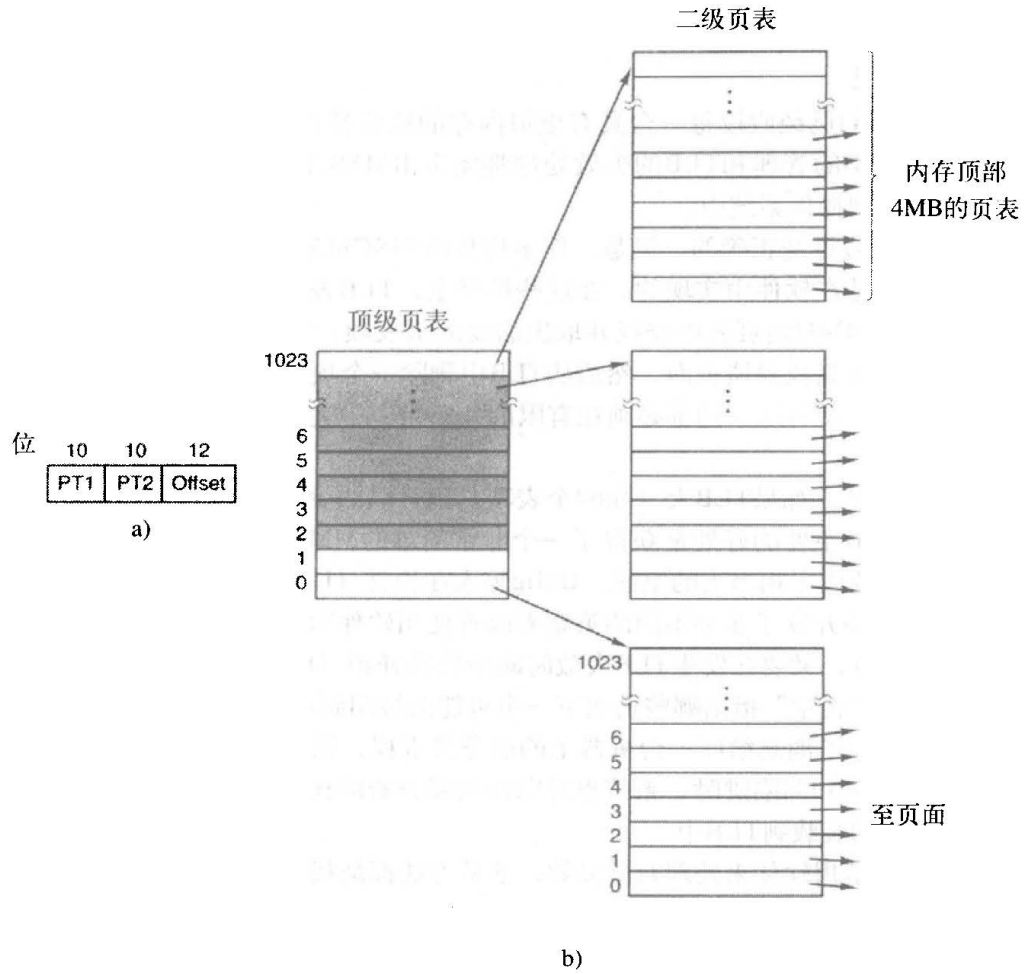

图3-13 a) 一个有两个页表域的32位地位；b) 二级页表

由索引顶级页表得到的表项中含有二级页表的地址或页框号。顶级页表的表项0指向程序正文的页表，表项1指向数据的页表，表项1023指向堆栈的页表，其他的表项（用阴影表示的）未用。现在把PT2域作为访问选定的二级页表的索引，以便找到该虚拟页面的对应页框号。

下面看一个示例，考虑32位虚拟地址0x00403004（十进制4 206 596）位于数据部分12 292字节处。它的虚拟地址对应PT1＝1，PT2＝2，Offset＝4。MMU首先用PT1作为索引访问顶级页表得到表项1，它对应的地址范围是4M~8M。然后，它用PT2作为索引访问刚刚找到的二级页表并得到表项3，它对应的虚拟地址范围是在它的4M块内的12 288~16 383（即绝对地址4 206 592~4 210 687）。这个表项含有虚拟地址0x00403004所在页面的页框号。如果该页面不在内存中，页表项中的“在/不在”位将是0，引发一次缺页中断。如果该页面在内存中，从二级页表中得到的页框号将与偏移量（4）结合形成物理地址。该地址被放到总线上并送到内存中。

值得注意的是，虽然在图3-13中虚拟地址空间超过100万个页面，实际上只需要四个页表：顶级页表以及0~4M（正文段）、4M~8M（数据段）和顶端4M（堆栈段）的二级页表。顶级页表中1021个表项的“在/不在”位都被设为0，当访问它们时强制产生一个缺页中断。如果发生了这种情况，操作系统将注意到进程正在试图访问一个不希望被访问的地址，并采取适当的行动，比如向进程发出一个信号或杀死进程等。在这个例子中的各种长度选择的都是整数，并且选择PT1与PT2等长，但在实际中也可能是其他的值。

图3-13所示的二级页表可扩充为三级、四级或更多级。级别越多，灵活性就越大，但页表超过三级会带来更大的复杂性，这样做是否值得令人怀疑。

2. 倒排页表

对32位虚拟地址空间，多级页表可以很好地发挥作用。但是，随着64位计算机变得更加普遍，情况