

# proxy lab 实验报告

赵浩宇,2016012390, 计科 60

2018 年 6 月 24 日

## 1 实验要求

本次实验中，要求编写一个缓存 Web 对象的简单 Http 代理。

## 2 算法概述

主要分为三部分，第一个部分实现一个顺序执行的 Http 代理，第二个部分对第一部分进行拓展，实现一个多线程的 Http 代理，第三个部分实现本地缓存。

## 3 详细实现描述

### 3.1 顺序执行 Http 代理

这个部分不是非常的难，但是很。首先要实现对请求的字符串进行分析，需要实现一个 parse 函数，还需要构建 header。在 doit 函数中，感觉就是按部就班的去把每一个需要实现的功能实现出来，但是错误处理是比较麻烦的。

### 3.2 多线程 Http 代理

这个部分是最简单的，只需要加入 thread 函数，在函数内分离当前线程，使得执行完毕之后被自动回收即可。

### 3.3 本地缓存 Http 代理

这一部分感觉上是比较麻烦的，但是可能由于测例的原因，要想通过测试，并不用写得非常的完美。我采用的方法是首先将每一块 cache block 抽象成一个结构体，之后再将 cache block 的数组抽象成 cache 结构体。cache 的替换策略就采用 LRU。但是一个比较重要的问题是，当多线程的时候，如果不加 sem，那么读和写都是非常不安全的，这样可能会两个页面都写到一个 cache block 中去，也有可能多线程在同时读写，这非常不安全。我的实现方法是，对每一个 cache block 加一个锁，不管是读还是写都只能有一个线程进行操作。同时对于全局计数器 counter，也用一个信号量实现互斥。这样安全性可以得到保证，但是性能上肯定是有些慢的。

## 4 思考与讨论

通过这个实验，我更加深入地理解了 IO，网络，以及并发编程的相关内容，我的 cache 的实现感觉依然可以改的更好，比如说读和写是互斥的，但是可以允许多个进程来读。