

# ОПЕРАЦИОННАЯ СИСТЕМА

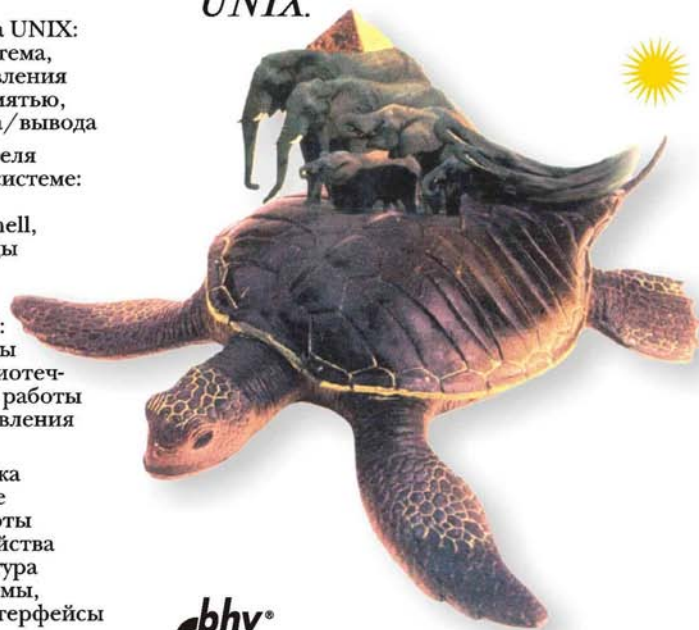
# UNIX®

## 2-Е ИЗДАНИЕ

*Принципы организации, идеология и архитектура,  
пользовательский и программный интерфейсы —  
все, что объединяет различные версии  
операционной системы под общим названием*

*UNIX.*

- Архитектура ядра UNIX: файловая подсистема, подсистема управления процессами и памятью, подсистема ввода/вывода
- Работа пользователя в операционной системе: командный интерпретатор shell, основные команды и утилиты
- Программный интерфейс UNIX: системные вызовы и основные библиотечные функции для работы с файлами и управления процессами
- Сетевая поддержка в UNIX: описание и принципы работы протоколов семейства TCP/IP, архитектура сетевой подсистемы, программные интерфейсы сокетов и TLI



**Андрей Робачевский  
Сергей Немнюгин  
Ольга Стесик**

---

ОПЕРАЦИОННАЯ СИСТЕМА

---

**UNIX<sup>®</sup>**

---

**2-е издание**

Санкт-Петербург

«БХВ-Петербург»

2005

УДК 681.3.06  
ББК 32.973.26-018.2  
P58

**Робачевский А. М., Немнюгин С. А., Стесик О. Л.**

P58 Операционная система UNIX. — 2-е изд., перераб. и доп. — СПб.: БХВ-Петербург, 2005. — 656 с.: ил.

ISBN 5-94157-538-6

Книга является вторым изданием рекомендованного Министерством общего и профессионального образования России одноименного учебного пособия.

Даны основы организации, идеологии и архитектуры, объединяющие различные версии UNIX. Рассматриваются: архитектура ядра (подсистемы ввода/вывода, управления памятью и процессами, а также файловая подсистема), программный интерфейс (системные вызовы и основные библиотечные функции), пользовательская среда (командный интерпретатор shell, основные команды и утилиты) и сетевая поддержка (протоколы семейства TCP/IP, архитектура сетевой подсистемы, программные интерфейсы сокетов и TLI).

Во второе издание включен новый материал по операционным системам Linux и FreeBSD, удалены темы, утратившие актуальность, скорректирован набор описываемых системных вызовов и библиотечных функций, добавлен глоссарий англоязычных терминов.

*Для студентов, преподавателей, пользователей и системных администраторов*

УДК 681.3.06  
ББК 32.973.26-018.2

### **Группа подготовки издания:**

Главный редактор	<i>Екатерина Кондукова</i>
Зам. гл. редактора	<i>Евгений Рыбаков</i>
Зав. редакцией	<i>Григорий Добин</i>
Редактор	<i>Алия Амирова</i>
Компьютерная верстка	<i>Натальи Караваевой</i>
Корректор	<i>Виктория Пиотровская</i>
Дизайн обложки	<i>Игоря Цырульниковца</i>
Зав. производством	<i>Николай Тверских</i>

Лицензия ИД № 02429 от 24.07.00. Подписано в печать 21.07.05.

Формат 70×100<sup>1</sup>/<sub>16</sub>. Печать офсетная. Усл. печ. л. 52,89.

Тираж 4000 экз. Заказ №

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

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

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

ISBN 5-94157-538-6

© Робачевский А. М., Немнюгин С. А., Стесик О. Л., 2005  
© Оформление, издательство "БХВ-Петербург", 2005

# Оглавление

<b>Введение</b> .....	<b>1</b>
Причины популярности UNIX.....	1
Общий взгляд на архитектуру UNIX .....	2
Ядро системы .....	3
Файловая подсистема .....	5
Подсистема управления процессами .....	5
Подсистема ввода-вывода .....	6
История создания.....	6
Исследовательские версии UNIX .....	7
Генеалогия UNIX .....	9
System V UNIX.....	9
System V Release 4 (SVR4) .....	10
UNIX компании Berkeley Software Distribution .....	10
OSF/1.....	11
Версии UNIX, использующие микроядро .....	11
Свободно распространяемая система UNIX .....	12
Основные стандарты .....	13
IEEE и POSIX .....	14
ANSI .....	15
SVID .....	15
X/Open и SUS.....	15
Некоторые известные версии UNIX.....	16
<b>Глава 1. Работа в операционной системе UNIX</b> .....	<b>21</b>
Файлы и файловая система.....	22
Типы файлов .....	23
Обычный файл .....	23
Каталог .....	23
Специальный файл устройства.....	24
FIFO (First-In-First-Out) или именованный канал .....	24
Связь .....	25

Сокеты .....	29
Структура файловой системы UNIX .....	30
Владельцы файлов .....	33
Права доступа к файлу .....	35
Дополнительные атрибуты файла .....	40
Скрытые специфические атрибуты файлов .....	43
Процессы .....	46
Программы и процессы .....	46
Типы процессов .....	47
Системные процессы .....	47
Демоны .....	48
Прикладные процессы .....	48
Атрибуты процесса .....	49
Идентификатор процесса Process Identifier (PID) .....	49
Идентификатор родительского процесса Parent Process ID (PPID) .....	49
Приоритет процесса (Nice Number) .....	49
Терминальная линия (TTY) .....	49
Реальный (RID) и эффективный (EUID) идентификаторы пользователя .....	50
Реальный (RGID) и эффективный (EGID) идентификаторы группы .....	50
Жизненный путь процесса .....	50
Сигналы .....	53
Устройства .....	56
Файлы блочных устройств .....	56
Файлы символьных устройств .....	57
Мнемоника названий специальных файлов устройств в файловой системе UNIX .....	58
Пользователи системы .....	60
Атрибуты пользователя .....	61
Поле <i>name</i> .....	62
Поле <i>passwd-encod</i> .....	62
Поле <i>UID</i> .....	63
Поле <i>GID</i> .....	63
Поле <i>comments</i> .....	63
Поле <i>home-dir</i> .....	63
Поле <i>shell</i> .....	63
Пароли .....	64
Стандартные пользователи и группы .....	65
Пользовательская среда UNIX .....	66
Командный интерпретатор shell .....	67
Синтаксис языка Bourne shell .....	69
Общий синтаксис команд .....	70
Именованные переменные .....	72
Встроенные переменные .....	77

Перенаправление ввода-вывода.....	80
Встроенные функции.....	83
Подстановки, выполняемые командным интерпретатором.....	94
Система управления заданиями .....	96
Основные утилиты UNIX.....	98
Утилиты для работы с файлами .....	98
Утилиты для управления процессами.....	104
Заключение .....	106
<b>Глава 2. Среда программирования UNIX.....</b>	<b>107</b>
Программный интерфейс ОС UNIX.....	108
Системные вызовы и функции стандартных библиотек.....	108
Обработка ошибок .....	109
Создание программы .....	115
Исходный текст.....	116
Заголовки.....	116
Компиляция.....	120
Форматы исполняемых файлов.....	122
Формат a.out.....	124
Формат ELF .....	125
Формат COFF .....	129
Выполнение программы в операционной системе UNIX.....	132
Запуск С-программы .....	132
Завершение С-программы .....	137
Работа с файлами .....	139
Основные системные функции для работы с файлами .....	140
Функция <i>open(2)</i> .....	141
Функция <i>creat(2)</i> .....	143
Функция <i>close(2)</i> .....	144
Функции <i>dup(2)</i> и <i>dup2(2)</i> .....	144
Функция <i>lseek(2)</i> .....	145
Функция <i>read(2)</i> и <i>readv(2)</i> .....	146
Функции <i>write(2)</i> и <i>writen(2)</i> .....	147
Функция <i>pipe(2)</i> .....	148
Функция <i>fcntl(2)</i> .....	149
Стандартная библиотека ввода-вывода .....	151
Связи .....	154
Файлы, отображаемые в памяти .....	158
Владение файлами .....	162
Права доступа.....	162
Перемещение по файловой системе.....	165
Метаданные файла.....	166
Процессы.....	169
Идентификаторы процесса.....	170

Выделение памяти .....	174
Создание процессов и управление ими .....	178
Сигналы .....	184
Надежные сигналы.....	192
Группы и сеансы.....	200
Текущие и фоновые группы процессов.....	203
Ограничения.....	205
Примеры программ .....	211
Демон .....	211
Командный интерпретатор .....	215
Заключение .....	218
<b>Глава 3. Подсистема управления процессами .....</b>	<b>219</b>
Основы управления процессом .....	220
Структуры данных процесса.....	222
Состояния процесса .....	229
Принципы управления памятью .....	232
Виртуальная и физическая память .....	233
Сегменты .....	236
Страничный механизм.....	239
Адресное пространство процесса.....	242
Управление памятью процесса .....	244
Области .....	245
Замещение страниц.....	248
Управление памятью в ОС Linux.....	254
Планирование выполнения процессов.....	255
Обработка прерываний таймера.....	257
Отложенные вызовы.....	258
"Будильники" (алармы) .....	259
Контекст процесса.....	260
Принципы планирования процессов .....	262
Планирование выполнения процессов в ОС Linux .....	266
Создание процесса .....	269
Запуск новой программы .....	275
Выполнение в режиме ядра.....	278
Сон и пробуждение .....	279
Завершение выполнения процесса.....	281
Сигналы .....	282
Группы и сеансы.....	282
Управление сигналами .....	282
Отправление сигнала .....	283
Доставка и обработка сигнала .....	284

Взаимодействие между процессами .....	286
Каналы .....	287
FIFO .....	288
Идентификаторы и имена в IPC .....	291
Сообщения .....	294
Семафоры .....	300
Разделяемая память .....	305
Межпроцессное взаимодействие в BSD UNIX. Сокеты .....	312
Программный интерфейс сокетов .....	314
Пример использования сокетов .....	325
Сравнение различных систем межпроцессного взаимодействия .....	329
Заключение .....	330
<b>Глава 4. Файловая подсистема .....</b>	<b>331</b>
Базовая файловая система System V .....	332
Суперблок .....	333
Индексные дескрипторы .....	334
Имена файлов .....	338
Недостатки и ограничения .....	339
Файловая система BSD UNIX .....	340
Каталоги .....	344
Файловая система ext2fs .....	345
Журнальные файловые системы .....	351
Файловая система ext3fs .....	353
Файловая система ufs2 .....	353
Архитектура виртуальной файловой системы .....	354
Виртуальные индексные дескрипторы .....	355
Монтирование файловой системы .....	359
Трансляция имен .....	367
Доступ к файловой системе .....	369
Файловые дескрипторы .....	370
Файловая таблица .....	372
Блокирование доступа к файлу .....	374
Буферный кэш .....	376
Внутренняя структура буферного кэша .....	378
Операции ввода-вывода .....	379
Кэширование в SVR4 .....	382
Целостность файловой системы .....	383
Заключение .....	387
<b>Глава 5. Подсистема ввода-вывода .....</b>	<b>389</b>
Драйверы устройств .....	390
Типы драйверов .....	390



Базовая архитектура драйверов .....	392
Файловый интерфейс .....	400
Клоны .....	403
Встраивание драйверов в ядро .....	406
Блочные устройства .....	407
Символьные устройства .....	410
Интерфейс доступа низкого уровня .....	411
Буферизация .....	412
Архитектура терминального доступа .....	414
Псевдотерминалы .....	415
Подсистема STREAMS .....	418
Архитектура STREAMS .....	420
Модули .....	424
Сообщения .....	426
Типы сообщений .....	429
Передача данных .....	431
Управление передачей данных .....	433
Драйвер .....	437
Головной модуль .....	438
Доступ к потоку .....	440
Создание потока .....	442
Управление потоком .....	444
Мультиплексирование .....	446
STREAMS в ОС Linux .....	450
Заключение .....	452
<b>Глава 6. Поддержка сети в операционной системе UNIX .....</b>	<b>453</b>
Семейство протоколов TCP/IP .....	454
Краткая история TCP/IP .....	455
Архитектура TCP/IP .....	457
Общая модель сетевого взаимодействия OSI .....	462
Протокол IP .....	465
Адресация .....	469
Протокол IP версии 6 .....	472
Формат заголовка IPv6 .....	472
Адресация IPv6 .....	475
Протоколы транспортного уровня .....	478
User Datagram Protocol (UDP) .....	480
Transmission Control Protocol (TCP) .....	482
Состояния TCP-сеанса .....	484
Передача данных .....	489
Стратегии реализации TCP .....	492
Синдром "глупого окна" .....	492

Медленный старт .....	495
Устранение затора .....	496
Повторная передача .....	498
Программные интерфейсы .....	499
Программный интерфейс сокетов .....	499
Программный интерфейс TLI/XTI .....	505
Программный интерфейс высокого уровня. Удаленный вызов процедур .....	522
Передача параметров .....	524
Связывание .....	525
Обработка особых ситуаций .....	526
Семантика вызова .....	526
Представление данных .....	527
Сеть .....	527
Как это работает? .....	528
Поддержка сети в BSD UNIX .....	535
Структуры данных .....	536
Маршрутизация .....	541
Реализация TCP/IP .....	548
Модуль IP .....	550
Модуль UDP .....	552
Модуль TCP .....	554
Поддержка сети в UNIX System V .....	555
Интерфейс TPI .....	557
Взаимодействие с прикладными процессами .....	569
Интерфейс DLPI .....	574
Доступ к среде передачи .....	577
Протокол LLC .....	579
Инкапсуляция IP .....	580
Внутренняя архитектура .....	581
Примитивы DLPI .....	585
Заключение .....	590
<b>Приложения .....</b>	<b>591</b>
<b>  Приложение А. Дополнительная информация     об операционной системе UNIX .....</b>	<b>593</b>
Книги .....	593
Информация в Интернете .....	595
<b>  Приложение Б. Глоссарий .....</b>	<b>599</b>
<b>  Предметный указатель .....</b>	<b>619</b>

# Введение

Возраст операционной системы (ОС) UNIX насчитывает уже несколько десятков лет. Это рекордный результат для операционных систем, век которых, как правило, недолог. ОС UNIX оказалась замечательным исключением из правила, она продолжает развиваться и набирать популярность. Изначально написанная для выброшенного компьютера PDP-7 с 4 килобайтами оперативной памяти, UNIX работает на множестве аппаратных платформ, начиная с обыкновенного PC и заканчивая мощными многопроцессорными системами и суперкомпьютерами.

Созданная небольшой группой разработчиков, система UNIX расширялась и дорабатывалась; тысячи специалистов вложили в нее свой талант, десятки тысяч обогатили приложениями, и сегодня она служит миллионам людей, которые используют эту операционную систему в своей деятельности.

За время своего существования система UNIX претерпела значительные изменения, стала мощней, сложнее и удобней. Однако основные идеи сохранились, удивляя нас своим изяществом и простотой. Именно они определяют "генотип" операционной системы, позволяя увидеть за красивыми названиями различных версий лаконичное слово UNIX. Именно изящество и простота этих идей являются основой жизненной силы UNIX, ее способности всегда идти в ногу со временем.

## Причины популярности UNIX

Можно смело сказать, что операционная система UNIX полностью выдержала проверку временем. На каждом этапе своего развития UNIX решала определенные задачи, и сегодня, несмотря на появление более простых и удобных, с точки зрения администрирования, систем, она прочно занимает место среди лидеров. Самое удивительное, что, несмотря на наличие версий-лидеров, во многих случаях речь при этом идет не о конкретной реализации, например Linux, FreeBSD, Solaris или SCO, а именно о системе UNIX как таковой.

Перечислим основные черты UNIX, позволяющие понять причины долговечности этой операционной системы:

- *ясность и переносимость.* Код системы написан на языке высокого уровня C, что сделало ее простой для переноса на разные платформы. По оценкам одного из создателей UNIX, Дэнниса Ритчи, система на языке C имела на 20—40 % больший размер, а производительность ее была на 20 % ниже аналогичной системы, написанной на ассемблере. Однако ясность и переносимость, полученные в результате, сыграли решающую роль в популярности UNIX;
- *модифицируемость.* Код системы изначально распространялся в виде исходных текстов, что позволяло легко вносить в нее изменения. В связи с этим новые компании и новые специалисты (и не только специалисты) легко вливались в ряды разработчиков UNIX;
- *модульность.* Разработчики заложили в основу системы модульную структуру: пользователь имел возможность дописать и включить в нее то, чего там, по его мнению, не хватало;
- *многозадачность.* Удачно написанная система разделения времени в полной мере отвечала потребностям начала эры стремительного роста производительности процессоров. В результате UNIX оказалась многозадачной многопользовательской системой с широким спектром услуг, работая в которой можно было не замечать присутствия пользователей-конкурентов;
- *стандартизация.* Несмотря на многообразие версий UNIX, основой всего семейства являются принципиально одинаковая архитектура и ряд стандартных интерфейсов. Опытный администратор без большого труда сможет обслужить другую версию системы, для пользователей переход на другую версию и вовсе может оказаться незаметным;
- *универсальность.* Следствием популярности системы стало очень большое количество приложений, в том числе свободно распространяемых, начиная от простейших текстовых редакторов и заканчивая мощными системами управления базами данных.

## Общий взгляд на архитектуру UNIX

Самый общий взгляд позволяет увидеть двухуровневую модель системы так, как она представлена на рис. В.1.

В центре находится ядро системы. Ядро непосредственно взаимодействует с аппаратной частью компьютера, изолируя прикладные программы от особенностей ее архитектуры. Ядро имеет набор услуг, предоставляемых прикладным программам. К услугам ядра относятся операции ввода-вывода (открытия, чтения, записи и управления файлами), создания и управления процессами, их синхронизации и межпроцессного взаимодействия. Все приложения запрашивают услуги ядра посредством *системных вызовов*.

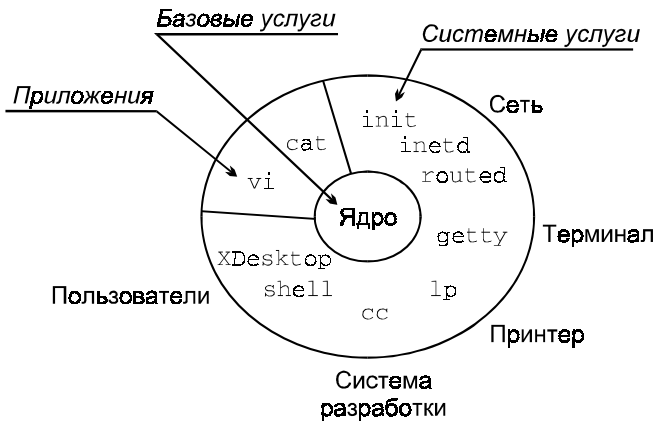


Рис. В.1. Модель системы UNIX

Второй уровень составляют приложения или задачи, как системные, определяющие функциональность системы, так и прикладные, обеспечивающие пользовательский интерфейс UNIX. Однако, несмотря на внешнюю разнородность приложений, схемы их взаимодействия с ядром одинаковы.

Рассмотрим более внимательно отдельные компоненты ядра системы.

## Ядро системы

Ядро обеспечивает базовую функциональность операционной системы: создает процессы и управляет ими, распределяет память и обеспечивает доступ к файлам и периферийным устройствам.

Взаимодействие прикладных задач с ядром происходит посредством стандартного интерфейса системных вызовов. *Интерфейс системных вызовов* представляет собой набор услуг ядра и определяет формат запросов на услуги. Процесс запрашивает услугу посредством системного вызова определенной процедуры ядра, внешне похожего на обычный вызов библиотечной функции. Ядро от имени процесса выполняет запрос и возвращает процессу необходимые данные.

В приведенном примере программа открывает файл, считывает из него данные и закрывает этот файл. При этом операции открытия (`open`), чтения (`read`) и закрытия (`close`) файла выполняются ядром по запросу задачи, а функции `open(2)`, `read(2)` и `close(2)` являются системными вызовами:

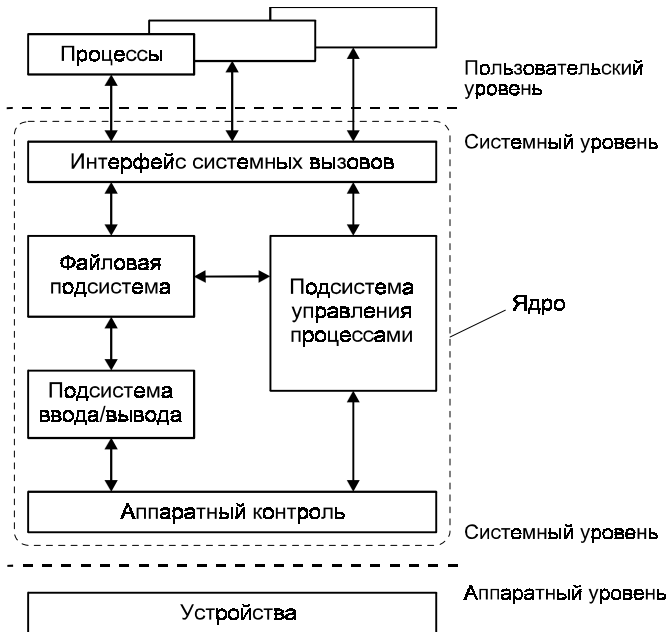
```
#include <sys/types.h>
#include <sys/stat.h>
#include <fcntl.h>
main()
```

```

{
    int fd;
    char buf[80];
    /*Откроем файл – получим ссылку (файловый дескриптор) fd*/
    fd = open("file1", O_RDONLY);
    /*Считаем в буфер buf 80 символов*/
    read(fd, buf, sizeof(buf));
    /*Закроем файл*/
    close(fd);
}

```

Структура ядра представлена на рис. В.2.



**Рис. В.2.** Внутренняя структура ядра UNIX

Ядро состоит из трех основных подсистем:

1. Файловая подсистема;
2. Подсистема управления процессами и памятью;
3. Подсистема ввода-вывода.

## Файловая подсистема

Файловая подсистема обеспечивает унифицированный интерфейс доступа к данным, расположенным на дисковых накопителях, и к периферийным устройствам. Одни и те же функции `open(2)`, `read(2)`, `write(2)` могут использоваться как при чтении или записи данных на диск, так и при выводе текста на принтер или терминал.

Файловая подсистема контролирует права доступа к файлу, выполняет операции размещения и удаления файла, а также выполняет запись/чтение данных файла. Поскольку большинство прикладных функций выполняется через интерфейс файловой системы (в том числе и доступ к периферийным устройствам), права доступа к файлам определяют привилегии пользователя в системе.

Файловая подсистема обеспечивает перенаправление запросов, адресованных периферийным устройствам, соответствующим модулям подсистемы ввода-вывода.

## Подсистема управления процессами

Запущенная на выполнение программа порождает в системе один или более *процессов* (или *задач*). Подсистема управления процессами контролирует:

- создание и удаление процессов;
- распределение системных ресурсов (памяти, вычислительных ресурсов) между процессами;
- синхронизацию процессов;
- межпроцессное взаимодействие.

Очевидно, что в общем случае число активных процессов превышает число процессоров компьютера, но в каждый конкретный момент времени на каждом процессоре может выполняться только один процесс. Операционная система управляет доступом процессов к вычислительным ресурсам, создавая ощущение одновременного выполнения нескольких задач.

Специальная задача ядра, называемая *распорядителем* или *планировщиком* процессов (scheduler), разрешает конфликты между процессами в конкуренции за системные ресурсы (процессор, память, устройства ввода-вывода). Планировщик запускает процесс на выполнение, следя за тем, чтобы процесс монопольно не захватил разделяемые системные ресурсы. Процесс освобождает процессор, ожидая длительной операции ввода-вывода, или по прошествии кванта времени. В этом случае планировщик выбирает следующий процесс с наивысшим приоритетом и запускает его на выполнение.

*Модуль управления памятью* обеспечивает размещение оперативной памяти для прикладных задач. Оперативная память является дорогостоящим ресурсом, и, как правило, ее редко бывает "слишком много". В случае, когда для

всех процессов недостаточно памяти, ядро перемещает части процесса или нескольких процессов во вторичную память (как правило, в специальную область жесткого диска), освобождая ресурсы для выполняющегося процесса. Все современные системы реализуют так называемую *виртуальную память*: процесс выполняется в собственном логическом адресном пространстве, которое может значительно превышать доступную физическую память. Управление виртуальной памятью процесса также входит в задачи модуля управления памятью.

*Модуль межпроцессного взаимодействия* отвечает за уведомление процессов о событиях с помощью сигналов и обеспечивает возможность передачи данных между различными процессами.

## Подсистема ввода-вывода

Подсистема ввода-вывода выполняет запросы файловой подсистемы и подсистемы управления процессами для доступа к периферийным устройствам (дискам, магнитным лентам, терминалам и т. д.). Она обеспечивает необходимую буферизацию данных и взаимодействует с *драйверами устройств* — специальными модулями ядра, непосредственно обслуживающими внешние устройства.

## История создания

В 1965 году Bell Telephone Laboratories (подразделение AT&T) совместно с General Electric Company и Массачусетским институтом технологии (MIT) начали разрабатывать новую операционную систему, названную MULTICS (MULTiplexed Information and Computing Service). Перед участниками проекта стояла цель создания многозадачной операционной системы разделения времени, способной обеспечить одновременную работу нескольких сотен пользователей. От Bell Labs в проекте приняли участие два сотрудника — Кен Томпсон (Ken Thompson) и Дэннис Ритчи (Dennis Ritchie). Хотя система MULTICS так и не была завершена (в 1969 году Bell Labs вышла из проекта), она стала предтечей операционной системы, впоследствии получившей название UNIX.

Однако Томпсон, Ритчи и ряд других сотрудников продолжили работу над созданием удобной среды программирования. Используя идеи и разработки, появившиеся в результате работы над MULTICS, они создали в 1969 году небольшую операционную систему, включавшую файловую систему, подсистему управления процессами и небольшой набор утилит. Система была написана на ассемблере и применялась на компьютере PDP-7. Эта операционная система получила производное от MULTICS название UNICS (UNiplexed Information and Computing Service), придуманное другим членом группы разработчиков, Брайаном Керниганом (Brian Kernighan).



### Примечание

Официальной датой рождения UNIX можно считать 1 января 1970 года. Именно с этого момента UNIX отсчитывает свое системное время.

Хотя ранняя версия UNIX многое обещала, она не смогла бы реализовать весь свой потенциал без применения в каком-либо реальном проекте. И такой проект нашелся. Когда в 1971 году патентному отделу Bell Labs понадобилась система обработки текста, в качестве операционной системы была выбрана UNIX. К тому времени система UNIX была перенесена на более мощный PDP-11, да и сама немного подросла: 16 Кбайт занимала собственно система, 8 Кбайт отводились прикладным программам, максимальный размер файла был установлен равным 64 Кбайт при 512 Кбайт дискового пространства.

Вскоре после создания первых ассемблерных версий Томпсон начал работать над компилятором для языка FORTRAN, а в результате разработал язык В. Это был интерпретатор со всеми свойственными интерпретатору ограничениями, и Ритчи переработал его в другой язык, названный С, позволявший генерировать машинный код. В 1973 году ядро операционной системы было переписано на языке высокого уровня С, — неслыханный до этого шаг, оказавший громадное влияние на популярность UNIX. Это означало, что теперь система UNIX могла быть перенесена на другие аппаратные платформы за считанные месяцы, кроме того, значительная модернизация системы и внесение изменений не представляли особых трудностей. Число работающих систем в Bell Labs превысило 25, и для сопровождения UNIX была сформирована группа UNIX System Group (USG).

## Исследовательские версии UNIX

В соответствии с федеральным законодательством США AT&T не имела права коммерческого распространения UNIX и использовала ее для собственных нужд, но начиная с 1974 года операционная система стала передаваться университетам для образовательных целей.

Операционная система модернизировалась, каждая новая версия снабжалась соответствующей редакцией Руководства Программиста, откуда и сами версии системы получили название *редакций*. Всего было выпущено 10 версий-редакций, первая из которых вышла в 1971, а последняя — в 1989 году. Первые семь редакций были разработаны в Bell Labs Группой компьютерных исследований (Computer Research Group, CRG) и предназначались для компьютеров PDP-11, позже — для VAX. Другая группа, UNIX System Group, отвечала за сопровождение системы. Третья группа (Programmer's WorkBench, PWB) занималась разработкой среды программирования, ей мы обязаны появлением системы SCCS, именованных каналов и других важных идей.

Вскоре после выпуска Седьмой редакции разработкой системы стала заниматься USG.

Наиболее важные версии описаны в табл. В.1.

*Таблица В.1. Наиболее значительные версии UNIX*

Редакция	Год выпуска	Описание
Первая редакция	1971	Первая версия UNIX, написанная на ассемблере для PDP-11. Включала компилятор В и много известных команд и утилит, в том числе <code>cat(1)</code> , <code>chdir(1)</code> , <code>chmod(1)</code> , <code>cp(1)</code> , <code>ed(1)</code> , <code>find(1)</code> , <code>mail(1)</code> , <code>mkdir(1)</code> , <code>mkfs(1M)</code> , <code>mount(1M)</code> , <code>mv(1)</code> , <code>rm(1)</code> , <code>rmdir(1)</code> , <code>wc(1)</code> , <code>who(1)</code> . В основном использовалась как инструментальное средство обработки текстов для патентного отдела Bell Labs
Третья редакция	1973	В системе появилась команда <code>cc(1)</code> , запускавшая компилятор С. Число установленных систем достигло 16
Четвертая редакция	1973	Первая система, в которой ядро написано на языке высокого уровня С
Шестая редакция	1975	Первая версия системы, доступная за пределами Bell Labs. Система полностью переписана на языке С. С этого времени появляются новые версии, разработанные за пределами Bell Labs, и растет популярность UNIX. В частности, эта версия системы была установлена Томпсоном в Калифорнийском университете в Беркли, и на ее основе вскоре была выпущена первая версия BSD (Berkeley Software Distribution) UNIX
Седьмая редакция	1979	Эта версия включала командный интерпретатор Bourne Shell и компилятор языка С от Кернигана и Ритчи. Ядро было переписано для упрощения переносимости системы на другие платформы. Лицензия на эту версию была куплена фирмой Microsoft, которая разработала на ее базе операционную систему XENIX

Популярность UNIX росла, и к 1977 году число работающих систем уже превысило 500. В 1977 году компания Interactive Systems Corporation стала первым VAR (Value Added Reseller) системы UNIX, расширив ее для использования в системах автоматизации. Этот же год стал годом первого портирования UNIX с незначительными изменениями на компьютер, отличный от PDP.

# Генеалогия UNIX

Хотя в книге речь пойдет о системах с общим названием UNIX, стоит оговориться, что обсуждать мы будем различные операционные системы. Не существует некоторой "стандартной" системы UNIX. Вы столкнетесь со множеством операционных систем, имеющих собственные названия и особенности. Но за этими особенностями и названиями все же нетрудно увидеть архитектуру, пользовательский интерфейс и среду программирования UNIX. Объясняется это достаточно просто — все эти операционные системы являются близкими или дальними родственниками. Поэтому знакомство с ними мы начнем с рассказа о генеалогии UNIX.

## System V UNIX

Начиная с 1975 года фирма AT&T начала предоставлять лицензии на использование операционной системы как научно-образовательным учреждениям, так и коммерческим организациям. Поскольку основная часть системы поставлялась в исходных текстах, написанных на языке C, опытным программистам не требовалось детальной документации, чтобы разобраться в архитектуре UNIX. С ростом популярности микропроцессоров другие компании переносили UNIX на различные платформы, причем простота и ясность операционной системы искушали многих на ее расширение и модификацию, в результате чего появилось много различных вариантов базовой системы.

Не желая терять инициативу, AT&T в 1982 году объединила несколько существующих версий UNIX и создала версию под названием System III. В отличие от редакций, предназначавшихся, в первую очередь, для внутреннего использования и не получивших дальнейшего развития, System III была создана для распространения за пределами Bell Labs и AT&T и положила начало мощной ветви UNIX, которая и сегодня жива и развивается.

В 1983 году Bell Labs выпустила новую версию системы — System V. В 1984 году группа USG была трансформирована в лабораторию (UNIX System Development Laboratory, USDL), которая вскоре выпустила новую модификацию системы — System V Release 2 (SVR2). В этой версии были реализованы такие механизмы управления памятью, как замещение страниц и копирование при записи, и представлена система межпроцессного взаимодействия (InterProcess Communication, IPC) с разделяемой памятью, очередью сообщений и семафорами.

В 1987 году появилась следующая версия — System V Release 3 (SVR3). За ее разработку отвечало новое подразделение AT&T — Информационные системы AT&T (AT&T Information Systems, ATTIS). Эта версия отличалась большим набором дополнительных возможностей, включавших:

- подсистему ввода-вывода, основанную на архитектуре STREAMS;

- переключатель файловой системы, обеспечивавший одновременную поддержку различных файловых систем;
- разделяемые библиотеки;
- программный интерфейс сетевых приложений Transport Layer Interface (TLI).

## System V Release 4 (SVR4)

В 1989 году была выпущена новая основная версия — System V Release 4. По существу, она объединила возможности нескольких известных версий UNIX: SunOS фирмы Sun Microsystems, BSD UNIX компании Berkeley Software Distribution и предыдущих версий System V.

Новые черты системы включали:

- командные интерпретаторы Korn и C (BSD);
- символические ссылки;
- систему терминального ввода-вывода, основанную на STREAMS (System V);
- отображаемые в память файлы (SunOS);
- сетевую файловую систему NFS и систему вызова удаленной процедуры RPC (SunOS);
- быструю файловую систему FFS (BSD);
- сетевой программный интерфейс сокетов (BSD);
- поддержку диспетчеризации реального времени.

Многие компоненты системы были поддержаны стандартами ANSI, POSIX, X/Open и SVID.

## UNIX компании Berkeley Software Distribution

Четвертая редакция UNIX была установлена в Калифорнийском университете в Беркли в 1974 году. С этого момента начинает свою историю ветвь UNIX, известная под названием BSD UNIX. Первая версия этой системы основывалась на Шестой редакции и была выпущена в 1978 году. В 1979 году на базе Седьмой редакции была разработана новая версия UNIX — 3BSD. Она явилась первой версией BSD, перенесенной на ЭВМ VAX. В этой системе, в частности, были реализованы виртуальная память и страничное замещение по требованию.

Важным для развития системы стал 1980 год, когда фирма Bolt, Beranek and Newman (BBN) подписала контракт с Отделом перспективных исследовательских проектов (DARPA) Министерства обороны США на разработку поддержки семейства протоколов TCP/IP в BSD UNIX. Эта работа была закончена в конце 1981 года, а ее результаты интегрированы в 4.2BSD UNIX.

Версия 4.2BSD была выпущена в середине 1983 года и включала поддержку работы в сетях, в частности, в сетях Ethernet. Это способствовало широкому распространению локальных сетей, основанных на этой технологии. Система 4.2BSD также позволяла подключиться к сети ARPANET, быстрый рост которой наблюдался с начала 80-х. Разумеется, такая операционная система не могла не пользоваться большой популярностью. К тому же, в отличие от положения в AT&T, где сетевые разработки обычно не выходили за пределы компании, результаты, полученные в Беркли, были широко доступны. Поэтому 4.2BSD стала наиболее популярной системой в исследовательских кругах.

Однако большое количество нововведений привело к тому, что система получилась сырой, содержала ряд ошибок и имела определенные проблемы с быстродействием. В 1986 году была выпущена следующая версия — 4.3BSD, более надежная и с лучшей производительностью. В период с 1986 по 1990 год в систему было внесено много дополнений, включая сетевую файловую систему NFS, виртуальную файловую систему VFS, отладчик ядра и мощную поддержку сети.

Последними версиями, выпущенными в Беркли, стали системы 4.4BSD и BSD Lite, появившиеся в 1993 году.

## OSF/1

В 1988 году AT&T и Sun Microsystems заключили соглашение о сотрудничестве в области разработки будущих версий System V. В ответ на это ряд компаний, производящих компьютеры или имеющих отношение к вычислительной технике, включая IBM, DEC, Hewlett Packard, создали организацию под названием Open Software Foundation (OSF), целью которой являлась разработка независимой от AT&T версии операционной системы. Результатом деятельности этой организации стала операционная система OSF/1. Хотя ряд коммерческих операционных систем связывают себя с этой ветвью, нельзя сказать, что OSF/1 явилась новым словом в мире UNIX. Скорее, это был политический шаг, призванный снизить доминирующую роль ряда фирм, занимавшихся разработкой UNIX System V.

## Версии UNIX, использующие микроядро

Идея *микроядра* заключается в сведении к минимуму функций, выполняемых ядром операционной системы, и, соответственно, предоставляемых базовых услуг. При этом основные компоненты операционной системы являются модулями, работающими на базе микроядра. С одной стороны, такой подход делает микроядро более универсальным, позволяя конструировать

специализированные операционные системы<sup>1</sup>, а с другой — упрощает настройку и конфигурирование.

Наиболее известны следующие версии микроядра:

- ❑ микроядро Mach, разработанное в университете Карнеги-Меллона. На его основе разработаны многие известные операционные системы: NeXT, Flux, GNU Hurd, MkLinux, Darwin;
- ❑ микроядро Chorus. Разработка Sun Microsystem<sup>2</sup>. На базе этого микроядра были созданы системы Chorus/MiX V.3, Chorus/MiX V.4 и ChorusOS;
- ❑ QNX — быстрое микроядро для платформы x86, соответствующее стандарту POSIX 1. Основа UNIX-подобной ОС реального времени QNX Neutrino;
- ❑ Minix — микроядро очень популярной версии UNIX для PC, которая была разработана Энди Тэненбаумом (Andy Tanenbaum) как приложение к его книге по архитектуре UNIX. Широко использовалось в научных и академических кругах, сыграло громадную роль в распространении UNIX.

## Свободно распространяемая система UNIX

Система UNIX никогда бы не завоевала той популярности, которой она пользуется, если бы не было свободно распространяемых версий этой системы для платформы x86. Дешевые персональные компьютеры и свободно распространяемая система UNIX делают эту систему сегодня доступной практически каждому. Соединение изящества и прозрачности UNIX с дешевизной и производительностью Intel стало причиной повсеместного внедрения и распространения UNIX.

Во многом развитие свободно распространяемого программного обеспечения обязано Фонду свободного программного обеспечения (Free Software Foundation, FSF), проекту GNU (GNU project) и их руководителю Ричарду Столмену (Richard Stallman). Он основал FSF в 1984 году для разработки программного обеспечения, главным образом, прикладного, на общественных началах. За двадцать лет работы FSF пришел к полной комплектации свободно распространяемых ОС прикладными и сервисными программами, разработанными в рамках проекта GNU.

Наиболее известны четыре версии, ориентированные на архитектуру x86 и распространяемые бесплатно. Это NetBSD, FreeBSD, OpenBSD и Linux. Первые три произошли от BSD-ветви UNIX и базируются на BSD 4.4-Lite, не попавшей под лицензионные ограничения AT&T. ОС Linux была создана

---

<sup>1</sup> Зачастую это системы реального времени.

<sup>2</sup> В 2003 году Sun объявила о прекращении поддержки Chorus.

"из ничего" сотрудником университета Хельсинки Линусом Торвальдсом (Linus Torvalds). Получилось это так. Очень популярная версия UNIX для PC, называемая Minix, была разработана Энди Тэненбаумом как приложение к его книге по архитектуре UNIX. Книга Тэненбаума содержала полные листинги исходных текстов системы. Дополнительный набор дискет позволял установить Minix даже на PC с процессором 8086 (если найдется такой компьютер). В 1991 году тогда еще студент университета Хельсинки Линус Торвальдс объявил в группе новостей comp.os.minix, что он разрабатывает (как любитель) новую Minix-подобную операционную систему для 386(486) AT. Через полтора месяца версия 0.01 операционной системы, названной Linux, была выпущена. Первый десяток версий Linux еще находился в зависимости от Minix, но Линус упорно трудился. Разработка системы была поддержана организацией Free Software Foundation, к работе над проектом присоединились сотни программистов. Вскоре началась разработка многочисленных коммерческих версий. Операционная система Linux стала настоящим всемирным проектом. Первый ошутимый результат — Linux RedHat 1.0 — был выпущен в ноябре 1994 года, к началу 2000-х годов Linux RedHat стала реальным конкурентом Microsoft Windows.

Популярность свободно распространяемых клонов BSD несколько меньше. NetBSD и отпочковавшаяся от нее OpenBSD не ориентировались на конечного потребителя: первая система задумывалась как надежная и универсальная (работающая на всех платформах), вторая — как надежная сетевая серверная система. Потенциальную конкуренцию Linux могла бы составить только FreeBSD, изначально ориентированная на широкий круг пользователей, универсальность приложений и платформу x86. Первая версия этой системы была выпущена в декабре 1993 года, почти на год раньше первой версии RedHat. FreeBSD, как правило, превосходила Linux в надежности и устойчивости, но проигрывала ей в удобстве, сначала немного, а с годами значительно. Однако в 2003 году компания RedHat, самый крупный издатель Linux, объявила о прекращении поддержки ранее выпущенных бесплатных версий (правда, осталась "наследница" свободно распространяемых версий Linux RedHat — Fedora Core). Дурной знак или хороший шанс для других?

Надо сказать, что и Linux, и FreeBSD, не говоря об Open- и NetBSD, выпускаются не только для платформы Intel, а почти для всех известных аппаратных платформ, конкурируя с операционными системами фирм-производителей компьютеров.

## Основные стандарты

UNIX явилась первой действительно переносимой системой, и в этом одна из причин ее успеха.

Как в ранние, бесплатно распространяемые, исследовательские версии, так и в сегодняшние коммерческие и свободно распространяемые версии UNIX

постоянно вносятся изменения. С одной стороны, это расширяет возможности системы, делает ее мощнее и надежнее, с другой — ведет к значительным различиям между существующими версиями, отсутствию канонической ОС UNIX.

Чем больше появлялось версий UNIX (и особенно коммерческих), тем очевиднее становилась необходимость стандартизации системы. Наличие стандартов облегчает переносимость приложений и защищает как пользователей, так и производителей. В результате возникло несколько организаций, связанных со стандартизацией, и был разработан ряд стандартов, оказавших и продолжающих оказывать значительное влияние на развитие UNIX.

## IEEE и POSIX

В 1980 году была создана инициативная группа под названием /usr/group с целью стандартизации программного интерфейса UNIX, то есть формального определения услуг, предоставляемых операционной системой приложениям. Решение этой задачи упростило бы переносимость приложений между различными версиями UNIX. Такой стандарт был создан в 1984 году и использовался комитетом ANSI, отвечающим за стандартизацию языка C, при описании библиотек. Однако с ростом числа версий операционной системы эффективность стандарта уменьшилась, и через год, в 1985 году, был создан Portable Operating System Interface for Computing Environment ("переносимый интерфейс операционной системы для вычислительной среды"), сокращенно POSIX.

В 1988 году группой был разработан стандарт POSIX 1003.1-1988, который определил программный интерфейс приложений (Application Programming Interface, API). Этот стандарт нашел широкое применение во многих операционных системах, в том числе и с архитектурой, отличной от UNIX. Спустя два года стандарт был принят как стандарт IEEE 1003.1-1990. Заметим, что поскольку этот стандарт определяет интерфейс, а не конкретную реализацию, он не делает различия между системными вызовами и библиотечными функциями, называя все элементы программного интерфейса просто функциями.

Другие наиболее значительные стандарты POSIX, относящиеся к UNIX, перечислены в табл. В.2.

**Таблица В.2.** Некоторые стандарты POSIX

POSIX 1003.2-1992	Включает определение командного интерпретатора UNIX и набора утилит
POSIX 1003.1b-1993	Содержит дополнения, относящиеся к поддержке приложений реального времени
POSIX 1003.1c-1995	Включает определения "нитей" (threads) POSIX, известных также как pthreads



## ANSI

В конце 1989 года Американским национальным институтом стандартов (American National Standards Institute, ANSI) был утвержден стандарт X3.159-1989 языка программирования C. Целью появления этого стандарта являлось улучшение переносимости программ, написанных на языке C, в различные операционные системы (не только UNIX). Стандарт определяет не только синтаксис и семантику языка, но и содержимое стандартной библиотеки.

## SVID

Вскоре после выхода в свет в 1984 году версии SVR2, группа USG выпустила документ под названием System V Interface Definition (SVID), в котором описывались внешние интерфейсы UNIX версий System V. По существу, этот труд (в двух томах) определял соответствие операционной системы версии System V.

В дополнение к SVID был выпущен так называемый System V Verification Suite (SVVS), — набор тестовых программ, позволяющих производителям получить ответ, достойна ли их система права носить имя System V.

С появлением SVR4 было выпущено новое издание SVID (уже в четырех томах) и, соответственно, новый SVVS.

## X/Open и SUS

В 1984 году ряд европейских компьютерных компаний сформировал некоммерческую организацию, получившую название X/Open. Название полностью отражает цель этой организации — разработку общего набора интерфейсов операционной системы, согласованного между различными производителями, и создание действительно открытых систем, для которых стоимость переносимости приложений как между различными версиями одной операционной системы, так и между системами различных производителей была бы минимальной.

Основной задачей организации X/Open являлось согласование и утверждение стандартов для создания общего программного интерфейса и программной среды для приложений. В 1992 году появился документ, известный под названием X/Open Portability Guide версии 3 или XPG3, который включал POSIX 1003.1-1988 и стандарт на графическую систему X Window System, разработанную в Массачусетском институте технологии.

В дальнейшем интерфейсы XPG3 были расширены, включив базовые API систем BSD и System V (SVID), в том числе и архитектуру STREAMS. В результате была выпущена спецификация, ранее известная как Spec 11/70, а в 1994 году получившая название XPG4.2.

В 1993 году AT&T отказалась от производства операционных систем и продала свои права фирме Novell, которая незамедлительно возбудила иск против BSDi. В том же году 75 производителей систем и программного обеспечения, включая все крупные компании, выпускающие UNIX, согласились поручить разработку общего стандарта UNIX X/Open. Тогда Novell передала права на торговую марку Unix X/Open. Объединенный стандарт получил название Single Unix Standard version 1. В 1996 году OSF и X/Open объединились под названием The Open Group. В 1997 году последовала версия 2. В 1999 году размах The Open Group поглотил всю деятельность POSIX.

После 1997 года возникло стремление к дальнейшей концентрации усилий в разработке стандартов и объединении разрозненных документов в единое целое. В сентябре 1998 года была создана "Остинская группа" (Austin Group), целью которой стала выработка единого стандарта для UNIX-подобных операционных систем, Single UNIX Specification, SUS. В состав группы вошли представители комитета Стандартов Переносимых приложений (Portable Applications Standards Committee) IEEE, сотрудники The Open Group и представители 1-ого Объединенного технического комитета (Joint Technical Committee 1) ISO/IEC. Перед группой была поставлена задача пересмотреть, объединить и обновить следующие стандарты: ISO/IEC 9945-1, ISO/IEC 9945-2, IEEE Std 1003.1, IEEE Std 1003.2, а также Базовую Спецификацию The Open Group. В 2001 году The Open Group выпустила третью версию Единого Стандарта UNIX. Эта версия соединила в одном документе все направления стандартизации для API UNIX и набор тестов операционной системы на соответствие стандартам.

Стандарты для свободно распространяемых систем Single UNIX Specification и деятельность по стандартизации коммерческих поставщиков UNIX явно не устраивали производителей свободно распространяемых систем. Скорость обновления Linux и темпы сертификации по SUS катастрофически не соответствовали друг другу. Объединение производителей Linux объявило о создании собственного стандарта, Linux Standard Base, LSB. Позже была образована группа Free Standards Group, аналог The Open Group. Главным направлением стандартизации свободно распространяемых UNIX-систем стала стандартизация бинарного кода, а не стандартизация исходных кодов, как в стандартах коммерческих систем.

## Некоторые известные версии UNIX

Сегодня существуют десятки различных операционных систем, которые можно называть UNIX. В случае коммерческих версий, производитель операционной системы, как правило, является и производителем аппаратной платформы, для которой эта система предназначена. В качестве примеров можно привести операционные системы SunOS и Solaris фирмы Sun Microsystems, HP-UX фирмы Hewlett Packard, AIX фирмы IBM, IRIX фирмы

Silicon Graphics. Вполне естественно, что производитель хочет сделать операционную систему привлекательнее, чем у конкурентов, и не только за счет лучшей производительности, но и за счет расширений и дополнительных возможностей, отсутствующих у других. С другой стороны, производитель желает, чтобы его операционная система оставалась открытой: сегодня закрытые корпоративные решения отпугивают потребителя. Понятно, что в такой ситуации единства и борьбы противоположностей вряд ли найдется система, которую можно назвать "чистой системой UNIX". Да и вряд ли существует сегодня такое понятие. По мнению некоторых разработчиков, последней "чистой системой UNIX" являлась Седьмая редакция, сегодня же можно говорить только о наличии в операционной системе черт той или иной ветви — System V, BSD или OSF/1. Можно, например, сказать, что, с точки зрения администрирования и набора утилит, Digital UNIX представляет смесь System V и BSD UNIX, но с точки зрения интерфейсов и организации системы — это BSD.

Точно так же обстоит дело и в сообществе свободно распространяемых UNIX. Несомненно, Net BSD и FreeBSD, включив все полезное, что нашлось в System V, все же остаются представителями BSD-ветви. Но ОС Linux, заявленная изначально как System V, местами неотличима от FreeBSD.

Поэтому определение принадлежности конкретной операционной системы к той или иной генеалогической ветви носит весьма условный характер. С этой оговоркой в табл. В.3 приведены несколько индикаторов (с точки зрения пользователя и администратора) принадлежности UNIX одной из двух основных ветвей.

**Таблица В.3.** К какой генеалогической ветви принадлежит ваша система?

Индикатор	Типично для SVRx	Типично для xBSD
Имя ядра	<code>/unix</code>	<code>/vmunix</code>
Терминальная специализация	<code>/etc/inittab</code>	<code>/etc/tty</code>
Файлы инициализации системы	каталоги <code>/etc/rc*.d</code>	файлы <code>/etc/rc.*</code>
Конфигурация монтируемых файловых систем	<code>/etc/mnttab</code>	<code>/etc/mtab</code>
Обычный командный интерпретатор	<code>sh(1)</code> , <code>ksh(1)</code>	<code>csh(1)</code>
"Родная" файловая система	S5 (размер блока: 512–2048 байт), имена файлов $\leq 14$ символов	UFS (размер блока: 4 Кбайт–8 Кбайт), имена файлов $< 255$ символов

Таблица В.3 (окончание)

Индикатор	Типично для SVRx	Типично для xBSD
Система печати	lp(1), lpstat(1), cancel(1)	lpr(1), lpq(1), lprm(1M) (демон lpd)
Управление терминалами	terminfo(4)	termcap(4)
Отображение активности процессов	ps -ef	ps -aux

Ниже приведены краткие характеристики наиболее популярных версий UNIX.

### AIX

Версия UNIX фирмы IBM на базе SVR2 со многими чертами SVR4, BSD и OSF/1. Собственная система администрации (SMIT).

### HP-UX

Версия UNIX фирмы Hewlett Packard. Полностью 64-разрядная операционная система.

### IRIX

Версия UNIX фирмы Silicon Graphics, предназначенная для аппаратной платформы этого производителя (MIPS). Ранние версии системы включали много черт BSD UNIX, однако современную систему IRIX (6.x) скорее можно отнести к ветви System V Release 4. Полностью 64-разрядная операционная система.

### Digital UNIX

Версия системы OSF/1 фирмы Digital Equipment Corporation (DEC). В прошлом система называлась DEC OSF/1 и, по сути, являлась BSD UNIX. В то же время в ней есть много черт ветви System V. Полностью 64-разрядная операционная система, разработанная в первую очередь для аппаратной платформы Alpha, содержит все возможности, присущие современным UNIX. После присоединения DEC к Compaq — дочерней фирмы Hewlett Packard — Digital UNIX продолжила свою жизнь под названием Tru64 UNIX.

### SCO UNIX

В 1988 году компании Santa Cruz Operation (SCO), Microsoft и Interactive Systems завершили совместную разработку версии System V Release 3.2 для платформы Intel 386. В том же году SCO получила от AT&T лицензию на торговую марку, и операционная система стала называться SCO UNIX System V/386. В 1995 году компания SCO выпустила версию системы под названием SCO OpenServer Release 5 (кодовое название Everest) — UNIX-

версии SVR3.2 со многими чертами SVR4. Поддерживает более 900 аппаратных платформ, включая мультипроцессорные вычислительные системы, и более 2000 периферийных устройств.

### **SunOS и Solaris**

Версии UNIX фирмы Sun Microsystems для SPARC-архитектуры. SunOS до версии 4.1.4 включительно принадлежат BSD-ветви UNIX, версии старше 5.0 — реализации SVR4. Solaris это SunOS, дополненная графическим интерфейсом и собственной нумерацией версий. Система Solaris 1.0 включала в себя SunOS 4.1.1, Solaris 2.0 — SunOS 5.0, так что Solaris 2.5 — это SunOS 5.5. После Solaris 2.6 из нумерации были исключены цифра 2 и точка, то есть после Solaris 2.6 последовали Solaris 7, Solaris 8 и т. д. Поддерживаются многие распространенные аппаратные платформы, в том числе SPARC, UltraSPARC, x86, PowerPC и 64-разрядная архитектура.

### **NetBSD**

Выпускается с 1993 года. Свободно распространяемый клон ОС BSD, нацеленный на универсальность: разработчики стремятся воссоздать эту систему для любой аппаратной платформы. Вплоть до 1999 года распространялась исключительно по сети.

### **OpenBSD**

Выпускается группой разработчиков, отделившихся от NetBSD в 1994 году для создания системы повышенной безопасности, актуальной для поддержки основных сетевых сервисов. Выпускается с 1995 года. Ориентирована на сетевых администраторов, поддерживает эмуляцию других клонов UNIX для запуска бинарных файлов: Linux, FreeBSD, Solaris, BSD/OS, HP-UX и SunOS.

### **FreeBSD**

Свободно распространяемая BSD-система, изначально ориентированная на массового пользователя и платформу x86. Выпускается с 1993 года. Самая известная из свободно распространяемых BSD-систем. Выпускается для x86 и AMD64-совместимых, Alpha/AXP, IA-64, PC-98 и UltraSPARC архитектур.

### **Linux**

Свободно распространяемая под General Public License "народная" версия UNIX, сочетающая в себе черты BSD и System V. Не представляет собой единую систему — "Linux" такое же собирательное название, как и "UNIX". Существует несколько десятков компаний, формирующих разновидности этой системы, общим в которых является ядро, определяющее принадлежность системы к Linux. Среди них есть и коммерческие, и поставляющие свою продукцию бесплатно. Самые известные поставщики Linux — RedHat, Mandrake (ныне Mandriva), Slackware, Gentoo, SuSE, Debian. Кроме того, фирмы, производящие компьютеры, например Hewlett Packard и Silicon Graphics продвигают разработку Linux для своих аппаратных платформ.

# Глава 1



## Работа в операционной системе UNIX

Операционная система (ОС) UNIX используется на самых разнообразных аппаратных платформах — от персональных рабочих станций до мощных серверов с тысячами пользователей. Это связано с тем, что UNIX — многозадачная многопользовательская система, обладающая широкими возможностями.

С точки зрения пользователя, в операционной системе UNIX существуют два типа объектов: *файлы* и *процессы*. Все данные хранятся в виде файлов, доступ к периферийным устройствам осуществляется посредством чтения/записи в специальные файлы. Когда вы запускаете программу, ядро загружает соответствующий исполняемый файл, создает образ процесса и передает ему управление. Более того, во время выполнения процесс может считывать или записывать данные в файл. С другой стороны, вся функциональность операционной системы определяется выполнением соответствующих процессов. Работа системы печати или обеспечения удаленного доступа зависит от того, выполняются ли те или иные процессы в системе.

### Примечание

Конечно, возможность печати документа или работы в Интернете зависят также от наличия принтера или сетевого адаптера, правильности их настройки, работы соответствующих пользовательских и системных приложений, умения пользоваться этими приложениями и многого другого. В следующих главах мы затронем эти аспекты.

В этой главе мы познакомимся с пользовательской средой операционной системы UNIX; попробуем взглянуть на UNIX глазами обычного пользователя и администратора системы; не углубляясь во внутреннюю архитектуру, обсудим, что такое файлы и файловая система, рассмотрим ее организацию