

Язык программирования



Лекция № 7

Владимир Владимирович Руцкий
rutsky.vladimir@gmail.com



План занятия

- Разбор домашнего задания
- Взаимодействие по сети. Сокеты

Компьютерные сети

- Сетевой стек. Модель OSI: https://ru.wikipedia.org/wiki/Сетевая_модель_OSI
- Протокол IP: <https://ru.wikipedia.org/wiki/IP>
Адреса узлов. Подсети. IPv4, IPv6. Маршрутизация. DNS.
- Протокол UDP: <https://ru.wikipedia.org/wiki/UDP>
Негарантированная доставка сообщений (дейтаграмм).
- Протокол TCP: <https://ru.wikipedia.org/wiki/TCP/IP>
Установление соединения. Плавающее окно передачи данных.

Сокеты

- **Сокет** — интерфейс для взаимодействия между процессами
- Существуют *клиентские* и *серверные* сокеты
- Предоставляет файло-подобный интерфейс для передачи данных между клиентом и сервером.

[https://ru.wikipedia.org
/wiki/Сокет_\(программный_интерфейс\)](https://ru.wikipedia.org/w/index.php?title=%D0%A1%D0%BE%D0%BA%D0%BE%D0%B6%D0%BD%D0%BE%D1%82%D0%BE%D0%BC_%28%D0%BF%D1%80%D0%BB%D0%BC%D0%BC%D0%MM%D0%BD%D0%BD%D0%BE%D0%BC%D0%BF%D0%B5%D1%80%D0%BD%D0%BE%D0%BC%D0%BF%D1%8F%29)

Сокеты в Python. Сервер

```
# Echo server program
import socket

HOST = ''                      # Слушать все интерфейсы
PORT = 50007                    # Порт, который необходимо слушать
# Создаём интернет-сокет для использования по протоколу TCP
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
# Будем слушать все интерфейсы на порту PORT
s.bind((HOST, PORT))
# Переводим сокет в режим ожидания соединений
s.listen(1)
# Ждём соединения от клиента
conn, addr = s.accept()
print('Connected by', addr)
while True:
    # Считываем данные от клиента
    data = conn.recv(1024)
    if not data:
        break
    # Пишем считанные данные обратно клиенту
    conn.sendall(data)
# Закрываем соединение
conn.close()
```

Сокеты в Python. Клиент

```
# Echo client program
import socket

HOST = 'localhost'          # Имя сервера, к которому будем коннектиться
PORT = 50007                 # Номер порта на сервере
# Создаём интернет-сокет для использования по протоколу TCP
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
# Соединяемся с сервером
s.connect((HOST, PORT))
# Посыпаем серверу байтовую строку
s.sendall(b'Hello, world')
# Принимаем от сервера данные
data = s.recv(1024)
# Закрываем соединение
s.close()
print('Received', repr(data))
```

Документация: <https://docs.python.org/3/library/socket.html>

SocketServer

```
import socketserver

class MyTCPHandler(socketserver.BaseRequestHandler):
    """
        На каждое соединение клиента будет создан экземпляр данного класса
        и вызван метод handle().
    """

    def handle(self):
        # self.request это сокет, соединённый с клиентом
        self.data = self.request.recv(1024).strip()
        print("{} wrote:".format(repr(self.client_address)))
        print(self.data)
        # Отправим клиенту те же данные, что он прислал, только в верхнем
        # регистре
        self.request.sendall(self.data.upper())

    if __name__ == "__main__":
        HOST, PORT = "localhost", 9999

        # Создаём сервер, который будет слушать (HOST, PORT)
        server = socketserver.TCPServer((HOST, PORT), MyTCPHandler)

        # Запускаем сервер. Он будет работать "вечно" (пока не прервут по Ctrl+C)
        server.serve_forever()
```

Документация: <https://docs.python.org/3/library/socketserver.html>

Чат на Python