# 基于Python编写设计的多线程web服务器

一、项目概述

该项目开发了一个利用python编写的web多线程服务器, 主要是利用python引入线程模块和socket模块并用以编写服务器从而能处理简单的web 页面请求，该服务器能够接收消息,经过分析后构造响应消息,然后交付给浏览器显示。

二、项目实现的具体功能

1、向服务器请求指定文件，从服务器文件系统读取请求的文件，并创建一个由请求的文件的html内容组成的HTTP响应报文发送至客户端浏览器并显示出来

2、当多个客户端访问服务器时，都可以并发地访问，并且服务器为每个客户端联系时均建立了连接套接字

三、项目功能实现的主要方法

1、当客户端和服务端建立连接成功，开启一个单线程，使用子线程专门处理客户端的请求直到连接成功，开始执行该线程，防止主线程阻塞。

print("waiting for connection......\n")

while True:

ClientSock, addr = SeverSocket.accept()

thread = threading.Thread(target=Server, args=(ClientSock, addr))

thread.start() # 开始执行该线程

2、当寻找到指定文件时，将指定文件反馈给服务器并发送至客户端

if os.path.exists(direction) and direction.endswith(".html"):

file=open(direction,encoding="utf-8") # 打开路径中的文件

SUCCESS\_PAGE = "HTTP/1.1 200 OK\r\n\r\n" + file.read()

print(SUCCESS\_PAGE)

ClientSock.sendall(SUCCESS\_PAGE.encode())

ClientSock.close()

3、针对需要使用线程而利用socket编写的函数

(1)def Server(ClientSock, addr):

BUFSIZE = 1024 # 将缓冲区大小设置为 1KB

print('connected from:', addr)

recData = ClientSock.recv(BUFSIZE)

data = recData.decode()

(2)HOST = ""

PORT = 4004

ADDR = (HOST, PORT) # 地址 = IP + 端口号

SeverSocket = socket(AF\_INET, SOCK\_STREAM)

SeverSocket.bind(ADDR)

四、开发环境

此系统使用python开发，在Windows10系统下运行，使用前应安装好python，并且配置相关环境变量。

五、运行结果

见软件运行说明文档

六、项目完整代码

见server.py文件