Lab 2 Report

CSL 6010 - Cyber Security

Rahul Barodia

B20CS047

**** I have MacBook, so Ubuntu was not installed on it. That’s why I have used VS code to execute these codes and the output is shown in the terminal of VS code itself ****

a) Single Process Server

server1.py

```
#Importing the socket library
import socket


HOST = "127.0.0.1"
PORT = 2002  #socket server port number


#creating a socket object and passing two parameters to it
with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as ser:


#AF_INET refers to the address family ipv4
# SOCK_STREAM refers to onnection-oriented TCP protocol



   ser.bind((HOST, PORT)) # bind to the port
   ser.listen() # putting the socket into listening mode
   print('Socket is listening..')
   connection, addr = ser.accept()  # establishing connection with client
```

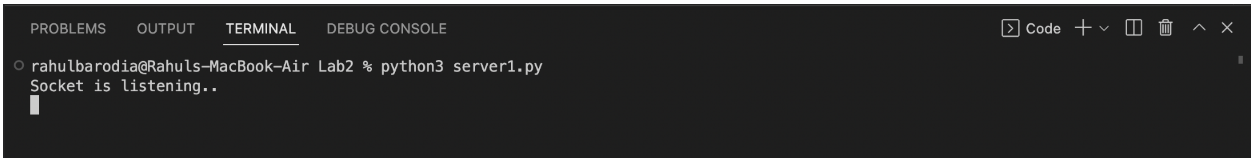
```
    with connection:
        print(f"Connected by {addr}")
        #running a infinite loop until we interrupt or error occurs
        while True:
            data = connection.recv(1024)
            if not data:
                break
            s1=data
            res=eval(s1)  # eval solves mathematical expressions
            s1_val=str(res) # passing a string to eval
            connection.sendall(s1_val.encode()) # Encoding to send byte type
```

client1.py

```
#Importing the socket library
import socket
HOST = "127.0.0.1"
PORT = 2002  #socket server port number

with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as ser:
    ser.connect((HOST, PORT))  #connecting to the server
    comm=input("Enter the expression:") #asking for input expression
    ser.sendall(bytes(comm,encoding='utf8'))
    data = ser.recv(1024)

print(data.decode('utf-8')) #print the ans from server
```

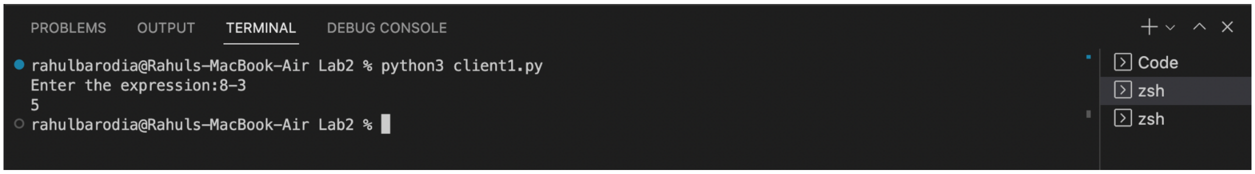
Output:

Server:

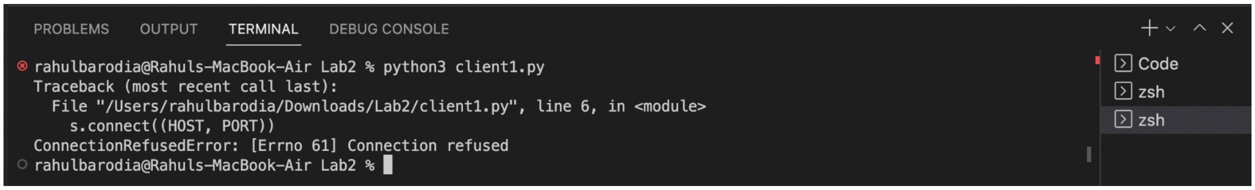


```
rahulbarodia@Rahuls-MacBook-Air Lab2 % python3 server1.py
Socket is listening..
```

Client:



```
rahulbarodia@Rahuls-MacBook-Air Lab2 % python3 client1.py
Enter the expression:8-3
5
rahulbarodia@Rahuls-MacBook-Air Lab2 %
```



```
rahulbarodia@Rahuls-MacBook-Air Lab2 % python3 client1.py
Traceback (most recent call last):
  File "/Users/rahulbarodia/Downloads/Lab2/client1.py", line 6, in <module>
    s.connect((HOST, PORT))
ConnectionRefusedError: [Errno 61] Connection refused
rahulbarodia@Rahuls-MacBook-Air Lab2 %
```

As we can see from the above images that when a 2nd client tries to access the server, the connection is refused. Thus server1 can handle only a single client at a time.

b) Multi-Process Server

server2.py

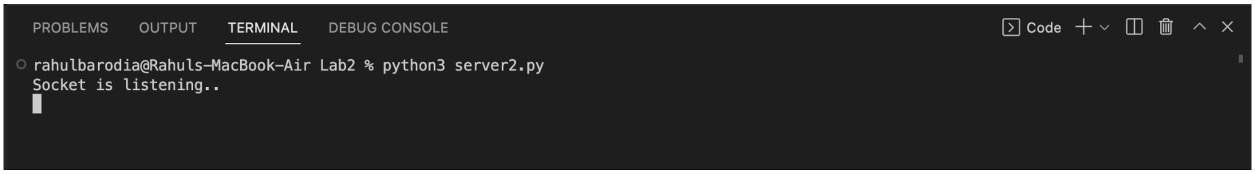
```
#Importing the essential libraries
import socket
import os
from _thread import *
#creating a socket object
ser = socket.socket()
host = '127.0.0.1'
port = 2010  #socket server port number
#creating a variable named ThreadCount to count the no. of running processes
ThreadCount = 0
try:
   ser.bind((host, port)) # bind to the port
except socket.error as e:
   print(str(e))

ser.listen(5) # putting the socket into listening mode
print('Socket is listening..')

def multi_threaded_client(connection):
   connection.send(str.encode('Server is working:'))
   while True:
       data = connection.recv(2048)
       response = 'Server message: ' + data.decode('utf-8')
       if not data:
           break
       s1=data
       res=eval(s1)
       s1_val=str(res)
       connection.sendall(s1_val.encode())
   connection.close()

while True:
   Client, address = ser.accept()
   print('Connected to: ' + address[0] + ':' + str(address[1]))
   start_new_thread(multi_threaded_client, (Client, ))
   ThreadCount += 1 #increementing variable
```

client2.py

```
#Importing the socket library
import socket

mul = socket.socket() #creating a socket object

host = '127.0.0.1'
port = 2010  #socket server port number
print('Waiting for connection response')
try:
   mul.connect((host, port))
except socket.error as err:
   print(str(err))
res = mul.recv(1024)

while True:
   Input = print('Enter your expression:')
   mul.send(str.encode(Input))

   res = mul.recv(1024)

   print(res.decode('utf-8'))

mul.close()  #close the connection
```

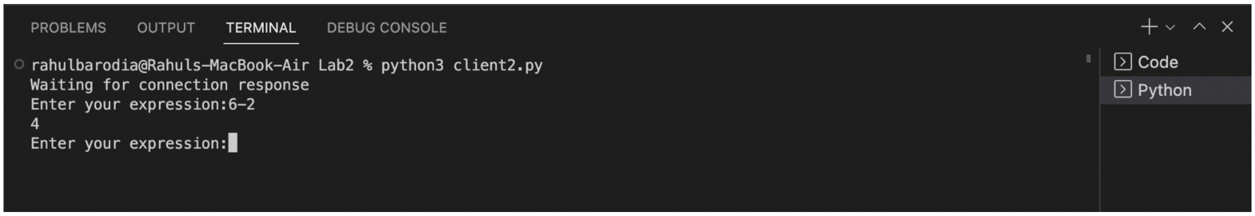
Output:

Server:

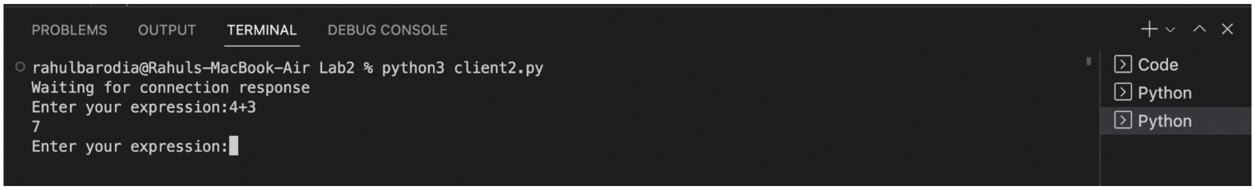


```
rahulbarodia@Rahuls-MacBook-Air Lab2 % python3 server2.py
Socket is listening..
```

Client:



```
rahulbarodia@Rahuls-MacBook-Air Lab2 % python3 client2.py
Waiting for connection response
Enter your expression:6-2
4
Enter your expression:
```



```
rahulbarodia@Rahuls-MacBook-Air Lab2 % python3 client2.py
Waiting for connection response
Enter your expression:4+3
7
Enter your expression:
```

As we can see from the above images, server2 allows multiple clients to access at the same time. The above example shows two clients accessing server2 and getting the correct response.