САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ
ФАКУЛЬТЕТ ИНФОКОММУНИКАЦИОННЫХ ТЕХНОЛОГИЙ

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 1

«Работа с сокетами»

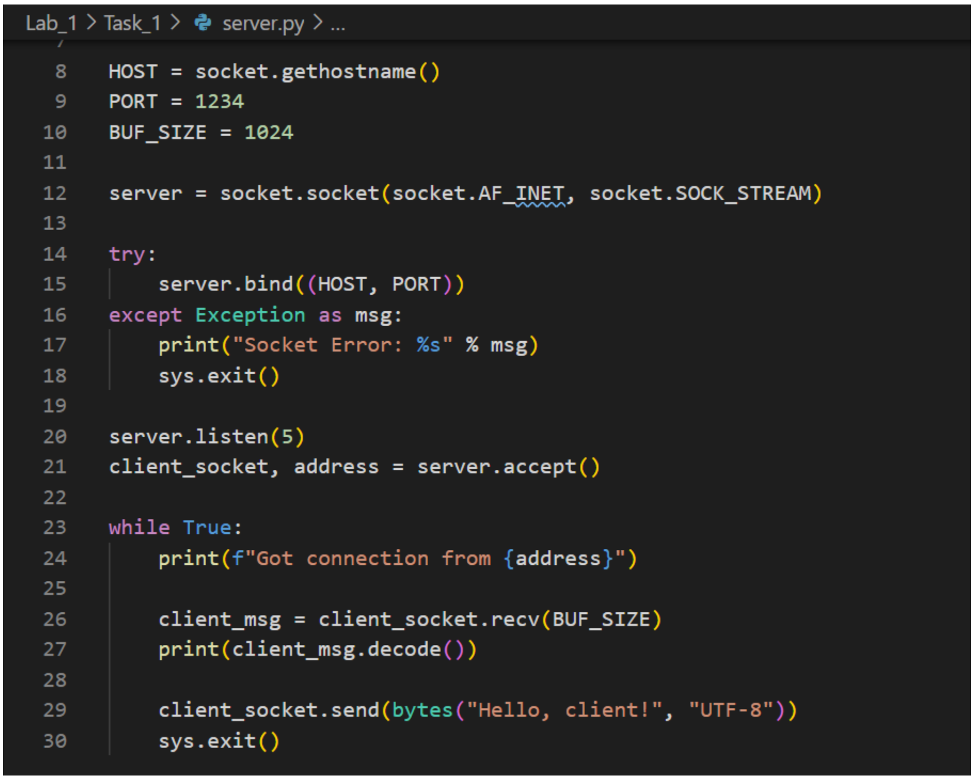
Выполнил:
Нгуен Тхюи Чанг
Группа К33402
Преподаватель:
Говоров А.И.

2020
г. Санкт-Петербург

1. **Цель работы:** Овладеть практическими навыками и умениями реализации web-серверов и использования сокетов.

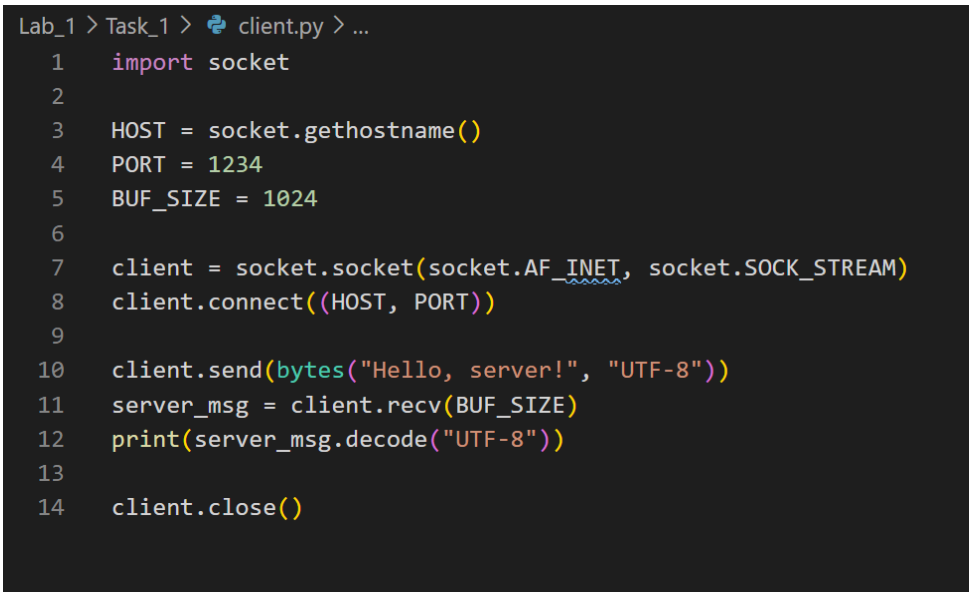
2. Индивидуальное задание

2.1. Реализовать клиентскую и серверную часть приложения. Клиент отсылает серверу сообщение «Hello, server». Сообщение должно отразиться на стороне сервера. Сервер в ответ отсылает клиенту сообщение «Hello, client». Сообщение должно отобразиться у клиента.



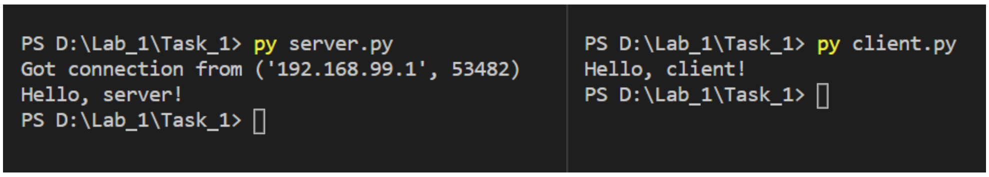
```python
HOST = socket.gethostname()
PORT = 1234
BUF_SIZE = 1024

server = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

try:
    server.bind((HOST, PORT))
except Exception as msg:
    print("Socket Error: %s" % msg)
    sys.exit()

server.listen(5)
client_socket, address = server.accept()

while True:
    print(f"Got connection from {address}")

    client_msg = client_socket.recv(BUF_SIZE)
    print(client_msg.decode())

    client_socket.send(bytes("Hello, client!", "UTF-8"))
    sys.exit()
```

Figure 1 - Code for server-side



```python
import socket

HOST = socket.gethostname()
PORT = 1234
BUF_SIZE = 1024

client = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
client.connect((HOST, PORT))

client.send(bytes("Hello, server!", "UTF-8"))
server_msg = client.recv(BUF_SIZE)
print(server_msg.decode("UTF-8"))

client.close()
```

Figure 2 - Code for client-side

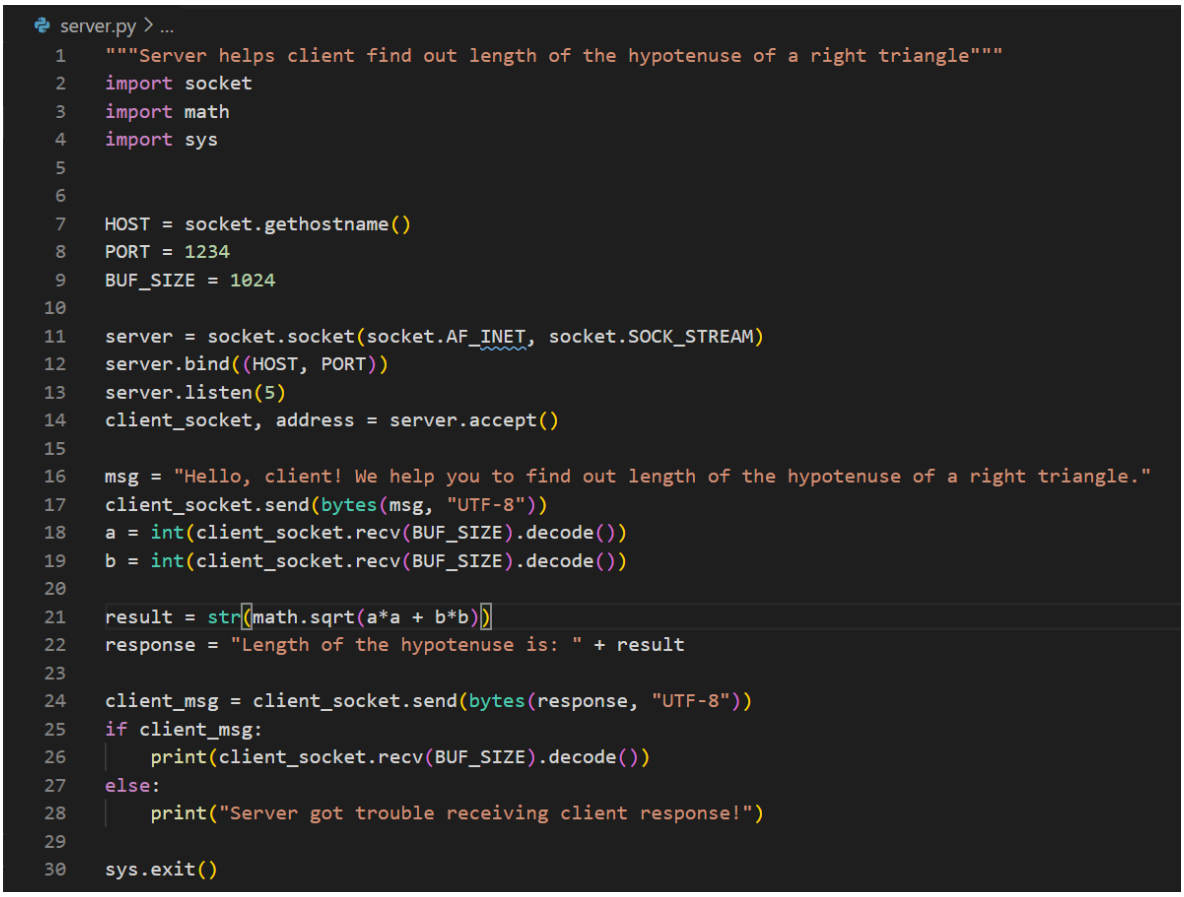


```
PS D:\Lab_1\Task_1> py server.py
Got connection from ('192.168.99.1', 53482)
Hello, server!
PS D:\Lab_1\Task_1> 
```

```
PS D:\Lab_1\Task_1> py client.py
Hello, client!
PS D:\Lab_1\Task_1> 
```

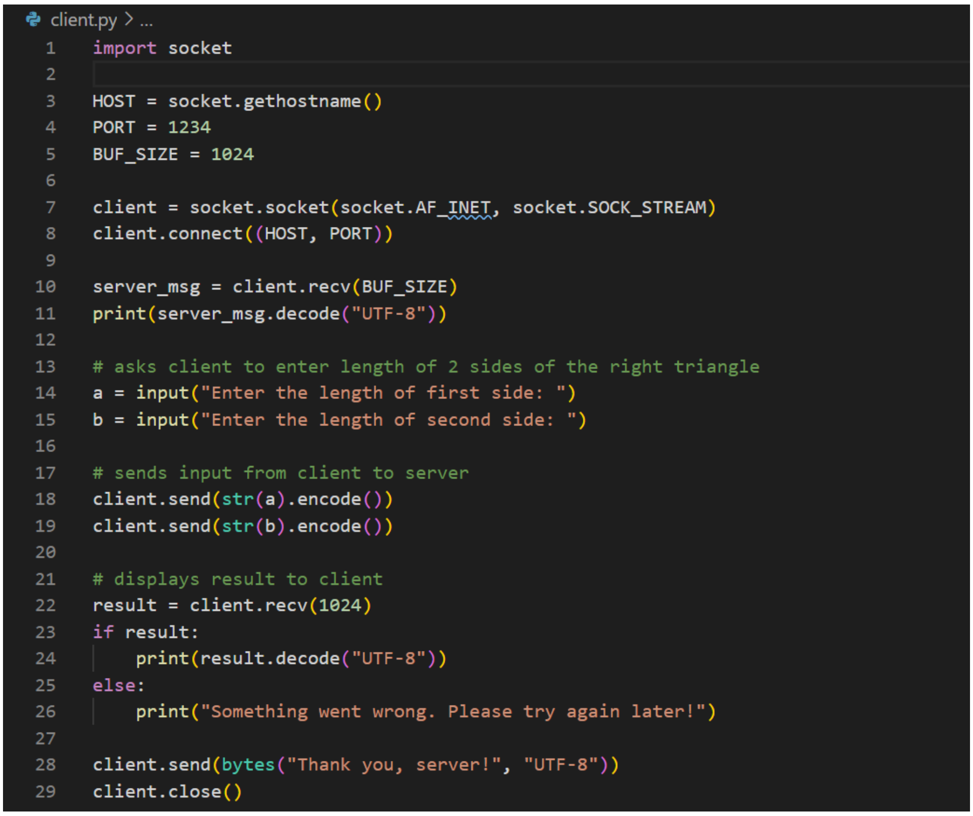
Figure 3 - Result when running 2 sides simultaneously

2.2. Реализовать клиентскую и серверную часть приложения. Клиент запрашивает у сервера выполнение математической операции «*Теорема Пифагора*», параметры, которые вводятся с клавиатуры. Сервер обрабатывает полученные данные и возвращает результат клиенту.



```python
"""Server helps client find out length of the hypotenuse of a right triangle"""
import socket
import math
import sys


HOST = socket.gethostname()
PORT = 1234
BUF_SIZE = 1024

server = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
server.bind((HOST, PORT))
server.listen(5)
client_socket, address = server.accept()

msg = "Hello, client! We help you to find out length of the hypotenuse of a right triangle."
client_socket.send(bytes(msg, "UTF-8"))
a = int(client_socket.recv(BUF_SIZE).decode())
b = int(client_socket.recv(BUF_SIZE).decode())

result = str(math.sqrt(a*a + b*b))
response = "Length of the hypotenuse is: " + result

client_msg = client_socket.send(bytes(response, "UTF-8"))
if client_msg:
    print(client_socket.recv(BUF_SIZE).decode())
else:
    print("Server got trouble receiving client response!")

sys.exit()
```

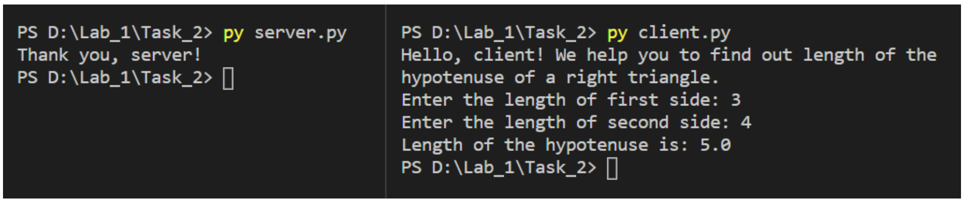
Figure 4 - Code for server-side



client.py > ...

```python
import socket

HOST = socket.gethostname()
PORT = 1234
BUF_SIZE = 1024

client = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
client.connect((HOST, PORT))

server_msg = client.recv(BUF_SIZE)
print(server_msg.decode("UTF-8"))

# asks client to enter length of 2 sides of the right triangle
a = input("Enter the length of first side: ")
b = input("Enter the length of second side: ")

# sends input from client to server
client.send(str(a).encode())
client.send(str(b).encode())

# displays result to client
result = client.recv(1024)
if result:
    print(result.decode("UTF-8"))
else:
    print("Something went wrong. Please try again later!")

client.send(bytes("Thank you, server!", "UTF-8"))
client.close()
```

Figure 5 - Code for client-side

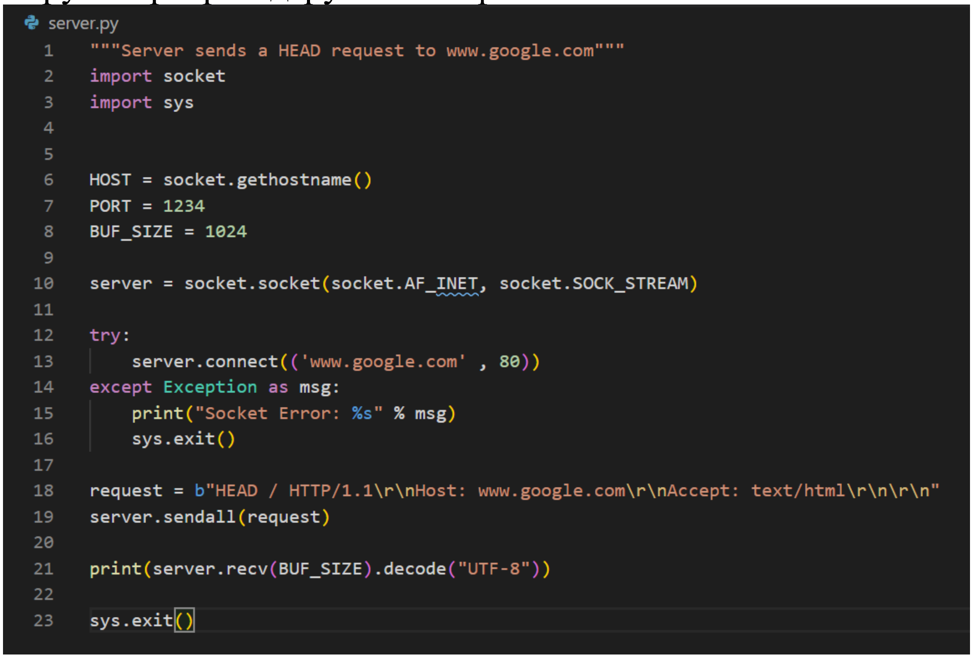


```
PS D:\Lab_1\Task_2> py server.py
Thank you, server!
PS D:\Lab_1\Task_2>
```

```
PS D:\Lab_1\Task_2> py client.py
Hello, client! We help you to find out length of the
hypotenuse of a right triangle.
Enter the length of first side: 3
Enter the length of second side: 4
Length of the hypotenuse is: 5.0
PS D:\Lab_1\Task_2>
```

Figure 6 - Result when running 2 sides simultaneously

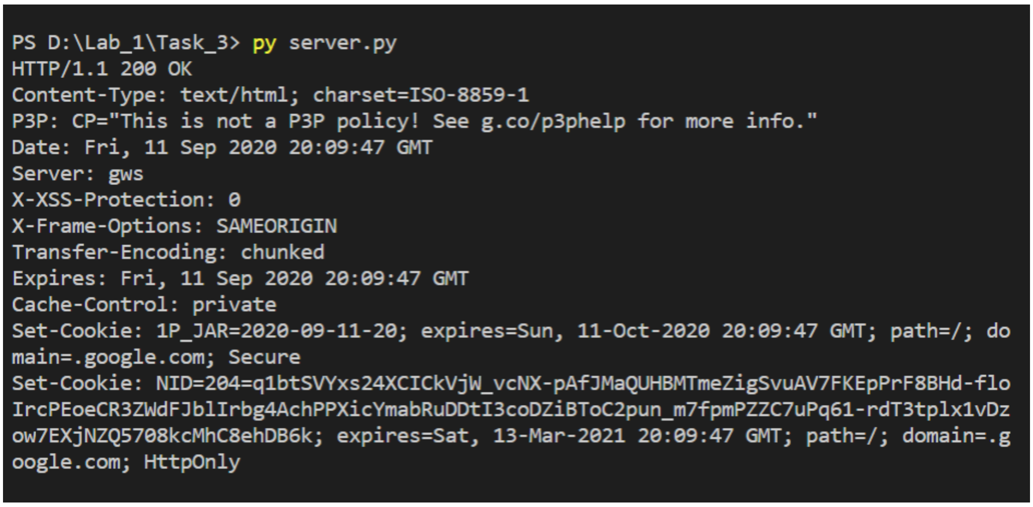
2.3. Реализовать серверную часть приложения. Клиент подключается к серверу. В ответ клиент получает http-сообщение, содержащее html-страницу, которую сервер подгружает из файла index.html.



server.py

```python
"""Server sends a HEAD request to www.google.com"""
import socket
import sys


HOST = socket.gethostname()
PORT = 1234
BUF_SIZE = 1024

server = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

try:
    server.connect(('www.google.com' , 80))
except Exception as msg:
    print("Socket Error: %s" % msg)
    sys.exit()

request = b"HEAD / HTTP/1.1\r\nHost: www.google.com\r\nAccept: text/html\r\n\r\n"
server.sendall(request)

print(server.recv(BUF_SIZE).decode("UTF-8"))

sys.exit()
```

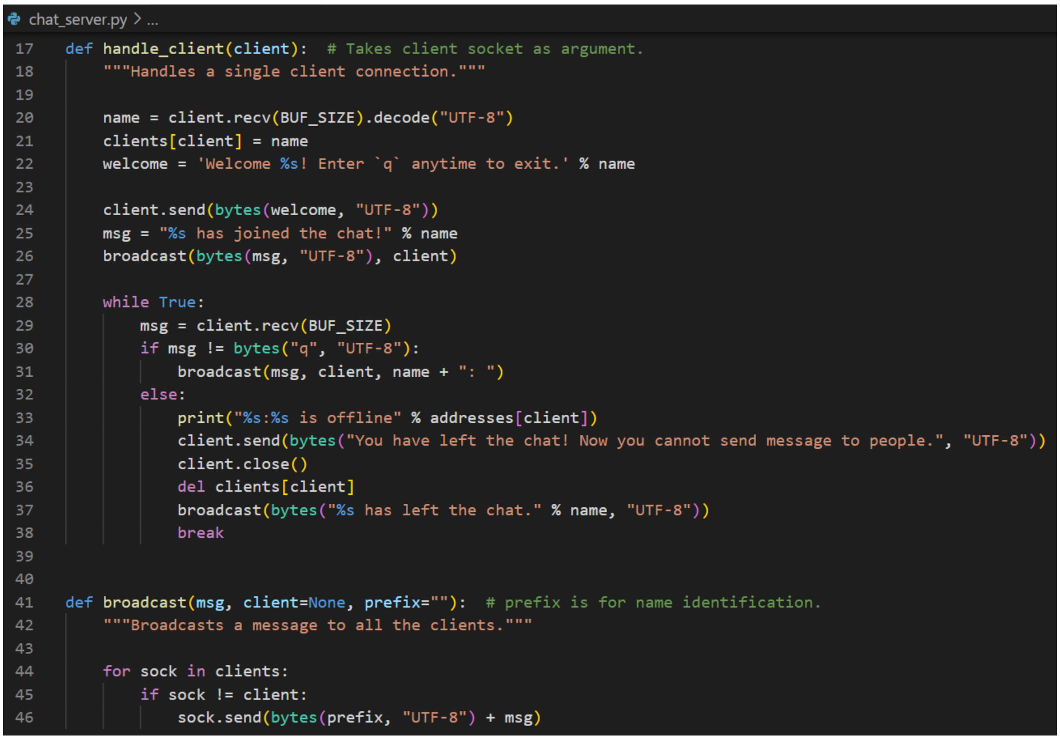
Figure 7 - Code for HEAD request



```
PS D:\Lab_1\Task_3> py server.py
HTTP/1.1 200 OK
Content-Type: text/html; charset=ISO-8859-1
P3P: CP="This is not a P3P policy! See g.co/p3phelp for more info."
Date: Fri, 11 Sep 2020 20:09:47 GMT
Server: gws
X-XSS-Protection: 0
X-Frame-Options: SAMEORIGIN
Transfer-Encoding: chunked
Expires: Fri, 11 Sep 2020 20:09:47 GMT
Cache-Control: private
Set-Cookie: 1P_JAR=2020-09-11-20; expires=Sun, 11-Oct-2020 20:09:47 GMT; path=/; do
main=.google.com; Secure
Set-Cookie: NID=204=q1btSVYxs24XCICkVjW_vcNX-pAfJMaQUHBMTmeZigSvuAV7FKEpPrF8BHd-flo
IrcPEoeCR3ZWdFJblIrbg4AchPPXicYmabRuDDtI3coDZiBToC2pun_m7fpmPZZC7uPq61-rdT3tplx1vDz
ow7EXjNZQ5708kcMhC8ehDB6k; expires=Sat, 13-Mar-2021 20:09:47 GMT; path=/; domain=.g
oogle.com; HttpOnly
```

Figure 8 - Result after sending a HEAD request to www.google.com

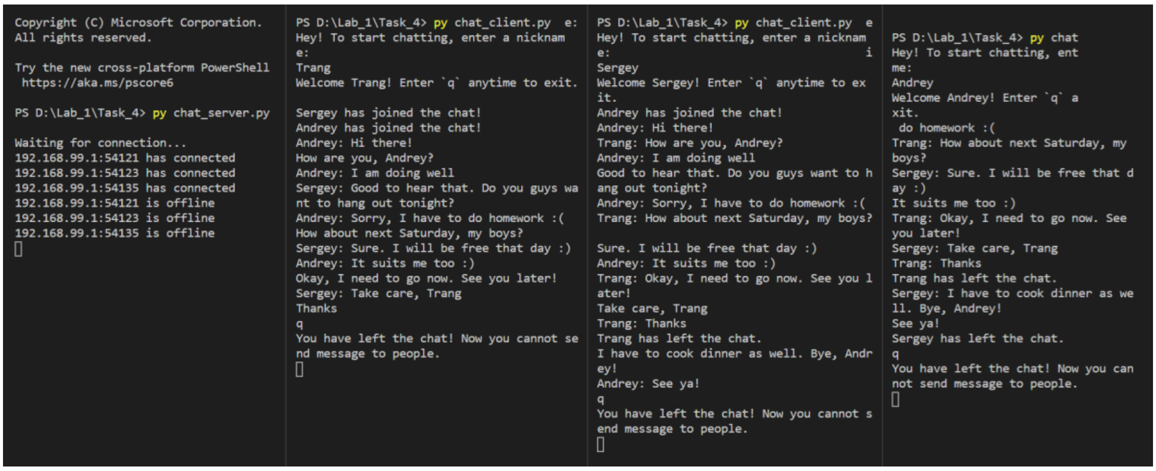
2.4. Реализовать двухпользовательский или многопользовательский чат.



chat_server.py > ...

```python
def handle_client(client):  # Takes client socket as argument.
    """Handles a single client connection."""

    name = client.recv(BUF_SIZE).decode("UTF-8")
    clients[client] = name
    welcome = 'Welcome %s! Enter `q` anytime to exit.' % name

    client.send(bytes(welcome, "UTF-8"))
    msg = "%s has joined the chat!" % name
    broadcast(bytes(msg, "UTF-8"), client)

    while True:
        msg = client.recv(BUF_SIZE)
        if msg != bytes("q", "UTF-8"):
            broadcast(msg, client, name + ": ")
        else:
            print("%s:%s is offline" % addresses[client])
            client.send(bytes("You have left the chat! Now you cannot send message to people.", "UTF-8"))
            client.close()
            del clients[client]
            broadcast(bytes("%s has left the chat." % name, "UTF-8"))
            break


def broadcast(msg, client=None, prefix=""):  # prefix is for name identification.
    """Broadcasts a message to all the clients."""

    for sock in clients:
        if sock != client:
            sock.send(bytes(prefix, "UTF-8") + msg)
```

Figure 9 - A piece of code of server-side



```
Copyright (C) Microsoft Corporation.
All rights reserved.

Try the new cross-platform PowerShell
 https://aka.ms/pscore6

PS D:\Lab_1\Task_4> py chat_server.py

Waiting for connection...
192.168.99.1:54121 has connected
192.168.99.1:54123 has connected
192.168.99.1:54135 has connected
192.168.99.1:54121 is offline
192.168.99.1:54123 is offline
192.168.99.1:54135 is offline
```

```
PS D:\Lab_1\Task_4> py chat_client.py  e:
Hey! To start chatting, enter a nicknam
e:
Trang
Welcome Trang! Enter `q` anytime to exit.

Sergey has joined the chat!
Andrey has joined the chat!
Andrey: Hi there!
How are you, Andrey?
Andrey: I am doing well
Sergey: Good to hear that. Do you guys wa
nt to hang out tonight?
Andrey: Sorry, I have to do homework :(
How about next Saturday, my boys?
Sergey: Sure. I will be free that day :)
Andrey: It suits me too :)
Okay, I need to go now. See you later!
Sergey: Take care, Trang
Thanks
q
You have left the chat! Now you cannot se
nd message to people.
```

```
PS D:\Lab_1\Task_4> py chat_client.py  e
Hey! To start chatting, enter a nicknam
e:                                      i
Sergey
Welcome Sergey! Enter `q` anytime to ex
it.
Andrey has joined the chat!
Andrey: Hi there!
Trang: How are you, Andrey?
Andrey: I am doing well
Good to hear that. Do you guys want to h
ang out tonight?
Andrey: Sorry, I have to do homework :(
Trang: How about next Saturday, my boys?

Sure. I will be free that day :)
Andrey: It suits me too :)
Trang: Okay, I need to go now. See you l
ater!
Take care, Trang
Trang: Thanks
Trang has left the chat.
I have to cook dinner as well. Bye, Andr
ey!
Andrey: See ya!
q
You have left the chat! Now you cannot s
end message to people.
```

```
PS D:\Lab_1\Task_4> py chat
Hey! To start chatting, ent
me:
Andrey
Welcome Andrey! Enter `q` a
xit.
 do homework :(
Trang: How about next Saturday, my
boys?
Sergey: Sure. I will be free that d
ay :)
It suits me too :)
Trang: Okay, I need to go now. See
you later!
Sergey: Take care, Trang
Trang: Thanks
Trang has left the chat.
Sergey: I have to cook dinner as we
ll. Bye, Andrey!
See ya!
Sergey has left the chat.
q
You have left the chat! Now you can
not send message to people.
```

Figure 10 – A short conversation of 3 people

3. Вывод

Разбираться в концепции разработки веб сервисов и компоненты клиент-серверного взаимодействия.