计算机处理的数据和指令一律用二进制数表示

顺序执行程序

计算机运行过程中，把要执行的程序和处理的数据首先存入主存储器（内存），计算机执行程序时，将自动地并按顺序从主存储器中取出指令一条一条地执行，这一概念称作顺序执行程序。

计算机硬件由运算器、控制器、存储器、进程vs.线程

进程(process)与线程(thread)最大的区别是进程拥有自己的地址空间，某进程内的线程对于其他进程不可见，即进程A不能通过传地址的方式直接读写进程B的存储区域。进程之间的通信需要通过进程间通信(Inter-process communication，IPC)。与之相对的，同一进程的各线程间之间可以直接通过传递地址或全局变量的方式传递信息。

此外，进程作为操作系统中拥有资源和独立调度的基本单位，可以拥有多个线程。通常操作系统中运行的一个程序就对应一个进程。在同一进程中，线程的切换不会引起进程切换。在不同进程中进行线程切换，如从一个进程内的线程切换到另一个进程中的线程时，会引起进程切换。相比进程切换，线程切换的开销要小很多。线程于进程相互结合能够提高系统的运行效率。

线程可以分为两类：

一类是用户级线程(user level thread)。对于这类线程，有关线程管理的所有工作都由应用程序完成，内核意识不到线程的存在。在应用程序启动后，操作系统分配给该程序一个进程号，以及其对应的内存空间等资源。应用程序通常先在一个线程中运行，该线程被成为主线“程。在其运行的某个时刻，可以通过调用线程库中的函数创建一个在相同进程中运行的新线程。 用户级线程的好处是非常高效，不需要进入内核空间，但并发效率不高。

另一类是内核级线程(kernel level thread)。对于这类线程，有关线程管理的所有工作由内核完成，应用程序没有进行线程管理的代码，只能调用内核线程的接口。内核维护进程及其内部的每个线程，调度也由内核基于线程架构完成。内核级线程的好处是，内核可以将不同线程更好地分配到不同的CPU，以实现真正的并行计算。

事实上，在现代操作系统中，往往使用组合方式实现多线程，即线程创建完全在用户空间中完成，并且一个应用程序中的多个用户级线程被映射到一些内核级线程上，相当于是一种折中方案。输入设备和输出设备五大部分组成。

请问死锁的条件是什么？以及如何处理死锁问题？

解答：互斥条件(Mutual exclusion)：

资源不能被共享，只能由一个进程使用。

请求与保持条件(Hold and wait)：已经得到资源的进程可以再次申请新的资源。

非剥夺条件(No pre-emption)：已经分配的资源不能从相应的进程中被强制地剥夺。

循环等待条件(Circular wait)：系统中若干进程组成环路，该环路中每个进程都在等待相邻进程正占用的资源。

如何处理死锁问题：

忽略该问题。例如鸵鸟算法，该算法可以应用在极少发生死锁的的情况下。为什么叫鸵鸟算法呢，因为传说中鸵鸟看到危险就把头埋在地底下，可能鸵鸟觉得看不到危险也就没危险了吧。跟掩耳盗铃有点像。

检测死锁并且恢复。

仔细地对资源进行动态分配，以避免死锁。

通过破除死锁四个必要条件之一，来防止死锁产生。