

client cmd.py使用说明

运行环境

python3.6

使用方法

前提条件：已启动服务器。

在client/src文件夹，运行

```
1 python cmd.py [ip [port]]
```

例如，当目标服务器的ip地址为127.0.0.1，端口号为2000，运行

```
1 python cmd.py 127.0.0.1 2000
```

此后会进入ftp客户端的交互界面，使用方式非常类似于unix系统下的ftp命令。

一开始会提示你输入用户名和密码，用户名请输入"anonymous",密码可以随意输入。

以下列出所有支持的操作。

open host[port]: 建立指定ftp服务器连接，可指定连接端口

close: 中断与远程服务器的ftp会话(与open对应)

bye: 退出ftp会话过程。

quit: 同bye, 退出ftp会话

cd remote-dir: 进入远程主机目录。

ls[remote-dir][local-file]: 显示远程目录remote-dir，并存入本地文件local-file

mkdir dir-name: 在远程主机中建一目录

rmdir dir-name: 删除远程主机目录

rename[from][to]: 更改远程主机文件名

pwd: 显示远程主机的当前工作目录

get remote-file[local-file]: 将远程主机的文件remote-file传至本地硬盘的local-file

reget remote-file[local-file]: 类似于get，但若local-file存在，则从上次传输中断处续传

put local-file[remote-file]: 将本地文件local-file传送至远程主机

append local-file[remote-file]: 将本地文件追加到远程系统主机，若未指定远程系统文件名，则使用本地文件名

reput local-file[remote-file]: 类似于put, 但若remote-file存在, 则从上次传输中断处续传

delete remote-file: 删除远程主机文件

size file-name: 显示远程主机文件大小

passive: 进入被动传输方式

sendport: 设置PORT命令的使用

system: 显示远程主机的操作系统类型

注1: reput不是标准的ftp命令, 是为了实现文件上传时的断点续传而引入的新命令, 实现方式是通过STAT命令获取远程文件大小。经过实验, 发现向common ftp server发送能返回预期的结果。

注2: reget一个ASCII文件时, 需要common ftp server支持对ASCII的resume transfer。经过实验, 发现向common ftp server发送该类请求会被拒绝。