1.4 什么是中断？

​ 中断是指计算机运行过程中，**出现某些意外情况需主机干预时**，机器能**自动停止正在运行的程序并转入该事件的中断处理程序**，**处理完毕后又返回原程序的断点处继续执行。**

1.5 多个中断的处理方式是什么？

​ 处理多中断有两种方法。

第一种方法是当正在处理一个中断时，禁止再发生中断。

第二种方法是定义中断优先级，允许高优先级的中断打断低优先级的中断处理器的运行。

**2.1 操作系统设计的三个目标是什么？**

* **方便**：操作系统使计算机更易于使用。
* **有效**：操作系统允许以更有效的方式使用计算机系统资源。
* **扩展的能力**：在构造操作系统时，应该允许在不妨碍服务的前提下有效地开发、测试和引入新的系统功能。

**3.9 列出进程控制块中的三类信息。**

**进程标识（进程标识符、父进程的标识符、用户进程名、用户组名），**

**处理器状态信息（寄存器的内容、控制寄存器(如PSW寄存器)的内容、系统/用户堆栈指针）**

**进程控制信息（进程状态、等待事件和原因、进程优先级、正文段指针、数据段指针）**

**3.10 为什么需要两种模式（用户模式和内核模式）？**

**用户模式**下可以执行的指令和访问的内存区域都受到限制。这是为了防止操作系统受到破坏或者修改。**而在内核模式**下则没有这些限制，从而使它能够完成其功能。

5.7 列出对互斥的要求。

要提供对互斥的支持，必须满足以下要求:

1. **强制实施互斥(忙则等待):**在与相同资源或共享对象的临界区有关的所有进程中，一次只允许一个进程进入临界区。

2. 一个在非临界区停止的进程不能干涉其他进程。

3. **有限等待：**绝不允许出现需要访问临界区的进程被无限延迟的情况，即不会死锁或饥饿。

4. **空闲让进：**没有进程在临界区中时，任何需要进入临界区的进程必须能够立即进入。

5. 对相关进程的**执行速度**和**处理器的数量**没有任何要求和限制。

6. 一个进程驻留在临界区中的时间必须是有限的。

7、**让权等待：**当进程不能进入自己的临界区时，应立即释放处理机，以免陷入“忙等”

5.8 在信号量上可以执行什么操作？

为达到预期效果，可把信号量视为-一个值为整数的变量，整数值上定义了三个操作:

一个信号量可以初始化成非负数。

semWait 操作使信号量减1。若值变成负数，则阻塞执行semWait的进程，否则进程继续执行。

semSignal 操作使信号量加1。若值小于等于零，则被semwait操作阻塞的进程解除阻塞。

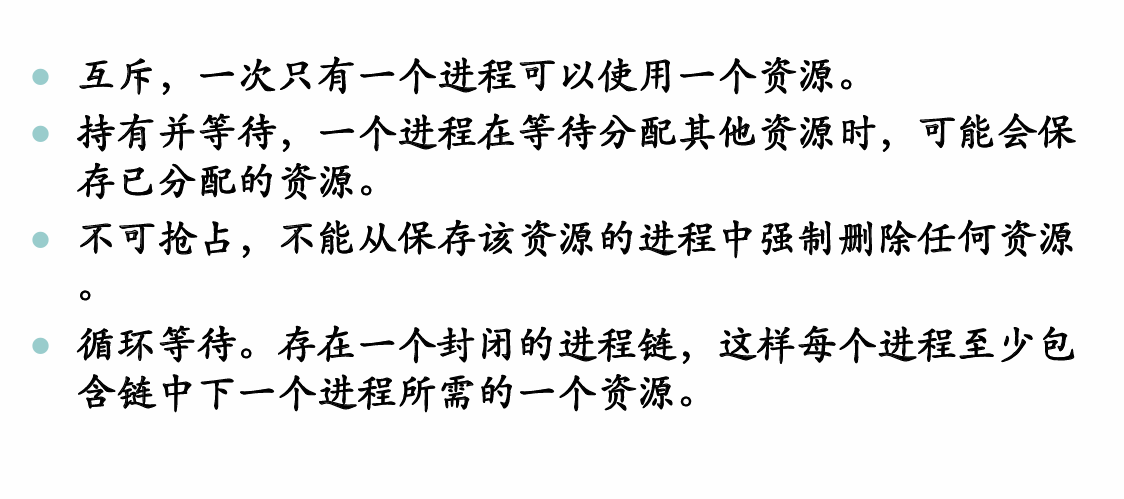
6.3死锁产生的四个必要条件

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{P1,P2,…,Pn},其中P1等待P2占有的资源，P2等待P3占有的资源，…，Pn等待P1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。



6.7 死锁避免、 检测和预防之间的区别是什么?

死锁预防是通过间接地限制三种死锁必要条件的至少一个或直接地限制循环等待的发生来避免死锁的出现。死锁避免允许可能出现的必要条件发生,但是采取措施确保不会出现死锁的情况。而死锁检测允许资源的自由分配,采取周期性的措施来发现并处理可能存在的死锁情况。

7.2为何需要重定位进程的能力?

在多道程序设计系统中，可用的内存空间通常被多个进程共享。通常情况下，程序员并不能事先知道在某个程序执行期间会有其他哪些程序驻留在内存中。此外还希望通过提供一个巨大的就绪进程池，能够把活动进程换入或换出内存，以便使处理器的利用率最大化。一旦程序被换出到磁盘，当下一次被换入时，如果必须放在和被换出前相同的内存区域，那么这将会是一个很大的限制。为了避免这种限制，需要把进程重定位到内存的不同区域。

7.7逻辑地址、相对地址和物理地址有何区别?

逻辑地址也称为虚拟地址，是相对于程序的起始地址的，独立于物理内存，在运行时通过地址映射机制转换为物理地址。

相对地址(relative address) 是逻辑地址的一个特例，是相对于某个基准地址（通常是某个段的起始地址或当前指令的地址）的地址。

物理地址( physical address) 或绝对地址是数据在物理内存中的实际位置。

**8.3 为何在使用虚拟内存时，局部性原理至关重要?**

​

程序局部性原理：进程运行一段时间内，其程序的执行往往呈现出高度的局部性，包括时间局部性和空间局部性。

根据局部性原理，程序执行过程中只要保证其局部性就可以一直执行，这是虚拟存储管理的基础。

8.5 转换检测缓冲区的目的是什么?

转换检测缓冲区是计算机系统中用于提高虚拟地址到物理地址转换效率的硬件。它存储了最近使用的虚拟地址到物理地址的映射，以加速地址转换过程。当 CPU 需要访问内存时，首先查询 TLB，如果TLB 命中则直接返回对应的物理地址，省去了访问页表的开销。

**9.1简要描述三种类型的处理器调度。**

* 长程调度：决定是否能加入待执行进程池。
* 中称调度：决定加入部分或全部位于内存中的进程集合。
* 短程调度：决定哪一个可用进程将被处理器执行。

9.5 抢占式调度和非抢占式调度有何区别?

在抢占式调度中，操作系统可以在任何时候中断正在运行的进程，并将 CPU分配给另一个处于就绪状态的进程。这意味着，一个高优先级的进程可以随时抢占正在运行的低优先级进程的 CPU时间片。这种方式可以保证高优先级进程得到更快的响应时间，但可能会导致低优先级进程的运行时间不确定。

相反，在非抢占式调度中，一个进程只有在自愿放弃CPU或者因为等待某个事件而被阻塞时，操作系统才会将 CPU分配给另一个进程。这种方式可以保证低优先级进程得到更稳定的运行时间，但可能会导致高优先级进程得不到及时响应。

**11.4 为什么希望用双缓冲区而不是单缓冲区来提高I/O的性能？**

双缓冲允许两个操作并行处理，而不是依次处理。典型的，在一个进程往一个缓冲区中传送数据（从这个缓冲区中取数据）的同时，操作系统正在清空（或者填充）另一个缓冲区。

IMG_256

12.5 列出并简单定义五种**文件组织**。

堆是最简单的文件组织形式。**数据按它们到达的顺序被采集，每个记录由一串数据组成**。

顺序文件是最常用的文件组织形式。在这类文件中，每个记录都使用一种固定的格式。**所有记录都具有相同的长度，并且由相同数目、长度固定的域按特定的顺序组成**。由于每个域的长度和位置已知，因此只需要保存各个域的值，每个域的域名和长度是该文件结构的属性。

索引顺序文件保留了顺序文件的关键特征：**记录按照关键域的顺序组织起来**。但它还增加了两个特征：用于支持随机访问的文件索引和溢出文件。**索引提供了快速接近目标记录的查找能力**。溢出文件类似于顺序文件中使用的日志文件，但是溢出文件中的记录可以根据它前面记录的指针进行定位。

索引文件：**只能通过索引来访问记录。其结果是对记录的放置位置不再有限制**，只要至少有一个索引的指针指向这条记录即可。此外，还**可以使用长度可变的记录**。

直接文件或散列文件：**直接文件使用基于关键字的散列**。

12.6 为什么在索引顺序文件中查找一个记录的平均搜索时间小于在顺序文件中的平均搜索时间？

在顺序文件中，查找一个记录**是按顺序检测每一个记录**直到有一个包含符合条件的关键域值的记录被找到。索引顺序文件提供一个**执行最小穷举搜索的索引结构**。

12.10 列出并简单定义**三种组块方式**。

固定组块（fixed blocking）：**使用固定长度的记录**，并且**若干条完整的记录被保存在一个块中**。在每个块的**末尾可能会有一些未使用的空间**，**称为内部碎片**。

可变长度跨越式组块（variable-length spanned blocking）：使用**长度可变的记录**，并且**紧缩到块中，使得块中没有未使用空间**。因此，**某些记录可能会跨越两个块，通过一个指向后继块的指针连接。**

可变长度非跨越式组块（variable-length unspanned blocking）：**使用可变长度的记录**，但并不采用跨越的方式。如果下一条记录比块中剩余的未使用空间大，则无法使用这一部分，因此在**大多数块中都会有未使用的空间**。

12.11 列出并简单定义三种文件分配方法。

连续分配:是指在创建文件时，**给文件分配一组连续的块。**

链式分配:基于单个块组成的链表，**链中的每一块都包含指向下一块的指针。**

索引分配：每个文件在文件分配表中**有一个一级索引**，分配给该文件的**每个分区在索引中都有一个表项。**