

WAC CHAT Server

CSI 470 - Computer Networks

Much of the text here will feel familiar. I encourage you to read it carefully, as things have changed slightly to highlight server components. In this assignment you'll be writing a chat server application to allow clients to connect to remotely. The application layer protocol for this chat server is described below. For this assignment, use python to write a server application that automatically allows client connections with the following protocol description. This document first lists the application layer protocol that describes how the client and server communicate. Then it discusses other nuances of the server. Optional components are also discussed.

1 Application Layer Protocol

This describes how the client and server talk to one another at the application layer. Often, the actual client application “hides” much of this and that will be our goal. While there are three commands listed here that the client may send to the server, two of those application layer protocols do not need to be known by the client application in order to be used. However, they must all be processed by the server. For “LIN”, the server associates the name with the socket. For MSG, the server deletes the MSG and prepends the message with the name associated with the sending socket. Then the server sends the message to every other client that is connected. For SEC, the server just responds with their own secret. You can test your servers using telnet or with the client you wrote in the previous homework.

1.1 Messages from client to server

Messages that may be sent to server include:

- **LIN “name”**

This is the login message. After this message runs, the user will forever be known by the “name” given after the LIN. In code, the server should associate this name with the particular socket that issued the LIN. This should happen automatically by the client application, but your server should be prepared for clients to operate inappropriately. Technically though, in a different client application (or a rogue one), it would be possible for a user to issue a LIN at any moment by sending this message or to never send one at all. Your server should be prepared for these cases and respond appropriately. Note that I'm being slightly ambiguous about what

appropriately is for some aspects of this. This is intentional. Make a good design decision. However, if a client is not “LIN” then the server should not forward along their messages as defined below.

- **MSG “msg”**

This is the send my message with my name to everyone on the server message. When this message runs, every user connected to the server will receive whatever appears in “msg”. In the chat client terminology, this tells the server to send “msg” to everyone else and not yourself. The server should also prepend the message with the user’s name. Some chat servers send an echo of your message back to yourself so you can see how it is ordered among the other messages, but your server should not be designed this way.

- **SEC**

This is a quirky little message in which the server sends a secret to the client. Your server can have whatever secret it might like but it must send a message in response to this message.

- *OPTIONAL*

- **W “name” “msg”**

This is an optional command. It would allow users to whisper to other users on the system. When a user sends a W command, the name listed next should be used to find the socket of the logged in user with that name. Then, the msg should be prepended with the user name of the sender and the word whisper. In this way, clients receive whispers directly from others and that whisper is seen by noone else. Jack might type “W Jane Hi, How are you” to the server. If a socket associated with the username Jane exists, then a message will be sent to Jane’s socket only that might appear as: “Jack whispers, “Hi, How are you?””

- *OPTIONAL*

- **WHO**

This is an optional command in which the server sends the list of all logged in users to the client.

1.2 Messages from server to client

Messages that may be sent to the client include:

- **“msg”**

Any information sent to the client is a message from the server. The client requires no state at the application layer and should be displaying any message it receives from the server.

2 Requirements for Full Marks

There are a few requirements for full marks on this assignment. Be sure you include the following:

- Server allows connections from clients and associates a client socket to their name after a LIN
- Server doesn't allow a client to MSG unless they have LIN. Make
- Server appropriately sends MSG messages with a prepended name.
- Server does not send MSG message back to the sender of the MSG itself.
- Server is bound to the appropriate port.