**题目：HTTP协议设计与实现**

**姓名: 王翰墨**

**学号：2019302683**

**班号：10011902**

**课程代号：网络试点班**

**计算机学院**

**时间:2022年1月8日**

**目 录**

摘 要

[1 目的 1](#_Toc341302685)

[2 要求 1](#_Toc341302686)

[3 相关知识 2](#_Toc341302687)

[4 实现原理 4](#_Toc341302688)

[5 运行结果与分析 7](#_Toc341302690)

[6 参考文献 13](#_Toc341302690)

# 题目：HTTP协议设计与实现

# 目的

在网络协议栈中对HTTP协议的支持，并利用HTTP协议实现一个WEB浏览器。

# 2、要求

实验报告包括两部分内容：①是完善自己先前五次实验的网络协议栈，加入HTTP协议支持（后面用my\_netstack表示）；②是利用参考代码构建Web服务器和客户端具有图形界面的浏览器（后面用web\_netstack表示）。

①my\_netstack:在前五次课程所搭建的网络协议栈中，加入HTTP协议的客户端和服务器端功能的代码，使其支持HTTP1.0的请求和应答，从而实现传输html文件。网络协议栈包括Ethernet协议、IPv4协议、ARP协议、ICMP协议、UDP协议、TCP协议和HTTP协议，分为客户端和服务器两台端设备进行通讯。

②web\_netstack:在课程资料给出的参考代码基础上，在Linux系统上部署tiny服务器，在Windows系统上部署Web客户端，实现浏览器的图形界面，并利用浏览器向服务器发送http请求，包括静态网页和动态网页，服务器端发送http应答，并在客户端浏览器显示html文件页面，实现C/S模式的通信。

# 3、相关知识

**统一资源定位符 URL ：**可以唯一的标识因特网上任一网络资源（位置+名称）和访问方法。

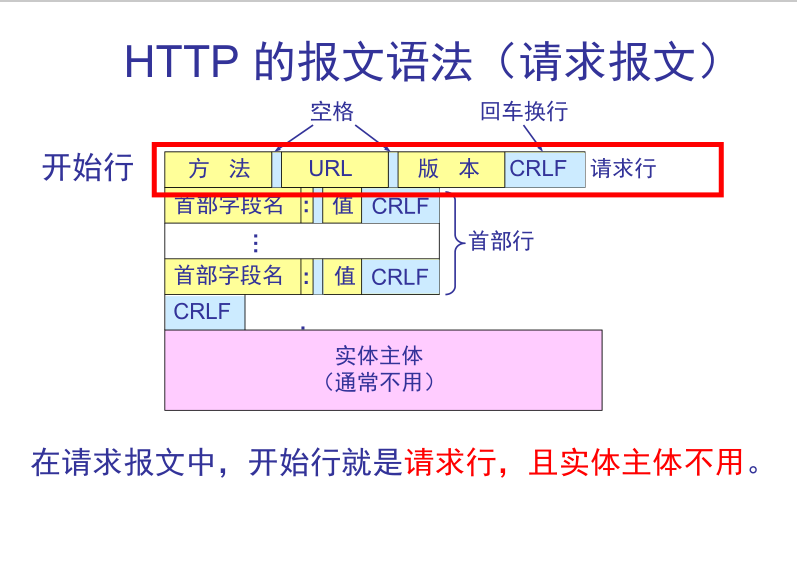
URL 给资源（位置+名称）提供一种抽象的标记和识别方法，用该方法可对资源定位。

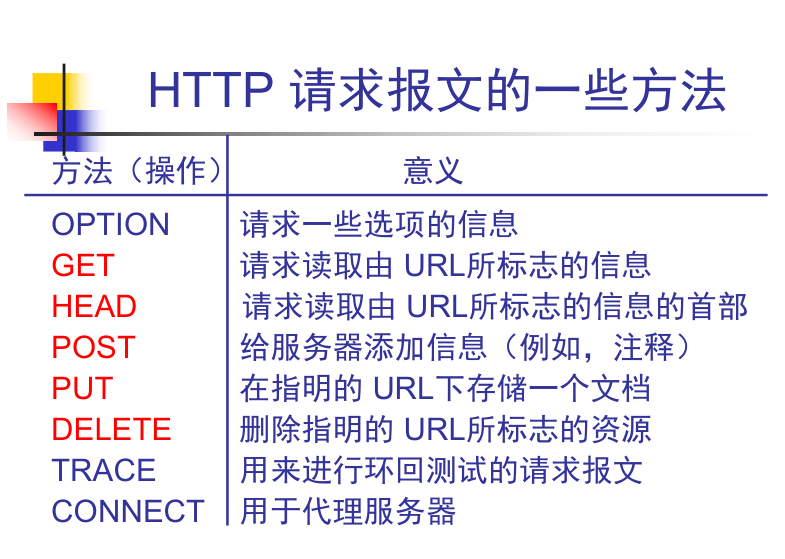
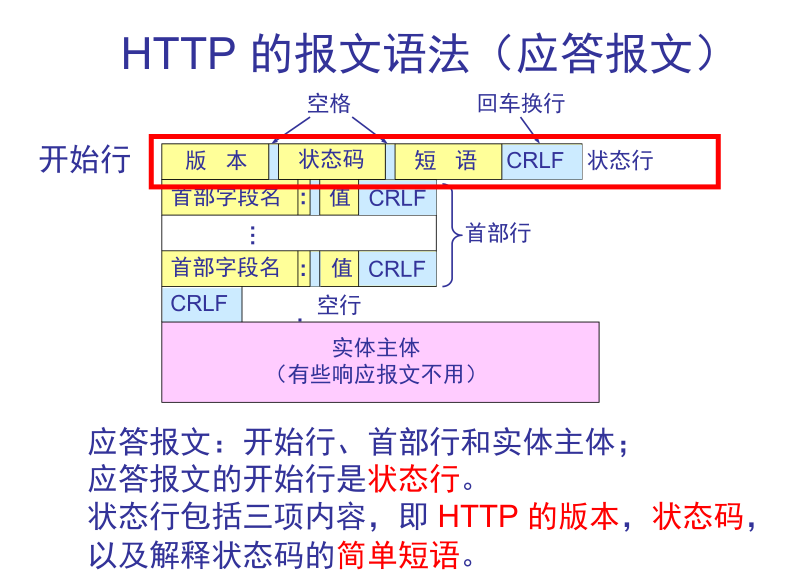
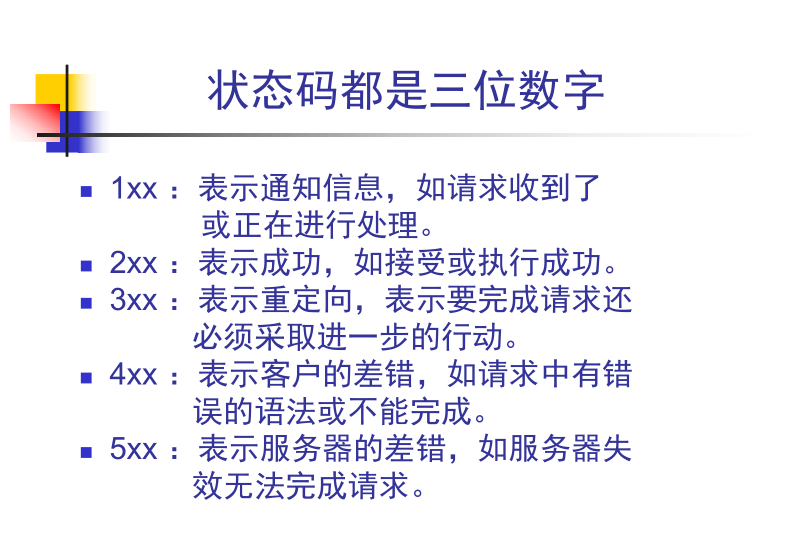
只要能够对资源定位，系统就可以对资源进行各种操作：存取、更新、替换和查找其属性。

URL 相当于一个文件名在网络范围的扩展。

使用 HTTP 的 URL 的一般形式：

http://<主机>:<端口>/<路径+名称>



# 4、实现原理

**①my\_netstack**

**总体构思：**如下图所示，不改变Ethernet和IPv4协议的四线程生产者-消费者处理流程，在应用层的HTTP协议直接调用传输层UDP协议的接口函数进行通信。该网络协议栈支持Ethernet、IPv4、ARP、ICMP、UDP、TCP和HTTP。



**实现HTTP的思路：** 在客户端，首先对URL进行解析，获得主机域名、端口号和文件名三个部分的字符串，再通过解析域名取得IP地址。创建socket套接字后，对服务器地址和端口进行connect，下一步构造HTTP的GET请求报文，并通过send函数发送，用recv函数等待接收应答，获得HTTP应答报文后，解析并将报文打印出来，用close结束链接并释放资源。如果数据中还引用到了其他资源，如bmp位图，则再次构造GET请求，如此循环，直到全部资源获取完毕，将文件保存在本地目录中。

在服务器端，建立socket套接字并用bind绑定服务器地址和端口，用listen和accept函数建立TCP连接，在recv函数收到HTTP的GET请求报文后，解析客户端需要的文件名并读取该文件。如果没有该文件，则构造一个应答报文，代码为404；如果找到该文件，则将其数据写入应答报文的数据部分，然后用send发送给客户端，随后收到客户端挥手并close释放链接，服务器端一直循环工作。

**②Web\_netstack**

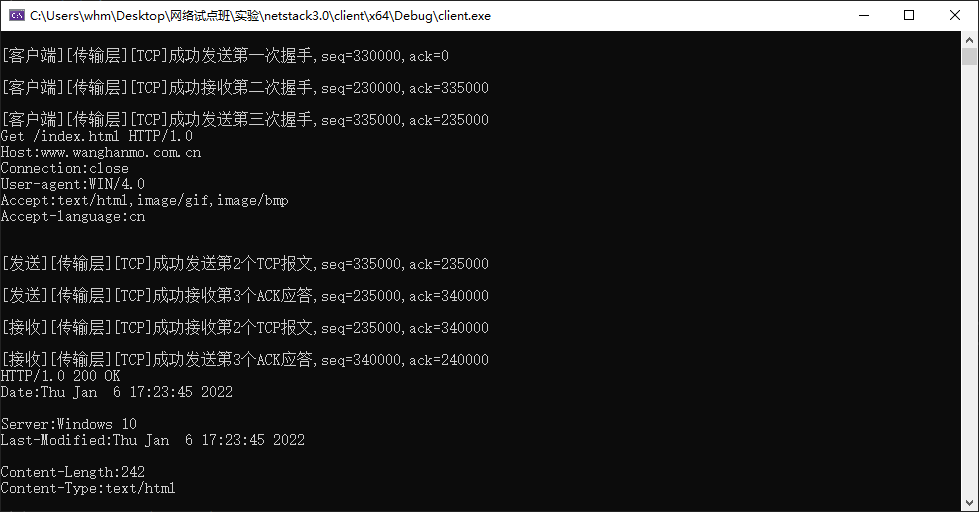
**服务器：**服务器使用tiny服务器源代码，部署在Unbuntu16.04系统上，编译后使用命令 “./tiny 端口号”启动服务器运行。在tiny根目录下放置静态网页index.html，在/cgi-bin/目录下放置adder.c，编译后为adder，实现动态网页，使用方法是请求文件/cgi-bin/adder?A&B，其中A和B是两个数字，网页会返回两数之和的html文件，如果请求的文件未找到，则会构造404代码的应答报文并发送。

**客户端：**客户端部署在Windows10系统上，浏览器图形界面使用Visual Studio 2022中的C++ MFC应用程序框架开发。图形界面包括输入网址的编辑框、浏览的按钮和显示网页的WebBrowser组件。在输入网址并点击浏览后，客户端代码会解析URL中的主机、端口和文件名，随后同上建立TCP链接并发送HTTP的GET请求，收到应答报文后将其解析并将数据部分保存在本地的index.html文件中，随后释放链接完成通讯，并在WebBrowser组件中显示本地的index.html文件，实现用户浏览网页的功能。

# 5、运行结果与分析

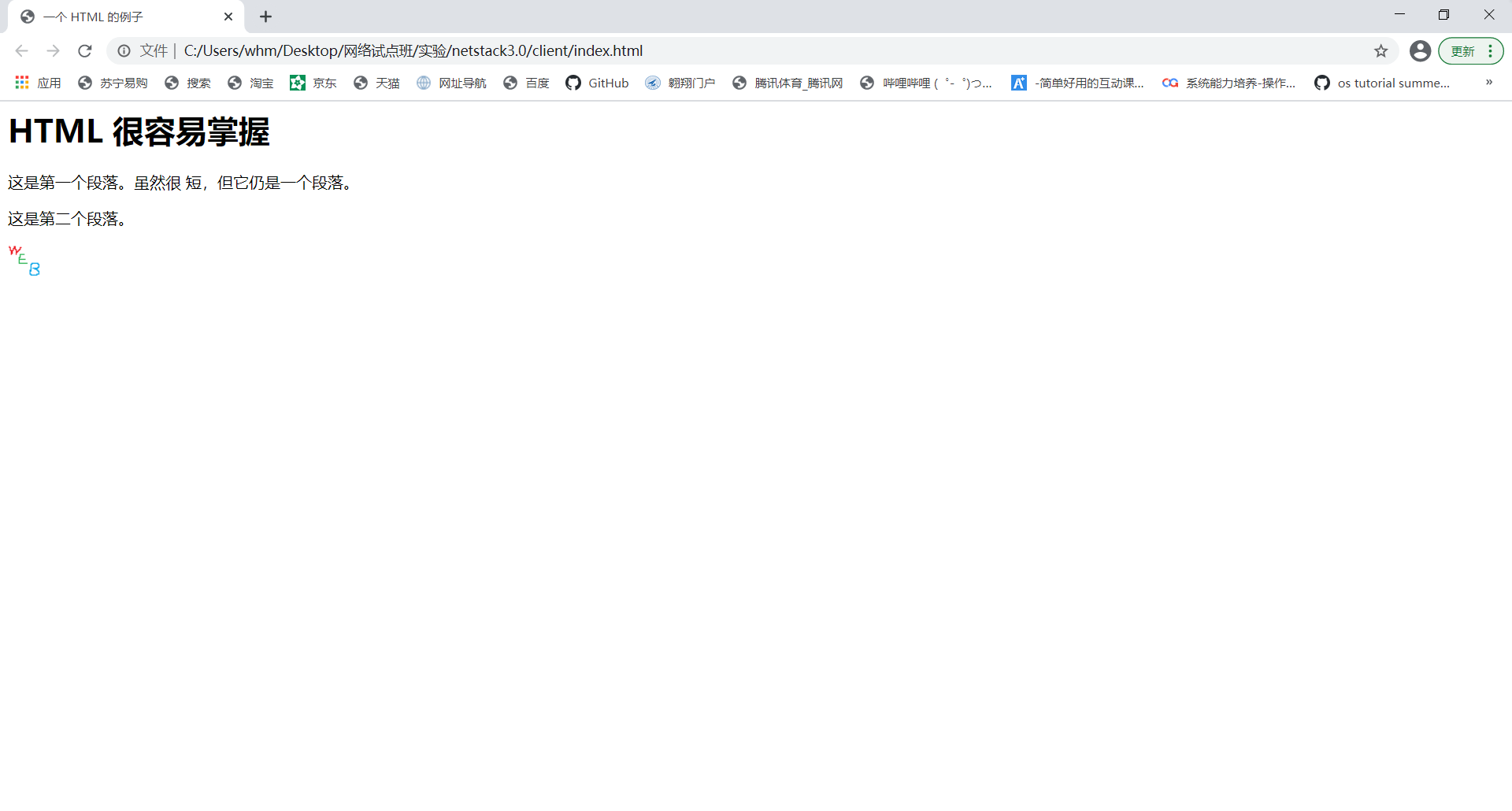
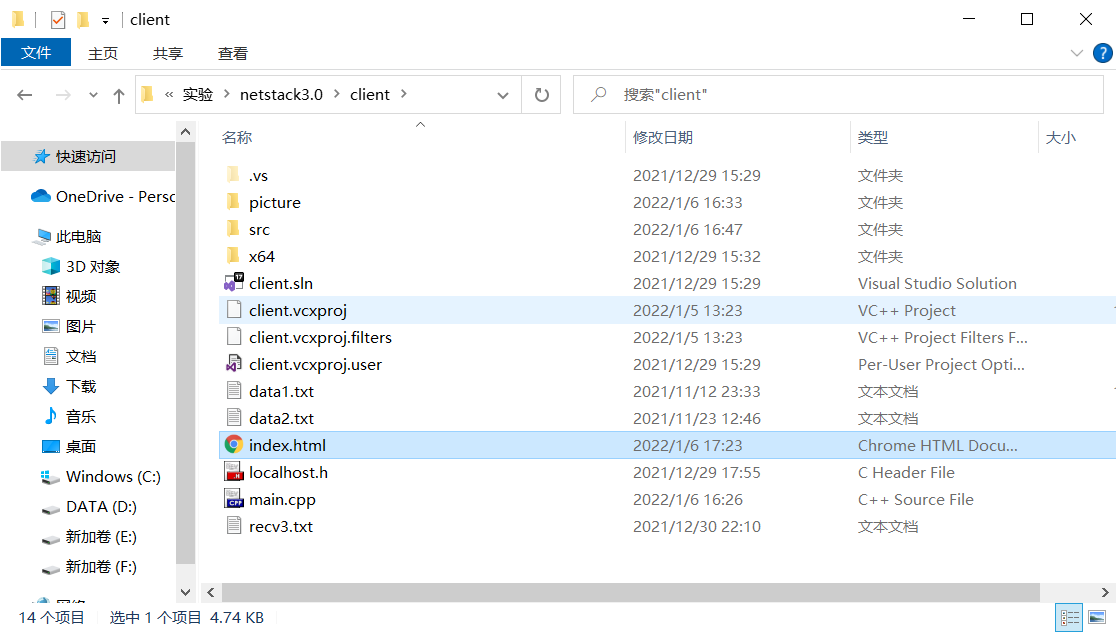
**①my\_netstack**

**客户端：**先建立TCP链接，随后构造HTTP的Get请求报文，并通过TCP发送出去，然后等到TCP接收到应答，解析出HTTP应答报文并显示出来，随后挥手释放TCP链接。

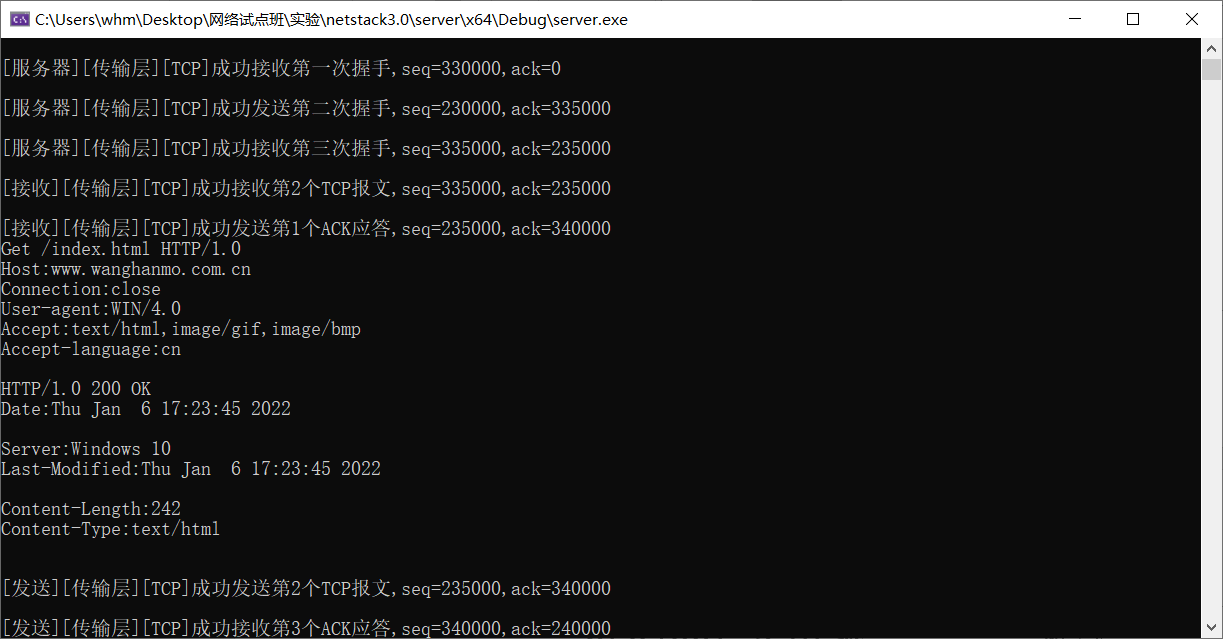


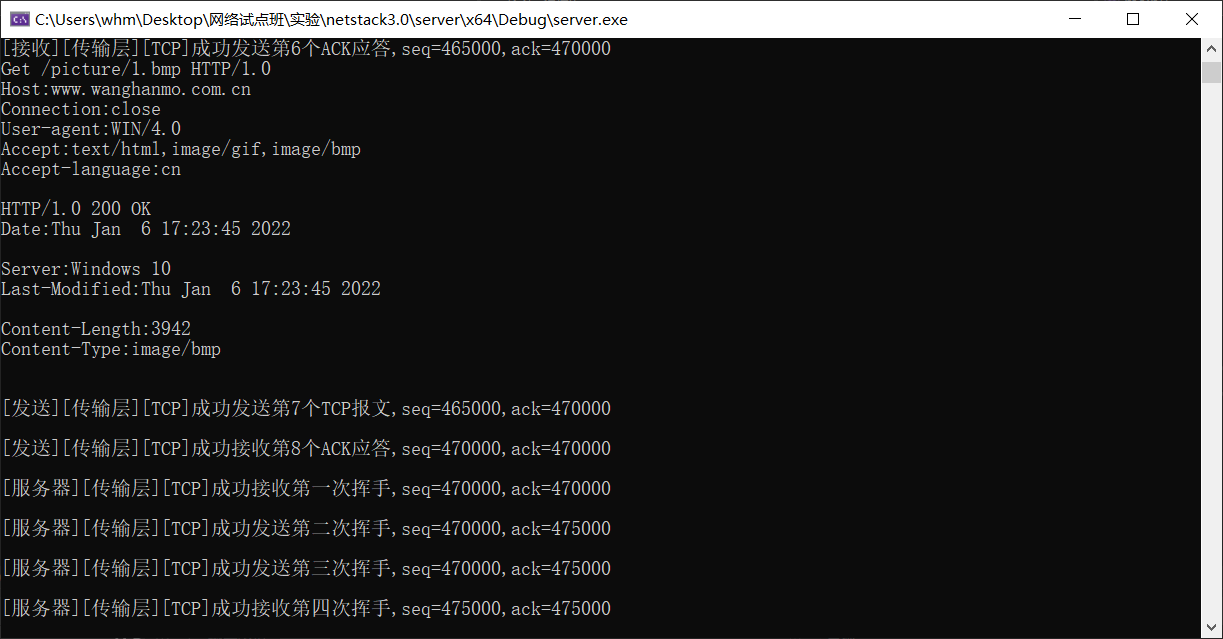


在客户端本地保存了接收到的文件，用浏览器可正常打开，效果如下图所示，传输的文件包括一个html文件和一个在/picture/目录下的1.bmp位图：



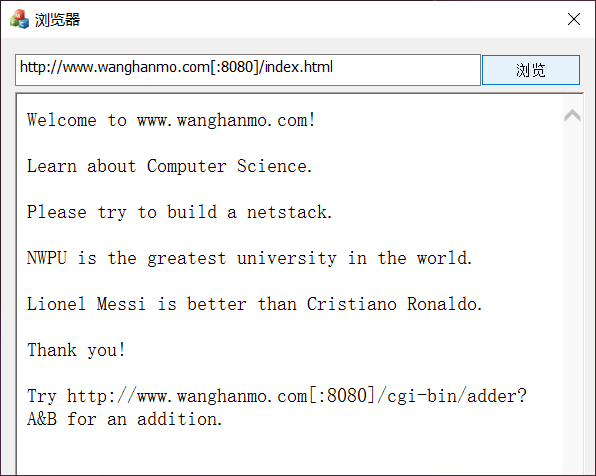
**服务器：**接收到客户端请求后先建立TCP链接，收到客户端HTTP报文后解析，并将客户端请求的文件读取到数据部分，构造HTTP应答报文并发送出去，最后挥手释放TCP链接，如此循环不停地工作。

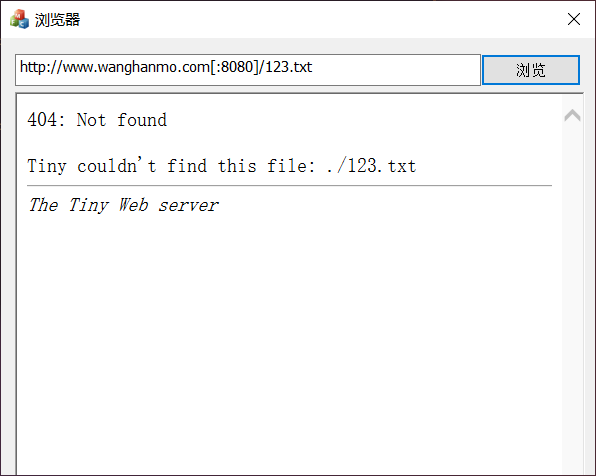
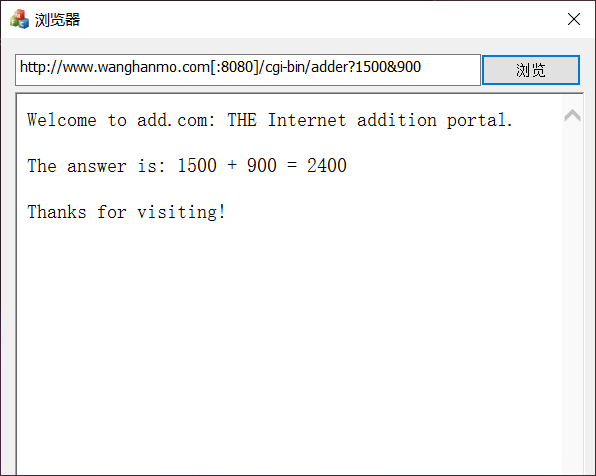




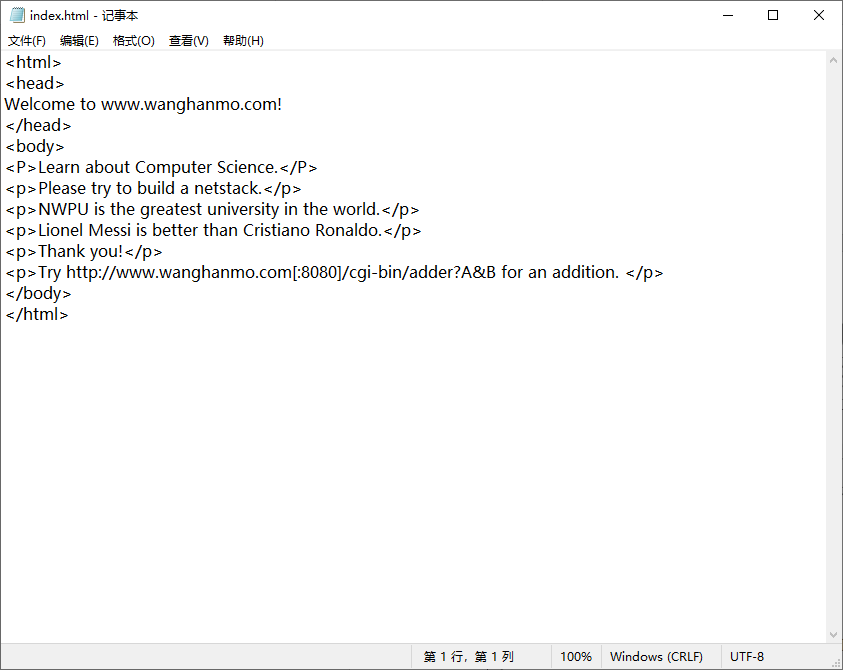
**②web\_browser**

**客户端：**打开x64/bebug/browser.exe启动浏览器界面，输入框输入服务器域名和端口，后面分别加上静态网页、动态网页和不存在的网页，浏览效果如下所示：

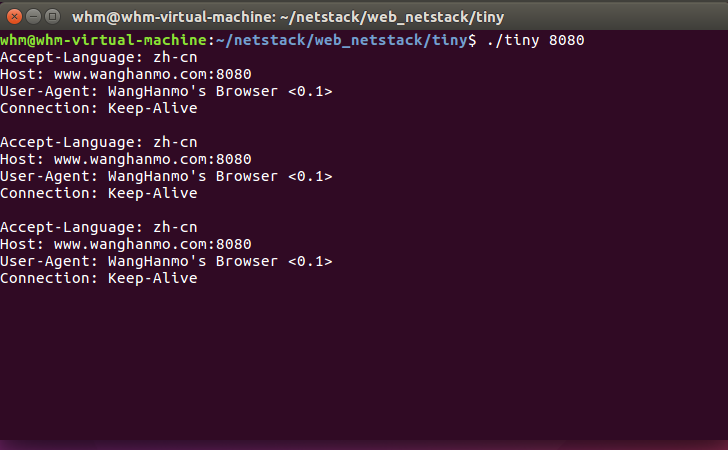




在客户端本地会保存接收到数据文件，其内容如下图所示：



服务器：在ubuntu虚拟机上运行tiny服务器程序，在接收到客户端浏览器的报文后会在屏幕上打印解析的请求报文信息，并根据请求内容发送相应的HTTP应答报文。



# 6、参考文献

* 《计算机网络协议分析与实践》姚烨,朱怡安 电子工业出版社