计算机网络：大作业 1

Socket Programming

林嘉纹 2021080077

程序简介

文件传输协议（FTP）是客户端和服务端在进行文件传输时所遵循的标准协议。在此次大作业中，通过 C 语言实现了 FTP server 程序和 FTP client 程序，其中，server 程序可以接收来自客户端的 FTP 指令，对其进行解析和操作后，再将返回码和信息响应给客户端，以此进行信息交互；而 client 程序可以将 FTP 指令发送给服务端，对服务端上的文件进行一定的操作。

完成此次作业所使用的环境为 Linux 虚拟环境，操作系统为 Ubuntu-20.04-64 位。在编程中，为实现进程时间的通信，使用了 Berkeley 套接字应用程序接口（Berkeley Socket API）。

运行方式

所写 FTP server 位于 server.c 中，FTP client 则在 client.c 中。为了运行简便起见，也提供了一个 Makefile 供编译运行使用。

使用自写的 FTP server 和 client 的方式如下：

1. 分别通过 gcc -Wall -o server server.c 和 gcc -Wall -o client client.c 编译 server 和 client 程序，若使用 make 命令，则可用 make server 和 make client 代替
2. 运行 server，若有指定的根目录和端口号，可通过 -root 和-port 参数传入，例 ./server -root /tmp/a -port 21000，若不特别指定，则默认根目录为 /tmp，端口号为 21。若使用 make 指令进行测试，可运行 make s，使用默认根目录及端口号 21012
3. 运行 client 的方式和 server 相似，需传入 ip 和端口号为参数，例：./client 127.0.0.1 21000。若使用 make 指令进行测试，可运行 make c，使用端口号 21012
4. Server 和 client 运行成功后，即可开始使用 client 向 server 传输 FTP 指令

FTP 指令

在此次实现的 FTP server 中，可以接收并进行处理的指令总共 16 个，其功能与使用方法如下表所示：

指令	参数	描述
USER	username	• Client 的第一个有效指令 • 在程序中设定的默认用户名为 anonymous • 若第一个指令不是 USER，或是用户名不存在，会输出错误提示
PASS	email	– Client 的第二个有效指令 – 在程序中设定的默认密码为 anonymous@ – 若 USER 后指令不是 PASS，或是密码错误，会输出错误提示
SYST	-	• 只要接收 SYST 指令，直接返回规定语句
TYPE	I	– 只接受参数为 I 的指令，并返回语句提示已经设置成功 – 对于其他的参数，概不接受
MKD	filename	• 在 server 的当前工作目录中创建一个名为 filename 的文件夹
RMD	filename	– 在 server 的当前工作目录中删除一个名为 filename 的文件夹
CWD	directory	• 从 server 的当前工作目录中进入到指定的 directory 路径
PWD	-	– 输出 server 的当前工作目录 – 为了保证服务器的访问安全，在输出当前工作目录时会隐藏服务器的根目录，由此，当前工作目录是根目录时会输出空串
RNFR	originalname	• 在当前工作目录中锁定一个要进行重命名的文件或文件夹 • 需要和 RNTO 成对使用，才能完成重命名操作
RNTO	newname	– 需要先运行 RNFR 指令才能使用 RNTO – 在运行了 RNFR 锁定待重命名的文件或文件夹后，运行 RNTO 再传入要命名的新名，即可完成重命名操作
PORT	p1,p2,p3,p4,h1,h2	• Client 指定使用 PORT 模式传输文件相关的资料 • 需要传入的参数为 client 方的 ip 和指定的端口号 • 获取 client 的 ip 和 port 后暂不进行连接，等待 client 输入操作文件的指令后才进行连接

		• PORT 的声明是一次性的，若 PORT 之后输入的指令无效或是已执行，需要再次发送 PORT 指令才能使用文件相关指令
PASV	-	– Client 指定使用 PASV 模式传输文件相关的资料 – 向 server 发出 PASV 命令后，等待 server 告知 ip 和端口号，进行连接后再获取 client 操作文件的指令 – PASV 的声明是一次性的，若 PASV 之后输入的指令无效或是已执行，需要再次发送 PASV 指令才能使用文件相关指令
LIST	-	• 将 server 当前工作目录中的内容罗列出来 • 需要在运行了 PORT 或 PASV 后才能使用
STOR	filename	– 将指定的文件从 client 储存到 server 的当前工作目录 – 需要在运行了 PORT 或 PASV 后才能使用 – 若上传的文件较大，需耐心等待指导提示传输完成
RETR	filename	• 将 server 当前目录下的文件下载到 client • 需要在运行了 PORT 或 PASV 后才能使用 • 若下载的文件较大，需耐心等待指导提示传输完成
QUIT	-	– Client 发送 QUIT 请求后，断开与 server 的连接

遇到的问题

在编程的过程中，遇到的大大小小的问题其实挺多的，以下列出一些较为特殊的问题：

问题：在使用 autograde.py 自动评测程序时，发现在使用 STOR 命令后，可以完成传输操作并返回正确的返回码，但是会在 filecmp.cmp 处出现错误，表示 server 和 client 两处的文件并不相同。

解决方法：经过一番研究和检查后，发现问题在于 server 在处理传来的指令时，没有去除字符串后的 \r\n，导致 server 方的文件名后存在 \r\n，肉眼上直观地看无法发现，在使用命令行 ls 后才发现是文件名不一致所致的。

思考：由此，让我对编程有了一些感悟，很多时候我们可能会忽略一些细微的地方，而正是这些小细节决定了我们程序的稳定性和运行时是否符合我们的预期。

```
if not ftp2.storbinary('STOR %s' % filename, open(filename, 'rb')).startswith('226'):
  print('Bad response for STOR')
  credit -= minor
if not filecmp.cmp(filename, directory + '/' + filename):
  print('Something wrong with STOR')
  credit -= major
```