

python的多线程与多进程

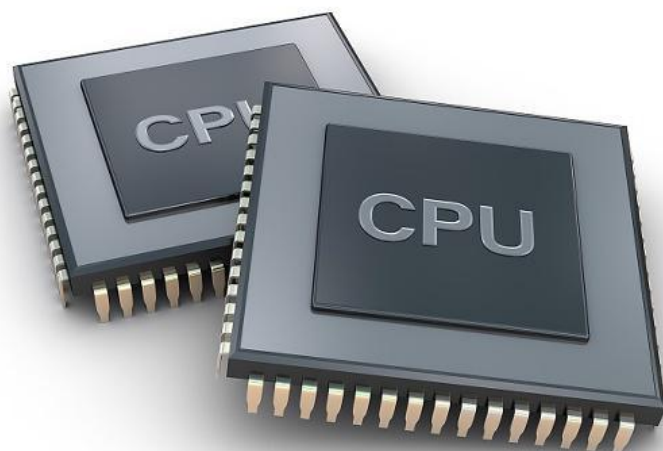
郭靖愕然
18-12-13 17:49

最近用python给朋友的手游工作室写了一个游戏辅助程序，实现服务器里二十个模拟器的自动运行，包括账户登陆、切换、挂机、整理背包等简单操作。最初的实现方案是采取多线程，为每一个模拟器开启一个线程，主线程定期检查各子线程的状态，并进行相应处理。程序完成后测试的结果并不理想，因为python的多线程无法实现线程的并行执行，最后采取的方案是采用多进程，为每个模拟器启动独立的进程，程序运行速度得到明显提高。本文简单梳理一下python多线程与多进程的相关知识，希望对初学者有所帮助。



进程与线程

进程是操作系统进行资源分配的最小单元，资源包括CPU、内存、磁盘等IO设备等等，而线程是CPU调度的基本单位。举个简单的例子来帮助理解：我们电脑上同时运行的浏览器和视频播放器是两个不同的进程，进程可能包含多个子任务，这些子任务就是线程，比如视频播放器在播放视频时要同时显示图像、播放声音、显示字幕，这就是三个线程。



郭靖愕然

最近更新 :18-12-13 17:49

简介:每天学一点python，成为更好的自己

作者最新文章

python进阶：用argparse模块为python代码添加命令行参数

用python+OpenCV实现视频中的物体检测

YOLOv3：用python+OpenCV实现物体检测

相关文章

to Data Science? 新接触数据! Github标星超7k！从开始，最简明扼要的...
python fath 基础 Python 高级 Python 高级 数学 数据科学 Python 数据科学 数学
ed Python ed Math for Data Science



编程面试闲聊：进程与线程的关系（附示意...



Class程序猿 04-09



学习Java必须避开的十大致命雷区

读芯术 04-10

数和函数
a,b):
abs(a);
abs(b);
m=a+b;
名称解量sum变量
d;
变量
m(-3,5))
函数
d(-3,5))

Python函数作为函数的参数来使用

米粒教育 04-10



YOLOv3：用python+OpenCV实现...

郭靖愕然 04-08

操作系统通过给不同的线程分配时间片（CPU运行时长）来调度线程，当CPU执行完一个线程的时间片后就会快速切换到下一个线程，时间片很短而且切换速度很快以至于用户根本察觉不到。早期的计算机是单核单线程的，多个线程根据分配的时间片轮流被CPU执行，如今绝大多数计算机的CPU都是多核的，多个线程在操作系统的调度下能够被多个CPU并发执行，程序的执行速度和CPU的利用效率大大提升。绝大多数主流的编程语言都能很好地支持多线程，然而python由于GIL锁无法实现真正的多线程。



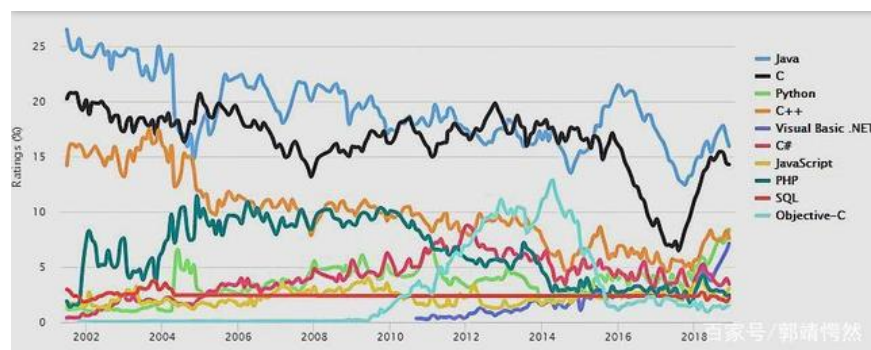
GIL锁

GIL是什么呢？仍然用篮球比赛的例子来帮助理解：把篮球场看作是CPU，一场篮球比赛看作是一个线程，如果只有一个篮球场，多场比赛要排队进行，就是一个简单的单核多线程的程序；如果有多块篮球场，多场比赛同时进行，就是一个简单的多核多线程的程序。然而python有着特别的规定：每场比赛必须要在裁判的监督之下才允许进行，而裁判只有一个。这样不管你有几块篮球场，同一时间只允许有一个场地进行比赛，其它场地都将被闲置，其它比赛都只能等待。

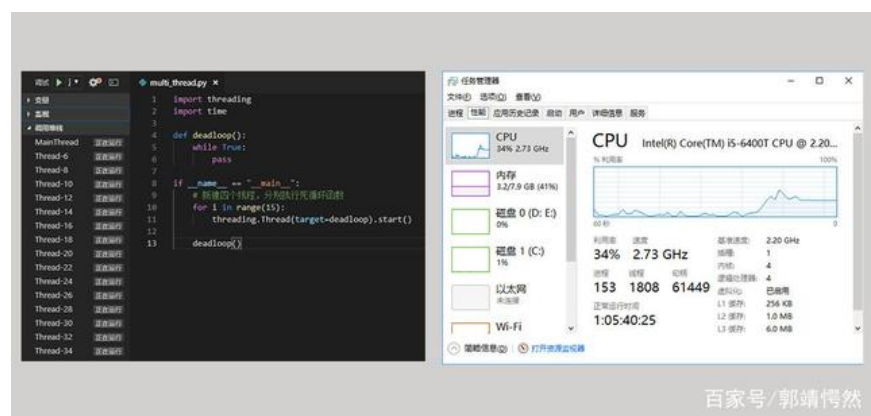


多进程

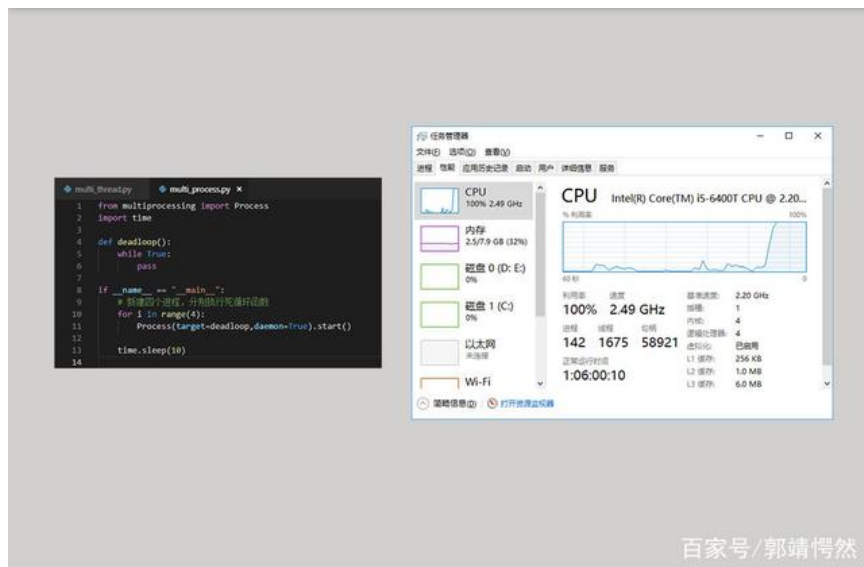
既然不能实现真正的多线程，python为什么能在20年的时间里发展为全世界排名前三（根据2018年12月10日TIOBE编程语言排行）的编程语言呢？其实python有很多方法能解决这一问题，比如使用多进程、C语言扩展、ctypes，本文暂为大家介绍第一种：多进程。



每个进程都包含至少一个线程：主线程，每个主线程可以开启多个子线程，由于GIL锁机制的存在，每个进程里的若干个线程同一时间只能有一个被执行；但是使用多进程就可以保证多个线程被多个CPU同时执行。如果觉得不好理解，请看下面的两段代码。我分别在我的电脑上运行python多线程和多进程的代码，大家对比一下就明白了。我的CPU是i5-6400T，2.20Ghz，4核4线程。



上图左边用python多线程写了一段代码，主线程加上15个子线程总共16个线程（见最左边的线程列表），每个线程都运行了一个死循环，可是CPU的使用率一直稳定在34%，接近三分之二的CPU被闲置。如果类似的代码用C、C++或Java写，CPU使用率会迅速飙升到100%，电脑运行变得非常卡甚至死机。下面将代码改成多进程，看看运行情况。修改后CPU的利用率，一直保持在100%。通过对比我们不难发现python编写多进程能更充分地利用多核CPU的性能，大大提升程序的运行速度。我们用死循环这种极端的例子只是为了对比python多线程与多进程的差别，实际编程中肯定不会这么写。



百家号/郭靖愕然

总结

python多线程和多进程不存在优劣之分，两者都有着各自的应用环境。线程几乎不占资源，系统开销少，切换速度快，而且同一个进程的多个线程之间能很容易地实现数据共享；而创建进程需要为它分配单独的资源，系统开销大，切换速度慢，而且不同进程之间的数据默认是不可共享的。掌握了两者各自的特点，才能在实际编程中根据任务需求采取更加适合的方案。本文从理论上对python多线程和多进程做了粗浅的介绍，希望对初学者有帮助，接下来的文章会详细介绍我在使用上的心得体会和我曾踩过的坑。