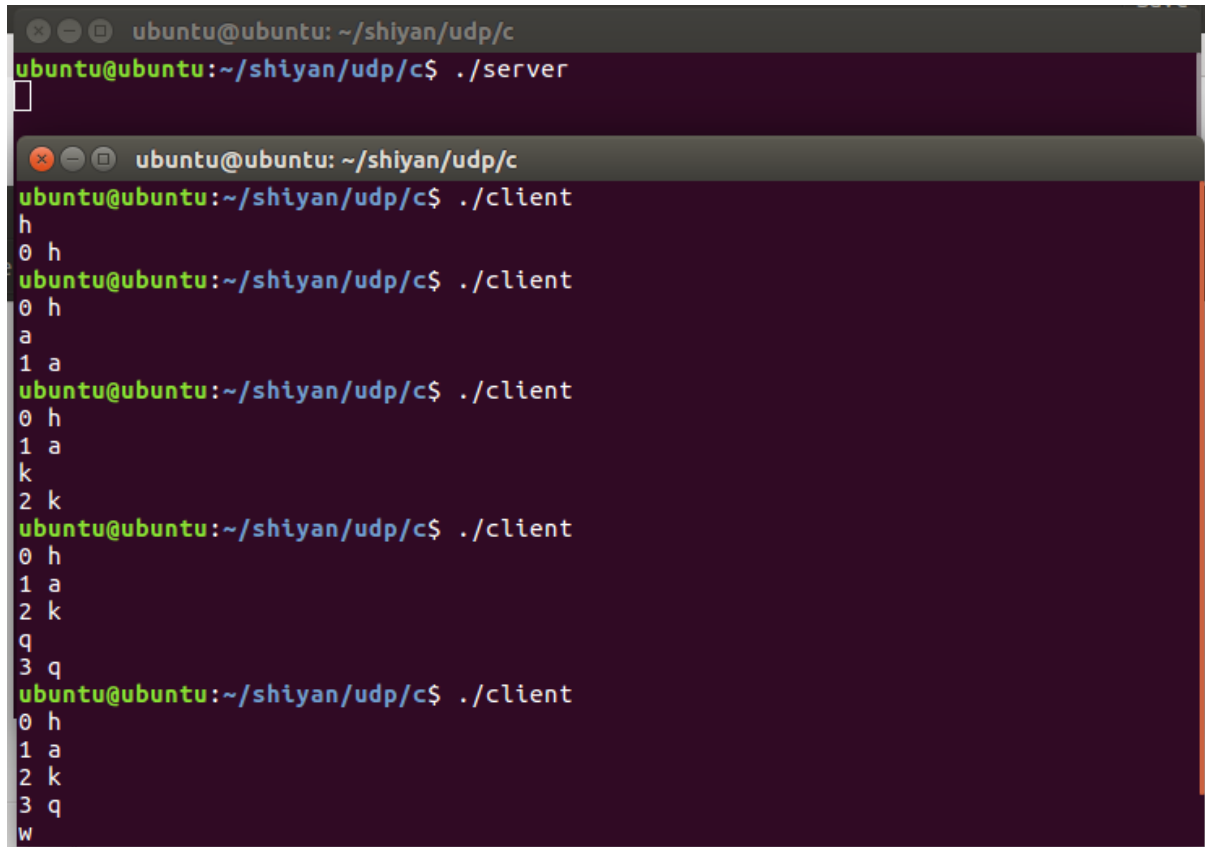


Socket Programming

201508012 软工52 李在弦

UDP

A terminal window with a dark purple background. The title bar reads 'ubuntu@ubuntu: ~/shiyang/udp/c'. The prompt is 'ubuntu@ubuntu:~/shiyang/udp/c\$'. The user has entered './server' and the prompt is now blank. A second terminal window is overlaid on top of the first. Its title bar also reads 'ubuntu@ubuntu: ~/shiyang/udp/c'. The prompt is 'ubuntu@ubuntu:~/shiyang/udp/c\$'. The user has entered './client'. The output shows a sequence of characters: 'h', '0 h', 'a', '1 a', 'k', '2 k', 'q', '3 q', 'w'. The user then enters './client' again, and the output shows: '0 h', '1 a', '2 k', '3 q', 'w'.

```
ubuntu@ubuntu: ~/shiyang/udp/c
ubuntu@ubuntu:~/shiyang/udp/c$ ./server
ubuntu@ubuntu:~/shiyang/udp/c$ ./client
h
0 h
a
1 a
k
2 k
q
3 q
w
ubuntu@ubuntu:~/shiyang/udp/c$ ./client
0 h
1 a
2 k
3 q
w
```

实现了UDP的功能

(Optional) Answer: How to write a chat program (two clients chat with each other) with UDP?

当用户要给别的用户发信息的时候，先尝试收对方发的信息，然后如果对方一段时间没有发信息的话就开始发自己的信息。然后如果对方收到了，那么要给我发标志字符（用来对方确认收到）

(Optional) Answer: Can we use the UDP to transfer a file? If so, how?

TCP

```
ubuntu@ubuntu: ~/shiyang/tcp/server
ubuntu@ubuntu:~/shiyang/tcp/server$ make
gcc -pthread -w -o server server.c
ubuntu@ubuntu:~/shiyang/tcp/server$ ./server -port 5050 -root ~/shiyang/tcp/server
ClientNum : 1
accept 127.0.0.1
[ ]

ubuntu@ubuntu: ~/shiyang/tcp/client
ubuntu@ubuntu:~/shiyang/tcp/client$ make
gcc -w -o client client.c
ubuntu@ubuntu:~/shiyang/tcp/client$ ./client -ip 127.0.0.1 -port 5050
FROM SERVER: 220 FTP server ready
[ ]
```

server启动的时候要输入 `-port` 和 `-root`，其中port是server的端口号，root是指定的目录。

做了一个makefile。我是分开server的makefile和client的makefile的。

在server的终端里面可以看到client的IP地址。

client登录了之后server回复信息220，欢迎client。

```
ubuntu@ubuntu: ~/shiyang/tcp/server
ubuntu@ubuntu:~/shiyang/tcp/server$ make
gcc -pthread -w -o server server.c
ubuntu@ubuntu:~/shiyang/tcp/server$ ./server -port 5050 -root ~/shiyang/tcp/server
ClientNum : 1
accept 127.0.0.1
ID : 1 >> USER anonymous
█

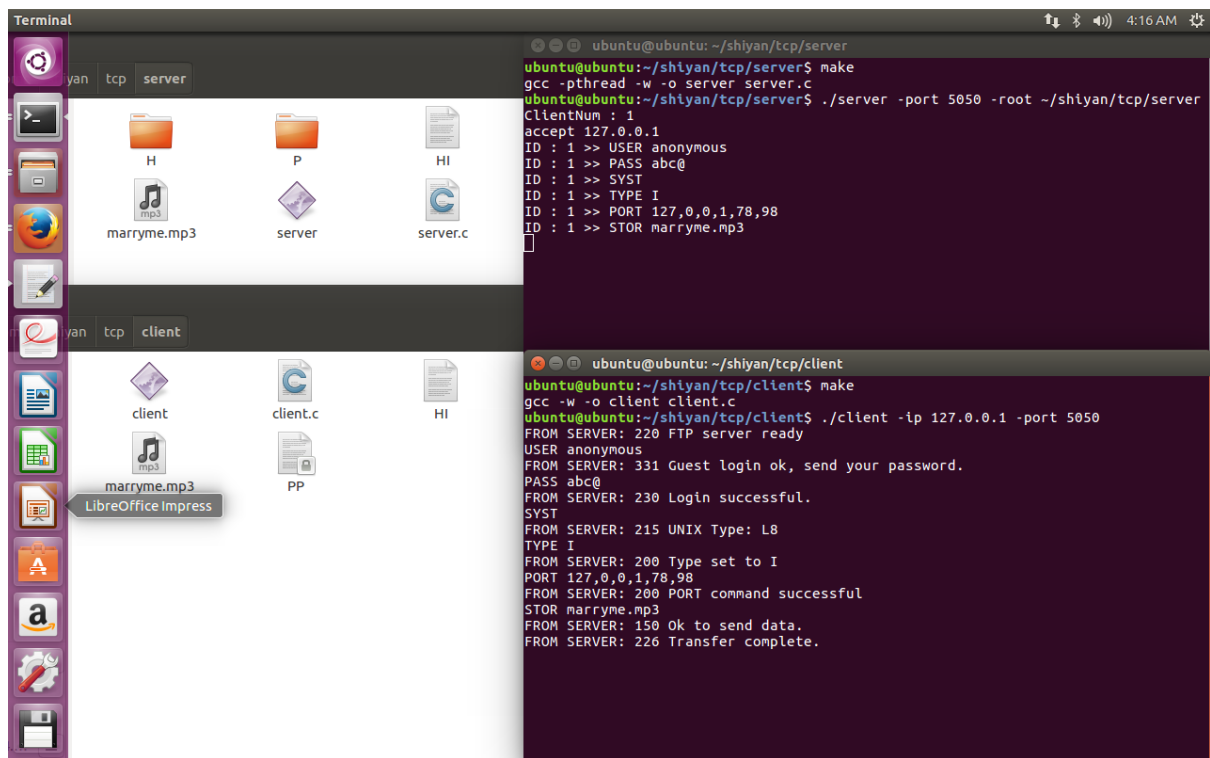
ubuntu@ubuntu: ~/shiyang/tcp/client
ubuntu@ubuntu:~/shiyang/tcp/client$ make
gcc -w -o client client.c
ubuntu@ubuntu:~/shiyang/tcp/client$ ./client -ip 127.0.0.1 -port 5050
FROM SERVER: 220 FTP server ready
USER anonymous
FROM SERVER: 331 Guest login ok, send your password.
█
```

实现了可以检查各个用户输入的命令的功能。client登录成功之后server回复331。

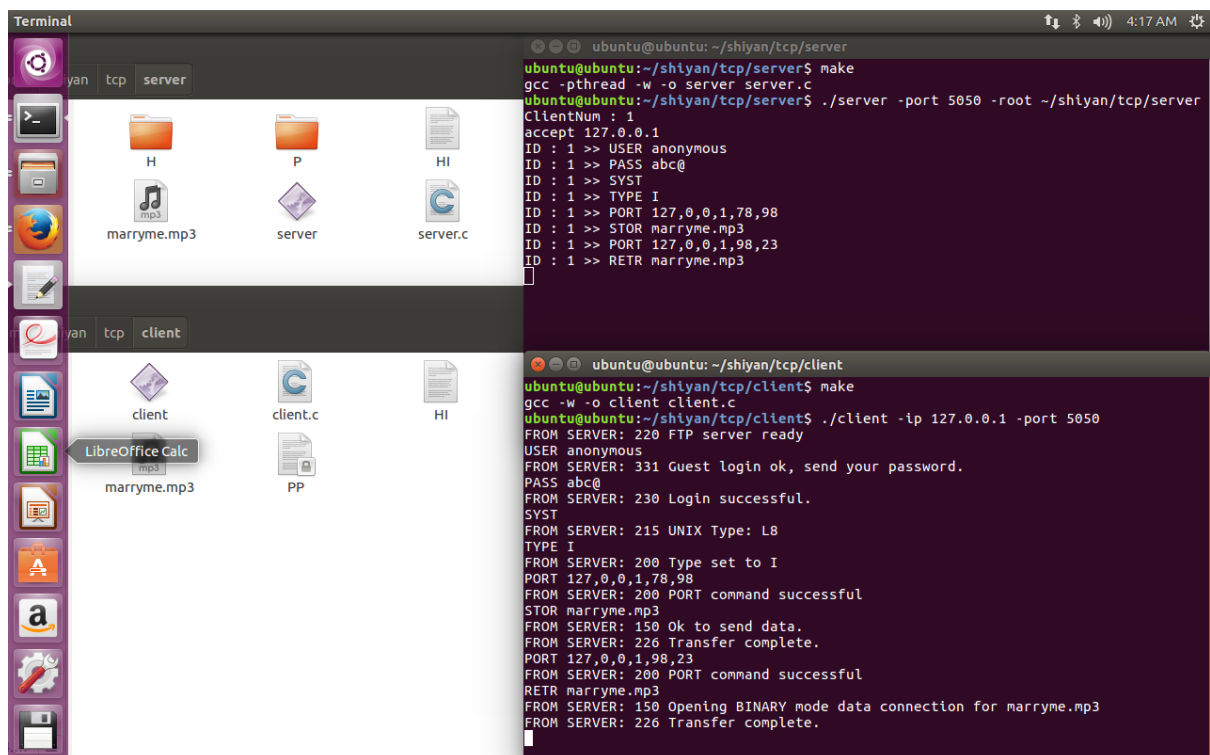
```
ubuntu@ubuntu: ~/shiyang/tcp/server
ubuntu@ubuntu:~/shiyang/tcp/server$ make
gcc -pthread -w -o server server.c
ubuntu@ubuntu:~/shiyang/tcp/server$ ./server -port 5050 -root ~/shiyang/tcp/server
ClientNum : 1
accept 127.0.0.1
ID : 1 >> USER anonymous
ID : 1 >> PASS abc@
█

ubuntu@ubuntu: ~/shiyang/tcp/client
ubuntu@ubuntu:~/shiyang/tcp/client$ make
gcc -w -o client client.c
ubuntu@ubuntu:~/shiyang/tcp/client$ ./client -ip 127.0.0.1 -port 5050
FROM SERVER: 220 FTP server ready
USER anonymous
FROM SERVER: 331 Guest login ok, send your password.
PASS abc@
FROM SERVER: 230 Login successful.
█
```

client输入正确的ID和PASSWORD的话，回复230，给他可以用我的server的权限。



上面是SYST,TYPE,PORT,STOR的回复信息。本来server文件夹里面没有了marryme.MP3文件，stor之后才产生的。



输入PORT命令之后可以用STOR,RETR的下载，上传文件的功能。

```
Terminal
ubuntu@ubuntu: ~/shiyang/tcp/server
ubuntu@ubuntu:~/shiyang/tcp/server$ make
gcc -pthread -w -o server server.c
ubuntu@ubuntu:~/shiyang/tcp/server$ ./server -port 5050 -root ~/shiyang/tcp/server
ClientNum : 1
accept 127.0.0.1
ID : 1 >> USER anonymous
ID : 1 >> PASS abc@
ID : 1 >> SYST
ID : 1 >> TYPE I
ID : 1 >> PORT 127,0,0,1,78,98
ID : 1 >> STOR marryme.mp3
ID : 1 >> PORT 127,0,0,1,98,23
ID : 1 >> RETR marryme.mp3
ID : 1 >> PASV
ID : 1 >> RETR server
[]

ubuntu@ubuntu:~/shiyang/tcp/client
USER anonymous
FROM SERVER: 331 Guest login ok, send your password.
PASS abc@
FROM SERVER: 230 Login successful.
SYST
FROM SERVER: 215 UNIX Type: L8
TYPE I
FROM SERVER: 200 Type set to I
PORT 127,0,0,1,78,98
FROM SERVER: 200 PORT command successful
STOR marryme.mp3
FROM SERVER: 150 Ok to send data.
FROM SERVER: 226 Transfer complete.
PORT 127,0,0,1,98,23
FROM SERVER: 200 PORT command successful
RETR marryme.mp3
FROM SERVER: 150 Opening BINARY mode data connection for marryme.mp3
FROM SERVER: 226 Transfer complete.
PASV
FROM SERVER: 227 Entering Passive Mode (127,0,0,1,78,32)
RETR server
FROM SERVER: 150 Opening BINARY mode data connection for server
FROM SERVER: 226 Transfer complete.
```

```
Terminal
ubuntu@ubuntu: ~/shiyang/tcp/server
ubuntu@ubuntu:~/shiyang/tcp/server$ make
gcc -pthread -w -o server server.c
ubuntu@ubuntu:~/shiyang/tcp/server$ ./server -port 5050 -root ~/shiyang/tcp/server
ClientNum : 1
accept 127.0.0.1
ID : 1 >> USER anonymous
ID : 1 >> PASS abc@
ID : 1 >> SYST
ID : 1 >> TYPE I
ID : 1 >> PORT 127,0,0,1,78,98
ID : 1 >> STOR marryme.mp3
ID : 1 >> PORT 127,0,0,1,98,23
ID : 1 >> RETR marryme.mp3
ID : 1 >> PASV
ID : 1 >> RETR server
ID : 1 >> PASV
ID : 1 >> STOR client

ubuntu@ubuntu:~/shiyang/tcp/client
FROM SERVER: 215 UNIX Type: L8
TYPE I
FROM SERVER: 200 Type set to I
PORT 127,0,0,1,78,98
FROM SERVER: 200 PORT command successful
STOR marryme.mp3
FROM SERVER: 150 Ok to send data.
FROM SERVER: 226 Transfer complete.
PORT 127,0,0,1,98,23
FROM SERVER: 200 PORT command successful
RETR marryme.mp3
FROM SERVER: 150 Opening BINARY mode data connection for marryme.mp3
FROM SERVER: 226 Transfer complete.
PASV
FROM SERVER: 227 Entering Passive Mode (127,0,0,1,78,32)
RETR server
FROM SERVER: 150 Opening BINARY mode data connection for server
FROM SERVER: 226 Transfer complete.
PASV
FROM SERVER: 227 Entering Passive Mode (127,0,0,1,78,33)
STOR client
FROM SERVER: 150 Ok to send data.
FROM SERVER: 226 Transfer complete.
```

实现了PASV被动连接功能，连接之后可以自由地上传，下载文件。client输入了STOR,RETR,LIST之后重新要用PORT或PASV来连接到server的数据端口。

```
ubuntu@ubuntu: ~/shiyang/tcp/server
ubuntu@ubuntu:~/shiyang/tcp/server$ make
gcc -pthread -w -o server server.c
ubuntu@ubuntu:~/shiyang/tcp/server$ ./server
ClientNum : 1
accept 127.0.0.1
ID : 1 >> USER anonymous
ID : 1 >> PASS abc@
ID : 1 >> SYST
ID : 1 >> TYPE I
ID : 1 >> PORT 127,0,0,1,78,98
ID : 1 >> STOR marryme.mp3
ID : 1 >> PORT 127,0,0,1,98,23
ID : 1 >> RETR marryme.mp3
ID : 1 >> PASV
ID : 1 >> RETR server
ID : 1 >> PASV
ID : 1 >> STOR client
ID : 1 >> powr

ubuntu@ubuntu:~/shiyang/tcp/client
FROM SERVER: 200 Type set to I
PORT 127,0,0,1,78,98
FROM SERVER: 200 PORT command successful
STOR marryme.mp3
FROM SERVER: 150 Ok to send data.
FROM SERVER: 226 Transfer complete.
PORT 127,0,0,1,98,23
FROM SERVER: 200 PORT command successful
RETR marryme.mp3
FROM SERVER: 150 Opening BINARY mode data connection for marryme.mp3
FROM SERVER: 226 Transfer complete.
PASV
FROM SERVER: 227 Entering Passive Mode (127,0,0,1,78,32)
RETR server
FROM SERVER: 150 Opening BINARY mode data connection for server
FROM SERVER: 226 Transfer complete.
PASV
FROM SERVER: 227 Entering Passive Mode (127,0,0,1,78,33)
STOR client
FROM SERVER: 150 Ok to send data.
FROM SERVER: 226 Transfer complete.
powr
FROM SERVER: 500 Unknown command.
```

如果client输入非法的命令，则server回复相应的信息。（在这个图片里面是500）

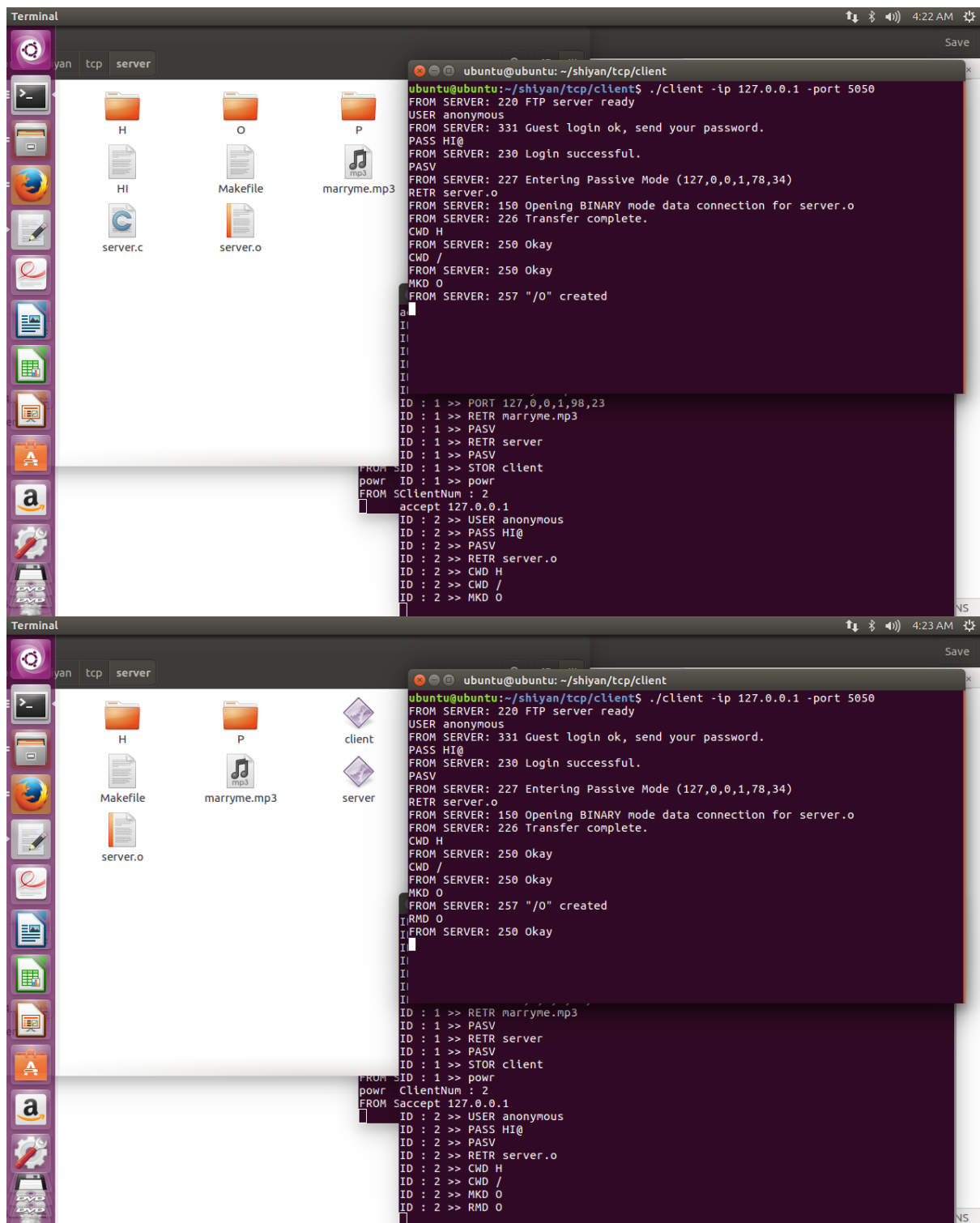
```
Terminal
shiyang tcp server
H P client
mp3

ubuntu@ubuntu:~/shiyang/tcp/server
gcc -pthread -w -o server server.c
ubuntu@ubuntu:~/shiyang/tcp/server$ ./server
ClientNum : 1
accept 127.0.0.1
ID : 1 >> USER anonymous
ID : 1 >> PASS abc@
ID : 1 >> SYST
ID : 1 >> TYPE I
ID : 1 >> PORT 127,0,0,1,78,98
ID : 1 >> STOR marryme.mp3
ID : 1 >> PORT 127,0,0,1,98,23
ID : 1 >> RETR marryme.mp3
ID : 1 >> PASV
ID : 1 >> RETR server
ID : 1 >> PASV
ID : 1 >> STOR client
ID : 1 >> powr
ClientNum : 2
accept 127.0.0.1
ID : 2 >> USER anonymous
ID : 2 >> PASS HI@
ID : 2 >> PASV
ID : 2 >> RETR server.o

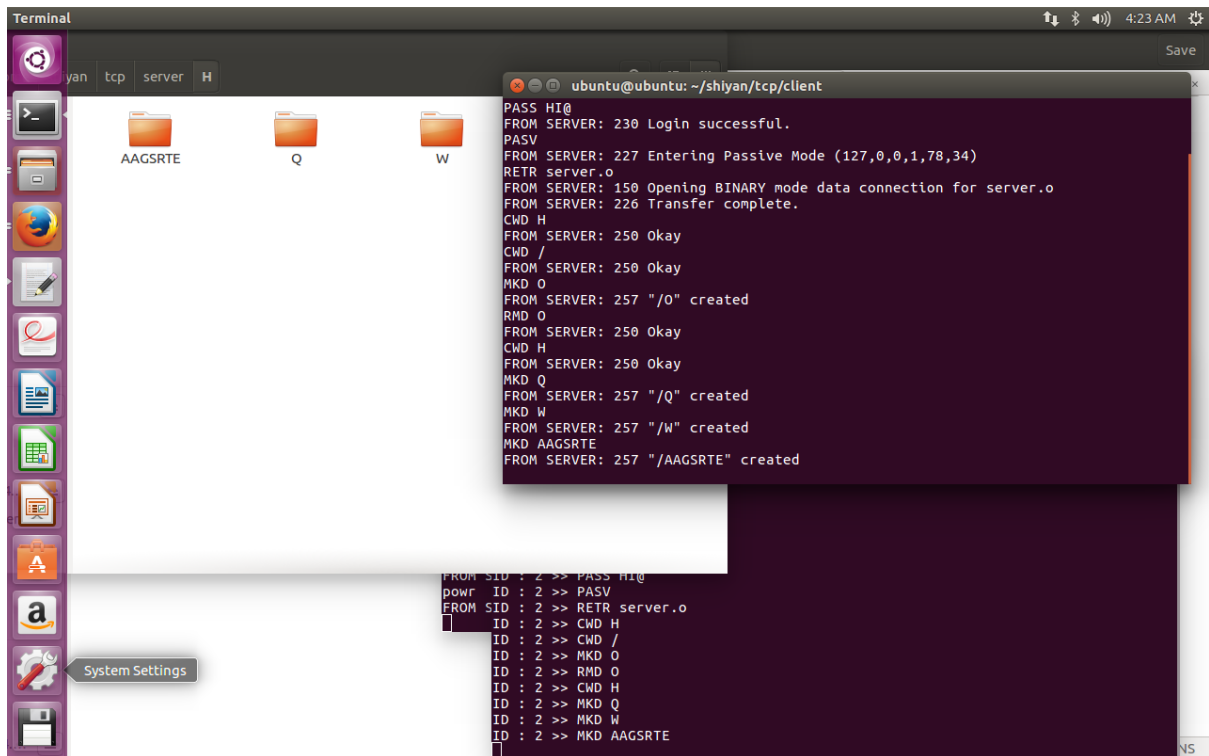
ubuntu@ubuntu:~/shiyang/tcp/client
ubuntu@ubuntu:~/shiyang/tcp/client$ ./client -ip 127.0.0.1 -port 5050
FROM SERVER: 220 FTP server ready
USER anonymous
FROM SERVER: 331 Guest login ok, send your password.
PASS HI@
FROM SERVER: 230 Login successful.
PASV
FROM SERVER: 227 Entering Passive Mode (127,0,0,1,78,34)
RETR server.o
FROM SERVER: 150 Opening BINARY mode data connection for server.o
FROM SERVER: 226 Transfer complete.

strlen("530 Please login with USER and PASS.");
C Tab Width: 8 Ln 58, Col 18 INS
```

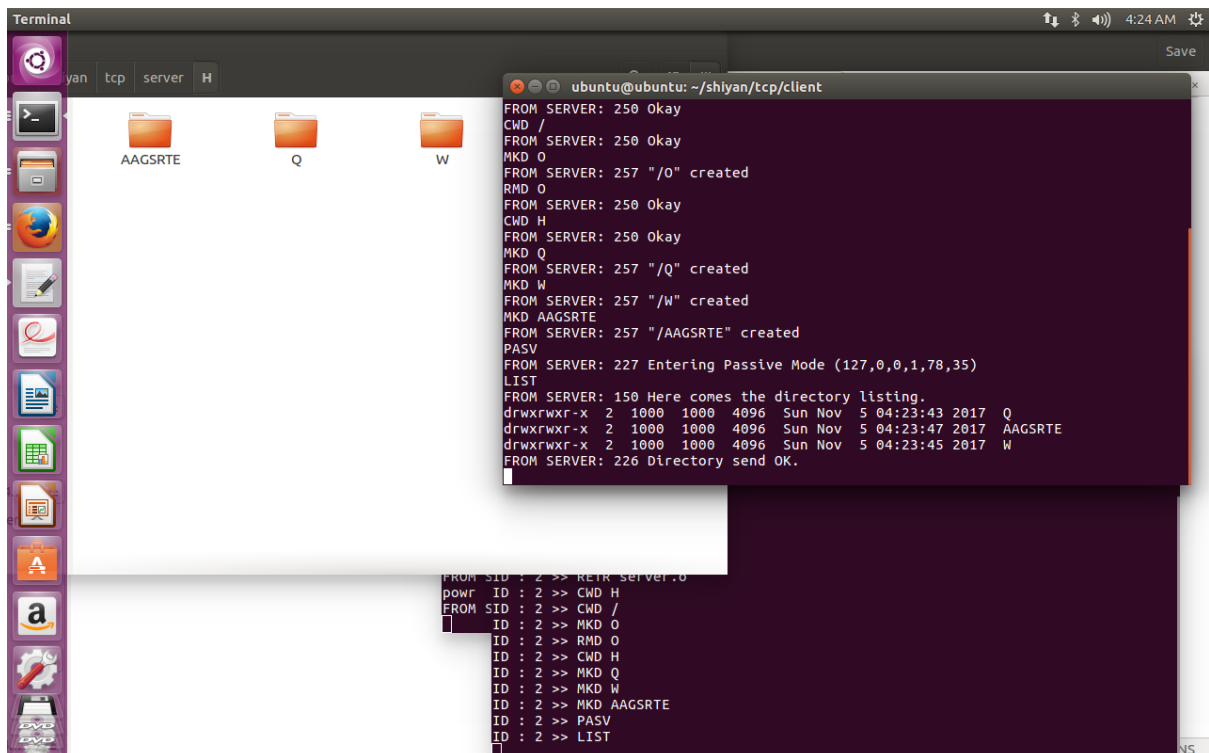
我用pthread实现了Multi Server。现在最多10个用户可以同时连接server。如果想增加最多人数，就可以改。



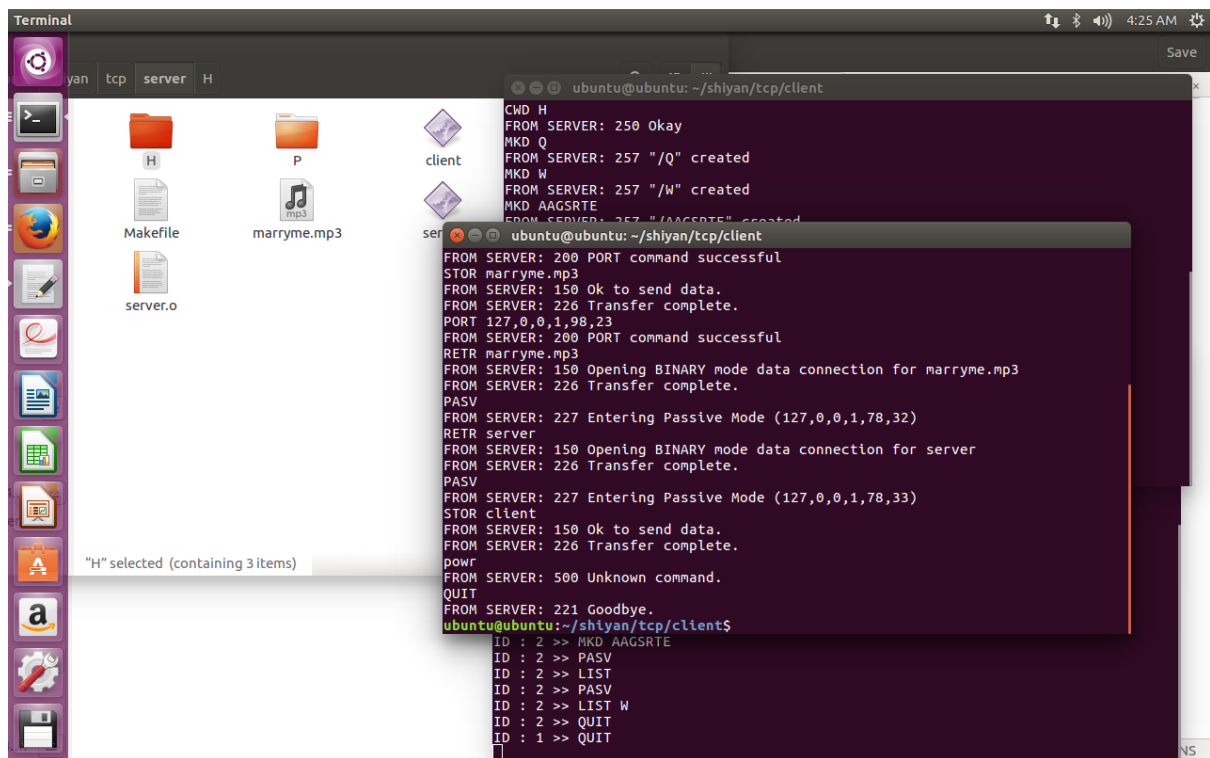
实现了CWD,MKD,RMD的功能。用MKD,RMD命令可以自由地删除，创建文件夹。用CWD命令可以自由地改变当前路径



用CWD命令进入到H文件夹之后再创建文件夹。



client用PORT或PASV模式连接新的TCP之后，用LIST命令可以看到文件的信息。如果LIST没有参数，那就直接传送当前路径里面的文件的信息。如果LIST有路径参数，那么访问那个路径之后传送文件信息。如果不是路径，而是文件的话，可以看到那个文件的信息。



用QUIT和ABOR可以推出系统。

还有很多提示异常情况的回复信息（比如说，client输入非法的命令的话，server就回复500）。

(Optional) How to transfer large files without blocking the server?

用pthread做多线程的程序，然后一边听着命令，然后一边发或者收信息。