SE2302 2024年春季学期 实验10：Web代理 分配日期：5月10日，截止日期：5月31日，晚上11:59 负责人：Xingyu Yan (xyyan0123@sjtu.edu.cn)

引言

Web代理是一个程序，它充当Web浏览器和最终服务器之间的中间人。代理不是直接联系最终服务器来获取网页，而是浏览器先联系代理，代理再将请求转发给最终服务器。当最终服务器回复代理时，代理将回复转发给浏览器。 代理被用于许多目的。有时代理被用于防火墙中，这样代理就是防火墙内浏览器联系外部最终服务器的唯一途径。代理可能对页面进行转换，例如，使其可以在支持Web的手机上调阅。代理也被用作匿名器。通过剥离请求中的所有识别信息，代理可以使浏览器对最终服务器匿名。代理甚至可以用于缓存Web对象，通过存储一个副本，比如一个图像，当首次请求它时，然后在响应未来的请求时提供该图像，而不是去最终服务器。 在这个实验中，你将编写一个并发的Web代理，记录请求。在实验的第一部分，你将编写一个简单的顺序代理，它反复等待请求，将请求转发给最终服务器，并将结果返回给浏览器，同时在磁盘文件中记录这些请求的日志。这一部分将帮助你理解网络编程和HTTP协议的基础知识。 在实验的第二部分，你将升级你的代理，使其使用线程来同时处理多个客户端。这一部分将给你一些并发和同步的经验，这是计算机系统概念中至关重要的。

物流

像往常一样，你应该独立完成这个实验。任何关于作业的澄清和修订将发布在课程网页上。

分发说明

实验在svn的lab10目录中分发给你。分发包含几个文件：

1. **proxy.c**: 这是骨架，虽然非常简单。它包含了你的代理的大部分逻辑。
2. **csapp.c**: 这是CS:APP教科书中描述的同名文件。它包含了错误处理包装器和辅助函数，如RIO（鲁棒I/O）包（CS:APP 10.5），open clientfd（CS:APP 11.4.8），和open listenfd（CS:APP 11.4.8）。
3. **csapp.h**: 这个文件包含了一些显式常量、类型定义和csapp.c中函数的原型。
4. **Makefile**: 编译并链接proxy.c和csapp.c成为可执行文件proxy。
5. **grade.py**: 用Python编写的实验评分脚本。更多信息将在评估部分给出。
6. **client, server**: grade.py将使用的助手。不要删除或修改它们。 你的proxy.c文件可以调用csapp.c文件中的任何函数。你可以自由修改任何东西，包括Makefile和grade.py。但是，当我们评分你的实验时，我们会用标准版本替换你的grade.py，所以你不能在评分脚本中作弊。

第一部分：实现顺序Web代理

在这一部分，你将实现一个顺序记录日志的代理。你的代理应该打开一个套接字并监听连接请求。当它收到连接请求时，它应该接受连接，读取HTTP请求，并解析它以确定最终服务器的名称。然后，它应该打开一个连接到最终服务器，发送请求，接收回复，并将回复转发给浏览器，如果请求没有被阻止的话。 由于你的代理是客户端和最终服务器之间的中间人，它将同时具有两者的元素。它将作为服务器对Web浏览器，作为客户端对最终服务器。因此，你将获得客户端和服务器编程的经验。

日志记录

你的代理应该通过在屏幕上打印日志来跟踪所有请求。每个日志条目应该是这样一行：

```
日期: 浏览器IP URL 大小
```

其中浏览器IP是浏览器的IP地址，URL是所请求的URL，大小是返回对象的字节大小。例如：

```
Fri 10 May 2024 16:58:24 UTC: 127.0.0.1 http://127.0.0.1:45659/ 581
```

注意大小基本上是从打开连接到关闭连接期间从最终服务器接收到的字节数。只有最终服务器响应满足的请求才应该被记录。我们已经提供了一个格式化日志条目的函数，以创建所需的格式日志条目。

端口号码

你的代理应该在命令行上传递的端口号上监听其连接请求：

```
unix> ./proxy 15213
```

你可以使用任何端口号p，其中1024 ≤ p ≤ 65536，并且p当前不被任何其他系统或用户服务（包括其他学生的代理）使用。查看/etc/services以获取其他系统服务保留的端口号列表。

第二部分：同时处理多个请求

真正的代理不会顺序处理请求。它们同时处理多个请求。一旦你有一个工作顺序记录日志的代理，你应该修改它以同时处理多个请求。最简单的方法是为每个新到达的连接请求创建一个新线程（CSAPP 12.3.8）。 通过这种方法，多个对等线程可能会同时访问日志文件。因此，你需要使用一个信号量来同步对文件的访问，以便只有一个对等线程一次可以修改它。如果你不同步线程，日志文件可能会被损坏。例如，文件中的一行可能在另一行中间开始。

评估

我们使用grade.py来评分你的实验。运行这个Python脚本需要你安装一个名为pexpect的Python库。在大多数情况下，sudo apt-get update然后sudo apt-get install python-pexpect将为你完成它。你被允许检查或修改评分脚本。如果你发现任何错误或改进，你可以与TA讨论。

- **基本代理功能（35分）**。你的顺序代理应该正确地接受连接，将请求转发给最终服务器，并将响应传回浏览器，为每个请求制作一个日志条目。我们通过从客户端发出代理请求并通过你的代理尝试访问服务器来测试这部分。 为了获得满分，你需要正确处理各种请求，包括异常的请求。
- 处理并发请求（30分）

  。 你的代理应该能够处理多个并发连接。我们在两种情况下测试它：

  - 下载一个大文件并同时访问许多小文件。在大文件访问中，服务器将分部分发送内容，你应该立即将接收到的部分重定向到客户端，而不是缓冲它们的内容，直到所有部分都接收完毕。如果你缓冲它们，用户会认为他的下载卡住了。
  - 同时访问许多小文件。在这个测试中，小访问不再顺序进行。我们随机选择一个HTTP连接并发送几个字节来模拟分段和乱序。

提示

- **开始你的代理的最佳方式是从一个基本的echo服务器（CS:APP 11.4.9）开始，然后逐步添加功能，将服务器变成一个代理。**
- **你应该使用telnet作为客户端来调试你的代理（CS:APP 11.5.3），并将你的访问目标定位到一些仅限HTTP的网站，比如http://www.sjtu.edu.cn/。请注意，由于大多数网站已经转移到更安全的https，你可能会得到HTTP/1.1 301 Moved permanently。**
- **由于我们希望你专注于这个实验的网络编程问题，我们已经为你提供了两个额外的帮助程序：parse uri，它从URI中提取主机名、路径和端口组件，以及format log entry，它以正确的格式为日志文件构建条目。**
- **注意内存泄漏。如果HTTP请求的处理由于任何原因失败，线程必须在终止之前关闭所有打开的套接字描述符并释放所有内存资源。**

- 你会觉得为每个线程分配一个小型的唯一整数ID（例如当前请求号）并将其作为线程例程的参数之一传递非常有用。如果你在每个调试输出语句中显示此ID，则可以准确跟踪每个线程的活动。
- 为了避免潜在的致命内存泄漏，你的线程应该作为分离的运行，而不是可加入的（CS:APP 12.3.6）。
- 对于套接字的所有I/O，请使用RIO（鲁棒I/O）包（CS:APP 10.5）。不要在套接字上使用标准I/O。如果你这样做，你很快就会遇到问题。但是，对于日志文件的I/O，使用fopen和fwrite等标准I/O调用是可以的。
- csapp.c中的Rio readn、Rio readlineb和Rio writen错误检查包装器不适用于现实代理，因为它们在遇到错误时终止进程。相反，你应该编写新的包装器，称为Rio readn w、Rio readlineb w和Rio writen w，它们在I/O失败时仅返回，并在打印警告消息后继续。当任一读取包装器检测到错误时，它应该返回0，就好像它在套接字上遇到了EOF。
- 读取和写入可能因多种原因失败。最常见的读取失败是errno = ECONNRESET错误，这是由于从另一端的对等体已经关闭的连接中读取引起的，通常是超负荷的最终服务器。最常见的写入失败是errno = EPIPE错误，这是由于写入到另一端的对等体已经关闭的连接引起的。例如，当用户在长时间传输期间点击他们浏览器的停止按钮时，就可能发生这种情况。
    - 首次写入已被对等体关闭的连接会触发errno设置为EPIPE的错误。第二次写入这样的连接会触发其默认动作是终止进程的SIGPIPE信号。为了防止你的代理崩溃，你可以使用signal函数的SIG IGN参数（CS:APP 8.5.3）来明确忽略这些SIGPIPE信号。
    - Wireshark和strace有助于监控你的程序在网络上做了什么。你可以在Google或百度上搜索它们的教程。

提交说明

只需像以前实验一样将你的代码提交到svn。不要忘记你所做的任何修改。如果你的代码在TA的计算机上无法编译，你将得到零分。