操作系统
第十五讲

张涛

Review

覆盖与交换技术

分页存储管理

Today we focus on…

虚拟存储器

请求分页存储管理

请求分页原理

页面置换算法

性能分析

方案评价

4.5 虚拟存储器(Virtual storage)

- 解决的问题：内存小，作业大、作业多的矛盾
- 实现的原理：程序执行时的局部性原理：1968年， Denning.P
  - 时间局限性
  - 空间局限性
- “扩充”主存 —— 虚拟存贮技术
- 实现方式：操作系统统一管理各级存储器；内存中只存放当前要执行的程序部分，其余的保存在外存上，OS根据需要随机地将需要的部分对换到内存执行

- 虚拟存储技术的物质基础：二级存储器结构和动态地址转换机构
  - 二级存储结构：内存、外存
  - 动态地址转换：在操作系统的改造下，内、外存地址统一编址，扩大寻址空间
- 虚空间独立于实空间
  - 虚空间>>实空间(也可以虚空间<实空间)
  - 指令中表示地址的位数越长，可寻址空间越大，但不是无限的，一般为32位、64位。
  - 主存＋辅存≠虚存(仅与地址结构有关)

虚空间大小

- 虚空间大小
  - 虚空间的逻辑大小 = 可寻址范围
  - 虚空间的实际大小 = 内存+外存对换区
- 例：32位操作系统的可寻址范围是2^{32}=4GByte，Windows98系列系统。
- 例：在window系统盘根目录下，有兑换文件——外存对换区。如XP系统的pagefile.sys文件

引入虚拟存储技术的好处

- **大程序：** 可在较小的可用内存中执行较大的用户程序；
- **大的用户空间：** 提供给用户可用的虚拟内存空间通常大于物理内存(real memory)
- **并发：** 可在内存中容纳更多程序并发执行；
- **易于开发：** 与覆盖技术比较，不必影响编程时的程序结构

虚拟存储技术的特征

- 不连续性：物理内存分配的不连续，虚拟地址空间使用的不连续（数据段和栈段之间的空闲空间，共享段和动态链接库占用的空间）
- 部分交换：与交换技术相比较，虚拟存储的调入和调出是对部分虚拟地址空间进行的；
- 虚拟扩充：通过物理内存和快速外存相结合，提供大范围的虚拟地址空间
  - 总容量不超过物理内存和外存交换区容量之和
- 多次对换：程序运行期间，分别在内、外存中的程序多次对换

虚存管理三大策略

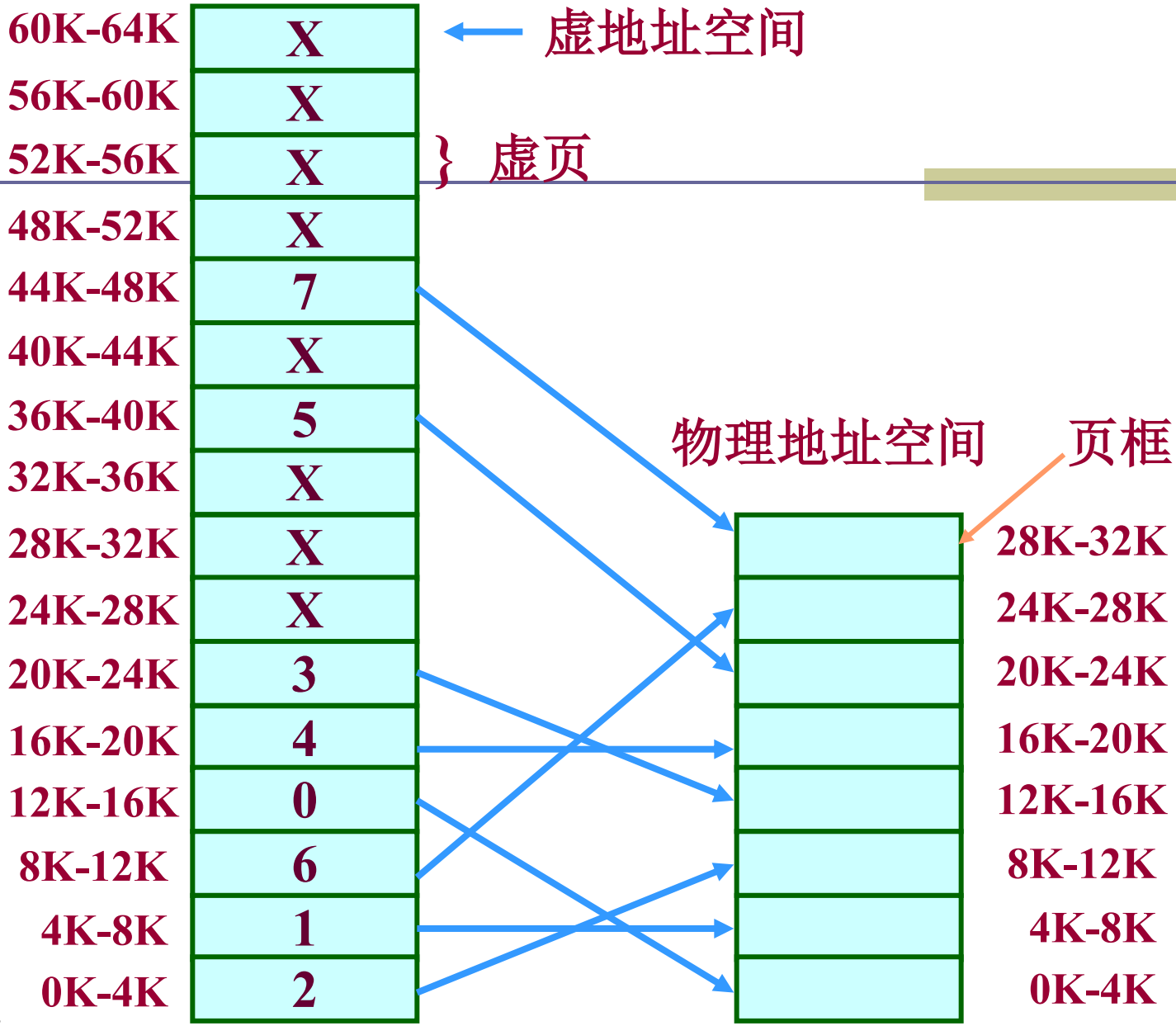
- 调入策略(把哪部分装入主存)
- 放置策略(放在主存什么地方)
- 淘汰策略(主存不足时，把哪部分淘汰出主存。)

4.6 请求分页存储管理

- 请求分页存储管理——虚拟页式存储管理
  - 基本原理
  - 页面置换（淘汰）算法
  - 性能分析

4.6.1 基本原理

- **目标：** 实现小内存大作业
- **实现方法：** 作业运行时，只将当前的一部分装入内存其余的放在辅存，一旦发现访问的页不在主存中，则发出缺页中断，由O.S将其从辅存调入主存，如果内存无空块，则选择一个页淘汰。
- 分页存储管理系统根据请求装入所需页面的方法称为请求分页存储管理。
- 必须解决以下两个问题：
  - （1）当程序要访问的某页不在内存时，如何发现这种缺页情况？
  - （2）当需要把外存上的某个页面调入内存时，此时内存中没有空闲块应怎么办？


60K-64K
X
虚地址空间
56K-60K
X
52K-56K
X
虚页
48K-52K
X
44K-48K
7
40K-44K
X
36K-40K
5
32K-36K
X
28K-32K
X
24K-28K
X
20K-24K
3
16K-20K
4
12K-16K
0
8K-12K
6
4K-8K
1
0K-4K
2
物理地址空间
页框
28K-32K
24K-28K
20K-24K
16K-20K
12K-16K
8K-12K
4K-8K
0K-4K

基本问题

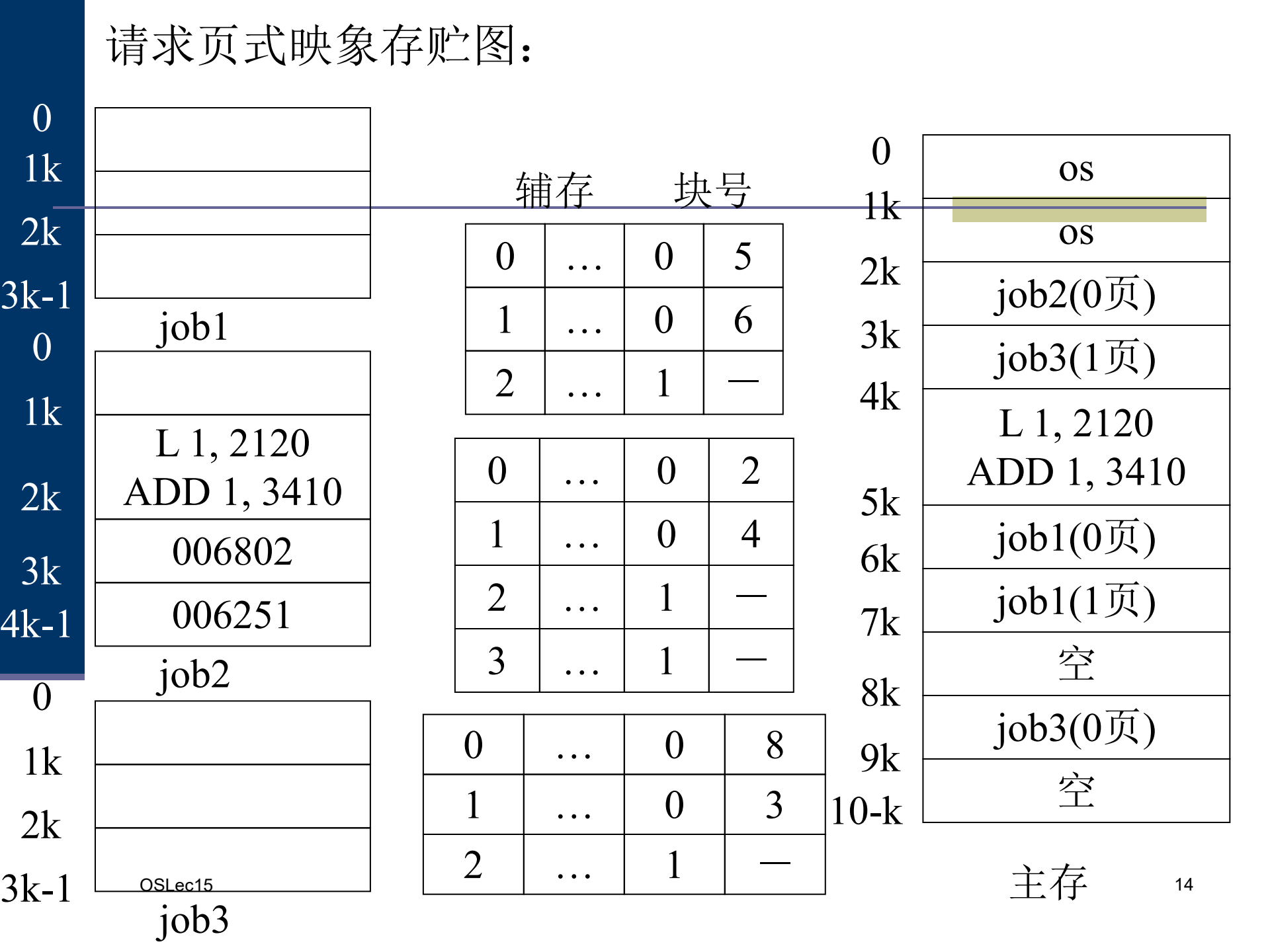
- 一、怎样发现页不在内存？
  - 方法：扩充页表，增加两个数据项：中断位、辅存的位置

页号	块号	中断位	辅存

1. 不在内存
0. 在内存

  - 1：发生缺页中断，调用OS的调页程序。

请求页式映象存贮图：


0
1k
2k
3k-1
job1
0
1k
2k
3k
4k-1
L 1, 2120
ADD 1, 3410
006802
006251
job2
0
1k
2k
3k-1
job3
辅存
块号
0 … 0 5
1 … 0 6
2 … 1 —
0 … 0 2
1 … 0 4
2 … 1 —
3 … 1 —
0 … 0 8
1 … 0 3
2 … 1 —
0
1k
2k
3k
4k
5k
6k
7k
8k
9k
10-k
os
os
job2(0页)
job3(1页)
L 1, 2120
ADD 1, 3410
job1(0页)
job1(1页)
空
job3(0页)
空
主存

基本问题

- 二、如何处理缺页—缺页中断管理？
  - 发现中断、调入所缺页
  - 如果内存不足，选一页淘汰，选择哪一页呢？
  - 淘汰时，需要注意什么？

引用位
- “0”表示没有访问过
- “1”表示已被访问过

修改位
- 0修改位 —— 写回辅存
- 1未修改过 —— 不必写回辅存

页表表项

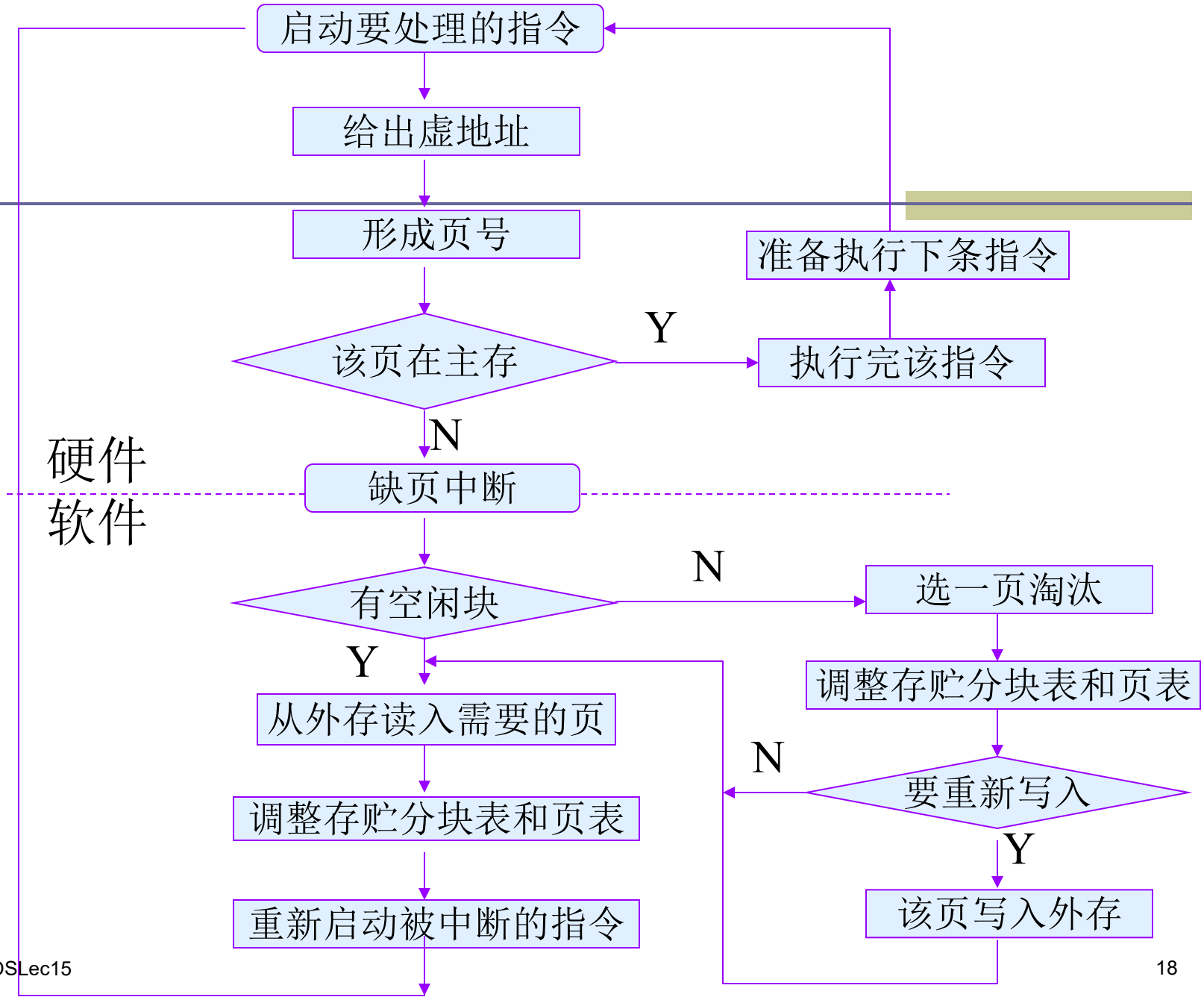
- 页号、驻留位、内存块号、外存地址、访问位、修改位
  - 驻留位（中断位）：表示该页是在内存还是在外存
  - 访问位：根据访问位来决定淘汰哪页（由不同的算法决定）
  - 修改位：查看此页是否在内存中被修改过

页号	中断位	内存块号	外存地址	访问位	修改位

缺页中断（Page Fault）处理

- 在地址映射过程中，在页表中发现所要访问的页不在内存，则产生缺页中断。
- 操作系统接到此中断信号后，就调出缺页中断处理程序，根据页表中给出的外存地址，准备将该页调入内存
- 此时应将缺页的进程挂起（调页完成唤醒）
- 如果内存中有空闲块，则分配一个块，将要调入的页装入该块，并修改页表中相应页表项目的驻留位及相应的内存块号
- 若此时内存中没有空闲块，则要淘汰某页（若被淘汰页在内存期间被修改过，则要将其写回外存）

指令执行和缺页中断处理



思考

- 缺页中断同一般中断都是中断。区别？
  - 相同点：
    - 保护现场 中断处理 恢复现场
  - 不同点：
    - 一般中断是一条指令完成后中断，缺页中断是一条指令执行时中断
    - 一条指令执行时可能产生多个缺页中断。如指令可能访问多个内存地址，这些地址在不同的页中。

4.6.2 页面置换（淘汰）算法

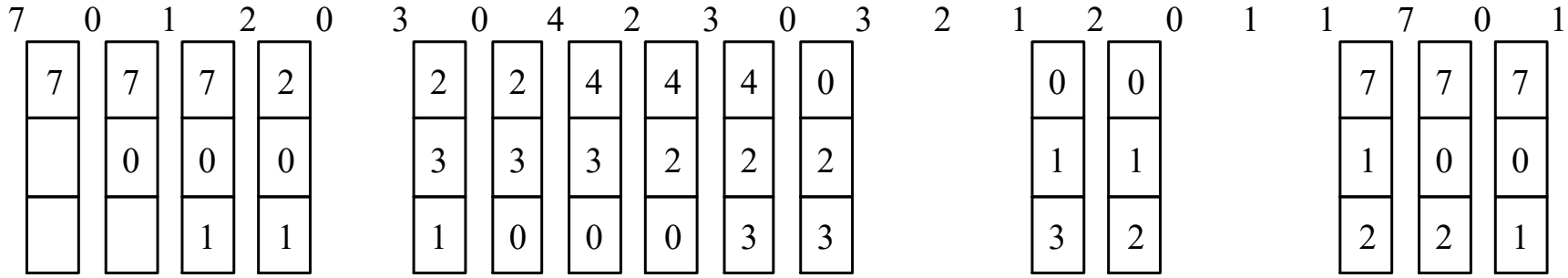
- 功能：需要调入页面时，选择内存中哪个物理页面被置换。称为replacement policy。
- 目标：把未来不再使用的或短期内较少使用的页面调出，通常只能在局部性原理指导下依据过去的统计数据进行预测；
- 页面锁定(frame locking)：用于描述必须常驻内存的操作系统的关键部分或时间关键(time-critical)的应用进程。实现方法为在页表中加上锁定标志位(lock bit)。

常用页面置换算法

- 先进先出页面算法（FIFO）：选择在内存中驻留时间最长的页并淘汰之
- 最近最久未使用置换算法（LRU，Least Recently Used）：淘汰没有使用的时间最长的页
- 最佳页面算法（OPT，Optimal）：淘汰以后不再需要的或最远的将来才会用到的页面
- 最不经常使用（LFU，Least Frequently Used）：选择访问次数最少的页面淘汰之

先进先出(FIFO)页面置换算法

引用率

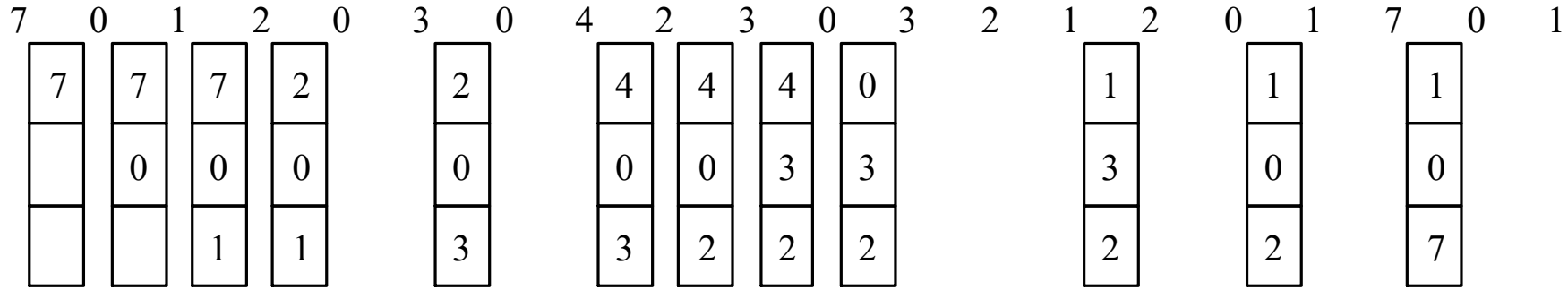


页框

$$缺页率为=\frac{15}{20}\times 100\%=75\%$$

最近最久未使用(LRU)置换算法

引用率



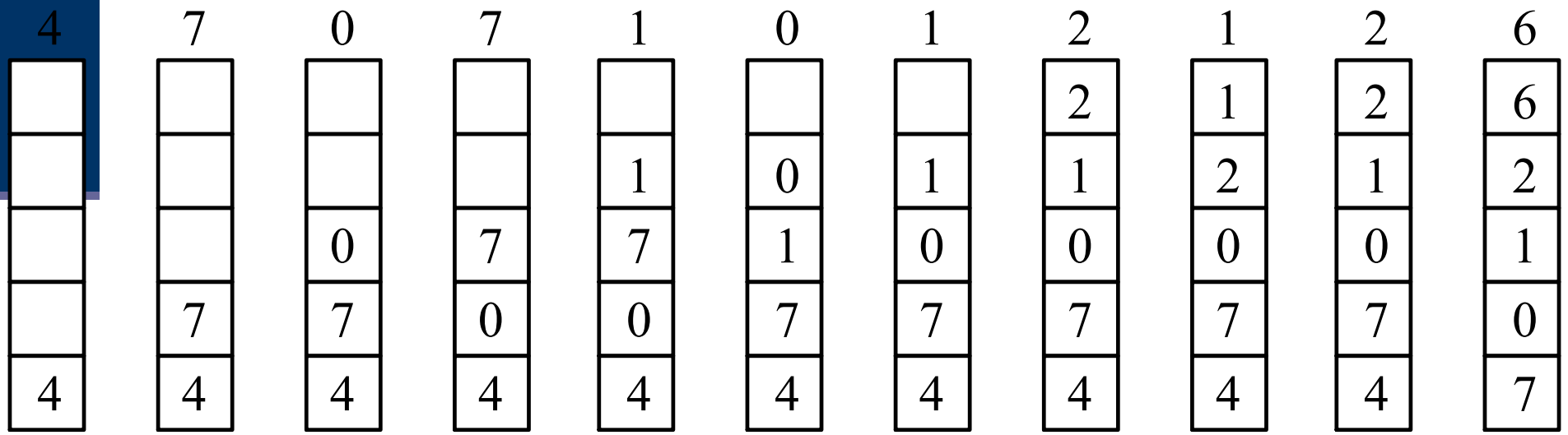
$$\text{缺页率为} = \frac{12}{20} \times 100\% = 60\%$$

LRU实现代价

- LRU硬件解法：系统为每页设置一个寄存器R，每当访问这一页时，将该页对应的寄存器R置1，以后每个时间间隔将所有的R左移一位，当淘汰一页时就选择R值最大的页。

$$R=R_{n-1}R_{n-2}R_{n-3}\ \ldots\ R_2R_1R_0$$

- LRU软件解法：设置一个页号栈，当访问一个页面时将它的页号压入栈，若页号栈中有与刚压入栈顶的相同的页号，则从页号栈中抽出，保证页号栈中无相同的页号。当系统要淘汰一节时，总是从页号栈底取出一个页号淘汰，即淘汰的页是最久未使用的。

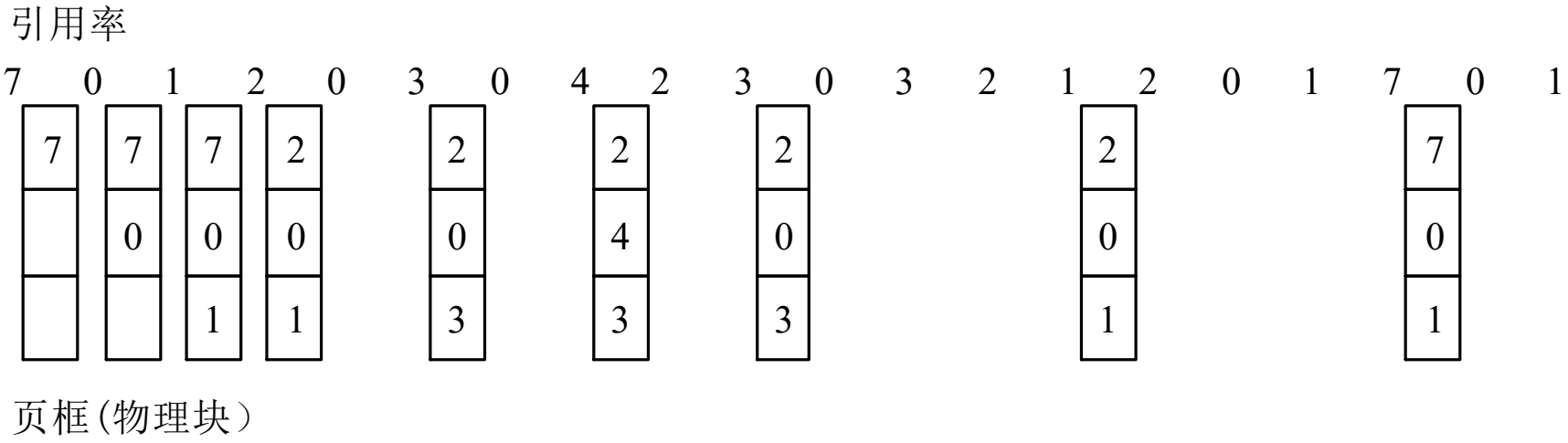


LRU近似算法

- 在页表中增加一访问位，每当访问一页时，将该页的访问位由硬件置1，软件周期性地将所有访问位置0。在时间T内，访问过的页其访问位为1，反之为0，淘汰为0 的页。
- 缺点：T难定。太小，访问位为0的页相当多，所选的不一定是最久未用的。太大，所有页的引用位可能都为1，找不到合适的淘汰页。

最佳(Optimal)置换算法

- 最佳置换算法是由Belady于1966年提出的一种理论上的算法。
- 其所选择的被淘汰页面，将是以后永不使用的， 或许是在最长(未来)时间内不再被访问的页面。
- 采用最佳置换算法，通常可保证获得最低的缺页率。



置换算法举例

OPT	4	3	2	1	4	3	5	4	3	2	1	5
页1	4	3	2	1	1	1	5	5	5	2	1	1
页2		4	3	3	3	3	3	3	3	5	5	5
页3			4	4	4	4	4	4	4	4	4	4
	x	x	x	x	3	3	x	3	3	x	x	3

共缺页中断7次

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	3	3	x	x	x

共缺页中断10次

FIFO	4	3	2	1	4	3	5	4	3	2	1	5
页1	4	3	2	1	4	3	5	5	5	2	1	1
页2		4	3	2	1	4	3	3	3	5	2	2
页3			4	3	2	1	4	4	4	3	5	5
	x	x	x	x	x	x	x	3	3	x	x	3

共缺页中断9次

4.6.3 性能分析

- 影响缺页次数的因素
  - (1) 分配给进程的物理块数
  - (2) 页本身的大小
  - (3) 程序的编制方法
  - (4) 页面淘汰算法

常驻集(resident set)

- 常驻集指虚拟页式管理中给进程分配的物理页面数目
- 常驻集与缺页率的关系：
  - 每个进程的常驻集越小，则同时驻留内存的进程就越多，可以提高并行度和处理器利用率；另一方面，进程的缺页率上升，使调页的开销增大。
  - 进程的常驻集达到某个数目之后，再给它分配更多页面，缺页率不再明显下降。该数目是“缺页率－常驻集大小”曲线上的拐点(curve)。
- 试验分析表明：对于所有程序来说，要使其有效地工作，它在主存中的页面数应不低于它的总页面数的一半。

颠簸（抖动）

- 在虚存中，页面在内存与外存之间频繁调度，系统效率急剧下降，甚至导致系统崩溃。这种现象称为颠簸或抖动。
- Question：常驻集越大，缺页中断率越小吗？

$$缺页中断率=\frac{F(不成功的访问次数)}{S(成功的访问次数)+F}=\frac{失败}{成功+失败}$$

Belady现象

- Belady现象的描述：一个进程P要访问M个页，OS分配N个内存页面给进程P；对一个访问序列S，发生缺页次数为PE（S,N）。当N增大时，PE(S, N)时而增大，时而减小。
- Belady现象的原因：FIFO算法的完全没有考虑进程访问内存的动态特征，即被置换的页面并不是进程不会访问的。

Belady现象的例子：进程P有5页程序访问页的顺序为：
1，2，3，4，1，2，5，1，2，3，4，5；

FIFO	1	2	3	4	1	2	5	1	2	3	4	5
页 0	1	2	3	4	1	2	5	5	5	3	4	4
页 1		1	2	3	4	1	2	2	2	5	3	3
页 2			1	2	3	4	1	1	1	2	5	5
缺页	x	x	x	x	x	x	x	√	√	x	X	√

FIFO	1	2	3	4	1	2	5	1	2	3	4	5
页 0	1	2	3	4	4	4	5	1	2	3	4	5
页 1		1	2	3	3	3	4	5	1	2	3	4
页 2			1	2	2	2	3	4	5	1	2	3
页 3				1	1	1	2	3	4	5	1	2
缺页	x	x	x	x	√	√	x	x	x	x	x	x

请求式分页系统的性能分析

设一次不发生缺页的页面访问时间＝MA≈0.1微秒

设一次缺页的页面访问时间

＝ 缺页中断服务时间 ＋缺页读入时间

＝Rp≈ 25000微秒

从磁盘中读入

所以一次有效访问时间

＝（1－P）MA＋P×Rp＝0.1＋24999×P

设△＝有效访问时间－MA

若期望 $\frac{\triangle}{MA} < 10\%$ 则 $P < 4\times10^{-7}$

即希望请求分页系统比分页系统只多10%的开销时

页的共享和保护

- 多个作业同时运行相同的程序时，可以使它们共享页面。办法是使这些相关进程的逻辑空间的页指向相同的内存块。
- 产生的问题：有的页面可以共享，有的不能
- 解决的办法：在页表中设置读写控制字段来进行页面保护。
- 分页系统中实现页面保护具有一定困难：
  - 程序页面划分对用户是透明的，可共享与不能共享的程序有可能划分到同一个页面中。

页面共享的特点

- 优点：
  - 共享页面有效地减小了运行多道程序所需要的主存容量，可能使一个给定的系统去支持更多的用户(在分时系统中尤为重要)。
- 缺点：
  - 被共享的程序部分不见得被包含在一个完整的页面中这样如果共享页中有进程的私用数据时也被共享了，不利于保密。
  - 共享部分的起始单元在各进程的地址空间划分成页面过程中，在各自的页面中的相对偏移量(页内地址)不同也使共享更为困难。

例：设某一分时系统中，现有40个用户要求运行，每个用户都要求执行正文编辑程序，如果这个正文编辑程序有30k代码并且每个用户在编辑时需要10k的数据区，这个系统必须提供(10+30)×40=1600k内存才能同时支持40个用户上机。

若采用页面共享，将30k的正文编辑程序作为块则仅需10×40＋30=430k内即可。

请求分页管理方案的评价

- 优点：
  - 提供了虚存管理方式，作业地址空间不再受实存容量的限制；
  - 更有效的利用了主存，方便于多道程序运行，方便了用户；
- 缺点：
  - 为处理缺页中断，增加了处理机时间的开销。用时间的代价换取了空间的扩大；
  - 可能因作业地址空间过大或程序数目过多等造成系统抖动；为此采取措施会增加的系统的复杂度。

What you need to do?

- 复习课本4.4节的内容
- 课后作业：习题11、14、15、21

See you next time!