Group Practice - FTP System

David Prieto, marta simon, ernesto uson, ignacio hernandez

2022

# Implemented Modules

1. Base
2. Authentication
3. User Control
4. Directory System
5. Delete and rename files
6. Passive Mode

## Base Module

The base module consists in the command “port”, “list”, “retr”, “stor” and “quit”.

### Port

For the command port we face some problems due to the “BufferedReader”. Once you transmit the message and the server tries to read it the reader return nulls which break the server. In order to fix this issue, we change the command to “prt” following the advice of the teacher.

The behavior of the command is the following: the main class of the server ask the sub connection class to start the sub connection at the port specified by the user.

### List

The command “list” must start a sub connection if there is no one running yet. Once the sub connection is ready, the server stars reading all the names of the files that are inside the folder specified by the user inside the command. If the user did not specify any path, the server will start reading from the root of the server.

All these names are stored inside a txt file which will be transmitted to the client. The client will read it, print the information in the console and then delete the file. The server will also destroy the file to avoid problems reusing the command.

### Retr

As the previous commands, if there is no sub connection running, the first thing the command will do is to start a sub connection following the method specified by the server inside the settings file.

Once the sub connection is running, the server will start by converting all the file into an array of bytes. As this array can be extremely long and the “PrintWritter” could not be able to transmit the whole array in only one package, we have decided to divide the information of the array into small packets following the specified size in the settings file.

Once we have the name of the file, the full length of this array and the number of packages, this information is transmitted to the server. Once the client knows all this information the server can start creating the file and the byte array to store the information.

Then the server starts to send the packages one by one. On the client side, the packages are being read and stored in the byte array.

Once all these packages have been received by the user, all the information stored inside the byte array will be written inside the file.

### Stor

This command works exactly as the “retr”. The only difference is that is the client who transmit the file. Other difference is that the file will be stored in the working path of the server.

### Quit

The command enable a boolean who stops the current session in the server and destroy all the connection, readers and other stuff related with the session to create a new ones for other session.

On the client side, this command ends the session and stops the program.

## Authentication

The authentication consists in a new command: “user”. The behavior of this command is the following:

the client will introduce “user <space> <UserName>” the server will ask the client to send the command “pass <space> <Password>” once the server knows both username and password the server will respond that the user is logged.

## User Control

Following with the Authentication module, in this module the server is required to have the usernames and the passwords stored inside in order to check that the combnination of username and password provided is a valid combination.

The server stores this data in a txt file reader at the start of the server program and stored inside an arraylist inside a class named “UserControl”. The main class can ask this class if a pair (username + password) is valid. Depending on the respond of the “UserControl” the server will respond to the client with a valid respond or a failure one.

## Directory System

This module asks the server to be able to manage different folders and to be able to move inside this one. In order to be capable of do these actions the server need new commands.

* “Pwd”: The server will return the current working path where the files will be stored if transmitted via “stor”.
* “Cwd <space> <path>”: allows the user to change the working path of the server by introducing the name of the folder which wants to add to the path.
* “Cd”: allows the user to exit the current folder and to return to the parent of the actual folder.

## Delete and rename Files

This module requires the server to be able to create new folders, delete folders and files and to rename existing files. To be able to do these actions the server adds the following commands:

* “Mkd <space> <nameOfFolder>”: Creates a folder inside the actual working path.
* “Rmd <space> <nameOfFolder>”: Removes the provided directory, the folder must be in the working path.
* “Dele <space> <nameOfFile>”: Removes the provided file, the file must be in the working path.
* “Rnfr <space> <nameOfExistingFile>”: Will select the file or folder in the working path, then the server will ask for a “Rnto <space> <newNameOfFile>” in order to know the new name of the file or folder.

## Passive Mode

The passive mode is a method used to start the sub conection. In the active mode, the client is the one who tells the server the port which must connect to. In the passive mode is the server the one who tells the client the port which must connect to.