

4 设备管理

- 4. 1 设备管理概述
- 4. 2 设备控制方法
- 4. 3 缓冲技术
- 4. 4 输入输出软件
- 4. 5 设备分配与回收



4.1 设备管理概述

- 计算机系统中有各种设备，通常情况下**除了CPU和内存以外**的所有其他硬件资源统称为计算机外部设备，简称设备。
- 设备管理主要指除CPU和内存之外的设备的**分配、控制、管理和回收**。



4.1.1 设备分类



中国矿业大学

按服务功能分类：

(1) **存储类设备**。通常以存贮大量信息和快速检索为目标，也称外存或后备存储器、辅助存储器，是计算机系统用以存储信息的主要设备，如U盘、光盘等。

(2) **输入/输出类设备**。主要完成把外界信息输入计算机，或者把运算结果从计算机输出的功能，例如键盘、显示器、打印机、音响、摄像头、扫描仪。



4.1.1 设备分类



中国矿业大学

按服务功能分类：

(3) **通信类设备**。这类设备主要完成计算机和外界的通信过程，如网卡、红外设备、蓝牙设备等。



4.1.1 设备分类



中国矿业大学

按每次信息交换的单位分类

(1) **字符设备**。指以单个字符为单位来传递信息的设备，如字符显示终端、键盘、打印机、异步通讯接口等。

(2) **块设备**。指以数据块为单位来组织和传送数据的设备，属于有结构设备，如磁盘、摄像头等。磁盘输入输出以一个扇区为基本单位，摄像头(或图形屏幕)抓取(或显示)以一帧为单位。



4.1.1 设备分类



中国矿业大学

按使用特征分类

(1) **独占设备**。这类设备在用户作业整个运行期间必须为此用户所独占，才能保证传送信息的连贯性。独占性是设备本身的属性。

(2) **共享设备**。这类设备通常指磁带、磁盘一类的存取设备。这里的共享是指多个用户进程运行期间可以交替地使用它们，对它们进行读写。



4.1.1 设备分类



中国矿业大学

按使用特征分类

(3) **虚拟设备**。为了将慢速的独占设备改造成多个用户可共享的设备，以提高设备的利用率、提高系统进程并行的程度，可借助于假脱机技术进行模拟。模拟独占设备的那部分共享设备的空间称为虚拟设备。



4.1.2 设备管理的目标、功能和结构



中国矿业大学

设备管理的目标

(1) **提高使用效率**。要尽量提高CPU和外设之间以及外设与外设之间的并行度，均衡系统中各设备的负载，最大限度地发挥所有设备的潜力，以使操作系统获得最佳效率。

(2) **提供便捷的界面**。所谓便捷，一方面是指用户能独立于具体设备的复杂物理特性而方便地使用设备；另一方面是指对多种不同设备尽量能有统一的操作方式。



4.1.2 设备管理的目标、功能和结构



中国矿业大学

设备管理的功能

(1) **设备的分配与回收**。在多道程序环境下，多个用户或进程往往同时要求使用同一类或同一台设备。操作系统一方面根据进程的请求分配设备，使用结束后，回收设备。

(2) **缓冲区管理**。为了实现低速的输入/输出设备与高速处理器之间的协调工作，一般都在内存中开辟一块存储区作为缓冲区，使CPU和设备通过缓冲区传送数据，从而使设备与设备之间、设备与CPU之间的工作协调起来。



4.1.2 设备管理的目标、功能和结构



中国矿业大学

设备管理的功能

(3) **设备控制和中断处理**。根据用户提出的I/O要求，在未设置通道的系统中，由设备管理软件对设备I/O请求做必要的处理。

(4) **实现虚拟设备**。为了实现多进程并发对独占设备的需求，设备管理实现了虚拟设备功能，将一台独占的物理设备变为多个逻辑设备，从而能够接受多个进程对设备的请求。



4.1.2 设备管理的目标、功能和结构



中国矿业大学

设备管理的结构

设备管理的结构描述了设备、设备控制器、设备驱动程序与应用程序之间的逻辑关系。

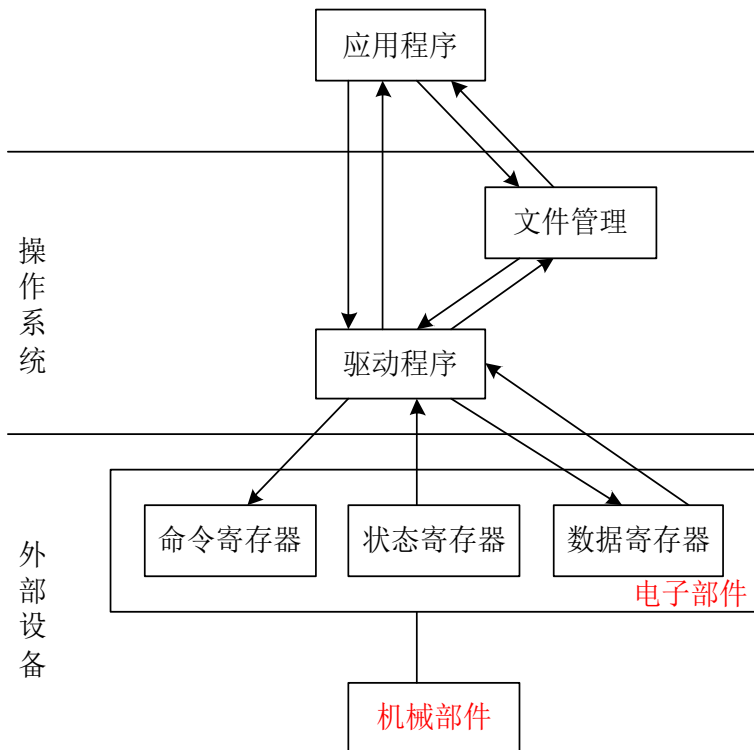
设备一般包括其机械部件和电子部件。为了设计和制造的模块性和通用性，一般设备将这两部分分开。电子部件称为设备控制器或适配器，机械部件则为物理设备，图4.1描述了设备控制器和设备的分离。



4.1.2 设备管理的目标、功能和结构



中国矿业大学



4.1.2 设备管理的目标、功能和结构



中国矿业大学

设备控制器起着承上启下的作用，是**物理硬件**和**逻辑软件的桥梁**，由三部分构成。

控制器与CPU的接口

控制器与设备的接口

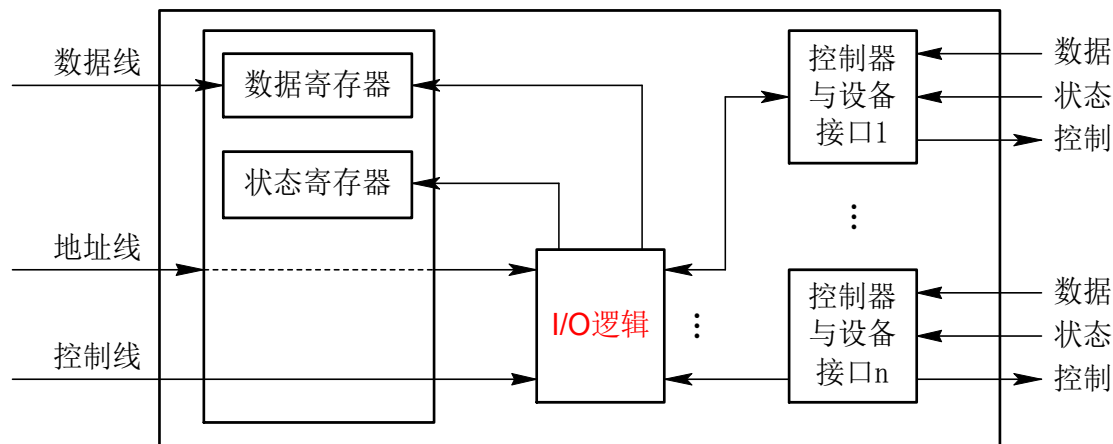


图4.2 设备控制器的构成



4.1.2 设备管理的目标、功能和结构



中国矿业大学

1. 控制器与CPU的接口，主要用于通过数据线、地址线、控制线实现设备控制器与CPU之间的通信。

2. 控制器与设备的接口。在一个控制器中有一个或多个设备接口，一个接口连接一台设备，在每个接口中都有数据、控制和状态三种类型的信号。

3. I/O逻辑。主要用于对I/O的控制。



4.1.2 设备管理的目标、功能和结构



中国矿业大学

每种设备有自己的**驱动程序**，它有两个主要的功能：一是为应用程序实现一个抽象应用编程接口（API），二是提供设备相关的操作，发出适当的命令，实现在应用程序接口中定义的功能。





4.2 设备控制方法

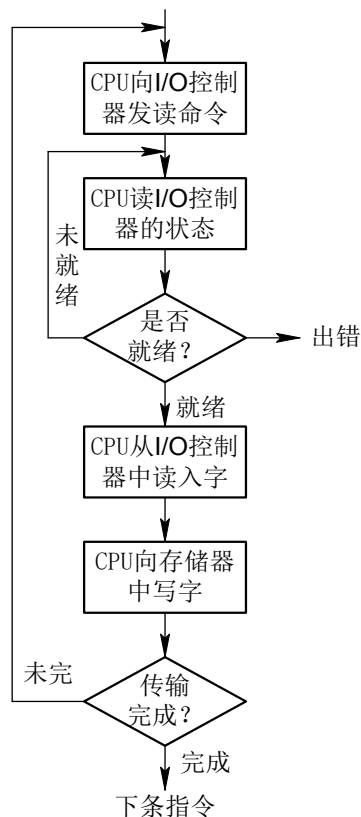
按照 I/O 控制器与 CPU 之间联系方式的不同，可把 I/O 设备的控制方式分为四类：**查询方式、中断方式、DMA方式和通道方式。**

差别：中央处理器和外围设备并行工作的方式不同，并行工作的程度不同。





4.2.1 程序循环查询方式

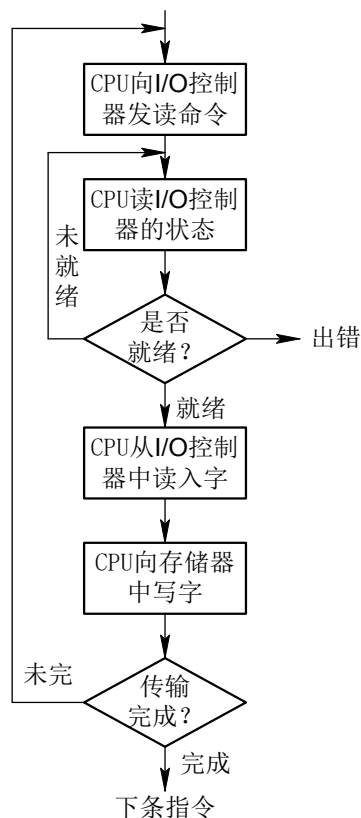


在简单的微型计算机系统中，经常采用此方式，CPU**不断的发送I/O测试指令**用以测试设备控制器的忙/闲（busy）标志位，若设备不忙，则主存与外部设备交换一个字符或一个字。若设备忙，则I/O测试指令不断对它进行测试，直到设备空闲为止。





4.2.1 程序循环查询方式



由于CPU的速度远高于I/O设备的速度，使得CPU绝大部分时间都处于等待I/O完成的循环测试之中。

显然，这对CPU是极大的浪费。但是，它的控制简单，在CPU速度慢、要求不高的场合下常被采用。





4.2.2 中断驱动方式

- 在查询方式中，CPU之所以要不断地测试I/O设备的状态，就是因为在CPU中无中断机构，使I/O设备无法向CPU报告它已完成了一个字符的输入操作。
- 引入中断机构是为了消除设备驱动程序不断地轮询控制器状态寄存器的开销，当I/O操作结束后，由设备控制器“自动地”通知设备驱动程序。
- 使用中断机构的系统中，由硬盘读入到内存的请求执行步骤示例如下：





4.2.2 中断驱动方式

- ① 应用程序请求读操作。
- ② 设备驱动程序查询设备状态寄存器，确定设备是否空闲；如果设备忙，则驱动程序等待，直到它变为空闲为止。
- ③ 驱动程序向设备控制器命令寄存器输入命令，启动设备。
- ④ 设备驱动程序根据操作情况保存相应信息，如最初调用的返回地址，以及I/O操作的一些参数。然后设备管理调用进程调度程序，原进程被挂起，CPU分配给其他进程使用。





4.2.2 中断驱动方式

- ⑤ 设备完成了I/O操作，中断CPU，引起中断处理程序的运行。
- ⑥ 中断处理程序确定是哪个设备引起的中断，然后转移到该设备对应的设备处理程序执行。
- ⑦ 设备处理程序重新从设备表中找到等待I/O操作的状态信息。
- ⑧ 设备处理程序拷贝控制器数据寄存器的内容到用户进程的内存区。
- ⑨ 设备处理程序返回控制权给应用进程，从而用户的程序可以继续运行。





4.2.2 中断驱动方式

由上可见，采用中断驱动进行传输数据时，无须CPU干预，仅当输完数据时，才需CPU花费极短的时间去做些中断处理，因而可使CPU与I/O设备在一定程度上并行工作，从而提高了整个系统的资源利用率及吞吐量。



4.2.3 直接内存访问方式



中国矿业大学

- 中断处理方式虽然提高了主机的利用率，但是每次中断都要保存现场、恢复现场等，仍占用大量CPU时间。在传送数据量大、速度高的情况下，频繁地中断不适应。
- 目前个人计算机块设备传输系统普遍采用DMA（Direct Memory Access）方式，通过DMA控制器控制从内存向设备直接输入输出。

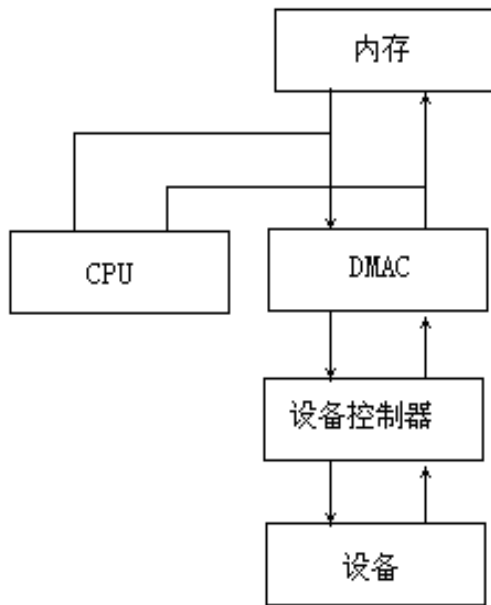


图4.5 直接内存访问





4.2.3 直接内存访问方式

- DMAC内有控制寄存器、状态寄存器、传送字数计数器和内存地址寄存器以及可能的数据缓冲寄存器。它通常采用偷窃总线控制权的方法，由DMAC送出内存地址和发出内存读、设备写或者设备读、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干预。





4.2.3 直接内存访问方式

- 当进程要求输入时，CPU首先为DMA设置输入的字节数、输入到内存的地址，开放中断，启动DMA。然后CPU和DMA并行工作。DMA每传送一个数据并不产生中断，只是字数计数器减1，内存地址寄存器加1。传送字数计数器减至0时，也就是本次DMA传送的数据全部传送完毕时，产生中断，请求CPU进行结束处理。





4.2.3 直接内存访问方式

- DMA方式的特点：

1. 数据传输的基本单位是数据块，即每次传送至少一个数据块；
2. 所传送的数据是从设备直接送入内存，或者直接读出内存的；
3. 在传输时CPU参与更少，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。





4.2.4 通道方式

- 通道方式是DMA方式的发展，进一步减少CPU的干预，把对一个数据块的读(或写)为单位的干预，减少为对一组数据块的读(或写)为单位的干预。
- 通道是一个比DMA功能更强的接口设备，它有专门用于I/O的处理单元。在进行I/O操作时，接受CPU的委托，独立地执行自己的通道程序来实现内存与外设之间的数据传输。





4.2.4 通道方式

- 当主机委托的I/O任务完成后，通道发出中断，请求CPU进行结束处理。这使CPU从对I/O设备的繁忙直接控制中解脱出来，极大地提高了CPU与外设并行工作的程度，从而更有效地提高整个系统的资源利用率。
- 例如，当CPU要完成一组相关的读(或写)操作及有关控制时，只需向I/O通道发送一条I/O指令，以给出其所要执行的通道程序的首址和要访问的I/O设备，通道接到该指令后，通过执行通道程序便可完成CPU指定的I/O任务。

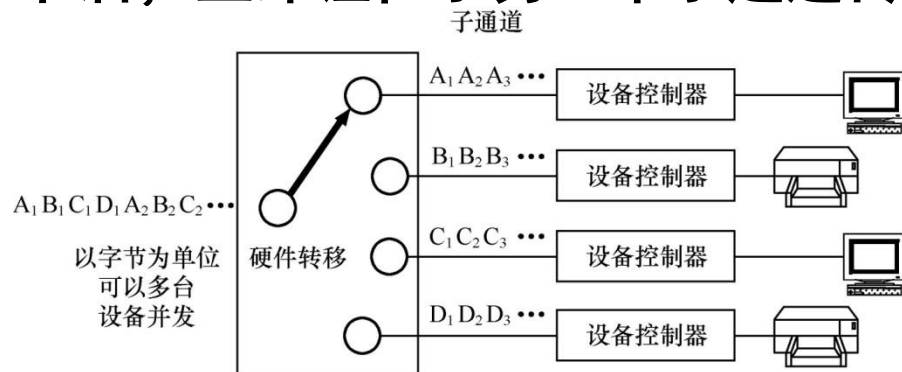




4.2.4 通道方式

1. 通道分类

(1) 字节多路通道：适用于连接打印机、终端等低速或中速的I/O设备。每个通道可以连接8、16甚至更多的子通道。子通道所连接的I/O设备以字节为单位经通道与内存交换数据。一个子通道传送一个字节后，立即让位于另一个子通道传送下一个字节。





4.2.4 通道方式

(2) 选择通道

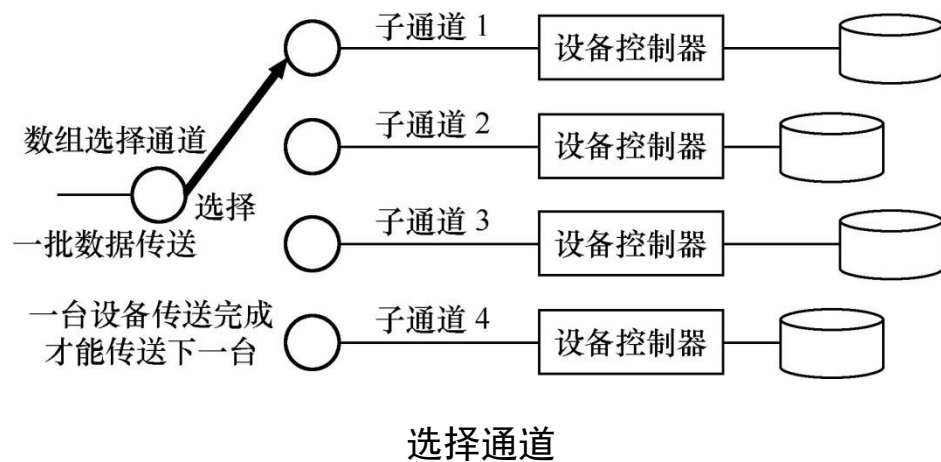
- 字节多路通道不适于连接高速设备，按成组方式进行数据传送的**选择通道**适用于连接高速I/O设备，如磁带、磁盘。
- 选择通道每次传送一批数据，传送速率很高。由于选择通道只有一个分配型子通道，虽然这个子通道可以连接多台设备，但每次只能把子通道分配给一台设备独占使用，直到它被释放，再选择另一台设备为其服务。所以这种通道的利用率并不高。



4.2.4 通道方式



中国矿业大学





4.2.4 通道方式

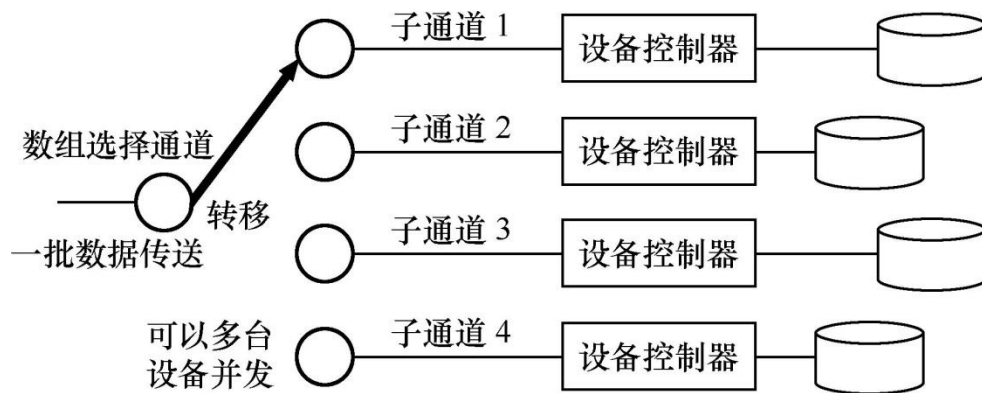
(3) 成组多路通道

- 综合了字节多路通道分时工作和选择通道传输速率高的特点。
- 具有多个非分配型子通道，每个子通道连接一台中、高速I/O设备，因而通道所连接的几个设备可并行工作，而且每台设备的数据传送都是按成组方式进行的。因此，这种通道既具有很高的数据传输速率，又能获得令人满意的通道利用率。





4.2.4 通道方式



成组多路通道





4.2.4 通道方式

2. 通道的工作方式

- 通道I/O操作由两种命令实现控制：CPU的I/O指令和通道本身提供的通道程序。
- CPU的I/O指令的功能一般包括有：启动、停止、查询、清除等功能，
- 通道程序一般有读、写、查询、控制和转移等功能。
- 在通道I/O工作过程中，CPU对通道的通信是向通道发出启动、查询和停止通道指令；而通道向CPU的通信则采用中断方式向CPU汇报。





4.2.4 通道方式

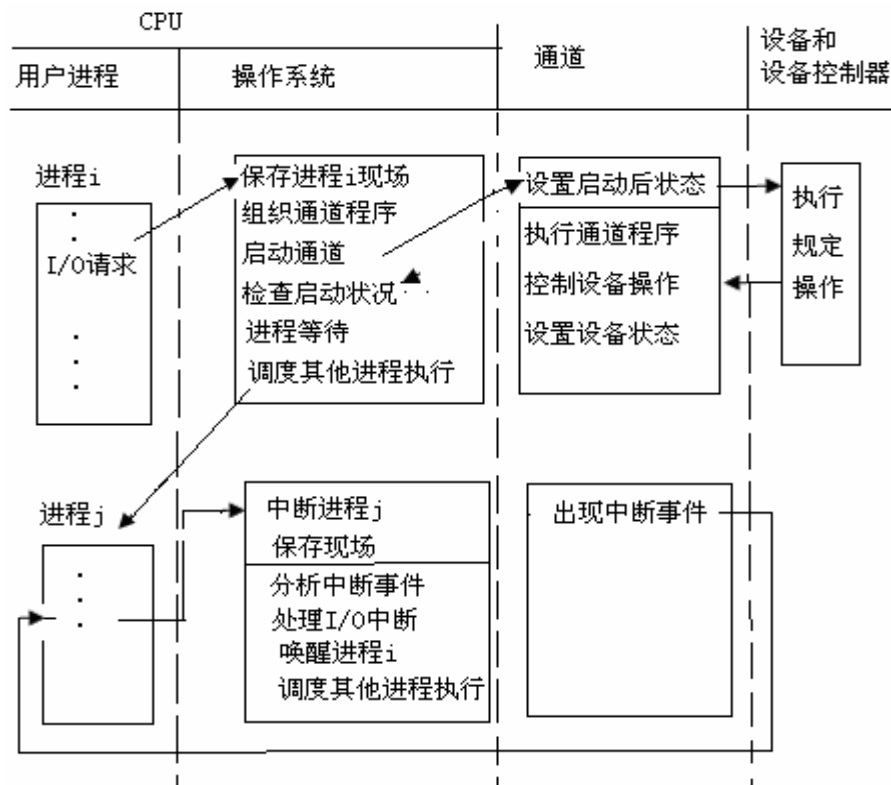


图4.6 通道与CPU之间通信示意图





4.3 缓冲技术

缓冲技术作用：

1. 改善中央处理器与外围设备之间速度不匹配的矛盾，提高CPU 和I/O 设备的并行性。
2. 减少I/O 对CPU 的中断次数和放宽对CPU 中断响应时间的要求。
3. 协调逻辑记录大小与物理记录大小不一致的问题。





4.3 缓冲技术

- 缓冲有硬件缓冲和软件缓冲之分。
- 硬件缓冲是以专用的寄存器作为缓冲器。软件缓冲即在操作系统的管理下，在内存中划出若干个单元作为缓冲区。
- 软件缓冲的好处是易于改变缓冲区的大小和数量，缺点是占据一部分内存空间。
- 根据缓冲区个数的多少和结构，缓冲可分为：
单缓冲、双缓冲、多缓冲、循环缓冲与缓冲池。





4.3.1 单缓冲

- 单缓冲指当一个进程发出I/O请求时，操作系统便在主存中为之分配一个缓冲区，用来临时存放输入 / 输出数据。

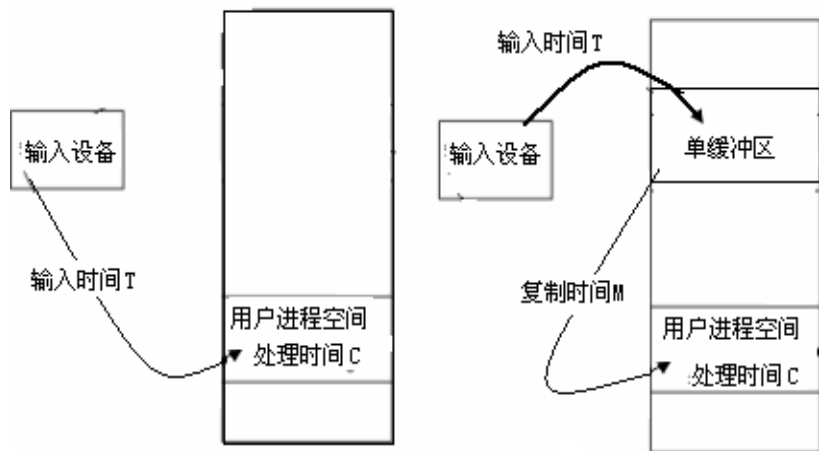


图4.7 无缓冲区(左)与单缓冲区输入(右)



4.3.1 单缓冲



中国矿业大学

- 在数据的输入或者输出过程中，在某一时刻该缓冲区只能存放输入数据或输出数据，而不能既是输入数据又是输出数据，否则会引起缓冲区中数据的混乱。对缓冲区来说，信息的输入和输出实际上是串行工作的。





4.3.2 双缓冲

- 双缓冲指在操作系统中为某一设备设置两个缓冲区，当一个缓冲区中的数据尚未被处理时可使用另一个缓冲区存放从设备读入或读出的数据，以此来进一步提高CPU和外设的并行程度。





4.3.2 双缓冲

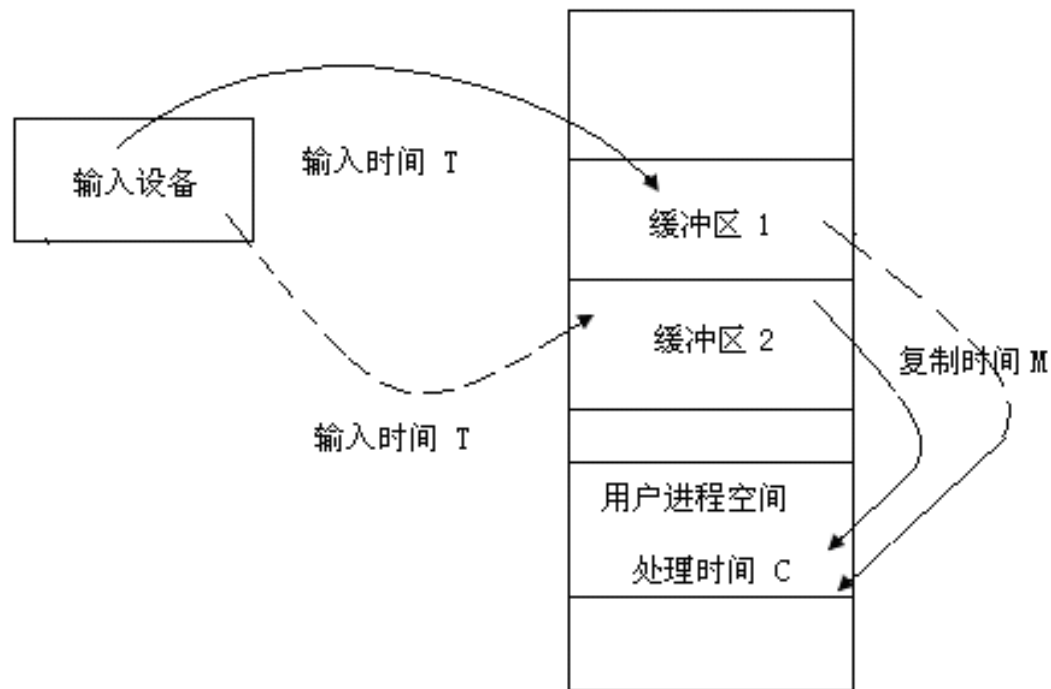


图4.9 双缓冲工作示意图





4.3.2 双缓冲

- 在输入数据时，首先填满缓冲区1，操作系统可从缓冲区1把数据送到用户进程区，用户进程便可对数据进行加工计算；与此同时，输入设备填充缓冲区2。当缓冲区1空出后，输入设备再次向缓冲区1输入。操作系统又可以把缓冲区2的数据传送到用户进程区，用户进程开始加工缓冲区2的数据。这样就两个缓冲区交替使用，使CPU和I/O设备的并行性进一步提高。仅当两个缓冲区都空或者都满，进程还要提取数据或者写入数据时，它才被迫等待。



4.3.3 多缓冲



中国矿业大学

1. 循环缓冲

- 为了使缓冲区资源得到充分利用，操作系统从主存区域中分配一组缓冲区，如图4.10所示，根据用途分为：输入缓冲区组和输出缓冲区组。
- 每个缓冲区有一个链接指针指向下一个缓冲区，最后一个缓冲区指针指向第一个缓冲区，组成了循环缓冲。每个缓冲区的大小可以等于物理记录的大小。
- 循环式的多缓冲区是系统的公共资源，由系统统一分配和管理，供各进程共享。



4.3.3 多缓冲



中国矿业大学

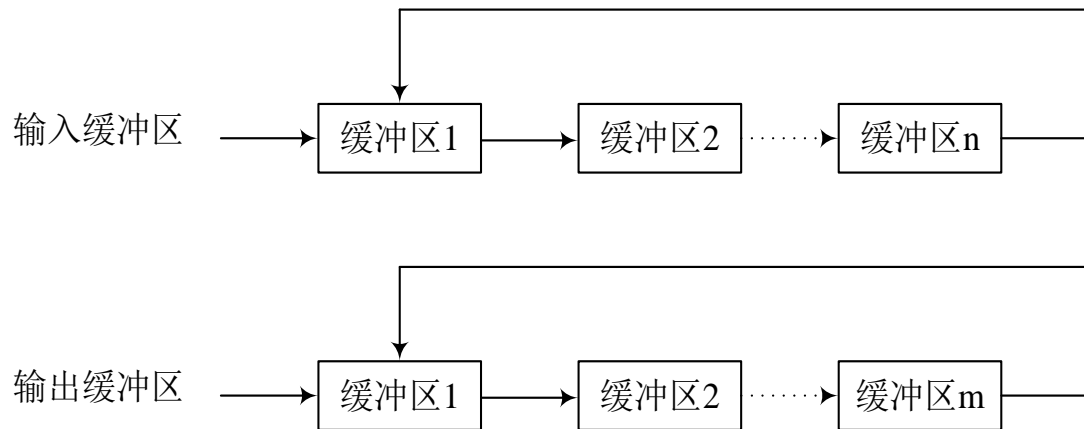


图4.10 循环缓冲



4.3.3 多缓冲



中国矿业大学

2. 缓冲池

- 缓冲池由内存中的一组缓冲区构成, 各缓冲区之间并不一定采用循环链表的方式进行链接, 操作系统与用户进程将轮流地使用各个缓冲区, 以改善系统性能。
- 缓冲池中多个缓冲区可供多个进程使用, 既可用于输出又可用于输入, 是一种现代操作系统经常采用的一种公用缓冲技术。



4.3.3 多缓冲



中国矿业大学

2. 缓冲池

- 缓冲池中的缓冲区组织成三个队列：
空闲缓冲区队列emq
输入缓冲区队列inq
输出缓冲区队列outq
- 每个缓冲区根据其当前的工作性质不同，分为四种状态：
收容输入、提取输入、收容输出、提取输出。



4.3.3 多缓冲



中国矿业大学

2. 缓冲池

对缓冲池的管理有两个基本操作：

- **Getbuf (type)** 用于从type所指定的队列的队首摘下一个缓冲区；
- **Putbuf (type, number)** 用于将由参数number所指示的缓冲区挂在type队列上。



4.3.3 多缓冲



中国矿业大学

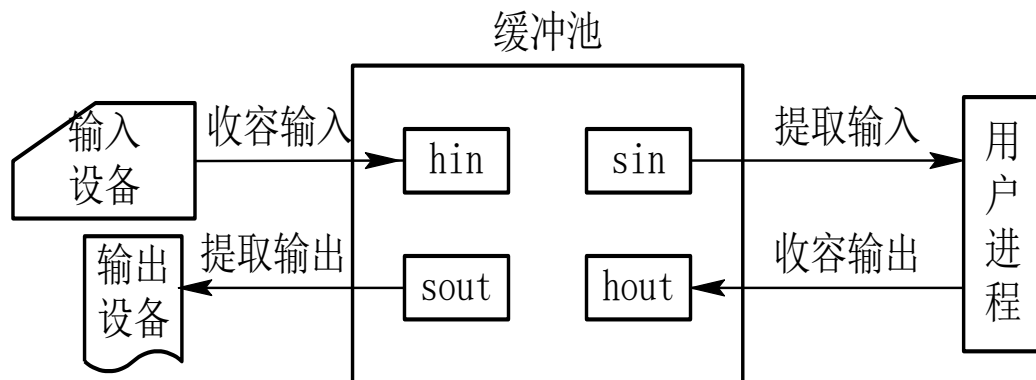


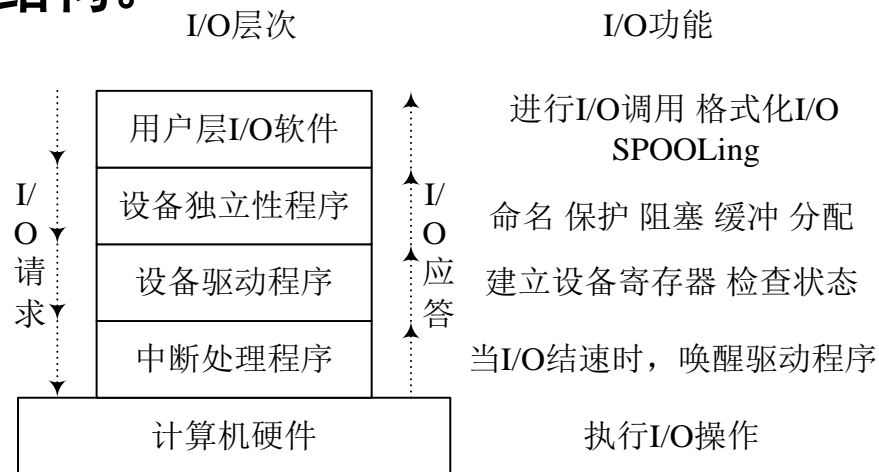
图4.11 缓冲池的工作方式





4.4 输入输出软件

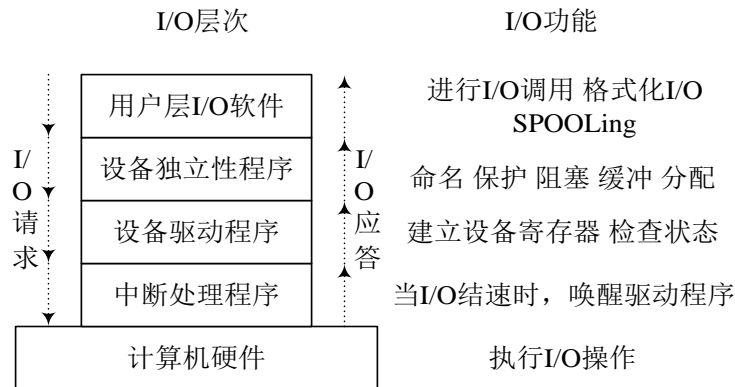
输入输出软件是实现**I/O管理的软件部分**，其目标是改善输入输出设备的效率，实现统一标准的输入输出设备管理方式，通常被组织成一种层次结构。





4.4 输入输出软件

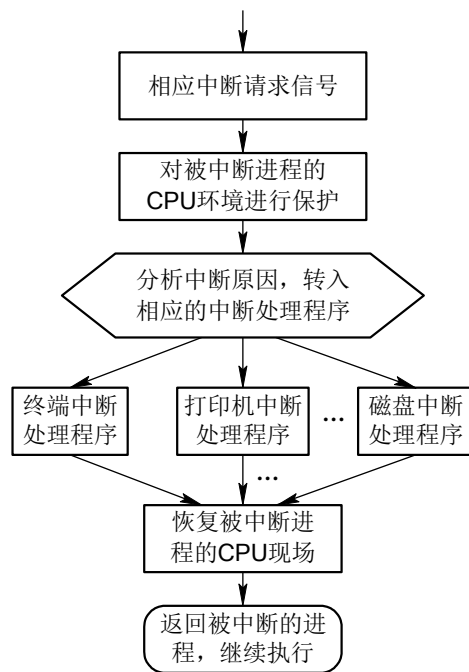
处于底层的软件用来屏蔽输入 / 输出硬件的细节，从而实现上层的**设备无关性**，高层软件则主要为用户提供一个**统一、规范、方便的接口**。





4.4.1 中断处理程序

凡是涉及到输入/输出的**开始、结束或者异常**时，一般都会发出中断信号，正在运行的进程将被中断运行，保护现场环境，之后**CPU便转向中断处理程序，最后解除相应进程的阻塞状态，恢复执行。**





4.4.2 设备驱动程序

- 设备驱动程序是指驱动物理设备和DMA控制器或I/O控制器等**直接进行I/O操作的程序集合**。
- **不同类型的设备应有不同的设备驱动程序**，设备驱动程序主要**负责启动指定设备**，即负责设置与相关设备有关的寄存器的值，启动设备进行I/O操作，指定操作的类型和数据流向等。





4.4.2 设备驱动程序

- (1) 接收由I/O进程发来的命令和参数，并将命令中的**抽象要求转换为具体要求**。
- (2) 检查用户I/O请求的合法性，了解I/O设备的状态，传递有关参数，**设置设备的工作方式**。
- (3) **发出I/O命令**，如果设备空闲，便立即启动I/O设备去完成指定的I/O操作；如果设备处于忙碌状态，则将请求者的请求挂在设备队列上等待。





4.4.2 设备驱动程序

(4) **及时响应由控制器或通道发来的中断请求**，并根据其中断类型调用相应的中断处理程序进行处理。对有通道的计算机，驱动程序还应能够根据用户的I/O请求，自动地构成通道程序。

(5) I/O完成后，由通道（或设备）**产生中断信号**。CPU接到中断请求后，则响应中断，然后转去执行相应的中断处理程序，唤醒因等待I/O完成而阻塞的进程，调度用户进程继续运行。





4.4.3 设备独立性程序

- 所谓设备独立性，也称为**设备无关性**，是指在用户程序中不要直接使用物理设备名（或设备的物理地址），而只**使用逻辑设备名**。
- **逻辑设备是实际物理设备属性的抽象**，它不限于某类具体设备。逻辑设备究竟和哪一个具体的物理设备相对应，还要由系统根据当时的设备忙、闲情况来决定或由系统管理员指定。





4.4.3 设备独立性程序

引入设备独立性概念，**用户程序可使用逻辑设备名，而不必使用物理设备名**，有以下优点：

(1) **设备分配更加灵活，提高了设备的利用率。**当多用户多进程请求分配设备时，系统可根据设备当时的忙闲情况，合理调整逻辑设备名与物理设备名之间的对应情况，以保证设备的充分使用。

(2) **可以实现I/O重定向。**所谓I/O重定向是指可以更换I/O操作的设备而不必改变应用程序。





4.4.4 用户层软件

- 用户层I/O软件实际上就是**面向设备具体应用的软件**，它是I/O系统软件的最上层，它面向程序员，当接收到用户的I/O指令后，把具体的请求发送到设备无关的I/O软件，进行进一步的处理。
- 针对I/O设备的用户层软件**主要包含**用于I/O操作的库函数和和前文提到的SP00Ling假脱机系统。





中国矿业大学

4.5 设备分配与回收

设备分配和回收的任务是**按照一定的算法将设备有关资源分配**给申请进程，在进程使用完毕后还要负责**收回相关设备资源**。



4.5.1 设备信息描述



中国矿业大学

设备信息
描述表

系统设备表 SDT (System Device Table)

设备控制表 DCT (Device Control Table)

控制器控制表 COCT (Controller Control Table)

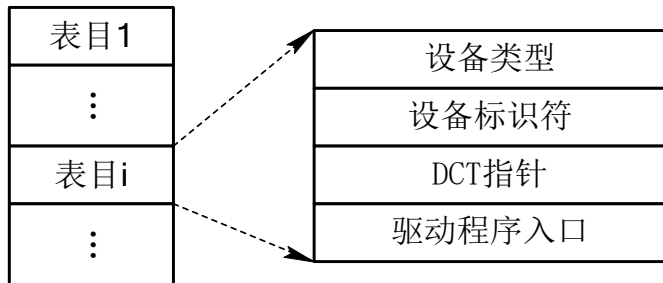
通道控制表 CHCT (Channel Control Table)





4.5.1 设备信息描述

1. 系统设备表: 整个系统中设置惟一的**系统设备表SDT**, 记录了系统中全部设备情况, 并为每个设备设置了一个表项。SDT的每个表项主要包括: 设备类型、设备标识符、指向DCT的指针以及驱动程序入口。

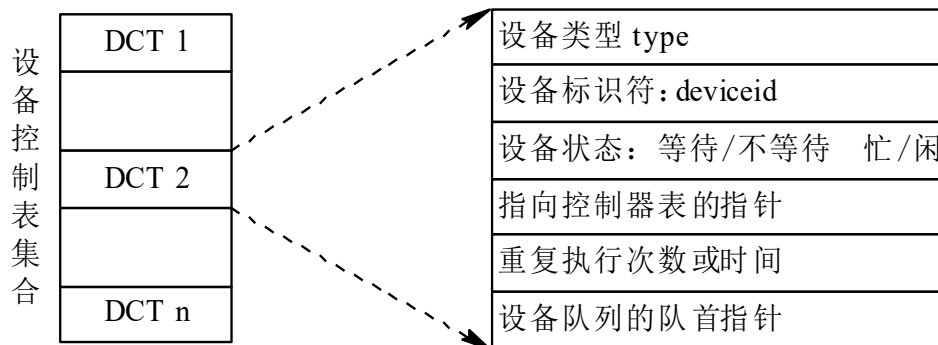




4.5.1 设备信息描述

2. 设备控制表DCT：用来记录系统内任一设备的特性、设备和I/O控制器的连接情况以及设备的分配和使用情况。

DCT在系统生成时或在该设备和系统连接时创建，但表中的内容则可根据系统执行情况动态修改。





4.5.1 设备信息描述

3. 控制器控制表：系统为每个控制器都设置了一个COCT，记录I/O控制器的使用情况以及所连接的通道情况。

主要表项包括：控制器标识符、控制器的状态、与控制器相连接的通道表指针、控制器队列的队首指针、控制器队列的队尾指针，各相应项意义与DCT类似。





4.5.1 设备信息描述

4. 通道控制表：在设置有通道的系统中，操作系统为每个通道都配备了一张通道控制表，与COCT类似，不再赘述。





4.5.2 设备分配策略

按照设备本身使用属性，设备分成**独占设备、共享设备和虚拟设备**三类，相应的设备分配策略就叫**独占方式、共享方式和虚拟方式**。

独占方式就是把一台设备**固定地分配**给一个用户或进程，直到它运行结束。这种方式最方便，管理起来也简单，但易造成资源浪费。因为用户程序或进程运行过程中不会自始至终都使用像打印机这类设备。



4.5.2 设备分配策略



中国矿业大学

共享方式是指设备可以在多个用户（或进程）**“交替”使用**，即一个进程需要时，便申请它，获得后使用它，用完就释放它。其他进程也如此方式使用。磁盘、磁带就是可共享方式分配、使用的设备。





4.5.2 设备分配策略

虚拟方式：为了提高独占设备的利用率，提高进程并行程度，引入了虚拟设备技术。**虚拟设备技术就是利用快速、共享设备（例如磁盘）把慢速、独占设备模拟成为高速设备。**

例如，某用户需要打印机，操作系统分配给它的不是物理打印机，而是一个磁盘文件，用户进程输出打印结果，是向该磁盘文件输出，系统再把该磁盘文件传输给打印机进程，依次排队打印输出。**从用户看来，系统为自己提供了一台高速的物理打印机。**





4.5.3 SPooling技术

早期为了缓和CPU的高速性与I/O设备的低速性之间的矛盾,引入**脱机输入输出技术**,该技术是利用专门的外围控制机,实现低速I/O设备与高速磁盘的数据传输。

当出现多道程序并发技术后,可以利用其中的一**道程序,来模拟脱机输入输出时的外围控制机功能**,把低速I/O设备上的数据传送到高速磁盘上;或者把数据从磁盘传送到低速输出设备上。





4. 5. 3 SP00Ling技术

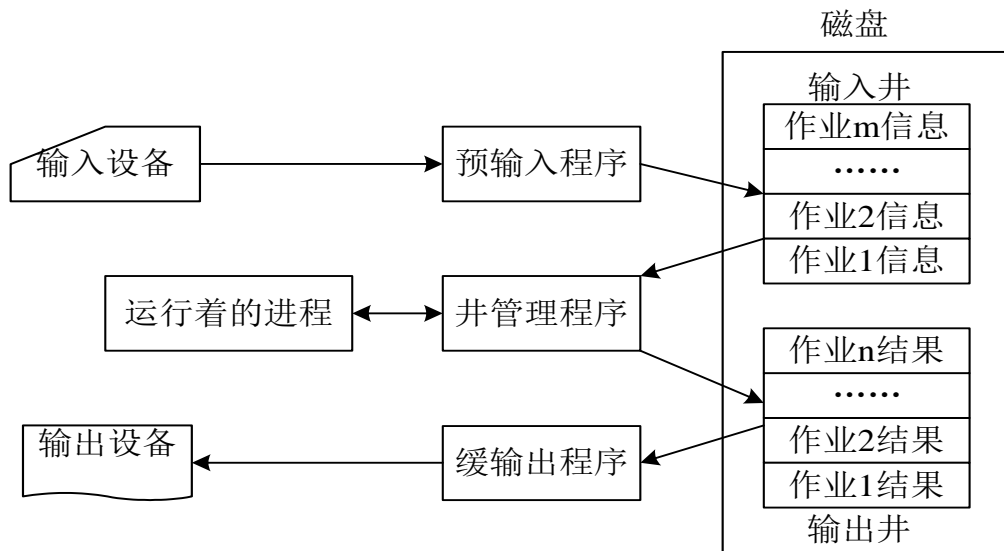
这样，便可在主机的直接控制下，实现**脱机输入输出**功能。此时的外围操作与CPU对数据的处理同时进行，这种在联机情况下实现的同时外围操作称为**SP00Ling**，或称为**假脱机系统**。





4.5.3 SPooling技术

输入井模拟假脱机输入，用于**收容输入**的数据。
输出井模拟脱机输出，用于**收容用户程序的输出数据**。





4.5.3 SPooling技术

预输入程序模拟脱机输入时的**外围控制机**，将输入设备的输入信息送到输入井，当相关进程需要输入数据时，直接从输入井读入到内存中的用户程序区。

缓输出程序模拟脱机输出时的外围控制机，把用户要求输出的信息从用户程序区送到输出井，待输出设备空闲时，将输出井中的信息送到输出设备上。

井管理程序负责管理**输入井与输出井**的协调工作。





4. 5. 3 SP00Ling技术

- 在进程执行过程中，如果请求启动某台设备的输入或者输出工作，操作系统得到该请求并调出井管理程序，控制从相应输入井读取信息或将信息送至输出井内。
- 当输入井中的信息被进程取出后，相应的井区应归还。通过预输入（出）管理程序从输入（出）井读入（出）信息和通过设备管理从设备上输入（往设备上输出）信息，其结果是一样的。





4.5.3 SP00Ling技术

特点：

(1) **提高了I/O的速度**，缓和了高速的处理器与低速输入输出设备之间的矛盾。

(2) 将独占设备改造为共享设备，**提高了设备的利用率**。

(3) **实现了虚拟设备功能**，将物理的单个设备变换为多个对应的逻辑设备。





4.5.4 设备分配算法

设备分配时，要将请求设备的进程组织成请求队列。对该请求队列，操作系统按照一定的算法将设备分配给进程。设备分配算法与进程调度算法有相似之处：

(1) **先来先服务算法**。系统允许多个进程请求同一个设备，也允许一个进程请求多个设备。系统按照进程对某设备提出I/O请求的先后顺序，将对应进程组织成一个设备请求队列。当设备空闲时，设备分配程序总是把此设备首先分配给队首进程。



4.5.4 设备分配算法



中国矿业大学

(2) **优先级高者优先算法**。对有I/O的进程按照其优先级的高低进行排列，当有一个新进程要加入设备请求队列中时，根据进程的优先级插在适当的位置。保证在该设备空闲时，系统能从I/O请求队列的队首取下一个具有最高优先级进程，并将设备分配给它。

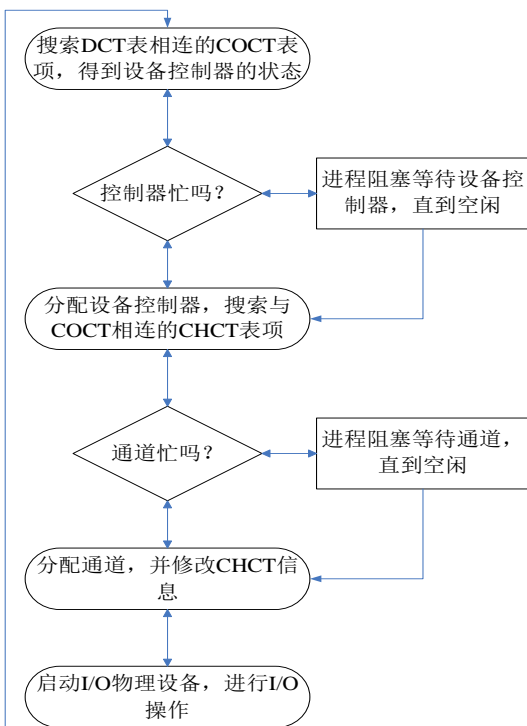
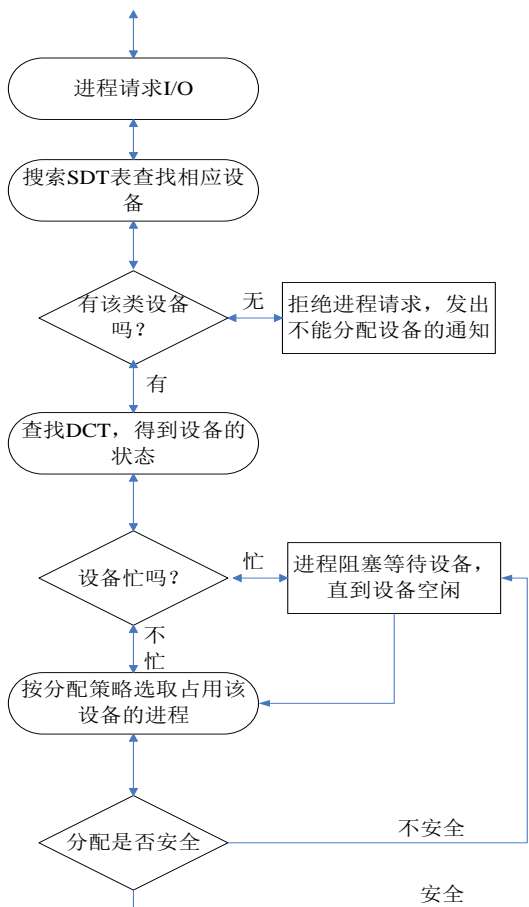




4.5.5 设备分配与回收过程

设备分配过程是**逐步由抽象信息表格到具体物理设备的过程**，首先使用了管理逻辑设备信息的系统设备表，继而查找设备控制表，然后通过设备控制器和通道启动设备，传输信息。





中国矿业大学





4.5.5 设备分配与回收过程

- 当某一作业（或进程）使用完设备后，则需“释放设备”，设备回收过程是设备分配过程的逆过程。
- 回收时，要请求操作系统依次修改与设备有关的通道控制表、设备控制器控制表、设备控制表和系统设备表，主要是修改其中的状态信息，以使设备能够及时被下一个进程使用。

