

## 9.1 I/O管理概述

### 1. I/O 设备

连接到计算机上的设备多种多样。某些设备一次传输一个字或一块字。某些设备只能顺序访问，而有些可以随机访问。某些设备同步传输数据，而有些是异步传输。有些是专用的，而有些则是共享的。有些设备只读而有些则是读写皆可。它们的速度差别很大。不过大部分情况下，它们都是整个计算机系统中最慢的部分。

既然设备之间存在很大差别，操作系统需要为应用程序提供大量功能以控制这些设备。操作系统的 I/O 子系统的关键目的就是为了提供尽可能简洁的接口。并且由于设备是整个计算机系统的一个瓶颈，所以 I/O 子系统的另一个关键是尽可能地进行优化。

### 2. I/O 管理目标

I/O 管理主要有两个目标：

- 提高设备的利用率。应尽量提高 CPU 与 I/O 设备之间的并行操作程度，主要利用的技术有：中断技术、DMA 技术、通道技术、缓冲技术。
- 为用户提供方便、统一的界面。所谓方便，是指用户能独立于具体设备的复杂物理特性之外而方便地使用设备。所谓统一，是指对不同的设备尽量使用统一的操作方式，例如各种字符设备用一种 I/O 操作方式。这就要求用户操作的是简便的逻辑设备，而具体的 I/O 物理设备由操作系统去实现，这种性能常常被称为设备的独立性。

### 3. I/O 管理功能

I/O 管理功能主要有：

- 设备分配。设备管理程序按照一定的算法把某一个 I/O 设备、及其相应的设备控制器分配给某一进程，对于未分配到的进程，则在等待队列中等待。
- 缓冲区管理。为了解决 CPU 与 I/O 之间速度不匹配的矛盾，在它们之间配置了缓冲区。设备管理程序需要负责管理缓冲区的建立、分配和释放等。
- 实现物理 I/O 设备的操作。对于具有通道的系统，设备管理程序根据用户提出的 I/O 请求，生成相应的通道程序并提交给通道，然后用专门的通道指令启动通道，对指定的设备进行 I/O 操作，并能响应通道的中断请求。对于未设置通道的系统，设备管理程序直接驱动设备进行 I/O 操作。

### 4. I/O 应用接口

I/O 系统调用实现统一的 I/O 接口。I/O 系统调用封装了设备通用类型行为。每个通用类型都可以通过一组标准函数（即接口）来访问，如块设备 I/O 系统调用包括磁盘、磁带、光盘等一系列块设备的 read、write、seek。具体的 I/O 差别被内核模块（称设备驱动程序）所封装，这些设备驱动程序一方面可以定制以适合各种设备，另一方面也提供了一组标准接口。图 9.1 说明了内核与 I/O 相关部分是如何按软件层来组织的。

设备驱动程序的作用是为内核 I/O 子系统隐藏设备控制器之间的差异

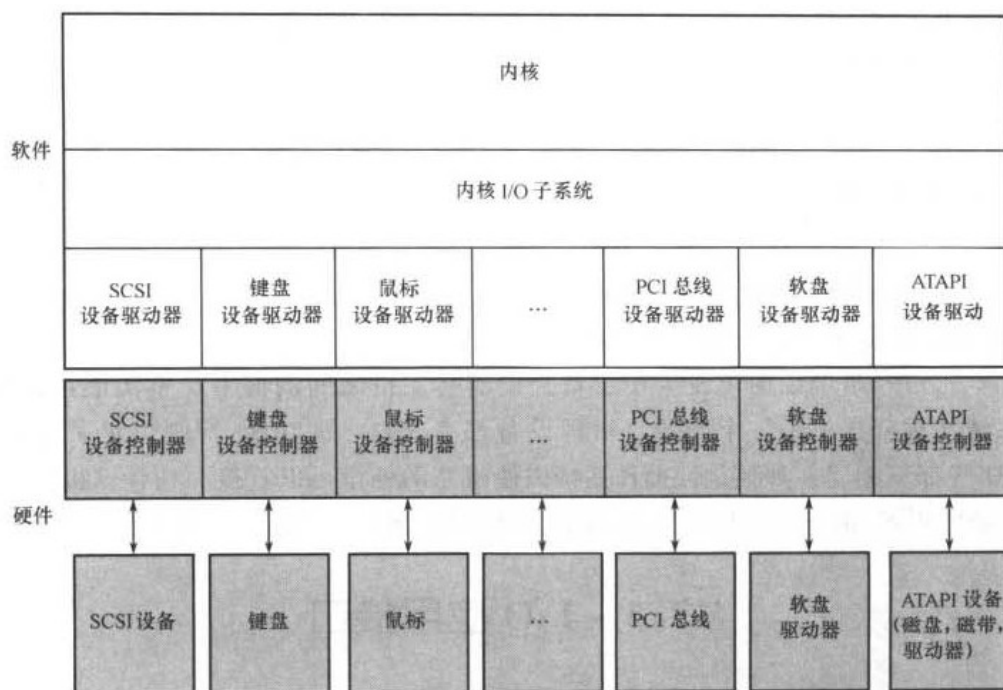


图 9.1 内核 I/O 结构

## 5 I/O 控制方式

I/O 的控制方式可分为：程序 I/O 方式、中断方式、DMA 方式、通道控制方式。

### (1) 程序 I/O 方式

在早期的计算机系统中，处理机对 I/O 设备直接进行控制，采取程序 I/O (Programmed I/O) 方式，也称为轮询方式，即在 CPU 向设备控制器发出一条 I/O 指令启动 I/O 设备进行数据传输时，要同时把状态寄存器中的忙/闲标志 busy 置为 1，然后便不断地循环测试 busy。当 busy=1 时，表示该 I/O 设备尚未输入完一个字(符)，CPU 应继续对该标志进行测试，直至 busy=0，表示该 I/O 设备已将输入数据送入到 I/O 控制器的数据寄存器中，于是 CPU 将从数据寄存器中取出数据，送入内存的指定单元，接着，再启动去读下一个数据，并置 busy=1。

在程序 I/O 方式中，由于 CPU 的速度远远高于 I/O 设备，导致 CPU 的绝大部分时间都处于等待 I/O 设备完成而循环测试之中，造成了 CPU 的极大浪费。但是它管理简单，在要求不高的场合可以被采用。

### (2) 中断控制方式

在现代计算机系统中，对 I/O 设备的控制，广泛地采用中断驱动 (Interrupt-Driven) 方式，即当某进程要启动某个 I/O 设备时，便由 CPU 向相应的设备控制器发出一条 I/O 命令，然后立即返回继续执行原来的任务。设备控制器便按照该命令的要求去控制 I/O 设备。此时，CPU 与 I/O 设备处于并行工作状态。例如，在输入时，当设备控制器收到 CPU 发来的读命令后，便准备接收从相应输入设备送来的数据。一旦数据进入数据寄存器，控制器便通过控制线向 CPU 发送一中断信号，由 CPU 检查输入过程中是否出错，若无错，便向控制器发取走数据的信号，然后便通过控制器将数据写入指定内存单元。

中断驱动方式在 I/O 设备输入数据的过程中，无需 CPU 干预，可以使 CPU 与 I/O 设备并行工作。仅当输完一个数据时，才需 CPU 花费极短的时间去进行中断处理。从而大大地提高了整个系统的资源利用率及吞吐量，特别是 CPU 的利用率。

### (3) DMA 控制方式

中断驱动 I/O 方式虽然大大提高了主机的利用率，但是它以字（节）为单位进行数据传送，每完成一个字（节）的传送，控制器便要向 CPU 请求一次中断（做保存现场信息，恢复现场等工作），仍然占用了 CPU 的许多时间。这种方式对于高速的块设备的 I/O 控制显然是不适合。为了进一步减少 CPU 对 I/O 的干预，引入了直接存储器访问（Direct Memory Access）控制方式。该方式的特点是：

- 它作为高速的外围设备与内存之间成批的数据交换，但是不对数据再做加工处理，数据传输的基本单位是数据块，I/O 操作的类型比较简单；
- 它需要使用一个专门的 DMA 控制器（DMAC）；
- 它采用盗窃总线控制权的方法，由 DMAC 送出内存地址和发出内存读、设备写或设备读、内存写的控制信号来完成内存与设备之间的直接数据传送，而不用 CPU 的干预。有的 DMA 传送甚至不经过 DMAC 的数据缓冲寄存器的再吞吐，传输速率非常高；
- 仅在传送一个或多个数据块的开始和结束时，才需 CPU 干预，整块数据的传送是在控制器的控制下完成的。

DMA 方式较之中断驱动方式，减少了 CPU 对 I/O 控制的干预，进一步提高了 CPU 与 I/O 设备的并行操作程度。

### (4) 通道控制方式

在大、中型计算机系统中，普遍采用由专用的 I/O 处理机来接受 CPU 的委托，独立执行自己的通道程序来实现 I/O 设备与内存之间的信息交换，这就是通道技术。通道技术可以进一步减少 CPU 的干预，即把对一个数据块为单位的读（或写）的干预，减少到对一组数据块为单位的读（或写）的有关的管理的干预。这样可实现 CPU、通道和 I/O 设备三者之间的并行工作，从而更有效地提高了整个系统的资源利用率和运行速度。

通道是通过执行通道程序，并与设备控制器来共同实现对 I/O 设备的控制。通道程序是由一系列的通道指令所构成。通道指令与一般的机器指令不同，在每条指令中包含的信息较多，有操作码、内存地址、计数（读或写数据的字节数）、通道程序结束位 P 和记录结束标志 R。

## 9.2 IO 设备类型

I/O 设备的种类繁多，从操作系统观点来看，其重要的性能指标有：数据传输速率、数据的传输单位、设备的共享属性等。所以可以从不同角度对它们进行分类：

### 1. 按传输速率分类

1) 低速设备：指传输速率为每秒钟几个字节到数百个字节的设备。典型的设备有键盘、鼠标、语音输入设备等；

2) 中速设备：指传输速率在每秒钟数千个字节至数十千个字节的设备。典型的设备有行式打印机、激光打印机等；

3) 高速设备：指传输速率在数百千个字节至数兆字节的设备。典型的设备有磁带机、磁盘机、光盘机等。

### 2. 按信息交换的单位分类

1) 块设备（Block Device）：指以数据块为单位来组织和传送数据信息的设备。这类设备用于存储信息，有磁盘和磁带等。

2) 字符设备 (Character Device) : 指以单个字符为单位来传送数据信息的设备。这类设备一般用于数据的输入和输出, 有交互式终端、打印机等。

### 3. 按资源分配的角度分类

1) 独占设备: 指在一段时间内只允许一个用户 (进程) 访问的设备, 大多数低速的 I/O 设备, 如用户终端、打印机等属于这类设备。因为独占设备属于临界资源, 所以多个并发进程必须互斥地进行访问。

2) 共享设备: 指在一段时间内允许多个进程同时访问的设备。显然, 共享设备必须是可寻址的和可随机访问的设备。典型的共享设备是磁盘。共享设备不仅可以获得良好的设备利用率, 而且是实现文件系统和数据库系统的物质基础。

3) 虚拟设备: 指通过虚拟技术将一台独占设备变换为若干台供多个用户 (进程) 共享的逻辑设备。一般可以利用假脱机技术 (SPOOLing 技术) 实现虚拟设备。