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	傅浩东		院系	软件工程	
班级	1937102		学号	1190202105	
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					

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；了解代理服务器的 Cache 缓存功能；熟悉代理服务器的钓鱼、禁止用户（用户过滤）以及禁止访问特定网站（网站过滤）等；掌握 HTTP 代理服务器设计与编程实现的基本技能。

实验内容：

概述本次实验的主要内容，包含的实验项等。

(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。

(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) 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。

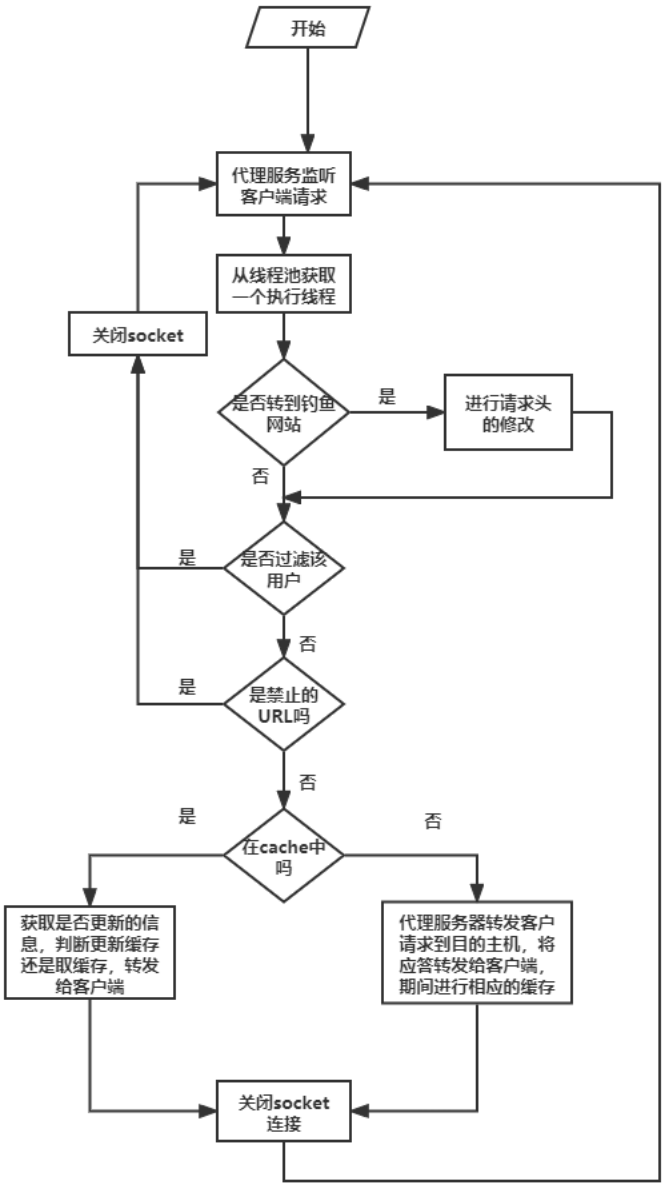
一．HTTP代理服务器基本原理

在计算机网络中，**代理服务器**是一种服务器应用程序，它充当请求资源的客户端和提供该资源的服务器之间的中介。客户端不是直接连接到可以满足请求资源（例如文件或网页）的服务器，而是将请求定向到代理服务器，代理服务器评估请求并执行所需的网络事务。这是一种简化或控制请求复杂性的方法，或提供额外的好处，例如负载平衡、隐私或安全性。代理被设计为向分布式系统添加结构和封装。因此，代理服务器在请求服务时代表客户端运行，可能会掩盖对资源服务器的请求的真实来源。

二．HTTP代理服务器主要流程

当客户端对 web 服务器请求时，此端提出请求时，此请求会首先发送到代理服务器；代理服务器接收到客户端请求后，会检查是否过滤网站、是否过滤用户以及是否钓鱼；紧接着，代理服务器会检查缓存中是否存有客户端所需要的数据；如果代理服务器没有客户端所请求的数据，它将会向 WEB 器提交请求；WEB 服务器响应请求的数据，代理服务器向客户端转发 Web 服务器的数据；若代理服务器查找缓存记录，确认已经存在 WEB 服务器的相关数据，代理服务器直接回应查询的信息，而不需要再去服务器进行查询，从而达到节约网络流量和提高访问的速度目的。

将这个过程概况，可以有下图所示：



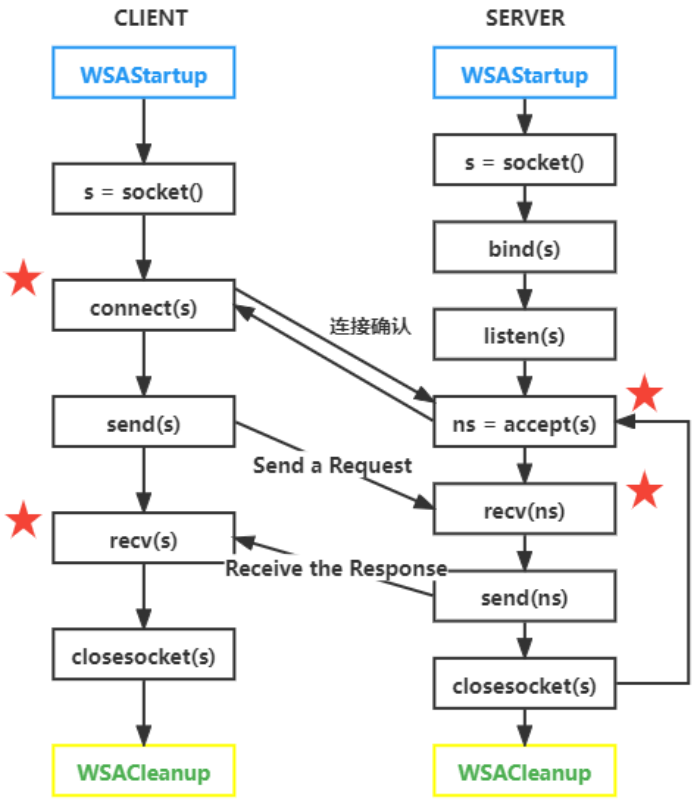
三．具体实现

1. 基本功能

服务器端：首先建立一个主套接字socket；接着调用bind函数，将主socket绑定IP地址和端口号；调用listen函数，持续监听主socket，以将接收到的“客户端连接请求”放入队列；调用accept函数，从队列获取请求；若无请求便会阻塞，直到接收到请求，然后返回一个Socket用于和客户端通信；通过“三次握手”建立TCP连接；调用IO函数和客户端双向通信；通信结束后，关闭accept返回的socket。

客户端：创建Socket；调用connect函数，向服务器发起连接请求；当服务器通过accept函数成功接收到请求之后，双方进行“三次握手”完成TCP连接；调用IO函数和服务器双向通信；通信结束后，关闭Socket。

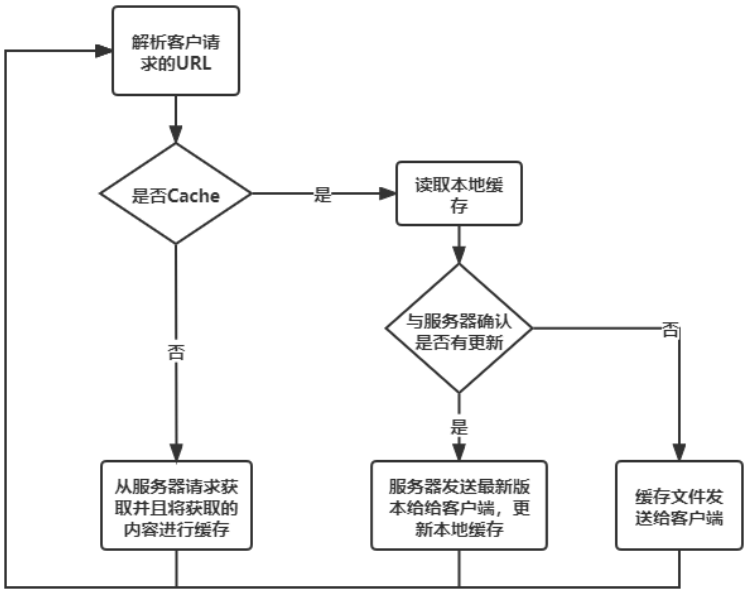
网络应用的Socket API(TCP)调用基本流程可以概括为如下流程图：



2. 缓存功能

代理服务器在指定端口（例如：8080）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，并向原Web服务器转发修改后的请求报文。构造条件CET方法(增加if-modified-since头部)访问目的服务器，如果得到的结果为304，说明缓存的就是最新的文件，并且不再从目的服务器获得请求文件，而是由本地磁盘将信息发送给源主机；否则，目标文件已经发生了变化或者缺少Last-Modified头部，此时请求的对象很有可能发生了改变，向目的服务器发送请求，并更改本地缓存。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。

将这个过程概括总结为下图：



3. 网站过滤

当解析客户端请求的UPL后，判断URL是否与disabledHost中的网址有重合，如果有则禁止访问，否则运行访问并且返回服务器的内容给客户端（也有可能是缓存）。

关键代码：

```
// 网站屏蔽
if (SiteFilter(httpHeader->url)) {
    printf("Site %s is banned\n", httpHeader->host);
    goto error;
}

bool SiteFilter(char *host) {
    for (int i = 0; i < DISABLED_MAXSIZE; i++) {
        if (disabledHost[i] == NULL)
            continue;
        if (strcmp(disabledHost[i], host) == 0)
            return true;
    }
    return false;
}
```

4. 用户过滤

将Proxy的套接字绑定IP地址，监听连接请求的IP地址是否在disabledUser中，若是则断开socket连接，结束此次进程，否则继续接下来的进程。

关键代码：

```
if (UserFilter(acceptAddr.sin_addr) == true) {
    printf("The user is Banned\n");
    printf("Closing socket...\n\n");
    continue;
}
```

```
bool UserFilter(in_addr sin_addr) {
    for (int i = 0; i < DISABLED_MAXSIZE; i++) {
        if (disabledUser[i] == NULL)
            continue;
        if (strcmp(disabledUser[i], inet_ntoa(sin_addr)) == 0)
            return true;
    }
    return false;
}
```

5. 网站引导（钓鱼）

当检测完是否屏蔽用户以及是否屏蔽网站后，检测是否钓鱼，若是则修改HTTP报文的URL和HOST，将钓鱼内容返回给客户端。

关键代码：

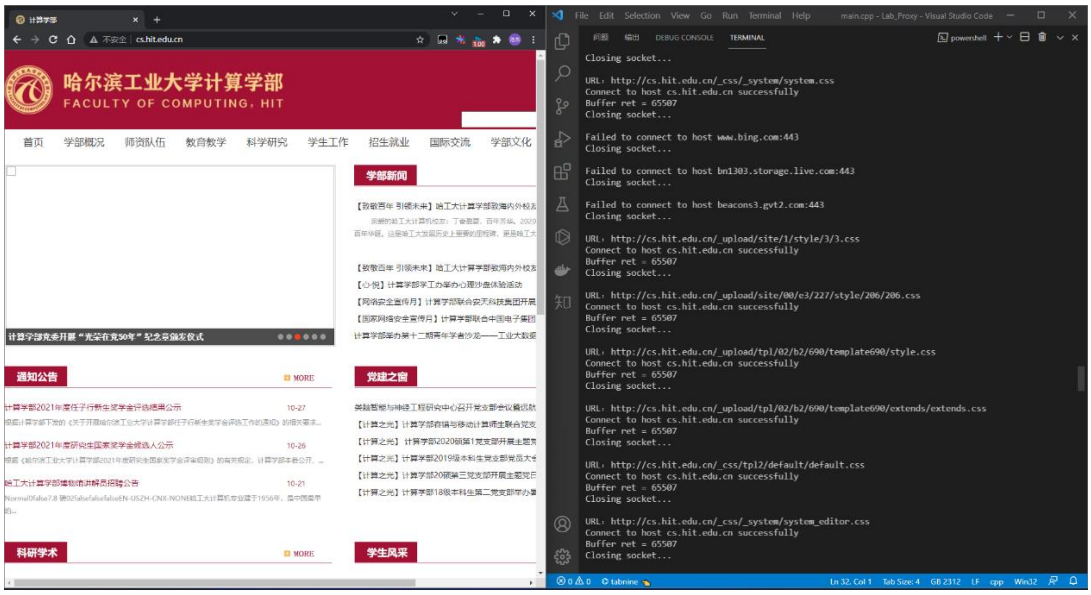
```
#define PHISHING_WEB_SRC "http://jwc.hit.edu.cn/"    // 钓鱼原网址
#define PHISHING_WEB_DEST "http://jwts.hit.edu.cn/" // 钓鱼目的网址
        char phishing_dest[] = "Location: ";
        strcat(phishing_dest, PHISHING_WEB_DEST);
        strcat(phishing_dest, "\r\n\r\n");
        phishing_len = strlen(phishing_dest);
        memcpy(pr, phishing_dest, phishing_len);
```

实验结果：

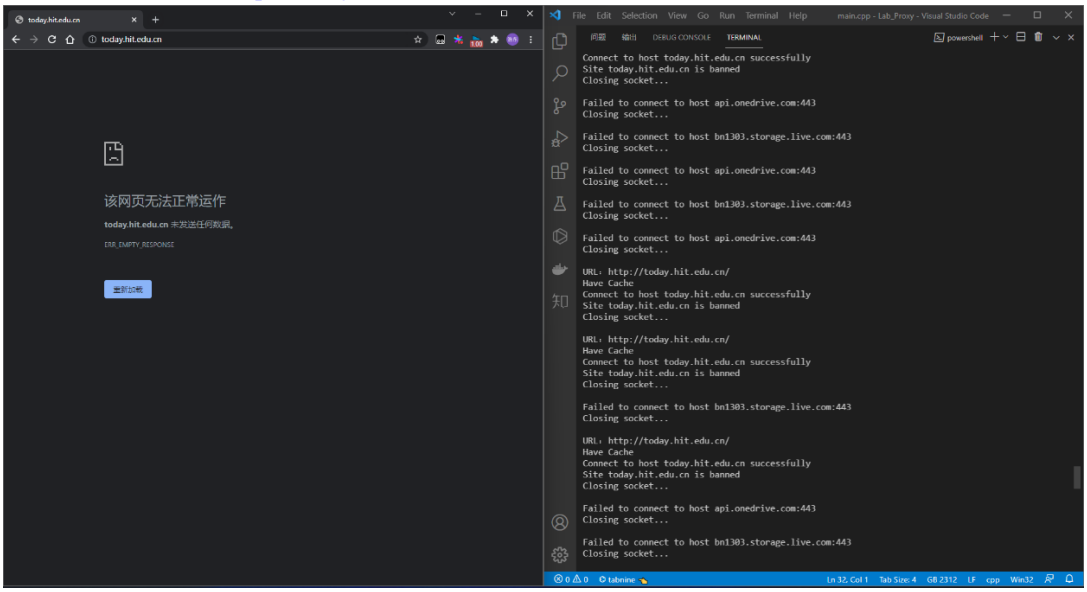
采用演示截图、文字说明等方式，给出本次实验的实验结果。

将代理服务器设置为本地，端口号为10240：

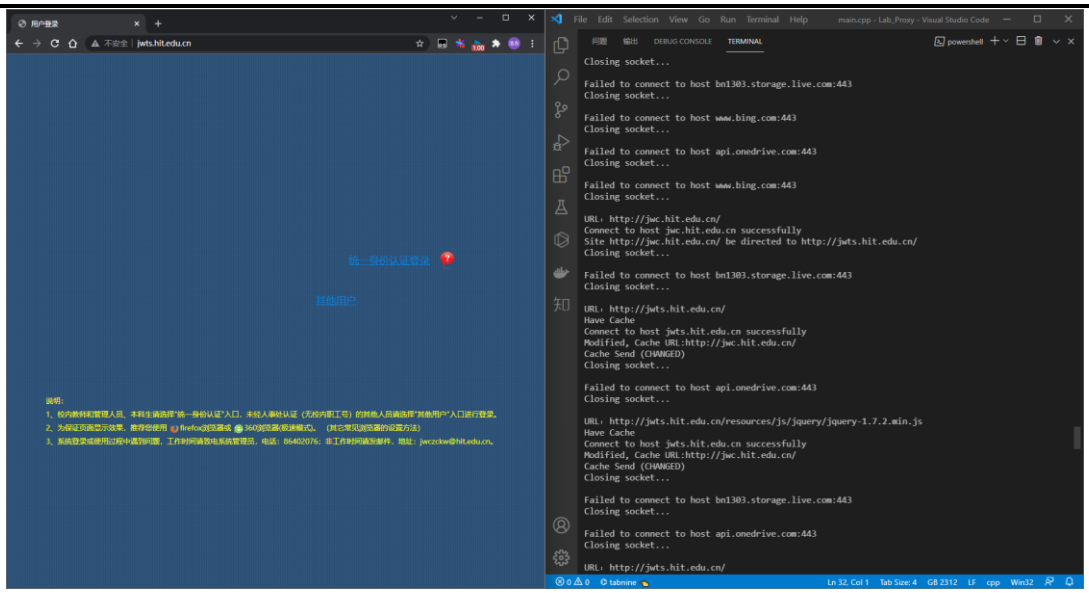
基本功能实现，访问计算学部官网：



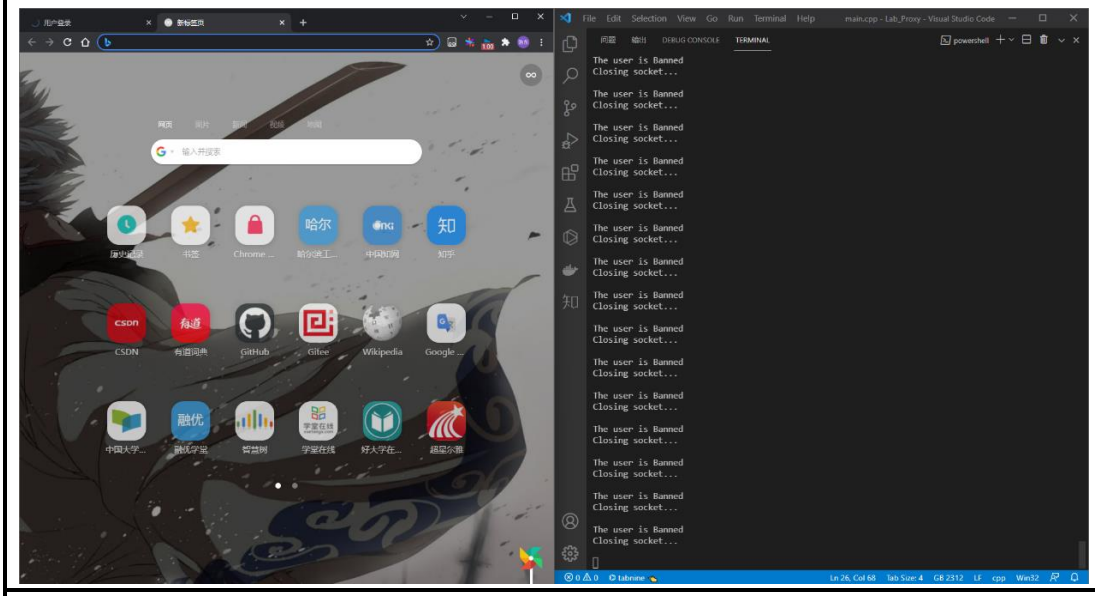
网站过滤，将网站http://today.hit.edu.cn/ 设为屏蔽网站，访问结果：



钓鱼，将http://jwc.hit.edu.cn/跳转到http://jwts.hit.edu.cn/ ，结果如下：



用户过滤，将自己本地过滤，即将127.0.0.1加入屏蔽用户，结果有：



问题讨论：

1. 使用Socket 相关的API，必须在CPP源文件中添加 #include <winsock2.h> 以及 #pragma comment(lib,"Ws2_32.lib") ，以确保导入了所需的库。
2. 对于缓存，可以缓存为文件保存到本地，而在这里我直接分配了一定的空间来存储缓存内容，并且会根据最新情况覆盖很早之前的缓存。
3. 本地主机的IP为127.0.0.1。
4. 实验中，关于钓鱼和禁用网站的实现可以直接丢弃相应的请求报文，并且关闭socket，但是也可以返回一个HTML文件，避免持续向服务器申请。

心得体会：

结合实验过程和结果给出实验的体会和收获。

本次实验，让我对socket编程有了初步的了解，进一步理解了基于TCP连接的通信过程，掌握了HTTP代理服务器的基本原理，对HTTP请求和响应原理有了更深的认识；同时，也对钓鱼功能、网站屏蔽等有了进一步的了解。

我使用C++（C）编写此次试验，它在处理HTTP段结构的时候繁杂但是又很细节，这更有助于我理解一些HTTP协议细节。

实验中实现的式基本的HTTP代理服务器，仅仅能实现的是对于HTTP协议的某些网站的访问，对于一些HTTPS协议的网站，还无法处理。与实际中所使用的代理服务器相比差距还很大，但是通过实现基本的代理服务器的功能，了解了代理服务器的基本工作原理，为了解socket编程提供了很多帮助。但也初步了解到除了GET、HOST等命令，HTTPS协议中还存在CONNET等。