计算机操作系统

Operating Systems

李　琳

第五章　虚拟存储器

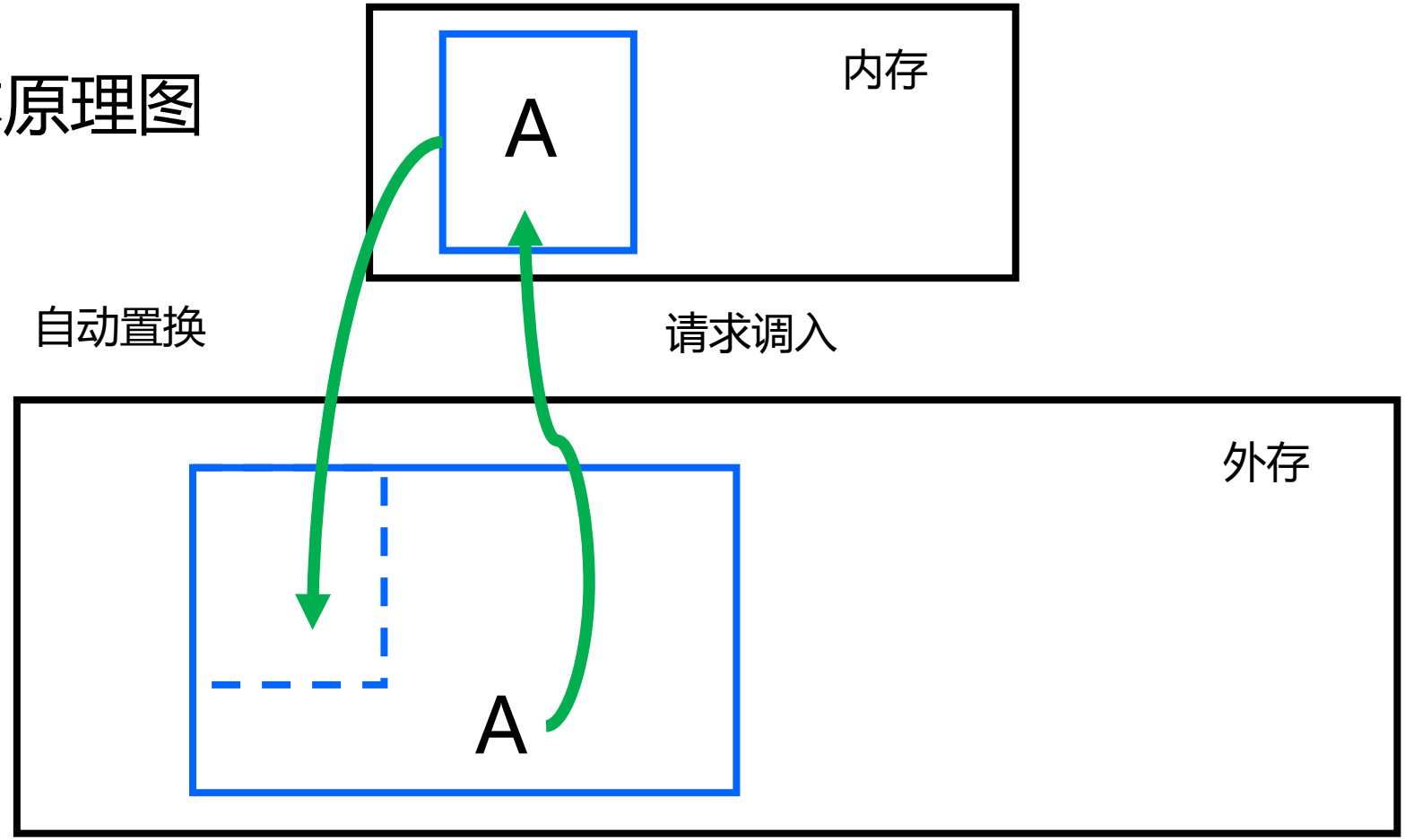
5.1 基本概念

- 存储管理的基本问题
  - ✓大作业如何在小主存上运行
  - ✓如何在给定大小的主存上运行更多程序
- 程序运行的基本特征：局部性原理
  - ✓时间局部性：一条指令被执行了，则在不久的将来它可能再被执行
  - ✓空间局部性：若某一存储单元被使用，在一定时间内，相邻的单元可能被使用

启示：是否可不将程序所有代码同时装入主存?

5.1 基本概念

基本原理图



5.1 基本概念

- 虚拟存储器
  - ✓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
  - ✓其逻辑容量由内存容量和外存容量之和所决定，其运行速度接近于内存速度，而每位的成本却又接近于外存
- 实现方法
  - ✓请求分页系统：在基本分页系统基础上，增加以页面为单位的请求调入和自动置换功能
  - ✓请求分段系统：在基本分段系统基础上，增加以分段为单位的请求调入和自动置换功能

为什么没有请求分段页系统?

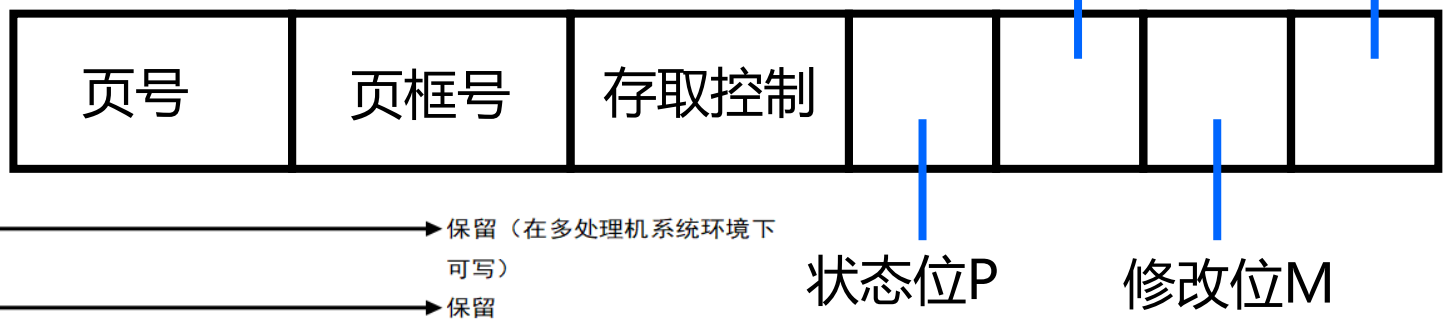
5.2 请求分页存储管理方式

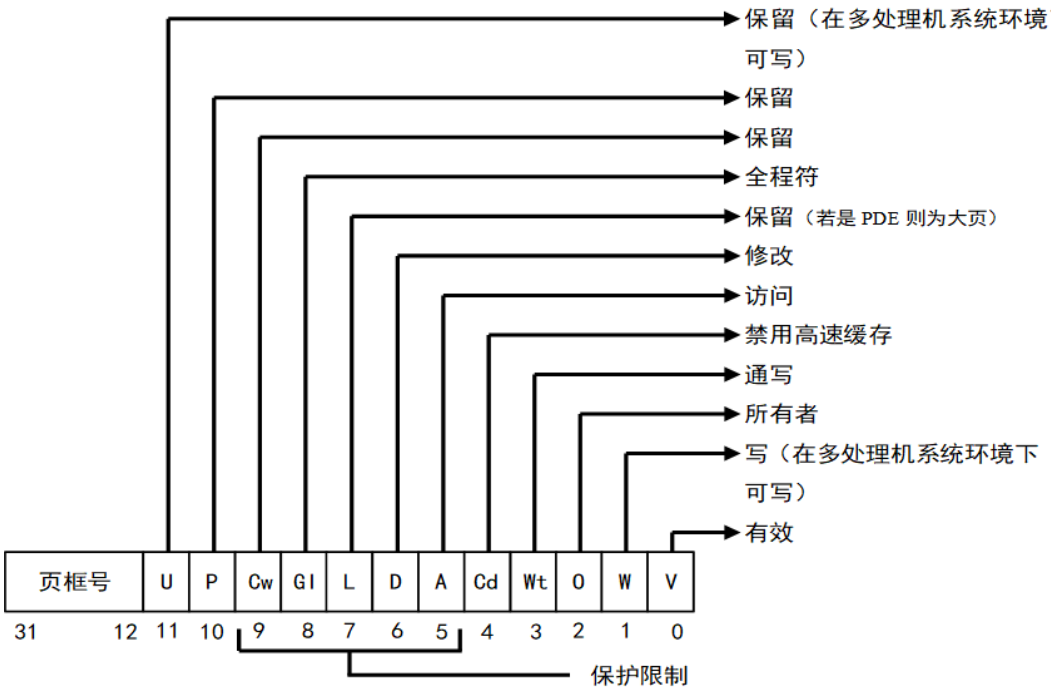
- 面临的问题
  - ✓进程的初始页面数量如何确定？中间是否可以发生变化？⟶ 分配问题
  - ✓页面可能不在主存，如何表示？ ⟶ 增加页表属性
  - ✓地址变换查找物理块可能失败，如何处理？ ⟶ 缺页中断
  - ✓应该调入哪个页面？应该换出哪个页面？ ⟶ 页面置换算法
  - ✓应该在何处调入页面？⟶ 外存虚拟空间

5.2 请求分页存储管理方式

5.2.1 请求分页中的硬件支持

- （1）新的页表





Intel x86 CPU页表

5.2 请求分页存储管理方式

5.2.1 请求分页中的硬件支持

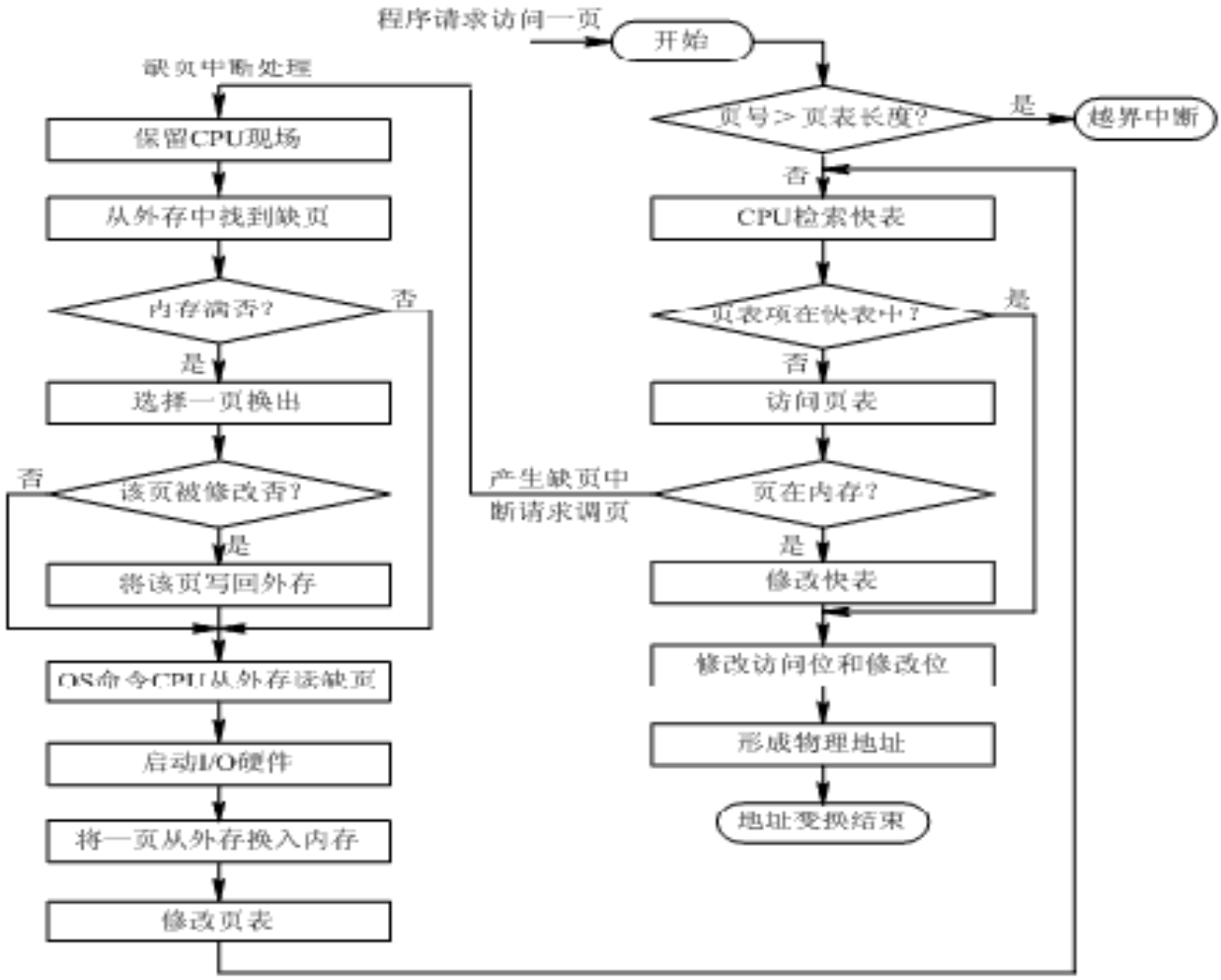
- （2）缺页中断机制
  - ✓一条指令可能发出多次缺页中断
  - ✓何时发出缺页中断

```
MOV AX, [1000];
CALL CS:[100];
JNZ CS:[2000]
CS:IP
```

5.2 请求分页存储管理方式

5.2.1 请求分页中的硬件支持

- （3）地址映射机
  - ✓右边：地址映射
  - ✓左边：中断服务程序



缺页中断程序返回位置?

试一试

1、下列关于虚拟存储器的叙述中，正确的是 （　）

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

2、进程在执行中发生了缺页中断，经操作系统处理后，应让其执行（　）指令

A、被中断的前一条　　B、被中断的后一条

C、被中断的　　D、启动时的第一条

3、段的逻辑地址形式是段号10位，段内地址20位，内存1MB，辅存10GB。那么虚拟存储器最大实际容量可能是（）

A、10GB+1MB　　B、10GB　　C、1024KB　　D、1024MB

4、某系统使用请求分页存储管理，如果页在内存中，满足一个内存请求需要120ns。如果页不在内存，则平均需要5ms。为了使用有效访问时间达到1us，要求的缺页率为多少?

5.2 请求分页存储管理方式

5.2.2 内存分配策略

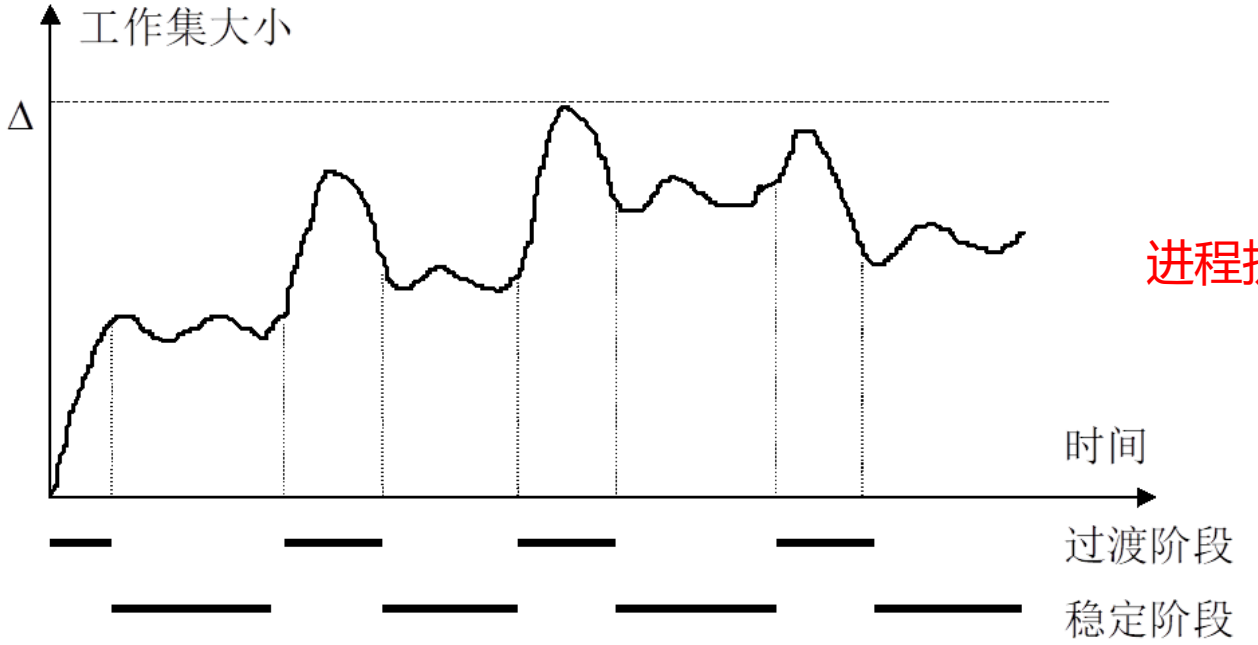
- （1）最小物理块个数：保证程序正常运行需要的最小物理块个数
  - ✓单字节指令，直接寻址：2块
  - ✓单字节指令，间接寻址：3块
  - ✓多字节指令，直接寻址：3块
  - ✓多字节指令，间接寻址：4块

由机器指令的结构决定！

5.2 请求分页存储管理方式

5.2.2 内存分配策略

- （2）工作集：驻留在物理内存中的虚拟页面的子集



进程执行时需要多少物理块？

工作集大小的变化：进程开始执行后，随着访问新页面逐步建立较稳定的工作集。当内存访问的局部性区域的位置大致稳定时，工作集大小也大致稳定；局部性区域的位置改变时，工作集快速扩张和收缩过渡到下一个稳定值。

5.2 请求分页存储管理方式

5.2.2 内存分配策略

- （3）初始分配策略
  - ✓平均分配
  - ✓按比例分配
  - ✓加权分配

5.2 请求分页存储管理方式

5.2.2 内存分配策略

- （4）运行时分配策略
  - ✓固定分配局部置换

    在进程运行期间，物理块个数不变；新页面只能置换到已分配的物理块
  - ✓可变分配全局置换

    在进程运行期间，物理块个数随时变化；新页面可以向系统申请新的物理块
  - ✓可变分配局部置换

    在进程运行期间，物理块个数随时变化；新页面只能置换到已分配的物理块；系统动态调整数量。

低限阈值 <= 缺页率 <= 高限阈值

5.2 请求分页存储管理方式

5.2.3 页面相关策略

- 何时调入
  - ✓请求式调页：需要时（缺页中断），调入页面
  - ✓预先调页：一次调入多个页面；减少I/O次数
- 何处调入
  - ✓外存对换区
  - ✓文件区
  - ✓UNIX方式：文件区（第一次）；对换区（兑换区）

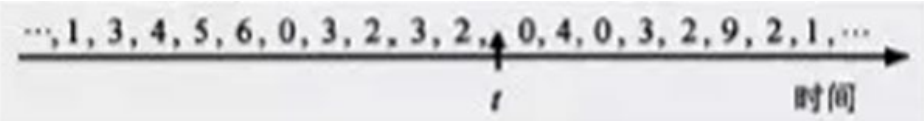
试一试

1、在请求分页系统中，页面分配策略与页面置换策略不能组合使用的是（　）

A．可变分配，全局置换　　B．可变分配，局部置换

C．固定分配，全局置换　　D．固定分配，局部置换

2、某进程访问页面的序列如下所示，若工作集的窗口大小为6，则在t时刻的工作集为（　）



A、{6，0，3，2}　　B、{2，3，0，4}

C、{0，4，3，2，9}　　D、{4，5，6，0，3，2}

3、在请求分页管理中，已修改过的页面再次装入时一般应来自（　　）

A、磁盘文件区　　B、后备作业区

C、I/O缓冲区　　D、磁盘对换区

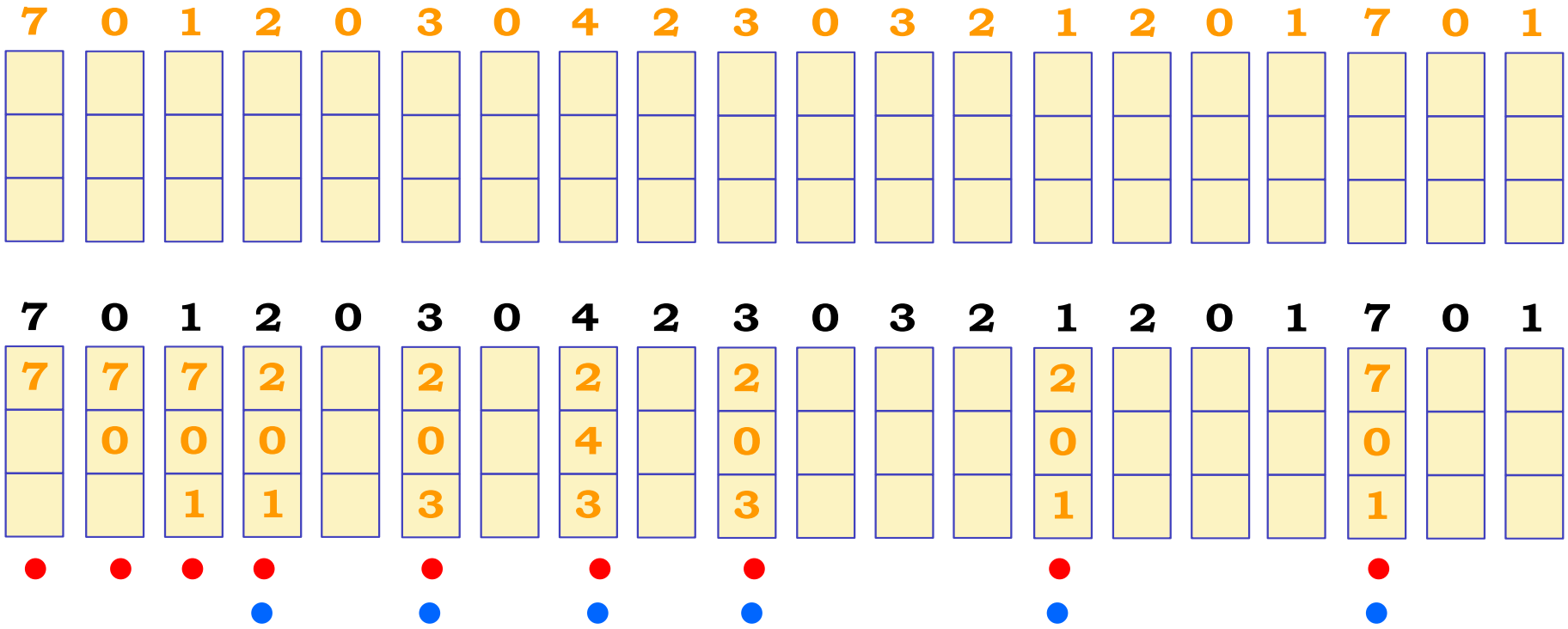
5.2 请求分页存储管理方式

5.2.4 页面置换算法

- 最佳置换算法（OPT算法）

- 基本思想
  - ✓ 选择以后再也不用的页面；
  - ✓ 没有的话，选择以后最长时间不用的页面；
- 实现
  - ✓ 无法实现，因为页面的访问顺序无法预知；
- 特点
  - ✓ 无法实现，仅具有理论意义；

最佳置换算法（OPT算法）



- 缺页次数：9次
- 置换次数：6次

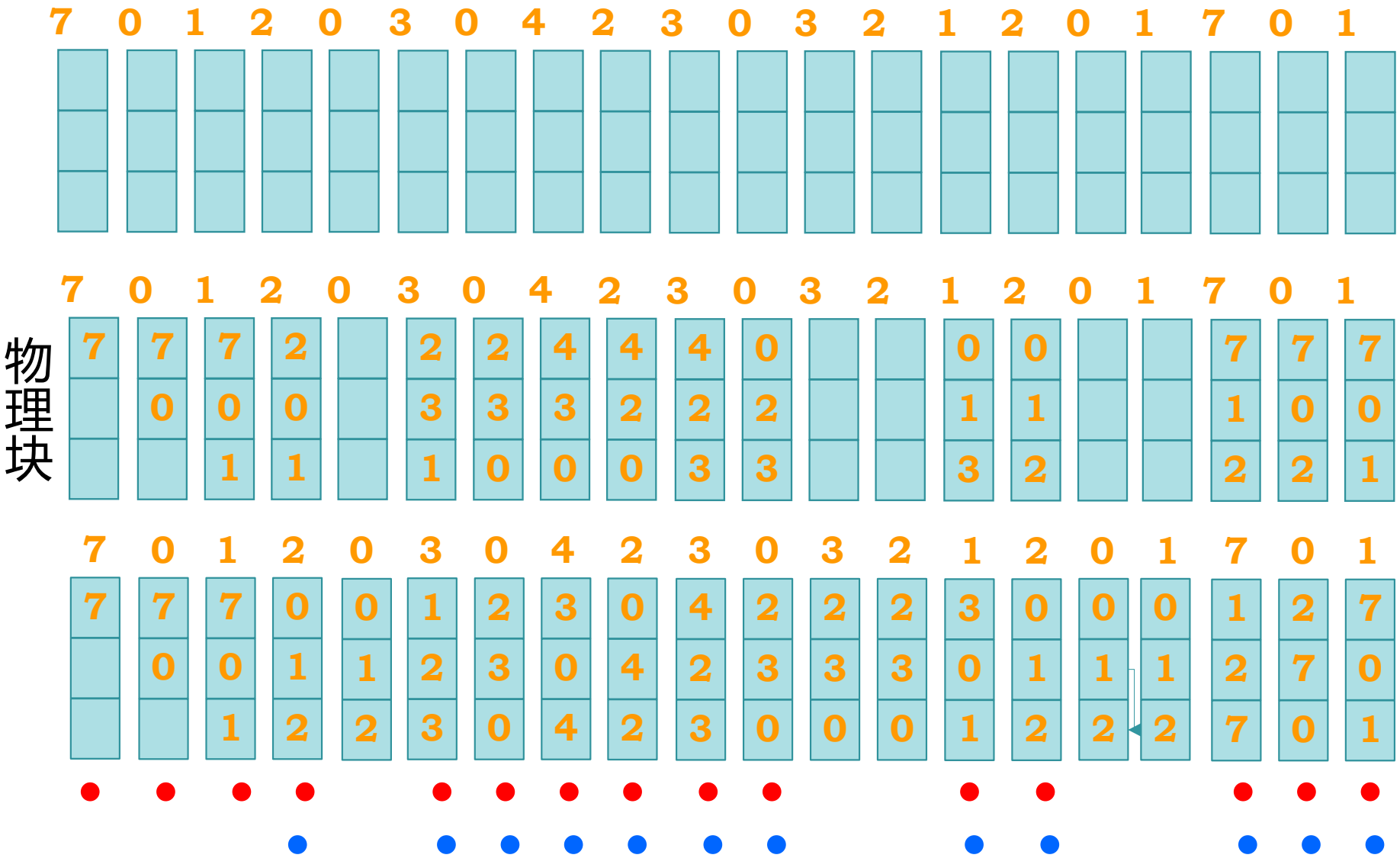
5.2 请求分页存储管理方式

5.2.4 页面置换算法

- 先进先出置换算法(FIFO)

- 基本思想
  - ✓ 程序的顺序执行特点；
  - ✓ 选择到达内存最早的页面，予以淘汰；
- 实现
  - ✓ 页面在内存中按时间排序；
- 特点
  - ✓ 效果不佳（程序不是严格顺序执行）；

先进先出置换算法（FIFO算法）



5.2 请求分页存储管理方式

5.2.4 页面置换算法

- 最近最久未使用置换算法(LRU)

■ 基本思想

✓ 基于：程序运行的局部性原理；

✓ 选择最近以来最久未使用的页面，予以淘汰；

■ 实现

✓ 移位寄存器

① 每个物理块设置一个移位寄存器，初值为0；

② 访问一次页面，高位置1；

③ 定时（100ms)将所有物理块的移位寄存器右移1位，高位补0；

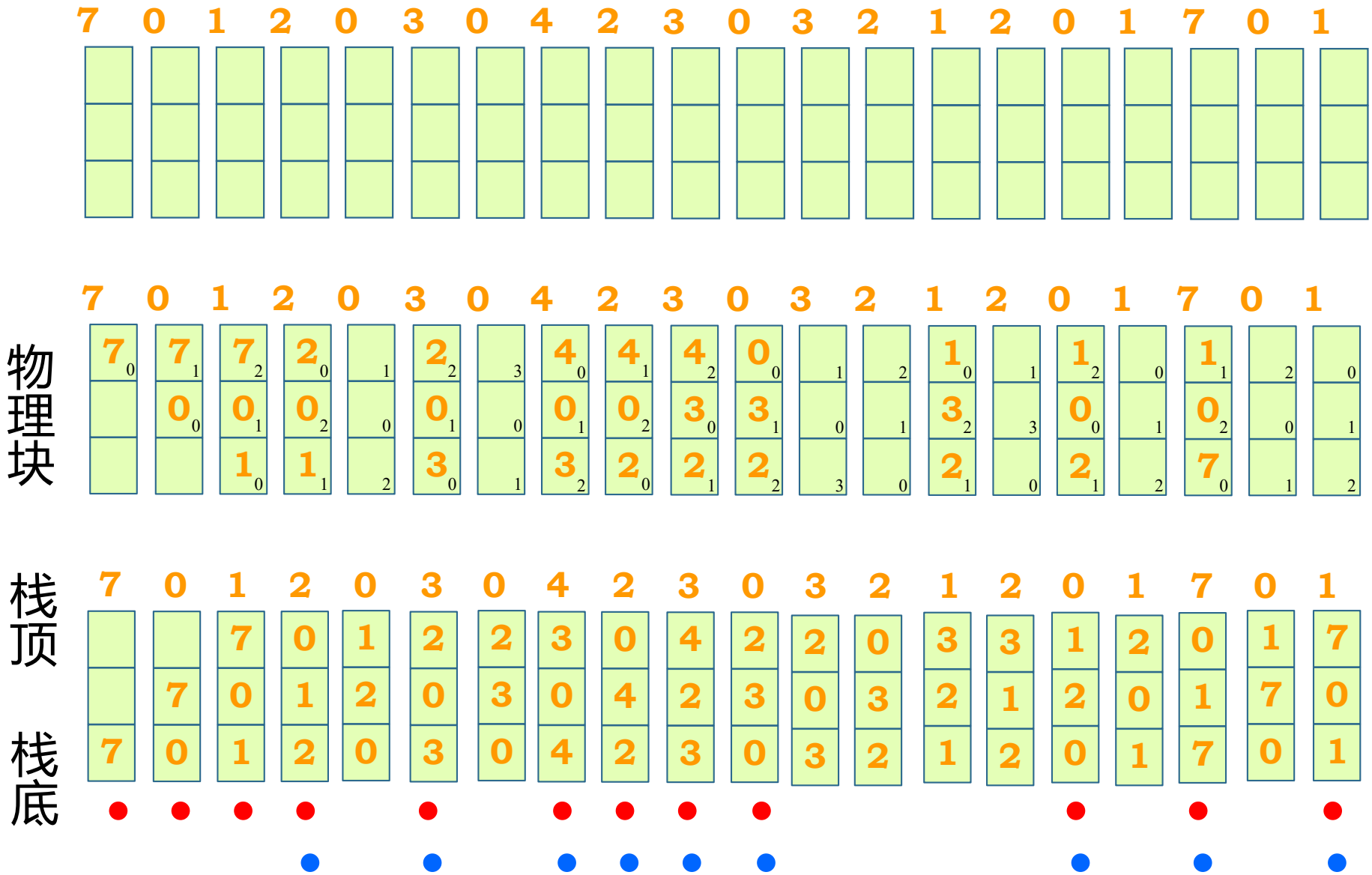
④ 选择移位寄存器数值最小的物理块，淘汰其中页面

✓ 栈的方法

■ 特点

✓ 调度性能教好；

最近最久未用置换算法（LRU算法）



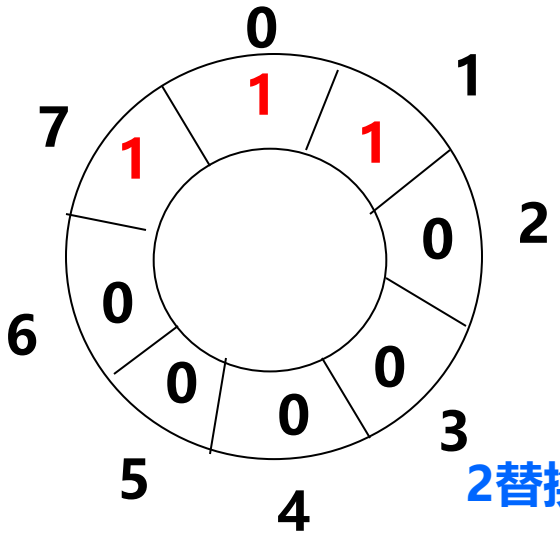
5.2 请求分页存储管理方式

5.2.4 页面置换算法

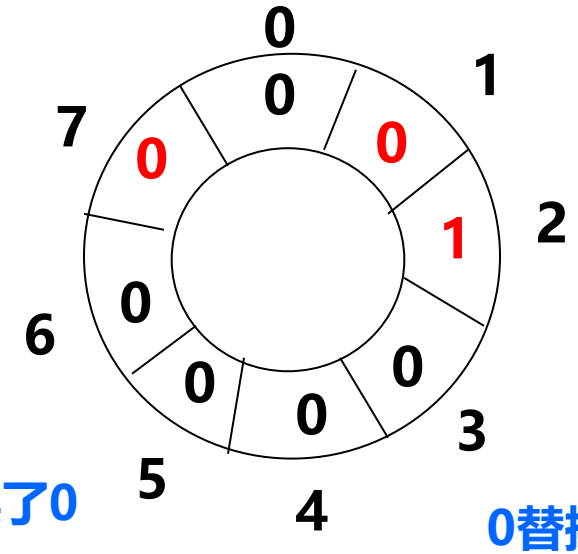
- Clock算法

- 基本思想
  - ✓ LRU的替代/近似算法；
  - ✓ 不精确计算时间；
- 实现
  - ✓ 页表增加标志位和修改位；
- 特点
  - ✓ 简单有效；

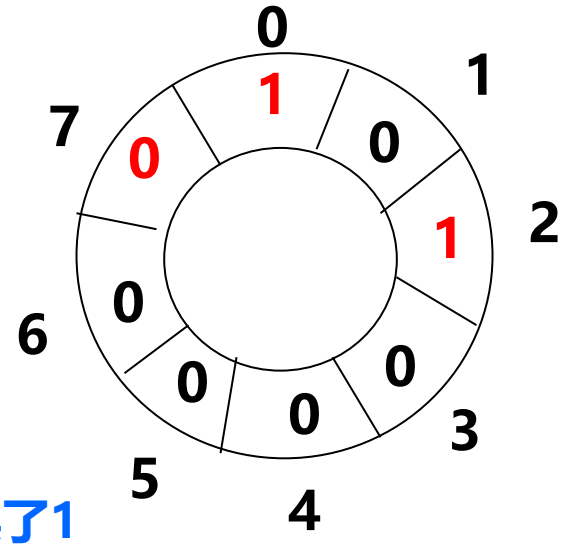
访问序列：7 0 1 2 0 3 0 4 2 3

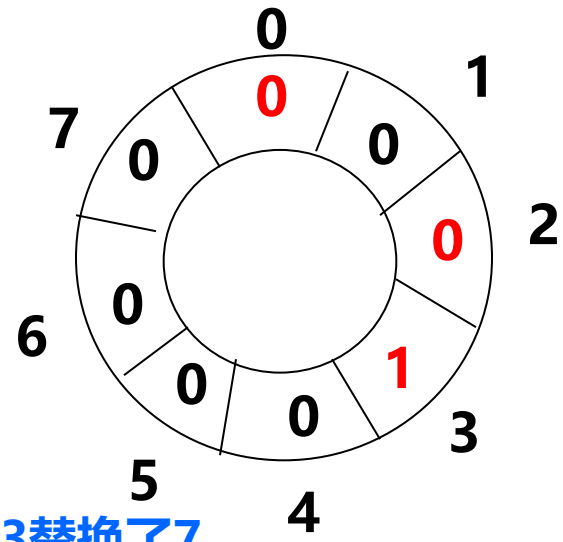


2替换了0

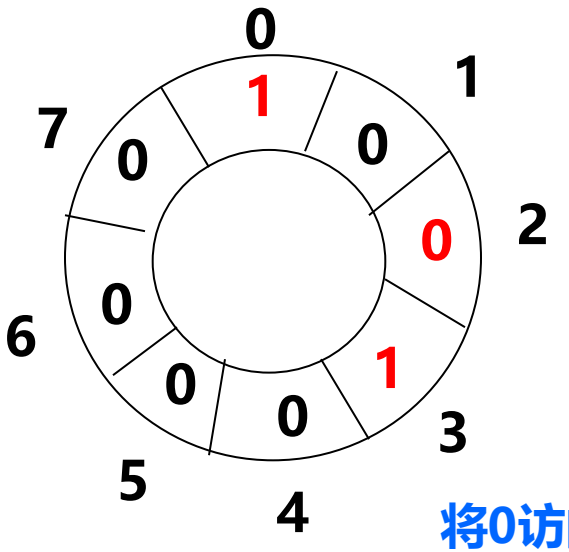


0替换了1





3替换了7



将0访问位变1

访问标志为1,则1—>0继续,访问标志为0,则替换
若第一轮没有为0的，第二轮一定能找到
访问则将0—>1

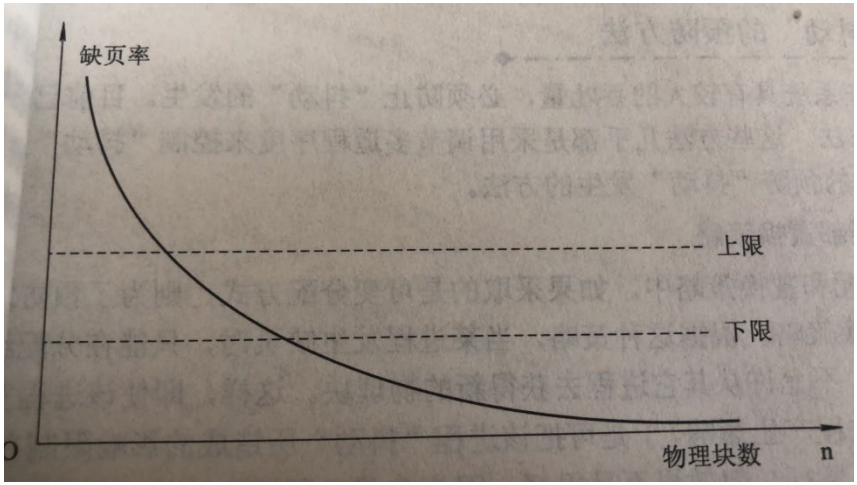
5.2 请求分页存储管理方式

5.2.5 抖动现象

- **抖动现象**
  - ✓多道程序系统：进程数量与物理块多少的平衡
  - ✓进程频繁缺页，CPU主要用于换页，CPU的利用率趋向于0
- **进程缺页频繁的原因**
  - ✓ 物理块太少，固定分配/全局分配但内存不足
  - ✓不恰当的调度算法
  - ✓糟糕的页面访问序列/置换算法
- **预防抖动的方法**
  - ✓可变分配局部置换
  - ✓考虑工作集的作业调度
  - ✓L=S准则

缺页之间的平均时间

平均缺页服务时间



请求分页系统地址映射过程分析

46.（8 分）请求分页管理系统中，假设某进程的页表内容见表 A-2。

表 A-2

页号	页框（Page Frame）号	有效位（存在位）
0	101H	1
1		0
2	254H	1

页面大小为 4KB，一次内存的访问时间为 100ns，一次快表（TLB）的访问时间为 10ns，处理一次缺页的平均时间为 10^8ns（已含更新 TLB 和页表的时间），进程的驻留集大小固定为 2，采用最近最少使用置换算法（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，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，请问：

（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？请说明理由。

解答：

(1) 根据页式管理的工作原理，应先考虑页面大小，以便将页号和页内位移分解出来。页面大小为 4KB，即 2^{12}，则得到页内位移占虚地址的低 12 位，页号占剩余高位。可得三个虚地址的页号 P 如下（十六进制的一位数字转换成 4 位二进制，因此，十六进制的低三位正好为页内位移，最高位为页号）：

2362H：P=2，访问快表 10ns，因初始为空，访问页表 100ns 得到页框号，合成物理地址后访问主存 100ns，共计 10ns+100ns+100ns=210ns。

1565H：P=1，访问快表 10ns，落空，访问页表 100ns 落空，进行缺页中断处理 10^8ns，访问快表 10ns，合成物理地址后访问主存 100ns，共计 10ns+100ns+10^8ns+10ns+100ns=100 000 220ns。

25A5H：P=2，访问快表，因第一次访问已将该页号放入快表，因此花费 10ns 便可合成物理地址，访问主存 100ns，共计 10ns+100ns=110ns。

(2) 当访问虚地址 1565H 时，产生缺页中断，合法驻留集为 2，必须从页表中淘汰一个页面，根据题目的置换算法，应淘汰 0 号页面，因此 1565H 的对应页框号为 101H。由此可得 1565H 的物理地址为 101565H。

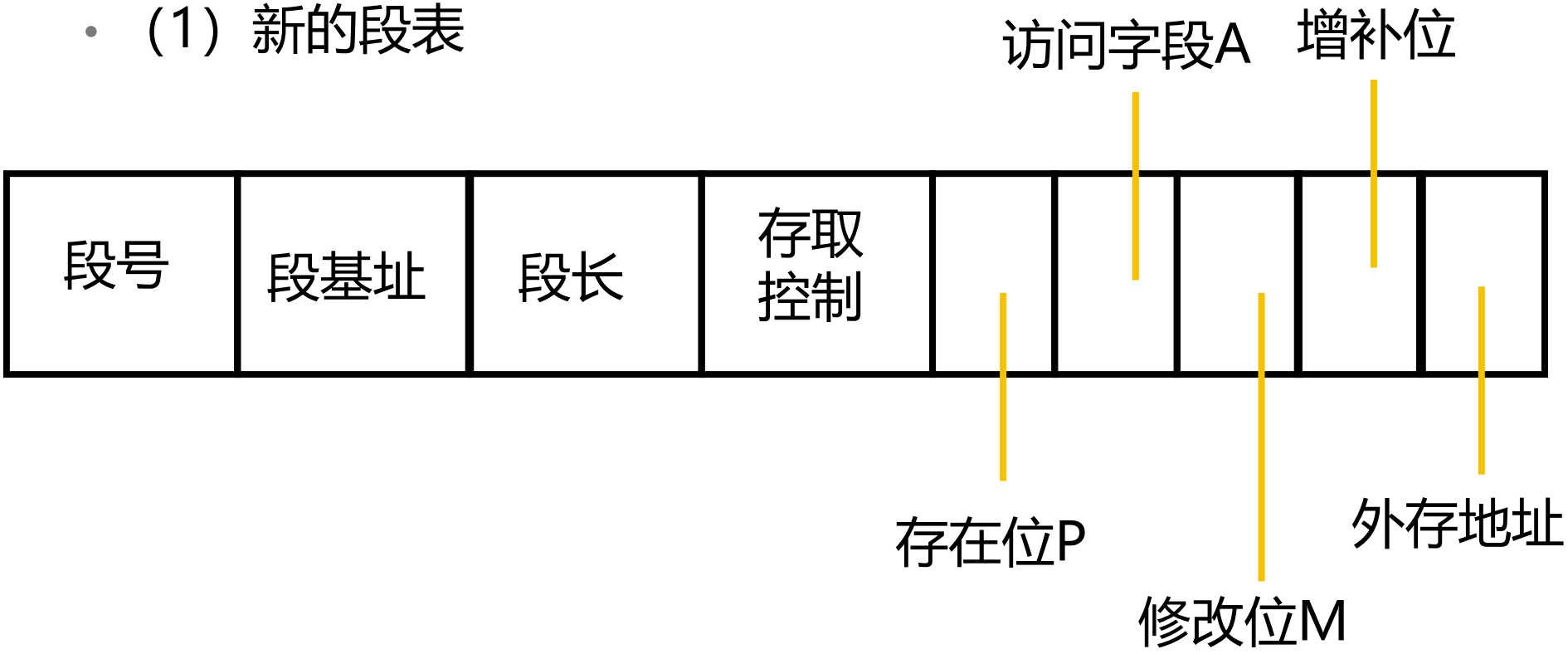
5.3 请求分段存储管理方式

- 面临的问题
  - ✓ 分段现在可能不在主存了，如何表示； ⟶ 段表新字段
  - ✓ 何时启动分段调入； ⟶ 缺段中断
  - ✓ 地址变换； ⟶ 缺段服务程序
  - ✓ 如何实现分段共享？ ⟶ 共享段表

5.3 请求分段存储管理方式

5.3.1 请求分段中的硬件支持

- （1）新的段表



5.3 请求分段存储管理方式

5.3.1 请求分段中的硬件支持

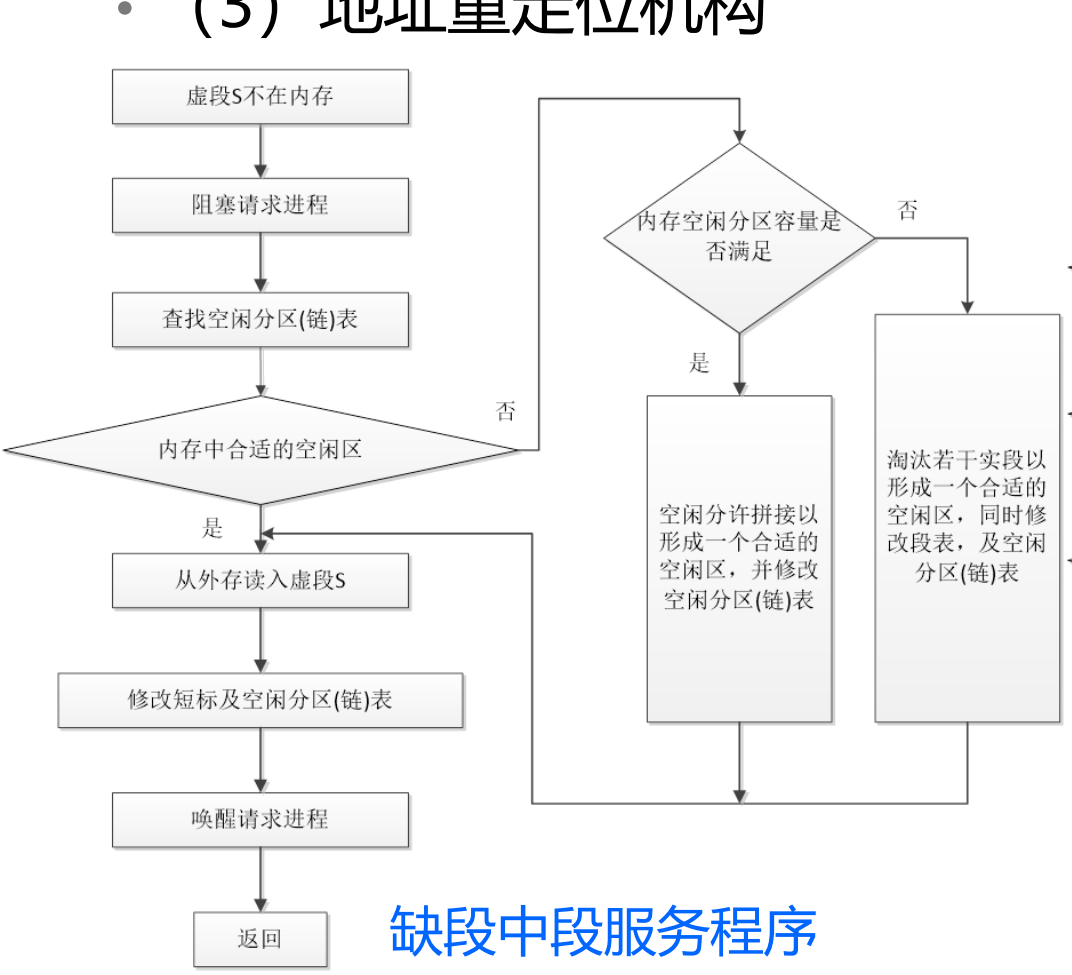
- （2）缺段中断机制
  - ✓ 指令执行期间：发出中断并响应和处理中断，返回
  - ✓ 一条指令可能发出多次缺段中断?
  - ✓ 何时发出缺段中断

```
MOV AX, [1000];
CALL CS:[100];
 JNZ CS:[2000];
CS:IP
```

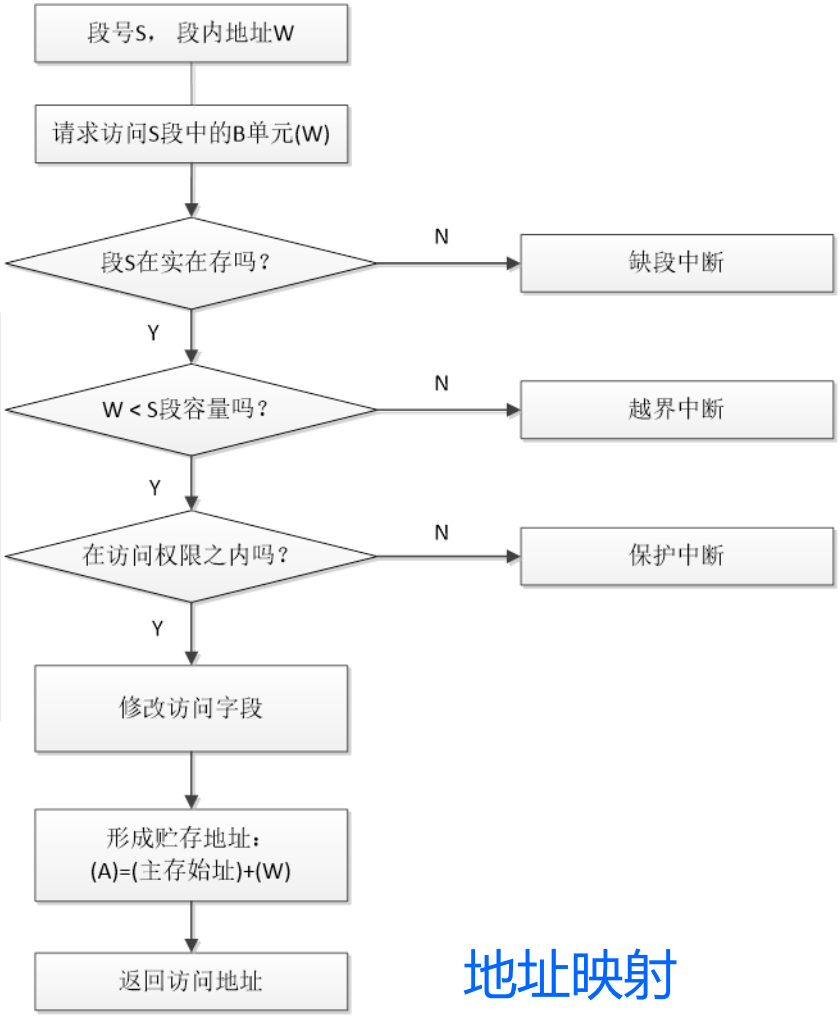
5.3 请求分段存储管理方式

5.3.1 请求分段中的硬件支持

- （3）地址重定位机构



缺段中段服务程序



地址映射

5.3 请求分段存储管理方式

5.3.2 分段共享和保护

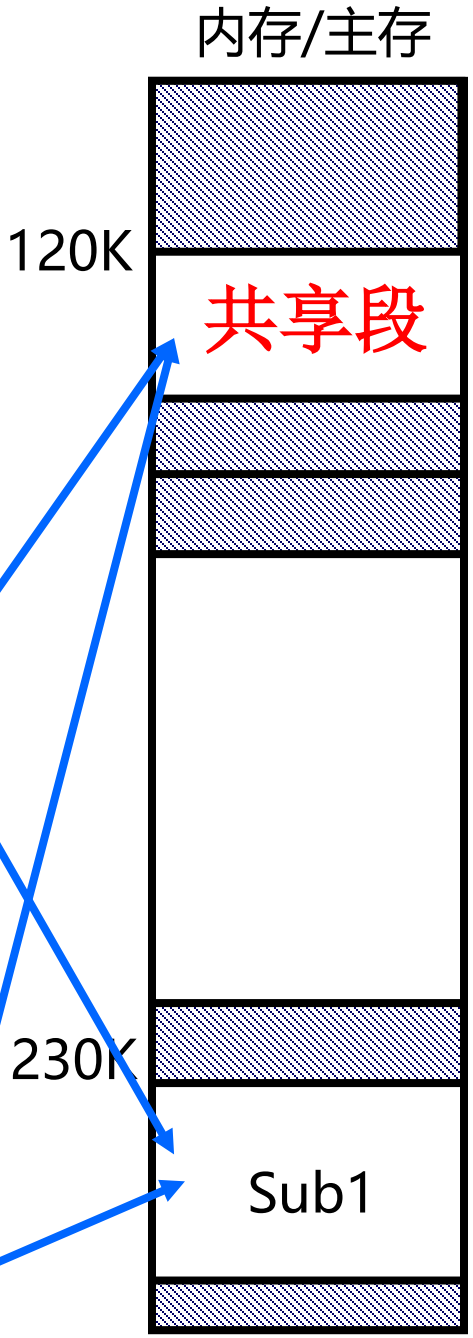
- 数据结构
  - ✓共享段表：整个系统设置一个共享段表
  - ✓进程段表：每个进程的私有段表，

A进程段表

段号	段基址	段长	存取控制
0	230K	2K	X
1	40K	1K	RW
2	120K	1.8K	X
3	20K	1.6K	X

B进程段表

段号	段基址	段长	存取控制
0	160K	2K	X
1	140K	1K	RW
2	120K	1.8K	X
3	20K	1.6K	X



5.3 请求分段存储管理方式

5.3.2 分段共享和保护

- 共享分段的分配与回收
  - ✓分配算法

    a) 检索共享分段，如果共享分段存在转c)；

    b)创建共享段表的段表项，填写分段信息，分配相应内存并调入分段;

    c) 记录本进程信息到共享段表的段表项,count+1；

    d) 复制共享段表的段表项相应信息（段基址、段长、存取控制）
  - ✓回收算法

    a) 撤消进程时，将所有该进程共享之分段的count-1 ；

    b) 当某分段count =0时，回收该分段所占据内存；

段页式存储增加虚拟存储的思考？

- 使用请求调页方式，同时还需要请求调段方式么？
- 使用预调页还是请求调页？如何利用段的局部性性质？
- 初始分配需要与段有关么？
- 如果存在共享分段，页面置换算法需要考虑这个因素么？

试一试

1、系统为某进程分配了4 个页框，该进程已访问的页号序列为2,0,2,9,3,4,2,8,2,3,8,4,5，若进程要访问的下一页的页号为7，依据LRU 算法，应淘汰页的页号是（　　）

A．2　B．3　C．4　D．8

2、在页式虚拟存储管理系统中，采用某些页面置换算法，会出现Belady异常现象，即进程的缺页次数会随着分配给该进程的页框个数的增加而增加。下列算法中，可能出现Belady异常现象的是　（　　）

Ⅰ．LRU算法　Ⅱ．FIFO算法　Ⅲ．OPT算法

A．仅Ⅱ　B．仅Ⅰ、Ⅱ　C．仅Ⅰ、Ⅲ　D．仅Ⅱ、Ⅲ

3、某系统采用改进型CLOCK置换算法，页表项中字段A为访问位，M为修改位。A=0表示页最近没有被访问，A=1表示页最近被访问过。M=0表示页没有被修改过，M=1表示页被修改过。按（A,M）所有可能的取值，将页分为四类：（0，0），（1，0），（0，1）和（1，1），则该算法淘汰页的次序为（　　）

A、（0，0），（0，1），（1，0），（1，1）

B、（0，0），（1，0），（0，1），（1，1）

C、（0，0），（0，1），（1，1），（1，0）

D、（0，0），（1，1），（0，1），（1，0）

试一试

4、一个进程分配得到4个页框，装入时间和上次访问时间如下表。请问LRU算法将置换的页面存放在第（　）页框中，如果是FIFO算法呢（　）

A、2　B、1　C、0　D、3

页面	装入时间	上次访问时间
0	126	280
1	230	265
2	140	270
3	110	285

5、在一个请求式分页系统中，
目前系统的利用率如下：
CPU操作　　：20%
分页磁盘的I/O操作：97.7%
其它I/O设备　　：5%
下列方法是否可以提高CPU利用率，分别说出你的理由。
1）安装一个更加快速的CPU；2）增加一个容量更加大的磁盘；
3）增加更多的内存；　4）增加页面的大小。

作业

1、P177 13题 注意比较结果是从影响缺页率的因素上考虑

2、假设有一个按需调页存储器，页表放在寄存器中。处理一个页错误，当有空的帧可用或被置换的帧没有被修改过时要用8ms，当被置换的帧被修改过时用20ms存储器存取时间为100ns。假设被置换的页中有70%被修改过，有效存取时间不超过200ns时，最大可以接受的缺页率为多少？

3、某系统采用页式虚拟存储管理，贮存每块为128个字节，现在要把一个128 × 128的二维数组置初值为“0”。在分页时把数组中的元素每一行放在一页中，假定系统只分给用户一页数据区。

（1）对如下数据段，执行完要产生多少次缺页中断？

```
var A：array[ 1．．128]of array［l．．128」of integer；
    for j ：=1 to 128
        do for i:=1 to 128
            do A[i,j]: =0；
```

（2）为减少缺页中断的次数，请改写上面的程序，使之仍能完成所要求的功能，并统计缺页次数。

4、

46.（8 分）设某计算机的逻辑地址空间和物理地址空间均为 64KB，按字节编址。若某进程最多需要 6 页（Page）数据存储空间，页的大小为 1KB，操作系统采用固定分配局部置换策略为此进程分配 4 个页框（Page Frame）。在时刻 260 前的该进程访问情况见表 B-2（访问位即使用位）。

表 B-2

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

当该进程执行到时刻 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）若采用先进先出（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）若采用时钟（CLOCK）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（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，示意图如下图所示）。