Python的进程和线程

**1、什么是进程**

对于操作系统来说，一个任务就是一个进程（Process）。比如打开一个浏览器就是启动一个浏览器进程，打开一个记事本就启动了一个记事本进程。

由于每个进程至少要干一件事，所以，一个进程至少有一个线程，即主线程。

**2、什么是线程**

有些进程还不止同时干一件事，比如Word，它可以同时进行打字、拼写检查、打印等事情。在一个进程内部，要同时干多件事，就需要同时运行多个“子任务”，我们把进程内的这些“子任务”称为线程（Thread）。同一个进程的不同线程之间共享Memary，不同的线程之间申请缩的过程，称为线程的数据的同步 。

不同的进程之间的Mem是相互独立的，一般通过管道或者队列来进行通信。

并行编程解决的问题，大部分可以理解为进程之间通信、线程之间同步的问题

**3、Python的多进程包multiprocessing**

使用的一般步骤如下：

[复制代码](javascript:void(0);)

#1. 导入包

from multiprocessing import Process

#2. 实例化对象

a=Process(target=func,args=(args,))

#3. 开始运行

a.start()

#4. 等待终止

a.join()

[复制代码](javascript:void(0);)

实例化对象，target是指的要让线程执行的任务的函数名，args参数传入函数的参数，元祖格式

**4、Python的多线程包threading**

[复制代码](javascript:void(0);)

#导入包

from threading import Thread

#实例化对象

a=Thread(target=func,args=(args,))

#开始运行

a.start()

#等待终止

a.join()

[复制代码](javascript:void(0);)

**5、并行编程——计算密集型任务举例**

假如我们要实现一个函数把COUNT减少到1，分别用多进程和多线程来实现，看一下运行时间

[复制代码](javascript:void(0);)

from threading import Thread

from multiprocessing import Process

import time

def countdown(n):

while n > 0:

n -= 1

COUNT = 100000000

def thread\_process\_job(n, Thread\_Process, job):

"""

n: 多线程或多进程数

Thread\_Process: Thread／Process类

job: countdown任务

"""

local\_time = time.time()

# 实例化多线程或多进程

threads\_or\_processes = [Thread\_Process(target=job, args=(COUNT // n,)) for i in range(n)]

for t in threads\_or\_processes:

t.start() # 开始线程或进程，必须调用

for t in threads\_or\_processes:

t.join() # 等待直到该线程或进程结束

print(n, Thread\_Process.\_\_name\_\_, " run job need ", time.time() - local\_time)

if \_\_name\_\_ == "\_\_main\_\_":

print( "Multi Threads")

for i in [1, 2, 4]:

thread\_process\_job(i, Thread, countdown)

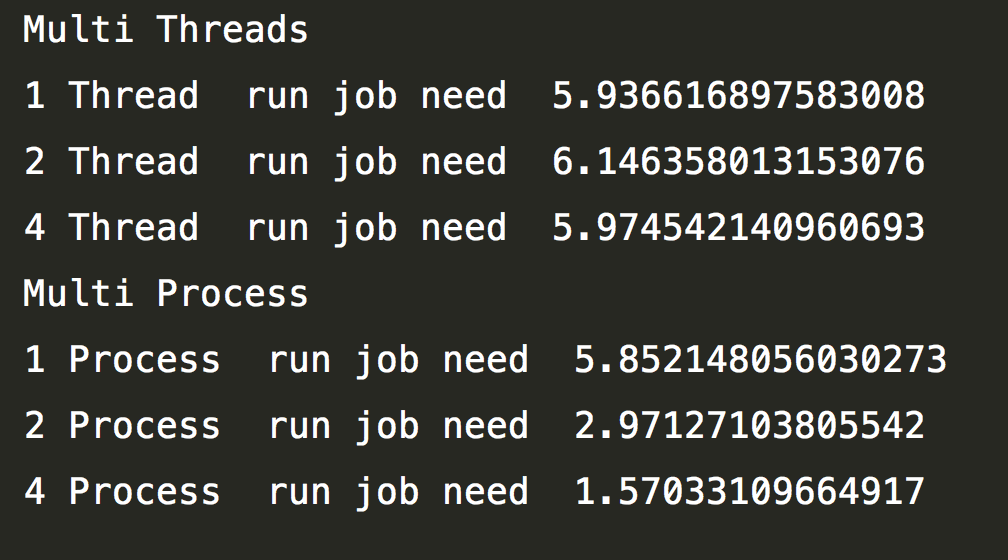
print("Multi Process")

for i in [1, 2, 4]:

thread\_process\_job(i, Process, countdown)

[复制代码](javascript:void(0);)

 在我的电脑上的运行结果如下：



可以看出，多线程的情况，当2个线程、4个线程并没有比1个线程使用的时间明显减少，2个线程的情况用时反而更长。

**为什么多线程反而慢了？**

计算密集型任务，占用的是CPU的时间，Python多线程之间有一个调度问题，全局解释器锁GIL。当有多个线程的时候，线程并不是并行在运行，他会申请一个全局解释器锁，谁申请到了，谁运行。线程在串行运行，所以并没有加快。

当电脑有多核的时候，进程是并行运行的，所以时间会缩短。

**6、参考资料**

[廖雪峰的官方网站](https://www.liaoxuefeng.com/wiki/0014316089557264a6b348958f449949df42a6d3a2e542c000/001431927781401bb47ccf187b24c3b955157bb12c5882d000)