Intro. to Network Programming 2020 Fall

Homework 1 - Bulletin Board System: Part 1

Description

In this project, you are asked to design Bulletin Board System (BBS) server and client. Your program should be able to handle multiple connections and receive user command from **standard input**. After receiving command, the server send the corresponding message back.

Requirement

The service accepts the following commands and **at least 10 clients**:

When client enter command incompletely E.g., missing parameters, the server should show **command format** for client.

Command format	Description	Result	
register <username> <email> <password>	Register with username, email and password. <username> must **be unique.** <email> and <password> have no limitation. If username is already used, show failed message, otherwise it is success. **Note:** You have to send this command and get associated message by **UDP**.	Success	Register successfully.
		Fail	Username is already used.
login <username> <password>	Login with username and password. Fail (1): User already login. Fail (2): Username or password is incorrect. If login successfully, server should send a **randomly generated number** as the identification for the subsequent udp command. **Note:** You have to send this command and get associated message by **TCP**.	Success	Welcome, <username>.
		Fail (1)	Please logout first.
		Fail (2)	Login failed.
logout	Logout account. If you haven't logged in yet, show failed message, otherwise logout successfully. **Note:** You have to send this command and get associated message by **TCP**.	Success	Bye, <username>.
		Fail	Please login first.
whoami	Show your username. Send **whoami <received random number>** to server. If you haven't logged in yet, show failed message. Otherwise, show username. **Note:** You have to send this command and get associated message by **UDP**.	Success	<username>
		Fail	Please login first.

list-user	List all users in BBS. **Note:** You have to send this command and get associated message by **TCP**.	Name Email <Name1> <Email1>
exit	Close connection. **Note:** You have to send this command by **TCP**.	

General

Please make sure you develop your program on **Linux**. For development environment, you can just apply **NCTU CSCC account** to use Linux workstation. If you don’t want to apply anything, you can use VM for develop.

Use “% “ as the command line prompt. Notice that there is **only one space** after the prompt.

The server close connection if client use **exit** command, but server still running and client can connect again.

For manage user information, storing data in your server is necessary. Therefore, you must have some methods to handle this, like manage a simple **database** e.g. SQLite and then design tables by yourself or only use **data structure** to store data.

To run your server, you must to provide **port number** for your program. E.g., bash$ ./server 7890

Now, you can use your client program to connect to your server, when client connect to server, the server print message “**New connection.**”

Assume your server is running on **localhost** and listening at port 7890. E.g., bash$ client 127.0.0.1 7890

Scenario

```
bash$ client 127.0.0.1 7890
********************************
** Welcome to the BBS server. **
********************************
% register
Usage: register <username> <email> <password>
% register Bob bob@qwer.asdf 123456
Register successfully.
% register Bob asdf@asdf.asdf 123456
Username is already used.
% login
Usage: login <username> <password>
% login Bob
Usage: login <username> <password>
% login Bob 654321
Login failed.
% login Tom 654321
Login failed.
% login Bob 123456
Welcome, Bob.
% login Bob 123456
Please logout first.
% whoami
Bob
% logout
Bye, Bob.
% logout
Please login first.
% whoami
Please login first.
% list-user
Name     Email
Bob      bob@qwer.asdf
% exit
```

Grade (100%)

- Socket connection and print welcome message to client. - (30%)
- **register** command. - (20%)
- **login** / **logout** command. - (20%)
- **whoami** command. - (10%)
- **list-user** command. - (10%)
- **exit** command. - (10%)

Submission

Please upload a zip file called “hw1_{$student_id}.zip” that includes your source code (server and client). Submission that don’t follow the rule will **get 20% punishment** on the grade.

You will get **0 points** on this project for **plagiarism**. Please don’t copy-paste any code!

Reference

1. C/C++ Socket
2. SQLite C/C++ Interface
3. Linux socket SELECT
4. Database table example

```
CREATE TABLE USERS(
    UID INTEGER PRIMARY KEY AUTOINCREMENT,
    Username TEXT NOT NULL UNIQUE,
    Email TEXT NOT NULL,
    Password TEXT NOT NULL
);
```