

FTP大作业报告

软73 沈冠霖 2017013569

0.UDP

客户端通过一个循环将0-50发给服务器，服务器初始化seq为0,然后每收到一个就更新seq，将收到的数据和seq发回去就可以了。

1.服务器端

1.1 实现的指令

USER, PASS进行登录

PASV, PORT进行被动，主动模式连接

STOR, RETR下载和传输, REST断点续传

文件操作 LIST, CWD, MKD, RMD, PWD, RNFR, RNTD

其余的SYST, TYPE, QUIT, ABOR

1.2 支持多个客户端登录

每接收到一个链接请求，客户端就多开一个线程来处理, 进入处理指令的主函数。

1.3 传输文件不阻塞服务器

接收到RETR, STOR指令并且连接数据连接后，多开一个线程进行数据传输。同时，为了方便管理数据传输，把当前连接状态，连接类型，文件名，文件大小，开始位置等存入一个结构体进行操作。

1.4 文件操作方法

CWD, MKD, RMD, PWD, 重命名指令等用C语言的各种函数实现，参考了[这个网页](#)。

LIST指令通过调用linux命令行实现，使用popen函数调用ls -l指令获取文件信息，参考了[这个网页](#)。

2.客户端

客户端使用pyqt5在ubuntu16.04中实现。

2.1 登录

点击按钮后连接服务器的IP和端口，然后发送USER, PASS指令登录。

2.2 文件操作等指令（进入，后退，跳转，增加，删除，重命名）

计算出对应的文件名或目录，然后发送对应的指令

2.3 文件列表

连续发送PASV和LIST获取数据，然后提取对应的文件信息。使用QTableWidget进行显示，同时使用Qt.CustomContextMenu实现了右击显示菜单功能，便于对单一文件/文件夹进行操作。

2.4 上传，下载列表

用QTableWidget进行显示，用SetCellWidget方法设置里面的单元格是按钮和进度条，也用这个方法进行更新。

2.5 下载和上传文件

下载连续发送PASV，RETR两个指令进行。为了不阻塞客户端，用QThread多开一个线程来接收数据，用信号-槽机制更新进度条，参考了[这个网页](#)。在传输中如果用到涉及数据连接的其他指令会报错。

2.6 暂停和断点续传

更新全局变量来暂停传输，客户端切断socket连接，并且记录当前传输的大小。此时服务器能通过read/write接口的-1返回值捕捉到错误，返回错误码，接收错误码即可暂停。

然后连续发送PASV，REST，RETR/STOR来进行断点续传。

2.7 退出

发送QUIT指令。

3.总结

这次大作业，我了解了Socket api，FTP协议，以及pyqt5的使用方法。主要遇到的困难是如何不阻塞进程进行上传，下载，这个用多线程实现;还有pyqt5的各种操作也让我费了很大劲。感谢从业臻同学的帮助，也感谢王泽宇学长提供客户端设计参考。