Faraon, Conard James B.

CSS 432 Computer Networking

20 May 2017

Final Project: FTP Client

Algorithm Documentation

OPEN, PASS/SYST, and QUIT

The command “open” establishes a connection with the server, and only tries to log in the user if the FTP client is started with a server or port arguments such as the following:

```
"usage: ./ftp <hostname>" && "usage: ./ftp <hostname> <port>"
```

When command open is invoked, the FTP client connects to the server using the IP name through a socket. If the FTP client was not started with any argument, the user must manually login prior to requesting services from the server. Figure 1 illustrates the algorithm used for a manual login.

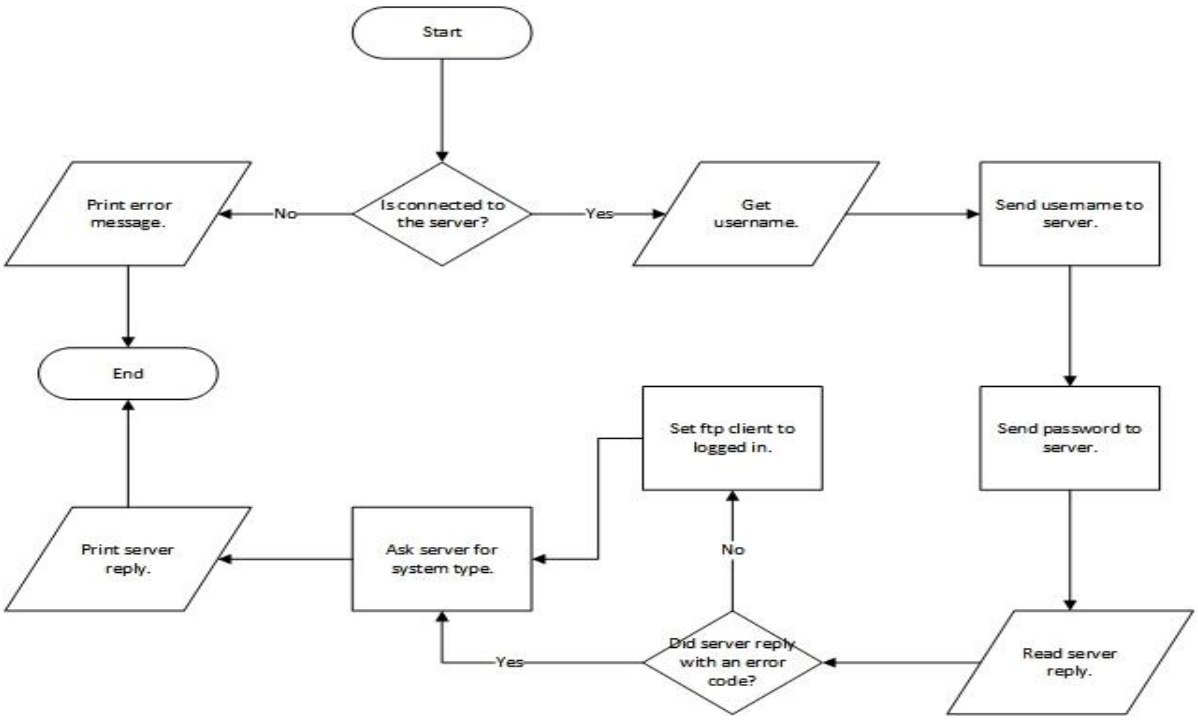


Figure 1. Shows the login process using OPEN, USER, and PASS.

The FTP client program has commands “close” and “quit” (or “exit) for disconnecting the connection with the server and exiting the program respectively. If the “quit” or “exit” command is invoked then the FTP client program disconnects only if the program is still connected to the server, and then gracefully terminates the program using the UNIX `exit()` command. The command close only disconnects the program from the server. Figure 2 illustrates the algorithm used for the command close.

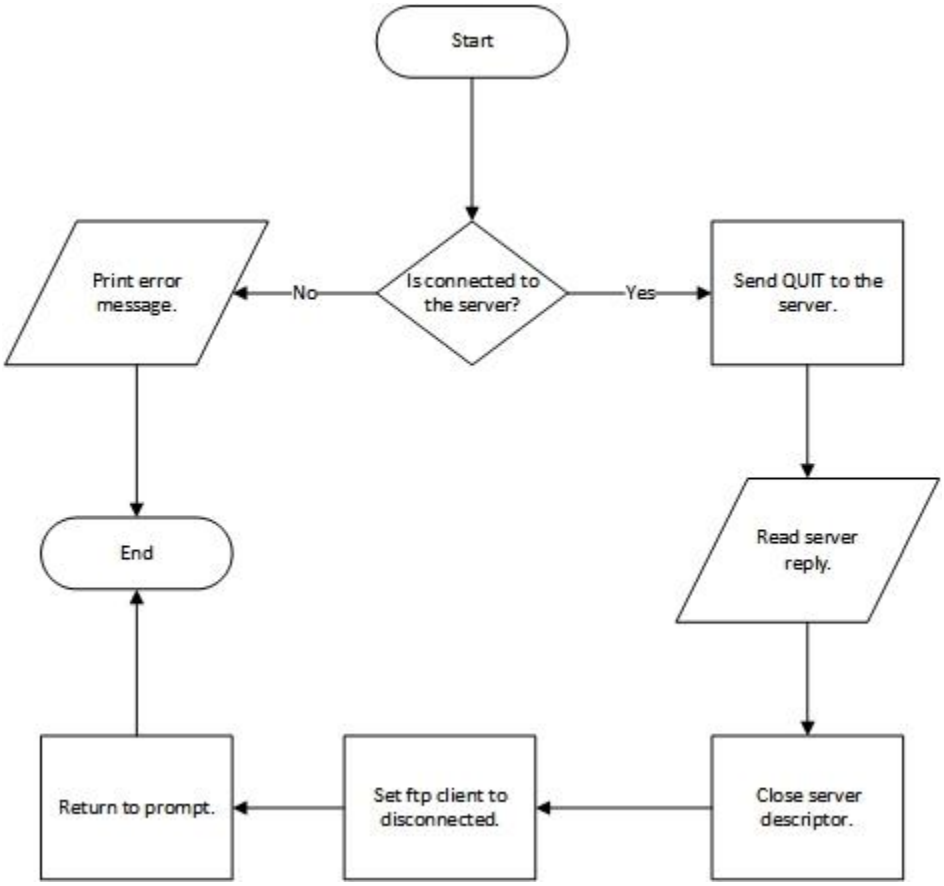


Figure 2. Shows the process of closing the connection with the server using “close.”

PASV, RETR

The FTP client program modularizes a function for enabling passive mode. The client sends a “PASV” command through the control channel, and then process the reply from the server to

tokenize the numbers needed to calculate the port for the server's data channel such as the following where MULTIPLER is equivalent to 256:

```
int dataPort = (num1 * MULTIPLIER) + num2;
```

The "dataPort" value is used to construct a socket and to establish connection with the server's data channel. After enabling passive mode, the client can issue the commands "TYPE I" and "RETR <filename>" to the server. If the file that we are trying to retrieve exists within the server, then we fork a child to download the file from the server. Figure 3 illustrates how the command "get" works.

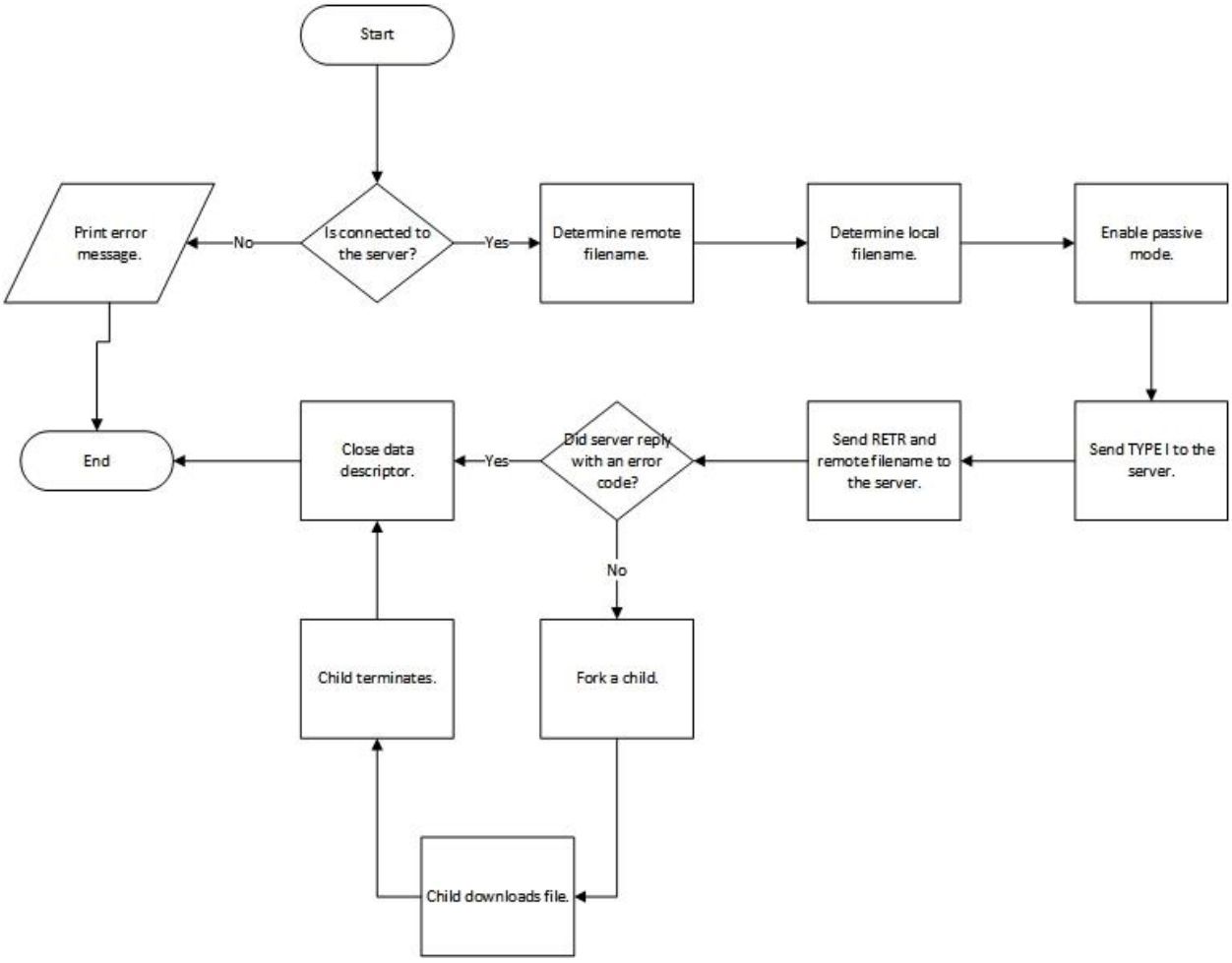


Figure 3. Shows how the FTP client retrieves a file from the server.

CWD

The command “cd” or changing the working directory is pretty straightforward. The FTP client program sends the command “CWD <dirname>” to the server, and retrieves the reply from the server. Figure 4 illustrates how command “cd” works.

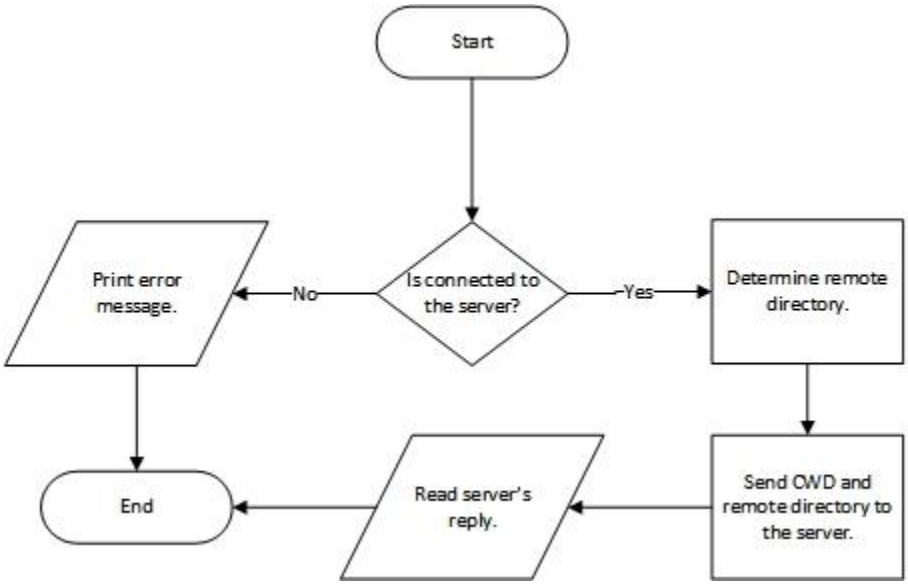


Figure 4. Shows how the FTP client changes directory.

PASV, LIST

The command “ls” lists the contents of the current working directory. This is done by enabling passive mode, then reading the standard output from the data channel. The implementation requires for the parent to wait for the child to terminate prior to reading the last reply from the server. Figure 5 illustrates how command “ls” works.

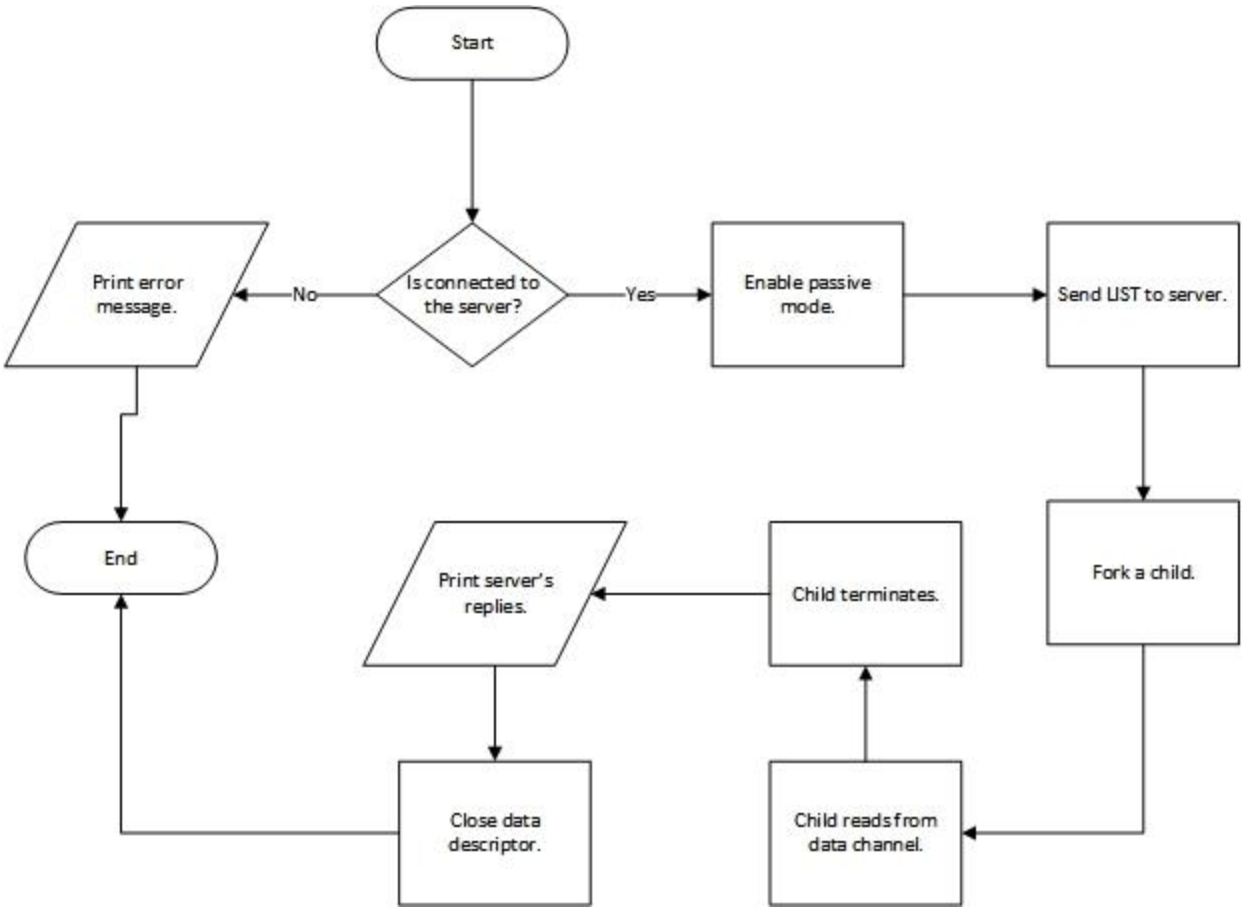


Figure 5. Shows how "ls" uses a child process to read from the data channel.

PASV, STOR

The command "put" uses passive mode and follows the reverse algorithm of the command "get." A "FIN" flag is sent to the data channel after the last chunk of data is sent to the data channel. Additionally, the "put" command only opens the file in read only mode to prevent the losing or corrupting the file. Figure 6 illustrates how command "put" works.

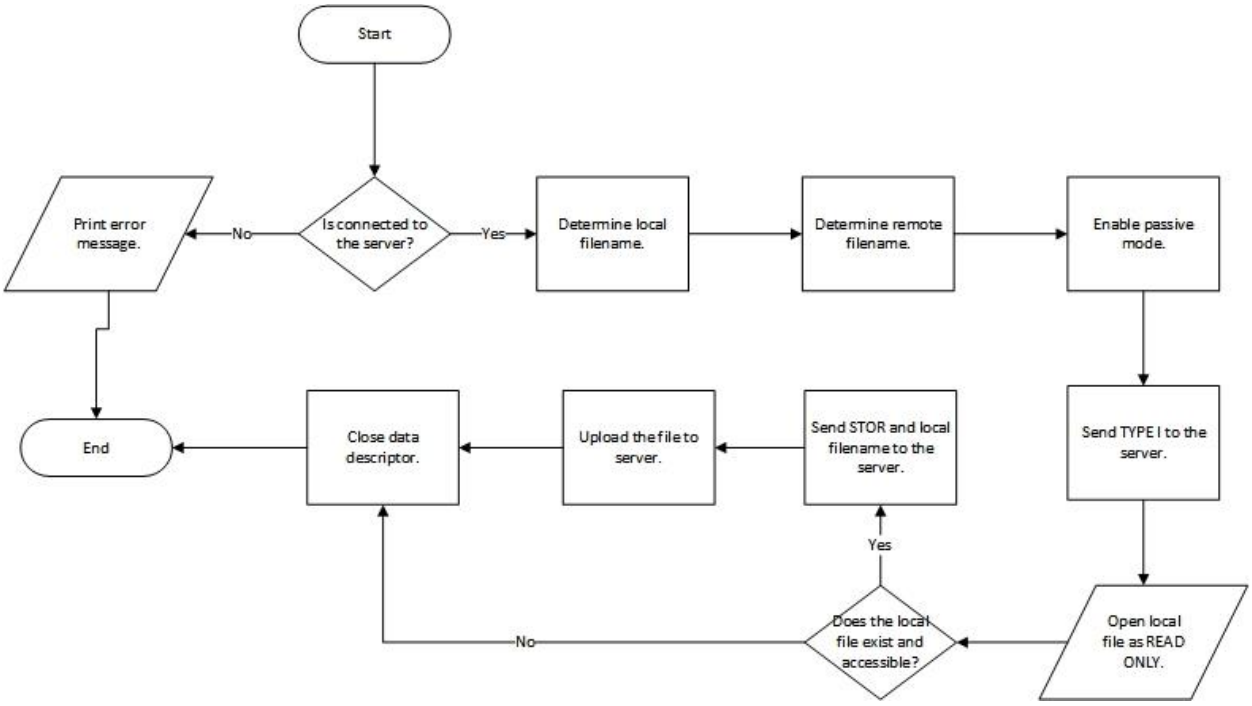


Figure 6. Shows how command “put” verifies if a local file is accessible prior to uploading.

EXTRA CREDIT: PWD, DEL

In addition to required commands for the assignment, the commands “pwd” and “del <remote filename>” are implemented. The command “pwd” is implemented as follows:

```
void FTP::pwd()
{
    if (!isConnected())
    {
        cerr << NOT_CONNECT << endl;
        return;
    }

    string reply;
    unsigned long length;
    //send pwd command to the server
    char *command = input.toCharArr(PWD, length);
    int replyCode;
    if (sendServerCommand(command, length) > 0)
    {
        //read server's reply
        if (receiveServerReply(reply) > 0)
```

```
        {
            replyCode = getReplyCode(reply);
            cout << reply;

            //server crashes
            if (replyCode == 421)
            {
                close(serverDescriptor);
                isOnline = false;
                loggedIn = false;
            }
        }
    }
}
void FTP::pwd()
{
    if (!isConnected())
    {
        cerr << NOT_CONNECT << endl;
        return;
    }

    string reply;
    unsigned long length;
    //send pwd command to the server
    char *command = input.toCharArr(PWD, length);
    int replyCode;
    if (sendServerCommand(command, length) > 0)
    {
        //read server's reply
        if (receiveServerReply(reply) > 0)
        {
            replyCode = getReplyCode(reply);
            cout << reply;

            //server crashes
            if (replyCode == 421)
            {
                close(serverDescriptor);
                isOnline = false;
                loggedIn = false;
            }
        }
    }
}
```

Lastly, the command “del” is comparable to the command “cd.” This is also implemented as follows:

```
void FTP::del(const string in)
{
    if (!isConnected())
    {
        cerr << NOT_CONNECT << endl;
        return;
    }

    string remotefilename = in;

    if (remotefilename == "")
    {
        cout << "(remote-file) ";
        getline(cin, remotefilename);
    }

    if (remotefilename == "")
    {
        cerr << "usage: del remote-file" << endl;
        return;
    }

    string reply;
    unsigned long length;
    char *command = input.toCharArr(DELE, remotefilename, length);
    int replyCode;
    //send delete command to the server
    if (sendServerCommand(command, length) > 0)
    {
        if (receiveServerReply(reply) > 0)
        {
            replyCode = getReplyCode(reply);
            cout << reply;

            //server crashes
            if (replyCode == 421)
            {
                close(serverDescriptor);
                isOnline = false;
                loggedIn = false;
            }
        }
    }
}
```

Limitations

Due to the time constraint and the size of the original UNIX FTP client program, some of the commands were not implemented. Although error code 421 (server crashes and server-initiated disconnection) is handled in the student's FTP client program, handling for other errors codes 400's and 500's were not implemented.

The student attempted to mimic the original UNIX FTP client program, and the user of the student's program should see and should feel similarities with the original program.

Additionally, a known bug exists for which the server's data channel is stuck open after an error 550 happens, which is a file not found error after the client issued the command "get" for an invalid file. This bug is common and this was discovered by the CSS 432 students from last quarter. A fix has not been determined yet. **The professor and student attempted to debug the issue, but they failed to resolve the bug. The student is still awaiting for guidance from the professor. It is assumed that this bug is with the server side.**

The bug is shown as follows:

```
ftp> get noSuchFile
227 Entering Passive Mode (209,202,252,54,199,109)
200 Type set to 'I' (IMAGE aka BINARY).
550 No such file as 'noSuchFile'.
ftp server: Invalid file format or file does not exist.
225 Data connection open; no transfer in progress.
ftp> ls
227 Entering Passive Mode (209,202,252,54,199,187)
150 Opening ASCII mode data connection for LIST.
226 Transfer complete.
ftp>
```

As you can observe, the data channel was successfully established but no data is being read through the data channel.

The following are some of the attempts done by the student to solve the bug if error 550 happens:

- Kill the child process.
- Send a FIN flag using the `shutdown()` command.
- Send the command "ABOR" (abort) to the control or data channel (or both).
- Poll multiple times.

Implementing a function "`flush()`" can temporarily fix the bug by enabling the passive mode and issuing the command "LIST" to the server and NOT print any output after an error 550 happened. However, the student finds this fix inefficient and improper, and the student would rather report the bug in order to resolve the issue.

Execution Output

Connection Establishment

The following are some of the sample ways to establish a connection and login to the server:

```
faraonc@uw1-320-00:~/ftpTest$ ./ftp ftp.tripod.com
220 Welcome to Tripod us FTP.
Name (ftp.tripod.com:faraonc): css432s17
331 Username set to css432s17. Now enter your password.
Password: *********
230- ============================================================
230-                      IMPORTANT NOTICE
230- ============================================================
230-
230- Powerful building tools. Traffic-generating, money-making
230- programs. It's all waiting for you at Tripod.
230-
230-   http://www.tripod.lycos.com/
230-
230- ============================================================
230-
230- Got a great idea for a website?  Then don't
230- wait, get your own web address today!
230-
230-   http://www.tripod.lycos.com/domains/
230-
230- ============================================================
230-
230- We heard you loud and clear! You love your site, but
230- you don't like the ads.  Remove those pesky popups
230- forever!
230-
230-   http://www.tripod.lycos.com/web-hosting/compare_plans.pl
230-
230- ============================================================
230 User 'css432s17' logged on.
215 UNIX Type: L8
ftp>
```

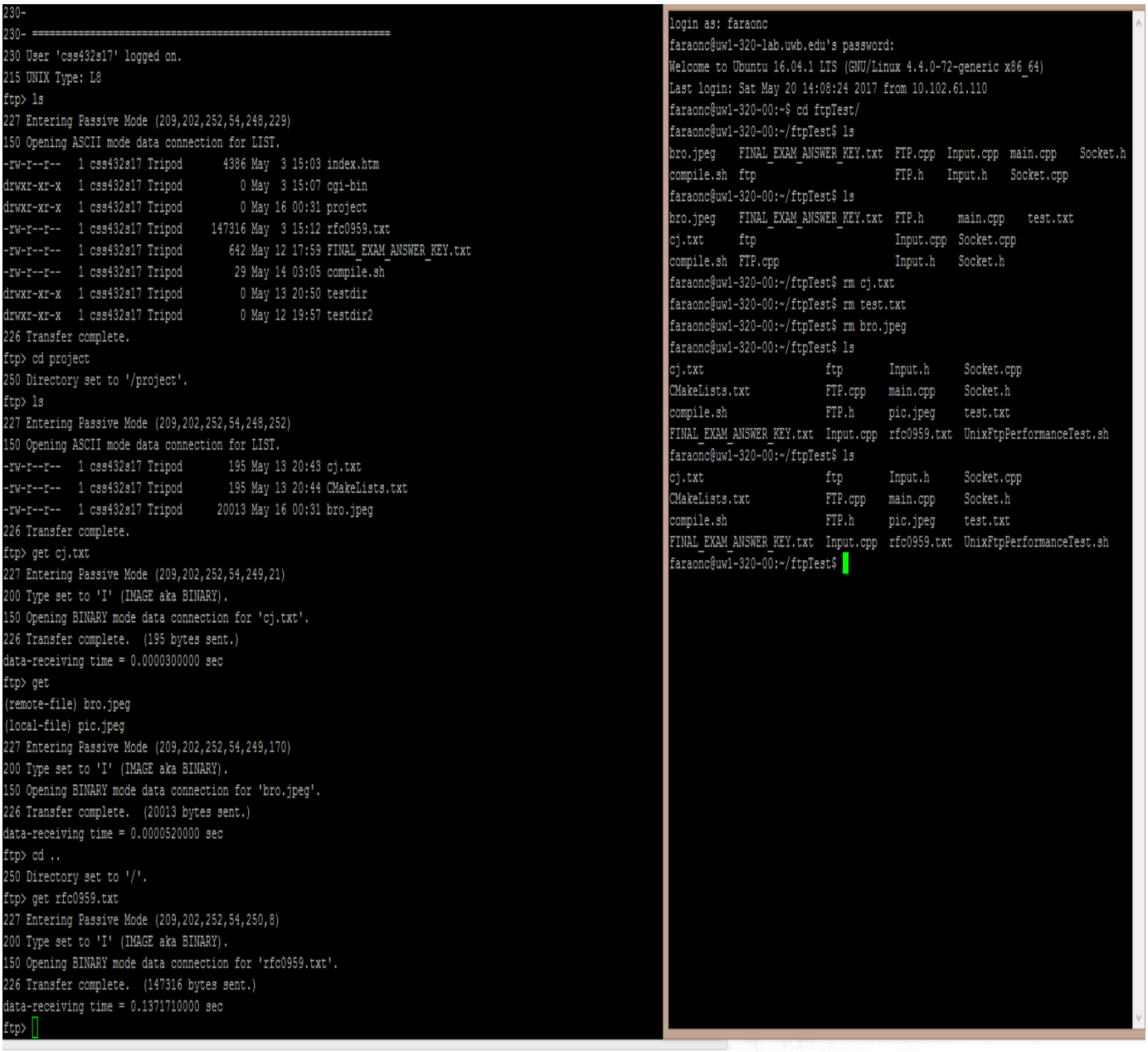
```
faraonc@uw1-320-00:~/ftpTest$ ./ftp ftp.tripod.com 21
220 Welcome to Tripod us FTP.
Name (ftp.tripod.com:faraonc): css432s17
331 Username set to css432s17. Now enter your password.
Password: *********
230- ================================================================
230-                       IMPORTANT NOTICE
230- ================================================================
230-
230- Powerful building tools. Traffic-generating, money-making
230- programs. It's all waiting for you at Tripod.
230-
230-   http://www.tripod.lycos.com/
230-
230- ================================================================
230-
230- Got a great idea for a website?  Then don't
230- wait, get your own web address today!
230-
230-   http://www.tripod.lycos.com/domains/
230-
230- ================================================================
230-
230- We heard you loud and clear! You love your site, but
230- you don't like the ads.  Remove those pesky popups
230- forever!
230-
230-   http://www.tripod.lycos.com/web-hosting/compare_plans.pl
230-
230- ================================================================
230 User 'css432s17' logged on.
215 UNIX Type: L8
ftp>
```

```
faraonc@uw1-320-00:~/ftpTest$ ./ftp
ftp> open ftp.tripod.com
220 Welcome to Tripod us FTP.
Name (ftp.tripod.com:faraonc): css432s17
331 Username set to css432s17. Now enter your password.
Password: *********
230- ================================================================
230-                      IMPORTANT NOTICE
230- ================================================================
230-
230- Powerful building tools. Traffic-generating, money-making
230- programs. It's all waiting for you at Tripod.
230-
230-   http://www.tripod.lycos.com/
230-
230- ================================================================
230-
230- Got a great idea for a website?  Then don't
230- wait, get your own web address today!
230-
230-   http://www.tripod.lycos.com/domains/
230-
230- ================================================================
230-
230- We heard you loud and clear! You love your site, but
230- you don't like the ads.  Remove those pesky popups
230- forever!
230-
230-   http://www.tripod.lycos.com/web-hosting/compare_plans.pl
230-
230- ================================================================
230 User 'css432s17' logged on.
215 UNIX Type: L8
ftp> close
221 Goodbye...
```

```
ftp> open ftp.tripod.com
220 Welcome to Tripod us FTP.
Name (ftp.tripod.com:faraonc): a
331 Username set to a. Now enter your password.
Password: *
501  Please verify your Password
Login failed.
215 UNIX Type: L8
ftp> user css432s17
331 Username set to css432s17. Now enter your password.
Password: *********
230- ================================================================
230-                      IMPORTANT NOTICE
230- ================================================================
230-
230- Powerful building tools. Traffic-generating, money-making
230- programs. It's all waiting for you at Tripod.
230-
230-   http://www.tripod.lycos.com/
230-
230- ================================================================
230-
230- Got a great idea for a website?  Then don't
230- wait, get your own web address today!
230-
230-   http://www.tripod.lycos.com/domains/
230-
230- ================================================================
230-
230- We heard you loud and clear! You love your site, but
230- you don't like the ads.  Remove those pesky popups
230- forever!
230-
230-   http://www.tripod.lycos.com/web-hosting/compare_plans.pl
230-
230- ================================================================
230 User 'css432s17' logged on.
215 UNIX Type: L8
ftp>
```

Get Function

The following is a snapshot for the get function:



CD, LS & PWD Functions

The following is snapshot for cd, ls, and pwd functions:

```
ftp> cd project
250 Directory set to '/project'.
ftp> ls
227 Entering Passive Mode (209,202,252,54,253,154)
150 Opening ASCII mode data connection for LIST.
-rw-r--r--   1 css432s17 Tripod        195 May 13 20:43 cj.txt
-rw-r--r--   1 css432s17 Tripod        195 May 13 20:44 CMakeLists.txt
-rw-r--r--   1 css432s17 Tripod      20013 May 16 00:31 bro.jpeg
226 Transfer complete.
ftp> cd ..
250 Directory set to '/'.
ftp> ls
227 Entering Passive Mode (209,202,252,54,253,170)
150 Opening ASCII mode data connection for LIST.
-rw-r--r--   1 css432s17 Tripod       4386 May  3 15:03 index.htm
drwxr-xr-x   1 css432s17 Tripod          0 May  3 15:07 cgi-bin
drwxr-xr-x   1 css432s17 Tripod          0 May 16 00:31 project
-rw-r--r--   1 css432s17 Tripod     147316 May  3 15:12 rfc0959.txt
-rw-r--r--   1 css432s17 Tripod        642 May 12 17:59 FINAL_EXAM_ANSWER_KEY.txt
-rw-r--r--   1 css432s17 Tripod         29 May 14 03:05 compile.sh
drwxr-xr-x   1 css432s17 Tripod          0 May 13 20:50 testdir
drwxr-xr-x   1 css432s17 Tripod          0 May 12 19:57 testdir2
226 Transfer complete.
ftp> pwd
257 "/" is current directory.
ftp> cd project
250 Directory set to '/project'.
ftp> pwd
257 "/project" is current directory.
ftp> cd ..
250 Directory set to '/'.
ftp> cd testdir
250 Directory set to '/testdir'.
ftp> pwd
257 "/testdir" is current directory.
ftp>
```

PUT & DEL Functions

The following is snapshot for put and del functions:

```
ftp> put compile.sh
227 Entering Passive Mode (209,202,252,54,4,60)
200 Type set to 'I' (IMAGE aka BINARY).
150 Opening BINARY mode data connection for 'compile.sh'.
226 Transfer complete.  (29 bytes sent.)
ftp> ls
227 Entering Passive Mode (209,202,252,54,4,90)
150 Opening ASCII mode data connection for LIST.
-rw-r--r--   1 css432s17 Tripod        195 May 13 20:43 cj.txt
-rw-r--r--   1 css432s17 Tripod        195 May 13 20:44 CMakeLists.txt
-rw-r--r--   1 css432s17 Tripod      20013 May 16 00:31 bro.jpeg
-rw-r--r--   1 css432s17 Tripod         29 May 20 18:03 compile.sh
226 Transfer complete.
ftp> del compile.sh
200 File compile.sh deleted. (1073468002 bytes now available)
ftp> ls
227 Entering Passive Mode (209,202,252,54,4,119)
150 Opening ASCII mode data connection for LIST.
-rw-r--r--   1 css432s17 Tripod        195 May 13 20:43 cj.txt
-rw-r--r--   1 css432s17 Tripod        195 May 13 20:44 CMakeLists.txt
-rw-r--r--   1 css432s17 Tripod      20013 May 16 00:31 bro.jpeg
226 Transfer complete.
ftp> put
(local-file) compile.sh
(remote-file)
227 Entering Passive Mode (209,202,252,54,4,151)
200 Type set to 'I' (IMAGE aka BINARY).
150 Opening BINARY mode data connection for 'compile.sh'.
226 Transfer complete.  (29 bytes sent.)
ftp> ls
227 Entering Passive Mode (209,202,252,54,4,179)
150 Opening ASCII mode data connection for LIST.
-rw-r--r--   1 css432s17 Tripod        195 May 13 20:43 cj.txt
-rw-r--r--   1 css432s17 Tripod        195 May 13 20:44 CMakeLists.txt
-rw-r--r--   1 css432s17 Tripod      20013 May 16 00:31 bro.jpeg
-rw-r--r--   1 css432s17 Tripod         29 May 20 18:04 compile.sh
226 Transfer complete.
ftp> put compile.sh
227 Entering Passive Mode (209,202,252,54,4,204)
200 Type set to 'I' (IMAGE aka BINARY).
150 Opening BINARY mode data connection for 'compile.sh'.
226 Transfer complete.  (29 bytes sent.)
ftp> put
(local-file) compile.sh
(remote-file) compile1.sh
227 Entering Passive Mode (209,202,252,54,5,14)
200 Type set to 'I' (IMAGE aka BINARY).
150 Opening BINARY mode data connection for 'compile1.sh'.
ls226 Transfer complete.  (29 bytes sent.)
ftp>
227 Entering Passive Mode (209,202,252,54,5,37)
150 Opening ASCII mode data connection for LIST.
-rw-r--r--   1 css432s17 Tripod        195 May 13 20:43 cj.txt
-rw-r--r--   1 css432s17 Tripod        195 May 13 20:44 CMakeLists.txt
-rw-r--r--   1 css432s17 Tripod      20013 May 16 00:31 bro.jpeg
-rw-r--r--   1 css432s17 Tripod         29 May 20 18:04 compile.sh
-rw-r--r--   1 css432s17 Tripod         29 May 20 18:04 compile1.sh
226 Transfer complete.
ftp>
```

CLOSE, QUIT & EXIT Functions

The following is a snapshot of close, quit and exit functions:

```
ftp> del compile.sh
200 File compile.sh deleted. (1073447960 bytes now available)
ftp> ls
227 Entering Passive Mode (209,202,252,54,255,253)
150 Opening ASCII mode data connection for LIST.
-rw-r--r--   1 css432s17 Tripod       4386 May  3 15:03 index.htm
drwxr-xr-x   1 css432s17 Tripod          0 May  3 15:07 cgi-bin
drwxr-xr-x   1 css432s17 Tripod          0 May 20 18:10 project
-rw-r--r--   1 css432s17 Tripod     147316 May  3 15:12 rfc0959.txt
-rw-r--r--   1 css432s17 Tripod        642 May 12 17:59 FINAL_EXAM_ANSWER_KEY.txt
drwxr-xr-x   1 css432s17 Tripod          0 May 13 20:50 testdir
drwxr-xr-x   1 css432s17 Tripod          0 May 12 19:57 testdir2
226 Transfer complete.
ftp> close
221 Goodbye...
ftp> exit
================================================================
FTP Program Exit!
================================================================
faraonc@uw1-320-00:~/ftpTest$
```

```
faraonc@uw1-320-00:~/lamp$ ./ftp ftp.tripod.com
220 Welcome to Tripod us FTP.
Name (ftp.tripod.com:faraonc): css432s17
331 Username set to css432s17. Now enter your password.
Password: *********
230- ============================================================
230-                       IMPORTANT NOTICE
230- ============================================================
230-
230- Powerful building tools. Traffic-generating, money-making
230- programs. It's all waiting for you at Tripod.
230-
230-   http://www.tripod.lycos.com/
230-
230- ============================================================
230-
230- Got a great idea for a website?  Then don't
230- wait, get your own web address today!
230-
230-   http://www.tripod.lycos.com/domains/
230-
230- ============================================================
230-
230- We heard you loud and clear! You love your site, but
230- you don't like the ads.  Remove those pesky popups
230- forever!
230-
230-   http://www.tripod.lycos.com/web-hosting/compare_plans.pl
230-
230- ============================================================
230 User 'css432s17' logged on.
215 UNIX Type: L8
ftp> quit
221 Goodbye...
================================================================
FTP Program Exit!
================================================================
faraonc@uw1-320-00:~/lamp$
```

Performance Evaluation

Based on Table 1, we can observe that the UNIX FTP client performs better with small and medium size files. The assumption is that the Unix Ubuntu FTP client program was implemented back in the 90's (see Reference), and handling such a large file may be difficult. Additionally, The UNIX FTP client program is more complete than the student's ftp client program, therefore the student is not handling other edge cases that the UNIX implementations have such as other error codes and undefined behaviors not known to the student.

Table 1. Shows the performance comparison between the student's and the UNIX FTP client.

FTP Client	Small (29 Bytes)	Medium (20013 Bytes)	Large (147316 Bytes)
Student's	0.007615 sec	0.06 sec	0.1343 sec
UNIX	0.00439 sec	0.049 sec	0.69 sec

Discussion

Although the commands "del" and "pwd" are also implemented, the student's FTP client program only has limited commands implemented. Commands such as "rmdir," "mkdir," "mget," and commands for mode switching can be useful if these commands are also implemented.

Several error codes (400's and 500's) are not handled with the current implementations. Not handling these error codes may cause bugs and undefined behaviors. These error codes must be handled to ensure robustness of the program if the student is to extend the FTP client by adding more features.

The student's FTP client also inefficiently use C-string (char*) to string conversion or vice-versa. This may introduce unnecessary overhead and performance reduction.

Although the student's FTP client performed faster when downloading a large file than the UNIX FTP client, the UNIX FTP client may provide better security and error-handling than the student's FTP client.

Lastly, the FTP client may be extended further by making a graphical user interface or GUI to make the program more user friendly while separating the model, view, and controller. This can provide novice users the flexibility to use the FTP client program without the command line terminal.

Reference:

Index of /ubuntu/pool/main/n/netkit-ftp. N.p., n.d. Web. 20 May 2017.