

# Программирование на Bash: Полезные утилиты



Артем  
Палецкий



**Артем Палецкий**

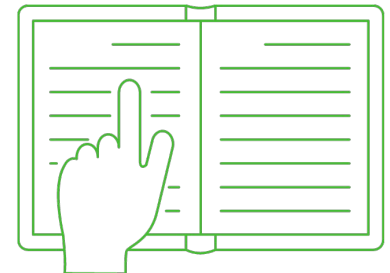
**Ведущий инженер, InfoWatch**

---

# Предисловие

На этом занятии мы рассмотрим основные **инструменты для работы с текстом и администрирования ОС Linux:**

- cat, more и less;
- head и tail;
- sort;
- split;
- wc;
- grep;
- locate;
- cut;
- find;
- vi/vim;
- nano.





# План занятия

1. [Предисловие](#)
2. [Работа с текстом](#)
3. [Текстовые редакторы](#)
4. [Итоги](#)
5. [Домашнее задание](#)



# Работа с текстом

---

## cat, more и less

**cat** (concatenate) — команда cat читает содержимое одного или нескольких файлов и копирует его в стандартный вывод.

```
# cat [файл...]
```

**more** и **less** — большой текстовый файл намного удобнее просматривать с помощью команд less или more.

Программа less удобнее, чем more, если она есть в вашей системе.

```
# cat /var/log/messages | less
```



---

## head и tail

**head** и **tail** — команда head выводит первые десять строк файла, а tail — последние десять.

Количество строк может регулироваться с помощью параметра **-n**.

```
# head -n 10 /var/log/messages
```

```
# tail -n 15 /var/log/messages
```



---

# sort

**sort** — сортирует все указанные файлы, результат сортировки всех указанных файлов и перенаправленного отправляется на стандартный вывод.

```
# sort [параметр]... [файлы]
```





# sort

Параметр	Описание
<b>-b</b>	Пробелы в начале сортируемых полей или начале ключей будут игнорироваться.
<b>-d</b>	При сортировке будут игнорироваться все символы, кроме букв, цифр и пробельных символов.
<b>-f</b>	Игнорировать регистр букв.
<b>-r</b>	Сортировка в обратном порядке.
<b>-o файл</b>	Вывод результатов сортировки в файл.
<b>-t символ</b>	Использование указанного символа в качестве разделителя полей.

---

# split

**split** — используется для разделения файлов на части.

По умолчанию создаются части размером в 1000 строк.

Изменить размер можно, указав количество строк, например:

```
# split -500 файл1
```



---

## WC

**WC** — подсчет слов в файле.

- для подсчета слов, строк и байт в текстовом файле:

```
# wc /var/log/messages
```

- для подсчета строк в текстовом файле:

```
# wc -l /var/log/messages
```

- для подсчета байт в текстовом файле:

```
# wc -c /var/log/messages
```



# locate

**locate** — простой способ поиска файлов

Программа locate выполняет быстрый поиск в базе данных имен файлов и выводит все имена, соответствующие искомой строке.

Допустим, например, что нужно **найти все программы с именами, начинающимися с zip:**

```
# locate zip | grep bin
```

```
/bin/bunzip2
```

```
/bin/bzip2
```

```
/bin/bzip2recover
```

```
/bin/gunzip
```

```
/bin/gzip
```

```
/usr/bin/funzip
```

```
/usr/bin/gpg-zip
```

---

# cut

**cut** — удаление фрагментов из всех строк в файлах.

cut используется для извлечения фрагментов текста из строк и вывода их в стандартный вывод.

Она может принимать имена файлов в аргументах или данные со стандартного ввода.



# cut

## Параметры команды cut для выбора фрагментов:

Параметр	Описание
<b>-c</b>	<b>список символов: --characters=</b>  Извлекает фрагмент строки, определяемый списком символов. Список может включать один или несколько числовых диапазонов, разделенных запятыми.
<b>-f</b>	<b>список полей: --fields=</b>  Извлекает одно или несколько полей из строки, как определено аргументом список символов. Список может включать одно или несколько полей или диапазонов полей, разделенных запятыми.

# cut

## Параметры команды cut для выбора фрагментов:

Параметр	Описание
<b>-d</b>	<p>символ разделитель: <code>--delimiter=</code></p> <p>В присутствии параметра <code>-f</code>, в качестве разделителя полей используется символ_разделитель.</p> <p>По умолчанию поля должны отделяться друг от друга одним символом табуляции <code>--complement</code>.</p> <p>Извлекает строку текста целиком, кроме фрагментов, определяемых параметром <code>-c</code> и/или <code>-f</code>.</p>

## cut

Если воспользоваться программой `cat` с параметром `-A`, можно увидеть, отвечает ли файл требованию в отношении использования символа табуляции в качестве разделителя полей.

```
# cat -A distros.txt
```

```
Debian^I10.5^I01/08/2020$
```

```
Ubuntu^I20.04^I23/04/2020$
```

```
RHEL^I8.3^I03/11/2020^I$
```

```
CentOS^I8.3^I07/12/2020$
```

```
Fedora^I33^I10/27/2007$
```

```
openSUSE^I15.3^I10/04/2020$
```



---

## cut

С помощью cut мы можем извлечь третий столбец:

```
# cut -f 3 distros.txt
```

```
01/08/2020
```

```
23/04/2020
```

```
03/11/2020
```

```
07/12/2020
```

```
10/27/2007
```

```
10/04/2020
```

---

# find

**find** — сложный способ поиска файлов.

В отличие от программы **locate**, выполняющей поиск файлов по именам, программа **find** ищет файлы согласно заданным атрибутам в указанном каталоге и во вложенных подкаталогах.

В простейшем случае программе **find** можно передать одно или несколько имен каталогов для поиска.

Например, с ее помощью можно получить список содержимого домашнего каталога:

```
# find ~
```



---

# find

Для большинства активных пользователей она выдаст длинный список. Так как список выводится в стандартный вывод, его можно передать по конвейеру другим программам.

Воспользуемся программой `wc` , чтобы подсчитать число файлов:

```
# find ~ | wc -l
```

```
363168
```

---

# find

Допустим, мы хотим получить список каталогов. Для этого добавим в команду следующую проверку:

```
# find ~ -type d | wc -l
```

```
33440
```

Добавив проверку `-type d`, мы ограничились поиском только каталогов. Но точно так же можно ограничить поиск только обычными файлами:

```
# find ~ -type f | wc -l
```

```
322504
```

---

# find

Параметр	Описание
<b>b</b>	Специальный файл блочного устройства.
<b>c</b>	Специальный файл символьного устройства.
<b>d</b>	Каталог.
<b>f</b>	Обычный файл.
<b>l</b>	Символическая ссылка.

---

## find

Добавив дополнительные проверки, можно выполнять поиск файлов по размеру и имени.

Давайте найдем все обычные файлы с именами, соответствующими шаблону \*.JPG, и имеющие размер больше 1 мегабайта:

```
# find ~ -type f -name "*.JPG" -size +1M | wc -l
```

```
15
```

---

# find

## Единицы измерения, поддерживаемые командой find:

<b>b</b>	Блоки размером по 512 байт (используется по умолчанию, если иное не указано явно).
<b>c</b>	Байты.
<b>w</b>	2-байтные слова.
<b>k</b>	Килобайты (Kilobytes, блоки по 1024 байт).
<b>M</b>	Мегабайты (Megabytes, блоки по 1 048 576 байт).
<b>G</b>	Гигабайты (Gigabytes, блоки по 1 073 741 824 байт).



# Текстовые редакторы



---

# vi/vim

**vi/vim** — текстовый редактор.

**Редактор vi может работать в трех режимах:**

- **основной** (визуальный) **режим** — в нем и осуществляется редактирование текста;
- **командный режим** — в нем выполняется ввод специальных команд для работы с текстом (если сравнивать vi с нормальным редактором, то этот режим ассоциируется с меню редактора, где есть команды вроде «сохранить», «выйти» и т. д.);
- **режим просмотра** — предназначен только для просмотра файла (если надумаете использовать этот режим, лучше вспомните про команду less ).

---

## vi/vim

После запуска редактора вы можете переключать режимы (как — будет сказано позже), но выбрать режим можно и при запуске редактора:

```
# vi файл
```

```
# vi -e файл
```

```
# vi -R файл
```

## Команды vi/vim

<b>:q!</b>	Выход без сохранения.
<b>:w</b>	Сохранить изменения.
<b>:w &lt;файл&gt;</b>	Сохранить изменения под именем <файл>.
<b>:wq</b>	Сохранить и выйти.
<b>:q</b>	Выйти, если нет изменений.
<b>i</b>	Перейти в режим вставки символов в позицию курсора.
<b>a</b>	Перейти в режим вставки символов в позицию после курсора.



## Команды vi/vim

<b>o</b>	Вставить строку после текущей.
<b>O</b>	Вставить строку над текущей.
<b>x</b>	Удалить символ в позицию курсора.
<b>dd</b>	Удалить текущую строку.
<b>u</b>	Отменить последнее действие.

---

# nano

**nano** — текстовый редактор

nano запускается аналогично:

```
# nano <имя файла>
```





# Итоги

---

## Итоги

Сегодня мы познакомились с базовыми командами и текстовыми редакторами в ОС Linux.





# Домашнее задание



---

## Домашнее задание

Давайте посмотрим ваше [домашнее задание](#).

Настоятельно рекомендуем вам выполнять ДЗ в том же ритме, что и просмотр лекций.

- Вопросы по домашней работе задавайте **в чате** мессенджера Slack.
- Задачи можно сдавать **по частям**.
- Зачёт по домашней работе проставляется после того, как **приняты все задачи**.

**Задавайте вопросы и  
пишите отзыв о лекции!**

**Артем Палецкий**