Computer Networking
Assignment 2 report

5120309491 Chen Shangjie

1 Description & Runtime environment

I implement the project in C Language and the whole system is working on Linux System. User can easily use my ftp client and server by command-line operations. The interface for both local ftp server and the real ftp server are the same, i.e, you can access to the real ftp server freely, as well.

2 Build & Run

You may use 'cd' to move to the prj directory and type 'make' to build the whole prj. The executable will be located at `bin/client` and `bin/server`.

!!! Make sure you have already create 'bin/server', 'bin/client', and 'obj' those three directory in the current directory.

Use `./S <port number>` and `./C <host name> <port number>` to run the programm.

3 Command and detailed description

3.1 Commands you may use:

- USER < *username* > (`not necessary for local`)
- PASS < *password* > (`not necessary for local`)
- PASV (`to enter the passive mode for transmission`)
- NLST (`list all the files and directory`)

- DELE < $filename$ > (`delete the file`)
- RETR < $filename$ > (`download file from the ftp`)
- STOR < $filename$ > (`upload file to the ftp`)
- CWD < $dirname$ > (`change the current directory`)
- BYE (`change the current directory`)

Other commands will not be received by the server, the server will response 500.

3.2 Implementation on local server

NLST:

Invoke $system()$ the system call ls and load the result to a tmp file, read the file and then unlink the tmp file.

DELE:

Invoke $remove()$ to remove the file directly.

CWD:

Invoke $chdir()$ to change the current directory. if the return value equals to 0, invoke $getcwd()$ to get the current directory.

RETR & STOR:

According to the FTP protocol, when there need a data transmission, we need to create another data socket to transfer file. The user should send the 'PASV' command first to change the mode, then receive the new port number provided by the server, the client will connect to the new port and do the transfer. At this time, use $fread()$ and $fwrite()$ to do the file operation, continue write data which received from the data socket to the file. The detailed code will not be covered in this report.

3.3 FTP protocol related

3.3.1 FTP reply codes

Below is a summary of FTP reply codes that may be returned by an FTP server. The first digit is used to indicate one of three possible outcomes —success, failure, or to indicate an error or incomplete reply:

- 2yz - Success reply
- 4yz or 5yz - Failure reply
- 1yz or 3yz - Error or Incomplete reply

The second digit defines the kind of error:

- x0z - Syntax. These replies refer to syntax errors.
- x1z - Information. Replies to requests for information.
- x2z - Connections. Replies referring to the control and data connections.
- x3z - Authentication and accounting. Replies for the login process and accounting procedures.
- x4z - Not defined.
- x5z - File system. These replies relay status codes from the server file system.

The third digit of the reply code is used to provide additional detail for each of the categories defined by the second digit.

3.3.2 About commands

Each command must be ended up with `\r\n` (no bother to type it the program will solve it), the specific commands can be found in the internet...

4 Socket Programming

The main process of the socket connection is showed in Figure 1.

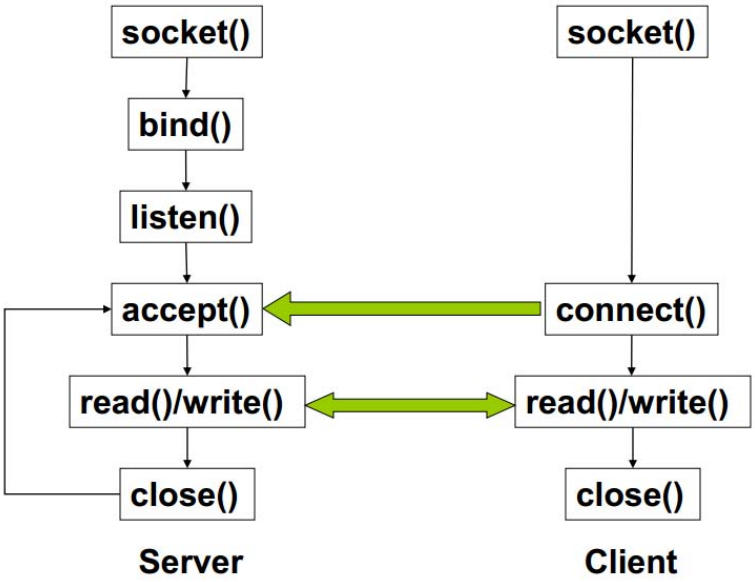

Figure 1: Socket

5 Trouble shooting & experience

5.1 About the receive command

There are two types of received command from the FTP server, one of them begin with number and space, such as '220 ', and then some words, finished with `\r\n`. The other begin with number and '-', such as '220-' ,which means there still exist some more information to transfer to the client.

In order to analyze the received command, I put all the received message into one buffer, use *strtok*() split it into pieces by `'\n'`. Judge every piece of command whether the fourth char of it is space, if true, end of *recv*().

The detailed implementation is in function *parse*() within C.c .

5.2 About the data port

When the client send the PASV to the server, the server will send the new data port. The return information will be like this:

227 Entering Passive Mode (h1,h2,h3,h4,p1,p2)

The first four number are the IP address, then the port number. We can get the port number by $p1 * 256 + p2$.

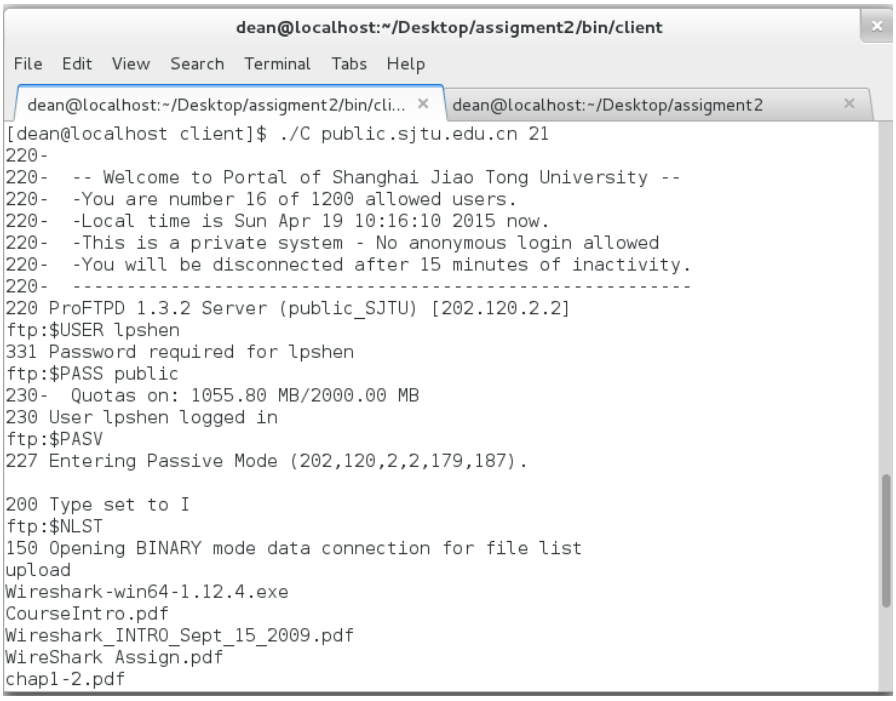

Figure 2: Connect with FTP server (1)

5.3 About the type

Some kind of file may not be correctly download, we need to change the type of FTP to binary, so that the format won't be a problem so that I encapsulate the TYPE command in the PASV command.

5.4 Some Comments

Working on this assignment is challenging though interesting, we can work on real FTP protocol which helps us understanding the SOCKET and FTP better. After all the job is done, I really gains a lot.

Thanks for Prof.Shen and TA's work, thanks for my classmate's help.

6 Screenshot

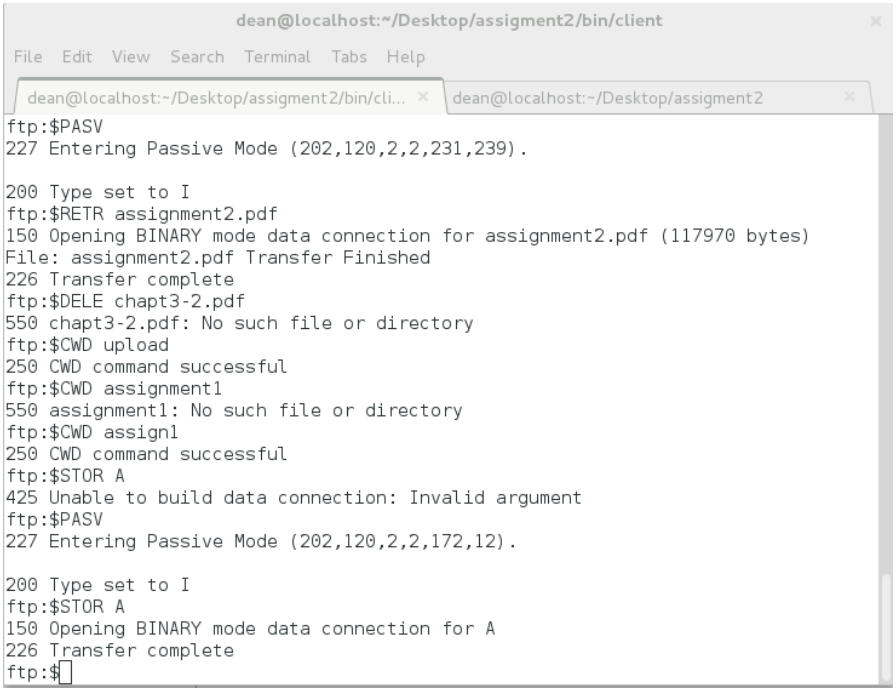

Figure 3: Connect with FTP server (2)

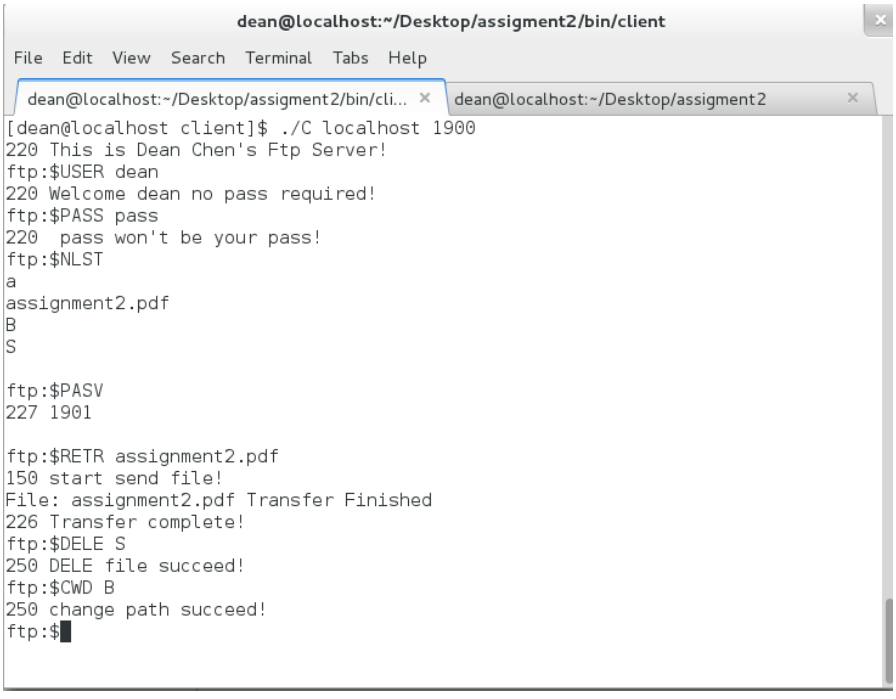

Figure 4: Connect with local server (1)

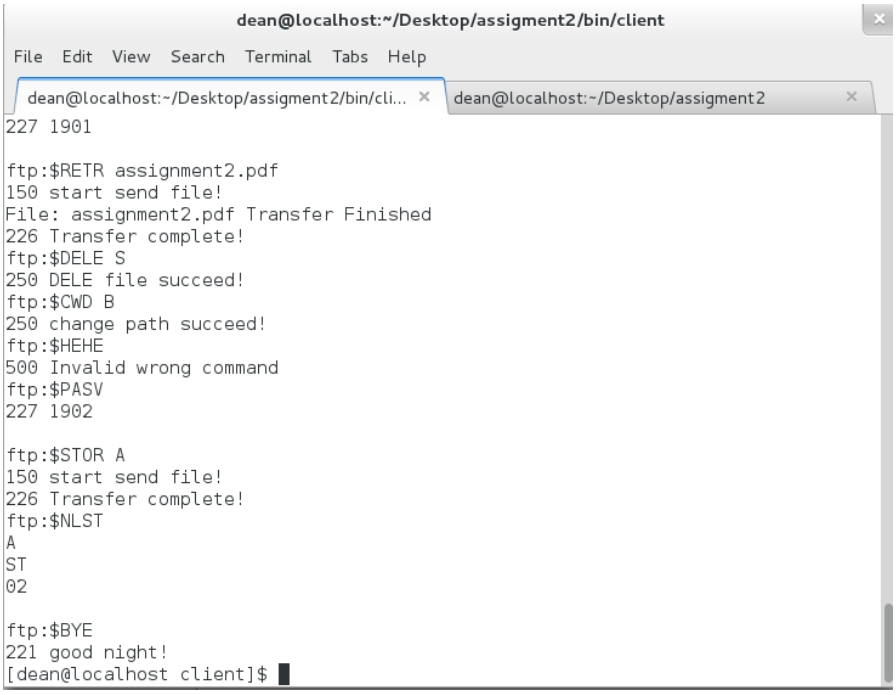

Figure 5: Connect with local server (2)

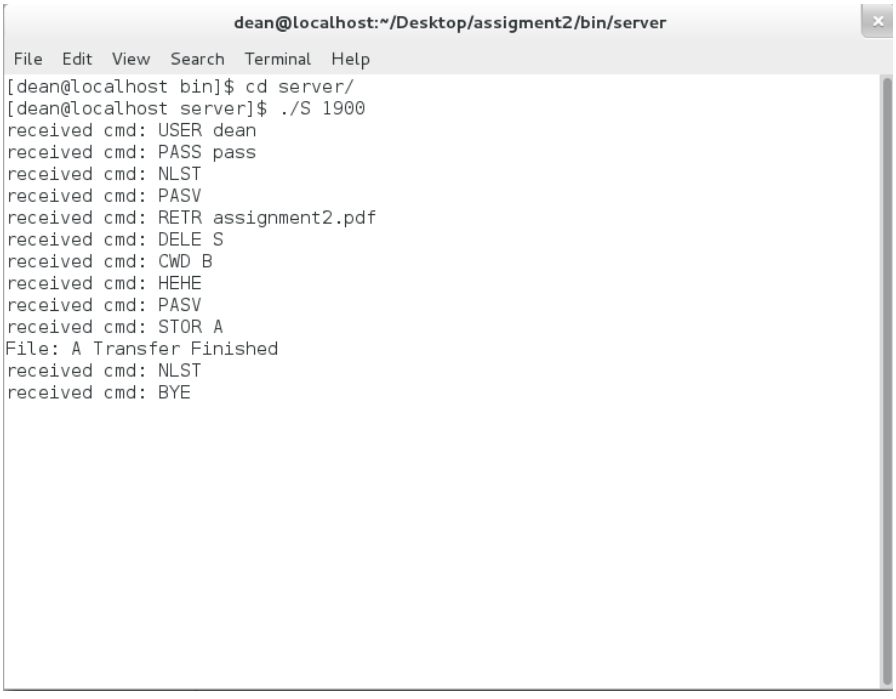

Figure 6: The local server