

# 分页存储管理系统中内存有效访问时间的计算

夏春梅

(滨州学院 信息工程系, 山东 滨州 256603)

**摘 要:** 为让学生更好地掌握分页系统的基本原理, 针对分页存储管理中的内存有效访问时间的计算问题, 对基本分页存储系统和请求分页存储系统分别进行了探讨, 总结了多种情况下内存有效访问时间的计算公式, 并通过案例加以分析应用。通过分析, 使学生对分页存储管理系统的实现过程更加清晰。

**关键词:** 存储器; 页表; 快表; 有效访问时间; 计算

**中图分类号:** G 420      **文献标识码:** A      **DOI:** 10.13486/j.cnki.1673-2618.2015.06.018

## 0 引言

在现代操作系统中, 存储器是非常重要的资源之一。任何程序要想执行, 必须首先调入内存。现在内存的容量增长很快, 但正如帕金森定律所指出的“不管存储器有多大, 程序总可以把它填满”。<sup>[1]</sup> 因此, 能否合理使用存储器, 在很大程度上反映了操作系统的性能, 并直接影响整个计算机系统的工作效率。

分页存储管理方式是很多操作系统中使用的一种内存管理方式。它的基本思想是: 把内存空间分成若干个大小相等的固定的物理块或称之为页框, 并且从 0 开始进行编号, 相应地, 进程逻辑地址空间也按照物理块的大小划分成大小相等的页面或称之为页, 并且每一个进程的页面也从 0 开始进行编号。页面大小是操作系统可以选择的一个参数, 为了确定最佳的页面大小, 需要在几个互相矛盾的因素之间进行权衡。<sup>[2]</sup> 在为进程分配内存时, 以块为单位进行分配, 即每页获得一个物理块。将进程中的每一个页面映射到内存中的一个物理块, 这样就得到进程的页表。页表的作用是实现从页

号到物理块号的地址映射。根据程序运行时是必须装入所有页面还是可以仅装入部分页面, 又可以将分页存储管理方式分为基本分页存储管理方式和请求分页存储管理方式。

相对于连续分配方式, 分页存储管理系统中程序的运行过程发生了变化。内存有效访问时间的计算问题, 根据系统的实现方式(基本的还是虚拟的)、有无快表、程序运行的具体情况而有所不同。分析不同情况下的内存有效访问时间, 有利于学生透彻理解分页存储管理系统的基本原理和实现过程。

## 1 基本分页存储系统中的有效访问时间

从进程发出指定逻辑地址的访问请求开始, 经过地址变换, 到在内存中找到相应的实际物理地址单元并取出数据, 所需要花费的总时间, 称为内存的有效访问时间<sup>[3]</sup>, 简称 EAT。

### 1.1 没有快表时有效访问时间的计算

在进程运行期间, 需要将每条指令的逻辑地址变换为物理地址, 这个过程需借助页表来实现。由于页表的访问频率非常高, 页表大多驻留在内存

收稿日期: 2015-08-21

基金项目: 滨州学院专业核心课程项目(BYHXKC201412)

作者简介: 夏春梅(1977—), 女, 山东济宁人, 讲师, 硕士, 从事数据库技术与系统集成研究。

E-mail: xiachunmei123@163.com

中。因此,CPU 在执行指令时,先访问内存中的页表,根据页号检索到相应的物理块号,再根据块号和页内偏移量得到物理地址;然后根据前面得到的这个物理地址去访问相应的内存单元,取得或存入数据。假设访问一次内存的时间为  $t$ ,在这种系统中内存的有效访问时间为两次访问内存的时间之和。

### 1.2 有快表时有效访问时间的计算

在分页式系统中,需要考虑虚拟地址到物理地址的映射速度问题。在很多系统中设置了快表<sup>[3]</sup>这种结构,暂存当前访问的那些页表项。在进行地址变换时,首先检索快表,如果在快表中能找到相应的页表项,则拼接得到物理地址,不再查找内存中的页表。然后根据物理地址访问相应的内存地址单元。如果在快表中未找到相应的页表项,则仍需访问内存中的页表,还是需要两次访问内存。访问快表的时间和访问内存的时间相比非常小。假设快表的命中率为  $a$ ,访问快表的时间为  $\lambda$ ,引入快表后的有效访问时间分为查找页表项获得物理地址的平均时间和访问物理地址的时间之和,则有

$$EAT = [a \times \lambda + (1-a)(t+\lambda)] + t = 2t + \lambda - at。$$

也可以这样推导

$$EAT = a \times (\lambda + t) + (1-a)(\lambda + 2t) = 2t + \lambda - at。$$

如果忽略访问快表的时间,则又有

$$EAT = a \times t + (1-a) \times 2t = 2t - at。$$

### 1.3 案例分析

1.3.1 案例 有一个将页表存放在内存中的分页系统,在下面两种情况下,请计算有效访问时间分别为多少:

系统中未设置快表,访问一次内存需要  $0.2 \mu s$ ;系统中设置了快表,快表的命中率为  $90\%$ ,并且假定查快表需花的时间为  $0$ ,访问一次内存仍需  $0.2 \mu s$ 。

1.3.2 分析 本案例需要计算在有快表和没有快表两种情况下的有效访问时间,而且在计算时访问快表的时间忽略不计。应用上面的公式进行计算即可:需两次访问内存,故有效访问时间为  $2 \times 0.2 \mu s = 0.4 \mu s$ 。若能从快表中直接找到相应的页表项,则可立即形成物理地址去内存访问相应的数据;否则,仍需两次访问内存。有效访问时间为

$$0.9 \times 0.2 \mu s + (1-0.9) \times 0.2 \mu s \times 2 = 0.22 \mu s。$$

## 2 请求分页存储系统中的有效访问时间

在请求分页存储系统中,进程运行时,只需先调入部分页面,在运行过程中如果要访问的页不在内存,则发出缺页中断,由系统负责从外存调入。在调入时,如果内存已满,还需要从内存中选择一页换出到外存上,然后再调入所缺页面,即进行页面置换。在请求分页存储系统中,应用程序采用虚拟地址亦即逻辑地址,程序在装入时采用按需调入的存储管理策略。通过这种空间虚拟技术,可以从逻辑上扩大物理内存。<sup>[4]</sup>所以,在请求分页存储系统中,内存的有效访问时间不仅要考虑访问页表和访问物理地址数据的时间,还需要考虑缺页中断的处理时间。

### 2.1 三种情况下的时间计算

(1)要访问的页在内存,且其对应的页表项已放入快表中。此时内存有效访问时间分为查找快表的时间  $\lambda$  与访问物理地址的时间  $t$  之和:

$$EAT = \lambda + t。$$

(2)要访问的页在内存,但其对应的页表项不在快表中。此时,系统先查找快表,花费时间  $\lambda$ ,未找到;再查找内存中的页表,花费时间  $t$ ;将访问页的页表项放入快表中即更新快表,花费时间  $\lambda$ ;根据物理地址访问内存单元存取数据,花费时间  $t$ 。所以,有效访问时间由四部分组成:

$$EAT = \lambda + t + \lambda + t = 2(\lambda + t)。$$

(3)要访问的页不在内存中。系统先查找快表,花费时间  $\lambda$ ,未找到;再查找内存中的页表,花费时间  $t$ ,未找到;进行缺页中断处理,花费时间  $\epsilon$  (包含了更新页表的时间  $t$ );更新快表,花费时间  $\lambda$ ;根据物理地址访问内存单元存取数据,花费时间  $t$ 。所以,有效访问时间由五部分组成:

$$EAT = \lambda + t + \epsilon + \lambda + t = \epsilon + 2(\lambda + t)。$$

可以把以上三种情况结合起来,用一个公式统一表示。假设快表的命中率为  $a$ ,缺页率为  $b$ ,则有效访问时间可表示为

$$EAT = \lambda + a \times t + (1-a) \times [t + b \times (\epsilon + \lambda + t) + (1-b) \times (\lambda + t)]。$$

为了防止系统发生抖动,需要控制缺页率。当一个进程缺页率太高时,一般会调入更多的页面,反之可能调出部分页面。可以为所期望的缺页率

设置上限和下限<sup>[5]</sup>。在工作集算法中,保证最少缺页次数是通过调整工作集的大小来间接实现的,一种直接改善系统性能的方法是使用缺页频率替换算法<sup>[6]</sup>。

如果忽略访问快表的时间,即令  $\lambda=0$ ,则有  $EAT=a \times t+(1-a) \times [t+b \times (\epsilon+t)+(1-b) \times t]$ 。

## 2.2 案例分析

在某页式虚拟系统中,假定访问内存的时间是 10 ms,快表的命中率为 90%,平均缺页中断处理为 25 ms,平均缺页中断率为 5%,试计算在不考虑快表访问时间的情况下,该虚存系统的平均有效访问时间是多少?

本案例只考虑快表的命中率和缺页率,忽略快表访问时间。 $a=90\%$ , $b=5\%$ ,应用上面公式可得平均有效访问时间为

$$90\% \times 10 \text{ ms} + (1-90\%) \times [10 \text{ ms} + 5\% \times$$

$$(25 \text{ ms} + 10 \text{ ms}) + (1-5\%) \times 10 \text{ ms}] = 11.125 \text{ ms}。$$

## 3 结论

本文针对分页存储管理中的内存有效访问时间的计算问题,对基本分页存储系统中有无快表,请求分页存储系统中快表是否命中、是否发生缺页等多种情形下的有效访问时间组成分别进行了探讨,总结了多种情况下内存有效访问时间的计算公式,并通过案例加以分析应用。通过分析,使分页存储系统中的进程执行过程和地址变换过程更加清晰,可以帮助学生加深对操作系统中存储器管理的理解。当然,操作系统的类型非常多,存储器的管理方式在硬件和软件上也存在着很大差别。存储器的访问是一个很抽象的概念,上机实验也很难直观展现存储器的访问过程。在讲授这部分内容时可借助课件和动画演示,帮助学生理解。

## 参 考 文 献:

- [1] 刘铁武. 请求分页式系统设计的性能保证[J]. 湖南工程学院学报, 2011, 21(2): 46-48.
- [2] 周苏, 金海溶, 王文, 等. 操作系统原理[M]. 北京: 机械工业出版社, 2013: 177-178.
- [3] 汤小丹, 梁红兵, 哲凤屏, 等. 计算机操作系统[M]. 4 版. 西安: 西安电子科技大学出版社, 2014: 141-142.
- [4] 孙琳, 刘志丹. 嵌入式操作系统内存管理分析与探讨[J]. 科技信息, 2012(24): 278-279.
- [5] Abraham Silberschatz, Peter Baer Galvin, Greg Gagne. 操作系统概念[M]. 7 版. 郑扣根译. 北京: 高等教育出版社, 2010: 297-298.
- [6] 孙钟秀. 操作系统教程[M]. 4 版. 北京: 高等教育出版社, 2008: 272-273.

## Calculation of Effective Access Time in Paging Storage Management System

XIA Chun-mei

(Department of Information Engineering, Binzhou University, Binzhou 256603, China)

**Abstract:** Memory management is one of the five major functions of the operating system, and it is also an important part of the operating system course teaching. Memory is a very important resource in a computer system, and any program must be loaded into memory to run. Aiming at the problem of calculation of effective access time in paging storage management system, the different situations are discussed in paging storage management system and paging request storage management system, and the calculation formulas of effective access time under various conditions are summed up to make the students better grasp the basic principles of the paging system. Some cases are analyzed in this paper to make students more clear of the implementation process in paging storage management system.

**Key words:** memory; page-table; associative memory; effective access time; calculation

(责任编辑: 贾晶晶)