

05. Обработка текста

- 1 Просмотр текста
- 2 Редактирование текста
- 3 Основы написания скриптов

Раздел 1

Просмотр текста

Неинтерактивный просмотр и фильтрация

- **cat** - Конкатенация файлов (зачастую используется для вывода на экран)

```
cat [OPTION]... [FILE]...
```

Конкатенация - берет все переданные параметрами файлы и выводит на терминал

```
$ cat /etc/passwd
```

Вывод содержимого текстового файла на экран

```
$ cat file1 file2 file3 > bigfile
```

```
$ cat file1 file2 file3 | program
```

Удобна для перенаправления текста на другие команды для организации конвеерной обработки текста

head

Вывод начала файла

```
$ head /etc/passwd
```

- Отображают первые несколько (head) строчек текстового файла - по умолчанию 10

```
$ head -n 3 /etc/passwd
```

```
$ head -3 /etc/passwd
```

- Отображение указанного количества строчек файла

tail

Вывод конца файла

```
$ tail /etc/passwd
```

- Отображают последние несколько (tail) строчек текстового файла - по умолчанию 10

```
$ tail -f /var/log/messages
```

- Интерактивный режим - отображает последние строчки файла по мере их изменения - часто используется с файлами журнала

grep - General Regular Expression Print

Поиск подстроки (регулярного выражения) в тексте

```
$ grep bash /etc/passwd
```

- Фильтрация текстовых данных (файл, стандартный вход) по шаблону. Выдает все строки, где встретился указанный шаблон

- **-color** - подсвечивать вывод
- **-c** - количество найденных вхождений
- **-n** - отображать порядковый номер строчек

- **-v** - инвертировать поиск - строки, где не встретился шаблон
- **-i** - игнорировать регистр символов
- **-w** - искать только вхождение слов, целиком соответствующих шаблону

```
$ cat /etc/passwd | grep --color -n bash
```

cut

Вырезание текста по полям

- **-d** - разделитель, по умолчанию Tab
- **-f** - указание номеров столбцов
- **-c** - указание порядковых номеров символов

```
$ cat /etc/passwd | cut -d: -f1,5-7
```

```
$ ls -l | cut -c1-11,50-
```

Другие инструменты

- **sed** - Потоковый редактор текста
- **awk** - Табличный текстовый процессор

Утилита wc

Статистика текста (Word Count) - количество строк, слов, байт

- **-l** - только количество строк
- **-w** - только слов
- **-c** - только байт
- **-m** - количество символов

Интерактивное отображение текста

\$PAGER

- Переменная окружения, позволяющая определять используемое средство интерактивного отображения (т.е. пейджер)
- **more**
- **less**

\$ less /etc/passwd

Раздел 2

Редактирование текста

Редакторы текста

- **vi/vim** - Очень мощный, но кривая кривая обучения очень крута на старте
- **emacs** - Еще более мощный, крутая кривая обучения, избыточен для задач системного администратора
- **joe** - По проще в освоении, чем vi, но так же не сильно интуитивен
- **nano** - Золотая середина, просто в освоении, достаточно для простых правок текста, типа редактирования скриптов, конфигурационных файлов и т.п.
- **mcedit** - Встроенный редактор консольного файл-менеджера **mc** - наиболее близок к редактированию текста а-ля из **Norton Commander/Far**

Использование nano

```
$ nano file.txt
```

- Команды редактора вводятся вместе с клавишей **Ctrl**
Сводка по командам - в нижней части экрана

Команды	Описание
[^] O	Сохранить, запросит подтверждение файла
[^] X	Выйти
[^] W	Поиск
[^] W, [^] R	Поиск и замена
[^] Y	страница вверх
[^] V	страница вниз
[^] G	справка по командам

Редакторы vi/vim

```
$ vim file.txt
```

- **командный** режим - ввод команд - режим по-умолчанию, клавиша **i** - переход в режим редактирования
- **режим редактирования** - правка текста
- подрежимы вставки и замены - переключение между ними клавиша **Ins**,
- выход в командный режим - клавиша **Esc**, команды начинаются с двоеточия

Комбинации клавиш командного режима

- **:w** - сохранение отредактированного файла
- **:q** - выход
- **:wq** - выход с сохранением
- **:q!** - выход без сохранения

Справка по vim

- **:help <topic>** - в командном режиме редактора
- **vimtutor** - обучающий режим

Раздел 3

Основы написания скриптов

Общий синтаксис сценариев

```
$ cat hello.sh
#!/bin/sh
echo "Hello, world!"
```

- Первая строка - специальный комментарий, начинающийся с `#!` (называется *sha-bang*), указывающий, какой интерпретатор команд необходимо вызывать для интерпретации данного сценария.

Sha-bang

```
#!/bin/sh  
#!/bin/bash  
#!/usr/bin/perl  
#!/usr/bin/tcl  
#!/bin/sed -f  
#!/usr/awk -f
```

Создание файла сценария

```
$ touch bf.sh  
$ nano bf.sh
```

- Создаем и открываем на редактирование

```
#!/bin/sh
tar -cJf ~/backup.tar.xz ~/work
```

- Добавляем текст скрипта

Запуск сценариев

```
$ sh ./bf.sh
```

- ➊ Передаем скрипт соответствующему интерпретатору (**bash**)

```
$ chmod +x bf.sh  
$ ./bf.sh
```

- ② Делаем файл скрипта исполняемым командой **chmod** и запускаем непосредственно

```
$ mv bf.sh bin/  
$ PATH=$HOME/bin:$PATH  
$ bf.sh
```

- При необходимости добавляем путь к скрипту в переменную окружения PATH