

操作系统原理

第八章 设备管理

授课人：郑然



- 设备管理概述
- 缓冲技术
- 设备分配
- I/O控制

设备管理概述

1. 设备分类

(1) 存储设备

存储设备又称块设备，是存储信息的设备，如：磁盘、磁鼓（以块为单位传输信息）。

(2) 输入输出设备

输入输出设备又称字符设备，能将信息从计算机外部输入到机内，或反之，如：键盘、显示器、打印机（以字符为单位传输信息）。

(3) 通信设备

通信设备负责计算机之间的信息传输，如调制解调器、网卡等。

计算机外部设备的特征：

- (1) 速度差异大
- (2) 传输单位不同
- (3) 容许的操作种类不同
- (4) 出错条件不同

2. 设备管理的目标

(1) 提高设备利用率

- ① 合理分配设备
- ② 提高设备与CPU、各外部设备之间的并行性

(2) 方便用户的使用

提供使用方便且独立于设备的界面

- ① 统一：对各种不同的设备提供一致的界面
- ② 独立于设备：用户使用的设备与物理设备无关

3. 设备管理功能

(1) 状态跟踪

动态地记录各种设备的状态。

(2) 设备分配与回收

① 静态分配 —— 应用程序级

程序进入系统时进行分配，退出系统时收回全部资源。

② 动态分配 —— 进程级

进程提出设备申请时进行分配，使用完毕后立即收回。

(3) 设备控制

实施设备驱动和中断处理的工作。

4. 设备独立性

(1) 设备独立性概念

① 什么是设备独立性

所谓设备独立性是指，用户在程序中使用的设备与实际使用的设备无关，也就是在用户程序中仅使用逻辑设备名。

② 逻辑设备名

逻辑设备名，是用户自己指定的设备名（或设备号），它是暂时的、可更改的。

③ 物理设备名

物理设备名，是系统提供的设备的标准名称，它是永久的、不可更改的。

(2) 两种类型的设备独立性

① 一个程序独立于分配给它的某种类型的具体设备

系统可以根据设备的使用情况，动态地分配给程序某类设备中的任一物理设备，程序都能正确地执行。

② 程序应尽可能与它所使用的I/O设备类型无关

在输入(或输出)信息时，信息可以从不同类型的输入(或输出)设备上输入(或输出)，若要改变输入(或输出)设备的类型，程序只需进行最少的修改。

(3) 设备独立性的实现

① 在高级语言中用软通道实现

使用高级语言提供的指派语句，通过指派一个逻辑设备名(通道号)来定义一个设备或文件。

如：fd = open(“/dev/lp”, mode)

② 在批处理系统中，用联接说明语句来定义

如：OUTPUT1 = LPT

③ 在交互系统中，用指派命令来定义

如：PDP系列机上的RT11系统

ASSIGN 设备物理名 设备逻辑名

(4) 设备独立性的优点

- ◆ 方便用户
- ◆ 改善设备利用率
- ◆ 提高系统的可扩展性和可适应性

5. 设备控制块

(1) 什么是设备控制块 (DCB)

系统为每一台设备都配置了一个用来记录设备的硬件特性、连接和使用情况的一组数据，称为设备控制块。

(2) 设备控制块的内容

设备名
设备属性
指向命令转换表的指针
在I/O总线上的设备地址
设备状态
当前用户进程指针
I/O请求队列指针

设备控制块dcb

① 设备名

设备的系统名，即设备的物理名。

② 设备属性

描述设备现行状态的一组属性。

③ 命令转换表

转换表包含设备特定的I/O例程地址，不具备相应功能的设备在其例程地址上可以填“-1”。

缓存技术

1. 缓冲概念

(1) 什么是缓冲

缓冲是两种不同速度的设备之间传输信息时平滑传输过程的常用手段。

(2) 缓冲类别

① 缓冲器

缓冲器是用来暂时存放数据的一种存储装置，它容量较小，存取速度快。

② 软件缓冲

在I/O操作期间用来临时存放I/O数据的一块存储区域。

(3) 为什么要引入缓冲

① 处理数据流的生产者与消费者间的速度差异

如：从调制解调器收到一个文件，并保存到硬盘上。

② 协调传输数据大小不一致的设备

如：在计算机网络中用来处理消息的分段和重组。

③ 应用程序的拷贝语义

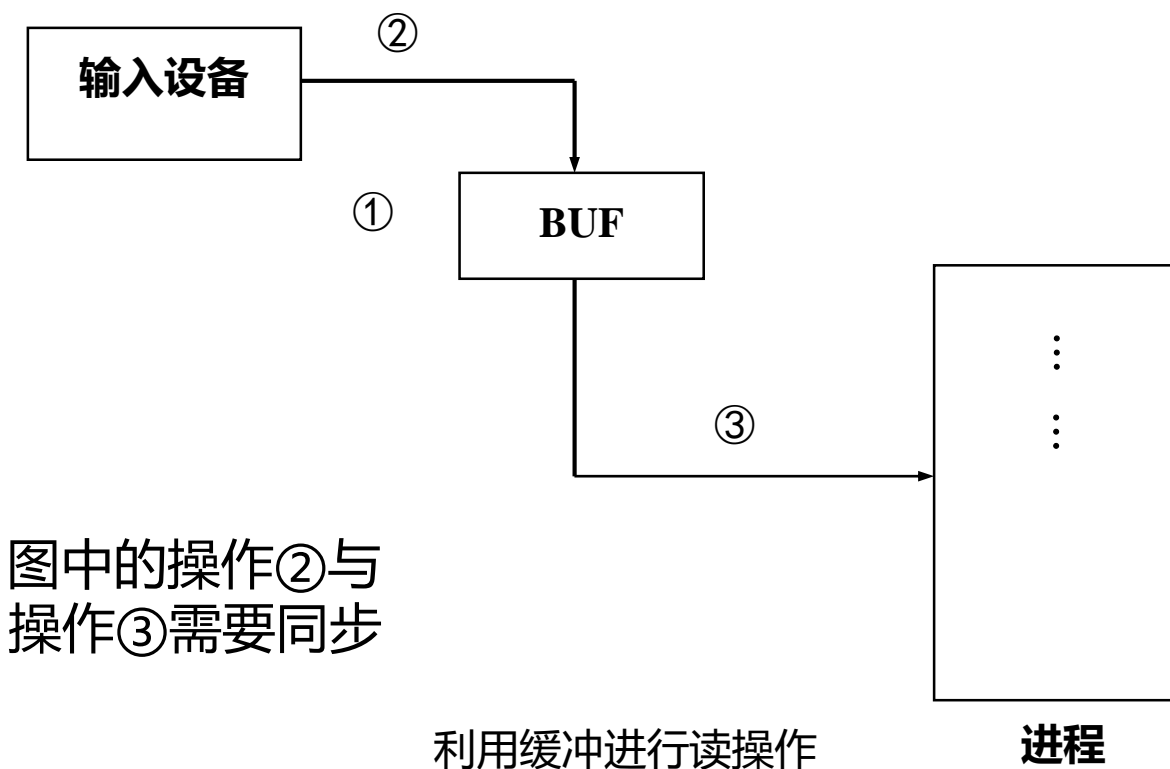
如：操作系统为保证系统调用write的正确语义（应用程序要写入磁盘的数据就是write系统调用发生时的版本）。

方法：在系统调用返回前将应用程序缓冲区复制到内核缓冲区。

2. 利用缓冲技术如何进行I/O操作

(1) 进程活动期间，请求从某字符设备读入数据

① 进程请求从输入设备进行读操作的图示



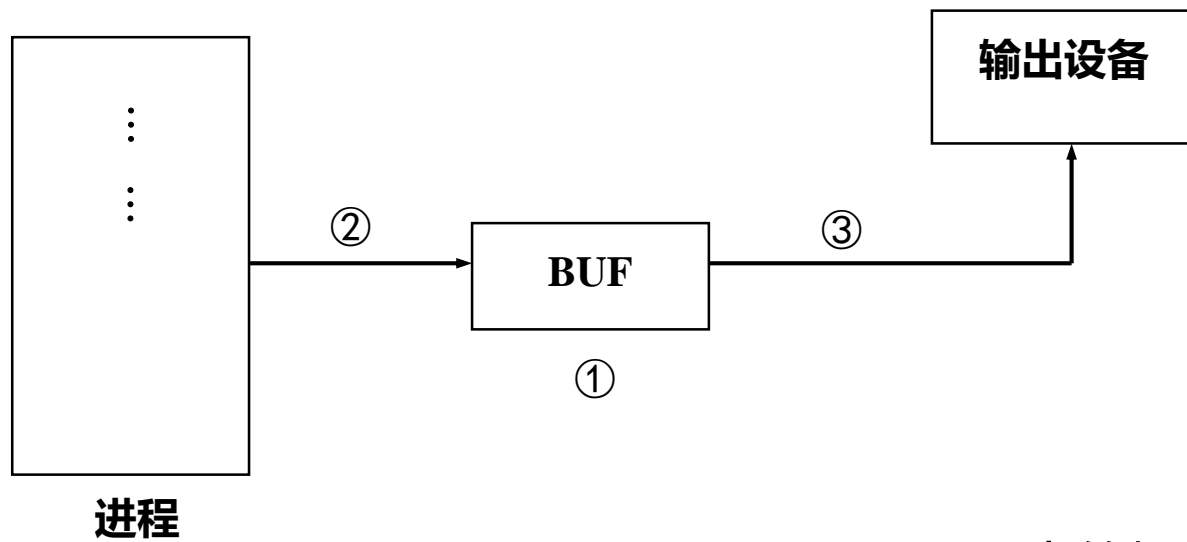
② 进程请求从输入设备进行读操作的步骤

- i 当用户要求在某个设备上读操作时，首先从系统中获得一个空的缓冲区 (图中标注的操作①)；
- ii 将一个物理记录送到缓冲区中 (图中标注的存在②)；
- iii 当用户请求这些数据时，系统将依据逻辑记录特性从缓冲区中提取并发送到用户进程存储区中 (图中标注的操作③)；
- iv 当缓冲区空而进程又要从中取用数据时该进程被迫等待。
此时，操作系统需要重新送数据填满缓冲区，进程才能从中取数据继续运行。

要注意操作②与操作③的同步关系

(2) 进程活动期间，请求从输出设备输出数据

① 进程请求从输出设备进行写操作的图示



图中的操作②与
操作③需要同步

利用缓冲进行写操作

② 进程请求从输出设备进行写操作的步骤

- i 当用户要求进行写操作时，首先从系统中获得一个空的缓冲区(图中标注的操作①)；
- ii 将一个逻辑记录从进程存储区传送到缓冲区中(图中标注操作②)；
- iii 当缓冲区写满时，系统将缓冲区的内容作为物理记录文件写到设备上，使缓冲区再次为空(图中标注的操作③)；
- iv 只有在系统还来不及腾空缓冲区之前，进程又企图输出信息时，它才需要等待。

要注意操作②与操作③的同步关系

3. 常用的缓冲技术

双缓冲、环形缓冲、缓冲池

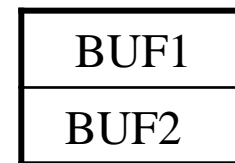
(1) 双缓冲

在双缓冲方案下，为输入或输出分配两个缓冲区 buf_1 、 buf_2 。

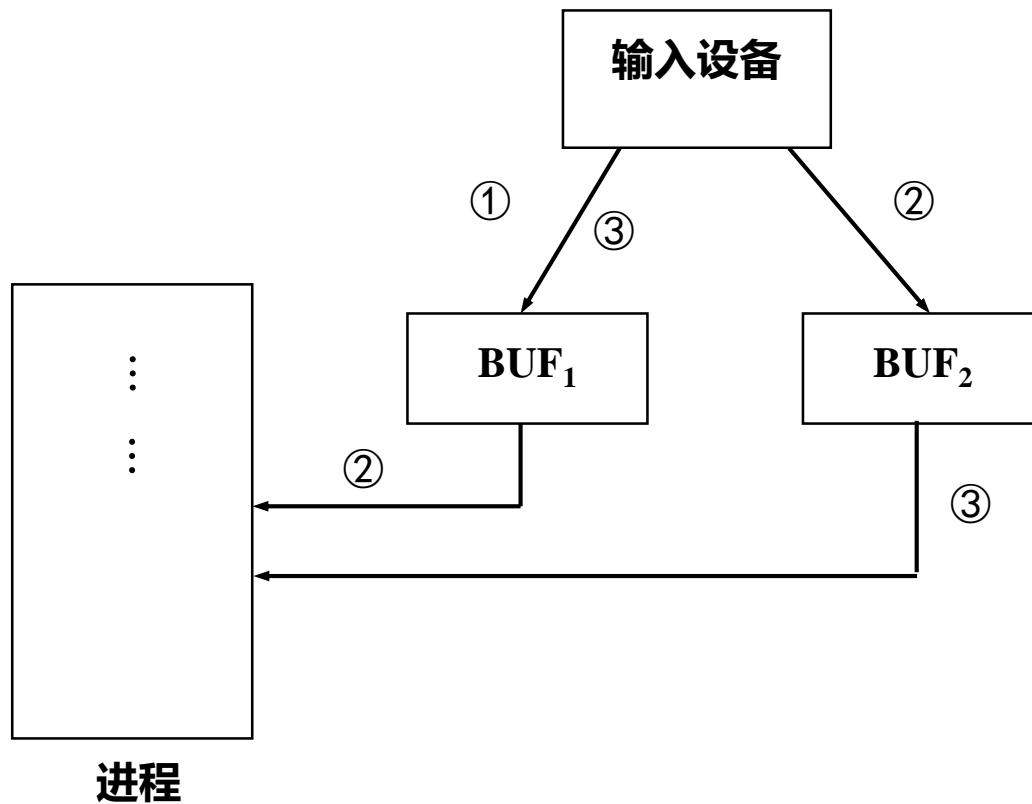
◆ 输入数据时，如何利用双缓冲

◆ 输出数据时，如何利用双缓冲

◆ 缓冲区既用于输入，也用于输出数据时，如何利用双缓冲

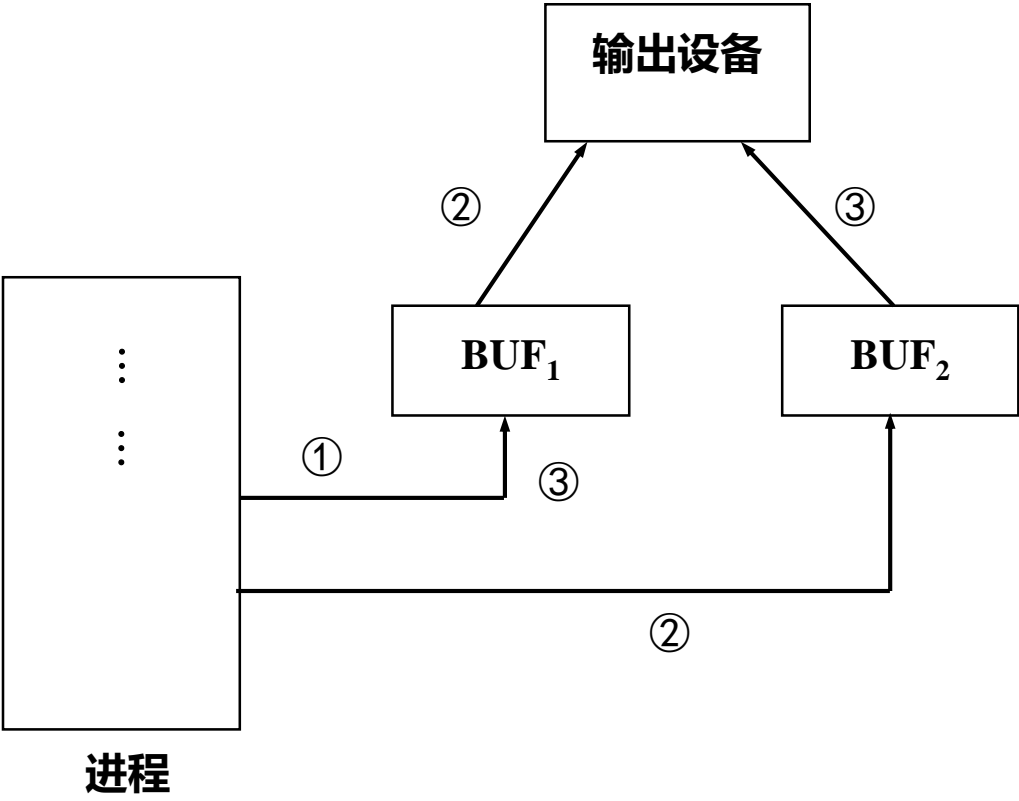


(2) 利用双缓冲输入数据



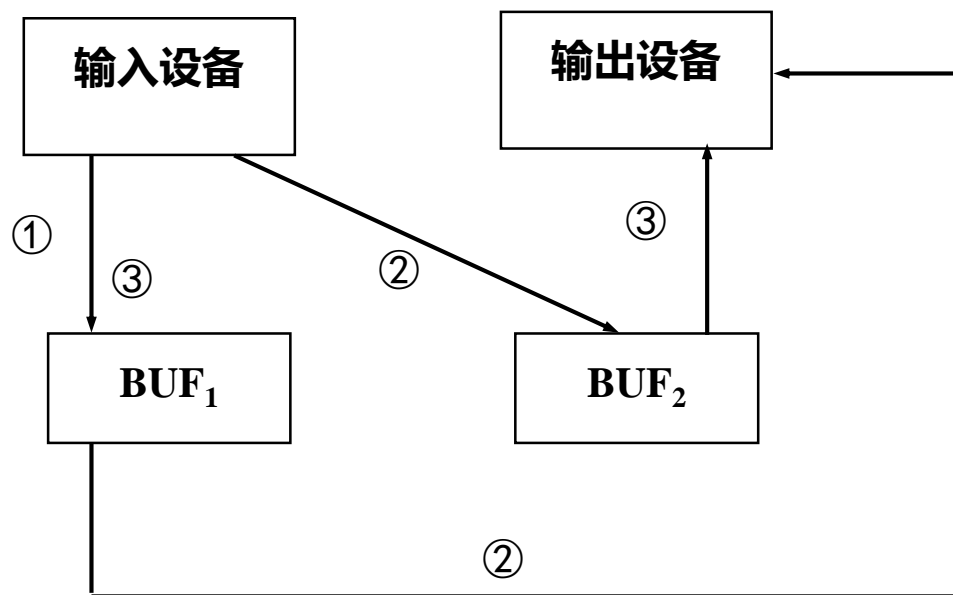
双缓冲读入数据的操作

(3) 利用双缓冲输出数据



双缓冲输出数据的操作

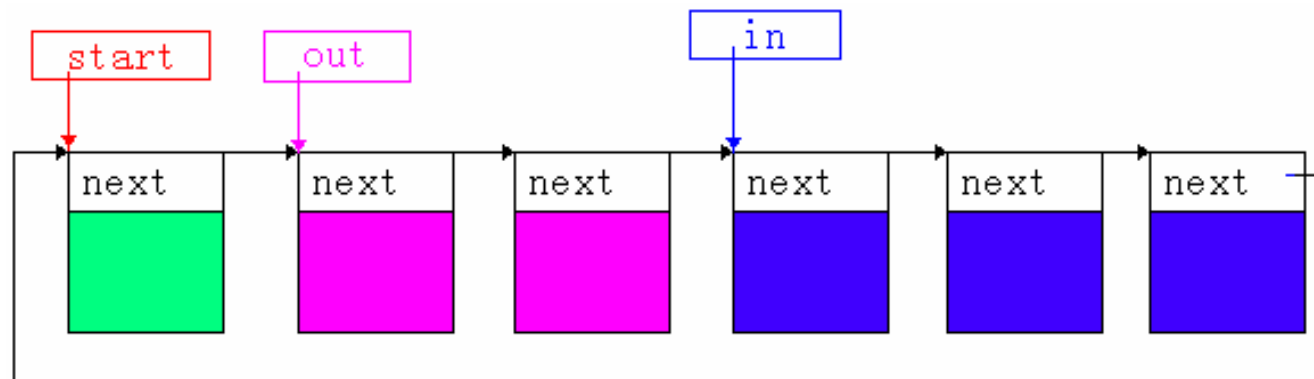
(4) 双缓冲同时用于输入/输出数据的操作



双缓冲同时用于输入/输出数据的操作

4. 环形缓冲

在系统中设置若干个缓冲区，并把这些缓冲区链接起来，这样若干个缓冲区就形成了一个环，故称环形缓冲区。



5. 缓冲池

系统设置多个缓冲区，形成一个缓冲池。这个池中的缓冲区为系统中所有的进程共享使用。

6. UNIX系统的缓冲区管理

(1) UNIX系统缓冲管理的目的

- ◆ 加快系统响应、增强系统吞吐量
- ◆ 减少对磁盘的I/O操作次数

(2) UNIX系统缓冲管理的思路

- ◆ 当进程要从磁盘读数据时，首先考虑从高速缓冲中读

预先缓存

- ◆ 当进程要写数据到磁盘时，先写入高速缓冲中

延迟发送

(3) 缓冲管理数据结构

① 缓冲区的组成

- i 缓存数组 —— 含有磁盘上的数据的存储器数组
- ii 缓存首部 —— 描述缓冲区特性的数据结构

② 缓存首部结构

缓存首部结构

设备号	dev
块号	blkno
状态	flag
指向数据区域的指针	
传送字节数	
返回的I/O出错信息	
b_forw	设备缓冲区队列前向指针
b_back	设备缓冲区队列后向指针
av_forw	空闲缓冲区队列前向指针
av_back	空闲缓冲区队列后向指针

缓冲首部结构

i 设备号dev

- ◆ 缓冲区所包含的信息所属设备的设备号

ii 块号blkno

- ◆ 由设备号指出的设备上相对于第0块的块号

iii 状态flag——描述了缓冲区当前的状态

- ◆ 忙标志BUSY：缓冲区当前正“忙”
- ◆ 有效位AVE：缓冲包含的数据有效
- ◆ 延迟写DELWR：核心在某缓冲区重新分配出去之前必须把缓冲区内容写到磁盘上
- ◆ 写标志WRITE：核心当前正把缓冲区的内容写到磁盘
- ◆ 读标志READ：核心当前正从磁盘往缓冲区写信息
- ◆ 等待位 WAIT：一个进程当前正在等候缓冲区变为空闲

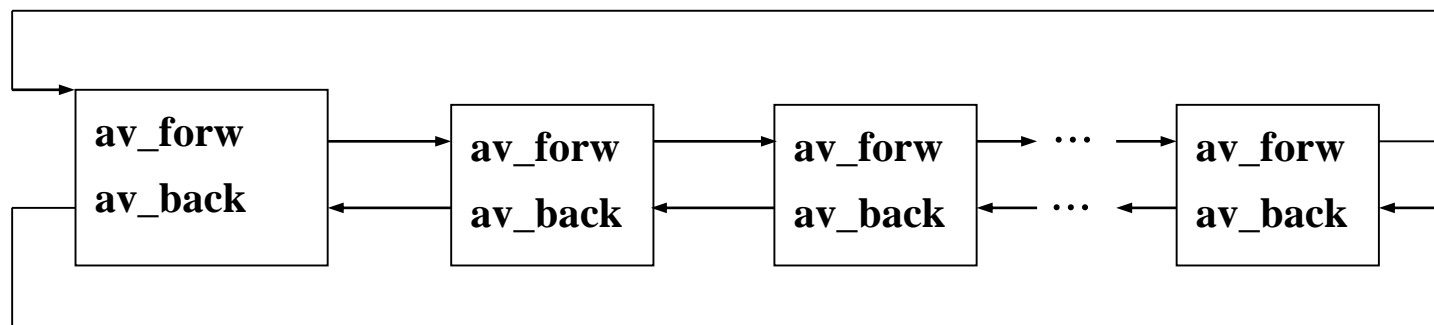
③ 缓冲区队列结构

i 设备缓冲区队列

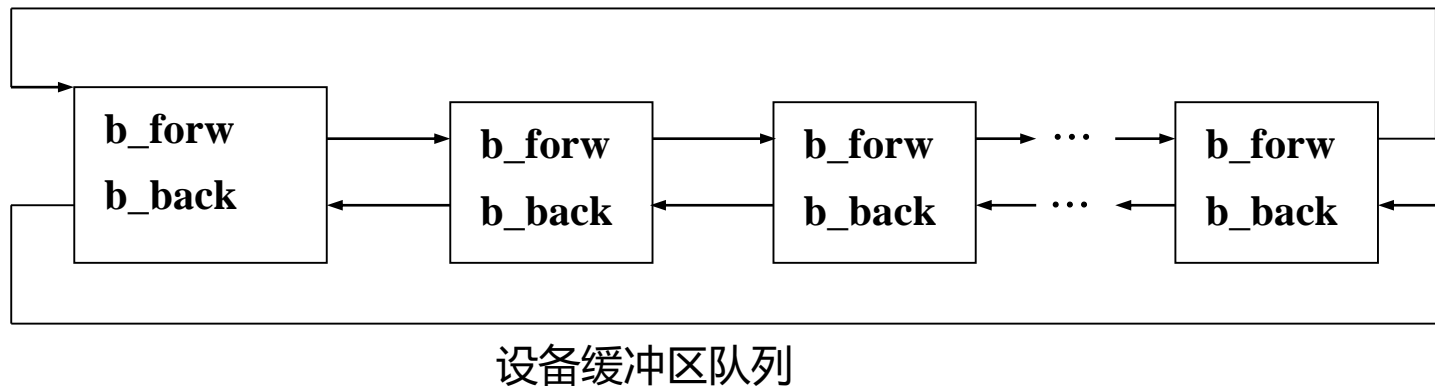
与某类设备有关的所有缓冲区组成的队列称为设备缓冲区队列，简称b链。

ii 空闲缓冲区队列

可供重新分配使用的缓冲区组成的队列称为空闲缓冲区队列，简称av链。



空闲缓冲区队列



④ 缓冲区队列指针

i b 链指针

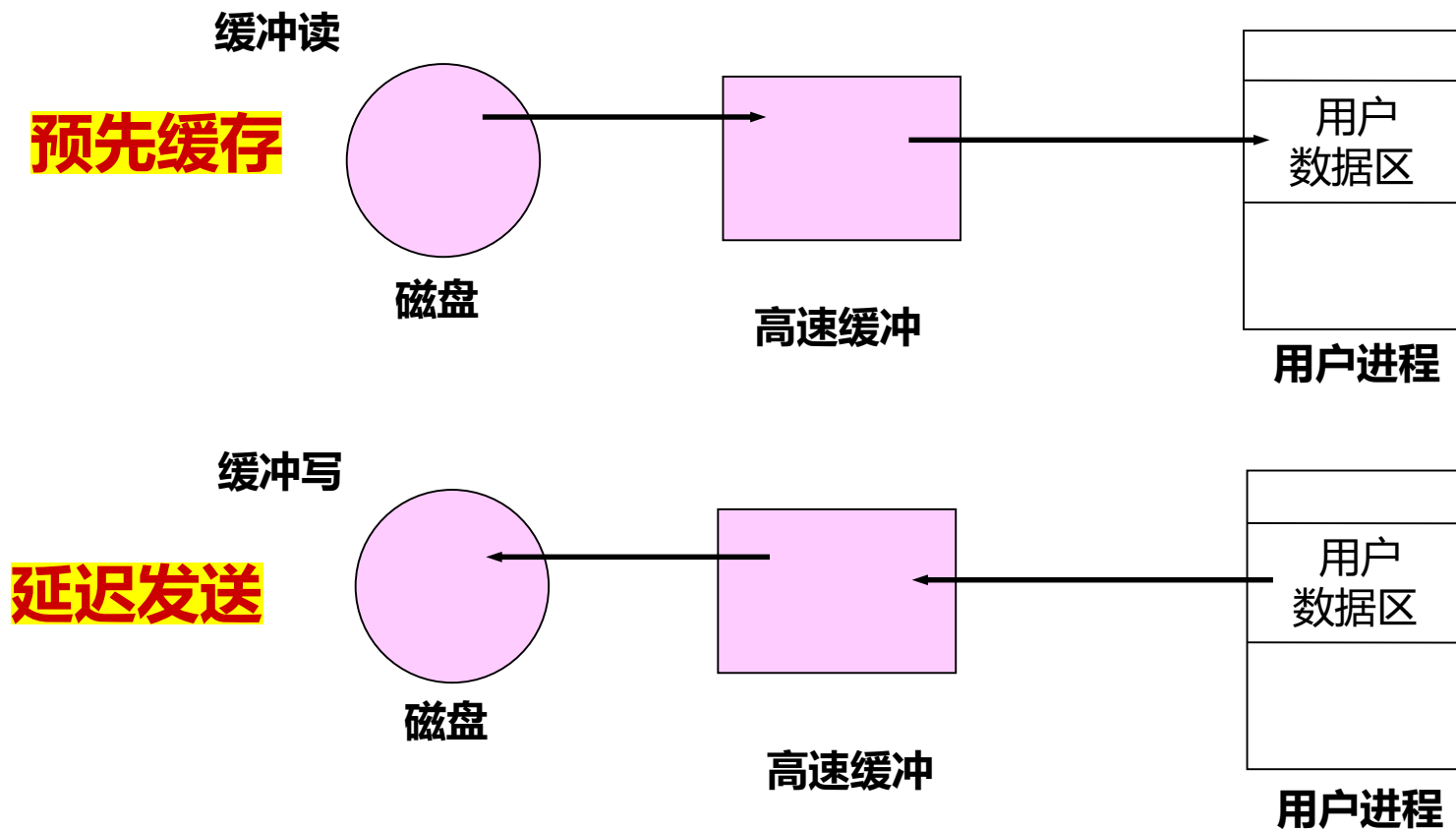
- ◆ **b_forw** : 指向设备缓冲区队列上的下一个缓冲区的指针
- ◆ **b_back** : 指向设备缓冲区队列上的上一个缓冲区的指针

ii av 链指针

- ◆ **av_forw** : 指向空闲缓冲区队列上的下一个缓冲区的指针
- ◆ **av_back** : 指向空闲缓冲区队列上的上一个缓冲区的指针

(4) UNIX缓冲管理算法

① 缓冲读、写示意图



缓冲读、写示意图

② UNIX缓冲管理算法

i 一个buf被分配用于读/写某设备上的块时

置B_BUSY=1, 位于b链上, 不在av链上;

ii 当读/写操作结束时

释放该buf, 置B_BUSY=0, 仍留在b链上, 并送入av链尾;

iii 若进程需要的信息在buf中时

在该设备的b链上找到, 置B_BUSY=1; 从av链上摘除, 使用完后, 又送入av链, 链入队尾。

iv 对空闲buf空队列的处理

当需要一个空闲buf时，总是取空闲buf队列(av链)的**首元素**；
一个使用过的buf释放时，插入到空闲buf队列(av链)的**队尾**。

实现了精确的最久未使用淘汰算法 (LRU算法)

v 对延迟写的处理

当一个具有延迟写标记的buf移到av链头，要用于分配时，
立即进行写操作。从av链上摘除，使用完后又送入av头部。

设备分配

■ 分配原则

• 静态分配

当一个作业运行时，系统如果能满足作业要求的设备，则将其要求的设备全部分配给它，然后开始运行，运行完成释放其占用的所有设备。

优点：系统绝不会出现死锁，缺点是设备利用率太低。

• 动态分配

在作业（或进程）运行的过程中，需要使用设备时，就向系统申请，系统根据某种分配原则进行分配。

优点：设备的利用率高，缺点是系统有出现死锁的可能。

设备分配算法

先来先服务算法

优先级高者优先

设备分配的安全性

- 对于共享设备，不论采用静态分配还是采用动态分配都不会出现死锁
- 对于独占设备，采用动态分配有可能造成死锁

常用的设备分配技术：独享分配、共享分配和虚拟分配

1. 独享分配

(1) 独享设备

① 让一个作业在整个运行期间独占使用的设备

② 特点

- i 临界资源
- ii 费时的I/O操作或需人工干预

(2) 独享分配

在一个作业执行前，将它所要使用的设备分配给它；当它结束撤离时，将分配给它的这类设备收回。

2. 共享分配

(1) 共享设备

① 由多个作业、进程共同使用的设备称为共享设备。

② 特点

- i 旋转设备，可直接或随机访问
- ii 便于共享，转接简单，耗费较少

(2) 共享分配

在一个作业执行并发出设备申请时，将它所要使用的设备分配给它；当作业释放设备或结束撤离时，将分配给它的设备收回。

考虑调度性能

3. 虚拟分配

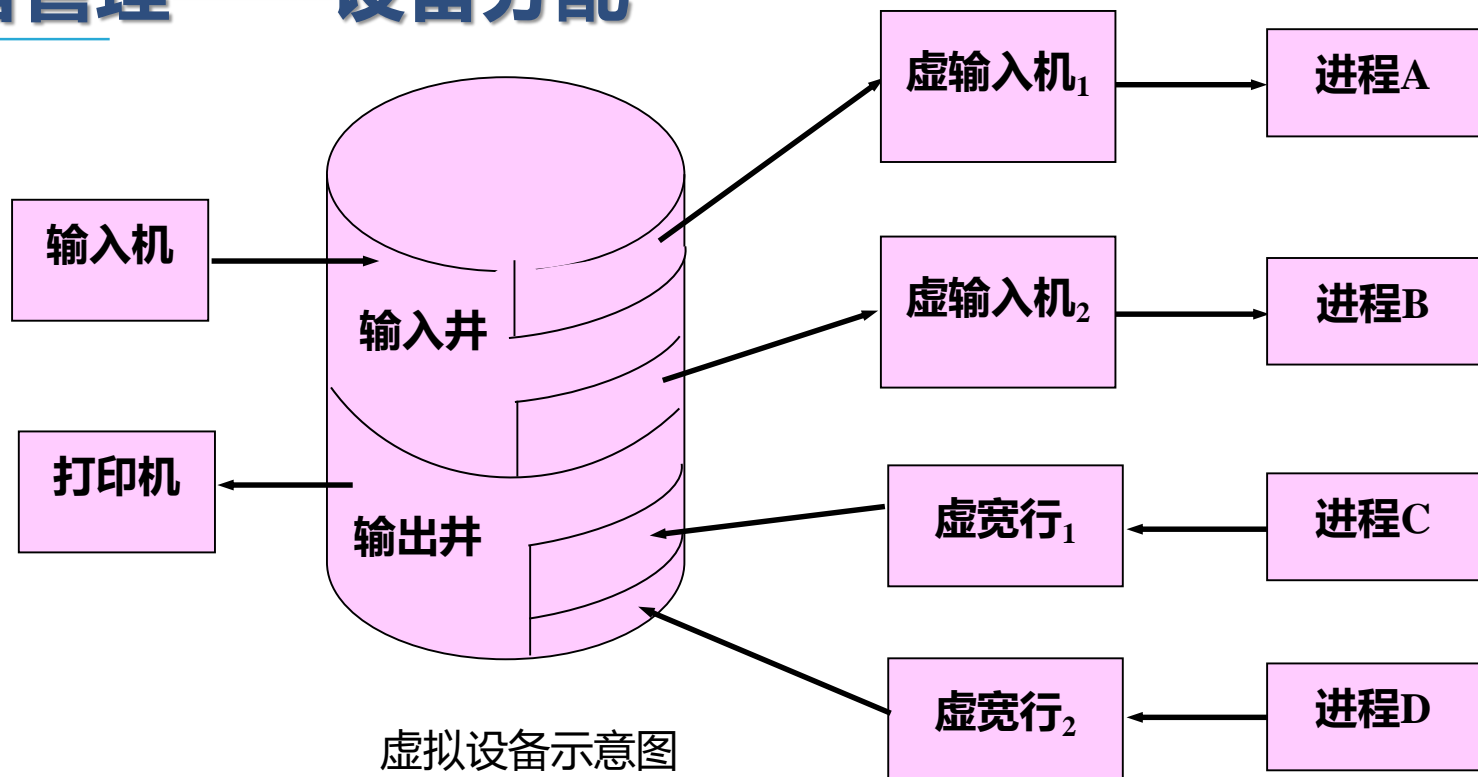
有限的独占设备，低速的独占设备，严重影响整个计算机系统效率
——在高速共享设备上能否模拟低速设备的功能

(1) 虚拟技术

所谓虚拟技术，是在一类物理设备上模拟另一类物理设备的技术，是将独占设备转化为共享设备的技术。

(2) 虚拟设备

通常把用来代替独占型设备的那部分外存空间 (包括有关的控制表格)称为虚拟设备。



(3) 虚拟分配

当进程需要与独占型设备交换信息时，系统将分配磁盘空间，并建立相应的数据结构，这种分配方法称为设备的虚拟分配。

4. SPOOLING系统

SPOOLING系统提供外围设备同时联机操作的功能。

(1) 设计思想

① 预输入

在应用程序需要数据前，OS已将所需数据预先输入到辅存输入井存放。当应用程序(或进程)需要数据时，可直接从辅存中读入主存。

② 缓输出

在应用程序执行时，将输出数据写入辅存输出井中。当应用程序(或进程)执行完毕(或需要数据时)，由操作系统将数据输出。

(2) 什么是SPOOLING系统

利用通道和中断技术，在主机控制之下，由通道完成输入输出工作。系统提供一个软件系统（包括预输入程序、缓输出程序、井管理程序、预输入表、缓输出表）。它提供输入收存和输出发送的功能，使外部设备可以并行操作。这一软件系统称为SPOOLING系统。

(3) SPOOLING系统的优点

- ① 提供虚拟设备
- ② 外围设备同时联机操作
- ③ 加快作业处理速度

(4) 实现SPOOLING系统的基础

① 大容量的辅存空间

在辅存上需开辟两个较大的输入井和输出井，用以存放大量应用程序的输入信息和输出信息。

② 硬件基础

通道装置、中断系统

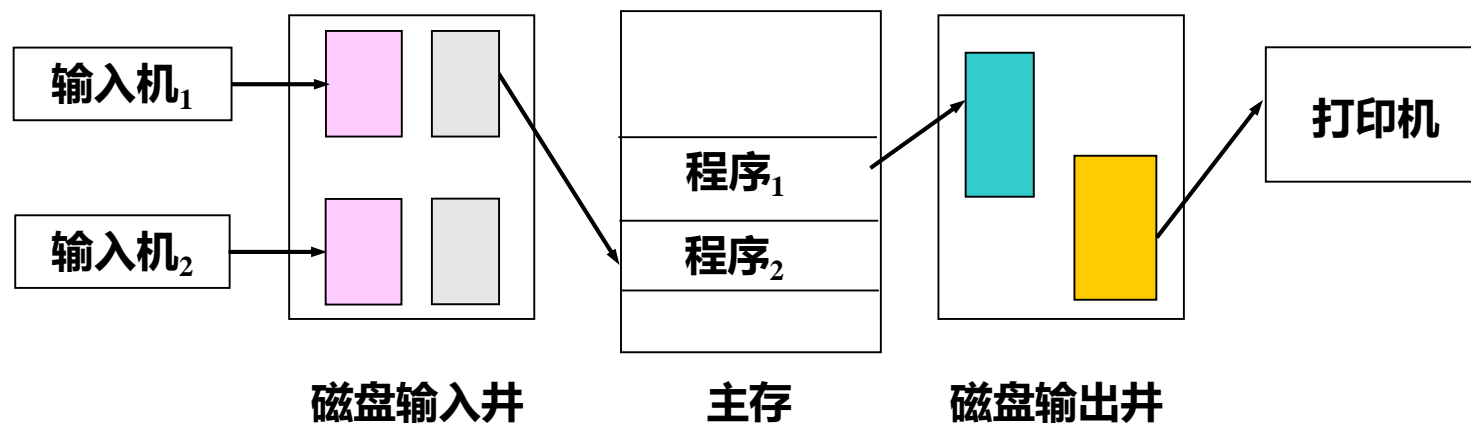
③ 数据结构

预输入表、缓输出表：描述辅存输入井和输出井的状态变化。

如：输入信息从哪台设备输入，存放在辅存输入井什么位置；输出信息存放在辅存输出井什么位置，从哪台输出设备输出。

④ 所需的软件程序

- i **预输入程序** 控制信息从独占设备输入到辅存
- ii **缓输出程序** 控制信息从辅存输出到独占设备
- iii **井管理程序** 控制用户程序和辅存之间的信息交换



SPOOLING系统工作示意图

设备控制

1. 输入/输出控制方式

CPU通过I/O控制器与物理设备打交道。按照I/O控制器智能化程度的高低，将I/O设备的控制方式分为四类。

- ◆ 循环测试I/O方式

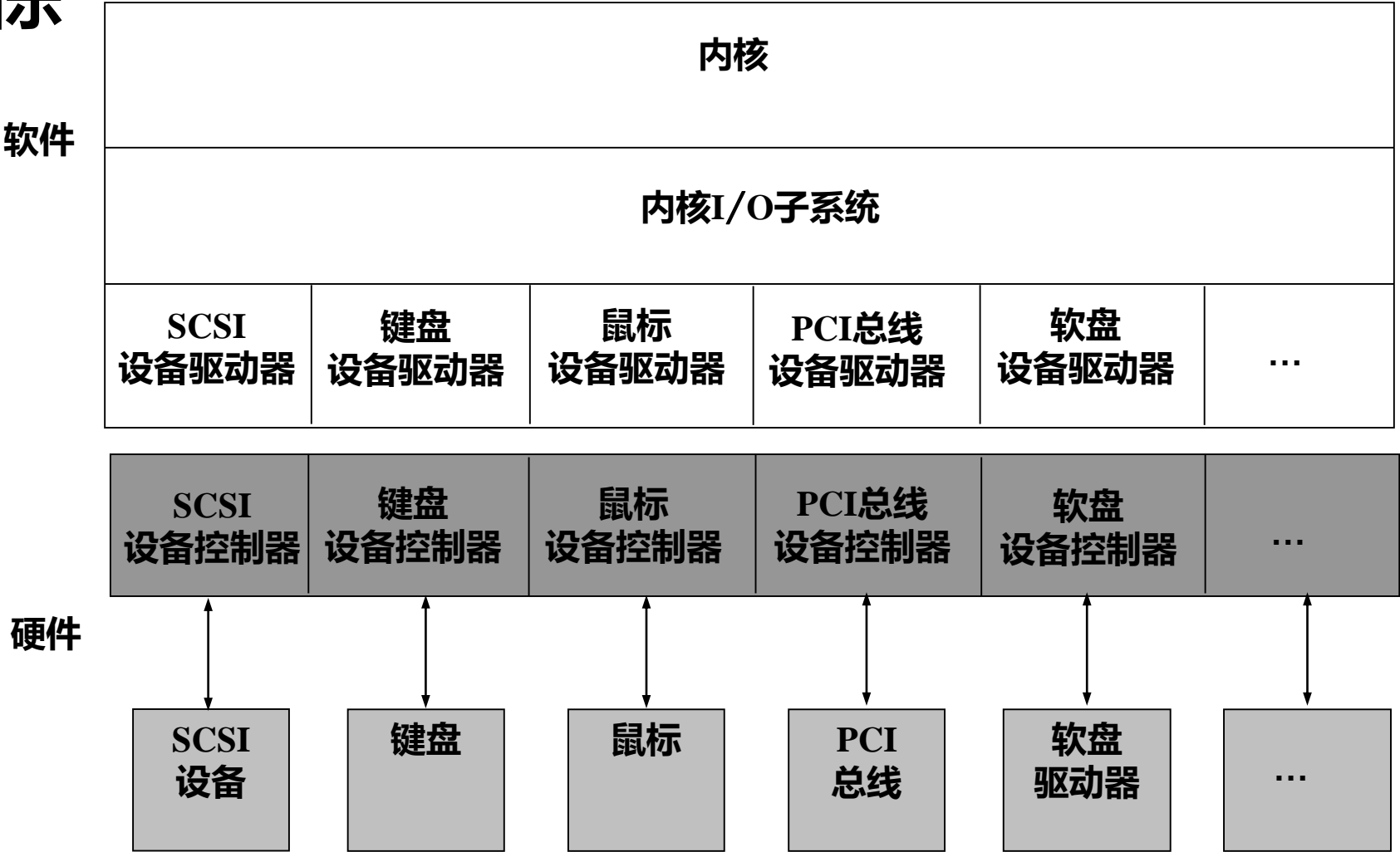
- ◆ I/O中断方式

- ◆ DMA方式

- ◆ 通道方式

2. I/O子系统

(1) 图示



一个典型的PC总线结构

(2) I/O子系统的特点

① 在应用层为用户提供 I/O应用接口

对设备的控制和操作则由内核I/O子系统来实施。

② 每个通用设备类型都通过一组标准函数 (及接口)来访问

具体的差别被I/O子系统内的内核模块(称为设备驱动程序)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的接口。

设备驱动程序层的作用是为内核I/O子系统隐藏设备控制器之间的差异。将I/O子系统与硬件分离, 简化了操作系统开发人员的任务, 也有利于设备的设计与制造。

(3) I/O控制的功能

- ① 解释用户的I/O系统调用
- ② 设备驱动
- ③ 中断处理

(4) 设备处理程序

设备处理程序是能直接控制设备运转的程序，它根据各类设备的特点和性能来编写。每一类设备有一个相应的设备处理程序，能控制同类中多台物理设备同时工作。

(5) 控制I/O核心模块的方式

① 以设备处理进程的方式

- i 为每一类设备设置一个设备处理进程 (对应的程序就是设备处理程序);
- ii 当有I/O请求来到时该进程被唤醒, 进行设备驱动工作; 当没有I/O请求时, 该进程睡眠。

由**I/O控制模块的接口程序**负责解释用户的I/O系统调用, 将其转换成I/O控制模块认识的命令形式后, 将I/O请求发给对应的设备处理进程。

② 将设备与文件一样对待

将设备与文件一样对待, 使用文件系统的系统调用命令进行设备的读、写。

3. 输入/输出控制的例

(1) 用户进程请求I/O的系统功能调用

系统功能调用的形式为：

```
doio(ldev,mode,amount,addr);
```

ldev: 逻辑设备名

mode: 操作模式

amount: 传输数据的数目

addr: 传送地址

(2) I/O接口程序 (I/O过程)

① 将逻辑设备转换为物理设备

- i 获得 I/O系统调用中给出的逻辑设备名 (ldev);
- ii 根据逻辑设备描述器, 将逻辑设备名转换为物理设备名。

② 合法性检查

- i 获得 I/O系统调用中给出的操作模式mode;
- ii 根据DCB中命令转换表中允许的操作, 检查操作的合法性。

③ 形成I/O请求块, 发消息给对应的设备处理进程

- i 根据请求的参数形成I/O请求块 (IORB);
- ii 将I/O请求块 (IORB)挂到对应的设备请求队列。

(3) I/O接口程序的描述

算法 doio

输入：设备的逻辑名 ldev

操作类型 mode

传送数据数目 amount

传数数据地址 addr

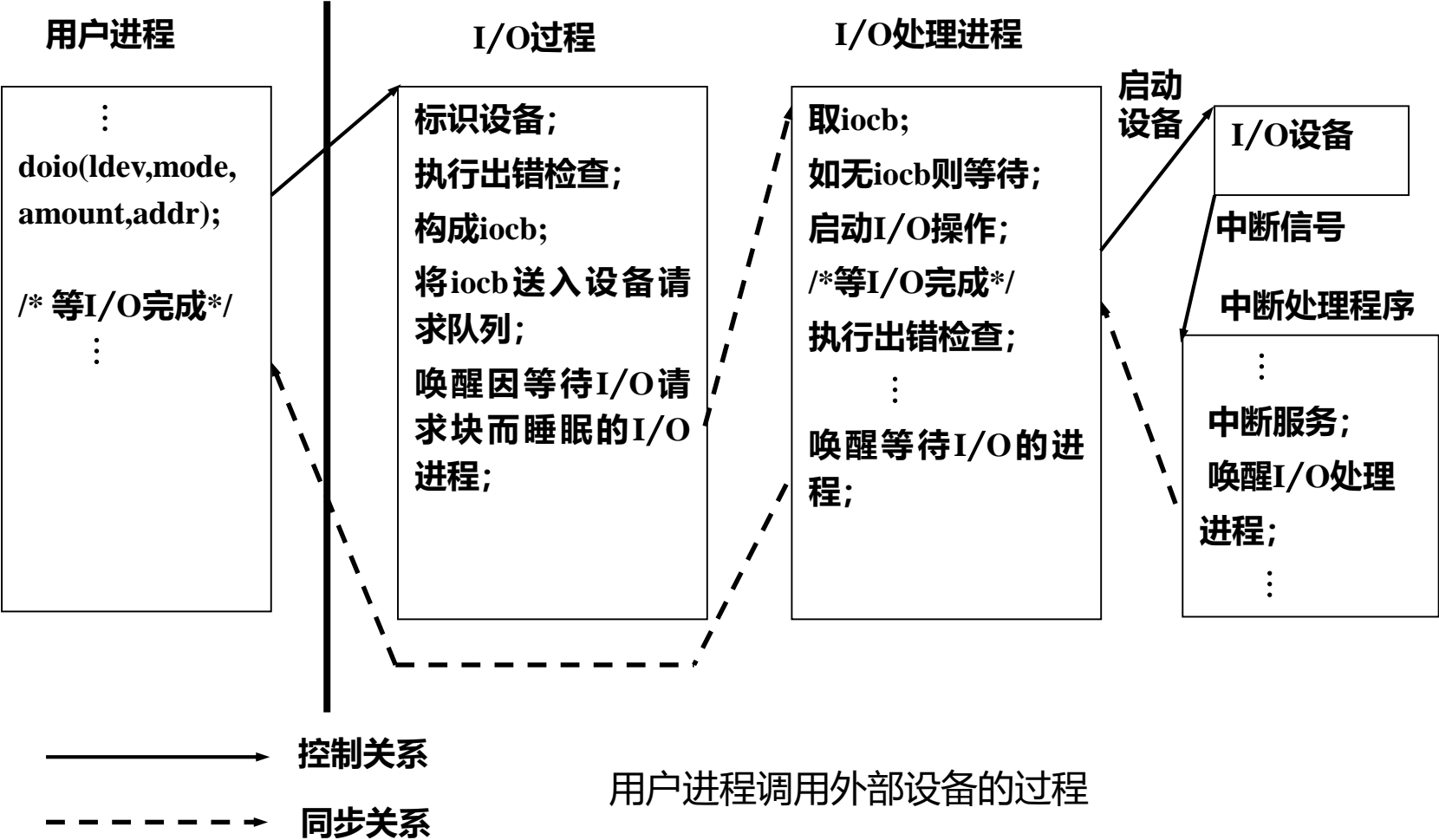
输出：如果传送出错，则带错误码
返回，否则正确返回

```
{ while (该进程的逻辑设备描述器队列不空)
{
    if (与ldev相联结的物理设备找到)
        break;    /* 找到 */
}
if (该进程的逻辑设备描述器队列为空)
    return(错误码);    /* 设备逻辑名错 */
检查参数与该设备特性是否一致;
if (不一致)
    return (错误码);    /* 传送参数错 */
构造iorb;
把iorb插入到该设备的请求队列中;
唤醒因等待I/O请求块而睡眠的进程;
}
```

(4) 设备处理进程

```
process io
{ 1: while (设备请求队列不空)
    {  取一个iorb;
      提取请求的详细信息;
      启动I/O操作;
      sleep (事件: I/O完成) /* I/O操作 */
      /* 等I/O完成后, 进入中断处理程序, 并在那里唤醒设备处理进程 */
      if (出错) 将错误信息写在该设备的dcb中;
      传送数据到目的地;
      唤醒请求此I/O操作的进程;
      删除iorb;
    }
    sleep (事件: 因无I/O请求);
    goto 1;
}
```

4. 请求I/O的进程、I/O过程、设备处理进程、中断处理程序之间的同步关系



第8章 设备管理

小结

■ 设备管理的基本概念

- I/O管理的功能
- 设备独立性 定义 优点
- 设备控制块 定义

■ 缓冲技术

- 什么是缓冲，引入缓冲的目的
- 常用的缓冲技术
- 双缓冲技术
- UNIX缓冲管理中的空闲缓冲区的淘汰算法

■ 设备分配

- 常用的设备分配技术
- 独享设备 独享分配 定义
- 共享设备 共享分配 定义
- 虚拟设备 虚拟技术 定义
- Spooling系统 定义

■ I/O控制

- I/O控制的主要功能
- 请求I/O的进程、I/O过程、设备处理进程、中断处理程序之间的同步关系