CS 4850 Computer Networks I

Spring 2019

Project 3: ChatRoom Version1

Due Date: Wednesday May 1st, 11:59pm on Canvas.

1. Overview

In this project, you will implement a simple chat room that includes a client and an echo server. You are permitted to use your language of choice as long as it **utilizes the socket API**.

The client program provides commands: **login** (allow users to join the chat room), **newuser** (create a new user account), **send** (send a message to other clients; actually send the message to the server and the server forwards the message to other clients), and **logout** (quit the chat room).

The server runs a chat room service and echoes messages back to the client.

2. Description

You will implement a server and a client. The server will use 1 plus the last four digits of your student ID as the server port number to avoid conflicting with other students' server program. For example, if the last four digits of your student ID is 3456, then as the server port number is 13456.

In this project, only one active client connects to the server. The following commands need to be called at the client side and the server side implements the corresponding functions required to support these commands. When the server starts, it should first read the user account information from a file. For grading purpose, the initial user accounts (UserID, Password) are (Tom, Tom11), (David, David22) and (Beth, Beth33).

1. **login** UserID Password

If the server can verify the UserID and the Password, the server will send a confirmation message to the client; otherwise, the server will decline login and send an error message to the client.

2. **newuser** UserID Password

Create a new user account. The length of the UserID should be less than 32, and the length of the Password should be between 4 and 8 characters. The server will reject the request if the UserID is already there and send an appropriate error message. When a new user is successfully created the program should then prompt the user to login. The users' IDs and passwords should be kept in a text file.

3. **send** message

Send the "message" to the server. The server will precede the message with the UserID and send it back.

4. **logout**

Logout from the chat room. Once logout, the connection between the server and client will be closed. The server should allow the next client to connect.

3. Programming Language

You can use any programming language you like (C, C++, Java, Python, Ruby,…etc). If you

4. Grading (Total 100 Points)

Demonstrate your application to the TAs during their office hours in the week specified.

- 15 points for each of the four commands. You will lose points if the commands are not implemented as shown (60 points total)
- 40 points for the neat source code and implementing appropriate error messages. Your source code must be well commented, including an overall header with student name, date, program description, etc.
- You will lose 80 points for any bug that causes the program to crash or makes the program exit abnormally even if all commands can be demonstrated.
- You will lose 100 points is you do not utilize the socket API

5. Code submission and test

You must submit your source code files through the course Canvas site. Late or email submissions, or submission of executables will not be accepted. After the submission deadline, you will have to meet with the course TA to demonstrate the functionality of your program(s) by downloading your source code from Canvas and compiling and running it in on your own computer in front of the TA.

6. Sample Outputs

Sample client output.

My chat room client. Version One.
>**send**
>*Server*: Denied. Please login first.
>**login** Tom Tom11
>*Server*: Tom joins
>**send** Hello.

>Tom: Hello.
>**send** Bye.
>Tom: Bye.
> **logout**
>*Server*: Tom left.

Sample server output.

```
My chat room server. Version One.
Tom login.
Tom: Hello.
Tom: Bye.
Tom logout.
```