

Nunzio D'Amelio & Sam Francese  
CSC 315: OS  
Project 4

### **Project Details:**

A command line-based chatting program using socket library, that supports chat rooms, username handling, file transfers, and more.

### **Project Requirements: Successfully completed checkpoint 1 and 2, issues and half implementation at checkpoint 3 and 4**

1a. Main\_server can display the up-to-date list of connected clients. When a new client connects or any client disconnects, the up-to-date list will be printed in the server-side terminal.

1b. This requirement was met in its entirety

2a. Main\_server accepts multiple main\_client connections and broadcast messages from one client to all connected clients properly

2b. This requirement was met in its entirety

3a. The main\_client can specify a username when connecting to the server. The server will maintain the user names and broadcast the names to clients properly.

3b. This requirement was met in its entirety.

4a. Use a random, unique color for each client in a room.

4b. This requirement was not fully met. Although colors are used, they are only different in a window. That is, all names in one client window will be white, for example, and the other client window will show all the users in blue.

5a. Main\_server allows multiple chatting rooms running simultaneously...

5b. This requirement was met, but now has errors after moving on to part 6. The current feature allows the user to join a room by putting new into the command line as desired. The program blocks missed input. If a user tries to put in a specific room the program will then end after the user enters the user name. The server detects this but we couldn't figure out the error in the code. We set a max of three rooms.

6a. In case the client neither specify the room number nor used new keyword, it will retrieve the list of the rooms available currently and choose from the list

6b. We implemented the new input of room on the client with success however like mentioned below there are errors after selecting a specific room. There is no error when the user uses new, they are properly put into a new chat room. If no room is mentioned the program successfully shows how many people are in each room.

7a. Allow file transfer. A client can send a local file to a remote client with a special command SEND

7b. Not fully implemented users can select to send a file using the SEND command, the program will read an error if nothing is sent or tell if it is a success. The receiver of the file can decide if they want to accept or deny. From what we can tell the file is properly read and transferred.