

# FTP 实验文档

陈刚 2016010579

## 一、项目简介

### 1.1 server

项目文件包括 func.c, func.h server.c Usrc, Usrc.h 以及 makefile。

其中 func.c、func.h 实现了各种命令处理函数，以 resp<Commandname>命名这些函数，同时包括各种辅助函数如命令的读取回复、参数解析、文件的传输、目录的管理等等。Resp<Commandname>函数大部分实现流程都比较相似，一般是：检查用户状态->检查参数合法性&&试图获取参数->处理命令->回复客户端。其中涉及到目录操作的函数均比较复杂，需要处理相对路径绝对路径的情况以及考虑路径合法长度、目录存在性等问题。处理数据传输的几个函数也比较复杂，除了连接的建立，也需要考虑文件路径的各种问题。

Usrc,Usrc.h 实现了简单的用户结构体，用户变量包括用户当前状态、操作根目录、数据传输 socket，断点续传 position 以及和 port, pasv 指令相关的变量。此外，还有一些简单的函数实现用户初始化、状态清除和释放内存。

server.c 中执行 main()函数，通过 pthread\_create()创建多线程实现同一时刻多用户连接。在线程处理函数中使用 while 循环不断读取用户命令调用 func 函数进行相应处理。server.c 也实现了命令行参数解析函数，以获得程序运行的必要参数。同时 server.c 中还定义了一些全局变量，多文件可见的变量包括默认的根目录和主机 IP，还有部分 server.c 文件内可见的变量如命令字符串数组、线程数组、用户套接字数组等，便于在一些函数中共享使用。

### 1.2 client

Client 使用 Qt 实现图形界面，项目文件包括 ftpgui.cpp, ftpgui.h, ftpgui.ui, QTclient.pro, main.cpp, picrsc.qrc。ftpgui.ui 是 ui 设计文件。在 ftpgui.cpp 通过信号槽机制，连接槽函数处理 ui 组件发出的各种信号。主要槽函数包括处理连接、处理登陆登出、处理本地目录、处理远程目录、处理文件操作、处理 list 操作等等，这些函数再调用一些更简单的小函数进行复杂操作。

## 二、实验环境

服务器和客户端都在 Ubuntu 18.04.1 虚拟机下实现。

客户端使用 QT 5.13.1 GCC 64bit。

## 三、项目特点

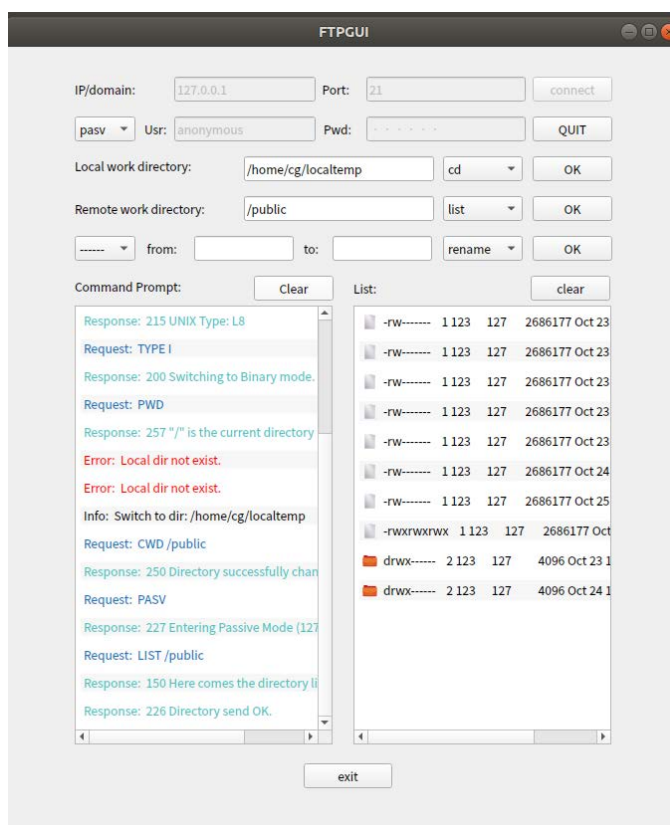
### 3.1 server

1. 项目组织结构清晰。
2. 在各种操作进行前都进行了状态检查，避免了未登录非法访问以及错误命令顺序的情况。
3. 对命令进行参数解析，处理不正确命令参数的问题。
4. 文件目录操作有详细处理：检查了文件目录长度是否合法；处理相对路径和绝对路

- 径两种情况；使用 strcpy()、sprintf()处理文件目录长度时候检查缓存长度等等。
5. 严格依据 rfc959 规定处理各种指令，有非常详细的各种错误码返回。

### 3.2 client

1. 用户操作简单，只需要填写参数框、选择命令、点击确认按钮就可以实现各种功能。
2. 用户操作安全，只能按照一定的顺序执行操作，不会出现操作顺序不正确的情况。
3. 用户能得到详细信息，包括客户端发送给服务器的信息、服务器返回的信息、操作错误信息、本地操作成功信息都会显示在 prompt 框中，且以不同的颜色显示，简洁明了。List 指令得到的目录会经过排序显示在特定的文本框中，且通过图标表明是文件还是文件夹，方便浏览查看。
4. 界面比较美观。



## 四、遇到的问题

1. C 语言中文件读写函数很多，puts(), gets(), read(), write(), fputs(), fgets(), fread(), fwrite()。这些函数有一些细微差异，稍不注意就有可能使用中出现错误，挑选合适的文件读写函数花费了一定时间。
2. 二进制字符串有一定特殊性，使用 strlen() 不一定能得到正确的长度，需要使用文件二进制读写时候返回的长度。刚开始使用 strlen() 导致文件传输不完整。
3. 各种文件文件夹函数需要注意使用方法，比如 rmdir() 只能删除空目录，需要自己实现函数递归删除目录。
4. port, pasv 指令有详细规定，并不是在命令后立即建立数据连接，而是在 retr, stor 命令处理过程中再进行相应的连接，需要格外记录一些状态，增加了编程复杂性。完成大作业过程中一波三折，使用 vsftpd 测试时候发现需要先建立数据连接才会回复 150，又经历了很多修改。

5.客户端意外关闭导致主线程直接退出的问题，用 `send()` 指定最后一个参数为 `MSG_NOSIGNAL` 替代 `write()` 可以解决问题。