**Типы записей**

‘-‘ – файл ; ‘d’ – каталог ; ‘l’ – символическая ссылка

**Права доступа**

Файл: r – чтение w – запись x – выполнение

Каталог: r – возможность просмотра списка файлов ; w – возможность создания/удаления файлов; x – переход в каталог

| **Значение** | **Права доступа файла** | **Список файлов каталога** |
| --- | --- | --- |
| 0 | Ничего не разрешено | --- |
| 1 | Нельзя читать и писать, разрешено исполнять | --x |
| 2 | Нельзя читать и исполнять, разрешено писать | -w- |
| 3 | Нельзя читать, разрешено писать и исполнять | -wx |
| 4 | Разрешено читать, нельзя писать и исполнять | r-- |
| 5 | Разрешено читать и исполнять, нельзя писать | r-x |
| 6 | Разрешено читать и писать, нельзя исполнять | rw- |
| 7 | Разрешено все | Rwx |

chmod <опции> <права> <объект или регулярное выражение>

Например

$chmod 740 tempfile.txt

Установит следующие права: разрешено всё для создателя файла, разрешение на чтение для группы и ничего не разрешено для остальных пользователей.

Другой способ: использование маски режима доступа. Помимо прав указывается, кому выставить эти права:

* **u** - владельцу объекта;
* **g** - группе объекта;
* **o** - пользователю «все остальные»;
* **a** - все вышеперечисленное.

Для назначения прав используются три знака: минус, плюс или равно:

* **-** - убрать указанные права с объекта;
* **+** - добавить указанные права к существующим правам объекта;
* **=** - заменить права объекта на указанные.

**SUID и SGID**

установка битов SUID или SGID позволит пользователям запускать исполняемые файлы от имени владельца (SUID) или группы(SGID) запускаемого файла. Например, команду chmod по умолчанию может запускать только root. Если мы установим SUID на исполняемый файл /bin/chmod, то обычный пользователь сможет использовать эту команду без использования sudo, так, что она будет выполнятся от имени пользователя root. В некоторых случаях очень удобное решение. Кстати по такому принципу работает команда passwd, c помощью которой пользователь может изменить свой пароль.

SUID: восьмеричное значение 4000, символьное u+s

SGID: восьмеричное значение 2000, символьное g+s

Например

$chmod 4755 StupidFile

Устанавливает SUID бит и заменяет обычные права на 755

$chmod 2755 AnotherStupidFile

Устанавливает SGID бит и заменяет обычные права на 755

$chmod 6755 JustOneMoreStupidFile

Установит SUID и SGID бит и заменит обычные права на 755

**Sticky-bit в каталогах**

Установка sticky-bit’а в каталоге означает, что удалять файл из каталога может только владелец файла или суперпользователь.

Восьмеричное значение: 1000

Символьное: +t

Пример:

$chmod +t TemporaryFolder

**ls:** просмотр содержимого каталога

* -a (all) — выводит список всех файлов каталога, в том числе и скрытые файлы. .. и . в верхней части списка относятся, соответственно, к каталогу на уровень выше текущего и текущему.
* -l (long) — выводит информацию о содержимом каталога, включая права доступа, владельца, группу, размер, время создания, является ли файл символьной ссылкой на другой файл и на какой именно.
* -F (file type) — добавляет символ в конце строк вывода: / означает каталог, @ — символьную ссылку на другой файл и \* означает, что файл является исполняемым.
* -r (reverse) — выводит список содержимого каталога в обратном порядке.
* -R (recursive) — будет выведено также содержимое всех подкаталогов текущего каталога рекурсивным образом.
* -S (size) — сортирует файлы по размеру.
* -i - вывод перед каждым файлом номера inode (уникальный номер файла в файловой системе).
* -d - обработка каталогов, указанных в командной строке так, как если бы они были обычными файлами, вместо вывода списка их файлов.

Пример:

$ls –ls

Выведет список всех файлов и каталогов (кроме скрытых) в виде таблицы и отсортирует по размеру.

*Не считывает данные из потока ввода, сл-но, не является фильтром*

**pwd –** определяет текущий каталог

*Не считывает данные из потока ввода, сл-но, не является фильтром*

**cd** – смена директории

Пример:

$cd ~

*Не считывает данные из потока ввода, сл-но, не является фильтром*

**rm** – удалить файлы

      -i - просить подтверждения на каждое удаление   
      -r - рекурсивно удалить вместе с подкаталогами   
      -f - не просить подтверждения, а сразу удалять

Для удаления пустой директории можно воспользоваться командой rmdir

Пример:

$rm –f idontneedthisfileanymore

*Не считывает данные из потока ввода, сл-но, не является фильтром*

**mv** Переместить (переименовать) файлы.

Формат команды:

$mv файл1 файл2

$mv файл1 [файл2 ...] каталог

Пример: $mv /TempFile/Medved ~/Preved

*Не считывает данные из потока ввода, сл-но, не является фильтром*

**cp** Копировать файл

* -i (interactive) — интерактивный режим. Требует подтверждения, если командой будет переписан файл в месте назначения. Это помогает избежать возможных ошибок.
* -f: противоположность -i, заменяет любые существующие файлы без запроса подтверждения.
* -r (recursive) — рекурсивный режим. Вместо копирования всех указанных файлов и каталогов скопирует все дерево каталогов, подкаталоги и прочее.
* -v (verbose) — режим дополнительных сообщений. Выводить на экран сообщение о каждой выполняемой операции.

*Не считывает данные из потока ввода, сл-но, не является фильтром*

**cat** – объединить и вывести содержимое файла

-b Нумеровать непустые выводимые строки, начиная с **1**.

-e Выводить непечатаемые символы (см. опцию **-v**), и выводить знак доллара ( "$" ) в конце каждой строки.

-n Нумеровать выводимые строки, начиная с **1**.

-s Подавлять вывод нескольких пустых строк, следующих друг за другом, заменяя их на одну пустую строку.

-t Выводить непечатаемые символы (см. опцию **-v**), и выводить символы табуляции в виде "**^I**".

Запретить буферизацию вывода.

-v Выводить непечатаемые символы в читабельном виде. Управляющие символы типа *control-X* печатаются как **^X**; символ *delete* (восьмеричное значение **0177**) печатается как **^?**. Не-ASCII символы (с ненулевым старшим битом) печатаются как **M-** ( *meta* ) с последующим символом, полученным из младших **7** битов.

СТАТУС ЗАВЕРШЕНИЯ  
Утилита *cat* возвращает **0** в случае успешного завершения и **>0** в случае возникновения ошибки.

*Команда cat считывает данные из потока ввода, сл-но, является фильтром.*

Пример конвеера: $cat File | tail

Выводит последние 10 строк из файла File

**mkdir** – создание каталога(-ов)

-m, --mode=режим установить код доступа (как в chmod)

-p, --parents не выдавать ошибок если существует, создавать родительские каталоги если необходимо

-v, --verbose печатать сообщение о каждом созданном каталоге

-Z, --context=CTX установить контекст безопасности SELinux для каждого создаваемого каталога равным CTX

Пример:

$mkdir HelloWorld

*Не считывает данные из потока ввода, сл-но, не является фильтром*

**rmdir** – см. “rm”

**file** – выводит тип данных файла на основе правил, определённых в файле /usr/lib/magic

file [опции]файлы

* -f file: Считывает из указанного файла список файлов для проверки.
* -L : Определяет тип файлов, указанных по ссылке.
* -z : Определяет тип файлов, находящихся в сжатых файлах.

*Не считывает данные из потока ввода, сл-но, не является фильтром*

Пример:

$file NewFile

**find** – возвращает все файлы в текущей директории

* -name — искать по имени файла, при использовании подстановочных образцов параметр заключается в кавычки.
* -type — тип искомого: f=файл, d=каталог, l=ссылка (link), p=FIFO (First In, First Out), s=сокет.
* -user — владелец: имя пользователя или UID.
* -group — владелец: группа пользователя или GID.
* -perm — указываются права доступа.
* -size — размер: указывается в 512-байтных блоках или байтах (признак байтов — символ «c» за числом).
* -atime — время последнего обращения к файлу.
* -ctime — время последнего изменения владельца или прав доступа к файлу.
* -mtime — время последнего изменения файла.
* -newer другой\_файл — искать файлы созданные позже, чем другой\_файл.
* -delete — удалять найденные файлы.
* -ls — генерирует вывод как команда ls -dgils.
* -print — показывает на экране найденные файлы.
* -exec command {} \; — выполняет над найденным файлом указанную команду; обратите внимание на синтаксис.
* -ok — перед выполнением команды указанной в -exec, выдаёт запрос.
* -depth или -d — начинать поиск с самых глубоких уровней вложенности, а не с корня каталога.
* -maxdepth — максимальный уровень вложенности для поиска. «-maxdepth 0» ограничивает поиск текущим каталогом.
* -prune — используется, когда вы хотите исключить из поиска определённые каталоги.
* -mount или -xdev — не переходить на другие файловые системы.
* -regextype тип — указание типа используемых регулярных выражений.
* -P — не разворачивать символические ссылки (поведение по умолчанию).
* -L — разворачивать символические ссылки.

*Не считывает данные из потока ввода, сл-но, не является фильтром*

**chmod** – смена прав доступа к файлу

*Не считывает данные из потока ввода, сл-но, не является фильтром*

**ln** – устанавливает связь между файлом и именем файла

**ln** файл1 файл2

# создаётся «жёсткая» ссылка (hard link)

**ln** -s файл1 файл2

# создаётся «символьная» ссылка (symbolic link)

*Не считывает данные из потока ввода, сл-но, не является фильтром*

**wc** – подсчет строк, слов и символов

-l – число строк

-w – число слов

-c – число символов

*Является фильтром, т.к. считывает данные из потока ввода*

wc AnotherFile | more – вывод информации о файле через more.

**tee** отображает результаты на стандартном выводе и в то же самое время перенаправляет вывод в файл.

-а – не перезаписывать файл, а дописывать в конец

*Является фильтром, т.к. считывает данные из потока ввода*

Пример конвеера:

$tee | wc –w

Вывод количества слов во введенном тексте

**type** – отобразить информацию о команде

*Не считывает данные из потока ввода, сл-но, не является фильтром*

**tail** – выводит последние 10 строк

-l – строка

-b – блок

-c – символ

-f – для проверки хода генерации файла

*Является фильтром*

$man echo | tail -37

Вывод последних 37 строк описания команды echo

**head** – выводит первые строки

**head** [*-n*|*-c N*] [*-q*|*-v*] [*file1 file2 ...*]

|  |  |  |
| --- | --- | --- |
| где | *N* | - число выводимых строк (по умолчанию n=10), |
| *c N* | - вывод N первых символов (байт) файла (по умолчанию N=10), |
| *Q* | - никогда не выводить заголовки файлов, |
| *V* | - всегда выводить заголовки файлов, |
| file1, file2 ... | - список файлов, из которых выводятся строки. |

*Является фильтром*

Пример фильтра:

$man head | head -29

Вывод первых 29 строк описания команды man

**more** - утилита, которая отображает содержимое файла постранично. Команда more используется для вывода информации на экран "без прокрутки", то есть информация отображается порциями, помещающимися на одном экране.

*Является фильтром*

Пример команды:

$man man | more

Вывод описания команды man при помощи утилиты more

**echo** – отображение строки текста

|  |  |
| --- | --- |
| **-n** | не выводить в конце символ новой строки (По умолчанию echo добавляет к выводу символ перевода строки) |
| **-e** | включить интерпретацию управляющих символов |
| **-E** | отключить интерпретацию управляющих символов |
| **--help** | выдает краткую справку и заканчивает работу (только с одним этим параметром) |
| **--version** | выдает информацию о версии и заканчивает работу (только с одним этим параметром) |

*Не является фильтром*

**Перенаправление ввода-вывода** - это возможность командной оболочки перенаправлять стандартные потоки в определенное пользователем место, например в файл.

Перенаправление осуществляется при помощи специального символа **>** между командами.

$ команда1 > файл1

Выполняет команду1, помещая стандартный вывод в файл1

$ команда1 < файл1

Выполняет команду1, используя в качестве источника ввода файл 1 (вместо клавиатуры).

$ команда 1< файл 1 > файл 2

Выполняет команду1, вводя из файла1 и выводя в файл2

**Перенаправление в/из стандартных дескрипторов**

|  |  |  |
| --- | --- | --- |
| Дескриптор | Название | Описание |
| 0 | stdin | Стандартный ввод |
| 1 | stdout | Стандартный вывод |
| 2 | stderr | Стандартный вывод ошиб. |

Например

$ команда1 2> файл1

Выполнит команду 1 и перенаправит поток ошибок в файл1

## Жесткая и символьная ссылки

**Жёсткой ссылкой** в UFS-совместимых файловых системах называется структурная составляющая файла.

Файл в UFS представляет собой структуру блоков данных на диске, имеющую уникальный индексный дескриптор (или i-node) и набор атрибутов (метаинформацию). Жёсткая ссылка связывает индексный дескриптор файла с каталогом и дает ему имя.

**Символьная ссылка** — специальный файл в файловой системе, для которого не формируются никакие данные, кроме одной текстовой строки с указателем. Эта строка трактуется как путь к файлу, который должен быть открыт при попытке обратиться к данной ссылке (файлу).