

## Типы файлов

В UNIX существует шесть типов файлов, различающихся по строению и поведению при выполнении операций над ними:

- **Обычный файл** (regular file): наиболее общий тип файлов, содержащий данные в некотором формате. Для ОС это просто последовательность байт. Интерпретация содержимого производится прикладной задачей.

Пример: текстовый файл, двоичные данные, исполняемый ф. Команды: cat имя и less имя.

- **Каталог** (directory): Это файл, содержащий имена находящихся в нем файлов, а также указатели на метаданные этих файлов, позволяющие ОС производить операции над ними. Каталоги определяют положение файла в дереве файловой системы, так как сам файл не содержит информации о своем местонахождении. Каталоги образуют дерево. Первые два байта в каждой строке каталога являются единственной связью между именем файла и его содержимым. Именно поэтому *имя файла в каталоге называют связью*. Оно связывает имя в иерархии каталогов с индексным дескриптором и, тем самым, с информацией.

- **Специальный файл устройства** (special device file): Обеспечивает доступ к физическому устройству. Различают символьные и блочные файлы устройств. Доступ к устройствам происходит путем открытия, чтения/записи в специальный файл устройства. *Символьные файлы* позволяют небуферизованный обмен данными (посимвольно), а *блочные* - обмен пакетами определенной длины - блоками. К некоторым устройствам доступ возможен как через символьные, так и через блочные файлы. Для создания - mknod.

- **FIFO или именованный канал** (named pipe). Используется для связи между процессами.

- **Связь** (ссылка)

- **Жесткая ссылка**: Связь имени файла с его данными называется *жесткой ссылкой* (hard link). Имена жестко связаны с метаданными и, соответственно, с данными файла, в то время, как файл существует независимо от того, как его называют в файловой системе. Такая система позволяет одному файлу иметь несколько имен в файловой системе.

Пример: \$ pwd /home/stud1 \$ln first /home/stud2 second # создание жесткой ссылки.

Все жесткие ссылки на файл абсолютно равноправны.

Файл существует в системе до тех пор, пока существует хотя бы одна жесткая связь, указывающая на него, то есть пока у него есть хотя бы одно имя.

- **Символическая ссылка**: Особый тип связи - символическая связь, позволяющая косвенно адресовать файл, в отличие от жесткой, обращающейся напрямую. Символическая ссылка содержит в себе имя файла, на который ссылается, а не его данные.

- **Сокет** (socket): Используются для межпроцессного взаимодействия.

## Структура файловой системы

Все Unix-системы имеют сходную систему расположения и именования файлов и каталогов. Использование общепринятых имен файлов и структуры каталогов в UNIX-подобных ОС облегчает работу и перенос. Нарушение структуры ведет к нарушениям в работе.

Корневой каталог "/" является основой FS. Все остальные файлы и каталоги располагаются в рамках структуры, порождаемой корневым каталогом.

*Абсолютное или полное имя* файла определяет точное местонахождение файла в структуре файловой системы. Начинается с "/" (в корневом каталоге) и содержит полный путь подкаталогов, которые нужно пройти, чтобы достичь файла.

*Относительное имя* определяет местонахождение файла через текущий каталог. Никогда не начинается с "/".

*Каталог-предок* - это тот, который содержит другой каталог (...). Корневой каталог не имеет предка. Каталог, находящийся в другом каталоге, называется *каталогом-потомком или подкаталогом*. К текущему каталогу можно обратиться по имени ".": ./file1.

*Домашним или начальным каталогом* называется область, которая выделяется каждому пользователю и в которой он может хранить свои файлы и программы "~": ~/file.txt.

## Основные каталоги

/bin - наиболее часто употребляемые файлы и утилиты.

/dev - содержит специальные файлы устройств, являющиеся интерфейсом доступа к периферийным устройствам. Может содержать подкаталоги, группирующие устройства по типам. Например, /dev/dsk - доступ к дискам.

/etc - системные конфигурационные файлы и утилиты. Иногда утилиты отсюда выносятся в /sbin и /usr/sbin.

/lib - библиотеки Си и других языков программирования. Часть библиотек - в /usr/lib.

/usr, /usr/bin - утилиты; /usr/include - заголовочные файлы Си; /usr/man - справочная система; /usr/local - дополнительные программы; /usr/share - файлы, разделяемые между различными программами.

/var - временные файлы сервисных подсистем (печати, почты, новостей).

/tmp - каталог временных файлов. Обычно открыт на запись для всех пользователей системы.

## Атрибуты файлов

Группой называется определенный список пользователей системы. Пользователь может быть членом нескольких групп, одна из которых является первичной, а остальные - дополнительными. В UNIX любой файл имеет двух владельцев: владельца-пользователя и владельца-группу. При этом владелец-пользователь не обязательно принадлежит владельцу-группе.

Сменить владельца-пользователя может либо текущий владелец, либо администратор (root). Сменить владельца-группу может либо владелец-пользователь для группы, к которой он сам принадлежит (POSIX), либо администратор.

## Права доступа к файлам

У каждого файла существуют атрибуты, называемые правами доступа. В UNIX существует три базовых типа доступа:

u (user), g (group), o (other), a (all).

В каждом из этих классов установлены три основных права доступа:

r (read), w (write), x (execute)

Права может изменять владелец-пользователь и(или) администратор. Для изменения прав доступа используется команда `chmod`:

## Последовательность проверки прав

-если вы администратор (root), доступ разрешен. Права не проверяются.

-если операция запрашивается владельцем, идет проверка его прав. В соответствии с ними ему разрешается выполнение операции или нет.

-если операция запрашивается пользователем, входящим в группу, владеющую файлом, идет проверка его прав. Соответственно, он либо получает разрешение, либо нет. аналогично для всех остальных пользователей.

## Дополнительные атрибуты файла (также устанавливаются с помощью `chmod`)

Для обычных файлов:

t - "*sticky bit*" (бит липучка)- сохранить образ выполняемого файла в памяти после выполнения (устаревший атрибут)

s - set UID, *SUID* - установить права у процесса, как у запущенного файла, а не как у пользователя, запустившего программу (по умолчанию)

s - set GID, *SGID* - то же для группы

1 - блокирование - в каждый момент времени с файлом может работать только одна задача

Для каталогов: t - пользователь может удалять только те файлы и каталоги, которыми владеет или имеет право на запись;