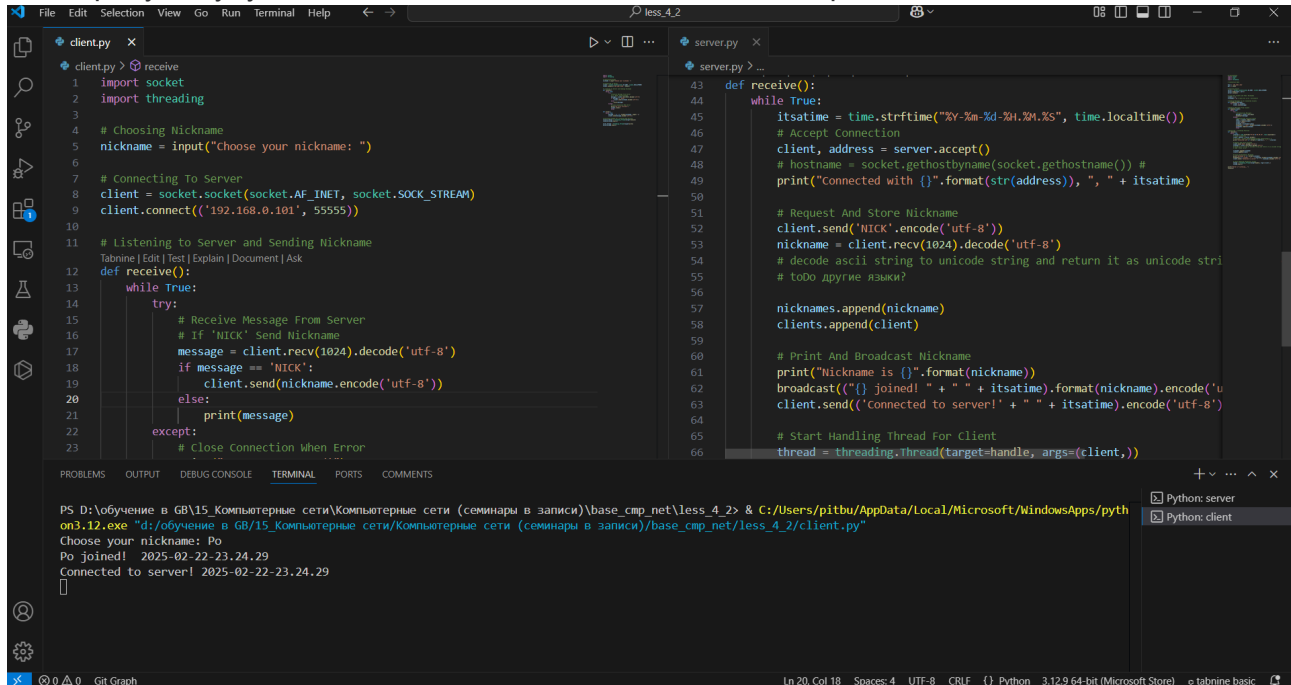


Урок 6. Основы компьютерных сетей. Транспортный уровень. UDP и TCP.

1. Напишите свою программу сервер и запустите её (если опыта в python нет, запустите готовый код и разберитесь, как он работает — [файл с кодом готового клиента](#), [файл с кодом готового сервера](#)).

*** попробуйте улучшить код, опишите что сделали, какие фишки добавили.

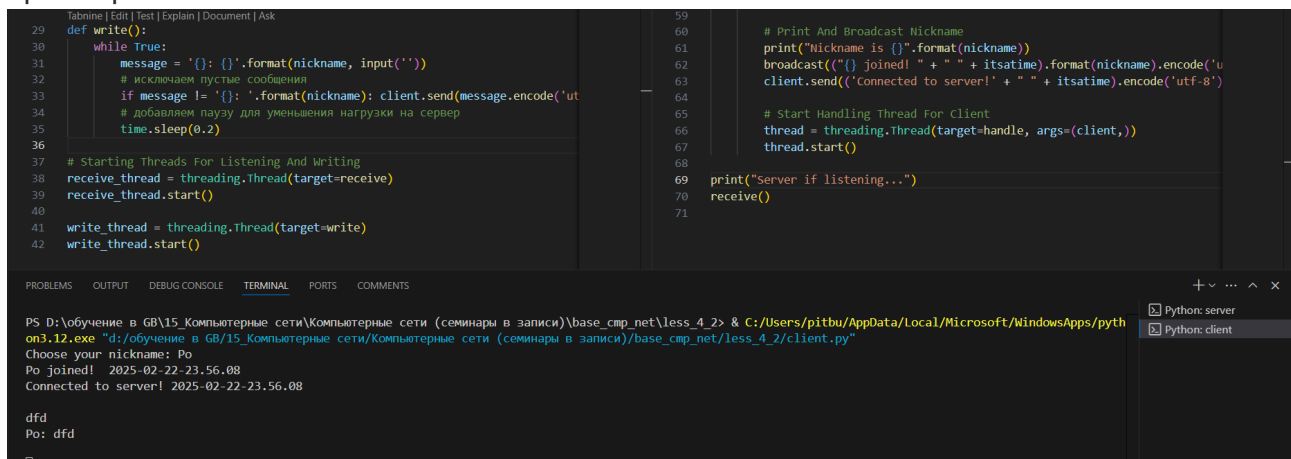


```
client.py
1 import socket
2 import threading
3
4 # Choosing Nickname
5 nickname = input("Choose your nickname: ")
6
7 # Connecting To Server
8 client = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
9 client.connect(('192.168.0.101', 55555))
10
11 # Listening to Server and Sending Nickname
12 def receive():
13     while True:
14         try:
15             # Receive Message From Server
16             # If 'NICK' Send Nickname
17             message = client.recv(1024).decode('utf-8')
18             if message == 'NICK':
19                 client.send(nickname.encode('utf-8'))
20             else:
21                 print(message)
22         except:
23             # Close Connection When Error
```

```
server.py
43 def receive():
44     while True:
45         itsatime = time.strftime("%Y-%m-%d-%H.%M.%S", time.localtime())
46         # Accept Connection
47         client, address = server.accept()
48         # hostname = socket.gethostname(socket.gethostname()) #
49         print("Connected with {}".format(str(address)), ", " + itsatime)
50
51 # Request And Store Nickname
52 client.send('NICK'.encode('utf-8'))
53 nickname = client.recv(1024).decode('utf-8')
54 # decode ascii string to unicode string and return it as unicode str
55 # todo другие языки?
56
57 nicknames.append(nickname)
58 clients.append(client)
59
60 # Print And Broadcast Nickname
61 print("Nickname is {}".format(nickname))
62 broadcast("{} joined! " + " " + itsatime).format(nickname).encode('u
63 client.send('connected to server!' + " " + itsatime).encode('utf-8')
64
65 # Start Handling Thread For Client
66 thread = threading.Thread(target=handle, args=(client,))
```

```
PS D:\обучение в GB\15_Компьютерные сети\Компьютерные сети (семинары в записи)\base_cmp_net\less_4_2> & C:\Users\pitbu\AppData\Local\Microsoft\WindowsApps\python3.12.exe "d:/обучение в GB/15_Компьютерные сети/Компьютерные сети (семинары в записи)/base_cmp_net/less_4_2/client.py"
Choose your nickname: Po
Po joined! 2025-02-22-23.24.29
Connected to server! 2025-02-22-23.24.29
```

Добавил вывод времени, поменял кодировку, исключил пустые сообщения, добавил паузу при отправке



```
client.py
29 def write():
30     while True:
31         message = '{}: {}'.format(nickname, input(''))
32         # исключаем пустые сообщения
33         if message != '{}: {}'.format(nickname): client.send(message.encode('utf-8'))
34         # добавляем паузу для уменьшения нагрузки на сервер
35         time.sleep(0.2)
36
37 # Starting Threads For Listening And Writing
38 receive_thread = threading.Thread(target=receive)
39 receive_thread.start()
40
41 write_thread = threading.Thread(target=write)
42 write_thread.start()
```

```
server.py
59 # Print And Broadcast Nickname
60 print("Nickname is {}".format(nickname))
61 broadcast("{} joined! " + " " + itsatime).format(nickname).encode('u
62 client.send('connected to server!' + " " + itsatime).encode('utf-8')
63
64 # Start Handling Thread For Client
65 thread = threading.Thread(target=handle, args=(client,))
66 thread.start()
67
68
69 print("Server if listening...")
70 receive()
71
```

```
PS D:\обучение в GB\15_Компьютерные сети\Компьютерные сети (семинары в записи)\base_cmp_net\less_4_2> & C:\Users\pitbu\AppData\Local\Microsoft\WindowsApps\python3.12.exe "d:/обучение в GB/15_Компьютерные сети/Компьютерные сети (семинары в записи)/base_cmp_net/less_4_2/client.py"
Choose your nickname: Po
Po joined! 2025-02-22-23.56.08
Connected to server! 2025-02-22-23.56.08
```

```
def write():
    while True:
        message = '{}: {}'.format(nickname, input(''))
        # исключаем пустые сообщения
        if message != '{}: {}'.format(nickname): client.send(message.encode('utf-8'))
        # добавляем паузу для уменьшения нагрузки на сервер
        time.sleep(0.2)
```

```
def receive():
    while True:
        itsatime = time.strftime("%Y-%m-%d-%H.%M.%S", time.localtime())
        # Accept Connection
        client, address = server.accept()
```

```

# hostname = socket.gethostname(socket.gethostname()) #
print("Connected with {}".format(str(address)), ", " + itsatime)

# Request And Store Nickname
client.send('NICK'.encode('utf-8'))
nickname = client.recv(1024).decode('utf-8')
# decode ascii string to unicode string and return it as unicode string with
optional encoding
nicknames.append(nickname)
clients.append(client)

# Print And Broadcast Nickname
print("Nickname is {}".format(nickname))
broadcast(("{} joined! " + " " + itsatime).format(nickname).encode('utf-8'))
client.send(('Connected to server!' + " " + itsatime).encode('utf-8'))

# Start Handling Thread For Client
thread = threading.Thread(target=handle, args=(client,))
thread.start()

```

2. Запустите несколько клиентов. Сымитируйте чат.

3. Отправьте мне код написанного сервера (можете через github, если удобно или прямо здесь в txt формате) и скриншоты работающего чата.

```

client.py
1 import socket
2 import threading
3
4 # Choosing Nickname
5 nickname = input("Choose your nickname: ")
6
7 # Connecting To Server
8 client = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
9 client.connect(('192.168.0.101', 55555))
10
11 # Listening to Server and Sending Nickname
12 def receive():
13     while True:
14         try:
15             # Receive Message From Server
16             # If 'NICK' Send Nickname
17             message = client.recv(1024).decode('ascii')
18             if message == 'NICK':
19                 client.send(nickname.encode('ascii'))
20             else:
21                 print(message)
22         except:
23             # Close Connection When Error

```

```

server.py
4 import threading
5
6 # Connection Data
7 host = '192.168.0.101'
8 port = 55555
9
10 # Starting Server
11 server = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
12 server.bind((host, port))
13 server.listen()
14
15 # Lists For Clients and Their Nicknames
16 clients = []
17 nicknames = [] # todo: попробовать dictionary?
18
19 # Sending Messages To All Connected Clients
20 def broadcast(message):
21     for client in clients:
22         client.send(message)
23
24 # Handling Messages From Clients

```

```

Server if listening...
Connected with ('192.168.0.101', 57452)
Nickname is Petr
Connected with ('192.168.0.101', 57452)
Nickname is Petr
Connected with ('192.168.0.101', 57459)
Nickname is Petr
Connected with ('192.168.0.101', 57459)
Connected with ('192.168.0.101', 57459)
Nickname is Nick

```

4. Отследите сокеты с помощью команды netstat (тоже пришлите скриншот именно сокетов вашего чата).

