

Home Assignment-1: Multi-threaded Socket Programming

Group Chat

Description: In this assignment, you have to implement a group chatting system where multiple clients can join the server and can chat together like in a chat-group (think of facebook messenger group).

When we run the server, server will wait until a client request comes. When we run a client program, this client request will go to the server and the server will create a new thread for this client. So the server can handle multiple clients by using different threads for each client (as we have seen or will see in the class).

So when a new client joins the server, first thing the server will do is to create a separate thread for him/her. Then the server will ask his/her name by sending a message like “Enter your name:”. After getting this message from the server, we shall take input of client’s name from the client side program’s console. The new client will send this name to the server; server will store this information. Let’s assume, the new client’s name is Ironman. Server will also send a server-message to all of the existing clients that “**Ironman has joined the group chat.**”

Let’s assume that there are two clients at this moment. Their names are: Ironman and Thor. Now we run another client program and give “Hulk” in the console as the input of client’s name. Now other existing client (in this example: Ironman and Thor) will see in their console that “**Hulk has joined the group chat.**” Now suppose, Ironman want to send a message in the group, he wants to say “**Hello**”. So he will write the message in his console and press enter. And it will go to server and the server will send this message to all of the existing clients. So all of the existing clients will see in their windows (consoles) that- “**Ironman: Hello.**”

Now suppose, Ironman wants to leave the group-chat, he will have to simply stop his program by closing it. Then the server will have to notify other existing clients by sending them a server-message that “**Ironman has left the group chat.**”

Bonus: The server side program will keep a count of the existing clients all the time. Every time a new client has joins, the server side console will print “A new client has joined” and print the latest count of existing clients. Similarly, every time a client leaves, the server side console will print “One client left!” and also print the latest count of the existing clients. (Remember that this whole bonus part will only print on the server side console, NOT CLIENT SIDE).

Conditions:

- a. There are many functions for sending and reading data, you can explore them on your own. **But you must use only writeBytes() and readLine() function on both server and client sides for this assignment.**
- b. Any kind of gui/jFrame/jPane etc. is completely prohibited for this assignment.

Hints:

1. You need a ClientHandler thread on the Server Side program for handling multiple clients.
2. Keep a list of existing clients' names and output-streams on the server side to send any message to all of them at once.
3. You need to implement simultaneous input/output for any client for this assignment. So you need to use two separate threads on the client side: one for input-stream and one for output-stream.