

Report

1. 实验环境

- UDP: python 3.9
- FTP server: Linux Ubuntu 20.04
- FTP client: python3.9 + PyQt5

2. 实验成果

- UDP成功实现了数据交互
- FTP server实现了要求的所有指令，支持多客户端连接，可以处理常见的异常情况发送合适的错误码。使用 `autograde.py` 评测达到满分。
- FTP将所有要求的指令通过UI展现，接入本地虚拟机的server和科协提供的服务器进行了多次测试，能够正确解析目录下的文件，完成文件上传和下载，由于时间关系，能够处理一部分的异常，并未进行完全的单元测试。

3. FTP指令

- `USER` : 发送用户名，server接收用户名
- `PASS` : 发送密码，服务器确认登录
- `RETR` : 服务器获取当前目录下的指定文件
- `STOR` : 将本地文件上传到远程服务器的目录下
- `QUIT` : 结束登录，返回统计信息，退出连接，再次登录需要重新连接。
- `ABOR` : 结束登录，返回统计信息，中断连接，再次登录需要重新连接。
- `SYST` : 查询服务器的操作系统类型
- `TYPE` : `TYPE I`指定文件的传输方式是二进制。
- `PORT` : 下一次文件传输为主动连接，客户端开启端口等待服务器的连接。
- `PASV` : 下一次文件传输为被动连接，服务器端生成一个随机端口并返回，等待客户端的数据连接。
- `MKD` : 创建新的目录。
- `CWD` : 修改当前的工作目录
- `PWD` : 返回当前的工作目录
- `LIST` : 生成当前工作目录下的所有文件信息
- `RMD` : 删除当前工作目录下的一个空目录
- `RNFR` : 指定要重命名的目录或文件名称
- `RNT0` : 指定目录或者文件的新名称

4. FTP Server

数据结构

为了支持多客户端同时访问，设计了 `Clients` 结构体，记录每个客户端的连接、读写状态、传输文件名称，传输字节大小等信息，详见 `config.h` 文件

功能划分

我将程序用到的函数划分成了四种类型，包括命令处理、路径处理、socket处理和其它类型。

- `cmd_handle` : 包括了指令分发的函数 `cmd_handler` 和所有指令的具体实现
- `path_utils` : 包括了获取文件绝对路径，检验是否是文件夹、文件，删除文件等函数的实现
- `socket_utils` : 包括初始化客户端结构体套接字、生成新的套接字、关闭传输控制命令的套接字和传输数据的套接字等函数的实现
- `utils` : 包括了检查ip和端口、发送控制命令、传输数据的函数的实现。

核心功能实现

- 多客户端工作目录的维护：server维护根目录、clients结构体维护每个客户端的相对的工作目录，通过这种方式，每个客户端可以在不同的工作目录下传输数据。
- 传输文件：通过反复从文件中读取固定字长的字节并发送，用读取的长度作为传输文件停止、继续的标志。同时用读取的字节长度来更新客户端结构体的偏移量，方便后续断点重发功能的实现。但是很遗憾由于时间原因，该选做功能并未实现。
- 多用户支持：如同guide里面所说，不一定要使用多线程实现。只要把客户端的状态合理划分好，每一次都遍历所有处于连接状态的客户端并分别接收命令进行处理，同样可以实现对多客户端的支持。

5. FTP Client

- UI界面实现

使用 QT Designer 设计好后用pyUIC生成python代码。之后感觉原生的QT界面美观性欠缺，使用QSS进行美化

- 核心功能实现

见 `FTPclient.py` 文件，顶层类 `FTPClient` 有成员类 `client`、`login`、`window`，功能如下

- `login` 是登录界面，需要用户输入用户名、密码、ip地址和端口信息
- `window` 是处于登录状态后实际操作的页面
- `client` 实现了所有要求的指令和server进行交互

`FTPClient` 其它的成员函数主要用于 `window` 页面按钮的槽函数，对FTP的命令实现了UI的封装。

实验感想

这次实验对我来说难度比较大，一开始对着文档看了两天一直不知道如何下手，后来学习了socket之后简单实现了几个命令后才有一点思路。然后设计好程序的结构后每天都不断的重复看文档、写代码、调试。先实现了server之后，有一点点写完了client，之后发现连实验室的服务器后还会阻塞住，又调试两天后终于赶在ddl前做完了。在实验过程中对UDP、TCP的原理更加清楚了，也学会了简单的socket编程和PyQt5，我感觉收获非常大，感谢老师和助教学长的辛勤付出！