

# UDP&FTP 实验报告

孙士杰 软件 71 20160111119

## 一 . 实验环境

CPU: Intel(R) Core(TM) i7-6700HQ CPU @ 2.60GHz

RAM: 1GB

Operation System: (Virtual Box) Ubuntu 16.04 LTS

Programming language: C

## 二 . UDP

### 1. Socket 客户端主要步骤如下:

创建一个 socket->与服务器连接->发送数据->等待服务器应答->关闭 socket

### 2. Socket 服务器端主要步骤如下:

创建一个 socket->接收连接的请求->接受命令并处理->关闭 socket

### 3. UDP 问题回答:

1. 如何利用UDP 编写聊天软件: 客户端使用UDP 协议发出消息后, 如果服务器收到该消息, 则使用UDP 协议返回一个应答消息。

2. 是否可以利用UDP 传输文件? 如果可以, 如何实现: 可以在每次传输时在头部加上一个序列号, 这可以让接收端按照正确的顺序储存这些数据, 而不必管他们到达的先后顺序。为解决丢包问题可以让接收端每次接收到一个数据之后向发送端发出信号说明自己收到。

## 三 . FTP

ftp 中我实现了了 stage1 要求的内容, 实现了要求中需要实现的全部命令  
server 在 autograde.py 下测试得分是 40 分

client:

client 中, 由于未明确规定 PASV 模式返回的地址和端口模式, 这里我采用了说明文档中示例部分的(xxx,xxx,xxx,xxx,xxx,xxx)模式, 即 server 给出的地址需要是这种形式的才能够被识别。

client 中，由于使用了 socket 等 Linux 下特有的函数，目前尚未找到合适的方法在 Windows 下生成合适的可执行文件，但有 Linux 下的可执行文件，即 client.o 可供使用。另外，由于上述原因以及时间及能力所限，我未能成功做出 client 的 GUI 界面，在设计以及编写 UI 方面的能力尚需锻炼。

下面对几个命令进行说明：

- 1.USER，只能对 anonymous 有效；
- 2.PASS，未做检测，任意字符串皆可；
- 3.TYPE，只能跟 I；
- 4.RETR，STOR，LIST，REST 必须在 PASV 或 PORT 建立连接之后；
- 5.PORT，参照了历程中的格式，即后面的字符串是六个数字，由逗号隔开，前四个是 ip，后两个是 port；
- 6.REST 后加上跳过的字节数，然后就可传输文件；
- 7.LIST 后要再次建立连接才能传递文件；
- 8.RNFR 和 RNT0 必须搭配使用；
- 9.CWD，MKD 不成功会返回失败语句

Server：

Server 对于目录操作未做限制，密码未作要求，另外，在 PASV 模式中，直接将本机地址设为 127.0.0.1，在不同平台下运行，可能会有问题，目前尚未找到解决方法，但在本机上测试无问题。

另外，server 对于文件目录的处理采用了绝对路径。

Server 在 stage1 的基础上，实现了使用 FD\_SET 以及 select 等方法为基础的多用户连接；使用 pthread 的多线程使得大文件不阻塞，这一部分在 RETR 和 STOR 部分有体现，使用了一个结构体来传输参数；使用 REST 实现了文件的断点续传，输入偏移量使得传文件从那里接着开始在已有文件之后续写。

运行方式：

Server：在 server 目录下 make，之后输入 `sudo ./server`，-port 和 -root 是可选的，若没有，root 默认为 /tmp，port 默认为 21，

Client：在 client 目录下 make，之后输入 `./client`

加载完成后在 client 输入 USER anonymous, 然后 PASS xxx 登录, 上传/下载文件之前需要先指定模式 (PASV/PORT), RNFR 和 RNT0 需要搭配使用

#### 四 . 心得体会

这次大作业的难度是比较高的, 从来没有接触过网络编程的我一开始面对这个作业无从下手, 在参考了许多资料, 查阅博客以及询问助教和同学之后终于做出了一个可以运行的 FTP 客户端和服务端, 尽管还有很多问题, 但基本要求确实能够实现。在 stage2 的实验中, 在尝试使用 pthread 的过程中, 曾经出现过编译错误, 难以解决, 后来在助教的帮助下通过修改 makefile 实现了成功编译并运行。此外, 由于 recv 自有的阻塞性质和融合机制, 在程序运行时曾经出现两条指令合二为一的情况, 后来加了一个检测指标后程序可以运行。另外, 在 server 中, 为了解决多线程的参数问题, 设计了一个结构体, 并且把一些需要的变量声明与定义放在了 main 的外面, 使得多线程的函数可以共享参数。

这次实验我收获很多, 熟悉了网络编程的一些基本的知识, 当程序能够运行的时候心情还是非常激动。不过一开始思路还是比较混乱, 遇到了很多问题, 结构也没有很好, 后来才逐步解决。在这里也要向所以帮助过我的老师, 助教与同学表示感谢。