TCP Multi-Client Chat Server

- Compile server.c and client.c programs
- Run the server by using ./server
- We can note the PORT to which the server is Listening for Client Connections as below:

```
gokulsreekumar@ubuntu:~/Codes/Networks/TCP_Chat$ gcc server.c -o server && ./server
TCP Server Bound to Port: 8080
TCP Server Waiting for Client Requests...
```

- Run the client program by using ./client in another terminal
- The user is **prompted to Enter a Username.**
- Once Username is entered, a WELCOME TO CHATROOM message is shown and in the Server side, details about new connection (PORT and IP of client) is shown.

Client-1:

Server:

```
gokulsreekumar@ubuntu:~/Codes/Networks/TCP_Chat$ gcc server.c -o server && ./server
TCP Server Bound to Port: 8080
TCP Server Waiting for Client Requests...
New connection at 127.0.0.1 : 42876
```

- Since only Person_1 is currently in chat, the list of Online Members is Empty and is not shown.
- When we create another Terminal window and run the client program, we can see that in this Client's Terminal (Client-2), it shows that Person_1 is present! message.

Client-2:

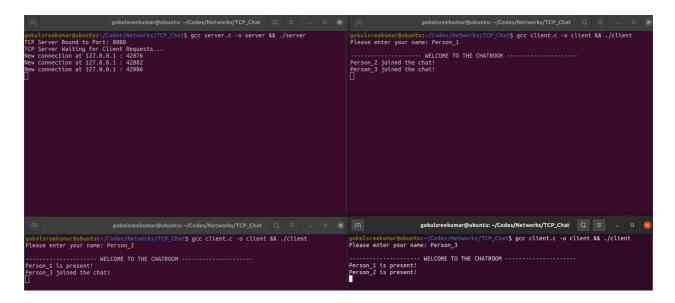
• In Client-1 we can see that, Person_2 joined the chat! message is shown and in the Server we can see another new connection message for the Client-2 as shown below:

Client-1:

Server:

```
gokulsreekumar@ubuntu:~/Codes/Networks/TCP_Chat$ gcc server.c -o server && ./server
TCP Server Bound to Port: 8080
TCP Server Waiting for Client Requests...
New connection at 127.0.0.1 : 42876
New connection at 127.0.0.1 : 42882
```

 When we add another Client to the Chatroom, the list of people present in the chatroom already is displayed (Person_1 and Person_2) and the following outputs are obtained in each other clients and the server:



When Chats start, from the perspective of Person_1, we can see:

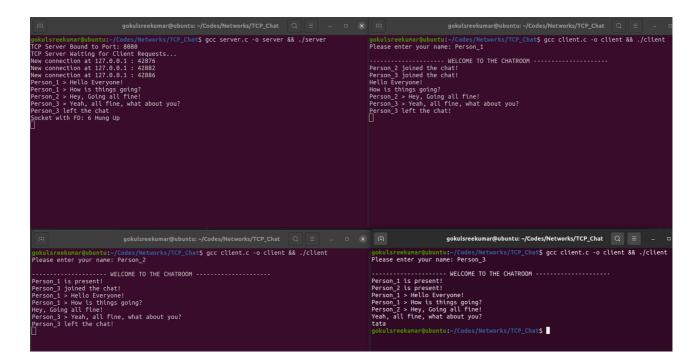
• From perspective of Person 2 we can see:

• And from **perspective of Person_3** as:

All messages are logged and visible in the server, in the format: client_name > message, shown as below:

```
gokulsreekumar@ubuntu:~/Codes/Networks/TCP_Chat$ gcc server.c -o server && ./server
TCP Server Bound to Port: 8080
TCP Server Waiting for Client Requests...
New connection at 127.0.0.1 : 42876
New connection at 127.0.0.1 : 42882
New connection at 127.0.0.1 : 42886
Person_1 > Hello Everyone!
Person_1 > How is things going?
Person_2 > Hey, Going all fine!
Person_3 > Yeah, all fine, what about you?
```

- We can use the word "tata" for leaving the Chat Room.
- If Person_3 leaves the Chatroom a message saying, Person_3 left the chat!, is send to ALL other connected clients in the chat (here, Person_1 and Person_2) and also in the Server side we can see Person_3 left the chat! Message as well as it's corresponding Socket which hot hung up:



 If another client now joins, then Person_1 and Person_2 (not Person_3) is listed as being present:

 We can use "tata" to exit from All the clients, and then Close the Running Server by using Control-C Keys, essentially closing the Chat Room.

```
gokulsreekumar@ubuntu:~/Codes/Networks/TCP_Chat$ gcc server.c -o server && ./server
TCP Server Bound to Port: 8080
TCP Server Waiting for Client Requests...
New connection at 127.0.0.1 : 42876
New connection at 127.0.0.1 : 42882
New connection at 127.0.0.1 : 42886
Person_1 > Hello Everyone!
Person_1 > Hello Everyone!
Person_2 > Hey, Going all fine!
Person_3 > Yeah, all fine, what about you?
Person_3 > Yeah, all fine, what about you?
Person_3 left the chat
Socket with FD: 6 Hung Up
New connection at 127.0.0.1 : 42900
Person_1 left the chat
Socket with FD: 4 Hung Up
Person_2 left the chat
Socket with FD: 5 Hung Up
Person_4 left the chat
Socket with FD: 6 Hung Up
AC
gokulsreekumar@ubuntu:~/Codes/Networks/TCP_Chat$
```