

## Sockets Programming Assignment

In this assignment you will develop a multi-threaded FTP server that is capable of processing multiple simultaneous service requests in parallel and an FTP client application to test the FTP server. FTP is defined by Internet Standard 9 and RFC 959.

We will not be implementing FTP following the complete standard, as we will not be using two ports for communication. We will be implementing FTP-Simple, which will use one port (socket) for all communication. The FTP-Simple is a client-server application using TCP stream sockets. The FTP-Simple protocol specifications are:

- Upload a file (send from client to server)
  - Client will send “SEND” command to server
  - Client will send either
    - Filename of file to upload
    - “File Not Found”
  - Server will respond with one of the following:
    - “READY” if file doesn’t exist or it exists and should be overwritten
    - “STOP” if file exists and should not be overwritten
  - Client sends file to server, line by line
  - Client sends “DONE” to server when entire file sent
- Download a file (send from server to client)
  - Client will send “GET” command to server
  - Client will send filename of file to download
  - 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 or it exists and should be overwritten
    - “STOP” if file exists and should not be overwritten
  - Server sends file to client server, line by line
  - Server sends “DONE” to client when entire file sent
- Disconnect
  - Client will send “DISCONNECT” to server
  - Server will close connection / end thread

The client should accept the server’s hostname and port number as a command-line argument.

The server should accept the port number as a command-line argument. The server will be able to handle multiple simultaneous service requests in parallel. This means that the FTP server is multi-threaded. In the main thread, the server listens to a fixed port. When it receives a TCP connection request, it sets up a TCP connection through another port and services the request in a separate thread. To simplify this programming task, skeleton programs will be provided.

You will be graded on documentation, format and correctness.

To begin, copy the APPROPRIATE file from the csc311 directory in my public directory (/export/home/public/frye/csc311):

- Java
  - ftpServer.java – server program
  - ftpRequest.java – additional functions for server program
  - ftpClient.java – client program
- C++
  - ftpserver.cpp – server program
  - ftpclient.cpp – client program
  - makefile – makefile for both programs
  - utils.c – some utility functions that may be helpful

You will need to complete ONE of the following files: ftpRequest.java / ftpserver.cpp OR ftpClient.java / ftpclient.cpp. If you modify and/or turn in the wrong file or both files, you will receive a 0 for the assignment. If there is any evidence of plagiarism, you will receive a 0 for the assignment.

There are copies of a java and a C++ webserver in the csc311 public directory for your reference.

### Turning in Assignment

Turn the program in via turnin. You must first setup turnin for CSC311. To do this, follow these steps:

1. /export/home/public/frye/turnin.pl 311
2. If you use the bash shell, run the script (from professor's public directory) turnin.bash.sh  
/export/home/public/frye/turnin.bash.sh
3. Either logout or do a "source .login" from your home directory before submitting the assignment.

To submit the assignment, run  
turnin311 <filename>

Here is the file you are assigned to modify to meet the requirements of the FTP-Simple protocol described above:

- Section 010 (8:00am)
  - If your last name begins with A through P – modify ftpRequest.java
  - If your last name begins with R through Z – modify ftpClient.java
- Section 020 (9:30am)
  - If your last name begins with A through Kel – modify ftpRequest.java
  - If your last name begins with Ker through Z – modify ftpClient.java