# Client Functions:

connect\_to: socket, get\_hostname, then connect. Return the socket id

process\_command: sends and receives a string from the server. Use the received string to generate the reply.

process\_chatmode: connects to the chatroom. Makes a new thread for the client to receive messages. Checks for received messages and sends some until the room is closed.

# Server Functions:

Forks don’t work.

There is a chatroom structure that holds the following data:

* Process ID
* #Users
* Port
* Chatroom name
* Socket ID
* Vector of client Socket IDs

Main Command Server: Socket, bind, and listen. Accept new sockets continually and create a thread to handle their request. The worker thread will receive the message, process it, and send it back once finished.

DELETE: Looks through the existing chatrooms list and tries to find the one with the name it’s looking for. Once found, shoots a warning message to all clients connected before closing.

CREATE: Super hard. Opens up a new chatroom and records it on the active rooms list. Uses threads to create the new chatroom. Uses a similar setup to the main server connection. Socket, bind listen. Afterwards, creates a new thread to handle each client connection.

LIST: Looks at the list of active chatroom and returns a string of all the room names in one sentence.

JOIN: First checks if the room exists, and returns the host and port number for the client to connect to. From there, the chatroom will see if there are any connection being made to it. Once done, they will try to send and receive all the messages they get from the clients in the chatroom.