

1、什么是临界区？

答：

指的是访问资源的那段代码，任意时刻只允许一个线程对临界资源进行访问。拥有临界区对象的线程可以访问该临界资源，其它试图访问该资源的线程将被挂起，直到临界区对象被释放。

2、什么是死锁？

答：

是指两个或两个以上的进程在执行过程中，因争夺资源而造成的一种互相等待的现象，若无外力作用，它们都将无法推进下去。此时称系统处于死锁状态或系统产生了死锁，这些永远在互相等待的进程称为死锁进程。

3、介绍一下银行家算法

答：

主要思想是避免系统进入不安全状态，在每次进行资源分配时，它首先检查系统是否有足够的资源满足要求，如果有，则先试行分配，并对分配后的新状态进行安全性检查。如果新状态安全，则正式分配上述资源，否则拒绝分配上述资源。这样就保证系统始终处于安全状态，从而避免死锁现象的发生

4、分页和分段存储的区别

答：

一、分页存储

将程序的逻辑地址空间划分为固定大小的页(page)，而物理内存划分为同样大小的页框(page frame)。程序加载时，可将任意一页放入内存中任意一个页框，这些页框不必连续，从而实现了离散分配。

二、分段存储

在分段存储管理中，将程序的地址空间划分为若干个段(segment)，这样每个进程有一个二维的地址空间。每个段分配一个连续的分区，而进程中的各个段可以不连续地存放在内存的不同分区中。程序加载时，操作系统为所有段分配其所需内存，这些段不必连续。

三、分页和分段的不同点

分页机制和分段机制都是为了提高内存利用率，产生较少的内存碎片。

页和段都是离散存储的，所以两者都是离散分配内存的方式。但是每个页和段中的内存是连续的。

四、分页和分段的区别

页的大小是固定的，由操作系统决定，而段的大小不固定，取决于我们当前运行的程序。

分页仅仅是为了满足操作系统内存管理的需求，而段是逻辑信息的单位，它含有一组其意义相对完整的信息，在程序中可以体现为代码段，数据段，是为了满足用户的需要。

分页的作业地址空间是维一的，即单一的线性空间，程序员只须利用一个记忆符，即可表示一地址。分段的作业地址空间是二维的，程序员在标识一个地址时，既需给出段名，又需给出段内地址。

段一般比页大，因而段表比页表短，可以缩短查找时间，提高访问速度。

5、介绍一下时钟置换算法？

答：

简单的 CLOCK 算法是给每一帧关联一个附加位，称为使用位。当某一页首次装入主存时，该帧的使用位设置为 1；当该页随后再被访问到时，它的使用位也被置为 1。对于页替换算法，用于替换的候选帧集合看做一个循环缓冲区，并且有一个指针与之相关联。当某一页被替换时，该指针被设置成指向缓冲区中的下一帧。当需要替换一页时，操作系统扫描缓冲区，以查找使用位被置为 0 的一帧。每当遇到一个使用位为 1 的帧时，操作系统就将该位重新置为 0；如果在这个过程开始时，缓冲区中所有帧的使用位均为 0，则选择遇到的第一个帧替换；如果所有帧的使用位均为 1，则指针在缓冲区中完整地循环一周，把所有使用位都置为 0，并且停留在最初的位置上，替换该帧中的页。由于该算法循环地检查各页面的情况，故称为 CLOCK 算法，又称为最近未用(Not RecentlyUsed, NRU)算法。