

# Семинар #4: Linux. Потоки, перенаправление. Практика.

## Задача 1. Простые команды

- a. **Привет мир**  
Введите команду, которая будет печатать на экран "Hello world". Используйте команду `echo`.
- b. **Привет мир 2**  
Введите команду, которая будет печатать "Hello world". Используйте команду `printf`. Вывод должен быть аналогичен предыдущей команде, то есть в конце строки нужно печатать перевод на новую строку.
- c. **Where am I?**  
Введите команду, которая будет печатать на экран текущую директорию, в которой вы находитесь.
- d. **Who am I?**  
Введите команду, которая будет печатать на экран ваше имя пользователя.
- e. **Время и дата**  
Введите команду, которая будет печатать текущие время и дату.
- f. **Календарь**  
Введите команду, которая будет печатать календарь текущего месяца.
- g. **Информация о системе**  
Введите команду, которая будет печатать полную информацию о системе. Используйте команду `uname -a`.
- h. **cat**  
Выполните команду, которая печатает на экран содержимое файла `happiness.txt`.
- i. **tac**  
Выполните команду, которая печатает на экран строки файла `happiness.txt` в обратном порядке.
- j. **Начало**  
Выполните команду, которая печатает на экран первые 5 строк файла `happiness.txt`.
- k. **Конец**  
Выполните команду, которая печатает на экран последние 5 строк файла `happiness.txt`.
- l. **Строки, слова и символы**  
Используйте программу `wc`, чтобы найти количество строк, слов и символов в файле `happiness.txt`. Что делают опции `-l`, `-w` и `-c` этой команды?
- m. **less**  
Используйте команду `less` для просмотра файла `invisible_man.txt`.  
Стрелочки вверх/вниз, `PageUp/PageDown` – перемещение, `g/G` – начало/конец файла, `q` – выйти из `less`.
- n. **Опции ls**
  - (a) Введите команду `ls`, чтобы напечатать файлы в текущей директории.
  - (b) Введите команду `ls -l`, чтобы напечатать файлы в текущей директории. Один файл на строку.
  - (c) Введите команду `ls -a`, чтобы напечатать все файлы в текущей директории, в том числе скрытые.
  - (d) Введите команду `ls -l`. Подробная информация о файлах в текущей директории.
  - (e) Введите команду, которая бы печатала все файлы подробно, но так, чтобы они были отсортированы по их размеру.
  - (f) Введите команду, которая бы печатала все файлы подробно, но так, чтобы они были отсортированы по дате последнего изменения.
  - (g) Введите команду, которая бы печатала все файлы подробно, но так, чтобы они были отсортированы по их расширению.
  - (h) Введите команду, которая бы печатала все файлы текущей директории, а потом все файлы поддиректорий и так далее.

## Задача 2. Простое перенаправление в файл

### a. Запись в файл

Введите команду, которая будет создавать файл `2a.txt` и записывать туда строку `"Sapere Aude"`. Если файл с таким именем уже существует, то сначала всё его содержимое должно удаляться, а потом уже производиться запись.

### b. Запись времени в файл

Введите команду, которая будет создавать файл `2b.txt` и записывать туда текущее время. Если файл с таким именем уже существует, то сначала всё его содержимое должно удаляться, а потом уже производиться запись. Текущее время можно напечатать командой:

```
date +%H:%M:%S
```

### c. Дозапись времени в файл

Введите команду, которая будет создавать файл `2c.txt` и записывать туда текущее время. Если файл с таким именем уже существует, время должно записываться в новую строку в конец файла.

### d. Сортировка строк

Используйте программу `sort`, чтобы отсортировать все строки в файле `names.txt`. Сохраните отсортированные имена в файле `2d.txt`.

### e. Обращение каждой строки

Используйте программу `rev`, чтобы обратить каждую строку в файле `names.txt`. Сохраните обращённый файл в имена в файле `2e.txt`.

### f. Конкатенация

Используйте программу `cat`, чтобы конкатенировать два файла: `dream.txt` и `happiness.txt`. Результат конкатенации запишите в файл `2f.txt`.

### g. Две строки

Напишите команду (одну строку), которая будет печатать в файл `2g.txt` две строки:

```
first
second
```

## Задача 3. Синтаксис `$(...)`

### a. Ссылка

В файле `ref.txt` хранится название некоторого файла. Вам нужно выполнить команду, которая читает файл `ref.txt`, считывает оттуда название другого файла и печатает содержимое этого файла на экран.

### b. Привет user

Введите команду, которая будет создавать файл `3b.txt` и записывать туда строку `"Hello <user>"`, где вместо `<user>` будет записано ваше имя пользователя. Решите эту задачу четырьмя способами:

- i. С использованием опции `-e` команды `echo`.
- ii. С использованием команды `printf`.
- iii. С использованием синтаксиса `$(...)`.
- iv. С использованием опции `-n` команды `echo` и объединения нескольких команд.

### c. `alice.txt`

Выполните команду (одна строка), которая бы создавала файл по имени `<user>-<seconds>.txt`, где вместо `<user>` команда должна подставить имя пользователя, а вместо `<seconds>` команда должна подставить количество секунд, прошедших с 1 января 1970 года (команда `date +%s`). Команда должна записывать в новый файл календарь за текущий месяц.

## Задача 4. Стандартный вход

### a. `cat` из `stdin` в `stdout`

Запустите программу `cat`, не указывая файл. В результате `cat` будет требовать ввести текст через стандартный вход. После ввода строки и нажатия `Enter`, `cat` будет печатать на экран строку, которую вы ввели. Введите строки:

```
one
two
three
```

После каждой строки `cat` будет печатать её же в стандартный выход `stdout`. Для завершения ввода введите `Ctrl-D`.

### b. `cat` из `stdin` в файл

Запустите программу `cat` не указывая названия файла как аргумента, а просто перенаправив вывод в файл:

```
$ cat > 3b.txt
```

В результате `cat` будет требовать ввести текст через стандартный вход (`stdin`). Введите строки:

```
one
two
three
```

и нажмите `Ctrl-D`. Посмотрите на содержимое файла `3b.txt`.

### c. Сортировка строк из `stdin`

Запустите программу `sort`, не указав файл в качестве аргумента. В результате `sort` будет требовать ввести текст через стандартный вход. Введите:

```
one
two
three
four
```

и нажмите `Ctrl-D`. Эти строки должны напечататься в алфавитном порядке.

### d. Программы, которые читают из `stdin`

Протестируйте, какие из следующих программ читают из `stdin`, если им не передать аргументы, а какие не читают:

```
echo ls wc cp mkdir touch tac head tail rev
```

### e. Аргумент или перенаправление

Что делают следующие команды и чем они отличаются?

```
$ cat dream.txt
$ cat < dream.txt
$ wc dream.txt
$ wc < dream.txt
```

### f. Перенаправление из файла в `stdin`

Введите команду `bash` (одну строку), которая будет вычислять и печатать на экран количество строк файла `invisible_man.txt` в следующем формате:

```
file has <number> lines
```

где вместо `<number>` будет поставляться количество строк.

## Задача 5. Pipe (из stdout одной программы в stdin другой)

### a. Полное обращение

Выполните команду (одна строка bash), которая делает следующее:

- Сортирует строки файла `names.txt`.
- Обращает полученные отсортированные строки. Первая строка должна стать последней, вторая – предпоследней и так далее.
- Обращает каждую строку этого же файла.
- Сохраняет результат в файл `totally_reversed.txt`.

Первые строки результирующего файла должны иметь следующий вид:

```
urahcaZ
araY
reivaX
...
```

Используйте программы `sort`, `tac` и `rev`.

### b. Удаление дубликатов

Файл `logs.txt` содержит список сайтов, при этом некоторые строки повторяются. Необходимо выполнить команду (одна строка), которая:

- Отбирает только уникальные строки.
- Сортирует их в алфавитном порядке.
- Сохраняет результат в файл `uniq_logs.txt`.

В итоговом файле должно остаться ровно 20 строк.

### c. Количество файлов

Выполните команду, которая будет печатать количество файлов в текущей директории.

### d. Просмотр файлов

Напишите команду, для просмотра подробной информации (`ls -l`) о всех файлах из директории `/usr/bin` с использованием команды `less`.

### e. Самые часто посещаемые сайты

Введите команду (одна строка), которая будет печатать топ-5 самых часто встречающихся строк в `logs.txt`.

## Задача 6. Перенаправление из stderr

Файл `printerr.py` – это python скрипт, который печатает одну строку в stdout и одну строку в stderr. Если запустить этот скрипт:

```
$ python printerr.py
```

то оба сообщения напечатаются на экран, так как по умолчанию stdout и stderr выводятся на экран терминала.

### a. Перенаправление в файлы

Выполните команду, которая будет перенаправлять поток stdout в файл `out.txt`, в поток stderr в файл `err.txt`.

### b. Перенаправление в /dev/null

Выполните команду, которая будет печатать на экран только поток stderr, а поток stdout игнорировать.

### c. Перенаправление в один файл

Выполните команду, которая будет перенаправлять оба потока в файл `all.txt`.

### d. Pipe stderr

По умолчанию pipe работает с stdout. Выполните команду, которая будет перенаправлять поток stderr на вход программе `rev`. Поток stdout должен игнорироваться. В результате на экране должно быть напечатано:

```
.rredts ot seog sihT
```

## Задача 7. Wildcards

В директории `wild` содержится набор различных файлов. Выполните команды, которые делают следующее:

- a. Печатает на экран все файлы (то есть имена всех файлов).
- b. Печатает все файлы, с именами, начинающимися на `article`.
- c. Печатает все файлы, с расширением `.txt`.
- d. Печатает все файлы, с именем в формате `article?.txt`, где вместо вопроса расположена цифра.
- e. Печатает все файлы, с именем в формате `article?.txt`, где вместо вопроса расположена заглавная буква.
- f. Печатает все файлы, начинающиеся на `backup` и имеющие в своём названии дату, соответствующую октябрю 2024-го года.
- g. Печатает все файлы, начинающиеся на `backup` и имеющие в своём названии дату, соответствующую 2023-му или 2025-му году.
- h. Печатает все файлы, начинающиеся на букву `a` или на букву `c`.
- i. Печатает все файлы, имена которых содержат хотя бы один символ подчёркивания.
- j. Печатает все файлы, имена которых содержат хотя бы одну цифру.
- k. Удаляет все файлы, заканчивающиеся на `.bin`.
- l. Создаёт папку `logs` и копирует туда все файлы, заканчивающиеся на `.log`.
- m. Создаёт файл `filenames.txt` и записывает туда все файлы, которые начинаются на `a` или на `c`.

## Задача 8. Brace expansion

- a. Создайте папку `brace` и создайте файлы с именами:

```
hello.txt hello.log hello.xml hello.md hello.json hello.cfg
```

Создать все файлы нужно, используя только одну строку. Сделайте эту строку как можно короче, используя `brace expansion`.

- b. Перейдите в папку `brace` и создайте файлы с именами:

```
alice1.txt alice2.txt alice3.txt bob1.txt bob2.txt bob3.txt casper1.txt casper2.txt casper3.txt
```

Создать все файлы нужно, используя только одну строку. Сделайте эту строку как можно короче, используя `brace expansion`.

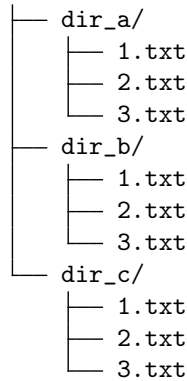
- c. Выполните команду, которая бы печатала на экран все числа от 1 до 100 с шагом 5. Используйте `brace expansion`.

- d. Перейдите в папку `brace` и создайте файлы с именами:

```
file001.txt file002.txt file003.txt ... file100.txt
```

Создать все файлы нужно, используя только одну строку. Сделайте эту строку как можно короче, используя `brace expansion`.

- e. Создайте следующую структуру файлов в папке `brace`:



Создать все файлы нужно, используя только одну строку. Сделайте эту строку как можно короче, используя brace expansion.

- f. Перейдите в папку `wild` и напечатайте имена всех файлов с расширениями `.jpg` или `.png`,

## Задача 9. `find`

В этой задаче, под термином "все файлы в директории" подразумевается все файлы в этой директории плюс все файлы в поддиректориях плюс все файлы в поддиректориях поддиректорий и так далее. Если не сказано иначе, при выполнении команд данной задачи вывод в `stderr` нужно игнорировать и не отображать его на экран.

a. **Печать всех файлов**

Перейдите в директорию `files`. Напечатайте все файлы в текущей директории, используя программу `find`.

b. **Печать файлов папки `/etc`**

Выполните команду, которая бы печатала все файлы внутри папки `/etc`.

c. **Поиск файла**

Напечатайте все файлы по имени `README.txt` в директории `/usr`.

d. **Поиск файлов с определённым расширением**

Напечатайте все файлы с расширением `.sh` (скрипты) в директории `/usr`.

e. **Поиск директорий**

Напечатайте все директории, внутри папки `/opt`. Все ошибки, возникшие при выполнении этой команды перенаправьте в файл `errors.log`.

f. **Поиск больших файлов**

Напечатайте рекурсивно все файлы внутри папки `/usr`, чей размер составляет больше 50-ти мегабайт.

g. **Поиск и копирование**

Создайте папку `scripts`. Скопируйте в эту папку все файлы с расширением `.sh` из директории `/usr`.

h. **Поиск и отображение подробной информации**

Найти все файлы с расширением `.conf` в директории `/etc` и вывести подробную информацию о каждом из них с помощью команды `ls -l`.

## Задача 10. `xargs` (из `stdin` в аргументы другой программы)

- a. **Программа `xargs`** Перейдите в папку `files` и выполните команду:

```
$ xargs cp
```

В результате команда будет требовать ввести текст через стандартный вход. Введите строку:

```
dream.txt copy.txt
```

и нажмите `Ctrl-D`. Программа `xargs` считала строку из стандартного входа и сама вызовет программу `cp` передав ей строку `"dream.txt copy.txt"` в качестве аргументов. Затем программа `cp` скопирует файл `dream.txt` в файл `copy.txt`.

- b. **Программа xargs вместе с pipe** Перейдите в папку `files` и выполните команду:

```
$ echo "dream.txt copy2.txt" | xargs cp
```

В результате выполнения этой команды файл `dream.txt` должен быть скопирован в файл `copy2.txt`.

- c. **Количество строк**

Выполните команду, которая бы считала количество строк во всех файлах папки `/etc` и рекурсивно всех файлах всех поддиректорий.

- d. **Массовое изменение прав**

Выполните команду, которая будет забирать права на выполнение у все скриптов из папки `scripts`. Проверьте результат с помощью `ls -l`. Выполните команду, которая будет давать права на выполнение у все скриптов из папки `scripts`.

- e. **Подробная информация об определённых файлах**

Найдите все файлы в папке `/etc`, в названии которых встречается дефис и выведете подробную информацию о всех этих файлах. Для просмотра используйте `less`.

- f. **Самые большие скрипты**

Введите команду, которая будет печатать топ-5 самых больших файлов с расширением `.sh` в директории `/usr`.

## Задача 11. `grep`

- Использование дефиса (-)
- **base64/shasum**