

COMP3331 Assignment

Design Structure:

The forum program is implemented using java. In the assign file, the server is contributed by Server.java, package Credentials and package Forum. The client is only consisted of Client file and this file can be moved to other directories to run the client program.

As for the design of forum server, Server.java is to the TCP server to handle the received information and reply them. Credentials package is used to load and save credential.txt, and it will handle all the credential operations like login. Forum package stored the forum information (threads, messages and uploaded file) and it will handle all the forum operations like CRT, DWN and so on.

The server receives the message and send back the corresponding responses to the client. The client work in similarly way, except they will ask for the command input from the user then send them to the server.

Authentication:

As the client initiates, it will read the credentials.txt for user information. And when it shuts down, it will delete the credentials.txt in the current directory. If credentials.txt does not exist, it will create an empty one.

As the user enter the username, the client will send the following to the server.

Username username

Then, after server will do the credential operation, it will return the following message depends on different situations.

Password (if the username exists)

New Password (if the username does not exist)

Already_Login (If the username is online)

After the client receive the above message, it will ask the user to enter password and send the message in similar format to the server to verify the password and the server will tell the client whether the password is correct or not.

Forum Operation:

As for CRT, MSG, DLT, EDT, LST, RDT, RMV and XIT commands, they are implemented in a similar way as above. After the client check the method and syntax are correct, it will directly message the command to the server. Then, the server will do the forum operation in the Forum package and send the result back to the client. These commands always complete in 1 RRT.

As for UPD and DWN, they need at least 2 RRT to complete the operation. As for UPD, the client will first check whether the file exists. If so, it will send the command to the server. After the server tells the client that it is ready to receive the file, then the client will send the file in binary to the server and the server will keep receiving the file until it receives the upload completion message from the client. The DWN works in the similar way.

As for SHT, any client can shut down the server with correct admin password. When the client is initiated, it will create a new thread to keep sending check message to the server for the purpose to check the server is on or shutting down. Therefore, once the server is shutting down by one of the clients, the other clients can also detect the shutdown of the server. Then, it will say goodbye to the user and shutdown the client.

Limitations and Possible Improvement:

When the client accidentally encountered an unexpected bug, it may just be stuck at the stage waiting for the server's response. And the user cannot do anything unless manually force to shutdown the client. If so, the user cannot log in anymore because the server did not detect the manually forced termination of the client and it will just keep assuming it is online. Also, if the server is manually forced terminated, it will not delete the files in the working directory. So, we must manually delete those files, otherwise, it will cause some errors when the server is turned on again.

The possible improvement for the above scenarios is to take account of the unexpected bug and add some functionality to take care of them. For example, once the specific exception is caught, it will do something to protect the client or server before the termination.

Some Advice for Testing:

I think my program has all the functionality in the specifications and I have solved all the bugs that I can find. However, just in case, you find a bug when testing which causes a stuck at the client and you have to manually terminate the client. Just remember to manually delete all the files (including replacing the credentials.txt) in the server working directory, so that the server can work normally again for the following tests. Or you can use another client to normally shut down the server if the server is not stuck at some points.