

# Администрирование операционной системы Linux

Лекция 9. Текстовые файлы и потоки (часть 2)

Ст. преп., к.т.н. Митричев Иван Игоревич

Москва 2020

# План лекции

- Простое форматирование текста.
- Подготовка текста к печати.
- Замена символов табуляции на пробелы.
- Команды выбора, объединения строк.
- Замена символов с помощью команды tr.
- Способы объединения файлов.
- Получение дампа.
- Разделение файлов на части.
- Команда xargs.
- Сортировка sort.

# Простое форматирование текста

`fmt` ← • форматирует текст так, чтобы каждая строка имела заданную равную длину (по умолчанию – 75 символов).

Опции: -w – позволяет указать ширину текста в символах;  
-s – только разделять строки, но не объединять их.

## Подготовка текста к печати

`pr +2` ← • вывести в стандартный поток вывода подготовленный к печати текст файла, начиная со второй страницы;

`pr -2` ← • форматировать для печати в две колонки.

Опции: -w – устанавливает ширину страницы (по умолчанию – 72 символа);  
-o – создать отступ слева в заданное количество символов;  
-d – двойной интервал.

# Команда expand

expand

- преобразование символов табуляции в пробелы (8 пробелов вместо 1 TAB).

unexpand

- преобразование пробелов в табуляцию.

-ЧИСЛО – указать количество пробелов;

Опции:

-t – установить позиции табуляции (если указывается только одно значение, то позиции табуляции будут расставлены через данный интервал).

-a – заменить все пробелы, а не только ведущие в строках.

Пример:

```
$ cat text2
4      plum
1      banana
9      apple
$expand -1 text2
4 plum
1 banana
9 apple
```

# Команда uniq

uniq

- ← • удаление повторяющихся строк из потока (в т.ч. - файла).

Опции:

**Работает с предварительно отсортированными файлами.**

-c – посчитать количество повторов каждой строки;

-d – вывести только повторяющиеся строки;

-u – только уникальные строки;

-i – игнорировать регистр;

Примеры:

-f<n> – пропустить n полей до определения уникальности.

```
cat file.txt
```

```
Привет Иванов И.И. MAR 2019 4.290
```

```
Кошка Петров В.В. FEB 2019 3e10
```

```
Кошка Петров В.В. FEB 2008 3e10
```

```
Алтын Friday В.В. MAY 1994 1.000
```

```
uniq -f5 -d file.txt
```

- ← • вывести дублирующиеся строки, сравнивать, начиная с bго поля

```
Кошка Петров В.В. FEB 2019 3e10
```

# Команда comm

comm

- для **отсортированных файлов** вывести в три столбца строки, (1) которые есть только в первом файле, (2) которые есть только во втором файле, (3) которые есть в обоих файлах.

Опции: 1 2 или 3 – отменить вывод указанных столбцов.

Пример:

```
cat 1.txt      cat 2.txt
Кошь           Мышь
Мышь           Кошь
Еж             Еж
Ёрш           ёрш
```

comm 1.txt 2.txt

```
Кошь
      Мышь
comm: данные файла 1 не отсортированы
comm: данные файла 2 не отсортированы
Еж
Ёрш
      Кошь
      Еж
      ёрш
```

- Вывод в три столбца

# Команда tr

tr

- удаление или замена символов в тексте.

## Опции:

- c/-C - использовать не указанное множество символов, а его дополнение до полного множества символов (т.е., инвертировать выбор символов);
- d - удалять символы из множества;
- s - заменить повторяющиеся символы из множества на один символ;

Без опций: заменяет найденные в тексте элементы из первого множества символов на соответствующие элементы из второго множества символов

```
$ echo Privet | tr a-z A-Z  
PRIVET
```

- замена символов;

```
$ ps | tr -d '\n '  
PIDTTYTIMECMD3402pts/400:00:00bash3600pts/4  
pts/400:00:00bash3724pts/400:00:03bash3736p  
pts/400:00:00awk5057pts/400:00:00cat6649pts
```

- удаление символов «перевод строки» и «пробел»;

```
tr -cs "[:alpha:]" "\n" <file1 >file2
```

- все символы, кроме букв, заменить на символ новой строки. -s позволяет заменить идущие вплотную небуквенные символы только на один символ новой строки.

# Команды join, nl

join

← • объединить два отсортированных по заданному полю файла.

Опции:

-t – разделитель полей;

-1 <n> -2 <m> – номера общих полей;

-a<n> – вывести также непарные строки из файла n;

-v<n> – вывести только непарные строки из файла n.

Пример:

**cat 1.txt**

drwxr-xr-x 2 root root 4096 Jan 9 21:18 mysql

**cat 2.txt**

2 10 20

join -1 2 -2 1 1.txt 2.txt

drwxr-xr-x 2 root root 4096 Jan 9 21:18 mysql 10 20

nl или  
cat -b

← • нумерация строк.

# Команды создания дампов и paste

```
paste <file1> <file2>
```

- соединить файлы построчно;

```
paste -d ' ' <file1> <file2>
```

- соединить через разделитель-пробел

```
paste - - <file>
```

- вывести файл, соединив по две строки в одну

```
paste -s <file>
```

- соединить все строки файла;

```
hexdump  
xxd  
od
```

- получить двоичную, восьмеричную, десятичную или шестнадцатеричную репрезентацию файла

двоичная:

```
xxd -b file.txt
```

шестнадцатеричная:

```
xxd file.txt
```

```
od -x file.txt
```

восьмеричная:

```
hexdump -o file.txt
```

```
hexdump -C file.txt
```

```
od -o file.txt
```

# Разделение файлов на части

split

- разделить файл на части именами хаа, хаб, хас ...

По умолчанию команда записывает в отдельный файл каждые 1000 строк текста.

Опции: `-l<кол-во строк>` – указать количество строк в частях файла;

`-b<кол-во байтов>` – указать количество байтов в частях файла;

`-n` – разрезать на `n` файлов; `-d` – использовать численный суффикс `x00, x01, ...`

`csplit <file> <expression>`

- разделить файл на две части: до найденной строки и после нее;

`csplit <file> <expression> {n}`

- повторить поиск и деление `n` раз (на выходе – файлы `xx00, xx01, ...`);

# Команда xargs

xargs

← • выполнить команду, составив ее из входящих аргументов.

Часто используют связку xargs + find.

Пример:

поиск и удаление временных файлов (начинаются с «~»)

```
find ~ -type f -name '^~*' -exec rm -f {} \;
```

← • опция exec;

ИЛИ

```
find ~ -type f -name '^~*' | xargs rm -f
```

← • конвейер + xargs.

# Сортировка строк

sort

←• сортировка строк текста.

## cat file.txt

```
Привет Иванов И.И. March 2019 4.290
Кошка Петров В.В. February 2019 3e10
Кошка Петров В.В. February 2008 3e10
Алтын Friday В.В. May 1994 1.000
```

## Опция -k

-k, --key POS - сортировать по столбцу POS, а далее, - по столбцам после него.

```
sort -k4,4 file.txt
```

## Опция -k POS1,POS2

Сортировка по 4-му полю. После этого - сортировка по строке целиком (POSIX-свойство), поэтому Петров 2008 поднимается выше, чем Петров 2019.

```
Кошка Петров В.В. FEB 2008 3e10
Кошка Петров В.В. FEB 2019 3e10
Привет Иванов И.И. MAR 2019 4.290
Алтын Friday В.В. MAY 1994 1.000
```

-k POS1,POS2 - сортировка по полям от POS1 до POS2, далее - с начала строки. Чтобы отключить сортировку с начала строки, используйте -s.

# Сортировка по полям

## Опция -k ЧИСЛО.ЧИСЛО

Сортировка со второй буквы первого слова

```
sort -k1.2 file.txt
```

```
Алтын Friday B.W. MAY 1994 1.000  
Кошка Петров B.B. FEB 2008 3e10  
Кошка Петров B.B. FEB 2019 3e10  
Привет Иванов И.И. MAR 2019 4.290
```

## Опция -s

Устойчивая сортировка со второй буквы первого слова до конца слова (без ",1" устойчивая сортировка не произойдет, так как сортировка выполняется до конца строки, и Петров 2008 поднимется выше Петров 2019)

```
sort -k1.2,1 -s file.txt
```

```
Алтын Friday B.W. MAY 1994 1.000  
Кошка Петров B.B. FEB 2019 3e10  
Кошка Петров B.B. FEB 2008 3e10  
Привет Иванов И.И. MAR 2019 4.290
```

# Численная сортировка

## Опция -g

Сортировка по числу с плавающей запятой в 6-ом столбце

```
sort -k6 -g file.txt
```

Алтын Friday B.W. MAY 1994 1.000  
Привет Иванов И.И. MAR 2019 4.290  
Кошка Петров В.В. FEB 2008 3e10  
Кошка Петров В.В. FEB 2019 3e10

## Опция -n

Простая численная сортировка сортирует по числу до точки или буквы "e"

```
sort -k6 -n file.txt
```

Алтын Friday B.W. MAY 1994 1.000  
Кошка Петров В.В. FEB 2008 3e10  
Кошка Петров В.В. FEB 2019 3e10  
Привет Иванов И.И. MAR 2019 4.290

# Смена разделителя для сортировки

## Опция -t

Сортировка по отчеству (W<B<И)

```
sort -t '.' -k2 file.txt
```

Алтын Friday B.W. MAY 1994 1.000  
Кошка Петров B.B. FEB 2008 3e10  
Кошка Петров B.B. FEB 2019 3e10  
Привет Иванов И.И. MAR 2019 4.290

# Другие опции команды сортировки

- b, --ignore-leading-blanks - опустить ведущие пробелы;
- d, --dictionary-order - словарная сортировка (учитываются буквы, цифры и пробелы);
- f, --ignore-case - игнорировать регистр символов;
- h, --human-numeric-sort - сравнение размеров («5К», «2М»);
- i, --ignore-nonprinting - сортировать только по печатным символам;
- r, --reverse - сортировка в обратном порядке;