南京信息工程大学

本科生课程设计

题　　目　　《Python 应用实践》

学生姓名　朱宸扬

学　　号　202183760012

学　　院　计算机学院

专　　业　信息安全(奇安信)

指导教师　闫雷鸣

二〇二三年六月九日

目　录

（居中、黑体、三号，“目”与“录”字中间留一个汉字空格）

（空 1 行，五号）

读者写者问题与异步爬虫

朱宸扬

南京信息工程大学计算机与软件学院，江苏 南京 210044

摘要：当多个进程并发读、写同一个文件时，怎样解决读、写冲突问题？这就是多进程同步问题中经典的“读者-写者”问题，读者和写者是互斥的，写者和写者也是互斥的，而读者和读者不存在互斥问题。为此，设计了两种调度规则，读者优先与公平调度。网络爬虫也叫网络机器人，可以代替人们自动化浏览网络中的信息，进行数据的采集与整理，异步爬虫可以大大加快爬虫的效率。

关键词：进程；线程；共享内存；异步爬虫

Reader writer problem and asynchronous crawler

Zhu Chenyang

School of Computer, NUIST, Nanjing 210044, China

Abstract： When multiple processes concurrently read and write the same file, how do I solve the read and write conflict problem? This is the classic "reader-writer" problem in the multiprocess synchronization problem, where the reader and the writer are mutually exclusive, the writer and the writer are mutually exclusive, and the reader and the reader are not mutually exclusive. Therefore, two scheduling rules are designed, reader priority and fair scheduling. Web crawler is also called network robot, which can replace people to automatically browse information in the network, collect and organize data, and asynchronous crawler can greatly speed up the efficiency of the crawler.

Key words： Process; Threads; Shared memory; Asynchronous crawler

1 读者写者问题

1.1 背景

当多个进程并发读、写同一个文件时，怎样解决读、写冲突问题？这就是多进程同步问题中经典的“读者-写者”问题，如图 1 所示。此时，允许多个读者进程同时 对文件执行读操作，只允许一个写者进程向文件中写信息，任一写者在完成写操作之前不允许其他读者或写者工作，读者和写者是互斥的，写者和写者也是互斥的，而读者和读者不存在互斥问题。

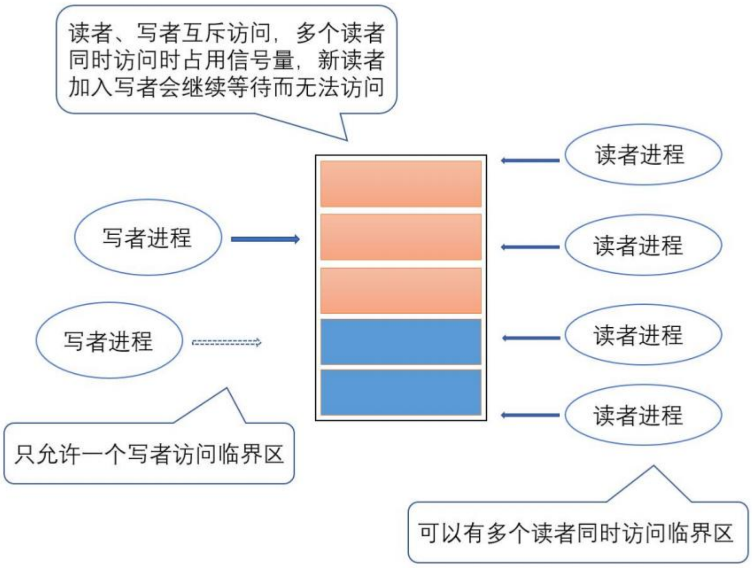

读者、写者互斥访问，多个读者同时访问时占用信号量，新读者加入写者会继续等待而无法访问
读者进程
写者进程
读者进程
写者进程
读者进程
读者进程
只允许一个写者访问临界区
可以有多个读者同时访问临界区

图 1 读者-写者问题

1.2 信号量

信号量机制是一种可以帮助解决此类问题的机制，信号量可以理解为一种锁，这种锁可以规定同时可以执行的操作数量。Semaphore，它内部维护了一个计数器，每一次 acquire 操作都会让计数器减 1，每一次 release 操作都会让计数器加 1，当计数器为 0 时，任何线程的 acquire 操作都不会成功，Semaphore 确保对资源的访问有一个上限，这样，就可以控制并发量。如果使用 Lock，RLock，那么只能有一个线程获得对资源的访问，但现实中的问题并不总是这样，假设这样一个场景，一个线程安全的操作，同一个时刻可以允许两个线程进行，如果太多了效率会降低，那么 Lock，Rlock，包括 Condition 就不适合这种场景。

1.3 读者优先

读者与写者是两种不同的行为，为了让读者优先工作，可以使读者开始工作之前，锁

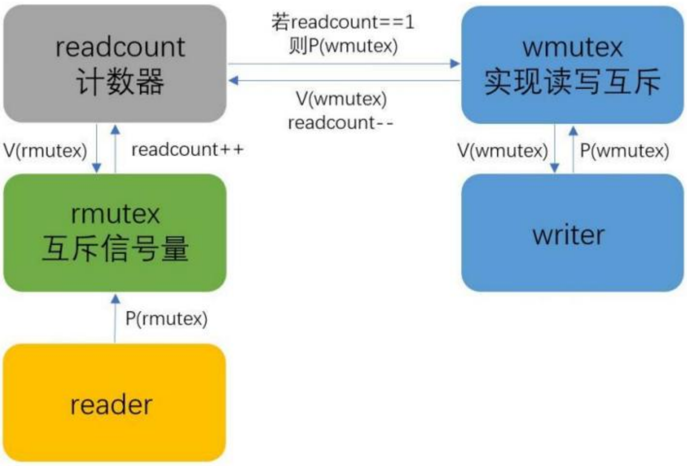
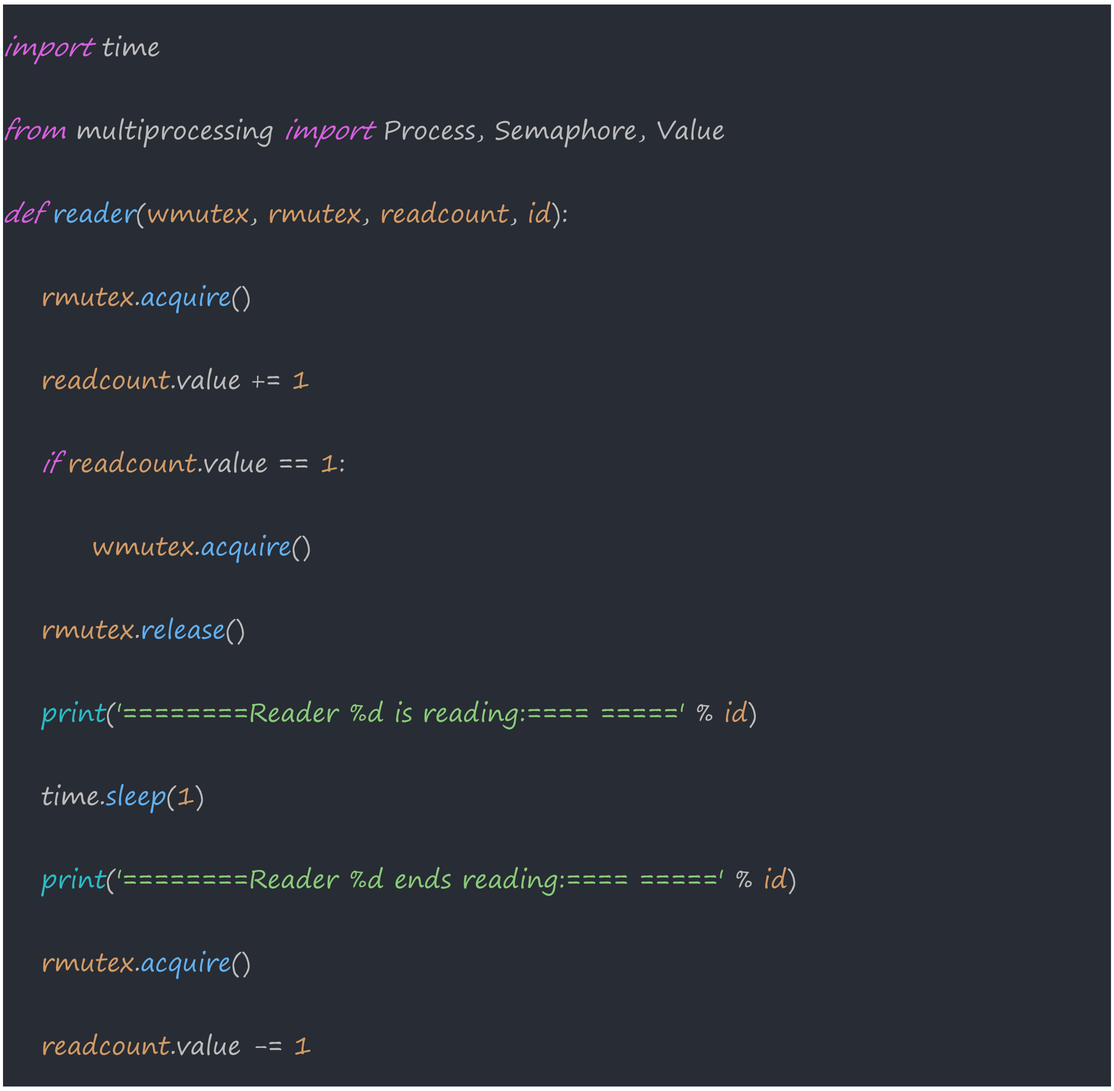
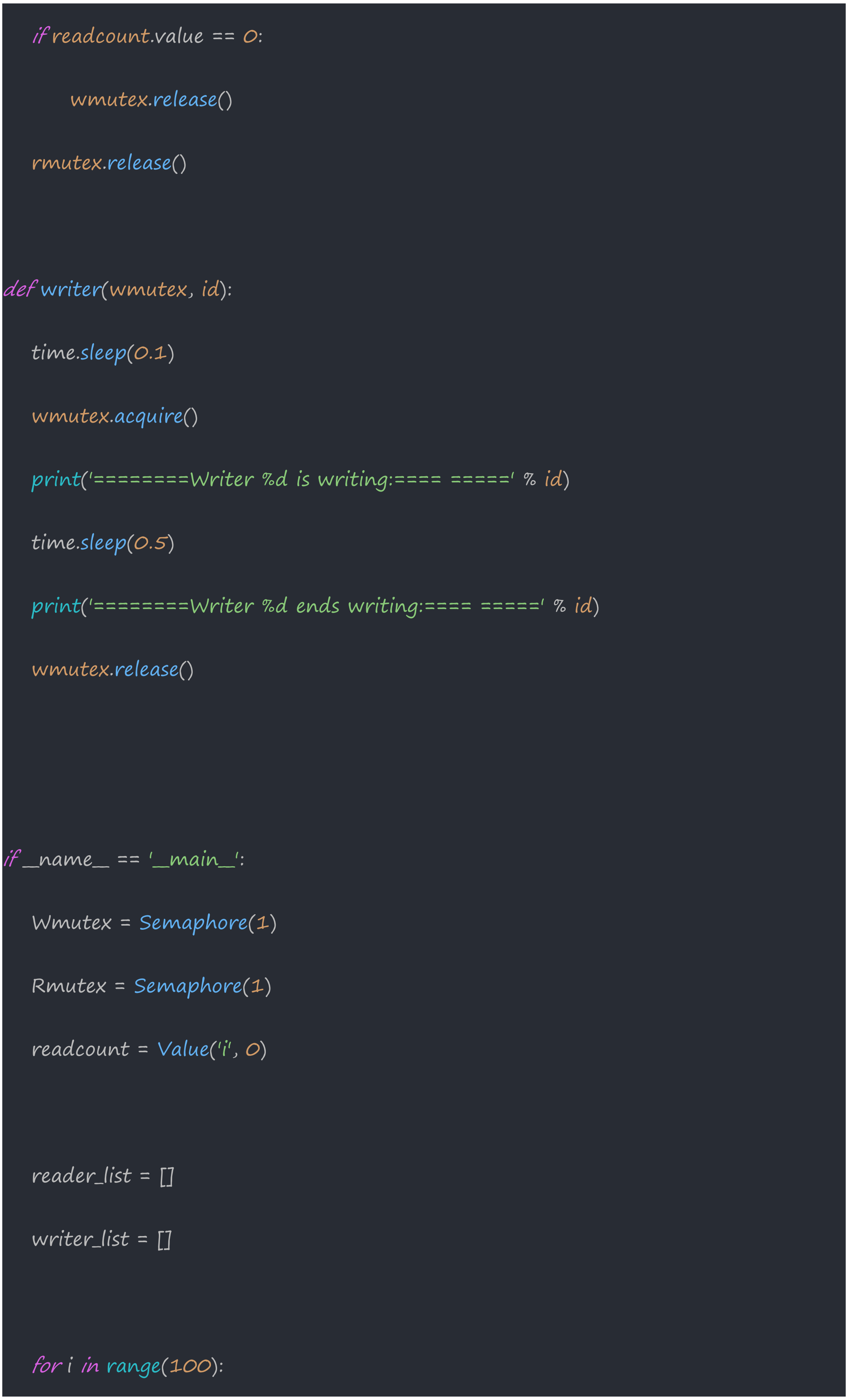


图 2 读者优先流程图

以下为实现代码：


```
import time
from multiprocessing import Process, Semaphore, Value
def reader(wmutex, rmutex, readcount, id):
    rmutex.acquire()
    readcount.value += 1
    if readcount.value == 1:
        wmutex.acquire()
    rmutex.release()
    print('========Reader %d is reading:==== =====' % id)
    time.sleep(1)
    print('========Reader %d ends reading:==== =====' % id)
    rmutex.acquire()
    readcount.value -= 1
```



```
    if readcount.value == 0:
        wmutex.release()
    rmutex.release()

def writer(wmutex, id):
    time.sleep(0.1)
    wmutex.acquire()
    print('========Writer %d is writing:==== =====' % id)
    time.sleep(0.5)
    print('========Writer %d ends writing:==== =====' % id)
    wmutex.release()


if __name__ == '__main__':
    Wmutex = Semaphore(1)
    Rmutex = Semaphore(1)
    readcount = Value('i', 0)

    reader_list = []
    writer_list = []

    for i in range(100):
        p = Process(target=reader, args=(Wmutex, Rmutex, readcount, i))

        time.sleep(0.2)

        reader_list.append(p)

        p.start()


    for j in range(100):

        p = Process(target=writer, args=(Wmutex, j))

        writer_list.append(p)

        p.start()
```

代码运行结果：

```
========Reader 0 is reading:==== =====

========Reader 1 is reading:==== =====

========Reader 2 is reading:==== =====

========Reader 3 is reading:==== =====

========Reader 4 is reading:==== =====

========Reader 0 ends reading:==== =====

========Reader 5 is reading:==== =====

========Reader 1 ends reading:==== =====

========Reader 6 is reading:==== =====

========Reader 2 ends reading:==== =====

========Reader 7 is reading:==== =====

========Reader 3 ends reading:==== =====

========Reader 8 is reading:==== =====
========Reader 4 ends reading:==== =====
========Reader 9 is reading:==== =====
========Reader 5 ends reading:==== =====
========Reader 10 is reading:==== =====
========Reader 6 ends reading:==== =====
========Reader 11 is reading:==== =====
========Reader 7 ends reading:==== =====
========Reader 12 is reading:==== =====
========Reader 8 ends reading:==== =====
========Reader 13 is reading:==== =====
========Reader 9 ends reading:==== =====
========Reader 14 is reading:==== =====
========Reader 10 ends reading:==== =====
========Reader 15 is reading:==== =====
========Reader 11 ends reading:==== =====
========Reader 16 is reading:==== =====
========Reader 12 ends reading:==== =====
========Reader 17 is reading:==== =====
========Reader 13 ends reading:==== =====
========Reader 18 is reading:==== =====
========Reader 14 ends reading:==== =====
========Reader 19 is reading:==== =====
========Reader 15 ends reading:==== =====
========Reader 16 ends reading:==== =====
========Reader 17 ends reading:==== =====

========Reader 18 ends reading:==== =====

========Reader 19 ends reading:==== =====

========Writer 1 is writing:==== =====

========Writer 1 ends writing:==== =====

========Writer 2 is writing:==== =====

========Writer 2 ends writing:==== =====

========Writer 3 is writing:==== =====

========Writer 3 ends writing:==== =====

========Writer 0 is writing:==== =====

========Writer 0 ends writing:==== =====

========Writer 4 is writing:==== =====

========Writer 4 ends writing:==== =====

========Writer 7 is writing:==== =====

========Writer 7 ends writing:==== =====

========Writer 9 is writing:==== =====

========Writer 9 ends writing:==== =====

========Writer 5 is writing:==== =====

========Writer 5 ends writing:==== =====

========Writer 6 is writing:==== =====

========Writer 6 ends writing:==== =====

========Writer 10 is writing:==== =====

========Writer 10 ends writing:==== =====

========Writer 8 is writing:==== =====
========Writer 8 ends writing:==== =====

========Writer 12 is writing:==== =====

========Writer 12 ends writing:==== =====

========Writer 11 is writing:==== =====

========Writer 11 ends writing:==== =====

========Writer 14 is writing:==== =====

========Writer 14 ends writing:==== =====

========Writer 16 is writing:==== =====

========Writer 16 ends writing:==== =====

========Writer 15 is writing:==== =====

========Writer 15 ends writing:==== =====

========Writer 19 is writing:==== =====

========Writer 19 ends writing:==== =====

========Writer 17 is writing:==== =====

========Writer 17 ends writing:==== =====

========Writer 13 is writing:==== =====

========Writer 13 ends writing:==== =====

========Writer 18 is writing:==== =====

========Writer 18 ends writing:==== =====


Process finished with exit code 0
```

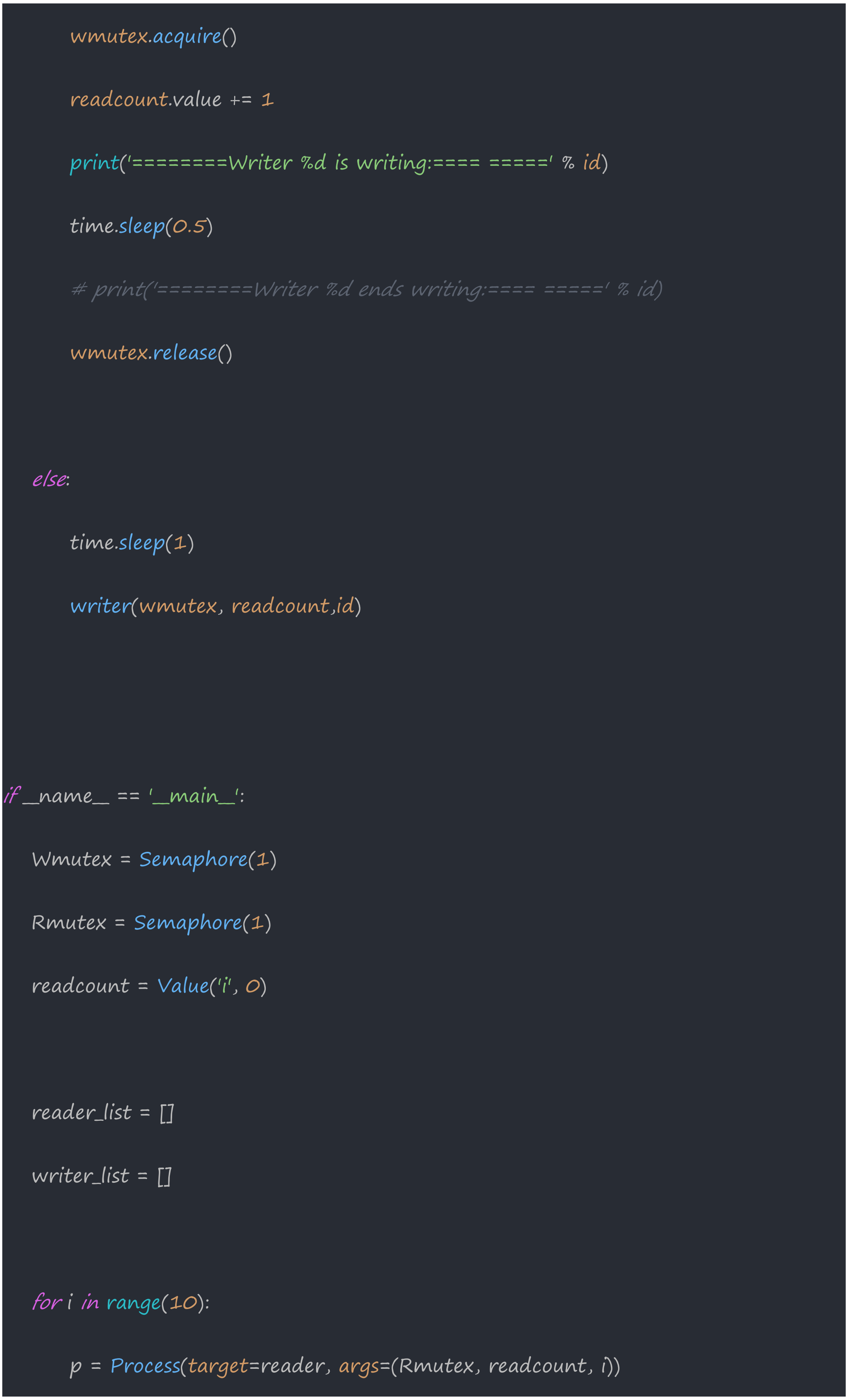
可以明显看出，读者优先运行

1.4公平调度

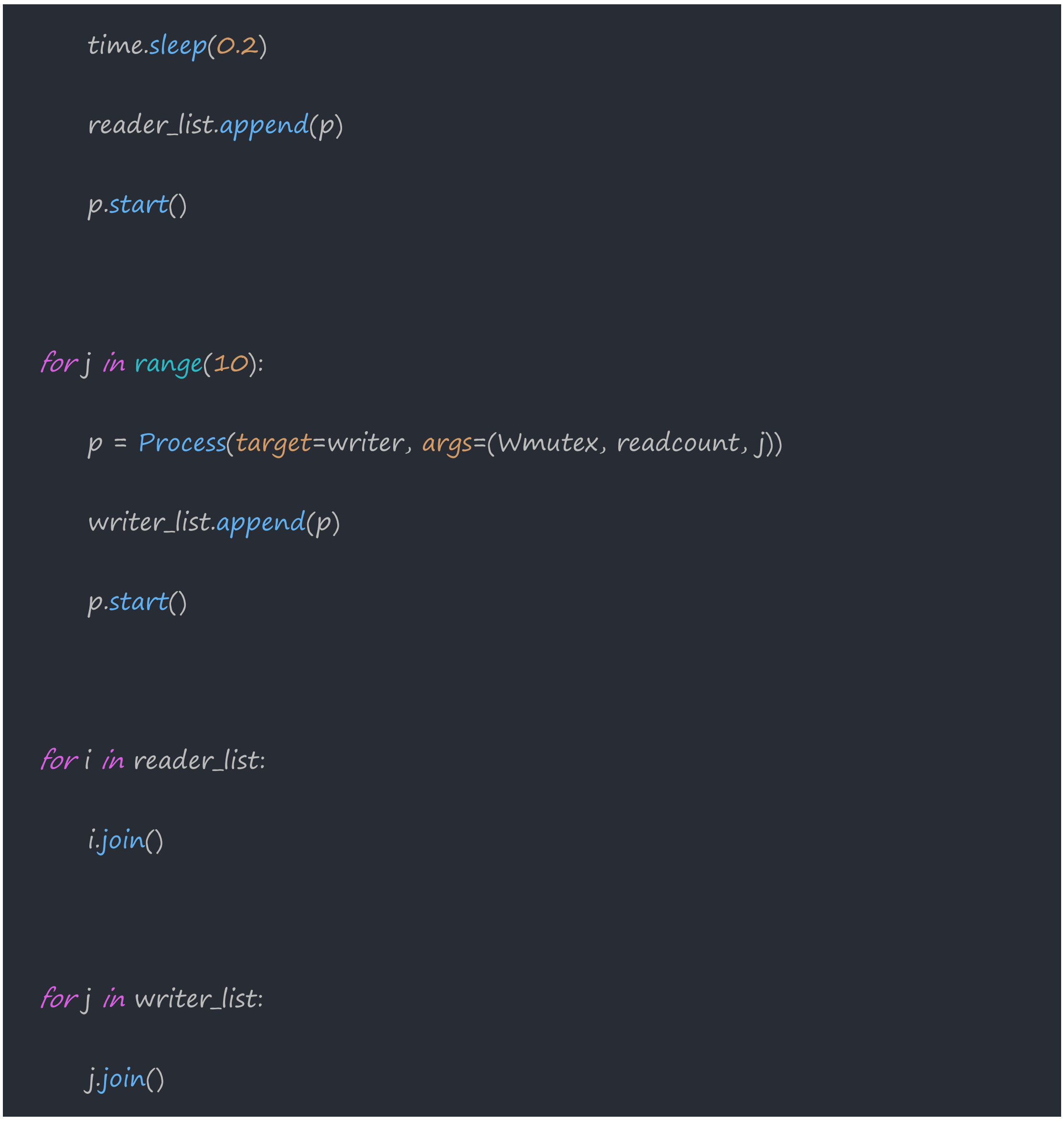
设计思路为设置一个进程共用的变量 count，每次读者或写者运行后便会自增 1，当读者和写者获得锁之前，会判断 count 的值，如为奇数，读者放行，如为偶数，写者放行。

以下为实现代码：

```python
import time

from multiprocessing import Process, Semaphore, Value



def reader(rmutex, readcount, id):

    if (readcount.value % 2) == 0:

        rmutex.acquire()

        readcount.value += 1

        print('========Reader %d is reading:==== =====' % id)

        time.sleep(0.5)

        # print('========Reader %d ends reading:==== =====' % id)

        rmutex.release()


    else:

        time.sleep(1)

        reader(rmutex, readcount, id)



def writer(wmutex, readcount, id):

    if (readcount.value % 2) == 1:
```



```
        wmutex.acquire()

        readcount.value += 1

        print('========Writer %d is writing:==== =====' % id)

        time.sleep(0.5)

        # print('========Writer %d ends writing:==== =====' % id)

        wmutex.release()


    else:

        time.sleep(1)

        writer(wmutex, readcount,id)




if __name__ == '__main__':

    Wmutex = Semaphore(1)

    Rmutex = Semaphore(1)

    readcount = Value('i', 0)


    reader_list = []

    writer_list = []


    for i in range(10):

        p = Process(target=reader, args=(Rmutex, readcount, i))
```



```python
        time.sleep(0.2)
        reader_list.append(p)
        p.start()

    for j in range(10):
        p = Process(target=writer, args=(Wmutex, readcount, j))
        writer_list.append(p)
        p.start()

    for i in reader_list:
        i.join()

    for j in writer_list:
        j.join()
```

运行结果：

```
========Reader 0 is reading:==== =====

========Writer 0 is writing:==== =====

========Reader 5 is reading:==== =====

========Writer 1 is writing:==== =====

========Reader 9 is reading:==== =====

========Writer 4 is writing:==== =====

========Reader 2 is reading:==== =====
========Writer 6 is writing:==== =====

========Reader 1 is reading:==== =====

========Writer 8 is writing:==== =====

========Reader 7 is reading:==== =====

========Writer 5 is writing:==== =====

========Reader 3 is reading:==== =====

========Writer 7 is writing:==== =====

========Reader 8 is reading:==== =====

========Writer 9 is writing:==== =====

========Reader 6 is reading:==== =====

========Writer 2 is writing:==== =====

========Reader 4 is reading:==== =====

========Writer 3 is writing:==== =====
```

可以明显看出，很公平。

1 网络异步爬虫

1.1 背景

本次演示挑选目标比较简单，爬取一个小网站的页面标签信息

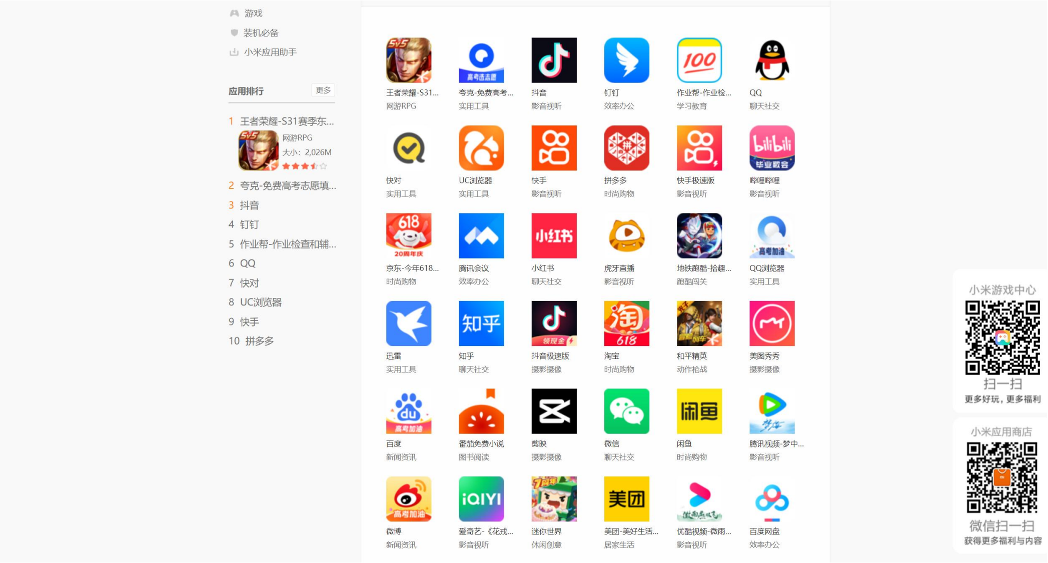


图 3 小米应用商店

1.2 进程与线程池

所谓的异步，就是多个事件互不阻塞，同时运行。

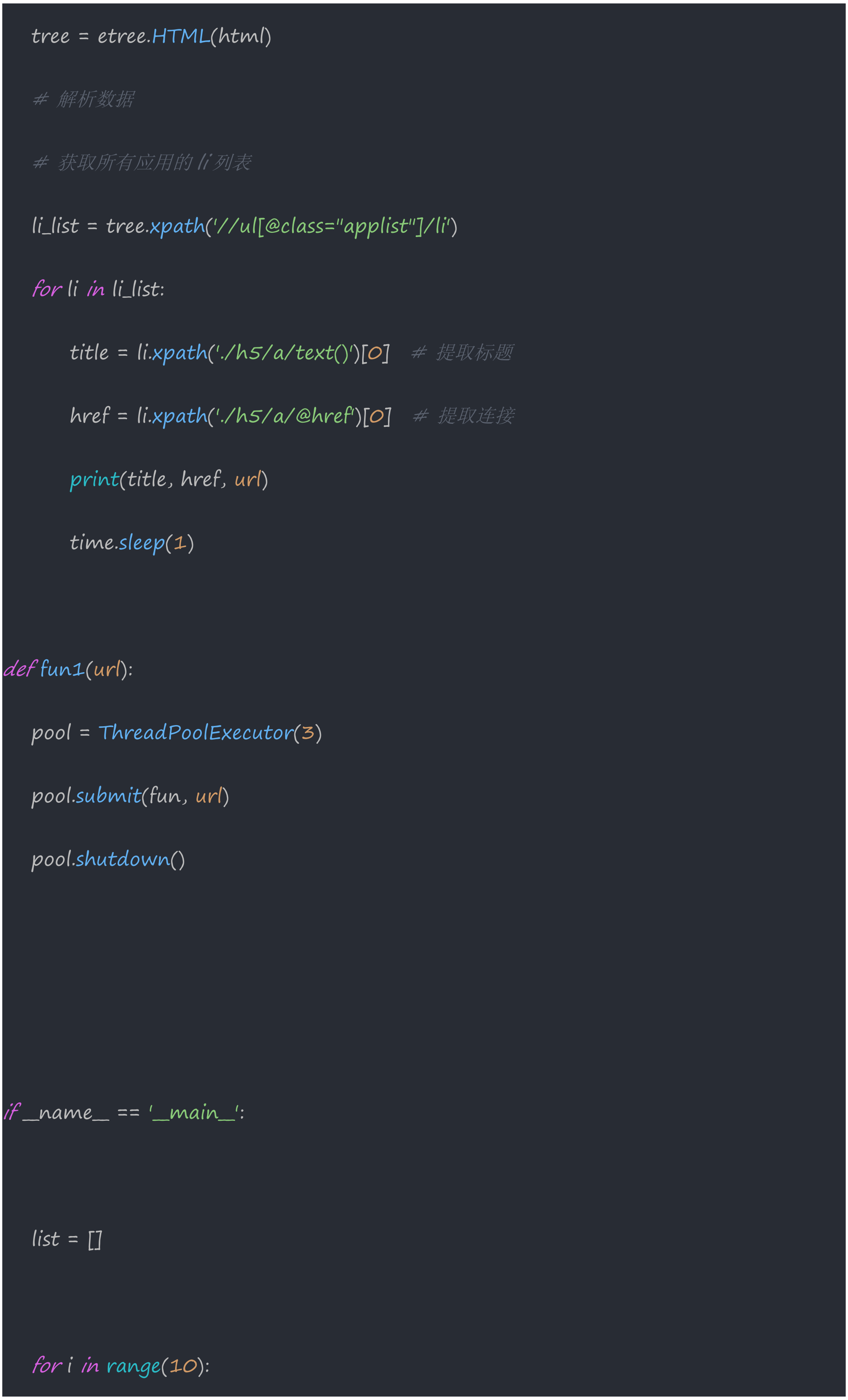
本次实验每一个页面用一个进程去抓取，而每一个进程会使用线程池加速抓取，为了让实验明显，每次抓取一个数据标签，会 time.sleep(1)，这样更容易看出进程是异步的。

1.3 爬取数据

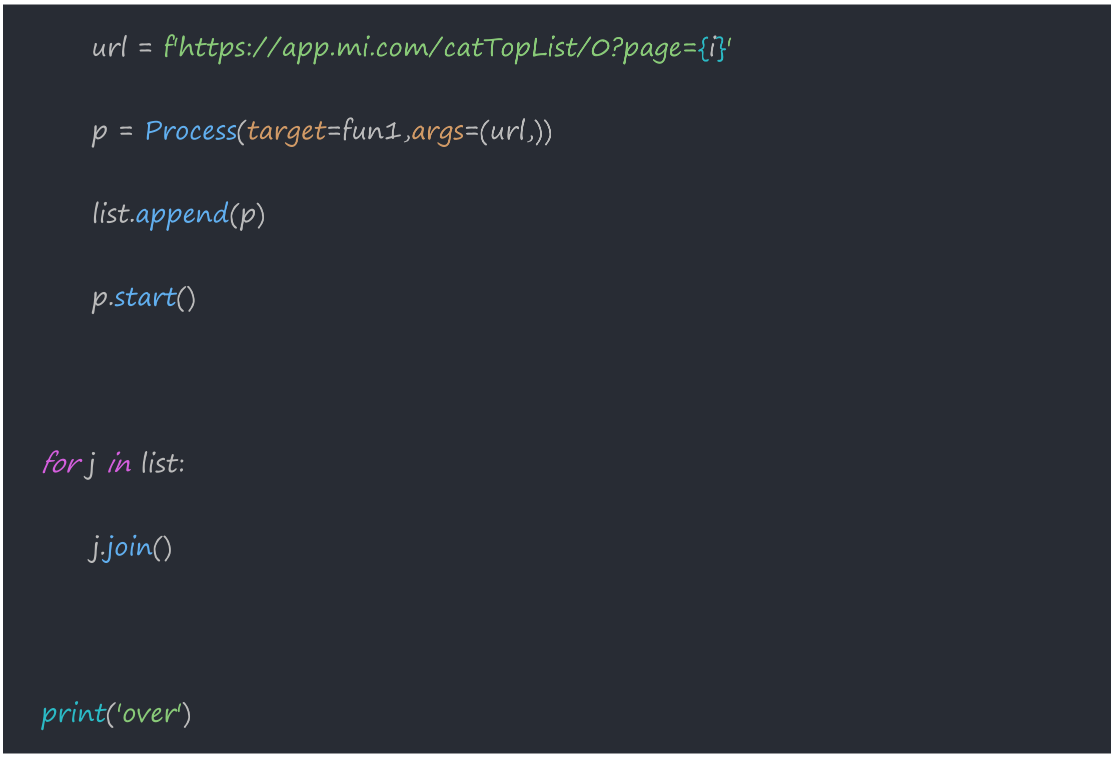
导入 request 包，向页面发起 http 请求，将 response 包中需要的信息用 xpath 定位并打印

代码如下：

```python
import time

import requests
from lxml import etree
from concurrent.futures import ThreadPoolExecutor
from multiprocessing import *

def fun(url):
    headers = {
        'user-agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/109.0.0.0 Safari/537.36'
    }
    response = requests.get(url, headers=headers)
    html = response.content.decode()
    # print(html)
```



```
    tree = etree.HTML(html)

    # 解析数据

    # 获取所有应用的 li 列表

    li_list = tree.xpath('//ul[@class="applist"]/li')

    for li in li_list:

        title = li.xpath('./h5/a/text()')[0]   # 提取标题

        href = li.xpath('./h5/a/@href')[0]   # 提取连接

        print(title, href, url)

        time.sleep(1)


def fun1(url):

    pool = ThreadPoolExecutor(3)

    pool.submit(fun, url)

    pool.shutdown()





if __name__ == '__main__':


    list = []


    for i in range(10):
```



```
    url = f'https://app.mi.com/catTopList/0?page={i}'

    p = Process(target=fun1,args=(url,))

    list.append(p)

    p.start()



for j in list:

    j.join()



print('over')
```

结果:

```
QQ音乐 /details?id=com.tencent.qqmusic
https://app.mi.com/catTopList/0?page=2
王者荣耀-S31赛季东方来客 /details?id=com.tencent.tmgp.sgame
https://app.mi.com/catTopList/0?page=1
番茄畅听 /details?id=com.xs.fm https://app.mi.com/catTopList/0?page=3
酷狗音乐 /details?id=com.kugou.android
https://app.mi.com/catTopList/0?page=2
夸克-免费高考志愿填报助手 /details?id=com.quark.browser
https://app.mi.com/catTopList/0?page=1
英雄联盟手游 /details?id=com.tencent.lolm
https://app.mi.com/catTopList/0?page=3
小猿搜题 /details?id=com.fenbi.android.solar
https://app.mi.com/catTopList/0?page=2
抖音 /details?id=com.ss.android.ugc.aweme
https://app.mi.com/catTopList/0?page=1
QQ阅读 /details?id=com.qq.reader
https://app.mi.com/catTopList/0?page=3
百度极速版 /details?id=com.baidu.searchbox.lite
https://app.mi.com/catTopList/0?page=2
钉钉 /details?id=com.alibaba.android.rimet
https://app.mi.com/catTopList/0?page=1
唯品会-新人享豪礼 /details?id=com.achievo.vipshop
https://app.mi.com/catTopList/0?page=3
得物-有毒的运动x潮流x好物 /details?id=com.shizhuang.duapp
```

https://app.mi.com/catTopList/0?page=2

作业帮-作业检查和辅导工具 /details?id=com.baidu.homework https://app.mi.com/catTopList/0?page=1

木鱼-电子木鱼 /details?id=com.hcj.wood https://app.mi.com/catTopList/0?page=3

西瓜视频 /details?id=com.ss.android.article.video https://app.mi.com/catTopList/0?page=2

QQ /details?id=com.tencent.mobileqq https://app.mi.com/catTopList/0?page=1

斗鱼 /details?id=air.tv.douyu.android https://app.mi.com/catTopList/0?page=3

咪咕视频-看NBA欧冠电竞体育直播 /details?id=com.cmcc.cmvideo https://app.mi.com/catTopList/0?page=2

快对 /details?id=com.kuaiduizuoye.scan https://app.mi.com/catTopList/0?page=1

禅之消方块-羊羊消除 /details?id=com.zentriple3d.mi https://app.mi.com/catTopList/0?page=3

醒图 /details?id=com.xt.retouch https://app.mi.com/catTopList/0?page=2

UC浏览器 /details?id=com.UCMobile https://app.mi.com/catTopList/0?page=1

Keep /details?id=com.gotokeep.keep https://app.mi.com/catTopList/0?page=3

我的世界 /details?id=com.netease.mc.mi https://app.mi.com/catTopList/0?page=2

快手 /details?id=com.smile.gifmaker https://app.mi.com/catTopList/0?page=1

作业精灵 /details?id=com.pcncn.jj https://app.mi.com/catTopList/0?page=3

Soul /details?id=cn.soulapp.android https://app.mi.com/catTopList/0?page=2

拼多多 /details?id=com.xunmeng.pinduoduo https://app.mi.com/catTopList/0?page=1

起点读书 /details?id=com.qidian.QDReader https://app.mi.com/catTopList/0?page=3

饿了么 /details?id=me.ele https://app.mi.com/catTopList/0?page=2

快手极速版 /details?id=com.kuaishou.nebula https://app.mi.com/catTopList/0?page=1

探探 /details?id=com.p1.mobile.putong https://app.mi.com/catTopList/0?page=3

网易云音乐 /details?id=com.netease.cloudmusic https://app.mi.com/catTopList/0?page=2

哔哩哔哩 /details?id=tv.danmaku.bili https://app.mi.com/catTopList/0?page=1

芒果TV-青年派计划 /details?id=com.hunantv.imgo.activity
https://app.mi.com/catTopList/0?page=3
影视大全 /details?id=com.le123.ysdq
https://app.mi.com/catTopList/0?page=2
京东-今年618，够省才京东 /details?id=com.jingdong.app.mall
https://app.mi.com/catTopList/0?page=1
百度地图 /details?id=com.baidu.BaiduMap
https://app.mi.com/catTopList/0?page=3
全民K歌 /details?id=com.tencent.karaoke
https://app.mi.com/catTopList/0?page=2
腾讯会议 /details?id=com.tencent.wemeet.app
https://app.mi.com/catTopList/0?page=1
58同城-招聘找工作租房家政买车 /details?id=com.wuba
https://app.mi.com/catTopList/0?page=3
MOMO陌陌 /details?id=com.immomo.momo
https://app.mi.com/catTopList/0?page=2
小红书 /details?id=com.xingin.xhs
https://app.mi.com/catTopList/0?page=1
大学搜题酱-作业帮大学版 /details?id=com.zmzx.college.search
https://app.mi.com/catTopList/0?page=3
支付宝 /details?id=com.eg.android.AlipayGphone
https://app.mi.com/catTopList/0?page=2
虎牙直播 /details?id=com.duowan.kiwi
https://app.mi.com/catTopList/0?page=1
逃跑吧！少年 /details?id=com.bairimeng.dmmdzz.mi
https://app.mi.com/catTopList/0?page=3
王者营地 /details?id=com.tencent.gamehelper.smoba
https://app.mi.com/catTopList/0?page=2
地铁跑酷-拾趣缤纷六一 /details?id=com.kiloo.subwaysurf
https://app.mi.com/catTopList/0?page=1
手机天猫 /details?id=com.tmall.wireless
https://app.mi.com/catTopList/0?page=3
Edge /details?id=com.microsoft.emmx
https://app.mi.com/catTopList/0?page=2
QQ浏览器 /details?id=com.tencent.mtt
https://app.mi.com/catTopList/0?page=1
央视频 /details?id=com.cctv.yangshipin.app.androidp
https://app.mi.com/catTopList/0?page=3
淘特 /details?id=com.taobao.litetao
https://app.mi.com/catTopList/0?page=2
迅雷 /details?id=com.xunlei.downloadprovider
https://app.mi.com/catTopList/0?page=1
阿里巴巴 /details?id=com.alibaba.wireless
https://app.mi.com/catTopList/0?page=3

快影 /details?id=com.kwai.videoeditor
https://app.mi.com/catTopList/0?page=2
知乎 /details?id=com.zhihu.android
https://app.mi.com/catTopList/0?page=1
光·遇 /details?id=com.netease.sky.mi
https://app.mi.com/catTopList/0?page=3
喜马拉雅-听书听播客 /details?id=com.ximalaya.ting.android
https://app.mi.com/catTopList/0?page=2
抖音极速版 /details?id=com.ss.android.ugc.aweme.lite
https://app.mi.com/catTopList/0?page=1
中国联通 /details?id=com.sinovatech.unicom.ui
https://app.mi.com/catTopList/0?page=3
WiFi 万能钥匙 /details?id=com.snda.wifilocating
https://app.mi.com/catTopList/0?page=2
淘宝 /details?id=com.taobao.taobao
https://app.mi.com/catTopList/0?page=1
汤姆猫跑酷 /details?id=com.outfit7.talkingtomgoldrun.mi
https://app.mi.com/catTopList/0?page=3
中国移动-全国统一官方服务平台
/details?id=com.greenpoint.android.mc10086.activity
https://app.mi.com/catTopList/0?page=2
和平精英 /details?id=com.tencent.tmgp.pubgmhd
https://app.mi.com/catTopList/0?page=1
原神 /details?id=com.miHoYo.ys.mi
https://app.mi.com/catTopList/0?page=3
美团外卖 /details?id=com.sankuai.meituan.takeoutnew
https://app.mi.com/catTopList/0?page=2
美图秀秀 /details?id=com.mt.mtxx.mtxx
https://app.mi.com/catTopList/0?page=1
轻颜相机 /details?id=com.gorgeous.lite
https://app.mi.com/catTopList/0?page=3
转转-二手官方验 /details?id=com.wuba.zhuanzhuan
https://app.mi.com/catTopList/0?page=2
百度 /details?id=com.baidu.searchbox
https://app.mi.com/catTopList/0?page=1
穿越火线-枪战王者 /details?id=com.tencent.tmgp.cf
https://app.mi.com/catTopList/0?page=3
航旅纵横 /details?id=com.umetrip.android.msky.app
https://app.mi.com/catTopList/0?page=2
番茄免费小说 /details?id=com.dragon.read
https://app.mi.com/catTopList/0?page=1
作业帮口算-家长检查作业好帮手 /details?id=com.zybang.parent
https://app.mi.com/catTopList/0?page=3
高德地图-高德打车，导航公交地铁 /details?id=com.autonavi.minimap

https://app.mi.com/catTopList/0?page=2

剪映 /details?id=com.lemon.lv https://app.mi.com/catTopList/0?page=1

点淘-淘宝直播官方APP /details?id=com.taobao.live https://app.mi.com/catTopList/0?page=3

X浏览器 /details?id=com.mmbox.xbrowser.pro https://app.mi.com/catTopList/0?page=2

微信 /details?id=com.tencent.mm https://app.mi.com/catTopList/0?page=1

肯德基 /details?id=com.yek.android.kfc.activitys https://app.mi.com/catTopList/0?page=3

爱奇艺极速版-《花戎》全网独播 /details?id=com.qiyi.video.lite https://app.mi.com/catTopList/0?page=2

闲鱼 /details?id=com.taobao.idlefish https://app.mi.com/catTopList/0?page=1

国家反诈中心 /details?id=com.hicorenational.antifraud https://app.mi.com/catTopList/0?page=3

七猫免费小说 /details?id=com.kmxs.reader https://app.mi.com/catTopList/0?page=2

腾讯视频-梦中的那片海全网独播 /details?id=com.tencent.qqlive https://app.mi.com/catTopList/0?page=1

快看-原快看漫画 /details?id=com.kuaikan.comic https://app.mi.com/catTopList/0?page=3

百度贴吧 /details?id=com.baidu.tieba https://app.mi.com/catTopList/0?page=2

微博 /details?id=com.sina.weibo https://app.mi.com/catTopList/0?page=1

抖音火山版 /details?id=com.ss.android.ugc.live https://app.mi.com/catTopList/0?page=3

一甜相机 /details?id=com.kwai.m2u https://app.mi.com/catTopList/0?page=2

爱奇艺-《花戎》全网独播 /details?id=com.qiyi.video https://app.mi.com/catTopList/0?page=1

BOSS直聘 /details?id=com.hpbr.bosszhipin https://app.mi.com/catTopList/0?page=3

今日头条-搜高考报志愿 /details?id=com.ss.android.article.news https://app.mi.com/catTopList/0?page=2

迷你世界 /details?id=com.minitech.miniworld.TMobile.mi https://app.mi.com/catTopList/0?page=1

京喜特价-省出新生活 /details?id=com.jd.jdlite https://app.mi.com/catTopList/0?page=3

粤省事 /details?id=com.digitalgd.dgyss https://app.mi.com/catTopList/0?page=2

美团-美好生活小帮手 /details?id=com.sankuai.meituan https://app.mi.com/catTopList/0?page=1

```
QQ邮箱 /details?id=com.tencent.androidqqmail
https://app.mi.com/catTopList/0?page=3
学习强国 /details?id=cn.xuexi.android
https://app.mi.com/catTopList/0?page=2
优酷视频-微雨燕双飞 全网独播 /details?id=com.youku.phone
https://app.mi.com/catTopList/0?page=1
云闪付 /details?id=com.unionpay
https://app.mi.com/catTopList/0?page=3
作业帮家长版 /details?id=com.zuoyebang.knowledge
https://app.mi.com/catTopList/0?page=2
百度网盘 /details?id=com.baidu.netdisk
https://app.mi.com/catTopList/0?page=1
biubiu 加速器-畅玩全球 /details?id=com.njh.biubiu
https://app.mi.com/catTopList/0?page=3
Over
```

可以看出三个页面的抓取是异步的。

3 心得体会

进程，线程，协程，锁的机制十分重要，这次的Python实训加强了我的理解与认知。

参考文献：

[1] 闫雷鸣 Python高级编程——并行编程

致谢：感谢闫雷鸣老师的陪伴，感念一起学习的时光。