



南京大学  
NANJING UNIVERSITY

# 计算机与操作系统

## 第九讲 设备管理

南京大学软件学院



# 第九讲 设备管理

- \* 9.1 I/O硬件
- \* 9.2 I/O缓冲
- \* 9.3 独占型外围设备的分配
- \* 9.4 共享型设备的驱动调度
- \* 9.5 虚拟设备
- \* 9.6 I/O设备管理的实现与层次



南京大学  
NANJING UNIVERSITY

# 本主题教学目标

1. 复习、了解I/O设备、I/O控制方式
2. 掌握I/O缓冲区的设计
3. 掌握磁盘调度
4. 掌握I/O软件系统的设计与实现
5. 掌握虚拟设备



# 设备管理简介

- \* 设备管理是操作系统中最庞杂和琐碎的部分，通常使用I/O中断、缓冲区管理、通道、设备驱动调度等多种技术
- \* 目标：克服由于设备和CPU速度的不匹配所引起的问题，使主机和设备并行工作，提高设备使用效率
- \* 另一方面，为了方便用户或高层进程使用，设备管理还对各种设备进行抽象，配置驱动程序，提供统一界面，屏蔽设备的物理细节和操作过程



南京大學  
NANJING UNIVERSITY

# 设备管理的功能

- 设备中断处理
- 缓冲区管理
- 设备的分配和去配
- 设备驱动调度
- 虚拟设备及其实现



南京大學  
NANJING UNIVERSITY

# 9.1 I/O硬件



# I/O 系統

- \* I/O 系統：I/O 設備及其接口線路、控制部件、通道和管理軟件
- \* I/O 硬件
  - \* 設備與介質
  - \* 設備控制器
  - \* I/O 控制方式



# I/O设备分类

## \* 设备类型

- \* 字符型设备

- \* 块存储设备

- \* 网络设备

## \* 从直观使用的角度

- \* 输入设备

- \* 输出设备

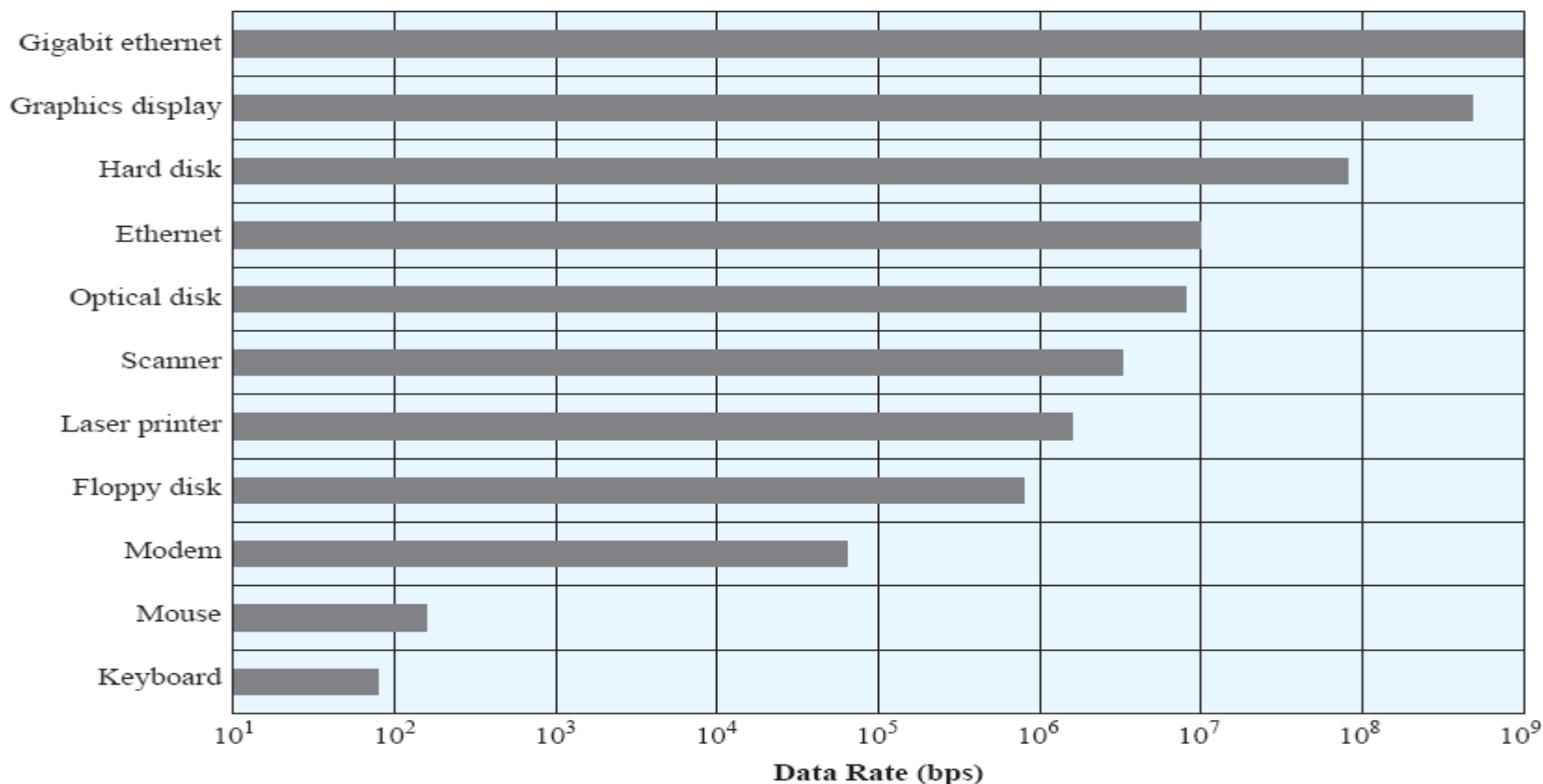
- \* 输入/输出设备





# I/O设备的速度差异

\* 数据率：数据传送率可能相差几个数量级





# I/O控制方式

## \* 程序控制I/O (轮询方式)

- \* 处理器代表给I/O模块发送一个I/O命令，该进程进入忙式等待 (busy-waiting)，等待操作的完成，然后才可以继续操作

## \* 中断驱动 I/O

- \* 处理器代表进程向I/O模块发出一个I/O命令，然后继续执行后续指令，当I/O模块完成工作后，处理器被该模块中断
- \* 如果该进程不需要等待I/O完成，则后续指令可以仍是该进程中的指令，否则，该进程在这个中断上挂起，处理器执行其他工作

## \* 直接存储器访问(DMA)

- \* 一个DMA模块控制主存和I/O模块之间的数据交换。为传送一块数据，处理器给DMA模块发请求，只有当整个数据块传送结束后，处理器才被中断

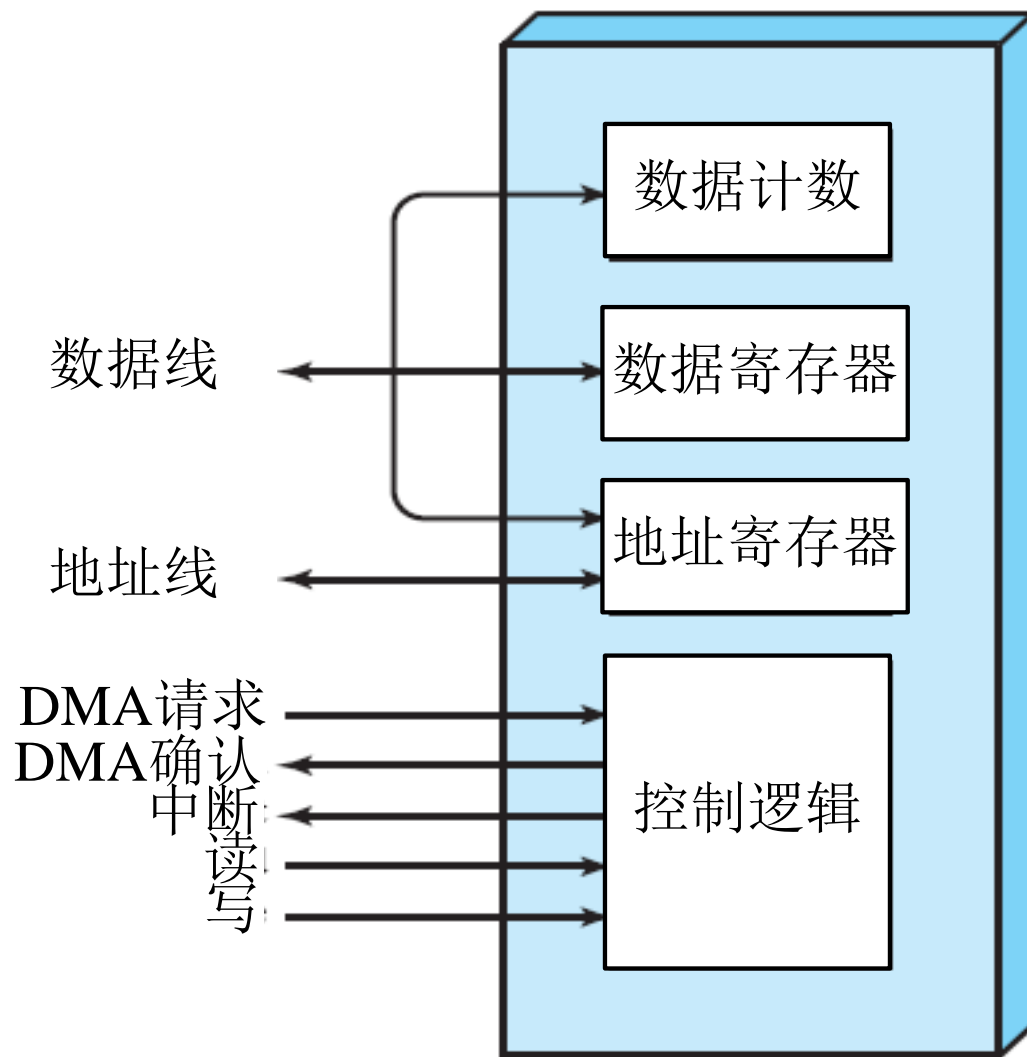


# DMA

- \* 替代 CPU, DMA 控制内存与磁盘之间数据传输
- \* 周期窃取是指利用 CPU 不访问存储器的那些周期来实现 DMA 操作, 此时 DMA 可以使用总线而不用通知 CPU 也不会妨碍 CPU 的工作
- \* 周期挪用并不减慢 CPU 的操作, 但可能需要复杂的时序电路, 而且数据传送过程是不连续的和不规则的
- \* 在这种方法中, 每当 I/O 设备发出 DMA 请求时, I/O 设备便挪用或窃取总线占用权一个或几个主存周期, 而 DMA 不请求时, CPU 仍继续访问主存
- \* 不发生中断
- \* 不保存上下文

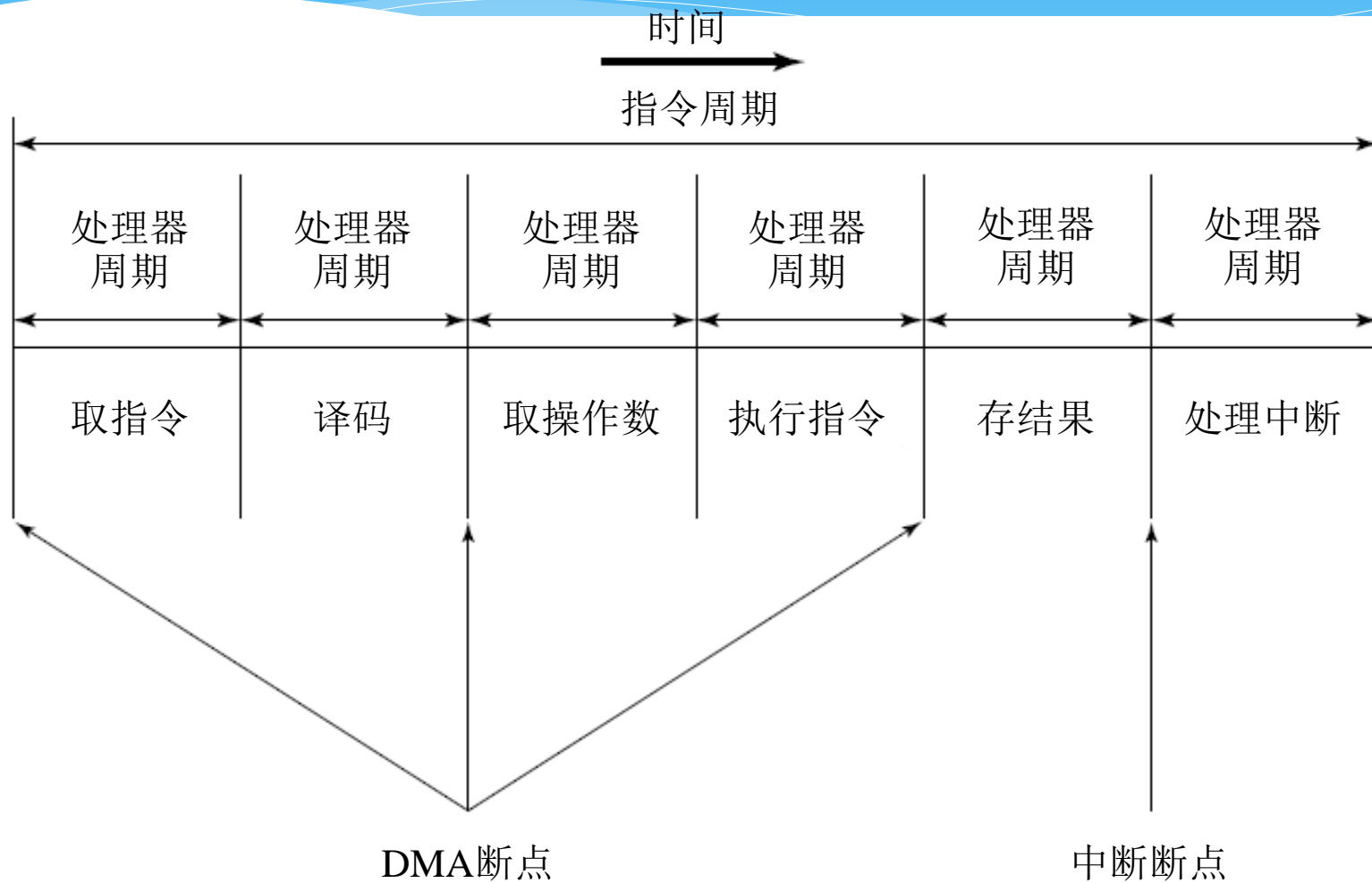


# DMA



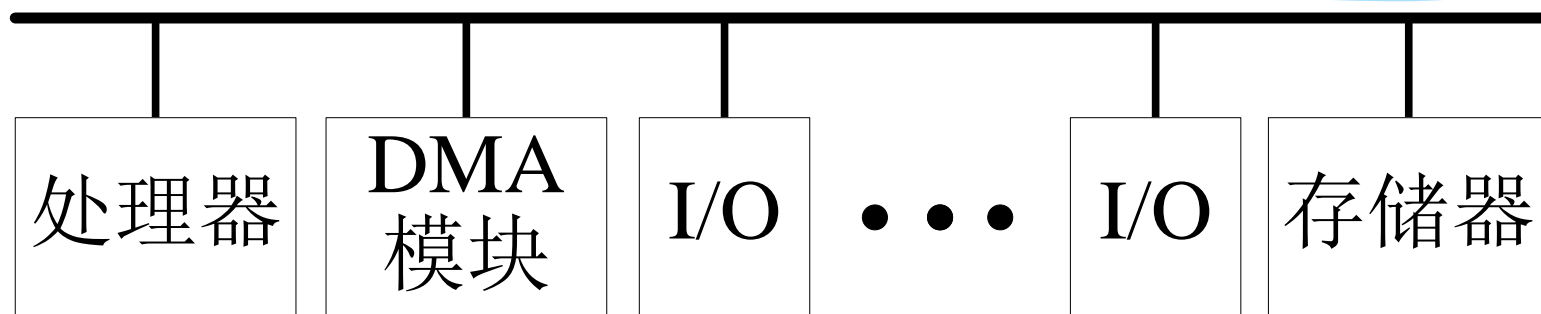


# DMA





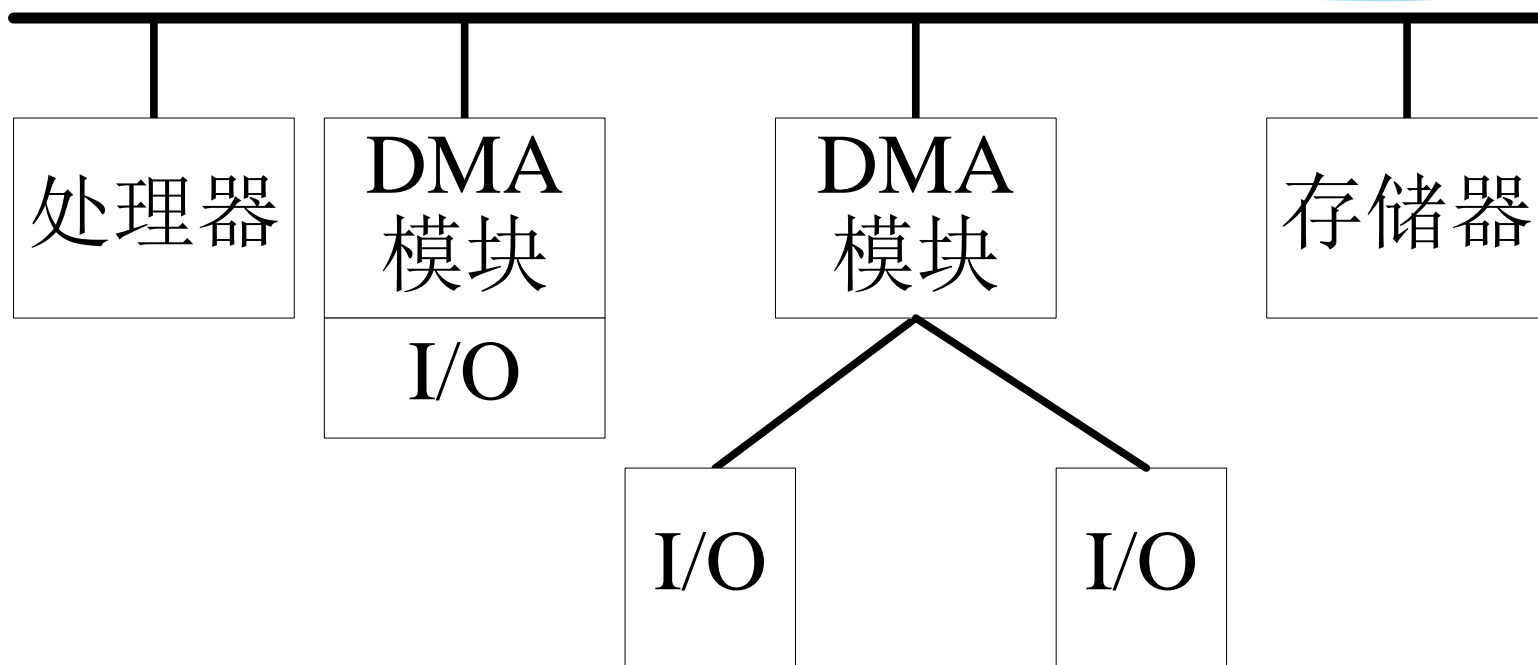
# DMA



(a) 单总线，分离的DMA



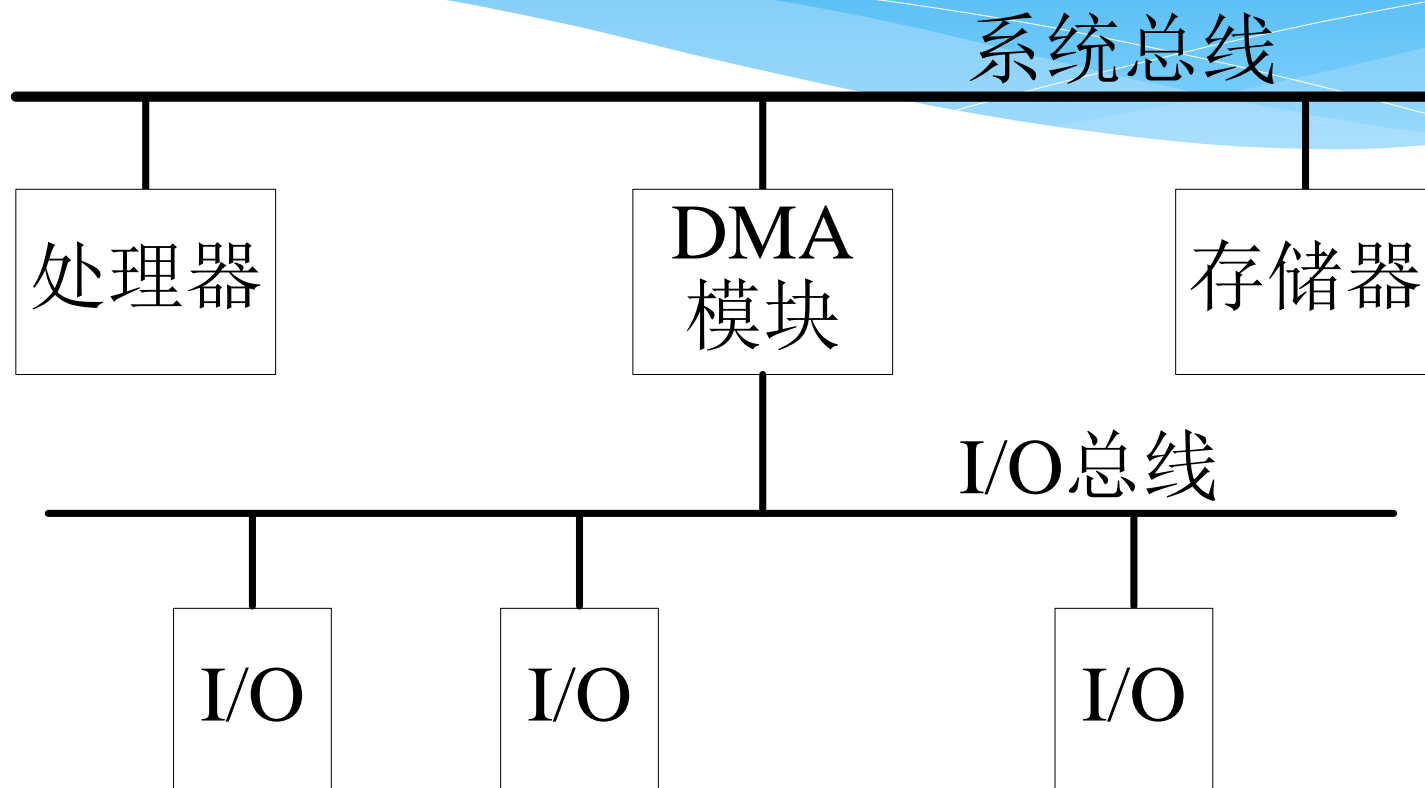
# DMA



(b) 单总线，集成的DMA-I/O



# DMA



(c) 可选的DMA配置





# I/O控制方式

## \* 通道方式

- \* 为获得CPU和外围设备间更高的并行工作能力，为让种类繁多，物理特性各异的外围设备能以标准的接口连接到系统中，计算机系统引入了自成独立体系的通道结构

- \* 采用通道后的I/O操作过程：

CPU在执行主程序时遇到I/O请求，它启动指定通道上选址的外围设备，一旦启动成功，通道开始控制外围设备进行操作。CPU就可执行其他任务并与通道并行工作，直到I/O操作完成。通道发出操作结束中断时，CPU才停止当前工作，转向处理I/O操作结束事件



# 具有通道的I/O

- \* 通道状态字(CSW)

通道状态字是存放在内存固定单元的控制字，专门用于记录通道和设备执行操作的情况

- \* 输入输出中断



# 具有通道的I/O系统

- \* 输入输出指令

- \* 启动输入输出指令(SIO)

- \* 查询输入输出指令(TIO)

- \* 查询通道指令(TCH)

- \* 停止输入输出指令(HIO)

- \* 停止设备指令(HDV)

- \* 通道地址字(CAW)

通道地址字是存放在内存固定单元的控制字，专门用于存放通道程序首地址



# 具有通道的I/O系统

## \* 处理步骤

- \* 组织通道程序，通道程序首地址送CAW
- \* 执行启动输入输出指令，启动通道工作
- \* 通道根据自身状态形成条件码；若通道可用，从CAW中取得通道程序首地址，得到第一条通道指令，启动设备控制器；设备控制器再检查设备状态；若设备不忙，则告知通道释放CPU
- \* 通道独立执行通道程序，控制外设进行输入输出
- \* 通道完成输入输出后，形成CSW和置中断字寄存器，发出输入输出中断，请求CPU处理



# I/O功能的演化

- \* 处理器直接控制外围设备
- \* 增加了控制器或者I/O模块
  - \* 处理器使用非中断的可编程I/O
  - \* 处理器开始从外围设备接口的具体细节中分离出来
- \* 采用中断方式的控制器或者I/O模块
  - \* 处理器无需花费等待执行一个I/O操作所需的时间，因而提高了效率



# I/O功能的演化

- \* I/O模块通过DMA直接控制存储器
  - \* 可以在没有处理器参与的情况下，从主存中移出或者往主存中移入一块数据，仅仅在传送开始和结束时需要用到处理器
- \* I/O模块被增强为单独的处理器，有专门为I/O设计的指令集
  - \* I/O处理器在没有CPU干涉的情况下取指令并执行这些指令，使得CPU可以指定一些列I/O活动，并只有当整个序列执行完成后CPU才被中断
- \* I/O模块有自己的局部存储器，其本身就是一台计算机



南京大學  
NANJING UNIVERSITY

## 9.2 I/O 缓冲



## 9.2 I/O 缓冲

### \* 使用缓冲的原因

- \* 进程经常需要等I/O完成才能继续后续的执行
- \* 在存储管理中使用分页机制，那么需要将包括目标地址单元的页锁定在主存中





# I/O 缓冲

- \* 面向块 (Block-oriented)

- \* 数据存储和传送以块为单位，块的大小通常是固定的
- \* 通常可以通过块号访问数据
- \* 例如：磁盘和磁带

- \* 面向流 (Stream-oriented)

- \* 设备以字节流的方式传输数据
- \* 例如：终端、打印机、通信端口、鼠标和其他指示设备以及大多数非辅存的其他设备



# 单缓冲

- \* 操作系统在主存中为I/O请求分配一个缓冲区
- \* 面向块 (Block-oriented)
  - \* 输入传送的数据被放到缓冲区中
  - \* 当传送完成时，进程把该块移动到用户空间，并立即请求另一块，这种方式成为超前(预先输入)



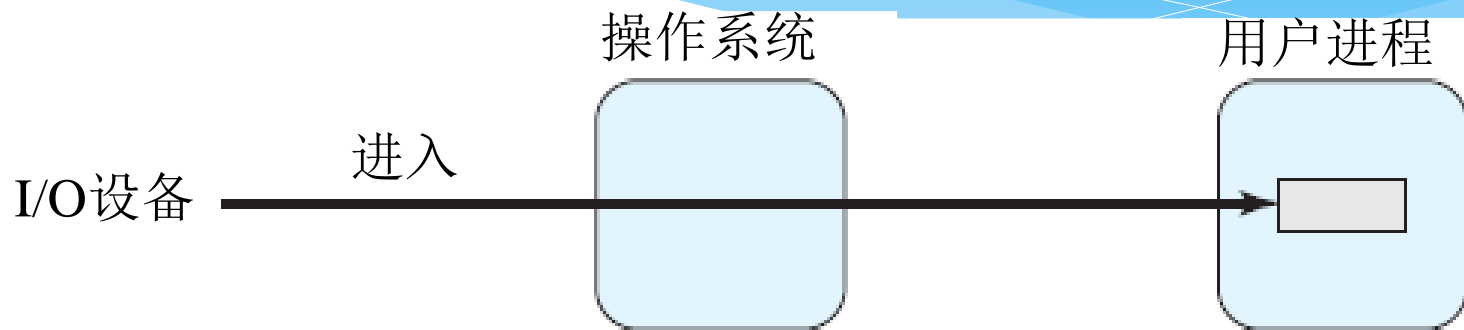
# 单缓冲

## \* 面向流 (Stream-oriented)

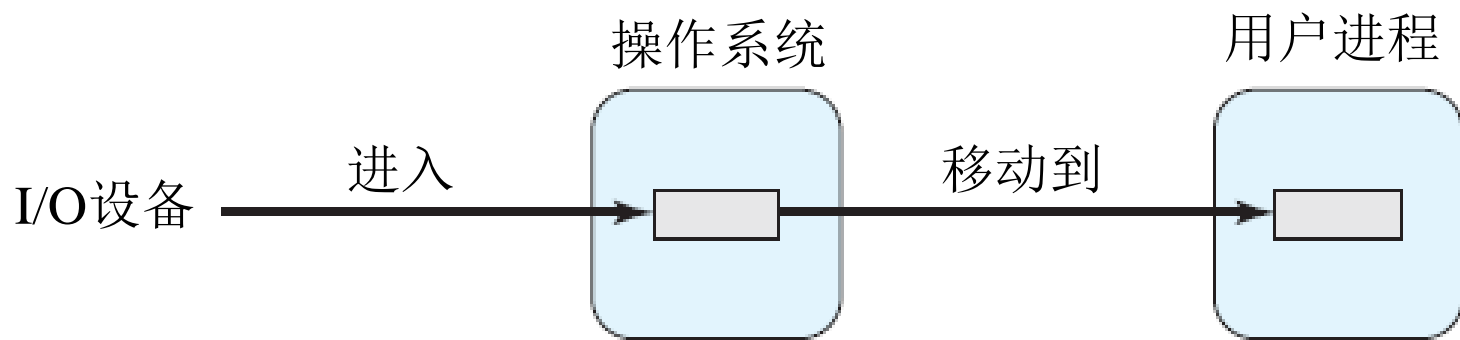
- \* 单缓冲方案可以以每次传送一行的方式或者每次传送一个字节的方式使用
- \* 每次传送一行的方式适合于滚动方式的终端(哑终端)
- \* 对于这类终端, 用户每次输入一行, 用回车符表示到达行尾, 并且输出到终端时也是类似的每次输出一行
- \* 行式打印机是这类设备的另一个例子
- \* 还有其他许多外设, 如传感器和控制器都属于这种类型



# I/O 缓冲



(a) 没有缓冲

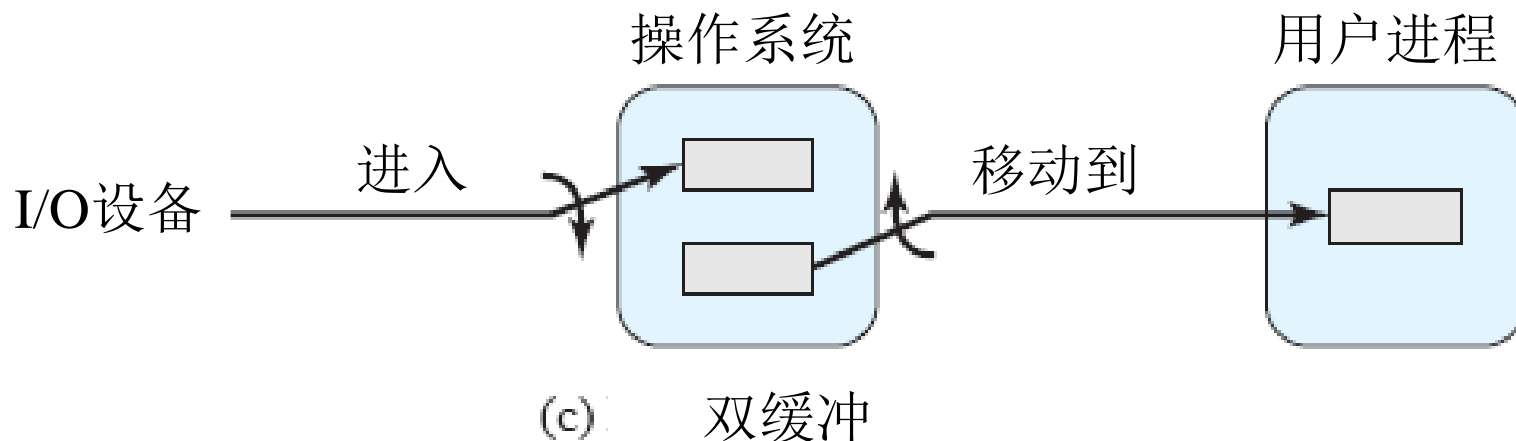


(b) 单缓冲



# 双缓冲

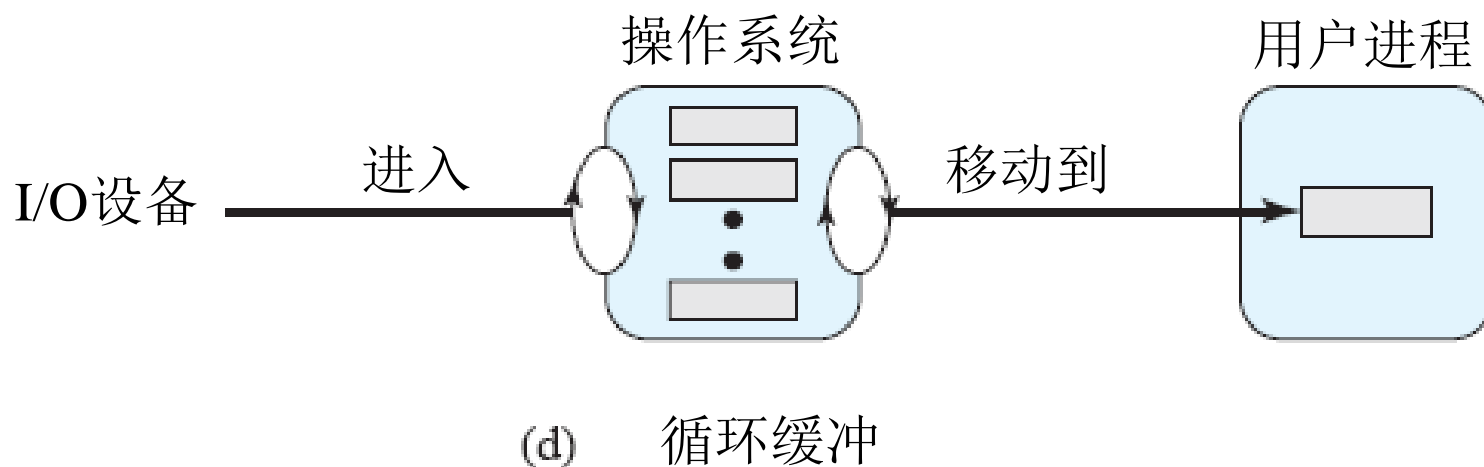
- \* 最多两个缓冲
- \* 在一个进程往一个缓冲区中传送数据(从这个缓冲区取数据)的同时，操作系统正在清空(或者填充)另一个缓冲区





# 循环缓冲

- \* 使用两个以上的缓冲区
- \* 使得I/O操作能跟得上进程的执行





南京大學  
NANJING UNIVERSITY

## 9.3 独占型外围设备的分配



# 设备分配

- \* 从设备的特性来看，可以把设备分成独占设备、共享设备和虚拟设备三类：
- \* 相应的管理和分配外围设备的技术可分成：独占方式、共享方式和虚拟方式





南京大学  
NANJING UNIVERSITY

# 独占型外围设备的分配

## \* 设备独立性

用户作业通过设备的逻辑号使用设备，由设备管理来建立逻辑设备与具体物理设备之间的对应关系



南京大学  
NANJING UNIVERSITY

# 独占型外围设备的分配

## \* 设备的分配

- \* 静态分配方式

- \* 动态分配方式

- \* 设备类表：设备类，总台数，空闲台数，设备表起始地址

- \* 设备表：物理设备名，逻辑设备名，占有进程号，是否分配，好/坏



# 独占型外围设备的分配

## \* 设备分配算法

- 1) 根据设备类查设备类表
- 2) 若无空闲设备则等待设备
- 3) 取得设备表首地址
- 4) 查设备表发现空闲设备栏
- 5) 若未发现空闲设备栏则等待设备
- 6) 填占用进程号, 逻辑设备名, 分配设备



南京大學  
NANJING UNIVERSITY

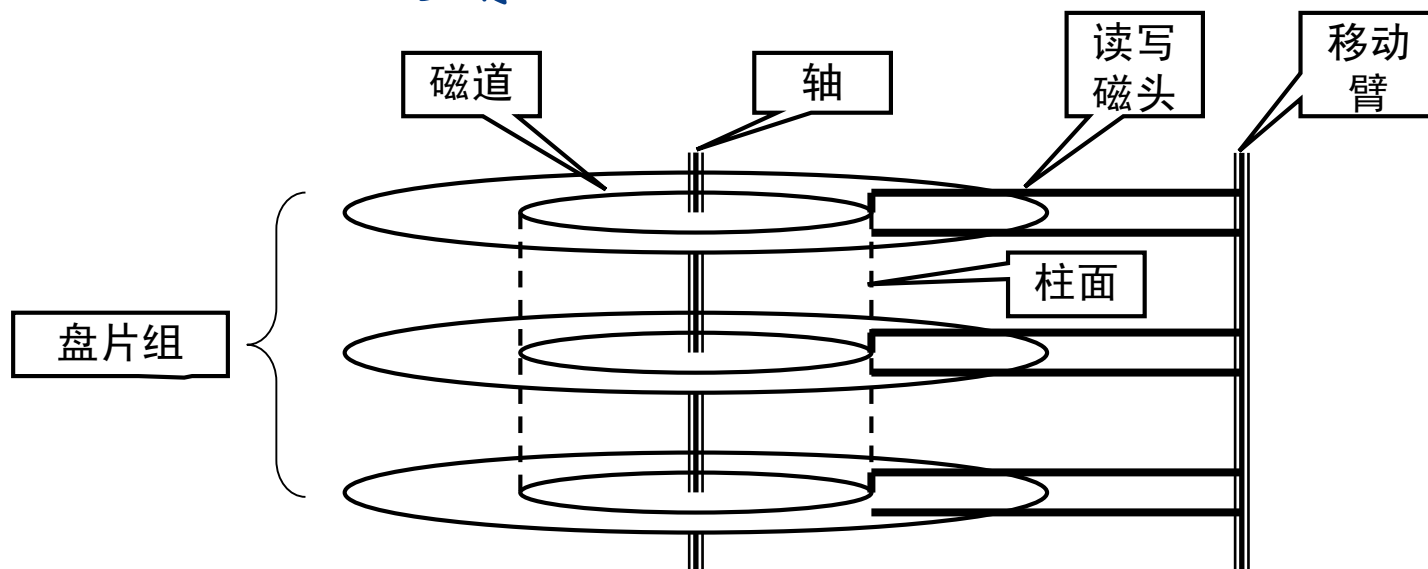
## 9.4 共享型设备的驱动调度



# 磁盘

## \* 磁盘结构

- \* 一个磁道分为固定多个扇区
- \* 磁盘物理块地址: (柱面号, 磁头号, 扇区号)
- \* 磁盘读写方式: 移臂 → 旋转 → 读写





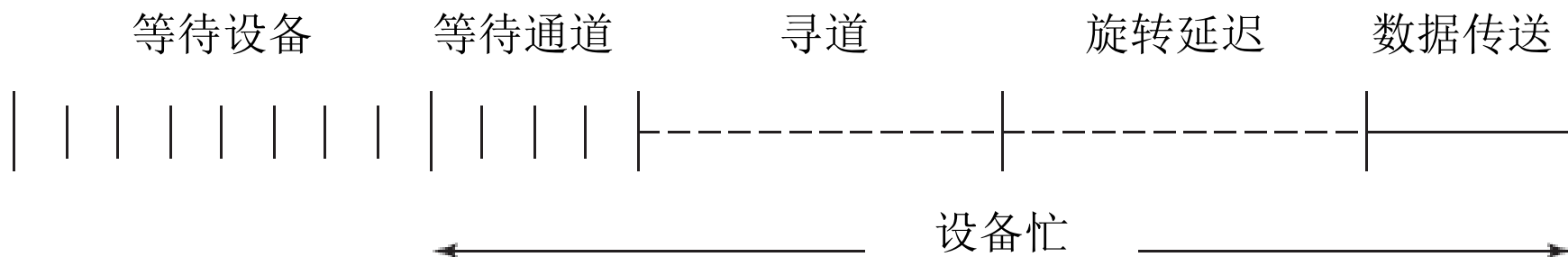
# 磁盘性能参数

- \* 为了读或者写，磁头必须定位到指定的磁道(track)和该磁道中指定的扇区(sector)的开始处
- \* 寻道时间 (记为 $T_s$ )
  - \* 将磁盘磁头臂移动指定磁道所需要的时间
- \* 旋转延迟 (Rotational delay or rotational latency)
  - \* 要读写的扇区到达磁头的旋转时间
- \* 传送时间



南京大学  
NANJING UNIVERSITY

# 磁盘I/O传送的时序



磁盘I/O传送的时序



# 磁盘性能参数

- \* 当要读写的扇区(sector)移动到磁头下方, 开始数据传输
- \* 存取时间 (Access time) (记为  $T_a$ )
  - \* 达到读或者写的位置所需要的时间
  - \* 是寻道时间、旋转延迟、传送时间的总和
  - \*  $b$  表示要传送的字节数,  $N$  表示一个磁道中的字节数,  $r$  表示旋转速度(单位为转/秒)

$$T_a = T_s + \frac{1}{2r} + \frac{b}{rN}$$





南京大学  
NANJING UNIVERSITY

# 磁盘调度策略

- \* 寻道时间是影响性能差异的重要原因
- \* 一个磁盘可能接收到一组I/O请求
- \* 如果随机选择并响应这些I/O请求，可能得到最坏的性能



南京大学  
NANJING UNIVERSITY

# 磁盘调度策略

- \* 先进先出 First-in, first-out (FIFO)
  - \* 按顺序处理请求
  - \* 对于所有进程是公平的



# 磁盘调度策略

## \* 优先级 (Priority)

- \* 这种方法并不会优化磁盘的使用率，但可以满足操作系统的其他目标
- \* 通常比较短的批作业和交互作业的优先级较高，而较长计算时间的长作业优先级低，这就使得大量短作业能够迅速地通过系统，并且可以提供比较好的交互响应时间
- \* 但是长作业不得不等待很长的时间



# 磁盘调度策略

## \* 后进先出 (Last-in, first-out)

- \* 在事务处理系统中，把设备资源提供给最近的用户，会导致磁头臂在一个顺序文件中移动时移动得很少，甚至不移动
- \* 利用这种局部性可以提高吞吐量，减少队列长度
- \* 只要一个作业积极地使用文件系统，它就可以尽可能快地得到处理
- \* 如果由于工作量大而磁盘保持忙状态，就有可能出现饿死的情况
- \* 当一个作业已经往队列中送入一个I/O请求，并且错过了可以提供服务的位置时，该作业就有可能永远得不到服务，除非它之前的队列变为空



# 磁盘调度策略

- \* **最短服务时间优先(SSTF, Shortest Service Time First)**
  - \* 选择使磁头臂从当前位置开始移动最少的磁盘I/O请求，因此SSTF策略总是选择导致最小寻道时间的请求
  - \* 总是选择最小寻道时间并不能保证平均寻道时间最小，但是，它的性能比FIFO更好



# 磁盘调度策略

## \* 扫描 (SCAN)

- \* 要求磁头臂仅仅沿一个方向移动，并在途中满足所有为完成的请求，直到它到达这个方向上的最后一个磁道，或者在这个方向上没有其他请求为止，后一种改进有时候称为LOOK策略
- \* 接着反转服务方向，沿着相反方向扫描，同样按顺序完成所有请求



# 磁盘调度策略

## \* 循环扫描 (C-SCAN)

- \* 把扫描限定在一个方向
- \* 当访问到沿某个方向的最后一个磁道时，磁头臂返回到磁盘相反方向磁道的末端，并再次开始扫描



# 磁盘调度策略

## \* N-step-SCAN

- \* 把磁盘请求队列分成长度为 $N$ 的子队列，每一次用SCAN处理一个子队列
- \* 在处理一个队列时，新请求必须添加到其他某个队列总
- \* 如果在扫描的最后剩下的请求数小于 $N$ ，则它们全部将在下一次扫描时处理
- \* 当 $N$ 很大时，N-step-SCAN的性能接近SCAN；当 $N=1$ 时，实际上是FIFO





# 磁盘调度策略

## \* FSCAN

- \* 使用两个子队列
- \* 当开始扫描时，所有请求都在一个队列中，而另一个队列为空
- \* 在扫描过程中，所有新到的请求都被放入另一个队列中
- \* 因此，对新请求的服务延迟到处理完成所有老请求之后



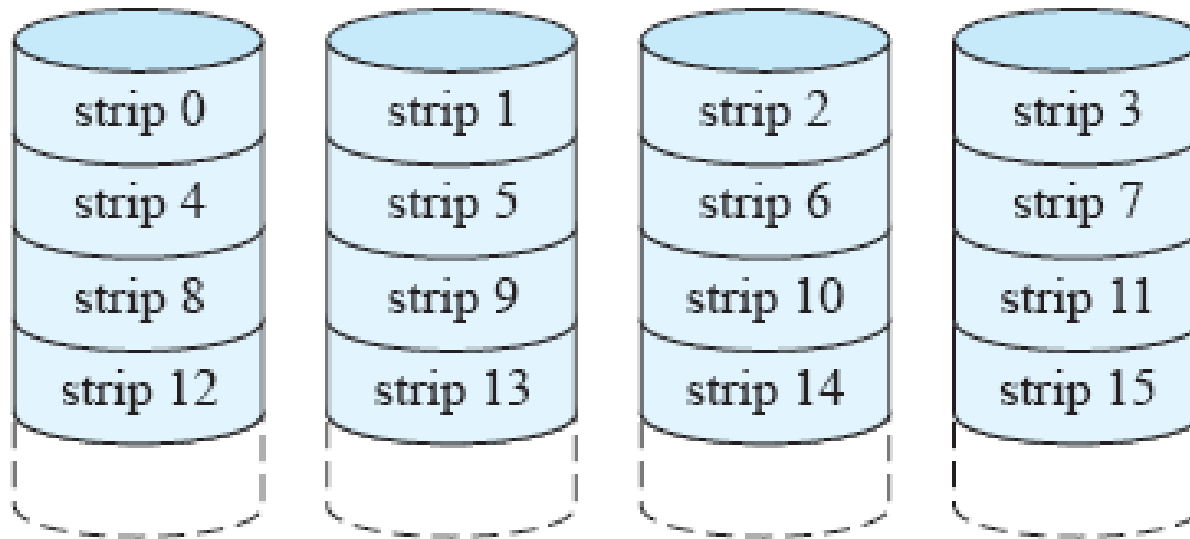
南京大学  
NANJING UNIVERSITY

# 旋转调度 (Rotation Schedule)

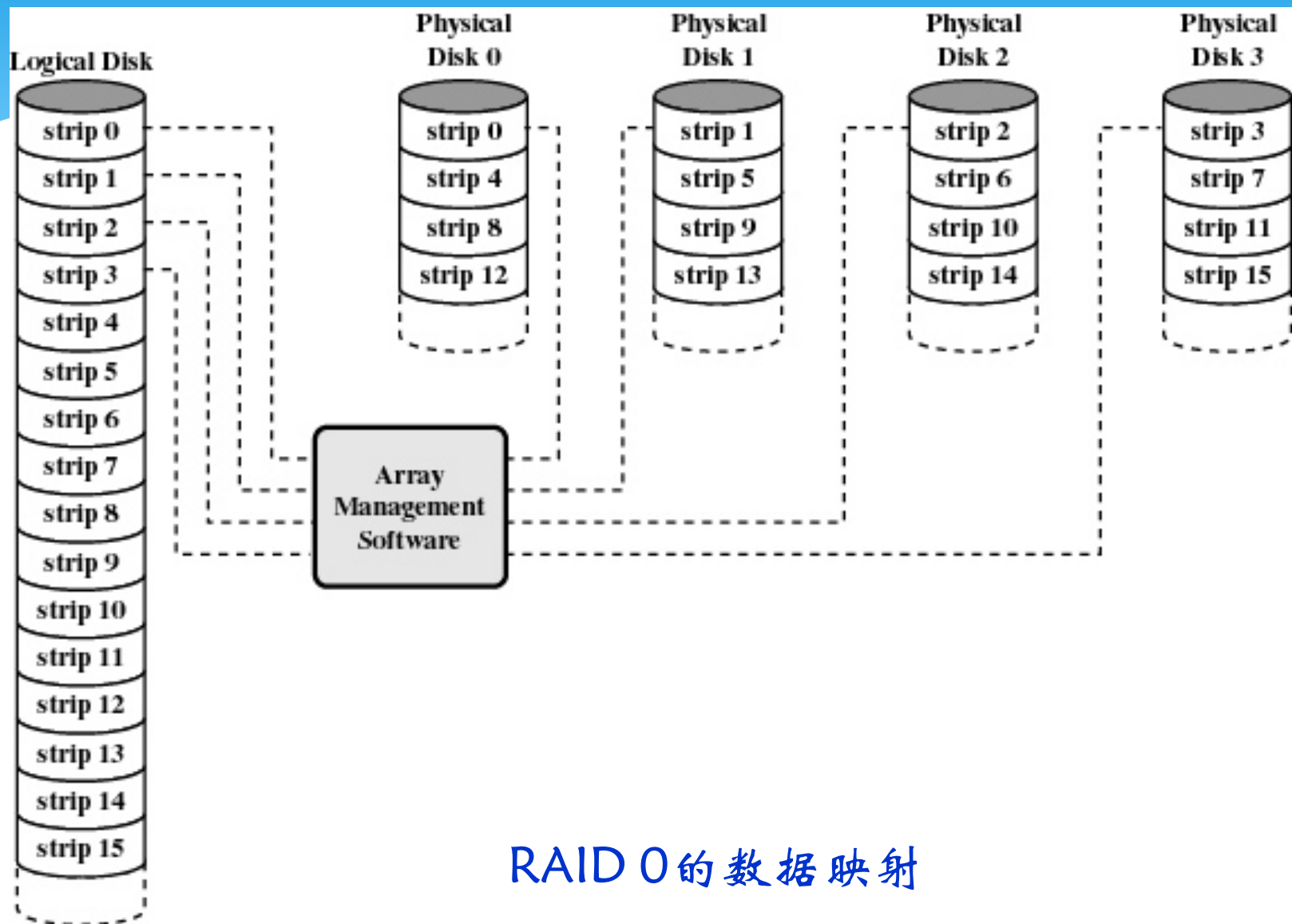
- \* 循环排序：在最少旋转圈数内完成位于同一柱面的访问请求；旋转位置测定硬件和多磁头同时读写技术有利于提高旋转调度的效率
- \* 优化分布
- \* 交替地址



# RAID 0 (无冗余)



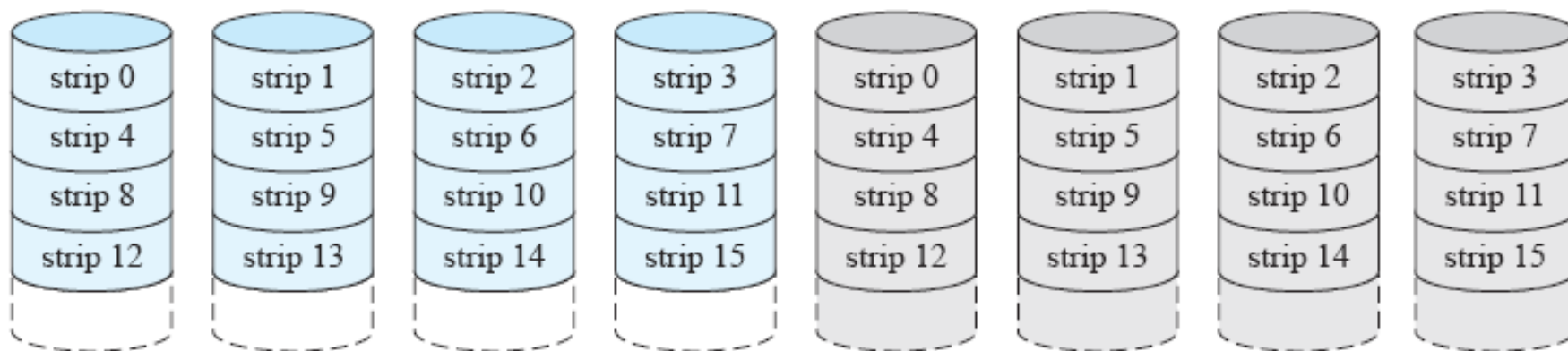
(a) RAID 0 (无冗余)



RAID 0的数据映射



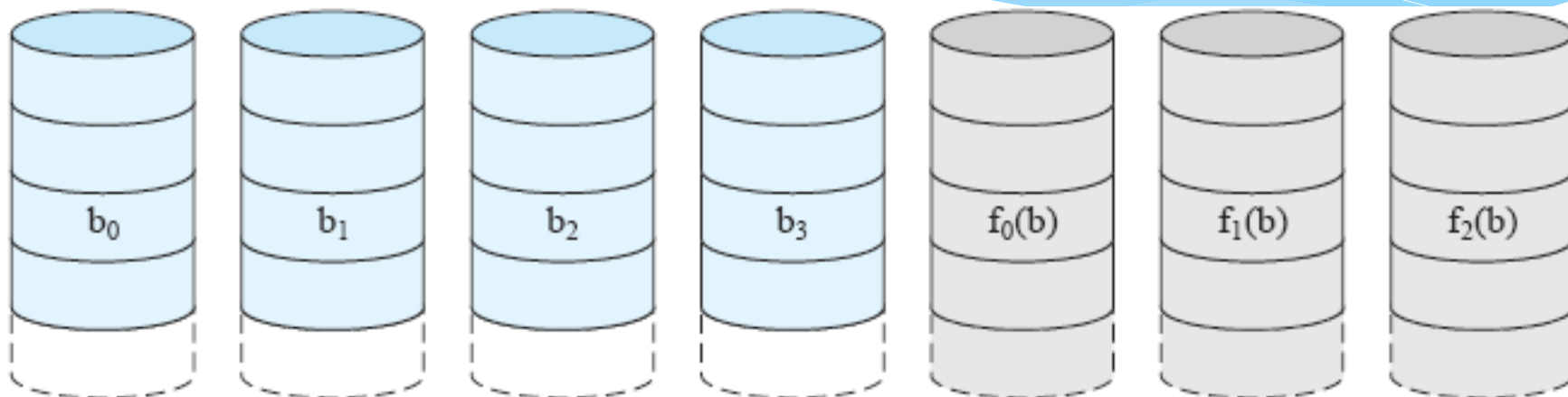
# RAID 1 (鏡像)



(b) RAID 1 (鏡像)



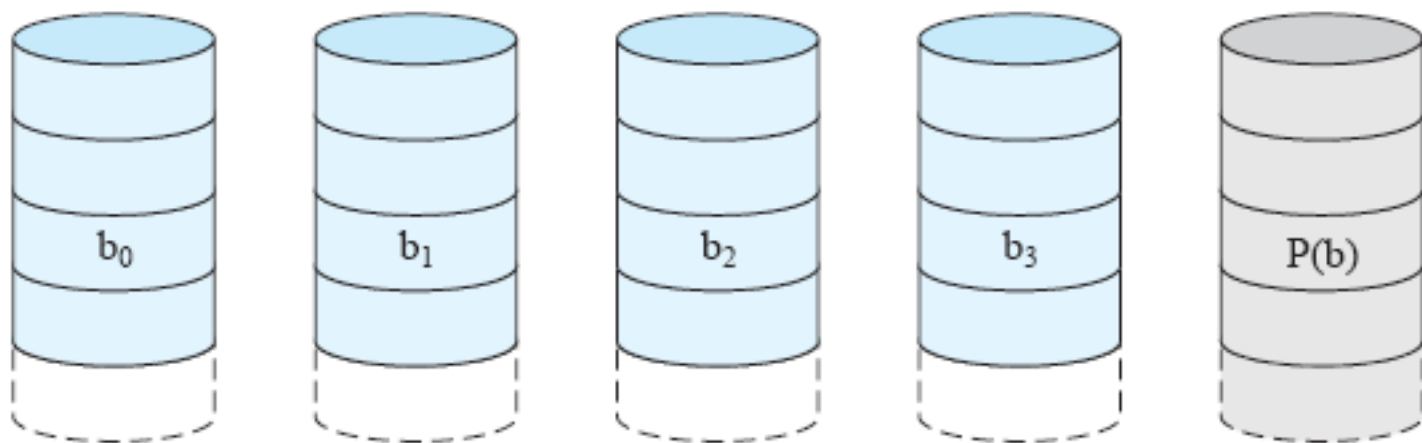
# RAID 2 (通过海明码冗余)



(c) RAID 2 (通过海明码冗余)



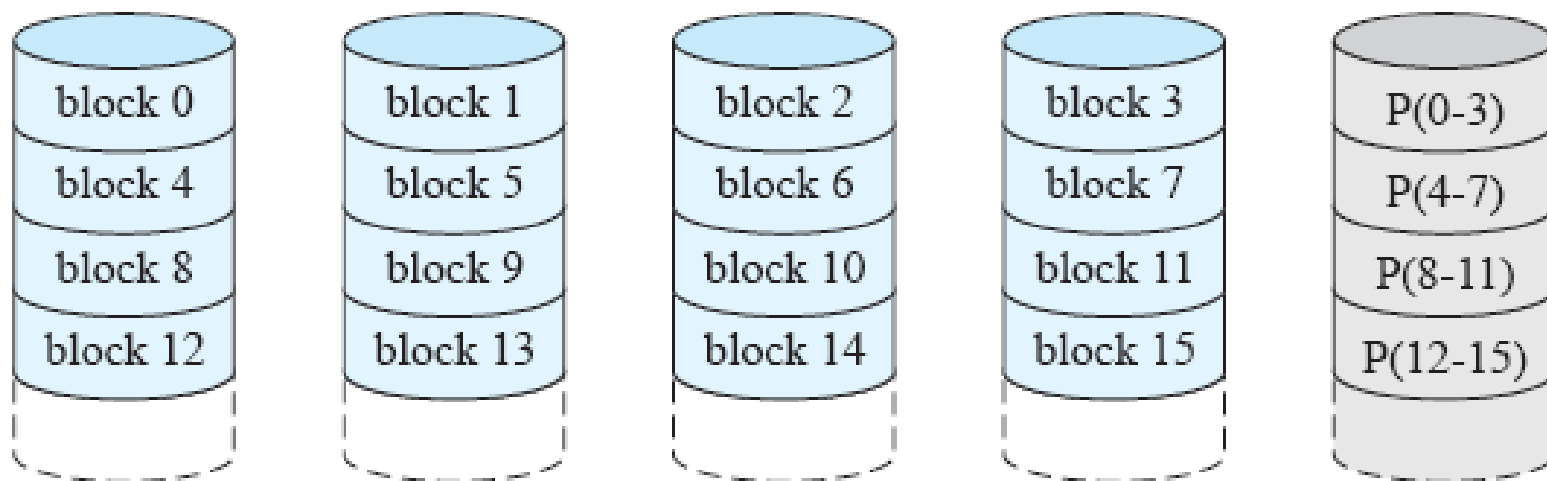
# RAID 3 (交錯位奇偶校驗)



(d) RAID 3 (交錯位奇偶校驗)



# RAID 4 (块奇偶校验)

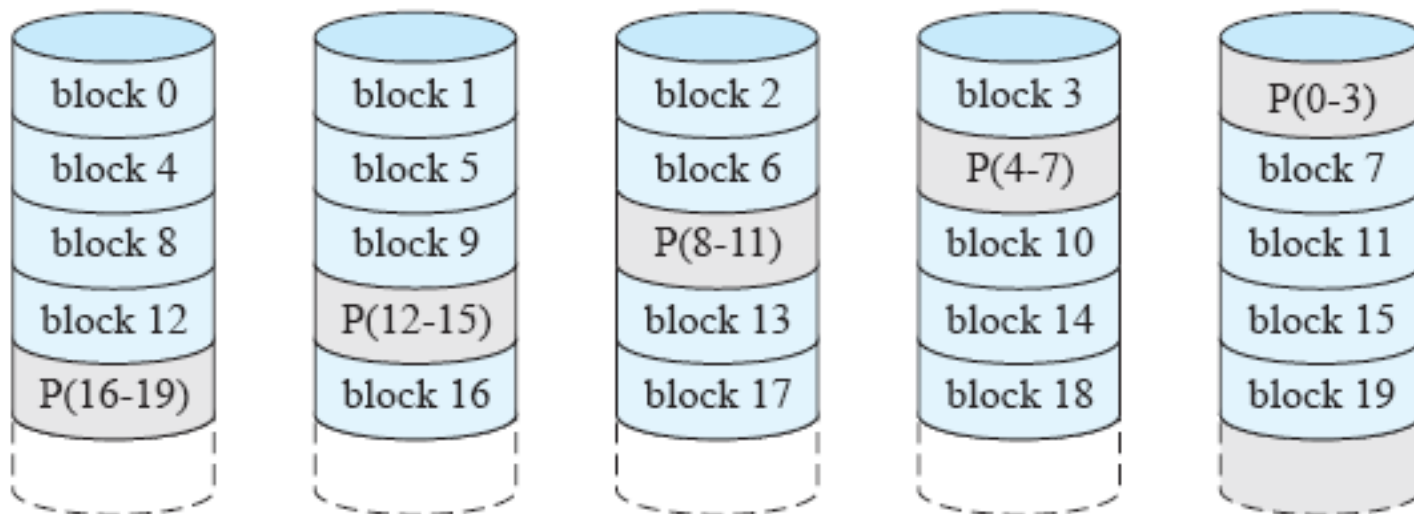


(e) RAID 4 (块奇偶校验)





# RAID 5 (块分布奇偶校验)



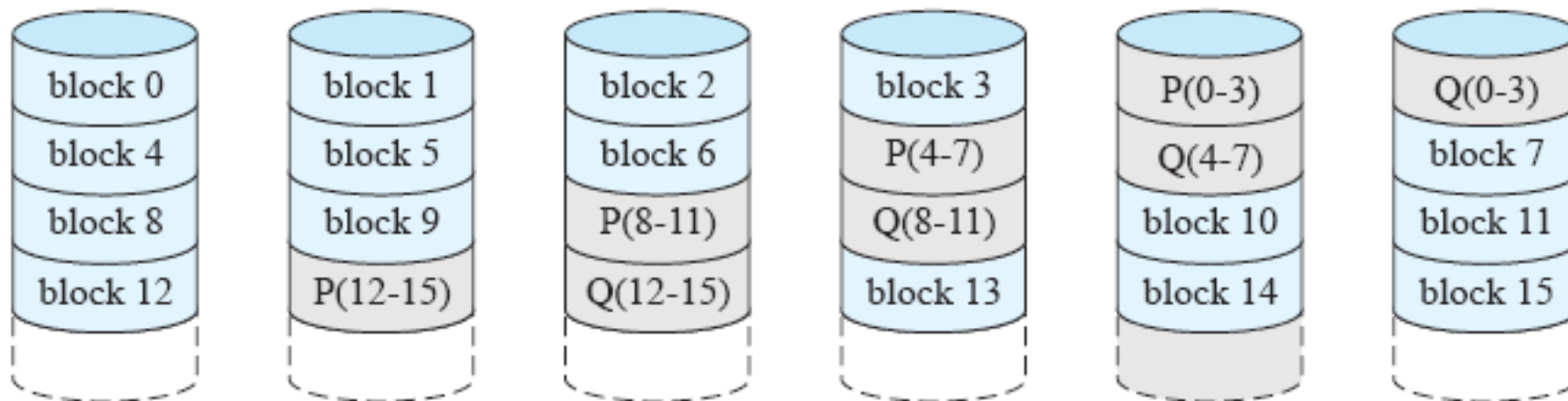
(e) RAID 5 (块分布奇偶校验)



南京大學

NANJING UNIVERSITY

# RAID 6 (双重冗余)



(f) RAID 6 (双重冗余)



南京大學  
NANJING UNIVERSITY

# 磁盘 Cache

- \* 磁盘高速缓存是主存中为磁盘扇区设置的一个缓冲区
- \* 包含磁盘中某些扇区的副本
- \* 利用局部性原理，可以减少平均存储器存取时间



# 替换策略.LRU

- \* 替换在高速缓存中未被访问的时间最长的块
- \* 逻辑上，高速缓存有一个关于块的栈组成，最近访问过的块在栈顶，当高速缓存中的一个块被访问到时，它从栈中当前的位置移到栈顶
- \* 当一个块从辅存中取入时，把位于栈顶的那一块移出，并把新到来的块压入栈顶
- \* 并不需要在主存中真正移动这些块，有一个栈指针与高速缓存相关联



# 替换策略.LFU

## (Least Frequently Used)

- \* 替换集合中被访问次数最少的块
- \* LFU可以通过给每个块关联一个计数器来实现
- \* 当一个块被读入时，它的计数器被指定为1；当每次访问到这一块时，它的计数器增1
- \* 当需要替换时，选择计数器值最小的块
- \* 直觉上，LFU比LRU更适合，因为LFU使用了关于每个块的更多的相关信息



南京大學  
NANJING UNIVERSITY

## 9.5 虚拟设备



## 9.5 虚拟设备

- \* 虚拟设备
  - \* 使用一类物理设备模拟另一类物理设备的技术
  - \* 通常是使用共享型外围设备模拟独占型外围设备
- \* 脱机同时外围设备操作



# SPOOLing

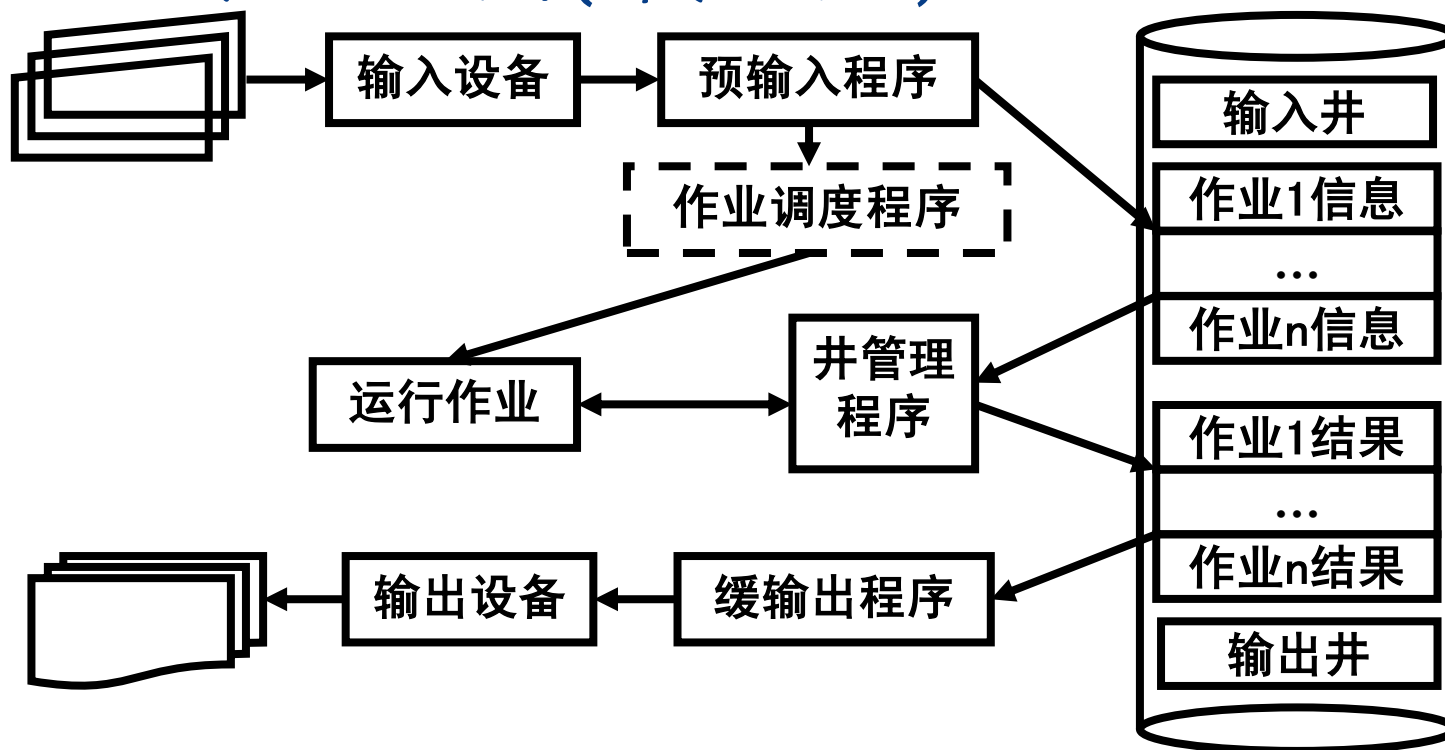
- \* “井”是用作缓冲的存储区域，采用井的技术能调节供求之间的矛盾，消除人工干预带来的损失。
- \* “预输入程序”
  - \* 操作系统将一批作业从输入设备上预先输入到磁盘的输入缓冲区中暂时保存，这称为“预输入”，此后，由作业调度程序调度作业执行，作业使用数据时不必再启动输入设备，只要从磁盘的输入缓冲区中读入
- \* “缓输出程序”
  - \* 作业执行中不必直接启动输出设备，只要将作业的输出数据暂时保存到磁盘的输出缓冲区，当作业执行完毕后，由操作系统组织信息成批输出
- \* “井管理程序”





# SPOOLing

\* 联机同时外围设备操作(斯普林系统)

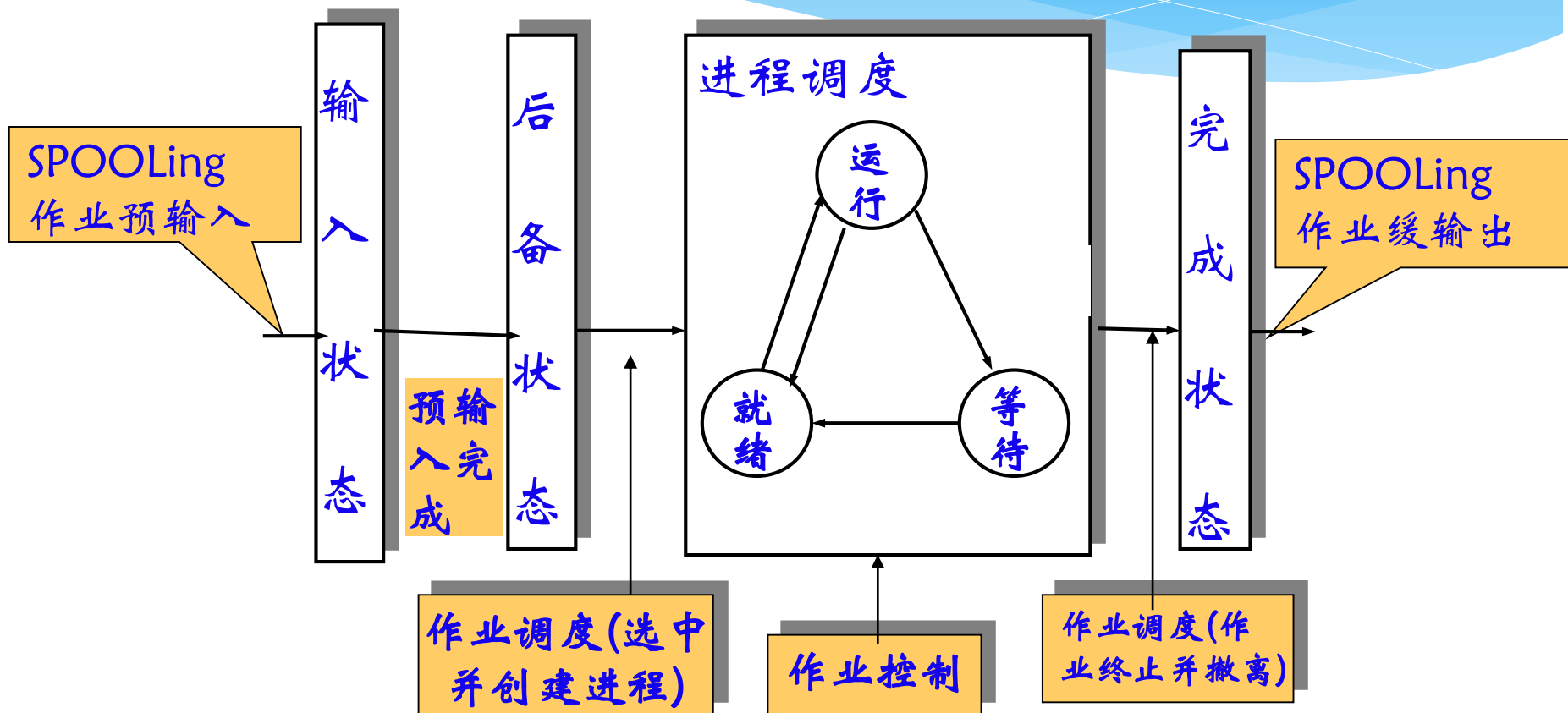




南京大学

NANJING UNIVERSITY

# 作业调度与进程调度的关系





南京大學  
NANJING UNIVERSITY

## 9.6 I/O设备管理的实现与层次



# I/O 软件

## \* I/O 软件的设计目标

把软件组织成一种层次结构，低层软件用来屏蔽硬件的具体细节，高层软件则主要向用户提供一个简洁、规范的界面

## \* I/O 软件要解决的主要问题：

- \* 设备无关性
- \* 出错处理
- \* 同步(阻塞)——异步(中断驱动)传输
- \* 独占型外围设备和共享型外围设备



# I/O软件的层次

## \* I/O软件的层次

- \* 用户层I/O软件
- \* 与设备无关的操作系统I/O软件
- \* 设备驱动程序
- \* I/O中断处理程序(底层)

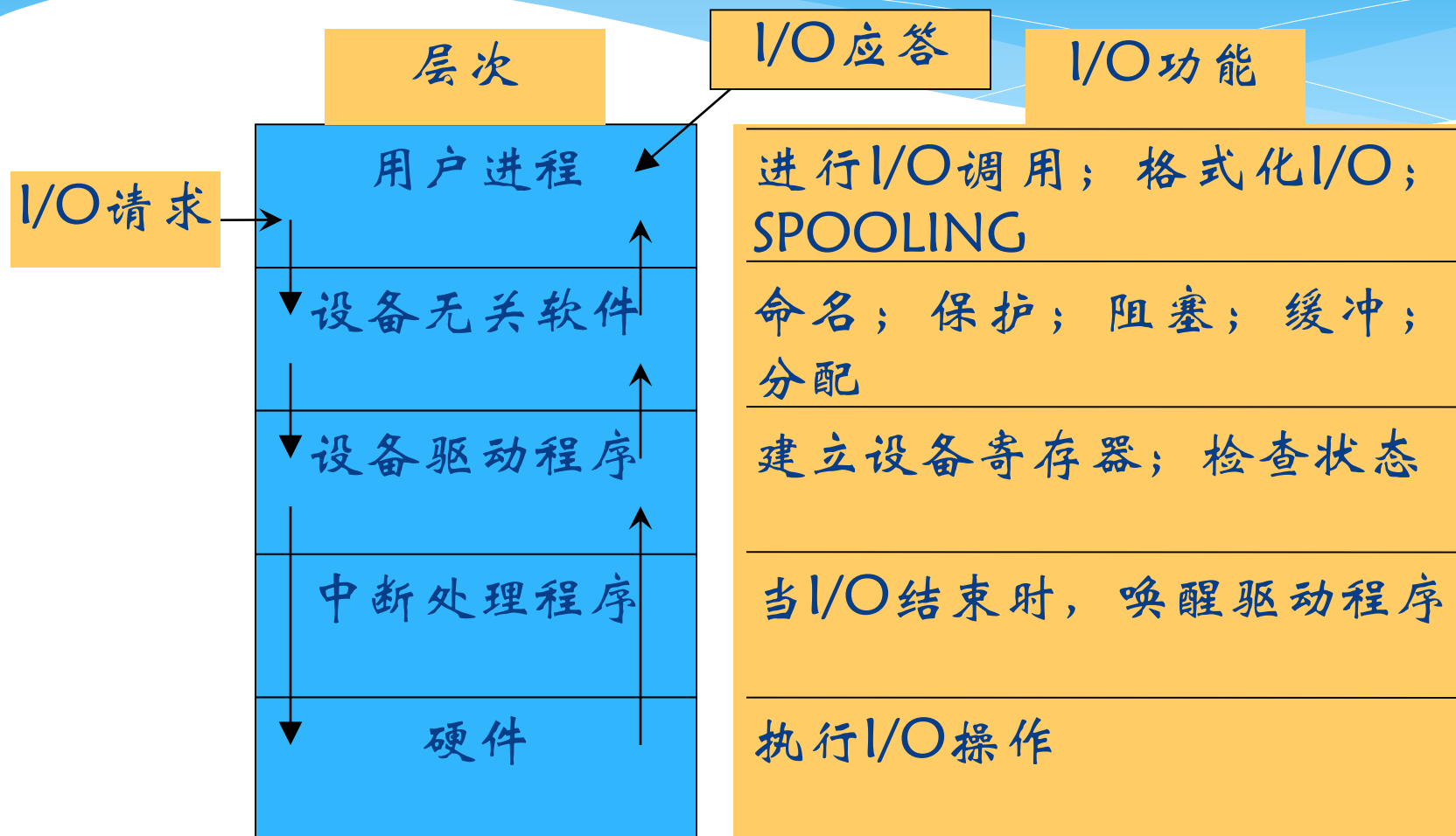
高



低



# I/O系统各层软件及其功能





南京大學  
NANJING UNIVERSITY

# I/O 系統各層軟件及其功能

## \* I/O 中斷處理程序

- \* 通知用戶程序 I/O 操作沿鏈推進的程度
- \* 通知用戶程序輸入輸出操作正常結束
- \* 通知用戶程序發現的輸入輸出操作異常
- \* 通知程序外圍設備上重要的异步信號



南京大学  
NANJING UNIVERSITY

# I/O系统各层软件及其功能

## \* 设备驱动程序

- \* 设备驱动程序中包括了所有与设备相关的代码
- \* 每个设备驱动程序只处理一种设备，或者一类紧密相关的设备





# I/O系统各层软件及其功能

- \* 与设备无关的I/O软件
- \* 执行适用于所有设备的常用I/O功能，并向用户层软件提供一个一致的接口
  - \* 对设备驱动程序的统一接口
  - \* 设备命名
  - \* 设备保护
  - \* 提供独立于设备的块大小
  - \* 缓冲区管理
  - \* 块设备的存储分配
  - \* 独占型外围设备的分配和释放
  - \* 错误报告



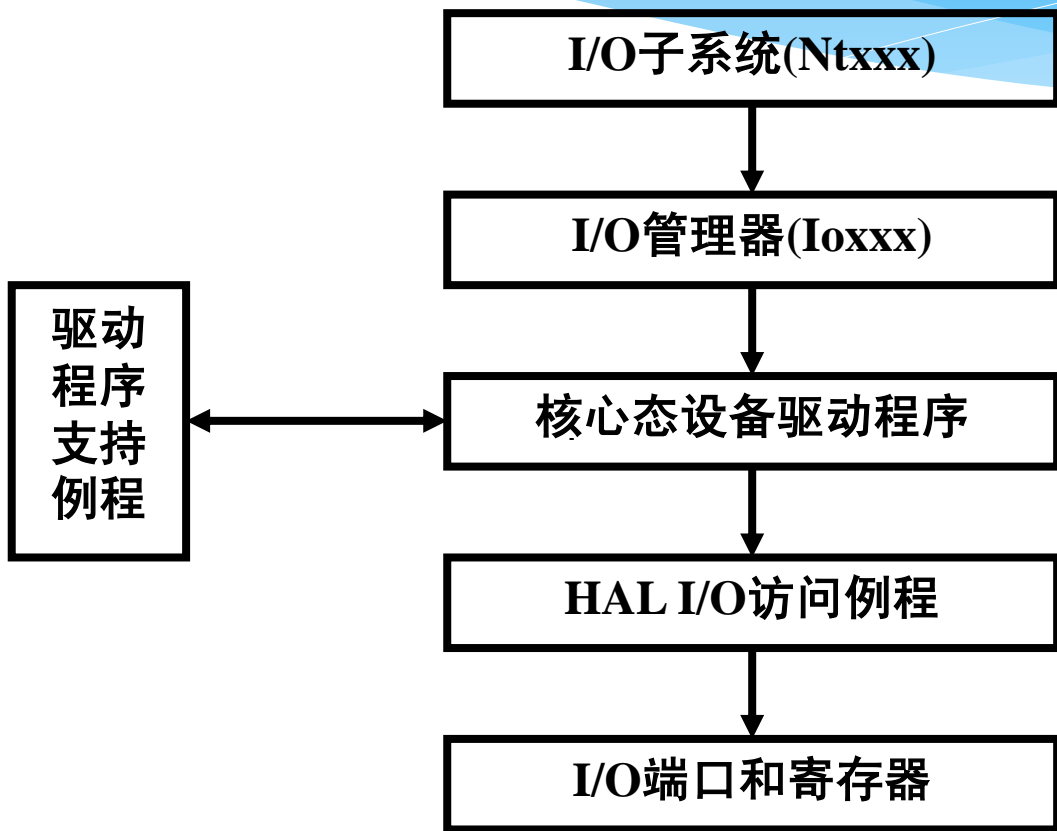
南京大學  
NANJING UNIVERSITY

# I/O 系統各層軟件及其功能

- \* 用戶層I/O軟件
  - \* I/O庫例程構
  - \* SPOOLing系統
  - \* 守護進程(daemon)



# Windows 的 I/O 管理





南京大學

NANJING UNIVERSITY

# Windows 的 I/O 管理

## **I/O Manager**

**Cache  
Manager**

**File System  
Drivers**

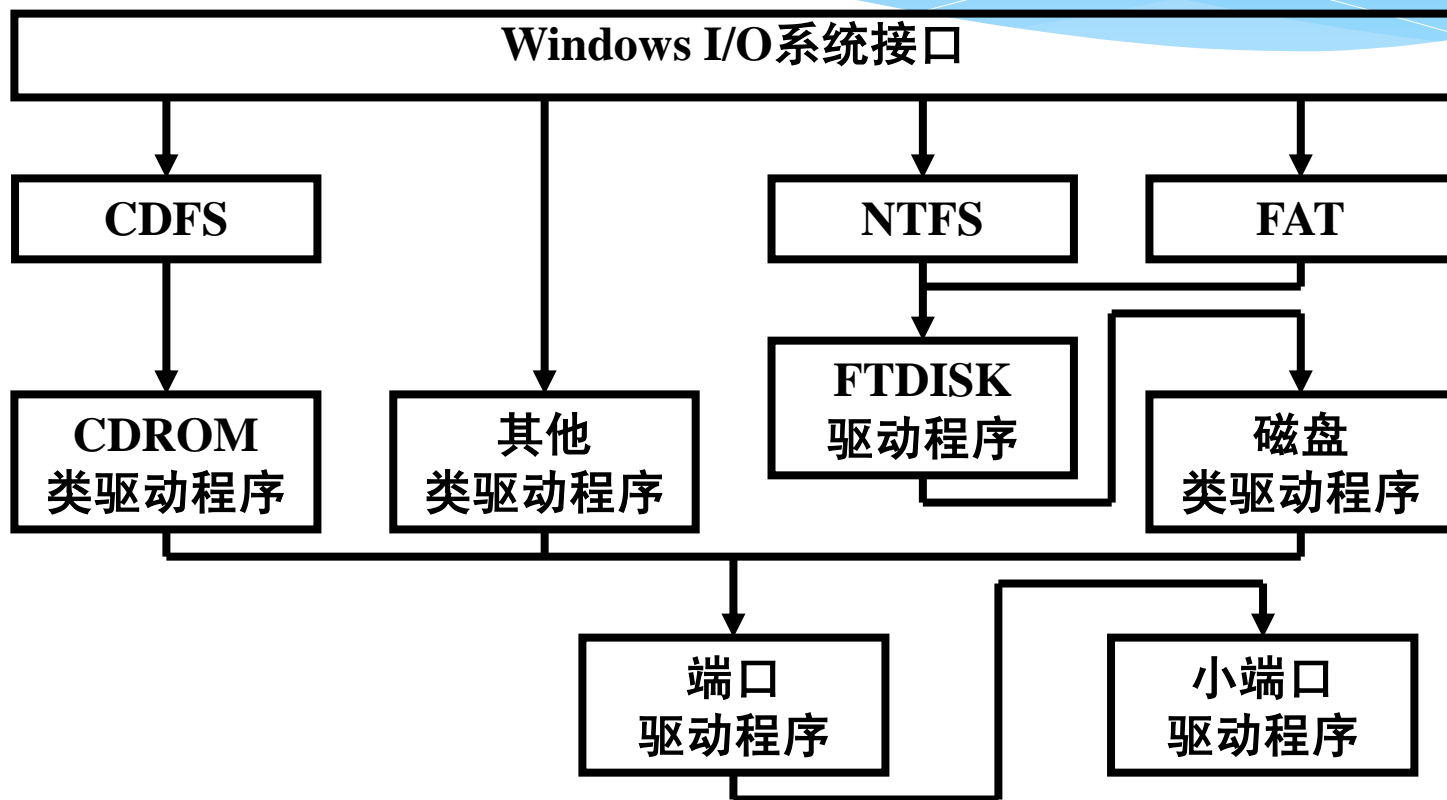
**Network  
Drivers**

**Hardware  
Device Drivers**



# Windows的设备管理

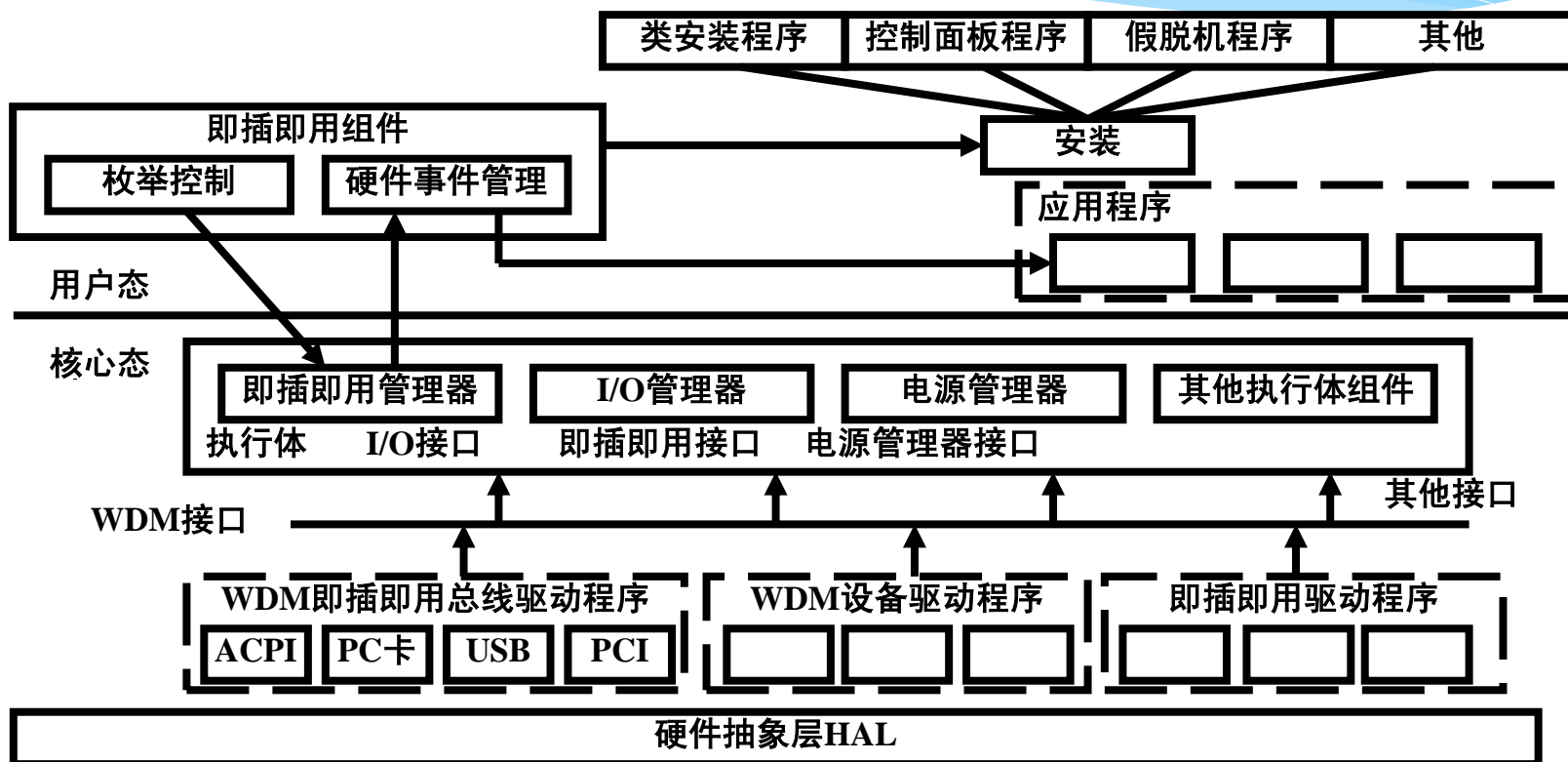
## \* 设备驱动程序





# Windows设备管理的扩展

## \* 即插即用结构





南京大學  
NANJING UNIVERSITY

# 本主题教学目标

1. 复习、了解I/O设备、I/O控制方式
2. 掌握I/O缓冲区的设计
3. 掌握磁盘调度
4. 掌握I/O软件系统的设计与实现
5. 掌握虚拟设备