

在前一种情况下,阻塞的驱动程序可以被中断唤醒。在后一种情况下,驱动程序根本就不会休眠。无论是哪一种情况,操作完成之后驱动程序都必须检查错误。如果一切顺利,驱动程序可能要将数据(例如刚刚读出的一个磁盘块)传送给与设备无关的软件。最后,它向调用者返回一些用于错误报告的状态信息。如果还有其他未完成的请求在排队,则选择一个启动执行。如果队列中没有未完成的请求,则该驱动程序将阻塞以等待下一个请求。

这一简单的模型只是现实的粗略近似,许多因素使相关的代码比这要复杂得多。首先,当一个驱动程序正在运行时,某个I/O设备可能会完成操作,这样就会中断驱动程序。中断可能会导致一个设备驱动程序运行,事实上,它可能导致当前驱动程序运行。例如,当网络驱动程序正在处理一个到来的数据包时,另一个数据包可能到来。因此,驱动程序必须是重入的(reentrant),这意味着一个正在运行的驱动程序必须预料到在第一次调用完成之前第二次被调用。

在一个热可插拔的系统中,设备可以在计算机运行时添加或删除。因此,当一个驱动程序正忙于从某设备读数据时,系统可能会通知它用户突然将设备从系统中删除了。在这样的情况下,不但当前I/O传送必须中止并且不能破坏任何核心数据结构,而且任何对这个现已消失的设备的悬而未决的请求都必须适当地从系统中删除,同时还要为它们的调用者提供这一坏消息。此外,未预料到的新设备的添加可能导致内核重新配置资源(例如中断请求线),从驱动程序中撤除旧资源,并且在适当位置填入新资源。

驱动程序不允许进行系统调用,但是它们经常需要与内核的其余部分进行交互。对某些内核过程的调用通常是允许的。例如,通常需要调用内核过程来分配和释放硬接线的内存页面作为缓冲区。还可能需其他有用的调用来管理MMU、定时器、DMA控制器、中断控制器等。

5.3.3 与设备无关的I/O软件

虽然I/O软件中有一些是设备特定的,但是其他部分I/O软件是与设备无关的。设备驱动程序和与设备无关的软件之间的确切界限依赖于具体系统(和设备),因为对于一些本来应按照与设备无关方式实现的功能,出于效率和其他原因,实际上是由驱动程序来实现的。图5-13所示的功能典型地由与设备无关的软件实现。

与设备无关的软件的基本功能是执行对所有设备公共的I/O功能,并且向用户层软件提供一个统一的接口。下面我们将详细介绍上述问题。

设备驱动程序的统一接口
缓冲
错误报告
分配与释放专用设备
提供与设备无关的块大小

图5-13 与设备无关的I/O软件的功能

1. 设备驱动程序的统一接口

操作系统的一个主要问题是如何使所有I/O设备和驱动程序看起来或多或少是相同的。如果磁盘、打印机、键盘等接口方式都不相同,那么每次在一个新设备出现时,都必须为新设备修改操作系统。必须为每个新设备修改操作系统决不是一个好主意。

设备驱动程序与操作系统其余部分之间的接口是这一问题的一个方面。图5-14a所示为这样一种情形:每个设备驱动程序有不同的与操作系统的接口。这意味着,可供系统调用的驱动程序函数随驱动程序的不同而不同。这可能还意味着,驱动程序所需要的内核函数也是随驱动程序的不同而不同的。综合起来看,这意味着为每个新的驱动程序提供接口都需要大量全新的编程工作。

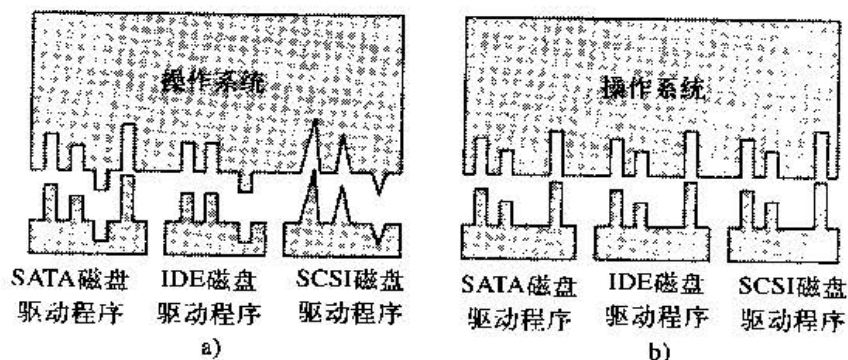


图5-14 a) 没有标准的驱动程序接口; b) 具有标准的驱动程序接口