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	王昊峰		院系	计算学部	
班级	1903601		学号	1190201004	
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	周六 3、4 节	
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。
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：

一、 服务器主体部分

服务器的主体流程如下：

```
int _tmain(int argc, char** argv)
{
    printf("Proxy Server starting...\n");
    printf("Initializing...\n");

    if(!InitSocket())
    {
        printf("Failed to initialize socket\n");
        return -1;
    }
    printf("Listening on port %d\n", ProxyPort);
    SOCKET acceptSocket = INVALID_SOCKET;
    ProxyParam *lpParameter;
    HANDLE hThread;
    DWORD dwThreadID;
    sockaddr_in peerAddr;
    int peerLen;
    //读取名单
    ReadList("black.txt", &black);
    ReadList("fishing.txt", &fish);
    ReadList("blackip.txt", &blackip);
    //代理服务器不断监听
    while(true)
    {
        acceptSocket = accept(ProxyServer, NULL, NULL);
        getpeername(acceptSocket, (struct sockaddr *)&peerAddr, &peerLen);
        if(InList(inet_ntoa(peerAddr.sin_addr), &blackip))
            continue;
        lpParameter = (ProxyParam *)malloc(sizeof(ProxyParam));
        if(lpParameter == NULL)
            continue;
        lpParameter->clientSocket = acceptSocket;
        hThread = (HANDLE) _beginthreadex(NULL, 0, &ProxyThread, (LPVOID)lpParameter, 0, 0)
        CloseHandle(hThread);
        Sleep(200);
    }
    closesocket(ProxyServer);
    WSACleanup();
    return 0;
}
```

1. 初始化作为服务器所使用的套接字并监听端口

```
/*****************************************************
 * Funtion name :   InitSocket
 * Description  :   初始化服务器 socket
 * Return       :   初始化成功返回 true，否则返回 false
*****************************************************/
bool InitSocket()
```

首先使用WSAStartup函数初始化微软的socket动态库，之后使用socket函数新建一个套接字，并利用bind函数将套接字与本地的端点信息绑定，再利用listen函数使主机持续监听端口。设置监听端口2080。

2. 不断监听端口2080

当有客户端向2080端口发起连接请求后，服务器利用accept函数与客户端建立连接，并创建一个新线程用于处理客户端的报文，主线程继续等待其他客户端的请求。

二、 代理服务器线程

```
/*****************************************************
 * Funtion name :   ProxyThread
 * Description  :   执行线程
 * Parameters   :   LPVOID lpParameter
 * Return       :   unsigned int
*****************************************************/
unsigned int __stdcall ProxyThread(LPVOID lpParameter)
```

代理服务器首先使用recv函数接收来自客户端的报文，并解析http报文头部的基本信息，包括目的主机的url，请求的文件、cookie等信息。

```
/*********************************************************************
 * Funtion name :   ParseHttpHead
 * Description  :   解析 TCP 报文中的 HTTP 头部并存入 httpHeader 指向的空间中
 * Parameters   :   char *buffer, HttpHeader *httpHeader
 * Return       :   unsigned int
*********************************************************************/
void ParseHttpHead(char *buffer, HttpHeader *httpHeader)
```

之后，代理服务器利用解析出的信息，建立一个新的socket用于与客户端的目标主机建立连接。

```
/*****************************************************
 * Funtion name :   ConnectToServer
 * Description  :   根据主机创建目标服务器套接字，并连接
 * Parameter    :   SOCKET *serverSocket, char *host
 * Return       :   成功返回 true，否则返回 false
*****************************************************/
bool ConnectToServer(SOCKET *serverSocket, char *host)
```

连接建立后，服务器将从客户端收到的报文用send函数转发给目的主机，并利用recv接受目的主机返回的报文，再用send函数将报文转发给客户端。至此完成了一次报文转发工作，实现了代理服务器的基本功能。

三、 附加功能

1. 网站过滤

首先，在磁盘中存储着一个网站黑名单。初始化的时候将黑名单中的网站读入内存中，并使用vector存储。当代理服务器解析出目的主机的地址后，服务器会将地址与黑名单对比：若目的主机在黑名单中，提前终止代理进程。

2. 用户过滤

基本流程与网站过滤流程相同。在磁盘中存储着一个用户黑名单。初始化的时候将黑名单中的IP读入内存中，并使用vector存储。当客户端发起连接请求后，服务器会将客户端IP与黑名单对比：若IP在黑名单中，提前终止代理进程。

3. 网站引导

网站引导只需将报文重定向到修改后的目的主机即可。为实现这个功能，我对每个报文中的目的主机字段、url字段进行修改，将其中的原目的主机修改为需要的目的主机。

```
/*****************************************************
 * Funtion name :   ChangeHttpHead
 * Description  :   修改 http 报文头部信息
 * Parameter    :   char *buffer, char *url, char *host
*****************************************************/
void ChangeHttpHead(char *buffer, char *url, char *host)
```

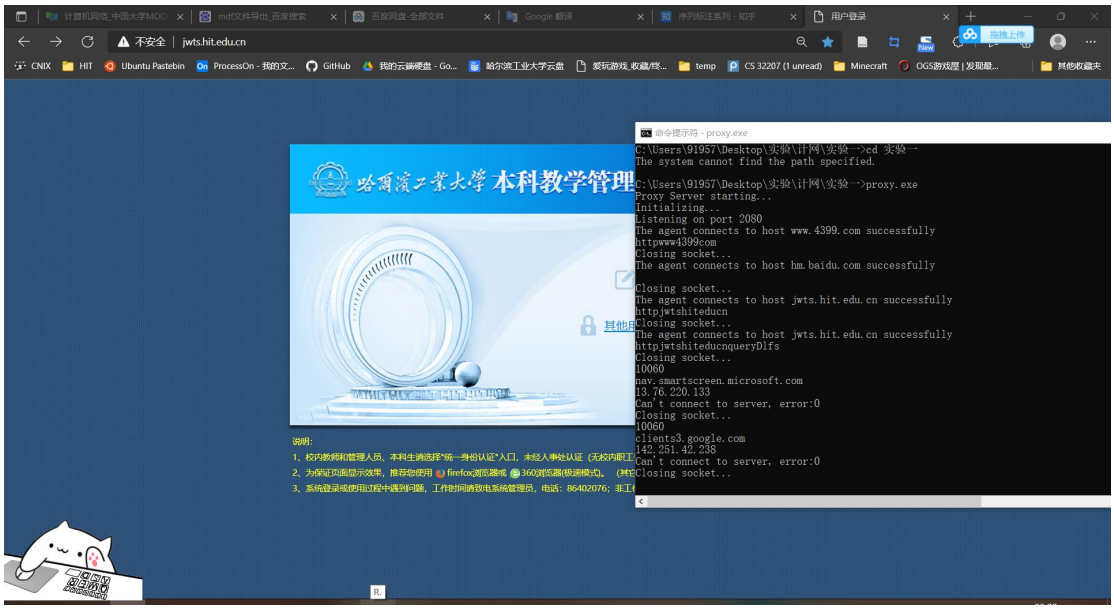
4. cache

对于cache的实现，最重要的是确定当前的缓存内容是否是为最新，因此需要在报文中插入if-modified-since 字段来向目的主机确定当前缓存是否是最新版本。若目的主机返回304，则表明当前cache是最新版本，直接将cache内容转发给客户端即可，否则需要建立缓存。

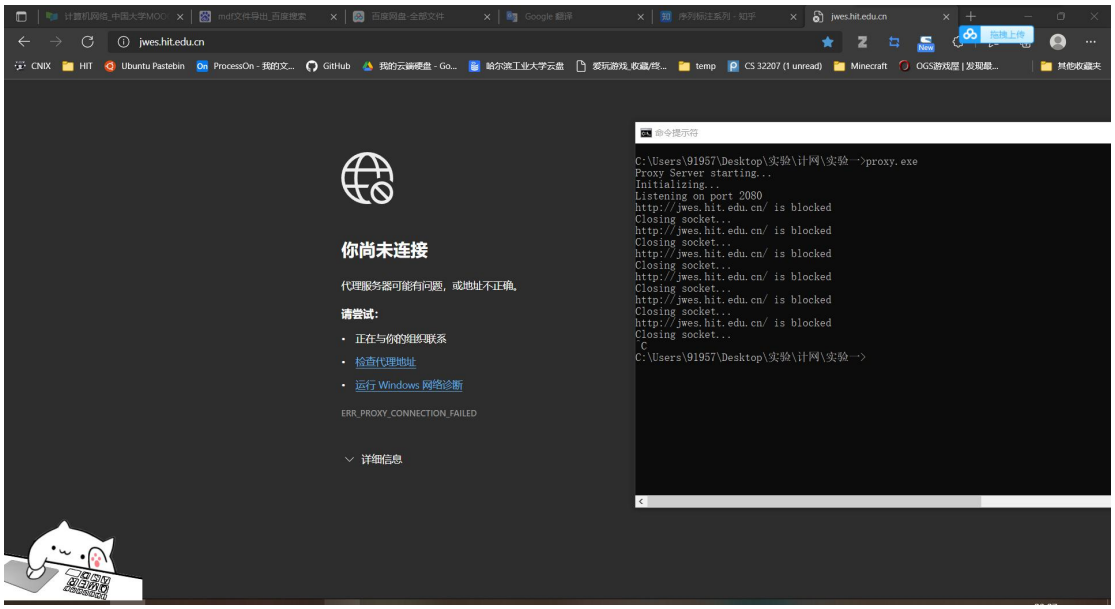
```
/*************************************************
 * Funtion name :   MackHttp
 * Description  :   插入 If-Modified-Since 字段
 * Parameter    :   char *Buffer, char *Date
*************************************************/
void MakeHttp(char *Buffer, char *Date)
/*************************************************
 * Funtion name :   MackCache
 * Description  :   缓存文件对应报文
 * Parameter    :   char *Buffer, char *url, char *Date
*************************************************/
void MakeCache(char *Buffer, char *url, char *Date)
/*************************************************************
 * Funtion name :   GetCache
 * Description  :   读取缓存文件
 * Parameter    :   char *Buffer, char *url, char *Date
 * Return       :   若缓存文件存在则返回 true，否则返回 false
*************************************************************/
bool GetCache(char *Buffer, char *url, char *Date)
```

实验结果：

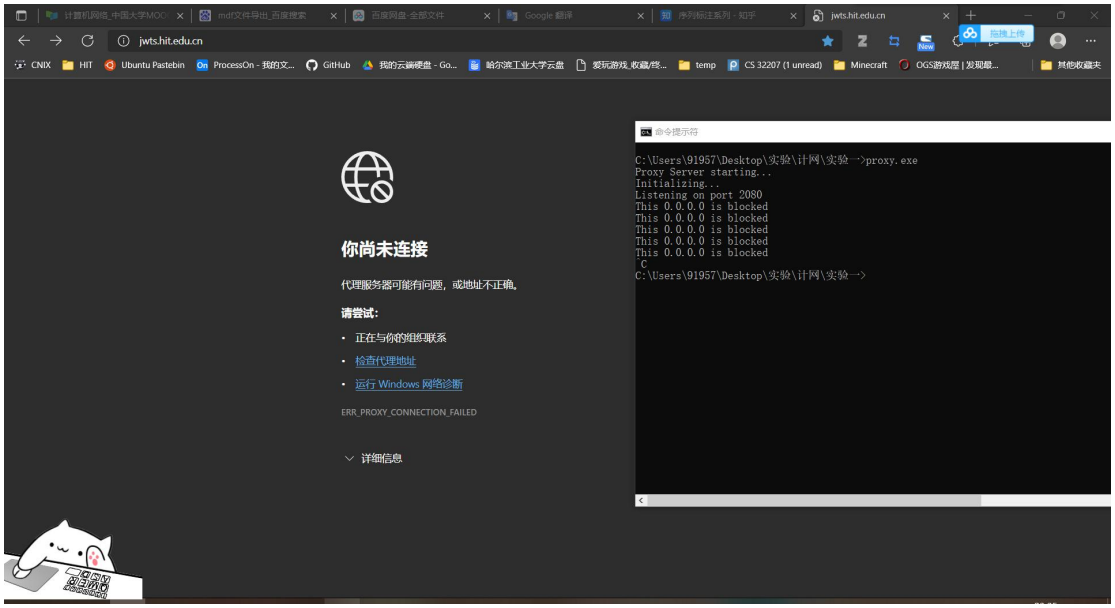
基本的代理功能



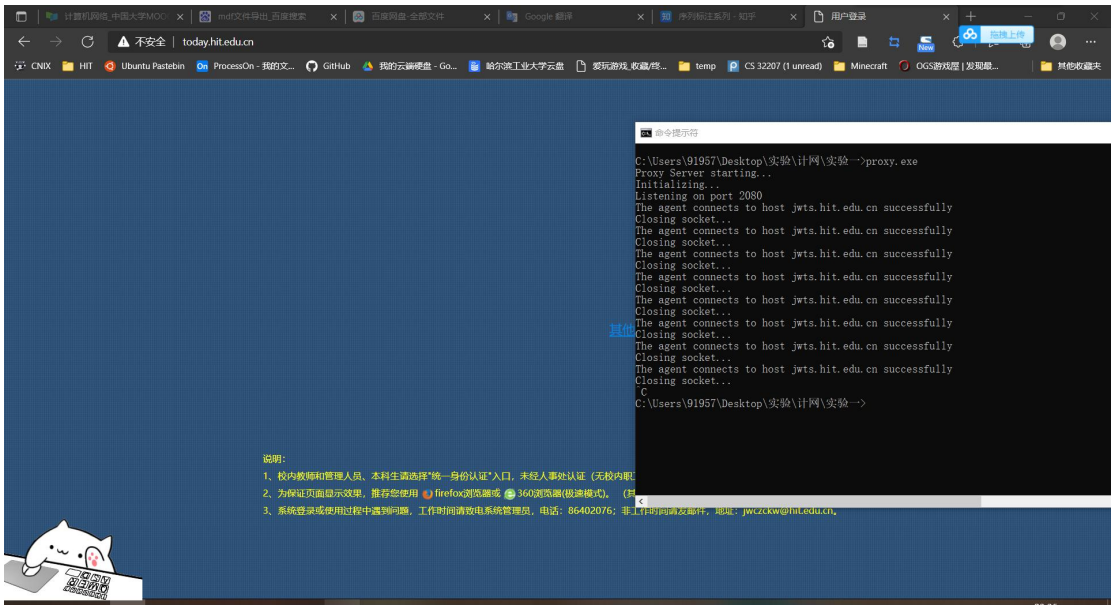
网站过滤


jwes.hit.edu.cn
你尚未连接
代理服务器可能有问题，或地址不正确。
请尝试：
正在与你的组织联系
检查代理地址
运行 Windows 网络诊断
ERR_PROXY_CONNECTION_FAILED
详细信息
命令提示符
C:\Users\91957\Desktop\实验\计网\实验一>proxy.exe
Proxy Server starting...
Initializing...
Listening on port 2080
http://jwes.hit.edu.cn/ is blocked
Closing socket...
http://jwes.hit.edu.cn/ is blocked
Closing socket...
http://jwes.hit.edu.cn/ is blocked
Closing socket...
http://jwes.hit.edu.cn/ is blocked
Closing socket...
http://jwes.hit.edu.cn/ is blocked
Closing socket...
^C
C:\Users\91957\Desktop\实验\计网\实验一>

用户过滤

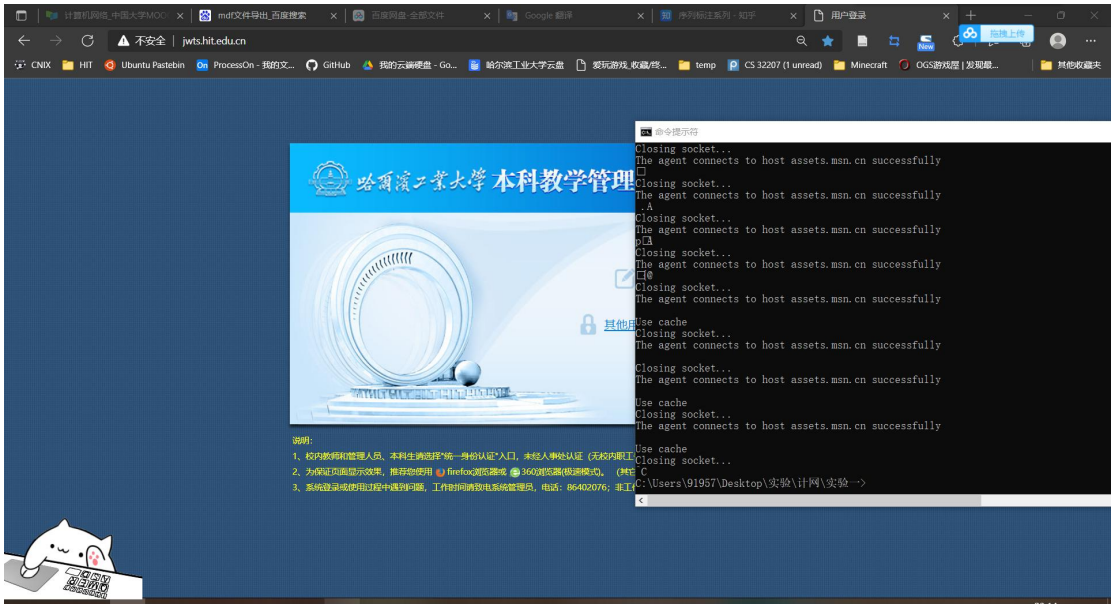

jwts.hit.edu.cn
你尚未连接
代理服务器可能有问题，或地址不正确。
请尝试：
正在与你的组织联系
检查代理地址
运行 Windows 网络诊断
ERR_PROXY_CONNECTION_FAILED
详细信息
命令提示符
C:\Users\91957\Desktop\实验\计网\实验一>proxy.exe
Proxy Server starting...
Initializing...
Listening on port 2080
This 0.0.0.0 is blocked
This 0.0.0.0 is blocked
This 0.0.0.0 is blocked
This 0.0.0.0 is blocked
This 0.0.0.0 is blocked
^C
C:\Users\91957\Desktop\实验\计网\实验一>

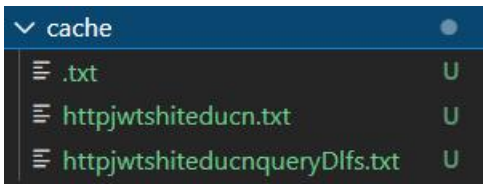
网站引导


today.hit.edu.cn
C:\Users\91957\Desktop\实验\计网\实验一>proxy.exe
Proxy Server starting...
Initializing...
Listening on port 2080
The agent connects to host jwts.hit.edu.cn successfully
Closing socket...
C:\Users\91957\Desktop\实验\计网\实验一>

cache


cache
~$90201004-王昊峰-实验一.docx U


jwts.hit.edu.cn
哈爾濱工業大學 本科教学管理
Use cache
C:\Users\91957\Desktop\实验\计网\实验一>


cache
.txt U
httpjwtshiteducn.txt U
httpjwtshiteducnqueryDlfs.txt U

问题讨论：

客户端socket编程主要步骤

1. 确定服务器IP地址和端口号
2. 创建套接字
3. 分配本地端点地址
4. 连接服务器
5. 遵循应用层协议进行通信
6. 关闭、释放套接字

服务器socket编程基本流程

主线程1：创建（主）套接字，并绑定端口号

主线程2：设置（主）套接字为被动监听模式，准备用于服务器

主线程3：反复调用accept()函数接受收下一个连接请求（通过主套接字），并创建一个新的子线程处理该客户响应

子线程1：接收一个客户的服务请求（通过创建新套接字）

子线程2：遵循应用层协议与特定客户进行交互

子线程3：关闭、释放连接并退出（线程终止）

代理服务器的基本原理

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<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。

心得体会：

为了编写能够处理多个请求的服务器，学会了如何编写多线程程序。

通过阅读参考代码以及自己尝试编写，理解了如何使用windows提供的编程接口实现一个代理服务器，将课上一个服务器需要完成的功能以及程序流程图转化为了可以运行的程序。

了解了代理服务器的基本原理。