Server Program – Download Server

In this assignment you will develop a network application, an anonymous download application. The server will be a concurrent, connection-oriented download server. The server should accept an optional port number as a command-line argument. The server will allow anonymous downloads; no username/password is required.

The client will be a download client to test the server. The client should include a menu or list of available commands for the user. The client should accept the server's hostname and optionally, the port number as a command-line argument.

You will also create a readme file, which should include the answers to the *Decisions* presented, along with the traditional contents of a readme file. The readme file will include, at a minimum, the following information:

- a description of the project
- how to compile
- how to run
- a brief design overview
- the protocol developed, including the syntax for the messages and how the client/server will know when it has received all the data from the other host.
- any known issues

The supported messages of this download protocol are:

- HELLO – sent by server upon client connection
- BYE – sent by client on disconnect
- PWD – current directory on the server
- DIR – directory listing on the server
- CD – change directory on the server
- DOWNLOAD – client downloads requested file

The download protocol specifications are:

- When client connects, the server will send a "HELLO" message.
- When client disconnects from server, client sends a "BYE" message to the server.
- Current directory
    - Client will send "PWD" command to the server
    - Server will respond with the absolute directory name, or an error
    - Decision? How will the client know it has received the entire directory name?
- Directory listing
    - Client will send "DIR" command to server
    - Server will respond with the filenames of each file in the current directory, or an error
    - Decision: How will the server indicate the end of the directory listing?

- Change directory on server
    - Client will send "CD" command, followed by a directory (absolute or relative)
    - Server will change to that directory, if possible
    - Server must handle errors, such as "directory doesn't exist" or "permission denied"
    - Server will send response to client
    - Decision: What will server response be for successful and unsuccessful cd commands?
- Download a file (send from server to client)
    - Client will send "DOWNLOAD" command to server, followed by a filename
    - Server will respond with one of the following:
        - "READY" if file exists
        - "File Not Found" if file doesn't exist
    - Client will respond with one of the following:
        - "READY" if file doesn't exist locally, or it exists locally and should be overwritten (must ask user if they want to overwrite file or not)
        - "STOP" if file exists and should not be overwritten
    - Server sends file to client, line by line (if receive READY from client)
    - Decision: How does the client know when the server has completed sending the file?

This will be a pair assignment. Identify a partner to work with ASAP. The split of duties will be:

- Person 1 – client implementation and readme file (both people will contribute information)
- Person 2 – server implementation

This will also be a two-phase project. The phases will be:

- Phase 1
    - Accept and validate command-line arguments
    - Create a connection from client to server
    - HELLO message
    - BYE message
- Phase 2
    - All specifications for the download application

Submit via turnin on acad (csit):

- Readme file
- Source files
- Makefile, if necessary