

[ JACO98b ] 考察了多种微处理器中的虚拟内存的硬件组织。

[ IBM86 ] 非常中立地说明了在优化虚拟内存策略 MVS 时，站点管理员可以使用的工具和选项，非常值得一读。该文档阐述了问题的复杂度。

[ VAHA96 ] 是对于各种 UNIX 中使用的内存管理方案的最佳总结。[ GORM04 ] 深入分析了 Linux 内存管理。

**CARR84** Carr, R. *Virtual Memory Management*. Ann Arbor, MI: UMI Research Press 1984.

**DENN70** Denning, P. "Virtual Memory." *Computing Surveys*, September 1970.

**DOWD93** Dowdy, L., and Lowery, C. *P.S. to Operating Systems*. Upper Saddle River, NJ: Prentice Hall, 1993.

**GORM04** Gorman, M. *Understanding the Linux Virtual Memory Manager*. Upper Saddle River, NJ: Prentice Hall, 2004.

**IBM86** IBM National Technical Support, Large Systems. *Multiple Virtual Storage (MVS) Virtual Storage Tuning Cookbook*. Dallas Systems Center Technical Bulletin G320-0597, June 1986.

**JACO98a** Jacob, B., and Mudge, T. "Virtual Memory: Issues of Implementation." *Computer*, June 1998.

**JACO98b** Jacob, B., and Mudge, T. "Virtual Memory in Contemporary Microprocessors." *IEEE Micro*, August 1998.

**MILE92** Milenkovic, M. *Operating Systems: Concepts and Design*. New York: McGraw-Hill, 1992.

**VAHA96** Vahalia, U. *UNIX Internals: The New Frontiers*. Upper Saddle River, NJ: Prentice Hall, 1996.

## 推荐网站

- The Memory Management Reference ([www.memorymanagement.org](http://www.memorymanagement.org)): 包含了关于内存管理的所有特征的多个文档和链接。

## 8.8 关键术语、复习题和习题

### 关键术语

关联映射	内部碎片	分页	分段
请求分页	局部性	预先分页	slab 分配
外部碎片	页	实存	抖动
读取策略	缺页中断	驻留集	转换检测缓冲区
页框	页放置策略	驻留集管理	虚拟内存
散列表	页面置换策略	段	工作集
散列法	页表	段表	

### 复习题

- 8.1 简单分页与虚拟内存分页有什么区别？
- 8.2 解释什么是抖动。
- 8.3 为什么在使用虚拟内存时，局部性原理是至关重要的？
- 8.4 哪些元素是页表项中可以典型找到的元素？简单定义每个元素。
- 8.5 转换检测缓冲区的目的是什么？
- 8.6 简单定义两种可供选择的页面读取策略。
- 8.7 驻留集管理和页面置换策略有什么区别？
- 8.8 FIFO 和 CLOCK 页面置换算法有什么联系？
- 8.9 页缓冲实现的是什么？
- 8.10 为什么不可能把全局置换策略和固定分配策略组合起来？
- 8.11 驻留集和工作集有什么区别？
- 8.12 请求式清除和预约式清除有什么区别？

**习题**

- 8.1 考虑这样一个系统，该系统用 3 位表示页面编号，用 5 位表示偏移量。在该系统中内存以字节为单位进行存取。现在假设一个进程有 6 页，其页表如下：

有效位	页框号	修改位
1	4	1
1	7	0
0	1	0
1	0	1
0	2	0
0	1	0

在下面的问题中，如果发生页面失效而不进行处理。

a) 虚拟地址 158 进行物理地址转换后是多少？

b) 虚拟地址 53 进行物理地址转换后是多少？

c) 虚拟地址 195 进行物理地址转换后是多少？

- 8.2 考虑一个使用 32 位的地址和 1KB 大小的页的分页虚拟内存系统。每个页表项需要 32 位。需要限制页表的大小为一个页。

a) 页表一共需要使用几级？

b) 每一级页表的大小是多少？提示：一个页表的大小比较小。

c) 在第一级使用的页较小与在最底下一级使用的页较小相比，那种策略使用最小个数的页？

- 8.3 a) 图 8.4 中的用户页表中需要多少内存空间？

b) 假设需要设计一个散列倒排页表来实现与图 8.4 中相同的寻址机制，使用一个散列函数来将 20 位页号映射到 6 位散列表。表项包含页号、页框号和链指针。如果页表可以给每个散列表项分配最多 3 个溢出项的空间，散列倒排页表需要占用多大的内存空间？

- 8.4 一个进程分配给 4 个页框（下面的所有数字均为十进制数，每一项都是从 0 开始计数的）。上一次把一页装入到一个页框的时间、上一次访问页框中的页的时间、每个页框中的虚拟页号以及每个页框的访问位（R）和修改位（M）如下表所示（时间均为从进程开始到该事件之间的时钟时间，而不是从事件发生到当前的时钟值）。

虚拟页号	页 框	加载时间	访问时间	R 位	M 位
2	0	60	161	0	1
1	1	130	160	0	0
0	2	26	162	1	0
3	3	20	163	1	1

当虚拟页 4 发生缺页中断时，使用下列内存管理策略，哪一个页框将用于置换？解释原因。

a) FIFO（先进先出）算法

b) LRU（最近最少使用）算法

c) 时钟算法

d) 最佳（使用下面的访问串）算法

e) 在缺页中断之前给定上述的内存状态，考虑下面的虚拟页访问序列：

4, 0, 0, 0, 2, 4, 2, 1, 0, 3, 2

如果使用窗口大小为 4 的工作集策略来代替固定分配策略，会发生多少次缺页中断？每个缺页中断何时发生？

- 8.5 一个进程访问 5 页：A、B、C、D 和 E，访问顺序如下：

A; B; C; D; A; B; E; A; B; C; D; E

假设置换算法为先进先出，该进程在内存中有三个页框，开始时为空，请查找在这个访问顺序中传送的页号。对于 4 个页框的情况，请重复上面的过程。

8.6 考虑以下来自一个 460 字节程序的虚拟地址序列：

10、11、104、170、73、309、185、245、246、434、458、364

1) 给出引用串，假设页面大小为 100 字节。

2) 下面给出了三个算法，针对 1) 部分的引用串，计算缺页次数。对每一种情况，假设页框初始时为空，在每一个给定的时间点，给出在内存中有哪些页面。

a) LRU (有 200 字节的物理内存可用)

b) FIFO (有 200 字节的物理内存可用)

c) OPT (有 200 字节的物理内存可用)

8.7 在 VAX 中，用户页表以系统空间的虚拟地址进行定位。让用户页表位于虚存而不是内存中有什么好处？有什么缺点？

8.8 假设在内存中执行下列程序语句：

```
for (i = 1; i <= n; i++)
    a [i] = b [i] + c [i];
```

页尺寸为 1000 个字。令  $n=1000$ 。使用一台具有所有寄存器指令并使用了索引寄存器的机器，写出实现上面语句的一个示例程序，然后给出在执行过程中的页访问顺序。

8.9 IBM System/370 体系结构使用两级存储器结构，并且分别把这两级称做段和页，这里的分段方法缺少本章所描述的关于段的许多特征。对于这个基本的 370 体系结构，页尺寸可以是 2KB 或 4KB，段大小固定为 64KB 或 1MB。对于 370/XA 和 370/ESA 体系结构，页尺寸为 4KB，段大小为 1MB。这种方案缺少一般分段系统的哪些优点？370 的分段方法有什么好处？

8.10 考虑一个 64 位地址空间的系统，页面大小为 4KB (4096 字节)

a) 虚拟地址空间有多少地址？

b) 如果要实现一个简单的一级页表，该页表中有多少表项？

c) 如果虚拟地址空间非常大，则 b) 部分提出了一个很严重的问题：请问它是个什么问题？如何解决它？

8.11 考虑这样一个分页系统，该系统在内存中存放了二级页表，在 TLB 中存储了最近的访问的 16 个页表项。如果内存访问需要 80ns，TLB 检查需要 20ns，页面交换（写/读磁盘）时间需要 5000ns。假设有 20% 的页面置换被更改，如果 TLB 的命中率是 95%，缺页率是 10%，那么访问一个数据项需要多长时间？

8.12 考虑一个进程的页访问序列，工作集为  $M$  页框，最初都是空的。页访问串的长度为  $P$ ，包含  $N$  个不同的页号。对任何一种页面置换算法，

a) 缺页中断次数的下限是多少？

b) 缺页中断次数的上限是多少？

8.13 在论述一种页面置换算法时，一位作者用一个在循环轨道上来回移动的雪犁机来模拟说明：雪均匀地落在轨道上，雪犁机以恒定的速度在轨道上不断地循环，轨道上被扫落的雪从系统中消失。

a) 8.2 节讨论的哪一种页面置换算法可以它来模拟？

b) 这个模拟说明了页面置换算法的哪些行为？

8.14 在 S/370 体系结构中，存储关键字是与实存中每个页框相关联的控制字段。这个关键字中与页面置换有关的有两位：访问位和修改位。当为读或写而访问到页框中的任何地址时，访问位被置成 1；当一个新页被装入到该页框中时，访问位被置成 0。当在页框中的任何单元执行写操作时，修改位被置为 1。请给出一种方法，仅仅使用访问位来确定哪个页框是最近最少使用的。

8.15 考虑如下的页访问序列（序列中的每一个元素都是页号）：

1 2 3 4 5 2 1 3 3 2 3 4 5 4 5 1 1 3 2 5

定义经过  $k$  次访问后平均工作集大小为  $s_k(\Delta)=\frac{1}{k} \sum_{t=1}^k |w(t, \Delta)|$ ，并且定义经过  $k$  次访问后错过页的

概率为  $m_k(\Delta)=\frac{1}{k} \sum_{t=1}^k F(t, \Delta)$ ，其中如果缺页中断发生在虚拟时间  $t$ ，则  $F(t, \Delta)=1$ ，否则  $F(t, \Delta)=0$ 。

a) 当  $\Delta=1,2,3,4,5,6$  时，绘制与图 8.19 类似的图表来说明刚定义的访问序列的工作集。

b) 写出  $s_{20}(\Delta)$  关于  $\Delta$  的表达式。

c) 写出  $m_{20}(\Delta)$  关于  $\Delta$  的表达式。

- 8.16 VSW 驻留集合管理策略的性能关键是  $Q$  的值。经验表明，如果对一个进程使用固定的  $Q$  值，则在不同的执行阶段，缺页中断发生的频率有很大的差别。此外，如果对不同的进程使用相同的  $Q$  值，则缺页中断发生的频率会完全不同。这些差别表明，如果有一种机制可以在一个进程的生命周期中动态地调整  $Q$  的值，则会提高算法的性能。请基于这种目标设计一种简单的机制。
- 8.17 假设一个任务被划分成 4 个大小相等的段，并且系统为每个段建立了一个有 8 项的页描述符表。因此，该系统是分段与分页的组合。假设页大小为 2KB。
- 每段的最大尺寸为多少？
  - 该任务的逻辑地址空间最大为多少？
  - 假设该任务访问到物理单元 00021ABC 中的一个元素，那么为它产生的逻辑地址的格式是什么？该系统的物理地址空间最大为多少？
- 8.18 考虑一个分页式的逻辑地址空间（由 32 个 2KB 的页组成），将它映射到一个 1MB 的物理内存空间。
- 该处理器的逻辑地址格式是什么？
  - 页表的长度和宽度是多少（忽略“访问权限”位）？
  - 如果物理内存空间减少了一半，则会对页表有什么影响？
- 8.19 UNIX 内核可以在需要时动态地在虚存中增加一个进程的栈，但却从不缩小这个栈。考虑下面的例子：一个程序调用一个 C 语言子程序，这个子程序在栈中分配一个本地数组，一共需要 10KB 大小，内核扩展这个栈段来适应它。当这个子程序返回时，内核应该调整栈指针并释放空间，但它却未被释放。解释这时为什么可以缩小栈以及 UNIX 内核为什么没有缩小栈。

## 附录 8A 散列表

考虑下面的问题。一个表中有  $N$  项，每一项都由一个标号和一些附加信息组成，这类信息可以称为该项的值。希望能够对表执行一些普通操作，如插入、删除以及根据标号查找某一项。

如果这些项的标号是数字，范围从 0 到  $M-1$ ，一种简单的解决方案是使用一个长度为  $M$  的表。标号为  $i$  的项被插入到表中的第  $i$  个单元。只要这些项的长度是固定的，对表的查找就是微不足道的，只需要根据该项的数字标号在表中检索即可。此外，没有必要在表中保存每一项的标号，因为标号可以隐含在该项的位置中。这样的表称做直接访问表（direct access table）。

如果标号不是数字的，仍然可以使用直接访问的方法。把这些项表示成  $A[1], \dots, A[N]$ 。每一项  $A[i]$  由标号（或关键字） $k_i$  和值  $v_i$  组成。定义一个映射函数  $I(k)$ ，使得对所有的关键字， $I(k)$  的值在 1 和  $M$  之间，并且对任意的  $i$  和  $j$  有  $I(k_i) \neq I(k_j)$ 。在这种情况下，也可以使用直接访问表，表的长度等于  $M$ 。

如果  $M$  远大于  $N$ ，则这些方案就会出现问题。此时，表中未使用的表项部分很大，从而使得内存的使用非常低效。一种可供选择的方案是使用长度为  $N$  的表，并且在  $N$  个表项中保存  $N$  项（标号和值）。这个方案的存储量最小，但在对表进行搜索时，需要一些处理负担。有以下几种可能的搜索方法：

- **顺序搜索：**对于大表来说，这种蛮力的方法非常费时。
- **关联搜索：**通过适当的硬件，可以同时搜索表中的所有元素。这种方法并不是通用的，不能用于所有表。
- **二分搜索：**如果标号或标号的数字映射在表中按升序排列，则二分搜索比顺序搜索（见表 8.7）要快得多，并且不需要专门的硬件。

表 8.7 长度为  $M$  的表中，搜索  $N$  项中的一项的平均搜索长度

技术	搜索长度
直接	1
顺序	$\frac{M+1}{z}$
二分	$\log_2^M$
线性散列	$\frac{2-N/M}{2-2N/M}$
散列（使用链的溢出）	$1+\frac{N-1}{2M}$