

曹张杰

October 28, 2015

# socket programming 文档

本文档结构，第一部分是server的功能说明，第二部分是这次实验后的感想。

server功能说明：(应用的命令)

1、USER和PASS，USER是用户名的命令，PASS是用户密码的命令，输入 USER 后接任何用户名均可登陆（包括空用户名），PASS或者为空，或者为邮箱格式（至少有...@...'. '...的格式）；

2、PASV和PORT，RETR和STOR命令，PASV将在server端开启线程和socket监听的client的接入请求，之后用RETR或STOR命令传文件时由client端接入server端的文件传输端口并传输server文件或获取client端文件；PORT是在客户端监听，之后用RETR或STOR命令传文件时由server端接入client端的文件传输端口并传输server文件或获取client端文件。PORT需要client带上IP地址和本地端口号的六元组，RETR和STOR都需要带上文件名参数。

3、QUIT命令，无参数，退出ftp并关闭socket，释放网络资源；

4、SYST命令，无参数，默认返回系统设置215 UNIX Type: L8；

5、TYPE命令，无参数，默认返回200 Type set to I.；

6、NOOP命令，无参数，只允许有“NOOP”4个字母才能被响应，被响应则返回“200 Your NOOP is responded.”，否则返回“500 NOOP shouldn't have any parameter.”；

7、PWD命令，无参数，返回当前工作目录；

8、CDUP命令，无参数，将工作目录切换到当前目录的上一级目录；

9、CWD命令，带一个目录参数，指明要切换到的目录；

10、LIST命令，无参数，返回当前目录详细文件信息列表；

- 11、NLST命令，无参数，返回当前目录简略文件信息列表；
- 12、RNFR和RNT0命令，两个命令必须按顺序输入，第一个用参数制定要重命名的文件名，第二个命令的参数为改文件的新文件名（可以改变改文件的目录）；
- 13、MKD命令，创建目录，一个参数指定要创建的目录名；
- 14、RMD命令，删除目录，一个参数指定要删除的目录名；
- 15、DELE命令，删除文件，一个参数指定要删除的文件；
- 16、APPE命令，一个参数表示存储的文件，与STOR不同的是以a模式存储，就是接在原文件之后；
- 17、ALLO，已被弃置不用，返回202，否则为504；
- 18、MODE，已被弃置不用，仅仅MODE S返回200，否则为504；
- 19、STRU，已被弃置不用，仅仅STRU F返回200，否则为504。

client 说明：

- 1、可以加-port 数字端口号参数运行，将去接入相应的端口
- 2、若访问了不存在的文件或输入的命令不合法导致异常（不包括输入无法识别的命令），客户端将会退出。

实验感想：

本次socket实验，被要求用纯c语言写程序，不能使用c++的库，我觉得这是本次实验的第一大难点，因为从大一开始，我写c程序就没有把c和c++的边界分的很清，往往会在不知不觉中用了c++的库，但这次是用了纯c的库，也找了不少资料，看到了许多以前不曾看到的函数，比如对文件目录的操作，运行终端程序等，我充分感受到了c语言的强大，也提升了更底层的编程能力；

第二，这次是第一次网络编程实验，我知道了网络协议的规范性，甚至一个'\r'就可能出错，这是我之前所不了解的，也只有socket编程才能知道这些；

第三，就是做所有大作业都会有的感想，就是查查文档，看看代码。根据文档并且借鉴网上别人写的代码，对自身的开发也十分有帮助，拿到一个全新的任务肯定没思路，首先借鉴一下别人的代码，可以开阔思路，至少能了解一下大致架构，然

后自己慢慢填充，碰到问题就去查库函数，这样一步步摸索出来，当做出第一个功能后，后面的功能就可以越来越快的实现。

第四，回答一下tcp的optional问题的第二个，首先每个接入的客户都各自有一条线程处理，因此一个用户的文件传输不会阻碍另一个用户，同时为了防止大文件传输阻碍同一客户与server的交流，可以在收到RETR和STOR命令时再开启一个线程来接受或传输文件，那样客户线程就不会被阻塞而可以继续接受客户的命令。