Program 5: FTP client

Aika Usui

# Client’s interface in terms of OPEN, PASS/SYST, and QUIT

## OPEN, PASS/SYST

A picture containing refrigerator

Description automatically generated

The program takes two arguments, hostname and port number. The program establishes a connection with the arguments and if it’s successful, the program proceeds to prompt username. Executes USER username command and gets message from server. After that, the program proceeds to prompt password. Executes PASS password command and gets message from server. If login is successful, the program sends SYST command and executes it.

## QUIT

A close up of text on a black background

Description automatically generated

**Close**

When the user input is close, the program closes the connection but not quit ftp. Executes QUIT command and gets message from server. At this point, connection is closed. Since ftp is still available, prompt ftp> for next action.

**Quit**

When the user input is quit, the program closes the connection and quit ftp. This is showed with purple allow in the diagram. If the program is still connected to FTP server, executes QUIT command, then exit out the program. If not, just exit out the program.

# Client’s get function in terms of PASV, RETR, and local file options (O\_CREAT | O\_WRONLY and S\_IRUSR | S\_IWUSR | S\_IRGRP | S\_IROTH)

A picture containing clock

Description automatically generated

When the user input is get filename, check if user inputs filename or not. Then, executes PASV commands and opens new socket by calculating port number based on the message. After it is successful, execute another command, TYPE I, and gets message. Finally, executes RETR filename command. This only copy and create a file in the local file, but it doesn’t copy the content of the file. Thus, you need to retrieve the content of file by reading and write it into the file. In the end, gets message form server.

# Client’s cd function

A close up of a device

Description automatically generated

When the user input it cd, check if there is a following argument for subdir. If there is, execute CWD subdir command and gets message from server. Then the directory is changed based on subdir input.

# Client’s ls function in terms of PASV, LIST, and file output to stdout

A close up of a map

Description automatically generated

When the user input is ls, the program goes into passive mode. Executes PASV commands and opens new socket by calculating port number based on the message. After it is successful, the program forks a child process. In the child process, it reads data and close the socket. (Ends passive mode) In the parent process, Execute LIST command and reads message. Then, wait for child process to finish and reads message again.

# Client’s put function in terms of PASV, STOR, and local file option (O\_RDONLY)

A close up of a sign

Description automatically generated

When the user input is put, prompt input for localfile and remortfile. Then, executes PASV commands and opens new socket by calculating port number based on the message. After it is successful, execute another command, TYPE I, and gets message. Finally, executes STOR remortfile command. This only copy and create a file in the server, but it doesn’t copy the content of the file. Thus, you need to write the content of file into the file in the server. In the end, gets message form server.