Hw06: Chat Room

Focus

- Concurrency

Spec

Write a chat room server and provide a corresponding client.

- The chat server's primary job is to pass messages from each client to all other current clients.
- The server will need to
  - allow connections from clients.
  - receive messages from clients
  - send the messages to all current clients.
  - Inform current clients when a client joins or leaves.
- Note that the server is not a client!

Notes

- Please make sure everything about your server and clients can be set at start up from the command line. There should not be any additional interaction after starting the program to, for example, set up the network connections.
- Each time a client connects to the server a new thread should be created in the server to handle the input from that client.
- Each time a message is sent by a client, the message should be added to a queue.
- A single separate thread should be responsible for removing the messages from the queue and sending them to the all of the clients. In terms of producers and consumers, there are as many producers as there are connected clients, and there is a single consumer.
- Don't poll! If you can't have what you want right now, block. Someone else should "wake" you when what you want is available.
- What happens when a thread is finished? Remember what happens when a process terminates? We have similar issues with threads...
- The server should log (i.e. display to standard output) the connect / disconnect activities of the clients and report them to the clients for display to the users.
- When a client logs in, he should be informed who else is currently logged in.

- Current users should be informed when a user joins or leaves.
- Note the main areas to watch for race conditions:
  - Maintaining the collection of clients currently logged in.
  - Maintaining the queue of messages to be sent out.
- In order for the server to know that the client is "signing off", the client should send a message to the server saying so.

Turn in:

Your solution should include:

- A makefile
- Necessary source files: Your server and a corresponding client. The latter may well be very similar to the client code that you used for the previous assignment.
- README.
  - Required! We need to know how to run your server and clients. We do *not* want to have to read your code to determine that.
  - Explain anything special about your program. Please note whether the client is the same as in the previous assignment, if not what changed. And, as always, explain if there is anything that didn't work.

t, polling, etc. For this project use the select (or pselect) system call.

Turn in:

As usual, in addition to your source files, provide a makefile and a Readme.