

Министерство науки и высшего образования Российской Федерации
федеральное государственное автономное образовательное учреждение
высшего образования
«Национальный исследовательский университет ИТМО»

ЛАБОРАТОРНАЯ РАБОТА №1 «Работа с сокетами»

Выполнил:

Студент III курса ИМРиП

Группы: D33101

Ф.И.О.: Чжан Цзяи

Проверил:

Говоров Антон Игоревич

Санкт-Петербург
2021

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.

КОД:

Сервер:

```
import socket

s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.bind(("127.0.0.1", 14900))
s.listen(3)
while True:
    c, addr = s.accept()

    data = c.recv(16384)
    while data:
        c.send(b"Hello client")
        udata = data.decode('utf-8')
        print(udata)
        break
    c.close()
```

Клиент:

```
server.py x client.py x
1 import socket
2
3 s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
4 s.connect(("127.0.0.1", 14900))
5 s.send(b"Hello, server")
6 data = s.recv(16384)
7 udata = data.decode('utf-8')
8 print(udata)
9 s.close()
```

Результат:

```
C:\Users\czaic\PycharmProjects\socket\venv\Scripts\python.exe C:/Users/czaic/PycharmProjects/socket/server.py
Hello, server
|
```

```
C:\Users\czaic\PycharmProjects\socket\venv\Scripts\python.exe C:/Users/czaic/PycharmProjects/socket/client.py
Hello client

Process finished with exit code 0
```

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

КОД:

Сервер:

```
import socket
import struct
from math import *

def max_l(a, b):
    return sqrt(pow(a, 2) + pow(b, 2))

def min_l(m, n):
    if m == n:
        return 'нельзя равно'
    else:
        return sqrt(abs(pow(m, 2) - pow(n, 2)))

s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.bind(("127.0.0.1", 14900))
s.listen(5)
while True:
    c, addr = s.accept()
    data = c.recv(16384)
    udata = list(data)
    if 1 in udata:
        result = max_l(udata[1], udata[2])
        a = struct.pack('!f', result) # Переводить тип float в тип байта
        c.send(a)
    elif 2 in udata:
        result1 = min_l(udata[1], udata[2])
        b = struct.pack('!f', result1) # Переводить тип float в тип байта
        c.send(b)
    else:
        c.send(b"write 'Y' or 'N'")
    c.close()
```

Клиент:

```
import socket
import struct

q = input("Хотите ли вы вычислить гипотенузу? Ответ по 'Y' или 'N'")
j = int(input("Первая длина стороны"))
k = int(input("Вторая длина стороны"))

if q == 'Y':
    lst = [1, j, k]
elif q == 'N':
    lst = [2, j, k]
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.connect(("127.0.0.1", 14900))
s.send(bytearray(lst))
data = s.recv(16384)
udata = struct.unpack('!f', data) # Переводить тип байта в тип float
print(udata)
s.close()
```

Результат:

```
C:\Users\czaic\PycharmProjects\socket\venv\Scripts\python.exe C:/Users/czaic/PycharmProjects/s
Хотите ли вы вычислить гипотенузу? Ответ по 'Y' или 'N'Y
Первая длина стороны6
Вторая длина стороны7
(9.219544410705566,)

Process finished with exit code 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.

КОД:

Сервер:

```
import random
import socket
import time

server = socket.socket()
host = '127.0.0.1'
port = 555
server.bind((host, port))
print('Starting server on', host, port)
print('The Web server URL for the would be http://%s:%d' % (host, port))
server.listen(5)
print('Entering infinite loop; hit CTRL-C to exit')
while True:
    client, (client_host, client_port) = server.accept()
    print('Got connection from', client_host, client_port)
    client.recv(1000)
    response_type = 'HTTP/1.0 200 ok\n'
    headers = 'Content-Type: text/html\n\n'
    body = """
        <html>
        <body>
        <h1>Hello World</h1>!
        </body>
        </html>
    """
    response = response_type + headers + body
    client.send(response.encode('utf-8'))
    client.close()
```

Клиент:

```
import socket

conn = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
conn.connect(('127.0.0.1', 555))
headers = [
    'GET / HTTP/1.1',
    'Host: 127.0.0.1',
    'Connection: keep-alive',
    'Accept: text/html',
    '\n'
]
content = '\n'.join(headers)
print(content)
conn.send(content.encode())
result = conn.recv(16384)
print(result.decode())
```

Результат:

```
GET / HTTP/1.1
Host: 127.0.0.1
Connection: keep-alive
Accept: text/html

HTTP/1.0 200 ok
Content-Type: text/html

    <html>
    <body>
    <h1>Hello World</h1>!
    </body>
    </html>

Process finished with exit code 0
```

4.Реализовать двухпользовательский или многопользовательский чат

Реализациямногопользовательского часа позволяет получить максимальное количество баллов.

Код:

Сервер:

```
import socket
import threading

s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.bind(("127.0.0.1", 14900))
s.listen()
clients = []
names = []

def talkbroad(message):
    for client in clients:
        client.send(message)

def handle(client):
    while True:
        try:
            message = client.recv(16384)
            talkbroad(message)
        except:
            index = clients.index(client)
            clients.remove(client)
            client.close()
            name = names[index]
            talkbroad('{} left'.format(name).encode('utf-8'))
            names.remove(name)
            break
```

```
def receive():
    while True:
        client, addr = s.accept()
        print("connected with {}".format(str(addr)))
        client.send('name'.encode('utf-8'))
        name = client.recv(16384).decode('utf-8')
        names.append(name)
        clients.append(client)
        print("name is {}".format(name))
        talkbroad("{} joined".format(name).encode('utf-8'))
        client.send('connected to server'.encode('utf-8'))
        thread = threading.Thread(target=handle, args=(client,))
        thread.start()

receive()
```

Клиент:


```

import socket
import threading

name = input("Запомните вашу имя")
c = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
c.connect(("127.0.0.1", 14900))

def receive():
    while True:
        try:
            message = c.recv(16384).decode('utf-8')
            if message == 'name':
                c.send(name.encode('utf-8'))
            else:
                print(message)
        except:
            print("Error")
            c.close()
            break

def write():
    while True:
        message = '{}: {}'.format(name, input(''))
        c.send(message.encode('utf-8'))

receive_thread = threading.Thread(target=receive)
receive_thread.start()
write_thread = threading.Thread(target=write)
write_thread.start()

```

Результат:

```

C:\Users\czaic\PycharmProjects\socket\venv\Scripts\python.exe C:/Users/czaic/PycharmProjects/socket/server4.py
connected with ('127.0.0.1', 65465)
name is Zhang
connected with ('127.0.0.1', 65467)
name is Wang
connected with ('127.0.0.1', 65469)
name is Hu

```

```
Запомните вашу имяZhang
Zhang joinedconnected to server
Wang joined
Hu joined
Привет
Zhang: Привет
Wang: Привет
Hu: Привет
```

```
Запомните вашу имяWang
Wang joinedconnected to server
Hu joined
Zhang: Привет
Привет
Wang: Привет
Hu: Привет
|
```

```
Запомните вашу имяHu
Hu joined
connected to server
Zhang: Привет
Wang: Привет
Привет
Hu: Привет
```