FTP project

Presented for :

Dr.Amjad Al-Hajjar

Presented by :

Mohmmad Haidar 3967

The objective of this project is to create a program to manage an FTP server and able to handle multiple connections using sockets, and a user prompt (DOS) application that send the commands to the server.

The following commands needs to be implemented:

- **connect IP port**: connect to the server specified with IP address and port.
- **LOGIN X Y**: user ID X and Password Y.
- **dir**: list the files in the current directory of the server side.
- **ldir**: list the files in the current directory of the client side.
- **cd X:** change the current directory on the server side to X.
- **lcd X**: change the current directory on the client side to X.
- **put X Y**: upload the local file X to the server and name it Y at server side.
- **get X Y**: download the server's file X to the client and name it X at client.
- **Disconnect.**
- **Bye**: closes the application.

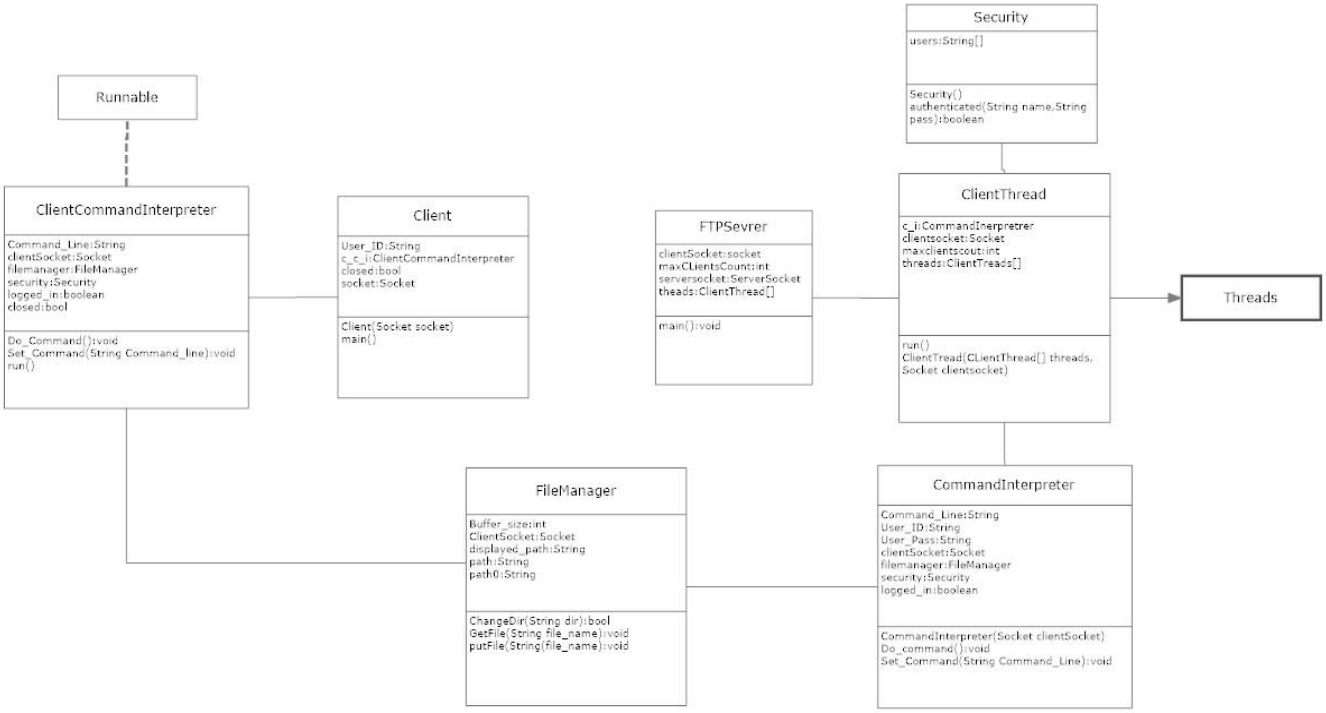
I used the java language in this project and took advantage of the ease of multithreading and working with the sockets.

I made the server able to handle 10 connection max, and uses the 2222 port for the control (commucation through Strings)

And the 2221 port for Data transfer.

The Data transmission is buffered on the acceptor side and I used a 1024 bits buffer size.

The following is the Design of the program:



I will next describe the function of each class.

Client:

The client class is the class that contains the main function for the client, it forwards every line written by the user to the class ClientCommandIntrpreter as a full line String.

ClientCommandIntrpreter:

This class implements Runnable so that it can have a run() method , and it takes the command line and extract the command from it using the set_command method. Then the method Do_Command identifies the command and proceed with it.

There is three different categories of commands :

1. Command that don't requires connection:
   Connect, help, ldir, lcd.
2. Commands that requires connection only:
   dir, cd, login, disconnect.
3. Commands that requires connetion and login:
   get, put.

The connection is done through the 2222 port, and when connected it creates a thread to listen to server and display what is received or to execute commands from the server.

cd and lcd can take as parameter the name of a folder existing in the directory or .. to go back.

The commands cd, dir, are sent to the server to get the and take the response from it.

The commands lcd, ldir are forwarded to he class FileManager in the client side.

The commands put and get are Forwarded to the FileManager in the client side and to the server.

FileManager(client side):

This class execute the commands related with files and folders,

ListDir and ChangeDir use the property user.dir .

GetFile and PutFile connects to he server through 2221 port and start sending or receiving binary file with a buffer of 1024 bits.

FTPserver:

This class waits for clients to connect and give each client a ClientThread into an array of MaxClientsCount (10) length.

ClientTread:

This class Extends Thread class , this allow it to use Thread methods including the run() method, this entire class is triggered from the FTPserver class and run in parallel with other ClientTread classes. It reads commands from the socket (from the user) and forward them to the CommandInterpreter.

CommandInterpreter:

This class is similar to the ClientCommandInterpreter but it executes the client commands like login that get authentication using the class Security, and return the answer to the Client through the socket.

The commands get and put are forwarded to the FileManager class server side but are swapped because the get command from the client is actually a put command for the server.

FileManager(server side):

This class is the same as the client side but the difference is in the GetFile and PutFile methods we create a serversocket for data stream and waits for the client to connect to it.

Security:

This class is used here for authentication it contains a function that checks for the username and password from the data base and return true if it is matched. The database contains only one user “admin” with password “admin”.

This class can be used in future to make the connection more secure, we can put functions for encryption and decryption and for hash to use them in the transmission of data.