

Client Server Socket Programmings

CN CSL317

Instructor Name: Dr. Anshul Agarwal Summer 2021

Abhishek Kumar Yadav BT18CSE106

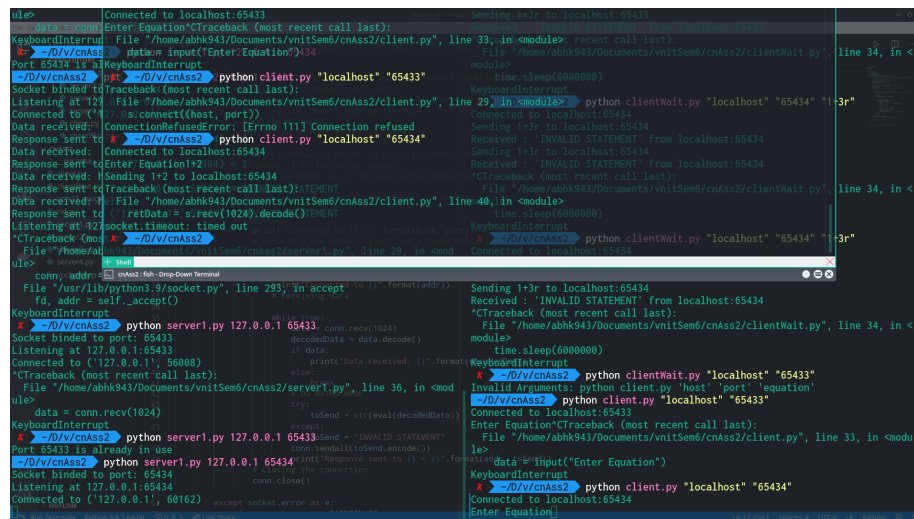
Requirements

Install python in your environment if not already installed: `pip install python`

How To Run

- Each of the server files can be run using: `python serverx.py "host" "port"`
 - Client files can be run using: `python client.py "host" "port"`
-

server1.py



Left console: server1.py, Right console: first client Top console: second client, trying to connect.

I have used `socket.listen(0)`(doesn't keep pending requests when already having a connection) to implement connection refusal when having a connection already. If that doesn't work, there is `timeout` provided in `client`, which wait for 2 seconds and if it doesn't receive any response it terminates. Connection is closed when `client` closes the connection.

server2.py



Top: server2.py Left: concurrently connected, invalid input Right: concurrently connected, valid input

Each connection spawns it's own thread and engages the client. Connection is closed when client closes the connection.

server3.py

[illegible]

Top: server2.py Left: concurrently connected, valid input Right: concurrently connected, invalid input

- Every socket is added to the buffer including server socket.
- Socket is selected using select function.
 - If socket is server
 - * We accept the connection in it and append it to buffer.
 - If socket is client data is taken from the socket
 - * If data is present, output is sent,
 - * If data is not present then the socket is removed from buffer and connection is closed.

server4.py

The screenshot displays two terminal windows on a Kali Linux desktop. The left window, titled 'x - /D:/v/cnAS2', shows the execution of a Python server script. It receives a request from '127.0.0.1:42516' and responds with 'hello'. The right window, titled 'x - /D:/v/cnAS2', shows the execution of a Python client script. It connects to 'localhost:12347' and sends 'hello', receiving a 'hello' response. Both windows show the file paths and line numbers of the scripts.

```

x - /D:/v/cnAS2 python server4.py "localhost" "12347"
data = input(Socket address to port: 12347)
KeyboardInterrupt
Listening at localhost:12347
x - /D:/v/cnAS2 Receive request from: (127.0.0.1, 42516)
Connected to localhost:12347
Enter EquationData received: hello
Sending hello to Response: hello
Received: 'hello'
Data received: hello
Enter EquationResponse sent= 'hello' call last):
File ~/home/ab4k34/Documents/vn15em6/cnAS2/client.py, line 29, in <module>
data = input(Socket address to port: 12347)
KeyboardInterrupt
x - /D:/v/cnAS2 python client.py "localhost" "12347"
Connected to localhost:12347
Enter EquationHello
Sending hello to localhost:12347
Received: 'hello' from localhost:12347
Enter Equation

```

Port already occupied

```
ConnectionRefusedError: [Errno 111] Connection refused
~/D/v/cnAs52 ➤ python server1.py 127.0.0.1 65433
Port 65433 is already in use
~/D/v/cnAs52 ➤
```

```
★ ~/D/v/cnAs52 ➤ python server1.py "localhost" 65433
Socket binded to port: 65433
Listening at localhost:65433
|
```

```
~/D/v/cnAss2 python client.py "localhost" "12347"
Traceback (most recent call last):
  File "/home/abhk943/Documents/vnitSem6/cnAss2/client.py", line 16, in <module>
    s.connect((host, port))
ConnectionRefusedError: [Errno 111] Connection refused
~/D/v/cnAss2
```

Server not started

Expression evaluation

For evaluating the mathematical expression the string is simply passed to `eval()` function available with python, so the server is capable to solving any type of equation.

Acknowledgement

- <https://www.positronx.io/create-socket-server-with-multiple-clients-in-python/>
- <https://steelkiwi.com/blog/working-tcp-sockets/>
- <https://yasooob.me/2013/08/06/python-socket-network-programming/>
- <https://docs.python.org/3/library/socket.html>
- <https://www.techbeamers.com/python-tutorial-write-multithreaded-python-server/>
- <https://docs.python.org/3/tutorial/errors.html>
- <https://docs.python.org/3/library/multiprocessing.html>
- <https://realpython.com/python-sockets/#echo-client-and-server>