

## 第七章 虚拟存储管理技术

— 通过地址空间部分装入，获得主存“**扩充**”效果



## § 1 虚拟存储器概念

### 一、局部性原理(Denning,1968)

**几乎所有的程序在执行时都呈现出局部性规律。**即在一较短时间间隔内，程序的执行总是局限于地址空间的某个局部区域，呈现出地址访问聚集成群的倾向。

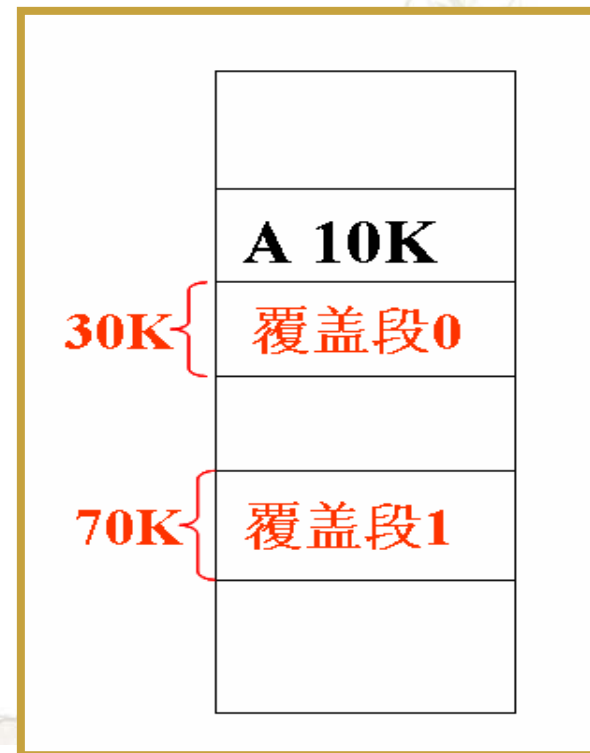
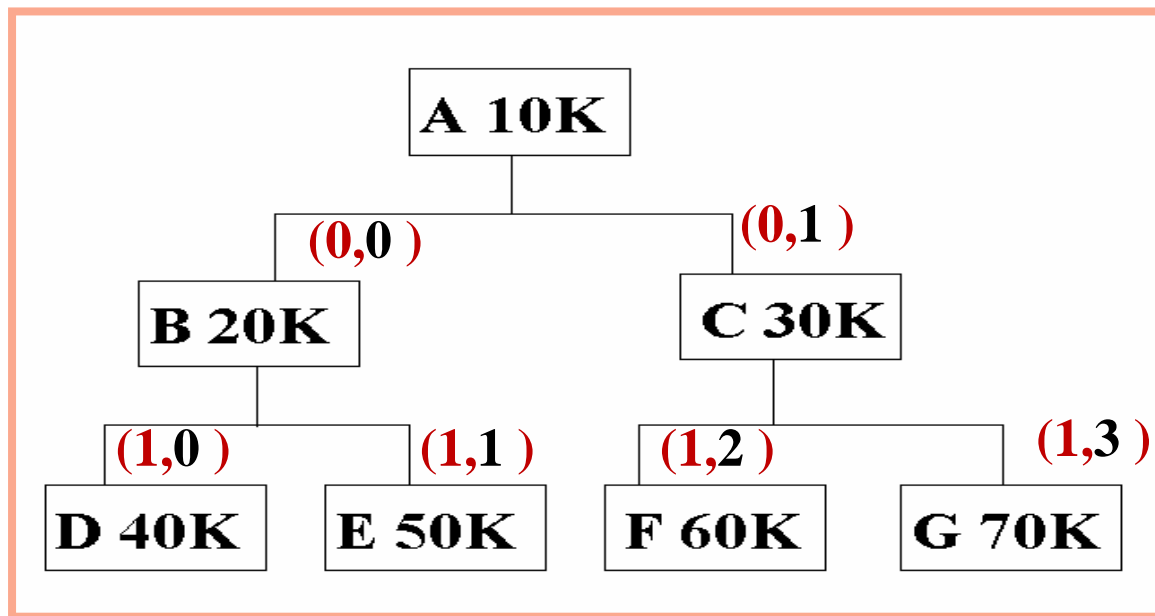
● **时间局部性：**一旦程序的某个位置 —— 数据或指令被访问，它常常再次被访问。

● **空间局部性：**一旦程序的某个位置 —— 数据或指令被访问，它附近的位置很快被访问。



# § 1 虚拟存储器概念

## 二、(自动)覆盖技术



$$10+20+30+40+50+60+70=280K$$

$$\text{覆盖: } 10+30+70=110K$$





## § 1 虚拟存储器概念

### 三、虚拟存储器定义

- **早期定义**：通过自动**覆盖技术**将多级存储器变成用户意义上的一级存储器，这一远**大于主存**的存储器称为虚拟存储器。
- **一般性定义**：系统通过软硬件技术的结合，为每个作业提供的地址空间是一个虚拟存储器。
- **虚存容量 —— 由CPU地址结构决定**
  - eg:总线寻址能力为20位，CPU有效地址为18位  
【分析】实存=1M，虚存=256K，虚存 < 实存
  - eg:总线寻址能力为20位，CPU有效地址为20位  
【分析】实存=1M，虚存=1M，虚存 = 实存
  - eg:总线寻址能力为20位，CPU有效地址为24位  
【分析】实存=1M，虚存=16M，虚存 > 实存
- **虚拟存储管理系统**：具有主存“**扩充**”功能的存储管理系统



## § 1 虚拟存储器概念

例1 (**南理工**)：虚拟存储器的容量是由计算机地址结构决定的，若CPU的地址为32位，则对于一个进程来说，其最大的虚拟存储空间为 **B** B。

- A. 2G      B. 4G      C. 0.5G      D. 1G

例2：设主存容量为8MB,辅存容量为50MB，计算机地址寄存器是24位，则虚存的最大容量为 **D**。

- A. 8MB      B. 50MB+8MB      C. 50MB+ $2^{24}$ B      D.  $2^{24}$ B

例3：判断题。

(1) (**东大**)虚拟存储器是一个假想的地址空间，因而这个地址的大小是没有限制的。( **F** )

(2) (**北大**)虚拟存储技术是一种拿时间换空间的技术。( **T** )

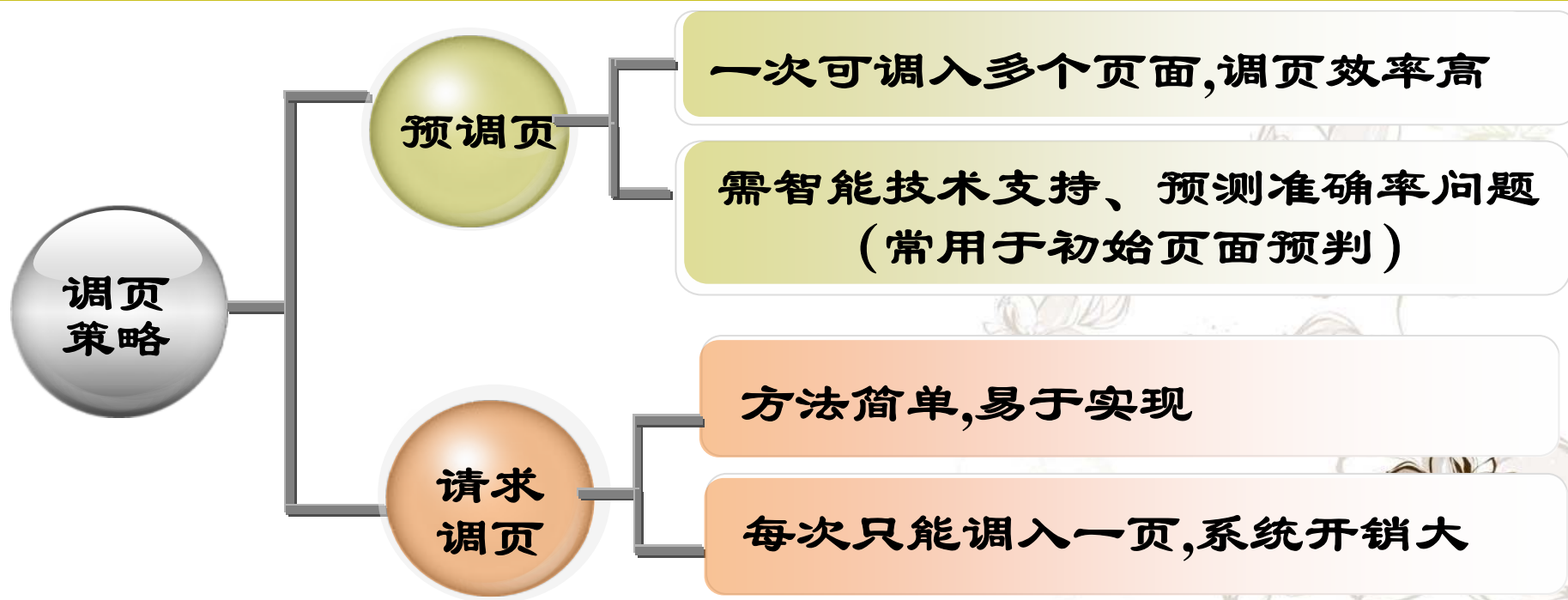


## § 2 请求分页存储管理

### 页式虚拟存储管理系统

#### 一、基本原理

1. **地址空间**等分为**虚页**，**存储空间**等分为**实页**
2. 运行过程中，只**装入**作业的**部分页面**
3. 如何调入所需页面：有两种不同的调页策略







## § 2 请求分页存储管理

### 一、基本原理

1. **地址空间**等分为**虚页**，**存储空间**等分为**实页**
2. 运行过程中，只**装入**作业的**部分页面**
3. 如何调入所需页面：有两种不同的调页策略
4. 如何判某页是否在主存？——扩充页表

状态位、有效位、中断位：指示对应虚页是否在主存

辅存地址：指示页面副本在辅存的位置

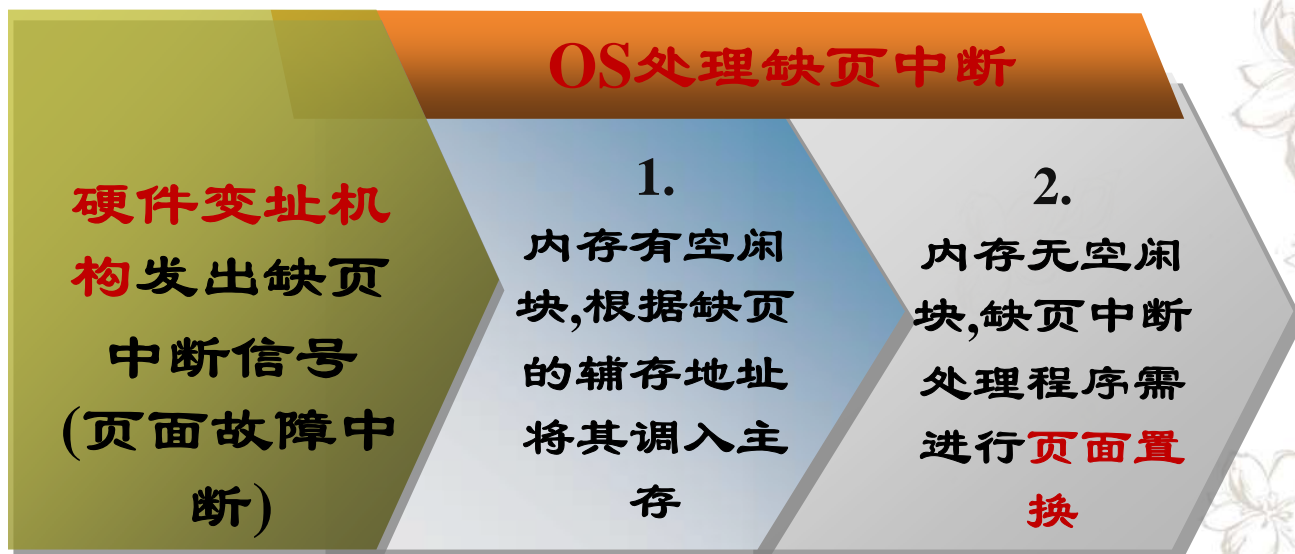
虚页号	主存块号	状态位	外存地址
0	$B_0$	0或1	(磁道号,盘面号,扇区号)
...	...	...	...



## § 2 请求分页存储管理

### 一、基本原理

- 1.地址空间等分为虚页，存储空间等分为实页
- 2.运行过程中，只装入作业的部分页面
- 3.如何调入所需页面：有两种不同的调页策略
- 4.如何判某页是否在主存？——扩充页表
- 5.如何处理页面不在主存的情况？







## § 2 请求分页存储管理

### 一、基本原理

- 1.地址空间等分为虚页，存储空间等分为实页
- 2.运行过程中，只装入作业的部分页面
- 3.如何调入所需页面：有两种不同的调页策略
- 4.如何判某页是否在主存？——扩充页表
- 5.如何处理页面不在主存的情况？

#### 固定分配 局部置换

进程驻留集  
大小不变，从  
进程内部**置换**  
页面

#### 可变分配 全局置换

可向OS申请  
新的主存块，  
否则从任一  
进程驻留集  
中**置换**

#### 可变分配 局部置换

在进程内部**置  
换**，若页面故  
障频繁，则追  
加主存



## § 2 请求分页存储管理

### 一、基本原理

- 1.地址空间等分为虚页，存储空间等分为实页
- 2.运行过程中，只装入作业的部分页面
- 3.如何调入所需页面：有两种不同的调页策略
- 4.如何判某页是否在主存？——扩充页表
- 5.如何处理页面不在主存的情况？
- 6.扩充的**页表**：

虚页号	主存块号	状态位P	访问字段A	修改位M	外存地址
-----	------	------	-------	------	------

**页面活跃度**：页面近期被访问的次数，或最近未被访问的时间

**页面不洁净位**：页面装入主存后是否被修改过。



## § 2 请求分页存储管理

例1(西安电子科技大学):在请求分页存储管理中,当所访问的页面不在内存时,便产生缺页中断,缺页中断是属于 B。

- A. I/O中断      B. 程序中断      C. 访管中断      D. 外中断

例2: 作业在执行中发生缺页中断, 经操作系统处理后, 应让其执行 C 指令。

- A. 被中断的前一条      B. 被中断的后一条  
C. 被中断的那一条      D. 启动时的第一条

例3(南理工): 在请求分页系统中, 凡未装入过的页都应从 B 调入主存。

- A. 系统区      B. 文件区      C. 交换区      D. 页面缓冲区



## § 2 请求分页存储管理

例4(苏大)：为了支持请求式分页内存管理，通常页表项内存有一**标志位**，用来记录相应页是否被写过，请解释该标志位的**操作者**及其**作用**。

该标志位是页表项中的“修改位”，其作用是指示该页调入内存后是否被修改过，它涉及若需淘汰该页时是否要将其写回到辅存的问题。

“修改位”是由**硬件地址变换机构**在进行重定位的过程中，若此次操作是“写”操作而置位的。

例5(11年)：在缺页处理过程中，操作系统执行的操作可能是   D  。

I修改页表    II磁盘I/O    III分配页框

A. 仅I、II      B. 仅II      C. 仅III      D. I、II和III





## § 2 请求分页存储管理

**例6(12年):** 某请求分页系统的局部页面置换策略如下：  
系统从0时刻开始扫描，每隔5个时间单位扫描一轮驻留集  
(扫描时间忽略不计)，本轮没有被访问过的页框将被系统  
回收，并放入到空闲页框链尾，其中内容在下一次被分配之  
前不被清空。当发生缺页时，如果该页曾被使用过且还在空  
闲页框链表中，则重新放回进程的驻留集中；否则，从空闲  
页框链表头部取出一个页框。

假设不考虑其它进程的影响和系统开销，初始时进程驻留集  
为空。目前系统空闲页框链表中页框号依次为32、15、21、  
41。进程P依次访问的<虚拟页号，访问时刻>是：<1,1>、  
<3,2>、<0,4>、<0,6>、<1,11>、<2,14>。

回答：(1) 访问<0,4>时，对应的页框号是什么？

(2) 访问<1,11>时，对应的页框号是什么？说明理由。

(3) 访问<2,14>时，对应的页框号是什么？说明理由。







## § 3 页面置换算法及性能

### 一、抖动(颠簸)

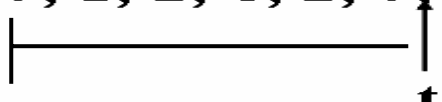
Thrashing是指不合适的页面置换算法所导致的页面频繁置换现象:即一页面刚换出不久又需调入,为此选择一页调出,而这一刚换出的页,很快又要访问又需将其调入.....如此频繁的更换页面,使大部分机器时间都花费在页面置换上。

### 二、工作集(Working set)

进程运行中距时刻 $t$ 最近的 $\Delta$ 次访问所涉及的页面集,称作进程的工作集,记为 $W(t, \Delta)$ 。其中数字 $\Delta$ 称为工作集窗口。

例1: 已知进程访问如下页面, 窗口尺寸 $\Delta=6$ , 试求 $t$ 时刻的工作集。

0, 9, 0, 1, 8, 1, 8, 7, 8, 7, 1, 2, 8, 2, 7, 8, 2, 3, 8, 3。



$t$

$W(t, 6) = \{1, 2, 7, 8\}$





## § 3 页面置换算法及性能

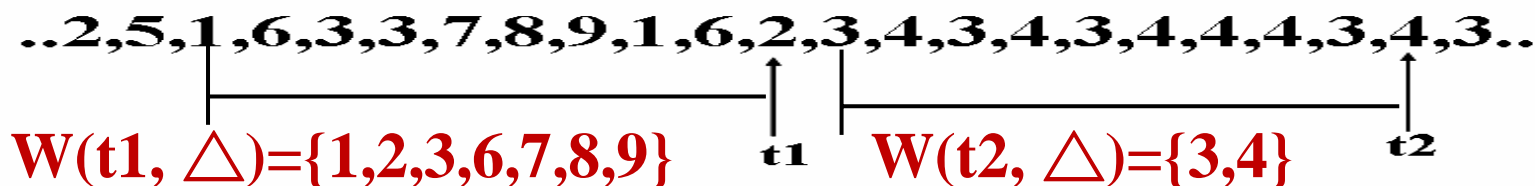
### 一、抖动(颠簸)

Thrashing是指不合适的页面置换算法所导致的页面频繁置换现象:即一页面刚换出不久又需调入,为此选择一页调出,而这一刚换出的页,很快又要访问又需将其调入.....如此频繁的更换页面,使大部分机器时间都花费在页面置换上。

### 二、工作集(Working set)

进程运行中距时刻 $t$ 最近的 $\Delta$ 次访问所涉及的页面集,称作进程的工作集,记为 $W(t, \Delta)$ 。其中数字 $\Delta$ 称为工作集窗口。

例2:  $\Delta=9$ , 试求 $w(t_1, \Delta)$ 和 $w(t_2, \Delta)$





## § 3 页面置换算法及性能

### 三、缺页中断率 $f$ —固定分配局部置换

- 设作业共计 $n$ 页,系统分配给该作业的主存块为 $m$ 块,且 $1 \leq m < n$ 
  - 若作业运行中成功访问次数为 $S$ ;
  - 若作业运行中不成功访问次数为 $F$ ;
  - 则总访问次数  $A=S+F$ ;
  - 定义:  $f=F/A$ 为作业的缺页中断率

### 四、影响 $f$ 的因素

- 分配给作业的主存块数  $m$
- 页面的尺寸大小
- 程序编制方法
- 页面置换算法性能优劣





## § 3 页面置换算法及性能

例：一程序将 $256 \times 256$ 矩阵A置初值0。该矩阵定义为：

VAR A:ARRAY[1..256,1..256] of INTEGER;

假定分配给该矩阵的内存块为1块，**页面大小为每页256个整数**，**矩阵按行存放，开始内存为空**。若程序和有关变量已存放主存。问：程序运行时共发生多少次缺页中断？

A.  $256-1$     B. 256    C.  $256^2-1$     D.  $256^2$

### 【程序1】

FOR J:=1 to 256

FOR I:=1 to 256

A[I,J]:=0;

**D**

### 【程序2】

FOR I:=1 to 256

FOR J:=1 to 256

A[I,J]:=0;

**B**



# § 3 页面置换算法及性能

## 五、常用页面置换算法

### 1.最佳置换算法OPT (Optimal)

选择以后**不再访问**的页或距现在**最长时间后**才需再访问的页进行置换

例:一作业在执行中的页面引用串为7,0,1,2,0,3,0,4,2,3,0,3,2,1,2,0,1,7,0,1  
分配给该作业的主存块m=3,试计算基于OPT的缺页中断率f。

解: 预调页后, 内存页面号: 7,0,1

地址访问	缺页中断次数	调出页	主存页面号
2	1	7	0, 1, 2
3	2	1	0, 2, 3
4	3	0	2, 3, 4
0	4	4	2, 3, 0
1	5	3	2, 0, 1
7	6	2	0, 1, 7

$f=6/20=30\%$



## § 3 页面置换算法及性能

### 2.先进先出算法FIFO (First-in, First-out) ——Belady现象

●思想：淘汰最早进入主存，在主存驻留时间最长的页面。

- ①建立一个存放内存页面号的链表,最老页在链首,最新页在链尾
- ②淘汰链首页面，将新调入的页面号结点插至链尾

例1:已知一作业获得3块主存,其页面访问次序：4,3,0,4,1,1,2,3,2。  
试计算基于FIFO的缺页中断次数。

解：预调页后初始页面：4，3，0

第几次缺页中断	调进/出页	内存页面号
1	1/4	3,0,1
2	2/3	0,1,2
3	3/0	1,2,3

$$F=3 \quad f=3/9 \approx 33\%$$



## § 3 页面置换算法及性能

例2 (南理工、中山大学): 什么是Belady现象? 举例说明。

**Belady现象是指:** 在请求分页系统中,若采用FIFO算法进行页面置换,有可能出现随着分配给进程的主存块增加,缺页中断次数也随之增加的异常现象。

**举例:** 页面走向为4,3,2,1,4,3,5,4,3,2,1,5。 **初始内存为空。** 分别以主存驻留集为3和4块计算。

第几次缺页中断	调进/出页	内存页面号	缺页中断	调进/出页	内存页面号
1	4	4	1	4	4
2	3	4,3	2	3	4,3
3	2	4,3,2	3	2	4,3,2
4	1/4	3,2,1	4	1	4,3,2,1
5	4/3	2,1,4	5	5/4	3,2,1,5
6	3/2	1,4,3	6	4/3	2,1,5,4
7	5/1	4,3,5	7	3/2	1,5,4,3
8	2/4	3,5,2	8	2/1	5,4,3,2
9	1/3	5,2,1	9	1/5	4,3,2,1
			10	5/4	3,2,1,5





## § 3 页面置换算法及性能

### 3. LRU算法(Least Recently Used)

● **思想**：淘汰最近一段时间最久未使用的页面

● **实现**：①建立堆栈存放进程当前在主存的页面号

②每访问一页，调整栈一次：将所访问页号调至栈顶

③淘汰栈底指示的页面，新进页压入栈顶

例1：某程序在内存中分配三个页面，初始为空，页面走向为4, 3, 2, 1, 4, 3, 5, 4, 3, 2, 1, 5。在LRU页面置换策略下，该程序执行中共产生几次缺页中断？

LRU	4	3	2	1	4	3	5	4	3	2	1	5
页1	4	3	2	1	4	3	5	4	3	2	1	5
页2		4	3	2	1	4	3	5	4	3	2	1
页3			4	3	2	1	4	3	5	4	3	2
	X	X	X	X	X	X	X	☐	☐	X	X	X

共缺页中断10次





# § 3 页面置换算法及性能

例2(09年)：请求分页管理系统中，假设某进程页表：

页面大小4KB,一次内存访问时间是100ns，一次快表(TLB)的访问时间是10ns,处理一次缺页的平均时间为 $10^8$ ns(已含更新TLB和页表的时间),进程的驻留

页号	页框号	有效位(存在位)
0	101H	1
1	/	0
2	254H	1

集大小固定为2,采用LRU和局部淘汰策略。假设①TLB初始为空；②地址转换时先访问TLB,若TLB未命中,再访问页表(忽略访问页表之后的TLB更新时间)；③有效位为0表示页面不在内存,产生缺页中断,处理后返回到产生缺页中断的指令处重新执行。设有虚地址访问序列2362H、1565H、25A5H，试问：(1)依次访问上述三个虚地址各需多少时间？给出计算过程。(2)基于上述访问序列，虚地址1565H的物理地址是多少？说明理由。



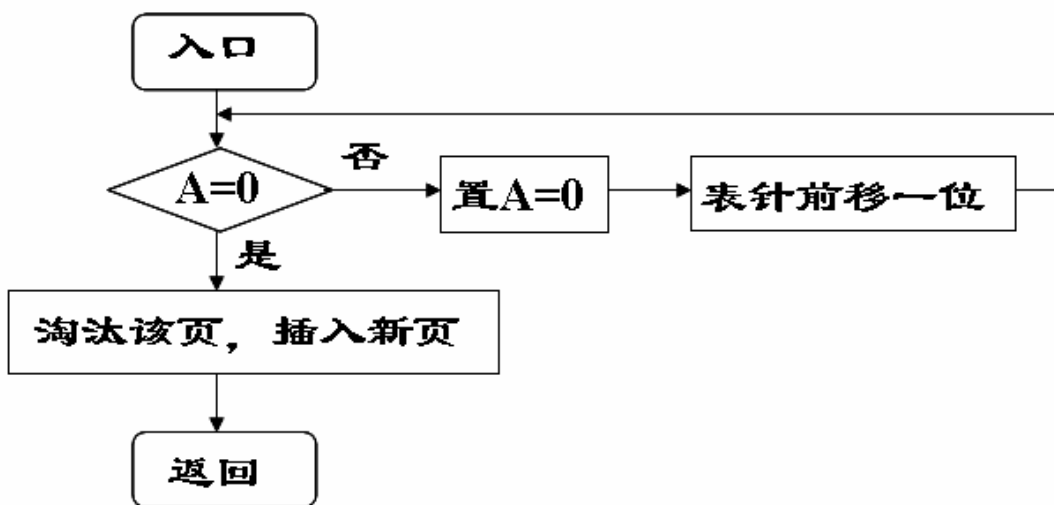
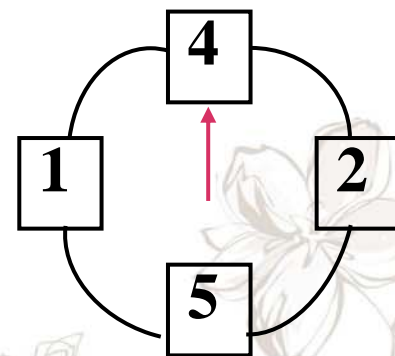
## § 3 页面置换算法及性能

### 4. 最近未用算法NRU(Not Recently Used) ——clock算法

- 为每页设置页面访问位A，当访问该页时置A = 1
- 将作业的主存页面链成循环链表
- 设查询指针，初始指向最老页

例：某时刻，一作业有4, 2, 5, 1页

按序进入主存

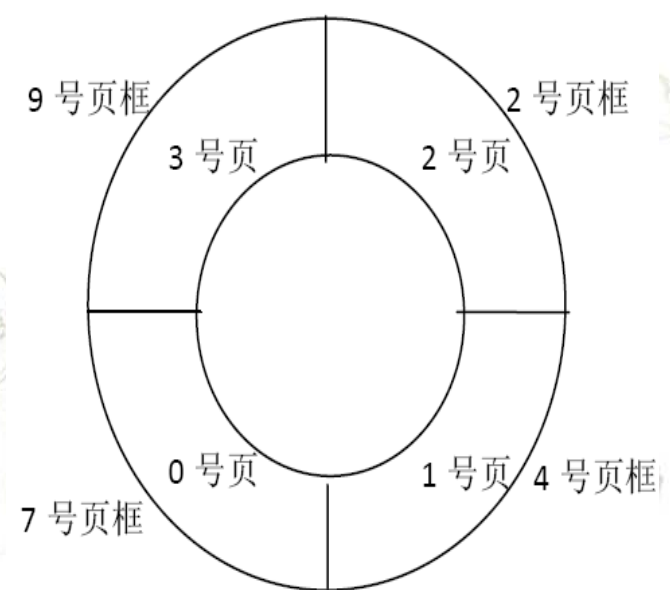




# § 3 页面置换算法及性能

**例(10年):**设某计算机的逻辑地址空间和物理地址空间均为64K,按字节编址。某进程最多需要6页数据存储空间, 页的大小为1KB, OS采用固定分配局部置换策略为此进程分配4个页框。当进程执行到时刻260时要访问逻辑地址为17CAH的数据, 请回答下列问题: (1)该逻辑地址对应的页号是多少? (2)若采用FIFO置换算法, 该逻辑地址对应的物理地址是多少? 要求给出计算过程。(3)若采用Clock算法, 该逻辑地址对应的物理地址是多少? 设搜索下一页的指针沿顺时针方向移动, 且当前指向2号页框, 示意图如下。

页号	页框号	装入时刻	访问位
0	7	130	1
1	4	230	1
2	2	200	1
3	9	160	1





## § 4 段式虚拟存储系统(请求分段)

### 一、基本原理

作业地址空间中所有段的副本均保存于辅存上。  
在作业运行前，只装入当前所需部分段。  
在运行过程中，以**缺段中断**处理方式调入所需的段。

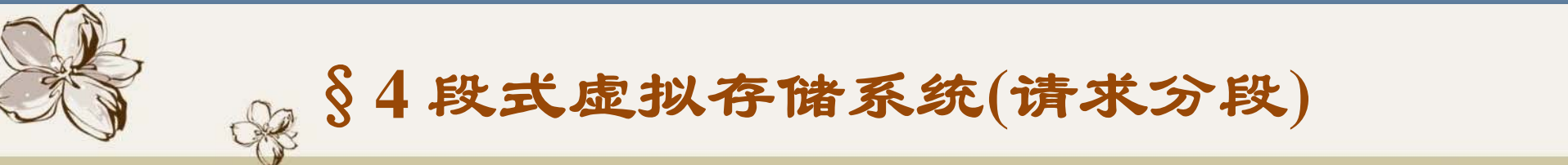
### 二、扩充的段表

段号	状态	段长	段始址	存取方式	访问位	修改位	增补位	外存地址
----	----	----	-----	------	-----	-----	-----	------

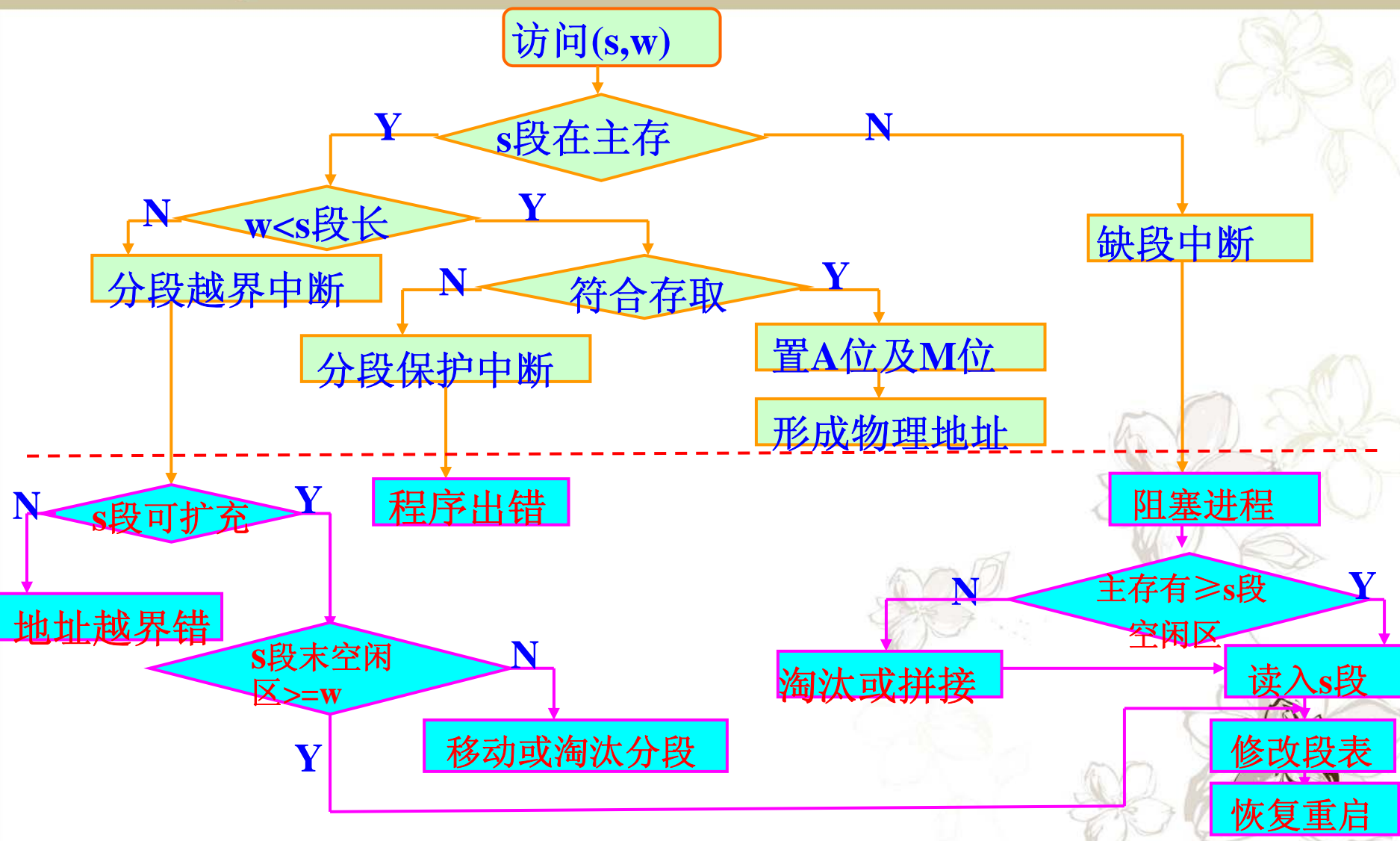
**状态：**该段是否在主存及可否共享

**整补位：**该段可否在最大段长范围内动态增长





## § 4 段式虚拟存储系统(请求分段)

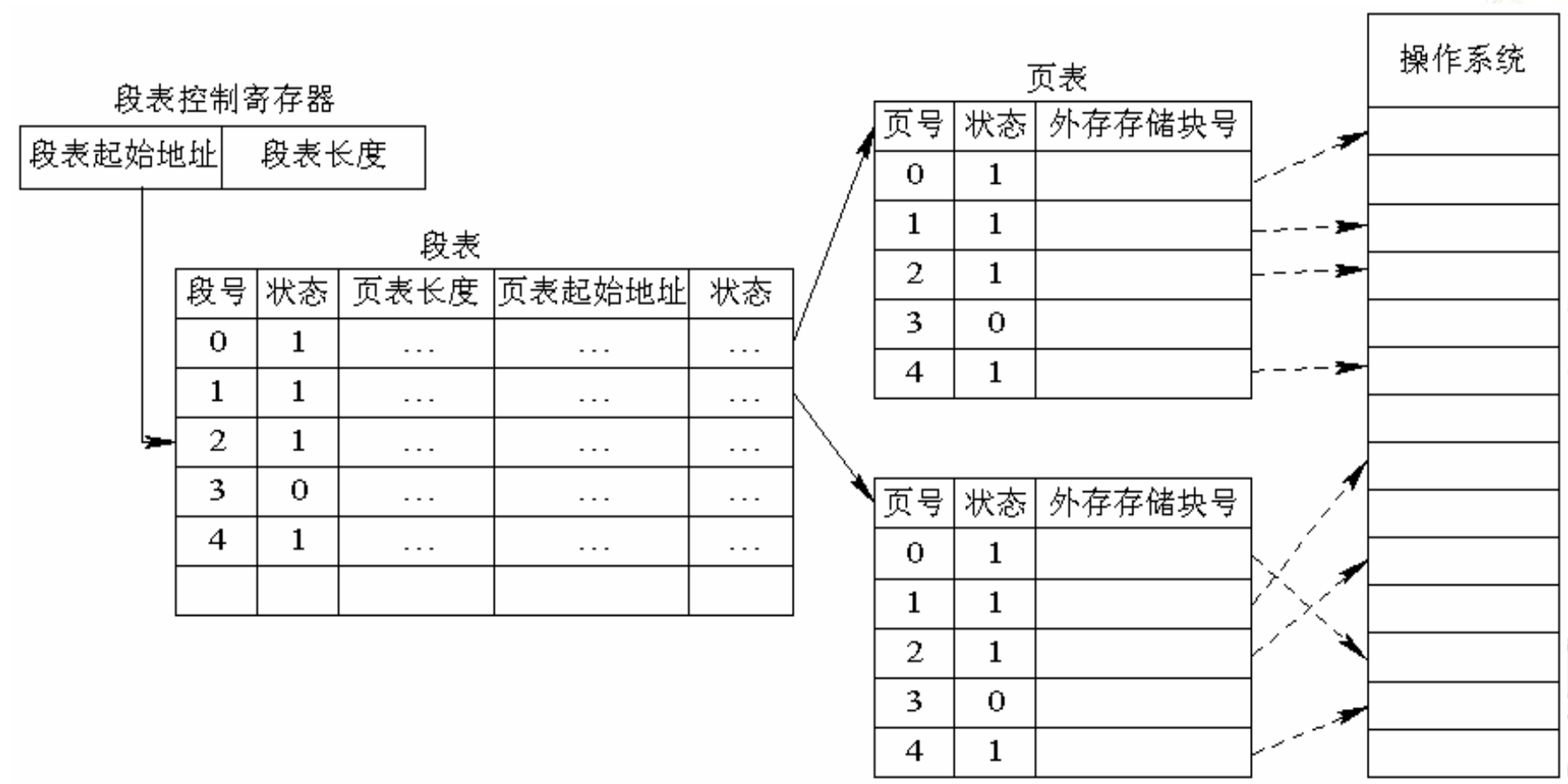






# § 5 段页式虚拟存储系统(请求段页式)

**在作业运行前，只装入部分所需段的部分所需页面**





## § 5 段页式虚拟存储系统(请求段页式)

**例1：虚拟存储中，在页表和段表中都应包含的项有 B、C、E。**

A.长度

B.中断位

C.修改位

D.内存始地址

E.外存地址

**例2（12年）：下列关于虚拟存储的叙述中，正确的是 B。**

A.虚拟存储只能基于连续分配技术

B.虚拟存储只能基于非连续分配技术

C.虚拟存储容量只受外存容量的限制

D.虚拟存储容量只受内存容量的限制



## § 5 段页式虚拟存储系统(请求段页式)

例3(哈工大): 从供选择的答案中选出与下列叙述关系最密切的存储管理方法, 把编号写在答卷对应栏内。

- A.支持多道设计, 算法简单, 但存储器碎片多 (5)。
- B.能消除碎片, 但用于存储器紧缩处理的时间长 (4)。
- C.克服了碎片多和紧缩处理时间长的缺点, 支持多道程序设计, 但不支持虚拟存储 (2)。
- D.支持虚拟存储, 但不能以自然方式提供存储器共享和存取保护机制 (3)。
- E.允许动态连接和装入, 能消除碎片, 支持虚拟存储 (1)。

供选择答案

- A~E: (1)请求段页式      (2)分页      (3)请求分页  
(4)可重定位式      (5)固定分区      (6)单一连续分配