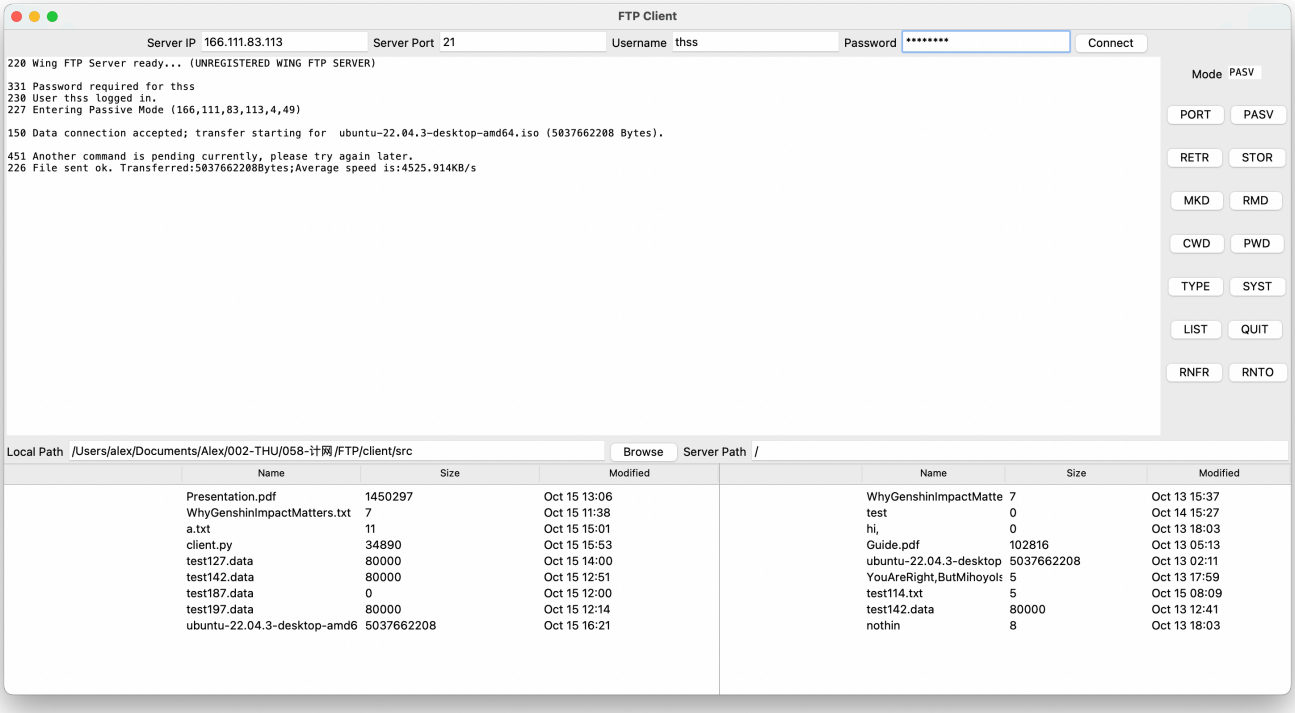
Project Report

1 FTP 介绍

我使用了c语言实现了FTP server的这些功能：`USER`,`PASS`,`RETR`,`STOR`,`QUIT`,`SYST`,`TYPE`,`PORT`,`PASV`,`MKD`,`CWD`,`PWD`,`LIST`,`RMD`,`RNFR`,`RNTO`,`ABOR`指令。

我还使用python语音实现了FTP client的这些功能：`USER`,`PASS`,`STOR`,`RETR`,`QUIT`,`SYST`,`TYPE`,`PORT`,`PASV`,`MKD`,`CWD`,`PWD`,`LIST`,`RMD`,`RNFR`,`RNTO`,`ABOR`指令。

还包括两个附加功能，分别是User-fiiendly GUI 和 File transmission without blocking the GUI.（见下图）



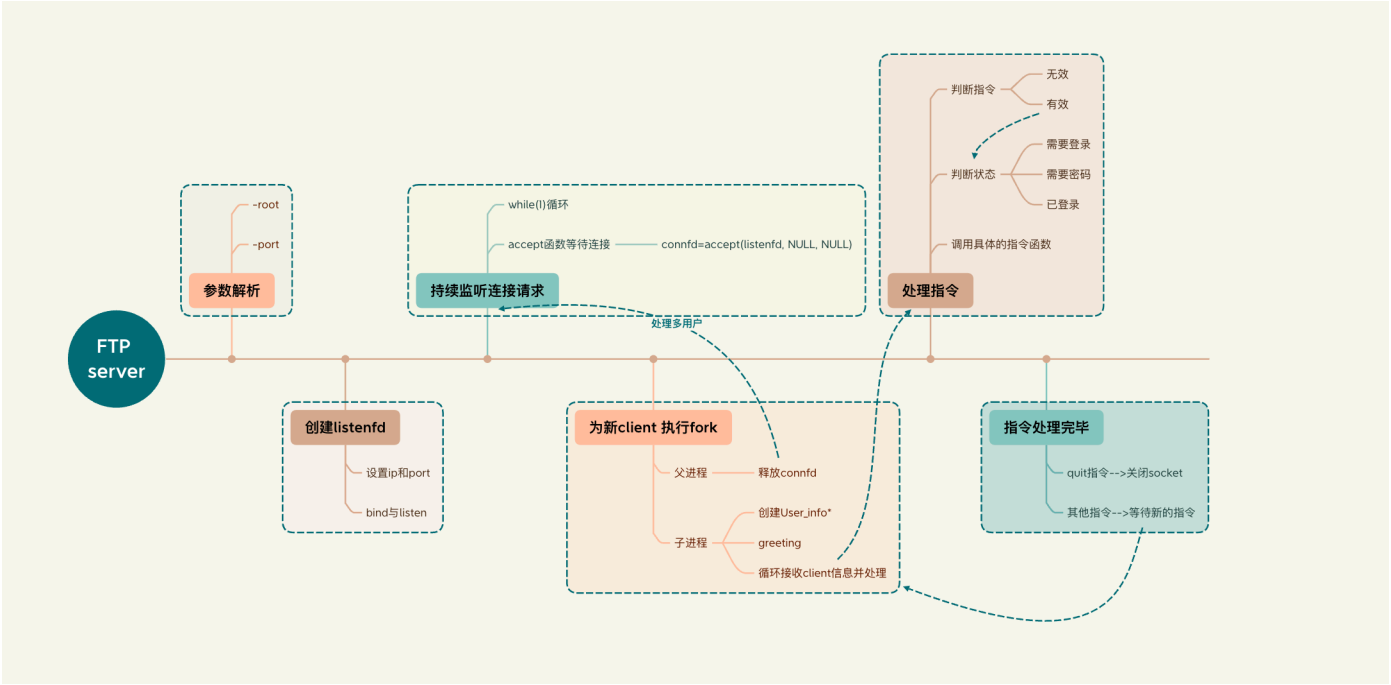
可以看到，当用户正在从服务器`RETR`文件`ubuntu-22.04.3-desktop-amd64.iso`的时候（文件大小为`5037662208 bytes`),GUI并没有被阻塞，仍然可以接受新的指令（见服务器回复451）

2 实现思路、难点与亮点

2.1 server

server一共有两个文件，分别是`server.h`和`server.c`。在`server.h`中主要定义了记录用户信息的`User_info`类、统一处理用户信息的`handle_client`函数，和处理具体的各个指令的`handle`开头的函数。在`server.c`中，主要是对输入参数的解析、监听socket的建立、用户的连接，以及对多用户的处理。

server的整体思路如下：



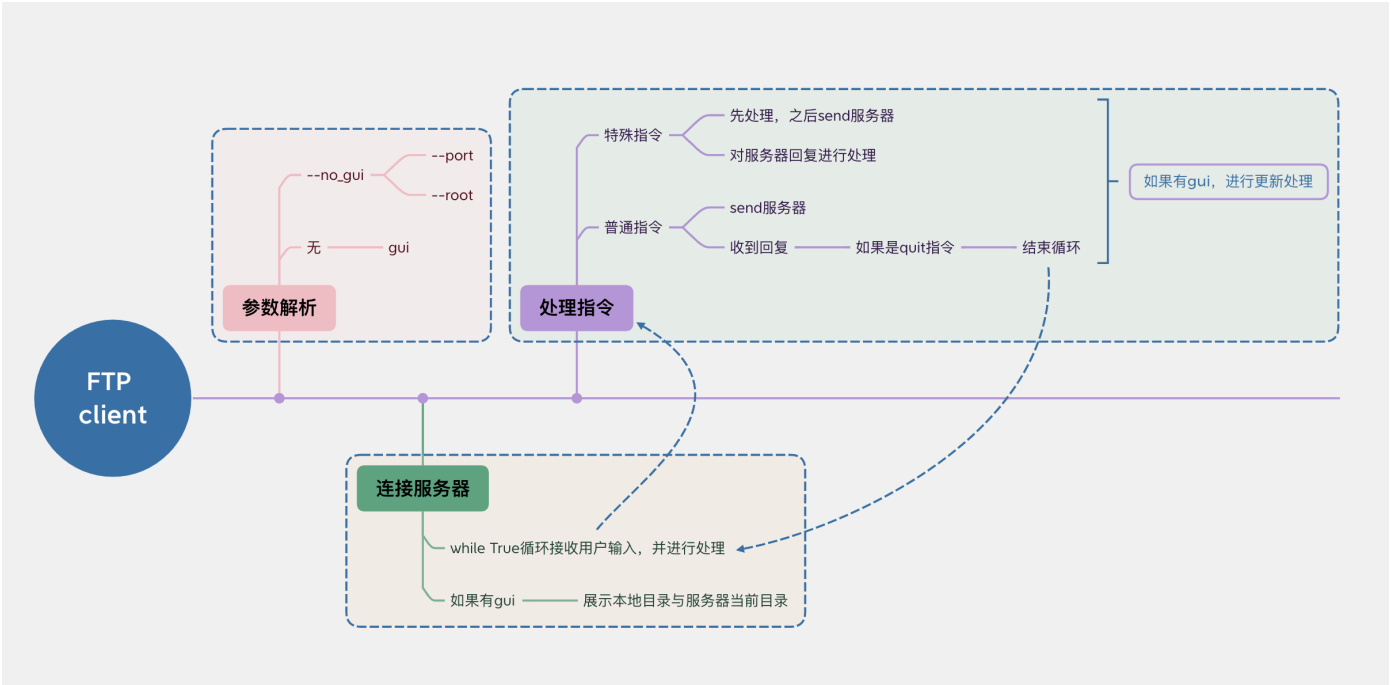
因为要处理多个用户，并且每个用户不同指令直接也存在一定逻辑关系，所以我创建了`User_info`类来记录重要信息，比如当前用户的状态`status`（需要登录/密码/已登录），用于文件传输的`file_socket`，传输文件采用的模式`data_mode`，方便判断重命名功能是否有效的`last_command`等。

此外，对于client的访问权限我也进行了设置。首先是，`LIST`，`RETR`，`STOR`的参数中不能包含`../`。这是为了保护文件的安全。其次，client不能访问运行`server`时指定好的根目录以外的地方，所以在处理`CWD`等函数的时候，我会检查该路径是否在根路径之外。

2.2 client

client部分只有一个文件，即`client.py`。在该文件中，我定义了`ClientGUI`类，用于创建GUI，以及`Client`类来处理用户指令和服务器的回复。

client的整体思路较为简单：



我还完成了GUI的两个附加功能，分别是User-fiiendly GUI 和 File transmission without blocking the GUI.在完成这两个功能的时候，遇到的主要问题有几个，第一个是及时更新服务器当前路径和路径下的文件。因为获取这些信息的唯一途径就是给服务器发送指令，并且进行信息提取，最终展示出来。获取服务器当前路径下的文件需要用到`LIST`指令，而这需要我先发送`PASV`指令建立数据连接。因此我写了一个`update_server_table`的函数，在里面我先给服务器发`PASV`指令，成功建立连接后，我再发送`LIST`指令。之后我对返回的信息进行parse，获取文件或文件夹的名字、大小和修改日期。再把每一条数据插入到`server_table`中进行展示。整个过程比较复杂繁琐。因此每次对服务器进行和文件相关的操作之后，我都调用`update_server_table`函数来进行更新。

另一个难题就是File transmission without blocking the GUI，为此，我在retr和stor的时候都使用了线程来处理真正的文件操作，而父线程继续接受用户的新指令。这样传输文件的时候，GUI并不会被阻塞。

3 总结

这次作业让我更深入的了解了FTP的工作原理与socket在计算机网络中发挥的作用。而这也是我第一次用python做GUI，我学习并使用了`tkinter`库，得到了新的历练和成长。总而言之，这是一次非常难忘的经历。