计算机网络 FTP 实验报告

软件92 周雨豪 2018013399

1 实验环境

操作系统： Ubuntu 18.04 for server, Windows 10 for client

内存： 8GB

语言： C for server, Python 3.6 for client, PySide2 for GUI

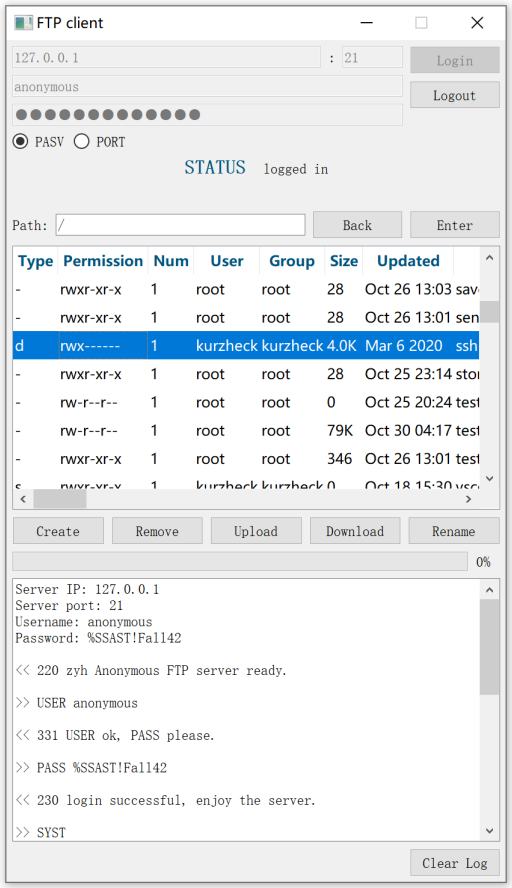
2 实现功能

2.1 Server

1. 实现的指令有 USER, PASS, SYST, TYPE, QUIT, PASV, PORT, MKD, CWD, PWD, RMD, RNFR, RNTO, LIST, STOR, RETR (作业要求的16个) 指令以及REST指令。已通过autograde.py脚本测试。
2. 用多线程实现对多客户端的支持。
3. 大文件传输。

2.2 Client

1. 实现的指令有 USER, PASS, SYST, TYPE, QUIT, PASV, PORT, MKD, CWD, PWD, RMD, RNFR, RNTO, LIST, STOR, RETR (作业要求的16个) 指令并能够与本实验的Server以及标准FTP服务器通信。
2. 支持浏览FTP服务器文件，创建、移除、重命名目录以及上传下载文件。
3. 有文本框输出通信记录。
4. 实现了GUI界面。



客户端界面如图所示，在上方输入合法的服务器信息后即可登录，单选框选择数据传输模式，FTP服务器的目录显示在中部，单击可以选中，双击进入文件夹，也可以通过直接输入路径进入对应目录，Back按钮返回上一层目录。下排按钮支持新建文件夹、移除文件夹、上传文件、下载文件、重命名。文件下载或上传进度显示在进度条。最下方的日志窗口显示通信记录，可以点击底部按钮清除日志。

3 实验方法及难点

3.1 Server

Server 的整体结构是主线程监听21端口，有连接则创建一个子线程持续收取指令，在通过USER和PASS后才处理其他指令，指令处理的函数定义在`command_handler.h`内，其处理过程调用的辅助函数定义在`util.h`内。当接收到用户的QUIT指令后退出关闭子线程。

多用户 理论上支持的客户端数量上限是10，定义结构体`ThreadParam`用来存储每个用户子线程的数据，包括连接模式、控制/数据连接标识、PORT传输端口等等。每个线程内部的逻辑其实相当简洁，即主要顺序执行两个函数`Login()`和`HandleCommand()`，前者处理USER和PASS指令，后者事件循环处理其余所有指令。

路径 工具函数中的`AbsPath()`负责了对路径的处理，将接收到指令中的路径参数，转换为服务器的本地绝对路径，支持了各种格式的路径参数，包括相对路径和绝对路径。由于参数可能会出现形如`..`，`../..`，`file`，`dir/file`，`/dir/file`等各种形式，所以用C处理这部分字符串写起来相对困难。并且客户端不能访问根目录上级的目录。

其他 由于网络上和主机上端口号的大小端表示不同，因此绑定地址时需要用`htons`转换成网络字节序。

3.2 Client

Client 使用PySide2提供的PyQt组件实现GUI界面，定义了ClientWindow类，使用Qt的信号槽机制，用户操作触发槽函数，槽函数中去调用相关的command handler和工具函数，设计时所有的command handler都仅负责发送指令和接受回复。

获取本机IP 向DNS 8.8.8.8发包后`getsockname`获取本机IP

目录显示 使用`QTableWidget`显示文件列表，支持单机选中和双击进入操作，工具函数`RefreshTable()`确保了列表在用户有新建/删除/切换目录后刷新内容。