CS 39006: Networks Lab
Assignment 8: Implement a FTP Server/Client using Stream Socket (C/C++)

Date: 04th April, 2017

Assignment Statement:

Write two separate programs, client.c and server.c, to implement a simplified version of FTP. The server process should work in connection-oriented and concurrent-server mode. You need to first start the server process in a server host and publish its host name and port number. A user on another host can then issue a command like;

```
%myftp server-host-name server-port-number
```

to download a file from or upload a file to the server. After accepting the above `myftp`, command, the client process should responds with prompt

```
myftp>
```

waiting for user's FTP commands.

Requirements:

1) myftp is so named as to tell from the standard ftp command.
2) You need to consider the following ftp commands:
   (a) `myftp>fput` filename to upload a file named filename to the server,
   (b) `myftp>fget` filename to download a file named filename from the server,
   (c) `myftp>servls` to list the files under the present directory of the server,
   (d) `myftp>servcd` to change the present working directory of the server,
   (e) `myftp>servpwd` to display the present working directory of the server
   (f) `myftp>clils` to list the files under the present directory of the client,
   (g) `myftp>clicd` to change the present directory of the client,
   (h) `myftp>clipwd` to display the present working directory of the client,
   (i) `myftp>quit` to quit from ftp session and return to Unix prompt.
3) Other commands except the above ones are considered as invalid ftp commands. When a user inputs an invalid ftp command, your program should respond with "An invalid FTP Command."
4) When put or get a non-existed file, your program should respond with " filename: no such file."

5) Note that FTP works on two connections – one connection is used for the control message transfer (fput, fget, ls etc.), and another connection is used for the actual file transfer. Therefore your server should handle two different stream sockets for control connection and data connection.

6) Multiple clients can connect to the server simultaneously.

7) Your program should be able to handle any file – text, PDF, video, audio etc.

8) You can use `system()` function call in your program.

Submission Instructions:

You need to submit relevant source files in a single compressed (tar.gz) file. Rename the compressed file as Assignment_8_Roll1_Roll2.tar.gz, where Roll1 and Roll2 are the roll numbers of the two members in the group. Submit the compressed file through Moodle by the submission deadline. The submission deadline is: **April 11, 2017 02:00 PM.**

Please note that your submission will be awarded zero marks without further consideration, if it is find to be copied. In such cases, all the submissions will be treated equally, without any discrimination to figure out who has copied from whom.