VAX/VMS 操作系统

虚存管理系统

国防科技大学研究所六〇二教研室

一九八五年六月

目　　录

前　言

VAX是数字设备公司（DEC）设计的32位小型机系列，它有大量的虚存空间和较强的功能。

硬件价格大幅度下降使计算机的运用日益广泛。如今操作系统已发展到分布式阶段。这种局面要求小型机上要有较强的功能。在这种情形下，DEC公司计划在原来使用已十分广泛的小型机PDP—11系列上进行功能扩充，但是这样做面临了一个十分棘手的问题，即：原有的存储量不是以应付操作系统中大量功能的渗入，因此必须对PDP—11进行大规模的存储扩充。设计VAX系列的主要目的正在于此。VAX就是英文Virtual Address eXtension的缩写。但是VAX并不是简单地对PDP—11进行空间扩充，由于两种系列字长的不同（PDP—11是16位，VAX是32位），因此VAX的设计完全是新的。它与PDP—11兼容，有一套独特的指令，为高级语言的编译与执行提供了有利的环境，并且提供了大量的虚地址空间。VAX—11/780是使得VAX系列闻名于世的第一种机器。VAX/VMS是VAX—11/780上配置的操作系统，它具有较强的功能。这些功能在该系统问世前一般仅为大型机所拥有。VAX/VMS的内核由三大部分组成，虚存管理，处理机及通讯管理和I/D子系统。虚存管理系统在该系统中的作用十分重要，它的性能直接关系到整个系统的性能。VMS就是英文。Virtual Management System的缩写，这也说明了虚存子系统在该系统中的核心作用。该系统的虚存管理系统基本上是成功的，它既照顾到小型机CPU速度慢，设备少，因此虚存子系

统中的各程序的执行时间要短，并且开销要低的特点，又用简单的方法使得该系统的效率有了显著提高。本文重点介绍VAX/VMS的虚存管理系统。

VMS 的虚存管理子系统由两部分组成Pager和SWapper。Pager 用于处理页故障。Pager 在调用它的进程的关系下执行。Pager 程序被系统中所有的进程共享。SWapper用于将整个进程调入或调出主存。以使具有高优先级的进程总在主存中。

1、页面调度（Pager）

VMS 对系统中的每个进程都给出了一个最多可占用主存量的限额，"驻留集上限"，进程运行时若访问的页不在驻留集中则出现页故障，这时该进程停止执行，待故障解决完后才能继续执行。程序开始执行时驻留集是空的，出现页故障后控制转到Pager，Pager 根据出故障的虚地址找到故障页的页表项，页表项中包含了寻找故障页在辅存中的位置所需的各种信息，然后Pager 找一页可以使用的主存页，并将故障页中的信息从辅存读到主存页中，并修改页表项，使页表项指向该主存页。这时这一页便已加入驻留集了。最后控制回到出页故障的进程，程序继续执行的过程中，凡出现页故障便进行同样的处理。这时驻留集不断加大，驻留集达到上限后，若又出现页故障，则在把故障页加入到驻留集之前必须从驻留集中淘汰一页出去。该系统使用的是修改了的先进先出淘汰原则，虽然先进先出是低效方法，但是经过一些巧妙的修改，该方法的效果已十分接近高耗费的最近最少使用方法。该系统中有两个链表，自由链表和修改链表。当进程需要一主存页时，便从自由链表头取走一页。当驻留集已满，必须淘汰一页出驻留集时，Pager 并不

马上把这一页放到辅存。而是根据该页是否被修改过而将其挂入自由链表（没被修改过）或修改链表（被修改过）的尾端。若淘汰的页是马上又要用到的页，则该页一般来讲还在自由或修改表中，这时把故障页从链表中取出来即可。这样省去了从辅存读页的大量时间。正是由于采用了这种方法才使得页故障率大大减少。进程执行时若要淘汰那些被修改过的页则要将这些页回写到辅存。但 Pager 并不每次都回写修改页。它将修改过的页放在修改链表中等到一定的时机一次将这些页写回辅存。这样大大减少了回写时间。并且，由于有的页可能因又出故障已从链表中取出因此免去了一些回写工作。由于过程开始执行时，驻留集是空的，因此故障会频繁出现，为了避免这种情况。该系统采用了成组读入的方法。在驻留集未满的情况下若出现页故障则可将与故障页相邻的若干页组成一束一次读入主存。该系统还提供了动态地调节驻留集大小的功能。

2、交换（SWapper）

驻留在主存中的所有进程的驻留集组成了系统的平衡集，该系统并不把系统中所有进程的驻留集都放在主存中，而只将那些优先级高的最有可能立刻活动的进程放在平衡集中 SWapper 的功能就是在辅助和平衡集中进行以进程为单位的调度。交换时 SWapper 将整个进程的驻留集作为一个单位进行入交换或出交换。这一点不同于其它系统。许多系统进行交换时用的是逐页换入或换出的方法，这样做需要不断请求 I／O，非常费时间。VMS 在辅存中设置了一个交换文件。交换时 SWapper 将整个驻留集中的页集中起来。只请求一次 I／O 活动。这大大地提高了效率。VMS 给每个驻留进程一个驻留时间片，这样对避免驻留进程没得到运行就被换

出主存有一定的好处。SWapper确定下一换入进程的方法十分直观，它选择非驻留进程中优先级最高的进程进入主存。另外SWapper还要保持内存中有足够的空间供换入进程使用。其方法是将自由链表维持在一定的长度之上。当自由链表中的页低于某一限度时，则回写修改链表中的页。回写过的页链入自由表中，若自由表仍旧低于下限，则在驻留集中选出活动性最小的进程将其换出。以腾出空间充实自由链表。

以下各章节详细介绍VMS的虚存管理子系统。

第一章　Pager

1·1　虚拟地址空间

VAX—11/780采用的是多虚拟存储系统。即为每个进程提供一个虚地址空间，操作系统驻留在一部分虚地址空间中，由各个进程共享使用。本系统的地址长度为32位因此可寻址的空间约为43亿字节。这43亿字节的虚地址空间被分成了两个相同大小部分。低半部称为“进程空间”是系统中运行的各个进程专用的，高半部分称为“系统空间”由所有进程共享。“进程空间”和“系统空间”又各分为两个区域“进程空间”的低半部分称为进程空间的程序区域（简称P0区域），其中包括由进程执行的本机或兼容方式映象以及包括由映象所访问的附加用户码，进程空间的高半部分称为进程的控制区域（简称P1区域），其中包括由系统保持的进程相关信息以及包括诸如内核栈，执行栈，管理栈和用户栈等进程控制结构和进程I／O数据集。系统空间的低半部称为系统区域，其中包括VMS执行程序和为控制进程，为保持系统内所有物理页和虚拟页的状态所要求的那些软件数据结构，系统空间的高半部分称为保留区域保留未用。见图1—1

VMS采用段页式存储管理方法。它将虚地址空间和物理地址空间均以512字节为单位划分，每一单位称作一个页面，页面成为地址浮动和存储保护的基本单位。

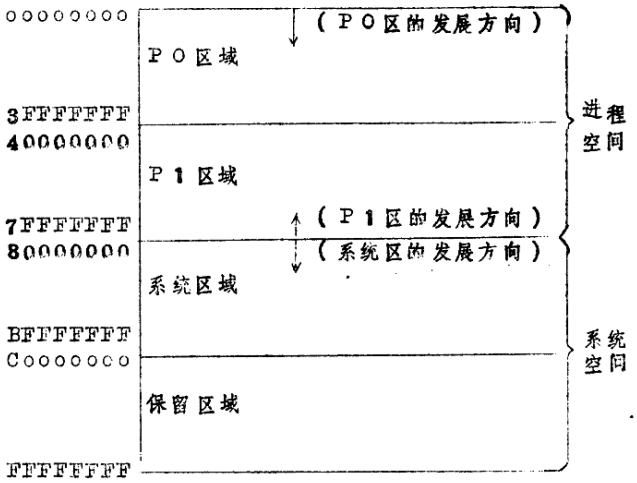

图1—1 虚地址空间的布局

VAX—11/780的虚地址结构如图1—2所示。其中

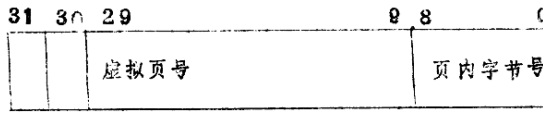

图1—2 虚地址结构

VA(31)VA(30)用于区别空间区域：

VA(31)	(30)	空间区域
0	0	P0
0	1	P1
1	0	系统
1	1	保留区

VA（29）～VA（9）表示该区域中的相对页号。VA（8）～VA（0）表示页内字节号。

1·2 地址转换机构

1·2·1 有关数据结构

VAX—11/780上的物理存储容量一般为2M～8M字节，这比起43亿字节的虚空间容量要小得多。因此处理器不能直接按虚拟地址形式来寻址，必须先把虚地址转换成物理地址。VAX—11/780由硬件提供这种地址转换机构，同时硬地址转换机构还提供进程所属页面间的保护。

页表是联接虚实地址间的主要桥梁，所有虚拟地址到实际地址之间的转换都要借助于页表由硬地址转换机构（或软件模仿硬地址转换机构的工作原理）来实现。系统中共有三个页表存在——P0页表（POPT），P1页表（P1PT）和系统页表（SPT），它们分别映射P0区域，P1区域和系统区域。页表是由页表项组成的，系统页表常驻内存。P1页表和P0页表属于系统空间中的虚页，它们又要由系统页表项来映射，因此它们可能在辅存也可能驻在主存。

每个页表都由基地址寄存器和长度寄存器来定义，基地址寄存器用于确定页表的位置，长度寄存器用于做有效性检查，以确保任何一个给定的虚拟页面都在定义的页表项目范围内。

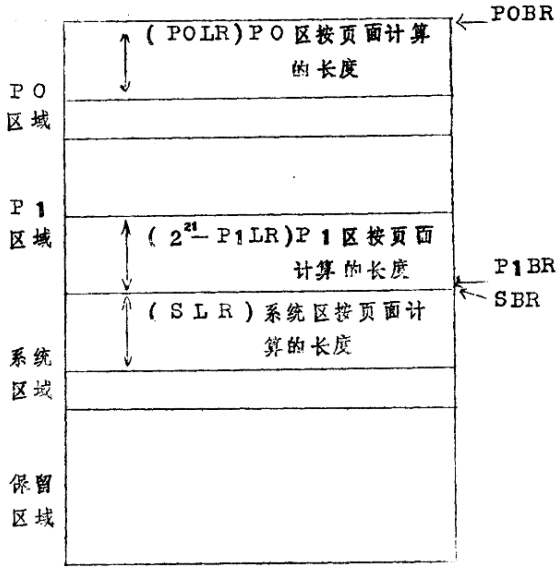

图1－3　虚空与各区域寄存器

SBR：系统页表在物理存储中的起始地址。

SLR：系统区域按页面计算的长度。

POBR：P0页表在系统空间中的起始虚地址。

POLR：P0区域按页面计算的长度。

P1BR：P1页表在系统空间中的起始虚地址。

P1LR：P1区域中没有使用的页面数目。P1区域中按页面计算的实际长度为2^{21}——P1LR。这样表示是为了以后计算方便。因为P1空间是反方向（由高地址向低地址）发展的。为了使得硬件对各基地址寄存器和长

度寄存器的解释能保持一致所以P1LR指的是P1区域中不能访问的部分。

各寄存器的实际表示如图1—4所示：

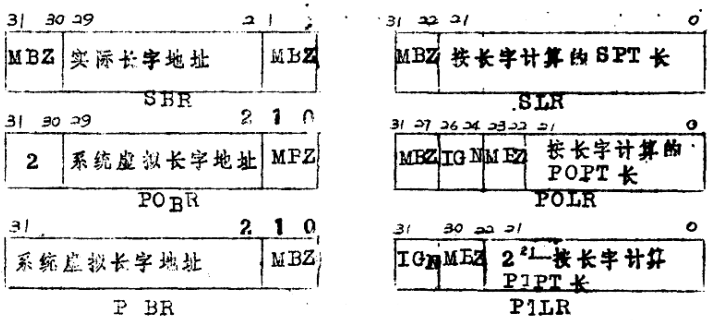

图1—4 各区域的寄存器

注：MBZ是一定为0的意思。

虚地址空间中的每一页在相应页表中都有一页表项与这对应。当一页正在驻留集中被使用时，其页表项的内容如图1—5所示：

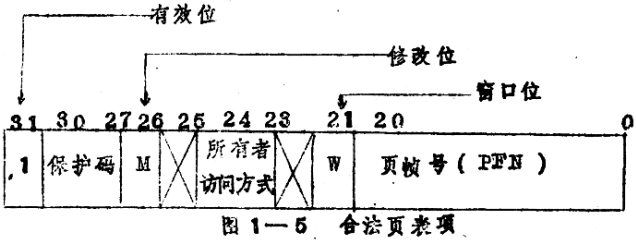

图1—5 合法页表项

其中：有效位置1表示该项对应的页在驻留集中。页帧号表示对应的物理页号。即该页在内存的实际起始地址。其它位以后给予说明。

硬件地址转换机构就是通过上述基地寄存器长度寄存器和页表项来完成从虚地址到物理地址的转换。下面描述它的工作原理。

1·2·2 地址转换的逻辑过程

对任一虚地址（如图1－2）VA。首先检查VA的高两位VA（31）VA（30）。若VA（31）VA（30）＝10则表明该虚地址属于系统虚空间，这首先检查页号超出对应的空间没有。即VA（29）～VA（9）是否大于SLR。若超出，则产生长度违章故障否则按SBR和VA（29）～VA（9）就可找到对应该页的页表项。若该页表项的有效位为1则页表项的PFN部分就是该页在物理存储器中的页号。用PFN和VA（8）～VA(0)就可定出所要的实际地址。若有效位为0则说明该页不在驻留集中，这时硬件发出页故障中断。若VA（31）VA（30）＝00或01则说明该虚地址属于P0空间或P1空间。由于P0页表和P1页表属于系统空间中的虚页。因此首先要通过系统页表来定位它们。若页表本身不在驻留集中则会产生故障待页表项的故障解决后才能根据它最后定出所要的实际地址。图1－6给出了地址转换的逻辑流程。其中VA表示虚地址．PA表示转换后得到的实际地址。VA—POPTE表示P0页表项的虚地址。PA—POPTE表示P0页表项的实际地址。余此类推。表示连接操作。X(a:b)表示从X的第a位到第b位。

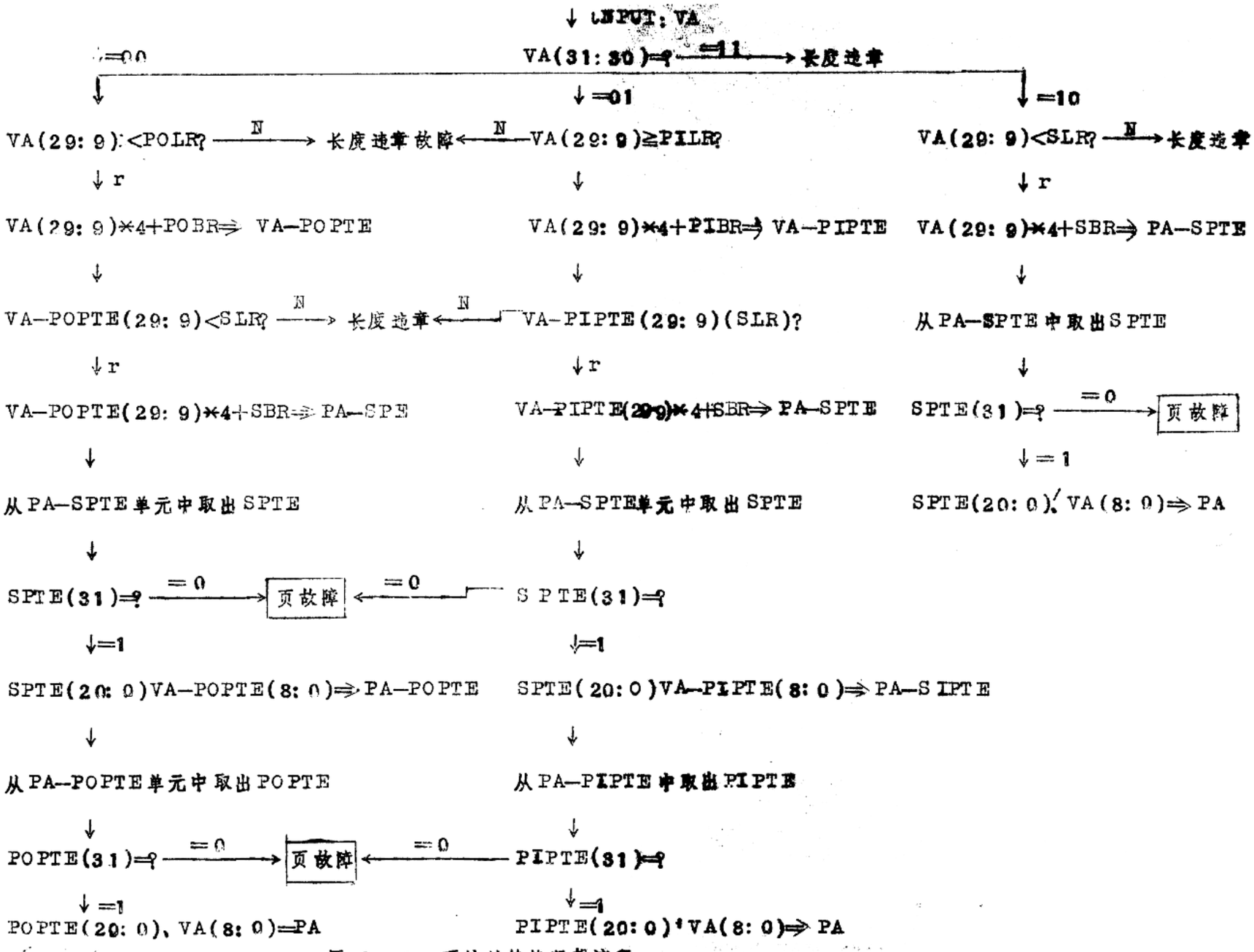

图 1—6　硬地址转换逻辑流程

1·2·3 几点说明

(1) 并非任何时候都需要进行虚实地址的转换。系统中有一个用于允许和禁止存储管理的特权寄存器。叫做“允许代换寄存器”（MAPEN）。如图1－7所示：

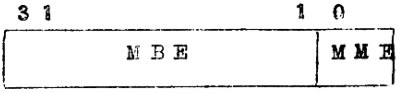

图1－7 允许代换寄存器

当MME为0时表示虚地址的若干低阶位（具体位数因机种而异）就是实际地址。此时不需要进行虚实代换，无页面保护。当MME为1时表示需要进行地址转换。

(2) 为了提高系统的可靠性，硬件还提供了页面保护。该系统用访问方式提供了四级层次的存储访问特权，它们由高到低分别是：核方式，执行方式，管理方式和用户方式。每个页面均有一个保护码（页表项中的第30到27位见图1－5）用来说明访问该页至少应该具有的访问方式。一个页面可以是不可访问的，只读的和可读／写的。实现保护的原则是：

a） 该页在某一级上是可读（或可读／写）的，则该页在具有更高特权的级上也是可读（或可读／写）。

b） 该页在某一级上可进行读（或读／写）访问，则在更低级上不可进行读（或读／写）访问。

这样就保证了在较高特权方式上运行的进程不被低特权方式上运行的进程所破坏。