哈爾濱工業大學
Harbin Institute of Technology

计算机网络
课程实验报告

实验名称	HTTP 代理服务器的设计与实现				
姓名	姜思琪		院系	计算机科学与技术学院	
班级	1603109		学号	1160300814	
任课教师	李全龙		指导教师	李全龙	
实验地点	格物 213		实验时间	2018.10.27	
实验课表现	出勤、表现得分(10)		实验报告得分(40)	实验总分	
	操作结果得分(50)				
教师评语					

计算机科学与技术学院 SINCE 1956...
School of Computer Science and Technology

实验目的：

本次实验的主要目的。熟悉并掌握 Socket 网络编程的过程与技术；深入理解 HTTP 协议，掌握 HTTP 代理服务器的基本工作原理；掌握 HTTP 代理服务器设计与编程实现的基本技能。

实验内容：

(1) 设计并实现一个基本 HTTP 代理服务器。在指定端口10240接收来自客户的 HTTP 请求并且根据其中的 URL 地址访问该地址所指向的 HTTP 服务器（原服务器），接收 HTTP 服务器的响应报文，并将响应报文转发给对应的客户进行浏览。

(2) 设计并实现一个支持 Cache 功能的 HTTP 代理服务器。能缓存原服务器响应的对象，并能够通过修改请求报文，向原服务器确认缓存对象是否是最新版本。

(3) 扩展 HTTP 代理服务器，支持如下功能：

a) 网站过滤：允许/不允许访问某些网站；

b) 网站引导：将用户对某个网站的访问引导至一个模拟网站（钓鱼）。

实验过程：

(1) 设计并实现一个基本 HTTP 代理服务器。

基于Soctet编程的基本 HTTP 代理服务器需要扮演两个角色，分别是客户端和服务端。

功能描述：作为服务端，在指定端口10240接收来自客户的 HTTP 请求并且根据其中的 URL 地址访问该地址所指向的 HTTP 服务器（原服务器），接收 HTTP 服务器的响应报文，并将响应报文转发给对应的客户进行浏览；代理服务器通过解析客户端的请求信息，再向真实服务器发送请求报文，获得请求的信息，此时代理服务器是作为一个客户端。作为客户端，能够创建新的Socket，绑定服务器host和端口号。

采用多线程的方式实现。首先，我在main函数中调用startProxy(10240,HttpProxy.class)，端口是10240，此函数会创建套接字并监听，创建HttpProxy。在生成子线程时，构造器中会接收来自主机的请求以及发出的响应。将其转化为字符流便于比较。如下图：

```
public HttpProxy(Socket cs) {// 构造器
    try {
        csocket = cs;
        cis = csocket.getInputStream(); // 代理服务器作为服务器接受客户端的请求
        cbr = new BufferedReader(new InputStreamReader(cis));
        cos = csocket.getOutputStream(); // 代理服务器作为服务器向客户端发出响应
        cpw = new PrintWriter(cos);
        start();
```

在run()中进行解析及实现附加功能。一开始设置一个计时器。然后获取响应报文的URL。我是通过getRequestURL()函数来获得URL。具体内容如下：

```
public String getRequestURL(String buffer) {
    String[] tokens = buffer.split(" ");
    String URL = "";
    if (tokens[0].equals("GET"))
        for (int index = 0; index < tokens.length; index++) {
            if (tokens[index].startsWith("http://")) {
                URL = tokens[index];
                break;
            }
        }
    return URL;
}
```

然后获取host和分析可能存在的端口号。最多5次尝试与目标主机连接，如果建立连接成功，则重设计时器，代理服务器作为客户端接受响应和发出请求，在缓存中寻找是否之前已

经缓存过这个url的信息（这个功能将在第二点中进行详解）。若没有缓存且是我要钓鱼的网站，就会发送我准备好的请求报文进行替换；若果是其他网站，则正常发送原报文。然后读取服务器反应，读取客户端的请求转给服务器，进行正常的访问。下面函数是建立通道的函数。

```
public void pipe(InputStream cis, InputStream sis, OutputStream sos, OutputStream cos) {// 建立通道
    try {
        int length;
        byte bytes[] = new byte[BUFSIZE];
        while (true) {
            try {
                if ((length = cis.read(bytes)) > 0) { // 读取客户端的请求转给服务器
                    sos.write(bytes, 0, length);
                    if (logging)
                        writeLog(bytes, 0, length, 1);
                } else if (length < 0)
                    break;
            } catch (SocketTimeoutException e) {
            } catch (InterruptedIOException e) {
                System.out.println("\nRequest Exception:");
                e.printStackTrace();
            }
            try {
                if ((length = sis.read(bytes)) > 0) {// 接受服务器的响应回传给请求的客户端
                    cos.write(bytes, 0, length); // 因为是按字节读取，所以将回车和换行符也传递过去了
                    if (logging) {
                        writeLog(bytes, 0, length, 1);
                        writeLog(bytes, 0, length, 3);
                    }
                }
            } catch (SocketTimeoutException e) {
            } catch (InterruptedIOException e) {
                System.out.println("\nResponse Exception:");
                e.printStackTrace();
            }
        }
```

(2)支持Cache功能。

如果建立连接，会先在缓存中寻找是否之前已经缓存过这个URL的信息，通过函数findCache（）实现直接在存有url和相应信息的文件中查找，如果找到相同的，那么在响应信息中找上次修改的时间。

若有缓存，向服务器发送确认修改时间请求，添加"If-modified-since: "，接受服务器发回的信息，如果没更改"Not Modified"，就使用缓存中的数据；否则使用新的数据。控制台输出的信息“使用缓存中的数据”、“上一次修改时间：”、“有更新，使用新的数据”、“找到相同的了”等，例如下图：

```
上一次修改的时间为：ri, 26 Oct 2018 18:45:38 GMT
向服务器发送确认修改时间请求:GET http://8.tlu.dl.c
Host: 8.tlu.dl.delivery.mp.microsoft.com
If-modified-since: ri, 26 Oct 2018 18:45:38 GMT
服务器发回的信息是：HTTP/1.1 304 Not Modified
使用缓存中的数据
HTTP/1.1 206 Partial Content
Date: Mon, 29 Oct 2018 10:36:04 GMT
Cache-Control: public, max-age=17280000
Content-Type: application/octet-stream
Last-Modified: Fri, 26 Oct 2018 18:45:38 GMT
Accept-Ranges: bytes
ETag: "PuxnKPsqB6RPKYFpPiG6UGw12/0="
```

文件log_D直接存储相关URl对应的响应报文。文件log_C是记录日志。通过函数writeLog()函数来写日志。

```
public void writeLog(int c, int browser) throws IOException {// 写日志char
    if (browser == 1)
        log_C.write((char) c);
    else
        log_D.write((char) c);
}

public void writeLog(byte[] bytes, int offset, int len, int browser) throws IOException {
    for (int i = 0; i < len; i++)
        writeLog((int) bytes[offset + i], browser);
}
```

log_c.txt

log_d.txt

(3)网站过滤

我设置的被过滤的网站是腾讯网站http://www.qq.com/，什么值也不返回，就是设置为空，让用户进不去此网页。

```
} else if (URL.equals("http://www.qq.com/")) {// 网站过滤
    URL = "";
}
```

(4)网站引导（钓鱼）

如果想进入哈工大的电工电子实验中心，就会被引导到哈工大物理实验中心。我的实现方式是重新组成一个报文头部将其传给目标主机，从而达到将其引导到其他网站的目的。重要代码如下：

```
if (URL.equals("http://eelab.hit.edu.cn/")) {// 钓鱼（网站引导）
    URL = "http://clop.hit.edu.cn/index";
    buffer = "GET " + URL + " HTTP/1.1";
    requestInfo.add("Accept: text/html, application/xhtml+xml, */*");
    requestInfo.add("Accept-Language: zh-Hans-CN,zh-Hans;q=0.8,en-US;q=0.5,en;q=0.3");
    requestInfo.add("User-Agent: Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.2; WOW64; Trident/6.0)");
    requestInfo.add("Accept-Encoding: gzip, deflate");
    requestInfo.add("Proxy-Connection: Keep-Alive");
    requestInfo.add("DNT: 1");
    requestInfo.add("Host: clop.hit.edu.cn");
    requestInfo.add(
            "Cookie: thw=cn; isg=0BC4B5EFD7C7FCFEB73317770EA7F3F5; l=AeVoHE44ZTsle7DjpW8fBSV7pbSl-2U7; cna=GCHeDZQAVwkCAdvZ9Apwg8rH; "
            + "t=1a1386bec550ab78d1aaf5ad5b90e044; mt=ci%3D-1_0; _med=dw:1366&dh:768&pw:1366&ph:768&ist:0");
```

实验结果：

(1) 实现基本HTTP代理服务器。比如打开哈工大教务处网站http://jwc.hit.edu.cn/

控制台输出：

```
启动线程：5
到了读取第一行
buffer:GET http://jwc.hit.edu.cn/ HTTP/1.1
http://jwc.hit.edu.cn/
端口号：80主机：jwc.hit.edu.cn
第一行是 4:GET http://jwc.hit.edu.cn/ HTTP/1.1
第一次得到：
要找的是：http://jwc.hit.edu.cn/
上一次修改的时间为：null
向服务器发送请求：GET http://jwc.hit.edu.cn/ HTTP/1.1
向服务器发送请求：Accept: text/html, application/xhtml+xml, image/jxr, */*
向服务器发送请求：Accept-Language: zh-CN
向服务器发送请求：Accept-Encoding: gzip, deflate
向服务器发送请求：Host: jwc.hit.edu.cn
向服务器发送请求：Proxy-Connection: Keep-Alive
向服务器发送请求：User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; Trident/7.0; rv:11.0) like Gecko Core/1.63.6726.400 QQBrowser/10.2.2265.400
向服务器发送请求：Cookie: JSESSIONID=88DD9FB86CA2E916D9732FCF7D3536F4
```

(2) 支持 Cache 功能。能缓存原服务器响应的对象，并能够通过修改请求报文，向原服务器确认缓存对象是否是最新版本。

图1（下图）是第一次访问，无缓存情况的。即上一次修改时间为：null

```
启动线程：5
到了读取第一行
buffer:GET http://jwc.hit.edu.cn/ HTTP/1.1
http://jwc.hit.edu.cn/
端口号：80主机：jwc.hit.edu.cn
第一行是 4:GET http://jwc.hit.edu.cn/ HTTP/1.1
第一次得到：
要找的是：http://jwc.hit.edu.cn/
上一次修改的时间为：null
向服务器发送请求：GET http://jwc.hit.edu.cn/ HTTP/1.1
向服务器发送请求：Accept: text/html, application/xhtml+xml, image/jxr, */*
向服务器发送请求：Accept-Language: zh-CN
向服务器发送请求：Accept-Encoding: gzip, deflate
向服务器发送请求：Host: jwc.hit.edu.cn
向服务器发送请求：Proxy-Connection: Keep-Alive
向服务器发送请求：User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; Trident/7.0; rv:11.0) like Gecko Core/1.63.6726.400 QQBrowser/10.2.2265.400
向服务器发送请求：Cookie: JSESSIONID=88DD9FB86CA2E916D9732FCF7D3536F4
```

图1

图2（下图）是有缓存且确认修改时间请求发生改变，使用新数据。即有更新，使用新的数据。

```
找到相同的了：GET http://8.tlu.dl.delivery.mp.microsoft.com/filestreamingservice/files/3e9241f5-fd65-4622-9fdf-c9f33caa7ea6?P1=1
info1是：
上一次修改的时间为：ed, 19 Sep 2012 08:18:05 GMT
向服务器发送确认修改时间请求:GET http://8.tlu.dl.delivery.mp.microsoft.com/filestreamingservice/files/3e9241f5-fd65-4622-9fdf-c9f
Host: 8.tlu.dl.delivery.mp.microsoft.com
If-modified-since: ed, 19 Sep 2012 08:18:05 GMT
服务器发回的信息是：HTTP/1.1 200 OK
有更新，使用新的数据
新的数据是：HTTP/1.1 200 OK
新的数据是：Date: Sun, 28 Oct 2018 11:45:30 GMT
新的数据是：Cache-Control: public, max-age=17280000
新的数据是：Content-Type: application/octet-stream
新的数据是：Last-Modified: Wed, 10 Oct 2018 15:02:17 GMT
新的数据是：Accept-Ranges: bytes
新的数据是：ETag: "u5Vj/fp12ohMTN0SW51THu+3V4o="
新的数据是：Server: Microsoft-IIS/10.0
新的数据是：X-AspNetMvc-Version: 5.2
新的数据是：MS-CorrelationId: ae308300-0e6e-4a28-b85a-33354c2211b0
新的数据是：MS-RequestId: 23d87bb8-eb77-4749-8eab-83dc19d7a0bd
新的数据是：MS-CV: /p9eog4hJEa33Lwv.0.3.10.2.1.0.0.10.2.7.2.1.1.0
新的数据是：Content-Disposition: attachment; filename=Microsoft.ZuneVideo_2019.18082.13811.0_neutral_~_8wekyb3d8bbwe.AppxBundle
新的数据是：X-AspNet-Version: 4.0.30319
新的数据是：X-Powered-By: ASP.NET
```

图2

图3（下图）是有缓存无更新。即使用缓存中的数据。

```
上一次修改的时间为：on, 01 Oct 2018 07:46:11 GMT
向服务器发送确认修改时间请求:GET http://8.tlu.dl.delivery.mp.microsoft.com/filestreamingservice/files/9cd5d22
Host: 8.tlu.dl.delivery.mp.microsoft.com
If-modified-since: on, 01 Oct 2018 07:46:11 GMT
服务器发回的信息是：HTTP/1.1 304 Not Modified
使用缓存中的数据
GET http://www.qq.com/favicon.ico HTTP/1.1
GET http://web.mit.edu/favicon.ico HTTP/1.1
HTTP/1.1 200 OK
Date: Sun, 28 Oct 2018 11:35:12 GMT
Content-Type: image/x-icon
Transfer-Encoding: chunked
Connection: keep-alive
Server: squid/3.5.24
Last-Modified: Wed, 19 Sep 2012 08:18:05 GMT
Vary: Accept-Encoding
Expires: Tue, 27 Nov 2018 11:35:12 GMT
Cache-Control: max-age=2592000
Vary: Accept-Encoding
Content-Encoding: gzip
Vary: Accept-Encoding
X-Cache: HIT from tianjin.qq.com
```

(3)网站过滤。不能访问腾讯网址http://www.qq.com/，加载不出来。

(4)钓鱼。如果想进入哈工大电工电子服务中心http://eelab.hit.edu.cn/，就会被引导到哈工大物理实验中心http://clop.hit.edu.cn/index。如下图所示，打开即哈工大物理实验中心：

问题讨论：

实验内容通过课堂学习及课后查找书籍大多可解决。在本次实验中，我遇到了如下一些小的问题。

1.有些网站打不开甚至代码会抛出异常。

答：因为本次实验实现的仅仅是一个简单的HTTP代理服务器，还很简陋，不是所有网页都能打开，比如https的网站就打不开。而且在我的电脑后台还运行着其他程序需要网络，比如QQ等，所以会导致代码抛出连接超时异常。

2.同一个网站有时能打开，有时转的很慢。

答：网速很慢，打开网页快慢受网速影响，同样的实现的还是简陋。

心得体会：

通过本次实验我更加深刻的理解了HTTP以及它是如何工作的。

遇到困难也能更多的通过自己来查找资料，而不是一味地问别人，依靠别人。

在这个实验中，我知道了要多思考，要多问自己为什么，怎么做。多多思考，思考全面很重要。

同时感觉还有很多东西要学，现在自己能做的只能实现一个基本的代理服务器，要学习的东西还有很多。