

Кафедра алгоритмов и технологий  
программирования



## Практикум по Linux

Ивченко Олег





# План

## Модули

- Linux: 35%
- WSL: 35%
- Git: 35%

## Оценка по каждому модулю

- Активность на занятии
- Тест

Получить удостоверение: > 70%



# Преподаватели



Ивченко Олег



Яндекс



Чернецкий Аркадий



ШКОЛА АНАЛИЗА ДАННЫХ

Опыт работы с Linux  
> 6 лет



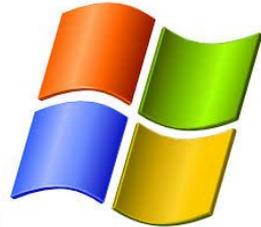
# План

- Linux vs. Windows
- Подключение к серверу и Jupyter
- Bash-скрипты
  - основные команды, permissions
  - shebang
  - переменные окружения
  - установка пакетов
- Tmux
- cron, at
- firewall



# Linux vs. Windows

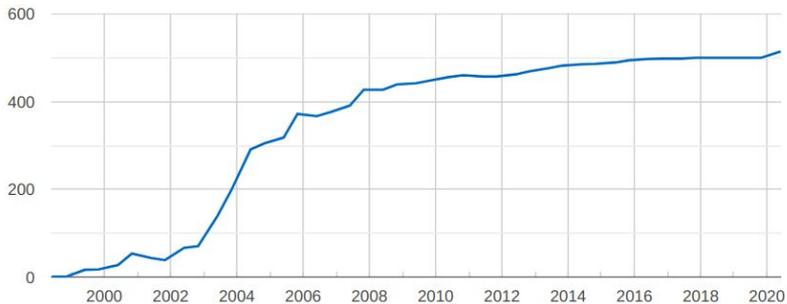
- Все файлы равнозначны с точки зрения системы
  - расширение (“суффикс”) не имеет значения
  - работа с файлом зависит от прав доступа
- Вместо задач - процессы
  - процесс может быть запущен пользователем
  - демоном
- проще ставить приложения
- нет проблем с лицензией ОС
- [для увлеченных]. Можно **полностью** настроить под себя



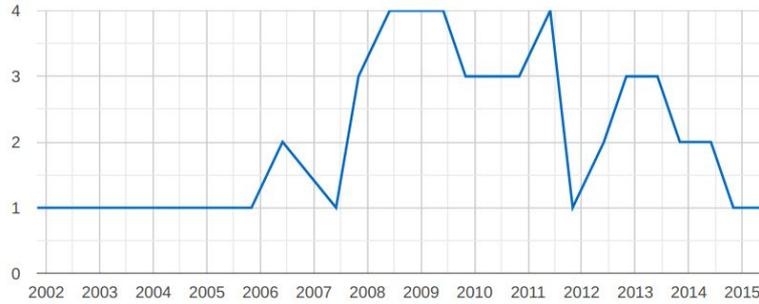


# Linux vs. Windows

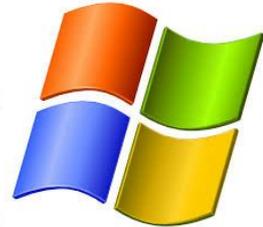
- Linux используется на всевозможных устройствах:
  - телефоны (Android!)
  - роутеры
- Почти все суперкомпьютеры используют Linux



Linux



Windows





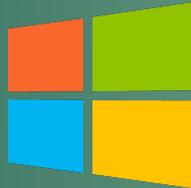
# Архитектура файловой системы

## ➤ Основные каталоги

```
~$ ls /  
bin      etc          lost+found  opt      run      srv      usr  
boot    home        lib         media   proc    sbin    sys  
var  
dev     lib64      mnt        root    snap    tmp
```



```
000011100010110000101010000101000000111000101110000110010001010000  
00111000101110001011110100001010000001110001011100001100100010100000011100010111100011011000101000  
000011100010110000101010000101000000111000101110000110010001010000001110001011110001101100010100070
```

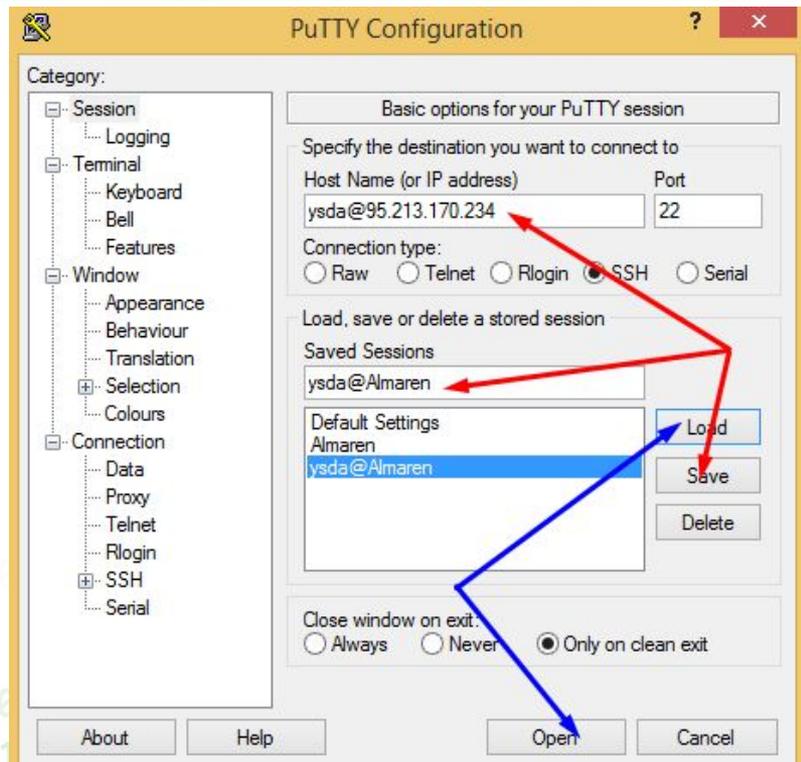


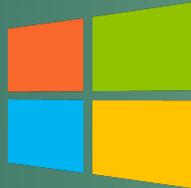
# Подключение к серверу

## Базовый алгоритм

- вводим USER@HOST, название сессии
- сохраняемся

```
95.213.170.234 - PuTTY
Using username "ysda".
ysda@95.213.170.234's password: █
```





# Подключение к серверу

- Подключиться к серверу:

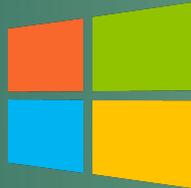
```
ssh USER@server_hostname
```

- **USER** - см. в почте
- `server_hostname = lorinen.atp-fivt.org`

- Далее вводим пароль.

- **Как проще? (и безопаснее)**





# Подключение. PowerShell

**А ещё есть PowerShell.** На Win 8.x неудобен.

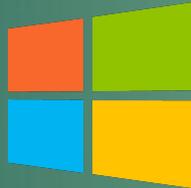
1. Не умеет ssh.

- a. Ставим `Open SSH`.
- b. Сервер должен поддерживать шифрование 'diffie-hellman-group1-sha1'
- c. OpenSSH генерит ключи с правами 644.

2. Не умеет ssh-copy-id

- a. Но есть `стороний скрипт`.
- b. К нему ещё нужен `PLink`.

**Сложно :(**



# Подключение. Git Bash

Нам понадобится **Git** (git bash)



- Есть среда CygWin и встроенные ssh, scp
- Работаем аналогично Linux
- **/etc/hosts** здесь лежит в **C:\Program Files\Git\etc\hosts**

```
vagrant@win-8-1-pro-x64 MINGW64 ~  
$ ssh-keygen.exe -q  
Enter file in which to save the key (/c/Users/vagrant/.ssh/id_rsa):  
Enter passphrase (empty for no passphrase):  
Enter same passphrase again:
```

```
vagrant@win-8-1-pro-x64 MINGW64 ~  
$ ssh-copy-id ysda@almaren  
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed: "/c/Users/vagrant/.ssh/id_rsa.pub"  
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed  
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys  
ysda@almaren's password:  
  
Number of key(s) added: 1  
  
Now try logging into the machine, with: "ssh 'ysda@almaren'"  
and check to make sure that only the key(s) you wanted were added.
```



# Обмен данными

1. Копирование файлов на сервер

```
scp [-r] /path/to/local_dir USER@almaren:/remote/path
```

2. Копирование файлов с сервера

```
scp [-r] USER@almaren:/remote/path /path/to/local_dir
```

3. Сокращения

```
/home/USER = ~/
```

```
/current/directory = .
```

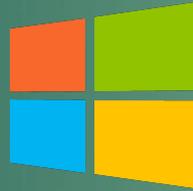
```
-c blowfish # для ускорения  
-P XXXX # ssh port
```

10110001010000

10110001010000

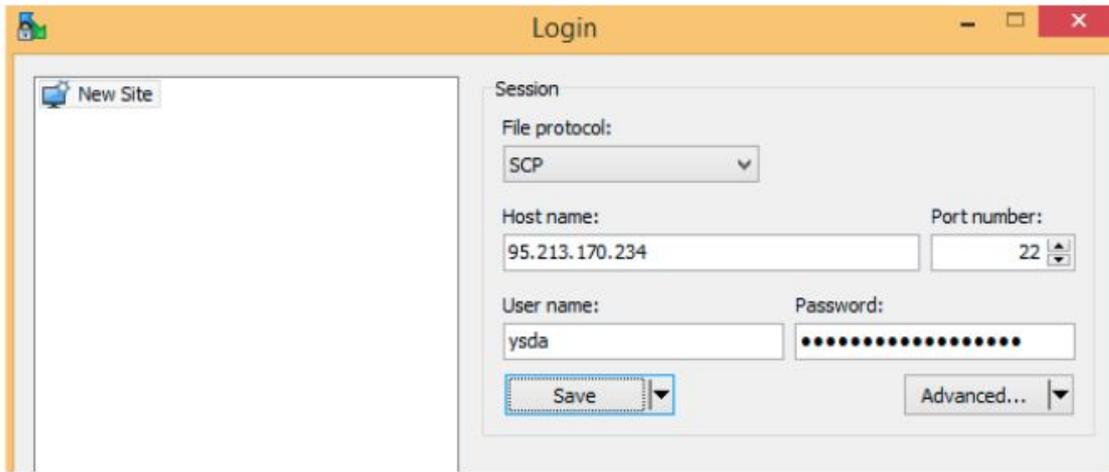
11011000101000

10110001010000



# Обмен данными

1. С помощью Git Bash аналогично Linux
2. Для UI - **WinSCP**
  - если нет ssh-ключа



Login

New Site

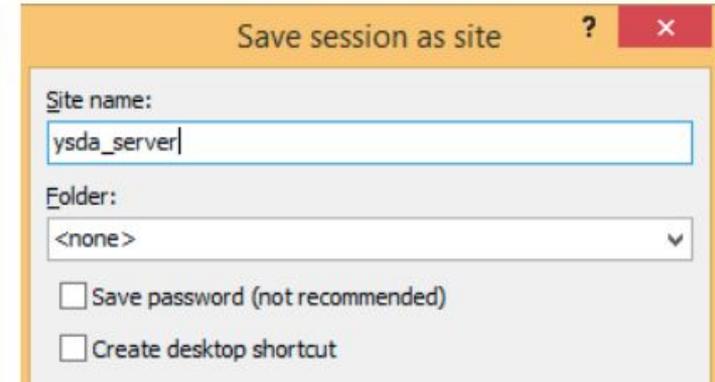
Session

File protocol:  
SCP

Host name: 95.213.170.234 Port number: 22

User name: ysda Password: .....

Save Advanced...



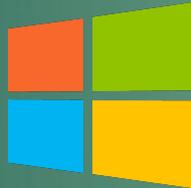
Save session as site

Site name:  
ysda\_server

Folder:  
<none>

Save password (not recommended)

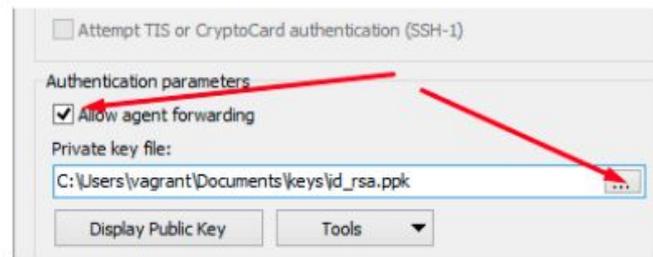
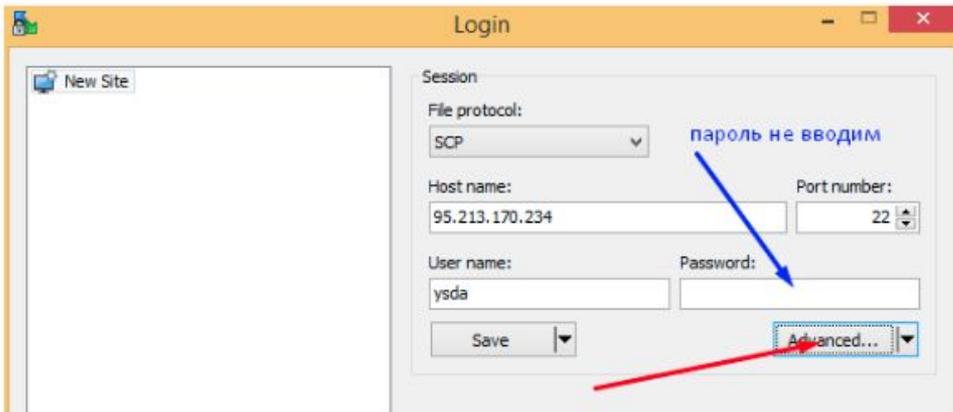
Create desktop shortcut



# Обмен данными

## 2. Для UI - WinSCP

> если есть ssh-ключ (лучше!)



сохраняем сессию и логинимся





# Jupyter. Quick start

## 1. Запуск Jupyter на сервере

```
jupyter notebook --no-browser --port PORT --ip 0.0.0.0
```

**PORT** - номер порта (от 1024 до 46656).

```
[I 14:57:40.595 NotebookApp] The port PORT is already in use,  
trying another port.
```

Jupyter ищет незанятый порт (**PORT++** или **PORT--**)

## 2. Токен

- генерится сам при старте `jupyter nb`
- можно сгенерить свой и сохранить в `.jupyter/jupyter_notebook_config.py`

```
c.NotebookApp.token = u'mY_seCure_tok#n99!_***'
```

**Если порты открыты, токен нужен ОБЯЗАТЕЛЬНО!!**

01010000

01010000

00101000

01010000



# Jupyter. Quick start

## 3. Подключение из локальной машины

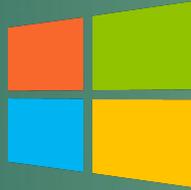
- Открыть в браузере SERVER\_IP:PORT
- ввести токен
- если доступа нет:
  - порт занят
  - неправильный токен

```
jupyter notebook list # выдаёт порт и токен для каждого NB
```

## 4. Используйте `tmux`

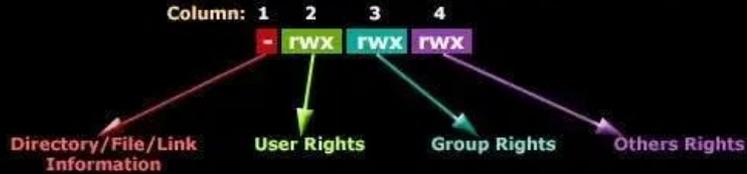
0000

```
00001110001011000010101000010100000011100010111000011001000101000000111000101111000110110001010000  
00111000101110001011110100001010000001110001011100001100100010100000011100010111100011011000101000  
00001110001011000010101000010100000011100010111000011001000101000000111000101111000110110001010000
```



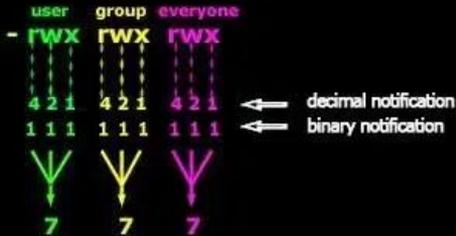
# Подключение к серверу

## Understanding The Linux File Permissions



While the first column defines a directory, file or link, the next 3 columns (2, 3, 4) define the permissions for the User, Group and Others (everyone else) groups.

## Linux Permissions Made Easy



decimal notation: add each number to obtain the value (4 + 2 + 1 = 7)

binary notation: convert it to decimal then you should have the value (r-x = 101 base 2 = 5 base 10)







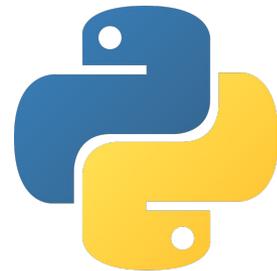
# Как поставить доп. библиотеку

## 1. С помощью `pip`

```
pipX install MY_LIB==MY_VERSION --user
```

$$X = \left\{ \begin{array}{l} 2 \text{ для Python 2.7} \\ 3 \text{ для Python 3.6} \end{array} \right\}$$

```
pythonX -m pip install -r requirements.txt
```





# Как поставить доп. библиотеку

## 2. С помощью `venv`

Создаем окружение

```
python -m virtualenv MY_ENV # для Python2.7  
python3 -m venv MY_ENV # для Python3.6
```

Заходим в окружение

```
source MY_ENV/bin/activate
```

Можем ставить пакеты (`pip install...`), открывать Jupyter и т.д.

Создаем в окружении ядро для Jupyter

```
ipython kernel install --user --name=MY_ENV
```

- Выйти из окружения - ввести `deactivate` находясь внутри него.
- Удалить окружение - удалить папку `~/MY_ENV`

110001010000

110001010000

011000101000

110001010000



# Как поставить доп. библиотеку

## 3. С помощью apt

1) Скачать пакет

```
apt-get install download MY_PACKAGE # для "install" нужно sudo!
```

2) Поставить локально

```
dpkg -x ~/downloaded_package.deb $HOME
```

3) В директории \$HOME появится папка, добавить её в PATH.

Дописываем в ~/.profile строки

```
if [ -d "${HOME}/usr/" ]  
then  
    PATH="${HOME}/usr/share:${HOME}/usr/games ... :${PATH}"
```

```
fi  
➤ source ~/.profile или перезайти в терминал
```

110001010000

110001010000

011000101000

110001010000



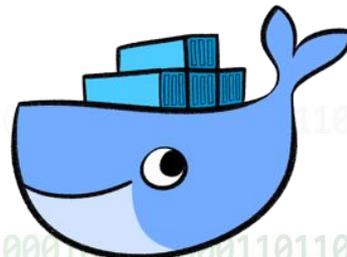
# Как поставить доп. библиотеку

## 4. С помощью `apt` - собрать из исходников

```
apt-get source MY_PACKAGE  
cd MY_PACKAGE  
./configure --prefix=$HOME  
make  
make install
```

## 5. Развернуть докер-контейнер

➤ Например, `nvidia-docker`.





# IPtables

- Что мы обычно делаем в IPtables?
  - Открываем / закрываем порты
  - Открываем для траффика с выбранных хостов

```
iptables --policy INPUT DROP # ставим политику DROP по умолчанию
iptables -I INPUT -p tcp -s СПИСОК_IP_ЧЕРЕЗ_ЗАПЯТУЮ -j ACCEPT # открываем всё внутри кластера
iptables -A INPUT -p tcp --dport 22 -j ACCEPT # открываем ssh
```

# Закрываем порт по UDP

```
iptables -A INPUT -p udp -s 10.10.10.0/24 --dport 111 -j ACCEPT && iptables -A INPUT -p udp -s 127.0.0.1
--dport 111 -j ACCEPT && iptables -A INPUT -p udp --dport 111 -j DROP && iptables-save
```

**Сложно!**



# IPtables

- Есть Uncomplicated firewall (UFW)
  - <https://help.ubuntu.com/community/UFW>
- А ещё лучше - ferm
  - ставим пакет ferm
  - настраиваем конфиг **/etc/ferm/ferm.conf**
  - ребутаемся



# IPtables

- Пример: <http://gg.gg/ferm-example>
- А как же IPv6?
  - Ferm с ним работает
  - Если не нужно, отключите IPv6 явно!

## 1. Добавить в /etc/sysctl/conf

```
net.ipv6.conf.all.disable_ipv6 = 1  
net.ipv6.conf.default.disable_ipv6 = 1  
net.ipv6.conf.lo.disable_ipv6 = 1
```

## 2. sysctl -p



# Cron

## ➤ Запуск задач по расписанию

- `crontab -l` # список задач
- `crontab -e` # редактирование задач

## ➤ Пример

```
0 */2 * * * /root/check_mem.py # запуск скрипта каждые 2 ч.
```

Время запуска представляется в таком виде:

минута час день\_месяца месяц день\_недели команда

Значение	Диапазон	Дополнительно
минуты	0-59	
часы	0-23	
дни месяца	1-31	
месяцы	1-12	можно задавать и в 3-х буквенном варианте
дни недели	0-6	можно задавать и в 3-х буквенном варианте (0=воскресенье)

Символ '\*' подразумевает - любое значение.

```
Select an editor. To change later, run 'select-editor'.
 1. /usr/bin/vim.nox
 2. /bin/nano      <---- easiest
 3. /usr/bin/vim.basic
 4. /usr/bin/mcedit
 5. /usr/bin/vim.tiny
 6. /bin/ed

Choose 1-6 [2]:
```

## Дополнительные переменные cron

Переменная	Описание	Эквивалент
@reboot	Запуск при загрузке	
@yearly	Раз в год	0 0 1 1 *
@annually	Тоже что и @yearly	
@monthly	Раз в месяц	0 0 1 **
@weekly	Раз в неделю	0 0 ** 0
@daily	Раз в день	0 0 ***
@midnight	В полночь (00:00)	
@hourly	Каждый час	0 ****



# Cron

## ➤ Отправка вывода по Email

- поставить mailutils
- ребутнуть service postfix reload
- В описание задачи добавить **MAILTO**

## ➤ Пример

```
MAILTO="k.velkerr@gmail.com"
```

```
* * * * * ls -l
```

```
Cron <velkerr@bilbo> ls -l
```



root@bilbo.atp-fivt.org Сегодня, 11:36

Кому: k.velkerr@gmail.com

```
total 12
```

```
-rw----- 1 velkerr velkerr 111 Aug 31 10:29 dead.letter  
drwxrwxr-x 12 velkerr velkerr 4096 Sep 20 2019 MTPProxy  
drwxrwxr-x 2 velkerr velkerr 4096 Jul 10 22:50 reboot_runners
```

## ➤ Выводится лишнее?

- grep
- [cronic](#)

010111000011001000101000000111000101111000110110001010000

010111000011001000101000000111000101111000110110001010000

010111000011001000101000000111000101111000110110001010000

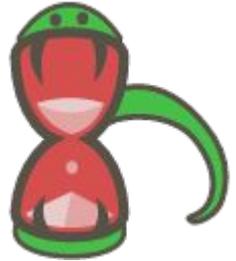
000011100001011000001010100000111000101111000110110001010000



# Cron

## ➤ Ссылки

- Конструктор cron-задач [Crontab Guru](#)
- [Tutorial](#) по Cron на русском.
- Python lib для управления cron-задачами [python-crontab](#)
  - Осторожно! `crontab.write()` багнутый!
- [Crontab-UI](#)



## ➤ Утилита АТ

- [Guide](#) от IBM



# Полезные ссылки

## ➤ Bash

- [права доступа](#)
- [ещё про права доступа и про sticky bit](#) ('t' в permissions на каталог).
- [Tutorial по командам Linux](#)
- [Tutorial по Bash](#)
- [Bash cheatsheet](#)
- [множественный if в bash](#)
- [Fish](#)