计算机操作系统

Operating Systems

李　琳

第六章　设备管理

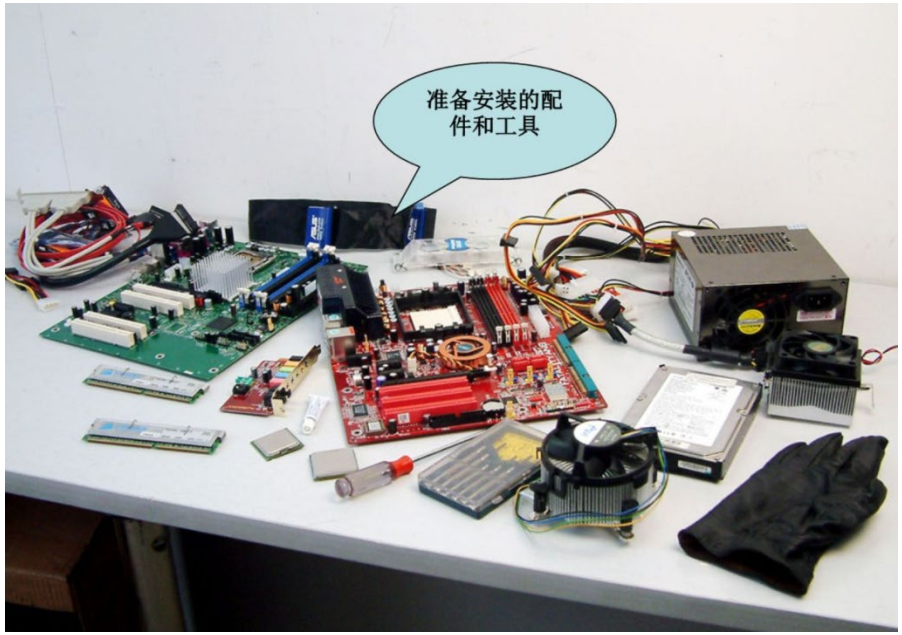
主要解决三个问题

- **设备的抽象**
  - ✓软件如何控制硬件的问题?
  - ✓硬件架构　6.1
- **设备的多样性问题**
  - ✓OS为每一个设备写控制程序?
  - ✓6.1 软件层次　6.3 设备独立性
- **速度不匹配的问题**
  - ✓与CPU之间、与其它外设之间，多样性的结果
  - ✓6.2 缓冲　6.3　spooling

6.1 I/O系统

6.1.1 I/O系统的基本结构

- **I/O系统**是用于实现数据输入、输出及数据存储的系统。
- 在I/O系统中，除了需要直接用于I/O和存储信息的**设备**外，还需要有相应的**设备控制器**和高速**总线**。在有的大、中型计算机系统中，还配置了**I/O通道**或I/O处理机。



全新高通道数扩展机箱帮助LabVIEW FPGA和C系列产品扩展I/O数

6.1 I/O系统

6.1.1 I/O系统的基本结构

- **I/O设备**
  - ✓定义：具体完成数据I/O的设备
  - ✓分类：

按速率分类：低速设备(键盘),中速设备(打印机),高速设备(磁盘).

按信息交换单位分类：字符设备(键盘), 块设备(磁盘)。







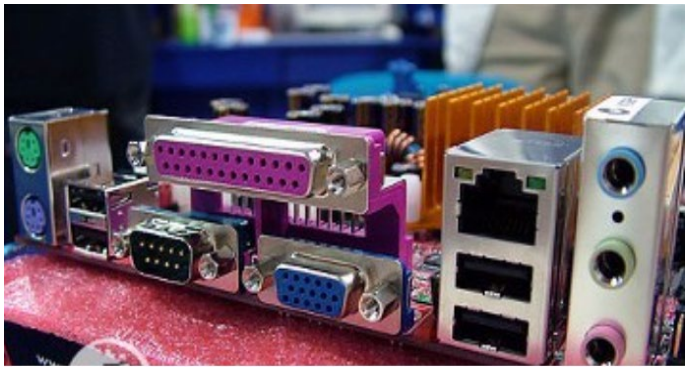

6.1 I/O系统

6.1.1 I/O系统的基本结构

- **设备控制器**
  - ✓I/O设备一般由电子和机械两部分组成，电子部分称作设备控制器或适配器，通常做成印刷电路卡形式．可以插入计算机中，是CPU与I/O设备之间的接口。

电路卡形式
（显卡、声卡等）

接口形式
（主板上的U盘接口、串口等）

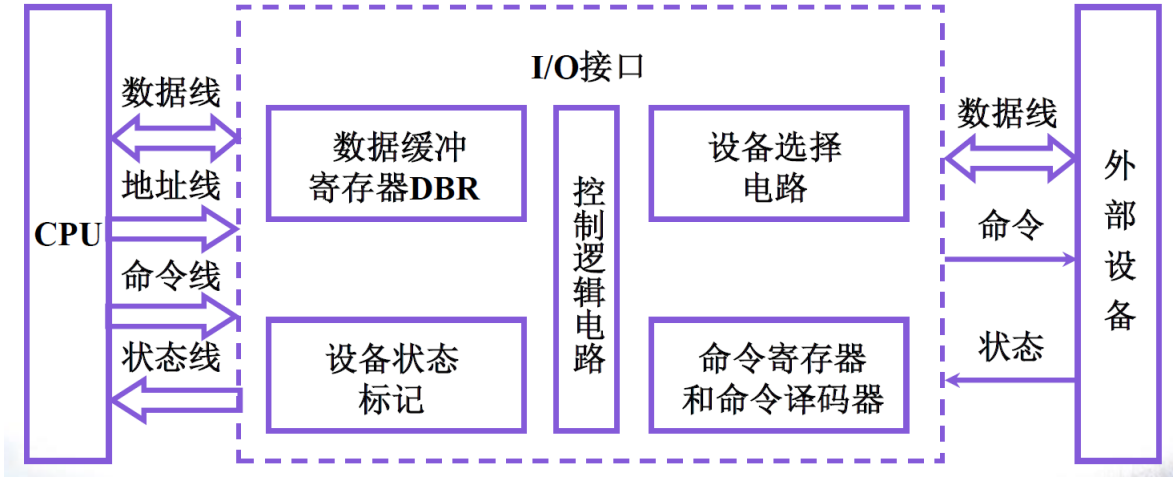


独立形式
（光驱、软驱等）

6.1 I/O系统

6.1.1 I/O系统的基本结构

- **设备控制器**
- 功能
  - ✓接受和识别命令
  - ✓数据交换
  - ✓标志和报告设备状态
  - ✓地址识别
  - ✓数据缓冲
  - ✓差错控制
- 端口（I/O端口）



事实上，操作系统是处理控制器而不是处理设备!!!

6.1 I/O系统

6.1.1 I/O系统的基本结构

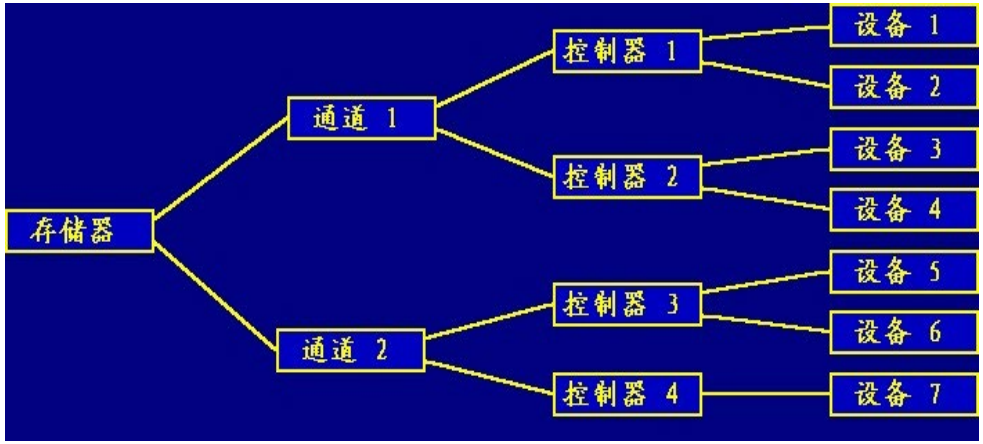
- **I/O通道**
  - ✓是一种特殊的处理器，专门负责输入/输出
  - ✓具有自己的指令系统，一般只有数据传送指令、设备控制指令
  - ✓没有自己的内存，它与CPU共享内存
- 编址（I/O端口）

让原来由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！

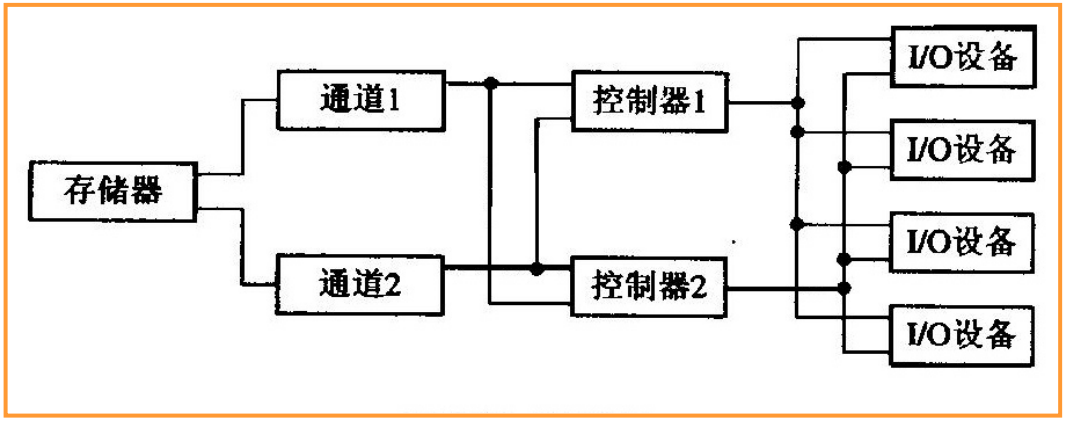
6.1 I/O系统

6.1.1 I/O系统的基本结构

- 单通路I/O系统
  - ✓无冗余设备，容错性差



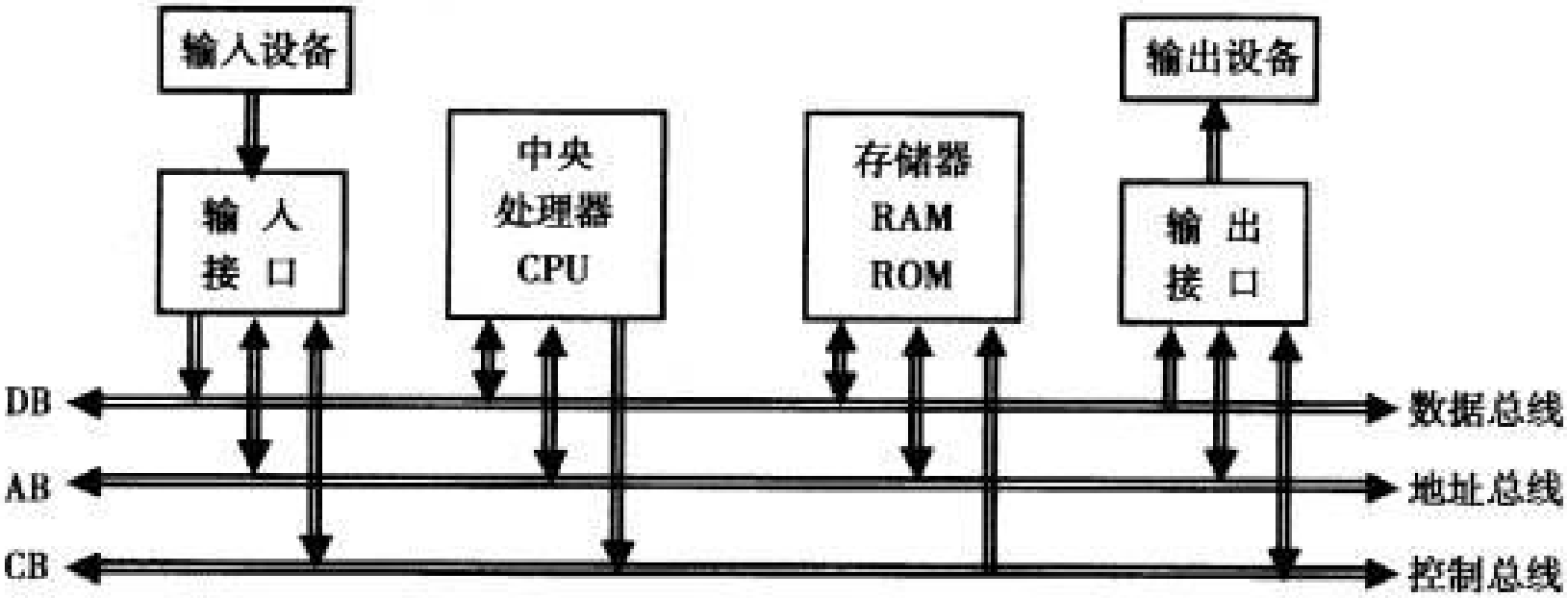
- 多通路I/O系统
  - ✓有冗余设备，容错性佳



6.1 I/O系统

6.1.1 I/O系统的基本结构

- **总线结构**



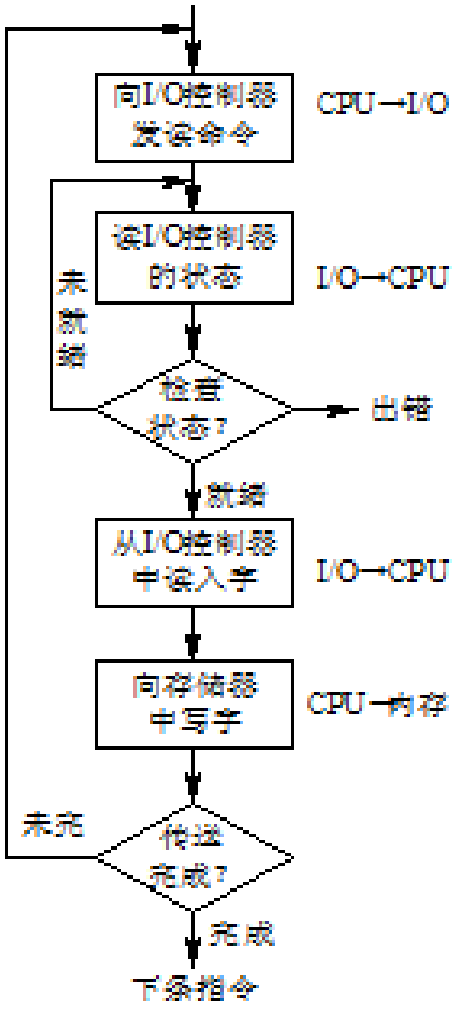
6.1 I/O系统

6.1.2 I/O控制方式

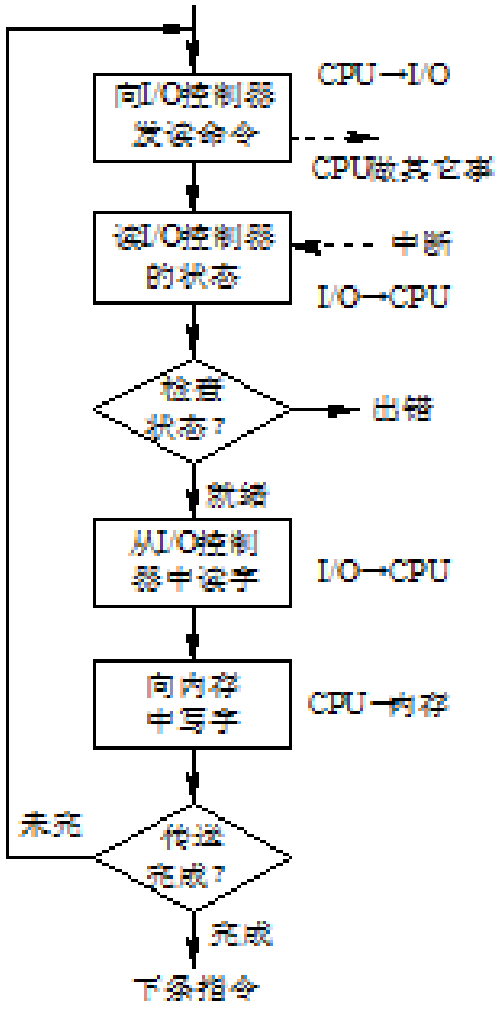
- 设备管理的主要任务之一，是控制设备和内存或CPU之间的数据传送，外围设备和内存之间常用的数据传送控制方式有四种：
  - ✓**程序I/O方式**
  - ✓**中断驱动I/O控制方式**
  - ✓**直接存储器存取方式-DMA**
  - ✓**通道控制方式**

6.1 I/O系统

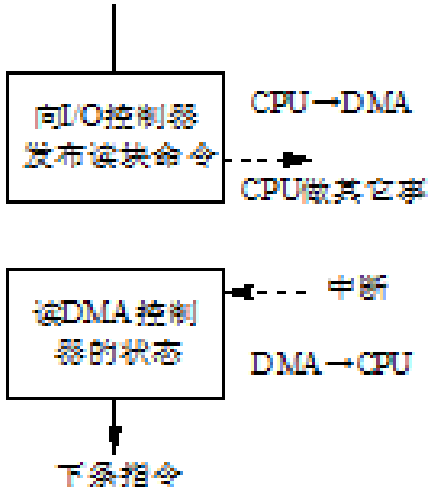
6.1.2 I/O控制方式



(a) 程序I/O方式



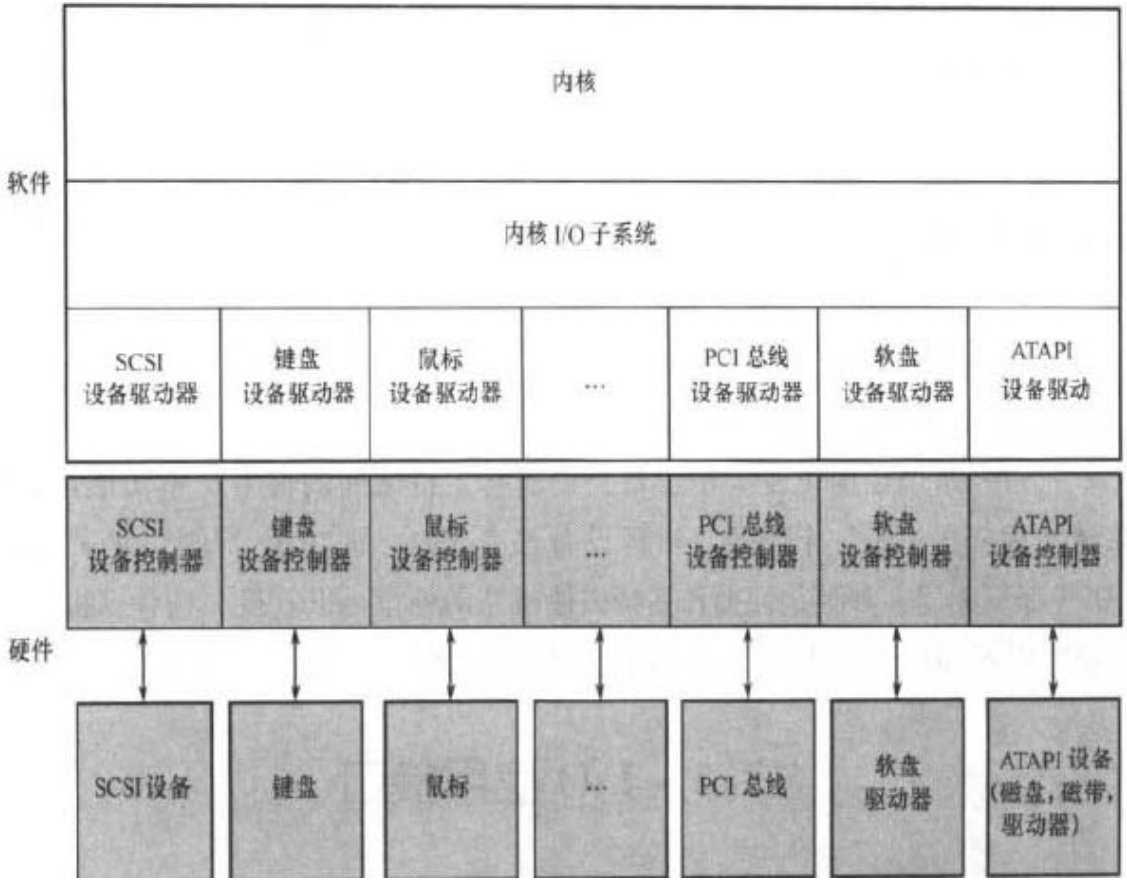
(b) 中断驱动方式

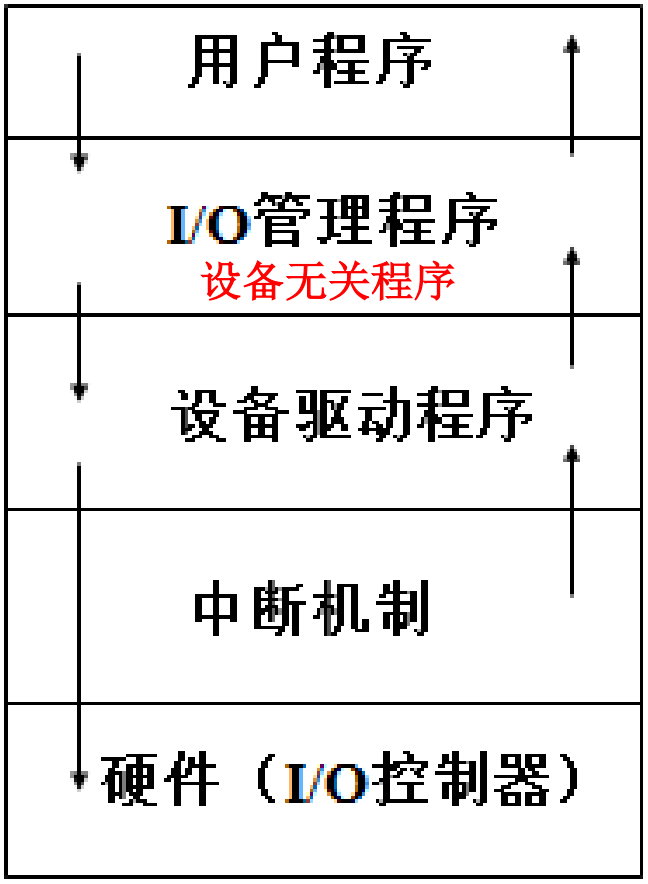


(c) DMA方式

6.1 I/O系统

6.1.3 I/O软件层次





6.1 I/O系统

6.1.4 设备驱动程序

- 统一管理
  - ✓为了隐藏不同设备的细节与特点，操作系统内核设计成使用**设备驱动程序模块**的接口。设备驱动由控制器厂家产生，符合特定OS的要求，OS通过调用驱动程序向控制器发送指令从而控制设备
- 功能与作用
  - 解释I/O进程的命令
  - 向设备传递参数、了解设备状态
  - 中断处理
  - 检查I/O合法性并安排设备服务顺序

Linux的设备驱动开发

- Linux下设备基本上分为三部分：字符设备、块设备、网络设备

（1）根据外设的接口协议编写相应的功能函数原型；

（2）将得到的函数原型用linux的驱动程序接口封装起来；

（3）将封装好的模块动态加入到linux内核中

接口函数为一个函数的集合，集合中包括了所有可能对该设备进行操作的动作，用一个struct file_operations数据结构来表示

```c
#include <linux/init.h>
#include <linux/kernel.h>
#include <linux/module.h>

static int hello_init(void)
{
    printk("Hello! This is the helloworld module!\n");
    return 0;
}

static void hello_exit(void)
{
    printk("Module exit! Bye Bye!\n");
    return;
}

module_init(hello_init);
module_exit(hello_exit);
MODULE_LICENSE("GPL");
```

```
CONFIG_MODULE_SIG=n
KERNELDIR = /lib/modules/$(shell uname -r)/build
PWD := $(shell pwd)

obj-m := hello.o
default:
	$(MAKE) -C $(KERNELDIR) M=$(PWD) modules
clean:
	rm -f hello.ko hello.mod.c hello.mod.o hello.o
```

```
root# insmod model.ko        //添加模块命令
root# rmmod model            //删除模块命令
```

试一试

1、I/O指令实现的数据传送通常发生在

A．I/O设备和I/O端口之间　B．通用寄存器和I/O设备之间

C．I/O端口和I/O端口之间　D．通用寄存器和I/O端口之间

2、系统将数据从磁盘读到内存的过程包括以下操作：

①DMA控制器发出中断请求 ②初始化DMA控制器并启动磁盘③从磁盘传输一块数据到内存缓冲区④执行“DMA结束”中断服务程序。正确的执行顺序是

A．③→①→②→④　　B．②→③→①→④

C．②→①→③→④　D．①→②→④→③

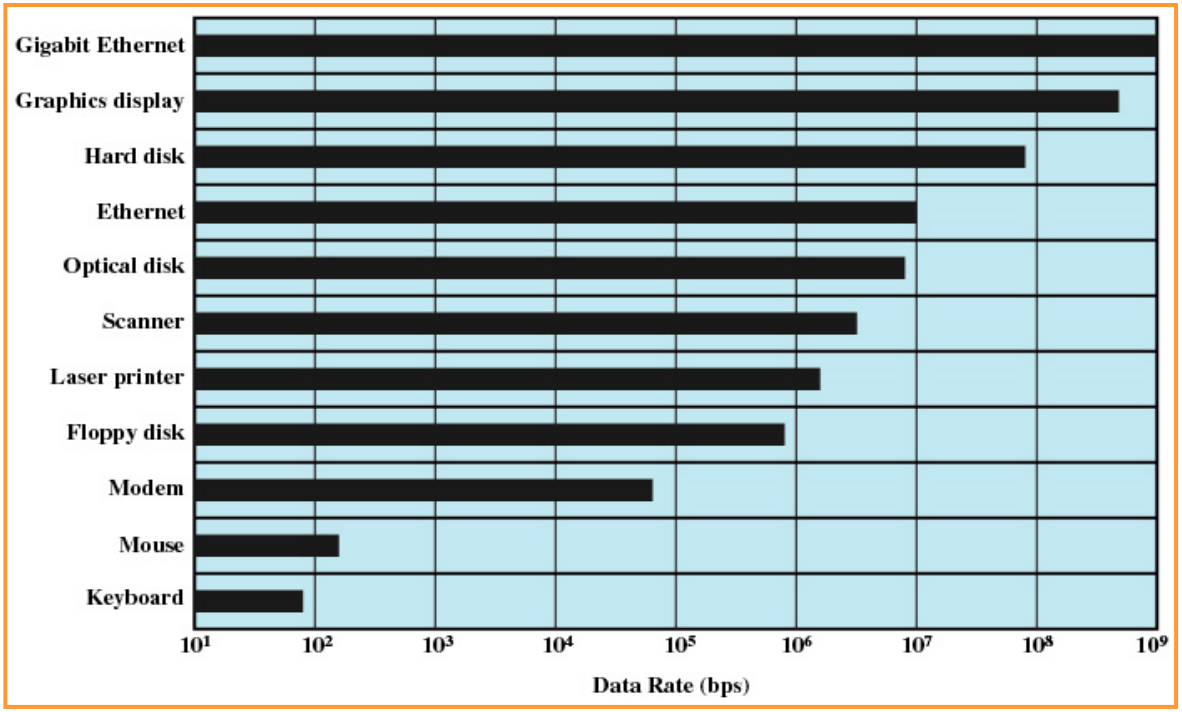
3、用户程序发出磁盘I/O请求后，系统的正确处理流程是（　）

A. 用户程序—>系统调用处理程序—>中断处理程序—>设备驱动程序

B. 用户程序—>系统调用处理程序—>设备驱动程序—>中断处理程序

C．用户程序—>设备驱动程序—>系统调用处理程序—>中断处理程序

D. 用户程序—>设备驱动程序—>中断处理程序—>系统调用处理程序



Device	Data rate
Keyboard	10 bytes/sec
Mouse	100 bytes/sec
56K modem	7 KB/sec
Telephone channel	8 KB/sec
Dual ISDN lines	16 KB/sec
Laser printer	100 KB/sec
Scanner	400 KB/sec
Classic Ethernet	1.25 MB/sec
USB (Universal Serial Bus)	1.5 MB/sec
Digital camcorder	4 MB/sec
IDE disk	5 MB/sec
40x CD-ROM	6 MB/sec
Fast Ethernet	12.5 MB/sec
ISA bus	16.7 MB/sec
EIDE (ATA-2) disk	16.7 MB/sec
FireWire (IEEE 1394)	50 MB/sec
XGA Monitor	60 MB/sec
SONET OC-12 network	78 MB/sec
SCSI Ultra 2 disk	80 MB/sec
Gigabit Ethernet	125 MB/sec
Ultrium tape	320 MB/sec
PCI bus	528 MB/sec
Sun Gigaplane XB backplane	20 GB/sec

I/O系统各组成速度差异很大！！

6.2 缓冲管理

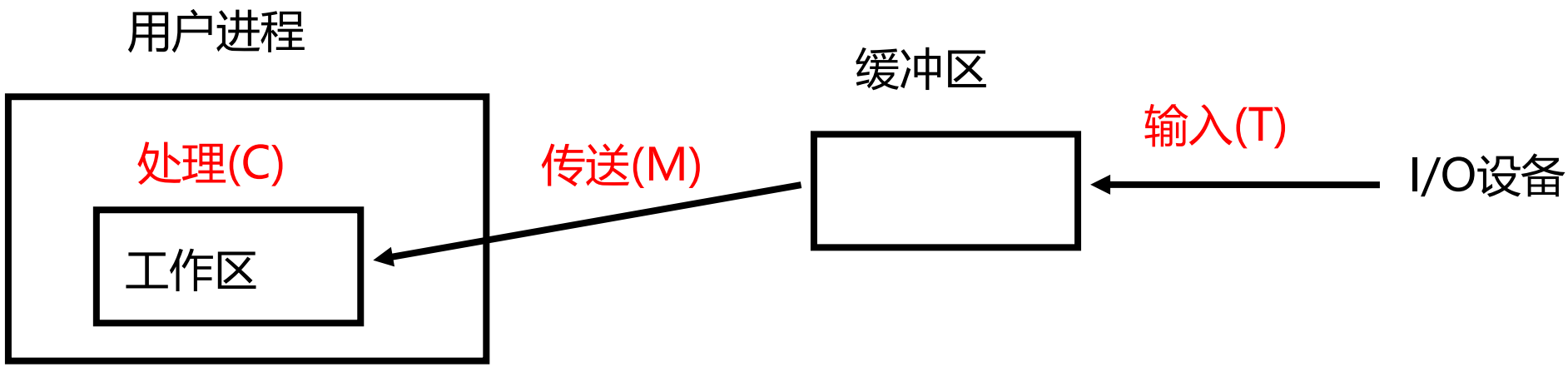
6.2.1 缓冲的引入

- 引入缓冲的原因
  - ✓缓和CPU和I/O设备的矛盾
  - ✓减少CPU中断的频率
  - ✓提高CPU和I/O设备的并行性

6.2 缓冲管理

6.2.2 单缓冲

- 原理



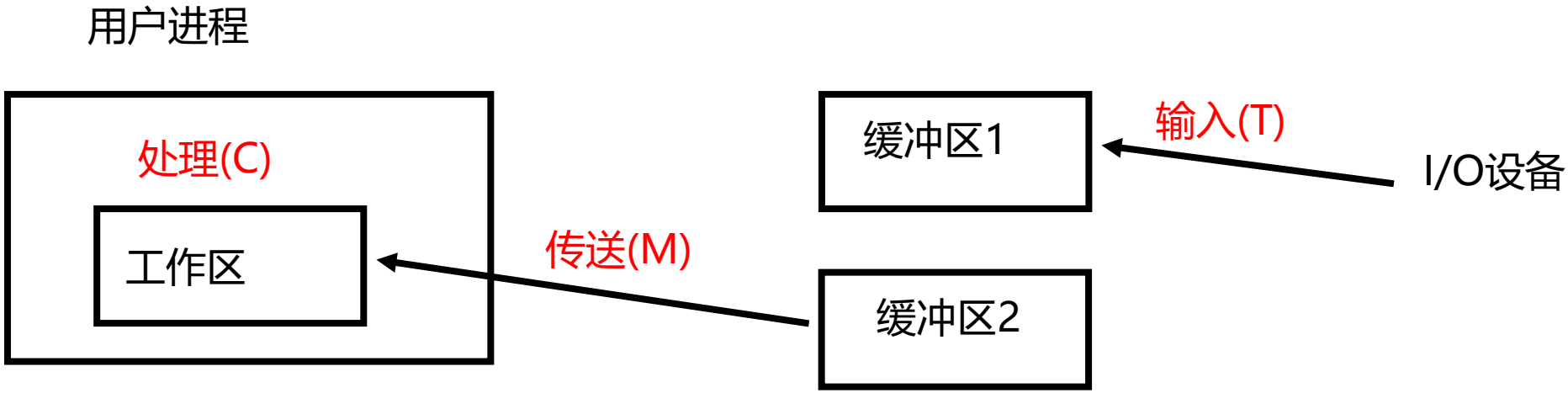
- 处理时间
  - ✓ **M+T，T>C；**
  - ✓ **M+C，T<C；**

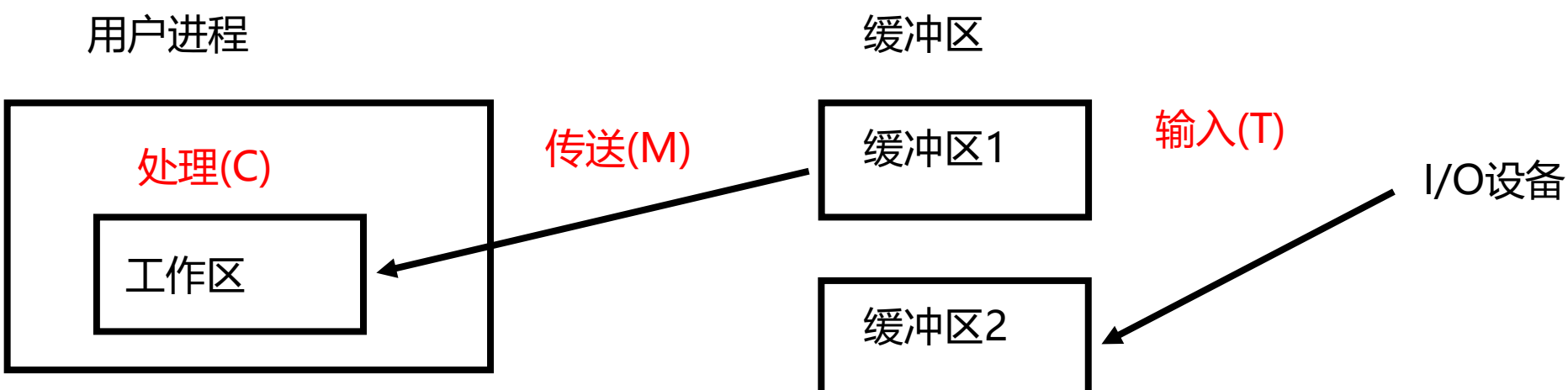
6.2 缓冲管理

6.2.3 双缓冲

- 原理

- 处理时间：**MAX(C,T)**



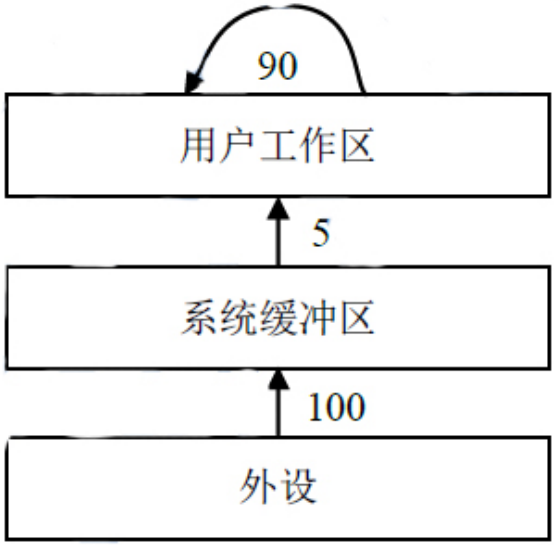


例：

- 设系统缓冲区和用户工作区使用单缓冲，从外读入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(如下图所示)。进程从外设读入并分析2个数据块的最短时间是 多少？如果采用双缓冲区应该如何设置，如果对换时间不计，那么读入并分析2个数据块的时间又是多少?

单缓冲：100+5+100+5+90=300

双缓冲：100+100+5+90=295



6.2 缓冲管理

6.2.4 循环缓冲

- 循环缓冲

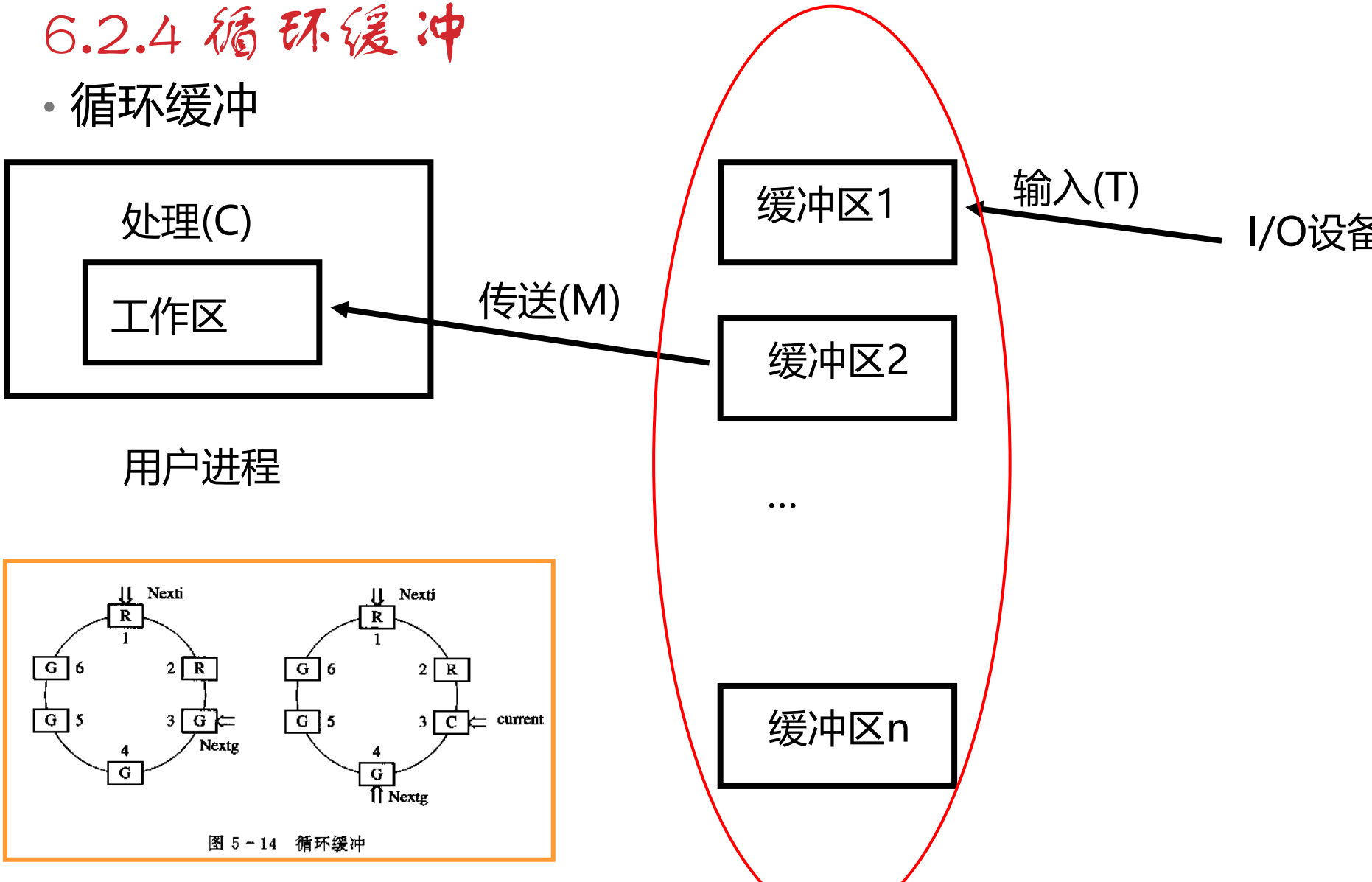
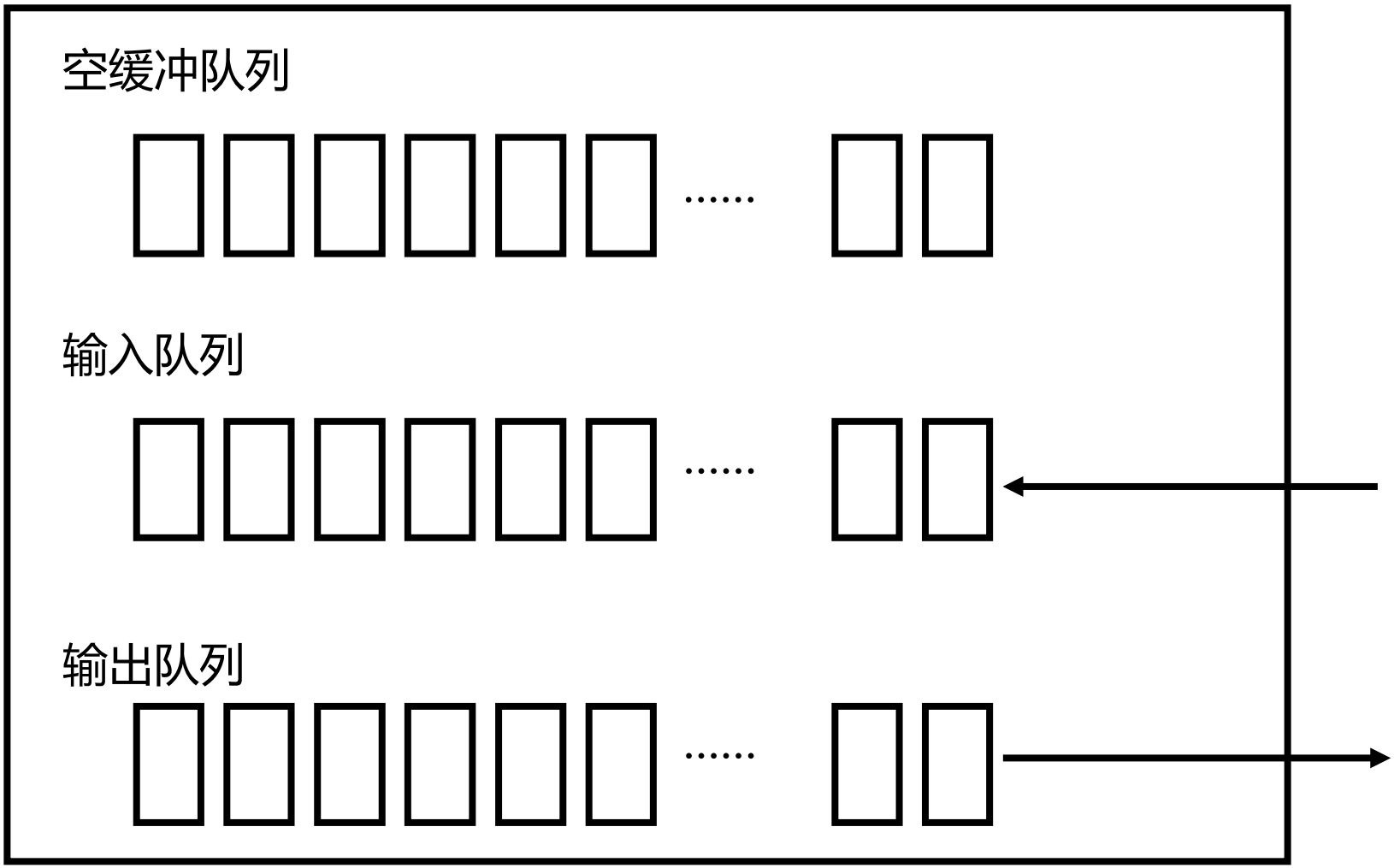


图 5－14　循环缓冲

6.2 缓冲管理

6.2.5 缓冲池



6.2 缓冲管理

6.2.5 缓冲池

- 基本操作（对三种队列都有）
  - ✓GetBuf(); 获取一个结点操作
  - ✓PutBuf()；连接一个结点操作
- 工作方式
  - ✓收容输入；—>空队列GetBuf—>输入数据—>输入队列PutBuf
  - ✓提取输入；—>输入队列GetBuf—>提取数据给用户进程—>空队列PutBuf
  - ✓收容输出；—>空队列GetBuf—>输出数据—>输出队列PutBuf
  - ✓提取输出；—>输出队列GetBuf—>提取数据给设备—>空队列PutBuf

6.3 设备分配

6.3.1 基本问题

- **设备类型对设备分配的影响**
- 独占设备
  - ✓指在一段时间内只允许一个用户(进程)使用的设备
  - ✓独占设备的分配可能会引起进程死锁
- 共享设备
  - ✓指在一段时间内允许多个进程同时访问的设备
  - ✓必须是可寻址的和可随机访问的设备，比如磁盘
- 虚拟设备
  - ✓指通过某种技术将一台独占设备变换为能供若干个用户共享的设备
  - ✓**SPOOLing技术**是一类典型的虚拟设备技术

6.3 设备分配

6.3.1 基本问题

- **设备分配算法**
  - ✓先来先服务
  - ✓优先级算法
  - ✓。。。
- **设备分配的安全性**
  - ✓安全方式：串行分配，一个一个分配
  - ✓不安全方式：并行分配，可同时分配多个设备资源

6.3 设备分配

6.3.2 设备独立性

- 基本定义：用户程序独立于具体使用的物理设备
- 实现方式：逻辑设备与物理设备
  - ✓核心：采用虚拟技术，编写应用程序时不考虑实际的物理设备
  - ✓物理设备：应用程序实际执行时，使用的特定类型设备（执行程序时）
  - ✓逻辑设备：是对一组具备相同功能物理设备的抽象（编写程序时）
  - ✓基本作法：程序执行时，使用逻辑设备表完成逻辑设备到物理设备的映射
- 逻辑设备表(LUT)

逻辑设备名	物理设备名	驱动程序入口
/dev/tty	3	0x00001024
/dev/printer	5	0x00002034
…	…	…

6.3 设备分配

6.3.2 SPOOLING技术

- Simultaneaus Periphernal Operating On-Line

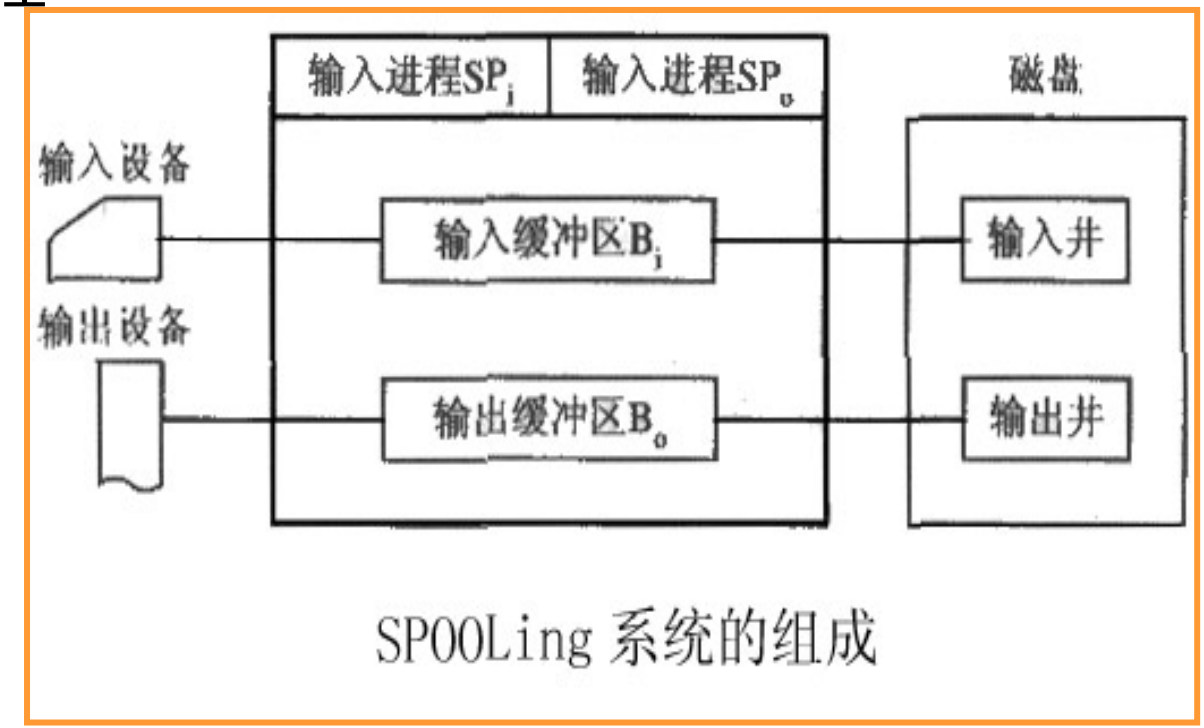
- 目的：为缓和CPU的高速性与I/O设备低速性之间的矛盾
- 早期：引入脱机输入、脱机输出技术：利用专门的外围控制机，将低速I/O设备上的数据传送到高速磁盘上；或者相反。
- 现在：在引入多道程序技术后，可以利用系统中的进程，来模拟外围控制机功能

- 把低速I/O设备上的数据传送到高速磁盘上；
- 把数据从磁盘传送到低速输出设备上。
- 主机控制的脱机输入、输出功能

6.3 设备分配

6.3.2 SPOOLING技术

- SPOOLING的基本组成
  - 输入井和输出井
  - 输入缓冲区和输出缓冲区
  - 输入进程和输出进程
- SPOOLing系统的典型应用
  - 共享打印机
  - 邮箱系统



SPOOLing 系统的组成

打印问题

- 假定你打开了Word、Excel等程序进行打印，总共发出了8个文档打印任务，但打完第3个文档后就没纸了，你关闭了Word、Excel去买纸，等你将纸买来装好，打印机自动从第4个文档开始接着打印。请问这是采用的什么技术？

- 假定你打开了Word、Excel等程序进行打印，总共发出了8个文档打印任务，但打完第3个文档后就没纸了，你只好关闭计算机系统去买纸。等你将纸买来装好，打开计算机系统，打印机会自动从第4个文档开始接着打印。请问这是采用的什么技术？

- 假定你打开了Word、Excel等程序进行打印，总共发出了8个文档打印任务，但打完第3个文档后就没纸了，你只好关闭计算机系统去买纸。等你将纸买来装好，没有开机打印机也会自动从第4个文档开始接着打印。请问这是采用的什么技术？

试一试

1、程序员利用系统调用打开I/O设备时，通常使用的设备标识是（ ）

A．逻辑设备名　B. 物理设备名　C. 主设备号　D. 从设备号

2、用户程序发出磁盘I/O请求后，系统的处理系统的处理流程是：用户程序→系统调用处理程序→设备骆动程序→中断处理程序。其中，计算数据所在磁盘的柱面号、磁头号、扇区号的程序是（）

A. 用户程序　　B. 系统调用处理程序　C. 设备驱动程序　D. 中断处理程序

3、下面的4个选项中，不属于设备管理的功能的是（）

A、实现外围设备的启动　　B、实现对磁盘的驱动调度

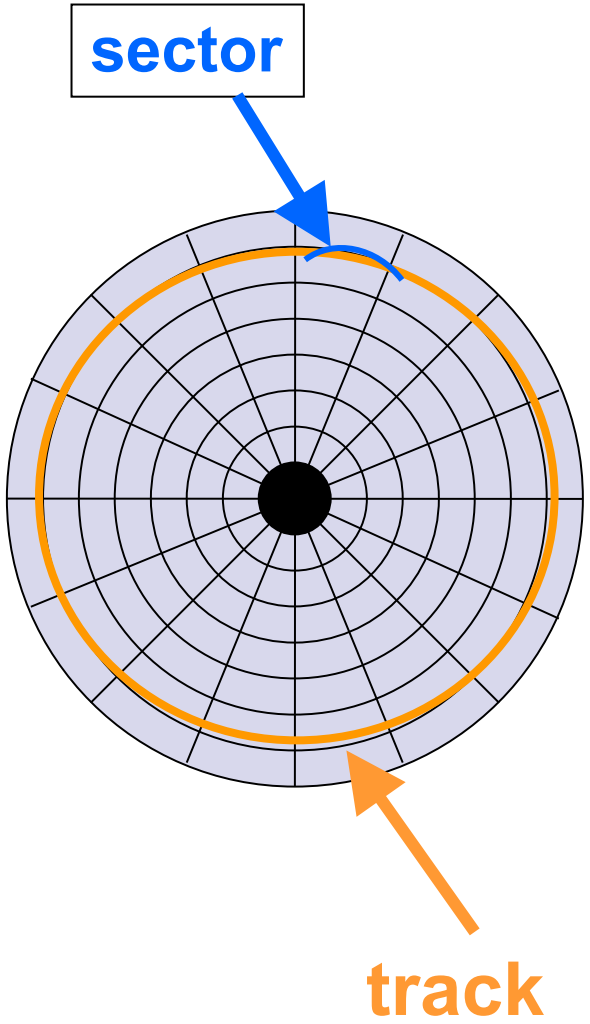
C、存储空间的分配与回收　　D、处理外围设备的中断事件

4、在采用SPOOLING技术的系统中，用户作业的打印输出结果首先被送到（ ）

A．磁盘固定区域　B．内存固定区域　C．终端　D．打印机

6.4 磁盘存储管理

6.4.1 磁盘工作原理



磁盘数据定位：

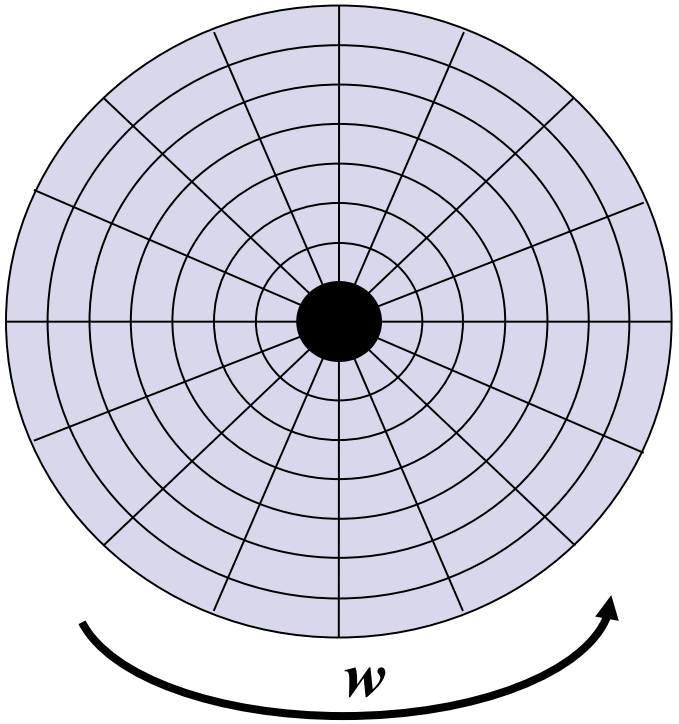
- 磁头号
- 磁道号
- 扇区号

6.4 磁盘存储管理

6.4.1 磁盘工作原理

磁盘访问时间

- 寻道时间 *Ts* ： *Ts* = 8 ~10ms
- 旋转延迟时间 *Tr* : *Tr* = 1/(2*r*)
- 传输时间 *Tt*: *Tt* = *b*/(*rN*)
- *r*: 为转速; *N*: 每磁道字节数; *b*: 传输字节数



转速(RPM)	*Ts*(ms)	*Tr*(ms)	*Tt*(ms)
5400	**8-10**	**5.5**	**0.1**
7200	**8-10**	**4.3**	**0.07**
10000	**8-10**	**3**	**0.05**

注： *b*/*N* = 1/128

T = *Ts* + *Tr* + *Tt*

Ts > *Tr* >> *Tt*

例

某磁盘的转速为10,000转/分，平均寻道时间是6ms，磁盘传输速率是20MB/s，磁盘控制器延迟为0.2ms，读取一个4KB的扇区所需平均时间约为多少？

60*1000/10000/2+6+0.2+1000/（20*1024）*4
=12+6+0.2+0.2
=18.4

6.4 磁盘存储管理

6.4.2 磁盘调度算法

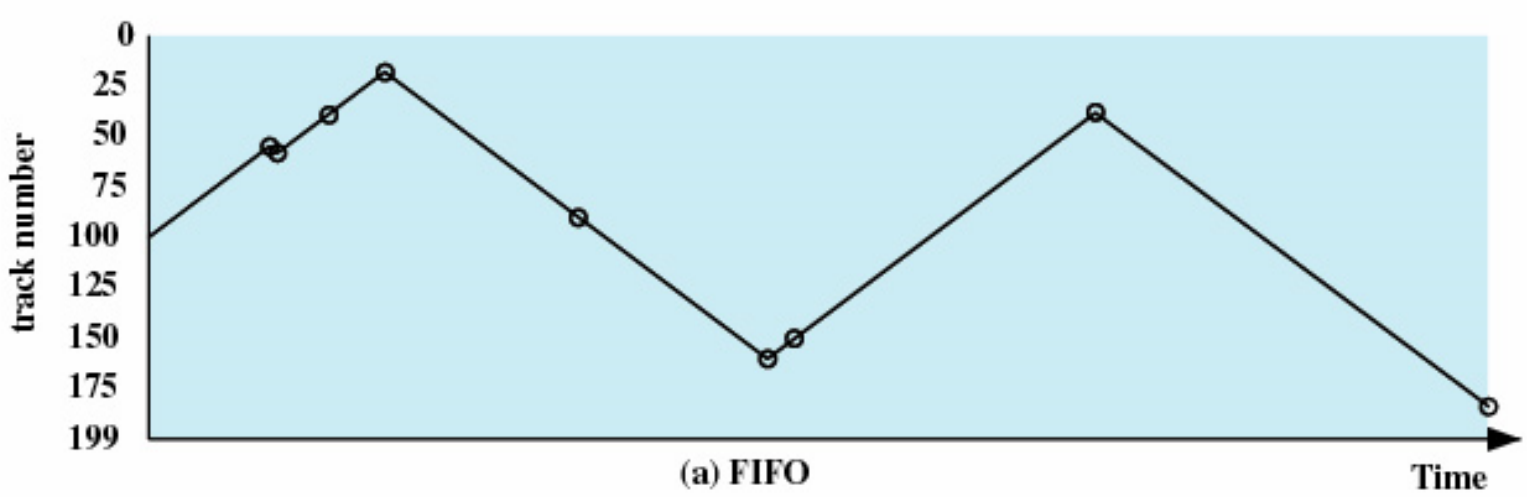
- 磁盘是可供多个进程共享的设备，当有多个进程都要求访问磁盘时，应采用一种最佳调度算法，以使各进程对磁盘的平均访问时间最小。
- 由于在访问磁盘的时间中，主要是寻道时间，因此，磁盘调度的目标，是使磁盘的平均寻道时间最少。

6.4 磁盘存储管理

6.4.2 磁盘调度算法

先来先服务调度算法

- 根据进程请求磁盘的先后次序进行调度
- 特点：公平、简单，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
- 磁道请求序列：55,58,39,18,90,160,150,38,184
- 磁道访问顺序： 55,58,39,18,90,160,150,38,184



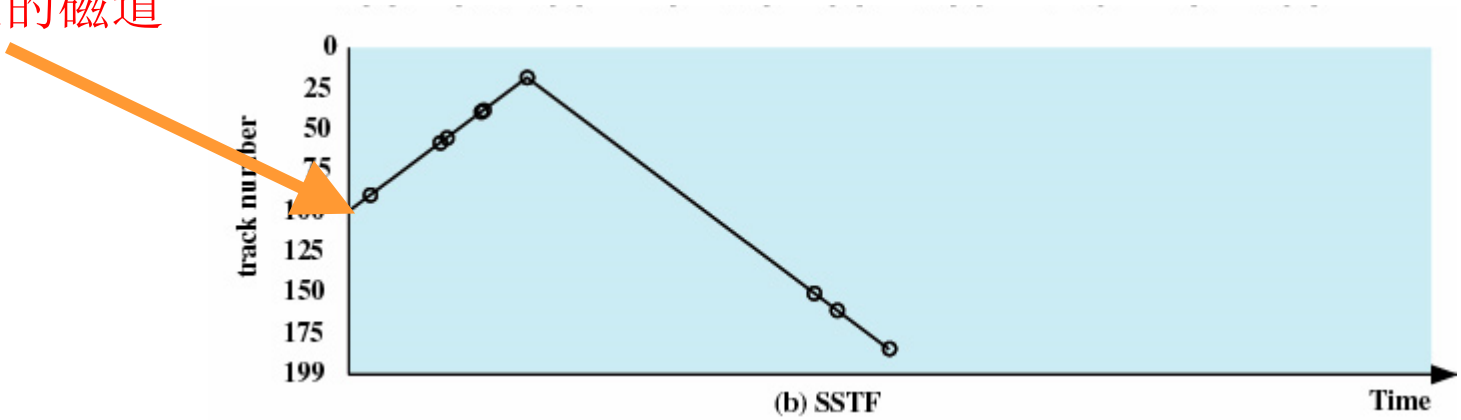
6.4 磁盘存储管理

6.4.2 磁盘调度算法

最短寻道时间优先调度算法（SSTF）

- 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
- 特点： SSTF的平均每次磁头移动距离，明显低于FCFS，因而SSTF较FCFS有更好的寻道性能。 “磁臂粘着”现象。
- 磁道请求序列：55,58,39,18,90,160,150,38,184
- 磁道访问序列：90,58,55,39,38,18,150,160,184

当前磁头所在的磁道



6.4 磁盘存储管理

6.4.2 磁盘调度算法

- **FCFS和SSTF算法比较**

（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	

图 5－23　FCFS 调度算法

（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	

图 5－24　SSTF 调度算法

6.4 磁盘存储管理

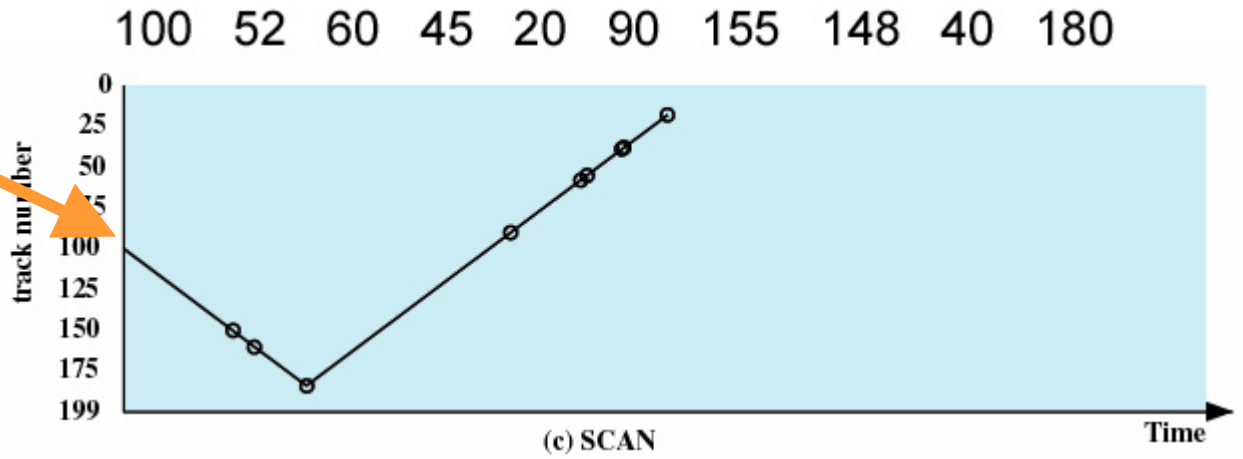
6.4.2 磁盘调度算法

扫描算法（SCAN）

- 在磁头移动方向上，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。
- 特点： 有效解决磁臂粘着现象。
- 磁道请求序列：55,58,39,18,90,160,150,38,184
- 磁道访问序列：150,160,184,90, 58,55,39,38,18

当前磁头所在的磁道

当前磁头移动方向：
磁道号增加。



6.4 磁盘存储管理

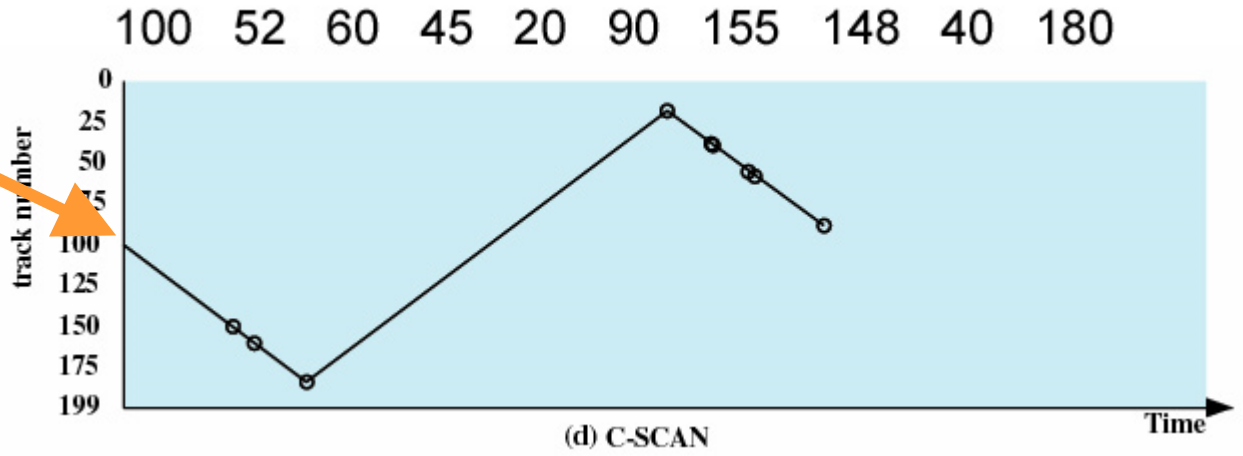
6.4.2 磁盘调度算法

循环扫描 算法（CSCAN）

- 单向SCAN。只在一个磁头移动方向上满足磁道访问请求。
- 特点： 有效降低磁道请求最大延迟。
- 磁道请求序列：55,58,39,18,90,160,150,38,184
- 磁道访问序列：150,160,184,18,,38,39,55,58,90

当前磁头所在的磁道

仅磁道号增加时，可满足磁道访问。



6.4 磁盘存储管理

6.4.2 磁盘调度算法

- **SCAN和CSCAN算法比较**

（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	

图 5－25　SCAN 调度算法示例

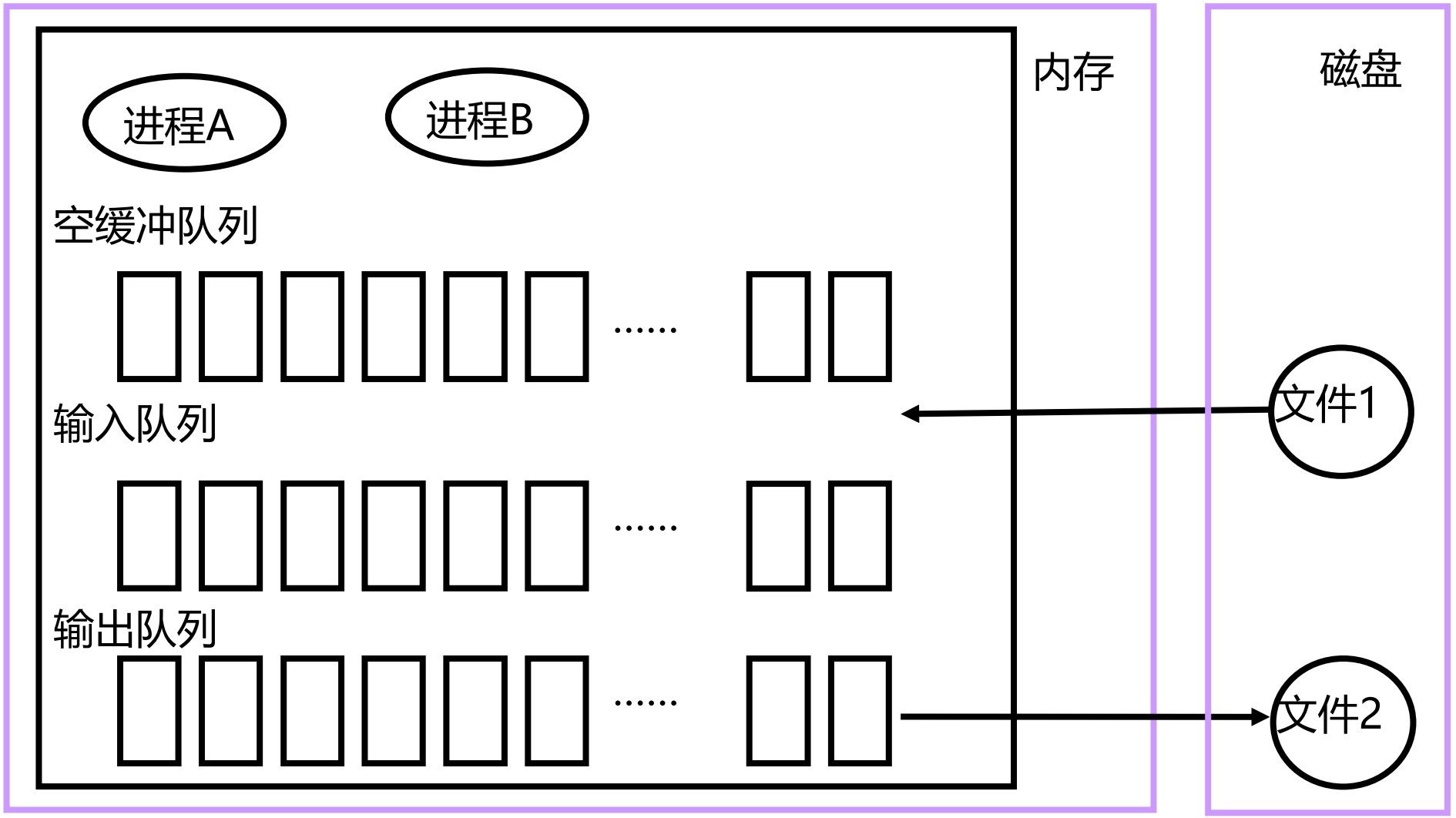
（从100#磁道开始，向磁道号增加方向访问）

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：27.5	

图 5－26　CSCAN 调度算法示例

6.4 磁盘存储管理

6.4.3 磁盘高速缓存（Disk Cache）



6.4 磁盘存储管理

6.4.3 磁盘高速缓存（Disk Cache）

- 数据交付
- 置换算法：LRU
- 周期性写回

6.4.4 提高磁盘I/O速度的其它方法

- 提前读
- 延迟写
- 优化物理块分布