

Assignment:

An Advanced Group Messaging Application [Deadline: **31st Oct, 2019**]

Resources:

- <http://tools.ietf.org/html/rfc959>
- <https://www.geeksforgeeks.org/socket-programming-cc/>

Description:

Implement an advanced group messaging application (for ex: Slack, Trello).

The list of features:

- The system consists of a server that can have multiple chat groups at a time.
- A cloud storage - stores shared media for every group chat
- Real-time messaging to all the members of the group
- File sharing features with other groups (can be done only if the user is common to both the groups)
- Each client user can log in to the server with valid credentials and can upload/ download/ delete any number of files he/she wishes to.

Implementation details:

The Server:

- The server will listen to two different ports: one for control messages and the other for data messages.
- The server consists of one root directory, inside which separate directories for separate “messaging groups” will be present. Each user of the group can upload/download/delete any number of files to/from the designated directory on the server. Users of other groups can only download the files (only if they have access to a particular file).
- The server has to maintain one log file for each group. After each successful user action, the server will log that action in the corresponding log file with the following details: filename, username (the user who took this action), action (upload/ download/ delete/ share), IP address (of the user machine), date and time.
- If a new user signs up, he gets to see a list of all groups and gets to join any group (note: all are public groups).

User Actions:

Users can perform different actions. A brief description of each of these is outlined below. For all the output actions (List files, Show log) present them in a proper format. Cleanliness will fetch bonus marks.

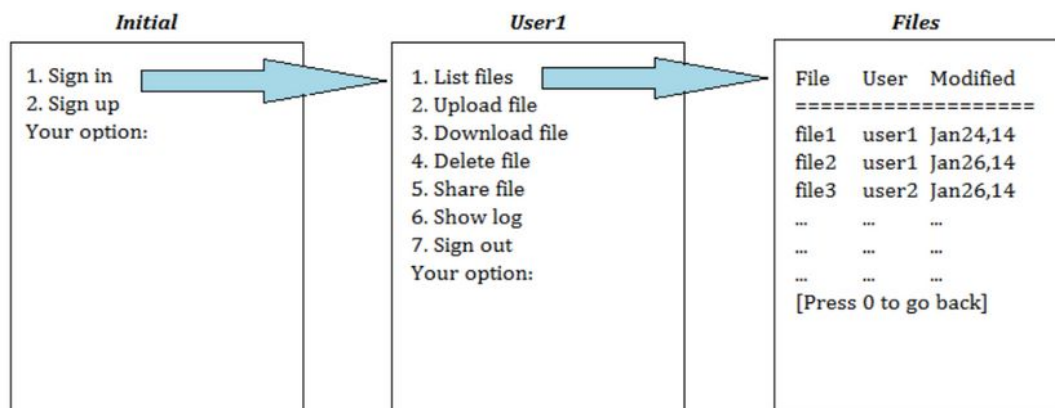
- I. **Sign in/ Sign up/ Sign Out:** User should send valid credentials to the server through the control channel. On successful signing, the server should send an ACK in return.
- II. **Join/ Leave/ List Group:** Each user has the option to join/leave a group. A user can also see a list of all the groups it is part of. Control and Data channels should be used for specific commands.
- III. **Broadcast a message to the group/ List all previous messages:** A user can send a group message. The users that are online get this message instantaneously. Save this message in a separate file inside the group folder. All users get an option to list all previous messages.
Format of message - [User, Timestamp, Message]
- IV. **List/ Delete files:** User should send proper “ls” or “rm” command to the server through the control channel. The server should execute those commands and return the output

(for “ls”) via the data channel as well as an ACK at the end via the control channel.

- V. **Upload/ Download files to a group:** Users should initiate the process by sending a suitable message via the control channel. The file data needs to be transferred via the data channel. In the end, an ACK message should be sent via the control channel.
- VI. **Share a file with other groups:** User should send proper “ln” (see the corresponding man page) command to the server via the control channel. After executing, the server should return an ACK via the same channel. Suppose user-1 of group-1 wants to share file-4 with group-2, then there has to be only one copy of file-4 inside group-1 directory in the server and a link to that should present in the group-2 directory in the server.
- VII. **Show log:** User should send proper “cat” command to the server via the control channel. After executing, the server needs to return the output via data channel and an ACK at the end via the control channel. The user gets to see the log in proper format.

Additional marks for any additional features/enhancements over the assigned features. Feel free to have a different design for the application, as long as the required functionalities serve.

Rough View of User-Actions



user1: Show log

File	User	Action	IP	Date/Time
file1	user1	upload	10.5.18.200	20 Jan'14
file2	user1	upload	10.5.18.200	20 Jan'14
file1	user1	share	10.5.18.200	20 Jan'14
file1	user2	download	10.5.18.204	20 Jan'14
file2	user1	delete	10.5.18.201	21 Jan'14
...
...
...