Milestone 1

2 October 2024

Deadline: Wednesday 16 October 11:59 pm

Important Note: Read the submission guideline well for software requirements and submission instructions

1 Requirements

In this Milestone , you are required to build a chatting application designed over a network composed of several clients and a server. You should edit your code of milestone 0 to allow for parallel connection of clients. The chatting application should feature the following:

1. The clients chat with the server at any time.
2. The server responds with the same message sent by the corresponding client in CAPITALIZED format.
3. Use multi-threading so the server can communicate with N clients at the same time.
4. Connection with any client should not be affected by other clients (independent connections).
5. The server should save the client information such as the connection socket and a unique ID for each client.
6. The connection between any client and server stays open until the client sends to the server a message contains CLOSE SOCKET then the connection between the server and the client closes.
7. The chatting connections are TCP based. Note: Server WILL NEVER TERMINATE * For Example: “When Client 1 sends any message to the server then terminates, the server should continue running (should stay available), so when you re-run Client code you should succeed connecting to the server again as client 2”

2 Server Code

```python
import socket
import select
import sys

#initiate server socket with the TCP connection
server_socket = socket.socket(socket.AF_INET,socket.SOCK_STREAM)

# binding the server socket with the localhost as ip and port number
port=5605
server_socket.bind(('127.0.0.1',port)) # '127.0.0.1' is the localhost in ipv4

# make the socket listen on this port
server_socket.listen(...)
def thread(c):
     while True:
        # recieve meassage as bytes
        # ( write your code)

        # decoding the bytes into characters
        # ( write your code)

        #Check if the message was 'CLOSE SOCKET' to close connection
        # ( write your code)

        # otherwise capitalize the decoded message
        # ( write your code)

        # send the response as bytes again
        # ( write your code)
        client.send(...)

        # Break the connection when 'CLOSE SOCKET' is recieved

def Main():
    # listening forever
    # ( write your code )

    # lock acquired by client
    threading.Lock().acquire()
    start_new_thread(thread(),..)

if __name__ == '__main__':
    Main()
```

3 Client Code

```
# Python program to implement server side of chat room.
import socket
import select
import sys

#initiate Client socket with the TCP connection
client_socket = socket.socket(socket.AF_INET,socket.SOCK_STREAM)

# binding the client socket with the localhost as ip and port number
port=5605
# try to connect to the server with associated port and id
client_socket.connect(('127.0.0.1',port)) #'127.0.0.1' is the localhost in ipv4

# open a connection until sending CLOSE SOCKET
while True:
    message=input("enter your message: ")

    # send message as bytes
    # (write your code)

    #recieve respose if exists
    # (write your code)
```

4 Submission

Deadline: Tuesday 16 October at 11:59 pm

Milestone 1 is an individual task, Any cheating case will be graded ZERO.

Read the submission guideline well before submitting the milestone!

1. Your project or notebook MUST be named as MS1_FirstName_LastName_id
   Ex: (MS1_Lucian_Youssef_58-1234).
2. Save your notebook as a copy in your Drive or upload the project in a zipped folder along with screenshots for the output.
3. I WILL NOT ACCEPT SUBMISSIONS THROUGH E-MAIL.

Goodluck!