1. 和进程相比，它是一种非常"节俭"的多任务操作方式。对进程而言，创建一个新的进程必须分配给它独立的地址空间，建立众多的数据表来维护它的代码段、堆栈段和数据段，这是一种"昂贵"的多任务工作方式。而运行于一个进程中的多个线程，它们彼此之间**使用相同的地址空间，共享大部分数据**，启动一个**线程所花费的空间远远小于启动一个进程所花费的空间**，而且，线程间彼此切换所需的时间也远远小于进程间切换所需要的时间；  
   2.与进程相比，线程间的通信机制更方便。对不同进程来说，它们具有独立的数据空间，要进行数据的传递只能通过通信的方式进行，这种方式不仅费时，而且很不方便。线程则不然，由于同一进程下的线程之间共享数据空间，所以一个线程的数据可以直接为其它线程所用，这不仅快捷，而且方便；  
   3.**提高应用程序响应**。使用多线程技术，将耗时长的操作（time consuming）置于一个新的线程；  
   4.**使多CPU系统更加有效**。操作系统会保证当线程数不大于CPU数目时，**不同的线程运行于不同的CPU上；**5.改善程序结构。**一个既长又复杂的进程可以考虑分为多个线程，成为几个独立或半独立的运行部分，这样的程序会利于理解和修改。**

何时使用多线程技术,何时避免用它,是我们需要掌握的重要课题。多线程技术是一把双刃剑,在使用时需要充分考虑它的优缺点。  
多线程处理可以同时运行多个线程。由于多线程应用程序将程序划分成多个独立的任务,因此可以在以下方面显著提高性能:   
**(1)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它工作的同时一直处于活动状态;**

**(2)当前没有进行处理的任务时可以将处理器时间让给其它任务;**

**(3)占用大量处理时间的任务可以定期将处理器时间让给其它任务;**

**(4)可以随时停止任务;**

**(5)可以分别设置各个任务的优先级以优化性能。**

**是否需要创建多个线程取决于各种因素。在以下情况下,最适合采用多线程处理:   
(1)耗时或大量占用处理器的任务阻塞用户界面操作;**

**(2)各个任务必须等待外部资源 (如远程文件或 Internet连接)。**

同样的 ,多线程也存在许多缺点 ,在考虑多线程时需要进行充分的考虑。多线程的主要缺点包括:   
(1)等候使用共享资源时造成程序的运行速度变慢。这些共享资源主要是独占性的资源 ,如打印机等。

(2)对线程进行管理要求额外的 CPU开销。线程的使用会给系统带来上下文切换的额外负担。当这种负担超过一定程度时,多线程的特点主要表现在其缺点上,比如用独立的线程来更新数组内每个元素。

(3)线程的死锁。即较长时间的等待或资源竞争以及死锁等多线程症状。

(4)对公有变量的同时读或写。当多个线程需要对公有变量进行写操作时,后一个线程往往会修改掉前一个线程存放的数据,从而使前一个线程的参数被修改;另外 ,当公用变量的读写操作是非原子性时,在不同的机器上,中断时间的不确定性,会导致数据在一个线程内的操作产生错误,从而产生莫名其妙的错误,而这种错误是程序员无法预知的。