

Project 1: Building an FTP Server

Grade: 100 pts

Team: 3-4 students

Introduction

Most network applications rely on file transfer protocols in one form or the other. For instance, the HTTP protocol used in the Web is a generic file transfer protocol. In this project, you will implement a FTP client program and a FTP server program for a simple file transfer. At any given time, the server could handle one or more file transfer to the client (s). The implemented FTP application supports text files transfer. The client program presents a command line interface that allows a user to:

- Connect to a server
- List files stored at the server.
- Download (retrieve) a file from the server.
- Upload (store) a file from the client to the server.
- Terminate the connection to the server.

The server program binds to a port and listens for requests from a client. After a client connects to the server, the server waits for commands. When the client sends a terminate message (quit), the server terminates the connection and waits for the next connection.

Implementation

You can implement your project using any programming language as two independent programs, a ftp client called **ftp_client** and a ftp server called **ftp_server**. The ftp_client program presents a command line interface. The communication between the client and the server needs to be implemented using TCP sockets. The client should be able to send FTP commands to the server. On receiving a command, the server should parse the command and perform the appropriate action. The format of the commands is such as follows:

1. **CONNECT** <server name/IP address> <server port>: This command allows a client to connect to a server. The arguments are the IP address of the server and the port number on which the server is listening for connections.
2. **LIST**: When this command is sent to the server, the server returns a list of the files in the current directory on which it is executing. The client should get the list and display it on the screen.
3. **RETRIEVE** <filename>: This command allows a client to get a file specified by its filename from the server.
4. **STORE** <filename>: This command allows a client to send a file specified by its filename to the server.
5. **QUIT**: This command allows a client to terminate the control connection. On receiving this command, the client should send it to the server and terminate the connection. When the ftp_server receives the quit command it should close its end of the connection.

What to turn in?

The report should include the basic logic of the program, what problems you have encountered, how you solved them and screen capture demonstrating how **each service** (connect, list, retrieve, store and quit) provided by the FTP server works. The report should be submitted in hard copy format and source code should be submitted using github. Please send me email with a link to your source code.

Team leads in bold underlined fonts.

Team Name				
Red	Banaszak, Nathan	Chlebek, Adam	Turnbull, Joseph	<u>Turnbull, Jemima</u>
Green	Grewal, Harsimran	<u>Hinkley, Shayla</u>	Kiser, Stephen	Knibbe, Joshua
Purple	<u>Crowe, Logan</u>	Dys, Tyler	Going, Griffin	Goncher, Noah
Orange	Hood, Paul	Isaacs, Samuel	<u>Kopplin, Aaron</u>	
Blue	<u>O'Laughlin, Kade</u>	Rufino, Kevin	Jacobson, Kyle	Jaglowski, Logan
Maroon	Strohschein, Dylan	Tiba, Mohamad	<u>Kingsley, Tristan</u>	