Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение

высшего образования

«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт цифрового развития

Кафедра инфокоммуникаций

**ОТЧЕТ**

**ПО ЛАБОРАТОРНОЙ РАБОТЕ №2.14**

**дисциплины «Программирование на Python»**

**Вариант\_\_\_**

|  |  |  |  |
| --- | --- | --- | --- |
|  | | Выполнил:  Иващенко Олег Андреевич  2 курс, группа ИВТ-б-о-22-1,  09.03.02 «Информационные и вычислительные машины», направленность (профиль) «Программное обеспечение средств вычислительной техники и автоматизированных систем»  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) | |
|  | | Руководитель практики:  Воронкин Роман Александрович, доцент кафедры инфокоммуникаций  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  (подпись) | |
|  | |  | |

Отчет защищен с оценкой \_\_\_\_\_\_\_\_\_\_\_ Дата защиты\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Ставрополь, 2023 г.

**Тема**: «Установка пакетов в Python. Виртуальные окружения»

**Цель**: Приобретение навыков по работе с менеджером пакетов pip и виртуальными окружениями с помощью языка программирования Python версии 3.x.

Порядок выполнения работы

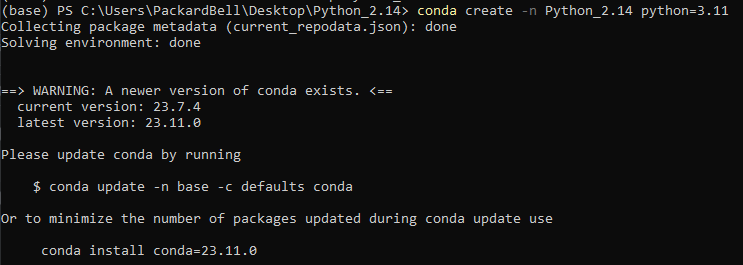


Рисунок 1 – Установка окружения conda



Рисунок 2 – Активация окружения

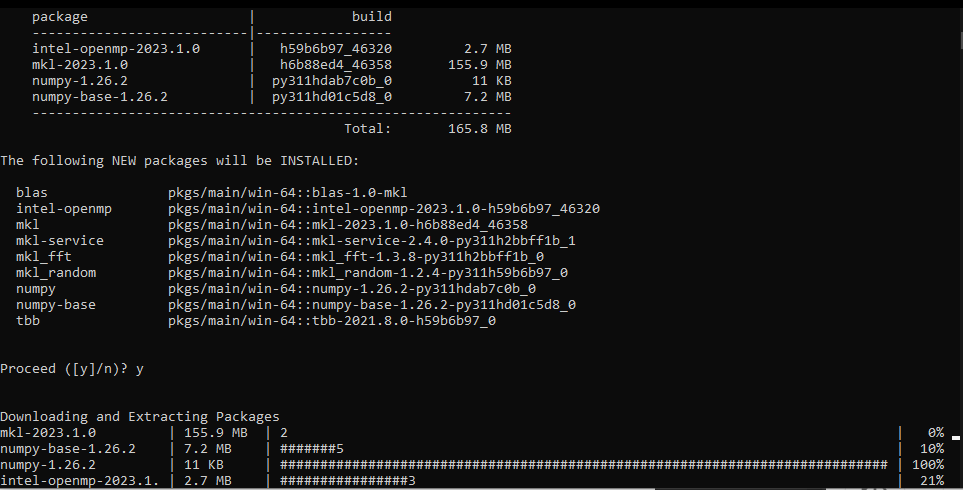


Рисунок 3 – Установка пакета NumPy

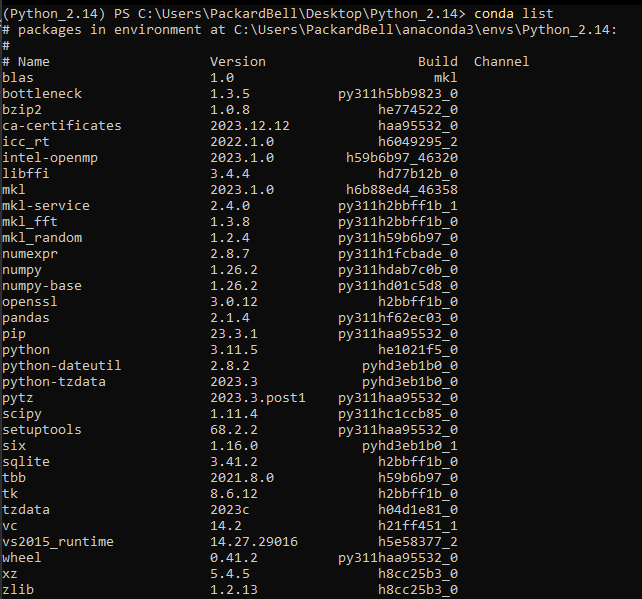


Рисунок 4 – Список установленных пакетов

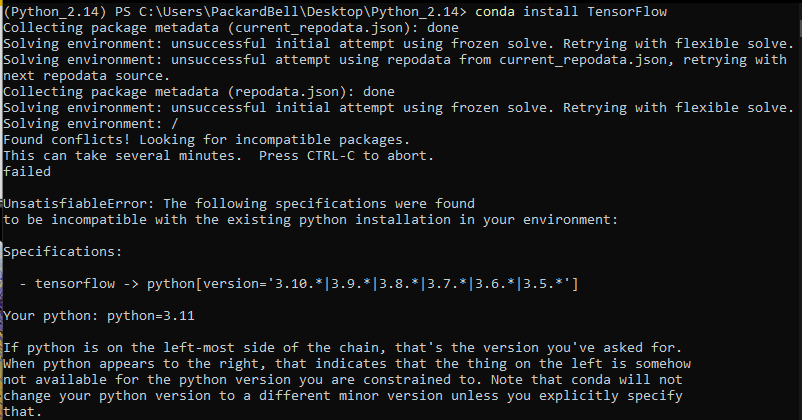


Рисунок 5 – Неудачные попытки установки пакета TensorFlow с помощью conda

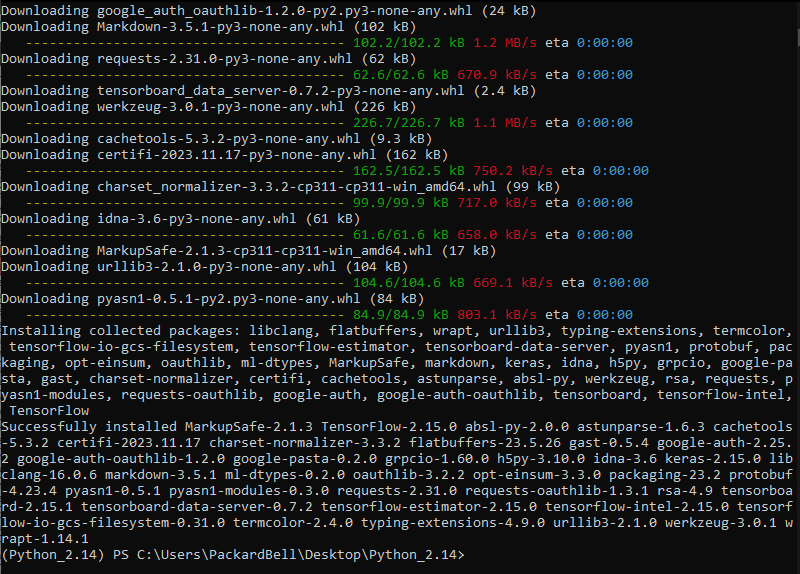


Рисунок 6 – Удачная установка TensorFlow с помощью pip

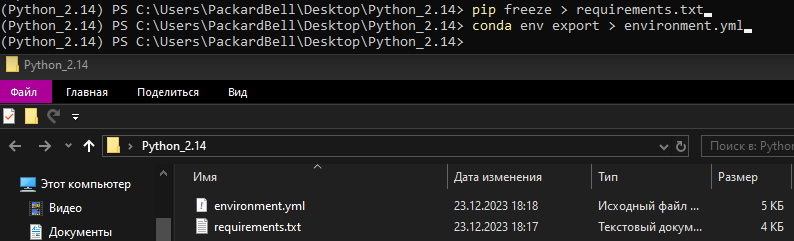


Рисунок 7 – Формирование файлов requirements.txt и environment.yml

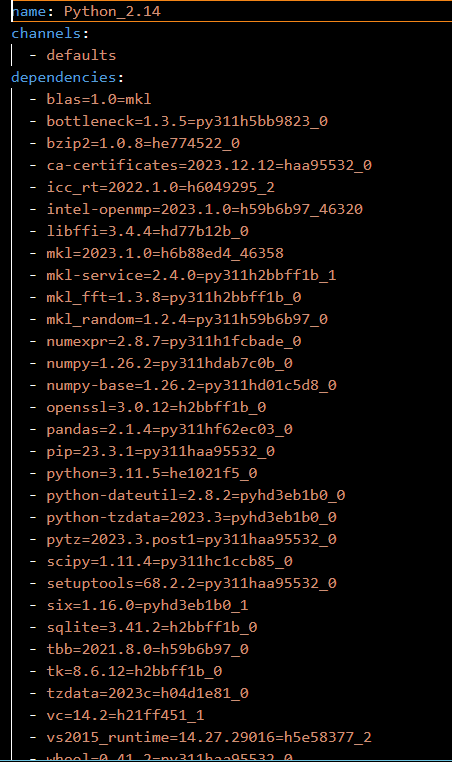


Рисунок 8 – Содержимое файла environment.yml

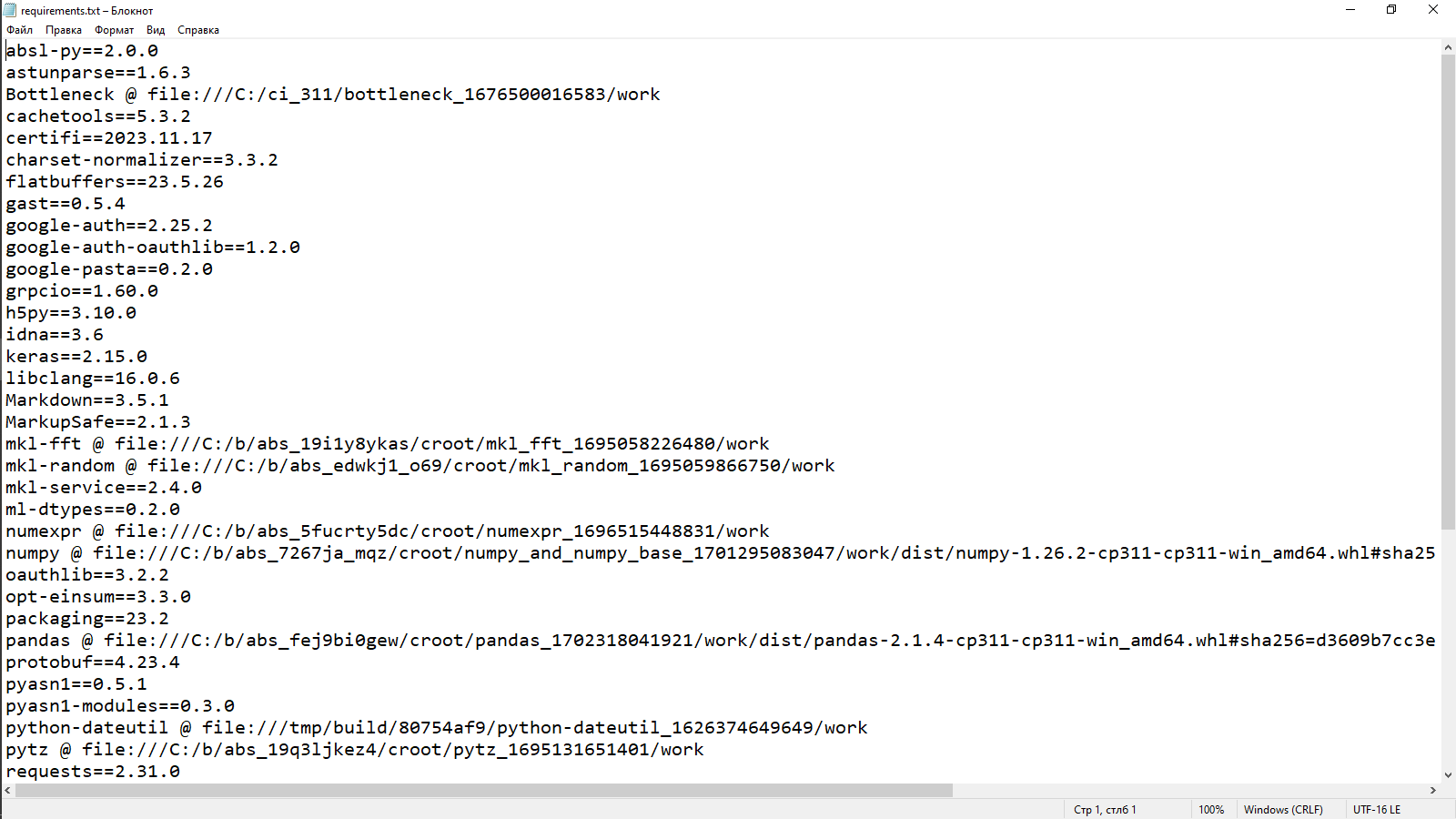


Рисунок 9 – Содержимое файла requirements.txt

Контрольные вопросы

1. Каким образом можно установить пакет Python, не входящий в стандартную библиотеку?

Можно использовать инструмент установки пакетов pip. Команда для установки пакета выглядит так:

pip install <package\_name>

1. Как осуществить установку менеджера пакетов pip?

В большинстве случаев pip поставляется вместе с Python. Однако, если его нет, его можно установить, выполнив команду:

python -m ensurepip --default-pip

1. Откуда менеджер пакетов pip по умолчанию устанавливает пакеты?

По умолчанию, pip устанавливает пакеты из Python Package Index (PyPI), который является репозиторием для пакетов Python.

1. Как установить последнюю версию пакета с помощью pip?

pip install <package\_name> --upgrade

1. Как установить заданную версию пакета с помощью pip?

pip install <package\_name>==<version>

1. Как установить пакет из git репозитория (в том числе GitHub) с помощью pip?

pip install git+<git-url>

1. Как установить пакет из локальной директории с помощью pip?

pip install <path>

1. Как удалить установленный пакет с помощью pip?

pip uninstall <package\_name>

1. Как обновить установленный пакет с помощью pip?

pip install <package\_name> --upgrade

1. Как отобразить список установленных пакетов с помощью pip?

pip list

1. Каковы причины появления виртуальных окружений в языке Python?

Виртуальные окружения позволяют изолировать зависимости и версий пакетов для конкретного проекта, предотвращая конфликты между различными проектами.

1. Каковы основные этапы работы с виртуальными окружениями?

Этапы:

* Создание виртуального окружения;
* Активация виртуального окружения;
* Установка необходимых пакетов;
* Работа с проектом внутри виртуального окружения.

1. Как осуществляется работа с виртуальными окружениями с помощью venv?

python -m venv myenv # Создание виртуального окружения

source myenv/bin/activate # Активация виртуального окружения (для Unix)

1. Как осуществляется работа с виртуальными окружениями с помощью virtualenv?

pip install virtualenv # Установка virtualenv

virtualenv myenv # Создание виртуального окружения

source myenv/bin/activate # Активация виртуального окружения (для Unix)

1. Изучите работу с виртуальными окружениями pipenv. Как осуществляется работа с виртуальными окружениями pipenv?

pip install pipenv # Установка pipenv

pipenv install # Создание и установка зависимостей

pipenv shell # Активация виртуального окружения

1. Каково назначение файла requirements.txt? Как создать этот файл? Какой он имеет формат?

Файл requirements.txt содержит список зависимостей проекта. Его можно создать вручную, перечислив зависимости и их версии. Пример формата файла:

requests==2.25.1

Flask==2.1.0

1. В чём преимущества пакетного менеджера conda по сравнению с пакетным менеджером pip?

Conda является пакетным менеджером и системой управления окружениями, позволяющим управлять зависимостями не только для Python, но и для других языков программирования. Conda также обеспечивает установку бинарных пакетов, что может быть полезным для библиотек, имеющих зависимости от сторонних библиотек.

1. В какие дистрибутивы Python входит пакетный менеджер conda?

Conda входит в дистрибутив Anaconda, Miniconda и некоторые другие дистрибутивы Python.

1. Как создать виртуальное окружение conda?

conda create --name myenv

1. Как активировать и установить пакеты в виртуальное окружение conda?

conda activate myenv

conda install <package\_name>

1. Как деактивировать и удалить виртуальное окружение conda?

conda deactivate

conda env remote --name myenv

1. Каково назначение файла environment.yml? Как создать этот файл?

Файл environment.yml в Conda используется для определения окружения, включая зависимости. его можно создать вручную, а также сгенерировать с помощью команды:

conda env export > environment.yml

1. Как создать виртуальное окружение conda с помощью файла environment.yml?

conda env create -f environment.yml

1. Самостоятельно изучите средства IDE PyCharm для работы с виртуальными окружениями conda. Опишите порядок работы с виртуальными окружениями conda в IDE PyCharm.

В PyCharm можно создать и использовать виртуальные окружения conda следующим образом:

* Открыть проект в PyCharm;
* Перейти в «File» -- «Settings» -- «Project: <project\_name>» -- «Python Interprete»;
* Нажать на иконку шестерёнки и выбрать «Add…»;
* Выбрать «Conda Environment»;
* Указать путь к интерпретатору conda и выбрать нужное окружение;
* Нажать «Ok».

1. Почему файлы requirements.txt и environment.yml должны храниться в репозитории git?

Файлы requirements.txt и environment.yml используются для описания зависимостей проекта, и их наличие в репозитории Git позволяет другим разработчикам воссоздать окружение проекта с необходимыми зависимостями. Это обеспечивает воспроизводимость окружения и согласованность зависимостей между разными средствами разработки.

**Выводы**: В процессе выполнения лабораторной работы были приобретены навыки по работе с менеджером пакетов pip и виртуальными окружениями с помощью языка программирования Python версии 3.x, было создано виртуальное окружение с помощью Anaconda, с помощью менеджера пакетов conda установлен ряд пакетов. Однако, с пакетом TenserFlow возникли проблемы при установке с помощью conda – решение заморожено. Но при помощи менеджера пакетов pip установка прошла успешно. Также были сформированы файлы environment.yml и requirements.txt. В первом файле находятся параметры окружения, нужные для восстановления окружения в любой момент. Во втором находятся пакеты и зависимости, которые были установлены на момент формирования файла.