

## ASSIGNMENT 2

### Multi-user chat system

#### Description of the code:

My folder consists of:

1. Server code
2. Client Code
3. Makefile

The communication method between the server and various clients chosen by me is Domain Sockets. The communication between the server and client takes place by giving the same path to both of them.

#### 1. Server Code:

In this code, a socket is opened using *AF\_UNIX* protocol. The code has an array of type *client*(a struct) which stores every new client requesting to connect to the server. There is an array of type *pthread\_t* which stores every thread created to execute the *connect\_clients* function. This function handles all the inputs given by the user. For reading the message, I have used *recv()* function to receive the message. Then, I compared if the string was '1' or '2' or '3' or '4'. After this, the further sub parsing according to the menu takes place.

#### 2. Client Code:

This code also creates a socket using *AF\_UNIX* protocol. Then the client is connected to server through *connect()* function. After this input is taken from the user based on the menu provided to them.

#### 3. Makefile:

Makefile is created to compile server and client's code.

#### Compiling:

Use the *make* command on your Linux Terminal to compile the program using Makefile. Run the server code on one terminal, and all client code on different terminals.

#### Testing and Inputs to be given:

The user is displayed the menu as follows after successful socket creation, binding and listening:

- 1) Send a message to other client through client number
- 2) Send the message to all the clients
- 3) See available clients
- 4) Exit from the client

The user should give input as:

- “1 (Client Number) (Message to be sent)” to send the message to individual client through client number.
- “2 (Message)” to send the message to all the clients
- “3” to see the list of available clients
- “4” to Exit from the current client

### **Expected Outputs:**

- If input was given as “1 (Client Number) (Message to be sent)”, the user can see their message on the terminal of client whose client number is given.
- If input was given as “2 (Message)”, the user can see message on all terminals of clients as well as a display on server that message has been sent to all the clients.
- If input was given as “3”, the user can see all the available clients along with their IDs on server’s terminal.
- If input was given as “4”, the code for client would terminate on that terminal with “Exiting” message and on the server’s terminal, display message on Connection broke as well as removal of client from the array takes place.

### **Error values and Interpretation:**

#### **Input errors:**

- If an incorrect input is given by the user, an error message of “Incorrect Input given” is displayed.
- If the user enters the Client Number which is not available, “Incorrect Client number entered” is displayed.

#### **Other errors:**

- If there is an error in socket creation it is displayed.
- If there is an error in binding, it is displayed.
- If there is an error in listening in the sockets.
- If the file exists or not on which the socket is opened for the client side code.
- If there is an error in sending or receiving among the clients and server.

### Images for the test cases and output:

The screenshot shows a Linux desktop environment with a dark theme. The top panel displays the system clock as 6:57 PM and various status icons. The desktop background is a dark, abstract image. On the left side, there is a vertical dock containing icons for a web browser, a file manager, a terminal, and several application launchers. Three terminal windows are open, each showing the execution of a program named 'myclint'.

The first terminal window (top left) shows the program starting and waiting for clients. It has received three connections, each with a client number (1, 2, and 3). It has also received a message from client 1 and sent a message to client 3.

The second terminal window (top right) shows the program starting and waiting for clients. It has received one connection from client 1 and sent a message to client 1.

The third terminal window (bottom right) shows the program starting and waiting for clients. It has received one connection from client 1 and sent a message to client 1.

The bottom of the screen shows a taskbar with icons for various applications, including a web browser, a file manager, a terminal, and several application launchers.

The screenshot displays a Linux desktop environment with a dark theme. On the left side, there is a vertical dock containing icons for a web browser, a file manager, a terminal, a presentation viewer, a calculator, an Amazon logo, a settings gear, and a terminal window. The top of the desktop features a system tray with icons for network, volume, and battery, along with a clock showing 6:58 PM. Three terminal windows are open, each showing a different perspective of a client-server application. The top terminal window, titled 'pragya@pragya-VirtualBox: ~/Desktop', shows a server process running. It receives messages from clients and broadcasts them. The middle terminal window, titled 'pragya@pragya-VirtualBox: ~/Desktop', shows a client process running. It connects to the server and sends messages. The bottom terminal window, titled 'pragya@pragya-VirtualBox: ~/Desktop', shows another client process running. It connects to the server and sends messages. The desktop also shows a system tray with icons for network, volume, and battery, and a clock showing 6:58 PM.

```
pragya@pragya-VirtualBox: ~/Desktop
Message received from Client 1
Message sent to the client 3
Message received from client 3
Message in broadcast sent to client 1
Message in broadcast sent to client 2
The clients available are:
Client 1 with ID 4
Client 2 with ID 5
Client 3 with ID 6
Connection broken from Client 3
The clients available are:
Client 1 with ID 4
Client 2 with ID 5

pragya@pragya-VirtualBox: ~/Desktop
./myclint
Connection established .....
.....
1) Send a message to other client through client number
2) Send the message to all the clients(Create a broadcast message)
3) See available clients
4) Exit from the client
1 3 hi, my name is Pragya
This is to test broadcasting

pragya@pragya-VirtualBox: ~/Desktop
./myclint
Connection established .....
.....
1) Send a message to other client through client number
2) Send the message to all the clients(Create a broadcast message)
3) See available clients
4) Exit from the client
hi, my name is Pragya
This is to test broadcasting
Exiting!
pragya@pragya-VirtualBox: ~/Desktop$
```

