FTP 实验报告

说明：

client 运行方式见同级目录下 pdf, server 运行方式为在 server/src 文件夹里输入以下 gcc 指令生成 server

gcc server.c processor.c processor.h -o server -lpthread

环境：gcc 7.5.0　Ubuntu 18.04　运行的脚本文件 autograde.py 要和 server 放在同级目录下

server

总体思路：

首先对命令行参数进行解析，然后创建并开始监听 socket，每当有一个用户连接就利用 pthread 新开一个线程，用户登出或断开连接后再关闭线程。线程中会存储当前客户端的一些信息，循环读入用户的输入并解析出命令和参数，进而为每一个命令调用相应的处理函数。

功能：

1. 能够接受 USER, PASS, RETR, STOR, QUIT, ABOR，SYST, TYPE, PORT, PASV, MKD, CWD, PWD, LIST, RMD,RNFR, RNTO,REST,APPE 命令
2. 同时支持多个用户
3. 支持大文件的传输
4. 支持文件的断点续传

实现过程中的一些难点和解决方案：

1. 命令行参数解析

   通过查阅博客学习使用 getopt_long_only 函数

2. 支持多用户

   使用 pthread_create 函数，为每一个客户端新开一个线程

3. PASV 和 PORT 命令中数据连接的建立过程

   若有已存在的数据连接，先进行关闭。

   PASV：通过博客上的方法，给公共 DNS 8.8.8.8 发送一个 UDP 包，根据其地址来确定本机 ip，然后随机生成端口号，将它们返回给用户，并和 socket 绑定，开始监听，下一次收到传输命令时调用 accept 函数接受客户端连接并传输数据。

   PORT：从输入中提取 ip 地址并用 inet_pton 函数进行转换，并将转换结果和端口号绑定给 socket，下一次收到传输命令时调用 connect 函数去连接客户端并传输数据。

4. 命令和文件数据的读取与写入

   因为 TCP 的传输是以流的形式，需要判断什么时候结束读取和写入，否则很容易发生阻塞。参考往届学长的笔记和自己实践得出，读取客户端的命令时，以'\r\n'作为结束标志，读取文件数据时，应在 while 循环里调用 read 函数，指定每次 read 的最大长度为 BUFFER_SIZE，若已知需读取的长度，则累加每次 read 的返回值（即实际读入长度）直到达到目标值，否则当某次 read 返回 0 时结束读取；write 函数同理。故在传输数据时，因缓冲区长度有限，应 read 一次后循环调用 write 将读取长度全部写入，循环这个过程直至读取完毕。

5. 文件路径的转换

对文件路径进行判断，若开头为‘/’则在前面连接上根目录路径，否则在前面连接上客户端当前的工作路径，从而转换为绝对路径。

6. 实现 LIST 功能（查看目录下的文件信息）

调用 popen 函数读入命令“ls -al”得到的内容

7. RETR，STOR 进行文件下载与上传

RETR：调用 open 函数以只读模式打开指定文件，若已有 REST 命令则调用 lseek 函数移动文件光标到指定读取点，随后读取文件内容并写入数据连接

STOR：调用 open 函数以只写模式打开指定文件（若无则创建新文件），读取数据连接内容并写入文件

8. REST，APPE 进行文件的断点续传

REST：将参数作为下次读文件时的起始字节记录下来，每次传输结束后重置为 0

APPE：基本实现同 STOR，只是将写入方式变为追加在末尾

9. 实现 MKD，RMD 功能（目录的创建与删除）

调用 mkdir 和 rmdir 函数实现

10. 实现 CWD 功能（当前工作目录的改变）

因本程序使用了 pthread，调用 chdir 函数会改变整个程序的工作路径，所以应通过字符串的拼接进行工作目录的记录和转换。将相对路径转换为绝对路径后，调用 access 函数判断是否存在，然后修改存储数据并输出

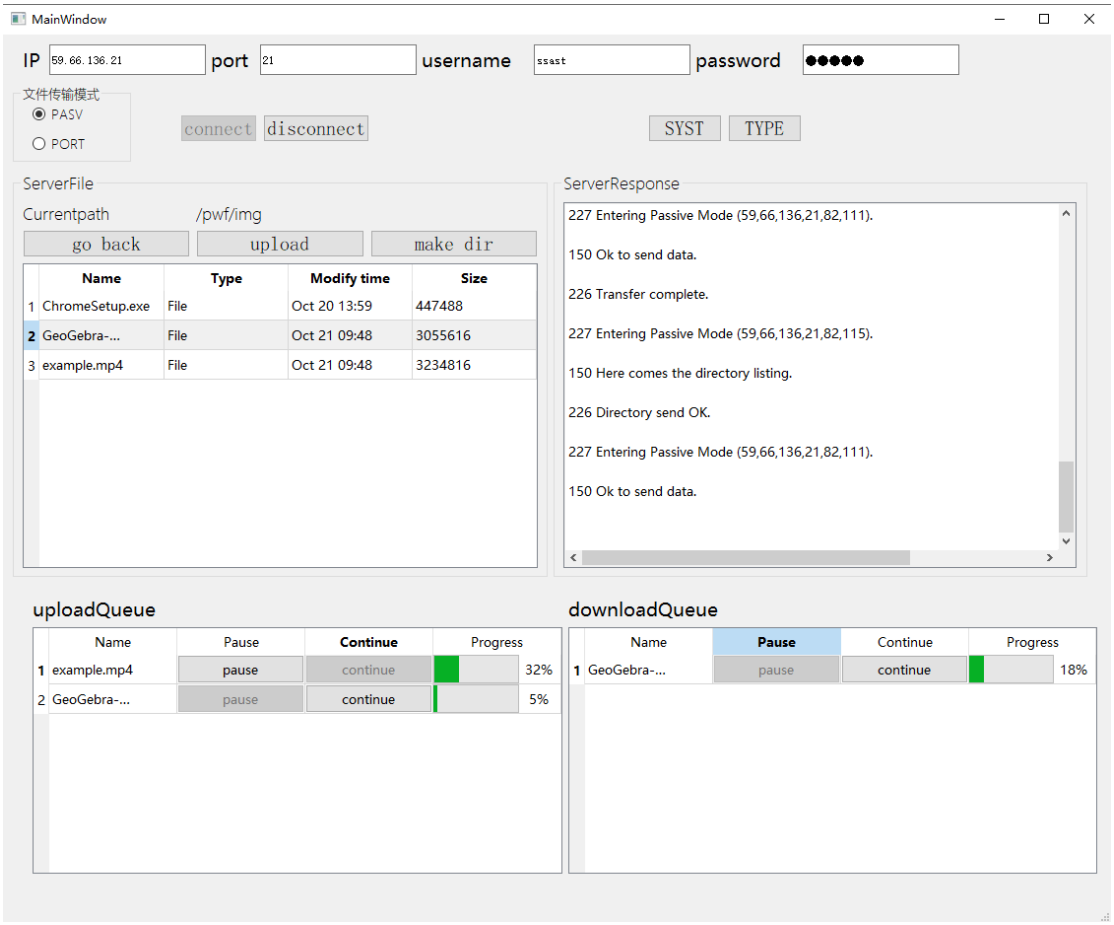
11. REFR，RETO 进行文件的重命名

REFR：调用 access 函数判断转换后的文件路径是否存在，若存在则暂存

RETO：检查是否有暂存的文件名，并调用 rename 函数实现重命名

client

界面：



实现过程中的一些难点和解决方案：

1. pycharm + Qt designer 环境配置

   参考博客在 pycharm 中添加外部工具 Qt designer 和 pyUIC

2. 如何展示文件列表和传输队列

   文件列表一开始计划使用 QtreeView 中的 QfileSystemModel，后来发现这个模型仅适用于读取本机上的文件，最后参考往届学长的实现方法，利用 Qtablewidget 进行文件数据信息的分列展示和整行选中。

3. 客户端对命令连接和数据连接传输内容的读取

   根据作业文档，服务器的命令回复结束是通过最后一行以 3 个数字的状态码及空格表示，而数据传输结束则与服务器的判断同理，循环调用 recv 函数直至读到的数据为空。