

Алексей Стахнов



# Linux



- Особенности инсталляции
- Настройка Proxy-сервера и брандмауэра
- Использование "экзотической" периферии
- Недокументированные возможности

**Наиболее  
полное  
руководство**

**В ПОДЛИННИКЕ®**

# Алексей Александрович Стахнов

# Linux

Серия «В подлиннике.  
Наиболее полное руководство»

*Текст предоставлен издательством*  
*[http://www.litres.ru/pages/biblio\\_book/?art=640705](http://www.litres.ru/pages/biblio_book/?art=640705)*  
*Linux: БХВ-Петербург; СПб; 2002*  
*ISBN 5-94157-146-1*

## Аннотация

Книга посвящена операционной системе Linux. Приводятся подробные сведения о ее особенностях и возможностях, идеологии файловой системы, инсталляции и основных командах, вопросах компиляции ядра, настройках и сервисах. Большое внимание уделяется организации на базе Linux различных серверов и служб: электронной почты, WWW, FTP, INN, Proxy, NTP, а также проблемам администрирования сети, обеспечения безопасной работы и другим вопросам. Описаны способы настройки под Linux рабочих станций, в т. ч. и бездисковых, установки и эксплуатации на них графических сред типа X Window, а также конфигурирование модемных соединений, принтеров и сканеров, отладка взаимодействия с Linux-машинами такой «экзотической» периферии, как карманные компьютеры, мобильные телефоны, TV-тюнеры и

т. п. Рассматриваемые в книге конфигурационные файлы и структура каталогов соответствуют дистрибутиву Red Hat Linux 7.x, тем не менее, при минимальной адаптации все упоминаемые в книге пакеты устанавливаются в любом дистрибутиве Linux. *Для начинающих администраторов или пользователей Linux.*

# Содержание

Часть I	8
Глава 1	8
DOS	11
Windows 3.1x	12
OS/2	13
Windows 9x	14
Windows NT (Windows 2000)	15
Mac OS	17
Mac OS X	17
Семейство UNIX	18
FreeBSD, OpenBSD, NetBSD	20
Linux	20
BeOS	22
QNX	23
Почему выбирают Linux	40
Разные факты	42
Ссылки	43
Глава 2	45
Сеть	46
Сетевые протоколы и аппаратура	46
Сетевые сервисы	48
Файловые менеджеры	51
Текстовые редакторы	54

Графические оболочки	55
Графические редакторы	57
Web-инструментарий	58
Офисные пакеты	59
StarOffice 5.2	61
OpenOffice	63
Koffice	64
GNOME Workshop	65
Базы данных	66
Эмуляторы Windows	67
Средства разработки программ	67
Kylix	68
Develop	69
Glade	70
VDK Builder	71
Motor	71
Rhide	71
SNiFF+ Penguin IDE	72
Code Forge	72
CodeWarrior	73
CRiSP	73
Мультимедиа-приложения	74
Аудио	74
Видео	76
Игры	77
Итоги	78

Ссылки	79
Часть II	81
Глава 3	81
Модели сетевых взаимодействий	81
Терминология	82
Модель взаимодействия открытых систем (OSI)	84
Модель сетевого взаимодействия TCP/IP	87
Сопоставление сетевых моделей OSI и TCP/IP	87
Сетевые протоколы	88
Семейство протоколов TCP/IP	88
Протоколы межсетевого уровня (интернет)	90
Протокол IP	90
Сетевые пакеты	98
Протокол адресации ARP/RARP	105
Протокол ICMP	105
Протоколы транспортного уровня	106
Протокол TCP	107
Протокол UDP	108
Протоколы уровня приложений	109
Протокол FTP	110
Протокол SMTP	110
Протокол Telnet	110
Сетевая файловая система NFS	111

Протокол IPX	111
Протокол AppleTalk	113
Протокол NetBIOS	113
Протокол DECnet	113
Стандарты в Интернете	113
Ссылки	115
Глава 4	116
История развития файловых систем Linux	116
Файл	117
Типы файлов	118
Владельцы файлов	121
Права доступа к файлам	122
Модификаторы прав доступа	124
Файловые системы	125
Типы файловых систем	126
Установка файловой системы	129
Монтирование и демонтирование файловой системы	130
Поддержка работоспособности файловых систем	133
Виртуальная файловая система (VFS)	135
Файловая система Ext2	138
Конец ознакомительного фрагмента.	144

# Алексей Александрович Стахнов Linux

## Часть I Введение в Linux

### Глава 1 Особенности ОС Linux

*Я считаю, что Microsoft создала объективно плохую операционную систему, и мне интересно наблюдать, как это постепенно доходит до людей.  
Линус Торвальдс, создатель ОС Linux*

Мир операционных систем предоставляет пользователям достаточно большое их количество. Мы не будем подробно останавливаться на истории и особенностях каждой операционной системы – для этого понадобится чрезмерно много места и времени. Да и не всем это интересно: раз вы читаете эту книгу, значит вас заинтересовала Linux. Особо любопытным можно предложить обратиться к соответствующей



литературе и Интернету – там можно найти много интересной информации на этот счет.

Какие же операционные системы используются в настоящее время? Вот некоторые из них:

- DOS (MS-DOS, DR-DOS и их клоны);
- Windows 3.1x;
- OS/2;
- Windows 9x;
- Windows NT (Windows 2000);
- Mac OS;
- Mac OS X;
- семейство UNIX;
- Free BSD, OpenBSD, NetBSD;
- Linux;
- BeOS;
- QNX.

Конечно, приведенный список далеко не полон, но мы сознательно ограничим его операционными системами, используемыми наиболее часто. Рассмотрим этот список с точки зрения человека, которому необходимо иметь на компьютере операционную систему, удовлетворяющую нескольким, порой противоречивым, требованиям.

Операционные системы можно классифицировать по многим параметрам. Во-первых, все они делятся на два вида – платные и бесплатные (условно-бесплатные). Во-вторых, правомерно разделить их на операционные системы с откры-

тым исходным кодом (с правом вносить изменения или без права внесения изменений) и с закрытым исходным кодом. В-третьих, операционные системы бывают одноплатформенные (способные функционировать только на одной платформе, например РС-совместимой) и многоплатформенные (способные функционировать на нескольких платформах, например РС-совместимой, Macintosh, Sun, PowerPC). В-четвертых, операционные системы могут быть однозадачными и многозадачными. В-пятых, однопользовательскими и многопользовательскими. В-шестых, серверными, клиентскими или универсальными. В-седьмых, иметь текстовый, графический или тот и другой интерфейсы. В-восьмых, ориентацией на работу с сетью и Интернетом. В-девятых, по потребляемым ресурсам и т. д. А ведь это только начало. Критериев, по которым выбирается операционная система, намного больше. Имеют право на существование и такие критерии, как "Она мне нравится" или "Друг себе поставил, чем я хуже?" А ведь выбор операционной системы определяет, как вы будете жить и работать ближайшие несколько лет (или десятилетий). Поэтому к выбору операционной системы следует относиться с большой тщательностью и достаточной долей скептицизма. Сжато охарактеризуем наиболее популярные операционные системы.

### **Замечание**

В 1980 году была организована инициативная группа под названием /usr/group с целью стандартизации

программного интерфейса UNIX. Стандарт был разработан к 1984 году и использовался комитетом ANSI при описании библиотек языка C. В 1985 году был создан Portable Operating System Interface for Computing Environment, сокращенно POSIX (переносимый интерфейс операционной системы для вычислительной среды). На сегодняшний день большинство операционных систем удовлетворяют (полностью или частично) стандарту POSIX.

## DOS

Производитель – Microsoft. Была создана на заре эры PC. 16-разрядная однопользовательская, однозадачная система. Платная, закрытый исходный код. Одноплатформенная (PC-совместимая). Текстовый интерфейс (командная строка). Достаточно простой процесс установки. Файловая система – FAT 12, FAT 16, FAT 32. Поддержка локальной сети – драйверы и программы сторонних производителей. Интернет – аналогично. Морально устарела еще до рождения. Разрабатывалась поспешно, без учета развития аппаратных и программных средств. При появлении нового аппаратного обеспечения срочно дорабатывалась, более или менее устойчивая система появилась только к 1990-му году (по прошествии 10 лет!). Последняя самостоятельная версия (как отдельного продукта) – MS-DOS 6.22. Последние версии включали в основном косметические изменения. Но (благодаря

дружбе IBM и Microsoft) – получила широчайшее распространение, имела малую потребность в ресурсах, поддерживала практически все выпускаемое для PC аппаратное обеспечение, и для своей ниши была почти оптимальным решением с точки зрения цена/ресурсы/производительность. Для DOS было выпущено несметное количество приложений, последние пять лет новые программные продукты для DOS не выпускаются. Существовали многочисленные клоны. Наиболее известные – IBM-DOS, PC-DOS, DR-DOS.

## **Windows 3.1x**

Строго говоря – это даже не операционная система, а графическая оболочка или, если хотите, надстройка над DOS. Почему версии 3.1x? Потому что, по большому счету, только к версии 3.1 была достигнута достаточная стабильность, неплохая функциональность, накоплена критическая масса приложений и реализована многозадачность. Благодаря Windows в мир PC был внесен дух однообразия и предсказуемости. Интерфейс приложений стал стандартизирован, системные библиотеки унифицированы, внедрена поддержка локальной сети, появился достаточно большой набор драйверов для аппаратного обеспечения, поставляемых в дистрибутиве Windows. В комплект Windows вошли несколько десятков приложений, благодаря чему потребитель получал почти готовую систему для офисной работы. Доступ

В Интернет по-прежнему осуществлялся с помощью программ сторонних производителей. Система получилась не очень надежная, достаточно часто происходили зависания. Для Windows 3.1x было выпущено много разнообразного программного обеспечения, впрочем, после 1996 года новые программные продукты для Windows 3.1x больше практически не выпускались. Достаточно простой процесс установки. Слабая поддержка мультимедийных устройств. Платная. Исходный код недоступен. Последняя версия (16 битная) Windows for Workgroups 3.11. Требования к аппаратному обеспечению – процессор 386, не менее 2 (лучше 4) Мбайт оперативной памяти, не менее 16 Мбайт места на жестком диске.

## OS/2

Совместная разработка IBM и Microsoft. Впоследствии IBM и Microsoft разошлись, и OS/2 заканчивала и выпускала в свет только IBM, которая всегда позиционировала свою операционную систему для корпоративного использования. Из-за слабой маркетинговой политики IBM, сильной конкуренции со стороны Microsoft, относительно малого количества разработанных под систему программных продуктов IBM отказалась от дальнейшего продвижения OS/2. Для своего времени (а это расцвет DOS и Windows 3.1x) была очень прогрессивна. В дальнейшем (версии 2–4) получи-

ла существенное расширение функциональности. 32-битная многозадачная операционная система. Отличная поддержка локальной сети и Интернета. Помимо программ, разработанных специально для OS/2, система позволяет запускать как DOS-приложения, так и приложения Windows 3.1x. Разрабатывается приложение Odin, позволяющее выполнять приложения Windows 9x. Файловая система – HPFS (High Performance File System). Поддерживает также достаточно большой список файловых систем, в том числе FAT 12, FAT 16, FAT 32. Требования к аппаратному обеспечению – процессор Pentium, не менее 32 Мбайт оперативной памяти, не менее 120 Мбайт места на жестком диске.

## **Windows 9x**

32-битная операционная система с поддержкой 16-битных приложений (Windows 3.1x). Первая версия появилась в 1995 году. Затем в течение почти трех лет выходили исправления многочисленных ошибок. Попытка Microsoft объединить лебедя, рака и щуку – в одной операционной системе без проблем запускать исполняемый код DOS, Windows 3.1x (16 бит) и Windows 9x (32 бит). В результате получилась не очень устойчивая система, которую удалось отладить только к 1998 году (Windows 98). Несмотря на то, что Windows 9x объявлена операционной системой, она по-прежнему базируется на MS-DOS версии 7.0 со всеми ее рудиментами и

наследственностью. В Windows 9x добавлено программное обеспечение, необходимое для работы в Интернете, и существенно расширен список драйверов для аппаратного обеспечения.

Windows 98 и последняя версия, Windows ME (Millennium Edition), фирмой Microsoft рассматриваются как переходный этап к Windows NT (Windows 2000), поэтому в Windows 9x частично включено программное обеспечение, характерное для серверов. Простой процесс установки и конфигурации системы, но отсутствует возможность тонкой ее настройки без стороннего программного обеспечения. Хорошая поддержка мультимедийных устройств и игр. Огромнейший список разработанного под систему программного обеспечения. На сегодняшний день Windows 9x и MS Office являются стандартом de-facto для настольных офисных систем. Платная. Исходный код недоступен. Требования к аппаратному обеспечению – процессор Pentium-166, не менее 32 Мбайт оперативной памяти, не менее 250 Мбайт места на жестком диске (для Windows ME не менее 64 Мбайт оперативной памяти, не менее 500 Мбайт места на жестком диске). Файловая система – FAT 32 (FAT 16).

## **Windows NT (Windows 2000)**

Ощущая бесперспективность развития ветки DOS—Windows 3.1x Microsoft разработала новую операционную

систему Windows NT (New Technology), базирующуюся на стандарте POSIX и новой файловой системе – NTFS (New Technology File System). Разработчики Windows NT серьезно взялись за проектирование операционной системы с учетом ее дальнейшего развития (совсем недавно вышла последняя в этой линейке операционная система – Windows XP). На сегодняшний день Windows NT – флагманская операционная система фирмы Microsoft. Существует две ее версии – Windows NT Workstation и Windows NT Server. Первая версия предназначена для настольных систем, вторая – серверная платформа. В обеих версиях используется графический интерфейс, что для сервера – в общем-то излишество и нерациональная трата ресурсов. Многозадачная, многопользовательская, одноплатформенная (PC), весьма устойчивая операционная система. Имеет встроенную поддержку многопроцессорных и кластерных систем. Хорошая поддержка мультимедийных устройств и игр (однако не все существующие игры надежно работают под Windows NT). Полного набора серверных приложений не имеет (нет, например, Web-сервера). Исходный код недоступен. Платная. Windows NT Server рассчитан на малые и средние рабочие группы, на большие нагрузки пока не рассчитан. Помимо NTFS поддерживает файловые системы FAT и OS/2 (HPFS). Требования к аппаратному обеспечению Windows NT Workstation – процессор Pentium, не менее 128 Мбайт оперативной памяти, не менее 500 Мбайт места на жестком диске; Windows NT



Server – процессор Pentium, не менее 256 Мбайт оперативной памяти, не менее 500 Мбайт места на жестком диске.

## **Mac OS**

Производитель – Apple. Операционная система для Macintosh и его пользователей, многозадачная, однопользовательская, графическая. Первая версия была выпущена более 15 лет назад. С выходом Mac OS X считается окончательно устаревшей и постепенно сходит со сцены. Великолепная система для пользователя по эргономичности, дружелюбности и простоте освоения. За свою историю накопила достаточно большое количество устаревших концепций, оставленных для совместимости. Последние версии Mac OS были рассчитаны на компьютеры Apple, базирующиеся на процессоре Motorola 68040 и PowerPC с не менее 16 Мбайт оперативной памяти.

## **Mac OS X**

Производитель – Apple. Новая операционная система для компьютеров Macintosh, базирующихся на процессоре PowerPC. UNIX-подобная, POSIX-совместимая, многозадачная операционная система с графическим интерфейсом. Способна выступать как в качестве сервера, так и в качестве клиентской операционной системы. Платная, закрытый ис-

ходный код. Фирма Apple заявила о скором переносе Mac OS X на платформу Intel. Поддерживает сетевую файловую систему NFS (Network File System). Mac OS X предназначена для работы на компьютерах Power Macintosh с процессором G3. Требования – не менее 128 Мбайт оперативной памяти. Рекомендуемый объем жесткого диска – 5 Гбайт.

## **Семейство UNIX**

Группа операционных систем, имеющих общего предка и традиционно носящих название UNIX. Фирмы производители – AT&T, DEC, Sun, Hewlett-Packard, IBM, SCO и многие другие. Несмотря на то что первая версия UNIX была выпущена еще тридцать лет назад, UNIX до сих пор считается наиболее современной, надежной и динамично развивающейся операционной системой. Большой вклад в успех UNIX внесли специалисты AT&T, студенты и преподаватели университета Беркли. На сегодняшний день UNIX той или иной фирмы-производителя установлен практически на каждом сервере уровня предприятия, больших кластерах и мультипроцессорных системах, а также на многих рабочих и графических станциях. Многоплатформенная, мультизадачная, многопользовательская операционная система. Поддерживает кластеризацию, мультипроцессорные системы, распределенные вычислительные среды, массивы накопителей огромной емкости и многое другое. На сегодняшний день

трудно найти компьютер, на котором не смогла бы работать одна из версий UNIX. Как правило, UNIX, выпускаемая фирмами, – платная, с закрытым исходным кодом. Тем не менее, существует достаточно много (например, семейство BSD, Linux) бесплатных, с открытым исходным кодом UNIX-совместимых операционных систем. Благодаря стандарту POSIX практически любое приложение можно перенести из одного представителя семейства UNIX в другой. Благодаря этому для UNIX имеется огромное количество как бесплатных, так и коммерческих программ. Как правило, для каждой разновидности UNIX разработана своя файловая система, но все разновидности UNIX могут работать с распространенными файловыми системами. В том числе существуют и т. н. *журналируемые* файловые системы. В журналируемых файловых системах для решения проблемы повреждения структуры файловой системы или хранения данных применяют транзакции, используемые практически в любой базе данных. Транзакция считается незавершенной до тех пор, пока все изменения не сохранены на диске. А чтобы сбои, происходящие до завершения всех операций, входящих в транзакцию, не приводили к необратимым последствиям, все действия и все изменяемые данные протоколируются. В том случае, если все-таки сбой произойдет, по протоколу можно вернуть систему в рабочее состояние. Требования к аппаратной платформе – самые разнообразные. Как уже упоминалось выше – трудно найти компьютер,

на котором не смогла бы работать одна из версий UNIX.

## **FreeBSD, OpenBSD, NetBSD**

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

- FreeBSD – очень надежная, достаточно консервативная (в хорошем смысле этого слова). Аппаратная платформа – Intel;
- NetBSD – переносимость на большое количество аппаратных платформ;
- OpenBSD – попытка объединить достоинства FreeBSD и NetBSD в одном дистрибутиве.

Бесплатные, открытый исходный код. На сегодняшний день наибольшее распространение из-за своей особой надежности получила FreeBSD. Двоичная совместимость со многими программами, построенными под SCO, BSD/OS, Net/Free/OpenBSD, 386BSD и Linux.

## **Linux**

POSIX-совместимая UNIX-подобная операционная система. На сегодняшний день – самая распространенная бесплатная операционная система с открытым исходным кодом. При ее разработке из мира семейства UNIX старались взять

все лучшее. Благодаря участию десятков тысяч разработчиков программного обеспечения и координации их действий через Интернет Linux и программное обеспечение для нее развивается очень динамично, ошибки и различные проблемы в программном обеспечении, как правило, исправляются в считанные часы после их обнаружения. Большую помощь в развитии и распространении Linux и сопутствующего ему программного обеспечения оказали фонд Свободного программного обеспечения (Free Software Foundation, USA) и лицензия GNU (The GNU General Public License, Универсальная общественная лицензия GNU) для программного обеспечения. На сегодняшний день существует одно ядро Linux, разработку которого координируют его создатели Линус Торвалдс и Алан Кокс, и множество дистрибутивов (не менее 2–3 десятков), отличающихся как функциональным назначением, так и составом программного обеспечения, входящим в дистрибутив. Существуют дистрибутивы, занимающие десяток компакт-дисков, и дистрибутивы, уместающиеся на одной-двух дискетах. Все, что справедливо для семейства UNIX – справедливо и для Linux. Широчайшая поддержка аппаратных платформ, малая требовательность к аппаратным ресурсам (процессор 486, 8 Мбайт оперативной памяти, винчестер 120 Мбайт). Масштабируемость, поддержка мультипроцессорных систем, кластеризация, поддержка распределенных вычислений, десятки графических оболочек и это далеко не все. Поддерживаются де-

сятки файловых систем, родная файловая система Ext2. И при всей мощи – достаточно дружелюбная операционная система, способная работать как на мощнейшем сервере, так и на стареньком «пентиуме» где-нибудь в офисе.

## BeOS

Производитель – Be Inc. UNIX-подобная графическая операционная система. Однопользовательская. Сами разработчики позиционируют BeOS как операционную систему для работы с мультимедиа. Графический интерфейс. Очень молодая операционная система (по сравнению с ранее рассмотренными), поэтому относительно небольшой список поддерживаемого оборудования и программного обеспечения. BeOS поддерживает компьютеры с симметричной мультипроцессорной архитектурой (SMP) (до 8-ми процессоров), файловая система BFS (Be File System), 64-битная и журналируемая. Вытесняющая многозадачность, почти полная POSIX-совместимость. На сегодняшний день существуют две версии операционной системы – BeOS 5 PE (Personal Edition) – бесплатная (дистрибутив можно взять в Интернете, [www.be.com](http://www.be.com)) и BeOS 5 Pro – платная (поставляется на CD-ROM в коробке и с документацией). Исходный код недоступен (несколько программ, входящих в BeOS, имеют открытый исходный код). Поддерживает работу с несколькими файловыми системами – FAT 16, FAT 32 (возможно чтение

ние и запись), NTFS (только чтение), HFS, ext2, CDFS. Аппаратные требования – Pentium-133, оперативная память – 32 Мбайт, место на винчестере – 512 Мбайт (минимально – 120 Мбайт). Возможен вариант установки под Windows и Linux. Аппаратная платформа – PC, PowerPC.

## QNX

Производитель QNX – QNX Software Systems. UNIX-подобная POSIX-совместимая операционная система реального времени. 32-битная, многозадачная, многопользовательская, микроядерная. Первоначальное предназначение – промышленная операционная система, предназначенная для работы в режиме 99,999 % надежности («пять девяток»). Используется для управления технологическими процессами, начиная от атомных электростанций и заканчивая производством мороженого. Исходный код закрыт. Проблемы с драйверами (малое количество). Минимальные требования для промышленного дистрибутива – 386-й процессор, 8 Мбайт ОЗУ. Помимо промышленных дистрибутивов QNX, стоящих немалые деньги, существует бесплатный вариант дистрибутива «QNX Real Time Platform», который загружается с сайта производителя ([www.qnx.com](http://www.qnx.com)). Минимальные требования для бесплатного дистрибутива – процессор Pentium-200, 32 Мбайт ОЗУ, 100 Мбайт на жестком диске.

Ознакомившись с вышеприведенным кратким обзором

операционных систем, можно представить в общих чертах их области применения, достоинства и недостатки. Поскольку наша книга посвящена Linux, а операционные системы Windows 9x или Windows NT/2000 установлены приблизительно на 90 % PC-совместимых персональных компьютеров, то все сравнения в дальнейшем мы будем проводить относительно этих трех операционных систем, не забывая, впрочем, и об остальных.

### **Небольшое отступление**

Что такое пользователь? Никогда не задумывались? Понятие «пользователь» не подведешь под «среднестатистическое» значение. Он многолик и разнообразен. Единственное, что есть общего у всех пользователей компьютера – они сидят за компьютером. Пользователи с точки зрения системного администратора – все те, кто входит в систему в качестве пользователя, «юзера». С точки зрения системного программиста – все, кто запускает компьютер. Для разработчика прикладного программного обеспечения – пользователи его программы. Для авторов книг "... для чайников" – это люди, знающие о компьютере только то, что у него есть шнур питания и какая-то доска с кнопками. И так далее. Если попытаться обобщить, основной пользователь – это человек, который не разбирается в устройстве компьютеров, не знает, как настроить модем, не обязан знать тонкости операционной системы и т. п. Пользователь решает на компьютере свои



профессиональные задачи, зачастую не имеющие с компьютерами ничего общего. На практике все это, конечно, не совсем так мрачно. Пользователь для успешной работы просто обязан знать, что такое файл, как настроить рабочий стол, установить программу, что такое вирусы и как с ними бороться и т. д. Пользователей условно можно разделить на три группы – не знающий о компьютере ничего, знающий кое-что и знающий многое. Соответственно, по уровням пользователей можно разделить операционные системы на три категории.

К первой категории можно отнести Mac OS и, в какой-то степени, Mac OS X, а также BeOS. Ко второй категории Windows 9x, OS/2. К третьей, как ни странно, – DOS, Windows 3.1x, Windows NT/2000, UNIX-семейство, BSD-семейство, Linux, QNX. Такое разбиение операционных систем не всегда соответствует официальному позиционированию фирм-разработчиков (например, Microsoft рекламирует Windows 9x как систему для домохозяек – включил и работай). Однако с точки зрения коллективного разума (по крайней мере, так считают авторы новостных конференций, посвященных сравнительному обзору операционных систем) данное нами разбиение операционных систем достаточно верно. Впрочем, жизнь, как всегда, не стоит на месте. Сейчас уже можно говорить, что Linux с ее графическими менеджерами окон KDE и GNOME постепенно переходит, если уже не перешла, ко второй категории (то есть для пользователей,

знающих об операционной системе кое-что), при этом не теряя ни мощности, ни настраиваемости всего и вся. Семейство Windows постепенно сдвигается к группе пользователей, не знающих об операционной системе ничего, при этом вызывая заметное раздражение знающих, или, как у них говорят – Advanced Users, своей уверенностью, что пользователь системе приносит только вред, а посему ничего настраивать он не должен, а если очень хочет – пусть платит за поддержку или специальное программное обеспечение. В идеале же операционная система должна удовлетворять, по меньшей мере, семи достаточно противоречивым требованиям.

1. Быть легкой в освоении и дружелюбной к пользователю (User Friendly).
2. Быть очень мощной и универсальной (способной работать на любом оборудовании).
3. В ней все должно настраиваться и достаточно просто.
4. Она должна быть очень надежна (в идеале – сверхнадежна).
5. Занимать как можно меньше места.
6. Разработчики моментально должны реагировать на проблемы, обнаруженные в процессе эксплуатации.
7. Под нее должен быть широкий выбор программного обеспечения.

В нескольких словах рассмотрим эти семь пунктов. Пункт *первый*. Тут, собственно, и так все ясно. От того, как быстро человек освоится с операционной системой и насколько

ко удобно ему в ней работать, напрямую зависит производительность труда, да и просто хорошее настроение. Пункт *второй*. Можно, конечно, возразить, что чем более универсальный инструмент, тем слабее он для какого-нибудь специфического применения, и чисто теоретически это так. Но давайте посмотрим на универсальность с другой стороны. Теоретические принципы построения операционной среды, по большому счету, одинаковы, что для старенькой 386-й, что для новейших мультипроцессорных систем. Специфику платформы (тип процессора, мультипроцессорность, кластеризацию и т. п.) всегда можно учесть при разработке специфического ядра операционной системы или драйверов. Некоторая потеря в производительности с лихвой окупается тем, что пользователю, поработавшему на мощнейшем сервере и перешедшему на офисный компьютер, графическую станцию или домашний ПК, не придется осваивать другую операционную систему – его операционная система может работать на любом компьютере. А способность работать на любом компьютере автоматически подразумевает, что операционная система должна занимать как можно меньше места и потреблять мало аппаратных ресурсов. Пункт *третий*. И тут все понятно без пространных пояснений. Пользователь должен иметь возможность настроить операционную систему под свои нужды, не прибегая к стороннему (не входящему в поставку операционной системы) программному обеспечению. Пункт *четвертый*. Правда, большое место?

У любого пользователя Windows со стажем наверняка происходило зависание компьютера, причем в самое неподходящее время. И каждый пользователь хочет, чтобы зависания никогда не происходили на его компьютере. Пункт *пятый*. Это тоже понятно. Уже надоело каждые год-полтора менять жесткий диск только из-за того, что следующая версия операционной системы требует «совсем немного, только каких-то 3 Гбайта места на жестком диске». Пункт *шестой*. И это очевидно. Пользователь должен получить исправления к своей операционной системе при обнаружении просчетов ее разработчиков. Причем, как можно скорее, если операционная система удовлетворяет п. 4. И притом абсолютно бесплатно, поскольку это просчет разработчика. Пункт *седьмой*. Пусть операционная система будет самой распрекрасной, но если для нее нет программ, она не будет востребована.

Теперь оценим операционные системы на соответствие вышеперечисленным требованиям.

- DOS – не удовлетворяет ни одному пункту, кроме п. 7.
- Windows 3.1x – удовлетворяет п. 1 с оговорками, частично п. 3 и п. 5, удовлетворяет п. 7.
- OS/2 – удовлетворяет п. 1, п. 2 (с учетом одноплатформенности), п. 3, частично п. 4, п. 5 и п. 7.
- Windows 9x – удовлетворяет п. 1, частично п. 3, безусловно удовлетворяет п. 7.
- Windows NT (Windows 2000) – удовлетворяет п. 1, п. 2 (с учетом одноплатформенности и непомерных требований

к аппаратному обеспечению), п. 3 и п. 4 с оговорками, безусловно удовлетворяет п. 7.

- Мае ОС – безусловно удовлетворяет п. 1, п. 2 (с учетом одноплатформенности), частично п. 3, п. 4, п. 5, п. 6, удовлетворяет п. 7.

- Мае ОС X – безусловно удовлетворяет п. 1, п. 2 (с учетом одноплатформенности и завышенных требований к аппаратному обеспечению), п. 3, п. 4, п. 6, пока не удовлетворяет п. 7.

- UNIX-семейство – безусловно удовлетворяет всем пунктам, кроме первого, да и то, в последнее время легкость освоения и дружелюбность у UNIX-разработчиков стоят на первом месте.

- FreeBSD, OpenBSD, NetBSD – все сказанное о UNIX-семействе справедливо и для этих операционных систем.

- Linux – безусловно удовлетворяет всем пунктам, особенно п. 2, п. 3, п. 6, п. 7.

- BeOS – удовлетворяет всем пунктам кроме (пока) п. 7.

- QNX – удовлетворяет всем пунктам.

Попробуем выбрать операционную систему, исходя из вышеперечисленных пунктов. DOS и Windows 3.1x отпадают сразу, как морально и физически устаревшие продукты. OS/2 – очень неплохая операционная система, имеющая несколько недостатков: отсутствие перспектив (IBM отказалась от выпуска следующих версий), не очень большой выбор программного обеспечения, одноплатформенность. Mac

OS, Mac OS X – также неплохие операционные среды как с точки зрения пользователя, так и с точки зрения администратора. Но – это операционные системы *только* для компьютеров фирмы Apple. А в нашей стране этих компьютеров не наберется и одного процента от общего количества персональных ЭВМ. QNX – достаточно специфичная система, рассчитанная для применения в сверхнадежных системах реального времени. Очень хорошая, но для нашего пользователя она стала доступна относительно недавно, поэтому в отношении к ней есть элементы недоверия и незнания, кроме того, у нее относительно малый список программного обеспечения общего назначения (офисные приложения, работа с графикой, игры, наконец). Что остается – семейство Windows 9х– Windows NT (включая Windows XP), семейство UNIX, а также представители «свободного мира» UNIX – FreeBSD, OpenBSD, NetBSD, Linux и стоящая немного особняком BeOS.

Теперь попытаемся максимально корректно сопоставить Windows-семейство и семейство UNIX. Сначала проведем четкий водораздел между операционными системами Windows 9х/ME и Windows NT/2000/XP. Подсознательно (в силу сходства названий, да и внешнего вида) пользователь, а иногда и администратор, отождествляет Windows 9х/ME и Windows NT/2000, хотя это далеко не одно и то же. Если внимательно посмотреть на характеристики Windows 9х/ME и немного сопоставить факты, станет понятно, что

Windows 9x/ME – это затянувшийся на шесть лет переход от DOS/Windows 3.1x к Windows NT/2000, принесший, однако Microsoft огромный доход. С чисто технологической стороны UNIX-семейство корректно сравнивать только с Windows NT/2000, поскольку только Windows NT/2000, как система истинно многозадачная и многопользовательская, поддерживающая мультипроцессорность и кластеризацию, корректно сопоставима с UNIX-подобными системами. Не надо сразу обижаться за любимую Windows 9x/ME. Если отбросить эмоции и посмотреть правде в глаза, Windows 9x/ME – операционная система для домохозяек и игрушек, которая благодаря простоте освоения методом "научного тыка" незаконно, или правильней сказать безосновательно (с точки зрения системного администратора, безопасности, надежности, да и по целому ряду других причин), проникла в корпоративный мир. Это, конечно, понятно – пользователи знают Windows 9x/ME (дома стоит, в институте на ней работали) и пытаются опробованное решение применить в масштабах предприятия. Когда предприятие это 3–5 машин в локальной сети – еще терпимо, но когда на предприятии хотя бы пара десятков компьютеров, и, не дай бог, несколько сетей, которые надо объединить – администратор начинает жить на работе. Вирусы, зависания машин, непонятные эффекты, отсутствие в операционной системе необходимого программного обеспечения для администрирования и управления сетями и рабочими местами – вот сотая часть того, с чем стал-

кивается администратор. А если в сети появляется сервер... Windows 9x/ME уже ничто не поможет. Ведь сервер – это и RAID-массив, и несколько процессоров, причем необязательно это процессоры от Intel. Windows 9x/ME на такую машину не поставишь.

Таким образом, корректно сравнивать (с учетом приведенных выше требований) можно только семейство UNIX и Windows NT/2000. Относительно Windows NT/2000 существует основанное на схожести интерфейса и названия с Windows 9x/ME заблуждение, что настроить Windows NT/2000 дело пяти минут, и после настройки все работает годами без вмешательства администратора. Внешнее сходство этих систем с Windows 9x/ME создает обманчивую иллюзию понимания там, где им и не пахнет, а увеличение нагрузки на сервер заставляет остро чувствовать программистскую поговорку "Памяти мало никогда не бывает". Поскольку книга эта об Linux, сравним семейство Windows с Linux.

Во-первых, что очень выгодно отличает Linux от Windows – ее *бесплатность*. За Windows 9x/ME по сегодняшним ценам придется уплатить около 150–200 долларов, а за Windows NT/2000 и того больше. Кроме того, для работы нужен и Microsoft Office, за стандартный вариант которого придется уплатить около 600 долларов, и, если надо еще что-то – продолжать платить и платить. Сегодня никого не удивляет, когда стоимость установленного программного обеспечения больше, чем стоимость самого компьюте-



ра. А если у вас несколько компьютеров – умножьте затраты на их количество. Вот и получается, что маленькая фирма с 5-ю компьютерами потратит 7–10 тыс. долларов только на программное обеспечение. Но это только начало. Политика Microsoft очень проста и действенна – раз в полгода-год выходит новая версия программного продукта, который все вольно или невольно вынуждены покупать, потому что партнеры присылают вам файлы в формате Excel 97, а ваш Excel 95 отказывается их понимать. В результате за всю жизнь компьютера (3–5 лет) только на программное обеспечение придется потратить порядка 2–5 тыс. долларов. С другой стороны, Linux обойдется в 5–15 долларов, за которые можно купить 2–3 компакт-диска, заполненных бесплатным, с открытым исходным кодом, программным обеспечением. Даже если скачивать дистрибутив Linux через Интернет – все равно не потратить больше 30 долларов (приблизительно столько стоит месяц неограниченного подключения к Интернету). И что характерно – с этого дистрибутива можно сколько угодно раз установить Linux *на абсолютно законных основаниях*. Можно возразить, что за потраченные на продукты Microsoft деньги пользователи получают поддержку сервис-центра Microsoft. Увы, это не так – нормальной поддержки на территории СНГ до недавнего времени не было, а звонить сейчас в Москву, например, из Беларуси или Владивостока и получать телефонную консультацию в течение пятнадцати-двадцати минут весьма накладно. Под-

держку же для Linux и ее программного обеспечения получить очень просто, нужно только знать, куда обращаться. Поскольку Linux – дитя Интернета, решение проблем надо искать там. Помимо Интернета, где находятся тысячи Web-сайтов, посвященных как Linux в целом, так и конкретному программному продукту для нее, существуют десятки групп новостей, а, помимо всего прочего, в дистрибутив входит более 15 тыс. страниц документации, описывающих все и вся. Есть правда одно неудобство – поскольку Linux разрабатывается и сопровождается людьми со всех стран мира, то и документация для него, в основном, на английском языке. Впрочем, это небольшая плата за обладание практически бесплатным программным обеспечением. Тем не менее, существует достаточно большой пласт литературы и на русском языке.

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

По поводу дружелюбности, легкости в освоении и инсталляции. На сегодняшний день *установить Linux на абсолютно чистый диск сможет любой пользователь*, для этого

нужно только взять соответствующий дистрибутив. Например, дистрибутив Red Hat Linux 7.1 все сделает сам (если, конечно, это нужно) – самостоятельно разобьет и отформатирует жесткий диск, настроит нужную раскладку языка и интерфейс (богатейший выбор из более чем ста языков: русский, украинский, белорусский в том числе), определит аппаратное обеспечение компьютера и настроит его на максимальную производительность. Установит необходимое программное обеспечение в зависимости от выбранного профиля компьютера (сервер, рабочая станция, ноутбук или выборочная установка), при этом ни в коей мере не ограничивая владельца в самостоятельной конфигурации. Что примечательно, установку Linux можно производить в текстовом (обычно так поступают опытные пользователи, или если слабая машина) или в графическом интерфейсе, с CD-ROM, жесткого диска или даже по сети, загрузив компьютер со специально изготовленной дискеты. При установке можно указать Linux при старте сразу загружать графическую оболочку. Поэтому миф о сложности установки можно считать не соответствующим действительности. С легкостью освоения, несомненно, похуже. Для грамотной работы в Linux необходимо иметь представление об операционной системе. К сожалению, Windows приучила пользователя щелкать мышкой и не думать. Плюс еще наш менталитет – «сами с усами», метод «тыка». В UNIX это не проходит. Там подход другой – прочитай, разберись и можешь

быть уверен, что это функционирует в любой UNIX-подобной системе одним и тем же способом. Еще нюанс – документация для Linux пишется в расчете на грамотного, способного размышлять человека. Это, разумеется, отпугивает пользователя, привыкшего руководствоваться инструкцией-комиксом, и порождает очередной миф о недружелюбности Linux. Однако приятно, что творцы документации считают тебя умным человеком, а не семилетним ребенком.

*По части настройки операционной системы.* Microsoft внедрила в свою операционную систему непродуманную идею – системный реестр. В результате получился монстрообразный (зачастую в 4–5 Мбайт) файл двоичного формата, от целостности которого зависит жизнеспособность операционной системы. Очевидно разработчики совсем забыли старое изречение «Не клади все яйца в одну корзину». Очень часто (по меньшей мере, в 30–40 % случаев) ошибки функционирования операционной системы связаны с повреждением файла реестра. Еще одна проблема настраиваемости системы – очень много настроек Windows не описаны в документации, и необходимо перерывать горы литературы, чтобы по крохам собирать информацию о тонкой настройке системы. Есть, конечно, программное обеспечение, позволяющее тонко настроить Windows, но, как правило, оно не бесплатно. В Linux все более надежно и доступно. Практически все о настройке системы или программного обеспечения можно узнать из документации. Конфигурационные

файлы обычно для каждой программы отдельные, и практически все имеют понятный текстовый формат с подробными комментариями. А настроить в Linux можно все, причем для каждого пользователя в системе отдельно.

*О надежности.* Конечно, семейство Windows – это не Windows 3.1x и даже не Windows 95, для которых ни дня не проходило без сбоя, но до надежности и живучести Linux (не говоря уже о проверенных десятилетиями UNIX) Windows еще далеко. О чем говорить, если во время демонстрации Биллом Гейтсом новых возможностей Windows 2000 (флагманского продукта!) операционная система дала сбой, выдав при этом на монитор «синий экран смерти».

*Занимаемое место* – в настоящее время, наверное, уже не совсем актуально, сколько операционная система занимает места на жестком диске – 500 Мбайт или Гбайт, но, все равно – чем меньше система, тем она быстрее и надежнее. Тут опять в лидерах Linux – ее можно установить на одну дискету 1,44 Мбайт. Вполне функциональный Интернет-сервер можно уместить в 80—150 Мбайт. С Windows 98, а уж тем более с Windows NT, такого сделать не удастся.

*Реакция разработчиков на ошибки.* До недавнего времени у Microsoft была достаточно интересная политика: извлекать прибыль даже из собственного брака. Для Windows NT было выпущено 6 сервис-паков, в которые вошли исправления тысяч мелких и крупных ошибок, и все эти сервис-паки продавались за деньги. Скорость исправления ошибок

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

Единственное, в чем Windows пока превосходит Linux – это *в количестве и разнообразии прикладного программно-*

го обеспечения. Тем не менее, в последние полтора-два года очень бурно пошел процесс переноса под Linux коммерческого программного обеспечения. Пожалуй, сейчас мало осталось направлений, для которых в Linux нет бесплатного или, на худой конец, платного программного обеспечения. Офисные программные комплексы, совместимые по форматам файлов с Microsoft Office есть, об интернет-приложениях и говорить нечего, базы данных, мультимедиа-приложения и т. д., и т. п. Конечно, есть и незанятые ниши – например, нет того изобилия приложений для многоцветной полиграфии, трехмерного моделирования и анимации, видеомонтажа или игр. Но давайте себя спросим – много ли людей занимаются видеомонтажом или анимацией? Наверное, даже не сотая часть процента компьютерных пользователей. А если на компьютере только играть – зачем, вообще, ПК? Есть ведь Sony Play Station, Microsoft Xbox.

И отдельного упоминания заслуживает *безопасность*. Нехорошо, когда чуть ли не каждую неделю по всему офису прокатывается эпидемия компьютерного вируса, который, ко всему, портит данные на жестком диске. Или кто-то удалил на вашем компьютере данные случайно. Или еще чего похуже. Ведь Windows 9x/ME не имеют абсолютно никакой политики безопасности, и навредить системе может практически каждый. На сегодняшний день для Windows существует более двадцати девяти тысяч вирусов или программ-троянских коней, причем каждый пятый несет в себе деструктивные

функции. Из-за совместимости с Windows 9x/ME Windows NT/2000 тоже не минула чаша сия. Конечно, с безопасностью в Windows NT намного лучше, но, тем не менее, ее намного чаще взламывают через сеть, чем UNIX. Без прав администратора пользователь может навредить только сам себе, не затрагивая других пользователей. В настоящее время существует около 40 (!) вирусов или троянских коней для Linux, причем реально опасных из них всего 2 или 3. И программы, через которые происходило проникновение троянских коней, давно уже избавлены от этого недостатка.

Подведем итог – почему выбирают Linux.

## **Почему выбирают Linux**

Приведем ряд аргументов.

- Самая лучшая операционная система – UNIX. Linux – это современный UNIX, работающий практически на всех платформах.
- В отличие от большинства операционных систем дистрибутивы Linux бесплатны, их можно скачивать из Интернета.
- В стандартный дистрибутив Linux входят сотни программ, с помощью которых можно решить 95 % задач, решаемых с помощью компьютера.
- Исходный код всех программ под Linux открыт, при желании его можно модифицировать так, как нужно.



- На базе Linux достаточно легко создать очень надежные (99,99 %) центры данных с поддержкой кластерных конфигураций и высокой степенью масштабирования.
- Корпоративная intranet-сеть "из коробки", элементарная установка интернет-сервисов и серверов, практически сразу настроенных для стандартного применения.
- Высокая степень безопасности и ограничения доступа к ресурсам и данным системы.
- Большое количество поддерживаемых Linux аппаратных платформ.
- Графический интерфейс с десятками оконных менеджеров, позволяющих создать эксклюзивную графическую среду, точно настроенную для нужд пользователя и аппаратных ресурсов.
- Относительно малые требования к аппаратным ресурсам, достаточно новый дистрибутив вполне можно установить на старших 486-х компьютерах.
- Огромнейшая библиотека документации, ежедневно улучшающаяся и дополняющаяся.
- Великолепная поддержка программного обеспечения, ответы практически на любой вопрос можно найти в Интернете, а на оставшиеся вопросы можно получить ответ у самих разработчиков, которые не скрываются за копирайтом большой фирмы.
- В Linux можно настроить все и вся. Простота конфигурации и подробное описание конфигурационных файлов вы-

годно отличают Linux от большинства коммерческих операционных систем.

- Можно установить Linux на одну дискету, и при этом она окажется способна выполнять функции маршрутизатора или отправлять электронную почту.
- Постоянное обновление и улучшение как ядра Linux, так и большинства программных продуктов для Linux
- Отсутствие зависимости от патентов и лицензий.

## Разные факты

Фирма Intel сотрудничает с основными разработчиками ядра Linux, а IBM вкладывает около миллиарда долларов в продвижение и поддержку Linux в своих офисах. Версия Red Hat Linux 4.1 использовалась для создания спецэффектов при съемках фильма «Титаник». Правительство Китая утвердило Linux как операционную систему для государственных структур. Кинокомпания Dream Works, создатель мультфильмов «Побег из курятника», «Принц Египта», «Шрек» и «Антс» полностью переводит весь цикл разработки мультфильмов на Linux. Компания Corel заявила о создании версий своего программного обеспечения для Linux. По заказу Национального научного фонда США (NSF) будет создан TeraGrid – самый мощный суперкомпьютер, используемый в научных целях. В TeraGrid будут применяться кластерные серверы IBM, работающие под управлением ОС

Linux и связанные между собой высокоскоростной оптической сетью Qwest.

## Ссылки

Сайты, посвященные BeOS:

- [www.benews.ru](http://www.benews.ru) – новости мира BeOS на русском языке;
- [www.bebits.com](http://www.bebits.com) – крупнейший ресурс программного обеспечения для BeOS;
- [besoft.org](http://besoft.org) – программное обеспечение и документация для BeOS.

Сайты, посвященные QNX:

- [www.qnx.com](http://www.qnx.com) – сайт фирмы QNX Software Systems, разработчика QNX;
- [qnx.boom.ru](http://qnx.boom.ru) – программы, документация, новости QNX;
- [qnxworld.main.ru](http://qnxworld.main.ru) – еще один сайт по QNX;
- [www.crosswinds.net/~kthulu/russian](http://www.crosswinds.net/~kthulu/russian) – много документации по QNX.

Сайты, посвященные FreeBSD:

- [www.freebsd.org](http://www.freebsd.org) – сайт FreeBSD;
- [www.freebsd.ru](http://www.freebsd.ru) – русскоязычный сайт.

Сайты, посвященные Linux:

- [www.linux.org.ru](http://www.linux.org.ru) – отличный сайт о Linux;
- [www.linux.org](http://www.linux.org) – сайт о Linux;
- [www.linuxdocs.org](http://www.linuxdocs.org) – много литературы о Linux;
- [www.linuxrsp.ru](http://www.linuxrsp.ru) – русскоязычный сайт;

- [www.redhat.com](http://www.redhat.com) – сайт версии Red Hat.

Сайты, посвященные Windows:

- [www.microsoft.com](http://www.microsoft.com) – официальный сайт фирмы

Microsoft;

- [www.winffles.com](http://www.winffles.com) – обширная коллекция программ для

Windows.

Сайты, посвященные Apple:

- [www.apple.com](http://www.apple.com) – официальный сайт Apple;
- [www.apple.ru](http://www.apple.ru) – русскоязычный сайт Apple.

## **Глава 2**

# **Возможности Linux**

В этой главе пойдет разговор об администраторах, офисном и домашнем применении Linux. Автор достаточно долго занимался сопровождением как программ, так и локальных сетей и компьютеров, поэтому не понаслышке знает проблемы администратора. Администратор – это человек, который во время рабочего дня ничего не делает, пьет кофе и играет в компьютерные игры. В идеале, администратору платят зарплату за то, что он бездельничает. В том смысле, что надежное и не весьма требовательное к сопровождению программное обеспечение (включая, разумеется, и операционную систему), будучи один раз правильно отлажено, должно потом долго работать, не требуя постоянного вмешательства администратора для дополнительных перенастроек, переналадок и инсталляций. Если в вашей организации это не так, вывод может быть один: либо у вас неудачное программное обеспечение, либо плохой администратор. Поэтому всех, кто отвечает за бесперебойную работу локальных сетей, весьма интересует, как проблема администрирования решена в той или иной операционной системе. И если решена она недостаточно хорошо, заставляя делать изо дня в день одно и то же, такая операционная система вызывает раздражение и желание сменить ее на более «дружелюбную». Офисное приме-

нение Linux интересует нас с точки зрения применимости ее на рабочем месте, в фирме, на предприятии. Домашнее применение тоже, разумеется, будет рассмотрено. О серверном использовании Linux в этой главе мы подробно говорить не будем, потому что применению Linux в качестве серверной ОС посвящена добрая половина книги. Помимо этого, практически все знают, что Linux и сервер – «близнецы-братья», а о применении ее в офисе или дома еще мало кто задумывался. Но начнем с азов. Как выразился один из грандов компьютерного бизнеса «Компьютер – это сеть».

## **Сеть**

### **Сетевые протоколы и аппаратура**

Linux по умолчанию работает со своим «родным» протоколом TCP/IP, протоколом, на котором функционирует Интернет. Но это вовсе не означает, что она, кроме этого протокола, ничего не понимает. При установке соответствующего программного обеспечения Linux способна также работать с протоколом IPX/SPX фирмы Novell Netware, протоколом NetBIOS (Microsoft Windows 3.1x, Windows 9x/ME, Windows NT/2000) и AppleTalk (Apple Mac OS). И это еще не все, что она понимает и поддерживает, хотя перечисленные четыре сетевых протокола сегодня используются, на-

верное, более чем в 95 % случаев. Из аппаратных сетевых средств Linux способна работать практически с любым оборудованием, предназначенным в том или ином виде для использования в сетевых соединениях: сетевые карты Ethernet, Radio Ethernet, ArcNet, аппаратура для спутникового Интернета, ISDN, ATM, обычные модемы и многое другое. Конечно, с аппаратным обеспечением не все так гладко, как хотелось бы. Не для всех устройств есть драйвера под Linux, однако, как правило, для всех распространенных устройств они есть. Были проблемы с драйверами для так называемых Win-модемов, но в последнее время решаются и они. Можно сказать, что при наличии соответствующего программного обеспечения и драйверов сетевые протоколы и аппаратура под Linux очень хорошо настраиваются с помощью текстовых конфигурационных файлов или специальными программами, например, netconf.

### **Замечание**

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

## Сетевые сервисы

О сетевых сервисах более подробно будет рассказано в пятой части книги, а сейчас – краткий обзор. Начнем с того, где зарождалась и развивалась Linux – с Интернета. Было бы удивительно, если бы дитя Интернета и представитель семейства UNIX (колыбели Интернета) не предоставлял всей полноты интернет-сервисов. Что интересует пользователя в Интернете? На первый, поверхностный, взгляд Web-сайты, FTP, электронная почта и новости. Но для нормального (и комфортного) функционирования Интернета необходимо множество других сервисов – это и DNS, и прокси-серверы, и серверы точного времени и многое-многое другое. Все это для Linux есть, и не в единственном экземпляре – нужно только выбрать, какой «тяжести» инструмент необходим. Сказанное касается и серверного программного обеспечения, и клиентского. Так, например, Web-браузеров существует более десяти: Lynx – текстовый браузер, Netscape Navigator, Mozilla, Opera, Konqueror и др.





Рис. 2.1. Web-браузер Mozilla

Почтовых клиентов также существует несколько десятков – как текстовых, так и графических: Pine, Netscape-клиент, Kmail, Evolution и т. д.

Можно рассматривать любой интернет-сервис, и всегда в список клиентских приложений для этого сервиса войдет не менее десятка программ. Если необходим файл-сервер – тоже есть большой выбор. Можно пользоваться «родным» NFS, можно Mars – файл-сервером для сетей Netware, можно Samba – файл-сервером для сетей Microsoft. Для всех

упомянутых типов файловых серверов, конечно же, есть и клиентское программное обеспечение. При желании можно создать сетевой компьютер – с отсутствующим жестким диском, без каких-либо накопителей, загружающийся через сеть и нормально функционирующий (причем, с графической оболочкой). Решены для Linux и вопросы статистики. Множество пакетов могут собрать, обработать, представить в текстовом и графическом виде информацию о любой стороне функционирования Linux, в частности, о загрузке сети, входящем и исходящем трафике, построить диаграммы, отобразить их на Web-странице и, если необходимо, адекватно отреагировать на какое-то отклонение в функционировании сети. Настройку множества сервисов можно произвести или с помощью специальных программ – например, `linuxconf`, или отредактировав конфигурационные файлы. У большинства сервисов есть еще одна возможность – настройка через Web-интерфейс. Существуют и совместимые с ICQ интернет-пейджеры: `licq`, `kicq`, `GNOMEICQ`, `micq`, в том числе, и для текстовой консоли.

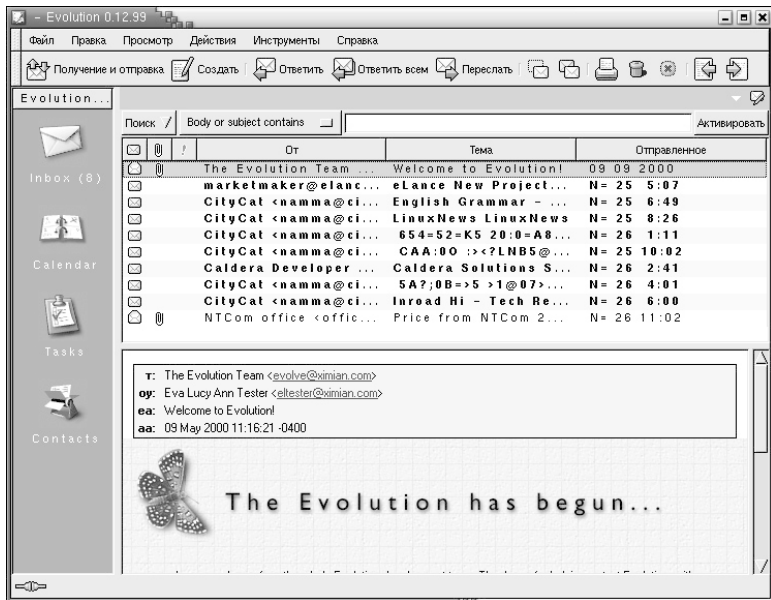


Рис. 2.2. Почтовый клиент Evolution, по совместительству – органайзер

## Файловые менеджеры

Для пользователей старой закалки, знакомых еще с DOS, неперменным атрибутом работы за компьютером был файловый менеджер, который подменял собой скуку командной строки и черноту экрана. Хотя адепты Linux упорно твердят о полной ненужности файлового менеджера для Linux, тем не менее, спрос порождает предложение. Есть

несколько файловых менеджеров и для нашей операционной системы. Как обычно, есть они и для текстовой консоли, и для X Window. Самый известный и, наверное, один из старейших текстовых файловых менеджеров – Midnight Commander (почти полный эквивалент Norton Commander).

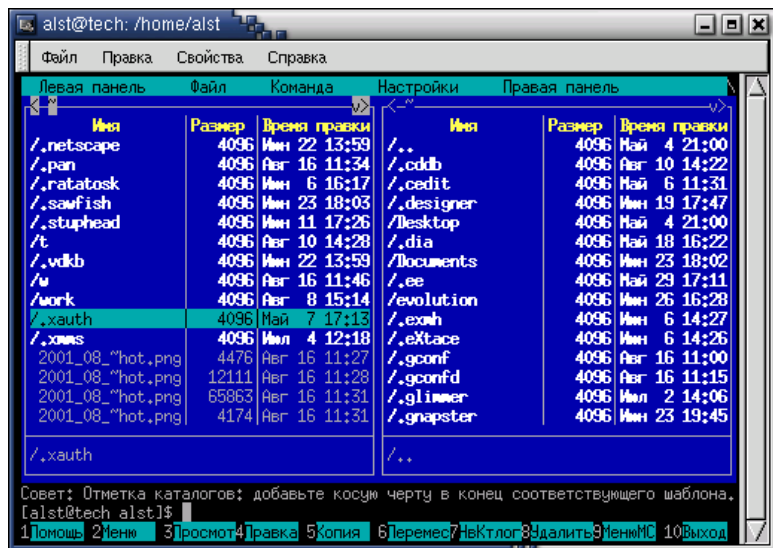


Рис. 2.3. Файловый менеджер Midnight Commander

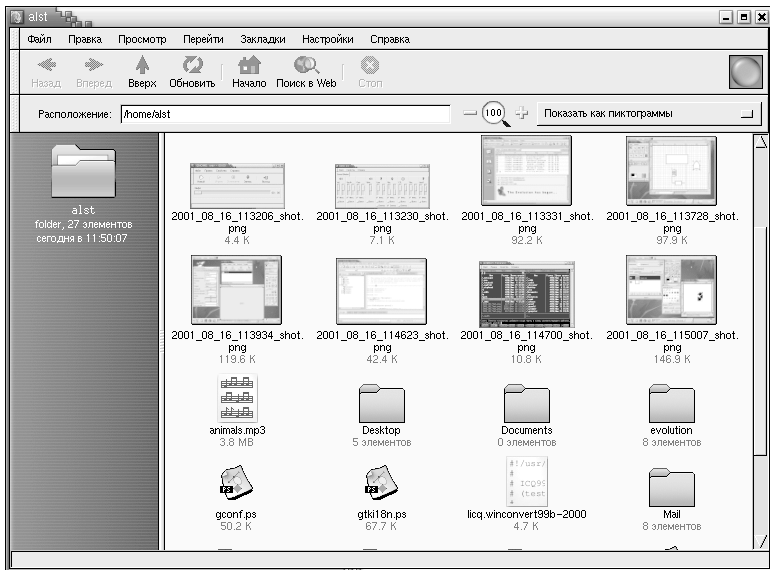


Рис. 2.4. Один из графических файловых менеджеров

Позволяет работать с файлами, редактировать их, осуществлять поиск, работать с архивами и т. д. Однако есть еще и другие достоинства – способность устанавливать пакеты RPM, работать с FTP, просматривать HTML-документы, подключаться к сетевым дискам. Приблизительно такого же плана текстовый файловый менеджер XNC. Помимо текстовых, достаточно много и графических файловых менеджеров для X Window, например, Kcommander или Kruiser.

# Текстовые редакторы

Тут выбор широчайший – от простейшего строчного текстового редактора до пакетов, которые текстовым редактором и назвать трудно. И такое разнообразие наблюдается и для текстовой консоли, и для X Window. Конечно, в сегодняшний век торжества «графики» многие удивляются наличию большого числа текстовых консольных редакторов. Однако надо вспомнить о широкой распространенности Linux, в том числе и на не очень мощных машинах, куда не имеет смысла устанавливать объемную графическую оболочку и не менее объемный графический текстовый редактор только для того, чтобы откорректировать несколько конфигурационных файлов. Не исключена также необходимость отредактировать тот или иной Web-скрипт на удаленной машине через Интернет. Или совсем неприятный вариант – сбой системы, не позволяющий загрузиться в графическом режиме. Поэтому до сих пор существуют текстовый редактор vi, появившийся в самом начале становления UNIX, или его более функциональные потомки vim, joe, pico, jed, встроенный редактор Midnight Commander, EMACS и система верстки (ее тоже можно отнести к текстовым редакторам) TeX.

Под X Window еще больше редакторов. Очень много простых, типа Windows Notepad, и, конечно, много мощных тек-

стовых процессоров, часть из которых входит в офисные пакеты. В качестве примера можно привести Kedit, Gedit, Kwrite, Kword, Ted, Abiword, StarWord и др. Более подробно о редакторах будет сказано ниже.

## **Графические оболочки**

Неоднократно опровергаемое утверждение, что Linux — чисто текстовая среда, почему-то очень живуче. Хотя по разнообразию графических оболочек (или менеджеров окон) он оставляет далеко позади семейство Windows, да и большинство UNIX-собратьев. В отличие от Windows, в Linux (UNIX) графическая оболочка (X Window) разделена на два приложения: X-сервер и менеджер окон. Сервер в какой-то мере специфичен для аппаратных средств (зависит от видеокарты, шины данных и т. д.) и выполняет роль рабочей лошади, а менеджер окон обеспечивает внешний вид приложений, отрисовку окон, меню и прочих элементов графического интерфейса. Благодаря такой независимости пользователь получает богатейший выбор средств для персонализации своего рабочего места. Можно поставить IceWM или AfterStep и получить легкую и мощную графическую среду (вполне нормально функционирующую на старших 486-х процессорах), для тех, кому надо «как в Windows» — FVWM95, для тех кому «как в Windows, но лучше» — KDE или GNOME.

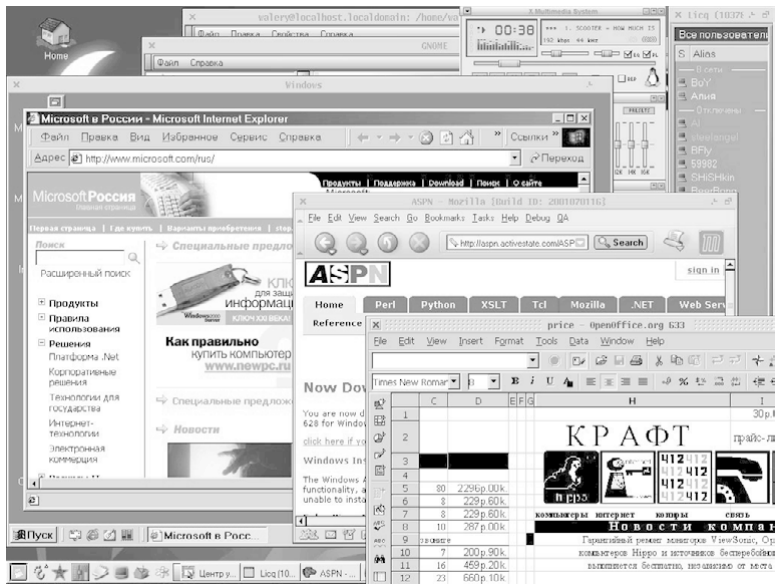


Рис. 2.5. GNOME, Win4Lin, OpenOffice, Licq

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



# Графические редакторы

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

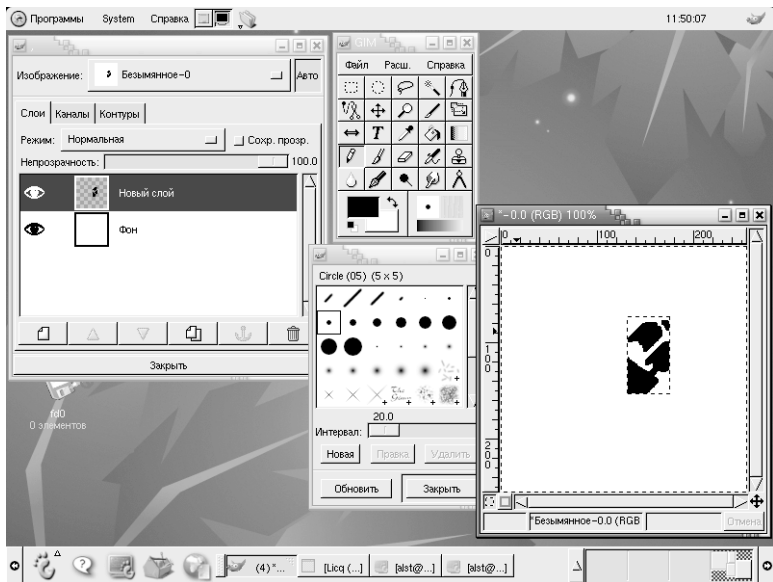


Рис. 2.6. Графический редактор Gimp

## Web-инструментарий

Традиционно лучшим редактором для Web-дизайнера считается простой текстовый редактор, однако достаточно много людей работают в специализированных HTML-редакторах. Для Linux, однако, выбор HTML-редакторов не очень большой. К примеру, программа подготовки HTML-файлов StarWriter/Web, Web Maker (разработка Алексея Дець, Россия) или Quanta Plus (разработка Дмитрия Поплавского и

## Офисные пакеты

Так исторически сложилось, что разработкой полноценного офисного пакета для Linux сообщество озаботилось только полтора-два года назад. По всей видимости, это связано с тем, что только сейчас Linux стала продвигаться на офисные рабочие места, оставаясь до последнего времени серверной операционной системой или системой, установленной на домашнем компьютере. Конечно, и до этого существовали текстовые редакторы, электронные таблицы, органайзеры и программы презентаций. Однако в полноценный офисный пакет они не складывались из-за ряда нерешенных проблем, в т. ч. несовместимости с Microsoft Office, отсутствия тесной интеграции программ от разных разработчиков и отсутствия одного разработчика, способного создать все составные части пакета, а также отсутствия полноценной поддержки русского языка.

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

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

- программу для работы с изображениями (или несколько);
  - персональный органайзер;
  - программу для организации работы в группе
- И т. д.

Офисный пакет может включать и другие программы или, наоборот, некоторые из упомянутых программ могут не входить в общий пакет. Но комплект программ можно назвать пакетом только тогда, когда все входящие в него программы обладают единым интерфейсом и позволяют обмениваться информацией между собой.

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

Поэтому рассмотрим офисные пакеты под Linux с учетом приведенных выше требований.

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

Начнем с платных пакетов.

Applixware компании Applix – судя по отзывам в Интернет, очень неплохой офисный пакет, работает стабильно и значительно быстрее, чем описанный ниже StarOffice версии 5.2. WordPerfect Office 2000 – в отличие от Applixware, менее надежен, работает помедленней и, что самое неприятное, наблюдаются проблемы с русским языком.

## **StarOffice 5.2**

Пакет StarOffice разработан немецкой фирмой Star Division, в последующем был куплен фирмой Sun Microsystems. В настоящее время пакет StarOffice 5.2 бесплатен, фирма Sun Microsystems открыла его исходный код и разрабатывает на его базе StarOffice 6.

В состав пакета входят (при инсталляции можно отказаться от установки некоторых частей пакета):

- текстовый процессор StarWriter;
- программа подготовки HTML-файлов StarWriter/Web;
- программа работы с электронными таблицами StarCalc;
- программа подготовки презентаций StarImpress;
- программа создания рисунков на основе векторной графики StarDraw;
- программа создания рисунков на основе битовых образов Staiimage;
- система управления базами данных StarBase;

- почтовая программа StarMail;
- StarDiscussion;
- StarChart;
- StarMath;
- StarSchedule;
- StarDesktop – основная оболочка, через которую организуется работа остальных частей пакета, и которая может полностью заменить интегрированную оболочку, такую как KDE или GNOME.

Непривычная для нас идеология – основная оболочка, из которой происходит запуск остальных приложений, с одной стороны, создает тесную интеграцию частей офисного пакета, но с другой – достаточно сильно замедляет работу системы и потребляет много ресурсов. Кроме того, редкий пользователь работает сразу со всеми приложениями пакета. Учитывая это, в 6-й версии и в OpenOffice отказались от основной оболочки, что в лучшую сторону сказалось на производительности и ресурсоемкости.

И, что особенно приятно, выпущена *русифицированная* версия StarOffice 5.2. Поскольку на сегодня это чуть ли не единственный приемлемый вариант офисного пакета, рассмотрим его подробнее.

Установка программ проходит без всяких проблем. Требования к компьютеру:

- процессор Pentium или выше;
- 64 Мбайт ОЗУ;

- не менее 180 Мбайт свободного места на жестком диске (в зависимости от типа инсталляции может потребоваться до 250 Мбайт);
- монитор VGA или выше с 256 цветами и разрешением не ниже 800x600;
- CD-ROM (это требование относится к тем, кто устанавливает StarOffice с CD-ROM);
- ядро Linux версии 2.0.x или выше;
- должна быть установлена система X Window с одним из оконных менеджеров;
- должна быть установлена библиотека GLibc версии 2.1.1 или выше.

На сервере Sun находится очень хорошая инструкция на русском языке по установке, настройке и решению возможных проблем. К сожалению, файл помощи пока не русифицирован. Совместимость с Microsoft Office удовлетворительная, однако могут возникать проблемы с таблицами и со связанными файлами (например, файл Excel, внедренный в файл Word).

## OpenOffice

Проект базирующийся на исходном коде StarOffice. На данный момент содержит следующие приложения:

- OpenCalc – электронные таблицы;
- OpenDraw – графический редактор;

- OpenWriter – текстовый редактор;
- Impress – программа презентации.

## Koffice

Очень динамично развивающийся пакет. Является частью проекта KDE. В состав входят:

- KSpread – электронные таблицы;
- KPresenter – создание презентаций;
- KChart – создание диаграмм;
- Krayon – растровый графический редактор;
- Kontour – векторный графический редактор;
- KFormula – математический пакет;
- KWord – WYSIWYG-текстовый редактор;
- KOrganizer – органайзер;
- Kivio – программа создания диаграмм;
- Kugar – инструмент для генерации бизнес-отчетов;
- Kplato – программа для планирования и управления проектами.

Помимо выдержанного в стиле KDE-интерфейса, отличной интеграции с другими KDE-приложениями и нормальной поддержкой русского языка, что немаловажно, заявлена совместимость с файлами Microsoft Office, а также возможность обработки файлов в форматах CSV, RTF. Очень простая инсталляция. Достаточно скромные требования к ресурсам.



# GNOME Workshop

Еще один офисный пакет от создателей GNOME. В него входят следующие программы:

- AbiWord – популярный мультиплатформенный текстовый редактор;
- Achtung – программа презентаций;
- Balsa – мощный почтовый клиент;
- Dia – отличное приложение для создания различных диаграмм, аналог Microsoft Visio;
- Eye of GNOME – программа просмотра графических изображений;
- Evolution – мощная программа, аналог Microsoft Outlook;
- Galeon – быстрый Web-браузер;
- Gfax – программа для приема и отправления факсов;
- GIMP – великолепный графический редактор;
- GNOME-DB – средство для работы с БД;
- Gnucash – персональный финансовый менеджер;
- Gnumeric – электронные таблицы;
- Guppi – программа для рисования;
- MrProject – инструмент управления проектами;
- Sketch – редактор векторной графики;
- Sodipodi – редактор векторной графики;
- Toutdoux – инструмент управления проектами.

Со временем разработчики обещают тесную интеграцию пакета с OpenOffice.

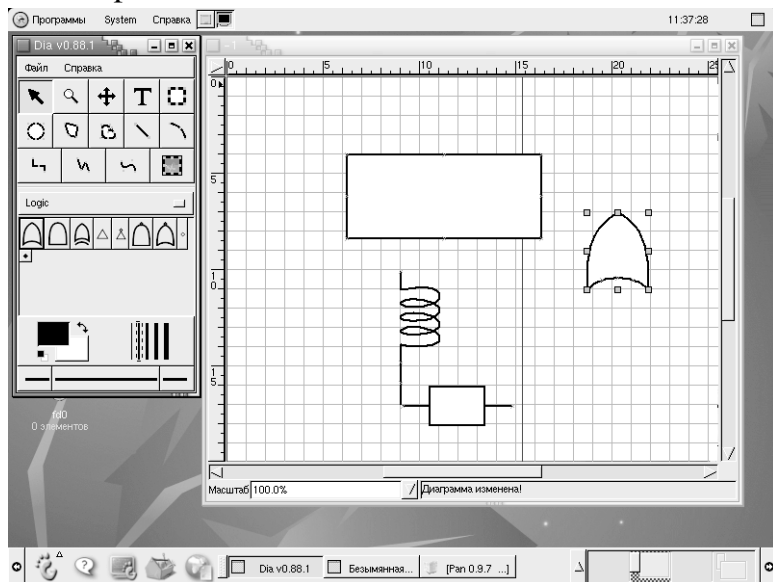


Рис. 2.7. Программа для создания диаграмм – Dia

## Базы данных

Вопреки распространенному мнению, под Linux разработано и перенесено большое количество серверов данных – от настольных до уровня предприятия. В их числе IBM DB2, Informix, Oracle, Sybase SQL Anywhere, Interbase, PostgreSQL, MySQL. Причем последние три – бесплатные,

с открытым исходным кодом. Остальные, при определенных условиях, можно получить бесплатно или пользоваться бесплатно определенное время.

## **Эмуляторы Windows**

Существует немало эмуляторов Windows в среде Linux: Citrix MetaFrame, Mainsoft's MainWin, Win4Lin, VMWare, WINE и др. Они различны по функциональным возможностям: одни обеспечивают работу приложений для Windows 9.x, другие способны запускать еще и продукты для Windows NT/2000. Есть здесь и сложность – отсутствие поддержки DirectX. Прекрасно запускаются под Linux игры, разработанные в расчете на OpenGL, но большинство самых современных игр, которые выпускаются в расчете на DirectX, пока не работают под эмуляторами. Однако, по заявлениям разработчиков, эта проблема может быть вскоре преодолена. Особняком стоит VMWare. Это не эмулятор Windows, это эмулятор компьютера! Благодаря чему под VMWare работает практически любая программа, но взамен потребляется много ресурсов.

## **Средства разработки программ**

Для Linux, как и для UNIX, «родным» языком является, естественно, C/C++, но это совершенно не означает, что кро-

ме них, никаких других компиляторов (или интерпретаторов) языков не существует. Большого разнообразия языков на одной платформе встретить невозможно. Настоящее вавилонское смешение! Трудно найти какой-либо язык, компилятора или интерпретатора которого не существует для Linux: C/C++, Pascal, Perl, Java, Lisp, Rexx, Fortran и т. д., и т. п. Не обойдены стороной и интегрированные среды разработки. Событием стал выпуск фирмой Borland интегрированной среды Kylix – Linux-аналога Delphi (Windows).

## Kylix

Приложения, написанные в Delphi 6 с использованием специальной библиотеки, можно практически без переделок перенести в Linux. Наряду с коммерческой версией Kylix существует и версия для разработки программного обеспечения с открытым исходным кодом, скачать которую можно бесплатно с Web-сайта фирмы Borland. Разработчики обещают обеспечить совместимость и с C Builder. Впрочем, и здесь есть своя ложка дегтя. Во-первых, при работе Kylix использует эмулятор Windows – Wine. Это понятно, программисты из Borland облегчили себе перенос Delphi в Linux, но поскольку Wine – программа, не до конца реализовавшая в себе Windows-совместимость и постоянно модернизируемая, Kylix временами работает нестабильно. И во-вторых, совместно с вновь созданным в Kylix приложением необхо-

димом распространять некоторые специфические библиотеки.

## Develop

Программа предназначена для разработки приложений под KDE с использованием библиотеки Qt. Можно разрабатывать консольные приложения. Обладает интерфейсом, похожим на MS Visual C++. Требуется много сторонних приложений типа a2ps, Khexedit, KTranslator и т. д. Встроен достаточно удобный интерактивный отладчик.

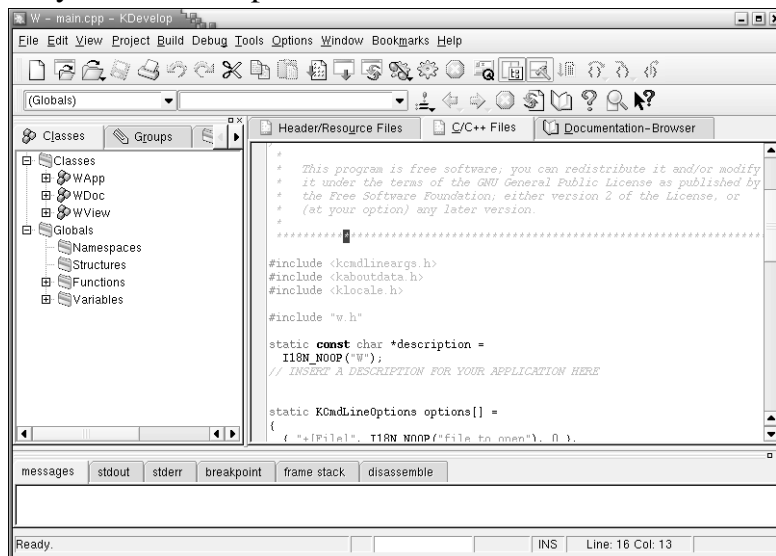


Рис. 2.8. Интегрированная оболочка разработки про-

## Glade

Визуальная среда для разработки приложений для GNOME. Достаточно неплоха, поддерживает несколько языков программирования. По сравнению с коммерческими средами оставляет ощущение незавершенности. Входит в состав GNOME.

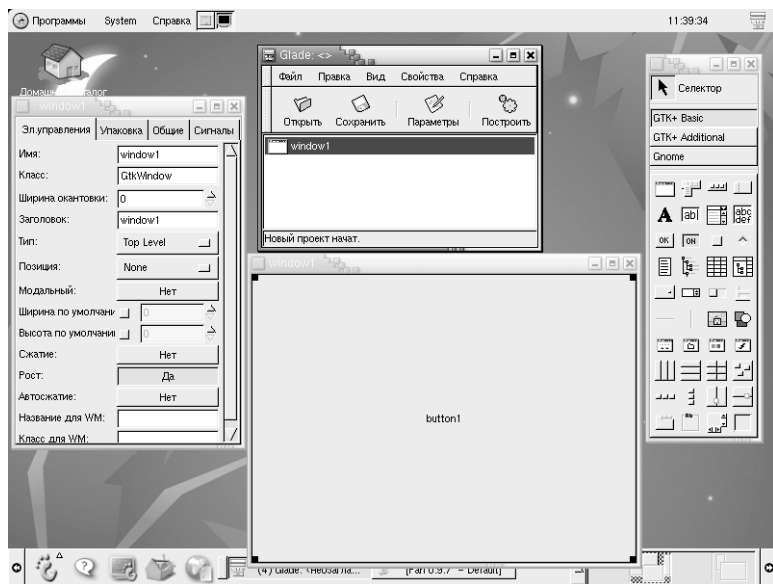


Рис. 2.9. Визуальная среда для разработки приложений Glade

## **VDK Builder**

По интерфейсу напоминает Borland Delphi/Borland C Builder, является развитием Glade. Позволяет разрабатывать приложения для GNOME. VDK – классы, позволяющие программисту получать GNOME-интерфейс, применяя только C++. Есть возможность создания консольных приложений. Нет полноценной системы справки.

## **Motor**

Редактор с подсветкой синтаксиса, менеджер проектов, генератор makefile, интегрирован с gcc и gdb. Поддерживает CVS. Умеет генерировать проекты из шаблонов. Полезная возможность – генерация RPM из проекта.

## **Rhide**

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

## **SNiFF+ Penguin IDE**

Качественный продукт, основан на коммерческой разработке. Включает в себя анализатор кода. Просмотр кода и навигация по коду производятся по дереву проекта. Поддерживает CVS и RCS. Имеет встроенный редактор документации, можно связывать пункты документации и исходный код. Сгенерированная документация для дальнейшей публикации может конвертироваться в различные форматы. Поддерживает командную разработку. Возможно использование внешних редакторов. Есть версии для Windows NT/2000, Linux, Solaris и других операционных систем.

## **Code Forge**

Платное программное обеспечение. В этой среде можно программировать на значительном количестве языков (C, C++, Assembler, Perl, GTK, TCL, TK, Tkl++, Python, Java, Basic, Fortran, Prolog, SGML, HTML и др.). Имеет настраиваемую подсветку синтаксиса, позволяет использовать любой компилятор (для C++ предлагает три разных), любые отладчики, вести версии, создавать документацию. Однако не имеет визуальных средств.



## **CodeWarrior**

Профессиональная, мощная среда разработчика IDE, интегрированная с EGCS/GNU. Редактор поддерживает подсветку синтаксиса, многооконность, быстрый доступ к функциям и многое другое. Имеется менеджер проектов с настройкой компилятора, линкера, отладчика и редактора. Можно использовать внешний редактор. Умеет запускать скрипты на этапах компиляции и линковки. Хорошо документирована. Кроме версии для Linux, имеются версии для Java, Mac OS, Windows и Solaris. Также указывается, что есть инструменты разработчика для PlayStation, Palm OS, PowerPC, MIPS.

## **CRiSP**

Работает как в X Window, так и в консоли. По внешнему виду напоминает HomeSite. Редактор поддерживает настраиваемую подсветку синтаксиса, работу с тэгами, многооконность. Имеется набор шаблонов языковых конструкций для Ada, C, C++, SQL, HTML, Latex и других. Встроен клиент FTP. Есть версии для Windows, BSD, SGI.

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

# Мультимедиа-приложения

## Аудио

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

Времена сложного ручного конфигурирования этих устройств (достаточно подробно описанные в литературе), похоже, закончились. По крайней мере, в Red Hat и его клонах поддержка звука предполагается по умолчанию. Поддерживаются почти все мало-мальски распространенные устройства. В том числе дешевые ISA– и PCI-карты. Обычно после инсталляции дистрибутива звуковая карта уже сконфигурирована и вполне работоспособна. Впрочем, иногда все же ее приходится настраивать. Для этого достаточно запустить в командной строке программу `sndconfig`. Она проведет тестирование звукового устройства и в случае благоприятного результата выдаст примеры WAV– и MIDI-звуков.

С аудиодисками также все просто. В состав KDE входит вполне нормальный (и несложный в использовании CD-плеер), аналогичный таковому из комплекта Windows. Кроме того, имеется еще несколько похожих средств как графиче-

ческих, так и консольных, например, несколько проигрывателей входят в состав GNOME.

Для управления звуком, как и в других операционных средах, используется микшер. Микшеров под Linux также очень много, существуют микшеры консольные и графические. Для примера, в составе KDE и GNOME имеется микшер, позволяющий регулировать громкость и баланс при воспроизведении звуков разного типа.

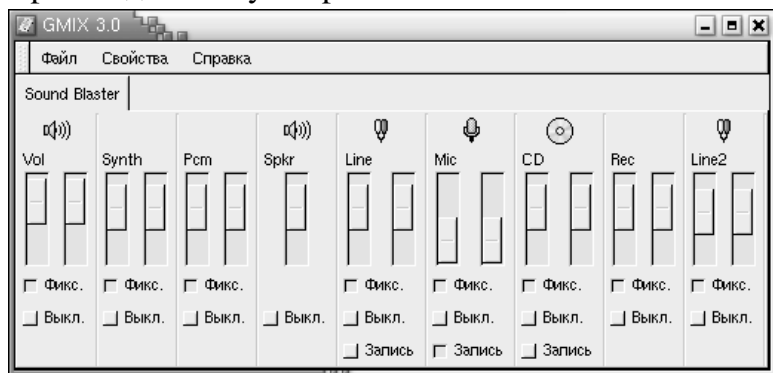


Рис. 2.10. Микшер GMIX, один из многих

KDE и GNOME также включают штатный проигрыватель для воспроизведения WAV-звука. Имеется несколько программ для проигрывания MPEG-3. Во-первых, это mpg123 – консольный проигрыватель, который потребляет очень мало ресурсов. Несколько простеньких MP3-проигрывателей входят в KDE и GNOME, имеются также XI lamp (полный

функциональный аналог известного Winamp для Windows) и KJukeBox. А если требуется преобразовать аудиодиск в MP3-файлы – есть программы, которые сделают и это, причем можно выбрать различные кодеки, качество оцифровки, получить названия треков из базы данных аудиодисков (при подключении к Интернету). Динамично развивается Vorbis – неплохой кодек для музыки.

## Видео

Начнем этот обзор с телевидения, которое принимается на компьютер с помощью TV-тюнера. Наиболее распространенные их модели поддерживаются Linux, правда не совсем понятна ситуация с USB TV-тюнерами. А для воспроизведения видеофайлов, как обычно, выбор достаточно большой. Во-первых, для KDE есть штатный видеоплеер широкого назначения – aKtion. Считывает форматы AVI, MPEG, Quick Time, а также FLI/FLC-анимации. Позволяет масштабирование (оригинальный размер, удвоенный, максимальный), а также полноэкранное воспроизведение, любой из вариантов может быть установлен по умолчанию. Имеет регулятор громкости. Есть возможность захвата кадров (в формате BMP и еще нескольких). Довольно много всяких настроек.

Для воспроизведения видео-CD специально предназначены программы mtv и Xtheater. Для воспроизведения видео, записанного в получающем все большее распростране-

ние формате MPEG-4 (DivX), можно воспользоваться программой Mplayer.

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

## Игры

С играми, наверное, хуже всего. Одно из основных применений домашнего компьютера помимо мультимедиа – это игры. Зачастую только из-за них приобретается четверть всех домашних компьютеров. Проблема с разнообразием и качеством игр напрямую вытекает из технологии их создания. Для Linux есть большое количество небольших игр типа тетриса, пасьянсов, шахмат, нардов, го и реверси. То есть таких, которые не требуют огромной работы по программированию, написанию сценария, разработке трехмерных моделей и невообразимого количества текстур и рисунков. Как только дело доходит до серьезных игр – сразу образуется вакуум. Фирмы-разработчики игр почему-то не считают рынок Linux перспективным. Игры разрабатываются в расчете на Sony Play Station, Nintendo или Windows. А на рынок Linux с играми они выходить не спешат. Тем не менее (вот парадокс)

программ-серверов, рассчитанных на Linux для игры через Интернет, достаточно много (те же Quake, Unreal). До недавнего времени только фирма Id Software выпускала Linux-версии своих игр. Фирма Loki Entertainment разработала специальную библиотеку и перенесла из Windows в Linux достаточно много популярных игр. Кроме коммерческих игр, есть несколько игр с открытым исходным кодом, хотя, в основном, это Linux-реализации давно известных коммерческих игр мира Windows. Самыми яркими представителями здесь являются FreeCiv и FreeCraft. Так что, если вы требовательны к разнообразию и качеству игр, к сожалению, Linux пока не для вас.

## Итоги

Как следует из материалов этой главы, Linux отлично справится со всякими серверными приложениями и сервисами. С точки зрения администрирования тоже особых проблем нет. Миф о том, что Linux – чисто серверная платформа, и решать на ней офисные задачи невозможно, – только миф. Буквально на днях вышла русская редакция OpenOffice 1, причем, как под Linux, так и под Windows. Помимо этого, динамично развиваются офисы KDE и GNOME. В части домашнего применения картина складывается противоречивая. С одной стороны, отличная поддержка мультимедиа, с другой стороны, практически полное отсутствие

современных игр под Linux. Остается надеяться, что с увеличением пользователей Linux производители игр будут выпускать и версии для Linux.

## Ссылки

- [www.freshmeat.net](http://www.freshmeat.net) – сайт, содержащий большое количество программ для Linux и не только.
- [www.openoffice.org](http://www.openoffice.org) – официальный сайт OpenOffice.
- [www.sun.com](http://www.sun.com) – сайт фирмы Sun.
- [koffice.kde.org](http://koffice.kde.org) – официальный сайт Koffice.
- [www.gnome.org/gnome-office](http://www.gnome.org/gnome-office) – официальный сайт GNOME-Office.
- [www.kdevelop.org](http://www.kdevelop.org) – официальный сайт KDevelop.
- [www.codeforge.com](http://www.codeforge.com) – официальный сайт Code Forge.
- [www.borland.com](http://www.borland.com) – официальный сайт фирмы Borland, разработчика Kylix.
- [www.gnome.org](http://www.gnome.org) – официальный сайт GNOME.
- [www.kde.org](http://www.kde.org) – официальный сайт KDE.
- [www.mozilla.org](http://www.mozilla.org) – официальный сайт Mozilla.
- [www.opera.com](http://www.opera.com) – сайт фирмы-разработчика Opera.
- [www.ximian.com](http://www.ximian.com) – сайт фирмы Ximian, внесшей весомый вклад в разработку GNOME, а также почтового клиента Evolution.
- [www.linuxdocs.org](http://www.linuxdocs.org) – одно из наиболее полных собраний документации о Linux.

- [www.linux-ve.chat.ru](http://www.linux-ve.chat.ru) – электронная библиотека, посвященная Linux.
- [www.citforum.ru](http://www.citforum.ru) – большое собрание русскоязычной документации и книг, в том числе посвященных Linux.
- [www.linux.org.ru](http://www.linux.org.ru) – один из основных русскоязычных сайтов, посвященных Linux.
- [www.applix.com](http://www.applix.com) – сайт фирмы-разработчика Applixware.
- [www.mysql.org](http://www.mysql.org) – официальный сайт MySQL.
- [www.interbase.com](http://www.interbase.com) – официальный сайт Interbase.
- [www.idsoftware.com](http://www.idsoftware.com) – разработчик игр Doom, Quake, Quake II, Quake III.
- [www.lokigames.com](http://www.lokigames.com) – сайт фирмы, которая переносит Windows-игры для Linux.
- [www.linuxgames.org.ru](http://www.linuxgames.org.ru) – сайт, посвященный играм для Linux.



# **Часть II**

## **Базовая информация о Linux**

### **Глава 3**

#### **Работа в сети. Основные понятия**

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

#### **Модели сетевых взаимодействий**

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

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

# Терминология

Для облегчения понимания содержимого этой главы приведем основные термины (табл. 3.1).

## *Таблица 3.1. Базовые сетевые термины*

Термин	Определение
Датаграмма	Пакет, пакет данных. Обозначает единицу информации при сетевом обмене
DNS (Domain Name Service, служба доменных имен)	Специально выделенные компьютеры, которые производят поиск соответствия символического имени хоста и цифрового адреса хоста
Интернет	Глобальная компьютерная сеть, основанная на семействе протоколов TCP/IP
FTP (File Transfer Protocol, протокол передачи файлов)	Используется для приема и передачи файлов между двумя компьютерами
IP (Internet Protocol, протокол Интернета)	Основа основ семейства протоколов TCP/IP. Практически любой протокол из этого семейства базируется на протоколе IP
NFS (Network File System, сетевая файловая система)	Система виртуальных дисков, позволяющая клиентским компьютерам использовать каталоги сервера в качестве диска
NIC (Network Information Center, сетевой информационный центр)	Организация, которая отвечает за администрирование и раздачу сетевых адресов и имен
Узел (Node, Host)	Компьютер в сети. Название применимо как к клиенту, так и к серверу
OSI (Open System Interconnection, взаимодействие открытых систем)	Модель взаимодействия открытых систем
RFC (Request For Comments, запрос для пояснений)	Стандарты протоколов Интернета и их взаимодействия
RIP (Routing Information Protocol, протокол маршрутизации информации)	Протокол, используемый для обмена информацией между маршрутизаторами
SMTP (Simple Mail Transfer Protocol, простой протокол передачи электронной почты)	Используется для обмена электронной почтой
SNMP (Simple Network Management Protocol, простой протокол управления сетью)	Используется для управления сетевыми устройствами
TCP (Transmission Control Protocol, протокол передачи данных)	Базовый протокол семейства TCP/IP, обеспечивающий надежную доставку данных

# **Модель взаимодействия открытых систем (OSI)**

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

В результате получилась семиуровневая модель:

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

Первый уровень самый элементарный, последующие – все более и более абстрагируются от особенностей физической среды передачи информации.

Каждый уровень модели OSI решает свои задачи, использует сервисы, предоставляемые предыдущим уровнем и, в свою очередь, предоставляет сервисы следующему уровню. Согласно этой модели, уровни не могут перескакивать через соседей, например, транспортный уровень не может непосредственно пользоваться сервисом физического уровня, он обязан пройти по цепочке: Сетевой уровень → Уровень дан-

ных → Физический уровень. В табл. 3.2 приведено описание уровней сетевой модели OSI.

**Таблица 3.2. Уровни сетевой модели OSI**

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

На каждом уровне блоки информации имеют собственное название (табл. 3.3).

**Таблица 3.3. Название блока информации в модели**

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

Несмотря на то что OSI является международным стандартом и на его основе правительство США выпустило спецификации GOSIP (Government Open Systems Interconnection Profile, Государственный регламент взаимодействия открытых систем), у производителей программного обеспечения стандарт OSI широкой поддержки не получил. Это объясняется несколькими причинами:

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

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

# **Модель сетевого взаимодействия TCP/IP**

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

Основываясь на этих трех объектах, разработчики выбрали четырехуровневую модель:

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

## **Сопоставление сетевых моделей OSI и TCP/IP**

Нетрудно заметить, что модель TCP/IP отличается от модели OSI. В табл. 3.4 показано соответствие модели TCP/IP и модели OSI.

---

<sup>1</sup> Здесь «интернет» – термин, указывающий на межсетевой характер взаимодействия, а, отнюдь, не Глобальная сеть Интернет. – Ред.

**Таблица 3.4. Соответствие модели TCP/IP и модели OSI**

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

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

## Сетевые протоколы

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

### Семейство протоколов TCP/IP

Семейство протоколов TCP/IP включает следующие про-



ТОКОЛЫ:

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

Схема протоколов семейства TCP/IP представлена в табл. 3.5.

**Таблица 3.5. Схема семейства протоколов TCP/IP**

Уровень приложений	FTP	SMTP	NFS	SNMP
Транспортный уровень	TCP		UDP	
Межсетевой уровень (интернет)	IP		ARP/RARP	ICMP
Уровень сетевого интерфейса	Ethernet, FDDI, ATM			
	Витая пара, коаксиальный кабель, оптический кабель и т. п.			

## **Протоколы межсетевого уровня (интернет)**

Протоколы межсетевого уровня (интернет) являются базовыми протоколами в семействе протоколов TCP/IP. Это протоколы TCP/IP, ARP/RARP и ICMP.

### **Протокол IP**

Первоначальный стандарт IP разработан в конце 1970-х годов и не был рассчитан на огромное количество хостов, которое сейчас находится в Интернете. Поэтому в настоящее время утвержден новый стандарт IP (в литературе часто старый стандарт встречается как IPv4, а новый – как IPv6). Однако массового применения он пока не нашел из-за огромного количества программных и аппаратных средств, не спо-

способных работать с IPv6, поэтому мы здесь будем рассматривать, в основном, протокол IPv4.

## Формат пакета IPv4

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

- поле *Номер версии* (VERS) указывает версию протокола IP. Сейчас повсеместно используется версия 4 и готовится переход на версию 6;
- поле *Длина заголовка* (HLEN) пакета IP. Занимает 4 бита и указывает значение длины заголовка, измеренное в 32-битовых словах. Обычно заголовок имеет длину в 20 байт (пять 32-битовых слов), но при увеличении объема служебной информации эта длина может быть увеличена за счет использования дополнительных байтов в поле *Резерв* (IP OPTIONS);
- поле *Тип сервиса* (SERVICE TYPE) занимает 1 байт и задает приоритетность пакета и вид критерия выбора маршрута. Первые три бита этого поля образуют подполе приоритета пакета (PRECEDENCE). Приоритет может иметь значения от 0 (нормальный пакет) до 7 (пакет управляющей информации). Поле *Тип сервиса* содержит также три бита, определяющие критерий выбора маршрута. Установленный бит D (delay) говорит о том, что маршрут должен выбираться для минимизации задержки доставки данного пакета, бит T – для

максимизации пропускной способности, а бит R – для максимизации надежности доставки;

- поле *Общая длина* (TOTAL LENGTH) занимает 2 байта и указывает общую длину пакета с учетом заголовка и поля данных;

- поле *Идентификатор пакета* (IDENTIFICATION) занимает 2 байта и используется для распознавания пакетов, образовавшихся путем фрагментации исходного пакета. Все фрагменты должны иметь одинаковое значение этого поля;

- поле *Флаги* (FLAGS) занимает 3 бита, оно указывает на возможность фрагментации пакета (установленный бит Do not Fragment, DF – запрещает маршрутизатору фрагментировать данный пакет), а также на то, является ли данный пакет промежуточным или последним фрагментом исходного пакета (установленный бит More Fragments, MF – говорит о том, что пакет переносит промежуточный фрагмент);

- поле *Смещение фрагмента* (FRAGMENT OFFSET) занимает 13 битов, оно используется для указания в байтах смещения поля данных этого пакета от начала общего поля данных исходного пакета, подвергнутого фрагментации. Используется при сборке/разборке фрагментов пакетов при передачах их между сетями с различными величинами максимальной длины пакета;

- поле *Время жизни* (TIME TO LIVE) занимает 1 байт и указывает предельный срок, в течение которого пакет может перемещаться по сети. Время жизни данного пакета измеря-

ется в секундах и задается источником передачи средствами протокола IP. На шлюзах и в других узлах сети по истечении каждой секунды из текущего времени жизни вычитается единица, единица вычитается также при каждой транзитной передаче (даже если не прошла секунда). По истечении времени жизни пакет аннулируется;

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

Максимальная длина *поля данных* пакета ограничена разрядностью поля, определяющего эту величину, и составляет 65 535 байтов, однако при передаче по сетям различного типа длина пакета выбирается с учетом максимальной дли-

ны пакета протокола нижнего уровня, несущего IP-пакеты. В большинстве типов локальных и глобальных сетей определяется такое понятие, как максимальный размер поля данных кадра, в который должен разместить свой пакет протокол IP. Эту величину обычно называют максимальной единицей транспортировки – MTU (Maximum Transfer Unit). К примеру, сети Ethernet имеют значение MTU, равное 1500 байтов, сети FDDI – 4096 байтов.

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

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

## **Протокол IPv6**

Основные причины, из-за которых разрабатывался IPv6:

- протокол IPv4 разрабатывался в конце 1970-х годов с

учетом существовавшей на тот момент сетевой инфраструктуры и аппаратного обеспечения. С того времени производительность массовых компьютеров увеличилась в десятки раз, и во столько же увеличилась пропускная способность сетей;

- появление приложений, использующих Интернет для передачи данных в реальном времени (звук, видео). Эти приложения чувствительны к задержкам передачи пакетов, т. к. такие задержки приводят к искажению передаваемых в реальном времени речевых сообщений и видеоизображений. Особенностью этих приложений является передача очень больших объемов информации. Однако в IPv4 не предусмотрено специального механизма резервирования полосы пропускания или механизма приоритетов;

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

Основным предложением по модернизации протокола IP является предложение, разработанное группой IETF (Internet Engineering Task Force, группа решения задач межсетевого взаимодействия). В предложении IETF протокол IPv6 оставляет неизменными основные принципы IPv4. К ним относятся датаграммный метод работы, фрагментация пакетов, разрешение отправителю задавать максимальное

число хопов (хоп – количество пересылок пакета от одного сетевого интерфейса к другому, иногда называется временем жизни пакета) для своих пакетов. Однако, в деталях реализации протокола IPv6 имеются существенные отличия от IPv4. Эти отличия коротко можно описать следующим образом:

- использование более длинных адресов. Новый размер адреса – наиболее заметное отличие IPv6 от IPv4. Версия 6 использует 128-битные адреса (16 байтов);
- гибкий формат заголовка. Вместо заголовка с фиксированными полями фиксированного размера (за исключением поля Резерв), IPv6 использует базовый заголовок фиксированного формата плюс набор необязательных заголовков различного формата;
- поддержка резервирования пропускной способности;
- поддержка расширяемости протокола. Это одно из наиболее значительных изменений в подходе к построению протокола – от полностью детализированного описания протокола к протоколу, который разрешает поддержку дополнительных функций.

## **Адресация в IPv6**

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

- Unicast – индивидуальный адрес. Определяет отдельный



узел – компьютер или порт маршрутизатора. Пакет должен быть доставлен узлу по кратчайшему маршруту;

- **Cluster** – адрес кластера. Обозначает группу узлов, которые имеют общий адресный префикс (например, присоединенных к одной физической сети). Пакет должен быть маршрутизирован группе узлов по кратчайшему пути, а затем доставлен только одному из членов группы (например, ближайшему узлу);

- **Multicast** – адрес набора узлов, находящихся в том числе в различных физических сетях. Копии пакета должны быть доставлены каждому узлу набора с использованием аппаратных возможностей групповой или широковещательной доставки, если это возможно.

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

Большая часть классов зарезервирована для будущего применения. Наиболее интересным для практического использования является класс, предназначенный для провайдеров услуг Интернета, названный **Provider-Assigned Unicast**.

Для обеспечения совместимости со схемой адресации версии IPv4, в версии IPv6 имеется класс адресов, имеющих 0000 0000 в старших битах адреса. Младшие 4 байта адреса этого класса должны содержать адрес IPv4. Маршрутизаторы, поддерживающие обе версии адресов, должны обеспечивать трансляцию при передаче пакета из сети, поддержива-

ющей адресацию IPv4, в сеть, поддерживающую адресацию IPv6, и наоборот.

## **Сетевые пакеты**

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

## **Маршрутизация пакетов**

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

# Протоколы маршрутизации

Протоколы маршрутизации классифицируются как протокол внутреннего шлюза (Interior Gateway Protocol, IGP) или протокол внешнего шлюза (Exterior Gateway Protocol, EGP).

Протокол внутреннего шлюза управляет маршрутизацией в пределах сети или группы сетей одного владельца, носящей название "автономная система". Внутри автономных систем имеется только список сетей, входящих в автономную систему, и известны точки взаимодействия с внешним миром.

Протокол внешнего шлюза отвечает за маршрутизацию между автономными системами.

На сегодняшний день широко используются следующие протоколы маршрутизации:

- RIP (Routing Information Protocol) – протокол данных маршрутизации. Устаревший протокол. Тем не менее, достаточно широко распространен благодаря утилите routed, которая является стандартной программой для операционных систем UNIX-семейства;
- OSPF (Open Shortest Path First) – протокол выбора кратчайшего пути. Протокол промышленного уровня. Рассчитан на крупные сети со сложной топологией. Более гибок, чем протокол RIP, однако по сравнению с ним сложнее в администрировании;

- IGRP (Interior Gateway Routing Protocol) – протокол маршрутизации внутреннего шлюза. Используется маршрутизаторами CISCO. По всей видимости, скоро сойдет со сцены;
- EGP (Exterior Gateway Protocol) – протокол внешнего шлюза. Старый протокол времен зарождения Интернета. Практически вытеснен протоколом BGP;
- BGP (Border Gateway Protocol) – протокол граничного шлюза. Протокол, в отличие от EGP, поддерживает сложную топологию сети. Имеет возможность широкой настройки стратегии маршрутизации;
- DVMRP (Vector Multicast Routing Protocol) – протокол групповой маршрутизации по вектору расстояния;
- RIP, OSPF и IGRP – внутренние протоколы; EGP и BGP – внешние протоколы.

## **Адресация в TCP/IP**

Каждый компьютер в сети IP имеет адреса трех уровней:

*1. Локальный адрес узла*, определяемый технологией (например, Ethernet), с помощью которой построена отдельная сеть, в которую входит данный узел. Для узлов, входящих в локальные сети – это MAC-адрес (Media Access Control) сетевого адаптера. MAC-адреса назначаются производителями оборудования и являются (теоретически) уникальными адресами, т. к. управляются централизованно, однако боль-

шинство производителей Ethernet-карт предоставляют утилиту для переназначения MAC-адреса. Для всех существующих технологий локальных сетей MAC-адрес имеет 6-байтовый формат: старшие 3 байта – идентификатор фирмы-производителя, а младшие 3 байта назначаются уникальным образом самим производителем.

- *IP-адрес*, состоящий из 4 байтов (стандарт IPv4) или 16 байтов (стандарт IPv6). Этот адрес используется на сетевом уровне. Он назначается администратором во время конфигурирования сети. IP-адрес состоит из двух частей: номера сети и номера узла. Номер сети может быть выбран администратором произвольно либо назначен по рекомендации специального подразделения Интернета (Network Information Center, NIC), если сеть должна работать как составная часть Интернета.

- Символьный *идентификатор-имя*, например, tosser.mail.ru. Этот адрес назначается администратором и состоит из нескольких частей, например, имени машины, имени фирмы, имени домена. Такой идентификатор-имя используется на прикладном уровне, например, в протоколе FTP.

IP-адрес состоит из двух частей: сетевой части и адреса хоста. На основании сетевой части адреса принимается решение о сетевой маршрутизации. Адрес хоста однозначно определяет сетевое устройство, которое, в большинстве случаев, совпадает с хостом (как обычно, не обойтись без

исключений – некоторые компьютеры имеют несколько IP-адресов). Стандартно IP-адреса записываются как десятичные числа (по одному на каждый байт адреса), разделенные точками. К примеру, 192.168.44.2. Однако не все сетевые адреса могут использоваться для назначения их компьютерам. Исключениями являются адреса 0.0.0.0, 127.0.0.1, 255.255.255.255 и некоторые другие. Существует несколько классов сетевых адресов (табл. 3.6).

**Таблица 3.6. *Распределение сетевых адресов по классам сетей***

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

\* В колонке «Формат» буквы C обозначают сетевую часть адреса, а буквы M – его компьютерную часть.

Выделением IP-адресов занимается служба регистрации информационного центра InterNIC, но если необходимо получить 4–5 IP-адресов, то их вполне может предоставить лю-

бой интернет-провайдер. Однако не все адреса предназначены для доступа из Интернета. Существует группа адресов, предназначенных для использования только в локальных сетях. Вот эти адреса:

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

Как можно видеть, это адреса классов А, В и С соответственно.

Несколько IP-адресов имеют специальное значение:

- адрес, в котором сетевая часть содержит нули, соответствует хосту в локальной сети. Например, 0.0.0.145 соответствует рабочей станции 145 в локальной сети, а адрес 0.0.0.0 – текущему хосту;
- сеть с адресом 127.X.X.X – является фиктивной сетью, не имеющей никаких аппаратных сетевых интерфейсов и состоящей только из локального компьютера. Адрес 127.0.0.1 всегда обозначает текущую машину и имеет символическое имя localhost;
- адрес, содержащий в какой-либо части число 255, обозначает широковещательный адрес. Например, пакет, посланный по адресу 192.168.3.255, будет отослан всем компьютерам в сети 192.168.3, а пакет, посланный по адресу 255.255.255.255, отправится по *всем* компьютерам Интернета.

Символьные имена Интернета имеют следующую струк-

туру:

**Имя\_компьютера.домен3уровня.домен2уровня.домен1уровня**

Например: [www.rambler.ru](http://www.rambler.ru), [www.yahoo.com](http://www.yahoo.com),  
[www.fkln.com.ua](http://www.fkln.com.ua).

Домены первого уровня стандартизированы и состоят из двух или трех букв латинского алфавита. Правда, в последнее время вводятся домены первого уровня, состоящие из более чем трех букв, но пока массового распространения они не получили. Как правило, домен первого уровня может иметь имя типа com, org, net, mil или двухсимвольного названия страны, за которой закреплен домен: ru – Россия, ua – Украина, uk – Великобритания. Относительно имени домена второго уровня строгих правил нет. Для доменов первого уровня типа com домен второго уровня имеет имя компании или фирмы. Для домена страны правило именования несколько другие. В частности, для России имя домена второго уровня определяется покупателем -к примеру, exler.ru, а для Украины имя домена второго уровня – это либо название областного центра (odessa.ua), либо имя типа com, org, net, mil. Похожая ситуация наблюдается и в других странах – Швеция, Франция, Германия имена доменов второго уровня жестко не закрепляют, а Великобритания, Тайвань, Япония – закрепляют.



# **Протокол адресации ARP/RARP**

Не смотря на то, что адресация IP-пакетов осуществляется при помощи IP-адресов, при передаче данных с компьютера на компьютер необходимо использовать аппаратные MAC-адреса (конечно, кроме тех случаев, когда используется соединение типа «точка-точка»). Для определения соответствия аппаратных MAC-адресов IP-адресам служит протокол ARP (Address Resolution Protocol) – протокол преобразования адресов. Он применяется в сетях любых типов, использующих широковещательный режим. ARP можно применять только в пределах одной сети. Однако это не мешает передавать пакет через несколько сетей, просто при прохождении пакетом маршрутизатора, он определяет новый MAC-адрес приемника. Каждый компьютер в сети создает кэш ARP, который содержит последние запросы.

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

## **Протокол ICMP**

Протокол ICMP – межсетевой протокол управления сообщениями (Internet Control Message Protocol) отвечает за низкоуровневую поддержку протокола IP, включая подтвержде-

ние получения сообщения, сообщения об ошибках и многое другое. В какой-то мере может использоваться для маршрутизации пакетов.

## **Протоколы транспортного уровня**

Протоколы транспортного уровня базируются на протоколе IP. Существуют два протокола транспортного уровня – TCP и UDP. Эти протоколы обеспечивают передачу данных с заданными характеристиками между источником и приемником данных. Эти протоколы вводят новый уровень адресации, так называемый номер порта (port number), который определяет, какому процессу на хосте передаются данные. Номера портов занимают два байта. Существует список соответствия номеров портов приложениям, определенный в RFC1700 (Request For Comments, запрос для пояснений. Данные документы описывают стандарты протоколов Интернета и их взаимодействия). Некоторые зарезервированные порты приведены в табл. 3.7.

**Таблица 3.7. Сервисы и  
закрепленные за ними порты**

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

## Протокол TCP

Протокол TCP поддерживает надежную передачу данных с предварительным установлением связи между источником и приемником информации. На базе этого протокола реализована большая часть протоколов уровня приложений.

Протокол TCP имеет следующие характеристики, обуславливающие его широкое использование:

- перед началом передачи данных протокол создает канал

между источником и приемником информации путем передачи запроса на начало сеанса и получение ответа. По окончании передачи данных сеанс должен быть явно завершен путем передачи соответствующего запроса;

- доставка данных является надежной. Перед отправкой следующего пакета источник информации должен получить подтверждение о приеме предыдущего пакета от приемника информации;

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

Эти возможности позволяют программам, использующим протокол TCP, не заботиться об организации надежной передачи данных. С другой стороны, использование этого протокола приводит к уменьшению скорости передачи данных.

## **Протокол UDP**

Протокол UDP обеспечивает логический канал между источником и приемником данных без предварительного установления связи. То есть пакеты, передаваемые по протоколу UDP, не зависят друг от друга, и никакого подтверждения доставки пакета протоколом не предусматривается. Это сильно напоминает бросание бутылки с запиской в море – авось дойдет. Поэтому программы, использующие этот протокол, должны сами организовывать проверку факта доставки информации. Однако благодаря своей простоте прото-

кол UDP может при нормальных условиях передать гораздо больше информации, чем парный ему протокол TCP.

В качестве примера приведем несколько приложений, использующих протокол UDP:

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

Для всех перечисленных программ предполагается, что в случае утери пакета необходимые действия (повторная посылка пакета, выдача сообщения и тому подобные действия) осуществляются самими программами. В случае необходимости гарантированной доставки данных используется протокол TCP.

## **Протоколы уровня приложений**

Последний, четвертый уровень – уровень приложений. К сожалению, почти каждый разработчик программ, использующих протокол уровня приложения, изобретает свой протокол или модифицирует уже существующие. Однако существует некий костяк протоколов, описанный в соответствующих RFC. В зависимости от используемого протокола транспортного уровня протоколы уровня приложений либо полагаются на надежную доставку данных (протокол TCP), ли-

бо придумывают свой способ контроля достоверности данных (при использовании протокола UDP). Большая часть протоколов уровня приложений в качестве команд используют обычные английские слова (к примеру, протокол SMTP, HTTP), что значительно упрощает отладку приложений.

## **Протокол FTP**

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

## **Протокол SMTP**

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

## **Протокол Telnet**

Протокол предназначен для удаленного доступа в систе-

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

## **Сетевая файловая система NFS**

Протокол, разработанный фирмой Sun, предназначен для использования дисков и каталогов сервера рабочими станциями в качестве «псевдодисков». Возник очень давно, когда винчестер в сто мегабайт стоил весьма дорого, и использование одного диска, распределяемого через сеть, приводило к существенной экономии денежных средств. Сегодня использование протокола NFS постепенно сходит на нет, одно из немногих мест его применения – бездисковые рабочие станции, использующие NFS в качестве «своей» файловой системы.

## **Протокол IPX**

IPX (Internet Packet Exchange) – протокол обмена пакетами между сетями, разработан фирмой Novell для своего программного продукта NetWare. Однако начиная с четвер-

той версии своей операционной системы, фирма Novell стала внедрять поддержку протокола TCP/IP, а в пятой версии протокол TCP/IP стал практически «родным» для NetWare. Тем не менее, протокол IPX еще достаточно широко используется.

Протокол IPX произошел от протокола межсетевых датаграмм IDP (Internet Datagram Protocol), разработанного в научно-исследовательском центре Xerox. Протокол IPX реализует механизм сокетов с негарантированной доставкой датаграмм. Поверх протокола IPX могут функционировать большое количество протоколов, в том числе:

- протокол данных маршрутизации RIP;
- протокол обмена нумерованными пакетами SPX (Sequenced Packet Exchange), гарантированная доставка;
- протокол Echo;
- протокол сообщений об ошибках;
- протокол обмена пакетами PEP (Packet Exchange Protocol);
- протокол сервисных объявлений SAP (Service Advertisement Protocol).

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



## **Протокол AppleTalk**

Протокол AppleTalk используется в сетях фирмы Apple. Реально, помимо компьютеров Apple, он не используется нигде. В операционной системе Linux существует поддержка этого протокола, что позволяет взаимодействовать с компьютерами Apple.

## **Протокол NetBIOS**

Протокол, используемый фирмой Microsoft в своих продуктах.

## **Протокол DECnet**

Группа сетевых продуктов фирмы DEC. Поддержка в операционной системе Linux этого протокола существует. Однако маловероятно, что вы столкнетесь с этим протоколом.

## **Стандарты в Интернете**

Стандарты Интернета описаны в документах, известных как RFC (Request For Comments). В табл. 3.8 приведены некоторые стандарты.

**Таблица 3.8. Список основных стандартов Интернета**

Номер стандарта	Комментарий
RFC	Описание протокола UDP
RFC791	Описание протокола IP
RFC792	Описание протокола ICMP
RFC793	Описание протокола TCP
RFC821	Описание протокола SMTP
RFC826, RFC903	Описание протокола ARP/RARP
RFC827, RFC904, RFC911	Описание протокола маршрутизации EGP
RFC854	Описание протокола Telnet
RFC950	Описание процедуры выделения подсетей
RFC959	Описание протокола FTP
RFC1058	Описание протокола RIP
RFC1094	Описание протокола NFS
RFC1157	Описание протокола SNMP
RFC1178	Рекомендации по выбору сетевого имени компьютера
RFC1180	Введение в TCP/IP
RFC1208	Сетевые термины
RFC1219	Порядок присвоения номеров подсетей
RFC1234	Спецификация по прохождению IPX пакетов по сетям IP
RFC1245, RFC1246, RFC1247, RFC1583	Описание протокола маршрутизации OSPF
RFC1267	Описание протокола BGP
RFC1597	Распределение локальных IP-адресов
RFC1700	Зарезервированные номера портов

## Ссылки

- [www.rfc-editor.org](http://www.rfc-editor.org) – сайт, посвященный RFC.
- Map-страница (встроенная страница помощи) арг – программы, работающей с ARP-таблицей.

# **Глава 4**

## **Идеология файловой системы**

Одним из столпов операционной системы является файловая система. От ее архитектуры, возможностей, надежности во многом зависит работоспособность операционной системы. Помимо продуманной «родной» файловой системы крайне желательно, чтобы была возможность также работать с другими наиболее распространенными файловыми системами (например, FAT 16/FAT 32). В этой главе мы подробно рассмотрим, что нам предлагает Linux.

### **История развития файловых систем Linux**

Первоначально Linux разрабатывался как расширение операционной системы Minix, и было вполне логично взять от предшественника все, что можно, поскольку такое решение позволяло достаточно быстро пройти этап проектирования (ведь все уже и так разработано, надо было только создать соответствующий программный код). На тот момент (начало 1990-х, компьютеры на базе 386-го процессора считались мощными, в порядке вещей был жесткий диск емкостью 120 Мбайт) файловая система Minix была доста-

точно эффективна. Однако ее архитектурные ограничения (адреса блоков 16-битные, что ставит предел максимально-го объема файловой системы в 64 Мбайт, каталоги содержат записи с ограниченным размером, имя файла не должно превышать 14 символов) очень скоро вынудили разработчиков задуматься об альтернативной файловой системе. Была разработана «Extended File System» (Ext FS – расширенная файловая система), затем ее сменила в качестве стандартной «Second Extended File System» (Ext2FS – вторая расширенная файловая система). Сегодня практически завершена разработка следующей версии файловой системы Ext3 – с поддержкой журналирования. Существуют также и другие журналируемые файловые системы: ReiserFS (стандарт de-facto журналируемых файловых систем для серверов на базе Linux) и JFS от фирмы IBM. По всей видимости, в ближайший год будет произведен полный перевод Linux на журналируемую файловую систему. Пока же большинство системных администраторов используют файловую систему ReiserFS. Однако достаточно много аналитиков предрекают большую вероятность использования системы JFS. Аргументы в ее пользу достаточно весомы: имя IBM, отлаженность, хорошая масштабируемость и надежность. Посмотрим, кто победит.

## **Файл**

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

## Типы файлов

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

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

## Файл

Содержит информацию в некотором формате. Для операционной системы – просто набор байтов. Вся интерпре-

тация содержимого файла осуществляется прикладной программой.

## **Каталог**

Каталоги являются элементами иерархического дерева. Любой каталог может содержать файлы и подкаталоги. Каталог – это файл, содержащий список записей. Каждая запись содержит номер индексного дескриптора и имя файла. Структуру записи см. в *разд. «Физическая структура Ext2»*.

## **Файл устройства**

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

Символьный файл устройства используется для небуферизированного обмена данными с устройством – байт за байтом.

Блочный файл устройства используется для обмена с устройством блоками данных. Некоторые устройства имеют как символьный, так и блочный интерфейс.

## Канал

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

## Ссылки

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

Так же существует еще один тип ссылок, называемый *символической* ссылкой. Эта ссылка содержит только имя файла. Так как символическая ссылка не указывает на индексный дескриптор, то возможно создание ссылок на файлы, расположенные в другой файловой системе. Эти ссылки могут указывать на файл любого типа, даже на несуществующий.



## Сокет

Сокеты предназначены для взаимодействия между процессами. Часто используются для доступа к сети TCP/IP.

## Владельцы файлов

Файлы в Linux имеют трех владельцев – собственно владельца, группу и прочих пользователей. Существует только один владелец, любое количество членов группы и все остальные, которые не входят в группу. Привилегия владения – одно из ключевых понятий в системе защиты операционной системы Linux.

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

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

## Права доступа к файлам

Права доступа к файлу или к каталогу описываются тремя восьмеричными цифрами, самая левая из этой тройки – права владельца, средняя – права группы, правая – права всех остальных. Каждая из этих восьмеричных цифр представляет собой битовую маску из трех битов. Эти биты отвечают за права на (слева направо) чтение, запись и исполнение файла или каталога. Если установлена единица – доступ разрешен, если ноль – запрещен. Таким образом, права доступа к файлу, описанные цифрой 644, означают, что владелец может писать и читать файл, группа и остальные пользователи – только читать.

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

- Чтение:

- возможность просмотра содержимого файла;
- возможность чтения каталога.

- Запись:

- возможность добавить или изменить файл;
- возможность удалять или перемещать файлы в каталоге.

- Выполнение:

- возможность запуска программы;
- возможность поиска в каталоге в комбинации с правом

чтения.

Узнать о том, какие права доступа установлены к файлам и каталогам, можно, используя команду `ls`. Ниже приведен результат выполнения команды `ls -l`

```
lrwxrwxrwx    1 root    root      4 Авг 31  10:15 [ -> test
-rwxr-xr-x    1 root    root     93 Янв 22  2001 4odb_clean
-rwxr-xr-x    1 root    root     93 Янв 22  2001 4odb_clear
-rwxr-xr-x    1 root    root     95 Янв 22  2001 4odb_create
-rwxr-xr-x    1 root    root     97 Янв 22  2001 4odb_destroy
-rwxr-xr-x    1 root    root     89 Янв 22  2001 4odb_dig
-rwxr-xr-x    1 root    root     93 Янв 22  2001 4odb_grant
-rwxr-xr-x    1 root    root     97 Янв 22  2001 4odb_metadig
-rwxr-xr-x    1 root    root     99 Янв 22  2001 4odb_odmsdump
drwxr-xr-x    1 root    root     99 Янв 22  2001 t
```

В первой колонке представлены права доступа к файлу, во второй – количество жестких ссылок, в третьей – имя владельца файла, в четвертой – название группы владельца файла, в пятой – дата создания и в шестой – имя файла или каталога. В первой строке листинга вы видите ссылку на `test` (буква `l` в правах доступа обозначает, что это не файл, а ссылка). В последней строке листинга вы видите каталог `t` (буква `d` в правах доступа обозначает, что это каталог (directory), а не файл). Остальные строки листинга – файлы. В правах доступа вы видите десять символов. Первый слева – тип файла (файл, ссылка, каталог и т. п.). Следующие три символа – права доступа владельца файла: `gwx` – чтение, запись, исполняемость файла. Следующие символы, соответственно, пра-

ва доступа группы и права доступа прочих.

## Модификаторы прав доступа

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

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

Рассмотрим эти атрибуты подробнее.

- Sticky bit для файлов. В современных операционных системах потерял свое значение.

Sticky bit для каталогов. Если sticky bit установлен для каталога, то пользователь, несмотря на то, что ему разрешена запись в этот каталог, может удалять только те файлы, владельцем которых он является или к которым ему явно заданы права записи.

- SUID для файлов. Если установлены права доступа SUID и файл исполняемый, то файл при запуске на выполнение получает не права пользователя, запустившего его, а права владельца файла. Такие фокусы используются для того, чтобы пользователь мог работать с некоторыми системными файлами, владельцем которых является некий привилегированный пользователь. К примеру, для того, чтобы

пользователь мог самостоятельно изменить свой пароль при помощи утилиты `passwd`, у этой утилиты (владельцем которой является пользователь `root`) должен быть установлен бит `SUID`, поскольку она работает с файлами (`/etc/passwd`), модификацию которых имеет право производить только пользователь `root`.

- `SGID` для файлов. Если установлены права доступа `SGID`, то это аналогично установке бита `SUID`, только вместо владельца файла используется группа владельца.

`SGID` для каталогов. В случае установки `SGID` для каталога файлы, содержащиеся в этом каталоге, будут иметь установки группы такие же, как у каталога.

Узнать о том, какие дополнительные права доступа к файлам и каталогам установлены, можно, используя команду `ls`. Ниже приведен результат выполнения `ls -l`

```
-r-s-x-x 1 root root 13536 Июл 12 2000 passwd
```

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

## **Файловые системы**

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

Перед тем как раздел или диск могут быть использованы для размещения файловой системы, она должна быть инициализирована, а требуемые служебные данные перенесены на этот раздел или диск. Этот процесс называется созданием файловой системы (иногда его еще называют форматированием, что в принципе неверно).

Основными понятиями в файловой структуре Linux (и в большинстве операционных систем UNIX-семейства) являются:

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

Подробную информацию см. в *разд. «Физическая структура Ext2»*.

## **Типы файловых систем**

Linux поддерживает большое количество типов файловых систем. Наиболее важные из них приведены ниже.

- Minix – старейшая файловая система, ограниченная в своих возможностях (у файлов отсутствуют некоторые временные параметры, длина имени файла ограничена 30-ю символами) и доступных объемах (максимум 64 Мбайт на

одну файловую систему).

- Xia – модифицированная версия системы minix, в которой увеличена максимальная длина имени файла и размер файловой системы.
- Ext – предыдущая версия системы Ext2. В настоящее время практически не используется.
- Ext2 – наиболее богатая функциональными возможностями файловая система Linux. На данный момент является самой популярной системой. Разработана с учетом совместимости с последующими версиями.
- Ext3 – модернизация файловой системы Ext2. Помимо некоторых функциональных расширений является журналируемой. Пока широкого распространения не получила. Конкурирующая журналируемая файловая система – ReiserFS.
- VFS – виртуальная файловая система. По сути – эмулятор-прослойка между реальной файловой системой (MS-DOS, Ext2, xia и т. д.) и ядром операционной системы Linux.
- Pgos – псевдо-файловая система, в которой посредством обычных файловых операций предоставляется доступ к некоторым параметрам и функциям ядра операционной системы.
- ReiserFS – журналируемая файловая система. Наиболее используемая среди журналируемых файловых систем для Linux.

В операционную систему Linux для обеспечения обмена файлами с другими операционными системами включена

поддержка некоторых файловых систем. Однако их функциональные возможности могут быть значительно ограничены по сравнению с возможностями, обычно предоставляемыми файловыми системами UNIX.

- `msdos` – обеспечивается совместимость с системой MS-DOS.

- `umsdos` – расширяет возможности драйвера файловой системы MS-DOS для Linux таким образом, что в Linux появляется возможность работы с именами файлов нестандартной длины, просмотра прав доступа к файлу, ссылок, имени пользователя, которому принадлежит файл, а также оперирования с файлами устройств. Это позволяет использовать (эмулировать) файловую систему Linux на файловой системе MS-DOS.

- `iso9660` – стандартная файловая система для CD-ROM.
- `xenix` – файловая система Xenix.
- `sysv` – файловая система System V (версия для x86).
- `hpfs` – доступ "только для чтения" к разделам HPFS.
- `nfs` – сетевая файловая система, обеспечивающая разделение одной файловой системы между несколькими компьютерами для предоставления доступа к ее файлам со всех машин.

В табл. 4.1 содержится общая информация о функциональных возможностях, предоставляемых различными файловыми системами.



## Таблица 4.1. Сравнение файловых систем

	Minix FS	Xia FS	Ext FS	Ext2 FS
Максимальный объем файловой системы	64 Мбайт	2 Гбайт	2 Гбайт	4 Тбайт
Максимальная длина файла	64 Мбайт	64 Мбайт	2 Гбайт	2 Гбайт
Максимальная длина имени файла	30 символов	248 символов	255 символов	255 символов
Поддержка трех ячеек времени изменения файла	Нет	Да	Нет	Да
Возможность расширения	Нет	Нет	Нет	Да
Изменяемый размер блока	Нет	Нет	Нет	Да
Защита информации	Да	Да	Нет	Да

## Установка файловой системы

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

Параметры командной строки, передаваемые `mkfs`, слегка различаются для разных типов файловых систем. Полное описание параметров командной строки `mkfs` можно найти

в соответствующем разделе map (справочной системы программы). С помощью параметров командной строки можно задать тип создаваемой файловой системы, произвести верификацию диска и маркировку сбойных блоков или получить список сбойных блоков из текстового файла.

## **Монтирование и демонтирование файловой системы**

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

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

У обоих вариантов имеются свои плюсы и минусы. Плюсы первого варианта – уменьшаются затраты времени на определение файловой системы и инициализацию таблиц ядра операционной системы. Минусы – невозможно "на ходу" заменить одно устройство (носитель информации) на другое (к примеру, диск Zip 100 на Zip250), поскольку в таблицах ядра зафиксированы емкость носителя, емкость кластеров, используемые блоки и тому подобная информация. Плюсы и

минусы второго варианта прямо противоположны первому – возможно "на ходу" заменить устройство (носитель информации), большие затраты времени на определение файловой системы и инициализацию таблиц ядра операционной системы. К тому же, во втором варианте намного труднее достичь надежности хранения данных.

Поэтому большинство операционных систем (не только UNIX) в явной или неявной форме используют первый вариант взаимодействия с файловой системой. Для этого в Linux используются операция «монтирования» и обратная ей «демонтирования» файловой системы. Подробную информацию см. в гл. 5.

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

Команда `mount` принимает несколько параметров, из которых обязательными являются всего два. Первый из них – файл устройства, соответствующий диску или разделу, на

котором расположена файловая система, или его псевдоним (к примеру – CD-ROM, floppy). Вторым параметром является имя каталога, к которому будет монтироваться система. Например, `mount /dev/hda1 /mnt`.

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

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

После того как отпала необходимость в использовании файловой системы, ее можно демонтировать. Чаще всего это необходимо при работе с дискетами или дисками CD-ROM (один диск необходимо заменить на другой). Для демонтажирования используется команда `umount`. В качестве параметра указывается файл устройства или точка монтирования. Например, `umount /dev/hda1` или `umount /mnt/floppy`.

По окончании работы со сменным носителем информации его обязательно необходимо отмонтировать. Поскольку ядро Linux осуществляет «отложенную» запись на диск, то к тому моменту, когда вы извлечете из дисковода дискету без

отмонтирования, информация еще может быть не записана на диск из системного буфера.

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

- в KDE или GNOME обычному пользователю можно монтировать CD-ROM и дисковод;
- осуществить временный вход в систему пользователем root, монтировать/демонтировать диск и немедленно выйти;
- применить программу sudo, позволяющую пользователям, для которых это разрешено, использовать команду mount;
- применить пакет mtools, используемый для работы с файловой системой MS-DOS;
- поместить список файлов устройств, используемых при работе с гибкими дисками, и доступных узлов монтирования вместе с нужными опциями (разрешением монтирования пользователем) в файл /etc/fstab.

## **Поддержка работоспособности файловых систем**

Даже самая надежная файловая система не обладает сто-процентной надежностью. Рано или поздно целостность файловой системы нарушается. Это может произойти от

некорректного завершения работы системы (нажата кнопка Reset, перебои в электропитании) или повреждения носителя информации. Для проверки и восстановления целостности файловой системы используется команда `fsck`. Она при загрузке системы запускается автоматически, поэтому возможные неполадки будут обнаружены (и может быть исправлены) перед использованием файловой системы.

Полная проверка файловой системы на современных жестких дисках может занять достаточно большое время, поэтому существуют некоторые способы избежать таких проверок. В файловой системе Ext2 существует специальный флаг, расположенный в суперблоке, который используется для выявления корректности демонтажа файловой системы при последнем выключении системы. Так же можно принудительно отключить проверку файловой системы, создав файл `/etc/fastboot`.

Автоматическая проверка используется только для файловых систем, монтируемых во время загрузки. Для проверки других систем команда `fsck` должна выполняться вручную.

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

Команда `fsck` должна использоваться только для демон-

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

Так же рекомендуется использовать утилиту `badblocks`. При ее выполнении выводится список номеров найденных на диске поврежденных блоков. Этот список может быть использован программой `fsck` для внесения изменений в структуру файловой системы.

## **Виртуальная файловая система (VFS)**

База, на которой основывается использование всего многообразия поддерживаемых файловых систем.

### **Принцип функционирования**

Ядро системы Linux содержит в себе программный код-посредник, выполняющий функции виртуальной файловой системы. Этот код обрабатывает запросы к файлам и вызывает необходимые функции соответствующей файловой системы для выполнения операции ввода/вывода. Такой механизм работы с файлами используется для упрощения объединения и использования нескольких типов файловых си-

стем.

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

## Структура VFS

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

Виртуальная файловая система использует таблицу, в ко-



торой во время компиляции ядра сохраняется информация о всех типах поддерживаемых файловых систем. Запись в таблице содержит тип файловой системы и указатель на соответствующую функцию монтирования файловой системы. При монтировании файловой системы эта функция возвращает виртуальной файловой системе дескриптор, который используется в дальнейшем в операциях ввода/вывода.

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

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

## **Файловая система Ext2**

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

### **Стандартные возможности Ext2**

Файловая система Ext2 поддерживает стандартные типы файлов UNIX:

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

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

## Дополнительные возможности Ext2

В файловой системе Ext2 может использоваться синхронная модификация данных. Она применяется для достижения высокой плотности записи информации, но одновременно приводит к ухудшению производительности.

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

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

Ext2 использует отдельное поле в суперблоке для индикации состояния файловой системы. Если файловая система смонтирована в режиме read/write, то ее состояние устанавливается как Clean. Если же она демонтирована или смонтирована заново в режиме read-only, то ее состояние устанавливается в Clean. Во время загрузки операционной системы и проверки состояния файловой системы эта ин-

формация используется для определения необходимости такой проверки. Ядро также помещает в это поле некоторые ошибки. При определении ядром какого-либо несоответствия файловая система помечается как `Erroneous`.

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

В системе Ext2 имеются утилиты для ее настройки. Так, программа `tune2fs` используется для определения порядка действий при обнаружении ошибки. Может быть выполнено одно из трех следующих действий:

- продолжение выполнения;
  - монтирование файловой системы заново в режиме `read-only`;
  - перезагрузка системы для проверки файловой системы.
- Кроме того, эта программа позволяет задать:

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

## **Физическая структура Ext2**

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

Блоки, в свою очередь, объединяются в группы блоков. Каждая группа блоков имеет одинаковое строение. Рассмотрим подробнее их структуру (рис. 4.1).

Суперблок (Superblock)
Описание группы блоков (Group Descriptors)
Битовая карта блока (Block Bitmap)
Битовая карта индексного дескриптора (Inode Bitmap)
Таблица индексных дескрипторов (Inode Table)
Блоки данных

Рис. 4.1. Структура группы блоков

Суперблок (Superblock) Описание группы блоков (Group Descriptors) Битовая карта блока (Block Bitmap) Битовая карта индексного дескриптора (Inode Bitmap) Таблица индексных дескрипторов (Inode Table) Блоки данных

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

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

храняются на диске. При описании структуры суперблока используются следующие значения:

# Конец ознакомительного фрагмента.

Текст предоставлен ООО «ЛитРес».

Прочитайте эту книгу целиком, [купив полную легальную версию](#) на ЛитРес.

Безопасно оплатить книгу можно банковской картой Visa, MasterCard, Maestro, со счета мобильного телефона, с платежного терминала, в салоне МТС или Связной, через PayPal, WebMoney, Яндекс.Деньги, QIWI Кошелек, бонусными картами или другим удобным Вам способом.