Tsinghua University, Architecture of computer and network(1)

Socket Programming Report

Shichen Liu, 2014013451

Date Submitted Oct.21, 2016

Project Statement

The project requires:

1. Modify the UDP server and client.
2. Create an FTP server and client from scratch. ...

Experiment environment

1. UDP
   (a) OS: Mac OS X 10.11.6
   (b) Python: 2.7.8
2. FTP
   (a) OS: Ubuntu 64-bit 14.04 / Mac OS X 10.11.6
   (b) GCC: 4.8.4
   (c) Python: 2.7.8

Answering questions

UDP

Q How to write a chat program(two clients chat with each other) with UDP?
A Each client serves as both client and server, which means using two threads to recieve from and send to each other simultaneously.
Q Can we use the UDP to transfer a file? If so, how?
A Of course we can. I come up with two different ideas.

1. First: use UPD as TCP, they establish several reliable transportations. For each package of data, make sure the client has received the data, then moves to the next package.
2. Second: like what we did in the assignment. First we send all packages in sequence with its serial number. Then we check what returned by the server. By analysing the returned serial number, we can see which blocks of data is missing. Then resend those data in series, till all the data blocks are assured.

FTP

Q How to transfer large files without blocking the server?
A We can use multi-threading to achieve this goal. Use an unblocking subprocess to transfer the file, and check it's status, or wait it trigger some kinds of event to finish the transmission.

Handled commands

Both FTP client and server can handle the following commands:

1. USER,PASS,RETR,STOR,SYST,TYPE,PORT,PASV
2. [**optional**]CWD and CDUP: Server check whether the given path is legal by comparing it with user current path ,specifically, when the required path exceeds the root, it will remain at the root. Accept either absolute path or relative path.
3. [**optional**]RMD and DELE: Server check whether the path is legal. Accept either absolute path or relative path.
4. [**optional**]RNFR and RNTO: Server check whether the path is legal. The server will only response to the *RNTO* command that directly appends after a *RNFR*. Accept either absolute path or relative path.
5. [**optional**]MKD: Accept either absolute path or relative path.
6. [**optional**]LIST

Features

server

ePoll The server uses e-poll to accept multi-client requests. It can receive events' trigger and therefore solve each event. In this way, the server will gain higher-concurrency.

options The server has been implemented with all of the optional commands.

platform The server should be used on ubuntu. Because e-poll can only be used on Linux. To build to execute file, use *make*.

client

GUI The client basically is the *client* executable binary file. However, the file *gui_client.py* at the same folder is the supporting GUI interface which should be used at the same path with the *client.c* file. Just execute *pythongui_client.py*, the python script can automatically generate executable file and start the GUI for you. The GUI interface using **Tkinter** lib of Python. Notice that in Mac OS X, when you execute the python file, the GUI will be displayed behind the termianl window, and you should manually bring it front.
The appearance looks like below:
Notice that the log info will be shown in corresponding color. What's more, by clicking the tree view, you can unfold the remote folder.

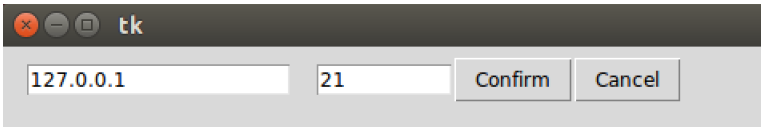

Figure 1: First step, host and port

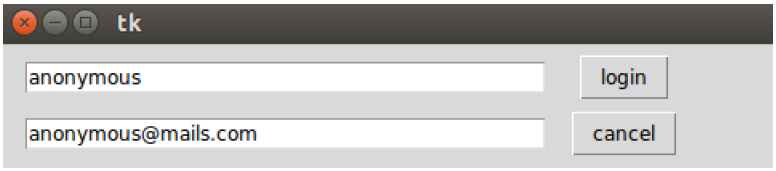

Figure 2: Second step, username and password

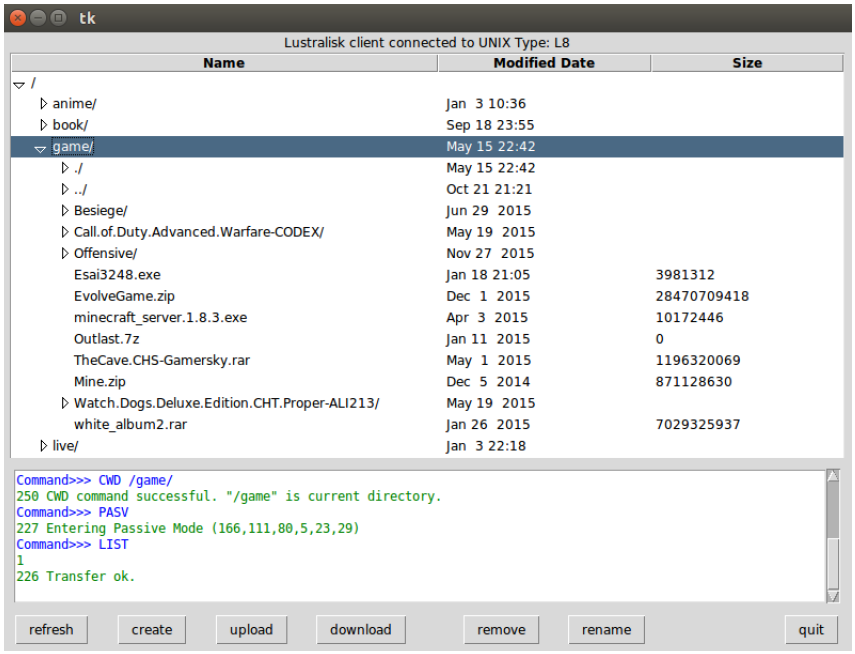

Figure 3: Third step, GUI

cross-platform The client can be built either on *Ubuntu* or *MacOSX* with the exactly same source code. Further more, the GUI interface can also run in both operating system.

encoding A serious problem the client faces is the encoding problem. In my implementation, the GUI client will ignore illegal characters.

GUI mode The client can tell the differences between GUI mode and manually commanding mode by the parameter $-gui$.

options The client has been implemented with all of the optional commands.

Deliverable

Several directories with *readme.md* file in it.