

Assignment Report

This report includes describing the program design, the application layer message format and a brief description of how my system works.

This assignment developed by Python 3.7.3.

The file include:

server.py, client.py and report.pdf

Program design

Server

In server.py, there are six Classes (User, UserDict, Message, File, Thread and Server) and 13 function. Those Class are used to represent an Entity and save the Entity's information. For example, User Class is used to save a user's information, such as username, password and login state(already login or not).

In Server Class, there are 18 functions. Function "start" is used to start the server. And function "acception" is used to bind and accept client, then allocates a thread for each client.

The rest of functions inside the Server Class are used to parsing the request from client and generate response for client, such as "recv_CRT".

The functions outside the Server Class are used to processing local files, such as "CRT" (which will CRT a Thread in the current working directory of the server).

1. When the server starts, server will load the data in "credentials.txt" and wait for the connection from the client
2. Here are the different scenarios that the server and client will give when dealing with different commands:
 - (1). Login stage
 - 1) Username stage: server receive username from client
If username already login, current user need to use another username to login
If username not exist, server will use current username to create a new user.
If username exist, current user can send password now
 - 2) password stage: server receive password from client
if password is correct, current user successfully log in(or create new user)
if password is wrong, current user failed to login, then repeat username stage
 - (2). CRT: Create Thread
If Thread exists, user fails to create Thread
If Thread not exist, user successfully creates new Thread
 - (3).MSG: Post Message
If Thread exist, user successfully posts to specified Thread
If Thread not exist, user fails to post message to specified Thread
 - (4). DLT: Delete Message
If Thread and message exist and the message belongs to current user, user

successfully deletes the message.

If message belongs to another, user fails to delete specified message.

If Message number not exists, user fails to delete specified message.

If Thread not exists, user fails to delete specified message.

(5). EDT: Edit Message

If message belongs to current user and message number is correct, user successfully edits message

If message belongs to another user, user fails to edit specified message

If Message or Thread not exists, user failed to edit specified message

(6). LST: List Threads

If no thread exists now, client will print No threads to list.

If Threads exist, client will list all exist Threads.

(7). RDT: Read Thread

If thread is empty, client will print Thread "threadtitle" is empty.

If thread has content, client will list the content.

If thread not exist, user fails to read from specified Thread.

(8). UPD: Upload file

If thread exist, file be uploaded successfully.

If Thread not exist, user fails to upload file.

If File already exist, user fails to upload file.

(9). DWN: Download file

If file exists, user downloads file successfully.

If file not exists, user fails to download file

If Thread not exist, user fails to download file

(10). RMV: Remove Thread

If Thread exists and thread belongs to current user, user removes Thread and all files belong to this Thread successfully.

If Thread not exist, user fails to remove thread.

If Thread belongs to other user, Thread cannot be removed.

(11). XIT: Exit

Current user successfully exits.

If current already logout (Just in case. Probably not happen.), current user fails to exit.

If User not exists (Just in case. Probably not happen.), current user fails to exit.

(12). SHT: Shutdown

If current user sends correct Admin password, the server will delete all Thread and related files and send shut down message to all client then server will shut down.

If current user sends incorrect Admin password, the server will not shut down.

Client

In client.py, there is only one class Client with 32 functions in it. In Class Client, several elements will be used to save message of the client, such as server's IP and server's Port. Each command has at least two functions for it (one for send message, another for receive message), such as

“send_CRT” and “recv_CRT”. Function “send_CRT” will handle command from users, then send request to server. Function “recv_CRT” will receive response from server, then give a result to user. Function “start” is used to start the client and connect to server.

When the client starts, it will read the user’s input and send to server. Then, based on the server's response, client will Output different results. The details are similar to previous scenarios.

how the system works

1. Open a new command line, type: `python3 server.py port_number admin_password`
2. Open a new command line, type: `python3 client.py 127.0.0.1 port_number`
3. To test multi-threading, please repeating the second step to create multiple clients.