

Санкт-Петербургский национальный исследовательский университет
информационных технологий, механики и оптики.
Факультет инфокоммуникационных технологий.

Лабораторная работа №1

Выполнила студентка группы К33402 Куценко В.С.
Преподаватель: Говоров А.И.

Санкт-Петербург, 2020

Ход работы

Цель: овладеть практическими навыками и умениями реализации web-серверов и использования сокетов.

Практическое задание:

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

Server.py:

```
import socket

sock = socket.socket()
sock.bind(('localhost', 9000))
sock.listen(1)
conn, addr = sock.accept()

while True:
    data = conn.recv(1024)
    if not data:
        break
    print(data.decode("utf-8"))
    conn.send(b'Hello, client!')

conn.close()
```

Client.py:

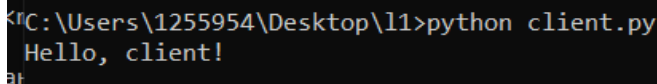
```
import socket

sock = socket.socket()
sock.connect(("localhost", 9000))
sock.send(b"Hello, server!")
```

```
data = sock.recv(1024)

print(data.decode("utf-8"))

sock.close()
```



```
C:\Users\1255954\Desktop\11>python client.py
Hello, client!
```

2. Реализовать клиентскую и серверную часть приложения. Клиент запрашивает у сервера выполнение математической операции, параметры, которые вводятся с клавиатуры. Сервер обрабатывает полученные данные и возвращает результат клиенту. Варианты:

а. Теорема Пифагора

server.py:

```
import socket
from math import sqrt

s = socket.socket(family=socket.AF_INET, type=socket.SOCK_STREAM)
s.bind((socket.gethostname(), 9090))
s.listen(1)

while True:
    client_socket, client_addr = s.accept()
    print(f"Connected client from {client_addr}")

    client_socket.send(bytes("Введите стороны треугольника:"
, "utf-8"))
    leg1, leg2 = client_socket.recv(1024).decode("utf-8").split(',')
    leg1 = int(leg1)
    leg2 = int(leg2)
```

```
res = sqrt(leg1 ** 2 + leg2 ** 2)
client_socket.send(bytes(f"Гиппотенуза: {res}", "utf-8"))
client_socket.close()
```

client.py:

```
import socket
```

```
s = socket.socket(family=socket.AF_INET, type=socket.SOCK_STREAM)
s.connect((socket.gethostname(), 9090))
```

```
msg = s.recv(1024)
print(msg.decode("utf-8"))
leg1, leg2 = input("Введите стороны треугольника: ").split()
s.send(bytes(f"{leg1},{leg2}", "utf-8"))
print(s.recv(1024).decode("utf-8"))
s.close()
```

```
Введите стороны треугольника: 3 4
Гиппотенуза: 5.0
```

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:

```
import socket

s = socket.socket(family=socket.AF_INET, type=socket.SOCK_STREAM)
s.bind((socket.gethostname(), 9090))
s.listen(1)

while True:
    client_socket, client_addr = s.accept()
    print(f"{client_addr} has connected")
    while True:
        data = client_socket.recv(1024)
        if not data:
            print(f"{client_addr} has disconnected")
            break
        with open('index.html', 'r') as file:
            html = file.read()
            client_socket.sendall(bytes(f'HTML Страница: \n\n{html}', 'utf-8'))
        client_socket.close()
```

Client.py:

```
import socket

s = socket.socket(family=socket.AF_INET, type=socket.SOCK_STREAM)
s.connect((socket.gethostname(), 9090))

s.sendall(bytes(f'.', 'utf-8'))
data = s.recv(1024)
```

```
print(data.decode('utf-8'))
s.close()
```

```
C:\Users\1255954\Desktop\l13>python client.py
HTML Страница:

<!DOCTYPE html>
<html lang="en">

<body>
  <h1>Hello</h1>
</body>

</html>
```

4. Реализовать двухпользовательский или многопользовательский чат. Реализация многопользовательского чата позволяет получить максимальное количество баллов.

Server.py

```
import socket
```

```
s = socket.socket(family=socket.AF_INET, type=socket.SOCK_DGRAM)
```

```
s.bind(('127.0.0.1', 9090))
```

```
connections = []
```

```
while 1:
```

```
    data, address = s.recvfrom(1024)
```

```
    print(address[0], address[1])
```

```
    if address not in connections:
```

```
        connections.append(address)
```

```
    for client in connections:
```

```
        if client == address:
```

```
            continue
```

```
s.sendto(data, client)
```

Client(N).py

```
import socket
```

```
import threading
```

```
server_address = '127.0.0.1', 9090
```

```
s = socket.socket(family=socket.AF_INET, type=socket.SOCK_DGRAM)
```

```
s.bind(('',0))
```

```
def readSocket():
```

```
    while 1:
```

```
        data = s.recv(1024)
```

```
        print(data.decode('utf-8'))
```

```
def runClient():
```

```
    print("Введите имя: ")
```

```
    nickname = input()
```

```
    s.sendto((nickname + ' Connect to server').encode('utf-8'),
```

```
            server_address)
```

```
    while 1:
```

```
        message = input()
```

```
        s.sendto((nickname + ': ' + message).encode('utf-8'), server_address)
```

```
thread1, thread2 = threading.Thread(target=readSocket), threading.Thread(target=runClient)
```

```
thread1.start(), thread2.start()
```

Вид окна клиентом 3:

```
Введите имя:  
Client1  
Client2 Connect to server  
Client3 Connect to server  
Client3: hi  
Client2: hiiiii
```