COMP 342: Data Communications and Networking

Spring 2023

Project 1: FTP Server and Client

Date Released: 3/17/2023

**Assignment:** In this project, we will implement an FTP-like server and client. The FTP protocol is used to transfer files between a client and server.

The FTP server starts by listening on a socket on port 9001. Note that FTP usually works on ports 20 and 21 but for this project we are implementing a simpler smaller FTP. The FTP server should accept the following commands:

* LIST: lists the files in the current directory. The current directory should be specified as a member variable currentDirectory in the FTPServer class. To accomplish this, look up how to list the files in a directory in Java. Hint: you can use the File class.
* RETR filename: sends the file specified in the filename to the client
* STOR filename: retrieves the file specified in the filename from the client to the server
* PWD: lists the current directory. This command returns the path stored in the member variable currentDirectory.
* QUIT: closes the connection with the client

The FTP client should start by connecting to the FTP server. Once connected, the user should be allowed to enter any one of the commands supported by the FTP server.

|  |  |
| --- | --- |
| FtpServer | FtpClient |
| Welcome to GCC FTP service!  Waiting for client commands…  Connection terminated by the client… | Welcome to GCC FTP client!  Command: LIST  myfile.txt  readme.txt  Command: PWD  C:\Users\student\342\prj1  Command: STOR readme.txt  File stored correctly  Command: RETR readme.txt  Here is the readme file contents!  Command: QUIT |

**Rubric:** To receive credit, submit a .zip file via mygcc containing all your source code, and a README.txt file, describing how to run the code. If you worked on the bonus part, please also submit a doc file.

Points will be awarded based on the following:

|  |  |
| --- | --- |
|  | Points |
| The project establishes a connection between the client and the server and terminates it using QUIT command. | 10 |
| The client sends the commands correctly. | 20 |
| The server implements the LIST command correctly. | 10 |
| The server implements the PWD command correctly. | 10 |
| The server implements the RETR command correctly. | 20 |
| The server implements the STOR command correctly. | 20 |
| Program works for txt and png files. | 10 |
| **Bonus**: Run client and server on two different PCs; then, send a large file (>10MB) from client/server to server/client and calculate how much time it needs to deliver the whole file. By ignoring the propagation time and the queueing time, estimate the Bandwidth.  Write a short report to explain the setup, the implementation (e.g. how to obtain the transmission time), and the result (e.g. does the estimated Bandwidth make sense? Why?). | 10 |

**Your code must execute under Eclipse or IntelliJ (preferred). Also, if your code crashes, no partial credit will be given**

Modularity is required in the design of your program. Use good software design and engineering techniques, such as procedural abstraction (as appropriate), to increase the extensibility, maintainability, and readability of your code.

In addition, you should include a comments section at the beginning of each of your files that provides information about the file and its intended purpose. For example,

/\*\* Author: David Zhang

\* Course: COMP 342 Data Communications and Networking

\* Date: 16 March 2022

\* Description: put a description here

\*/

**Without the comment section, 5% grade will be taken away.**

The following criteria will be used to grade your submission:

* Does the code function according to the problem specification?
* Is there an appropriate comments section at the beginning of each file (similar to the one shown above)?
* Are there inline comments before each block of code?
* Is the code readable and well-formatted? Is it well-documented and clear?

**A program that does not run without crashing will not be graded.**

**Extensions will not be granted for technology-related issues.** Leave yourself enough time to complete the assignment, submit the assignment using mygcc, and contact the instructor if you run into problems.

**Project Policies:**

* Assignments must be submitted electronically via my.gcc. Be sure to upload your files correctly the first time.
* This project is a group project. Every student needs to work as a part of a 2-person team. If you cannot find a team to work with, contact Dr. Zhang and he will assign you a team. Students are expected to stay in the same team used for Project 1.
* 20% of the grade will be weighed with the peer evaluation. Students are expected to turn in the peer evaluation form posted on mygcc at the end of the semester. If a student works on his or her own without a team, then 20% of the grade will be deducted from the project grade.
* Students are expected to keep the same team for the whole semester.

**Academic Integrity Policy:**

* Each team is expected to work on its own. Members of each team can work together, discuss ideas, help each other on debugging etc. **Please do not attempt to look at each other’s code unless you have to**.
* Students belonging to different teams should not discuss, share code that directly bears on this project, or look at each other’s code. Any instances of this will be considered a violation of the academic integrity policy of this course and will be reported to the SFRC committee.
* Use or possession of past solutions and similar solutions from online resources is strictly prohibited and is considered a violation of the academic integrity of this course.
* You may use online resources to look up how to use a function, but you may not copy code from online resources. Any copied code (whether cited or not) is considered a violation of the academic integrity policy of this course.