1. **进程和线程有什么区别？**

事实上，引入线程主要是为了提高系统的执行效率，减少处理机的空转时间和调度切换（保护现场信息）的时间，以及便于系统管理。

　　a. 进程是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。与进程相对应，线程与资源分配无关，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。它是CPU调度的基本单位。线程只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)。寄存器可被用来存储线程内的局部变量，但不能存储其他线程的相关变量。

　　b. 发生进程切换与发生线程切换时相比较，进程切换时涉及到有关资源指针的保存以及地址空间的变化等问题；线程切换时，由于同不进程内的线程共享资源和地址空间，将不涉及资源信息的保存和地址变化问题，从而减少了操作系统的开销时间。而且，进程的调度与切换都是由操作系统内核完成，而线程则既可由操作系统内核完成，也可由用户程序进行。

　　c.多线程之间的通信比较方便。同一进程下的线程共享数据（比如全局变量，静态变量），通过这些数据来通信不仅快捷而且方便，当然如何处理好这些访问的同步与互斥正是编写多线程程序的难点。而多进程之间的通信只能通过进程通信的方式进行，比较费时费力。

　　d.由b，可以轻易地得到结论：多进程比多线程程序要健壮。一个线程死掉整个进程就死掉了，但是在保护模式下，一个进程死掉对另一个进程没有直接影响。

（注意：实模式：是指直接访问物理内存，指令中的地址值是物理内存地址。保护模式：是指指令中的地址不再是物理地址，而是虚拟地址，要经过内存管理单元(MMU)的处理转换成物理地址。保护模式同实模式的根本区别是进程内存受保护与否）

　　e.线程的执行与进程是有区别的。每个独立的线程有有自己的一个程序入口，顺序执行序列和程序的出口，但是线程不能独立执行，必须依附与程序之中，由应用程序提供多个线程的并发控制。

**2. 什么是线程安全？（2**012年5月百度实习生面试）

　　如果多线程的程序运行结果是可预期的，而且与单线程的程序运行结果一样，那么说明是“线程安全”的。

**3. 线程的分类**

用户级线程：管理过程全部由用户程序完成，操作系统内核心只对进程进行管理。

系统级线程（核心级线程）：由操作系统内核进行管理。操作系统内核给应用程序提供相应的系统调用和应用程序接口API，以使用户程序可以创建、执行、撤消线程。

**4.什么是线程的同步与互斥？**

**互斥**：指在某一时刻指允许一个进程运行其中的程序片，具有排他性和唯一性。

对于线程A和线程B来讲，在同一时刻，只允许一个线程对临界资源进行操作，即当A进入临界区对资源操作时，B就必须等待；当A执行完，退出临界区后，B才能对临界资源进行操作。

**同步**：指的是在互斥的基础上，实现进程之间的有序访问。假设现有线程A和线程B，线程A需要往缓冲区写数据，线程B需要从缓冲区读数据，但他们之间存在一种制约关系，即当线程A写的时候，B不能来拿数据；B在拿数据的时候A不能往缓冲区写，也就是说，只有当A写完数据（或B取走数据），B才能来读数据（或A才能往里写数据）。这种关系就是一种线程的同步关系。

**5.多线程同步与互斥有几种实现方法？都是什么？**

　　关键段（CS:critical section临界区）、事件（Event）、互斥量（Mutex）、信号量（semaphores），需要注意的是，关键段是效率最高的，因为基本不需要其他的开销。互斥量，事件，信号量都是内核对象，可以跨进程使用，内核对象涉及到用户态和内核态的切换，开销较大，另外，关键段、互斥量具有线程所有权的概念，因此只可以用于线程之间互斥，而不能用到同步中。只有互斥量能完美解决进程意外终止所造成的“遗弃问题”。

**6.多线程同步和互斥有何异同，在什么情况下分别使用他们？举例说明**

　　所谓同步，表示有先有后，比较正式的解释是“线程同步是指线程之间所具有的一种制约关系，一个线程的执行依赖另一个线程的消息，当它没有得到另一个线程的消息时应等待，直到消息到达时才被唤醒。”所谓互斥，比较正式的说明是“线程互斥是指对于共享的进程系统资源，在各单个线程访问时的排它性。当有若干个线程都要使用某一共享资源时，任何时刻最多只允许一个线程去使用，其它要使用该资源的线程必须等待，直到占用资源者释放该资源。线程互斥可以看成是一种特殊的线程同步。”表示不能同时访问，也是个顺序问题，所以互斥是一种特殊的同步操作。

　　举个例子，设有一个全局变量global，为了保证线程安全，我们规定只有当主线程修改了global之后下一个子线程才能访问global，这就需要同步主线程与子线程，可用关键段实现。当一个子线程访问global的时候另一个线程不能访问global，那么就需要互斥。

**7.在Windows编程中互斥量与临界区比较类似，请分析一下二者的主要区别**。

1）互斥量是内核对象，所以它比临界区更加耗费资源，但是它可以命名，因此可以被其它进程访问。

2）从目的是来说，临界区是通过对多线程的串行化来访问公共资源或一段代码，速度快，适合控制数据访问。互斥量是为协调共同对一个共享资源的单独访问而设计的。

**8什么是临界资源和临界区呢？**

临界资源：能够被多个线程共享的数据/资源。

临界区：对临界资源进行操作的那一段代码

**9. 进程间通信的方式有几种？**

1). 无名管道( pipe )：管道是一种半双工的通信方式，数据只能单向流动，而且只能在具有亲缘关系的进程间使用。进程的亲缘关系通常是指父子进程关系。

2). 高级管道(popen)：将另一个程序当做一个新的进程在当前程序进程中启动，则它算是当前程序的子进程，这种方式我们成为高级管道方式。

3). 有名管道 (named pipe) ：有名管道也是半双工的通信方式，但是它允许无亲缘关系进程间的通信。

4). 消息队列( message queue ) ：消息队列是由消息的链表，存放在内核中并由消息队列标识符标识。消息队列克服了信号传递信息少、管道只能承载无格式字节流以及缓冲区大小受限等缺点。

5). 信号量( semophore ) ： 信号量是一个计数器，可以用来控制多个进程对共享资源的访问。它常作为一种锁机制，防止某进程正在访问共享资源时，其他进程也访问该资源。因此，主要作为进程间以及同一进程内不同线程之间的同步手段。

6). 信号 ( sinal ) ：信号是一种比较复杂的通信方式，用于通知接收进程某个事件已经发生。

7). 共享内存( shared memory ) ：共享内存就是映射一段能被其他进程所访问的内存，这段共享内存由一个进程创建，但多个进程都可以访问。共享内存是最快的 IPC(进程间通信)方式，它是针对其他进程间通信方式运行效率低而专门设计的。它往往与其他通信机制，如信号量，配合使用，来实现进程间的同步和通信。

8). 套接字( socket ) ： 套解口也是一种进程间通信机制，与其他通信机制不同的是，它可用于不同机器间的进程通信。