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	王家琪		院系	人工智能	
班级	1903601		学号	1190301610	
任课教师	李全龙		指导教师		
实验地点			实验时间		
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：

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。要求在指定端口（例如8080）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。（选作内容，加分项目，可以当堂完成或课下完成）

(3) 扩展HTTP代理服务器，支持如下功能：（选作内容，加分项目，可以当堂完成或课下完成）

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：

一、 socket编程客户端和服务器端的主要过程

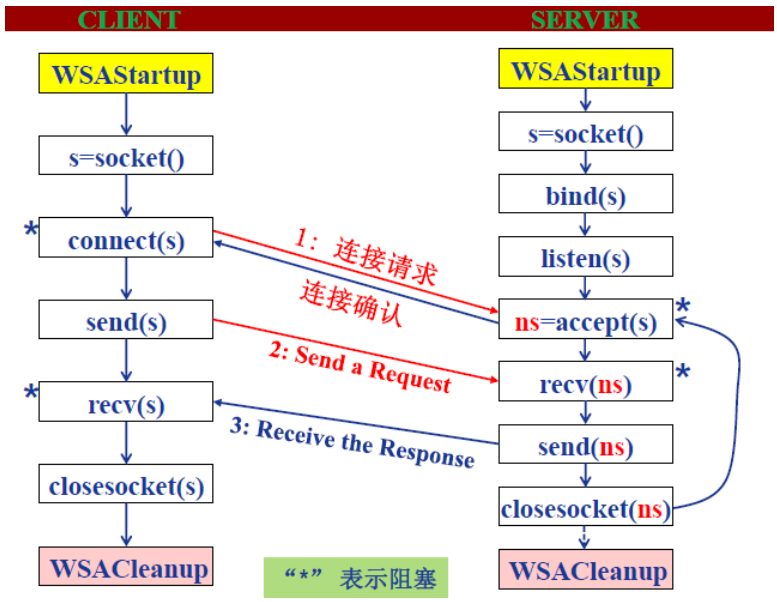
服务器段：

1) **初始化套接字，**和主机绑定端口，并开始监听
2) 接收到客户端的请求报文之后，进行**连接**确认，http协议要三次握手
3) 连接建立之后，和客户端进行**通信**，互发报文
4) 通信结束后，**关闭连接，重新返回到监听状态**
5) 服务器端停止工作，关闭socket

客户端：

1) **根据目标服务器的IP地址和端口号，建立socket，**并连接服务器（http需要进行3次握手）
2) 和服务器端进行**通信，**报文互发
3) 客户端通信完毕，关闭连接

如下图所示，套接字socket服务端和客户端工作时api调用流程：



二、 HTTP代理的基本原理

代理服务器，俗称“翻墙软件”，允许一个网络终端（一般为客户端）通过这个服务与另一个网络终端（一般为服务器）进行非直接的连接。如图1-2所示，为普通Web应用通信方式与采用代理服务器的通信方式的对比。

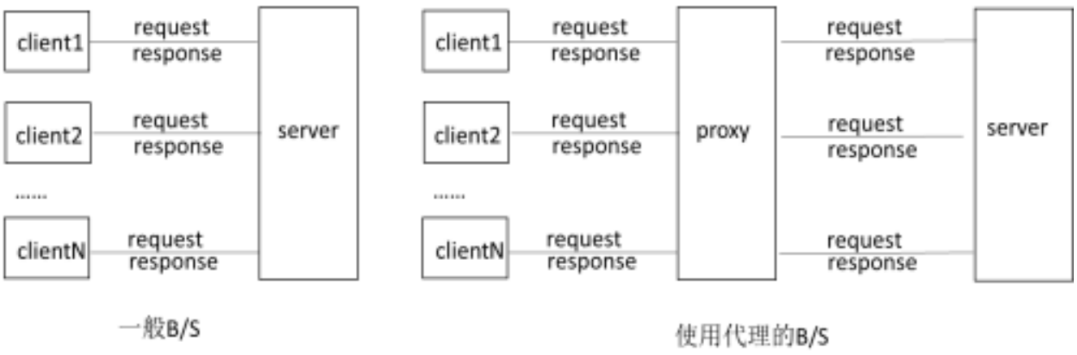


图 1-2 Web 应用通信方式对比

代理服务器在**指定端口（本程序是10240）监听**浏览器的访问请求（需要在客户端浏览器进行相应的设置），接收到浏览器对远程网站的浏览请求时，代理服务器开始在代理服务器的**缓存中检索URL对应的对象**（网页、图像等对象），找到对象文件后，提取该对象文件的最新被修改时间；代理服务器程序在客户的**请求报文首部插入<If-Modified-Since: 对象文件的最新被修改时间>**，并向原Web服务器转发修改后的请求报文。

如果代理服务器没有该对象的缓存，则会直接**向原服务器转发**请求报文，并将原服务器返回的响应直接转发给客户端，同时将对象缓存到代理服务器中。代理服务器程序会根据缓存的时间、大小和提取记录等对缓存进行清理。

http代理可以分为单用户代理和多用户代理两种方式。本实验采用的是多用户代理，**多用户的简单代理服务器可以实现为一个多线程并发服务器**。首先，代理服务器创建HTTP代理服务的TCP主套接字，通过该主套接字监听等待客户端的连接请求。当客户端连接之后，**创建一个子线程**，由子线程执行上述一对一的代理过程，服务结束之后子线程终止。与此同时，主线程继续接受下一个客户的代理服务。

三、 程序设计方案

1. InitSocket 初始化套接字
    a. socket 创建套接字
    b. bind 绑定套接字
    c. listen 转到监听模式
2. accept 接受连接请求
3. 判断用户是否在禁止名单中，如果是，返回 2
4. ProxyThread 线程执行函数
    a. recv 接受客户端的数据
    b. ParseHttpHead 解析 TCP 报文中的 HTTP 头部
    c. Replace 网址过滤，将禁止访问网站设为空，钓鱼网站替换 url 和 host。
    d. ConnectToServer 根据主机创建目标服务器套接字，并连接
        i. socket 创建套接字
        ii. connect 连接目标服务器
    e. send 将客户端发送的 HTTP 数据报文直接转发给目标服务器

f. recv 接受目标服务器返回数据
g. 判断接受的报文是否在 cache 里面，是否需要更改
h. send 将目标服务器返回的数据直接转发给客户端

5. closesocket 关闭套接字

需要补充的知识是 http 的状态响应码，除了我们熟知的 404 not found，200 OK 等等，还有一些涉及到时间变化的状态码。还有 304 not modified 码，代表的是如果客户端发送了一个带条件的 GET 请求且该请求已被允许，而文档的内容（自上次访问以来或者根据请求的条件）**并没有改变**，则服务器应当返回这个状态码。304 响应禁止包含消息体，因此始终以消息头后的第一个空行结尾。

四、 关键技术解析

1、 http 代理主要任务

主要参考以下几个函数：

a) BOOL InitSocket()

首先加载套接字库，使用以下几个 socket 函数 socket(AF_INET, SOCK_STREAM, 0); bind(ProxyServer, (SOCKADDR*)&ProxyServerAddr, sizeof(SOCKADDR))；和 listen(ProxyServer, SOMAXCONN)。实现了服务器流程中的 socket 和 bind 和 listen。

b) BOOL ParseHttpHead(char *buffer, HttpHeader * httpHeader)

对请求报文的头部文件 buffer 进行解析，得到请求报文中的 method, url, host 和 cookie 等，用于 ConnectToServer 函数与目标服务器建立连接。

c) BOOL ConnectToServer(SOCKET *serverSocket, char *host)

使用 socket 创建套接字，connect 连接至目标服务器

d) unsigned int __stdcall ProxyThread(LPVOID lpParameter)

实现了从客户端接收请求报文，向服务器发送请求报文，从服务器接收响应报文，向客户端送响应报文。通过 ParseHttpHead 函数基对请求报文头部进行解析，然后将得到的头部文件作为 ConnectToServer 函数与目标服务器建立链接。连接成功后，便将请求报文发送过去，接收收到响应报文，然后发送响应报文给浏览器即可。

2、 网页过滤

网页过滤的结果是由 map 储存的：

```
1. map<char*, char*, ptrCmp> transfer = {
2.  {host, blank},
3.  {host1, host2}
4. };
```

如果检测到了 socket 里面的 host 为 transfer 里面的 key 值，就会调用 replace 函数，在 socekt 暂时缓存的代理服务器的 buffer 里面，替换 host。Replace 函数的关键部分如下：

网站过滤和引导：

```
1. if (transfer.find(httpHeader->host) != transfer.end()) {
2. replace(Buffer, httpHeader->host, transfer[httpHeader->host]);
3. memcpy(httpHeader->host, transfer[httpHeader->host],
4. strlen(transfer[httpHeader->host]) + 1);
5.  }
```

3、 缓存技术

为了实现网页在代理服务器的缓存，我们首先定义 cache 的结构，类似于 http 的头部结构，存储 url，上次更改时间，缓存的内容 buffer，并初始化为 0。设置 cache 数组作为缓存。

```
struct Cache {
 char url[1024];  //url 地址
 char time[40];   //上次更新时间
 char buffer[MAXSIZE];   //缓存的内容
 Cache() {
  ZeroMemory(this, sizeof(Cache));
 }
}cache[CACHE_NUM]
```

在接收到用户请求后，首先判断请求是否为 GET 请求，只有 GET 请求才能缓存：

```
if (!strcmp(httpHeader->method, "GET"))
```

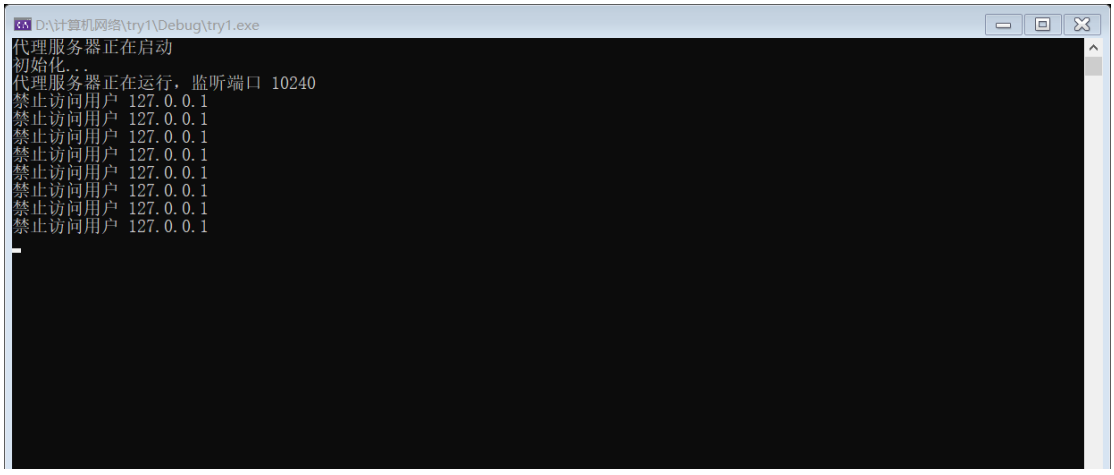
如果是 GET 请求，再根据报文中的 url 到 Cache 中寻找，如果 Cache 未命中，直接向服务器发送请求，并将服务器返回的 url 对象存到 Cache 中，如果命中，在原请求报文的基础上加入 If-Modified-Since 头行，再向服务器发送请求，如果服务器返回 304，则代表 Cache 是最新的，直接将 Cache 返回给客户即可，如果服务器返回 200，则代表 Cache 需要更新，用服务器返回的报文更新 Cache，再返回给客户即可。

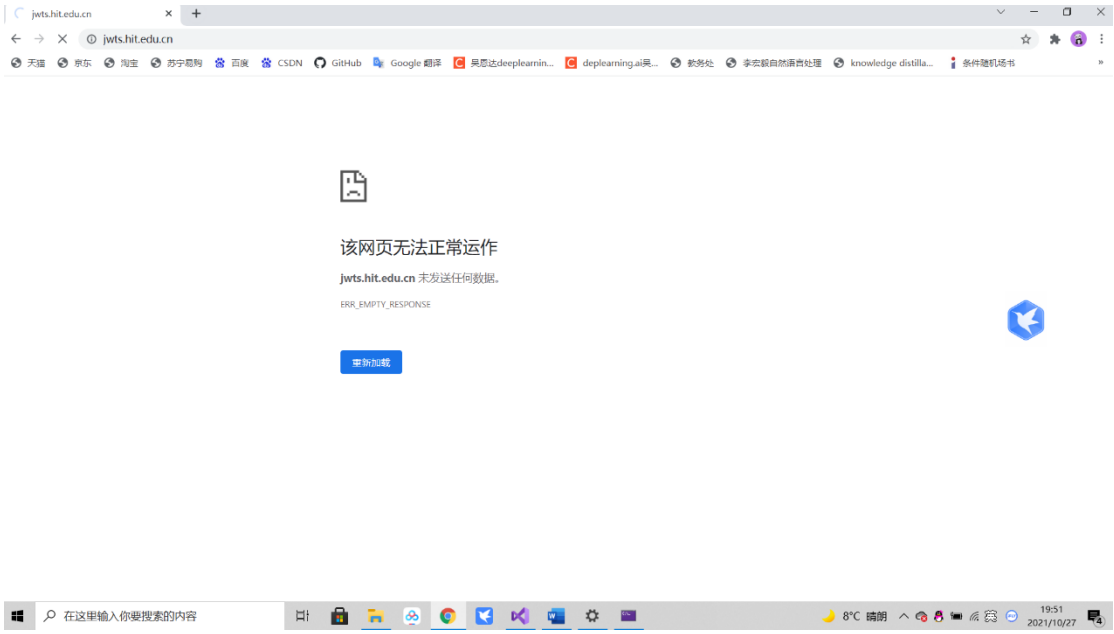
```
if (strlen(cache[i].url) != 0 &&
        !strcmp(cache[i].url, httpHeader->url)) {
    printf("cache 命中，
url=%s, time=%s\n", httpHeader->url, cache[i].time);
 if (!memcmp(&Buffer[9], "304", 3)) {//决策是 304
  ret = send(((ProxyParam*)lpParameter)->clientSocket, cache[i].b
uffer,
  sizeof(cache[i].buffer), 0);
 }
 else {
  if (!strcmp(httpHeader->method, "GET") &&
        !memcmp(&Buffer[9], "200", 3)) {
   char Buffer2[MAXSIZE];
   memcpy(Buffer2, Buffer, sizeof(Buffer));
   const char* delim = "\r\n";
   char* ptr;
   char* p = strtok_s(Buffer2, delim, &ptr);//分割字符串
   bool flag = false;
   while (p) {
    if (strlen(p) >= 15 && !memcmp(p, "Last-Modified: ", 15)) {
     flag = true;
```

```
20.      break;
21.     }
22.     p = strtok_s(NULL, delim, &ptr);
23.    }
24.    //添加缓存
25.    if (flag) {
26.     printf("添加缓存\n");
27.     last_cache++;
28.     last_cache %= CACHE_NUM;
29.     memcpy(cache[last_cache].url, httpHeader->url,
30.       sizeof(httpHeader->url));
31.     memcpy(cache[last_cache].time, p + 15, strlen(p) - 15);
32.     memcpy(cache[last_cache].buffer, Buffer, sizeof(Buffer));
33.     printf("\n 添加的缓存\n");
34.     printf("%s\n", cache[last_cache].url);
35.     printf("%s\n", cache[last_cache].time);
36.     printf("%s", Buffer);
37.     printf("\n-----------------------------------------------\n");
38.    }
39.    //将目标服务器返回的数据直接转发给客户端
40.   }
41.   printf("\n 代理服务器向用户发送数据\n");
42.   printf("%s", Buffer);
43.   printf("\n-----------------------------------------------\n");
44.   ret = send(((ProxyParam*)lpParameter)->clientSocket, Buffer,
45.    sizeof(Buffer), 0);
46. }
```

实验结果：

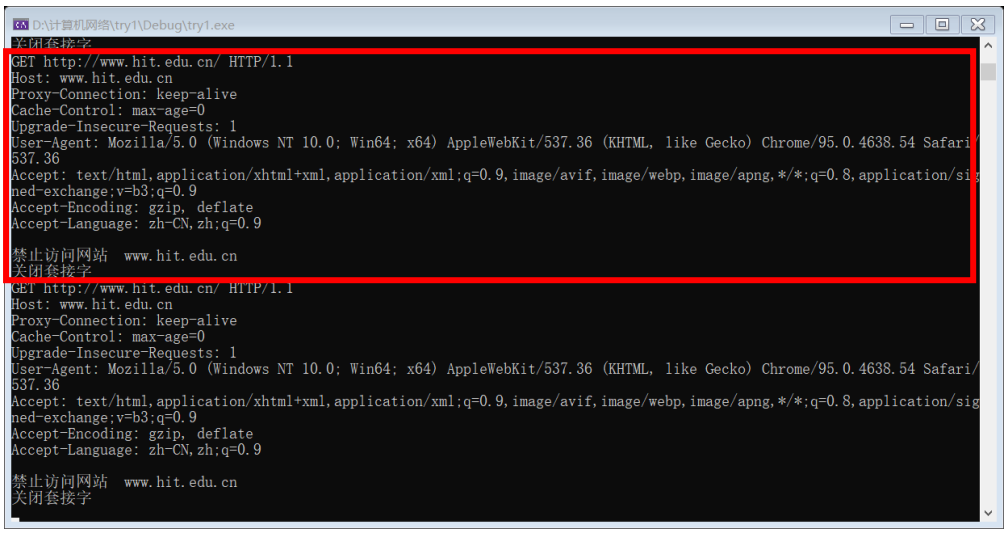
1、禁止用户访问。禁止IP为127.0.0.1的用户进行访问，效果如下：命令行显示，网页无法连接



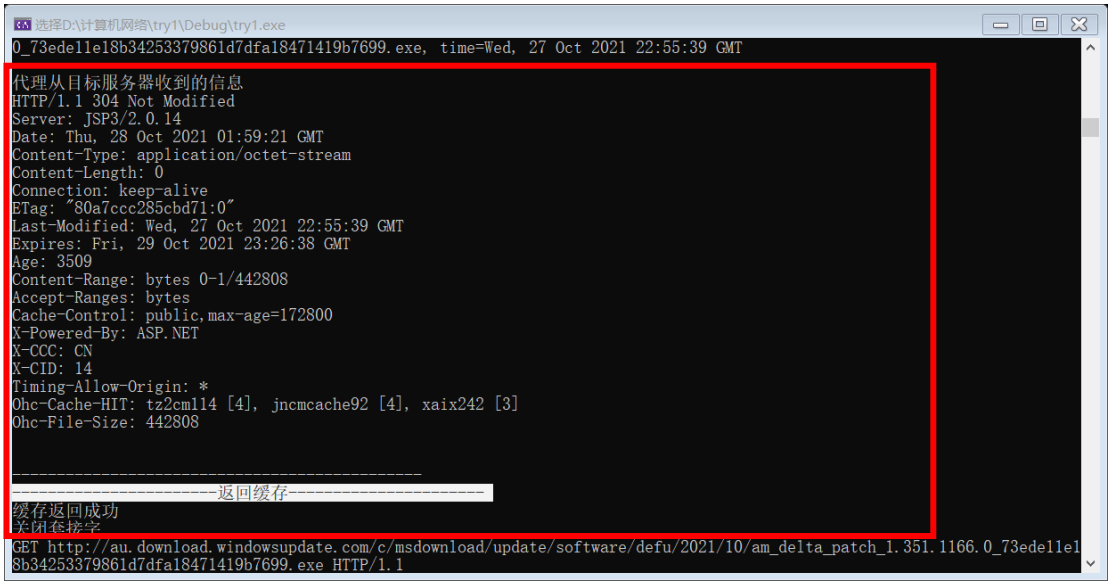


2、http代理功能，访问today.hit.edu.cn，可以正常访问。注意本http代理服务器只可以访问http开头的网站，https的网站是不能访问的。

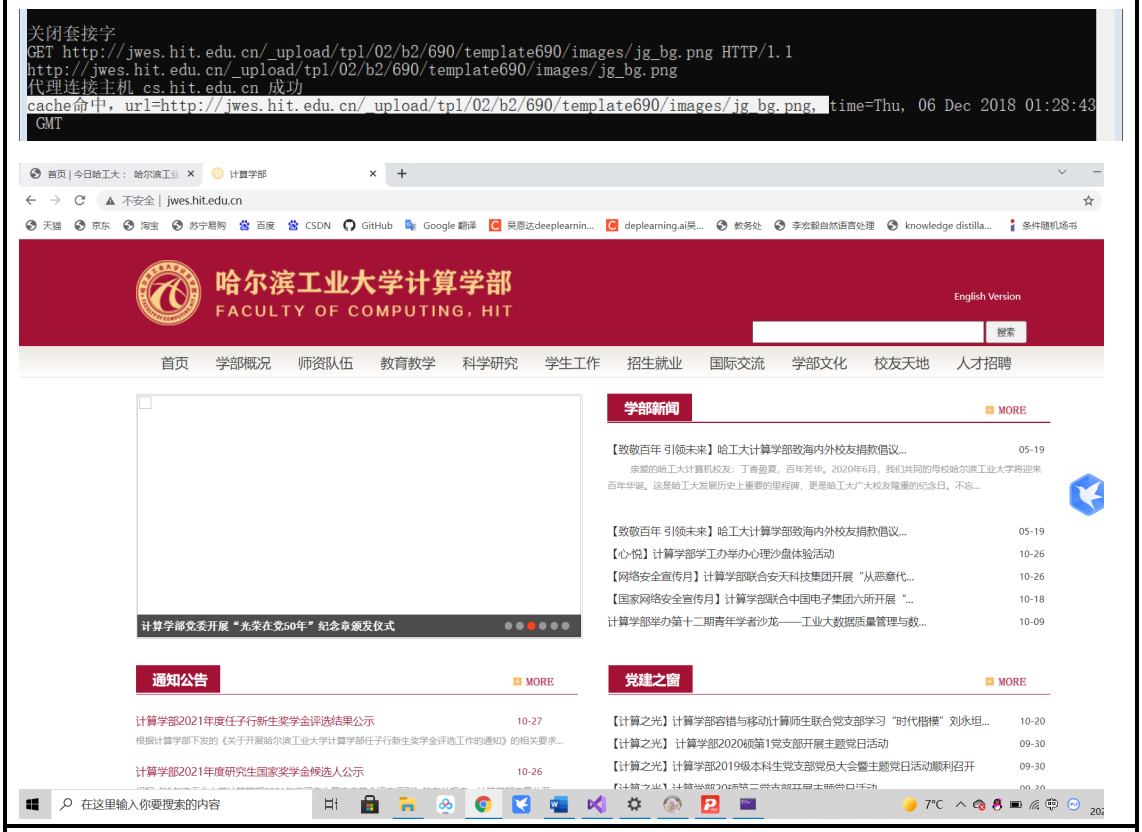
3、禁止访问网站，这里选择的是www.hit.edu.cn，发现网站无法正常访问，命令行出现了：禁止访问网站的提示。



4、缓存网页。如下图所示，代理服务器收到了目标服务器的请求报文，编号是304，证明可以在缓存中找到报文，所以返回报文。



5、网站钓鱼。如下图所示，访问的是jwes.hit.edu.cn，但是我们把他转接到了cs.hit.edu.cn的网站上。命令行显示代理链接主机成功，网页正常显示。



问题讨论：

1、发现老师给的代码有一些问题：比如#include "stdafx.h"这个函数现在已经放在预编译里面了，不用再引用了；goto error函数后面不能有初始化操作，要把httpHeader的初始化放在前面等等。

2、在实验的时候，遇到了很多不能理解的bug，比如有的时候代理连不上，有的时候关掉代理了但是还是显示了钓鱼的结果，这种问题挺玄学的……

3、有很多的函数都不经常用，但是在本代码中都出现了，从网上查到了相关用法和解析，大概整理到附录里面了。

心得体会：

经过此次实验，熟悉了 Socket 网络编程，清楚客户端和服务器之间 Socket 通信过程；掌握了 HTTP 代理服务器的基本工作原理；同时了解了钓鱼网站，禁止用户，禁止网站以及 Cache 等的原理。让我对网络编程更感兴趣。

附录：

1、**sockaddr 和 sockaddr_in 详解**

https://blog.csdn.net/will130/article/details/53326740

2、void *memset(void *s, int c, unsigned long n);函数的功能是：将指针变量 s 所指向的前 n 字节的内存单元用一个“整数” c 替换，注意 c 是 int 型。

3、accept 函数

https://blog.csdn.net/stpeace/article/details/13424223

4、inet_ntoa

https://baike.baidu.com/item/inet_ntoa%28%29/10082005

5、set<i>find（）

https://vimsky.com/examples/usage/set-find-function-in-c-stl.html

6、accept https://blog.csdn.net/stpeace/article/details/13424223

7、string::npos https://blog.csdn.net/jiejinquanil/article/details/51789682