Kyla Ramos & Robert Helck
CSC 345-01

Project 4 Report

- Requirements Implemented:
  - Connection list
  - Broadcast
  - User name
  - Multiple chat rooms
  - Room list display from client
  - Color
  - Notifications for entering and departing clients
  - New keyword
  - Room numbers
  - User neither entering a number nor "new"

- Checkpoints Reached:
  - 1 Checkpoint complete
    -The server and client connect, and the server prints an up-to-date list of connected clients to the server terminal.
    -The server connects to multiple clients (expanded by our chatroom feature) and these clients are kept track of in a linked list.
    -The server is able to broadcast the name of the client when they send messages to one another.
    -All connected clients are notified upon a new client joining or leaving the server. This is expanded upon in the chatroom feature.
    -Each client is assigned a unique color, and their texts show up.
  - 2 Checkpoint complete
    -Main server allows multiple chat rooms. These can be specified explicitly with a number, the keyword "new", or by leaving the room blank.
    -The max number of rooms is 5, with 7 people in each.
  - 3 Checkpoint complete
    -Main server can provide info about what rooms are available, and the client can join which room they would like.

We did not complete the extra credit. I commented out the small amount of code I wrote to that end to avoid the user crashing the program by entering the SEND command. I believe that the correct approach would be to read the contents of a binary file using the fopen() function, send this to the server, have the server write this to a location using fopen(), and then for the server to send the file this same way to the client.

Known bugs:

There are some bugs which we were unable to solve due to time constraints.

-Adding names out of order: if a user opens two (or more) terminals and begins to create a user, and then creates another user before finishing, this will interfere with the chat capability. Typically this bug is automatically resolved when the users try to send messages back and forth. A solution probably involves managing file descriptors better and relying less on send() and recv() functions “lining up” in the code.

-Joining a room using new when none are available. The program will prevent the user from joining a room if all are filled up, but the close() function doesn't work as we intended, and the user will be able to continue chatting, and crash the server. Solving this probably involves better use of close().

-The enter key does not always remove the client from the server. Despite a lot of debugging, this issue persists in the code. We do not have a solution for this problem.