

Создание простого Telegram-бота

1. Необходимый софт:

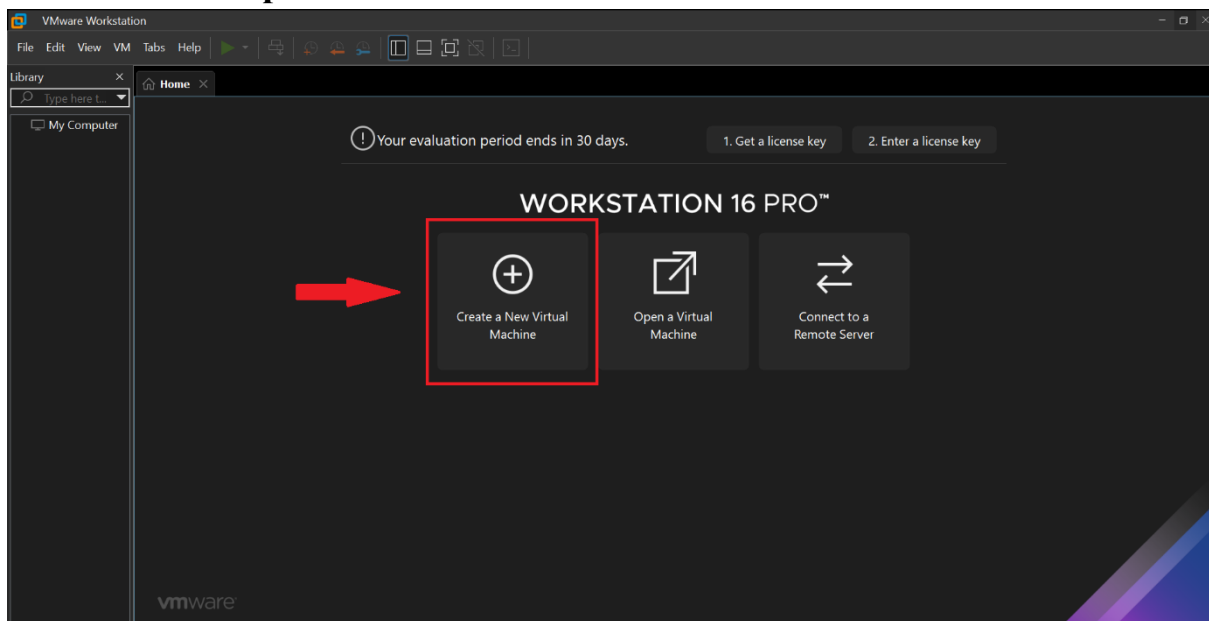
1.1. VMware Workstation Pro

<https://www.vmware.com/ru/products/workstation-pro/workstation-pro-evaluation.html>

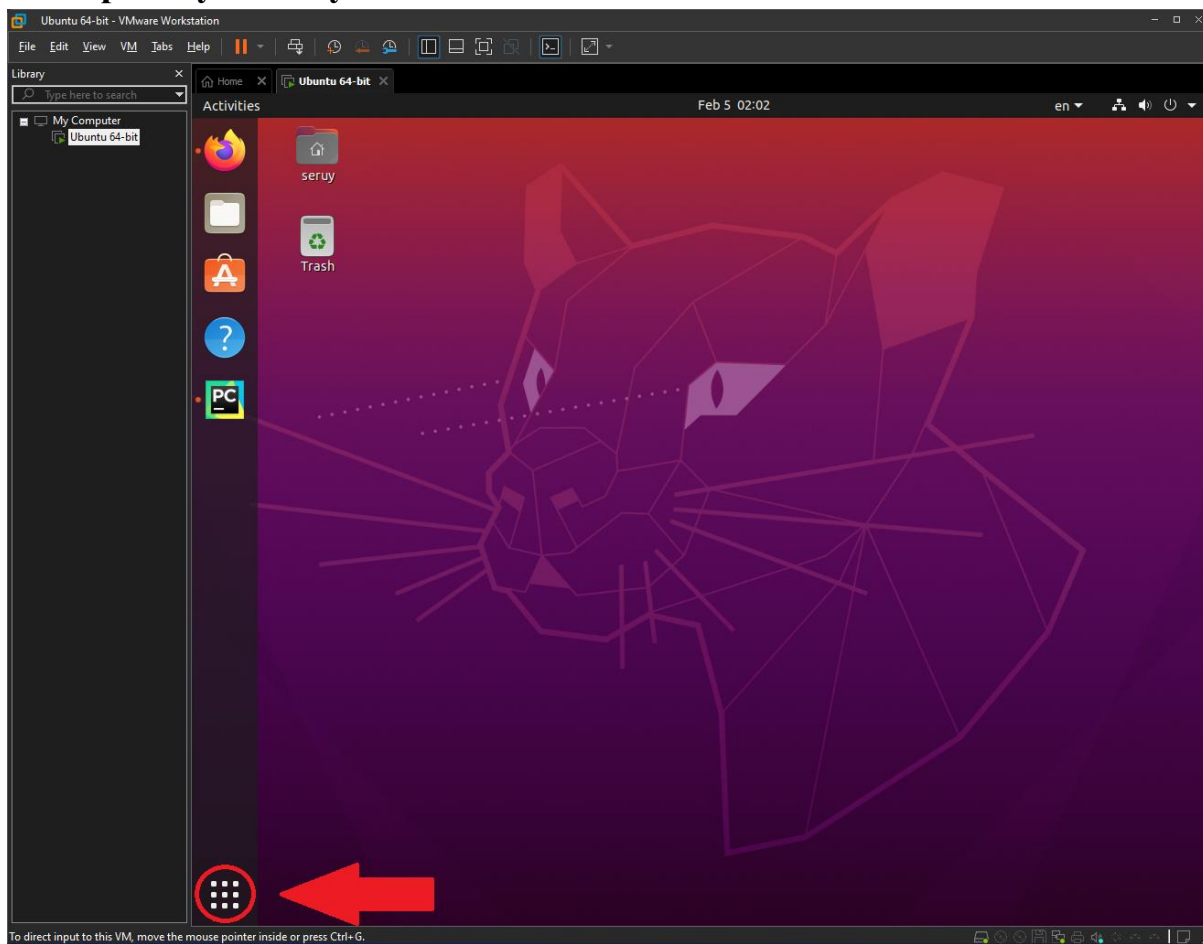
1.2. Ubuntu

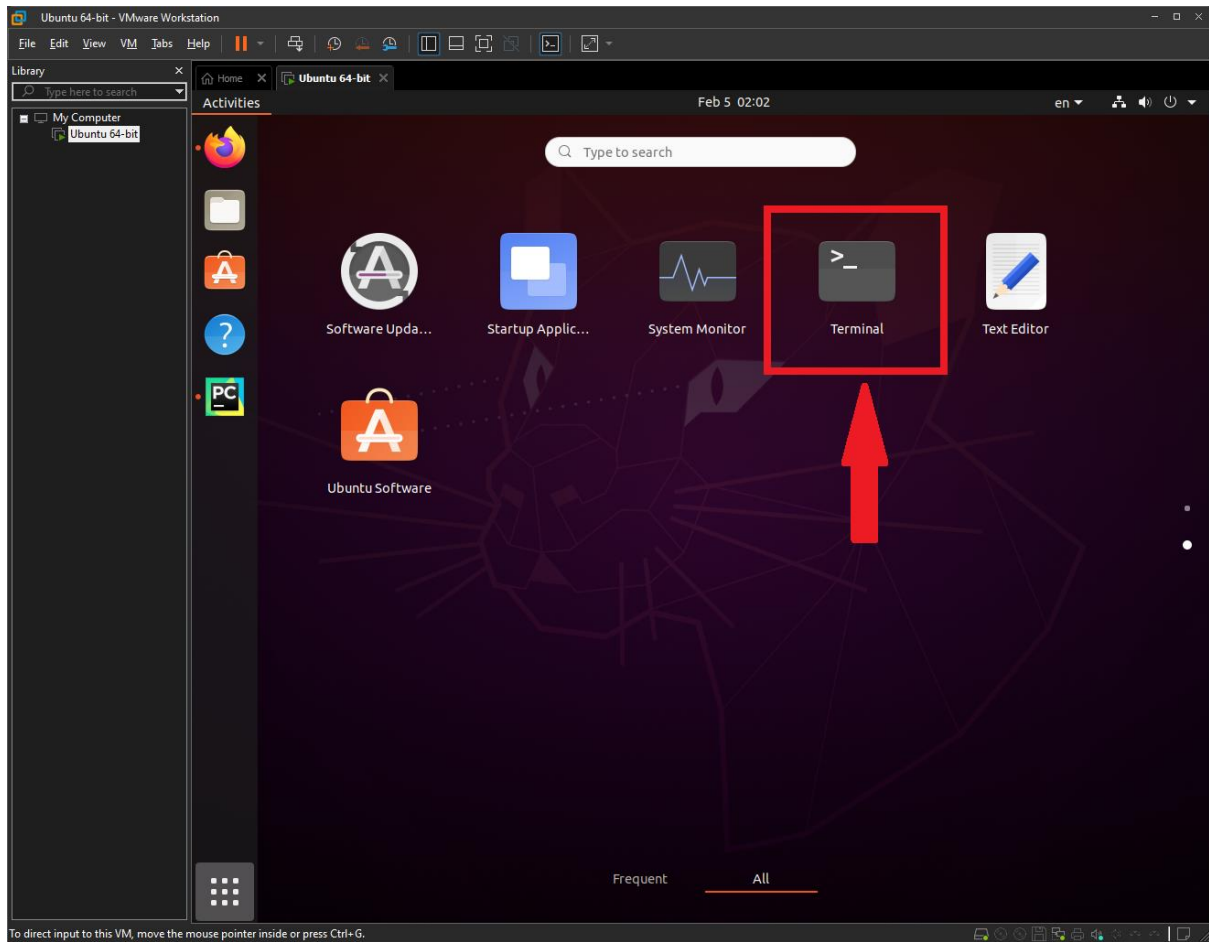
<https://ubuntu.com/>

2. Устанавливаем Ubuntu по этой кнопке, указывая образец диска, скачанный ранее с сайта



3. После установки linux ubuntu необходимо запустить терминал через эту кнопку





4. Устанавливаем Anaconda по инструкции с сайта

<https://losst.ru/ustanovka-anaconda-v-ubuntu>

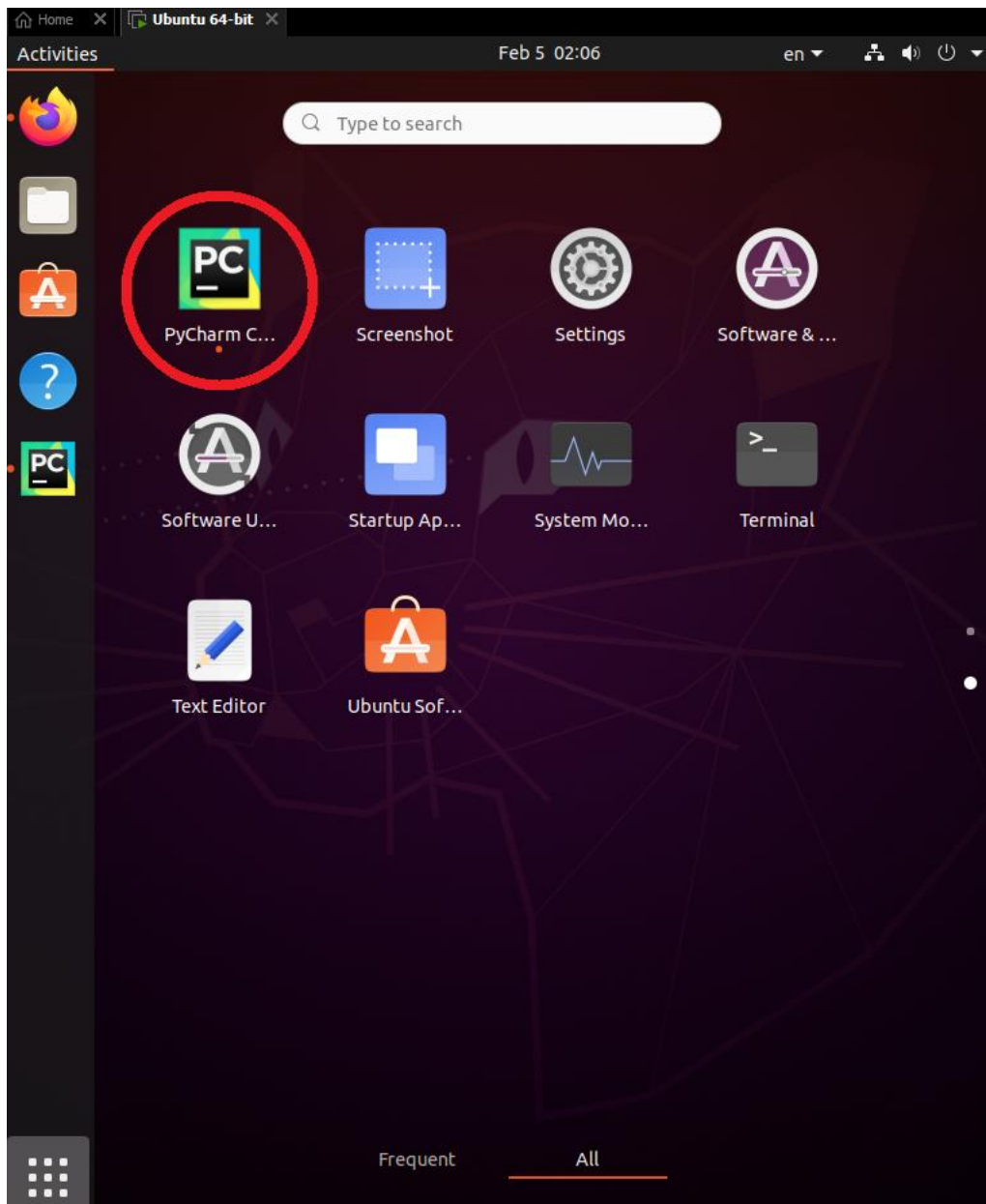
5. После успешной установки Anaconda, повторно открываем терминал в ubuntu и пишем

`sudo snap install pycharm-community --classic`

и устанавливаем среду разработки

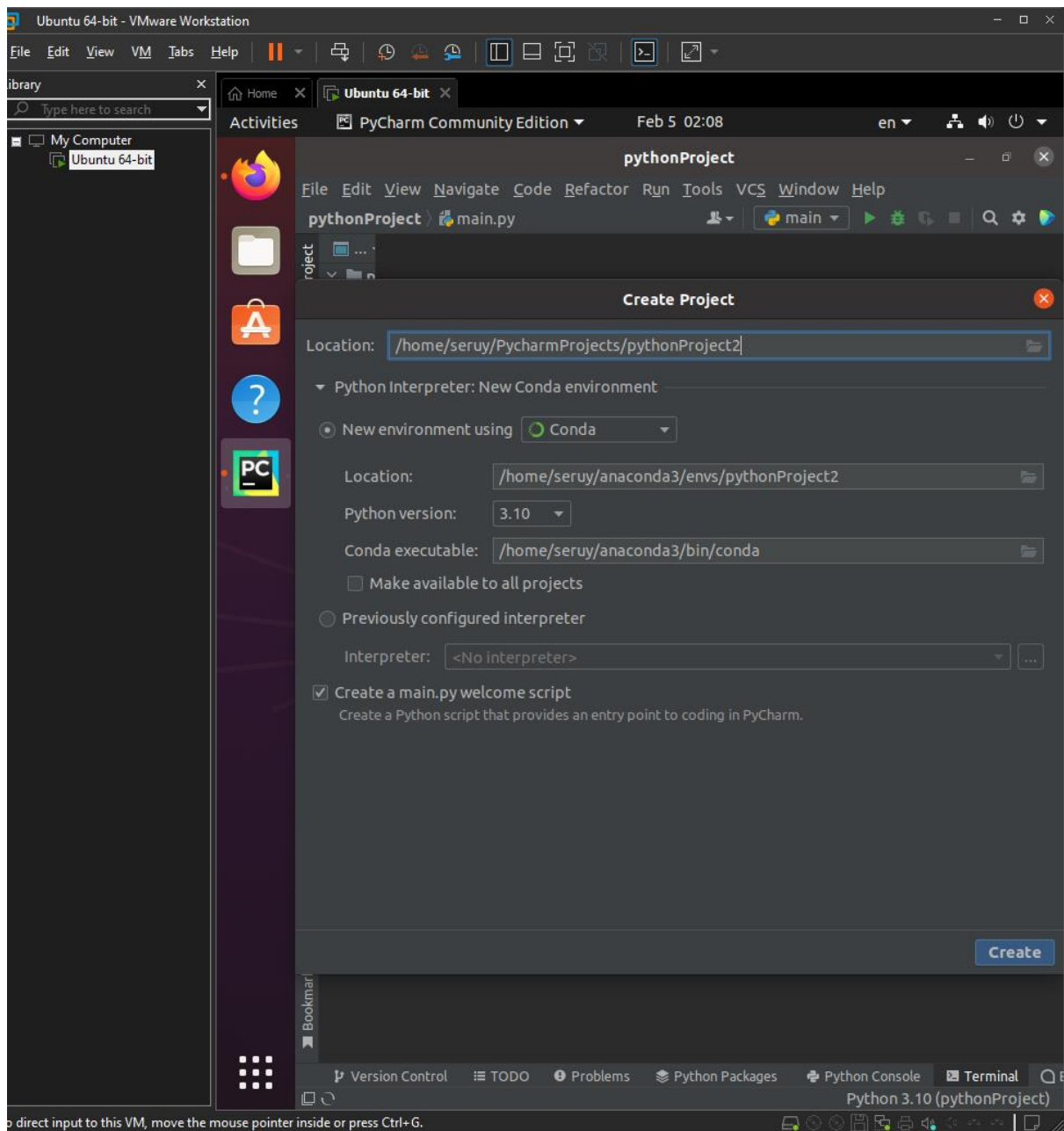
6. Вводим свой пароль пользователя

7. После установки найти во всех приложениях PyCharm



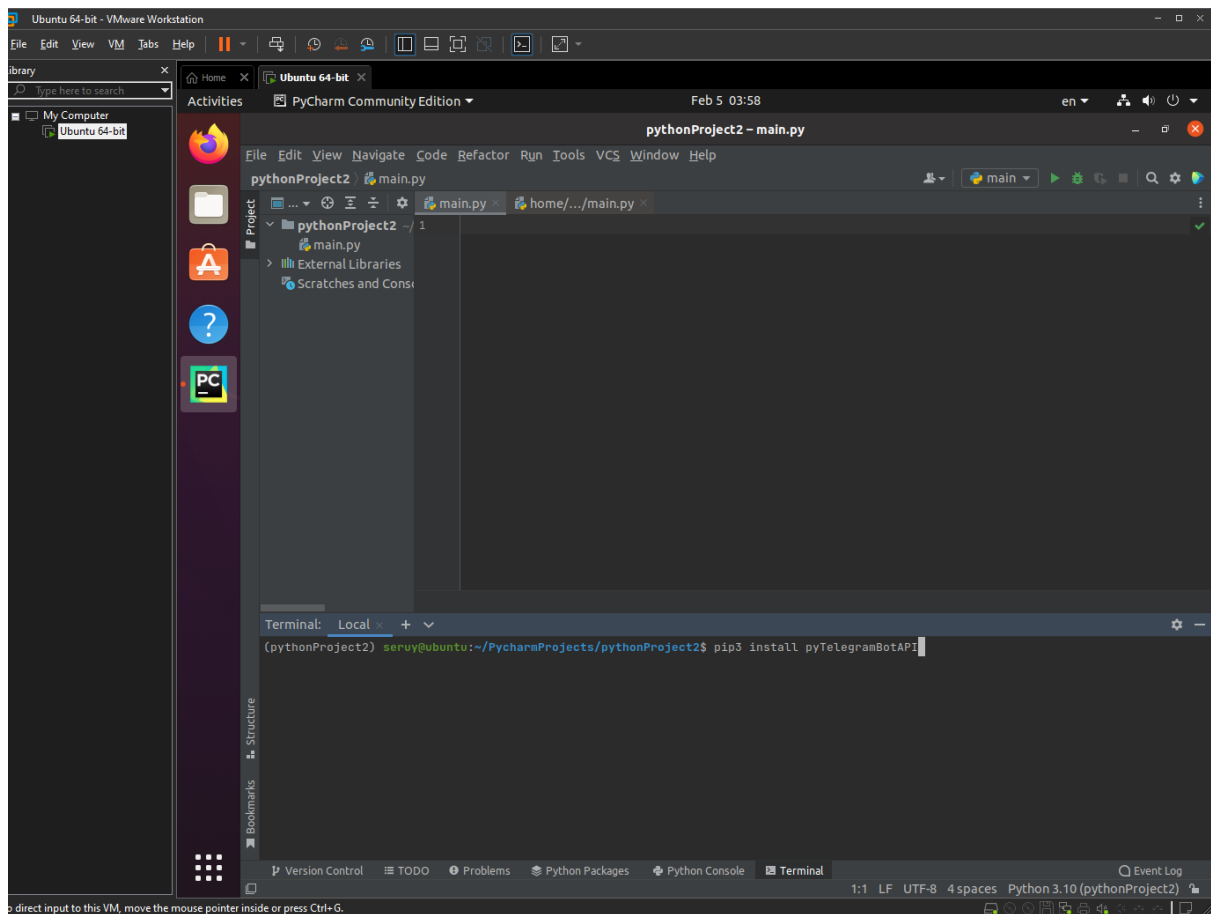
8. Создаем новый проект с именем, например, Simple-bot

Пример настроек, которые должны быть выставлены в PyCharm



(всё содержимое файла main.py можно удалить)

9. Открываем терминал в PyCharm и загружаем библиотеку для создания Telegram-ботов
pip3 install pyTelegramBotAPI



10. Регистрируем бота в Telegram

10.1. В поисковой строке ищем бота с именем @BotFather



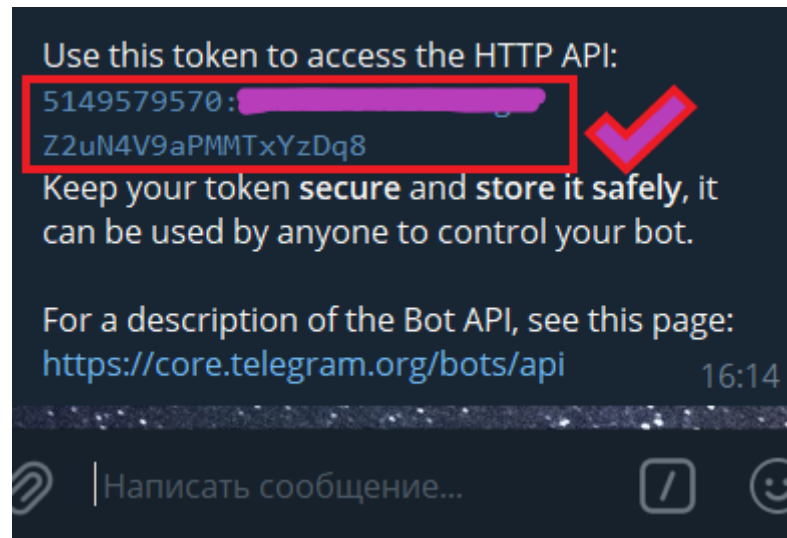


- 10.2. Отправляем боту команду/сообщение – /start
- 10.3. Отправляем боту команду/сообщение (/newbot - create a new bot) с именем бота, которого хотим зарегистрировать, в нашем случае <Ваша фамилия>_<Ваше имя>_bot («_bot» в конце **обязательно**).
- 10.4. Для ознакомления с перечнем команд @BotFather отправляем боту команду/сообщение – /help

11. В файл main.py импортируем библиотеки для создания back-end части бота

```
import telebot  
from telebot import types
```

12. Из сообщения BotFather об успешной регистрации бота копируем токен для управления нашим ботом. В файле main.py создаем переменную, хранящую в себе токен



```
token = "Ваш токен"
```

13. Создаем объект бота, к которому мы будем в дальнейшем обращаться

```
bot = telebot.TeleBot(token)
```

В класс TeleBot передаем переменную token, для того, чтобы обращаться именно к вашему боту

14. Создаем декоратор, отвечающий за команду /start

```
@bot.message_handler(commands=['start'])
def start(message):
    keyboard = types.ReplyKeyboardMarkup()
    keyboard.row("Хочу", "/help")
    bot.send_message(message.chat.id, 'Привет! Хочешь узнать
    свежую информацию о МТУСИ?', reply_markup=keyboard)
```

Класс ReplyKeyboardMarkup создает пользовательскую клавиатуру с текстовыми кнопками на месте стандартной клавиатуры.

Метод row() заполняет клавиатуру кнопками.

Метод `send_message` отправляет пользователю сообщение.

Аргумент `message.chat.id` используется для того, чтобы бот отправил сообщение тому пользователю, который отправил сообщение, на которое бот в данный момент времени отвечает.

Аргумент `reply_markup=keyboard` используется для отправки пользовательской клавиатуры, для ее дальнейшего отображения.

15. Создаем декоратор отвечающий за команду `/help`

```
@bot.message_handler(commands=['help'])
def start_message(message):
    bot.send_message(message.chat.id, 'Я умею...')
```

В сообщении вы можете указать что умеет бот, включая команды, на которые он умеет реагировать.

16. Создаем декоратор отвечающий за ответ на сообщение "Хочу"

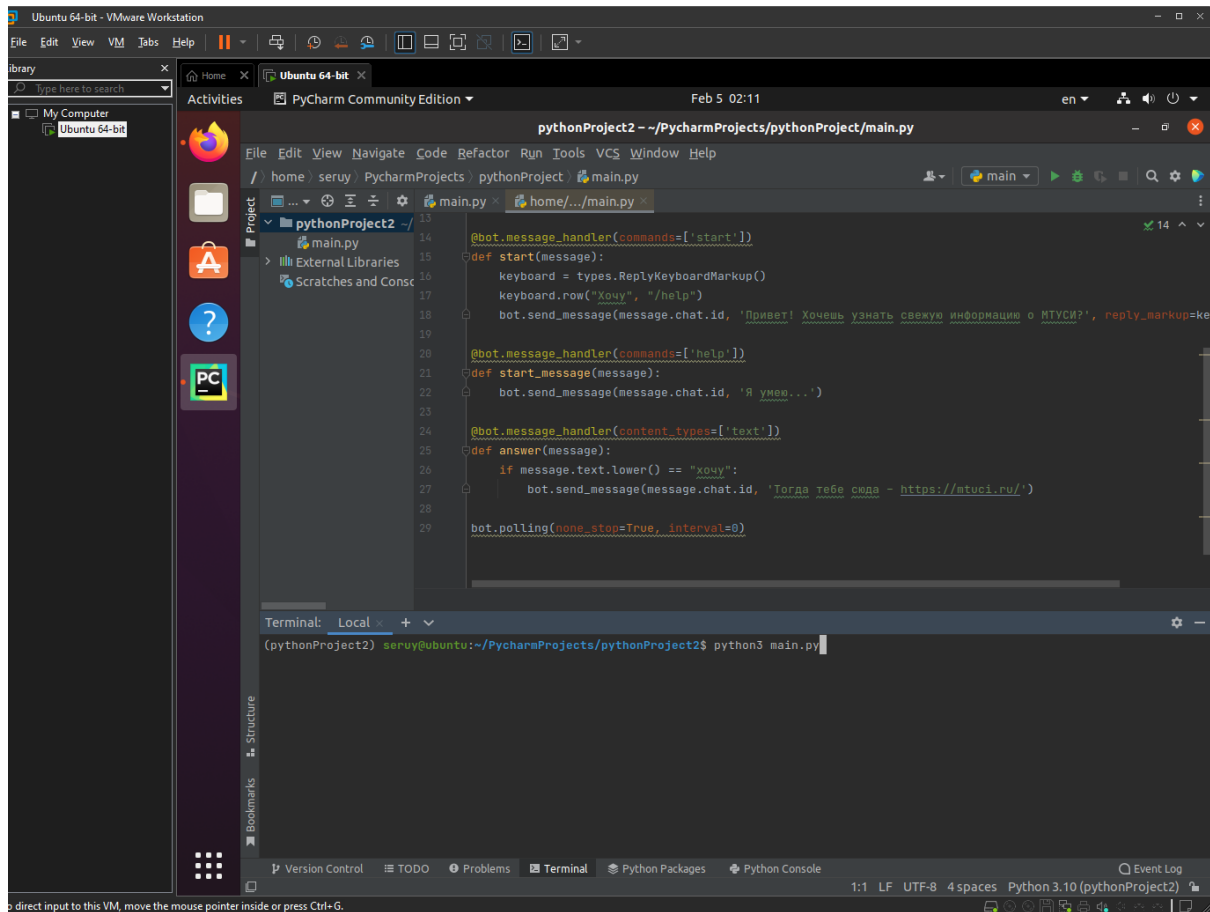
```
@bot.message_handler(content_types=['text'])
def answer(message):
    if message.text.lower() == "хочу":
        bot.send_message(message.chat.id, 'Тогда тебе сюда –
https://mtuci.ru/')

bot.polling(none_stop=True, interval=0)
```

Данный декоратор должен стоять ниже, чем декораторы команд, так как в противном случае декораторы команд обрабатываться не будут, потому что команды в своем роде тоже текстовые сообщения.

В этом декораторе аргумент `content_types=['text']` отвечает за реакцию на текстовый тип контента сообщения.

Для проверки конкретного текста используется условная конструкция с условием `message.text.lower() == "<текст>"`. Причем функция `lower()` отвечает за перевод текста в нижний регистр для удобства использования, и может применяться не только для библиотеки `telebot`, но и для любых строковых операций и переменных.



17. Открыть терминал и написать `python3 main.py`

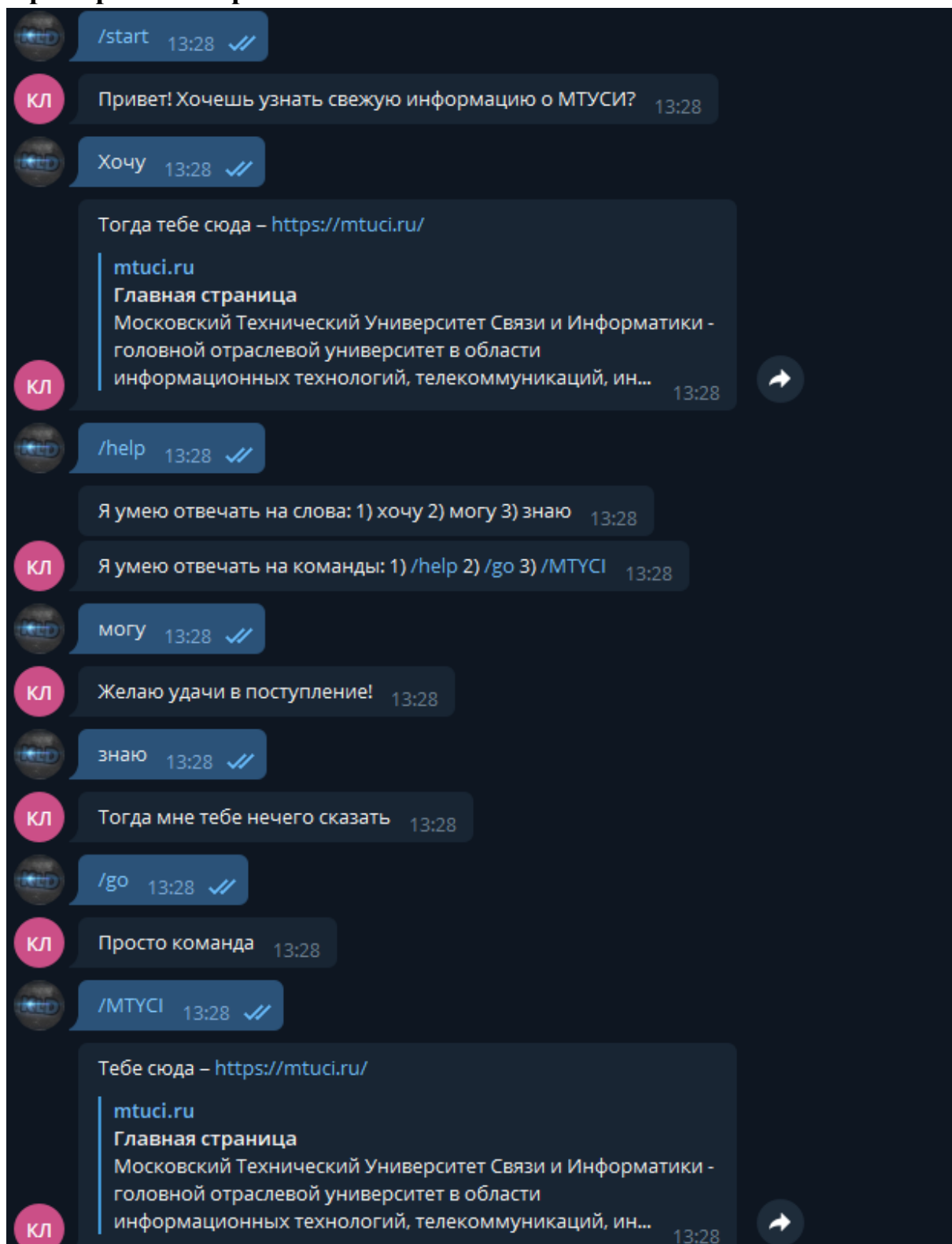
18. Нажать enter

Бот готов к использованию

Домашнее задание:

- Создать обработку трех любых сообщений
- Создать обработку трех любых команд
- Подробно расписать команду `/help`

Пример готовой работы:



Пример кода:

```
import telebot
```

```
from telebot import types
```

```
token = "51*****5:A*****"
```

```
bot = telebot.TeleBot(token)
```

```
@bot.message_handler(commands=['start'])
```

```
def start(message):
```

```
    keyboard = types.ReplyKeyboardMarkup()
```

```
    keyboard.row("Хочу", "/help")
```

```
    bot.send_message(message.chat.id, 'Привет! Хочешь узнать свежую  
информацию о МТУСИ?', reply_markup=keyboard)
```

```
@bot.message_handler(commands=['help'])
```

```
def start_message(message):
```

```
    bot.send_message(message.chat.id, 'Я умею отвечать на слова:' ' ' '1) хочу' '  
' '2) могу' ' ' '3) знаю')
```

```
    bot.send_message(message.chat.id, 'Я умею отвечать на команды:' ' ' '1)  
/help' ' ' '2) /go' ' ' '3) /MTYCI')
```

```
@bot.message_handler(commands=['go'])
```

```
def start_message(message):
```

```
    bot.send_message(message.chat.id, 'Просто команда')
```

```
@bot.message_handler(commands=['MTYCI'])
```

```
def start_message(message):
```

```
    bot.send_message(message.chat.id, 'Тебе сюда – https://mtuci.ru/)
```

```
@bot.message_handler(content_types=['text'])
```

```
def answer(message):
```

```
    if message.text.lower() == "хочу":
```

```
        bot.send_message(message.chat.id, 'Тогда тебе сюда – https://mtuci.ru/)
```

```
    if message.text.lower() == "могу":
```

```
        bot.send_message(message.chat.id, 'Желаю удачи в поступление!')
```

```
    if message.text.lower() == "знаю":
```

```
        bot.send_message(message.chat.id, 'Тогда мне тебе нечего сказать')
```

```
bot.polling(none_stop=True, interval=0)
```

Над проектом работали:

Груничев С. А.

Костенко Л. Д.

Гарцева В. Н.

Рябцев М. В.

Абдуллина К. В.