

来减少所需的寻道时间。进程预约式页面调度 (prepaging) 技术也用到了系统启动、把后台应用程序移到前台以及休眠之后重启系统当中。

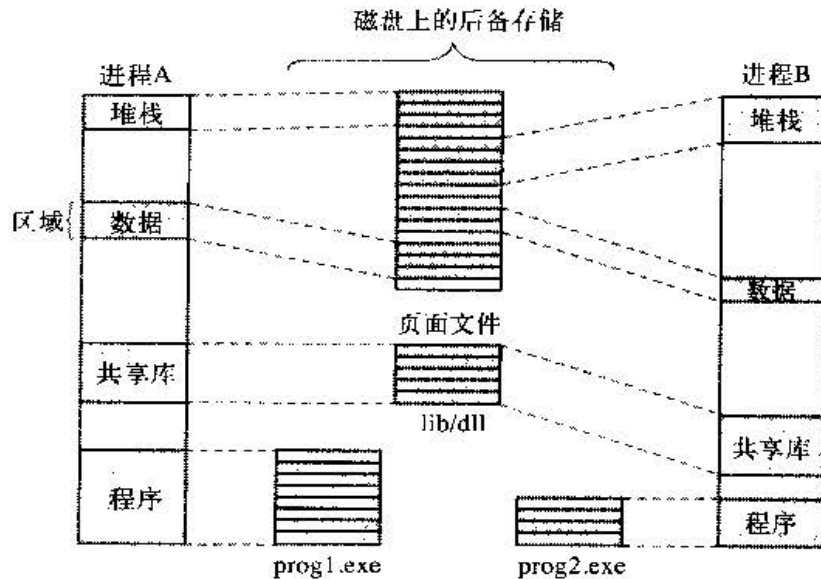


图11-32 被映射的区域以及它们在磁盘上的“影子”页面。lib.dll 文件被同时映射到两个地址空间中

存储管理器支持预约式页面调度,但是它被实现成系统中一个单独的组件。被读入到内存的页面不是插入到进程的页表中,而是插入到后备列表中,从而使得在需要时可以不访问磁盘就将它们插入到进程中。

未被映射的页面稍微有些不同。它们没有被通过读取文件来初始化。相反,一个未被映射的页面第一次被访问的时候,存储管理器会提供一个新的物理页面,该页面的内容被事先清零(为了安全方面的原因)。在后续的页面失效处理过程中,未被映射的页面可能会被从内存中找到,否则的话,它们必须被从页面文件中重新读入内存。

存储管理器中的按需分页是通过页面失效来驱动的。在每次页面失效发生的时候,会发生一次到内核的陷入。内核将建立一个说明发生了什么事情的机器无关的描述符,并把该描述符传递给存储管理器相关的执行部件。存储管理器接下来会检查引发页面失效的内存访问的有效性。如果发生页面失效的页面位于一个已提交的区域内,存储管理器将在VAD列表中查找页面地址并找到(或创建)进程页表项。对于共享页面的情况,存储管理器使用与内存区对象关联的原始页表项来填写进程页表中的新页表项。

不同处理器体系结构下的页表项的格式可能会不同。对于x86和x64,一个被映射页面的页表项如图11-33所示。如果一个页表项被标记为有效,它的内容会被硬件读取并解释,从而虚拟地址能够转换成正确的物理地址。未被映射的页面也有对应的页表项,但是这些页表项被标记成无效,硬件将忽略这些页表项除该标记之外的部分。页表项的软件格式与硬件格式有所不同,软件格式由存储管理器决定。例如,对于一个未映射的页面,它必须在使用前分配和清零,这一点可以通过页表项来表明。

页表项中有两个重要的位是直接由硬件更新的,它们是访问位(access bit)和脏位(dirty bit)。这两个位跟踪了什么时候一个特定的页面映射用来访问该页面以及这个访问是否以写的方式修改了页面的内容。这确实很有助于提高系统性能。因为存储管理器可以使用访问位来实现LRU(Least-Recently Used, 最近最少使用)类型的页面替换策略。LRU原理是,那些最长时间没有被使用过的页面有最小的可能性在不久的将来被再次使用。访问位使存储管理器知道一个页面被访问过了,脏位使存储管理器知道一个页面被修改了,或者更重要的是,一个页面没有被修改。如果一个页面自从从磁盘上读到内存后没有被修改过,存储管理器就没有必要在将该页面用到其他地方之前将页面内容写回磁盘了。

正如表11-33所示,x86体系结构通常使用32位大小的页表项,而x64体系结构使用64位大小的页表项。在域上面的唯一区别是x64的物理页号域是30位,而不是20位。然而,现今存在的任何x64处理器所支持的物理页面的数量都要远小于x64体系结构所能表示的数量。x86体系结构也支持一种特殊的物理地址扩展(Physical Address Extension, PAE)。PAE模式允许处理器访问超过4GB的物理内存,附加的物理页框位要求PAE模式下的页表项也是64位。