

2. 死锁产生的四个必要条件是什么？

- 互斥条件：资源是独占的且排他使用，进程互斥使用资源，即任意时刻一个资源只能给一个进程使用，其他进程若申请一个资源，而该资源被另一进程占有时，则申请者等待直到资源被占有者释放。
- 不可剥夺条件：进程所获得的资源在未使用完毕之前，不被其他进程强行剥夺，而只能由获得该资源的进程资源释放。
- 请求和保持条件：进程每次申请它所需要的一部分资源，在申请新的资源的同时，继续占用已分配到的资源。
- 循环等待条件：在发生死锁时必然存在一个进程等待队列 $\{P_1, P_2, \dots, P_n\}$ ，其中 P_1 等待 P_2 占有的资源， P_2 等待 P_3 占有的资源， \dots ， P_n 等待 P_1 占有的资源，形成一个进程等待环路，环路中每一个进程所占有的资源同时被另一个申请，也就是前一个进程占有后一个进程所深情地资源。

3. 某系统中有 n 个进程和 m 台打印机，系统约定：打印机只能一台一台地申请、一台一台地释放，每个进程需要同时使用的打印机台数不超过 m 。如果 n 个进程同时需要使用打印机的总数小于 $m+n$ ，试讨论，该系统可能发生死锁吗？并简述理由。

- 不会发生死锁， $\sum \text{MAX}(i) = \sum \text{LOC}(i) + \sum \text{NEED}(i) < m + n$
若发生死锁，则表明 m 个资源已分配，即 $\sum \text{LOC}(i) = m$ ，且进程无限等待资源。
 $\sum \text{NEED}(i) < n$ ，即至少有一个进程未申请资源，可以顺利执行完，并释放其所占资源，这与前面假设发生死锁相矛盾。

4. 线程的基本概念是什么？引入线程的好处是什么？

- 一个线程是进程的一个顺序执行流。同类的多个线程共享一块内存空间和一组系统资源
线程本身有一个供程序执行时的堆栈。线程在切换时负荷小，因此，线程也被称为轻负荷进程。一个进程中可以包含多个线程。
1. 线程占用的资源要比进程少的多（只占用一些执行线程所必要的信息，因而线程是程序执行的最小单位，进程是资源分配的最小单位）
 2. 根据资源占用情况，所以创建一个新的线程要比创建一个新的进程所多花费的代价小的多
 3. 进程之间的切换也比线程的切换小很多，也就是说：进程的切换，操作系统要处理很多东西
 4. 多线程的程序往往能很好的利用多处理机的并发性能
 5. 一个线程在等待慢速 I/O 结束的同时，此刻，另外一个线程可以计算其它的计算任务；而对于单线程而言的话，一旦线程阻塞，那么是不能执行其它的认为的
 6. 对于计算密集型任务或 I/O 密集型任务，多线程都能很容易的提高性能

5. 一个系统有 4 个进程和 5 个可分配资源，当前分配和最大需求如下：若保持该状态是安全状态，那么 x 的最小值是多少？
- 当前的 4 个进程中，只有 D 存在运行结束的可能性，进程 A, B, C 不可能立即运行完毕必须等待。因此第一个能够运行的进程为 D，且 $X \geq 1$ 。D 运行结束后可用资源量为 $(1, 1, 1+X, 2, 1)$ ， $X \geq 1$ 。只有 A 可以运行，进程 B 和 C 需要的等待第一类或第二类资源。此时，在 $X \geq 1$ 的操作下，A 已经可以运行了。A 运行完成后，可用资源量为 $(2, 1, 3+X, 3, 2)$ ， $X \geq 1$ 。只有 C 可以运行进程 B 需要等待第二类资源。此时 $X \geq 1$ 的条件下，C 已经可以运行了。C 运行完成后，可用的资源量为 $(3, 2, 3+X, 4, 2)$ ，此时，B 可以运行了，且不对 X 提出新的要求。因此 X 的最小值为 1。