

Homework 1

A chat room service

CSCE 438

Curtis Green - 422008537

Caleb Edens - 822007959

Overview:

Design and create a chat room service using sockets with TCP connections that allows the user to connect to any host and port that the server is on and create, join, delete, or list details about the rooms. Once inside the chat room the client should be able to send and receive messages to and from any other users also within that chat room. The only issue that our code has is that we can't get the program to gracefully handle closing with CTRL+C. If you close the client or server with that command, it just runs the last seen command in an infinite loop. We tried signal handling, thread closing, error handling, and buffer cleaning but none of it worked.

Server:

The server creates a thread that starts up a socket connection on the port passed in through the command line. For each additional client accessing the server it will create another thread to handle the connection and requests. Within the request handling function the server receives the commands from the user, processes the request, and sends back the appropriate data. After sending the data the connection is closed and re instantiated in the main file until the client disconnects. If the user requests to create a chat room it will create a new thread on a new port that simply opens a socket connection and relays messages between the users. If the user requests to join an existing server, a port number will sent to the client that it can use for the connection. A new thread is made for every user so that send and receive can be done by multiple people at the same time. These layers of threading is designed to make it to where a client will never have to wait on another clients join request or send/receive.

Client:

The client sets up a TCP socket connection with the server based on the command line host and port arguments then displays a menu of options the user can request from the server. After sending the command and then receiving the response the client displays the information relevant to the user's commands. If the user decides to join a chat server it will receive a port number from the server which it will connect to then enter into chat mode. Within chat mode the client creates two threads, one to handle receiving messages and another to handle sending them. This way the user can send and receive without needing to do each sequentially.