

第六章 输入输出系统

6.1 I/O系统的功能、模型和接口

6.2 I/O设备和设备控制器

6.3 中断机构和中断处理程序

6.4 设备驱动程序

6.5 与设备无关的I/O软件

6.6 用户层的I/O软件

6.7 缓冲区管理

6.8 磁盘存储器的性能和调度

6.1 I/O系统的功能、模型和接口

第六章 输入输出系统

- ❑ I/O系统是计算机系统的重要组成部分
- ❑ I/O系统包括用于实现信息输入、输出和存储功能的设备和相应的控制器
- ❑ 设备管理的基本任务是完成用户提出的I/O请求，提高I/O速率以及改善I/O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

6.1 I/O系统的功能、模型和接口

第六章 输入输出系统

I/O系统的基本功能

- 方便用户使用I/O设备
- 提高CPU和I/O设备的利用率
- 为用户在共享设备时提供方便
- 1、隐藏物理设备的细节
- 2、与设备的无关性
- 3、提供处理机和I/O设备的利用率（并行操作）
- 4、对I/O设备进行控制（四种控制方式）
- 5、确保对设备的正确共享（设备的共享属性）
- 6、错误处理

3

6.1 I/O系统的功能、模型和接口

第六章 输入输出系统

I/O系统的层次结构和模型

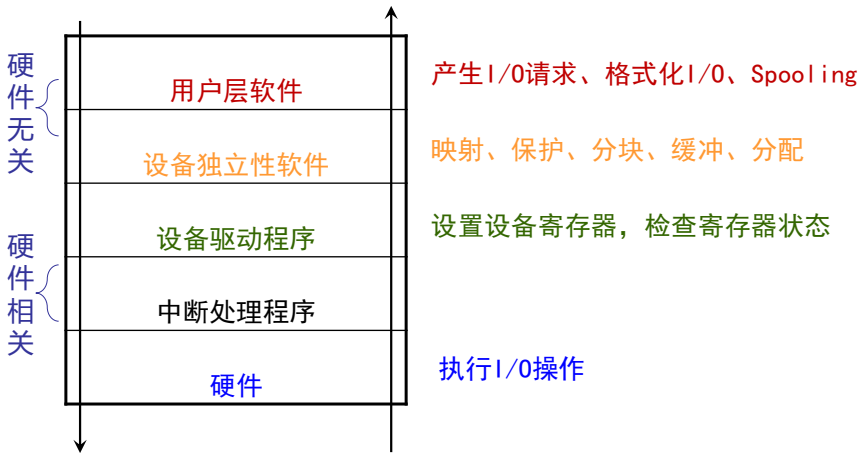
- I/O系统普遍采用层次式结构
- 将系统中的设备管理模块分为若干个层次
- 每一层都是利用其下层提供的服务
- 每一层完成输入输出功能中的某些子功能，并屏蔽这些功能实现的细节，向高层提供服务。

4

6.1 I/O系统的功能、模型和接口

第六章 输入输出系统

I/O系统的层次结构和模型



5

6.1 I/O系统的功能、模型和接口

第六章 输入输出系统

I/O系统的层次结构和模型

(1) 用户层软件

实现与用户交互的接口，用户可直接调用在用户层提供的、与I/O操作有关的库函数，对设备进行操作

(2) 设备独立软件

实现用户程序与设备驱动器的统一接口、设备命名、设备的保护以及设备的分配与释放等，同时为设备管理和数据传送提供必要的存储空间

(3) 设备驱动程序

具体实现系统对设备发出的操作指令，驱动I/O设备工作

(4) 中断处理程序

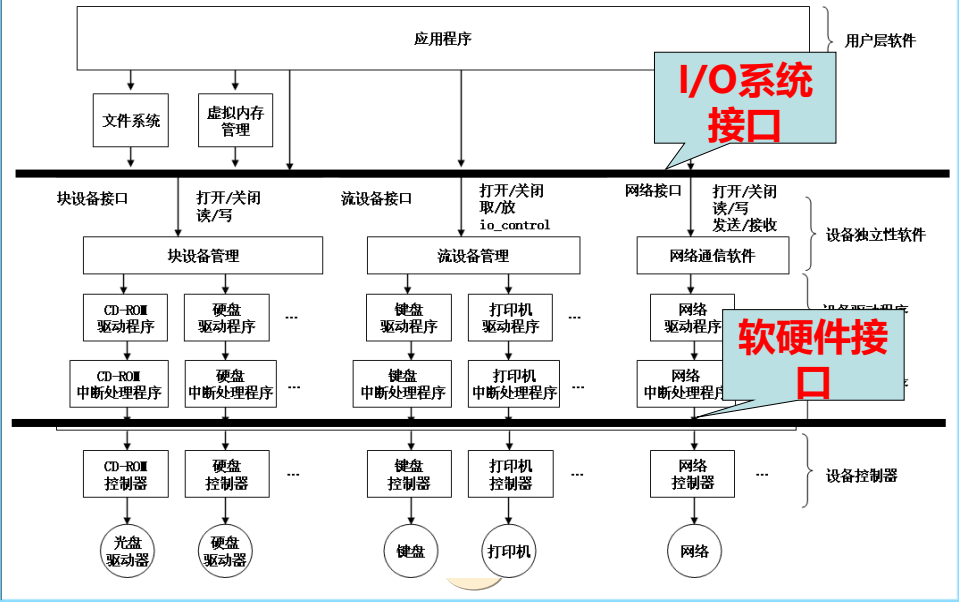
保存被中断进程的CPU环境，转入中断处理程序，处理完后恢复被中断进程的现场，返回被中断进程

6

6.1 I/O系统的功能、模型和接口

第六章 输入输出系统

I/O系统中各种模块之间的层次视图



6.1 I/O系统的功能、模型和接口

第六章 输入输出系统

I/O系统的层次结构和模型

- 中断处理程序
 - 处于I/O系统的底层，直接与硬件进行交互
- 设备驱动程序
 - 处于次底层，是进程和控制器之间的通信程序
 - 功能：将上层发来的抽象I/O请求，转换为对I/O设备的具体命令和参数，并把它装入到设备控制其中的命令和参数寄存器中
- 设备独立性软件

6.1 I/O系统的功能、模型和接口

第六章 输入输出系统

I/O系统接口

I/O系统与高层接口根据设备类型的不同分为若干个类型：

(1) 块设备接口

控制磁盘存储器和光盘存储器设备的输入或输出

(2) 流设备接口

控制字符设备的输入或输出

(3) 网络通信接口

提供相应的网络软件和网络通信接口

9

6.1 I/O系统的功能、模型和接口

第六章 输入输出系统

I/O系统接口

➤ 块设备接口

- ✓ 信息的存取以**数据块**为单位，有结构设备
- ✓ 传输速率较高，通常每秒钟为几兆位，可**寻址**，即对它可随机地读/写任一块。I/O采用**DMA方式**
- ✓ 例：磁盘

➤ 字符设备接口（流设备接口）

- ✓ 基本单位是**字符**，无结构设备
- ✓ 传输速率较低，通常每秒钟为几个字节到数千字节，不可寻址。I/O采用**中断驱动方式**
- ✓ 例：交互式终端、打印机

10

6.2 I/O设备和设备控制器

第六章 输入输出系统

- ✓ I/O设备
- ✓ 设备控制器
- ✓ 内存映像I/O
- ✓ I/O通道

11

6.2 I/O设备和设备控制器

第六章 输入输出系统

I/O设备

- 执行I/O操作的机械部分，执行控制I/O的电子部件
- 执行I/O操作的机械部分——一般的**I/O设备**
- 执行控制I/O的电子部件则称为**设备控制器**或适配器
- 在微型机和小型机中的控制器常做成印刷电路卡形式，因而也常称为控制卡、接口卡或网卡
- 大、中型计算机系统中，还配置了**I/O通道**或**I/O处理机**

12

6.2 I/O设备和设备控制器

第六章 输入输出系统

I/O设备的类型

❖ 按传输速率分类

- 低速设备
 - ✓ 每秒几个字节至数百字节
 - ✓ 键盘、鼠标、语音输入输出设备等
- 中速设备
 - ✓ 每秒数千至数万字节
 - ✓ 行式打印机、激光打印机等
- 高速设备
 - ✓ 每秒数百K至数十M字节
 - ✓ 磁盘机、磁带机、光盘机等

13

6.2 I/O设备和设备控制器

第六章 输入输出系统

I/O设备的类型

❖ 按信息交换的单位分类

- 块设备(Block Device)
 - ✓ 信息的存取总是以数据块为单位
 - ✓ 可寻址
 - ✓ 磁盘, 每个盘块的大小为512 B~4 KB
- 字符设备(Character Device)
 - ✓ 基本单位是字符
 - ✓ 不可寻址
 - ✓ 交互式终端、打印机

14

6.2 I/O设备和设备控制器

第六章 输入输出系统

I/O设备的类型

块设备 (Block Device)

- 用于存储以数据块为单位的信息
- 属于有结构设备
- 典型的块设备是**磁盘**，每个盘块的大小为512B~4KB
- 磁盘设备的基本特征
 - **传输速率较高**，每秒钟为几兆位
 - **可寻址**，即对它可随机地读/写任一块
 - 磁盘设备的I/O常采用**DMA方式**

15

6.2 I/O设备和设备控制器

第六章 输入输出系统

I/O设备的类型

字符设备 (Character Device)

- 用于数据**输入和输出**基本单位是**字符**
- 属于**无结构设备**
- **交互式终端、打印机等**
- 字符设备的基本特征：
 - **传输速率较低**
 - **不可寻址**（不能指定输入源地址及输出目标地址）
 - 字符设备在I/O时，常采用**中断驱动方式**

16

6.2 I/O设备和设备控制器

第六章 输入输出系统

I/O设备的类型

❖ 按资源分配角度分类（共享属性）

- 独占设备
 - ✓ 多数低速设备属独占设备，如打印机
- 共享设备
 - ✓ 可供多个进程同时访问，如磁盘
- 虚拟设备
 - ✓ 通过虚拟技术将一台独占设备变换为若干个逻辑设备，供若干个进程同时使用

17

6.2 I/O设备和设备控制器

第六章 输入输出系统

设备与控制器之间的接口

- 设备并不是直接与CPU进行通信，而是与设备控制器通信。
- 在设备与设备控制器之间应有一接口

18

6.2 I/O设备和设备控制器

第六章 输入输出系统

设备与控制器之间的接口

接口中有三种类型的信号，各对应一条线

(1) 数据信号线

– 用于在设备和设备控制器之间传送数据信号

(2) 控制信号线

– 作为由设备控制器向I/O设备发送控制信号（规定设备应执行的操作）时的通路

(3) 状态信号线

– 用于传送指示设备当前状态的信号

19

6.2 I/O设备和设备控制器

第六章 输入输出系统

设备控制器

- ❑ 设备控制器是CPU与I/O设备之间的接口
- ❑ 接收CPU发来的命令，控制一个或多个I/O设备工作，以实现I/O设备和计算机之间的数据交换，减轻CPU的负担
- ❑ 设备控制器是一个可编址的设备
- ❑ 设备控制器分类
 - ❖ 控制字符设备
 - ❖ 控制块设备

20

6.2 I/O设备和设备控制器

第六章 输入输出系统

设备控制器的基本功能

- 接收和识别命令
应有**控制寄存器**存放接受的命令和参数，并对其译码
- 数据交换
实现CPU与控制器、控制器与设备之间的数据交换，需设置**数据寄存器**
- 标识和报告设备的状态：需设置**状态寄存器**
- 地址识别
识别所控制的设备，需设置**地址译码器**
- 数据缓冲：解决I/O设备速率低的问题
- 差错控制：对I/O设备发送的数据

21

6.2 I/O设备和设备控制器

第六章 输入输出系统

设备控制器的组成

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

22

6.2 I/O设备和设备控制器

第六章 输入输出系统

设备控制器的组成

1) 设备控制器与处理机的接口

- 实现CPU与设备控制器之间通信
- 三类信号线
 - ✓ 数据线：连接
 - 数据寄存器：存放设备或CPU送来的数据
 - 控制/状态寄存器：存放CPU送来的控制信息或设备的状态信息
 - ✓ 地址线
 - ✓ 控制线

23

6.2 I/O设备和设备控制器

第六章 输入输出系统

设备控制器的组成

2) 设备控制器与设备的接口

- 一个设备控制器可连接一个或多个设备
- 控制器中则有一个或多个设备接口
- 接口中存在数据、控制、状态信号
- I/O逻辑根据处理机发来的地址信号选择一个设备接口

24

6.2 I/O设备和设备控制器

第六章 输入输出系统

设备控制器的组成

3) I/O逻辑：实现对设备的控制

- 处理机利用它向控制器发送I/O命令
- I/O逻辑对接受到的命令进行译码

每当CPU要启动一个设备时，发一方面将启动命令发送给控制器；另一方面又同时通过地址线把地址发送给控制器，由控制器的I/O逻辑对收到的地址进行译码，再根据所译出的命令对所选设备进行控制。

25

6.2 I/O设备和设备控制器

第六章 输入输出系统

内存映像I/O

- 驱动程序将抽象I/O命令，转换出的一系列具体的命令、参数等数据，装入设备控制器的相应寄存器，由控制器来执行这些命令，具体实施对I/O设备的控制
- 两种方法：
 - 利用特定的I/O指令
 - 内存映像I/O

26

6.2 I/O设备和设备控制器

第六章 输入输出系统

I/O通道

➤ 引入目的

建立独立I/O操作, 不仅使数据的传送能独立于CPU, 而且使CPU从对I/O操作的组织、管理中解脱

➤ 实现:

CPU只需发送I/O命令给通道, 通道通过调用内存中的相应通道程序完成任务

➤ 特殊的处理机:

- 指令类型单一, 只用于I/O操作
- 通道没有内存, 它与CPU共享内存

27

6.2 I/O设备和设备控制器

第六章 输入输出系统

I/O通道

通道类型

1. 字节多路通道(Byte Multiplexor Channel) :

- 按字节交叉方式工作
- 每个子通道连接一台I/O设备, 并控制该设备的I/O操作
- 各子通道以时间片轮转方式共享主通道, 适用于低、中速设备

28

6.2 I/O设备和设备控制器

第六章 输入输出系统

I/O通道

通道类型

2. 数组选择通道(Block Selector Channel) :

- 以数组方式进行数据传送
- 仅一通道，某时间由某设备独占，适于高速设备
- 但通道未共享，利用率低

29

6.2 I/O设备和设备控制器

第六章 输入输出系统

I/O通道

通道类型

3. 数组多路通道(Block Multiplexor Channel) :

- 将数组选择通道传输速率高和字节多路通道分时并行操作的优点相结合而形成的一种新通道
- 含多个非分配型子通道，因而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
- 广泛的用于连接多台高、中速的外围设备，其数据传送方式是按数组方式进行的

30

6.2 I/O设备和设备控制器

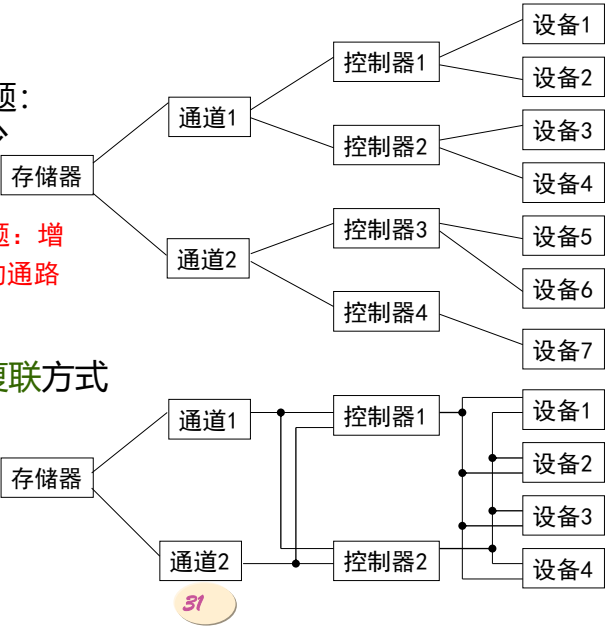
第六章 输入输出系统

I/O通道

通道“瓶颈”问题：
价格贵数量少

解决“瓶颈”问题：增
加设备到主机间的通路
而不是增加通道

解决：采用复联方式



6.3 中断机构和中断处理程序

第六章 输入输出系统

中断机构

- 中断指CPU对I/O设备发来的中断信号的一种响应
- 中断是外部设备引起的，又称外中断
- 陷入由CPU内部事件所引起的中断，称为内中断
- 主要区别是信号的来源（CPU外部/内部）

6.3 中断机构和中断处理程序

第六章 输入输出系统

中断机构

中断向量表

为每种设备配以相应的中断处理程序，并把该程序的入口地址，放在中断向量表的一个表项中，并为每一个设备的中断请求，规定一个中断号，它直接对应于中断向量表的一个表项中。

33

6.3 中断机构和中断处理程序

第六章 输入输出系统

中断机构

中断优先级

系统根据不同中断信号源，对服务要求的紧急程度的不同，分别规定不同的优先级

磁盘中断请求高于打印机中断请求高于键盘终端请求

34

6.3 中断机构和中断处理程序

第六章 输入输出系统

中断机构

对多中断源的处理方式

➤ 屏蔽（禁止）中断

- 处理机对任何新到的中断请求，都暂时不予理睬，而让它们等待
- 所有中断都将按顺序依次处理
- 优点简单，但不能用于对实时性要求较高的中断请求

➤ 嵌套中断

- CPU优先响应最高优先级的中断请求
- 高优先级的中断请求可以抢占正在运行的低优先级中断的处理机

35

6.3 中断机构和中断处理程序

第六章 输入输出系统

中断处理程序

中断处理程序的处理过程

- 检查是否有未响应的中断信号
- 唤醒正处于阻塞的驱动(程序)进程
- 保护被中断进程的CPU环境
- 转入相应的设备处理程序
- 中断处理
- 复CPU的现场并退出中断

36

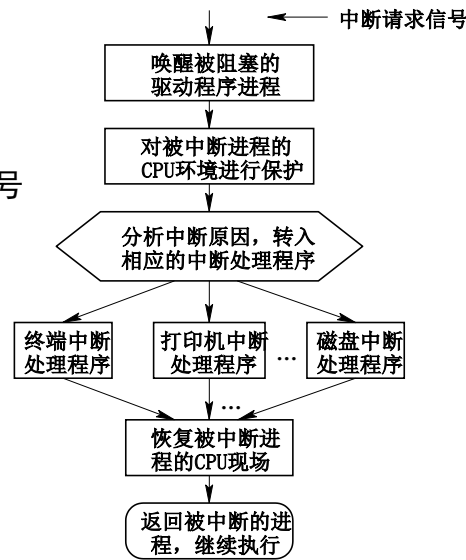
6.3 中断机构和中断处理程序

第六章 输入输出系统

中断处理程序

中断处理流程

- 测定是否有未响应的中断信号
- 保护被中断进程的CPU环境
- 转入相应的设备处理程序
- 中断处理
- 恢复CPU的现场



37

6.3 中断机构和中断处理程序

第六章 输入输出系统

中断处理程序

- ①进行进程上下文的切换
- ②对处理中断信号源进行测试
- ③读取设备状态
- ④修改进程状态

38

6.4 设备驱动程序

第六章 输入输出系统

- ✓ 设备驱动程序概述
- ✓ 设备驱动程序的处理过程
- ✓ I/O设备的控制方式

39

6.4 设备驱动程序

第六章 输入输出系统

设备驱动程序概述

□ 设备处理程序

I/O进程与设备控制器之间的通信程序

□ 主要任务

- 接受来自上层软件的抽象的I/O命令
- 把它转换成具体要求发送给设备控制器
- 启动设备进行数据传送
- 将由设备控制器发来的信号，传送给上层软件

40

6.4 设备驱动程序

第六章 输入输出系统

设备驱动程序概述

驱动程序的功能

- (1) 接收由与设备无关的软件发来的命令和参数，并将命令中的抽象要求，转换为与设备相关的低层操作序列；
- (2) 检查用户I/O请求的合法性，了解I/O设备的工作状态，传递与I/O设备操作有关的参数，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，完成指定的I/O操作；如果设备忙碌，则将请求者挂在设备队列上等待；
- (4) 及时响应由设备控制器发来的中断请求，并根据其中断类型，调用相应的中断处理程序进行处理。

41

6.4 设备驱动程序

第六章 输入输出系统

设备驱动程序概述

设备驱动程序的特点

- (1) 与设备无关的软件和设备控制器之间通信和转换的程序
- (2) 与设备控制器和I/O设备的硬件特性紧密相关
- (3) 驱动程序与I/O设备所采用的I/O控制方式紧密相关
- (4) 驱动程序部分必须用汇编语言编写
- (5) 驱动程序允许可重入，一个正在运行的驱动程序常会在一次调用完成前被再次调用

42

6.4 设备驱动程序

第六章 输入输出系统

设备处理方式

根据在设备处理时是否设置进程，以及设置什么样的进程，而把设备处理方式分成以下三类：

- 为每一类设备设置一个进程，专门用于执行这类设备的I/O操作
- 在整个系统中设置一个I/O进程，专门用于执行系统中所有各类设备的I/O操作
- 不设置专门的设备处理进程，而只为各类设备设置相应的设备驱动程序，供用户或系统进程调用

43

6.4 设备驱动程序

第六章 输入输出系统

设备处理方式

处理过程

- 主要任务是启动指定设备，完成上层指定的I/O工作

➤ 启动过程

将抽象要求转化为具体要求
检查I/O请求合法性
读出和检查设备状态
传送必要的参数
设置工作方式
启动I/O设备

在完成上述各项准备工作后，驱动程序可以向控制器的命令寄存器传送相应的命令，启动I/O设备开始I/O操作。

发出I/O命令后，基本的I/O操作是在设备控制器下进行的，此时驱动(程序)进程把自己阻塞起来，直到I/O中断到来时才被唤醒

44

6.4 设备驱动程序

第六章 输入输出系统

设备处理方式

对I/O设备的控制方式

➤ I/O控制方式的宗旨

尽量减少主机对I/O控制的干预，把主机从繁杂的I/O控制事务中解脱出来，以便更多地去完成数据处理任务。

➤ 方式

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

45

6.4 设备驱动程序

第六章 输入输出系统

设备处理方式

(1) 程序I/O方式

□ 查询方式：

- CPU需花代价不断查询I/O状态
- 状态寄存器中的忙/闲标志busy
- CPU资源浪费极大，绝大部分时间都处于等待
- I/O设备完成数据I/O的循环测试中

注：程序I/O方式也称作程序查询方式或轮询方式

46

6.4 设备驱动程序

第六章 输入输出系统

设备处理方式

(2) 中断驱动I/O方式

- 中断驱动方式：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
- 设备控制器按命令要求去控制指定的I/O设备，完成后，通过中断向CPU发送一中断信号
- 在I/O设备输入数据的过程中，无须CPU干预
- 每次传送一个字符，CPU与I/O设备并行操作

47

6.4 设备驱动程序

第六章 输入输出系统

设备处理方式

(2) 中断驱动I/O方式

在I/O中断方式下，数据的输入(或输出)步骤如下：

- 1、要求输入数据的进程把一个启动命令和允许中断位“1”写入相应设备的控制状态寄存器中，从而启动了该设备
- 2、该进程因等待输入的完成进入睡眠状态
- 3、当输入完成后，输入设备向CPU发出完成中断请求信号
- 4、处理机响应中断，处理该中断，并唤醒等待输入完成的进程
- 5、在以后的某个时期，该程序被调度到后继续运行

48

6.4 设备驱动程序

第六章 输入输出系统

设备处理方式

(2) 中断驱动I/O方式

- 向I/O发命令——返回——执行原来任务
- 设备控制器控制指定I/O设备——完成任务
- I/O产生中断——CPU转相应中断处理程序

如：

读数据
读完后以中断方式通知CPU
CPU完成数据从I/O——内存

49

6.4 设备驱动程序

第六章 输入输出系统

设备处理方式

(2) 中断驱动I/O方式

■ CPU

- 计算
- 启动设备
- 计算
- ...
- 计算
- 中断处理
- 计算

设备：

工
作

特点： CPU与设备并行工作；
当设备多时，对CPU打扰多

50

6.4 设备驱动程序

第六章 输入输出系统

设备处理方式

(3) 直接存储器访问DMA I/O控制方式

特点:

- 1) 数据传输的基本单位是数据块
- 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控制器的控制下完成的

51

6.4 设备驱动程序

第六章 输入输出系统

DMA控制器

DMA控制器由三部分组成

- (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

52

6.4 设备驱动程序

第六章 输入输出系统

DMA控制器

1. CPU设置DMA控制器实现DMA编程，同时启动磁盘控制器由磁盘读入数据到内部缓冲区
2. DMA控制器向磁盘控制器发出读请求
3. 磁盘控制器将字节传到内存指定单元
4. 磁盘控制器向DMA控制器发送回答
5. DMA控制器将MAR加1，DC减1，重复上述过程直至DC为0，DMA控制器向CPU发出中断请求

53

6.4 设备驱动程序

第六章 输入输出系统

DMA控制方式与中断控制方式的区别

- 中断控制方式在**每个数据传送完成**后中断CPU，而DMA控制方式则是在所要求传送的**一批数据全部传送**结束时才中断CPU
- 中断控制方式的数据传送是在中断处理时由**CPU控制**完成，而DMA控制方式则是在**DMA控制器**的控制下完成的

54

6.4 设备驱动程序

第六章 输入输出系统

I/O通道控制方式

- **DMA方式**：对需多个离散数据块的读取仍需要多次中断
- **I/O通道控制方式**是DMA方式的发展，减少为对**一组数据块**的读（写）及有关的控制和管理为单位的干预。同时可以实现CPU、通道和I/O设备三者并行操作
- **通道方式**：CPU只需给出
 - (1) 通道程序首址
 - (2) 要访问I/O设备

55

6.4 设备驱动程序

第六章 输入输出系统

I/O通道控制方式

- 通道是通过执行通道程序，并与设备控制器共同实现对I/O设备的控制的
- 通道程序由一系列通道指令所构成的
- 通道指令一般包含下列信息：
 - 操作码**——规定指令所执行的操作
 - 内存地址**
 - 计数**——表示本指令所要操作的字节数
 - 通道程序结束位**——用以表示程序是否结束
 - 记录结束标志**——表示该指令是否与下条指令有关

56

6.4 设备驱动程序

第六章 输入输出系统

DMA控制方式与通道控制方式的区别

- DMA控制方式中需要CPU来控制所传输数据块的大小、传输的内存，而通道控制方式中这些信息都是由通道来控制管理
- 一个DMA控制器对应一台设备与内存传递数据，而一个通道可以控制多台设备与内存的数据交换

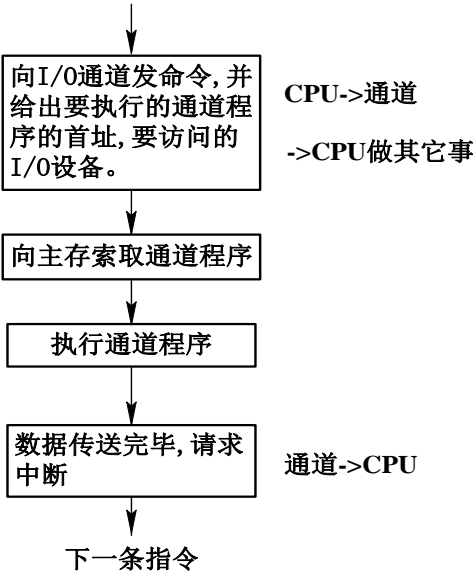
57

6.4 设备驱动程序

第六章 输入输出系统

I/O通道控制方式

通道的工作流程



58

6.5 与设备无关的I/O软件

第六章 输入输出系统

- ✓ 基本概念
- ✓ 实现与设备无关的软件
- ✓ 设备分配
- ✓ 逻辑设备名到物理设备名映射的实现

59

6.5 与设备无关的I/O软件

第六章 输入输出系统

与设备无关软件的基本概念

- ❖ 设备独立性的概念
 - 也称为设备无关性
 - 含义：应用程序独立于具体使用的物理设备，即是指用户在编程序时所使用的设备与实际设备无关
 - 引入逻辑设备和物理设备这两个概念
 - 在应用程序中，使用逻辑设备名称来请求使用某类设备；而系统在实际执行时，以物理设备名称来使用设备
 - 系统须具有将逻辑设备名称转换为某物理设备名称的功能，这非常类似于存储器管理中所介绍的逻辑地址和物理地址的概念

60

6.5 与设备无关的I/O软件

第六章 输入输出系统

与设备无关软件的基本概念

设备独立性

(Device Independence)

——应用程序独立于具体使用的物理设备
方便用户编程、便于程序移植

物理设备和逻辑设备：

类似于物理地址和逻辑地址的概念

使用逻辑设备名称来请求使用某类设备

系统实际执行时，必须使用物理设备名称

61

6.5 与设备无关的I/O软件

第六章 输入输出系统

与设备无关软件的基本概念

❖ 设备独立性的优点

* 设备分配时的灵活性

- 系统可将该逻辑设备类中的任一台分配给进程使用
- 所有设备均占用时才阻塞

* 易于实现I/O重定向

- 所谓I/O重定向, 指用于I/O操作的设备可以更换, 而不必变应用程序
- 如调试程序时输出到屏幕, 而实际应用时改为输出到打印机(逻辑设备表中的显示终端改为打印机)

62

6.5 与设备无关的I/O软件

第六章 输入输出系统

实现设备无关软件

设备独立性软件

执行所有设备的公有操作、向用户层（或文件层）软件提供统一的接口

- 1) 设备驱动程序的统一接口
 - 将逻辑设备名映射为物理设备名，进一步可以找到相应物理设备的驱动程序
 - 对设备进行保护，禁止用户直接访问设备
- 2) 缓冲管理
- 3) 差错控制
- 4) 独占设备的分配与回收
- 5) 独立于设备的逻辑数据块

63

6.5 与设备无关的I/O软件

第六章 输入输出系统

实现设备无关软件

设备分配内容

包括：对设备、设备控制器、通道的分配

- (1) 设备分配中的数据结构
- (2) 设备分配时应考虑的因素
- (3) 独占设备的分配程序

64

6.5 与设备无关的I/O软件

第六章 输入输出系统

实现设备无关软件

设备分配中的数据结构

- 设备控制表 (DCT)
- 控制器控制表 (COCT)
- 通道表 (CHCT)
- 系统设备表 (SDT)

65

6.5 与设备无关的I/O软件

第六章 输入输出系统

设备分配

设备分配时应考虑的因素

- 设备的固有属性
- 设备分配算法
- 设备分配中的安全性

66

6.5 与设备无关的I/O软件

第六章 输入输出系统

设备分配

设备的固有属性

- 独占设备: 在一段时间内只能由一个进程使用
- 共享设备: 允许多个进程共享
- 虚拟设备: 是经过某种处理由独占设备变为虚拟设备

67

6.5 与设备无关的I/O软件

第六章 输入输出系统

设备分配

设备分配算法

■ 先来先服务

根据请求的先后次序排成一个队列，设备总是分配给队首进程

■ 优先级高者优先

利用该算法形成队列时，将优先权高的进程安排在设备队列前面，优先级相同的先来先服务

68

6.5 与设备无关的I/O软件

第六章 输入输出系统

设备分配

设备的固有属性

- 独占设备: 在一段时间内只能由一个进程使用
- 共享设备: 允许多个进程共享
- 虚拟设备: 是经过某种处理由独占设备变为虚拟设备

69

6.5 与设备无关的I/O软件

第六章 输入输出系统

设备分配

设备分配算法

■ 先来先服务

根据请求的先后次序排成一个队列，设备总是分配给队首进程

■ 优先级高者优先

利用该算法形成队列时，将优先权高的进程安排在设备队列前面，优先级相同的先来先服务

70

6.5 与设备无关的I/O软件

第六章 输入输出系统

设备分配

设备分配中的安全性

■ 安全分配方式

- 每当进程发出I/O请求后便阻塞，直到I/O完成后被唤醒
- 优点：摒弃了“请求和保持”条件，不会产生死锁虽安全但缓慢

■ 不安全分配方式

- 不断发出I/O请求到所请求设备被另一进程占用才阻塞
- 优点：可操作多个设备，推进迅速
- 虽迅速但不安全

缺点：可能产生死锁

71

6.5 与设备无关的I/O软件

第六章 输入输出系统

设备分配

独占设备的分配程序

1) 基本的设备分配程序

- 分配设备
- 分配控制器
- 分配通道

只有在设备、控制器和通道三者都分配成功时，设备分配才算成功，系统可启动该设备进行数据传送

72

6.5 与设备无关的I/O软件

第六章 输入输出系统

设备分配

独占设备的分配程序

2) 基本设备分配程序存在的问题

- 进程以物理设备名提出I/O请求
- 采用单通路的I/O系统结构，产生“瓶颈”

3) 设备分配程序的改进

- 增加设备的独立性
- 考虑多通路情况

73

6.5 与设备无关的I/O软件

第六章 输入输出系统

设备分配

独占设备的分配程序

依次获得了设备、设备控制器和通道，进程才具备I/O条件

例：某进程申请Printer来进行打印

查SDT得知打印机的DCT

打印机忙则阻塞

打印机闲则分配①

查打印机DCT得知打印机控制器的COCT

控制器忙则阻塞

控制器闲则分配②

查打印机控制器COCT得知通道的CHCT

通道忙则阻塞

通道闲则分配③

仅当①→②→③依次成立的情况下，才表明此次分配成功！

74

6.6 用户层的I/O软件

第六章 输入输出系统

- ✓ 系统调用与库函数
 - 系统调用 (system call)
 - 库函数 (library functions)
- ✓ 假脱机 (Spooling) 系统
 - 假脱机技术
 - SPooling的组成
 - SPooling系统的特点
 - 假脱机打印机系统

75

6.6 用户层的I/O软件

第六章 输入输出系统

系统调用与库函数

库函数

- 用户程序通过调用对应的库函数使用系统调用
- 对于I/O方面，主要是对文件和设备进行读写，以及控制/检查设备状态的库函数，其集合也是I/O系统的组成部分
- 内核提供了OS的基本功能，而库函数扩展了OS内核，使用户能方便取得操作系统的服务

76

6.6 用户层的I/O软件

第六章 输入输出系统

假脱机（SPooling）系统

- 脱机输入、输出技术
- 在主机的直接控制下, 实现脱机输入、输出功
- 外围操作与CPU对数据的处理同时进行

77

6.6 用户层的I/O软件

第六章 输入输出系统

假脱机（SPooling）系统

- 用两道程序模拟外围控制机功能
- 在联机情况下实现的同时外围操作的技术，称为
SPooling(Simultaneous Peripheral Operating
OnLine)技术，或称为假脱机技术

78

6.6 用户层的I/O软件

第六章 输入输出系统

假脱机（SPooling）系统

- 将独占设备改造成共享设备，从而提高了设备利用率和系统效率，这种技术被称为SPooling技术
- 同时外围联机操作（Simultaneous Peripheral Operating On—Line），或称为假脱机操作
- 应用程序进行I/O操作时，只是和SPooling程序交换数据，可以称为“虚拟I/O”

79

6.6 用户层的I/O软件

第六章 输入输出系统

假脱机（SPooling）系统

三大部分

- 输入井和输出井——磁盘上开辟的两个大存储空间
- 输入井模拟脱机输入的磁盘设备，输出井模拟脱机输出时的磁盘
- 输入缓冲区和输出缓冲区——内存中开辟的两个缓冲区。输入缓冲区暂存由输入设备送来的数据，后送输入井；输出缓冲区暂存从输出井送来的数据，后送输出设备
- 输入进程和输出进程——内存中的两个进程利用两个进程模拟脱机I/O时的外围处理机

80

6.6 用户层的I/O软件

第六章 输入输出系统

假脱机（SPOOLing）系统

■ 提高了I/O的速度

利用输入输出井模拟脱机输入输出，缓和了CPU和I/O设备速度不匹配的矛盾

■ 将独占设备改造为共享设备

并没有为进程分配设备，而是为进程分配一存储区和建立一张I/O请求表

■ 实现了虚拟设备功能

多个进程同时使用一台独占设备

81

6.6 用户层的I/O软件

第六章 输入输出系统

假脱机（SPOOLing）系统

- 打印机属于独占设备
- 利用假脱机技术，将打印机改造为一台可供多个用户共享的打印设备，从而提高设备的利用率，也方便了用户
- 假脱机打印系统的构成
 - 磁盘缓冲区
 - 打印缓冲区
 - 假脱机管理进程和假脱机打印进程

82

6.6 用户层的I/O软件

第六章 输入输出系统

假脱机（SPooling）系统

- 打印机属于独占设备，利用SPooling技术可将其改造为一台可供多个用户共享的设备
- 当用户进程请求打印输出时，SPooling系统立即同意为它打印输出，但不真正把打印机分配给它而只为它做两件事：
 - 由输出进程SP₀在输出井中为之申请一个空闲的磁盘块区，并将要打印的数据送入其中
 - SP₀再为用户进程申请一张空白的用户请求打印表，并将用户的打印要求填入其中，然后将该表挂在打印机的请求打印队列中

83

6.6 用户层的I/O软件

第六章 输入输出系统

假脱机（SPooling）系统

- 打印输出过程：
 - 如果打印机空闲，输出进程将从请求打印队列的队首取出一张请求打印表，根据表中的要求将要打印的数据，从输出井传送到输出缓冲区，再由打印机进行打印
 - 打印完毕后，输出进程再查看请求打印队列中是否还有等待打印的请求表，如此下去，直至请求打印队列为空，输出进程才将自己阻塞起来
 - 仅当下次再有打印请求时，输出进程才被唤醒

84

6.6 用户层的I/O软件

第六章 输入输出系统

假脱机（SPooling）系统

假脱机打印机系统的特点

- (1) 提高了I/O的速度
- (2) 将独占设备改造为共享设备
- (3) 实现了虚拟设备功能

85

6.6 用户层的I/O软件

第六章 输入输出系统

守护进程（daemon）

- 凡是需要将**独占设备**改造为可供多个进程共享的设备时，都要为该设备配置一个**守护进程**和一个**假脱机文件队列**（目录）
- 守护进程是允许使用**该独占设备的唯一进程**，所有其他进程都不能直接使用该设备，只能将对该设备的使用要求**写入一份文件中**，放在假脱机目录中
- 由守护进程按照**目录中的文件**依次来**完成各进程对该设备的请求**，就把一台独占设备改造为可为多个进程共享的设备

86

6.6 用户层的I/O软件

第六章 输入输出系统

守护进程 (daemon)

利用守护进程实现打印机

- ① 为打印机建立一个守护进程，由它执行一部分原来由假脱机管理进程功能的功能
- ② 由请求进程生成一份要求打印的文件，并将用户请求打印文件放入假脱机文件队列（目录）中

87

6.6 用户层的I/O软件

第六章 输入输出系统

守护进程 (daemon)

- 对每个用户而言，系统只是即时将数据输出到缓冲区，并没真正被打印，只是让用户感觉系统已为他打印。
- 其真正的打印操作，是在打印机空闲且该打印任务在等待队列中已排到队首时进行的，并且，**打印操作本身也是利用CPU的一个时间片**
- 没有使用专门的外围机
- 过程用户不可见的

88

6.7 缓冲区管理

第六章 输入输出系统

- ✓ 缓冲的引入
- ✓ 单缓冲区和双缓冲区
- ✓ 环形缓冲区
- ✓ 缓冲池

89

6.7 缓冲区管理

第六章 输入输出系统

缓冲的引入

- (1) 缓和CPU与I/O设备间速度不匹配的矛盾
- (2) 减少CPU中断频率，放宽对CPU中断响应时间限制
- (3) 解决数据粒度不匹配的问题
- (4) 提高CPU和I/O设备之间的并行性

90

6.7 缓冲区管理

第六章 输入输出系统

单缓冲与双缓冲

1. 单缓冲区

每当用户进程发出一I/O请求时，
操作系统便在主存中为之分配一缓冲区。

91

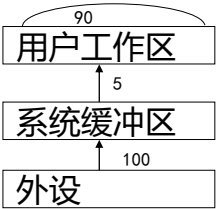
6.7 缓冲区管理

第六章 输入输出系统

单缓冲与双缓冲

1. 单缓冲区

例：设系统缓冲区和用户工作区均采用单缓冲，从外设读入1个数据块到系统缓冲区的时间为100，从系统缓冲区读入1个数据块到用户工作区的时间为5，对用户工作区中的1个数据块进行分析的时间为90，进程从外设读入并分析2个数据块的最短时间为：



92

6.7 缓冲区管理

第六章 输入输出系统

单缓冲与双缓冲

1. 单缓冲区

分析：

- 数据块1从外设到用户工作区的总时间为105，在这段时间中，数据块2没有进行操作
- 在数据块1进行分析处理时，数据块2从外设到用户工作区的总时间为105，这段时间是并行的
- 再加上数据块2进行处理的时间90，总共为300

93

6.7 缓冲区管理

第六章 输入输出系统

单缓冲与双缓冲

2. 双缓冲区

- 缓冲对换(Buffer Swapping)
- 在设备输入时
 - 先将数据送入第一缓冲区
 - 装满后便转向第二缓冲区
 - 操作系统可以从第一缓冲区中移出数据送入用户进程
 - 由CPU对数据进行计算

94

6.7 缓冲区管理

第六章 输入输出系统

单缓冲与双缓冲

双机通讯时缓冲区的设置

- 若我们在实现两台机器之间的通信时，仅为它们配置了单缓冲，那么它们之间任意时刻都只能实现单方向的数据传输，而绝不允许双方同时向对方发送数据。
- 为了实现双向数据传输，必须在两台机器中都设置两个缓冲区，一个用作发送缓冲区，另一个用作接收缓冲区。

95

6.7 缓冲区管理

第六章 输入输出系统

环形缓冲区(circular buffer)

- 双机速度相差甚远，双缓冲的效果不够理想
- 随着缓冲区数量的增加，情况有所改善
- 引入了多缓冲机制将缓冲区组织成循环缓冲

多个缓冲区，CPU和外设的处理速度可以相差较大

96

6.7 缓冲区管理

第六章 输入输出系统

环形缓冲区(circular buffer)

循环缓冲的组成

(1) 多个缓冲区

循环缓冲有多个大小相同的缓冲区，作为输入的缓冲区有三种类型：R:空缓冲；G:满缓冲；C:当前缓冲

(2) 多个指针

作为输入的缓冲区可设置三个指针：

- $next_g$ ：指示下一个应取数据的buf — G
- $next_i$ ：指示下一个空buf — R
- Current：正在使用的缓冲区C的指针

97

6.7 缓冲区管理

第六章 输入输出系统

环形缓冲区(circular buffer)

循环缓冲区的使用

$next_g$ ：指示下一个应取数据的buf

$next_i$ ：指示下一个空buf

Getbuf:

读取数据，取 $next_g$ 对应缓冲区提供使用，并改为C，

$current = next_g$; $next_g = (next_g + 1) \text{ Mod } N$

装入数据，将 $next_i$ 对应缓冲区提供使用，

$next_i = (next_i + 1) \text{ Mod } N$

Releasebuf:

若C满，装满缓存区时，释放C，则改为G；

若C空，数据提取完后，释放C，则改为R；

98

6.7 缓冲区管理

第六章 输入输出系统

环形缓冲区(circular buffer)

进程同步

- 使用输入循环缓冲，可使输入进程和计算进程并行执行
- 相应的两个指针不断顺时针方向移动：

(1) $Next_i$ 赶上 $Next_g$

意味着输入速度大于计算速度，缓冲区满，此情况称为系统受计算限制。

(2) $Next_g$ 赶上 $Next_i$

意味着输入速度低于计算速度，缓冲区空，此情况称为系统受I/O限制。

99

6.7 缓冲区管理

第六章 输入输出系统

缓冲池

缓冲池：系统提供的公用缓冲

(1) 组成：

三个队列：

空缓冲队列emq

输入队列inq

输出队列outq

■ 四个工作缓冲区：

□ hin：收容输入数据

□ sin：提取输入数据

□ hout：收容输出数据

□ sout：提取输出数据

100

6.8 磁盘存储器的性能和调度

第六章 输入输出系统

磁盘性能简述

1. 数据的组织和格式

- 磁盘包括一个或多个盘片，每片分2面
- 每面可分成若干条磁道，各磁道之间有间隙
- 每条磁道上可存储相同数目的二进制位
- 磁盘密度：每英寸之中所存储的位数
- 内层磁道的密度较外层磁道的密度大

101

6.8 磁盘存储器的性能和调度

第六章 输入输出系统

磁盘性能简述

2. 磁盘的类型

固定头磁盘

- 每条磁道上都有一读/写磁头，所有磁头被装在一刚性磁臂
- 通过这些磁头可访问所有各磁道，并进行并行读/写
- 主要用于大容量磁盘上

移动头磁盘

- 每一个盘面仅配有一个磁头，也被装入磁臂中
- 为访问该盘面上的所有磁道，磁头必须能移动以进行寻道
- 移动磁头仅能以串行方式读/写，I/O速度较慢
- 结构简单，广泛应用于中小型磁盘设备中

102

6.8 磁盘存储器的性能和调度

第六章 输入输出系统

磁盘性能简述

磁盘访问时间

1) 寻道时间 T_s

—— 指把磁臂(磁头)移动到指定磁道上所经历的时间

2) 旋转延迟时间 T_r

——指定扇区移动到磁头下面所经历的时间

3) 传输时间 T_t

——把数据从磁盘读出或向磁盘写入数据所经历的时间

103

6.8 磁盘存储器的性能和调度

第六章 输入输出系统

磁盘调度算法

1. 寻道时间 T_s : $T_s = m \cdot n + S$;

2. 旋转延迟时间 T_r : $T_r = 1/2r$

3. 数据传输时间 T_t : $T_t = b/rN$

访问时间: $T_a = T_s + 1/2r + b/rN$

在访问磁盘的时间中，主要是寻道时间，因此，磁盘调度的目标就是使磁盘的平均寻道时间最少。

104

6.8 磁盘存储器的性能和调度
磁盘调度算法

1. 先来先服务FCFS
(First-Come, First Served)

- ❖ 根据进程请求访问磁盘的先后次序进行调度
- ❖ 优点：简单、公平，不会出现请求长期得不到满足
- ❖ 缺点：未优化，平均寻道时间长

55、58、39、18、90、
160、150、38、184

100道开始	
被访问的下一个磁道	移动距离
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	

105

6.8 磁盘存储器的性能和调度
磁盘调度算法

2. 最短寻道时间优先SSTF
(Shortest Seek Time First)

- ❖ 要求访问的磁道与当前磁头所在的磁道距离最近
- ❖ 优点：使每次寻道时间最短
- ❖ 缺点：不能保证平均寻道时间最短；可能导致距离远的进程总也得不到服务

55、58、39、18、90、
160、150、38、184

100道开始	
被访问的下一个磁道	移动距离
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	

106

6.8 磁盘存储器的性能和调度
磁盘调度算法

3. 扫描(SCAN) 算法

进程“饥饿”现象

- 只要不断有新进程的请求到达
- 且其所要访问的磁道与磁头当前所在磁道的距离较近
- 这种新进程的I/O请求必须优先满足

6.8 磁盘存储器的性能和调度
磁盘调度算法

□ 扫描(SCAN) 算法

- ❖ 对SSTF算法略加修改后所形成的SCAN算法，即可防止进程出现“饥饿”现象
- ❖ SCAN算法不仅考虑欲访问的磁道与当前磁道的距离，更优先考虑的是磁头当前的移动方向
- ❖ 磁头移动：自里向外↔自外向里
- ❖ 又称为“电梯调度算法”

55、58、39、18、90、
160、150、38、184

100道开始，增加方向	
被访问的下一个磁道	移动距离
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	

6.8 磁盘存储器的性能和调度
磁盘调度算法

4. 循环扫描 (CSCAN) 算法

- ❖ 规定磁头单向移动
- ❖ 减少刚移过的磁道的等待时间

55、58、39、18、90、
160、150、38、184

100道开始，增加方向	
被访问的下一个磁道	移动距离
150	50
160	10
184	24
18	166
38	20
39	1
55	16
58	3
90	32
平均寻道长度：27.5	

109

6.8 磁盘存储器的性能和调度
磁盘调度算法

5. N-Step-SCAN和FSCAN调度算法

- ❖ FSCAN算法
 - FSCAN算法是N步SCAN算法的简化
 - 只将磁盘请求队列分成两个子队列
 - 一是由当前所有请求I/O的进程形成的队列，由磁盘调度按SCAN算法进行处理。
 - 在扫描期间，新出现的所有请求I/O的进程， 则放入另一个等待处理的请求队列

110