CSC 345-02

Project #4 Network Programming

Chris Jenson and Brian Worts

05/3/21

Sometime when the initial run of the main_client executable, the program will immediately exit. We were unable to isolate or consistently replicate this bug because of this seemingly random bug. If this occurs, disconnect all users and then restart the server. Otherwise, it operates as expected.

Program requirements(70 pts):

Implemented
•(10 points) 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.It should work exactly in the following fashion:

The server prints out the list of connected clients by nickname. If there was no nickname, it shows their IP. When a client disconnects, the list reprints without them.

Implemented
•(10points) main_server accepts multiple main_client connections and broadcast messages from one client to all connected clients properly.

The server can accept up to as many clients are specified in the listen function. Ours is currently set to six. The client can then broadcast to all other clients in the room.

Implemented
•(10points) main_client can specify username when connecting to the server. The server will maintain the user names and broadcast the names to clients properly.

The program asks for a nickname (username) upon initial connection. If the user enters nothing, their IP is used as their nickname. The server maintains all clients nicknames and broadcasts the name with their messages.

Implemented
•(15points) main_server allows multiple chatting rooms running simultaneously. For this purpose, the client behavior will be slightly changed. Clients can request whether it will open a new chatting room or join an existing room. If a client wants to open a new chat room, then it should connect to the server as

[in client terminal]
$ ./main_client IP-address-of-server new

If the connection was successful, then the client will be assigned a room number from the server and display it on the screen, e.g.

[in client terminal]
$ ./main_client IP-address-of-server new
Connected to IP-address-of-server with new room number XXX

If the client wants to connect to an existing room, then it can use a command as

[in client terminal]

$ ./main_client IP-address-of-server XXX

where XXX is the pre-existing room number. You can assume that all rooms are numbered.The main_server is responsible for maintaining the rooms and clients therein properly.If a main_client requests for non-existing room number, it should properly block the request.You can optionally set the maximum number of rooms to be managed by the server, but it should be at least 3.

Our program supports up to 5 rooms running simultaneously. By entering “new” the server will assign them to an empty room and display their connection with room number. The client can connect to an existing room by specifying the room number. The checks for invalid numbers are missing.

Mostly Implemented

•(15points) Incase the client either specify the room number nor used new keyword, it will retrieve the list of the rooms available currently and choose from the list as below:

[in client terminal]

$ ./main_client IP-address-of-server

Server says following options are available:

Room 1: 2 people

Room 2: 1 person

…

Room 10: 5 people

Choose the room number or type [new]to create a new room:_

If no room is available,it should automatically create and join to the new room (thus implicitly using the new command above).

If the client does not specify the room number or the new keyword. The list of current occupied rooms is printed to the client. From here the user must reconnect specifying the room they wish to join.

Implemented

•(10 points) Use a random, unique color for each client in a room. It is okay if clients are using different sets of colors for the same room information. However, within the same client, individual users must be assigned unique colors.

System messages are in white and all users get a unique color based on connection order.