

它，若分配给进程 P2，系统还剩下的资源情况为 (0, 4, 0, 0)，此时系统中的资源将无法再满足任何一个进程的资源请求，从而导致系统进入不安全状态，容易引起死锁的发生。

#### 第四章

##### 1. 为什么要配置层次式存储器？

这是因为：

- a. 设置多个存储器可以使存储器两端的硬件能并行工作。
- b. 采用多级存储系统，特别是 Cache 技术，这是一种减轻存储器带宽对系统性能影响的最佳结构方案。
- c. 在微处理机内部设置各种缓冲存储器，以减轻对存储器存取的压力。增加 CPU 中寄存器的数量，也可大大缓解对存储器的压力。

##### 2. 可采用哪几种方式将程序装入内存？它们分别适用于何种场合？

将程序装入内存可采用的方式有：绝对装入方式、重定位装入方式、动态运行时装入方式；绝对装入方式适用于单道程序环境中，重定位装入方式和动态运行时装入方式适用于多道程序环境中。

##### 3. 何为静态链接？静态链接时需要解决两个什么问题？

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

将几个目标链接装配成一个装入模块时，需解决以下两个问题：

将相对地址进行修改。即将除第一个模块外的相对地址修改成装入模块中的相应的相对地址。

变换外部调用符号。即将每个模块中所用的外部调用符号，都变换为相对地址。

##### 4. 何谓装入时动态链接？装入时动态链接方式有何优点？

答：装入时动态链接是指将用户源程序编译后所得到的一组目标模块，在装入内存时，采用边装入边链接的一种链接方式，即在装入一个目标模块时，若发生一个外部模块调用事件，将引起装入程序去找相应的外部目标模块，把它装入内存中，并修改目标模块中的相对地址。

装入时动态链接方式有以下优点：

- 1) 便于修改和更新
- 2) 便于实现对目标模块的共享

##### 5. 何谓运行时动态链接？运行时动态链接方式有何优点？

答：运行时动态链接是将对某些模块的链接推迟到程序执行时才进行链接，也就是，在执行过程中，当发现一个被调用模块尚未装入内存时，立即由 OS 去找到该模块并将之装入内存，把它链接到调用者模块上。

优点：凡是在执行过程中未被用过的目标模块，都不会被调入内存和被链接到装入模块上，这样不仅能加快程序的装入过程，而且可节省大量的内存空间。

6. 在动态分区分配方式中, 应如何将各空闲分区链接成空闲分区链?

应在每个分区的起始地址部分, 设置一些用于控制分区分配的信息, 以及用于链接各分区的前向指针; 在分区尾部则设置一后向指针, 通过前, 后向指针将所有的分区链接成一个双向链. P128 图 4-7。主要使用双向链表。

7. 为什么要引入动态重定位? 如何实现?

- 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 and 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。

8. 什么是基于顺序搜索的动态分区分配算法? 它可分为哪几种?

答: 为了实现动态分区式分配, 将系统中的空闲分区组织成空闲分区表或空闲分区链。所谓顺序搜索, 是指按表或链的组织顺序, 检索表或链上记录的空闲分区, 去寻找一个最符合算法的、大小能满足要求的分区。 分区存储管理中常采用的分配策略有: 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。

9. 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: a. 回收区与插入点的前一个分区相邻接, 此时可将回收区与插入点的前一分区合并, 不再为回收分区分配新表项, 而只修改前邻接分区的大小;

b. 回收分区与插入点的后一分区相邻接, 此时合并两区, 然后用回收区的首址作为新空闲区的首址, 大小为两者之和;

c. 回收区同时与插入点的前后两个分区邻接, 此时将三个分区合并, 使用前邻接分区的首址, 大小为三区之和, 取消后邻接分区的表项;

d. 回收区没有邻接空闲分区, 则应为回收区单独建立一个新表项, 填写回收区的首址和大小, 并根据其首址, 插入到空闲链中的适当位置。

10. 什么是基于索引搜索的动态分区分配算法? 它可分为哪几种?

答: P131. 快速适应算法、伙伴系统、哈希算法

11. 令  $buddyK(x)$  表示大小为 2 的 k 次方、地址为 x 的块的伙伴系统地址, 试写出  $buddyK(x)$  通用表达式?

答: P126。考研的同学可以关注一下。Buddy System 是一种经典的内存管理算法, 在 Unix 和 Linux 操作系统中都有用到, 其作用是减少存储空间中的空洞, 减少碎片, 增加利用率。在有的“数据结构”课本中有算法介绍。

这道题我看不懂其题意, 大概是分配内存时 x 要找的空闲块是:

$pow(2, k-1) \leq buddyK(x) \leq pow(2, k)$  // pow 是乘方函数

Buddy System 是一种连续的内存管理方法, 可以结合离散的分页分配方法使用, 有利于为进程分配连续的物理块, 以提高分页法的效率和程序的局部性。

12. 分区存储管理中常采用哪些分配策略? 比较它们的优缺点。

分区存储管理中常采用的分配策略有: 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。

a. 首次适应算法的优缺点: 保留了高址部分的大空闲区, 有利于后到来的大型作业的分配; 低址部分不断被划分, 留下许多难以利用的、小的空闲区, 且每次分区分配查找时都是从低址部分开始, 会增加查找时的系统开销。

b. 循环首次适应算法的优缺点: 使内存中的空闲分区分布得更均匀, 减少了查找时的系统开销; 缺乏大的空闲分区, 从而导致不能装入大型作业。

c. 最佳适应算法的优缺点: 每次分配给文件的都是最适合该文件大小的分区; 内存中留下许多难以利用的小的空闲区。

d. 最坏适应算法的优缺点: 给文件分配分区后剩下的的空闲区不至于太小, 产生碎片的几率最小, 对中小型文件分配分区操作有利; 使存储器中缺乏大的空闲区, 对大型文件的分区分配不利。

13. 为什么要引入对换？对换可分为哪几种类型？

答：在多道环境下，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，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，在操作系统中引入了对换（也称交换）技术。可以将整个进程换入、换出，也可以将进程的一部分（页、段）换入、换出。前者主要用于缓解目前系统中内存的不足，后者主要用于实现虚拟存储。

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？

答：对文件区管理的主要目标是提高文件存储空间的利用率，然后才提高对文件的访问速度，因此，对文件区空间的管理采取离散分配方式。

对对换空间管理的主要目标是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，因此，对对换区空间的管理采取连续分配方式，较少的考虑外存中的碎片问题。

15. 为实现对换，系统应具备哪几方面功能？

答：系统应具备三方面功能：对换空间管理，进程换出，进程换入。

16. 在以进程为单位进行对换时，每次是否将整个进程换出？为什么？

答：在以进程为单位进行对换时，并非每次将整个进程换出。这是因为：

- a. 从结构上讲，进程是由程序段、数据段和进程控制块组成的，其中进程控制块总有部分或全部常驻内存，不被换出。
- b. 程序段和数据段可能正被若干进程共享，此时它们也不能被换出。

17. 基于离散分配时所用的基本单位不同，可将离散分配分为哪几种？

答：分页存储管理方式、分段存储管理方式和段页式存储管理方式。

18. 什么是页面？什么是物理块？页面的大小应如何确定？

答：页面：分页存储管理将进程的逻辑地址空间分成若干个页，并为各页加以编号。物理块：把内存的物理地址空间分成若干个块，并为各块加以编号。页面大小应选择适中，且页面大小应该是 2 的幂，通常为 1KB~8KB。

19. 什么是页表？页表的作用是什么？

答：页表是分页式存储管理使用的数据结构。一个进程分为多少页，它的页表就有多少行。每一行记录进程的一页和它存放的物理块的页号、块号对应关系。页表用于进行地址变换。

20. 为实现分页存储管理，需要哪些硬件支持？

答：需要有页表机制、地址变换机构的硬件支持。

21. 在分页系统中是如何实现地址变换的？

答：利用地址变换机构实现从逻辑地址到物理地址的转变，通过页表来实现从页号到物理块号的变换，将逻辑地址中的页号转换为内存中的物理块号。

22. 具有快表时是如何实现地址变换的？

23. 较详细的说明引入分段存储管理方式是为了满足用户哪几个方面的需求。

答：1) 方便编程。用户通常把自己的作业按照逻辑关系划分为若干段，每段都从 0 编址，并有自己名字和长度。因此，希望要访问的逻辑地址是由段名和段内偏移量决定。

2) 信息共享。在实现对程序和数据的共享时，是以信息逻辑单位为基础。分页系统中的页是存放信息的物理单位，无完整意义，不便于共享；段是信息的逻辑单位。为了实现段的共享，希望存储管理能与用户程序分段的组织方式相适应。

3) 信息保护。对信息的逻辑单位进行保护,分段能更有效方便地实现信息保护功能。

4) 动态增长。在实际应用中,有些段特别是数据段,在使用过程中会不断增长,事先又无法确切知道增长多少。分段存储管理方式能较好解决这个问题。

5) 动态链接。运行时先将主程序对应的目标程序装入内存并启动运行,运行过程中又需要调用某段时,才将该段调入内存链接。所以动态链接也要求以段作为管理单位。

24. 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,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. 为什么说分段系统较之分页系统更易于实现信息共享和保护?

答:a. 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. 分页和分段有何区别?

答:a. 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。

27. 试全面比较连续分配和离散分配方式。

答:a. 连续分配是指为一个用户程序分配一个连续的地址空间,包括单一连续分配方式和分区式分配方式,前者将内存分为系统区和用户区,系统区供操作系统使用,用户区供用户使用,是最简单的一种存储方式,但只能用于单用户单任务的操作系统中;分区式分配方式分为固定分区和动态分区,固定分区是最简单的多道程序的存储管理方式,由于每个分区的大小固定,必然会造成存储空间的浪费;动态分区是根据进程的实际需要,动态地为之分配连续的内存空间,常用三种分配算法:首次适应算法,该法容易留下许多难以利用的小空闲分区,加大查找开销;循环首次适应算法,该算法能使内存中的空闲分区分布均匀,但会致使缺少大的空闲分区;最佳适应算法,该算法也易留下许多难以利用的小空闲区;

b. 离散分配方式基于将一个进程直接分散地分配到许多不相邻的分区中的思想,

分为分页式存储管理,分段存储管理和段页式存储管理。分页式存储管理旨在提高内存利用率,满足系统管理的需要,分段式存储管理则旨在满足用户(程序员)的需要,在实现共享和保护方面优于分页式存储管理,而段页式存储管理则是将两者结合起来,取长补短,即具有分段系统便于实现,可共享,易于保护,可动态链接等优点,又能像分页系统那样很好的解决外部碎片的问题,以及为各个分段可离散分配内存等问题,显然是一种比较有效的存储管理方式;

c. 综上所述,连续分配方式和离散分配方式各有各的特点,应根据实际情况加以改进和利用。

## 第五章

1. 常规存储器管理方式具有哪两大特征?它对系统性能有何影响?

答:一次性:进程必须全部装入内存,对空间浪费非常大;

驻留性:在程序运行过程中,进程全部驻留在内存,暂时不用的数据无法释放。