



Введение в `bash`. Работа с командной строкой

Операции с файлами. Копирование и перемещение файлов.

Редактирование файлов в текстовом редакторе `nano`.

Удаленное копирование: команды `scp` и `sftp`.

25 сентября 2025 г.

Иван Ильницкий
ФББ МГУ

Специальные символы. # - комментарии

В Bash символ # используется для обозначения начала комментария. Все, что следует за #, игнорируется интерпретатором Bash до конца строки. Это позволяет добавлять пояснения или временно отключать части кода.

```
echo Это не комментарий # Это комментарий
```

Специальные символы. Экранирование

Внутри двойных кавычек “ ” большинство специальных символов теряют своё значение, но есть исключения: \$, \ .

Одинарные кавычки ‘ ’ предотвращают интерпретацию всех специальных символов внутри них, кроме самой одинарной кавычки

Обратный слэш \ позволяет экранировать отдельные символы

Специальные символы. Шаблоны подстановки имен

- ***** — для замены нескольких символов (в том числе 0);
- **?** — для замены одиночного символа;
- **[]** — для замены определенного набора символов.

❖ Примеры

`ls *.txt` — все файлы с расширением .txt в текущей директории

`ls doc*.pdf` — все PDF-файлы, начинающиеся с "doc"

`ls report_?.docx` — найдет файлы типа report_1.docx, report_A.docx

`ls img_???.jpg` — найдет изображения с трехзначными номерами, например img_001.jpg, img_999.jpg

`ls file[123].txt` — найдет file1.txt, file2.txt и file3.txt.

Символ в шаблоне	Назначение	Пример шаблона	Пример подходящих файлов
*	Обозначает <i>любое</i> количество <i>любых</i> символов, в том числе и их отсутствие	*.txt	file.txt news.1.2.txt
		my*.png	my_dog.png my_bike.png
		human	grch38-human.fa 1human1 genome.human
?	Один любой символ	otchet.???	otchet.txt otchet.pdf
		k?t.jpg	kot.jpg kit.jpg
[]	Любой символ из перечня или диапазона	read_[12].fasta	read_1.fasta read_2.fasta
		[m-p]ap.*	map.txt nap.csv pap.R

Создание и редактирование файлов. Программа *nano*

nano — это командный текстовый редактор для Unix-подобных систем. Установлен по умолчанию на многих дистрибутивах Linux.

Использование: `nano [файл]` — открыть файл или создать новый.

Основные горячие клавиши:

`CTRL + O` — сохранить изменения;

`CTRL + X` — выйти из редактора;

`CTRL + G` — посмотреть справочную информацию о *nano*;

`CTRL + K` — вырезать строку;

`CTRL + U` — вставить строку;

`CTRL + W` — поиск текста.

Создание и редактирование файлов. Программа *nano*

Подсказка: Обычно, когда вы записали ваш текст в *nano* нужно последовательно нажать сочетания клавиш

1. **Ctrl + O**
2. **Enter**
3. **Ctrl + X**

Передача файлов с помощью SCP

- ❖ scp (secure copy) используется для безопасной передачи файлов между устройствами.
- ❖ `scp [опции] источник назначение`

Если нужно скачать с удаленного сервера, то сначала указываем путь на нем в следующем виде:

`scp имя_пользователя@имя_хоста:путь локальный_путь`

и наоборот:

`scp локальный_путь имя_пользователя@имя_хоста:путь`

- ❖ Опции:
 - P (порт) – в отличие от ssh используется заглавная буква
 - r – рекурсивное скачивание, когда нужно скачать директорию с ее содержимым
- ❖ Примеры:
 - `scp my_file.txt user@remote_host:/path/remote_dir`
 - `scp -P 8000 directory user@remote_host:/path/remote_dir`

Передача файлов с помощью SFTP

Подключение к удаленному серверу осуществляется схожим с ssh образом:

```
sftp имя_пользователя@имя_хоста
```

Основные команды SFTP после подключения

`put [локальный файл]` – загрузка файла на сервер.

`get [файл с сервера]` – скачивание файла с сервера.

`cd [директория]` – смена текущей директории на сервере.

`ls` – просмотр содержимого текущей директории на сервере.

`pwd` – показать текущую директорию на сервере.

`lls` – просмотр содержимого текущей локальной директории.

`lpwd` – показать текущую локальную директорию.

Команда grep

- ❖ Команда **grep** используется для поиска строк, соответствующих заданному шаблону, в текстовых файлах или стандартном вводе. Она полезна для фильтрации данных и быстрого нахождения нужной информации.

grep [опции] "шаблон" файл

- ❖ Основные опции:

-i – Игнорирует регистр при поиске. Например, `grep -i "example" file.txt` найдет строки с "example", "Example", "EXAMPLE" и т.д.

-v – Выводит строки, которые не содержат заданный паттерн. Например, `grep -v "text" file.txt` покажет все строки, кроме тех, что содержат "text".

-r – Рекурсивный поиск в директориях. Например, `grep -r "search" /path/to/directory/` будет искать во всех файлах в указанной директории и ее поддиректориях.

-n – Показывает номер строки перед выводом строки, содержащей соответствие.

-c – Подсчитывает количество найденных совпадений.

Команда grep: примеры с опциями

- ❖ Команда **grep** используется для поиска строк, соответствующих заданному шаблону, в текстовых файлах или стандартном вводе. Она полезна для фильтрации данных и быстрого нахождения нужной информации.
- ❖ Поиск с использованием нескольких файлов
grep -r "fox" path/to/dir
- ❖ Флаги -A и -B позволяют выводить строки до и после совпадения
grep -A 2 -B 2 "pattern" example.txt
- ❖ Поиск по всему слову
grep -w "a" example.txt
- ❖ Вывод только тех частей строк, которые соответствуют заданному шаблону
grep -o "fox" example.txt

Команда grep

Регулярные выражения и специальные символы

- ❖ Символ точки в регулярных выражениях grep соответствует любому одиночному символу, кроме символа новой строки.

```
grep "c.t" example.txt
```

- ❖ Символ звездочка соответствует нулю или более вхождений предыдущего символа

```
grep "ca*t" найдет ct, cat, caaat, ...
```

- ❖ Поиск строк, начинающихся с определенного слова

```
grep "^The" example.txt
```

- ❖ Поиск строк, заканчивающихся на определенное слово

```
grep "dog$" example.txt
```

```
grep "dog\.$" example.txt
```

– если в шаблоне/паттерне есть спецсимвол, его надо экранировать

- ❖ Поиск слов, начинающихся или заканчивающихся на определенные символы

```
grep "\bte" example.txt
```

```
grep "st\b" example.txt
```

Команда grep (более сложные примеры)

Регулярные выражения и специальные символы

- ❖ Поиск строк, содержащих цифры

```
grep "[0-9]" example.txt
```

```
grep "[A-Z]" example.txt
```

- ❖ Поиск с использованием `grep -e "cat" -e "dog" file.txt` найдет строки, содержащие либо "cat", либо "dog".

Специальные символы. Фигурные скобки

В Bash можно использовать фигурные скобки `{ }` для генерации различных последовательностей. Это позволяет быстро создавать наборы строк или чисел.

Подстановка фигурных скобок не проверяет наличия файлов

Примеры использования:

`touch file_{a..c}.txt` – генерирует имена файлов
Создаст файлы: `file_a.txt`, `file_b.txt`, `file_c.txt`

`echo {a..g}` – выводит от `a` до `g`
Вывод: `a b c d e f g`

`echo {a,b}{1..3}` – выводит все комбинации
Вывод: `a1 a2 a3 b1 b2 b3`

Загрузка файлов из интернета: *wget*

wget:

Основная ориентированность на скачивание файлов.

Поддерживает рекурсивное скачивание (можно скачивать целые сайты с директориями).

- ❖ `wget [опции] [ссылка]`
- ❖ Опции: `-c` (продолжить загрузку), `-P` (указать директорию для скачивания), `-r` (рекурсивное скачивание), `-O` (задать имя скачанному файлу)
- ❖ Примеры:
`wget http://example.com/file.txt`
`wget -r http://example.com/`
`wget -P directory http://example.com/`
`wget -O arch.zip http://example.com/archive2024.zip`

Операции с файлами

- `less [файл]` – просмотр файла с возможностью листать файл
- `cat [файл]` – полностью выводит содержимое файла в стандартный вывод
- `head [файл]` – выводит первые несколько строк файла
- `tail [файл]` – выводит последние несколько строк файла

❖ Примеры:

```
less documents/myscript.sh  
cat file.txt  
tail -n 3 file.txt
```

“q” – ВЫЙТИ ИЗ `less`



Команда WC

- ❖ Используется для подсчета количества строк, слов, символов или байт в файле/файлах

`wc [опции] [файл]`

Опции: `-m` (количество символов), `-l` (количество строк), `-w` (количество слов), `-c` (количество байт)

Создание и удаление директорий `mkdir`, `rm`, `rmdir`

`mkdir [опции] [директория]` — создает новую директорию

Опции: `-p` (создать вложенные директории)

```
mkdir new_folder
```

```
mkdir folder1 folder2 folder3
```

```
mkdir -p parent/child/grandchild
```

`rmdir [опции] [директория]` — удаляет директорию

```
rmdir new_folder
```

`rm [опции] [файл/директория]` — удаляет файлы и директории (`-r`)

Опции: `-r` (рекурсивно удаляет директории с их содержимым)

```
rm ~/file
```

```
rm -r folder
```

```
rm alpha1 alpha2 alpha3 = rm alpha*
```

Удаление с помощью `rm` необратимо, особенно аккуратно стоит быть с флагом `-f`, который подавляет запросы на подтверждение операции

Копирование и перемещение файлов: mv, cp

Опции:

-r (рекурсивное копирование/перемещение директорий)

Примеры:

mv:

`mv file1.txt file2.txt` (переименование)

`mv file.txt /home/user/docs/` (перемещение)

`mv dir1 dir2` (перемещение/переименование директории)

cp:

`cp file.txt file_copy.txt` (копирование файла)

`cp -r dir1 /home/user/docs/` (рекурсивное копирование директории)

`cp file1.txt file2.txt /home/user/backup/` (копирование нескольких файлов)

Горячие клавиши в Bash

- ❖ **Tab** – автозаполнение
- ❖ **Ctrl + C** – прервать выполнение текущей команды
- ❖ **Ctrl + Z** – приостановить выполнение текущей команды
- ❖ **Ctrl + D** – выйти из терминала/ закрыть сессию
- ❖ **Ctrl + A** – переместиться в начало строки
- ❖ **Ctrl + E** – переместиться в конец строки
- ❖ **Ctrl + W** – удалить одно слово слева
- ❖ **Ctrl + R** – поиск по истории команд

Полезные советы для повышения эффективности:

- ❖ Используйте автозаполнение с помощью клавиши *Tab*, чтобы сэкономить время на наборе текста.
- ❖ Горячие клавиши для перемещения в командной строке.
- ❖ Не ленитесь читать мануалы (команда `man`) для получения детальной информации о командах.