

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 1

ПО ДИСЦИПЛИНЕ:

«Web-программирование»

НА ТЕМУ:

«Работа с сокетами»

Работу выполнил:

Студент гр. К33422

Кириллов Николай Александрович

Преподаватель:

Говоров Антон Игоревич

Санкт-Петербург

2020

Цель работы

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

Практическое задание

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.

Выполнение работы

Код

Server

```
import socket

sock = socket.socket()
sock.bind(('', 9090))
sock.listen(1)
conn, addr = sock.accept()

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

Client

```
import socket

sock = socket.socket()
sock.connect(('localhost', 9090))
sock.send('Hello, server!'.encode('utf-8'))

data = sock.recv(1024)
sock.close()
print(data.decode('utf-8'))
```

Результаты работы программы

```
"C:\Program Files (x86)\Python38-32\python.exe" C:/Users/Nikolay/Desktop/Web_programming/Lab_1/1.1/1.1_client.py
Hello, client

Process finished with exit code 0

"C:\Program Files (x86)\Python38-32\python.exe" C:/Users/Nikolay/Desktop/Web_programming/Lab_1/1.1/1.1_server.py
Hello, server!

Process finished with exit code 0
```

Задание 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.

Вариант «Теорема Пифагора»

Выполнение работы

Код

Server

```
import socket
from math import sqrt

sock = socket.socket()
sock.bind(('', 9090))
sock.listen(1)
conn, addr = sock.accept()

data = conn.recv(1024)
leg1 = data.decode('utf-8').split()[0]
leg2 = data.decode('utf-8').split()[1]
hyp = sqrt(float(leg1)**2 + float(leg2)**2)
conn.sendall(str(hyp).encode('utf-8'))

conn.close()
```

Client

```
import socket

sock = socket.socket()
sock.connect(('localhost', 9090))

legs = input('Enter 2 legs: ')
sock.sendall(legs.encode('utf-8'))

data = sock.recv(1024)
sock.close()

print('Hypotenuse: ', data.decode('utf-8'))
```

Результаты выполнения программы

```
"C:\Program Files (x86)\Python38-32\python.exe" C:/Users/Nikolay/Desktop/Web_programming/Lab_1/1.2/1.2_client.py
Enter 2 legs: 3 4
Hypotenuse: 5.0

Process finished with exit code 0
```

Задание 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.

Выполнение работы

Код

Server

```
import socket
import codecs

server = socket.socket()
host = '127.0.0.1'
port = 555
server.bind((host, port))

link = str('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()
    client.sendall(link.encode('utf-8'))
    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 = codecs.open("index.html")
    response = response_type + headers + body.read()
    client.sendall(response.encode('utf-8'))
    client.close()
```

Client

```
import socket

client = socket.socket()
client.connect(('127.0.0.1', 555))

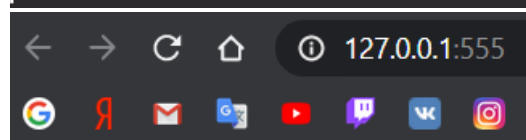
data = client.recv(1024)
client.close()

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

Результат выполнения программы

```
"C:\Program Files (x86)\Python38-32\python.exe" C:/Users/Nikolay/Desktop/Web_programming/Lab_1/1.3/1.3_client.py
http://127.0.0.1:555

Process finished with exit code 0
```



Hello, World!

Задание 4

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

Выполнение работы (реализация многопользовательского чата)

Код

Server

```
sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
sock.bind(('', 9090))
client = []
print('Server start')
while True:
    data, address = sock.recvfrom(1024)
    print('Got connection from: ', address[0], address[1])
    if address not in client:
        client.append(address)
    for clients in client:
        if clients == address:
            continue
        sock.sendto(data, clients)
```

Sample of client

```
import socket
import threading

def read_sok():
    while True:
        data = sor.recv(1024)
        print(data.decode('utf-8'))

server = 'localhost', 9090
name = input('Enter your nickname: ')
sor = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
sor.bind(('', 0))
sor.sendto((name + ' Connect to server').encode('utf-8'), server)
stream = threading.Thread(target=read_sok)
stream.start()
while True:
    message = input()
    sor.sendto(('[' + name + ']' + message).encode('utf-8'), server)
```

Результат выполнения программы

```
"C:\Program Files (x86)\Python38-32\python.exe" C:/Users/Nikolay/Desktop/Web_programming/Lab_1/1.4/1.4_client_1.py
Enter your nickname: Client 1
Client 2 Connect to server
Client 3 Connect to server
Hello, I'm client 1
[Client 2] Hello, I'm client 2
[Client 3] Hello, I'm client 3
```

```
"C:\Program Files (x86)\Python38-32\python.exe" C:/Users/Nikolay/Desktop/Web_programming/Lab_1/1.4/1.4_client_2.py
Enter your nickname: Client 2
Client 3 Connect to server
[Client 1] Hello, I'm client 1
Hello, I'm client 2
[Client 3] Hello, I'm client 3
```

```
"C:\Program Files (x86)\Python38-32\python.exe" C:/Users/Nikolay/Desktop/Web_programming/Lab_1/1.4/1..4_client_3.py
Enter your nickname: Client 3
[Client 1] Hello, I'm client 1
[Client 2] Hello, I'm client 2
Hello, I'm client 3
```

Вывод

В результате выполнения лабораторной работы получены практические навыки работы с web-серверами, использования сокетов и потоков.