**基于Python Socket实现的简单的FTP Server**

**服务器实现**

服务端主要由FTPServer, Server, DataSocket, IpListManagement, WebSocket几个类组成。

**Server(WebSocketServer)**

**程序的主要类，负责与单个客户端进行交互，并继承线程，用以实现多线程处理。**

**主结构由一个大的While循环组成，每当收到来自客户端的请求报文时，解析报文，查看报文包含的命令，并依次比对进行处理。**

**由于防火墙问题，不适用PORT传输模式。**

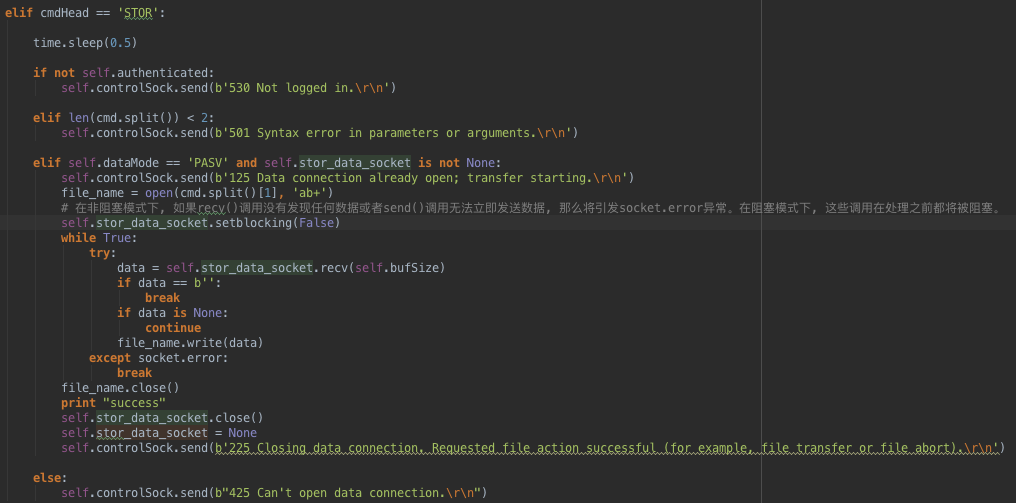
**controlSock对应control socket，用以进行命令的交互。通过PASV命令设置被动数据传输模式，并获取客户端用以连接服务端的地址和接口。**

**Control Socket的相应报文均符合RFC959要求。**

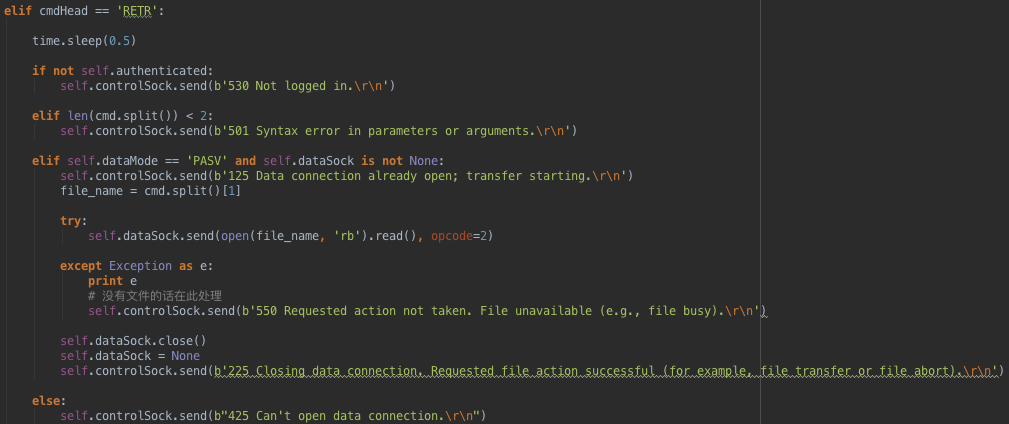
**dataSock对应data socket用以进行数据传输，每次数据传送完成则关闭该socket。**

**dataListenSock 用于监听用户发起的data socket连接。**

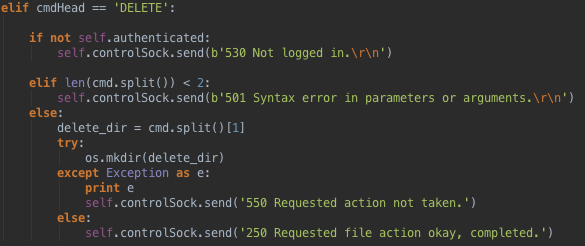
**Server 实现了主要的基础功能包括上传STOR：**

****

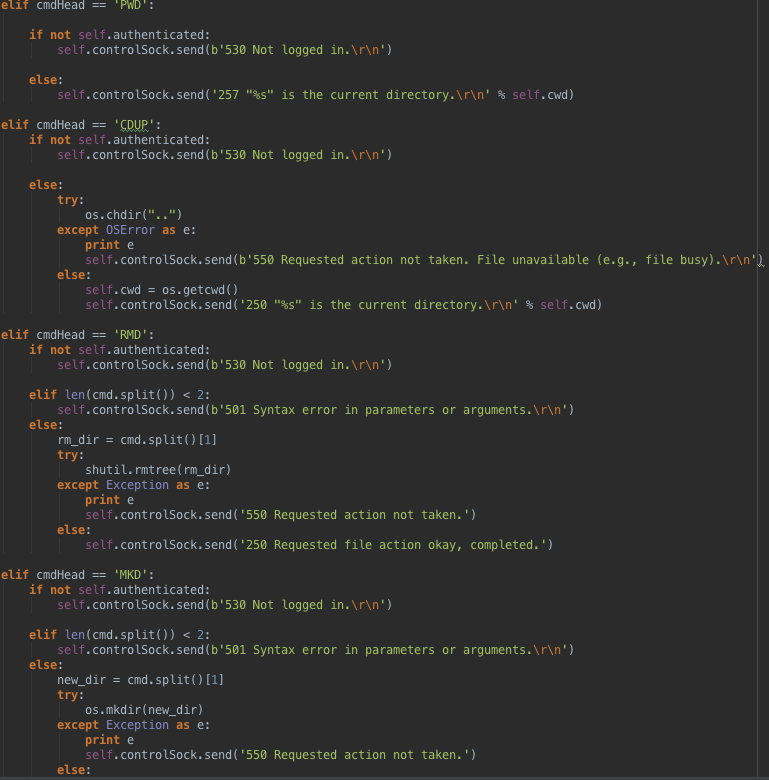
**下载RETR：**

****

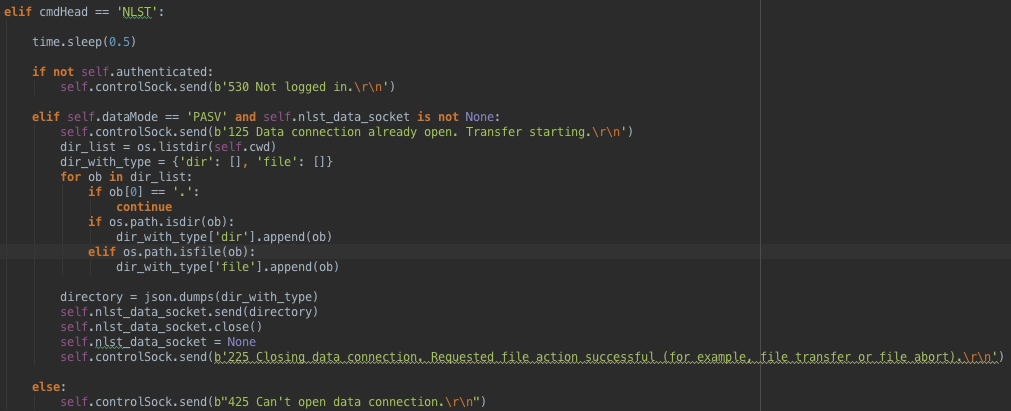
**删除DELETE:**

****

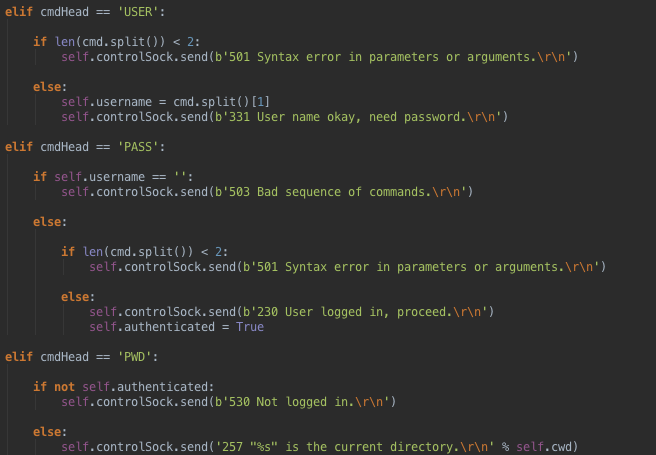
**目录相关操作的PWD, CDUP, RMD, MKD：**

****

**获取文件列表NLST:**

****

**以及用以用户认证的USER和PWD：**

****

**WebSocket是对socket进行的再封装。客户端使用web界面，通过JavaScript的WebSocket与服务器进行交互。**

**FTPServer:**

**对于连接上的客户端通过请求端口判断是来自WebSocket还是来自primitive socket，然后分配给不同的监听socket。**

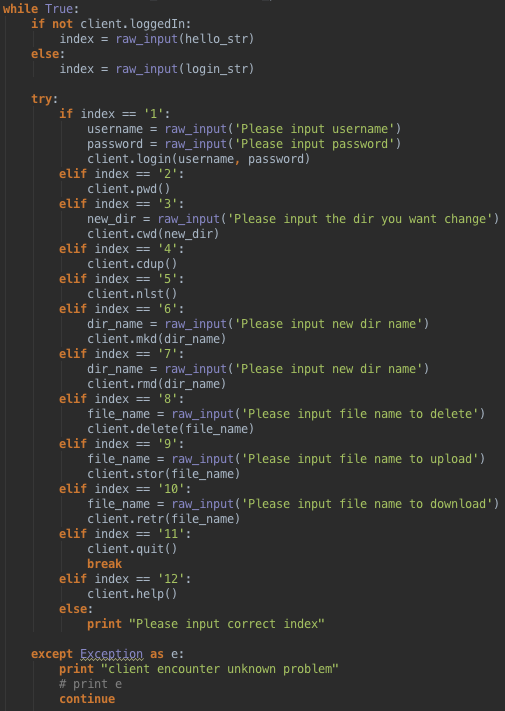
****

**客户端实现**

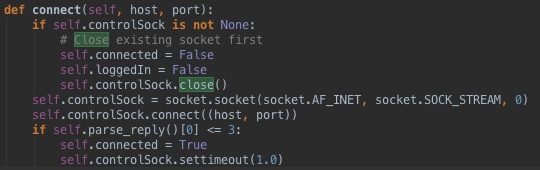
**Primitive Client:**

**通过原生的socket实现的FTPClient。**

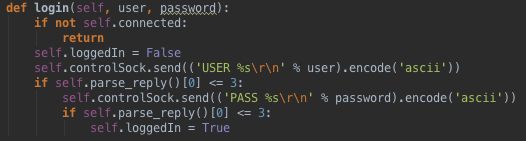
**主程序入口采用简单的列表式功能选择：**

****

**程序运行后首先会调用connect连接服务器：**

****

**选择登陆功能登陆后才可以再选择其他功能：**

****

**详细代码参阅client.py文件**

**Design of Client End**

Using HTML5 WebSocket to communicate with server.

**Methods:**

**updateClick():**

params:

return:

this method is used to dynamic update the click event which is binded on some DOMs like click to download, click to create using ‘on’.

**Log(message):**

params:

message : String

return:

this method is used to log information on console.

**Classes:**

**DataSocket:**

This class encapsulate the primitive WebSocket with some specific attributes so as to handle the data from server.

Attributes:

\_this: this is a various which point to the object pointer

type: this contains the type of usage of this socket like to download or to request the list of file or others.

socket: the websocket of this object.

Methods:

connect(ip, port):

params:

ip: the ip address of the server

port: port of the server socket

return:

This method is used to connect the socket to the server.

onMessage(event):

params:

event: the event of the recv method of the socket

return:

This method is used to handle the data from server.

**Client:**

This class is the main part of our client using a control socket to transfer control signals with server, and four data sockets to transfer data with server. The reason why we use four data sockets is that we can upload file while downloading without confliction.

Attributes:

pwd: this is used to store the current path.

\_this: this is a various which point to the object pointer

control\_socket: to transfer control signals with server

isPasv: distinguish whether the client is request a data socket

currentType: the type of current request data socket

currentFile: the name of current operating file

isTimeout: used to find time out

data\_socket: transfer data with server

re: used to execute regular expression

file\_row: the template of file row in table

fold\_row: the template of folder row in table

ip\_row: the template of ip row in table

username: the name of current user

names: the set of names of current path

ips: all ips

Methods:

connect():

params:

return:

This is used to connect control socket with server

checkTimeout:

params:  
 return:

This is used to find time out

login:

params:  
 return:

This is used to login in

onMessage(event):

params:

event: the event of the recv method of the socket

return:

This method is used to handle the data from server.

doSend(message):

params:

message: String

return:

This is used to send message through control socket.

pasv(type):

params:

type: String , the type of data socket

return:

Used to request a type of socket

nlst():

Show the files and folders in current path

processNlst(data):

params:  
 data: JSON, nlst data from server

Process nlst data from server

upload():

Upload file

sendFile():

Send the binary data of file to server

downloadData(name):

params:

name: String, name of the file

Download file from server

cwd(dir):

params:

dir: String

Change the path to dir

rmd(dir):

params:

dir:String name of the folder

Used to remove folder

delete(dir):

params:

dir: String name of the file

Used to delete file, dir

mkd(dir):

params:

dir: String name of the folder

Used to create folder

quit:

Quit

updateCwd:

used to update the UI which show the current path

checkExsit(name):

params:

name: String name of the file

return: Boolean

Used to check whether the file exist

ip:

Used to get all the ips

processIp(data):

params:

data: JSON the list data from server

Used to process the list data from server

checkExsit(ip):

params:

ip: String name of the ip

return: Boolean

Used to check whether the ip exist

addIp(ip):

params:

ip: String name of the ip

Used to add new ip

deleteIp(id):

params:

id: String id of the ip

Used to delete ip

modifyIp(id, ip):

params:

id: String id of the ip

ip: new ip of the ip

Used to update ip

**参考资料**

**RFC959 FTP protocol 文档**

**WebSocket 协议文档**