CS230 Spring 2019

PA #5: File Transfer Protocol

Due: June 19, 2019 23:59

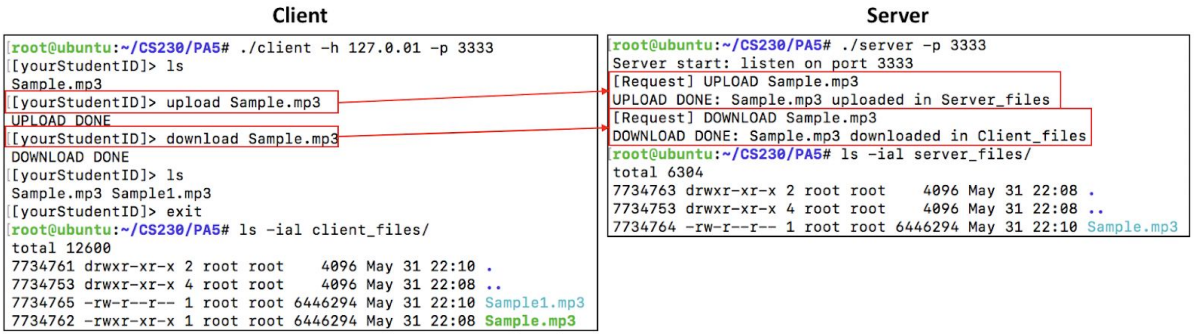
Introduction

- The goal of this task is to understand network socket programming.
- The File Transfer Protocol (FTP) is a standard network protocol used to transfer computer files between a client and server on a computer network.
- You are asked to implement a simple FTP server and a client in this assignment.

Problem Specification

- Our client application supports three functions: **LIST**, **UPLOAD**, and **DOWNLOAD**.
    - **LIST** function prints out the list of files in **Client_files** directory.
    - **UPLOAD** function uploads a file to the server. A file in **Client_files** is uploaded to **Server_files** directory.
    - **DOWNLOAD** function downloads a file from the server. A file in **Server_files** is downloaded to **Client_files**.
- Our FTP server listens on the specific port. After a client connects to the server, it requests to (1) upload a file to the server or (2) download a file from the server. The server handles these requests by (1) storing the file in **Server_files** or (2) sending the file in **Server_files** to the client.
- Note that exceptions need to be handled properly for the following cases.
    - Giving invalid IP address or port number
    - Giving invalid file name
    - In case name of the file downloaded from Server_files overlaps with the file in Client_files directory, download it with different name (e.g., “Sample.mp3” to “Sample(1).mp3”).

Output Sample



Execution and Submission Guidelines

- Unlike the previous programming assignments, Elice system is not used for this task. So, students are required to implement and run their codes in their own Linux machines. Using virtual machines such as VMWare or VirtualBox is also possible. Installation guidelines are covered in p.6 - 11 of Linux tutorial #1 (Week 4).
- The following are provided:
    - *Client_files (directory with a sample mp3 file)*
    - *Server_files (empty directory)*
    - *execute.sh*
    - *client.c*
    - *server.c*
- For compilation, run *execute.sh* (./execute.sh). If the source codes are successfully compiled, two executables ***client*** and ***server*** are generated.
- For execution,
    - Open one terminal and run the server executable with giving port number as the argument. (**./server -p [port number]**)
    - Open another terminal and run the client executable with giving IP address of the host (for evaluation, loopback IP address is used: 127.0.0.1) and the port number. (**./client -h [host IP address] -p [port number]**)
- For submission, compress all your files (listed above + **short report that describes your implementation**) into **PA5_yourStudentID.tar.gz** and upload it in KLMS.
- For evaluation, Ubuntu 64-bit will be used.
- If you copy codes from the internet or other students, you will get 0 points.