20377199 赵芮箐 第12周作业

作业内容： 以爬取网易云歌单为例，练习多线程的使用。

1. 获取一个分类下的所有歌单的id。观察url可以发现其页码规律：https://music.163.com/#/discover/playlist/?order=hot&cat=%E8%AF%B4%E5%94%B1&limit=35&offset=35。offset是本页开始的数据位置，第1页是0，第2页是35。分类参数是utf-8编码后的汉字"说唱"，使用str.encode可以处理。
2. 对每个id，获取歌单的详细信息，至少包括：歌单的封面图片（需把图片保存到本地）、歌单标题、创建者id、创建者昵称、介绍、歌曲数量、播放量、添加到播放列表次数、分享次数、评论数。可以自行实现其他信息的获取。基本信息汇总到同一张表中，以csv文件保存。https://music.163.com/playlist?id=3037221581
3. 要求使用生产者-消费者模式实现，要求1作为生产者，每次请求后产生新的任务交给消费者，消费者执行要求2。
4. （附加）爬虫程序往往需要稳定运行较长的时间，因此如果你的程序突然中断或异常（比如网络或被封），如何能够快速从断点重启？
5. （附加）爬虫程序往往需要比较友好的状态输出，因此可否专门有一个线程 动态地进行输出更新，来显示当前的状态，比如程序连续运行的时长，要完成的 总页面数，其中有多少已被爬取，已收集的文件占用了多少空间，大概还需要多 少时间才能完成，预计需要耗费多少硬盘空间等。

Part1: 获取一个分类下所有歌单的id

```
def producer(q, url):
    """
    生产者：获取一个分类下的所有歌单的id
    """
    headers = {
        'User-Agent': 'Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/63.0.3239.132 Safari/537.36'
    }
    response = requests.get(url=url, headers=headers)
    html = response.text
    soup = BeautifulSoup(html, 'html.parser')
    ids = soup.select('.dec a')                # 获取包含歌单详情页网址的标签
    q.put(ids)
```

Part2: 对每个id获取歌单的详细信息

```
def consumer(q):
    """
    消费者：对每个id获取歌单的详细信息
    """
    headers = {
        'User-Agent': 'Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/63.0.3239.132 Safari/537.36'
    }
    while True:
        ids = q.get()
        if ids is None:
            break
        else:
            for i in ids:
                url = 'https://music.163.com/' + i['href']       # 获取歌单详情页地址
                response = requests.get(url=url, headers=headers)
                html = response.text
                soup = BeautifulSoup(html, 'html.parser')
                # 获取歌单标题
                title = soup.select('h2')[0].get_text().replace(',', ', ')
                # 获取封面图片
                img = soup.select('img')[0]['data-src']
```

```
                res = requests.get(img)
                image = Image.open(BytesIO(res.content))
                image.save('week12/cover/'+ title + '.jpg')
                # 获取创建者id
                idd = soup.select('.s-fc7')[0]['href'].split('=')[-1]
                # 获取创建者昵称
                nickname = soup.select('.s-fc7')[0].get_text()
                # 获取歌单介绍
                if soup.select('#album-desc-more'):
                    text = soup.select('#album-desc-more')
[0].get_text().replace('\n', '').replace(',', ', ')
                else:
                    text = '无'
                # 歌单内歌曲数
                songs = soup.select('#playlist-track-count')[0].get_text()
                # 播放量
                plays = soup.select('.s-fc6')[0].get_text()
                # 添加进播放列表次数
                adds = soup.find('a','u-btni u-btni-fav')['data-count']
                # 分享次数
                shares = soup.find('a','u-btni u-btni-share')['data-count']
                # 歌单评论数
                comments = soup.select('#cnt_comment_count')
[0].get_text().replace('评论', '0')
                #print(title,idd,nickname,text,songs,plays,adds,shares,comments)
                # 将详情页信息写入CSV文件中
                with open('week12/music_message.csv', 'a+') as f:
                    f.write(title + ',' + idd + ',' + nickname + ',' + text +
',' + songs + ',' + plays + ',' + adds + ',' + shares + ',' + comments + '\n')
```

Part3: 主函数实现生产者-消费者模式

```
if __name__=='__main__':
    head = ['歌单标题','id','昵称','介绍','歌曲数量','播放量','添加到播放列表次数','分享次
数','评论数']
    with open('week12/music_message.csv', 'w', encoding='utf-8') as f:
        csv_writer = csv.writer(f)          # csv格式写入文件file
        csv_writer.writerow(head)

    q = Queue()
    plist = []
    clist = []
    for i in range(0, 1300, 35):
        time.sleep(2)
        url = 'https://music.163.com/discover/playlist/?cat=后摇
&order=hot&limit=35&offset=' + str(i)
        p = Thread(target=producer, args=(q, url))
        plist.append(p)
    for p in plist:
        p.start()
    for t in plist:
        p.join()
    for i in range(30):
        c = Thread(target=consumer, args=(q,))
        clist.append(c)
```

```
    for c in clist:
        c.start()
    for c in clist:
        q.put(None)
```

Part4: 如果你的程序突然中断或异常，如何能够快速从断点重启（附加）

修改producer代码如下：

```
try:
    response = requests.get(url=url, headers=headers)
    html = response.text
    soup = BeautifulSoup(html, 'html.parser')
    ids = soup.select('.dec a')              # 获取包含歌单详情页网址的标签
    q.put(ids)
except:
    print(f"Error, url: {url}")
```

我实现的想法比较简单，因为爬取网易云歌单的任务抓取的url都是有规律的，所以因为特殊情况停止之后可以直接从下一个url抓取的。所以在每次停止的时候我记录一下断点即可，下一次重启的时候就从该url再开始。

对于更大规模或更复杂的任务会有更好and更框架的写法，简单了解了一下，见Ref

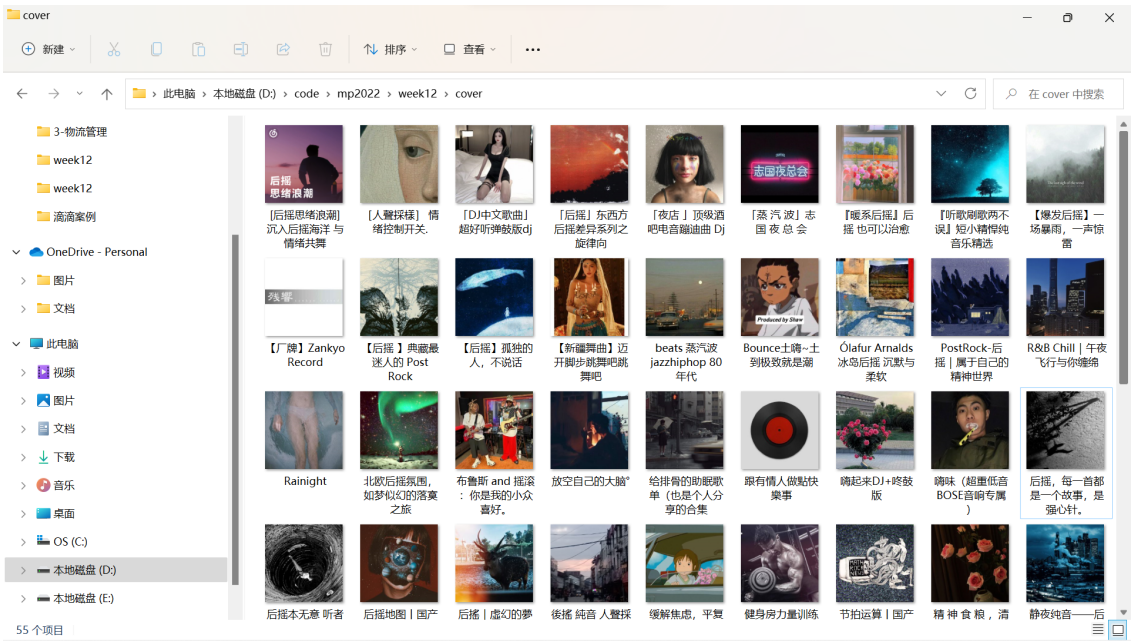
Part5: 实现一个线程显示当前爬取状态（附加）

> 可以查看程序连续运行的时长，要完成的总页面数，其中有多少已被爬取，已收集的文件占用了多少空间，大概还需要多少时间才能完成，预计需要耗费多少硬盘空间等。

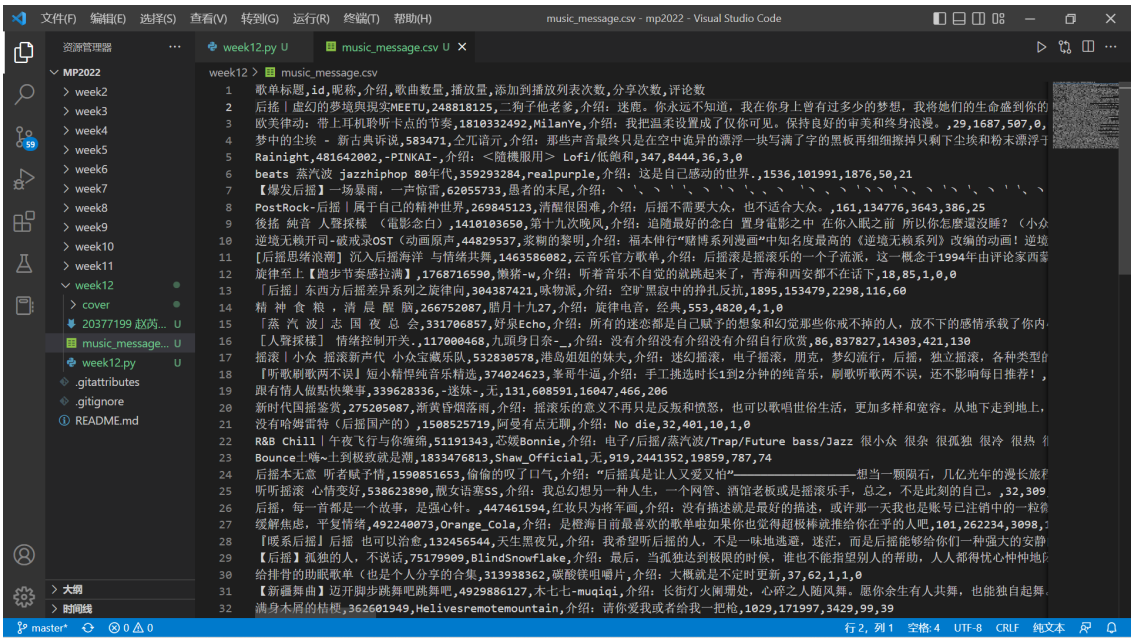
```
# 实时显示当前爬取状态的函数
def status():
    start = time.time()
    while True:
        time.sleep(0.5)
        now_time = time.time()-start
        with open('week12/music_message.csv', 'rb') as f:
            length = len(f.readlines()) - 1
        size = os.path.getsize('week12/music_message.csv')
        end_time = now_time / (length + 1) * 1300
        end_size = size / (length + 1) * 1300

        print(f'\r运行时间：{now_time: .4f} / {end_time: .4f} \
                爬取页数：{length} / 1300 \
                文件内存：{size} / {end_size: .0f}', end='')

# 并在main里设置成守护线程
if __name__=='__main__':
    s = Thread(target=status)
    s.daemon = True
    s.start()
```

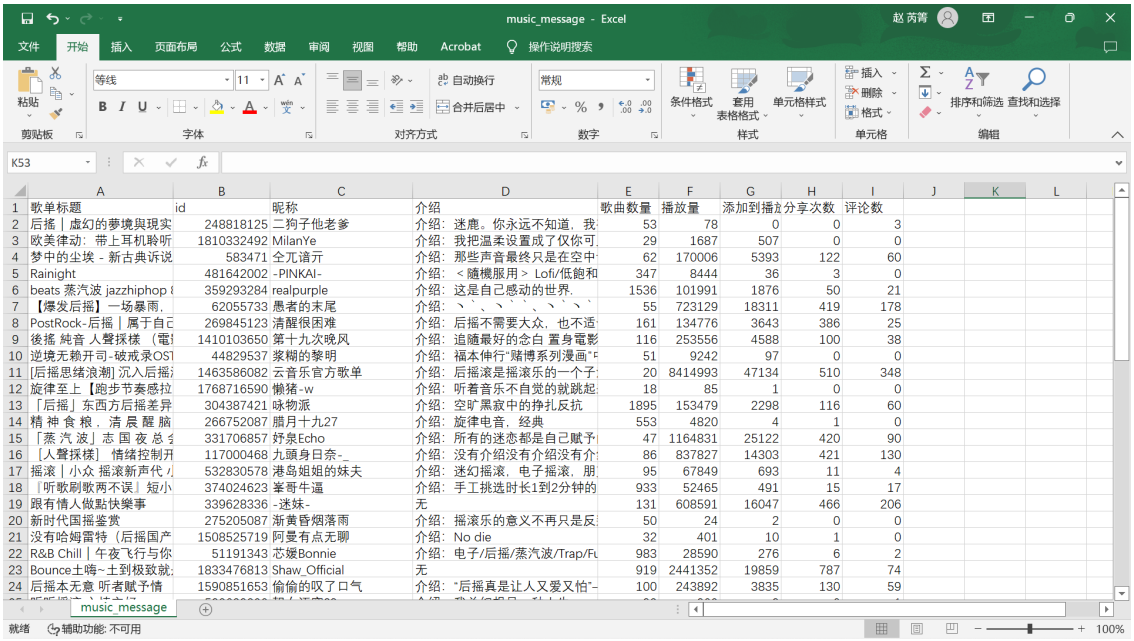
结果展示：

- 封面图片：



- 歌单信息：





	A	B	C	D	E	F	G	H	I
1	歌单标题	id	昵称	介绍	歌曲数量	播放量	添加到播放	分享次数	评论数
2	后摇丨虚幻的梦境與现实	248818125	二狗子他老爹	介绍：迷鹿。你永远不知道，我	53	78	0	0	3
3	欧美律动：带上耳机聆听	1810332492	MilanYe	介绍：我把温柔设置成了仅你可	29	1687	507	0	0
4	梦中的尘埃 - 新古典诉说	583471	仝兀谙亓	介绍：那些声音最终只是在空中	62	170006	5393	122	60
5	Rainight	481642002	-PINKAI-	介绍：＜随機服用＞ Lofi/低鲍和	347	8444	36	3	0
6	beats 蒸汽波 jazzhiphop	359293284	realpurple	介绍：这是自己感动的世界.	1536	101991	1876	50	21
7	【爆发后摇】一场暴雨，	62055733	愚者的末尾	介绍：丶丶丶丶丶丶丶丶丶丶	55	723129	18311	419	178
8	PostRock-后摇丨属于自己	269845123	清醒很困难	介绍：后摇不需要大众，也不适	161	134776	3643	386	25
9	後摇 纯音 人聲採樣 （電	1410103650	第十九次晚风	介绍：追随最好的念白 置身電影	116	253556	4588	100	38
10	逆境无赖开司-破戒录OST	44829537	浆糊的黎明	介绍：福本伸行"賭博系列漫画"F	51	9242	97	0	0
11	[后摇思绪浪潮] 沉入后摇	1463586082	云音乐官方歌单	介绍：后摇滚是摇滚乐的一个子	20	8414993	47134	510	348
12	旋律至上【跑步节奏感拉	1768716590	懒猪-w	介绍：听着音乐不自觉的就跳起	18	85	1	0	0
13	「后摇」东西方后摇差异	304387421	咏物派	介绍：空旷黑寂中的挣扎反抗	1895	153479	2298	116	60
14	精 神 食 粮 ， 清 晨 醒 脑	266752087	腊月十九27	介绍：旋律电音，经典	553	4820	4	1	0
15	「蒸 汽 波」志 国 夜 总	331706857	妤泉Echo	介绍：所有的迷恋都是自己赋予	47	1164831	25122	420	90
16	［人聲採樣］ 情绪控制开	117000468	九頭身日奈-_	介绍：没有介绍没有介绍没有介	86	837827	14303	421	130
17	摇滚丨小众 摇滚新声代	532830578	港岛姐姐的妹夫	介绍：迷幻摇滚，电子摇滚，朋	95	67849	693	11	4
18	「听歌刷歌两不误」短小	374024623	峯哥牛逼	介绍：手工挑选时长1到2分钟的	933	52465	491	15	17
19	跟有情人做點快樂事	339628336	-迷妹-	无	131	608591	16047	466	206
20	新时代国摇鉴赏	275205087	渐黄昏烟落雨	介绍：摇滚乐的意义不再只是反	50	24	2	0	0
21	没有哈姆雷特（后摇国产	1508525719	阿曼有点无聊	介绍：No die	32	401	10	1	0
22	R&B Chill丨午夜飞行与你	51191343	芯媛Bonnie	介绍：电子/后摇/蒸汽波/Trap/Fu	983	28590	276	6	2
23	Bounce土嗨~土到极致就	1833476813	Shaw_Official	无	919	2441352	19859	787	74
24	后摇本无意 听者赋予情	1590851653	偷偷的叹了口气	介绍："后摇真是让人又爱又怕"-	100	243892	3835	130	59

- 状态显示：

```
运行时间：  65.4647 /  630.4008          爬取页数：134 / 1300          文件内存：57245 /  551248
运行时间：  103.2392 /  353.1868         爬取页数：379 / 1300          文件内存：156334 /  534827
运行时间：  148.8509 /  318.2666         爬取页数：607 / 1300          文件内存：250582 /  535784
运行时间：  165.7526 /  305.6432         爬取页数：704 / 1300          文件内存：285467 /  526393
```

> 这里总时间和内存占用的估计都处理的比较简单，就均值*总任务数嘿嘿...

代码：

https://github.com/rachhhhing/mp2022_python/blob/master/week12/week12.py

Ref：

- 网易云歌单爬取代码参考：

  https://blog.csdn.net/weixin_53275593/article/details/126547090

  https://blog.csdn.net/i_csdn_water/article/details/107076440

- 断点续爬：

  用scrapy_redis：https://blog.csdn.net/qq_46092061/article/details/119957688