

Lab Sheet 5 - Chat Server Application

Team Members:

1. Aditya Dhananjay Singh - 22CS02001
 2. Kumar Snehal - 22CS02009
 3. Atharva Atul Penkar - 22CS02011
 4. Meysakthivelan T - 22CS02014
-

Github Repository Link (Contains ReadMe):

https://github.com/asingsh772004/Computer_Networks_Chat_Application

Screenshots:

Server (ADS):

```
Socket was successfully created.
Server binded successfully.
Server is listening
-----
Server-Client Connection Established
Assigned Socket 4 : Snehal
Snehal: I am not in Chat Room
Server-Client Connection Established
Assigned Socket 5 : atharva
Server-Client Connection Established
Assigned Socket 6 : sakthi
Snehal: CONNECT
Snehal has joined the ChatRoom
atharva: CONNECT
Snehal currently in Chat Room
atharva has joined the ChatRoom
Snehal: Hello to all
    Sending: [Snehal, to ALL] Hello to all
sakthi: CONNECT
Snehal atharva currently in Chat Room
sakthi has joined the ChatRoom
Snehal: @atharva hi penkar
    Sending: [Snehal] hi penkar
atharva: @sakthi hello
    Sending: [atharva] hello
sakthi: @ADS are you there ?
    Sending: [sakthi] are you there ?
sakthi: DISCONNECT
    Sending: sakthi has left the ChatRoom
sakthi: EXIT
sakthi: is EXITING
Snehal: @sakthi how are you
    Sending: [Snehal] how are you
atharva: EXIT
    Sending: atharva has left the ChatRoom
atharva: is EXITING
```

Client 1 (Snehal)

```
Enter Alias:
You: Snehal
Alias Assigned
You: I am not in Chat Room
You are not in Chat Room, Type CONNECT
You: CONNECT
Snehal has joined the ChatRoom
atharva has joined the ChatRoom
You: Hello to all
sakthi has joined the ChatRoom
You: @atharva hi penkar
sakthi has left the ChatRoom
You: @sakthi how are you
sakthi were not found in the Chat Room.
atharva has left the ChatRoom
EXIT
```

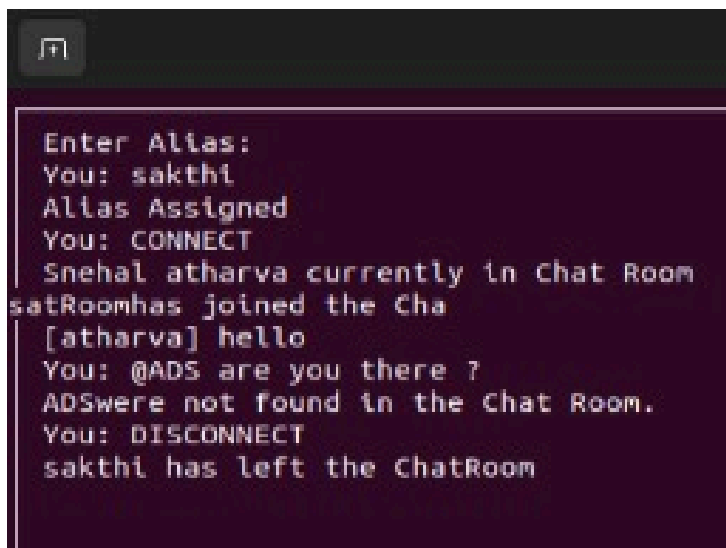
- Snehal's message "I am not in Chat Room" told him how to CONNECT
- Snehal's "Hello to all" was not visible to Sakthi (because he has not joined Chat Room by then)
- Snehal's private msg to Atharva won't be visible to Sakthi although he is in the Chat Room
- Pressing EXIT means directly getting DISCONNECTED from the Chat Room and closing the socket, thread, etc.
- After Sakthi left the Chat Room, Snehal's message to him was not reachable as there is no user named Sakthi.

Client 2 (Atharva)

```
Enter Alias:
You: atharva
Alias Assigned
You: CONNECT
Snehal currently in Chat Room
atharva has joined the ChatRoom
[Snehal, to ALL] Hello to all
sakthi has joined the ChatRoom
[Snehal] hi penkar
You: @sakthi hello
sakthi has left the ChatRoom
```

- Atharva's private msg to Sakthi is not visible to Snehal
- Snehal's broadcast message is visible as "to ALL" to Atharva

Client 3 (Sakthi)



```
Enter Alias:
You: sakthi
Alias Assigned
You: CONNECT
Snehal atharva currently in Chat Room
sakthi has joined the ChatRoom
[atharva] hello
You: @ADS are you there ?
ADS were not found in the Chat Room.
You: DISCONNECT
sakthi has left the ChatRoom
```

- Sakthi tried texting a Client named "ADS" who was not in the Chat Room so it shows he was not found.
- Disconnect means leaving the Chat Room but staying connected

Additional Functionalities:

Server Multithreading code:

1. Allows multiple users at once (upto 5 clients)
2. Unique Alias assigned to each user on joining: (If username already taken then prompts them to share it again).
3. **CONNECT:** Enter a ChatRoom
4. '@': symbol to privately message particular set of users (@<user1> @<user2> "Message") to send to multiple users
5. **Message:** Directly broadcast a message (send to all users in a ChatRoom)
6. **DISCONNECT:** Allows you to leave the ChatRoom (will not be able to send/receive messages, but be linked to server so that you can CONNECT to the chat room anytime.
7. **EXIT:** Break out of the ChatRoom if you are in it, close the thread, join it and this client disconnects from Server.
8. Server Log Maintenance: Server maintains the log for all actions such as: (these functionalities help in debugging as well as to know if there is loss of packets at any point of time).
 - a. Connection Establishment
 - b. Alias assignment
 - c. Message received from the Sender client
 - d. Message to be sent to the receiver clients
 - e. Disconnection of any client
 - f. Exit made by any client

9. Color codes for specific type of server message:

- a. **RED**: All socket/read/write errors
- b. **GREEN**: Successful socket bindings, any successful connection establishment messages
- c. **YELLOW**: Messages received by server from clients which are not yet in Chat room
- d. **WHITE (DEFAULT)**: Messages that are received from Clients who are in the Chat Room.
- e. **CYAN**: Final messages to be sent to the clients on the receiving end from the server (after parsing and editing out the keywords mentioned above).

NCurses and terminal.h (custom terminal.h for better functionality):

1. **Ncurses** is a library we used to alter the terminal. In the regular terminal, when multiple threads are writing then the text gets all mixed up. For example, if user 1 writes "Hi" and user 2 writes "Hello" then they might see "HeHillo" or similar instances.
2. Ncurses offers a solution by simply separating the Input dialogue box (where you can write the messages) and the Chat window (where you can see the other Client's/Server's messages). This works the same way as in WhatsApp/other chat applications.
3. **Terminal.h** is the header file we have used to include this library and help us to create the terminal we need.