

Технологии программирования

Занятие 1

Linux + Bash

Полезные ссылки

akht.pl/tp2021-drive - Google Drive

akht.pl/tp2021-channel - канал курса с объявлениями

akht.pl/tp2021-chat - чат курса

akht.pl/tp2021-grades - общая таблица

akht.pl/tp2021-seminar01 - ссылка на этот семинар

akht.pl/tp2021-signup - ссылка на форму регистрации

Какие задания будут в курсе?

- Технические задания - будут приниматься в автоматическом режиме с просмотром кода
- Статусы о получении задания проставляются в таблице в течение 10-15 минут
- Задания на проектирование и реализацию проекта - будет куратор у каждой группы
- Большие лекционные контрольные!

Цели курса

- Дать практические основы работы/инструментов разработчика в IT-индустрии
- Ознакомить с паттернами и антипаттернами при программировании / проектировании
- Улучшить качество кода
- Расширить кругозор (узнать best practices) в области IT-индустрии

Пора ставить Linux!

Хотя бы на виртуальную машину

Структура файловых систем: Linux vs Windows

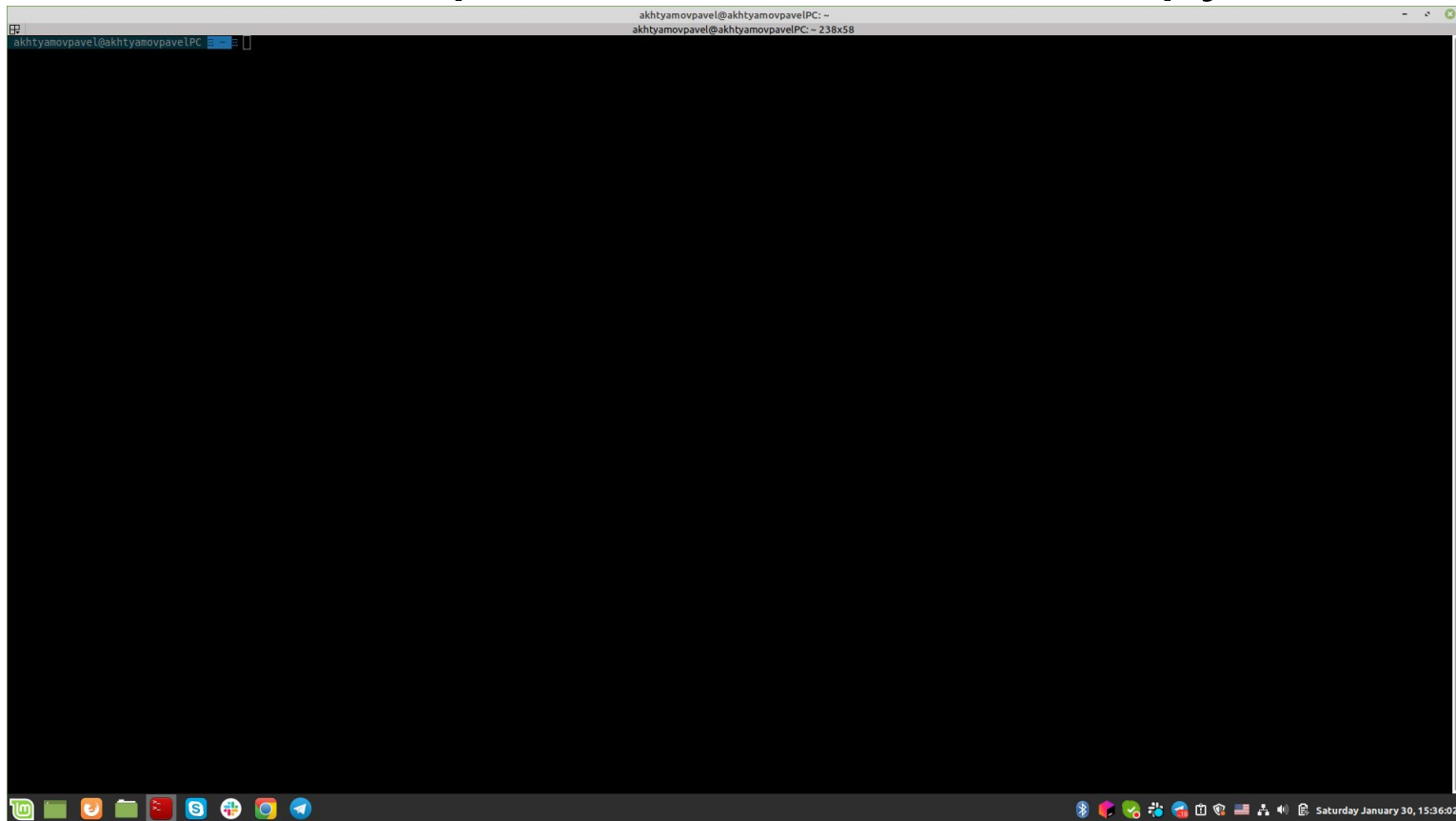
Linux:

- /
 - home/
 - pavlon/
 - 1.pdf
 - 1.doc
 - petya/
 - boss/
 - report.doc
 - etc/
 - usr/
 - lib/
 - local/
 - bin/
 - var/
 - dev/
 - boot/

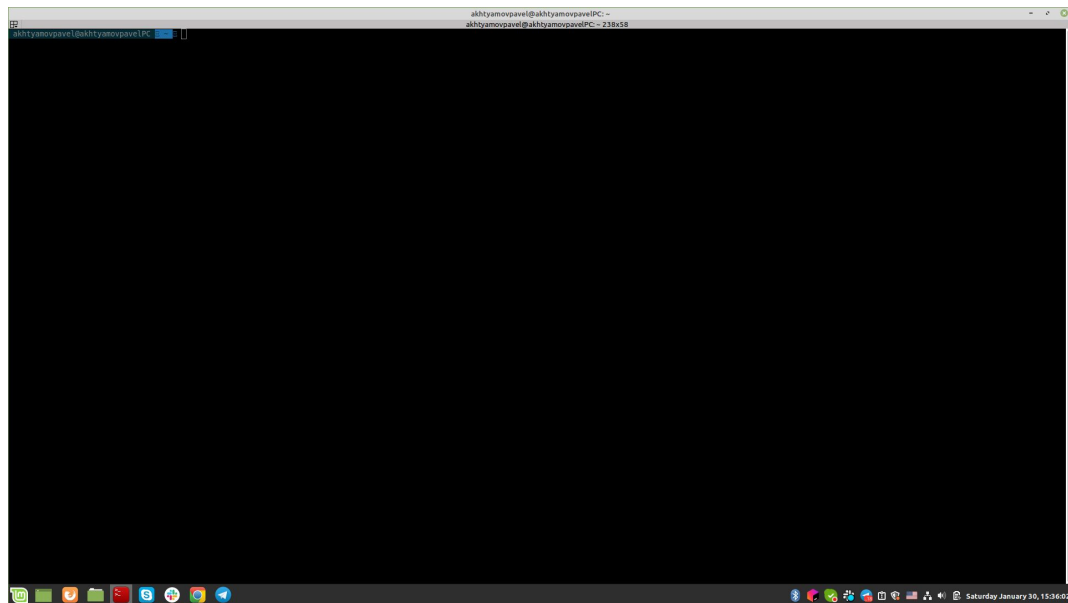
Windows:

- A:\
- C:\
 - Windows\
 - System32\
 - cmd.exe
 - Users\
 - pavlon\
 - petya\
 - boss\
- D:\

Знакомьтесь - терминал, ваш новый друг!



Знакомьтесь - терминал, ваш новый друг!



В Linux-подобных системах можно делать все через терминал! Честно-честно!

Основной инструмент - терминал!!

touch filename

создать файл: touch 1.txt

ls [path] (**LiSt**)

вывести список файлов: ls ./

mkdir name (**MaKe DiRectory**)

создать директорию: mkdir myfolder

cp from where (**CoPy**)

скопировать файл: cp 1.txt myfolder/1.txt

cd where (**Change Directory**)

перейти в каталог: cd myfolder

rm filename (**ReMove**)

удалить папку: rm myfolder/1.txt

pwd (**Print Work Directory**)

распечатать текущую директорию

wc [filename] (**Word Count**)

узнать количество символов/слов/строк в файле или stdin

А теперь важная команда!

man ls

All options?

```
LS(1)                                BSD General Commands Manual                                LS(1)

NAME
    ls -- list directory contents

SYNOPSIS
    ls [-ABCFGHLOPRSTUW@abcdefghijklmnopqrstuvwxyz1] [file ...]

DESCRIPTION
    For each operand that names a file of a type other than directory, ls displays its name as well as any requested, associated information. For each operand that names a file of type directory, ls displays the names of files contained within that directory, as well as any requested, associated information.

    If no operands are given, the contents of the current directory are displayed. If more than one operand is given, non-directory operands are displayed first; directory and non-directory operands are sorted separately and in lexicographical order.

    The following options are available:
```

А теперь вопросы на внимательность?

- Как создать папку **myfolder**, а внутри нее **mydir**?

А теперь вопросы на внимательность?

- Как создать папку **myfolder**, а внутри нее **mydir**?
- Как скопировать папку **mydir** в папку **mydir2**?

А теперь вопросы на внимательность?

- Как создать папку **myfolder**, а внутри нее **mydir**?
- Как скопировать папку **mydir** в папку **mydir2**?
- Создаем файл 1.txt. Как узнать его размер при помощи ls?
- Какой размер у директории?

А теперь вопросы на внимательность?

- Как создать папку **myfolder**, а внутри нее **mydir**?
- Как скопировать папку **mydir** в папку **mydir2**?
- Создаем файл 1.txt. Как узнать его размер при помощи ls?
- Какой размер у директории?
- Как посмотреть скрытые файлы в директории?

А теперь вопросы на внимательность?

- Как создать папку **myfolder**, а внутри нее **mydir**?
- Как скопировать папку **mydir** в папку **mydir2**?
- Создаем файл 1.txt. Как узнать его размер при помощи ls?
- Какой размер у директории?
- Как посмотреть скрытые файлы в директории?
- Как удалить папку?

А теперь вопросы на внимательность?

- Как создать папку **myfolder**, а внутри нее **mydir**?
- Как скопировать папку **mydir** в папку **mydir2**?
- Создаем файл 1.txt. Как узнать его размер при помощи ls?
- Какой размер у директории?
- Как посмотреть скрытые файлы в директории?
- Как удалить папку?
- А если не позволяет?

А теперь вопросы на внимательность?

- Как создать папку **myfolder**, а внутри нее **mydir**?
- Как скопировать папку **mydir** в папку **mydir2**?
- Создаем файл 1.txt. Как узнать его размер при помощи ls?
- Какой размер у директории?
- Как посмотреть скрытые файлы в директории?
- Как удалить папку?
- А если не позволяет?
 - **sudo** - запуск от прав администратора!
- Что будет, если удалить корневую папку??

Работа с регулярными выражениями

- **touch** 1.txt
- **touch** 2.txt
- **mkdir** texts_copy
- **cp** *.txt texts_copy - скопировать все файлы .txt в директорию
- **mkdir** texts
- **mv** *.txt texts - переместить все файлы .txt в директорию
- **rm** texts_copy/*.txt - удалить все файлы .txt в директории

Работа с файлами

<code>touch filename</code>	Создает файл: <code>touch test</code>
<code>command > input</code>	Запись вывода command в файл input : <code>echo "Hello World" > 1.txt</code>
<code>command >> input</code>	Добавление вывода command в файл input : <code>echo "Hello World" >> 1.txt</code>
<code>cat test</code>	Печатает содержимое файла
<code>less / more</code>	Просмотр файлов с прокруткой
<code>command 2> input</code>	Запись <i>stderr</i> command в файл input :
<code>cmd1 cmd2</code>	Перенаправление <i>stdout</i> cmd1 в <i>stdin</i> cmd2

Как узнать количество файлов в директории?

Подсказка: используйте | (pipe)

Как узнать количество файлов в директории?

```
ls <directory> | wc -l
```

Скрипты Bash

Скрипты Bash - последовательность команд на языке Bash.

Запуск `myprogram` - `./myprogram`

Запуск с правами администратора - `sudo ./myprogram`

Что будет, если запустить на выполнение `./1.txt` ?

Скрипты Bash

Скрипты Bash - последовательность команд на языке Bash.

Запуск `myprogram` - `./myprogram`

Запуск с правами администратора - `sudo ./myprogram`

Что будет, если запустить на выполнение `./1.txt` ?

permission denied!!

Права пользователей

В Linux есть три уровня прав к файлу/директории:

- Пользователя (**read**, **write**, **executable**)
- Группы пользователей (**read**, **write**, **executable**)
- Всех пользователей (**read**, **write**, **executable**)
 - Можно увидеть через `ls -l`

```
-rw-r--r--  1 akhtyamovpavel  staff      168 Oct 24  2017 pgadmin.log
-rw-r--r--  1 akhtyamovpavel  staff      381 Dec 17 12:08 process_task.sh
drwxr-xr-x 44 akhtyamovpavel  staff    1408 Nov 10 16:02 proj
drwxr-xr-x 30 akhtyamovpavel  staff     960 Nov  2 13:17 scikit-image
drwxr-xr-x  3 akhtyamovpavel  staff     96 Mar 12  2018 seaborn-data
drwxrwxr-x  6 akhtyamovpavel  staff    192 Sep 11  2017 ssh-keys
-rw-r--r--  1 akhtyamovpavel  staff   1914 Dec  1 14:29 stop_words_en.txt
-rw-r--r--  1 akhtyamovpavel  staff    680 Dec 24 00:23 test.txt
drwxr-xr-x 28 akhtyamovpavel  staff     896 Nov 30  2017 torch
```


Почему в Linux-е “нет вирусов”?

Почему в Linux-е “нет вирусов”?

- Попробуйте скачать exe-шник и запустить?

Почему в Linux-е “нет вирусов”?

- Попробуйте скачать exe-шник и запустить?
- Ой, он неисполняемый

Почему в Linux-е “нет вирусов”?

- Попробуйте скачать exe-шник и запустить?
- Ой, он неисполняемый
- Команда **chmod** в помощь!

Почему в Linux-е “нет вирусов”?

- Попробуйте скачать exe-шник и запустить?
- Ой, он неисполняемый
- Команда **chmod** в помощь!
- С какой опцией его надо запустить?

Почему в Linux-е “нет вирусов”?

- Попробуйте скачать exe-шник и запустить?
- Ой, он неисполняемый
- Команда **chmod** в помощь!
- С какой опцией его надо запустить?

`chmod +x`

Полезные команды*

- В работе полезны команды:
 - `htop/top` - “Диспетчер задач в Windows”
 - `tree` - показать список файлов в виде дерева
 - `realpath [file]` - показывает полный путь к файлу
 - `watch [cmd]` - позволяет запускать команду через некоторый интервал времени
- Поставить дополнительную утилиту:
 - `sudo manager install package`

manager бывает:

- **apt** - Ubuntu/Debian/Linux Mint
- **yum** - Red Hat (Fedora)
- **brew** - Mac OS (ставится отдельно)

Домашнее задание

1. Поставить ОС, основанную на Linux
2. Поставить git в ОС
3. Поставьте любимый браузер в ОС

Скриптинг (bash)

Зачем писать скрипты???

Скриптинг (bash)

Зачем писать скрипты???

Первый шаг в автоматизации процессов

Не надо повторять команды!

Список переменных в Bash

- **\$0** - имя скрипта
- **\$1 - \$9** - первые девять аргументов
- **\$#** - количество аргументов
- **\$@** - список всех аргументов
- **\$?** - статус самой последней завершенной задачи
- **\$\$** - id текущего скрипта
- **\$USER** - имя пользователя
- **\$HOSTNAME** - имя компьютера
- **\$SECONDS** - время со старта текущего скрипта
- **\$RANDOM** - рандом относительно времени
- **\$LINENO** - номер строки в скрипте

Переменные окружения

Ключевые переменные в Bash, которые используются утилитами в Linux.

- `$PATH` - путь к исполняемым файлам (не надо писать полный путь)
- `$HOME` - путь к домашней директории
- `$PWD` - текущая директория
- `$LANG` - язык системы (локаль - Locale)
- `$SHELL` - используемая утилита для скриптов (изначально - Bash)
- `$LD_LIBRARY_PATH` - где лежат библиотеки

Разделитель - :

01. Время программы

Напишите скрипт, который показывает, сколько секунд выполнялась программа (time использовать нельзя)

02. Количество файлов в папке

Чтобы записать результат работы скрипта в переменную, необходимо выполнить следующую запись:

```
result=$(your command)
```

Пример:

```
result=$(ls /)
```

```
echo $result
```

03. Форма логина / пароля

Для чтения ввода можно использовать следующие конструкции:

read varname # прочитать значение в переменную

read -p “Message” varname # прочитать значение + вывести Message

read -sp “Message” passvar # прочитать значение со скрытым вводом

Задача: создайте форму для ввода логина и пароля

04. Проверяемая форма пароля

Условия:

1. *if [cond]*
2. *then*
3. *<expr then>*
4. *else*
5. *<expr else>*
6. *fi*

Создайте форму пароля, которая проверяет, что пароль - secret.

05. Арифметика

Для того, чтобы производить арифметические операции над целыми числами, можно проводить конструкцию `((expression))`

1. `b=5`
2. `((b++))`
3. `echo b`
4. `a=$((b + 1))`

Напишите программу для сложения двух чисел.

Для работы с вещественными числами используется команда **`bc`**.

06. Циклы for

Цикл for

1. `x='A B C'`
2. `for var in x`
3. `do`
4. `echo $var`
5. `done`

Задача. Напишите скрипт, который позволяет пронумеровать список файлов в директории

07. Синтаксис функций

```
1.  #!/bin/bash
2.
3.  print_something () {
4.    echo Hello $1
5.  }
6.
7.  print_something Mars
8.  print_something Jupiter
```

Задача. Напишите программу, которая производит обход файловой системы по папкам.

Проверка на директорию [-d \$dir]

Локальные переменные: local x=1

Что мы узнали сегодня?

- Как устроена файловая система Linux/Unix
- Как работать с командной строкой в Linux/Unix/Mac OS
- Как писать скрипты на Bash