

Дмитрий Кетов

ВНУТРЕННЕЕ УСТРОЙСТВО

LINUX

2-е издание

Санкт-Петербург

«БХВ-Петербург»

2021

УДК 004.451
ББК 32.973.26-018.2
К37

Кегов Д. В.

К37 Внутреннее устройство Linux. — 2-е изд., перераб. и доп. — СПб.: БХВ-Петербург, 2021. — 400 с.: ил.

ISBN 978-5-9775-6630-8

Книга представляет собой введение во внутреннее устройство операционной системы Linux. Все положения наглядно проиллюстрированы примерами, разработанными автором и проверенными им на практике. Рассмотрены основные подсистемы ядра и их сущности — файлы и файловые системы, виртуальная память и отображаемые файлы, процессы, нити и средства межпроцессного взаимодействия, каналы, сокеты и разделяемая память. Раскрыты дискреционный и мандатный (принудительный) механизмы контроля доступа, а также привилегии процессов. Подробно описано пользовательское окружение и интерфейс командной строки CLI, оконные системы X Window и графический интерфейс GUI, а также сетевая подсистема и служба SSH. Особое внимание уделено языку командного интерпретатора и его использованию для автоматизации задач эксплуатации операционной системы.

Во втором издании добавлены новые главы, описывающие графическую систему Wayland, контейнеры, виртуализацию и функционирование Linux как единой системы всех своих компонент, учтены изменения в последних версиях ОС, а также пожелания и отзывы читателей.

*Для студентов, пользователей, программистов
и системных администраторов Linux*

УДК 004.451
ББК 32.973.26-018.2

Группа подготовки издания:

Руководитель проекта	<i>Евгений Рыбаков</i>
Зав. редакцией	<i>Екатерина Сависте</i>
Компьютерная верстка	<i>Ольги Сергиенко</i>
Дизайн обложки	<i>Марины Дамбиевой</i>
Оформление обложки	<i>Карины Соловьевой</i>

"БХВ-Петербург", 191036, Санкт-Петербург, Гончарная ул., 20.

Оглавление

Введение	9
О чем эта книга?	9
Кому адресована книга.....	10
Принятые соглашения и обозначения.....	11
Методические рекомендации.....	12
Что должен знать читатель.....	13
Совет для начинающих.....	16
Глава 1. Архитектура ОС Linux.....	17
1.1. Обзор внутреннего устройства.....	17
1.2. Внеядерные компоненты: программы и библиотеки	19
1.3. Ядерные компоненты: подсистемы управления процессами, памятью, вводом-выводом, файлами.....	19
1.4. Трассировка системных и библиотечных вызовов.....	20
1.5. Интерфейсы прикладного программирования	22
1.6. В заключение	23
Глава 2. Пользовательское окружение ОС Linux	25
2.1. Командный интерфейс.....	25
2.2. Виртуальные терминалы	27
2.2.1. Псевдотерминалы.....	29
2.3. Управляющие символы.....	31
2.4. Управляющие последовательности.....	38
2.5. Основной синтаксис командной строки.....	40
2.5.1. Опции командной строки.....	42
2.6. Справочные системы.....	43
2.6.1. Система страниц руководства.....	43
2.6.2. Справочная система GNU	47
2.6.3. Встроенная справка командного интерпретатора.....	47
2.7. Пользователи и группы.....	48
2.7.1. Передача полномочий.....	50
2.7.2. Хранилища учетных записей.....	51
2.8. Переменные окружения и конфигурационные dot-файлы	52
2.9. В заключение	59

Глава 3. Подсистема управления файлами и вводом-выводом	61
3.1. Файлы и дерево каталогов	61
3.1.1. Путь и имена файлов	62
3.2. Типы файлов	63
3.2.1. Обычные файлы	64
3.2.2. Каталоги	65
3.2.3. Имена, данные, метаданные и индексные дескрипторы	66
3.2.4. Ссылки	67
3.2.5. Специальные файлы устройств	71
3.2.6. Именованные каналы и файловые сокеты	74
3.3. Файловые дескрипторы	75
3.4. Файловые системы	78
3.4.1. Файловые системы и процедура монтирования	78
3.4.2. Дисковые файловые системы	80
3.4.3. Сетевые файловые системы	80
3.4.4. Специальные файловые системы	82
3.4.5. Внеядерные файловые системы	84
3.5. Дискреционное разграничение доступа	87
3.5.1. Владельцы и режим доступа к файлам	88
3.5.2. Базовые права доступа и дополнительные атрибуты	90
Режим доступа новых файлов	91
Семантика режима доступа разных типов файлов	93
Дополнительные атрибуты	95
3.5.3. Списки контроля доступа POSIX	99
Групповая маска	101
Права по умолчанию	102
3.6. Мандатное (принудительное) разграничение доступа	103
3.6.1. Модуль принудительного разграничения доступа AppArmor	104
3.6.2. Модуль принудительного разграничения доступа SELinux	107
3.7. Дополнительные свойства файлов	111
3.7.1. Расширенные атрибуты файлов	111
3.7.2. Флаги файлов	113
3.8. В заключение	114
Глава 4. Управление процессами и памятью	115
4.1. Программы и библиотеки	115
4.1.1. Ядро Linux	118
4.2. Процессы и нити	121
4.3. Порождение процессов и нитей, запуск программ	125
4.3.1. Параллельные многопроцессные программы	129
4.3.2. Параллельные многонитевые программы	130
4.3.3. Двойственность процессов и нитей Linux	133
4.4. Дерево процессов	134
4.5. Атрибуты процесса	137
4.5.1. Маркеры доступа	137
4.5.2. Привилегии	140
4.5.3. Другие атрибуты	144

4.6. Классы и приоритеты процессов.....	144
4.6.1. Распределение процессора между процессами.....	144
4.6.2. Распределение устройств ввода-вывода между процессами.....	151
4.7. Память процесса	160
4.7.1. Виртуальная память	161
4.7.2. Отображение файлов в память	163
4.7.3. Потребление памяти.....	167
4.8. Механизм сигналов.....	171
4.8.1. Сеансы и группы процессов: управление заданиями	176
4.9. Межпроцессное взаимодействие.....	179
4.9.1. Неименованные каналы.....	180
4.9.2. Именованные каналы	181
4.9.3. Неименованные локальные сокеты	182
4.9.4. Именованные локальные сокеты	184
4.9.5. Разделяемая память, семафоры и очереди сообщений.....	185
Разделяемая память	185
Семафоры и очереди сообщений.....	190
4.10. В заключение	191

Глава 5. Программирование на языке командного интерпретатора..... 193

5.1. Интерпретаторы и их сценарии	193
5.2. Встроенные и внешние команды	195
5.3. Перенаправление потоков ввода-вывода	196
5.4. Подстановки командного интерпретатора.....	202
5.4.1. Подстановки имен файлов	202
5.4.2. Подстановки параметров	204
Переменные — именованные параметры	204
Позиционные параметры	207
Специальные параметры.....	208
5.4.3. Подстановки вывода команд.....	209
5.4.4. Подстановки арифметических выражений	211
5.5. Экранирование.....	214
5.6. Списки команд.....	218
5.6.1. Условные списки	219
5.6.2. Составные списки: ветвление	221
5.6.3. Составные списки: циклы.....	226
5.6.4. Функции.....	231
5.7. Сценарии на языке командного интерпретатора.....	234
5.8. Инструментальные средства обработки текста	237
5.8.1. Фильтр строк <code>grep</code>	238
5.8.2. Фильтр символов и полей <code>cut</code>	240
5.8.3. Процессор текстовых таблиц <code>awk</code>	241
5.8.4. Поточковый редактор текста <code>sed</code>	243
5.9. В заключение	247

Глава 6. Сетевая подсистема	249
6.1. Сетевые интерфейсы, протоколы и сетевые сокеты	249
6.2. Конфигурирование сетевых интерфейсов и протоколов	253
6.2.1. Ручное конфигурирование	253
6.2.2. Автоматическое конфигурирование	256
6.3. Служба имен и DNS/mDNS-резолверы	258
6.4. Сетевые службы	262
6.4.1. Служба SSH	262
6.4.2. Почтовые службы SMTP, POP/IMAP	270
6.4.3. Служба WWW	272
6.4.4. Служба FTP	274
6.4.5. Служба NFS	276
NFS-клиент	276
NFS-сервер	277
6.4.6. Служба SMB/CIFS	279
Имена NetBIOS	279
CIFS-клиенты	280
6.5. Средства сетевой диагностики	282
6.5.1. Анализаторы пакетов tcpdump и tshark	282
6.5.2. Сетевой сканер nmap	285
6.5.3. Мониторинг сетевых соединений процессов	286
6.6. В заключение	288
Глава 7. Графическая система X Window System	291
7.1. X-сервер	291
7.2. X-клиенты и X-протокол	293
7.3. Оконные менеджеры	298
7.3.1. Декорирование на клиентской стороне	301
7.4. Настольные пользовательские окружения	302
7.5. Библиотеки интерфейсных элементов	305
7.6. Расширения X-протокола	308
7.6.1. Расширение Composite и композитный менеджер	310
7.6.2. GLX, DRI и 3D-графика	311
7.7. Запуск X Window System	313
7.7.1. Локальный запуск X-клиентов	313
7.7.2. Дистанционный запуск X-клиентов	313
7.7.3. Управление X-дисплеями: XDMCP-менеджер и протокол	317
7.8. Программный интерфейс X Window System	318
7.8.1. Трассировка X-библиотек и X-протокола	318
7.8.2. 3D-графика и инфраструктура прямого рендеринга DRI	323
7.9. В заключение	329
Глава 8. Графическая система Wayland	331
8.1. Wayland-композитор	333
8.2. Wayland-клиенты и Wayland-протокол	334
8.3. Запуск графической среды на основе Wayland	339
8.4. В заключение	340

Глава 9. Контейнеры и виртуальные машины	343
9.1. Члутизация.....	344
9.2. Пространства имен.....	348
9.3. Контейнеризация: <code>gunc</code> и <code>docker</code>	353
9.4. Группы управления (<code>cgroups</code>).....	357
9.5. В заключение	361
Глава 10. От отдельных компонент — к системе.....	363
10.1. Как Linux загружается	363
10.2. Как обнаруживаются драйверы устройств	367
10.3. Как запускаются системные службы	370
10.4. Linux своими руками	379
10.5. В заключение	387
Заключение	389
Список литературы	391
Для удовольствия	391
Начинающим.....	391
Программистам.....	391
Бесстрашным.....	392
Предметный указатель.....	393

О чем эта книга?

Книга, которую вы держите в руках, адресована начинающим пользователям¹ операционной системы Linux и представляет собой *иллюстрированное* введение в ее *внутреннее устройство* — от ядра до сетевых служб и от утилит командной строки до графического интерфейса.

По моему скромному мнению, никакие книги не могут превратить читателя из новичка в эрудированного специалиста — ни обзорные, рассказывающие «ничего обо всем», ни узкоспециализированные, повествующие «все об одном». Без самостоятельного опыта, получаемого в результате исследования происходящих процессов и явлений, осмысление любого книжного знания практически невозможно.

Именно поэтому данная книга преследует всего одну важную цель — привить читателю *навыки* самостоятельного *исследования* постоянно *эволюционирующей* операционной системы Linux через *умения* пользоваться соответствующим инструментарием и на основе теоретических *знаний* о ее устройстве и философии².

Порядок и стиль изложения материала, иллюстративные изображения и листинги, приведенные в книге, являются результатом обобщения моего достаточно долгого опыта преподавания технических дисциплин, посвященных операционным системам в целом и операционной системе Linux в частности. Именно такой мне представляется картина «правильного» минимального набора знаний, умений и навыков, необходимых для построения качественной *ментальной модели* современной ОС Linux.

¹ Хотя термин «пользователь» зачастую ассоциируется с таким домашним любителем или офисным работником, но никак не с IT-специалистом, книга адресована абсолютно всем, кому интересно понимать то, как устроена операционная система изнутри. Именно для ее компетентного применения в своей профессиональной деятельности такое понимание в первую очередь нужно программистам и системным администраторам — разработчикам и эксплуатационникам информационных систем на базе Linux.

² W: [Философия UNIX].

Основная задача книги — иллюстративно изложить внутреннее устройство операционной системы Linux, связав «*теорию*» и «*практику*». В отличие от многих изданий, скупо и сухо излагающих теоретические аспекты построения операционной системы или излишне концентрирующихся на деталях конкретной реализации того или иного ее программного обеспечения, в книге рассматриваются *абстрактные* концепты внутреннего устройства ОС, иллюстративно подкрепляемые примерами анализа (а иногда и синтеза) ее *конкретных* компонентов и связей между ними.

Все части операционной системы рассматриваются в контексте типичных задач, решаемых при их помощи на практике, а сама иллюстрация проводится *посредством* соответствующего инструментария пользователя, администратора и разработчика.

Многие концепты и компоненты, инструменты и задачи будут иметь *общность*, характерную для всех **W**:[дистрибутивов Linux]¹, но их реализации могут существенно *различаться*. Поэтому, преследуя цель сохранить практическую значимость и жертвуя некоторой потерей общности излагаемого материала, повествование ведется в контексте отдельно взятой операционной системы, в качестве которой выступает дистрибутив **W**:[Ubuntu Linux] (версии 19.10, выпущенной в октябре 2019 года). Читатель может *свободно* скачать этот дистрибутив из Интернета и использовать в абсолютно любых целях, например для проработки примеров, приведенных в книге.

Выбор в пользу дистрибутива **W**:[Ubuntu Linux], в частности, сделан еще и по совокупности присущих ему свойств, как то: большое сообщество его пользователей (включая русскоговорящих), громадная пакетная база, регулярность обновлений и их стабильность, дружелюбность к начинающим и, наконец, широкая распространенность технических решений на его основе — от бизнес-решений и до сертифицированных разработок в оборонной промышленности.

Кому адресована книга

Понимание внутреннего устройства важно как в системном администрировании для направленной диагностики проблем эксплуатации информационных систем, построенных на платформе Linux, так и для разработки программного обеспечения, учитывающего особенности ее внутреннего устройства.

Поэтому книга адресуется абсолютно всем, кто не любит использовать непрозрачный «черный ящик», кому интересна механика всех этих шестеренок, что вращаются внутри Linux, ежедневно решая задачи миллионов пользователей по всему

¹ А зачастую и для всех операционных систем семейства **W**:[UNIX], к которому Linux принадлежит.

Кинжал  предупреждает о выполнении некоторой деструктивной операции, а значки часов     символизируют ожидание завершения достаточно длительной операции.

Для удобства чтения листингов значки   указывают на «замещающий» вывод (т. е. без скроллинга), круговые стрелки  обозначают очередную итерацию некоторого цикла, а значок  указывает на его окончание. Ну и, наконец, разнообразные стрелки     связывают отдельные части листинга друг с другом.

Методические рекомендации

Многие иллюстративные листинги, приведенные в книге, нужно воспринимать не как «дополнение» к основному тексту, а как сам основной текст, просто (пока) на незнакомом и непонятном языке, а саму книгу — как учебник нового иностранного языка.

Получение читателем собственного отклика от взаимодействия с операционной системой представляется мне наиболее значимым результатом освоения содержимого этой книги. Поэтому чтение стоит непременно сопровождать самостоятельным выполнением команд листингов, т. к. успешное изучение нового иностранного языка никак невозможно без его непосредственного практического использования в процессе изучения.

Не менее важным для построения целостных взаимоотношений с системой является самостоятельное изучение первоисточников знаний¹ — ее внутренней документации, страниц руководства `man(1)`.

Для выполнения команд листингов и доступа к внутренним справочникам потребуется сама операционная система, что несложно получить, например установив настольную версию Ubuntu Linux в виртуальную машину `W:[VirtualBox]`.

Дополнительную пользу принесет ознакомление и с дополнительными источниками информации в Интернете, например со статьями на <http://help.ubuntu.ru>, а также поиск ответов на возникающие вопросы на <http://forum.ubuntu.ru>, <http://askubuntu.com> и даже на <http://UNIX.stackexchange.com> и <http://stackoverflow.com>.

Кроме текстовых источников информации в Интернете можно воспользоваться услугами таких очаровательных онлайн-сервисов, как <http://explainshell.com> для объяснений конструкций и <http://www.shellcheck.net> для проверки проблем в конструкциях командного интерпретатора. Для проверки примеров из книги и разработки собственных сценариев командного интерпретатора незаменимую помощь может оказать сервис http://www.tutorialspoint.com/execute_bash_online.php.

¹ Это обязательно потребует знания технического английского языка, но без него вообще трудно рассчитывать на сколько-нибудь значительные успехи в отрасли информационных технологий.

Что должен знать читатель

В операционной системе Linux программное обеспечение поставляется в виде так называемых *пакетов* (*package*) — специальным образом подготовленных архивов, содержащих само *программное* обеспечение, его *конфигурационные* файлы, его данные и *управляющую* информацию. Управляющая информация пакета включает контрольные суммы устанавливаемых файлов, зависимости устанавливаемого пакета от других пакетов, краткое описание пакета, сценарии установки, сценарии удаления пакета и прочие данные, необходимые *менеджеру* пакетов.

В примерах, приведенных в листингах, часто встречаются программы, которые, возможно, не будут установлены «по умолчанию», поэтому важно знать и понимать, как устроены подсистема управления установкой и удалением программного обеспечения — так называемые менеджеры пакетов и зависимостей¹.

Менеджер пакетов производит непосредственную установку и удаление пакетов программного обеспечения, а также ведет их учет в системе. Вспомогательная «грязная» работа по подбору зависящих друг от друга пакетов, получению их из *репозиториев*² (например, скачивание с FTP/HTTP-серверов), выбору правильных версий пакетов, определению правильного их порядка установки достается *менеджеру зависимостей*.

При помощи менеджера пакетов (листинг В1) всегда можно узнать имя пакета **❶**, в который входит та или иная *установленная* компонента операционной системы (например, `/bin/date`), или, наоборот, узнать список компонент **❷**, *установленных* из указанного пакета (например, `coreutils`).

Листинг В1. Пакетный менеджер dpkg

```
bart@ubuntu:~$ which -a date
```

```
/usr/bin/date
```

```
/bin/date
```

```
❶ bart@ubuntu:~$ dpkg -S /bin/date
```

```
coreutils: /bin/date
```

```
❷ bart@ubuntu:~$ dpkg -L coreutils
```

```
...
```

```
...
```

```
...
```

¹ Условно, можно выделить две ветви операционной системы Linux — ветвь `debian`, к которой относятся дистрибутивы `W:[Debian]` и `W:[Ubuntu]`, и ветвь `redhat`, куда нужно отнести `W:[RHEL]`, `W:[CentOS]` и `W:[Fedora]`. В `debian`-ветви используется пакетный менеджер `dpkg` и построенные над ним менеджеры зависимостей `apt`, `aptitude`, `synaptic` и `software-center`, а в ветви `redhat` — пакетный менеджер `rpm` и основной менеджер зависимостей `yum`.

² Хранилище пакетов у разработчиков дистрибутива Linux.

```

/bin/chmod
/bin/chown
/bin/cp
/bin/date
/bin/dd
/bin/df
...
bart@ubuntu:~$ dpkg -s coreutils
Package: coreutils
Essential: yes
Status: install ok installed
Priority: required
Section: utils
Installed-Size: 7196
Maintainer: Ubuntu Developers <ubuntu-devel-discuss@lists.ubuntu.com>
Architecture: amd64
Multi-Arch: foreign
Version: 8.30-3ubuntu2
Pre-Depends: libc1 (>= 2.2.23), libattr1 (>= 1:2.4.44), libc6 (>= 2.28), libselinux1 (>= 2.1.13)
Description: GNU core utilities
 This package contains the basic file, shell and text manipulation
 utilities which are expected to exist on every operating system.
.
 Specifically, this package includes:
 arch base64 basename cat chcon chgrp chmod chown chroot cksum comm cp
 csplit cut date dd df dir dircolors dirname du echo env expand expr
...
Homepage: http://gnu.org/software/coreutils
Original-Maintainer: Michael Stone <mstone@debian.org>

```

Если при попытке выполнить ту или иную команду операционной системы Ubuntu Linux (это одна из причин, по которой иллюстрация в книге ведется именно с ее помощью) обнаружится, что нужный пакет с программным обеспечением не установлен, то при наличии доступа в Интернет можно тривиальным способом доустановить недостающие компоненты (листинг В2).

Листинг В2. Установка недостающего программного обеспечения

```
bart@ubuntu:~$ finger
```

Команда «finger» не найдена, но может быть установлена с помощью:

```
• sudo apt install finger
```

```

bart@ubuntu:~$ sudo apt install finger
[sudo] password for bart:
Чтение списков пакетов... Готово
Построение дерева зависимостей
Чтение информации о состоянии... Готово
Следующие НОВЫЕ пакеты будут установлены:
  finger
Обновлено 0 пакетов, установлено 1 новых пакетов, для удаления отмечено 0 пакетов,
и 11 пакетов не обновлено.
Необходимо скачать 16,9 kB архивов.
После данной операции объём занятого дискового пространства возрастёт на 51,2 kB.
Пол:1 http://ru.archive.ubuntu.com/ubuntu eoan/universe amd64 finger amd64 0.17-17 [16,9 kB]
Получено 16,9 kB за 0с (200 kB/s)
      ...
Подготовка к распаковке .../finger_0.17-17_amd64.deb ...
Распаковывается finger (0.17-17) ...
Настраивается пакет finger (0.17-17) ...
Обрабатываются триггеры для man-db (2.8.7-3) ...

```

В редких случаях, когда нужно узнать, с каким, даже еще неустановленным, пакетом программного обеспечения поставляется тот или иной файл, может выручить утилита **apt-file(1)** (❶, листинг В3). В обратную сторону посмотреть список файлов, входящих в еще неустановленный пакет, можно этой же утилитой (❷).

Листинг В3. В каком пакете файл (файлы)?

```

❶ bart@ubuntu:~$ apt-file search bin/7z
Finding relevant cache files to search ...E: The cache is empty. You need to run "apt-file
update" first.
bart@ubuntu:~$ sudo apt-file update
Сущ:1 http://ru.archive.ubuntu.com/ubuntu eoan InRelease
Пол:2 http://ru.archive.ubuntu.com/ubuntu eoan-updates InRelease [97,5 kB]
Пол:3 http://ru.archive.ubuntu.com/ubuntu eoan-backports InRelease [88,8 kB]
      ...
Получено 91,9 MB за 37с (2 515 kB/s)
Чтение списков пакетов... Готово
Построение дерева зависимостей
Чтение информации о состоянии... Готово
bart@ubuntu:~$ apt-file search bin/7z
p7zip: /usr/bin/7zr
p7zip-full: /usr/bin/7z
p7zip-full: /usr/bin/7za

```

```
bart@ubuntu:~$ apt-file show p7zip
p7zip: /usr/bin/7zr
p7zip: /usr/bin/p7zip
p7zip: /usr/lib/p7zip/7zr
p7zip: /usr/share/doc/p7zip/NEWS.Debian.gz
p7zip: /usr/share/doc/p7zip/README.Debian
p7zip: /usr/share/doc/p7zip/changelog.Debian.gz
p7zip: /usr/share/doc/p7zip/copyright
p7zip: /usr/share/man/man1/7zr.1.gz
p7zip: /usr/share/man/man1/p7zip.1.gz
```

Совет для начинающих

И напоследок, самый важный совет для начинающих — начните!



1.1. Обзор внутреннего устройства

Операционная система (ОС) в общем и **W:[Linux]** в частности (рис. 1.1) представляет собой набор специализированных *программных* средств, обеспечивающих *доступ* потребителей (пользователей) к ресурсам (устройствам) и *распределение* последних между потребителями наиболее *удобным* и *эффективным* способом. Под ресурсами в первую очередь понимают аппаратные средства, такие как центральные процессоры, память, устройства ввода-вывода, дисковые и прочие накопители и другие периферийные устройства. Во вторую очередь к ресурсам относят такие «виртуальные» сущности (на рис. 1.1 не показаны), как процессы, нити, файлы, каналы и сокеты, семафоры и мьютексы, окна и прочие артефакты самой операционной системы, которые не существуют вне ее границ.

Как и во многих других операционных системах, в Linux выделяют два главных режима работы ее программных средств — ядерный режим (*kernel mode*), он же пространство ядра (*kernel space*), и пользовательский, внеядерный режим (*user mode*), или же пользовательское пространство (*user space*). Основное различие этих двух режимов состоит в привилегиях доступа к аппаратным средствам — памяти и устройствам ввода-вывода, к которым разрешен полный доступ из режима ядра и ограниченный доступ из режима пользователя.

Совокупность работающих в ядерном режиме программ называют *ядром*, которое в Linux состоит из основы (или же остова) и присоединяемых к ней объектов — динамически загружаемых модулей (подробнее *см. разд. 4.1.1*). Несмотря на компонентность, **W:[Ядро Linux]** относят к классу *монолитных* в силу того, что все его компоненты выполняются с одинаковыми (ядерными) привилегиями. В классе *микроядерных* систем, наоборот, компоненты ядра работают с разными привилегиями: основная компонента — микроядро (планировщик процессов/нитей и менеджер памяти) — с наибольшими привилегиями, менеджер ввода-вывода, драйверы устройств, файловые системы, сетевые протоколы и пр. — с другими, обычно меньшими привилегиями (возможно даже, с привилегиями пользовательского режима).

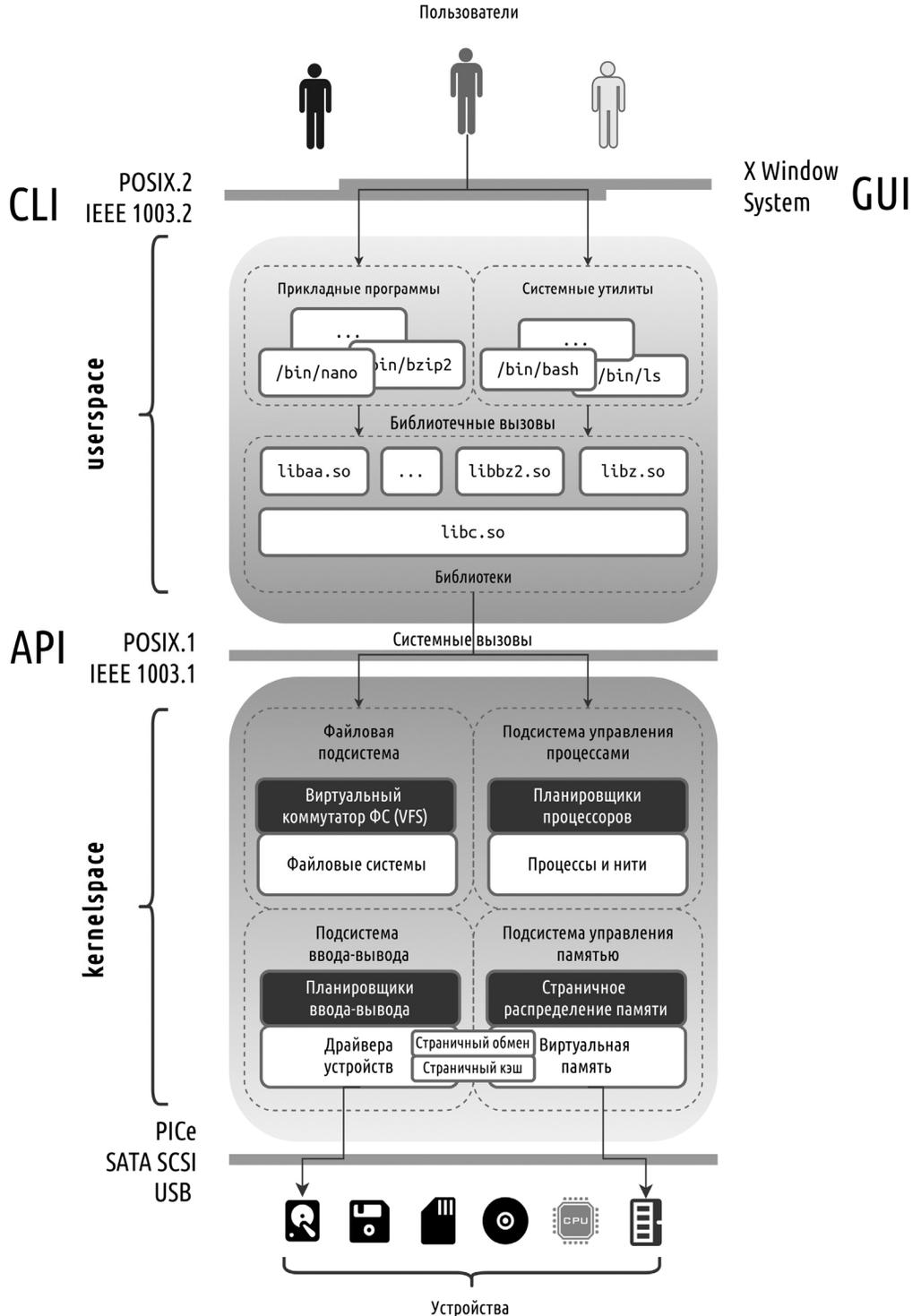


Рис. 1.1. Компоненты операционной системы Linux

Заключение

Изначально содержимое этой книги задумывалось как практическая иллюстрация внутреннего устройства различных компонент операционной системы, в равной степени подкрепляемая примерами их анализа и синтеза.

Аналитическая часть удалась примерно в той степени, в которой и задумывалась, а примеры синтеза выходили достаточно блеклыми, особенно по сравнению с теми, которые уже представлены в соответствующих изданиях, целиком посвященных программному окружению и разработке. Более того, сделать эти примеры доступными для их успешного понимания читателем, не владеющим базовыми знаниями по программированию на языке C (Си), не удалось вовсе.

Поэтому все эти не столь многочисленные примеры перекочевали в электронное приложение к книге, которое в виде архива доступно на сайте издательства www.bhv.ru со страницы книги или по ссылке:

<ftp://ftp.bhv.ru/9785977566308.zip>

Большинство примеров являются простейшими и достаточно грубыми моделями программ, уже существующих в операционной системе.

Так, например, работу с файлами и ссылками иллюстрируют модели программ `touch(1)`, `ln(1)`, `rm(1)`, `mv(1)`, `cp(1)`, а работу со специальными файлами устройств — модель программы `eject(1)`, открывающая лоток привода CD/DVD, и специальная программа `mouse`, обрабатывающая перемещения манипулятора «мышь». Управление процессами продемонстрировано при помощи программы `shell`, моделирующей простейший командный интерпретатор, а управление нитями представляет программа `pdu`, параллельная модель измерителя файлов в дереве каталогов `du(1)`. Сетевое взаимодействие иллюстрируется простейшими TCP-клиентом, моделирующим `telnet(1)` (или `netcat(1)`), и TCP-сервером, моделирующим сервер службы удаленного доступа, похожей на `telnetd(8)` (или `sshd(8)`, только без криптозащиты).

Завершаются иллюстрации четырьмя вариантами простейшей GUI-программы `W:[Hello_world!]`, выполненной на языках C, Python и Tcl и являющейся X-клиентом оконной системы X Window, использующим ее графические библиотеки интерфейсных элементов Qt, Gtk+ и Tk.

Кроме собственно иллюстрационных программ, архив содержит вспомогательные сценарии на языке командного интерпретатора, формирующие «программного гида», который поможет откомпилировать и запустить эти программы и вдобавок прокомментирует результат их выполнения в тех частях, где анализ результата не будет тривиальным. Сами сценарии «программного гида» тоже являются живыми примерами практик и приемов программирования на языке командного интерпретатора, анализируя которые можно расширить восприятие материала соответствующей главы, посвященной этой теме.

Как и ожидалось с самого начала, эта книга является лишь иллюстративным введением во внутреннее устройство ОС Linux, оставляющим за кадром массу специфичных деталей реализации конкретных версий ее ядер или ее дистрибутивов. Выстроив «правильную» начальную ментальную модель операционной системы, все эти тонкости можно без значительных усилий почерпнуть из собственной практики, документации, интернет-общения или других книг (см. список литературы).

Список литературы

Для удовольствия

Торвальдс Л., Даймонд Д. Just for Fun. Рассказ нечаянного революционера: Пер. с англ. — М.: Эксмо-Пресс, 2002. 288 с. — ISBN 5-04-009285-7.

Начинающим

Граннеман С. Linux. Карманный справочник: Пер. с англ. — М.: Вильямс, 2015. 416 с. — ISBN 978-5-8459-1956-4.

Пайк Р., Керниган Б. UNIX. Программное окружение: Пер. с англ. — СПб.: Символ-плюс, 2003. 416 с. — ISBN 5-93286-029-4.

Реймонд Э. Искусство программирования для UNIX: Пер. с англ. — М.: Вильямс, 2016. 544 с. — ISBN 978-5-8459-2064-5.

Тейнсли Д. Linux и UNIX: программирование в shell. Руководство разработчика: Пер. с англ. — СПб.: БХВ-Петербург, 2001. 464 с. — ISBN 5-7315-0114-9.

Фридл Дж. Регулярные выражения: Пер. с англ. — СПб.: Символ-плюс, 2008. 608 с. — ISBN 5-93286-121-5.

Dougherty D., Robbins A. sed & awk. — O'Reilly Media, 1997. 432 p. — ISBN: 1-56592-225-5.

Программистам

Керриск М. Linux API. Исчерпывающее руководство: Пер. с англ. — СПб.: Питер, 2019. 1248 с. — ISBN: 978-5-4461-0985-2.

Лав Р. Linux. Системное программирование: Пер. с англ. — СПб.: Питер, 2016. 448 с. — ISBN 978-5-496-01684-1.

Роббинс А. Linux: программирование в примерах: Пер. с англ. — СПб.: КУДИЦ-Пресс, 2008. 656 с. — ISBN:978-5-91136-056-6.

Стивенс У. Р., Раго С. UNIX. Профессиональное программирование:
Пер. с англ. — СПб.: Символ-плюс, 2014. 1104 с. —
ISBN 978-5-93286-216-2.

Бесстрашным

Бовет Д., Чезати М. Ядро Linux: Пер. с англ. — СПб.: БХВ-Петербург, 2007.
1104 с. — ISBN 0-596-00565-2, 978-5-94157-957-0.

Лав Р. Ядро Linux. Описание процесса разработки: Пер. с англ. — М.: Вильямс,
2014. 496 с. — ISBN 978-5-8459-1944-1.

Kroah-Hartman G. Linux Kernel in a Nutshell. — O'Reilly Media, 2006. 202 p. —
ISBN 978-0-596-10079-7.

McKellar J., Rubini A., Corbet J., Kroah-Hartman G. Linux Device Drivers,
3rd Edition. — O'Reilly Media, 2005. 640 p. — ISBN 978-0-596-00590-0.

Предметный указатель

А

Авторизация 48
Адрес виртуальный 160
Алгоритм 115
◇ планирования 145
 ▫ вытесняющий 145
Аутентификация 49

Б

Библиотека 116

В

Взаимодействие межпроцессное 179
Вызов:
◇ библиотечный 19
◇ системный 19
Выражение регулярное 237

Г

Группа процессов 176

Д

Демон 136
Дерево:
◇ каталогов 78
◇ процессов 134
Дескриптор файловый 75
Драйвер терминала 73

З

Задача 133

И

Имя:
◇ абсолютное путевое 62
◇ относительное путевое 62
Интерпретатор 115

К

Кадр страничный 160
Канал:
◇ именованный 63, 74, 181
◇ неименованный 180, 199
Каталог 63, 65
◇ рабочий 62
Ключ:
◇ закрытый 263, 266
◇ открытый 263, 266
◇ сеансовый 263
Компилятор 115

Л

Лидер:
◇ группы 176
◇ сеанса 176

М

Маркер доступа:
◇ DAC 137
◇ MAC 139
Метасимвол 202
Многозадачность 121
Монтирование 78

Н

Нить 19, 123

О

Обмен страничный 162
 Обработка конвейерная 198
 Однозадачность 121
 Отображение страничное 160

П

Память:

- ◇ виртуальная 162
- ◇ разделяемая 186

Параметр:

- ◇ позиционный 204, 207
- ◇ специальный 204, 208

Переменная 204

- ◇ окружения 53, 204

Планировщик 144

Подсистема:

- ◇ ввода-вывода 19
- ◇ управления памятью 19
- ◇ ядра, файловая 20

Подстановка:

- ◇ вывода команд 209
- ◇ значений параметров 204
- ◇ имен файлов 202

Политика SELinux 107

Поток текстовый 237

Программа 115

Прозрачность сетевая 291

Протокол сетевой 251

Процесс 19, 121, 122

- ◇ демон 136

- ◇ прародитель 134

- ◇ прикладной 135

- ◇ системный 136

Р

Режим:

- ◇ доступа к файлу 90
- ◇ пользовательский 17
- ◇ ядерный 17

С

Сеанс 176

Семафор 190

Сигнал 171

Символ управляющий 25

Система псевдофайловая 82

Служба имен 188

Смесь мультипрограммная 122

Сокет 182

- ◇ именованный локальный 184
- ◇ неименованный локальный 182
- ◇ сетевой 251
- ◇ файловый 63, 74

Список команд 218

- ◇ простой 218
 - асинхронный 218
 - синхронный 218
- ◇ условный 219, 221
- ◇ циклический 226

Ссылка:

- ◇ жесткая 67
- ◇ права доступа 94
- ◇ символическая 69
- ◇ сирота 69

Страница памяти 160

Сценарий на языке командного интерпретатора 234

Т

Таблица:

- ◇ имен 65
- ◇ страниц 160

Терминал 25

- ◇ алфавитно-цифровой 25
- ◇ аппаратный 73
- ◇ виртуальный 73
- ◇ дисплейный 25
- ◇ печатающий 25

У

Устройство:

- ◇ бесконечно нулевое 74
- ◇ всегда полное 74
- ◇ всегда пустое 74

Ф

Файл:

- ◇ FIFO 74
 - ◇ обычный 63, 64
 - ◇ специальный:
 - блочный 71
 - символьный 71
 - ◇ устройства, специальный 63
- Файловая система:
- ◇ дисковая 80
 - ◇ сетевая 80
- Функция 231

Ц

Цикл с параметром 226

Ш

Шифрование симметричное 263

Э

Экранирование:

- ◇ сильное 216
 - ◇ слабое 216
- Электронная почта 270

Я

Ядро 17, 118, 136



ЕСЛИ ВЫ ДОЛЖИСТАЛИ ФАЙЛ
ДО ЭТОЙ СТРАНИЦЫ,
ПОЛУЧИТЕ **СКИДКУ 30%**
НА КНИГУ
ПО ПРОМОКОДУ

LINUX2020

АКЦИЯ ДЕЙСТВУЕТ ДО 30 НОЯБРЯ 2020