**第四章 存储器管理**

**1、什么是静态链接、装入时动态链接和运行时的动态链接？**

答：静态链接是指在程序运行前，先将各目标模块及它们所需的库函数，链接成一个完整的装配模块，以后不再拆开的链接方式。

装入时动态链接是指将用户源程序编译后得到的一组目标模块，在装入内存时采用边装入边链接的链接方式。

运行时动态链接是指对某些目标模块的链接，是在程序执行中需要该目标模块时，才对它进行的链接

**2、简述分页系统和分段系统的异同点**

答：相同点：都采用离散分配的方式来提高内存利用率；都通过地址变换机构来实现地址变换。

区别：（1）页是信息的物理单位，分页是为了提高内存的利用率。段是信息的逻辑单位，含有一组意义相对比较完整的信息。分段是为了更好地满足用户的需求。

（2）页的大小固定且由系统决定。段的长度不固定，且由用户所编写的程序决定。

（3）分页的地址空间是唯一的，程序员只需要利用一个记忆符，便可以表示一个地址。分段的地址空间是二维的，程序员在标识一个地址时，既需要给出段名，有需要给出段内地址。

**3、什么情况下需要重定位？为什么要引入重定位？**

答：源程序经过编译、链接产生的装入模块一般是从0开始编制的，其中的地址都是相对于起始地址的相对地址。在将它装入内存时，其分配到的内存空间的起始地址通常不为0，因此**指令和数据的实际物理地址和装入模块中的相对地址不一致，此时，为了使程序能够正确执行，必须将相对地址转换为物理地址，即进行重定位。**

**引入重定位后，一方面可以保证CPU给出的逻辑地址能够准确映射到内存中的物理地址，另一方面也为程序在内存中移动提供了可能，便于实现内存“紧凑”，提高内存利用率。**

**4、在具有快表的段页式存储管理方式中，如何实现地址变换？**

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，并将此页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，与页内地址拼接成物理地址，然后访问内存，读出需要的指令或数据。

如快表中没有对应页表项，则首先需要访问在内存中的段表，从中读出对应段号页表所在内存的地址；其次再次访问内存，找到页表，从对应页表项中读出物理块号，与页内地址拼接成物理地址；然后按该地址第三次访问内存，找到需要的指令或数据。同时修改快表，将此页表项内容存入快表，若寄存器已满，则OS必须找到合适的页表项换出。

**5、什么是对换技术？为什么要引入对换？ 对换有哪些类型？**

答：对换时指内存中暂时不能运行的进程或者暂时不用的程序或数据换出到外存上，以便腾出足够的内存空间，把已经具备运行条件的进程或者进程所需要的程序和数据换入内存。

引入对换的目的是解决内存紧张问题，进一步提高了内存的利用率和系统吞吐量。

对换类型：整体对换、页面（分段）对换。

**6、在采用首次适应算法回收内存时，可能出现哪几种情况？应怎样处理这些情况？**

答：在采用首次适应算法回收内存时可能出现4种情况：

（1）回收区前邻空闲区。将回收区与前邻空闲区合并，将前邻空闲区大小修改为两者之和。

（2）回收区后邻空闲区。将两区合并，改后邻空闲区始址为回收区始址，大小为两者之和。

（3）回收区前后均邻空闲区。将三个分区合并，修改前邻空闲区大小为三者之和。

（4）回收区前后均不邻空闲区。为回收区设置空闲区表项，填入回收区始址和大小并插入空闲区队列。

**7、给定内存空闲分区，按地址从小到大为：100K、500K、200K、300K、600K。现有用户进程依次分别212K、417K、112K、426K。**

**（1）分别用首次适应算法、循环首次适应算法、最佳适应算法、最坏适应算法将它们装入到内存的哪个分区？**

**（2）哪个算法最有效利用内存？**

答：（1）按题意地址从小到大进行分区，如图所示：

|  |  |
| --- | --- |
| 分区号 | 分区长 |
| 1 | 100K |
| 2 | 500K |
| 3 | 200K |
| 4 | 300K |
| 5 | 600K |

1)首次适应算法：212K选中分区2，这时分区2还剩下288KB。417K选中分区5，这时分区5还剩下183KB。112K选中分区2，这时分区2还剩下176KB。426KB无分区能满足要求，等待。

2）循环首次适应算法：212K选中分区2，这时分区2还剩下288KB。417K选中分区5，这时分区5还剩下183KB。112K选中分区2，这时分区2还剩下176KB。426KB无分区能满足要求，等待。

3）最佳适应算法：212K选中分区4，这时分区4还剩88KB。417K选中分区2，这时分区2还剩83KB。112K选中分区3，这时分区3还剩88K。426K选中分区5，这时分区5还剩174KB。

4）最坏适应算法：212K选中分区5，这时分区5还剩388KB。417K选中分区2，这时分区2还剩83KB。112K选中分区5，这时分区5还剩176KB。426K无分区能满足，等待。

（2）最佳适应算法最有效使用内存。

**8、某系统采用页式存储管理策略，拥有逻辑空间32页，每页2KB，拥有物理空间1MB。**

**（1）写出逻辑地址的格式。**

**（2）若不考虑访问权限等，进程的页表有多少项？每项至少有多少位？**

**（3）如果物理空间减少一半，页表结构应相应作怎样的改变**？

答：（1）该系统拥有逻辑空间32页，故逻辑地址中页号必须用5位来描述；而每页为2KB，因此，页内地址必须用11位来描述，这样可得到它的逻辑地址格式：

|  |  |
| --- | --- |
| 15 11 | 10 0 |
| 页号 | 页内地址 |

（2）每个进程最多有32个页面，因此，进程的页表项最多为32项；若不考虑访问权限等，则页表项中只需给出页所对应的物理块块号，1MB的物理空间可分为29个内存块，故每个页表项至少有9位。

（3）如果物理地址空间减半，则页表中页表项数仍不变，但每项的长度可减少1位。

**9、已知某分页系统，主存容量为64K字节，页面大小为1K，对一个4页大的作业，其0、1、2、3页分别被分配到主存的2、4、6、7块中，试：将十进制的逻辑地址1023、2500、3500、4500转换成物理地址**。

答：逻辑地址1023。1023/1K，得到页号为0，页内地址为1023，查页表找到对应的物理块号为2，因此物理地址为2\*1K+1023=3071。

逻辑地址2500。2500/1K，得到页号为2，页内地址为452，查页表找到对应的物理块号为6，因此物理地址为6\*1K+452=6596。

逻辑地址3500。3500/1K，得到页号为3，页内地址为428，查页表找到对应的物理块号为7，因此物理地址为7\*1K+428=7596。

逻辑地址4500。4500/1K，得到页号为4，页内地址为404，越界，无法转换。

**10、对于下表所示的段表，请将逻辑地址（0,137）,(1,4000),(2,3600),(5,239)转换成物理地址。**

|  |  |  |
| --- | --- | --- |
| 段号 | 内存始址 | 段长 |
| 0 | 50K | 10K |
| 1 | 60K | 3K |
| 2 | 70K | 5K |
| 3 | 120K | 8K |
| 4 | 150K | 4K |

答：（1）段号0小于段表长5，故段号合法；由段表的第0项可获得段的内存始址位50K，段长位10K，由于段内地址137，小于段长10K，故段内地址也是合法的，因此可以得出对应的物理地址为50K+137=51337

（2）段号1小于段表长5，故段号合法；由段表的第1项可获得段的内存始址位60K，段长位3K，经检查，段内地址4000超过段长3K，因此产生越界中断。

（3）段号2小于段表长5，故段号合法；由段表的第2项可获得段的内存始址位70K，段长位5K，由于段内地址3600，小于段长10K，故段内地址也是合法的，因此可以得出对应的物理地址为70K+3600=75280.

（4）段号5等于段表长，故段号不合法，产生越界中断。