

进程

关注者

被浏览

558

190,765

## 线程和进程的区别是什么？

关注问题

写回答

添加评论

分享

邀请回答

...

21 个回答

默认排序



zhonyong

人生就是修行，在享受中，更在苦难中。

844 人赞同了该回答

不请自来。

看见上面几位的回答我真的是醉了。说几句我的理解。

首先来一句概括的总论：**进程和线程都是一个时间段的描述，是CPU工作时间段的描述。**

下面细说背景：

CPU+RAM+各种资源（比如显卡，光驱，键盘，GPS, 等等外设）构成我们的电脑，但是电脑的运行，实际就是CPU和相关寄存器以及RAM之间的事情。

**一个最最基础的事实：**CPU太快，太快，太快了，寄存器仅仅能够追的上他的脚步，RAM和别的挂在各总线上的设备完全是望其项背。那当多个任务要执行的时候怎么办呢？轮流着来?或者谁优先级高谁来？不管怎么样的策略，一句话就是在CPU看来就是轮流着来。

**一个必须知道的事实：**执行一段程序代码，实现一个功能的过程介绍，当得到CPU的时候，相关的资源必须也已经就位，就是显卡啊，GPS啊什么的必须就位，然后CPU开始执行。这里除了CPU以外所有的就构成了这个程序的执行环境，也就是我们所定义的程序上下文。当这个程序执行完了，或者分配给他的CPU执行时间用完了，那它就要被切换出去，等待下一次CPU的临幸。在被切换出去的最后一步工作就是保存程序上下文，因为这个是下次他被CPU临幸的运行环境，必须保存。

**串联起来的事实：**前面讲过在CPU看来所有的任务都是一个一个的轮流执行的，具体的轮流方法就是：**先加载程序A的上下文，然后开始执行A，保存程序A的上下文，调入下一个要执行的程序B的程序上下文，然后开始执行B,保存程序B的上下文。**。。。

===== **重要的东西出现了** =====

进程和线程**就是这样的背景出来的，两个名词不过是对应的CPU时间段的描述，名词就是这样的功能。**

- 进程就是包换上下文切换的程序执行时间总和 = CPU加载上下文+CPU执行+CPU保存上下文**

**线程是什么呢？**

进程的颗粒度太大，每次都要有上下的调入，保存，调出。如果我们把进程比喻为一个运行在电脑上的软件，那么一个软件的执行不可能是一条逻辑执行的，必定有多个分支和多个程序段，就好比要实现程序A，实际分成 a，b，c等多个块组合而成。那么这里具体的执行就可能变成：

程序A得到CPU =》CPU加载上下文，开始执行程序A的a小段，然后执行A的b小段，然后再执行A的c小段，最后CPU保存A的上下文。

这里a，b，c的执行是共享了A的上下文，CPU在执行的时候没有进行上下文切换的。这里的**a，b，c就是线程，也就是说线程是共享了进程的上下文环境，的更为细小的CPU时间段。**

到此全文结束，再一个总结：

**进程和线程都是一个时间段的描述，是CPU工作时间段的描述，不过是颗粒大小不同。**

编辑于 2016-01-12



南理汉子

国电南自电气研发部 助理工程师

175 人赞同了该回答

开个QQ，开了一个进程；开了迅雷，开了一个进程。

在QQ的这个进程里，传输文字开一个线程、传输语音开了一个线程、弹出对话框又开了一个线程。

所以运行某个软件，相当于开了一个进程。在这个软件运行的远

支撑的完成QQ的运行，那么这“多个工作” 分别有一个线程。



下载知乎客户端

与世界分享知识、经验和见解

相关问题

我一个朋友竟然告诉我计算机处理多进程是排队处理，计算机无法同时处理多个进程。我应该怎么说服他? 62 个回答

进程和线程之间有什么根本性的区别，我总感觉线程是进程的进化版？求解答 20 个回答

线程崩溃是否会造成进程崩溃？ 12 个回答

Linux系统中 进程、线程、时间片的关系？ 7 个回答

进程和线程有什么区别？ 6 个回答

相关推荐



职场穿搭礼仪课

李昀

共 18 节课

试听



中国服装设计师之国际化进程

31 人参与



编程小白学 Python

kula 等

229,795 人读过

阅读

刘看山 · 知乎指南 · 知乎协议 · 应用 · 工作

申请开通知乎机构号

侵权举报 · 网上有害信息举报专区

违法和不良信息举报：010-82716601

儿童色情信息举报专区

联系我们 © 2018 知乎

所以一个进程管着多个线程。

通俗的讲：“进程是爹妈，管着众多的线程儿子” ...

编辑于 2016-11-18

▲ 175 ▼ ● 16 条评论 ➦ 分享 ★ 收藏 ♥ 感谢



炊夜冷面  
前端

68 人赞同了该回答

进程是cpu资源分配的最小单位，线程是cpu调度的最小单位。以前进程既是资源分配也是调度的最小单位，后来为了更合理的使用cpu(实际上是cpu性能越来越好)，才将资源分配和调度分开，就有了线程。线程是建立在进程的基础上的一次程序运行单位。

发布于 2016-04-14

▲ 68 ▼ ● 3 条评论 ➦ 分享 ★ 收藏 ♥ 感谢



知乎用户  
一个纯碎的人，唯爱技术与动漫

113 人赞同了该回答

1. 单进程单线程：一个人在一个桌子上吃菜。
2. 单进程多线程：多个人在同一个桌子上一起吃菜。
3. 多进程单线程：多个人每个人在自己的桌子上吃菜。

发布于 2015-10-03

▲ 113 ▼ ● 12 条评论 ➦ 分享 ★ 收藏 ♥ 感谢



王辉  
热情洋溢的程序员 [hui-wang.info](http://hui-wang.info)

26 人赞同了该回答

这个问题，是操作系统里问的最多的问题之一，也是被误解最深的概念之一。Alan Kay说过，好的角度可以提升80分的智商。理解它们的差别，我从资源使用的角度出发。所谓的资源就是计算机里的中央处理器，内存，文件，网络等等。

进程，在一定的环境下，把静态的程序代码运行起来，通过使用不同的资源，来完成一定的任务。比如说，进程的环境包括环境变量，进程所掌控的资源，有中央处理器，有内存，打开的文件，映射的网络端口等等。

这里我把进程对内存的管理稍微展开说一下。一个系统中，有很多进程，它们都会使用内存。为了确保内存不被别人使用，每个进程所能访问的内存都是圈好的。一人一份，谁也不干扰谁。还有内存的分页，虚拟地址我就不深入探讨了。这里给大家想强调的就是，进程需要管理好它的资源。

其中，线程作为进程的一部分，扮演的角色就是怎么利用中央处理器去运行代码。这其中牵扯到的最重要资源的是中央处理器和其中的寄存器，和线程的栈（stack）。这里想强调的是，线程关注的是中央处理器的运行，而不是内存等资源的管理。

当只有一个中央处理器的时候，进程中只需要一个线程就够了。随着多处理器的发展，一个进程中可以有多个线程，来并行的完成任务。比如说，一个web服务器，在接受一个新的请求的时候，可以大动干戈的fork一个子进程去处理这个请求，也可以只在进程内部创建一个新的线程来处理。线程更加轻便一点。线程可以有很多，但他们并不会改变进程对内存（heap）等资源的管理，线程之间会共享这些资源。

总结一下，我上面的解释是通过计算机操作系统的角度出发的。进程和线程不是同一个层面上的概念，线程是进程的一部分，线程主抓中央处理器执行代码的过程，其余的资源的管理和保护由整个进程去完成。

编辑于 2017-10-28

▲ 26 ▼ ● 5 条评论 ➦ 分享 ★ 收藏 ♥ 感谢



知乎用户

15 人赞同了该回答

进程是应用程序的执行实例。比如说，当你双击的Microsoft Word的图标，你就开始运行的Word的进程。线程是执行进程中的路径。另外，一个过程可以包含多个线程。启动Word时，操作系统创建一个进程并开始执行该进程的主线程。

由于一个进程可以由多个线程，线程可以被认为是“轻量级”的过程。因此，一个线程和一个进程之间的本质区别在于，每一个用来完成的工作。线程用于小任务，而进程用于更多的重量级的任务。应用基本执行。

▲ 844 ▼ ● 80 条评论 ★ 收藏 ♥ 感谢



一个线程和进程之间的另一个区别是，在同一进程中的线程共享相同的地址空间，而不同的进程没有。因此线程可以读写同样的数据结构和变量，便于线程之间的通信。相反，进程间通信（IPC）很困难且消耗更多资源。

发布于 2014-11-17

▲ 15 ▼ ● 2 条评论 ➦ 分享 ★ 收藏 ♥ 感谢



酷尔伯特

10 人赞同了该回答

Why:

由于CPU与其他PC资源之间速度的不协调，人们想提高资源利用率，所以人们提出了多任务系统。得益于CPU的计算速度，我们可以“同时”运行多个任务，实质上是多个任务之间轮流使用CPU资源，由于速度超快，给用户的感觉就是连续的。

How:

1) 任务的执行需要依赖各个PC资源，我们可以称为计算机执行的上下文环境。要实现“同时执行”，就需要不断轮换，为了后来继续从当前状态执行下去，计算机需要保存切换前的程序上下文。所以有了进程：用进程去描述程序当前上下文的状态信息-----内存位置、变量值、任务ID.....所以，进程是资源分配的单位。一般来说宏观上可以看做是一个软件的运行，例如一个word文档的打开。

2) 多个任务之间切换因为要保存上下文、调入上下文，一旦多了的时候，还是有一定的时间消耗的。为了进一步提高资源利用率，人们在进程中，引入了线程，线程只是CPU轮流调度的单位，其他上下文信息用所在进程中的。这样上下文切换的耗时就降了下来。同样的，宏观上来可以看做是一个软件中的多个处理功能，例如上述打开word中拼写检查功能、字体加粗.....

So, What:

一般来说，进程是资源的分配单位，线程是CPU在进程内切换的单位，线程属于进程。

发布于 2017-04-20

▲ 10 ▼ ● 3 条评论 ➦ 分享 ★ 收藏 ♥ 感谢



知乎用户  
Android菜鸟 程序猿 创业者

9 人赞同了该回答

1. 定义方面：进程是程序在某个数据集上的一次运行活动；线程是进程中的一个执行路径。
2. 角色方面：在支持线程机制的系统中，进程是系统资源分配的单位，线程是系统调度的单位。
3. 资源共享方面：进程之间不能共享资源，而线程共享所在进程的地址空间和其它资源。同时线程还有自己的栈和栈指针，程序计数器等寄存器。
4. 独立性方面：进程有自己独立的地址空间，而线程没有，线程必须依赖于进程而存在。

详情请移步

[android 线程与进程 区别 联系](#)

关于线程和进程的一篇文章。欢迎大佬拍砖

编辑于 2017-01-22

▲ 9 ▼ ● 添加评论 ➦ 分享 ★ 收藏 ♥ 感谢



后语留香  
计算机潜伏者

6 人赞同了该回答

### 1.7.2 线程

尽管通常我们认为一个进程只有单一的控制流，但是在现代系统中，一个进程实际上可以由多个称为线程的执行单元组成，每个线程都运行在进程的上下文中，并共享同样的代码和全局数据。由于网络服务器中对并行处理的要求，线程成为越来越重要的编程模型，因为多线程之间比多进程之间更容易共享数据，也因为线程一般都比进程更高效。在第 13 章中，你将学习到并行的基本概念，也包括线程化的概念。

### 1.7.1 进程

像 `hello` 这样的程序在现代系统上运行时，操作系统会提供一种假象，就好像系统上只有这个程序在运行。程序看上去独占地使用处理器、主存和 I/O 设备，而处理器看上去就像在不断地一条接一条地执行程序中的指令。该程序的代码和数据就好像是系统存储器中惟一的对象。这些假象是通过进程的概念来实现的，进程是计算机科学中最重要和最成功概念之一。

编辑于 2017-12-26



▲ 6 ▼ 1 条评论 ↗ 分享 ★ 收藏 ♥ 感谢



**吾术家**  
前端工程师

2 人赞同了该回答

我觉得核心点，进程是操作系统分配的，对于开发者来说，线程是你自己的程序分配的，简言之，你控制的了线程，却控制不了进程，进程之间通信比较困难，但并不是不可以，常用的操作时safari来呼起特定的app,但这需要操作系统预留通道

发布于 2017-01-07

▲ 2 ▼ 1 条评论 ↗ 分享 ★ 收藏 ♥ 感谢