

## 1.线程和进程的区别

他们都是**操作系统对系统资源的不同管理方式**。进程有独立的地址空间，而线程则是进程的一个执行单位。线程有独立的栈空间，同一进程下的线程共享同一片内存空间。

在cpu调度方面，**线程是调度的基本单位**。同一进程下的线程调度不会引起进程的切换。

在执行过程方面，**进程是系统资源分配的独立单位**。进程是有独立的内存空间。同一进程上的线程共享这片内存，同高通信效率。

逻辑上看，多线程的意义在于能够把同一进程上的多个部分同时执行。但操作系统没有把线程看作独立的应用。

## 2.内存管理：页式、段式、段页式

页式：根据我们用户程序的内存地址划分出若干固定大小的页。在物理内存空间上又划分出固定大小相等的块。将用户程序的页存放在物理内存的块上，实现离散分配。**优点：不存在内存的碎片化问题。**

**缺点：每一页的内存都没有独立的逻辑，都不能独立的编译。程序需要完全装载入内存。**

段式：跟页式相反，将我们用户内存空间根据逻辑划分出若干大小不同的段。

**优点：每一段都有自己独立的逻辑信息。都能够独立的编译。**

**缺点：因为每段大小都不固定，因此会产生内存的碎片化问题。**

## 3.进程的几种状态

创建、就绪、执行、阻塞、退出

**创建 -> 就绪：** 进程创建并初始化，等待分配处理机资源

**就绪 -> 执行：** 等待进程调度，等待系统分配**处理机资源**，一旦获取到处理机资源就变执行状态。

**执行 -> 阻塞：** 进程发生某种**中断事件**，比如 IO 请求。进程就会进入阻塞状态。

**执行 -> 就绪：** 时间片执行完毕，或者在抢占式调度算法中被更高优先级进程抢占。

**阻塞 -> 就绪：** 中断完成返回，比如 IO 请求完成。转化为就绪态等待系统调度。

## 4.虚拟内存

虚拟内存是内存管理的一种技术，允许进程程序不完全加载进内存空间，即一部分在内存空间，另一部分留在硬盘，当后期需要硬盘部分程序时再通过页面置换算法加载进内存(**LRU 算法**)。

**优势：**内存的逻辑地址空间大于物理地址空间。降低内存的限制。

## 5.死锁的必要条件

死锁的必要条件有 4 个：

**互斥**（一个资源只能被一个进程占用）

**请求与保持**（一个进程请求资源而阻塞，另一进程获取资源一直不放）

**不剥夺**（进程不能强行获取资源）

**循环等待**（若干进程间形成头尾循环等待资源）

**处理死锁：**

**预防：**不能达成 4 个死锁必要条件

避免：银行家算法

解决：(1) 重启系统

(2) 逐步撤回进程，解除死锁进程。

## 6. IPC 的几种通信方式

### 管道 Pipe：

所谓管道就是一份共享文件。管道通信指通过 写进程、读进程 来实现数据（字符流）交互。管道通信有 3 个协议：

1. 读/写进程间互斥

2. 同步（写进程写入一定量级数据如 4k 就需要睡眠等待，读进程读取，直到读取为空时睡眠并唤醒写进程写入。

3. 确认对方存在才能通信。

缺点：缓存区大小比较有限

### 套接字 Socket：

基于文件型：本地机器上进行进程通信。通过本地特殊文件读写进行通信。

基于网络型：通过远程主机 ip+端口进行进程间通信。

缺点：传输效率比较低

共享内存：通过共享文件方式进行进程间通信，通信速度快。

缺点：是进程间同步问题操作系统无法解决，需要利用工具手动进行同步。

### 消息队列：

不必借助共享存储区文件或数据结构来实现，而是通过以格式化的消息 Message 为单位，将通信数据封装到消息上，然后进程间进行消息传递。完成数据交互。

缺点：消息需要复制两次，额外消耗 cpu 资源。不适合大数据的交互。

### 信号量：

信号量是一个计数器，用于控制多个进程对共享资源的访问。信号量常作为一种锁机制。主要作为进程间或同一进程下的线程间的同步实现手段。

缺点：不能传递复杂消息，只能用来同步

## 7. 虚拟地址、逻辑地址、物理地址、线性地址区别

物理地址：机器中实际的物理内存地址。与内存存储单元的地址一一对应。物理地址的取值取决于机器实际内存大小。

逻辑地址/虚拟地址：线性地址的表示 段地址：段内偏移。而逻辑地址就是其中段内偏移的地址。逻辑地址一般相对于程序而言。我们编程一般用到的就是逻辑地址。

线性地址：是逻辑地址转换成物理地址的中间层媒介。程序代码一般会产生逻辑地址，也就是偏移地址，再加上相应的段地址就构成了线性地址也就是上面的 段地址：段内偏移，加入采用分页机制，那么线性地址通过页式管理单元转换成物理内存地址。

总结：逻辑地址/虚拟地址 -> [段式内存管理单元] -> 线性地址 -> [页式内存管理单元] -> 物理地址

注：假如没有开启分页机制，线性地址就是物理地址！