



哈尔滨工业大学
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	陈一帆		院系	计算学部软件工程		
班级	1937102		学号	1191000606		
任课教师	李全龙		指导教师	李全龙		
实验地点	格物 207		实验时间	2021.10.30		
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						

实验目的：

掌握并熟悉Socket网络编程的过程与技术；深入理解HTTP协议，掌握HTTP的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。

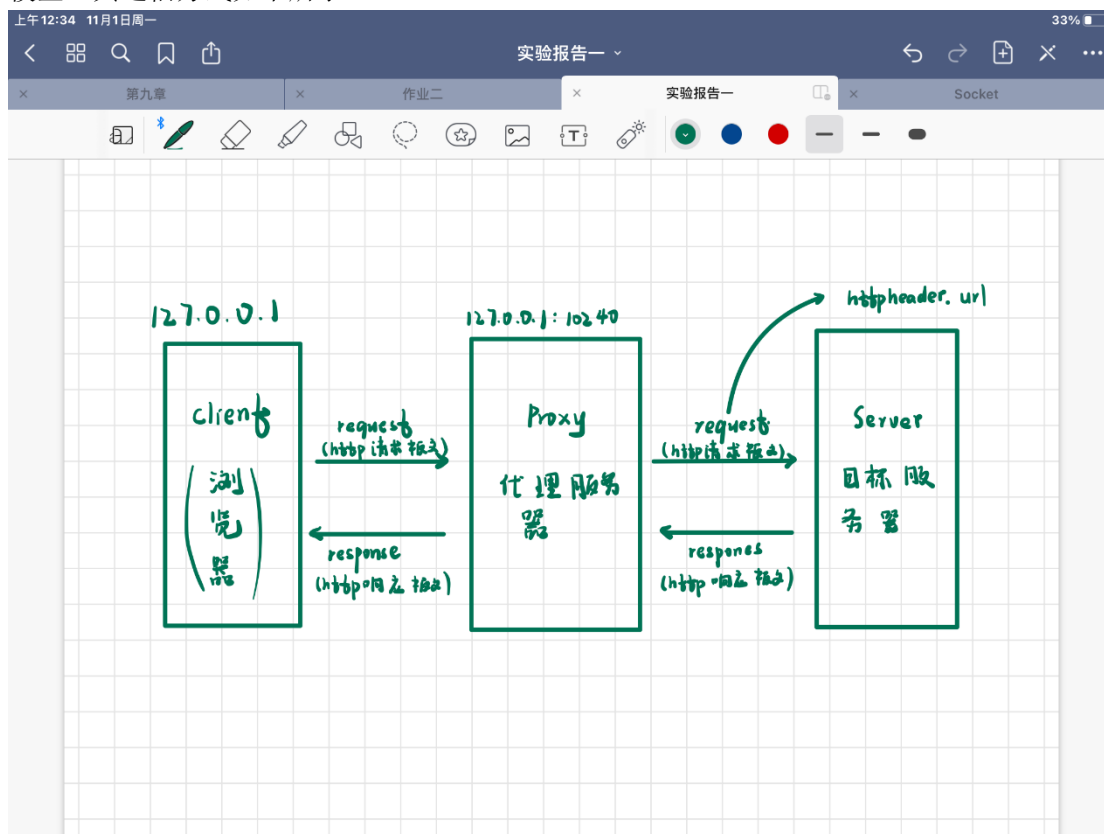
实验内容：

1. 设计并实现一个基本HTTP 代理服务器。要求在指定端口（例如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。要求能缓存原服务器响应的对象，并能够通过修改请求报文（添加if-modified-since头行），向原服务器确认缓存对象是否是最新版本。
3. 实现扩展功能：
 - a) 网站过滤：允许/不允许访问某些网站；
 - b) 用户过滤：支持/不支持某些用户访问外部网站；
 - c) 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）

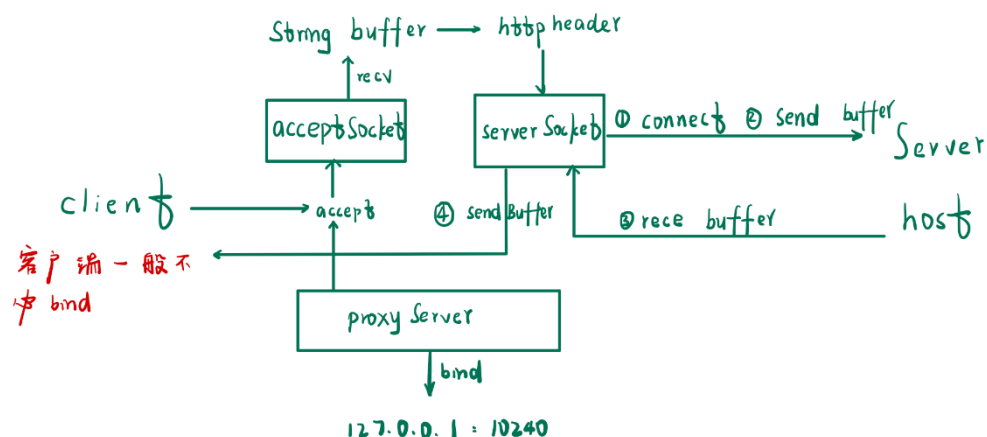
实验过程：

以文字描述、实验结果截图等形式阐述实验过程，必要时可附相应的代码截图或以附件形式提交。

1. 首先打开系统设置，为浏览器设置代理，使之将所有的网路请求发至本机10240端口。
2. 由于本次实验是初次Socket编程，因此首先完成了一个单线程，无拓展功能，无cache的基础模型。其通信方式如下所示。

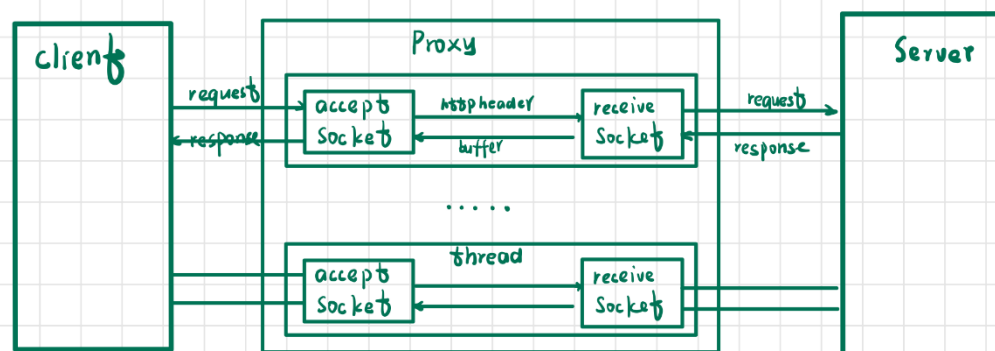


其中的Socket编程中用到的主要的函数关系为：



该种情况下只能处理一个请求，经多次尝试，服务器仅返回了空HTTP头部，并未返回主要的静态页面资源。但是该基础部分已经实现了客户，代理，服务器之间的通信，初步实现了代理功能。

- 在实现基础的通信之后，参考实验参考书以及网络上的资料，实现多线程的功能，即代理每通过accept收到一个请求，便开启一个线程，每一个线程中再初始化acceptSocket以接受用户输入，初始化serverSocket用来向服务器端发起请求，接受返回的数据，然后通过acceptSocket将返回的数据返回给客户。由此实现服务器多线程通信的功能。通信机制如下图所示。



由此成功实现了完整的代理功能，代理服务器可以成功返回完整的页面，基本复现了实验指导书上的代码。

- 在实现了完整的代理功能后，通过以下代码完成屏蔽特定网站，屏蔽特定用户，钓鱼三个扩展功能，主要的原理是代理服务器收到用户发来的请求后，分析HTTP请求消息的头部行，

若符合屏蔽规则拒绝转发至目标服务器，符合钓鱼规则构造新的HTTP请求消息，发给新的目标服务器。

```
HTTPHeader *httpheader = parseHttp(Buffer, receSize);

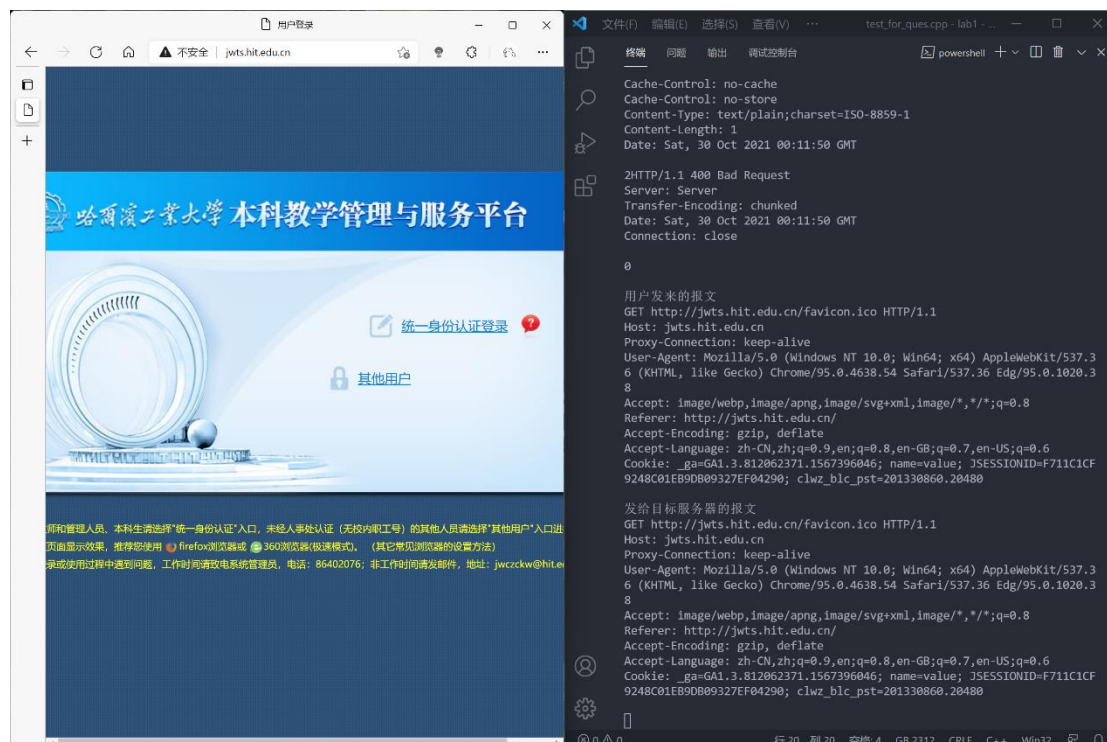
if (strcmp(httpheader->url, INVALID_WEBSITE) == 0)
{
    printf("\n=====\\n\\n");
    printf("-----Sorry!!!这个网站不能访问-----\\n");
    closesocket(acceptSocket);
    delete Buffer;
    return 0;
}

if (strstr(httpheader->url, FISHING_WEB_SRC) != NULL)
{
    printf("\n=====\\n\\n");
    printf("-----已从%s转移到%s -----\\n", FISHING_WEB_SRC, FISHING_WEB_DEST);
    memcpy(httpheader->host, FISHING_WEB_HOST, strlen(FISHING_WEB_HOST) + 1);
    memcpy(httpheader->url, FISHING_WEB_DEST, strlen(FISHING_WEB_DEST));
}
```

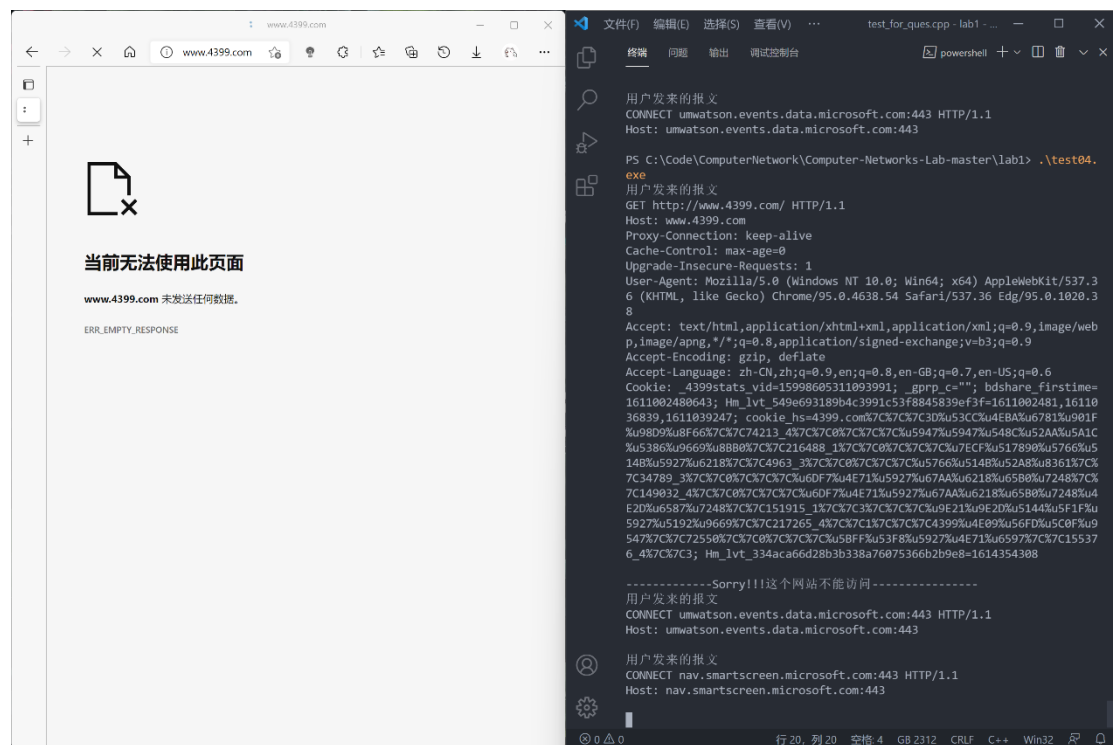
5. 完成上述功能后，最后增加cache功能。cache功能的原理：收到用户的HTTP请求报文后，首先根据URL判断本地有无缓存，若有缓存则首先读取缓存，得到缓存的更新时间，基于此在用户发来的HTTP请求消息头中加入if-modified-since行，然后转发给目标服务器，若目标服务器返回304，则将本地缓存返回给用户，否则将目标服务器发来的报文返回给用户，并更新本地缓存。如果本地没有缓存，则将用户请求HTTP报文发给目标服务器，得到目标服务器的响应消息后，除转发给用户外，还将相应消息缓存在本地，以URL为基础构造缓存名，方便后续检索。

实验结果：

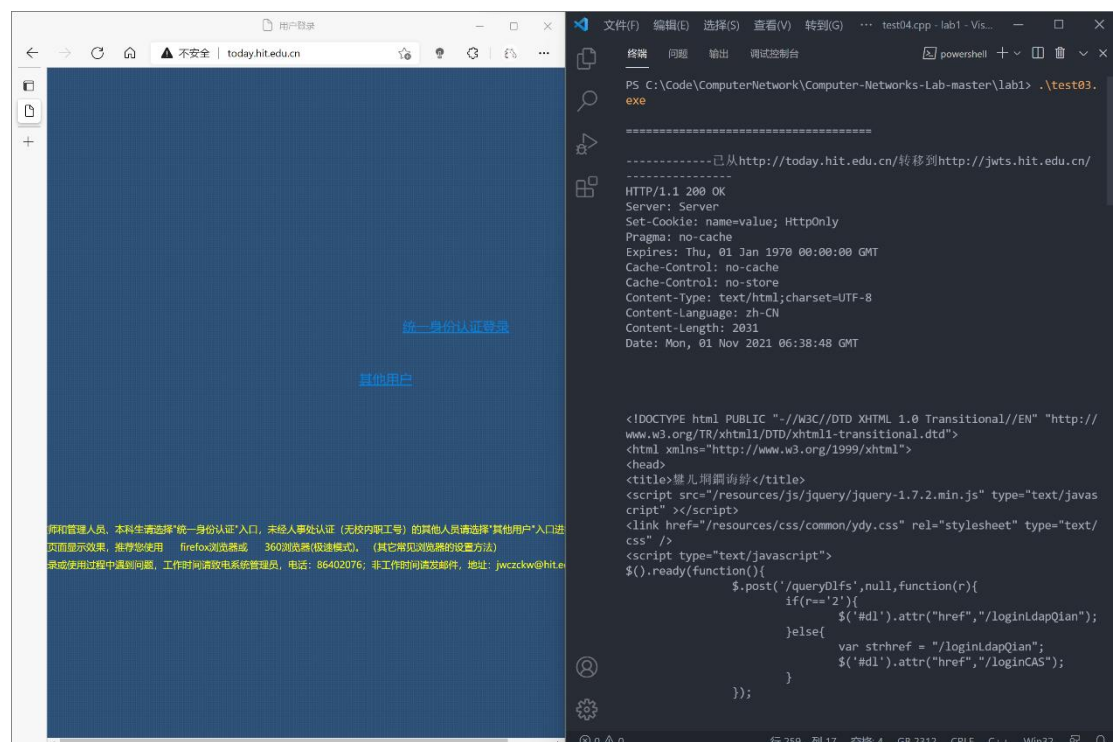
1. 基本的代理功能



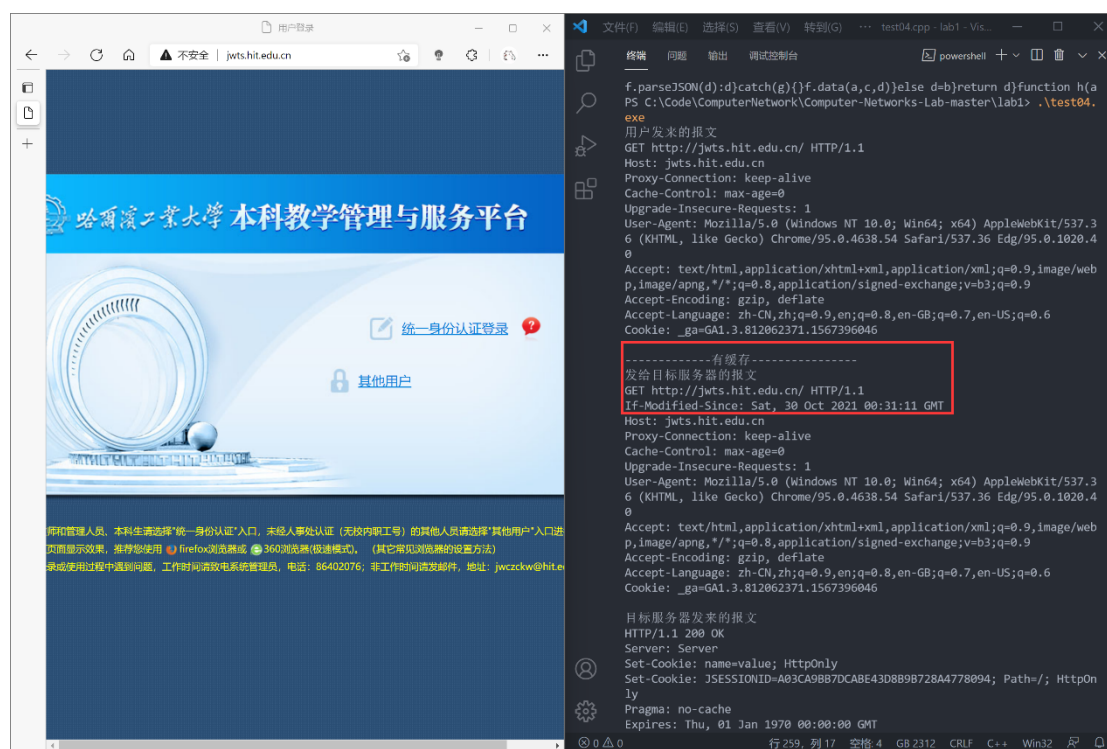
2. 屏蔽功能



3. 钓鱼功能



4. 缓存功能



问题讨论:

在上述实验过程中遇到了许多困难。

1. 实验环境中的困难，实验指导书代码错误颇多是我逐渐迭代，完善功能的主要原因。尤其是在环境配置上，这里介绍一下我的代码环境，C语言编译工具链为MinGW-W64-builds-4.3.5，编译环境为PowerShell，编译命令为gcc test01.c -o test01 -lws2_32，生成debug版本的命令为gcc test01.c -g -o test01 -lws2_32，可以使用MinGW工具链中的gdb进行调试，但是经测试gdb调试时display，layout命令不能正常使用，所以debug也可以在vscode等其他ide下进行，注意在debug的参数中加上-lws2_32。不建议使用tmain，正常int main()即可。
2. C语言中字符串的处理，申请内存后需要及时调用ZeroMemory初始化内存，字符串的长度（strlen方法返回值）不包括结尾的“\0”，使用字符串数组分配空间的情况下需要计算结尾的“\0”，而使用malloc则不必。若是使用字符串直接赋值的过程中出现了warning，可以加上const标记。strtok函数，注意使用方法。
3. accept函数会一直等待请求来临，虽然浏览器会自动发送一些请求，但是只要接收处理完了一般不会重复发送请求。

心得体会:

这次实验第一次接触socket编程，并且使用较为繁琐的C语言，不过也正如实验课上与老师交流的一样，对Socket通信的过程有了很清晰的认识，建议以后的实验中可以提供一些可运行的字符串处理函数。