华中农业大学
HUAZHONG AGRICULTURAL UNIVERSITY

计算机操作系统

倪福川
fcni_cn@mail.hzau.edu.cn

华中农业大学 信息学院

第五章　虚拟存储器

5.1 虚拟存储器概述

5.2 请求分页存储管理方式

5.3 页面置换算法

5.4 抖动与工作集

5.5 请求分段存储管理方式

5.1 虚拟存储器概述

P153

虚拟存储器的引入

虚拟存储器的实现方法

虚拟存储器的特征

5.1.1、虚拟存储器的引入

1、内存空间的限制

常规存储器管理特点：一次性、驻留性

一：内存空间装不下的大作业无法运行

二：作业量大时，无法允许更多的作业并发

2、局部性原理

1968，P.Denning:在一较短的时间内，程序的执行仅限于某个部分；它所访问的存储空间也局限于某个区域。

提出的论点：

（1）程序的顺序执行

（2）过程调用深度有限

（3）程序中的循环结构

（4）程序中有许多对数据结构的处理

局部性还表现在：

（1）时间局限性

（2）空间局限性

- **3、虚拟存储器的定义**
  - **虚拟存储器是指具有请求调入功能和置换功能，能从逻辑上对内存容量加以扩充的一种存储器系统。**

物理上不存在，利用海量外存进行内存“空间”扩展。

允许作业部分装入，需要时再临时装入所需的部分。直到作业退出，某些部分也有可能没被装入过。

5.1.2、虚拟存储器的实现方法

须建立在离散分配的内存管理技术基础上。

1、请求分页系统

基本分页系统+请求调页功能+页面置换功能

＝页式虚拟存储系统

硬/软件支持：

- 请求分页的页表机制、
- 缺页中断机构、
- 动态地址变换机构。

- 2、请求分段系统
  - 基本分段系统 + 请求调段功能 + 分段置换功能
    ＝段式虚拟存储系统
  - 硬/软件支持：
    请求分段的段表机制、
    缺段中断机构、
    动态地址变换机构。

5.1.3、虚拟存储器的特征

多次性

一个作业被分成多次调入内存运行；

对换性

允许在作业的运行过程中进行换进、换出；

虚拟性

逻辑扩充内存，使用户“看到”的内存容量远大于实际大小。

该特征是以上两个特征为基础的。

5.2 请求分页存储管理方式

请求分页中的硬件支持

内存分配策略和分配算法

请求分页策略

5.2.1、请求分页中的硬件支持

1、页表机制

用于地址转换；增加页表项：

页号	物理块号	状态位P	访问字段A	修改位M	外存地址

状态位P：用于指示该页是否已调入内存
访问字段A：记录本页在一段时间内被访问的次数
修改位M：该页在调入内存后是否被修改过
外存地址：指示该页在外存上的地址

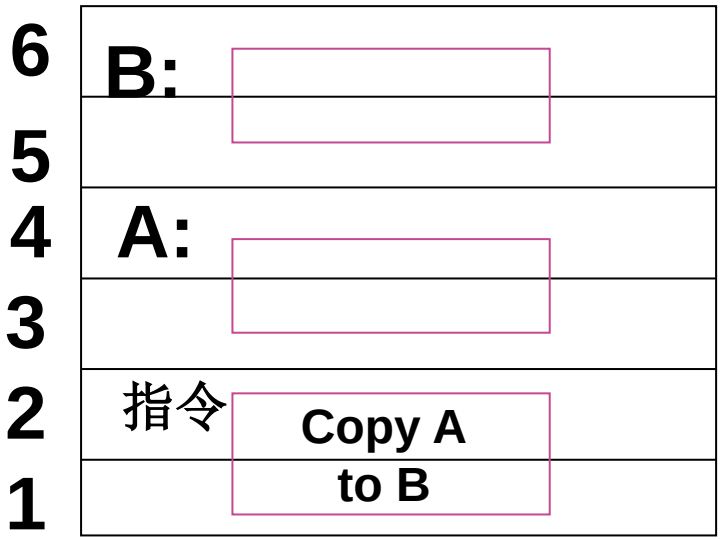
P158

2、缺页中断机构

- 所要访问的页不在内存时，便引发一次缺页中断

缺页中断与其他中断的不同：

- 在指令执行期间产生和处理中断信号
- 一条指令在执行期间可能产生多次缺页中断



12

- 3、地址变换机构
  - 情况一： 首先检索快表，若找到，修改页表项中的访问位，然后利用页表项中给出的物理块号和页内地址，形成物理地址。

页号	物理块号	状态位P	访问字段A	修改位M	外存地址

访问字段A：记录本页在一段时间内是否被访问

- 地址变换机构
  - 情况二： 如果在快表中未找到相应的页表项，检索内存中的页表，查看页表中的状态位，若该页已经调入内存，填写快表，当快表满时，应淘汰一个页表项；若该页尚未调入内存，产生缺页中断，请求OS把该页调入。

页号	物理块号	状态位P	访问字段A	修改位M	外存地址

状态位P：用于指示该页是否已调入内存

缺页中断

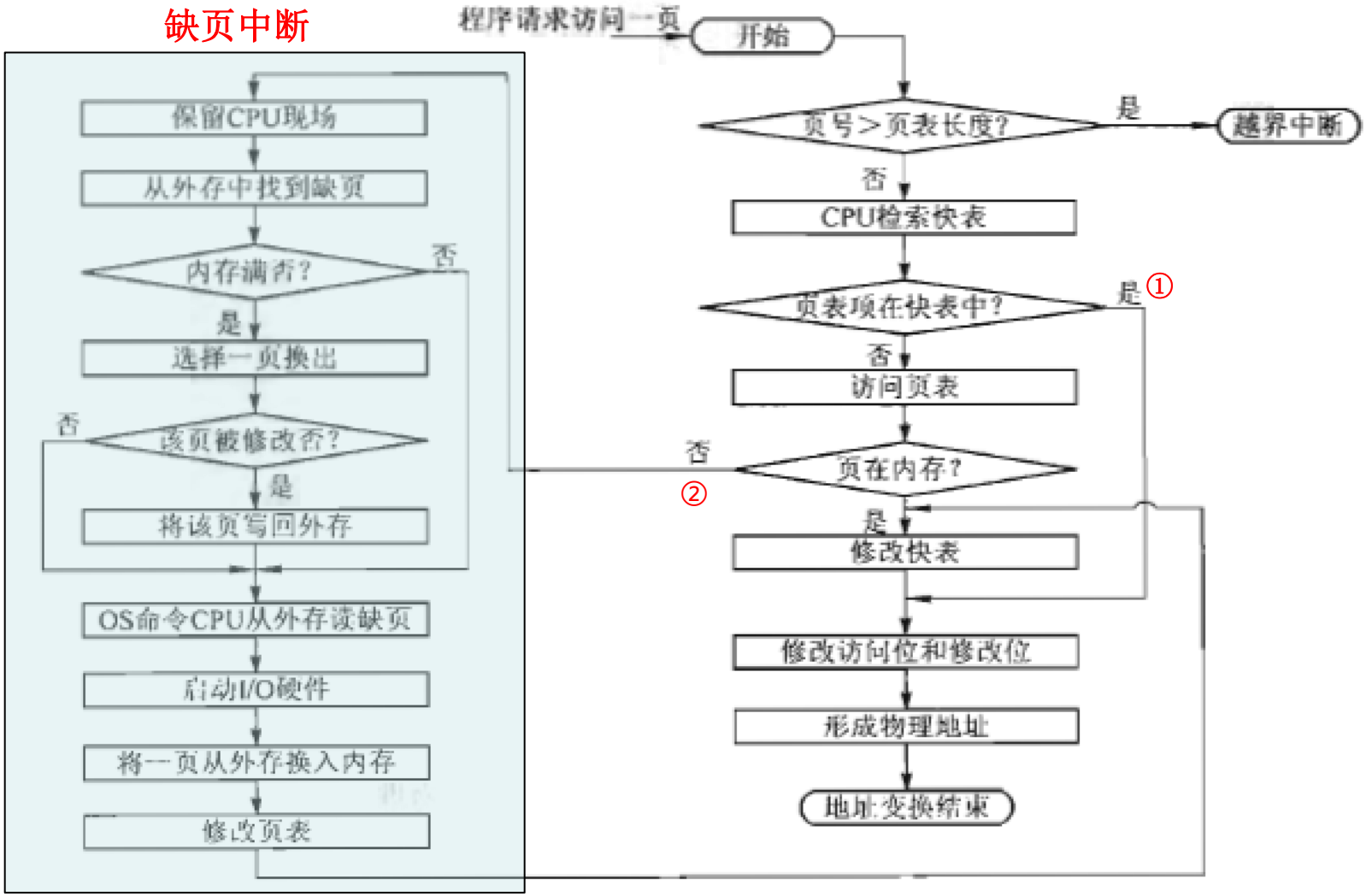


图 5-2　请求分页中的地址变换过程

5.2.2、内存分配策略和分配算法

1、最小物理块数的确定

保证进程正常运行所需的最少物理块数；

与硬件结构有关，取决于指令格式、功能和寻址方式。

2、物理块的分配策略

（1）固定分配局部置换：为进程分配的物理块数在整个运行期间都不再改变。若某个进程发生缺页，则只能将自己的某个内存页换出。

- **（2）可变分配全局置换：**为每个进程分配一定数目的物理块，当进程发生缺页，若系统中有空闲的物理块，则分配一个物理块并装入缺页；页面的置换范围是任一个进程。

- **（3）可变分配局部置换：**为每个进程分配一定数目的物理块后，若某个进程发生缺页，则只能将自己的某个内存页换出。OS根据缺页率进行物理块分配的调整。

3、物理块的分配算法

- 平均分配算法
  - 将空闲物理块，平均分配给各个进程。
- 按比例分配算法
  - 根据进程的大小按比例分配物理块的。
- 考虑优先权的分配算法
  - 按比例分配给各进程
  - 优先权高的一次分得的物理块数多。

5.2.3、请求分页策略

1、调入页面的时机

确定系统将进程运行时所缺的页面调入内存的时机

预调页策略：首次调入内存时

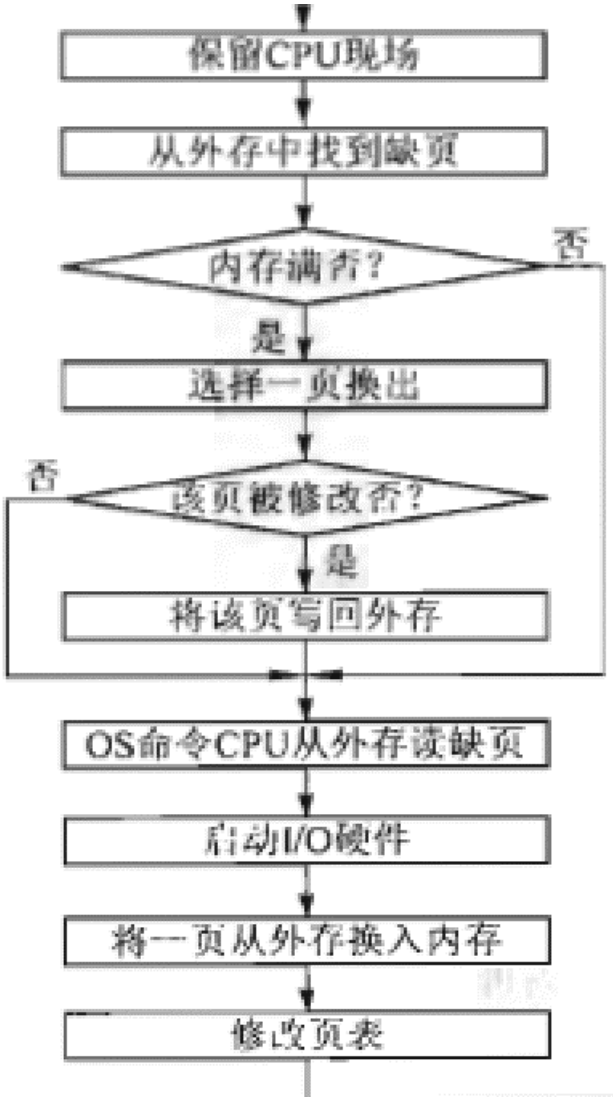
请求调页策略：运行中的发生缺页现象时

2、确定从何处调入页面

- 系统拥有足够的对换区空间
- 系统缺少足够的对换区空间
- UNIX方式

3、页面调入过程

- 向CPU发出缺页中断
- 中断处理程序保存CPU环境转中断处理程序
- 该程序查找页表，得到该页在外存中的块号。
- 若内存未满，启动磁盘I/O读入；若内存已满，先置换，再调入；
- 最后修改页表对应项的内容。



5.3 页面置换算法

当内存中没有可以利用的页架时

根据一定的策略从内存中选择一个页面，把它置换到外存

- **最佳置换算法（OPT）**
- **先进先出（FIFO）**
- **最近最久未使用置换算法（LRU）**
- **Clock置换算法（NRU）**
- **最少使用置换算法（LFU）**
- **页面缓冲置换算法（PBA）**

5.3.1、最佳置换算法和先进先出置换算法

1、最佳置换算法（OPT）

选择以后永远（相比之下，最长时间）不会被使用的页淘汰出去。

特点：

理论上，性能最佳；实际上，无法实现；

通常用该算法来评价其他算法的优劣。

如果所访问的页还没有装入内存，将发生一次缺页中断。

访问过程中发生缺页中断的次数就是缺页次数。缺页次数除以总的访问次数，就是缺页率。

例1：在一个请求分页系统中，假定系统分给一个作业的物理块数为3，并且此作业的页面走向为2，3，2，1，5，2，4，5，3，2，5，2。

用FIFO、LRU、OPT计算缺页次数和缺页率。

使用OPT算法：

将来再也不用或最长时间不用的页面
——黄色标志

页面走向	2	3	2	1	5	2	4	5	3	2	5	3
1	2	2	2	2	2	2	4	4	4	2	2	2
2		3	3	3	3	3	3	3	3	3	3	3
3				1	5	5	5	5	5	5	5	5
缺页中断	+	+		+	+		+			+		

缺页次数：6，缺页率：6/12，页面置换3次

2、先进先出置换算法（FIFO）

总是先淘汰那些最先进入系统，即驻留主存时间最长的页。

特点：

- *实现简单*
- *与进程实际的运行不相适应*

使用FIFO算法:

驻留内存最久的页面—黄色标志

页面走向	2	3	2	1	5	2	4	5	3	2	5	2
1	2	2	2	2	5	5	5	5	3	3	3	3
2		3	3	3	3	2	2	2	2	2	5	5
3				1	1	1	4	4	4	4	4	2
缺页中断	+	+		+	+	+	+		+		+	+

缺页次数：9，缺页率：9/12；页面置换6次

5.3.2 最近最久未使用置换算法（LRU）

选择在最近一段时间最久未被使用（访问）的页淘汰出去。

P165

特点：

性能较好

实现复杂，需要硬件支持（每页配置一个寄存器、栈），增加系统负担。

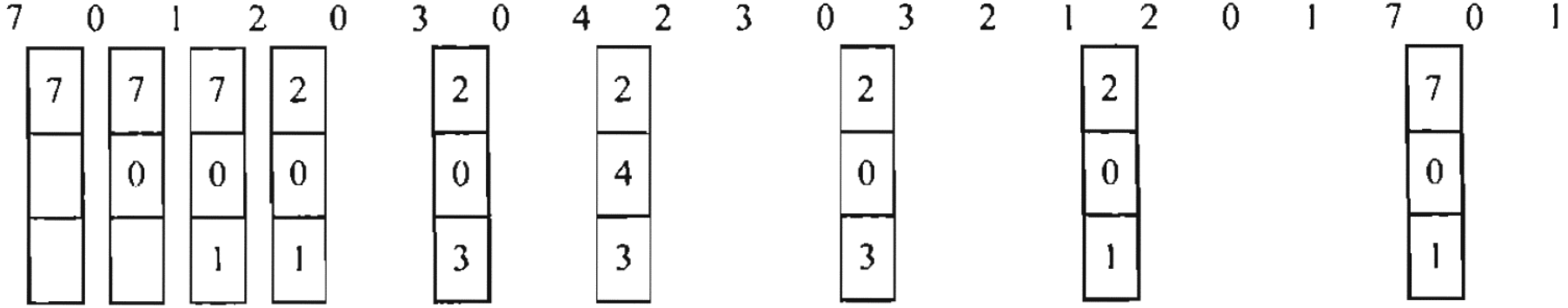
使用LRU算法：

长时间没有访问的页面——黄色标志
刚刚访问过的页面——绿色标志

页面走向	2	3	2	1	5	2	4	5	3	2	5	2
1	2	2	2	2	2	2	2	2	3	3	3	3
2		3	3	3	5	5	5	5	5	5	5	5
3				1	1	1	4	4	4	2	2	2
缺页中断	+	+		+	+		+		+	+		

缺页次数：7，缺页率：7/12；页面置换4次

LRU最接近OPT，表明LRU优于FIFO。



例2：在一个请求分页系统中，假如一个作业的页面走向为1，2，3，4，1，2，5，1，2，3，4，5，

当分给该作业的物理块数M分别为3和4时，请用FIFO计算缺页次数和缺页率，并比较所得的结果。

使用FIFO算法——物理块数为3:

页面走向	1	2	3	4	1	2	5	1	2	3	4	5
1	1	1	1	4	4	4	5	5	5	5	5	5
2		2	2	2	1	1	1	1	1	3	3	3
3			3	3	3	2	2	2	2	2	4	4
缺页中断	+	+	+	+	+	+	+			+	+	

缺页次数：9，缺页率：9/12

使用FIFO算法——物理块数为4:

页面走向	1	2	3	4	1	2	5	1	2	3	4	5
1	1	1	1	1	1	1	5	5	5	5	4	4
2		2	2	2	2	2	2	1	1	1	1	5
3			3	3	3	3	3	3	2	2	2	2
4				4	4	4	4	4	4	3	3	3
缺页中断	+	+	+	+			+	+	+	+	+	+

缺页次数：10，缺页率：10/12

异常现象称为Belady现象。

2、LRU置换算法的硬件支持 P165

1）寄存器

为每个在内存中的页面配置一个移位寄存器，表示为： $R=R_{n-1}R_{n-2}\cdots R_1R_0$

当进程访问此物理块时，将R_{n-1}位置1。

定时信号将每隔一定时间将寄存器右移一位。

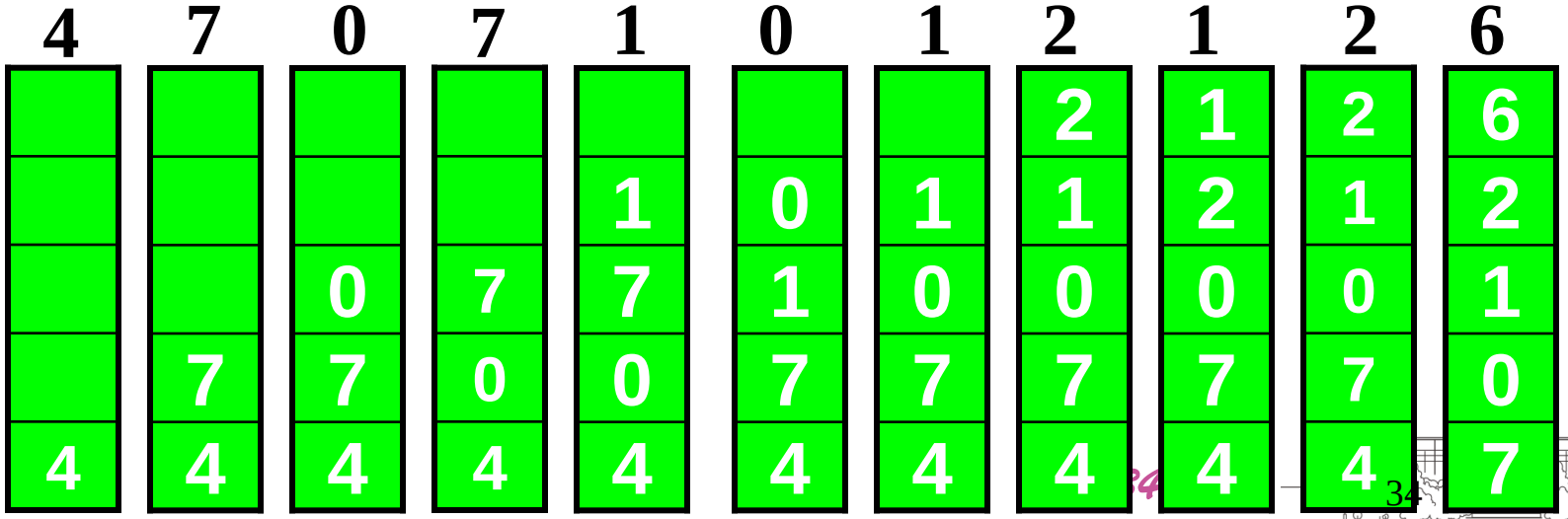
具有最小数值的寄存器所对应的页面就是最近最久未使用的页面。

实页 \ R	R_7	R_6	R_5	R_4	R_3	R_2	R_1	R_0
1	0	1	0	1	0	0	1	0
2	1	0	1	0	1	1	0	0
3	0	0	0	0	0	1	0	0
4	0	1	1	0	1	0	1	1
5	1	1	0	1	0	1	1	0
6	0	0	1	0	1	0	1	1
7	0	0	0	0	0	1	1	1
8	0	1	1	0	1	1	0	1

某进程具有8个页面时的LRU访问情况

2）栈　利用栈来保存当前使用的各个页面的页面号。当进程访问此页面时，将该页面的页面号从栈中移出，压入栈顶。因此栈顶是最新被访问页面的编号，栈底是最近最久未使用页面的页面号。

例：现有一进程所访问的页面号序列如下：
4，7，0，7，1，0，1，2，1，2，6　栈的变换情况为：



5.3.3、Clock置换算法

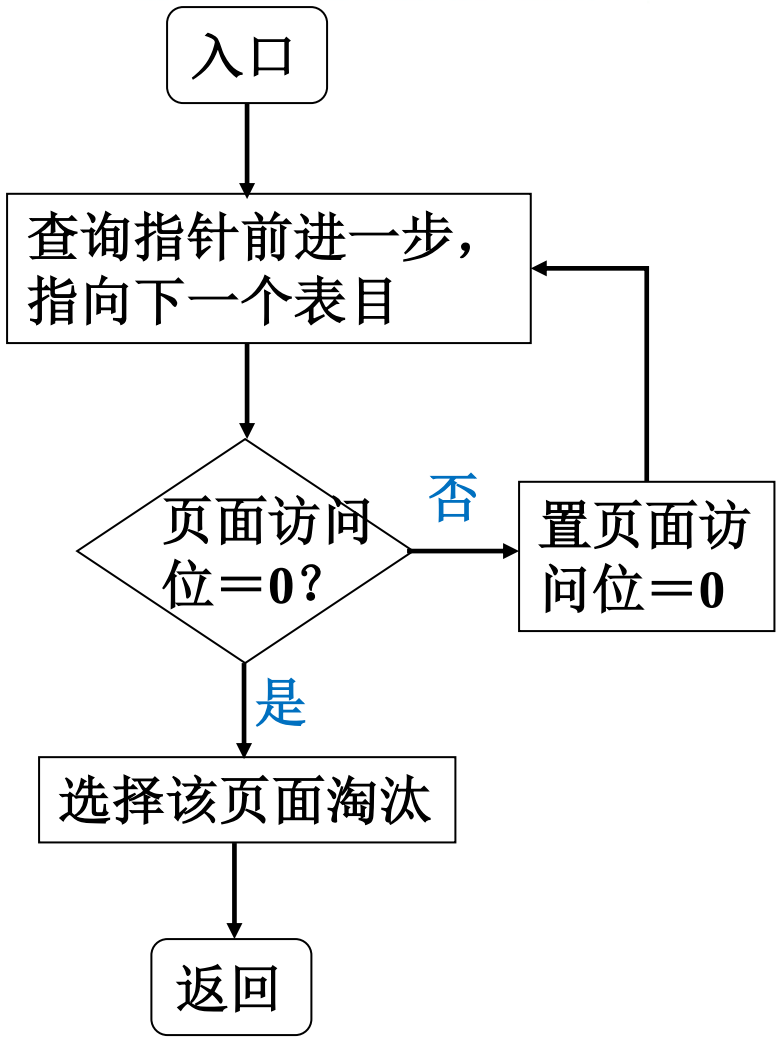
又称为“最近未使用”置换算法（NRU）

1、简单的Clock置换算法

每页设置一位访问位。当某页被访问了，则访问位置“1”。内存中的所有页链接成一个循环队列；

算法描述：

循环检查各页面的使用情况。若访问位为“0”，选择该页淘汰；若访问位为“1”，复位访问位为“0”，查询指针前进一步。



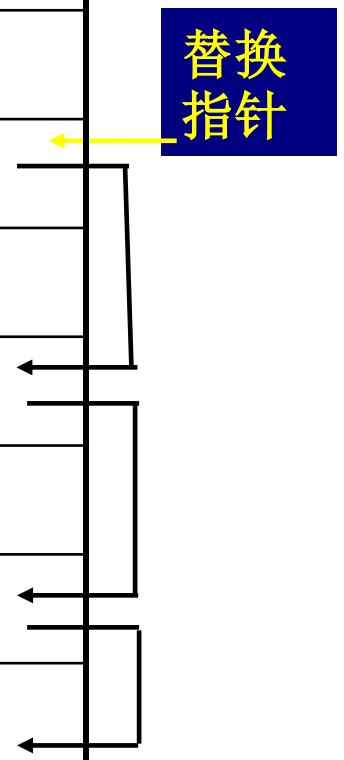
块号	页号	访问位	指针
0			
1			
2	4	0	
3			
4	2	0	
5			
6	5	0	
7	1	0	

替换指针

2、改进型Clock置换算法：

- 访问位A、修改位M，共同表示一个页面的状态
- 四种状态：
  - 00： (A=0;M=0)最近未被访问也未被修改
  - 01： (A=0;M=1)最近未被访问但已被修改
  - 10： (A=1;M=0)最近已被访问但未被修改
  - 11： (A=1;M=1)最近已被访问且被修改
- 三轮扫描：
  - 第一轮：查找00页面，未找到，下一步；
  - 第二轮：查找01页面，A位复位为“0”，未找到，下一步；
  - 第三轮：重复第一轮，必要时再重复第二轮。

块号	页号	访问位	修改位	指针
0				
1				
2	4	0	0	
3				
4	2	0	1	
5				
6	5	1	0	
7	1	1	1	



5.3.4、其它置换算法

1、最少使用置换算法（LFU）

选择在最近时期使用最少的页面淘汰。（频率）

为在内存中的每个页面设置一个移位寄存器，用来记录该页面被访问的频率。每次访问某页时，便将该移位寄存器的最高位置1，此后每隔一定时间自动右移一位。

最近一段时间最少使用的页面是$\sum R_i$最小的页。

R / 实页	R_7	R_6	R_5	R_4	R_3	R_2	R_1	R_0
1	0	1	0	1	0	0	1	0
2	1	0	1	0	1	1	0	0
3	0	0	0	0	0	1	0	0
4	0	1	1	0	1	0	1	1
5	1	1	0	1	0	1	1	0
6	0	0	1	0	1	0	1	1
7	0	0	0	0	0	1	1	1
8	0	1	1	0	1	1	0	1

2、页面缓冲算法（PBA）

算法：FIFO，淘汰页放入空闲页面链表或已修改页面链表末尾。

空闲页面链表：不需写回磁盘，装入新页面时，在该表中选择一个页架。

已修改页面链表：该链表中的页面数达到一定数目时，才集中写回到磁盘上。

优点：消除频繁的磁盘I/O

5.4 “抖动”与工作集

请求分页系统性能优越，但若在系统中运行的进程太多，使缺页现象频繁发生，就会对系统的性能产生很大影响，因此，需对分页系统的性能做简单分析。

多道程序度与“抖动”

工作集

抖动的预防方法

5.5 请求分段存储管理方式

以基本分段内存管理为基础，程序运行前，只需先调入部分分段，便启动执行。其它分段在需要时，通过缺段中断处理程序临时调入。

请求分段中的硬件支持

请求分段中的分段共享

请求分段中的分段保护

5.5.1、请求分段中的硬件支持

1、段表机制

段名(号)	段长	段的基址	存取方式	访问字段A	修改位M	存在位P	增补位	外存始址

(1) 存取方式：存取属性为只执行、只读或允许读/写

(2) 访问字段A：记录该段被访问的频繁程度

(3) 修改位M：该段在进入内存后是否被修改过

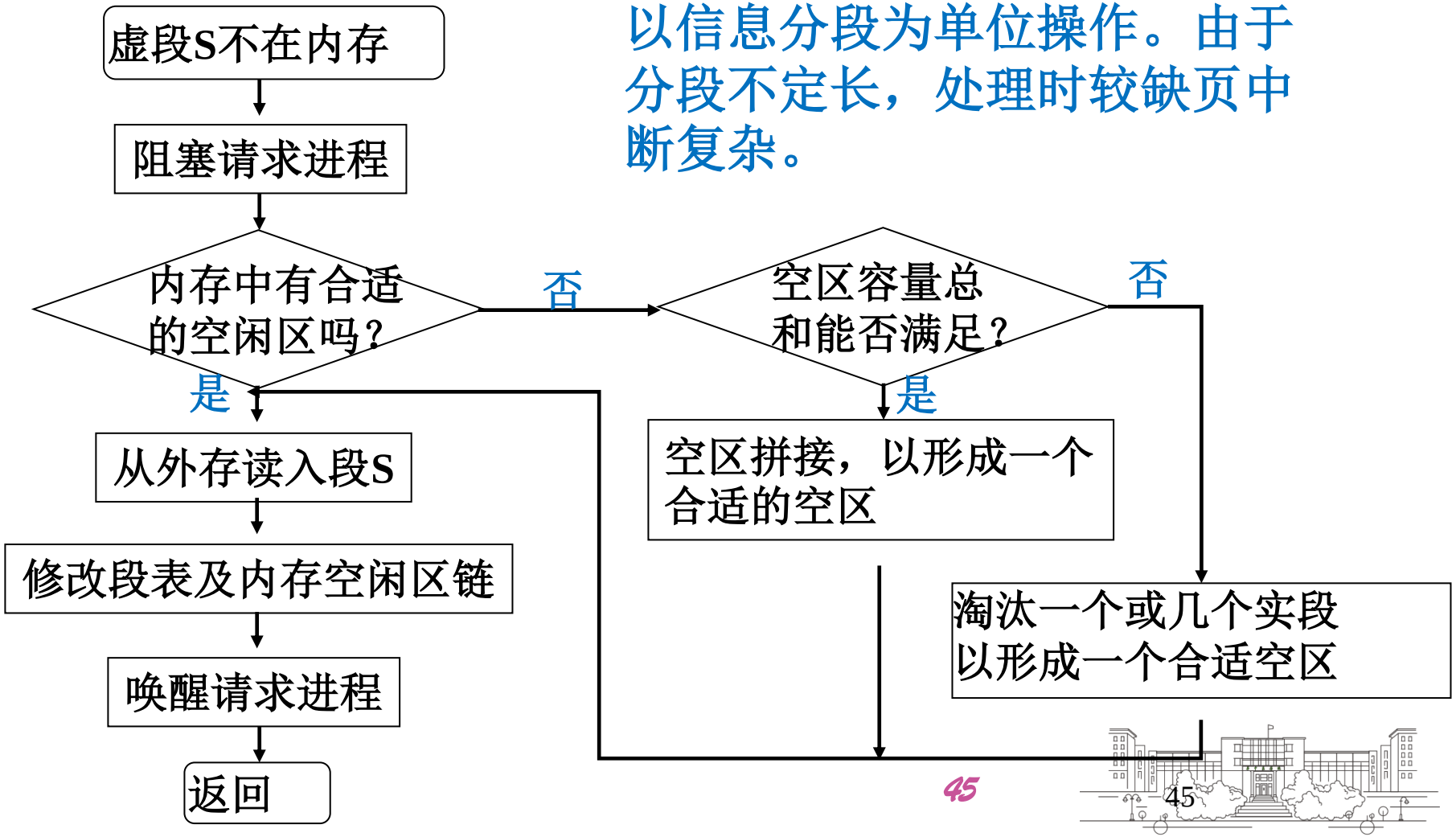
(4) 存在位P：指示本段是否已调入内存

(5)增补位：表示本段在运行过程中，是否做过动态增长

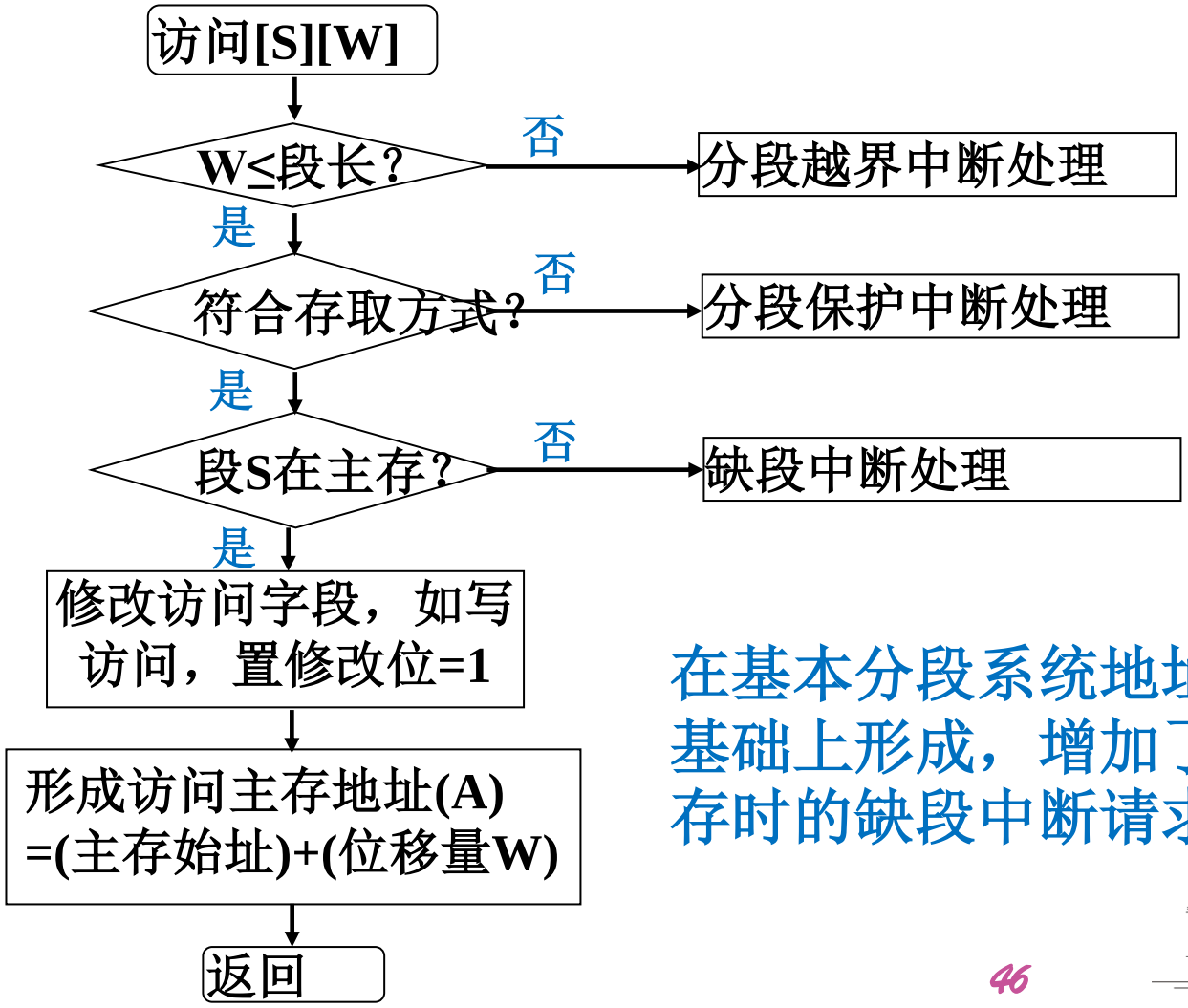
(6)外存始址：本段在外存中的起始地址，即起始盘块号

2、缺段中断机构

以信息分段为单位操作。由于分段不定长，处理时较缺页中断复杂。



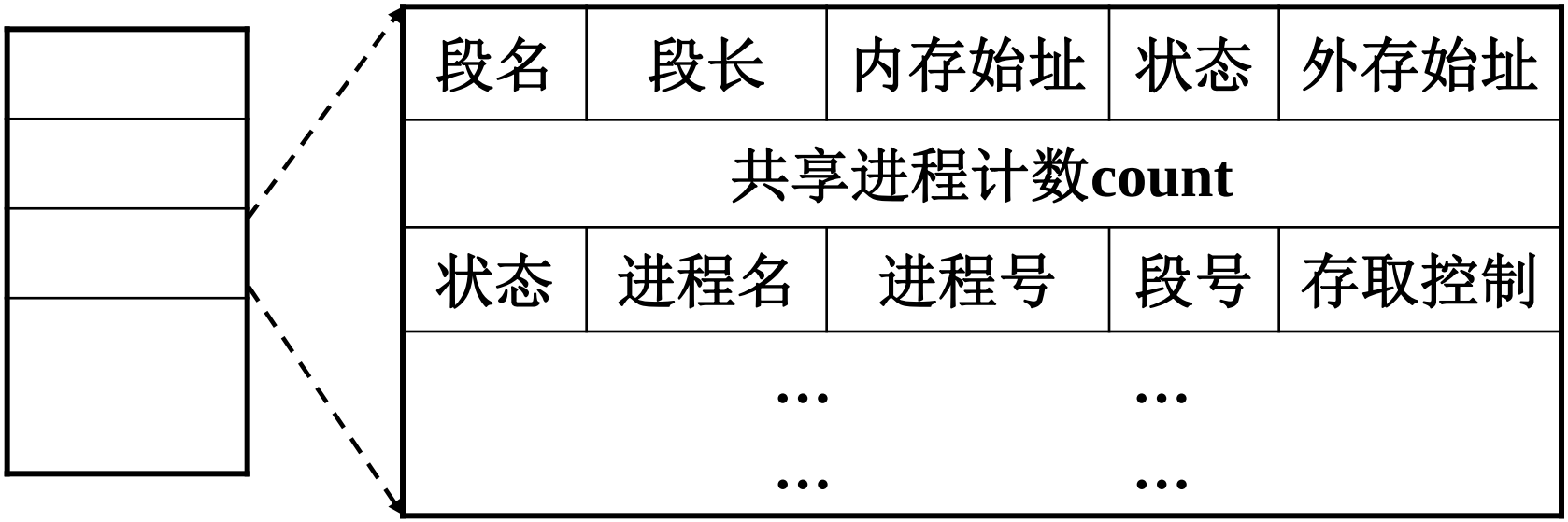
3、地址变换机构



在基本分段系统地址变换机构的基础上形成，增加了分段不在内存时的缺段中断请求。

5.5.2、分段的共享与保护

1、共享段表



共享进程计数count： 记录要共享该段的进程数目

存取控制字段： 对一共享段，给不同进程不同权限

段号： 不同进程可以各用不同段号去共享某共享段

2、共享段的分配

对第一个请求使用该共享段的进程，由系统为该共享段分配一物理区，在把共享段调入该区。置count=1

其他进程要调用该共享段时，填写共享段的段表，执行count=count+1

3、共享段的与回收

当进程不再需要共享段时，先释放，撤消共享段的表项，执行count=count-1

仅当count=0时，由系统回收共享段的物理内存。

4、分段保护

1）越界检查

段表寄存器：段表始址、段表长度

比较：段号—段表长度、段长—段内地址

2）存取控制检查

存取控制字段

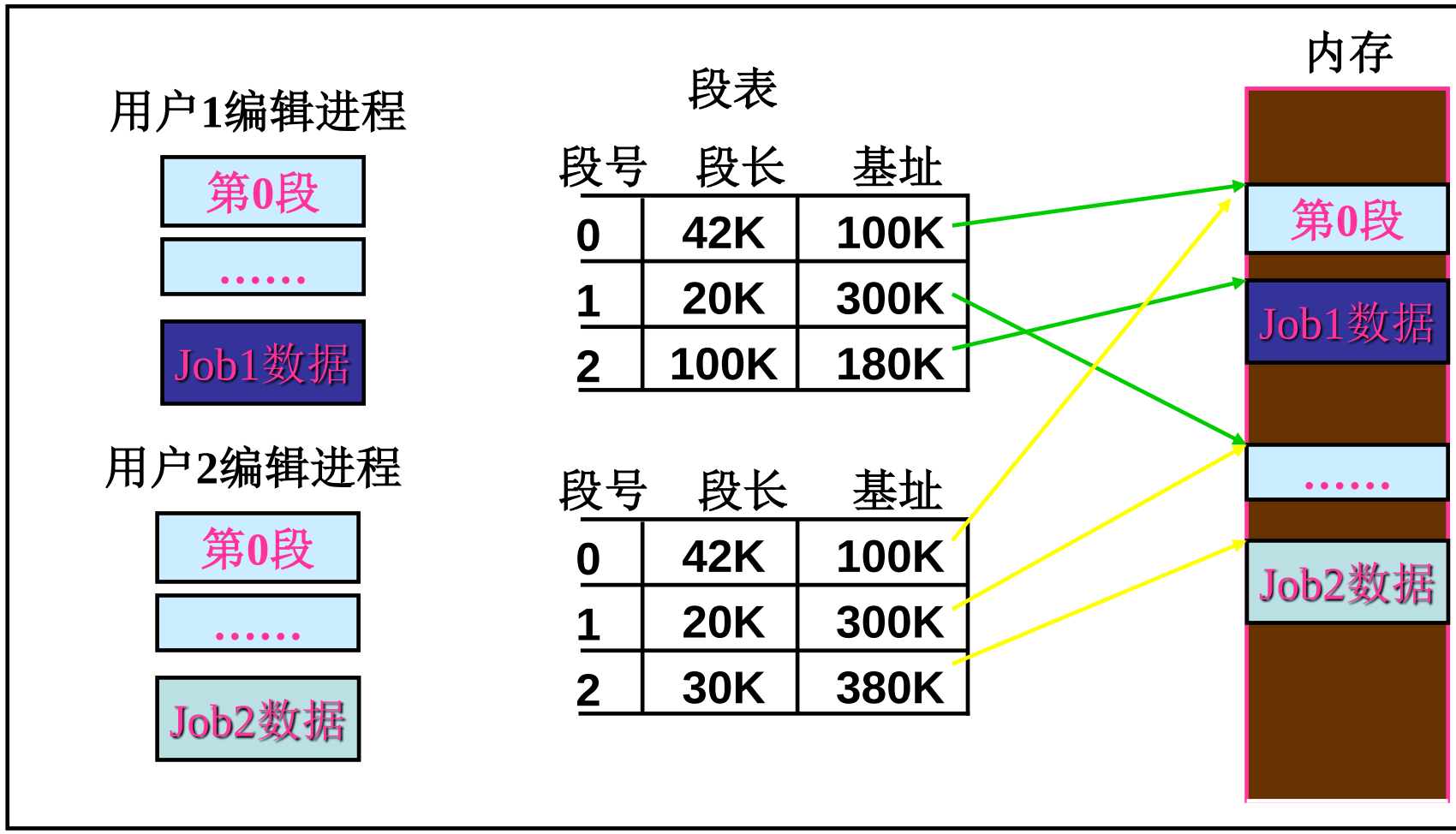
只读、只执行、读/写

3）环保护机构

低编号的环具有高优先权。

程序可以访问驻留在相同环或较低特权环中的数据；

程序可以调用驻留在相同环或较高特权环中的服务。


用户1编辑进程
第0段
……
Job1数据
用户2编辑进程
第0段
……
Job2数据
段表
段号 段长 基址
0 42K 100K
1 20K 300K
2 100K 180K
段号 段长 基址
0 42K 100K
1 20K 300K
2 30K 380K
内存
第0段
Job1数据
……
Job2数据

主观题 10分

设置

HUAZHONG AGRICULTURAL UNIVERSITY

在本课堂上，你想学习那些内容？
你有那些建议？

作答

正常使用主观题需2.0以上版本雨课堂

投票(匿名) 最多可选5项

设置

HUAZHONG AGRICULTURAL UNIVERSITY

在线听课评价:

A 教师准备充分,讲述条理清晰

B 能调动学习的积极性

C 教学方式单一、缺乏互动

D 学生的收获少，获得感不强

E 内容枯燥、听不懂

提交