

# Основы терминала



АДИЛЕТ АСАНКОЖОЕВ



**Адилет Асанкожоев**

Python-разработчик в Makers.kg



# План занятия

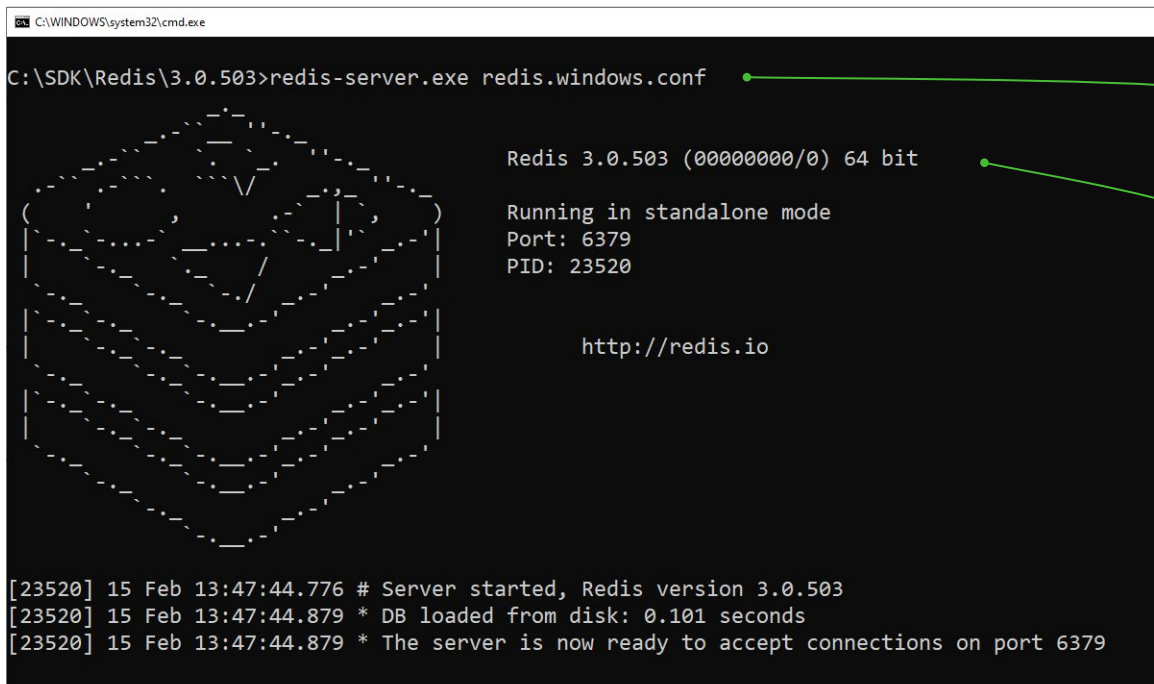
1. [Зачем нужен терминал?](#)
2. [Виды терминалов](#)
3. [Команды и ключи](#)
4. [Перенаправление потоков ввода/вывода](#)
5. [Переменные окружения](#)
6. [Пакеты](#)
7. [Итоги](#)
8. [Домашнее задание](#)



# Зачем нужен терминал?

# Что такое терминал?

**Терминал** (командная строка) — это программа, которая позволяет управлять компьютером посредством текстовых команд.



```
C:\WINDOWS\system32\cmd.exe

C:\SDK\Redis\3.0.503>redis-server.exe redis.windows.conf

Redis 3.0.503 (00000000/0) 64 bit

Running in standalone mode
Port: 6379
PID: 23520

http://redis.io

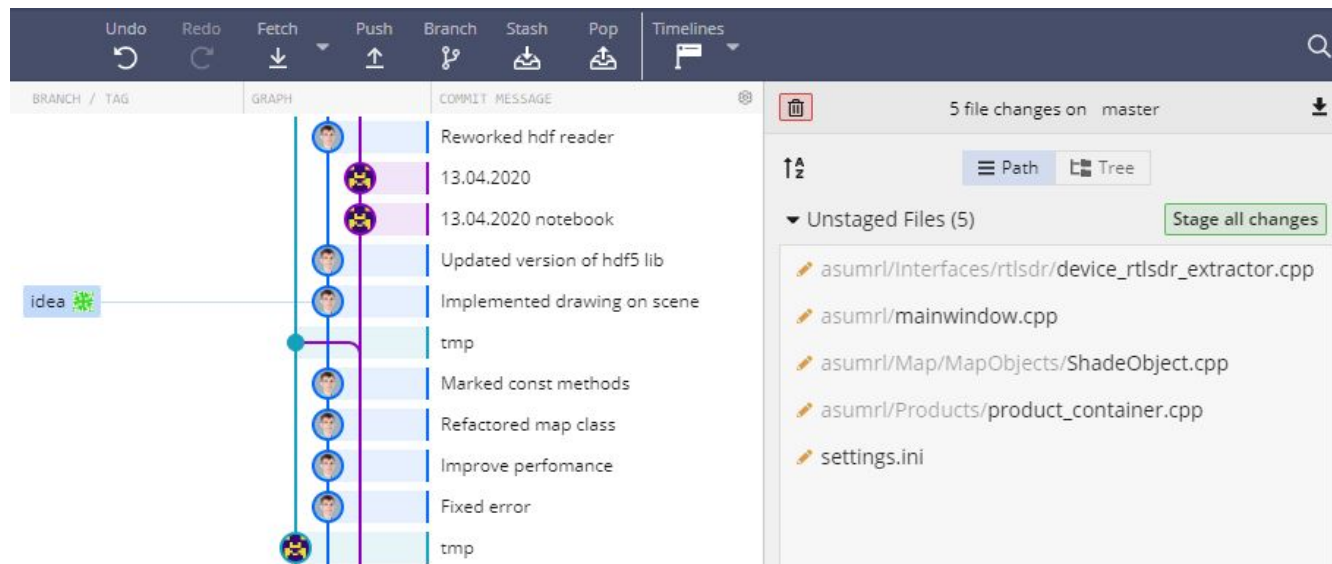
[23520] 15 Feb 13:47:44.776 # Server started, Redis version 3.0.503
[23520] 15 Feb 13:47:44.879 * DB loaded from disk: 0.101 seconds
[23520] 15 Feb 13:47:44.879 * The server is now ready to accept connections on port 6379
```

текстовая команда

результат

# Что такое терминал?

Но зачем терминал, если есть программы с красивым **графическим интерфейсом**?



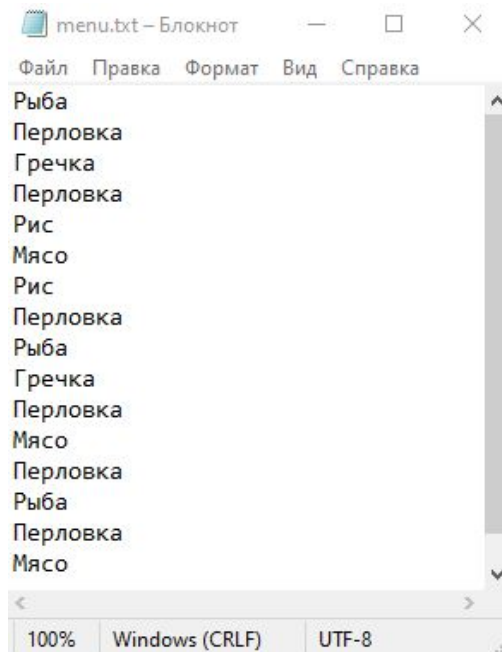
---

# Зачем нужен терминал?

- Некоторые действия можно сделать **только из командной строки**, так как графический интерфейс предоставляет не весь функционал;
- Серверы, на которых вы будете размещать свои проекты, чаще всего далеко от вас и доступ к ним **только через командную строку**;
- Мощь конвейера :)

# Задача для подогрева интереса

У нас есть **меню** (файл menu.txt) с повторяющимися строками:



Необходимо выяснить, **сколько** каких блюд заказали и **отсортировать** по убыванию.



# Решение в терминале

Для решения в командной строке мы воспользуемся следующей командой:

```
cat menu.txt | sort | uniq -c | sort -r
```

## Комментарий

Результатом этой текстовой команды будет:

7 Перловка

4 Мясо

4 Гречка

3 Рыба

2 Рис

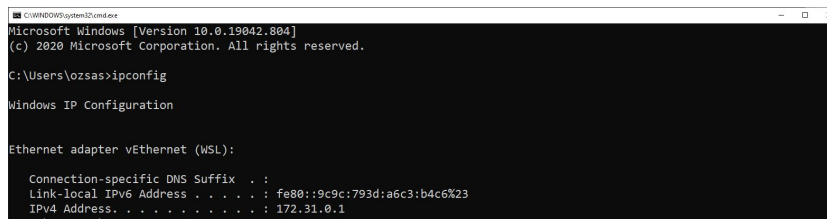


# Виды терминалов

# Windows терминал

В Windows терминал по умолчанию может отличаться в зависимости от версии самой Windows:

- **cmd;**



```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 10.0.19042.804]
(c) 2020 Microsoft Corporation. All rights reserved.

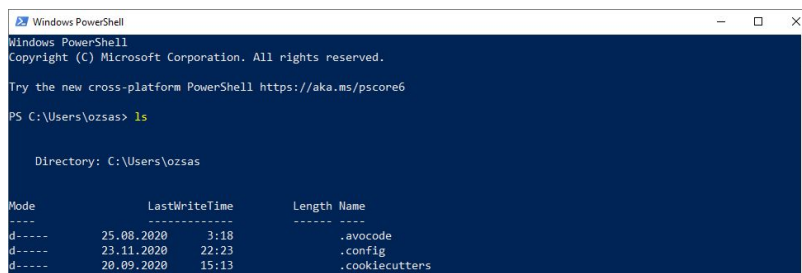
C:\Users\ozsas>ipconfig

Windows IP Configuration

Ethernet adapter vEthernet (WSL):

    Connection-specific DNS Suffix  . : 
    Link-local IPv6 Address . . . . . : fe80::9c9c:793d:a6c3:b4c6%23
    IPv4 Address. . . . . : 172.31.0.1
    Subnet Mask . . . . . : 255.255.240.0
```

- **PowerShell.**



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\ozsas> ls

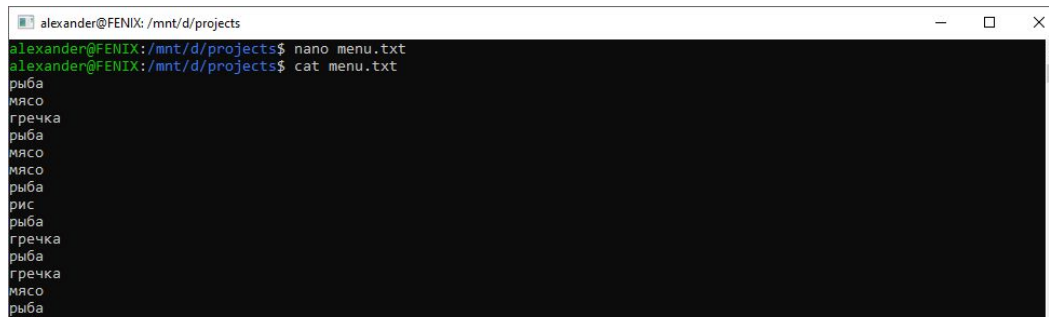
Directory: C:\Users\ozsas

Mode                LastWriteTime         Length Name
----                -
d-----          25.08.2020     3:18         .avocode
d-----          23.11.2020    22:23         .config
d-----          20.09.2020    15:13         .cookiecutters
d-----          20.09.2020    15:13         .cookiecutters
```

# Классический терминал

Классическим терминалом считается юниксовый **bash**. Он является терминалом по умолчанию в таких системах, как:

- Linux;
- MacOS.



```
alexander@FENIX: /mnt/d/projects
alexander@FENIX:/mnt/d/projects$ nano menu.txt
alexander@FENIX:/mnt/d/projects$ cat menu.txt
рыба
мясо
гречка
рыба
мясо
мясо
рыба
рис
рыба
гречка
рыба
гречка
мясо
рыба
```

Мы с вами будем изучать **bash**, т.к. он будет в дальнейшем использоваться для docker и чаще всего вы будете иметь дело именно с linux-подобными серверами.

---

# Что делать с Windows?

Если все-таки работать на **Windows**, то есть два варианта:

- использовать **"обрезанный" bash**:
  - с помощью [cygwin](#)
  - с помощью [git bash](#) (устанавливается вместе с git)
- использовать [wsl/wsl2](#) (для ОС Windows 10) - предпочтительный вариант.

---

# Что такое WSL?

**WSL** (Windows Subsystem for Linux) - подсистема для запуска Linux-программ **непосредственно в Windows** без каких-либо изменений и необходимости использовать традиционную виртуальную машину или двойную загрузку.





# Команды и ключи

# Команды и ключи

**Команды** — это подпрограммы, аналоги функций в python, бывают с параметрами, бывают без. Например, подпрограмма вывода содержимого файла/файлов:

```
cat имя_файла
```

Сам **python** - это тоже подпрограмма, которая запускает интерпретатор.



# Команды и ключи

Рассмотрим **структуру** и несколько примеров простых команд.

**<command> [<argument1> <argument2> ...]**

```
uniq --help  
echo 'Hello World!'  
cat menu.txt  
sort menu.txt -r  
grep -e 'hello' file.txt
```

## Комментарий

Общая структура команд включает в себя:

- саму команду
- аргументы (параметры)

# Команды и ключи

Рассмотрим **структуру** и несколько примеров простых команд.

`<command> [<argument1> <argument2> ...]`

`uniq --help`  
`echo 'Hello World!'`  
`cat menu.txt`  
`sort menu.txt -r`  
`grep -e 'hello' file.txt`

## Комментарий

Аргументы (параметры) бывают:

- обязательные и необязательные
- позиционные и непозиционные
- с дополнительными параметрами и без

---

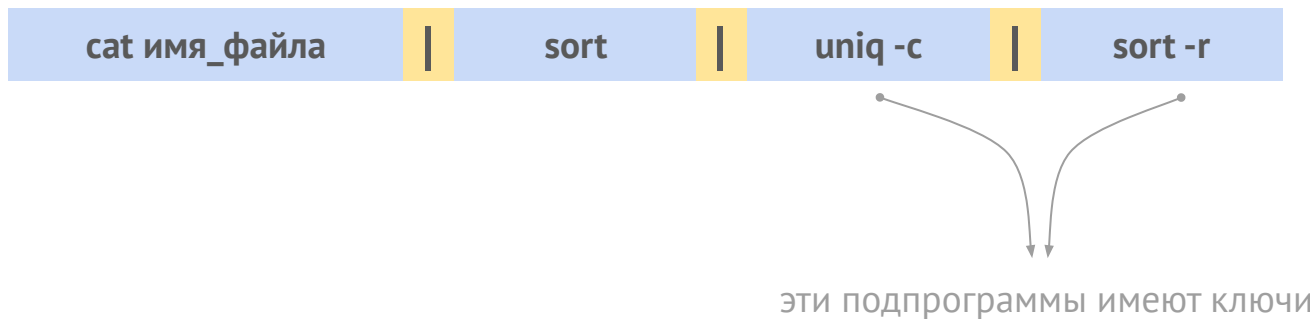
# Команды и ключи

Также стоит отметить, что названия **встроенных команд** обычно короткие и являются сокращениями слов:

- **cat** = concatenate;
- **ls** = list;
- **cd** = change directory.

# Команды и ключи

**Ключи** — это необязательные параметры подпрограмм. Они обычно указываются с одним дефисом и одной буквой. Как в нашем примере с меню:



У ключа бывают две формы:

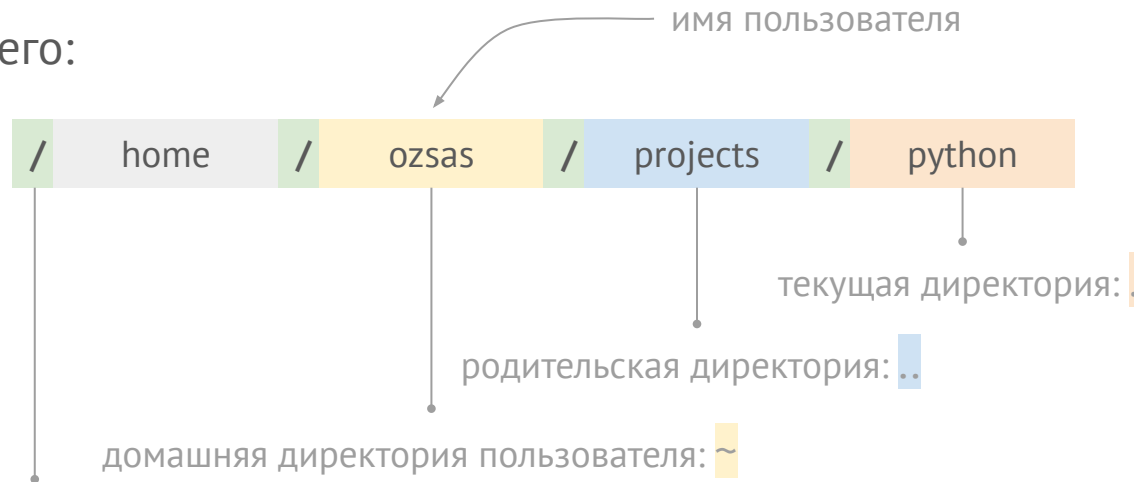
- **короткая** (один дефис и одна буква): `-h`;
- **полная** (два дефиса и слово): `--help`.

# Пути в \*nix

Разберем **пример** пути:




Вынесем его:



корень файловой системы: /

# Полезные хоткеи

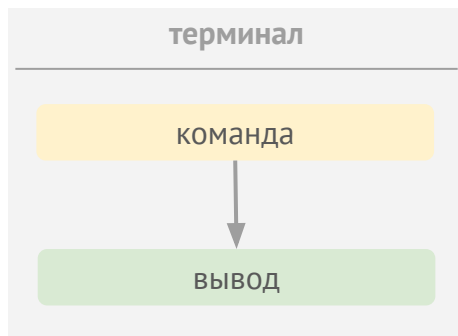
Полезные хоткеи	
Хоткей	Описание
<b>tab</b>	автодополнение
<b>ctrl + C</b>	принудительное завершение большинства процессов
<b>q / exit</b>	корректное завершение большинства процессов
<b>ctrl + W</b>	удалить символы от текущей позиции до начала слова
<b>ctrl + стрелка вправо/влево</b>	переместить курсор к следующему слову в направлении стрелки
<b>стрелка вверх/вниз</b>	"достать" предыдущую/следующую команду
<b>Home</b>	переместиться в начало команды
<b>End</b>	переместиться в конец команды



# Перенаправление потоков ввода/вывода

# Перенаправление потоков ввода/вывода

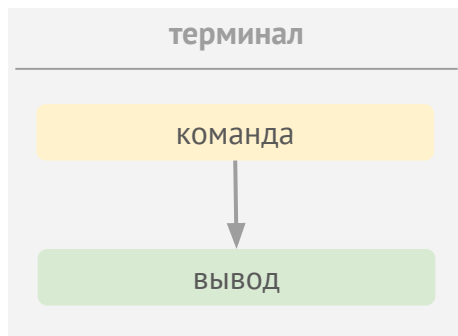
Вы могли заметить, что когда мы **пишем команду в терминал** - вывод её направляется в стандартный поток вывода - то есть туда же в командную строку.





# Перенаправление потоков ввода/вывода

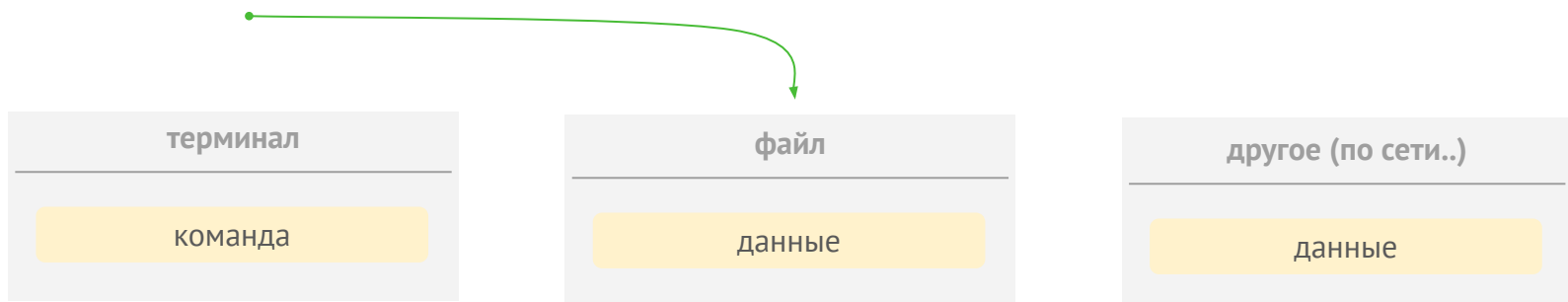
Вы могли заметить, что когда мы **пишем команду в терминал** - вывод её направляется в стандартный поток вывода - то есть туда же в командную строку. Однако поведение не всегда такое и зависит от самой команды.



# Перенаправление потоков ввода/вывода

Данные для команд могут поступать из **стандартного потока ввода** (терминала), а могут и из **файла** или еще **откуда-нибудь** (например, по сети):

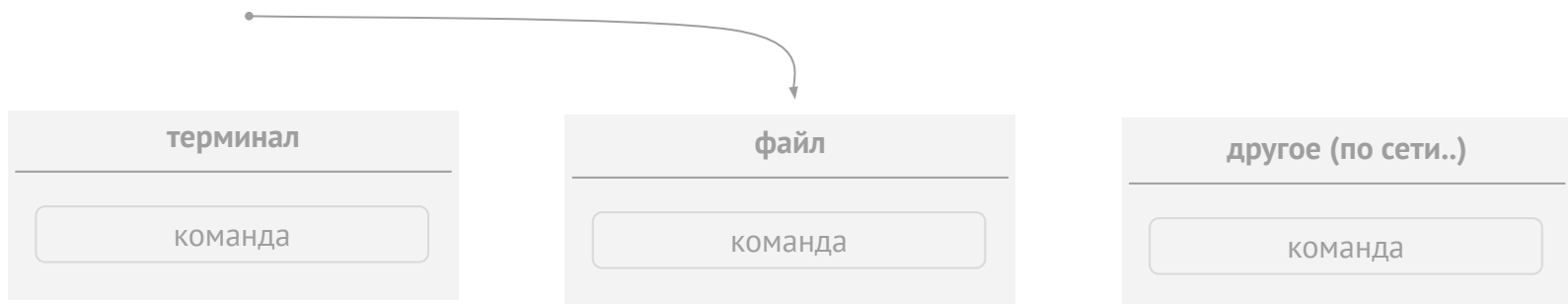
- **< имя\_файла** - перенаправление ввода из файла;



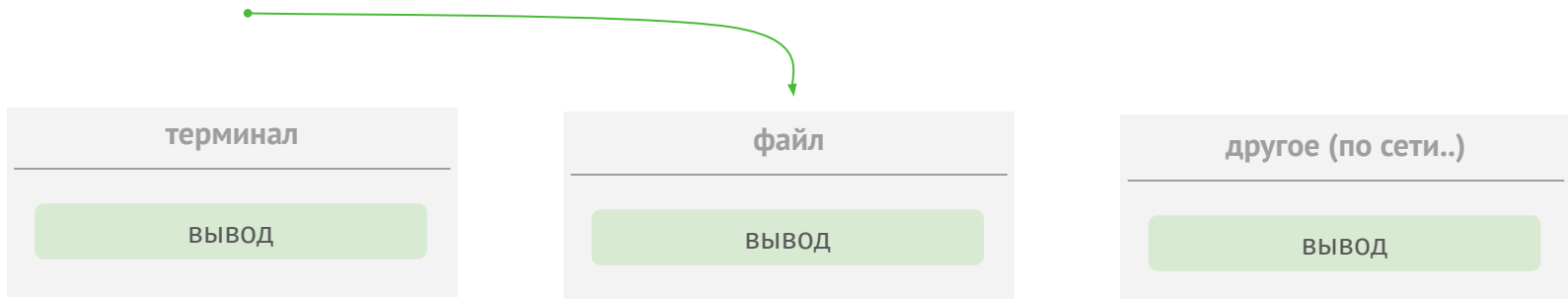
# Перенаправление потоков ввода/вывода

Данные для команд могут поступать из **стандартного потока ввода** (терминала), а могут и из **файла** или еще **откуда-нибудь** (например, по сети):

- **< имя\_файла** - перенаправление ввода из файла;

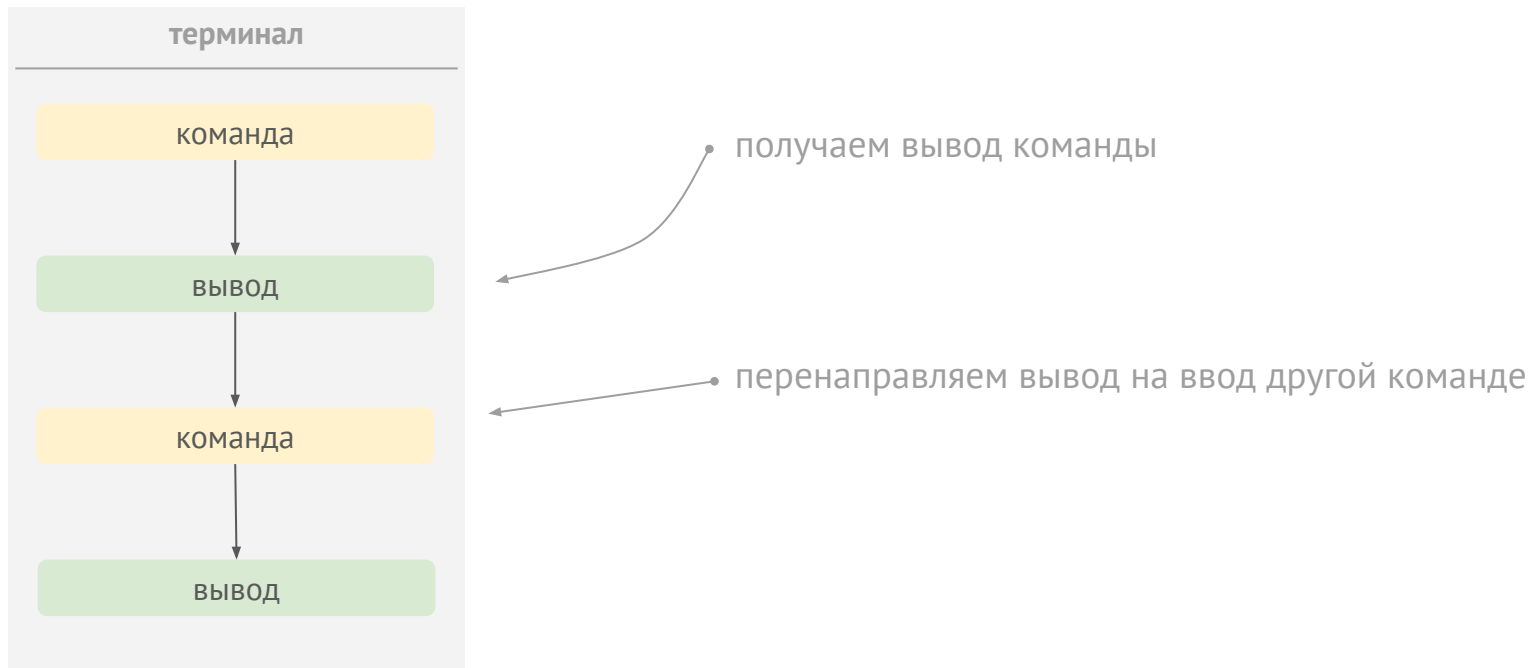


- **> имя\_файла** - перенаправление вывода в файл;



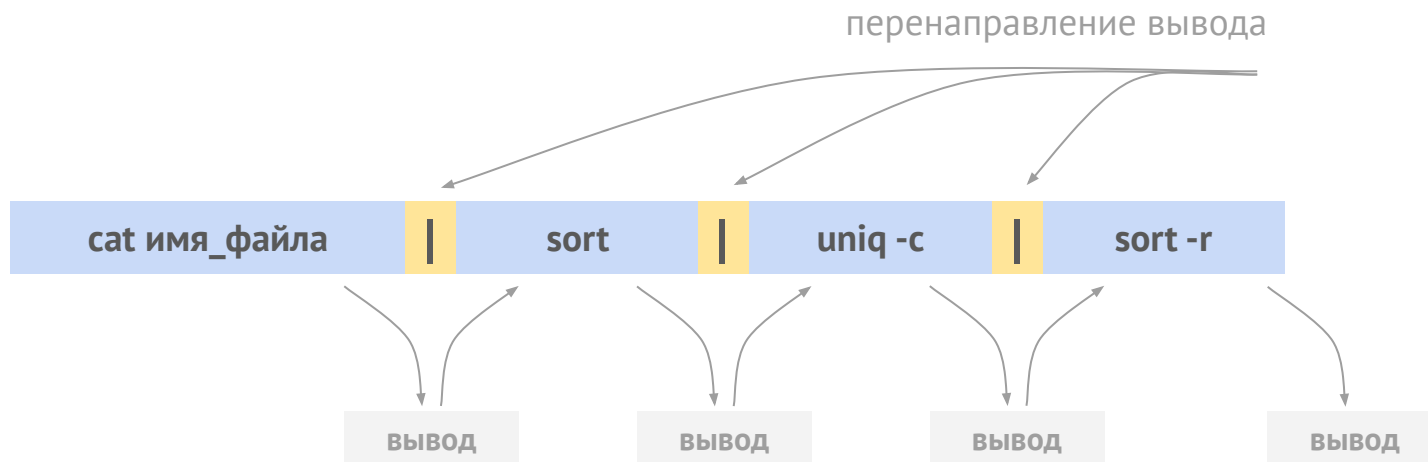
# Конвейер

**Конвейер** - это перенаправление вывода одной команды на ввод другой команде.



# Конвейер

**Конвейер** - это перенаправление вывода одной команды на ввод другой команде. Разберем наш пример: `cat menu.txt | sort | uniq -c | sort -r`



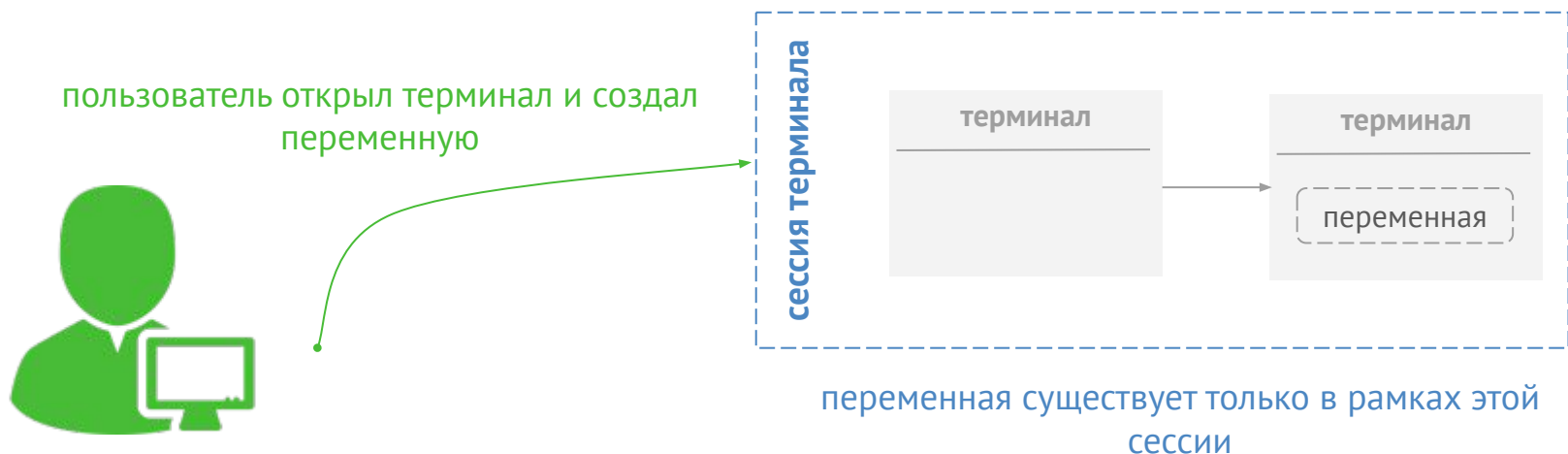
**Перенаправление вывода** реализуется с помощью символа "|" между командами.



# Переменные окружения

# Переменные окружения

Бывают команды или программы, которые реагируют на **переменные окружения** - специальные переменные, которые существуют в рамках сессии терминала (от открытия и до закрытия терминала, в котором она была создана).

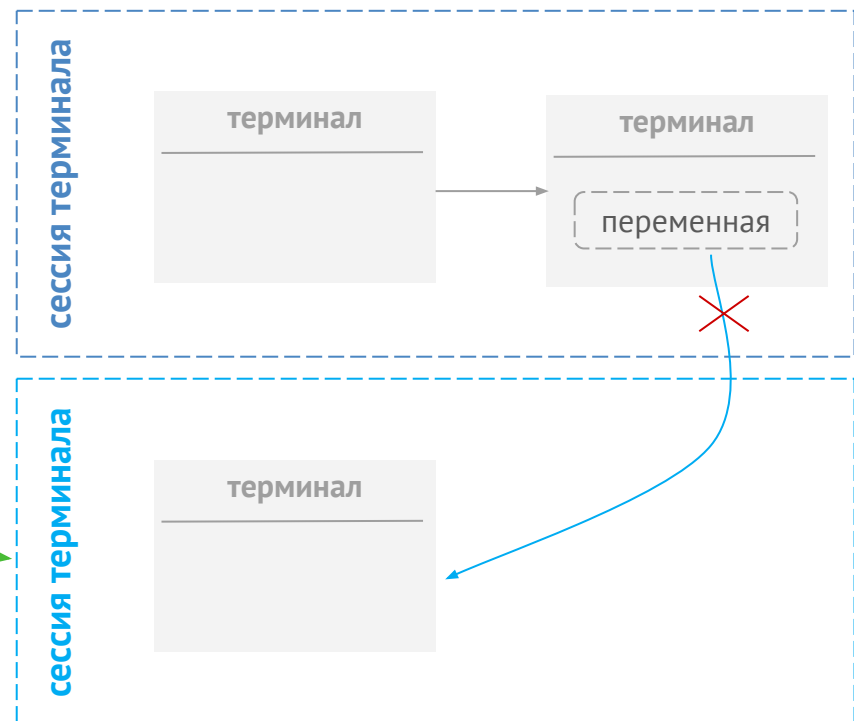


# Переменные окружения

И если открыть новый терминал, то это будет **новая сессия**, в которую переменная перейти **не может**.



пользователь открыл еще один терминал





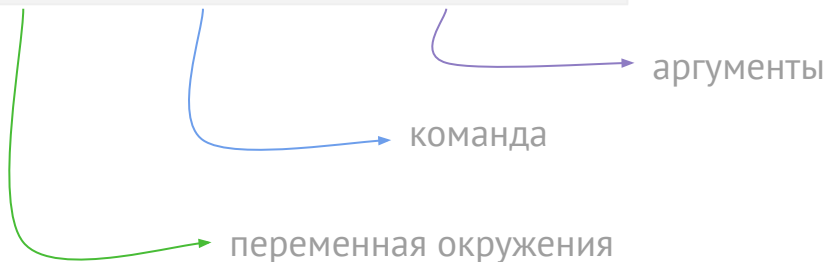
# Переменные окружения

Чтобы **установить переменную**, используется команда `export`:

```
export PORT = 2222
```

Либо можно **установить переменную для одной команды**, написав ее перед запуском:

```
PORT = 2222 python manage.py runserver
```



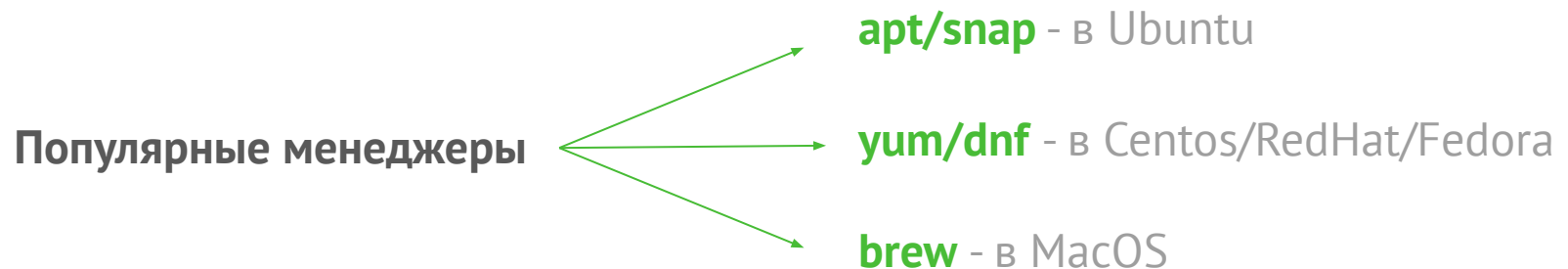


# Пакеты

# Дополнительные программы

Терминал хорош не только тем, что позволяет использовать **существующие команды**, но и устанавливать **новые**. Причем "не отходя от кассы".

В каждом уважающем себя терминале есть **пакетный менеджер** - специальная команда, которая ищет, скачивает, устанавливает новые пакеты - наборы дополнительных программ (иногда пакеты состоят из одной программы, иногда из нескольких). Также пакетный менеджер занимается управлением и удалением пакетов.



## Пример

В Ubuntu по-умолчанию не установлена команда `ifconfig` - команда для удобного просмотра сетевых интерфейсов. Эта команда входит в пакет `net-tools`, который мы сейчас установим.

Для установки воспользуемся **пакетным менеджером**:

```
apt install net-tools
```

Если у вас нет прав суперпользователя (администратора системы), то надо добавить `sudo` - указание выполнить команду от имени суперпользователя:

```
sudo apt install net-tools
```

---

## Практика на хостинге

На практике мы будем использовать хостинг, предоставляемый компанией [REG.RU](https://reg.ru)

При выполнении домашнего задания также вы сможете попрактиковаться на виртуальных серверах с сервисными функциями для разработчиков.



**ДОМЕНЫ  
ХОСТИНГ  
СЕРВЕРЫ**

## Дополнительные материалы

- где искать **команды** и их описание: <https://www.opennet.ru/man.shtml>;
- про **WSL/WSL2**: <https://docs.microsoft.com/ru-ru/windows/wsl/about>;
- **набор команд** для ознакомления:

набор команд					
cd	grep	uniq	ll	head	df
pwd	awk	touch	tail	mv	sort
rm	sed	echo			



# Итоги

---

# Итоги

- **Узнали**, что такое терминал;
- **Узнали** про разные виды терминалов;
- **Рассмотрели** команды и ключи;
- **Узнали** про перенаправление потоков ввода/вывода;
- **Узнали** про переменные окружения;
- **Узнали** про пакетные менеджеры.



❖ нетология

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

**Адилет Асанкожоев**