# Chapter 3

**1．高级调度不低级调度的主要仸务是什么？为什么要引入中级调度？**

答：高级调度的主要任务是根据某种算法，把外存上处于后备队列中的那些作业调入内存。低级调度是保存处理机的现场信息，按某种算法先取进程，再把处理器分配给进程。引入中级调度的主要目的是为了提高内存利用率和系统吞吐量。使那些暂时不能运行的进程不再占用内存资源，将它们调至外存等待，把进程状态改为就绪驻外存状态或挂起状态。

**6.为什么要引入高响应比优先调度算法？它有何优点？**

答：高响应比优先调度算法在执行时既考虑了作业的长、短，有考虑了作业的等待时间，当等待时间相同时，执行时间越短的作业越先执行，随着等待时间的增加，对应作业的优先级也会增加，因此长作业在等待一段时间后也会被调用。

**23. 何谓“优先级倒置”现象，可采取什么方法来解决？**

答：当进程之间存在资源争抢的时候，就可能产生“优先级倒置”的现象。即高优先级进程（或线程）被低优先级进程（或线程）延迟或阻塞，使得低优先级进程先与高优先级进程执行完。

解决方法：（优先级）

1. 规定不允许抢占进入临界区的低优先级进程的处理机。
2. 另一个比较实用的方式是当高优先级进程要进入临界区，去使用临界资源R，如果已有一个低优先级进程正在使用该资源，此时一方面被阻塞，另一方面有 继承 的优先级，并一直保持到的优先级，直至退出临界区。这样做的目的在于不让比优先级稍高，但比优先级低的进程如 进程插进来，导致延缓退出临界区。

**31. 在银行家算法中，若出现下述资源分配情况：**

|  |  |  |  |
| --- | --- | --- | --- |
| Process | Allocation | Need | Availabel |
|  | 0 0 3 2 | 0 0 1 2 | 1 6 2 2 |
|  | 1 0 0 0 | 1 7 5 0 |  |
|  | 1 3 5 4 | 2 3 5 6 |  |
|  | 0 3 3 2 | 0 6 5 2 |  |
|  | 0 0 1 4 | 0 6 5 6 |  |

1. 该状态是否安全？
2. 若进程P2提出请求Request(1，2，2，2)后，系统能否将资源分配给它？

答：（1）安全，可以找到一个安全序列P0-> P3-> P1-> P4-> P2，或P0-> P3-> P4-> P1-> P2

（2）不能，应该分配后，找不到一个安全序列，因此分配后，系统进入了不安全状态。

# Chapter 4

**7. 为什么要引入动态重定位？如何实现？**

答：程序在运行过程中经常要在内存中移动位置，为了保证这些被移动了的程序还能正常执行，必须对程序和数据的地址加以修改，即重定位。引入重定位的目的就是为了满足程序的这种需要。

要在不影响指令执行速度的同时实现地址变换，必须有硬件地址变换机构的支持，即须在系统中增设一个重定位寄存器，用它来存放程序在内存中的起始地址。程序在执行时，真正访问的内存地址是相对地址与重定位寄存器中的地址相加而形成的。

**13. 为什么要引入对换？对换可分为哪几种类型？**

答：

1. 原因：在多道程序环境下，一方面，在内存中的某些进程由于某事件尚未发生而被阻塞运行，但它却占用了大量的内存空间，甚至有时可能出现在内存中所有进程都被阻塞而迫使CPU停止下来等待的情况；另一方面，却又有着许多作业在外存上等待，因无内存而不能进入内存运行的情况。显然这对系统资源是一种严重的浪费，且使系统吞吐量下降。为了解决这一问题，在系统中又增设了对换(也称交换)设施。所谓“对换”，是指把内存中暂时不能运行的进程或者暂时不用的程序和数据调出到外存上，以便腾出足够的内存空间，再把已具备运行条件的进程或进程所需要的程序和数据调入内存。对换是提高内存利用率的有效措施。
2. 类型
   1. 如果对换是以整个进程为单位的，便称之为“整体对换”或“进程对换”。
3. 如果对换是以“页”或“段”为单位进行的，则分别称之为“页面对换”或“分段对换”，又统称为“部分对换”。

**14. 对文件区管理的目标和对对换空间管理的目标有何不同？**

1. 答：对文件区管理的主要目标：文件区占用磁盘空间的大部分，用于存放各类文件。由于通常的文件都是较长时间地驻留在外存上，对它访问的频率是较低的，故对文件区管理的主要目标是提高文件存储空间的利用率，然后才是提高对文件的访问速度。因此，对文件区管理采取离散分配的方式。
2. 对对换空间管理的主要目标：对换空间只占用磁盘空间的小部分，用于存放内存换出的进程。由于这些进程在对换区中驻留的时间是短暂的，而对换操作的频率却较高，故对对换空间管理的主要目标，时提高进程换入和换出的速度，然后才是提高文件存储空间的利用率。为此，对对换空间的管理采取连续分配的方式，较少考虑外存的碎片问题。

**22. 具有快表时是如何实现地址变换的？**

答：在CPU给出有效地址后，由地址变换机构自动将页号P送入高速缓冲寄存器，并将此号与高速缓存中的所有页号比较，若找到匹配页号，表示要访问的页表项在快表中。可直接从快表读出该页对应物理块号，送到物理地址寄存器中。如快表中没有对应页表项，则再访问内存页表，找到后，把从页表项中读出物理块号送地址寄存器；同时修改快表，将此页表项存入快表。但若寄存器已满，则OS必须找到合适的页表项换出。

**25. 为什么说分段系统比分页系统更易于实现信息的共享和保护？**

答：对于分页系统，每个页面是分散存储的，为了实现信息共享和保护，则页面之间需要一一对应起来，为此需要建立大量的页表项；

b.而对于分段系统，每个段都从0开始编址，并采用一段连续的地址空间，这样在实现共享和保护时，只需为所要共享和保护的程序设置一个段表项，将其中的基址与内存地址一一对应起来即可。

26. 分页和分段存储管理有何区别？

答：分页和分段都采用离散分配的方式，且都要通过地址映射机构来实现地址变换，这是它们的共同点；

b.对于它们的不同点有三，第一，从功能上看，页是信息的物理单位，分页是为实现离散分配方式，以消减内存的外零头，提高内存的利用率，即满足系统管理的需要，而不是用户的需要；而段是信息的逻辑单位，它含有一组其意义相对完整的信息，目的是为了能更好地满足用户的需要；第二页的大小固定且由系统确定，而段的长度却不固定，决定于用户所编写的程序；第三分页的作业地址空间是一维的，而分段的作业地址空间是二维的。