

计算机网络 FTP 实验文档

叶宸 软 82 2018013400

1 实现功能

1.1 Server

1. 能够接受并处理 USER, PASS, RETR, STOR, QUIT, SYST, TYPE, PORT, PASV, MKD, CWD, PWD, LIST, RMD, RNFR, RNTD (文档中提到的所有) 指令
2. 支持多客户端, 通过多线程 (pthread) 实现
3. 能够完整传输大文件
4. 能够接受并处理 APPE 及 REST 指令, 即支持断点续传

1.2 Client

1. 基于 PyQt 实现
2. 能够通过 USER, PASS, RETR, STOR, QUIT, SYST, TYPE, PORT, PASV, MKD, CWD, PWD, LIST, RMD, RNFR, RNTD 等 (文档中提到的所有) 指令与标准的 FTP 服务器通信
3. 能够在上一点的基础上, 浏览 FTP 服务器文件, 并完成文件上传、下载等功能, 支持完整的大文件传输
4. 实现了简洁的 GUI 界面

1.2.1 GUI

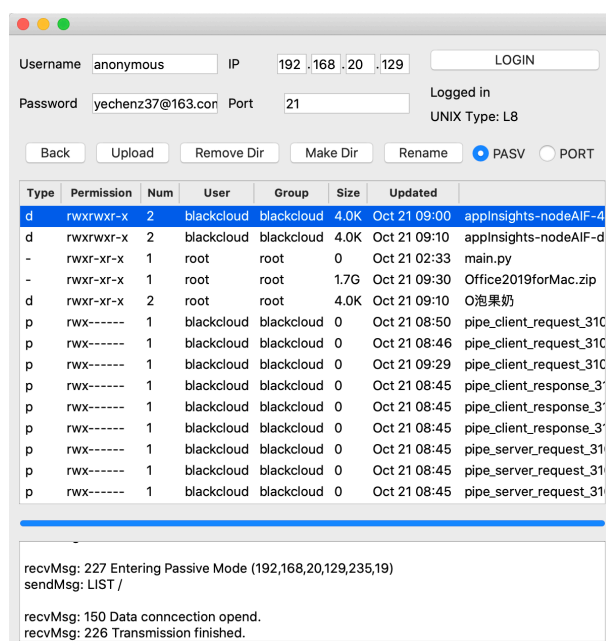
GUI 界面如图 1 所示, 其中, 最上部分的输入框分别用于输入各种必要信息, 在输入合法的 Username, Password 及 IP:Port 之后, 可以点击 Login 按钮登录 FTP 服务器。中间的框显示了文件列表, 其内容按 Unix 的 ls 风格解析向服务器请求 LIST 返回的文件列表得到, 展示了 PWD 下所有的文件 (目录) 信息, 具体信息格式遵循 Unix 的 ls 返回值规范。双击一个文件即可弹出下载框, 指定保存位置, 双击目录则会进入该目录。

点击 Back 可以返回上一级目录 (向服务器发送 CWD ..), 点击 Upload 会弹出文件选择框, 指定要上传的本地文件, **文件将被传到当前显示的目录, 即 PWD 下**, 点击 Remove Dir 可以删除一个目录 (但不能删除文件), 点击 Make Dir 将在当前显示的目录下创建新文件夹, 点击 Rename 可以对当前选中的文件条目做重命名。PASV 和 PORT 按钮用于指定传输模式 (默认为 PASV)。

不支持同时下载和上传及同时下载/上传多个文件, 但客户端不会在下载/上传过程中假死, 而是会在文件列表底部显示一个进度条以指示真实的下载进度。

窗口底部有一个显示客户端通信行为的日志窗口, 方便在遇到问题时查看。

图 1: GUI 界面



2 技术细节及难点

2.1 Server

整体逻辑 主线程监听新的 Connect 到 21(default) 端口的 Tcp 连接请求，有则为它初始化一个 ClientSession 对象，并开辟子线程进入其事件循环。事件循环不断从命令端口阻塞式地读取客户端指令，每条指令经过 parser，判断合法性并调用对应 verb 的 handler，handler 内实现指令的具体逻辑，操作结束后进入读取下一条指令的等待。

Socket 读写 Socket 的读写操作全按阻塞处理。对于指令传输封装了 recvMsg 和 sendMsg 方法，以接受和发送以 <CRLF> 为结尾的 request 及 response。对于数据传输封装了 sendRecvData 方法，在其内部进行数据连接的建立、数据发送与数据连接的关闭。

多客户端支持 一开始听文档说“A single threaded server is entirely adequate if you choose to use the select() call to implement multiple connections on your server.”，便兴致冲冲地用 select 写了第一版，遂发现所有 socket 连接和文件的 Handler 都是 FileDescriptor，需要附加很多信息来在 select 后区分它们以执行不同的操作，导致代码逻辑复杂，分支很多。后来改用多线程，主线程监听指令端口是否有新的客户端连接请求，有则为该客户端单独创建一线程并进行对应的监听处理，与每个客户端的会话信息都保存在一个 ClientSession 结构中，与每个客户端的会话互不干扰，因此在逻辑上清晰许多，同时也避免了全局变量的使用，最终只有一个全局变量 serv 储存了必要的服务器相关的信息，避免了滥用全局变量的污染。

路径管理 用 serv.root 和 session.pwd 两个路径信息管理当前工作路径，前者记录了服务器工作的绝对路径，后者记录这个客户端相对于 root 的工作路径，最终用一个函数 produceAbsPath 组合二者及客户端传来的文件路径信息，生成一个可用的绝对路径，以供具体函数使用。如果路径不合法，则 produceAbsPath 返回负值指示调用者，以进行错误信息的处理，同时 root 的

上级路径不会作为合法路径，以此保证合理的访问权限。

网络字节序 初期测试时，自己的测试脚本无论如何都连不上 server，参考网上实现后发现绑定地址时 IP 和 Port 没有通过对应的函数转义成网络字节序，导致绑定了错误的地址。

获取本机 IP 向任意公网服务器发包后 getsockname 获取 IP，为了稳定发包可以选用一些公共 DNS 服务器。

2.2 Client

整体逻辑 采用 PyQt 开发，程序整体是事件驱动的，点击某个按钮触发某个事件，再调用封装好的函数向服务器发消息即可。消息收发参考 Server，也对发送/接受指令做了封装，同时用装饰器实现了日志 (@message_log)、错误捕获 (@catch_exception) 和建立、销毁数据传输连接 (@need_data_conn) 的逻辑。