

第五章 虚拟存储器

一、教学目的要求：

1. 掌握虚拟存储器的概念
2. 掌握请求页式存储管理的实现
3. 掌握页式置换算法
4. 熟悉请求段式存储管理的实现

二、内容分析：

1. 概述：本部分内容从整体上讲存储管理所要实现的功能，从分区到页式、段式、段页式的存储管理在内存利用率、内存扩充、内存共享与保护等方面的优化。

2. 教学重点：

1) 虚拟存储器的概念、特点

2) 请求页式置换算法

3. 教学难点：

1) 虚拟存储器的概念

2) 置换算法

虚拟存储器的概念

问题的提出：当一个大作业的地址空间大于整个内存的空间时，在上述的几种存储管理技术中，这个作业因不能全部装入内存而不能进入系统运行。

解决问题的办法：1. 扩大内存的容量。但受到机器硬件结构的限制，且系统成本增加；2. 另辟蹊径。程序部分装入内存，即可运行！

作业执行前全部装入内存的不合理性：造成内存浪费方面，有许多代码几乎不被执行，数组、队列等分配比实际需要大得多的内存；另一方面，即使整个程序在执行过程中都用得到，但也不是同时都用得到。因此，只把当前运行的那部分程序和数据装入内存，其余部分暂时放在外存上，待需要它们时，再分别调入内存。

这种做的好处，除解决上述问题外，还有：

① 用户编程不必考虑内存容量的限制，只要按照实际问题的需要来确定合适的算法和数据结构，从而简化了设计任务；

② 由于每个作业只装入一部分，因而在一定容量的内存中可同时装入更多的作业，增加 CPU 的利用率和系统的吞吐量。

操作系统把各级存储器统一管理起来，把一个程序当前正在使用的部分放在磁盘上，就启动执行它。操作系统根据程序执行时的要求和内存的实际使用情况随机地对每一个程序进行换入/换出。这样，就给用户提供了一个比真实的内存空间大得多的地址空间，这就是虚拟存储器（Virtual Memory），即是用户能作为可编址内存对待的存储空间，在这种计算机系统中虚拟地址被映象成实地址。

简单地说，虚拟存储器是由操作系统提供的一个假想的特大存储器。

实现虚拟存储技术的物质基础是二级存储器结构和动态地址转换机构（DAT）。虚拟存储技术的动态地址转换机构是在程序运行时把逻辑地址转换成物理地址，以实现动态定位。

虚拟存储器根据地址空间的结构不同可以分为两类：分页的虚拟存储器和分段的虚拟存储器，也可以将二者结合起来，构成段页式的虚拟存储器。

常规存储器管理方式的特征

(1) 一次性：即要求将一个作业全部装入内存方能运行。而正是这一特征导致了上述两种情况的发生。此外，还有许多作业在每次运行时，并非其全部程序和数据都要用到。如果一次性地装入其全部程序，也是一种对内存空间的浪费

(2) 驻留性：所谓驻留性就是指作业被装入内存后，整个作业都会一直驻留在内存中，直至作业运行结束。作业常驻内存存在不合理性，比如：因输入输出操作尚未完成而处于长期等待状态的运行进程或某些一次性运行程序对宝贵内存资源的占据

由此可以看出，上述的一次性及驻留性，使一些需要运行的作业无法装入运行，从而严重降低内存利用率和减少系统吞吐量。

接下来我们一起来看看关于局部性原理的知识

程序运行时存在的局部性现象，很早就被人发现，但直到 1968 年，P.Denning 才真正指出：程序在执行时将呈现出局部性规律，即在一较短的时间内，程序的执行仅局限于某个部分，相应地，它所访问的存储空间也局限于某个区域。

局限性又表现在下述两个方面：

(1) 时间局限性。如果程序中的某条指令一旦执行，则不久以后该指令可能再次执行；如果某数据被访问过，则不久以后该数据可能再次被访问。产生时间局限性的典型原因是 由于在程序中存在着大量的循环操作。

(2) 空间局限性。一旦程序访问了某个存储单元，在不久之后，其附近的存储单元也将 被访问，即程序在一段时间内所访问的地址，可能集中在一定的范围之内，其典型情况便是程序的顺序执行。

基于局部性原理，我们就有了虚拟存储器的定义

所谓虚拟存储器，是指具有请求调入功能和置换功能，能从逻辑上对内存容量加以扩充的一种存储器系统。其逻辑容量由内存容量和外存容量之和所决定，其运行速度接近于内存速度，而每位的成本却又接近于外存。可见，虚拟存储技术是一种性能非常优越的存储器管理技术，故被广泛地应用于大、中、小型机器和微型机中。

那么虚拟存储器是如何实现的呢？我们一起来看一下它的实现方法：

虚拟存储器的实现，都毫无例外地建立在离散分配的存储管理方式的基础上。目前，所有的虚拟存储器都是采用下述方式之一实现的。

1. 分页请求系统

它的技术构成为：分页 + 请求调页 + 页面置换

它的主要的硬件支持有：(1) 请求分页的页表机制。(2) 缺页中断机构。(3) 地址变换机构。

实现请求分页的软件有：请求调页、页面置换

2. 请求分段系统

它的技术构成为：分段 + 请求调段 + 分段置换

它的主要的硬件支持有：(1) 请求分段的段表机制。(2) 缺页中断机构。(3) 地址变换机构。

它的主要软件支持有：请求调段、分段置换

由此可见虚拟存储器技术有以下相关技术要点

作业部分装入内存即可启动运行，其余部分暂留磁盘。

程序执行过程采用页段访问机制：即已调入内存则直接访问、尚未调入内存则缺页/段中断及请求调入，引入页段置换功能 这样大用户程序可以在小内存空间的运行，多道程序度的得到提高

同样我们也可以得出虚拟存储器的特征

离散性：存储器分配是采用离散分配方式

多次性：作业被分成多次调入内存和运行

对换性：程序和数据在作业运行过程中的换进/出

虚拟性：内存容量的逻辑扩充

虚拟存储器的基本特征：

虚拟扩充：不是物理上，而是逻辑上扩充了内存容量；

部分装入：每个作业不是一次性装入，而是只装入一部分；
 离散分配：不必占用连续的内存空间，而是“见缝插针”；
 多次对换：所需的全部程序和数据要分次调入内存。
 虚拟存储器的容量受到两方面的限制：指令中表示地址的长度；外存的容量。

请求分页的基本思想

在简单分页系统中，要求运行的作业必须全部装入内存。而请求分页提供虚拟存储器，它的基本思想是：当要执行一个程序时才把它换入内存；但并不把全部程序都换入内存，而是用到哪一页时才换入它。由于这种页面是根据请求而装入的，所以这种存储管理方法叫做请求分页存储管理。

1. 硬件支持及缺页处理

- (1) 页表机制；
- (2) 缺页中断机构；
- (3) 具有快表的地址转换机构。

2. 页的共享和保护

为避免同时在内存中有同一页面的两个副本，需采用共享页面。其方法是使这些相关进程的逻辑空间中的页指向相同的内存块。类似于主程序多次调用子程序。

在页表的表项中设置存取控制字段，用于指明对应内存块中的内容允许执行何种操作，从而禁止非法访问。应当指出：在分页系统中实现页的共享是比较困难的。

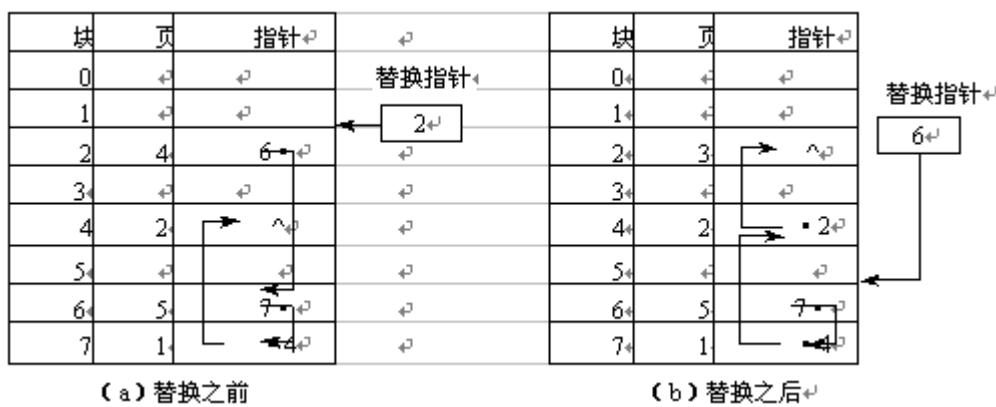
页的置换算法

1. 最优算法 (OPT 算法)

最理想的页面置换算法是：从内存中移出以后不再使用的页面；如无这样的页面，则选择以后最长时间内不需要访问的页。这就是最优算法的思想。这种算法本身不是一种实际的方法，因为页面访问的顺序是很难预知的。但是，可把它作为一种评价标准，比较其他实用方法的优劣，所以，最优算法只具有理论上的意义。

1. 先进先出算法 (FIFO 算法)

这种算法的基本思想是：总是先淘汰那些驻留在内存时间最长的页面，即先进入内存的页面先被置换掉。理由是：最先进入内存的页面不再被访问的可能性最大。



注意：有可能会产生 Belady 现象（内存分页越多，缺页率越高）

2. 最久未使用页面置换算法 (LRU 算法)

这种算法的基本思想是，如果某一页被访问了，那么它很可能马上又被访问；反之，如果某一页很长时间没有被访问，那么最近也不太可能会被访问。这种算法考虑了程序设计的局部性原理。其实质是，当需要置换一页时，选择在最近一段时间最久未使用的页面予以淘汰。实现这种算法可通过周期性地对“引用位”进行检查，并利用它来记录一页自上次被访问以来所经历的时间 t ，淘汰时选择 t 最大的页面。

3. LRU 近似算法

这种算法，只要在存储分块表（或页表）中设一个“引用位”，当存储分块表中的某一页被访问时，该位由硬件自动置 1，并由页面管理软件周期性把所有引用位置 0。这样，在一个时间周期 T 内，某些被访问过的页面其引用位为 1，而未被访问过的页面其引用位为 0。因此，可根据引用位的状态来判别各页面最近的使用情况。当需要置换一页时，选择其引用位为 0 的页。

碎片与抖动问题

1. 碎片问题

解决碎片问题的比较好的方法是采用分页技术，在纯分页存储管理系统中，因存储区划分成固定大小的块，而用户作业也划分成与块相等的若干页，每个作业调入内存时，除最后一个页面可能有页内碎片出现外，其余页不存在碎片问题，一般来说，平均每个作业可能有半页的内碎片。

2. 抖动现象

避免抖动现象最根本的方法是控制多道程序的道数，使得每个用户作业都有足够的内存空间可供使用。但作业的个数又不能太少，否则，会影响处理机的利用率。最好是使处理机利用率较高，又不至于使系统发生抖动，这是一个很难解决的问题，牵扯到程序的局部性问题，并需借助于工作集模型。

三、本章小结：

本节主要介绍了虚拟存储器的概念、请求分页存储管理和请求分段存储管理方式的思想和实现、请求分页管理方式中的几种页面置换算法，如先进先出置换算法、最佳置换算法、最近最久未使用置换算法、最近最少未使用置换算法。