一：几个基本概念

异步与多线程

1）基本概念

1. 并发：在操作系统中，是指一个时间段中有几个程序都处于已启动运行到运行完毕之间，且这几个程序都是在同一个处理机上运行。其中两种并发关系分别是同步和互斥

2. 互斥：进程间相互排斥的使用临界资源的现象，就叫互斥。

3. 同步：进程之间的关系不是相互排斥临界资源的关系，而是相互依赖的关系。进一步的说明：就是前一个进程的输出作为后一个进程的输入，当第一个进程没有输出时第二个进程必须等待。具有同步关系的一组并发进程相互发送的信息称为消息或事件。

其中并发又有伪并发和真并发，伪并发是指单核处理器的并发，真并发是指多核处理器的并发。

4. 并行：在单处理器中多道程序设计系统中，进程被交替执行，表现出一种并发的外部特种；在多处理器系统中，进程不仅可以交替执行，而且可以重叠执行。在多处理器上的程序才可实现并行处理。从而可知，并行是针对多处理器而言的。并行是同时发生的多个并发事件，具有并发的含义，但并发不一定并行，也亦是说并发事件之间不一定要同一时刻发生。

5. 多线程：多线程是程序设计的逻辑层概念，它是进程中并发运行的一段代码。多线程可以实现线程间的切换执行。

6. 异步：异步和同步 是相对的，同步就是顺序执行，执行完一个再执行下一个，需要等待、协调运行。异步就是彼此独立,在等待某事件的过程中继续做自己的事，不需要等待这一事件完成后再工作。线程就是实现异步的一个方式。异步是让调用方法的主线程不需要同步等待另一线程的完成，从而可以让主线程干其它的事情。

异步和多线程并不是一个同等关系,异步是最终目的,多线程只是我们实现异步的一种手段。异步是当一个调用请求发送给被调用者,而调用者不用等待其结果的返回而可以做其它的事情。实现异步可以采用多线程技术或则交给另外的进程来处理。

异步和同步的区别， 在io等待的时候，同步不会切走，浪费了时间。

多线程的好处，比较容易的实现了 异步切换的思想， 因为异步的程序很难写的。多线程本身程还是以同步完成，但是应该说比效率是比不上异步的。 而且多线很容易写， 相对效率也高。

2）深层次理解

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。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。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。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。