

第五章 输入输出系统

电子科技大学信息与软件工程学院

5 输入输出系统

- I/O 系统简介
- 中断处理程序
- 设备驱动程序
- 磁盘系统及磁盘调度



5.1 I/O 系统简介

- **设备管理的对象：** 主要是I/O设备。
- **设备管理的基本任务：** 完成用户提出的I/O请求，提高I/O速率以及改善I/O设备的利用率。
- **设备管理的主要功能有：** 缓冲区管理、设备分配、设备处理、虚拟设备及实现设备独立性等。



5.1 I/O 系统简介

I/O 系统的基本功能

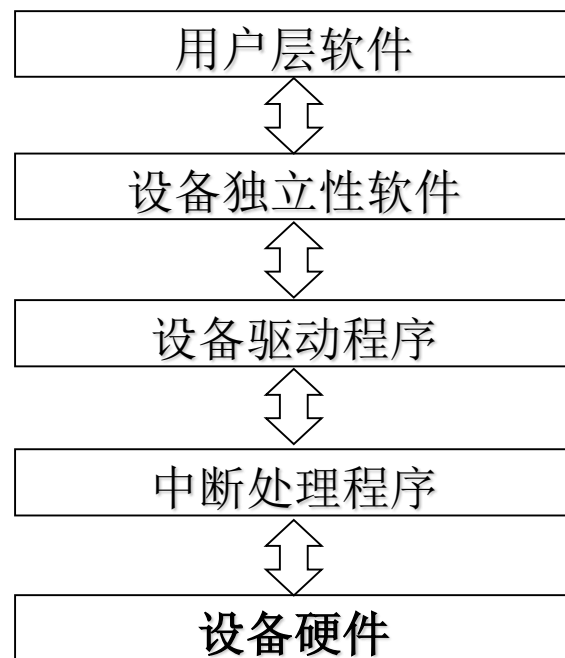
- 1) 设备分配
- 2) 设备映射
- 3) 设备驱动
- 4) I/O缓冲区的管理



5.1 I/O 系统简介

通用设备管理分层模型

为了使得复杂的 I/O 软件具有清晰的结构、更好的可移植性和易适应性，I/O 系统目前普遍采用层次式的结构。通常划分为四层。



I/O系统的层次结构



5.2 中断处理程序

5.2.1 中断简介

- 1) 中断和陷入—CPU 外部事件和内部事件导致
- 2) 中断向量表和中断优先级
- 3) 对多中断源的处理方式
 - 屏蔽（禁止）中断
 - 嵌套中断



5.2 中断处理程序

5.2.1 中断简介

基本概念

- **中断源**：引起中断发生的事件
- **中断请求**：中断源向CPU发出的请求中断处理信号
- **中断响应**：CPU收到中断请求后转到相应的事件处理程序的过程
- **关中断/开中断**：CPU内部的PSW的中断允许位被清除/被设置，不允许/允许CPU响应中断。用于保证某段程序执行的原子性
- **中断屏蔽**：在中断请求产生后，系统有选择地封锁一部分中断而允许另一部分仍能得到响应。有些具有最高优先级的中断不允许被屏蔽。



5.2 中断处理程序

5.2.2 中断处理程序

对于为每一类设备设置一个I/O进程的设备处理方式，其中断处理程序的处理过程分成以下几个步骤：

1. 测定是否有未响应的中断信号

- 程序完成当前指令后测试是否有未响应的中断信号。
- 如果没有，继续执行下一条指令。
- 如果有，则停止原有进程的执行，准备转去执行中断处理程序，为把处理机的控制权转交给中断处理程序做准备。



5.2 中断处理程序

5.2.2 中断处理程序

2、保护被中断进程的运行环境（中断现场）

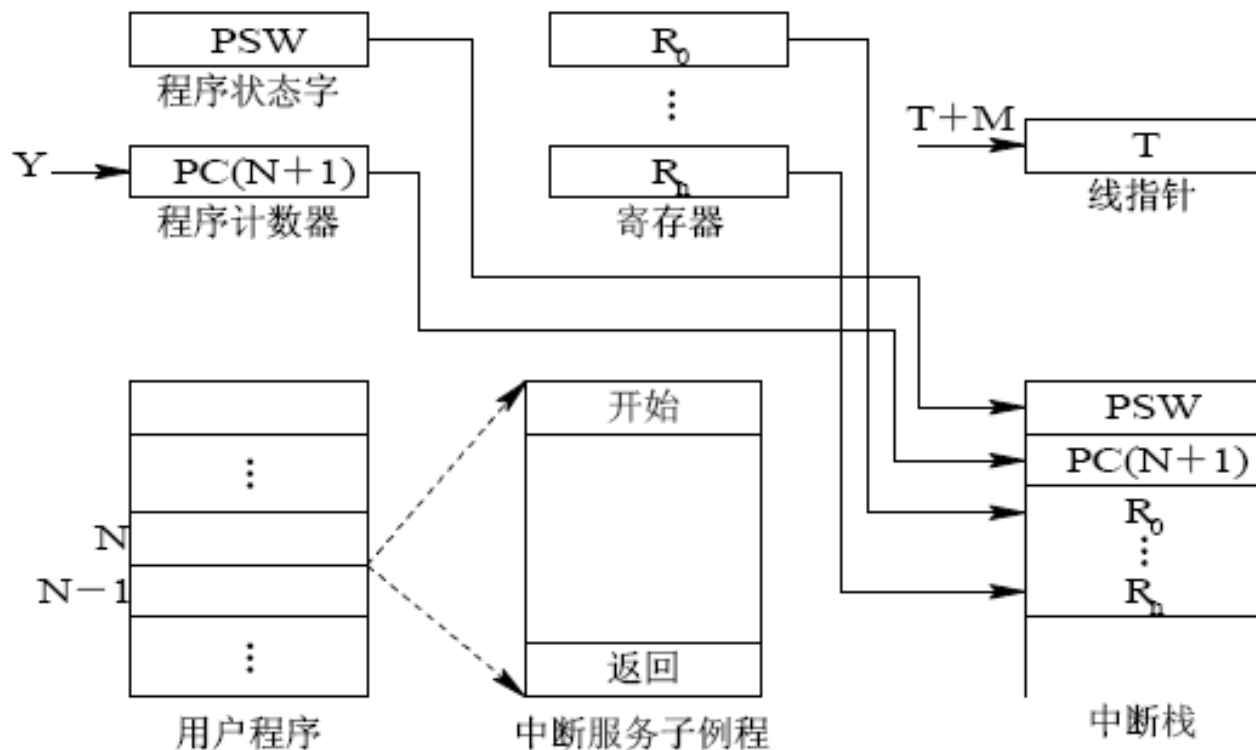
- 通常由**硬件自动**将处理机状态字PSW 和程序计数器(PC)中的内容，保存在中断保留区(栈)中
- 然后把被中断进程的**CPU现场信息**(即包括所有的CPU寄存器，如通用寄存器、段寄存器等内容)都**压入中断栈中**



5.2 中断处理程序

5.2.2 中断处理程序

2、保护被中断进程的运行环境（中断现场）



中断现场保护示意图



5.2 中断处理程序

5.2.2 中断处理程序

3、转入相应的设备处理程序

- 由处理机对各个中断源进行测试，以确定引起本次中断的I/O 设备，并发送一应答信号给发出中断请求的进程，使之消除该中断请求信号
- 然后将相应的设备中断处理程序的入口地址装入到程序计数器中，使处理机转向中断处理程序。



5.2 中断处理程序

5.2.2 中断处理程序

4、执行中断处理

- 该程序首先从设备控制器中读出设备状态，以判断本次中断是否正常完成中断，还是异常结束中断。
- 若是前者，中断程序便进行结束处理；若还有命令，可再向控制器发送新的命令，进行新一轮的数据传送。
- 若是异常结束中断，则根据发生异常的原因做相应的处理。



5.2 中断处理程序

5.2.2 中断处理程序

5、恢复被中断进程的现场

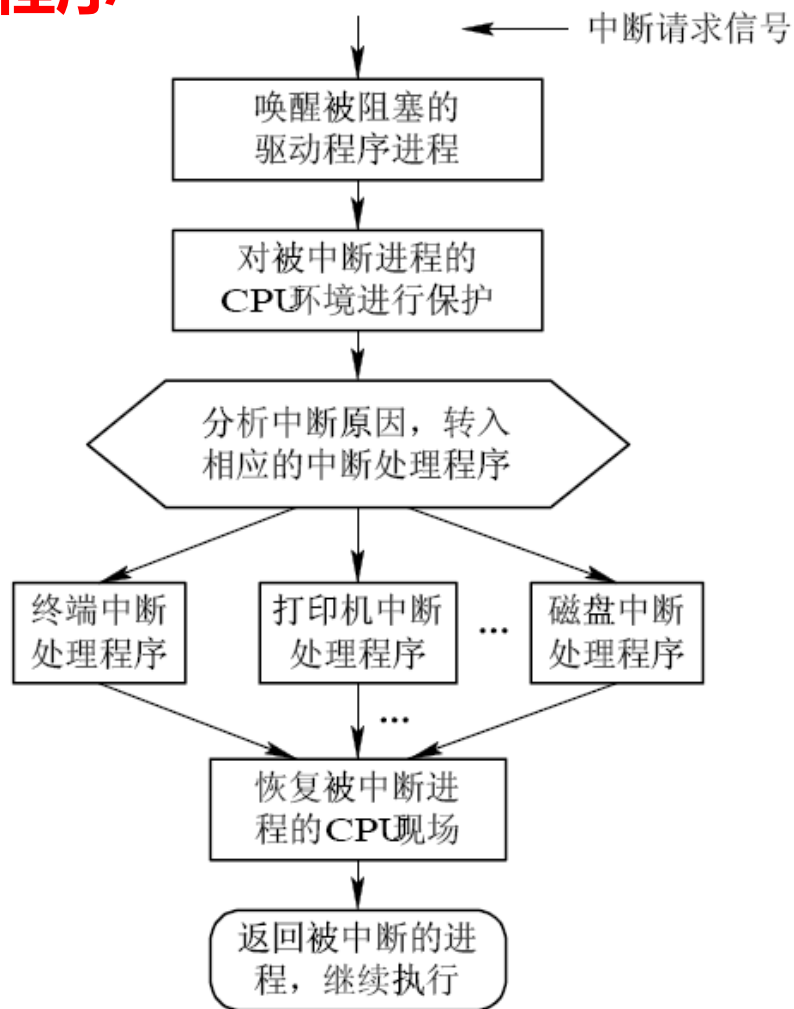
- 当中断处理完成以后，便可将保存在中断栈中的被中断进程的现场信息取出，并装入到相应的寄存器中，其中包括该程序下一次要执行的指令的地址 $N+1$ 、处理机状态字PSW，以及各通用寄存器和段寄存器的内容。
- 这样，当处理机再执行本程序时，便从 $N+1$ 处开始，最终返回到被中断的程序。



5.2 中断处理程序

5.2.2 中断处理程序

中断处理流程





5.3 设备驱动程序

对I/O设备的**控制方式**主要分为以下四种：

- 使用轮询的可编程I/O方式
- 使用中断的可编程I/O方式
- 直接存储器访问方式
- I/O通道控制方式



5.3 设备驱动程序

5.3.1 使用轮询的可编程I/O方式

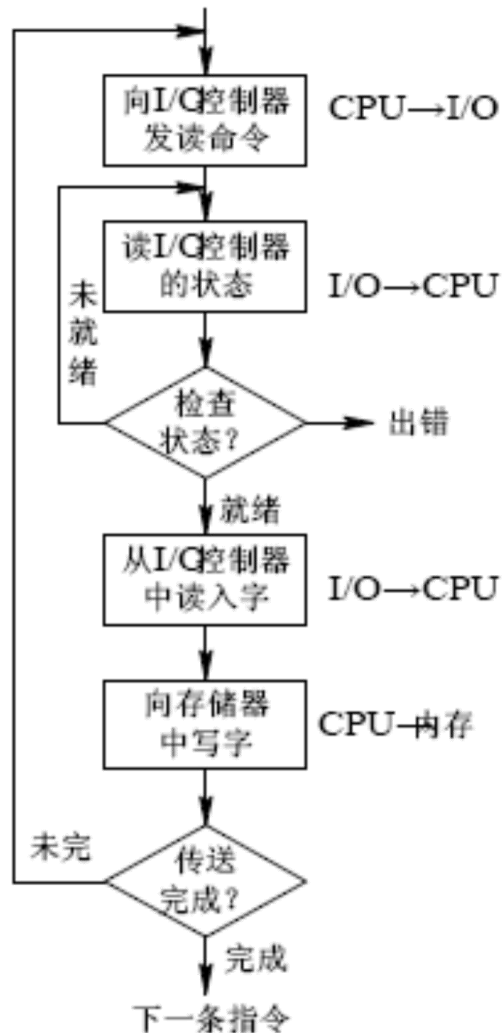
- 程序I/O (Programmed I/O) 方式，或称为**忙--等待**方式。处理机向控制器发出一条I/O指令启动输入设备输入数据时，同时把busy置为1，并不断循环测试busy。
- 当Busy=0，完成输入，处理机读取数据，送入指定单元，完成一次I/O。
- 通过对状态寄存器中的忙/闲标志busy的检查实现控制。



5.3 设备驱动程序

5.3.1 使用轮询的可编程I/O方式 (续)

可编程I/O 方式



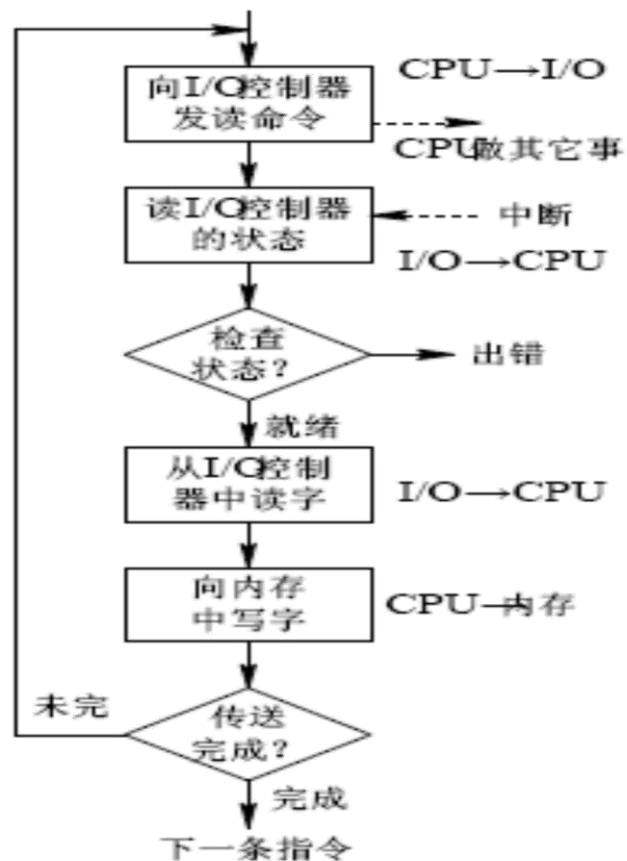


5.3 设备驱动程序

5.3.2 使用中断的可编程I/O方式

- 中断驱动方式可以大幅度地提高CPU的利用率。
- CPU与I/O 设备并行工作。

中断驱动方式





5.3 设备驱动程序

5.3.3 直接存储器访问方式

1) DMA控制方式的引入

- 为了进一步减少CPU对I/O的干预而引入了直接存储器访问方式
- DMA控制方式的特点：
 - ①数据传输的基本单位是数据块，即在CPU与I/O设备之间，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的，或者相反；
 - ③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设备控制器的控制下完成的。

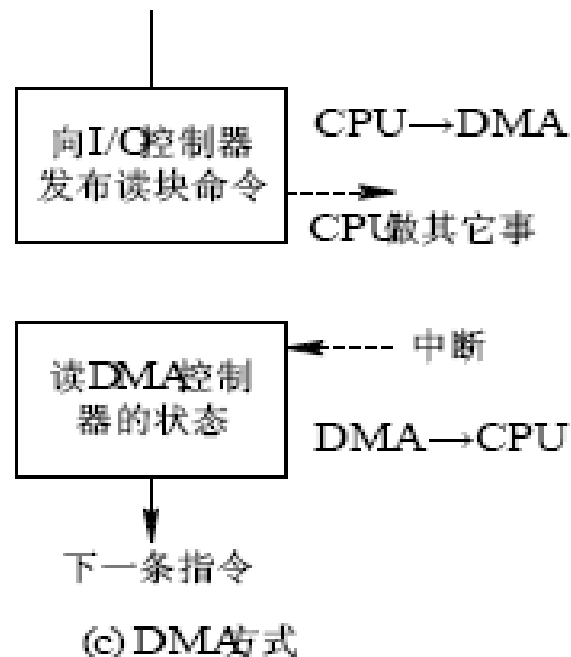


5.3 设备驱动程序

5.3.3 直接存储器访问方式 (续)

1、DMA控制方式的引入

- DMA方式较之中断驱动方式，又是成百倍地减少了CPU对I/O的干预，进一步提高了CPU与I/O设备的并行操作程度。



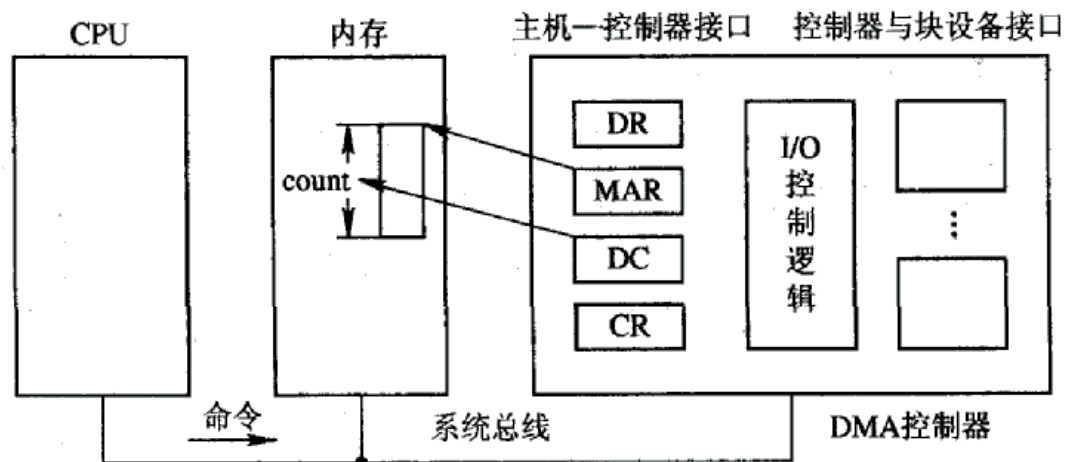


5.3 设备驱动程序

5.3.3 直接存储器访问方式 (续)

2、DMA控制器的组成

- ① 主机与DMA控制器的接口；
- ② DMA控制器与块设备的接口；
- ③ I/O控制逻辑。

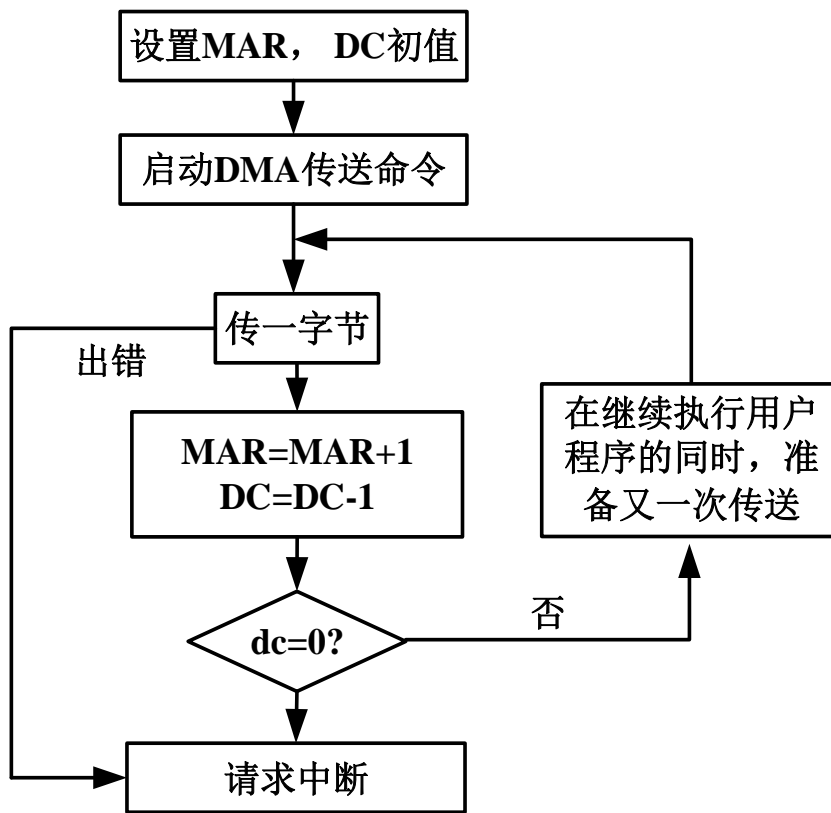




5.3 设备驱动程序

5.3.3 直接存储器访问方式 (续)

3、DMA的工作过程



DMA工作过程



5.3 设备驱动程序

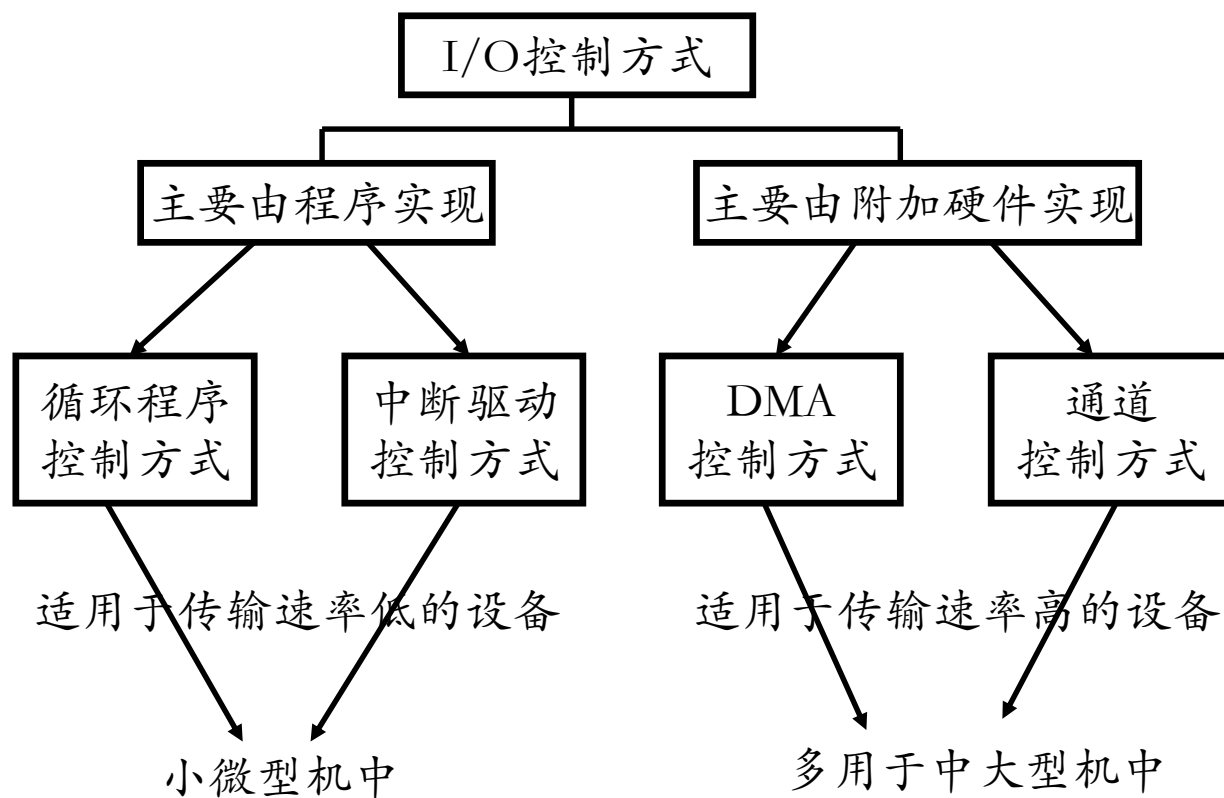
5.3.4 I/O通道控制方式

- I/O通道方式是DMA方式的发展，它可进一步减少CPU的干预，即把对**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）及有关的管理为单位的干预。
- 可实现CPU、通道和I/O设备三者的**并行操作**，从而更有效地提高整个系统的资源利用率。



5.3 设备驱动程序

5.3.5 I/O控制方式小结



I/O控制方式



5.4 磁盘系统与磁盘调度

- 现代计算机系统中，都配置了磁盘存储器，并以它为主来存放文件。
- 对文件的操作，都将涉及到对磁盘的访问，磁盘I/O速度的高低和磁盘系统的可靠性，都将直接影响到整个系统的性能。因此，设法改善磁盘系统的性能，已成为现代操作系统的重要任务之一。



5.4 磁盘系统与磁盘调度

提高磁盘I/O性能的主要途径：

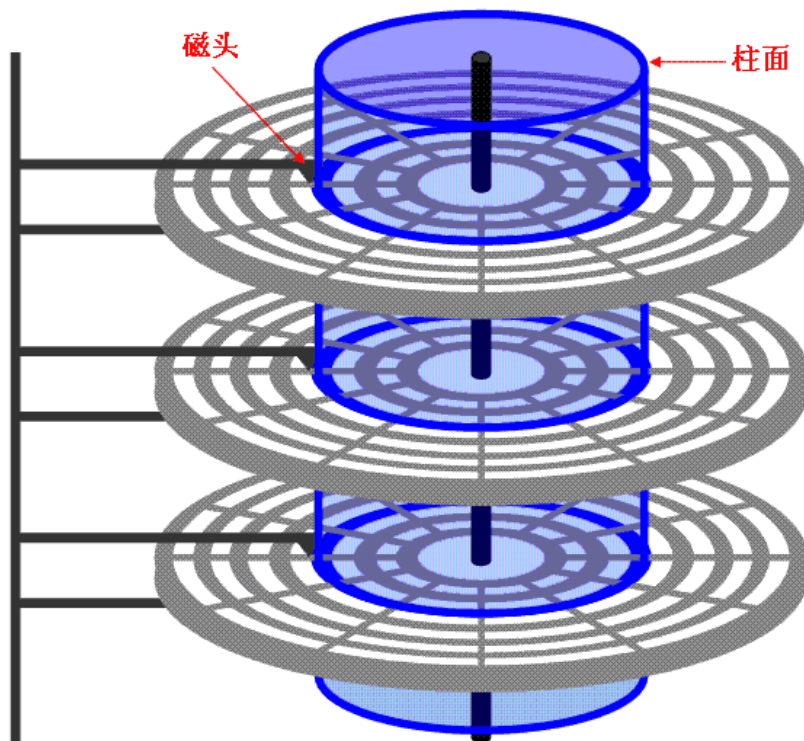
- (1) 选择性能好的**磁盘产品**
- (2) 采用好的**磁盘调度算法**
- (3) 设置**磁盘高速缓存** (Disk Cache)
- (4) 构建**磁盘冗余阵列**
- (5) 其它方法



5.4 磁盘系统与磁盘调度

5.4.1 数据的组织和格式

- 存储面 (surface)
- 磁道 (track)
- 柱面 (cylinder)
- 扇区 (sector)





5.4 磁盘系统与磁盘调度

5.4.2 磁盘的访问时间

磁盘访问时间分成以下三部分：

1) 寻道时间 T_s ：

这是指把磁臂（磁头）移动到指定磁道上所经历的时间。

$$T_s = m \times n + s$$

s ：启动磁臂的时间

n ：磁头移动 n 条磁道

m ：移动每一条磁道所花费的时间



5.4 磁盘系统与磁盘调度

5.4.2 磁盘的访问时间

2) 旋转延迟时间 T_r :

- 这是指定扇区移动到磁头下面所经历的时间。
- 例如：
 - 软盘旋转速度为 300 r/min或600 r/min, 这样, 平均 T_r 为50 ~ 100 ms。
 - 硬盘旋转速度为15 000 r/min, 每转需时4 ms, 平均旋转延迟时间 T_r 为2 ms;



5.4 磁盘系统与磁盘调度

5.4.2 磁盘的访问时间

3) 传输时间 T_t

- 这是指把数据从磁盘读出或向磁盘写入数据所经历的时间。
- T_t 的大小与每次所读/写的字节数 b 和旋转速度有关:

$$T_t = \frac{b}{rN}$$

- r 为磁盘每秒钟的转数; N 为一条磁道上的字节数



5.4 磁盘系统与磁盘调度

5.4.3 磁盘调度

磁盘是可供多个进程共享的设备，当有多个进程都要求访问磁盘时，应采用一种最佳调度算法，以使各进程对磁盘的平均访问时间最小。目前常用的磁盘调度算法有：

1. 先来先服务 (FCFS)
2. 最短寻道时间优先 (SSTF)
3. 扫描算法 (SCAN)
4. 循环扫描算法 (CSCAN)



5.4 磁盘系统与磁盘调度

5.4.3 磁盘调度

1. 先来先服务FCFS

根据进程请求访问磁盘的先后次序进行调度。**此算法的优点**是公平、简单，且每个进程的请求都能依次地得到处理，不会出现某一进程的请求长期得不到满足的情况。

请求序列：55, 58, 39, 18, 90, 160, 150, 38, 184

(从 100 号磁道开始)	
被访问的下一个磁道号	移动距离 (磁道数)
55	45
58	3
39	19
18	21
90	72
160	70
150	10
38	112
184	146
平均寻道长度: 55.3	



5.4 磁盘系统与磁盘调度

5.4.3 磁盘调度

2. 最短寻道时间优先SSTF

- 该算法选择这样的进程，其要求访问的磁道，与当前磁头所在的磁道距离最近，以使每次的寻道时间最短，
- SSTF算法虽然能获得较好的寻道性能，但却可能导致某个进程发生“饥饿”现象。



5.4 磁盘系统与磁盘调度

5.4.3 磁盘调度

2. 最短寻道时间优先SSTF

55
58
39
18
90
160
150
38
184



(从 100 号磁道开始)	
被访问的下一个磁道号	移动距离 (磁道数)
90	10
58	32
55	3
39	16
38	1
18	20
150	132
160	10
184	24
平均寻道长度: 27.5	



5.4 磁盘系统与磁盘调度

5.4.3 磁盘调度

3. 扫描(SCAN)算法

- SSTF算法虽然能获得较好的寻道性能，但却可能导致某个进程发生“饥饿”(Starvation)现象。
- **SCAN算法**优先考虑的是磁头当前的移动方向。例如，磁头自里向外移动，并同时自里向外地访问，直至再无更外的磁道需要访问时，才将磁臂换向自外向里移动。
(又常称之为电梯调度算法)



5.4 磁盘系统与磁盘调度

5.4.3 磁盘调度

3. 扫描(SCAN)算法

55
58
39
18
90
160
150
38
184



(从 100#磁道开始, 向磁道号增加方向访问)	
被访问的下一个磁道号	移动距离 (磁道数)
150	50
160	10
184	24
90	94
58	32
55	3
39	16
38	1
18	20
平均寻道长度: 27.8	



5.4 磁盘系统与磁盘调度

5.4.3 磁盘调度

4. 循环扫描(CSCAN)算法

- CSCAN算法规定磁头**单向移动**，例如，只是自里向外移动，当磁头移到最外的磁道并访问后，磁头立即返回到最里的欲访问磁道，亦即将最小磁道号紧接着最大磁道号构成循环，进行循环扫描。



5.4 磁盘系统与磁盘调度

5.4.3 磁盘调度

4. 循环扫描(CSCAN)算法

55
58
39
18
90
160
150
38
184



(从 100#磁道开始, 向磁道号增加方向访问)	
被访问的下一个磁道号	移动距离 (磁道数)
150	50
160	10
184	24
18	166
38	20
39	1
55	16
58	3
90	32
平均寻道长度: 35.8	