|  |
| --- |
| 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。   1. 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： |
| 使用python3.12.0编写  一．Socket编程的客户端和服务器端主要步骤  客户端：  1.根据目的服务器IP地址和端口号创建套接字，连接服务器  2.发送请求报文  3.接受返回报文  4.关闭连接  服务器端：  1.创建套接字，绑定IP地址和端口号，监听端口  2.从连接队列中取出一个连接请求，三次握手创建连接  3.接收请求报文  4.发送响应报文  5.关闭当前连接，继续监听端口  二．HTTP代理服务器的基本原理  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。  三．HTTP代理服务器的程序流程图    四．实现HTTP代理服务器  1.基本代理功能    2.Cache      3.过滤 |
| 实验结果： |
| 1.系统代理    2.基本功能实现    3.cache实现      4.过滤以及重定向 |
|  |
| 问题讨论： |
| **HTTP代理服务器的基本工作原理**  HTTP代理服务器的基本原理如图所示。代理服务器对于原客户端而言扮演的是服务器的角色，对于原服务器而言扮演的是客户端的角色。因此代理服务器的任务便是准确高效地将客户端的HTTP请求报文转发到服务器端，将服务器端的HTTP响应报文返回到客户端。    代理服务器的基本原理  这个过程中，代理服务器需要分别对客户端和服务器端建立SOCKET连接。通过接受客户端的SOCKET信息获取目标服务器的IP和PORT，从而对服务器建立SOCKET连接。  缓存的基本原理为在请求报文中解析URL，将处理过的URL（处理是为了解决URL中/与路径冲突以及文件最长名称比URL最长名称短等问题）当作文件名，在该文件中存储用GET方法访问该URL时得到的资源。在每次访问时查看是否存在该文件，不存在则创建。如果存在，则读取该文件最终修改时间，添加到If-modified-since报文属性中发送给对应URL，通过查看响应是否为302来判断该缓存是否可用。如果可用则返回该文件中数据，如果不可用，则重新访问该URL，返回响应报文并且更新文件中内容。  对于目标用户IP地址的屏蔽和目标网站的屏蔽，是通过解析客户端SOCKET信息实现的。如果识别为需要屏蔽，则不像服务器转发请求报文，客户端自然也收不到来自服务器的响应报文。这样便实现了屏蔽功能。而钓鱼网站则通过HTTP协议的302重定向来实现。  **Cache功能中304响应未生效，**  检查代码发现if-modified-since头行格式错误，未按RFC标准将时间格式化为“EEE,ddMMMyyyyHH:mm:ssGMT”，修正格式后成功触发304；  **网站过滤与钓鱼功能冲突，访问被过滤网站时仍触发钓鱼跳转，**  原因是功能执行优先级未定义，调整逻辑为“先执行网站过滤，过滤不通过则直接拒绝，通过再判断是否钓鱼”，解决了冲突问题。这些问题多源于细节把控不足，需结合协议标准和逻辑优先级排查。 |
| 心得体会： |
| 这次实验让我从理论走向实践，深刻理解了HTTP代理的工作原理。起初觉得基本代理只需“转发请求”，实际编写时才发现需处理请求头解析、响应报文完整性校验等细节；调试Cache功能时，反复对比日志排查304响应问题，让我学会用协议标准验证代码逻辑。扩展功能开发中，为避免过滤与钓鱼冲突，我主动梳理功能优先级，培养了系统化思维。此外，实验也让我意识到网络安全的重要性，比如钓鱼功能虽为实验需求，但实际中需严格把控伦理边界。整体而言，这次实验不仅提升了编码与调试能力，更让我体会到“理论指导实践，实践反哺理论”的学习逻辑。 |