

含着段，一些则成了空闲区，这种现象称为棋盘形碎片或外部碎片 (external fragmentation)。空闲区的存在使内存被浪费了，而这可以通过内存紧缩来解决。如图3-34c所示。

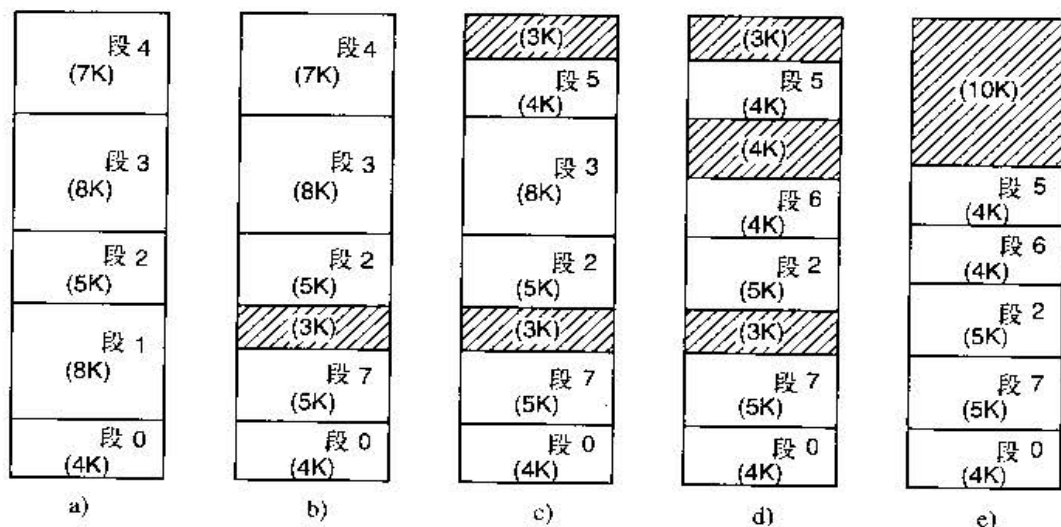


图3-34 a)-d)棋盘形碎片的形成；e)通过紧缩消除棋盘形碎片

3.7.2 分段和分页结合：MULTICS

如果一个段比较大，把它整个保存在内存中可能很不方便甚至是不可能的，因此产生了对它进行分页的想法。这样，只有那些真正需要的页面才会被调入内存。有几个著名的系统实现了对段进行分页的支持，在本节我们将介绍第一个实现了这种支持的系统——MULTICS。在下一节我们将介绍一个更新的例子——Intel Pentium。

MULTICS运行在Honeywell 6000计算机和它的一些后继机型上。它为每个程序提供了最多 2^{18} 个段（超过250 000个），每个段的虚拟地址空间最长为65 536个（36位）字长。为了实现它，MULTICS的设计者决定把每个段都看作是一个虚拟内存并对它进行分页，以结合分页的优点（统一的页面大小和在只使用段的一部分时不用把它全部调入内存）和分段的优点（易于编程、模块化、保护和共享）。

每个MULTICS程序都有一个段表，每个段对应一个描述符。因为段表可能会有大于25万个的表项，段表本身也是一个段并被分页。一个段描述符包含了一个段是否在内存中的标志，只要一个段的任何一部分在内存中这个段就被认为是在内存中，并且它的页表也会在内存中。如果一个段在内存中，它的描述符将包含一个18位的指向它的页表的指针（见图3-35a）。因为物理地址是24位

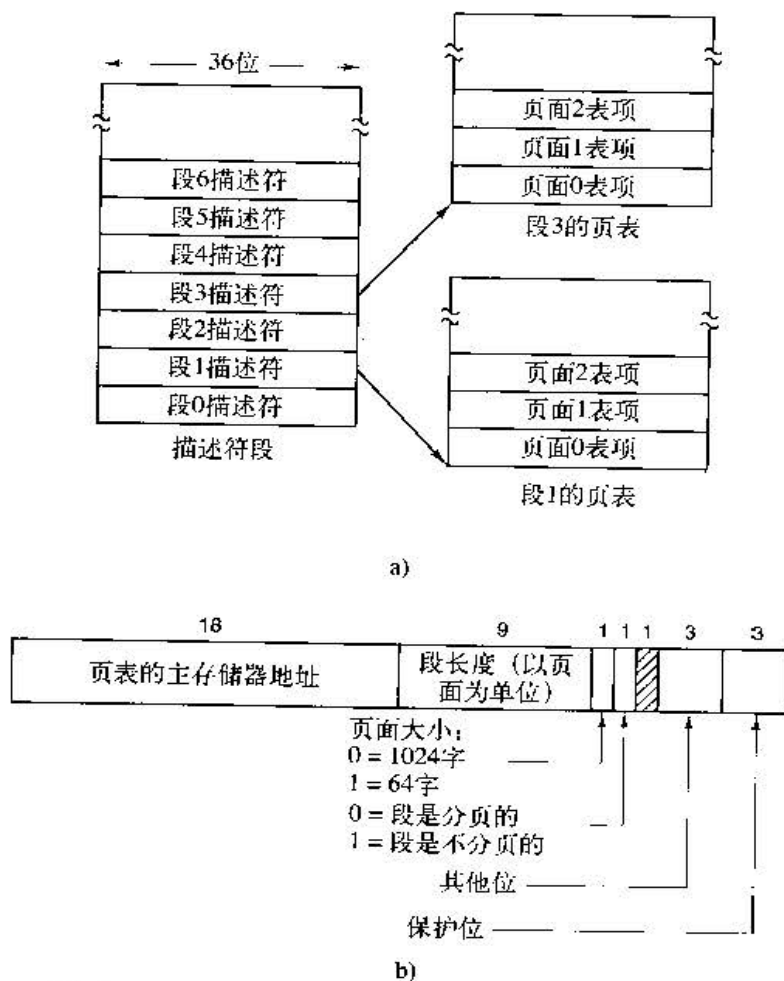


图3-35 MULTICS的虚拟内存：a) 描述符段指向页表；
b) 一个段描述符，其中的数字是各个域的长度