CS 407/CS 508 Distributed Computing February 23, 2022

Homework 3

Your goal in this assignment is to implement the basic functionality of a 1-room chat server.

Supplied here is a simple client, with which your server must be compatible. Here we outline the chat protocol that your server must implement. For additional detail or clarification, you may consult the provided client source code.

1. A client initiates a session with the chat server by opening a TCP connection to the server. The session remains active as long as the TCP connection is alive. A client with such an active session is said to be "connected" to the server.
2. Upon connecting to the server, the client immediately sends a single line of text over the TCP connection containing the username that the client would like to use in the chat room.
3. Each subsequent line sent by the client represents the content of a message the client wants to send to the chat.
4. Upon accepting a new connection, the server waits for the username.
5. Upon receiving a message from any connected client, the server transmits the message to all connected clients.
6. To transmit a message to the client, the server writes a single line of text to that client's TCP connection. The line is expected by the client to consist of the author's username, a colon, and then the message content.

This is not intended to be a production-quality chat system. In particular, there is only 1 chat room, and no record of chat history, so you cannot receive messages that were sent while you were disconnected. Also, the wire format does not support multiline messages. Because of this the task has just enough complexity to give you a taste of writing concurrent code that talks over a network, without wasting your time on all the extra features that a real-world chat system would incorporate. However, if you follow the protocol and implement the behavior described above then you will have a functional program that you really can use to talk to your friends over the internet if you want to.

The problem you are solving in this assignment provides an excellent demonstration of why programming distributed systems generally necessitates the use of concurrent code. In this case, the chat is an inherently concurrent task, since any client may independently arrive or leave or send a message at any time. There may be several approaches you can take to implement your server, but you will find it easiest to allocated a dedicated thread for each client that waits to receive messages from only that client, and then behaves accordingly when it receives one.