

久性存储介质共享,并且通常以两种方法中的一种进行管理。首先,一些操作系统采用一种十分简单的方法,内存根本不分页。在这些类型的系统中,上下文切换意味着分配操作空间(比如堆空间),同时所有进程间共享这些操作空间。这种方法在进程的存储区域几乎没有保护,信任进程间可以很好地工作。Palm操作系统使用这种简单的方式进行内存管理。第二种方法是使用一种更加有规则的方法,内存被切分成为页,这些页按照操作需要分配。操作系统管理一个空闲列表来保存页,按照需要分配给操作系统和用户进程。在这种方法中(由于没有虚拟内存),当页的空闲列表用光时,系统就会没有内存,从而不会再有分配发生。Symbian操作系统是第二种方法的例子。

#### 12.4.2 Symbian操作系统的寻址方式

由于Symbian操作系统是32位系统,因此寻址范围可以达到4GB。它与更大的系统一样使用同样的抽象方式:程序必须使用由操作系统映射到物理地址的虚拟地址。和大多数系统一样,Symbian操作系统把内存划分为虚拟页面和物理页框。页框的大小通常是4KB,但也是可变的。

因为最大具有4GB的地址空间,因此4KB的页框大小就意味着具有超过100万条目的页表。Symbian操作系统只有有限的内存,因此不能拿出1MB内存专用于页表。而且,对这么大的一张表的搜索和访问对系统都是很大的负担。为了解决这个问题,Symbian操作系统采用2级页表方式,如图12-2所示。称作页面目录的第一级提供一个到第二级的链接,可以使用虚拟地址的一部分进行检索(前12位)。该目录驻留在内存中,由TTBR(转换表基址寄存器)指向。每个目录条目指向第二级,也就是页表的集合。这些页表提供到某一内存中特定页的链接,由虚拟地址的一部分检索(中间8位)。最后,虚拟地址的低12位索引检索页的字。在这一虚拟-物理地址映射计算中,硬件起辅助作用。尽管Symbian操作系统不能假定任何辅助硬件的存在,但是在大多数体系中这一映射转换都是由MMU完成的。比如ARM处理器就具有扩展的MMU,带有转换后备缓冲器来辅助地址计算。

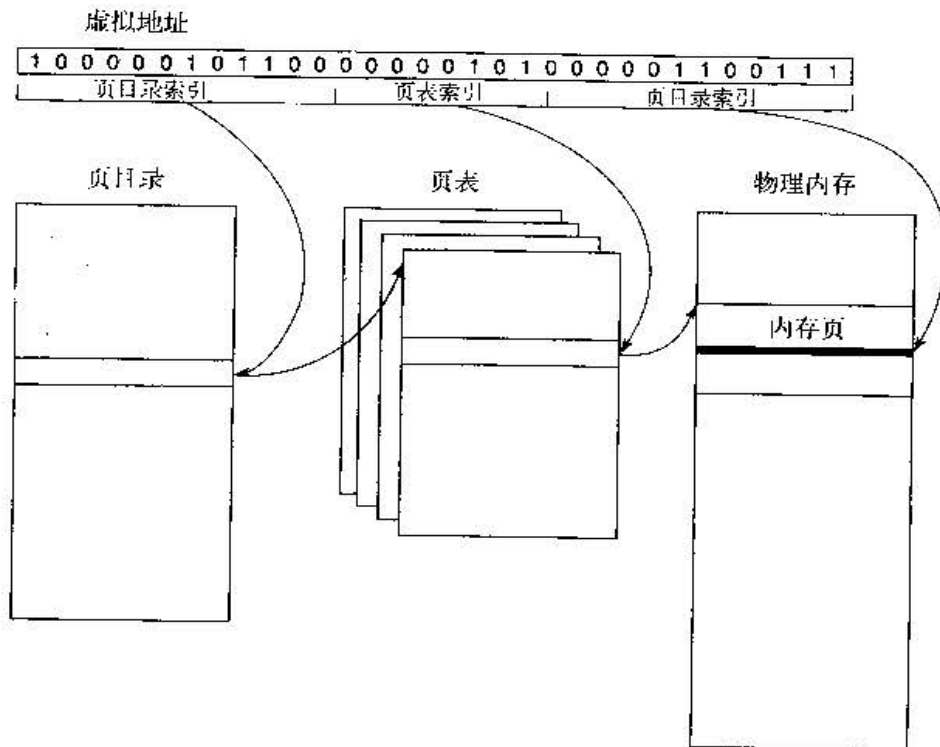


图12-2 Symbian操作系统使用2级页表来减少页表访问时间和占用空间

当一个页面不在内存中时,就会出现错误状态,这是因为当一个应用程序启动的时候所有的应用程序内存页面都应该已经被加载(没有请求页面调度)。链接到可执行应用中的小的桩代码显式地把动态加载库加载到内存中,而不是通过页失效方式。

Symbian操作系统中尽管没有页交换,但内存却不可思议地是动态的。应用程序通过内存进行上下文切换,同时正如上面所说的,当应用程序开始执行时将它们所需要的存储空间加载到内存中。每个应