进程（Process）和线程（Thread）都是操作系统中的基本概念，它们之间有一些优劣和差异。

# 进程

进程是程序执行时的一个实例，是系统进行资源分配的基本单位。所有与该进程有关的资源，都被记录在进程控制块(PCB)中。以表示该进程拥有这些资源或正在使用它们。另外，进程也是抢占处理机的调度单位，它拥有一个完整的虚拟地址空间。当进程发生调度时，不同的进程拥有不同的虚拟地址空间，而同一进程内的不同线程共享同一地址空间。

# 线程

线程，有时也被称为轻量级进程，是程序执行流的最小单元，是进程中的一个实体，是被系统独立调度和分派的基本单位。与进程不同，线程与资源分配无关，线程自己不拥有系统资源，它属于某一个进程，并与进程内的其他线程一起共享进程的资源。线程只由相关堆栈（系统栈或用户栈）寄存器和线程控制表TCB组成。

# 联系

通常在一个进程中可以包含若干个线程，它们可以利用进程所拥有的资源。但是，一个线程只属于一个进程。进程间相互独立，同一进程的各线程间共享。某进程内的线程在其它进程不可见。而且需要注意的是，线程不是一个可执行的实体。

# 区别

进行和线程之间的差异可以从下面几个方面来阐述：

调度 ：在引入线程的操作系统中，线程是调度和分配的基本单位 ，进程是资源拥有的基本单位 。把传统进程的两个属性分开，线程便能轻装运行，从而可 显著地提高系统的并发程度 。在同一进程中，线程的切换不会引起进程的切换；在由一个进程中的线程切换到另一个进程中的线程时，才会引起进程的切换。

并发性 ：在引入线程的操作系统中，不仅进程之间可以并发执行，而且在一个进程中的多个线程之间亦可并发执行，因而使操作系统具有更好的并发性，从而能 更有效地使用系统资源和提高系统吞吐量。

拥有资源 ：不论是传统的操作系统，还是设有线程的操作系统，进程都是拥有资源的一个独立 单位，它可以拥有自己的资源。一般地说，线程自己不拥有系统资源（只有一些必不可少的资源，但它可以访问其隶属进程的资源。

系统开销：由于在创建或撤消进程时，系统都要为之分配或回收资源，因此，操作系统所付出的开销将显著地大于在创建或撤消线程时的开销。进程切换的开销也远大于线程切换的开销。

通信：进程间通信IPC，线程间可以直接读写进程数据段（如全局变量）来进行通信——需要进程同步和互斥手段的辅助，以保证数据的一致性，因此共享简单。但是线程的数据同步要比进程略复杂。

# 总结

从上面的分析可以看到，似乎线程有很多优势，比如，数据共享效率高，可应对并发操作，有效利用等待时间等等，但是多线程的编程比多进程要复杂，同时，多进程的可靠性较好，因为进程间不会相互影响。实际情况还是需要自己分析拿捏的。但是一般来说，实际应用中常常采用“进程+线程”结合的方式，而不是非此即彼，因为它们两者没有绝对的好与不好，而是适合于不同场景。