腾讯面试题04 进程和线程的区别

进程和线程的区别?

进程和线程的主要差别在于它们是不同的操作系统资源管理方式。进程有独立的地址空间，一个进程崩溃后，在保护模式下不会对其它进程产生影响，而线程只是一个进程中的不同执行路径。线程有自己的堆栈和局部变量，但线程之间没有单独的地址空间，一个线程死掉就等于整个进程死掉，所以多进程的程序要比多线程的程序健壮，但在进程切换时，耗费资源较大，效率要差一些。**但对于一些要求同时进行并且又要共享某些变量的并发操作，只能用线程，不能用进程。**

1) 简而言之,一个程序至少有一个进程,一个进程至少有一个线程.

2) 线程的划分尺度小于进程，使得多线程程序的并发性高。

3) 另外，进程在执行过程中拥有独立的内存单元，而多个线程共享内存，从而极大地提高了程序的运行效率。

4) 线程在执行过程中与进程还是有区别的。每个独立的线程有一个程序运行的入口、顺序执行序列和程序的出口。**但是线程不能够独立执行，**必须依存在应用程序中，由应用程序提供多个线程执行控制。

5) 从逻辑角度来看，多线程的意义在于一个应用程序中，有多个执行部分可以同时执行。但操作系统并没有将多个线程看做多个独立的应用，来实现进程的调度和管理以及资源分配。**这就是进程和线程的重要区别。**

线程与进程总结：

1.进程和线程

1.1 概述:

进程是具有一定独立功能的程序关于某个数据集合上的一次运行活动,进程是系统进行资源分配和调度的一个独立单位.

线程是进程的一个实体,是CPU调度和分派的基本单位,它是比进程更小的能独立运行的基本单位.线程自己基本上不拥有系统资源,只拥有一点在运行中必不可少的资源(如程序计数器,一组寄存器和栈),但是它可与同属一个进程的其他的线程共享进程所拥有的全部资源.

一个线程可以创建和撤销另一个线程;同一个进程中的多个线程之间可以并发执行.

相对进程而言，线程是一个更加接近于执行体的概念，它可以与同进程中的其他线程共享数据，但拥有自己的栈空间，拥有独立的执行序列。

在串行程序基础上引入线程和进程是为了提高程序的并发度，从而提高程序运行效率和响应时间。

1.2 区别:

进程和线程的主要差别在于它们是不同的操作系统资源管理方式。进程有独立的地址空间，一个进程崩溃后，在保护模式下不会对其它进程产生影响，而线程只是一个进程中的不同执行路径。线程有自己的堆栈和局部变量，但线程之间没有单独的地址空间，一个线程死掉就等于整个进程死掉，所以多进程的程序要比多线程的程序健壮，但在进程切换时，耗费资源较大，效率要差一些。**但对于一些要求同时进行并且又要共享某些变量的并发操作，只能用线程，不能用进程。**

1) 简而言之,一个程序至少有一个进程,一个进程至少有一个线程.

2) 线程的划分尺度小于进程，使得多线程程序的并发性高。

3) 另外，进程在执行过程中拥有独立的内存单元，而多个线程共享内存，从而极大地提高了程序的运行效率。

4) 线程在执行过程中与进程还是有区别的。每个独立的线程有一个程序运行的入口、顺序执行序列和程序的出口。**但是线程不能够独立执行，**必须依存在应用程序中，由应用程序提供多个线程执行控制。

5) 从逻辑角度来看，多线程的意义在于一个应用程序中，有多个执行部分可以同时执行。但操作系统并没有将多个线程看做多个独立的应用，来实现进程的调度和管理以及资源分配。**这就是进程和线程的重要区别。**

1.3 优缺点:

线程和进程在使用上各有优缺点：线程执行开销小，但不利于资源的管理和保护；而进程正相反。同时，线程适合于在SMP机器上运行，而进程则可以跨机器迁移。

2.多进程，多线程

2.1 概述:

进程就是一个程序运行的时候被CPU抽象出来的，一个程序运行后被抽象为一个进程，但是线程是从一个进程里面分割出来的，由于CPU处理进程的时候是采用时间片轮转的方式，所以要把一个大个进程给分割成多个线程，例如：网际快车中文件分成100部分 10个线程 文件就被分成了10份来同时下载 1-10 占一个线程 11-20占一个线程,依次类推,线程越多,文件就被分的越多,同时下载 当然速度也就越快

进程是程序在计算机上的一次执行活动。当你运行一个程序，你就启动了一个进程。显然，程序只是一组指令的有序集合，它本身没有任何运行的含义，只是一个静态实体。而进程则不同，它是程序在某个数据集上的执行，是一个动态实体。它因创建而产生，因调度而运行，因等待资源或事件而被处于等待状态，因完成任务而被撤消，反映了一个程序在一定的数据集上运行的全部动态过程。**进程是操作系统分配资源的单位。**在Windows下，进程又被细化为线程，也就是一个进程下有多个能独立运行的更小的单位。**线程(Thread)是进程的一个实体，是CPU调度和分派的基本单位。**线程不能够独立执行，必须依存在应用程序中，由应用程序提供多个线程执行控制。

线程和进程的关系是：线程是属于进程的，线程运行在进程空间内，同一进程所产生的线程共享同一内存空间，当进程退出时该进程所产生的线程都会被强制退出并清除。线程可与属于同一进程的其它线程共享进程所拥有的全部资源，但是其本身基本上不拥有系统资源，只拥有一点在运行中必不可少的信息(如程序计数器、一组寄存器和栈)。

在同一个时间里，同一个计算机系统中如果允许两个或两个以上的进程处于运行状态，这便是多任务。现代的操作系统几乎都是多任务操作系统，能够同时管理多个进程的运行。 多任务带来的好处是明显的，比如你可以边听mp3边上网，与此同时甚至可以将下载的文档打印出来，而这些任务之间丝毫不会相互干扰。那么这里就涉及到并行的问题，俗话说，一心不能二用，这对计算机也一样，原则上一个CPU只能分配给一个进程，以便运行这个进程。我们通常使用的计算机中只有一个CPU，也就是说只有一颗心，要让它一心多用，同时运行多个进程，就必须使用并发技术。实现并发技术相当复杂，**最容易理解的是"时间片轮转进程调度算法"，**它的思想简单介绍如下：在操作系统的管理下，所有正在运行的进程轮流使用CPU，每个进程允许占用CPU的时间非常短(比如10毫秒)，这样用户根本感觉不出来CPU是在轮流为多个进程服务，就好象所有的进程都在不间断地运行一样。但实际上在任何一个时间内有且仅有一个进程占有CPU。

如果一台计算机有多个CPU，情况就不同了，如果进程数小于CPU数，则不同的进程可以分配给不同的CPU来运行，这样，多个进程就是真正同时运行的，这便是并行。但如果进程数大于CPU数，则仍然需要使用并发技术。

在**Windows中，进行CPU分配是以线程为单位的，**一个进程可能由多个线程组成，这时情况更加复杂，但简单地说，有如下关系：

总线程数<= CPU数量：并行运行

总线程数> CPU数量：并发运行

并行运行的效率显然高于并发运行，所以在多CPU的计算机中，多任务的效率比较高。但是，如果在多CPU计算机中只运行一个进程(线程)，就不能发挥多CPU的优势。

多任务操作系统(如Windows)的基本原理是:操作系统将CPU的时间片分配给多个线程,每个线程在操作系统指定的时间片内完成(注意,这里的多个线程是分属于不同进程的).操作系统不断的从一个线程的执行切换到另一个线程的执行,如此往复,宏观上看来,就好像是多个线程在一起执行.由于这多个线程分属于不同的进程,因此在我们看来,就好像是多个进程在同时执行,这样就实现了多任务.

2.2 分类

根据进程与线程的设置，操作系统大致分为如下类型：

(1) 单进程、单线程，MS-DOS大致是这种操作系统；

(2) 多进程、单线程，多数UNIX(及类UNIX的LINUX)是这种操作系统；

(3) 多进程、多线程，Win32(Windows NT/2000/XP等)、Solaris 2.x和OS/2都是这种操作系统；

(4) 单进程、多线程，VxWorks是这种操作系统。

2.3 引入线程带来的主要好处：

(1) 在进程内创建、终止线程比创建、终止进程要快；

(2) 同一进程内的线程间切换比进程间的切换要快,尤其是用户级线程间的切换。

进程（process）是指在系统中正在运行的一个应用程序，是**系统资源分配的基本单位**，在内存中有其完备的数据空间和代码空间，拥有完整的虚拟空间地址。一个进程所拥有的数据和变量只属于它自己。

线程（thread）是进程内相对独立的可执行单元，所以也被称为轻量进程（lightweight processes）；是操作系统进行**任务调度的基本单元**。它与父进程的其它线程**共享该进程所拥有的全部代码空间和全局变量**，但**拥有独立的堆栈（即局部变量对于线程来说是私有的）。**

进程和线程都具有就绪、阻塞和运行三种基本状态。

联系

一个进程至少拥有一个线程——主线程，也可以拥有多个线程；一个线程必须有一个父进程。多个进程可以并发执行；一个线程可以创建和撤销另一个线程；同一个进程中的多个线程之间可以并发执行。

区别

系统开销：在创建或撤消进程时，由于系统都要为之分配和回收资源，导致系统的开销明显大于创建或撤消线程时的开销

资源管理：进程有独立的地址空间，一**个进程崩溃后，在保护模式下不会对其它进程产生影响，而线程只是一个进程中的不同执行路径。线程有自己的堆栈和局部变量，但线程之间没有单独的地址空间，一个线程死掉就等于整个进程死掉**，所以多进程的程序要比多线程的程序健壮，但在进程切换时，耗费资源较大，效率要差一些。但对于一些要求同时进行并且又要共享某些变量的并发操作，只能用线程，不能用进程。

通信方式：**进程间通信主要包括管道、系统IPC(包括消息队列,信号量,共享存储)、SOCKET，具体说明参考linux进程间通信方式。进程间通信其实是指分属于不同进程的线程之间的通讯，**所以进程间的通信方法**同样适用于线程间的通信。但对应归于同一进程的不同线程来说，使用全局变量**进行通信效率会更高。