

项目文档

金昕祺

2016010524

一、项目简介

本项目实现的ftp服务器、客户端均基于Ubuntu系统，具体的系统版本为：内核版本为Linux version 4.15.0-33-generic (buldd@lcy01-amd64-024)，发行版本为Ubuntu 18.04.1 LTS。本项目的make基于GNU Make 4.1，gcc版本为7.4.0。

本项目实现的ftp服务器、客户端均支持USER、PASS、RETR、STOR、QUIT、ABOR、SYST、TYPE、PORT、PASV、MKD、CWD、PWD、LIST、RMD、RNFR、RNTO、REST命令。

客户端基于python3.6。

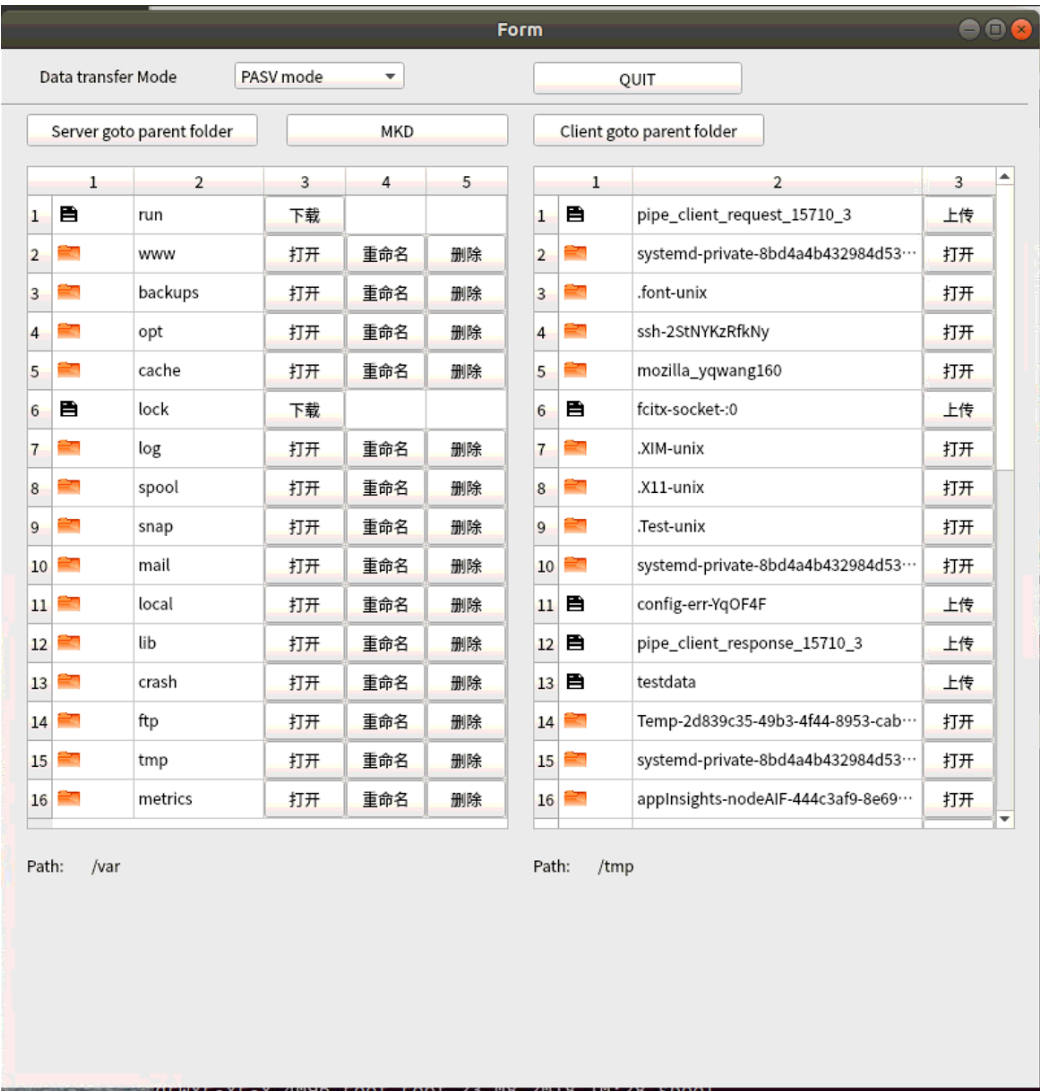
二、服务器端亮点

1. 服务器端支持多客户端同时连接，具体的实现方案是在提交的server.c的main函数中，程序通过while循环不断处理客户端连接，具体的方式是使用BSD socket提供的accept函数接受客户端连接请求。accept函数接收到客户端连接请求后会创建一个新的socket来处理该客户端，这之后我的程序会创建一个子进程，在子进程中使用accept函数创建的新socket来处理客户端各种命令，而在父进程中继续使用accept函数接收其他客户端发起的连接请求。
2. 较高的安全性：服务器端程序在处理RETR、STOR命令时，会检测命令的参数是否以“../”开头，如果是的话会拒绝进行相应的操作，并返回合适的错误码。
3. 服务器端支持RFC 959中提到的三种LIST命令的形式，包括LIST <CRLF>、LIST [<SP> <file_name>] <CRLF>和LIST [<SP> <folder_name>] <CRLF>。
4. 服务器端以64bit的int64_t类型的偏移量支持断点续传功能，这意味着我们的程序可以支持较大文件的断点续传功能，具体的实现方式是在提交的utils.h中定义的结构体connectionInfo内以int64_t类型的start_position记录REST命令指定的文件偏移量。除此之外，由于本人在实现断点续传时，使用了<unistd.h>中的lseek函数，而该函数的offset参数的类型位off_t，为了保证将记录偏移量的int64_t类型的变量赋给lseek函数的off_t类型时不会溢出，我在makefile中的编译命令里使用了-D_FILE_OFFSET_BITS=64这个flag。
5. 服务器端考虑并处理了大量复杂情况，具有较好的稳定性，包括但不限于：（1）检查了密码格式是否合法，定义的规则是密码要么等于“anonymous@”要么是形如^[a-zA-Z0-9_-]+@[a-zA-Z0-9_-]+(\.[a-zA-Z0-9_-]+)+\$这个正则表达式的字符串，检查的过程由utils.c中的checkEmailAddr函数实现。（2）检查了收到PASV命令后服务器端创建数据传输socket是否失败，若失败会返回相应的错误码给客户端（除此之外，收到PASV命令后服务器端创建数据传输socket时，使用了我定义的gen_port函数，该函数在20000-65535这些端口号上尝试绑定一个数据传输socket，尝试的顺序是随机的，但能保证执行gen_port函数时每个端口号只被尝试最多一次，且直到所有端口号都被尝试绑定一次仍未成功时才确认socket创建失败）。（3）在客户端只是连接了服务器但未登录的情况下，客户端发送除QUIT、ABOR、SYST、USER、PASS以外的其他指令时会被服务器拒绝执行，服务器还会返回相应的错误码。（4）检测了几乎所有指令的格式是否合法，包括但不限于SYST指令是否严格为SYST <CRLF>格式、PWD指令是否严格为PWD <CRLF>格式。（5）能识别RNFR/RMD/CWD指令指定的文件夹不存在的情况、MKD指令创建文件夹失败的情况、RMD指令删除文件夹失败的情况，并返回合适的错误码。
6. 为了提升服务器端处理RETR指令时数据传输的速率，本项目直接采用了Linux特有的<sys/sendfile.h>中的sendfile函数。sendfile函数是在内核内完成了拷贝，而常见的实现方式是通过结合read和write函数来传输数据，这种结合read和write的方式需要将数据传输到用户空间并从用

户空间中传输出来，性能较差。因而，本项目比结合read和write这种常见的RETR指令处理方式高效很多。

三、客户端亮点

客户端界面如下图所示：



如图所示，我们的客户端具有如下优点：

1. 能同时显示客户端的工作目录和目录中的内容、服务器端的工作目录和目录中的内容，并支持客户端的工作目录向上级目录跳转、服务器端的工作目录向上级目录跳转等功能。
2. 在客户端工作目录的文件浏览器中提供了两种按钮，分别用于切换客户端工作目录、上传文件（实现方式是基于STOR指令），在客户端工作目录的文件浏览器中提供了四种按钮，分别用于下载文件和打开、重命名、删除文件夹。通过这种良好的接口，避免了用户手动输入多条指令来进行文件下载时的麻烦和可能出现的错误，提高了程序的易用性，降低了程序的使用门槛，便于非专业人员操作。
3. 程序具有自动刷新界面的功能，具体来说，程序的窗口类Ui_mainWindow中有myDirChanged和myServerChanged两个成员变量，每当执行创建文件夹、下载文件、上传文件等操作时都会相应地修改这两者的值（如从服务器端下载文件到客户端工作目录下之后，程序会修改myDirChanged为1，保证之后会自动刷新客户端工作目录的文件浏览器）。具体的刷新方式是使用QTimer定时调用update函数，该函数会调用loadServerDir和loadMyDir，loadServerDir会

根据self.serverDirChanged决定是否刷新界面，并在刷新完后将self.serverDirChanged置零（loadMyDir同理）。

4. 客户端程序的提供了良好的debug功能，在命令行内使用python3.6 login.py --port 21启动图形界面程序后，命令行会显示出所有的request和response内容，方便进行debug操作。