

**//////发现交错文件了QAQ**

**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 | 陈佳琦 | | 院系 | 计算机科学与技术学院 | | |
| 班级 | 1837101 | | 学号 | 1183710124 | | |
| 任课教师 | 李全龙 | | 指导教师 | 李全龙 | | |
| 实验地点 | 格物213 | | 实验时间 | 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 | | | | | | |
|  | | | | | | |

**计算学部**

|  |
| --- |
| 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： |
| 概述本次实验的主要内容，包含的实验项等。   1. 设计基本代理服务器。   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。在此次实验中我使用的端口是8080。   1. 为代理服务器添加cache功能。   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 -modif ied-since 头行），向原服务器确认缓存对象是否是最新版本。当然也可以通过一个http请求来判断网页是否有所更新。结果是一样的。   1. 扩展 HTTP 代理服务器，支持如下功能： 2. 网站过滤   允许/不允许访问某些网站；   1. 用户过滤   支持/不支持某些用户访问外部网站；   1. 网站引导   将用户对某个网站的访问引导至一个模拟网站，也就是通常所说的钓鱼。 |
| 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.基础功能：  1.1语言选择  套接字编程在不同语言下，表现在使用略有不同，比如在C++中关于套接字的构造函数，仅仅是返回一个数据结构，要想建立连接，必须首先初始化环境，然后创建套接字，并配置相关协议以及ip端口号等信息，协议主要还是IP协议，TCP协议，然后如果是server的话，还需要调用listen函数进行监听，然后是accept函数，来进行建立连接；另外针对于客户端虽然简单了一点，但是过程也比较繁琐，最后要使用connect函数来建立连接。在进行一般工作的时候，对底层的配置没有太多的变化，特别是针对像计网这样的第一次实验，这些初始化的过程可以认为都是一样的，所以使用C++进行编程也只是凭空增加自己的工作负担而已，经过认真考虑，决定使用应用较为广泛的java语言实现。将繁琐的过程省略之后，也能更好的体会这个过程，从而实现这个实验更好的效果。  1.2过程逻辑      1.3过程详解  1.3.1服务器端与客户端使用  使用java.net.Socket;和java.net.ServerSocket;进行套接字的创建，以及连接的建立和服务器监听，这两个类，支持底层操作，也提供默认的封装操作。具体使用如下，分为服务器端和客户端：   |  | | --- | |  |   1.3.2处理客户请求  在处理客户端请求时，为了实现多用户的代理服务器，应该新开一个线程进行处理，也就是每一个线程处理一个客户的一个请求：   |  |  | | --- | --- | | 建立连接 | Socket socket = proxyServerSocket.accept();  (new processClient(socket)).start(); | | 处理请求 | public void run(){  byte[] bytes = new byte[4096];//将每次读取的数据存储到bytes中  try {  inputStream = clientSocket.g etInputStream();//获取请求输入流  outputStream= clientSocket.g etOutputStream();//获取响应输出流  inputStream.read(bytes);//读取请求  String messag e = new String (bytes);  String patternString = "Host: [0-9a-zA-Z.]+";  Pattern pattern = Pattern.compile(patternString );  Matcher matcher=pattern.matcher(messag e);//获取访问网站名，提取Host关键字  if (matcher.find()) {  String host = matcher.g roup().replace("Host: ", "");  if (haveThisF ile(host) == f alse) {  Socket serverSocket = new Socket(host, 80);  //与服务器建立连接  serverOutputStream = serverSocket.g etOutputStream();  serverOutputStream.write(bytes);//将请求传给目标服务器 |   1.3.3信息传输  在进行客户请求处理，转发请求，接受响应，转发响应的过程中，经常会出现超时的问题，抛出connect timed out异常。首先可以肯定是中间处理时间太长了，导致访问超时。可以通过两个线程并行执行解决，也就是一个线程读取数据的同时，另一个数据进行数据的输出：   |  | | --- | | class thread2 extends Thread {  private InputStream input;//定义输入流  private OutputStream output;//定义输出流  public thread2(InputStream input, OutputStream output) {//构造函数  this.input = input;  this.output = output;  }  public void run() {  try {//每次读一个byte传递  while (true) {  int next = input.read();  if (next == -1) break;  output.write(next);  }  } catch (IOException e) {//异常处理  System.out.println("Maybe something wrong happened!");  }  }  } |   2.附加功能：  2.1添加cache功能  2.1.1cache信息存储：  添加cache功能，必须要做的就是将之前从服务器传过来的网页文件进行保存：使用本地缓存存储，即设定一个ArrayList将访问文件作为string 存储，在使用的时候再调用getByte函数进行转换。仅使用一个HashMap就可以实现cache的功能：   |  | | --- | | if (matcher.f ind()) {  String host = matcher.g roup().replace("Host: ", "");  if (haveThisF ile(host) == f alse) {  Socket serverSocket = new Socket(host, 80);  serverOutputStream = serverSocket.g etOutputStream();  serverOutputStream.write(bytes);  (new thread2(inputStream,serverOutputStream)).start();  serverInputStream = serverSocket.g etInputStream();  //开一个新的list存储网页文件信息  ArrayList<Integ er> tmp = new ArrayList<Integ er>();  while(true) {  int next = serverInputStream.read();  if (next == -1) break;  outputStream.write(next);  tmp.add(new Integ er(next));  }  proxy.cache.put(host, tmp);//将网页信息与网站名对应，存储映射表中  //创建文件，将信息存储在本地文件中  F ile f ile = new F ile(host + ".txt");  f ile.createNewF ile();  PrintStream printor = new PrintStream(f ile);  printor.write(tmp.toString ().g etBytes());  printor.close();  } |   2.1.2更新  构建一个http请求报文，向服务器请求上一次更新时间，并和本地文件的last-modif ied时间进行一个对比。如果在上次请求之后更新了文件，那就重新发送请求，按照之前的过程再走一遍，如果没有更新，就将本地文件发送给客户即可：   |  | | --- | | private long getUpdateTime(String host) throws IOException {  URL u = new URL("http://" + host);  HttpURLConnection http = (HttpURLConnection) u.openConnection();  http.setReq uestMethod("HEAD");  Date lastModif y =new Date(http.g etLastModif ied());  return lastModif y.g etTime();  } |   2.2过滤与引导  2.2.1网站过滤：  提取了主机名之后，判断一下是不是要过滤的目标网站，如果是就转发请求，否则不转发。  2.2.2用户过滤：  从请求报文中提取用户名，判断一下是不是要过滤的目标用户，如果是就转发请求，否则不转发。  2.2.3钓鱼：  当检测到用户访问特定的源网站之后，向相应的目标网站发送请求，并将其相应转发给客户即可。 |
| 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。   1. 基本代理服务器的实现   通过查找使用http协议的网站，发现有如下网站满足条件，并将其定为此次实验的目标网站，如：jwts.hit.edu.cn     1. Cache   在进行了一次访问之后，第二次访问时，就输出了像在表格最后的信息，已经有这个文件了，就直接将这个文件传输为客户就行了，并且从时间上判断，网页还没有更新:   |  | | --- | | - - - - - - - - - - - - - - - - www.4399.com  ......  - - - - - - - - - - - - - - - - img a2.5054399.com  ......  - - - - - - - - - - - - - - - - g .live.com  ......  - - - - - - - - - - - - - - - - www.4399.com  have this f ile!!!!!!!!!!!!  www.4399.com.txt  lastModified time  1964505342608  upload f rom localhost |  1. 过滤与引导：   3.1网站过滤  访问被过滤的网站，一直无响应    3.2用户过滤  过滤掉127.0.0.2  将自己的host设置为127.0.0.2，过滤掉自己，同样页面也无响应    3.3钓鱼  源地址为4399，目的地址为一张图片 |
| 问题讨论： |
| 对实验过程中的思考问题进行讨论或回答。  在实现细节上的几个小问题：  （1）传输图片资源的时候会花掉，但是如果把buffer大小改成1个字节，就不会花了  （2）在程序执行的过程中，通常连接已经建立了，但是总是不能完整地打开网页，同时会抛出异常write time out。为了解决这样的问题，尝试将数据读取到byte类型的变量中，然后通过输出流的write函数将其输出   |  | | --- | | inputStream.read(bytes);  String messag e = new String (bytes);  serverOutputStream.write(bytes);  但在这个过程中不能使用string 类型的变量进行传递，否则会出现静态类型错误。 | |
| 心得体会： |
| 结合实验过程和结果给出实验的体会和收获。  实验时间紧张，加上前期划水，以及个人水平所限，所以完成的很匆忙。  但总的来说也复习运用了课上学习的内容，作为简单的练习很有意义。 |