COMP9331 Assignment - Online Discussion Forum

Program design

- server.py
  - function & class
    1. get_users(): reads credentials.txt and store username and password in users(a dictionary)
    2. test_username(username): check whether the username is already exists/logged in, and return its state
    3. login(username, password): check whether the password correct. If username is new user, add it into users and credentials.txt
    4. do_command(username, command): parses and executes the command. print error or success messages on server side and return a response string to be sent to client
  - Data Structures
    1. login_user: list to record all users that have logged in(delete element when XIT)
    2. users: dict(username : password), all users record in credentials.txt(include new users)
    3. threads: list to record all threads
    4. messages: dict(thread name : all message in the thread), all message in the thread is a list, each element has (number, user, message). If the message is UPD message, number should be 0
    5. created: dict(thread name : user that created the thread), use to check the RMV power
    6. adr_thread: dict(address : thread)
  - Workflow
- client.py
  - Workflow
    1. read port
    2. user login
    3. send command in a loop
    4. user exit

Message format

- In most cases, client directly send raw command and server send message string without special format.
- Special cases
  - login: when server find the user can login, it will send "new" when user is new user, or "current" when user is old user
  - TCP-UPD/TCP-DWN threadname-filename: when server get UPD or DWN command and checks the request, it will send TCP-UPD/TCP-DWN message to notify the client to start a TCP connection

How system works

- Server
  1. read port and record credentials.txt
  2. use UPD to receive message and check whether create thread by address
  3. In the thread, firstly handle login by check username and password

4. After successful login, enter loop to process client command continuously
5. If receive XIT command, the session ends and server will restart the login process next time it gets message from same address
6. When client sends UPD or DWN, firstly check whether there has error, if all OK, send message to call client use TCP do file transfer. Close TCP connection when transfer done.

- client
    1. read server port and use UPD to send information
    2. login first, and use the message from server check whether successfully login (If not, still try login)
    3. enter a loop to read command from user
    4. sent command to server and get message
    5. If get message to open TCP connection, set TCP socket to transfer file, when transfer done, close TCP socket

Handle multiple concurrent clients

The server uses multithreading to support multiple clients at same time.
All clients send messages to server via a UPD socket. The server listens using recefrom() and checks address.
If address is new, server creates new thread to handle that client. Each thread is responsible for only one client. It should start from login. The server stores each message in message_queue in different thread.