

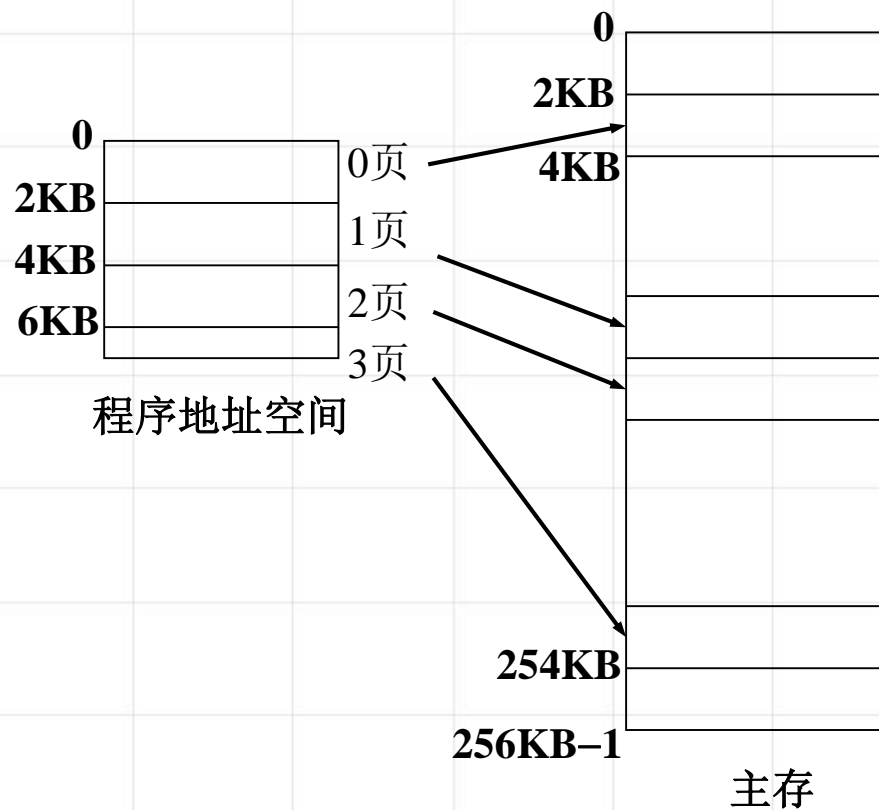


# 页式存储管理

## 1. 页式系统的基本概念

### (1) 页面

程序的地址空间被等分成大小相等的片，称为页面，又称为虚页。



### (2) 主存块

主存被等分成大小相等的片，称为主存块，又称为实页。

### (3) 页面与主存块的关系

等分主存和虚地址空间

## (4) 页表

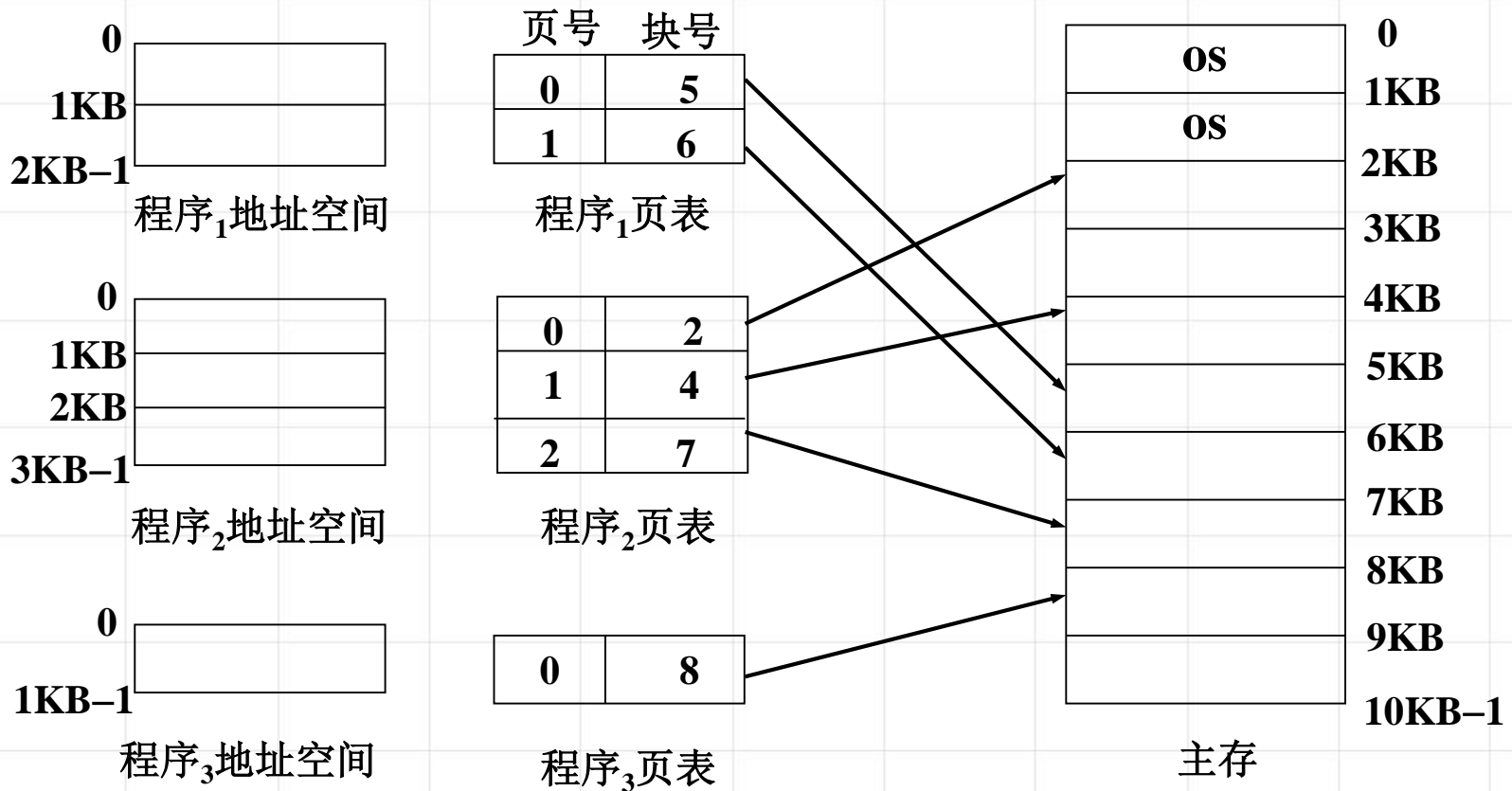
### ① 什么是页表

为了实现从地址空间到物理主存的映象，系统建立的记录页与内存块之间对应关系的地址变换的机构称为页面映像表，简称页表。

### ② 页表的组成

- i 高速缓冲存储器：地址变换速度快，但成本较高
- ii 主存区域：地址变换速度比硬件慢，成本较低

## (5) 分页映像存储的例



分页映像存储

## 2. 页式地址变换

### (1) 页表

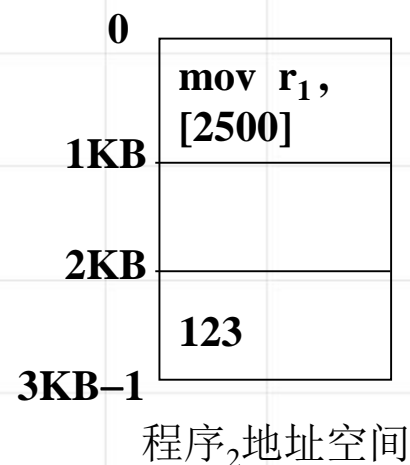
记录页与块之间对应关系的地址变换的机构。

### (2) 虚地址结构

当CPU给出的虚地址长度为16位，页面大小为1KB时，在分页系统中地址结构的格式如下：

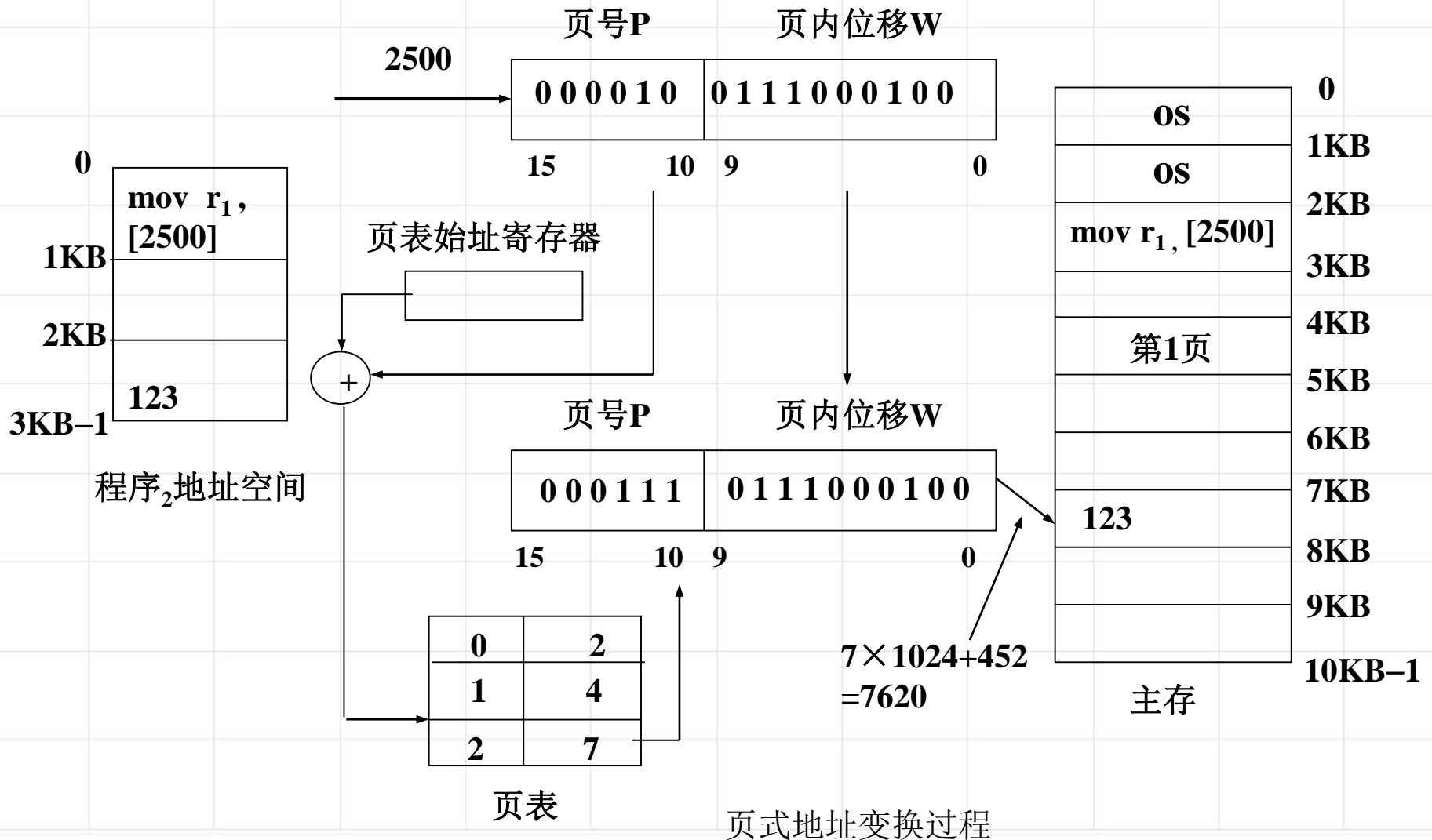


虚地址结构





## ② 页式地址变换过程



## ③ 页式地址变换步骤

- i CPU给出操作数地址 (为2500) ;
- ii 由分页机构自动地把逻辑地址分为两部分, 得到页号 $p$ 和页内相对位移 $w$  ( $p = 2, w = 452$ );
- iii 根据页表始址寄存器指示的页表始地址, 以页号为索引, 找到第2页所对应的块号 (为7) ;
- iv 将块号 $b$ 和页内位移量 $w$ 拼接在一起, 就形成了访问主存的物理地址 ( $7 \times 1024 + 452 = 7620$ )



## (4) 采用联想存储器加快查表速度

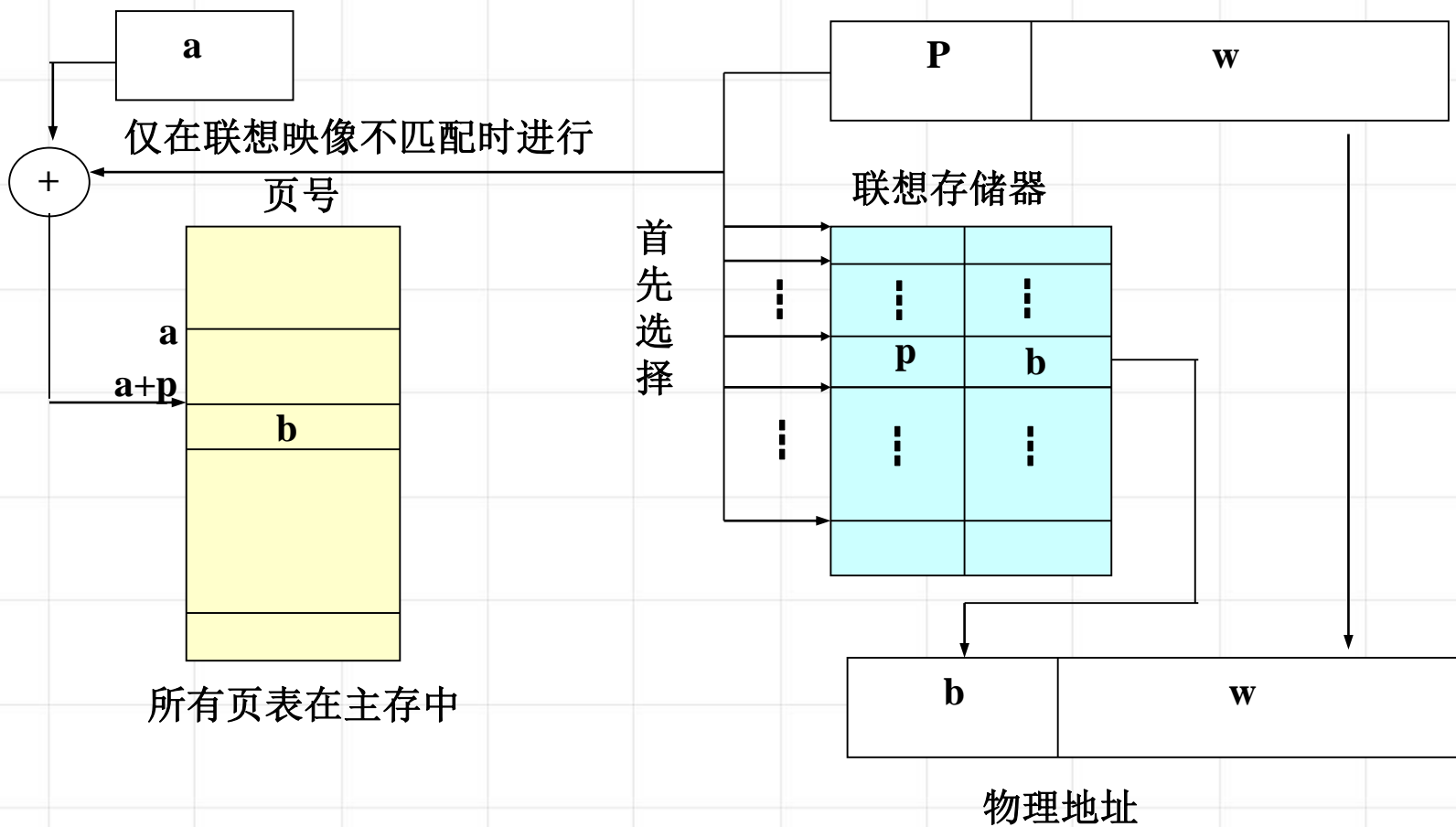
### ① 什么是联想存储器

高速、小容量半导体存储部件，又称缓冲存储器。

### ② 快表

在缓冲存储器中存放正在运行的进程当前用到的页号和对应的块号，又称为快表。

## ③ 利用快表进行地址映射



联想寄存器和主存页表相结合的分页地址变换

### 3. 请调页面的机制

#### (1) 两种页式系统

- ① 简单页式系统：装入一个程序的全部页面才能投入运行。
- ② 请求页式系统：装入一个程序的部分页面即可投入运行。

请求页式系统需解决的问题

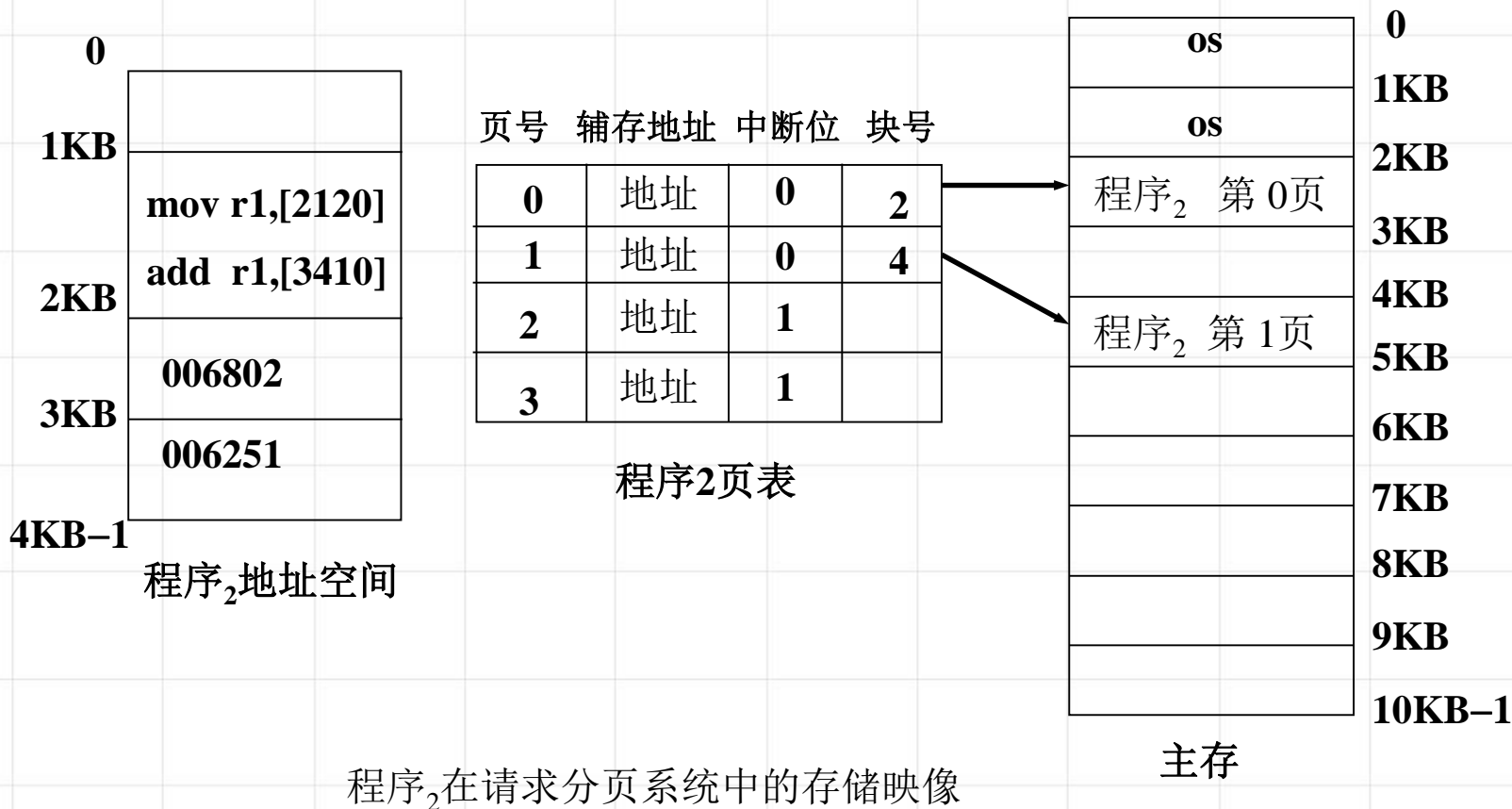
#### (2) 扩充页表功能

| 页号 | 主存块号 | 中断位 | 辅存地址 |
|----|------|-----|------|
|----|------|-----|------|

- 中断位 $i$ ：标识该页是否在主存  
若 $i=1$ ，表示此页不在主存；若 $i=0$ ，表示该页在主存
- 辅存地址：该页面在辅存的位置

## (3) 缺页处理

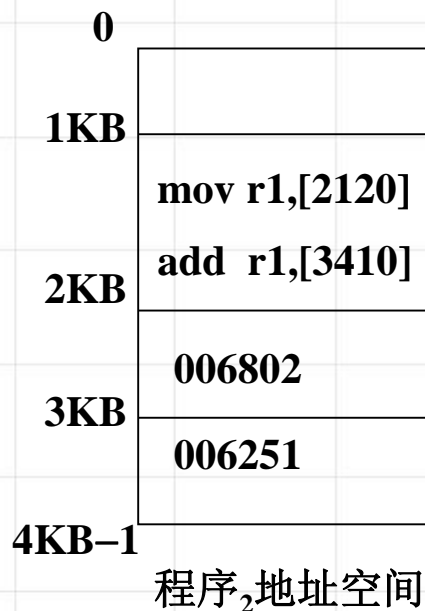
### ① 程序<sub>2</sub>在请求分页系统中的存储映像



## ② 缺页处理的例

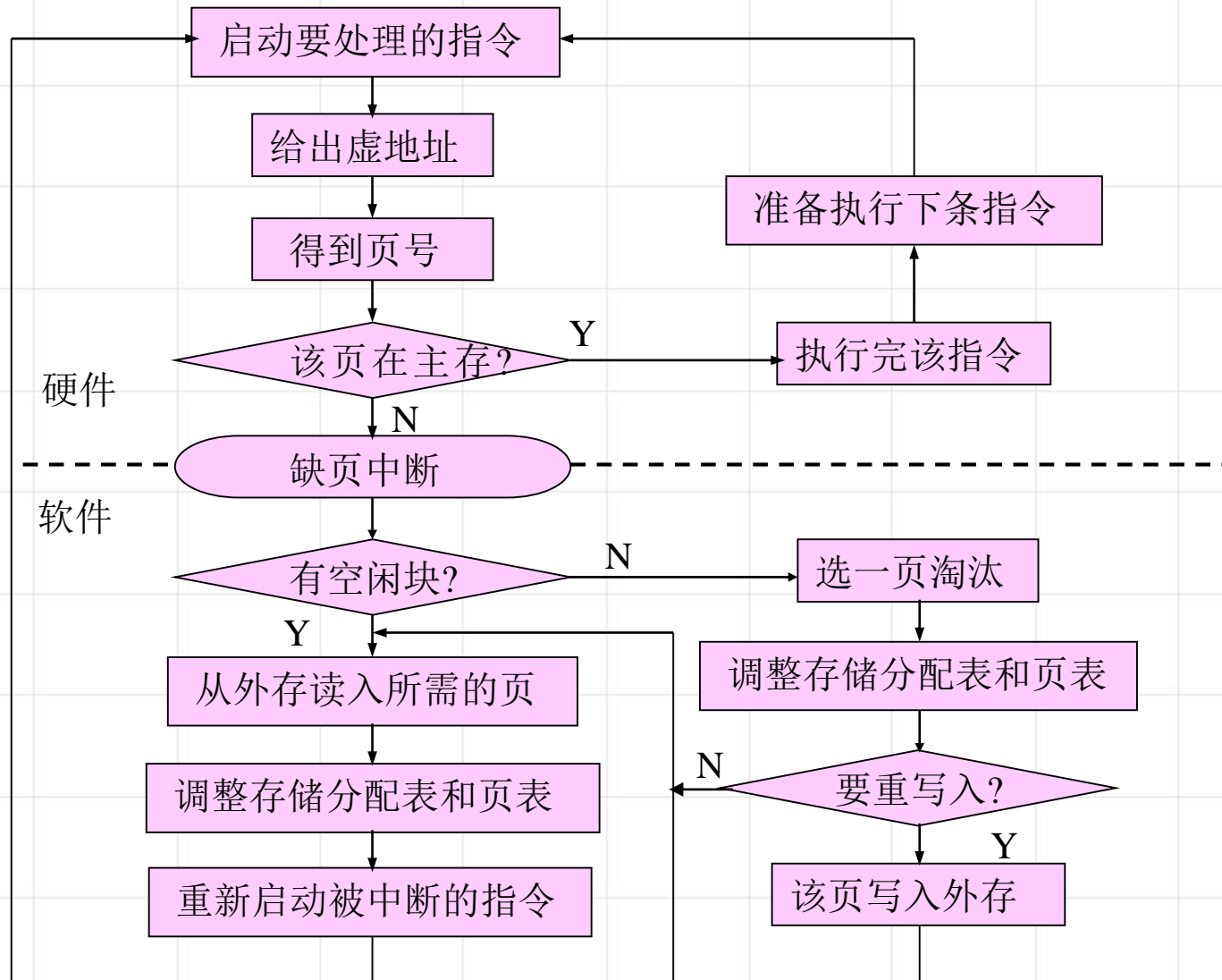
程序2的主存块数为  $m_2=3$ ，讨论程序执行

“`mov r1, [2120]`”指令时的情况。



- CPU产生的虚地址为2120
- 分页机构得  $p=2$ ,  $w=72$
- 查页表。该页中断位 $i=1$
- **发生缺页中断！**
- 如主存中有空白块，且 $n < m$  则直接调入
- 如主存中无空白块，或 $n \geq m$ ，则需淘汰该程序在主存中的一页

## ③ 缺页处理过程图示



指令执行步骤和缺页中断处理过程

## 4. 淘汰机制与策略

### (1) 什么是淘汰策略

用来选择淘汰哪一页的规则叫做置换策略，或称淘汰算法。

如何决定淘汰哪一页？

### (2) 扩充页表功能

| 页号 | 主存块号 | 中断位 | 辅存地址 | 引用位 | 改变位 |
|----|------|-----|------|-----|-----|
|----|------|-----|------|-----|-----|

① 引用位 —— 标识该页最近是否被访问

为“0”——该页没有被访问；为“1”——该页已被访问

② 改变位 —— 表示该页是否被修改

为“0”——该页未被修改；为“1”——该页已被修改

## (3) 颠簸

颠簸(thrashing), 又称为“抖动”。

简单地说, 导致系统效率急剧下降的主存和辅存之间的频繁页面置换现象称为“抖动”。

## (4) 缺页中断率

- 假定程序p共有n页, 系统分配m块, 有  $1 \leq m \leq n$ ;
- 若程序p在运行中: 成功的访问次数为s, 不成功的访问次数为f;
- 缺页中断率:  $f' = f / (s + f)$

$$f' = f(r, m, p);$$

r: 置换算法; m: 系统分配的块数; p: 程序特征



## (5) 常用的置换算法

### ① 最佳算法(OPT算法)

当要调入一新页而必须先淘汰一旧页时，所淘汰的那一页应是以后不再要用的，或者是在最长的时间以后才会用到的那页。

## ② 先进先出淘汰算法(FIFO算法)

### i 什么是先进先出淘汰算法

总是选择在主存中居留时间最长 (即最早进入主存) 的一页淘汰。

### ii 先进先出淘汰算法的实现

- 建立一个页面进入主存的先后次序表;
- 建立一个替换指针, 指向最早进入主存的页面;
- 当需要置换一页时, 选择替换指向的那一页, 然后调整替换指针的内容。

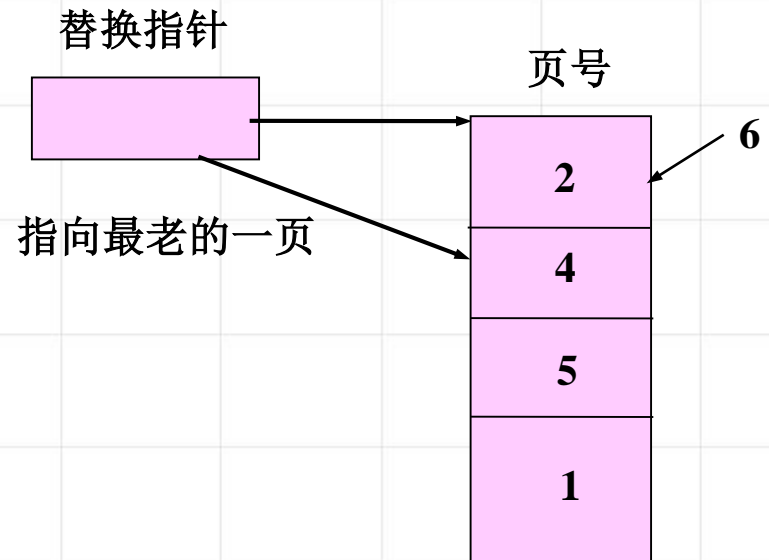
## iii 页号表

页号表记录页面进入  
主存的先后次序：

2→4→5→1

当要调入第6页时：

- 置换第2页
- 将第2页改为6
- 替换指针指向第4页



先进先出淘汰算法图例

iv 在存储分块表中建立次序表  
在存储分块表中记录页面进入主存的先后次序：

4→5→1→2

当要调入第6页时：

如何处理？

5→1→2 → 6

块号 页号 指针

|   |   |     |
|---|---|-----|
| 0 |   |     |
| 1 |   |     |
| 2 | 4 | 6   |
| 3 |   |     |
| 4 | 2 | → ^ |
| 5 |   |     |
| 6 | 5 | 7 ← |
| 7 | 1 | 4 ← |

替换指针

2

块号 页号 指针

|   |   |     |
|---|---|-----|
| 0 |   |     |
| 1 |   |     |
| 2 | 6 | → ^ |
| 3 |   |     |
| 4 | 2 | → 2 |
| 5 |   |     |
| 6 | 5 | 7 ← |
| 7 | 1 | 4 ← |

替换指针

6

先进先出淘汰算法存储块构造

## ③ 最久未使用淘汰算法(LRU算法)

### i 什么是最久未使用淘汰算法

总是选择最长时间未被使用的那一页淘汰。

### ii 最久未使用淘汰算法的实现

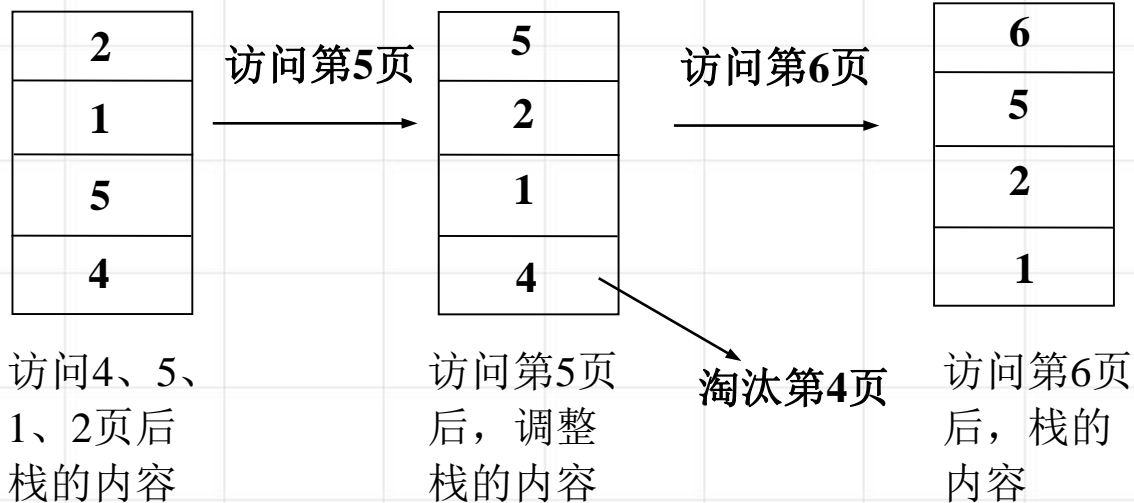
- 用引用位考察页面的使用情况；
- 当访问页面时，将引用位置1，并计时；
- 当要淘汰一页时，选择时间最长的一页淘汰。

要精确实现很困难

- 硬件方法：采用计数器
- 软件方法：采用页号栈

## iii 软件方法：采用页号栈

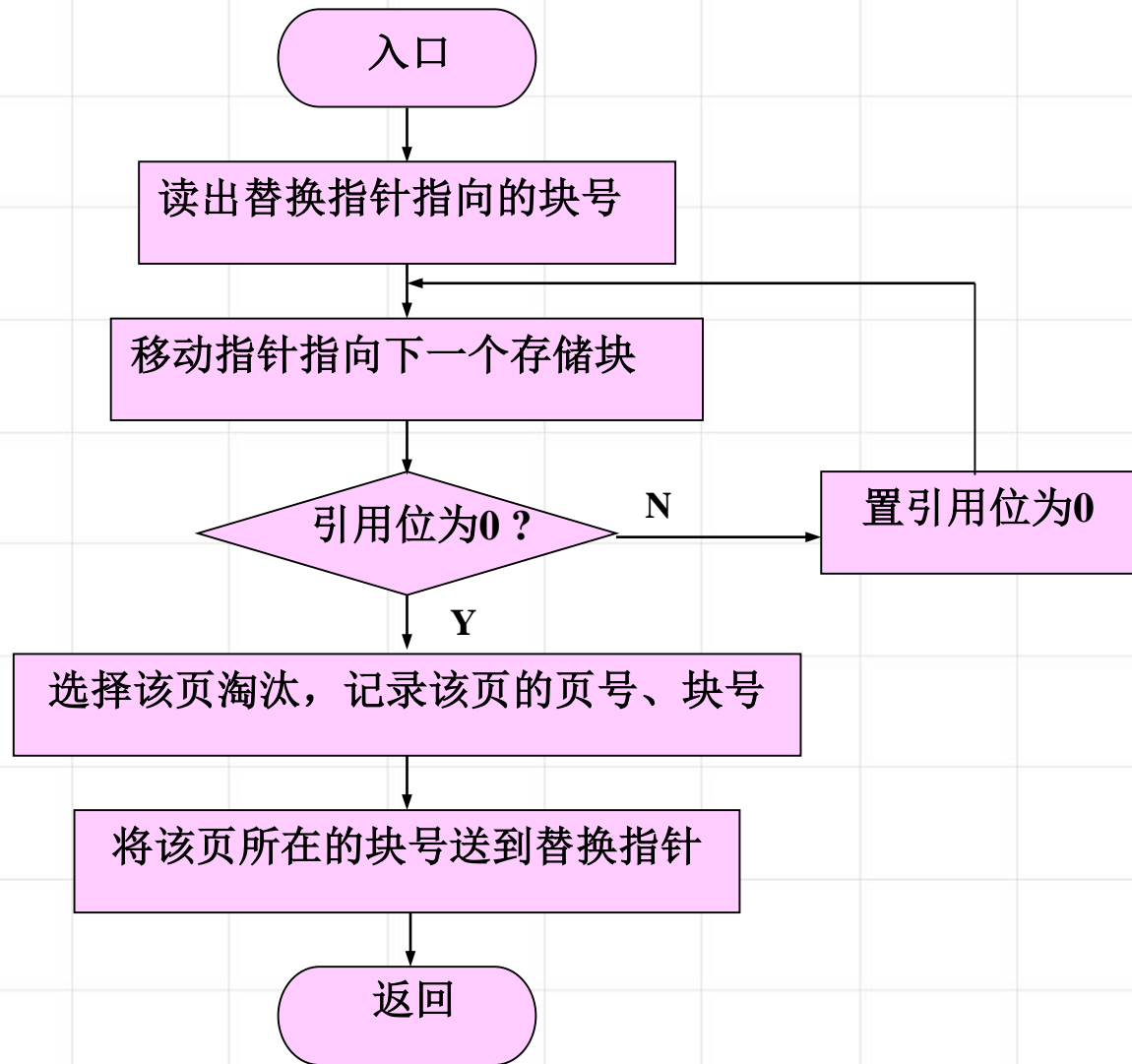
页面访问轨迹：4→5→1→2→5→6



用页号栈记录最近访问的页

## ④ LRU近似淘汰算法

### i 流程图



LRU近似算法流程图

## ii LRU近似淘汰算法举例

块号 页号 引用位 指针

|   |   |   |     |
|---|---|---|-----|
| 0 |   |   |     |
| 1 |   |   |     |
| 2 | 4 | 0 | → 4 |
| 3 |   |   |     |
| 4 | 2 | 1 | ← 6 |
| 5 |   |   |     |
| 6 | 5 | 1 | ← 7 |
| 7 | 1 | 0 | ← 2 |

替换指针

4

块号 页号 引用位 指针

|   |   |   |     |
|---|---|---|-----|
| 0 |   |   |     |
| 1 |   |   |     |
| 2 | 4 | 0 | → 4 |
| 3 |   |   |     |
| 4 | 2 | 1 | ← 6 |
| 5 |   |   |     |
| 6 | 5 | 0 | ← 7 |
| 7 | 6 | 1 | ← 2 |

替换指针

7

LRU近似算法举例

当要调入第6页时，如何处理？





# 段页式存储管理

## 1. 段式地址空间

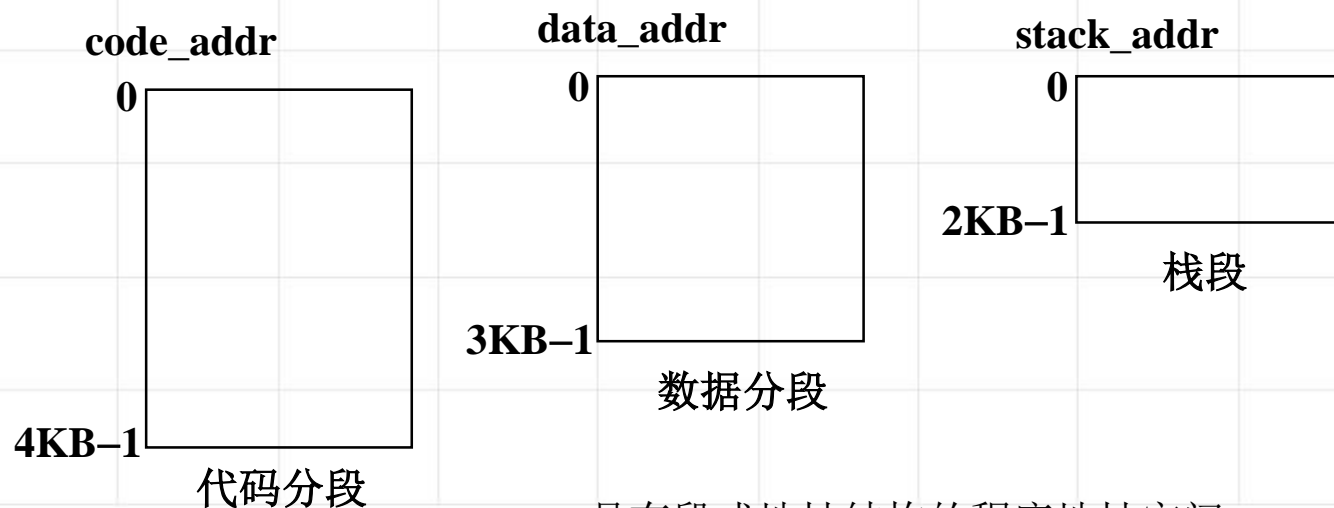
### (1) 什么是段

分段是程序中自然划分的一组逻辑意义完整的信息集合。

分段的例：代码分段、数据分段、栈段页。

### (2) 程序地址空间

由若干个逻辑分段组成，每个分段有自己的名字，对于一个分段而言，它是一个连续的地址区。



具有段式地址结构的程序地址空间

## (3) 段式地址结构



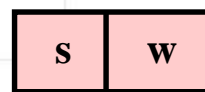
段式地址结构

## 2. 段式地址变换

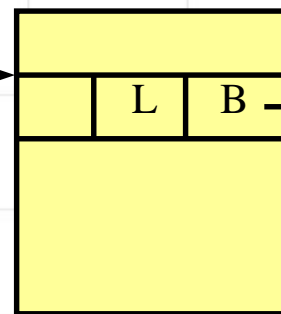
### 段式地址步骤

- 取出程序地址( $s, w$ );
- 用 $s$ 检索段表;
- 如 $w < 0$ 或 $w \geq L$ 则主存越界;
- $(B + w)$ 即为所需主存地址

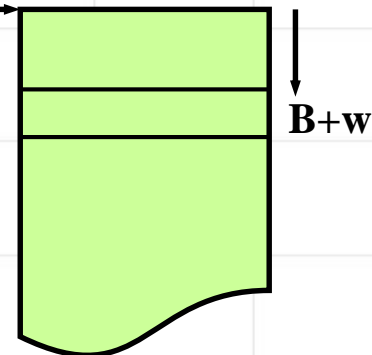
段号 段内位移



段号 长度 基址



第 $S$ 段



段式地址变换构

### 3. 页式系统与段式系统的区别

#### (1) 用户地址空间的区别

- ① 页式系统中用户地址空间： 一维地址空间
- ② 段式系统中用户地址空间： 二维地址空间

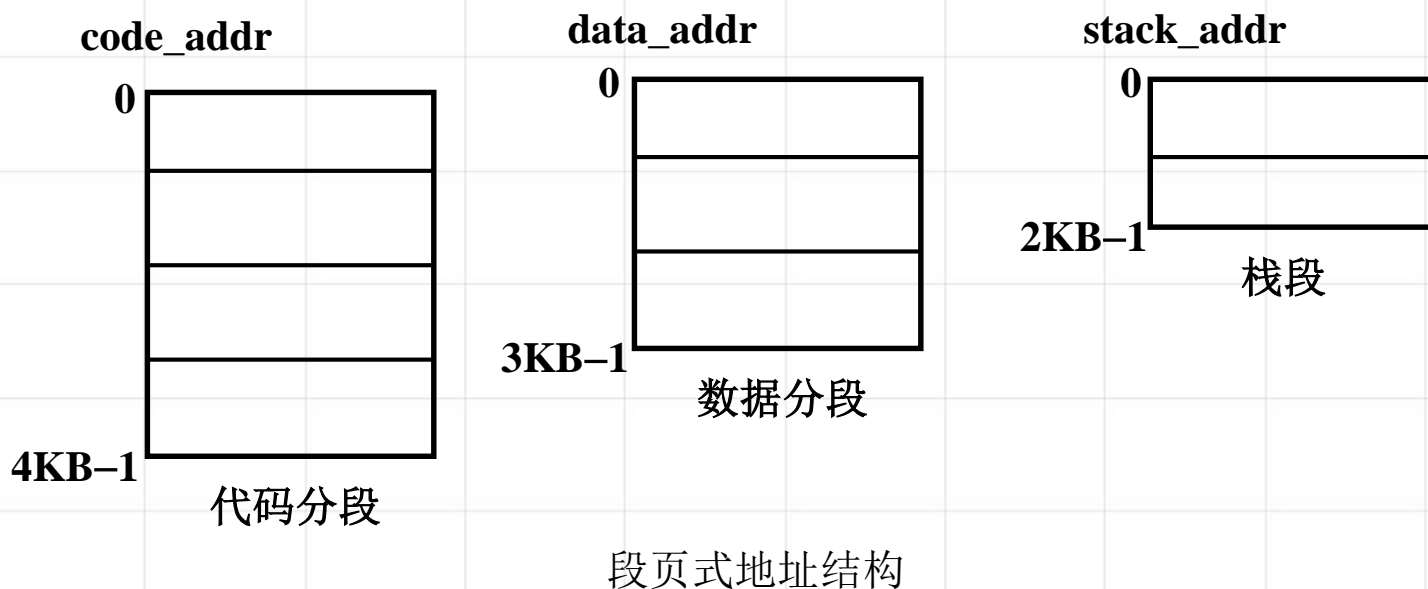
#### (2) 分段和页面的区别

| 分段            | 页面             |
|---------------|----------------|
| 信息的逻辑划分       | 信息的物理划分        |
| 段长是可变的        | 页的大小是固定的       |
| 用户可见          | 用户不可见          |
| w字段的溢出将产生越界中断 | w字段的溢出自动加入到页号中 |

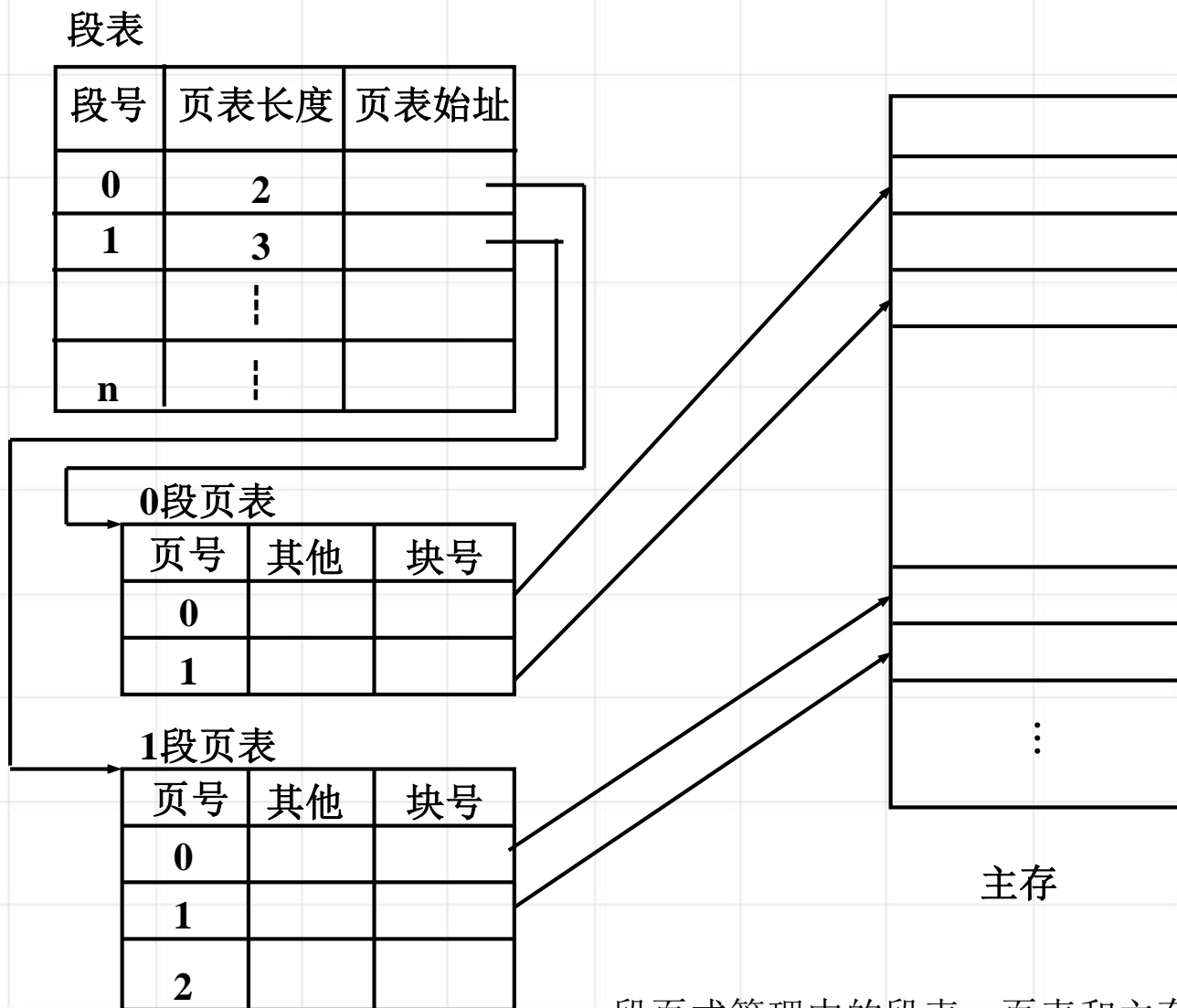
## 4. 段页式系统

在段式存储管理中结合分页存储管理技术，在一个分段内划分页面，就形成了段页式存储管理。

### (1) 段页式地址结构的程序地址空间



## (2) 段页式系统中段表、页表与主存的关系



段页式管理中的段表、页表和主存的关系

# 第6章 主存管理 小结

- 基本概念
  - 虚、实分离
    - 逻辑地址 作业地址空间
    - 物理地址 存储空间
  - 地址映射
    - 定义
    - 类型——静态地址重定位    定义    实现
    - 动态地址重定位    定义    实现
  - 虚存
  - 存储保护
    - 定义
    - 界地址保护的实现方法



- 分区存储管理
  - 分区分配中的数据结构
    - 空闲区队列结构
  - 放置策略
    - 首次适应算法、最佳适应算法、最坏适应算法
    - 的定义、特点
    - 三种放置策略的讨论
  - 分区的缺点及解决
    - 碎片与拼接

## ● 页式存储管理

### ● 页式地址变换

- 页表 虚地址结构
- 页式地址变换过程

### ● 请调策略

- 扩充页表功能 —— 中断位 辅存地址
- 缺页处理

### ● 淘汰策略

- 扩充页表功能 —— 引用位 改变位
- 抖动
- 置换算法
  - 定义
  - 先进先出淘汰算法、最久未使用淘汰算法

- 段页式存储管理
  - 段式系统的二维地址结构
  - 段页式系统中段表、页表与主存的关系