操作系统
第十四讲

张涛

Review

存储管理的基本概念

存储系统的组织与任务

地址再定位

早期的存储管理

分区存储管理

本次内容

覆盖与交换技术

分页存储管理

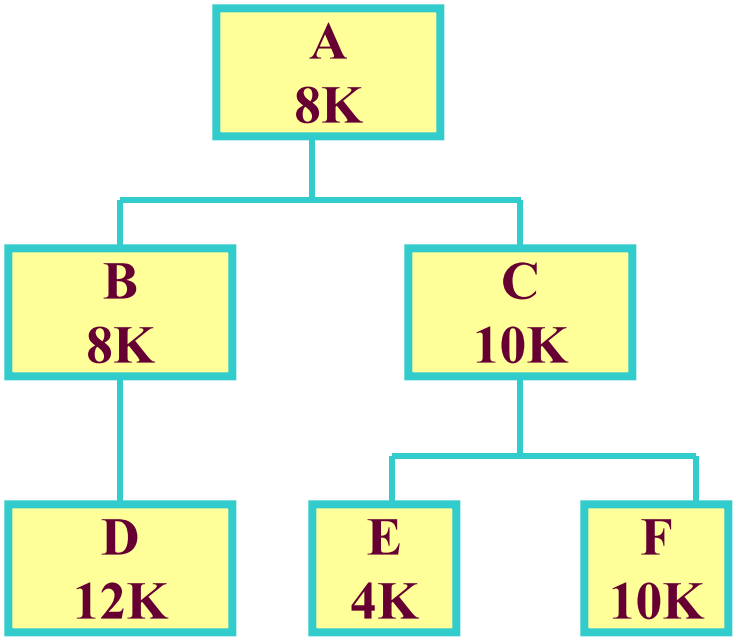
4.3 覆盖与交换技术

- 引入：在多道环境下扩充内存的方法，用以解决在较小的存储空间中运行较大程序时遇到的矛盾。
- **覆盖技术：**一个作业的若干程序段，或几个作业的某些部分共享某一个存储空间。
- **交换技术：**系统将内存中某些进程暂时移到外存，把外存中某些进程换进内存，占据前者所占用的区域。

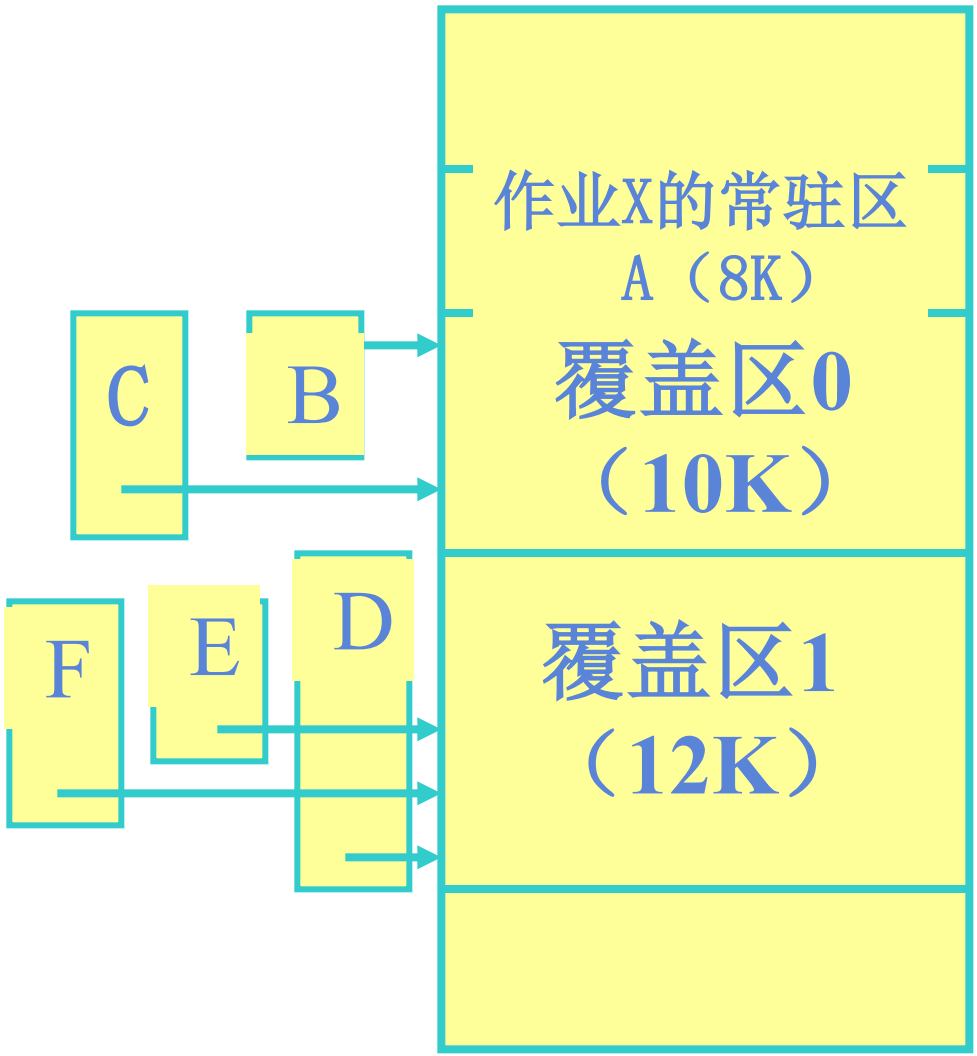
覆盖技术(overlay)

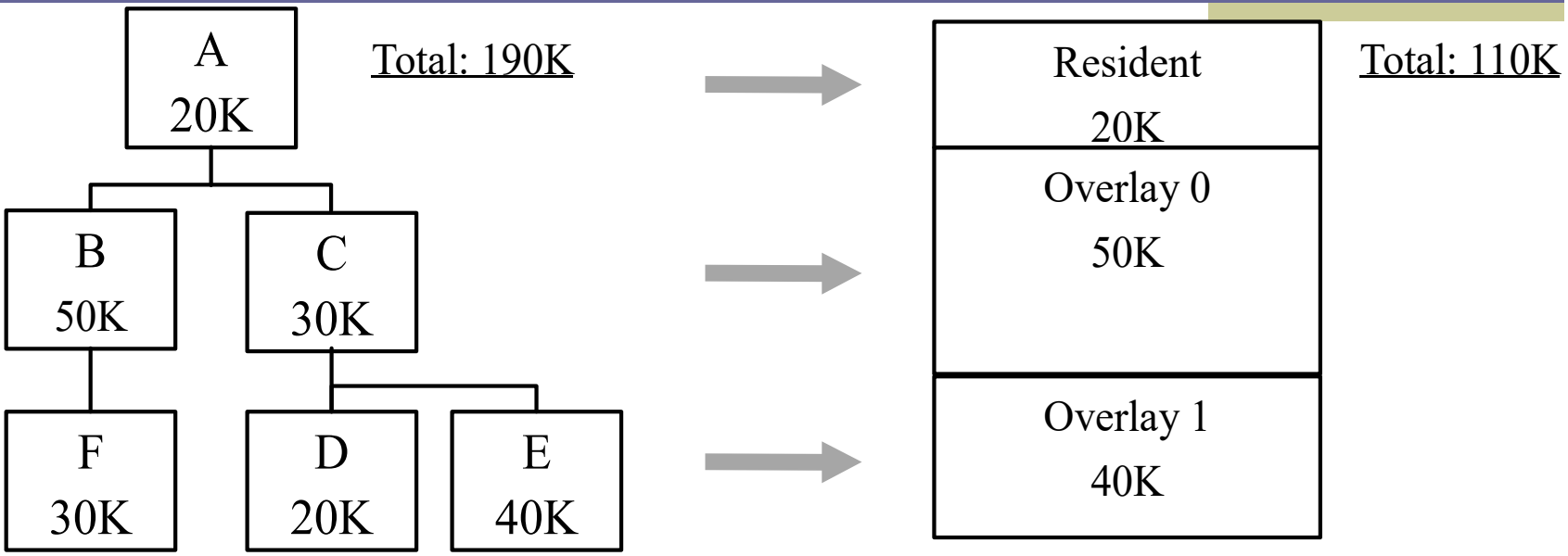
- **原理：** 一个程序的几个代码段或数据段，按照时间先后来占用公共的内存空间。（并不是作业的每一部分都是时时要用的）
  - 把程序划分为若干个功能上相对独立的程序段，按照其自身的逻辑结构将那些不会同时执行的程序段共享同一块内存区域
  - 程序段先保存在磁盘上，当有关程序段的前一部分执行结束，把后续程序段调入内存，覆盖前面的程序段（内存“扩大”了）
  - 一般要求作业各模块之间有明确的调用结构，程序员要向系统指明覆盖结构，然后由操作系统完成自动覆盖

图示


A
8K
B
8K
C
10K
D
12K
E
4K
F
10K

作业X的调用结构


作业X的常驻区
A（8K）
覆盖区0
（10K）
覆盖区1
（12K）
C
B
F
E
D



注： 另一种覆盖方法：(100K)

- A(20K)占一个分区：20K；
- B(50K)、D(20K)和E(40K)共用一个分区：50K；
- F(30K)和C(30K)共用一个分区：30K；

覆盖技术分析

- **缺点：**
  - 编程时必须划分程序模块和确定程序模块之间的覆盖关系，增加编程复杂度。
  - 从外存装入覆盖文件，以时间延长来换取空间节省。
- **实现：** 函数库（操作系统对覆盖不得知），或操作系统支持

- **例子：** 目前这一技术用于小型系统中的系统程序的内存管理上，MS-DOS的启动过程中，多次使用覆盖技术；启动之后，用户程序区TPA的高端部分与COMMAND.COM暂驻模块也是一种覆盖结构。

交换技术(swapping)

- 引入：多个程序并发执行，可以将暂时不能执行的程序送到外存中，从而获得空闲内存空间来装入新程序。
- 原理：暂停执行内存中的进程，将整个进程的地址空间保存到外存的交换区中（换出swap out），而将外存中由阻塞变为就绪的进程的地址空间读入到内存中，并将该进程送到就绪队列（换入swap in）。
  - 交换单位为整个进程的地址空间。
  - 与分区存储管理配合使用
  - 又称作"对换"或"滚进/滚出(roll-in/roll-out)"；

交换技术实现中的几个问题

- **选择原则：** 将哪个进程换出内存？
  - 例子：分时系统，时间片轮转法或基于优先数的调度算法，在选择换出进程时，换出要长时间等待的进程。
- **交换时机的确定：** 何时需发生交换？
  - 只要不用就换出（或很少再用）
  - 只在内存空间不够或有不够的危险时换出
- **交换时需要做哪些工作？**
  - 需要一个盘交换区：必须足够大以存放用户程序的内存映像的拷贝；必须对这些内存映像直接存取。
- **换入回内存时位置的确定**
  - 换出后再换入的内存位置一定要在换出前的原来位置上吗？
  - 受地址映射技术的影响，即绝对地址产生时机的限制

交换技术的特点

- 优点：
  - 增加并发运行的程序数目，并且给用户提供适当的响应时间；
  - 编写程序时不影响程序结构。
- 缺点：
  - 对换入和换出的控制增加处理机开销；
  - 程序整个地址空间都进行传送，没有考虑执行过程中地址访问的统计特性。

覆盖与交换的比较

- 共同点：
  - 进程的程序和数据主要放在外存，当前需要执行的部分放在内存，内外存之间进行信息交换
- 不同点：
  - 交换发生在进程或作业之间，而覆盖发生在同一进程或作业内。
  - 与覆盖技术相比，交换技术不要求用户给出程序段之间的逻辑覆盖结构；

4.4 分页存储管理

为什么要引进分页技术?

分区存储管理会产生碎片，需移动分区

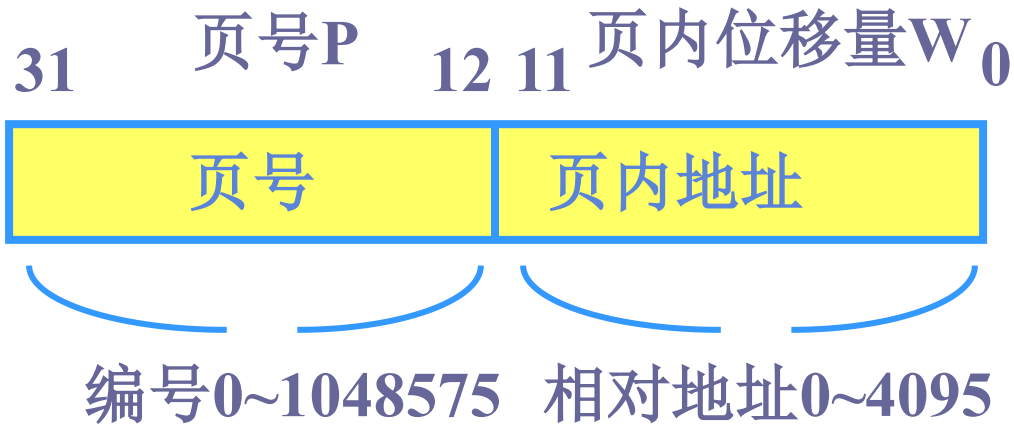
- 分页存储管理的基本思想
- 页表
- 地址变换机构
- 两级页表和多级页表
- 快表

分页存储管理的基本思想

- 主存分成多个固定大小的块
  - 主存划分为大小相等的区域，称为块或内存块（物理页面，页框）
- 作业按照主存块大小分页
  - 把用户程序按逻辑页划分成大小相等的部分，称为页（page）。从0开始编制页号，页内地址是相对于0编址
- 连续的页存放在离散的块中
  - 以页为单位进行分配，并按作业的页数多少来分配。逻辑上相邻的页，物理上不一定相邻

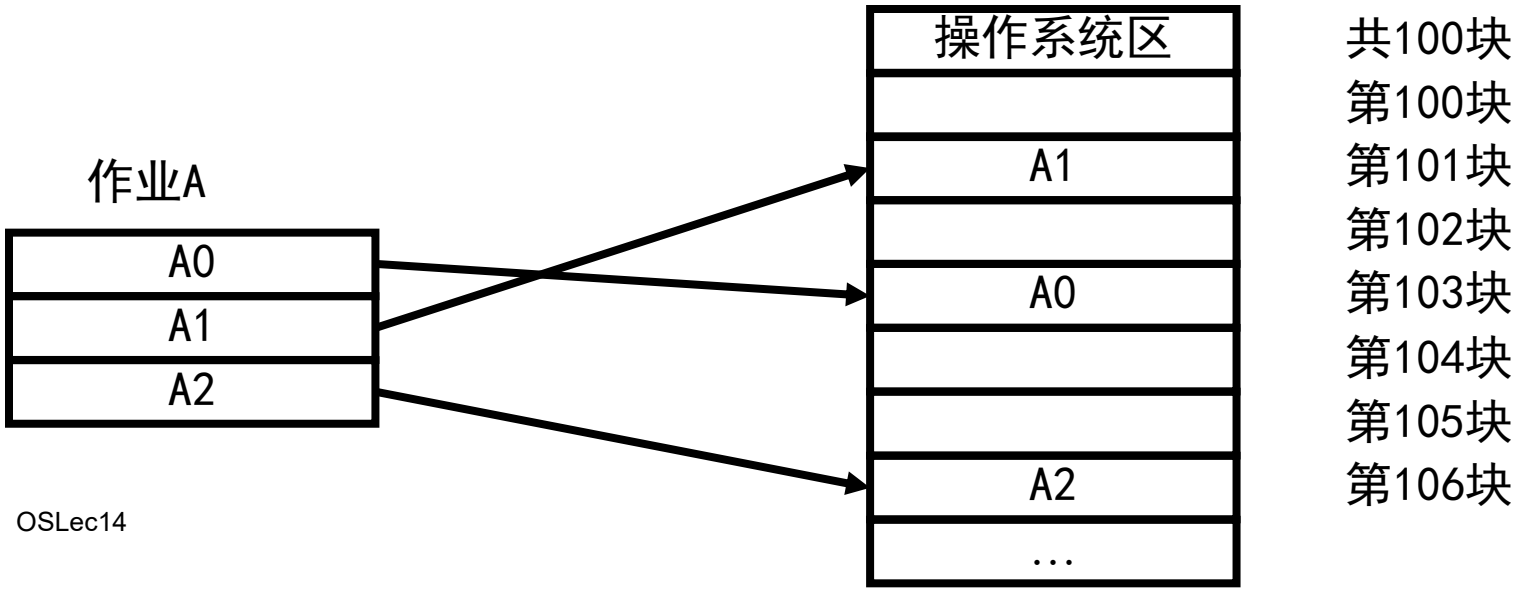
逻辑地址

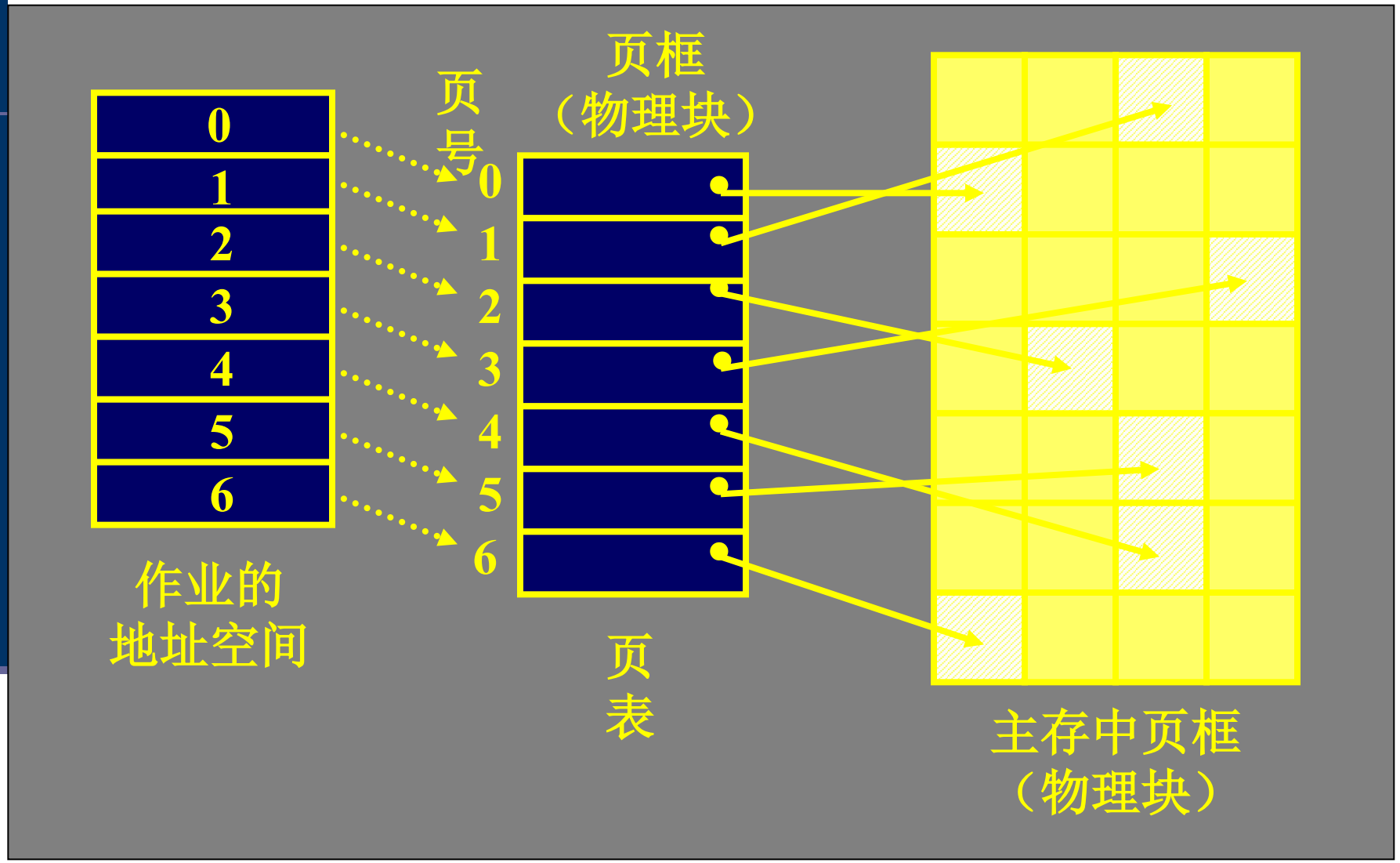
- 一般一页的大小为2的整数次幂，通常为512 B~8 KB。
- 地址的高位部分为页号，低位部分为页内地址



$$P = INT\left[\frac{A}{L}\right]$$

$$w = [A]MODL$$

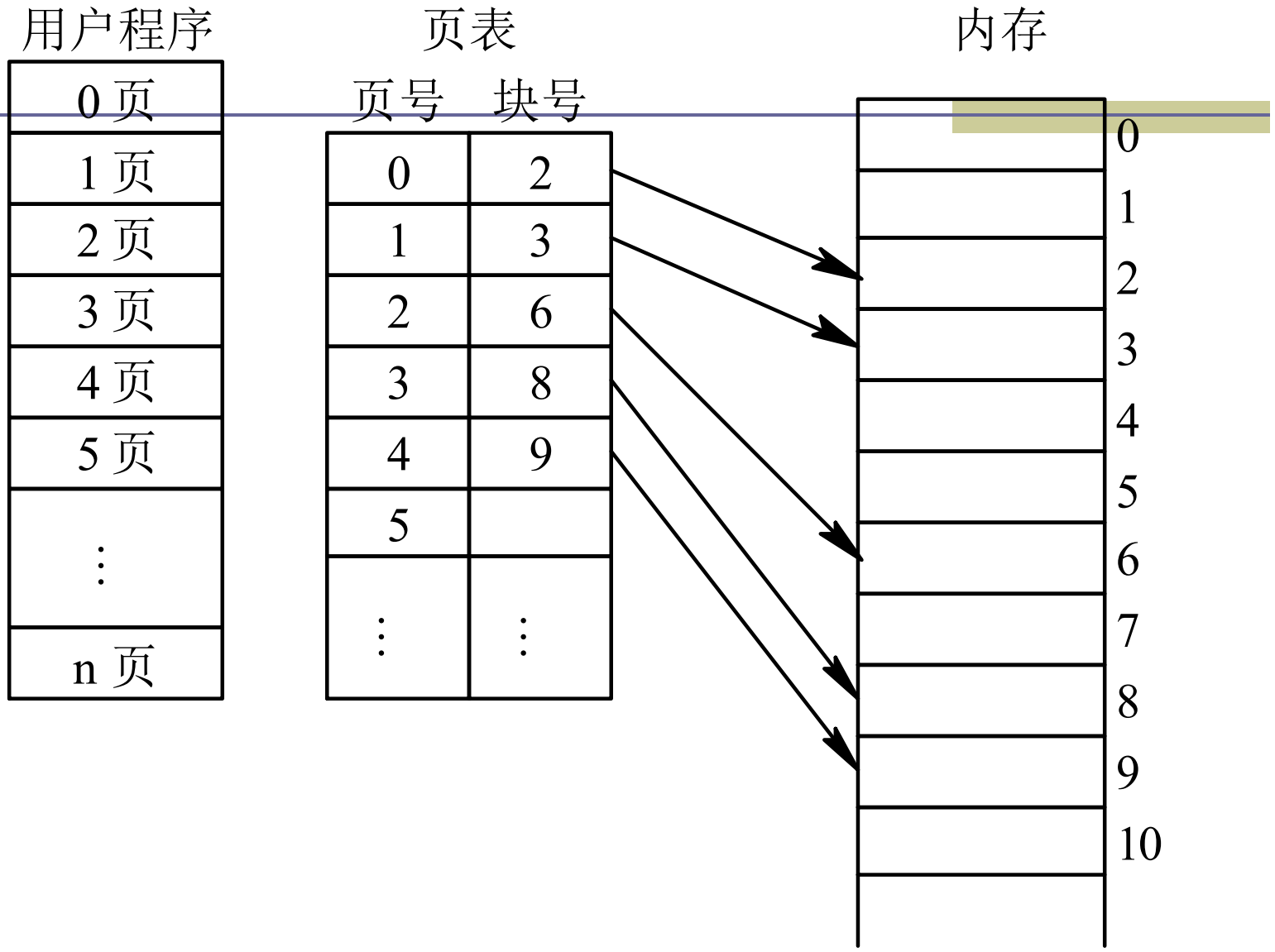



0
1
2
3
4
5
6
作业的
地址空间
页
号
0
1
2
3
4
5
6
页框
（物理块）
页
表
主存中页框
（物理块）

页表

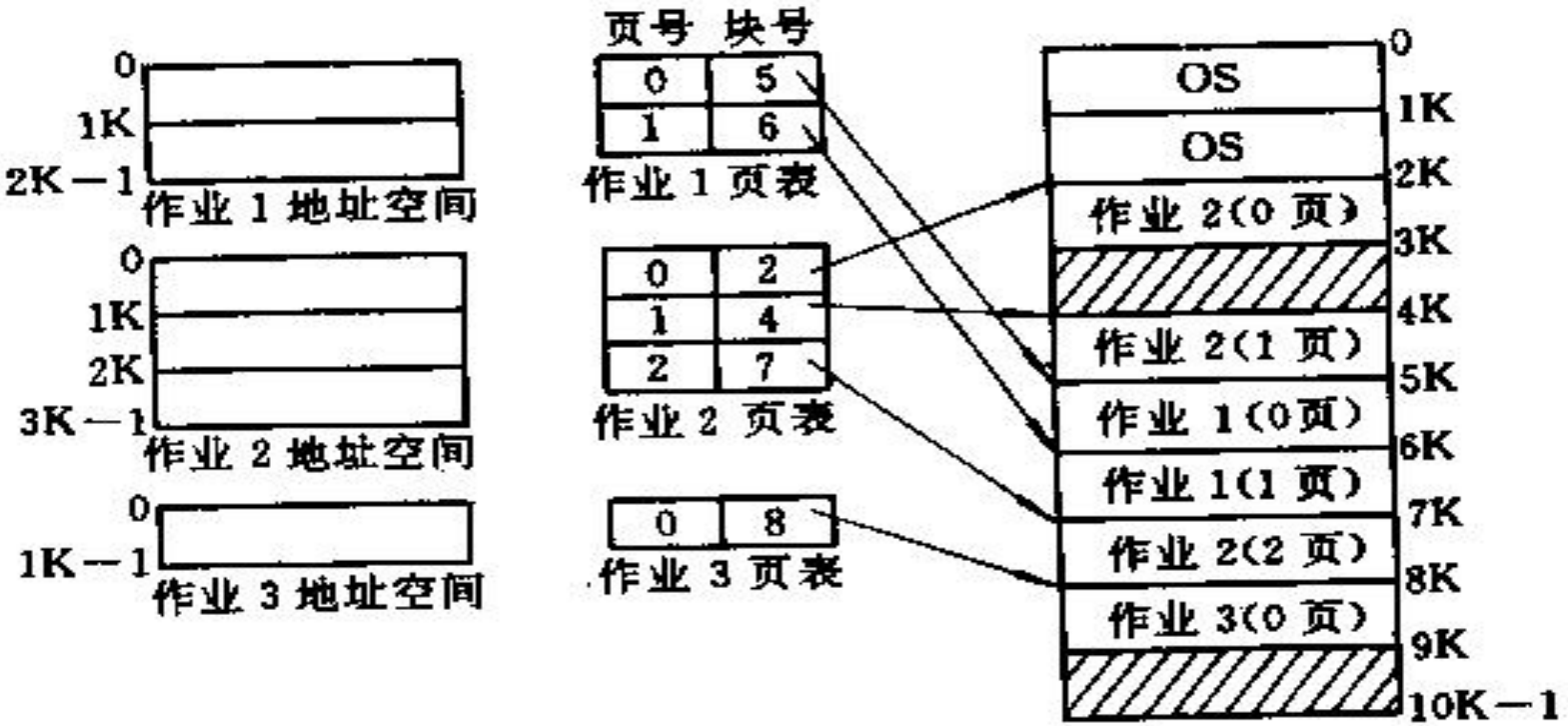
- 将页号和页内地址转换成内存地址，必须要有一个数据结构，用来登记页号和块的对应关系和有关信息。这样的数据结构称为**页表**。
- 系统为每个进程建立一个页表，页表的长度和首地址存放在该进程的进程控制块（PCB）中。
- 页表包含以下几个表项：
  - 页号：登记程序地址空间的页号
  - 块号：相应的页所对应的内存块号
  - 其它：与存储信息保护有关的信息。

页号	块号	其它
0	5	…
1	65	…
2	13	…


用户程序
0 页
1 页
2 页
3 页
4 页
5 页
n 页
页表
页号 块号
0 2
1 3
2 6
3 8
4 9
5
内存
0
1
2
3
4
5
6
7
8
9
10

例

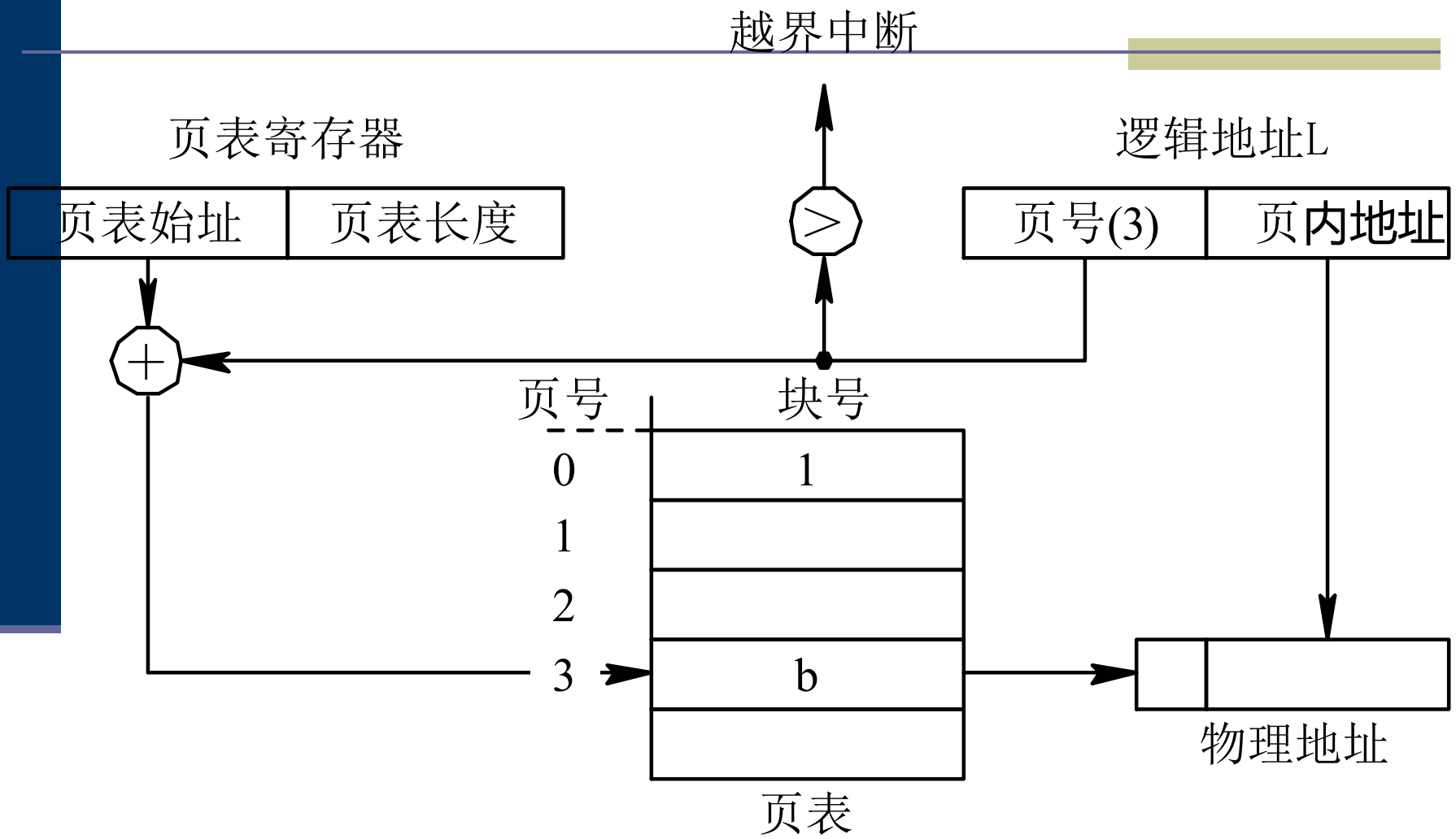
- 如图，作业1有2页分别装入内存的第5、6块；作业2有3页装入内存的第2、4、7块；作业3有1 页装入内存的第8块。



页大小的选择

- 太大：浪费；太小：页表过长。
- 页的大小是2^K ， k： 9-12。
  - IBM AS/400　VAX　NS32032　：512字节
  - Intel 80386　Motorola 68030　：4096字节

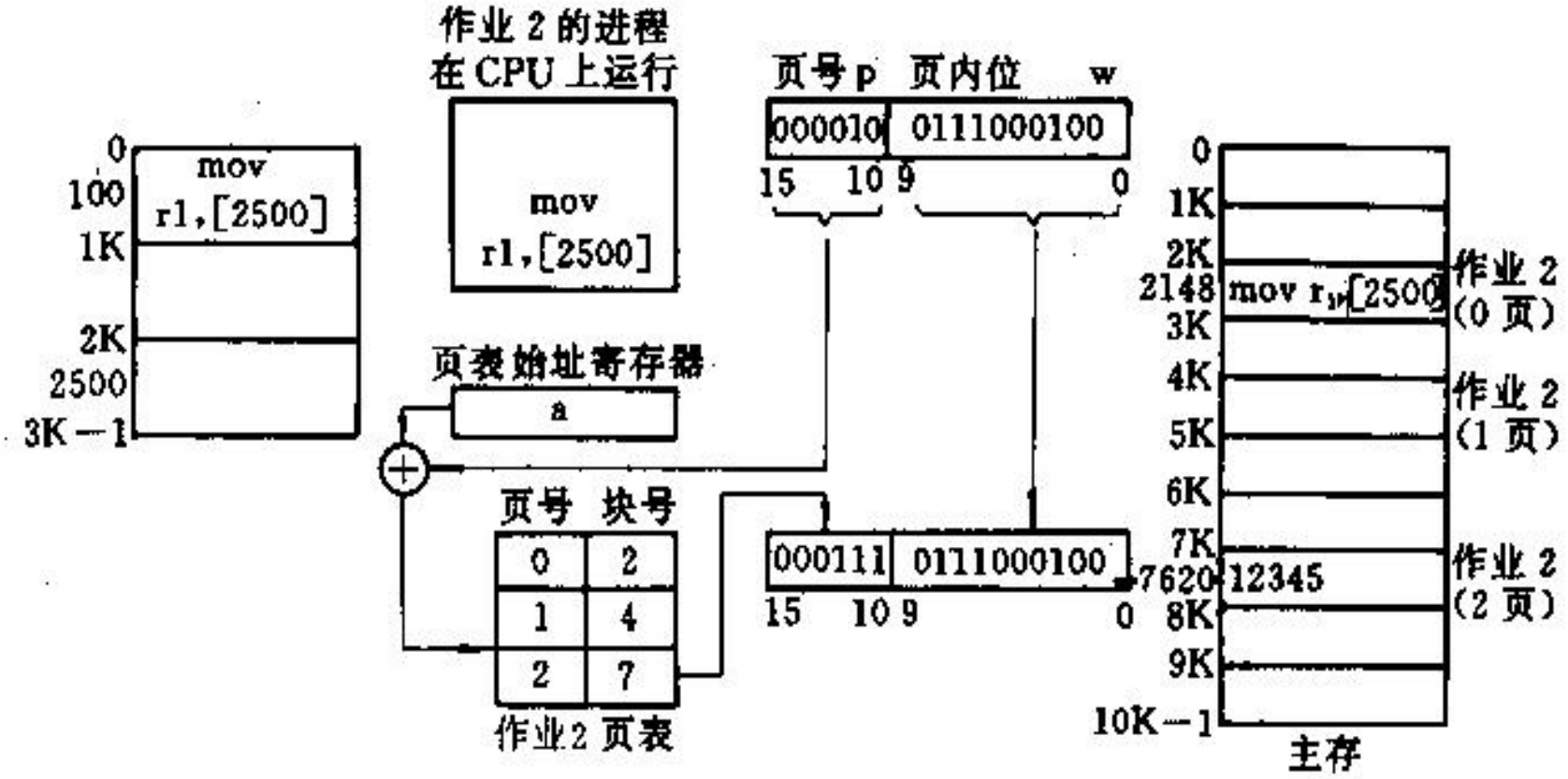
地址变换机构



分页系统的地址变换机构

例：页地址映射

- 设页长为1K，程序地址字长为16位，用户程序空间和页表如图。



计算页号和页内地址

- 若给出的地址字为16进制，则将其转换为二进制，然后，根据页长及程序地址字的长度，分别取出程序地址字的高几位和低几位就得到页号及页内地址。如页长为2K，程序地址字为16位，则高5位为页号，低11位为页内地址。
- 若给出的地址字为10进制，则用公式：

  程序地址字/页长

  商为页号，余数为页内地址。

  如程序地址为8457， 页长为4KB，则8457/4096可得：商为2，余数为256。

例：设虚地址为$(357101)_8$ 每一块为128字节

$$128 = 2^7$$

$$(357101)_8 = (011, 101, 111, 001, 000, 001)_2$$

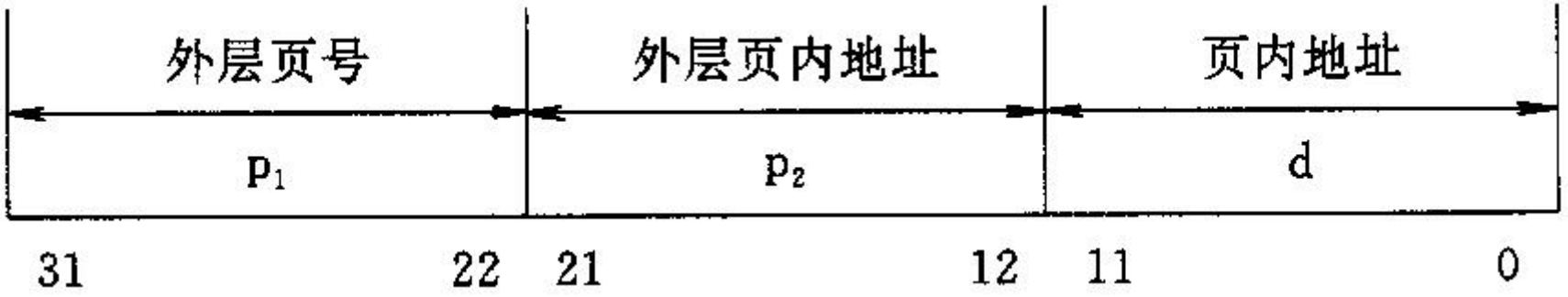
$$1 \quad 6 \quad 7 \quad 4 \quad 1 \quad 0 \quad 1$$

偏移为$(101)_8$，　　页号为$(1674)_8$

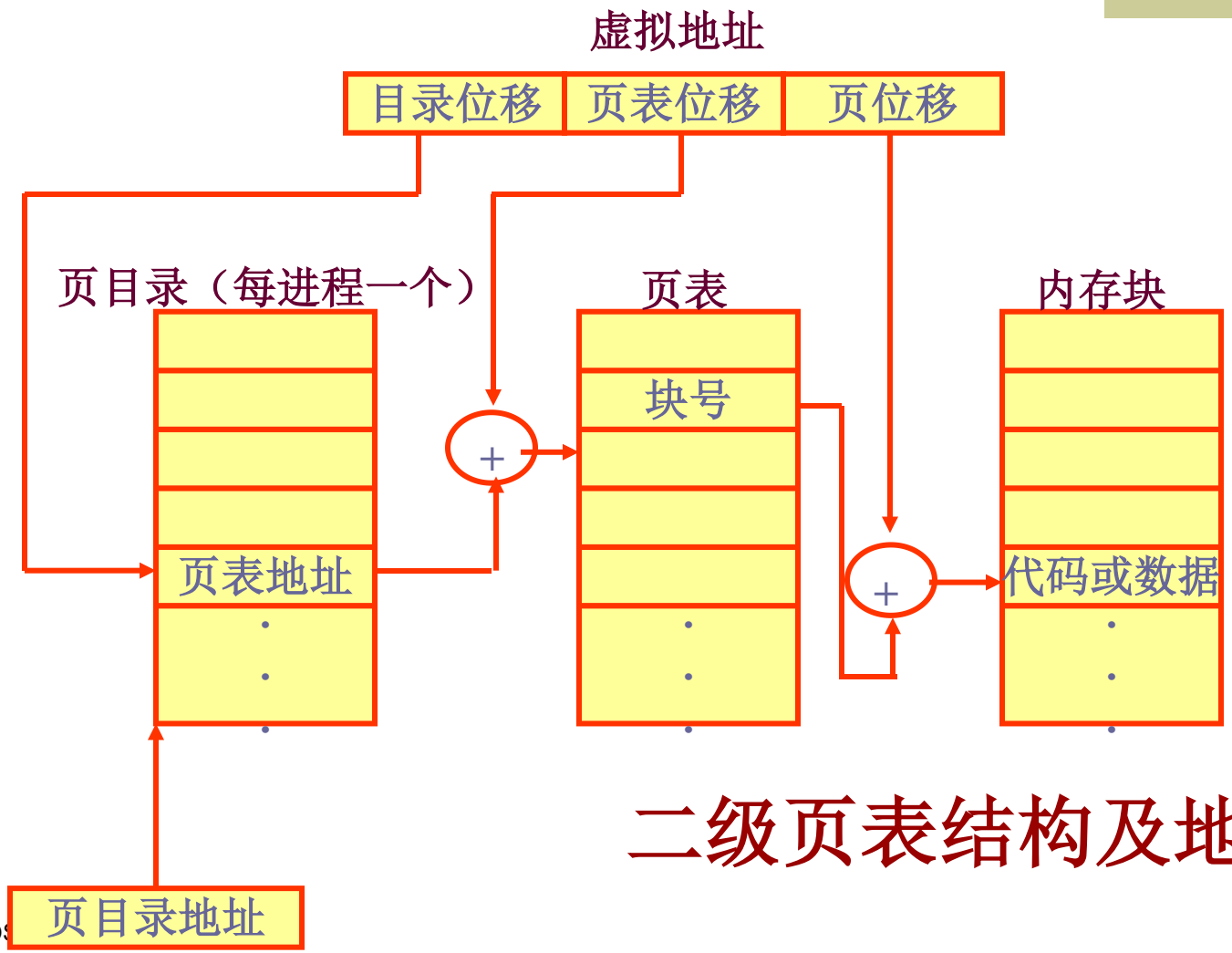
块号由页表指定，偏移不变

两级页表(Two-Level Page Table)

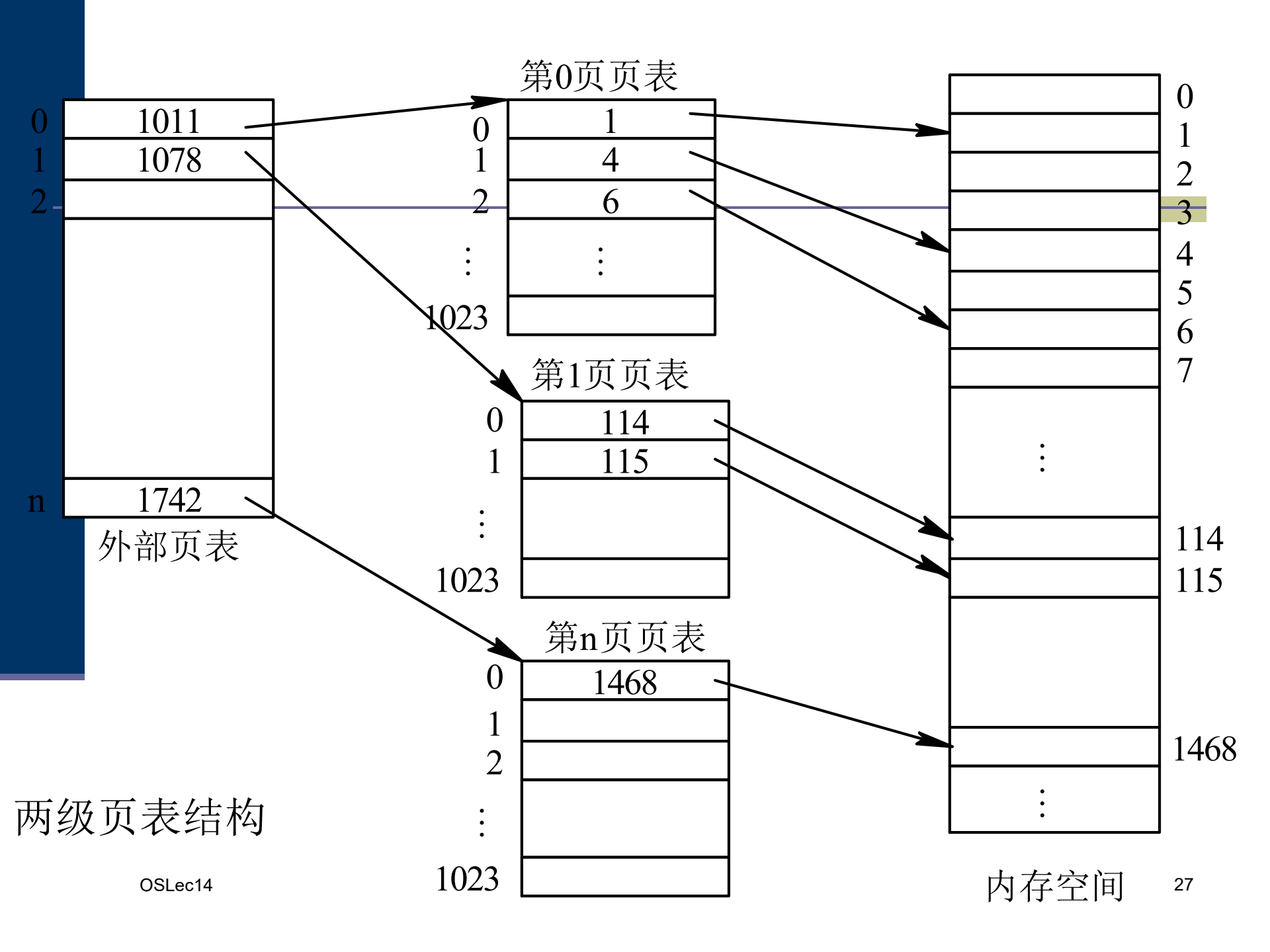
逻辑地址结构可描述如下：



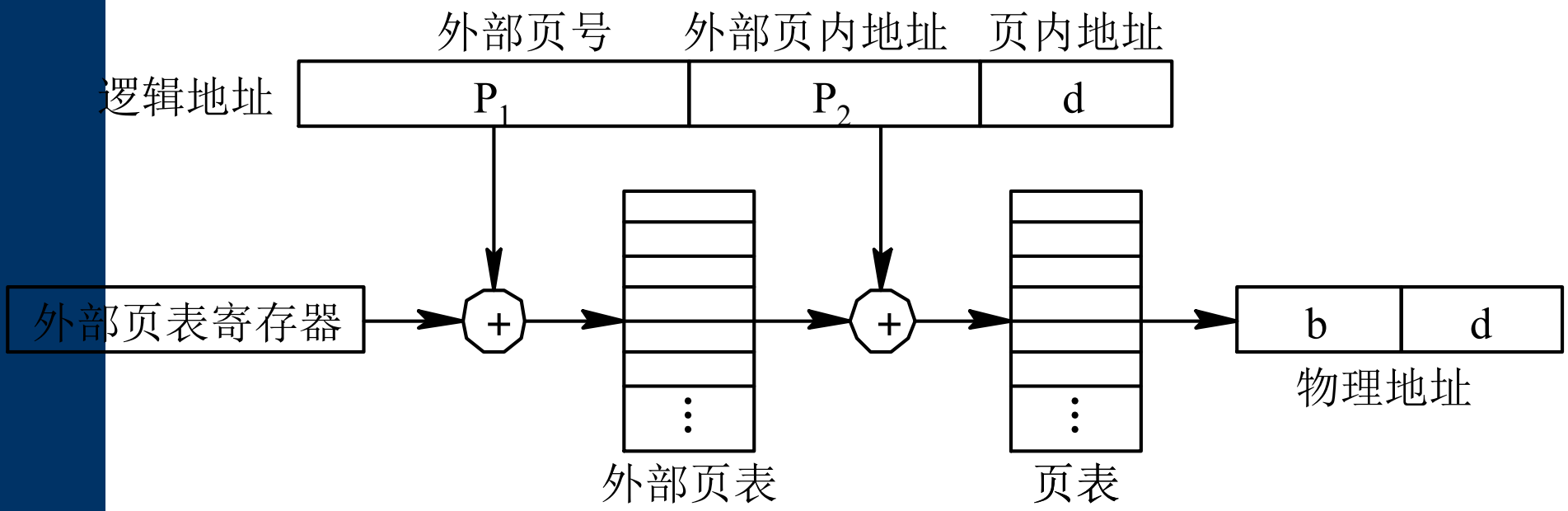
两级页表和多级页表



二级页表结构及地址映射

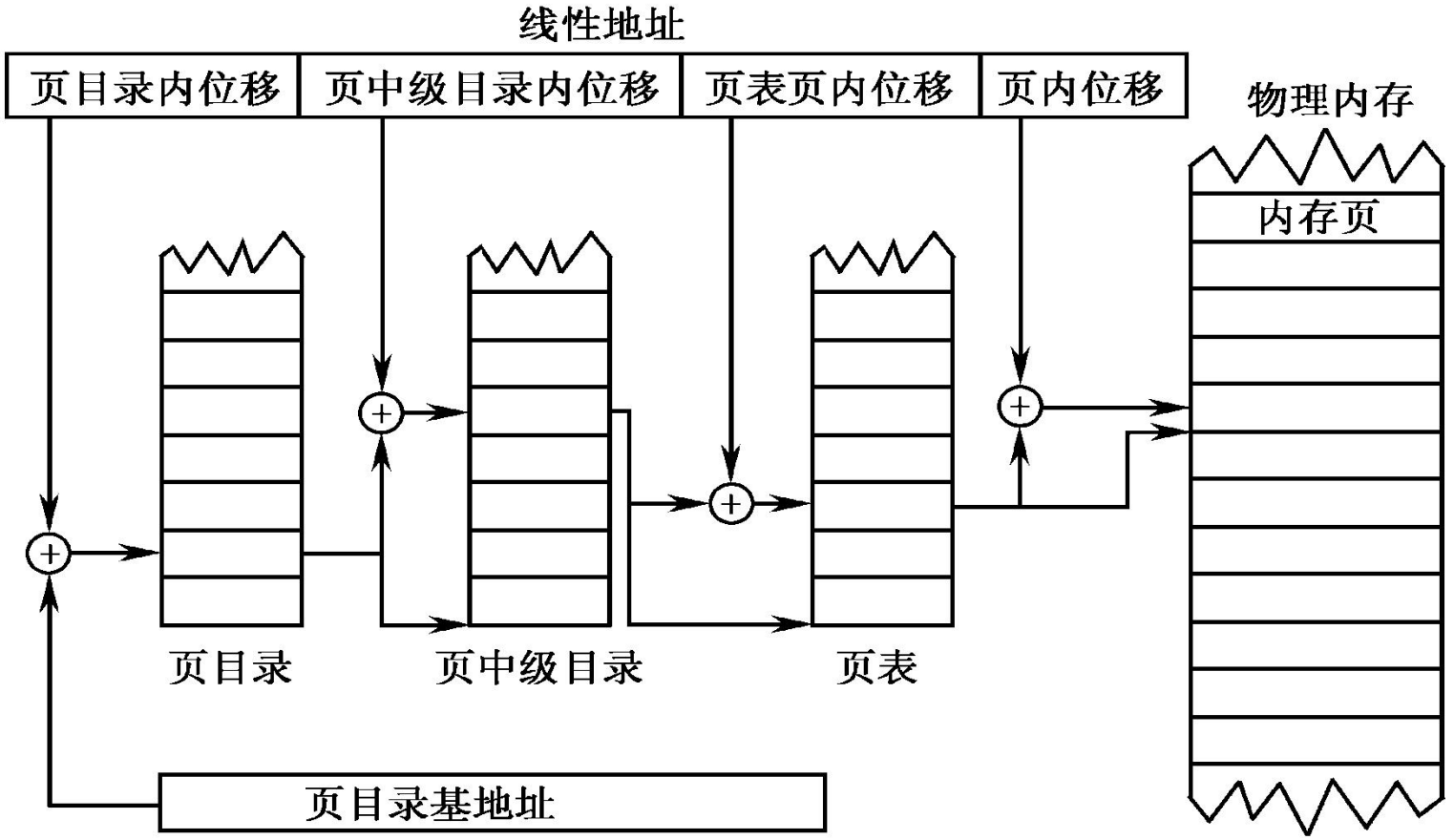

第0页页表
第1页页表
第n页页表
0
1
2
n
1011
1078
1742
外部页表
0
1
2
1023
1
4
6
0
1
1023
114
115
0
1
1023
1468
0
1
2
3
4
5
6
7
114
115
1468
内存空间

两级页表结构



具有两级页表的地址变换机构

三级页表结构及其地址映射过程



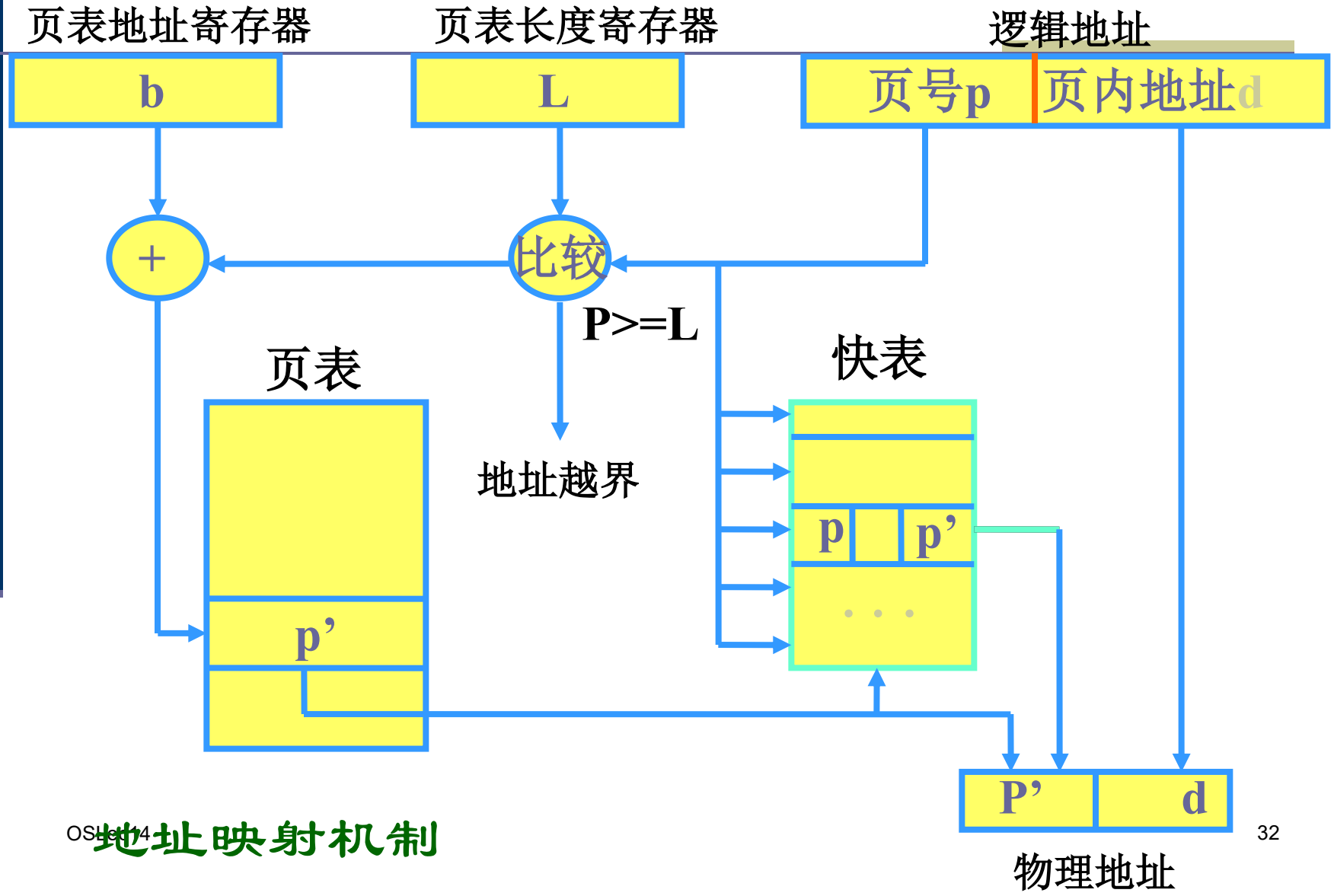
快表

- 页地址变换过程中有一个严重的问题：
  - 访问主存＝访页表＋访主存
- 解决方法：把页表放在一组快速存储器中（Cache），从而加快访问内存的速度。
- 相联存储器（associative memory）：为加快地址转换速度，系统专门设置的一个高速存储器，用于存放页表的一部分。又叫TLB（Translation lookaside buffers）
- 把这种存放在快速存储器中的页表称为快表，把存放在内存中的页表称为慢表。
- 快表快表具有并行查询能力：一次进行
  - 查联想表－物理地址(访问一次主存)
  - 查页表－物理地址(访问二次主存)

快表的大小

- 快速存储器是非常非常昂贵的。
- 并不需要一个很大的快速存储器，有一个能存放16个页表表目的快速存储器就够了。
- 硬件根据需要将页表中当前需要的少量表目读入快表，其它表目仍留在内存的页表中，当需要时读入新的表目，并淘汰适当的表目。
- 快表表项：
  - 页号；内存块号；标识位；淘汰位

具有快表的地址变换机构



地址映射机制

例：设访问主存时间为200ns,访问联想存贮器为40ns，命中率为90%，则平均存取时间为多少？

解：

方法1：

查页表两次访存：平均为200＋200＝400ns

方法2：

查块表(200+40)×90%＋(200+200)×10%＝256ns

分页存储管理

- 应维护的表项：
  - 作业表（JT）：整个系统一张表，每个作业在表中对应一个表目。
  - 存储分块表（MBT）：整个系统一张表，表中每一个表目对应一个存储块，记录了该块的状态：已分配或空闲。
  - 页面变换表（PMT）：每个作业一张表，用于该作业的地址变换，该作业有多少页面就有多少表目。
- 操作系统实施管理。

页式存储管理方案小结

- 优点：解决了碎片问题
  便于管理

- 缺点：增加成本，多占用主存空间
  多花费处理机时间

What you need to do?

- 复习课本4.3节的内容
- 课后作业：习题2、4、8、10

See you next time!