**1. 概念**

1. 进程  
   进程是具有一定独立功能的程序关于某个数据集合上的一次运行活动,进程是系统进行资源分配和调度的一个独立单位。每个进程都有自己的独立内存空间，不同进程通过进程间通信来通信。由于进程比较重量，占据独立的内存，所以上下文进程间的切换开销（栈、寄存器、虚拟内存、文件句柄等）比较大，但相对比较稳定安全。
2. 线程  
   线程是进程的一个实体,是CPU调度和分派的基本单位,它是比进程更小的能独立运行的基本单位.线程自己基本上不拥有系统资源,只拥有一点在运行中必不可少的资源(如程序计数器,一组寄存器和栈),但是它可与同属一个进程的其他的线程共享进程所拥有的全部资源。线程间通信主要通过共享内存，上下文切换很快，资源开销较少，但相比进程不够稳定容易丢失数据。
3. 协程  
   协程是一种用户态的轻量级线程，协程的调度完全由用户控制。协程拥有自己的寄存器上下文和栈。协程调度切换时，将寄存器上下文和栈保存到其他地方，在切回来的时候，恢复先前保存的寄存器上下文和栈，直接操作栈则基本没有内核切换的开销，可以不加锁的访问全局变量，所以上下文的切换非常快。

**2. 区别**

1. 进程多与线程比较  
   线程是指进程内的一个执行单元,也是进程内的可调度实体。线程与进程的区别:
   1. 地址空间:线程是进程内的一个执行单元，进程内至少有一个线程，它们共享进程的地址空间，而进程有自己独立的地址空间
   2. 资源拥有:进程是资源分配和拥有的单位,同一个进程内的线程共享进程的资源
   3. 线程是处理器调度的基本单位,但进程不是
   4. 二者均可并发执行
   5. 每个独立的线程有一个程序运行的入口、顺序执行序列和程序的出口，但是线程不能够独立执行，必须依存在应用程序中，由应用程序提供多个线程执行控制
2. 协程多与线程进行比较
   1. 一个线程可以多个协程，一个进程也可以单独拥有多个协程。
   2. 线程进程都是同步机制，而协程则是异步
   3. 协程能保留上一次调用时的状态，每次过程重入时，就相当于进入上一次调用的状态