异步：一种通讯方式，对设备需求简单。我们的[PC](https://baike.baidu.com/item/PC)机提供的标准[通信接口](https://baike.baidu.com/item/%E9%80%9A%E4%BF%A1%E6%8E%A5%E5%8F%A3)都是异步的。

异步双方不需要共同的时钟，也就是接收方不知道发送方什么时候发送，所以在发送的信息中就要有提示接收方开始接收的信息，如开始位，同时在结束时有停止位。

异步的另外一种含义是**计算机多线程的异步处理**。与同步处理相对，异步处理**不用阻塞当前**[**线程**](https://baike.baidu.com/item/%E7%BA%BF%E7%A8%8B)**来等待处理完成**，而是允许后续操作，直至其它线程将处理完成，并回调通知此线程。

**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。

**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**。

**上下文切换**，有时也称做进程切换或任务切换，是指[CPU](https://baike.baidu.com/item/CPU) 从一个[进程](https://baike.baidu.com/item/%E8%BF%9B%E7%A8%8B)或线程切换到另一个进程或线程。

**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。

**并发**

并发，在[操作系统](https://baike.baidu.com/item/%E6%93%8D%E4%BD%9C%E7%B3%BB%E7%BB%9F)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](https://baike.baidu.com/item/%E5%A4%84%E7%90%86%E6%9C%BA)上运行，但任一个时刻点上只有一个程序在处理机上运行。

**并发**当有多个线程在操作时,如果系统只有一个CPU,则它根本不可能真正同时进行一个以上的线程，它只能把CPU运行时间划分成若干个时间段,再将时间 段分配给各个线程执行，在一个时间段的线程代码运行时，其它线程处于挂起状。.这种方式我们称之为并发(Concurrent)。

**并行：**当系统有一个以上CPU时,则线程的操作有可能非并发。当一个CPU执行一个线程时，另一个CPU可以执行另一个线程，两个线程互不抢占CPU资源，可以同时进行，这种方式我们称之为并行(Parallel)。

DMA（直接存储器访问）

DMA(Direct Memory Access，直接内存存取) 是所有现代电脑的重要特色，它允许不同速度的硬件装置来沟通，而不需要依赖于 CPU 的大量中断负载。否则，CPU 需要从来源把每一片段的资料复制到[暂存器](https://baike.baidu.com/item/%E6%9A%82%E5%AD%98%E5%99%A8)，然后把它们再次写回到新的地方。在这个时间中，CPU 对于其他的工作来说就无法使用。

DMA 传输将数据从一个地址空间复制到另外一个地址空间。当CPU 初始化这个传输动作，传输动作本身是由 DMA 控制器来实行和完成。典型的例子就是移动一个外部内存的区块到芯片内部更快的内存区。像是这样的操作并没有让[处理器](https://baike.baidu.com/item/%E5%A4%84%E7%90%86%E5%99%A8)工作拖延，反而可以被重新排程去处理其他的工作。DMA 传输对于高效能 嵌入式系统算法和网络是很重要的。