Design Document: Simple Chat

Aaron Nguyen
CruzID: anguy200

1 Goals

The goal of this assignment was to create a simple chat system using c/c++. Using the shell program from the last assignment I built my chat system off of this system.

2 Design

To achieve code from Lab 2 was used primarily the TCP socket code. Like creating the socket and filing it with the destination info, and establishing the connection between the source and destination. The client portion of the assignment was simple using functions like readline, send, and recv. The client was much more complicated due to the chat system needing multiple states (WAIT, INFO, CHAT). In WAIT mode I would have two threads running one that receives all inputs from stdin and one that would wait for messages coming from the server. Messages like connect requests or server connection drops. To quit wait mode /quit has to be inputted and a message will be sent to the server to notify it to remove the client from possible connections. Chat Mode was done in a similar fashion, but instead listened to only another client like P2P. For INFO mode that was primarily handled by the main function. All the main functions did was read lines from input and send the captured lines to the server, and wait for a response. The server was much easier all it did was receive commands from the clients and compile the appropriate information into a response. To allow for multiple client connections to the server I used pthreads so whenever a client connected to the server a new thread was spawned and whenever a connection was closed the thread would in response shutdown appropriately.

Protocol and Handshake

The protocol used for communication between client and server was TCP. To indicate the end of the message was the use of “\r\n\r\n” being added at the end of each message sent between server and client. Both the client and server utilize a recv while loop so if recv returned a 0 that indicates that the connection between the server and client has been closed. In terms of processing the server is handling a majority of the load due to it having to execute the command in the machine's local terminal.