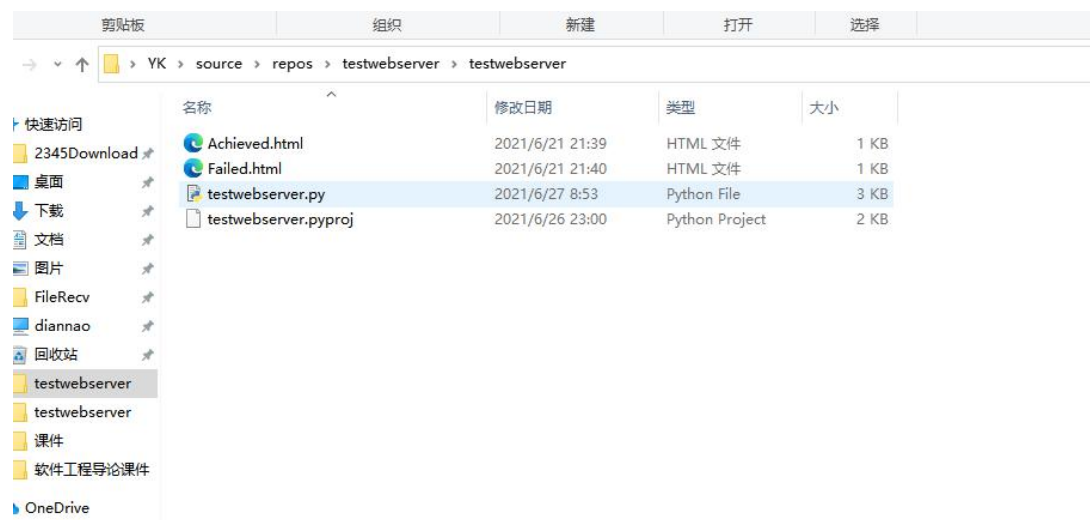


## 一. 功能说明:

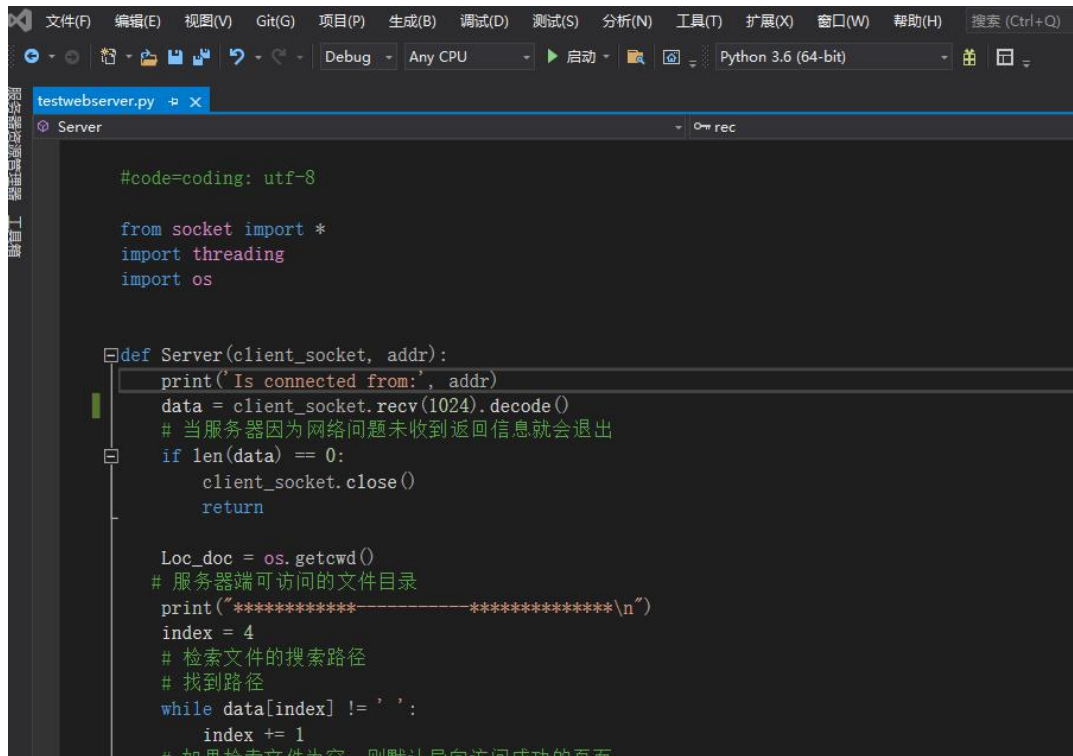
1. 向 **web** 服务器请求特定文件，从服务器文件系统读取请求的文件并创建一个由请求的文件的 **html** 内容组成的 **HTTP** 响应报文反馈给浏览器并显示出来。
2. 对意外情况的反馈：断网时显示连接失败信息、访问不存在的文件时会提示失败信息。
3. 多线程：多个客户端访问 **web** 服务器时，可以并发地访问，通过后台观察打印出的地址可以发现服务器已经接收到了多个客户端的访问（在开发过程文档可以看到运行结果截图），并且 **web** 服务器为每个客户端联系时均建立了连接套接字。

## 二. 使用步骤:

1. 在一台主机的程序中运行项目，在项目文档中加入 **Achieved.html** 和失败页面 **fail.html**



2. 打开项目运行



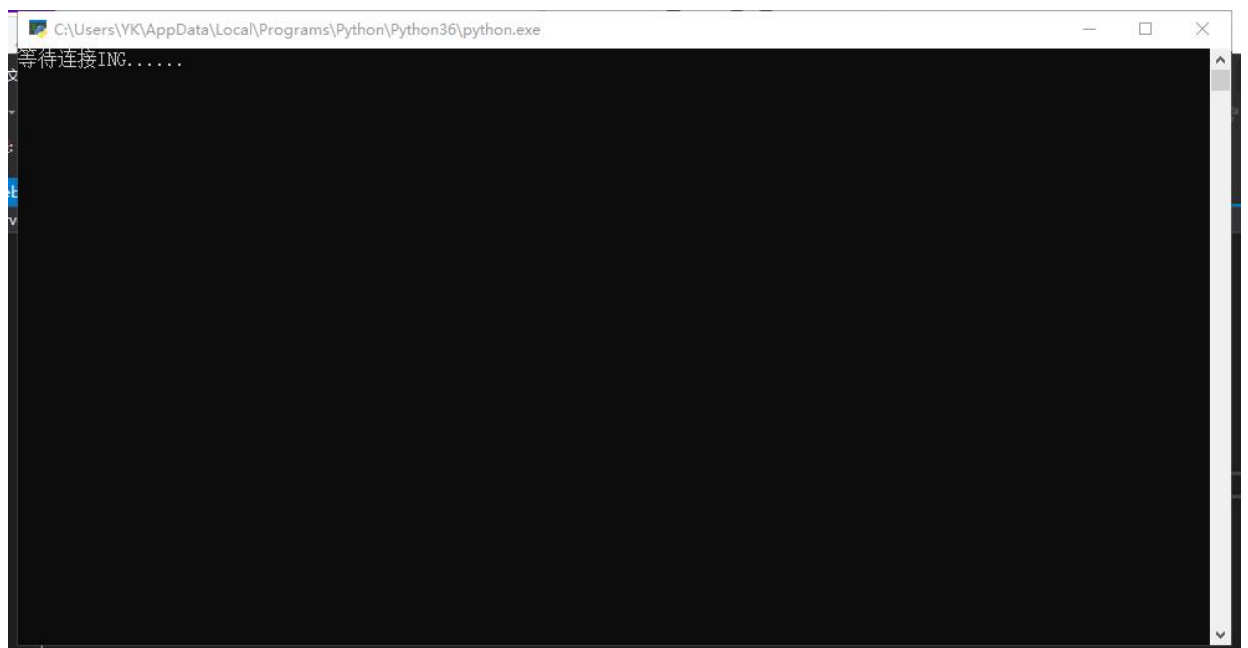
```
#code=coding: utf-8

from socket import *
import threading
import os

def Server(client_socket, addr):
    print('Is connected from:', addr)
    data = client_socket.recv(1024).decode()
    # 当服务器因为网络问题未收到返回信息就会退出
    if len(data) == 0:
        client_socket.close()
        return

    Loc_doc = os.getcwd()
    # 服务器端可访问的文件目录
    print("*****-----*****\n")
    index = 4
    # 检索文件的搜索路径
    # 找到路径
    while data[index] != ' ':
        index += 1
    # 如果检索文件为空 则默认导向访问成功的页面
```

### 3. 等待连接



### 4. 在浏览器输入 <http://localhost:2016/>，在服务器端查看连接信息

```
选择C:\Users\YK\AppData\Local\Programs\Python\Python36\python.exe
等待连接ING.....
Is connected from: ('127.0.0.1', 52677)
*****-----*****
Is connected from: ('127.0.0.1', 50797)
HTTP/1.1 200 Found successfully!

<!DOCTYPE html>
<html>
<head>
  <title></title>
<meta http-equiv="content-type" content="text/html"; charset="utf-8"/>
</head>
<body>
</br>
<center><h1>Successsul connecting!!!</h1></center>
</br>
</body>
</html>
Is connected from: ('127.0.0.1', 59788)
```

**Successsul connecting!!!**