## ****多线程和异步操作的异同****

多线程和异步操作两者都可以达到避免调用线程阻塞的目的，从而提高软件的可响应性。甚至有些时候我们就认为多线程和异步操作是等同的概念。但是，多线程和异步操作还是有一些区别的。而这些区别造成了使用多线程和异步操作的时机的区别

## 异步操作的本质

所有的程序最终都会由计算机硬件来执行，所以为了更好的理解异步操作的本质，我们有必要了解一下它的硬件基础。

熟悉电脑硬件的朋友肯定对DMA这个词不陌生，硬盘、光驱的技术规格中都有明确DMA的模式指标，其实网卡、声卡、显卡也是有DMA功能的。

1. DMA就是直接内存访问的意思，也就是说，拥有DMA功能的硬件在和内存进行数据交换的时候可以不消耗CPU资源。
2. 只要CPU在发起数据传输时发送一个指令，硬件就开始自己和内存交换数据，在传输完成之后硬件会触发一个中断来通知操作完成。这些无须消耗CPU时间的I/O操作正是异步操作的硬件基础。
3. 所以即使在DOS这样的单进程（而且无线程概念）系统中也同样可以发起异步的DMA操作

　线程的本质

　　线程不是一个计算机硬件的功能，而是操作系统提供的一种逻辑功能，线程本质上是进程中一段并发运行的代码，所以线程需要操作系统投入CPU资源来运行和调度。

**异步操作的优缺点**

　　因为异步操作无须额外的线程负担，并且使用回调的方式进行处理，在设计良好的情况下，处理函数可以不必使用共享变量（即使无法完全不用，最起码可以减少共享变量的数量），减少了死锁的可能。当然异步操作也并非完美无暇。编写异步操作的复杂程度较高，程序主要使用回调方式进行处理，与普通人的思维方式有些初入，而且难以调试。

**多线程的优缺点**  
　　多线程的优点很明显，线程中的处理程序依然是顺序执行，符合普通人的思维习惯，所以编程简单。但是多线程的缺点也同样明显，线程的使用（滥用）会给系统带来上下文切换的额外负担。并且线程间的共享变量可能造成死锁的出现。

**适用范围**

　　在了解了线程与异步操作各自的优缺点之后，我们可以来探讨一下线程和异步的合理用途。我认为：当需要执行I/O操作时，使用异步操作比使用线程+同步I/O操作更合适。I/O操作不仅包括了直接的文件、网络的读写，还包括数据库操作、Web Service、HttpRequest以及.Net Remoting等跨进程的调用。  
　　而线程的适用范围则是那种需要长时间CPU运算的场合，例如耗时较长的图形处理和算法执行。但是往往由于使用线程编程的简单和符合习惯，所以很多朋友往往会使用线程来执行耗时较长的I/O操作。这样在只有少数几个并发操作的时候还无伤大雅，如果需要处理大量的并发操作时就不合适了。

## 线程同步与异步区别

线程同步是多个线程同时访问同一资源，等待资源访问结束，浪费时间，效率低    线程异步：访问资源时在空闲等待时同时访问其他资源，实现多线程机制

异步处理就是,你现在问我问题,我可以不回答你,等我用时间了再处理你这个问题.同步不就反之了，同步信息被立即处理 -- 直到信息处理完成才返回消息句柄；异步信息收到后将在后台处理一段时间 -- 而早在信息处理结束前就返回消息句柄