多进程下Python爬虫的示例：

例：程序crawler\_test/crawler\_test10.py

import requests  
import re  
import time  
from multiprocessing import Pool  
import crawler\_test1  
  
  
# 获取信息的函数  
def re\_scraper(url):  
 res = requests.get(url, headers=crawler\_test1.headers)  
 ids = re.findall('<h2>(.\*?)</h2>', res.text, re.S)  
 contents = re.findall('<div class="content">.\*?</span>(.\*?)</span>', res.text, re.S)  
 laughs = re.findall('<span class="stats-vote"><i class="number">(\d+)</i>', res.text, re.S)  
 comments = re.findall('<i class="number">(\d+)</i> 评论', res.text, re.S)  
 for id1, content, laugh, comment in zip(ids, contents, laughs, comments):  
 info1 = {  
 'id': id1,  
 'content': content,  
 'laugh': laugh,  
 'comment': comment  
 }  
 return info1  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 urls = ['http://www.qiushibaike.com/text/page/{0}'.format(str(i)) for i in range(1, 36)]  
 start\_1 = time.time()  
 for url in urls:  
 re\_scraper(url)  
  
 end\_1 = time.time()  
 print('串行爬虫', end\_1 - start\_1)  
  
 start\_2 = time.time()  
 pool = Pool(processes=2) # 2个进程  
 for url in urls:  
 pool.apply\_async(re\_scraper, args=(url,))  
  
 pool.close() # 禁止新的子进程创建  
 pool.join() # 等待所有子进程执行完毕

end\_2 = time.time()  
 print('两个进程', end\_2 - start\_2)  
  
 start\_3 = time.time()  
 pool = Pool(processes=4) # 4个进程  
 for url in urls:  
 pool.apply\_async(re\_scraper, args=(url,))  
  
 pool.close()  
 pool.join()  
  
 end\_3 = time.time()  
 print('四个进程', end\_3 - start\_3)

输出为：

串行爬虫 12.397147178649902

两个进程 5.056334733963013

四个进程 2.321122169494629

例：程序crawler\_test/crawler\_test11.py

import requests  
from lxml import etree  
import pymongo  
from multiprocessing import Pool  
from mongodb\_test import mongo\_port  
from mongodb\_test import mongo\_host  
from crawler\_test1 import headers  
import time  
  
client = pymongo.MongoClient(mongo\_host, mongo\_port)  
mydb = client['mydb']  
jianshu = mydb['jianshu']  
  
  
def get\_jianshu(url):  
 html = requests.get(url, headers=headers)  
 selector = etree.HTML(html.text)  
 # 获取总的标签  
 infos = selector.xpath('//ul[@class="note-list"]/li')  
 for info in infos:  
 try:  
 author = info.xpath('div/div[1]/a/text()')[0]  
 title = info.xpath('div/a/text()')[0]  
 content = info.xpath('div/p/text()')[0].strip()  
 comment = info.xpath('div/div[1]/a[2]/text()')[0]  
 like = info.xpath('div/div[1]/span[1]/text()')[0].strip()  
 detailed\_url = "http://www.jianshu.com" + info.xpath('a[1]/@href')[0]  
 time = get\_time(detailed\_url)  
  
 data = {  
 'author': author,  
 'title': title,  
 'time': time,  
 'content': content,  
 'comment': comment,  
 'like': like  
 }  
 jianshu.insert\_one(data)  
 except IndexError:  
 pass  
  
  
def get\_time(url):  
 html = requests.get(url, headers=headers)  
 selector = etree.HTML(html.text)  
 time = selector.xpath('//span[@class="publish-time"]/text()')[0]  
 return time  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 urls = ['http://www.jianshu.com/c/bDHhpk?order\_by=added\_at&page={0}'.format(str(i)) for i in range(1, 1001)]  
 start1 = time.time()  
 for url in urls:  
 get\_jianshu(url)  
 end1 = time.time()  
 print('串行爬虫', end1 - start1)  
  
 start2 = time.time()  
 pool = Pool(processes=4)  
 for url in urls:  
 pool.apply\_async(get\_jianshu, args=(url,))  
 pool.close()  
 pool.join()  
 end2 = time.time()  
 print('四个进程', end2 - start2)