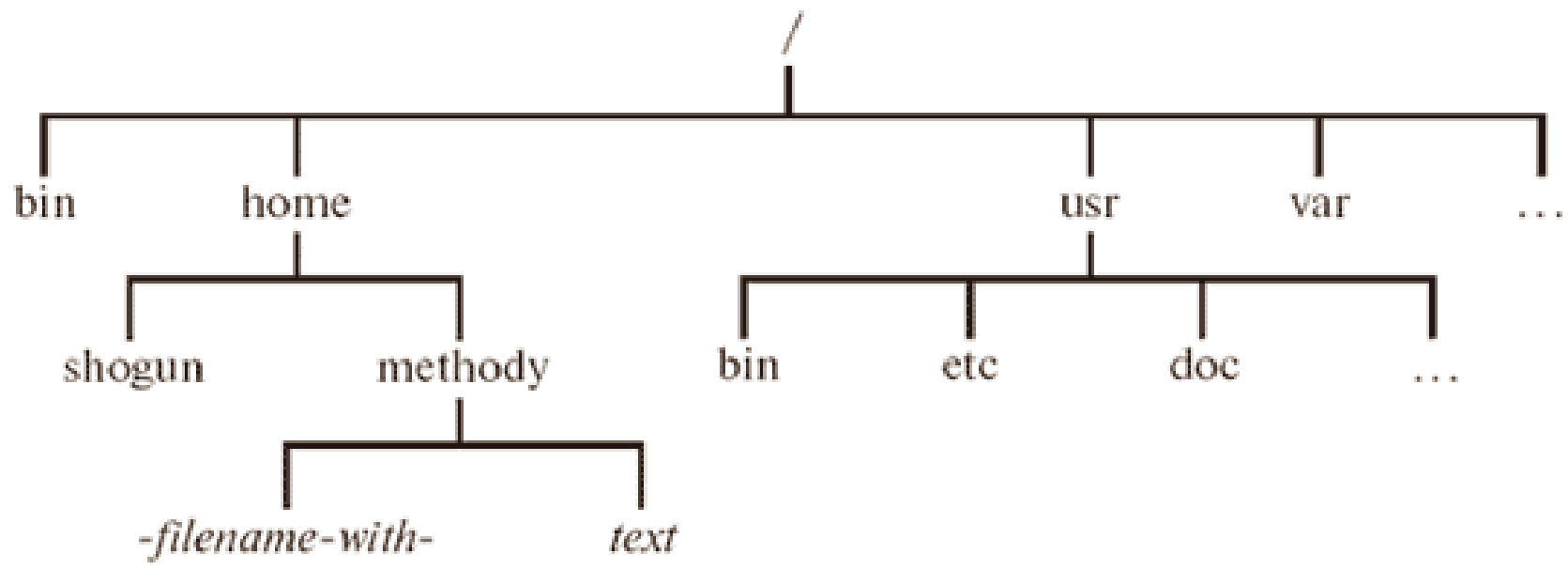


# Файловая система LINUX

Операционные системы хранят данные на диске при помощи файловых систем. Классическая файловая система представляет данные в виде вложенных друг в друга каталогов (папок), в которых содержатся файлы. Один из каталогов является «вершиной» файловой системы (корнем), в нём содержатся все остальные каталоги и файлы. Если жёсткий диск разбит на разделы, то на каждом разделе организуется отдельная файловая система с собственным корнем и структурой каталогов.



В Linux корневой каталог обозначается /. Имена каталогов в пути также разделяются символом / («слэш»). Например, запись /home обозначает каталог “home” в корневом каталоге (“/”), а/home/user — каталог “user” в каталоге “home”. Перечисленные таким образом каталоги, завершающиеся именем файла, составляют полный путь к файлу (абсолютный). Относительный путь начинается от текущего каталога.

Основные каталоги файловой системы Linux перечислены в следующей таблице.

Каталог	Описание
/	Корневой каталог, содержащий всю файловую иерархию.
/bin	Основные утилиты, необходимые как в однопользовательском режиме, так и при обычной работе всем пользователям (например: cat,ls, cp).
/boot	Загрузочные файлы (в том числе файлы загрузчика, ядро). Часто выносятся на отдельный раздел.
/dev	Основные файлы устройств (например, /dev/null).
/etc	Общесистемные конфигурационные файлы.
/etc/X	Файлы конфигурации X Window System
/home	Содержит домашние каталоги пользователей, которые в свою очередь содержат персональные настройки и данные пользователя.

/lib	Основные библиотеки, необходимые для работы программ из /bin и /sbin.
/media	Точки монтирования для сменных носителей, таких как CD-ROM, DVD-ROM
/mnt	Содержит временно монтируемые файловые системы.
/proc	Виртуальная файловая система, представляющая состояние ядра операционной системы и запущенных процессов в виде файлов.
/root	Домашний каталог пользователя root.
/sbin	Основные системные программы для администрирования и настройки системы
/tmp	Временные файлы
/usr	Содержит большинство пользовательских приложений и утилит, используемых в многопользовательском режиме.

<code>/usr/bin</code>	Дополнительные программы для всех пользователей
<code>/usr/lib</code>	Библиотеки для программ, находящихся в <code>/usr/bin</code> и <code>/usr/sbin</code> .
<code>/var</code>	Изменяемые файлы, такие как файлы регистрации, временные почтовые файлы, файлы спулеров.
<code>/var/log</code>	Различные файлы регистрации.
<code>/var/mail</code>	Почтовые ящики пользователей.
<code>/var/run</code>	Информация о запущенных программах
<code>/var/spool</code>	Задачи, ожидающие обработки (например, очереди печати, непрочитанные или неотправленные письма).
<code>/var/tmp</code>	Временные файлы, которые должны быть сохранены между перезагрузками.

## Имена файлов

ОС Linux для работы с файлами не использует их имя. Вся информация, необходимая операционной системе располагается в массиве дескрипторов, в котором метаданные файла однозначно связаны с порядковым номером соответствующего дескриптора.

Имя файла хранится вместе с номером его дескриптора в файлах специального типа — каталогах.

Имена файлов ОС Linux состоят из комбинации символов ASCII. Длина имени может достигать 255 символов.

В именах не разрешается использовать символы, имеющие специальное назначение:

; | < > ` " ' \$ ! % & \* ? \ ( ) [ ]

Имена файлов, которые начинаются с символа (.) относятся к скрытым (hidden) файлам, которые не выводятся по умолчанию командой ls. Для того, чтобы вывести имена скрытых файлов, необходимо использовать опцию -a в команде ls.

В именах файлов различаются заглавные и прописные буквы, поэтому имена Test и test относятся к различным файлам.



Файловая система ОС Linux поддерживает несколько типов файлов: обычные файлы, каталоги, символические связи, специальные файлы устройств.

Если выполнить команду `ls -l`, которая выводит информацию о содержимом каталога, то можно увидеть: первый символ первого слова каждой строки указывает на тип соответствующего файла. Значение символов следующее: - обычный файл, d – каталог, l – символическая связь, c, b - файл устройства.

Обычные файлы содержат неструктурированную последовательность байтов. Приложение, работающее с таким файлом, определяет его структуру и содержание. Обычно такие файлы можно отнести к одной из следующих категорий:

- Текстовые, содержащие набор символов. Например, письма, отчеты, командные файлы, используемые интерпретатором shell.
- Файлы, содержащие наборы числовых или текстовых данных какого-либо приложения: Например, электронные таблицы, базы данных или документы текстовых процессоров.
- Исполняемые программы в двоичном виде, содержащие машинные команды и данные. Например, программы, связанные с выполнением команд ОС Linux или программы приложений.

Специальный файл устройства обеспечивает доступ к устройству. В Linux различают символьные (character) и блочные (block) файлы устройств.

Символьные файлы связаны с драйверами устройств использующими небуферизированный обмен данными с устройствами, к которым относятся терминал, принтер, мышь.

Блочные файлы связаны с драйверами устройств, которые позволяют производить обмен данными в виде пакетов фиксированной длины – блоков.

Типичные примеры блочных устройств: жёсткий диск, CD-ROM.

Доступ к некоторым устройствам может осуществляться как через символьные, так и через блочные специальные файлы. Имена файлов устройств содержатся в каталоге /dev.

Файл устройства `/dev/fd0` флоппи-дисковод:

```
$ls-l /dev/fd0
```

```
brw-rw---- 1 root floppy 2, 0 Jan 1 11:17 /dev/fd0
```

Первый символ в расширенном выводе `ls` (`ls -l`) для блочных устройств — буква `b` (block). Вместо размера файла в пятом поле таблицы показывается два числа — мажорный и минорный номер устройства (major and minor numbers). Первый номер отражает тип устройства, а второй — номер устройства в системе или режим его работы.

Любые физические устройства файловой системы Linux лежат в каталоге /dev файлового дерева. Жёсткие диски, дисководы CD ROM, приводы под флешки и другие носители, все находятся там. Имена, присваиваемые системой этим устройствам выглядят следующим образом.

- /dev/hda — первый на первом канале IDE (Primary master);
- /dev/hdb — второй на первом канале IDE (Primary slave);
- /dev/hdc — первый на втором канале IDE (Secondary master);
- /dev/hdd — второй на втором канале IDE (Secondary slave);

Жёсткие диски SCSI обозначаются `/dev/sdX`, где `X` — `a,b,c,d,e,...`, в зависимости от порядкового номера диска на шине SCSI. Раздел диска обозначается числом после его имени. Например, `/dev/hdb4` — четвёртый раздел второго диска на первом канале IDE.

Символические связи – это специальные файлы, которые содержат в качестве данных символическую ссылку на другой файл. Эти данные указывают путь (path) к файлу внутри файловой системы. Ядро системы автоматически определяет по содержимому файла символической связи место расположения файла в файловом дереве.

В файловой системе Linux содержимое файла связывается с его именем при помощи **жестких ссылок**. Создание файла с помощью любой программы означает, что будет создана жесткая ссылка - имя файла, и открыта новая область данных на диске.

Пользователь Linux может добавить файлу еще одно имя (создать еще одну жесткую ссылку на файл) при помощи утилиты `ln`. Первый параметр - это имя файла, на который нужно создать ссылку, второй - имя новой ссылки. По умолчанию ссылка будет создана в текущем каталоге.

Поскольку благодаря жестким ссылкам у файла может быть несколько имен, вся существенная информация о файле в файловой системе привязана не к имени. В файловых системах Linux вся информация, необходимая для работы с файлом, хранится в индексном дескрипторе.



Индексный дескриптор- это описание файла, в котором содержится:

- тип файла (обычный файл, каталог и т. д.);
- права доступа к файлу;
- информация о том, кому принадлежит файл;
- отметки о времени создания, модификации, последнего доступа к файлу;
- размер файла;
- указатели на физические блоки на диске, принадлежащие этому файлу - в этих блоках хранится "содержимое" файла.

Все индексные дескрипторы пронумерованы, поэтому номер индексного дескриптора - это уникальный идентификатор файла в файловой системе.

Узнать номер индексного дескриптора любого файла можно при помощи утилиты `ls` с ключом `-i`.

**Символьная ссылка** (symbolic link, файл-ссылка) - это файл особого типа ("l"), в котором содержится путь к другому файлу. Символьную ссылку можно создать при помощи команды `ln` с ключом `-s`.

Каталог создается командой **mkdir** и удаляется (при условии, что он пуст) командой **rmdir**. Непустые каталоги можно удалить командой **rm -r**.

Специальные ссылки “.” и “..” обозначают сам каталог и его родительский каталог соответственно.

## Использование джокеров

При повседневном использовании Linux часто случается, когда вам нужно выполнить единичную операцию (например `rm`) на множестве объектов файловой системы за раз.

Так, если у вас есть файлы `file1` до `file8` в текущей рабочей директории, то можете их удалить набрав:

```
$ rm file[1-8]
```

Или если хотите удалить все файлы с именами начинающимися с `file`, можете набрать:

```
$ rm file*
```

В качестве джокеров используются специальные символы:

\* — совпадает с нулевым или большим количеством символов.

Примеры:

**/etc/g\*** совпадает со всеми файлами в /etc, начинающимися с g и самим файлом g;

**/tmp/my\*1** совпадает со всеми файлами в /tmp, которые начинаются с my и заканчиваются 1, включая файл my1.

? — равен любому одному символу.

Примеры:

**myfile?** совпадает с любым файлом, чье имя составляет myfile и следующим за этим какой-либо один символ;

**/tmp/notes?.txt** совпадет, например, с /tmp/notes.txt и /tmp/notes\_txt, если они существуют.

## Синтаксис джокера: []

Примеры:

**myfile[12]** совпадет с **myfile1** и **myfile2**. Джокер сработает если хотя бы один из этих файлов существует в текущей директории;

**[Cc]hange[L]og** совпадет с **Changelog**, **ChangeLog**, **changeLog** и **changelog**;

**ls /etc/[09]\*** покажет все файлы в **/etc**, начинающиеся с десятичной цифры;

**ls /tmp/[A-Za-z]\*** отобразит все файлы в **/tmp**, которые начинаются с большой или маленькой латинской буквы.

Конструкция `[!]` эквивалентна конструкции `[]`, за исключением того, что вместо совпадения с символами внутри скобок, она удовлетворяет любому символу, который НЕ перечислен между `[!` и `]`.

Пример:

**`rm myfile[!9]`** удалит все файлы с названием `myfile` плюс один символ, кроме `myfile9`.



Установка прав доступа к файлам в ОС Linux позволяет эффективно защищать файлы от несанкционированного доступа к нему с целью изменения его содержимого, считывания из него информации или его исполнения. К файлами могут обращаться различные категории пользователей: владельцы файла, члены группы, к которым принадлежит владелец и члены других групп зарегистрированных пользователей. Рассмотрим результат вывода команды `ls -l`, с помощью которой имеем возможность вывода всех атрибутов файла.

**\$ ls -l dat**

```
-rw-r--r-- 1 dk30101 dk301 24 22 Sep 11:30 dat
```

Первое слово (-rw-r--r--) в первом символе информацию о типе файла, затем девятью символами определяются права доступа к файлу. Второе слово (1) указывает число ссылок к файлу.

Третье слово (dk30101) определяет имя владельца файла.

Четвертое слово (dk301) определяет имя группы, к которой относится владелец файла.

Пятое слово (24) указывает на размер файла в байтах.

Далее (22 Sep 11:30) показывает дату и время создания файла (11:30) в часах и минутах.

В конце строки указывается имя файла (dat).

Изменить имя владельца может только администратор (**root**) или текущий владелец файла. Это он может сделать с помощью команды **chown**:

**chown** имя\_владельца имя\_файла [имя\_файла ...]

Изменить имя группы владельца может администратор (**root**) или владелец файла. Это он может сделать с помощью команды **chgrp**:

**chgrp** имя\_группы имя\_файла [имя\_файла ...]

Права доступа, которые устанавливает владелец файла (или администратор) определяют — кто имеет доступ к файлу и что он может сделать с этим файлом.

Права доступа определяются для трех категорий пользователей:

- владельца (user),
- члена группы владельца (group),
- прочих пользователей (other).

Права доступа определяют три вида доступа:

- разрешение на чтение (read) файла (r),
- разрешение на запись (write) (w),
- разрешение на исполнение (execute) или поиск (search) (x).

Следует различать назначение прав доступа к файлам и каталогам.

## **Права доступа к файлу:**

- **Read (r)** дает возможность пользователю открыть файл и просмотреть его содержимое. Команды, требующие доступ к содержимому файла, требуют права доступа на чтение, например, команды `cat`, `cp`.
- **Write (w)** позволяет изменять содержимое файла. Команды, требующие доступ к изменению файла, требуют права доступа на запись, например, команды `vi`, `mail`, `cp`.
- **execute (x)** дает возможность пользователю выполнить двоичную программу или командный файл подобно любой команде ОС Linux.

## Права доступа к каталогу:

- **Read (r)** дает возможность пользователю вывести список имен файлов и каталогов (прочитать содержимое каталога).
- **Write (w)** позволяет изменять содержимое каталога, т.е. добавить или удалить имя (файла или каталога) из каталога. Обычно только владелец каталога имеет разрешение на запись в каталог.
- **eXecute (x)** дает возможность осуществить поиск по каталогу. Без этого права доступа невозможно использовать имя каталога как компоненту полного имени файла. Отсутствие права на исполнение не позволит сменить каталог при выполнении команды `cd`.

Права доступа могут модифицироваться владельцем файла или пользователем root с помощью команды:

**chmod** режим [режим ...] имя\_файла/[имя\_файла ...]