CSC/CPE 4750 Lab 3
Assigned: 4/19/19
Due: 5/1/19* by 11:59pm
You may work in pairs (or individually)

Overview

An FTP server is live on the computer in the CSC Projects Room[1] and will be through the due date for the assignment. The server is located at 72.233.138.35* and should be accessible on campus. The main goal is to make a client to send arbitrary commands to the FTP server, including retrieving/sending files. You will test your client by downloading a file from the FTP server, modifying it, and pushing it back.

*the IP address will be listed on the Canvas assignment page, and Slack #assignments channel. If the IP address changes, it will be updated in both spots. When you see **IP** in this assignment, replace with the IP address from Canvas/Slack.

Before you Start

Contact Dr. Dingler to have a user created on the server; provide the username and password you want -- we do not care much about security here, but make it moderately strong, *and do not use a password you use for anything else*.

Note: You can use any programming language you choose for this assignment.

Hint: WireShark can be very helpful when working on this lab, e.g., to see the messages sent between the command line *ftp* program (available in Windows, etc.) and the server.

Goals

1) Write a program to initiate communication with the FTP server, it should:
    a. Initiate a TCP connection to the server on **port 24601** to send *commands*.
    b. Authenticate with the FTP server using your username and password (you can hardcode this, to simplify testing).
        i. Make sure to follow the format for FTP messages (look at RFCs, online, etc.).
        ii. Output all FTP responses to the screen.
    c. Ask the user (via keyboard input) for an FTP command. Modify the user's input to ensure it is terminated correctly. Test by writing a "PWD" message and printing the response.
    d. Next, test with "LIST" (again, print the response). Why didn't we see the list of files? Since FTP sends data out of band, the server will attempt to open a *second* TCP connection to the client. We cannot do that at SPU (and possibly on your home network) due to our firewall, so…

[1] This is not directly accessible from off campus. You can use the RD Gateway instructions to access the server. Or, you could install `vsftpd` on a machine of your own to test. If you do that, contact Dr. Dingler for a config file to be sure the settings match.

- i. There is an FTP command (PASV) we can send that will have the server listen on a new port. Do some research to find this command, its format, and (this is key) its response.
- ii. After sending the appropriate command, initiate a second TCP connection to the FTP server to receive data. This 2-step process has to be done for **each** request, FTP uses non-persistent connections for data. Make sure you use the correct port (not the same as for the command socket).
- iii. Print the response (i.e., the *data*) from the server and close the socket.

e. Now that we can initiate data connections and request lists of files, modify your program so you can retrieve and push files to the server.

- i. Contents of files retrieved should be printed to the screen *and* written to disk.
- ii. It may be helpful to push a file and test **deletion** as you'll likely need it in the final step.

f. Next, you will retrieve, modify, and push a file for your user:

- i. A file called `file.txt` has been placed in a sub-folder *in your home directory*: `/home/username/complete/file.txt`
- ii. Have your program retrieve `file.txt` and modify its contents. This can be done in the program itself, or you can `RETR` the file, modify it with a text editor, and `STOR` it back. You may modify the contents to be anything you choose.
- iii. Finally, push your changes back to `file.txt` on the server. Double-check the `STOR` works by doing a `RETR`, or using another FTP program.

g. Make sure your program closes any open sockets when completed.

Important warning: students in 2015 had trouble completing this lab when testing on a *wireless connection*, primarily after implementing the data band connection in step d. *In addition*, some students had issues connecting from behind a router that uses NAT *even with a wired connection*.
Tip: if you are testing d, run Wireshark to keep an eye on the traffic. The symptom to look for is repeated TCP packet retransmission on the port(s) you are using. If this happens, try your program from `luke`.

What to Turn In

Via Canvas, turn in (one per group):

- Your code for the FTP program
- An output listing for the program – it should indicate messages entered by user, and all server responses. To summarize: the client authenticates, the user enters commands to test PWD, LIST. The user then enters a command to retrieve `file.txt`, modify it, and push it back to the server. Dr. Dingler will check your directory on the server to ensure the file was uploaded, and also run your program and try the process above.