

ОП.14
ОСНОВЫ
функционирования UNIX -
СИСТЕМ

30 ЭИТННЗ
ЗАНЯТИЕ 03

Файловая система

Организация данных в операционных системах определяется **структурой файловой системы**.

Файловая система является одной из важнейших функциональных частей UNIX. Она **обеспечивает**:

- **хранение данных**, принадлежащих операционной системе и пользователям, а также обеспечение их целостности;
- **эффективный доступ к данным**, находящимся на запоминающих устройствах длительного хранения (накопителях на жестких, на оптических дисках и т. д.);

Файловая система

Файловая система UNIX также **обеспечивает:**

- **эффективное выполнение операций** восстановления данных в случае их повреждения;
- **единообразный механизм доступа** ко всем объектам файловой системы.

Файловая система

Файловая система UNIX устроена таким образом, чтобы соответствовать одной из основных концепций этой операционной системы — **представлению всех объектов операционной системы, независимо от их природы, в виде файлов.**

Здесь привычное для многих пользователей понятие файла приобретает более широкий смысл.

В привычных для многих операционных системах MS-DOS и в ранних версиях Windows термином "файл" обозначался двоичный образ данных, записанных на диск.

Файловая система

В UNIX к файлам относятся:

- дисковые файлы или каталоги,
- программные объекты — именованные и неименованные каналы,
- сокеты (гнёзда) – названия программных интерфейсов для обеспечения обмена данными между процессами,
- терминальные линии,
- а также физические устройства ввода/вывода, такие, например, как накопители на жестких дисках, параллельный и последовательный порты и т. д.

Файловая система

При этом устройства ввода/вывода представлены специальными файлами, которые имеют название ***файлов устройств***.

Подобное представление означает, что ко всем объектам файловой системы можно обращаться, используя стандартный программный интерфейс, предоставляемый UNIX.

Например, к дисковому файлу, именованному каналу или параллельному порту можно обращаться, используя системные вызовы `open ()`, `read ()`, `write ()` и `close ()`.

Файловая система

Сказанное касается всех пользовательских и части системных программ, для которых собственно и создана такая модель файловой системы.

Операционная система UNIX на уровне ядра и драйверов устройств **обрабатывает запросы к разным устройствам**, дифференцируя их типы и используя различные подходы.

Файловая система

Есть еще одна **причина**, по которой выбрана именно такая архитектура файловой системы, — это необходимость обеспечить **надежность системы**.

Вспомним, что в UNIX пользовательские приложения **не могут** обращаться **напрямую** к аппаратным устройствам иначе как через системные вызовы.

Поэтому разработчики операционной системы использовали один и тот же программный интерфейс как для дисковых файлов, так и для устройств ввода/вывода.

Файловая система

Лучше понять механизм взаимодействия пользовательского приложения и объекта файловой системы UNIX позволяет следующий рисунок.



Файловая система

Здесь показана схема обращения из программы пользователя к двум объектам файловой системы:

- параллельному порту компьютерной системы (файл устройства `/dev/lp0`), к которому подсоединен принтер, для чего используется системный вызов `open ()`:

```
open ("/dev/lp0", O_RDWR);
```

- дисковому файлу `file`:

```
open ("file", O_RDONLY);
```

Файловая система

Для программы пользователя обращение к разным по природе типам устройств (параллельный порт и жесткий диск) **прозрачно**, т. е. программно они различаются лишь именами устройств.

В то же самое время для операционной системы оба этих системных вызова могут обрабатываться **различными способами**, поскольку оба устройства управляются различными драйверами, оперирующими как различными типами данных (символьными и блочными), так и различными аппаратными интерфейсами.

Файловая система

Подобная структура файловой системы очень удобна для разработчиков программного обеспечения, поскольку она **обеспечивает унифицированный программный интерфейс** для работы с объектами файловой системы.

Рассмотрим более подробно основные типы файлов, используемые в UNIX.

Файловая система

К ним относятся:

- **бинарные файлы**, содержащие двоичные данные (например, программы или текст), записанные на жесткий диск,
- **специальные файлы устройств**,
- **сокеты**,
- **именованные каналы**,
- а также **символические и жесткие ссылки**.

Файловая система

Бинарные файлы

Бинарные файлы содержат наборы двоичных битов, в которых в закодированном виде находится та или иная информация, например, текст, рисунки, программы, аудиоданные и т. д.

В большинстве случаев, когда используют термин "**файл**", то имеют в виду **именно такие файлы**.

Данные, записанные в такие файлы, должны интерпретироваться:

- или **операционной системой**, если это программный файл,
- или другими **приложениями**.

Файловая система

Файлы устройств

Специальные *файлы устройств* позволяют операционной системе UNIX и другим программам **взаимодействовать** с аппаратными средствами и периферийными устройствами системы.

Файлы устройств **не эквивалентны драйверам** устройств — драйверы обеспечивают доступ к устройству на уровне аппаратно-программного интерфейса, преобразуя пакеты запросов, поступающие от ядра в соответствующие инструкции процессора.

Файловая система

Файлы устройств

Файлы устройств можно представить как шлюзы, через которые драйвер получает запросы. Когда ядро получает запрос к файлу устройства, оно просто передает этот запрос соответствующему драйверу.

Структура файла устройства **отличается** от той, которую имеет файл данных. Сами файлы устройств обрабатываются базовыми средствами файловой системы, а их характеристики записываются на диск.

Взаимодействие пользовательской программы, файла устройства и драйвера было показано на рисунке (обращение к параллельному порту, которому соответствует файл устройства `/dev/lp0`).

Файловая система

Файлы устройств

Файлы устройств могут иметь один из двух типов: **файл байт-ориентированного** ("символьного") или **блок-ориентированного** ("блочного") устройств.

Файлы байт-ориентированных устройств позволяют связанным с ними драйверам выполнять собственную буферизацию ввода/вывода, в то время как файлы блок-ориентированных устройств обрабатываются драйверами, манипулирующими большими блоками данных и возлагающими буферизацию на ядро. Некоторые типы аппаратных средств, например, накопители на жестких дисках, представляются файлами обоих типов.

Файловая система

Файлы устройств

Поскольку в системе могут присутствовать устройства **одного типа**, то файлы устройств распознаются **по двум номерам** — старшему и младшему.

Старший номер устройства информирует ядро, к какому **драйверу** относится данный файл, а младший номер сообщает драйверу, к какому физическому **устройству** следует обращаться.

Так, например, старший номер устройства 6 в Linux обозначает драйвер параллельного порта. Первый параллельный порт `/dev/lp0` будет иметь старший номер 6 и младший номер 0.

Файловая система

Файлы устройств

Некоторые драйверы используют младший номер устройства **нестандартным способом.**

Например, драйверы накопителей на магнитных лентах часто руководствуются им при выборе плотности записи, а также определяют, нужна ли перемотка ленты после закрытия файла устройства.

В некоторых системах драйвер терминала, управляющий последовательными устройствами, использует младшие номера устройств для идентификации модемов.

Файловая система

Сокет

Еще один тип файла UNIX — *сокет* (гнездо). Чаще всего сокеты используются для взаимодействия между независимыми процессами, выполняющимися на одной и той же (локальные сокеты) или на разных системах (сокеты протокола TCP).

Сокеты TCP позволяют взаимодействовать процессам, выполняющимся на разных машинах в сетях TCP/IP, хотя могут применяться для обмена данными между процессами, работающими на одном и том же хосте (в этом случае используется так называемый *интерфейс обратной связи*, который часто называют *loopback interface*, он имеет IP-адрес 127.0.0.1).

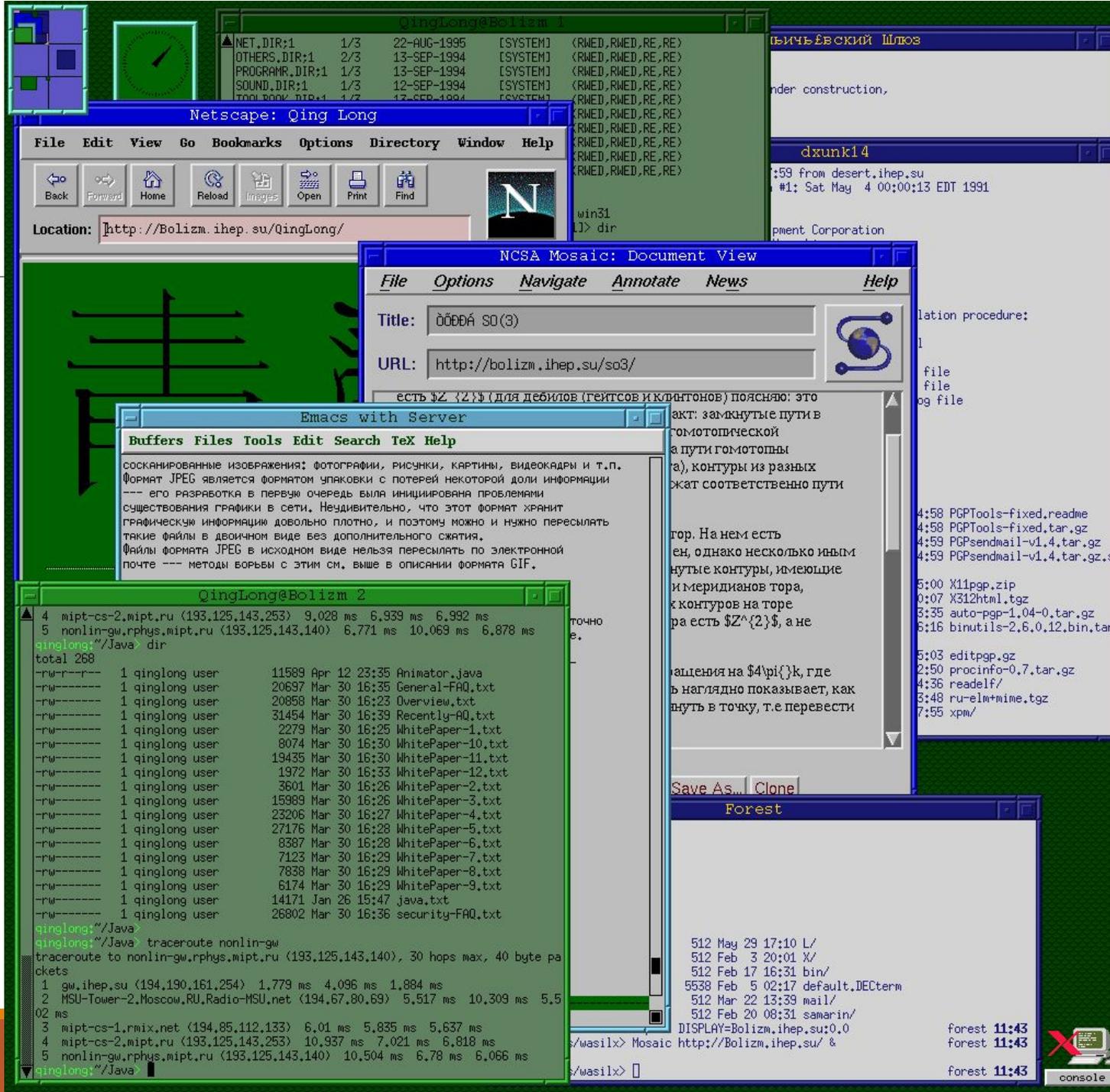
Файловая система

Сокет

Хост (от англ. host — «хозяин, принимающий гостей») — любое **устройство**, предоставляющее **сервисы** формата «клиент-сервер» в режиме сервера по каким-либо интерфейсам и уникально определённое на этих интерфейсах.

Взаимодействие процессов посредством сокетов очень широко применяется в самой операционной системе. Например, система печати, система X Window активно использует сокет для обмена данными.

X Window System — оконная система, обеспечивающая стандартные инструменты и протоколы для построения графического интерфейса пользователя. Используется в UNIX-подобных ОС.



NET_DIR:1	1/3	22-AUG-1995	[SYSTEM]	(RMED,RMED,RE,RE)
OTHERS_DIR:1	2/3	13-SEP-1994	[SYSTEM]	(RMED,RMED,RE,RE)
PROGRAMR_DIR:1	1/3	13-SEP-1994	[SYSTEM]	(RMED,RMED,RE,RE)
SOUND_DIR:1	1/3	12-SEP-1994	[SYSTEM]	(RMED,RMED,RE,RE)
TOOL_DIR:1	1/3	12-SEP-1994	[SYSTEM]	(RMED,RMED,RE,RE)

Netscape: Qing Long

File Edit View Go Bookmarks Options Directory Window Help

Back Forward Home Reload Images Open Print Find

Location:

NCSA Mosaic: Document View

File Options Navigate Annotate News Help

Title:

URL:

Emacs with Server

Buffers Files Tools Edit Search TeX Help

сосканированные изображения: фотографии, рисунки, картины, видеокдры и т.п.
 Формат JPEG является форматом упаковки с потерей некоторой доли информации
 --- его разработка в первую очередь была инициирована проблемами
 существования графики в сети. Неудивительно, что этот формат хранит
 графическую информацию довольно плотно, и поэтому можно и нужно пересылать
 такие файлы в двоичном виде без дополнительного сжатия.
 Файлы формата JPEG в исходном виде нельзя пересылать по электронной
 почте --- методы борьбы с этим см. выше в описании формата GIF.

QingLong@Bolizm 2

```

4 mipt-cs-2.mipt.ru (193.125.143.253) 9,028 ms 6,939 ms 6,992 ms
5 nonlin-gw.rphys.mipt.ru (193.125.143.140) 6,771 ms 10,069 ms 6,878 ms
qinglong:~/Java dir
total 268
-rw-r--r-- 1 qinglong user 11589 Apr 12 23:35 Animator.java
-rw-r--r-- 1 qinglong user 20697 Mar 30 16:35 General-FAQ.txt
-rw-r--r-- 1 qinglong user 20858 Mar 30 16:23 Overview.txt
-rw-r--r-- 1 qinglong user 31454 Mar 30 16:39 Recently-AQ.txt
-rw-r--r-- 1 qinglong user 2279 Mar 30 16:25 WhitePaper-1.txt
-rw-r--r-- 1 qinglong user 8074 Mar 30 16:30 WhitePaper-10.txt
-rw-r--r-- 1 qinglong user 19435 Mar 30 16:30 WhitePaper-11.txt
-rw-r--r-- 1 qinglong user 1972 Mar 30 16:33 WhitePaper-12.txt
-rw-r--r-- 1 qinglong user 3601 Mar 30 16:26 WhitePaper-2.txt
-rw-r--r-- 1 qinglong user 15989 Mar 30 16:26 WhitePaper-3.txt
-rw-r--r-- 1 qinglong user 23206 Mar 30 16:27 WhitePaper-4.txt
-rw-r--r-- 1 qinglong user 27176 Mar 30 16:28 WhitePaper-5.txt
-rw-r--r-- 1 qinglong user 8387 Mar 30 16:28 WhitePaper-6.txt
-rw-r--r-- 1 qinglong user 7123 Mar 30 16:29 WhitePaper-7.txt
-rw-r--r-- 1 qinglong user 7838 Mar 30 16:29 WhitePaper-8.txt
-rw-r--r-- 1 qinglong user 6174 Mar 30 16:29 WhitePaper-9.txt
-rw-r--r-- 1 qinglong user 14171 Jan 26 15:47 java.txt
-rw-r--r-- 1 qinglong user 26802 Mar 30 16:36 security-FAQ.txt
qinglong:~/Java
qinglong:~/Java traceroute nonlin-gw
traceroute to nonlin-gw.rphys.mipt.ru (193.125.143.140), 30 hops max, 40 byte packets
 1 gw.ihep.su (194.190.161.254) 1,779 ms 4,096 ms 1,884 ms
 2 MSU-Tower-2.Moscow,RU,Radio-MSU.net (194.67.80.69) 5,517 ms 10,309 ms 5,502 ms
 3 mipt-cs-1.rmwx.net (194.85.112.133) 6,01 ms 5,835 ms 5,637 ms
 4 mipt-cs-2.mipt.ru (193.125.143.253) 10,937 ms 7,021 ms 6,818 ms
 5 nonlin-gw.rphys.mipt.ru (193.125.143.140) 10,504 ms 6,78 ms 6,066 ms
qinglong:~/Java

```

ичь Евский Шлоз

nder construction,

dxunk14

:59 from desert.ihep.su

#1: Sat May 4 00:00:13 EDT 1991

win31

dir

ment Corporation

lation procedure:

file

file

og file

4:58 PGTools-fixed.readme

4:58 PGTools-fixed.tar.gz

4:58 PGSendmail-v1.4.tar.gz

4:59 PGSendmail-v1.4.tar.gz.s

5:00 X11pgp.zip

0:07 X312html.tgz

3:35 auto-pgp-1.04-0.tar.gz

6:16 binutils-2.6.0.12.bin.tar

5:03 editpgp.gz

2:50 procinfo-0.7.tar.gz

4:36 readelf/

3:48 ru-elm+mime.tgz

7:55 xpm/

гор. На нем есть
 ен, однако несколько иным
 нутые контуры, имеющие
 и меридианов тора,
 контуров на торе
 ра есть Z^2 , а не

ащения на \mathbb{R}^3 , где
 ь наглядно показывает, как
 нуть в точку, т.е. перевести

Save As... Clone

Forest

512 May 29 17:10 L/
 512 Feb 3 20:01 X/
 512 Feb 17 16:31 bin/
 5538 Feb 5 02:17 default.DECTerm
 512 Mar 22 13:39 mail/
 512 Feb 20 08:31 samarin/
 DISPLAY=Bolizm.ihep.su:0.0

forest 11:43

forest 11:43

forest 11:43

console

Файловая система

Сокет

Более детально взаимодействие процессов посредством сокетов мы рассмотрим при анализе сетевого взаимодействия.

Сейчас же стоит отметить, что сокеты:

- создаются с помощью **системного вызова** `socket ()`,
- а закрываются только при закрытии соединения с обеих сторон.

Файловая система

Именованные каналы

К объектам файловой системы UNIX относят **именованные каналы**.

Так же как и локальные сокеты, именованные каналы позволяют взаимодействовать процессам, выполняющимся на одной машине.

Именованные каналы создаются командой `mknod`, а удаляются командой `rm`.

Файловая система

Символические и жесткие ссылки

К отдельному типу объектов файловой системы можно отнести ***символические*** и ***жесткие ссылки***.

Это такие объекты файловой системы, которые служат для обеспечения **альтернативных** способов обращения к файлам.

Файловая система

Большинство операционных систем UNIX **поддерживает несколько типов** файловых систем.

Кроме базовой версии, берущей начало от FreeBSD 4.3, поддерживаются и другие файловые системы, например, обладающие повышенной надежностью или упрощенными средствами восстановления после сбоев (HP-UX), а также системы, поддерживающие другую семантику (Solaris).

Большинство современных файловых систем имеет определенные **отличия от базовой версии**, поэтому в данный момент говорим об обобщенном типе файловой системы.

Файловая система

Иерархия файловой системы

Файловые системы UNIX располагаются, как правило, на жестких дисках, каждый из которых состоит из одной или нескольких логически связанных групп цилиндров, называемых **разделами** (partitions).

Физическое **расположение и размер** раздела устанавливаются **при форматировании** диска.

В UNIX-системах разделы являются **независимыми**, доступ к ним осуществляется как к различным носителям данных, при этом один раздел содержит, как правило, **только одну** физическую файловую систему.

Файловая система

Иерархия файловой системы

Файловую систему можно рассматривать, с одной стороны, как логическую структуру **в виде дерева каталогов и файлов** с четко установленной иерархией.

Именно в таком виде и представляется файловая система UNIX пользователю.

С другой стороны, файловая система — это **совокупность** расположенных на физическом носителе упорядоченных и неупорядоченных **двоичных данных**.

Файловая система

Иерархия файловой системы

Пользователь обычно **не имеет** доступа непосредственно к блокам данных на носителях, хотя и может управлять ими при помощи программ, в которых используются соответствующие системные вызовы.

Управление физической файловой системой — **прерогатива** функций **ядра**, поэтому **вмешательство** в этот процесс **нежелательно**, даже если вы хорошо представляете себе, что делаете.

Файловая система

Иерархия файловой системы

В операционных системах UNIX используются различные типы файловых систем, каждая из которых имеет свои особенности.

Но операционная система и программы пользователей имеют дело не с **физическими блоками** на жестком диске, а с **логическими структурами** данных, которые образуют логическую файловую систему.

В дальнейшем, если особо не оговорено, будет использоваться термин "**файловая система**" — при этом будет подразумеваться именно логическая файловая система.

Файловая система

Иерархия файловой системы

Основой любой файловой системы является корневой каталог (обозначается как /).

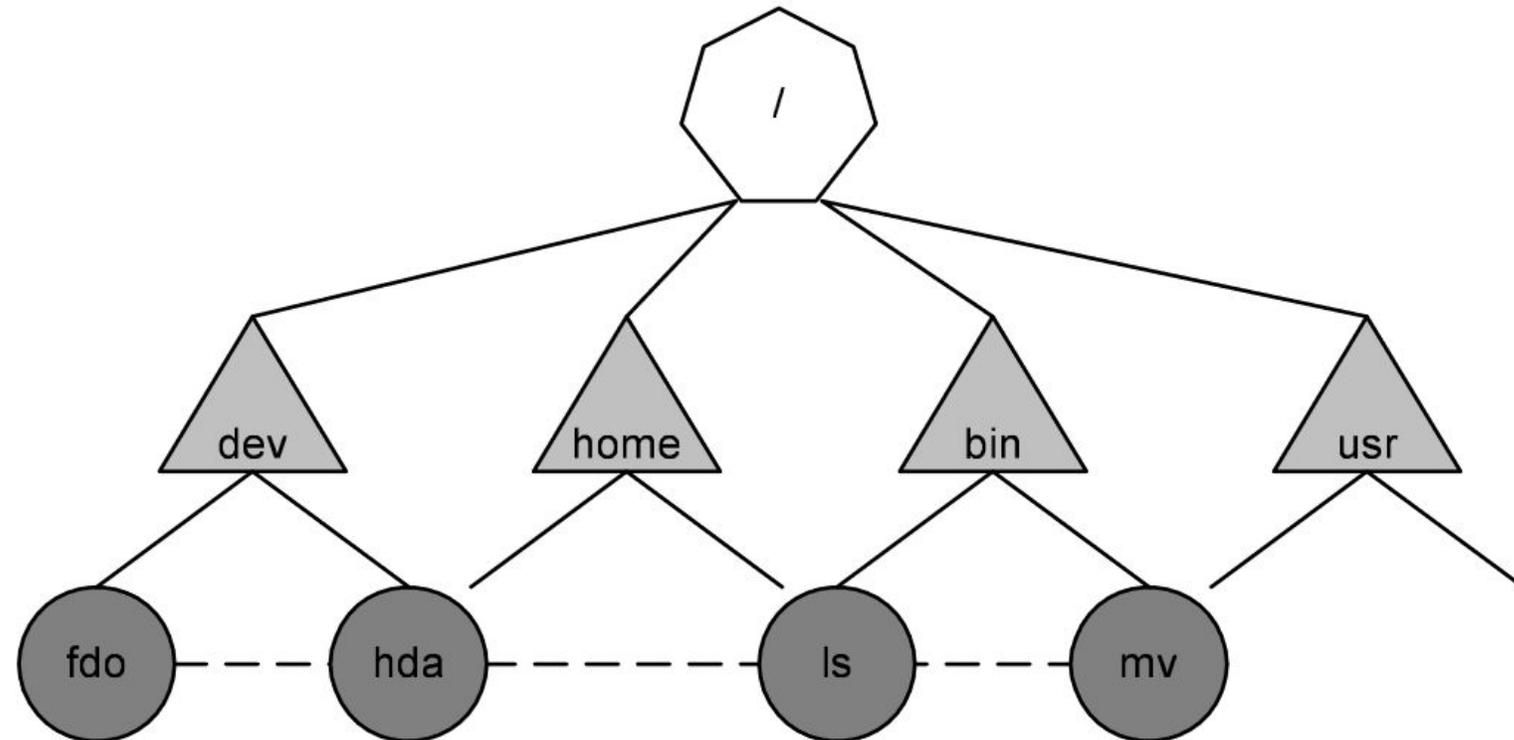
Корневой каталог, как и другие каталоги, является **обычным файлом**, содержащим информацию о других файлах, позволяя четко структурировать объекты файловой системы.

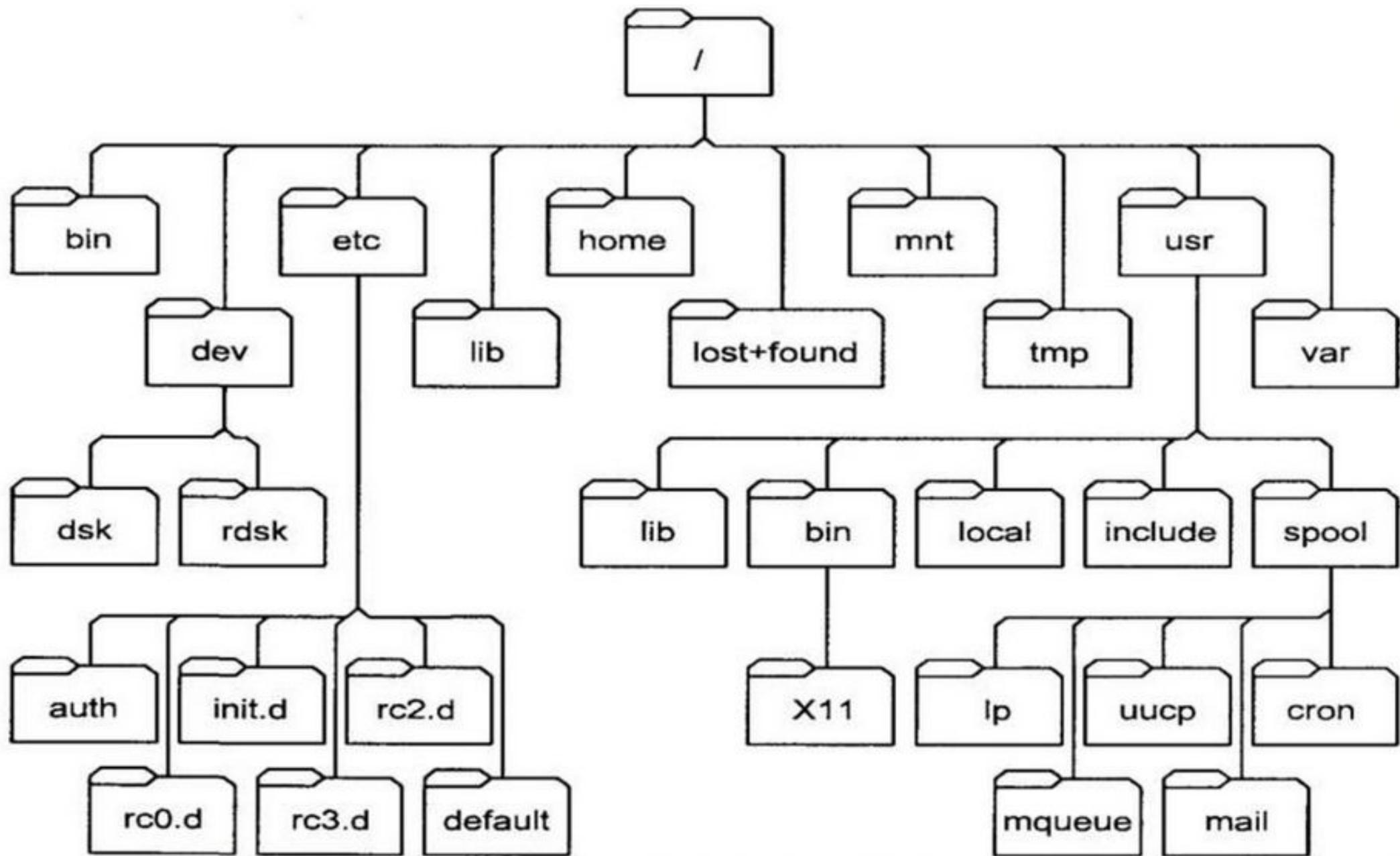
Все остальные каталоги и файлы располагаются в рамках структуры, порожденной корневым каталогом (в нем и в его подкаталогах), независимо от их физического местонахождения.

Файловая система

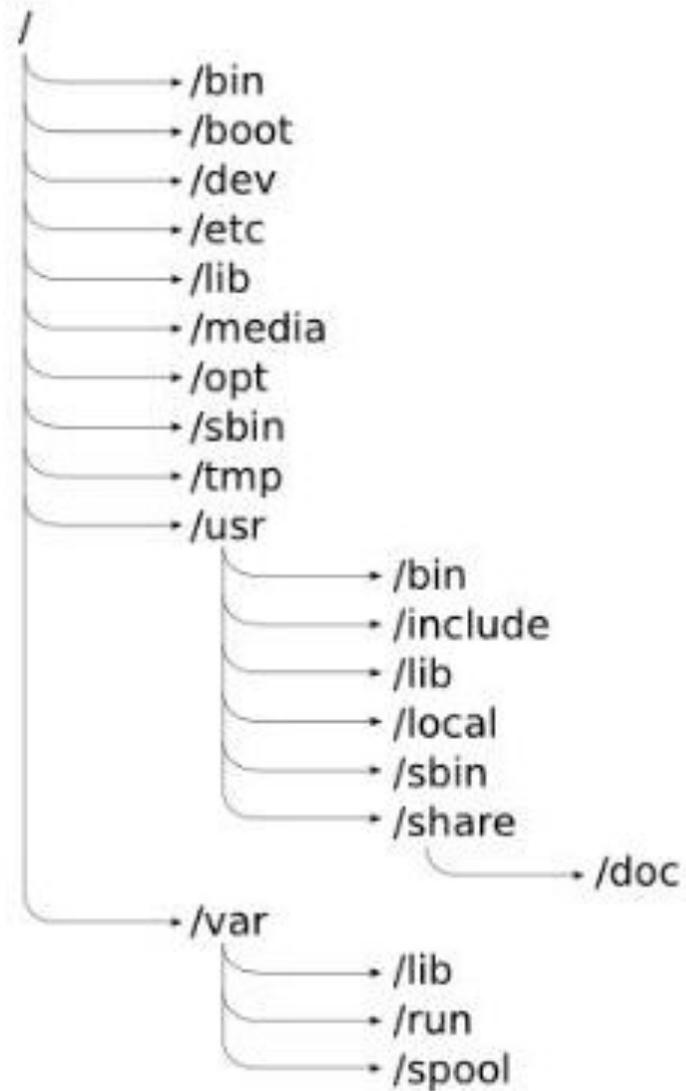
Иерархия файловой системы

Структуру файловой системы UNIX можно представить себе так, как показано на рисунке.





- **Стандартные каталоги в файловой системе UNIX**



Файловая система

Иерархия файловой системы

Каталоги и файлы располагаются в рамках структуры, порожденной корневым каталогом, независимо от их физического местонахождения, которое можно проиллюстрировать следующим рисунком.



Файловая система

Иерархия файловой системы

Совокупность имен каталогов, через которые проходит путь к заданному файлу, образует, вместе с именем этого файла включительно, **путевое имя**.

Путевые имена могут быть **абсолютными** (например, `/tmp/myfile`) или **относительными** (например, `local/filesystem`), при этом относительное имя указывается по отношению к текущему каталогу.

Максимальный размер имени каталога не должен превышать 255 символов, а отдельное путевое имя не должно быть более 1023 символов.

Файловая система

Иерархия файловой системы

Для корневого каталога обязательно создается **отдельная физическая файловая система**, а сам он является точкой ее монтирования, о чем свидетельствует наличие подкаталога `lost+found`.

Корневой каталог должен быть всегда доступен и монтируется автоматически при запуске системы, поэтому с формальной точки зрения все остальные физические файловые системы для функционирования UNIX не нужны.

Файловая система

Иерархия файловой системы

Большинство операционных систем придерживается более-менее стандартной структуры каталогов файловой системы, устанавливая для них predetermined назначения, хотя это и не обязательно:

- `/bin` — содержит программы пользователей — в большинстве современных систем обычно является символической ссылкой на каталог `/usr/bin`;
- `/dev` — каталог для специальных файлов устройств — обычно содержит подкаталоги для различных классов и типов устройств, например, `dsk`, `rdsk`, `rmt`, `inet` (в UNIX System V);

Файловая система

Иерархия файловой системы

- `/etc` — каталог для конфигурационных файлов — может включать подкаталоги для различных компонентов и сервисов;
- `/home` — каталог, в котором размещаются начальные каталоги пользователей — во многих случаях к нему монтируется (присоединяется) отдельная физическая файловая система;
- `/lib` — каталог для библиотек — обычно является символической ссылкой на `/usr/lib`;

Файловая система

Иерархия файловой системы

- `/lost+found` — этот подкаталог содержится в каждом каталоге, являющемся точкой монтирования физической файловой системы;
- `/mnt` — точка монтирования для файловых систем на съемных носителях или дополнительных дисках (например, CD-ROM или флоппи-диски);
- `/opt` — каталог для дополнительного программного обеспечения (может быть пустым или отсутствовать);

Файловая система

Иерархия файловой системы

- `/proc` — каталог псевдо-файловой системы, предоставляющей в виде каталогов и файлов информацию о ядре, памяти и процессах, работающих в системе;
- `/sbin` — каталог для системных программ, необходимых для системного администрирования UNIX;
- `/tmp` — каталог для временных файлов;

Файловая система

Иерархия файловой системы

- `/usr` — в этом каталоге находятся программы, библиотеки заголовочные файлы, справочные руководства (`/usr/share/man`), исходные тексты ядра и утилит системы (в Linux), файлы очереди печати (`/usr/spool` в BSD-системах) и т. д. Часто каталог является точкой монтирования отдельной физической файловой системы.
- `/var` — в UNIX System V и Linux этот каталог является аналогом каталога, используемого для хранения файлов различных системных сервисов, например, файлов журналов системы.

Файловая система

Иерархия файловой системы

В каталог `/usr` могут находиться следующие подкаталоги:

- `/usr/bin` — основные программы и утилиты операционной системы;
- `/usr/include` — заголовочные файлы библиотек функций.

Здесь же могут находиться подкаталоги;

Файловая система

Иерархия файловой системы

- `/usr/lib` — статически и динамически компоуемые библиотеки; может содержать подкаталоги;
- `/usr/local` — каталог для дополнительного свободно распространяемого программного обеспечения.

Данный подкаталог содержит иерархию подкаталогов, аналогичную корневому (`bin`, `etc`, `include`, `lib` и т.д.);

Файловая система

Иерархия файловой системы

Следует сказать, что использование других каталогов верхнего уровня и подкаталогов зависит:

- от версии операционной системы UNIX,
- от установленного системного и прикладного программного обеспечения,
- от изменений, внесенных в базовую конфигурацию пользователем.

Файловая система

Иерархия файловой системы

В операционной системе UNIX путь к файлу принято называть *жесткой ссылкой*.

Причем для большинства файлов имеется всего лишь одна жесткая ссылка, хотя при помощи команды `ln` пользователь может создать дополнительные жесткие ссылки для каждого из файлов.

Файловая система

Иерархия файловой системы

Например, для файла с путевым именем
`/home/user1/text` можно создать жесткую ссылку
`/home/user1/text.new`, выполнив команду

```
ln /home/user1/text /home/user1/text.new
```

После этого при любых операциях с данным файлом к нему можно обращаться по этой ссылке.

Файловая система

Иерархия файловой системы

Другой тип ссылки — **символическая**, или, как ее называют по-другому, "**мягкая**" ссылка — **позволяет** вместо путевого имени файла **указывать псевдоним**.

При обнаружении символической ссылки, например, во время поиска файла, ядро извлекает из нее **путевое имя**.

Жесткая ссылка отличается от символической тем, что она указывает непосредственно на индексный дескриптор файла, т. е. на **физическое расположение** первого блока файла, в то время как символическая ссылка указывает на файл по **его имени**.

Файловая система

Иерархия файловой системы

Файл, адресуемый символической ссылкой, и сама ссылка являются физически **разными объектами** файловой системы.

Символическая ссылка создается командой `ln -s`, а удаляется командой `rm`. Такая ссылка может указывать на файл, находящийся в другой файловой системе, или даже на несуществующий файл.

В рассмотренном ранее примере можно жесткую ссылку заменить символической:

```
ln -s /home/user1/text /home/user1/text.new
```

Список литературы:

1. Юрий Магда. UNIX для студентов, Санкт-Петербург «БХВ-Петербург», 2007.
2. Unix и Linux: руководство системного администратора, 4-е издание, 2012, Э. Немет, Г. Снайдер, Т. Хейн, Б. Уэйли
3. Организация UNIX систем и ОС Solaris 9, Торчинский Ф.И., Ильин Е.С., 2-е издание, исправленное, 2016.

Ссылки:

1. <http://bolizm.ihep.su/images/X/X-windows.example.gif>
2. <https://cf.ppt-online.org/files/slide/u/u50SQ4JW896iL2eDtodqKBcmalAnZHTypjRCvU/slide-4.jpg>
3. https://myslide.ru/documents_3/7fcd04f3851dd2390fc0dd725815997d/img3.jpg
4. <https://prezentacii.org/uploads/files/18/10/85160/data/pres/screen13.jpg>

Спасибо за внимание!

Преподаватель: Солодухин Андрей Геннадьевич

Электронная почта: asoloduhin@kait20.ru