

可以有效地选择要使用的页表。类似地，在需要的时候可以使用三级分页，此时把上级目录域的大小设置为0就可以了。

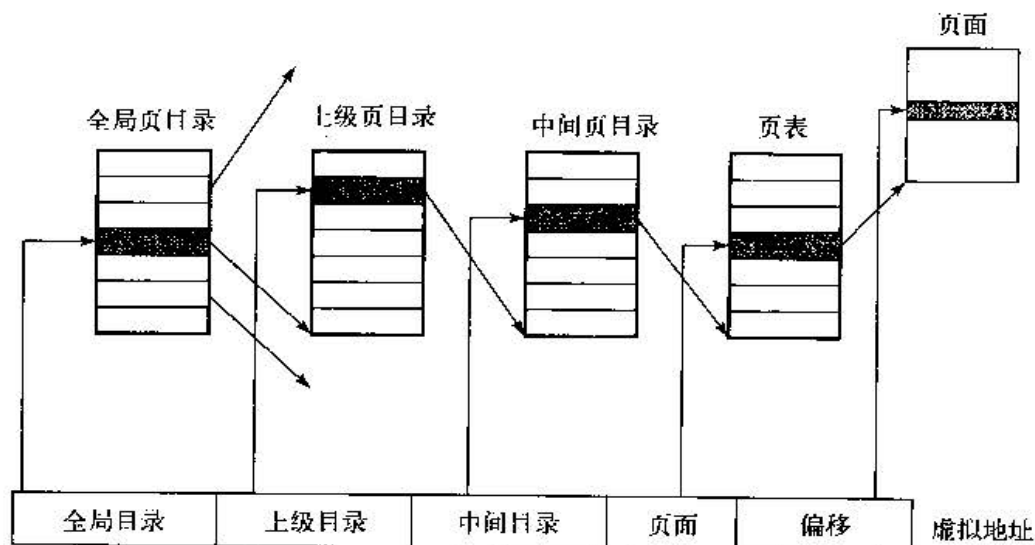


图10-16 Linux使用四级页表

物理内存可以用于多种目的。内核自身是完全“硬连线”的，它的任何一部分都不会换出。内存的其余部分可以作为用户页面、分页缓存和其他目的。页面缓存包含最近已读的或者由于未来有可能使用而预读的文件块，或者需要写回磁盘的文件块页面，例如那些被换出到磁盘的用户进程创建的页面。分页缓存并不是一个独立的缓存，而是那些不再需要的或者等待换出的用户页面集合。如果分页缓存当中的一个页面在被换出内存之前复用，它可以被快速收回。

此外，Linux支持动态加载模块，最常见的是设备驱动。它们可以是任意大小的并且必须分配一个连续的内核内存。这些需求的一个直接结果是，Linux用这样一种方式来管理物理内存使得它可以随意分配任意大小的内存片。它使用的算法就是伙伴算法，下面给予描述。

2. 内存分配机制

Linux支持多种内存分配机制。分配物理内存页框的主要机制是页面分配器，它使用了著名的伙伴算法。

管理一块内存的基本思想如下。刚开始，内存由一块连续的片段组成，图10-17a的简单例子中是64个页面。当一个内存请求到达时，首先上舍入到2的幂，比如8个页面。然后整个内存块被分割成两半，如图b所示。因为这些片段还是太大了，较低的片段被再次二分（c），然后再二分（d）。现在我们有了一块大小合适的内存，因此把它分配给请求者，如图d所示。

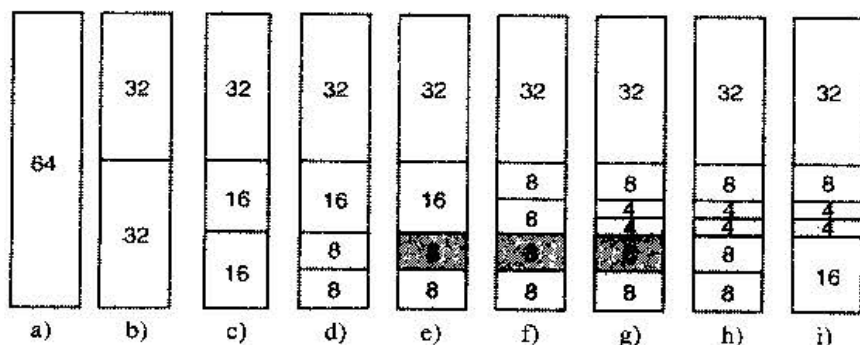


图10-17 伙伴算法的操作

现在假定8个页面的第二个请求到达了。这个请求有（e）直接满足了。此时4个页面的第三个请求到达了。最小可用的块被分割（f），然后其一半被分配（g）。接下来，8页面的第二个块被释放（h）。最后，8页面的另一个块也被释放。因为刚刚释放的两个邻接的8页面块来自同一个16页面块，它们合并