

# 设备管理

1 设备管理概述

2 I/O控制方式

3 设备分配

4 Linux 中的设备管理

# 1 设备管理概述

## 1.1 I/O设备的类型

按设备的所属关系可以将I/O设备分为以下两类。

### (1) 系统设备

系统设备是在系统生成时已登记于系统中的标准设备，属于系统的基本配置。

### (2) 用户设备

用户设备是在系统生成时未登记在系统中的非标准设备。

按设备的信息交换的单位可将I/O设备分为以下两类。

### **(1) 字符设备**

字符设备是以字符为单位进行输入和输出的设备。

### **(2) 块设备**

块设备的输入和输出是以数据块为单位的。

按设备的共享属性可将I/O设备分为以下三类。

### *(1) 独占设备*

所有的字符设备都是独占设备。独占设备是指一段时间内只允许一个用户（进程）访问的设备，即临界资源。

## (2) 共享设备

块设备都是共享设备。共享设备是指一段时间内允许多个进程同时访问的设备。

## (3) 虚拟设备

通过虚拟设备技术把一台独占设备变换为若干台逻辑设备，供若干个用户（进程）同时使用，以提高设备的利用率。

## 1.2 设备管理的任务和功能

设备管理是对计算机的输入/输出系统的管理，它是操作系统中最具有多样性和复杂性的部分。其主要任务如下所述。

(1) 选择和分配I/O设备以便进行数据传输操作。

(2) 控制I/O设备和CPU（或内存）之间交换数据。

(3) 为用户提供一个友好的透明接口，把用户和设备硬件特性分开，使得用户在编制应用程序时不必涉及具体设备，由系统按用户的要求来对设备的工作进行控制。另外，这个接口还为新增加的用户设备提供一个和系统核心相连接的入口，以便用户开发新的设备管理程序。

(4) 提高设备和设备之间、CPU和设备之间以及进程和进程之间的并行操作程度，以使操作系统获得最佳效率。

为了完成上述主要任务，设备管理程序一般要提供下述功能。

- (1) 提供和进程管理系统的接口
- (2) 进行设备分配
- (3) 实现设备和设备、设备和CPU等之间的并行操作
- (4) 进行缓冲管理
- (5) 设备控制与驱动

## 1.3 设备控制器

设备控制器是CPU与I/O设备之间的接口，它接收从CPU发来的命令并去控制I/O设备工作。设备控制器是一个可编址设备，当它仅控制一个设备时，它只有一个唯一的设备地址；当它控制多个设备时，则应具有多个设备地址，使每一个地址对应一个设备。

为实现设备控制器的功能，大多数设备控制器都由以下三部分组成。

- 1 . 设备控制器与处理机的接口
- 2 . 设备控制器与设备的接口
- 3 . I/O逻辑

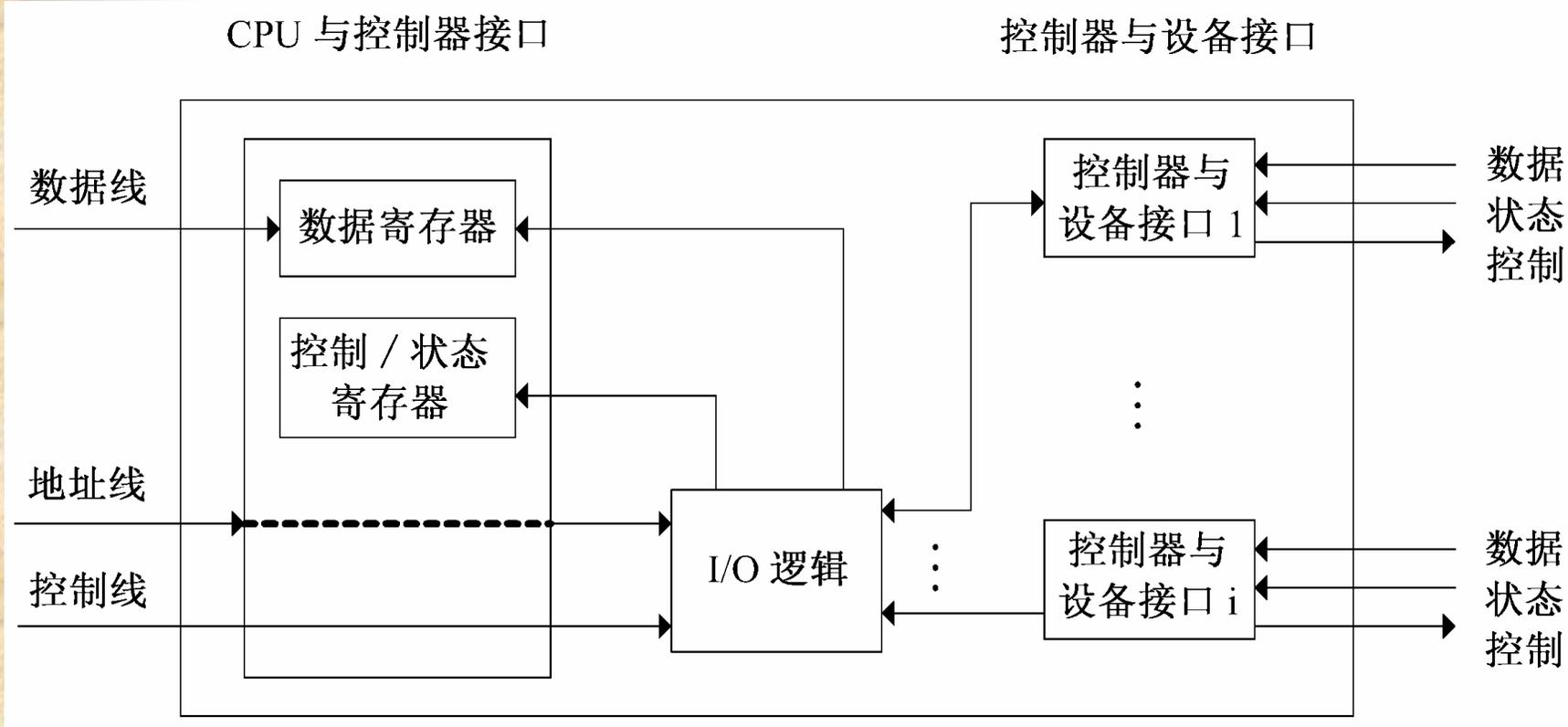


图10.1 设备控制器的组成

## 1.4 I/O通道

设置I/O通道的目的是使一些原来由CPU处理的I/O任务转由通道来承担，从而把CPU从繁杂的I/O任务中解脱出来。

在设置了通道后，CPU只需向通道发送一条I/O指令。通道在收到该指令后，便从内存中取出本次要执行的通道程序，然后执行该通道程序，仅当通道完成了规定的I/O任务后，才向CPU发中断信号。

实际上，I/O通道是一种特殊的处理机，它具有执行I/O指令的能力，并通过执行通道（I/O）程序来控制I/O操作。

通道有两种基本类型：选择通道和多路通道。

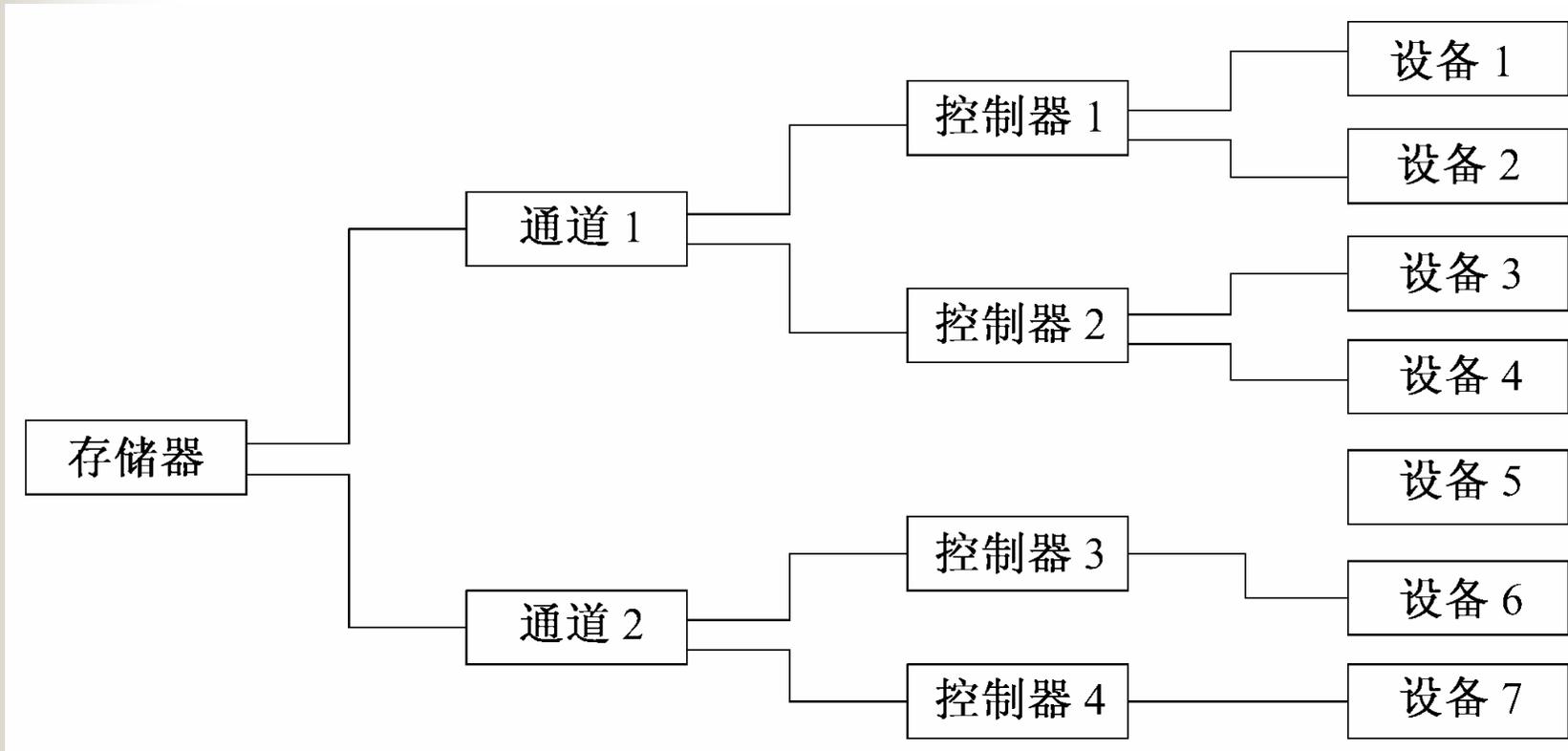


图10.2 单通路I/O系统

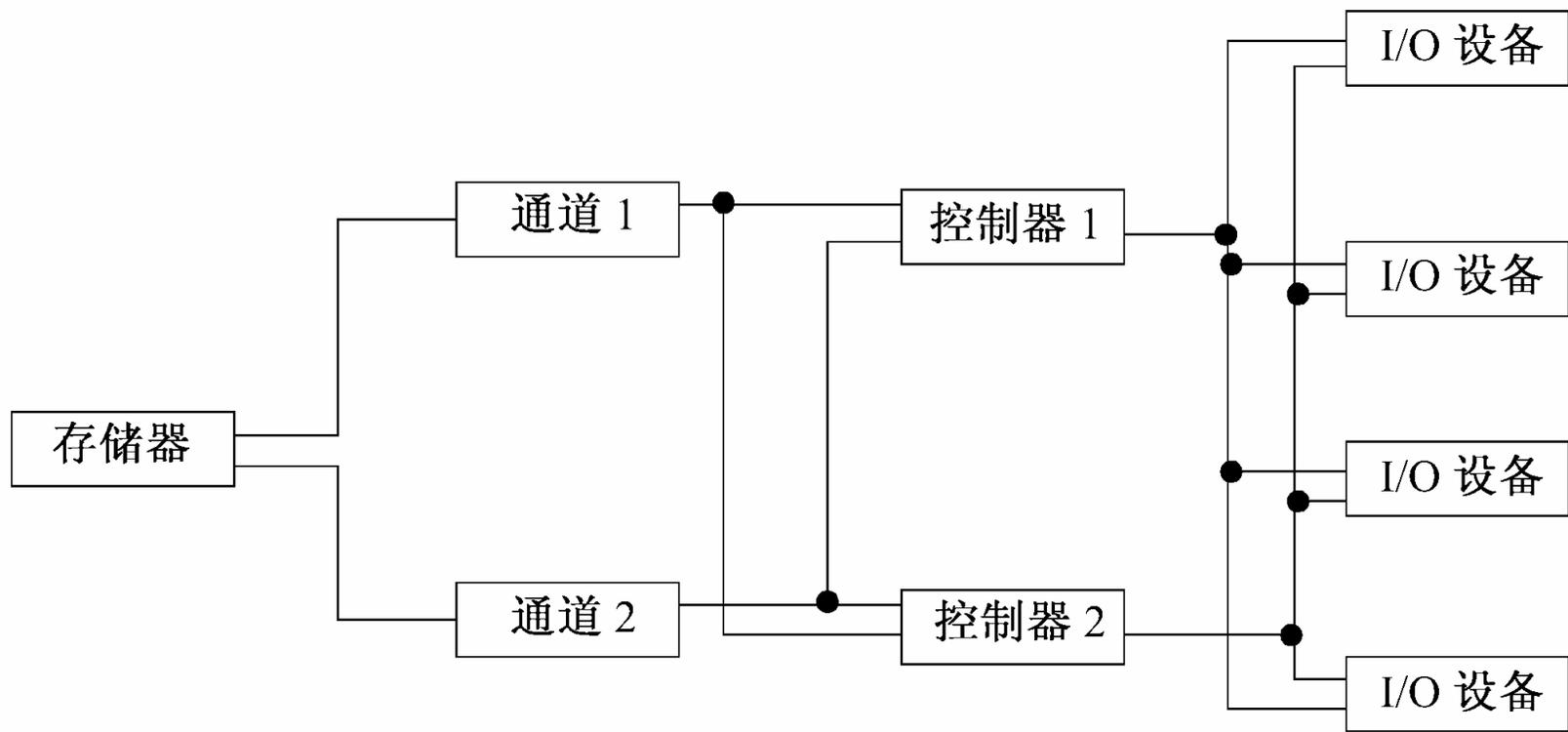


图10.3 多通路I/O系统

## 1.6 设备驱动

设备驱动程序的主要任务，是接收上层软件发来的抽象要求，如read或write命令，再把它转换为具体要求，发送给设备控制器；此外，它也将由设备控制器发来的信号传送给上层软件，从而完成两者间的相互通信。

## 设备驱动程序的处理过程。

- (1) 将抽象要求转换为具体要求
- (2) 检查I/O请求的合法性
- (3) 读出和检查设备的状态
- (4) 传送必要的参数
- (5) 工作方式的设置
- (6) 启动I/O设备

# 2 I/O控制方式

## 2.1 程序I/O方式

图10.4示出了程序I/O方式的流程。

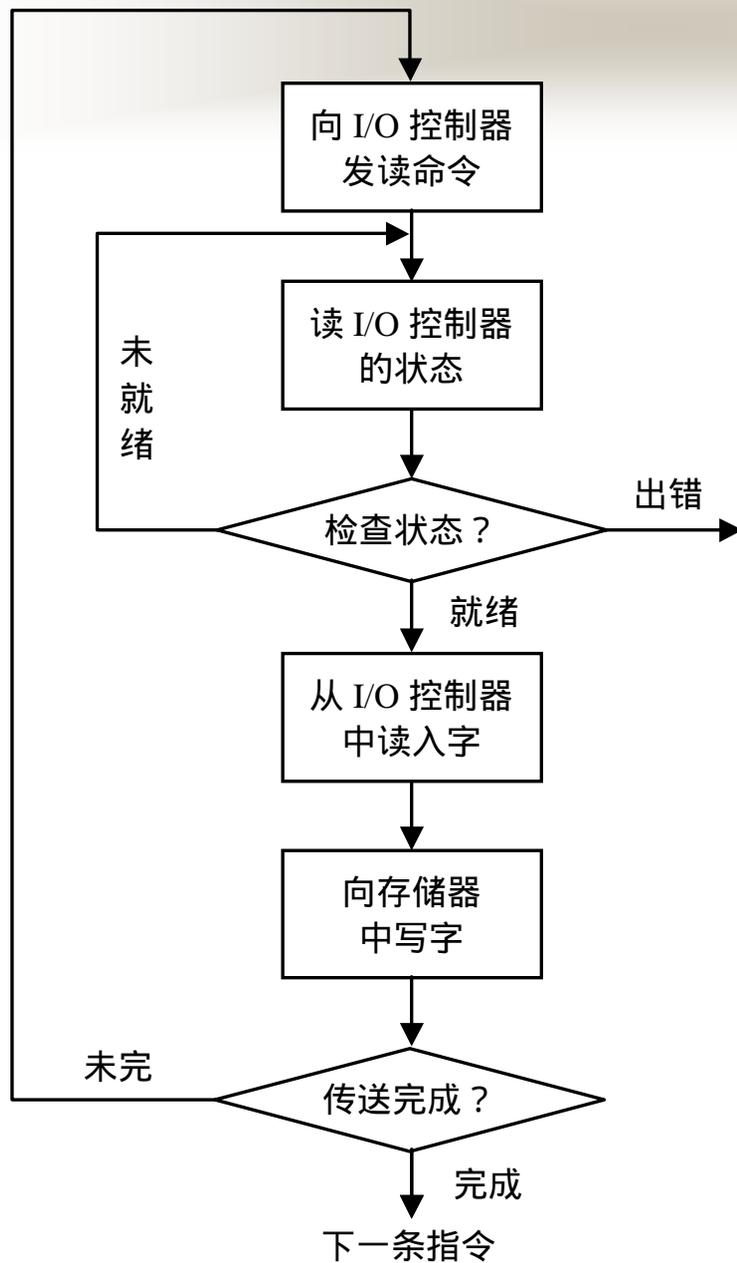


图 10.4 程序 I/O 方式的流程

## 2.2 中断驱动I/O控制方式

在现代计算机系统中，都毫无例外地引入了中断机构，致使对I/O设备的控制，广泛采用中断驱动（Interrupt Driven）方式。即当某进程要启动某个I/O设备工作时，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，设备控制器则按照该命令的要求去控制指定I/O设备。此时，CPU与I/O设备并行操作。

例如，在输入时，当设备控制器收到CPU发来的读命令后，便去控制相应的输入设备读数据。一旦数据进入数据寄存器，控制器便通过控制线向CPU发送一个中断信号，由CPU检查输入过程中是否出错，若无错，便向控制器发送取走数据的信号，然后再通过控制器及数据线，将数据写入内存指定单元中。图10.5示出了中断驱动I/O控制方式的流程。

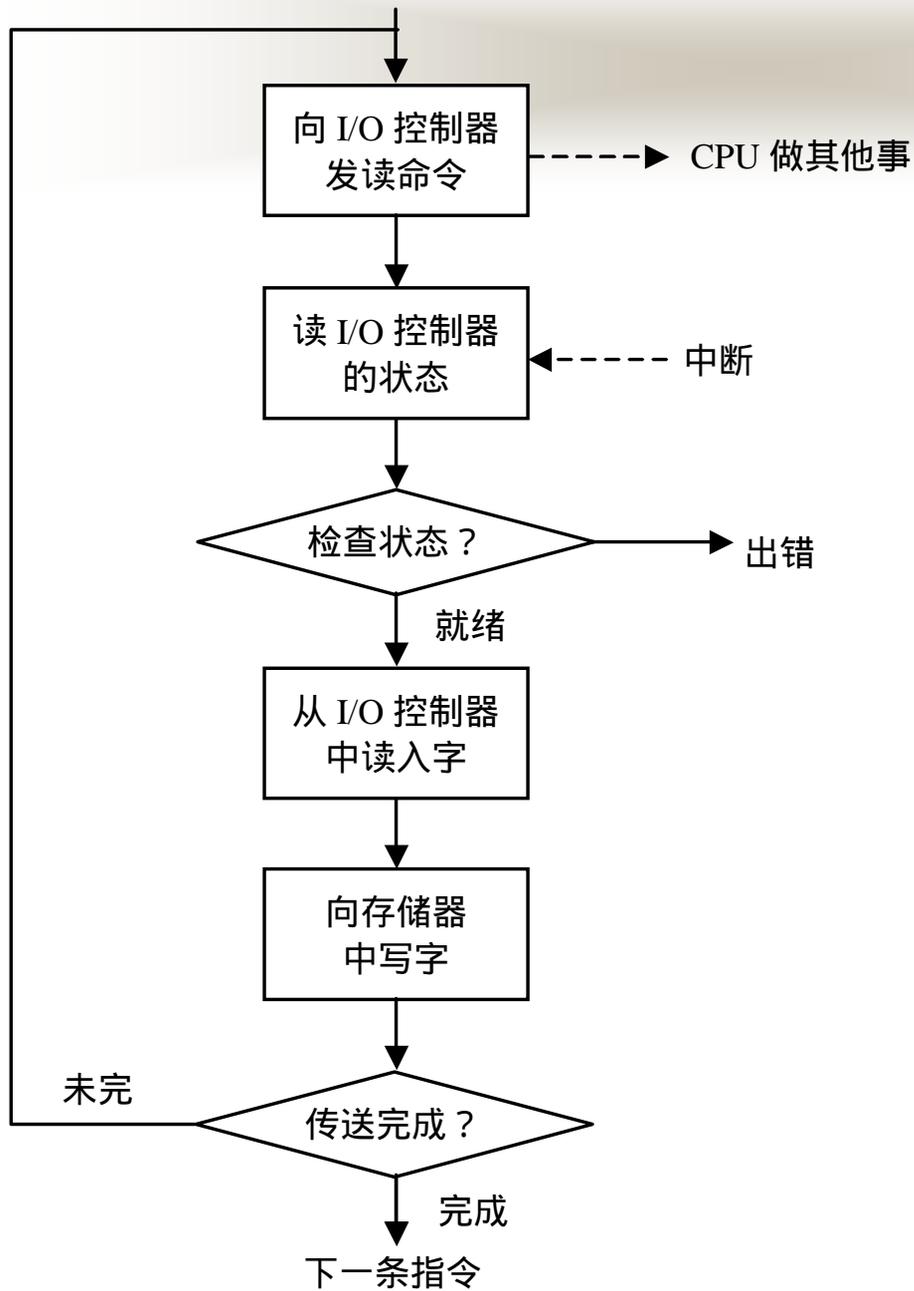


图 10.5 中断驱动 I/O 控制方式的流程

## 2.3 DMA控制方式

DMA方式的基本思想是，在外围设备和内存之间开辟直接的数据交换通路。在DMA控制器中，除了控制状态寄存器和数据缓冲寄存器外，还包括传送字节计数器和内存地址寄存器以及控制电路等。因此，DMA控制器可用来代替CPU，以实现内存和设备之间进行成批的数据交换。DMA方式的传送结构如图10.6所示。

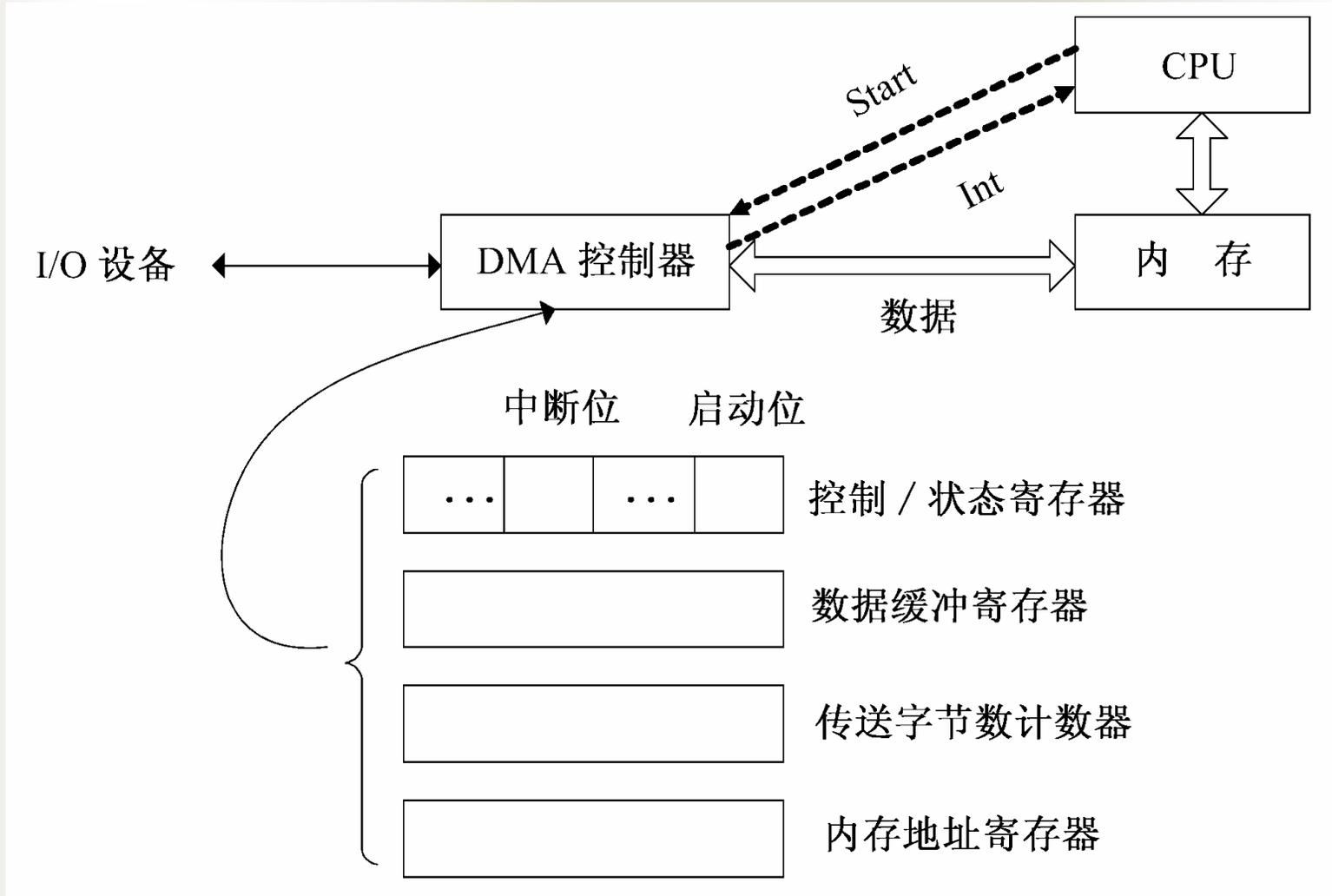


图10.6 DMA方式的传送结构

## 3.2 设备分配时应考虑的因素

### 1. 设备的固有属性

### 2. 设备分配算法

I/O设备的分配，除了与I/O设备的固有属性有关之外，还与系统所采用的设备分配算法有关。I/O设备的分配与进程调度很相似，同样可采用如下一些算法。

(1) 先来先服务

(2) 优先级最高者优先

### 3 . 设备分配中的安全性

从进程运行的安全性方面考虑，设备分配方式有以下两种。

*(1) 静态分配*

*(2) 动态分配*

## 4 . 设备独立性

为了提高系统的可适应性和可扩展性，应使所编制的用户程序与实际使用的物理设备无关，即应用程序独立于具体使用的物理设备，这就是所谓的设备独立性。

# 6 Linux中的设备管理

## 6.1 Linux设备管理概述

在Linux系统中，用户是通过文件系统与设备接口的。所有设备都作为特别文件，从而在设备管理上具有下列特性。

(1) 每个设备都对应文件系统中的索引节点，都有一个文件名。

(2) 应用程序通常可以通过系统调用 `open()` 打开设备文件，建立起与目标设备的连接。

(3) 对设备的使用类似于对文件的存取。

(4) 设备驱动程序都是系统内核的一部分，它们必须为系统内核或者它们的子系统提供一个标准的接口。

(5) 设备驱动程序使用一些标准的内核服务，如内存分配等。

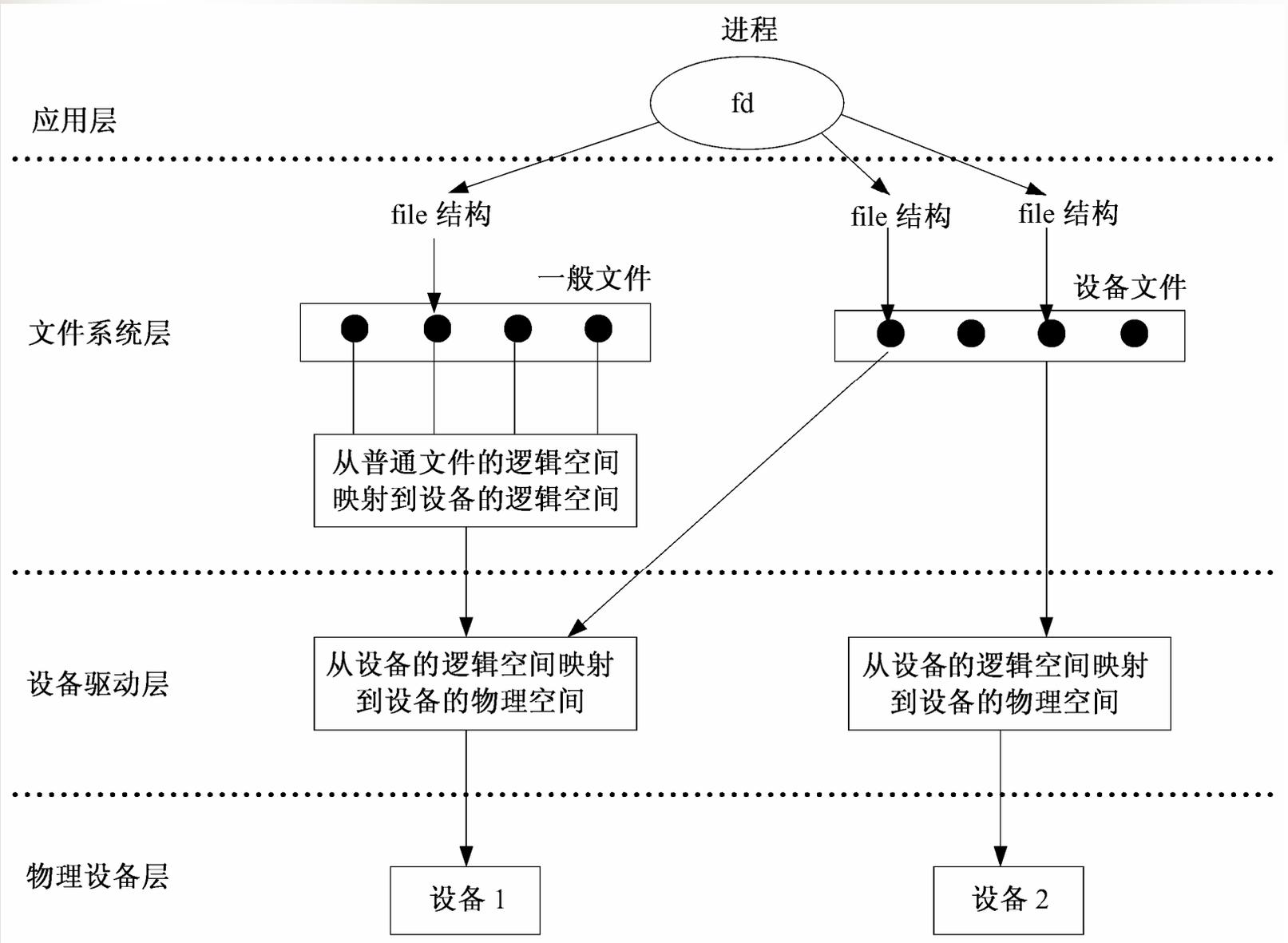


图10.18 设备驱动分层结构