

## Типы файлов

1. — обычный файл, хранящий данные.
2. d — directory — файл директории; содержит список файлов "имя - i-node". i-node — индексный узел — структура, в которой хранится служебная информация о файле: расположение данных файла на диске, длина файла, владелец, права доступа и т.п.
3. c — character special file — файл символьного устройства; интерфейс, обеспечивающий последовательный небуферизированный ввод/вывод для устройств с переменным размером блока (терминал, принтер, СОМ-порт).
4. b — файл блочного устройства; интерфейс, обеспечивающий ввод/вывод в произвольном порядке блоками фиксированного размера для устройств (жёсткий диск).
5. s — сокет домена Unix.
6. p — именованный канал.
7. l — символическая ссылка — файл, хранящий путь до файла, на который он ссылается; интерпретируется ОС в качестве пути.

## Права доступа

Права доступа к файлу разделяются на три группы: права владельца, права группы, права для остальных пользователей.

### Для обычных файлов:

- r — 4 — право на чтения данного файла;
- w — 2 — право на запись в данный файл;
- x — 1 — право на исполнение данного файла.

### Для директории:

- r — 4 — право на получение списка содержимого директории;
- w — 2 — право на изменение содержимого директории: удаление / создание файлов;
- x — 1 — право на "исполнение" директории: возможность сделать его текущим, получить возможность работы с содержимым и обработки метаинформации.

## Операторы

1. cmd1 && cmd2 — выполнение cmd2 лишь при успешном завершении cmd1 (если код возврата cmd1 == 0);
2. cmd1 || cmd2 — выполнение cmd2 лишь при неудачном завершении cmd1 (если код возврата cmd1 != 0);
3. cmd1 & — выполнение cmd1 в фоне;
4. cmd1; cmd2; — последовательное исполнение команд;
5. cmd1; cmd2 — группировка команд и исполнение в текущем шелле;
6. ( cmd1; cmd2; ) — группировка команд и исполнение в подшелле.

## Сочетания по командной строке:

^C — сигнал процессу, с которым пользователь сейчас работает; принудительное завершение процесса.

^D — EOF - Символ окончания ввода (в командной строке приведёт к закрытию текущего shell-a).

### Навигация:

- ^P — предыдущая набранная команда.
- ^N — следующая набранная команда (если команд нет, ни к чему не приведёт).
- ^F — курсор вправо на один символ.
- ^B — курсор влево на один символ.
- ^A — курсор в начало строки.
- ^E — курсор в конец строки.
- ^H — удалить символ после курсора.
- ^U — удалить всю строку после курсора.
- ^W — удалить всё слово после курсора.
- Esc-B — переместить курсор на одно слово назад.
- Esc-F — переместить курсор на одно слово вперёд.

# Команды

## 1. ls

ls – “list”

ls [option] [files] Выводит информацию о files.

Files: список файлов, файл, директория, список директорий.

Опции:

- a – не игнорирует файлы, начинающиеся с ‘.’;
- c – вывод ctime с ключом -l, сортировка по ctime с ключом -t;
- t – сортировка по заданному ключу, default – последняя модификация;
- u – вывод access time;
- i – вывод inode для каждого файла;
- l – вывод информации в длинном формате;
- g – то же, что и -l, но без вывода владельца;
- o – то же, что и -l, но без вывода группы;
- d – вывод информации о директории, а не о её содержимом.

Пример:

```
ls -l test
-rwxr-r-- 1 s191944 studs 5 июля 10 23:34 test
```

-rwxr-x--- :

- ‘.’ – тип файла: обычный файл;
- ‘rwx’ – права для владельца: чтение, запись, исполнение;
- ‘r--’ – права для группы: чтение, исполнение;
- ‘---’ – права для остальных пользователей: нет прав;
- 1 – количество жёстких ссылок на файл;

s191944 – имя пользователя;

studs – главная группа пользователя;

5 – размер в байтах;

июля 10 23:34 – время последней модификации;

test – имя файла.

На поведение утилиты влияют переменные окружения COLUMNS и переменные языковых и региональных настроек (LANG, LC\_ALL, TZ, etc).

## 2. pwd – “print working directory”

pwd [options]

Выводит имя текущей директории.

Options:

- L – logical – учитывая symlinks;
- P – physical – не учитывая symlinks.

## 3. cd – “change the working directory”

cd [option] args

Изменяет рабочую директорию.

Options:

- L – учитывая symlinks;
- P – не учитывая symlinks.

Т.е. с флагом -P текущей рабочей директорией будет считаться та, на которую указывала ссылка, а не сама ссылка, как с флагом -L (default). args:

- ‘.’ – предыдущая директория; берёт значение из переменной окружения OLDPWD;
- ‘ ’ – директория, заданная в переменной окружения HOME;
- ‘~’ – домашняя директория;
- ‘\’ – корневая директория.

Переменные окружения: HOME, OLDPATH, PWD (имя текущей директории), CDPATH (если указанная директория не находится в данной относительной ссылке, проверяются пути из CDPATH, где бы данная директория

могла находиться), а также настройки локали.

#### 4. **rm** – “remove”

`rm [option] [files]`

Удаляет файлы.

Options:

- r – рекурсивно удалять директории и поддиректории с содержимым; требует права на чтение директории;
- f – удалять файлы без запроса к пользователю;
- i – запрос к пользователю на разрешение удаления.

#### 5. **mv** – “move files”

`mv [options] source target`

`mv [options] source... directory`

`mv [options] -t dir source...`

Option:

- f – без запроса к пользователю на разрешение;
- i – с запросом к пользователю на разрешение.

Если `target` и `source` обычные файлы, `mv` меняет имя `source` на имя `target`, даже если `target` уже существовал.

#### 6. **cp** – “copy”

`cp [options] source target`

`cp [options] source... directory`

Копирует файл.файлы в директорию.

Option:

- r – рекурсивно копировать директории и все их файлы в `target`;
- L – копирует содержимое файла, на который указывает `symlink`;
- P – копирует сам `symlink`;
- f – не просить разрешение;
- i – просить разрешение.

#### 7. **mkdir** – “make directories”

`mkdir [options] name`

Создает директорию.

Options:

- m `mode` – задаёт режим `mode` для новой директории;
- p – создать всё несуществующие директории, заданные в пути `dir`.

#### 8. **rmdir** – “remove directory”

`rmdir [option] directory`

Удаляет директорию.

Option:

- p – удаляет все пустые директории в пути `dir`, пока не встретит директорию с иными файлами;
- s – подавляет выводимые в `stderr` сообщения при -p.

#### 9. **type** – “type”

`type [options] names`

Описание типа команды: встроенная в shell, функция, псевдоним etc.

Options:

- p – возвращает имя программы, которая будет исполнена;
  - a – возвращает все исполняемые имена файла, включая алиасы.
- Переменные окружения: `PATH` (пути поиска для определения типа).

#### 10. **file** – “file”

`file [option] files`

Определяет тип `file`.

Options:

- f – интерпретирует содержимое `file` в качестве списка файлов на проверку;
- h – может распознать файл как `symlink`.

## 11. find – “find”

find [option] path [expression]

path – каталог, с которого нужно начать поиск.

Option:

-H – не следовать symlink.

-L – следовать symlink.

Expression:

-prune – остановить поиск на текущем путевом имени;

-atime n – время последнего доступа к файлу n дней назад;

-ctime n – время последнего изменения статуса файла n дней назад;

-mtime n – время последнего изменения содержимого файла n дней назад;

-name шаблон – поиск по шаблону по именам файлов;

-exec cmd; – выполнить команду cmd, аргументы которой оканчиваются “;”, подставляя в “” имя файла;

-inum n – находит файл с inode n;

-type c – поиск по типу файла; f – file, d – directory, l – symbolic link etc.

Operators: 1. ! expr – not expr;

2. expr -a expr – expr AND expr;

3. expr -o expr – expr OR expr;

Переменные окружения: PATH (поиск исполняемых файлов для -exec), а также важны настройки локали для интерпретации шаблонов и выражений.

## 12. chmod – “change mode”

chmod [option] mode files

Изменяет или назначает права доступа к файлу.

Options: -R – рекурсивно изменить права у файлов и поддиректорий. 1. Absolute mode:

Использует восьмиричные числа nnnn; n – числа от 0 до 7:

n0 – режим для остальных пользователей;

n1 – режим для группы;

n2 – режим для владельца;

n3 – установка SUID/SGID.

SUID/SGID: Set User ID / Set Group ID – выполнение от имени пользователя / группы для исполняемых файлов. Для директории sgid означает, что все файлы, созданные в данной директории, будут наследовать gid от этой директории.

n[0-2]:

1 – x – исполнение;

2 – w – запись;

4 – r – чтение.

n3:

2 – права группы;

4 – права пользователей.

### 2. Symbolic mode

Использует символьные обозначение по шаблону:

who operator permissions

-> who:

- u – user;

- g – group;

- o – other;

- a – all;

- ‘ ’ – all by default.

-> operator:

- ‘+’ – добавление прав;

- ‘-’ – отнимает права;

- ‘=’ – установить perm, очистив пустые биты.

-> permissions:

- ‘r’ – read;

- ‘w’ – write;

- ‘x’ – execute;

- ‘s’ – set-ID;

- 't' – sticky bit;  
Sticky bit – защищает файлы директории от удаления не владельцем файла/директории/root; кэширует исполняемые файлы.  
chmod не меняет права на саму символическую ссылку.

### 13. ln – “link”

ln [option] source... target

Создание жёсткой или символической ссылки на файл. Source – файл / файлы, target – директория: создание жёстких ссылок на файл / файлы в target.

Option:

-f – выполнение без запроса к пользователю.

-s – создание символической ссылки.

### 14. wc – “word count”

wc [options] files

Подсчёт количества единиц, заданных параметром. Default: линии, слова, символы.

Options:

-c – число байтов.

-m – число символов.

-l – число линий.

-w – число слов (слово – последовательность символов, оканчивающаяся пробелом или символом новой строки).

Изменение переменной окружения LANG меняет вывод программы вследствие изменения интерпретации символов.

Фильтр:

```
who | wc -l
```

### 15. tee – “tee” – “тройник”

tee [options] files

Копирует из stdin в stdout, делая копию в файл.

Option:

-a – дописывает в файлы, а не перезаписывает их.

-i – игнорирует прерывания.

Фильтр:

```
find . -name "[0-9][0-9]*" | tee found | wc -l
```

### 16. cat – “catenate”

cat [options] files

Объединяет и выводит файлы на stdout.

Option:

-n – добавляет каждой линии её порядковый номер перед выводом;

-v – режим отображения непечатаемых символов.

-ve – в конце каждой строки печатает \$.

files:

' ' – stdin, вывод на stdout; '-' – читает из stdin, без вывода на stdout.

Фильтр:

```
find . -name "[0-9][0-9]*" | tee found | cat -n
```

### 17. tail – ‘tail’ – “задний”

tail [option] file

Выводит на stdout содержимое файла с конца, отсчитывая определённое количество единиц; default – 10 линий.

Option:

-n – n - число линий;

-c – единица - байт;

-b – единица - блок;

-l – единицы - линия.

Фильтр:

```
dmesg | tail -15
```

### 18. head – “head” – “передний”

head [-number] files

Выводит number количество линий каждого из files.

Фильтр:

```
listusers -g staff | head -5
```

### 19. more – “more”

more [options] files

Постранично отображает текстовый файл на окно терминала.

Option:

-c – очищает терминал перед выводом;

-e – выход после вывода конца файла;

Навигация:

v – войти в режим редактора.

Переменные окружения: COLUMNS (ширина отображаемого экрана), LINES (высота отображаемого экрана), MORE (параметры запуска more), EDITOR (используется с командой v).

Фильтр:

```
env | more
```

### 20. pg – ‘page’

pg [-number] [-p str] [option] [pattern] file

Отображение текста на экран.

Опции:

-number – длина “окна” в линиях;

-p str – символ в конце строки.

Options:

-e – не делает паузу при достижении конца файла;

+/pattern/ – начинает с первой строки, подходящей под шаблон.

Переменные окружения: COLUMNS, LINES, SHELL (определяет оболочку для выполнения команд типа !command из pg), TERM (определяет атрибуты терминала, используемые для настройки отображения и навигации).

Фильтр:

```
env | pg
```

### 21. touch –

touch [option] files

Устанавливает время доступа и модификации для файла.

Option:

-a – меняет время доступа;

-m – меняет время модификации;

-c – не создает файл, если он не существовал до вызова;

-d datetime – использует время из datetime для обновления.

Format: YYYY-MM-DDThh:mm:SS

-t time – использует time.

Format: CCYYMMDDhhmm

Переменные окружения: региональные настройки (TZ – time zone).

### 22. su – “superuser”

su [-] username [options]

Позволяет сменить пользователя без выхода из системы.

Default username: root.

‘-’ – меняет текущие переменные окружения на переменные окружения username.

Options:

-c command – запускает command от имени username.

Переменные окружения: SUDOLOG (указывает путь до файла логгирования), SYSLOG (определяет поведение функции syslog), CONSOLE(определяет логгирование попыток su root на экран устройства), PATH (устанавливает начальное значение для оболочки), SUPATH (PATH для root), а также важны региональные настройки и настройки локали.

### 23. chown – “change owner”

chown [options] owner[:group] files.

Устанавливает owner ID и, опционально, group ID на files.

Options:

-f – не сообщать об ошибках.

-h – если file - символная ссылка, изменить владельца ссылки.

-R – изменить владельца во всех файлах директории.

### 24. chgrp – “change group”

chgrp [options] group files

Устанавливает GID group на files.

Options – аналогично chown.

### 25. less – “opposite of more”

less [options] files

Станичезатор с расширенным функционалом от more и vi.

Options:

Позволяет задать большое количество параметров, изменяющих поведение less. Фишка: не нуждается в чтении всего файла для загрузки, как, скажем, vi.

-h/-help/-? – мануал по использованию всех прелестей less.

-c – очищает экран.

-C – ОЧИЩАЕТ ЭКРАН.

-r patten – подсвечивает все найденные pattern в тексте.

Переменные окружения: LESS (устанавливает параметры запуска), LESS\_IS\_MORE (1 - поведение аналогично с спецификацией POSIX more), ряд переменных LESS\*, определяющие специфичные особенности less, а также аналогично rg, учитывая настройки локали. Фильтр:

```
find . -name "*.rc" | xargs less
```

### 26. split - “split”

split [options] filename prefix

Читает filename и записывает кусочки, поделенные на число linescount в файлы с префиксом prefix; если он не указан, то выходные файлы начинаются с префикса “x”.

Options:

-l linescount – число линий, по которым необходимо делит файл.

-a suffixlength – число символов в суффиксе выходного файла (default:2).

Если выходных файлов больше, чем хватает суффиксов для обозначения их, то возникает ошибка.

-b n – делить по байтам, nk – Кб, nm – Мб.

Фильтр:

```
listusers | split -l 10 - list
```

### 27. join

join [options] file1 file2

Формирует объединение двух строк из file1 и file2 по параметрам, заданным options. Default: вывод строк с совпадающими первыми полями.

Options:

-t s – задаёт разделитель полей s;

-a n – включает строку без найденных соответствий из файла под номером n [1-2];

-o format – вывод под формату; format (номер-файла.поле) задаёт правило формирования строчек;

-1 / -2 – задаёт поле, по которому объединять.

Фильтр:

```
sort usernames | join useraddress | cat -n
```

### 28. paste

paste [-s] [-d list] files

Сливает соответствующие строки из files, следуя заданным параметрам, и выводит результат на stdout. Default: выводит соответствующие строки, заменяя символ перевода строки на ТАВ.

Options:

-s – последовательный вывод строк из файлов, а не параллельный.

-d list – список разделителей, взамен ТАВ.

Фильтр:

```
cut -d':' -f1 /etc/passwd | paste - -
```

### 29. cut

cut [options] files

Вырезает выбранные соответственно параметрам столбцы файла и отправляет текст на stdout.

Options:

-b list – вывод list-позиций байт (-b3-34 – с 3 по 34 байт);

-c list – вывод list-позиций символов;

-f list – вывод list-позиций полей, разграниченных разделителем ТАВ;

-d delim – переопределяет разделитель для -f.

Фильтр:

```
echo $PWD | cut -d'/' -f3
```

### 30. tr – “translate”

tr [-cds] string1 [string2]

Преобразует текст со stdin согласно параметрам и выводит на stdout.

Options:

-s string1 string2 – заменяет символы из string1 соответствующими символами из string2.

-d string1 – удаляет символы, перечисленные в string1.

-c string1 – дополняет набор символов, заданных string1

Поведение меняется в зависимости от LANG: на затейливый string даёт “Bad string”.

Фильтр:

```
env | cut -d'=' -f1 | tr "\n" ", "
```

### 31. cmp – “compare”

cmp [-l] [-s] file1 file2 [skip]

Сравнивает два файла построчно, пока не наткнётся на различие или конец файла одного или обоих файлов. В первом случае вывод номера байта и номера строки, где начинаются отличия, во втором – сообщение о EOF. В случае совпадения ничего не выводит.

Options:

-l – печатает номер строки (10) и два восьмиричных кода различающихся байт.

-s – ничего не выводит при отличии; изменяется лишь exit status.

Фильтр:

```
./test_program | tee user_answers | cmp -s - right_answers && echo "Good work!"
```

### 32. diff - “diffetence”

diff [options] file1 file2

Сравнивает два файла и выводит минимальный список различий.

Options:

-i – игнорирует регистр;

-c – дополняет вывод датами последней модификации файлов и разделителями.

Переменные окружение: TZ (используется при ключе -c), а также региональные настройки.

Фильтр:

```
cat -n code.c | diff -c - code.c
```

### 33. sort – “sort”

sort [options] [more options] files

Сортирует объединение всех указанных файлов и записывает результат на stdout.



Options:

- r – сортировка в обратном порядке;
- c – отсортирован ли файл;
- u – удаление повторяющихся строк;
- o – вывод в указанный файл;
- n – режим числовой сортировки;
- t – задание разделителя;
- i – игнорировать повторяющиеся имена при сортировке;
- M – интерпретировать первые три символа в верхнем регистре в качестве наименования месяцев (DEC, JAN, FEB);
- k key – сортировать, следуя заданным параметрам key; вариация: field\_start[type][,field\_end[type]], field – номер поля, type – тип сортировки.

Переменные окружения: LC\_COLLATE (принцип сортировки меняется в зависимости от установленной переменной: C – сортирует в порядке ASCII, en\_US – без учёта регистра, etc), LC\_STYPE, а также прочие настройки локали.

Фильтр:

```
cat /etc/passwd | sort -t: -k3,3n | head -10
```

### 34. **uniq** – “unique”

uniq [options] input\_file [output\_file]

Читает и сравнивает соседние строки, записывая копию каждой линии в выходной файл. Если файлы не указаны, использует stdin и stdout.

Options:

- c – перед каждой строкой пишет количество её повторений;
- d – выводит лишь строку, которая имела дубликаты;
- fn – игнорировать первые n полей.

Фильтр:

```
sort list | uniq -d - uniq_list
```

### 35. **echo** – “echo”

echo [options] [args]

Пишет аргументы, разделённые пробелами, на stdout; в случае отсутствия аргументов выводит символ перевода строки; отображает переменные среды (для отображения следует указать символ \$). А также в аргументах можно передать программу, заключённую в ‘cmd’ кавычки; тогда echo отобразит вывод команды на экран.

Пример: echo "Good morning, \$USER! Now is 'date'"

Options:

- e – экранирование спец. символов в аргументах (\n, \t, \c, \xp (где n – ASCII 16-ричный код)).

### 36. **alias** – “alias”

alias

Создаёт псевдоним для команды или серий команд или печатает список псевдонимов. Созданные алиасы хранятся на протяжении одной сессии; для продолжительного использования следует исправить конфигурационные файлы.

### 37. **ulimit** – “user limit”

ulimit [options]

Выводит/устанавливает ограничения на системные ресурсы для пользователя.

Options:

- a – all – весь список лимитов;
- S – soft limits;
- H – hard limits.

Установить/показать ограничение на:

- n – максимальное количество открытых файлов;
- u – максимальное количество процессов;
- f – максимальный размер файла; default.

### 38. **umask** – “user mask”

umask [-S] [mode]

Устанавливает или показывает режим создания файлов.

Option:

-S – при выводе или установке использовать символьное представление прав.

## Дополнительное задание

1.

```
* paste -d'\0' <(cat file | tr "\n" " " | cut -d" " -f8-12) <(cat file | tr "\n" " " | cut -d" " -f1-7) |  
tr -s '[:blank:]' " " >&2
```

```
*paste -d'\0' <(tail -2 file | tr "\n" " ") <(head -5 file | tr "\n" " " ) |  
tr -s '[:blank:]' " " >&2
```

```
* A='tail -2 file | tr "\n" " "' ; B='head -5 file | tr "\n" " "' ; echo "$A$B" |  
tr -s '[:blank:]' " " >&2
```

2.

```
* A='paste -d" " <(cat file | tr "\n" " " | cut -d" " -f3-5) <(cat file | tr "\n" " " |  
cut -d" " -f6-7 | tr "this" "THAT")'; echo "$A -- password" >&2
```

```
*echo " -- password" | cat <( paste -d" " <(cat file | tr "\n" " " | cut -d" " -f -d" " -f6-7 |  
tr "this" "THAT")) - | tr "\n" "\0" >&2
```

## Выводы:

1. Несмотря на специфичные отношения некоторых утилит с переменными окружения, на них всех влияют настройки локали (LANG, LC\_ALL, etc).
2. Для сохранения значений переменных необходимо прописать их в конфигурационных файлах, которые инициализируются при пользователя регистрации в системе, иначе установленные значения будут существовать в пределах одной сессии.
3. Ядро не делает различий между типами regular-файлов: если сказать ему исполнить обычный файл (-rwxr-x--- ./simple ), оно постарается это сделать и вернёт ошибку.
4. Программа – исполняемый код. Утилита – системная программа. Команда – просьба интерпретатору попытаться организовать исполнение программ в заданной организации.