

缓存和MMU的出现对系统的性能有着重要的影响。在多道程序系统中,从一个程序切换到另一个程序,有时称为上下文切换(context switch),有必要对缓存中来的所有修改过的块进行写回磁盘操作,并修改MMU中的映像寄存器。但是这两种操作的代价很昂贵,所以程序员们努力避免使用这些操作。我们稍后将看到这些操作产生的影响。

1.3.4 磁带

在存储器体系中的最后一层是磁带。这种介质经常用于磁盘的备份,并且可以保存非常大量的数据集。在访问磁带前,首先要把磁带装到磁带上,可以人工安装也可用机器人安装(在大型数据库中通常安装有自动磁带处理设备)。然后,磁带可能还需要向前绕转以便读取所请求的数据块。总之,这一切工作要花费几分钟。磁带的最大特点是每个二进制位的成本极其便宜,并且是可移动的,这对于为了能在火灾、洪水、地震等灾害中存活下来,必须离线存储的备份磁带而言,是非常重要的。

我们已经讨论过的存储器体系结构是典型的,但是有的安装系统并不具备所有这些层次,或者有所差别(诸如光盘)。不过,在所有的系统中,当层次下降时,其随机访问时间则明显地增加,容量也同样明显地增加,而每个二进制位的成本则大幅度下降。其结果是,这种存储器体系结构似乎还要伴随我们多年。

1.3.5 I/O设备

CPU和存储器不是操作系统惟一需要管理的资源。I/O设备也与操作系统有密切的相互影响。如图1-6所示,I/O设备一般包括两个部分:设备控制器和设备本身。控制器是插在电路板上的一块芯片或一组芯片,这块电路板物理地控制设备。它从操作系统接收命令,例如,从设备读数据,并且完成数据的处理。

在许多情形下,对这些设备的控制是非常复杂和具体的,所以,控制器的任务是为操作系统提供一个简单的接口(不过还是很复杂的)。例如,磁盘控制器可以接受一个命令从磁盘2读出11206号扇区,然后,控制器把这个线性扇区号转化为柱面、扇区和磁头。由于外柱面比内柱面有较多的扇区,而且一些坏扇区已经被映射到磁盘的其他地方,所以这种转换将是很复杂的。磁盘控制器必须确定磁头臂应该在哪个柱面上,并对磁头臂发出一串脉冲使其前后移动到所要求的柱面号上,接着必须等待对应的扇区转动到磁头下面并开始读出数据,随着数据从驱动器读出,要消去引导块并计算校验和。最后,还得把输入的二进制位组成字并存放到存储器中。为了要完成这些工作,在控制器中经常安装一个小的嵌入式计算机,该嵌入式计算机运行为执行这些工作而专门编好的程序。

I/O设备的另一个部分是实际设备的自身。设备本身有个相对简单的接口,这是因为接口既不能做很多工作,又已经被标准化了。标准化是有必要的,这样任何一个IDE磁盘控制器就可以适应任一种IDE磁盘,例如,IDE表示集成驱动器电子设备(Integrated Drive Electronics),是许多计算机的磁盘标准。由于实际的设备接口隐藏在控制器中,所以,操作系统看到的是对控制器的接口,这个接口可能和设备接口有很大的差别。

每类设备控制器都是不同的,所以,需要不同的软件进行控制。专门与控制器对话,发出命令并接收响应的软件,称为设备驱动程序(device driver)。每个控制器厂家必须为所支持的操作系统提供相应的设备驱动程序。例如,一台扫描仪会配有用于Windows 2000、Windows XP、Vista以及Linux的设备驱动程序。

为了能够使用设备驱动程序,必须把设备驱动程序装入到操作系统中,这样它可在核心态中运行。理论上,设备驱动程序可以在内核外运行,但是几乎没有系统支持这种可能的方式,因为它要求允许在用户空间的设备驱动程序能够以控制的方式访问设备,这是一种极少得到支持的功能。要将设备驱动程序装入操作系统,有三个途径。第一个途径是将内核与设备驱动程序重新链接,然后重启动系统。许多UNIX系统以这种方式工作。第二个途径是在一个操作系统文件中设置一个入口,并通知该文件需要一个设备驱动程序,然后重启动系统。在系统启动时,操作系统去找寻所需的设备驱动程序并装载之。Windows就是以这种方式工作。第三种途径是,操作系统能够在运行时接受新的设备驱动程序并且立即将其安装好,无须重启动系统。这种方式采用的较少,但是这种方式正在变得普及起来。热插拔设备,诸如USB和IEEE1394设备(后面会讨论)都需要动态可装载设备驱动程序。

每个设备控制器都有少量的用于通信的寄存器。例如,一个最小的磁盘控制器也会有用于指定磁盘地址、内存地址、扇区计数和方向(读或写)的寄存器。要激活控制器,设备驱动程序从操作系统获得一条命令,然后翻译成对应的值,并写进设备寄存器中。所有设备寄存器的集合构成了I/O端口空间,