

COMP3331 Report

Z5308268, Thomas Forster

Program design

The program was written in python 3.7.

How the system works

The server handles multiple clients concurrently using the `select()` function. Creating TCP sockets for each client, and handling any commands received in order.

The usage for the select code was borrowed from <https://pymotw.com/2/select/>

Any action can be done while any other action is happening, except for upload. This means that a user can still be in the authentication phase, having just entered a username, while another user can login, download files and do any other command.

Application layer format

All messages are sent as a string, with the first character being the command, and a Unicode character (■) to separate each message.

Server to client:

Command	Name	Description
E <i>msg</i>	Error	Prints an error message in red
M <i>msg</i>	Message	Prints a normal message
I <i>msg</i>	Input user	Gets input from the user, sends a login attempt with the response
N <i>msg</i>	Input register name	Gets input from the user, sends response to register the name
C <i>msg</i>	Input command	Gets input from the user, sends response as a command
L <i>msg</i>	Input login	Gets input from the user, sends response with the given username and attempts to login
R <i>msg</i>	Input register	Gets input from the user, sends response with the given username and registers a new user
F <i>filename</i> <i>fileid</i>	File upload	Receives a filename and signals to send the file
D <i>filename</i> <i>filesize</i>	File download	Receives a filename and size and signals to begin receiving the file
U <i>username</i>	Username	Sets the username for the client
Q	Quit	Signals to close the socket and exit

Client to server:

Command	Name	Description
C <i>user cmd</i>	Command	Sends the username and given command to be executed
U <i>username</i>	Username	Sends the username to attempt to login with
N <i>username</i>	Register name	Sends the username to attempt to login with, or register if the name does not exist
L <i>username</i> <i>password</i>	Login	Sends the username and password to attempt to login with

R <i>username</i> <i>password</i>	Register	Sends the username and password to register a new account
F <i>fileid</i> <i>filesize</i>	File	Sends the fileid and size of the file to be uploaded. Signals the server to begin receiving

Limitations

Shutdown currently does not interrupt the client asking for input from the user, meaning that the user must enter any input before the client closes.

Users can not do any commands while a file is being uploaded by any user.

Possible Improvements

Fixing shutdown, which can be done with a better implementation of multithreading.

Allowing multiple uploads and downloads at the same time, along with progress bars for these on the client side. This would require implementing multithreading on the server side.

A GUI or a better way to select and execute commands.

Currently there is no explanation for what the commands do, or the correct usage for these commands. Meaning that every user will have to know from the manual/assignment specification. A help menu or welcome screen could solve this.