INDIAN INSTITUTE OF TECHNOLOGY, PATNA

Computer Networks Lab: Tut-3

Piyush Singh

2001CS51

OVERVIEW

Implemented 4 types of client server system, where the client chats with a 'calculator' server. and requests for an expression to be evaluated.

- `client.py`: A program for client through which the user requests for an expression to be evaluated. The client prompts the user after each expression whether they want to continue.
- `server1.py`: A program for first server, which can evaluate expressions and handles stricly one client at a time.
- `server2.py`: A program for second server, which can evaluate expressions and is a multi-threaded server, and can handle multiple clients concurrently.
- `server3.py`: A program for third server, which can evaluate expressions and is a single process server which can handle multiple clients concurrently using the `select.select()` method.
- `server4.py`: A program for fourth server, which echoes back the message as received by the client and is a single process server which can handle multiple clients concurrently using the `select.select()` method.

FUNCTIONALITIES IMPLEMENTED

- All the servers except the echo server, can evaluate an expression having a valid syntax, including expression involving multiple operands, parenthesis etc. For instance, it can evaluate expressions like, "$5+(21-(-2+11)$", "$2/4+(27-9)$" etc.
- All 4 servers and the client have been implemented as specified.
- The code has several `try-except` blocks, for cases like, 'Invalid IP or Port Number', 'Permission Denied for the specified port number', 'invalid syntax for the expression to be evaluated'.

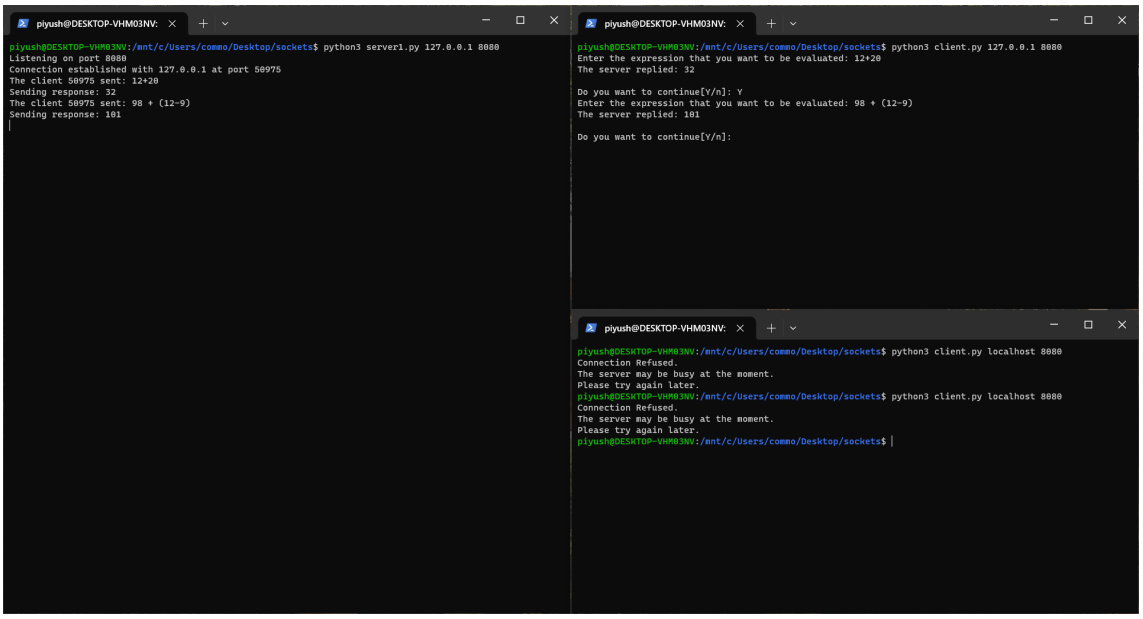
INSTRUCTIONS TO RUN

- The client should be run as
  `python3 <File Name> <IP Address> <Port Number>`

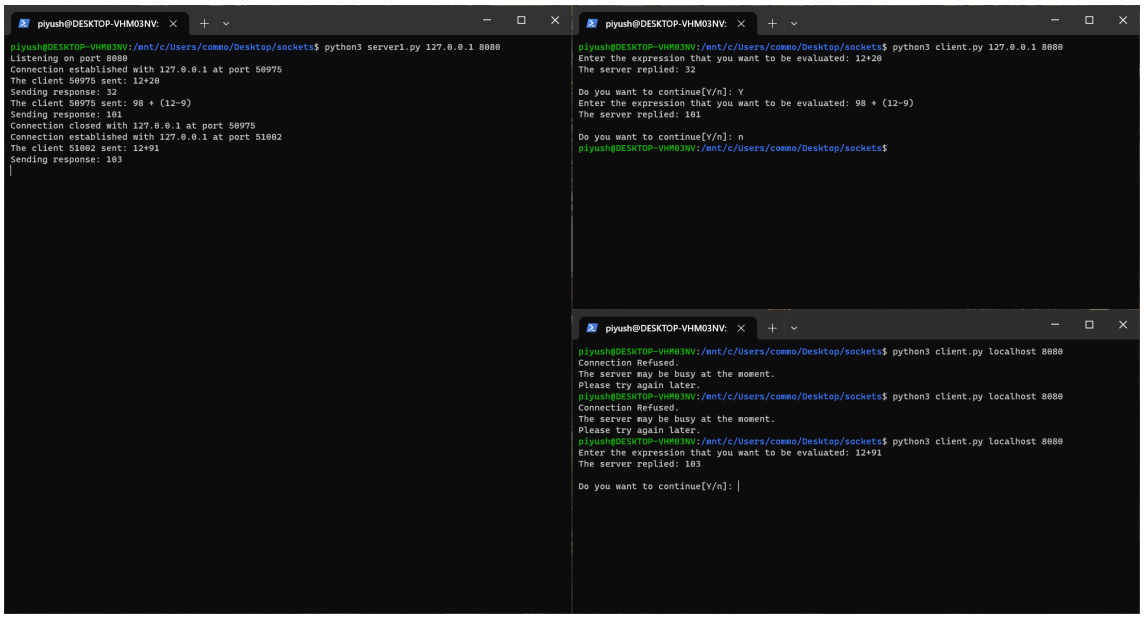
- The server should be run as
  `python3 <File Name> <IP Address> <Port Number>`

  If hosted on localhost, 'localhost' can also be specified for IP instead of `'127.0.0.1'`. For instance 'python3 client.py localhost 8080'

SERVER 1

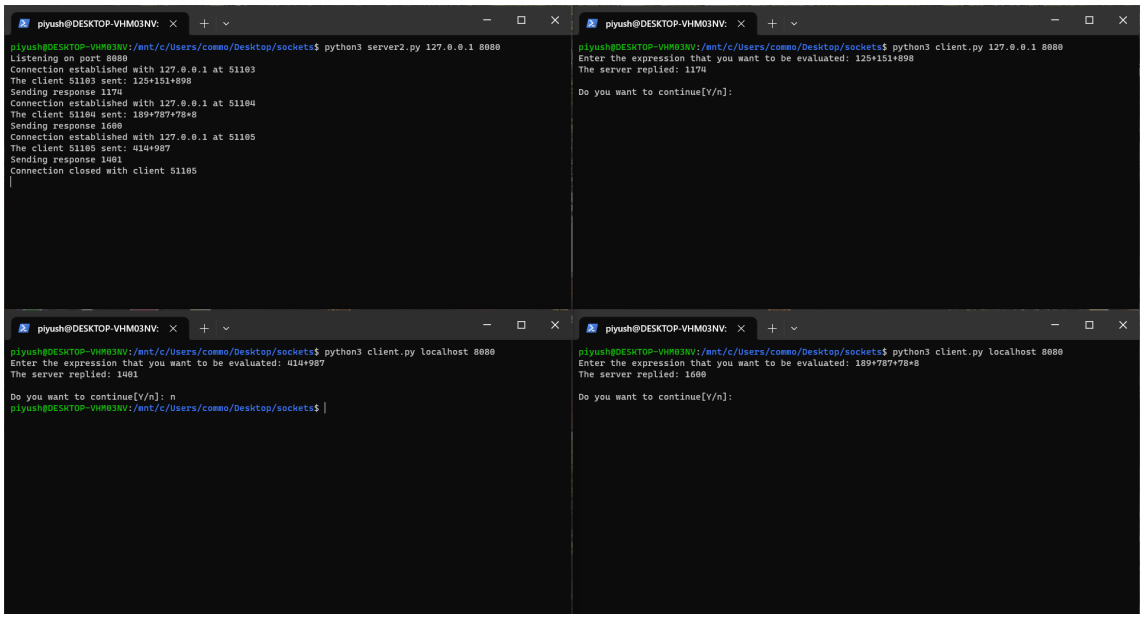


Connection Request refused for the second client (bottom right) while the first client is running(top right)



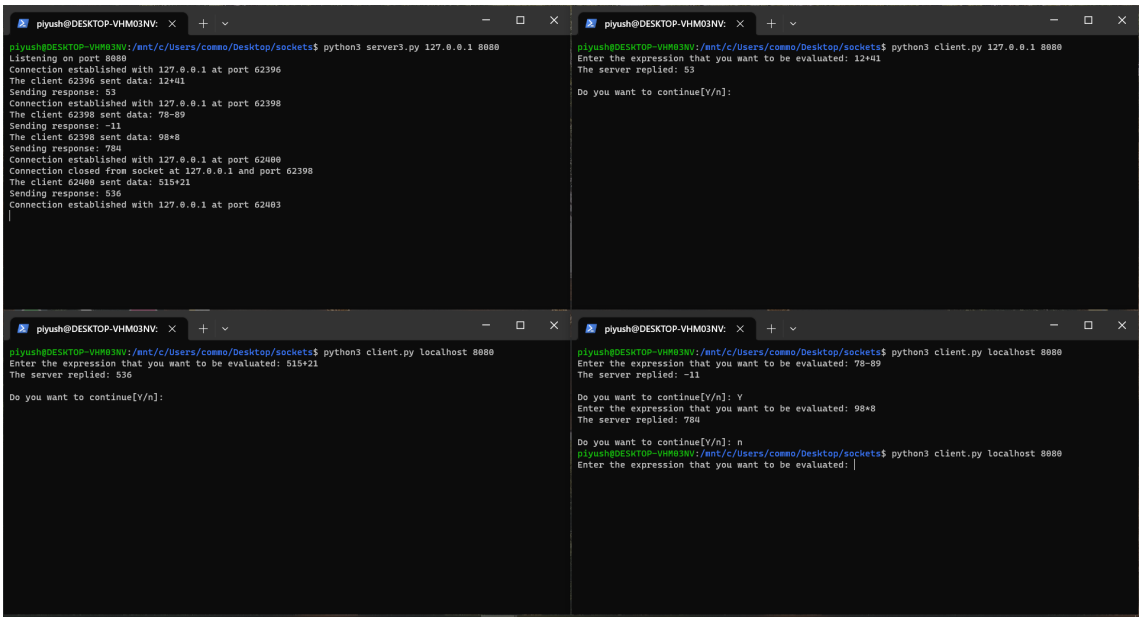
Second client is able to connect after the first client has terminated.

SERVER 2



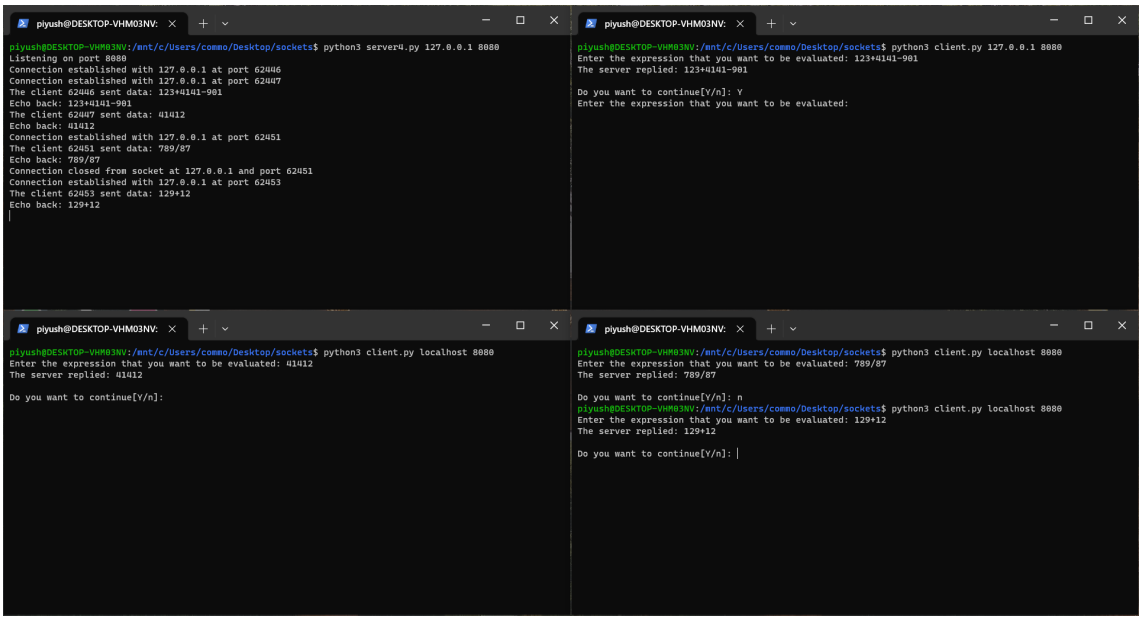
All three clients are able to connect simultaneously and the server handles all of them concurrently. The server uses multi-threading

SERVER 3



All three clients are able to connect simultaneously and the server handles all of them concurrently. The server uses the `select.select()` method to implement this.

SERVER 4



This is an echo server. All three clients are able to connect simultaneously and the server handles all of them concurrently. The server uses the `select.select()` method to implement this.