

是相同的。装入64个512字节的页面可能需要 $64 \times 10\text{ms}$ ，而装入4个8KB的页面可能只需要 $4 \times 12\text{ms}$ 。

在某些机器上，每次CPU从一个进程切换到另一个进程时都必须把新进程的页表装入硬件寄存器中。这样，页面越小意味着装入页面寄存器花费的时间就会越长，而且页表占用的空间也会随着页面的减小而增大。

最后一点可以从数学上进行分析，假设进程平均大小是 s 个字节，页面大小是 p 个字节，每个页表项需要 e 个字节。那么每个进程需要的页数大约是 s/p ，占用了 se/p 个字节的页表空间。内部碎片在最后一页浪费的内存是 $p/2$ 。因此，由页表和内部碎片损失造成的全部开销是以下两项之和：

$$\text{开销} = se/p + p/2$$

在页面比较小的时候，第一项（页表大小）大。在页面比较大时第二项（内部碎片）大。最优值一定在页面大小处于中间的某个值时取得，通过对 p 一次求导并令右边等于零，我们得到方程：

$$-se/p^2 + 1/2 = 0$$

从这个方程可以得出最优页面大小的公式（只考虑碎片浪费和页表所需的内存），结果是：

$$P = \sqrt{2se}$$

对于 $s = 1\text{MB}$ 和每个页表项 $e = 8$ 个字节，最优页面大小是4KB。商用计算机使用的页面大小一般在512字节到64KB之间，以前的典型值是1KB，而现在更常见的页面大小是4KB或8KB。随着存储器越来越大，页面也倾向于更大（但不是线性的）。把RAM扩大4倍极少会使页面大小加倍。

3.5.4 分离的指令空间和数据空间

大多数计算机只有一个地址空间，既存放程序也存放数据，如图3-25a所示。如果地址空间足够大，那么一切都好。然而，地址空间通常太小了，这就使得程序员对地址空间的使用出现困难。

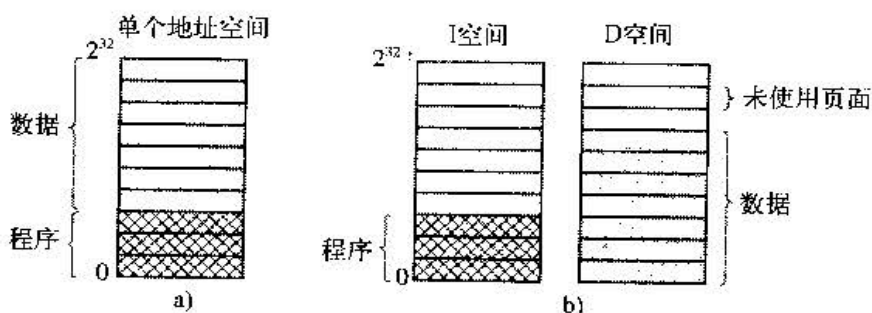


图3-25 a) 单个地址空间，b) 分离的I空间 and D空间

首先在PDP-11（16位）上实现的一种解决方案是，为指令（程序正文）和数据设置分离的地址空间，分别称为I空间和D空间，如图3-25b所示。每个地址空间都从0开始到某个最大值，比较有代表性的是 $2^{16}-1$ 或者 $2^{32}-1$ 。链接器必须知道何时使用分离的I空间和D空间，因为当使用它们时，数据被重定位到虚拟地址0，而不是在程序之后开始。

在使用这种设计的计算机中，两种地址空间都可以进行分页，而且互相独立。它们分别有自己的页表，分别完成虚拟页面到物理页框的映射。当硬件进行取指令操作时，它知道要使用I空间和I空间页表。类似地，对数据的访问必须通过D空间页表。除了这一区别，拥有分离的I空间和D空间不会引入任何复杂的设计，而且它还能使可用的地址空间加倍。

3.5.5 共享页面

另一个设计问题是共享。在大型多道程序设计系统中，几个不同的用户同时运行同一个程序是很常见的。显然，由于避免了在内存中有一个页面的两份副本，共享页面效率更高。这里存在一个问题，即并不是所有的页面都适合共享。特别地，那些只读的页面（诸如程序文本）可以共享，但是数据页面则不能共享。

如果系统支持分离的I空间和D空间，那么通过让两个或者多个进程来共享程序就变得非常简单了，