CS359 Assignment 3 - Client Server Calculator

Mihir Parag Mantri
2001CS46

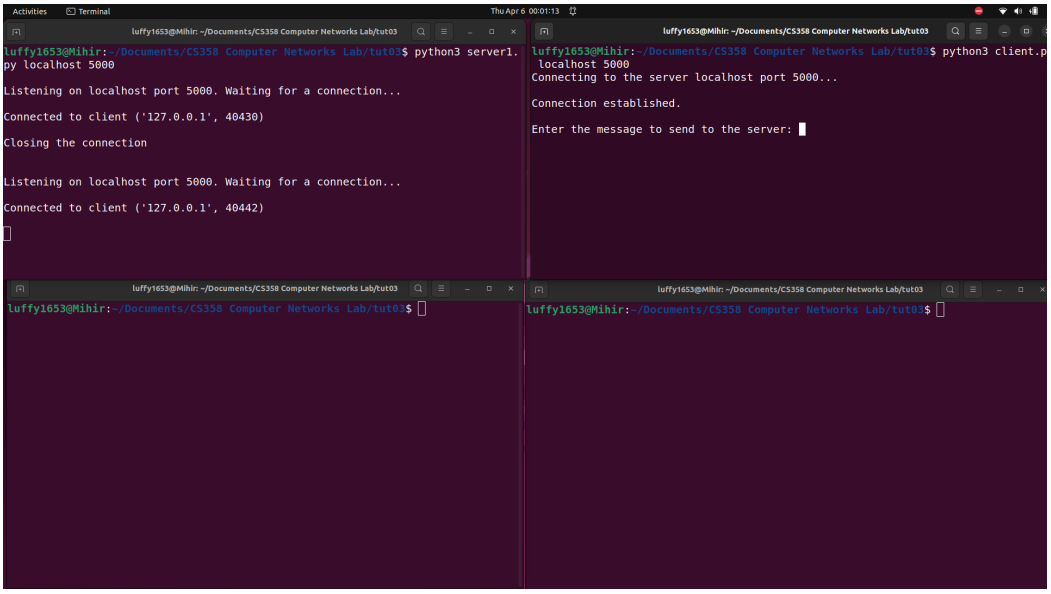
Running the client: python[version] client.py [IP Address] [Port Number]
E.g. *python3 client.py localhost 5000* (used throughout the report)

Running the servers: python[version] server[number].py [IP Address] [Port Number]
E.g. *python3 server1.py localhost 5000*

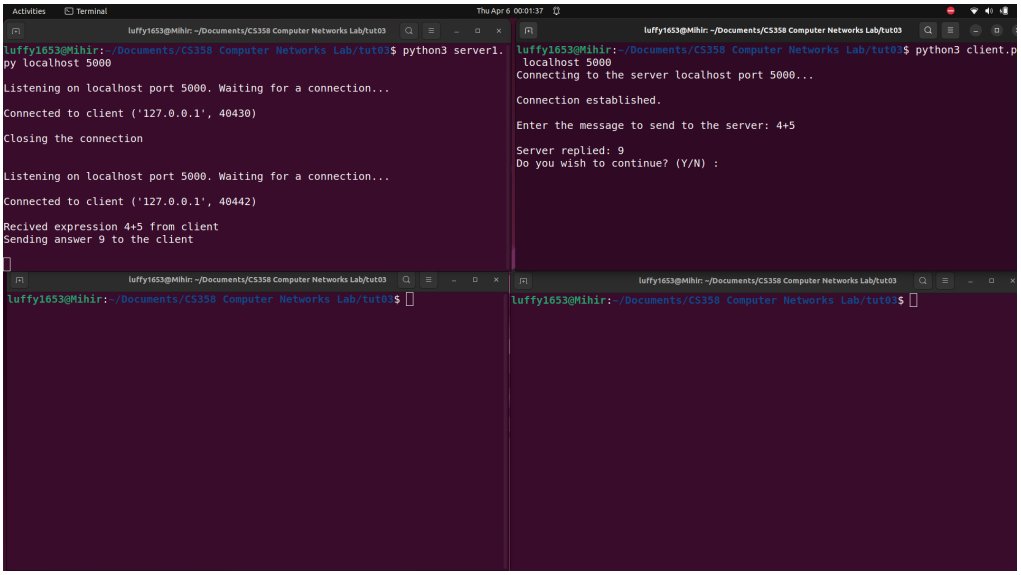
Server 1: Single process server - handles one client at a time

Connecting one client to server 1. Successful connection is seen, as shown below.

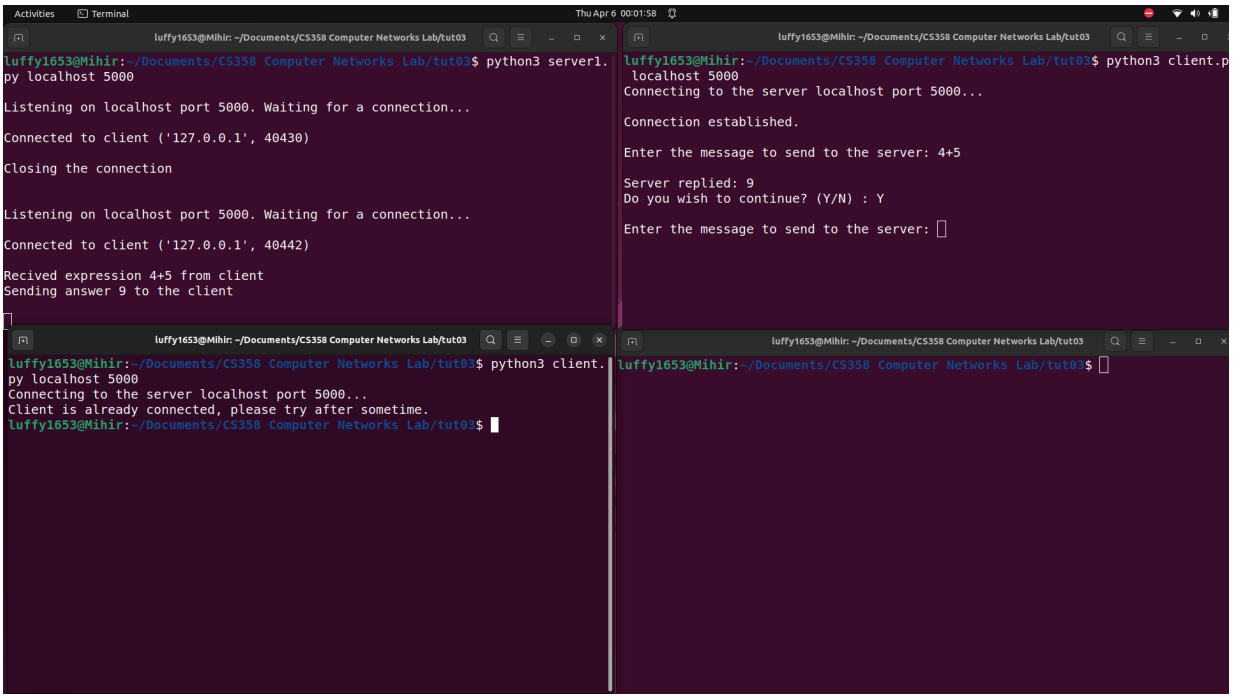
Note: There are two client connections seen on the server side. As explained in the code and in the video, it is done to solve the problem of two clients connecting at a time. We create a 'fake' client



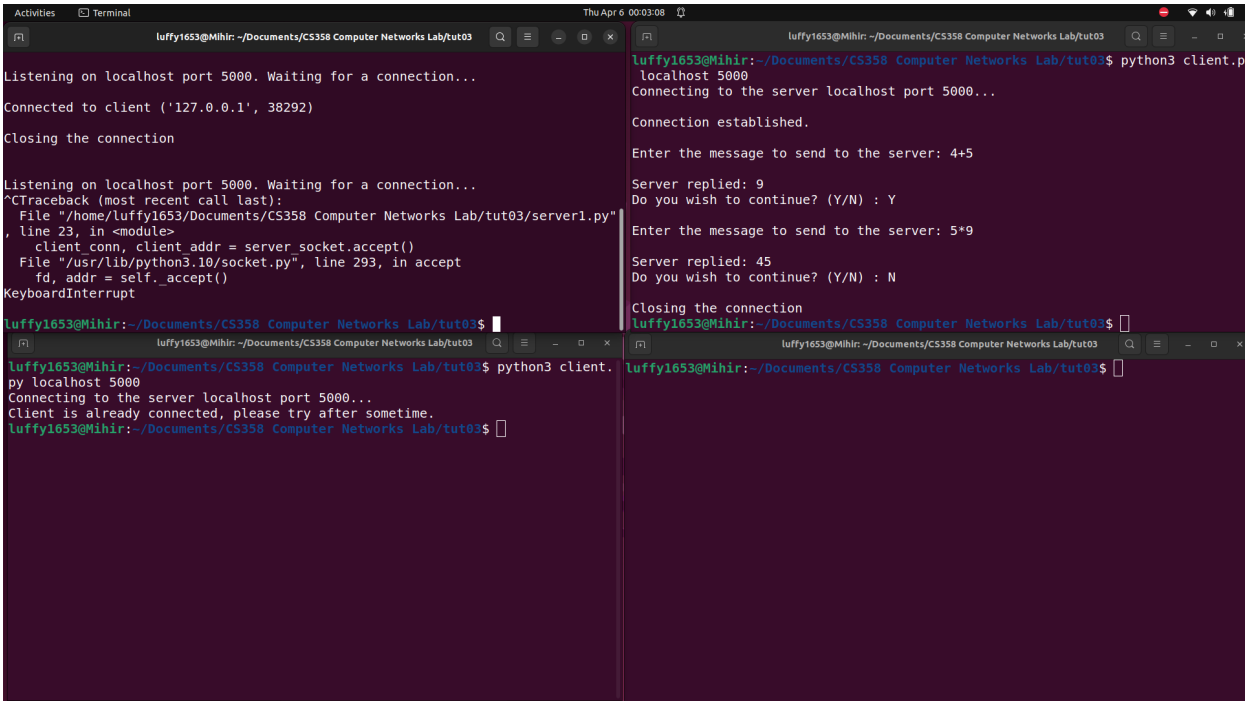
Sending an expression to the server for calculation. Server returns the correct answer.



Attempting to connect two clients at a time. As shown below, the second client is unable to connect and gets an error saying that another client is already connected, please try after sometime.

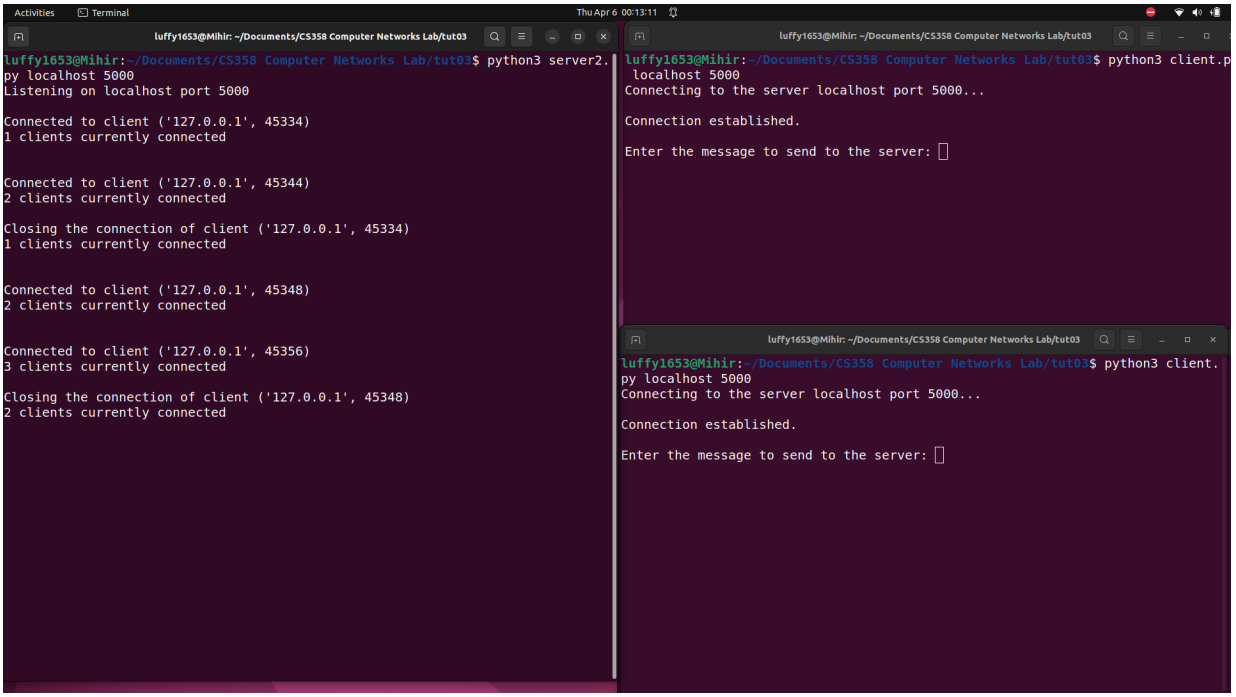


Closing the connection from the client side, by entering N. Further, the server is closed by pressing Ctrl+C.

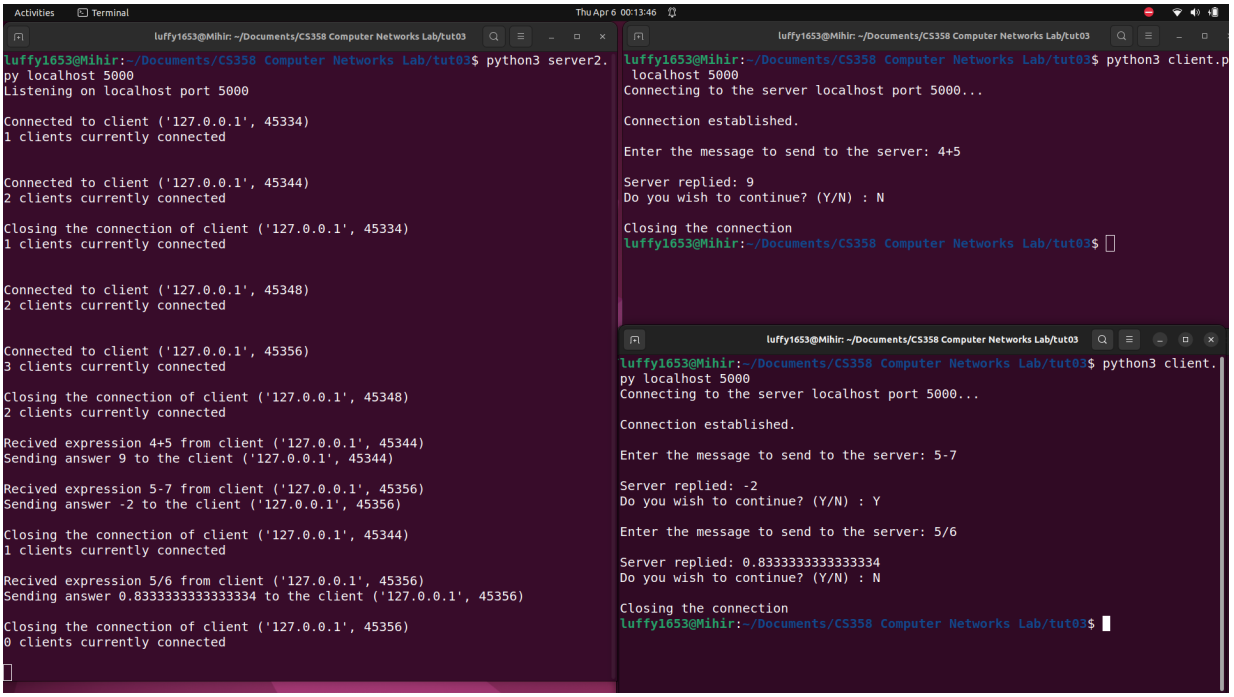


Server 2: Multi-threaded server - handles multiple clients

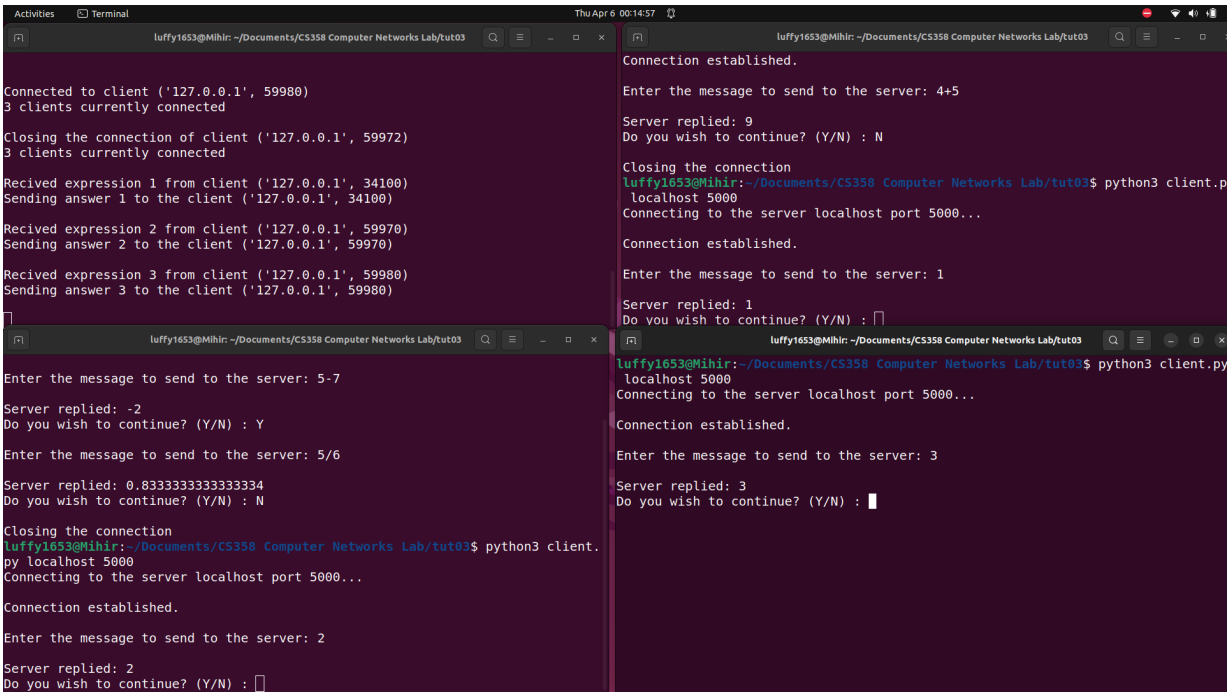
Connecting two clients at a time. As shown below, the connection is successful. Each client is running on a separate thread. Also, the server displays the number of clients currently connected.



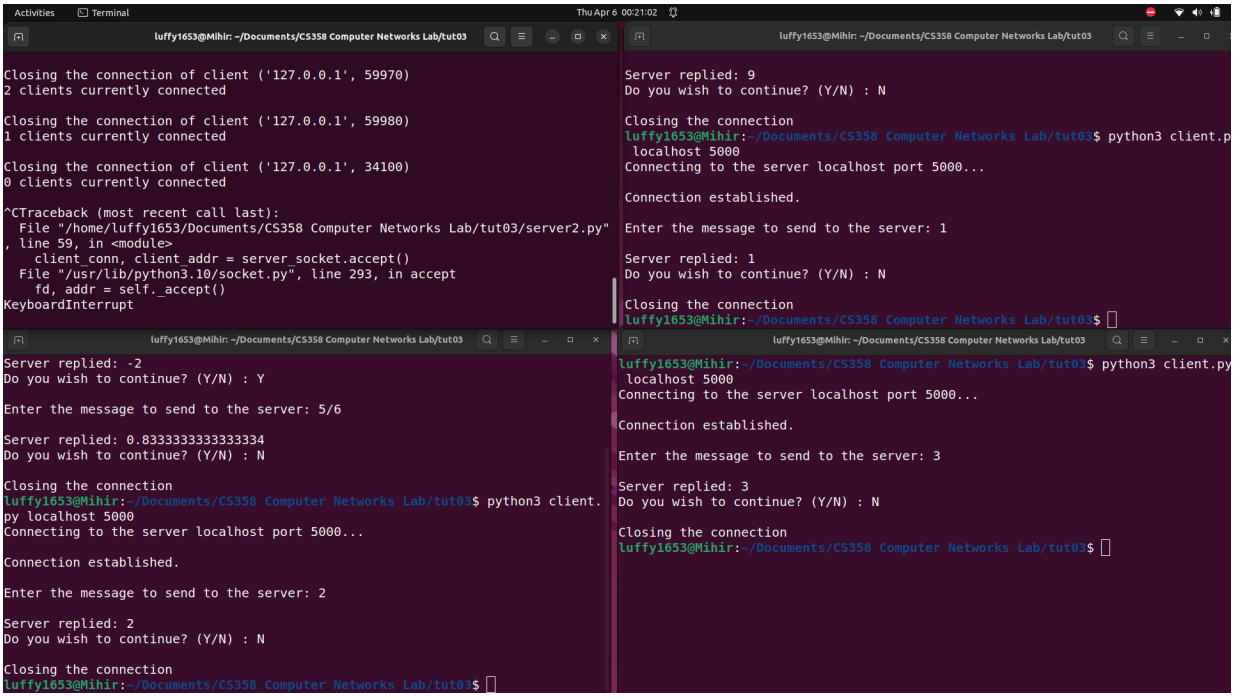
Both the clients send requests to the server. The server answers all the requests. The clients then close the connection by pressing N. All this is also shown in the server as notifications.



Connecting 3 clients to the server at a time. Each client sends an expression to the server, and the server answers all the requests as shown below.

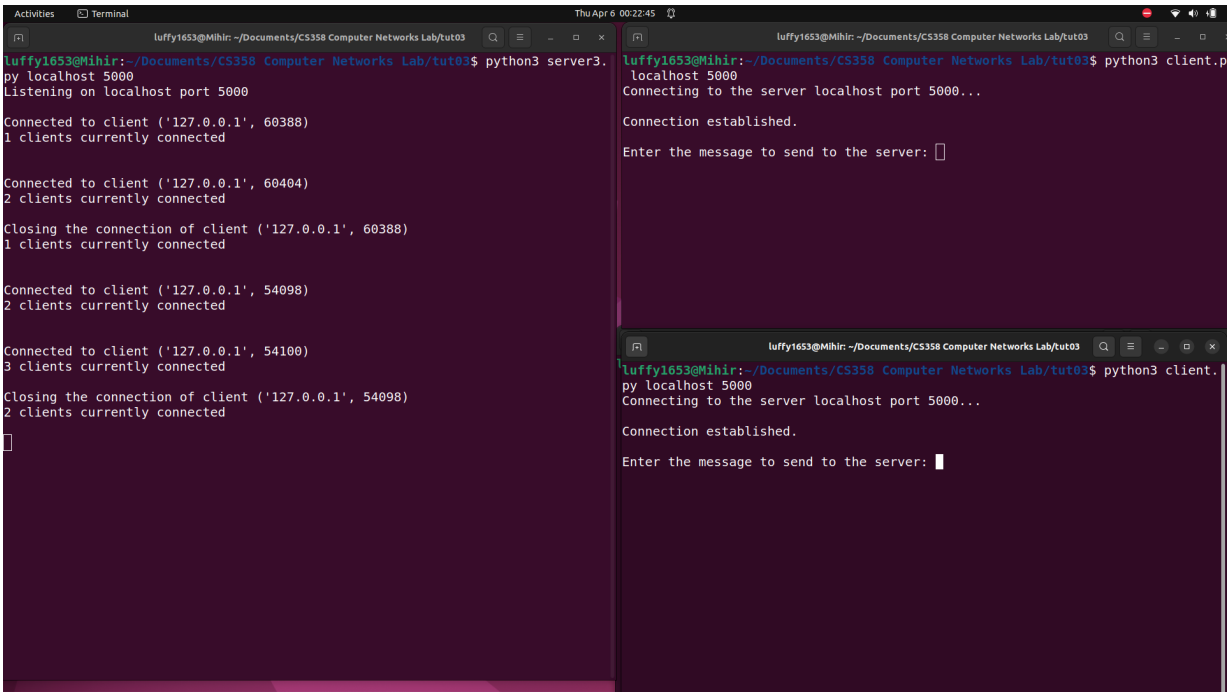


Closing the connection of all the three clients. The server is further closed by using Ctrl+C

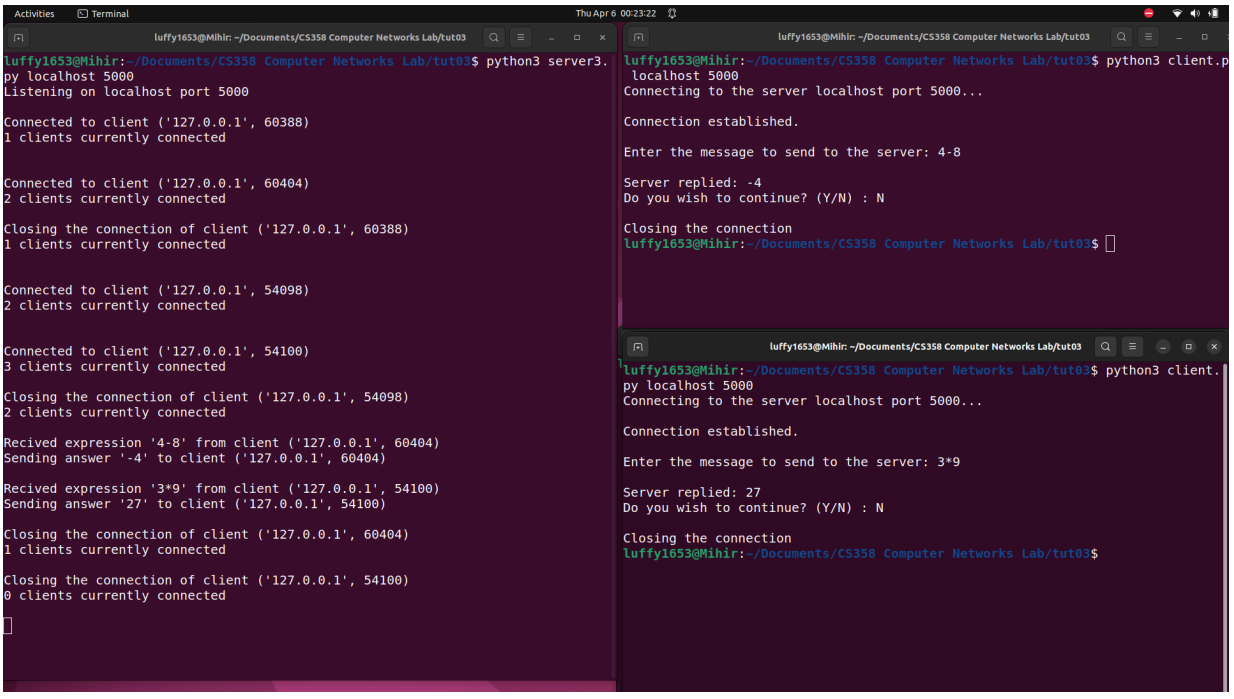


Server 3: Single process server using ‘select’ method - handles multiple clients

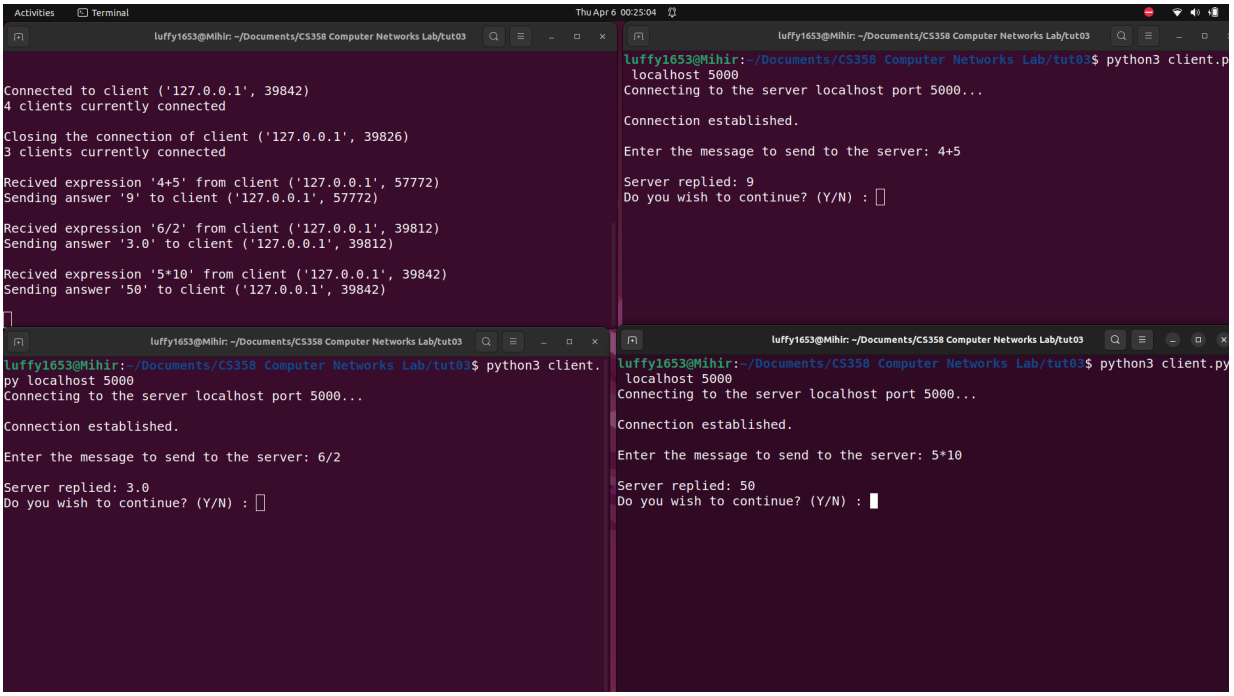
Connecting two clients at a time. As shown below, the connection is successful. Both the clients are running on the same program, on two different sockets. Each client will be assigned a separate socket, message queue, etc and other such data structures. The server stores all the pending input and output requests in separate lists. The select method gives the order of which requests to address at what time, etc.



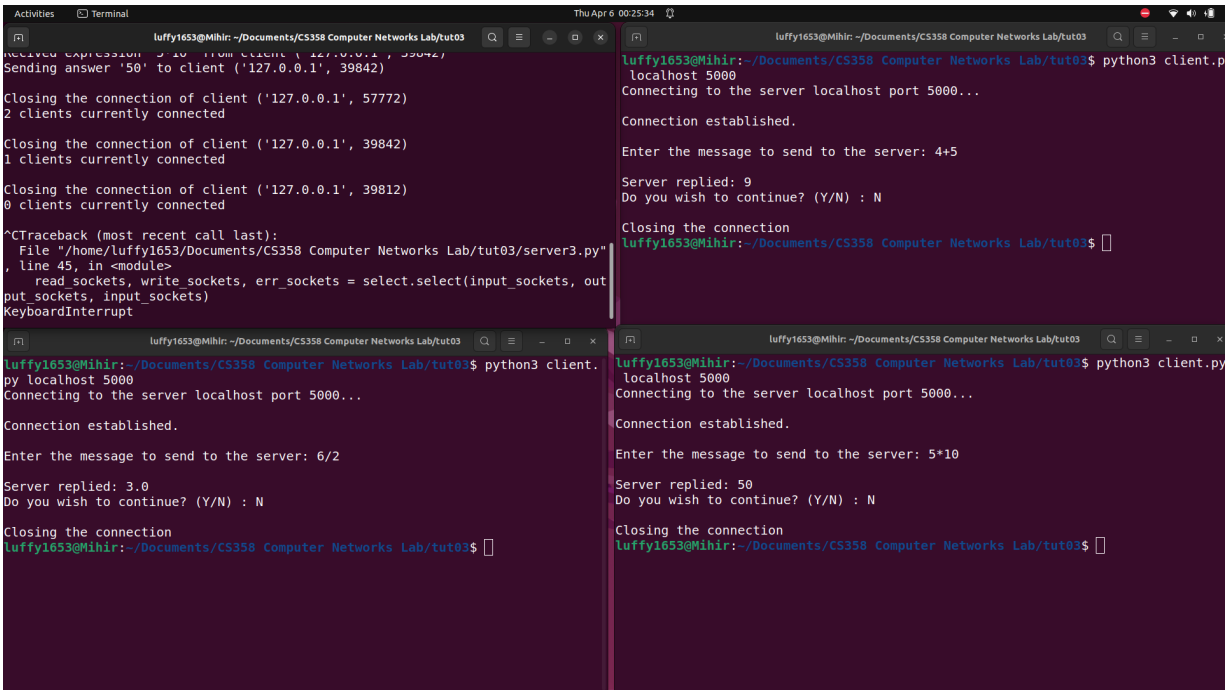
Both the clients send expressions to the server. The server calculates the value and sends replies back to the clients, as shown below. The clients then close the connection by pressing N. This is also notified in the server.



Connecting three clients to the server at a time. All the clients send some requests to the server and receive an answer respectively. This is also notified on the server side.



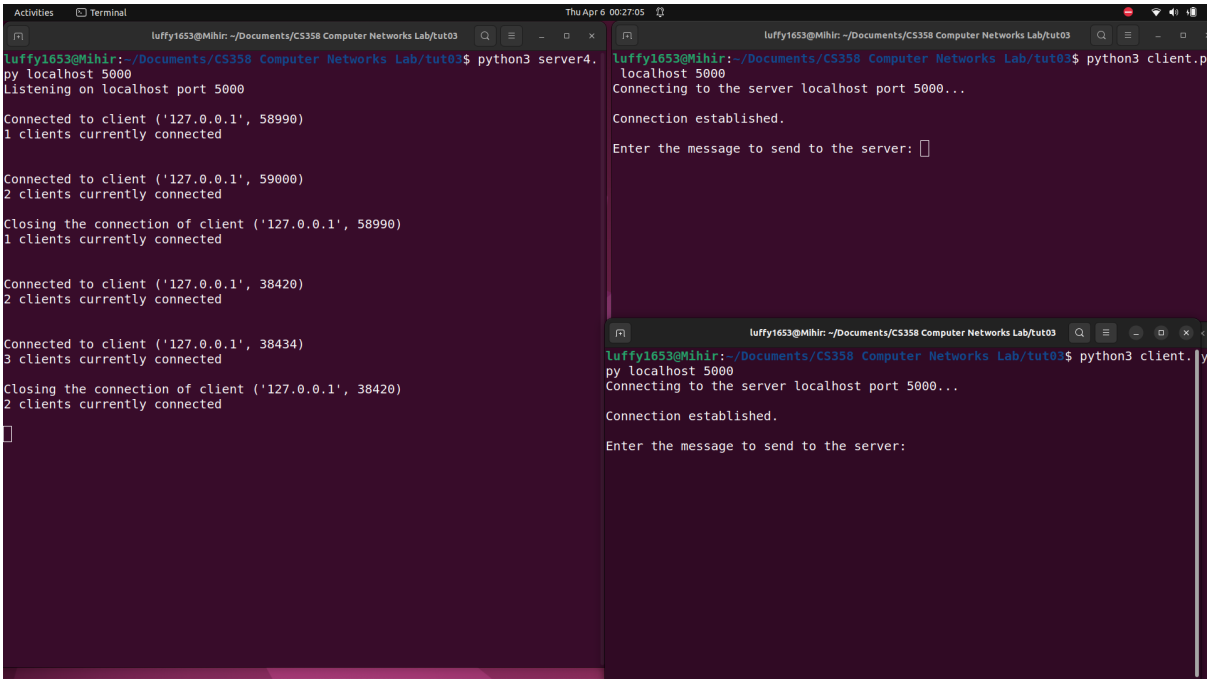
All the three clients close the connection by pressing N. This is notified in the server. Further, the server closes by Ctrl+C.



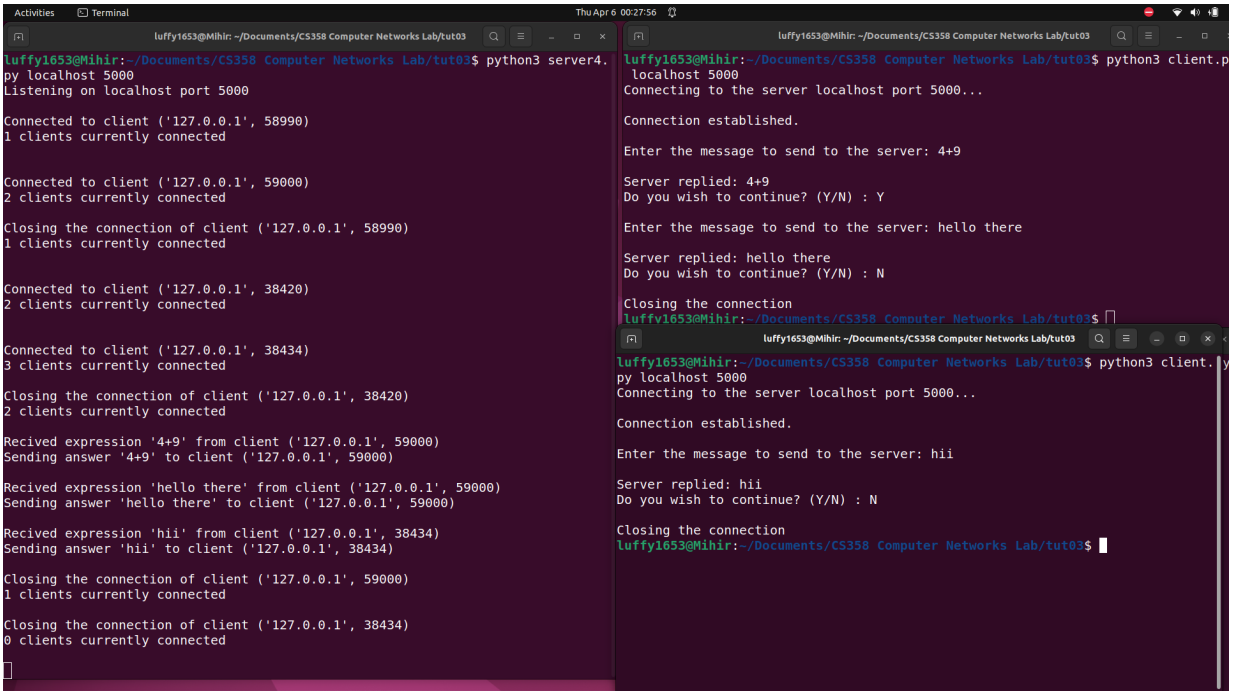
Server 4: Single process echo server using ‘select’ method - handles multiple clients

This server works in the same way as server 3. The only difference is that, this server will echo whatever it receives from the client, back to the client. Basically, an “echo” server.

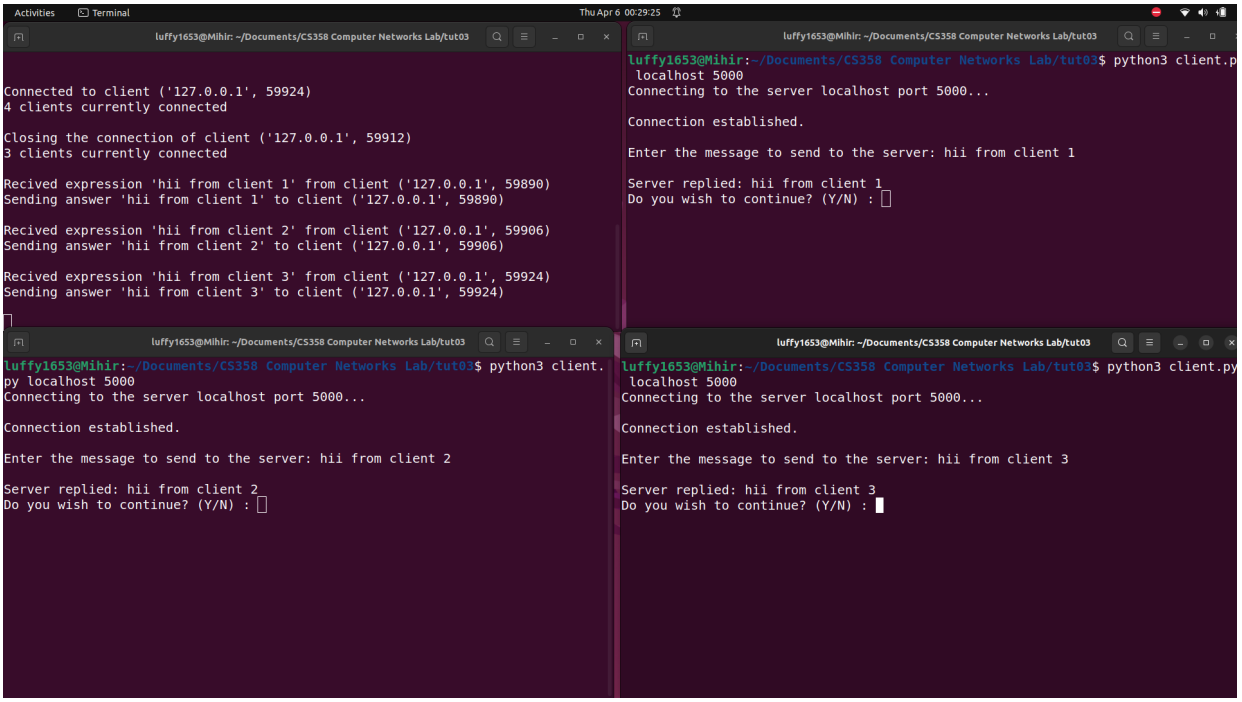
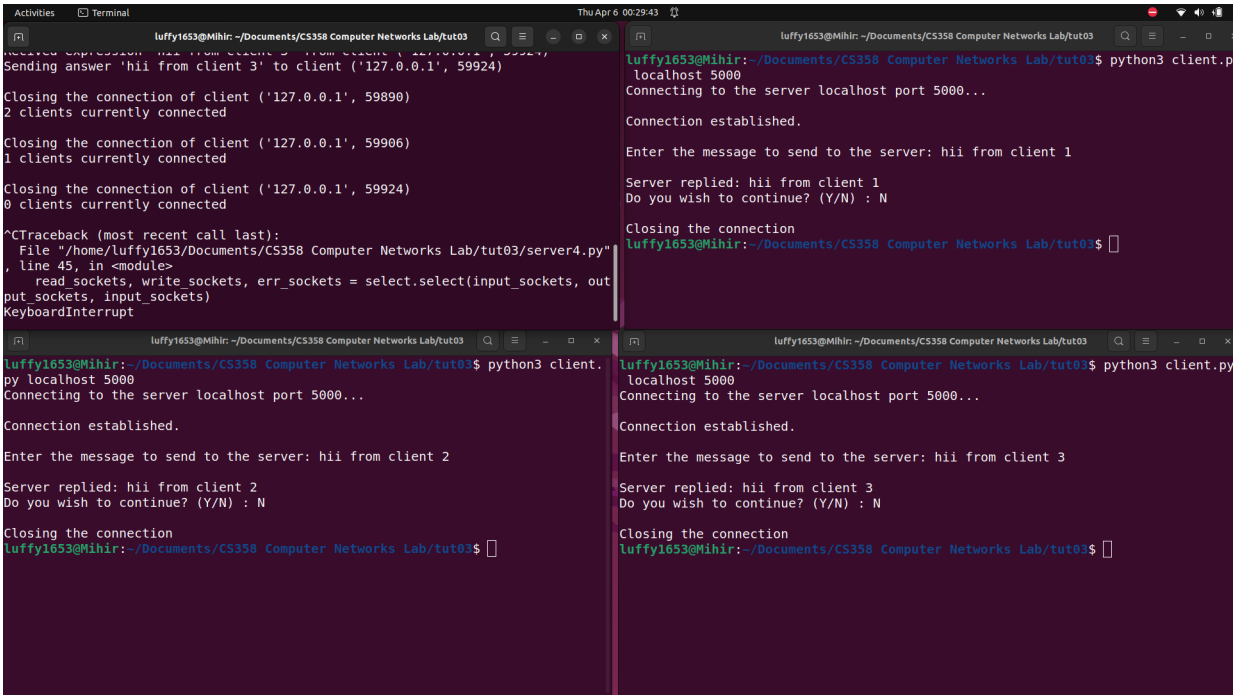
Connecting two clients at a time. As shown below, the connection is successful. Both the clients are running on the same program, on two different sockets. The select method gives the order of which requests to address at what time, etc.



Both the clients send messages to the server. The server receives the messages and echoes them back to the clients, as shown below. The clients then close the connection by pressing N. This is also notified in the server.



Connecting three clients to the server at a time. All the clients send some requests to the server and receive an answer respectively. This is also notified on the server side.



All the three clients close the connection by pressing N. This is notified in the server. Further, the server closes by Ctrl+C.