CN Final Project Report

Feature:

Basic:

1. registration
2. messaging
3. file transfer(multiple files)
4. user interface

Bonus:

1. auto reconnect
2. simple encrytion
3. add friends
4. send stickers
5. change between chat rooms

Video demo available here:

https://youtu.be/87d5uMe9r5w

How to compile our program?

$ make

User guide:

1. Start the program:

   Server:

   $ ./server <port_number> (Ex. $./server 11111)

   Client:

   $ ./client <IP:port_number> (Ex. $./client 127.0.0.1:11111)

2. Register and login to start charting with someone.
3. Type ":chat" to enter (or change) a chat room with someone.
4. To chat with someone, you have to add him/her as your friend, and he/she must add you as friend too.(type ":add" to add friend)
5. You can send multiple files using ":sendfile" command.
6. If someone has sent you some files, they will be stored in the server unless you type ":readfile" command to save the file to your local directory.
7. Type ":sticker" to send a sticker.
8. Type "logout" to logout.

Protocol specification:

TCP/IP

System/ Program design:

The whole system of our program is based on sending commands to the server, and receive response from the server.

Registration & Login System:

After connecting to the server, you have two choices which are registration or login. When you enter in registration, client will send the username and password to the server. Then, server will store the information into a file called member.txt containing user's information. Therefore, when you type your username and password to login, server will check whether this username is in member.txt and whether the password is correct. If both of the username and password is correct, you can enter into following system.

Adding friend & Changing chatting room:

If you type in :add, server will store the friend you want to add into a file called [username].txt at "Relation" folder. Thus, every user which has been registered has there own friend list in "Relation" folder. You can only add friends who are already registered. When you want to chat with someone, server will check whether your friend is in your friend list and whether you are in your friend's friend list. If both of the conditions are satisfied, you can chat with your friend.

Messaging system:

When the user want to send some message to another person through the client, the client will not only send the message, but also put headers in front of the message which contains some useful information for server to identify who is sending the message and whom to send the message to. For reading the message, the client will continuously send read message request to the server, and server will check the message file stored in server's directory to see if there is any new message to send to the client.

File transfer system:

When user A wants to send some files through the client to user B, the client will read the file from user A' s local directory, and send the content to the server. Once the server received the file, the server will first store them in the remote directory in the server. File transfer will be indicated by string like this <FILE--sample.txt> in the message history between the two users. When user B wants to download the file, he can download it by typing ":readfile", and the file will be stored to user B' s local directory.

Encryption function:

When two clients have a communication (for example, A and B have a talk), when A sends a message to B, client A sends a message to the server and sever displays the client-related information containing the sent content, in the same time server creates or search related file to store the users' conversation. Server encrypts the conversation content by using Caesar cipher, and then writes the encrypted content to the relevant file. After client B sends a read message request to the server, the server decrypts the encrypted content in the file through the relevant decryption function and the correct content is then sent to client B. Only the two users can know the chat content. No matter who opens the file, only a bunch of garbled characters will be seen.