

## Programming Project 1: Simple FTP Client and Server

The aim of this project is to introduce you to the basics of the client-server model of distributed systems. In this project, you will be designing and implementing simplified versions of FTP client and server. The client executable will be called “myftp” and the server executable will be called “myftpserver”. You are only required to implement the following FTP commands. The syntax of the command is indicated in the parenthesis.

1. `get (get <remote_filename>)` -- Copy file with the name <remote\_filename> from remote directory to local directory.
2. `put (put <local_filename>)` -- Copy file with the name <local\_filename> from local directory to remote directory.
3. `delete (delete <remote_filename>)` – Delete the file with the name <remote\_filename> from the remote directory.
4. `ls (ls)` -- List the files and subdirectories in the remote directory.
5. `cd (cd <remote_directory_name> or cd ..)` – Change to the <remote\_directory\_name > on the remote machine or change to the parent directory of the current directory
6. `mkdir (mkdir <remote_directory_name>)` – Create directory named <remote\_directory\_name> as the sub-directory of the current working directory on the remote machine.
7. `pwd (pwd)` – Print the current working directory on the remote machine.
8. `quit (quit)` – End the FTP session.

The workings of the client and server are explained below.

**FTP Server (myftpserver program)** – The server program takes a single command line parameter, which is the port number where the server will execute. Once the myftpserver program is invoked, it should create a TCP socket, bind, listen and block on accept (for incoming connections). Appropriate error messages need to be displayed in case any of the above fail. When a client connection arrives, it should start accepting commands and execute them. Appropriate error messages need to be communicated to the client upon failure of commands. Upon receiving the quit command, the server should close the connections do all housekeeping and be ready to accept another connection. The directory where the server program resides should be current working directory for each incoming client connection (i.e., if the first command entered by a client is “pwd” it should see the path of the directory where the server program resides).

**FTP Client:** The ftp client program will take two command line parameters the machine name where the server resides and the port number. Once the client starts up, it will display a prompt “mytftp>”. It should then accept and execute commands by relaying the commands to the server and displaying the results and error messages where appropriate. The client should terminate when the user enters the “quit” command.

**Points to note:**

1. The server can be either iterative or concurrent (single or multi-threaded) for this project. However, please be advised that your second project will build upon the first project and you will be asked to have multi-threading at the server end. So, in order to avoid duplication, you may as well do multi-threading in the first project.
2. You can assume that the user uses the correct syntax when entering various commands (i.e., you are not required to check for syntax).
3. You may use C/C++/Java for your project.
4. You do not need to support user logins. Also you do not need to check for user permissions when executing various commands at the server end.
5. All programs will be tested on Linux platform (nike). It is your responsibility to test and make sure that your program works correctly on a Linux machine.

**What to Submit:**

A Zip folder containing:

1. Your code for client and server.
2. A readme file containing (a) names of the students in the project group; (b) any special compilation or execution instruction; and (c) the following statement – if it is true – “This project was done in its entirety by <Project group members names>. We hereby state that we have not received unauthorized help of any form”. If this statement is not true, you should talk to me before submission.