FILE TRANSFER APPLICATION USING SOCKET PROGRAMMING

Abhinay podugu -- CS18B027
Sai Thanoj   -- CS18B005

Introduction

- ❖ This application is made using python language.
- ❖ Server can have multiple client requests concurrently.
- ❖ This is achieved using threading.
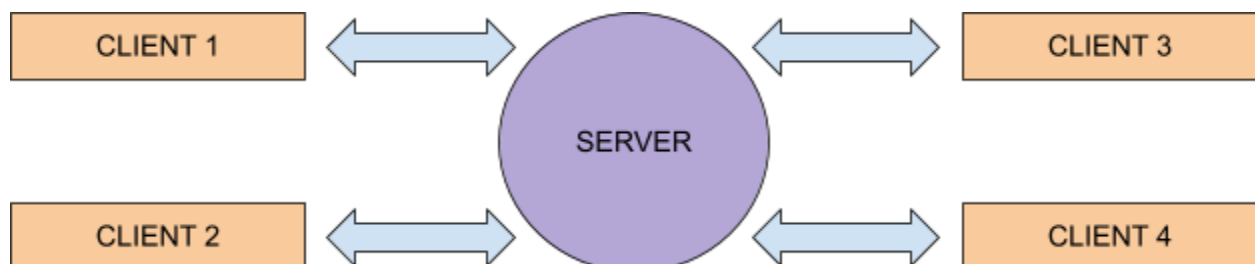- ❖ Each client is assigned a thread by the server as they connect.

Features of Application

- ❖ **LIST** **:** List all files in the server.
- ❖ **HELP** **:** Show all the commands available in the application.
- ❖ **UPLOAD :** Upload a file to the server. File should be given with the path in the client system.
- ❖ **DOWNLOAD :** Download a file from the server.
- ❖ **DELETE :** Delete a file in the server using the filename.
- ❖ **LOGOUT :** Logout from the server.

Compiling Instructions

- ❖ Refer to the screenshots below for compiling instructions.

Design of our implementation

Screenshots of some scenarios

View Help Commands: run **make**

```
abhi@Abhinaypodugu:/mnt/c/Users/COMP/Desktop/Socket PRogramming$ make
--------------HELP--------------------
To setup the server type ---- make server
To setup the client type ---- make client
------------------------------------------
```

Run Server: run **make server**

```
abhi@Abhinaypodugu:/mnt/c/Users/COMP/Desktop/Socket PRogramming$ make
--------------HELP--------------------
To setup the server type ---- make server
To setup the client type ---- make client
------------------------------------------
abhi@Abhinaypodugu:/mnt/c/Users/COMP/Desktop/Socket PRogramming$ make server
python3 server.py
[STARTING] Server is starting
[LISTENING] Server is listening on 127.0.1.1:5050.
```

Now the server is listening on port number 5050.

Run Client: run **make client**

<u>Server Terminal</u>

```
abhi@Abhinaypodugu:/mnt/c/Users/COMP/Desktop/Socket PRogramming$ make
--------------HELP--------------------
To setup the server type ---- make server
To setup the client type ---- make client
------------------------------------------
abhi@Abhinaypodugu:/mnt/c/Users/COMP/Desktop/Socket PRogramming$ make server
python3 server.py
[STARTING] Server is starting
[LISTENING] Server is listening on 127.0.1.1:5050.
[NEW CONNECTION] ('127.0.0.1', 55688) connected.
[ACTIVE CONNECTIONS] 1
```

<u>Client Terminal</u>

```
abhi@Abhinaypodugu:/mnt/c/Users/COMP/Desktop/Socket PRogramming$ make client
python3 client.py
Welcome to the File Server.
---------- Commands are not case sensitive ----------.
LIST             : List all the files from the server.
UPLOAD <path>    : Upload a file to the server. Give the path of the file to be uploaded to the server.
DELETE <filename>: Delete a file from the server.
LOGOUT           : Disconnect from the server.
HELP             : List all the commands.

> 
```

Number of active client connections is shown in the server terminal.
New connection details are also shown in the server side terminal.

View Files List in the server: use **list** command

```
abhi@Abhinaypodugu:/mnt/c/Users/COMP/Desktop/Socket PRogramming$ make client
python3 client.py
Welcome to the File Server.
---------- Commands are not case sensitive ----------.
LIST             : List all the files from the server.
UPLOAD <path>    : Upload a file to the server. Give the path of the file to be uploaded to the server.
DELETE <filename>: Delete a file from the server.
LOGOUT           : Disconnect from the server.
HELP             : List all the commands.

> list
The server directory is empty
> 
```

No files in the server_data directory

Upload Files: use **upload** command

```
abhi@Abhinaypodugu:/mnt/c/Users/COMP/Desktop/Socket PRogramming$ make client
python3 client.py
Welcome to the File Server.
---------- Commands are not case sensitive ----------.
LIST             : List all the files from the server.
UPLOAD <path>    : Upload a file to the server. Give the path of the file to be uploaded to the server.
DELETE <filename>: Delete a file from the server.
LOGOUT           : Disconnect from the server.
HELP             : List all the commands.

> list
The server directory is empty
> upload client_data/singleline.txt
File uploaded successfully.
> upload client_data/multiline.txt
File uploaded successfully.
> list
multiline.txt
singleline.txt
> 
```

Initially the server directory was empty. Then we upload two files which can be confirmed with the **list** command.

Delete, Help and Logout: use **delete**, **help** and **logout** commands respectively.

<u>Client Terminal</u>

<u>Server Terminal</u>

Delete a file in the server using the file name. List the commands using the help command. Logout the client from the server using **logout** reflects in server Terminal**.**

Download a file from server: use download <filename> command

```
abhi@Abhinaypodugu:/mnt/c/Users/COMP/Desktop/cs18b027_lab1$ make client
python3 client.py
Welcome to the File Server.
---------- Commands are not case sensitive ----------.
LIST               : List all the files from the server.
UPLOAD <path>      : Upload a file to the server. Give the path of the file to be uploaded to the server.
DELETE <filename>  : Delete a file from the server.
DOWNLOAD <filename> : Download a file from the server.
LOGOUT             : Disconnect from the server.
HELP               : List all the commands.

> list
readme.txt
> download readme.txt
File downloaded successfully.
>
```

list command lists the files available in the server. **download** command can be used by the client to download the file. The file downloaded can be found in client_data directory.