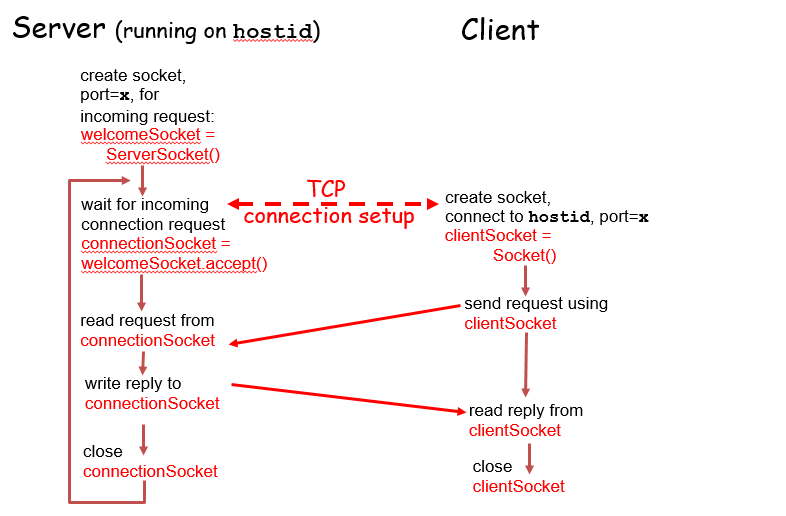
Socket Programming Stage1 Report

学号：2015013248 班级：软件52 姓名：刘祥

Task2 FTP

1. 实验原理



1. 实现指令

Server/ Client：USER, PASS, SYST, TYPE, QUIT,ABOR

目录操作：

MKD：在当前目录下新建文件夹

CWD：可以进入当前目录下的文件夹，也可以通过CWD ..进行回退

PWD：显示当前工作目录

LIST：将当前目录下的文件信息通过文件传输socket传输给client

RMD：删除当前目录下的文件

RNTO, RNFR：两条指令共同实现文件重命名的操作

传输文件：PORT, PASV,RETR, STOR,LIST

1. 重点

为了使代码相对整洁和有较高可读性，我采取如下方法：

1. 无论是client和server，有相同的用户状态标志和指令类型，并定义在相同的头文件const.h里.
2. 在传输文件的过程中，严格按照要求，每次传输完成即RETR，STOR，LIST指令执行结束后，立刻关闭相应的传输socket和监听，将server状态还原。
3. 代码运行：

Server端：gcc server.c –o server

sudo ./server (-port num) (-root path)

这里注意的是端口和工作目录可选，默认为21， /tmp

Client端：gcc client.c –o client

./client (–ip ipv4Address) (–port port)

这里注意的是server ip和端口可选，默认为127.0.0.1，21

1. 总结与反思

在本次实验中基本是按照文档的要求来进行编写，没有什么亮点。

遇到最大的难题就是在文件传输时的阻塞问题，比如传输文件后还需要接收或者发送指令，server和client一定要协调一致，这就需要我对每个socket控制好，如何时监听绑定，何时连接，何时关闭等等，注意好这些问题就可以实现文件的正常传输。

当前也增强了编程的能力，较熟悉python的一些常用指令和字符串操作、文件操作、传参等。

当然对Socket编程的学习也使得我对网络通信有了更深的理解，这点我认为还是很有帮助的。

最后感谢助教大大的批改！

祝好！