**中文网页爬虫程序使用手册**

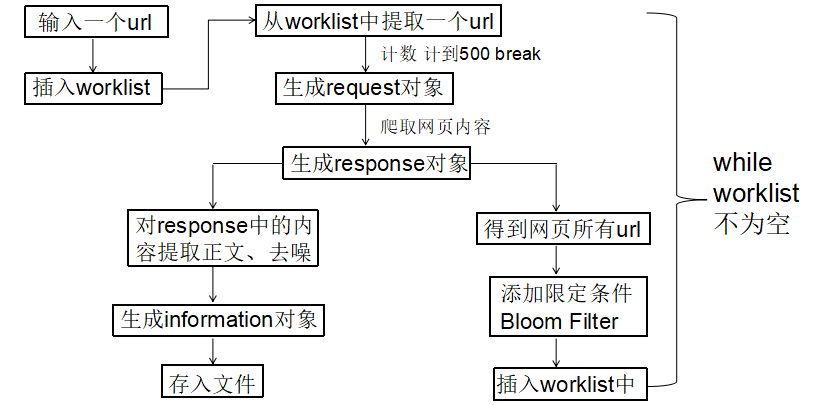
**1.主要功能介绍**

本系统主要功能为从指定的基于http协议的网站，持续爬取网页文本，并去除html标签仅保留正文。

**2.原理介绍**

首先，用户输入想要爬取页面的网址。其后运行本系统，即可开始爬取首个页面。从爬的页面中，提取超链接中的url，经过和已处理的url、待处理的url比较，看是否经过处理，若没有，则将其放入待处理队列中，即worklist中。最后从html文本中提取正文，并按照层次存储到文件。遍历待处理的url队列，重复前两个操作，直到队列中没有新的url。

整体流程如下图所示：



**2.1.爬虫**

爬虫部分主要由socket发送http请求和接收响应，通过报文找出新的url，再继续访问获取响应来实现。通过两个类实现，分别为Request和Response类

获得raw\_url即网页未处理过的url

通过函数parseHostAndPagePath函数提取出host以及path

通过SetHeader函数生成请求报文头

通过GetPageContent函数发送请求报文，

并返回一个Response对象，

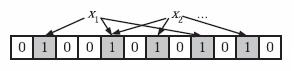
包含有响应报文以及爬取的网页内容

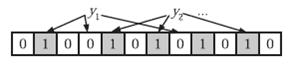
**2.2.布隆过滤器**

Bloom Filter是一种空间效率很高的随机数据结构，它利用位数组很简洁地表示一个集合，并能判断一个元素是否属于这个集合。但同时Bloom Filter的这种高效是有一定代价的：在判断一个元素是否属于某个集合时，有可能会把不属于这个集合的元素误认为属于这个集合（false positive）。在本程序中，错误率大概为0.00008%。

Bloom Filter使用k个相互独立的哈希函数（Hash Function），它们分别将集合中的每个元素映射到{1,…,m}的范围中。对任意一个元素x，第i个哈希函数映射的位置hi(x)就会被置为1（1≤i≤k）。

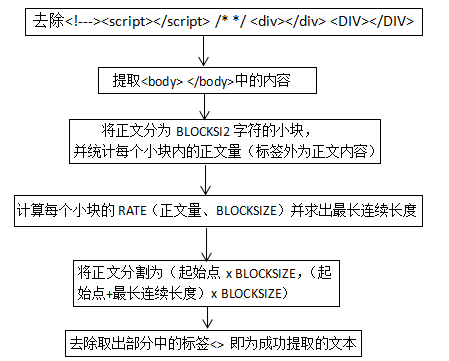
在判断*y*是否属于这个集合时，我们对*y*应用*k*次哈希函数，如果所有*hi(y)*的位置都是*1*（*1*≤*i*≤*k*），那么我们就认为*y*是集合中的元素，否则就认为*y*不是集合中的元素。





**2.3.文本分布**

程序工作流程如下：



**2.4.存储**

将获取到的网页信息存储于本地，将信息输出到指定txt文本中。

**3.使用方式**

**3.1.输入**

在main函数中输入网址



**3.2.输出**

爬取到的每个有效网页的正文与其他信息的文件（pages/hostname/filename）

**4.技术规范**

**4.1代码规范**

程序用C++实现

函数命名：驼峰式命名

变量命名：小写命名

类的声明：头文件

成员函数定义：cpp文件

**4.2注释规范**

要求：

/\*=================================

\*函数名称

\*功能简介

\*参数列表和解释

\*返回值类型

\*=================================\*/

**4.3编译规范**

使用的操作系统和gcc版本如下：

|  |  |
| --- | --- |
| 操作系统 | GCC/G++ Version |
| Ubuntu16 | 5.3.2 |
| Ubuntu17 | 6.3.0 |
| centos7 | 4.9.2 |
| centos7 | 6.0.22 |

使用makefile编译，makefile内容如下：

all: Spider splitString.o main TextFilter.o

@echo "making spider"

g++ main.o Spider.o splitString.o BloomFilter.o Saver.o TextFilter.o -o spider -std=c++11

Spider:Spider.cpp Spider.h BloomFilter.o TextFilter.o Saver.o

g++ -c Spider.cpp -std=c++11

main:main.cpp

g++ -c main.cpp -std=c++11

#spider:spider.cpp Spider

# g++ -c spider.cpp -std=c++11

splitString.o:splitString.cpp

g++ -c splitString.cpp -std=c++11

BloomFilter.o:BloomFilter.cpp BloomFilter.h

g++ -c BloomFilter.cpp -std=c++11

TextFilter.o:TextFilter.cpp TextFilter.h

g++ -c TextFilter.cpp -std=c++11

Saver.o:Saver.cpp Saver.h

g++ -c Saver.cpp -std=c++11

clear:

rm \*.o spider

**4.4库**

|  |  |
| --- | --- |
| fstream | netdb.h |
| sys/types.h | arpa/inet.h |
| iostream | sys/socket.h |
| cstring | sys/stat.h |
| unistd.h | bits/algorithmfwd.h |
| vector | fcnt1.h |
| string.h | cstdio |
| regex | stdio.h |
| iconv.h | dirent.h |
| stdlib.h | algorithm |
| time.h | bitset |
| cmath | cstdio |
| cstdlib | functional |

1. **待改进的地方**
2. 输入网址需从main 函数中输入，与用户交互性不高
3. 存在一定丢包率，若收到响应为空，则直接放弃
4. 代码运行速率待提高