

6.1 需求和解决思路

1. 背景

内存管理算法都基于一个基本要求：执行指令必须在物理内存中。满足这一要求的第一种方法是将整个进程放在内存中，这使得程序的大小被限制在物理内存的大小以内。动态载入能帮助减轻这一限制，但是它需要程序员特别小心并且需要一些额外的工作。事实上，研究实际程序会发现，在许多情况下并不需要将整个程序放到内存中。

在1968年P. Denning指出，程序在执行时将呈现出局部性规律，即在一段时间内，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内。局部性原理(principle of locality)是指程序在执行过程中的一个较短时期，所执行的指令地址和指令的操作数地址，分别局限于一定区域。还可以表现为：

(1) 时间局部性，即一条指令的一次执行和下次执行，一个数据的一次访问和下次访问都集中在一个较短时期内；

(2) 空间局部性，即当前指令和邻近的几条指令，当前访问的数据和邻近的数据都集中在一个较小区域内。

在实际程序中，局部性通常表现出以下一些特点：

(1) 程序在执行时，大部分是顺序执行的指令，少部分是转移和过程调用指令。

(2) 过程调用的嵌套深度一般不超过5，因此执行的范围不超过这组嵌套的过程。

(3) 程序中存在相当多的循环结构，它们由少量指令组成，而被多次执行。

(4) 程序中存在相当多对一定数据结构的操作，如数组操作，往往局限在较小范围内。

虚拟存储技术允许执行进程不必完全在内存中。虚拟存储(virtual memory)将内存抽象成一个巨大的、统一的存储数组，进而将用户逻辑内存与物理内存分开。其基本原理如下：

(1) 在程序装入时，不必将其全部读入到内存，而只需将当前需要执行的部分页或段读入到内存，就可让程序开始执行。

(2) 在程序执行过程中，如果需执行的指令或访问的数据尚未在内存（称为缺页或缺段），则由处理器通知操作系统将相应的页或段调入到内存，然后继续执行程序。

(3) 操作系统将内存中暂时不使用的页或段调出保存在外存上，从而腾出空间存放将要装入的程序以及将要调入的页或段——具有请求调入和置换功能，只需程序的一部分在内存就可执行，对于动态链接库也可以请求调入

这种方案的一个显著的优点就是程序可以比物理内存大，只有程序的部分在内存中，从而在现有物理内存有限的情况下，可执行较大的用户程序(见图6.1)。

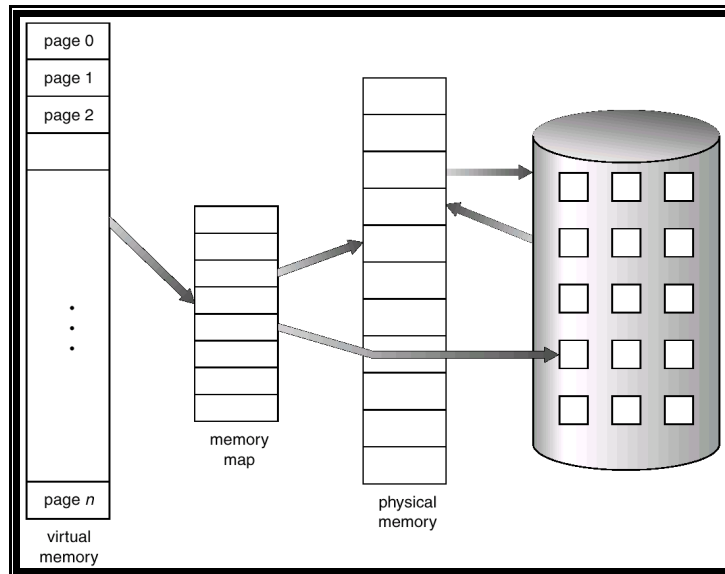


图 6.1 虚拟存储大于物理内存

能够执行只有部分在内存中的程序可带来很多好处：

(1) 程序不再受现有的物理内存空间限制。用户可以为一个巨大的虚拟地址空间(virtual address space)编写程序，简化了编程工作量。并且不会影响编程时的程序结构(与覆盖技术比较)。

(2) 因为每个用户程序使用了更少的物理内存，所以更多的程序可以同时执行，CPU 使用率也相应增加，而响应时间或周转时间并不增加。

(3) 由于载入或交换每个用户程序到内存内所需的 I/O 会更少，用户程序会运行得更快。

除了将逻辑内存与物理内存分开，虚拟存储也允许文件和内存通过共享页而为两个或多个进程所共享。这带来了如下优点：

(1) 通过将共享对象映射到虚拟地址空间，系统库可为多个进程所共享。虽然每个进程都认为共享库是其虚拟地址空间的一部分，而共享库所用的物理内存的实际页是为所有进程所共享。通常，库是按只读方式来链接每个进程的空间。

(2) 虚拟存储允许进程共享内存。两个或多个进程之间可以通过使用共享内存来通信。虚拟存储允许一个进程创建内存区域，以便与其他进程进行共享。共享该内存区域的进程认为它是其虚拟地址空间的一部分，而事实上这部分是共享的，如图 6.2 所示。

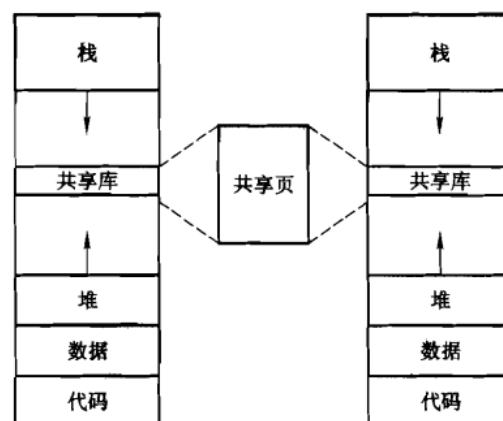


图 6.2 使用虚拟存储的共享库

(3) 虚拟存储可允许在用系统调用 `fork()` 创建进程期间共享页，从而加快进程创建。

虚拟存储是具有请求调入功能和置换功能，能仅把进程的一部分装入内存便可运行进程的存储器系统，它能从逻辑上对内存容量进行扩充的一种虚拟的存储器系统。

在虚拟存储技术下，程序员不受内存存储的限制，虚拟存储也允许进程很容易地共享文件和地址空间，还为创建进程提供了有效的机制，使得编程更容易。程序员不再需要担心可用的有限物理内存空间，只需要关注所要解决的问题。但是，虚拟存储的实现并不容易，如果使用不当可能会极大地降低性能。

2. 虚拟地址空间

进程的虚拟地址空间就是进程如何在内存中存放的逻辑(或虚拟)视图。通常，该视图为进程从某一逻辑地址(如地址 0)开始，连续存放，如图 6.2 所示。当然，物理地址可以按页帧来组织，且分配给进程的物理页帧也可能不是连续的。这就需要内存管理单元 (MMU) 将逻辑页映射到内存的物理页帧。

如图 6.3 所示，允许随着动态内存分配，堆可向上生长。类似地，还允许随着子程序的不断调用，堆可以向下生长。堆与栈之间的巨大空白空间(或洞)为虚拟地址的二部分，只有在堆与栈生长时，才需要实际的物理页。包括空白的虚拟地址空间称为稀地址空间。采用稀地址空间的优点是：随着程序的执行，栈或堆段的生长或需要载入动态链接库(或共享对象)时，这些空白可以填充。

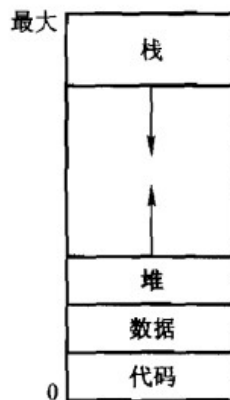


图 6.3 虚拟地址空间

所以虚拟存储的特征如下：

- (1) 物理内存分配不连续，虚拟地址空间使用也不连续（数据段和栈段之间的空闲空间，共享段和动态链接库占用的空间）。
- (2) 调入和调出是对部分虚拟地址空间进行(与交换的比较)。
- (3) 通过物理内存和快速外存相结合，提供大范围的虚拟地址空间：
 - i. 范围大，但占用容量不超过物理内存和外存交换区容量之和。
 - ii. 占用容量包括：进程地址空间中的各个段以及操作系统代码。