

Ю. С. Магда

UNIX

HDD



ДЛЯ
СТУДЕНТА



Виртуальная

Юрий Магда

**УМІХ
ДЛЯ СТУДЕНТА**

Санкт-Петербург

«БХВ-Петербург»

2007

УДК 681.3.068(075.8)
ББК 32.973.26-018.1я73
М12

Магда Ю. С.

М12 UNIX для студента. — СПб.: БХВ-Петербург,
2007. — 480 с.: ил.

ISBN 978-5-94157-972-3

Рассматривается широкий круг вопросов функционирования операционной системы UNIX, в том числе: базовые вопросы построения операционной системы, принципы организации файловой системы UNIX, учетные записи пользователей, установка, запуск и функционирование популярных операционных систем Linux, FreeBSD и Solaris, взаимодействие пользователя с операционной системой и командные оболочки. С позиции пользователя изложены базовые концепции работы операционной системы UNIX в сетях TCP/IP, настройка Интернета и электронной почты. Рассмотрены вопросы разработки программного обеспечения, в том числе программ на языке C, приложений на Java и командных сценариев на языке Perl. Приведены исходные коды целого ряда программ и примеров. В доступной форме даны принципы построения графических интерфейсов пользователя и работа с ними.

Для широкого круга пользователей UNIX

УДК 681.3.068(075.8)
ББК 32.973.26-018.1я73

Группа подготовки издания:

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

Лицензия ИД № 02429 от 24.07.00. Подписано в печать 28.12.06.

Формат 60×90^{1/16}. Печать офсетная. Усл. печ. л. 30.

Тираж 2500 экз. Заказ №

"БХВ-Петербург", 194354, Санкт-Петербург, ул. Есенина, 5Б.

Санитарно-эпидемиологическое заключение на продукцию № 77.99.02.953.Д.006421.11.04 от 11.11.2004 г. выдано Федеральной службой по надзору в сфере защиты прав потребителей и благополучия человека.

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

ISBN 978-5-94157-972-3

© Магда Ю. С., 2007

© Оформление, издательство "БХВ-Петербург", 2007

Оглавление

Введение	1
Благодарности	5
Глава 1. Обзор операционных систем UNIX	7
1.1. Solaris	11
1.2. FreeBSD	11
1.3. Linux	12
Глава 2. Основы организации UNIX	15
2.1. Ядро	20
2.1.1. Программы, процессы и потоки	24
2.1.2. Взаимодействие процессов	33
2.2. Системные процессы	39
Глава 3. Файловая система	43
3.1. Иерархия файловой системы	48
3.2. Функции API для работы с файлами	53
3.3. Операции с файлами. Индексные дескрипторы	58
3.4. Права доступа к файлам	68
3.5. Операции с дисковыми файлами	80
3.5.1. Копирование файлов	80
3.5.2. Перемещение файлов	83
3.5.3. Удаление файлов и каталогов	85
3.5.4. Создание каталогов	86
3.6. Поиск файлов и каталогов	87

Глава 4. Учетные записи пользователей	93
4.1. Команды UNIX для работы с учетными записями.....	111
4.2. Программное управление учетными записями.....	123
Глава 5. Установка, запуск и функционирование UNIX.....	131
5.1. Этапы установки системы.....	134
5.2. Основы создания файловых систем.....	138
5.2.1. Файловая система UFS	141
5.2.2. Примеры создания файловых систем.....	145
5.2.3. Диагностика файловых систем.....	160
5.3. Особенности установки различных операционных систем...	161
5.3.1. Установка операционной системы Solaris.....	180
5.3.2. Установка Linux	183
5.4. Запуск и остановка UNIX.....	190
5.4.1. Загрузка FreeBSD.....	191
5.4.2. Запуск Solaris 9.....	201
5.4.3. Запуск и останов Linux	209
Глава 6. Взаимодействие пользователя с операционной системой: командные интерпретаторы	217
6.1. Элементы языка shell.....	220
6.2. Командные файлы	240
6.3. Логические структуры командного интерпретатора	247
Глава 7. Сетевые настройки UNIX.....	259
7.1. Топология сетей	261
7.2. Модели сетевого взаимодействия.....	267
7.2.1. Модель OSI.....	267
7.2.2. Стек протоколов TCP/IP	272
7.3. Сетевые приложения	276
7.4. Адресация в Интернете	282
7.5. Маршрутизация	287
7.6. Электронная почта и Интернет.....	299
7.6.1. Программа <i>mail</i>	301
7.6.2. Программа <i>sendmail</i>	306
7.6.3. World Wide Web.....	307

7.7. Сетевые интерфейсы.....	311
7.8. Статистика работы сети.....	315
7.9. Диагностика сети и поиск неисправностей	319
7.10. Основы программирования сетевых приложений.....	325
Глава 8. Разработка программного обеспечения в среде UNIX.....	335
8.1. Разработка приложений на C++	339
8.2. Java	347
8.2.1. Первая программа на Java	351
8.2.2. Синтаксис языка.....	354
8.2.3. Введение в классы	356
8.2.4. Обработка ошибок.....	362
8.2.5. Работа со строками.....	364
8.2.6. Пакеты	371
8.2.7. Ввод/вывод в Java	372
8.2.8. Апплеты	383
8.3. Perl	384
8.3.1. Запуск программ	387
8.3.2. Скалярные переменные и массивы	388
8.3.3. Хэши	398
8.3.4. Операции и выражения.....	399
8.3.5. Логические структуры языка.....	403
8.3.6. Регулярные выражения	407
8.3.7. Обработка файлов и каталогов.....	411
8.3.8. Сетевое программирование в Perl.....	417
Глава 9. Графический интерфейс пользователя.....	421
9.1. Архитектура системы X Window	422
9.2. Команды X Window.....	438
9.3. Оконные менеджеры и графические оболочки	448
9.3.1. Оконные менеджеры	448
9.3.2. Графические оболочки.....	453
Заключение	459
Предметный указатель.....	461

Введение

В данной книге рассматриваются основы функционирования одной из популярных и наиболее динамично развивающихся в последнее время операционных систем — UNIX. Материал книги будет полезен в первую очередь студентам соответствующих специальностей и начинающим пользователям, изучающим эту операционную систему. Изложение материала базируется на принципе "о сложном — доступно", причем основное внимание уделяется теоретическим аспектам построения UNIX. В то же время автор книги старается избегать излишне упрощенного подхода при объяснении принципов организации и работы системы, поскольку при этом в тени остаются очень важные нюансы функционирования UNIX.

Для большей наглядности многие теоретические вопросы подкреплены практическими примерами программного кода, позволяющего закрепить полученные знания и, что очень важно, почувствовать, "как это работает" — при этом от читателя требуются лишь минимальные знания языка программирования C. Основное внимание уделяется описанию функционирования системы на уровне ядра и системных служб, что является основой для дальнейшего углубленного изучения операционной системы UNIX. Поскольку операционная система UNIX является прекрасной средой разработки программного обеспечения, особенно Web-приложений, и во многих случаях используется именно в этом качестве, — в книге приводится описание программных средств разработки, используемых для этих целей.

Отличие этой книги от подобных изданий заключается в том, что простота изложения не сказывается на глубине анализа материала. Здесь делается попытка объединения различных подходов к анализу операционной системы — при этом программная архитектура UNIX, программный интерфейс разработчика и интерфейс пользователя рассматриваются как части единого целого, а именно операционной системы. По этой причине многие аспекты взаимодействия пользователя и операционной системы

иллюстрируются примерами программного кода на языке C, что, по мнению автора, способствует глубокому пониманию функционирования различных подсистем UNIX и осмысленным действиям по конфигурированию и настройке операционной системы пользователем.

Материал книги охватывает следующий круг вопросов:

- анализ архитектуры UNIX, позволяющий понять основные аспекты функционирования операционных систем;
- принципы построения файловых систем UNIX;
- основы программной архитектуры системы X Window, а также настройка и использование графического интерфейса пользователя и графических оболочек;
- сетевые настройки операционной системы UNIX, а также практические аспекты функционирования Интернета и электронной почты;
- принципы разработки программного обеспечения в UNIX с использованием языков C, Java и Perl.

Книга состоит из девяти глав, краткое описание каждой из них приведено далее.

- *Глава 1. "Обзор операционных систем UNIX".*

В главе рассматриваются стандарты операционных систем UNIX и проводится сравнительный анализ наиболее популярных программных продуктов, таких как Linux, FreeBSD и Solaris. Рассмотрены также и наиболее распространенные аппаратные платформы, на которых работает UNIX.

- *Глава 2. "Основы организации UNIX".*

Материал главы посвящен обзору архитектуры операционных систем UNIX. Рассматриваются принципы построения UNIX, взаимодействие ядра с другими подсистемами, а также механизмы запуска и выполнения программ пользователя. Здесь же анализируются механизмы реализации многозадачности и многопользовательского режима. Подробно рассматриваются механизмы взаимодействия процессов, выполняющихся в операционной системе, с ядром.

□ *Глава 3. "Файловая система".*

Эта глава посвящена анализу работы файловой системы. Рассмотрена организация и структура файловых систем, операции с файлами и каталогами, а также назначение и изменение прав доступа к объектам файловой системы. Анализируются практические аспекты создания и управления файловыми системами.

□ *Глава 4. "Учетные записи пользователей".*

В главе детально рассматриваются вопросы организации доступа пользователей в UNIX. Основное внимание уделено вопросам создания, удаления и изменения учетных записей пользователя. Проанализированы основные аспекты безопасности системы и политика выбора паролей. Здесь же рассмотрен вход в систему и проверка подлинности учетных записей. Кроме того, приводится методика настройки рабочей среды пользователя.

□ *Глава 5. "Установка, запуск и функционирование UNIX".*

Детально описаны основные этапы установки и настройки операционных систем UNIX. Рассмотрены основные аспекты инсталляции операционных систем, включая выбор и подготовку дисковых накопителей, планирование, создание и монтирование файловых систем, включая примеры создания файловых систем. Здесь же проанализирована последовательность операций при запуске и остановке UNIX.

□ *Глава 6. "Взаимодействие пользователя с операционной системой: командные интерпретаторы".*

В этой главе рассматривается использование встроенного командного интерпретатора shell. Приводится описание синтаксиса shell, включая описание переменных и логических программных структур интерпретатора. Представлены многочисленные примеры применения командного интерпретатора shell для решения различных задач.

□ *Глава 7. "Сетевые настройки UNIX".*

Глава знакомит читателя с принципами функционирования операционной системы UNIX в сетях TCP/IP. Детально рассматриваются основы построения сетей и базовые протоколы

взаимодействия сетевых приложений, а также вопросы настройки и конфигурирования протокола TCP/IP в операционной системе UNIX. Большое внимание уделено тестированию сетей и поиску неисправностей. Здесь же анализируются основы работы в Интернете, а также описан протокол HTTP, являющийся базисным при обмене информацией между клиентами и Web-узлами.

Часть материала главы посвящена анализу основ функционирования систем электронной почты в UNIX. Здесь рассматриваются теоретические основы построения таких систем, а также анализируются основные аспекты использования наиболее популярных программ электронной почты, включая описание основных протоколов электронной почты.

□ *Глава 8. "Разработка программного обеспечения в среде UNIX".*

В главе обсуждаются наиболее важные аспекты разработки программного обеспечения для систем UNIX с использованием популярных языков программирования C, Java и Perl. Здесь рассмотрены принципы компиляции и сборки приложений с помощью популярного компилятора g++. Значительная часть материала главы посвящена описанию синтаксиса языков Java и Perl с детальным обсуждением примеров программного кода.

□ *Глава 9. "Графический интерфейс пользователя".*

В главе рассматриваются основы построения графических интерфейсов UNIX-систем. Значительная часть материала главы посвящена конфигурированию и настройке графической системы X Window, являющейся основой создания графических приложений для операционных систем UNIX. Кроме этого, рассматриваются основные принципы функционирования и настройки популярных графических оболочек GNOME и KDE.

Материал книги окажет существенную помощь читателям, имеющим определенный опыт работы с операционными системами UNIX, которые хотели бы существенно пополнить свои знания в этой области. Книга может быть полезна также системным администраторам и разработчикам программного обеспечения.

Благодарности

Автор выражает огромную благодарность сотрудникам издательства "БХВ-Петербург" за подготовку материалов книги к изданию. Особая признательность жене Юлии за неоценимую помощь и поддержку при подготовке рукописи.

Глава 1



Обзор операционных систем UNIX

Операционные системы UNIX переживают в настоящее время бум популярности. Будучи фактически единственным конкурентом другой популярной операционной системы — Windows, UNIX отвоевывает с каждым днем все больше и больше позиций в нише наиболее распространенных операционных систем.

Еще с момента своего появления операционная система UNIX стала широко применяться в вычислительных системах различной производительности, начиная персональными компьютерами и заканчивая большими вычислительными комплексами. Популярность этой операционной системы обусловлена несколькими факторами.

Во-первых, это более чем тридцатилетний цикл развития — за этот период операционная система UNIX выдержала проверку временем и проявила себя как очень эффективная программная среда. Во-вторых, программный код системы полностью написан на языке высокого уровня C, что делает ее понятной для пользователей и разработчиков программного обеспечения. Это позволяет относительно легко вносить изменения как в существующие версии UNIX, так и переносить операционную систему на другие аппаратные платформы. Кроме этого, во многих случаях версии этой операционной системы поставляются вместе с исходными текстами, что позволяет легко адаптировать UNIX

под специфические требования, после чего перекомпилировать систему.

Операционная система UNIX создавалась как многопользовательская и многозадачная система, ориентированная в первую очередь на выполнение серверных или управляющих функций для клиентских компьютеров. Такие подходы, а также мощные встроенные средства удаленного администрирования, способствовали тому, что UNIX заняла лидирующие позиции на рынке серверов Интернета и серверов баз данных. Существенным фактором в популяризации операционной системы является и структура ее файловой системы, представляющая единую иерархию объектов с унифицированным доступом как к файлам данных, так и к аппаратным ресурсам, таким как диски, терминалы, принтеры, сеть, память и т. п.

В практическом плане операционная система UNIX предоставляет пользователю целый ряд преимуществ. Перечислим только некоторые из них:

- все популярные приложения (пакеты офисных программ, базы данных и др.) известных производителей программного обеспечения, как правило, реализованы для работы в операционной системе UNIX;
- UNIX поддерживает широкий спектр средств передачи информации, включая многочисленные сетевые и коммуникационные протоколы, что обеспечивает эффективную работу операционной системе в сетях;
- операционная система UNIX имеет весьма развитые средства системного и сетевого управления, включая эффективные инструменты для удаленного администрирования и настройки;
- UNIX является мощной платформой для разработки приложений, в нее включены наиболее популярные языки разработки приложений, средства работы с базами данных, а также другое программное обеспечение;
- операционная система поддерживает все основные аппаратные процессорные архитектуры, включая надежную поддерж-

ку SMP, ММР и кластерных систем. В других серверных средах такая поддержка отсутствует;

- практически все важнейшие промышленные, международные, официально утвержденные и неофициальные стандарты были впервые разработаны для UNIX и только впоследствии распространились и на другие операционные системы. В настоящее время основным стандартом является разработанная консорциумом X/Open Единая спецификация UNIX, содержащая более тысячи интерфейсов прикладных программ и поддерживаемая всеми основными производителями операционной системы UNIX. Несмотря на то, что различные версии UNIX обладают уникальными возможностями, все они отвечают требованиям стандарта POSIX (Portable Operating System Interface, переносимый интерфейс операционной системы), а также стандарта X/Open Portability Guide, Edition 4 (XPG 4) и сертифицированы X/Open на соответствие стандартам UNIX 93;
- операционная система UNIX к настоящему времени утвердилась как платформа для персональных приложений, приложений для рабочих групп и приложений корпоративного класса.

В настоящее время единого стандарта для системы UNIX не существует, и на рынке присутствует множество версий этой операционной системы, имеющих свои названия и особенности. Тем не менее, все без исключения UNIX-системы имеют однотипную архитектуру, интерфейсы и среду программирования, что дает основание считать все UNIX-системы в той или иной степени родственными.

Простота и способность операционных систем UNIX к расширению и модификациям являются весьма серьезными преимуществами по сравнению с другими системами, поэтому UNIX стали переносить на множество платформ. Тем не менее, несмотря на множество реализаций базовой системы, среди них выделяются две основные ветви, берущие начало от System V UNIX и BSD UNIX.

Одна ветвь происходит от версий 4.2, 4.3 или 4.4BSD, другая базируется на системах SVR3 (System V Release 3) или SVR4 (System V Release 4). На протяжении ряда лет версия BSD пользовалась большей популярностью в академических и научных кругах, в то время как версии System V, разработанные компанией AT&T, занимали лидирующие позиции в коммерческих организациях и в промышленности. Несмотря на существующие различия между этими основными типами UNIX-систем, подавляющее большинство пользовательских команд работает одинаково и имеет один и тот же синтаксис во всех версиях операционных систем, независимо от того, используется ли AIX, BSD, HP/UX, Linux или Solaris.

Существующие в настоящее время различия между операционными системами не имеют принципиального значения, и определить, к какой из ветвей принадлежит та или иная реализация операционной системы, иногда бывает довольно сложно. Вот основные различия между операционными системами System V и BSD:

- разные способы установки и настройки терминальных устройств;
- разные способы инициализации, именованя конфигурационных файлов и файлов инициализации системы;
- различная настройка параметров файловой системы;
- различные методы получения диагностической информации и ее отображения на консоли

и т. п.

С точки зрения пользователя принципиальных различий между разными ветвями операционной системы UNIX не существует. Справедливости ради следует отметить, что BSD и System V — далеко не единственные реализации операционной системы UNIX. Рассмотрим особенности реализации некоторых, наиболее популярных версий операционной системы UNIX и начнем с Solaris.

1.1. Solaris

Исходный код UNIX System V Release 4 был доработан компанией Sun, в результате чего появилась реализация операционной системы UNIX, названная Solaris. Эта операционная система имеет несколько основных отличий от базовой операционной системы. В частности, в Solaris были добавлены следующие возможности:

- симметричная многопроцессорная обработка;
- режим реального времени.

В настоящее время Solaris является одной из самых распространенных версий операционной системы UNIX и работает на платформах SPARC и Intel86. Для развития продукта фирма Sun Microsystems предоставила более открытый доступ к кодам операционной среды Solaris.

1.2. FreeBSD

Другая ветвь UNIX — FreeBSD, берет свое название от "Berkeley Software Distribution". Эта операционная система основана на версии 4.4BSD-Lite и сохраняет специфичные черты модели развития BSD-систем.

На базе версии 4.4BSD-Lite было создано несколько операционных систем с открытыми исходными кодами, среди которых особо следует выделить проект GNU. Операционная система FreeBSD позволяет:

- выполнять одновременно несколько задач с динамическим регулированием приоритетов, что распределяет ресурсы компьютера между приложениями и пользователями оптимальным образом;
- одновременно работать многим пользователям и использовать систему совместно для решения ряда задач. Это означает, что системные ресурсы, такие, например, как принтеры и накопители на магнитных лентах, могут распределяться между

пользователями в системе или сети, при этом для каждого пользователя или группы пользователей могут быть установлены определенные ограничения на использование того или иного ресурса. Это позволяет избежать перегрузок в работе операционной системы;

- работать с распространенными сетевыми протоколами и стандартами, такими как SLIP, PPP, NFS, DHCP и NIS. Это позволяет операционной системе FreeBSD эффективно функционировать совместно с другими операционными системами, например, Windows. Кроме того, FreeBSD может использоваться в качестве интернет-сервера, предоставляя полный спектр сервисов (WWW, FTP, маршрутизация);
- использовать стандарт X Window System (X11R6), предоставляющий графический интерфейс пользователю.

Операционная система FreeBSD обеспечивает совместимость на уровне программного кода с большинством программ, разработанных для Linux и System V Release 4, обладая полным комплектом инструментальных средств для разработки программ (языки C, C++, Fortran и Perl). Исходные тексты FreeBSD свободно распространяются через Интернет, так что систему можно оптимизировать для специальных приложений или проектов. Для коммерческих операционных систем такая возможность отсутствует.

FreeBSD очень часто используется как платформа для высокопроизводительных рабочих станций, при этом она оказывается более эффективной по сравнению с другой, не менее популярной операционной системой Linux. Системы на основе BSD могут демонстрировать бóльшую по сравнению с Linux производительность, обеспечивая при этом более высокую надежность. Наконец, операционная система FreeBSD может выполнять код, разработанный для Linux, но не наоборот.

1.3. Linux

Еще одной, очень популярной реализацией UNIX является Linux. Эта версия UNIX обладает большинством свойств, при-

сущих другим реализациям, и, кроме того, включает некоторые дополнительные возможности. Linux — это полная многозадачная многопользовательская операционная система, допускающая одновременную работу многих пользователей.

Операционная система Linux очень популярна среди миллионов пользователей. GNU/Linux вместе с набором инструментальных средств по оценкам экспертов охватывает около 40% рынка UNIX. Многие компании выпускают дистрибутивы Linux — пакеты, включающие ядро, множество утилит, приложений и программное обеспечение для установки ОС. GNU/Linux получила поддержку у таких компаний, как Sun и IBM.

Для разработки программного обеспечения, работающего в Linux, был создан специальный фонд под названием Free Software Foundation (FSF), цель которого заключается в поиске источников финансирования разработки программного обеспечения GNU. Несмотря на относительно короткую историю существования, под эгидой проекта GNU было создано и адаптировано огромное количество программ, среди которых наиболее известными являются утилиты Emacs, gcc (компилятор GNU C) и bash (командная оболочка).

Необходимо отметить, что эта операционная система хорошо совместима с рядом стандартов для UNIX на уровне исходных текстов, включая IEEE POSIX.1, System V и BSD, поскольку создавалась с расчетом на такую совместимость.

Большинство из свободно распространяемого через Интернет программного обеспечения для UNIX может быть откомпилировано для работы в Linux с минимальными изменениями. Более того, все исходные тексты для Linux, включая ядро, драйверы устройств, библиотеки, пользовательские программы и инструментальные средства, также распространяются свободно.

К специфическим особенностям Linux следует отнести контроль работ по стандарту POSIX (используемый оболочками, такими как `csh` и `bash`), работу с псевдотерминалами, поддержку национальных и стандартных клавиатур с динамически загружаемыми драйверами клавиатур.

Операционная система Linux поддерживает различные типы файловых систем. Некоторые из них, например, файловая система ext2fs, были созданы специально для Linux. Кроме того, поддерживаются и другие типы файловых систем, такие, например, как Minix и Xenix. Linux включает реализацию файловой системы MS-DOS, позволяющей прямо обращаться к файлам в формате FAT на жестком диске. Наконец, для работы с CD-ROM поддерживается стандарт файловой системы ISO 9660.

Как и другие операционные системы, Linux обеспечивает полный набор протоколов TCP/IP для сетевой работы. Сюда входят драйверы устройств для многих сетевых карт Ethernet, SLIP (Serial Line Internet Protocol), обеспечивающий пользователям доступ по TCP/IP при последовательном соединении, PLIP (Parallel Line Internet Protocol), PPP (Point-to-Point Protocol), NFS и т. д. В систему включена поддержка всего спектра сервисов TCP/IP (FTP, telnet, NNTP и SMTP).

Глава 2



Основы организации UNIX

Материал этой главы посвящен анализу основных принципов, на которых базируется операционная система UNIX. Эти принципы просты и субъективно понятны.

- Операционная система должна быть максимально надежной — это означает, в первую очередь, что система должна сохранять работоспособность при возникновении каких-либо неисправностей (отказы в работе аппаратных средств, ошибки в функционировании программного обеспечения). В более узком смысле надежность означает, что система должна по возможности обеспечивать некоторый минимальный уровень функционирования при возникновении неисправностей, а также способность к быстрому восстановлению. Например, при отказе одного из жестких дисков система должна выполнить ряд восстановительных операций, чтобы оставаться в работоспособном состоянии и обеспечивать работу пользователей — при этом самым существенным является минимизация потерь критически важных данных пользователей.

Принципиально избежать отказов нельзя — аппаратные средства имеют свойство давать сбои, а приложения пользователя, впрочем, как и программные модули самой системы, могут содержать ошибки. Важным является то, с какими "потерями" система выходит из этой ситуации. Самым неприятным следствием сбоя в работе может быть потеря критически важных данных пользователей, поэтому способность быстро восстановления системы после сбоев, включая восстанов-

ление целостности данных, является наиболее существенным критерием при оценке надежности.

Операционная система UNIX изначально проектировалась как отказоустойчивая система. Несмотря на некоторые различия в программной архитектуре, во всех современных UNIX-системах предусмотрены базовые механизмы, гарантирующие высокий уровень надежности. Вот основные из них:

- разделение программного кода операционной системы и пользовательских приложений. В практическом плане это означает, что большая часть программного кода операционной системы, включая драйверы устройств и системные сервисы, а также системные области данных и стека, изолирована от пользовательского программного кода и реализована в форме *ядра*. Если, например, пользовательский процесс попытается напрямую, минуя стандартные вызовы операционной системы, обратиться к жесткому диску, то механизмы защиты ядра не позволят выполнить эту операцию, поскольку могут быть разрушены важные дисковые структуры, что сделает систему неработоспособной. Доступ пользовательских приложений к ресурсам системы возможен только посредством определенных стандартных функций, предоставляемых системой UNIX специально для этих целей. В этом случае используется механизм системных вызовов, который мы рассмотрим далее;
- использование механизмов безопасности для доступа пользовательских процессов к ресурсам операционной системы. Каждый процесс может обращаться только к тем ресурсам (файлам и устройствам), для которых система предоставляет ему доступ. При этом каждому пользовательскому процессу назначается индивидуальный уровень привилегий, определяющий, какие операции и с какими ресурсами процесс может выполнять;
- использование механизмов резервного копирования и восстановления данных, включая полное восстановление системы.

- UNIX по своей природе является многопользовательской ОС, обеспечивая возможность одновременной работы нескольких пользователей в одной системе. Поскольку пользователи, работающие в системе, могут использовать один и тот же ресурс одновременно, например, дисковый файл или принтер, система предусматривает механизмы разделения доступа к общим ресурсам, обеспечивая эффективную работу. При доступе к общим ресурсам, кроме разделения доступа, применяется и механизм безопасности, поскольку одни пользователи могут читать и записывать данные в файл, в то время как другие могут только читать данные из файла или вообще не иметь доступа к данным. Операционные системы, допускающие работу в таком режиме, называются *многопользовательскими*.
- Операционная система UNIX позволяет выполняться множеству процессов одновременно, включая как системные, так и пользовательские процессы. Само понятие "*процесс*" является комплексным, но в первом приближении его можно представить как двоичный образ исполняемого дискового файла (программы), загруженный в память и содержащий инструкции процессора, подлежащие выполнению. Выполнение процессов обеспечивается механизмами синхронизации процессов, выполняющимися на уровне ядра системы. Операционная система управляет процессом в течение всего жизненного цикла, от начала создания и до его уничтожения.
- Единый подход, касающийся доступа к ресурсам операционной системы. В операционной системе UNIX все ресурсы, к которым может иметь доступ процесс, являются *объектами файловой системы*. Этот подход лежит в основе построения и функционирования файловой системы UNIX. UNIX рассматривает дисковые файлы программ или данных, принтеры, жесткие диски, терминальные линии и т. д. как объекты файловой системы, доступ к которым осуществляется с помощью системных вызовов. Такой механизм очень удобен, поскольку позволяет использовать единый программный интерфейс как для работы с файлами данных, так и с устройствами, независимо от физической природы объектов. Кроме этого, можно

использовать единые подходы для установки атрибутов безопасности для объектов файловой системы.

Перечисленные особенности UNIX делают ее очень удобной средой для функционирования пользовательских приложений и позволяют легко переносить ее на разные аппаратные платформы.

Посмотрим, как отображены вышеперечисленные возможности операционной системы UNIX в ее программной архитектуре. В упрощенном виде структура операционной системы UNIX показана на рис. 2.1.



Рис. 2.1. Архитектура UNIX

Основой операционной системы UNIX является *ядро*, которое взаимодействует, с одной стороны, с аппаратными средствами, а с другой стороны обеспечивает работу пользовательских программ.

Часть функций операционной системы выполняется *системными процессами*, реализованными, как правило, в форме *процессов-серверов*. Большинство таких процессов запускается обычно при начальной загрузке системы и функционирует непрерывно вплоть до ее останова. Процесс-сервер более известен под названием "*демон*" (daemon), и его основная функция — выполнение запросов на обслуживание клиентских процессов. Например, сетевой демон `inetd` управляет сетевыми сервисами, демон

telnetd обслуживает клиентские запросы по протоколу telnet и т. д. Процессы-демоны будут рассмотрены далее в этой главе.

Следует сказать, что в ядре операционной системы часть процессов также функционирует подобно демонам, хотя бóльшая часть функций ядра реализована в виде отдельных потоков (kernel threads), а не процессов.

Уровень интерфейса системных вызовов отвечает за взаимодействие программ пользователя и операционной системы. Ни одна программа пользователя не может получить доступ к ресурсам системы иначе, как посредством системных вызовов. *Системные вызовы* (system calls) реализованы как функции интерфейса прикладного программирования API (Application Programming Interface), и их чаще называют "функциями API".

Системные вызовы инициируют смену контекста выполнения процесса: процесс, работающий в режиме пользователя, переключается на выполнение в защищенном режиме (режим ядра). Такое переключение позволяет процессу вызывать защищенные процедуры ядра для выполнения системных функций. Таким образом, системные вызовы обеспечивают программный интерфейс для доступа к управлению системными ресурсами, например, памятью, дисковым пространством и периферийными устройствами. Системные вызовы реализованы в виде "библиотеки времени выполнения" (run-time library), а многие из них используются командными интерпретаторами shell. Следует заметить, что системные функции ядра для корректной работы требуют упаковки аргументов специальным образом.

В качестве примеров системных вызовов можно привести низкоуровневые функции ввода/вывода, такие как `open()`, `read()`, `write()` и `close()`.

Системные вызовы обеспечивают выполнение целого ряда операций:

- трансляции операций пользователя в запросы к драйверам устройств;
- создания, запуска и уничтожения процессов;
- ввода/вывода;

- доступа к файлам и дисковым устройствам;
- поддержки терминальных устройств.

Наличие интерфейсов в форме системных вызовов предоставляет определенные преимущества разработчикам программ. Во-первых, процесс разработки программ становится намного легче, поскольку программисту нет необходимости изучать особенности программных интерфейсов низкого уровня для каждого из устройств.

Во-вторых, повышается надежность системы в целом, поскольку ядро UNIX может проверить корректность запроса программы пользователя на уровне интерфейса, прежде чем ответить на такой запрос.

Наконец, наличие таких интерфейсов позволяет легко переносить программы на другие реализации UNIX.

На самом верхнем уровне модели, показанной на рис. 2.1, находятся программы пользователей. К таким программам относятся как программы, созданные самим пользователем, так и различные офисные приложения, текстовые редакторы, графические оболочки, программы отправки и получения электронной почты и т. д. Большинство пользовательских программ взаимодействует с системой посредством системных вызовов, хотя в некоторых случаях такие программы вообще не обращаются к ядру.

Проанализируем более детально функциональные компоненты операционной системы UNIX и начнем с ядра.

2.1. Ядро

Ядро операционной системы UNIX отвечает за выполнение базовых функций системы, таких как управление процессами, файловой системой и устройствами ввода/вывода, а также безопасностью системы. Функционирование операционной системы в основном определяется ядром, поэтому остановимся на этом более подробно.

Во многих версиях операционной системы UNIX ядро реализовано в виде большого исполняемого дискового файла, который

запускается в момент инициализации системы — так называемое *монолитное ядро*. Монолитное ядро включает все необходимые модули для работы, в том числе и драйверы устройств. Для изменения функциональности такого ядра, например, при необходимости включить новый драйвер устройства, исходный текст ядра нужно полностью перекомпилировать.

Модульное ядро содержит базовые компоненты, необходимые для загрузки системы, при этом дополнительные модули программного кода (обычно это драйверы устройств) загружаются в оперативную память и подключаются к ядру динамически, по мере необходимости, например, при включении аппаратного устройства. Такие дополнительные модули реализованы в виде загружаемых модулей ядра.

Преимуществом модульного ядра является то, что базовый модуль ядра имеет небольшой размер, быстрее загружается и требует меньше ресурсов операционной системы. В то же время монолитное ядро работает чуть быстрее, поскольку не требуется переключений контекста выполняемых процессов (что имеет место в случае загрузки/выгрузки дополнительных модулей) и дополнительной синхронизации, как при использовании отдельных модулей.

В большинстве современных операционных систем UNIX используется комбинированный тип ядра, которое, обладая широким диапазоном функциональности, допускает загрузку дополнительных модулей во время работы операционной системы.

Еще одна особенность функционирования ядра современных UNIX-систем — возможность одновременного выполнения нескольких потоков, функционирующих относительно независимо друг от друга в контексте какого-либо процесса (kernel threads). В первом приближении поток можно представить себе как отдельный выполняющийся фрагмент программного кода в рамках одного процесса, при этом ядро управляет синхронизацией выполнения потоков. В этом случае говорят о *многопоточности ядра*, которая позволяет повысить эффективность функционирования как самого ядра, так и операционной системы в целом.

Высокая эффективность многопоточности объясняется тем, что синхронизация отдельных потоков одного и того же процесса выполняется с минимальными издержками и требует значительно меньше времени, чем запуск отдельного процесса, поскольку не требуется выделение системных ресурсов и обновление системных таблиц. Потоки выполняются в контексте процесса, их создавшего, и не требуют выделения отдельного адресного пространства, в то время как для каждого вновь созданного процесса требуется копия таблиц дескрипторов родительского процесса.

Большинство современных операционных систем UNIX поддерживает выполнение не только отдельных потоков ядра, но и многопоточность на уровне пользовательских программ, повышая тем самым их производительность. В этом случае многопоточные приложения выполняются как совокупность *элементарных* (lightweight) *процессов*, которые используют общее адресное пространство, общие страницы памяти и открытые файлы. Естественно, что для получения выигрыша в производительности выполняющаяся программа должна поддерживать реализацию многопоточности. Еще одной особенностью ядра UNIX является и то, что оно функционирует в *режиме невытесняющей многозадачности* (non-preemptive multitasking). Это означает, что система не может прерывать выполнение процесса, выполняющегося в режиме ядра.

Ядро UNIX обеспечивает выполнение таких функций:

- синхронизация процессов (создание, выполнение, остановка и завершение процессов);
- планирование приоритетов выполнения процессов посредством выделения им времени процессора — в этом случае процесс выполняется в течение определенного ядром интервала (кванта) времени, после чего он приостанавливается, и ядро возобновляет или начинает выполнение другого процесса. Через определенный интервал времени приостановленный процесс возобновляет выполнение и т. д.;
- распределение памяти выполняемым процессам — при этом каждому процессу выделяется определенный объем памяти,

причем адресное пространство процесса недоступно другим процессам. Кроме того, в функции ядра входит управление общим адресным пространством нескольких процессов и областью подкачки;

- создание, изменение и удаление файловых систем, расположенных на устройствах постоянного хранения информации (жесткие диски и магнитные ленты), а также управление данными пользователей. Функции ядра управляют выделением дискового пространства, выполняют отображение физической структуры файловой системы в логическую форму, доступную для работы пользователей, а также устанавливают атрибуты доступа к объектам файловой системы, защищая пользовательские файлы от несанкционированного доступа;
- управление доступом процессов к периферийным устройствам ввода/вывода, таким как терминалы, накопители на магнитных лентах и сетевое оборудование.

Ядро операционной системы является прозрачным для пользовательской программы. Это означает, что детали взаимодействия программы пользователя и операционной системы скрыты от пользователя. Например, если программа пользователя обращается к какому-либо файлу и записывает в него данные, то ядро системы выполняет приблизительно такую последовательность действий:

1. Определяет местоположение файла на носителе.
2. Получает информацию о расположении требуемых данных в физических секторах накопителя.
3. Определяет место записи блока данных в физическую область дискового пространства и т. д.
4. Наконец, ядро вызывает драйвер устройства и передает ему параметры и код операции (записи данных), после чего и выполняется запись данных.

Кроме вышеперечисленных, ядро реализует ряд необходимых функций по обеспечению выполнения процессов пользовательского уровня, за исключением функций, которые реализуются на самом пользовательском уровне.