Due date

24th January 2021 January, 23.55

Goal

In this assignment, you will implement server side and client side socket programming.

Implementation Details & Requirements

Nowadays, instant messaging applications are used very often by end users. WhatsApp, the most popular of these applications, recently announced that it will make some changes in the security policy it will follow in our country. With this development, such applications immediately have been discussed by people and the search for a more secure application has started.

With this assignment, you will start to develop an instant messaging application. You will implement the first phase for this application. In the first phase, people can send non-secure messages to others using the messaging server. They can be sent and received using JSON format that contains `from`, `to` and `message` key-fields. In this system, we must have a messaging server and we may have more than a client.

The server must be running up always for the system to work. It executes commands which are taken by clients and controls and manages the system. The server will be hosted on **`localhost`** and its port is **`3205`**. When the client connects successfully to the server, it will enter its private *`phone_number (username)`* initially. Then it is defined in the server system.

The commands are described in below can be used in the system:

- **-gcreate** *`phone_number+group_name`*: Creates a new specified group. The groups have been protected with non-encrypted passwords. The system will ask to define a password.
- **-join** *`username/group_name`*: Enter to the specified username or group name. If the group is private, the client must know the password for entering.
- **-exit** *`group_name`*: Quit from the group that you are in.
- **-send** *`message_body`*: Send a JSON-formatted message to the group that you are in.
- **-whoami** Shows your own *`username (phone_number)`* information.
- **-exit** Exit the program.

Submission

Submission will be via Github and SAKAI.

- GitHub Classroom Invitation link:
  https://classroom.github.com/a/xh-KXdZo
- Name your code files for server and client such as:
  - *StudentNumber_server.c*
  - *StudentNumber_client.c*

  (do not use another naming standard.)
- You will lose credit for not naming your submission properly. (15 pts)
- Late submission is not accepted.
- For this assignment you will work individually.
- The POSIX library (pthread) will be used.
- I will compile your code with `gcc`.

Academic Dishonesty

Your submissions will be scanned among each other as well as the Internet repository. Any assignments that are over the similarity threshold of a system for Detecting Software Similarity will get zero. We strongly encourage you not to submit your assignment rather than a dishonest submission.

Grading policy

- Documenting code and coding style
  (proper indentation, describe important/critical functions) (20 pts)
- Implementation of server side programming (50 pts)
- Implementation of client side programming (30 pts)

For Questions

For any questions about the assignment, please use the forum in the SAKAI system. Before asking your question, please check carefully previous questions and answers, where similar questions were already asked by someone else already answered.

- **No private questions via email will be answered!!!**
- Use forum public messages to ask questions, where someone else may benefit from and learn something from your question and its answers.
- We will try to answer any of your questions as soon as possible.

Good luck!!!

Please READ all of the instructions carefully before you ask help!