

《操作系统》



主讲教师：翟高寿

联系电话：010-51684177 (办)

电子邮件：gszhai@bjtu.edu.cn

制作人：翟高寿

制作单位：北京交通大学计算机学院

第五章 设备管理

5.1 I/O系统组成

5.2 I/O控制方式

5.3 设备管理目标、功能及结构

5.4 缓冲管理

5.5 设备分配

5.6 设备处理

5.7 磁盘存储器管理

5.1 I/O系统组成

5.1.1 I/O系统结构

5.1.2 I/O设备分类及接口

5.1.3 设备控制器

5.1.4 I/O通道

微机总线型I/O系统结构

磁盘驱动器



打印机



其它设备



Extended Industry Standard Architecture

扩展版工业标准体系结构

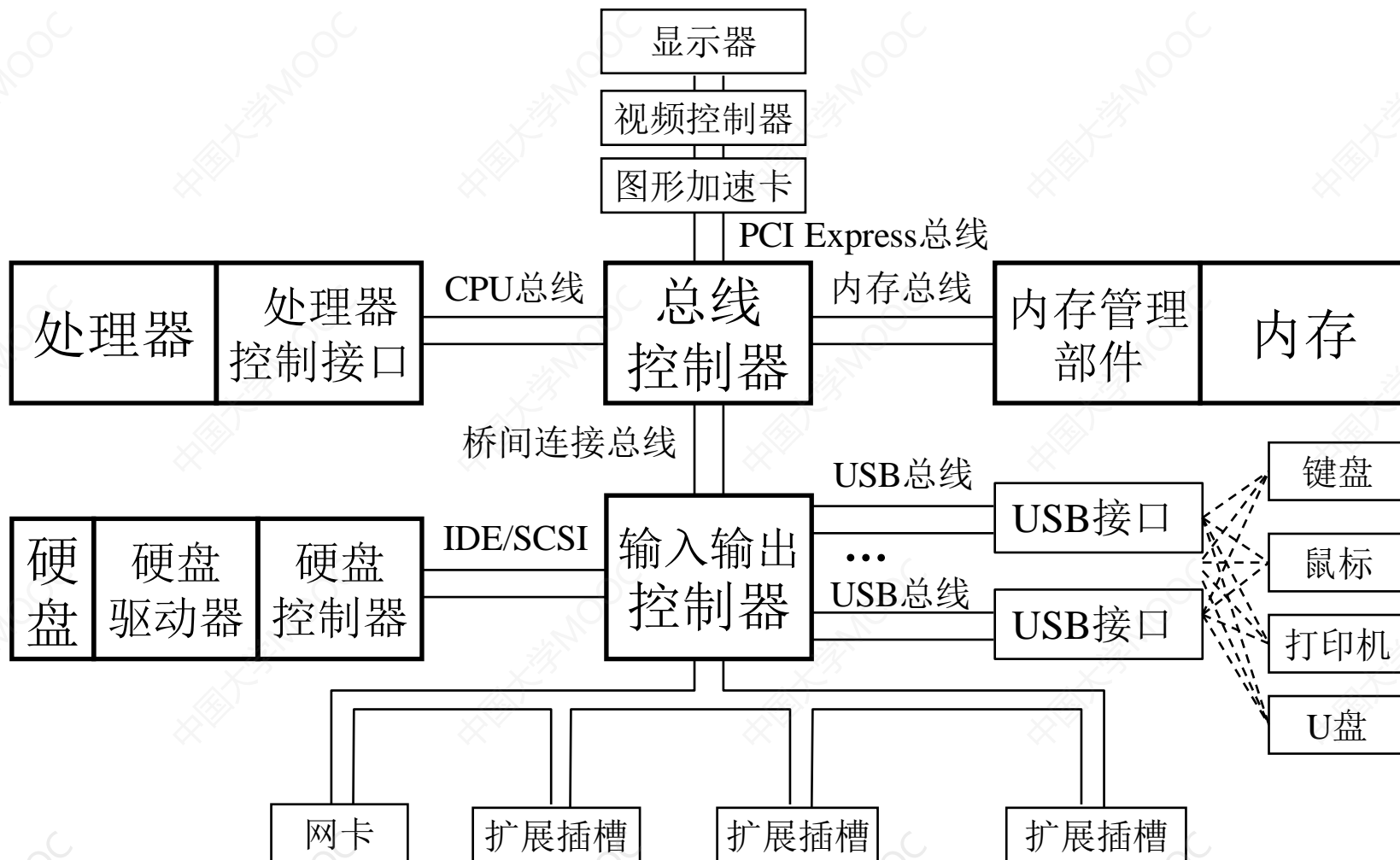
Peripheral Accelerated Graphics Port

图形快速处理端口

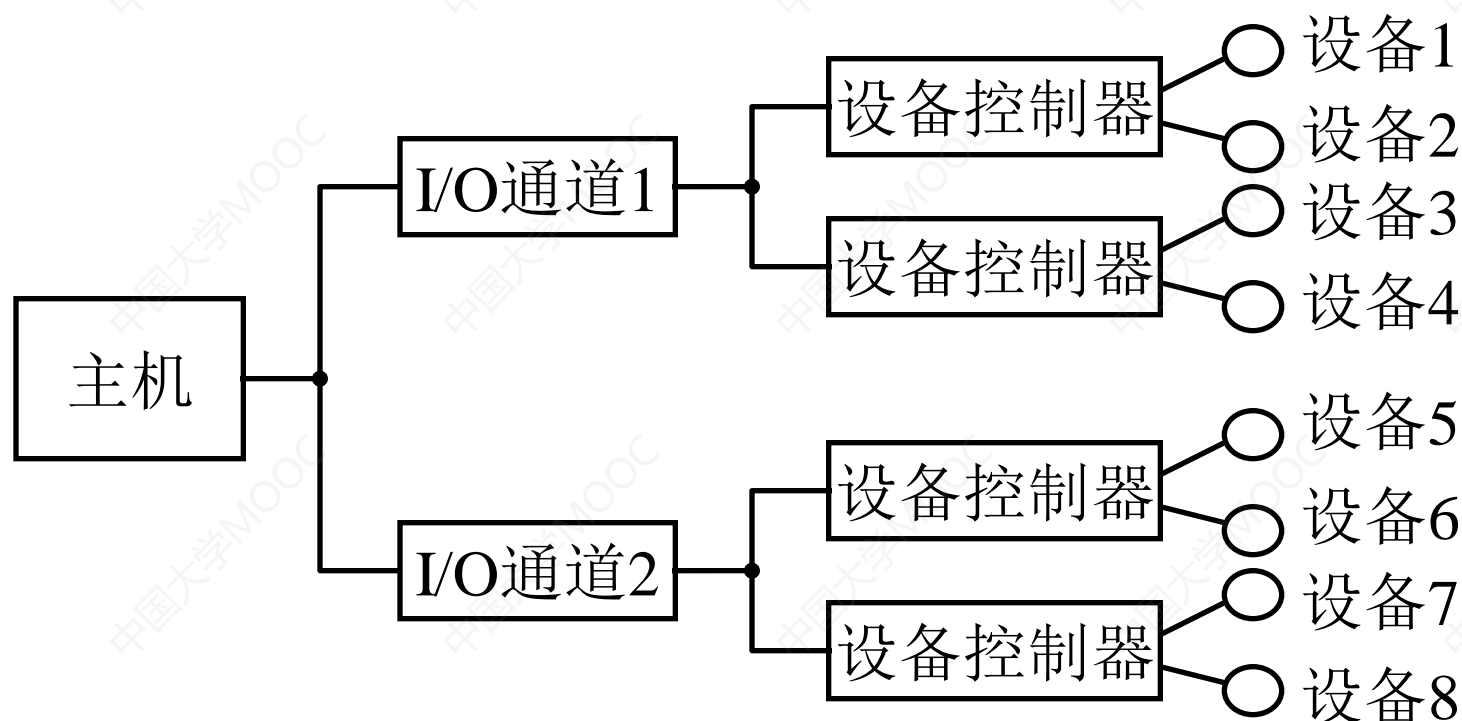
➤ VESA总线

➤ PCI总线、AGP总线、PCI-Express总线

微机总线型I/O系统结构 (南北桥)



主机通道型I/O系统结构



5.1 I/O系统组成

5.1.1 I/O系统结构

5.1.2 I/O设备分类及接口

5.1.3 设备控制器

5.1.4 I/O通道

I/O设备分类

□ 按数据传输速率分类

- 高速设备、中速设备、低速设备

□ 按信息交换单位分类

网络设备

- 块设备（传输速率高、可寻址、DMA）
- 字符设备（传输速率低、不可寻址、中断）

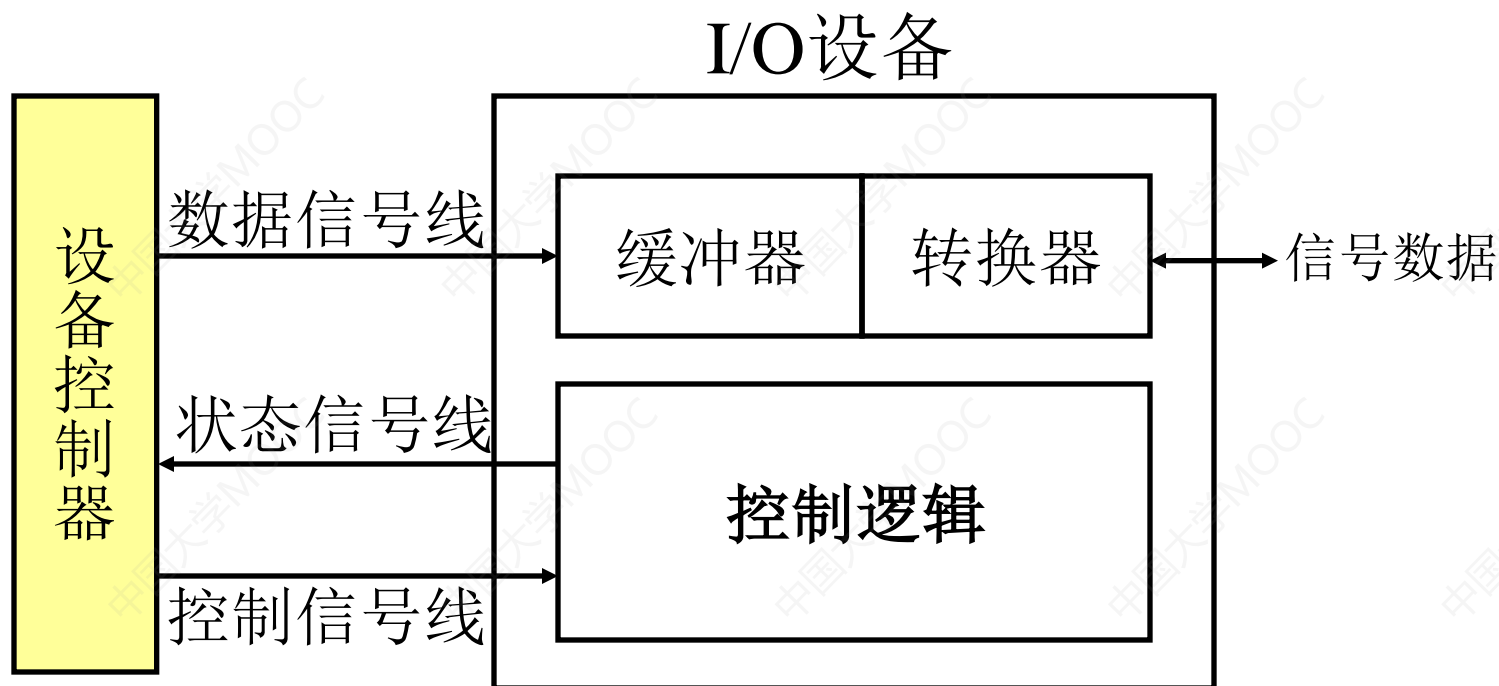
□ 按设备共享属性分类

Direct Memory Access

内存直接存取

- 独占设备
- 共享设备
- 虚拟设备

设备与设备控制器之间的接口



5.1 I/O系统组成

5.1.1 I/O系统结构

5.1.2 I/O设备分类及接口

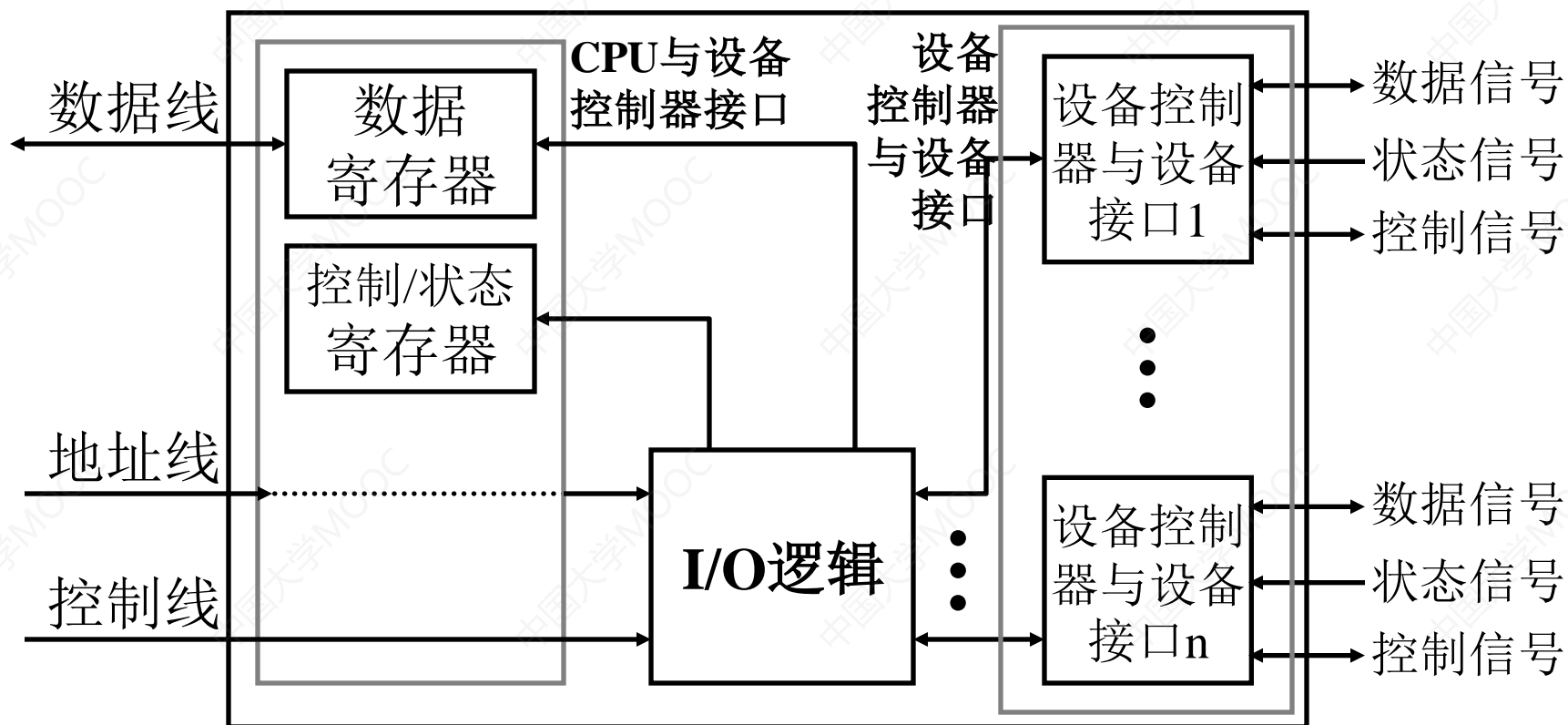
5.1.3 设备控制器

5.1.4 I/O通道

设备控制器的功能

- ❑ 接收和识别命令
 - 控制寄存器/命令译码器
- ❑ 数据交换（数据寄存器）
- ❑ 标识和报告设备状态（状态寄存器）
- ❑ 地址识别（地址译码器）
- ❑ 数据缓冲（缓冲器）
- ❑ 差错控制（差错检查码）

设备控制器的组成



5.1 I/O系统组成

5.1.1 I/O系统结构

5.1.2 I/O设备分类及接口

5.1.3 设备控制器

5.1.4 I/O通道

I/O通道及引入

- 通道引入的主要目的在于提高I/O过程的独立性，不仅使数据的传送能独立于CPU，而且使有关I/O操作的组织、管理与结束尽量独立，从而保证CPU有更多的时间进行数据处理
- 设置通道后，CPU只需向通道发出一条I/O指令；通道收到该指令后，便从内存取出本次要执行的通道程序并予执行；仅当其完成规定的I/O任务后，才向CPU发出中断信号
- 通道为特殊的处理机，其指令类型单一、无专有内存（与CPU共享内存），通过执行通道I/O程序来控制I/O操作

通道指令与通道程序

□ 通道指令

- 读指令：外设 \Rightarrow 内存
- 写指令：内存 \Rightarrow 外设
- 控制指令：具体功能与所控设备有关，如磁带机反绕、磁盘机寻道、打印机换页等

□ 通道程序

- 通道指令的逻辑集合，由设备驱动程序自动生成，存放于内存特定区域

通道类型

□ 字节多路通道

- 含有许多非分配型子通道分别连接在低、中速I/O设备上，子通道按时间片轮转方式共享主通道，按字节方式进行数据传送

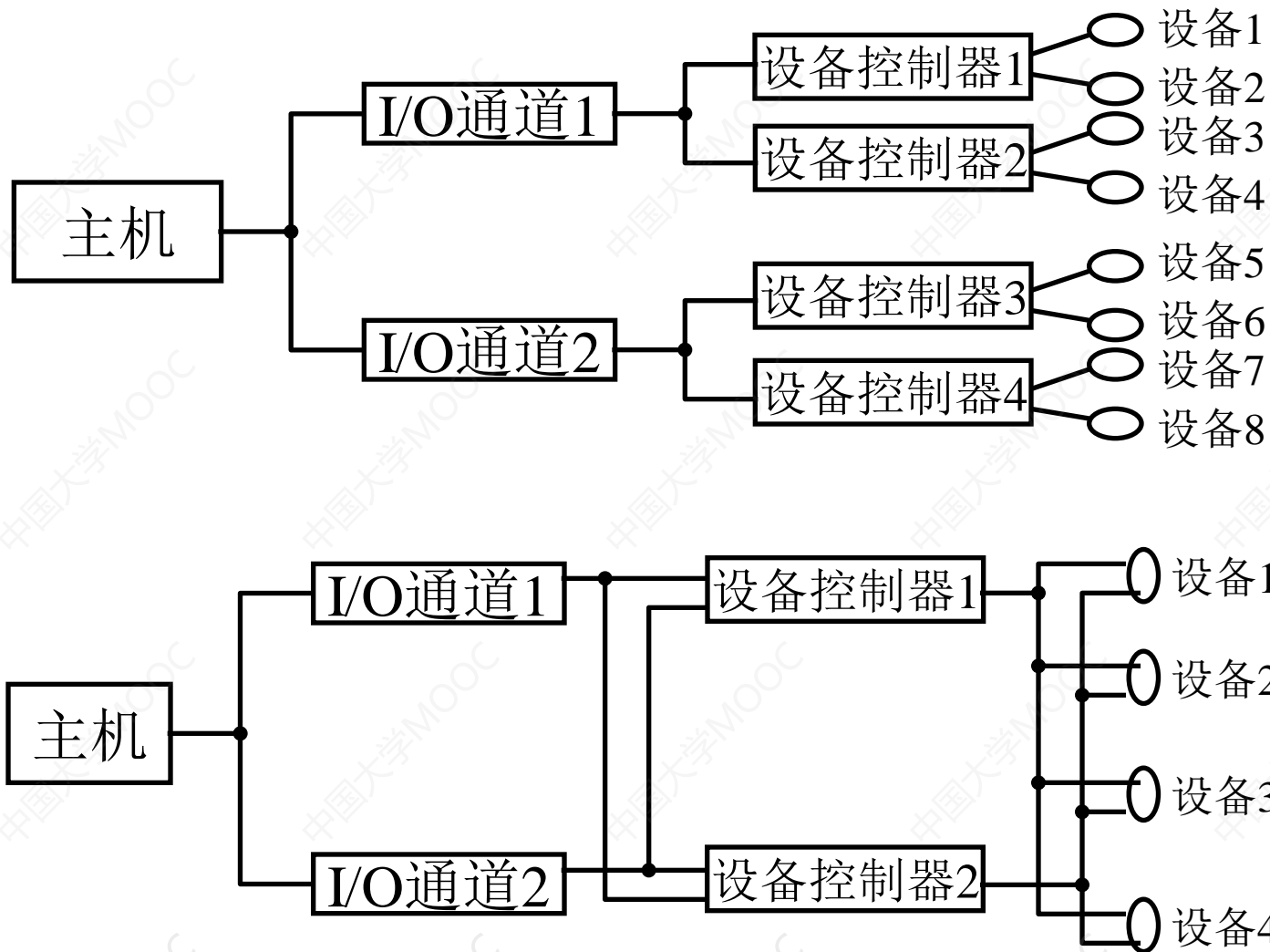
□ 数组选择通道

- 只含有一个分配型子通道，一段时间内只能执行一道通道程序、控制一台设备按数组方式进行数据传送，通道独占且利用率低，用于连接多台高速设备

□ 数组多路通道

- 含有多个非分配型子通道分别连接在高、中速I/O设备上，子通道按时间片轮转方式共享主通道，按数组方式进行数据传送

通道瓶颈问题及解决方案



5.1 I/O系统组成

5.1.1 I/O系统结构

5.1.2 I/O设备分类及接口

5.1.3 设备控制器

5.1.4 I/O通道

第五章 设备管理

5.1 I/O系统

5.2 I/O控制方式

5.3 设备管理目标、功能及结构

5.4 缓冲管理

5.5 设备分配

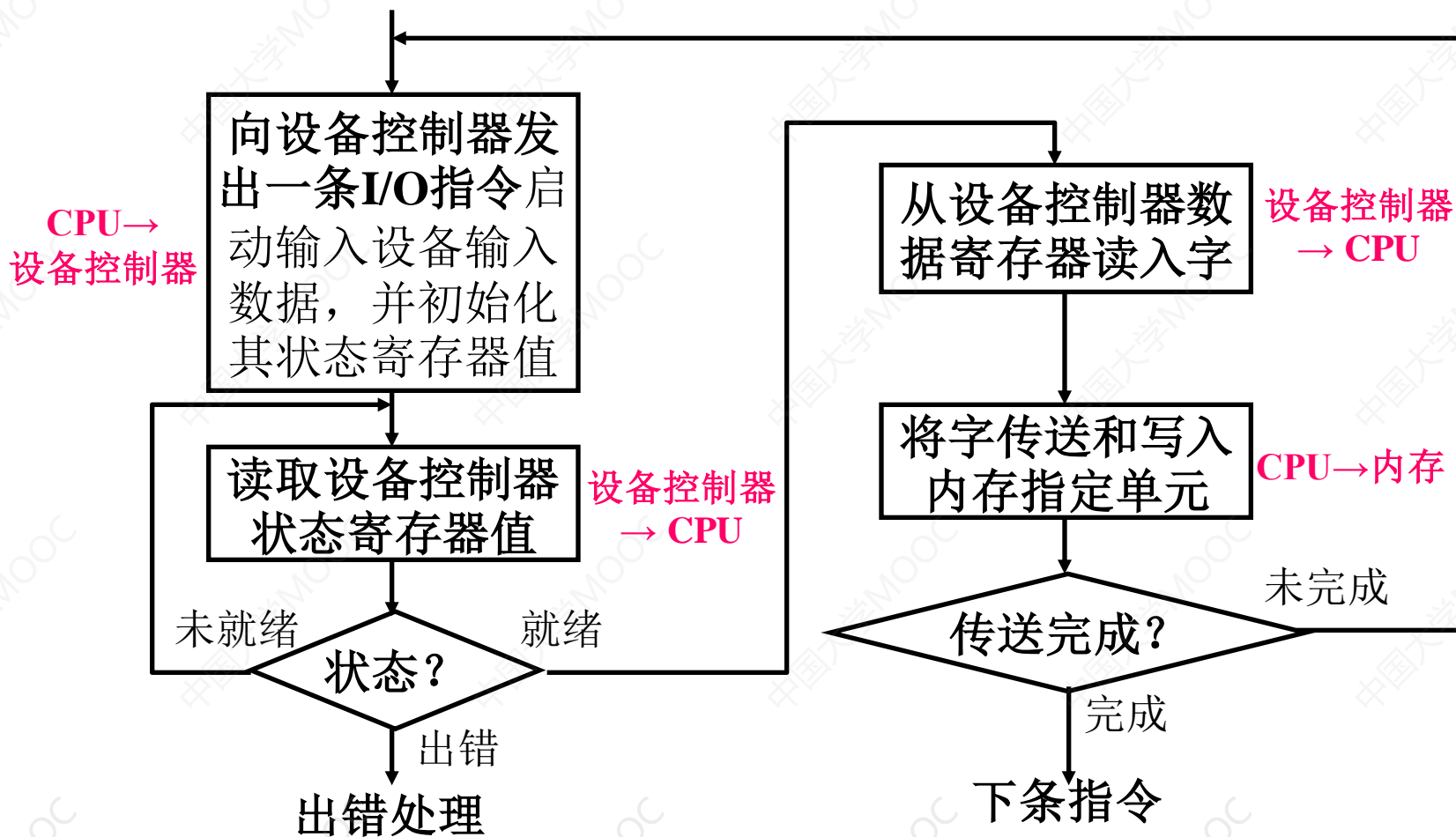
5.6 设备处理

5.7 磁盘存储器管理

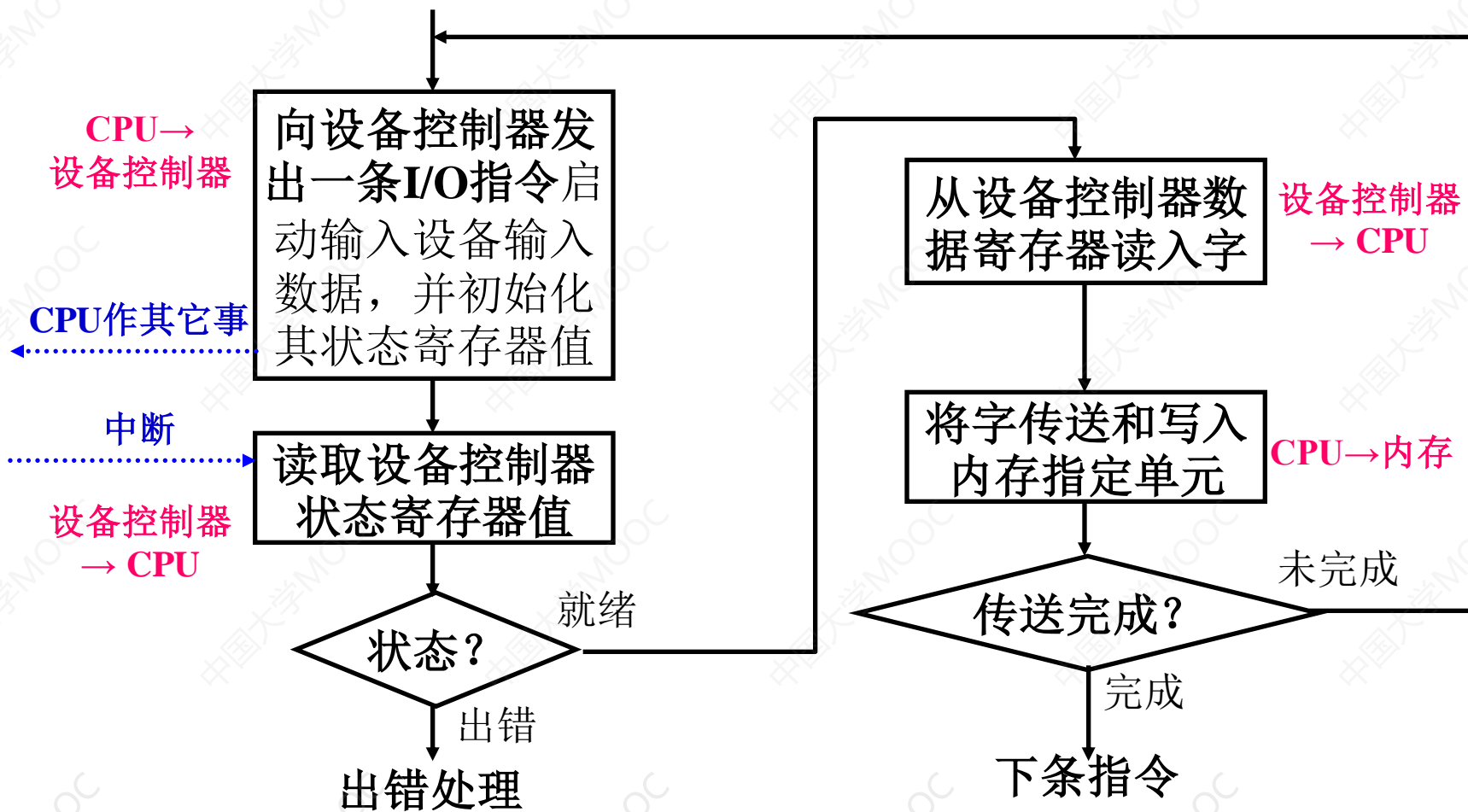
5.2 I/O控制方式

- ❑ 程序I/O控制方式
 - 简单的忙-等待方式
- ❑ 中断驱动I/O控制方式
 - 中断机制的引入
- ❑ 直接内存存取型I/O控制方式
 - DMA控制器、数据传输单位扩大
- ❑ 通道型I/O控制方式
 - 通道、I/O操作组织和数据传送的独立

程序I/O控制方式 读操作为例

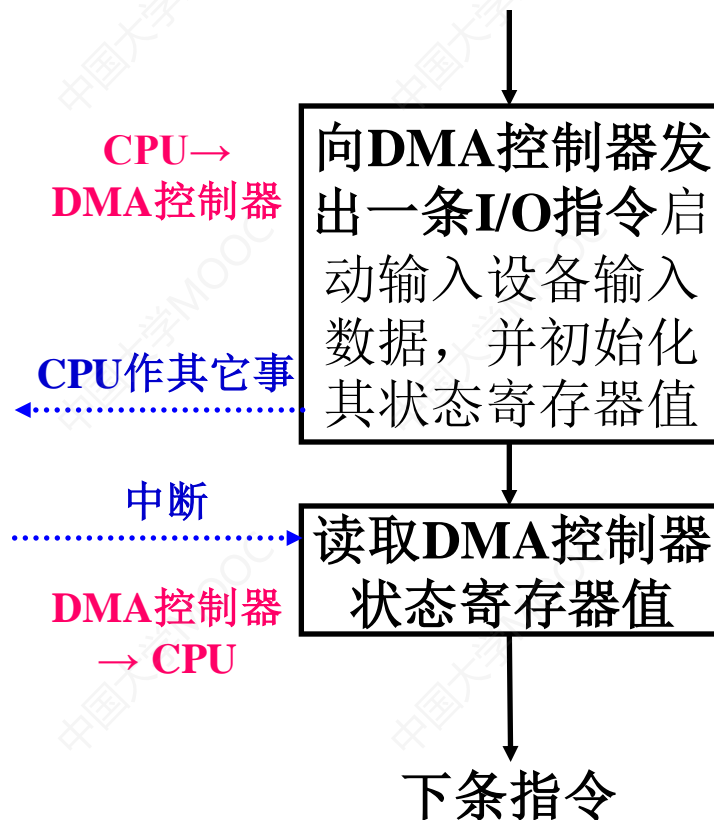


中断驱动I/O控制方式 读操作为例



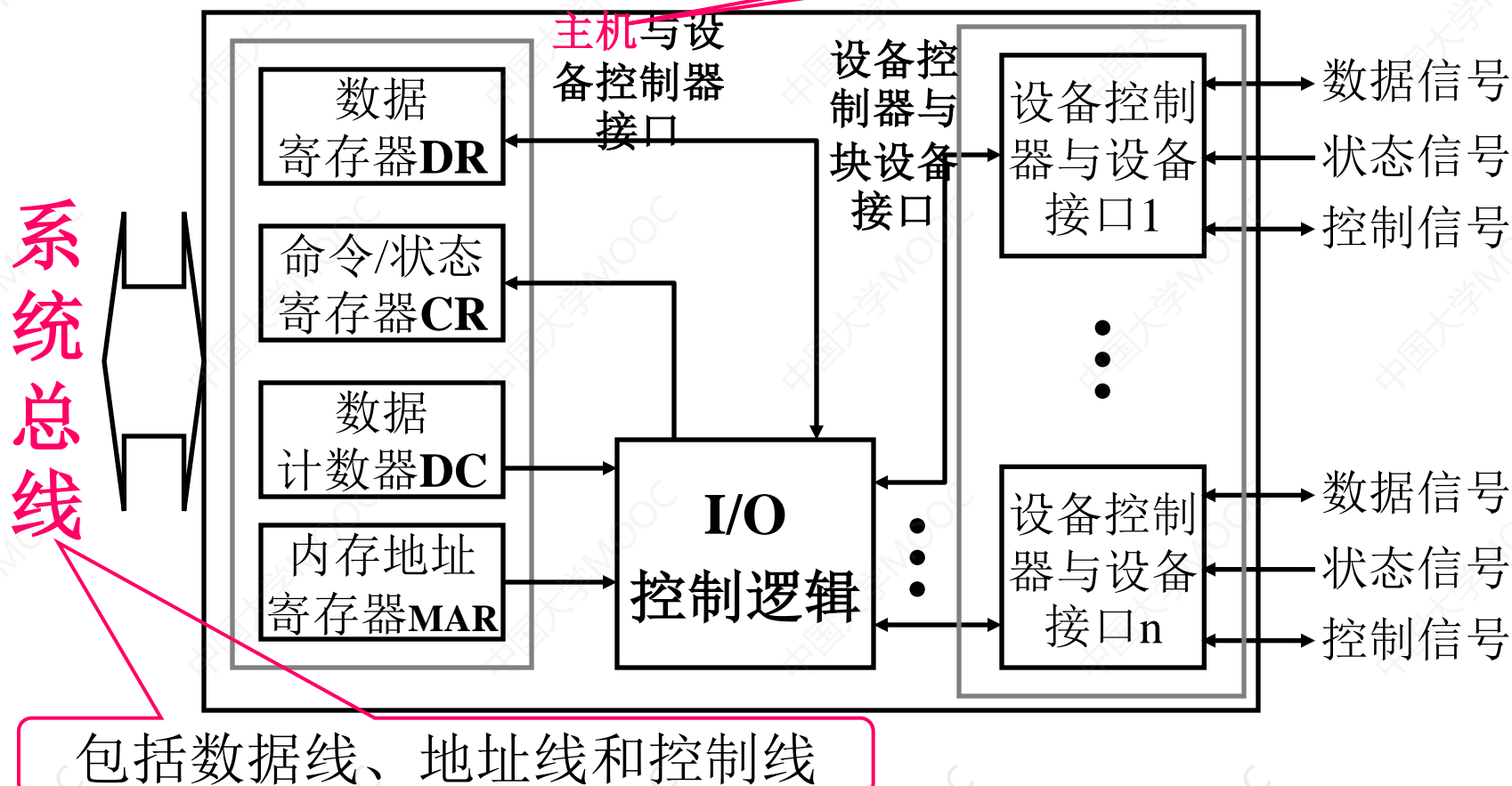
直接内存存取型I/O控制方式

- ❑ 数据传输单位
 - 数据块
- ❑ 数据传输途径
 - 设备 \leftrightarrow 内存
- ❑ CPU干预
 - 限于数据块传送开始与结束

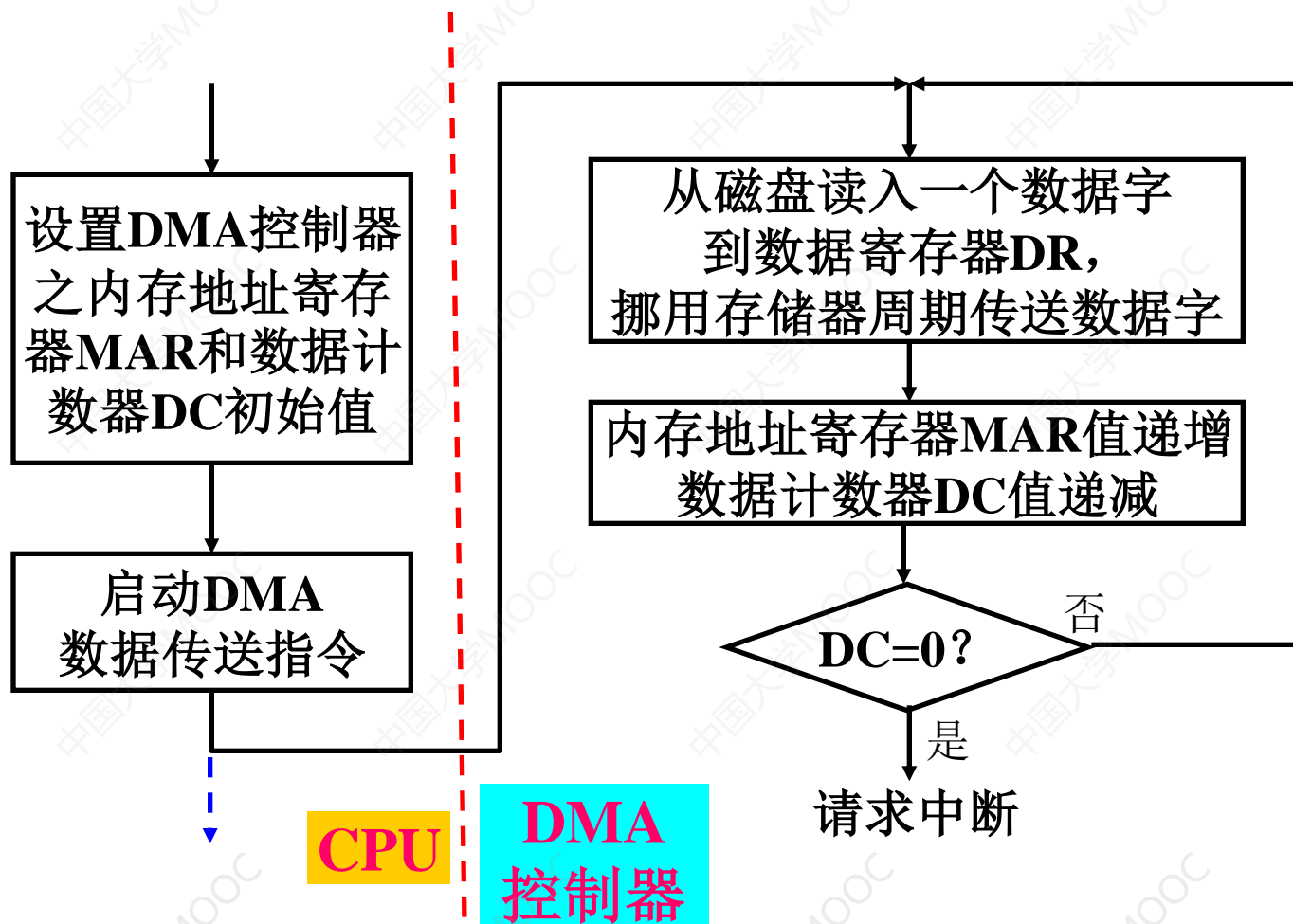


DMA控制器组成

包括CPU和内存



基于DMA磁盘读入数据流程



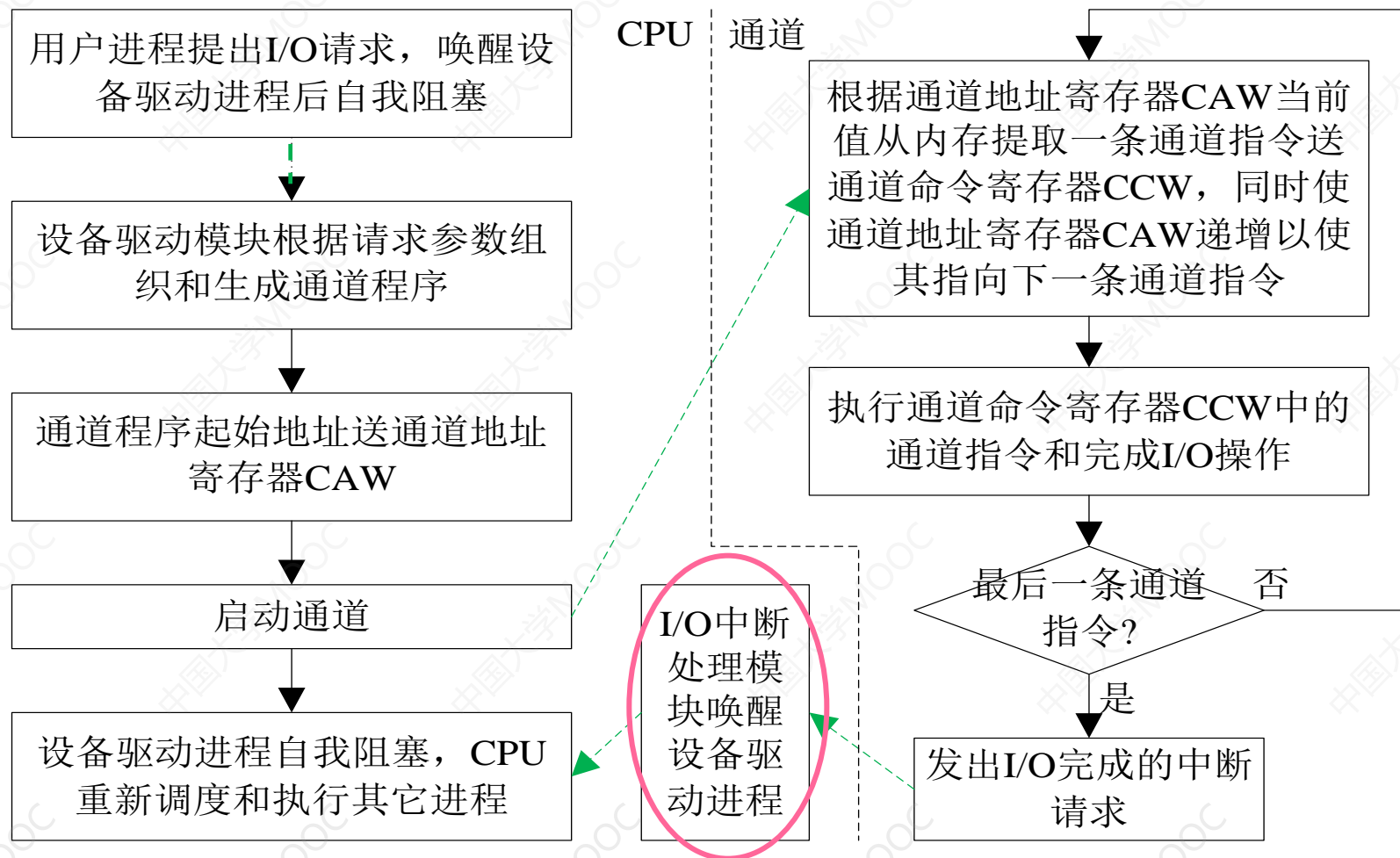
通道型I/O控制方式

| 操作码 | 通道程序 结束标志 | 记录 结束标志 | 数据读写 字节计数 | 内存地址 |
|-------|--------------|------------|--------------|------|
| WRITE | 0 | 0 | 80 | 813 |
| WRITE | 0 | 0 | 140 | 1034 |
| WRITE | 0 | 1 | 60 | 5830 |
| WRITE | 0 | 1 | 300 | 2000 |
| WRITE | 0 | 0 | 250 | 1850 |
| WRITE | 1 | 1 | 250 | 720 |

通道程序

同一记录操作

通道程序的执行流程



关于内核编

中断请求号（设备/端口）



BIOS中断号

■ 中断控制方式与

- 两片级联的8259A可编程中断控制器组成中断控制器，可以支持15台设备的中断I/O控制功能
- 计算机启动时，ROM-BIOS对其进行初始化，并在内存0x000~0xFFF区域建立一个中断向量表
- 不过有关设置与Intel公司要求相违背，所以操作系统设计人员需要重新设置

■ 内存直接存取控制器

- 采用Intel 8237芯片或其兼容芯片实现，两片可提供8个独立通道
- 使用一个通道前必须首先进行设置，主要涉及页面寄存器端口、偏移地址寄存器端口和数据计数寄存器端口

关于内核编程的思考²

□ 定时/计数器

- Intel 8253/8254 可编程计数器/定时器提供了3

Intel 8042 单片微处理器芯片或其兼容电路

输出端口P2用处特殊：

P20 引脚用于实现处理器的复位操作

P21 引脚用于控制A20信号线的开启与否

□ 键盘

- 键盘上有一个称为键盘编码器的处理器（Intel 8048 或兼容芯片）专门扫描收集按键按下和松开的状态信息（即**通/断开**扫描码）
- 主机**键盘控制器**则负责对接收到的键盘扫描码进行解码并将之发送到操作系统的键盘数据队列中

关于内核编程的思考³

□ 串行控制器

- 计算机一般带有两个符合RS232C标准的串行接口，并使用由通用异步接收/发送器控制芯片UART（Universal Asynchronous Receiver/Transmitter）组成的串行控制器来处理串行数据的收发工作

Monochrome Display Adapter

Color Graphics Adapter

Video Graphics Adapter

视频图形阵列适配器

- MDA显示标准（8KB：0xB0000~0xB1FFF）
- CGA显示标准（16KB：0xB8000~0xBBFFF）
- EGA/VGA显示标准（32KB：0xA0000~）

关于内核编程的思考4

❑ 软盘控制器与硬盘控制器

- 软盘驱动控制器、硬盘控制器
- 软盘控制器需要DMA控制器实施数据传输
- 硬盘控制器采用高速数据块进行传输，不需要DMA控制器的介入
- 三维物理地址
- 磁盘控制编程主要通过I/O端口设置控制器的相关寄存器内容或获取操作的结果信息

作业题

- **5.1** 试从操作系统设备管理的角度谈谈你对计算机输入输出系统的组成结构及组成要素的认识与理解。
- **5.2** 比较和阐述各种I/O控制方式的技术实现要领。

第五章 设备管理

5.1 I/O系统组成

5.2 I/O控制方式

5.3 设备管理目标、功能及结构

5.4 缓冲管理

5.5 设备分配

5.6 设备处理

5.7 磁盘存储器管理

设备管理及目标

- ❑ 设备管理程序用于控制与管理实现信息输入、输出和存储的设备。由于I/O设备不仅种类繁多，而且彼此特性和操作方式相差较大，从而使设备管理成为操作系统中最繁杂且与硬件紧密相关的部分
- ❑ 主要目标
 - 方便用户使用，实现对不同类型设备的统一管理、使用并协调各台外围设备的并发运作
 - 提高系统I/O效率，缓解CPU和多种不同处理速度的外围设备之间的矛盾
 - 设备管理的可扩充性，包括设备的增加和删除以及对新的设备类型的适应性

设备管理的功能

- ❑ 提供设备使用的用户接口
 - 命令接口和编程接口
- ❑ 设备分配和释放
 - 分配设备及相应的通道、设备控制器
- ❑ 设备的访问和控制
 - 并发访问及差错处理
- ❑ I/O缓冲和调度
 - 提高I/O访问效率，缓解CPU与外设矛盾

设备管理层次结构

用户进程

用户空间I/O请求支持层

设备独立性软件层

设备调度与控制层

硬件

设备管理结构

□ 用户空间I/O请求支持层

- 提供对逻辑设备的控制。具体来说，针对用户接口，提供抽象的命令，如Open, Close, Read, Write；针对通信设备，提供通信体系结构如网络协议栈；针对文件存储设备，提供文件系统的逻辑结构控制。

□ 设备独立性软件层

- 逻辑设备与物理设备间过渡协调机构，实现用户命令到设备操作序列的转换，提供缓冲机制

□ 设备调度和控制层

- 实现硬件物理设备的设备驱动、设备控制、状态维护、中断处理及并发I/O访问调度

作业题

- 5.3 谈谈你对操作系统设备管理的目标、功能及层次结构设计认识与理解。

第五章 设备管理

5.1 I/O系统组成

5.2 I/O控制方式

5.3 设备管理目标、功能及结构

5.4 缓冲管理

5.5 设备分配

5.6 设备处理

5.7 磁盘存储器管理

5.4 缓冲管理

5.4.1 缓冲技术的引入

5.4.2 单缓冲

5.4.3 双缓冲

5.4.4 循环缓冲

5.4.5 缓冲池

缓冲技术的引入

□ 操作系统引入缓冲机制的原因

- 缓和CPU与I/O设备速度不匹配的矛盾
- 减少对CPU的中断频率，放宽对中断响应时间的限制
- 提高CPU与I/O设备之间的并行性

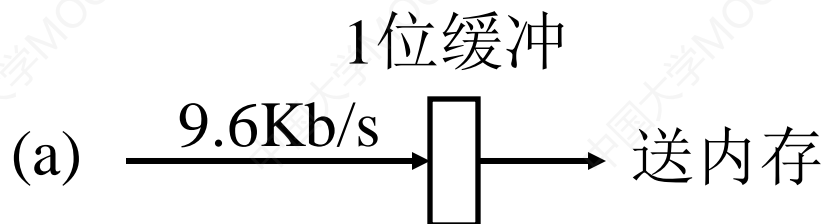
□ 缓冲管理的主要功能

- 缓冲区的组织
- 缓冲区的获得与释放

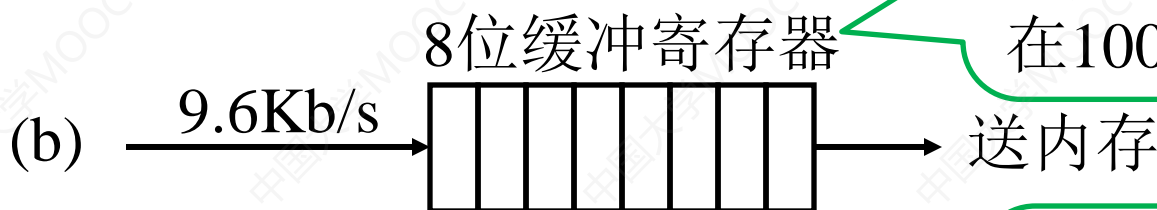
利用缓冲降低中断频率

举例说明

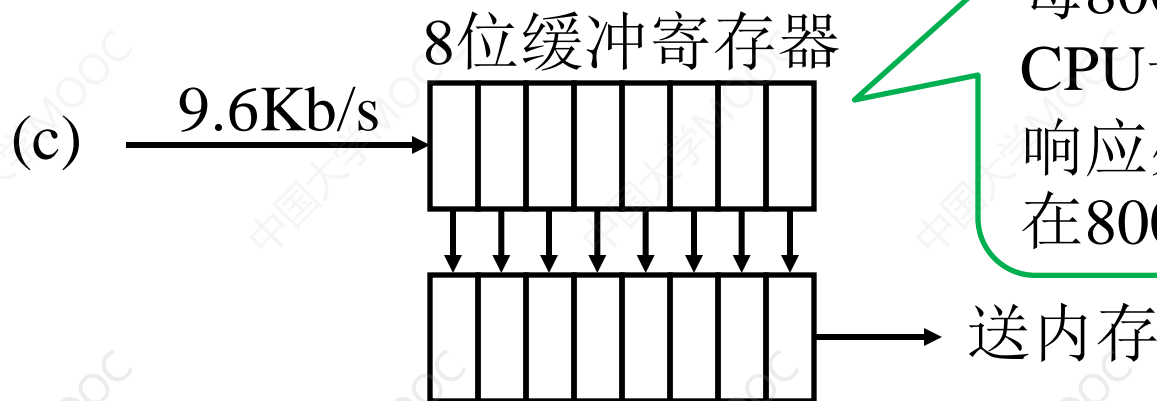
远程通信系统



每100 μ s中断
每800 μ s中断
CPU一次，且
响应处理必须
在100 μ s内完成



每800 μ s中断
CPU一次，且
响应处理必须
在800 μ s内完成



5.4 缓冲管理

5.4.1 缓冲技术的引入

5.4.2 单缓冲

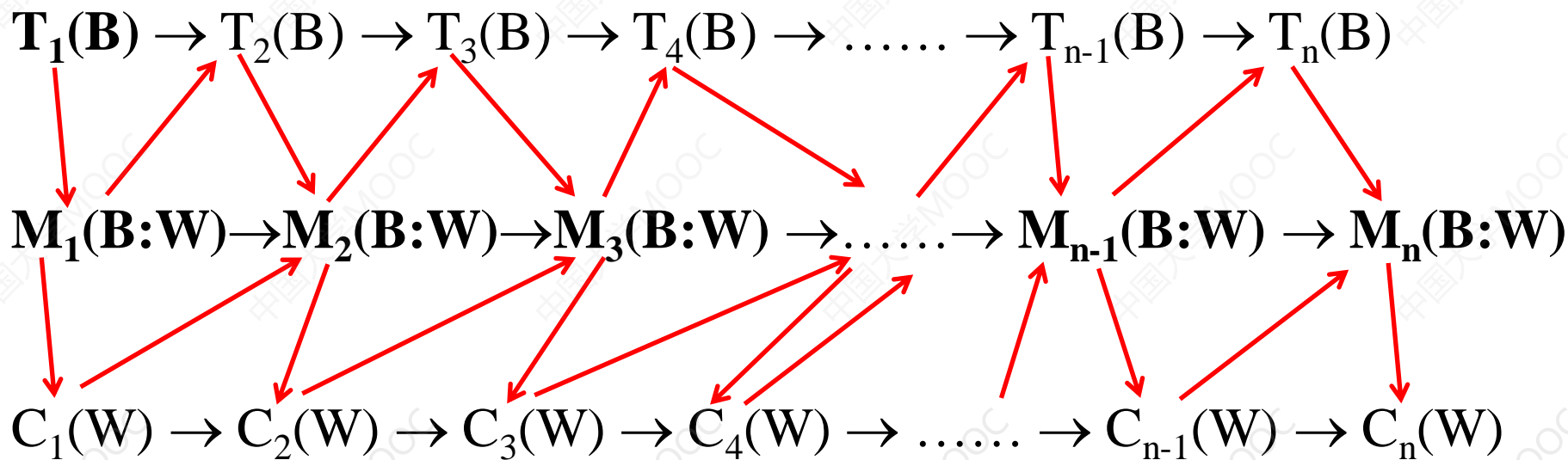
5.4.3 双缓冲

5.4.4 循环缓冲

5.4.5 缓冲池

单缓冲

$$\frac{T_1 + \sum_{i=1}^{n-1} \max(T_{i+1}, C_i) + \sum_{i=1}^n M_i + C_n}{n}$$



5.4 缓冲管理

5.4.1 缓冲技术的引入

5.4.2 单缓冲

5.4.3 双缓冲

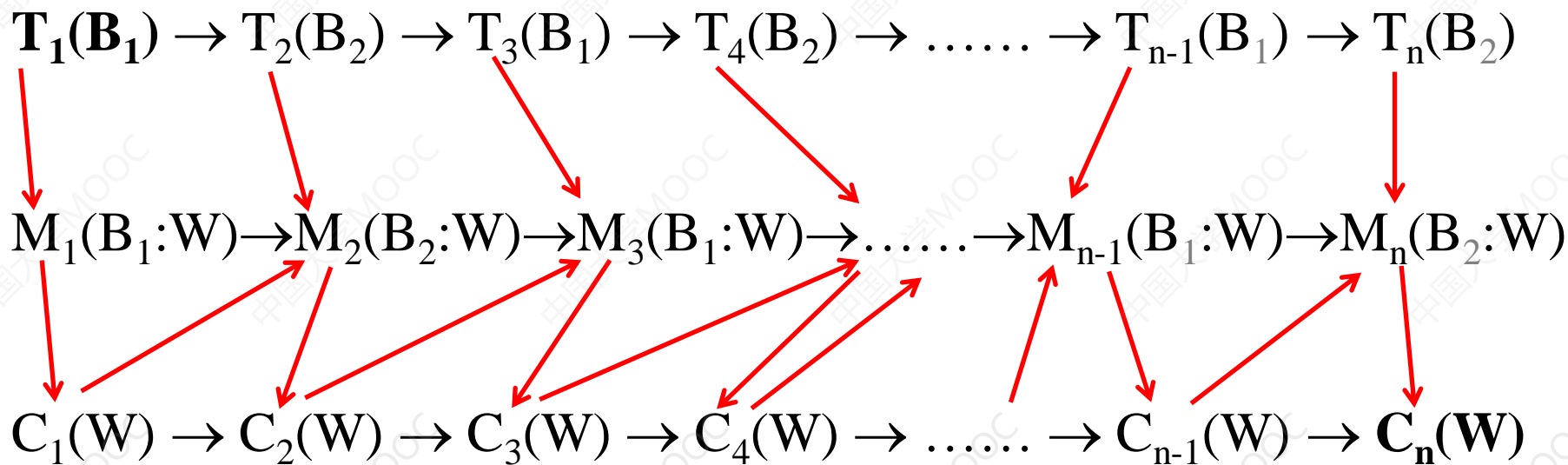
5.4.4 循环缓冲

5.4.5 缓冲池

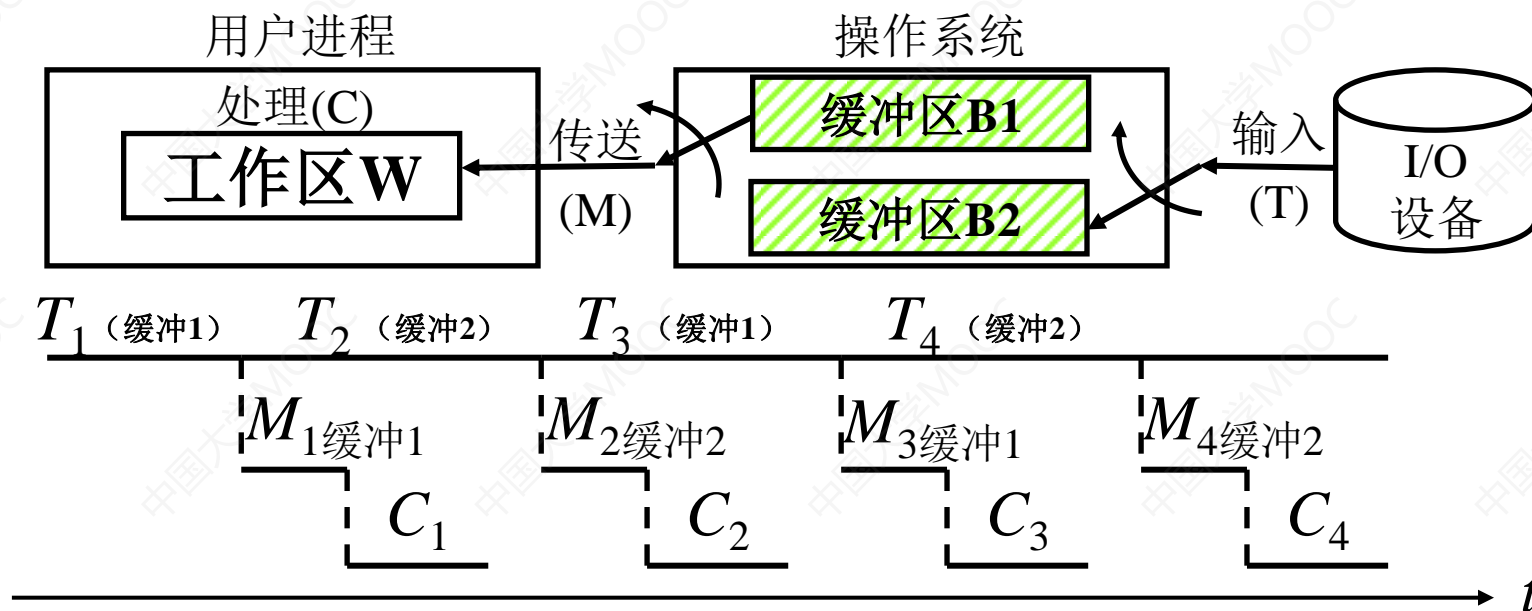
双缓冲

工作区是瓶颈！

$$\frac{T_1 + \sum_{i=1}^{n-1} \max(T_{i+1}, M_i + C_i) + M_n + C_n}{n}$$



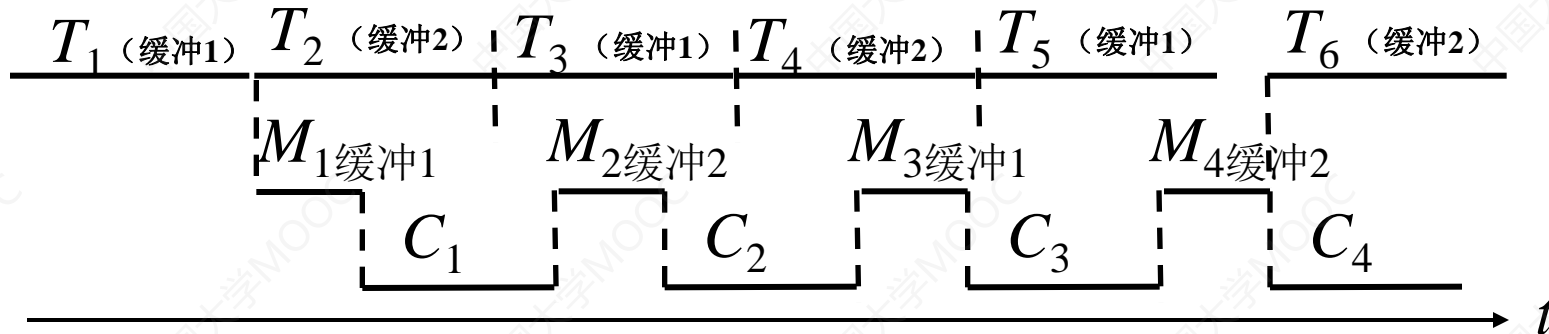
双缓冲



$$T_{i+1} \geq M_i + C_i : \frac{\sum_{i=1}^n T_i + M_n + C_n}{n} = T$$

双缓冲

$$T_{i+1} \leq M_i + C_i : \frac{T_1 + \sum_{i=1}^n (M_i + C_i)}{n} = M + C$$



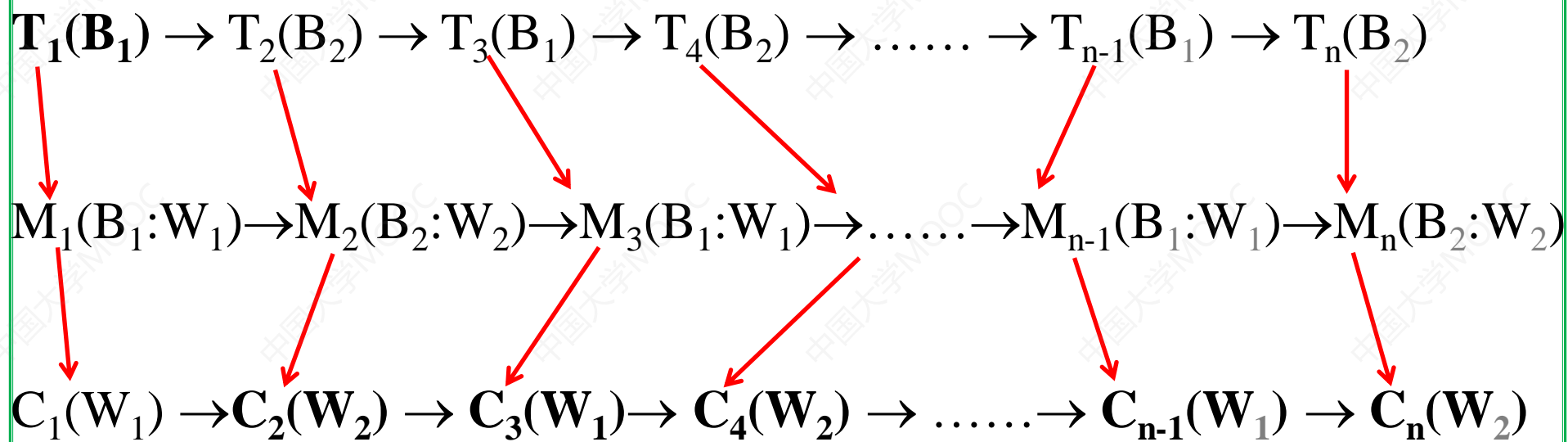
系统对每一整块数据的平均处理时间为

$$\max(M+C, T)$$

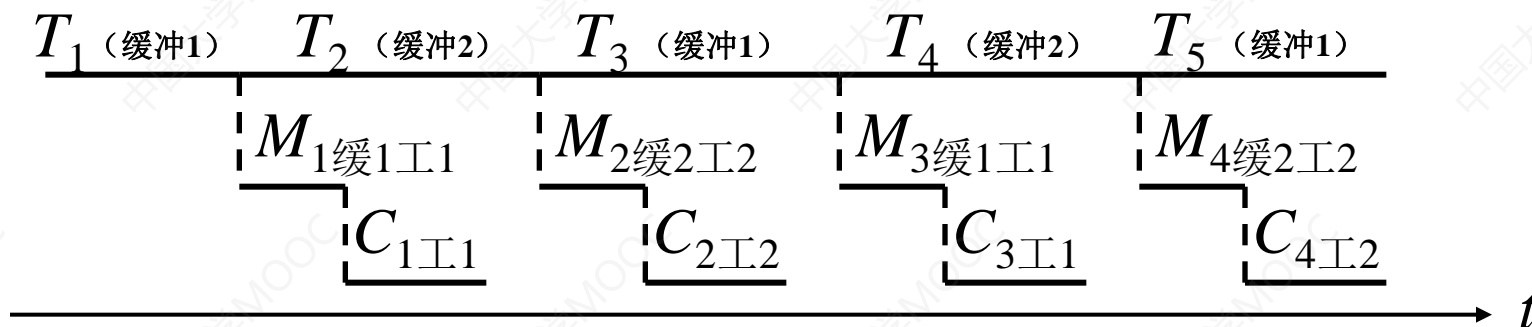
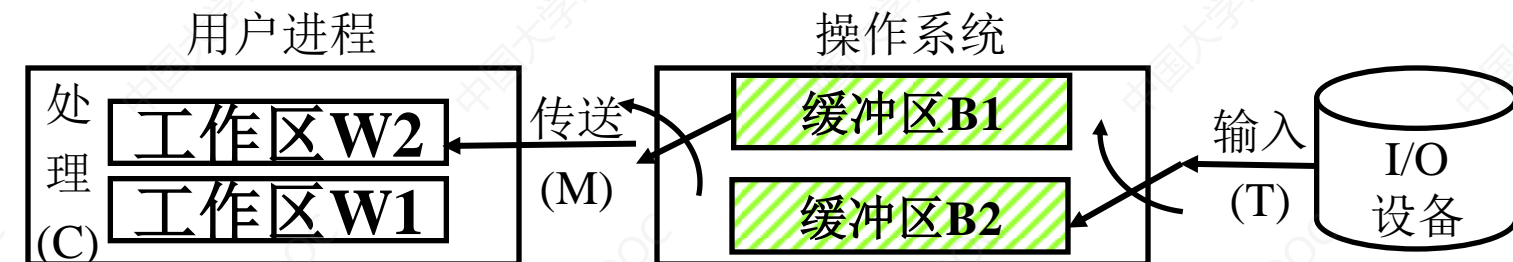
双缓冲

工作区的变化!
计算成为瓶颈!

$$\frac{T_1 + \sum_{i=1}^{n-1} \max (T_{i+1} + M_{i+1}, M_i + C_i) + C_n}{n}$$

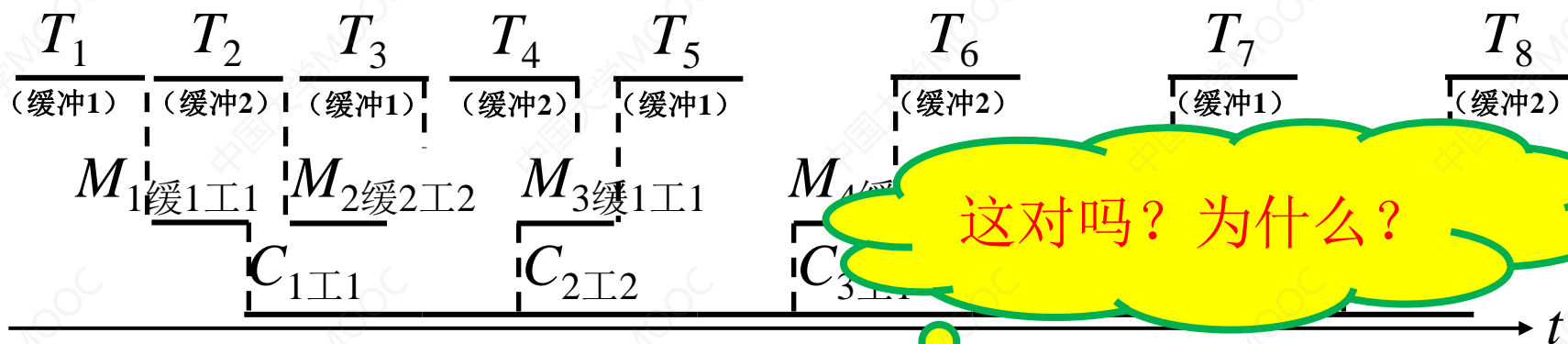
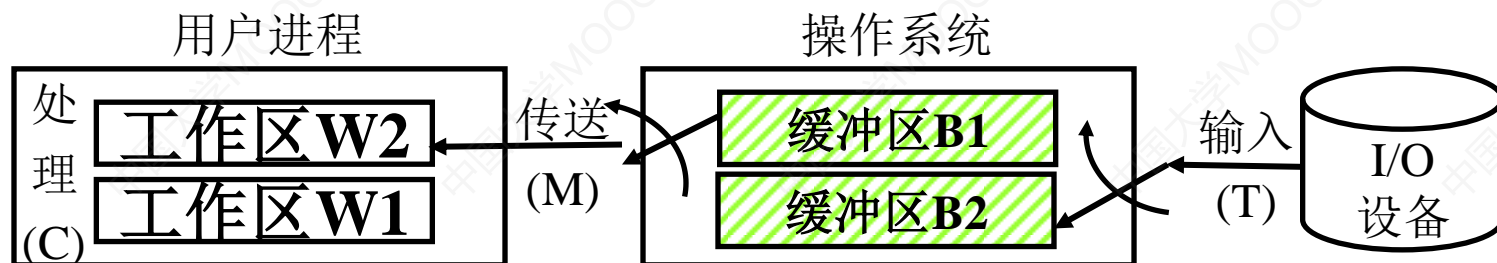


双缓冲



$$T_{i+1} + M_{i+1} \geq M_i + C_i : \frac{T_1 + \sum_{i=2}^n (T_i + M_i) + C_n}{n} = T + M$$

双缓冲

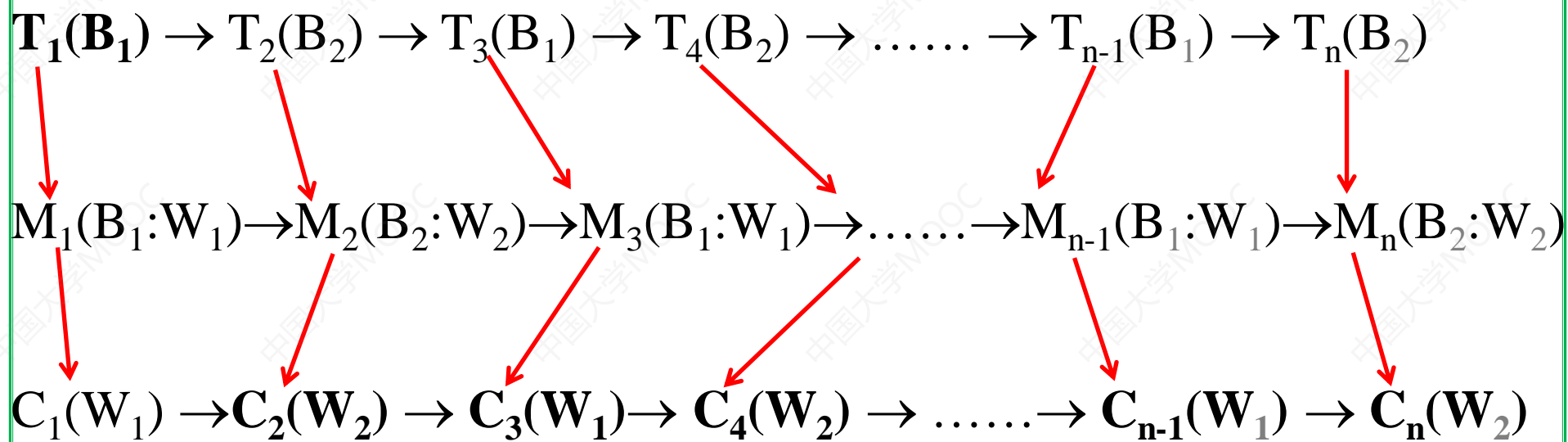


这对吗？为什么？

$$T_{i+1} + M_{i+1} \leq M_i + C_i : \frac{T_1 + \sum_{i=1}^{n-1} (M_i + C_i) + C_n}{n} = M + C$$

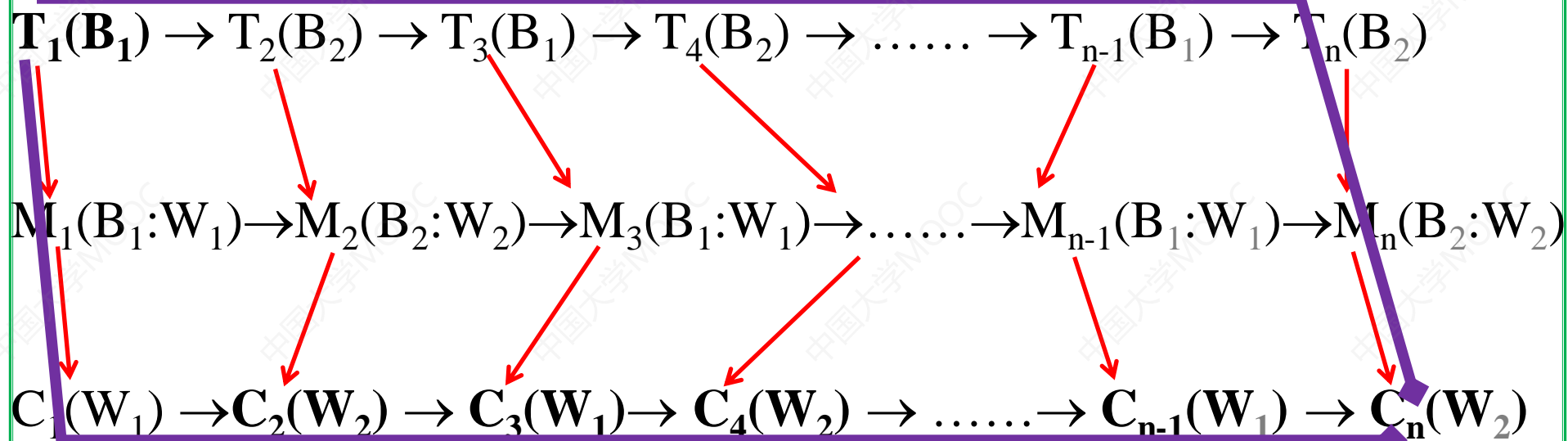
双缓冲

$$\frac{T_1 + \sum_{i=1}^{n-1} \max (T_{i+1} + M_{i+1}, M_i + C_i) + C_n}{n}$$

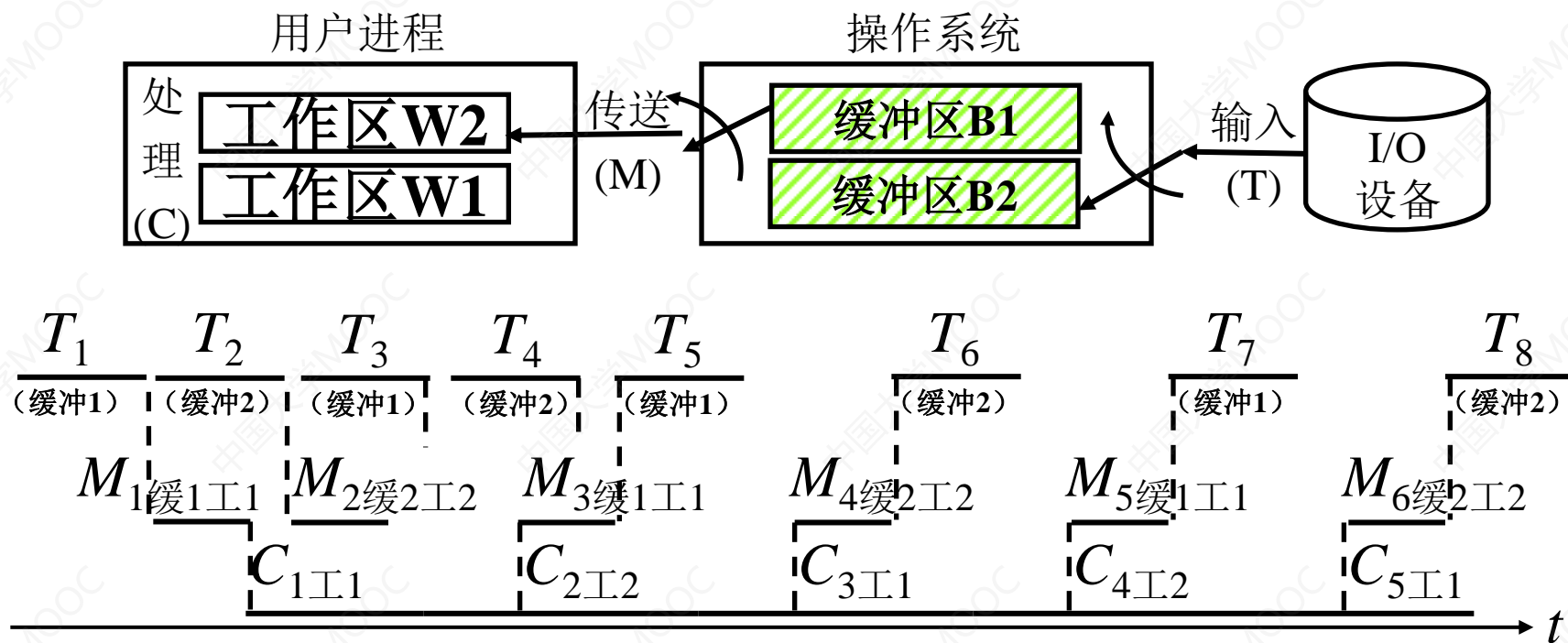


双缓冲

$$T_{i+1} + M_{i+1} \geq M_i + C_i : \frac{\sum_{i=1}^n T_i + M_n + C_n}{n} = T$$



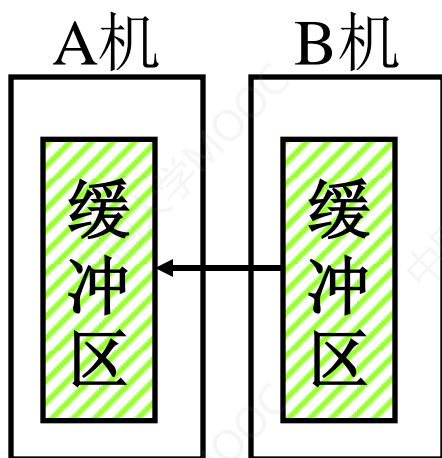
双缓冲



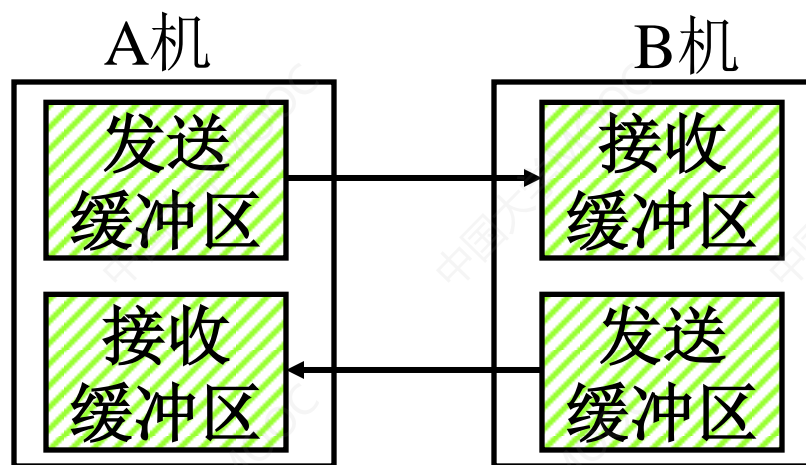
系统对每一整块数据的平均处理时间为

$$\max(C, T)$$

利用双缓冲实现双向通信



双机通信单缓冲
只能实现单向数据传输



双机通信双缓冲
可以实现双向数据传输

5.4 缓冲管理

5.4.1 缓冲技术的引入

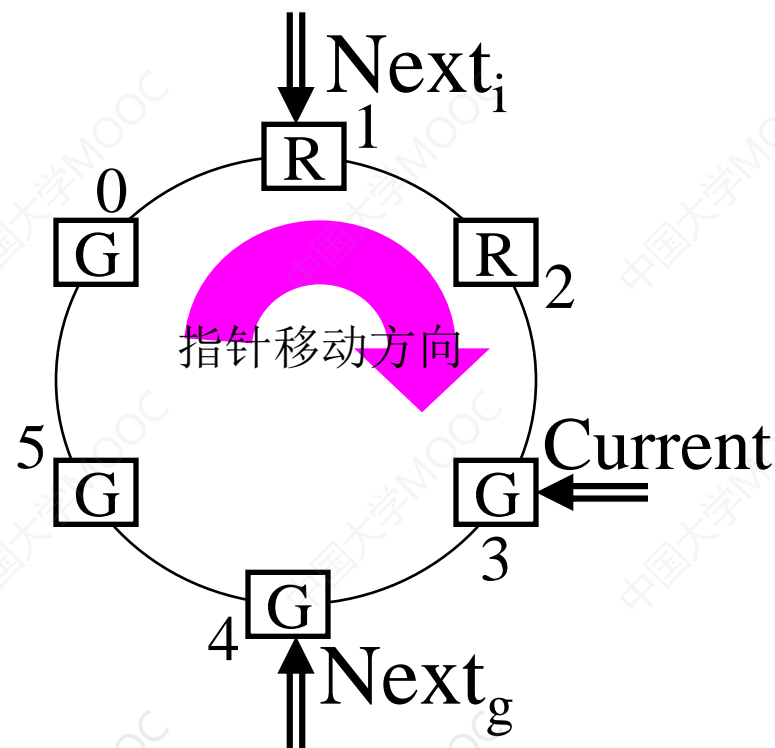
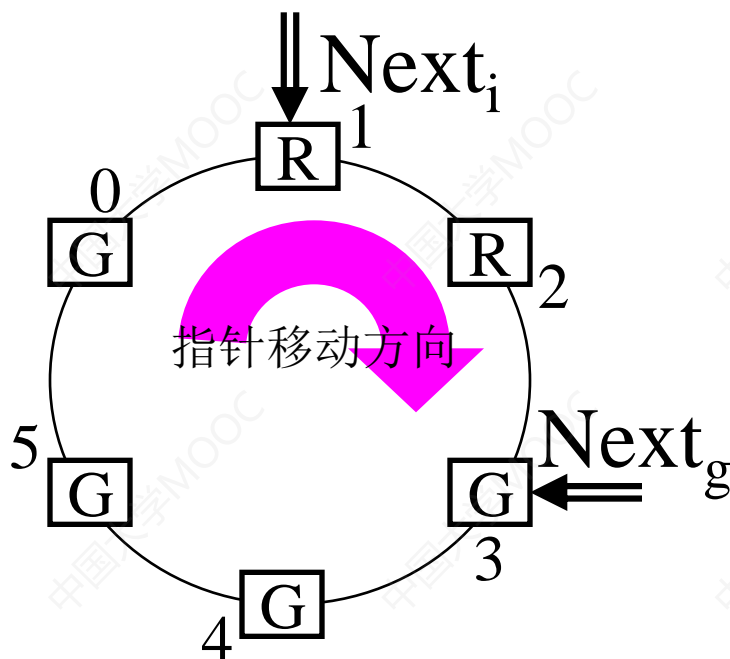
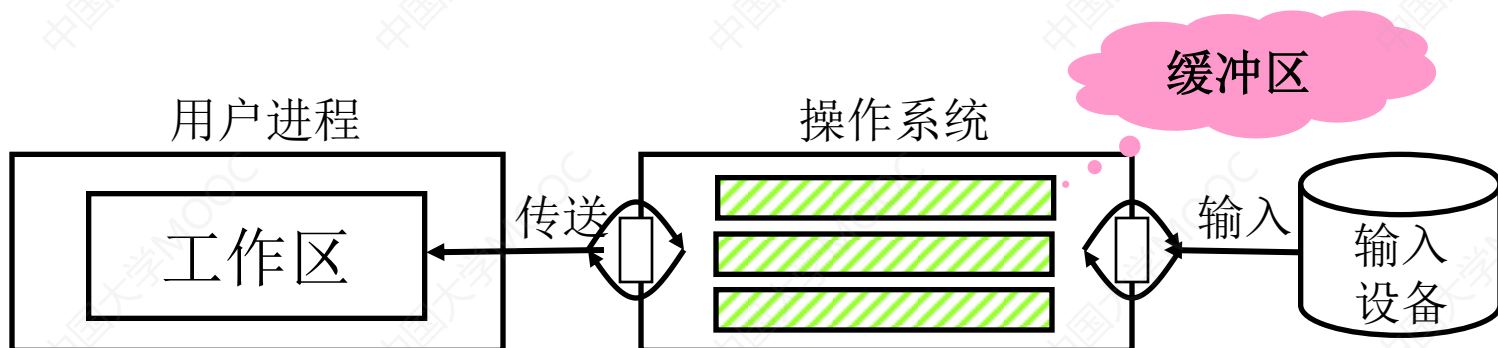
5.4.2 单缓冲

5.4.3 双缓冲

5.4.4 循环缓冲

5.4.5 缓冲池

（输入）循环缓冲的组成



缓冲区的使用及进程同步

□ 缓冲区使用过程

- GetBuf (BufType, Current)过程
- ReleaseBuf (BufType, Current) 过程

□ 进程同步

- $Next_i$ 指针追赶上 $Next_g$ 指针时，输入进程应阻塞，直至计算进程把某满缓冲区数据取走并调用ReleaseBuf 过程将之唤醒
- $Next_g$ 指针追赶上 $Next_i$ 指针时，计算进程应阻塞，直至输入进程装满某空缓冲区并调用ReleaseBuf 过程将之唤醒

5.4 缓冲管理

5.4.1 缓冲技术的引入

5.4.2 单缓冲

5.4.3 双缓冲

5.4.4 循环缓冲

5.4.5 缓冲池

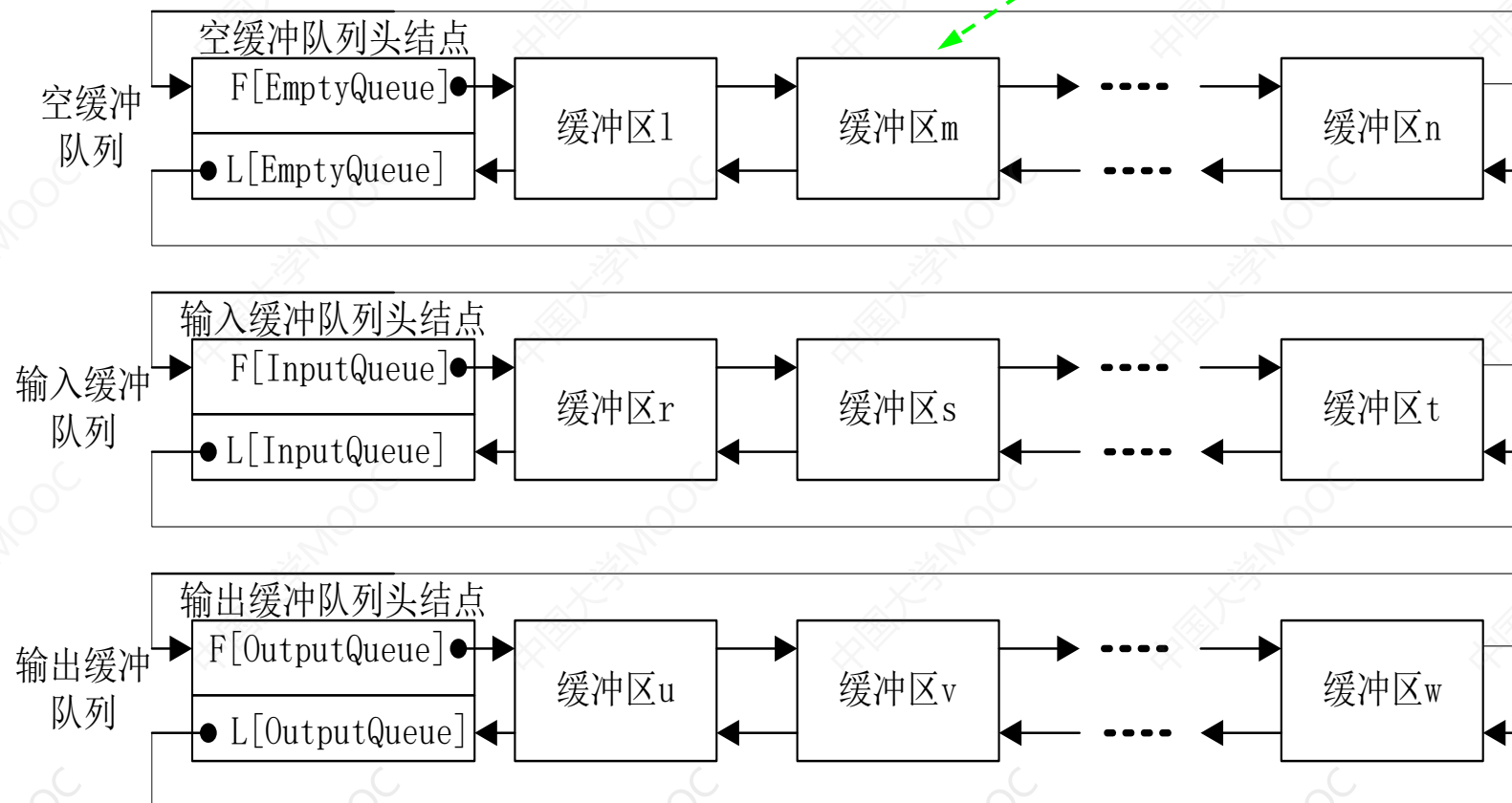
缓冲池的组成

- ❑ 三个队列及队首尾指针 ($F[\dots]/L[\dots]$)
 - 空缓冲队列 EmptyQueue
 - 输入队列 InputQueue
 - 输出队列 OutputQueue
- ❑ 四种工作缓冲区
 - 用于收容输入数据的工作缓冲区 Hin
 - 用于提取输入数据的工作缓冲区 Sin
 - 用于收容输出数据的工作缓冲区 Hout
 - 用于提取输出数据的工作缓冲区 Sout

缓冲队列组织形式

缓冲首部

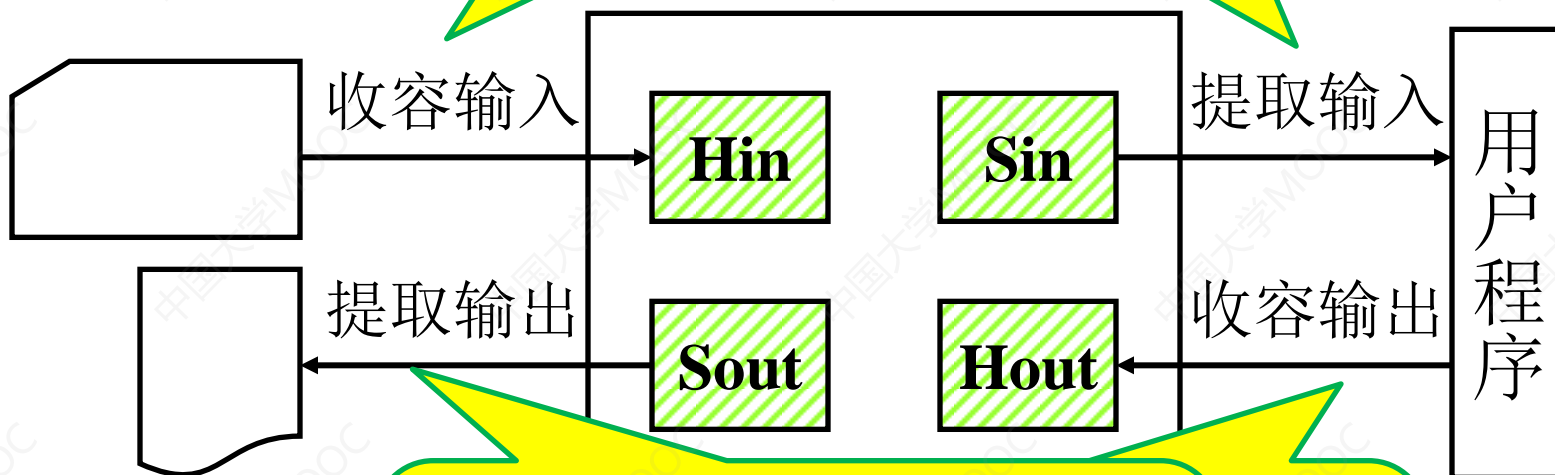
| |
|-------|
| 设备号 |
| 数据块号 |
| 缓冲器号 |
| 互斥标识位 |
| 连接指针 |



缓冲区

GetBuf
输入数据
PutBuf
从Sin中提取输入数据;
PutBuf(EmptyQueue, Sin);

缓冲池



GetBuf(OutputQueue, Sout);
从Sout中提取输出数据输出;
PutBuf(EmptyQueue, Sout);
Hout);

❑ Procedure

❑ Procedure

GetBuf过程和PutBuf过程

VAR

integer: EmptyQueue:=0, InputQueue:=1, OutputQueue:=2;
semaphore: RS[0..2]:={N,0,0}, MS[0..2]:={1,1,1};

Procedure GetBuf(QType,
pBuf)

begin

wait(RS[QType]);

wait(MS[QType]);

pBuf := TakeBuf(QType);

signal(MS[QType]);

end

Procedure PutBuf(QType,
pBuf)

begin

wait(MS[QType]);

AddBuf(QType, pBuf);

signal(MS[QType]);

signal(RS[QType]);

end

5.4 缓冲管理

5.4.1 缓冲技术的引入

5.4.2 单缓冲

5.4.3 双缓冲

5.4.4 循环缓冲

5.4.5 缓冲池

作业题

- **5.4** 为什么要引入缓冲机制？主要有哪些缓冲类型？并就循环缓冲和缓冲池的组织与实现机制展开详细阐述。
- **5.5** 为什么在单缓冲情况下，系统对一块数据的处理时间为 $\max(C, T) + M$ ？
- **5.6** 分析讨论在双缓冲情况下，系统对一块数据的平均处理时间。

第五章 设备管理

5.1 I/O系统组成

5.2 I/O控制方式

5.3 设备管理目标、功能及结构

5.4 缓冲管理

5.5 设备分配

5.6 设备处理

5.7 磁盘存储器管理

5.5 设备分配

5.5.1 设备分配用数据结构

5.5.2 设备分配考虑因素

5.5.3 设备独立性

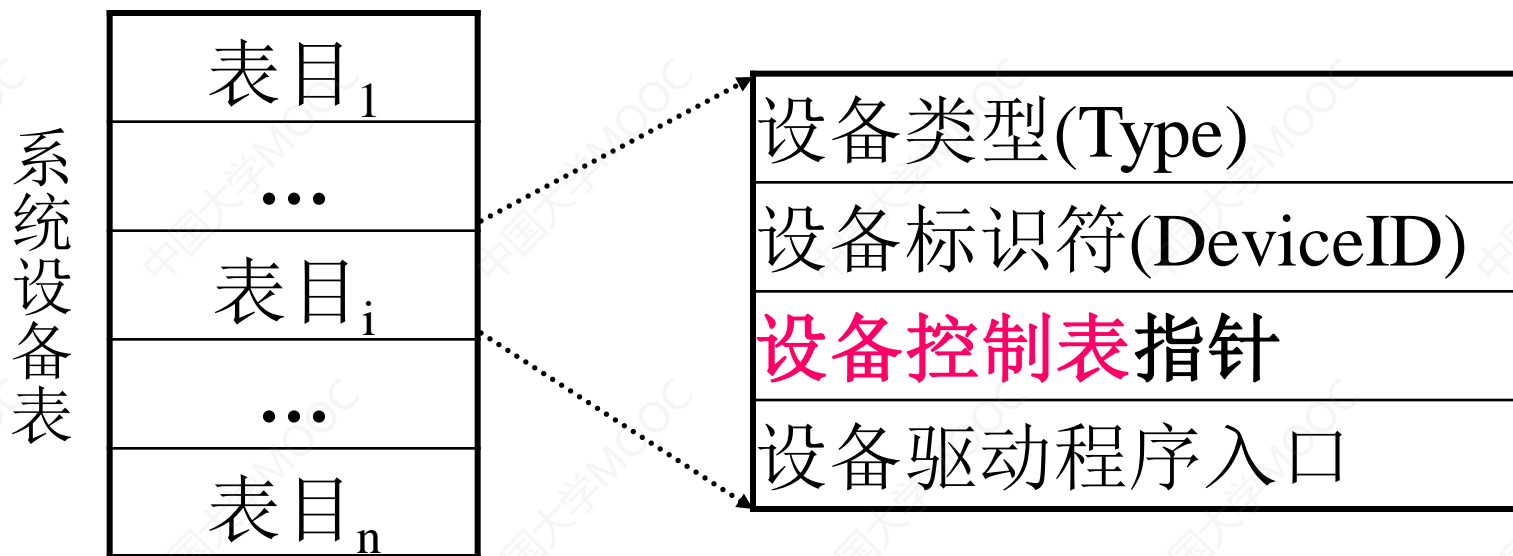
5.5.4 独占设备的分配程序

5.5.5 假脱机技术

设备分配用数据结构

- ❑ 系统设备表
 - 记录系统全部设备的情况
- ❑ 设备控制表
 - 针对每台设备而设置和记录对应情况
- ❑ 设备控制器控制表
 - 针对每个设备控制器而设置和记录对应情况
- ❑ 通道控制表
 - 针对每个通道而设置和记录对应情况

系统设备表SDT



设备控制表DCT

设备控制表集合

| |
|------------------|
| DCT ₁ |
| ... |
| DCT _i |
| ... |
| DCT _n |

设备类型(Type)

设备标识符(DeviceID)

设备状态：等待/不等待；忙/闲

与设备连接的**设备控制器控制表**的指针

重复执行次数或时间

设备请求队列队首及队尾指针

?

设备控制器控制表COCT

设备控制器控制表集合

| |
|-------------------|
| COCT ₁ |
| ... |
| COCT _j |
| ... |
| COCT ₁ |

设备控制器标识符(ControllerID)

设备控制器状态：忙/闲

与设备控制器连接的通道控制表的指针

设备控制器请求队列队首指针

设备控制器请求队列队尾指针

通道控制表CHCT

通道控制表集合

| |
|-------------------|
| CHCT ₁ |
| ... |
| CHCT _k |
| ... |
| CHCT _m |

| |
|----------------------------|
| 通道标识符(ControllerID) |
| 通道状态：忙/闲 |
| 与通道连接的 设备控制器控制表 的指针 |
| 通道请求队列队首指针 |
| 通道请求队列队尾指针 |

5.5 设备分配

5.5.1 设备分配用数据结构

5.5.2 设备分配考虑因素

5.5.3 设备独立性

5.5.4 独占设备的分配程序

5.5.5 假脱机技术

设备分配考虑因素

- ❑ 设备的固有属性
 - 独占/共享/虚拟设备
- ❑ 设备分配算法
 - 先来先服务/优先级高者优先
- ❑ 设备分配中的安全性
 - 安全分配方式/不安全分配方式
- ❑ 设备独立性

设备分配中的安全性

□ 安全分配方式

- 每当进程发出I/O请求和获得某种设备(资源)后,便进入**阻塞状态**(使其不可能再请求任何资源而在它运行时又不保持任何资源),直到其I/O操作完成时才被唤醒
- 分配安全,但CPU与I/O设备串行工作,进展慢

□ 不安全分配方式

- 进程发出I/O请求后仍继续运行,需要时又可发出第二、第三个I/O请求;仅当进程所请求的设备已被另一进程占用时,进程才进入阻塞状态
- 同一进程可同时操作多台设备,故推进迅速
- 为避免死锁,需进行安全性计算

5.5 设备分配

5.5.1 设备分配用数据结构

5.5.2 设备分配考虑因素


5.5.3 设备独立性

5.5.4 独占设备的分配程序

5.5.5 假脱机技术

设备独立(无关)性概念及目标

□ 概念

- 应用程序独立于具体使用的物理设备
 - 应用程序以逻辑设备名称来请求使用某类设备；而系统实际执行时则使用物理设备名称
- 

□ 目标

- 设备分配时的灵活性、可扩展性/适应性
- 易于实现I/O重定向(指用于I/O操作的设备可以更换即重定向，而不必改变应用程序)

逻辑设备名到物理设备名的映射

□ 逻辑设备表LUT及设置问题

- 当进程用逻辑设备名来请求分配I/O设备时，系统为它分配相应的物理设备，并在逻辑设备表上建立一个表目，填上应用程序中使用的逻辑设备名和系统分配的物理设备名，以及该设备的驱动程序入口地址；当以后进程再利用逻辑设备名请求I/O操作时，系统通过查找逻辑设备表即可找到物理设备和驱动程序
- 两种方式：1、整个系统设置一张逻辑设备表，仅适用于单用户系统；2、为每个用户设置一张逻辑设备表并将该表放入进程的进程控制块，可用于多用户系统中

逻辑设备表LUT

系统设备表

| |
|-----------------|
| 表目 ₁ |
| ... |
| 表目 _i |
| ... |
| 表目 _n |

| |
|-----------------|
| 设备类型(Type) |
| 设备标识符(DeviceID) |
| 设备控制表指针 |
| 设备驱动程序入口 |



| 逻辑设备名 | 物理设备名 | 驱动程序入口地址 |
|--------------|-------|----------|
| /dev/tty | 3 | 1024 |
| /dev/printer | 5 | 2046 |
| ... | ... | ... |

整个系统设置
一张逻辑设备表

| 逻辑设备名 | 系统设备表指针 |
|--------------|---------|
| /dev/tty | 3 |
| /dev/printer | 5 |
| ... | ... |

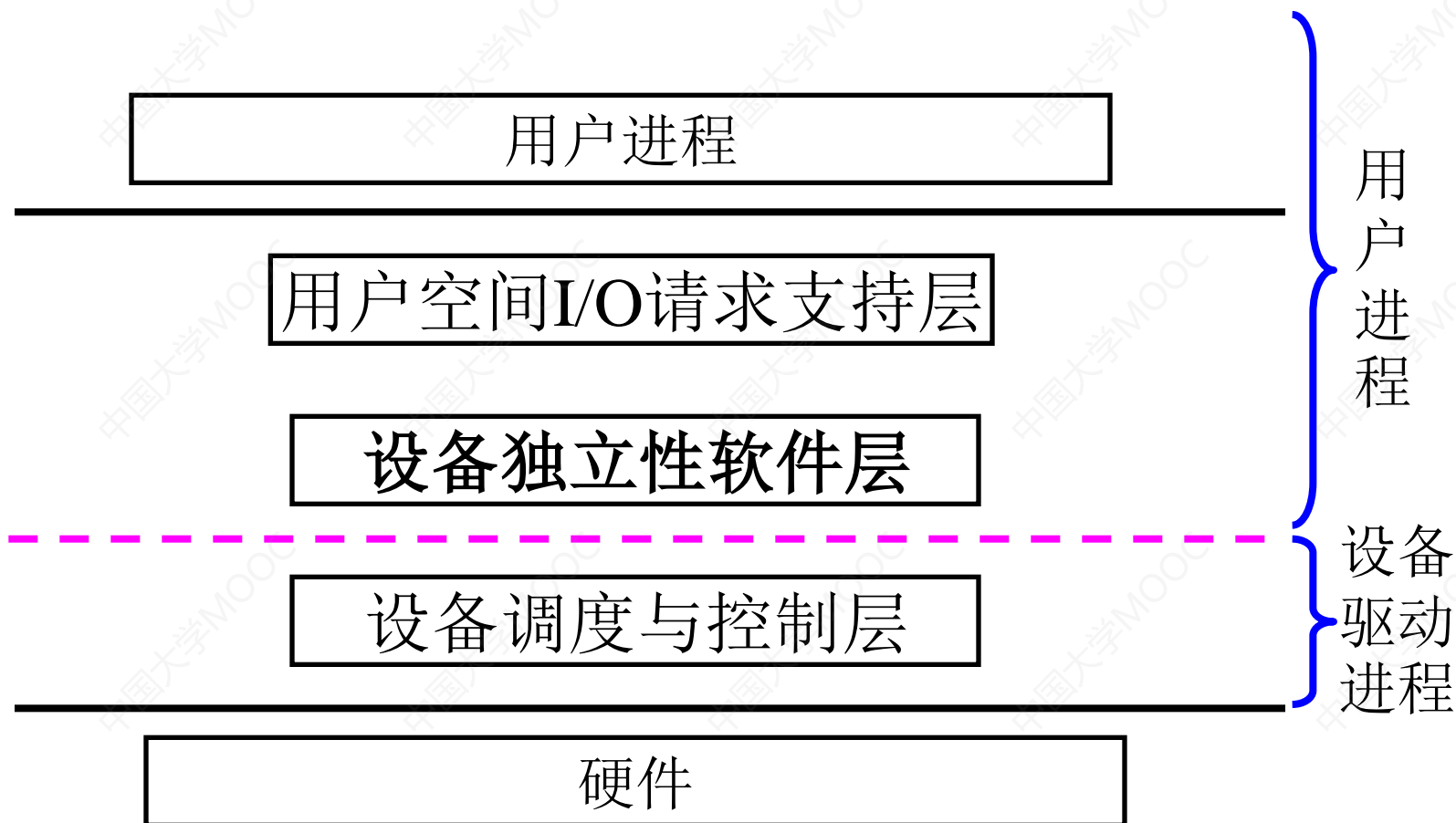
为每个用户设置
一张逻辑设备表

Linux逻辑设备列表

```
stduy-own-kos@ubuntu: ~/kedr20180517/kedr-build
stduy-own-kos@ubuntu:~/kedr20180517/kedr-build$ ls /dev
agpgart      hpet          network_throughput  rtc          tty15        tty38        tty60          ttyS24       vcs6
autofs       hugepages     null                rtc0         tty16        tty39        tty61          ttyS25       vcs7
block        hwrng         port               sda          tty17        tty4         tty62          ttyS26       vcsa
bsg          initctl       ppp               sda1         tty18        tty40        tty63          ttyS27       vcsa1
btrfs-control input         psaux             sda2         tty19        tty41        tty7           ttyS28       vcsa2
bus          kmsg          ptmx              sda5         tty2         tty42        tty8           ttyS29       vcsa3
cdrom        lightnv       pts               sg0          tty20        tty43        tty9           ttyS3         vcsa4
cdrw         log           ram0              sg1          tty21        tty44        ttyprintk      ttyS30       vcsa5
char         loop0         ram1              sg2          tty22        tty45        ttyS0          ttyS31       vcsa6
console      loop1         ram10             shm          tty23        tty46        ttyS1          ttyS4        vcsa7
core         loop2         ram11             snapshot     tty24        tty47        ttyS10         ttyS5        vfio
cpu          loop3         ram12             snd           tty25        tty48        ttyS11         ttyS6        vga_arbiter
cpu_dma_latency loop4         ram13             sr0          tty26        tty49        ttyS12         ttyS7        vhci
cuse         loop5         ram14             sr1          tty27        tty5         ttyS13         ttyS8        vhost-net
disk         loop6         ram15             stderr       tty28        tty50        ttyS14         ttyS9        vmci
dmideid      loop7         ram2              stdin        tty29        tty51        ttyS15         uhid         vsock
dri          loop-control ram3              stdout       tty3         tty52        ttyS16         uinput       zero
dvd          mapper        ram4              tty          tty30        tty53        ttyS17         urandom
ecryptfs     mcelog        ram5              tty0         tty31        tty54        ttyS18         userio
fb0          mem           ram6              tty1         tty32        tty55        ttyS19         vcs
fd           memory_bandwidth ram7              tty10        tty33        tty56        ttyS2         vcs1
fd0          midi          ram8              tty11        tty34        tty57        ttyS20         vcs2
full         nqueue        ram9              tty12        tty35        tty58        ttyS21         vcs3
fuse         net           random            tty13        tty36        tty59        ttyS22         vcs4
hidraw0      network_latency rfkill            tty14        tty37        tty6         ttyS23         vcs5

stduy-own-kos@ubuntu:~/kedr20180517/kedr-build$
```


设备管理层次结构



设备独立性软件

- ❑ 向用户空间I/O请求支持层软件提供统一接口
- ❑ 执行所有设备的公有操作
 - （独占/块）设备的分配与回收
 - 逻辑设备到物理设备的映射→驱动程序
 - 一维逻辑盘块号到一维物理盘块号的转换
 - 缓冲管理（缓冲分配回收、记录成组分解）
 - 设备保护
 - 差错控制（处理设备驱动程序返回错误）

5.5 设备分配

5.5.1 设备分配用数据结构

5.5.2 设备分配考虑因素

5.5.3 设备独立性

5.5.4 独占设备的分配程序

5.5.5 假脱机技术

基本的设备分配程序

- 对于具有I/O通道的系统，在进程提出I/O请求后，系统的设备分配程序可按下述步骤进行设备分配
 - 分配设备
 - 分配设备控制器
 - 分配通道
- ✓ 物理设备名 \Rightarrow SDT \Rightarrow DCT \Rightarrow COCT \Rightarrow CHCT
- ✓ 状态&计算分配安全性 \Rightarrow 分配或插入请求进程队列

设备分配程序的改进

❑ 基本设备分配程序的不足

- 进程以物理设备名提出I/O请求
- 采用单通路I/O系统结构，易产生瓶颈现象

❑ 改进措施

- 进程应以逻辑设备名提出I/O请求，从而增加设备的独立性
 - ❑ 逻辑设备名 \Rightarrow SDT \Leftrightarrow DCT
- 考虑多通路I/O系统结构
 - ❑ 设备控制器或通道分配“失败”需经过多次尝试

5.5 设备分配

5.5.1 设备分配用数据结构

5.5.2 设备分配考虑因素

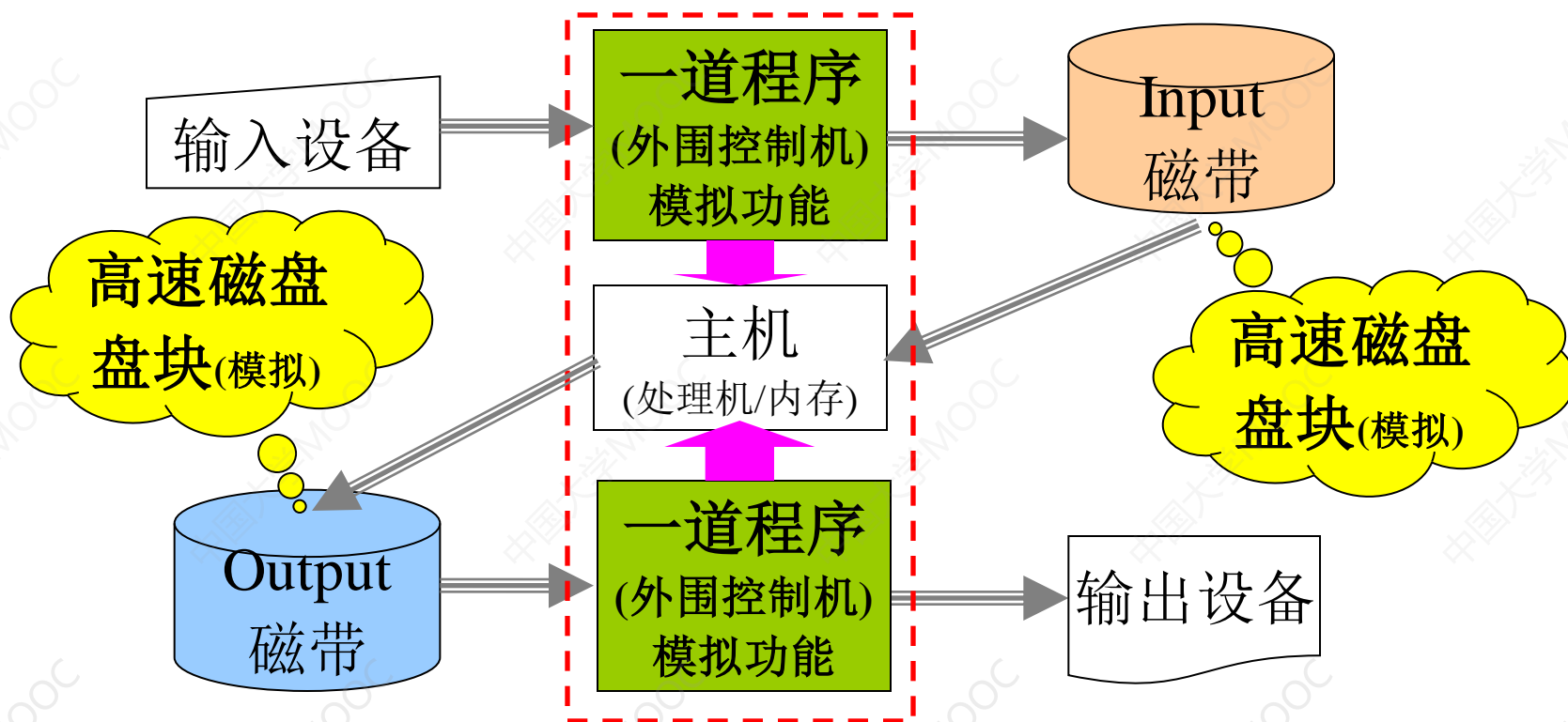
5.5.3 设备独立性

5.5.4 独占设备的分配程序

5.5.5 假脱机技术

假脱机技术(SPOOLing)

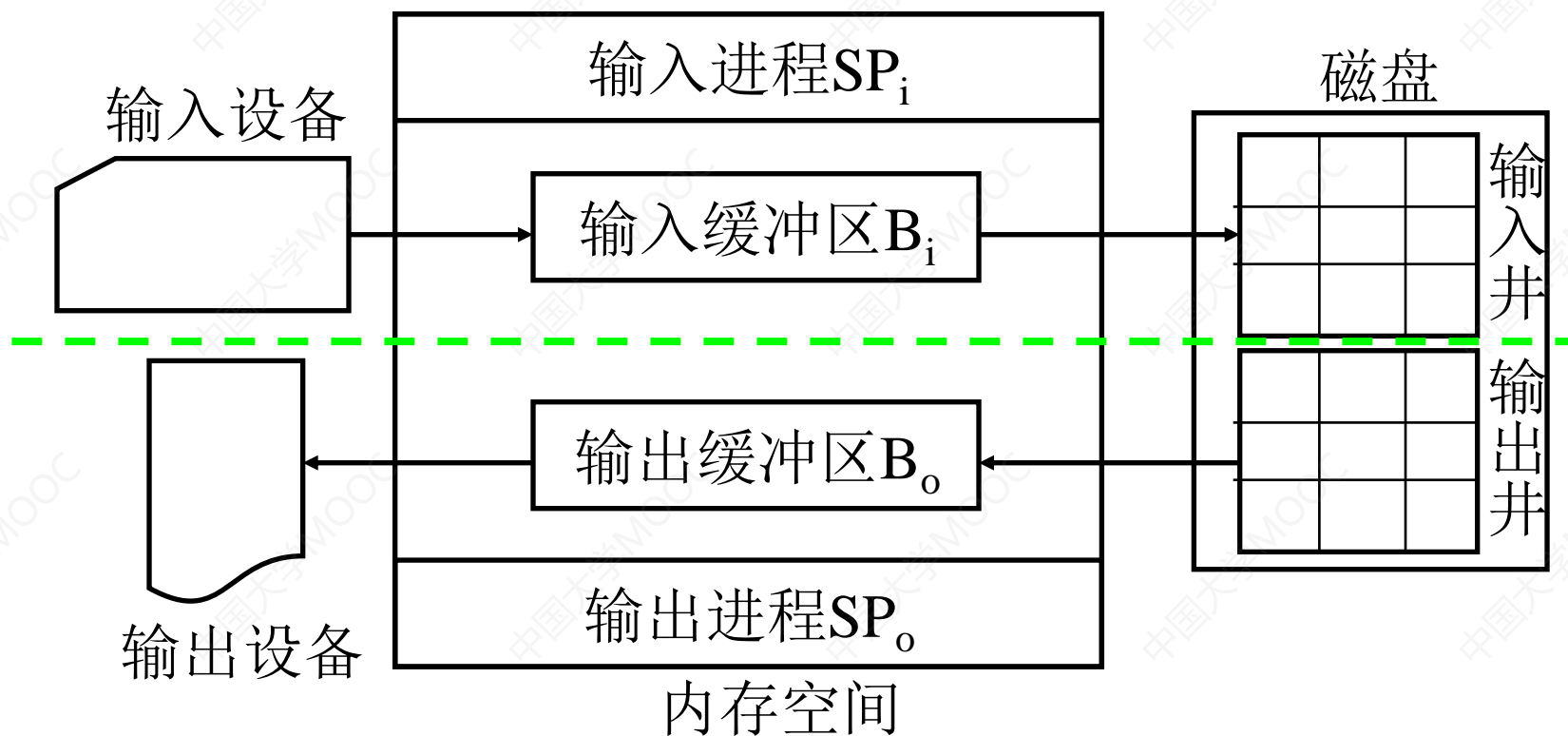
- 将一台独占设备改造成共享设备



SPOOLing系统的组成

- ❖ 建立在具有多道程序功能的操作系统上，且应有高速随机外存（磁盘存储技术）的支持
- 输入井（或输出井）
 - 磁盘上开辟的存储空间用以收容数据
- 输入缓冲区（或输出缓冲区）
 - 内存中开辟的存储空间用以暂存数据
- 输入进程 SP_i （或输出进程 SP_o ）
 - 模拟脱机输入（或输出）时的外围控制机
- 请求输入队列（或请求输出队列）
 - 由用户请求输入（或输出）表构成

SPOOLing系统组成示意图



共享打印机

□ 广泛应用于多用户系统和局域网络中

- 1、用户进程提出打印输出请求
- 2、输出进程在输出井中为之申请一空闲盘块区，并将要打印的数据送入其中
- 3、输出进程为用户进程申请一张空白的用户请求打印表，并将用户打印要求填入其中和把该表挂到**请求打印队列**上
- 4、输出进程视打印机空闲与否从请求打印队列队首取出一张用户请求打印表，并按对应要求将打印数据从输出井传送到内存缓冲区和打印
- 5、输出进程在请求打印队列为空时将阻塞自己，直到再次有打印请求出现时才被唤醒

SPOOLing系统的特点

□ 提高了I/O速度

- 数据I/O操作演化为对输入/出井的存取
- CPU数据处理与设备I/O操作的并行化

□ 将独占设备改造为共享设备

- 设备分配实质为在输入/出井中为用户进程分配一空闲盘块区及建立一张I/O请求表

□ 实现了虚拟设备功能

- 一台独占设备变换为若干台逻辑独占设备

5.5 设备分配

5.5.1 设备分配用数据结构

5.5.2 设备分配考虑因素

5.5.3 设备独立性

5.5.4 独占设备的分配程序

5.5.5 假脱机技术

作业题

- **5.7** 谈谈你对设备分配的关键问题和解决方案的认识与理解，特别提请关注相应数据结构的设计、设备独立性、分配策略与分配算法及假脱机技术等。

重点问题解析

——内存管理地址映射

64位微处理器内存管理系统

- 支持分段和分页机制（段页式）
 - 内存按字节编址
 - 段号采用26位编址
 - 分页大小为16KB
 - 段表项和页表项均为32位
 - 段表和页表均基于分页策略进行存放
 - TLB可包含64项

| | | | | | |
|------|-------|-------|-------|-------|---|
| 63 | 50 49 | 38 37 | 26 25 | 14 13 | 0 |
| 二级段号 | 一级段号 | 二级页号 | 一级页号 | 页内偏址 | |
| 14位 | 12位 | 12位 | 12位 | 14位 | |

请求分页内存管理地址转换A

□ 有效地址为32个比特位、页长8KB

➤ 进程地址空间大小？ 4GB

➤ 页表长度（页表项数）？ 512K

➤ 若处理器给出的有效地址ABCDE（16进制）送内存地址变换机构，地址变换机构将分离出页号（16进制）为：0x55

如果该页所对应的物理块号为222（十进制），则对应物理地址（16进制）为：

0x1BDCDE

请求分页内存管理地址转换B

- 基于32位微处理器的计算机系统，内存按字节编址，物理块大小为4K字节，页表项占32个二进制位

| | | | | | | | | | | | | | | | | |
|------|-----|-----|-----|-----|-----|-----|---|-----|---|-----|----|-----|----|----|----|-----|
| 页号 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| 物理块号 | 411 | 520 | 601 | 712 | 325 | 456 | — | 656 | — | 312 | — | 566 | — | — | — | 760 |
| 状态位 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 |

- 假定进程P的页表部分内容参表所示（其中数值均为十进制；状态位为1表示在内存）
进程P的逻辑地址所对应物理地址（16进制）：

0000AE25H ⇔ 页失效/缺页中断

第五章 设备管理

5.1 I/O系统组成

5.2 I/O控制方式

5.3 设备管理目标、功能及结构

5.4 缓冲管理

5.5 设备分配

5.6 设备处理

5.7 磁盘存储器管理

5.6 设备处理

5.6.1 设备处理程序的功能

5.6.2 设备处理方式

5.6.3 设备处理程序的特点

5.6.4 设备驱动过程

5.6.5 中断处理过程

设备处理程序的功能

- 1、接收由I/O请求进程发来的命令和参数，并将命令中的抽象要求转化为具体要求
- 2、检查用户I/O请求的合法性
- 3、检查I/O设备状态，若忙则挂在设备队列上等待
- 4、传递和设置I/O设备的有关参数与工作方式，包括根据用户I/O请求构造必要的通道程序
- 5、发出I/O命令，启动分配到的I/O设备去完成指定的I/O操作
- 6、及时响应来自通道或设备控制器的中断请求，并根据其中断类型调用相应中断处理程序进行处理

5.6 设备处理

5.6.1 设备处理程序的功能

5.6.2 设备处理方式

5.6.3 设备处理程序的特点

5.6.4 设备驱动过程

5.6.5 中断处理过程

设备处理方式

□ 根据设备处理时是否设置进程及设置什么样的进程而划分

唤醒问题

- 为每一类设备设置一个进程
- 整个系统设置一个I/O进程或设置一个输入进程和一个输出进程
- 不设置专门的设备处理进程，而是只为各类设备设置相应的设备处理程序（模块）供设备独立性软件和中断处理程序调用

5.6 设备处理

5.6.1 设备处理程序的功能

5.6.2 设备处理方式

5.6.3 设备处理程序的特点

5.6.4 设备驱动过程

5.6.5 中断处理过程

设备处理程序的特点

- ❑ 用于I/O请求进程与设备控制器间通信
 - I/O请求进程 \Leftrightarrow 设备控制器
- ❑ 与I/O设备特性紧密相关
 - 不同类型设备应配置不同驱动程序
- ❑ 与I/O控制方式紧密相关
 - 中断/DMA/通道I/O控制方式
- ❑ 与硬件紧密相关，故部分程序代码必须用汇编语言书写，且基本部分可固化和存放于ROM

5.6 设备处理

5.6.1 设备处理程序的功能

5.6.2 设备处理方式

5.6.3 设备处理程序的特点

5.6.4 设备驱动过程

5.6.5 中断处理过程

设备驱动过程

- 1、化抽象I/O操作请求为具体要求
- 2、检查用户I/O请求的合法性
- 3、检查I/O设备状态，判断是否可用
- 4、传递和设置I/O设备的有关参数与工作方式，包括根据用户I/O请求构造必要的通道程序
- 5、发出I/O命令，启动分配到的I/O设备
 - ❖ 基本的I/O操作是在设备控制器或通道的控制下进行的，鉴于通常I/O操作所要完成的工作较多且需要一定时间，故此时驱动（程序）进程应阻塞自己直至中断到来时才将它唤醒

5.6 设备处理

5.6.1 设备处理程序的功能

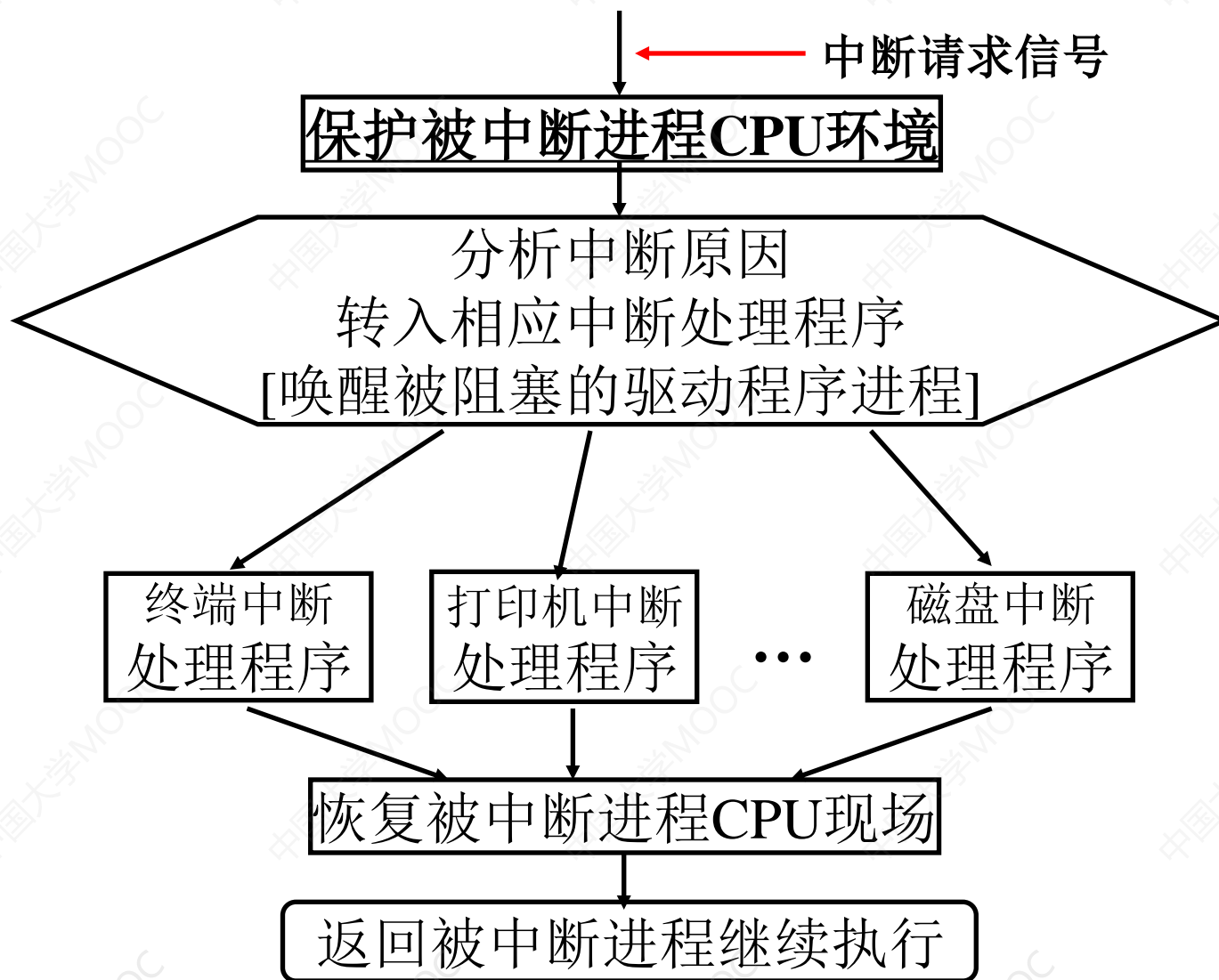
5.6.2 设备处理方式

5.6.3 设备处理程序的特点

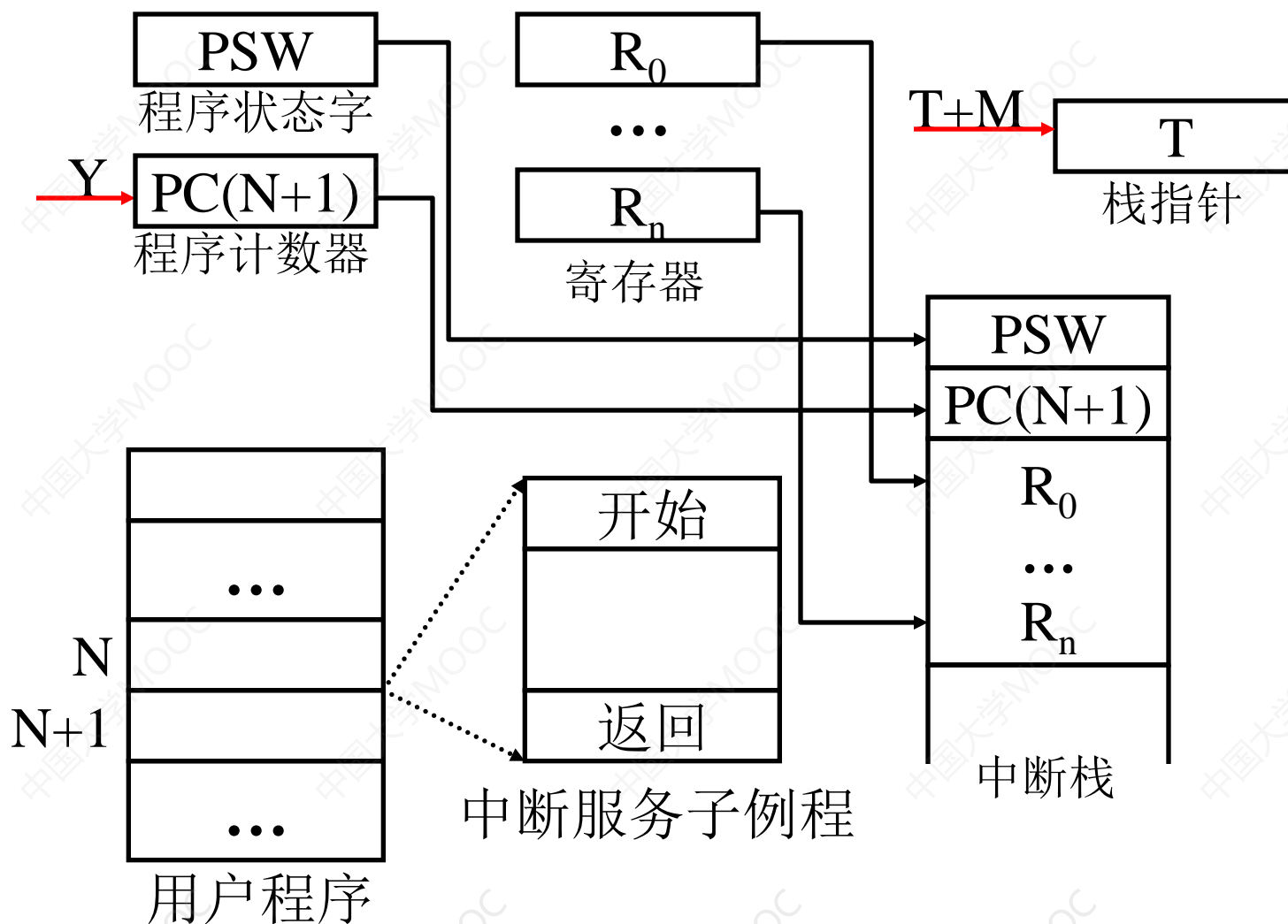
5.6.4 设备驱动过程

5.6.5 中断处理过程

中断处理程序的处理过程



中断现场保护示意图



5.6 设备处理

5.6.1 设备处理程序的功能

5.6.2 设备处理方式

5.6.3 设备处理程序的特点

5.6.4 设备驱动过程

5.6.5 中断处理过程

作业题

- 5.8 谈谈你对设备处理程序的功能、特点、实现方式及设备驱动进程与中断处理过程的认识与理解。

实验课题12

Linux 设备驱动程序设计与实现：✚

- I. 基于 **Linux** 设计和实现一种设备驱动程序，并启用调试和运行，深入领会和理解相应操作系统设备管理的体系结构以及设备驱动程序的设计框架、工作机理和设计要领。✚
- II. 撰写实验报告，阐述开发/运行/测试环境、实验步骤、技术难点及解决方案、关键数据结构和算法流程、编译运行测试过程及结果截图、结论与体会等；✚
- III. 要求提交实验报告、源程序、可执行程序及 Makefile 文件（如果有的话）。✚
- IV. 可选提交材料（动画或视频演示）✚

实验课题13

Windows 设备驱动程序设计与实现：✚

- I. 基于 Windows 设计和实现一种设备驱动程序，并启用调试和运行，深入领会和理解相应操作系统设备管理的体系结构以及设备驱动程序的设计框架、工作机理和设计要领。✚
- II. 撰写实验报告，阐述开发/运行/测试环境、实验步骤、技术难点及解决方案、关键数据结构和算法流程、编译运行测试过程及结果截图、结论与体会等；✚
- III. 要求提交实验报告、源程序、可执行程序及 Makefile 文件（如果有的话）。✚
- IV. 可选提交材料（动画或视频演示）✚

实验课题14

Linux 设备驱动程序设计探析：✦

- I. 研读 Linux 内核源码（任意版本均可），探析 Linux 设备管理体系结构及特定设备驱动程序的设计框架、工作机理和设计实现要领。✦
- II. 提交对应源码分析报告、Linux 内核源码及相关参考资料。
- III. 可选提交材料（动画或视频演示）✦

第五章 设备管理

5.1 I/O系统组成

5.2 I/O控制方式

5.3 设备管理目标、功能及结构

5.4 缓冲管理

5.5 设备分配

5.6 设备处理

5.7 磁盘存储器管理

5.7 磁盘存储器管理

5.7.1 磁盘性能简述

5.7.2 磁盘调度

5.7.3 磁盘高速缓冲

5.7.4 提高磁盘I/O速度的其它方法

5.7.5 廉价磁盘冗余阵列

磁盘存储器及管理任务

□ 磁盘存储器

- 容量大、存取速度快，且可实现**随机存取**
- 实现**虚拟存储器/虚拟设备**的必需硬件
- 存放（程序和数据）文件的主要外存设备

□ 磁盘存储器管理的主要任务

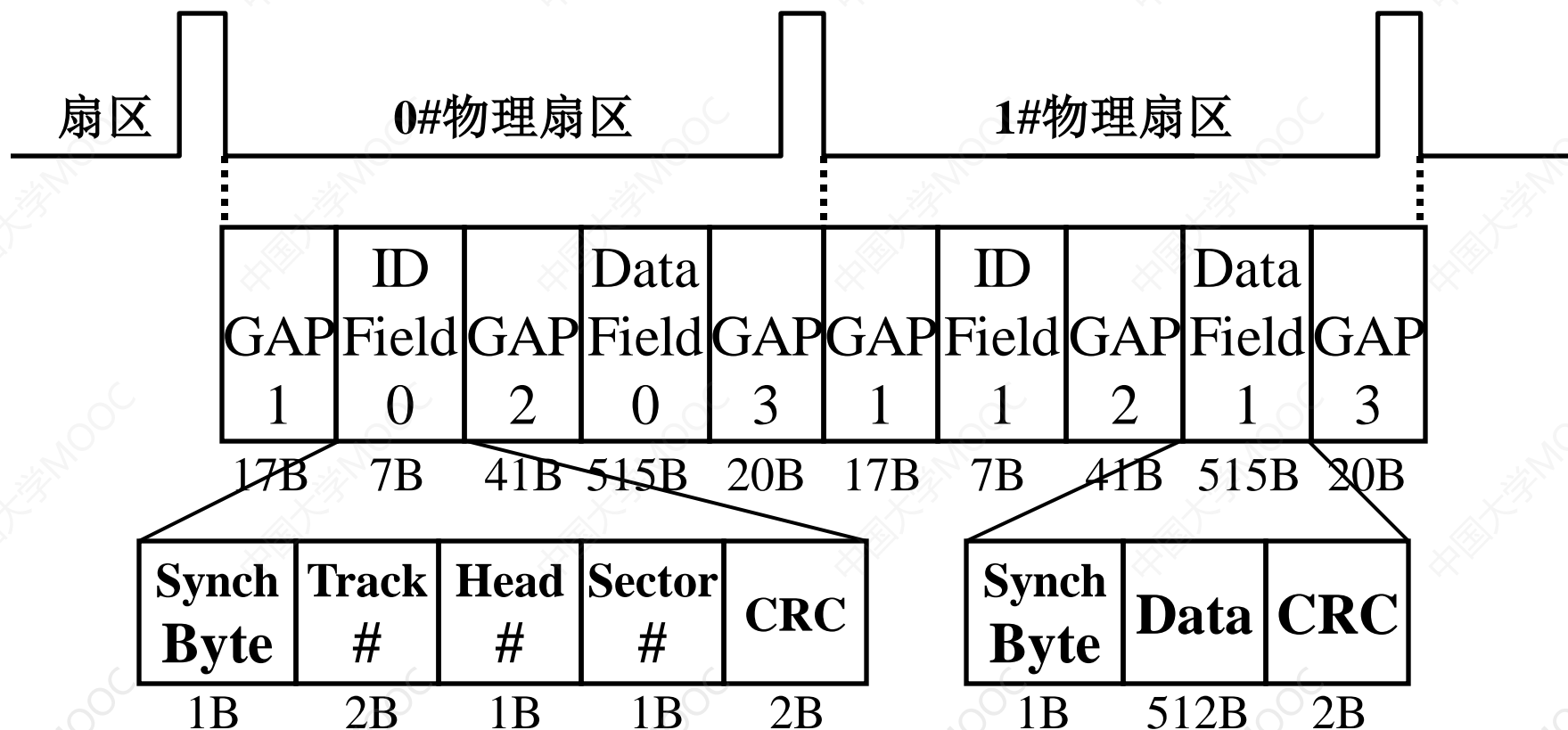
- 为文件分配必要的存储空间，使各得其所
- 提高磁盘存储空间的利用率
- 合理组织**文件结构**，以提高文件访问速度
- 提高磁盘**I/O**速度，以改善文件系统性能
- 采取必要的冗余措施，确保文件系统的可靠性

数据

磁头号：柱面号：扇区号

基本数据组织层次

➤ 盘片（[盘面]）：磁道：扇区（盘块）



磁盘的类型

固态磁盘？

- ❑ 硬盘和软盘、单片盘和多片盘

- ❑ 固定头磁盘

- 在每条磁道上都有一个读写磁头，所有磁头装在一刚性磁臂上，通过这些磁臂可访问所有的磁道，并进行并行读写，能有效提高磁盘I/O速度，这种结构主要用于大容量磁盘

- ❑ 活动头（移动头）磁盘

- 每个盘面仅配有一个读写磁头，也被装入磁臂中，为了能访问该盘面上的所有磁道，该磁头必须能够移动和进行寻道；移动头磁盘结构简单，但只能进行串行读写故而I/O速度较慢，多用于中小型磁盘设备中

$$\frac{\frac{1}{r} \times \sum_{i=0}^{n-1} \frac{i}{n}}{n} = \frac{\frac{1}{r} \times \frac{(0 + \frac{n-1}{n}) \times n}{2}}{n} = \frac{1}{2r}$$

□ 旋转延迟时间 T_r

- 指定扇区旋转到磁头下面所经历的时间 $T_r = 1/2r$

□ 数据传输时间 T_t

- 把数据从磁盘读出或向磁盘写入数据所经历的时间 $T_t = \text{bytes} / (r \times \text{bytesPerTrack})$

- ❖ 访问时间中，数据传输时间所占比例相当小，而寻道时间和旋转延迟时间基本上均与所读写的数据量无关，所以适当地集中数据传输（不要太零散）将有利于提高传输效率

5.7 磁盘存储器管理

5.7.1 磁盘性能简述

5.7.2 磁盘调度

5.7.3 磁盘高速缓冲

5.7.4 提高磁盘I/O速度的其它方法

5.7.5 廉价磁盘冗余阵列

磁盘调度目标及算法

□ 磁盘调度目标

- 磁盘是能被多个进程共享的设备，当有多个进程请求访问磁盘时，应采用一种适当的调度算法，使各进程对磁盘的平均访问时间（主要是寻道时间）最小

□ 常用的磁盘调度算法

- 先来先服务调度算法FCFS
- 最短寻道时间优先调度算法SSTF
- 扫描算法、循环扫描算法

先来先服务调度算法

□ 基本思想

- 根据进程请求访问磁盘的先后次序进行调度，其优点是公平、简单且每个进程的请求都能依次得到处理，但寻道时间可能较长

□ 磁盘I/O请求队列示例（磁头当前位置100#磁道，且正向磁道号增加方向移动）

- 55、58、39、18、90、160、150、38、184

| | | | | | | | | | | |
|------|----|----|----|----|----|-----|-----|-----|-----|------|
| 访问次序 | 55 | 58 | 39 | 18 | 90 | 160 | 150 | 38 | 184 | 平均 |
| 移动距离 | 45 | 3 | 19 | 21 | 72 | 70 | 10 | 112 | 146 | 55.3 |

最短寻道时间优先调度算法

□ 基本思想

- 选择所访问磁道与磁头当前所在磁道距离最近的进程优先调度，但不能保证平均寻道时间最短
- 具较好的寻道性能，但可能导致进程饥饿现象

□ 磁盘I/O请求队列示例（磁头当前位置100#磁道，且正向磁道号增加方向移动）

- 55、58、39、18、90、160、150、38、184

| | | | | | | | | | | |
|------|----|----|----|----|----|----|-----|-----|-----|------|
| 访问次序 | 90 | 58 | 55 | 39 | 38 | 18 | 150 | 160 | 184 | 平均 |
| 移动距离 | 10 | 32 | 3 | 16 | 1 | 20 | 132 | 10 | 24 | 27.5 |

扫描算法（电梯调度算法）SCAN

□ 基本思想

- 不仅考虑欲访问磁道与磁头当前所在磁道的间距，更优先考虑的是磁头当前移动的方向
- 既能获得较好的寻道性，又能防止进程饥饿，广泛用于大中小型机及网络中

□ 磁盘I/O请求队列示例（磁头当前位置100#磁道，且正向磁道号增加方向移动）

- 55、58、39、18、90、160、150、38、184

| | | | | | | | | | | |
|------|-----|-----|-----|----|----|----|----|----|----|------|
| 访问次序 | 150 | 160 | 184 | 90 | 58 | 55 | 39 | 38 | 18 | 平均 |
| 移动距离 | 50 | 10 | 24 | 94 | 32 | 3 | 16 | 1 | 20 | 27.8 |

循环扫描算法CSCAN

□ 基本思想

- 规定磁头单向移动，避免某些进程磁盘请求的严重延迟

□ 磁盘I/O请求队列示例（磁头当前位置100#磁道，且正向磁道号增加方向移动）

- 55、58、39、18、90、160、150、38、184

| | | | | | | | | | | |
|------|-----|-----|-----|-----|----|----|----|----|----|------|
| 访问次序 | 150 | 160 | 184 | 18 | 38 | 39 | 55 | 58 | 90 | 平均 |
| 移动距离 | 50 | 10 | 24 | 166 | 20 | 1 | 16 | 3 | 32 | 35.8 |

N-步扫描算法与FSCAN算法

□ N-步扫描算法（N-Step-SCAN）

- ❖ 克服前述调度算法均具的磁臂粘着现象即磁臂停留在某处不动的情况（特别是高密度磁盘）
- 将磁盘请求队列分成若干个长度为N的子队列，按先来先服务算法依次处理这些子队列，而各队列以扫描算法处理（新请求放入后面队列）

□ FSCAN算法

- 实质为N-步扫描算法的简化，将磁盘请求队列分成两个子队列：① 当前所有请求磁盘I/O的进程形成的队列，按扫描算法处理；② 在扫描期间新出现的所有请求磁盘I/O进程形成的等待队列，本次扫描结束后②添加到①的队尾，从而使所有新要求都被推迟到下一次扫描时处理

5.7 磁盘存储器管理

5.7.1 磁盘性能简述

5.7.2 磁盘调度

5.7.3 磁盘高速缓冲

5.7.4 提高磁盘I/O速度的其它方法

5.7.5 廉价磁盘冗余阵列

磁盘高速缓冲的形式

□ 磁盘高速缓冲的概念

- 并非通常意义下在内存和CPU之间增设的小容量高速存储器，而是指利用内存中的存储空间来暂存从磁盘中读出的一系列盘块中的信息，因此，其为一组在逻辑上属于磁盘而物理上驻留在内存的盘块

□ 磁盘高速缓冲在内存的形式

- A. 开辟单独的固定大小的空间
- B. 与虚拟存储器共享内存缓冲池

磁盘高速缓冲关键环节

❑ 数据交付

- 将磁盘高速缓冲中的数据传送给请求者进程
- 交付方式：数据/指针

❑ 置换算法（同页面置换算法相比较）

- 最近最久未使用置换算法LRU
- 考虑访问频率、可预见性及数据一致性原则
- LRU链的链头与链尾策略

❑ 周期性写回磁盘

- 防止经常被访问的盘块数据一直存放在高速缓冲不被写回磁盘而可能随故障发生丢失
- 写穿透高速缓存Write-Through Cache

5.7 磁盘存储器管理

5.7.1 磁盘性能简述

5.7.2 磁盘调度

5.7.3 磁盘高速缓冲

5.7.4 提高磁盘I/O速度的其它方法

5.7.5 廉价磁盘冗余阵列

磁盘I/O速度提高的其它方法

□ 提前读

- 用户（进程）对文件访问常采用顺序访问方式，即顺序地访问文件各盘块的数据，所以在读入当前块的同时，可将下一盘块数据提前读入缓冲区

□ 延迟写

- 指缓冲区中的数据本应立即写回磁盘，但考虑到它们不久之后可能还会再被本进程或其它进程访问，因而并不立即将其写入磁盘，而是将它挂在空闲缓冲区队列的末尾

□ 虚拟盘（RAM盘）

- 指利用内存空间仿真磁盘，常用于存放临时文件
- 与磁盘高速缓冲区别：内容由用户/OS控制

数据分布优化

□ 优化物理块的分布

- 尽可能把属于同一个（索引或链接）文件的数据盘块安排在同一条磁道或相邻的磁道上，使磁头移动距离最小
- 位示图或以簇为单位进行分配

□ 优化索引结点的分布

- 将磁盘上所有磁道分成若干组，在每组中都含有索引结点、盘块和空闲盘块表，以保证索引结点盘块与存放文件的数据盘块间距离最小

5.7 磁盘存储器管理

5.7.1 磁盘性能简述

5.7.2 磁盘调度

5.7.3 磁盘高速缓冲

5.7.4 提高磁盘I/O速度的其它方法

5.7.5 廉价磁盘冗余阵列

廉价磁盘冗余阵列RAID

- ❑ 利用一台磁盘阵列控制器来统一管理和控制一组磁盘驱动器，组成一个高度可靠的、快速的大容量磁盘系统，分为0-7级
- ❑ 可靠性高、磁盘I/O速度高、性能/价格比高
- ❑ 并行交叉存取
 - 每一盘块数据分为若干个子盘块数据，并把每个子盘块数据分别存储到各个不同磁盘中的相同位置；以后一个盘块数据向内存传送时采用并行传输方式

5.7 磁盘存储器管理

5.7.1 磁盘性能简述

5.7.2 磁盘调度

5.7.3 磁盘高速缓冲

5.7.4 提高磁盘I/O速度的其它方法

5.7.5 廉价磁盘冗余阵列

第五章 设备管理

5.1 I/O系统组成

5.2 I/O控制方式

5.3 设备管理目标、功能及结构

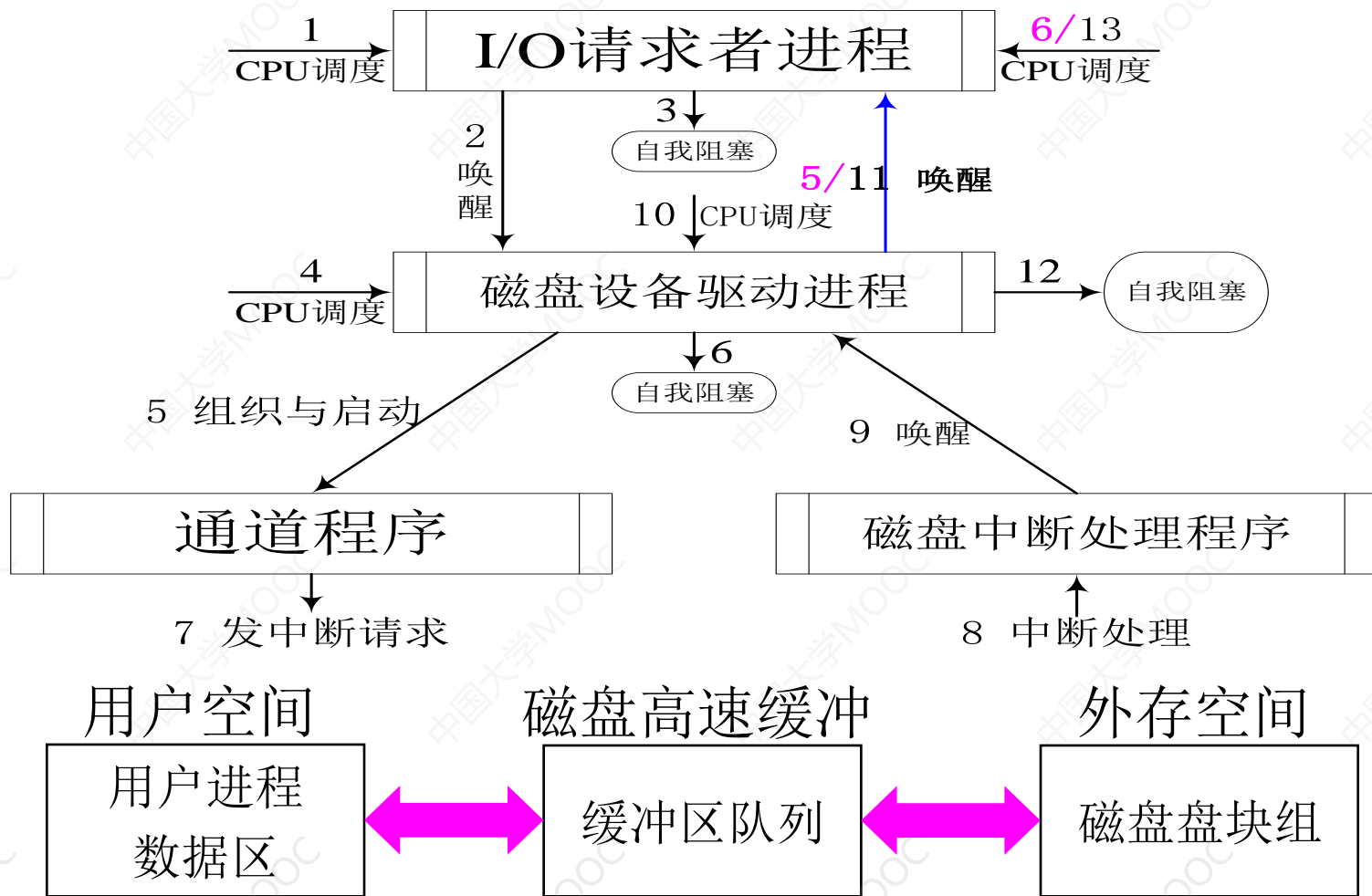
5.4 缓冲管理

5.5 设备分配

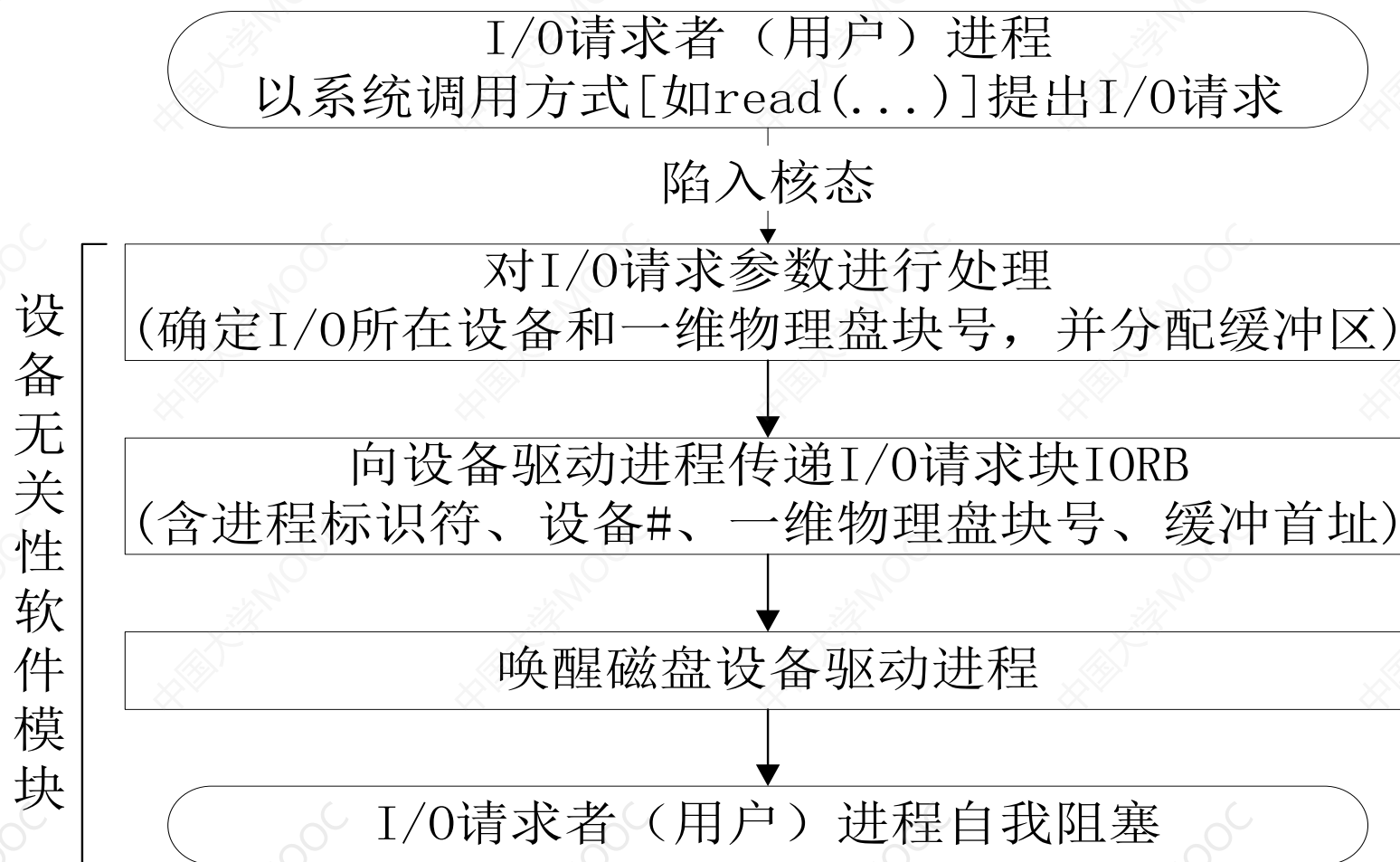
5.6 设备处理

5.7 磁盘存储器管理

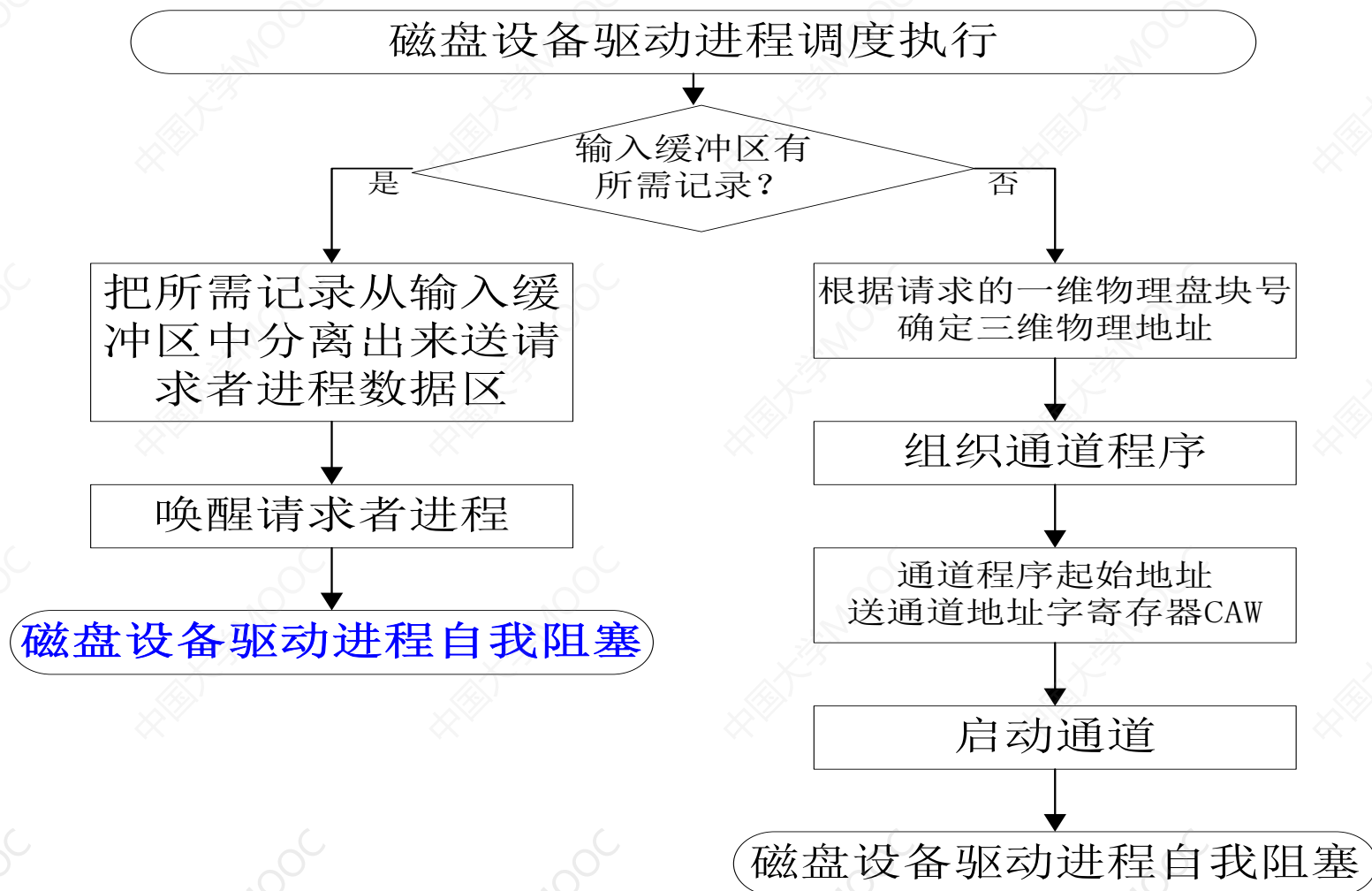
补充内容：磁盘I/O全过程剖析



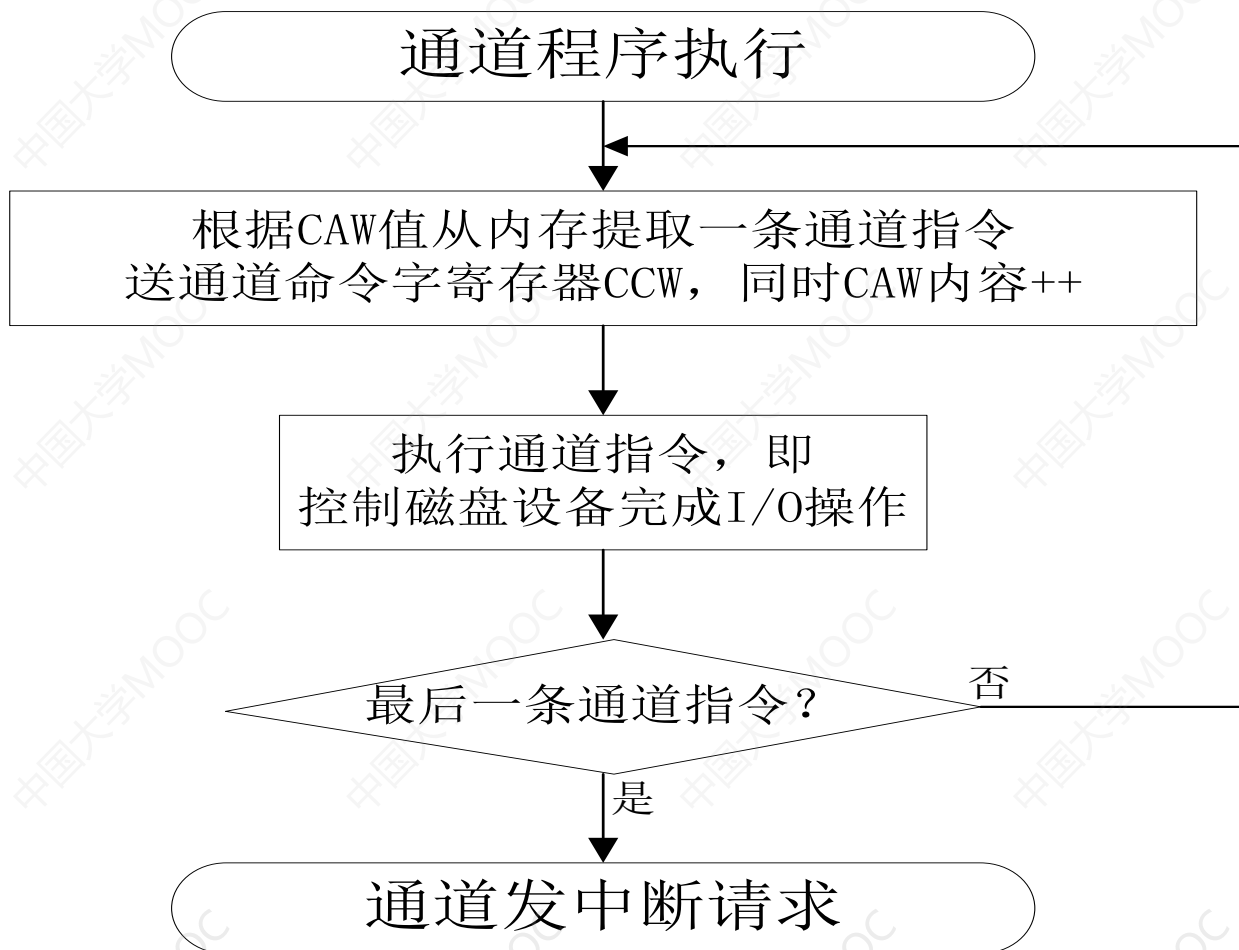
I/O请求者进程（上）



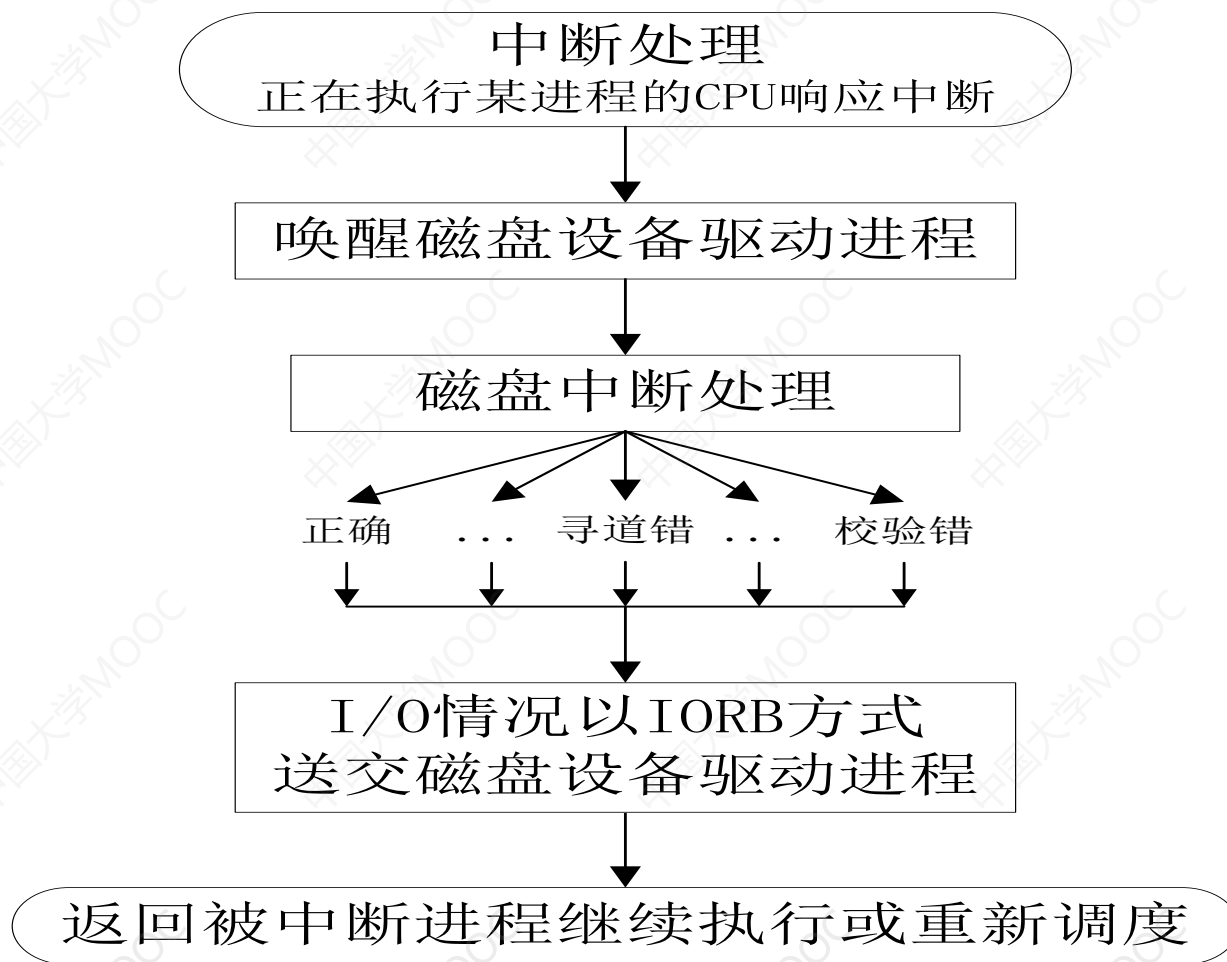
磁盘设备驱动进程 (上)



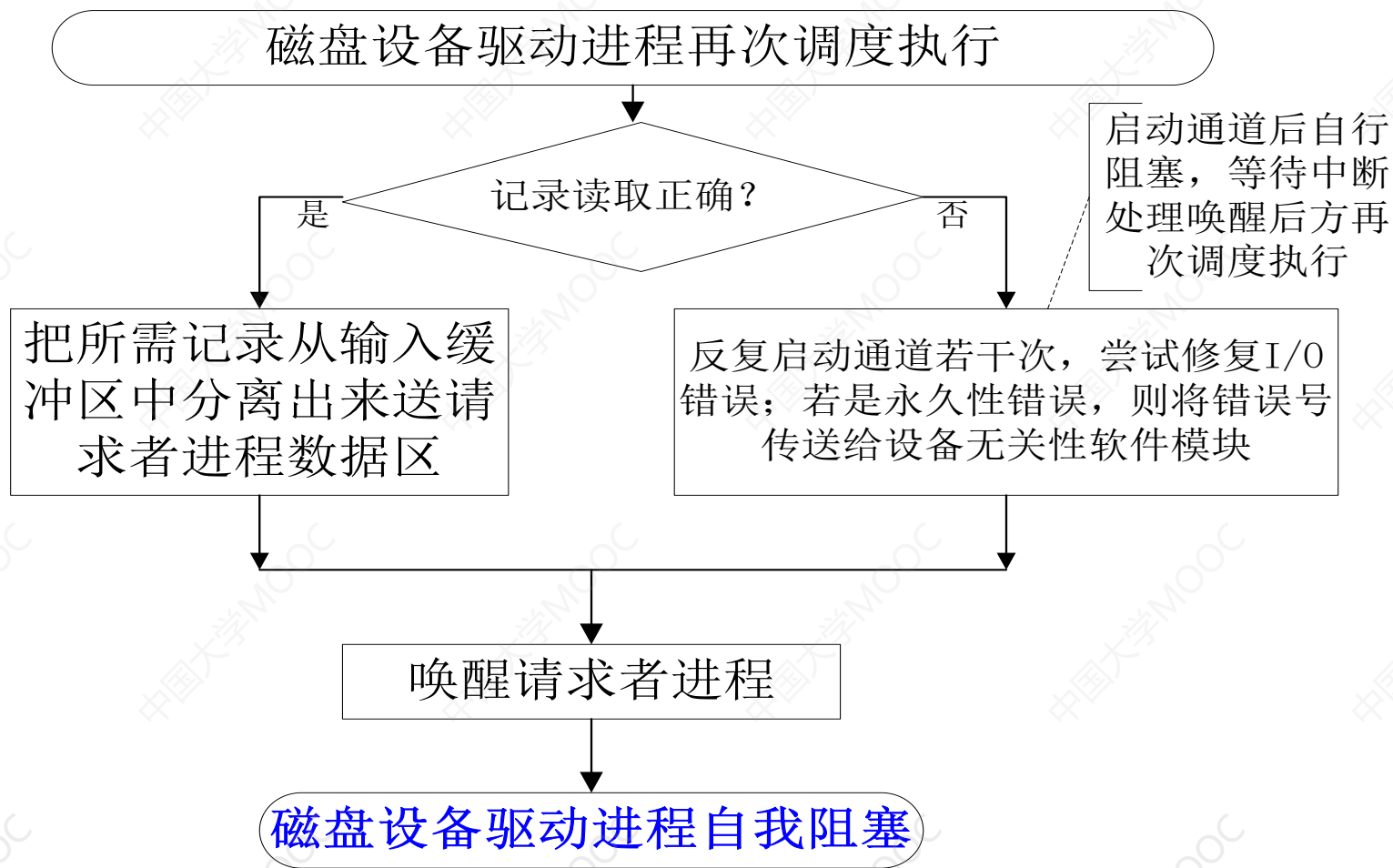
通道程序



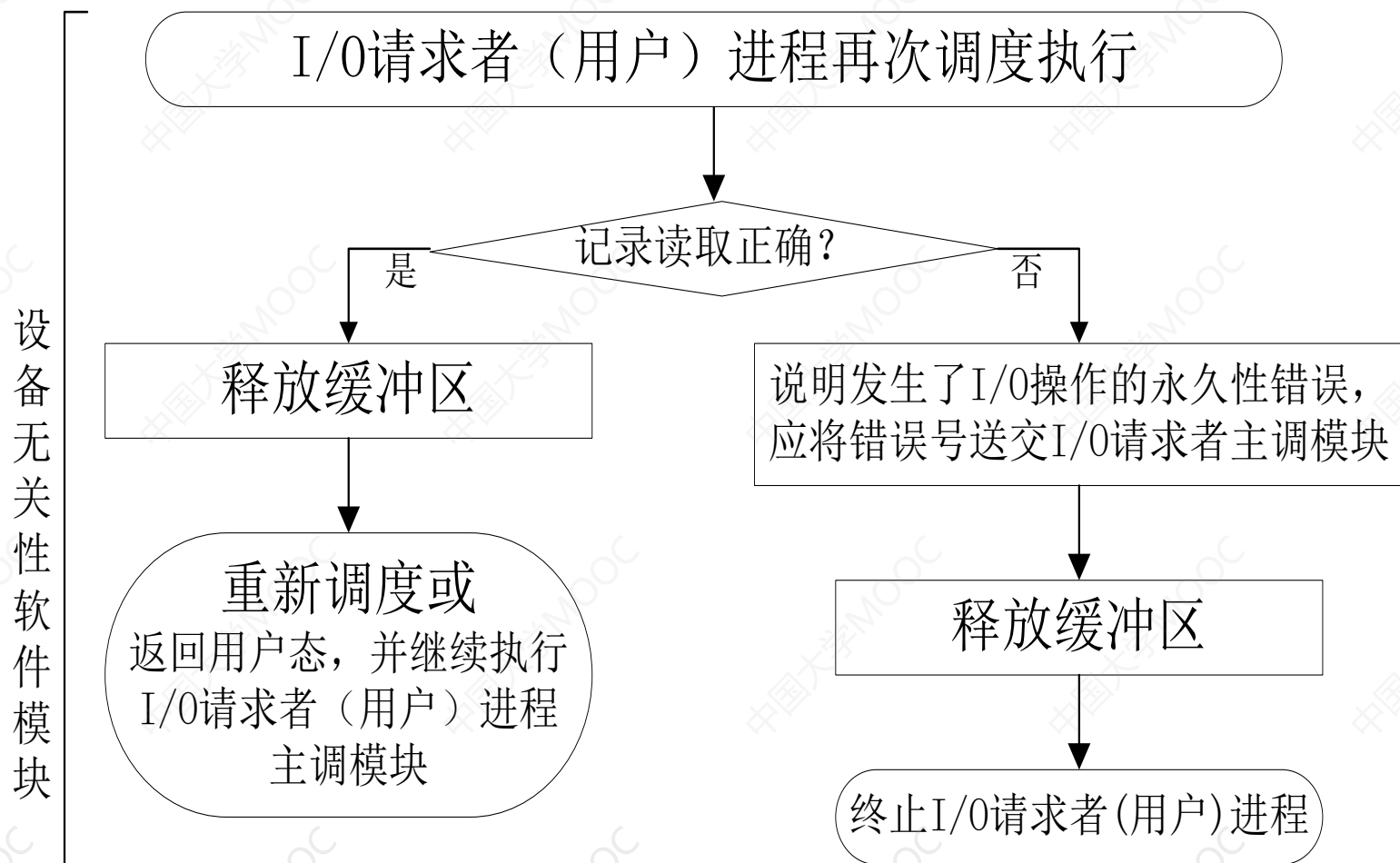
中断处理



磁盘设备驱动进程（下）



I/O请求者进程（下）



作业题

- **5.9** 谈谈你对磁盘设备管理的基本任务和磁盘访问及速度提高、磁盘调度算法、高速缓冲等关键问题与技术要领的认识与理解。
- **5.10** 谈谈你对从一个用户进程提出磁盘输入输出请求到完成该请求的整个过程的认识与理解。



**同学们，
再见！**

2021年4月27日星期二