

# 实验二报告

2013605 张文迪

## 实验要求

1. 搭建Web服务器（自由选择系统），并制作简单的Web页面，包含简单文本信息（至少包含专业、学号、姓名）和自己的LOGO。
2. 通过浏览器获取自己编写的Web页面，使用Wireshark捕获浏览器与Web服务器的交互过程，并进行简单的分析说明。
3. 提交实验报告。

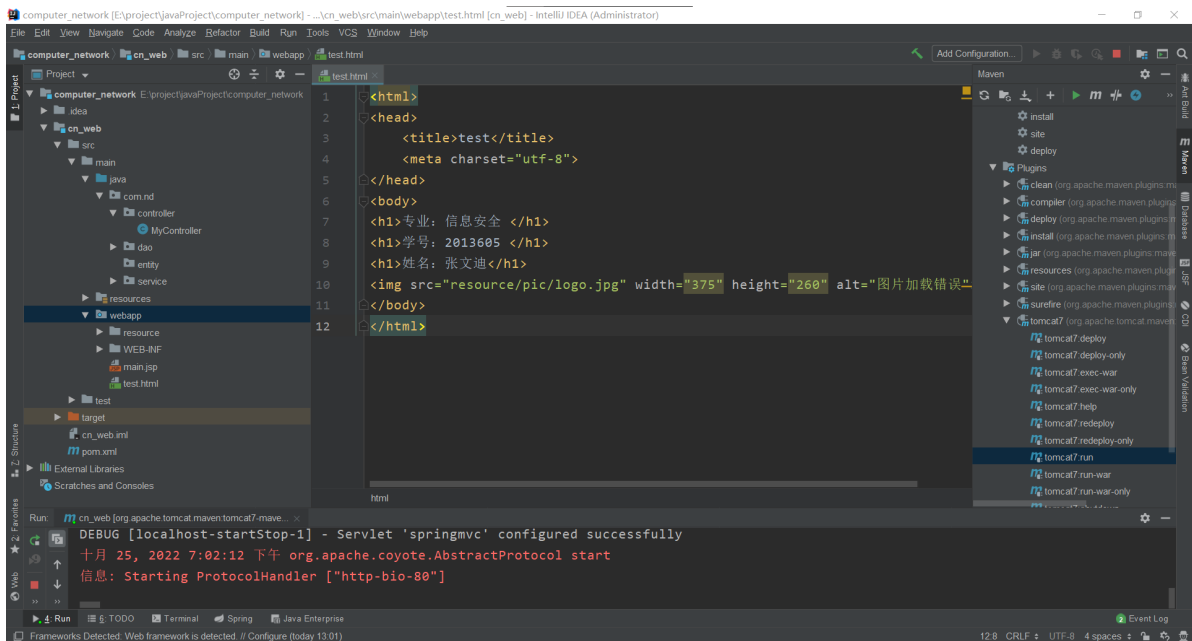
## Web服务器搭建与Web页面制作

### Web服务器搭建

- 使用maven构建JavaWeb项目，在 `pom.xml` 文件中添加对应的依赖
- 使用嵌入式的tomcat插件来运行Web服务，需要设置版本和端口号等

```
1  <plugins>
2      <!-- tomcat插件控制 -->
3      <plugin>
4          <groupId>org.apache.tomcat.maven</groupId>
5          <artifactId>tomcat7-maven-plugin</artifactId>
6          <version>2.2</version>
7          <configuration>
8              <!--指定服务器的端口号为80，即默认的80端口 -->
9              <port>80</port>
10             <path>/</path>
11             <ignorePackaging>true</ignorePackaging>
12         </configuration>
13     </plugin>
14 </plugins>
```

其总体框架如下图所示：



## HTML页面的设计

```
1 <html>
2 <head>
3     <title>test</title>
4     <!-- 指定编码格式, 防止中文乱码 -->
5     <meta charset="utf-8">
6 </head>
7 <body>
8 <h1>专业: 信息安全 </h1>
9 <h1>学号: 2013605 </h1>
10 <h1>姓名: 张文迪</h1>
11 
12 </body>
13 </html>
```

## 访问

由于指定Web服务为默认端口号80, 所以可直接通过 127.0.0.1/test.html 来进行访问, 不需要指定端口号。结果如下所示:

专业：信息安全

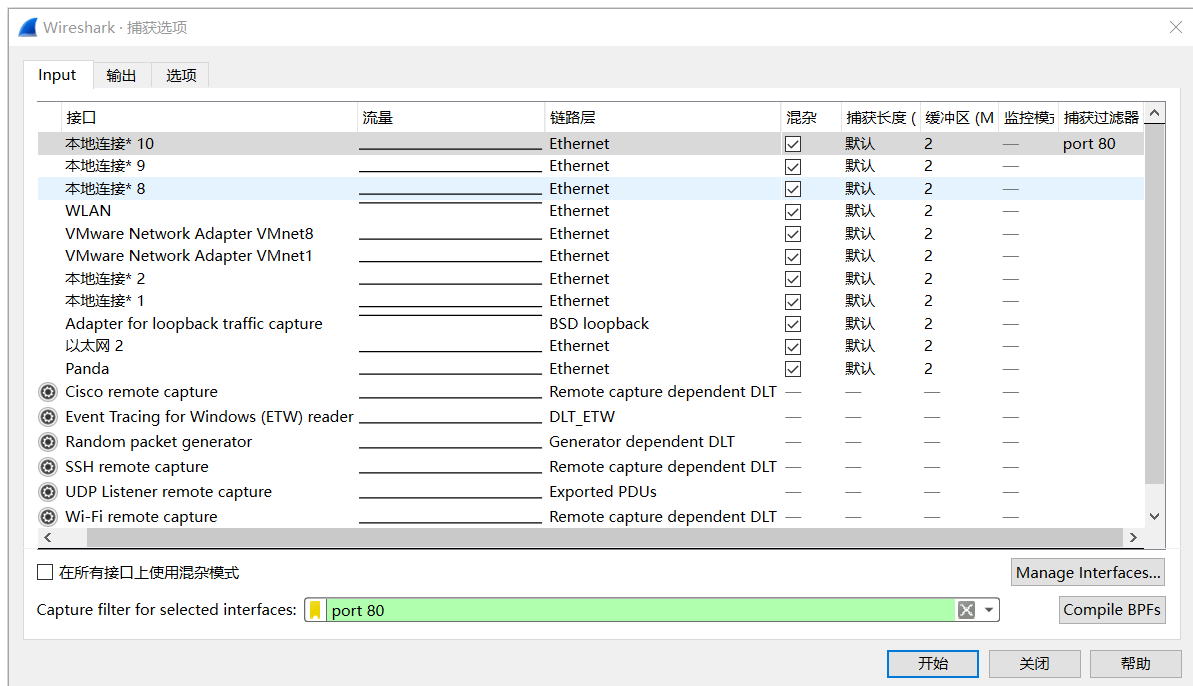
学号：2013605

姓名：张文迪

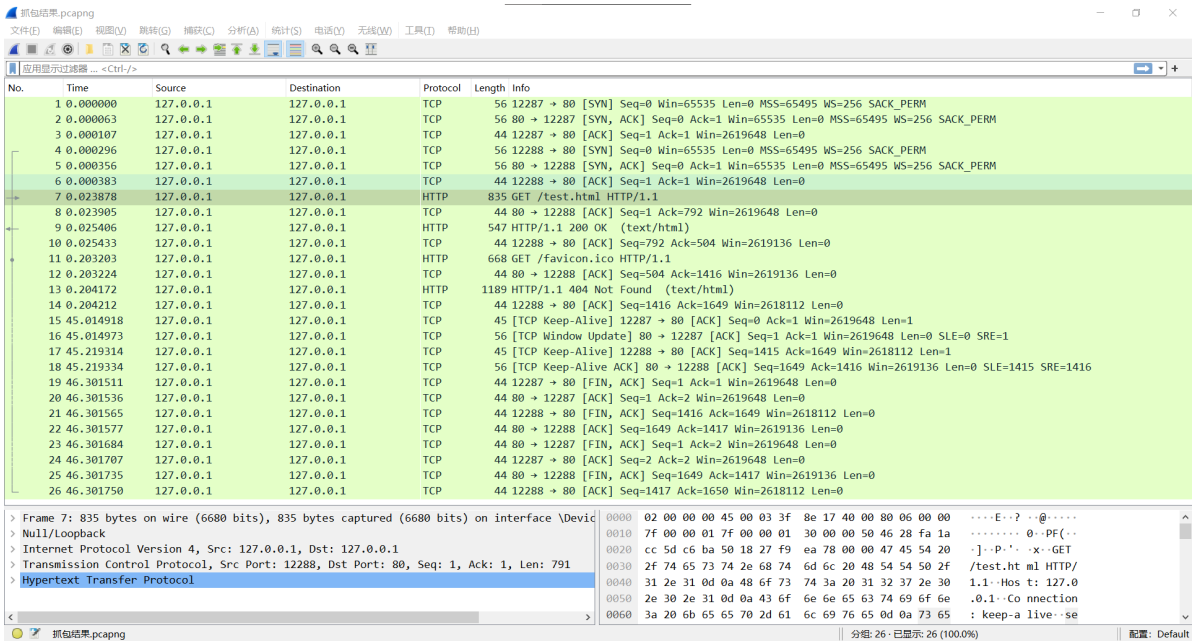


## 捕获交互过程并分析

1. 首先使用tomcat插件开启Java Web服务，使其运行在本机的 80 端口
2. 然后开启Wireshark，由于是在本机开启的Web服务并且使用本机浏览器进行访问，所以捕获选项的输入选择 Adapter for loopback traffic capture，即抓取本地环回的数据。
3. 由于服务器运行在 80 端口，那么我们只关心与 80 端口相关的数据包，那么可以设置过滤规则：port 80。然后开始开始捕获。



4. 在浏览器中输入 127.0.0.1/test.html 可以访问到事先设计好的html页面，然后查看Wireshark捕获到的数据报文和分组



## HTTP相关数据包

7	0.023878	127.0.0.1	127.0.0.1	HTTP	835 GET /test.html HTTP/1.1
8	0.023905	127.0.0.1	127.0.0.1	TCP	44 80 → 12288 [ACK] Seq=1 Ack=792 Win=2619648 Len=0
9	0.025406	127.0.0.1	127.0.0.1	HTTP	547 HTTP/1.1 200 OK (text/html)
10	0.025433	127.0.0.1	127.0.0.1	TCP	44 12288 → 80 [ACK] Seq=792 Ack=504 Win=2619136 Len=0
11	0.203203	127.0.0.1	127.0.0.1	HTTP	668 GET /favicon.ico HTTP/1.1
12	0.203224	127.0.0.1	127.0.0.1	TCP	44 80 → 12288 [ACK] Seq=504 Ack=1416 Win=2619136 Len=0
13	0.204172	127.0.0.1	127.0.0.1	HTTP	1189 HTTP/1.1 404 Not Found (text/html)

### 首先查看两个GET请求：

> Frame 7: 835 bytes on wire (6680 bits), 835 bytes captured (6680 bits) on interface \Device\NPF{...}	0000	02 00 00 00 45 00 03 f3	8e 17 40 00 00 06 00 00	....E...? ..@.....
> Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1	0010	7f 00 00 01 7f 00 00 01	30 00 00 50 46 28 fa 1a	..... 0..PF(.....
> Transmission Control Protocol, Src Port: 12288, Dst Port: 80, Seq: 1, Ack: 1, Len: 791	0020	cc 5d c6 ba 50 18 27 f9	ea 78 00 00 47 45 54 20	..P..'.x:GET
> Hypertext Transfer Protocol	0030	2f 74 65 73 74 2e 68 74	6d 6c 20 48 54 54 50 2f	/test.ht ml HTTP/
> GET /test.html HTTP/1.1\r\n	0040	31 2e 31 0d 0a 48 6f 73	74 3a 20 31 32 37 2e 30	1.1..Hos t: 127.0
Host: 127.0.0.1\r\n	0050	2e 30 2e 31 0d 0a 43 6f	6e 6e 65 63 74 69 6f 6e	.0.1..Co nnection
Connection: keep-alive\r\n	0060	3a 20 6b 65 65 70 2d 61	6c 69 76 65 0d 0a 73 65	. keep-a live..se
sec-ch-ua: "Chromium";v="106", "Microsoft Edge";v="106", "Not;A=Brand";v="99"\r\n	0070	63 2d 63 68 2d 75 61 3a	20 22 43 68 72 6f 6d 69	c-ch-ua: "Chromi
sec-ch-ua-mobile: ?0\r\n	0080	75 6d 22 3b 76 3d 22 31	30 36 22 2c 20 22 4d 69	um";v="1 06", "Mi
sec-ch-ua-platform: "Windows"\r\n	0090	63 72 6f 73 6f 66 74 20	45 64 6f 65 22 3b 76 3d	crosoft Edge";v=
Upgrade-Insecure-Requests: 1\r\n	00a0	22 31 30 36 22 2c 20 22	4e 6f 74 3b 41 3d 42 72	"106", " Not;A=Br
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/106.0.0.0 Safari/537.36\r\n	00b0	61 6e 64 22 3b 76 3d 22	39 39 22 0d 0a 73 65 63	and";v=" 99"..sec
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8\r\n	00c0	2d 63 68 2d 75 61 2d 6d	6f 62 69 6c 65 3a 20 3f	-ch-ua-m obile: ?
Accept-Encoding: gzip, deflate, br\r\n	00d0	30 0d 0a 73 65 63 2d 63	68 2d 75 61 2d 70 6c 6f	0..sec-c h-ua-pla
Accept-Language: zh-CN,zh;q=0.9,en;q=0.8,en-GB;q=0.7,en-US;q=0.6\r\n	00e0	74 66 6f 72 6d 3a 20 22	57 69 6e 64 6f 77 73 22	tform: " Windows
If-None-Match: W/"276-1666693513237"\r\n	00f0	0d 0a 55 70 67 72 61 64	65 2d 49 6e 73 65 63 75	..Upgrad e-Insecu
If-Modified-Since: Tue, 25 Oct 2022 10:25:13 GMT\r\n	0100	72 65 2d 52 65 71 75 65	73 74 73 3a 20 31 0d 0a	re-Reque sts: 1..
\r\n	0110	55 73 65 72 2d 41 67 65	6e 74 3a 20 4d 6f 7a 69	User-Age nt: Mozi
[Full request URI: http://127.0.0.1/test.html]	0120	6c 6c 61 2f 35 2e 30 20	28 57 69 6e 64 6f 77 73	lla/5.0 (Windows
[HTTP request 1/2]	0130	20 4e 54 20 31 30 2e 30	3b 20 57 69 6e 36 34 3b	NT 10.0 ; Win64;
[Response in frame: 9]	0140	20 78 36 34 29 20 41 70	70 6c 65 57 65 62 4b 69	x64) Ap pleWebKi
[Next request in frame: 11]	0150	74 2f 35 33 37 2e 33 36	20 28 4b 48 54 4d 4c 2c	t/537.36 (KHTML,
	0160	20 6c 69 6b 65 20 47 65	63 6b 6f 29 20 43 68 72	like Ge cko) Chr
	0170	6f 6d 65 2f 31 30 36 2e	30 2e 30 2e 30 20 53 61	ome/106. 0.0.0 Sa
	0180	66 61 72 69 2f 35 33 37	2e 33 36 20 45 64 6f 2f	fari/537 .36 Edg/
	0190	31 30 36 2e 30 2e 31 33	37 30 2e 35 32 0d 0a 41	106.0.13 70.52-A
	01a0	63 63 65 70 74 3a 20 74	65 78 74 2f 68 74 6d 6c	ccept: t ext/html
	01b0	2c 61 70 70 6c 69 63 61	74 69 6f 6e 2f 78 68 74	,applica tion/xht
	01c0	6d 6c 2b 78 6d 6c 2c 61	70 70 6c 69 63 61 74 69	ml+xml,a pplicati

> Frame 11: 668 bytes on wire (5344 bits), 668 bytes captured (5344 bits) on interface \Device\NPF{...}	0000	cc 5d c8 b1 50 18 27 f7	bf 14 00 00 47 45 54 20	..P..'. ....GET
> Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1	0010	2f 66 61 76 69 63 6f 6e	2e 69 63 6f 20 48 54 54	/favicon .ico HTT
> Transmission Control Protocol, Src Port: 12288, Dst Port: 80, Seq: 792, Ack: 504, Len: 624	0020	50 2f 31 2e 31 0d 0a 48	6f 73 74 3a 20 31 32 37	P/1.1..H ost: 127
> Hypertext Transfer Protocol	0030	2e 30 2e 30 2e 31 0d 0a	43 6f 6e 6e 65 63 74 69	.0.0.1.. Connecti
> GET /favicon.ico HTTP/1.1\r\n	0040	6f 6e 3a 20 6b 65 65 70	2d 61 6c 69 69 65 0d 0a	on: keep -alive..
Host: 127.0.0.1\r\n	0050	73 65 63 2d 63 68 2d 75	61 3a 20 22 43 68 72 6f	sec-ch-ua : "Chro
Connection: keep-alive\r\n	0060	6d 69 75 6d 22 3b 76 3d	22 31 30 36 22 2c 20 22	mium";v="106", "
sec-ch-ua: "Chromium";v="106", "Microsoft Edge";v="106", "Not;A=Brand";v="99"\r\n	0070	4d 69 63 72 6f 73 6f 66	74 20 45 64 67 65 22 3b	Microsof t Edge";
sec-ch-ua-mobile: ?0\r\n	0080	76 3d 22 31 30 36 22 2c	20 22 4e 6f 74 3b 41 3d	v="106", " Not;A=
sec-ch-ua-platform: "Windows"\r\n	0090	42 72 61 6e 64 22 3b 76	3d 22 39 39 22 0d 0a 73	Brand";v ="99"..s
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/106.0.0.0 Safari/537.36\r\n	00a0	65 63 2d 63 68 2d 75 61	2d 6d 6f 62 69 6c 65 3a	ec-ch-ua -mobile:
Accept: image/webp,image/apng,image/svg+xml,image/*,*/*;q=0.8\r\n	00b0	20 3f 30 0d 0a 55 73 65	72 2d 41 67 65 6e 74 3a	70..Use r-Agent:
Sec-Fetch-Site: same-origin\r\n	00c0	20 4d 6f 74 69 6c 6c 61	2f 35 2e 30 20 28 57 69	Mozilla /5.0 (Wi
Sec-Fetch-Mode: no-cors\r\n	00d0	6e 64 6f 77 73 20 4e 54	20 31 30 2e 30 3b 20 57	ndows NT 10.0; W
Sec-Fetch-Dest: image\r\n	00e0	69 6e 6e 34 3b 20 78 36	34 29 20 41 70 70 6c 65	in64; x6 4) Apple
Referer: http://127.0.0.1/test.html\r\n	00f0	57 65 62 4b 69 74 2f 35	63 67 2e 33 36 20 28 4b	WebKit/5 37.36 (K
Accept-Encoding: gzip, deflate, br\r\n	0100	48 54 4d 4c 2c 20 6c 69	6b 65 20 47 65 63 6b 6f	HTML, li ke Gecko
Accept-Language: zh-CN,zh;q=0.9,en;q=0.8,en-GB;q=0.7,en-US;q=0.6\r\n	0110	29 20 43 68 72 6f 6d 65	2f 31 30 36 2e 30 2e 30	) Chrome /106.0.0
\r\n	0120	2e 30 20 53 61 66 61 72	69 2f 35 33 37 2e 33 36	.0 Safari i/537.36
[Full request URI: http://127.0.0.1/favicon.ico]	0130	20 45 64 6f 2f 31 30 36	2e 30 2e 31 33 37 30 2e	Edg/106 .0.1370.
[HTTP request 2/2]	0140	35 32 0d 0a 73 65 63 2d	63 68 2d 75 61 2d 70 6c	52..sec- ch-ua-pl
[Prev request in frame: 7]	0150	61 74 66 6f 72 6d 3a 20	22 57 69 6e 64 6f 77 73	atform: "Windows
[Response in frame: 13]	0160	2d 0d 0a 41 63 63 65 70	74 3a 20 69 6d 61 67 65	..Accep t: image
	0170	2f 77 65 62 70 2c 69 6d	61 67 65 2f 61 70 6e 6f	/webp,i m age/apng
	0180	2c 69 6d 61 67 65 2f 73	76 67 2b 78 6d 6c 2c 69	,image/s vg+xml,i
	0190	6d 61 67 65 2f 2a 2c 2a	2f 2a 3b 71 3d 30 2e 38	mage/*,* /*;q=0.8

从HTTP请求(request)报文来看，它主要由两部分组成：请求行和请求头

### 请求行

包括请求方法、请求URL、HTTP版本和回车换行字符

上图中显示请求方法为GET；URL分别为/test.html，/favicon.ico；采取的协议版本为HTTP/1.1。其中/test.html是我们在浏览器中输入的想要请求的页面，而/favicon.ico即 Favorites Icon 的缩写，是指显示在浏览器收藏夹、地址栏和标签标题前面的个性化图标，方便以图标的方式区别不同的网站。

## 请求头

请求头是以冒号分隔的键名与键值对，以回车(CR)加换行(LF)符号序列结尾。它们定义了一个超文本传输协议事务中的操作参数。

Host是指访问的URL；Connection:keep alive指定为持久连接；sec-ch-ua是被提出用来代替User-Agent的，提供与浏览器关联的品牌和重要版本，在受支持的浏览器上默认发送，可以被用户手动禁止。标题可以包含任何位置和任何名称的“假”身份。此功能旨在防止服务器直接拒绝未知用户代理，迫使用户代理对其身份标识撒谎；sec-ch-ua-mobile表示是否为移动端用户；sec-ch-ua-platform表示操作系统名称；Accept 浏览器能够接受的内容消息；Sec-Fetch-Site 请求发起者的来源与目标资源来源之间的关系；Sec-Fetch-Mode 该请求头表明了一个请求的模式；sec-Fetch-Dest 表示请求的目的地，即如何使用获取的数据；referrer 用于指明当前流量的来源参考页面；Accept-Encoding 声明浏览器支持的编码类型；Accept-Language 浏览器所支持的语言类型。

If-None-Match 允许在对应的内容未被修改的情况下返回304未修改（304 Not Modified），在本次实验中，如果在html文件没有修改的前提下重复访问该页面，服务器则会响应304

If-Modified-Since 一个文件最后修改日期，允许在对应的内容未被修改的情况下返回304未修改

7 0.009866	127.0.0.1	127.0.0.1	HTTP	835 GET /test.html HTTP/1.1
8 0.009928	127.0.0.1	127.0.0.1	TCP	44 80 → 11071 [ACK] Seq=1 Ack=792 Win=2619648 Len=0
9 0.015564	127.0.0.1	127.0.0.1	HTTP	166 HTTP/1.1 304 Not Modified
10 0.015616	127.0.0.1	127.0.0.1	TCP	44 11071 → 80 [ACK] Seq=792 Ack=123 Win=2619648 Len=0

## 下面查看两个response回应

> Null/Loopback

> Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1

> Transmission Control Protocol, Src Port: 80, Dst Port: 12288, Seq: 1, Ack: 792, Len: 50

> Hypertext Transfer Protocol

> HTTP/1.1 200 OK\r\n

Server: Apache-Coyote/1.1\r\n

Accept-Ranges: bytes\r\n

ETag: W/"277-1666693584286"\r\n

Last-Modified: Tue, 25 Oct 2022 10:26:24 GMT\r\n

Content-Type: text/html\r\n

> Content-Length: 277\r\n

Date: Tue, 25 Oct 2022 10:26:43 GMT\r\n

\r\n

[HTTP response 1/2]

[Time since request: 0.001528000 seconds]

[Request in frame: 7]

[Next request in frame: 11]

[Next response in frame: 13]

[Request URI: http://127.0.0.1/favicon.ico]

File Data: 277 bytes

> Line-based text data: text/html (12 lines)

<html>\r\n

<head>\r\n

<title>test</title>\r\n

<meta charset="utf-8">\r\n

</head>\r\n

<body>\r\n

<h1>专业：信息安全 </h1>\r\n

<h1>学号：2013605 </h1>\r\n

<h1>姓名：张文迪 </h1>\r\n

\r\n

</body>\r\n

</html>

0000 02 00 00 00 45 00 02 1f 8e 19 40 00 80 06 00 00 .....E....@.....

0010 7f 00 00 01 7f 00 00 01 00 50 30 00 cc 5d c6 ba .....P0...]

0020 46 28 fd 31 50 18 27 f9 5b a7 00 00 48 54 54 50 F(.P.''.[...HTTP

0030 2f 31 2e 31 20 32 30 30 20 4f 4b 0d 0a 53 65 72 /1.1 200 OK<Ser

0040 76 65 72 3a 20 41 70 61 63 68 65 2d 43 6f 79 6f ver: Apa che-Coyo

0050 74 65 2f 31 2e 31 0d 0a 41 63 63 65 70 74 2d 52 te/1.1.. Accept-R

0060 61 6e 67 65 73 3a 20 62 79 74 65 73 0d 0a 45 54 anges: b ytes<ET

0070 61 67 3a 20 57 2f 22 32 37 37 2d 31 36 36 36 36 ag: W/"2 77-16666

0080 39 33 35 38 34 32 38 36 22 0d 0a 4c 61 73 74 2d 93584286 "...Last-

0090 4d 6f 64 69 66 69 65 64 3a 20 54 75 65 2c 20 32 Modified : Tue, 2

00a0 35 20 4f 63 74 20 32 30 32 32 20 31 30 3a 32 36 5 Oct 20 22 10:26

00b0 3a 32 34 20 47 4d 54 0d 0a 43 6f 6e 74 65 6e 74 >24 GMT> <Content

00c0 2d 54 79 70 65 3a 20 74 65 78 74 2f 68 74 6d 6c <Type: t ext/html

00d0 0d 0a 43 6f 6e 74 65 6e 74 2d 4c 65 6e 67 74 68 <<Conten t-Length

00e0 3a 20 32 37 37 0d 0a 44 61 74 65 3a 20 54 75 65 : 277>>D ate: Tue

00f0 2c 20 32 35 20 4f 63 74 20 32 30 32 32 20 31 30 , 25 Oct 2022 10

0100 3a 32 36 3a 34 33 20 47 4d 54 0d 0a 0d 0a 3c 68 >26:43 G MT>....<h

0110 74 6d 6c 3e 0d 0a 3c 68 65 61 64 3e 0d 0a 20 20 <html>...<h ead>...

0120 20 20 3c 74 69 74 6c 65 3e 74 65 73 74 3c 2f 74 <title >test</t

0130 69 74 6c 65 3e 0d 0a 20 20 20 3c 6d 65 74 61 <title> <meta

0140 20 63 68 61 72 73 65 74 3d 22 75 74 66 2d 38 22 charset ="utf-8"

0150 3e 0d 0a 3c 2f 68 65 61 64 3e 0d 0a 3c 62 6f 64 >...</hea d>...<bod

0160 79 3e 0d 0a 3c 68 31 3e e4 b8 93 e4 b8 9a ef bc y>...<h1> .....

0170 9a e4 bf a1 e6 81 af e5 ae 89 e5 85 a8 20 3c 2f </>...<h1>...</

0180 68 31 3e 0d 0a 3c 68 31 3e e5 ad a6 e5 8f bf ef h1>...<h1>...</

0190 bc 9a 32 30 31 33 36 30 35 20 3c 2f 68 31 3e 0d >201360 5 </h1>

01a0 0a 3c 68 31 3e e5 a7 93 e5 90 8d ef bc 9a e5 bc <h1>...</h1>...</

01b0 a0 e6 96 87 e8 bf aa 3c 2f 68 31 3e 0d 0a 3c 69 >...</h1>...<i

01c0 6d 67 20 73 72 63 3d 22 72 65 73 6f 75 72 63 61 mg src=" resource

01d0 2f 70 69 63 2f 6c 6f 67 6f 2e 6a 70 67 22 20 77 /pic/log o.jpg" w

01e0 69 64 74 68 3d 22 33 37 35 22 20 68 65 69 67 68 idth="37 5" heigh

01f0 74 3d 22 32 36 30 22 20 61 6c 74 3d 22 e5 9b be t="260" alt="...

> Frame 13: 1189 bytes on wire (9512 bits), 1189 bytes captured (9512 bits) on interface 0

> Null/Loopback

> Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1

> Transmission Control Protocol, Src Port: 80, Dst Port: 12288, Seq: 504, Ack: 1416, Len: 11

> Hypertext Transfer Protocol

> HTTP/1.1 404 Not Found\r\n

Server: Apache-Coyote/1.1\r\n

Content-Type: text/html; charset=utf-8\r\n

Content-Language: en\r\n

> Content-Length: 973\r\n

Date: Tue, 25 Oct 2022 10:26:43 GMT\r\n

\r\n

[HTTP response 2/2]

[Time since request: 0.000969000 seconds]

[Prev request in frame: 7]

[Prev response in frame: 9]

[Request in frame: 11]

[Request URI: http://127.0.0.1/favicon.ico]

File Data: 973 bytes

> Line-based text data: text/html (1 lines)

[truncated]<html><head><title>Apache Tomcat/7.0.47 - Error report</title><style><!--H1

0020 46 28 ff a1 50 18 27 f7 89 f5 00 00 48 54 54 50 F(.P.''.[...HTTP

0030 2f 31 2e 31 20 34 30 34 20 4e 6f 74 20 46 6f 75 /1.1 404 Not Fou

0040 6e 64 0d 0a 53 65 72 76 65 72 3a 20 41 70 61 63 nd> Serv er: Apac

0050 68 65 2d 43 6f 79 6f 74 65 2f 31 2e 31 0d 0a 43 he-Coyot e/1.1..<C

0060 6f 6e 74 65 6e 74 2d 54 79 70 65 3a 20 74 65 78 ontent-T ype: tex

0070 74 2f 68 74 6d 6c 3b 63 68 61 72 73 65 74 3d 75 t/html;c harset=u

0080 74 66 2d 38 0d 0a 43 6f 6e 74 65 6e 74 2d 4c 61 t/html> charset=La

0090 6e 67 75 61 67 65 3a 20 65 6e 0d 0a 43 6f 6e 74 nguage: en-<Cont

00a0 65 6e 74 2d 4c 65 6e 67 74 68 3a 20 39 37 33 0d ent-Leng th: 973>

00b0 0a 44 61 74 65 3a 20 54 75 65 2c 20 32 35 20 4f >Date: T ue, 25 O

00c0 63 74 20 32 30 32 32 20 31 30 3a 32 36 3a 34 33 ct 2022 10:26:43

00d0 20 47 4d 54 0d 0a 0d 0a 3c 68 74 6d 6c 3e 3c 68 GMT>...<html><h

00e0 65 61 64 3e 3c 74 69 74 6c 65 3e 41 70 61 63 68 ead><tit le>Apach

00f0 65 20 54 6f 6d 63 61 74 2f 37 2e 30 2e 34 37 20 e Tomcat /7.0.47

0100 2d 20 45 72 72 6f 72 20 72 65 70 6f 72 74 3c 2f - Error report</

0110 74 69 74 6c 65 3e 3c 73 74 79 6c 65 3e 3c 21 2d >title><s tyle><!--

0120 2d 48 31 20 7b 66 6f 6e 74 2d 66 61 6d 69 6c 79 <H1 (fon t-Famil

0130 3a 54 61 6d 68 6f 6d 61 2c 41 72 69 61 6c 2c 73 61 >:Tahoma, Arial,sa

0140 6e 73 2d 73 65 72 69 66 3b 63 6f 6c 6f 72 3a 77 ns-serif ;color:w

0150 68 69 74 65 3b 62 61 63 6b 67 72 6f 75 6e 64 2d hite;back ground-

从HTTP相应行来看，它主要由三部分组成：响应行、响应头和响应体

### 响应行

其中包括HTTP版本号、状态码和解释

本次实验请求/test.html得到的状态码为200，代表请求成功；请求/favicon.ico得到的状态码为404，代表请求的资源不存在，因为我并未在服务器根目录下添加网站图标文件。

### 响应头

`server` 代表服务器的名字；`Content-Type` 当前内容的 `MIME` 类型（一个拓展的电子邮件标准）；`Content-Language` 内容所使用的语言；`Content-Length` 回应消息的长度，以字节为单位；`Date` 此条消息被发送时的日期和时间(按照 RFC 7231 中定义的“超文本传输协议日期”格式来表示)；

### 响应体

响应的数据部分

## TCP相关数据包

### TCP相关协议结构

TCP表头																																	
偏移	字节	0								1								2								3							
字节	比特	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	来源连接端口																目的连接端口															
4	32	序列号码																															
8	64	确认号码 (当ACK设置)																															
12	96	资料偏移				保留 0 0 0			N	C	E	U	A	P	R	S	F	窗口大小															
	S								W	C	R	C	S	S	Y	I																	
	R								E	G	K	H	T	N	N																		
16	128	校验和																紧急指针 (当URG设置)															
20	160	选项 (如果资料偏移 > 5, 需要在结尾添加0。)																															
...	...	...																															

序列号 (seq, 32位长)

- 如果含有同步化旗标 (SYN)，则此为最初的序列号；第一个资料比特的序列码为本序列号加一。
- 如果没有同步化旗标 (SYN)，则此为第一个资料比特的序列码。

确认号 (ack, 32位长) — 期望收到的数据的开始序列号。也即已经收到的数据的字节长度加1。

### TCP三次握手

TCP用三次握手过程创建一个连接。在连接创建过程中，很多参数要被初始化。

服务器端执行了listen函数后，就在服务器上创建起两个队列：

- SYN队列：存放完成了二次握手的结果。队列长度由listen函数的参数backlog指定。
- ACCEPT队列：存放完成了三次握手的结果。队列长度由listen函数的参数backlog指定。

三次握手协议的过程：

- 客户端通过执行connect函数向服务器端发送一个SYN包，请求一个主动打开。该包携带客户端为这个连接请求而设定的随机数A作为消息序列号。进入SYN\_SEND状态
- 服务器端收到一个合法的SYN包后，把该包放入SYN队列中；回送一个SYN/ACK。ACK的确认码应为A+1，SYN/ACK包本身携带一个随机产生的序号B。进入SYN\_RECV状态。



3. 客户端收到SYN/ACK包后，发送一个ACK包，该包的序号被设定为A+1，而ACK的确认码则为B+1。然后客户端的connect函数成功返回。当服务器端收到这个ACK包的时候，把请求帧从SYN队列中移出，放至ACCEPT队列中；这时accept函数如果处于阻塞状态，可以被唤醒，从ACCEPT队列中取出ACK包，重新创建一个新的用于双向通信的sockfd，并返回。进入Established状态。

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	127.0.0.1	127.0.0.1	TCP	56	12287 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM
2	0.000063	127.0.0.1	127.0.0.1	TCP	56	80 → 12287 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM
3	0.000107	127.0.0.1	127.0.0.1	TCP	44	12287 → 80 [ACK] Seq=1 Ack=1 Win=2619648 Len=0
4	0.000296	127.0.0.1	127.0.0.1	TCP	56	12288 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM
5	0.000356	127.0.0.1	127.0.0.1	TCP	56	80 → 12288 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM
6	0.000383	127.0.0.1	127.0.0.1	TCP	44	12288 → 80 [ACK] Seq=1 Ack=1 Win=2619648 Len=0

如上图所示，浏览器通常会创建多个线程与服务器建立连接，所以我们此次分析只针对与其中一个TCP连接，即port 1288

第一次握手：首先由本机的12288端口向80发送一个SYN包，包含的序列号（seq）为零、接受窗口大小（Win）为65535，后面为可选位（最大报文段长度、窗口扩大因子）

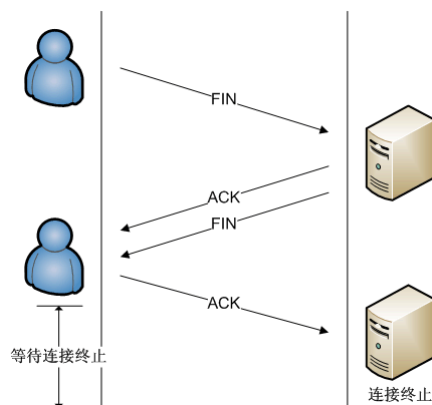
第二次握手：服务器向客户端发送一个SYN/ACK包，其确认码（ACK）为1，序列号（seq）为零，服务器端如果在一定时间内没有收到客户端的ACK包会重发SYN-ACK

第三次握手：本机向服务器发送一个ACK包，序列号（seq）为1，确认码（ACK）为1

除了首部外所有数据的长度均为零。

## TCP的四次挥手

客户端和服务端都可以主动发起挥手。四次挥手的过程：



在这个过程中连接的每一侧都独立地被终止。当一个端点要停止它这一侧的连接，就向对侧发送FIN，对侧回复ACK表示确认。因此，拆掉一侧的连接过程需要一对FIN和ACK，分别由两侧端点发出。

首先发出FIN的一侧，如果给对侧的FIN响应了ACK，那么就会超时等待2\*MSL时间，然后关闭连接。在这段超时等待时间内，本地的端口不能被新连接使用；避免延时的包的到达与随后的新连接相混淆。

19	46.301511	127.0.0.1	127.0.0.1	TCP	44	12287 → 80 [FIN, ACK] Seq=1 Ack=1 Win=2619648 Len=0
20	46.301536	127.0.0.1	127.0.0.1	TCP	44	80 → 12287 [ACK] Seq=1 Ack=2 Win=2619648 Len=0
21	46.301565	127.0.0.1	127.0.0.1	TCP	44	12288 → 80 [FIN, ACK] Seq=1416 Ack=1649 Win=2618112 Len=0
22	46.301577	127.0.0.1	127.0.0.1	TCP	44	80 → 12288 [ACK] Seq=1649 Ack=1417 Win=2619136 Len=0
23	46.301684	127.0.0.1	127.0.0.1	TCP	44	80 → 12287 [FIN, ACK] Seq=1 Ack=2 Win=2619648 Len=0
24	46.301707	127.0.0.1	127.0.0.1	TCP	44	12287 → 80 [ACK] Seq=2 Ack=2 Win=2619648 Len=0
25	46.301735	127.0.0.1	127.0.0.1	TCP	44	80 → 12288 [FIN, ACK] Seq=1649 Ack=1417 Win=2619136 Len=0
26	46.301750	127.0.0.1	127.0.0.1	TCP	44	12288 → 80 [ACK] Seq=1417 Ack=1650 Win=2618112 Len=0

由于客户端开启了多线程，我们同样只对12288端口分析

第一次挥手：客户端向服务器发送FIN，序列号为1416，确认码为1649。然后进入FIN-WAIT-1状态表示本方的数据发送全部结束，等待TCP连接另一端的ACK确认包或FIN&ACK请求包。

第二次挥手：服务发送一个ACK给C客户端，序列码为1649，确认码为1417，说明FIN和SYN类似占用一个序号，然后客户端进入**FIN-WAIT-2**状态其这时可以接收数据，但不再发送数据。服务器进入**CLOSE-WAIT**状态，这时可以发送数据，但不再接收数据。

第三次挥手，服务器发送一个FIN给客户端，序列码为1649，确认码为1417，进入**LAST-ACK**状态等待确认包

第四次挥手，客户端收到FIN后，发送一个ACK包，序列码为1517，确认码为1650，同时进入**TIME-WAIT**状态，等待足够时间以确保被动关闭端收到了终止请求的确认包按照RFC 793，一个连接可以在TIME-WAIT保证最大四分钟，即[最大分段寿命](#)（maximum segment lifetime）的2倍）。然后server进入**CLOSED** 状态，完全没有连接。