CP372 Socket Programming Assignment

Report

Ryan Chisholm
Michael Wu

February 10 2025

Description of Source Code

Server

The server.py file contains the source code for the server side of the client-server communication application. This program contains two functions: start_server and handle_client. The start_server function creates a new socket and binds the port and host and opens the socket for listening, it then invokes threading to call the handle_client function and allow for multiple simultaneous client to run. The handle_client function takes the client address, client name and client socket as parameters. It assigns a client id to the client before storing the address and connection time in a local cache, it then allows the server to receive messages from the new client and contains an if statement fall through to handle messages from the client in different scenario's. Lastly, it logs the disconnection time upon termination and closes the socket. Throughout this process, the server keeps track of the number of clients currently running, and terminates if more then 3 clients attempt to connect.

Client

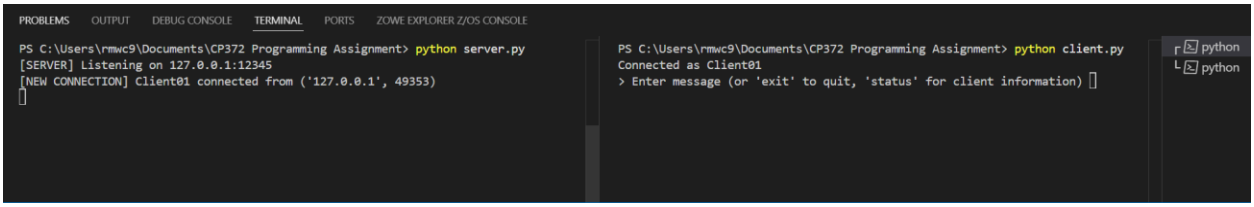
The client.py file contains the source code for the client side of the client-server communication application. This program contains a singular function: start_client. The start_client function starts the client, by connecting to the host and port. It then allows for the user to enter text to be sent to the server. Any message sent from the client will be sent to the server, and then receive confirmation of such, by receiving that same message back with 'ACK' appended to the end to indicate it has been acknowledged by the server. In addition, the start_client function also contains code to close the socket and terminate the client if the server goes over the maximum number of clients.

Challenges and Difficulties

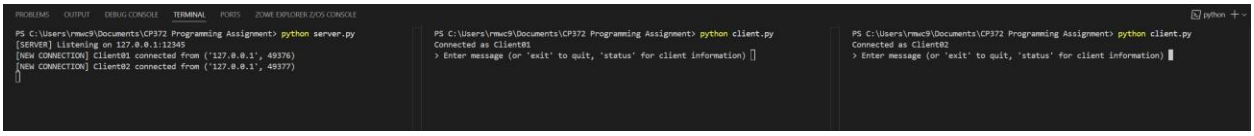
The main challenges we faced throughout the duration of this assignment was the multithreading as well as understanding how the client and server communicate between each other. It took some time to understand the process of communication and learn how to use the socket library and multithreading packages. But in the end we were able to figure it out and implement these concepts to build our communication application.

Testing

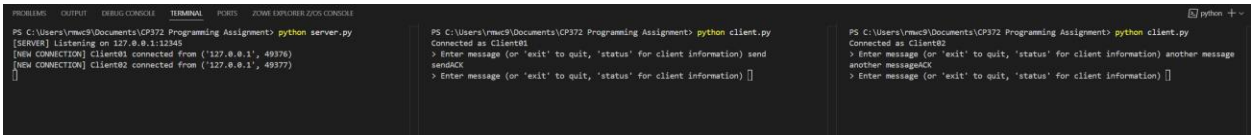
Program can create a server and client



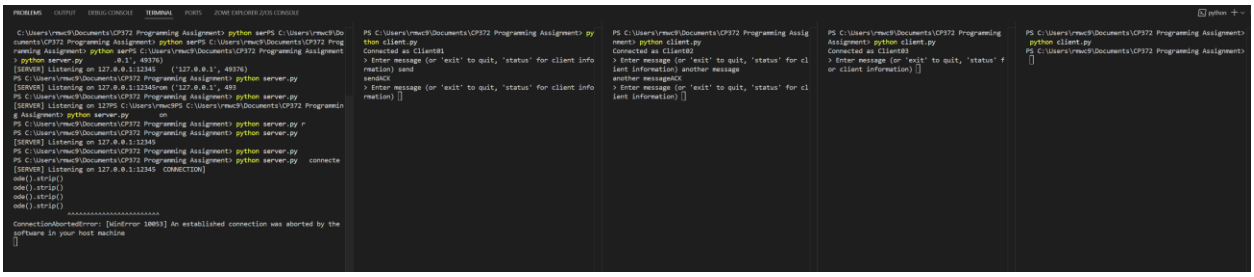
Each client created is assigned a name with correct the number



Server can handle multiple clients

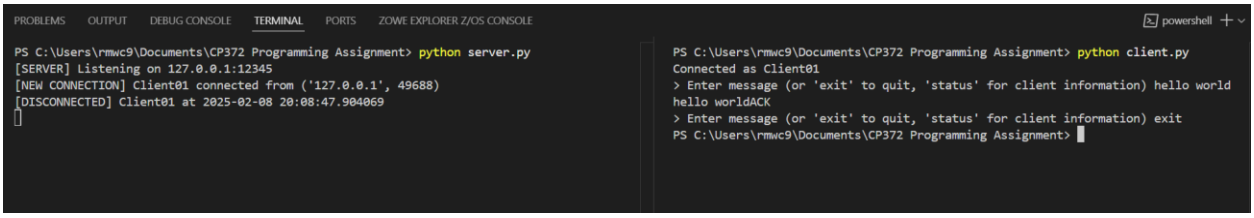


Server limits the number of connected clients to 3

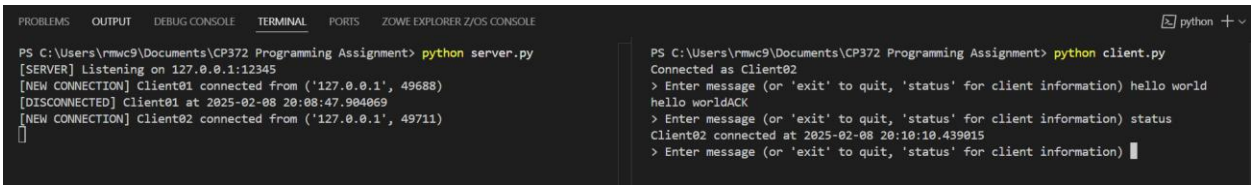


Server and client can exchange messages

Client can send “exit” and server will cleanly disconnect the client



Server maintains clients connection details and sends to client when “status” is requested



Potential Improvements

While we feel that in the allotted time we accomplished everything necessary, had we been given more time, we would have improved the efficiency of the program, as well as the functionality of the bonus requirement to include more types of files.