COMPUTER NETWORKS ASSIGNMENT-3

Rajneesh Kumar
Roll no: 2001CS53

REPORT

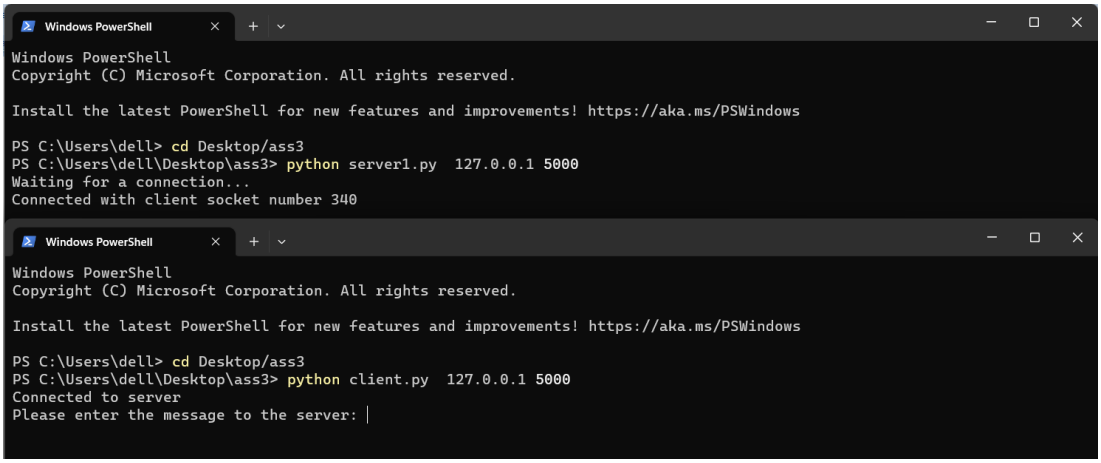
1) *Client:*

A Python code for a TCP client that connects to a server using the socket library. The code takes two arguments from the command line, the IP address of the server and the port number to connect to. If the user does not provide these two arguments, the code prints a usage message and exits.

The code creates a socket and attempts to connect to the server using the IP address and port number provided. If the connection is successful, it enters into a loop where it prompts the user to enter a message to send to the server, sends the message to the server, receives a response from the server, and prints it on the screen. It then prompts the user whether to continue or exit.

If the connection to the server fails or the server is currently connected to another client, the code prints an error message and exits.

Finally, the code closes the socket when the loop ends or when the user decides to exit.



This is a sample of how we connect client to a server

2) *Server1*

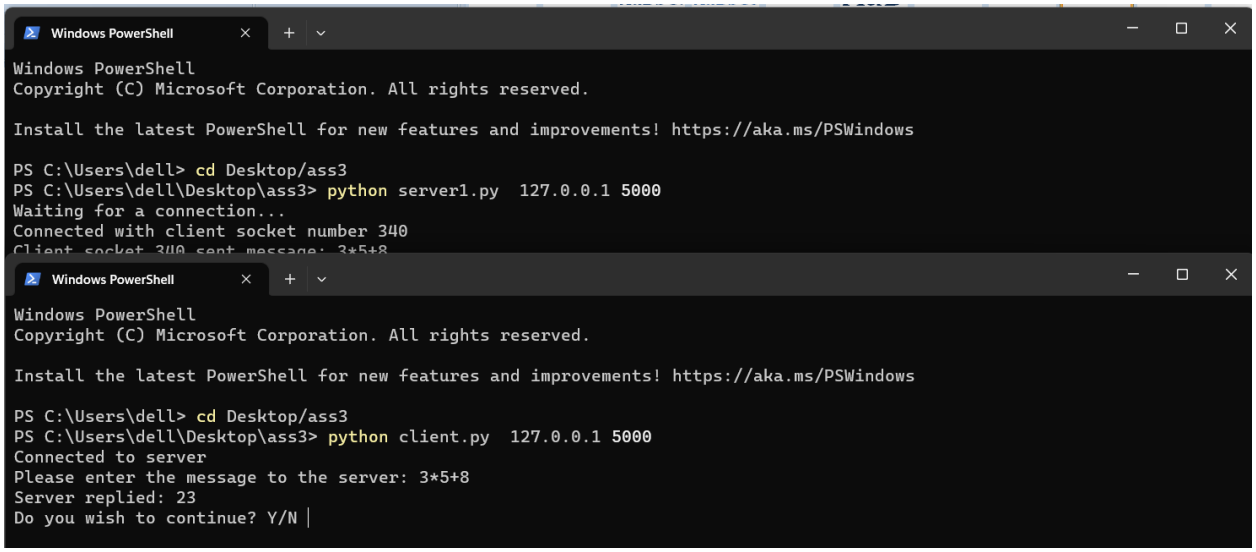
a TCP server that listens for incoming connections from clients using the socket library. The code takes two arguments from the command line, the IP address of the server and the port number to bind to.

The code creates a socket object and binds it to the IP address and port number provided. It then enters into a loop where it waits for incoming connections from clients using the accept() method. Once a connection is established, it prints a message indicating the client socket number.

The server then enters into another loop where it waits for messages from the connected client. It receives the message, evaluates it as an arithmetic expression, and sends the result back to the client. If the input is invalid, it sends an error message back to the client.

If there's an error, the code closes the client socket and breaks the inner loop. When the client disconnects, it prints a message indicating the client socket number that disconnected.

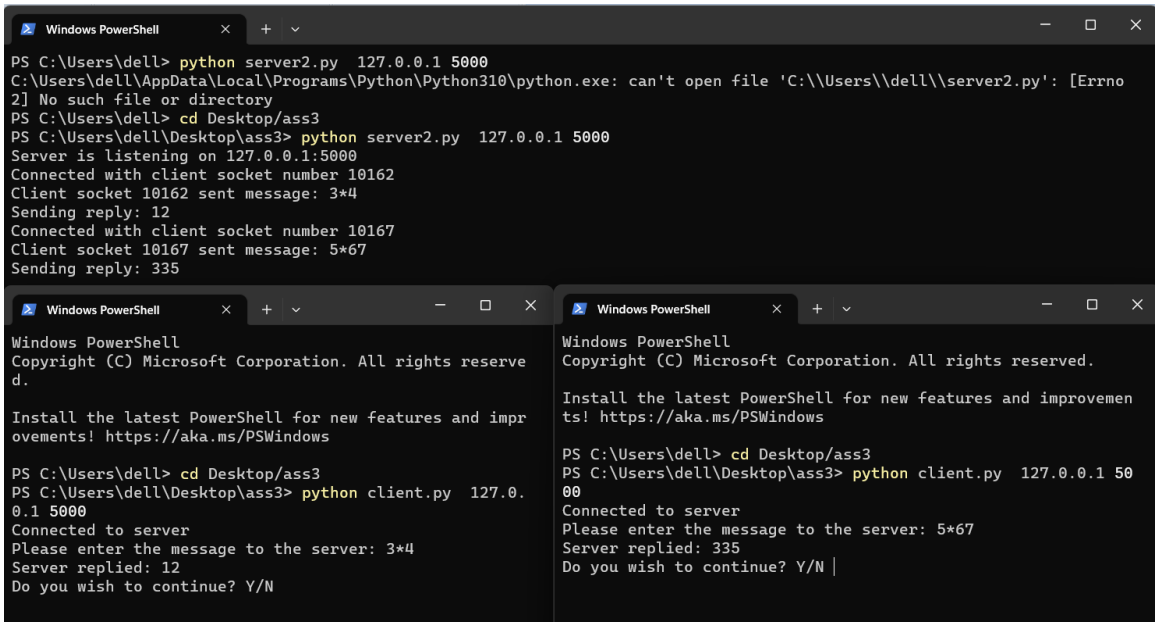
Finally, the code closes the server socket when the loop ends.



a sample of how we connect client to server1

3) *Server2*

The difference between server1 and server2 is it can handle multiple connections at the same time

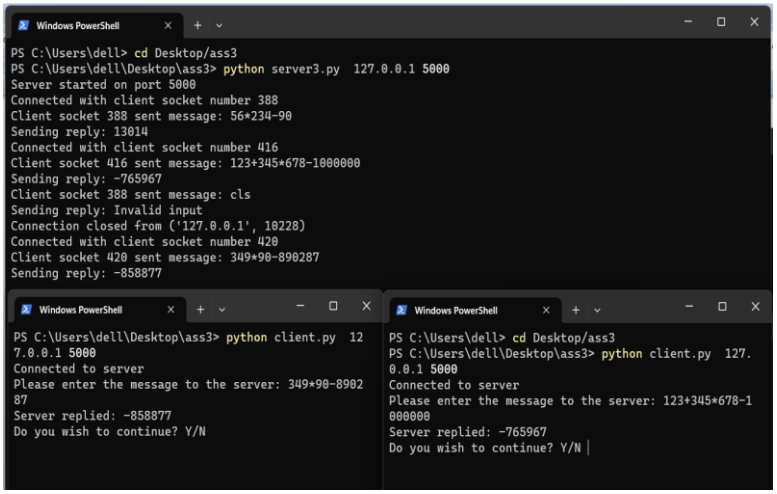


. *server 2 can connect to multiple clients simultaneously*

4) *Server3:*

it's functionality is same as previous servers, It listens for incoming TCP connections on a specified host and port. It uses the select module to multiplex between sockets and handle incoming data from multiple clients simultaneously. When a new client connects, the server adds its socket to a list of sockets

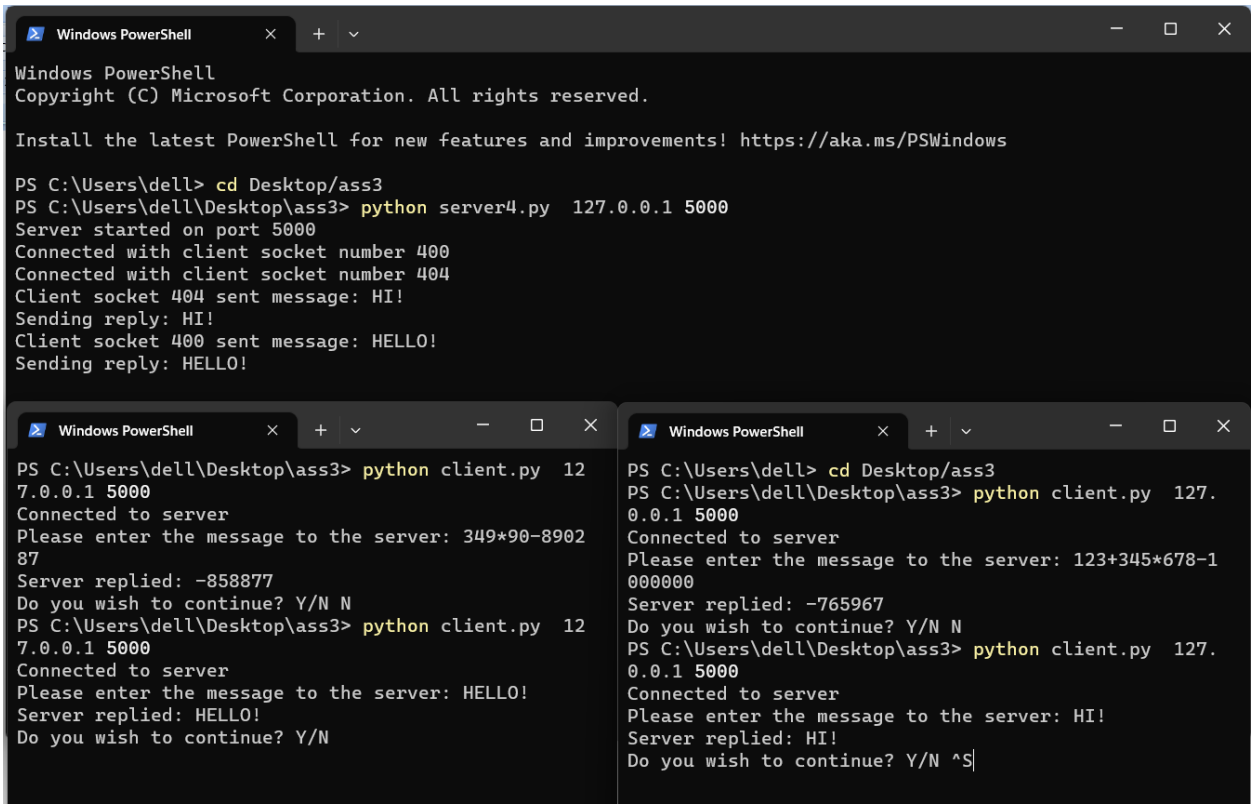
to monitor. When a client sends data, the server parses the arithmetic expression, evaluates it, and sends the result back to the client. If the client disconnects, the server removes its socket from the list and closes the connection. If an exceptional condition occurs on a socket, such as a timeout or a socket error, the server also removes the socket from the list and closes the connection.



Server 3

5) *Server4:*

It echoes the same input to the clients and it can handle multiple clients at a time.



Server 4 replying to the client with same input message