**实验4：多线程路径暴力破解实验报告**

姓名：江继军 学号：21373222 班级：213921

实验日期：2024年4月17日 实验地点：一号馆406 成绩：

1. **实验目的**

1.掌握Web交互式信息收集中对目标网站进行路径暴力破解的基本方法。

2.熟悉基于Python的多线程编程基本方法，掌握文件锁的使用。

3.基于但不限于requests、urllib3、threading库，编写多线程的网络路径暴力破解程序。

**二、实验内容**

1. 环境搭建

1. 安装Python requests、urllib3、threading库及相关支持库，并验证安装成功。

2. 编写多线程网络路径暴力破解程序

1. 编写url\_generator函数：生成暴力破解的URL路径。
2. 编写url\_test函数：测试给定的URL是否可访问。
3. 编写run\_tasks\_threads函数：实现以上任务的多线程运行。

**三、实验环境**

分项列出主要实验环境（主机/操作系统、Python开发环境、Python版本、第三方库版本等）：

1. 操作系统：Win11
2. Python开发环境：pycharm-community
3. Python版本：3.9
4. requests版本：2.31.0
5. urllib3版本：2.2.1

**四、实验步骤**

写出设置线程锁、生成暴力破解的URL路径、测试URL路径是否存在、多线程任务执行的步骤（采用1.2.3.4的序号方式列出详细步骤；写明编写代码；贴出效果示意图，例如生成URL路径、测试URL路径、多线程任务执行结果等）*本次报告最多8页，内容清晰，关键代码完整*。

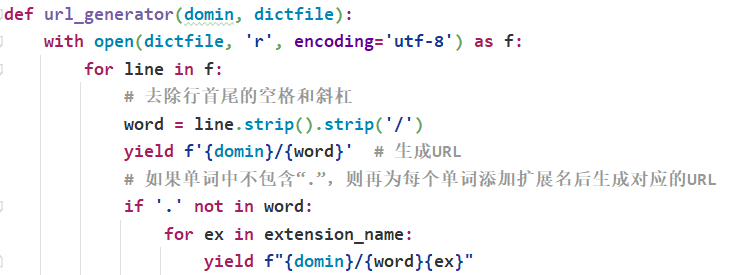
1. 生成暴力破解的URL路径：

编写url\_generator函数，通过设置的扩展名列表、暴力破解字典，结合给定的域名生成暴力破解URL路径，推荐使用yield生成器，避免消耗大量内存（写明代码，给出生成URL路径的结果截图）。

暴力破解字典：github.com/rootphantomer/Blasting\_dictionary （可以选择其它字典或根据网站路径手动编写字典）*有效的非手动字典改进方法+（2-4）分，和网站路径结合后自动生成针对某网站的高效字典+6分*

网站域名：http://testfire.net（可选择其它网站，但注意因网站技术区别，可能无法采用示例程序方法暴力破解）*选择其他网站并有破解难度，采用有效破解措施，酌情+（2-6）分。*

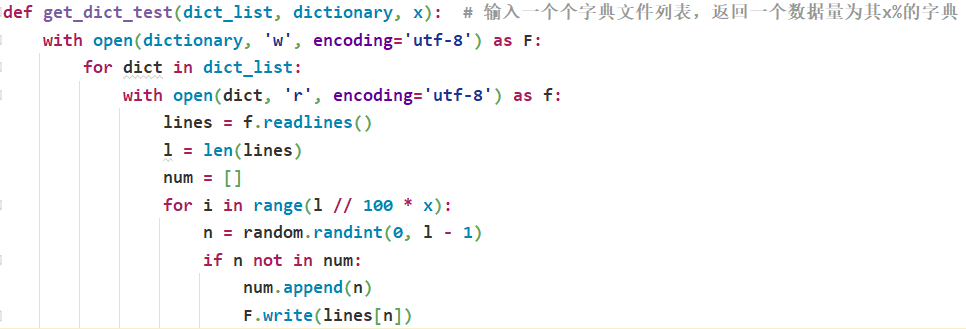
url\_generator函数如下



生成的URL路径结果如下



其中暴力破解字典选取如下：根据英语词频统计规律，选取单词、词组频率较高的组合成文件名字典，再与其他域名爆破常用字典结合形成最终的爆破字典，由于字典数据量过大，所以只选取其中5%内容来进行实验，代码如下：



1. 测试生成的URL是否存在：

编写url\_test函数，测试给定的URL是否可访问，并将结果写入（.TXT）文件。其中写入文件需要通过锁的获取和释放保证线程安全（写明所编写的代码，给出URL测试和文件写入的结果截图，分析文件锁的作用）。

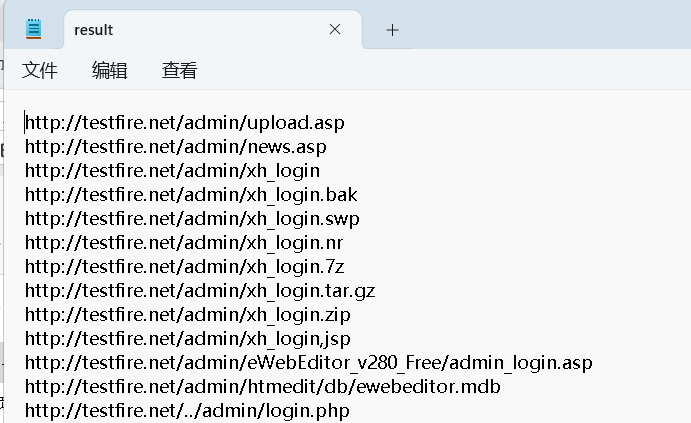
url\_test函数如下



URL测试结果如下



存入文件截图如下



当多个线程同时对一个文件进行写入操作时，如果不加锁，可能会出现以下问题：

数据覆盖：一个线程正在写入数据，另一个线程也开始写入，可能会导致前一个线程写入的数据被覆盖。

数据不一致：如果多个线程同时写入，由于写入的顺序无法保证，可能会导致文件中的数据状态与预期不符。

因此，需要使用文件锁来确保在任何时刻，只有一个线程能够写入文件。这样可以确保数据的一致性和完整性。

1. 多线程任务设置：

编写run\_tasks\_threads 函数，实现多线程任务的运行，使用ThreadPoolExecutor类创建线程池运行多个任务，合理设置最大线程数，实现以上两步操作的并发处理（写明所编写的代码，说明线程数设置）。*采用更为高效的多线程方法，酌情+（1-4）分。*

run\_tasks\_threads函数如下

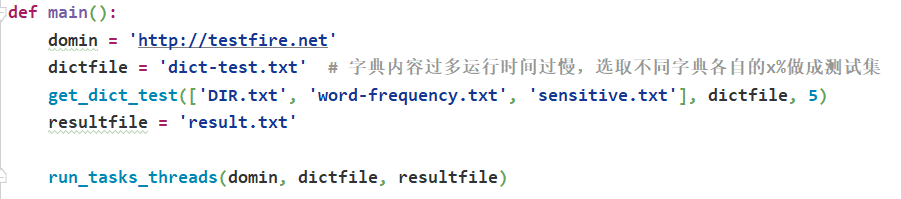


其中MAX\_THEARDS变量是同时运行的最大线程数量，此处设置为64。

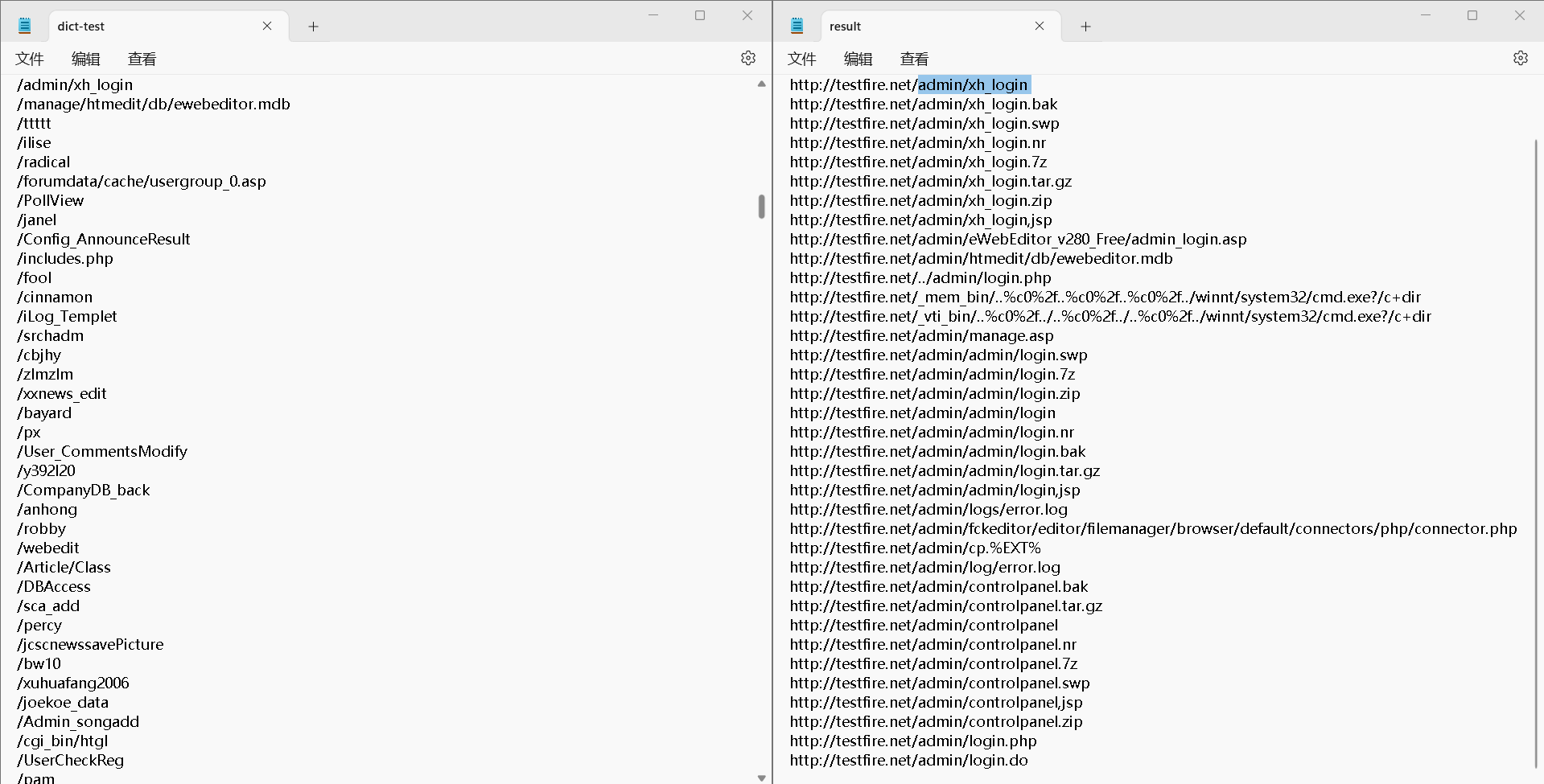
1. 多线程网络路径暴力破解：

编写主函数，整合以上函数，实现指定目标网站、字典文件和输出文件，进行目标网站多线程路径暴力破解（写明所编写的代码，给出路径暴力破解结果截图）。

主函数代码如下



结果如下



**五、问题和解决方法**

1. 在对URL进行暴力破解时发现无论运行多少时间多少次只会输出一个结果，开始以为是字典质量差，检查后发现是忘记释放锁，加解锁操作一定是一个完整的过程，这提示我以后编程要更加仔细。
2. 在解决上述问题后，发现运行时间也还是很长，且效率较低，查看词典文件后发现文件数据量过大，所以选取数据量的5%以供测试。

**六、实验总结**

结合暴力破解思想，谈一下此次实验体会，并进行实验总结。

在对域名破解时，由于是要进行暴力破解，所以程序执行的效率就尤为重要，采用更高效的爆破字典、多线程等可显著提高破解效率，但多线程如果涉及到共享资源的读写操作时要注意加解锁，做到互斥访问；同时爆破字典的选取也同样重要，可以根据所要爆破的网站特点来构造高质量字典。

总结：本次实验实现了对目标网站进行路径暴力破解，通过选取合适的字典和多线程加速这一过程的实施。暴力破解是一种比较古老但是有效的攻击手法，其原理是通过对特定字典的穷举来达到获得密码或者类似信息的目的。对目标网站进行路径暴力破解，可以使渗透者进一步的了解目标网站的结构。在实现多线程路径暴力破解的过程中，我深刻体会到了并行计算的优势以及良好字典的重要性，可以明显提高暴力破解的效率和速度。

**七、附件**

提交**可执行**的原程序压缩包，随电子版实验报告提交，文件名称为学号+姓名+实验4。