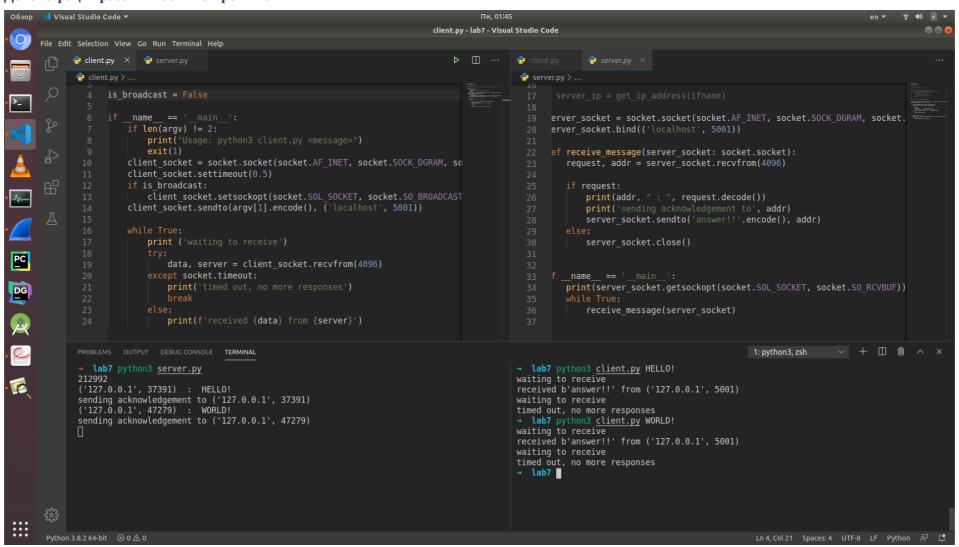
## Отчет по лабораторной № 7

## Задание Зб

Было написано небольшое консольное приложение на python. Клиент отправляет прослушивающему на 127.0.0.1:5001 серверу заданное сообщение и получает ответ.

Демонстрация работы в обычном режиме



212992 - размер максимальной дейтаграммы, которую можно отправить, в linux можно получить с помощью

```
socket.getsockopt(socket.SOL_SOCKET, socket.SO_RCVBUF)
```

Отправленные пакеты можно посмотреть в дампе lab7.pcapng (5 и 6 пакеты - сообщение HELLO! и ответ на него, 13 и 14 пакеты - сообщение WORLD! и ответ на него)

## Задание 4а

Аналогично и для задания 4a было написано консольное приложение для обмена MULTICAST сообщениями.

Демонстрация работы приложения

```
client_multicast.py - lab7 - Visual Studio Code
e client_multicast.py ×
                                                                                                           ▶ 🗓 ··· 🕏 server_multicast.py >
                                                                                                                                    multicast_group = '224.3.29.71
server_address = ('', 5001)
                        sock.settimeout(0.5)
                                                                                                                                     sock.bind(server address)
                       sock.setsockopt(socket.IPPROTO_IP, socket.IP_MULTICAST_TTL, ttl)
                                                                                                                                    group = socket.inet_aton(multicast_group)
                                                                                                                                    mreq = struct.pack('4sL', group, socket.INADDR_ANY)
sock.setsockopt(socket.IPPROTO_IP, socket.IP_ADD_MEMBERSHIP, mreq)
                            print(f'sending {message}')
sent = sock.sendto(message.encode(), multicast_group)
                                                                                                                                         print(f'received {len(data)} bytes from {address}')
print(data)
DG
                                                                                                                                         print('sending acknowledgement to', address)
sock.sendto('answer!!'.encode(), address)
                                      print('timed out, no more responses')
6
                 PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
                                                                                                                                                                                             → lab7 python3 server_multicast.py
received 6 bytes from ('192.168.0.101', 57817)
                                                                                                                                → lab7 python3 client_multicast.py
sending HELLO!
waiting to receive
received b'answer!!' from ('192.168.0.101', 5001)
                 sending acknowledgement to ('192.168.0.101', 57817)
                                                                                                                                 waiting to receive timed out, no more responses
```

В файле lab7\_multicast.pcapng можно увидеть как с помощью IGMP пакетов 7 и 9 сервер покидает группу 224.3.29.7 (тк сервер перезапускался) и затем вначале своей работы присоединяется к этой же группе для всех интерфейсов - пакеты 16 и 19. Это действие выполняется в следующем фрагменте кода:

```
group = socket.inet_aton('224.3.29.71')
mreq = struct.pack('4sL', group, socket.INADDR_ANY)
sock.setsockopt(socket.IPPROTO_IP, socket.IP_ADD_MEMBERSHIP, mreq)
```

У клиента в заголовке IP пакета выставляется TTL=1, чтобы гарантировать, что пакет гарантированно не выйдет во внешнюю сеть. Это выполняется в:

```
ttl = struct.pack('b', 1)
sock.setsockopt(socket.IPPROTO_IP, socket.IP_MULTICAST_TTL, ttl)
```

Исходный код на гитхабе