

Assignment on Socket Programming

In this assignment, you will design a very simple chatbox application. The server must accept multiple clients. The client can send four types of messages to the server.

1. LMessage – This is a login message from a client to the server. It includes

- a. username
- b. password
- c. type (admin/normal)

You can keep track of a handful of <username, password, type> from a file (6 entries should suffice). If the username, password, type matches then the server will send the client a “login successful” message. The server should keep track of which clients are currently logged in. If a client is already logged in, the server can ignore this LMessage. Also, keep in mind that a user won’t be able to send the following messages without login.

2. SMessage – This is a message from a client to the server. It includes

- a. command
- b. text

The command can be either “show” or “logout”. When a client sends the command “show”, the server will return the list of all the clients (username) who are currently logged in. If a client sends the command “logout”, then that client will be logged out (client program will be terminated). The “text” will be displayed in the Server console.

3. BMessage – This is a broadcast message from one client to all other clients who are currently logged in. It includes

- a. text

The “text” will reach all other clients if and only if the sender client is of type admin. This checking must be performed at the server. That means the message must go from the sender to the server first and then the server will forward this message to others (who are logged in) if the sender is an admin. The server must also send the sender appropriate messages (successful/failure). Please note that the sender will not receive this “text”.

4. CMessage – This is a client message from one client to another client. It includes

- a. receiver
- b. text
- c. sendFile

The message should be sent to the server first and then the server should forward this message to the receiver (if the receiver is logged in). The “receiver” field should be the username of the “receiver”. The “sendFile” field is optional. If the sender wishes to send a file to a receiver, he must mention the name of the file in this field.

Design Guidelines

You should implement the above messages as different classes. That means you should have four different classes of message namely LMessage, SMessage, BMessage, and CMessage. Another thing is - to classify among different types of messages you can use a “message type” at the beginning of each message. For example, consider the following message:

```
#C#papon##hello papon. Please find the file##test.txt
```

The server can understand that this is a CMessage from #C#. Similarly, you can use #L#, #S# and #B# for other messages. The next part of the message is “papon” which is the username of the receiver. To distinguish among different attributes ## is being used as the delimiter. The “hello papon. Please find the file” is the text and “test.txt” is the file that needs to be sent.

You can use any other techniques suitable for you to classify the messages and identify different parts of the message. Keep in mind that all messages must go to the server first and then the server will identify what type of message this is and what needs to be done.

Use of GUI is not mandatory. However, you can get some bonus if you develop a GUI.

Marking will be heavily dependent on the design of the application.

Deadline: January 26, 2019 (02:00 AM)