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	王丙昊		院系	计算机科学与技术	
班级	1603108		学号	1160300302	
任课教师	聂兰顺		指导教师	聂兰顺	
实验地点	格物 207		实验时间	2018-10-31	
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：

（注：实验报告模板中的各项内容仅供参考，可依照实际实验情况进行修改。）

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：

概述本次实验的主要内容，包含的实验项等。

1．实现一个基本的代理服务器，在制定端口接收来自客户的ＨＴＴＰ请求并根据其中的ｕｒｌ地址访问服务器，接收HTTP服务器的响应报文并将其转发给客户进行浏览。

2．实现一个支持Ｃａｃｈｅ的代理服务器。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首部行），向原服务器确认是否为最新版本。

3．扩展功能，包括网站过滤、用户过滤和网络钓鱼

实验过程：

一、HTTP代理服务器实现的基本步骤

1．指定代理服务器的IP地址（使用回环地址１２７．０．０．１）以及端口号，在该端口创建一个套接字并初始化，等待客户的HTTP请求。

2．每收到客户发来的HTTP请求，创建一个新的线程处理客户的请求。

3．接收客户端的请求报文，首先判断该报文是否为空，如果为空，直接关闭与客户端的连接，并结束线程。否则，构造代理服务器的请求报文，并创建与原服务器进行通信的套接字与之进行通信。构造请求报文的过程中，可以通过修改请求报文，来实现网络过滤、用户过滤和网络钓鱼等功能；并可以根据是否已经缓存该ｕｒｌ的响应报文，在首部行中加入ｉｆ－ｍｏｄｉｆｉｅｄ－ｓｉｎｃｅ字段

4．对于请求报文加入ｉｆ－ｍｏｄｉｆｉｅｄ－ｓｉｎｃｅ字段的情况，需要根据响应报文中的Last－modified字段与缓存文件中Last－modified相比较的情况，来决定是否要将新的报文发送给客户端（并缓存），还是将缓存的报文发送给客户端；对于请求报文中没有加入ｉｆ－ｍｏｄｉｆｉｅｄ－ｓｉｎｃｅ字段的情况，接收响应报文，直接缓存并发送给客户端即可

二、实现HTTP代理服务器的关键技术及解决方案

1．创建HTTP代理服务器套接字

```
def init_socket(host, port):
    s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    s.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
    s.bind((host, port))
    s.listen(5)
    return s
```

2．网络过滤

通过判断ｕｒｌ．ｈｏｓｔｎａｍｅ是否在禁止访问的域名集合中

```
Network_Filtering = [
    # 'www.qq.com',
    'today.hit.edu.cn'
]
```

```
# 实现网络过滤
if hostname in Network_Filtering:
    source_socket.send('403 Forbidden\r\n')
    source_socket.close()
    return
```

3．用户过滤

通过判断客户端的ＩＰ地址是否在禁止用户的集合中。

```
User_Filtering = [
    # '127.0.0.1'
]
```

```
# 实现用户过滤
if addr[0] in User_Filtering:
    source_socket.send('403 Forbidden\r\n')
    source_socket.close()
    return
```

4．网络钓鱼

使用一个字典，如果收到的ｕｒｌ．ｈｏｓｔｎａｍｅ在该字典的ｋｅｙｓ集合中，则将ｈｏｓｔｎａｍｅ替换成它所对应的ｖａｌｕｅ值。

```
SITE_GUIDE = {
    'hitgs.hit.edu.cn': 'jwts.hit.edu.cn'
}
```

```
# 实现网站引导（钓鱼）
if hostname in SITE_GUIDE.keys():
    send_message = send_message.replace(url.hostname, SITE_GUIDE[url.hostname])
    hostname = SITE_GUIDE[url.hostname]
```

5．Ｃａｃｈｅ的实现

在本地维持维护一个缓存文件夹，每当从原服务器收到响应报文，如果没有缓存或者缓存的数据已经被修改过，则将新的响应报文缓存，缓存文件采用路径的md5码命名。如果HTTP代理服务器发送请求报文的过程中，发现本地有缓存，需要在原请求报文中加入ｉｆ－ｍｏｄｉｆｉｅｄ－ｓｉｎｃｅ字段，报文中的时间要采用ＧＭＴ格式

```
new_message = request_line + '\n'
t = (time.strptime(time.ctime(os.path.getmtime(filename)), "%a %b %d %H:%M:%S %Y"))
# print time.strftime('%a, %d %b %Y %H:%M:%S GMT', t)
new_message += 'If-Modified-Since: ' + time.strftime('%a, %d %b %Y %H:%M:%S GMT', t) + '\n'
for line in request_message.split('\n')[1:]:
    new_message += line + '\n'
dest_socket.send(new_message)
```

报文中时间与本次发送时间进行比较时，需要将两个ＧＭＴ格式的时间转化为ｄａｔｅｔｉｍｅ格式的时间，以便于进行比较

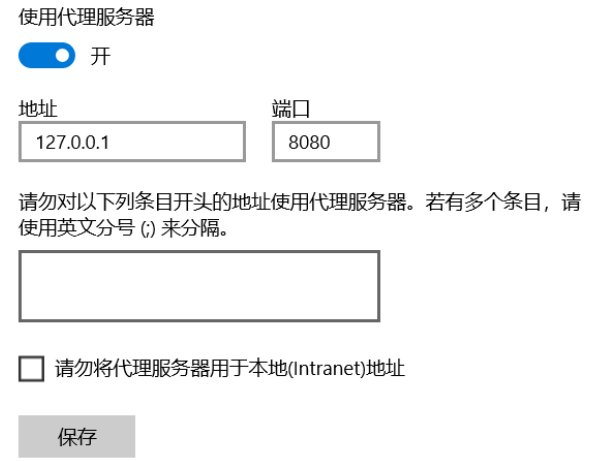
```
now_time = datetime.datetime.strptime(time.strftime('%a, %d %b %Y %H:%M:%S GMT', t), GMT_FORMAT)
last_time = datetime.datetime.strptime(c, GMT_FORMAT)
# print "now time: ", now_time
# print "last modified time: ", last_time
if now_time > last_time:    # 说明没有被修改过
    print "--------------this page has not been modified-----------------"
    source_socket.send(open(filename, 'rb').read())
```

实验结果：

（1）基本功能的测试：

HTTP代理服务器在１２７．０．０．１：８０８０等待客户的HTTP请求，验证前要配置浏览器的代理服务器。

```
proxy_socket = init_socket('127.0.0.1', 8080)
proxy_run(proxy_socket)
```



访问　ｊｗｔｓ．ｈｉｔ．ｅｄｕ．ｃｎ，可以访问

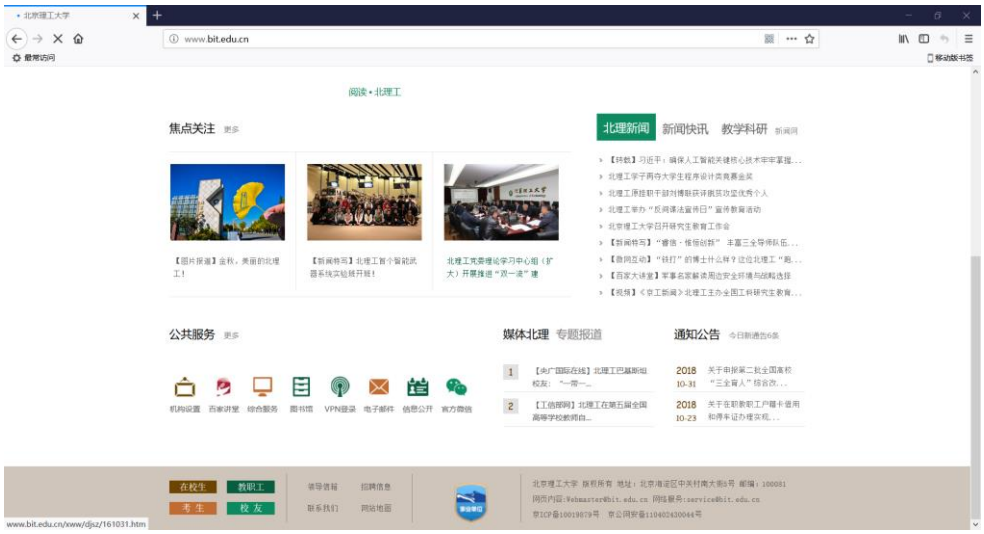
下面为访问ｊｗｔｓ时的请求报文

```
GET http://jwts.hit.edu.cn/ HTTP/1.1
Host: jwts.hit.edu.cn
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:63.0) Gecko/20100101 Firefox/63.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Connection: keep-alive
Cookie: _ga=GA1.3.2021268132.1540800418; name=value; JSESSIONID=PVR5bZYH829ccTcghT0kGsQ1rd1yggd4RWm8dMjc6LjJmk9vchj6!-1891391065; clwz_blc_pst=201330860.24859
Upgrade-Insecure-Requests: 1


POST http://jwts.hit.edu.cn/queryDlfs HTTP/1.1
Host: jwts.hit.edu.cn
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:63.0) Gecko/20100101 Firefox/63.0
Accept: */*
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Referer: http://jwts.hit.edu.cn/
X-Requested-With: XMLHttpRequest
Connection: keep-alive
Cookie: _ga=GA1.3.2021268132.1540800418; name=value; JSESSIONID=PVR5bZYH829ccTcghT0kGsQ1rd1yggd4RWm8dMjc6LjJmk9vchj6!-1891391065; clwz_blc_pst=201330860.24859
Content-Length: 0
```

（2）Cache的测试：

这里使用BIT的官网进行测试(因为jwts.hit.edu.cn的返回报文中没有Last-Modified字段)，首先第一次访问www.bit.edu.cn，可以访问



报文如下

```
GET http://www.bit.edu.cn/images/2013zzgb/icon_05.jpg HTTP/1.1
Host: www.bit.edu.cn
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:63.0) Gecko/20100101 Firefox/63.0
Accept: */*
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Referer: http://www.bit.edu.cn/css/style_2013zzgb.css
Connection: keep-alive
Cookie: __jsluid=6a35e435f7cd27c162184830bcf702f7
If-Modified-Since: Wed, 24 Jun 2015 08:45:38 GMT
If-None-Match: "119b4a-56f-5193f87fccde1"


GET http://www.bit.edu.cn/images/content/2018-10/20181010093147680669.jpg HTTP/1.1
If-Modified-Since: Thu, 01 Nov 2018 21:47:01 GMT
Host: www.bit.edu.cn
```

可以在本地Cache文件夹里发现缓存文件：

名称	修改日期	类型	大小
0a677d78c3b349c9108bbafb998499...	2018/11/1 21:48	CACHED 文件	1 KB
01d78bcfb4227407f0e2be250b999e...	2018/11/1 21:49	CACHED 文件	1 KB
1b527af6e00f79b1d987b81dfd5c57...	2018/11/1 21:49	CACHED 文件	1 KB
2ff695ac3f97adfd976b74c94636825...	2018/11/1 21:47	CACHED 文件	1 KB
3c181ae4a15f34cce3fb7358e3e9f57...	2018/11/1 21:49	CACHED 文件	0 KB
3d2e99e73d93d6f18b69926972f37c...	2018/11/1 21:49	CACHED 文件	1 KB
5baf3f858c0928cdbc5dd3657fdc4c2...	2018/11/1 21:48	CACHED 文件	1 KB
5d250073dfa60086ab1c0725b16060...	2018/11/1 21:48	CACHED 文件	2 KB
5d848459e7ccaa4a8e6bd0ca5409de...	2018/11/1 21:48	CACHED 文件	0 KB
5de8300a7e17f39cf8be1469c531569...	2018/11/1 21:49	CACHED 文件	1 KB
8eab60cf6cc391374688ffa5b9f36d66...	2018/11/1 21:49	CACHED 文件	1 KB
9b6383e1031fc84798226d80757386...	2018/11/1 21:49	CACHED 文件	2 KB
9d2ad954f549832bcf454a550b67c5f...	2018/11/1 21:48	CACHED 文件	2 KB
9f7a4017d33e1ec19b243e172bc54a...	2018/11/1 21:48	CACHED 文件	1 KB
20a07b1cf8dda5df3618fa1e6e689a7...	2018/11/1 21:49	CACHED 文件	1 KB
73c4e43404d545a7aca7070ea1f939...	2018/11/1 21:48	CACHED 文件	1 KB
75de60ecb71078a9de3bc8086b3aa0...	2018/11/1 21:49	CACHED 文件	1 KB
79c30234bb73e2d2a26c9457c92002...	2018/11/1 21:49	CACHED 文件	1 KB
95d8eb5bc554c5900b92bd8ce3b7fc...	2018/11/1 21:49	CACHED 文件	2 KB
96a8f7d1a9649efa08e36b1955f3052...	2018/11/1 21:48	CACHED 文件	1 KB
238d59a4ac08710df7766cabc54621...	2018/11/1 21:49	CACHED 文件	1 KB
581efda47d9a4ac03498b75fec4a13b...	2018/11/1 21:49	CACHED 文件	1 KB

可以查看缓存文件的内容：

31061dd73e2db09d51a315fda41bf74f.cached | 5d250073dfa60086ab1c0725b1606063.cached

```
HTTP/1.1 200 OK
Expires: Thu, 01 Nov 2018 21:46:47 GMT
Date: Thu, 01 Nov 2018 13:46:47 GMT
Server: nginx
Content-Type: image/gif
Last-Modified: Wed, 06 Jul 2016 03:06:11 GMT
Transfer-Encoding: chunked
Cache-Control: max-age=28800
X-Host: p2
Content-Encoding: gzip
Age: 1
X-Via: 1.1 PSzjjxdx9vu66:3 (Cdn Cache Server V2.0), 1.1 PSjszjdxxz2tk63:8 (Cdn Cache Serve
Connection: keep-alive

320
US?NUL NUL NUL NUL NUL NUL EPX s鳝嘲Ldd`dh``?  EPX EPX取??ENQ榜扯 mc_Q恼滇Z  敛灩g  ~鱱%告DC4[
琚雛茉扯厞軠鼕?□
[ツ厩  + ESC$  ?□VR扳ES扪NUL匍涕敲`猥膾l`f燻f hlbnn  `d`h嬷`  h  hnelde`?ENQJ@蹙R椰倥軯vS
`SI[e鉱?n棚籍鄭?N晋@氧  ES榲@Lx?r嗌>SI槎嫈墐?靠▯~  沱矾_>▯  猃环o^縵  RS?z  粀n吆y
0
```

当再一次访问www.bit.edu.cn时，同样可以访问，查看报文

```
GET http://www.bit.edu.cn/images/content/2018-10/20181010093147680669.jpg HTTP/1.1
If-Modified-Since: Thu, 01 Nov 2018 21:49:43 GMT
Host: www.bit.edu.cn
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:63.0) Gecko/20100101 Firefox/63.0
Accept: */*
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Referer: http://www.bit.edu.cn/
Connection: keep-alive
Cookie: __jsluid=6a35e435f7cd27c162184830bcf702f7



----------this page has not been modified----------
----------this page has not been modified----------
```

可以看到此时提示，该页面没有被更新过

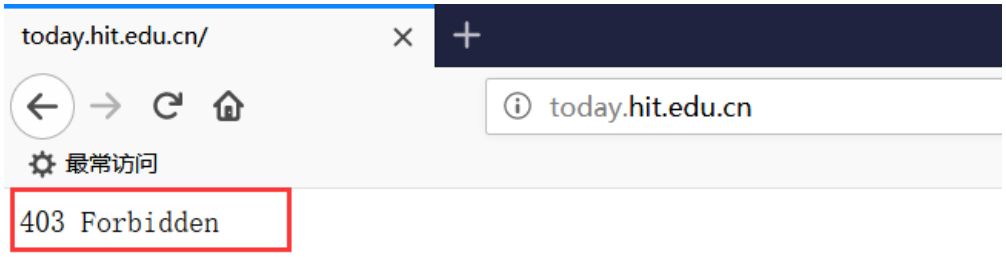
31061dd73e2db09d51a315fda41bf74f.cached 5d250073dfa60086ab1c0

```
1  HTTP/1.1 304 Not Modified
2  Date: Thu, 01 Nov 2018 13:46:54 GMT
3  Connection: keep-alive
4  Cache-Control: max-age=10800
5  ETag: "1bff06-3404f-5790330a572da"
6  Last-Modified: Thu, 25 Oct 2018 01:04:51 GMT
7  X-Via-JSL: 9587073,-
8  X-Cache: hit
9
```

（3）网站过滤功能的测试：

```
Network_Filtering = [
    # 'www.qq.com',
    'today.hit.edu.cn'
]
```

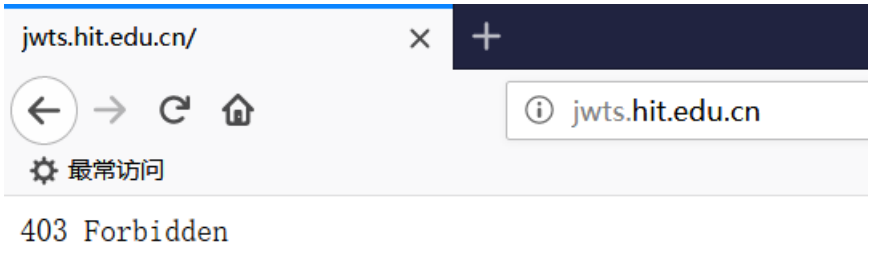
设today.hit.edu.cn为禁止访问网站，此时HTTP代理服务器会返回给客户端403 Forbidden



（4）用户过滤功能的测试：

```
User_Filtering = [
    '127.0.0.1'
]
```

测试时，将回环测试IP 127.0.0.1设置为禁止访问IP，则访问jwts.hit.edu.cn



此时会禁止访问

（5）网络钓鱼功能的测试：

测试时，设置为，如果访问hitgs.hit.edu.cn，则会跳转到jwts.hit.edu.cn，结果如下：

```
SITE_GUIDE = {
    'hitgs.hit.edu.cn': 'jwts.hit.edu.cn'
}
```

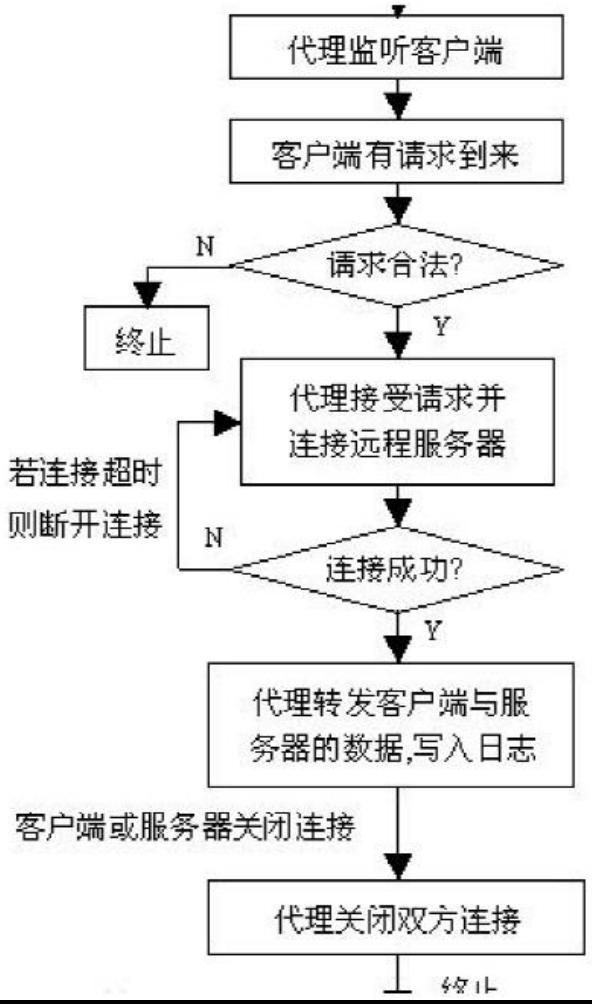
可以发现，输入的是hitgs.edu.cn，但是却被引导至了jwts.hit.edu.cn

问题讨论：

一、HTTP代理服务器的基本原理

HTTP代理服务器是能够代表初始Ｗｅｂ服务器来满足HTTP请求的网络实体，它有自己的磁盘存储空间。可以通过浏览器设置，将客户所有的ＨＴＴＰ请求首先指向ＨＴＴＰ代理服务器。HTTP代理服务器是服务器同时又是客户，当它接收浏览器的请求并发回响应时，它是一个服务器。当它向初始服务器发出请求并接收响应时，它是一个客户。

二、HTTP代理服务器的程序流程图



三、实现HTTP代理服务器的关键技术及解决方案

见**实验过程**中的网络过滤、用户过滤、Cache以及网络钓鱼等技术的实现

四、源代码

创建HTTP代理服务器的Socket

```
def init_socket(host, port):
    s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    s.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
    s.bind((host, port))
    s.listen(5)
    return s
```

网络过滤、用户过滤以及网络钓鱼的实现：

```
# 实现网络过滤
if hostname in Network_Filtering:
    source_socket.send('403 Forbidden\r\n')
    source_socket.close()
    return

# 实现用户过滤
if addr[0] in User_Filtering:
    source_socket.send('403 Forbidden\r\n')
    source_socket.close()
    return

# 实现网站引导（钓鱼）
if hostname in SITE_GUIDE.keys():
    send_message = send_message.replace(url.hostname, SITE_GUIDE[url.hostname])
    hostname = SITE_GUIDE[url.hostname]
```

Cache的实现（主要部分）：

```
# 在目录下创建一个文件夹，并将所有的缓存文件存在该目录下
filepath = os.path.join(os.path.dirname(__file__), 'cache')
if not os.path.exists(filepath):
    os.mkdir(filepath)

filename = os.path.join(filepath, m.hexdigest() + '.cached')
if os.path.exists(filename):
    # 如果有缓存，则需要If-Modified-Since首部字段来确定是否为最新的
    new_message = request_line + '\n'
    t = (time.strptime(time.ctime(os.path.getmtime(filename)), "%a %b %d %H:%M:%S %Y"))
    new_message += 'If-Modified-Since: ' + time.strftime('%a, %d %b %Y %H:%M:%S GMT', t) + '\n'
    for line in request_message.split('\n')[1:]:
        new_message += line + '\n'
    dest_socket.send(new_message)
    print new_message
```

```
while True:
    data = dest_socket.recv(max_len)
    fp = open(filename, 'wb')
    if count == 0:
        loc1 = data.find("Last-Modified")
        if loc1 >= 0:
            b = data[loc1:]
            loc2 = b.find('\r\n')
            c = b[15:loc2]          # 为GMT格式
            now_time = datetime.datetime.strptime(time.strftime('%a, %d %b %Y %H:%M:%S GMT', t), GMT_FORMAT)
            last_time = datetime.datetime.strptime(c, GMT_FORMAT)
            if now_time > last_time:    # 说明没有被修改过
                print "--------------this page has not been modified------------------"
                source_socket.send(open(filename, 'rb').read())
                break
            else:                       # 如果被修改过，则将新的报文返回给客户端
                if len(data) > 0:
                    source_socket.send(data)
                    fp.write(data)
                else:
                    break
```

```
else:
    # 如果没有缓存，则直接请求对象
    print send_message
    dest_socket.send(send_message)
    # 等待响应报文，接收后需缓存
    fp = open(filename, 'ab')
    while True:
        data = dest_socket.recv(max_len)
        if len(data) > 0:
            fp.write(data)
            source_socket.send(data)
        else:
            break
```

心得体会：

1.更加熟悉并掌握了HTTP代理服务器的基本原理

2.对Python的Socket编程进行了实践，有了更深的理解

3.掌握了HTTP代理服务器设计与实现的基本技能