

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

Лабораторная работа №1
«Работа с сокетами»
по дисциплине:
«Web-программирование»

Выполнил:
Студент III курса ИМР и П
Группы: D33101
Ф.И.О.: Ван Цюаньюй

Проверил:
Говоров Антон Игоревич

Санкт-Петербург
2021

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

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.

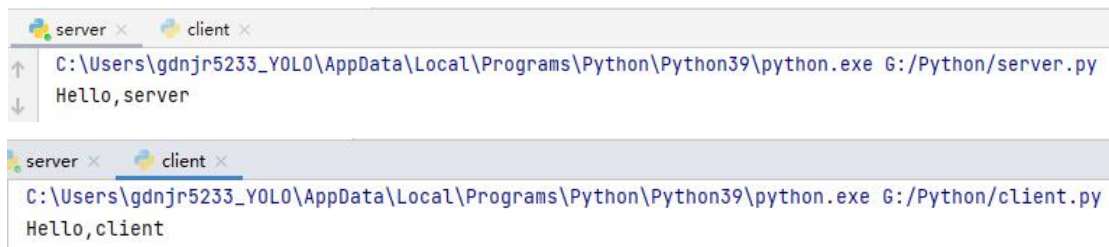
Код(сервер):

```
server.py x client.py x
1  import socket
2
3  s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
4  s.bind(("127.0.0.1", 10034))
5  s.listen(5)
6  while True:
7      clientsocket, address = s.accept()
8      msg = clientsocket.recv(1024)
9      print(msg.decode("utf-8"))
10     clientsocket.send(bytes("Hello,client", "utf-8"))
11     clientsocket.close()
12
```

Код(клиент):

```
server.py x client.py x
1  import socket
2
3  c = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
4  c.connect(("127.0.0.1", 10034))
5  c.send(b"Hello,server")
6  msg = c.recv(1024)
7  print(msg.decode("utf-8"))
8  c.close()
9
```

Результат:

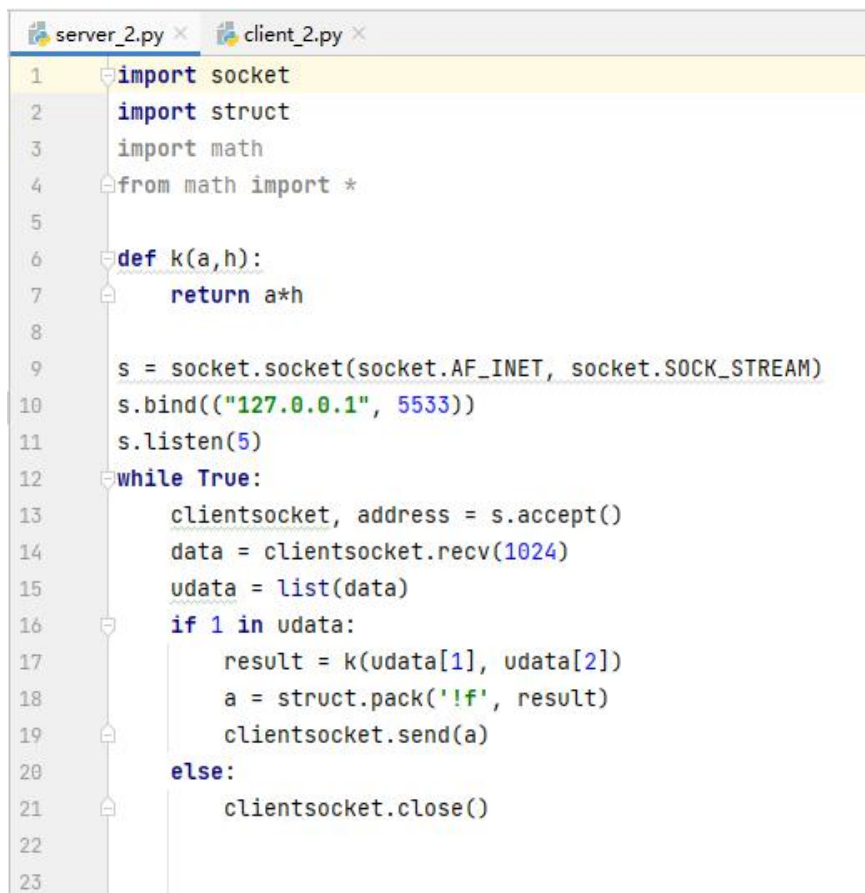


The screenshot shows two terminal windows. The top window, titled 'server', shows the command prompt running 'C:\Users\gdnjr5233_YOLO\AppData\Local\Programs\Python\Python39\python.exe G:/Python/server.py' and the output 'Hello,server'. The bottom window, titled 'client', shows the command prompt running 'C:\Users\gdnjr5233_YOLO\AppData\Local\Programs\Python\Python39\python.exe G:/Python/client.py' and the output '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: d. Поиск площади параллелограмма.

Код(сервер):



The screenshot shows a code editor with two tabs: 'server_2.py' and 'client_2.py'. The 'server_2.py' tab is active, showing the following Python code:

```
1 import socket
2 import struct
3 import math
4 from math import *
5
6 def k(a,h):
7     return a*h
8
9 s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
10 s.bind(("127.0.0.1", 5533))
11 s.listen(5)
12 while True:
13     clientsocket, address = s.accept()
14     data = clientsocket.recv(1024)
15     udata = list(data)
16     if 1 in udata:
17         result = k(udata[1], udata[2])
18         a = struct.pack('!f', result)
19         clientsocket.send(a)
20     else:
21         clientsocket.close()
22
23
```

Код(клиент):

```
server_2.py x client_2.py x
1 import socket
2 import struct
3
4 a = int(input("введите сторону a:"))
5 h = int(input("введите сторону h:"))
6
7 lst = [1, a, h]
8
9 s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
10 s.connect(("127.0.0.1", 5533))
11 s.send(bytearray(lst))
12 data = s.recv(1024)
13 updata = struct.unpack('!f', data)
14 print(updata)
15 s.close()
16
```

Результат:

```
C:\Users\gdnjr5233_Y0L0\AppData\Local\Programs\Python\Python39\python.exe G:/Python/client_2.py
введите сторону a:4
введите сторону h:6
(24.0,)
```

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):

```
server_3.py x client_3.py x
1 import socket
2 import time
3 import random
4
5 server = socket.socket()
6 host = '127.0.0.1'
7 port = 998
8 server.bind((host, port))
9 print("starting server on", host, port)
10 print("the web server URL for the would be http://%s:%d" % (host, port))
11 server.listen(5)
12 print("entering infinite loop: hit CTRL-C to exit")
13 while True:
14     client, (client_host, client_port) = server.accept()
15     print("got connection from", client_host, client_port)
16     client.recv(1024)
17     response_type = 'HTTP/1.0 200 ok\n'
18     headers = 'content-type: text/html\n\n'
19     body = """
20     <html>
21     <body>
22     <h1>Hello World</h1>!
23     </body>
24     </html>
25     """
26     response = response_type + headers + body
27     client.send(response.encode('utf-8'))
28     client.close()
```

Код(клиент):

```
server_3.py x client_3.py x
1 import socket
2
3 c = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
4 c.connect(("127.0.0.1", 998))
5 headers = [
6     'get / HTTP/1.1',
7     'host: 127.0.0.1',
8     'connection: keep-alive',
9     'accept: text/html',
10    '\n'
11 ]
12 content = '\n'.join(headers)
13 print(content)
14 c.send(content.encode())
15 result = c.recv(1024)
16 print(result.decode())
```

Результат:



The image shows two terminal windows side-by-side. The top window, titled 'server_3', shows the execution of a Python script 'server_3.py'. The output indicates the server is starting on 127.0.0.1:998, provides the URL 'http://127.0.0.1:998', and enters an infinite loop. The bottom window, titled 'client_3', shows the execution of a Python script 'client_3.py'. It displays an HTTP GET request to the server, the server's response (200 OK), and the received HTML content: '<html><body><h1>Hello World</h1></body></html>'. The paths in the terminal are 'C:\Users\gdnjr5233_Y0L0\AppData\Local\Programs\Python\Python39\python.exe' and 'G:/Python/'.

```
server_3 x client_3 x
C:\Users\gdnjr5233_Y0L0\AppData\Local\Programs\Python\Python39\python.exe G:/Python/server_3.py
starting server on 127.0.0.1 998
the web server URL for the would be http://127.0.0.1:998
entering infinite loop: hit CTRL-C to exit
got connection from 127.0.0.1 55906

server_3 x client_3 x
C:\Users\gdnjr5233_Y0L0\AppData\Local\Programs\Python\Python39\python.exe G:/Python/client_3.py
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>
```

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

Код(сервер):

```
server_4.py x client_4.py x client_4_2.py x client_4_3.py x
1 import socket
2 import threading
3 s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
4 s.bind(("127.0.0.1", 10101))
5 s.listen()
6 clients = []
7 names = []
8
9 def talkbroad(message):
10     for client in clients:
11         client.send(message)
12
13 def handle(client):
14     while True:
15         try:
16             message = client.recv(23233)
17             talkbroad(message)
18         except:
19             index = clients.index(client)
20             clients.remove(client)
21             client.close()
22             name = names[index]
23             talkbroad('{} left'.format(name).encode('utf-8'))
24             names.remove(name)
25             break
26
27 def receive():
28     while True:
29         client, addr = s.accept()
30         print("connected with {}".format(str(addr)))
31         client.send("name".encode("utf-8"))
32         name = client.recv(23233).decode("utf-8")
33         names.append(name)
34         clients.append(client)
35         print("name is {}".format(name))
36         talkbroad("{} joined".format(name).encode("utf-8"))
37         client.send("connected to server".encode("utf-8"))
38         thread = threading.Thread(target=handle, args=(client,))
39         thread.start()
40
41 receive()
```


Код(клиент):

```
server_4.py x client_4.py x client_4_2.py x client_4_3.py x
1 import socket
2 import threading
3
4 name = input("введите ваш имя")
5 c = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
6 c.connect(("127.0.0.1", 10101))
7
8 def receive():
9     while True:
10         try:
11             message = c.recv(23233).decode("utf-8")
12             if message == 'name':
13                 c.send(name.encode("utf-8"))
14             else:
15                 print(message)
16         except:
17             print("error")
18             c.close()
19             break
20
21 def write():
22     while True:
23         message = "{}:{}".format(name, input(""))
24         c.send(message.encode("utf-8"))
25
26 receive_thread = threading.Thread(target=receive)
27 receive_thread.start()
28 write_thread = threading.Thread(target=write)
29 write_thread.start()
```


Результат:

```
server_4 x client_4 x client_4_2 x client_4_3 x
C:\Users\gdnjr5233_YOLO\AppData\Local\Programs\Python\Python39\python.exe G:/Python/server_4.py
connected with ('127.0.0.1', 64756)
name is neymar jr
connected with ('127.0.0.1', 64766)
name is gdragon
connected with ('127.0.0.1', 64769)
name is YOLO

server_4 x client_4 x client_4_2 x client_4_3 x
C:\Users\gdnjr5233_YOLO\AppData\Local\Programs\Python\Python39\python.exe G:/Python/client_4.py
введите ваш имя neymar jr
neymar jr joined
connected to server
gdragon joined
YOLO joined
whassup
neymar jr:whassup
gdragon:hello
YOLO:hey dude

server_4 x client_4 x client_4_2 x client_4_3 x
C:\Users\gdnjr5233_YOLO\AppData\Local\Programs\Python\Python39\python.exe G:/Python/client_4_2.py
введите ваш имя g`dragon
gdragon joined
connected to server
YOLO joined
neymar jr:whassup
hello
gdragon:hello
YOLO:hey dude

server_4 x client_4 x client_4_2 x client_4_3 x
C:\Users\gdnjr5233_YOLO\AppData\Local\Programs\Python\Python39\python.exe G:/Python/client_4_3.py
введите ваш имя YOLO
YOLO joined
connected to server
neymar jr:whassup
gdragon:hello
hey dude
YOLO:hey dude
```