



张涛

Review

虚拟存储器

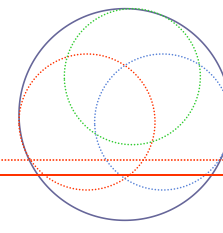
请求分页存储管理

请求分页原理

页面置换算法

性能分析

方案评价



Today we focus on...

分段存储管理

分段存储管理基本思想

段地址映射

分段的共享与保护

分段管理方案评价

4.7.1 分段存储管理基本思想

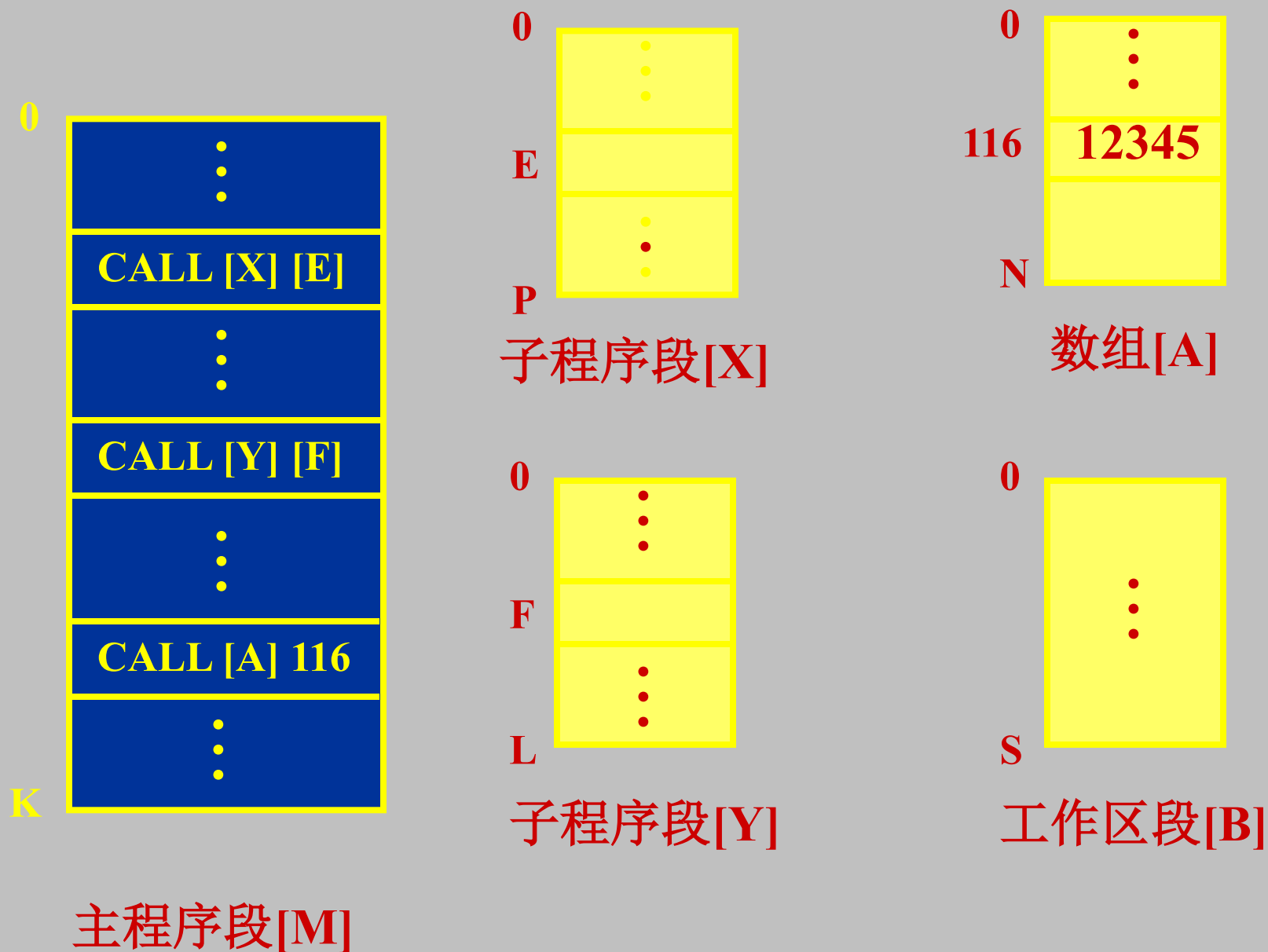
■ 用户程序划分

按程序自身的逻辑关系划分为若干个程序段，每个程序段都有一个段名，且有一个**段号**。段号从0开始，每一段段内也从0开始编址，段内地址是连续的

■ 逻辑地址：分段地址系统中的地址结构形式：



如下图，可见整个作业的地址空间是二维的

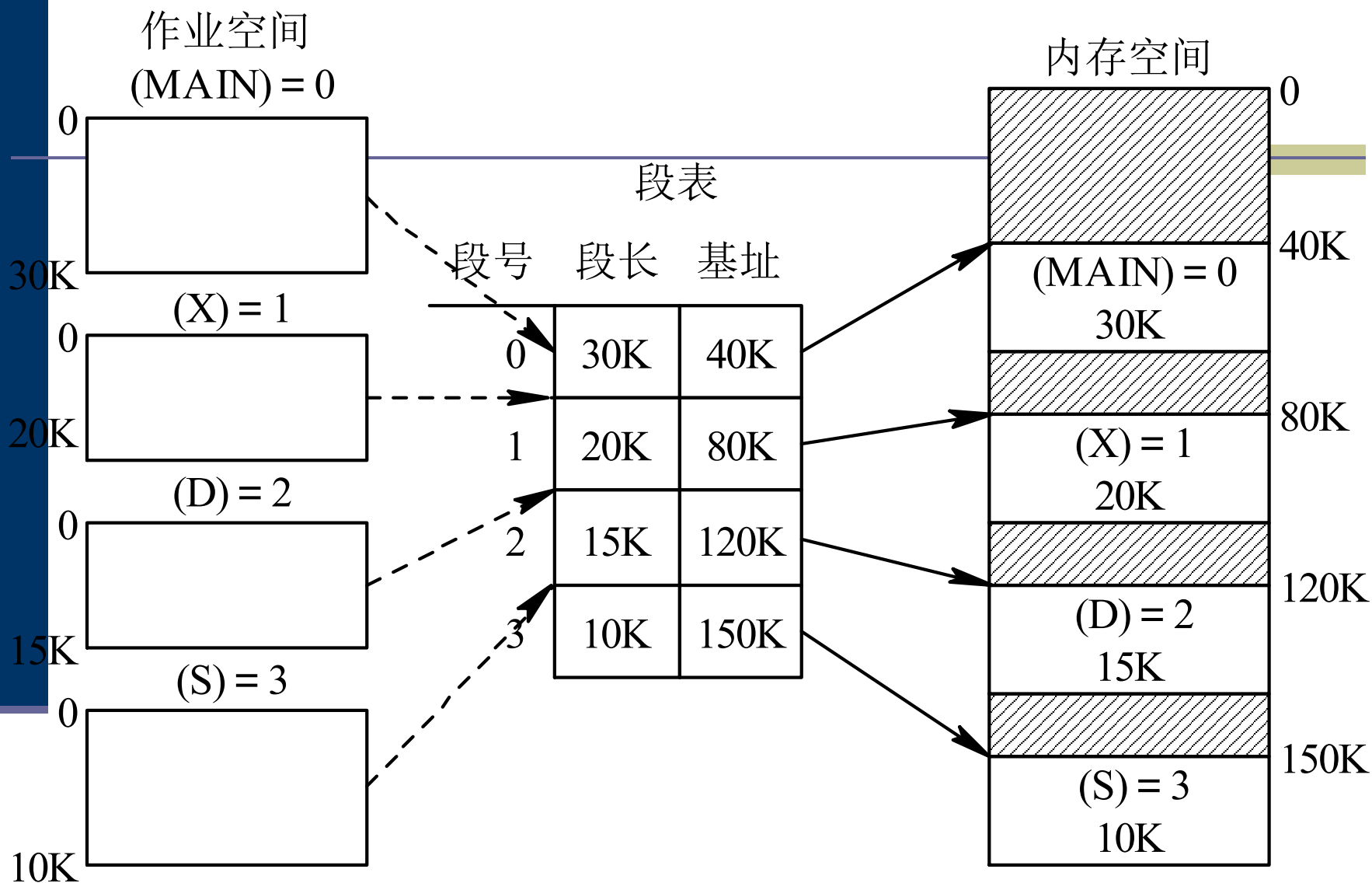


■ 内存划分

内存空间被动态的划分为若干个长度不相同的区域，称为**物理段**，每个物理段由起始地址和长度确定

■ 内存分配

以段为单位分配内存，每一个段在内存中占据连续空间（内存随机分割，需要多少分配多少），但各段之间可以不连续存放



系统需要维护的数据结构

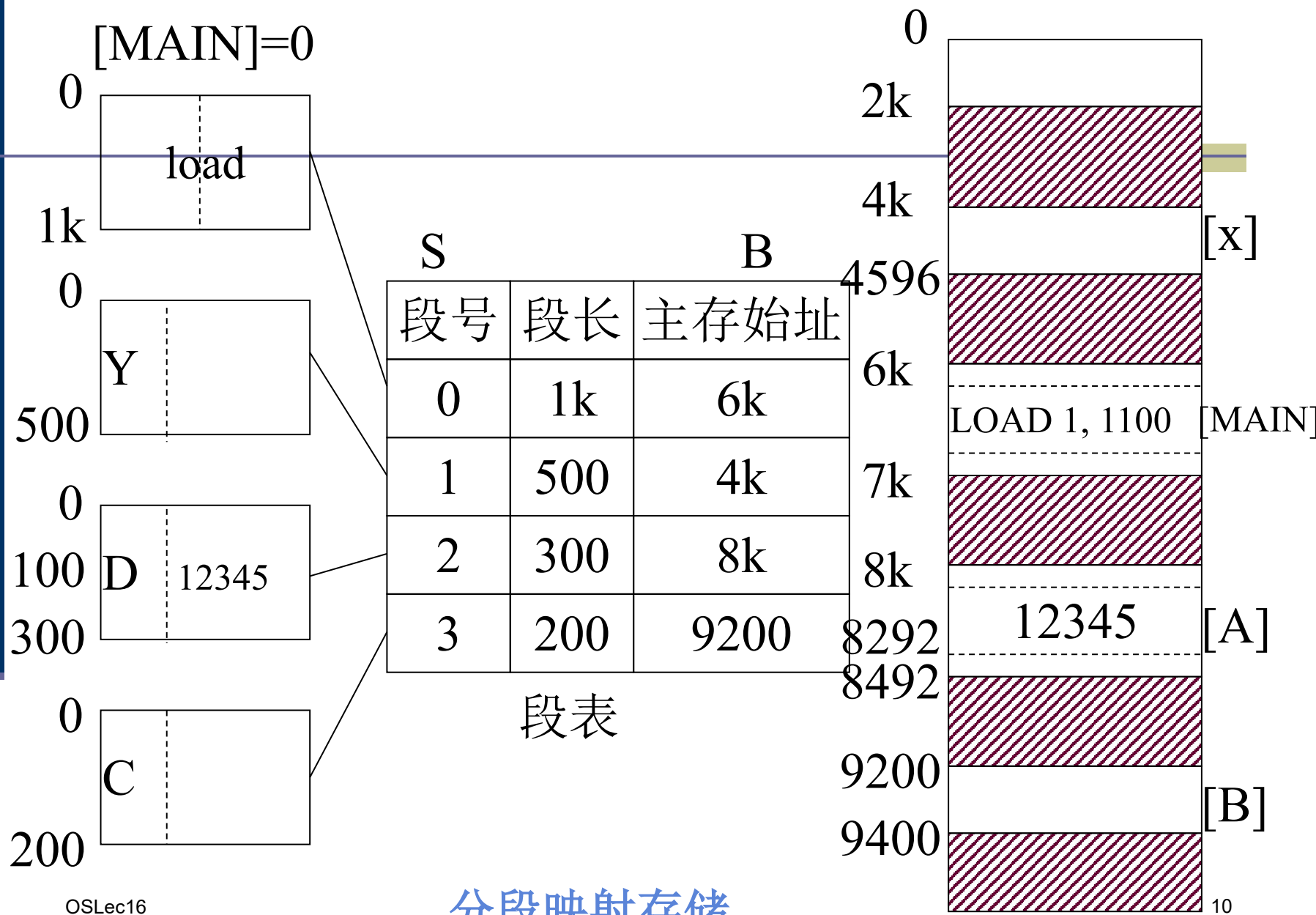
- **进程段表**：描述组成进程地址空间的各段，可以是指向系统段表中表项的索引。每段有段基址(base address)和段长度
- **系统段表**：系统内所有占用段
- **空闲段表**：内存中所有空闲段，可以结合到系统段表中

4.7.2 段地址映射

■ 地址映射数据结构

- 段地址映射的数据结构有**段表**、**段表首址指针**和**段表的长度**。段表首址指针和段表长度存放在进程自己的PCB中。
- 每一进程有个段表，程序的每一个段在段表中占用一个表目。

段号	段首址	段长度
0	58K	20K
1	100K	110K
2	260K	140K



段地址映射过程

- 程序地址字送入虚地址寄存器VR中。
- 取出段号S和段内位移W。
- 根据段表首址指针找到段表，查找段号为S的表目，得到该段的首地址。
- 把段首地址与段内位移相加，形成内存地址送入MR中，并以此地址访问内存。

控制寄存器

段表始址

段表长度

越界

段号S

位移量W

2

100100100

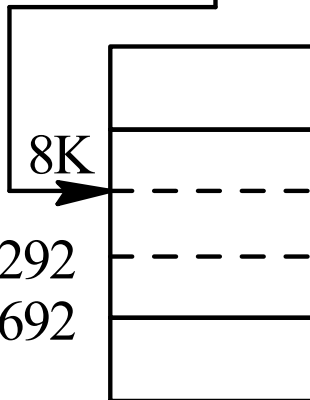
有效地址

+

段号	段长	基址
0	1 K	6 K
1	600	4 K
2	500	8 K
3	200	9200

8292

物理地址



主存

分段系统的地址变换过程

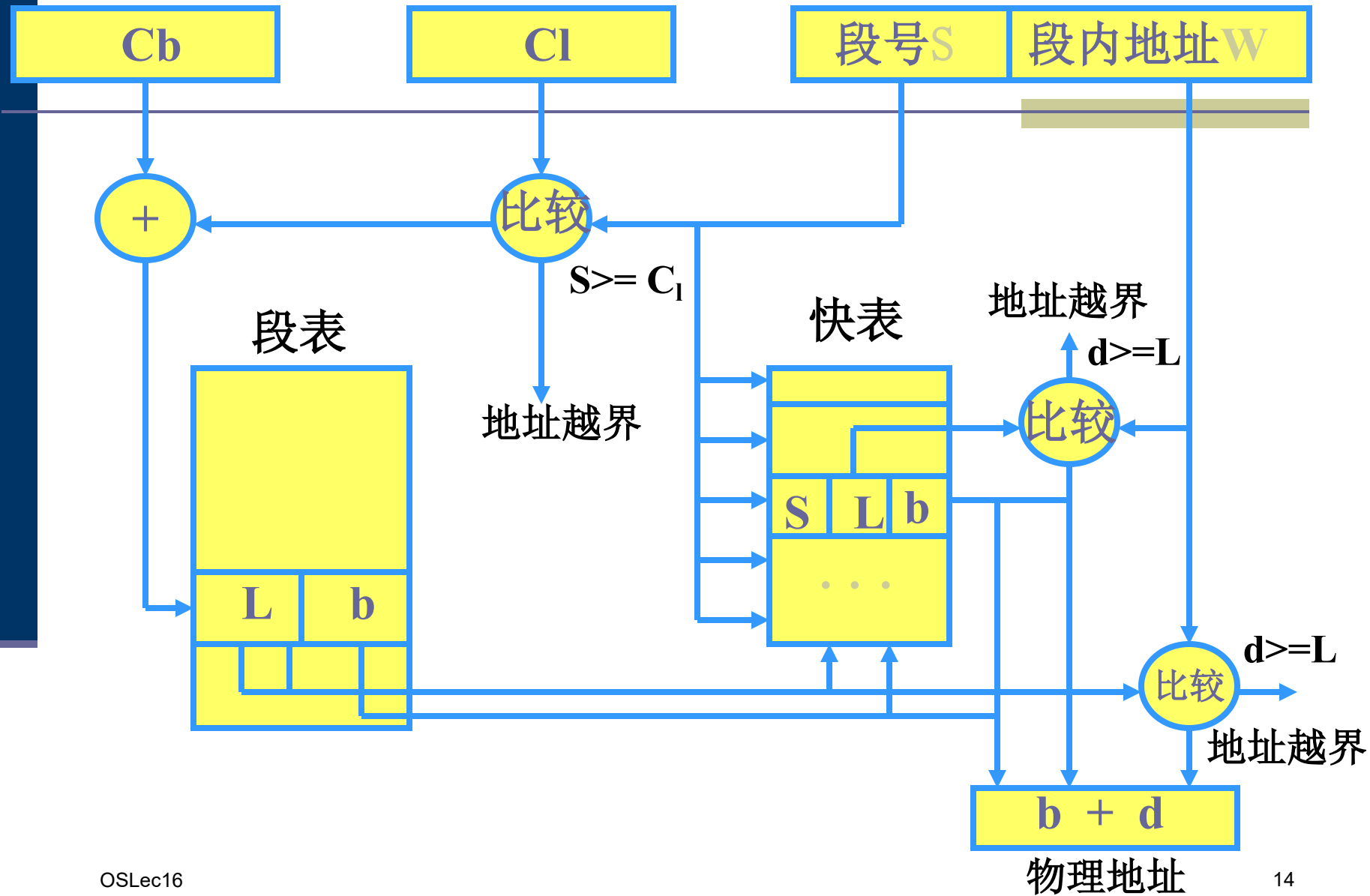
快表

- 同页地址变换一样，在段地址变换过程中，也有两次访问内存的问题。为了加快访问内存的速度也可采用快速存储器组成快表。

段表始址寄存器

段表长度寄存器

逻辑地址



内存的分配

- 与动态分区管理相同
- 空闲块管理： 空闲块表（队列）
- 内存分配算法（三种）： 首次 最佳 最坏

请求段式存储管理

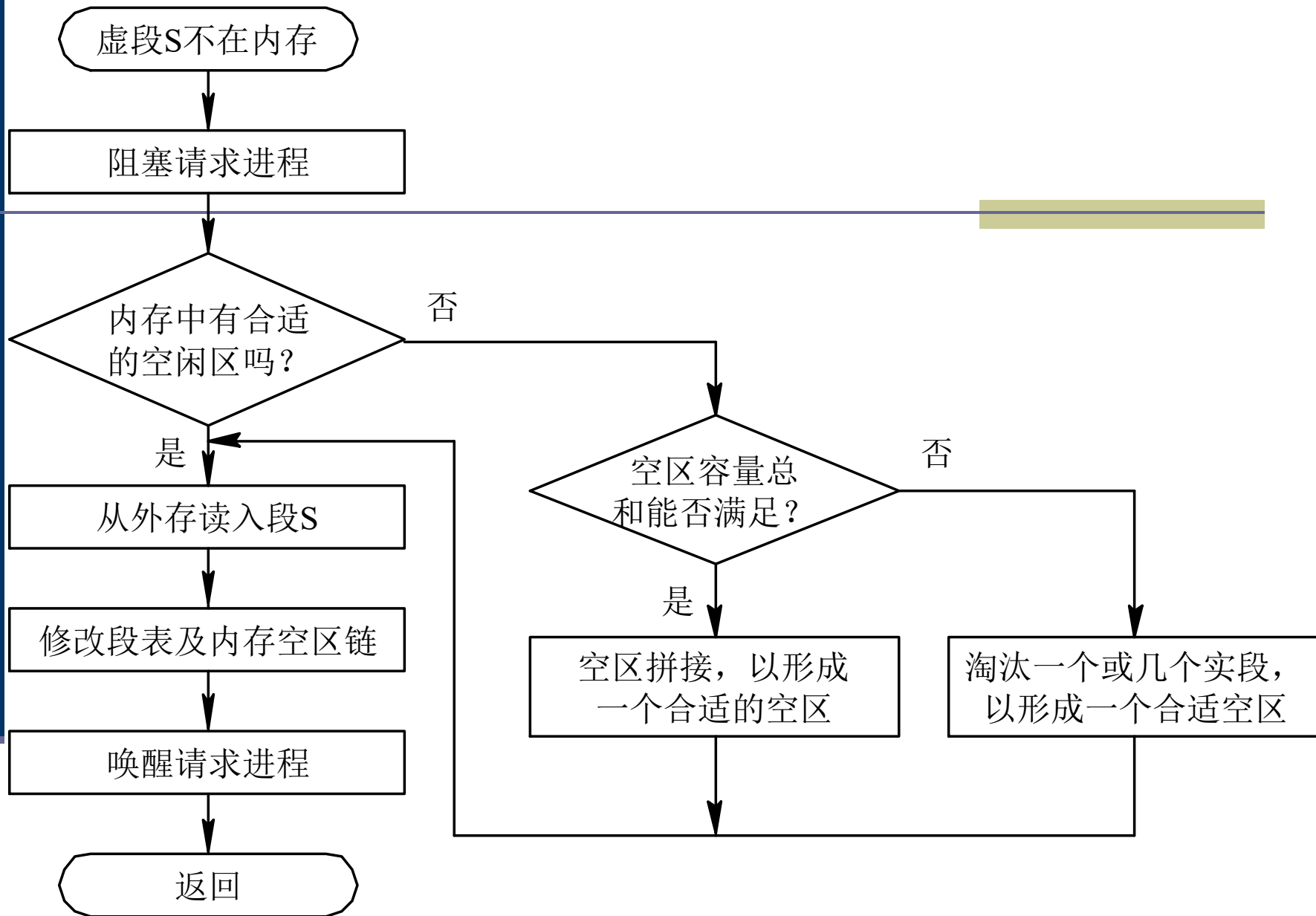
段表SMT内容：段号（段名）、内存始址、段长，
增加：

- 特征位（在/不在内存，是否可共享）
- 存取权限位（读，写，执行）
- 标志位（是否修改过，能否移动）
- 扩充位（固定长/可扩充）
- 外存始址

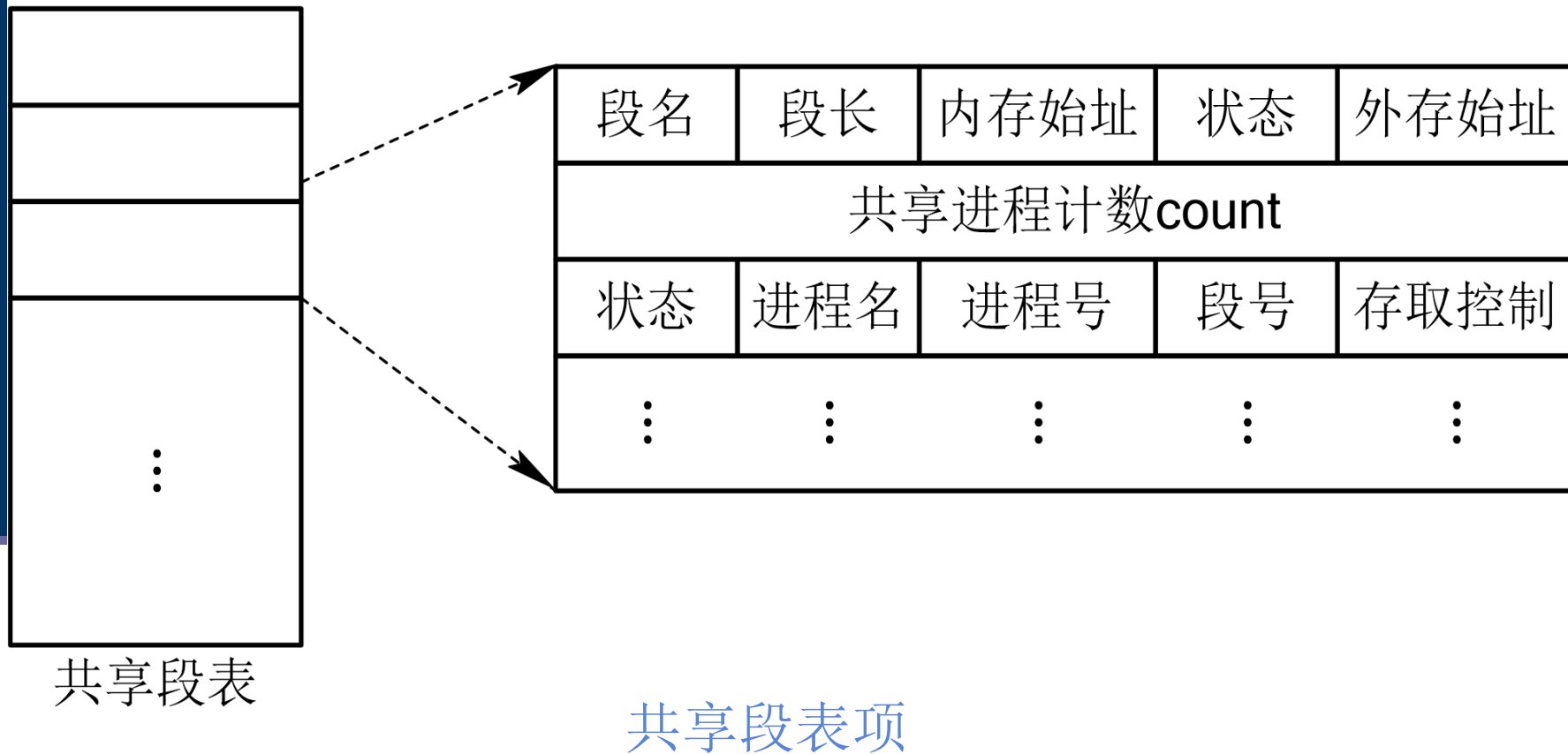
缺段中断处理

检查内存中是否有足够的空闲空间

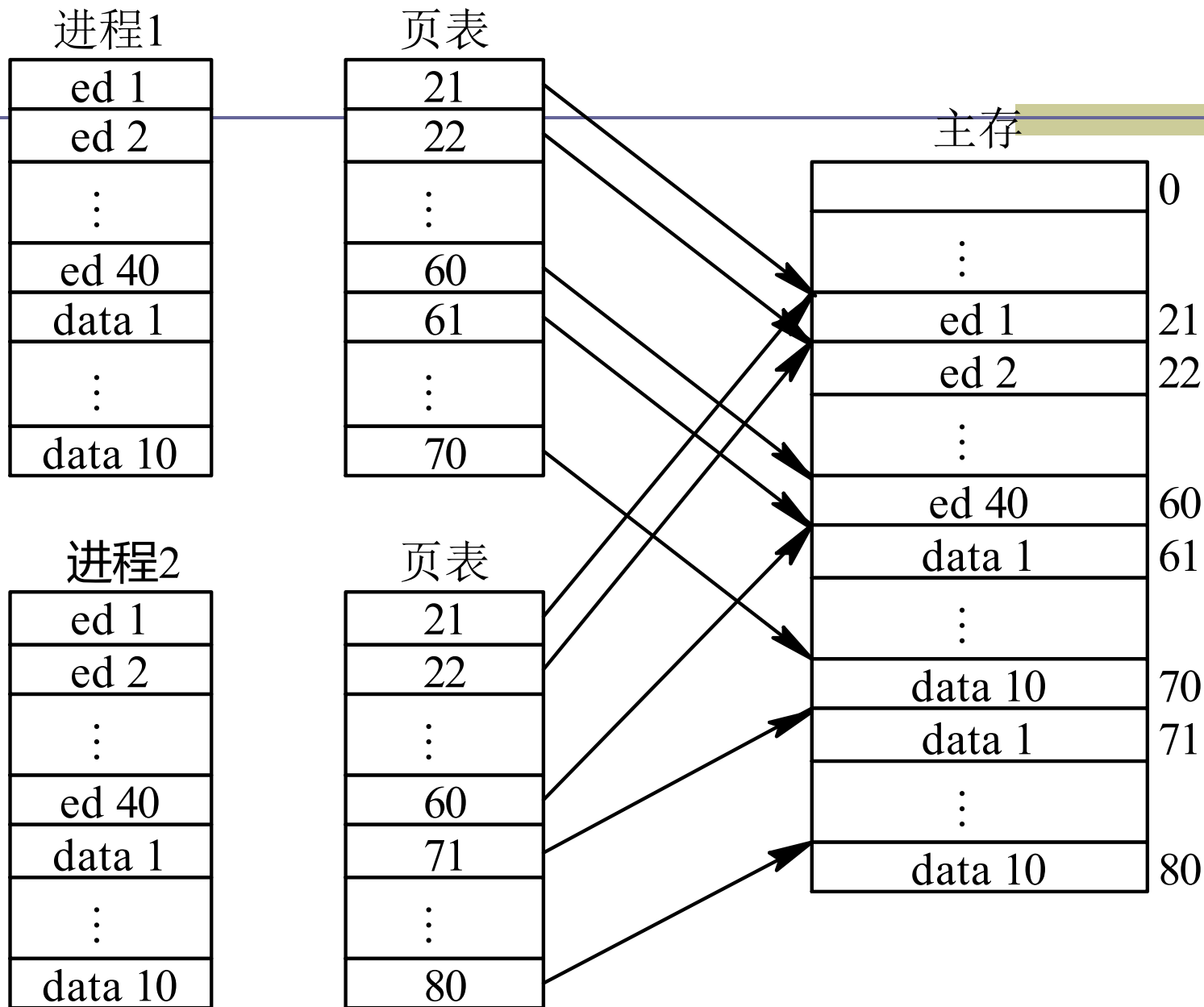
- ①若有，则装入该段，修改有关数据结构，中断返回
- ②若没有，检查内存中空闲区的总和是否满足要求，是则应采用紧缩技术，转①；否则，淘汰一（些）段，转①



4.7.3 分段的共享与保护



信息共享

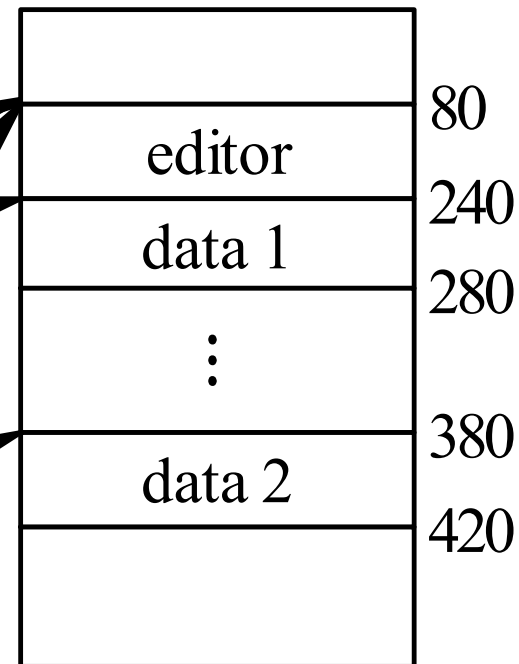




段表

段长	基址
160	80
40	240

160	80
40	380



分段系统中共享editor的示意图

共享段的分配与回收

■ 共享段的分配

- 第一个请求使用该共享段的进程，由系统为该共享段分配一物理区，再把共享段调入该区，并填写共享段表。
- 又有其它进程需要调用该共享段时只需在调用进程的段表中，增加一表项，填写该共享段的物理地址；修正共享段的段表。

■ 共享段的回收

- 执行 $\text{count} := \text{count} - 1$ 操作。若结果为0，则须由系统回收该共享段的物理内存，以及取消在共享段表中该段所对应的表项，否则只是取消调用者进程在共享段表中的有关记录。

分段保护

- 越界检查

- 存取控制检查

 - 只读

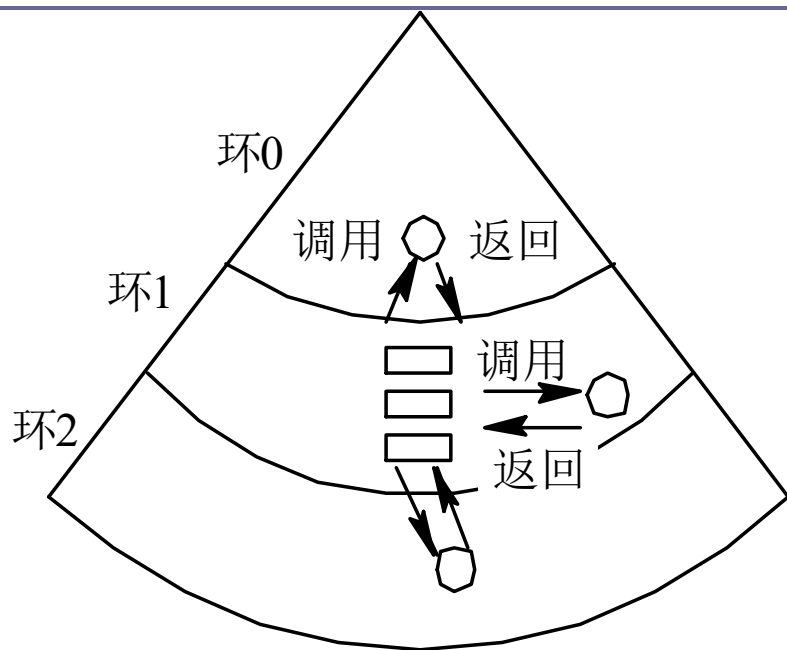
 - 只执行

 - 读/写

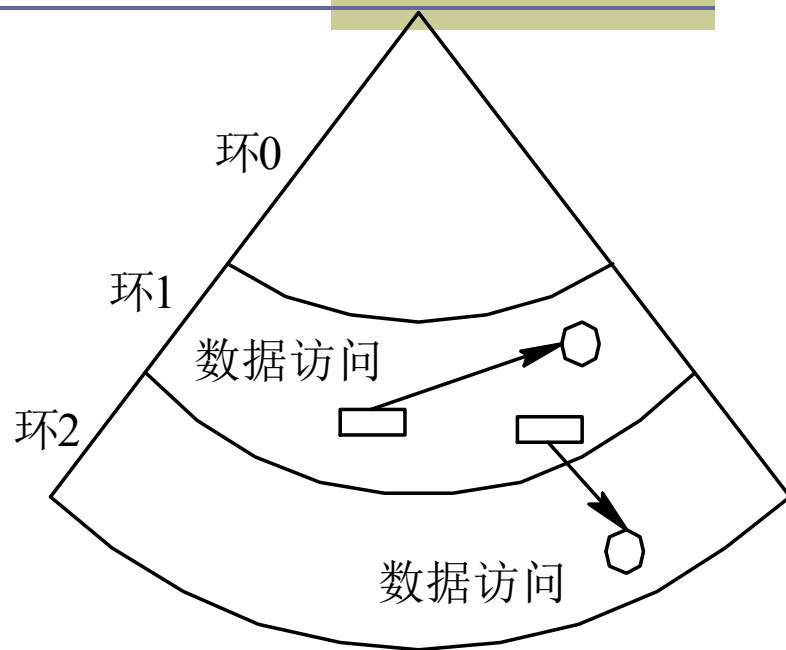
- 环保护机构

 - 一个程序可以访问驻留在相同环或较低特权环中的数据。

 - 一个程序可以调用驻留在相同环或较高特权环中的服务。



(a) 程序间的控制传输

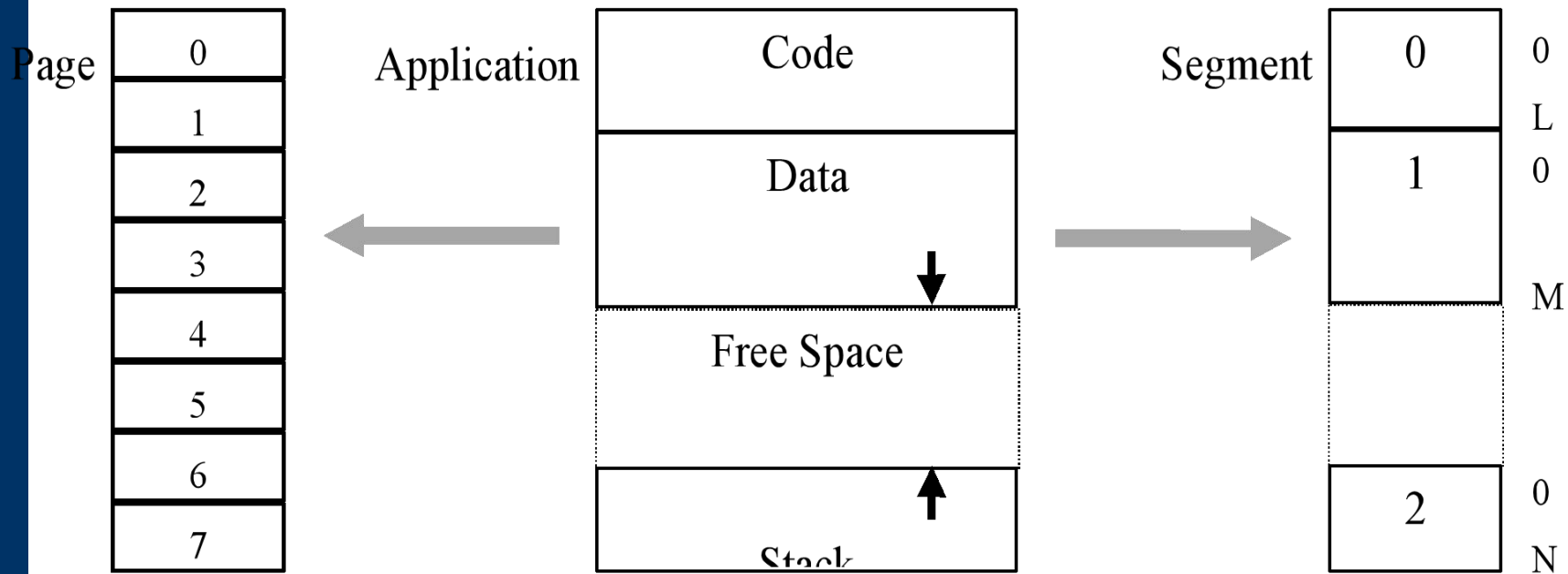


(b) 数据访问

环保护机构

4.7.4 分段管理方案评价

- 分段与分页主要有以下差别：
 - 段是依据程序的逻辑结构划分的，页是按内存线性空间物理划分的。
 - 段式技术中程序地址空间是二维的，分页技术中程序地址空间是一维的。
 - 段是面向用户的，页对用户而言是透明的。
 - 段长由用户决定，且各段的大小一般不相等，唯一的限制是最大长度。页长是由系统决定的，各页的长度必须相等。
 - 段的共享比页的共享更容易。



Note: **↑↓** Dynamic Data Increment

页式管理与段式管理的比较

段式存储管理方案小结

■ 优点：

- 消除了内碎片
- 通过请求分段存储管理方式提供了大量虚存
- 允许动态增加段的长度
- 便于动态装入和链接
- 便于程序共享
- 便于存储保护

■ 缺点：

- 进行地址变换和实现内存紧凑（靠拢）要花费处理机时间；
- 在辅存上管理可变长度的段比较困难；

思考：与可变分区存储管理方案的相同点与不同点？

What you need to do?

- 复习课本4.5节的内容
- 课后作业：习题17、22

See you next time!