

report

report

1. 项目说明
 - 1.1 项目简介
 - 1.2 文件说明
2. 功能实现
 - 2.1 功能设计
 - 2.2 界面设计
3. 实现方式
 - 3.1 server端多用户
 - 3.2 传输文件不阻塞
 - 3.3 断点续传

1. 项目说明

1.1 项目简介

本实现目标为依据 RFC 959 标准实现简单FTP协议。设计server端与用户友好的client端（图形界面）。

实现 USER, PASS, RETR, STOR, QUIT, SYST, TYPE, PORT, PASV, MKD, CWD, PWD, LIST, RMD, MV, REST 命令。

1.2 文件说明

- server:
 - handler(.c .h) : 实现RETR等命令。
 - server.c : 实现监听端口，并为用户分配线程。
 - utli.h : 配置文件，设置返回信息及命令列表等。
- client:
 - fileprocess(.cpp .h) : 实现文件处理操作，提取server端返回的文件列表信息。
 - filethread(.cpp .h) : 实现**上传及下载**文件的多线程。
 - msgprocess(.cpp .h) : 处理server端返回的信息。
 - mainwindow(.cpp .h) : 定义界面及响应事件。

2. 功能实现

本次实验的 server 端与 client 端的均实现了 USER, PASS, RETR, STOR, QUIT, SYST, TYPE, PORT, PASV, MKD, CWD, PWD, LIST, RMD, MV, REST 命令。

下面为重点功能的实现列表：

2.1 功能设计

功能	实现说明
server端多用户	server端每接受一个连接，则为这个连接新开一个子线程。
大文件传输不阻塞	server端：每个文件传输(RETR 、 STOR)均为一个子线程 client端： PASV 模式下，每个文件传输均在子线程中进行。
断点续传	实现REST命令，对于 RETR 操作支持断点续传。

2.2 界面设计

主要设计 用户验证区、命令行区、文件列表区、任务列表区。在每个区域右侧为与其相关的操作button或者输入框。

- 文件列表区域自动更新：每次影响文件列表的操作执行后就会更新列表，如：STOR、MKD、MV等。
- 文件列表区域：
 - 双击文件夹所在行可以进入目录
 - 点击选中一行之后，在选择右侧button，即可对这一行进行操作。

3. 实现方式

3.1 server端多用户

使用 pthread 的方式，为每个新的连接分配一个 Connection 实例，存储其线程独享的内容。

3.2 传输文件不阻塞

- server端：对于每一个连接（一个子线程），在进行文件操作时，我们通过 pthread 创建一个新的线程。并为其分配 conPack 实例，存储 socket 等内容。
- client端：对于client发出的每一个文件传输指令，我们为使用 QThread 为其分配新的线程。
- **client端文件传输结束之后自动更新目录：**

因为在QT不支持目标函数类型的多线程，所以，我们继承QThread，在类中定义 SIGNAL，在传输结束之后发送。在主线程中实现槽函数，实现了进程间通信。

3.3 断点续传

在server端定义REST指令，client端把当前文件的byte数目传给server。server从下一个字节开始传输。