

Фильтры

(команды обработки текстовых файлов)

- Читают стандартный ввод и генерируют стандартный вывод.
- Осуществляют фильтрацию входного потока или файла.
- Посылают результаты на экран, не изменяя входной поток или файл.
- Обработывают вывод других команд, когда они используются совместно с перенаправлением вывода

Примеры фильтров - cat, grep, sort, wc

WC - подсчет количества строк, слов и символов

Синтаксис:

`wc [-lwc] [файл...]` Подсчитывает число строк(-l), слов (-w) и символов (c) в файле

Примеры

`$ wc file` *рассматривается как аргумент команды*

115 567 3345 file

`$ wc -l file`

115 file

`$ ls > ls.out` *подсчет количества файлов в каталоге*

`$ wc -w ls.out`

72 ls.out

WC - подсчет количества строк

Примеры

```
$ wc <Enter>
```

```
ab cde
```

```
Dvbnvnsbvbnd
```

```
dhjhjd hfkdhjf
```

```
<Ctrl><d>
```

```
3 4 30
```

подсчитывается количество строк, слов и символов, вводимых со стандартного ввода

```
$ wc < myfile
```

```
105 715 2456
```

```
$ wc -w myfile
```

```
715 myfile
```

стандартный файл заменяется на файл myfile, имя файла не выводится

Сравнение файлов

cmp *опции* **файл_1** **файл_2** - побайтовое сравнение файлов

Пример

cmp -l file1 file2 - сравнение file1 и file2 и вывод номера байта и его значения при несовпадении

comm [-123] **файл_1** **файл_2** – построчное сравнение файлов и вывод результата сравнения в 3 столбца:

- ☐ • строки, содержащиеся только в файле_1
- ☐ • строки, содержащиеся только в файле_2
- ☐ • общие строки файлов

sort - алфавитная и числовая сортировка

Синтаксис:

sort [-ndutX] [-k номер_поля] [файл...] Сортирует строки

Примеры

\$ sort file *рассматривается как аргумент команды*

\$ sort < file *перенаправление ввода*

\$ tail -1 /etc/passwd

user3:xyzbkd:303:30:student:user3:/home/user3:

/usr/bin/sh

\$ sort -nt: -k 3 < /etc/passwd

\$ who > who1

\$ sort who1 *упорядочивание пользователей в системе по алфавиту*

\$ sort -u who1 *упорядочивание с исключением повторяющихся строк*

sort - опции сортировки

none	лексикографическая (по кодам ASCII)
-d	по принципу словаря (игнорируя любые символы, не являющиеся буквами, цифрами или пробелами)
-n	числовая
-u	уникальная (исключаются повторяющиеся строки)
-tX	указание разделителя (X - символ-разделитель полей)
-kn	выбор базового поля n для сортировки(по умолчанию n=1)
-r	сортировка в обратном порядке

grep - поиск по модели

Синтаксис:

```
grep [-cinv] [-e] модель [-e модель] [файл...]
```

```
grep [-cinv] -f файл_с_моделями [файл...]
```

Примеры

```
$ grep user /etc/passwd
```

```
$ grep -v user /etc/passwd
```

```
$ grep -in -e like -e love file
```

```
$ who > who1
```

```
$ grep -f who1 /etc/passwd
```

grep - опции

- c подсчитывать только число строк, содержащих заданную модель
- i игнорировать регистр букв в модели
- n сопровождать выводимые строки номерами
- v выводить строки, не содержащие заданную модель поиска

grep - построение модели

В качестве модели можно задать некоторый текст. Такая модель называется регулярным выражением. Для их задания можно использовать некоторые специальные символы:

- ^ поиск совпадения в начале строки
- \$ поиск совпадения с конца строки
- .
- *
- [] задается совокупность символов
- [-] задается диапазон символов
- [^] инвертирует процесс отбора

! Во избежание проблем при интерпретации shell'ом указанных специальных символов, лучше заключить регулярное выражение в кавычки.

Команда cut

Синтаксис:

cut -c*список* [*файл...*] Выделяет колонки или поля из файлов или стандартного ввода

cut -f*список* [**-d***символ*] [**-s**] [*файл...*]

Опции:

- c выделить колонок (столбцов)
- f выделить полей (ограниченных разделителем)
- d*символ* указать символа - разделителя полей
- s указать на пропуск любой строки, в которой нет разделителя

Команда cut

Список - последовательность чисел, которая указывает на подлежащие выделению поля или колонки.

Форматы для задания списка полей или колонок:

- A-B Поля или колонки от A до B включительно
- A- От поля или колонки A до конца строки
- B С начала строки до поля или колонки B
- A,B Поля или колонки A и B

Допустимо использовать любую комбинацию форматов.

Команда cut

Примеры:

```
$ date | cut -c1-3
```

```
Fri
```

```
$ tail -1 /etc/passwd
```

```
user3:mdhbmkdj:303:30:student:  
/home/user3:/usr/bin/sh
```

```
$ cut -f1,6 -d":" /etc/passwd
```

```
user3: /home/user3
```

```
$ cut -f1,6 -d":" /etc/passwd |sort -r
```

```
...
```

```
$ cut -f1,3,5-7 -d":" /etc/passwd
```

```
user3:303:student user3:/home/user3:/usr/bin/sh
```

Команда tr

Синтаксис:

`tr [-s] строка1 [строка2]` Преобразует символы *строки1* на соответствующие им символы *строки2*

Опции:

`-s строка1` Заменяет все символы строки1, встречающиеся несколько раз подряд на одиночное вхождение

Примеры

`$ who | tr -s " "` преобразование нескольких последовательно расположенных пробелов в один

`$ date | cut -c1-3 | tr "[:lower:]" "[:upper:]"`
перевод символов строк из нижнего регистра в верхний

Команда pr

Синтаксис:

pr [-опция] [файл...] Форматирует стандартный ввод и направляет его в стандартный вывод

Опции:

- k Обеспечивает вывод k столбцов сверху вниз
- a Обеспечивает вывод в несколько столбцов, используется совместно с опцией - k, печать осуществляется построчно
- t Удаляет заголовок и хвостовик на страницах
- d Разделяет строки пустой строкой
- wN Устанавливает длину строки в N символов
- lN Устанавливает N строк на странице
- nCK Задаёт нумерацию из K цифр, отделяемых от строки символом C (по умолчанию C - знак табуляции <Tab>)
- oN Смещает вывод на N столбцов от левого края страницы
- p Устанавливает режим ожидания нажатия клавиши <Enter> перед каждой страницей
- h*строка* Следующая за этой опцией *строка* используется в качестве заголовка

Команда pr

Примеры:

\$ pr -n3 file нумерация строк

\$ pr -n3 file | more

\$ ls | pr -3 вывод 3 столбцов

\$ grep /home /etc/passwd | pr -h "User accounts"
 добавление заголовка

Shell-подстановки

- Подстановка переменных
- Подстановка команд
- Тильда-подстановка

Подстановка переменных

Синтаксис:

`$переменная` Указывает shell'у выполнить подстановку переменной

Пример:

```
$ echo $PATH
/usr/bin:/usr/contrib/bin:/usr/local/bin

$PATH=$PATH:$HOME:.

$ echo $PATH
/usr/bin:/usr/contrib/bin:/usr/local/bin:/home/user3:.

$echo $HOME
/home/user3

$ file_name=$HOME/file1

$ more $file_name
<contents of /home/user3/file1>
```

Использование фигурных скобок ({})

```
$ file=this
$ file1=that
$ echo $fileand$file1
sh: fileand: parameter not set
$ echo ${file}and$file1
thisandthat
```

Подстановка команд

Синтаксис:

`$(команда)` или ``команда``

Указывает shell'у

выполнить подстановку команды

Пример:

```
$pwd
/home/user2
$curdir=$(pwd)
$echo $curdir
/home/user2
$cd /tmp
$pwd
/tmp
$cd $curdir
$pwd
/home/user2
$curdir2=`pwd`
$echo $curdir
/home/user2
```

на букве Ё



Тильда-подстановки

- Отдельный знак `~` или `~/` заменяется на значение переменной *HOME*.
- `~+` заменяется на значение переменной *PWD*.
- `~-` заменяется на значение переменной *OLDPWD*.
- Если за знаком тильда следуют несколько символов, а затем символ `/`, то shell пытается установить не являются ли эти символы именем пользователя в системе. Если это так, то последовательность символов заменяется маршрутным именем начального каталога пользователя.

Символ `~` может быть помещен в псевдоним команды.

Тильда-подстановки - примеры

```
$ echo $HOME
```

```
/home/user3
```

```
$ echo ~
```

```
/home/user3
```

```
$ echo $PWD
```

```
/home/user3/tree
```

```
$ ls ~+/poodle
```

```
/home/user3/tree/poodle
```

```
$ echo $OLDPWD
```

```
/home/user3/mail
```

```
$ ls ~-
```

```
/home/user3/mail/from.mike /home/user3/mail/from.jim
```

```
$ echo ~vova/file1
```

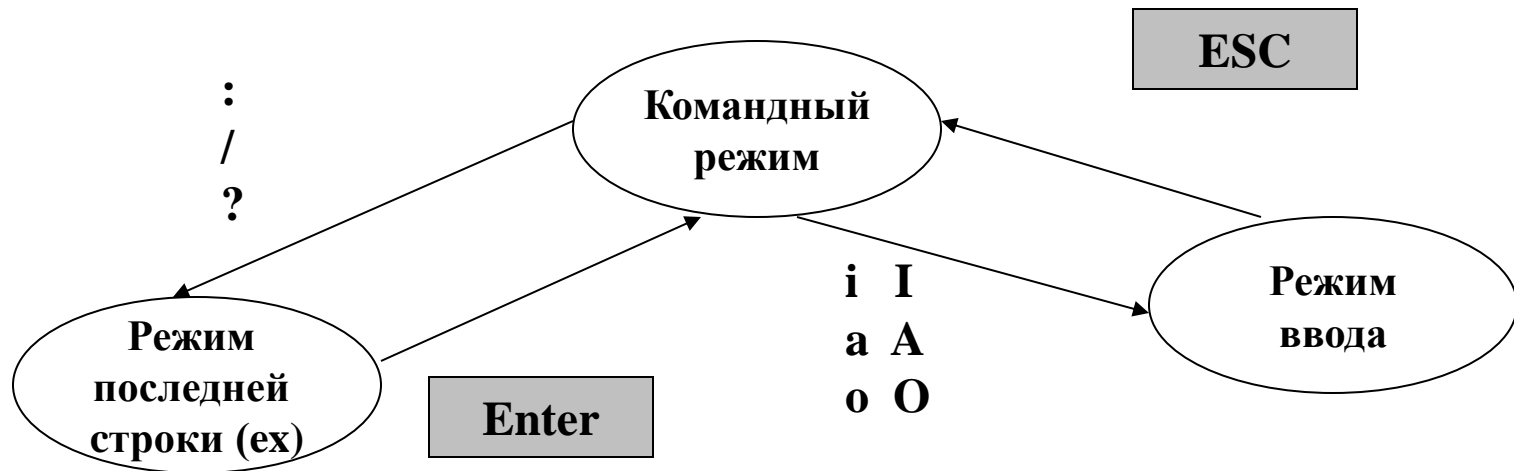
```
/home/vova/file1
```

Текстовый редактор vi

- поставляется в большинстве версий системы UNIX
- управляется командами
- включает следующие классы команд:
 - общеорганизующие
 - управления курсором
 - вставки текста
 - удаления текста
 - вставки ранее удаленного текста
 - изменения текста

vi: Режимы работы редактора

- **Командный режим** - нажатия клавиш интерпретируются как команды
 - является исходным режимом при начале сеанса vi
 - нажатие клавиши <Esc> - переход в командный режим
 - vi команды не отражаются на экране
- **Режим ex** - команды префиксируются двоеточием(:) и отображаются на экране
- **Режим ввода** - информация, соответствующая нажатым клавишам вводится в файл



vi: Сеанс редактирования

содержимое файла

Это пример содержимого
файла test, который будет
рассматриваться далее

курсор

~
~
~
~
~
~

Пометки строк,
которые не существуют
в файле

: ex команды

сообщения
о режиме
работы

vi: Завершение сеанса редактирования

<Esc> ZZ <Enter>

Сохранение файла на диске
(на экране не отображается)

<Esc>:wq <Enter>

Сохранение файла на диске и
выход из редактора

<Esc>:q! <Enter>

Выход без сохранения

Команды управления курсором

h или <BackSpace>	Перемещение курсора на одну позицию влево
j	Перемещение вниз на одну строку
k	Перемещение вверх на одну строку
l или <Space>	Перемещение курсора на одну позицию вправо
#w	Перемещение вперед на слово (word) за словом
#b	Перемещение назад на слово (backword) за словом
#e	Перемещение к концу (end) следующего слова
\$	Перемещение в конец текущей строки
^ или 0	Перемещение в начало текущей строки

Команды управления курсором (продолжение)

G	Переход (Go) в конец файла
#G	Переход (Go) на строку с номером #
:#	Переход на строку с номером #
<Ctrl>+<g>	Вывод номера текущей строки
<Enter>	Переход к первому непустому символу на следующей строке
<Ctrl>+	Вывод предыдущего (back) содержимого экрана
<Ctrl>+<f>	Вывод следующего (forward) содержимого экрана
<Ctrl>+<u>	Возврат вверх (up) на половину экрана
<Ctrl>+<d>	Переход вниз (down) на половину экрана
L	Переход к последней (last) строке экрана
M	Переход к средней (middle) строке экрана
H	Переход к первой строке экрана (на первую строку, первый символ)
<Ctrl>+<l>	Перерисовка экрана

Режим ввода

Для переключения в режим ввода используются следующие команды:

- a** Добавление (append) текста после курсора
- i** Добавление (insert) текста перед курсором
- O** Открытие (Open) пустой строки над текущей строкой
- o** Открытие (open) пустой строки под текущей строкой
- A** Добавление (Append) текста в конец строки
- I** Вставка (Insert) текста с начала строки

В режиме ввода:

- <Enter>** Появление новой пустой строки (можно разрезать строки)
- <BackSpase>** Возвращение влево для исправления введенного
- <Esc>** Окончание режима ввода. Возвращение в командный режим

Удаление текста

- #x** Удаление (**X out**) символа над курсором
- #dw** Удаление (**Delete**) текущего слова (**Word**)
- #dd** Удаление текущей строки
- dG** Удаление всех строк от текущей до последней строки файла
- d\$** Удаление конца строки (от текущей позиции)
- d^** Удаление начала строки (до текущей позиции)

Можно задать число, показывающее сколько раз должна быть повторена команда. Например: 3dd - удалить 3 строки

Отмена произведенных изменений

- u** Отмена (**undo**) последнего изменения
- U** Отмена (**Undo**) всех изменений в текущей строке

Перемещение текста

d <i>объект</i>	Удаление <i>объекта</i> и сохранение его в буфере
p	Вставка (paste) содержимого буфера в текст после курсора
P	Вставка (Paste) содержимого буфера в текст перед курсором

Копирование текста

#yw	Копирование (yank) текущего слова в буфер
#yy	Копирование текущей строки
yG	Копирование строк от текущей до конца файла
y\$	Копирование части строки от курсора до конца строки
y^	Копирование части строки от курсора до начала строки

Список команд копирования и вставки

<i>уобъект</i>	Копирование (yank) <i>объекта</i> в буфер
p	Вставка (paste) содержимого буфера в текст после курсора
P	Вставка (Paste) содержимого буфера в текст перед курсором

Изменение текста

r <i>символ</i>	Замена (replace) символа над курсором на <i>символ</i>
R	Замена (Replace) всех символов строки, пока не нажата клавиша <Esc>
#cw	Изменение текущего слова
#cc	Изменение всей текущей строки
cG	Изменение всех строк файла от текущей до последней
c\$	Изменение части строки от курсора до конца строки
c^	Изменение части строки от курсора до начала строки
.	Повтор последнего действия по изменению текста, включая удаление

Поиск текста (команды ex)

<i>/текст</i>	Поиск строки, содержащей заданный <i>текст</i> , начиная от текущей строки до конца файла с переходом на начало файла
<i>?текст</i>	Поиск строки, содержащей заданный <i>текст</i> , начиная от текущей строки до начала файла с переходом на конец файла
n	Поиск следующего вхождения в файл заданной строки текста
N	Поиск предыдущего вхождения в файл заданной строки текста

Поиск текста по шаблону

<i>[o0]ld_text</i>	Поиск <i>old_text</i> или <i>0ld_text</i> .
<i>^текст</i>	Поиск строки начинающейся с <i>текста</i>
<i>текст\$</i>	Поиск строки заканчивающейся на <i>текст</i>
<i>.</i>	Поиск по любому одиночному символу.
<i>символ*</i>	Поиск по <i>символу</i> , который может не встретиться ни разу или любое число раз.
<i>.*</i>	Поиск по любому символу, который может не встретиться ни разу или любое число раз.

Список регулярных выражений (шаблонов для поиска)

[a-zA-Z0-9] Определяет класс символов, из которого формируется образец для сопоставления. Знак "тире" (—) означает допустимый диапазон символов. Квадратные скобки [] определяют только *один* символ.

^*текст* *Текст* от начала строки.

***текст*\$** *Текст* от конца строки.

. Соответствует любому одному символу.

символ* Определяет ноль или любое число повторений *символа*

Примеры шаблонов

`/[Tt]he`

`/[oO][nN][eE]`

`/bo*t`

`/^[abc].*`

`/finally.$`

Глобальный поиск и замена - команды ex

:m,ns/шаблон/новый_текст/[g]

Примеры

:1,\$s/one/two/

:.,10s/[oO][nN][eE]/two/g

Другие команды ex (1)

:w	запись текущего содержимого буфера на диск
:m,nw <i>файл</i>	запись строк с m-ой по n-ую из текущего буфера в файл
:w <i>файл</i>	запись текущего буфера в файл
:e <i>файл</i>	загрузка файла в буфер редактирования, замещающего старое содержимое буфера
:e !	отмена всех изменений в буфере перезагрузкой в него файла с диска
:r <i>файл</i>	добавление содержимого файла после текущего положения курсора
:! <i>команда</i>	выполнение команды
:set all	вывод всех опций сеанса редактирования
:set nu	включение опции нумерования строк
:set nonu	выключение опции нумерования строк

Другие команды ex (2)

:set <i>опция</i>	включение опции режима редактирования
:set <i>noопция</i>	выключение опции режима редактирования
:set all	вывод всех опций сеанса редактирования
: map	отображение на экране клавиш, используемых в командном режиме
:map!	отображение на экране клавиш, используемых в режиме ввода

Пример

:set showmod	включение вывода сообщения о режиме работы
:set noshowmod	выключение вывода сообщения о режиме работы