## 设计目的

爬虫一般指网络爬虫。网络爬虫（又称为网页蜘蛛，网络机器人，在FOAF社区中间，更经常的称为网页追逐者），是一种按照一定的规则，自动地抓取万维网信息的程序或者脚本。

网络在我们的生活中越来越重要，网络的信息量也越来越大，研究该课题可以更好的理解网络爬虫在搜索引擎中的作用以及网络爬虫的原理。

本系统是人民网新闻爬虫系统，采集网络上的热点新闻，方便用户对实时社会新闻热点的快速掌握，新闻内容的整体归档，大大提升了用户的工作效率，节约了时间。

## 系统的需求及其基本功能

网络爬虫是一个专门从网络世界上下载网页并分析网页的程序。它将下载的网页和收集到的信息存储在本地，以供为后期各种数据的分析做数据基础。网络爬虫的工作原理是多个网页的连接。去请求网页的数据，得到返回结果，并分析处理该数据，最后保存起来，以供后期使用。

## 总体设计方案

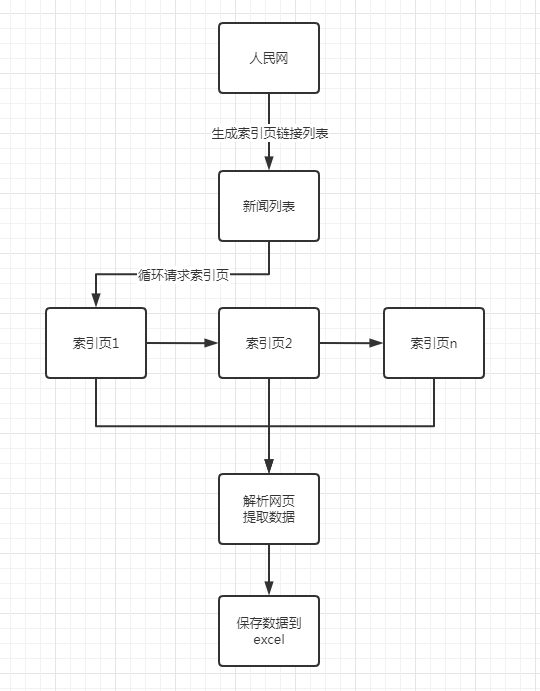
### 构造人民网爬取链接的列表

### 提取每页新闻索引页的新闻链接

### 访问新闻内容页，并解析提取标题，发布时间，内容等信息

### 把得到数据保存到excel表中

### 流程图



## 详细设计项目源程序代码

import requests

from lxml import etree

import openpyxl

import pymysql

url = 'http://gs.people.com.cn/GB/183283/index{}.html'

home = 'http://gs.people.com.cn'

db = pymysql.connect(user="root", password="123456", host="localhost", database="spider")

cursor = db.cursor()

# 生成表格

def excel():

wb = openpyxl.Workbook()

ws = wb.active

ws.append(['标题', '发布时间', '作者', '内容', '网址'])

cursor.execute('select \* from renminwang')

results = cursor.fetchall()

for li in results:

ws.append(li)

wb.save('结果.xlsx')

# 解析索引页

def spider\_index(n: int):

response = requests.get(url.format(n))

response.encoding = 'GBK'

html = etree.HTML(response.text)

index\_url = html.xpath('//ul[contains(@class,"list\_16")]/li/a/@href')

index\_url = [home + str(li) for li in index\_url]

return index\_url

# 解析内容页

def parser\_content(url: str):

try:

response = requests.get(url)

response.encoding = response.apparent\_encoding

html = etree.HTML(response.text)

title = html.xpath('//h1[@id="newstit"]/text()')[0].strip()

datetime = str(html.xpath('//div[@class="box01"]/div/text()')[0].strip()).split(' ')[0]

author = html.xpath('//p[@class="author"]/text()')[0].strip()

content = html.xpath('//div[@class="box\_con"]')[0]

content = str(etree.tostring(content, encoding="utf-8", pretty\_print=True, method="html"),

encoding='utf8').strip()

print(title, datetime)

cursor.execute(

"INSERT INTO `renminwang` (`title`, `datetime`, `author`, `content`, `url`) VALUES ('%s', '%s', '%s', '%s', '%s');" % (

title, datetime, author, content, url))

db.commit()

except Exception as e:

pass

# 入口

def main(star: int, end: int):

for n in range(star, end + 1):

index\_url = spider\_index(n)

for url in index\_url:

parser\_content(url)

excel()

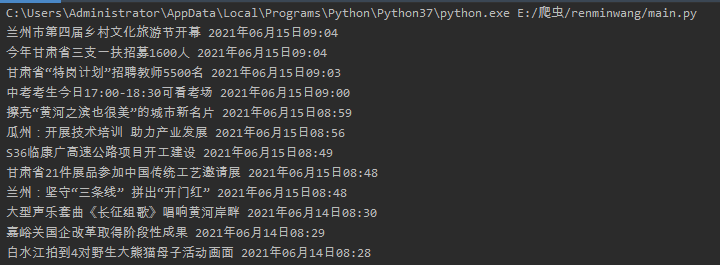
db.close()

if \_\_name\_\_ == '\_\_main\_\_':

main(1, 100)

## 运行与测试

运行结果如下：





## 成员分工

## 总结与收获

通过本系统的设计编写，对网络爬虫的技术有了更深的理解，对网络爬虫的使用更加熟练，熟悉掌握了requests模块的使用、xpath提取元素的规则。对于网络爬虫这个庞大的知识体系，现在只是一点皮毛，在以后的时间里，我将继续研究网络爬虫技术，系统能更上一层楼。

## 指导教师评语