Exercise M3

For this exercise, you will write a network file server program. Your program is expected to run on top of the TCP protocol. The program is a simple file server that makes a collection of files available for transmission to clients. When the server starts up, it needs to know the name of the directory that contains the collection of files. This information can be provided as a command-line argument. You can assume that the directory contains only regular files (that is, it does not contain any sub-directories). You can also assume that all the files are text files.

When a client connects to the server, the server first reads a one-line command from the client. The command can be the string "index". In this case, the server responds by sending a list of names of all the files that are available on the server. Or the command can be of the form "get <file>", where <file> is a file name. The server checks whether the requested file actually exists. If so, it first sends the word "ok" as a message to the client. Then it sends the contents of the file and closes the connection. Otherwise, it sends the word "error" to the client and closes the connection.

Submission Instructions

Create a zip file containing all of your source code files and submit via Canvas. Besides turning in the code, you must show your program running to the TA. The TA will ask you questions about it, and may want to see it executing in different configurations.

It is not enough to mindlessly type lines of code. You need to make sure you understand what each line of code is doing. You will be quizzed about it.

Rubric

**Exercise M3 Rubric**

| Exercise M3 Rubric | | |
| --- | --- | --- |
| **Criteria** | **Ratings** | **Pts** |
| This criterion is linked to a Learning OutcomeAllow for configurable folder location on the server side | |  |  | | --- | --- | | **2 pts**  **Full Marks** | **0 pts**  **No Marks** | | 2 pts |
| This criterion is linked to a Learning OutcomeCorrect output for all commands | |  |  | | --- | --- | | **11 pts**  **Full Marks** | **0 pts**  **No Marks** | | 11 pts |
| This criterion is linked to a Learning OutcomeImplement a client-server model (two programs need to be written, i.e., a client and a server) | |  |  | | --- | --- | | **2 pts**  **Full Marks** | **0 pts**  **No Marks** | | 2 pts |
| This criterion is linked to a Learning OutcomeUse TCP | |  |  | | --- | --- | | **5 pts**  **Full Marks** | **0 pts**  **No Marks** | | 5 pts |
| This criterion is linked to a Learning OutcomeAnswers about code/concepts sufficiently | |  |  | | --- | --- | | **5 pts**  **Full Marks** | **0 pts**  **No Marks** | | 5 pts |
| This criterion is linked to a Learning OutcomeImplement a multithreaded server | |  |  | | --- | --- | | **0 pts**  **Full Marks** | **0 pts**  **No Marks** | | 0 pts |
| Total Points: 25 | | |