можете загрузить первый дистрибутив. Так, Fedora, например, напрочь игнорирует все установленные до нее дистрибутивы, и поэтому после установки этого дистрибутива вы можете запустить только его и Windows;
- ❖ или в меню GRUB отображаются оба дистрибутива и Windows, но запустить вы можете только последний установленный дистрибутив (и, понятно, Windows). Такую картину я наблюдал после установки openSUSE — в моем загрузочном меню появилась метка для загрузки ранее установленного дистрибутива Fedora, но загрузить его не получалось.

Понятно, что восстановить загрузчик первого дистрибутива, воспользовавшись рекомендациями из предыдущего раздела, мы не можем, поскольку после этого мы сможем запустить только первый дистрибутив и Windows (на момент формирования файла `grub.conf` первого дистрибутива еще ничего не было известно о втором дистрибутиве, который вы недавно установили).

Наши действия будут зависеть от конкретной ситуации. Для большей определенности предположим, что первый дистрибутив был установлен в раздел `/dev/sda5`, а второй — в раздел `/dev/sda6`.

Если у вас проблема по первому случаю (когда ранее установленного дистрибутива вообще нет в загрузочном меню), тогда вам нужно примонтировать раздел первого дистрибутива (у нас это `/dev/sda5`) к каталогу `/mnt` (или к любому другому):

```
# mount /dev/sda5 /mnt
```

Затем надо открыть файл `/mnt/boot/grub/grub.conf` (`/mnt/boot/grub/menu.lst`).

ВНИМАНИЕ!

Исходя из приведенного здесь пути к файлу, мы понимаем, что открываем файл `grub.conf` первого дистрибутива.

Скопируйте из него метку загрузки первого дистрибутива. У меня сначала был установлен `openSUSE 11.2`, а потом я установил `Fedora 12`, поэтому загрузочная метка в моем случае выглядела так:

```
title openSUSE 11.2
    root (hd0,4)
    kernel /boot/vmlinuz-2.6.31-14-default root=/dev/disk/by-id/scsi-
SATA_WDC_WD1600JB-00_WD-WCANM7959048-part5 vga=0x317 resume=/dev/sda7
splash=silent showopts
    initrd /boot/initrd-2.6.31-14-default
```

СОВЕТ

Обратите внимание — параметр `root` содержит постоянное (длинное) имя, поэтому его не придется изменять. Если же в вашем варианте параметр `root` содержит короткое имя вида `/dev/sd*`, его желательно заменить постоянным именем.

Скопированную загрузочную метку нужно вставить в файл `/boot/grub/grub.conf` — это файл конфигурации GRUB, используемый в настоящий момент. Файл сохраните, но пока не закрывайте и не перезагружайте компьютер. Обратите внимание — для загрузки нашего первого дистрибутива требуются файлы `vmlinuz-2.6.22.5-31-default` и `initrd-2.6.22.5-31-default`. Их нужно скопировать из каталога `/mnt/boot` в каталог `/boot`:

```
cp /mnt/boot/vmlinuz* /boot
cp /mnt/boot/initrd* /boot
```

Теперь можно перезагрузить компьютер. Первый дистрибутив, установленный в `/dev/sda5`, будет загружен.

Перейдем ко второму случаю. Он проще тем, что нам не нужно редактировать `grub.conf`, поскольку за нас это уже сделала программа установки второго дистрибутива. Вам нужно только подмонтировать каталог `/dev/sda5` к каталогу `/mnt` и скопировать файлы `vmlinuz*` и `initrd*` из каталога `/mnt/boot` в каталог `/boot`. Вот и все.

Напоследок рекомендую прочитать тему форума, непосредственно относящуюся к рассматриваемому вопросу:

<http://www.dkws.org.ua/phpbb2/viewtopic.php?t=3085>.

16.9. Загрузка с ISO-образов

Предположим, вы скачали ISO-образ новой версии Ubuntu, но у вас нет "болванки", чтобы записать на нее образ и загрузиться с полученного диска. Могу вас обрадовать: "болванка" вам для этого не понадобится — GRUB2 умеет использовать ISO-образы в качестве загрузочных устройств. Просто пропишите ISO-образ в конфигурационном файле GRUB2 и перезагрузите компьютер. Новая загрузочная метка появится в меню GRUB2, и если ее выбрать, система загрузится с ISO-образа.

Итак, создайте в каталоге /boot подкаталог iso (название, сами понимаете, может быть любым), загрузите в него ISO-образ дистрибутива. Теперь вам осталось лишь отредактировать конфигурационный файл /boot/grub/grub.cfg, добавив в него вот такую загрузочную запись (выделенный полужирным шрифтом текст нужно записать в одну строку):

```
menuentry "Ubuntu LiveCD" {  
    loopback loop /boot/iso/ubuntu.iso  
    linux (loop)/casper/vmlinuz boot=casper iso-  
scan/filename=/boot/iso/ubuntu.iso noeject noprompt --  
    initrd (loop)/casper/initrd.lz  
}
```

Перезагружаемся и выбираем пункт меню **Ubuntu LiveCD**.

16.10. Установка пароля загрузчика

Теперь самое время защитить ваш загрузчик. По умолчанию любой желающий может изменить параметры ядра. Достаточно злоумышленнику передать параметры `rw, signle` или `rw, init=/bin/bash`, после загрузки он сможет сделать с системой все, что захочет, например изменить пароль `root`. А получив `root`-доступ, можно настроить систему так, как ему это выгодно (или полностью уничтожить ее, хотя это можно было бы сделать и на первом этапе). Поэтому мы должны защитить загрузчик паролем. Загрузка операционных систем будет осуществляться без пароля, однако если кто-то захочет изменить параметры ядра, то у него ничего не получится — GRUB попросит ввести пароль. Для самых "образованных" доброжелателей, которые могут подключить жесткий диск к Windows-системе и с помощью Total Commander просмотреть конфигурационный файл GRUB, мы закодируем наш пароль с помощью алгоритма MD5 — это самый стойкий алгоритм шифрования на сегодняшний день. Поэтому, даже если злоумышленник и просмотрит конфигурационный файл загрузчика, пароль он все равно не узнает.

16.10.1. Загрузчик GRUB

Введите команду `grub`. Появится приглашение:

```
grub>
```

В ответ на приглашение введите команду:

```
md5crypt
```

После этого программа запросит вас ввести пароль, который будет закодирован, и на экране появится шифр введенного пароля:

```
Password: *****
```

Вы получите зашифрованный пароль. Перепишите данный шифр (а еще лучше выделите его и выполните команду меню терминала **Правка | Копировать**).

После этого введите команду:

```
quit
```

На всякий случай сделайте копию конфигурационного файла загрузчика:

```
sudo cp /boot/grub/grub.conf /boot/grub/grub.conf_backup
```

Теперь откройте файл `/boot/grub/grub.conf` в любом текстовом редакторе:

```
gksudo gedit /boot/grub/grub.conf
```

Найдите секцию пароля:

```
## password ['--md5'] passwd
# If used in the first section of a menu file,
# disable all interactive editing
# control (menu entry editor and command-line) and entries protected # by the
command 'lock'
# e.g. password topsecret
#      password --md5 $1$gLhU0/$aW78kHK1QfV3P2b2znUoe/
# password topsecret
```

После нее вставьте строку:

```
password --md5 ваш-шифр
```

После опции `md5` вы должны указать свой шифр, который вы получили в ответ на введенный пароль.

Мы задали пароль, с помощью которого можно редактировать загрузочное меню GRUB. Пока не будет указан заданный пароль, GRUB не разрешит редактировать загрузочное меню.

16.10.2. Загрузчик GRUB2

Как уже отмечалось, начиная с версии 9.10, в Ubuntu используется загрузчик GRUB2 вместо обычного GRUB. По сравнению с GRUB, новый загрузчик одновременно и проще в обращении, и сложнее в настройке. Настраивать GRUB2 при-

дется реже, но к его сложной настройке надо будет привыкать, — практически все современные дистрибутивы перешли на GRUB2.

В GRUB можно было задать общий пароль для всех загрузочных меток, а также установить пароль только на некоторые загрузочные метки. В GRUB2 можно сделать то же самое, но, кроме самого пароля, понадобится указать еще и имя пользователя, что усложняет злоумышленнику взлом системы, поскольку ему нужно будет знать не только пароль, но и имя пользователя. Защита отдельных загрузочных меток, как правило, используется редко, чаще устанавливается пароль на все метки сразу, что и будет продемонстрировано в этой главе.

Сначала установим простой (незашифрованный) пароль, а затем зашифруем его, чтобы никто не смог его прочитать, загрузившись с LiveCD. Прежде всего, откройте файл `/etc/grub.d/00_header`:

```
sudo nano /etc/grub.d/00_header
```

В конец файла добавьте строки:

```
cat << EOF
set superusers="den"
password den 1234
EOF
```

Здесь имя пользователя `den`, пароль — `1234`.

Теперь обновите GRUB2:

```
sudo update-grub
```

Можно также напрямую редактировать файл конфигурации GRUB2 — `grub.cfg`. В него следует добавить вот такие строки:

```
set superusers="user1"
password user1 password1
password user2 password2
```

Обратите внимание, что командами `password` заданы два пользователя: `user1` и `user2` с паролями `password1` и `password2` соответственно. Но пользователь `user1` является суперпользователем, т. е. может редактировать загрузочные метки GRUB2, а обычный пользователь (`user2`) может только загружать метки. Таким образом, у пользователя `user1` получится передать ядру новые параметры, а пользователь `user2` сможет только загрузить Linux с параметрами по умолчанию.

Можно даже задать условие, что метку Windows будет загружать только пользователь `user2`:

```
menuentry "Windows" --users user2 {
    set root=(hd0,2)
    chainloader +1
}
```

Теперь разберемся с шифрованием пароля. Команда `password` поддерживает только незашифрованные пароли. Если вы хотите использовать зашифрованные пароли, то нужно применить команду `password_pbkdf2`. Например:

```
password_pbkdf2 den зашифрованный_пароль
```


Получить зашифрованный пароль можно командой:

```
grub-mkpasswd-pbkdf2
```

После программа запросит у вас пароль и сообщит его хэш:

Your PBKDF2 is grub.pbkdf2.зашифрованный_пароль

Пример пароля:

```
grub.pbkdf2.sha512.10000.9290F727ED06C38BA4549EF7DE25CF5642659211B7FC076F2D28F
EFD71784BB8D8F6FB244A8CC5C06240631B97008565A120764C0EE9C2CB0073994D79080136.88
7CFF169EA8335235D8004242AA7D6187A41E3187DF0CE14E256D85ED97A97357AAA8FF0A3871AB
9EEFF458392F462F495487387F685B7472FC6C29E293F0A0
```

Весь этот хэш нужно скопировать в конфигурационный файл GRUB2:

```
password_pbkdf2 den
grub.pbkdf2.sha512.10000.9290F727ED06C38BA4549EF7DE25CF5642659211B7FC076F2D28F
EFD71784BB8D8F6FB244A8CC5C06240631B97008565A120764C0EE9C2CB0073994D79080136.88
7CFF169EA8335235D8004242AA7D6187A41E3187DF0CE14E256D85ED97A97357AAA8FF0A3871AB
9EEFF458392F462F495487387F685B7472FC6C29E293F0A0
```

Если вы не использовали файл `00_header`, а редактировали непосредственно файл `grub.cfg`, то команду `update-grub` вводить не нужно!

Дополнительную информацию вы сможете получить по адресам:

<http://ubuntuguide.net/how-to-setup-boot-password-for-grub2-entries>;

<http://grub.enbug.org/Authentication>.

ГЛАВА 17



Системы инициализации Linux

17.1. Начальная загрузка Linux

Давайте разберемся, как загружается Linux. В этой книге мы уже упоминали о начальной загрузке компьютера, поэтому сейчас начнем с того момента, когда загрузчик BIOS нашел загрузочное устройство, например, жесткий диск. Далее загрузчик BIOS считывает первый (нулевой) сектор и передает ему управление. На этом работа загрузчика BIOS заканчивается.

В первом секторе находится главная загрузочная запись (Master Boot Record, MBR), состоящая из трех частей: первичного загрузчика, таблицы разделов диска (partition table) и флага загрузки.

Итак, из первой части MBR вызывается первичный загрузчик. Действия этого загрузчика зависят только от него самого. Предположим, что у нас установлен загрузчик LILO — намного проще рассматривать работу загрузчика на конкретном примере.

Загрузчик LILO состоит из двух частей: первая содержится в MBR, а вторая находится на диске в виде файла `/boot/boot.b`. Задача первой части — запуск вторичного загрузчика (второй части), который и производит дальнейшую загрузку системы. Первая часть ничего не знает о файловых системах, поэтому местонахождение второй части записано в "физических координатах", т. е. явно указаны цилиндр, головка, сектор жесткого диска.

Вторая часть загрузчика более интеллектуальна. Она уже "знает", что такое файловая система, а карта размещения файлов записана в файле `/boot/map`. Этот файл используется для поиска ядра и образа виртуального диска. Для чего нужен виртуальный диск? Представим, что мы еще не установили Linux, а только собираемся это сделать. Вставляем загрузочный диск, и загрузчик запускает не просто инсталлятор — на самом деле запускается операционная система Linux, ясно виден процесс загрузки ядра, и потом уже запускается программа установки. Но ядру нужно же откуда-то прочитать модули поддержки устройств и файловой системы — ведь корневая файловая система еще не создана. Вот все эти модули и находятся на виртуальном диске. Виртуальный диск загружается в память, ядро монти-

рует его, как обычную файловую систему, и загружает с него все необходимые модули. После этого виртуальный диск размонтируется и — в случае нормальной загрузки, а не установки Linux, — вместо него монтируется обычная корневая файловая система.

Для работы с виртуальным диском используется технология `initrd` (INITial Ram Disk). Файл образа виртуального диска находится в каталоге `/boot` и называется `initrd-<версия ядра>`.

17.2. Система инициализации `init`

В процессе запуска ядра монтируется корневая файловая система и запускается программа `init`, которая и выполняет дальнейшую инициализацию системы. Программа `init` — часть самой надежной и распространенной системы инициализации Linux, которая используется многими дистрибутивами: Fedora, ASPLinux, Mandriva, openSUSE и др.

Кроме системы инициализации `init`, существуют и другие системы, например, `initng` и `upstart`, которые мы также рассмотрим в этой книге:

- ❖ система `initng` позволяет существенно ускорить запуск Linux, но она почему-то не прижилась в мире Linux, и ни один дистрибутив не использует ее по умолчанию. Очевидно, это из-за сложной настройки данной системы. Прочитать об этой системе можно в моей статье: <http://www.dkws.org.ua/index.php?page=show&file=a/system/initng/initng;>
- ❖ система `upstart` была специально разработана для дистрибутива Ubuntu Linux, но ее при желании можно установить в любом дистрибутиве.

17.2.1. Файл `/etc/inittab`

Итак, программа `init` читает конфигурационный файл `/etc/inittab` и запускает другие процессы, согласно инструкциям этого файла (листинг 17.1).

Листинг 17.1. Файл `/etc/inittab`

```
id:5:initdefault:

# Инициализация системы
si::sysinit:/etc/rc.d/rc.sysinit
10:0:wait:/etc/rc.d/rc 0
11:1:wait:/etc/rc.d/rc 1
12:2:wait:/etc/rc.d/rc 2
13:3:wait:/etc/rc.d/rc 3
```

```
l4:4:wait:/etc/rc.d/rc 4
l5:5:wait:/etc/rc.d/rc 5
l6:6:wait:/etc/rc.d/rc 6

# Что делать при нажатии CTRL-ALT-DELETE
ca::ctrlaltdel:/sbin/shutdown -t3 -r now

# От UPS была получена команда, что пропало питание.
# Немного ждем и выключаем компьютер
pf::powerfail:/sbin/shutdown -f -h +2 "Power Failure; System Shutting Down"

# От UPS получена команда, что питание возобновилось.
# Отменяем shutdown
pr:12345:powerokwait:/sbin/shutdown -c "Power Restored; Shutdown Cancelled"

# Запуск gettys
1:2345:respawn:/sbin/mingetty tty1
2:2345:respawn:/sbin/mingetty tty2
3:2345:respawn:/sbin/mingetty tty3
4:2345:respawn:/sbin/mingetty tty4
5:2345:respawn:/sbin/mingetty tty5
6:2345:respawn:/sbin/mingetty tty6

# Однопользовательский режим
~~:S:wait:/bin/sh
```

Одна из главных инструкций файла `/etc/inittab` выглядит так:

```
id:<число>:initdefault:
```

Эта инструкция задает уровень запуска по умолчанию. Уровень запуска определяет, какие действия будут выполнены программой `init` (какие процессы будут запущены). Всего предусмотрено шесть уровней запуска:

- ❖ 0 — останов системы (ясно, что в качестве уровня по умолчанию этот уровень быть не может);
- ❖ 1 — однопользовательский режим (в него можно перейти сразу при загрузке, передав ядру параметр `single`);
- ❖ 2 — многопользовательский режим без поддержки сети;
- ❖ 3 — многопользовательский режим с поддержкой сети;
- ❖ 4 — не используется;
- ❖ 5 — многопользовательский графический режим с загрузкой X11 и поддержкой сети;
- ❖ 6 — перезагрузка системы.

В большинстве случаев в качестве уровня запуска по умолчанию устанавливается 3 или 5.

17.2.2. Команда *init*

Перейти на тот или иной уровень можно и после загрузки системы. Для этого используется команда:

```
# /sbin/init <уровень_запуска>
```

ПРИМЕЧАНИЕ

Напомню, что решетка (#) перед командой означает, что команда должна быть выполнена от имени пользователя root.

"Вычислив" уровень запуска, *init* поочередно запускает сценарии из каталога */etc/rc.d/rcX.d*, где *X* — это номер уровня запуска. Если зайти в один из этих каталогов, например, в */etc/rc.d/rc3.d*, то можно увидеть ссылки формата:

S<номер><имя>

Параметр *<номер>* определяет порядок запуска сценария (например, *S10network* запустится раньше, чем *S11internet*), а параметр *<имя>* задает имя сценария. Сами сценарии находятся в каталоге */etc/rc.d/init.d*.

Ссылки, начинающиеся с символа *S*, — это ссылки запуска (от англ. *start*), при запуске соответствующих сценариев им будет передан аргумент *start*. Например, если программа *init* обнаружила в */etc/rc.d/rc3.d* файл *S10network*, то она выполнит команду:

```
/etc/rc.d/init.d/network start
```

Если имя ссылки начинается на букву *K* (от англ. *kill*), то это ссылка останова сервиса, например, *K01service*. Данная ссылка указывает на команду:

```
/etc/rc.d/init.d/service stop
```

Вы можете запустить любой сценарий из каталога *init.d* непосредственно, передав ему параметры *start* (запуск), *stop* (останов) и другие (зависит от сервиса).

17.2.3. Команда *service*

А можете воспользоваться командой *service*:

```
# service <имя_сервиса> <start|stop|...>
```

Здесь *<имя_сервиса>* — это имя файла в каталоге */etc/rc.d/init.d*.

17.2.4. Редакторы уровней запуска

Редактировать уровни запуска можно вручную или с помощью программ-конфигураторов. В Fedora (и ASPLinux) для редактирования уровней запуска ис-

пользуется конфигуратор `system-config-services` (рис. 17.1), а в Mandrake (Mandriva) можно воспользоваться командой `ntsysv` с параметром `--level <номер сервиса>` (рис. 17.2).

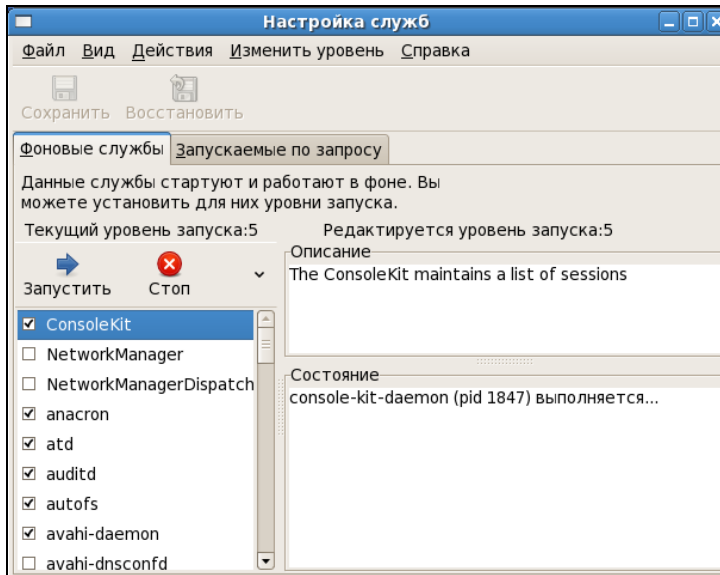


Рис. 17.1. Конфигуратор `system-config-services`



Рис. 17.2. Конфигуратор `ntsysv`

Конфигуратор в Fedora более удобен. Выбрать редактируемый уровень запуска можно с помощью меню **Изменить уровень**.

Конфигуратор drakboot (рис. 17.3), имеющийся в Linux Mandrake (Mandriva), позволяет указать, в каком режиме будет запускаться система — в графическом или в режиме консоли. По сути, конфигуратор позволяет выбрать уровень запуска (3 — консоль, 5 — графический режим).

В случае если система будет запускаться в графическом режиме, данный конфигуратор позволяет включить автовход. Для функции автовхода нужно указать два параметра — имя пользователя и графическую среду.

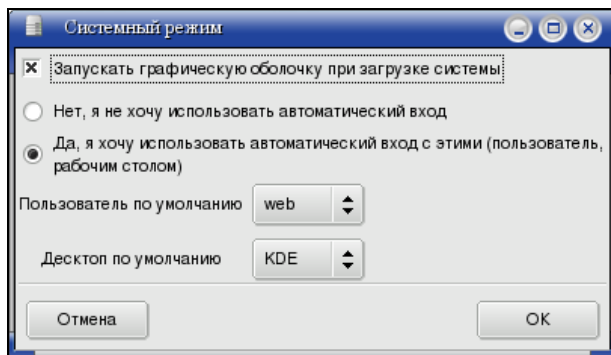


Рис. 17.3. Конфигуратор drakboot

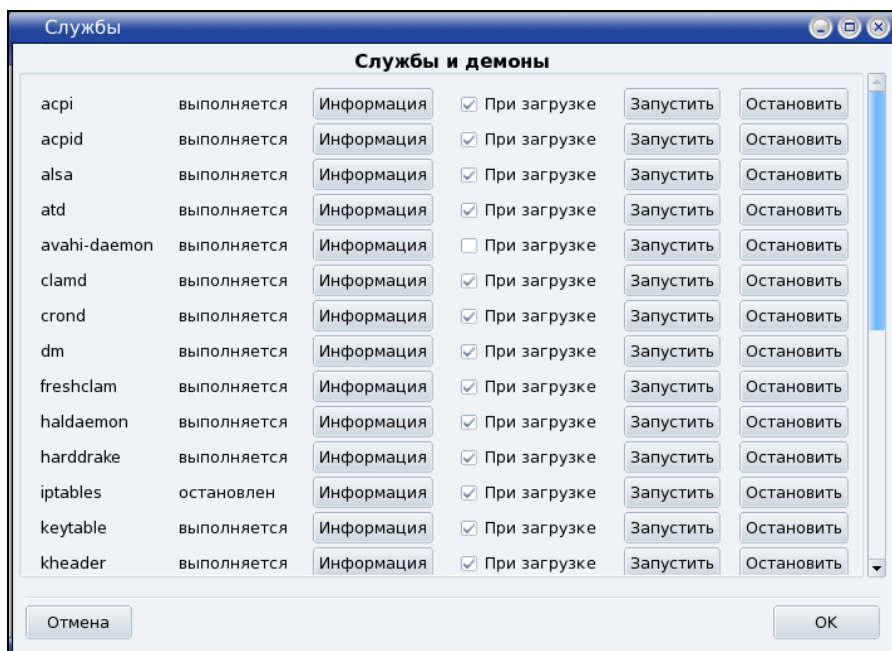


Рис. 17.4. Конфигуратор drakxservices

Если автовход выключен, то при запуске XOrg система запросит у вас имя пользователя и пароль. Также у вас будет возможность выбрать графическую среду, с которой вы хотите работать. Если автовход включен, то будет выполнена автоматическая регистрация в системе выбранного пользователя с запуском выбранной графической среды. После этого вы можете работать в системе от имени этого пользователя. Из соображений безопасности конфигуратор не позволяет выбрать пользователя root.

Еще в Mandriva есть конфигуратор drakxservices (рис. 17.4), позволяющий редактировать сервисы, запускаемые на пятом уровне запуска (обычно этот уровень запуска и используется).

17.3. Система инициализации upstart

Система инициализации upstart была разработана Скотом Джеймсом Ремнантом (Scott James Remnant) для дистрибутива Ubuntu, однако upstart, если она вам понравилась, можно с успехом использовать в других дистрибутивах. Мы не будем рассматривать установку upstart на другой дистрибутив, а разберемся, как с ней работать в Ubuntu.

17.3.1. Как работает upstart

Upstart заменяет инициализирующие сценарии для поддержки событийно-ориентированного режима действий. Проще говоря, в upstart есть собственный процесс init, который запускается при запуске системы (аналогично программам init и initng). При запуске генерируется событие startup, при завершении работы — shutdown, при нажатии клавиатурной комбинации <Ctrl>+<Alt>+ — событие ctrl-alt-delete.

Вы можете создавать собственные события. Вот небольшой пример создания события my_event:

```
on my_event
exec echo event received
console output
```

При получении этого события на консоль будет выведено сообщение:

```
event received
```

Файлы событий хранятся в каталоге /etc/event.d. Создайте в этом каталоге файл с именем my_event и поместите в него приведенный код. После этого вызвать событие вы можете командой:

```
initctl emit my_event
```

Подробнее об этой команде вы сможете прочитать на странице руководства (в Ubuntu оно на русском языке): `man initctl`.

17.3.2. Конфигурационные файлы upstart

Исследуйте содержимое каталога `/etc/event.d`. В нем вы найдете файлы событий перехода на определенный запуск. В листинге 17.2 представлен файл события перехода на пятый уровень запуска — `/etc/event.d/rc5`.

Листинг 17.2. Файл события `/etc/event.d/rc5`

```
start on runlevel 5

stop on runlevel [!5]

console output
script
    set $(runlevel --set 5 || true)
    if [ "$1" != "unknown" ]; then
        PREVLEVEL=$1
        RUNLEVEL=$2
        export PREVLEVEL RUNLEVEL
    fi

    exec /etc/init.d/rc 5
end script
```

Не нужно быть гуру в программировании, чтобы понять, что делает этот сценарий — он выполняет сценарий `/etc/init.d/rc`, передав ему значение 5 — номер уровня запуска. Сценарий `/etc/init.d/rc` занимается запуском/остановкой служб на определенном уровне, который ему передается в качестве параметра.

Но самое интересное в `upstart`, что уровни запуска здесь — виртуальные. На самом деле, номера уровней запуска остались только ради совместимости с `init`, чтобы человеку, который впервые увидел `upstart` (точнее, дистрибутив с установленной системой инициализации `upstart`), было проще с ней разобраться. В `upstart`, благодаря событийно-ориентированному режиму, вообще отпадает необходимость в уровнях запуска, подобных тем, которые использовались в `init`. Загрузка того или иного сервиса происходит при наличии нужного аппаратного обеспечения: нет устройства — не будет загружен и сервис, требующий его.

Кстати, в последних версиях `Ubuntu` имеется команда `service`, что очень удобно, особенно, если вы до этого привыкли к `init`. Формат вызова команды `service` такой же.

`Upstart` можно использовать в режиме "горячей замены" — если вы в процессе работы системы подключите какое-то устройство, например, PCMCIA-карту или USB-устройство, будет сгенерировано соответствующее событие. После этого будут запущены все необходимые для обеспечения работы этого устройства процес-

сы. Так, при подключении сетевой карты PCMCIA будет сгенерировано событие `network-interface-added`, которое запустит процесс настройки сетевой карты по DHCP, при этом будет сгенерировано новое событие — `network-interface-up` и т. д. Понятно, что если нет сетевых устройств, то и соответствующие им события не будут генерироваться.

17.4. Система инициализации Slackware

Система инициализации Slackware отличается от привычной системы `init`, используемой в SysV-системах. Она больше похожа на систему инициализации BSD-систем, хотя некоторые сходства с SysV все же есть.

ПОЯСНЕНИЕ

Если вы совсем незнакомы с историей UNIX, то вам неизвестны и термины "SysV" (System V) и "BSD". Считается, что UNIX "родилась" в 1969 году. В то время над проектом работали сотрудники компании Bell Labs (это подразделение AT&T) Руд Кенедей (Rudd Canaday), Дуг Макилрой (Doug McIlroy), Деннис Ритчи (Dennis Ritchie) и Кен Томпсон (Ken Thompson). Позже UNIX заинтересовались другие организации, в частности, институт Беркли (Калифорния, США). В 1975 году появилась слегка модифицированная версия UNIX от института Беркли, которая получила название BSD (Berkeley Software Distribution), а версия от AT&T (Bell Labs) стала называться System V (SysV). Обе системы были очень похожи друг на друга, но в то же время имели свои особенности. Например, BSD имела собственную систему инициализации, которая очень напоминает ту, что сейчас используется в Slackware Linux.

Если говорить о сходстве систем инициализации в стилях SysV и BSD, то у обеих систем присутствуют уровни запуска, имеется файл `/etc/inittab` — таблица инициализации (см. *ранее*). Однако имена файлов системы инициализации BSD-стиля немного отличаются от имен файлов SysV-стиля.

Система инициализации Slackware построена таким образом, что вне зависимости от уровня запуска первым всегда запускается сценарий `/etc/rc.d/rc.S`. Он монтирует псевдофайловые системы `/proc`, `sysfs` и `devfs`, запускает систему `hotplug` (драйвер устройств, обеспечивающий их "горячее" подключение, т. е. подключение без выключения компьютера, например, USB-устройств), подключает разделы свопинга, монтирует и проверяет корневую файловую систему, монтирует другие файловые системы и т. д. Как видите, сценарий `/etc/rc.d/rc.S` выполняет большую часть действий по инициализации системы. Обычно данный файл не требует изменения. Но иногда его приходится редактировать. Например, если вы создали файл подкачки и хотите, чтобы он подключался при загрузке системы, то команду `swapon <имя_файла>` нужно добавить в файл `/etc/rc.d/rc.S` после команды `/sbin/swapon -a`.

Сценарий `/etc/rc.d/rc.S` проверяет наличие файла `/etc/rc.d/rc.modules.local`, обеспечивающего загрузку модулей при старте системы. При условии, что файл `rc.modules.local` существует, он запускается. В противном случае происходит поиск файла `/etc/rc.d/rc.modules<-версия_ядра>`, а если и его нет, тогда сценарий

/etc/rc.d/rc.S пытается запустить файл /etc/rc.d/rc.modules. Один из этих файлов должен существовать, иначе система будет загружена без модулей, а это означает, что не будут работать некоторые устройства и поддерживаться некоторые файловые системы.

Кроме файла /etc/rc.d/rc.modules.local (или другого файла загрузки модулей, см. *ранее*), также используется файл /etc/rc.d/rc.netdevice. Он служит для загрузки модулей сетевых карт (точнее, сетевых интерфейсов).

Как уже было отмечено, файл /etc/rc.d/rc.S запускается вне зависимости от уровня запуска. Кроме этого файла в каталоге etc/rc.d вы найдете серию файлов rc.N, где N — номер уровня запуска. Данные файлы запускаются в зависимости от выбранного уровня запуска — например, на третьем уровне запуска будет запущен файл /etc/rc.d/rc.3. Каждый такой файл подготавливает систему к работе на выбранном уровне запуска. Уровень запуска по умолчанию, как и в случае с системой инициализации в стиле SysV, задается в файле /etc/inittab.

Сценарий /etc/rc.d/rc.inet1 отвечает за инициализацию сетевых интерфейсов и построение таблицы маршрутизации. Конфигурация сетевых интерфейсов хранится в файле /etc/rc.d/rc.inet1.conf. Вот фрагмент этого файла:

```
IPADDR[0]="192.168.1.1"
NETMASK[0]="255.255.255.0"
USE_DHCP[0]=""
DHCP_HOSTNAME[0]=""
```

Сценарий /etc/rc.d/r.inet2 управляет запуском сетевых служб и подключением сетевых файловых систем. Именно в этом файле происходит попытка монтирования файловых систем NFS и smbfs. Также из этого файла происходит запуск сетевых служб. Сценарии для запуска сетевых служб называются /etc/rc.d/rc.<название службы>, например, /etc/rc.d/rc.sshd — сценарий запуска SSH-сервера. Однако некоторые сетевые сервисы, например, sendmail и samba, в силу своих особенностей запускаются из файлов rc.N.

Иногда нужно обеспечить запуск сетевой службы, для которой нет собственного rc-файла. Тогда ее запуск можно или описать в файле /etc/rc.d/rc.local (что довольно просто), или создать собственный rc-файл и добавить его вызов в один из файлов rc.N. Шаблон собственного rc-файла приведен в листинге 17.3.

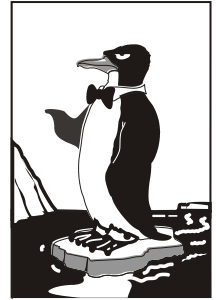
Листинг 17.3. Шаблон rc-файла для запуска сетевой службы

```
#!/bin/bash
start()
{
    echo "Service started"
    service_start
}
stop()
{
```

```
    echo "Service stoped"
    killall service
}

case $1 in
    start)
        start ;;
    stop)
        stop ;;
    restart)
        stop
        sleep 2
        start ;;
    *)
        echo "Usage: service start|stop|restart"
esac
```

ГЛАВА 18



Процессы. Управление процессами. Сервисы

18.1. Управление процессами

Каждому процессу в Linux присваивается уникальный номер — идентификатор процесса (PID, Process ID). Зная ID процесса, вы можете управлять процессом, а именно: можете завершить процесс или изменить приоритет процесса. Принудительное завершение процесса необходимо, если процесс "завис", и его нельзя завершить обычным образом. А изменение приоритета может понадобиться, если вы хотите, чтобы процесс доделал свою работу быстрее.

Предположим, что у вас зависла какая-то программа, например пусть это будет командер `tc`. Хотя и маловероятно (не помню, чтобы `tc` зависал), но, для примера, пусть будет так. Принудительно завершить ("убить") процесс можно с помощью команды `kill`. Формат ее вызова следующий:

```
kill [параметры] PID
```

PID (Process ID) — это идентификатор процесса, который присваивается процессу системой и уникален для каждого процесса. Но мы знаем только имя процесса (имя команды), мы не знаем идентификатор процесса. Узнать идентификатор процесса позволяет программа `ps`. Предположим, что `tc` находится на первой консоли. Поскольку он завис, вы не можете более использовать консоль, и вам нужно переключиться на вторую консоль (`<Alt>++<F2>`). Зарегистрировавшись на второй консоли, введите команду `ps`. Она выведет список процессов, запущенных на второй консоли — это будет `bash` и сам `ps` (рис. 18.1).

Чтобы "добраться" до нужного нам процесса (`tc`), который запущен на первой консоли, введите команду `ps -a` или `ps -U root`. В первом случае вы получите список процессов, запущенных вами, а во втором — список процессов, запущенных от вашего имени (я предполагаю, что вы работаете под именем `root`). Обратите внимание — вы запустили процессы `tc` и `ps` (рис. 18.2), а от вашего имени (`root`) система запустила множество процессов. Обратите внимание: программа `ps` выводит также имя терминала (`tty1`), на котором запущен процесс. Это очень важно, если на

разных консолях у вас запущены одинаковые процессы — ведь можно легко ошибиться и завершить не тот процесс (см. рис. 18.2).

```
Mandriva Linux release 2006.0 (Official) for i586
Kernel 2.6.12-12mdkmp on an i686 / tty2
host login: root
Password:
Last login: Fri Aug  4 01:29:58 on tty1
[root@host ~]# ps
  PID TTY          TIME CMD
 2440 tty2      00:00:00 bash
 2521 tty2      00:00:00 ps
[root@host ~]# _
```

Рис. 18.1. Список процессов на текущей консоли

```
Mandriva Linux release 2006.0 (Official) for i586
Kernel 2.6.12-12mdkmp on an i686 / tty2
host login: root
Password:
Last login: Fri Aug  4 01:29:58 on tty1
[root@host ~]# ps
  PID TTY          TIME CMD
 2440 tty2      00:00:00 bash
 2521 tty2      00:00:00 ps
[root@host ~]# ps -a
  PID TTY          TIME CMD
 2484 tty1      00:00:00 mc
 2581 tty2      00:00:00 ps
[root@host ~]# _
```

Рис. 18.2. Определение PID программы mc

Теперь, когда мы знаем PID нашего процесса, мы можем его "убить":

```
# kill 2484
```

Перейдите на первую консоль после выполнения этой команды — mc на ней уже не будет. Если выполнить команду `ps -a`, то в списке процессов mc тоже не будет.

Вообще-то все эти действия, связанные с вычислением PID процесса, мы рассмотрели только для того, чтобы познакомиться с командой `ps`. Если вы знаете только имя процесса, то гораздо удобнее использовать команду:

```
# killall <имя процесса>
```

Но имейте в виду, что данная команда завершит все экземпляры данного процесса. А вполне может быть, что у нас на одной консоли находится mc, который нужно "убить", а на другой — нормально работающий mc. Команда `killall` "убьет" оба процесса.

При выполнении команд `kill` и `killall` нужно помнить, что они могут убить только те процессы, которые вам принадлежат, если вы работаете от имени обыч-

ного пользователя. Если вы работаете от имени пользователя root, то можете завершить любой процесс в системе.

Иногда бывает, что система ужасно тормозит. Весь день работала нормально и вдруг начала тормозить. Если вы даже не догадываетесь, из-за чего это случилось, то вам нужно использовать программу `top` (рис. 18.3). Она выводит список процессов с сортировкой по процессорному времени. То есть на вершине списка будет процесс, который занимает больше процессорного времени, чем сама система. Вероятно, из-за него и происходит эффект "торможения". На рис. 18.3 показано, что больше всего процессорного времени (0.3%) занимает программа `top`. Конечно, в реальных условиях все будет иначе. Выйти из программы `top` можно, нажав клавишу `<Q>`.

Программа `top` еще полезна выводимой информацией о системе. Так, в первой строчке выводится время, которое работает система с момента загрузки (`up`), количество пользователей (`users`), общая нагрузка системы (`load average`) — за одну минуту, пять минут и пятнадцать минут соответственно (три числа после строки `load average`).

Во второй строке выводится информация о процессах: общее количество процессов (`total`), количество запущенных (`running`), "спящих" (`sleeping`), остановленных (`stopped`) процессов и процессов-зомби (`zombie`). Зомби — это уже "мертвый" (завершенный процесс), но информация о нем еще не удалена из таблицы процессов. Если вы обладаете минимальными навыками программирования на C, вам будет интересна следующая статья:

<http://www.dkws.org.ua/index.php?page=show&file=a/dev/process2>

В ней я показываю, как можно создать зомби. Заодно поймете, как так получается, что система не успевает удалять информацию о процессе из служебной таблицы.

Третья строка выводит информацию о распределении процессорного времени. Первые два числа в ней отражают работу CPU по обработке процессов. Если первые два числа у вас стабильно высокие (99—100%), какой-то процесс (или группа процессов) очень сильно нагружают CPU или же CPU очень слабый.

Остальные параметры в строке CPU не очень важны, хотя на параметр `wa` тоже следует обратить внимание. Он сообщает нам о простое во время ввода/вывода. Если он содержит постоянно больше 80%, то это говорит о том, что процессор проводит много времени в ожидании ввода/вывода. Косвенно это может говорить о проблемах с жестким диском: возможно, он скоро выйдет из строя. Если же с жестким диском все нормально, а процессор далеко не слабый, тогда проблему нужно искать на программном уровне. Скорее всего, причина кроется в некоторых процессах. Для начала нужно определить эти процессы, а затем разбираться, что с ними делать. Определить процессы, потребляющие много оперативной памяти и процессорного времени, поможет команда `ps axfu`. Команда помимо всего выводит состояние процесса:

- ◆ R — запущен;
- ◆ D — ожидание (например, ввода или вывода);
- ◆ S — процесс спит.

```
top - 01:39:31 up 10 min, 3 users, load average: 0.00, 0.00, 0.00
Tasks: 58 total, 1 running, 57 sleeping, 0 stopped, 0 zombie
Cpu(s): 0.0% us, 0.3% sy, 0.0% ni, 99.7% id, 0.0% wa, 0.0% hi, 0.0% si
Mem: 189720k total, 68224k used, 121496k free, 5088k buffers
Swap: 128984k total, 0k used, 128984k free, 38072k cached
```

PID	USER	PR	NI	UIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
2599	root	16	0	1996	1012	804	R	0.3	0.5	0:00.06	top
1	root	16	0	1564	540	472	S	0.0	0.3	0:00.55	init
2	root	RT	0	0	0	0	S	0.0	0.0	0:00.00	migration/0
3	root	34	19	0	0	0	S	0.0	0.0	0:00.00	ksoftirqd/0
4	root	10	-5	0	0	0	S	0.0	0.0	0:00.02	events/0
5	root	16	-5	0	0	0	S	0.0	0.0	0:00.08	khelper
6	root	11	-5	0	0	0	S	0.0	0.0	0:00.00	kthread
8	root	20	-5	0	0	0	S	0.0	0.0	0:00.00	kacpid
61	root	10	-5	0	0	0	S	0.0	0.0	0:00.03	kblockd/0
93	root	20	0	0	0	0	S	0.0	0.0	0:00.00	pdflush
94	root	15	0	0	0	0	S	0.0	0.0	0:00.05	pdflush
96	root	16	-5	0	0	0	S	0.0	0.0	0:00.00	aio/0
95	root	25	0	0	0	0	S	0.0	0.0	0:00.00	kswapd0
684	root	16	0	0	0	0	S	0.0	0.0	0:00.00	kseriod
766	root	13	-5	0	0	0	S	0.0	0.0	0:00.00	ata/0
775	root	18	0	0	0	0	S	0.0	0.0	0:00.00	scsi_eh_0
784	root	16	0	0	0	0	S	0.0	0.0	0:00.02	kjournald
924	root	15	-4	1564	496	420	S	0.0	0.3	0:00.08	udevd

Рис. 18.3. Программа top

Две последние строки выводят информацию об использовании оперативной памяти (memory) и подкачки (swap): всего (total), использовано (used), свободно (free), буферизировано (buffres), прокэшировано (cached).

Предположим, что вы работаете с видео, и вам нужно перекодировать файл из одного видеформата в другой. Конвертирование видео занимает много процессорного времени, а хотелось бы все сделать как можно быстрее и уйти раньше домой. Тогда вам поможет программа nice — она позволяет запустить любую программу с указанным приоритетом. Ясно: чем выше приоритет, тем быстрее будет выполняться программа. Формат вызова команды следующий:

```
nice -n <приоритет> команда аргументы
```

Максимальный приоритет задается числом -20 , а минимальный — числом 19 . Приоритет по умолчанию равен 10 .

18.2. Управление сервисами

Сервис (служба, демон) — специальная программа, выполняемая в фоновом режиме и выполняющая определенные действия. Например, демон печати ждет, пока одно из приложений отправит на печать документ. После этого демон активизируется и выполняет определенные действия, а именно — печать самого документа на выбранном принтере.

Для управления сервисами применяется команда `service`. Использовать ее нужно так:

```
# service сервис параметр-команда
```


Обычно для запуска, останова и перезапуска сервиса используются команды `start`, `stop` и `restart` соответственно. Вот примеры запуска, перезапуска и останова сервиса `proftpd` (FTP-сервер):

```
# service proftpd start
# service proftpd restart
# service proftpd stop
```

В ранних версиях Ubuntu нет команды `service`, поэтому ее можно создать вручную. (В последних версиях эта команда есть!) Введите следующие команды:

```
sudo touch /bin/service
sudo chmod +x /bin/service
gksudo gedit /bin/service
```

Введите следующие строки:

```
$!/bin/bash
/etc/init.d/$1 $2
```

Сохраните файл. Теперь у вас тоже есть команда `service`!

18.3. Отключение неиспользуемых сервисов

После установки Linux по умолчанию включены все сервисы, которые только могут быть включены. Спрашивается, зачем вам демон печати, если у вас нет принтера? А зачем вам целых три планировщика заданий, если вы еще ни одним не умеете пользоваться?

Вы можете сказать: "У меня очень мощный компьютер, на его производительности это никак не отразится". Вы заблуждаетесь. Отключить ненужные сервисы нужно по трем причинам.

- ❖ Увеличивается время загрузки — понятно, что чем меньше сервисов запускается при запуске системы, тем быстрее загружается система.
- ❖ Увеличивается загрузка процессора и увеличивается использование оперативной памяти — если бы основная нагрузка приходилась только на загрузку! Но ведь сервисы находятся в памяти до тех пор, пока система не завершит работу. Следовательно, они занимают процессорное время и оперативную память системы.
- ❖ Каждый сервис нужно расценивать как потенциальную дыру в системе безопасности: запущенный сервис не настроен — он работает с настройками по умолчанию. Ясно, что настройки по умолчанию предназначены только для того, чтобы показать, что сервис запускается, а не для реальных условий. Как говорится, "стандартные средства стандартно и взламываются". Одно дело, если ваш компьютер не подключен к локальной сети или к Интернету — взламывать будет некому. А вот если вы подключены к Интернету, то любой

желающий может попытаться взломать вашу систему. Что после? Узнаете — это зависит от подлости крекера.

Учитывая все сказанное, выведем основные правила:

- ◆ если сервис не нужен — выключите;
- ◆ если вам нужен сервис — настройте его и используйте;
- ◆ если сервис нужен, но нет времени его настроить (или вы не знаете, как это сделать), выключите его — включить всегда успеете.

В Fedora (и ASPLinux) для настройки сервисов используется конфигуратор `system-config-services`, а в Mandrake — `drakxservices`. С этими конфигураторами мы знакомы из предыдущей главы.

Нужно отметить одну особенность конфигулятора `system-config-services`: для сохранения изменений вам нужно нажать кнопку **Сохранить**, в то время как конфигулятор `drakxservices` сохраняет изменения автоматически.

Вот некоторые ненужные сервисы, которые можно отключить:

- ◆ `abrt` — помогает определить баги в различных программах и отправить информацию о них разработчикам;
- ◆ `acpid` — управляет ACPI-событиями, по большому счету, он просто не нужен;
- ◆ `arpm` — 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. На сервере планировщик нужен, но вам нужно решить, какой именно: использовать все три — это явный перебор;
- ◆ `auditd` — демон аудита, на сервере нужен, на рабочей станции — нет;
- ◆ `Bluetooth` — если вы не собираетесь использовать технологию Bluetooth, данный сервис можете просто выключить;
- ◆ `cruspeed`, `haldaemon` — не вдаваясь в подробности, просто отключите эти сервисы;
- ◆ `cups*` — система печати CUPS (Common UNIX Printing System). Нужна только, если есть принтер;
- ◆ `dm` — диспетчер дисплея (`display manager`). Нужен, если вы планируете работать в графическом режиме;
- ◆ `firstboot` — данный сервис есть в Fedora, он проверяет, первая ли это загрузка. В общем, как вы догадались, после первой же загрузки его можно смело отключать;
- ◆ `hidd` — Human Interface Device Daemon, управляет устройствами ввода (клавиатуры, мыши), подключаемыми по Bluetooth. Если у вас таких нет, можете смело отключить;
- ◆ `isdn` — сервис поддержки ISDN-линий. Если у вас нет ISDN, выключите этот сервис;
- ◆ `irqbalance` — нужен только на SMP-машинах (многопроцессорных машинах);
- ◆ `kheader` — выполняет автоматическую генерацию заголовков ядра в `/boot`. Не отключайте этот сервис;

- ❖ 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;
- ❖ mandi — демон мониторинга сети. В большинстве случаев не нужен;
- ❖ mdadm — демон мониторинга и управления программными RAID-массивами;
- ❖ 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);
- ❖ netplugd — демон управления нестатическими сетевыми интерфейсами. Можно с чистой совестью отключить;
- ❖ pcmcia — нужен для поддержки PCMCIA-карт. Если у вас не ноутбук, можете смело выключить этот сервис;
- ❖ pcscd — обслуживает PC/CS-карты, если у вас не ноутбук с поддержкой таких карт, то сервис можно смело выключить;
- ❖ 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;
- ❖ sendmail (или postfix) — используется для организации собственного SMTP-сервера, т. е. 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;
- ❖ shorewall (в других дистрибутивах iptables) — пакетный фильтр (брандмауэр). Если вы еще не настраивали пакетный фильтр, то лучше пока его отключить;
- ❖ smartd — нужен для поддержки S.M.A.R.T.-устройств. Если у вас нет таких устройств, можете выключить этот сервис;
- ❖ sshd — используется для безопасного удаленного доступа к консоли системы. В большинстве случаев не нужен;
- ❖ rhnsd — можно смело выключить.

Нужно отметить, что в вашем дистрибутиве может не быть некоторых указанных сервисов или, наоборот, будут сервисы, не представленные в списке. Набор сервисов зависит от дистрибутива и установленного программного обеспечения.

Чтобы почувствовать, насколько сократилось время загрузки системы, ее нужно перезагрузить:

```
# reboot
```



ЧАСТЬ V

КОМАНДНАЯ СТРОКА

ГЛАВА 19



Консоль Linux

19.1. Что такое консоль

Настоящий линуксоид должен уметь работать в консоли. Ведь когда только появился Linux, была одна консоль, о графическом интерфейсе не было и речи. Знаете, почему UNIX и Linux отталкивали обычных пользователей? Потому что не было хорошего графического интерфейса. Раньше в Linux работали одни профессионалы. Сейчас все изменилось: в Linux очень удобный графический интерфейс, который с удовольствием используют и профессионалы (дождались наконец-то!), забывая о командной строке.

Обычные пользователи туда ни ногой — даже принципиально: мол, зачем в DOS возвращаться? Под "DOS" имелась в виду командная строка Linux. Да, ее вид не очень дружелюбный, но это только кажется. Стоит вам поработать в консоли, и вы поймете все ее прелести. Начнем с того, что командная строка Linux намного удобнее командной строки DOS — об этом мы еще поговорим. В консоли можно выполнять те же операции, что и в графическом режиме, причем все намного быстрее. Хотите бороздить просторы Интернета? Пожалуйста, но без картинок. Не так красиво, но зато сэкономите трафик. А на обмен электронными сообщениями это никак не влияет. В консоли также можно работать и с документами, правда, тоже о графике можно забыть. На старых компьютерах консоль позволяет эффективно использовать ресурсы компьютера. Да, в графическом режиме на стареньком Pentium не поработаешь, зато в текстовом режиме его можно быстро превратить в очень полезный для всей сети компьютер — в шлюз, через который его более мощные собратья будут получать доступ к Интернету.

Как уже было отмечено ранее, для переключения между консолями используются комбинации клавиш `<Alt>+<Fn>` (n от 1 до 6). Для переключения в графический режим служит комбинация клавиш `<Alt>+<F7>`. Чтобы вернуться из графического режима в консоль, предназначена комбинация клавиш `<Ctrl>+<Alt>+<Fn>`, где n — номер нужной вам консоли.

19.2. Правильная работа в консоли

Работа в консоли заключается во вводе нужной команды. Вы вводите команду (например, команду создания каталога, просмотра файла, вызова редактора и т. д.) и нажимаете клавишу <Enter>. Команда содержит как минимум имя запускаемой программы. Кроме имени программы, команда может содержать параметры, которые будут переданы программе, а также символы перенаправления ввода/вывода (об этом чуть позже). Естественно, вам нужно знать имя программы, а также параметры, которые нужно ей передать.

Если вы помните название программы, а назначение параметров забыли, вспомнить поможет команда `man`. `Man` — это справочная система Linux. В ней есть информация о каждой программе, которая установлена в вашей системе. Как система знает все обо всех программах? Все очень просто. Разработчики программ под Linux договорились, что вместе с программой будет поставляться специальный `man`-файл — файл справочной системы. Понятно, если разработчик недобросовестный, он может и не создать файл справочной системы, но это происходит очень редко. Чтобы получить справку по какой-нибудь программе, нужно ввести команду:

`man имя_программы`

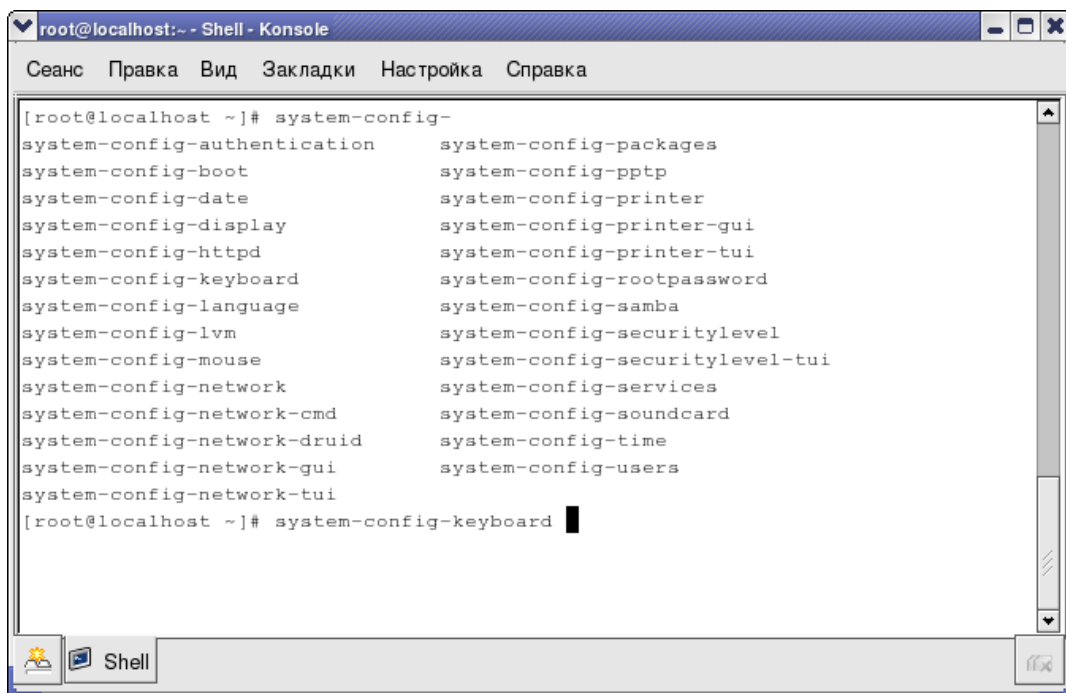


Рис. 19.1. Автодополнение командной строки

Вы никак не можете запомнить, как пишется та или иная команда? Если вы помните хотя бы, на какую букву она начинается, то воспользуйтесь функцией автодополнения командной строки: введите первые буквы команды и нажмите клавишу <Tab>. При первом нажатии система попытается дополнить команду, если это возможно. Иногда дополнить команду невозможно: например, вы ввели букву *a* и нажали <Tab>. Ясное дело, в системе есть несколько команд, которые начинаются на букву "a". Тогда система не дополнит командную строку. Если вы хотите просмотреть все команды на букву "a", тогда нажмите еще раз <Tab>.

Посмотрите на рис. 19.1. Сначала я ввел: `system-config-` и нажал <Tab>. Система не смогла дополнить командную строку, поскольку команд, начинающихся строкой "system-config-", несколько. При втором нажатии <Tab> система выдала список команд, которые начинаются строкой "system-config-". Затем я ввожу `system-config-ke`, и система сама за меня дописывает команду.

19.3. Служебные команды. Псевдонимы команд

Вам лень писать (даже с автодополнением) длинные команды? Тогда можно создать псевдонимы команд. Для этого в файл `.bash_profile` добавьте строки вида:

```
alias псевдоним='команда'
```

Например,

```
alias cfg-net='system-config-network'
```

ПРИМЕЧАНИЕ

С одной стороны, псевдонимы использовать удобно, с другой — не всегда полезно: вы привыкнете к "своим" командам, и, когда будете работать с другой системой, где не будут определены ваши псевдонимы, вам придется вспоминать "оригинальные" команды.

Для того чтобы изменения вступили в силу, выйдите из консоли (команда `logout`) и заново зарегистрируйтесь.

ПРИМЕЧАНИЕ

В файл `.bash_profile` можно добавить команды, которые будут выполнены при входе пользователя в систему. Например, для автоматического запуска файлового менеджера Midnight Commander в конец этого файла нужно добавить команду `mc`.

Кроме файла `.bash_profile`, в вашем домашнем каталоге вы обнаружите еще два файла:

- ◆ `.bash_history` — содержит историю введенных ранее команд;
- ◆ `.bash_logout` — содержит команды, которые будут выполнены при выходе пользователя из системы.

19.4. Приглашение командной строки и права пользователя

При работе с консолью обратите внимание на приглашение командой строки. Если оно заканчивается символом `$`, значит, вы работаете как обычный пользователь, а если вы работаете с правами `root`, то приглашение командной строки заканчивается символом `#`.

19.5. Эмуляторы консоли

Вам не хочется переходить из графического режима только для того, чтобы ввести пару команд? Тогда воспользуйтесь терминалом — эмулятором консоли. В большинстве случаев терминал можно вызвать из меню KDE: **К | Система | Терминалы**. На рис. 19.1 изображен самый популярный терминал — Konsole, входящий в состав KDE. Вообще-то консоль по-английски пишется `console`, но разработчики KDE решили явно указать принадлежность терминала к KDE, назвав его Konsole.

Напоследок позвольте представить одну очень небольшую команду. В DOS была очень полезная команда — `cls`, она очищала экран. Такая команда есть и в Linux, но называется она иначе — `clear`. Если хотите начать с "чистого листа", введите эту команду.

19.6. Перенаправление ввода/вывода

Иногда бывает полезно вывод одной программы перенаправить другой программе или в файл. Предположим, что есть команда `cmd`, которая выводит очень много информации — вы ее просто не успеваете прочитать. Тогда вывод этой программы можно перенаправить программе-просмотрщику, например, программе `less`, которая с помощью клавиш `<Pg Up>` и `<Pg Dn>` позволяет организовать просмотр длинного, как лимузин, фрагмента текста. На практике в роли команды `cmd` может выступить программа `cat`, которая используется для просмотра текстовых файлов. Если файл содержит текст, который не умещается на одном экране, целесообразно перенаправить его команде `less` для комфортного просмотра. Делается это так:

```
cat big_file.txt | less
```

Вы хотите сохранить вывод программы в файл, чтобы передать потом кому-то по электронной почте? Тогда вам нужно использовать символ `>`:

команда > файл

Например,

```
dmesg > kernel.txt
```

Данная команда перенаправляет вывод программы `dmesg`, выводящей загрузочные сообщения ядра, в файл `kernel.txt`. Если файл `kernel.txt` не существовал, он будет создан. А если существовал, то будет перезаписан. Если вы не хотите, чтобы `kernel.txt` был перезаписан, нужно использовать два символа `>>`:

```
dmesg >> kernel.txt
```

В этом случае вывод программы `dmesg` будет дописан в конец файла `kernel.txt`.

ГЛАВА 20



Полезные команды

20.1. Команды, о которых нужно знать каждому администратору

В Linux есть команды, которые нужно знать каждому администратору. О них мы и поговорим в этой главе. Для большего удобства команды разбиты на группы: общие команды, команды для работы с текстом, команды для работы в Интернете и команды системного администратора.

20.2. Общие команды

20.2.1. Команда *arch* — вывод архитектуры компьютера

Данная команда поможет узнать тип аппаратной платформы, например: i386, i586, i686 и др.

Пример использования:

```
$ arch  
i686
```

20.2.2. Команда *clear* — очистка экрана

Команда `clear` очищает экран при работе в консоли (терминале).

Пример использования:

```
$ clear
```

20.2.3. Команда *date*

Команда `date` используется для вывода текущей даты. Эта команда может применяться также для установки даты, если запущена от имени администратора.

Пример использования:

```
$ date
# date 1609171707
```

Первая команда выводит дату, а вторая команда устанавливает дату (при условии, что команда запущена от имени `root`) 16 сентября (1609) 2007 года (07) и время 17:17. Как видите, установка даты осуществляется в формате `MMddhhmmYY` (MM — месяц, dd — число, hh — часы, mm — минуты, YY — год).

Команда `date` может вывести дату в указанном вам формате. Для изучения форматов даты введите команду `man date`.

20.2.4. Команда *echo*

Команда `echo` выводит текстовую строку, указанную в качестве аргумента, например:

```
$ echo "Hello world!"
Hello world!
```

Обычно данная команда используется в сценариях командного интерпретатора для вывода сообщений на экран.

20.2.5. Команда *exit* — выход из системы

Для завершения сеанса работы в системе (при условии, что вы работаете в консоли) нужно использовать команду `exit`. Если не завершить сеанс работы, кто угодно сможет работать в системе под вашим именем (понятно, что во время вашего отсутствия за компьютером).

20.2.6. Команда *man* — вывод справки

Команда `man` применяется для получения справки о любой команде системы. Например, команда `man ls` выведет справку об использовании команды `ls`, которая выводит содержимое каталога. О том, как правильно использовать саму справочную систему, вам расскажет команда `man man`.

20.2.7. Команда *passwd* — изменение пароля

С этой командой мы уже знакомы. Данная команда обеспечивает изменение пароля пользователя, который ее запустил.

Суперпользователь `root` имеет право изменить пароль любого пользователя так:

```
# passwd ИМЯ_ПОЛЬЗОВАТЕЛЯ
```

20.2.8. Команда `startx` — запуск графического интерфейса X Org

Linux может запускаться на разных уровнях. На пятом уровне запуска графический интерфейс X Org (бывшее название X Window) запускается автоматически, если он был вообще установлен. На третьем же уровне запуск графического интерфейса не производится. Если же вам очень он нужен, то его можно запустить с помощью команды `startx`. Никаких параметров не нужно.

20.2.9. Команда `uptime` — информация о работе системы

Команда `uptime` (рис. 20.1) выводит статистическую информацию о работе системы: сколько времени прошло с момента последней перезагрузки (собственно, это и есть время "uptime"), сколько пользователей в данный момент подключено к системе и среднюю загрузку системы за последние 1, 5 и 15 минут.

20.2.10. Команда `users` — информация о пользователях

Команда выводит пользователей, подключенных к системе в данный момент (рис. 20.2).

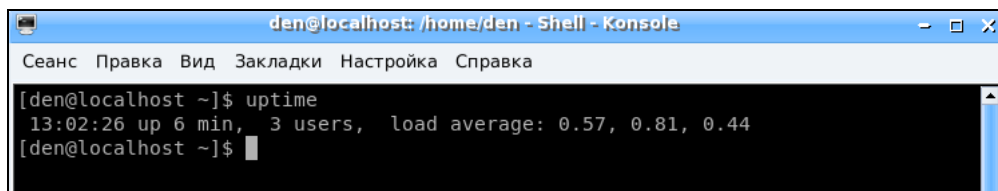


Рис. 20.1. Команда `uptime`

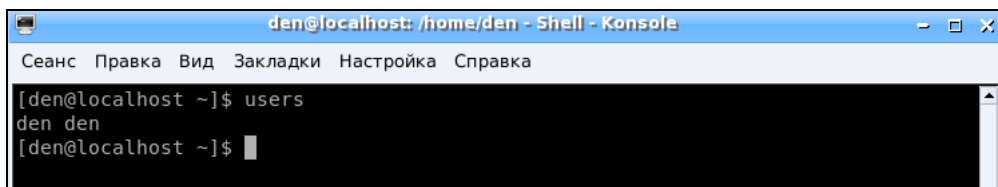


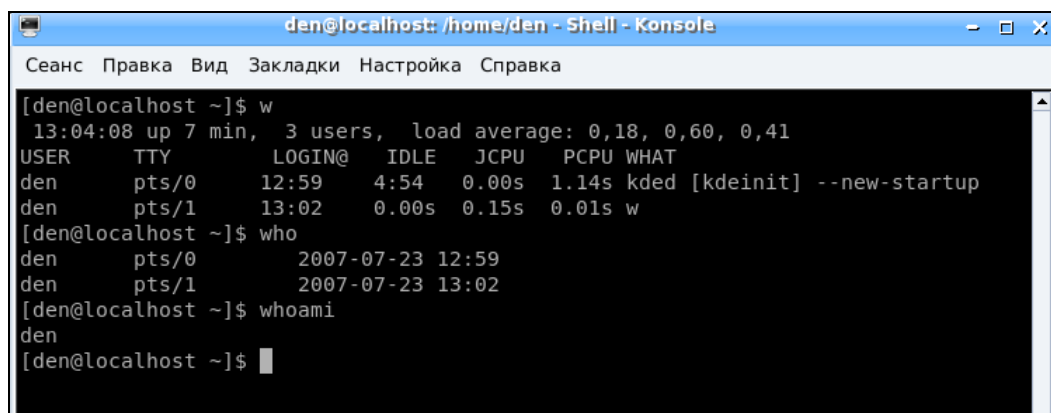
Рис. 20.2. Команда `users`

Из рис. 20.2 видно, что пользователь den подключился к системе двумя способами: в консоли и в графическом режиме (или по FTP, ssh, telnet — способы подключения к системе разные).

20.2.11. Команды *w*, *who* и *whoami* — информация о пользователях

Эти три родственные команды выводят следующую информацию:

- ♦ команда *w* — список пользователей, подключенных к системе; виртуальный терминал, с которого работает пользователь; время входа в систему для каждого пользователя, статистику использования системы (IDLE — время простоя, JCPU — использование процессора), выполняемые каждым пользователем задачи;
- ♦ команда *who* — список пользователей, подключенных к системе; время и дату входа каждого пользователя;
- ♦ команда *whoami* — имя пользователя, который ввел команду (рис. 20.3).



```
den@localhost: /home/den - Shell - Konsole
Сеанс  Правка  Вид  Закладки  Настройка  Справка

[den@localhost ~]$ w
 13:04:08 up 7 min,  3 users,  load average: 0,18, 0,60, 0,41
USER      TTY      LOGIN@  IDLE   JCPU   PCPU   WHAT
den       pts/0    12:59   4:54   0.00s   1.14s  kded [kdeinit] --new-startup
den       pts/1    13:02   0.00s   0.15s   0.01s  w
[den@localhost ~]$ who
den       pts/0    2007-07-23 12:59
den       pts/1    2007-07-23 13:02
[den@localhost ~]$ whoami
den
[den@localhost ~]$
```

Рис. 20.3. Команды *w*, *who* и *whoami*

20.2.12. Команда *xf86config* — настройка графической подсистемы

Текстовый конфигуратор системы — X Org (она же X Window). Использовать его нужно, только если в вашем дистрибутиве нет более удобных графических или псевдографических конфигураторов.

20.3. Команды для работы с текстом

20.3.1. Команда *diff* — сравнение файлов

Команда используется для сравнения двух файлов. Ее формат вызова такой:

```
diff параметры файл1 файл2
```

Результат сравнения выводится так: отличающиеся строки помечаются символами > и <. Строка из первого файла помечается символом <, а строка из второго файла — символом >.

Самые полезные параметры программы *diff* приведены в табл. 20.1.

Таблица 20.1. Некоторые параметры программы *diff*

Параметр	Описание
-b	Программа будет игнорировать пробельные символы в конце строки
-B	Игнорирует пустые строки
-e	Используется для создания сценария для редактора ed, который будет использоваться для превращения первого файла во второй
-w	Игнорирует пробельные символы
-y	Вывод в два столбца
-r	Используется для сравнения файлов в подкаталогах. Вместо первого файла указывается первый каталог, вместо второго файла указывается, соответственно, второй каталог

20.3.2. Команда *grep* — текстовый фильтр

Предположим, что у нас есть файл протокола `/var/log/messages`, и вы хотите вывести все сообщения, связанные с демоном `pppd`. Понятно, что вручную выделить все нужные сообщения будет довольно трудно. Но с помощью *grep* можно автоматизировать данную задачу:

```
cat /var/log/messages | grep ppp
```

Команда `cat /var/log/messages` передаст содержимое файла `/var/log/messages` на стандартный ввод программы *grep*, которая, в свою очередь, выделит строки, содержащие строку `ppp`.

Вообще, просматривать журналы удобнее с помощью команды *tac*, которая выводит строки файла в обратном порядке — ведь сообщения дописываются в конец журнала, следовательно, если выводит строки в обратном порядке, то сначала получим самые новые сообщения, а потом уже все остальные:

```
tac /var/log/messages | grep ppp
```

20.3.3. Команды *more* и *less* — постраничный вывод

Большой текстовый файл намного удобнее просматривать с помощью программ *less* или *more*. Программа *less* удобнее, чем *more*, если она есть в вашей системе:

```
tac /var/log/messages | grep ppp | less
```

20.3.4. Команды *head* и *tail* — вывод начала и хвоста файла

Команда *head* выводит первые десять строк файла, а *tail* — последние десять. Вообще количество строк может регулироваться с помощью параметра *-n*.

Пример использования:

```
head -n 10 /var/log/messages
```

```
tail -n 15 /var/log/messages
```

20.3.5. Команда *wc* — подсчет слов в файле

Команда *wc* служит для подсчета слов в текстовом файле для подсчета количества строк (если задан параметр *-l*) и символов (параметр *-c*).

Пример использования:

```
wc /var/log/messages
```

```
wc -l /var/log/messages
```

```
wc -c /var/log/messages
```

20.4. Команды системного администратора

20.4.1. Команды *free* и *df* — информация о системных ресурсах

Команда *free* выводит информацию об использовании оперативной и виртуальной памяти, а *df* — об использовании дискового пространства.

Из рис. 20.4 видно, что в системе установлено всего 384 Мбайт ОЗУ, из них 247 Мбайт занято и 137 Мбайт — свободно. На жестком диске */dev/sda1* всего 2,8 Гбайт дискового пространства, из них свободно — 1,66 Гбайт.

```

den@localhost: /home/den - Shell - Konsole
Сеанс  Правка  Вид  Закладки  Настройка  Справка

[root@localhost den]# free
              total        used         free       shared    buffers     cached
Mem:           385628        247604        138024           0         30584        116056
-/+ buffers/cache:      100964        284664
Swap:          168640           0         168640

[root@localhost den]# df
Файловая система  Разм  Исп  Дост  Исп% смонтирована на
/dev/sda1          2,8G  1,1G  1,6G  42% /
[root@localhost den]#

```

Рис. 20.4. Команды `free` и `df`

20.4.2. Команда `md5sum` — вычисление контрольного кода MD5

С целью проверки подлинности некоторых файлов, передаваемых через Интернет, используется алгоритм MD5, точнее контрольный код, вычисленный с использованием этого алгоритма. Разработчик программы выкладывает в Интернете пакет со своей программой и на своем сайте публикует контрольный код. Вы скачиваете пакет и вычисляете его контрольный код. Если коды отличаются, то файл при передаче был поврежден (или это другая версия пакета, которая, возможно, была подложена злоумышленником с целью установки вражеского кода в вашу систему).

Использовать программу нужно так:

```
md5sum файл
```

20.4.3. Команды `ssh` и `telnet` — удаленный вход в систему

Подробнее эти команды будут рассмотрены в *части VIII* этой книги, а пока можете почитать страницу руководства (`man`) по этим программам, если есть желание.

20.5. Команды `vi`, `nano`, `ee`, `mcedit`, `pico`: текстовые редакторы

Со времен первых версий UNIX в современные системы перекочевал текстовый редактор `vi`. То, что ему больше тридцати лет — видно сразу. Более неудобного редактора я не видел! Согласен, что тогда это был прорыв, но сегодня редактор смотрится уж очень архаично.

Некоторые гурманы (я бы их назвал мазохистами) говорят, что к нему нужно привыкнуть. Может и так, но сначала нужно изучить длинный `man` и выучить наи-

зудь команды редактора. Как такового интерфейса пользователя практически нет, можно сказать, что вообще нет — то, что есть, сложно назвать интерфейсом. Однако в этой книге мы рассмотрим `vi` хотя бы вкратце. Тому есть две причины. Первая — это критики. Мол, как это в книге, посвященной командной строке, не будет "классики". Вторая — некоторые системы, где по непонятным мне причинам до сих пор используется по умолчанию `vi`, а другие редакторы недоступны. Да, можно изменить переменную окружения `EDITOR`, но нет никакой гарантии, что в системе будет установлен какой-нибудь другой редактор.

Итак, приступим к рассмотрению редактора `vi`. Редактор `vi` может работать в трех режимах:

- ❖ *основной* (визуальный) *режим* — в нем и осуществляется редактирование текста;
- ❖ *командный режим* — в нем осуществляется ввод специальных команд для работы с текстом (если сравнивать `vi` с нормальным редактором, то этот режим ассоциируется с меню редактора, где есть команды вроде сохранить, выйти и т. д.);
- ❖ *режим просмотра* используется только для просмотра файла (если надумаете использовать этот режим, вспомните про команду `less`).

После запуска редактора вы можете переключать режимы (как, будет сказано позже), но выбрать режим можно и при запуске редактора:

`vi` *файл*

`vi -e` *файл*

`vi -R` *файл*

Первая команда запускает `vi` и загружает файл. Вторая команда запускает `vi` в командном режиме и загружает файл. Третья команда — это режим просмотра файла. Если указанный файл не существует, то он будет создан. По умолчанию активируется именно командный режим, поэтому в ключе `-e` нет смысла.

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
proxy:x:13:13:proxy:/bin:/bin/sh
www-data:x:33:33:www-data:/var/www:/bin/sh
backup:x:34:34:backup:/var/backups:/bin/sh
list:x:38:38:Mailing List Manager:/var/list:/bin/sh
irc:x:39:39:ircd:/var/run/ircd:/bin/sh
gnats:x:41:41:Gnats Bug-Reporting System (admin)/var/lib/gnats:/bin/sh
nobody:x:65534:65534:nobody:/nonexistent:/bin/sh
libuuid:x:100:101::/var/lib/libuuid:/bin/sh
syslog:x:101:102::/home/syslog:/bin/false
messagebus:x:102:106::/var/run/dbus:/bin/false
hplip:x:103:7:HPLIP system user,.,./var/run/hplip:/bin/false
avahi-autoipd:x:104:110:Avahi autoip daemon,.,./var/lib/avahi-autoipd:/bin/false
/etc/passwd [readonly] 33 lines, 1617 characters
```

Рис. 20.5. Редактор `vi`

После запуска `vi` главное знать, как из него выйти. Ведь в нем не будет привычной строчки меню, также редактор не будет реагировать на привычные комбинации клавиш вроде `<Alt>+<X>`, `<Ctrl>+<C>` тоже не поможет. На рис. 20.5 представлен редактор `vi`, в который загружен файл `/etc/passwd`.

В табл. 20.2 приведены основные команды редактора `vi`.

Таблица 20.2. Основные команды редактора `vi`

Команда	Описание
<code>:q!</code>	Выход без сохранения
<code>:w</code>	Сохранить изменения
<code>:w <файл></code>	Сохранить изменения под именем <code><файл></code>
<code>:wq</code>	Сохранить и выйти
<code>:q</code>	Выйти, если нет изменений
<code>i</code>	Перейти в режим вставки символов в позицию курсора
<code>a</code>	Перейти в режим вставки символов в позицию после курсора
<code>o</code>	Вставить строку после текущей
<code>O</code>	Вставить строку над текущей
<code>x</code>	Удалить символ в позицию курсора
<code>dd</code>	Удалить текущую строку
<code>u</code>	Отменить последнее действие

Команды, которые начинаются с двоеточия, будут отображены в нижней строке, остальные просто выполняются, но не отображаются. Как уже было отмечено, у редактора `vi` есть два основных режима (режим просмотра не считается) — режим команд и режим редактирования (визуальный). Переключение в режим команд осуществляется нажатием клавиши `<Esc>`. Нажатие клавиш `<i>`, `<a>` и других переключает редактор в режим вставки, когда набираемые символы трактуются именно как символы, а не как команды. Для переключения обратно в командный режим используется клавиша `<Esc>`. В некоторых случаях (например, когда вы пытаетесь передвинуть курсор левее первого символа в строке) переход в командный режим осуществляется автоматически.

Теперь немного практики, введите команду:

```
$ vi file.txt
```

Далее нажмите `<i>`, чтобы переключиться в режим вставки. Наберите любой текст, но постарайтесь не ошибаться, поскольку исправление ошибок в `vi` — дело, требующее отдельного разговора.

Затем нажмите <Esc> и введите :wq. После выхода из редактора введите команду:

```
cat file.txt
```

Так вы убедитесь, что файл создан и в него сохранен введенный вами текст. Теперь приступим к дальнейшему рассмотрению редактора. Если ввести не команду i, а команду a, то вы тоже перейдете в режим вставки, но с одним отличием. Введенный текст будет вставляться не перед символом, в котором находится курсор, а поле него. Также в режим вставки можно перейти командами o и O. В первом случае будет добавлена пустая строка после текущей строки, а во втором — перед текущей строкой, а весь дальнейший ввод воспринимается именно как ввод текста, а не команд.

Чтобы удалить символ, нужно перейти в режим команд и над удаляемым символом нажать <x>. Да, <Backspace> и <Delete> тут не работают. Точнее, <Backspace> работает, но для удаления последней непрерывно введенной последовательности символов. Например, у нас есть текст "vi — текстовый редактор". Вы перейдете в режим вставки и измените текст так "vi — неудобный текстовый редактор". Нажатие <Backspace> удалит слово "неудобный", но не сможет удалить тире и другие символы.

Чтобы удалить строку, в которой находится курсор, нужно использовать команду dd. Помните, что vi считает строкой не то, что вы видите на экране, а последовательность символов до первого символа новой строки (\n). Если строка длиннее 80 символов, то она переносится на две экранных строки и визуально выглядит как две строки, а не как одна.

Чтобы перейти в конец строки (<Home> и <End> тоже не работают, как вы успели заметить, если уже запускали vi), нужно ввести команду \$. При навигации курсор перемещается не по экранным линиям, а как раз по строкам текста.

Для отмены последней операции используется команда u. Вот только истории изменений нет, да и по команде u отменяется вся предыдущая команда целиком. Например, вы создали файл, перешли в режим вставки (команда i) и ввели весь текст большой медицинской энциклопедии. Если вы введете команду u, то она отменит всю предыдущую команду, т. е. удалит весь введенный вами текст. Так что будьте осторожны.

Азы vi я вам преподнес. Но не думаю, что вы будете им пользоваться. Если есть желание продолжить знакомство, введите команду:

```
man vi
```

А мы тем временем познакомимся с другими текстовыми редакторами. Самый удобный из известных мне текстовых редакторов — редактор nano (раньше он назывался pico и входил в состав почтового клиента pine). Редактор nano изображен на рис. 20.6.

Внизу (под текстом) есть подсказка по комбинациям клавиш для управления редактором. Символ ^ означает <Ctrl>. То есть для выхода из редактора нужно нажать <Ctrl>+<X>, а для сохранения текста — <Ctrl>+<O>.

В некоторых системах (например, в FreeBSD) вместо nano используется редактор ee. Он похож на nano, но подсказки выводятся до текста (вверху экрана), а не после него, но идея та же. Также довольно удобный редактор joe.

```
GNU nano 2.0.9          Файл: /etc/passwd

root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
proxy:x:13:13:proxy:/bin:/bin/sh
www-data:x:33:33:www-data:/var/www:/bin/sh
backup:x:34:34:backup:/var/backups:/bin/sh
list:x:38:38:Mailing List Manager:/var/list:/bin/sh
irc:x:39:39:ircd:/var/run/ircd:/bin/sh
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/bin/sh
nobody:x:65534:65534:nobody:/nonexistent:/bin/sh
libuuid:x:100:101::/var/lib/libuuid:/bin/sh

[ Прочитано 33 строки ]
^G Помощь      ^O Записать   ^R ЧитФайл   ^Y ПредСтр   ^K Вырезать   ^C ТекПозиц
^X Выход      ^J Выворнять ^W Поиск     ^V СледСтр   ^U ОтмВырезк  ^T Словарь
```

Рис. 20.6. Редактор nano

В пакет mc (файловый менеджер) входит довольно удобный редактор mcedit, который запускается при нажатии клавиши <F4> в mc (рис. 20.7). Но вы можете запустить редактор отдельно:

```
mcedit <имя файла>
```

Кстати, редакторы joe, nano и ее запускаются так же:

```
joe <имя файла>
```

```
nano <имя файла>
```

```
ee <имя файла>
```

```
/etc/passwd      [----]  0 L: [ 1+ 0 1/ 34] *(0 /1617b)= r 114 0x72
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
proxy:x:13:13:proxy:/bin:/bin/sh
www-data:x:33:33:www-data:/var/www:/bin/sh
backup:x:34:34:backup:/var/backups:/bin/sh
list:x:38:38:Mailing List Manager:/var/list:/bin/sh
irc:x:39:39:ircd:/var/run/ircd:/bin/sh
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/bin/sh
nobody:x:65534:65534:nobody:/nonexistent:/bin/sh
libuuid:x:100:101::/var/lib/libuuid:/bin/sh
syslog:x:101:102::/home/syslog:/bin/false
messagebus:x:102:106::/var/run/dbus:/bin/false
hplip:x:103:7:HPLIP system user...:/var/run/hplip:/bin/false
```

Рис. 20.7. Редактор mcedit

ГЛАВА 21



Командный интерпретатор bash

21.1. Автоматизация задач с помощью bash

Представим, что нам необходимо выполнить резервное копирование сервера. Нам нужно создать архивы каталогов `/etc`, `/home` и `/usr`. Понятно, что будут три команды вида:

```
tar -cvjf имя_архива.tar.bz2 каталог
```

Затем нам нужно записать все эти три файла на DVD с помощью любой программы для прожига DVD.

Если выполнять данную операцию раз в месяц (или хотя бы раз в неделю), то ничего сложного в ней нет. Но представьте, что вам нужно выполнять эту операцию каждый день или даже несколько раз в день? Думаю, эта рутинная работа вам быстро надоест. А ведь можно создать сценарий, который будет создавать резервные копии и записывать их на DVD! Все, что вам нужно — это вставить чистый DVD перед запуском сценария.

Можно также пойти иным путем. Написать сценарий, который будет делать резервные копии системных каталогов и записывать их на другой раздел жесткого диска. Ведь не секрет, что резервные копии делаются не только на случай сбоя системы, но и на случай некорректного изменения данных пользователем. Помню, удалил важную тему форума и попросил своего хостинг-провайдера сделать откат. Я был приятно удивлен, когда мне предоставили на выбор три резервные копии — мне осталось лишь выбрать наиболее подходящую. Не думаете же вы, что администраторы провайдера только и занимались тем, что три раза в день копировали домашние каталоги пользователей? Поэтому автоматизация — штука полезная, и любому администратору нужно знать, как автоматизировать свою рутинную работу.

21.2. Привет, мир!

По традиции напомним первый сценарий, выводящий всем известную фразу: "Привет, мир!" (Hello world!). Вся работа со сценариями выполняется обычно в консоли (или в терминале), но для редактирования сценариев вы можете использовать любимый графический редактор, например kedit (листинг 21.1).

Листинг 21.1. Первый сценарий

```
#!/bin/bash
echo "Привет, мир!"
```

Первая строка нашего сценария — это указание того, что он должен быть обработан программой `/bin/bash`. Обратите внимание: если между `#` и `!` будет пробел, то данная директива не сработает, поскольку будет воспринята как обычный комментарий. Комментарии начинаются, как вы уже догадались, с решетки:

```
# Комментарий
```

Вторая строка — это оператор `echo`, выводящий нашу строку. Сохраните наш сценарий под именем `hello` и введите команду:

```
$ chmod +x hello
```

Для запуска сценария введите команду:

```
./hello
```

Вы увидите строку:

```
Привет, мир!
```

Чтобы вводить просто `hello` (без `./`), сценарий нужно скопировать в каталог `/usr/bin` (если точно, то в любой каталог из переменной окружения `PATH`):

```
# cp ./hello /usr/bin
```

21.3. Использование переменных в собственных сценариях

В любом серьезном сценарии вы не обойдетесь без использования переменных. Переменные можно объявлять в любом месте сценария, но до места их первого использования. Рекомендуется объявлять переменные в самом начале сценария, чтобы потом не искать, где вы объявили ту или иную переменную.

Для объявления переменной используется следующая конструкция:

переменная=значение

Пример объявления переменной:

```
ADDRESS=www.dkws.org.ua
echo $ADDRESS
```

Обратите внимание на следующие моменты:

- ❖ при объявлении переменной знак доллара не используется, но он обязателен при использовании переменной;
- ❖ при объявлении переменной не должно быть пробелов до и после знака =.

Значение для переменной указывать вручную необязательно: его можно прочесть с клавиатуры или со стандартного вывода программы:

```
read ADDRESS
ADDRESS=`hostname`
```

Чтение значения переменной с клавиатуры осуществляется с помощью инструкции `read`. При этом указывать символ доллара не нужно. Вторая команда устанавливает в качестве значения переменной `ADDRESS` вывод команды `hostname`.

В Linux часто используются переменные окружения. Это специальные переменные, содержащие служебные данные. Вот примеры некоторых часто используемых переменных окружения:

- ❖ `HOME` — домашний каталог пользователя, который запустил сценарий;
- ❖ `RANDOM` — случайное число в диапазоне от 0 до 32 767;
- ❖ `UID` — ID пользователя, который запустил сценарий;
- ❖ `PWD` — текущий каталог.

Для установки собственной переменной окружения используется команда `export`:

```
# присваиваем переменной значение
$ADDRESS=ww.dkws.org.ua
# экспортируем переменную — делаем ее переменной окружения
# после этого переменная ADDRESS будет доступна в других сценариях
export $ADDRESS
```

21.4. Передача параметров сценарию

Очень часто сценариям нужно передавать различные параметры, например режим работы или имя файла/каталога. Для передачи параметров используются следующие специальные переменные:

- ❖ `$0` — содержит имя сценария;
- ❖ `$n` — содержит значение параметра (`n` — номер параметра);
- ❖ `$#` — позволяет узнать количество параметров, которые были переданы.

Рассмотрим небольшой пример обработки параметров сценария. Я понимаю, что конструкцию `case-esac` мы еще не рассматривали, но общий принцип должен быть понятен (листинг 21.2).

Листинг 21.2. Пример обработки параметров сценария

```
# сценарий должен вызываться так:
# имя_сценария параметр
# анализируем первый параметр
```

```
case "$1" in
  start)
    # действия при получении параметра start
    echo "Запускаем сетевой сервис"
    ;;
  stop)
    # действия при получении параметра stop
    echo "Останавливаем сетевой сервис"
    ;;
  *)
    # действия в остальных случаях
    # выводим подсказку о том, как нужно использовать сценарий, и
    # завершаем работу сценария
    echo "Usage: $0 {start|stop}"
    exit 1
    ;;
esac
```

Думаю, приведенных комментариев достаточно, поэтому подробно рассматривать работу сценария из листинга 21.2 не будем.

21.5. Массивы и bash

Интерпретатор bash позволяет использовать массивы. Массивы объявляются подобно переменным. Вот пример объявления массива:

```
ARRAY[0]=1
ARRAY[1]=2
echo ${ARRAY[0]}
```

21.6. Циклы

Как и в любом языке программирования, в bash можно использовать циклы. Мы рассмотрим циклы `for` и `while`, хотя вообще в bash доступны также циклы `until` и `select`, но они довольно редко используются.

Синтаксис цикла `for` выглядит так:

```
for переменная in список
do
    команды
done
```


Работает цикл так: при каждой итерации переменной будет присвоен очередной элемент списка, над которым будут выполнены указанные команды. Чтобы было понятнее, рассмотрим небольшой пример:

```
for n in 1 2 3;
do
    echo $n;
done
```

Обратите внимание: список значений и список команд должны заканчиваться точкой с запятой.

Как и следовало ожидать, наш сценарий выведет на экран следующее:

```
1
2
3
```

Синтаксис цикла *while* выглядит немного иначе:

```
while условие
do
    команды
done
```

Цикл *while* выполняется до тех пор, пока истинно заданное условие. Подробно об условиях мы поговорим в следующем разделе, а сейчас напомним аналог предыдущего цикла, т. е. нам нужно вывести 1, 2 и 3, но с помощью *while*, а не *for*:

```
n=1
while [ $n -lt 4 ]
do
    echo "$n "
    n=$(( $n+1 ));
done
```

21.7. Условные операторы

В *bash* доступно два условных оператора, *if* и *case*. Синтаксис *if* следующий:

```
if условие_1 then
    команды_1
elif условие_2 then
    команды_2
...
elif условие_N then
    команды_N
else
    команды_N+1
fi
```

Оператор `if` в `bash` работает аналогично оператору `if` в других языках программирования. Если истинно первое условие, то выполняется первый список команд, иначе — проверяется второе условие и т. д. Количество блоков `elif`, понятно, не ограничено.

Самая ответственная задача — это правильно составить условие. Условия записываются в квадратных скобках. Вот пример записи условий:

```
# переменная N = 10
[ N==10 ]
# переменная N не равна 10
[ N!=10 ]
```

Операции сравнения указываются не с помощью привычных знаков `>`, `<`, а с помощью следующих выражений:

- ◆ `-lt` — меньше;
- ◆ `-gt` — больше;
- ◆ `-le` — меньше или равно;
- ◆ `-ge` — больше или равно;
- ◆ `-eq` — равно (используется вместо `==`).

Использовать данные выражения нужно так:

```
[ переменная выражение значение|переменная ]
```

Например:

```
# N меньше 10
[ $N -lt 10 ]
# N меньше A
[ $N -lt $A ]
```

В квадратных скобках вы также можете задать выражения для проверки существования файла и каталога:

- ◆ `-e файл` — условие истинно, если файл существует;
- ◆ `-d каталог` — истина, если каталог существует;
- ◆ `-x файл` — условие истинно, если файл является исполнимым.

С оператором `case` мы уже немного знакомы, но сейчас рассмотрим его синтаксис подробнее:

```
case переменная in
    значение_1) команды_1 ;;
...
    значение_N) команды_N ;;
*) команды_по_умолчанию;;
esac
```

Значение указанной переменной по очереди сравнивается с приведенными значениями (`значение_1`, ..., `значение_N`). Если есть совпадение, то будут выполнены команды, соответствующие значению. Если совпадений нет, то будут выполнены команды по умолчанию. Пример использования `case` приведен в листинге 21.2.

ГЛАВА 22



Планировщики задач

22.1. Зачем нужен планировщик задач

Очень часто нужно периодически выполнять одни и те же действия. Например, каждый день проверять обновление антивируса (или раз в неделю — в зависимости от того, как часто выходят для него обновления) или каждые 30 минут — почту. Можно выполнять эти действия самому, но это не совсем удобно. Представьте, что ваш рабочий день будет начинаться с команды запуска программы обновления антивируса, а каждые 30 минут вы будете вводить программу проверки почты. Во-первых, это не очень удобно, а во-вторых, можно легко забыть выполнить ту или иную команду. Например, в пятницу вечером вы можете забыть выполнить команду создания резервной копии, а в понедельник утром что-то случится с сервером, и вы не досчитаетесь всего пользовательского каталога. Не очень приятно, правда?

22.2. Планировщик crond

В Linux есть специальный демон `crond`, позволяющий выполнять программы по расписанию. Откройте конфигурационный файл демона `crond` — `/etc/crontab` (листинг 22.1).

Листинг 22.1. Пример файла `/etc/crontab`

```
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root
HOME=/

# run-parts
01 * * * * root nice -n 19 run-parts --report /etc/cron.hourly
02 4 * * * root nice -n 19 run-parts --report /etc/cron.daily
```

```
22 4 * * 0 root nice -n 19 run-parts --report /etc/cron.weekly
42 4 1 * * root nice -n 19 run-parts --report /etc/cron.monthly
```

Параметр `SHELL` задает имя программы-оболочки, параметр `PATH` — путь поиска программ, `MAILTO` — имя пользователя, которому будет отправлен отчет о выполнении расписания, а `HOME` — домашний каталог `cron`.

Но самое главное — не эти параметры, а таблица расписаний. В нашем случае она выглядит так:

```
01 * * * * root nice -n 19 run-parts --report /etc/cron.hourly
02 4 * * * root nice -n 19 run-parts --report /etc/cron.daily
22 4 * * 0 root nice -n 19 run-parts --report /etc/cron.weekly
42 4 1 * * root nice -n 19 run-parts --report /etc/cron.monthly
```

Согласно этой таблице, каждый час будут выполняться программы из каталога `/etc/cron.hourly`, каждый день — из каталога `/etc/cron.daily`, каждую неделю — из каталога `/etc/cron.weekly` и раз в месяц — из каталога `/etc/cron.monthly`.

Предположим, вам нужно каждый день выполнять команду `update_av ftp://server.ru/bases/`. В каталоге `/etc/cron.daily` создайте файл `update_av` следующего содержания:

```
#!/bin/bash
update_av ftp://server.ru/bases/
```

Мы создали небольшой `bash`-сценарий (сценарий командного интерпретатора). Теперь сделаем его исполняемым, и все будет готово:

```
# chmod +x update_av
```

Правда, удобно? Но иногда нам нужно создать более гибкое расписание. Например, мы хотим, чтобы одна программа выполнялась в 7:00, а другая в 7:20. Тут простым добавлением сценария в каталог `/etc/cron.daily` уже не отделаешься. Чтобы создать такое расписание, вам нужно изучить формат записей таблицы расписаний. А формат следующий:

минуты (0-59) часы (0-23) день (1-31) месяц (1-12) день_недели (0-6, 0 — Вс)
команда

Чтобы реализовать наше расписание, нам нужно добавить в `/etc/crontab` следующие строки:

```
0 7 * * * /usr/bin/command1 arguments
20 7 * * * /usr/bin/command2 arguments
```

Первая команда будет запускаться каждый день в 7 часов утра, а вторая команда — тоже каждый день, но в 7:20.

Зная формат файла `crontab`, мы можем отредактировать стандартную таблицу расписания. Обратите внимание: команды, выполняемые ежедневно, будут запускаться в 4 часа утра. Это, конечно, удобно, но они не будут выполнены, если вы выключаете сервер на ночь. Поэтому давайте установим другое время, например, 8 часов утра:

```
0 8 * * * root nice -n 19 run-parts --report /etc/cron.daily
```

Аналогичная ситуация и с еженедельным запуском. Программы будут запущены не только в 4:22 утра, но еще и в воскресенье. На выходные вы точно выключите свой сервер (хотя это зависит от политики организации — ведь во многих организациях выключают на выходные все компьютеры). Поэтому целесообразно назначить запуск на понедельник в 8 часов 22 минуты:

```
22 8 * * 1 root nice -n 19 run-parts --report /etc/cron.weekly
```

С ежемесячным запуском вроде бы все нормально: программы будут выполняться в 4:42 первого числа каждого месяца. Хотя время можно было бы и изменить на 8:42:

```
42 8 1 * * root nice -n 19 run-parts --report /etc/cron.monthly
```

22.3. Планировщик anacron

Планировщик anacron — непосредственный родственник crond, дальнейшее его развитие. Главное преимущество anacron заключается в том, что он, в отличие от crond, учитывает время, когда компьютер был выключен. Планировщик crond родом из UNIX, а эта операционная система устанавливалась только на серверах, которые всегда включены. Предположим, что вам нужно каждый понедельник в 7 часов утра рассылать некоторую информацию вашим сотрудникам. Вы настроили crond так, чтобы он запускал сценарий отправки сообщений каждый понедельник в 7 утра. Но вот беда — в 6 часов утра выключили электричество, а включили его, скажем, в 7:20. Но 7:20 — это не 7:00, следовательно, crond не выполнит задание по отправке сообщений, а ваши сотрудники не получают важную информацию.

Anacron работает не так. Если он обнаружил, что некоторые задания не выполнены по тем или иным причинам (выключение электричества, перезагрузка компьютера), он обязательно выполнит их. Поэтому ваши сотрудники получают информацию, но с небольшой задержкой. Все же лучше, чем получить важную информацию лишь в следующий понедельник.

Но и у anacron есть свои недостатки. В частности, пользователи не могут создавать собственные расписания, а файл `/etc/anacrontab` может редактировать только root. К тому же crond является более гибким в настройке — например, вы можете точно указать часы и минуты, а в случае с anacron можно указать только период, когда будет выполнена команда.

Формат файла `/etc/anacrontab` выглядит так:

Период	Задержка	ID	Команда
--------	----------	----	---------

Например:

1	5	cron.daily	run-parts /etc/cron.daily
7	10	cron.weekly	run-parts /etc/cron.weekly
30	75	cron.monthly	run-parts /etc/cron.monthly

22.4. Разовое выполнение команд — демон atd

Иногда нужно просто выполнить конкретные команды в определенное время (однократно), поэтому редактировать таблицу crontab не совсем уместно. Данную задачу можно решить более рационально. Убедитесь, что у вас установлен и запущен демон atd. После этого введите команду:

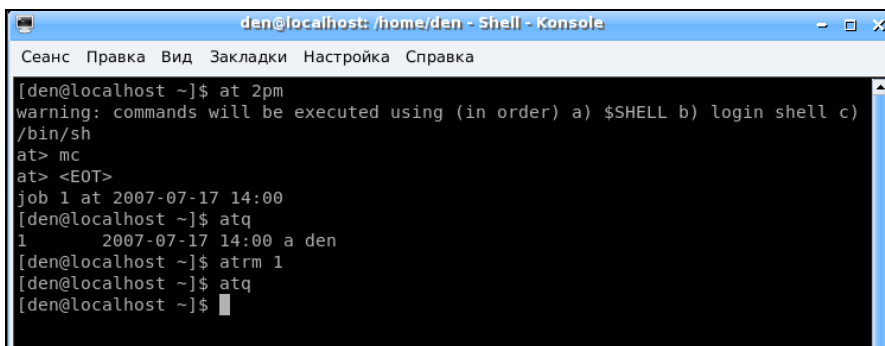
```
at <время> [дата]
```

После этого просто вводите команды, которые вы хотите выполнить в указанное время. Для завершения ввода нажмите комбинацию клавиш <Ctrl>+<D>. Время указывается в АМ/РМ-формате, например, если вам нужно выполнить команды в 14:00, то вы должны ввести команду:

```
at 2pm
```

Просмотреть очередь заданий можно командой atq, а удалить какое-либо задание — командой atrm.

На рис. 22.1 изображено добавление команды в очередь atd, просмотр очереди, удаление задачи и повторный просмотр очереди.



```
den@localhost: /home/den - Shell - Konsole
Сеанс  Правка  Вид  Закладки  Настройка  Справка
[den@localhost ~]$ at 2pm
warning: commands will be executed using (in order) a) $SHELL b) login shell c)
/bin/sh
at> mc
at> <EOT>
job 1 at 2007-07-17 14:00
[den@localhost ~]$ atq
1      2007-07-17 14:00 a den
[den@localhost ~]$ atrm 1
[den@localhost ~]$ atq
[den@localhost ~]$
```

Рис. 22.1. Использование atd



ЧАСТЬ VI

УСТАНОВКА ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

ГЛАВА 23



Пакет. Менеджер пакетов RPM

23.1. Что такое пакет?

В Windows программное обеспечение устанавливается с помощью мастера установки — программы `setup.exe` или `install.exe`. Мастер установки свой для каждой программы, т. е. программа `setup.exe`, предназначенная для установки MS Office, не установит Photoshop.

В Linux все иначе. В основном в Linux используются два способа установки программного обеспечения:

- ◆ с помощью пакетов;
- ◆ из исходных кодов.

Пакет содержит все необходимое для установки программы. Существуют два основных типа пакетов:

- ◆ RPM-пакеты — используются во всех Red Hat-совместимых дистрибутивах (Red Hat, Fedora Core, Mandrake, Mandriva, ALT Linux, ASPLinux и др.);
- ◆ DEB-пакеты — используются в дистрибутиве Debian и в дистрибутивах, основанных на Debian (Ubuntu, Kubuntu, Edubuntu и др.).

Если в вашем дистрибутиве нет нужной вам программы, попробуйте найти ее пакет на следующих сайтах: <http://rpmfind.net> и <http://rpm.pbone.net> (для RPM-пакетов) или на <http://www.debian.org/distrib/packages> и <http://packages.ubuntu.com/> (для DEB-пакетов).

ПРИМЕЧАНИЕ

Если же вы не можете найти пакет программы в Интернете, тогда придется компилировать программу самому (при условии, что вы нашли архив с исходным кодом программы). Да, в Linux некоторые программы распространяются только в исходных кодах. Для установки такой программы нужно распаковать архив (желательно в `/usr/src`) с исходными кодами, а затем перейти в только что созданный каталог (содержащий исходные коды устанавливаемой программы) и выполнить следующие команды:

```
./configure  
make  
make install
```


Сценарий `configure` проверит, содержит ли ваша система необходимые библиотеки или программы, после чего, если все нормально, будет создан `Makefile`. Если вы увидели сообщение об ошибке, внимательно прочитайте его и попытайтесь устранить причину ошибки, например, установить недостающую библиотеку. Ясно, что в случае ошибки вводить последние две команды не нужно.

Вторая команда на основании созданного файла `Makefile` компилирует программу. А последняя команда устанавливает программу и дополнительные файлы в дерево файловой системы (программы — обычно в `/usr/bin`, документацию — в `/usr/share/doc`, конфигурационные файлы — в `/etc` и т. д.).

Для получения подробных инструкций по установке и удалению таких программ лучше всего просмотреть файл `README`, который обычно присутствует в архиве.

Ясно, что устанавливаемая программа редко когда состоит из одного файла. Чаще она состоит из набора файлов, например исполняемый файл, конфигурационный файл, файл справки. В зависимости от организации программы установки Windows-программы все эти файлы могут быть:

- ❖ заархивированы каждый отдельно — в этом случае мы получаем набор из $N + 1$ файлов (N — это файлы программы плюс программа установки);
- ❖ заархивированы в один общий архив — у нас будут 2 файла, архив и программа установки;
- ❖ заархивированы в саму программу установки — самый удобный случай, когда у нас всего один файл — программа установки.

Как уже было отмечено, в Linux все файлы, относящиеся к той или иной программе, помещаются в один файл — пакет. Пакет — это не простой архив, содержащий файлы программы. В пакете, кроме файлов программы, хранится служебная информация, описывающая процесс установки программы, например:

- ❖ пути — ведь один файл нужно скопировать, например, в `/usr/bin`, а другой — в `/usr/share/doc`;
- ❖ дополнительные действия — например, создание каталога, установка тех или иных прав доступа к файлам и каталогам программы;
- ❖ зависимости — одна программа для своей работы может требовать какую-то библиотеку, без нее она не будет запускаться, поскольку использует функции этой библиотеки. Тогда в пакете указывается, что данный пакет зависит от другого пакета, содержащего библиотеку. При установке менеджер пакетов проверяет зависимости: если установлены не все пакеты, от которых зависит устанавливаемый пакет, установка будет прервана — пока вы не установите все необходимые пакеты. Правда, имеется возможность установки программы без удовлетворения зависимостей (тогда информация о зависимостях будет просто проигнорирована), но в большинстве случаев установленная программа работать не будет;
- ❖ конфликты — аналогично, программа может конфликтовать с другой программой в системе. Например, программа `sendmail` является МТА-агентом и программа `postfix` — тоже МТА-агент. В системе может быть только один МТА (Mail Transfer Agent). Следовательно, в системе пакет `sendmail` конфликтует с пакетом `postfix`, и наоборот.

Пакеты также называются RPM-файлами (DEB-файлами). С Debian все просто: пакеты были названы DEB-файлами, потому что последние три символа имени файла у файлов пакетов — `deb` (сокращение от Debian). А с RPM-файлами такая история: компания Red Hat разработала технологию RPM. Тогда в дистрибутиве Red Hat появился менеджер пакетов `rpm` (Red hat Package Manager) — отсюда и название пакетов.

В имени пакета зашифрована некоторая информация о программе. Сделано это исключительно для удобства: можно узнать версию и другую информацию о программе, только лишь взглянув на название пакета. Например:

```
program-1.5-14.i586.rpm
```

Здесь `program` — название программы, `1.5` — ее версия, `14` — выпуск пакета, `i586` — архитектура, на которую рассчитана программа. Не нужно пытаться устанавливать программы для архитектур `i586/686` на компьютер с процессором Intel 386 или 486. Если программа не зависима от архитектуры, то указывается `noarch` (обычно `noarch` указывается для документации, примеров конфигурационных файлов, т. е. для пакетов, содержащих информацию, которая не зависит от архитектуры).

23.2. Программы для управления пакетами

Для управления пакетами в разных дистрибутивах используются разные программы. В табл. 23.1 приведены программы управления пакетами, которые можно встретить в современных дистрибутивах.

ПРИМЕЧАНИЕ

Наверное, в таблице вы обратили внимание на фразу "умеет разрешать зависимости пакетов". Это значит следующее. Если будет обнаружено, что для установки пакета нужны дополнительные пакеты, то менеджер пакетов установит их. Если же менеджер пакетов не умеет разрешать зависимости, то он только сообщит, что установить пакет невозможно, и выведет список файлов (файлов, а не пакетов!), которые нужны для установки данного пакета. А уж какой файл в каком пакете находится, вам придется догадываться самостоятельно.

Таблица 23.1. Программы управления пакетами

Программа	Дистрибутив	Описание
<code>rpm</code>	RH-совместимые дистрибутивы (Fedora, Mandriva, ALT Linux, ASPLinux и др.)	Простой менеджер пакетов. Работает в текстовом режиме. Не умеет разрешать зависимости пакетов
<code>urpmi</code>	Mandriva	Программа для установки пакетов в Mandriva. Умеет разрешать зависимости пакетов и поддерживает источники пакетов

Таблица 23.1 (окончание)

Программа	Дистрибутив	Описание
rpmrake	Дистрибутивы, основанные на Mandrake (Mandriva)	Графический менеджер пакетов. Умеет разрешать зависимости и управлять источниками пакетов
dpkg	Дистрибутивы, основанные на Debian (Ubuntu, Kubuntu и др.)	Простой менеджер пакетов. Работает в текстовом режиме. Не умеет разрешать зависимости пакетов
apt-get	Debian, Ubuntu (и клоны), ALT Linux и др.	Мощный менеджер пакетов, работающий в текстовом режиме. Умеет разрешать зависимости пакетов и поддерживает репозитории (источники пакетов)
yum	Fedora и др.	Мощный менеджер пакетов, работающий в текстовом режиме. Умеет разрешать зависимости пакетов и поддерживает репозитории (источники пакетов)
zypper	openSUSE 10/11	Менеджер пакетов, работающий в текстовом режиме. Файлы конфигурации этого менеджера хранятся в каталоге /etc/zypp. Графическую оболочку менеджера можно запустить с помощью конфигулятора yast2. Программа описана в <i>главе 26</i>
system-config-packages	Fedora до версии 9 и дистрибутивы, основанные на нем (ASPLinux)	Графический менеджер пакетов. Впервые появился в одной из последних версий дистрибутива Red Hat, затем "перекочевал" в Fedora. По функциям похож на rpmrake, хотя последний все же удобнее. В любом случае в Fedora вам придется довольствоваться только этим менеджером (если не считать yum)
gtk-application	Fedora 9-13	Графический менеджер пакетов нового поколения. Более удобен, чем system-config-packages. Впервые появился в Fedora 9
gtk-repo	Fedora 9-13	Утилита управления источниками пакетов в Fedora 9. В более ранних версиях ее дистрибутива нет

В этой главе мы рассмотрим программы rpm и rpmrake. А в двух других главах поговорим о программах apt, dpkg, yum и system-config-packages.

23.3. Программа RPM (все RH-совместимые дистрибутивы)

Если вы хотите установить пакет, который не входит в состав дистрибутива, например вы загрузили его из Интернета, то вам нужно использовать программу `rpm` (для установки пакетов, которые входят в состав дистрибутива, намного удобнее использовать графический менеджер пакетов `rpm-drake`).

Данная программа — полноценный текстовый менеджер пакетов, позволяющий устанавливать, удалять пакеты, просматривать информацию об уже установленным и новым пакетах, обновлять пакеты.

Установить пакет с помощью `rpm` очень просто:

```
# rpm -ihv <имя_пакета>
```

Удалить пакет тоже просто:

```
# rpm -e <имя_пакета>
```

Для обновления пакета используется команда:

```
# rpm -U <имя_пакета>
```

Просмотреть, установлен ли тот или иной пакет, можно с помощью команды:

```
# rpm -qa | grep <имя_пакета>
```

Если вы хотите просмотреть информацию о пакете, то введите команду:

```
# rpm -qi <имя_пакета>
```

Просмотреть список файлов, входящих в состав пакета, можно командой:

```
# rpm -ql <имя_пакета>
```

Наконец, вывести все пакеты можно командой:

```
$ rpm -qa | grep more
```

ПРИМЕЧАНИЕ

Программа `rpm` может также использоваться для сборки собственных пакетов, но данная операция выходит за рамки этой книги. Если вам интересно, вы можете прочитать мою статью о сборке собственных RPM-пакетов на сайте http://www.dkws.org.ua/index.php?page=show&file=a/system/rpm_create.

23.4. Графический менеджер пакетов rpm-drake (Mandrake и Mandriva)

Для установки пакетов в Mandriva выполните команду главного меню **Установка и удаление программ**. Программа попросит ввести пароль `root` для продолжения работы.

ПРИМЕЧАНИЕ

Лично я предпочитаю открыть терминал и ввести команду `rpm-drake`, а не бродить по меню KDE/GNOME.

Программа `rpm-drake` (она же `drakrpm`) имеет несколько режимов отображения списка пакетов (выбор Mandriva, все пакеты по алфавиту, пакеты по группе) и два режима отображения информации о пакете (стандартная, максимальная информация). Если вы знаете, как называется пакет (хотя бы приблизительно), лучше просматривать список пакетов в режиме **Все**. Первый выпадающий список окна менеджера пакетов позволяет выбрать категорию пакетов (например, **Все**, **Пакеты с графическим интерфейсом** и т. д.). Дополнительные параметры списка пакетов можно найти в меню **Вид**. Второй выпадающий список позволяет отфильтровывать уже установленные пакеты и пакеты, доступные для установки. Если вы даже и приблизительно не знаете, что именно хотите установить, оптимальным является просмотр списка пакетов в сортировке по группам. Можно также ввести начальные буквы названия пакета в поле поиска и нажать клавишу <Enter> (рис. 23.1).

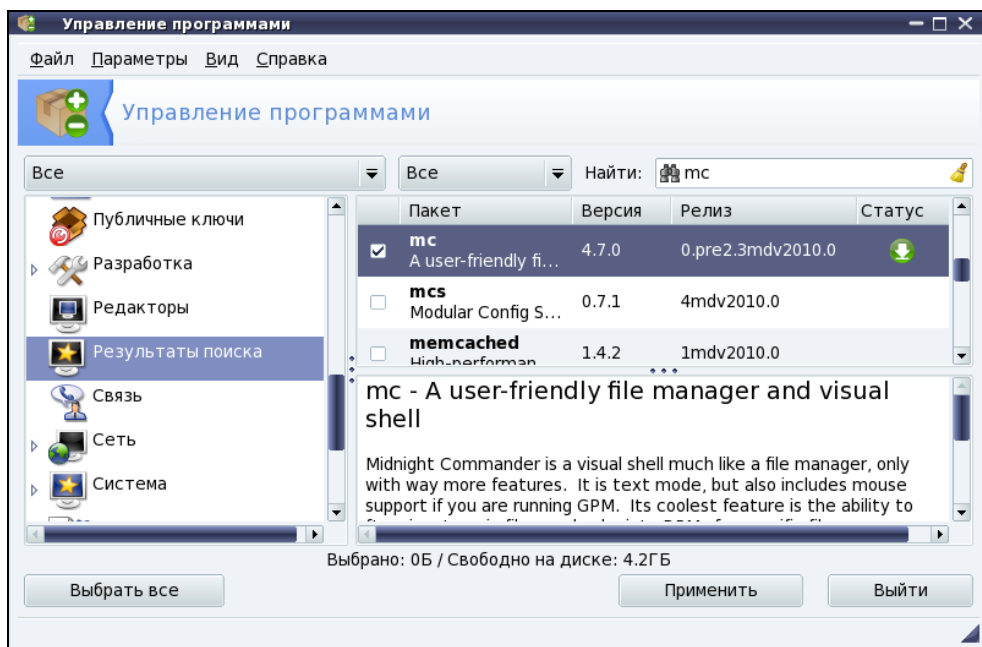


Рис. 23.1. Поиск пакетов

Искать можно в названиях пакетов, в описаниях и в именах файлов (способ поиска задается в меню **Вид**). Первый режим (**Все пакеты, по алфавиту**) удобен, если вы знаете приблизительное название пакета. Второй (**Все пакеты, по группам**) — если вы хотите найти сами не знаете что. Например, вы ищете игрушку, но не знаете, какую именно, — просто вам захотелось во что-то поиграть. Тогда в по-

ле поиска введите слово `game`, выберите режим **в описаниях** и нажмите кнопку **Поиск**.

Чтобы установить пакеты, отметьте их (возле каждого пакета выводится флажок) и нажмите кнопку **Применить**. Напротив уже установленных пакетов выводится зеленая пиктограмма со стрелкой вниз (справа от описания пакета). Если `rpm` обнаружит, что для установки вашего пакета нужно удовлетворить зависимости (т. е. установить дополнительные пакеты), то задаст вам соответствующий вопрос. Если вы согласитесь, установка будет продолжена, в противном случае — прервана.

Ранее для удаления пакетов использовался отдельный конфигуратор. Сейчас для удаления пакета достаточно снять флажок, выводящийся слева от имени пакета. При этом значок статуса пакета будет изменен: пакеты, помеченные для удаления, отмечаются красным значком со стрелкой вверх (рис. 23.2). Для применения изменения (т. е. для удаления пакетов) нужно нажать кнопку **Применить**.

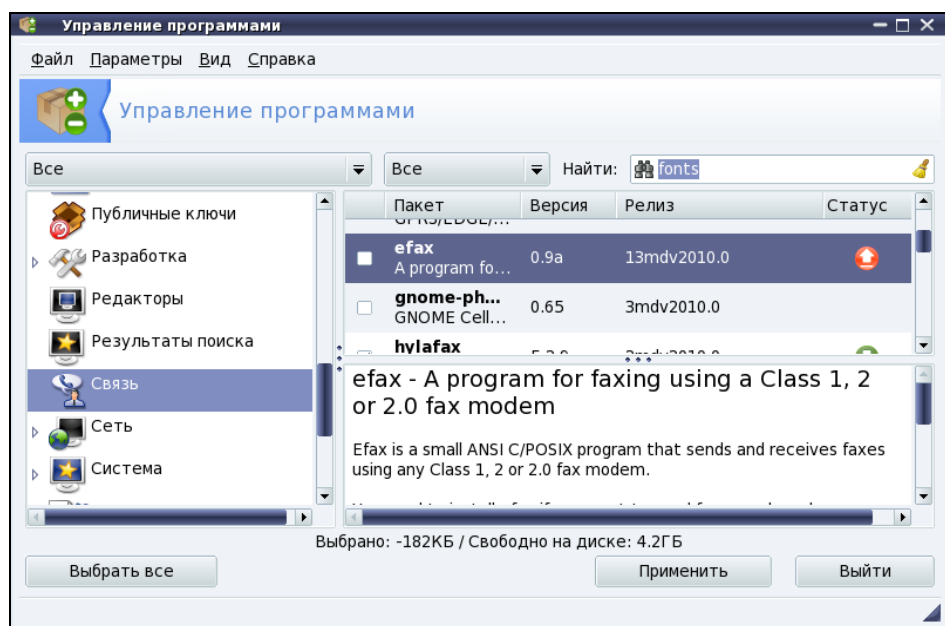


Рис. 23.2. Пакет `efax` помечен для удаления

Осуществляя поиск, программа `rpm` просматривает список еще не установленных пакетов, который формируется в результате исключения уже установленных пакетов из общей базы пакетов. Общая база пакетов — это совокупность дистрибутивных дисков, которые называются *источниками пакетов*. При желании вы можете добавить в список источники пакетов с Web- и FTP-серверов. Делать это нужно только, если у вас высокоскоростной (и дешевый) доступ к Интернету. В противном случае проще через некоторое время купить следующую версию дистрибутива.

Для редактирования источников пакетов выполните команду **Параметры | Менеджер источников** (рис. 23.3). Как видно из рис. 23.3, по умолчанию Mandriva настроена на установочный диск, а не на репозиторий Интернета.

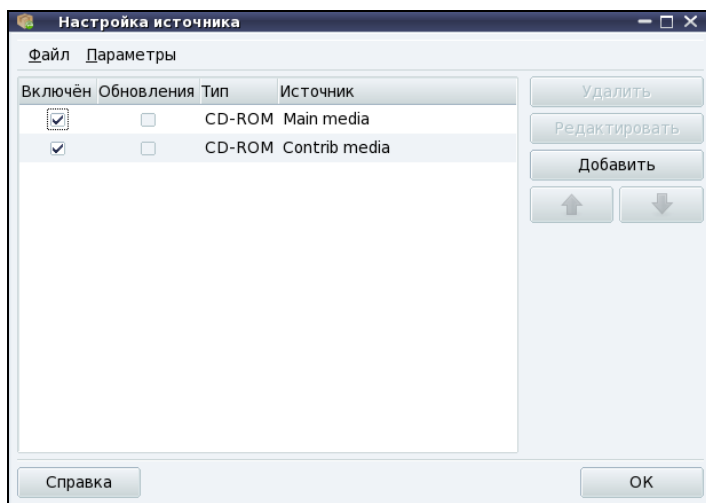


Рис. 23.3. Менеджер источников программ

23.5. Программа `urpmi`

Программа `urpmi` представляет собой систему управления пакетами, использующуюся в Mandriva. Как уже было отмечено в табл. 23.1, `urpmi` поддерживает зависимости пакетов. Конечно, обычным пользователям намного проще использовать программу `rpm-drake` для установки/удаления пакетов и управления источниками пакетов. Но `rpm-drake` — это всего лишь оболочка для системы `urpmi`, поэтому настоящий линуксоид должен знать, как работает `urpmi`.

Не нужно расценивать `urpmi` как замену `rpm` — система `urpmi` просто делает управление пакетами проще (хотя желающие могут использовать утилиту `rpm`, если сочтут ее более удобной).

ПРИМЕЧАНИЕ

Я, например, предпочитаю использовать `rpm` для локальной установки пакетов (когда пакет из какого-либо источника уже закачан на мой компьютер).

23.5.1. Установка пакетов. Управления источниками пакетов

Для установки пакета служит команда:

```
# urpmi <имя пакета>
```

Так, для установки пакета `mc` (файловый менеджер Midnight Commander) следует ввести команду:

```
# urpmi mc
```

Программа просматривает список источников пакетов, хранящийся в файле `/etc/urpmi/urpmi.conf`. Если она находит пакет в одном из источников, то устанавливает его вместе со всеми необходимыми для его работы пакетами (при этом `urpmi` автоматически разрешает зависимости пакетов).

Существуют три вида репозитариев, поддерживаемых `urpmi`:

- ❖ хранилища на съемных носителях (removable) — репозитории на компакт-дисках, DVD, ZIP-носителях, Flash-дисках и т. д.;
- ❖ локальные (local) — находятся в каталоге на жестком диске;
- ❖ удаленные (distant server) — пакеты находятся на удаленном FTP- или HTTP-сервере.

Просмотреть список источников пакетов можно с помощью команды:

```
# urpmq --list-media
```

Добавить источники пакетов можно с помощью команды:

```
# urpmi.addmedia <источник>
```

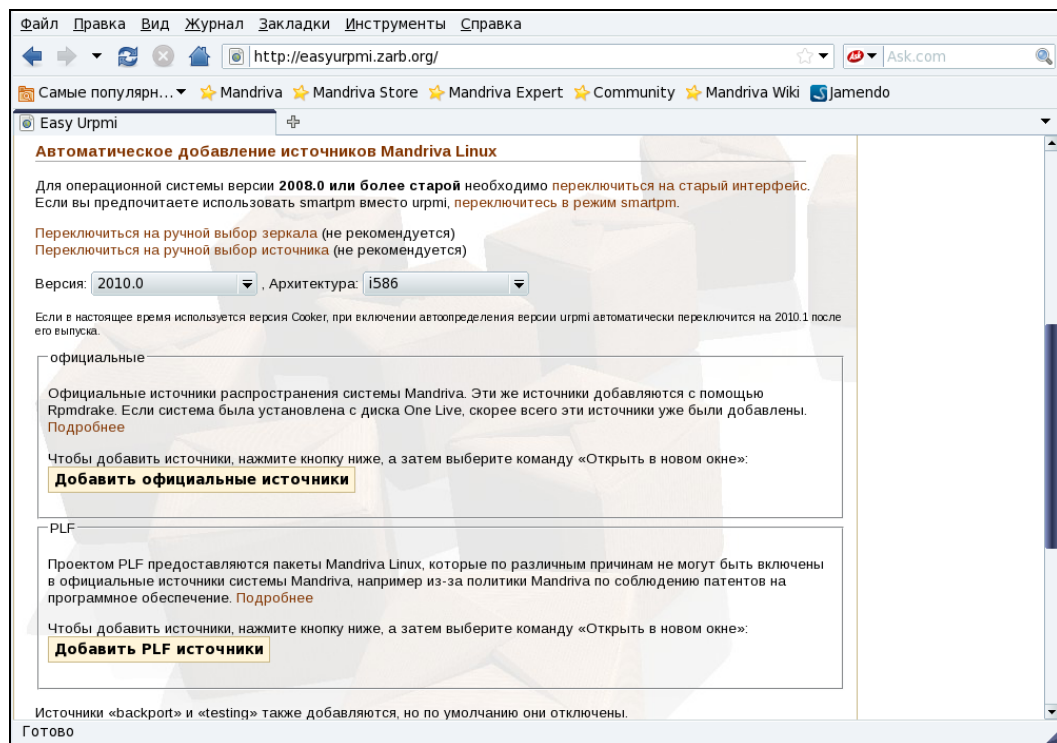


Рис. 23.4. Сайт easyurpmi.zarb.org

Получить список источников можно на сайте <http://easyurpmi.zarb.org> (рис. 23.4). Зайдите на этот сайт, выберите версию вашего дистрибутива (**2010.0**), архитектуру и нажмите кнопку **Добавить официальные источники**. В появившемся окне нажмите кнопку **ОК** (рис. 23.5). Браузер скачает файл источника пакетов, запустит средство добавления источника, которое запросит у вас пароль root, после этого нужно нажать кнопку **Да** (рис. 23.6) для установки источника пакетов. После установки официальных источников установите PLF-источники.



Рис. 23.5. Установка источника пакетов

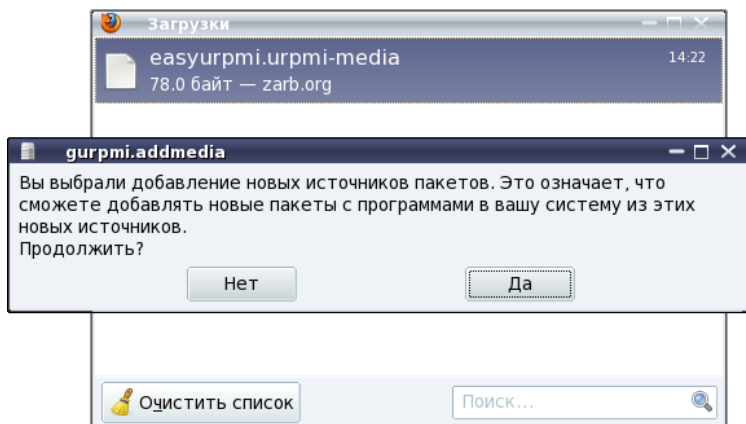


Рис. 23.6. Нажмите кнопку **Да** для установки источника пакетов

После добавления источников пакетов мой файл конфигурации `/etc/urpmi/urpmi.cfg` (Mandriva 2010, платформа i586) стал выглядеть так, как показано в листинге 23.1. Листинг я несколько сократил, потому что в противном случае он бы растянулся на 4 страницы.

Листинг 23.1. Фрагмент файла /etc/urpmi/urpmi.cfg

```
{
}

Main\ media cdrom://i586/media/main {
key-ids: 70771ff3
}

Contrib\ media cdrom://i586/media/contrib {
key-ids: 78d019f5
}

Main {
key-ids: 70771ff3
mirrorlist: http://api.mandriva.com/mirrors/basic.2010.0.i586.list
with-dir: media/main/release
}

Main\ Updates {
key-ids: 22458a98
mirrorlist: http://api.mandriva.com/mirrors/basic.2010.0.i586.list
update
with-dir: media/main/updates
}

...
Contrib {
key-ids: 78d019f5
mirrorlist: http://api.mandriva.com/mirrors/basic.2010.0.i586.list
with-dir: media/contrib/release
}

...
Non-free {
key-ids: 70771ff3
mirrorlist: http://api.mandriva.com/mirrors/basic.2010.0.i586.list
with-dir: media/non-free/release
}

...
debug_non-free_release {
ignore
key-ids: 70771ff3
mirrorlist: http://api.mandriva.com/mirrors/basic.2010.0.i586.list
```

```

with-dir: media/debug_non-free/release
}

...

PLF\ Free {
key-ids: caba22ae
mirrorlist: http://plf.zarb.org/mirrors/2010.0.i586.list
update
with-dir: media/../../../../2010.0/free/release/binary/i586
}

PLF\ Free\ debug {
ignore
key-ids: caba22ae
mirrorlist: http://plf.zarb.org/mirrors/2010.0.i586.list
with-dir: media/../../../../2010.0/free/release/debug/i586
}

...

```

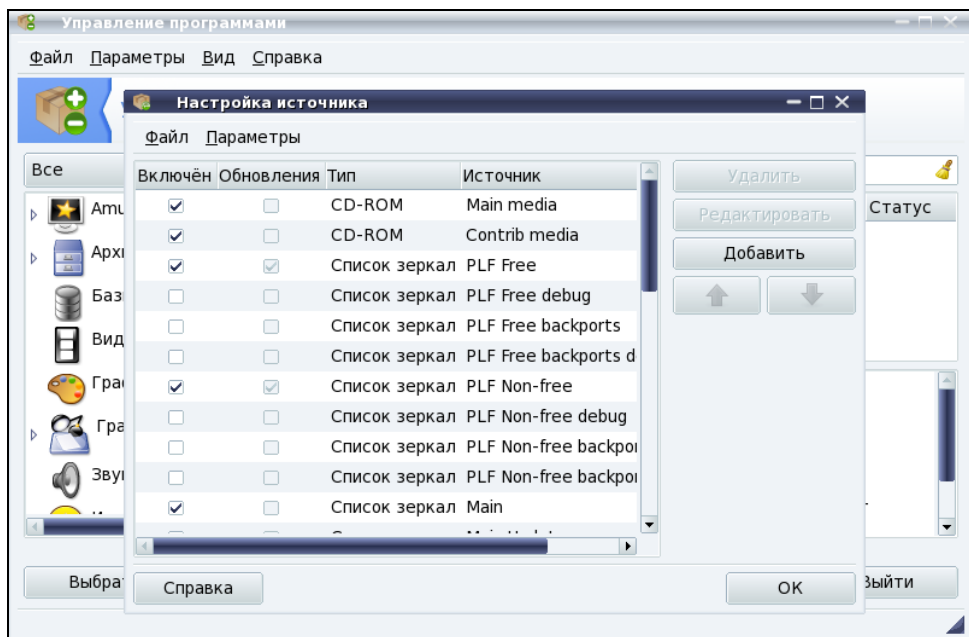


Рис. 23.7. Менеджер источников программ
(после установки дополнительных источников)

Для обновления репозитория (списка пакетов) используется команда:

```
# urpmi.update <ИМЯ ИСТОЧНИКА>
```

Удалить источник пакетов можно или путем удаления информации о нем из файла `urpmi.cfg`, или с помощью команды:

```
# urpmi.removemedиа <ИМЯ ИСТОЧНИКА>
```

Для обновления всего списка пакетов используется команда:

```
# urpmi.update -a
```

Если вам от редактирования конфигурационных файлов вручную становится не по себе, вы можете использовать один из графических менеджеров управления источниками пакетов — тот, который вам больше понравится. Для этого выполните команду меню **Параметры | Менеджер источников** (см. рис. 23.3). На рис. 23.7 изображен **Менеджер источников** после установки дополнительных источников пакетов.

23.5.2. Обновление и удаление пакетов

Для удаления пакета нужно ввести команду:

```
# urpme <пакет>
```

Если пакет нужен для работы других пакетов, то программа спросит у вас, хотите ли вы удалить и эти пакеты, иначе придется отказаться от удаления выбранного пакета.

Для обновления всей системы, т. е. получения списка новых версий пакетов, используется команда:

```
# urpmi --auto-select
```

23.5.3. Поиск пакета. Получение информации о пакете

Найти пакеты, содержащие в названии определенную строку, можно с помощью команды:

```
# urpmq <строка>
```

Команда `urpmf` позволяет получить различную информацию о пакете, например:

- ◆ `urpmf <файл>` — выводит пакеты, содержащие указанный файл;
- ◆ `urpmf --group <группа>` — выводит пакеты, входящие в указанную группу;
- ◆ `urpmf --size <пакет>` — выводит размер указанного пакета;
- ◆ `urpmf --summary <пакет>` — выводит общую информацию о пакете.

ГЛАВА 24



Программы dpkg и apt: установка пакетов в Debian/Ubuntu

24.1. Программа dpkg

Программа `dpkg` используется для установки, удаления и управления пакетами Debian/Ubuntu. Программа `dpkg` вызывается из командной строки. Формат вызова следующий:

`dpkg [ключи] действие`

Для запуска `dpkg` нужно обладать полномочиями `root`, получить которые можно с помощью команды `sudo`. Рассмотрим, как нужно использовать программу `dpkg`.

Предположим, у нас есть пакет `package.deb`. Для его установки откройте **Терминал (Приложения | Стандартные | Терминал)** и введите команду:

```
sudo dpkg -i /путь/package.deb
```

Как видите, для установки пакета не нужно делать ничего сложного. Если вам интересно, то процесс установки пакета состоит из следующих шагов:

1. Извлечение управляющих файлов из пакета.
2. Если уже была установлена старая версия этого пакета, тогда из старого пакета запускается сценарий `prepm` (данный сценарий подготавливает систему к удалению старой версии пакета). Другими словами, если нужно, то обновление пакета выполняется автоматически.
3. Выполняется сценарий `preinst`, если он есть в данном пакете.
4. Распаковываются остальные файлы из пакета (если был установлен старый пакет, то файлы не удаляются, а сохраняются в другом месте, чтобы их можно было восстановить, если что-то пойдет не так).
5. Если была установлена старая версия пакета, то выполняется сценарий `postrm` (действия после удаления) из старого пакета. Данный сценарий запускается сразу после выполнения сценария `preinst` нового пакета, поскольку старые файлы удаляются во время записи новых файлов.

6. Выполняется настройка пакета:

- ◆ распаковываются новые конфигурационные файлы, а старые сохраняются, если нужно будет их восстановить в случае ошибки во время установки нового пакета;
- ◆ запускается сценарий `postinst`, если он есть в данном пакете.

Удалить пакет тоже просто:

```
sudo dpkg -r package
```

При удалении пакета не нужно указывать путь к пакету и "расширение" пакета, т. е. символы `.deb` в конце имени файла.

Но установка и удаление пакетов — далеко не единственные действия, которые можно выполнить с помощью программы `dpkg`. Другие действия программы `dpkg`, которые могут быть интересны каждому пользователю *Ubuntu*, представлены в табл. 24.1.

Таблица 24.1. Вспомогательные действия программы `dpkg`

Действие	Описание
<code>-l [образец]</code>	Вывести все установленные пакеты, имена которых соответствуют образцу. Образец задается с помощью масок <code>*</code> и <code>?</code> , например, образец <code>a*</code> соответствует любому имени пакета, начинающемуся на букву "а". Если образец не задан, выводятся все пакеты
<code>-L <имя_пакета></code>	Выводит имена файлов из указанного пакета (пакет должен быть установлен)
<code>-p <имя_пакета></code>	Вывести информацию об установленном пакете
<code>-s <имя_пакета></code>	Выводит информацию о статусе пакета
<code>--unpack <имя_пакета.deb></code>	Распаковать, но не устанавливать пакет (полезно, если устанавливать пакет не нужно, а нужно достать из него один или несколько файлов)

Если вы хотите получить более подробную информацию о программе `dpkg`, введите команду (страница руководства будет на русском языке):

```
man dpkg
```

24.2. Программа `apt`

Относительно программы `apt` нужно отметить следующее. Данная программа используется не только в *Debian/Ubuntu*, но и в других дистрибутивах, причем даже в RH-совместимых, например в *ALT Linux*, но там она применяется для установки RPM-пакетов, а не DEB. Вообще, выбор менеджера пакетов зависит от разработчи-

ков дистрибутива. В одной версии дистрибутива может использоваться apt, в другой — uim, а в третьей — какой-то новый и перспективный менеджер пакетов.

Предположим, что у нас есть пакет `package.deb`. При его установке обнаружилось, что он требует пакет `lib.deb`, который не установлен. Вы находите в Интернете нужный пакет, устанавливаете его, а затем устанавливаете пакет `package.deb`. Не очень удобно, правда?

Намного проще выполнить команду:

```
sudo apt-get install package
```

Программа `apt-get` просматривает файл `/etc/apt/sources.list` — в этом файле перечислены источники (репозитории) DEB-пакетов. В качестве источника может выступать как компакт-диск, содержащий пакеты, так и сервер в Интернете. Программа находит указанный пакет, читает служебную информацию о нем, затем разрешает зависимости (т. е. устанавливает все другие пакеты, нужные для работы программ устанавливаемого пакета), а после устанавливает нужный нам пакет. Все загруженные программой `apt-get` и менеджером Synaptic (о нем — далее) пакеты записываются в каталог `/var/cache/apt/archives`.

Взглянем на файл `/etc/apt/sources.list`:

```
sudo gedit /etc/apt/sources.list
```

ПРИМЕЧАНИЕ

В Ubuntu стандартный текстовый редактор называется `gedit`. В Kubuntu его нет, поэтому для правки файла нужно использовать текстовый редактор Kate. А в Xubuntu текстовый редактор называется `mousepad`.

Наверное, вам интересно, какие программы находятся в том или ином репозитории Ubuntu? В репозитории `main` находятся основные программы, они распространяются свободно и регулярно поддерживаются (обновляются). В репозитории `restricted` содержатся программы, которые распространяются по несвободным лицензиям, а также имеют ограниченную поддержку. Репозиторий `universe` содержит программы с открытыми лицензиями, поддержка программ из этого репозитория не гарантируется, но вполне возможна, все зависит от разработчика программы. В репозитории `multiverse` содержатся программы, которые распространяются несвободно и без всякой поддержки и гарантий. Репозиторий `security` содержит исправления пакетов из репозитория `main` и `restricted`. Наконец, в репозитории `backports` есть неофициальные пакеты свежих версий программ, собранные из исходных текстов энтузиастами Ubuntu (а не разработчиками программ).

Чтобы настроить менеджер пакетов на русские репозитории (соответственно скорость загрузки пакетов будет выше), замените во всех строках файла `/etc/apt/sources.list` адрес `archive.ubuntu.com` на `ru.archive.ubuntu.com`.

Понятно, что программа `apt-get` может использоваться не только для установки пакетов. Общий формат вызова этой программы следующий:

```
apt-get [опции] команды [пакет]
```

Основные команды `apt-get` представлены в табл. 24.2.

Таблица 24.2. Основные команды `apt-get`

Команда	Описание
<code>update</code>	Синхронизирует файлы описаний пакетов (внутреннюю базу данных о пакетах) с источниками пакетов, которые указаны в файле <code>/etc/apt/sources.list</code>
<code>upgrade</code>	Обновляет указанный пакет. Может использоваться для обновления всех установленных пакетов. При этом установка новых пакетов не производится, а загружаются и устанавливаются только новые версии уже установленных пакетов
<code>dist-upgrade</code>	Обновление дистрибутива. Для обновления всех пакетов рекомендуется использовать именно эту команду
<code>install</code>	Установка одного или нескольких пакетов
<code>remove</code>	Удаление одного или нескольких пакетов
<code>check</code>	Используется для поиска нарушенных зависимостей
<code>clean</code>	Используется для очистки локального хранилища полученных пакетов (перед установкой пакет загружается в локальное хранилище, а затем устанавливается оттуда; данная команда может очистить хранилище для экономии дискового пространства)

24.3. Установка RPM-пакетов в Debian/Ubuntu

Если у вас есть RPM-файл, его можно преобразовать в формат DEB с помощью команды `alien`. Сразу хочу заметить, что установка таких пакетов не желательна, поскольку нет никакой гарантии, что установленная программа будет работать, но если другого выхода нет, можно попробовать:

```
sudo alien package_file.rpm
```

Если система сообщит вам, что команда `alien` не найдена, тогда ее нужно установить с помощью команды:

```
sudo apt-get install alien
```

Перед выполнением этой команды нужно подключиться к Интернету.

24.4. Графический менеджер Synaptic в Ubuntu

Дистрибутивы Debian/Ubuntu включают удобный графический менеджер пакетов Synaptic (рис. 24.1), запустить который можно с помощью команды меню **Сис-**

тема | **Администрирование | Менеджер пакетов Synaptic**. На самом деле Synaptic — просто оболочка для apt-get, но оболочка очень удобная. Рассматривать Synaptic подробно мы здесь не будем — он очень прост, и вы разберетесь без моих комментариев.

При инсталляции Debian следует иметь в виду, что дистрибутив Debian 4 поставляется на трех DVD, но по умолчанию в качестве репозитория прописывается только первый DVD, остальные два диска не задействуются. Понятно, что хочется использовать размещенные на них пакеты. Запустите Synaptic (**Система | Администрирование | Программа управления пакетами Synaptic**). Выполните команду меню **Настройки | Репозитории**, а в открывшемся окне нажмите кнопку **Добавить Cdrom**. Вставьте второй диск и нажмите кнопку **ОК**. Программа добавит второй DVD в список репозитариев и выведет сообщение о необходимости обновления источников пакетов. Для этого нажмите кнопку **Получить сведения** на панели Synaptic. Повторите все сказанное и для третьего DVD.

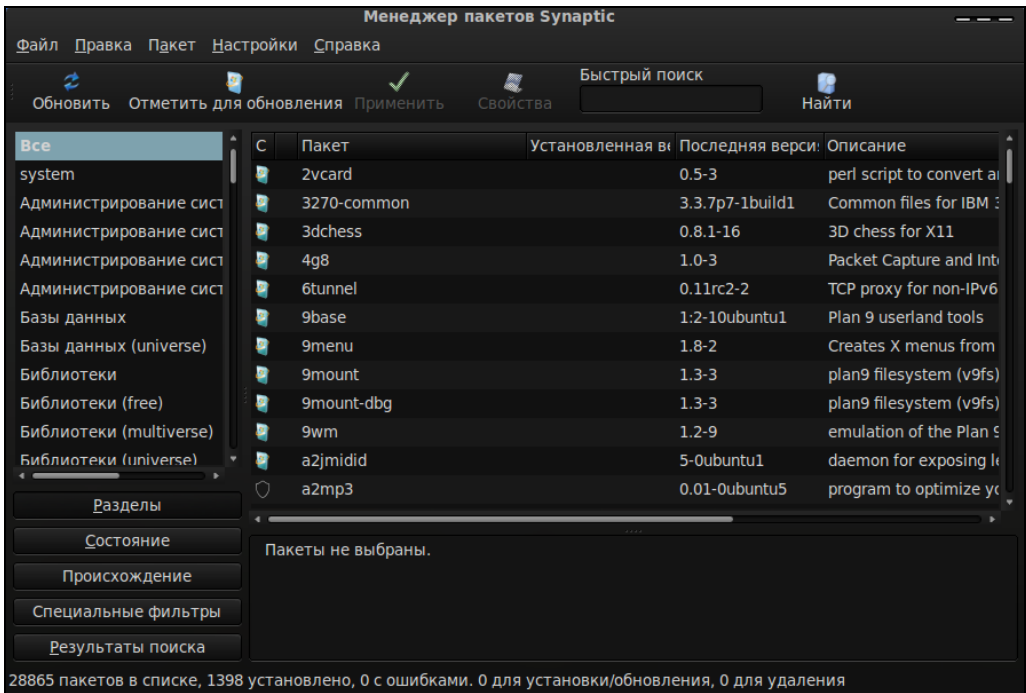


Рис. 24.1. Менеджер пакетов Synaptic

ГЛАВА 25



Программы yum и gprk-application

25.1. Программа yum

25.1.1. Общая информация о программе

Программа yum (Yellow dog Updater Modified) используется во многих дистрибутивах, в том числе и в Fedora.

Yum работает аналогично apt: когда вы устанавливаете пакет, yum производит поиск пакета в репозиториях, перечисленных в конфигурационном файле, загружает пакет и устанавливает его. В качестве репозитория могут выступать как дистрибутивные диски, так и серверы Интернета.

О настройке yum и его конфигурационных файлах мы поговорим в следующем разделе, а сейчас рассмотрим несколько простых команд, которые нужно знать каждому пользователю, работающему с yum (табл. 25.1).

Таблица 25.1. Использование yum

Команда	Описание
<code>yum install пакет</code>	Установка пакета из репозитория (также устанавливаются пакеты, необходимые для работы устанавливаемого пакета, т. е. разрешаются зависимости)
<code>yum remove пакет</code>	Удаляет пакет, а также все пакеты, которые зависят от данного
<code>yum update</code>	Проверить наличие обновлений всех пакетов. Если обновления есть, то они будут установлены
<code>yum update пакет</code>	Проверить обновления конкретного пакета. Если есть свежая версия, то она будет установлена
<code>yum check-update</code>	Только проверка наличия обновлений (обновления не устанавливаются)
<code>yum check-update пакет</code>	Проверка наличия обновлений конкретного пакета (обновления не устанавливаются)

Таблица 25.1 (окончание)

Команда	Описание
<code>yum info пакет</code>	Вывести информацию о пакете
<code>yum list</code>	Выводит список всех пакетов. Выводятся как установленные, так и доступные для установки (в репозиториях) пакеты
<code>yum list a*</code>	Вывести список всех пакетов, которые начинаются на букву "a"
<code>yum search строка</code>	Найти все пакеты, в описаниях которых есть указанная строка
<code>yum groupinstall "группа"</code>	Установить все пакеты из указанной группы
<code>yum grouplist</code>	Вывести список групп пакетов

25.1.2. Установка пакетов

При установке пакетов с помощью `yum` не нужно далеко отходить от компьютера. Довольно часто нужные пакеты находятся не на локальных источниках, а на серверах в Интернете, поэтому `yum` выведет общий объем пакетов, которые вы хотите установить, и спросит вас, хотите ли вы их установить или нет:

Total download size: 10.5 M

It this ok [Y/N]:

Если вы согласны для установки выбранных пакетов загрузить 10,5 Мбайт файлов, нажмите клавишу <Y>, если передумали — нажмите <N>. Довольно удобно, иначе (с учетом того, что при разрешении зависимостей будут установлены дополнительные пакеты) можно при установке одного небольшого, на первый взгляд, пакета превысить месячную норму по трафику.

Получить информацию о пакете, как было показано в табл. 25.1, можно с помощью команды:

```
yum info пакет
```

При этом на экран выводится следующая информация (рис. 25.1):

- ◆ **Name** — имя пакета;
- ◆ **Arch** — архитектура компьютера;
- ◆ **Epoch** — как бы подверсия пакета, поле Epoch используется, когда требуется уменьшить версию или релиз пакета по сравнению с имеющимся в репозитории;
- ◆ **Version** — версия пакета;
- ◆ **Release** — релиз пакета (можете считать это подверсией пакета);
- ◆ **Size** — размер занимаемого места на диске;
- ◆ **Repo** — хранилище пакета или значение **installed**, если пакет уже установлен;
- ◆ **Summary** — общая информация о пакете;
- ◆ **URL** — Web-страничка разработчика программы;
- ◆ **License** — лицензия, по которой распространяется программа;
- ◆ **Description** — описание пакета.

Для вывода всех пакетов можно использовать команду `yum list`, но пакетов слишком много, поэтому использовать ее неудобно. Удобнее задать маску имени пакета, например, `yum list a*` — в этом случае будут выведены все пакеты, начинающиеся на букву "а".

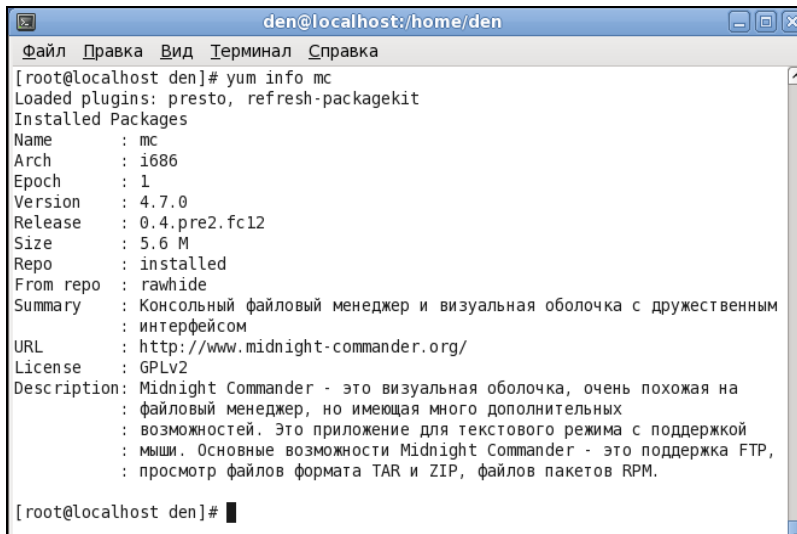


Рис. 25.1. Вывод информации о пакете

25.1.3. Управление источниками пакетов

Источники пакетов yum описываются в файле конфигурации `/etc/yum.conf`. Откройте этот файл (листинг 25.1).

СОВЕТ

Обычно файл `/etc/yum.conf` приходится редактировать редко. Но помните, что делать это можно только от имени пользователя `root`. Если вы привыкли к графическому режиму, тогда в терминале для редактирования этого файла нужно ввести команду:

```
su -c <редактор> /etc/yum.conf
```

В качестве редактора могут выступать программы `gedit` (если у вас GNOME), `kwrite` или `kate` (если у вас KDE). Если открыть данный файл в редакторе без прав `root`, то просмотреть его вы сможете, но не сможете сохранить изменения.

Листинг 25.1. Конфигурационный файл yum.conf

```
[main]
cachedir=/var/cache/yum
keepcache=0
debuglevel=2
logfile=/var/log/yum.log
```

```

exactarch=1
obsoletes=1
gpgcheck=1
plugins=1
metadata_expire=1800

# PUT YOUR REPOS HERE OR IN separate files named file.repo
# in /etc/yum.repos.d

```

Ранее репозитории описывались непосредственно в файле `yum.conf` (как в случае с `urpmi.cfg`). Но потом было принято решение хранить описания репозитариев в отдельных файлах (РЕПО-файлах) в каталоге `/etc/yum.repos.d`. Каждый файл в этом каталоге называется так: *<имя репозитория>.repo*.

В листинге 25.2 приведен пример описания источника пакетов Fedora, взятый из файла `fedora.repo`.

Листинг 25.2. Пример описания источника пакетов

```

[fedora]
name=Fedora $releasever - $basearch
baseurl=http://download.fedora.redhat.com/
pub/fedora/linus/releases/$releasever/Everything/$basearch/os/
mirrorlist=http://mirrors.fedoraproject.org/mirrorlist?repo=fedora-
$releasever&arch=$basearch
enabled=1
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-fedora file:///etc/pki/rpm-gpg/RPM-
GPG-KEY

```

Теперь разберемся, что здесь что. В квадратных скобках указывается сокращенное имя репозитария. Параметр `name` задает полное имя источника пакетов. Интернет-адрес (URL) источника пакетов указан параметром `baseurl`, а параметр `mirrorlist` задает список зеркал — копий репозитария, которые будут использоваться, если URL источника, указанный в `baseurl`, недоступен.

Параметр `enabled`, установленный в 1, указывает на то, что данный источник активный, и `yum` использует его при установке пакетов. Следующий параметр, `gpgcheck`, указывает на то, что `yum` должен проверить подпись источника (если `gpgcheck=1`), а ключ, используемый для проверки подписи, задан параметром `gpgkey`.

Добавление источника производится путем добавления соответствующего ему РЕПО-файла в каталог `/etc/yum.repos.d`. Где этот файл взять? Обычно такие файлы представлены в виде RPM-пакетов на Web-серверах репозитариев. Поэтому нужно просто скачать RPM-пакет и установить его. Например, для установки РЕПО-файла популярного репозитария RPM Fusion нужно выполнить команду:

```

su -c 'rpm -Uvh http://download1.rpmfusion.org/free/fedora/rpmfusion-free-
release-stable.noarch.rpm
http://download1.rpmfusion.org/nonfree/fedora/rpmfusion-nonfree-release-
stable.noarch.rpm'

```

Что делает данная команда, ясно и без комментариев. Если вы не можете найти соответствующий источнику REPO-файл, его можно написать вручную по формату листинга 25.2. При этом нужно еще знать базовый URL источника пакетов.

Удалять файлы источников пакетов, если сам источник уже не нужен, совсем не обязательно. Достаточно установить параметр `enabled` для источника в 0. Тогда этот источник не будет использоваться.

25.1.4. Установка пакетов через прокси-сервер

По умолчанию yum полагает, что наш компьютер напрямую подключен к Интернету (не через прокси-сервер). Если вы подключаетесь к Интернету по локальной сети, т. е. через прокси-сервер, данный факт нужно отразить в файле `yum.conf`, иначе вы не сможете устанавливать пакеты.

Узнайте у администратора сети параметры подключения к прокси-серверу (адрес, порт, имя пользователя и пароль) и пропишите их в файле `yum.conf` таким вот образом:

```
# Адрес прокси и его порт
proxy=http://proxy.company.ru:8080
# Имя пользователя и его пароль
proxy_username=dhsilabs
proxy_password=secret
```

25.1.5. Плагины для yum

Для yum доступно множество плагинов. Мы установим два плагина: `fastestmirror` и `presto`. Первый плагин позволяет найти самый быстрый источник пакетов, что существенно сокращает время установки пакетов. А второй пытается загружать только обновленные части пакетов вместо полной загрузки пакетов при обновлении, что сокращает трафик и уменьшает время обновления.

Для установки этих плагинов введите команды:

```
# yum install yum-plugin-fastestmirror
# yum install yum-presto
```

25.2. Графический менеджер пакетов в Fedora: gpk-application

В последних версиях Fedora используется графический менеджер пакетов, запустить который можно командой `gpk-application` или с помощью меню **Система | Администрирование | Add/Remove Software** (рис. 25.2). В старых версиях Fedora использовались конфигураторы `pirut` и `system-config-packages`.

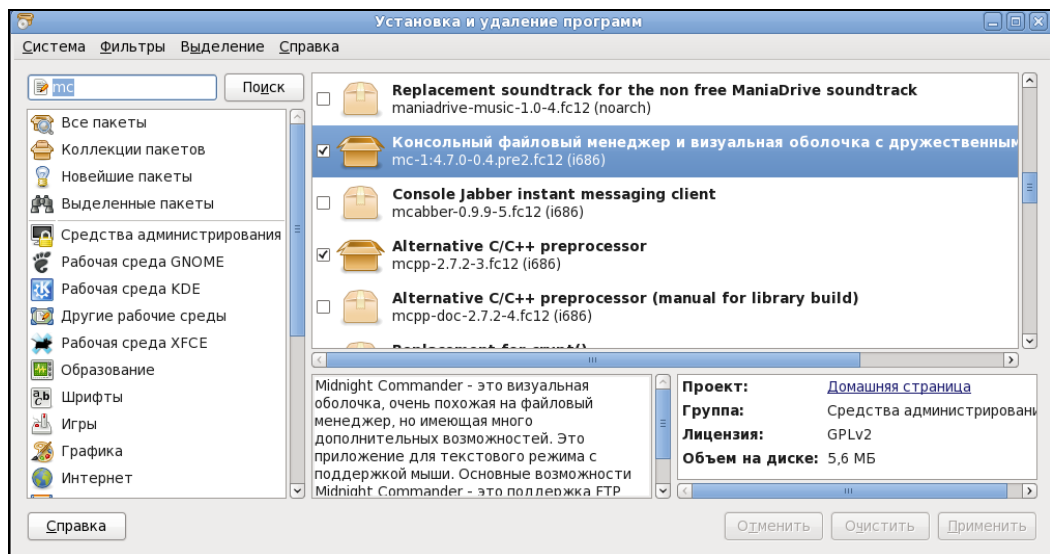


Рис. 25.2. Менеджер пакетов grk-application

Использовать этот графический менеджер не сложнее, чем любой графический менеджер пакетов (тот же rpm-drake). Слева от имени пакета выводится переключатель: включив его, вы помечаете пакет для установки, а выключив — для обновления. Нажав кнопку **Применить**, вы примените изменения, т. е. удалите или установите пакеты.

Как и во всех предыдущих версиях, менеджер пакетов Fedora по умолчанию настроен на использование интернет-репозитория для установки и обновления пакетов (а не на установочный диск, как Mandriva). В других моих книгах (как правило, изданных до 2010 года) описывалось, как заставить менеджер пакетов Fedora устанавливать пакеты с дистрибутивного диска. В этой книге подобного материала для Fedora 12/13 не будет. Во-первых, скорость Интернета выросла, стоимость доступа — снизилась и высокоскоростной Интернет теперь доступен почти каждому. А при установке пакетов с Интернета у вас будут всегда самые новые версии пакетов. Во-вторых, чуть ранее в этой главе мы установили два плагина, уменьшающие время загрузки пакетов и экономящие ваш трафик, поэтому не вижу более смысла использовать устаревшие пакеты с установочного DVD. Если у вас медленное соединение или вы принципиально желаете устанавливать пакеты с установочного диска, а не с интернет-репозитория, тогда посетите следующую страничку:

<http://www.dkws.org.ua/phpbb2/viewtopic.php?p=23984>

На ней, хотя и описывается настройка менеджера пакетов Fedora 9, не составит большого труда настроить "по образу и подобию" Fedora 12/13.

ГЛАВА 26



Управление пакетами в openSUSE

26.1. Источники пакетов *zypper*

Менеджер пакетов *zypper* работает по уже знакомому нам сценарию. Имеется список источников пакетов (каталог `/etc/zypp/repos.d`), который просматривается перед установкой пакета с целью определения хранилища, в котором находится устанавливаемый пакет. Затем менеджер пакетов загружает необходимый пакет (или пакеты) и устанавливает его.

Зайдите в каталог `/etc/zypp/repos.d`. В нем вы обнаружите несколько `REPO`-файлов, в каждом из которых прописан один репозиторий. В листинге 26.1 представлен репозиторий установочного DVD.

Листинг 26.1. Репозиторий установочного DVD (локальный репозиторий)

```
[openSUSE 11.3-0]
name=openSUSE 11.3-0
enabled=1
autorefresh=0
baseurl=cd:///
path=/
type=yast2
gpgcheck=1
keeppackages=0
```

Параметр `baseurl` задает путь к источнику пакетов, а параметр `enabled`, установленный в 1, говорит о том, что этот репозиторий активный.

Параметр `gpgcheck` означает проверку подписей GPG. Если ключей для репозитория нет, можно выключить этот параметр (однако из соображений безопасности это не рекомендуется). Выключать данный параметр нужно, только если вы полно-

стью доверяете источнику пакетов. Если включен параметр `keeppackages`, менеджер пакетов не будет удалять пакеты после их установки.

Пример сетевого источника пакетов Main Repository (OSS) приведен в листинге 26.2.

Листинг 26.2. Пример сетевого репозитория

```
[repo-oss]
name=openSUSE-11.3-Oss
enabled=1
autorefresh=1
baseurl=http://download.opensuse.org/distribution/11.3/repo/oss/
path=/
type=NONE
keeppackages=0
```

Как видите, параметр `baseurl` указывает не на локальное устройство, а на сервер в Интернете. Также обратите внимание на опцию `autorefresh` (автоматическое обновление) — для сетевого репозитория она установлена в 1, поскольку пакеты в хранилище могут меняться (например, там появляются новые версии пакетов). А для локального репозитория автоматическое обновление отключено, потому что пакеты в нем будут одни и те же.

Если установить опцию `keeppackages` в 1, то для этого репозитория менеджер пакетов будет сохранять все загруженные пакеты. Если `keeppackages = 0`, то после установки загруженный пакет удаляется.

Основной файл конфигурации менеджера пакетов называется `/etc/zypp/zypp.conf`, но в нем нет ничего интересного — обычно все опции там закомментированы, поскольку параметры по умолчанию устраивают всех, и их редко приходится менять.

ПРИМЕЧАНИЕ

Если у вас нет соединения с Интернетом или же оно медленное, вам придется использовать только один источник пакетов — локальный установочный DVD. Поэтому откройте терминал, введите команду `su`, а затем — `gedit`. Вы запустите обычный текстовый редактор от имени администратора. Перейдите в каталог `/etc/zypp/repos.d` и откройте все файлы кроме `openSUSE 11.2-0.repo` (этот файл описывает установочный DVD). Установите для всех сетевых источников пакетов параметр `enabled` в 0.

26.2. YMP-файлы

Файлы репозитариев обычно не нужно подключать вручную — вы скачиваете из Интернета YMP-файл, в котором описаны все необходимые репозитории и пакеты, которые нужно установить (хотя могут быть прописаны только репозитории — без пакетов). Данный файл представлен в формате XML (eXtended Markup Language). В секции `<repository>` описывается один репозитарий. Если репозитариев

несколько, то и секций `<repository>` будет несколько. В листинге 26.3 представлена секция `<repository>` YMP-файла для главного сетевого репозитория — Main Repository (OSS).

Листинг 26.3. Секция `<repository>` YMP-файла для главного сетевого репозитория

```
<repository recommended="true">
  <name>Main Repository (OSS)</name>
  <summary>Main OSS Repository</summary>
  <description>The largest and main repository from openSUSE for open source
software</description>
  <url>http://download.opensuse.org/repositories/openSUSE:11.3/standard/</url>
</repository>
```

Каждый пакет, который нужно установить, прописывается в отдельной секции YMP-файла: `<item>` (листинг 26.4).

Листинг 26.4. Секция `<item>` YMP-файла для установки пакета `w32codec-all`

```
<item>
  <name>w32codec-all</name>
  <summary>Win 32 Codecs</summary>
  <description>This packages contains the media player windows codec dlls
for several multimedia formats.</description>
</item>
```

Понятно, что если нужно установить несколько пакетов, то и секций `<item>` будет несколько.

ПОЯСНЕНИЕ

В листингах 26.3 и 26.4 приведены фрагменты файла `codecs-gnome.ymf`, благодаря которому в openSUSE устанавливается поддержка форматов мультимедиа.

ПРИМЕЧАНИЕ

Приведенная здесь информация нужна лишь для общего развития — вам никогда не придется изменять YMP-файлы (хотя кто знает, что нас ждет в этой жизни?), а установка таких файлов производится автоматически, практически без вмешательства пользователя.

26.3. Использование *zypper*

Теперь перейдем непосредственно к использованию менеджера пакета *zypper*. Формат вызова *zypper* следующий:

```
zypper <команда> [пакеты]
```

Основные команды `zypper` приведены в табл. 26.1.

Таблица 26.1. Основные команды `zypper`

Команда	Описание
<code>sl</code>	Выводит список используемых репозитариев
<code>sa URL имя</code>	Добавляет репозиторий (URL — адрес репозитория, а имя — имя, под которым он будет отображаться). Пример: <code>zypper sa http://ftp.uni-kl.de/pub/linux/suse/update/10.3 SUSE-Linux-10.3-Updates</code>
<code>sd URL имя</code>	Удаляет репозиторий. При удалении вы можете указать URL или имя репозитория
<code>install пакеты</code>	Устанавливает пакеты. Пример: <code>zypper install mc</code> Если нужно установить несколько пакетов, то имена пакетов разделяются пробелами
<code>search маска</code>	Ищет пакеты по маске. Маска — это часть имени (или полное имя) пакета. Пример: <code>zypper search mc*</code>
<code>list-updates</code>	Отображает доступные обновления
<code>update пакет</code>	Обновляет пакет. Если пакет не задан, обновляет всю систему
<code>info пакет</code>	Выводит информацию о пакете
<code>remove пакет</code>	Удаляет пакет



ЧАСТЬ VII

СЕТЬ И ИНТЕРНЕТ

ГЛАВА 27



Настройка локальной сети

27.1. Локальная сеть с использованием технологии Fast Ethernet

Существует много сетевых технологий, но в этой книге мы будем рассматривать настройку локальной сети, построенной на технологии Fast Ethernet. Зато мы рассмотрим ее полностью — от обжатия кабеля до конфигурирования сети в Linux.

Основные характеристики стандарта Fast Ethernet:

- ◆ скорость передачи данных — 100 Мбит/с;
- ◆ метод доступа к среде передачи данных — CSMA/CD;
- ◆ среда передачи данных — витая пара UTP 3-, 4- или 5-й категории (лучше 5-й), оптоволоконный кабель;
- ◆ максимальное количество компьютеров — 1024;
- ◆ максимальная длина сети — 200 м (272 м для оптоволоконка).

Прежде всего, вам нужно убедиться, что компьютеры, предназначенные для соединения в сеть, оснащены сетевыми адаптерами, поддерживающими технологию Fast Ethernet. Как правило, сейчас сетевые адаптеры интегрированы в материнскую плату и устанавливать их отдельно не нужно. Но встречаются материнские платы и без интегрированных сетевых адаптеров. В этом случае вам нужно их купить (рис. 27.1). Стоят они очень дешево — от 150 руб. за штуку. А за 500—800 руб. можно купить сетевой адаптер, поддерживающий технологию Gigabit Ethernet — модификацию Fast Ethernet, позволяющую передавать данные со скоростью до 1000 Мбит/с. Правда, коммутаторы (*см. далее*) для Gigabit Ethernet стоят чуть дороже, чем для Fast Ethernet.

Установка сетевого адаптера проблем не вызывает — просто вставьте ваш сетевой адаптер в свободный разъем шины PCI (все адаптеры Fast Ethernet выполнены как платы расширения именно для шины PCI). Существуют и USB-сетевые адаптеры, позволяющие подключиться к сети, не разбирая компьютер. Но такие адаптеры стоят очень дорого и встречаются пока редко. Тем более, точно не известно, как будет работать Linux с таким вот чудом научно-технического прогресса (рис. 27.2).

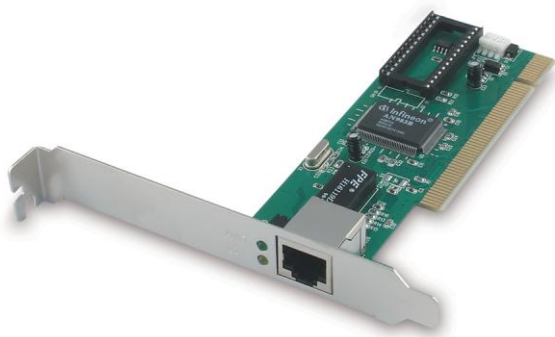


Рис. 27.1. Сетевой адаптер Fast Ethernet



Рис. 27.2. USB-сетевой адаптер

Ясно, что устанавливать сетевой адаптер полагается при выключенном компьютере — шина PCI пока еще не поддерживает "горячей замены". После этого нужно подключить к сетевому адаптеру коннектор сетевого кабеля. Коннекторы крепятся на концах отрезков кабеля (кабель "обжимается"), как правило, администратором сети.

ПОЯСНЕНИЕ

Обжать кабель — значит, особым образом закрепить на его концах специальные наконечники-коннекторы (см. далее).

Вы сами администратор сети и не знаете, как это 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. Дизайном и количеством портов коммутаторы могут отличаться друг от друга. На рис. 27.3 изображен 24-портовый коммутатор, больше подходящий для корпоративной сети (и внешним видом, и возможностью помещения в специальную стойку) нежели для дома. А для домашней сети можно найти более "симпатичное" устройство;
- ♦ сетевой кабель (витая пара 5-й категории) — приобретайте именно такой тип кабеля и такой длины, чтобы нормально хватило для соединения каждого компьютера сети с коммутатором;
- ♦ коннекторы RJ-45 — таких коннекторов вам понадобится в два раза больше, чем компьютеров, поскольку каждый отрезок кабеля нужно будет обжать с двух концов. Но я рекомендую купить еще несколько лишних штук — если вы будете обжимать кабель впервые, думаю, без ошибок не обойдется. Не пожалейте пару копеек, а то придется сбегать в магазин еще раз;
- ♦ инструмент (специальные обжимные щипцы) 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. 27.3.** Коммутатор (switch)

Теперь приступим к самому процессу обжимки. Внутри кабеля идут 4 витые пары проводов, причем у каждого провода своя цветовая маркировка. Суть процесса обжимки заключается в том, чтобы подключить каждый из проводов к нужному контакту коннектора. Сначала надо поместить провода в коннектор (защищать их необязательно — за вас это сделает инструмент), затем коннектор обратной частью (той, которой будет вставляться в сетевой адаптер) помещается в специальное гнездо обжимных щипцов, и их рукоятки сильно сжимаются. Используя данные табл. 27.1, вы без проблем сможете обжать кабель.

Таблица 27.1. Обжим витой пары

Контакт	Цвет провода	Контакт	Цвет провода
1	Бело-оранжевый	5	Бело-синий
2	Оранжевый	6	Зеленый
3	Бело-зеленый	7	Бело-коричневый
4	Синий	8	Коричневый

Один конец обжатого отрезка кабеля своим коннектором подключается к коммутатору (концентратору), а второй — к сетевому адаптеру компьютера. Если вы неправильно (или несильно) обожмете 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 имеются по два индикатора. Если горят оба — все нормально. Если же горит только один из них, то данный порт работает в режиме 10 Мбит/с. А если вообще не горит ни один из индикаторов, вам нужно переобжать кабель — отрезать плохо обжатые коннекторы и обжать концы кабеля новыми коннекторами заново.

Как видите, в процессе обжима нет ничего сложного.

27.2. Файлы конфигурации сети в Linux

Прежде чем приступить к настройке сети, следует ознакомиться с файлами конфигурации сети, которые имеются в любом дистрибутиве Linux, вне зависимости от его версии (табл. 27.2).

Таблица 27.2. Общие файлы конфигурации сети в Linux

Файл	Описание
/etc/aliases	База данных почтовых псевдонимов. Формат этого файла очень прост: <i>псевдоним пользователь</i>
/etc/aliases.db	Системой на самом деле используется не файл /etc/aliases, а файл /etc/aliases.db, который создается программой newaliases по содержимому файла /etc/aliases. Поэтому после редактирования этого файла не забудьте выполнить от имени root команду newaliases
/etc/hosts.conf	Содержит параметры разрешения доменных имен. Например, директива <code>order hosts,bind</code> означает, что сначала поиск IP-адреса по доменному имени будет произведен в файле /etc/hosts, а затем лишь будет произведено обращение к DNS-серверу, заданному в файле /etc/resolv.conf Директива <code>multi on</code> означает, что одному доменному имени могут соответствовать несколько IP-адресов
/etc/hosts	В этом файле можно прописать IP-адреса и имена узлов локальной сети, но обычно здесь указывается только IP-адрес узла localhost (127.0.0.1), потому что сейчас даже в небольшой локальной сети устанавливается собственный DNS-сервер
/etc/hosts.allow	Содержит IP-адреса узлов, которым разрешен доступ к сервисам данного узла
/etc/hosts.deny	Содержит IP-адреса узлов, которым запрещен доступ к сервисам данного узла
/etc/hostname	В Debian/Ubuntu содержит имя узла
/etc/iftab	Содержит таблицу интерфейсов, т. е. соответствие имен интерфейсов и их MAC-адресов
/etc/motd	Файл задает сообщение дня (Message of the day). Данный файл используется многими сетевыми сервисами (например, FTP- и SSH-серверами), которые при регистрации пользователя могут выводить сообщение из этого файла
/etc/network/interfaces	В Debian и Ubuntu используется для ручной настройки сетевых интерфейсов (не с помощью NetworkManager). Вообще принято настраивать сетевые интерфейсы с помощью NetworkManager, но некоторые администраторы предпочитают отключать NetworkManager и настраивать сетевые интерфейсы вручную — по старинке
/etc/rc.config	В openSUSE содержит имя компьютера, IP-адрес интерфейса и другую сетевую информацию

Таблица 27.2 (окончание)

Файл	Описание
/etc/resolv.conf	<p>Задаёт IP-адреса серверов DNS. Формат файла прост:</p> <pre>nameserver IP-адрес</pre> <p>Всего можно указать четыре DNS-сервера. В Ubuntu этот файл автоматически перезаписывается при установке соединения с Интернетом — сюда записываются адреса DNS-серверов, полученных от провайдера, что не совсем хорошо, особенно, когда вы настроили собственный DNS-сервер и желаете его использовать. О моей борьбе с перезаписью этого файла можно прочитать статью по адресу:</p> <p>http://www.dkws.org.ua/index.php?page=show&file=a/ubuntu/static-dns-ubuntu9</p>
/etc/route.conf	В старых версиях SUSE данный файл содержит описание статических маршрутов, в том числе и маршрут по умолчанию
/etc/services	База данных сервисов, задающая соответствие символического имени сервиса (например, ror3) и номера порта (110/tcp, tcp — это наименование протокола)
/etc/sysconfig/network	Параметры сетевого интерфейса в Fedora, Red Hat и других дистрибутивах, основанных на Fedora/Red Hat, например ASPLinux, Mandriva
/etc/sysconfig/static-routes	Статические маршруты в Fedora/CentOS/ASPLinux
/etc/sysconfig/network/routes	Статические маршруты в современных версиях openSUSE
/etc/sysconfig/network-scripts/ifcfg-имя	Параметры конкретного сетевого интерфейса, например параметры интерфейса eth0 хранятся в файле /etc/sysconfig/network-scripts/ifcfg-eth0 (дистрибутив Fedora)
/etc/sysconfig/network/ifcfg-имя	Параметры конкретного сетевого интерфейса (имя — имя сетевого интерфейса). Дистрибутив openSUSE
/etc/xinetd.conf	Файл конфигурации суперсервера xinetd, предназначенного для запуска сетевых сервисов, которые не работают в автономном режиме

27.3. Настройка сети с помощью конфигулятора

Настроить сеть в Linux можно за несколько минут. Ведь в большинстве случаев ваш сетевой адаптер поддерживается ядром, поэтому для настройки сети достаточно лишь указанной здесь командой запустить соответствующий конфигулятор:

♦ `drakconnect` — в Linux Mandriva;

- ◆ `system-config-network` — в Fedora и ASPLinux;
- ◆ `network-admin` — в Debian и Ubuntu;
- ◆ `nm-connection-editor` — в новых версиях Debian, Ubuntu и Fedora;
- ◆ `netconfig` — в Slackware.

А если в вашей сети организован DHCP-сервер, то настраивать сеть в современных дистрибутивах вовсе не придется — Linux автоматически распознает ваш адаптер, активирует соответствующие модули ядра и установит сетевые параметры, полученные от DHCP-сервера. Настраивать сеть придется в двух случаях:

- ◆ если у вас небольшая сеть, использующая статические IP-адреса — ради всего 2—3 компьютеров вы не стали настраивать DHCP-сервер;
- ◆ если вы настраиваете сеть "с нуля" и компьютер, на который вы установили Linux, как раз и будет тем DHCP-сервером, который потом станет настраивать остальные узлы сети.

ПРИМЕЧАНИЕ

Даже если у вас небольшая домашняя сеть из 2—3 компьютеров, совсем не обязательно отсутствие DHCP-сервера. Часто DHCP-сервер "крутится" на точке доступа Wi-Fi или на DSL-модеме, совмещающем также и функции коммутатора. Современные сетевые устройства позволяют существенно снизить стоимость монтажа сети, особенно домашней сети. Так, вы можете купить точку доступа с четырьмя Ethernet-портами (к которым могут подключаться стационарные компьютеры) и DSL-модемом. По сути, это единственное устройство обеспечивает все необходимые функции: ноутбуки будут подключаться по Wi-Fi, стационарные компьютеры — к встроенным портам Ethernet, а само подключение к Интернету будет происходить через встроенный DSL-модем. Вот только на предприятии от подобных устройств толку мало, разве что в самых небольших офисах, поскольку количество Ethernet-портов редко превышает 4, чего явно недостаточно для предприятия. Поэтому понадобятся дополнительные устройства — как минимум еще один коммутатор для подключения остальных компьютеров.

27.3.1. Настройка сети в Linux Mandriva

Перед началом настройки убедитесь, что сетевой кабель подключен и что запущен сервис `network`, обеспечивающий поддержку сети. Настройку сети мы будем производить на примере последней версии Mandriva — 2010.1 Spring. В принципе, конфигуратор настройки сети практически не изменился, поэтому все иллюстрации будут также соответствовать и предыдущей версии — 2010.0.

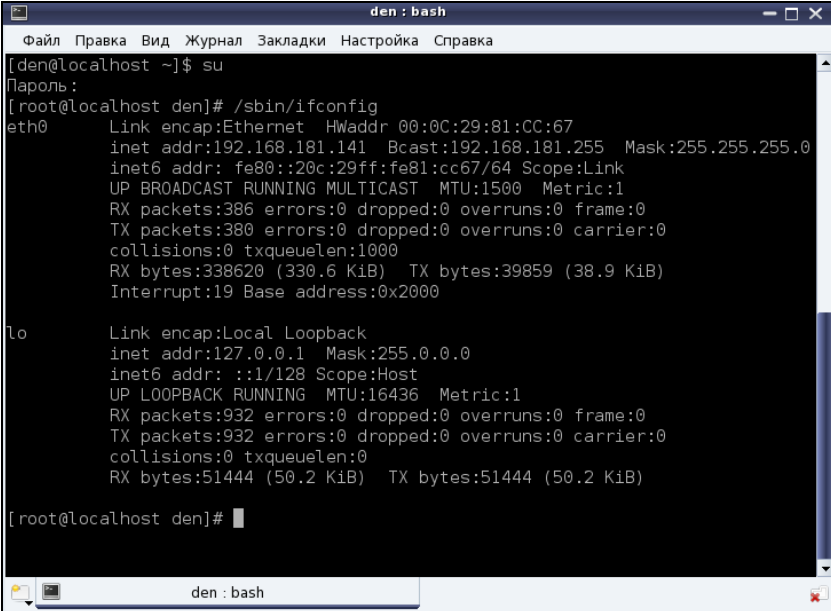
Убедиться в том, что запущен сервис `network`, очень просто — достаточно от имени пользователя `root` выполнить команду `/sbin/ifconfig`. Если в выводе команды вы увидите информацию об интерфейсе `lo` — все нормально (рис. 27.4).

ПРИМЕЧАНИЕ

Можно также запустить от имени `root` конфигуратор `drakxservices` и убедиться, что сервис `network` запущен.

ПОЯСНЕНИЕ

Интерфейс `lo` — это интерфейс обратной петли, использующийся преимущественно для тестирования поддержки сети.



```
den : bash
Файл Правка Вид Журнал Закладки Настройка Справка
[den@localhost ~]$ su
Пароль:
[root@localhost den]# /sbin/ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0C:29:81:CC:67
          inet addr:192.168.181.141  Bcast:192.168.181.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe81:cc67/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:386 errors:0 dropped:0 overruns:0 frame:0
          TX packets:380 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:338620 (330.6 KiB)  TX bytes:39859 (38.9 KiB)
          Interrupt:19 Base address:0x2000

lo        Link encap:Local Loopback
          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:932 errors:0 dropped:0 overruns:0 frame:0
          TX packets:932 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:51444 (50.2 KiB)  TX bytes:51444 (50.2 KiB)

[root@localhost den]#
```

Рис. 27.4. Вывод команды `ifconfig`

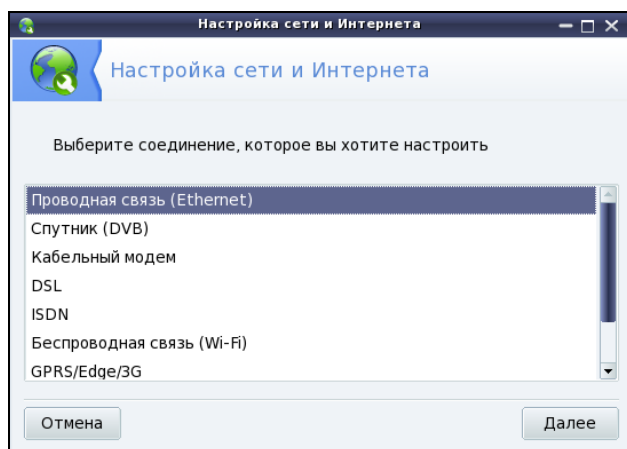


Рис. 27.5. Конфигуратор `drakconnect`: создание соединения по локальной сети

Если интерфейса `lo` нет в выводе программы, значит, вам нужно запустить сервис `network`:

```
# service network start
```

Для настройки локальной сети запустите конфигуратор drakconnect (рис. 27.5) и выберите тип соединения **Проводная связь (Ethernet)**¹.

Конфигуратор предложит вам выбрать устройство, которое будет использоваться для этого соединения, попросту говоря — сетевую плату (рис. 27.6). Если в вашем компьютере несколько сетевых плат, нужно выбрать именно ту, к которой подсоединен сетевой кабель, ведущий к сети, подключение к которой вы хотите настроить.

ПРИМЕЧАНИЕ

Если вы заметили, то до этого момента ничего не было сказано ни о моделях сетевых плат, ни о поддержке сетевых плат операционной системой. А дело в том, что Linux поддерживает практически все сетевые платы. Во всяком случае, неподдерживаемая сетевая плата мне еще не попадалась.

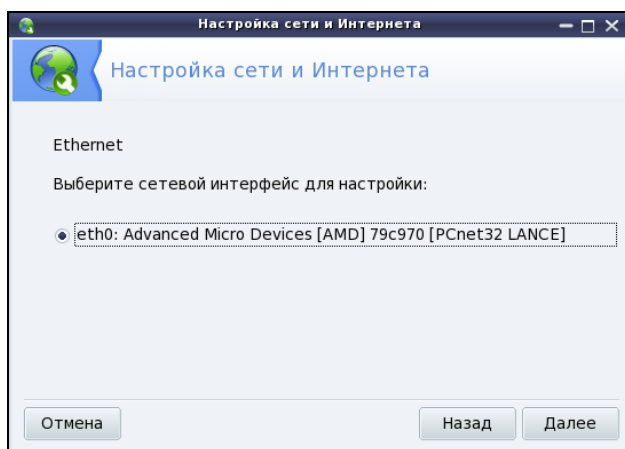


Рис. 27.6. Выбор сетевой платы

Следующий этап — выбор типа настройки (рис. 27.7): автоматический (с помощью DHCP) или ручной — в этом случае параметры TCP/IP вам нужно будет ввести вручную. Выбирать наугад не нужно — уточните тип настройки у администратора. Если в вашей сети развернут DHCP-сервер, то никаких параметров сети вам вводить не понадобится — в общем, на этом настройка вашей сети и закончится, поэтому далее мы будем рассматривать именно ручное конфигурирование сети.

ПРИМЕЧАНИЕ

Если вы выберете автоматическую настройку, то конфигуратор предложит вам изменить только параметры DNS: имя компьютера и IP-адреса DNS-серверов. Эту информацию можно или ввести вручную, или получить от DHCP (конфигуратор допускает выбор любого варианта — на ваше усмотрение).

¹ В предыдущей версии Mandriva — **Проводная сеть (Ethernet)**, в еще более древней версии — **Соединение по локальной сети** или просто **Ethernet**.

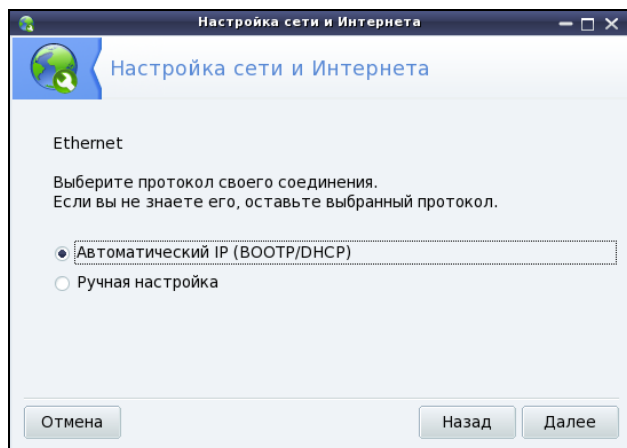


Рис. 27.7. Автоматическая или ручная настройка?

Получите у системного администратора значения параметров сети (IP-адрес сетевого интерфейса, маску сети, IP-адрес шлюза и адреса DNS-серверов), введите IP-адрес сетевого интерфейса и проверьте предложенную конфигуратором маску сети (рис. 27.8). Нужно отметить, что конфигуратор сам пытается вычислить маску сети по введенному IP-адресу, и в большинстве случаев у него это получается. В этом же окне можно ввести IP-адрес шлюза (если он есть в вашей сети), а также IP-адреса серверов DNS. В самом нижнем поле следует ввести имя узла (хоста).

Обратите внимание: что угодно вводить нельзя — имя узла должно быть зарегистрировано на DNS-сервере вашей сети.

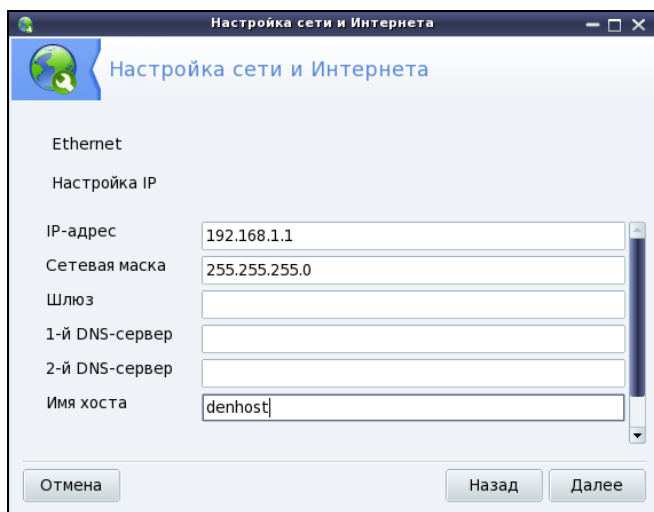


Рис. 27.8. Параметры TCP/IP

Если вы сам себе администратор, тогда для настройки локальной сети вы можете использовать следующие параметры:

- ◆ IP-адреса в диапазоне: 192.168.0.1—192.168.0.254;
- ◆ маска сети: 255.255.255.0 (сеть класса C);
- ◆ IP-адрес шлюза равен IP-адресу компьютера, подключенного к Интернету;
- ◆ если вы настраиваете шлюз, т. е. компьютер, который будет предоставлять доступ к Интернету другим компьютерам сети, то в его настройках IP-адрес шлюза указывать не нужно, а в качестве DNS-серверов можно указать IP-адрес этого компьютера (если вы планируете настройку собственного DNS-сервера) или IP-адреса DNS-серверов провайдера;
- ◆ имена узлов можно установить любые — главное, чтобы эти имена были уникальными (как и IP-адреса). Далее можно или настроить сервер DNS или, если сеть небольшая, прописать соответствие IP-адресов именам компьютеров в файле /etc/hosts. После редактирования этого файла (а редактировать его можно как в любом текстовом редакторе, так и с помощью конфигуратора сети) его нужно скопировать на все компьютеры сети.

Если вы хотите, чтобы соединение устанавливалось при загрузке системы (в большинстве случаев желательно, чтобы это было так), установите соответствующий флажок (рис. 27.9). Разрешать управлять соединением другим пользователям не стоит — ведь это соединение по локальной сети. Другое дело — модемное соединение, которое нужно включать и останавливать по несколько раз в день. Можно также включить подсчет трафика, а просмотреть информацию о трафике можно будет через средство мониторинга сети. Кстати, конфигуратор позволяет включить подсчет трафика и для автоматически настраиваемого интерфейса.

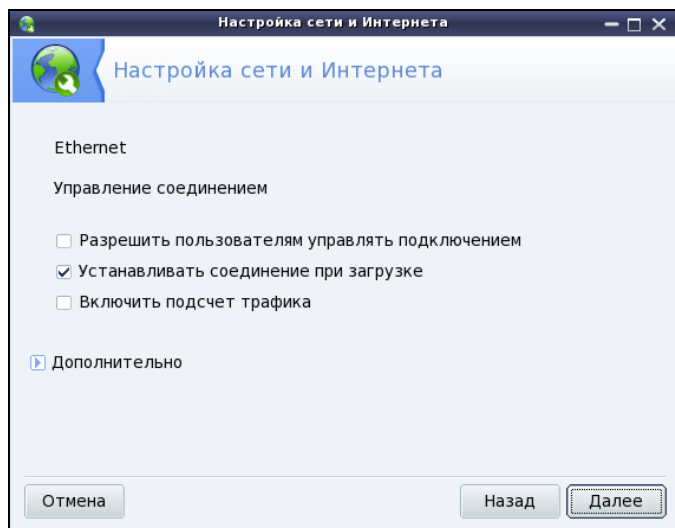


Рис. 27.9. Дополнительные параметры соединения

Далее конфигуратор предложит вам запустить созданное соединение — соглашайтесь. Все! При успешном "поднятии" сети (или автоматически, или вручную) вы увидите сообщение, подобное изображенному на рис 27.10.

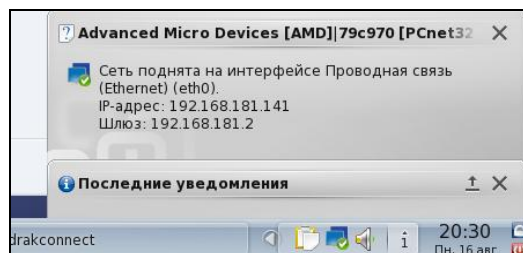


Рис. 27.10. Сеть успешно поднята по DHCP: компьютеру присвоен IP-адрес 192.168.181.141

Сеть настроена, можно приступить к тестированию ее работы, т. е. проверить правильность настроек.

Прежде всего, убедимся, что интерфейс eth0 (это ваша первая сетевая плата) поднят (т. е. включен и работает нормально). Введите команду:

```
ifconfig
```

В ее выводе (рис. 27.11) вы увидите информацию об интерфейсе eth0 (а также о других активных интерфейсах). Здесь же вы можете узнать IP-адрес интерфейса, маску сети, аппаратный MAC-адрес сетевой платы (HWaddr), количество принятых и переданных байтов (RX и TX соответственно).

ВНИМАНИЕ!

Если вы изменили имя узла, то нужно перезагрузить компьютер (команда `reboot`) или, хотя бы, перезапустить X.Org во избежание проблемы с графической подсистемой X.Org, которая не сможет нормально работать после изменения имени компьютера. Для перезапуска X.Org нужно завершить сеанс пользователя и снова войти в систему. Можно использовать также комбинацию клавиш `<Ctrl>+<Alt>+<Backspace>`, но это решение грубое и больше подходит для аварийного завершения X.Org в случае его зависания.

Итак, мы убедились, что интерфейс eth0 поднят, теперь пропингуем¹ свой узел по IP-адресу (рис. 27.12):

```
# ping 192.168.1.1
```

Для завершения работы программы `ping` нажмите комбинацию клавиш `<Ctrl>+<C>`.

Если ошибок не случилось, можно пропинговать удаленный узел, например ваш шлюз. Если произойдет ошибка при попытке пропинговать удаленный узел, это еще не означает, что ваш компьютер сконфигурирован неверно — вполне может быть, что удаленный компьютер просто выключен.

¹ Пропинговать — послать на проверяемый адрес специальный тестовый сигнал (ping).

```

den: bash
Файл Правка Вид Журнал Закладки Настройка Справка
[root@localhost den]# /sbin/ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0C:29:31:A4:46
          inet addr:192.168.1.1  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe31:a446/64  Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:668  errors:0  dropped:0  overruns:0  frame:0
          TX packets:561  errors:0  dropped:0  overruns:0  carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:335959 (328.0 KiB)  TX bytes:48349 (47.2 KiB)
          Interrupt:19  Base address:0x2000

lo        Link encap:Local Loopback
          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:1386  errors:0  dropped:0  overruns:0  frame:0
          TX packets:1386  errors:0  dropped:0  overruns:0  carrier:0
          collisions:0 txqueuelen:0
          RX bytes:76940 (75.1 KiB)  TX bytes:76940 (75.1 KiB)

[root@localhost den]#

```

Рис. 27.11. Информация об интерфейсе eth0

```

[root@localhost den]# ping 192.168.1.1
PING 192.168.1.1 (192.168.1.1) 56(84) bytes of data:
64 bytes from 192.168.1.1: icmp_seq=1 ttl=64 time=1.66 ms
64 bytes from 192.168.1.1: icmp_seq=2 ttl=64 time=0.171 ms
64 bytes from 192.168.1.1: icmp_seq=3 ttl=64 time=0.115 ms
^C
--- 192.168.1.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2652ms
rtt min/avg/max/mdev = 0.115/0.651/1.669/0.720 ms
[root@localhost den]#

```

Рис. 27.12. Пингуем адрес 192.168.1.1: сразу после настройки сети

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, системе нужно получить IP-адрес удаленного узла. Во-вторых, увидите, что маршрутизация нормально работает, и у вас есть доступ к Интернету. Если же пропинговать удаленный узел не удалось, вот наиболее вероятные причины сбоя:

- ❖ вы ошиблись при указании сетевых параметров — проверьте их;
- ❖ вы указали неправильный IP-адрес или имя компьютера — проверьте его;
- ❖ удаленный компьютер просто выключен или временно недоступен, например из-за сбоя интернет-канала, по которому удаленный компьютер подключается к Всемирной сети (такое бывает чаще, чем можно предположить);
- ❖ в вашей сети не настроен или не работает шлюз.

ПОЯСНЕНИЕ

Дело в том, что когда пакет адресуется компьютеру, находящемуся за пределами локальной сети, он посылается на шлюз, а уже потом шлюз передает его удаленному компьютеру. Если сеть настраивали не вы, вполне вероятно, что шлюз уже настроен администратором сети, и вы сразу получите доступ к Интернету — вам нужно лишь правильно указать параметры сети. А вот если вы сам себе администратор, то вам нужно настроить брандмауэр на шлюзе (компьютере, подключенном к Интернету) так, чтобы он предоставлял другим компьютерам локальной сети доступ к Интернету. О настройке общего доступа к Интернету мы поговорим в *главе 37*.

Других причин недоступности удаленного компьютера не должно быть, если исключить неисправность сетевого оборудования.

Изменить параметры сетевого интерфейса можно с помощью конфигулятора `drakconf`, для запуска которого нажмите комбинацию клавиш `<Alt>+<F2>` и введите команду:

```
drakconf
```

После этого перейдите в раздел **Сеть и Интернет** и выберите конфигуратор для изменения параметров сетевого интерфейса (**Настройка сетевого интерфейса**). Там же вы найдете и конфигуратор для удаления сетевых интерфейсов. Как видите, все просто.

27.3.2. Настройка сети в Fedora

Последовательность действий по настройке сети в Fedora такая же, как и в Linux Mandriva, только используются другие конфигураторы. Первым делом командой `/sbin/ifconfig` убедитесь, что подключен сетевой кабель и активен интерфейс `lo`. С другой стороны, не припомню, чтобы на работающей Linux-машине интерфейс `lo` не был активен.

Честно говоря, не помню, в какой версии Fedora появился NetworkManager — диспетчер сети (кажется, в 9-й). Поначалу эта программа глючила так, что многие администраторы попросту отказывались от нее и настраивали сеть вручную с помощью конфигулятора `system-config-network`. Позже конфигуратор NetworkManager (хотя это не обычный конфигуратор в прямом смысле слова: это системная служба в сочетании с графическим интерфейсом настройки сети) появился в других дистрибутивах, в частности в Ubuntu.

Сейчас вроде бы NetworkManager работает вполне достойно, но ради экономии количества страниц в книге (значит, и ваших денег!) мы рассмотрим конфигуратор `system-config-network`, которым также можно настроить сеть в Fedora (и не только в последних версиях, но и в самых ранних). Мы также рассмотрим, как отключить NetworkManager в Fedora, если у вас с ним возникнут проблемы. А вот сам NetworkManager мы рассмотрим на примере дистрибутива Ubuntu/Denix (*см. разд. 27.3.3*) — он там такой же, как и в Fedora. Для предельной точности отмечу, что в книге рассматриваются последние версии дистрибутивов — Fedora 13 и Ubuntu 10.04. Я специально отмечаю номера версий, чтобы не получать письма от

читателей примерно такого содержания: "А почему в таком-то окне такая-то кнопка называется так, а на рисунке — иначе?" А все потому, что по непонятным мне причинам разработчики программ для Linux частенько любят менять названия всевозможных кнопок, хотя внешний вид окна и действия кнопки остаются теми же.

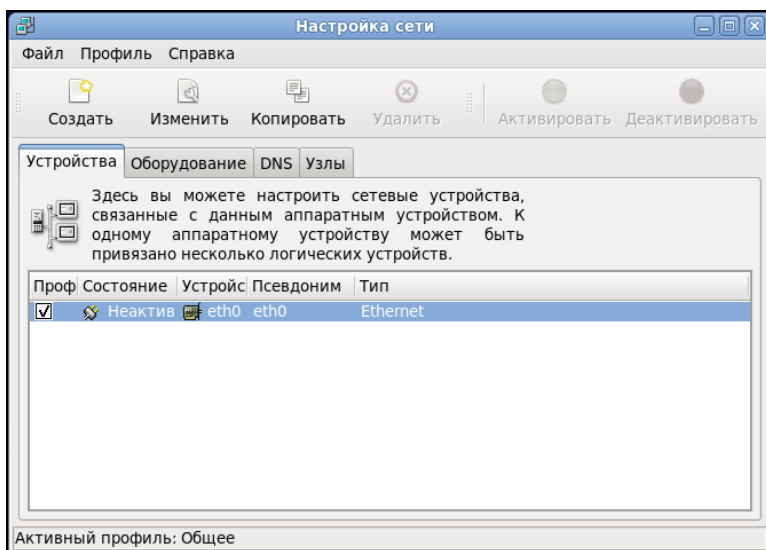


Рис. 27.13. Окно Настройка сети

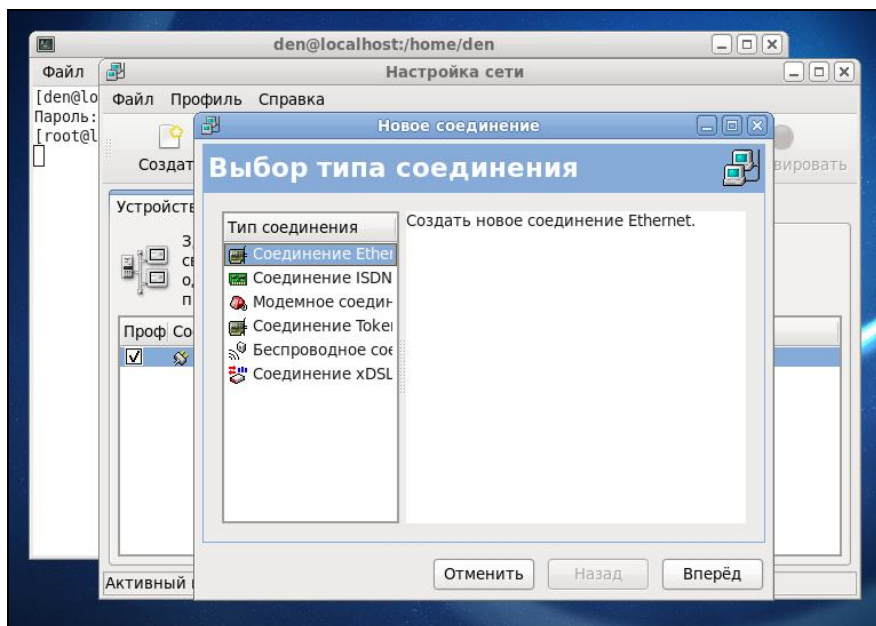


Рис. 27.14. Создание Ethernet-соединения

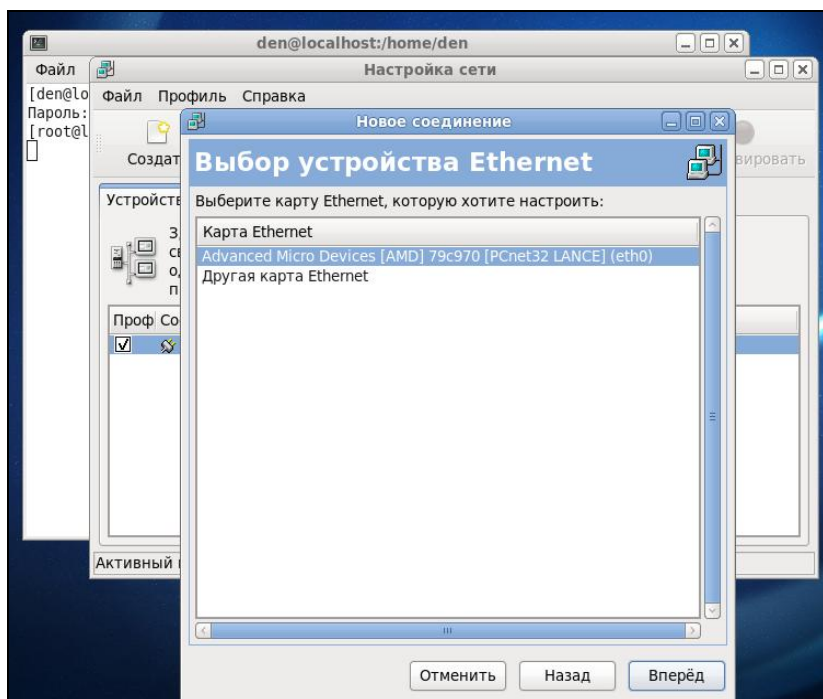


Рис. 27.15. Выбор сетевой платы

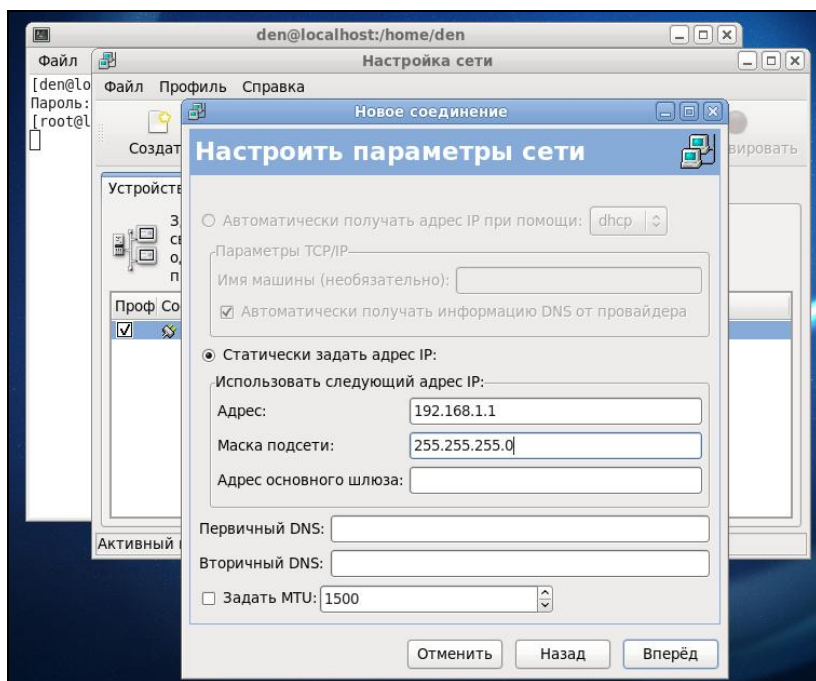


Рис. 27.16. Ввод параметров сети

И еще одно: конфигурактор `system-config-network` используется только для задания статического IP-адреса. Впрочем, Fedora, как и любой другой дистрибутив, отлично дружит с DHCP, поэтому в случае его наличия вообще не придется запускать какой-либо конфигурактор для настройки сети.

Итак, приступим к рассмотрению `system-config-network`. После этого введите команду:

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

Откроется окно конфигулятора сети (рис. 27.13). Если соединение по локальной сети уже у вас создано (что происходит при загрузке), выделите его и нажмите кнопку **Изменить**. После чего установите параметры сети.

Если же соединений в окне конфигулятора сети нет, нажмите кнопку **Создать**, а затем выберите **Соединение Ethernet** и нажмите кнопку **Вперёд** (рис. 27.14).

Следующий шаг — это выбор сетевой платы (рис. 27.15). Выделите сетевую плату, через которую осуществляется настраиваемое соединение с сетью. Если у вас всего одна сетевая плата, просто подтвердите выбор.

Теперь введите параметры сети: IP-адрес, маску сети и IP-адрес шлюза по умолчанию (рис. 27.16).

На этом настройка сетевого интерфейса завершена. Проверьте введенные вами данные и, если все правильно, нажмите кнопку **Вперёд** — откроется основное окно **Настройка сети** конфигулятора сети `system-config-network`, в котором будет отображен только что созданный вами интерфейс.

Сразу после настройки сетевой интерфейс неактивен. Нажмите кнопку **Активировать** для его активации.

ПРИМЕЧАНИЕ

Кнопки **Активировать** и **Деактивировать** станут активными только, если сервис `NetworkManager` отключен! Ведь именно он по умолчанию управляет сетевыми настройками. А после его отключения "бразды правления" передаются конфигуратору `system-config-network`. Спрашивается, а почему я тогда вообще рассматриваю `system-config-network`, если он устарел? Тому есть три причины. Первая — если вам нужно задать статический адрес. Вторая — если у вас старый дистрибутив Fedora. Третья — если у вас в сети есть DHCP-сервер и вы хотите использовать `NetworkManager`, то вам вообще ничего не нужно делать. Только подключите сетевой кабель — и все.

Изменить параметры интерфейса можно, нажав кнопку **Изменить**. В открывшемся окне вы сможете переназначить различные параметры сети, в том числе выбрать использование протокола DHCP для автоматического конфигурирования интерфейса.

Конфигуратор `dracconf` позволяет установить параметры DNS сразу при конфигурировании каждого сетевого интерфейса. С одной стороны — это удобно. С другой — несколько неправильно, потому что установки DNS общие для всех интерфейсов. Если вы зададите одни параметры DNS при настройке одного интерфейса и совершенно другие параметры DNS при настройке другого интерфейса, последние указанные параметры перезапишут параметры, заданные ранее. Разработчики Fedora поступили правильно — они вынесли параметры DNS на отдельную страничку конфигулятора (рис. 27.17). Теперь ясно, что параметры одни для всех, а не разные для каждого интерфейса, как можно было подумать в Linux Mandriva.

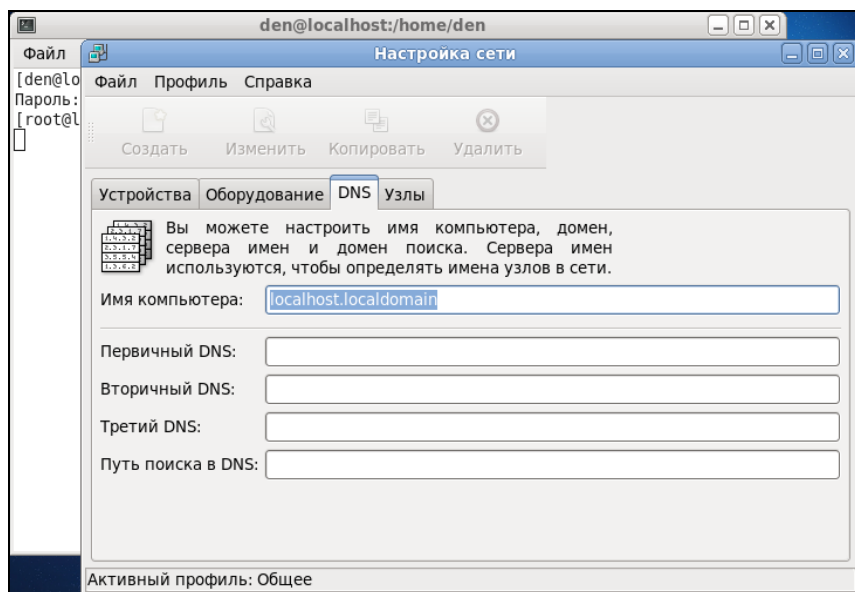


Рис. 27.17. Редактировать параметры DNS

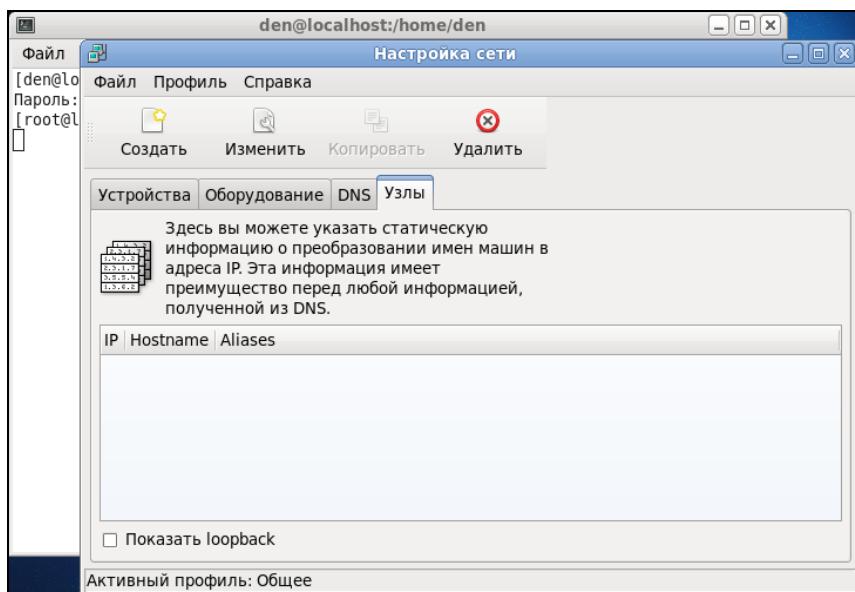


Рис. 27.18. Редактирование файла /etc/hosts

На вкладке **DNS** (см. рис. 27.17) 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. 27.18) 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, поскольку IP-адреса могут периодически меняться.

Для добавления записи в файл `/etc/hosts` нажмите в окне **Настройка сети** (см. рис. 27.18) кнопку **Создать**. Откроется небольшое окно, в котором нужно будет ввести IP-адрес узла, его доменное имя и псевдоним (обычно — сокращенное имя). Например, если имя узла **den.mycompany.com.ru**, то сокращенное имя можно установить типа `den`.

Настало время проверить работу сетевого интерфейса. Для этого сначала введем команду `ifconfig`, чтобы убедиться, что сетевой интерфейс активен, а затем пропиnguем сетевой интерфейс по его адресу, который вы узнаете из вывода `ifconfig` — хотя и так должны его помнить, ведь вы только что настраивали сеть!

Как уже отмечалось, в Fedora 12 и 13 сервис NetworkManager работает без особых нареканий. А вот в Fedora 9 и 10 мой сетевой интерфейс отказывался подниматься до тех пор, пока я не отключил NetworkManager и не вернулся к старому доброму сервису `network` (кстати, в Mandriva 2010.1 сервис NetworkManager отсутствует, а до сих пор используется по умолчанию сервис `network` — наверное, не зря). Отключить NetworkManager и включить сервис `network` можно следующими командами (возможно, они вам пригодятся):

```
# /etc/init.d/NetworkManager stop
# /sbin/chkconfig --level 235 NetworkManager off
# /etc/init.d/network start
Bringing up loopback interface:           [ OK ]
Bringing up interface eth0:               [ OK ]
Bringing up interface isp:                [ OK ]
# /sbin/chkconfig --level 235 network on
```

Приведенные команды:

- ❖ останавливают сервис NetworkManager;
- ❖ отключают NetworkManager на уровнях запуска 2, 3 и 5;
- ❖ запускают сервис `network`;
- ❖ включают сервис `network` на уровнях запуска 2, 3 и 5.

27.3.3. Настройка сети в Debian, Ubuntu и Denix

Конфигураторы nm-connection-editor (NetworkManager) и network-admin

В старых версиях Ubuntu (кажется, до версии 27.10) и Debian для настройки используется конфигуратор `network-admin`, запустить который можно так:

```
sudo network-admin
```

Хотя, в этом случае я предпочитаю редактировать файл `/etc/network/interfaces` вручную (чуть позже я приведу ссылку на свою статью с подробным описанием этого файла).

В новых версиях Ubuntu и Denix (это мой собственный дистрибутив на базе Ubuntu; а книга — лучшее средство "пропиарить" свое "детище") используется конфигуратор `nm-connection-editor` (NetworkManager Connection Editor, редактор соединений NetworkManager), запустить который можно или командой меню **Система | Параметры | Сетевые соединения**, или командой:

```
sudo nm-connection-editor
```

Да, вы правильно догадались: данный конфигуратор является графическим интерфейсом для сервиса NetworkManager. Точно такой же конфигуратор используется в Fedora.

Конфигуратор `nm-connection-editor` позволяет настроить Ethernet-соединения, беспроводные соединения (Wi-Fi), мобильные широкополосные соединения (GPRS/EDGE/3G), VPN (виртуальную частную сеть) и DSL-соединения. Нужно отметить, что этот конфигуратор намного лучше старого `network-admin`.

ВНИМАНИЕ!

Если у вас используется DHCP-сервер, то вообще ничего не нужно настраивать — все будет настроено автоматически (в том числе и для Wi-Fi-соединения). В крайнем случае для Wi-Fi придется ввести пароль доступа, если, конечно, система корректно распознала ваш Wi-Fi-адаптер.

После запуска конфигулятора (рис. 27.19) вы увидите список созданных сетевых интерфейсов. Если нужно установить какие-то определенные параметры интерфейса, выделите интерфейс и нажмите кнопку **Изменить**.

Рассмотрим окно изменения параметров сетевого интерфейса (рис. 27.20):

- ♦ на вкладке **Проводные** можно просмотреть (и даже изменить — в случае необходимости) MAC-адрес сетевого интерфейса и изменить MTU (Maximum Transfer Unit);

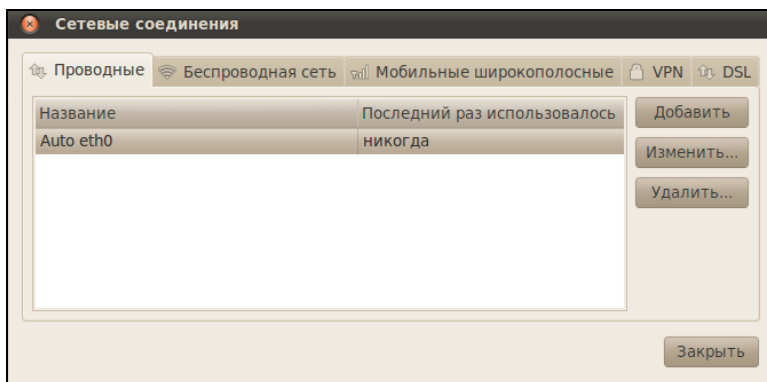


Рис. 27.19. Конфигуратор `nm-connection-editor`

- ❖ вкладка **Защита 802.1x** используется для задания специальных опций защиты интерфейса (используется редко);
- ❖ на вкладке **Параметры IPv4** можно изменить сетевые параметры, относящиеся к протоколу IPv4. Чтобы задать статический IP-адрес, выберите метод **Вручную**, затем нажмите кнопку **Добавить** и добавьте IP-адрес;
- ❖ Ubuntu поддерживает концепцию VLAN, позволяющую одному сетевому интерфейсу присвоить несколько IP-адресов. Если вы хотите использовать DHCP, но хотите указать свои DNS-серверы, то выберите метод **Автоматически (DHCP, только адрес)**;
- ❖ на вкладке **Параметры IPv6** можно указать параметры, относящиеся к протоколу IPv6, если вы таковой используете.

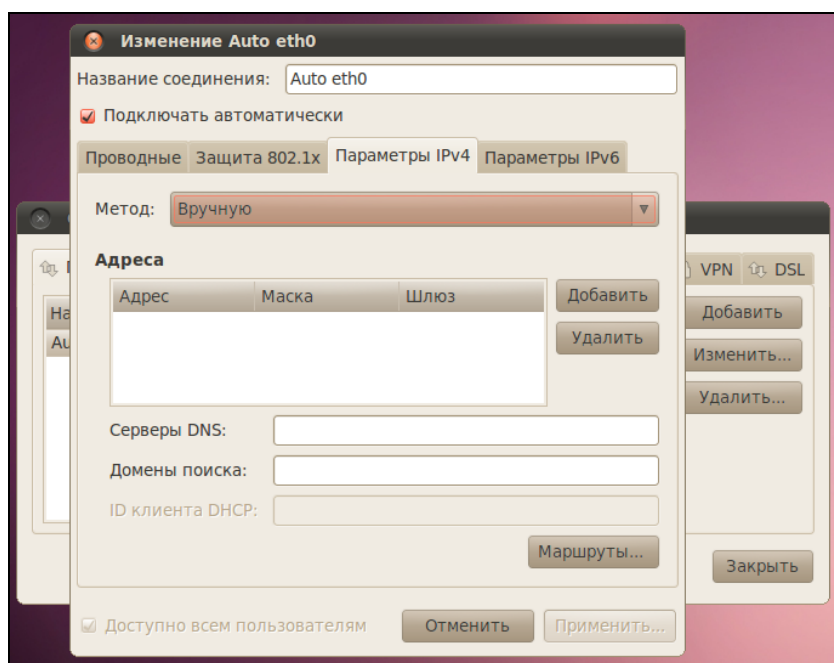


Рис. 27.20. Изменение параметров сетевого интерфейса

Если в вашей сети нет DHCP-сервера, выполняющего автоматическую настройку рабочих станций, тогда перейдите на вкладку **Параметры IPv4** и выберите конфигурацию **Вручную**. После этого введите свой IP-адрес, маску сети и IP-адрес шлюза (gateway). Всю эту информацию вы сможете узнать у администратора сети.

Как уже отмечалось, Ubuntu поддерживает технологию VLAN (Virtual LAN), что позволяет одному сетевому адаптеру назначить несколько IP-адресов. На практике данная возможность используется редко, но вы должны знать, что поддержка VLAN в Ubuntu есть.

Дополнительную информацию о VLAN можно получить в моих статьях:

<http://www.xakep.ru/magazine/xa/121/122/1.asp>

<http://www.dkws.org.ua/index.php?page=show&file=a/ubuntu/network-interfaces>

В заключение этого раздела приведу несколько полезных ссылок:

- ◆ <http://dkws.org.ua/index.php?page=show&file=a/ubuntu/static-ip-ubuntu9> — если у вас возникнут проблемы с установкой статического IP-адреса;
- ◆ <http://dkws.org.ua/index.php?page=show&file=a/ubuntu/static-dns-ubuntu9> — как установить вручную IP-адрес DNS-сервера в Ubuntu 9.04;
- ◆ <http://dkws.org.ua/index.php?page=show&file=a/ubuntu/network-interfaces> — если вы решите отказаться от NetworkManager и использовать старый сервис network, настоятельно рекомендую ознакомиться с форматом файла конфигурации /etc/network/interfaces (кстати, этот файл конфигурации используется для задания сетевых параметров в Debian);
- ◆ <http://dkws.org.ua/index.php?page=show&file=a/ubuntu/vpn-ubuntu9> — как настроить VPN-соединение в Ubuntu (хотя это и не относится к настройке локальной сети, но, думаю, вам пригодится).

27.3.4. Конфигуратор netconfig в Slackware

Конфигуратор netconfig в Slackware можно запускать даже в консоли (рис. 27.21). Он поочередно задаст вам ряд вопросов — от имени компьютера до IP-адреса шлюза. По сути, его работа ничем не отличается от работы прочих рассмотренных здесь конфигураторов, просто у него несколько своеобразный интерфейс пользователя.



Рис. 27.21. Конфигуратор netconfig

27.4. Проблемы с ноутбуком Acer eMachines E525

На ноутбуке Acer E525 сетевой адаптер не определяется, соответственно, к Интернету вы не подключитесь.

Чтобы исправить данную проблему, нужно установить дополнительный драйвер, скачать который можно по адресу:

<http://partner.atheros.com/Drivers.aspx>

Для установки драйвера нужно ввести в терминале следующие команды:

```
tar -xvzf AR81Family-linux-v1.0.0.10.tar.gz
cd src
make
sudo make install
sudo modprobe at11e
```

27.5. Утилиты для диагностики соединения

Причины отказа сети могут быть физические или программные. Физические связаны с неработающим сетевым оборудованием или повреждением среды передачи данных. Программные связаны с неправильной настройкой сетевого интерфейса. Как правило, избавиться от программных проблем помогает конфигуратор сети — вы его еще раз запускаете и настраиваете сетевые интерфейсы, только правильно. Если сомневаетесь в ваших действиях, обратитесь за помощью к более опытному коллеге.

Для диагностики работы сети мы будем использовать стандартные сетевые утилиты, которые входят в состав любого дистрибутива Linux. Предположим, что у нас не работает PPPoE/DSL-соединение. Проверить, "поднят" ли сетевой интерфейс, можно с помощью команды `ifconfig`. На рис. 27.22 изображено, что сначала я предпринял попытку установить соединение (ввел команду `sudo pon dsl-provider`), а затем вызвал `ifconfig` для того, чтобы убедиться, установлено ли соединение. В случае, если соединение не было бы установлено, интерфейса `ppp0` в списке бы не было. Интерфейс `eth0` (рис. 27.22) относится к первой сетевой плате (вторая называется `eth1`, третья — `eth2` и т. д.), а интерфейс `lo` — это интерфейс обратной петли, который используется для тестирования программного обеспечения (у вас он всегда будет "поднят").

Если же интерфейс не поднят, нам нужно просмотреть файл `/var/log/messages` сразу после попытки установки сообщения:

```
tail -n 10 /var/log/messages
```

Данная команда просматривает "хвост" файла протокола (выводит последние 10 сообщений). В случае удачной установки соединения сообщения в файле протокола будут примерно следующими:

```
Feb  6 14:28:33 user-desktop pppd[5176]: Plugin rp-pppoe.so loaded.
Feb  6 14:28:33 user-desktop kernel: [17179852.932000] CSLIP: code copyright
198 9 Regents of the University of California
Feb  6 14:28:33 user-desktop kernel: [17179852.944000] PPP generic driver ver-
sio n 2.4.2
Feb  6 14:28:33 user-desktop pppd[5183]: pppd 2.4.4b1 started by root, uid 0
```

```

Feb  6 14:28:33 user-desktop pppd[5183]: PPP session is 2838
Feb  6 14:28:33 user-desktop kernel: [17179852.984000] NET: Registered proto-
col family 24
Feb  6 14:28:33 user-desktop pppd[5183]: Using interface ppp0
Feb  6 14:28:33 user-desktop pppd[5183]: Connect: ppp0 <--> eth0
Feb  6 14:28:33 user-desktop pppd[5183]: Remote message: Login ok
Feb  6 14:28:33 user-desktop pppd[5183]: PAP authentication succeeded
Feb  6 14:28:33 user-desktop pppd[5183]: peer from calling number
00:15:F2:60:28 :97 authorized
Feb  6 14:28:33 user-desktop pppd[5183]: local  IP address 193.254.218.243
Feb  6 14:28:33 user-desktop pppd[5183]: remote IP address 193.254.218.129
Feb  6 14:28:33 user-desktop pppd[5183]: primary  DNS address 193.254.218.1
Feb  6 14:28:33 user-desktop pppd[5183]: secondary DNS address 193.254.218.27

```

```

user@user-desktop:~$ sudo pon dsl-provider
Password:
Plugin rp-pppoe.so loaded.
user@user-desktop:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0D:87:88:BC:96
          inet6 addr: fe80::20d:87ff:fe88:bc96/64 Диапазон:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:629 errors:0 dropped:0 overruns:0 frame:0
          TX packets:121 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:104484 (102.0 KiB)  TX bytes:11682 (11.4 KiB)
          Interrupt:11 Base address:0xe800

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Диапазон:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:25 errors:0 dropped:0 overruns:0 frame:0
          TX packets:25 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:1744 (1.7 KiB)  TX bytes:1744 (1.7 KiB)

ppp0      Link encap:Point-to-Point Protocol
          inet addr:193.254.218.243  P-t-P:193.254.218.129  Mask:255.255.255.255
          UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1488  Metric:1
          RX packets:107 errors:0 dropped:0 overruns:0 frame:0
          TX packets:95 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:3
          RX bytes:32174 (31.4 KiB)  TX bytes:6001 (5.8 KiB)

user@user-desktop:~$

```

Рис. 27.22. Программа ifconfig

Первая строчка — сообщение о том, что загружен модуль поддержки PPPoE. Следующие два сообщения информируют нас о поддержке нашим компьютером протоколов CSLIP и PPP. После сообщается, что демон `pppd` запущен, указывается, от чьего имени он запущен (`root`) и версия самого `pppd`. Далее сообщается имя используемого интерфейса (`ppp0`) и имя вспомогательного интерфейса (помните, что протокол PPPoE подразумевает передачу кадров PPP по Ethernet) — `eth0`.

Следующие два сообщения свидетельствуют об удачной регистрации:

```
Feb  6 14:28:33 user-desktop pppd[5183]: Remote message: Login ok
Feb  6 14:28:33 user-desktop pppd[5183]: PAP authentication succeeded
```

Затем система сообщает нам наш IP-адрес, адрес удаленного компьютера, который произвел аутентификацию, а также IP-адреса серверов DNS.

А вот пример неудачной попытки соединения:

```
Feb  6 09:23:48 user-desktop pppd[6667]: PPP session is 2336
Feb  6 09:23:48 user-desktop pppd[6667]: Using interface ppp1
Feb  6 09:23:48 user-desktop pppd[6667]: Connect: ppp1 <--> eth0
Feb  6 09:23:48 user-desktop pppd[6667]: Remote message: Login incorrect
Feb  6 09:23:48 user-desktop pppd[6667]: Connection terminated.
```

Причина неудачи понятна: имя пользователя или пароль неправильные, о чем красноречиво свидетельствует сообщение "Login incorrect". Для того чтобы изменить имя пользователя или пароль, запустите конфигуратор `pppoeconf`. Но не спешите этого делать: если в предыдущий раз соединение было установлено (а настройки соединения вы не изменяли), возможно, нужно обратиться к провайдеру — это явный признак неправильной работы оборудования на стороне провайдера.

Вот еще один пример, характерный для PPPoE:

```
Feb  6 09:23:48 user-desktop pppd[6667]: PPP session is 2336
Feb  6 09:23:48 user-desktop pppd[6667]: Using interface ppp1
Feb  6 09:23:48 user-desktop pppd[6667]: Connect: ppp1 <--> eth0
Feb  6 09:23:48 user-desktop pppd[6667]: Connection terminated.
```

Это явный пример неправильной работы оборудования провайдера. Возможно, нужно перезагрузить точку доступа (access point), т. е. просто выключите и включите ее. Если это не помогает, тогда обращайтесь к провайдеру.

Наиболее простая ситуация, когда сеть вообще не работает. В этом случае очень легко обнаружить причину неисправности. Если работает устройство, значит, повреждена среда передачи данных (сетевой кабель). В случае с модемной линией нужно проверить, нет ли ее обрыва. В случае с витой парой обрыв маловероятен (хотя возможен), поэтому нужно проверить, правильно ли обжат кабель (возможно, нужно обжать витую пару заново).

Намного сложнее ситуация, когда сеть то работает, то нет. Например, вы не можете получить доступ к какому-нибудь узлу, хотя пять минут назад все работало отлично. Если исключить неправильную работу удаленного узла, к которому вы подключаетесь, следует поискать решение в маршруте, по которому пакеты добываются от вашего компьютера до удаленного узла. Сначала пропингуем удаленный узел. Для этого используется команда `ping` (прервать выполнение команды `ping` можно с помощью нажатия комбинации клавиш `<Ctrl>+<C>`):

```
ping dkws.org.ua
```

```
PING dkws.org.ua (213.186.114.75) 56(84) bytes of data.
```

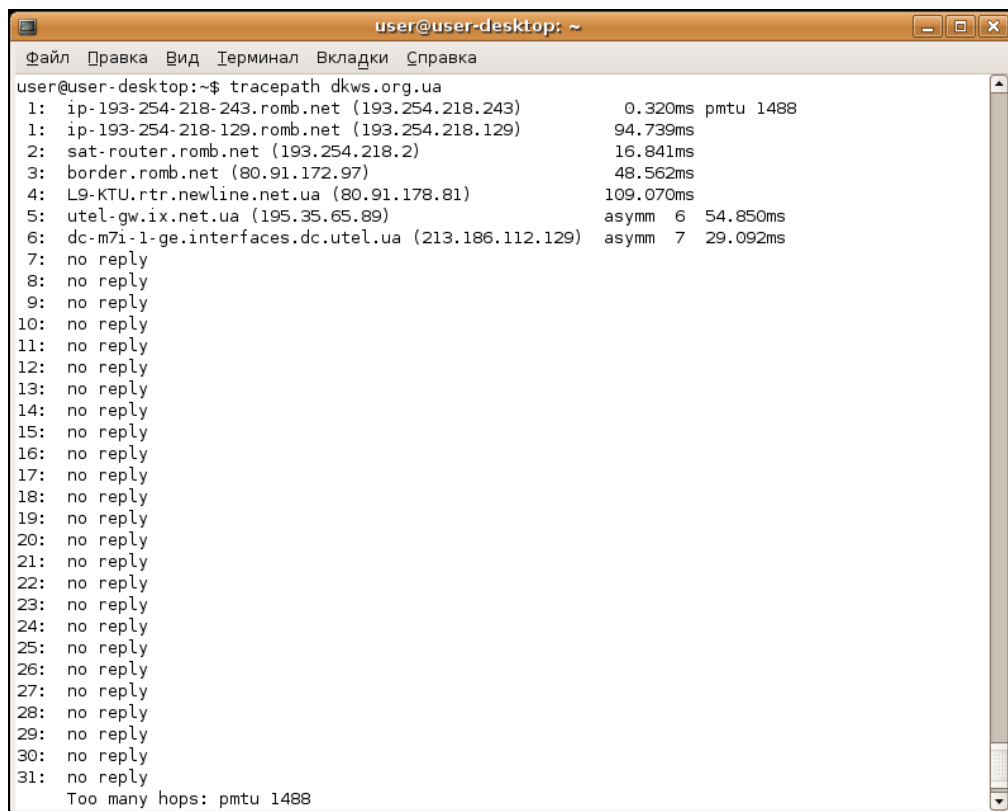
```
64 bytes from wdt.org.ru (213.186.114.75): icmp_seq=1 ttl=58 time=30.7 ms
```

```
64 bytes from wdt.org.ru (213.186.114.75): icmp_seq=2 ttl=58 time=24.8 ms
64 bytes from wdt.org.ru (213.186.114.75): icmp_seq=5 ttl=58 time=12.2 ms
64 bytes from wdt.org.ru (213.186.114.75): icmp_seq=6 ttl=58 time=159 ms
64 bytes from wdt.org.ru (213.186.114.75): icmp_seq=7 ttl=58 time=19.3 ms
64 bytes from wdt.org.ru (213.186.114.75): icmp_seq=9 ttl=58 time=29.0 ms
...
```

В этом случае все нормально. Но иногда ответы от удаленного сервера то приходят, то не приходят. Чтобы узнать, в чем причина (где именно теряются пакеты), нужно выполнить трассировку узла:

```
tracertpath dkws.org.ua
```

В других дистрибутивах вместо команды `tracertpath` используется команда `traceroute`, а в Windows — `tracert`. На рис. 27.23 изображено выполнение команды `tracertpath`. Сразу видно, что есть определенные проблемы с прохождением пакетов до удаленного узла.



```
user@user-desktop: ~  
Файл Правка Вид Терминал Вкладки Справка  
user@user-desktop:~$ tracertpath dkws.org.ua  
1: ip-193-254-218-243.romb.net (193.254.218.243) 0.320ms pmtu 1488  
1: ip-193-254-218-129.romb.net (193.254.218.129) 94.739ms  
2: sat-router.romb.net (193.254.218.2) 16.841ms  
3: border.romb.net (80.91.172.97) 48.562ms  
4: L9-KTU.rtr.newline.net.ua (80.91.178.81) 109.070ms  
5: utel-gw.ix.net.ua (195.35.65.89) asymm 6 54.850ms  
6: dc-m7i-1-ge.interfaces.dc.utel.ua (213.186.112.129) asymm 7 29.092ms  
7: no reply  
8: no reply  
9: no reply  
10: no reply  
11: no reply  
12: no reply  
13: no reply  
14: no reply  
15: no reply  
16: no reply  
17: no reply  
18: no reply  
19: no reply  
20: no reply  
21: no reply  
22: no reply  
23: no reply  
24: no reply  
25: no reply  
26: no reply  
27: no reply  
28: no reply  
29: no reply  
30: no reply  
31: no reply  
Too many hops: pmtu 1488
```

Рис. 27.23. Проблема с прохождением пакетов

Понятно, что по пути пакеты теряются. Для того чтобы выяснить причину, вам нужно обратиться к администратору того маршрутизатора, который не пропускает

дальше пакеты. Причина именно в нем. В данном случае, как видно из рисунка, пакеты доходят до маршрутизатора `dc-m7i-1-ge.interfaces.dc.utel.ua`, а после него движение пакетов прекращается.

Если соединение установлено (о чем свидетельствует наличие поднятого интерфейса в выводе `ifconfig`), а Web-страницы не открываются, попробуйте пропинговать любой удаленный узел по IP-адресу. Если не знаете, какой узел пинговать (т. е. не помните ни одно IP-адреса), пропингуйте узел `213.186.114.75`. Если вы получите ответ, а странички по-прежнему не открываются, когда вы вводите символическое имя, значит, у вас проблемы с DNS: сервер провайдера почему-то не передал вашему компьютеру IP-адреса DNS-серверов. Позвоните провайдеру, выясните причину этого, а еще лучше уточните IP-адреса серверов DNS и укажите их в файле `/etc/resolv.conf`. Формат этого файла прост:

```
nameserver IP-адрес
```

Например:

```
nameserver 193.254.218.1
```

```
nameserver 193.254.218.27
```

Всего можно указать до четырех серверов DNS.

Если же не открывается какая-то конкретная страничка, а все остальные работают нормально, тогда, понятно, что причина в самом удаленном сервере, а не в ваших настройках.

27.6. Для фанатов, или как настроить сеть вручную

Иногда мои книги критикуют за то, что при настройке сети я использую только графические конфигураторы. С одной стороны, конфигураторы просты и удобны. Ведь в Windows вы пользуетесь Панелью управления, а не редактором реестра, хотя можно изменять сетевые настройки и через `regedit`. С другой стороны, редактирование конфигурационных файлов позволяет глубже познать Linux. Если вам интересно, в какие файлы сохраняются сетевые настройки после нажатия кнопки **ОК** в окне конфигуратора, тогда данный раздел — для вас. А если вы не считаете, что на это нужно тратить свое время (ведь за считанные секунды можно все настроить конфигуратором), тогда можете смело приступать к чтению следующей главы. Хотя, я вовсе не исключаю и такого развития ситуации: вы с интересом прочитаете этот раздел, но в будущем будете использовать конфигураторы, потому что это просто.

Таблица 27.2 — сводная, когда вы будете знать что и к чему, вы, используя ее, быстро вспомните, какой вам нужно редактировать конфигурационный файл. Далее мы поговорим о конфигурационных файлах конкретных дистрибутивов.

27.6.1. Конфигурационные файлы Fedora

Мне не нравится дистрибутив Fedora. Но не принимать его во внимание я не могу, поскольку это классика дистрибутивостроения. Это все равно, что говорить об автомобилестроении и забыть о марке "Форд" — это тоже классика. Но в последнее время наблюдается не очень хорошая тенденция: все классические марки портятся. Раньше я с удовольствием работал в Red Hat и восхищался "Фордами". Но мне не нравятся ни современные "Форды", ни современная реализация Red Hat — Fedora.

Но не считаться с Fedora я не могу, поэтому приступим к рассмотрению конфигурационных файлов этого дистрибутива. Начнем с файла `/etc/sysconfig/network`. В этом файле можно задать имя машины, шлюз по умолчанию и включить IP-переадресацию. Пример этого файла приведен в листинге 27.1.

Листинг 27.1. Файл `/etc/sysconfig/network`

```
NETWORKING=yes
NETWORKING_IPV6=no
HOSTNAME=den.dkws.org.ua
# Дополнительно
DHCP_HOSTNAME=den.dkws.org.ua
GATEWAY=192.168.0.1
GATEWAYDEV=eth0
FORWARD_IPV4=no
```

В большинстве случаев хватает первых трех параметров. Первый параметр определяет, будет ли включена поддержка сети. Обычно нужно включить поддержку сети (`yes`), т. к. даже функции печати в Linux требуют поддержки сети.

Второй параметр включает поддержку IPv6. Поскольку этот протокол еще не используется, то нужно задать значение `no`.

Третий параметр задает имя узла. Параметр `DHCP_HOSTNAME` задает имя узла при использовании DHCP. Если вы не задали значение параметра `DHCP_HOSTNAME`, то DHCP-сервер может назначить узлу другое имя. Если же значение задано, то DHCP не будет изменять имя узла.

Параметр `GATEWAY` задает шлюз по умолчанию. В этом конфигурационном файле указывать шлюз по умолчанию необязательно, поскольку его можно указать в файле `/etc/sysconfig/network-scripts/ifcfg-eth0` — конфигурационный файл сетевого интерфейса `eth0`.

Параметр `GATEWAYDEV` указывает имя интерфейса для доступа к шлюзу. Часто этот параметр опускается.

Последний параметр, `FORWARD_IPV4`, позволяет превратить ваш компьютер в шлюз.

После редактирования файла `/etc/sysconfig/network` нужно перейти в каталог `/etc/sysconfig/network-scripts/`, в котором содержатся конфигурационные файлы для

каждого сетевого интерфейса. Например, конфигурация интерфейса `eth0` содержится в файле `/etc/sysconfig/network-scripts/ifcfg-eth0`. Конфигурация интерфейса может отличаться в зависимости от того, как настраивается интерфейс — автоматически по DHCP или же сетевая информация присваивается статически. Как правило, на рабочих станциях сетевая информация присваивается автоматически — по DHCP. А вот на серверах (в том числе и на DHCP-сервере) сетевая информация указывается статически — вручную.

В листинге 27.2 приведена конфигурация интерфейса, настраиваемого по DHCP.

Листинг 27.2. Конфигурация интерфейса, настраиваемого по DHCP

```
DEVICE=eth0
BOOTPROTO=dhcp
HWADDR=XX:XX:XX:XX:XX:XX
ONBOOT=yes
TYPE=Ethernet
IPV6INIT=no
```

Первый параметр задает имя устройства (`eth0`), второй — тип конфигурации — по протоколу DHCP. Третий параметр позволяет изменить аппаратный MAC-адрес сетевого адаптера. Как правило, этот параметр указывается только тогда, когда нужно изменить MAC-адрес. В обычных условиях он не нужен.

Параметр `ONBOOT` определяет, будет ли "поднят" интерфейс при загрузке (`yes` — да, `no` — нет). Последние два параметра необязательны (первый — задает тип интерфейса, второй — включает IPv6 для интерфейса).

Пример статической настройки интерфейса приведен в листинге 27.3.

Листинг 27.3. Статическая настройка интерфейса

```
DEVICE=eth0
BOOTPROTO=none
HWADDR=XX:XX:XX:XX:XX:XX
ONBOOT=yes
#
NETMASK=255.255.255.0
IPADDR=192.168.0.10
GATEWAY=192.168.0.1
#
NETWORK=192.168.0.0
BROADCAST=192.168.0.255
USERCTL=no
```


Первые четыре параметра нам знакомы. Разница лишь в том, что параметр `BOOTPROTO` содержит значение `none` вместо `dhcp`. Параметр `NETMASK` задает сетевую маску, параметр `IPADDR` — IP-адрес узла, `GATEWAY` — шлюз по умолчанию для данного сетевого интерфейса.

Также можно задать необязательные параметры `NETWORK` (адрес сети), `BROADCAST` (широковещательный IP-адрес) и `USERCTL`. Если последний параметр включен (`yes`), то интерфейсом могут управлять не-root-пользователи. Обычно в этом нет необходимости, поэтому присваивается значение `no`.

С остальными файлами вы знакомы из табл. 27.2:

- ❖ `/etc/resolv.conf` — конфигурация DNS (здесь указываются DNS-серверы);
- ❖ `/etc/hosts` — статическая таблица поиска имен узлов, применяется если ваша сеть не использует DNS;
- ❖ `/etc/sysconfig/static-routes` — данный файл отсутствует по умолчанию, содержит список статических маршрутов, подробно описан в главе 37.

27.6.2. Конфигурационные файлы openSUSE

В openSUSE все конфигурационные файлы, относящиеся к настройкам сети, находятся в каталоге `/etc/sysconfig/network`:

- ❖ `/etc/sysconfig/network/ifcfg-имя` — содержит параметры сетевого интерфейса (здесь *имя* — это имя сетевого интерфейса);
- ❖ `/etc/sysconfig/network/ifroute-имя` — содержит маршруты для конкретного интерфейса;
- ❖ `/etc/sysconfig/network/routes` — список статических маршрутов (см. главу 37);
- ❖ `/etc/sysconfig/network/config` — различные переменные.

Основные файлы — это файлы `/etc/sysconfig/network/ifcfg-имя`. Рассмотрим пример файла `/etc/sysconfig/network/ifcfg-eth0`, задающего параметры сетевого интерфейса `eth0` (листинг 27.4).

Листинг 27.4. Файл `/etc/sysconfig/network/ifcfg-eth0`

```
BOOTPROTO='dhcp'
IPADDR=''
MTU=''
NAME='79c970 [PCnet32 LANCE]'
NETMASK=''
NETWORK=''
STARTMODE='auto'
USERCONTROL='no'
```

В файле конфигурации сетевого интерфейса может быть множество самых разных параметров. Все возможные параметры с пояснениями и допустимыми значе-

ниями описаны в файле `ifcfg.template`. Сейчас мы обсудим только параметры, приведенные в листинге 27.4.

Параметр `BOOTPROTO` задает протокол конфигурации интерфейса. Для автоматического назначения IP-адреса по DHCP используется значение `dhcp`. Если нужно назначить адрес вручную, то указывается значение `static`. Есть еще два полезных значения:

- ❖ `autoip` — производится поиск свободного IP-адреса, найденный IP-адрес назначается статически;
- ❖ `dhcp+autoip` — основной способ — DHCP, но если DHCP-сервер отсутствует, то работает вариант `autoip`.

Назначение остальных параметров ясно — это IP-адрес, размер MTU (Maximum Transmission Unit, максимальный блок передачи), описание устройства (ни на что не влияет), сетевая маска, адрес сети.

Параметр `STARTMODE` задает режим запуска интерфейса:

- ❖ `auto` — автоматический запуск при загрузке системы;
- ❖ `manual` — интерфейс будет подниматься вручную;
- ❖ `off` — интерфейс не используется.

Есть и другие режимы запуска — о них вы прочитаете в файле `ifcfg.template`. Последний параметр запрещает управление интерфейсом не-root-пользователям.

Еще следует упомянуть полезную опцию: `DHCLIENT_SET_HOSTNAME`. Данная опция определяет, будет ли DHCP-клиент изменять имя узла, что полезно, если не нужно изменять имя узла каждый раз при получении нового IP-адреса (значение `no`).

Также можно установить значение `no` для опции `DHCLIENT_SET_HOSTNAME` в файле `/etc/sysconfig/network/dhcp`. Разница заключается в том, что в первом случае вы изменяете параметр `DHCLIENT_SET_HOSTNAME` локально — только для конкретного интерфейса, а во втором случае — глобально, для всех интерфейсов.

А где же хранится имя узла? Привычного файла `/etc/hostname` я не нашел. Пришлось действовать старым проверенным способом: вызвать конфигуратор, установить имя узла, а потом смотреть, какой файл изменился. Меня ждал небольшой сюрприз. Да, файла `/etc/hostname` нет, но зато есть файл `/etc/HOSTNAME` (все буквы написаны прописными буквами) — этот файл я просто не заметил. В нем и хранятся имя узла и имя домена.

27.6.3. Конфигурационные файлы Debian/Ubuntu

Основной конфигурационный файл Debian (и Ubuntu при выключенном NetworkManager) — `/etc/network/interfaces`. В нем можно изменить все — от IP-адреса интерфейса до параметров маршрутизации. Файл `/etc/network/interfaces` подробно описан в моей статье (глава получилась и так достаточно большой):

<http://dkws.org.ua/index.php?page=show&file=a/ubuntu/network-interfaces>

Кроме файла `/etc/network/interfaces` вам еще пригодится файл `/etc/hostname`, содержащий имя узла.

Файл `/etc/resolv.conf`, как и в других дистрибутивах, содержит параметры DNS. Но этот файл перезаписывается системой при перезагрузке. Если у вас рабочая система, то такое поведение — оптимально. А вот на сервере хотелось бы больше контроля. О том, как побороть перезапись этого файла, рассказано в другой моей статье:

<http://dkws.org.ua/index.php?page=show&file=a/ubuntu/static-dns-ubuntu9>

27.7. Команда *mii-tool*

Современные сетевые адаптеры поддерживают несколько скоростей передачи данных — 10, 100 и 1000 Мбит/с, также поддерживают два режима передачи данных: полудуплексный и полнодуплексный.

Помню, настраивал PPPoE-соединение в Windows XP. Соединение отказывалось работать на скорости 100 Мбит/с — происходили постоянные обрывы через произвольный интервал времени с момента установки соединения. Пришлось "зажать" сетевой адаптер на скорости 10 Мбит/с — после этого проблема исчезла. На скорости самого соединения это никак не отразилось, поскольку оно ограничено провайдером — 5 Мбит/с.

До сих пор для меня загадка, почему все не работало по умолчанию. Возможно, дело в самом сетевом адаптере. А может, даже в коммутаторе. Ведь по умолчанию и сетевая плата, и порт коммутатора находятся в режиме автоматического согласования, когда оба устройства пытаются подобрать совместимые параметры. Как следствие — высокая потеря пакетов. Лучший способ зафиксировать скорость и режим работы сетевого адаптера и порта коммутатора.

В Windows изменение скорости и режима работы сетевого адаптера производится в окне изменения параметров сетевого адаптера. А в Linux нужно использовать команду `mii-tool`. Для изменения режима работы порта коммутатора используется Web-интерфейс коммутатора (как правило, дешевые коммутаторы не позволяют изменять свои параметры), о том, как его использовать, вы сможете прочитать в документации по коммутатору.

Для просмотра параметров сетевого интерфейса используется команда:

```
# mii-tool -v eth0
```

Вывод будет примерно такой:

```
eth0: negotiated 100baseTx-FD flow-control, link ok
product info: vendor 88:58:43, model 0 rev 0
basic mode: autonegotiation enabled
basic status: autonegotiation complete, link ok
capabilities: 100baseTx-FD 100baseTx-HD 10baseT-FD 10baseT-HD
advertising: 100baseTx-FD 100baseTx-HD 10baseT-FD 10baseT-HD flow control
link partner: 100baseTx-FD 100baseTx-HD 10baseT-FD 10baseT-HD flow control
```

Сейчас сетевой адаптер работает в режиме автоматического согласования режима (autonegotiation), текущий статус — автосогласование завершено, связь установлена. Поле `capabilities` содержит список поддерживаемых режимов, а поле `link partner` — список режимов, поддерживаемых коммутатором.

Для установки режима используется опция `-force`:

```
# mii-tool -force=режим интерфейс
```

Например:

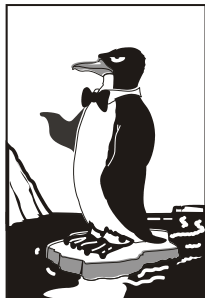
```
# mii-too -force=10baseT-FD eth0
```

27.8. Еще несколько слов о настройке сети

Напоследок отмечу, что в большинстве случаев вообще сеть настраивать не нужно — ведь DHCP-сервер сейчас не роскошь. Именно из-за этого конфигуратор сети openSUSE не рассмотрен (хотя вы без проблем разберетесь с ним, запустив Центр управления YaST), зато этот конфигуратор рассмотрен в следующей главе, где мы будем обсуждать настройку DSL-соединения — там его описание более уместно, чем здесь.

Спрашивается, а зачем была нужна эта глава, если все настраивается автоматически? А расслабляться тоже нельзя. Пользователю может и не обязательно все знать, а вот администратор знать обязан. Все пользователи сейчас немного расслабились, осознав, что Linux — это просто. Но, например, когда приходится присвоить статический IP-адрес (например, при настройке того же DHCP-сервера — у него должен быть статический адрес), они начинают "плавать". Самое интересное, что это ненадуманная проблема. Именно поэтому на главную страницу сайта я вынес ссылки на статьи, где объясняется, как присвоить статический адрес — чтобы не плодились темы на форуме. По запросу "статический ip адрес в ubuntu" Google выдает более 7000 результатов. А все из-за незнания. Надеюсь, что данная глава полностью заполнила пробел в ваших знаниях по настройке локальной сети в Linux.

ГЛАВА 28



Настройка ADSL-доступа к Интернету

DSL (Digital Subscriber Line) — цифровая абонентская линия, позволяющая производить двунаправленный обмен данными по телефонной линии. Существует несколько вариантов DSL-линий: ADSL, VDSL, SDSL, RADSL. Наиболее распространены ADSL-линии. ADSL (Asymmetric DSL) — асимметрическая цифровая линия. Для передачи данных используется витая пара телефонной сети. Скорость передачи данных зависит от расстояния, например, 1,5 Мбит/с при расстоянии в 5—6 км. Но обычно скорость ограничивается провайдером и зависит от тарифного плана. Самый доступный тарифный план подразумевает скорость передачи данных 64 Кбит/с.

28.1. Причина популярности DSL-соединений

Почему ADSL-соединения стали такими популярными? Основная причина популярности — это скорость и дешевизна. Именно эти два фактора. Даже в самом "дешевом" варианте обеспечивается скорость передачи данных 64 Кбит/с. Это в два раза быстрее, чем модем (конечно, в идеальных условиях из модема можно "выжать" 56 Кбит/с, но на практике это получается далеко не всегда). И при этом никаких разрывов соединений!

Да, за подключение к провайдеру нужно заплатить определенную сумму (напомню, что модемное подключение бесплатно), но, поверьте, оно того стоит. Также понадобится специальный ADSL-модем, который стоит дороже обычного модема, но в большинстве случаев есть возможность взять модем в аренду у провайдера, а стоимость такой аренды просто смешна.

Дешево, быстро — это все просто замечательно. Но есть и еще одно преимущество — когда вы работаете в Интернете, ваш телефон не занят, в отличие от модемного соединения.

Однако и здесь не без неожиданностей — ADSL-соединение возможно не на каждой телефонной линии. Ваша телефонная линия должна быть цифровой, иначе ничего не получится.

28.2. Физическое подключение ADSL-модема

ADSL-модем подключается к телефонной линии через специальное устройство — ADSL-сплиттер, который обычно входит в комплект поставки модема. К ADSL-сплиттеру также подключается и обычный параллельный телефон. В свою очередь ADSL-модем подключается к компьютеру с помощью Ethernet-кабеля (витой пары), также входящей в комплект поставки. Схема подключения изображена на рис. 28.1.

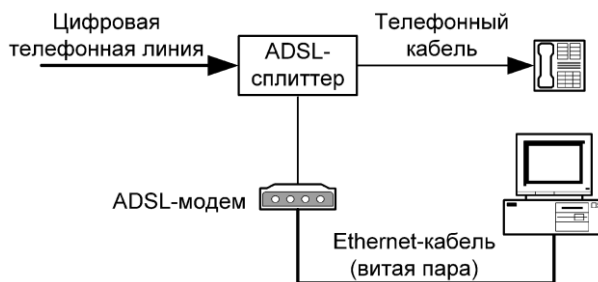


Рис. 28.1. Схема подключения ADSL-модема

ВНИМАНИЕ!

Если у вас есть дополнительные параллельные телефоны, то подключать их к телефонной линии напрямую не допускается! Подключать параллельные телефоны можно только через ADSL-сплиттер.

28.3. Настройка DSL-соединения

28.3.1. В Fedora

В Fedora соединение проще всего настроить конфигуратором system-config-network. Запустите его и нажмите кнопку **Создать**, затем выберите опцию **Соединение xDSL** (рис. 28.2).

Следующий шаг еще более важный — вам нужно выбрать устройство, которое соединено с точкой доступа, ввести имя провайдера, имя пользователя и пароль (рис. 28.3).

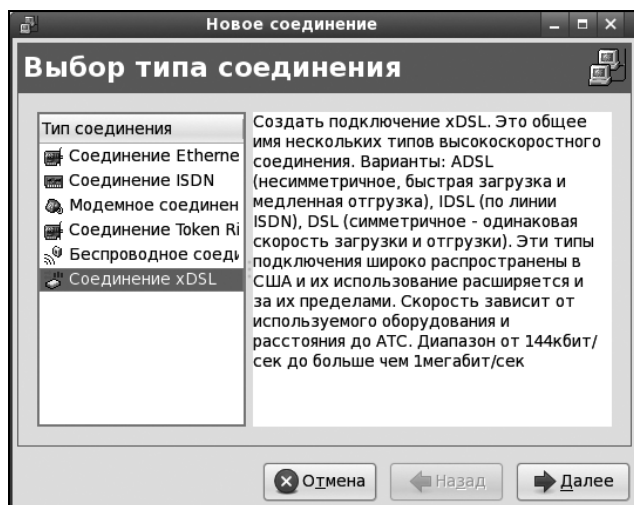


Рис. 28.2. Создание DSL-соединения в Fedora

После этого нужно нажать кнопку **Далее**, а затем кнопку **Применить**. Установить соединение можно командой `system-config-network` — выбрать ваше соединение и нажать кнопку **Активировать** (рис. 28.4). Для разрыва соединения служит кнопка **Деактивировать**.

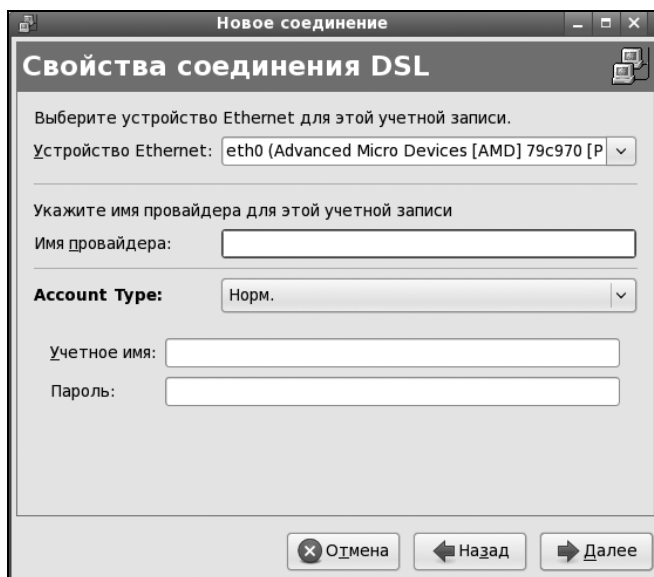


Рис. 28.3. Ввод параметров соединения

Для запуска конфигуратора `system-config-network` нужны полномочия `root`. Но обычные пользователи (у которых нет таких полномочий) тоже могут настроить

соединение с Интернетом — с помощью программы NetworkManager (**Система | Параметры | Сетевые соединения**), которая установлена в последних версиях Fedora. Некоторые администраторы предпочитают отключить NetworkManager (действительно, на сервере он не нужен). Для его отключения нужно ввести следующие команды (*подробно см. главу 27*):

```
# service NetworkManager stop
# chkconfig --level 2345 NetworkManager off
```

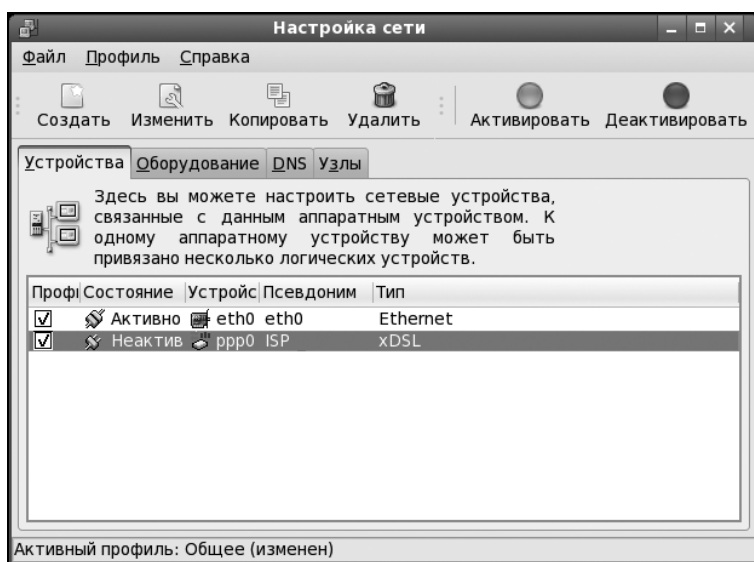


Рис. 28.4. Включение/выключение соединения

Первая команда выключает сервис NetworkManager (если он вообще запущен), а вторая отключает запуск этого сервиса на уровнях запуска 2, 3, 4 и 5.

28.3.2. В openSUSE

Запустите Центр управления и выберите **DSL**. Пользователям радиодоступа к Интернету (технология Radio Ethernet) тоже нужно использовать конфигуратор DSL — настройка Radio Eterhet осуществляется аналогично настройке DSL.

ПРИМЕЧАНИЕ

Для непосредственного запуска (не через Центр управления) конфигулятора модема используется команда `/sbin/yast2 dsl`.

Конфигуратор попытается найти DSL-устройства. Это может занять некоторое время, так что придется немного подождать (рис. 28.5).

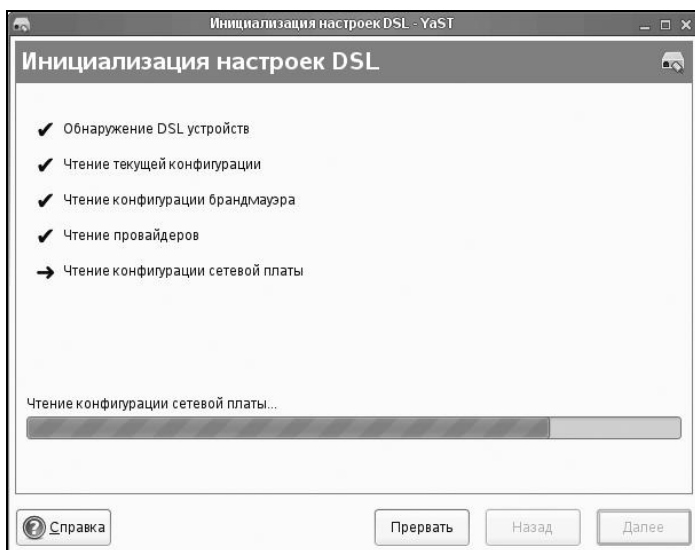


Рис. 28.5. Поиск DSL-устройств

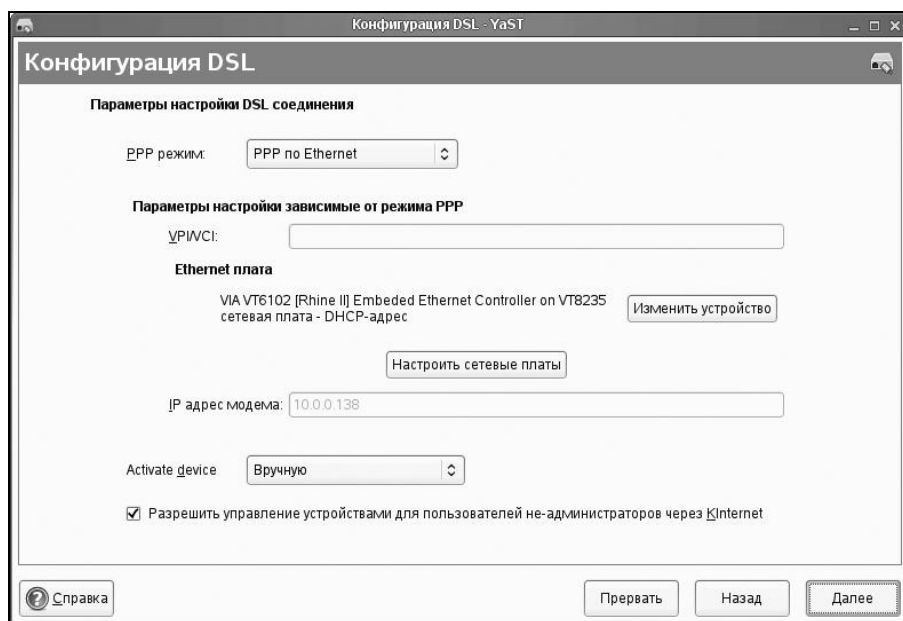


Рис. 28.6. Параметры DSL-соединения

Далее вы увидите пустое окно обзора настроек DSL (как будто не найдено ни одного DSL-устройства). Не пугайтесь — так и должно быть. Просто нажмите кнопку **Далее**. Вам нужно задать параметры DSL-соединения (рис. 28.6), а именно: выбрать режим PPP, сетевую плату, к которой подключен DSL-модем, режим акти-

зации устройства и обязательно разрешить управление соединением через KInternet (иначе вы просто не сможете использовать KInternet).

- ◆ Начнем с режима PPP — обычно используется режим **PPP по Ethernet**. Технология ADSL (как и другие технологии, например Radio Ethernet), использует протокол PPPoE (Point to Point Protocol over Ethernet).

ПОЯСНЕНИЕ

Протокол PPP используется обычным модемным соединением, а протокол PPPoE обеспечивает передачу PPP-кадров по сетевой плате (Ethernet) — это и есть суть режима PPP по Ethernet.

- ◆ Сетевая плата обычно выбирается конфигуратором правильно, поэтому ее не нужно изменять, тем более, что в большинстве случаев найденная сетевая плата является единственным сетевым адаптером в системе.
- ◆ Режим активации устройства (**Activate device**) позволяет определить, как будет активироваться устройство — вручную или автоматически при запуске системы. Тут решать вам — можно запускать DSL-соединение и при запуске системы, но тогда отпадает необходимость в использовании KInternet.

Следующий этап настройки DSL-соединения — это выбор провайдера. Вашего провайдера не будет в списке, поэтому сразу нажимайте кнопку **новый**, вводите имя провайдера, имя пользователя и пароль (рис. 28.7).

Рис. 28.7. Информация о провайдере

Теперь следует определить некоторые параметры соединения. Параметры, предложенные конфигуратором (рис. 28.8), вполне приемлемы и устроят большинство пользователей, поэтому просто просмотрите их и нажмите кнопку **Далее**.

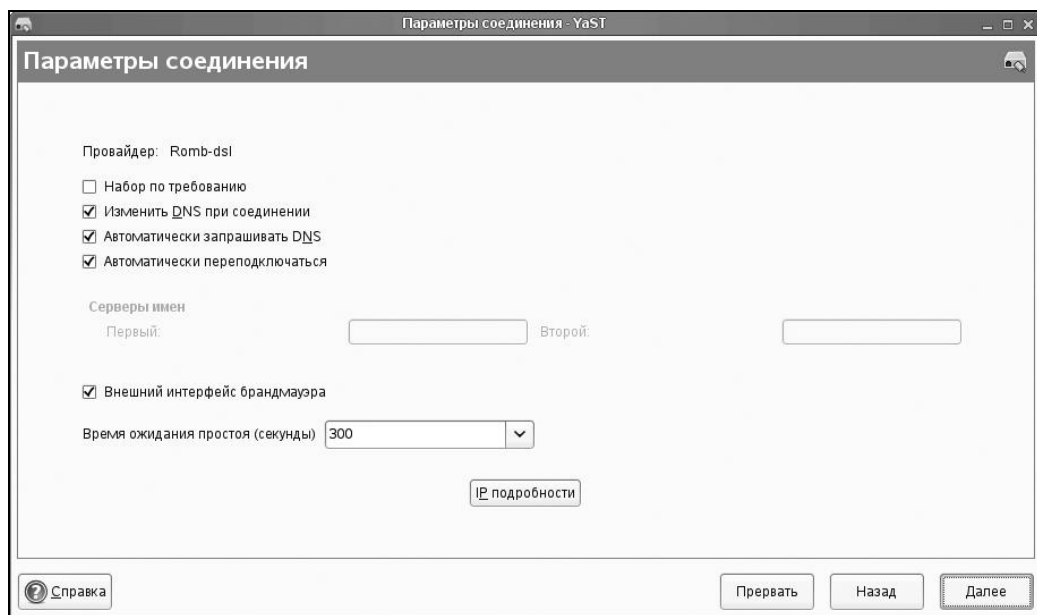


Рис. 28.8. Параметры соединения

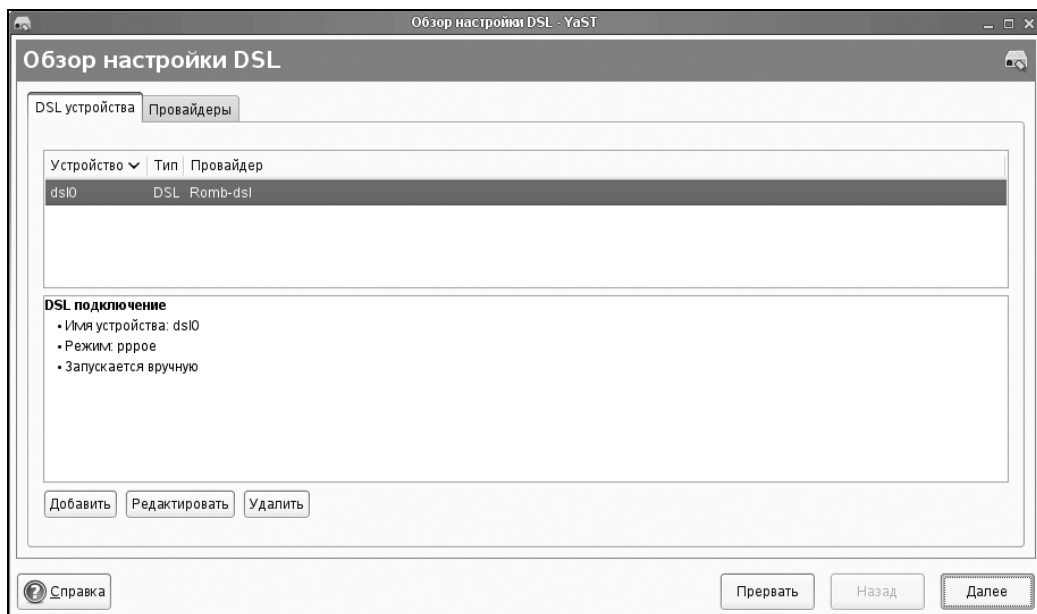


Рис. 28.9. Созданное соединение

Вы вернетесь в окно обзора DSL-соединений, которое теперь не будет пустым — в нем появится только что созданное соединение (рис. 28.9).

Все, что вам осталось, — это нажать кнопку **Далее** и подождать, пока YaST сохранит конфигурацию системы (рис. 28.10).

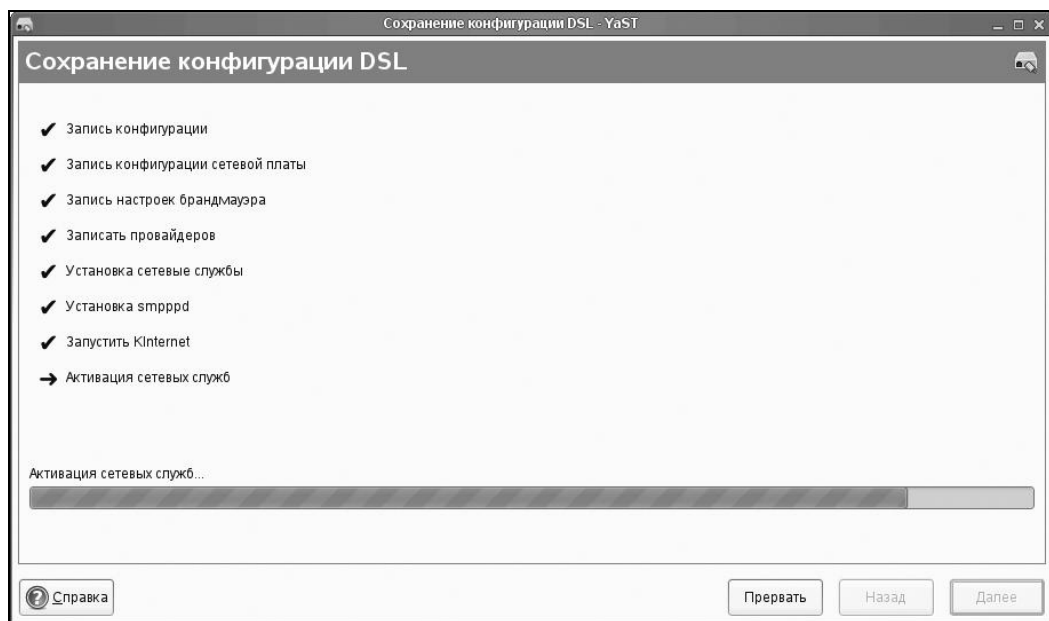


Рис. 28.10. Сохранение конфигурации

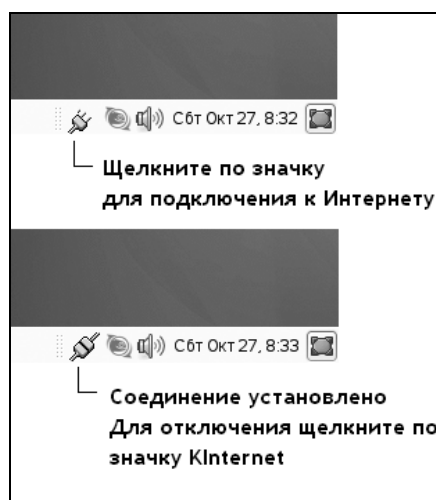


Рис. 28.11. Программа KInternet

Для подключения к Интернету нужно щелкнуть по значку **KInternet** (рис. 28.11). Но если вы до этого настраивали модемное соединение, вам теперь нужно выбрать DSL-подключение. Для этого щелкните правой кнопкой мыши по значку **KInternet** и выберите команду меню **Интерфейс | dsl0**. Вот теперь можно щелкнуть по значку левой кнопкой мыши для установки соединения. Для отключения, как обычно, нужно снова щелкнуть по значку **KInternet**.

28.3.3. В Ubuntu

В дистрибутивах Debian и Ubuntu для настройки DSL-соединений используется конфигуратор `pppoeconf`. Откройте терминал и введите команду:

```
sudo pppoeconf
```

ПРИМЕЧАНИЕ

Начиная с Ubuntu 8.10, появился новый графический конфигуратор сети — NetworkManager. Для его запуска выполните команду **Система | Параметры | Сетевые соединения** и перейдите на вкладку **DSL**. Но вы, по желанию, можете использовать и программу `pppoeconf` — она также работает в новых версиях Ubuntu. Тут уж дело вкуса: некоторые пользователи предпочитают графические конфигураторы, а кто-то — работает в консоли. Лично я настраивал DSL-соединение в своем Ubuntu 10.04 с помощью `pppoeconf`.

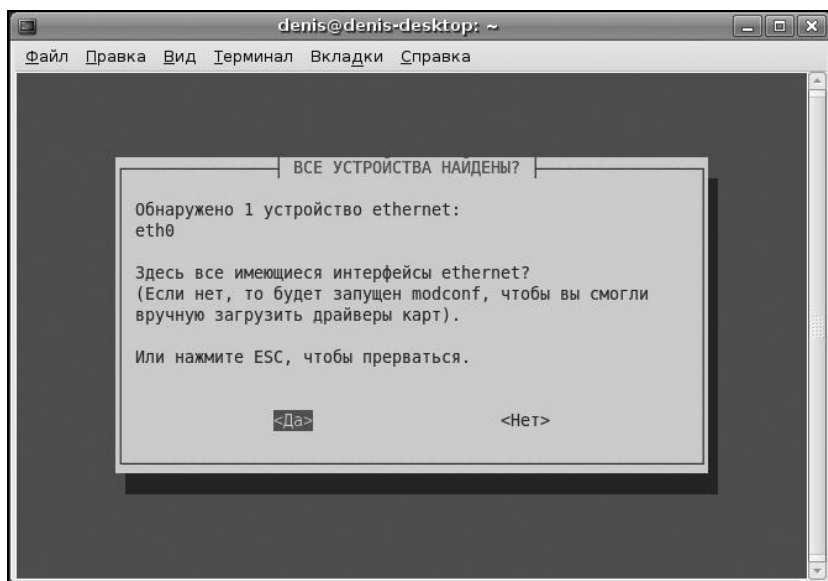


Рис. 28.12. Конфигуратор `pppoeconf` нашел Ethernet-устройство

Согласно спецификации PPPoE существуют две стадии: стадия поиска и стадия сессии. На первой стадии производится отправка специальных пакетов PADI (PPPoE Active Discovery Initiation), которые позволяют найти активные concentra-

торы доступа PPPoE (рис. 28.12). Стадия сессии — это само соединение и передача информации.

Затем конфигуратор попытается найти активный концентратор доступа (рис. 28.13). После того как концентратор доступа будет найден, программа предложит вам установить популярные опции соединения (`noauth` и `defaultroute`): не стоит от них отказываться, поскольку их использует большинство провайдеров (рис. 28.14).

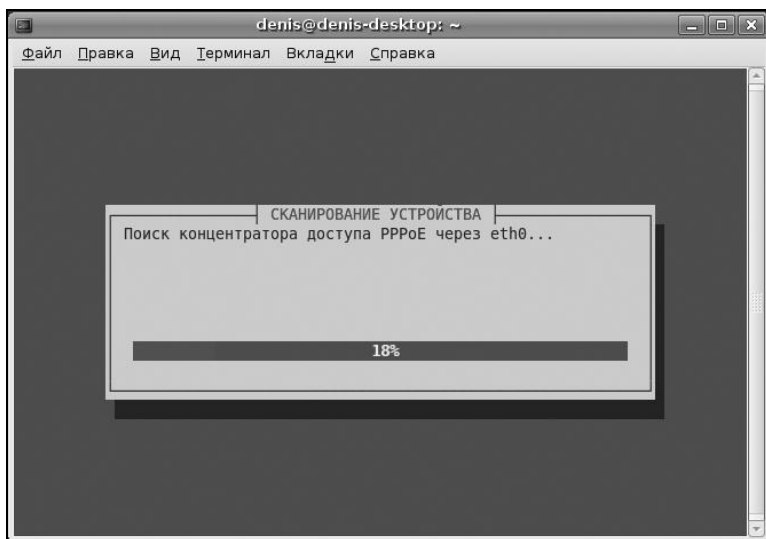


Рис. 28.13. Поиск активного концентратора доступа

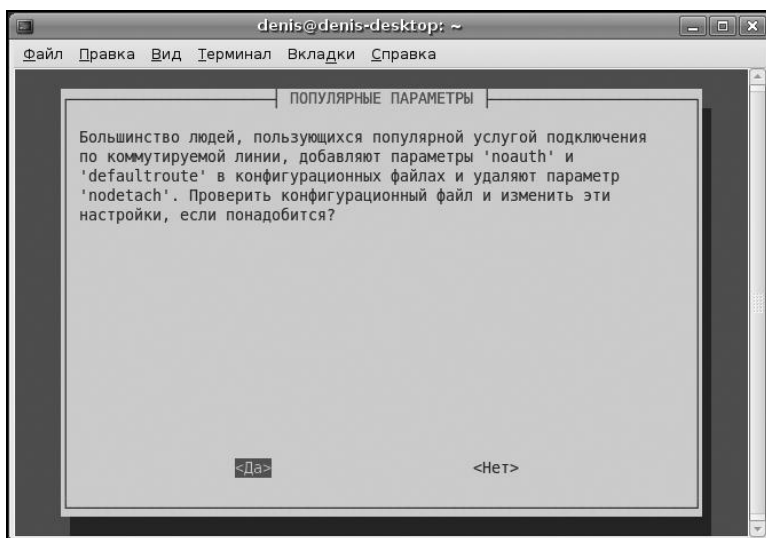


Рис. 28.14. Популярные опции соединения

Следующие два шага — ввод имени пользователя и пароля, которые используются для аутентификации на сервере провайдера. После этого программа предложит вам добавить полученные от провайдера IP-адреса DNS-серверов в файл `/etc/resolv.conf`. Не стоит от этого отказываться (рис. 28.15).

На следующий вопрос (рис. 28.16) можно просто ответить **Да**, не вникая в подробности. Если же вам интересно, прочитайте следующее примечание.

ПРИМЕЧАНИЕ

Параметр MTU (Maximum Transmit Unit) задает максимальный размер пакета. По умолчанию данное значение может быть установлено автоматически, но не всегда оптимально. Если размер пакета окажется по размеру больше, чем позволяет маршрутизатор провайдера, то пакет будет разделен на несколько пакетов, что, естественно, скажется на скорости и пропускной способности соединения. Если размер пакета будет меньше, чем положено, то это тоже не хорошо — канал будет использован нерационально, ведь будут проходить полупустые кадры. Поскольку мы работаем по протоколу PPPoE, то нужно учитывать несколько факторов. Максимальный размер кадра Ethernet составляет 1518 байтов, из которых 18 уходит на заголовок и контроль, поэтому для полезных данных остается 1500 байтов. Обычно данное значение и указывается для Ethernet. Но ведь по Ethernet мы собираемся передавать пакеты PPP, а PPPoE отбирает еще 6 байтов, PPP — 2 байта. Получается, что для PPPoE значение MTU должно быть равно 1492. При установке TCP-соединения каждая сторона устанавливает параметр MSS (Maximum Segment Size) — максимальный размер TCP-сегмента. По умолчанию его размер равен MTU минус размер заголовков TCP/IP, которые занимают еще 40 байтов. То есть размер MSS для PPPoE равен 1452 байта (для обычного Ethernet — 1460). Вот откуда взялось значение 1452.

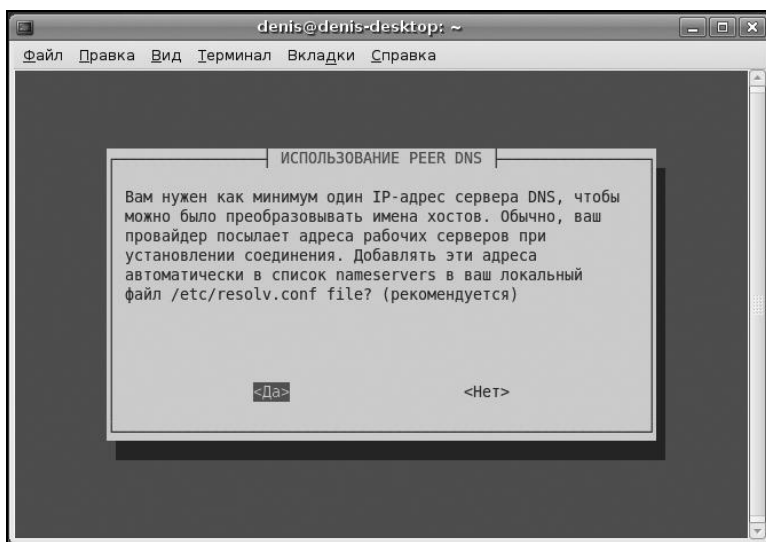


Рис. 28.15. Добавляем IP-адреса DNS-серверов в файл `/etc/resolv.conf`

Следующий вопрос — хотите ли вы устанавливать соединение при загрузке системы. Тут уж решайте сами. А после этого программа спросит вас, хотите ли вы

установить соединение немедленно. Конечно, да! Можно сразу запускать браузер и заходить на любимую страничку.

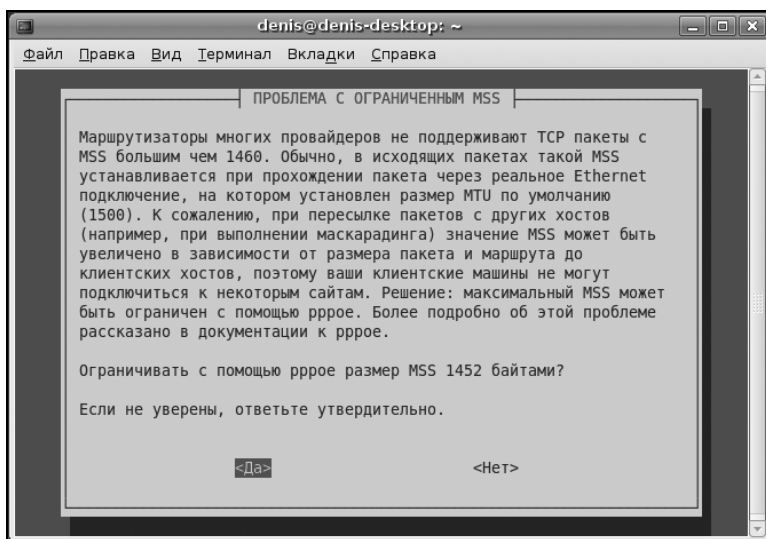


Рис. 28.16. Установка размера MSS

Для включения/отключения DSL-соединения используются следующие команды:

```
sudo pon dsl-provider
sudo poff dsl-provider
```

28.3.4. В Mandriva

Для экономии места в книге (а, значит, и для экономии ваших денег) подробно настройку DSL-соединения в Mandriva рассматривать мы не будем. Настройка производится с помощью конфигулятора `drakconnect`. Ответив на несложные вопросы конфигулятора, вы за считанные секунды настроите соединение с Интернетом.

ГЛАВА 29



Команды для работы с сетью и Интернетом

29.1. Команда *ifconfig*: управление сетевыми интерфейсами

Команда *ifconfig* используется для получения информации о сетевых интерфейсах и для установки параметров сетевых интерфейсов. Обычно эта команда вызывается при запуске системы сценариями инициализации системы, и вам не придется ее использовать для настройки интерфейсов вручную (разве что на самых древних дистрибутивах).

Формат вызова команды следующий:

```
ifconfig -a [параметры | семейство_протоколов]
ifconfig интерфейс [параметры | семейство_протоколов]
```

Чтобы просто просмотреть информацию о сетевых интерфейсах, введите команду:

```
ifconfig
```

Посмотрите на рис. 29.1. "Поднят" только интерфейс *lo* — это интерфейс локальной петли, используемой для тестирования сети. Если есть только интерфейс *lo*, значит, остальные сетевые интерфейсы не настроены. В современных дистрибутивах настройка интерфейсов производится автоматически при запуске системы — ведь практически всегда в сети есть DHCP-сервер, который и настраивает сетевые интерфейсы.

Настроить интерфейс можно и с помощью *ifconfig*. Вот пример настройки интерфейса *eth0* (первая сетевая плата), когда ему присваивается IP-адрес 192.168.1.7 и он поднимается ("up"):

```
# /sbin/ifconfig eth0 192.168.1.7 up
```

Для полной настройки сетевого интерфейса нужно использовать команду:

```
# /sbin/ifconfig eth0 адрес broadcast ш_адрес netmask маска
```

После имени интерфейса задается IP-адрес, затем широковещательный адрес (ш_адрес), потом сетевая маска. На рис. 29.2 показан уже настроенный интерфейс eth0.

```
[root@localhost ~]# ifconfig
lo          Link encap:Local Loopback
            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:62 errors:0 dropped:0 overruns:0 frame:0
            TX packets:62 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:0
            RX bytes:3914 (3.8 Kb)  TX bytes:3914 (3.8 Kb)

[root@localhost ~]#
```

Рис. 29.1. Настроен только интерфейс lo

```
[root@localhost ~]# ifconfig
eth0        Link encap:Ethernet  HWaddr 00:6D:97:88:BC:96
            inet addr:192.168.1.7  Bcast:192.168.1.255  Mask:255.255.255.0
            inet6 addr: fe80::20d:87ff:fe88:bc96/64 Scope:Link
            UP BROADCAST MULTICAST  MTU:1500  Metric:1
            RX packets:0 errors:0 dropped:0 overruns:0 frame:0
            TX packets:12 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:0 (0.0 b)  TX bytes:672 (672.0 b)
            Interrupt:11 Base address:0xe800

lo          Link encap:Local Loopback
            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 ~]#
```

Рис. 29.2. Интерфейсы lo и eth0

Вообще, перед тем как вводить команду `ifconfig`, неплохо было бы добавить модуль сетевой платы, без него `ifconfig` работать не будет. Вот пример добавления модуля для сетевой платы Realtek 8139:

```
# insmod rtl8139.o
```

29.2. Текстовые браузеры

Если графический режим недоступен (например, на сервере), а по сети побродить хочется, можно использовать текстовый браузер `lynx`. В некоторых Distribu-

тивах вместо `lynx` используются браузеры `links` и `elinks`, но суть остается та же — просмотр страниц Интернета в текстовом режиме.

В современных дистрибутивах текстовые браузеры не устанавливаются по умолчанию, поэтому их нужно установить отдельно.

29.3. Команда *ftp*: FTP-клиент

Для открытия соединения с любым FTP-сервером введите команду:

```
ftp <имя или адрес FTP-сервера>
```

Можно просто ввести команду `ftp`, а в ответ на приглашение

```
ftp>
```

ввести команду:

```
open <имя или адрес FTP-сервера>
```

Лично мне больше нравится первый вариант, поскольку он позволяет сэкономить время. При подключении к серверу вы сможете ввести имя пользователя и пароль:

```
[den@dhsilabs ~]$ ftp
ftp> open ftp.narod.ru
Connected to ftp.narod.ru.
220 ftp.narod.ru (Libra FTP daemon 0.17 20050906)
500 Unrecognized command AUTH
Name (ftp.narod.ru:den): den
331 Password required
Password:
230 Logged in, proceed
Remote system type is UNIX.
ftp>
```

Подключившись к серверу, вы можете ввести команду `help`, чтобы просмотреть список доступных команд. Для получения справки по той или иной команде введите `help <имя_команды>`. Наиболее популярные команды приведены в табл. 8.1.

Таблица 8.1. Некоторые команды FTP-клиента

Команда	Описание
<code>ls</code>	Вывод содержимого каталога
<code>get</code>	Загрузить файл с сервера
<code>put</code>	Загрузить файл на сервер

Таблица 8.1 (окончание)

Команда	Описание
<code>mget</code>	Получить несколько файлов с сервера. Допускается использование масок файлов, например <code>*.rpm</code>
<code>mput</code>	Загрузить несколько файлов на сервер
<code>cd</code>	Изменить каталог
<code>mkdir</code>	Создать каталог
<code>rmdir</code>	Удалить пустой каталог
<code>delete</code>	Удалить файл

Кроме `ftp`, в Linux есть и другие текстовые FTP-клиенты, например `NcFTP` (<http://www.ncftp.com>), `lukemftp` (<ftp://ftp.netbsd.org/pub/NetBSD/misc/lukemftp/>), `lftp` (<http://ftp.yars.free.net/projects/lftp/>) и др. Все эти FTP-клиенты не входят в состав дистрибутива, их нужно устанавливать самостоятельно. Но стоит ли это делать — решать вам. Ведь все они подобны стандартному клиенту `ftp` и обладают двумя-тремя дополнительными функциями, которые, возможно, вам и не понадобятся. Например, `NcFTP` умеет докачивать файлы, а `lftp` — загружать одновременно несколько файлов. В любом случае вы можете изучить документацию по тому или иному FTP-клиенту (ее легко найти в Интернете), а потом решить, стоит ли его использовать или нет.

29.4. Команда `wget`: загрузка файлов

Программа `wget` — это лучший текстовый менеджер закачки файлов. Программа поддерживает протоколы HTTP, HTTPS и FTP. Использовать ее нужно так:

```
wget [параметры] URL
```

Параметров у `wget` очень и очень много, и со всеми ними вы ознакомитесь на странице `man wget`. Самые полезные параметры собраны в табл. 8.2.

Таблица 8.2. Некоторые параметры `wget`

Параметр	Описание
<code>--background</code>	Перейти в фоновый режим после запуска
<code>--quiet</code>	Тихий режим, сообщения <code>wget</code> не выводятся
<code>--input-file=file</code>	Считать URL из файла <i>file</i> , файл не обязательно должен быть в формате HTML. Если вы указали URL в файле и в командной строке, то сначала будут загружены URL из командной строки, а потом из файла

Таблица 8.2 (окончание)

Параметр	Описание
<code>--force-html</code>	Обязательно считать файл, указанный в предыдущем параметре, HTML-файлом
<code>--tries=number</code>	Устанавливает количество попыток загрузки URL
<code>--no-clobber</code>	Если при загрузке файла оборвалось соединение, то этот параметр позволит продолжить загрузку с места обрыва
<code>--continue</code>	Возобновление загрузки файла, например, если прервалась связь. Этот параметр нужно использовать, если вы забыли указать параметр <code>--no-clobber</code> , а связь прервалась и вам нужно докачать файл, а не начинать его загрузку заново
<code>--wait=seconds</code>	Задаёт паузу в секундах между загрузками и повторами, что позволяет снизить нагрузку на сервер
<code>--quota=quota</code>	Задаёт максимальный размер загружаемых файлов (в байтах, килобайтах (после числа указывается <i>k</i>) и мегабайтах (после числа — <i>m</i>)). Квота не работает при загрузке одного файла, поскольку даже если квота превышена, то текущий файл загружается до конца (если есть физически место на диске)
<code>--http-user=user</code> <code>--http-passwd=pass</code>	Задают имя пользователя и пароль при HTTP-аутентификации, тип аутентификации устанавливается автоматически программой
<code>--proxy-user=user</code> <code>--proxy-passwd=pass</code>	Задаёт имя пользователя и пароль прокси-сервера
<code>--passive-ftp</code>	Пассивный режим FTP, обычно используется при наличии брандмауэра
<code>--recursive</code>	Включить рекурсивную загрузку, которая используется для рекурсивной загрузки сайтов.
<code>--level=depth</code>	Максимальная длина рекурсивной загрузки (по умолчанию 5 уровней)

Примеры использования:

```
wget --recursive http://dkws.org.ua
```

```
wget http://dkws.org.ua/1.zip
```

Первая команда создаст пятиуровневую копию сайта **http://dkws.org.ua**, а вторая просто загрузит файл **1.zip** с **http://dkws.org.ua**.

29.5. Команда *mail* — чтение почты и отправка сообщений

Программа `mail` — это простейший клиент для чтения и отправки почты. Позволяет читать только почту, принятую вашей системой. Если же нужно принять почту с других POP3-серверов, тогда нужно использовать другие почтовые клиенты, которые могут работать в консоли, например `mutt` или `pine`.

Для чтения предназначенных вам сообщений введите команду `mail` без параметров. Если хотите написать кому-то письмо, передайте в качестве параметра электронный адрес этого человека:

```
mail ivanov@firma.ru
```



ЧАСТЬ VIII

LINUX-СЕРВЕР

ГЛАВА 30



Суперсервер xinetd

30.1. Сетевые сервисы и суперсервер

Сетевые сервисы могут запускаться автономно или только по требованию, т. е. при получении от клиента запроса. Автономно запускаются те сервисы, от которых клиент ожидает немедленной реакции, например Web-сервер, FTP-сервер, DNS-сервер. Другие сетевые сервисы, например finger, tftp, могут позволить себе запуск по требованию. Но как передать запрос сервису, который не запущен? Ведь процесс не запущен, следовательно, некому и принять запрос от клиента.

Для запуска сетевых сервисов по требованию используется суперсервер xinetd. Данный сервер всегда находится в памяти и принимает на себя все запросы (кроме запросов, адресованных к автономным службам). Затем он анализирует запрос и запускает необходимую сетевую службу для его обработки. Такая схема позволяет экономить системные ресурсы, потому что не нужно держать в памяти все редко используемые сетевые сервисы.

Суперсервер xinetd имеется в дистрибутивах Fedora, Mandriva и всех их клонах, но вы не найдете его в Ubuntu — там его роль выполняет система инициализации upstart.

30.2. Конфигурационный файл суперсервера

Конфигурационный файл xinetd называется /etc/xinetd.conf. В современных дистрибутивах этот файл довольно небольшой (листинг 30.1), потому что осталось мало служб, запускаемых с помощью xinetd — в основном используется автономный запуск.

Листинг 30.1. Пример конфигурационного файла /etc/xinetd.conf

```
defaults
{
# максимальное число одновременно запущенных экземпляров сервера
    instances          = 60
```



```
# параметры протоколирования
    log_type           = SYSLOG authpriv
    log_on_success     = HOST PID
    log_on_failure     = HOST

# Параметр cps: первый аргумент — количество соединений в секунду;
# второй аргумент — число секунд, по истечении которых сервис снова будет
# доступен после превышения первого аргумента cps
    cps                = 25 30
}

# каталог, содержащий конфигурационные файлы отдельных сетевых сервисов
includedir /etc/xinetd.d
```

В каталоге, заданном директивой `includedir`, содержатся конфигурационные файлы сетевых сервисов. Каждый сервис описывается в собственном файле. Сервис описывается так:

```
service название
{
    параметры
}
```

Вот пример описания сервиса `rsync`:

```
service rsync
{
# сервис отключен
# чтобы его включить, нужно указать disable = no или вообще не указывать
# disable
    disable                = yes
# тип сокета (stream для TCP, dgram для UDP, raw — для сервисов, требующих
# прямого обращения к протоколу IP)
    socket_type            = stream
# для TCP нужно установить значение no, для UDP — yes
    wait                   = no
# пользователь, от имени которого работает сервис
    user                   = root
# вызываемый сервер (исполнимый файл сетевой службы)
    server                 = /usr/bin/rsync
# аргументы, которые будут переданы серверу (зависит от сервера)
    server_args            = --daemon
# что протоколировать при сбое (USERID — ID пользователя,
# HOST — имя удаленного узла)
    log_on_failure        += USERID
}
```

ГЛАВА 31



Web-сервер. Связка Apache + PHP + MySQL

31.1. Самый популярный Web-сервер

Apache — это Web-сервер с открытым исходным кодом. История его развития началась в 1995 году — тогда Apache был всего лишь "заплаткой", устраняющей ошибки популярного в то время Web-сервера NCSA HTTPd 1.3. Считается, что отсюда произошло и название Apache (от англ. *a patchy* — заплатка). Сейчас Apache — самый популярный Web-сервер в Интернете: в апреле 2007 года было подсчитано, что он установлен на 58% Web-серверов в Интернете.

Основные достоинства Apache — надежность, безопасность и гибкость настройки. Apache позволяет подключать различные модули, добавляющие в него новые возможности — например, можно подключить модуль, обеспечивающий поддержку PHP или любого другого Web-ориентированного языка программирования.

Но есть и недостатки — без этого никак, всегда есть обратная сторона медали. Основной недостаток — отсутствие удобного графического интерфейса администратора. Да, настройка Apache осуществляется путем редактирования его конфигурационного файла. В Интернете можно найти простые конфигураторы Apache, но их возможностей явно не хватает для настройки всех функций Web-сервера.

31.2. Установка Web-сервера и интерпретатора PHP. Выбор версии

Вы можете установить одну из версий Apache: Apache 1.3.34, Apache 2 или Apache 2.2. С одной стороны, версия Apache 2.2 более новая и современная, Apache 2 — стабильная и уже проверенная. С другой, версии 1.3.x все еще поддерживаются и есть в репозиториях некоторых дистрибутивов.

Однако я рекомендую установить Apache 2. И дело тут даже не в том, что эта версия более новая. Ради эксперимента я установил сначала версию 1.3.34, интерпретатор PHP4 и еще ряд дополнительных пакетов объемом 14 Мбайт. Оказалось, что в одной из библиотек или конфигурационном файле этого набора была ошибка — что я ни делал, поддержки PHP не было. А поддержка PHP очень нужна на современном Web-сервере! Поэтому я сразу установил вторую версию Apache и PHP 5. Кроме того, вторая версия проще в настройке, поддерживает протокол IPv6 и многопоточность, умеет выводить сообщения об ошибках на разных языках.

Что же касается версии 2.2, то она еще не достаточно "обкатанная". Если нужно настроить корпоративный сервер, вполне хватит версии 2.0, а если хочется поэкспериментировать с новой версией — можно установить версию 2.2.

Запустите менеджер пакетов (например, Synaptic, используемый в Ubuntu). Произведите поиск пакета apache. Выберите пакет apache2. Менеджер пакетов сообщит вам, что нужно установить дополнительные пакеты (рис. 31.1).

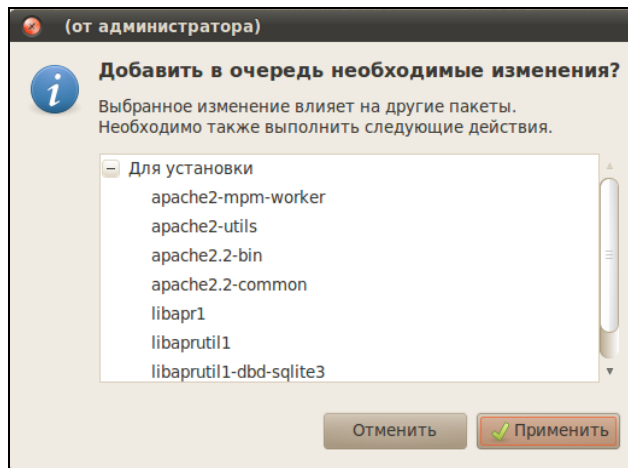


Рис. 31.1. Дополнительные пакеты для Apache

Чтобы сразу "убить двух зайцев", выберите еще и пакет php5. Он устанавливает PHP 5 и добавляет его поддержку в Apache. Опять менеджер предложит установить дополнительные пакеты, но для PHP (рис. 31.2).

Нажмем кнопку **Применить**, и машина установит все выбранное. После этого рекомендуется установить и следующие пакеты (их можно найти по запросу php):

- ◆ php5-cli — интерпретатор PHP, работающий в режиме командной строки (command-line interpreter);
- ◆ php5-imap — поддержка протоколов POP/IMAP для PHP;
- ◆ php5-gd — поддержка графических функций PHP;
- ◆ php5-mysql — поддержка функций для работы с базой данных MySQL.

Если вы выбрали PHP 4, тогда вам нужно установить эти же пакеты, но для PHP 5 (php4-*). Необходимые дополнительные пакеты будут установлены автома-

тически, об этом позаботится менеджер пакетов. Просмотрите весь список пакетов, возможно, нужные вам найдутся.

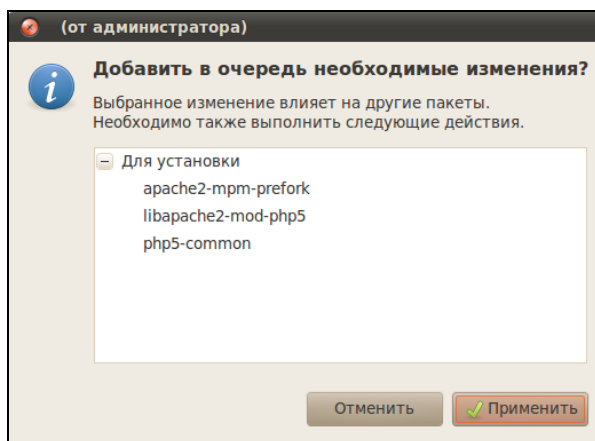


Рис. 31.2. Дополнительные пакеты для PHP

31.3. Тестирование настроек

Теперь протестируем Web-сервер. По идее, после установки сервер должен запуститься автоматически. Но в некоторых дистрибутивах его нужно запустить вручную (см. разд. 31.5).

Запустите сервер или убедитесь, что он запущен (см. разд. 31.5). Откройте браузер и введите адрес:

`http://localhost`

Должна открыться страница, изображенная на рис. 31.3.

После этого протестируем поддержку PHP. Поместите в каталог `/var/www/` файл `test.php` (листинг 31.1).

Листинг 31.1. Файл `test.php`

```
<?
phpinfo();
?>
```

Чтобы создать файл в этом каталоге, нужны права root. После того как файл будет создан, введите в строке браузера следующий адрес:

`http://localhost/test.php`

В окне браузера вы должны увидеть информацию о своем сервере и PHP (рис. 31.4).

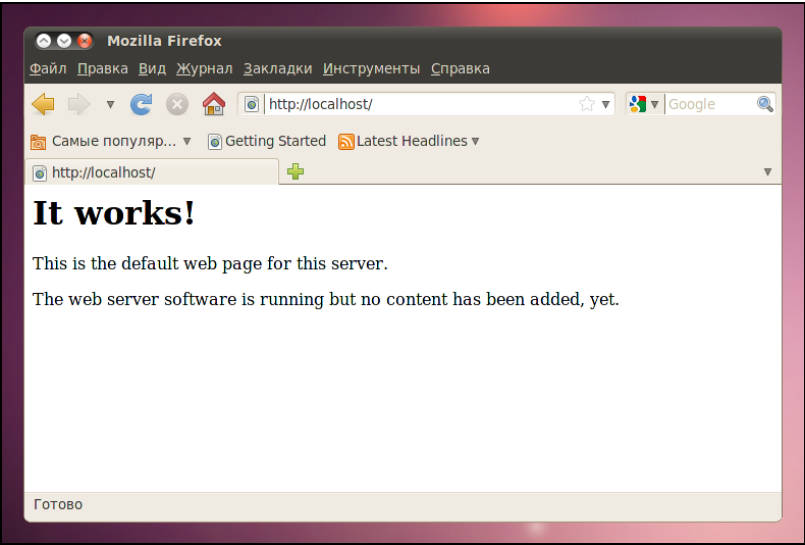


Рис. 31.3. Тестовая страница Apache

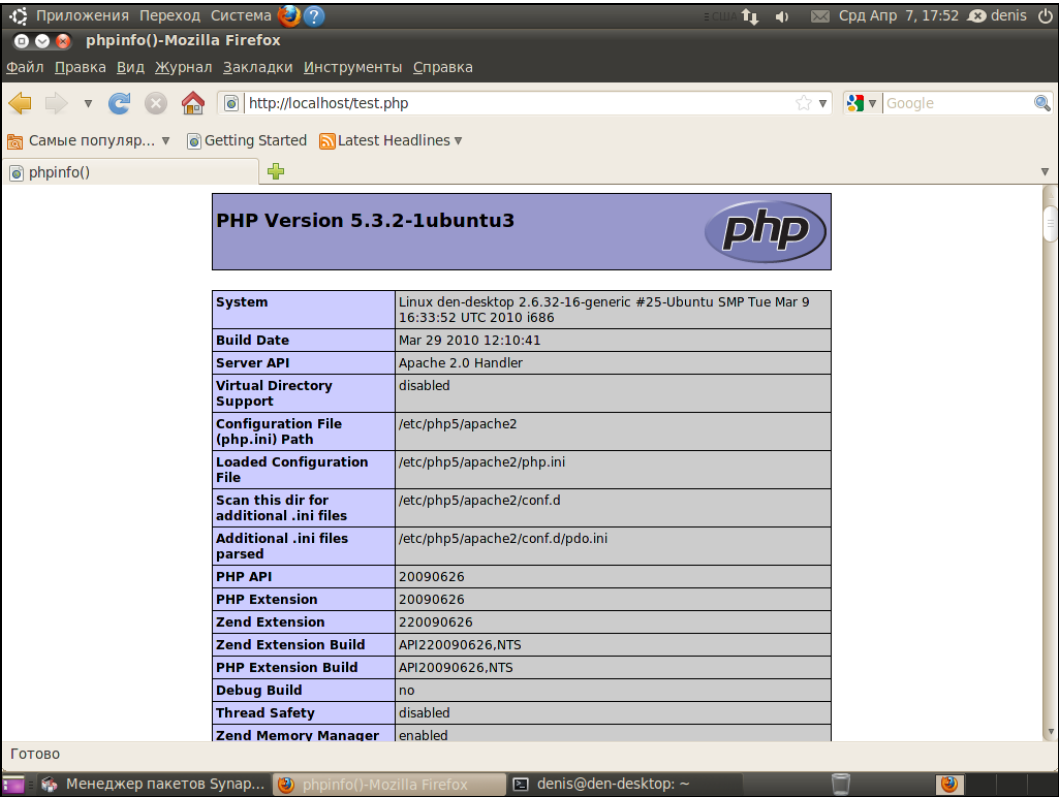


Рис. 31.4. Тестовый сценарий

ПРИМЕЧАНИЕ

Если вместо отображения тестовой странички, изображенной на рис. 31.4, браузер предлагает вам сохранить файл `test.php`, перезапустите Web-сервер (см. разд. 31.5).

Как вы уже догадались, каталог `/var/www` является корневым для вашего сервера. Если создать в нем файл `test.html`, то он будет доступен по адресу `http://localhost/test.html`.

31.4. Файл конфигурации Web-сервера

31.4.1. Базовая настройка

В зависимости от версии Apache и вашего дистрибутива, конфигурационные файлы Apache могут находиться в следующих каталогах: `/etc/apache`, `/etc/apache2`, `/etc/httpd` или `/etc/httpd2`. Основные конфигурационные файлы называются `httpd.conf`, `httpd2.conf` или `apache.conf` и `apache2.conf`. Название каталогов и файлов, содержащих слово "apache", характерно для дистрибутивов Debian и Ubuntu, а содержащих слово "httpd" — для Mandriva/Fedora. В любом случае найти конфигурационные файлы не сложно: ищите или `apache`, или `httpd` — и не промахнетесь!

ПРИМЕЧАНИЕ

После каждого изменения конфигурационных файлов сервера его нужно перезапустить (см. разд. 31.5)!

Раньше все настройки хранились в одном огромном файле конфигурации. Сейчас этот файл чаще содержит Include-инструкции подключения других файлов (более компактных) конфигурации. Все это сделано для удобства администраторов: проще работать с несколькими компактными файлами, чем с одним огромным.

Первым делом откройте конфигурационный файл (для определенности будем считать, что он называется `httpd2.conf`) и найдите директиву:

```
#ServerName new.host.name
```

Нужно ее раскомментировать и указать имя сервера, которое будут задавать пользователи в строке браузера. Данное имя должно быть зарегистрировано в DNS-сервере вашей сети (или указано в файле `/etc/hosts` каждого компьютера сети). Обычно здесь указывается имя компьютера, например:

```
ServerName user-desktop
```

После этого можно будет обращаться к серверу по адресу **`http://user-desktop/`**.

31.4.2. Самые полезные директивы файла конфигурации

Понятно, что для полноценной настройки сервера одной директивы `ServerName` недостаточно. В табл. 31.1 приведены самые полезные директивы файла конфигурации Apache. Нужно отметить, что в таблице не рассматриваются некоторые директивы (например, `Port`, `BindAddress`), которые не используются во второй версии Apache.

Таблица 31.1. Директивы файла конфигурации

Директива	Описание
<code>ServerName</code> <i>имя</i>	Задаёт имя Web-сервера, имя должно быть зарегистрированным на DNS-сервере, т. е. обычно это доменное имя сервера
<code>ServerAdmin</code> <i>e-mail</i>	Задаёт e-mail администратора сервера
<code>ServerRoot</code> <i>каталог</i>	Определяет каталог с конфигурационными файлами сервера
<code>PidFile</code> <i>файл</i>	Определяет имя файла, в котором будет храниться PID исходного процесса Web-сервера. Обычно изменять эту директиву не нужно
<code>DocumentRoot</code> <i>каталог</i>	Позволяет задать каталог, в котором хранятся документы Web-сервера — это корневой каталог документов. Обычно это <code>/var/www</code>
<code>StartServers</code> N, <code>MaxSpareServers</code> N, <code>MinSpareServers</code> N, <code>MaxClients</code> N	Директивы, непосредственно влияющие на производительность сервера. Мы их рассмотрим отдельно в разд. 31.6
<code>KeepAlive</code> On Off, <code>KeepAliveTimeout</code> N	Управляют постоянными соединениями, будут рассмотрены в разд. 31.6
<code>DirectoryIndex</code> <i>список</i>	Задаёт имена файлов, которые могут использоваться в качестве главной страницы (индекса). Значение по умолчанию: <code>index.html index.cgi index.pl index.php index.xhtml</code>
<code>HostnameLookups</code> On Off	Если директива включена (On), то IP-адрес клиента перед записью в журнал будет разрешен (т. е. Web-сервер вычислит доменное имя клиента перед записью информации о попытке доступа в журнал). Выключение (Off) этой опции позволяет повысить производительность сервера, поскольку не нужно тратить время на разрешение IP-адресов в доменные имена
<code>ErrorLog</code> <i>файл</i>	Задаёт журнал ошибок
<code>TransferLog</code> <i>файл</i>	Задаёт журнал обращений к серверу

Таблица 31.1 (окончание)

Директива	Описание
Timeout <i>N</i>	Тайм-аут в секундах (время, на протяжении которого сервер будет ждать возобновления прерванной попытки передачи данных)
User <i>пользователь</i> , Group <i>группа</i>	Директивы User и Group задают имя пользователя и группы, от имени которых запускается Web-сервер
FancyIndexing On Off	Если пользователь в запросе не укажет имя документа, а только каталог, но в нем не окажется главной страницы, заданной директивой DirectoryIndex, сервер передаст пользователю оглавление каталога. Данная директива определяет, в каком виде будет передано оглавление каталога: в более красивом, со значками каталогов и описаниями файлов (значение On), или в более простом (Off)
AddIcon <i>картинка</i> <i>список</i>	Если FancyIndexing включена, то AddIcon позволяет связать графическую картинку с типом файла, например: AddIcon /images/graphics.gif .gif, .jpeg, .bmp, .png, .tiff
DefaultIcon <i>картинка</i>	Позволяет задать картинку по умолчанию (AddIcon, FancyIndexing)
ErrorDocument <i>N</i> <i>файл</i>	Позволяет задать файл, содержащий сообщение об ошибке, для ошибки с номером <i>N</i> , например: ErrorDocument 404 /errors/file_not_found.html
Directory, Limit, Location, Files	Это так называемые <i>блочные</i> директивы, которые нельзя описать одной строкой, поэтому о них мы поговорим отдельно (см. <i>разд. 31.4.3</i>)

31.4.3. Директивы *Directory*, *Limit*, *Location*, *Files*

Рассмотрим сначала блочные директивы *Directory* и *Limit*.

- ✦ С помощью блочной директивы *Directory* можно установить параметры отдельного каталога. Внутри директивы *Directory* могут использоваться директивы *AllowOverride*, *Limit*, *Options*. Вот пример определения параметров корневого сервера:

```
<Directory />
AllowOverride None
Options None
</Directory>
```

Значения *None* для обеих директив (*AllowOverride* и *Options*) считаются самыми безопасными. *None* для *AllowOverride* запрещает использование файлов

.htaccess, которые могут переопределять директивы конфигурационного файла Apache. К тому же, `AllowOverride None` позволяет повысить производительность сервера.

Допустимые опции каталога (значения директивы `Options`) указаны в табл. 31.2.

Таблица 31.2. Опции каталога

Опция	Описание
None	Запрещены все опции
All	Все опции разрешены
Indexes	Если указана эта опция, при отсутствии файла, заданного <code>DirectoryIndex</code> , будет выведено оглавление каталога. Если <code>Options</code> установлена в <code>None</code> (или <code>Indexes</code> не указана в списке опций), то оглавление каталога выводиться не будет
Includes	Разрешает использование SSI (Server Side Includes)
IncludesNoExec	Более безопасный режим SSI: разрешает SSI, но запрещает запускать из включений внешние программы
ExecCGI	Разрешает выполнение CGI-сценариев
FollowSymLink	Разрешает использование символических ссылок. Довольно опасная опция, поэтому лучше ее не использовать

- ❖ Блочная директива `Limit` позволяет ограничить доступ. Внутри этой директивы можно использовать директивы `order`, `deny` и `allow` (вообще есть еще и директива `require`, но она очень редко используется). Директива `order` задает порядок выполнения директив `deny` и `allow`:

```
# сначала запретить, потом разрешить
order deny, allow

# сначала разрешить, потом запретить
order allow, deny

Директивы allow и deny нужно использовать так:
# запрещаем доступ всем
deny from all

# разрешаем доступ только нашей сети
allow from firma.ru
```

Пример использования директив `Directory` и `Limit` представлен в листинге 31.2.

Листинг 31.2. Фрагмент файла конфигурации Apache

```
<Directory />
AllowOverride None
```

```
Options None
<Limit>
    order deny, allow
    # запрещаем доступ всем
    deny from all
    # разрешаем доступ только нашей сети
    allow from firma.ru
</Limit>
</Directory>
```

В качестве параметра директиве `Limit` можно передать метод передачи данных (`GET`, `POST`), например:

```
<Limit GET>
<Limit POST>
```

Теперь обратимся к блочным директивам `Location` и `Files`.

- ❖ Директива `Location` очень похожа на директиву `Directory`. Только если `Directory` ограничивает доступ к каталогу, то `Location` предназначена для ограничения доступа к отдельным URL сервера:

```
<Location URL>
    директивы ограничения доступа
</Location>
```

К директивам ограничения доступа относятся `order`, `deny`, `allow`.

- ❖ Директива `Files` предназначена для ограничения доступа к отдельным файлам:

```
<Files файл>
    директивы ограничения доступа
</Files>
```

Вы можете указать как отдельный файл, так и регулярное выражение, которому должны соответствовать файлы:

```
# запрещаем доступ к файлу privat.html всем, кроме нашей сети
<Files privat.html>
    Order deny, allow
    Deny from all
    Allow from firma.ru
</Files>

# запрещаем доступ к файлам .ht* всем
<Files ~ "\.ht">
    Order allow, deny
    Deny from all
</Files>
```

Мы рассмотрели все самые полезные директивы конфигурационного файла Apache. Напомню, что директивы, непосредственно влияющие на производительность сервера, будут рассмотрены в *разд. 31.6*.

31.5. Управление запуском сервера Apache

Для управления Web-сервером можно использовать команду `service`:

```
# service httpd start - запуск сервера
# service httpd stop  - останов сервера
# service httpd restart - перезапуск сервера
```

Понятно, что Web-сервер запускается автоматически, поэтому каждый день вам не придется вводить команду `service httpd start`.

В новых версиях Ubuntu есть команда `serice`, поэтому управлять Apache можно, как показано ранее. В старых версиях Ubuntu и Debian нет команды `service`, поэтому управлять Apache можно так:

```
sudo /etc/init.d/apache2 start
sudo /etc/init.d/apache2 stop
sudo /etc/init.d/apache2 restart
```

ПРИМЕЧАНИЕ

Напомню, что в разных дистрибутивах сервис Apache называется по-разному: или `apache2`, или `httpd`.

31.6. Пользовательские каталоги

Если вы когда-нибудь настраивали сервер Apache, то наверняка знакомы с директивой `UserDir`. Я специально ее не описал в табл. 31.1, потому что она заслуживает отдельного разговора.

По умолчанию директива `UserDir` отключена:

```
UserDir disabled
```

Включить ее можно, указав вместо `disabled` любое другое значение, обычно указывается значение `public_html`:

```
UserDir public_html
```

Затем в пользовательском каталоге `/home/<имя>` создается каталог `public_html`, в него помещаются HTML/PHP-файлы персонального сайта пользователя. Обращение к сайту пользователя происходит по URL:

`http://имя_сервера/~имя_пользователя`

Например, если при включенной директиве `UserDir` вы поместили в каталог `/home/den/public_html` файл `report.xml`, то обратиться к нему можно по адресу:

`http://server/~den/report.xml`

Недавно настраивая сервер на базе openSUSE, столкнулся с небольшой проблемой. Ранее, во времена огромного конфигурационного файла, достаточно было раскомментировать эту директиву в конфигурационном файле. Сейчас, когда конфигурация сервера состоит из нескольких небольших файлов, добавление этой опции в основной конфигурационный файл не привело ни к каким изменениям. Оказалось, опцию `UserDir` нужно добавить (точнее, просто раскомментировать) в файл `/etc/apache2/mod_userdir.conf`. Затем нужно добавить следующую строку в самый конец файла `/etc/apache2/default-server.conf`:

```
Include /etc/apache2/mod_userdir.conf
```

После всего этого нужно перезапустить сервер.

31.7. Установка сервера баз данных MySQL

31.7.1. Установка сервера

Для организации связки Apache + PHP + MySQL нам осталось установить последний компонент — сервер баз данных MySQL. Для установки MySQL-сервера установите следующие пакеты:

- ◆ `mysql-server-5.0`;
- ◆ `mysql-client-5.0`;
- ◆ `mysql-admin`.

Первый пакет содержит последнюю версию MySQL-сервера (на данный момент это пятая версия), во втором пакете находится MySQL-клиент, т. е. программа, которая будет подключаться к MySQL-серверу, передавать ему SQL-запросы и отображать результат их выполнения. Третий пакет содержит программу для администрирования MySQL-сервера. Все необходимые дополнительные пакеты будут установлены автоматически.

ПРИМЕЧАНИЕ

Подробно о сервере MySQL мы поговорим в *главе 43*, а сейчас лишь рассмотрим, как связать вместе Apache, PHP и MySQL.

31.7.2. Изменение пароля root и добавление пользователей

Сразу после установки пакетов введите следующие команды:

```
# mysql_install_db
# mysqladmin -u root password ваш_пароль
```

ПРИМЕЧАНИЕ

В процессе выполнения команды `mysql_install_db` вы можете получить сообщение:

[ERROR] /usr/libexec/mysqld: Can't find file: './mysql/help_relation.frm' (errno: 13)

Поможет команда `chown -R mysql /var/lib/mysql`. После ее выполнения нужно заново выполнить команду `mysql_install_db`.

Первая команда (`# mysql_install_db`) создаст необходимые таблицы привилегий, а вторая (`# mysqladmin -u root password ваш_пароль`) — задаст пароль пользователя `root` для сервера MySQL. Этот пароль вы будете использовать для администрирования сервера (данный пароль может и должен отличаться от того, который вы используете для входа в систему). Для обычной работы с сервером рекомендуется создать обычного пользователя. Для этого введите команду:

```
mysql -u root -p mysql
```

Программа `mysql` является клиентом MySQL-сервера. В данном случае она должна подключиться к базе данных `mysql` (служебная база данных), используя имя пользователя `root` (`-u root`). Поскольку вы только что указали пароль для пользователя `root` (до этого пароль для `root` не был задан), вам нужно указать параметр `-p`. После того как программа `mysql` подключится к серверу, вы увидите приглашение программы. В ответ на него нужно ввести следующий SQL-оператор:

```
insert into user(Host, User, Password, Select_priv, Insert_priv, Update_priv,
Delete_priv)
values ('%', 'username', password('123456'), 'Y', 'Y', 'Y', 'Y');
```

Этим оператором мы создали пользователя с именем `username` и паролем `123456`. Данный пользователь имеет право использовать SQL-операторы `select` (выборка из таблицы), `insert` (добавление новой записи в таблицу), `update` (обновление записи), `delete` (удаление записи). Если вам нужно, чтобы ваш пользователь имел право создавать и удалять таблицы, тогда добавьте привилегии `Create_priv` и `Drop_priv`:

```
insert into user(Host, User, Password, Select_priv, Insert_priv, Update_priv,
Delete_priv, Create_priv, Drop_priv)
values ('%', 'username', password('123456'), 'Y', 'Y', 'Y', 'Y', 'Y', 'Y');
```

СОВЕТ

Приведенный здесь SQL-оператор можно записать в одну строку, можно разбить на несколько строк — как вам будет удобно. Но в конце каждого SQL-оператора должна быть точка с запятой! Помните об этом.

Для выхода из программы `mysql` нужно ввести команду `quit`.

Кроме программы `mysql`, в состав MySQL-клиента входит одна очень полезная программа — `mysqlshow`, которая может вывести список таблиц, находящихся в той или иной базе данных. Кроме этого, она еще много чего может, но в данный момент нам нужен пока список таблиц — чтобы вы знали, какие таблицы есть в базе данных:

```
mysqlshow -p <база данных>
```

31.7.3. Запуск и останов сервера

Для управления MySQL-сервером используется программа `/etc/init.d/mysql`. Чтобы запустить сервер, нужно передать этой программе параметр `start`, для останова — `stop`, а для перезапуска — `restart`:

```
sudo /etc/init.d/mysql start
sudo /etc/init.d/mysql stop
sudo /etc/init.d/mysql restart
```

В Mandriva/Fedora можно воспользоваться командой `service`:

```
# service mysql start
# service mysql stop
# service mysql restart
```

Также для управления сервером можно использовать программу `mysqladmin`, узнать больше о ней можно с помощью команды:

```
man mysqladmin
```

31.7.4. Программа MySQL Administrator

При установке сервера мы установили программу MySQL Administrator (пакет `mysql-admin`). Запустите программу командой меню **Приложения | Программирование | MySQL Administrator**. Укажите адрес сервера `localhost`, имя пользователя — `root`, пароль, который вы указали при установке сервера (рис. 31.5), и нажмите кнопку **Connect**. Далее управлять сервером будет существенно проще (рис. 31.6).



Рис. 31.5. Вход на сервер

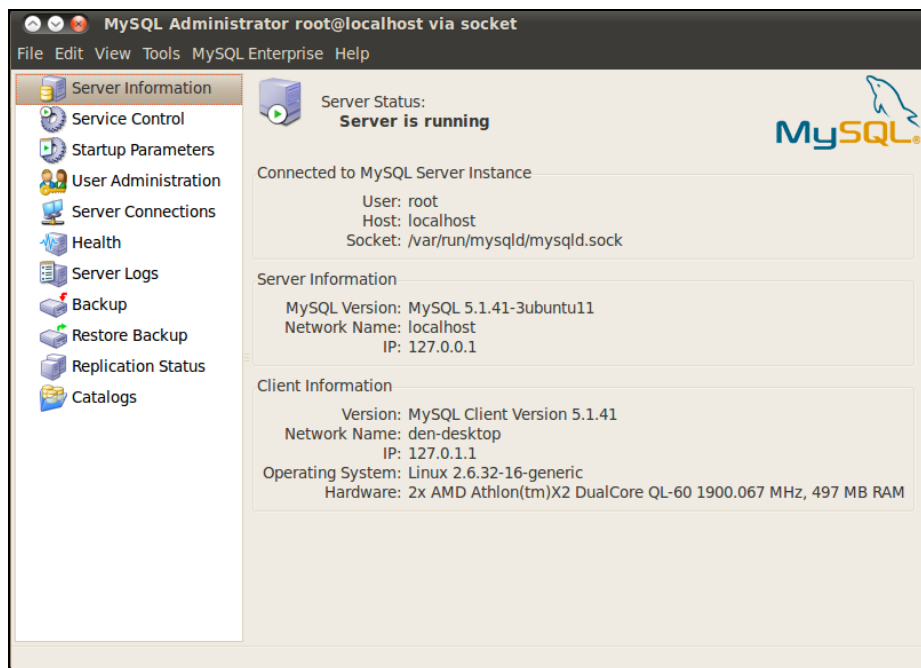


Рис. 31.6. Основное окно программы MySQL Administrator

Пройдемся по основным разделам программы MySQL Administrator:

- ◆ **Server Information** — общая информация о сервере (рис. 31.6);
- ◆ **Service Control** — управление запуском сервиса MySQL (здесь вы можете перезапустить сервер);
- ◆ **Startup Parameters** — параметры, указываемые при запуске сервера;
- ◆ **User Administration** — здесь можно добавить новых пользователей MySQL и установить права пользователей;
- ◆ **Server Connections** — позволяет просмотреть текущие соединения с сервером;
- ◆ **Health** — позволяет видеть состояние вашего сервера в реальном времени (текущая загрузка, использование памяти и т. д.);
- ◆ **Server Logs** — журналы сервера;
- ◆ **Backup** — создать резервную копию сервера;
- ◆ **Restore Backup** — восстановление из резервной копии;
- ◆ **Replication Status** — состояние репликации сервера;
- ◆ **Catalogs** — позволяет просмотреть имеющиеся базы данных и таблицы внутри них.

ГЛАВА 32



FTP-сервер

32.1. Зачем нужен FTP

Сервер FTP (File Transfer Protocol) используется для обмена файлами между системами Интернета. Принцип работы FTP следующий: на FTP-сервере размещается какой-нибудь файл. Пользователи Интернета с помощью FTP-клиента (в любой операционной системе есть стандартный FTP-клиент — программа ftp) подключаются к FTP-серверу и скачивают данный файл.

Права FTP-пользователя определяются администратором FTP-сервера. Одни пользователи могут загружать на сервер файлы в свои личные каталоги, другие имеют полный доступ к FTP-серверу (могут загружать файлы в любые каталоги, как правило, это администраторы FTP-сервера), третьи могут только скачивать публично доступные файлы. Третья группа пользователей — самая большая. Это так называемые анонимные пользователи. Чтобы не создавать учетную запись для каждого анонимного пользователя, все они работают под так называемой анонимной учетной записью, когда вместо имени пользователя указывается имя `anonymous`, а вместо пароля — адрес электронной почты пользователя.

В локальной сети для обмена файлами можно использовать сервер Samba, имитирующий работу рабочей станции под управлением Windows, в Интернете же для обмена файлами нужно использовать только FTP-сервер. С другой стороны, ничего не мешает вам организовать FTP-сервер для обмена файлами внутри локальной сети — это дело вкуса и предпочтений администратора.

Все необходимое для организации FTP-сервера программное обеспечение входит в состав дистрибутива или же бесплатно доступно для скачивания в Интернете. В этой главе мы рассмотрим самый удобный, на мой взгляд, FTP-сервер ProFTPD. Это не единственный FTP-сервер для Linux, например есть еще `wu-ftp`, но ProFTPD является одним из самых защищенных и удобных в настройке.

32.2. Установка FTP-сервера

Для установки FTP-сервера нужно установить пакет `proftpd` (рис. 32.1). Можно также установить конфигуратор `gproftpd`, если он доступен в вашем дистрибутиве.

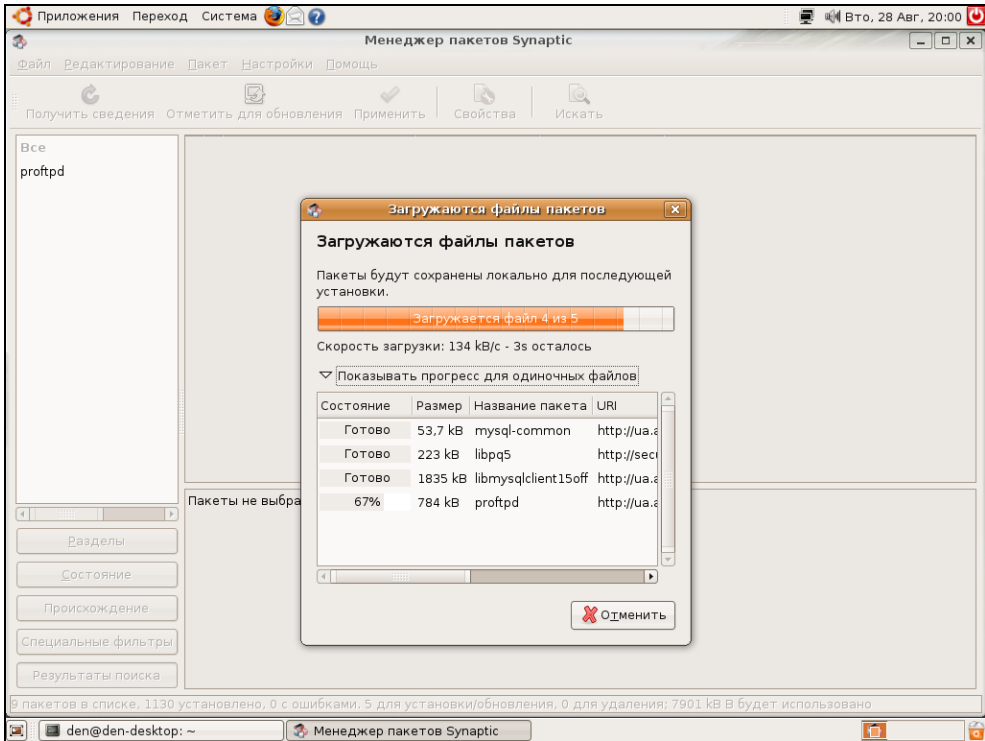


Рис. 32.1. Установка ProFTPD в Ubuntu

Для запуска и останова сервера можно использовать команду `service`:

```
service proftpd start
service proftpd stop
```

32.3. Конфигурационный файл

Основной конфигурационный файл сервера ProFTPD называется `/etc/proftpd/proftpd.conf`. В листинге 32.1 представлен его простейший пример.

Листинг 32.1. Пример файла конфигурации `/etc/proftpd/proftpd.conf`

```
# Подключаем файл с модулями
Include /etc/proftpd/modules.conf
```

```

ServerName          "My server"      # можно написать все, что угодно
ServerType          standalone        # автономный
DeferWelcome         off              # вывести приветствие до
                                     # аутентификации

MultilineRFC2228     on               # поддержка RFC2228
DefaultServer        on               # сервер по умолчанию
ShowSymlinks         on               # показывать символические ссылки

# настройка тайм-аутов
TimeoutNoTransfer    600
TimeoutStalled       600
TimeoutIdle          1200

DisplayLogin         welcome.msg      # файл с приветствием
DisplayFirstChdir    .message         # отобразить этот файл при
                                     # каждой смене каталога

# запрещает использовать данное выражение в FTP-командах (все файлы
# (маска *.* ) вы уже не сможете удалить, придется удалять по одиночке!)
DenyFilter           \ *.* /

Port                 21                # стандартный порт

MaxInstances         30                # количество копий proftpd
# пользователь и группа, от имени которых работает proftpd
User                 proftpd
Group                nogroup

Umask                022 022          # см. man umask

AllowOverwrite       on               # можно перезаписывать файлы
# Журналы сервера
TransferLog /var/log/proftpd/xferlog
SystemLog /var/log/proftpd/proftpd.log

# Параметры подключаемых модулей. Изменять не нужно
<IfModule mod_tls.c>
TLSEngine off
</IfModule>

<IfModule mod_quota.c>
QuotaEngine on
</IfModule>

```

```

<IfModule mod_ratio.c>
Ratios on
</IfModule>

<IfModule mod_delay.c>
DelayEngine on
</IfModule>

<IfModule mod_ctrls.c>
ControlsEngine          on
ControlsMaxClients      2
ControlsLog              /var/log/proftpd/controls.log
ControlsInterval        5
ControlsSocket           /var/run/proftpd/proftpd.sock
</IfModule>
<IfModule mod_ctrls_admin.c>
AdminControlsEngine on
</IfModule>

```

В конфигурационном файле `proftpd.conf` вы можете использовать обычные директивы, задающие одиночные свойства, и блочные директивы, определяющие группы свойств (параметров). Например, директива `ServerName` — обычная, она задает одно свойство, а директива `Directory` — блочная, позволяющая задать несколько параметров для одного каталога.

С директивами файла конфигурации можно ознакомиться в табл. 32.1. В этой таблице указаны не все директивы, а только самые полезные. С остальными вы всегда можете ознакомиться, прочитав документацию по ProFTPD.

Таблица 32.1. Директивы файла конфигурации *proftpd.conf*

Директива	Описание
<code>AccessGrantMsg</code> "сообщение"	Задаёт сообщение, которое будет отправлено пользователю при его регистрации на сервере. Можно задать грозное сообщение, напоминающее о том, что попытка несанкционированного доступа карается статьёй такой-то уголовного кодекса
<code>Allow from all узел сеть [, узел сеть[, ...]]</code>	Данная директива может использоваться только в блоке <code>Limit</code> . Директива разрешает доступ к серверу. По умолчанию используется значение <code>all</code> , которое разрешает доступ к серверу всем узлам со всех сетей
<code>AllowAll</code>	Разрешает доступ всем. Может использоваться в блоках <code>Directory</code> , <code>Anonymous</code> , <code>Limit</code>

Таблица 32.1 (продолжение)

Директива	Описание
AllowForeignAddress On Off	Разрешает узлу при подключении к серверу указывать адрес, не принадлежащий ему. По умолчанию используется значение <code>off</code> (т. е. доступ запрещен), рекомендуется не изменять его. Директива может использоваться в блоках <code>Anonymous</code> , <code><Global></code>
AllowGroup <i>список_групп</i>	Разрешает доступ к серверу указанным группам пользователей (группы должны быть зарегистрированы на этом сервере)
AllowOverwrite On Off	Разрешает (<code>On</code>) перезаписывать существующие файлы
AllowUser <i>список_пользователей</i>	Разрешает доступ к серверу указанным группам пользователей (пользователи должны быть зарегистрированы на этом сервере)
Anonymous <i>каталог</i>	Разрешает анонимный доступ к указанному каталогу. Указанный каталог будет корневым каталогом анонимного FTP-сервера
AuthGroupFile <i>файл</i>	Задаёт альтернативный файл групп. По умолчанию <code>/etc/group</code>
AuthUserFile <i>файл</i>	Задаёт альтернативный файл паролей. По умолчанию <code>/etc/passwd</code>
Bind <i>IP-адрес</i>	Выполняет привязку дополнительного адреса к FTP-серверу
DeferWelcome On Off	Вывести приветствие после аутентификации (<code>On</code>) или до нее (<code>Off</code>)
Deny from all <i>узел</i> <i>сеть</i>	Директива запрещает доступ к FTP-серверу. Используется в блоке <code>Limit</code>
DenyAll	Запрещает доступ всем к объектам, указанным в <code>Directory</code> , <code>Anonymous</code> , <code>Limit</code>
DenyUser <i>список_пользователей</i>	Запрещает доступ указанным пользователям
DefaultRoot <i>каталог</i>	Определяет корневой каталог FTP-сервера. В качестве значения этого параметра полезно указать значение <code>~</code> , тогда в качестве корневого каталога будет использоваться домашний каталог пользователя, который зашел на сервер
DisplayLogin <i>файл</i>	Указанный текстовый файл будет отображен, когда пользователь зайдет на сервер
DisplayFirstChdir <i>файл</i>	Отобразить указанный файл при каждой смене каталога

Таблица 32.1 (окончание)

Директива	Описание
Directory <i>каталог</i>	Задает параметры доступа к каталогу и его подкаталогам
Global	Задает глобальные параметры FTP-сервера
Limit <i>команда</i>	Накладывает ограничение на выполнение FTP-команд, например READ, WRITE, STOR, LOGIN
MaxClients <i>число сообщение</i>	Максимальное количество одновременно работающих клиентов. Если указанное число будет превышено, FTP-сервер отобразит указанное сообщение
MaxLoginAttempts	Максимальное количество попыток регистрации на сервере. По умолчанию 3. Указывается в блоке Global
MaxInstances	Максимальное количество одновременно работающих экземпляров демона proftpd
ServerType <i>тип</i>	Задает тип запуска сервера. Значение по умолчанию — standalone (автономный запуск). Не нужно его изменять
ServerName <i>"имя"</i>	Задает имя сервера. Можете написать все, что угодно, например My server
ServerAdmin <i>e-mail</i>	Позволяет указать адрес электронной почты администратора сервера
ShowSymlinks <i>On Off</i>	Показывать символические ссылки (On) или сразу результирующие файлы (Off)
Order allow, deny deny, allow	Задает порядок выполнения директив Allow и Deny в блоке Limit
TimeoutIdle <i>секунды</i>	Определяет тайм-аут простоя. Если пользователь не проявит активности за указанное время, соединение будет разорвано. По умолчанию используется значение 60
TimeoutNoTransfer <i>секунды</i>	Тайм-аут начала передачи. Сколько времени нужно ждать до разъединения, если пользователь вошел, но не начал передачу
TimeoutStalled <i>секунды</i>	"Замирание" во время передачи файла. Бывает такое, что клиент начал передачу (или прием) файла, но связь оборвалась. Этот тайм-аут определяет, сколько нужно ждать до разъединения в такой ситуации. Данный тайм-аут нужен, потому что бывает другая ситуация — когда у пользователя очень медленный канал
Umask <i>маска</i>	Задает права доступа для созданного файла
User <i>имя_пользователя</i>	Пользователь, от имени которого работает демон ProFTPD

32.4. Настройка реального сервера

В этом разделе мы настроим реальный FTP-сервер, к которому смогут получить доступ как обычные (зарегистрированные) пользователи, так и анонимные.

Приведенная в листинге 32.1 конфигурация вполне работоспособна и может использоваться для создания обычного (не анонимного) FTP-сервера. Но в конфигурационный файл нужно добавить две директивы:

```
DefaultRoot      ~
MaxClients       20 "Server is full!!!"
```

Первая директива делает корневым домашний каталог пользователя (т. е. пользователь не может выйти за пределы своего домашнего каталога, следовательно, не может навредить системе, если администратор неправильно установил права к каким-нибудь системным каталогам).

Вторая директива ограничивает число одновременно работающих клиентов во избежание перегрузки сервера.

Остальные параметры вы можете задать по своему усмотрению. Рассмотрим несколько примеров использования блоков `Directory` и `Login`:

```
<Directory upload>
  <Limit READ>
    DenyAll
  </Limit>
  <Limit WRITE>
    AllowAll
  </Limit>
</Directory>
```

Директива `Directory` определяет две директивы `Limit` для каталога `upload`. Первая запрещает всем читать этот каталог, а вторая — разрешает всем записывать новые файлы в этот каталог. Каталог `upload`, таким образом, полностью оправдывает свое название — только для загрузки файлов.

Рассмотрим еще один пример, запрещающий доступ к серверу всех узлов из подсети 192.168.1.0:

```
<Limit LOGIN>
  DenyAll
  Deny from 192.168.1.
</Limit>
```

Если нужно, наоборот, разрешить доступ к серверу только пользователей из сети 192.168.1.0, то нужно использовать следующий блок `Limit`:

```
<Limit LOGIN>
  Order deny, allow          # порядок действия deny-allow
  DenyAll                   # запрещаем доступ всем
  Allow from 192.168.1.      # разрешаем доступ только из сети 192.168.1.0
</Limit>
```

Теперь перейдем к анонимного доступа. Для организации анонимного доступа нужно добавить в файл конфигурации следующую директиву `Anonymous`:

```
<Anonymous ~ftp>
```

```
User                                ftp
Group                              nogroup
```

```
# Определяем псевдоним "anonymous" для пользователя "ftp"
```

```
# Клиенты смогут войти под обоими именами
```

```
UserAlias                          anonymous ftp
```

```
# Все файлы принадлежат пользователю ftp
```

```
DirFakeUser    on ftp
```

```
DirFakeGroup   on ftp
```

```
# Не нужно требовать "правильную" оболочку
```

```
# "Правильной" считается оболочка, указанная в /etc/shells
```

```
RequireValidShell off
```

```
# Максимальное число анонимных пользователей
```

```
MaxClients      10
```

```
# Файлы с сообщениями
```

```
DisplayLogin      welcome.msg
```

```
DisplayFirstChdir .message
```

```
# Ограничим WRITE для анонимных пользователей
```

```
  <Directory *>
```

```
    <Limit WRITE>
```

```
      DenyAll
```

```
    </Limit>
```

```
  </Directory>
```

```
</Anonymous>
```

32.5. Программы `ftpwho` и `ftpcount`

Вспомогательные программы `ftpwho` и `ftpcount` помогут администратору FTP-сервера определить, какие пользователи в данный момент зарегистрированы на сервере (`ftpwho`), и узнать общее число зарегистрированных на сервере в данный момент пользователей (`ftpcount`). Вывод обеих программ показан на рис. 32.2.

```
den@den-desktop:~$ ftpwho
standalone FTP daemon [5176], up for 37 min
 7378 den      [ 0m10s]   0m7s idle
Service class                -    1 user
den@den-desktop:~$ ftpcount
Master proftpd process 5176:
Service class                -    1 user
den@den-desktop:~$ █
```

Рис. 32.2. Программы ftpwho и ftpcount

32.6. Конфигуратор gproftpd

Графический конфигуратор gproftpd (рис. 32.3) позволяет быстро и комфортно настроить FTP-сервер. С этим конфигуратором вы разберетесь и без моих комментариев: там все очень просто, особенно сейчас, когда вы знаете назначение основных директив сервера. Но не следует забывать, что это всего лишь конфигуратор, который может помочь настроить только базовые возможности сервера.

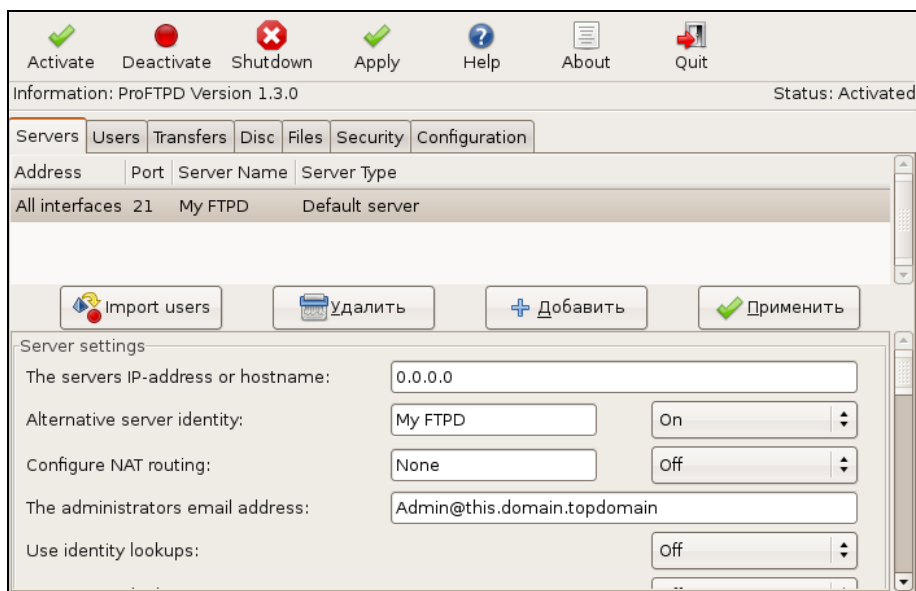


Рис. 32.3. Конфигуратор gproftpd

ГЛАВА 33



Почтовый сервер

33.1. Что такое Qmail

Почтовый сервис состоит из агента отправки почты (MTA, Mail Transfer Agent), реализующего протокол SMTP (Simple Mail Transfer Protocol), и сервиса POP3 (Post Office Protocol v3), который используется для приема почты. Вы можете установить любой POP3-сервис, например `cyrus-pop3d`, и любой MTA-агент, например `sendmail` или `postfix`. "Пошаманить" над конфигурацией этих сервисов и получить работающий почтовый сервер. Один сервис будет принимать почту для ваших пользователей, а другой — отправлять ее.

Если вы установите и настроите Qmail, то получите два средства в одном флаконе: Qmail заменит и MTA, и POP3-сервис. Это намного проще и удобнее, особенно при настройке почтового сервера. Судите сами: если у вас два разных сервиса, тогда для правильной их настройки нужно прочитать в два раза больше документации. Для правильной настройки Qmail вполне хватит документации на сайте производителя, ну и, конечно же, этой книги.

Qmail — это очень гибкий и очень мощный почтовый сервер, но у него есть одна особенность — он не совсем простой в настройке. Я не сказал, что он сложный, но и не нужно думать, что вы за пять минут его настроите. Если вам нужно поднять почтовый сервер за пять минут, тогда установите `cyrus-pop3d` и `postfix`. Правда, знайте, что и работать такой сервер будет как "настроенный за 5 минут".

33.2. Подготовка к установке Qmail

Первым делом поговорим о необходимом дисковом пространстве. В корневой файловой системе должно быть не менее 70 Мбайт свободного места, а в файловой

системе, примонтированной к /var, должно быть не менее 20 Мбайт доступного дискового пространства. Всего нужно чуть меньше 100 Мбайт, но помните, что также нужно дисковое пространство для писем пользователей, а тут рекомендация одна: чем больше места, тем лучше.

Перед установкой Qmail нужно удалить все SMTP и POP3-сервисы: postfix, sendmail, Cyrus, Qpopper и др. Это нужно для того, чтобы не возник конфликт между этими сервисами и Qmail.

Также перед установкой Qmail нужно отключить SELinux. Как это сделать, вы сможете прочитать на моем сайте:

<http://www.dkws.org.ua/phpbb2/viewtopic.php?p=29972>.

Теперь рассмотрим список программного обеспечения, которое должно быть установлено до Qmail:

- ◆ Web-сервер Apache (версия сервера 1.3 или 2.x, роли не играет);
- ◆ интерпретатор PHP с поддержкой imap и mysql (нужно дополнительно установить пакеты php-imap и php-mysql);
- ◆ интерпретатор Perl (версия 5.x);
- ◆ сервер MySQL (подойдут версии 3.x и 4.x);
- ◆ библиотека OpenSSL, а также файлы для разработчиков — OpenSSL-devel (пакеты называются openssl и openssl-devel);
- ◆ программа wget (нужна для загрузки файлов из Интернета);
- ◆ пакеты patch и patchutils (нужны для накладки патчей).

Кроме этого, вам нужно установить следующие модули Perl:

```
Digest::SHA1
Digest::HMAC
Net::DNS
Time::HiRes
HTML::Tagset
HTML::Parser
```

Давайте сделаем это прямо сейчас. Подключитесь к Интернету и введите команду:

```
perl -MCPAN -e shell
```

Затем введите команду:

```
install ИМЯ_МОДУЛЯ
```

Данную команду нужно ввести для каждого устанавливаемого модуля. Подробнее об установке модулей Perl можно прочитать по адресу:

<http://www.dkws.org.ua/phpbb2/viewtopic.php?topic=3035>.

Вам также придется перенастроить свой брандмауэр. Нужны порты, указанные в табл. 33.1.

Таблица 33.1. Порты, которые необходимо открыть

Входящие TCP-соединения	Исходящие TCP-соединения
25 — SMTP	25 — SMTP
80 — HTTP	110 — POP
110 — POP	143 — IMAP
143 — IMAP	783 — Spamassassin
443 — HTTPS	993 — IMAPS
783 — Spamassassin (защита от спама)	
993 — IMAPS	

33.3. Установка Qmail и необходимых дополнений

33.3.1. Загрузка и установка Qmail

Сначала создадим каталог, в который мы загрузим исходные коды Qmail. Да, установку будем производить из исходных кодов, а не из RPM-пакета:

```
# mkdir /qmail
# cd /qmail
```

Скачайте файл `qmailrocks.tar.gz` с сайта **qmailrocks.ru**:

```
# wget http://www.qmailrocks.ru/downloads/qmailrocks.tar.gz
```

Затем распакуйте его:

```
# tar zxvf qmailrocks.tar.gz
```

После этого запустите сценарий `qmr_install_linux-s1.script`:

```
# /qmail/qmailrocks/scripts/install/qmr_install_linux-s1.script
```

Данный сценарий создаст пользователей и группы, необходимые каталоги и установит права доступа к ним.

После этого нужно "пропатчить" Qmail. В исходные коды Qmail будет в общем добавлено 15 патчей, которые добавляют новые возможности к Qmail. Особо вникать в эти патчи не будем, а просто введем команду:

```
/qmail/qmailrocks/scripts/util/qmail_big_patches.script
```

Данный сценарий добавит все 15 патчей, поэтому вам не нужно устанавливать каждый патч отдельно.

Следующий шаг — это компиляция Qmail. Для этого введите следующие команды:

```
# cd /usr/src/qmail/qmail-1.03
# make man && make setup check
# ./config-fast полное_имя_узла
```

В качестве параметра последней команды вы должны указать полное доменное имя почтового сервера, например

```
# ./config-fast mail.firma.ru
```

Теперь у нас есть установленный Qmail. Сразу после этого нужно сгенерировать сертификаты, которые используются для шифрования SMTP-сессии:

```
# make cert
```

Данная команда задаст вам несколько вопросов, вроде вашего местоположения, названия фирмы и т. д. Сгенерированный сертификат будет помещен в каталог `/var/qmail/control/`. Файл сертификата называется `servercert.pem`. Файл `clientcert.pem` — это ссылка на файл `servercert.pem`.

Сразу после создания сертификатов нужно установить права доступа к нему:

```
# cd /var/qmail/control/
```

```
# chown -R vpopmail:qmail clientcert.pem servercert.pem
```

33.3.2. Установка `ucspi-tcp` и `daemontools`

Вот теперь можно приступить к сборке `ucspi-tcp` (используется для создания клиент-серверных приложений, работающих по протоколу TCP) и `daemontools` (не путать с эмулятором виртуального CD-ROM в Windows!):

```
# cd /usr/src/qmail/ucspi-tcp-0.88/
```

```
# patch < /qmail/qmailrocks/patches/ucspi-tcp-0.88.errno.patch
```

```
# make && make setup check
```

```
# cd /package/admin/daemontools-0.76/src
```

```
# patch < /qmail/qmailrocks/patches/daemontools-0.76.errno.patch
```

```
# cd /package/admin/daemontools-0.76
```

```
# package/install
```

После этого в вашей системе появится работающий сервис `svscanboot`.

33.3.3. Установка `EZmlm` — средства для создания рассылки

Вы планируете создавать списки рассылки? Если да, тогда вам нужно установить дополнение `EZmlm`, которое будет интегрировано в утилиту управления `Qmailadmin`. Для установки `EZmlm` введите команды:

```
# cd /qmail/qmailrocks/
```

```
# tar zxvf ezmlm-0.53-idx-0.41.tar.gz
```

```
# cd ezmlm-0.53-idx-0.41
```

```
# make && make setup
```

Если в ответ "тишина", значит, все прошло успешно. А вот если произошли ошибки, тогда вы увидите их описание.

33.3.4. Установка Autoresponder — автоответчика

Autoresponder позволяет настраивать автоответчики для почтовых ящиков. Для его установки введите команды:

```
# cd /qmail/qmailrocks
# tar zxvf autorespond-2.0.5.tar.gz
# cd autorespond-2.0.5
#make && make install
```

33.3.5. Установка MailDrop — фильтра для сообщений

MailDrop — фильтр сообщений, приходящий на почтовый сервер. Подробно мы его обсуждать не будем.

Мы рассмотрим только его установку, а с документацией по MailDrop вы сможете ознакомиться по адресу: <http://www.courier-mta.org/maildrop/>.

Итак, для установки MailDrop нужно ввести следующие команды:

```
# cd /qmail/qmailrocks
# tar zxvf maildrop-1.6.3.tar.gz
# cd maildrop-1.6.3
# ./configure --prefix=/usr/local --exec-prefix=/usr/local --enable-maildrop-uid=root --enable-maildrop-gid=vchkpw --enable-maildirquota
# make && make install-strip && make install-man
```

33.3.6. Установка QmailAdmin — Web-интерфейса для настройки Qmail

У Qmail нет обычного конфигуратора, зато есть конфигуратор с Web-интерфейсом, что еще удобнее, поскольку вы сможете настраивать Qmail с любого компьютера вашей сети (и не только с вашей — при должной настройке брандмауэра и Apache).

Для установки Web-интерфейса QmailAdmin введите следующие команды:

```
# cd /qmail/qmailrocks
# tar zxvf qmailadmin-1.2.9.tar.gz
# cd qmailadmin-1.2.9
# ./configure --enable-autoresponder-path=/usr/local/bin --enable-cgibindir=/путь_к_cgi-bin --enable-htmldir=/путь_к_каталогу/html
# make && make install-strip
```

Обратите внимание: вы должны указать путь к каталогам cgi-bin и html вашего Web-сервера. Обычно это /var/www/cgi-bin и /var/www/html.

После этого откройте браузер и введите следующий URL:

```
http://localhost/cgi-bin/qmailadmin
```

Вы увидите форму для ввода имени пользователя и пароля. Вы должны войти под пользователем postmaster (соответственно, указав пароль этого пользователя, а не пользователя root). После этого вы можете настраивать Qmail с помощью QmailAdmin — это действительно очень легко.

33.4. Настройка после установки и запуск Qmail

Почти все готово к запуску вашего почтового сервера. Осталось совсем немного. Вам нужно создать сценарий для запуска и управления Qmail. Для этого запустите следующий сценарий, который автоматизирует эту задачу:

```
# /qmail/qmailrocks/scripts/finalize/linux/finalize_linux.script
```

После этого откройте в любом текстовом редакторе файл `/var/qmail/supervise/qmail-pop3d/run`. Найдите в нем строку `mail.example.com` и замените ее именем вашего почтового сервера, например `mail.firma.ru`, после чего сохраните файл.

Теперь введите команду:

```
# qmailctl stop
```

Как вы уже догадались, данная команда завершает все запущенные процессы Qmail. Нужно установить релей (relay) для локальной машины:

```
# echo '127.:allow,RELAYCLIENT=""' >> /etc/tcp.smtp
```

```
# qmailctl cdb
```

Теперь нужно установить адрес электронной почты, на который должны приходить системные сообщения:

```
# echo postmaster@firma.ru > /var/qmail/alias/.qmail-root
```

```
# echo postmaster@firma.ru > /var/qmail/alias/.qmail-postmaster
```

```
# echo postmaster@firma.ru > /var/qmail/alias/.qmail-mailer-daemon
```

Понятно, что `firma.ru` нужно заменить именем вашего домена.

Вам осталось ввести всего две команды:

```
# ln -s /var/qmail/alias/.qmail-root /var/qmail/alias/.qmail-anonymous
```

```
# chmod 644 /var/qmail/alias/.qmail*
```

После этого у вас есть полноценный почтовый сервер. Перед запуском Qmail нужно убедиться, что мы все сделали правильно. Запустите следующий сценарий для проверки вашей конфигурации:

```
# /qmail/qmailrocks/scripts/util/qmr_inst_check
```

Если вы в ответ получите "congratulations", вы все сделали правильно, поздравляю!

А вот если будут сообщения об ошибках, внимательно прочитайте их, исправьте и запустите сценарий проверки конфигурации снова.

Для управления Qmail используется программа `qmailctl`:

```
# qmailctl start      - запуск Qmail
# qmailctl stop       - запуск Qmail
# qmailctl stat       - вывод статистики
```

Запустите Qmail:

```
# qmailctl start
```

Сейчас проверим его работоспособность с помощью `telnet`:

```
# telnet localhost 25
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
220 somewhere.anywhere.com ESMTP
```

ehlo localhost

```
250-somewhere.anywhere.com
250-AUTH LOGIN CRAM-MD5 PLAIN
250-AUTH=LOGIN CRAM-MD5 PLAIN
250- STARTTLS
250-PIPELINING
250 8BITMIME
```

starttls

```
220 ready for tls
```

quit

quit

```
Connection closed by foreign host.
```

Жирным шрифтом выделены команды, которые вы должны ввести в `telnet`-сессии. Если вы в ответ на команду `starttls` увидели сообщение "454 TLS not available: missing RSA private key (#4.3.0)", проверьте, есть ли в каталоге `/var/qmail/control/` файл `servercert.pem`. Если его нет, нужно сгенерировать сертификат заново.

Если же файл `/var/qmail/control/servercert.pem` существует, установите для него права:

```
chown vpopmail:qmail /var/qmail/control/servercert.pem
```

Теперь проверим работу POP3:

```
# telnet localhost 110
Trying 127.0.0.1...
Connected to 127.0.0.1.
Escape character is '^]'.
+OK
```

```
user den@firma.ru
```

```
+OK
```

```
pass den_password
```

```
+OK
```

```
quit
```

```
+OK
```

```
Connection closed by foreign host.
```

Как и в прошлом случае, команды, которые вы должны ввести, выделены. Команда `user` позволяет указать пользователя, к почтовому ящику которого вы хотите подключиться. Пользователь должен быть зарегистрирован на сервере. Команда `pass` задает пароль пользователя (без кавычек). Если ваша POP3-сессия похожа на приведенную, тогда вы все сделали правильно.

33.5. Настройка почтовых клиентов

При настройке почтовых клиентов в качестве имени пользователя нужно указывать полное имя пользователя в формате *имя_пользователя@сервер*, например **den@firma.ru** (рис. 33.1).

Настроенный нами SMTP-сервер требует SMTP-аутентификации для отправки писем, поэтому не забудьте при настройке почтового клиента указать это (рис. 33.2).

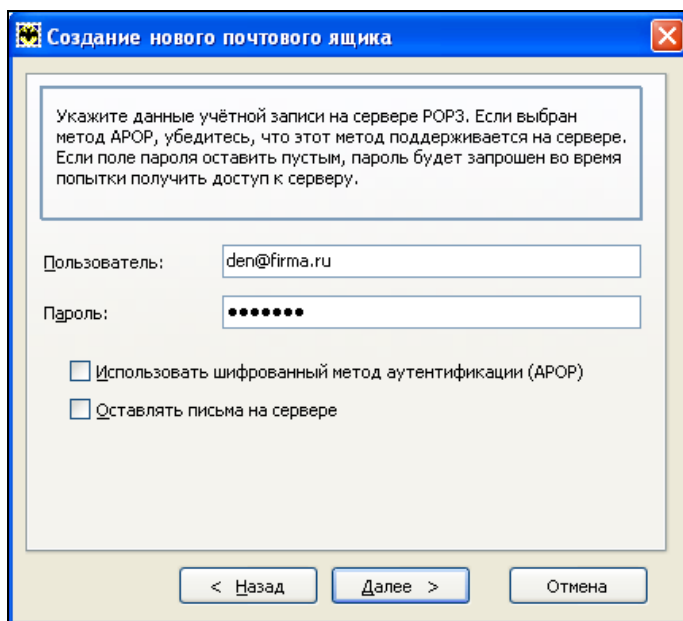


Рис. 33.1. Настройка программы The Bat!

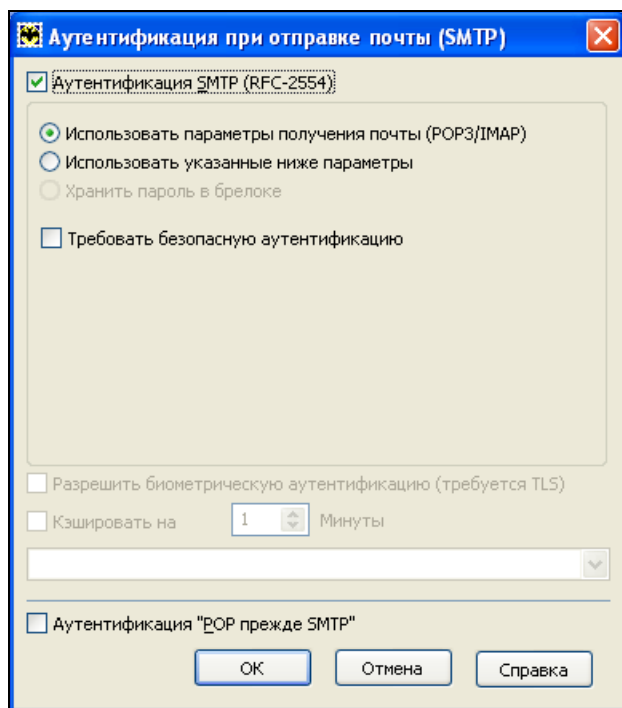


Рис. 33.2. Включение SMTP-аутентификации в The Bat!

33.6. Дополнительная информация

Вам нужно настроить Qmail на FreeBSD или установить фильтр спама? Тогда настоятельно рекомендую посетить сайт <http://www.qmailrocks.ru/>, где вы найдете много информации относительно настройки Qmail и антиспамовых фильтров.

ГЛАВА 34



DNS-сервер

34.1. Еще раз о том, что такое DNS

Система доменных имен (DNS, Domain Name System) используется для преобразования IP-адресов в доменные имена и обратно. Компьютеру намного проще работать с числами, человеку же проще запомнить символьное имя узла, чем его IP-адрес.

Система DNS имеет древовидную иерархическую структуру (рис. 34.1). Список корневых серверов DNS хранится на каждом DNS-сервере (позже мы узнаем, где именно, и как его обновлять).

На рис. 34.1 изображен корень системы DNS, домены первого уровня (.ru, .com, .org) и домен второго уровня (firma). Доменов первого уровня (их еще называют TLD, Top Level Domains) довольно много: com, biz, org, info, gov, net, ws, домены стран (ru, ua, uk, ...) и т. д. Понятно, что доменов второго уровня еще больше, не говоря уже о доменах третьего и последующих уровней.

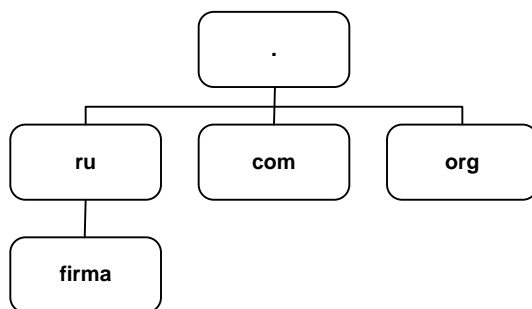


Рис. 34.1. Иерархическая структура DNS

Доменное имя компьютера имеет следующий формат:

[имя_компьютера].[домен_N] [домен.TLD]

Например,

ftp.sales.firma.ru

При запросе к DNS-серверу доменное имя обрабатывается в доменном порядке. Сначала наш DNS-сервер посылает запрос к DNS-серверу домена `ru`: знает ли он что-нибудь о домене `firma`? DNS-сервер домена `ru`, если домен `firma` найден, сообщает IP-адрес сервера DNS домена `firma`. Потом наш DNS-сервер (или наш собственный сервер имен, или же это сервер имен провайдера) обращается к серверу имен домена `firma.ru`. Ему нужно узнать, знает ли он что-то о домене `sales`. Получив IP-адрес DNS-сервера домена `sales.firma.ru`, мы можем к нему обратиться, чтобы получить IP-адрес компьютера с именем `ftp.sales.firma.ru` (очевидно, это FTP-сервер отдела продаж какой-нибудь фирмы).

Приведенная схема разрешения доменного имени называется рекурсивной, а наш запрос — рекурсивным запросом. Конечно, саму схему я немного упростил, но общий смысл должен быть понятен. Понятно также, и что такой запрос занимает довольно много времени и ресурсов, поэтому целесообразно настроить кэширующий сервер DNS, даже если у вас нет собственного домена. Всю грязную работу (т. е. рекурсивные запросы) будут делать серверы DNS провайдера, а наш сервер будет только кэшировать результаты запросов — так можно повысить скорость разрешения доменных имен, следовательно, ускорить работу Интернета в целом. Поэтому кэширующий сервер можно установить не только на шлюзе, но и на домашнем компьютере, где он также будет с успехом выполнять свою функцию.

Настройку сервера DNS мы начнем именно с кэширующего сервера DNS. Во-первых, он настраивается проще, чем полноценный сервер DNS, но зато в процессе его настройки мы познакомимся с основными конфигурационными файлами, и при настройке полноценного DNS-сервера нам будет проще. Во-вторых, не всегда есть необходимость настраивать полноценный DNS-сервер. У вас может быть локальная сеть с выходом в Интернет, но у нее не обязательно должен быть свой собственный домен.

34.2. Кэширующий сервер DNS

Что же такое кэширующий сервер DNS? Наверняка все мы знакомы с так называемыми "ускорителями" Интернета — программами, якобы помогающими сделать Интернет намного быстрее. Второе название этих программ — оптимизаторы Интернета. Как правило, это Windows-программы, которые распространяются за определенную плату в Интернете. Иногда их даже можно скачать бесплатно. В первом случае, если программа распространяется за деньги, "ускоритель" Интернета ничего вообще не делает. Он запускается, пользователь устанавливает параметры, но на самом деле никакого ускорения не происходит. Просто кто-то таким не очень честным образом зарабатывает деньги. Во втором случае, когда программа распространяется бесплатно, также не наблюдается никакого ускорения, а наоборот, падение скорости и повышенный расход трафика. Почему? Да потому что "оптимизаторы" Интернета в большинстве случаев являются вирусами-троянами. Пользователи добровольно устанавливают программу, которая потом передаст

секретную информацию (например, ключи от электронного кошелька) злоумышленнику. Помните, что бесплатный сыр только в мышеловке.

Linux же позволяет организовать настоящий "ускоритель" Интернета. Впрочем, не нужно ожидать, что ваш Интернет будет работать на 70, а то и на все 100% быстрее, как это обещают оптимизаторы-вирусы. Ускорение будет заключаться в установке кэширующего сервера DNS. Установка DNS-сервера позволяет:

- ❖ сократить время разрешения доменных имен, поскольку свой DNS-сервер будет в нашей сети — ответы на запросы о разрешении доменных имен будут приходиться от локального сервера, а не от загруженного DNS-сервера провайдера;
- ❖ немного сэкономить трафик, поскольку локальный трафик не будет учитываться, чего не скажешь о трафике между вами (вашей сетью) и провайдером.

Итак, кэширующий DNS-сервер — дело нужное, поэтому не будем терять времени и приступим к настройке. Установите пакет `bind9`. Обратите внимание, что пакет называется `bind9` (Berkley Internet Nameserver Daemon), а сам сервер — `named`. В старых версиях дистрибутивов данный пакет называется просто `bind`, скорее всего, этот пакет содержит восьмую версию BIND. Для большей точности отмечу, что настройка сервера будет производиться на примере дистрибутива Debian, но в других дистрибутивах процесс настройки должен быть аналогичен при условии использования девятой версии BIND.

После установки пакета `bind9` нужно отредактировать файл `/etc/bind/named.conf` — это основной файл конфигурации `named` (листинг 34.1). Комментарии в оригинальном файле будут, понятно, на английском, но для книги я их перевел на русский язык.

Листинг 34.1. Файл конфигурации `/etc/bind/named.conf`

```
// Это основной конфигурационный файл DNS-сервера BIND
// См. файл /usr/share/doc/bind9/README.Debian.gz для
// получения информации о структуре конфигурационных файлов BIND
// *ДО* изменения этого файла

// Если вам нужно добавить зоны, сделайте это в
// файле /etc/bind/named.conf.local
include "/etc/bind/named.conf.options";

// Зона корневых серверов имен
zone "." {
    type hint;
    file "/etc/bind/db.root";
};

// Локальная зона localhost
zone "localhost" {
```

```
type master;
file "/etc/bind/db.local";
};
zone "127.in-addr.arpa" {
type master;
file "/etc/bind/db.127";
};
zone "0.in-addr.arpa" {
type master;
file "/etc/bind/db.0";
};
zone "255.in-addr.arpa" {
type master;
file "/etc/bind/db.255";
};
include "/etc/bind/named.conf.local";
```

В основном конфигурационном файле прописываются корневая и локальная зоны. Локальная зона служит для преобразования имени localhost в IP-адрес 127.0.0.1 и наоборот. Да, и все. Корневая зона содержит список корневых серверов DNS.

Раньше все, что касалось настройки DNS-сервера — и описание зон, и настройки сервера, хранилось в файле named.conf. Сейчас принято в основном конфигурационном файле хранить только описание корневой и локальной зон. Описание опций выносится в файл /etc/bind/named.conf.options, а описание зон, обслуживаемых сервером, выносится в файл в /etc/bind/named.conf.local.

Вообще-то, собственные зоны вы можете описать в файле named.conf — особой разницы нет. Но если ваш DNS-сервер описывает много зон или одну большую зону (где много компьютеров), тогда целесообразно вынести описание этих зон в named.conf.local — вам так будет удобнее настраивать DNS-сервер.

Рассмотрим файл, содержащий опции DNS-сервера — /etc/bind/named.conf.options (листинг 34.2). Опять-таки оригинальный файл содержит комментарии на английском, а в книге будут комментарии на родном языке.

Листинг 34.2. Файл опций /etc/bind/named.conf.options

```
options {
    directory "/var/cache/bind";

    // Если "между" вашим DNS-сервером и форвард-серверами находится брандмауэр,
    // вы должны его настроить должным образом. О настройке брандмауэра можно
    // прочитать по адресу: http://www.kb.cert.org/vuls/id/800113

    // Здесь прописываются форвард-серверы
    // forwarders {
```

```
//      0.0.0.0;
// };
auth-nxdomain no;      # в соответствии с RFC1035
listen-on-v6 { any; };
};
```

Разберемся что к чему. Основной рабочий каталог — `/var/cache/bind` (задается параметром `directory`).

А вот с этого момента начинается самое интересное. Напомню, что мы сейчас создаем кэширующий сервер, позволяющий ускорить процесс разрешения доменных имен. Но мы можем ускорить работу самого сервера, указав форвард-серверы. В обычном режиме наш сервер сам формирует кэш, но так как сеть у нас относительно небольшая, кэш будет формироваться долго — как долго зависит от количества запросов, поступающих от клиентов сети. А если вы установили кэширующий сервер только для обслуживания своего компьютера, то сначала вообще не почувствуете никакой разницы. Ведь серверу, прежде чем добавить IP-адрес в кэш, нужно его разрешить. Это уже при втором обращении к доменному имени его IP-адрес будет получен из кэша. Так вот, мы можем использовать кэш от форвард-серверов. Как правило, форвард-серверами выступают серверы провайдера. Как правило, у провайдера уже сформирован довольно большой кэш, который мы можем использовать.

Все, что нужно для использования форвард-сервера, — это добавить его IP-адрес в блок `forwarders`:

```
forwarders {
    # Все запросы будут переадресованы к DNS-серверу провайдера 192.168.99.1.
    # Если с этим сервером что-то случится, то локальный сервер
    # попытается найти ответ в своем кэше или обратится к другим
    # DNS-серверам, которые указываются в /etc/resolv.conf
    192.168.99.1;
};
```

Параметр `forwarders` задает заключенный в фигурные скобки список IP-адресов, соответствующих DNS-серверам, которым наш DNS-сервер будет переадресовывать запросы, вместо того чтобы отвечать на них самому. IP-адреса перечисляются через точку с запятой.

Кроме параметра `forwarders` можно использовать параметр `forward`, который может принимать следующие значения:

- ◆ `only` — наш DNS-сервер никогда не должен предпринимать попыток обработать запрос самостоятельно;
- ◆ `first` — наш сервер должен пытаться сам обработать запрос, если указанные далее параметром `forwarders` сервера DNS не были найдены.

Использование параметра `forward` лишено смысла без использования параметра `forwarders`.

Параметр `forward` обычно нужно указывать до параметра `forwarders`:

```
forward first;
forwarders {
    192.168.99.1;
    192.168.99.2;
};
```

Где взять адреса форвард-серверов? Обычно они находятся в `/etc/resolv.conf`.

Вот вроде бы и все. Можно приступить к запуску сервера. Но перед этим отмечу, что так как мы создавали кэширующий сервер, то отсутствует блок `controls{}`. Пустой или отсутствующий блок `controls{}` нужен для того, чтобы `named` не обращал внимания на отсутствие ключа `rndc.key`, который нужен для программы удаленного управления сервером — `rndc`. Правда, это не вполне корректно, поскольку для останова сервера нужно будет использовать команду `killall named`, но для нас это не существенно, поскольку мы не будем часто его останавливать.

Теперь можно запустить ваш сервер имен:

```
sudo /etc/init.d/bind9 start
```

Поскольку сервер может быть запущен (при установке пакета он запускается автоматически), то его нужно перезапустить:

```
sudo /etc/init.d/bind9 restart
```

Если нет ошибок в конфигурационных файлах, вы получите сообщение:

```
* Starting domain name service... bind9 [ OK ]
```

В Fedora/Mandriva/Ubuntu можно использовать команду `service` для управления сервером:

```
# service bind9 start
# service bind9 restart
# service bind9 stop
```

Если у вас восьмая версия BIND, то сервис, скорее всего, будет называться `named`, поэтому команды управления будут такими:

```
# service named start
# service named restart
# service named stop
```

Хотя имя сервиса зависит от используемого дистрибутива и его версии. Например, у меня был установлен BIND версии 9.2.3, но сервис назывался `bind`, вот посмотрите сами вывод команды `tail /var/log/messages`:

```
# tail /var/log/messages
```

```
Aug 8 9:58:16 den named[3140]: starting BIND 9.2.3
```

```
Aug 8 9:58:16 den named[3140]: using 1 CPU
```

```
Aug 8 9:58:16 den named[3140]: loading configuration from
'/etc/bind/named.conf'
```

```
Aug 8 9:58:16 den named[3140]: listening on IPv4 interface lo, 127.0.0.1#53
```

```
Aug 8 9:58:16 den named[3140]: listening on IPv4 interface eth0,
192.168.0.1#53
Aug 8 9:58:16 den named[3140]: zone 0.0.127.in-addr.arpa/IN: loaded serial
1997022700
Aug 8 9:58:16 den named[3140]: running
```

Кстати, данный вывод я привел не просто так: последняя строка свидетельствует о том, что сервер запущен. Первая запись сообщает нам версию BIND, вторая — то, что используется 1 процессор, далее сообщается: используемый конфигурационный файл, прослушиваемые интерфейсы (lo и eth0) и порт — 53, а также загруженная локальная зона. Число в квадратных скобках (3140) — это PID процесса (идентификатор процесса), "убить" процесс в данном случае можно так:

```
# kill 3140
```

Проверить, работает ли сервер, можно и другим способом, например:

```
# ps -ax | grep named
# ps -ax | grep bind9
```

Теперь осталось в файле /etc/resolv.conf прописать IP-адрес собственного сервера DNS. То же самое нужно сделать на всех остальных компьютерах сети:

```
domain firma.ru
# IP адрес или 127.0.0.1
nameserver 127.0.0.1
# или IP-адрес DNS-сервера — для остальных компьютеров сети
nameserver 10.0.0.1
```

Протестировать настройки можно с помощью программы nslookup:

```
# nslookup yandex.ru
Server: localhost.firma.ru
Address: 127.0.0.1
Non-authoritative answer:
Name: yandex.ru
Address: 213.180.216.200
```

Если вы получили подобный ответ, то это означает, что наш сервер работает нормально. Обратите внимание, что ответ пришел не от DNS-сервера провайдера, а от нашего локального сервера.

В Ubuntu есть небольшая проблема с перезаписью resolv.conf. Как только вы его перезапишете, он будет возвращен в исходное состояние при установке соединения или при перезагрузке. О моей борьбе с Ubuntu можно прочитать по адресу:

<http://www.dkws.org.ua/index.php?page=show&file=a/ubuntu/static-dns-ubuntu9>

Вообще, можно было бы запретить изменение файла с помощью команды `chattr`, зато я докопался до истины. В книге весь процесс для экономии места рассматривать не будем, но все желающие смогут ознакомиться с ним по указанному ранее адресу.

34.3. Полноценный DNS-сервер

Теперь можно перейти к настройке полноценного сервера DNS, если, конечно, он вам нужен. Но сначала нужно поговорить о том, что такое зона, поскольку полноценный DNS-сервер обслуживает одну или несколько зон. Ошибочно считать зоной обслуживаемый домен — это не так. Давайте разберемся, в чем разница. Домен — это группа компьютеров с одинаковой правой частью доменного имени. Пусть у нас есть домен **firma.ru**. Компания, которой принадлежит этот домен, довольно большая, поэтому для каждого подразделения пришлось организовать свой домен — **sales.firma.ru**, **dev.firma.ru**, **orders.firma.ru** и т. д. Для управления всем доменом **firma.ru** (и всеми поддоменами) мы можем использовать или единственный DNS-сервер, или же создать независимые серверы для каждого поддомена либо только для некоторых поддоменов. Например, основной сервер будет обслуживать только домены **firma.ru** и **sales.firma.ru**, а дополнительный сервер — домены **dev.firma.ru** и **orders.firma.ru**. Домены **firma.ru** и **sales.firma.ru** образуют одну зону, а домены **dev.firma.ru** и **orders.firma.ru** — другую. Другими словами, зона — это часть домена, управляемая определенным DNS-сервером. Зона, которая содержит домены низшего уровня, называется *подчиненной зоной* (subordinate zone).

Вот теперь можно приступить к настройке сервера. Первым делом нам нужно настроить удаленное управление сервером, а именно: настроить секцию `controls`, которую мы оставили пустой в предыдущем примере. Выполните команду:

```
# /usr/sbin/rndc-confgen > rndc.conf
```

Откройте `rndc.conf` в любом текстовом редакторе. Нам нужно выделить и скопировать две директивы — `controls` и `key`:

```
key "rndc-key" {
    algorithm hmac-md5;
    secret "ключ";
};
controls {
# разрешаем "удаленное" управление только с локального компьютера
    inet 127.0.0.1 port 953
    allow { 127.0.0.1; } keys { "rndc-key"; };
};
```

Скопированный блок текста нужно вставить в `named.conf` (в самое начало). Понятно, что из него нужно удалить пустую директиву `controls`, если она есть в файле.

При настройке кэширующего сервера DNS мы в его конфигурационном файле описали две зоны: корневую и локальную. Теперь нам нужно описать две зоны: прямого и обратного преобразования, которые и будут обслуживать наш домен. Добавьте в файл конфигурации `named.conf` строки:

```
zone "firma.ru" {
    type master;
    file "firma.ru";
```

```

    notify no;
};

zone "10.0.0.10.in-addr.arpa" {
    type master;
    file "10.0.0.1";
    notify yes;
}

```

Файл `firma.ru` (он должен находиться в каталоге, заданном директивой `directory`) используется для прямого преобразования, т. е. для преобразования доменных имен в IP-адреса. В листинге 34.3 представлен пример этого файла.

Листинг 34.3. Пример файла прямого преобразования

```

@      IN      SOA      server.firma.ru.  hostmaster.firma.ru. (
                                20040603  ; серийный номер (можно узнать в файлах с примерами)
                                3600        ; обновление каждый час
                                3600        ; повтор каждый час
                                3600000     ; время хранения информации 1000 часов
                                3600        ; TTL записи
)

      IN NS      server.firma.ru.
      IN A       10.0.0.1
      IN MX      100      server.firma.ru.
www    IN CNAME  server.firma.ru.
ftp    IN CNAME  server.firma.ru.
mail   IN CNAME  server.firma.ru.
c2     IN A      10.0.0.2
c3     IN A      10.0.0.2
localhost.  IN A  127.0.0.1

```

Разберемся, что означают записи этого файла. Первым делом обратите внимание на то, что в конце каждого доменного имени ставится точка — это для того, чтобы сервер не приписывал имя домена (**firma.ru**) к доменному имени. Если лень писать имя домена, тогда можно просто указывать имя компьютера (`server` вместо **server.firma.ru**), но тогда не нужно ставить точку в конце доменного имени.

Разберемся с записью `IN SOA`. Она описывает начало полномочий (Start Of Authority, SOA). Первое имя после SOA — это имя данного компьютера (на котором запущен DNS-сервер). В нашем случае это **server.firma.ru**. Затем следует e-mail администратора сервера, но поскольку символ `@` зарезервирован, то вместо него используется точка. Остальные элементы записи SOA прокомментированы в листинге.

Запись NS (IN NS) задает имя сервера доменных имен, а запись A — его IP-адрес. Запись MX используется для задания почтового сервера. Как мы видим, в роли почтового сервера используется все тот же наш server.firma.ru. 100 — это приоритет почтового сервера. Приоритет используется, если указано два (или более) почтовых сервера. Чем меньше число, тем выше приоритет:

```
IN MX      100      mail1
IN MX      150      mail2
```

Запись CNAME используется для определения канонических имен, т. е. псевдонимов. Как мы видим, к нашему серверу server.firma.com можно обратиться по следующим именам: www.firma.ru, ftp.firma.ru, mail.firma.ru.

Далее описаны два компьютера — c2.firma.ru (мы не ставили точку после c2, поэтому firma.ru сервер "допишет" автоматически) и c3.firma.ru, с IP-адресами 10.0.0.2 и 10.0.0.3 соответственно.

Последняя запись — это определение имени localhost, желательно не забыть о нем.

Теперь пора приступить к рассмотрению файла обратного соответствия, который представлен в листинге 34.4. Напомню, что этот файл используется для преобразования IP-адресов в доменные имена.

Листинг 34.4. Пример файла обратного преобразования

```
@      IN      SOA    server.firma.ru.    hostmaster.firma.ru. (
                                20040603    ; серийный номер (можно узнать в файлах с примерами)
                                3600         ; обновление каждый час
                                3600         ; повтор каждый час
                                3600000      ; время хранения информации 1000 часов
                                3600         ; TTL записи
)
@      IN      NS     server.firma.ru
1      IN      PTR    server.firmaru
2      IN      PTR    c2.firma.ru
3      IN      PTR    c3.firma.ru
```

В данном файле, если вы успели заметить, можно полностью не указывать IP-адрес, но нужно полностью указывать доменное имя (точки в конце доменного имени не нужны). Если же вам хочется указать IP-адрес полностью, тогда нужно указывать его в обратном порядке, например:

```
2.0.0.10    IN      PTR    c2.firma.ru
```

Вот, практически, и все. Можно в целях защиты сервера добавить в блок options (конфигурационный файл named.conf.options) директиву allow-query:

```
allow-query {
    10.0.0.0/24;
    localhost;
}
```

Блок `allow-query` разрешает запросы к серверу только узлам подсети 10.0.0.0 и от узла `localhost`. Узлы других подсетей не смогут использовать наш сервер. Когда вы настраиваете DNS-сервер, который будет работать в локальной сети (обслуживать только клиентов нашей локальной сети), то, по большому счету, блок `allow-query` вам не нужен. Но когда вы настраиваете DNS-сервер провайдера или же сервер, работающий в сети с реальными IP-адресами, то директива `allow-query` просто необходима, чтобы "чужие" узлы не смогли использовать наш сервер.

Полный файл конфигурации полноценного DNS-сервера для домена `firma.ru` представлен в листинге 34.5. Описание зон и опций я не выносил в файлы `named.conf.options` и `named.conf.local` для наглядности — чтобы вы в одном листинге увидели все настройки сервера.

Листинг 34.5. Полная версия файла конфигурации `named.conf`

```
key "rndc-key" {
    algorithm hmac-md5;
    secret "ключ";
};

controls {
    inet 127.0.0.1 port 953
    allow { 127.0.0.1; } keys { "rndc-key"; };
};

options {
    directory "/etc/bind";
    allow-query {
        10.0.0.0/24;
        localhost;
    }
};

zone "." in {
    type hint;
    file "db.root";
};

zone "0.0.127.in-addr.arpa" in {
    type master;
    file "db.127";
};

zone "localhost" {
    type master;
    file "db.local";
};
```

```
zone "255.in-addr.arpa" {
    type master;
    file "db.255";
};

zone "firma.ru" {
    type master;
    file "firma.ru";
    notify no;
};

zone "1.0.0.10.in-addr.arpa" {
    type master;
    file "10.0.0.1";
    notify yes;
}
```

После настройки сервер нужно перезапустить:

```
# service named restart
```

34.4. Вторичный DNS-сервер

В идеале для поддержки домена должно быть выделено два сервера — первичный и вторичный. Вторичный используется для подстраховки, если вдруг с первичным что-то случится (например, банальная перезагрузка администратором).

Вторичный сервер DNS описывается аналогично первичному, но несколько иначе описывается зона домена:

```
zone "firma.ru" {
    type slave;
    file "firma.ru";
    masters { 10.0.0.1; };
};
```

Как видим, устанавливается тип сервера — подчиненный (slave), а в блоке `masters` описываются первичные серверы (у нас он один).

В файл конфигурации первичного сервера нужно добавить директиву `allow-transfer`, в которой нужно указать DNS-серверы, которым разрешен трансфер зоны, т. е. все вторичные серверы:

```
options {
    ...
    allow-transfer { 10.0.0.2; };
}
```

34.5. Обновление базы данных корневых серверов

Чтобы база данных корневых серверов всегда была актуальной, ее нужно регулярно обновлять. Получить ее можно по адресу **ftp://ftp.internic.net/domain/named.root**, а обновить — с помощью трех команд:

```
wget ftp://ftp.internic.net/domain/named.root
sudo cp named.root /etc/bind/db.root
sudo /etc/init.d/bind9 restart
```

В листинге 34.6 содержится самая актуальная на момент написания этих строк версия файла **named.root**.

Листинг 34.6. Файл **named.root (db.root)**

```
;      This file holds the information on root name servers needed to
;      initialize cache of Internet domain name servers
;      (e.g. reference this file in the "cache . <file>"
;      configuration file of BIND domain name servers).
;
;      This file is made available by InterNIC
;      under anonymous FTP as
;          file                /domain/named.cache
;          on server           FTP.INTERNIC.NET
;      -OR-                   RS.INTERNIC.NET
;
;      last update:    Jun 17, 2010
;      related version of root zone:    2010061700
;
; formerly NS.INTERNIC.NET
;
.                3600000   IN   NS      A.ROOT-SERVERS.NET.
A.ROOT-SERVERS.NET. 3600000       A      198.41.0.4
A.ROOT-SERVERS.NET. 3600000       AAAA   2001:503:BA3E::2:30
;
; FORMERLY NS1.ISI.EDU
;
.                3600000       NS      B.ROOT-SERVERS.NET.
B.ROOT-SERVERS.NET. 3600000       A      192.228.79.201
;
; FORMERLY C.PSI.NET
;
```

```

.                3600000      NS      C.ROOT-SERVERS.NET.
C.ROOT-SERVERS.NET. 3600000      A      192.33.4.12
;
; FORMERLY TERP.UMD.EDU
;
.                3600000      NS      D.ROOT-SERVERS.NET.
D.ROOT-SERVERS.NET. 3600000      A      128.8.10.90
;
; FORMERLY NS.NASA.GOV
;
.                3600000      NS      E.ROOT-SERVERS.NET.
E.ROOT-SERVERS.NET. 3600000      A      192.203.230.10
;
; FORMERLY NS.ISC.ORG
;
.                3600000      NS      F.ROOT-SERVERS.NET.
F.ROOT-SERVERS.NET. 3600000      A      192.5.5.241
F.ROOT-SERVERS.NET. 3600000      AAAA   2001:500:2F::F
;
; FORMERLY NS.NIC.DDN.MIL
;
.                3600000      NS      G.ROOT-SERVERS.NET.
G.ROOT-SERVERS.NET. 3600000      A      192.112.36.4
;
; FORMERLY AOS.ARL.ARMY.MIL
;
.                3600000      NS      H.ROOT-SERVERS.NET.
H.ROOT-SERVERS.NET. 3600000      A      128.63.2.53
H.ROOT-SERVERS.NET. 3600000      AAAA   2001:500:1::803F:235
;
; FORMERLY NIC.NORDU.NET
;
.                3600000      NS      I.ROOT-SERVERS.NET.
I.ROOT-SERVERS.NET. 3600000      A      192.36.148.17
I.ROOT-SERVERS.NET. 3600000      AAAA   2001:7FE::53
;
; OPERATED BY VERISIGN, INC.
;
.                3600000      NS      J.ROOT-SERVERS.NET.
J.ROOT-SERVERS.NET. 3600000      A      192.58.128.30
J.ROOT-SERVERS.NET. 3600000      AAAA   2001:503:C27::2:30
;

```

```
; OPERATED BY RIPE NCC
;
.                3600000      NS      K.ROOT-SERVERS.NET.
K.ROOT-SERVERS.NET. 3600000      A      193.0.14.129
K.ROOT-SERVERS.NET. 3600000      AAAA   2001:7FD::1
;
; OPERATED BY ICANN
;
.                3600000      NS      L.ROOT-SERVERS.NET.
L.ROOT-SERVERS.NET. 3600000      A      199.7.83.42
L.ROOT-SERVERS.NET. 3600000      AAAA   2001:500:3::42
;
; OPERATED BY WIDE
;
.                3600000      NS      M.ROOT-SERVERS.NET.
M.ROOT-SERVERS.NET. 3600000      A      202.12.27.33
M.ROOT-SERVERS.NET. 3600000      AAAA   2001:DC3::35
; End of File
```


ГЛАВА 35



DNCP-сервер

35.1. Протокол динамической конфигурации узла

DNCP (Dynamic Host Configuration Protocol) используется для автоматической настройки узлов сети. С помощью DNCP компьютер, подключенный к сети, в которой есть DNCP-сервер, может получить IP-адрес, маску сети, IP-адрес шлюза, адреса серверов DNS и другие сетевые параметры.

Особенно удобно использовать DNCP в средних и больших сетях. Вы только представьте, что у вас есть, скажем, 20 компьютеров. Если каждому компьютеру назначить IP-адрес статически, то вам нужно подойти к каждому компьютеру и указать его IP-адрес. Заодно вам нужно ввести IP-адрес сети, IP-адрес шлюза и адреса серверов DNS. Понятно, что эту процедуру нужно выполнить разово — при настройке сети. Но если через некоторое время конфигурация сети изменится (например, вы меняете провайдера), и нужно будет изменить IP-адреса DNS-серверов, то вам придется все повторить заново: подойти к каждому компьютеру и прописать DNS-серверы.

Если же потратить полчаса на настройку DNCP-сервера, можно будет централизованно управлять конфигурацией сети. Вам стоит изменить IP-адрес DNS-сервера в конфигурационном файле DNCP-сервера — на остальных компьютерах сети новые IP-адреса DNS-серверов "пропишутся" автоматически. Удобно? Я тоже так думаю.

Для установки DNCP-сервера вам достаточно установить пакет `dhcpr`. DNCP-клиенты входят в состав Linux и Windows, поэтому их устанавливать отдельно не нужно.

35.2. Конфигурационный файл DNCP-сервера

Конфигурационный файл DNCP-сервера называется `/etc/dhcpd.conf`. Пример этого файла вы можете найти в `/usr/share/doc/dhcp-<версия>/dhcpd.conf.sample`.

Относительно файла конфигурации нужно сделать два замечания:

- ❖ директивы не чувствительны к регистру символов, т. е. вы можете написать как `option`, так и `OPTION`, но принято писать строчными буквами;
- ❖ комментарии начинаются с символа решетки (#).

В начало файла конфигурации нужно поместить одну из директив:

```
ddns-update-style ad-hoc;
```

или

```
ddns-update-style interim;
```

Сейчас разберемся, что это такое. Существуют две схемы обновления DNS: непосредственное обновление (`ad-hoc`) и предварительное взаимодействие DHCP-DNS (`interim`). Вторая схема пока не утверждена комитетом по техническому развитию Интернета, но уже успешно применяется, и разработчики DHCP рекомендуют использовать именно ее. Тут выбирать вам: или использовать старую схему взаимодействия (первая директива), или остановиться на более перспективной (вторая директива).

По сути, весь конфигурационный файл DHCP-сервера будет состоять из директивы `ddns-update-style` и блочной директивы `section`, описывающей вашу сеть.

Рассмотрим пример объявления сети 192.168.1.0 (листинг 35.1).

Листинг 35.1. Описание сети 192.168.1.0

```
subnet 192.168.1.0 netmask 255.255.255.0 {  
# шлюз по умолчанию  
    option routers                192.168.1.1;  
# маска сети — этот параметр будет передан всем компьютерам сети  
    option subnet-mask            255.255.255.0;  
# наш домен  
    option domain-name            "example.ru";  
# IP-адрес сервера DNS  
    option domain-name-servers    192.168.1.1;  
# диапазон IP-адресов: компьютерам нашей сети будут присваиваться  
# IP-адреса из этого диапазона  
range 192.168.1.10 192.168.1.100;  
}
```

Если у нас есть большая сеть, и есть несколько подсетей, то все подсети (директива `subnet`) должны быть описаны в одной директиве `shared-network`. При этом все общие для подсетей параметры: описание маршрутизаторов, DNS-серверов, доменное имя — выносятся за пределы директив `subnet` (листинг 35.2).

Листинг 35.2. Большая сеть и ее подсети

```
shared-network имя_нашей_сети {  
# описываем глобальные для всех подсетей параметры
```

```
# домен
    option domain-name                "example.ru";
# серверы DNS
    option domain-name-servers        ns1.isp.com, ns2.isp.com;
# шлюз по умолчанию
    option routers                     192.168.0.1;

# описываем подсети 192.168.1.0 и 192.168.2.0
    subnet 192.168.1.0 netmask 255.255.252.0 {
        range 192.168.1.10 192.168.1.254;
    }
    subnet 192.168.2.0 netmask 255.255.252.0 {
        range 192.168.2.10 192.168.2.254;
    }
}
# конец директивы shared-network
```

35.3. База данных аренды

ДHCP-сервер назначает IP-адрес компьютеру не навсегда, а только на некоторое время, называемое временем аренды. По истечении данного времени компьютеру будет назначен другой IP-адрес.

Время аренды регулируется директивами `default-leased-time` и `max-leased-time`, но обычно не нужно изменять значения этих директив, потому что значения по умолчанию вполне приемлемы.

База данных аренды, т. е. сведения, кому и какой IP-адрес был назначен, находится в файле `/var/lib/dhcp/dhcpd.leases`. В этом файле указана следующая информация: уникальный MAC-адрес сетевого адаптера компьютера (аппаратный адрес), назначенный IP-адрес, дата и время окончания аренды и др.

Базу данных аренды нельзя редактировать вручную, ее можно только просматривать.

35.4. Полный листинг конфигурационного файла

Окончательный вариант конфигурационного файла для подсети 192.168.1.0 представлен в листинге 35.3.

Листинг 35.3. Окончательный вариант конфигурационного файла ДHCP-сервера

```
# схема взаимодействия с DNS
ddns-update-style ad-hoc;
```

```

subnet 192.168.1.0 netmask 255.255.255.0 {

# шлюз по умолчанию
    option routers                192.168.1.1;
# маска сети — этот параметр будет передан всем компьютерам сети
    option subnet-mask            255.255.255.0;
# наш домен
    option domain-name            "example.ru";
# IP-адрес сервера DNS
    option domain-name-servers    192.168.1.1;
# диапазон IP-адресов: компьютерам нашей сети будут присваиваться
# IP-адреса из этого диапазона
    range 192.168.1.10 192.168.1.100;
}

```

35.5. Управление сервером DHCP

Для запуска, перезапуска и останова сервера можно использовать команду `service`:

```

service dhcpd start
service dhcpd restart
service dhcpd stop

```

35.6. Настройка клиентов

Все клиенты вашей сети (разумеется, кроме серверов сети, у которых должны быть постоянные IP) должны быть настроены на автоматическое получение IP-адреса (рис. 35.1) и IP-адресов DNS-серверов.

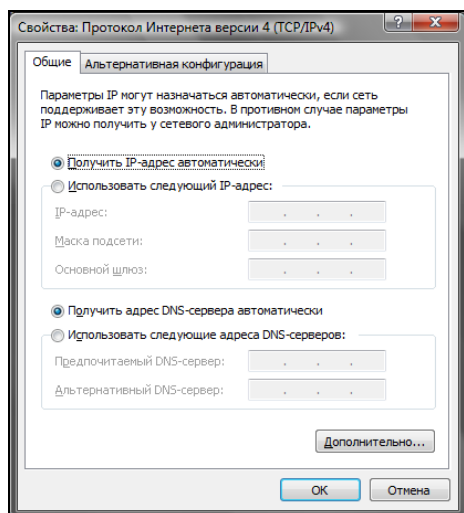


Рис. 35.1. Настройка Windows-клиента

ГЛАВА 36



Прокси-сервер SQUID

36.1. Зачем нужен прокси-сервер в локальной сети?

С помощью прокси-сервера Squid можно очень эффективно управлять ресурсами своей сети, например кэшировать трафик (http), "обрезать" баннеры, указать, какие файлы можно скачивать пользователям, а какие — нет, также можно указать максимальный объем передаваемого объекта и даже ограничить пропускную способность пользователей определенного класса.

Основная функция прокси-сервера — это кэширование трафика. Если в сети используется прокси-сервер, можно сократить кэш браузеров клиентов практически до нуля — он уже не будет нужен, поскольку кэширование будет выполнять прокси-сервер. Тем более, что он выполняет кэширование всех клиентов сети, и уже запрошенные ранее страницы доступны другим пользователям. Это означает, что если кто-то зашел на сайт **firma.ru**, то у всех остальных пользователей сети этот сайт будет открываться практически мгновенно, потому что его уже кэшировали.

Даже если у вас всего один компьютер, все равно есть смысл использовать Squid, хотя бы для того, чтобы "обрезать" баннеры — так можно сэкономить на трафике, да и страницы начнут открываться быстрее, не нужно уже грузить многочисленные баннеры.

Squid несложен в настройке, во всяком случае, не сложнее Samba и подобных сетевых сервисов. Нужно установить пакет squid. После установки пакета у вас в системе появится новый сервис — squid. Основной конфигурационный файл — `/etc/squid/squid.conf`.

36.2. Базовая настройка Squid

Сейчас приступим к редактированию основного конфигурационного файла `/etc/squid/squid.conf` (листинг 36.1).

Листинг 36.1. Файл /etc/squid/squid.conf

```
# Порт для прослушивания запросов клиентов.
# Задается в формате http_port <порт> или http_port <узел>:<порт>.
# Последний случай подходит, если SQUID запущен на машине с несколькими
# сетевыми интерфейсами
http_port 192.168.0.1:3128

# Адрес прокси провайдера, нужно согласовать с провайдером
# cach_peer proxy.your_isp.com

# Объем оперативной памяти в байтах, который будет использоваться
# прокси-сервером (85 Мбайт). Не устанавливайте более трети физического объема
# ОЗУ, если данная машина должна использоваться еще для чего-либо.
# Можно задать в мегабайтах, но тогда между числом и МВ обязательно
# должен быть пробел: cache_mem 85 MB
cache_mem 87040

# Где будет размещен кэш.
# Первое число – это размер кэша в мегабайтах. Не устанавливайте кэш на весь
# раздел. Если нужно, чтобы он занимал весь раздел, отнимите от размера
# раздела 20% и укажите это значение. Например, если раздел 1024 Мбайт,
# то для кэша – только 820 Мбайт. Второе – количество каталогов первого
# уровня. Третье – количество каталогов второго уровня.
cache_dir /usr/local/squid 1024 16 256

# Максимальный размер кэшируемого объекта.
# Если размер объекта превышает указанный здесь, то объект не будет
# сохранен на диске.
# maximum_object_size 4096 KB

# Хосты, с которых разрешен доступ к прокси
acl allowed_hosts src 192.168.1.0/255.255.255.0
acl localhost src 127.0.0.1/255.255.255.255

# разрешенные порты:
acl allow_ports port 80      # http
acl allow_ports port 21      # ftp
# SSL-порты
acl SSL_ports port 443 563

# Запрещаем все порты, кроме указанных в allow_ports
http_access deny !allow_ports
```

```
# Запрещаем метод CONNECT для всех портов, кроме указанных в acl SSL_ports
http_access deny CONNECT !SSL_ports

# Запретим доступ всем, кроме тех, кому можно
http_access allow localhost
http_access allow allowed_hosts
http_access allow SSL_ports
http_access deny all

# Пропишем пользователей, которым разрешено пользоваться squid (ppt, admin):
ident_lookup on
acl allowed_users ppt admin
http_access allow allowed_users
http_access deny all
```

Базовый конфигурационный файл с успехом выполняет только функцию кэширования, а в следующем разделе мы поговорим о более тонкой настройке Squid.

36.3. Практические примеры

36.3.1. Управление доступом

Управление доступом осуществляется с помощью ACL (Access Control List) — списков управления доступом.

Разберемся, как работать с ACL. Создадим список AllowedPorts:

```
acl AllowedPorts port 80 8080 3128
```

Имя списка — AllowedPorts, тип списка — port. Далее мы можем использовать этот список в http_access для разрешения/запрещения указанных портов:

```
http_access allow AllowedPorts # разрешение портов
http_access deny AllowedPorts # запрещение портов
```

Кроме типа port часто используются следующие типы списков:

- ◆ proto — протокол (HTTP или FTP);
- ◆ method — метод передачи данных (GET или POST);
- ◆ src — IP-адреса (или диапазоны адресов) клиентов;
- ◆ dst — IP-адреса/URL сайтов, к которым обращаются клиенты.

Вы также можете создать список узлов, которым разрешен доступ к прокси:

```
acl allowed_hosts src "/etc/squid/allowed-hosts.txt"
```

Сам файл /etc/squid/allowed-hosts.txt будет выглядеть так:

```
# den
192.168.0.2/255.255.255.255
# admin
192.168.0.3/255.255.255.255
```

Отдельный файл использовать удобнее, чтобы не "засорять" основной конфигурационный файл. Обратите внимание: права доступа к `allowed-hosts.txt` должны быть такие же, как к `squid.conf`.

36.3.2. Создание черного списка URL

Теперь попробуем создать черный список URL:

```
acl blacklist url_regex adult
http_access deny blacklist
http_access allow all
```

Данный черный список не пропускает URL, содержащие слово "adult". По аналогии можно было бы создать отдельный файл и записать в него все "плохие" URL (но это довольно накладно, проще использовать регулярные выражения).

36.3.3. Отказ от баннеров

С помощью ACL можно отказаться и от баннеров — принцип тот же. Для этого добавьте в файл конфигурации следующие ACL:

```
acl banners urlpath_regex "/etc/squid/banners.txt"
http_access deny banners
```

В файл `banners.txt` нужно внести URL баннерных сетей, например

```
^http://www.clickhere.ru
^http://banner.kiev.ua
...
```

Создание этого файла пусть будет вашим домашним заданием — все равно все баннерные сети в книге не приведешь.

36.4. Управление прокси-сервером

Для запуска, перезапуска и остановки прокси-сервера нужно использовать следующие команды:

```
# service squid start
# service squid restart
# service squid stop
```

36.5. Настройка клиентов

Все браузеры на компьютерах вашей сети нужно настроить на использование порта 3128 (именно этот порт мы установили в конфигурационном файле). На рис. 36.1 изображена настройка браузера Opera.

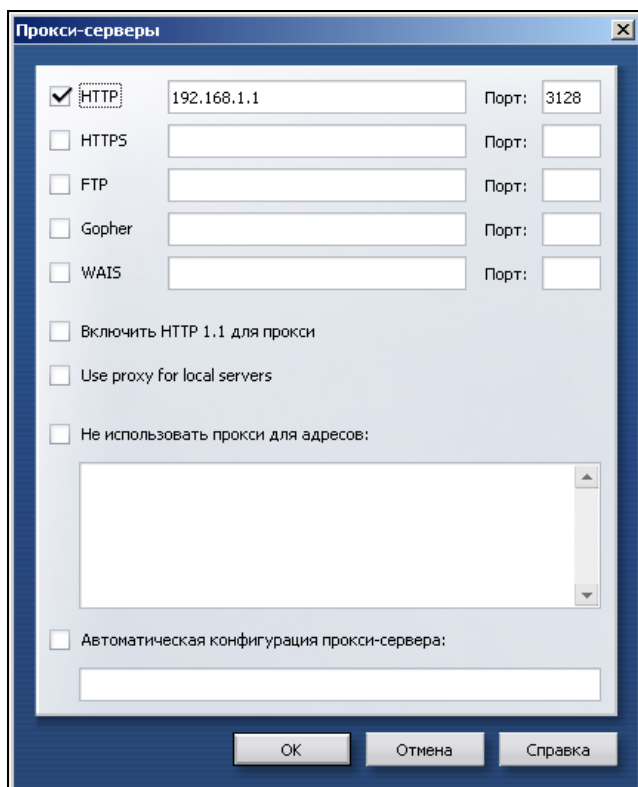


Рис. 36.1. Настройка клиента

36.6. Прозрачный прокси-сервер

С прокси-сервером часто связаны две проблемы. Первая заключается в том, что нужно настраивать всех клиентов для работы через прокси-сервер. Если сеть большая, скажем, 100 компьютеров, можете себе представить, сколько это займет времени — ведь нужно подойти к каждому компьютеру. Даже если на настройку одного компьютера потребуется 5 минут, то всего нужно 500 минут — целый рабочий день. Но настройкой браузера дело может и не обойтись. Ведь у пользователей могут быть и другие интернет-программы, работающие с WWW/FTP, которые также нужно будет настроить.

Проблема настройки не самая страшная. Понятно, что если в сети организации 100 или более компьютеров, то администратор будет не один. А вдвоем-втроем можно настроить все 100 компьютеров за 2—3 часа.

Вторая проблема более серьезная. Представим, что в сети у нас есть продвинутые пользователи (а они-таки есть), которые знают, для чего используется прокси-сервер. Они могут просто изменить настройки и вместо работы через прокси ис-

пользовать прямое соединение с Интернетом, т. е. будут работать в обход Squid. Вы так старались, создавая список черных URL (преимущественно это сайты для взрослых и всевозможные чаты/форумы), а они с помощью пары щелчков мыши сведут все ваши старания к нулю.

Обе проблемы можно решить, если настроить *прозрачный прокси-сервер*: пользователи даже не будут подозревать, что он есть. Во-первых, это решит проблемы с настройкой — вам не нужно настраивать браузеры пользователей, потому что все HTTP-запросы будут автоматически поступать на прокси-сервер. Во-вторых, прозрачный прокси обеспечит принудительное кэширование информации, соответственно, принудительный контроль за страницами, которые посещают пользователи.

Для настройки прозрачного прокси вам нужно изменить как конфигурационный файл самого прокси-сервера, так и правила брандмауэра iptables (его мы рассмотрим в следующей главе). Вот правила iptables:

```
iptables -t nat --new-chain TransProxy
# только порт 80 (HTTP) и 443 (SSL, https) — остальные обрабатывать не будем
iptables -t nat -A PREROUTING -p tcp --dport 80 -j TransProxy
iptables -t nat -A PREROUTING -p tcp --dport 443 -j TransProxy
iptables -t nat -A TransProxy -d 127.0.0.1/8 -j ACCEPT
# укажите IP-адрес своей сети
iptables -t nat -A TransProxy -d 192.168.1.0/24 -j ACCEPT
# все запросы перенаправляются на прокси-сервер 192.168.1.1, порт 3128
iptables -t nat -A TransProxy -p TCP -j DNAT --to 192.168.1.1:3128
```

Теперь займемся настройкой Squid. В конфигурационный файл squid.conf добавьте следующие директивы:

```
# серверу назначается реальный IP-адрес, его и нужно указать
tcp_outgoing_address ваш_реальный_IP
httpd_accel_host virtual
httpd_accel_with_proxy on
httpd_accel_uses_host_header on
```

Более подробно о настройке iptables мы поговорим в следующей главе. Забегая вперед, скажу, что iptables обычно устанавливается на шлюзе — компьютере, который предоставляет доступ к Интернету других компьютеров сети. На этом же компьютере должен быть установлен и Squid.

36.7. Расширение squidGuard

Чуть ранее были приведены примеры создания черных списков, ограничивающие доступ к сайтам с запрещенным контентом. Но пока вы сформируете базу черных списков, пройдет время. Для автоматизации этого процесса вы можете применить squidGuard, использующий уже готовые черные списки, сформированные большим сообществом пользователей и тщательно проверенные разработчиками

squidGuard. Расширение squidGuard не только сэкономит трафик, но и эффективно защитит вашу сеть от запрещенного контента.

Черный список squidGuard обновляется постоянно. Скачать его можно на сайте <http://www.squidguard.org/>. Там же вы найдете альтернативные черные списки. В базу squidGuard внесены самые известные сайты с запрещенным контентом, а именно: насилие, порнография, наркотики, азартные игры и т. д.

Для установки squidGuard достаточно установить одноименный пакет. После этого черный список узлов будет помещен в каталог `/usr/share/squidGuard-1-3-0/db` (версия squidGuard у вас может быть иная). В некоторых дистрибутивах или если вы устанавливаете squidGuard с исходных кодов, база будет помещена в каталог `/usr/local/squidGuard/db`.

Чтобы база данных была самой актуальной, скачайте последнюю версию базы по адресу: <http://www.squidguard.org/blacklists.html>. Вы скачаете файл `blacklist.tar.gz`. Теперь его нужно распаковать в каталог `/usr/share/squidGuard-1-3-0/db` или в `/usr/local/squidGuard/db`:

```
cp blacklist.tar.gz /usr/local/squidGuard/db
gzip -d blacklist.tar.gz
tar xfv blacklist.tar
```

После этого нужно немного отредактировать файл конфигурации squidGuard. Скопируйте файл `/etc/squid/squidGuard.conf.sample` в файл `/etc/squid/squidGuard.conf` и откройте его в текстовом редакторе. Весь файл редактировать не нужно, полный листинг этого файла тоже приводить не стану — он слишком длинный.

Первым делом нужно указать путь к базе и к журналам:

```
dbhome /usr/local/squidGuard/db
logdir /var/log/squidGuard
```

Теперь опишем разрешенное время работы:

```
# s = Вс, m = Пн, t =Вт, w = Ср, h = Чт, f = Пт, a = Сб
```

```
time workhours {
    weekly m 08:00-12:00 13:00-19:00
    weekly t 08:00-11:00 12:00-19:00
    weekly w 08:00-12:00 12:00-18:00
    weekly h 08:00-13:00 13:00-18:00
    weekly f 08:00-12:00 13:30-18:00
    weekly a 11:20-14:00
    weekly s 11:32-14:00
}
```

Опишем две зоны. К первой будут относиться наши пользователи, а ко второй — администраторы сети. Пользователи, не относящиеся к первым двум группам, вообще не будут иметь доступа к Интернету.

```
src users {
    ip 192.168.1.5-192.168.1.200
}
```

```
src admins {  
    ip 192.168.1.1-192.168.1.4  
}
```

Описываем списки доступа, определяющие, кому и к каким узлам разрешен доступ. Администраторам разрешаем доступ ко всем узлам, кроме рекламных баннеров, а вот пользователям запрещаем доступ по максимуму.

```
acl {  
    admins {  
  
        pass !advertising all  
# запрещенные запросы перенаправляем на следующий адрес  
        redirect http://server.ru/error.html  
    }  
  
    users {  
        pass !adult !audio-video !forums !hacking !redirector !warez  
            !ads !aggressive !drugs !gambling !publicite !violence  
            !banneddestination !advertising all  
  
        redirect http://server.ru/error.html  
    }  
  
# остальным пользователям доступ к Интернету запрещен (pass none)  
    default {  
        pass none  
        redirect http://server.ru/error.html  
    }  
}
```

Почти все. Осталось только "прописать" расширение squidGuard в конфигурационном файле Squid. Откройте файл /etc/squid/squid.conf и добавьте в него следующие строки:

```
redirector_bypass on  
redirect_program /usr/local/squidGuard/bin/squidGuard  
redirect_children 1
```

Сохраните файл и перезапустите Squid:

```
# service squid restart
```

Теперь выполните следующую команду:

```
tac /var/log/squidGuard/squidGuard.log | less
```

Вы должны увидеть сообщение о том, что squidGuard запущен (started) и готов к обработке запросов (ready for requests). Если вы увидели заветные строки, значит, вы все сделали правильно.

ГЛАВА 37



Маршрутизация и настройка брандмауэра

37.1. Краткое введение в маршрутизацию

Для начала нужно сказать, что такое маршрутизация. *Маршрутизация* — это процесс перенаправления пакета по сетям, находящимся между отправителем и получателем. Представьте, что вам нужно поехать в гости к другу в город, в котором вы никогда не были. Понимаю, что на дворе XXI век, и GPS-навигатор — больше не принадлежность Джеймса Бонда, но все же о навигаторах на минуту забудем. Итак, сначала вам нужно выяснить, как проехать в город, в котором живет ваш друг. Если вы живете в относительно большом городе, то первым делом нужно узнать, как выехать из своего города — можно выехать в любом направлении, но потом придется проехать лишнее расстояние, чего бы не хотелось. Поэтому спрашиваем у таксиста, куда вам ехать. После того как вы выбрались из своего города и знаете примерное направление, в котором вам нужно ехать, вы себе спокойно едете, пока не начнете сомневаться в правильности маршрута. Тогда вы остановитесь на придорожной АЗС или посту ДПС и узнаете, куда вам ехать дальше. Возможно, придется проехать еще через несколько городов, в каждом городе вам нужно будет спросить, куда ехать. Можно и не спрашивать — если есть знаки. Одним словом, либо человек, либо дорожный знак укажут вам дорогу. Когда вы приедете в город друга, вам нужно будет узнать, где находится улица, на которой он живет. А когда окажетесь на нужной вам улице, наверняка попросите прохожих подсказать, где находится дом с нужным номером.

Маршрутизация пакетов выполняется примерно так же. В приведенном примере с путешественником "пакетом" были именно вы, а роль маршрутизаторов играли люди, которые подсказывали, куда вам ехать.

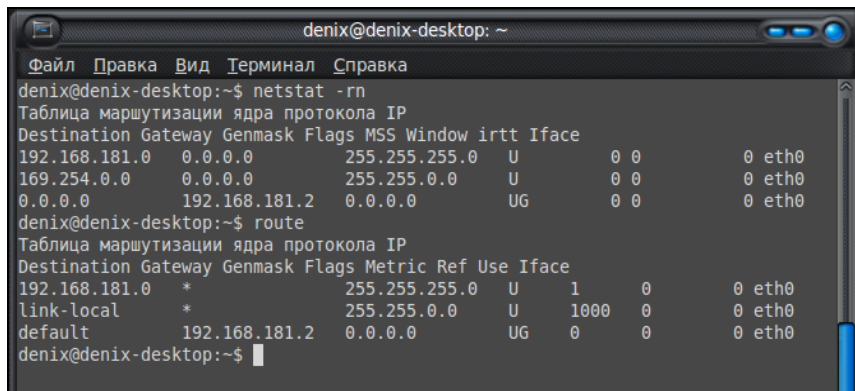
В TCP/IP-сетях информация о маршрутах имеет вид правил, например, чтобы добраться к сети А, нужно отправить пакеты через компьютер Д. Ничего удивительного и необычного: примерно так же выглядит и информация о маршрутах на дороге — чтобы доехать до города А нужно проехать через город Д. Кроме набора

маршрутов есть также и стандартный маршрут — по нему отправляют пакеты, предназначенные для отправки в сеть, маршрут к которой явно не указан. Компьютер, на который отправляются эти пакеты, называется *шлюзом по умолчанию* (default gateway). Получив пакет, шлюз решает, что с ним сделать: или отправить дальше, если ему известен маршрут в сеть получателя пакета, или же уничтожить пакет, как будто бы его никогда и не было. В общем, что сделать с пакетом — это личное дело шлюза по умолчанию, все зависит от его набора правил маршрутизации. Наше дело маленькое — отправить пакет на шлюз по умолчанию.

Данные о маршрутах хранятся в таблице маршрутизации ядра Linux. Каждая запись этой таблицы содержит несколько параметров: адрес сети назначения, сетевую маску и т. д. Если пакет не удалось отправить ни по одному маршруту (в том числе и по стандартному), отправителю пакета передается ICMP-сообщение "сеть недоступна" (network unreachable). Далее мы подробно рассмотрим работу с таблицей маршрутизации ядра.

37.2. Таблица маршрутизации ядра. Установка маршрута по умолчанию

Для просмотра таблицы маршрутизации используются команды `netstat -r` и `netstat -rn`. Можно также по старинке воспользоваться командой `route` без параметров. Разница между командами `netstat -r` и `netstat -rn` заключается в том, что параметр `-rn` запрещает поиск доменных имен в DNS, поэтому все адреса будут представлены в числовом виде (подобно команде `route` без параметров). А вот разница между выводом `netstat` и `route` заключается в представлении маршрута по умолчанию (`netstat` выводит адрес 0.0.0.0, а `route` — метку default) и в названии полей самой таблицы маршрутизации. На рис. 37.1 изображен вывод команд `netstat -rn` и `route`.



```

denix@denix-desktop: ~
Файл Правка Вид Терминал Справка
denix@denix-desktop:~$ netstat -rn
Таблица маршрутизации ядра протокола IP
Destination Gateway Genmask Flags MSS Window irtt Iface
192.168.181.0 0.0.0.0 255.255.255.0 U 0 0 0 eth0
169.254.0.0 0.0.0.0 255.255.0.0 U 0 0 0 eth0
0.0.0.0 192.168.181.2 0.0.0.0 UG 0 0 0 eth0
denix@denix-desktop:~$ route
Таблица маршрутизации ядра протокола IP
Destination Gateway Genmask Flags Metric Ref Use Iface
192.168.181.0 * 255.255.255.0 U 1 0 0 eth0
link-local * 255.255.0.0 U 1000 0 0 eth0
default 192.168.181.2 0.0.0.0 UG 0 0 0 eth0
denix@denix-desktop:~$

```

Рис. 37.1. Команды `netstat -rn` и `route`

Какую команду использовать — решать вам. Раньше я использовал `route` и для просмотра, и для редактирования таблицы маршрутизации. Теперь для просмотра таблицы я использую команду `netstat -rn`, а для ее изменения — команду `route`.

В нашем случае есть две сети 192.168.181.0 и 169.254.0.0 — обе на интерфейсе `eth0`. Такая ситуация сложилась из-за особенностей NAT/DHCP виртуальной машины VMWare, в которой была запущена Linux. В реальных условиях обычно будет по одной подсети на одном интерфейсе. С другой стороны, данный рисунок (рис. 37.1) демонстрирует поддержку VLAN, когда один интерфейс может использоваться двумя подсетями. Шлюз по умолчанию — компьютер с адресом 192.168.181.2, о чем свидетельствует таблица маршрутизации.

Поля таблицы маршрутизации объясняются в табл. 37.1.

Таблица 37.1. Поля таблицы маршрутизации

Поле	Описание
Destination	Адрес сети назначения
Gateway	Шлюз по умолчанию
Genmask	Маска сети назначения
Flags	Содержит флаги маршрута: <ul style="list-style-type: none"> ▪ U — маршрут активен; ▪ H — маршрут относится не к сети, а к хосту; ▪ G — данная машина является шлюзом, поэтому при обращении к ней нужно заменить MAC-адрес машины получателя на MAC-адрес шлюза (если MAC-адрес получателя почему-то известен); ▪ D — динамический маршрут, установлен демоном маршрутизации; ▪ M — маршрут, модифицированный демоном маршрутизации; ▪ C — запись кэширована; ▪ ! — запрещенный маршрут
Metric	Метрика маршрута, т. е. расстояние к цели в хопх (переходах). Один хоп (переход) означает один маршрутизатор
Ref	Количество ссылок на маршрут. Не учитывается ядром Linux, но в других операционных системах, например в FreeBSD, вы можете столкнуться с этим полем
Use	Содержит количество пакетов, прошедших по этому маршруту
Iface	Используемый интерфейс
MSS	Максимальный размер сегмента (Maximum Segment Size) для TCP-соединений по этому маршруту
Window	Размер окна по умолчанию для TCP-соединений по этому маршруту

Таблица 37.1 (окончание)

Поле	Описание
irtt	Протокол TCP гарантирует надежную доставку данных между компьютерами. Для такой гарантии используется повторная отправка пакетов, если они были потеряны. При этом ведется счетчик времени: сколько нужно ждать, пока пакет дойдет до назначения и придет подтверждение о получении пакета. Если время вышло, а подтверждение-таки не было получено, то пакет отправляется еще раз. Это время и называется round-trip time (время "путешествия туда-обратно"). Параметр irtt — это начальное время rtt. В большинстве случаев подходит значение по умолчанию, но для некоторых медленных сетей, например для сетей пакетного радио, значение по умолчанию слишком короткое, что вызывает ненужные повторы. Параметр irtt можно увеличить командой <code>route</code> . По умолчанию его значение — 0

Добавить маршрут в таблицу маршрутизации можно статически (с помощью команды `route`), динамически или комбинированно (например, статические маршруты добавляются при запуске системы, а динамические — по мере работы системы). Статические маршруты добавляются, как правило, командой `route`, запущенной из сценария инициализации системы. Например, следующая команда задает шлюз по умолчанию для интерфейса `eth0`:

```
# route add default gw 192.168.181.2 eth0
```

Но после перезагрузки системы добавленная нами запись исчезнет из таблицы маршрутизации. Можно добавить данную команду в сценарии инициализации системы, но это будет некорректно. Есть более корректный способ установки шлюза по умолчанию. В Fedora, Red Hat и других совместимых с ними дистрибутивах (CentOS, ASPLinux) нужно отредактировать файл `/etc/sysconfig/network`. Переменная `GATEWAY` содержит IP-адрес шлюза по умолчанию. Пример этого файла приведен в листинге 37.1.

Листинг 37.1. Файл `/etc/sysconfig/network`: основные сетевые параметры в Fedora

```
NETWORKING=yes
FORWARD_IPV4=yes
HOSTNAME=den.dkws.org.ua
GATEWAY=0.0.0.0
```

ПРИМЕЧАНИЕ

С некоторыми конфигурационными файлами, рассматриваемыми в этой главе, вы уже знакомы, но в этой главе они рассматриваются в разрезе маршрутизации.

Параметр `NETWORKING` определяет, будет ли включена поддержка сети (`yes` — поддержка сети включена, `no` — выключена). Параметр `FORWARD_IPV4` определяет, будет ли включено перенаправление пакетов. На компьютере, являющемся шлю-

зом, данный параметр должен быть включен (значение `yes`), на остальных компьютерах сети — выключен (значение `no`).

Параметр `HOSTNAME` задает имя узла, `GATEWAY` — шлюз по умолчанию. Если компьютер является шлюзом, то обычно для этого параметра устанавливается IP-адрес 0.0.0.0.

В SUSE для задания шлюза по умолчанию нужно отредактировать файл `/etc/route.conf` или `/etc/sysconfig/network/routes` (современные версии `openSUSE`). В него нужно добавить строку вида:

```
default      адрес      [маска]      [интерфейс]
```

Например:

```
default      192.168.181.2
```

Маску и интерфейс указывать необязательно. В этом же файле можно указать все остальные маршруты, т. е. по сути, этот файл хранит таблицу маршрутизации. Маршрут по умолчанию, как правило, указывается последним. Пример файла конфигурации `/etc/sysconfig/network/routes` (`/etc/route.conf`) приведен в листинге 37.2.

Листинг 37.2. Файл `/etc/route.conf`

```
#
# /etc/sysconfig/network/routes (/etc/route.conf)
#
# Данный файл содержит описание статических маршрутов
#
# Назначение  Шлюз           Маска           Устройство
#
192.168.0.0   0.0.0.0           255.255.255.128 eth0
default      192.168.0.1
```

Кроме файла `route.conf` в SUSE вы можете редактировать файл `/etc/rc.config`, содержащий информацию о сетевых интерфейсах. В этом файле содержится вся информация об имеющихся сетевых интерфейсах. Здесь важно отметить, что речь идет о старых версиях SUSE. А в *главе 27* мы рассматривали конфигурационные файлы современных версий `openSUSE`.

В Debian и Ubuntu вам нужно редактировать файл `/etc/network/interfaces`. Шлюз по умолчанию задается параметром `gateway`. В листинге 37.3 приведен пример файла `/etc/network/interfaces`. Напомню, что подробно синтаксис этого файла описан в моей статье по адресу: <http://dkws.org.ua/index.php?page=show&file=a/ubuntu/network-interfaces>. Но позволю себе несколько комментариев. Как видно из листинга 37.3, производится конфигурация интерфейса `eth0`, IP-адрес задается статически (`static`), присваивается IP-адрес 192.168.1.11, маска 255.255.255.0. Шлюз по умолчанию — это компьютер с IP-адресом 192.168.1.1.

Листинг 37.3. Файл /etc/network/interfaces

```
iface eth0 inet static
address 192.168.1.11
netmask 255.255.255.0
gateway 192.168.1.1
```

37.3. Изменение таблицы маршрутизации.

Команда *route*

Мы уже знакомы с командой *route*, но использовали ее для просмотра таблицы маршрутизации. Сейчас мы научимся ее применять для изменения таблицы маршрутов.

Маршрутизация осуществляется на сетевом уровне модели OSI. Когда маршрутизатор получает пакет, предназначенный для другого узла, его IP-адрес получателя сравнивается с записями в таблице маршрутизации. Если есть хотя бы частичное совпадение с каким-то маршрутом из таблицы, пакет отправляется по IP-адресу шлюза, связанного с данным маршрутом.

Если совпадений не найдено (т. е. вообще нет маршрута, по которому можно было бы отправить пакет), тогда пакет отправляется на шлюз по умолчанию, если таковой задан в таблице маршрутизации. Если шлюза по умолчанию нет, отправителю пакета посылается ICMP-сообщение "сеть недоступна" (network unreachable).

Команда *route* за один вызов может добавить или удалить только один маршрут. Другими словами, вы не можете сразу добавить или удалить несколько маршрутов. Формат вызова *route* следующий:

```
# route [операция] [тип] адресат gw шлюз [метрика] [dev интерфейс]
```

ПРИМЕЧАНИЕ

Команды добавления/удаления маршрута нужно вводить от имени *root*. В современных системах под именем *root* входить необязательно: нужно использовать или команду *sudo*, или команду *su* для получения *root*-доступа.

Параметр *операция* может принимать два значения: *add* (добавить маршрут) и *del* (удалить маршрут). Параметр *тип* необязательный, он задает тип маршрута: *-net* (маршрут к сети), *-host* (маршрут к узлу) или *default* (маршрут по умолчанию). Параметр *адресат* содержит адрес сети (если задается маршрут к сети), адрес узла (при добавлении маршрута к сети) или вообще не указывается, если задается маршрут по умолчанию.

Параметр *шлюз* задает IP-адрес (или доменное имя) шлюза. Последние два параметра — *метрика* и *dev* необязательны. Параметр *метрика* задает максимальное число переходов (через маршрутизаторы) на пути к адресату. В Linux он необязательный, в отличие от других ОС. Последний параметр имеет смысл задавать, если

в системе установлено несколько сетевых интерфейсов и нужно указать, через какой именно сетевой интерфейс следует отправить пакеты по указанному маршруту.

Команда удаления маршрута выглядит так:

```
# route del адрес
```

В других UNIX-системах есть параметр `-f`, удаляющий все маршруты (`route -f`), но в Linux такого параметра нет. Следовательно, для очистки всей таблицы маршрутизации вам нужно будет ввести серию команд `route del`. Изменять таблицу маршрутизации нужно только, зарегистрировавшись на компьютере локально. При удаленной регистрации (например, по `ssh`) легко удалить ошибочно маршрут, по которому вы "вошли в систему". О последствиях такого действия, думаю, говорить не нужно.

Примеры использования команды `route`:

```
route add -net 192.76.16.0 netmask 255.255.255.0 dev eth0
```

Добавляет маршрут к сети 192.76.16.0 (сеть класса C, о чем свидетельствует сетевая маска, заданная параметром `netmask`) через устройство `eth0`. Шлюз не указан, просто все пакеты, адресованные сети 192.76.16.0, будут отправлены на интерфейс `eth0`.

```
route add -net 192.16.16.0 netmask 255.255.255.0 gw 192.76.16.1
```

Добавляет маршрут к сети 192.16.16.0 через маршрутизатор 192.76.16.1. Сетевой интерфейс задавать не обязательно, но можно и указать при особом желании.

```
route add default gw gate1
```

Добавляет маршрут по умолчанию. Все пакеты будут отправлены компьютеру с именем `gate1`. Обратите внимание: мы указываем доменное имя узла вместо IP-адреса.

```
route add -net 10.1.0.0 netmask 255.0.0.0 reject
```

Добавляет запрещающий маршрут. Отправка пакетов по этому маршруту (в сеть 10.1.0.0) запрещена.

Итак, мы добавили необходимые маршруты, пропинговали удаленные узлы, все работает. Теперь нужно сохранить установленные маршруты, чтобы они были доступны при следующей загрузке системы. Для этого в openSUSE нужно отредактировать файл `/etc/sysconfig/network/routes` (`/etc/route.conf` — в старых версиях). Мы уже рассматривали этот файл (см. главу 27 и листинг 37.2), поэтому переходим сразу к другому дистрибутиву.

В Fedora/CentOS/ASP Linux (и других Red Hat-совместимых дистрибутивах) статические маршруты хранятся в файле `/etc/sysconfig/static-routes`. Строки в этом файле имеют вид:

```
any net адрес_сети netmask маска gw адрес_шлюза
```

Здесь `any` означает любой интерфейс. Можно указать конкретный интерфейс, например:

```
eth0 net 192.168.2.0 netmask 255.255.255.0 gw 192.168.1.1
```

Файл `/etc/sysconfig/static-routes` по умолчанию отсутствует, при необходимости его нужно создать самостоятельно.

В Debian/Ubuntu статические маршруты прописываются вместе с конфигурацией сетевого интерфейса в файле `/etc/network/interfaces`. С помощью параметров `up` и `down` этого файла можно задать команды, которые будут выполняться при "поднятии" (`up`) и "закрытии" (`down`) интерфейса. После параметров `up` и `down` может следовать любая Linux-команда. Обычно это команда `route`. Например, при запуске интерфейса `eth0` будет добавлен статический маршрут к сети `192.168.3.0` через шлюз `192.168.1.2`:

```
up route add -net 192.168.3.0 netmask 255.255.255.0 gw 192.168.1.2
```

Можно также добавить маршрут по умолчанию:

```
up route add default gw 192.168.1.2
```

При "закрытии" интерфейса нужно удалить маршруты, которые использовали этот интерфейс, для этого служит параметр `down`:

```
down route del default gw 192.168.1.2
```

```
down route del -net 192.168.3.0
```

Подробное описание файла `/etc/network/interfaces` вы найдете по адресу:

<http://www.dkws.org.ua/index.php?page=show&file=a/ubuntu/network-interfaces>

37.4. Включение IPv4-переадресации или превращение компьютера в шлюз

Основное предназначение шлюза (маршрутизатора) — это пересылка (`forwarding`) пакетов. Чтобы включить пересылку пакетов протокола IPv4 (IPv4 forwarding), нужно записать значение `1` в файл `/proc/sys/net/ipv4/ip_forward`:

```
# echo 1 > /proc/sys/net/ipv4/ip_forward
```

Но включить пересылку мало — нужно еще сохранить это значение, иначе при перезагрузке будет восстановлено значение по умолчанию (`0`). Для этого нужно в файл `/etc/sysctl.conf` добавить строку:

```
net.ipv4.ip_forward=0
```

В некоторых дистрибутивах, например в openSUSE, можно воспользоваться конфигуратором (**YaST | Сетевые настройки | Маршрутизация**), см. рис. 37.2. В некоторых переадресацию можно включить путем редактирования конфигурационных файлов сети, например в Fedora (см. листинг 37.1).

Подробно о настройке системы с помощью псевдофайловой системы `/proc` см. в *приложении 3*.

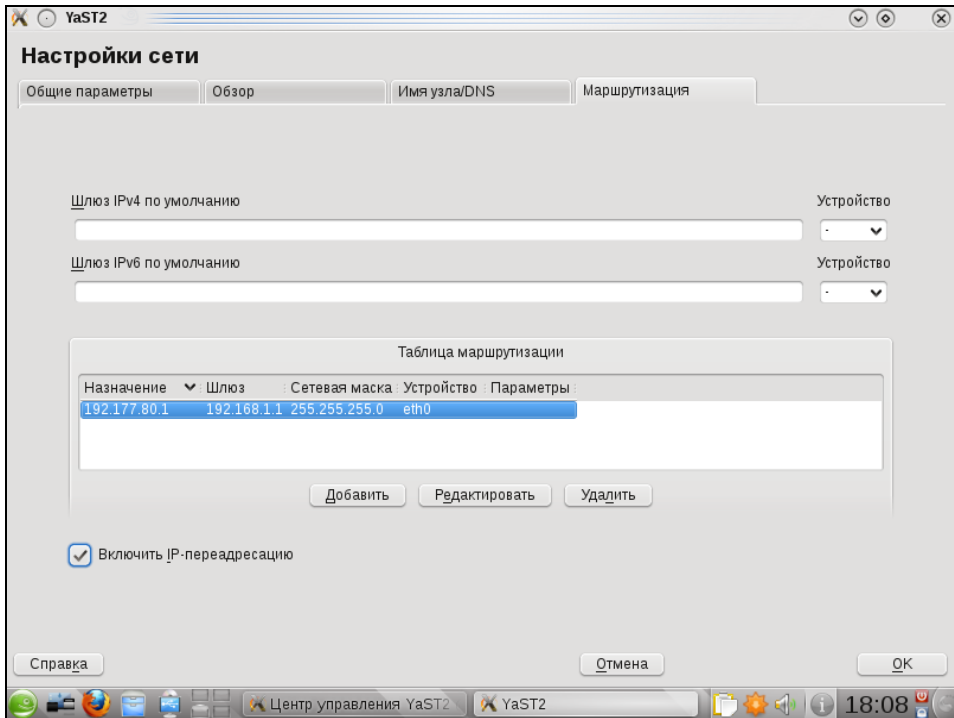


Рис. 37.2. Включение IP-переедресации в openSUSE 11.3

37.5. Настройка брандмауэра

37.5.1. Что такое брандмауэр

Брандмауэр (он же *firewall*, *бастион*, *межсетевой экран*) предназначен для защиты внутренней сети (или всего одного компьютера, напрямую подключенного к Интернету) от вторжения извне. С помощью брандмауэра вы можете контролировать доступ пользователей Интернета к узлам вашей внутренней сети. Также можно контролировать доступ локальных пользователей к ресурсам Интернета — например, вы можете запретить им посещать определенные узлы с целью экономии трафика.

Прежде чем перейти к настройке межсетевого экрана, определимся с терминологией и, в частности, с понятием "шлюз". *Шлюзом* называется компьютер, предоставляющий компьютерам локальной сети доступ к Интернету. Шлюз выполняет как бы маршрутизацию пакетов. Но не нужно путать шлюз с обычным маршрутизатором. Маршрутизатор осуществляет простую пересылку пакетов, поэтому его можно использовать для соединения сетей одного типа, например локальной и локальной, глобальной и глобальной. А шлюз служит для соединения сетей разных

типов, например локальной и глобальной, как в нашем случае. Конечно, сейчас можно встретить маршрутизаторы с функцией шлюза, но это уже, скорее, аппаратные шлюзы, чем простые маршрутизаторы. Поэтому часто термины "маршрутизатор" и "шлюз" употребляются как синонимы, хотя это не совсем так.

Сложность в соединении сетей разных типов заключается в различной адресации. Как мы знаем, в локальной сети обычно используются локальные адреса, которые не допустимы в Интернете, например: 192.168.*.* (сеть класса С), 10.*.*.* (сеть класса А) и 172.16.*.*—172.31.*.* (класс В). Поэтому шлюз должен выполнить преобразование сетевого адреса (NAT, Network Address Translation). Сейчас поясню, что это такое. Предположим, у нас есть шлюз и локальная сеть с адресами 192.168.*.*. Реальный IP-адрес (который можно использовать в Интернете) есть только у шлюза, пусть это 193.254.219.1. У всех остальных компьютеров — локальные адреса, поэтому при всем своем желании они не могут обратиться к интернет-узлам.

У нашего шлюза два сетевых интерфейса. Один из них, пусть `ppp0`, используется для подключения к Интернету. Его IP-адрес, как уже было отмечено, 93.254.219.1. Для подключения к локальной сети используется другой сетевой интерфейс — `eth0` (сетевая плата) с IP-адресом 192.168.1.1.

Все узлы нашей локальной сети используют в качестве шлюза компьютер с адресом 192.168.1.1. Это означает, что все запросы будут переданы на узел 192.168.1.1. Запросы передаются в виде:

Назначение: IP-адрес узла Интернета

Источник: адрес компьютера локальной сети, пусть 192.168.1.10

Наш шлюз принимает запрос и перезаписывает его так:

Назначение: IP-адрес узла Интернета

Источник: 193.254.219.1

То есть шлюз подменяет адрес источника, устанавливая в качестве этого адреса свой реальный IP-адрес, иначе бы любой интернет-узел не принял бы запрос с локального адреса. Получив ответ от узла, он направляет его нашему узлу:

Назначение: 192.168.1.10

Источник: IP-адрес узла Интернета

Нашему локальному узлу "кажется", что он получил ответ непосредственно от узла Интернета, а на самом деле ответ приходит от шлюза.

Теперь, когда мы разобрались с теорией, самое время перейти к практике.

37.5.2. Цепочки и правила

Основная задача брандмауэра — это фильтрация пакетов, которые проходят через сетевой интерфейс. При поступлении пакета брандмауэр анализирует его и затем принимает решение: принять пакет (АССЕПТ) или избавиться от него (DROП). Брандмауэр может выполнять и более сложные действия, но часто ограничиваются именно этими двумя действиями.

Прежде чем брандмауэр примет решение относительно пакета, пакет должен пройти по цепочке правил. Каждое правило состоит из условия и действия (цели). Если пакет соответствует условию правила, то выполняется указанное в правиле действие. Если пакет не соответствует условию правила, он передается следующему правилу. Если же пакет не соответствует ни одному из правил цепочки, выполняется действие по умолчанию.

Вроде бы все понятно, но чтобы лучше закрепить знания, рассмотрим табл. 37.2, демонстрирующую принцип работы цепочки правил.

Таблица 37.2. Цепочка правил

Номер правила	Условие	Действие (цель)
1	Пакет от 192.168.1.0	ACCEPT
2	Пакет от 192.168.0.0	DROP
3	Пакет для 192.168.2.0	ACCEPT
DEFAULT	*	DROP

Предположим, что пакет пришел из сети 192.168.4.0 для узла 192.168.1.7 (это наша сеть). Пакет не соответствует первому правилу (отправитель не из сети 192.168.1.0), поэтому он передается правилу 2. Пакет не соответствует и этому правилу. Пакет адресован компьютеру 192.168.1.7, а не компьютеру из сети 192.168.2.0, поэтому он не соответствует и третьему правилу. Брандмауэру остается применить правило по умолчанию — пакет будет отброшен (действие DROP).

Цепочки правил собираются в три основные таблицы:

- ❖ **filter** — таблица фильтрации, основная таблица;
- ❖ **nat** — таблица NAT, используется при создании пакетом нового соединения;
- ❖ **mangle** — используется, когда нужно произвести специальные действия над пакетом.

ПРИМЕЧАНИЕ

Ранее брандмауэр в Linux поддерживал только цепочки правил и назывался `ipchains`, сейчас брандмауэр поддерживает и цепочки правил, и таблицы цепочек и называется `iptables`. Это примечание сделано, чтобы вы понимали разницу между старым брандмауэром `ipchains` (ядра 2.2 и ниже) и новым `iptables` (ядра 2.4 и выше).

Если необходимо, вы можете создать собственные таблицы. В состав таблицы входят три цепочки:

- ❖ **INPUT** — для входящих пакетов;
- ❖ **OUTPUT** — для исходящих пакетов;
- ❖ **FORWARD** — для пересылаемых (транзитных) пакетов.

Над пакетом можно выполнить следующие действия:

- ❖ *<имя цепочки>* — пакет будет отправлен для обработки в цепочку с указанным именем;
- ❖ **ACCEPT** — принять пакет;
- ❖ **DROP** — отбросить пакет, после этого пакет удаляется, больше над ним не выполняются какие-либо действия;
- ❖ **MASQUERADE** — скрыть IP-адрес пакета.

Это не все действия, но пока нам больше знать не нужно. На рис. 37.3 изображена схема обработки пакета. Входящий пакет (на схеме **Пакет IN**) поступает в цепочку **PREROUTING** таблицы **mangle**. После чего (если он не был отброшен правилами таблицы **mangle**) пакет обрабатывается правилами цепочки **PREROUTING**, но таблицы **nat**. На этом этапе проверяется, нужно ли модифицировать назначение пакета (этот вид NAT называется Destination NAT, DNAT).

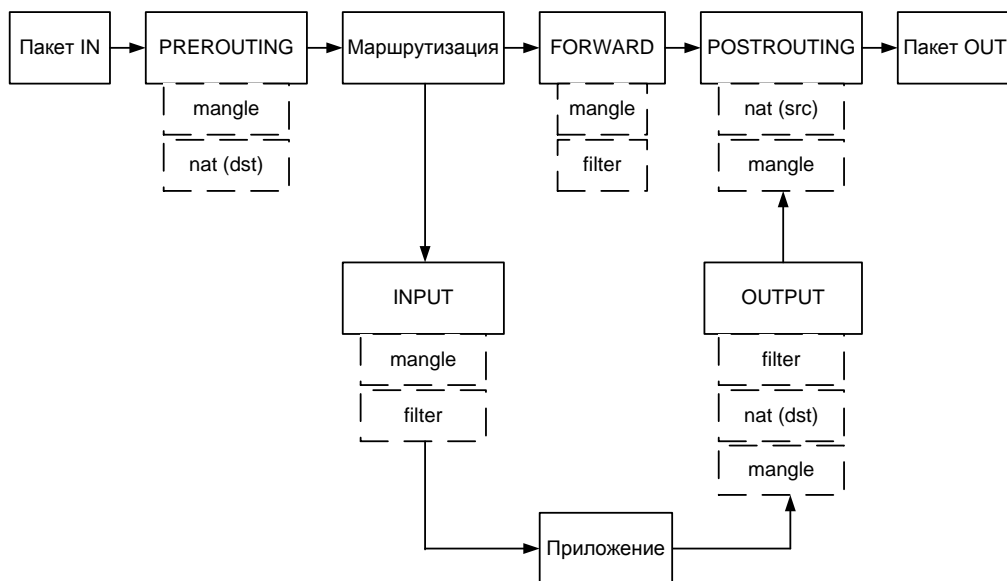


Рис. 37.3. Схема обработки пакета

Затем пакет может быть направлен либо в цепочку **INPUT** (если получателем пакета является этот компьютер), либо в цепочку **FORWARD** (если пакет нужно передать другому компьютеру).

Если получатель компьютера — сам шлюз (на нем может быть запущен, например, почтовый или Web-сервер), то пакет сначала обрабатывается правилами цепочки **INPUT** таблиц **mangle** и **filter**. Если пакет не был отброшен, он передается приложению (например, почтовому серверу). Приложение получило пакет, обработало его и отправляет ответный пакет. Этот пакет обрабатывается цепочкой **OUTPUT** таблиц **mangle**, **nat** и **filter**. Далее пакет отправляется на цепочку **POSTROUTING** и обрабатывается правилами таблиц **mangle** и **nat**.

Если пакет нужно передать другому компьютеру, то он обрабатывается правилами цепочки **FORWARD** таблиц **mangle** и **filter**, а после этого к нему применяются правила цепочки **POSTROUTING**. На этом этапе используется подмена источника пакета (этот вид NAT называется Source NAT, SNAT).

После всех правил пакет "выжил"? Тогда он становится исходящим пакетом (на схеме **Пакет OUT**) и отправляется в сеть.

37.5.3. Использование iptables

Теперь, когда мы разобрались с правилами и цепочками, самое время научиться использовать iptables. Для себя сразу определитесь, что вы настраиваете. Можно настраивать просто брандмауэр, защищающий локальный компьютер от всевозможных атак. А можно настраивать шлюз сети, предоставляющий всем остальным компьютерам сети доступ к Интернету. В последнем случае нужно включить IP-переадресацию (IPv4-forwarding). О том, как это сделать, было сказано ранее. В большинстве случаев хватит вот такой команды:

```
sudo sysctl -w net.ipv4.ip_forward="1"
sudo echo 1 > /proc/sys/net/ipv4/ip_forward
```

Сейчас можно перейти к iptables. Для изменения правил брандмауэра нужны полномочия root, поэтому все команды iptables нужно вводить или через команду `sudo` (для этого ваш пользователь должен иметь право использовать `sudo`), или с предварительно полученными полномочиями root (команда `su`).

Для добавления правила в цепочку используется команда:

```
sudo iptables -A цепочка правило
```

Например:

```
sudo iptables -A INPUT правило
```

Данная команда добавит правило в цепочку **INPUT** таблицы **filter** (это таблица по умолчанию). Если вы желаете добавить правило в другую таблицу, нужно указать ее в параметре `-t`:

```
sudo iptables -t таблица -A цепочка правило
```

Например:

```
sudo iptables -t nat -A INPUT правило
```

Действие по умолчанию задается ключом `-P`:

```
sudo iptables -P INPUT DROP
```

Обычно устанавливаются вот такие действия по умолчанию:

```
sudo iptables -P INPUT DROP
sudo iptables -P FORWARD ACCEPT
sudo iptables -P OUTPUT DROP
```

Перед рассмотрением табл. 37.3 нужно поговорить о фазах установки TCP-соединения. Соединение устанавливается в три этапа (фазы). Сначала первый ком-

пьютер отправляет второму компьютеру SYN-пакет, запрашивая открытие соединения. Второй компьютер отправляет ему подтверждение SYN-пакета — ACK-пакет. После этого соединение считается установленным (ESTABLISHED). Открытое, но не установленное соединение (когда компьютеры обмениваются пакетами SYN-ACK) называется новым (NEW). Слова в скобках я привел не просто так, а для понимания табл. 37.3. В таблице при описании параметров будут указываться не полные команды iptables, а только их фрагменты, имеющие отношение к тому или иному параметру.

Таблица 37.3. Параметры фильтрации пакетов

Параметр	Описание
--source	Позволяет указать источник пакета. Можно указывать, как доменное имя компьютера (den.dkws.org.ua), так и его IP-адрес (192.156.1.1) и даже набор адресов (192.168.1.0/255.255.255.0). Пример: iptables -A FORWARD --source 192.168.1.11 ...
--destination	Задаёт назначение (адрес получателя) пакета. Синтаксис такой же, как и у --source
-protocol (или -p)	Задаёт протокол. Чаще всего работают с tcp, icmp или udp, но можно указать любой протокол, определенный в файле /etc/protocols. Также можно указать all, что означает все протоколы. Примеры: iptables -A FORWARD -protocol tcp ... iptables -A FORWARD -p tcp ...
--source-port (или --sport)	Определяет порт отправителя. Данная опция может использоваться только вместе с параметром -p. Например: iptables -A FORWARD -p tcp -source-port 23 ...
--destination-port (или --dport)	Задаёт порт-назначение. Опция возможна только с параметром -p. Синтаксис такой же, как и в случае с -source-port
-state	Позволяет отфильтровать пакеты по состоянию. Параметр -state доступен только при загрузке модуля state с помощью другого параметра -m state. Состояния пакета: <ul style="list-style-type: none"> ▪ NEW — новое соединение (еще не установленное); ▪ ESTABLISHED — установленное соединение; ▪ RELATED — пакеты, которые не принадлежат соединению, но связаны с ним; ▪ INVALID — неопознанные пакеты. Пример: iptables -A FORWARD -m state -state RELATED,INVALID

Таблица 37.3 (окончание)

Параметр	Описание
<code>-in-interface</code> (или <code>-i</code>)	Определяет интерфейс, по которому прибыл пакет. Пример: <code>iptables -A FORWARD -i eth1</code>
<code>-out-interface</code> (или <code>-o</code>)	Определяет интерфейс, по которому будет отправлен пакет: <code>iptables -A FORWARD -o ppp0</code>
<code>-tcp-flags</code>	Производит фильтрацию по TCP-флагам (man iptables)

Ранее мы познакомились с основными действиями iptables. В табл. 37.4 представлены все действия iptables (цели iptables). Действие задается параметром `-j`.

Таблица 37.4. Цели iptables

Действие	Описание
ACCEPT	Принять пакет. При этом пакет уходит из этой цепочки и передается дальше
DROP	Уничтожить пакет
REJECT	<p>Уничтожает пакет и сообщает об этом отправителю с помощью ICMP-сообщения. Параметр <code>-reject-with</code> позволяет уточнить тип ICMP-сообщения:</p> <ul style="list-style-type: none"> ■ <code>icmp-host-unreachable</code> — узел недоступен; ■ <code>icmp-net-unreachable</code> — сеть недоступна; ■ <code>icmp-port-unreachable</code> — порт недоступен; ■ <code>icmp-proto-unreachable</code> — протокол недоступен. <p>По умолчанию отправляет сообщение о недоступности порта. Но, используя сообщение <code>icmp-host-unreachable</code>, можно сбить злоумышленника с толку. Предположим, что вы просто решили отбрасывать неугодные вам пакеты (действие DROP). Но злоумышленник будет посылать и посылать вам эти пакеты, чтобы брандмауэр только и делал, что занимался фильтрацией и удалением этих пакетов (один из видов атаки на отказ). А если вы ответите сообщением <code>icmp-host-unreachable</code>, то злоумышленник будет думать, что узел недоступен, т. е. что компьютер выключен либо он уже достиг своей цели — добился отказа компьютера. С другой стороны, помните, что данное действие порождает ответный ICMP-пакет, что нагружает исходящий канал, который в некоторых случаях (например, одностороннее спутниковое соединение) очень "узкий". Если злоумышленник пришлет вам 1 млн пакетов, то вы должны будете отправить 1 млн сообщений в ответ. Подумайте, готовы ли вы к такой нагрузке на исходящий канал</p>

Таблица 37.4 (окончание)

Действие	Описание
LOG	Заносит информацию о пакете в протокол. Полезно использовать для протоколирования возможных атак — если вы подозреваете, что ваш узел атакуется кем-то. Также полезно при отладке настроек брандмауэра
RETURN	Возвращает пакет в цепочку, откуда он прибыл. Действие возможно, но лучше его не использовать, т. к. легко ошибиться и создать непрерывный цикл: вы отправляете пакет обратно, а он опять следует на правило, содержащее цель RETURN
SNAT	Выполняет подмену IP-адреса отправителя (Source NAT). Используется в цепочках POSTROUTING и OUTPUT таблицы nat
DNAT	Выполняет подмену адреса получателя (Destination NAT). Используется только в цепочке POSTROUTING таблицы nat
MASQUERADE	Похож на SNAT, но "забывает" про все активные соединения при потере интерфейса. Используется при работе с динамическими IP-адресами, когда происходит "потеря" интерфейса при изменении IP-адреса. Применяется в цепочке POSTROUTING таблицы nat

37.5.4. Шлюз своими руками

Создать шлюз в Linux очень просто. Сейчас вы сами в этом убедитесь. Гораздо сложнее правильно настроить его, чтобы шлюз не только выполнял свою непосредственную функцию (т. е. передачу пакетов из локальной сети в Интернет и обратно), но и защищал сеть.

В последнее время очень популярны DSL-соединения, поэтому будем считать, что для подключения к Интернету используется именно DSL-соединение. Хотя вся разница только в названии интерфейса — `ppp0`. Вполне может быть, что у вас иная конфигурация. Например, у вас может быть два сетевых интерфейса — `eth0` и `eth1`. Первый "смотрит" в локальную сеть, а второй — подключен к Интернету. Тогда и правила будете формировать, исходя из того, что соединение с Интернетом происходит по интерфейсу `eth1`.

В случае с DSL-соединением у нас тоже будут два сетевых адаптера. Первый (`eth0`) будет подключен к локальной сети, а ко второму (`eth1`) будет подключен DSL-модем. Перед настройкой шлюза проверьте, действительно ли это так. Вполне может оказаться, что сетевая плата, к которой подключен DSL-модем — это интерфейс `eth0`, а не `eth1`. В этом случае вам нужно или изменить названия интерфейсов при формировании правил, или просто подключить модем к другому сетевому адаптеру.

IP-адрес DSL-соединения будет динамическим (обычно так оно и есть), а вот IP-адрес сетевого адаптера, обращенного к локальной сети, пусть будет `192.168.1.1`.

Вы можете использовать и другой адрес (адрес должен быть локальным, если только у вас нет подсети с реальными IP-адресами).

Итак, мы настроили локальную сеть, узнали имена сетевых адаптеров, включили IP-переедресацию. Осталось только ввести команду:

```
sudo iptables -t nat -A POSTROUTING -o ppp0 -j MASQUERADE
```

Установите на всех компьютерах вашей сети IP-адрес 192.168.1.1 в качестве шлюза по умолчанию (можно настроить DHCP-сервер, чтобы не настраивать все компьютеры вручную) и попробуйте пропинговать с любого узла какой-то сайт. Оказывается, вы прочитали всю эту главу ради одной строчки. Так и есть. Но, сами понимаете, на этом настройка шлюза не заканчивается. Нужно еще защитить вашу сеть. Как минимум, вам нужно установить следующие действия по умолчанию:

```
sudo iptables -P INPUT DROP
sudo iptables -P FORWARD ACCEPT
sudo iptables -P OUTPUT DROP
```

Разрешим входящие соединения на шлюз только от узлов нашей внутренней сети 192.168.1.0:

```
sudo iptables -A INPUT -i eth0 --source 192.168.1.0/24 --match state --state NEW,ESTABLISHED -j ACCEPT
```

Нужно также установить правило для цепочки **OUTPUT** — оно разрешает шлюзу отвечать компьютерам нашей локальной сети:

```
sudo iptables -A OUTPUT -o eth0 --destination 192.168.1.0/24 --match state --state NEW,ESTABLISHED -j ACCEPT
```

Будьте внимательны при указании имен интерфейсов и IP-адресов. Очень легко запутаться, а потом полчаса разбираться, почему шлюз не работает.

Нам осталось только запретить соединения из Интернета (компьютеры нашей сети смогут устанавливать соединения с серверами Интернета, зато интернет-пользователи не смогут установить соединения с компьютерами нашей сети):

```
sudo iptables -A FORWARD -i eth0 --destination 192.168.1.0/24 --match state --state ESTABLISHED -j ACCEPT
```

У нас получилась простенькая конфигурация. Компьютеры нашей сети могут выступать инициаторами соединения, а интернет-узлы могут передавать данные в нашу сеть только в том случае, если инициатором соединения выступил локальный компьютер.

Но и это еще не все. Как вы уже догадались, поскольку мы не сохранили правила брандмауэра, при перезагрузке компьютера его придется настраивать заново. Поскольку мне лень описывать настройку брандмауэра (сохранение и восстановление правил) в каждом дистрибутиве (пусть это будет ваше домашнее задание), рассмотрим универсальный способ. Способ заключается в создании bash-сценария, вызывающего необходимые нам команды настройки iptables. После написания сценария вам останется вызвать его при загрузке системы. А для этого нужно изучить строение системы инициализации в вашем дистрибутиве (см. главу 17).

Вместо того чтобы объяснять вам, как вызвать сценарий, загружающий правила брандмауэра (с этим вы и сами разберетесь), я лучше приведу сценарий (понятно, с комментариями), реализующий более сложную конфигурацию iptables. Данный сценарий будет не только выполнять все функции шлюза, но и защищать сеть от разного рода атак (листинг 37.4). Сценарий лучше сразу поместить в каталог /etc/init.d (это моя вам подсказка) и сделать исполняемым:

```
# touch /etc/init.d/firewall_start
# chmod +x /etc/init.d/firewall_start
```

Листинг 37.4. Сценарий firewall_start

```
# Путь к iptables
IPT="/sbin/iptables"

# Сетевой интерфейс, подключенный к Интернету
INET="ppp0"

# Номера непривилегированных портов
UPOINTS="1024:65535"

# Включаем IPv4-forwarding (чтобы не думать, почему шлюз не работает)
echo 1 > /proc/sys/net/ipv4/ip_forward

# Удаляем все цепочки и правила
$IPT -F
$IPT -X

# Действия по умолчанию.
$IPT -P INPUT DROP
$IPT -P FORWARD ACCEPT
$IPT -P OUTPUT DROP

# Разрешаем все пакеты по интерфейсу lo (обратная петля)
$IPT -A INPUT -i lo -j ACCEPT
$IPT -A OUTPUT -o lo -j ACCEPT

# Запрещаем любые новые соединения с любых интерфейсов, кроме lo,
# с нашим компьютером
$IPT -A INPUT -m state ! -i lo --state NEW -j DROP
$IPT -A INPUT -s 127.0.0.1/255.0.0.0 ! -i lo -j DROP

# Отбрасываем все пакеты со статусом INVALID
$IPT -A INPUT -m state --state INVALID -j DROP
$IPT -A FORWARD -m state --state INVALID -j DROP
```

```
# Принимаем все пакеты из уже установленного соединения
# Состояние ESTABLISHED
$IPT -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
$IPT -A FORWARD -m state --state ESTABLISHED,RELATED -j ACCEPT

# Мой провайдер использует IP-адреса из сети 10.0.0.0 для доступа к своим
# локальным ресурсам. Ничего не поделаешь, нужно разрешить эти адреса,
# иначе даже не сможет войти в билинговую систему. В вашем случае, может и
# не нужно будет добавлять следующее правило, а может, у вас будет такая же
# ситуация, но адрес подсети будет другим
$IPT -t nat -I PREROUTING -i $INET -s 10.0.0.1/32 -j ACCEPT

# Защищаемся от SYN-наводнения (довольно популярный вид атаки)
$IPT -A INPUT -p tcp ! --syn -m state --state NEW -j DROP
$IPT -A OUTPUT -p tcp ! --syn -m state --state NEW -j DROP

# Защищаемся от UDP-наводнения
$IPT -A INPUT -p UDP -s 0/0 --dport 138 -j DROP
$IPT -A INPUT -p UDP -s 0/0 --dport 113 -j REJECT
$IPT -A INPUT -p UDP -s 0/0 --sport 67 --dport 68 -j ACCEPT
$IPT -A INPUT -p UDP -j RETURN
$IPT -A OUTPUT -p UDP -s 0/0 -j ACCEPT

# Защищаемся от ICMP-перенаправления.
# Данный вид атаки может использоваться злоумышленником
# для перенаправления своего трафика через вашу машину
$IPT -A INPUT --fragment -p ICMP -j DROP
$IPT -A OUTPUT --fragment -p ICMP -j DROP

# Но обычные ICMP-сообщения мы разрешаем
$IPT -A INPUT -p icmp -m icmp -i $INET --icmp-type source-quench -j ACCEPT
$IPT -A OUTPUT -p icmp -m icmp -o $INET --icmp-type source-quench -j ACCEPT

# Разрешаем себе пинговать интернет-узлы
$IPT -A INPUT -p icmp -m icmp -i $INET --icmp-type echo-reply -j ACCEPT
$IPT -A OUTPUT -p icmp -m icmp -o $INET --icmp-type echo-request -j ACCEPT

# Разрешаем передачу ICMP-сообщения "неверный параметр"
$IPT -A INPUT -p icmp -m icmp -i $INET --icmp-type parameter-problem -j ACCEPT
$IPT -A OUTPUT -p icmp -m icmp -o $INET --icmp-type parameter-problem -j
ACCEPT
```

```
# Запрещаем подключение к X.Org через сетевые интерфейсы.
$IPT -A INPUT -p tcp -m tcp -i $INET --dport 6000:6063 -j DROP --syn

# Указываем порты, открытые в системе, но которые должны быть
# закрыты на сетевых интерфейсах. Я пропишу только порт 5501:
$IPT -A INPUT -p tcp -m tcp -m multiport -i $INET -j DROP --dports 5501

# Разрешаем DNS
$IPT -A OUTPUT -p udp -m udp -o $INET --dport 53 --sport $UPTS -j ACCEPT
$IPT -A OUTPUT -p tcp -m tcp -o $INET --dport 53 --sport $UPTS -j ACCEPT
$IPT -A INPUT -p udp -m udp -i $INET --dport $UPTS --sport 53 -j ACCEPT
$IPT -A INPUT -p tcp -m tcp -i $INET --dport $UPTS --sport 53 -j ACCEPT

# Разрешаем AUTH-запросы к удаленным серверам, но запрещаем такие
# запросы к своему компьютеру
$IPT -A OUTPUT -p tcp -m tcp -o $INET --dport 113 --sport $UPTS -j ACCEPT
$IPT -A INPUT -p tcp -m tcp -i $INET --dport $UPTS --sport 113 -j ACCEPT ! --syn
$IPT -A INPUT -p tcp -m tcp -i $INET --dport 113 -j DROP

# Далее мы открываем некоторые порты, необходимые
# для функционирования сетевых служб.

# FTP-клиент (порт 21)
$IPT -A OUTPUT -p tcp -m tcp -o $INET --dport 21 --sport $UPTS -j ACCEPT
$IPT -A INPUT -p tcp -m tcp -i $INET --dport $UPTS --sport 21 -j ACCEPT ! --syn

# SSH-клиент (порт 22)
$IPT -A OUTPUT -p tcp -m tcp -o $INET --dport 22 --sport $UPTS -j ACCEPT
$IPT -A INPUT -p tcp -m tcp -i $INET --dport $UPTS --sport 22 -j ACCEPT ! --syn
$IPT -A OUTPUT -p tcp -m tcp -o $INET --dport 22 --sport 1020:1023 -j ACCEPT
$IPT -A INPUT -p tcp -m tcp -i $INET --dport 1020:1023 --sport 22 -j ACCEPT ! --syn

# SMTP-клиент (порт 25)
$IPT -A OUTPUT -p tcp -m tcp -o $INET --dport 25 --sport $UPTS -j ACCEPT
$IPT -A INPUT -p tcp -m tcp -i $INET --dport $UPTS --sport 25 -j ACCEPT ! --syn

# HTTP/HTTPS-клиент (порты 80, 443)
$IPT -A OUTPUT -p tcp -m tcp -m multiport -o $INET --sport $UPTS -j ACCEPT -dports 80,443
```



```
$IPT -A INPUT -p tcp -m tcp -m multiport -i $INET --dport $UPOINTS -j ACCEPT --  
sports 80,443 ! --syn
```

```
# POP-клиент (порт 110)
```

```
$IPT -A OUTPUT -p tcp -m tcp -o $INET --dport 110 --sport $UPOINTS -j ACCEPT
```

```
$IPT -A INPUT -p tcp -m tcp -i $INET --dport $UPOINTS --sport 110 -j ACCEPT ! -  
--syn
```

```
# Разрешаем прохождение DHCP-запросов через iptables
```

```
# Необходимо, если IP-адрес динамический
```

```
$IPT -A OUTPUT -p udp -m udp -o $INET --dport 67 --sport 68 -j ACCEPT
```

```
$IPT -A INPUT -p udp -m udp -i $INET --dport 68 --sport 67 -j ACCEPT
```

Вот практически и все. Приведенное в этой книге описание iptables нельзя назвать полным. Но если описать iptables полностью, то можно смело издавать отдельную книгу под названием "Брандмауэр в Linux". В Интернете я нашел одно из наиболее полных руководств по iptables на русском языке. Так вот, если его распечатать, то оно займет 121 страницу формата A4. Лист книги обычно меньше A4, поэтому смело можно говорить, что объем нашей книги составил бы около 200 страниц. Адрес этого руководства: <http://www.opennet.ru/docs/RUS/iptables/>.

Вот еще одна очень хорошая статья по iptables:

<http://ru.wikipedia.org/wiki/Iptables>

А для пользователей Debian и Ubuntu будет полезным следующее руководство:

http://www.linux.by/wiki/index.php/Debian_Firewall

ГЛАВА 38



Сервер времени

38.1. Проблема синхронизации времени

Компьютерные таймеры работают с большой погрешностью — это вам не швейцарские часы. На домашнем компьютере проблема синхронизации времени не очень актуальна, но на производстве очень важно, чтобы на всех компьютерах было указано одно и то же время. Например, Иванов запустил на своем компьютере выполнение технологической операции в 11:45, а сервер запротоколировал, что операция началась в 11:47 или даже в 11:53. Когда нет никаких ЧП, то на данную погрешность времени никто не обратит внимание. А вот когда произойдет что-то неприятное и будет начато служебное расследование, все погрешности во времени только усложнят ситуацию.

В этой главе мы рассмотрим настройку собственного сервера времени. Работать все будет так: мы настроим сервер времени и запустим синхронизацию времени на всех рабочих станциях сети. Конечно, "синхронизаторы" будут "направлены" на наш сервер времени. Все, что вам остается — это следить за тем, чтобы время на сервере времени было точным, хотя это уже и не столь важно — ведь время на всех компьютерах сети будет одинаковым.

Спрашивается, зачем это нужно, если есть сервер **time.windows.com**, который можно использовать для нашей цели? Этот сервер использовать не всегда возможно. Ведь не у всех компьютеров есть доступ к Интернету, и перенастраивать ради этого брандмауэр не совсем корректно, тем более, что проблему очень легко и просто можно решить локально, своими силами, не прибегая к помощи Microsoft.

38.2. Настройка сервера и Linux-клиентов

Одним из самых удачных серверов времени для Linux является **ntpd** (пакет называется **ntp**). На каждой Linux-машине нужно установить этот пакет. Одна Linux-машина будет эталонной, т. е. на ней время будете устанавливать вы. Если вручную

устанавливать время вам лень, тогда можно настроить синхронизацию с каким-то удаленным сервером (с тем же **time.windows.com**).

Итак, после установки ntpd приступим к его настройке. На всех клиентах конфигурационный файл /etc/ntp.conf будет выглядеть, как показано в листинге 38.1.

Листинг 38.1. Файл /etc/ntp.conf для клиента

```
restrict default ignore
restrict 127.0.0.1

# это IP-адрес эталонной машины
server 192.168.1.1

driftfile /etc/ntp/drift
broadcastdelay 0.008
authenticate yes
keys /etc/ntp/keys
```

Данный конфигурационный файл вполне приемлем. Единственное, что нужно изменить, — это IP-адрес эталонной машины (сервера). У вас он будет, скорее всего, другим.

Теперь перейдем к конфигурационному файлу сервера (листинг 38.2).

Листинг 38.2. Файл /etc/ntp.conf для сервера

```
restrict default ignore
restrict 127.0.0.1

# Можно указать IP-адрес какого-то удаленного сервера,
# если не хочется вручную устанавливать время
server 127.127.1.0    # локальное время
fludge 127.127.1.0 stratum 10
driftfile /etc/ntp/drift
broadcastdelay 0.008
authenticate yes
keys /etc/ntp/keys
```

После того как сервер и клиенты настроены, на всех компьютерах нужно запустить сервер ntpd:

```
# service ntpd start
```

или

```
# /etc/init.d/ntpd start
```

38.3. Настройка Windows-клиентов

Тут все просто: щелкните правой кнопкой мыши на индикаторе времени в области уведомлений Windows и выберите команду **Настройка даты/времени**. В появившемся окне перейдите на вкладку **Время Интернета** (рис. 38.1), нажмите кнопку **Изменить параметры**.

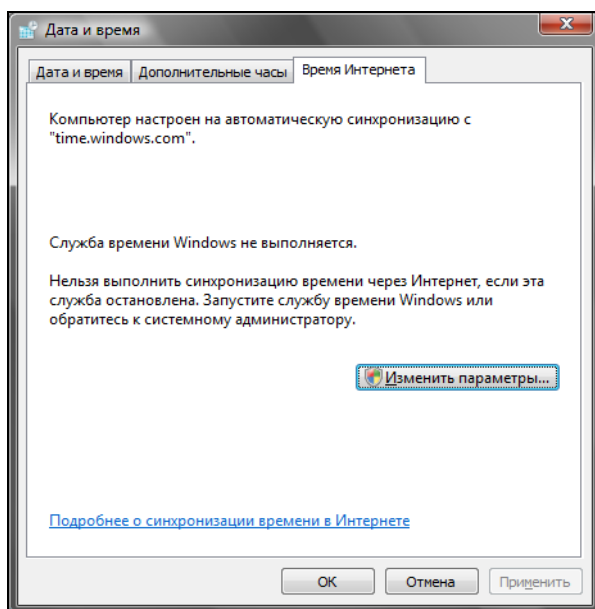


Рис. 38.1. Настройка Windows-клиента

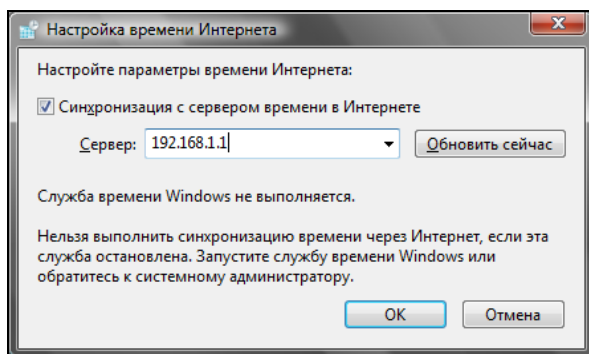


Рис. 38.2. Установка эталонной машины

В окне **Настройка времени Интернета** (рис. 38.2) включите синхронизацию, установите IP-адрес эталонной машины (в нашем случае 192.168.1.1) и нажмите кнопку **Обновить сейчас**. Затем нажмите кнопку **ОК**. Обратите внимание на

рис. 38.2: в нашем случае служба времени выключена, поэтому обновление времени невозможно. Для ее запуска нажмите кнопку **Пуск**, введите команду `services.msc` и нажмите клавишу <Enter>. В появившемся окне (рис. 38.3) включите службу времени Windows.

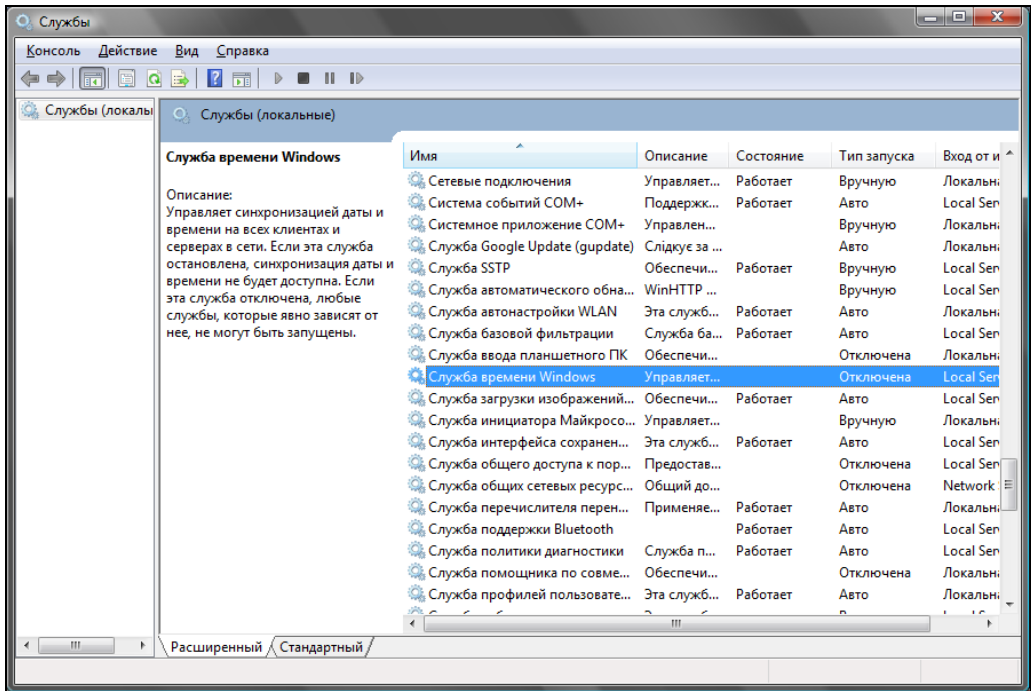


Рис. 38.3. Включение службы времени Windows

ГЛАВА 39



Виртуальные частные сети

39.1. Для чего нужна виртуальная частная сеть

Предположим, что пользователям нашей организации нужно обращаться к ресурсам корпоративной сети, когда они находятся за ее пределами, например в другом городе. Первое, что приходит в голову, — это настроить сервер удаленного доступа (Remote Access Server, RAS или dial-in сервер). Пользователь с помощью модема "дозванивается" к серверу удаленного доступа, сервер идентифицирует пользователя, после чего последний подключается к сети предприятия и работает в ней как обычно (разве что скорость передачи данных будет значительно ниже, чем обычно).

Но использование RAS — затея довольно дорогая и неудобная. Во-первых, нужно организовать модемный пул, а это недешево и накладно: нужна или многоканальная линия, или же несколько телефонных линий (обеспечить одновременную работу нескольких пользователей). Во-вторых, нужно оплачивать междугородные и даже международные звонки пользователей (для удобства самих пользователей нужно организовать callback-режим). В-третьих, далеко не всегда у пользователя есть возможность подключиться к телефонной сети. В-четвертых, RAS не может обеспечить связь нескольких филиалов компании.

Выходом из сложившейся ситуации является использование виртуальной частной сети (Virtual Private Network, VPN). В случае с VPN данные передаются по каналам Интернета. Это существенно упрощает и удешевляет нашу задачу. Доступ к Интернету есть везде, пользователи сами смогут выбрать провайдера и способ (соответственно, и скорость) подключения к Интернету. Понятно, чтобы оградить себя от перехвата информации, данные при передаче через VPN шифруются. Вот основные преимущества VPN.

- ❖ Не нужно никакое дополнительное оборудование (модемный пул) и какие-либо дополнительные ресурсы (например, многоканальная телефонная линия). Все, что нужно, — это подключение к Интернету, а поскольку нет такого частного

предприятия, которое не было бы подключено к Интернету, будем считать, что все необходимое для организации VPN уже есть.

- ◆ Безопасность передачи данных по сравнению с обычной передачей данных по Интернету.
- ◆ Возможность как соединения филиалов компании, так и подключения отдельных пользователей к корпоративной сети. При этом мобильные пользователи могут подключаться к Интернету с помощью GPRS, что делает подключение к VPN максимально гибким — пользователю не придется искать свободную телефонную розетку.

39.2. Необходимое программное обеспечение

Для организации соединений типа "сеть—сеть", т. е. для связи двух сетей одной компании в одну VPN используется протокол IpSec. В Linux его реализация называется OpenS/WAN (<http://www.openswan.org>). OpenS/WAN — это потомок самой популярной Linux-реализации IpSec — FreeS/WAN (<http://www.freeswan.org>). Проект OpenS/WAN — более современный и поддерживает ядра 2.4 и 2.6, в то время как FreeS/WAN поддерживает только старые ядра (2.2 и 2.4).

Для подключения удаленных пользователей к корпоративной сети используется протокол PPTP (Point to Point Tunneling Protocol). Настройку этого протокола мы также рассмотрим в этой главе.

39.3. Канал для передачи данных VPN

39.3.1. Соединение "сеть—сеть"

Предположим, что нам нужно объединить два офиса компании. Один пусть находится в Москве, а другой, скажем, во Владивостоке. Для связи офисов будут использоваться VPN-маршрутизаторы. В роли такого маршрутизатора может выступать любой Linux-компьютер с установленным программным обеспечением (OpenS/WAN).

Важно правильно выбрать канал для подключения VPN-маршрутизатора к Интернету. Канал не должен быть "узким", иначе данные между филиалами компании будут передаваться очень медленно. Обычные выделенные линии, понятно, отпадают. Также отпадают различные беспроводные соединения, вроде RadioEthernet. Конечно, беспроводное соединение — это удобно, но его качество очень сильно зависит от "зашумленности" эфира и от погоды. Если на улице плохая погода, скорость заметно падает, не говоря уже о том, что беспроводная точка доступа может сгореть во время грозы. Беспроводные соединения можно охарактеризовать как не

очень надежные — ведь чуть ли не после каждой грозы вам придется менять точку доступа. Выключить точку доступа на время непогоды может себе позволить отдельный пользователь, но не предприятие. Поэтому беспроводные соединения нас не устраивают.

Если нужна надежность и независимость, можно выбрать синхронные (двухнаправленные) спутниковые соединения, но в этом случае оборудование, да и содержание такого канала (лицензия на передатчик, оплата) будет совсем не дешевым. Если организация может себе позволить такое удовольствие, то, уверен, не пожалеет о таком решении.

Наиболее оптимальным для многих организаций будет использование DSL-соединений. Вполне приличная скорость соединения — до нескольких мегабит в секунду, да и такие соединения надежнее беспроводных (имеется в виду RadioEthernet, а не спутниковое соединение).

Если вы остановили свой выбор на DSL-соединении, то нужно выбирать SDSL-соединение: оно синхронное, т. е. скорость приема и передачи данных будет одинаковой. Кроме SDSL-соединения, есть еще и ADSL-соединение — оно асинхронно, и скорость приема в несколько раз ниже скорости передачи. Такое соединение больше подходит для домашнего использования, чем для связи офисов компании.

39.3.2. Соединение "клиент—сеть"

В этом случае пользователь может сам выбрать то соединение, которое предпочтительно для него. В большинстве случаев мобильные пользователи в командировках будут использовать GPRS-соединения. Да, недостатков у GPRS достаточно (самые главные — низкая скорость передачи данных и дороговизна), но зато подключиться к родной сети можно практически откуда угодно — мобильный телефон всегда под рукой, лишь бы быть в зоне покрытия мобильного оператора.

39.4. Настройка соединения "сеть—сеть"

39.4.1. Установка OpenS/WAN

По адресу <http://www.openswan.org/download/binaries/> вы найдете уже откомпилированные пакеты OpenS/WAN для дистрибутивов Fedora Core, Mandriva, Mandrake, OpenWRT, Red Hat, RHEL, SUSE.

Перед установкой пакетов, возможно, понадобится перекомпиляция ядра (см. главу 53). Вам нужно включить опции PF_KEY, AH, ESP и все опции в группе CryptoAPI.

ПРИМЕЧАНИЕ

Мы рассматриваем установку OpenS/WAN 2.4.x в систему с ядром 2.6.x.

39.4.2. Немного терминологии

Предположим, что нам нужно связать два офиса компании. Один будет находиться в Москве, другой — во Владивостоке. Посмотрите на рис. 39.1. Москва находится на западе (слева), Владивосток — на востоке (справа), поэтому московская сеть будет называться *left*, а сеть Владивостока — *right*. Хотя это не принципиально. Просто так принято.



Рис. 39.1. Пример VPN-сети

Понятно, что VPN-маршрутизатор будет выходить в Интернет через какой-то обычный маршрутизатор. В терминологии VPN маршрутизатор, через который подключается к Интернету левый VPN-маршрутизатор, называется *leftnexthop* (соответственно правый — *rightnexthop*).

39.4.3. Генерирование ключей

Перед настройкой OpenS/WAN нужно сгенерировать ключи на обоих VPN-маршрутизаторах. Для этого на каждом VPN-маршрутизаторе введите команду:

```
# ipsec newhostkey
```

Просмотреть ключ можно с помощью одной из команд:

```
# ipsec showhostkey --left
```

```
# ipsec showhostkey --right
```

39.4.4. Конфигурационный файл

OpenS/WAN использует один основной файл конфигурации — `/etc/ipsec/ipsec.conf`. Данный файл состоит из трех разделов: общие настройки (`config setup`), настройки по умолчанию (`conn %default`) и настройки соединения (`conn <название соединения>`). Понятно, что последних разделов может быть несколько, поскольку каждый такой раздел задает параметры конкретного соединения.

Рассмотрим пример раздела, содержащий параметры по умолчанию (листинг 39.1).

Листинг 39.1. Параметры по умолчанию

```
config setup
# указывает интерфейсы, которые будут использоваться для VPN-соединений
interfaces=%defaultroute

# управляют протоколированием KLIPS (Kernel IP Security) и демоном Pluto
klipsdebug=none
plutodebug=none

# Эти параметры лучше не изменять
plutoload=%search
plutostart=%search
```

ПРИМЕЧАНИЕ

Табуляция перед именем директивы (именно директивы, а не раздела файла конфигурации) обязательна! Иначе при обработке конфигурационного файла будет выведено сообщение об ошибке: "... has wrong number of fields ...".

Обратите внимание на параметр `interfaces`. В большинстве случаев подойдет значение `%defaultroute`, но можно указать имя интерфейса явно, например:

```
interfaces="ipsec0=ppp1"
```

Теперь рассмотрим раздел с настройками по умолчанию. Данный раздел вообще не обязателен, но если он есть, то обычно в нем указываются две директивы: `authby` и `keyingtries`. Первая задает метод аутентификации, а вторая — количество попыток установки соединения (по умолчанию 0, т. е. соединение будет устанавливаться бесконечно, пока не будет установлено). Пример данного раздела приведен в листинге 39.2.

Листинг 39.2. Пример раздела настроек по умолчанию

```
conn %default
    authby=rsasig
    keyingtries=3
```

Основной раздел конфигурационного файла описывает VPN-соединения. Для написания этого раздела нам нужно обратиться к рис. 39.1.

В конфигурационном файле обоих VPN-маршрутизаторов нужно указать сведения, приведенные в табл. 39.1.

Таблица 39.1. Параметры VPN-соединения типа "сеть—сеть"

Директива	Назначение
left	IP-адрес левого VPN-маршрутизатора (вместо него можно указать значение %defaultroute). В нашем случае это 192.168.1.1
leftsubnet	IP-адрес левой сети. В нашем случае это 192.168.1.0/24
leftnexthop	IP-адрес левого маршрутизатора (можно указать значение %defaultroute)
leftrsasigkey	Ключ левого маршрутизатора (можно узнать с помощью команды <code>ipsec showhostkey --left</code>)
leftid	Идентификатор левой сети. Например, @moscow.firma.ru. Можно в разделе <code>config setup</code> указать опцию <code>uniqueids=yes</code> . Это избавит вас от указания идентификаторов сети
right	IP-адрес правого VPN-маршрутизатора (вместо него можно указать значение %defaultroute). В нашем случае это 192.168.2.1
rightsubnet	IP-адрес правой сети (192.168.2.0/24)
rightnexthop	IP-адрес правого маршрутизатора (можно указать значение %defaultroute)
rightrsasigkey	Ключ правого маршрутизатора (можно узнать с помощью команды <code>ipsec showhostkey --right</code>)
rightid	Идентификатор правой сети. Например, @vladivostok.firma.ru
leftfirewall	Если левая сторона защищена брандмауэром, то нужно установить значение <code>yes</code> для этой директивы
auto	Управляет автоматической установкой соединений. Если указать <code>auto=start</code> , соединение будет автоматически установлено. Чтобы директива <code>auto</code> работала, нужно в <code>config setup</code> указать <code>plutostart=%search</code>

Пример раздела параметров соединения приведен в листинге 39.3.

Листинг 39.3. Пример раздела параметров VPN-соединения

```
conn my_vpn
    left=192.168.1.1
    leftsubnet=192.168.1.0/24
    leftnexthop=10.0.0.1
    leftrsasigkey= 0sAQtyjh9345...
    leftid=@moscow.firma.ru

    right=192.168.2.1
    rightsubnet=192.168.2.0/24
```

```
rightnexthop=10.1.0.1
rightrsasigkey=0sAQ65jh92...
rightid=@vladivostok.firma.ru

auto=start
```

Полная версия файла `ipsec.conf` представлена в листинге 39.4.

Листинг 39.4. Пример файла `ipsec.conf`

```
config setup
    # указывает интерфейсы, которые будут использоваться для VPN-соединений
    interfaces=%defaultroute

    # управляют протоколированием KLIPS (Kernel IP Security) и демоном Pluto
    klipsdebug=none
    plutodebug=none

    # Эти параметры лучше не изменять
    plutoload=%search
    plutostart=%search

conn %default

    authby=rsasig
    keyingtries=3

conn my_vpn
    left=192.168.1.1
    leftsubnet=192.168.1.0/24
    leftnexthop=10.0.0.1
    leftrsasigkey= 0sAQtyjh9345...
    leftid=@moscow.firma.ru

    right=192.168.2.1
    rightsubnet=192.168.2.0/24
    rightnexthop=10.1.0.1
    rightrsasigkey=0sAQ65jh92...
    rightid=@vladivostok.firma.ru

    auto=start
```

39.4.5. Установка VPN-соединения

Для запуска демона OpenS/WAN нужно выполнить команду:

```
ipsec start
```

При этом будут запущены все соединения, для которых вы указали `auto=start`. Команду `ipsec start` нужно выполнить на обеих сторонах.

Проверить, запущено ли соединение, можно с помощью команды:

```
ipsec look
```

39.4.6. Настройка *iptables*

Для работы IpSec нужно должным образом настроить `iptables`, а именно: разрешить порт 500, который используется для обмена сертификатами и ключами:

```
iptables -A INPUT -i eth0 -p udp -s $IP --sport 500 --dport 500 -j ACCEPT
iptables -A OUTPUT -o eth0 -p udp -d $IP --sport 500 --dport 500 -j ACCEPT
```

```
iptables -A INPUT -i eth0 -p 50 -s $IP -j ACCEPT
iptables -A OUTPUT -o eth0 -p 50 -d $IP -j ACCEPT
```

```
iptables -A INPUT -i eth0 -p 51 -s $IP -j ACCEPT
iptables -A OUTPUT -o eth0 -p 51 -d $IP -j ACCEPT
```

```
iptables -A FORWARD -p all -s 192.168.2.0/24 -d 192.168.1.0/24 -j ACCEPT
iptables -A FORWARD -p all -s 192.168.1.0/24 -d 192.168.2.0/24 -j ACCEPT
```

В нашем случае:

- ❖ `$IP` — это IP-адрес шлюза на противоположной стороне, т. е. 192.168.2.1 для левой стороны и 192.168.1.1 для правой стороны;
- ❖ `eth0` — это внешний интерфейс сети;
- ❖ `ipsec0` — это VPN-интерфейс.

39.5. Настройка соединения "клиент—сеть"

В этом разделе мы рассмотрим настройку соединения "клиент—сеть", когда нужно обеспечить подключение отдельного пользователя к локальной сети. Для настройки соединения такого типа используется протокол PPTP.

Нам понадобятся следующие пакеты:

- ❖ `pptpd` или `pptp-server` — PPTP-сервер;
- ❖ `pptp-linux`, `pptp-client`, `pptp-adsl` — PPTP-клиент.

Названия пакетов (зависят от дистрибутива), правда, могут немного отличаться. Найти данные пакеты можно с помощью сайтов <http://rpmfind.net> (или rpm.pbone.net), если у вас Red Hat-совместимый дистрибутив, и <http://packages.ubuntu.com>, если у вас Ubuntu или Debian.

Еще нам нужен пакет `ppp`, но в большинстве случаев он устанавливается по умолчанию, поэтому вам нужно только проверить его наличие в вашей системе.

Нужно отметить, что VPN-сервер в современных дистрибутивах настраивается на порядок проще, чем в дистрибутивах, основанных на ядре 2.4. Ведь в старых дистрибутивах вам надо было добавить поддержку MPPE (патч) для `ppp` и ядра, а в новых дистрибутивах, основанных на ядре 2.6, всего этого делать не нужно. Даже не нужно перекомпилировать ядро, поскольку в большинстве случаев расширение MPPE включено по умолчанию. Почему в большинстве случаев? Откуда же я знаю, какой у вас дистрибутив? Может, у вас какой-то экзотический дистрибутив, разработчики которого посчитали, что MPPE вам не нужен, и отключили его.

ПРИМЕЧАНИЕ

Microsoft Point-to-Point Encryption (MPPE) — протокол шифрования данных, используемый поверх соединений PPP

39.5.1. Редактирование конфигурационных файлов

После установки пакета `ppptd` (или `pptp-server`) можно отредактировать его конфигурационный файл `/etc/pptpd.conf` (листинг 39.5).

Листинг 39.5. Конфигурационный файл `/etc/pptpd.conf`

```
speed 115200
option /etc/ppp/options.vpn
debug
#
remoteip 192.168.1.12-22
```

Чтобы основной конфигурационный файл был компактным, дополнительные опции вынесем в файл `/etc/ppp/options.vpn` (листинг 39.6). Думаю, назначение этих опций понятно и без моих комментариев. IP-адреса VPN-клиентов вам нужно изменить (параметр `remoteip`). В этом примере предполагается, что максимум может быть 10 VPN-клиентов, которым будут назначены IP-адреса из диапазона 192.168.1.12—192.168.1.22.

Теперь отредактируем файл `/etc/ppp/options.vpn` (понятно, его еще нужно создать).

Листинг 39.6. Конфигурационный файл /etc/ppp/options.vpn

```
ipparam PoPToP

lock
mtu 1000
mru 1000

ms-dns 192.168.1.1
name server.com
# Нужен для того, чтобы после подключения к серверу удаленный пользователь
# мог обратиться к узлам виртуальной сети.
# Чтобы эта опция работала правильно, нужно включить
# форвардинг пакетов (Ipv4 Forwarding)
proxyarp
# Нужен, если вы планируете использовать аутентификацию
auth
# Если аутентификация не нужна, тогда укажите
#noauth

# Протоколы аутентификации, если noauth, то они не нужны
refuse-pap
refuse-chap
refuse-chapms
require-mschap-v2

ipcp-accept-local
ipcp-accept-remote
lcp-echo-failure 30
lcp-echo-interval 5
# Если deflate = 0, то сжатие не используется
#deflate 0
```

Этот файл конфигурации немного сложнее, чем предыдущий. Если особо разбираться что есть что не хочется, тогда просто измените IP-адрес DNS-сервера (опция `ms-dns`) и имя узла (опция `name`). В не самых свежих версиях `ppp` вместо опций `refuse-pap`, `refuse-chap`, `refuse-chapms`, `require-mschap-v2` нужно использовать опции (соответственно):

```
-pap
-chap
-chapms
+chapms-v2
```

Данные опции управляют аутентификацией VPN-пользователя. Мы используем протокол аутентификации MS CHAP v2, как самый безопасный.

Практически все настроено. Осталось только отредактировать файл `/etc/ppp/options`. Добавьте в него всего одну опцию:

```
lock
```

Имена VPN-пользователей можно определить в файле `/etc/ppp/chap-secrets`. Формат этого файла такой:

```
имя сервер.домен пароль IP
```

ПРИМЕЧАНИЕ

Если аутентификация вам не нужна, то не надо редактировать файл `/etc/ppp/chap-secrets` и создавать VPN-пользователей. При настройке клиента не нужно указывать имя пользователя и пароль, а также следует отключить шифрование.

Вот небольшой пример:

```
vpn1 server.com "" *
```

`vpn1` — имя пользователя, `server.com` — имя нашего VPN-сервера. Пароль мы указали пустой, это означает, что пароль будет браться из `/etc/shadow`. IP-адрес мы тоже не указывали — VPN-пользователь сможет аутентифицироваться с любого IP. Пользователь `vpn1` должен существовать в системе (добавить пользователя можно командой `adduser`).

Вот сейчас все готово. Для запуска PPTP-сервера используется команда:

```
service pptpd start (или /etc/init.d/pptpd start)
```

Не забудьте разрешить на брандмауэре прохождение пакетов на порты 47 и 1723 (ведь наверняка в вашей сети есть брандмауэр):

```
iptables --append INPUT --protocol 47 --jump ACCEPT
```

```
iptables --append INPUT --protocol tcp --match tcp --destination-port 1723 --  
jump ACCEPT
```

Подробно настройка брандмауэра для взаимодействия с PPTP описана на страничке: <http://asplinux.net/node/1918>. А на следующей страничке подробно описаны ошибки PPTP-клиента, что пригодится при анализе журналов системы в случае возникновения проблем:

http://pptpclient.sourceforge.net/howto-diagnosis.phtml#running_pptp

39.5.2. Настройка Linux-клиента

В этом разделе мы рассмотрим настройку клиента, работающего под управлением Windows. Понятно, что на VPN-сервере не обязательно устанавливать PPTP-клиент.

1. Для установки VPN-клиента нужно установить пакет `pptp-linux` (рис. 39.2) или `pptp-client`. После установки запустите сценарий `pptp-command`. Сценарий отобразит меню из четырех пунктов, нужно выбрать пункт **Setup** (рис. 39.3).
2. Вы увидите еще одно меню, в котором нужно выбрать пункт **Manage CHAP secrets** (см. рис. 39.3), после чего следует выбрать команду **Add a New CHAP secret**. Сценарий попросит ввести вас имя локальной машины, имя удаленной машины (вводить необязательно), имя пользователя и пароль.

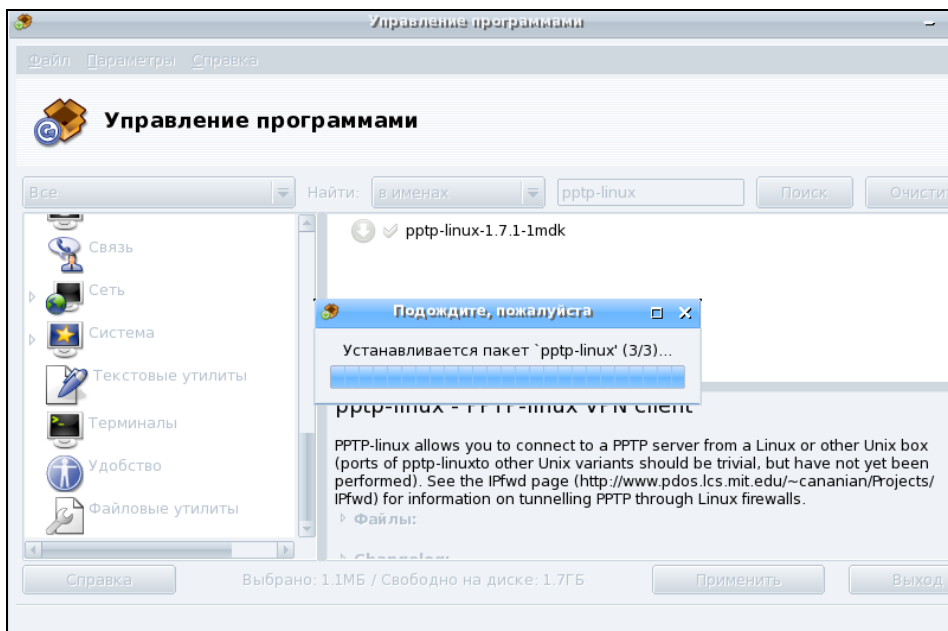


Рис. 39.2. Установка пакета `pptp-linux` в Linux Mandriva

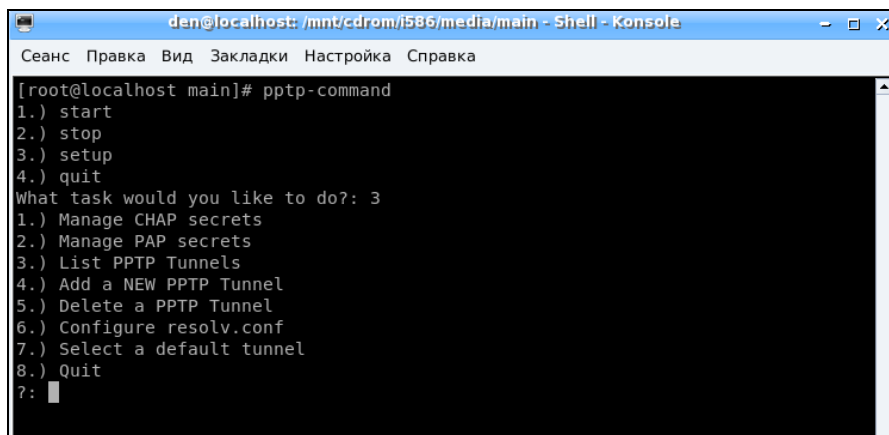


Рис. 39.3. Сценарий `pptp-command`

3. Затем вы вернетесь в меню настройки. Нужно будет выбрать команду **Add a new PPTP tunnel**, а затем пункт **Other**. Вам нужно ввести следующую информацию: имя и IP-адрес VPN-сервера, а также параметры маршрутизации.
4. После этого нужно выбрать пункт меню **Configure resolv.conf** и указать IP-адреса DNS-серверов.
5. Настройка почти закончена. Следует в уже хорошо знакомом нам меню выбрать команду **Select a default tunnel**, позволяющую выбрать туннель по умолчанию. Нужно выбрать туннель, который вы только что создали.
6. Для подключения к VPN надо опять запустить сценарий `pptp-command` и выбрать команду **Start**. Понятно, что перед этим вы должны подключиться к Интернету.

В Ubuntu в окне **Сетевые соединения** (NetworkManager) есть вкладка **VPN**, но кнопка **Добавить** неактивна. А все потому, что не установлены пакеты, реализующие поддержку VPN. Чтобы настроить VPN-соединение через NetworkManager, вам нужно скачать с **packages.ubuntu.com** пакеты `network-manager-pptp` и `network-manager-pptp-gnome`. Конечно, если вы используете не протокол PPTP, а какой-нибудь другой, тогда вам нужно скачать и установить соответствующие пакеты, например `network-manager-vpnc`. Если быть предельно точным, то нужно установить следующие пакеты (и все пакеты, от которых зависят эти пакеты):

- ✧ `pptp-linux`;
- ✧ `network-manager-pptp`;
- ✧ `network-manager-pptp-gnome`;
- ✧ `network-manager-vpnc`;
- ✧ `network-manager-openvpn`;
- ✧ `network-manager-openvpn-gnome`;
- ✧ `network-manager-strongswan`.

Первые три пакета необходимы для поддержки PPTP, четвертый — для протокола VPNC (Cisco), следующие два — для протокола OpenVPN, последний — для strongSwan.

А как же скачать пакеты, если соединение с Интернетом осуществляется по VPN? Есть три варианта:

- ✧ перезагрузиться в Windows, если она установлена, подключиться к Интернету, скачать пакеты;
- ✧ найти другой компьютер или использовать альтернативное соединение (например, 3G-модем или мобильный телефон);
- ✧ использовать Denix (<http://denix.dkws.org.ua>): в нем по умолчанию есть все необходимые пакеты для установки VPN-соединений.

Ради справедливости нужно отметить, что в Ubuntu 10.04 появилась поддержка VPN "из коробки", но поддерживается только PPTP, а вот если нужно установить соединение по другому протоколу, см. *действия ранее*.

Если у вас все-таки Ubuntu 9.x и переходить на 10.x вы пока не планируете, обязательно прочитайте мою статью по настройке VPN-соединения в Ubuntu 9:

<http://www.dkws.org.ua/index.php?page=show&file=a/ubuntu/vpn-ubuntu9>

Скачанные пакеты можно установить командой `dpkg` (подробно об этом вы прочитаете в соответствующей главе книги).

При настройке VPN-соединения в Mandriva могут возникнуть некоторые проблемы. Решение проблемы описано по адресу:

<http://www.dkws.org.ua/phpbb2/viewtopic.php?p=18432>

Подробно процесс настройки VPN-соединения в Mandriva описан по адресу:

http://wiki.mandriva.com/ru/Настройка_VPN_в_Mandriva

39.5.3. Настройка Windows-клиента

В Windows 2000/XP

Настройка VPN-подключения в Windows 2000/XP намного проще. Во-первых, вам не нужно устанавливать VPN-клиент — он уже входит в состав Windows и установлен по умолчанию. Во-вторых, интерфейс мастера новых подключений в Windows дружелюбнее и привычнее интерфейса `pptp-command` (хотя, кому как).

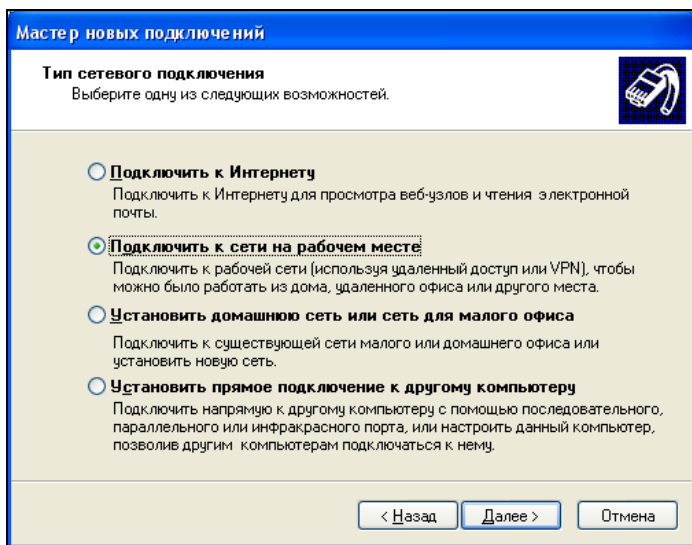


Рис. 39.4. Мастер новых подключений

Чтобы создать VPN-подключение, выполните команду меню **Пуск | Настройка | Сетевые подключения | Создание нового подключения**. В окне Мастера новых подключений выберите **Подключить к сети на рабочем месте**

(рис. 39.4). Затем выберите **Подключение к виртуальной частной сети** (рис. 39.5).

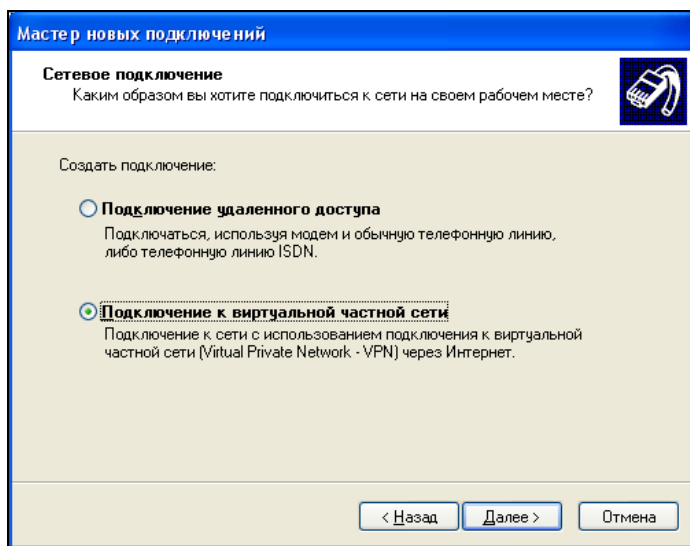


Рис. 39.5. Подключение к VPN

Перед подключением к VPN нужно установить соединение с Интернетом, поэтому мастер новых подключений предложит вам выбрать соединение с Интернетом, которое будет установлено перед подключением к VPN (рис. 39.6).

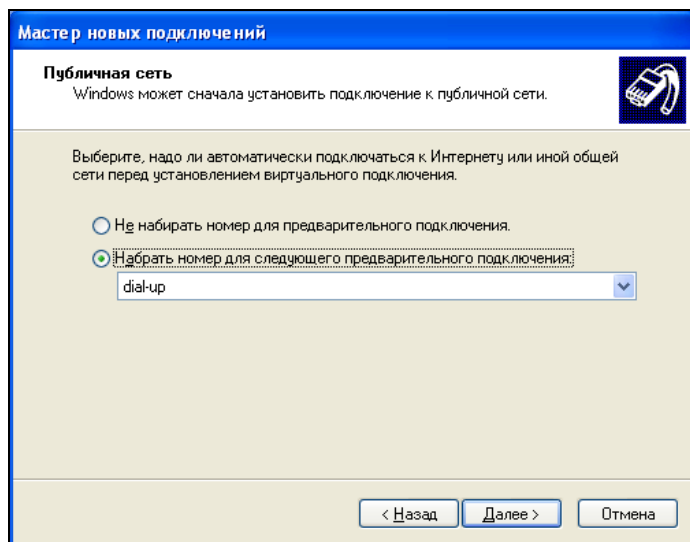


Рис. 39.6. Выбор подключения к Интернету

Затем вам останется лишь ввести параметры подключения: имя VPN-сервера (рис. 39.7), имя пользователя и пароль. Вместо имени VPN-сервера можно ввести его IP-адрес.

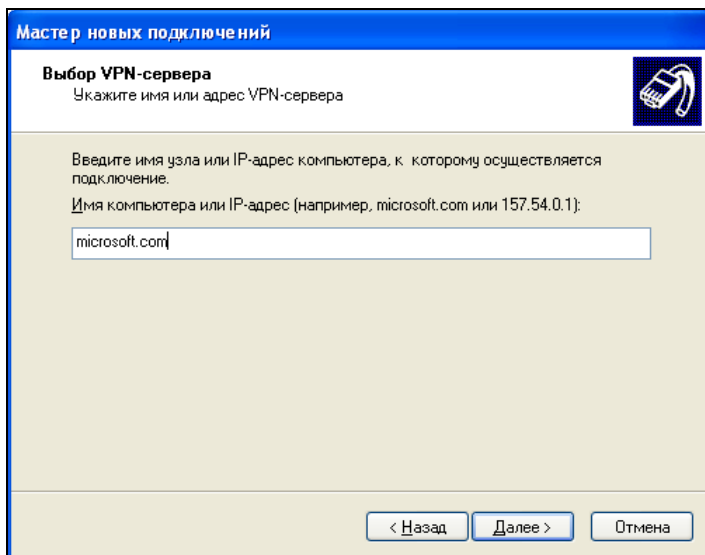


Рис. 39.7. Ввод имени VPN-сервера

После создания VPN-подключения его можно запустить из системной папки **Сетевые подключения** (рис. 39.8).

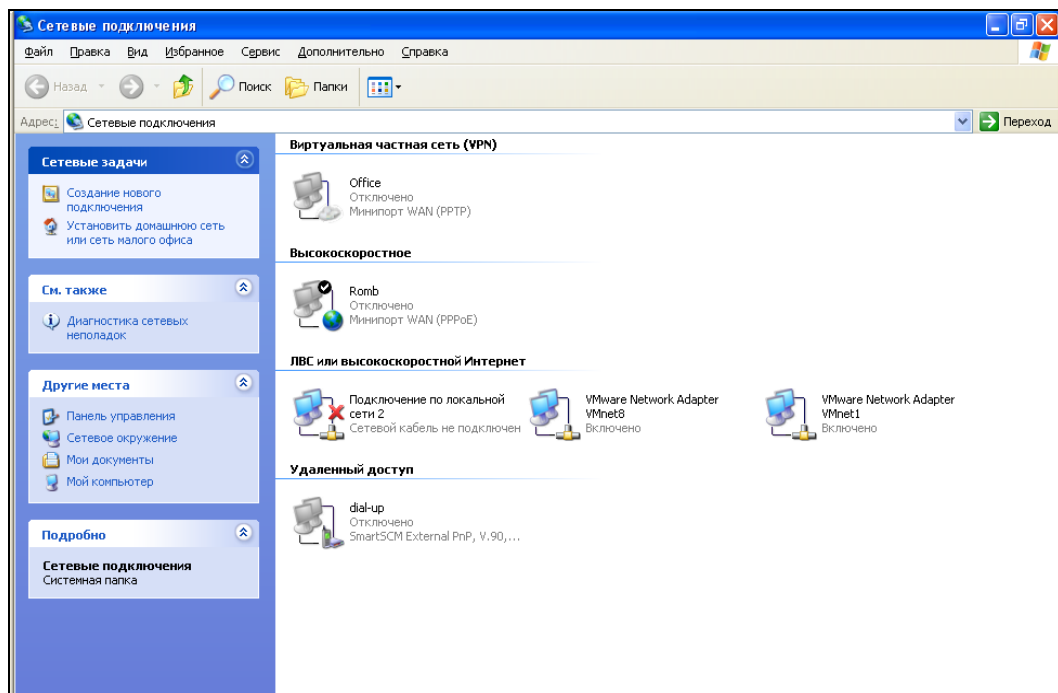


Рис. 39.8. Сетевые подключения

В Windows Vista/Windows 7

В Windows Vista/7 VPN-подключение создается аналогично. Выберите команду **Пуск | Подключение**. В открывшемся окне (рис. 39.9) перейдите по ссылке **Установка подключения или сети**.

В открывшемся окне выберите **Подключение к рабочему месту** (рис. 39.10).

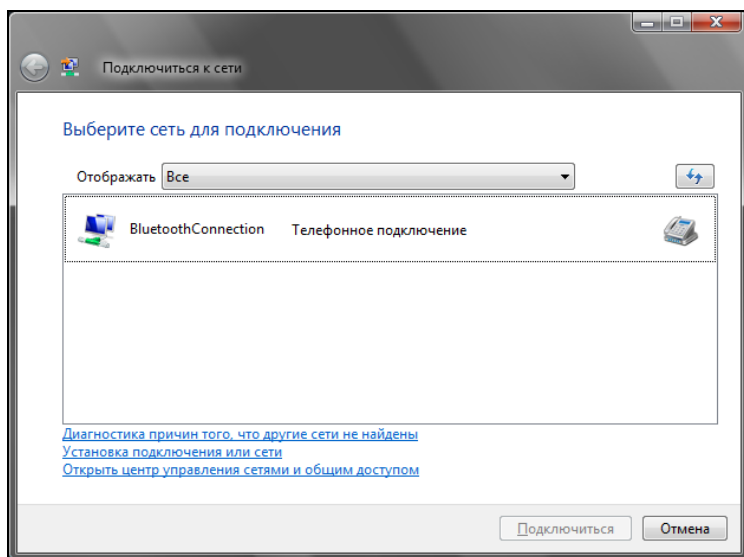


Рис. 39.9. Подключиться к сети

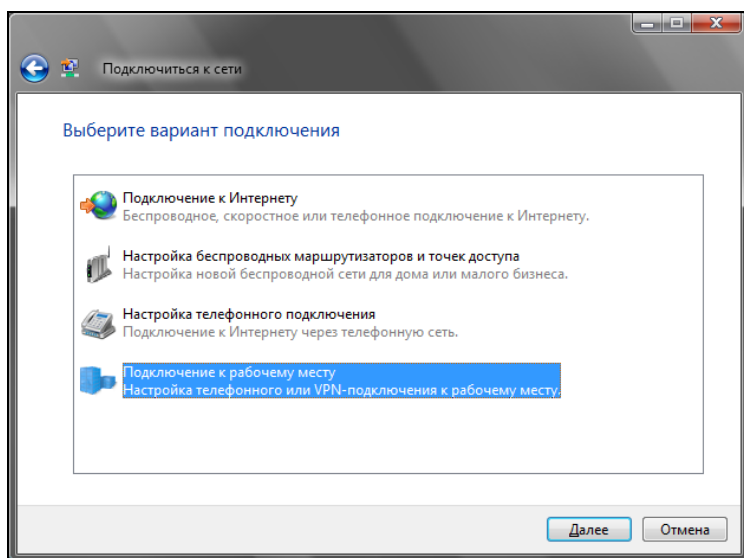


Рис. 39.10. Выбор варианта подключения

Далее процесс настройки ничем не отличается от процесса настройки в Windows XP: нужно выбрать интернет-соединение (рис. 39.11), ввести имя VPN-сервера и т. д.

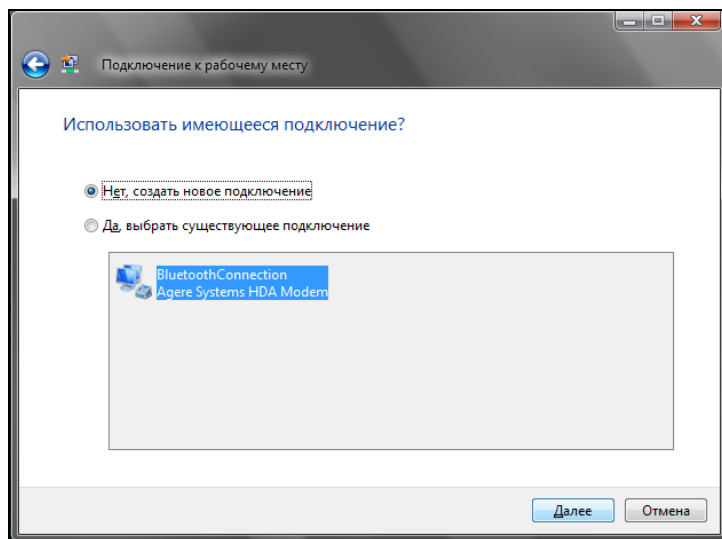


Рис. 39.11. Выбор интернет-соединения

ГЛАВА 40



Сервис Samba

40.1. Установка Samba

Linux — отличная операционная система, но от Windows нам не уйти. Windows будет окружать нас всегда, будь то домашняя, корпоративная сеть или интернет-кафе. Нам постоянно предстоит обмениваться документами с Windows-компьютерами — ведь далеко не все пользователи предпочитают работать в Linux. В этой книге особое внимание было уделено взаимодействию с Windows-компьютерами, и было бы нелогично не сказать о подключении Linux к сети Microsoft.

В Linux для взаимодействия с сетью Microsoft служит пакет `samba-server`. Если вы хотите использовать общие ресурсы Windows-сети, установите этот пакет. Он позволяет не только пользоваться общими ресурсами сети, но и предоставлять собственные ресурсы Windows-пользователям. Причем все происходит так, что Windows-пользователи даже не заметят разницы.

После установки этого пакета будет установлен сервис `smb` — это и есть основной сервис Samba. Запускать и останавливать его можно командами:

```
service smb start
service smb stop
```

40.2. Базовая настройка Samba

Основной конфигурационный файл Samba — `/etc/samba/smb.conf`. Откройте его. Сейчас мы изменим пару параметров. Первым делом измените параметр `WORKGROUP` — он задает имя рабочей группы или домена NT:

```
WORKGROUP = MSHOME
```

Конечно, имя группы у вас, скорее всего, будет другим. Можете также установить параметр `server string` — это описание вашего компьютера:

```
server string = My Linux computer
```

Установите параметр `security`. Если у вас клиент-серверная сеть, то нужно выбрать параметр `server`, а если одноранговая сеть (т. е. сеть без выделенного сервера), то нужно выбрать `user` или `share`:

```
security = share
```

Имя гостевой учетной записи установите так:

```
guest account = guest
```

Также нужно настроить кодировки:

```
unix charset = UTF-8
```

```
dos charset = UTF-8
```

```
display charset = UTF-8
```

Для того чтобы Samba работала быстрее, установите следующие опции:

```
socket options = TCP_NODELAY SO_RCVBUF=8192 SO_SNDBUF=8192
```

```
dns proxy = no
```

Что они означают, мы разберемся чуть позже.

Параметр `interfaces` указывает интерфейсы, на которых должен работать сервис `smb`. Укажите те интерфейсы, которые связывают вашу машину с Windows-сетями:

```
interfaces = 192.168.0.22/24
```

А теперь позволю себе несколько комментариев для пользователей предыдущей версии Samba. В более ранних версиях Samba параметр `server string` назывался `comment`. Теперь вместо параметров `client code page` и `character set` используются параметры `unix charset`, `dos charset` и `display charset`. Текущая версия Samba полностью поддерживает UTF-8 (как и современные версии Linux и Windows), поэтому проблем с UTF-8 возникнуть не должно. Параметры `client code page` и `character set` больше не поддерживаются.

Параметр `unix charset` задает кодировку, в которой хранятся файлы конфигурации Samba, `dos charset` — кодировку для Windows-клиентов, а `display charset` — кодировку для Samba-клиентов.

40.3. Настройка общих ресурсов

Теперь осталось сконфигурировать ресурсы, которые вы хотите предоставить в общее пользование (листинг 40.1). Фрагмент, приведенный в листинге 40.1, нужно добавить в файл конфигурации Samba.

Листинг 40.1. Секция [public]

```
[public]
```

```
# общий каталог, комментарий для ресурса задается директивой comment
```

```
comment = Public Directory
```

```
# путь
path = /var/samba
# не только чтение
read only = no
# разрешить запись
writable = yes
# разрешить гостевой доступ
guest ok = yes
# разрешить просмотр содержимого каталога
browseable = yes
```

В этом случае общим ресурсом нашего компьютера будет каталог `/var/samba`. В него другие пользователи смогут записывать свои файлы (`read only = no`, `writable = yes`), естественно, они смогут их и читать (`browseable = yes`). Проверка имени пользователя и пароля для доступа к ресурсу не нужна (`guest ok = yes`) — используется так называемый *гостевой* доступ. Комментарий `Public Directory` увидят другие пользователи Windows-сети при просмотре ресурсов вашего компьютера.

Рассмотрим еще один пример, позволяющий сделать общими домашние каталоги пользователей — секция `[homes]` (листинг 40.2).

Листинг 40.2. Секция `[homes]`

```
[homes]
comment = Home Directories
browseable = no
valid users = %S

# запись запрещена, только просмотр
writable = no

# маска при создании файлов, нужна если writable=yes
create mask = 0600
# маска при создании каталогов, нужна если writable=yes
directory mask = 0700
```

В листинге 40.3 приведен пример предоставления общего доступа к CD/DVD. Будем считать, что наш CD/DVD смонтирован в `/cdrom`.

Листинг 40.3. Пример общего доступа к CD/DVD

```
[cdrom]
comment = Samba server's CD-ROM
writable = no
locking = no
```

```
# каталог /cdrom должен существовать и являться точкой монтирования CD/DVD
path = /cdrom
public = yes

# следующие два параметра нужны для автоматического монтирования CD/DVD
# они будут работать, если /etc/fstab содержит следующую строку:
# /dev/scd0 /cdrom iso9660 defaults,noauto,ro,user 0 0
# /dev/scd0 – имя устройства CD/DVD
# /cdrom – точка монтирования (каталог должен существовать)
preexec = /bin/mount /cdrom
postexec = /bin/umount /cdrom
```

40.4. Просмотр ресурсов Windows-сети

Просмотреть ресурсы Windows-сети можно с помощью программы smbclient, но она работает в текстовом режиме, поэтому не совсем удобна. В современных дистрибутивах ресурсы Windows-сети можно просмотреть средствами графической среды. В KDE откройте файловый менеджер Dolphin, а в боковой панели выберите **Сеть**, после — **Samba Shares** (рис. 40.1).

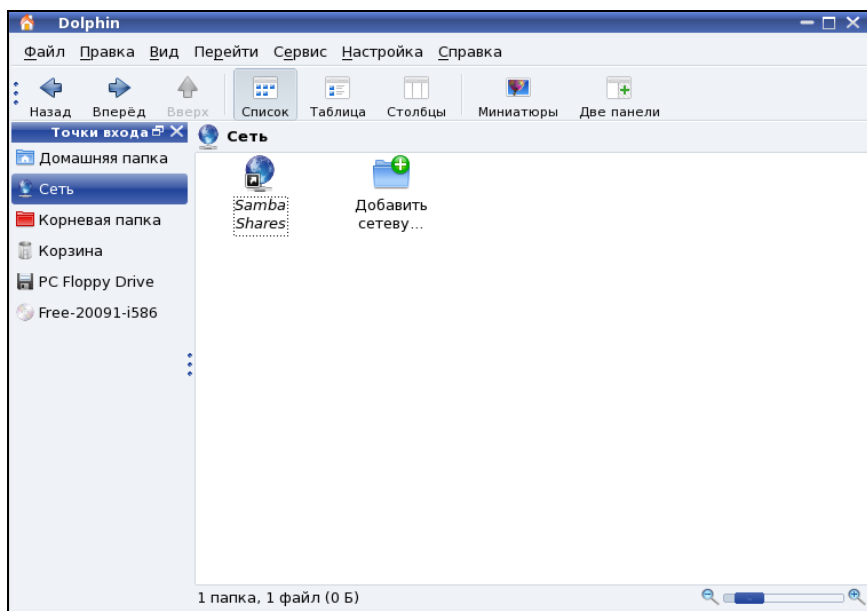


Рис. 40.1. Просмотр ресурсов сети с помощью Dolphin

Если вы используете GNOME, то для просмотра ресурсов сети можно использовать команду главного меню **Переход | Сеть**, что даже проще и удобнее, чем описанная здесь программа.

40.5. Оптимизация Samba

Если открыть файл конфигурации `smb.conf`, вы найдете в нем параметр `wide links`. Никогда не устанавливайте его в `no`! Так вы существенно снизите производительность Samba. Наоборот, если вы установите его в `yes` (если до этого параметр `wide links` был отключен), то вы можете существенно повысить производительность.

Параметр `wide links` определяет, как Samba будет следовать по символическим ссылкам. Если `wide links = no`, то Samba не будет следовать по символическим ссылкам вне экспортируемой области. Сначала Samba следует по символической ссылке, а затем выполняет так называемый `directory path lookup` (системный вызов, определяющий, где завершилась ссылка). Данная операция подразумевает на 6 системных вызовов больше, нежели в случае, если `wide links = yes`. Учитывая, что подобных операций делается очень много, то выключение `wide links` снижает производительность Samba приблизительно на 30%.

Протокол TCP/IP — штука тонкая. Производительность сетевых приложений во многом зависит от того, правильно ли настроен TCP/IP. Samba — настоящее сетевое приложение, которое к тому же работает по протоколу TCP/IP. При использовании TCP/IP, если размер запросов и ответов не фиксирован (как в случае с Samba), рекомендуется применять протокол TCP с опцией `TCP_NODELAY`. Для этого в файл `smb.conf` нужно добавить строку:

```
socket options = TCP_NODELAY
```

Тесты показывают, что с указанными опциями Samba при больших нагрузках работает в три раза быстрее, чем без указания этих опций. Если Samba используется в локальной сети (в большинстве случаев так оно и есть) рекомендуется еще указать и такую опцию `IP_TOS_LOWDELAY`:

```
socket options = IPTOS_LOWDELAY TCP_NODELAY
```

Если есть желание "выжать" из Samba еще больше, тогда установите следующие параметры буферизации: `SO_RCVBUF=8192 SO_SNDBUF=8192`. Например:

```
socket options = TCP_NODELAY SO_RCVBUF=8192 SO_SNDBUF=8192
```

ГЛАВА 41



Удаленный доступ

41.1. Зачем нужен удаленный доступ

Для сервера очень важна возможность удаленного доступа. Ведь не всегда есть возможность получить физический доступ к серверу: вы можете находиться в другом конце города или даже в другой стране, а сервер предприятия будет требовать вашего "оперативного" вмешательства.

Для организации удаленного доступа мы будем использовать два совершенно разных способа: протокол SSH и X-терминалы. В первом случае мы получим доступ к консоли сервера. Именно это нам и нужно, если мы подключаемся по медленному каналу (модем, мобильный телефон) — ведь для передачи текста большая скорость не нужна.

Второй способ подходит для более скоростного канала — выделенной линии или же локальной сети. Но зато X-терминалы позволяют работать с удаленным компьютером, как с локальным, т. е. с полным эффектом присутствия. В отдельном окне вы будете видеть графический интерфейс удаленного компьютера. А если активизировать полноэкранный режим, тогда вообще нельзя будет даже и предположить, что работаешь за удаленным компьютером — разницы никакой не будет. Все будет работать немного медленнее, ведь данные нужно передать по сети, а не по внутренней шине компьютера. Но тут все зависит от конфигурации самих компьютеров и, конечно же, от самой сети.

41.2. Протокол SSH

Раньше для организации удаленного доступа к консоли сервера использовался протокол Telnet. В каждой сетевой операционной системе, будь то FreeBSD или Windows 7 (которую, впрочем, сложно назвать сетевой), есть telnet-клиент. Данная программа так и называется — telnet (в Windows — telnet.exe) (рис. 41.1).

ПРИМЕЧАНИЕ

Начиная с Windows Vista, программа telnet больше не устанавливается по умолчанию. Инструкции по установке этой программы можно прочитать по адресу: <http://system-administrators.info/?p=3329>.

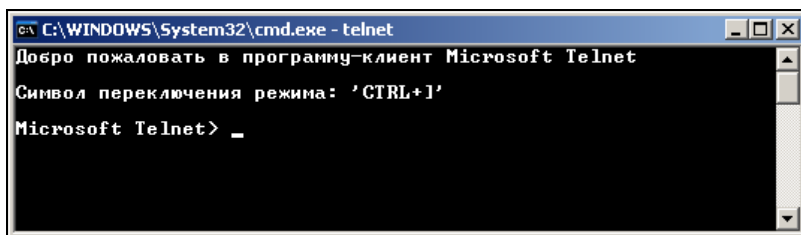


Рис. 41.1. Telnet-клиент в Windows XP Pro

После подключения с помощью telnet к удаленному компьютеру вы можете работать с ним как обычно. В окне telnet-клиента вы увидите как бы консоль удаленного компьютера: вы будете вводить команды и получать результат их выполнения — все так, как если бы вы работали непосредственно за удаленным компьютером.

Но технологии не стоят на месте, и протокол Telnet устарел. Сейчас им практически никто не пользуется. На его смену пришел SSH (Secure Shell). SSH, как видно из названия, представляет собой безопасную оболочку. Главное отличие от telnet состоит в том, что все данные (включая пароли доступа к удаленному компьютеру, передаваемые по SSH файлы) передаются в зашифрованном виде. Во времена telnet участились случаи перехвата паролей и другой важной информации, что и стало причиной создания SSH.

SSH использует следующие алгоритмы для шифрования передаваемых данных: BlowFish, 3DES (Data Encryption Standard), IDEA (International Data Encryption Algorithm) и RSA (Rivest-Shamir-Adelman algorithm). Самыми надежными являются алгоритмы IDEA и RSA. Поэтому, если вы передаете действительно конфиденциальные данные, лучше использовать один из этих алгоритмов.

В состав любого дистрибутива Linux входит SSH-сервер (программа, обеспечивающая удаленный доступ к компьютеру, на котором она установлена) и SSH-клиент (программа, позволяющая подключаться к SSH-серверу). Для установки SSH-сервера нужно установить пакет openssh (это разновидность SSH-сервера), а для установки SSH-клиента — пакет openssh-clients.

Если у вас на рабочей станции установлена система Windows, и вам нужно подключиться к SSH-серверу, запущенному на Linux-машине, то по адресу <http://www.cs.hut.fi/ssh/> вы можете скачать Windows-клиент для SSH. Нужно отметить, что Windows-клиент, в отличие от Linux-клиента, не бесплатен.

Работать с SSH-клиентом очень просто. Для подключения к удаленному компьютеру введите команду:

```
ssh [опции] <адрес_удаленного_компьютера>
```

В качестве адреса можно указать как IP-адрес, так и доменное имя компьютера. В табл. 41.1 приведены часто используемые опции программы ssh.

Таблица 41.1. Опции программы ssh

Опция	Описание
-c blowfish 3des des	Служит для выбора алгоритма шифрования, при условии, что используется первая версия протокола SSH (об этом позже). Можно указать blowfish, des или 3des
-c шифр	Задаёт список шифров, разделённых запятыми в порядке предпочтения. Опция используется для второй версии SSH. Можно указать blowfish, twofish, arcfour, cast, des и 3des
-f	Переводит ssh в фоновый режим после аутентификации пользователя. Рекомендуется использовать для запуска программы X11. Например: ssh -f server xterm
-l имя_пользователя	Указывает имя пользователя, с правами которого нужно зарегистрироваться на удалённом компьютере. Опцию использовать не обязательно, поскольку удалённый компьютер и так запросит имя пользователя и пароль
-p порт	Определяет порт SSH-сервера (по умолчанию используется порт 22)
-q	"Тихий режим". Будут отображаться только сообщения о фатальных ошибках. Все прочие предупреждающие сообщения в стандартный выходной поток выводиться не будут
-x	Отключает перенаправление X11
-X	Задействовать перенаправление X11. Полезна при запуске X11-программ
-1	Использовать только первую версию протокола SSH
-2	Использовать только вторую версию протокола SSH. Вторая версия протокола более безопасна, поэтому при настройке SSH-сервера нужно использовать именно её

Теперь можно приступить к конфигурированию SSH-сервера. Если вы используете OpenSSH (в большинстве случаев так оно и есть), все настройки SSH-сервера хранятся в одном-единственном файле — /etc/sshd_config, а настройки программы-клиента — в файле /etc/ssh_config. Настройки программы-клиента обычно задавать не нужно, поскольку они приемлемы по умолчанию. На всякий случай вы можете заглянуть в файл /etc/ssh_config — его формат, как и назначение опций (большая часть из них закомментирована), вы поймете без моих описаний.

В данный момент нас больше интересует файл sshd_config, содержащий конфигурацию SSH-сервера. Рассмотрим пример файла конфигурации SSH-сервера

(листинг 41.1). Чтобы понять назначение директив, внимательно читайте комментарии, приведенные в листинге.

Листинг 41.1. Пример файла конфигурации /etc/ssh/sshd_config

```
# $OpenBSD: sshd_config,v 1.72 2005/07/25 11:59:40 markus Exp $

# Задаёт порт, на котором будет работать SSH-сервер. Если директива
# не указана (закомментирована), то по умолчанию используется порт 22
#Port 22

# Директива Protocol позволяет выбрать версию протокола,
# рекомендуется использовать вторую версию
#Protocol 2,1
Protocol 2

# Директива AddressFamily задаёт семейство интерфейсов, которые должен
# прослушивать SSH-сервер
#AddressFamily any

# Локальный адрес, который должен прослушиваться SSH-сервером
#ListenAddress 0.0.0.0

# Ключевой файл для протокола SSH версии 1
# HostKey for protocol version 1
HostKey /etc/ssh/ssh_host_key
# Ключевые файлы для второй версии протокола SSH
HostKey /etc/ssh/ssh_host_rsa_key
HostKey /etc/ssh/ssh_host_dsa_key

# Время жизни ключа протокола первой версии. Время можно задавать в секундах
# или в часах (постфикс h, например, 1h – это 1 час или 3600 секунд).
# По истечении указанного времени ключевой файл будет сгенерирован заново
#KeyRegenerationInterval 1h

# Разрядность ключа сервера в битах (только для первой версии протокола SSH)
#ServerKeyBits 768

# Директивы управления протоколированием (можно не изменять)
#SyslogFacility AUTH
#LogLevel INFO

# Директивы аутентификации
```

```
# Время, предоставляемое клиенту для аутентификации. Задается в секундах
# или минутах (1m = 60 секунд). Если за это время клиент не
# аутентифицировал себя, соединение будет прекращено
#LoginGraceTime 2m

# Директива разрешает (yes) удаленный доступ пользователя root
PermitRootLogin yes

# Максимальное количество попыток аутентификации
#MaxAuthTries 6

# Использование RSA (yes)
#RSAAuthentication yes
# Аутентификация с открытым ключом (при значении yes)
#PubkeyAuthentication yes
#AuthorizedKeysFile .ssh/authorized_keys

# Использование rhosts-аутентификации с поддержкой RSA.
# Rhosts-аутентификацию использовать не рекомендуется, поэтому по умолчанию
# для этой директивы указано значение no. Если вы все-таки установите
# значение yes для этой директивы, то не забудьте указать в файле
# /etc/ssh/ssh_known_hosts IP-адреса компьютеров, которым разрешен доступ
# к SSH-серверу. Только для первой версии протокола
#RhostsRSAAuthentication no

# Если вы используете вторую версию протокола и хотите разрешить
# rhosts-аутентификацию, то вам нужно включить директиву
# HostbasedAuthentication,
# а разрешенные узлы указываются в файле ~/.ssh/known_hosts
# HostbasedAuthentication no

# Если вы не доверяете пользовательским файлам ~/.ssh/known_hosts,
# установите значение yes для директивы IgnoreUserKnownHosts. Тогда будет
# использован только файл /etc/ssh/ssh_known_hosts
#IgnoreUserKnownHosts no

# Игнорировать файлы ~/.rhosts и ~/.shosts (рекомендуется установить yes)
#IgnoreRhosts yes
# Следующие директивы не рекомендуется изменять из соображений безопасности —
# они включают аутентификацию по паролю (а не IP-адресу компьютера, указанному
# в файле /etc/ssh/ssh_known_hosts) и запрещают использование пустых паролей
#PasswordAuthentication yes
#PermitEmptyPasswords no
```

```
# Параметры протокола аутентификации Kerberos
# Рекомендуется использовать RSA-аутентификацию
#KerberosAuthentication no
#KerberosOrLocalPasswd yes
#KerberosTicketCleanup yes
#KerberosGetAFSToken no

# Параметры GSSAPI
#GSSAPIAuthentication no
#GSSAPICleanupCredentials yes

# Использовать для аутентификации модули PAM (по умолчанию они
# не используются)
#UsePAM no

# Разрешить TCP-форвардинг
#AllowTcpForwarding yes

# Использовать порты шлюза
#GatewayPorts no

# Использовать X11-форвардинг (для запуска X11-приложений)
X11Forwarding yes

# Выводить сообщение дня (содержится в файле /etc/motd)
#PrintMotd yes

# Выводить время последней регистрации пользователя
#PrintLastLog yes

# Не обрывать TCP-соединения после выполнения команды по SSH
#TCPKeepAlive yes
# Отключение (значение no) этой опции позволяет немного ускорить работу
# SSH, поскольку DNS не будет использоваться для разрешения доменных имен
#UseDNS yes

# Остальные параметры рекомендуется оставить как есть
#UseLogin no
UsePrivilegeSeparation yes
#PermitUserEnvironment no
#Compression delayed
#ClientAliveInterval 0
#ClientAliveCountMax 3
```

```
#PidFile /var/run/sshd.pid
#MaxStartups 10

#Banner /some/path

Subsystem sftp /usr/lib/ssh/sftp-server
```

После установки пакетов `openssh` и `openssh-clients` можно приступить к тестированию работы SSH-сервера. Для запуска сервера в Mandriva или Fedora Core можно использовать команду:

```
# service sshd start
```

А для останова (в Mandriva или Fedora) — ту же команду, но с параметром `stop`:

```
# service sshd stop
```

В Debian/Ubuntu для запуска/останова сервера используются команды (соответственно):

```
sudo /etc/init.d/ssh start
sudo /etc/init.d/ssh stop
```

Также запустите конфигуратор управления сервисами (`drakxservices` в Mandriva и `system-config-services` в Fedora) и убедитесь, что сервис `sshd` запускается при запуске системы. В старых версиях Ubuntu конфигуратор, управляющий службами (сервисами), можно запустить с помощью команды меню **Система | Администрирование | Службы**. В новых версиях Ubuntu этот конфигуратор отсутствует, а вместо него предлагается установить программу `bum` (Boot Up Manager):

```
sudo apt-get install bum
```

После этого можно ввести команду:

```
ssh 127.0.0.1
```

для подключения к локальному компьютеру. Можно также подключиться с удаленного компьютера. Если сеть на локальном и удаленном компьютере настроена правильно, проблем не должно возникнуть.

41.3. X-терминалы

В последние годы все чаще говорят о "тонких" клиентах под Windows. Суть "тонкого" клиента заключается в том, что рабочая станция подключается к серверу терминалов. После этого происходит процесс регистрации пользователя в системе. А затем пользователь может работать с графическим интерфейсом сервера так, как если бы он непосредственно находился за клавиатурой и монитором сервера терминала. Прелесть такого решения заключается вот в чем. В качестве рабочей станции могут выступать компьютеры самой минимальной конфигурации. Главное, чтобы на таком компьютере можно было запустить операционную систему, способную подключиться к серверу терминалов. Не нужны ни большие объемы оперативной памяти, ни дисковой памяти. Необходима только сетевая карта: по ней действия пользова-

теля будут передаваться на сервер терминалов, по ней будет также передаваться "картинка" с сервера — результат выполнения этих команд. Все программы, запускаемые пользователем, будут выполняться на сервере терминалов, а компьютер пользователя будет только отображать результат их выполнения, ну и, разумеется, будет передавать нажатия клавиш и перемещения мыши серверу терминалов.

Удобно? С одной стороны — да, с другой — нет. В первую очередь в голову приходит мысль о том, что можно сэкономить на рабочих станциях. Но сервер терминалов должен быть очень мощным компьютером. Очень. Тут все зависит от поставленной задачи. Иногда бывает дешевле или проще купить несколько самых дешевых рабочих станций, чем покупать мощный сервер. Тем более, все равно скорость выполнения задач будет не такой высокой, как ожидается. Во-первых, данные передаются по сети, на что требуется дополнительное время. Во-вторых, к серверу терминалов одновременно подключается множество рабочих станций, иначе зачем он нам нужен?

ОТСТУПЛЕНИЕ

Итак, зачем же он нам нужен? Предположим, вы хотите сэкономить. Идея заключается в следующем. Вы хотите построить сеть предприятия, которая будет использовать сервер терминалов. Вы покупаете мощный сервер терминалов (его стоимость будет исчисляться тысячами долларов), а также определенное количество самых дешевых новых рабочих станций. Но простой расчет показывает, что с экономической точки зрения затея не окупится. Дешевле купить более мощные рабочие станции и не покупать сервер терминалов. Тем более, что вы можете немного сэкономить, если правильно подойдете к процессу покупки рабочих станций. Ведь не всем пользователям нужны мощные компьютеры. Например, секретарю и бухгалтеру вычислительные мощности не нужны. В первом случае компьютер будет использоваться как электронная печатающая машинка, а во втором все запросы к базе данных будет обрабатывать сервер баз данных. Сервер баз данных, на котором будет установлена база данных "1С" (чего греха таить, большая часть предприятий использует именно эту программу, поэтому не нужно думать, что это скрытая реклама), должен быть, конечно, мощнее, но, учитывая тенденции цен на современные компьютеры, разница между компьютером бухгалтера и сервером баз данных вряд ли превысит 300—400 долларов (конечно, если у вас несколько бухгалтерских машин, а не целая армия).

Выходит, сэкономить не получилось. Купить нужное количество бывших в употреблении компьютеров не всегда возможно, да и вся ваша сеть в глазах ваших клиентов будет выглядеть не совсем солидно. Много написано, а на вопрос так и не был получен ответ. Зачем современному предприятию сервер терминалов? Если ваше предприятие работает, скажем, 5—10 лет, то наверняка на складе найдутся списанные компьютеры, которые и продать нельзя, и выбросить жалко. Точнее, продать-то можно, но за гроши. Выбрасывать тоже жалко — ведь компьютеры нормально работают, просто скорость выполнения задач уже не соответствует современным меркам. Так вот, если у вас есть несколько таких компьютеров, в них можно вдохнуть вторую жизнь благодаря серверу терминалов. Причем в качестве сервера терминалов уже не нужно покупать дорогостоящий сервер стоимостью несколько тысяч долларов. Вполне подойдет обычный компьютер с производительностью современной рабочей станции. Ведь ему нужно будет обслуживать не все компьютеры сети, а только несколько компьютеров-ветеранов. Единственное требование к серверу — это объем оперативной памяти, он должен быть 1 Гбайт или даже больше. Ясно, что скорость выполнения программ при одновременной работе, скажем, 5 пользователей будет оставлять желать лучшего. Но сейчас речь идет не о скорости, а о возможности запуска современных приложений, которые на тех машинах просто не запустишь. Пользователи будут работать с последними версиями графических редакторов и текстовых процессоров. Да, они будут работать относительно мед-

ленно, но все же будут работать. Единственное капиталовложение — это дополнительные модули памяти для компьютера, который выделен под нужды сервера терминалов. В случае с Windows вам также понадобится купить дополнительное программное обеспечение. А в случае с Linux — только один дистрибутив Linux. И ваш старенький Pentium 100 заживет совершенно новой жизнью.

Теперь еще раз о Windows и Linux. Если о "тонких" клиентах под Windows начали говорить относительно недавно (примерно с 2001 г.), то в UNIX/Linux возможность организации "тонкого" клиента была с момента появления графической системы X Window, а это начало 90-х годов прошлого века. Раньше настройка X-сервера и X-терминала — компонентов тонкого клиента — занимала достаточно много времени, особенно если настройка производилась впервые. Сейчас же для настройки сервера и клиента достаточно выполнить ряд простых действий. Итак, приступим к настройке. При настройке мы будем подразумевать, что на сервере и на всех клиентах установлен дистрибутив Ubuntu, а графическая система X.Org корректно работает.

Первым делом нужно разрешить доступ к нашему рабочему столу удаленным клиентам. Для этого выберите **Система | Параметры | Удаленный рабочий стол**. В появившемся окне (рис. 41.2) вы можете разрешить удаленным пользователям видеть ваш рабочий стол и даже управлять им (если есть такая необходимость). Из соображений безопасности рекомендуется установить пароль для доступа к рабочему столу.

После установки параметров можно попробовать подключиться с другого компьютера. Для этого запустите программу **Приложения | Интернет | Просмотр удаленных рабочих столов** и нажмите кнопку **Подключиться**. Далее введите IP-адрес узла и нажмите кнопку **Подключиться** (рис. 41.3). Кстати, из списка **Протокол** можно выбрать SSH, тогда приложение Просмотр удаленных рабочих столов превратится в графический SSH-клиент.

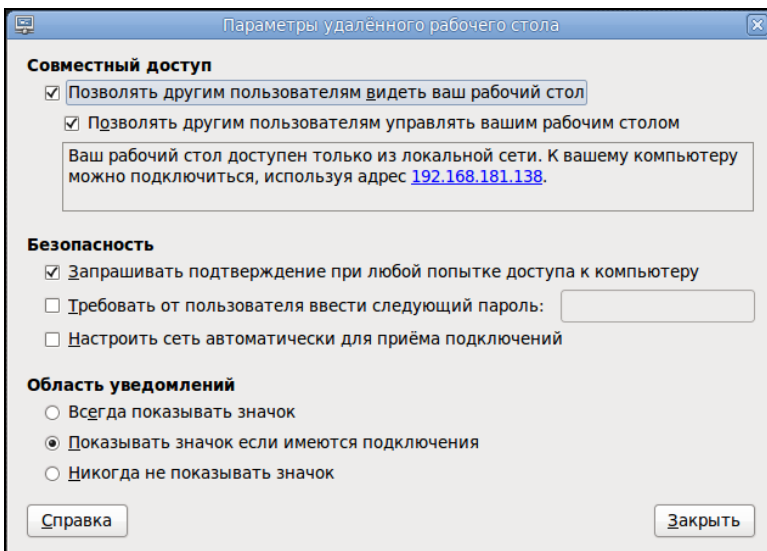


Рис. 41.2. Параметры удаленного рабочего стола

Если вам нужно подключиться к Windows-серверу терминалов, тогда выберите приложение **Приложения | Интернет | Клиент терминального сервера** (рис. 41.4).

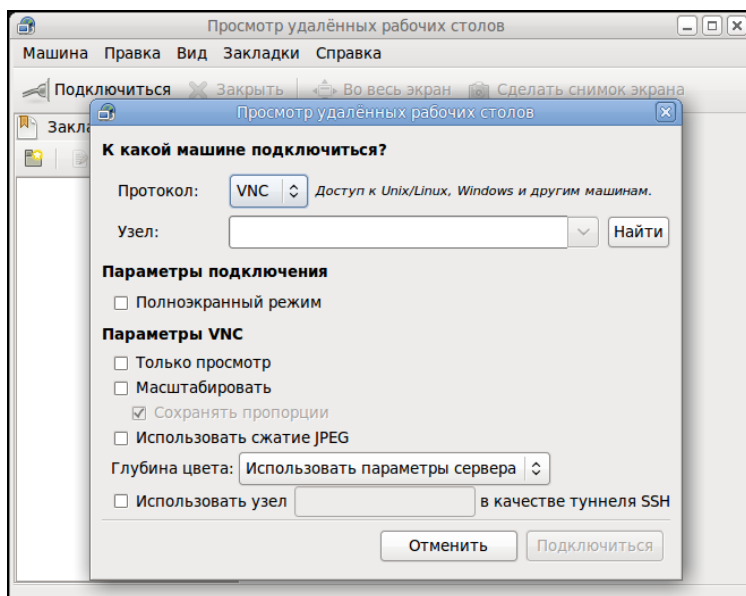


Рис. 41.3. Просмотр удаленных рабочих столов

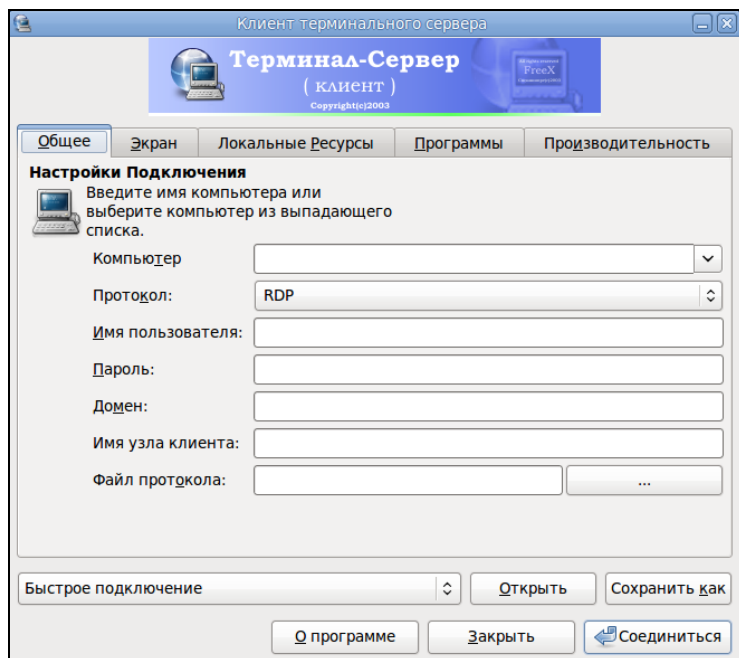


Рис. 41.4. Клиент терминального сервера

ГЛАВА 42



Оптимизация сервера и рабочей станции

42.1. Общая оптимизация Linux

В этом разделе мы поговорим об общей оптимизации Linux как сервера, так и рабочей станции, а в следующем — рассмотрим оптимизацию отдельных сетевых сервисов.

42.1.1. Оптимизация подкачки

Операционная система Linux не очень требовательная к памяти: для нормальной работы даже шлюза небольшой сети вполне хватит 128 Мбайт оперативной памяти. Не верите? Посмотрите на рис. 42.1: 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:        12588        113776
Swap:         240964           0        240964
[root@localhost ~]# _
```

Рис. 42.1. Команда `free` — сведения об использовании оперативной памяти

Но это только в том случае, если не запущена система X Org. После ее запуска Linux превращается в настоящего "обжору", съедающего десятки мегабайт памяти. Сама система X Org 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 384 Мбайт оперативной памяти.

ПРИМЕЧАНИЕ

Да, реалии таковы, что Linux со временем становится все более требовательной к оперативной памяти. Так, для Fedora 13 в текстовом режиме необходимо 256 Мбайт (минимум) оперативной памяти. Не верите? Прочитайте официальные системные требования: http://docs.fedoraproject.org/ru-RU/Fedora/13/html/Release_Notes/index.html#sect-Release_Notes-Hardware_Requirements. А для запуска графического интерфейса требуется 384 Мбайт ОЗУ, хотя я бы установил необходимый минимум в 512 Мбайт — так система более или менее нормально работает. А если у вас старый компьютер с ОЗУ 128 Мбайт, то нужно найти более старый дистрибутив, например Fedora 10, для запуска графического режима требует 256 Мбайт. Вот только найти старый дистрибутив становится все сложнее и сложнее.

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, Amarok, GIMP. Не все сразу, а только те, которые вы часто используете одновременно. Затем введите команду `free` и посмотрите, сколько мегабайт оперативной памяти у вас свободно. Также обратите внимание на "остаток" области подкачки (swap). Если и там, и там осталось всего несколько мегабайт памяти, значит, 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.

В главе 8 мы научились создавать файл подкачки. Но 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 программками и часто переключаетесь между ними, можно установить значение меньше 50, например 40, или даже 30. В этом случае переключение между приложениями будет мгновенным, однако 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 OpenOffice, 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 медленным, 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
```

42.1.2. Изменение планировщика ввода/вывода

Производительность многозадачной системы в целом сильно зависит от правильного планирования процессов системы. 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;
- ❖ `cfg` — "честная очередь";
- ❖ `deadline` — планирование крайних сроков.

Для домашнего компьютера больше подойдут значения `as` и `cfg`. 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 `cfg`) 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 42.1).

Листинг 42.1. Фрагмент файла `/etc/grub/grub.conf`

```
...
title Linux
root (hd1,0)
kernel /boot/vmlinuz-2.6.9 ro root=/dev/sda1 elevator=as
...
```

Если у вас LILO, отредактируйте ваш файл `/etc/lilo.conf` так (листинг 42.2).

Листинг 42.2. Фрагмент файла `/etc/lilo.conf`

```
image=/boot/vmlinuz-2.6.9
label=Linux
root=/dev/sda1
append="elevator=as"
...
```

После изменений файла `/etc/lilo.conf` не забудьте выполнить команду `lilo` для того, чтобы изменения вступили в силу.

Счастливым (или не очень) обладателям GRUB2 нужно редактировать файл `/etc/default/grub`. В нем нужно отредактировать строку параметров по умолчанию:

```
GRUB_CMDLINE_LINUX_DEFAULT="quiet splash elevator=as"
```

Вообще-то правильнее редактировать файл `/boot/grub/grub.cfg`, чтобы изменить параметры только одной загрузочной записи, но параметр выбора загрузчика не является критическим. Если вы измените его для всех загрузочных записей (в файле `/etc/default/grub`), система все равно корректно загрузится. Самое неприятное — вам может не понравиться, как система работает с тем или иным значением. Но это решается просто — выбором другого значения. После редактирования `/etc/default/grub` выполните следующую команду:

```
sudo update-grub
```

42.2. Оптимизация сетевых сервисов

После общей оптимизации сервера нужно заняться оптимизацией отдельных сетевых сервисов. Особую нагрузку на сервер производит сервис Samba, поэтому с него и начнем.

42.2.1. Секреты оптимизации Samba

Если открыть файл конфигурации `smb.conf`, вы найдете в нем параметр `wide links`. Никогда не устанавливайте его в `no`! Так вы существенно снизите производительность Samba. Наоборот, если вы установите его в `yes` (если до этого параметр `wide links` был отключен), то вы можете существенно повысить производительность.

Параметр `wide links` определяет, как Samba будет следовать по символическим ссылкам. Если `wide links = no`, то Samba не будет следовать по символическим ссылкам вне экспортируемой области. Сначала Samba следует по символической ссылке, а затем выполняет так называемый `directory path lookup` (системный вызов, определяющий, где завершилась ссылка). Данная операция подразумевает на 6 системных вызовов больше, нежели в случае, если `wide links = yes`. Учитывая, что подобных операций делается очень много, то выключение `wide links` снижает производительность Samba приблизительно на 30%.

Протокол TCP/IP — штука тонкая. Производительность сетевых приложений во многом зависит от того, правильно ли настроен TCP/IP. Samba — настоящее сетевое приложение, которое к тому же работает по протоколу TCP/IP. При использовании TCP/IP, если размер запросов и ответов не фиксирован (как в случае с Samba), рекомендуется применять протокол TCP с опцией `TCP_NODELAY`. Для этого в файл `smb.conf` нужно добавить строку:

```
socket options = TCP_NODELAY
```

Тесты показывают, что Samba при больших нагрузках работает в 3 раза быстрее, чем без указания этих опций. Если Samba используется в локальной сети (в большинстве случаев так оно и есть), рекомендуется еще указать опцию `IPTOS_LOWDELAY`:

```
socket options = IPTOS_LOWDELAY TCP_NODELAY
```

Если есть желание "выжать" из Samba еще больше, тогда установите следующие параметры буферизации: `SO_RCVBUF=8192 SO_SNDBUF=8192`. Например:

```
socket options = TCP_NODELAY SO_RCVBUF=8192 SO_SNDBUF=8192
```

42.2.2. Оптимизация ProFTPД

Оптимизировать ProFTPД можно по трем направлениям: ускорить авторизацию, равномерно распределить нагрузку на сервер и помочь серверу избежать перегрузки "узкого" канала.

Начнем с авторизации. Ускорить авторизацию помогут директивы `IdentLookup` и `UseReverseDNS`. Первая управляет использованием протокола `ident`, но поскольку этот протокол давно не используется, данную директиву можно отключить. Вторая определяет доменное имя клиента по его IP-адресу, но это занимает некоторое время, поэтому для ускорения доступа к FTP-серверу ее нужно отключить. Добавьте в файл конфигурации `proftpd.conf` следующие строки:

```
IdentLookups off
UseReverseDNS off
```

Еще к авторизации относится директива `MaxLoginAttempts`, задающая максимальное число попыток регистрации пользователя на сервере:

```
MaxLoginAttempts 3
```

Теперь приступим к распределению нагрузки на сервер. Первым делом нужно задать максимальное число клиентов:

```
MaxClients число
```

Понятно, что чем быстрее наш канал подключения к Интернету, тем больше клиентов наш сервер сможет принять.

С помощью директивы `MaxClientsPerHost` можно установить максимальное число клиентов с одного узла:

```
MaxClientsPerHost число
```

Не нужно устанавливать значение 1 для этой директивы. Представьте, что есть сеть, доступ к Интернету которой осуществляется через один сервер — шлюз. То есть у всей сети только один реальный IP. Получается, что у всех пользователей этой сети один IP. Если установить этот параметр в 1, то из всей сети на наш FTP сможет зайти только один пользователь. Понятно, что все пользователи сети тоже не будут одновременно заходить на наш FTP, поэтому для директивы `MaxClientsPerHost` нужно установить небольшое значение, например 2 или 3.

Предположим, что доступ к нашему FTP разрешен только зарегистрированным (а не анонимным) пользователям. Но некоторые пользователи могут "одолжить" свой логин и пароль другим, незарегистрированным на сервере пользователям, чтобы они тоже смогли использовать ресурсы нашего сервера. Это нехорошо, поэтому с помощью директивы `MaxClientsPerUser` мы можем контролировать максимальное число FTP-клиентов от одного пользователя. Вот тут самое время установить значение 1:

```
MaxClientsPerUser 1
```

Но пользователи хотят нас обхитрить. Они заходят под одним и тем же логином, но с разных узлов (например, с разных сетей). Нужно запретить им делать это:

```
MaxHostsPerUser 1
```

Директива `MaxHostsPerUser`, как понятно из ее названия, ограничивает количество узлов на одного пользователя.

Еще нужно установить директиву `MaxInstances`, задающую максимальное число параллельно запущенных экземпляров сервера `proftpd` (для каждого нового клиента запускается своя копия `proftpd` для обработки его запросов). Ее значение зависит от возможностей вашего сервера. Предположим, что для директивы `MaxClients` мы задали значение 10, т. е. одновременно могут работать 10 пользователей. Поскольку мы установили для `MaxClientsPerUser` и `MaxHostsPerUser` значение 1, то для `MaxInstances` можно установить значение 10. Но если мы разрешим использовать каждому пользователю более одного FTP-клиента или разрешим регистрироваться одновременно с разных узлов под одним и тем же логином, тогда нужно увеличить `MaxInstances`. Например, если для `MaxHostsPerUser` мы установили значение 2, то `MaxInstances` будет равен 20 (2×10). В общем, вам, учитывая три значения (`MaxClients`, `MaxClientsPerUser` и `MaxHostsPerUser`), нужно высчитать максимальное значение `MaxInstances`, чтобы в моменты пиковой нагрузки все клиенты получили доступ к серверу:

```
MaxInstances 10
```

С помощью директивы `MaxLoginAttempts` можно задать, сколько раз пользователь может ввести пароль. После последней попытки сервер разорвет соединение. Рекомендуемое значение — 3.

`MaxRetrieveFileSize` — максимальный размер получаемого файла. Можно не устанавливать, потому как файлы, загружаемые на сервер вами, будете контролировать вы сами, а файлы, которые загружают пользователи — с помощью следующей директивы. Если никто не "залет" на сервер файл размером, скажем, в 1 Гбайт, то никто не сможет и скачать этот файл.

`MaxStoreFileSize` — максимальный размер файла, загружаемого на сервер пользователями. Тут все зависит от "ширины" канала и места на диске, даже больше от второго, нежели от первого. Решайте сами.

Нам осталось ограничить скорость передачи данных, чтобы сервис FTP не узурпировал под себя весь трафик. Особенно это важно, если канал "узкий", и на сервере запущены другие сетевые сервисы, например Apache.

Ограничить пропускную способность можно или с помощью устаревших директив `Rate*`, или с помощью новой `TransferRate`. Последнюю использовать удобно, если сервер подключен к Интернету по синхронному каналу. Если же сервер подключен по асинхронному каналу, т. е. скорости приема и передачи разные, удобнее использовать директивы `Rate*`, потому что они могут ограничить как скорость чтения, так и скорость записи:

- ❖ `RateReadBPS` *байт-в-секунду* — задает скорость чтения данных в байтах в секунду;
- ❖ `RateWriteBPS` *байт-в-секунду* — максимальная скорость записи данных в байтах в секунду;
- ❖ `TransferRate` *байт-в-секунду* — одновременно ограничивает как скорость чтения, так и записи.

42.2.3. Оптимизация Apache

Конфигурационный файл сервера Apache `httpd.conf` находится в каталоге `/etc/apache` или в `/etc/httpd/conf` (в зависимости от дистрибутива и версии Apache). В этом файле, как и в `proftpd.conf`, есть директива `MaxClients`, позволяющая ограничить число одновременно работающих клиентов.

Чтобы правильно установить это значение, нужно знать, сколько пользователей может одновременно зайти на сервер. При небольшой посещаемости вполне хватит значения 30—50, при большой загрузке количество одновременно работающих клиентов может исчисляться сотнями. Следите за посещаемостью вашего сервера и корректируйте это значение, иначе какая-то часть пользователей может остаться "за бортом", а им это очень не понравится (или же, наоборот, ресурсы сервера будут использоваться нерационально).

Директива `StartServers` задает количество экземпляров сервера, которые будут созданы при запуске исходной копии сервера. Для этой директивы можно установить значение, равное 10% от `MaxClients`. Устанавливать большое значение не нужно, поскольку вы будете нерационально использовать ресурсы компьютера.

Рассмотрим обычную ситуацию. Для `MaxClients` вы установили значение 200, а для `StartServers` — 20. Запросы первых 20 клиентов будут обрабатываться очень быстро, поскольку сервисы уже запущены. Запрос 21 клиента будет обслужен чуть медленнее, поскольку нужно запустить еще одну копию Apache, но не нужно устанавливать в нашем случае (`MaxClients` = 200) для `StartServers` значение больше 20 — ведь не всегда даже 20 человек одновременно заходят на сервер.

Если же на сервере постоянно находятся как минимум 20 человек, тогда нужно увеличить и `MaxClients`, и `StartServers`. Хотя бывают исключения, например сервер внутренней корпоративной сети. Вы точно знаете, сколько клиентов в вашей сети, следовательно, можно точно знать, какое значение установить для `MaxClients` и `StartServers`. Но все равно для `MaxClients` нужно установить чуть большее значение, чем для `StartServers` — на всякий случай:

```
MaxClients 150
```

```
StartServers 100
```

Чтобы еще эффективнее оптимизировать работу Web-сервера, нужно знать, как он работает. Клиент посылает запрос, Web-сервер его обрабатывает и посылает клиенту ответ. После этого соединение можно закрывать и завершать копию Apache, обслуживающую это соединение. Как видите, постоянных соединений, как в случае с FTP-сервером, здесь нет.

Но зачем завершать копию Web-сервера, если сейчас же на сайт зайдет другой пользователь, и опять нужно будет запускать еще одну копию сервера, а это увеличивает загрузку процессора. Поэтому с помощью директивы `MaxSpareServers` можно установить максимальное число серверов, которые будут находиться в памяти уже после закрытия соединения с пользователем — они будут просто ждать своего пользователя.

Теоретически, чтобы сбалансировать нагрузку, значение для `MaxSpareServers` можно установить такое же, как и для `StartServers`, т. е. 10% от `MaxClients`.

Вы не задумывались, что если Web-сервер будет работать в режиме постоянного соединения, как в случае с FTP, то это повысит его производительность? Если вы подумали об этом, то вы мыслите в правильном направлении. Представим, что у нас на сайте есть форум. Человек редко заходит на форум, чтобы посмотреть одну страничку. Обычно он может находиться на форуме часами. Так зачем же закрывать соединение? Чтобы потом опять тратить время и ресурсы сервера на его открытие? Разрешить постоянные соединения можно с помощью директивы `KeepAlive`. Она задает максимальное число таких соединений:

```
KeepAlive 5
```

А директива `KeepAliveTimeout` задает тайм-аут для постоянного соединения в секундах:

```
KeepAliveTimeout 15
```

Используя все упомянутые в этом разделе директивы, вы сможете добиться существенного повышения производительности вашего Web-сервера.

ГЛАВА 43



Сервер MySQL

43.1. Сервер баз данных MySQL

В мире Linux сервер баз данных MySQL является одним из самых популярных серверов баз данных. Его популярности способствует высокая производительность, ну и, конечно же, свободное распространение.

Сервер MySQL нужен на любом Web-сервере и на рабочей станции программиста, для тестирования программного обеспечения. Не секрет, что многие Web-приложения используют в качестве сервера баз данных именно MySQL, поэтому не установить его на Web-сервере просто нельзя. Ведь ни один серьезный PHP-проект не обходится без использования MySQL. Несмотря на то, что PHP поддерживает и другие серверы баз данных, в PHP-проектах используется именно MySQL, что также обуславливает его популярность.

Кстати, совсем не обязательно устанавливать сервер MySQL на одном компьютере с Web-сервером, вы можете установить его на другом компьютере сети, тем самым сбалансировав нагрузки на серверы сети.

43.2. Установка сервера

Для установки MySQL-сервера запустите менеджер пакетов и выберите следующие пакеты:

◆ mysql-server-5.0; ◆ mysql-client-5.0; ◆ mysql-admin.

На рис. 43.1 изображен менеджер пакетов Synaptic (Ubuntu Linux).

Первый пакет содержит последнюю версию MySQL-сервера (на данный момент это пятая версия), во втором пакете находится MySQL-клиент, т. е. программа, которая будет подключаться к MySQL-серверу, передавать ему SQL-запросы и отображать результат их выполнения. Третий пакет содержит программу для администрирования MySQL-сервера. Все необходимые дополнительные пакеты будут установлены автоматически. Выбранная конфигурация занимает 32 Мбайт в сжатом виде и 73 Мбайт займет после установки на жесткий диск. Другими слова-

ми, на момент установки на вашем жестком диске должно быть не менее 105 Мбайт (рис. 43.2).

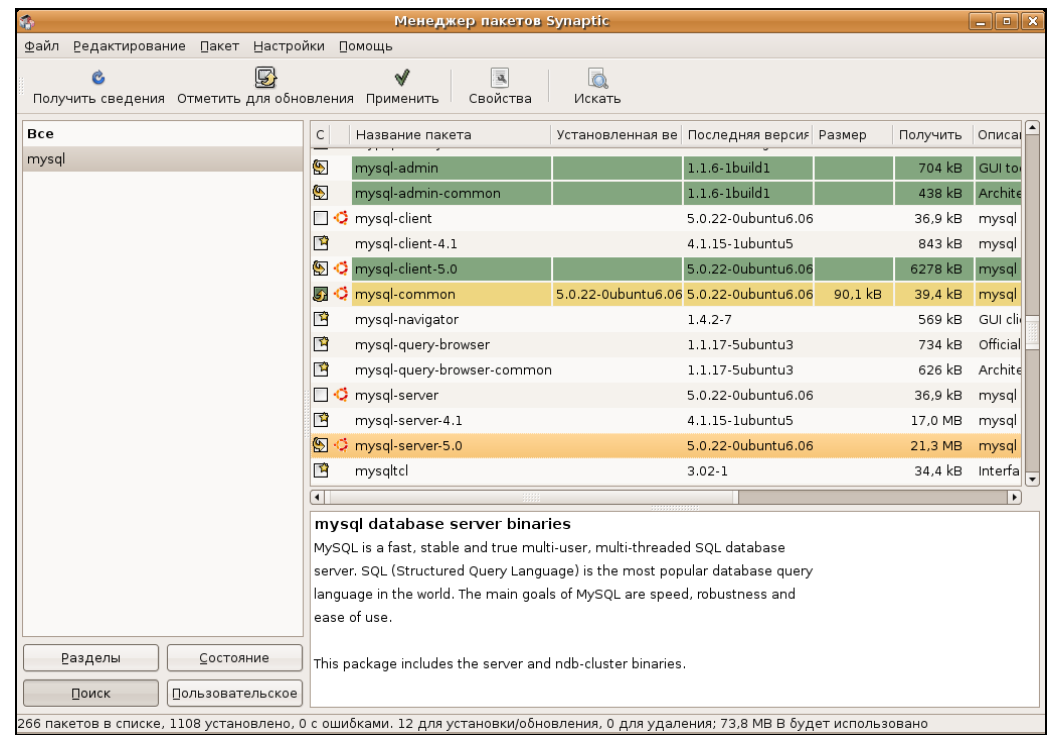


Рис. 43.1. Выбранные пакеты

43.3. Изменение пароля root и добавление пользователей

Сразу после установки выполните команду:

```
sudo mysqladmin -u root password ваш_пароль
```

Данный пароль вы будете использовать для администрирования сервера (данный пароль может и должен отличаться от того, который вы используете для входа в систему).

Для обычной работы с сервером рекомендуется создать обычного пользователя. Для этого введите команду:

```
mysql -u root -p mysql
```

Программа mysql является клиентом MySQL-сервера. В данном случае она должна подключиться к базе данных mysql (служебная база данных), используя

имя пользователя `root` (`-u root`). Поскольку вы только что указали пароль для пользователя `root` (до этого пароль для `root` не был задан), вам нужно указать параметр `-p`. После того как программа `mysql` подключится к серверу, вы увидите приглашение программы.

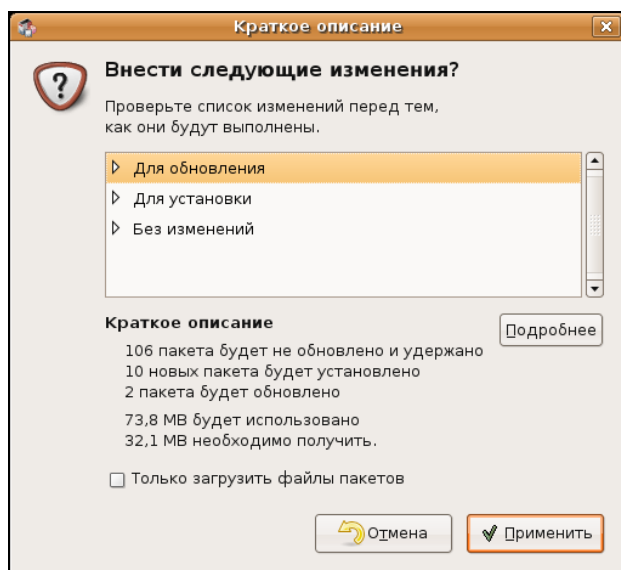


Рис. 43.2. Занимаемое место

В ответ на него нужно ввести следующий SQL-оператор:

```
insert into user(Host, User, Password, Select_priv,
                Insert_priv, Update_priv, Delete_priv)
values ('%', 'username', password('123456'), 'Y', 'Y', 'Y', 'Y');
```

Только что мы создали пользователя с именем `username` и паролем `123456`. Данный пользователь имеет право использовать SQL-операторы `select` (выборка из таблицы), `insert` (добавление новой записи в таблицу), `update` (обновление записи), `delete` (удаление записи). Если вам нужно, чтобы ваш пользователь имел право создавать и удалять таблицы, тогда добавьте привилегии `Create_priv` и `Drop_priv`:

```
insert into user(Host, User, Password, Select_priv, Insert_priv,
                Update_priv, Delete_priv, Create_priv, Drop_priv)
values ('%', 'username', password('123456'), 'Y', 'Y', 'Y', 'Y', 'Y', 'Y');
```

Данный SQL-оператор можно записать в одну строку, можно разбить на несколько строк — как вам будет удобно. Но в конце каждого SQL-оператора должна быть точка с запятой! Помните об этом.

Для выхода из программы `mysql` нужно ввести команду `quit`.

Кроме программы `mysql`, в состав MySQL-клиента входит одна очень полезная программа — `mysqlshow`, которая может вывести список таблиц, которые находят-

ся в той или иной базе данных. Кроме этого, она еще много чего может, но в данный момент нам нужен пока список таблиц — чтобы вы знали, какие таблицы есть в той или иной базе данных:

```
mysqlshow -p <база данных>
```

43.4. Базовые MySQL-операторы

В этом небольшом разделе, который никак не претендует на полное руководство по SQL, мы рассмотрим пять базовых SQL-операторов: CREATE, INSERT, UPDATE, SELECT и DELETE (имена операторов можно писать как строчными, так и заглавными буквами, разницы нет).

Оператор CREATE используется для создания различных объектов в базе данных. Мы его будем использовать для создания таблиц. Синтаксис данного оператора следующий:

```
CREATE table имя_таблицы
(имя_поля : тип_поля модификатор,
...
имя_поля : тип поля модификатор,
[ключи]);
```

Ключ управляет порядком отображения записей в таблице. Бывают первичный и вторичный ключи. Первичный строится только по одному полю таблицы, значение этого поля должно быть уникальным, т. е. в таблице не может быть двух полей с одинаковыми значениями ключевого поля. Вторичный ключ может строиться по нескольким полям, и условие уникальности не обязательное. Обычно вторичные ключи используются для связи таблиц.

Создадим небольшую таблицу:

```
CREATE table friends
(id          : int auto_increment,
name        : varchar(50),
email       : varchar(30),
comment     : text,
primary key (id));
```

Наша таблица friends будет содержать четыре поля:

- ❖ поле id — может принимать целые значения (это будет номер записи), значения поля увеличиваются автоматически (auto_increment). Это сделано для нашего удобства: при добавлении записи вам не нужно будет помнить номер последней записи — его MySQL укажет за вас;
- ❖ поле name — текстовое поле для имени вашего друга, максимальная длина этого поля — 50 символов;
- ❖ поле email — тоже текстовое поле, но с длиной строки 30 символов;

❖ поле `comment` — поле типа `TEXT`, может содержать небольшой текст объемом до 64 Кбайт.

Первичным ключом является поле `id`.

После этого добавим запись в таблицу. Для этого используется оператор `INSERT`. Синтаксис его следующий:

```
INSERT INTO таблица(список полей, в которые нужно вставить значения)
VALUES(список значений);
```

Пример:

```
INSERT INTO friends (name, email, comment)
VALUES ('Игорь', 'igor@somehost.ru', 'Мой друг');
```

Как вы видите, мы не указываем значение для поля `id`, потому что его вообще нет в списке полей. Это правильно: его значение будет указано автоматически.

Если вам лень писать список полей (это в нашей таблице всего 4 поля, причем одно можно не указывать, а в реальных таблицах у вас будут десятки полей), то список полей можно не указывать. Но тогда вам нужно будет указывать значения для каждого поля, т. е. не указывать значение для поля `id` мы уже не сможем:

```
INSERT INTO friends
VALUES (0, 'Игорь', 'igor@somehost.ru', 'Мой друг');
```

Для поля `id` мы указали значение 0, вообще можно указать любое целое значение, оно все равно будет установлено автоматически. Если в таблице вам понадобятся не целые, а вещественные значения, тогда вам нужно использовать тип поля `DOUBLE`, обеспечивающий наибольшую точность.

Для выборки записей из таблицы используется оператор `SELECT`. Синтаксис этого оператора довольно сложный, поэтому рассмотрим его сокращенную версию:

```
SELECT список_полей
FROM таблица
[WHERE условие];
```

Например:

```
SELECT *
FROM friends;
```

Данный оператор выведет все записи из таблицы `friends`. Вместо списка полей мы указали `*`, поэтому будут выведены все поля. Понятно, что в нашей таблице есть поле `comment`, которое может содержать большой текст, и его вывод, мягко говоря, не желателен, поэтому можно использовать следующий оператор:

```
SELECT id, name, email
FROM friends;
```

Данный оператор также выведет все записи (пока у нас только одна запись) из таблицы `friends`, будут выведены все поля, кроме поля `comment`. Чтобы увидеть поле `comment`, которое соответствует определенной записи, нам нужно использовать условие оператора `SELECT`.

Предположим, что вам нужно вывести поле `comment` для первой записи (обычно ее `id` будет равен 1):

```
SELECT comment
FROM friends
WHERE id=1;
```

Для обновления записей используется оператор `UPDATE`. Вот его синтаксис:

```
UPDATE таблица
SET список_полей_и_значений
WHERE условие;
```

Список полей и значений задается так:

```
поле = значение
```

Например:

```
UPDATE friends
SET name='Igor', email='777@host.ru'
...
```

В случае с оператором `UPDATE` мы обязательно должны указать условие, иначе указанные нами значения полей будут установлены для всех записей, а это не желательно. Чтобы изменить запись, мы должны ее идентифицировать. Поскольку у нас есть ключ, сделать это достаточно просто: нужно указать значение ключевого поля:

```
UPDATE friends
SET name='Igor', email='777@host.ru'
WHERE id=1;
```

Данный запрос найдет запись, для которой поле `id` равно 1, и изменит значения ее полей `name` и `email`. Поле `comment` останется без изменений, как и поле `id`.

Совсем другая была бы ситуация, если бы поле `id` не было ключевым, т. е. могло содержать повторяющиеся значения — теоретически в нашей таблице могло существовать две записи с одинаковым значением этого поля. Поскольку нам нужно обновить только одну запись, а не все, которые содержат такое же значение поля `id`, то нам нужно указать дополнительное условие `LIMIT`:

```
UPDATE friends
SET name='Igor', email='777@host.ru'
WHERE id=1 LIMIT 1;
```

`LIMIT` задает количество записей, которые будут затронуты в результате обработки запроса. В этом случае будет изменена одна из записей (обычно первая попавшаяся). Но что делать, если нам нужно изменить не любую запись с заданным условием, а конкретную запись? Наверное, вы уже догадались: нам нужно идентифицировать запись по другому полю, например по `email`:

```
UPDATE friends
SET name='Igor', email='777@host.ru'
WHERE email='user@somehost.ru' LIMIT 1;
```

Понятно, что если запись со значением `user@somehost.ru` поля `email` не будет найдена, то и обновления не будет. Для большей точности можно еще указать поле `id`:

```
UPDATE friends
SET name='Igor', email='777@host.ru'
WHERE (id=1) and (email='user@somehost.ru')
LIMIT 1;
```

Вот теперь точно будет обновлена нужная запись.

Условия удаления записи устанавливаются аналогично. Для удаления записей используется оператор `DELETE`:

```
DELETE
FROM таблица
WHERE условие;
```

Вот небольшой пример:

```
DELETE
FROM friends
WHERE id=1;
```

Данный оператор удаляет первую запись в таблице. Для удаления всех записей в таблице используется оператор:

```
DELETE
FROM friends;
```

Надеюсь, вы получили минимальное представление о SQL. Если это вас заинтересовало, то дополнительную информацию о синтаксисе SQL вы найдете на сайтах <http://dkws.org.ua/> и <http://mysql.ru>.

43.5. Запуск и останов сервера

Для управления сервером используется программа `/etc/init.d/mysql`. Для запуска сервера нужно передать этой программе параметр `start`, для останова `stop`, а для перезапуска `restart`:

```
sudo /etc/init.d/mysql start
sudo /etc/init.d/mysql stop
sudo /etc/init.d/mysql restart
```

В Mandriva/Fedora можно использовать команду `service`:

```
# service mysql start
# service mysql stop
# service mysql restart
```

Также для управления сервером можно использовать программу `mysqladmin`, узнать больше о ней можно с помощью команды:

```
man mysqladmin
```

ГЛАВА 44



Сетевая файловая система NFS

44.1. Установка сервера и клиента

Сетевая файловая система (Network File System) позволяет монтировать файловые системы, физически расположенные на удаленных компьютерах локальной сети. При этом работа с файловой системой осуществляется совершенно прозрачно, т. е. создается ощущение, что файловая система локальная, а не удаленная. Конечно, скорость доступа будет меньше — ведь данные нужно еще передать по сети, да и команда монтирования не совсем простая. Но все это нюансы.

Сетевая файловая система по принципу своей работы чем-то напоминает общие файлы и папки в Windows — там тоже можно предоставить свои ресурсы другим пользователям. Конечно, реализация другая, но общий принцип почти такой же. Не нужно путать NFS с Samba, средством для использования ресурсов сети Microsoft. Компьютеры, работающие под управлением Windows, не могут использовать NFS, равно как и с помощью NFS-клиента нельзя подключить общий ресурс Windows-станции. Поэтому первое, что нужно вам знать о NFS: эта служба может работать только между UNIX-компьютерами.

Архитектура NFS ничем не отличается от обычной архитектуры клиент-сервер. В сети есть один (или несколько) NFS-серверов, к которым подключаются NFS-клиенты с целью монтирования сетевых файловых систем (примонтировать можно не все файловые системы сервера, а лишь те, которые разрешил администратор). Если нужно, NFS-серверов может быть несколько.

Для установки сервера в Ubuntu/Debian нужно установить пакеты `nfs-common` и `nfs-user-server`. Для установки клиента хватит одного пакета `nfs-common`. В Mandriva/Fedora Core нужно установить пакет `nfs-utils`. Данный пакет содержит как NFS-сервер, так и NFS-клиент.

44.2. Настройка сервера

В файле `/etc/exports` прописываются экспортируемые файловые системы (которые могут монтировать удаленные пользователи). В листинге 44.1 приведен небольшой пример этого файла (по умолчанию файл пуст).

Листинг 44.1. Пример файла `/etc/exports`

```
/mnt/disk1 (ro, all_squash)
/mnt/upload admin.firma.ru(rw)
```

Формат этого файла следующий:

файловая_система [*компьютер*] (*опции*)

Первое поле файла — это экспортируемая файловая система. Она может экспортироваться на все компьютеры или же на один. Поле *компьютер* не обязательное, его надо указывать, если нужно предоставить доступ только определенному компьютеру или же указать специальные параметры доступа для определенного компьютера. Например, одна и та же файловая система может быть доступна всем компьютерам сети для чтения, а одному компьютеру сети — и для записи. Третье поле (*опции*) позволяет задать параметры доступа к файловой системе.

Проанализируем листинг 44.1. Файловая система `/mnt/disk1` доступна всем компьютерам только для чтения. Файловую систему `/mnt/upload` может использовать только пользователь `root` компьютера `admin.firma.ru`. Доступ полный (чтение/запись).

Опции, которые можно использовать в файле `exports`, приведены в табл. 44.1.

Таблица 44.1. Опции NFS

Опция	Описание
<code>secure</code>	Запросы на монтирование файловой системы могут поступать от портов с номерами меньше 1024. Такие порты может создавать только <code>root</code> , поэтому соединение считается безопасным (его не могут создать обычные пользователи). Используется по умолчанию
<code>insecure</code>	Запросы могут поступать с любых портов
<code>ro</code>	Монтирование экспортируемой файловой системы возможно в режиме "только чтение"
<code>rw</code>	К экспортируемой файловой системе разрешен полный доступ. Используйте с осторожностью!
<code>noaccess</code>	Запрещает доступ к файловой системе. Может использоваться для запрещения доступа конкретному компьютеру: <code>/mnt/public comp.firma.ru (noaccess)</code>

Таблица 44.1 (окончание)

Опция	Описание
<code>link_absolute</code>	Не изменяет символические ссылки. Используется по умолчанию
<code>link_relative</code>	Преобразует абсолютные ссылки в относительные
<code>all_squash</code>	Идентификаторы групп и пользователей будут преобразованы в анонимные
<code>no_all_squash</code>	Противоположна предыдущей опции. Используется по умолчанию
<code>root_squash</code>	Используется для преобразования всех запросов от <code>root</code> в запросы от анонимного пользователя. Используется по умолчанию
<code>no_root_squash</code>	Разрешает доступ к файловой системе от имени <code>root</code> . Противоположна опции <code>root_squash</code>

44.3. Монтирование удаленных файловых систем

Подмонтировать удаленную файловую систему можно с помощью все той же команды `mount`. Формат команды следующий:

```
mount -t nfs сервер:ФС точка_монтирования
```

Например,

```
mount -t nfs 192.168.1.1:/mnt/disk1 /mnt/remote
```

В нашем случае файловая система `/mnt/disk1` экспортируется сервером `192.168.1.1`. Она будет примонтирована к каталогу `/mnt/remote`. Параметр `-t` задает тип файловой системы — `nfs`.

Если нужно, чтобы данная файловая система монтировалась автоматически при загрузке системы, в файл `/etc/fstab` нужно добавить следующую запись:

```
192.168.1.1:/mnt/disk1 /mnt/remote nfs bg,hard,rw 0 0
```



ЧАСТЬ IX

ЗАЩИТА LINUX-СЕРВЕРА

ГЛАВА 45



Антивирус ClamAV

45.1. Зачем нужен антивирус в Linux

Linux считается одной из самых безопасных операционных систем. Она устойчива, ее сетевые сервисы надежны и... для Linux существует очень мало вирусов. Почему? Давайте подумаем. Представим на некоторое время, что мы — вирусописатели. Для какой операционной системы вы бы написали вирус? Для той, в которой работает на данный момент большинство компьютеров и которая более доступна в плане внедрения вируса? Или для той, которая не так популярна, как первая, и в несколько раз неприступнее? Думаю, вы бы выбрали первый вариант. Вот такой вариант как раз и есть Windows. Начнем с того, что для DOS было написано очень много вирусов, и все они по наследству перешли в Windows. Но система Windows несла в себе не только новые функции, но и новые ошибки, каждая из которых порождала новую волну вирусов. Не успевали в Microsoft закрыть одну "дыру", как появлялась следующая. Чего только стоит дырявый Internet Explorer, через который буквально за 10—15 минут в Интернете может проникнуть в систему целая армия троянов, сетевых червей и прочей нечисти. Windows, с ее передовыми и непроверенными технологиями — отличная цель для вирусописателей. Ведь это, в какой-то степени, творческие люди. И им интересно, чтобы их "творение" развивалось. А в Linux развитие вируса пресекает сама операционная система. Предположим, что Linux-пользователь скачал какой-то вирус для Linux. И даже запустил его. Максимум, что может сделать вирус — это повредить файлы в домашнем каталоге пользователя. Ведь для всего остального у него не хватит полномочий. А если вирус запустит пользователь root? Да, вирус в этом случае сможет нанести ущерб системе. Но, скажем так, это единичный случай. Все грамотные Linux-пользователи никогда не запускают ничего подозрительного под пользователем root и вообще ежедневную работу выполняют под обычным пользователем, а под пользователем root выполняют только системно-важные операции, а просмотр WWW к ним, как мы знаем, не относится. Да и Linux-браузеры не содержат такого огромного количества "дыр", как IE.

Если вирусов под Linux нет, спрашивается: а зачем же тогда нужен антивирус? Антивирус нужен как раз для обеспечения безопасности Windows-машин. Боль-

шинство антивирусов для Linux предназначены для установки на шлюзах — машинах, которые предоставляют доступ к Интернету. Установив антивирус на шлюзе, вы сможете контролировать трафик, проходящий через шлюз. Таким образом, вы защитите Windows-машины от проникновения вируса. Охрану ставят на входе, не так ли? Конечно, антивирус на шлюзе — это не панацея. Не нужно рассчитывать, что он на все 100% обезопасит вашу сеть. Желательно, чтобы на каждой Windows-машине был установлен отдельный антивирус, работающий в режиме монитора.

В этой главе мы будем рассматривать бесплатный антивирус ClamAV (<http://www.clamav.net>). Почему именно ClamAV, а не какой-нибудь коммерческий антивирус вроде DrWeb или Kaspersky AntiVirus? Коммерческие антивирусы сопровождаются хорошей документацией, в которой вы разберетесь и без моих комментариев, да и не хочется отбирать хлеб у службы поддержки коммерческих антивирусов.

45.2. Установка ClamAV

Для работы ClamAV нужно установить три пакета (если пакетов нет в составе вашего дистрибутива, то их можно скачать с сайта www.clamav.net):

- ◆ clamav — сканер;
- ◆ clamav-db — антивирусная база данных;
- ◆ clamd — демон Clam.

Сразу после установки нужно установить соединение с Интернетом (если оно еще не установлено) и выполнить обновление антивирусной базы данных:

```
# clamd
# freshclam
```

Первая команда запускает демон Clam, чтобы у `freshclam` (выполняет обновление базы данных) была возможность сообщить демону об удачном обновлении баз данных.

ПРИМЕЧАНИЕ

Команды `clamd` и `freshclam` нужно запускать от имени пользователя `root`. Напомню, что для этого не нужно входить в систему как `root`: достаточно использовать команды `su` или `sudo`.

45.3. Проверка файловой системы

Сомневаюсь, что в вашей файловой системе будут вирусы (не забываем, что мы используем одну из самых безопасных операционных систем), но все же лучше запустить сканер:

```
# clamscan -r /
```

Данная команда проверит всю файловую систему. Если нужно проверить только отдельный каталог, то вместо / укажите имя каталога.

45.4. Прозрачная проверка почты

Сейчас мы настроим прозрачный почтовый антивирус. Почтовый антивирус чрезвычайно актуален, ведь большинство так называемых сетевых червей распространяются именно с помощью электронной почты.

Конечно, антивирус ClamAV можно использовать и в режиме обычного сканера, но наиболее интересен он в режиме почтового антивируса. Чуть раньше было сказано, что данный антивирус является прозрачным. Почему прозрачным? Обычный почтовый антивирус "прикручивается" к МТА-агентам путем внесения изменений в их конфигурационные файлы. Агент МТА "знает", что прежде чем передать письмо, его нужно проверить, вызвав прописанный в конфигурационном файле антивирус. Прозрачный антивирус действует независимо от МТА-агента. Более того, МТА-агент даже не подозревает о его существовании. Это очень удобно, хотя бы потому, что нам не нужно изменять конфигурацию МТА-агента. Вы когда-нибудь "прикручивали" антивирус, например, к sendmail? Если нет, то обязательно попробуйте, когда у вас будет свободное время. После этого вы оцените технологию "прозрачности" ClamAV.

Но простота внедрения — это не единственное преимущество ClamAV. Представьте, что у вас есть почтовый сервер, на котором вы развернули почтовый антивирус. Все бы хорошо — почта ведь проверяется. Но! Ведь у ваших сотрудников есть ящики не только на локальном почтовом сервере. Наверняка найдется несколько человек (если не подавляющее большинство), у которых есть почтовые ящики на бесплатных почтовых серверах, например на Mail.Ru. В этом случае вирус может попасть в вашу сеть, когда пользователь получает почту с сервера Mail.Ru. Наш антивирус будет бессилен, поскольку он контролирует только наш локальный сервер. Правильно настроенный ClamAV будет проверять абсолютно все почтовые соединения, т. е. соединения с 25 и 110 портами любых серверов.

Сам ClamAV является обычным антивирусом, а "прозрачным" его делает сервер P3Scan, скачать который можно по адресу <http://sourceforge.net/projects/p3scan/>.

Антивирус у нас уже установлен и работает, поэтому можно приступить к настройке P3Scan. Работать все будет так: iptables брандмауэра будет перенаправлять пакеты на порт, на котором запущен P3Scan. После этого начинает работать ClamAV, которому P3Scan передает для проверки почту. Неинфицированная почта будет отправлена клиенту.

Теперь, собственно, настройка. Отредактируйте файл /etc/p3scan/p3scan.conf следующим образом:

```
virusregex = .*: (.* ) FOUND
scanner   = /usr/bin/clamscan --no-summary -i
scannertype = basic
```

Если нужно, измените путь к ClamAV.

Все, что осталось сделать — это создать правило перенаправления POP3-трафика на порт 8110 (на этом порту работает P3Scan):

```
# iptables -t nat -A PREROUTING -p tcp --dport 110 -j REDIRECT --to 8110
```

45.5. Проверка Web-трафика

Почта — это не единственный способ распространения сетевых червей и прочей нечисти. Очень много вирусов распространяются по WWW, поэтому нам нужно (на шлюзе) перехватить WWW-трафик, проверить его антивирусом, и если трафик "чистый", передать его пользователю.

Работать прозрачный антивирус Web-трафика будет на базе уже установленного и настроенного прокси-сервера SQUID: SQUID будет получать запрашиваемый пользователем по WWW файл и с помощью программы Viralator передавать его антивирусу. Кроме программы Viralator, есть и другие программы, которые можно использовать для этой цели, но работать с Viralator проще. Также можно организовать передачу файлов между прокси-сервером и антивирусом с помощью стандартных редиректоров SQUID, но они не всегда работают корректно, поэтому мы их использовать не будем.

Скачать программу Viralator можно на сайте <http://viralator.sourceforge.net/>.

Кроме Viralator, нам понадобится запущенный на шлюзе Web-сервер Apache: через него и будет запускаться сценарий Viralator.

Теперь можно приступить к настройке. Настройки SQUID рассматривать не будем — с ними мы уже знакомы. На уже настроенный SQUID нужно установить SquidGuard (см. главу 36) и отредактировать его конфигурационный файл `/etc/squid/squidGuard.conf` (листинг 45.1).

Листинг 45.1. Конфигурационный файл `etc/squid/squidGuard.conf`

```
# Путь к базе SquidGuard и журналам
dbhome /usr/share/squidGuard-1.2.0/db
logdir /var/log/squidGuard

dest files {
expressionlist files-to-check.reg
}

acl {

# 10.0.0.1 — это IP Web-сервера, на котором установлен Viralator
default {
pass !files all
```

```
redirect
http://10.0.0.1/cgi-bin/viralator.cgi?url=%u
}
}
```

Данный конфигурационный файл заставляет SquidGuard передавать файлы, имена которых соответствуют регулярному выражению из файла `files-to-check.reg`, сценарию `viralator.cgi`, расположенному на Web-сервере.

Нам нужно создать файл `/usr/share/squidGuard-1.2.0/db/files-to-check.reg` и поместить в него следующее регулярное выражение:

```
(\.exe$|\.bat$|\.zip$|\.bin$|\.sys$|\.rar$)
```

Как несложно догадаться, данная строка задает типы файлов для проверки — эти типы файлов потенциально могут содержать вирусы. Можете отредактировать эту строку так, как считаете нужным.

Мы пока что связали сценарий `Viralator` со `SquidGuard`, но не связали сам `SquidGuard` со `SQUID`. Для этого откройте файл `/etc/squid/squid.conf` и добавьте в него следующие строки:

```
redirector_bypass on
redirect_program /usr/local/squidGuard/bin/squidGuard
```

```
# максимальное количество копий SquidGuard в памяти
redirect_children 20
redirector_access deny SSL_ports
redirector_access deny localhost
```

Теперь нужно настроить `Apache`. Откройте его конфигурационный файл `/etc/httpd/conf/httpd.conf` и отредактируйте следующие директивы:

```
# указываем IP нашего Web-сервера
Listen 10.0.0.1:80
ServerName 10.0.0.1
```

Не забудьте после этого запустить `Apache`. Теперь приступим непосредственно к настройке `Viralator`. Сценарий `viralator` нужно распаковать в каталог `/var/www/cgi-bin`, после чего нужно изменить владельца и права доступа сценария:

```
# chown apache:apache /var/www/cgi-bin/viralator.cgi
# chmod +x /var/www/cgi-bin/viralator.cgi
```

Сценарий `Viralator` требует дополнительный Perl-модуль `LWP`. Для установки этого модуля нужно ввести команду:

```
# perl -MCPAN -e shell
```

А когда увидите приглашение `span>`, то введите команду:

```
install LWP
```

После этого перейдите в каталог `/var/www/cgi-bin` (именно в него вы должны были распаковать архив с `viralator`). В этом каталоге будет подкаталог `etc`, а в

нем — подкаталог `viralator`. Этот каталог `viralator` нужно скопировать в каталог `/etc`. После чего удалите каталог `etc` из каталога `/var/www/cgi-bin`.

Почти все готово. Осталось только отредактировать конфигурационный файл `Viralator` — `/etc/viralator/viralator.conf` (листинг 45.2).

Листинг 45.2. Файл `/etc/viralator/viralator.conf`

```
servername -> 10.0.0.1           # IP-адрес Web-сервера
antivirus -> CLAMAV              # мы используем ClamAV
virusscanner -> clamscan         # так называется программа-сканер
scannerpath -> /usr/bin          # а это путь к сканеру
viruscmd -> --remove             # опция сканера для удаления вирусов
alert -> FOUND                   # сообщение сканера о том, что найден вирус
downloads -> /var/www/html/downloads # этот каталог нужно создать
downloadsdir -> /downloads
default_language -> english.txt  # язык по умолчанию (русского нет)
# остальное можно не изменять
scannersummary -> true
popupfast -> false
popupback -> false
popupwidth -> 600
popupheight -> 400
filechmod -> 644
BAR -> bar.png
PROGRESS -> progress.png
```

Создайте каталог `downloads` и установите права доступа:

```
# mkdir /var/www/html/downloads
# chown apache:apache /var/www/html/downloads
# chmod 777 /var/www/html/downloads
```

Все настроено! Теперь машины наших клиентов нужно настроить на использование нашего прокси-сервера (10.0.0.1, порт 3128) и приступить к тестированию!

45.6. Клиентский антивирус

Какой антивирус лучше всего установить на компьютерах нашей сети, которые работают под управлением Windows? Несмотря на то, что есть Windows-версия ClamAV, я бы порекомендовал антивирус Касперского, поскольку ClamAV не всегда эффективно справляется с некоторыми угрозами. ClamAV, установленный на шлюзе, "отсеет" большую часть вирусов, а с теми, которые ClamAV пропустит, справится антивирус Касперского.

ГЛАВА 46



Защита популярных сетевых сервисов

46.1. Защита Apache

Apache — довольно безопасный сервис, поэтому его защита сводится к установке определенных прав доступа к его конфигурационным файлам. Для начала установим права 700 к каталогам `/etc/httpd/conf` и `/var/log/httpd`:

```
# chmod 700 /etc/httpd/conf
# chmod 700 /var/log/httpd
```

После этого никто, кроме вас, не сможет ни просмотреть, ни изменить конфигурационные файлы и файлы протоколов, которые обычно доступны всем желающим для чтения.

Также нужно защитить конфигурационный файл `httpd2.conf` от изменения:

```
# chattr +i /etc/httpd/conf httpd2.conf
```

После этого даже вы не сможете изменить данный файл. Если же вам понадобится отредактировать его, тогда нужно снять атрибут `i`:

```
# chattr -i /etc/httpd/conf httpd2.conf
```

46.2. Защита FTP

ProFTPD тоже является весьма защищенным сервисом, а его взлом является только следствием неправильной настройки. Взять бы даже директиву `DefaultRoot`, задающую корневой каталог для сервера.

Рекомендуется установить значение этой директивы в `~`. Как мы знаем, тильда (`~`) означает домашний каталог пользователя. Следовательно, каждый раз при регистрации пользователя на FTP-сервере корневым каталогом FTP-сервера станет домашний каталог пользователя. В результате пользователь не сможет прочитать (а при неправильных правах доступа — изменить) важные системные файлы.

Также рекомендуется включить директиву `RequireValidShell:`
`RequireValidShell on`

Если данная директива включена, тогда злоумышленник не сможет установить в качестве оболочки какую-нибудь вредоносную программу. FTP-сервер будет проверять, указана ли программа-оболочка в `/etc/shells`. Если программа не указана в этом файле, то FTP-сервер не будет ее запускать.

46.3. Защита DNS

Серверы DNS обмениваются между собой информацией о зоне. О том, как ограничить передачу зоны, мы говорили в *главе 34*. Но там мы ограничивали передачу зоны по IP-адресу. А что, если злоумышленник каким-то образом подменил целевой DNS-сервер с указанным адресом (например, вывел его из строя и запустил собственный с таким же IP)? Тогда информация о зоне будет передана на сервер злоумышленника.

Чтобы такого не случилось, нужно использовать механизм транзакций TSIG (Transaction SIGnatures). Данный механизм предусматривает перед передачей зоны проверку секретного ключа. Если ключ совпадает, информация о зоне будет передана/принята. Если же ключ не совпадает, информация о зоне не будет передана или не будет принята (если злоумышленник вывел из строя первичный DNS-сервер и пытается передать измененную информацию о зоне на вторичный DNS-сервер).

Первым делом нужно сгенерировать ключ, который затем указать в файле конфигурации каждого сервера. Для этого используется команда:

```
# dnssec-keygen -a hmac-md5 -b 128 -n HOST host1-host2
```

Вам изменять эту команду не стоит. Команда выведет на экран следующую строку:

```
Khost1-host2.код
```

Также в результате выполнения этой команды будут созданы два файла: `Khost1-host2.код.key` и `host2.код.private`. Откройте второй файл. В нем будут следующие строки:

```
Private-key-format: v1.2  
Algorithm: 157 (HMAC_MD5)  
Key: ключ
```

Строку *ключ* нужно скопировать в буфер обмена (или записать на бумаге, если у вас нет графического интерфейса). После этого нужно добавить следующие строки в файлы `named.conf` обоих DNS-серверов (первичного и вторичного):

```
key host1-host2 {  
    algorithm hmac-md5;  
    secret "ключ";  
};
```

ПРИМЕЧАНИЕ

Директиву `key` нужно добавить в начало файла конфигурации.

А теперь будьте внимательны и следуйте моим инструкциям. Откройте файл первичного DNS-сервера и добавьте следующие строки после директивы `key`:

```
server 10.0.0.2 {  
    key { host1-host2; };  
};
```

Затем в директиву `options` добавьте `allow-transfer`, если вы этого еще не сделали:

```
allow-transfer { 10.0.0.2; };
```

Директива `server` указывает, что для обмена зоной с сервером 10.0.0.2 (это IP-адрес вторичного сервера) нужно использовать ключ `host1-host2`.

Теперь откройте файл конфигурации вторичного DNS-сервера. Добавьте следующие строки:

```
server 10.0.0.1 {  
    key { host1-host2; };  
};
```

В директиву `options` нужно добавить вот такую директиву:

```
allow-transfer { none; };
```

После всего этого нужно перезапустить DNS-сервер:

```
# service named restart
```

46.4. Защита Samba

По умолчанию доступ к серверу Samba предоставляется всем желающим. Это не всегда необходимо. В целях большей безопасности нужно предоставлять доступ только определенным пользователям. Создать пользователей можно с помощью команды `adduser` (в Fedora нужно использовать конфигуратор `system-config-users`):

```
adduser -s /bin/false win_user  
passwd win_user  
smbpasswd win_user
```

Первая команда добавляет пользователя и устанавливает в качестве его командной оболочки программу `/bin/false`, которая запускается, возвращает код 0 и завершает работу. Даже если кто-то узнает пароль пользователя, `/bin/false` не позволит пользователю зарегистрироваться в системе обычным способом.

Вторая команда устанавливает пароль пользователя. Поскольку пользователь не сможет войти в систему, можно установить любой пароль и даже не запоминать его. А вот третья команда изменяет пароль пользователя, который он дол-

жен будет указать при регистрации на сервере Samba. Этот пароль нужно сообщить пользователю.

Затем нужно открыть `smb.conf` и в секции `global` изменить параметр `security`:
`security = user`

46.5. DHCP: привязка к MAC-адресу

Раньше все компьютеры сети настраивались вручную, сейчас настройкой узла занимается DHCP-сервер. Следовательно, если узел сети не получит настроек от DHCP-сервера, то и доступ к сети он не получит.

Со стороны DHCP-сервера можно блокировать доступ нежелательных компьютеров, точнее разрешить передачу настроек только тем компьютерам, которым нужно. Делается это путем привязки IP-адресов к MAC-адресам (аппаратные адреса сетевых адаптеров). При такой настройке мы убиваем сразу двух зайцев:

- ❖ одному и тому же компьютеру (точнее, MAC-адресу) будет всегда назначаться один и тот же IP-адрес, что очень удобно, если система статистики подсчитывает трафик по IP-адресу без аутентификации пользователя;
- ❖ компьютеры, MAC-адреса сетевых адаптеров которых вы не "прописали" в конфигурационном файле DHCP-сервера, не получают доступ к сети, потому что не получают сетевые настройки.

Следует помнить, что защита средствами DHCP-сервера весьма посредственна, но как дополнительный барьер вполне сойдет. Нужно помнить, что даже если узел не получит сетевые настройки, то их можно указать вручную. Узнать их для злоумышленника не составит особого труда. Кроме средств DHCP-сервера, нужно еще контролировать пространство IP-адресов вашей сети. Например, вы выделяете своим клиентам адреса из диапазона 192.168.1.50—192.168.1.100 (с помощью DHCP), но злоумышленник может указать IP-адрес 192.168.1.101. Если дальше никакие средства (ни прокси-сервер, ни брандмауэр) не осуществляют контроль IP-адресов, толку от контроля MAC-адресов не будет.

К тому же MAC-адрес довольно легко подделать. В Linux MAC-адрес для интерфейса `eth0` (первая сетевая плата — у многих она не только первая, но и единственная) можно изменить командами (XX:XX:XX:XX:XX:XX — MAC-адрес):

```
# ifconfig eth0 down
# ifconfig eth0 hw ether XX:XX:XX:XX:XX:XX
# ifconfig eth0 up
```

В Ubuntu (и других дистрибутивах, где используется NetworkManager) MAC-адрес можно изменить с помощью графического интерфейса, в openSUSE Yast, если я не ошибаюсь (а проверять лень, потому что сам для смены адреса использую приведенные ранее команды), тоже позволяет изменить MAC-адрес. В Windows MAC-адрес можно изменить в свойствах сетевой платы, на вкладке **Дополнительно**, свойство **Сетевой адрес** (рис. 46.1).

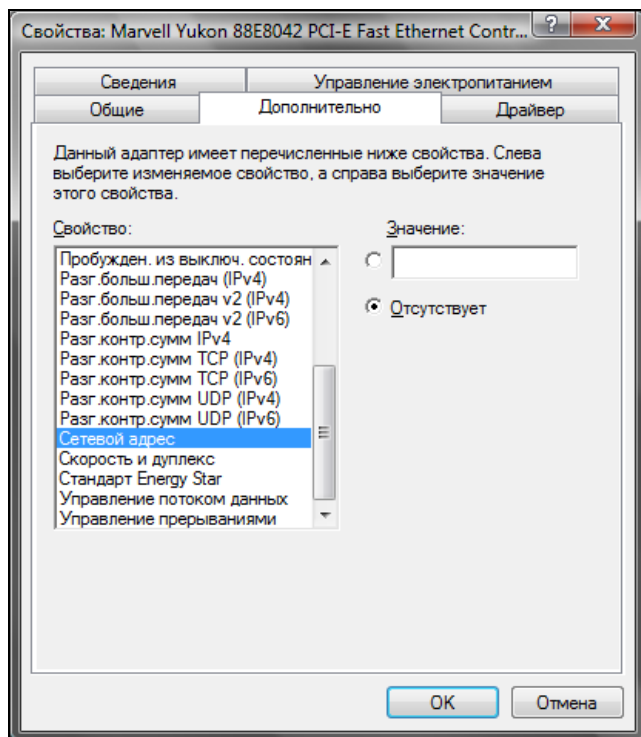
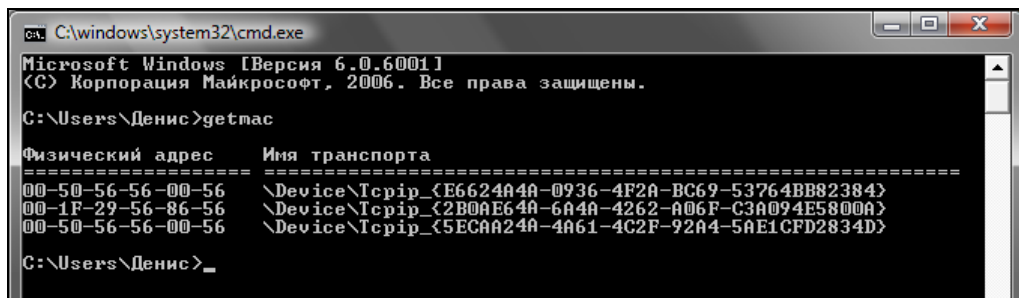


Рис. 46.1. Изменение MAC-адреса сетевого адаптера в Windows

После смены MAC-адреса нужно проверить, установился ли он:

- ♦ в Linux — `ifconfig -a | grep HWaddr`;
- ♦ в Windows — `ipconfig /all` или команда `getmac` (рис. 46.2);
- ♦ в FreeBSD — `ifconfig|grep ether`.

Рис. 46.2. Команда `getmac` в Windows

Спрашивается, а как злоумышленник узнает, какой MAC-адрес допустим? Ему достаточно подключиться (физически) к вашей сети, что он уже и сделал, раз пытается узнать MAC-адрес (и вправду, зачем мне MAC-адрес одного из компьютеров

Пентагона, если я не собираюсь подключаться к его сети?). После этого он может запустить одну из программ для сбора MAC-адресов, например TCPNetView (вы ее без проблем найдете в Интернете — программа распространяется бесплатно). Кстати, эта программа полезна и для системного администратора — ведь вам же не хочется вводить команду определения MAC-адреса на каждом компьютере? Вы запускаете ее и получаете MAC-адреса всех подключенных к сети в данный момент компьютеров. Удобно? Я тоже так думаю.

Учитывая все сказанное ранее, можно сделать вывод: защита средствами DHCP-сервера подходит больше для внутреннего контроля, нежели для защиты от взлома сети. Но как дополнительный барьер она вполне действительна.

Если вы-таки надумали реализовать привязку IP-адресов к MAC-адресам, тогда для каждого компьютера сети добавьте в конфигурационный файл следующую конструкцию:

```
host compN {  
    hardware ethernet xx:xx:xx:xx:xx:xx;  
    fixed-address IP-адрес;  
}
```

Все эти инструкции (для каждого компьютера) нужно добавить в секцию subnet, например:

```
subnet 192.168.1.0 netmask 255.255.255.0 {  
    ...  
    host comp3 {  
        hardware ethernet 00:40:AA:24:70:2E;  
        fixed-address 192.168.1.3;  
    }  
}
```

46.6. Защита от спама: greylisting и qmail

Спам — одна из потенциальных угроз для вашего сервера. Одно дело, что спам — это просто неприятно. Но совсем другое, когда поток спама "забивает" весь жесткий диск вашего сервера, и тогда почтовый сервер вообще не будет функционировать. Это типичный пример атаки на отказ (DoS, Denial of Service).

Для борьбы со спамом используются две общепринятые стратегии: черные списки и серые списки. Что такое черный список знают все: в черный список заносят адреса машин, с которых производится рассылка спама. Впоследствии наш почтовый сервер не принимает почту с этих адресов.

А теперь представим такую ситуацию. Сейчас редко используются статические адреса — только для серверов, а для клиентов в основном используют динамические адреса. Спамер подключается к Интернету как обычный клиент. Использоуе-

мый спамером динамический адрес будет внесен в черный список. На следующий день этот адрес будет назначен нормальному законопослушному пользователю. Когда он попытается отправить почту, то получит сообщение, что его адрес внесен в черный список.

Серые списки работают иначе. Если адрес компьютера есть в сером списке, то почтовый сервер в первый раз отправляет этому клиенту ответ, что сервер занят и что нужно повторить попытку позже. Спамер не будет повторять отправку письма — ему это не нужно, т. к. он уже начал отправлять письмо по другому адресу. А вот нормальный пользователь, получив такой ответ от сервера, через некоторое время повторит отправку письма. После этого адрес клиента будет внесен в белый список, и сервер всегда будет принимать с него письма.

В этой главе мы рассмотрим, как интегрировать механизм серых списков (greylisting) с qmail — поскольку в этой книге мы рассматриваем именно этот почтовый сервер. А вот по следующему адресу вы найдете инструкции по интеграции greylisting с другими МТА:

<http://spamlinks.net/filter-server-greylist.htm>

Для начала разберемся, что нам нужно:

- ❖ уже установленный МТА qmail или netqmail (<http://netqmail.org/>);
- ❖ патч QMAILQUEUE (<http://www.qmail.org/qmailqueue-patch>), позволяющий вставить другую программу в структуру очереди qmail;
- ❖ программа qmail-qfilter (<http://untroubled.org/qmail-qfilter/>), выполняющая фильтрацию писем.

После этого нужно создать скрипт `/var/qmail/control/qmail-qfilter/greylisting.qfilter`:

```
exec /var/qmail/bin/qmail-qfilter \  
/var/qmail/control/qmail-qfilter/greylisting.qfilter -- \  
/var/qmail/control/qmail-qfilter/yourfilter1 -- \  
/var/qmail/control/qmail-qfilter/yourfilter2
```

Затем следует установить права доступа к нему:

```
chown root:qmail /var/qmail/control/qmail-qfilter/greylisting.qfilter  
chmod 0755 /var/qmail/control/qmail-qfilter/greylisting.qfilter
```

Скопируйте cron-скрипт greylisting в каталог crontab и установите права доступа:

```
chown root:root greylisting.cron  
chmod 0755 greylisting.cron
```

Данный скрипт управляет базой данных Greylisting. Для самой базы данных нужно создать каталог и установить разрешения:

```
mkdir -p /var/greylisting  
chown qmail:nofiles /var/greylisting  
chmod 0700 /var/greylisting
```

Затем нужно установить переменную окружения QMAILQUEUE — в ней нужно указать путь к сценарию `/var/qmail/control/qmail-qfilter/greylisting.qfilter`. Более подробные инструкции вы найдете по адресу:

<http://spamlinks.net/filter-server-greylist.htm#implement-qmail>

ГЛАВА 47



Chroot-окружения

47.1. Песочница

Представьте детей, играющих в песочнице. Песочница у нас не обычная, а с высокими бортами, поэтому дети не могут самостоятельно из нее выбраться. Понятно, что в песочнице дети в большей безопасности, нежели чем за ее пределами.

Теперь представим, что дети — это пользователи системы. chroot-окружение похоже на песочницу, но оно предназначено не для защиты пользователей, а для защиты системы от действий пользователей. Сейчас разберемся, как работает chroot-окружение.

Предположим, что у нас есть сетевая служба, например FTP-сервер. Если пользователю удастся каким-то образом "взломать" данную службу, то он получит доступ к корневой файловой системе сервера, что нежелательно.

При создании chroot-окружения создается набор файлов, содержащий все необходимое для запуска того или иного сетевого сервиса. Под набором файлов подразумевается отдельный каталог, в который копируются все необходимые файлы: конфигурационный файл, исполняемые файлы самого сервиса, библиотеки, вспомогательные программы. Затем производится системный вызов chroot, делающий подмену файловой системы. Наш сетевой сервис уже запускается внутри chroot-окружения. Если даже пользователь взломает сервис, то он получит доступ не к файловой системе сервера, а к файловой системе chroot-окружения. Чтобы он ни сделал, его действия не причинят системе никакого вреда.

Но дети вырастают и со временем могут выйти за пределы песочницы. Точно так же растут возможности злоумышленников — выход за пределы chroot-окружения возможен. Но, несмотря на это, chroot-окружение остается очень мощным барьером для злоумышленников.

47.2. Пример создания chroot-окружения

Давайте рассмотрим создание chroot-окружения для Web-сервера Apache. Первым делом нужно создать каталог, в котором мы будем формировать chroot-окружение. Пусть это будет каталог chroot:

```
# mkdir /chroot
```

Далее нужно создать все необходимые для работы Apache каталоги (позже мы скопируем в них необходимые файлы):

```
# mkdir -p /chroot/etc
# mkdir -p /chroot/dev
# mkdir -p /chroot/usr/lib
# mkdir -p /chroot/usr/libexec
# mkdir -p /chroot/usr/local/apache/bin
# mkdir -p /chroot/usr/local/apache/logs
# mkdir -p /chroot/usr/local/apache/conf
# mkdir -p /chroot/var/www/html
# mkdir -p /chroot/var/run
```

Установим права доступа:

```
# chown -R root:sys /chroot
```

После этого мы должны создать устройства dev/log и dev/null. Первое необходимо для нормальной работы демона syslogd в chroot-окружении, а второе будет использоваться в качестве домашнего каталога Web-сервера:

```
# mknod /chroot/dev/null c 2 2
# chown root:sys /chroot/dev/null
# chmod 666 /chroot/dev/null
# mknod /chroot/dev/log c 21 5
# chown root:sys /chroot/dev/log
# chmod 666 /chroot/dev/log
```

После этого скопируйте все необходимые для работы Web-сервера файлы:

- ❖ конфигурационные файлы (находятся в каталоге /etc/httpd2);
- ❖ каталог документов (/var/www/html);
- ❖ все остальные файлы.

Чтобы понять, какие файлы нужны для работы Apache, выполните команды:

```
# ldd /usr/sbin/apache2
# strings /usr/sbin/apache2
# strace /usr/sbin/apache2
```

Внимательно следите за выводом этих команд. Если в выводе встретится название файла, данный файл нужно скопировать в каталог /chroot (точнее, в соответствующий подкаталог каталога /chroot). Например, если серверу нужен файл

/var/www/html/index.php, то данный файл следует скопировать в каталог /chroot/var/www/html.

ПРИМЕЧАНИЕ

Программа `strace` выводит список всех системных вызовов, которые порождает Apache во время своей работы. Вам нужно обращать внимание только на системные вызовы `open()`.

Нам осталось лишь создать базу данных паролей в chroot-окружении и запустить Apache. Для создания базы паролей введите следующие команды:

```
# touch /chroot/etc/passwd
# echo "nobody:x:65534:65534:none:/:sbin/nologin" >> /chroot/etc/passwd
# echo "www:x:80:80:www:/:sbin/nologin" >> /chroot/etc/passwd
# touch /chroot/etc/group
# echo "nobody:x:65534:" >> /chroot/etc/group
# echo "www:x:80:" >> /chroot/etc/group
```

Теперь запустим Apache в созданном нами chroot-окружении:

```
# /usr/sbin/chroot /chroot /usr/sbin/apache2
```

Первый аргумент команды `chroot` — это каталог, в котором мы создали chroot-окружение, а второй — это исполняемый файл Web-сервера.

ГЛАВА 48



Управление доступом

48.1. Что такое Tomsy

Все мы знакомы с системами ограничения доступа SELinux, LIDS и GrSecurity. В этой небольшой главе вы познакомитесь с модулем безопасности Tomsy. SELinux в этой книге рассматривать не будем — в Интернете, да и в других моих книгах, есть необходимая информация. SELinux — уже не интересно. Вместо этой системы в этой книге мы рассмотрим Tomsy.

Tomsy исследует поведение каждого процесса, просматривает используемые процессом ресурсы и на основании полученной информации разрешает или запрещает выполнение процесса. Кроме того, Tomsy можно использовать в качестве утилиты системного анализа, т. е. этот модуль можно использовать для отладки приложений, написания технической документации и изучения принципов работы системы. Инструмент для настоящих хакеров. (Не забывайте, что хакер — это не тот, кто взламывает и разрушает, а тот, кто создает!)

Tomsy можно использовать для защиты вашей системы, например для защиты от операций внедрения команд операционной системы, ограничения действий SSH-сервисов и т. д.

Сразу нужно отметить, что данная глава не для начинающих пользователей. Как минимум вы должны знать, как откомпилировать ядро в вашем дистрибутиве (процедура компиляции ядра в разных дистрибутивах слегка отличается).

48.2. Установка Tomsy. Готовые LiveCD

Прежде чем установить Tomsy, можно скачать уже готовые LiveCD, собранные с поддержкой Tomsy. В практическом плане толку от этих LiveCD мало, но в теоретическом — это как раз то, что вам нужно. Вы можете увидеть Tomsy в дей-

ствии, не устанавливая на свой сервер (да, именно на сервер, поскольку на домашнем компьютере толку от Tomoyo мало). Скачать LiveCD можно по адресам:

<http://tomoyo.sourceforge.jp/en/1.6.x/1st-step/centos5-live/>
<http://tomoyo.sourceforge.jp/en/1.6.x/1st-step/mandriva2009.0/>
<http://tomoyo.sourceforge.jp/en/1.6.x/1st-step/f8/>
<http://tomoyo.sourceforge.jp/en/1.6.x/1st-step/ubuntu8.04-live/>

Первая ссылка — это дистрибутив CentOS 5, собранный с поддержкой Tomoyo. Остальные ссылки — это, соответственно, дистрибутивы Mandriva 2009, Fedora 8 и Ubuntu 8.04. Да, дистрибутивы не очень новые, но для ознакомления вполне сойдут.

Если Томою вам понравился, самое время его скачать и установить. Современная версия Томою требует ядро Linux 2.6.30 или более новое. Самая новая версия ядра на момент написания этих строк — 2.6.32. Инструкция по установке Томою на дистрибутив Linux с более старым ядром находится по адресу: <http://tomoyo.sourceforge.jp/1.6/index.html.en>.

Поддержка Томою уже включена в состав ядра, но ее нужно только активировать. А поэтому вам придется перекомпилировать ядро. Посетите сайт www.kernel.org и скачайте последнюю версию ядра, например

<http://www.kernel.org/pub/linux/kernel/v2.6/linux-2.6.31.5.tar.bz2>

В моем дистрибутиве использовалась версия ядра 2.6.31, поэтому чтобы не было каких-либо осложнений, я скачал не версии 2.6.32, а версии 2.6.31.5. Для компиляции ядра нужно установить пакеты `gcc`, `make` и `ncurses` (остальные пакеты будут установлены автоматически).

Распакуйте архив с ядром в `/usr/src` и введите команду:

```
$ make -s menuconfig
```

Включите параметры ядра **Enable different security models** и **TOMOYO Linux Support**. После этого сохраните конфигурацию ядра и введите команды:

```
$ make -s
```

```
$ su
```

```
$ make -s modules_install install
```

После установки ядра с поддержкой Томою нужно скачать и откомпилировать утилиты, необходимые для работы с этим модулем:

```
# wget http://osdn.dl.sourceforge.jp/tomoyo/41908/tomoyo-tools-2.2.0-20090727.tar.gz
# tar -zxvf tomoyo-tools-2.2.0-20090727.tar.gz
# make -C tomoyo-tools/ install
```

48.3. Инициализация системы

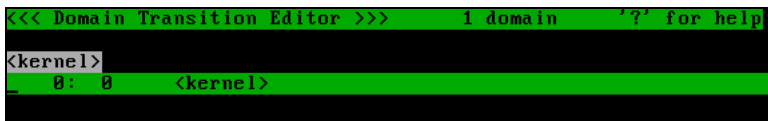
После установки нового ядра и утилит Томою систему нужно перезагрузить. Следующий шаг — инициализация политик Томою, для чего следует ввести команду:

```
# /usr/lib/tomoyo/tomoyo_init_policy
```

В зависимости от производительности вашего компьютера инициализация может занять несколько минут. Как только инициализация будет завершена, можно запускать редактор политик:

```
# /usr/sbin/tomoyo-editpolicy /etc/tomoyo/
```

Поскольку вы еще не создавали никаких политик, у вас будет только один домен — `kernel` (рис. 48.1). Когда вы все настроите, доменов будет существенно больше. Можете проверить это, запустив редактор политик, предварительно загрузившись с LiveCD.



```
<<< Domain Transition Editor >>> 1 domain '?' for help
<kernel>
_ 0: 0 <kernel>
```

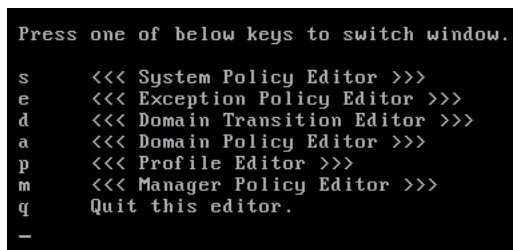
Рис. 48.1. Редактор политик

Процесс может принадлежать только одному домену, но он может во время своего выполнения переходить к другому домену. Процесс не может одновременно принадлежать двум и больше доменам. Ядро принадлежит домену `<kernel>`, система инициализации `init` — домену `<kernel>` `/sbin/init`, поскольку она была запущена ядром, а процесс, запущенный `init`, будет находиться в домене "`<kernel>` `/sbin/init` процесс". Другими словами, домен — это история выполнения процесса. По домену можно понять, какой процесс является родительским, а какой — дочерним.

Посмотрите на второе число в строке домена (см. рис. 48.1):

```
0: 0 <kernel>
```

Второе число (в данном случае — 0) — это номер профиля (может быть от 0 до 255). Сейчас посмотрим доступные профили. Нажмите клавишу `<w>` для входа в меню редактора (рис. 48.2), а затем `<r>` для просмотра доступных профилей (рис. 48.3).



```
Press one of below keys to switch window.
s <<< System Policy Editor >>>
e <<< Exception Policy Editor >>>
d <<< Domain Transition Editor >>>
a <<< Domain Policy Editor >>>
p <<< Profile Editor >>>
m <<< Manager Policy Editor >>>
q Quit this editor.
_
```

Рис. 48.2. Меню редактора политик

Строки, содержащие слово `COMMENT`, являются просто комментариями. Существуют три профиля (режима доступа), задающие уровень MAC (Mandatory Access Control):

♦ 0 (disabled) — контроль доступа к файлам отключен;

- ❖ 1 (learning) — обучающий режим, все выполненные операции заносятся в политику как разрешенные;
- ❖ 2 (permissive) — разрешающий режим, в этом режиме даже если операция запрещена, она выполняется, но не заносится в политику (полезен для отладки);
- ❖ 3 (enforcing) — режим ограничения доступа, если операция запрещена, то она не выполняется, а сообщение о нарушении доступа заносится в журнал.

```
<<< Profile Editor >>>      16 entries      '?' for help
0: 0-COMMENT=-----Disabled Mode-----
1: 0-MAC_FOR_FILE=disabled
2: 0-MAX_ACCEPT_ENTRY=2048
3: 0-TOMOYO_VERBOSE=disabled
4: 1-COMMENT=-----Learning Mode-----
5: 1-MAC_FOR_FILE=learning
6: 1-MAX_ACCEPT_ENTRY=2048
7: 1-TOMOYO_VERBOSE=disabled
8: 2-COMMENT=-----Permissive Mode-----
9: 2-MAC_FOR_FILE=permissive
10: 2-MAX_ACCEPT_ENTRY=2048
11: 2-TOMOYO_VERBOSE=enabled
12: 3-COMMENT=-----Enforcing Mode-----
13: 3-MAC_FOR_FILE=enforcing
14: 3-MAX_ACCEPT_ENTRY=2048
15: 3-TOMOYO_VERBOSE=enabled
```

Рис. 48.3. Доступные профили

Параметр `MAC_FOR_FILE` регулирует принудительный контроль доступа (Mandatory Access Control) к файлам. Параметр `MAX_ACCEPT_ENTRY` используется для ограничения максимального количества записей в списке доступа. Записи добавляются автоматически в обучающем режиме. По умолчанию используется значение 2048. Параметр `TOMOYO_VERBOSE` протоколирует случаи нарушения доступа с помощью syslog.

Самый главный параметр — `MAC_FOR_FILE`, именно им и отличаются режимы контроля доступа.

В политику Томоюо по умолчанию добавлены исключения, которые необходимы для нормальной работы системы. Для просмотра исключений нажмите клавишу `<e>` (рис. 48.4).

Для выхода из редактора политик нажмите клавишу `<q>`. Сейчас попробуем настроить Томоюо в автоматическом обучающем режиме. Попробуем создать политику для DNS-сервера. Запустите его:

```
# service named start
```

Потом запустите редактор политик. Перейдите к процессу `named`, используя клавиши `<↑>` и `<↓>`. Для изменения профиля `named` нажмите клавишу `<s>`, а затем введите 1, что соответствует номеру профиля `MAC_FOR_FILE`. Строка, относящаяся к `named`, теперь будет выглядеть так:

```
число: 1      *      /usr/sbin/named
```

<< Exception Policy Editor >>> 939 entries '?' for help		
0: alias	/bin/bash	/bin/sh
1: alias	/bin/ed	/bin/red
2: alias	/bin/gawk	/bin/awk
3: alias	/bin/gawk	/usr/bin/awk
4: alias	/bin/grep	/bin/egrep
5: alias	/bin/grep	/bin/fgrep
6: alias	/bin/hostname	/bin/dnsdomainname
7: alias	/bin/hostname	/bin/domainname
8: alias	/bin/hostname	/bin/nisdomainname
9: alias	/bin/hostname	/bin/ypdomainname
10: alias	/bin/mail	/bin/mailx
11: alias	/bin/mail	/usr/bin/Mail
12: alias	/bin/tar	/bin/gtar
13: alias	/bin/tcsh	/bin/csh
14: alias	/bin/traceroute	/bin/tcptraceroute
15: alias	/bin/traceroute	/bin/traceroute6
16: alias	/bin/traceroute	/bin/tracert
17: alias	/bin/vi	/bin/ex
18: alias	/bin/vi	/bin/rvi
19: alias	/bin/vi	/bin/rview
20: alias	/bin/vi	/bin/view
21: alias	/etc/sysconfig/network-scripts/ifdown-ipp	/etc/sysconf

Рис. 48.4. Исключения

Значение 1 соответствует обучающему режиму (Learning Mode). В обучающем режиме нужно определить, какие файлы использует DNS-сервер при запуске, в процессе работы и при завершении работы. Поэтому DNS-сервер нужно перезапустить:

```
# service named restart
```

Снова запустите редактор политик и перейдите к процессу named. Нажмите клавишу <Enter>, чтобы просмотреть, какие разрешения предоставила система DNS-серверу во время перезапуска. После этого выйдите из редактора и сохраните созданную политику:

```
# /usr/sbin/tomoyo-savepolicy
```

Для загрузки политики используется команда:

```
# /usr/sbin/tomoyo-loadpolicy af
```

При сохранении политики в каталоге /etc/tomoyo создаются два файла: exception_policy.conf и domain_policy.conf. Первый — это политика исключений, а второй — политика домена. Параметр *a* при загрузке политики указывает, что загрузить нужно оба файла, а параметр *f* — присоединяет загружаемую политику к той, что сейчас находится в ядре. Если параметр *f* не указывать, то политика, имеющаяся в ядре, будет перезаписана загружаемой политикой.

Теперь перейдем в разрешающий режим. Запустите редактор политики и установите для процесса named профиль 2. Действия DNS-сервера запрещаться не будут, но мы получим сообщение о нарушении доступа, что позволит выяснить, какие еще файлы нужны DNS-серверу. Когда все будет настроено, нужно выбрать профиль 3.

После того как создадите политику для DNS-сервера, можно приступить к созданию политики для других процессов. Помните, что некоторые процессы могут запускать другие процессы. Например, Web-сервер может запускать sendmail для отправки писем и perl для запуска Perl-сценариев. Поэтому когда вы исследуете процесс, смотрите, какие процессы он запускает. Если вы для родительского процесса установили какой-то профиль, то этот же профиль нужно установить и для всех дочерних процессов.

Создание политик Томоюо — дело кропотливое, хотя и не очень сложное. Дополнительную информацию можно получить по следующим адресам:

<http://tomoyo.sourceforge.jp/2.2/tuning.html.en>

<http://tomoyo.sourceforge.jp/2.2/enforcing.html.en>

ГЛАВА 49



Защита точки доступа

В последнее время все большее распространение получают беспроводные сети. Причины, думаю, понятны. Вам не нужно прокладывать кабель отдельно к каждой рабочей станции. Достаточно установить одну точку беспроводного доступа и обеспечить, чтобы все компьютеры были в зоне действия этой точки доступа. А иногда кабельное соединение вообще невозможно организовать по тем или иным причинам. Тогда покупается комплект оборудования Radio Ethernet и обеспечивается беспроводное подключение компьютера к сети или беспроводное соединение двух и более подсетей. Однако беспроводные технологии при неправильной настройке очень уязвимы.

Есть три основных причины, ради которых нужно уделить внимание защите вашей беспроводной сети.

- ◆ Некто может использовать ваши сетевые ресурсы. Даже если этот некто не стащит вашу базу 1С, а просто скопирует музыку с вашего компьютера, вам будет приятно?
- ◆ Ваш канал может быть использован для незаконной деятельности, и вы потом будете долго доказывать, что банк взломали не вы.
- ◆ Ваш трафик может быть перехвачен (а это, прежде всего, пароли доступа к интернет-ресурсам), что нежелательно.

В этой главе мы рассмотрим десять шагов к безопасной беспроводной сети.

49.1. Изменение параметров по умолчанию

Нужно обязательно изменить пароль администратора и идентификатор SSID, задающий имя сети. Пароль администратора и идентификатор SSID в большинстве случаев свободно доступны в Интернете для большинства моделей беспроводного оборудования. А это означает, что злоумышленнику достаточно знать модель вашей беспроводной точки доступа, чтобы получить к ней доступ, предварительно узнав стандартный SSID и пароль администратора.

При изменении SSID помните, что новый SSID не должен содержать название вашей компании, адрес, вашу фамилию, номер телефона и прочую общедоступную информацию. Отнеситесь к SSID, как к паролю: используйте символы разного регистра и цифры.

49.2. Отключение широковещания SSID

Многие точки доступа по умолчанию транслируют всем свой SSID. Поэтому к вашей точке доступа может подключиться любой нежелательный гость, даже не преднамеренно: человек просто запустит поиск сети и найдет вашу точку доступа (рис. 49.1).

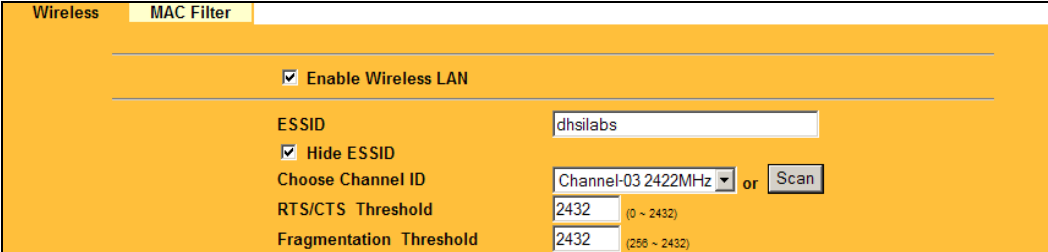
The image shows a web-based configuration interface for a wireless network. At the top, there are two tabs: 'Wireless' and 'MAC Filter', with 'Wireless' being the active tab. Below the tabs, there is a section for 'Enable Wireless LAN' with a checked checkbox. Underneath, the 'ESSID' is set to 'dhsilabs'. There is a checkbox for 'Hide ESSID' which is also checked. Below that, 'Choose Channel ID' is set to 'Channel-03 2422MHz' with a dropdown arrow, followed by the word 'or' and a 'Scan' button. The 'RTS/CTS Threshold' is set to '2432' with a range '(0 ~ 2432)' in parentheses. The 'Fragmentation Threshold' is also set to '2432' with a range '(256 ~ 2432)' in parentheses.

Рис. 49.1. Изменение SSID и отключение широковещания SSID (Hide ESSID)

49.3. Используйте WPA

Протоколы WPA (Wi-Fi Protected Access), WPA2 и WEP (Wired Equivalent Privacy) обеспечивают защиту и шифрование данных, передаваемых беспроводной точкой доступа и беспроводным клиентом. Предпочтительнее использовать WPA2, но если этот протокол не поддерживается, то нужно использовать WPA. Взломать защиту WEP можно с помощью ряда стандартных инструментов, т. е. взлом WEP — это обычная процедура. WEP заметно хуже, на фоне WPA, но это лучше, чем вообще ничего.

ПРИМЕЧАНИЕ

По адресу <http://www.thg.ru/network/20050806/index.html> вы найдете пошаговую инструкцию взлома протокола WEP.

Протокол WPA пришел на смену WEP. Для управления ключом и шифрования в WPA используется несколько алгоритмов, в их числе TKIP (Temporal Key Integrity Protocol) и AES (Advanced Encryption Standard).

Для использования WPA необходимо, чтобы все клиенты были совместимы с этим протоколом (не говоря уже о точке доступа). Все современные точки доступа поддерживают WPA.

WPA и WEP используются для шифрования данных, которые передаются между точкой доступа и беспроводным клиентом. Для шифрования данных используется специальный ключ. Завладев ключом, злоумышленник сможет не только установить соединение с беспроводной точкой доступа, но и расшифровать данные, передающиеся между клиентами беспроводной точки доступа.

Если используется протокол WEP, то ключ нужно вводить вручную. Это существенный недостаток, поскольку пользователи вводят ключ всего лишь раз, а затем им лень менять его еще раз. Протокол WPA периодически сам меняет ключ, причем делает он это автоматически. Даже если злоумышленник каким-нибудь образом узнает ключ, то он будет действовать только до момента изменения ключа беспроводной точкой доступа. Во многих точках доступа ключи меняются один раз в час (рис. 49.2).

WEP Encryption

Authentication Method

64-bit WEP: Enter 5 ASCII characters or 10 hexadecimal characters.
128-bit WEP: Enter 13 ASCII characters or 26 hexadecimal characters.
(Select one WEP key as an active key to encrypt wireless data transmission.)

128-bit WEP (selected)
Disable
64-bit WEP
128-bit WEP

☒ ASCII ☐ Hex

☐ Key 1: den2432exd
☒ Key 2: den2432exd43
☐ Key 3: den2432exd
☐ Key 4: den2432exd

Рис. 49.2. Старая точка доступа: поддерживается только WEP

49.4. Фильтрация MAC-адресов

Вы можете указать список MAC-адресов адаптеров компьютеров, которые смогут получить доступ к вашей точке доступа. Нужно отметить, что фильтрация MAC-адресов не обеспечивает надежной защиты, а служит просто дополнительным барьером. Опытный злоумышленник всегда сможет перехватить MAC-адреса и подменить свой адрес одним из разрешенных адресов. Зато фильтрация MAC-адресов эффективно срабатывает против дилетантов. Это как сигнализация в автомобиле: какая бы она ни была хорошая, опытный злоумышленник обойдет ее, а вот дилетанты и близко к машине не подойдут (рис. 49.3).

MAC Address Control					
Item	Setting				
MAC Address Control	<input checked="" type="checkbox"/> Enable				
<input type="checkbox"/> Connection control	Wireless and wired clients with C checked can connect to this device; and <input type="text" value="allow"/> unspecified MAC addresses to connect.				
<input type="checkbox"/> Association control	Wireless clients with A checked can associate to the wireless LAN; and <input type="text" value="deny"/> unspecified MAC addresses to associate.				
ID	MAC Address		IP Address	C	A
1	<input type="text"/>		192.168.1. <input type="text"/>	<input type="checkbox"/>	<input type="checkbox"/>
2	<input type="text"/>		192.168.1. <input type="text"/>	<input type="checkbox"/>	<input type="checkbox"/>
3	<input type="text"/>		192.168.1. <input type="text"/>	<input type="checkbox"/>	<input type="checkbox"/>
4	<input type="text"/>		192.168.1. <input type="text"/>	<input type="checkbox"/>	<input type="checkbox"/>
DHCP clients <input type="text" value="-- select one --"/> <input type="button" value="Copy to"/> ID <input type="text" value="--"/>					
<input type="button" value="Previous page"/> <input type="button" value="Next page"/> <input type="button" value="Save"/> <input type="button" value="Undo"/> <input type="button" value="Help"/>					

Рис. 49.3. Фильтрация по MAC-адресам

49.5. Обновление прошивки оборудования

Итак, все современные версии точек доступа поддерживают WPA. Устаревшие версии поддерживают только WEP. Иногда удастся с помощью обновления прошивки точки доступа добавить поддержку WPA. Но это удастся не всегда: далеко не все производители точек доступа выпускают прошивки для устаревших моделей точек доступа.

Но даже если у вас самая современная точка доступа, все равно рекомендуется зайти на сайт производителя: вдруг есть свежая версия прошивки? Дело в том, что в текущей версии могут быть найдены ошибки, которые будут устранены в новой версии прошивки. Также могут быть добавлены новые методы шифрования. Одним словом, обновление прошивки — дело полезное.

49.6. Использование аутентификации

Протоколы WPA и WPA2 значительно лучше, чем WEP, но они все же уязвимы. В Интернете можно найти пошаговые инструкции взлома протокола WPA и WEP. Да, взлом WPA довольно сложный, но все же возможен.

ПРИМЕЧАНИЕ

О протоколах EAP, WPA и WPA2 можно прочитать по адресу <http://blogs.zdnet.com/Ou/?p=67>. Протокол WPA и защита сети с его помощью подробно рассмотрена по адресу <http://www.ixbt.com/comm/prac-wpa-eap.shtml>.

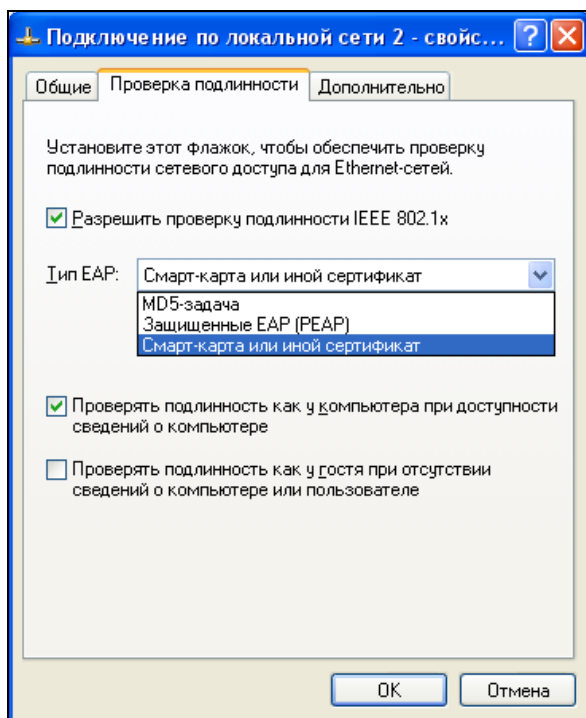


Рис. 49.4. Настройка аутентификации в Windows XP

Что же делать? Единственный выход — это использование аутентификации. Аутентификация требует от клиента регистрации в сети. Аутентификация может производиться с помощью сертификатов или с помощью паролей, которые проверяются на сервере аутентификации. Протоколы WEP, WPA и WPA2 поддерживают несколько типов аутентификации EAP (Extensible Authentication Protocol).

Настройка сервера аутентификации может быть довольно сложной задачей. Кто хотя бы раз настраивал RADIUS (Remote Authentication Dial-In User Service), тот знает, о чем я говорю. Если настраивать RADIUS не хочется, тогда можно воспользоваться альтернативными серверами доступа, например WSC Guard (<http://www.wirelesssecuritycorp.com/wsc/public/WirelessGuard.do>). Но данное решение не бесплатно. При желании в Интернете можно найти и бесплатные серверы доступа, более простые в настройке, чем RADIUS. Некоторые из них не всегда подходят для организаций, поскольку позволяют аутентифицировать небольшое количество пользователей (рис. 49.4).

49.7. Понижение мощности передачи

Некоторые точки доступа позволяют понизить мощность передачи, что позволит снизить число как преднамеренных несанкционированных подключений к точ-

ке доступа, так и случайных подключений. Понизив мощность передачи, можно добиться того, что ваша точка доступа будет доступна только в пределах офиса вашей компании. В любом случае использование мощной направленной антенны, позволяющей обнаружить даже самый слабый сигнал, сводит на нет все ваши старания, но, во всяком случае, вы оградите себя от случайных подключений к вашей точке доступа.

49.8. Отключение точки доступа, когда вы не работаете

Вы работаете ночью? Нет? Тогда выключайте точку доступа, когда вы не работаете. Можно настроить автоматическое выключение, а можно выключать все самому — так вы на 100% будете уверены, что никто не проникнет в вашу сеть.

49.9. Защита портов управления

Интерфейсы управления беспроводной сети не должны быть доступны по беспроводной сети. Желательно, чтобы управление беспроводной сетью осуществлялось только по внутренней (кабельной) сети. Также нужно обеспечить доступ к портам управления только конкретным станциям (1—2 станции): зачем разрешать доступ к портам управления всем компьютерам сети?

49.10. Защита от внешних угроз. Общая защита сети

Помните, что антивирусы и брандмауэры никто не отменял даже в случае с беспроводной связью. Желательно установить не один общий брандмауэр/антивирус — на сервере, но и установить клиентские брандмауэры и антивирусы для защиты каждого компьютера сети в отдельности. Но только после установки не забудьте настроить их должным образом, а то толку от них не будет.

В этой главе мы рассмотрели основные десять этапов защиты беспроводной сети. Подробнее о защите беспроводной сети вы можете прочитать по адресу <http://www.terralab.ru/networks/269553/>.

ГЛАВА 50



Защита маршрутизатора

Откройте любое пособие по сетевой безопасности. В большинстве случаев огромное внимание уделяется защите брандмауэров, но практически ничего не сказано о защите маршрутизатора, а ведь именно это устройство обеспечивает маршрутизацию пакетов. Нарушение работы этого устройства может парализовать работу всей сети. Но почему-то о маршрутизаторе забывают, рассматривая защиту отдельных служб сервера и конфигурирование брандмауэра. В этой главе мы поговорим о защите обычного управляемого маршрутизатора. По возможности, я не буду привязываться к конкретному бренду и конкретной модели, но некоторые пояснения буду делать на примере CISCO IOS, поскольку, что ни говори, маршрутизаторы CISCO являются одними из самых распространенных на наших просторах.

50.1. О маршрутизаторе

Маршрутизатор обеспечивает маршрутизацию пакетов, это его основная задача. Но современные роутеры не совсем похожи на первые маршрутизаторы. Да, маршрутизацию пакетов никто не отменял, но у многих современных маршрутизаторов есть функции для организации фильтрации пакетов (т. е. маршрутизатор играет роль брандмауэра), для организации VPN-шлюзов и много других полезных (и не очень) функций. По сути, современный маршрутизатор — это специальный компьютер, управляемый собственной операционной системой. Многие "встроенные" операционные системы отчасти похожи на UNIX. Но почему-то администрирование и настройку UNIX-серверов рассматривают все, а о роутерах забывают. Сегодня мы попробуем применить принципы защиты UNIX-сервера к обычному маршрутизатору. Что из этого получится — судить вам.

Настройку маршрутизатора будем рассматривать на примере управляемого маршрутизатора среднего класса. Мы не будем рассматривать сверхдорогие маршрутизаторы, оснащенные специальными функциями защиты от атак. Например, есть специальная версия IOS — Firewall Feature Set, в которой реализованы функции IDS (Intrusion Detection Systems), системы обнаружения вторжений, но мы ее

рассматривать не будем. Если вам нужны функции IDS, вы можете купить такой маршрутизатор и настроить его так, как описано в документации.

При настройке маршрутизатора мы сконцентрируем свое внимание на защите самого маршрутизатора, а не на защите узлов, которые находятся за ним.

50.2. Установка пароля

Любой управляемый маршрутизатор позволяет установить пароль для доступа к консоли управления. По умолчанию пароль или вообще отсутствует, или же используется стандартный пароль, который стандартно и "взламывается". Точнее, тут и взламывать нечего — злоумышленнику просто нужно знать модель вашего маршрутизатора, а пароль можно найти в Интернете, после этого доступ к маршрутизатору ему гарантирован. Поэтому не забываем установить хороший пароль. Напомню, что пароль не должен содержать личных данных, а также общедоступных данных о компании. Хороший пароль должен состоять не менее, чем из восьми символов разного регистра, желательно также использовать не только буквы, но и цифры.

Некоторые маршрутизаторы позволяют установить пароль длиной до 80 символов. Не стоит пользоваться этой возможностью. Такой пароль вы никогда не запомните, поэтому вы его где-нибудь сохраните, следовательно, теоретически можно его найти на вашем компьютере.

50.3. Ограничение доступа по сети

Получить доступ к консоли управляемого маршрутизатора можно локально, подключившись к AUX, или же по сети. Обычно используется протокол telnet или ssh. В случае сетевого доступа нужно ограничить узлы, с которых разрешен доступ к консоли управления маршрутизатором. Все управляемые маршрутизаторы позволяют сделать это.

Если есть выбор между telnet и ssh, то лучше выбрать ssh, поскольку telnet передает информацию по сети в открытом виде (в том числе и пароли), что нежелательно.

50.4. Только локальный доступ

Наверное, самый надежный способ ограничения доступа — это запрещение сетевого доступа. В этом случае доступ к маршрутизатору возможен только локально, с помощью AUX. То есть злоумышленнику, чтобы изменить параметры маршрутизатора, придется физически подойти к маршрутизатору и подключиться к AUX. А это сделать по ряду причин невозможно. Если же маршрутизатор разме-

щен в отдаленной части помещения, целесообразно установить скрытую видеокамеру, чтобы знать, кто и когда получал доступ к маршрутизатору.

50.5. Защита SNMP

SNMP (Simple Network Management Protocol), простой протокол управления сетью, очень часто используется для управления маршрутизаторами. Из соображений безопасности лучше вообще отключить SNMP, но если SNMP вам нужен, убедитесь, что используется более защищенная третья версия (SNMPv3), в более ранних версиях (SNMPv1, SNMPv2) авторизация и защита данных не предусмотрены.

Если SNMP все же нужен, то нужно принять минимальные меры по обеспечению безопасности, а именно:

- ✧ придумайте трудно подбираемое имя community;
- ✧ MIB (Management Information Base) должна работать в режиме "только чтение";
- ✧ ограничьте SNMP-доступ несколькими узлами (желательно одним — вашим).

50.6. Ведение журналов

Наверняка в вашей компании найдется хотя бы одна машина под управлением UNIX (Linux). Так вот в составе UNIX есть демон протоколирования — syslogd, который можно настроить для протоколирования событий маршрутизатора. Протоколировать нужно не все события, а только те, которые затрагивают сетевую безопасность, например попытки неудачной авторизации по ACL (спискам доступа).

Для более надежного протоколирования рекомендуется запустить syslogd на нескольких UNIX-машинах и настроить их на протоколирование событий маршрутизатора. Syslogd использует протокол UDP (а не TCP), который не гарантирует доставку пакетов, поэтому и нужно запускать несколько демонов протоколирования (если один не запροколирует, то у второго точно все получится).

Также желательно использовать протокол NTP (Network Time Protocol) для синхронизации времени, что существенно помогает при анализе протоколов (чтобы не было разницы во времени).

50.7. Отключение ненужных сервисов

Отключите все сервисы, которые вы не используете (например, finger, BOOTP, ARP Proxy). Ведь каждый такой сервис может стать "дырой" в системе безопасности нашего маршрутизатора.

50.8. Ограничение ICMP

Некоторые DoS-атаки используют данный протокол в качестве основного инструмента атаки, поэтому желательно ограничить использование ICMP, разрешив только пакеты определенных типов. В первую очередь, нужно ограничить пакеты PMTU (Path MTU discovery), пакеты с сообщением "packet-too-big". Что делать с другими типами ICMP-сообщений, зависит от вашей политики безопасности.

50.9. Отключение потенциально опасных опций

К потенциально опасным опциям относятся IP source route и IP unreachable. В первом случае злоумышленник может определить путь, по которому будет передаваться пакет. После этого он может послать пакет source routed на "узел-жертву", который находится за маршрутизатором, в результате чего он изменит маршрутизацию атакуемой сети.

Во втором случае (IP unreachable), если пакет отброшен в соответствии со списком доступа (ACL), злоумышленник получит ICMP-пакет (тип 3, код 13), на основании чего он сможет сделать вывод, что маршрутизатор защищен с помощью ACL, а это нежелательно. Чем меньше информации о нашем маршрутизаторе, тем меньше вероятность взлома. Поэтому опцию IP unreachable нужно отключить.

Для отключений опций IP source route и IP unreachable на маршрутизаторах Cisco нужно ввести IOS-команды:

```
no ip source-route  
no ip unreachable
```

50.10. Анти-spoofing и защита от DoS-атак

IP-spoofing — это неавторизованный доступ к компьютеру (серверу) путем подделки IP-адреса. Например, вы разрешили доступ к своему маршрутизатору узлу с определенным адресом, а злоумышленник может подделать этот адрес с целью получения доступа. Для защиты от этого нужно использовать *антиспуфинг*, основная идея которого заключается в том, что никто из внешней сети не имеет право отправлять пакеты, содержащие в поле адреса источника какой-нибудь адрес из вашей подсети. Для фильтрации таких пакетов нужно использовать списки доступа, а также желательно зафиксировать попытку подделки IP-адреса в журнале.

Защита от DoS-атак заслуживает отдельного разговора. В этой главе мы не будем рассматривать защиту от DoS-атак, а поговорим только о двух самых распро-

страненных DoS-атаках. Первая атака называется SYN flood. Она происходит, когда злоумышленник отправляет на открытый порт много SYN-пакетов с недостижимым адресом источника. Атакуемый маршрутизатор должен ответить пакетом <SYN, ACK>, но ведь узел, указанный в качестве источника, недоступен, поэтому трехступенчатая схема установления TCP-соединения не завершается. Учитывая, что таких SYN-пакетов очень много, лимит на количество открытых соединений очень быстро превышает, и жертва отказывается принимать запросы на установление соединения от обычных пользователей сети.

ПРИМЕЧАНИЕ

Руководство о защите от SYN-атак маршрутизаторов Cisco можно найти по адресу: <http://cio.cisco.com/warp/public/707/4.html>.

Вторая атака (она называется Land) заключается в том, что злоумышленник посылает пакет с одинаковыми портами и IP-адресами источника и получателя. Такие пакеты вызывают исключения во многих маршрутизаторах.

ПРИМЕЧАНИЕ

Информацию о Land-атаке можно найти по адресу: <http://www.cisco.com/warp/public/770/land-pub.shtml>.

50.11. Отключение CDP

CDP (Cisco Discovery Protocol) — протокол, работающий на всем оборудовании Cisco, используется для управления сетями. Протокол CDP позволяет оповестить другие устройства Cisco о присутствии в сети того или иного устройства от Cisco, другими словами, устройства Cisco с помощью этого протокола находят друг друга.

Используя CDP, можно получить информацию об устройстве, его конфигурации, низкоуровневых протоколах, а также информацию о соседних маршрутизаторах. Помните основной принцип защиты любой информационной системы: минимум информации. Чем больше информации вы предоставите злоумышленнику о своей сети, тем быстрее он ее взломает. Поэтому CDP нужно отключить. Для этого используется следующая IOS-команда:

```
no cdp run
```

Мы рассмотрели основные этапы защиты маршрутизатора. Вам остается лишь настроить ваш маршрутизатор, следуя рекомендациям, приведенным в этой главе.

ГЛАВА 51



Средства резервного копирования. Создание ISO-диска

51.1. Необходимость в "живой" резервной копии

Все мы помним "привидение от Нортона" — Norton Ghost®. В мире Windows — это незаменимый продукт. В этой главе мы поговорим о средствах резервного копирования для Linux. Но эти средства необычные: все они позволяют создать не просто резервную копию системы, а LiveCD/DVD. Да, эти средства способны заменить Norton Ghost, причем абсолютно бесплатны в отличие от продукта Symantec.

Для начала определимся, зачем нужны средства для создания LiveCD. Наша цель — резервное копирование системы, но причем здесь LiveCD? Оказывается, это довольно удобно. Мы убиваем вот столько зайцев сразу.

- ♦ Создаем средство для восстановления системы — предположим, что вы настроили свою систему, "подняли" все сетевые службы, отредактировали их конфигурационные файлы. Но завтра из-за очередного перепада напряжения сторел жесткий диск. Опять все заново настраивать? Если вы накануне создали LiveCD, то вам нечего беспокоиться. Заменяли жесткий диск, загрузились с LiveCD (смотря правде в глаза, учитывая размер системы, у нас будет LiveDVD, но по старинке мы здесь и далее будем называть его LiveCD) и установили систему вместе со всеми параметрами на новый винчестер. И все! На всю эту операцию будет потрачено полчаса, от силы минут 40 вместе с установкой нового жесткого диска. Пользователи и начальство будут вам благодарны за столь оперативное "воскрешение" сервера. А теперь представьте, что вы создали обычный "бэкап" с помощью tar/tgz. Вам нужно минимум 40 минут на установку системы, потом время на восстановление резервной копии, плюс одна лишняя перезагрузка. Однозначно времени будет потрачено больше.
- ♦ Создаем средство для клонирования системы — когда предприятие покупает компьютерный парк, то, как правило, все компьютеры однотипные (исключение составляют разве что серверы — они должны быть мощнее — и компьюте-

ры начальства — у них должна быть мощная видеокарта). Вот теперь представьте, что вам нужно настроить каждый новый компьютер. А их может быть 10, 20, 50! Можно поступить проще. Настроить один компьютер, создать LiveCD и развернуть его на всех остальных компьютерах сети. Пусть настройка одного компьютера займет полтора часа (установка системы и ее настройка), создание LiveCD — еще минут 30 (тут все зависит от производительности компьютера, потому что от вас требуется ввод всего одной команды), затем запись образа на болванки. Да, именно на "болванки", потому что вам нужно будет создать несколько копий LiveCD, чтобы вы могли одновременно устанавливать систему на несколько компьютеров. Затем еще минут 40 ожидания и будет настроено N компьютеров сразу, число N зависит от количества имеющихся болванок. Удобно? Думаю да. Без LiveCD вы бы потратили полтора часа на каждый компьютер. 10 компьютеров — это 15 часов (2 рабочих дня). А так будет потрачено примерно 4 часа. Созданные "клоны" системы можно использовать в будущем, если компьютерный парк будет расширяться.

- ◆ Возможность создания LiveUSB — загрузочная живая флэшка понадобится для восстановления/клонирования ОС нетбука и других компьютеров, где нет привода DVD. Средства создания LiveCD позволяют также создать и загрузочную флэшку.

Не нужно думать, что бэкап в виде LiveCD может использоваться только для копирования/восстановления файлов самой системы. Можно копировать и пользовательские данные из /home, лишь бы их размер не превысил размера DVD-диска. Хотя можно использовать двухслойные диски (двухсторонние использовать не удобно), что позволит увеличить объем резервируемой информации.

51.2. Какие средства мы будем рассматривать

Самым мощным средством для клонирования Linux является Clonezilla. Этот продукт может не только создать LiveCD, но и развернуть систему по сети. На сайте разработчиков <http://clonezilla.org/> можно найти следующую информацию: за 10 минут Clonezilla SE (SE — Server Edition) развернул по сети образ 5,6 Гбайт на 41 компьютер сети. В итоге все компьютеры были настроены всего за 10 минут. Правда, для такой сетевой установки нужно развернуть специальный сервер, но об этом позже. Кроме того, Clonezilla может использоваться для создания резервных копий компьютеров, работающих под управлением Windows и FreeBSD.

Если вам не нужно такое мощное средство, можно ограничиться утилитой Remastersys Backup (<http://www.geekconnection.org/remastersys/>). Правда, эта утилита рассчитана только на Debian и Ubuntu (а также на другие дистрибутивы, основанные на Debian), поэтому она не подойдет вам, если вы используете, скажем, Fedora или Mandriva.

Любителям Slackware подойдет скрипт Linux Live (<http://www.linux-live.org/>). Этот скрипт позволяет создать как LiveCD, так и LiveUSB.

Подобные утилиты можно найти и для других дистрибутивов, например утилита `mklivecd` (подобна Remastersys Backup) используется для создания LiveCD на базе Mandriva. Рассматривать все подобные утилиты не вижу смысла — мы обсудим одно универсальное средство и два дистрибутиво-ориентированных.

51.3. Clonezilla

Рассмотрим основные особенности Clonezilla:

- ❖ полностью бесплатна (распространяется по лицензии GPL);
- ❖ поддерживает файловые системы ext2, ext3, ext4, reiserfs, reiser4, xfs, jfs, FAT, NTFS, HFS (MacOS), UFS (FreeBSD, NetBSD, OpenBSD), VMFS (VMWare ESX), поэтому вы можете клонировать не только Linux, но и MS Windows, MacOS (Intel), FreeBSD, NetBSD и OpenBSD;
- ❖ поддержка LVM2 (LVM ver 1 не поддерживает);
- ❖ поддержка GRUB версий 1 и 2;
- ❖ версия Clonezilla SE (Server Edition) поддерживает Multicast для массового клонирования по сети при условии, что компьютеры поддерживают PXE и Wake-on-LAN;



Рис. 51.1. Загрузочное меню Clonezilla Live

- ♦ может сохранить не только отдельно взятый раздел, но и весь жесткий диск со всеми разделами.

Clonezilla — программа не простая. Сейчас мы рассмотрим лишь один из примеров ее использования (а именно создание LiveCD и восстановление системы с его помощью), а познакомиться с остальными возможностями программы можно в документации или на сайте разработчиков.

Итак, для создания/восстановления бэкапа нужно выполнить следующие действия:

1. Скачайте с <http://clonezilla.org/download/sourceforge/> ISO-образ Clonezilla Live и запишите его на болванку.
2. Загрузитесь с болванки Clonezilla Live (рис. 51.1). Нужно выбрать команду **Clonezilla live**. Если возникнут проблемы (например, с видеокартой), можно выбрать команду **Other modes of Clonezilla live** и задать другой режим загрузки Clonezilla. Вы увидите процесс загрузки Debian — тут все как обычно, нужно просто подождать (рис. 51.2).

```
[ 2.298874] scsi 1:0:1:0: Direct-Access      ATA          VMware Virtual I 0000 PQ: 0 ANSI: 5
[ 2.340195] ata2.00: ATAPI: VMware Virtual IDE CDROM Drive, 00000001, max UDMA/33
[ 2.341583] ata2.00: configured for UDMA/33
[ 2.342129] scsi 2:0:0:0: CD-ROM             NECUMWar VMware IDE CDR10 1.00 PQ: 0 ANSI: 5
[ 2.350556] sr0: scsi3-mmc drive: 1x/1x xa/form2 cdda tray
[ 2.352065] Uniform CD-ROM driver Revision: 3.20
[ 2.358812] sd 1:0:0:0: [sda] 31457216 512-byte logical blocks: (8.58 GB/8.00 GiB)
[ 2.359612] sd 1:0:0:0: [sda] Write Protect is off
[ 2.361466] sd 1:0:0:0: [sda] Write cache: disabled, read cache: enabled, doesn't support DPO or
FUA
[ 2.362200] sda: sda1 sda2 sda3 < sda5 >
[ 2.363092] sd 1:0:1:0: [sdb] 31457280 512-byte logical blocks: (16.1 GB/15.0 GiB)
[ 2.363185] sd 1:0:1:0: [sdb] Write Protect is off
[ 2.363228] sd 1:0:1:0: [sdb] Write cache: disabled, read cache: enabled, doesn't support DPO or
FUA
[ 2.380613] sdb: sdb1
[ 2.386360] sd 1:0:1:0: [sdb] Attached SCSI disk
[ 2.387994] sd 1:0:0:0: [sda] Attached SCSI disk
[ 2.391994] sd 1:0:0:0: Attached scsi generic sg0 type 0
[ 2.393897] sd 1:0:1:0: Attached scsi generic sg1 type 0
[ 2.400551] sr 2:0:0:0: Attached scsi generic sg2 type 5
Begin: Loading essential drivers ... [ 2.593615] Atheros(R) L2 Ethernet Driver - version 2.2.3
[ 2.593830] Copyright (c) 2007 Atheros Corporation.
[ 2.612151] Broadcom NetXtreme II 5771x 10Gigabit Ethernet Driver bnx2x 1.52.1 (2009/08/12)
[ 2.632202] device-mapper: uevent: version 1.0.3
[ 2.634009] device-mapper: ioctl: 4.15.0-ioctl (2009-04-01) initialised: dm-devel@redhat.com
done.
Begin: Running /scripts/init-premount ... done.
Begin: Mounting root file system ... [ 2.745155] Uniform Multi-Platform E-IDE driver
[ 2.745836] ide_generic: please use "probe_mask=0x3f" module parameter for probing all legacy ISA
IDE ports
[ 2.882403] aufs: module is from the staging directory, the quality is unknown, you have been war
ned.
[ 2.885440] aufs 2-standalone.tree-32-20100125
[ 2.930106] loop: module loaded
[ 3.041203] squashfs: version 4.0 (2009/01/31) Phillip Lougher
```

Рис. 51.2. Процесс загрузки Debian

3. Далее нужно выбрать язык (рис. 51.3). Русского, к сожалению, пока не предвидится. Далее нужно выбрать раскладку клавиатуры (рис. 51.4), но т. к. раскладку изменять нам не нужно (а зачем?), выберите вариант **Don't touch keymap**.

4. Выберите команду **Start Clonezilla** (рис. 51.5).



Рис. 51.3. Выбор языка Clonezilla

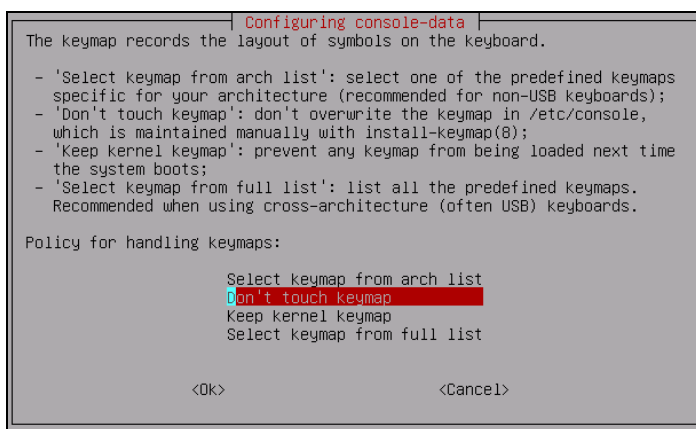


Рис. 51.4. Выбор раскладки

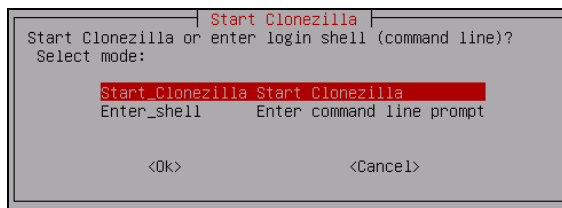


Рис. 51.5. Выберите команду **Start Clonezilla**

5. Выберите режим **device-image**: создание файла образа раздела (рис. 51.6). Режим **device-device** используется для бэкапа раздела, при этом сам бэкап будет помещен на другой раздел.
6. Далее нужно выбрать, куда будет сохранен образ или откуда он будет прочитан (в случае восстановления системы по образу), выберите **local_dev**, что означает локальное устройство (рис. 51.7). Также образ можно получить (или записать)

по SSH, NFS (Network File System, а не Need For Speed!) и из сети MS Windows (**samba_server**).

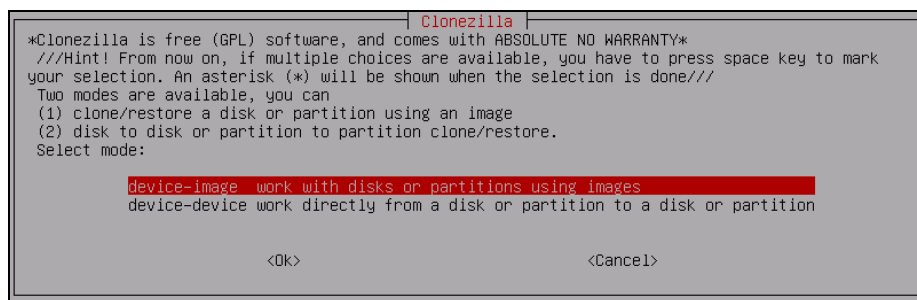


Рис. 51.6. Выберите режим **device-image**

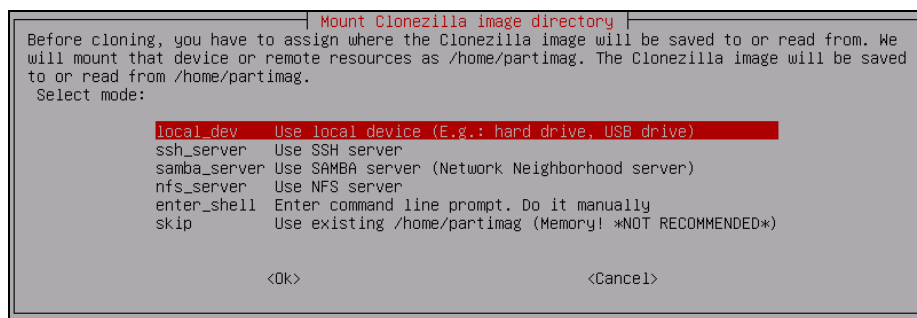


Рис. 51.7. Выбор носителя образа

7. Далее нужно выбрать раздел, где будут храниться образы. Если вы создаете образ, то на этот раздел он будет сохранен, а если восстанавливаете образ, то Clonezilla будет искать его на этом разделе.
8. Далее нужно выбрать одну из команд (рис. 51.8). Команда **savedisk** используется для сохранения всего диска, **saveparts** — для сохранения одного или нескольких разделов диска, **restoredisk** — для восстановления образа диска на локальный диск, **restoreparts** — для восстановления образа раздела, команда **recovery-iso-zip** используется для создания "живого" диска восстановления.
9. Если вы выбрали команду восстановления образа, то далее нужно выбрать образ, который нужно использовать (рис. 51.9).
10. Далее нужно ввести устройство (имена устройств соответствуют именам устройств в Linux), на которое нужно развернуть образ (рис. 51.10). Будьте внимательны, чтобы не развернуть образ раздела на весь диск — потеряете остальные разделы!
11. Если вы выбрали команду **recovery-iso-zip** (рис. 51.11) для создания LiveDVD/USB, то нужно также выбрать режим: **iso** — будет создан образ для записи на DVD; **zip** — образ для записи на LiveUSB; **both** — будут созданы оба

файла, которые можно использовать впоследствии как для создания LiveDVD, так и для создания LiveUSB. Созданный файл (файлы) будет сохранен в каталоге /home/partimag (рис. 51.12). На рис. 51.13 изображен процесс создания LiveCD, а из рис. 51.14 видно, что этот процесс удачно завершен.

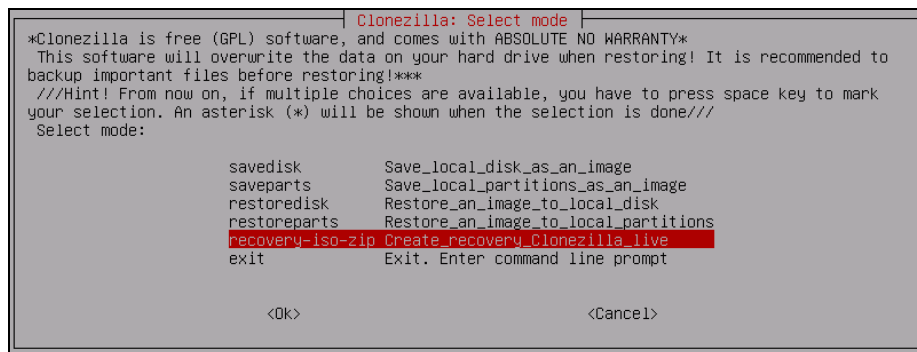


Рис. 51.8. Создать бэкап или восстановить?

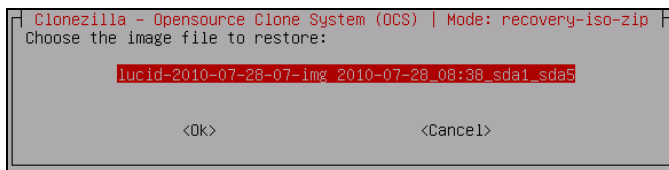


Рис. 51.9. Выбор образа для восстановления

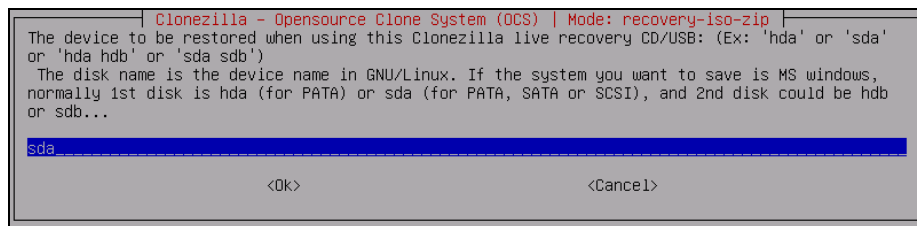


Рис. 51.10. На какое устройство "развернуть" образ

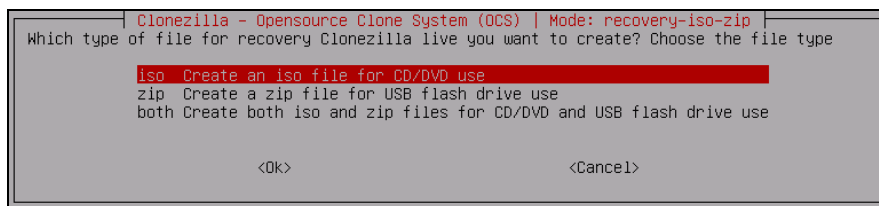


Рис. 51.11. Выбор режима команды recovery-iso-zip

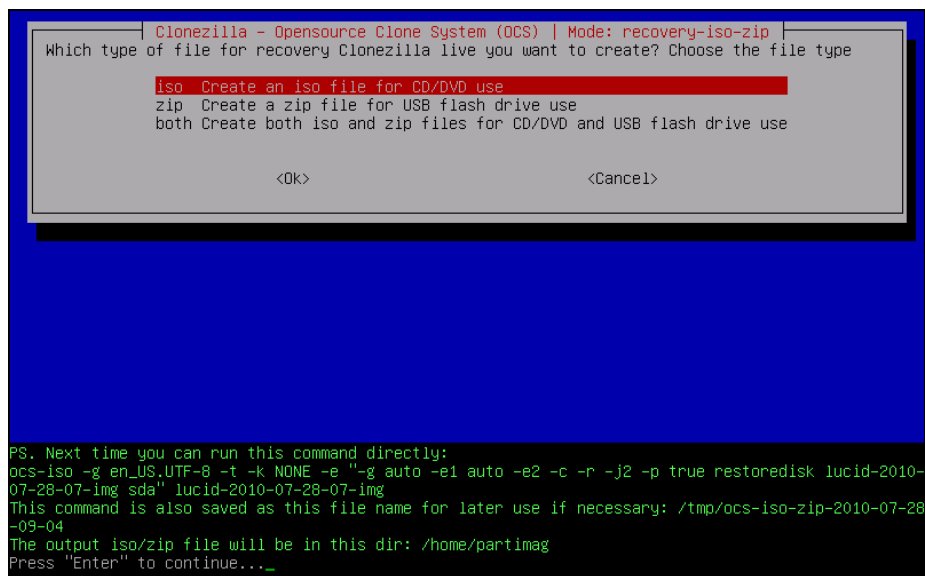


Рис. 51.12. Созданный файл будет сохранен в каталоге /home/partimag

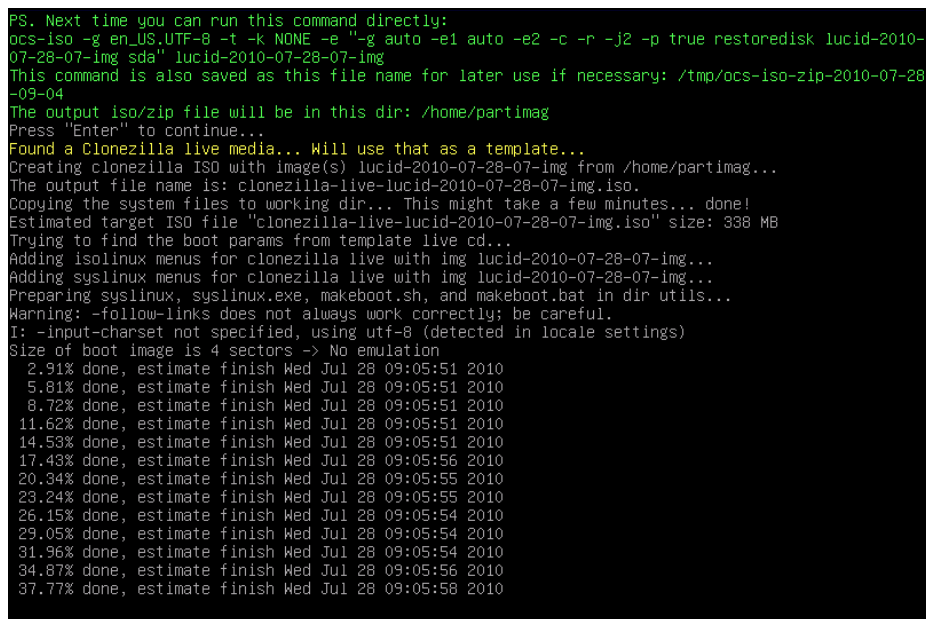


Рис. 51.13. Процесс создания LiveCD

Вот и все! Как видите, все довольно просто. Программа работает с устройствами (дисками, разделами) напрямую, поэтому при создании/восстановлении бэкапа все равно, под какой операционной системой работает компьютер.

```

55.19% done, estimate finish Wed Jul 28 09:06:36 2010
58.10% done, estimate finish Wed Jul 28 09:06:35 2010
61.00% done, estimate finish Wed Jul 28 09:06:35 2010
63.91% done, estimate finish Wed Jul 28 09:06:34 2010
66.81% done, estimate finish Wed Jul 28 09:06:34 2010
69.72% done, estimate finish Wed Jul 28 09:06:35 2010
72.62% done, estimate finish Wed Jul 28 09:06:36 2010
75.53% done, estimate finish Wed Jul 28 09:06:39 2010
78.43% done, estimate finish Wed Jul 28 09:06:47 2010
81.34% done, estimate finish Wed Jul 28 09:06:47 2010
84.24% done, estimate finish Wed Jul 28 09:06:46 2010
87.15% done, estimate finish Wed Jul 28 09:06:46 2010
90.05% done, estimate finish Wed Jul 28 09:06:45 2010
92.96% done, estimate finish Wed Jul 28 09:06:44 2010
95.86% done, estimate finish Wed Jul 28 09:06:44 2010
98.77% done, estimate finish Wed Jul 28 09:06:43 2010
Total translation table size: 2048
Total rockridge attributes bytes: 6390
Total directory bytes: 22528
Path table size(bytes): 168
Max brk space used 12000
172125 extents written (336 MB)
Cleaning tmp dirs...
Isohybridizing clonezilla-live-lucid-2010-07-28-07-img.iso... done!
You can burn this iso file onto a CD/DVD and then use it to boot other machines to use Clonezilla: c
lonezilla-live-lucid-2010-07-28-07-img.iso
*****
If you want to use Clonezilla again:
(1) Stay in this console (console 1), enter command line prompt
(2) Run command "exit" or "logout"
*****
When everything is done, remember to use 'poweroff', 'reboot' or follow the menu to do a normal powe
roff/reboot procedure. Otherwise if the boot media you are using is a writable device (such as USB f
lash drive), and it's mounted, poweroff/reboot in abnormal procedure might make it FAIL to boot next
time!
*****
Press "Enter" to continue...

```

Рис. 51.14. LiveCD создан, нажмите клавишу <Enter> для продолжения

Если у вас есть необходимость в серверной версии (Clonezilla Server Edition), прочитать руководство по ее использованию вы можете по адресу:

<http://clonezilla.org/clonezilla-server-edition/>

51.4. Remastersys Backup

В отличие от Clonezilla, которая напрямую работает с устройствами, Remastersys Backup устанавливается на компьютер, работающий под управлением Debian или Ubuntu, запускается под управлением этой операционной системы и создает ISO-образ системы, под управлением которой она запущена.

Порядок работы с Remastersys следующий: вы настраиваете свою систему, устанавливаете Remastersys, запускаете Remastersys, создаете ISO-образ, который потом нужно будет записать на болванку.

Первым делом установим Remastersys. Откройте файл sources.list:

```
sudo nano /etc/apt/sources.list
```

Добавьте в него одну из строк:

```
# Если у вас установлен GRUB v1
```

```
deb http://www.geekconnection.org/remastersys/repository ubuntu/
```

```
# Если у вас установлен GRUB2
deb http://www.geekconnection.org/remastersys/repository karmic/
```

Сохраните файл и введите две команды:

```
sudo apt-get update
sudo apt-get install remastersys
```

Формат вызова `remastersys` следующий:

```
sudo remastersys backup|clean|dist [cdfs|iso] [filename.iso]
```

Рассмотрим параметры программы:

- ❖ `backup` — создает резервную копию дистрибутива вместе с пользовательскими данными (каталог `/home`);
- ❖ `clean` — используется для очистки временных файлов, образующихся в процессе создания LiveCD, эту команду нужно вводить после каждого создания LiveCD, но только после того, как вы скопируете созданный ISO-образ в другой каталог, иначе он тоже будет удален при выполнении `remastersys clean`;
- ❖ `dist` — создает дистрибутивный образ, то же самое, что и `backup`, но не копирует пользовательские данные;
- ❖ `cdfs` — создает файл с файловой системы, но без создания ISO-образа (подойдет, если вы желаете создать ISO-образ другой программой вручную);
- ❖ `iso` — используется по умолчанию, создает ISO-образ LiveCD;
- ❖ `[filename.iso]` — имя ISO-образа, файл будет помещен в каталог `/home/remastersys`.

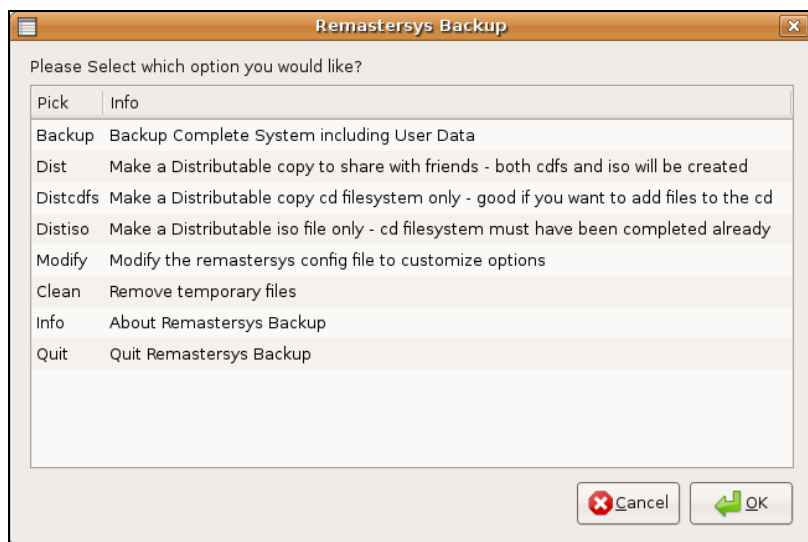


Рис. 51.15. Программа для создания LiveUSB

Я рекомендую использовать параметр `backup`: при создании образа будут сохранены и параметры пользователя. Но перед созданием образа убедитесь, что в

вашем домашнем каталоге нет видеофайлов, музыки и прочих больших по размеру файлов, иначе можете не вписаться в размер болванки.

Если вам больше нравится GUI, то можете использовать GUI-версию программы (ничего особенного она собой не представляет — только окно с прямоугольными некрасивыми кнопками, позволяющими запустить ту или иную функцию программы). Запустить ее можно командой:

```
sudo remastersys-gui
```

Для создания LiveUSB в Ubuntu используется стандартная программа, запустить которую можно командой **Система | Администрирование | Создание загрузочного USB-диска**. Запустите ее (рис. 51.15), подключите флэшку (4 Гбайт или больше, 2 Гбайт будет маловато) и нажмите кнопку **Make startup disk**. Через некоторое время загрузочная флэшка будет готова.

51.5. Linux Live

Теперь очередь дошла и до Slackware. Очень хороший дистрибутив, пусть, может, и не такой удобный как Ubuntu, зато очень надежный. Для создания LiveCD в Slackware выполните следующие действия:

1. Соберите (если это еще не сделано) модули ядра: aufs, squashfs. Если собирать ядро лень, его можно заполучить уже готовое на сайте Linux Live (<http://www.linux-live.org/>). Правда, доступно ядро версии 2.6.27.27 — не самое новое и для архитектуры i486, но обычно Slackware не устанавливается на самые новые компьютеры с 64-разрядными процессорами. В Slackware 13 используется 2.6.33, поэтому, возможно, вам захочется собрать ядро вручную, чтобы в вашем LiveCD использовалась последняя версия ядра. Все необходимое для сборки (aufs, squashfs и lzma) вы найдете на сайте Linux Live.
2. Удалите все лишнее, например, лишнюю документацию и лишние программы — чтобы уменьшить размер дистрибутива.
3. Скачайте скрипты Linux Live с <http://www.linux-live.org/> и распакуйте их в /tmp.
4. Отредактируйте .config, если нужно изменить переменные по умолчанию.
5. Запустите ./build (находится в /tmp) с правами root. В результате появится каталог с данными LiveCD — /tmp/live_data_NNNN, где NNNN — случайное число.
6. Запустите make_iso.sh, если хочется создать ISO-образ, или bootinst.sh для создания LiveUSB.

ГЛАВА 52



Что делать в случае взлома?

52.1. 100% безопасности не гарантируется

Рано или поздно ваш сервер могут взломать. Степень защищенности сервера тоже особой роли не играет — можно "вскрыть" как сервер с настройками по умолчанию, так и самый секретный сервер Пентагона. Важно понимать: то, что защитил один человек, может взломать другой. Просто первый компьютер (с настройками по умолчанию) будет взломан за какой-то час (если не раньше), а на преодоление бастионов более защищенного будет потрачено больше времени. Вот и вся разница.

Так может лучше вообще не тратить время на настройку сервера, а использовать параметры по умолчанию? Ведь это как с автомобилем. Можно вообще не устанавливать сигнализацию — она все равно отпугивает только дилетантов, а настоящие профи, если захотят, то смогут угнать автомобиль. А лучшая защита от угона — это полное КАСКО.

С одной стороны, правильно — можно даже сэкономить несколько десятков тысяч рублей (вряд ли стоит покупать противоугонную систему за 3—5 тысяч — это выброшенные на ветер деньги). Но с другой — даже простейшая сигнализация отпугнет простого воришку, который не захочет разбивать стекло, чтобы стащить магнитолу или документы из машины. Когда станет выбор, стекло какой машины разбить — той, которая с сигнализацией или там где ее нет, выбор будет очевиден. А украсть могут не только магнитолу — можно еще украсть подушку безопасности (стоит довольно дорого), изуродовать панель автомобиля в процессе извлечения магнитолы (на аккуратную работу времени, сами понимаете, нет). Так что наличие сигнализации отпугнет воришек. А наличие серьезной сигнализации — отпугнет даже профессиональных автоворов. Зачем связываться с машиной, где установлена серьезная противоугонка, если рядом стоит аналогичная с системой попроще? Думаю, логика ясна. Совсем другое дело, если "заказали" именно ваш автомобиль — тогда, действительно, лучшая защита — это КАСКО. Автомобиль будет угнан в любом случае, но вы, хотя бы, получите материальную компенсацию.

С сервером все аналогично. Многое зависит от уровня подготовки злоумышленника и от преследуемых им целей. Если это дилетант, то базовая защита сервера отпугнет его — он попросту не сможет его победить, и будет искать другую систему, на которой "тренироваться" проще. Смысл ему взламывать три дня ваш сервер и в итоге не взломать, если можно за эти три дня найти с десяток незащищенных и взломать их. А каждый сервер очень поднимает самооценку, ради которой и осуществляются подобные мероприятия. То есть человеку все равно, чей сервер взломать, лишь бы поставить еще одну галочку (крестик, звездочку) в своем "послужном списке".

Другое дело, если вы имеете дело с более подготовленным специалистом. Ваш сервер могут взломать по двум причинам: если он "заказан", например, конкурентами, или если профи нечего делать, а взламывать "простые" машины ему не интересно — тогда он выберет ваш. Будет ли он взломан, зависит, конечно же, от степени защиты. Может, на взлом уйдет столько времени, что информация, которая могла быть получена в результате сего процесса, уже станет неактуальной. А если это взлом ради интереса, тогда рано или поздно хакеру надоест им заниматься (или же "подойдет" настоящая работа), поэтому он забудет о вашей машине до лучших времен.

Далее мы разберемся, что нужно делать, если вашу машину-таки "скомпрометировали".

52.2. Ваши действия в наихудшем варианте развития событий

Первым делом нужно отключить все сетевые интерфейсы — ведь в большинстве случаев атака производится по сети. Желательно отключить сетевые интерфейсы физически — просто отключите кабель от сетевого адаптера.

Затем нужно определить, каким образом злоумышленник проник в систему. Здесь ничего посоветовать не могу, кроме как анализировать журналы системы. Хотя, он может эти журналы почистить, тогда анализировать будет нечего. Тогда нужно подумать, как злоумышленник мог попасть в вашу систему. Если были запущены сервисы SSH и FTP, то в большинстве случаев именно они и скомпрометированы, искать "дыру" нужно именно в них.

Не нужно забывать и о человеческом факторе. От него никто не защищен. Предположим, что у вас в подчинении есть младший администратор, установивший слишком простой пароль для своей учетной записи. Пароль могли подобрать или же узнать его каким-либо другим способом. Потом злоумышленник использовал его имя пользователя и пароль для входа в систему. Как видите, никакой дыры нет — система же не может убедиться, что пароль вводит именно Петров, а не Сидоров.

Ясно одно. Злоумышленник, после того как проник в вашу систему, постарается внедрить backdoor — так называемый "черный" ход. Ведь он прекрасно понимает, что рано или поздно администратор закроет "дыру". А "черный" ход позволит ему и после этого проникать в систему и оставаться незамеченным.

Далее мы рассмотрим самые часто используемые способы организации backdoor.

52.2.1. Своя учетная запись

Самый простой способ добавить backdoor — это создать еще одного пользователя. Проверьте свой `/etc/passwd` — вдруг в нем появилась новая учетная запись пользователя, которого вы явно не создавали. Удалите ее немедленно. Также проанализируйте, есть ли в системе пользователи, которые заходят очень редко. Возможно, злоумышленник изменил пароль одного из таких пользователей. А т. к. пользователь заходит в систему очень редко, он ничего пока не заметил. Желательно изменить пароли всех пользователей, если это возможно. Новый пароль пользователь получит при личном обращении — когда заметит, что пароль изменен.

Также не забудьте про файлы паролей отдельных сервисов, например сервера Apache (см. файлы `.htaccess`).

52.2.2. Файлы `hosts.allow` и `hosts.deny`

Предыдущий способ был довольно прост — администратору легко определить учетную запись злоумышленника. А этот способ (изменение файлов `hosts*`) еще и глуп, поскольку злоумышленник в этом случае "засветит" свой IP-адрес или IP-адрес узла, через который он осуществлял взлом. Но, тем не менее, некоторые злоумышленники ничем не брезгают, а некоторые администраторы забывают проверять эти файлы на предмет записи, разрешающей обращение к серверу компьютера злоумышленника.

52.2.3. Сетевая файловая система

Возможно, злоумышленник оставил в `/etc/exports` запись вида:

```
/ компьютер_хакера(rw,no_root,squash)
```

Данная запись предоставляет полный доступ к вашей корневой файловой системе. Способ тоже глуп, поскольку "светится" компьютер злоумышленника, но иногда и такой способ хорош. Особенно когда администратор никогда в жизни не использовал NFS и даже не подозревает о существовании и назначении файла `/etc/exports...`

52.2.4. Руткиты

А вот если у злоумышленника было немного времени, и он не поленился установить руткит, вычислить его будет очень сложно. Руткит — это не вирус и не вредоносная программа. Руткит — это набор утилит, позволяющий скрыть присутст-

вие злоумышленника в системе. Как правило, руткит заменяет набор стандартных утилит — `ls`, `cat`, `ps`, `adduser`, `passwd` и др. Потом эти утилиты не показывают вам, администратору, следы присутствия злоумышленника. Например, хакер может установить сетевой сервис — аналог SSH для входа в систему. А чтобы вы не заметили "лишний" сервис, программа `ls` (точнее, ее руткит-версия) будет скрывать наличие в файловой системе ее исполнимого файла, `ps` — наличие сервиса в списке процессов, `netstat` — наличие открытого порта.

Найти руткит довольно сложно. Если руткит работает на пользовательском уровне, т. е. представляет собой просто набор приложений, как было сказано ранее, то обнаружить его довольно просто — контрольная сумма файлов стандартных программ будет отличаться от эталонных — файлов, входящих в состав дистрибутива. Если есть подозрение, достаточно сравнить контрольные суммы файлов `/bin/ls`, `/bin/ps` (можно проверить и остальные программы). Если они отличаются от эталонных, в вашу систему внедрен руткит. Наличие дополнительного сервиса, запущенного на сервере, можно проверить сетевым сканером `nmap`. Настоятельно рекомендуется запускать его с другой машины.

Совсем другое дело, если руткит работает на уровне ядра. Внедрить такой руткит намного сложнее, поскольку нужно перекомпилировать ядро, а на это нужно время. Зато обнаружить его практически невозможно. Дело в том, что стандартные утилиты остаются нетронутыми. А руткит, внедренный в код ядра, изменяет предоставляемую ядром информацию — таблицу процессов, листинг каталога и т. д. То есть утилиты вроде `ls`, `ps` и `netstat` уже получают измененную информацию от ядра. Вычислить наличие такого руткита можно, сравнив вывод утилит `ps`, `netstat` и т. д. с выводом этих же утилит, но при условии, что система загружена на другом ядре — предварительно нужно установить это другое ядро.

В Интернете можно найти много программ для обнаружения руткитов, а также множество методик. Пока вы не столкнулись с руткитом лично, для общего развития рекомендую прочитать вот эту статью:

<http://www.xakep.ru/post/42886/default.asp>

52.2.5. Модули ядра

Устанавливать ядро на удаленный компьютер довольно проблематично. Самая большая проблема — нужна перезагрузка компьютера, а ее могут заметить администраторы. Однако свой код можно внедрить в ядро с помощью модулей. Злоумышленник загружает на ваш компьютер уже откомпилированные модули ядра и добавляет их в ядро командой `insmod`. Заметить это действие практически невозможно. Правда, есть одно "но". Наверняка злоумышленник захочет, чтобы его модуль был доступен и после перезагрузки — ведь рано или поздно она произойдет. Поэтому он может добавить его в файл `/etc/modules.conf` (или `conf.modules`). А в этом случае вы легко обнаружите лишний модуль ядра.

52.2.6. Удаленный командный интерпретатор

Зачем усложнять себе жизнь и создавать собственную версию SSH, устанавливать ее на компьютере жертвы? Ведь можно просто изменить `/etc/xinetd.conf` и добавить строки вида:

```
service myshell
{
    socket_type = stream
    protocol = tcp
    wait = no
    user = root
    server = /bin/bash -i
    port = 9099
    instances = UNLIMITED
}
```

После этого, подключившись по telnet к порту 9099, вы получаете доступ к командному интерпретатору bash с правами root (!). А больше ничего вам и не нужно — путем простого редактирования одного файла вы получаете практически полный контроль над системой!

Так что не забудьте посмотреть свой `/etc/xinetd.conf` или `/etc/inetd.conf` (на старых компьютерах) на предмет описания лишних сервисов.

Также проверьте файлы `/etc/hosts.equiv` и `~/.rhosts`, в которых могут быть строки, разрешающие ввод команд удаленному пользователю от имени локального пользователя (кроме root). А чтобы вообще обезопаситься и отключить данную возможность, отключите в вашем `inetd/xinetd` следующие сервисы:

```
in.rshd
in.rexecd
in.rlogind
```

52.2.7. Настройка PHP и CGI

В настройках вашего Web-сервера или в настройках интерпретатора (для PHP — это файл `php.ini`) отключите возможность использования команд `system()` и `exec()`, которые позволяют выполнять различные команды от имени пользователя, запустившего Apache. Некоторые администраторы запускают Apache от имени root... Последствия можете себе представить.

Для PHP откройте `php.ini` и добавьте в него следующую строку (или раскомментируйте ее, если она закомментирована):

```
disable_functions = exec,passthru,shell_exec,system, proc_open,
popen,curl_exec,curl_multi_exec,parse_ini_file,show_source
```

Для остальных интерпретаторов вам нужно прочитать соответствующее руководство. А вообще, по возможности, откажитесь от CGI в пользу PHP.

После редактирования файла `php.ini` нужно перезапустить Apache, чтобы изменения вступили в силу.

52.2.8. SSH — огромная дыра

Сервис SSH — это огромная черная дыра в безопасности вашего сервера. И вовсе не потому, что он дыряв, как первые версии sendmail. Вовсе нет — это довольно безопасный сервис, но при одном условии: он требует правильной настройки.

Начнем с аутентификации по ключу. Если у клиента есть ключ, `sshd` проверяет его, и если проверка прошла успешно, клиенту предоставляется доступ. Но если злоумышленник ранее получил полный доступ к системе (как он это сделал — это уже другой вопрос) и добавил свой ключ в `/etc/ssh_known_hosts`?

Советую открыть ваш `sshd_config` и установить директиву `IgnoreRhosts` в значение `yes`. Это защитит вашу систему от несанкционированного использования файлов `hosts.equiv` и `.rhosts`.

Также проверьте, чтобы в `root/.ssh/authorized.keys` не было "лишних" ключей. Ведь если злоумышленник поместит в этот файл свой ключ, то потом сможет войти как `root` на ваш компьютер с помощью следующей команды:

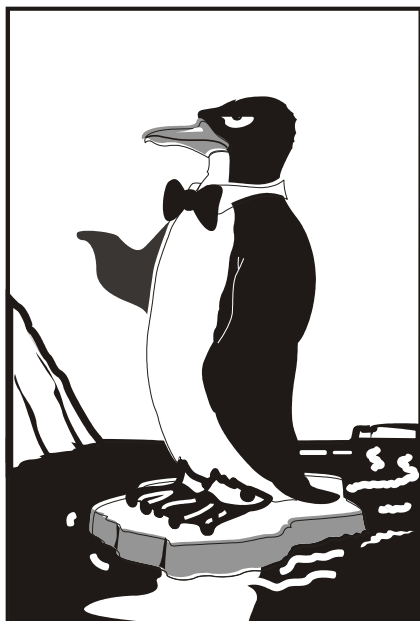
```
ssh -lroot адрес_вашего_компьютера
```

Как видите, SSH довольно защищен, но из-за "полиморфизма" аутентификации вы можете не заметить одну из "дыр". Правда, чтобы ее реализовать, злоумышленнику нужно получить `root`-доступ каким-либо другим способом.

Заключение

Надеюсь, что данная книга помогла вам настроить и защитить ваш сервер. Если у вас есть какие-нибудь вопросы относительно настройки сервера, вы можете их задать на форуме сайта **www.dkws.org.ua**.

Если же у вас есть комментарии или пожелания относительно данной книги, вы можете оставить свой отзыв на сайте издательства — **www.bhv.ru**.



ПРИЛОЖЕНИЯ

ПРИЛОЖЕНИЕ 1



Настройка принтера в Linux

О настройке принтера в Linux вам нужно знать следующие факты.

- ◆ Linux поддерживает как LPT-принтеры, так и USB-принтеры.
- ◆ Настройку принтера проще производить с помощью графического конфигуратора:
 - ◆ `printerdrake` — в Linux Mandriva;
 - ◆ `system-config-printer` — в Fedora Core;
 - ◆ в Ubuntu для вызова конфигуратора принтера командой меню **Система | Администрирование | Печать**.
- ◆ Если графический конфигуратор не смог определить ваш принтер, значит, он не поддерживается. Что делать? Возможны следующие варианты:
 - ◆ попробовать выбрать другой драйвер для похожей модели принтера. Однако никто не может гарантировать, что принтер будет работать корректно и что он будет работать вообще;
 - ◆ попробовать другой дистрибутив;
 - ◆ дождаться новой версии вашего дистрибутива (в случае с Ubuntu ждать нужно полгода, а со всеми остальными — больше года);
 - ◆ если у вас установлена не самая новая версия дистрибутива, установите самую последнюю. Наверняка в ней ваш принтер поддерживается;
 - ◆ если у вас GID-принтер (он же Windows-принтер), можете забыть о том, чтобы использовать его в Linux. Такие принтеры можно использовать только в Windows.
- ◆ Настроить сетевой принтер можно с помощью того же графического конфигуратора — просто при выборе типа принтера нужно указать **сетевой** (а не **локальный**).

GDI-принтеры заслуживают отдельного разговора, поскольку GDI-принтеры Linux не поддерживает. При покупке принтера вам нужно уточнить у продавца, сможет ли принтер работать с Linux. Скорее всего, на компакт-диске с драйверами будут драйверы для всего семейства Windows, а также для Mac OS, 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 не знать, 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 спросить 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. 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. 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 в Интернете:

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

ПРИЛОЖЕНИЕ 2



Параметры ядра

Параметры ядра позволяют управлять поведением ядра. Как уже было отмечено ранее, мы можем передать параметры ядра непосредственно при загрузке, используя меню загрузчика, или же прописать параметры ядра в файлах конфигурации загрузчика. Первый случай подходит для "одноразового" использования того или иного параметра, а второй — если параметр нужен для корректной работы системы, поэтому, чтобы не указывать его каждый раз при загрузке Linux, намного проще внести его в файл конфигурации загрузчика.

Параметров ядра очень много, поэтому в табл. П2.1 собраны самые полезные.

Таблица П2.1. Некоторые параметры ядра Linux

Параметр	Описание
<code>root=устройство</code>	Позволяет указать корневую файловую систему. Например, <code>root=/dev/sda5</code>
<code>ro</code>	Монтирует корневую файловую систему в режиме "только чтение". Используется по умолчанию. После проверки файловой системы программой <code>fsck</code> корневая файловая система перемонтируется в режим <code>rw</code>
<code>rw</code>	Монтирует корневую файловую систему в режиме "чтение/запись". При использовании этого параметра нельзя запускать программы типа <code>fsck</code> . Перед запуском <code>fsck</code> нужно перемонтировать корневую файловую систему в режиме <code>ro</code>
<code>mem=</code>	Определяет объем памяти, установленной в компьютере. Иногда ядро неправильно определяет объем оперативной памяти. Вы можете помочь ему в этом, указав параметр <code>mem</code> . Только указывать его нужно правильно, например: <code>mem=768M</code> После числа обязательно должна следовать буква <code>M</code> , иначе ядро "подумает", что объем оперативной памяти 768 байтов

Таблица П2.1 (окончание)

Параметр	Описание
init=	Позволяет задать программу инициализации. По умолчанию используется программа /sbin/init, но вы можете задать другую программу, например /bin/bash, если вам нужно обойти сценарии init (например, когда вы забыли пароль root)
reboot=	Позволяет задать тип перезагрузки компьютера. Возможные значения: cold и warm, т. е. "холодная" или "горячая" перезагрузка
single	Однопользовательский режим для администрирования системы, например в случае отказа
nodmraid	Отключает программные RAID-массивы, организованные на уровне BIOS
noapic	Полезен, если вы при загрузке увидите сообщение: kernel panic - not syncing: IO-APIC + timer doesn't work! Подробнее об этом параметре вы можете прочитать по адресу: http://www.dkws.org.ua/phpbb2/viewtopic.php?topic=2973&forum=5
norpcmcia	Отключает PCMCIA-карты (для ноутбуков). Полезен, если вы подозреваете, что у вас проблемы с PCMCIA-картой
nodma	Отключается DMA (Direct Memory Access, прямой доступ к памяти) для всех IDE-устройств
noapm	Отключает APM (Advanced Power Management) — расширенное управление питанием
nousb	Отключает поддержку USB
noscsi	Отключает поддержку SCSI
pci=noacpi	Не использовать ACPI для управления PCI-прерываниями
acpi=off	Полностью отключает ACPI (Advanced Configuration and Power Interface). Полезен на некоторых ноутбуках, когда не удается установить (а потом загрузить) Linux
edd=off	Отключает поддержку EDD (Enhanced Disk Device). Полезен, если установка зависает при определении параметров жесткого диска или же параметры жесткого диска, определенные программой установки, не соответствуют реальным (например, не совпадают размер и количество разделов)

ПРИМЕЧАНИЕ

С дополнительными параметрами ядра вы можете ознакомиться по адресу <http://dkws.org.ua/phpbb2/viewtopic.php?t=3031>.

ПРИЛОЖЕНИЕ 3



"Горячее" администрирование с помощью /proc

Виртуальная (псевдофайловая) система /proc — это специальный механизм, позволяющий посылать информацию ядру, модулям и процессам (кстати, потому данная файловая система так и называется: `proc` — это сокращение от англ. *process*). Также, используя /proc, вы можете получать информацию о процессах и изменять параметры ядра и его модулей "на лету". Для этого в /proc есть файлы, позволяющие получать информацию о системе, ядре или процессе, и есть файлы, с помощью которых можно изменять некоторые параметры системы. Первые файлы мы можем только просмотреть, а вторые — просмотреть и, если нужно, изменить.

Просмотреть информационный файл можно командой `cat`:

```
cat /proc/путь/<название_файла>
```

Записать значение в один из файлов `proc` можно так:

```
echo "данные" > /proc/путь/<название_файла>
```

П3.1. Информационные файлы

В табл. П3.1 представлены некоторые (самые полезные) информационные `proc`-файлы: с их помощью вы можете получить информацию о системе.

Таблица П3.1. Информационные `proc`-файлы

Файл	Описание
<code>/proc/version</code>	Содержит версию ядра
<code>/proc/cmdline</code>	Список параметров, переданных ядру при загрузке
<code>/proc/cpuinfo</code>	Информация о процессоре
<code>/proc/meminfo</code>	Информация об использовании оперативной памяти (почти то же, что и команда <code>free</code>)

Таблица ПЗ.1 (окончание)

Файл	Описание
/proc/devices	Список устройств
/proc/filesystems	Файловые системы, которые поддерживаются вашей системой
/proc/mounts	Список подмонтированных файловых систем
/proc/modules	Список загруженных модулей
/proc/swaps	Список разделов и файлов подкачки, которые активны в данный момент

ПЗ.2. Файлы, позволяющие изменять параметры ядра

Каталог `/proc/sys/kernel` содержит файлы, с помощью которых вы можете изменять важные параметры ядра. Конечно, все файлы мы обсуждать не будем, а рассмотрим лишь те, которые используются на практике (табл. ПЗ.2).

Таблица ПЗ.2. Файлы каталога `/proc/sys/kernel`

Файл	Каталог
<code>/proc/sys/kernel/ctrl-alt-del</code>	Если данный файл содержит значение 0, то при нажатии комбинации клавиш <code><Ctrl>+<Alt>+</code> будет выполнена так называемая "мягкая перезагрузка", когда управление передается программе <code>init</code> , и последняя "разгружает" систему, как при вводе команды <code>reboot</code> . Если этот файл содержит значение 1, то нажатие комбинации клавиш <code><Ctrl>+<Alt>+</code> равносильно нажатию кнопки <code>Reset</code> . Сами понимаете, значение 1 устанавливать не рекомендуется
<code>/proc/sys/kernel/domainname</code>	Здесь находится имя домена, например dkws.org.ua
<code>/proc/sys/kernel/hostname</code>	Содержит имя компьютера, например <code>den</code>
<code>/proc/sys/kernel/panic</code>	При критической ошибке ядро "впадает в панику" — работа системы останавливается, а на экране красуется надпись "kernel panic" и выводится текст ошибки. Данный файл содержит значение в секундах, которое система будет ждать, пока пользователь прочитает это сообщение, после чего компьютер будет перезагружен. Значение 0 (по умолчанию) означает, что перезагружать компьютер вообще не нужно

Таблица ПЗ.2 (окончание)

Файл	Каталог
/proc/sys/kernel/printk	Данный файл позволяет определить важность сообщения об ошибках. По умолчанию файл содержит значения 6 4 1 7. Это означает, что сообщения с уровнем приоритета 6 и ниже (чем ниже уровень, тем выше важность сообщения) будут выводиться на консоль. Для некоторых сообщений об ошибках уровень приоритета не задается. Тогда нужно установить уровень по умолчанию. Это как раз и есть второе значение — 4. Третье значение — это номер самого максимального приоритета, а последнее число — значение по умолчанию для первого значения. Обычно изменяют только первое значение, дабы определить, какие значения должны быть выведены на консоль, а какие — попасть в журнал демона syslog

ПЗ.3. Файлы, изменяющие параметры сети

В каталоге /proc/sys/net вы найдете файлы, изменяющие параметры сети (табл. ПЗ.3).

Таблица ПЗ.3. Файлы каталога /proc/sys/net

Файл	Описание
/proc/sys/net/core/message_burst	Опытные системные администраторы используют этот файл для защиты от атак на отказ (DoS). Один из примеров DoS-атаки — когда система заваливается сообщениями атакующего, а полезные сообщения системой игнорируются, потому что она не успевает реагировать на сообщения злоумышленника. В данном файле содержится значение времени (в десятых долях секунды), необходимое для принятия следующего сообщения. Значение по умолчанию — 50 (5 секунд). Сообщение, попавшее в "перерыв" (в эти 5 секунд), будет проигнорировано
/proc/sys/net/core/message_cost	Чем выше значение в этом файле, тем больше сообщений будет проигнорировано в перерыв, заданный файлом message_burst
/proc/sys/net/core/netdev_max_backlog	Задаёт максимальное число пакетов в очереди. По умолчанию 300. Используется, если сетевой интерфейс передает пакеты быстрее, чем система может их обработать
/proc/sys/net/core/optmem_max	Задаёт максимальный размер буфера для одного сокета

П3.4. Файлы, изменяющие параметры виртуальной памяти

В каталоге `/proc/sys/vm` вы найдете файлы, с помощью которых можно изменить параметры виртуальной памяти:

- ◆ в файле `buffermem` находятся три значения (разделяются пробелами): минимальный, средний и максимальный объем памяти, которую система может использовать для буфера. Значения по умолчанию: `2 10 60`;
- ◆ в файле `kswapd` тоже есть три значения, которые можно использовать для управления подкачкой:
 - ◆ первое значение задает максимальное количество страниц, которые ядро будет пытаться переместить на жесткий диск за один раз;
 - ◆ второе значение — минимальное количество попыток освобождения той или иной страницы памяти;
 - ◆ третье значение задает количество страниц, которые можно записать за один раз. Значения по умолчанию: `512 32 8`.

П3.5. Файлы, позволяющие изменить параметры файловых систем

Каталог `/proc/sys/fs` содержит файлы, изменяющие параметры файловых систем. В частности:

- ◆ файл `file-max` задает максимальное количество одновременно открытых файлов (по умолчанию `4096`);
- ◆ в файле `inode-max` содержится максимальное количество одновременно открытых индексных дескрипторов (максимальное значение также равно `4096`);
- ◆ в файле `super-max` находится максимальное количество используемых суперблоков;

ПРИМЕЧАНИЕ

Поскольку каждая файловая система имеет свой суперблок, легко догадаться, что количество подмонтируемых файловых систем не может превысить значение из файла `super-max`, которое по умолчанию равно `256`, чего в большинстве случаев вполне достаточно. Наоборот, можно уменьшить это значение, дабы никто не мог подмонтировать больше файловых систем, чем нужно (если монтирование файловых систем разрешено обычным пользователям).

- ◆ в файле `super-ng` находится количество открытых суперблоков в текущий момент. Данный файл нельзя записывать, его можно только читать.

ПЗ.6. Как сохранить изменения?

Итак, вы изменили некоторые параметры системы с помощью /proc, и теперь вам нужно их сохранить. Для этого их нужно прописать в файле /etc/sysctl.conf. Вот только формат этого файла следующий: надо отбросить /proc/sys/ в начале имени файла, а все, что останется, записать через точку, а затем через знак равенства указать значение параметра. Например, для изменения параметра /proc/sys/vm/buffermem нужно в файле etc/sysctl.conf прописать строку:

```
vm.buffermem = 2 11 60
```

Если в вашем дистрибутиве нет файла /etc/sysctl.conf, тогда пропишите команды вида `echo "значение" > файл` в сценарий инициализации системы.

Предметный указатель

A

ACL 358, 473
ADSL-сплиттер 284
AES 466
Apache 305, 449
ASP Linux 367
ASPDiskManager 110
AUX 472

B

BIOS 173
BlowFish 413
Bluefish 122

C

callback 389
CDP 475
CentOS 367
chroot-окружение 456
ClamAV 444
CUPS 189
cyrus-pop3d, сервис 328

D

Denix 269
Destination NAT 375
DHCP 352
Disk Druid 110
DNS 337
DNS-сервер 344, 348
DoS-атака 474

E

EAP 468
EDD 10

F

Fast Ethernet 251
Firewall 372
Flash-память 67
FreeS/WAN 390
FTP 319

G

Gigabit Ethernet 251
GNOME 21
GRUB2 171

H, I

Human Interface Device Daemon 189
IDEA 413
IDS 471
initrd 174
IOS 471
IpSec 390
IP-spoofing 474
iptables 361, 376
IPv4 forwarding 371, 376
ISO9660 91

K, L

KDE 21
Land-атака 475

LiveCD 76
LWP 447

M

MAC-адрес 269, 354, 467
MBR 173
MD5 122
MIB 473
MPPE 397
MTA 328, 445
MTU 269, 280
MySQL 430

N

NAT 373
NCSA HTTPd 1.3 305
Network File System 437
NetworkManager, отключение 268
ntfs-3g 77

O

OpenS/WAN 390
OpenSSL 329
OSI 369

P

P3Scan 445
PID 184
PMTU 474
POP3 328
POST 8
postfix, сервис 328
PPTP 390
ProFTPD 319, 449

Q

Qmail 328
QmailAdmin 332
Qpopper 329

R

Radio Ethernet 465
RADIUS 469

RAID 82, 190
RAS 389
Remote Administrator 119
RPC 190
Rpmdrake 226
RSA 413

S

S.M.A.R.T 190
Samba 437, 451
sendmail 328
Slackware 181
SMB 190
SMTP 328
SNMP 473
SOA 345
Source NAT 376
SQL-оператор 316, 432
 create 433
 delete 436
 insert 434
 select 434
 update 435
Squid, прокси-сервер 356
squidGuard, расширение 362
SSH 412, 413
SSID 465
SYN flood 475
Synaptic 238

T

Telnet 412
TKIP 466
TLD 337
Tomoyo 459
TSIG 450

U

Ubuntu, DNS 343
udev 46
UDF 91
Universally Unique Identifier (UUID) 65
UNIX 204
URL, черный список 359

V

Viralator 446
VLAN 270
VMWare 366

W

WEP 466
WPA 466, 467
wu-ftpd 319

X

xinetd, суперсервер 303
X-терминалы 412

Y

YaST 110

A

Антивирус 443

Б

Баннер, черный список 359
Библиотека, libata 66
Брандмауэр 372
Браузер:
 elinks 297
 links 297
 lynx 296

В

Вывод:
 перенаправление 196
 постраничный 203

Г

Графическая подсистема 21
Графическая среда 21

Д

Демон 187
Дефрагментация 15
Директива:
 AllowRootLogin 141
 default-leased-time 354

DefaultRoot 325
Directory 311
Files 313
Limit 312
MaxClients 325
max-leased-time 354
ServerName 310

Диск:

 USB 67
 виртуальный 173
 гибкий 61

Домен 344

Ж

Журнал 37
 файловой системы 76

З

Загрузчик:
 ASPLoader 8, 153
 GRUB 8, 153
 GRUB2 154
 LILO 8, 153, 173
Зона 344

И

Имя диска 65, 66
Интерфейс:
 eth0 261
 lo 256

К

Каталог 49, 52

- /etc 106
- /etc/cron.daily 216
- /etc/cron.hourly 216
- /etc/cron.weekly 216
- /etc/event.d 179
- /etc/rc.d 176
- /etc/rc.d/init.d 176
- /etc/skel 122
- /etc/xinetd.d 304
- /etc/zypp/repos.d 245
- /home 107
- /var/cache/apt/archives 236
- /var/lib/mysql 107
- /var/named 107
- /var/www/html 107
- домашний 52
- права доступа 55
- признак каталога 56
- родительский 52
- текущий 52

Квотирование 144

Команда:

- /sbin/grub-install 161
- /sbin/init 176
- adduser 120, 451
- alien 237
- apt 224
- apt-get 236
- arch 198
- at 218
- atq 218
- atrm 218
- cat 50
- cd 52
- cdrecord 71
- chmod 56
- chmod +x 210
- chown 57
- chroot 75
- clamscan 444
- clear 196, 198
- convert 165
- cp 50
- date 199
- dd 70
- df 203
- diff 202

- dmesg 197
- dpkg 224, 234
- dvd+rw-format 71
- echo 199
- edquota 147
- exit 28, 138, 199
- fdisk 46, 61, 110, 111
- find 58
- free 203, 422
- freshclam 444
- fsck 63, 75
- ftp 297
- gpart 75
- grep 202
- groupadd 122
- grub 170
- grub-mkconfig 161
- grub-mkpasswd-pbkdf2 172
- gzip 165
- hdparm 77
- head 203
- ifconfig 256, 272, 295
- insmod 296
- kdesu 138
- kill 184
- killall 185
- less 50, 203
- ln 55
- locate 50, 59
- logout 28, 195
- ls 53
- mail 300
- man 194
- md5sum 204
- mkdir 52
- mkfs 74
- mkisofs 72
- mkraid 85
- more 203
- mount 60, 439
- mv 50
- netstat 365
- nice 187
- nslookup 343
- ntsysv 177
- parted 111, 114
- passwd 31, 120, 199
- perl 447
- ping 274
- poweroff 28
- pppoeconf 291
- pptp-command 400

ps 184
qmailctl 333
quotaon 146
raidhotadd 85
raidhotremove 85
reboot 28
repquota 147
rm 50, 53
rmdir 53
rncd-configen 344
route 365, 369
rpm 223
scp 109
service 176, 187
shutdown 28
smbpasswd 451
ssh 204, 413
startx 200
su 137
sudo 61, 136
swapon 74, 181
system-config-packages 224
tac 50
tail 203
tar 107
telnet 204
top 186
touch 50
tracpath 275
traceroute 275
umount 61
update-grub 161
uptime 200
userdel 121
usermod 121
userpasswd 124
users 200
vi 147
visudo 137
w 201
wc 203
wget 298
which 50, 59
who 119, 201
whoami 201
xf86config 201
yum 224, 239
 для работы со ссылками 54
 псевдоним 195
Комментарий 210
Консоль 193
 эмулятор 196

Конфигуратор:
 diskdrake 46, 110
 drakboot 178
 drakconnect 255
 drakuser 124
 drakxservices 179, 189
 gproftpd 320, 327
 harddrake2 190
 netconfig 256
 network-admin 256
 pppoeconf 274
 rpm-drake 144, 224
 system-config-network 256
 system-config-packages 144
 system-config-services 177, 189
 system-config-users 123

М

Маршрутизатор 372, 471
Массив 212
Менеджер пакетов 246
Модуль ntfs-3g 77

О

Оператор 213, 214
Оптимизация:
 подкачки 422
 сетевых сервисов 425

П

Пакет 221
 bind 339
 clamav 444
 clamav-db 444
 clamd 444
 dhcp 352
 mysql-admin 315, 430
 mysql-client 315, 430
 mysql-server 315, 430
 nfs-common 437
 nfs-user-server 437
 nfs-utils 437
 ntp 385
 ph5-cli 306
 php5-gd 306
 php5-imap 306
 php5-mysql 306

- pptp-client 400
- pptp-linux 400
- proftpd 320
- raidtools 85
- samba-server 407
- программы управления 223
- Память, виртуальная 80, 504
- Параметры ядра 499
- Переменная 210
 - окружения 211
 - специальная 211
- Планировщик:
 - anacron 217
 - atd 218
 - crond 215
- Пользователь root 61
- Почта 300
- Программа:
 - /usr/sbin/grub-mkconfig 157
 - CloneCD 73
 - ftpcount 326
 - ftpwho 326
 - ISOpen 70
 - k3b 93
 - mc 109
 - Nero 101
 - rndc 342
 - testdisk 115
 - UltraISO 70
 - urpmi 228
 - zypper 247
 - управления пакетами 223
- Прокси-сервер 356, 359
 - прозрачный 360
- Прототипы 149
- Процесс 184

Р

- Разметка диска 15
 - diskdrake 115
 - gparted 115
- Редактор:
 - joe 207
 - mcedit 208
 - nano 207
 - pico 207
 - vi 204
- Режим:
 - journal 76
 - ordered 76
 - writeback 76

С

- Сервер X 21
- Сервис 189
 - network 256
 - сетевой 303
- Сеть, отказ работы 272
- Синхронизация времени 385
- Система доменных имен 337
- Система инициализации:
 - init 174
 - upstart 174, 179
- Служба 187
- Сменные носители 43
- Событие:
 - network-interface-added 181
 - network-interface-up 181
- Соединение DSL 283
- Справочная система 194
- Ссылка 54
 - жесткая 55
 - символическая 55
- Стример 104
- Сценарий 210

Т

- Точка монтирования 15, 49, 60, 62, 64

У

- Устройство 45, 61

Ф

- Файл 44, 50
 - .{ICE,X}authority 138
 - .bash_history 195
 - .bash_logout 195
 - .bash_profile 195
 - /boot/boot.b 173
 - /boot/grub/grub.cfg 157
 - /boot/grub/grub.conf 154
 - /boot/grub/menu.lst 154
 - /boot/map 173
 - /etc/apt/sources.list 236
 - /etc/bind/named.conf 339
 - /etc/bind/named.conf.local 340
 - /etc/bind/named.conf.options 340

/etc/crontab 215
 /etc/default/grub 159
 /etc/dhcpd.conf 352
 /etc/exports 438
 /etc/fstab 31, 63, 76, 144
 /etc/group 122
 /etc/hostname 280
 /etc/HOSTNAME 280
 /etc/httpd/conf 428
 /etc/init.d/rc 180
 /etc/inittab 32, 174
 /etc/ipsec/ipsec.conf 392
 /etc/kde/kdm/kdmrc 141
 /etc/lilo.conf 425
 /etc/network/interfaces 269, 280, 368
 /etc/ntp.conf 386
 /etc/p3scan/p3scan.conf 445
 /etc/passwd 121
 /etc/proftpd/proftpd.conf 320
 /etc/raidtab 85
 /etc/rc.d/rc.S 181
 /etc/resolv.conf 343
 /etc/route.conf 368
 /etc/samba/smb.conf 407
 /etc/shadow 122
 /etc/shells 450
 /etc/squid/squid.conf 447
 /etc/squid/squidGuard.conf 446
 /etc/sshd_config 414
 /etc/sudoers 137
 /etc/sysconfig/network 277
 /etc/sysconfig/network/config 279
 /etc/sysconfig/network/dhcp 280
 /etc/sysconfig/network/ifcfg-eth0 279
 /etc/sysconfig/network/routes 279, 368
 /etc/sysconfig/network-scripts/
 ifcfg-eth0 277
 /etc/sysconfig/static-routes 279
 /etc/urpmi/urpmi.conf 229
 /etc/xinetd.conf 303
 /etc/yum.conf 241
 /etc/zypp/zypp.conf 246
 /proc/sys/vm/swappiness 423
 /var/log/messages 272
 apache.conf 309
 apache2.conf 309
 dnssec-keygen 450
 etc/squid/squid.conf 356
 fstab 63

httpd.conf 309
 httpd2.conf 309
 lilo 425
 resolv.conf 343
 smb.conf 411, 425
 длина имени файла 44
 имя файла 44
 права доступа 55
 сравнение с файлом 202
 устройств 45, 61
 Файловая система 37, 38
 ext2 38
 ext3 38
 JFS 38
 ReiserFS 38
 XFS 38
 журналируемая 38, 76
 корневая 48
 монтирование 60
 размонтирование 61
 сетевая 437
 удаленная, монтирование 439
 Формат:
 DVD+R 93
 DVD-10 91
 DVD-18 91
 DVD-5 91
 DVD-Audio 91
 DVD-R 92
 DVD-ROM 91
 DVD-RW 92
 DVD-Video 91

Ц, Ш

Цикл 212, 213
 Шлюз 372

Я

Ядро 9
 параметр:
 init 33
 nofb 30
 rw 33
 single 31
 ядра 9
 системный вызов 9