

Student ID: z5195715

Name: Junyu Ren

COMP 9331 Computer Network and Applications

Assignment Report

Python version

I use Python 3.7.2

Program design & How system works

The whole program contains 2 source code files, server.py and client.py.

Server will listen the server port based on command line parameters. Whenever a client want to connect, server will start a new thread to communicate with the client. Server will give different response based on client side request. Server side will keep tracing the whole network condition (i.e online users, user login time, user block list, user message list, timeout checking etc). Server side will give the answer to the client terminal based on these variables.

Client will connect to the server subjectively when it starts. And then, user should input username and password to do authentication. After successful login, server will maintain connection with client, which allows client to input any supported commands and server will return the result. Moreover, client will act as a server to start a p2p thread for private message sending and receiving.

Application layer message format

(Only contains some self-defined message format, the standard command format not repeated here)

1. For wrong username: Invalid username, try again!
2. For wrong password: Invalid credentials, try again!
3. For continuous 3 times login failure: You are blocked to login, try again later!
4. For successful login: Welcome to this message application
5. Whoelse: Users current online: user1, user2...
6. Whoelsesince: Users that logged in: user1, user2...
7. Message: If A send message to an invalid target (i.e not in credentials.txt): Invalid receiver!
8. Broadcast: If you are blocked by someone: Your broadcast cannot be received by all online users because someone blocked you
9. Block: When block self or a non-exist user: Invalid user to block
10. Unblock: When unblock a user not blocked: Invalid user to unblock
11. Logout: other online user: xxx just logged out; Your terminal: You just logged out
12. Startprivate: start successfully: Connect to xxx successfully.
13. Private: If not startprivate beforehand: Have not built connection with xxx

14. Stopprivate:xxx stopped private conservation

Possible improvements and implementation

Maybe we can add a user registration function, thus we can write username and password into credentials.txt. As a result, the credentials.txt is very likely to perform as a database table.

For implementation, we can get the 'rw' right of the file and add new username and password at the end of file's last line by terminal input.

Moreover, we can let the main loop sleep a couple of seconds to release the CPU burden.

For implementation, we can simply add a line 'time.sleep(0.5)' at the top of 'while True' loop.

Addition

I did not attempt the p2p file exchange part because of time limitation.