http://blog.csdn.net/u012134199/article/details/46290465

什么是进程？  
当一个程序开始运行时，它就是一个进程，进程包括运行中的程序和程序所使用到的内存和系统资源。  
而一个进程又是由多个线程所组成的。  
  
  
什么是线程？  
线程是程序中的一个执行流，每个线程都有自己的专有寄存器(栈指针、程序计数器等)，但代码区是共享的，  
即不同的线程可以执行同样的函数。  
  
  
什么是多线程？  
多线程是指程序中包含多个执行流，即在一个程序中可以同时运行多个不同的线程来执行不同的任务，  
也就是说允许单个程序创建多个并行执行的线程来完成各自的任务。  
  
  
多线程的好处：  
可以提高CPU的利用率。在多线程程序中，一个线程必须等待的时候，CPU可以运行其它的线程而不是等待，  
这样就大大提高了程序的效率。   
  
  
多线程的不利方面：  
线程也是程序，所以线程需要占用内存，线程越多占用内存也越多；   
多线程需要协调和管理，所以需要CPU时间跟踪线程；   
线程之间对共享资源的访问会相互影响，必须解决竞用共享资源的问题；  
线程太多会导致控制太复杂，最终可能造成很多Bug；  
  
  
多线程与单线程的区别  
生活举例  
你早上上班，正要打卡的时候，手机响了。。你如果先接了电话，等接完了，在打卡，就是单线程。  
如果你一手接电话，一手打卡。就是多线程。  
2件事的结果是一样的。。你接了电话且打了卡。

多线程处理的优点  
  
同步应用程序的开发比较容易，但由于需要在上一个任务完成后才能开始新的任务，所以其效率通常比多线程应用程序低。如果完成同步任务所用的时间比预计时间长，应用程序可能会不响应。多线程处理可以同时运行多个过程。例如，文字处理器应用程序在您处理文档的同时，可以检查拼写（作为单独的任务）。由于多线程应用程序将程序划分成独立的任务，因此可以在以下方面显著提高性能：   
多线程技术使程序的响应速度更快，因为用户界面可以在进行其他工作的同时一直处于活动状态。   
当前没有进行处理的任务可以将处理器时间让给其他任务。   
占用大量处理时间的任务可以定期将处理器时间让给其他任务。   
可以随时停止任务。   
可以分别设置各个任务的优先级以优化性能。   
  
是否需要创建多线程应用程序取决于多个因素。在以下情况下，最适合采用多线程处理：  
耗时或大量占用处理器的任务阻塞用户界面操作。   
各个任务必须等待外部资源（如远程文件或 INTERNET 连接）。   
  
例如，用于跟踪 WEB 页上的链接并下载满足特定条件的文件的 INTERNET 应用程序“ROBOT”。这种应用程序可以依次同步下载各个文件，也可以使用多线程同时下载多个文件。多线程方法比同步方法的效率高很多，因为即使在某些线程中远程 WEB 服务器的响应非常慢，也可以下载文件。  
  
下面是多线程的例子  
还在DOS时代，人们就在寻求一种多任务的实现。于是出现了TSR类型的后台驻留程序，比较有代表性的有SIDE KICK、VSAFE等优秀的TSR程序，这类程序的出现和应用确实给用户使用计算机带来了极大的方便，比如SIDE KICK，们编程可以在不用进编辑程序的状态下，一边编辑源程序，一边编译运行，非常方便。但是，DOS单任务操作系统的致命缺陷注定了在DOS下不可能开发出真正的多任务程序。进入WINDOWS3.1时代，这种情况依然没有根本的改变，一次应用只能做一件事。比如数据库查询，除非应用编得很好，在查询期间整个系统将不响应用户的输入。  
 进入了WINDOWS NT和WINDOWS 9X时代，情况就有了彻底的改观，操作系统从真正意义上实现了多任务（严格地说，WIN9X还算不上）。一个应用程序，在需要的时候可以有许多个执行线程，每个线程就是一个小的执行程序，操作系统自动使各个线程共享CPU资源，确保任一线程都不能使系统死锁。这样，在编程的时候，可以把费时间的任务移到后台，在前台用另一个线程接受用户的输入。对那些对实时性要求比较高的编程任务，如网络客户服务、串行通信等应用时，多线程的实现无疑大大地增强了程序的可用性和稳固性。

=====================================================================================

坏处：增加了调度和管理的开销，带来了一些不确定性，需要复杂的同步机制，避免死锁等等。  
好处：一定程度上提高响应速度，在多核的情况下还是更能充分利用CPU资源的。

=====================================================================================

单线程的也就是程序执行时，所跑的程序路径（处理的东西）是连续顺序下来的，必须前面的处理好，后面的才会执行到。     
       多线程嘛，举个例子也就是说程序可以同时执行2个以上相同类似的操作，比如一些搜索代理或者群发email的多线程软件，由于操作一次需要网络的返回信息   花的时间比较长，而对cpu来说却是空闲的，如果是一个一个顺序执行，那么搜索几千个IP就会花上好久好久。   而如果用多线程就可以在等待期间   加入其他的搜索，然后等待，这样可以提高效率。不过多线程和多进程公用一些资源时要考虑的问题好像也是一样的，对于一些公共资源或者公共变量的访问和修改时要注意特别的，需要一些锁定什么的，还有顺序问题的考虑。    
       多线程编程的目的,就是"最大限度地利用CPU资源",当某一线程的处理不需要占用CPU而只和I/O,OEMBIOS等资源打交道时,让需要占用CPU资源的其它线程有机会获得CPU资源。每个程序执行时都会产生一个进程，而每一个进程至少要有一个主线程。这个线程其实是进程执行的一条线索，除了主线程外你还可以给进程增加其它的线程，也即增加其它的执行线索，由此在某种程度上可以看成是给一个应用程序增加了多任务功能。当程序运行后，您可以根据各种条件挂起或运行这些线程，尤其在多CPU的环境中，这些线程是并发运行的。多线程就是在一个进程内有多个线程。从而使一个应用程序有了多任务的功能。多进程技术也可以实现这一点，但是创建进程的高消耗（每个进程都有独立的数据和代码空间），进程之间通信的不方便（消息机制），进程切换的时间太长，这些导致了多线程的提出，对于单CPU来说（没有开启超线程），在同一时间只能执行一个线程，所以如果想实现多任务，那么就只能每个进程或线程获得一个时间片，在某个时间片内，只能一个线程执行，然后按照某种策略换其他线程执行。由于时间片很短，这样给用户的感觉是同时有好多线程在执行。但是线程切换是有代价的，因此如果采用多进程，那么就需要将线程所隶属的该进程所需要的内存进行切换，这时间代价是很多的。而线程切换代价就很少，线程是可以共享内存的。所以采用多线程在切换上花费的比多进程少得多。但是，线程切换还是需要时间消耗的，所以采用一个拥有两个线程的进程执行所需要的时间比一个线程的进程执行两次所需要的时间要多一些。即采用多线程不会提高程序的执行速度，反而会降低速度，但是对于用户来说，可以减少用户的响应时间。上述结果只是针对单CPU，如果对于多CPU或者CPU采用超线程技术的话，采用多线程技术还是会提高程序的执行速度的。因为单线程只会映射到一个CPU上，而多线程会映射到多个CPU上，超线程技术本质是多线程硬件化，所以也会加快程序的执行速度。

====================================================================================

如果线程出现死锁，唯一能证明的就是应用程序有问题，这并不是线程的缺点。  
  
线程相对于进程的优点：  
1、开销小  
2、资源共享性好。  
  
线程相对于进程的缺点：  
1、共享资源需要耗费一定的锁资源，同步相对复杂。  
2、一个线程崩溃可能导致整个进程崩溃，这个当然是自己的应用程序有问题

====================================================================================

CPU是以时间片的方式为进程分配CUP处理时间的，当一个进程以同步的方式去完成几件事情时，此进程必须完成了第一件事情以后再做第二件事，如此按顺序地向CPU请求完成要做的事情。在此单线程的工作模式下，如果把CUP看作是一共有100个时间片的话，CPU可能一直都只是花了其中的10个时间片来处理当前进程所要做的事情，只是用到了CPU的10%的时间片，而其他时间都白白浪费了，当然，实际上CPU的工作模式还是做完一件事以后再去做另一件事，只是CUP的处理速度非常快，很快就处理完成所请求的情事。

    为了提高CPU的使用率，采用多线程的方式去同时完成几件事情而互不干扰，如当前进程要完成三件事情1、2、3，那么CPU会分别用10%的时间来同时处理这3件事情，从而让CPU的使用率达到了30%，大大地提高了CPU的利用率。多线程的好处在处理一些特殊的场合其优势尤其明显。比如下载文件，你要一边下载一边显示进度一边保存，在这种情况下，如果没有用多线程的话，没有意外的话一般都会把主线程阻塞，比如进度条的进度根本没有随着已下载的量而变化，堪至是整个窗体都动不了，用多线程就可以很好地解决这个问题。

    这里有一个生活实例可能更好地去理解多线程：回去看你女朋友做饭，正常的话她都会把洗好的菜（肉）先放到锅里煮，然后一边洗别的菜或处理别的事情，如：洗碗、收拾桌台准备开饭，人还是一个人，但她同时做几件事情，这样就可以大大地提高效率。总的一句话就是：CPU还是要花同样多的时间去完成所有的事情，但多线程可以让CPU掺插地同时做多件事情，在视觉上让用户觉得计算机在同时帮他处理多件事情，更好地改善用户体验。

     了解了多线程的好处以后，就要了解应该在什么样的情况下使用多线程技术。因为并不是说所有情况下用多线程都是好事，因为多线程的情况下，CPU还要花时间去维护，CPU处理各线程的请求时在线程间的切换也要花时间，所以一般情况下是可以不用多线程的，用了有时反而会得不偿失。大多情况下，要用到多线程的主要是需要处理大量的IO操作时或处理的情况需要花大量的时间等等，比如：读写文件、视频图像的采集、处理、显示、保存等。