

数据通信作业

姓名: 刘浩文 学号: 517021911065 日期: 2020/5/8

一、 实验名称及内容

名称: 利用 Winsock 搭建 web 服务器

内容: 1、利用 Winsock 搭建 web 客户端、服务端，使服务端能接收处理一次客户端的 HTTP 请求并返回 HTML。服务端接收并打印从客户端收到的 HTTP 请求信息。客户端发送 HTTP 请求信息给服务端，并接收服务端返回的 HTML。之后用浏览器代替客户端访问服务器，观察浏览器是否能收到服务器返回的 HTML 并解析。

2、利用 Winsock 搭建 web 服务端，使服务端能接收处理多次客户端的 HTTP 请求，并能返回 GIF、TXT 等格式的文件。服务端接收并打印从客户端（浏览器）收到的 HTTP 请求信息。用浏览器代替客户端访问服务器，观察浏览器是否能收到服务器返回的 GIF、TXT 格式的文件并成功解析。自己还结合 web 设计进行了一些有趣的尝试。

二、 实验过程和结果

物理主机系统: macOS Catalina 10.15.4

虚拟机系统: Windows 10 专业版 x64； 计算机名: 691B

虚拟机软件: Parallels Desktop 15 for Mac Pro Edition, version 15.1.4 (47270)

浏览器: Edge

编程环境 (IDE): Visual Studio 2019

- **单次处理 web 服务器:**

- **客户端:**

1. 调用 WSAStartup 函数，初始化 winsock。
2. 调用 socket() 函数创建一个 Socket
3. 调用 connect() 函数和服务器进行连接
4. 调用 send() 和 recv() 函数向服务器收发消息 (HTTP request message)。
5. 调用 shutdown(), closesocket(), WSACleanup() 函数断开与服务器的连接，关闭 socket。

- **服务端:**

1. 调用 WSAStartup 函数，初始化 winsock。
2. 调用 socket() 函数创建一个 Socket
3. 调用 bind() 函数和 socket 绑定
4. 调用 listen() 函数监听一个 socket
5. 调用 accept() 函数接受由一个客户端发起的连接

6. 调用 send() 和 recv() 函数向客户端返回 HTML 消息。
 7. 调用 shutdown(), closesocket() 函数断开与服务器的连接，关闭 socket。
- 多线程处理 web 服务器：
 - 服务端：
 1. 调用 WSAStartup 函数，初始化 winsock。
 2. 调用 socket() 函数创建一个 Socket
 3. 调用 bind() 函数和 socket 绑定
 4. 调用 listen() 函数监听一个 socket
 5. 调用 accept() 函数接受由一个客户端发起的连接
 6. 为该连接创建一个线程来处理与对应客户端有关的服务
 7. 调用 send() 和 recv() 函数向客户端发送 HTML 消息或 GIF、TXT 格式的文件。
 8. 客户端断开连接后，调用 _endthread() 函数关闭线程
 9. 调用 shutdown(), closesocket() 函数断开与服务器的连接，关闭 socket

- 运行成功：

- 简单 web 服务器：左边窗口将运行客户端程序，右边窗口将运行简单 web 服务端程序。

```

Windows 10 on Mac
C:\Windows\system32\cmd.exe
Connection closed
C:\Users\hongxing\source\repos\webClient\x64\Debug>

C:\Windows\system32\cmd.exe
C:\Users\hongxing\source\repos\webServer\x64\Debug>cd x64\Debug
C:\Users\hongxing\source\repos\webServer\x64\Debug>dir
驱动器 C 中的卷没有标签。
卷的序列号是 2478-674A
C:\Users\hongxing\source\repos\webServer\x64\Debug 的目录
2020/05/06 09:12 <DIR> .
2020/05/06 09:12 <DIR> ..
2020/05/06 09:58 63,488 webServer.exe
2020/05/06 09:58 462,892 webServer.ilk
2020/05/06 09:58 552,960 webServer.pdb
3 个文件 1,079,340 字节
2 个目录 222,929,121,280 可用字节

C:\Users\hongxing\source\repos\webServer\x64\Debug>webServer.exe
Server listening ...
GET / HTTP/1.0
Bytes sent: 131
C:\Users\hongxing\source\repos\webServer\x64\Debug>webServer.exe
Server listening ...
GET / HTTP/1.0
Bytes sent: 131
C:\Users\hongxing\source\repos\webServer\x64\Debug>

```

1. 运行客户端程序访问 www.sjtu.edu.cn:

The screenshot shows two windows running on a Mac system with a Windows 10 interface. Both windows are titled 'Windows 10 on Mac' and show command-line output from 'cmd.exe'.
Left window (Client):
C:\Users\hongxing\source/repos\webClient\x64\Debug>webClient.exe www.sjtu.edu.cn
Bytes Sent: 16
recv failed: 10060
C:\Users\hongxing\source/repos\webClient\x64\Debug>
C:\Users\hongxing\source/repos\webClient\x64\Debug>
Right window (Server):
C:\Users\hongxing\source/repos\webServer\x64\Debug>cd x64\Debug
C:\Users\hongxing\source/repos\webServer\x64\Debug>dir
驱动器 C 中的卷没有标签。
卷的序列号是: 2478-67AA
C:\Users\hongxing\source/repos\webServer\x64\Debug 的目录
2020/05/06 09:12 <DIR> .
2020/05/06 09:12 <DIR> ..
2020/05/06 09:58 63,488 webServer.exe
2020/05/06 09:58 462,892 webServer.ilk
2020/05/06 09:58 552,960 webServer.pdb
2 个文件 1,109,340 字节
2 个目录 222,929,121,230 可用字节
C:\Users\hongxing\source/repos\webServer\x64\Debug>webServer.exe
Server listening ...
GET / HTTP/1.0
Bytes sent: 131
C:\Users\hongxing\source/repos\webServer\x64\Debug>

出现和老师上课时一样的问题，无法访问。于是尝试测试一下是什么原因：

- 1、将客户端发送的信息改为“`GET / HTTP/1.1\n\n`”，发现客户端程序没有收到任何信息就关闭了连接：

The screenshot shows two windows running on a Mac system with a Windows 10 interface. Both windows are titled 'Windows 10 on Mac' and show command-line output from 'cmd.exe'.
Left window (Client):
C:\Users\hongxing\source/repos\webClient\x64\Debug>webClient.exe www.sjtu.edu.cn
Bytes Sent: 16
Connection closed
C:\Users\hongxing\source/repos\webClient\x64\Debug>
Right window (Server):
C:\Users\hongxing\source/repos\webServer\x64\Debug>cd x64\Debug
C:\Users\hongxing\source/repos\webServer\x64\Debug>dir
驱动器 C 中的卷没有标签。
卷的序列号是: 2478-67AA
C:\Users\hongxing\source/repos\webServer\x64\Debug 的目录
2020/05/06 09:12 <DIR> .
2020/05/06 09:12 <DIR> ..
2020/05/06 09:58 63,488 webServer.exe
2020/05/06 09:58 462,892 webServer.ilk
2020/05/06 09:58 552,960 webServer.pdb
3 个文件 1,079,340 字节
2 个目录 222,929,121,230 可用字节
C:\Users\hongxing\source/repos\webServer\x64\Debug>webServer.exe
Server listening ...
GET / HTTP/1.0
Bytes sent: 131
C:\Users\hongxing\source/repos\webServer\x64\Debug>

- 2、将客户端发送的信息改为“`GET / HTTP/2.0\n\n`”，发现客户端程序大多数情况下没有收到任何信息就关闭了连接，但有一次返回了显示 400 Bad Request 的 HTML 代码：

```

Windows 10 on Mac

C:\Windows\system32\cmd.exe
C:\Users\hongxing\source\repos\webClient\x64\Debug>webClient.exe www.sjtu.edu.cn
Bytes Sent: 16
'C
C:\Users\hongxing\source\repos\webClient\x64\Debug>webClient.exe www.sjtu.edu.cn
Bytes Sent: 16
Bytes Received: 392
HTTP/1.1 400 Bad Request
Date: Fri, 08 May 2020 01:30:17 GMT
Server: Yxlink
Content-length: 226
Connection: close
Content-Type: text/html; charset=iso-8859-1

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
<title>Bad Request</title>
</head><body>


Your browser sent a request that this server could not understand.<br />


</body></html>
Connection closed
C:\Users\hongxing\source\repos\webClient\x64\Debug>webClient.exe www.sjtu.edu.cn
Bytes Sent: 16
Connection closed

C:\Windows\system32\cmd.exe
C:\Users\hongxing\source\repos\webServer\x64\Debug>cd x64\Debug
驱动器 C 中的卷没有标签。
卷的序列号是: 2478-67AA
C:\Users\hongxing\source\repos\webServer\x64\Debug 的目录
2020/05/06 09:12 <DIR> .
2020/05/06 09:12 <DIR> ..
2020/05/06 09:58 63,488 webServer.exe
2020/05/06 09:58 462,892 webServer.ilk
2020/05/06 09:58 552,960 webServer.pdb
2 个文件 1,103,340 字节
2 目录 222,929,121,280 可用字节
C:\Users\hongxing\source\repos\webServer\x64\Debug>webServer.exe
Server listening ...
GET / HTTP/1.0
Bytes sent: 131
C:\Users\hongxing\source\repos\webServer\x64\Debug>webServer.exe
Server listening ...
GET / HTTP/1.0
Bytes sent: 131
C:\Users\hongxing\source\repos\webServer\x64\Debug>

```

推测可能是 www.sjtu.edu.cn 不支持 HTTP1.1 和 HTTP2.0。但为何 HTTP1.0 时迟迟无法访问，根据同学所说查了 Host 域的相关资料，也根据资料改了发送的信息，但并无效果。由于时间不够也没有多做尝试。

2. 运行客户端程序访问 www.baidu.com:

```

Windows 10 on Mac

C:\Windows\system32\cmd.exe
C:\Users\hongxing\source\repos\webClient\x64\Debug>webClient.exe www.baidu.com
Bytes Sent: 16
Bytes received: 512
HTTP/1.1 200 OK
Accept-Ranges: bytes
Cache-Control: no-cache
Connection: keep-alive
Content-Length: 1405
Content-Type: text/html
Date: Fri, 08 May 2020 02:56:46 GMT
P3p: CP="OTI DSP COR IVA OUR IND COM "
P3p: CP="OTI DSP COR IVA OUR IND COM "
Pragma: no-cache
Server: Baidu/1.1
Set-Cookie: BAIDUID=60B3946E197BECEEEEEE1DA97BB88B9;PG=1; expires=Thu, 31-Dec-37 23:55:55
Gmt; max-age=2147483647; path=/; domain=.baidu.com
Set-Cookie: BIDV=096394681978CEEEEEE1DA97B788B9; expires=Thu, 31-Dec-37 23:55:55
Gmt; max-age=2147483647; path=/; domain=.baidu.com
Set-Cookie: PSTM=1588906060; expires=Thu, 31-Dec-37 23:55:55 GMT; max-age=2147483647; path=/
; domain=.baidu.com
Set-Cookie: BIDU=0963946E1978CEEEEEE1DA97BB88B9; expires=Thu, 31-Dec-37 23:55:55 GMT; max-age=31536000; expires=Sat,
01-Jan-2038 00:00:00 GMT; domain=.baidu.com; path=/; version=1; comment=bdt
Traceid: 158890606054957378113756953927782919
Vary: Accept-Encoding
X-UA-Compatible: IE-Edge,chrome=1
<!DOCTYPE html>|- STATUS OK-->
<html>
<head>
<meta http-equiv="content-type" content="text/html; charset=utf-8">
<meta http-equiv="X-UA-Compatible" content="IE-Edge">
<link rel="dns-prefetch" href="//sl.bdstatic.com/">
<link rel="dns-prefetch" href="//t1.baidu.com/">
<link rel="dns-prefetch" href="//t2.baidu.com/">
<link rel="dns-prefetch" href="//t3.baidu.com/">
<link rel="dns-prefetch" href="//t4.baidu.com/">
<link rel="dns-prefetch" href="//t11.baidu.com/">
<link rel="dns-prefetch" href="//t12.baidu.com/">
<link rel="dns-prefetch" href="//b1.bd
Bytes received: 512
static.com/">
<title>百度一下，你就知道</title>
<link href="http://sl.bdstatic.com/r/www/cache/static/home/css/index.css" rel="sty
C:\Windows\system32\cmd.exe
C:\Users\hongxing\source\repos\webServer\x64\Debug>cd x64\Debug
驱动器 C 中的卷没有标签。
卷的序列号是: 2478-67AA
C:\Users\hongxing\source\repos\webServer\x64\Debug 的目录
2020/05/06 09:12 <DIR> .
2020/05/06 09:12 <DIR> ..
2020/05/06 09:58 63,488 webServer.exe
2020/05/06 09:58 462,892 webServer.ilk
2020/05/06 09:58 552,960 webServer.pdb
2 个文件 1,079,340 字节
2 目录 222,929,121,280 可用字节
C:\Users\hongxing\source\repos\webServer\x64\Debug>webServer.exe
Server listening ...
GET / HTTP/1.0
Bytes sent: 131
C:\Users\hongxing\source\repos\webServer\x64\Debug>webServer.exe
Server listening ...
GET / HTTP/1.0
Bytes sent: 131
C:\Users\hongxing\source\repos\webServer\x64\Debug>

```

Windows 10 on Mac

```
C:\Windows\system32\cmd.exe
C:\Windows\system32\cmd.exe
```

显示了从客户端程序捕获到的HTTP响应和目录列表。左侧窗口显示了对baidu.com的请求，右侧窗口显示了webServer的目录结构。

客户端程序打印出从 www.baidu.com 收到的 HTTP 响应以及 HTML 文件。

3. 运行客户端程序访问 www.sina.com:

Windows 10 on Mac

```
C:\Windows\system32\cmd.exe
C:\Windows\system32\cmd.exe
```

显示了从客户端程序捕获到的HTTP响应和目录列表。左侧窗口显示了对sina.com的请求，右侧窗口显示了webServer的目录结构。

发送“GET / HTTP/1.0\r\n\r\n”的HTTP请求，收到拒绝访问的403的HTML信息。

The screenshot shows two side-by-side Windows command-line windows. The left window is titled 'Windows 10 on Mac' and contains the following text:

```
C:\Users\hongxing\source/repos\webClient\x64\Debug>webClient.exe www.sina.com
Connection closed
C:\Users\hongxing\source/repos\webClient\x64\Debug>webClient.exe www.sina.com
Bytes Sent: 16
HTTP/1.1 400 Bad Request
Server: nginx
Date: Fri, 08 May 2020 03:02:33 GMT
Content-Type: text/html
Content-Length: 150
Connection: close
X-Via:CNN: f=edge, s=cmcc.jiangxi.union.175.nb.sinaedge.com, c=117.136.21.204;
<html>
<head><title>400 Bad Request</title></head>
<body><h1>400 Bad Request</h1></body>
<hr><center>nginx</center>
</html>
Connection closed
C:\Users\hongxing\source/repos\webClient\x64\Debug>webClient.exe www.sina.com
Bytes Sent: 16
HTTP/1.1 505 HTTP Version Not Supported
Server: nginx
Date: Fri, 08 May 2020 03:03:44 GMT
Content-Type: text/html
Content-Length: 180
Connection: close
X-Via:CNN: f=edge, s=cmcc.jiangxi.union.173.nb.sinaedge.com, c=117.136.21.204;
<html>
<head><title>505 HTTP Version Not Supported</title></head>
<body>
<center><h1>505 HTTP Version Not Supported</h1></center>
<hr><center>nginx</center>
</body>
</html>
Connection closed
C:\Users\hongxing\source/repos\webClient\x64\Debug>
```

The right window is also titled 'Windows 10 on Mac' and contains the following text:

```
C:\Users\hongxing\source/repos\webServer>cd x64\Debug
C:\Users\hongxing\source/repos\webServer>dir
驱动器 C: 中的卷没有标签。
卷的序列号是: 2478-67AA
C:\Users\hongxing\source/repos\webServer\x64\Debug 的目录
2020/05/06 09:12 <DIR> .
2020/05/06 09:12 <DIR> ..
2020/05/06 09:58 63,488 webServer.exe
2020/05/06 09:58 462,892 webServer.ilk
2020/05/06 09:58 552,960 webServer.pdb
2 个文件 1,154,340 字节
2 个目录 222,929,121,230 可用字节
C:\Users\hongxing\source/repos\webServer\x64\Debug>webServer.exe
Server listening ...
GET / HTTP/1.0
Bytes sent: 131
C:\Users\hongxing\source/repos\webServer\x64\Debug>webServer.exe
Server listening ...
GET / HTTP/1.0
Bytes sent: 131
C:\Users\hongxing\source/repos\webServer\x64\Debug>
```

发送“GET / HTTP/1.1\r\n”的 HTTP 请求，收到 HTTP 版本不支持的 HTML 信息。

发送“GET / HTTP/2.0\r\n”的 HTTP 请求，收到 HTTP 版本不支持的 HTML 信息。

4. 用客户端程序访问服务器程序，接收到服务器响应的 HTML 信息。

The screenshot shows two side-by-side Windows command-line windows. Both windows are titled 'Windows 10 on Mac' and contain the following text:

```
C:\Users\hongxing\source/repos\webClient\x64\Debug>webClient.exe 127.0.0.1
Bytes Sent: 131
Bytes Received: 131
<html><body>This is a response <b>message</b> in HTML
t color="red">Wow!</font><br></body></html>
Connection closed
C:\Users\hongxing\source/repos\webClient\x64\Debug>
```

5. 用浏览器访问服务器程序，接收、解析接收到的 HTML 信息并将其显示在浏览器上：

```

Windows 10 on Mac
127.0.0.1
127.0.0.1/
This is a response message in HTML format. Wow!
C:\Windows\system32\cmd.exe
C:\Users\hongxing\source\repos\webServer\x64\Debug>webServer.exe
Server listening ...
GET / HTTP/1.1
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.8,zh-Hans-CN;q=0.5,zh-Hans;q=0.3
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/70.0.3538.102 Safari/537.36 Edge/18.18363
Accept-Encoding: gzip, deflate
Host: 127.0.0.1
Connection: Keep-Alive
Bytes sent: 131

C:\Users\hongxing\source\repos\webServer\x64\Debug>webServer.exe
Server listening ...
GET / HTTP/1.1
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.8,zh-Hans-CN;q=0.5,zh-Hans;q=0.3
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/70.0.3538.102 Safari/537.36 Edge/18.18363
Accept-Encoding: gzip, deflate
Host: 127.0.0.1
Connection: keep-Alive
Bytes sent: 131
C:\Users\hongxing\source\repos\webServer\x64\Debug>

```

■ 多线程处理 web 服务器:

- 从浏览器访问服务器的 TXT 文件，浏览器解析接收到的文件并打印在浏览器上：

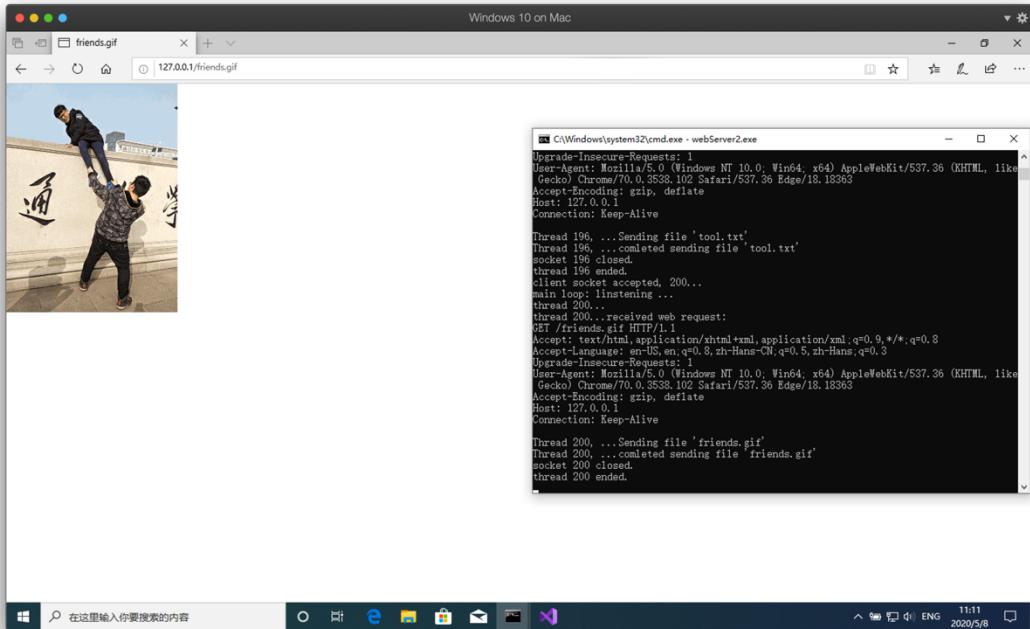
```

Windows 10 on Mac
127.0.0.1
127.0.0.1/tool.txt
A advanced tool based on bayesian attack graph theory and extending MulVAL, which can can quantitatively evaluate the security of a network efficiently.

C:\Windows\system32\cmd.exe - webServer2.exe
GET / HTTP/1.1
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.8,zh-Hans-CN;q=0.5,zh-Hans;q=0.3
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/70.0.3538.102 Safari/537.36 Edge/18.18363
Accept-Encoding: gzip, deflate
Host: 127.0.0.1
Connection: Keep-Alive
File '' not found --- sending an HTTP 404
client socket accepted, 196..
main loop is listening ...
thread 196...
thread 196...received web request:
GET /tool.txt HTTP/1.1
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.8,zh-Hans-CN;q=0.5,zh-Hans;q=0.3
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/70.0.3538.102 Safari/537.36 Edge/18.18363
Accept-Encoding: gzip, deflate
Host: 127.0.0.1
Connection: Keep-Alive
Thread 196, ...Sending file 'tool.txt'
Thread 196, ...completed sending file 'tool.txt'
socket 196 closed.
thread 196 ended.

```

- 从浏览器访问服务器的 GIF 文件，浏览器解析接收到的文件并展示在浏览器上：



3. 为了测试该服务器的性能，将原来显示“FILE NOT FOUND”的 HTML 信息改为更复杂的 HTML 信息：

```

<html>
<head>
    <meta charset="utf-8">
    <title>Live2D Display!</title>
    <link rel="stylesheet" href="live2d/css/live2d.css" />
</head>
<body>
<div id="landlord">
    <div class="message" style="opacity:0"></div>
    <canvas id="live2d" width="280" height="250"
class="live2d"></canvas>
    <div class="hide-button">Hide</div>
</div>
<script type="text/javascript"
src="https://cdn.bootcss.com/jquery/2.2.4/jquery.min.js"></script>
>
<script type="text/javascript">
    var message_Path = 'https://6dduu6.github.io/live2d/'
    var home_Path = 'https://6dduu6.github.io/'
</script>
<script type="text/javascript"
src="live2d/js/live2d.js"></script>
<script type="text/javascript"

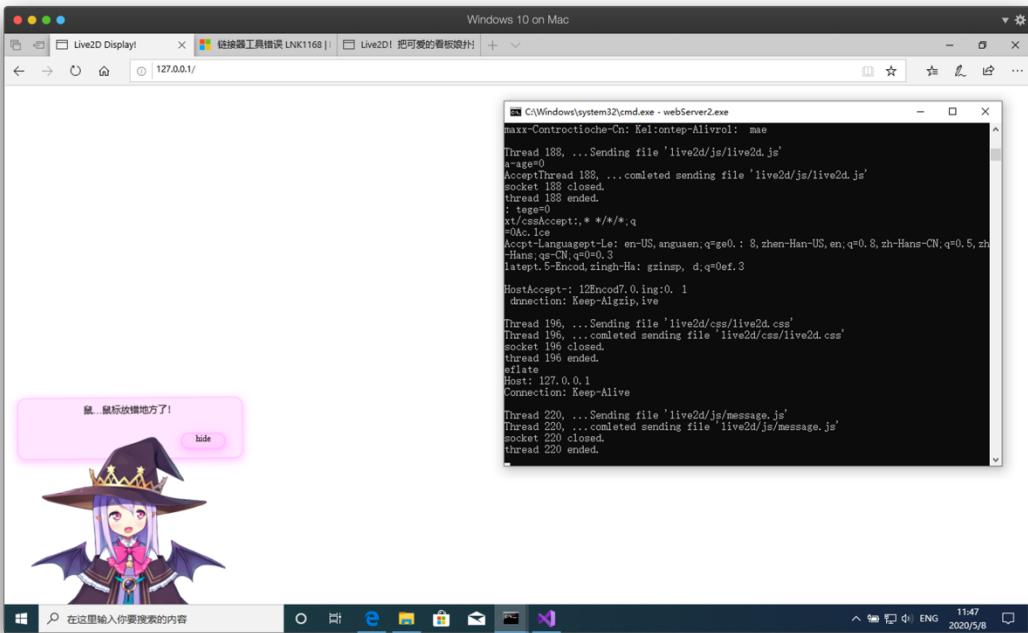
```

```

src="live2d/js/message.js"></script>
<script type="text/javascript">
    loadlive2d("live2d",
"https://6dduu6.github.io/live2d/Pio/model.json");
</script>
</body>
</html>

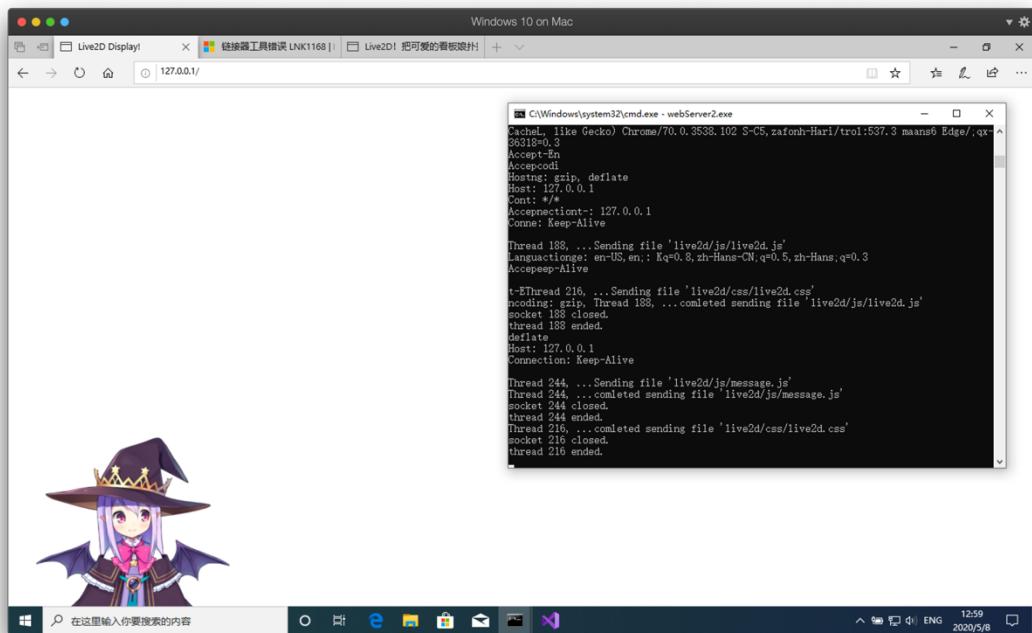
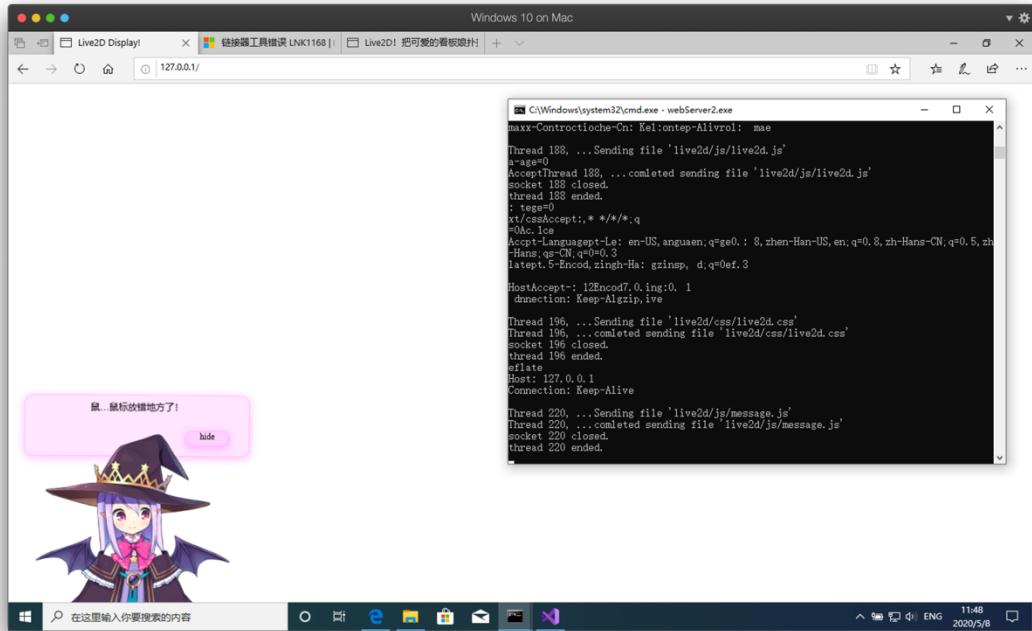
```

这是我之前建站的时候写的在网页前端插入可以和用户进行动态交互的 live2D 动漫小人的 HTML 代码。将 js 和 css 等控制文件放在相应位置。在浏览器访问 127.0.0.1，在网页上出现 live2D 小人，其视线跟随鼠标，并能响应用户点击做出动作与进行对话。



同时在服务器端可以看到，点击小人时，服务器开启不同线程来传输不同的 js 或 css 文件。

刷新之后正常进入网页。



事实上，只要服务器端能够正确响应浏览器的请求，支持多线程传输文件，那么任何形式的 HTML 网页都可以成功显示，只是不能实现网站的控制逻辑。

三、问题与思考

在搭建单次响应服务器时遇到了问题：利用自己写的客户端可以与服务器完成正确的交互，但用浏览器却无法与服务器完成交互。在上课时为了解决这个问题花了不少时间，因此没能在课堂上完成作业。于是下课后继续进行探究。虽然浏览器无法与服务器完成交互，但服务端会打印浏览器的请求并显示给浏览器发送了 HTML 数据，说明浏览器是可以访问到服务器的，但不知道什么原因浏览器一直在加载，无法将 HTML 显示出来。关闭浏

览器后，服务器程序会打印“Error recv”的错误，而关闭客户端程序，服务器程序会打印“Connect close”，这是两者访问服务器时服务器打印的信息中除访问者信息外唯一不同的地方。再深入发掘，发现浏览器是用 HTTP1.1 访问服务器程序的，而 HTTP1.1 是一种保持的连接，再结合我自己的程序，发现服务器的接收和发送函数是在一个循环内，只有客户端断开连接后服务器才会随之断开连接。这样答案就出来了。浏览器接收到的 HTML 语句并非文件，而是字符串，因此浏览器不知道服务器发送的响应信息是否结束，由于 HTTP1.1 的原因会一直等待服务器继续发送线程，而服务器也因为循环的原因会一直等待浏览器关闭连接。这样的相互等待造成了死锁，最后连接超时。而自己写的客户端程序是按字符串接收的，接收完字符串就自动关闭连接，因此能正常交互。

本次实验让利用 winsock 编程我们实现了功能较全的 web 服务器。这让最近在搭建网站的我感受颇深。之前并不了解 web 服务器的原理，只是按网上的教程配置 nginx、uWSGI 等热门服务器，现在自己用 C++ 写一个服务器，一百多行代码就完成了 web 服务器的基本功能，甚至能支持较复杂的网页，这实在是一个了不起的成就。之后我会继续完善 web 服务器的代码，希望有朝一日能写出一个 nginx 一样强大的服务器。