

Программная инженерия. Разработка ПО (Python для продвинутых специалистов. Машинное обучение)

Модуль: Введение в Python для машинного обучения

Лекция 5: Unix. База для уверенной работы

Дата: 07.04.2024

Q&A



Содержание лекции

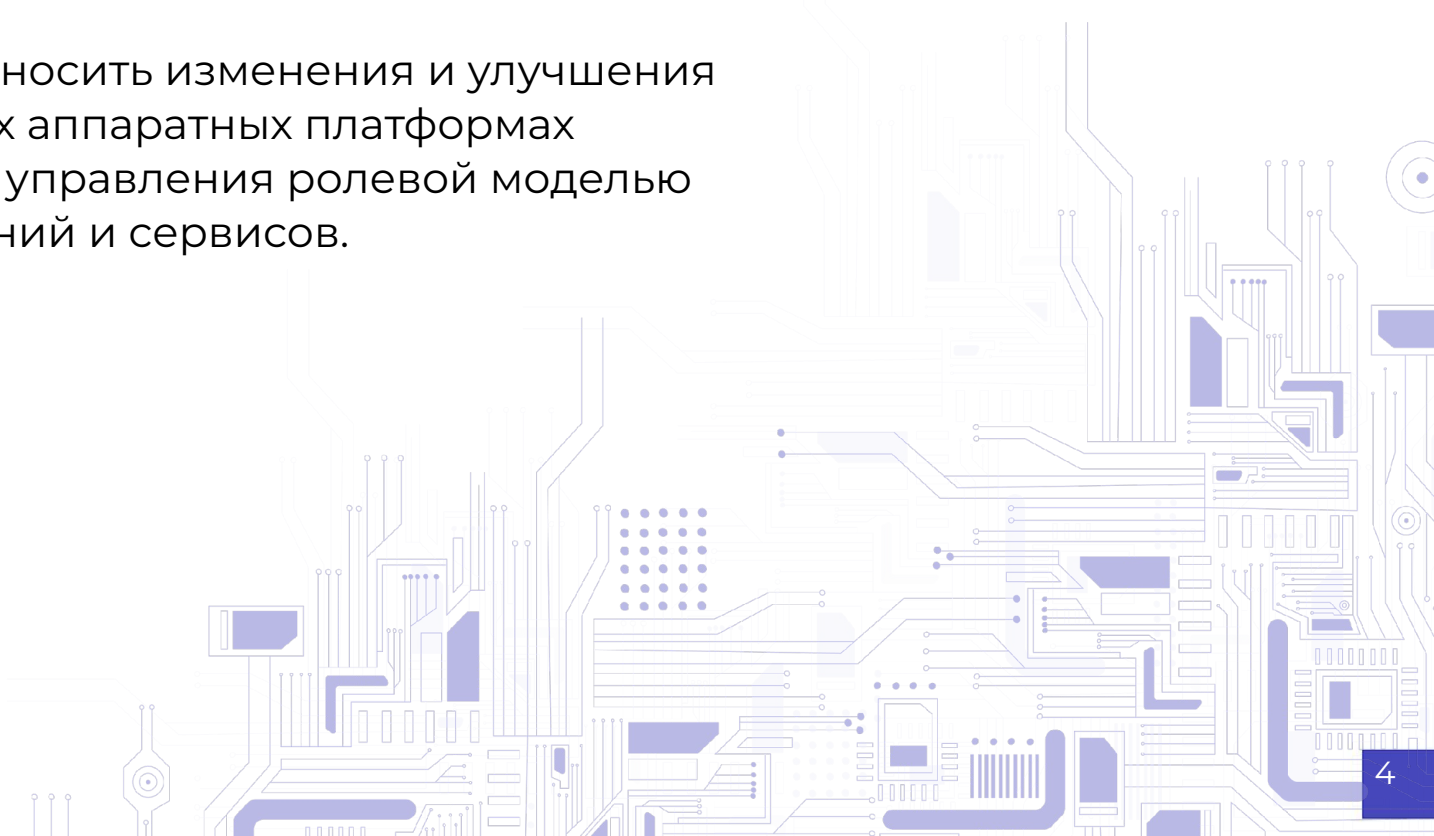
- Почему важно уметь пользоваться Unix системами?
- Джентельменский набор команд для уверенной работы
- Практическая часть



Операционные системы семейства Unix оказали огромное влияние на IT мир. Принципы и архитектура, заложенные в основу Unix стали де-факто стандартом в индустрии программного обеспечения.

Unix важен сегодня по нескольким причинам:

- **История и наследие:** основа многих современных операционных систем (Linux, macOS, Android ...)
- **Открытый исходный код:** возможность вносить изменения и улучшения
- **Масштабируемость:** работа на различных аппаратных платформах
- **Безопасность:** мощные инструменты для управления ролевой моделью
- **Совместимость** со множеством приложений и сервисов.



Как мы будем работать с Unix?

Для практики будем использовать Ubuntu (если у вас под рукой другой дистрибутив – вы можете воспользоваться им, но команды могут несколько отличаться).

Воспользуйтесь одним из трех способов:

1. **WSL – Windows Subsystem for Linux** - <https://learn.microsoft.com/ru-ru/windows/wsl/install>

WSL – это подсистема, с помощью которой можно одновременно получить доступ к возможности Windows и Linux на компьютере с Windows.

```
C:\Users\user>wsl
<3>WSL (21879 - Relay) ERROR: CreateProcessParseCommon:909: getpwuid(1000) failed 0
Welcome to Ubuntu 24.04.1 LTS (GNU/Linux 5.15.167.4-microsoft-standard-WSL2 x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

System information as of Sun Mar  9 13:53:45 MSK 2025

System load:  0.0          Processes:            40
Usage of /:   0.2% of 1006.85GB Users logged in:       1
Memory usage: 8%          IPv4 address for eth0: 172.20.204.115
Swap usage:   0%

This message is shown once a day. To disable it please create the
/root/.hushlogin file.
root@DESKTOP-46S670N:/mnt/c/Users/user# |
```

Как мы будем работать с Unix?

Для практики будем использовать Ubuntu (если у вас под рукой другой дистрибутив – вы можете воспользоваться им, но команды могут несколько отличаться).

Воспользуйтесь одним из трех способов:

2. **Docker** - <https://docs.docker.com/get-started/>

С помощью контейнеров Docker можно развернуть в локальном окружении различные сервисы без их установки напрямую на ваш ПК. Для этого установите Docker Desktop по инструкции из документации, а затем выполните команды:

- *docker pull ubuntu # скачает docker-образ Ubuntu последней версии*
- *docker run --name=ubuntu -it ubuntu # развернёт из образа контейнер, назовет его ubuntu и запустит его в интерактивном режиме*
- *docker start -i ubuntu # запустит существующий контейнер ubuntu в интерактивном режиме*

```
C:\Users\user>docker pull ubuntu
Using default tag: latest
latest: Pulling from library/ubuntu
5a7813e071bf: Download complete
Digest: sha256:72297848456d5d37d1262630108ab308d3e9ec7ed1c3286a32fe09856619a782
Status: Downloaded newer image for ubuntu:latest
docker.io/library/ubuntu:latest

C:\Users\user>docker run --name=ubuntu -it ubuntu
root@56f03efb1381:/# exit
exit

C:\Users\user>docker start -i ubuntu
root@56f03efb1381:/# |
```


Как мы будем работать с Unix?

Для практики будем использовать Ubuntu (если у вас под рукой другой дистрибутив – вы можете воспользоваться им, но команды могут несколько отличаться).

Воспользуйтесь одним из трех способов:

3. **Удаленная VM от РТ** - <https://195.19.105.128:7654>

Полноценная UNIX Red OS с предустановленным Python и необходимыми пакетами.

Затащить свои данные на машину можно, например, из гугл диска.

- *pip install gdown # установит утилиту для скачивания файлов*
- *gdown <file_id> # скачает файл по file_id (тот самый, что из url), но не забудьте выдать права!*



АРАСНЕ GUACAMOLE

Пароль

Вход

```
[seriln@vm-psql-adv ~]$ python3
Python 3.8.20 (default, Mar 4 2025, 00:00:00)
[GCC 11.4.1 20230605 (Red Soft 11.4.0-1)] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import pandas as pd
>>> exit()
```



Передовые
инженерные
школы



МИНОБРНАУКИ
РОССИИ



УНИВЕРСИТЕТ
ИННОПОЛИС



онлайн
университет

Базовые команды Unix

Навигация по файловой системе

Файловая система в Unix состоит из файлов и каталогов.

Специальные обозначения каталогов:

. # текущий каталог
.. # родительский каталог
/ # корневой каталог
~ # домашний каталог пользователя

Абсолютный путь:

/home/user/documents/doc1.txt

Относительный путь (если находимся в /home/user):

documents/doc1.txt

pwd # покажет каталог в котором вы сейчас находитесь (present working directory)

clear # очистит экран консоли

bash # включит программную оболочку bash

Задание: проверьте в каком каталоге вы находитесь

Навигация по файловой системе

<команда> [<аргумент 1> <аргумент 2> ...] *# общий формат команд*

ls [<каталог>] # выводит содержимое каталога

ls -l [<каталог>] # выводит расширенный список содержимого каталога

ls -lh [<каталог>] # выводит расширенный список содержимого каталога, но размер файлов – в «человекопонятных» единицах.

cd <каталог> # меняет текущий каталог на указанный (change directory)

Подсказки по аргументам команд:

- *man <команда> # встроенный в ОС мануал по командам*
- *www.opennet.ru (но будьте внимательны с типом операционной системы)*

Если чего-то не хватает (образ в docker может быть урезан):
apt-get install <имя пакета> # установит нужную программу

Задание: перейдите в домашний каталог вашего пользователя

Как закинуть файл в Docker и обратно

Для перемещения файла в Docker-контейнер воспользуйтесь командой:
docker cp <source file> <container>:<path to destination file>

Например:

docker cp foo.txt ubuntu:/foo.txt

Если хотите скачать файл из Docker контейнера, то просто поменяйте местами последние 2 атрибута:

docker cp ubuntu:/foo.txt foo.txt

```
C:\Users\user\Desktop\Work\Инно\Python ML\5>docker cp strange_text.txt ubuntu:/home/ubuntu  
Successfully copied 24.1kB to ubuntu:/home/ubuntu
```

```
C:\Users\user\Desktop\Work\Инно\Python ML\5>docker cp ubuntu:/home/ubuntu/strange_text.txt some_new_file.txt  
Successfully copied 24.1kB to C:\Users\user\Desktop\Work\Инно\Python ML\5\some_new_file.txt
```

Задание: скопируйте файлы из урока в домашний каталог вашего пользователя

Базовые команды с файлами каталогами

mkdir <имя каталога> # создает указанный каталог (make dir)

cp [-r] <источник> <приемник> # копирует файл [или каталог] из источника в приемник.

mv <источник> <приемник> # перемещает файл или каталог из источника в приемник. Эта же команда используется для переименования файлов и каталогов (когда указан один и тот же родительский каталог)

rm <файл> # удаляет файл

rm -r <каталог> # удаляет каталог и все его содержимое

touch <имя файла> # создает пустой файл

cat <имя файла> # выводит в консоль содержимое файла

more <имя файла> # выводит в консоль содержимое файла с возможностью прокрутки (выход - q)

head [-n <количество строк>] <имя файла> # выводит первые несколько строк файла

tail [-n <количество строк>] <имя файла> # выводит последние несколько строк файла

nano <имя файла> # открывает файл в текстовом редакторе nano

Задание:

- скопируйте файлы из 3 урока в домашний каталог вашего пользователя
- создайте новый каталог bkr и положите туда копию файла strange_text.txt
- прочитайте первые 5 строк из файла
- создайте новый файл и напишите в него что-нибудь

Перенаправление потоков и конвейер

Команды в Unix принимают на вход поток STDIN и отдают на выход потоки STDOUT и STDERR.

Перенаправление STDOUT в файл возможно указанием > (перезапись) или >> (дозапись):
head -5 file.txt > first_5_lines_from_file.txt # сохранит первые 5 строк из файла в новый файл

Поток STDOUT можно направить в качестве потока STDIN для другого пакета с помощью |
cat file.txt | wc # считывает файл и передает его в пакет wc (word count)

ls -l 2> error.log # перенаправление STDERR

ls -l 1> result.txt 2> error.log # одновременное перенаправление STDOUT и STDERR

ls -l &> all.log # одновременное перенаправление STDOUT и STDERR

Задание:

- выведите 5-ую строку из файла *strange_text.txt*
- сохраните результат в новый файл
- выполните команду, которая одновременно возвращает 2 потока (STDOUT и STDERR) и сохраните их в разные файлы

Текстовый процессинг

`grep [-i] [-v] <что искать> <где искать> # построчный поиск в тексте`
`tr <набор символов для замены> <набор символов на замену> # посимвольная замена`
`cut -d <разделитель> -f <номер поля> # выделение колонок из текстового файла`
`sed 's/<что искать>/<на что заменить>/g' <файл> # поиск с заменой`

Note: **/** - стандартный разделитель, можно заменить на другой спецсимвол

Задание:

Представьте, что у вас есть обработчик, который умеет создавать Датасеты из csv файла по продажам. Он принимает строки в формате: Дата|Сумма|Категория
Файл не должен содержать неопределенных значений.

- проверьте есть ли в файле `sales.csv` неопределенные значения
- замените разделитель в файле `sales.csv`
- уберите из файла лишние колонки
- заполните недостающие значения
- оставьте только дату, без времени
- сохраните результат в файл `for_upload.csv`

Bash-скрипты

- Как правило имеют расширение *.sh*
- В первую строку скрипта нужно помещать указание на интерпретатор (*#!/bin/bash*)
- Комментарии начинаются с *#*
- Запуск bash-скрипта: *./<имя bash-скрипта>*

Пример простейшего bash-скрипта, который возвращает строку, переданную ему в качестве параметра:

```
#!/bin/bash  
echo $1
```

\$1 – обращение к параметру, переданному в скрипт первым

Задание:

создайте скрипт *hw.sh* и попробуйте его запустить

Система прав в Unix

- Права делятся на: **чтение, запись, исполнение**.
- Назначаются для: **владельца, группы, всех** пользователей
- Простейшая команда `ls -l` покажет права на файлы
- Формат отображения прав:

`rw-rw-rwx = 777` – объект доступен всем для любых действий, опасно

`rw-r-xr-x = 755` – наиболее часто встречающаяся комбинация для исполняемых файлов

`rw-rw-r-- = 664` – стандартные права на чтение и запись

`rw-r--r-- = 644` – стандартные права на чтение и запись

`r----- = 400` – не делайте так!

`----- = 000` – и тем более так!

Задание:

выдайте права на исполнение вашего скрипта `hw.sh` и запустите его

Python-скрипт

Если python не установлен, то:

`apt-get update` *# обновит репозитории*

`apt-get install -y python3` *# скачает и установит python 3 версии*

`apt-get install python3-pandas` *# скачает и установит pandas (понадобится далее)*

Принцип создания Python-скрипта будет отличаться от bash-скрипта лишь указанием на интерпретатор.

Если задать скрипту расширение .py, то некоторые ОС подсвечивают код.

`which <имя пакета>` *# подскажет в какую директорию установлен пакет*

Задание:

Создайте простейший python скрипт и запустите его.

Не забудьте выдать права на исполнение.



Передовые
инженерные
школы



МИНОБРНАУКИ
РОССИИ



УНИВЕРСИТЕТ
ИННОПОЛИС



онлайн
университет

Практическая часть