Introduction to Computer Networks

Lab1 (Deadline: will be announced on ilms)

1. Description

Implement a simple FTP model.

The server should create a FTP server and allow users (clients) to upload and download files.

The client should be able to access the FTP server and upload and download files.

Please use TCP socket and the programs should be implemented in C/C++.

2. Requirements

- Server program

File name of the server program: studentID_ser.c or studentID_ser.cpp

The server program is executed using the command line by typing "./studentID_ser port" where "port" is a port number. Functions of the server:

(1) Allow client to access the server.

(2) If client accesses the server successfully, server needs to show the client's IP address.

(3) If client upload file to server successfully, server needs to show the size of the file.

- Client program

File name of the client program: studentID_cli.c or studentID_cli.cpp

The client program is executed using the command line by typing "./studentID_cli ip port" where "ip" is the IP address of the server and "port" is the port number for the server. Functions of the client:

(1) Should be able to connect the server.

(2) Should be able to upload files to server.

(3) Should be able to download files from server.

(4) Should be able to list all the files on server.

Please create two folders named "upload" and "download" on the client side and a folder named "upload" on the server side.

On client side, please implement three command functions:

(1) put: If client types "put filename", the client program sends the file in the "./upload" directory to the directory "./upload" on the server.

(2) get: If client types "get filename", the client program gets the file from the directory ". /upload" on the server and send to the directory "./download" on the client.

(3) dir : If client types "dir", the client gets all the files' information of directory "./upload" on the server and show on client's screen.(Please refer to windows cmd "dir")

(4) rename: If client types "rename oldfilename newfilename", the client sends a request to the server and changes the specific filename on the server. **(extra points)**

Note:

(1) All the files must be transmitted via socket.

(2) When using TCP, please set the maximum size of each packet to 1024 bytes.

(3) If the size of the file is over 1024 bytes, you need to split the file and transmit packets individually.

(4) You need to handle illegal-command exception. (ex, Invalid filename, Invalid command)

3. Evaluation:

(80%) Programs

(20%) Report

File name for report: studentID_report.pdf

Your report should include:

1. Details of your implementation, including server-side and client-side.
2. Screenshot or the instruction of the executing process on each function.
3. Descriptions of difficulties you encounter and how you solve them.

(0%) Readme

File name for readme: studentID_readme.txt

Please write a readme file to show how to execute and run your program.

If you finish "rename" function, you will get extra 10%

4. Submission

(1) Please upload the following files to ilms.

- studentID_ser.c
- studentID_cli.c
- studentID_report.pdf
- studentID_readme.txt

(2) All the files should be included, otherwise no grade will be given for the programming lab.

(3) Deadline: will be announced on ilms.