Comp3331 assignment report

z5248147

Program Design:

Language: Python 3

Platform: Linux

Transport Layer Protocol: TCP

Application layer message format: Encoded in utf-8.

How the system works:

Server: A dictionary is used at server as a global variable to store all the active threads, logged clients and connected sockets. Firstly, the server will start a new thread to accept new connection sockets, and the main thread will loop until the server shutdown. The thread that accepts new sockets will then loops and start a new thread 'client handle' to process requests when a client connects to the server. There are 2 loops in the handle function which are handling authentications and handling requests, and the user will have access to the second loop after successful authentication. If the socket that connected to the server sends heartbeat packets, the specified username will be the signal for the program to get into another loop in the first large loop in the handle function, and it will send messages to clients to prove the server is alive. In the second loop of handle function, a thread lock is used to avoid conflicts in these different threads, and if-else is used to process different requests.

Client: Firstly, the client will start a new thread to send heartbeat packets to server to make sure the server is alive, and it will kill the program when the server is shutdown. Then the main thread includes 2 loops which are handling authentications and handling requests, and the user will have access to the second loop after successful authentication.

Design trade-offs:

At first, only one thread is used at client side to process requests. However, that won't shutdown the clients when the server shutdown, so a heartbeat packets method is used to detect the status of the server.

Possible improvements and extensions:

There are many repetitive codes in both files, and they can be pull out and put in several functions to increase readability.