

第三章 内存管理

目录

1. 什么是内存管理?
2. 虚拟内存管理有什么不同?

1.什么是内存管理?

◆ 准备工作:

- ◆ 存储器结构
- ◆ 进程运行原理

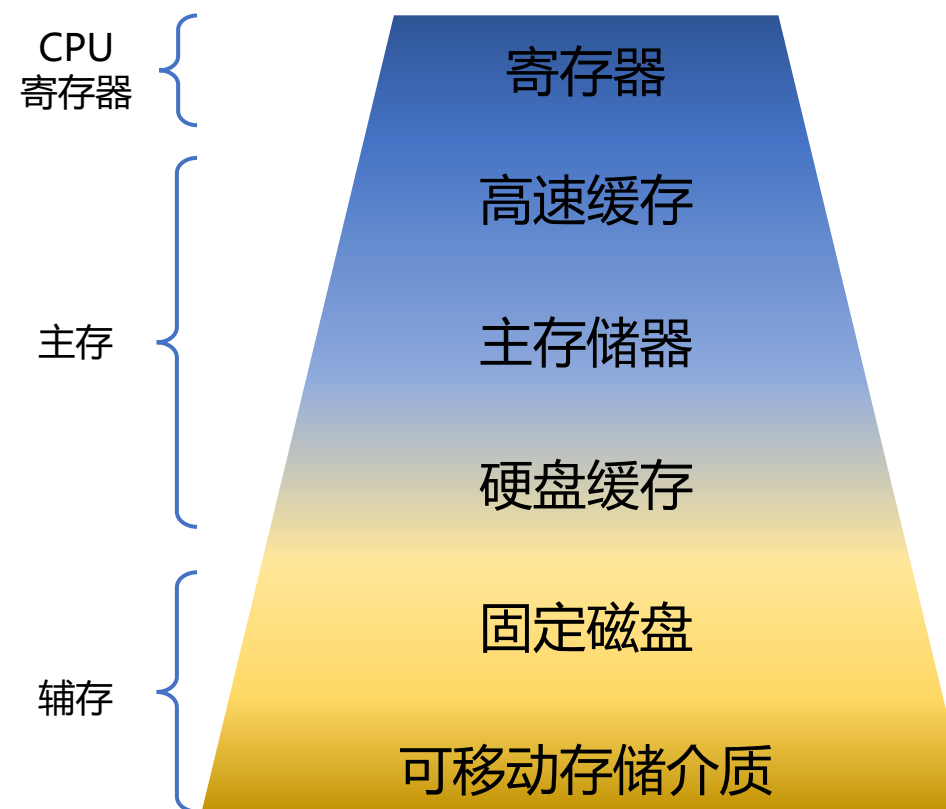
◆ 内存管理方式

- ◆ 连续分配管理方式
- ◆ 非连续分配管理方式

1.什么是内存管理?

• 存储器的多层结构

- ◆ 寄存器
- ◆ 高速缓存
- ◆ 主存储器
- ◆ 硬盘缓存
- ◆ 固定磁盘
- ◆ 可移动存储介质

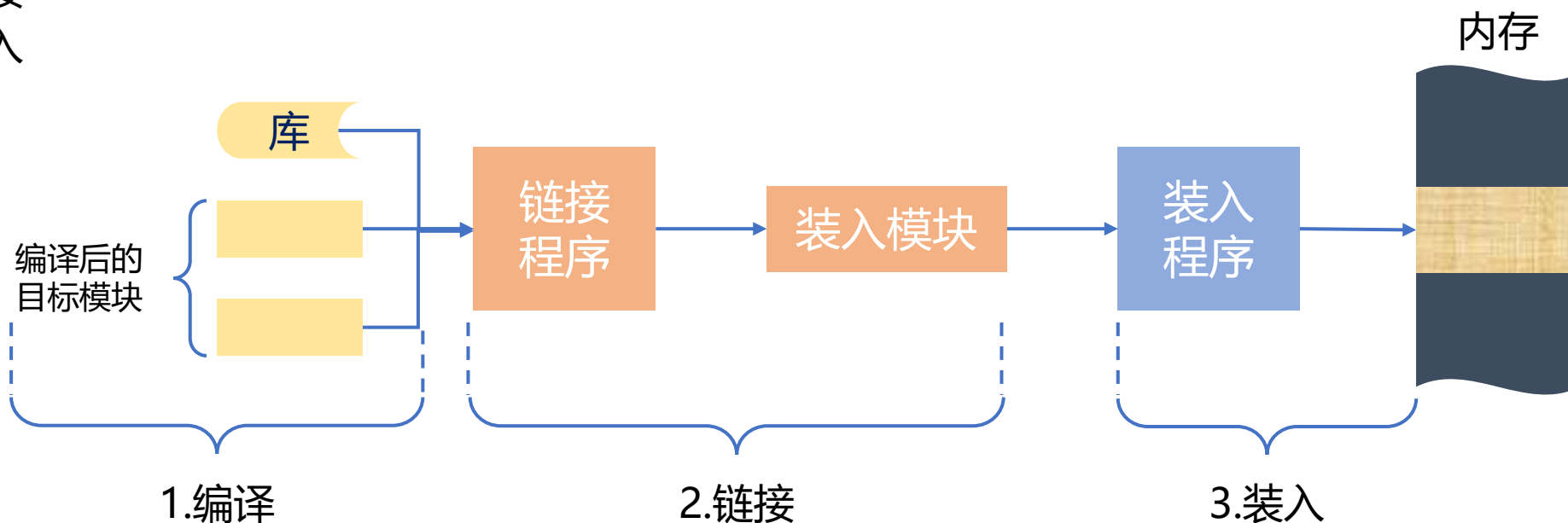


1.什么是内存管理?

• 进程运行的基本原理

◆ 用户程序 -> 进程

- ◆ 编译
- ◆ 链接
- ◆ 装入

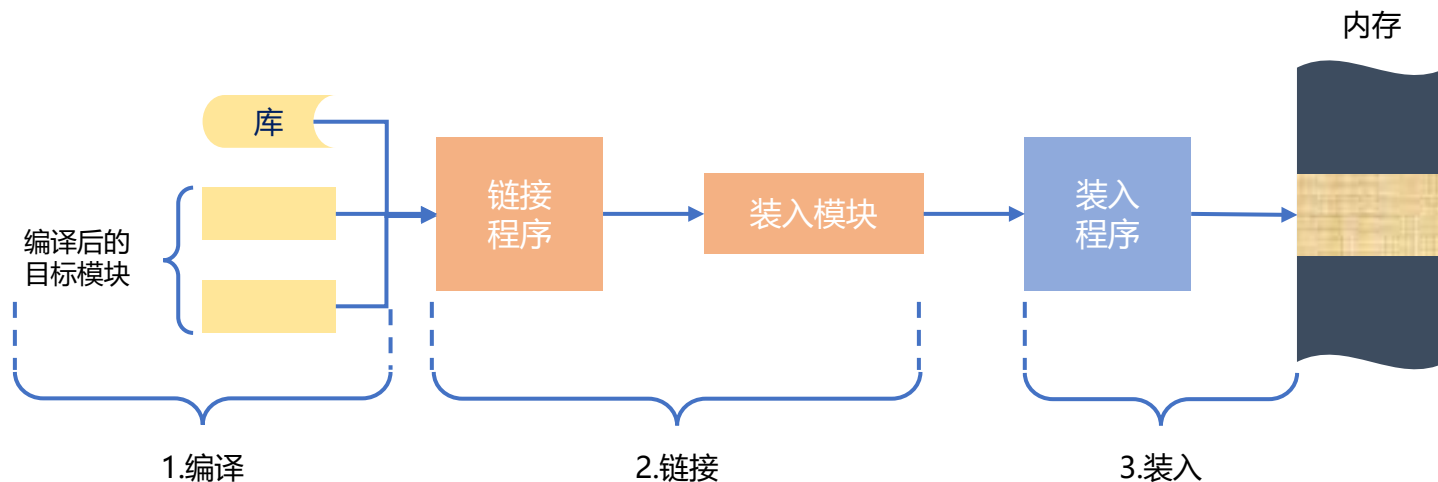


1.什么是内存管理?

• 进程运行的基本原理

◆ 程序的链接

- ◆ 静态链接
- ◆ 装入时动态链接
- ◆ 运行时动态链接



1.什么是内存管理?

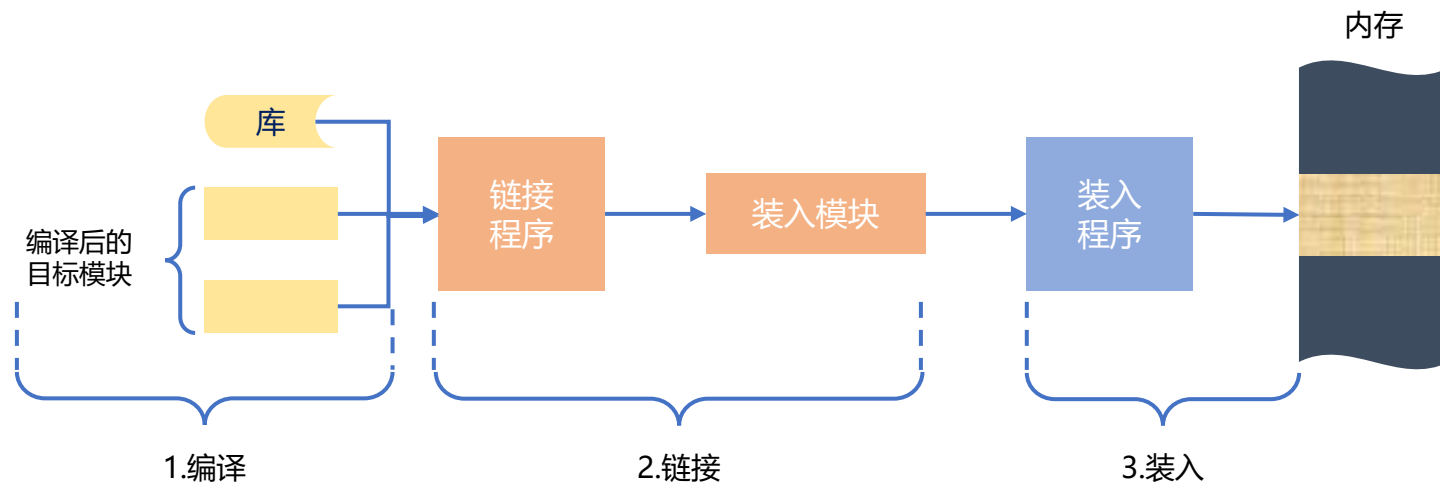
• 进程运行的基本原理

◆ 程序的装入

- ◆ 绝对装入
- ◆ 可重定位装入
- ◆ 动态运行时装入

◆ 两个细节

- ◆ 逻辑地址与物理地址
- ◆ 内存保护

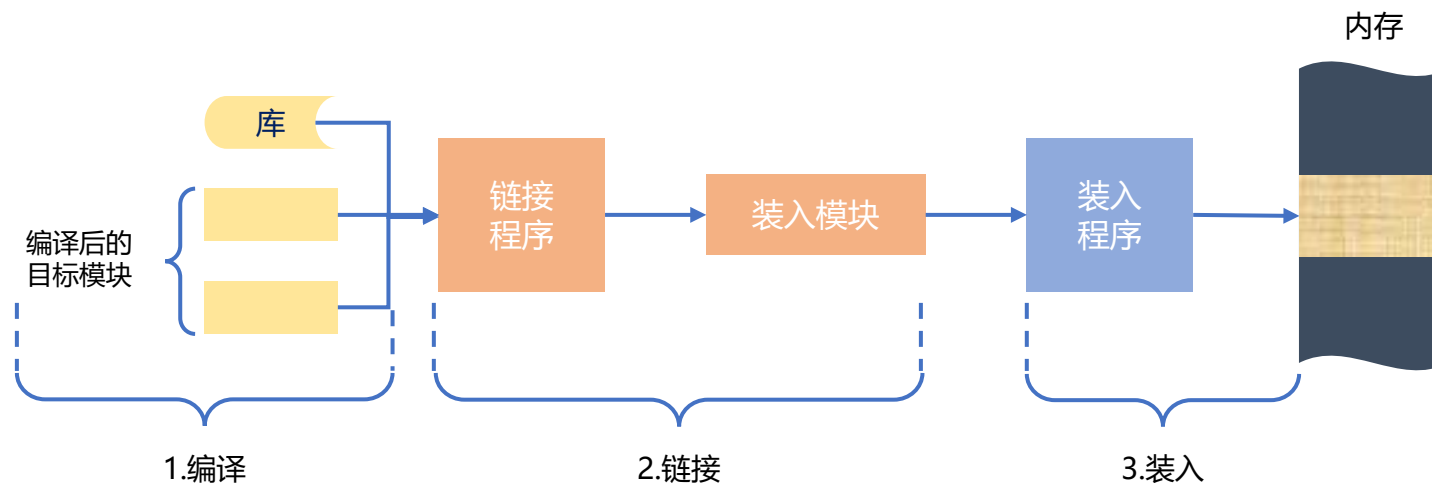


1.什么是内存管理?

• 进程运行的基本原理

◆ 内存扩充的两种方式

- ◆ 覆盖
- ◆ 交换



1.什么是内存管理?

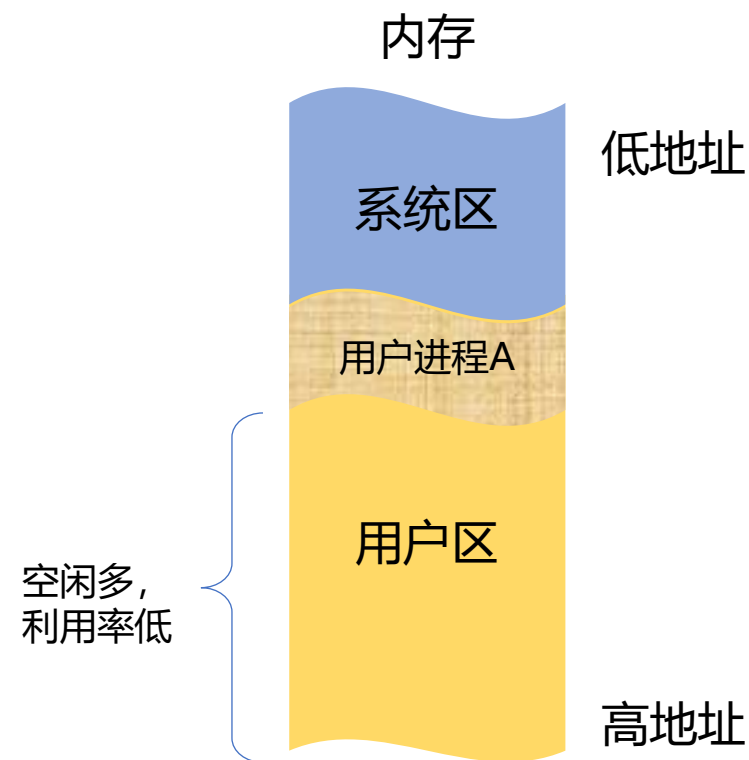
• 内存管理方式

◆ 连续分配管理方式

- ◆ 单一连续分配
- ◆ 固定分区分配
- ◆ 动态分区分配

优点:
实现简单;
无外部碎片;
不一定需要内存保护

缺点:
只能用于单用户、单任务OS;
有内部碎片;
存储器利用率低;



1.什么是内存管理?

• 内存管理方式

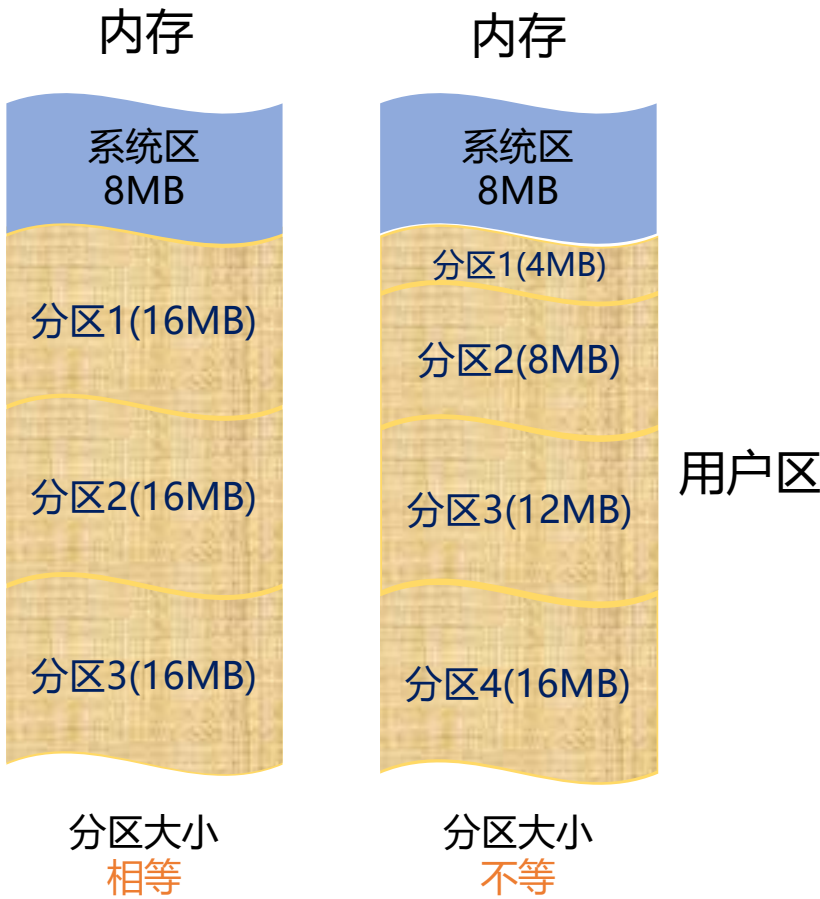
◆ 连续分配管理方式

- ◆ 单一连续分配
- ◆ 固定分区分配
- ◆ 动态分区分配

优点:
实现简单;
无外部碎片;

缺点:
1.较大用户程序时, 需要采用覆盖技术, 降低了性能;
2.会产生内部碎片, 利用率低

分区说明表: 记录各分区的分配与回收状态			
分区号	大小(MB)	起始地址	状态
1	4	2	未分配
2	8	7	未分配
3	12	15	已分配
...

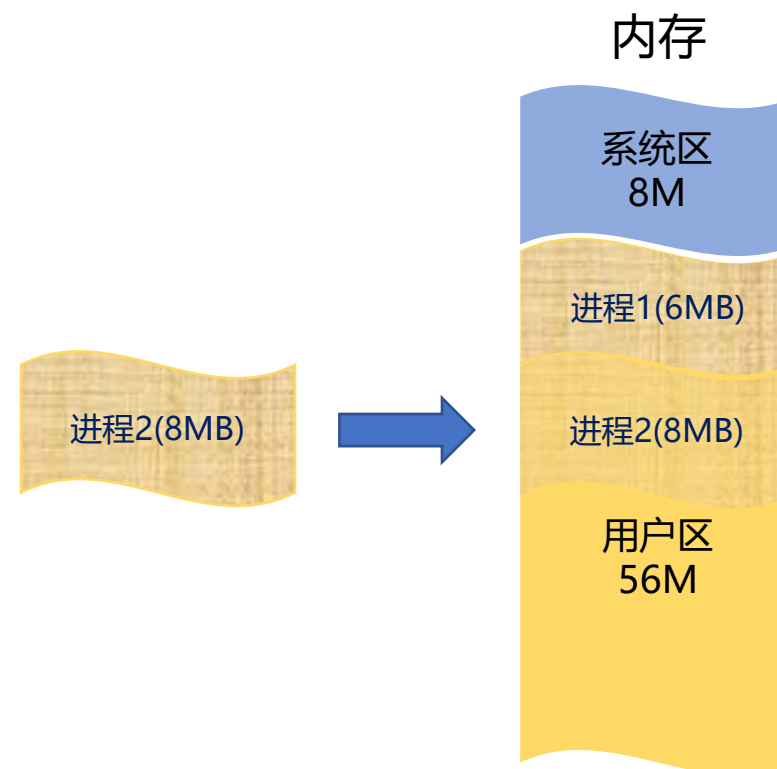


1.什么是内存管理?

• 内存管理方式

◆ 连续分配管理方式

- ◆ 单一连续分配
- ◆ 固定分区分配
- ◆ 动态分区分配
 - ◆ 怎么记录内存的使用情况?
 - ◆ 选择哪个分区给新进程?
 - ◆ 已使用的分区怎么回收?



1.什么是内存管理?

• 内存管理方式

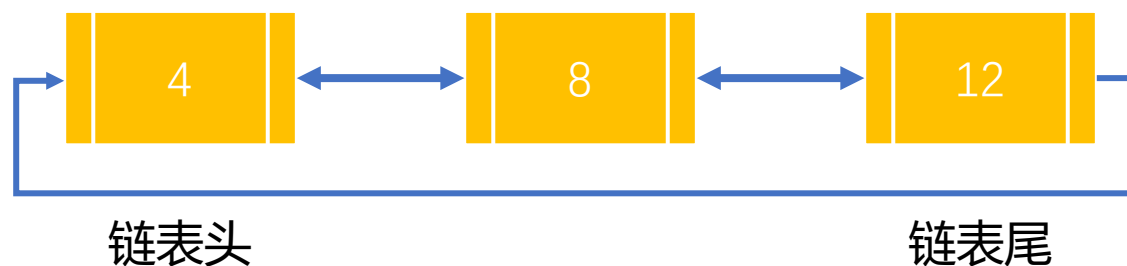
◆ 连续分配管理方式

- ◆ 单一连续分配
- ◆ 固定分区分配
- ◆ 动态分区分配
- ◆ 怎么记录内存的使用情况?

空闲分区表

空闲分区表：记录各分区的分配与回收状态			
分区号	大小(MB)	起始地址	状态
1	4	2	空闲
2	8	7	空闲
3	12	15	空闲

空闲分区链



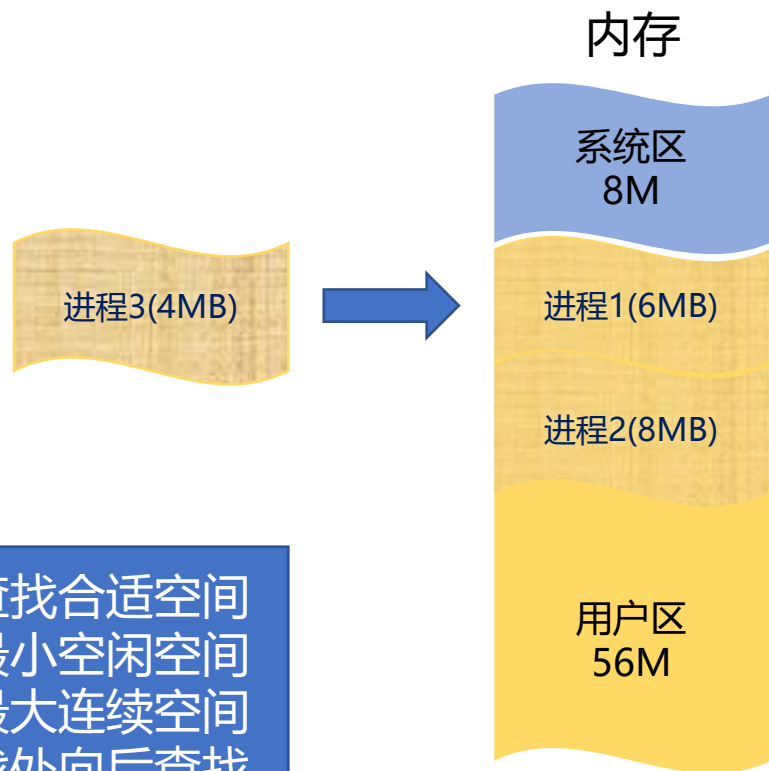
1.什么是内存管理?

• 内存管理方式

◆ 连续分配管理方式

- ◆ 单一连续分配
- ◆ 固定分区分配
- ◆ 动态分区分配
 - ◆ 选择哪个分区给新进程?

首次适应算法: 从低地址查找合适空间
最佳适应算法: 优先使用最小空闲空间
最坏适应算法: 优先使用最大连续空间
临近适应算法: 从上次查找处向后查找

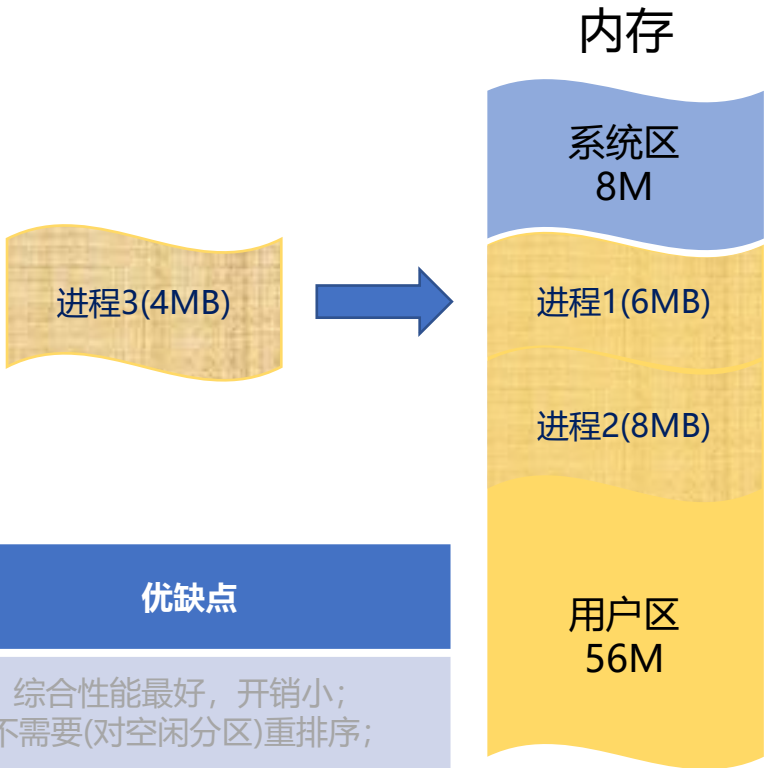


1.什么是内存管理?

• 内存管理方式

◆ 连续分配管理方式

- ◆ 单一连续分配
- ◆ 固定分区分配
- ◆ 动态分区分配
- ◆ 选择哪个分区给新进程?



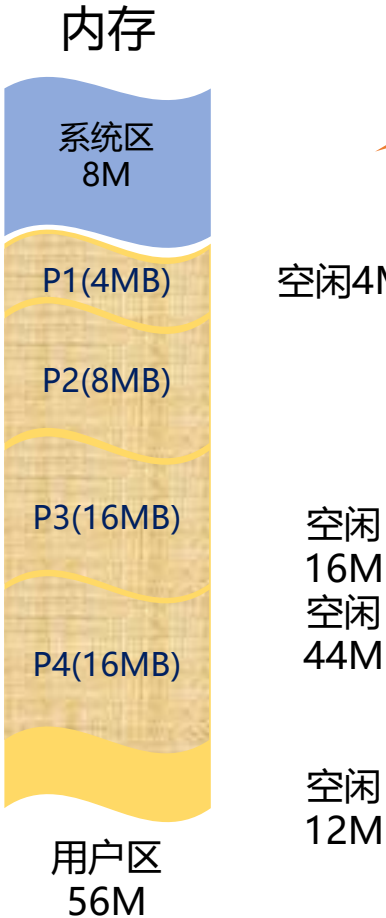
算法	算法思想	分区排序	优缺点
首次适应	从低地址查找合适空间	地址递增排列	综合性能最好, 开销小; 不需要(对空闲分区)重排序;
最佳适应	优先使用最小空闲空间	容量递增排列	更容易满足大进程需求; 小碎片多, 开销大, 需要重排序;
最坏适应	优先使用最大连续空间	容量递减排列	小碎片少; 不利于大进程, 开销大;
临近适应	从上次查找处向后查找	地址递增排列 (循环链表)	不用每次从链表头查找, 开销小; 会使高地址大分区被用完;

1.什么是内存管理?

• 内存管理方式

◆ 连续分配管理方式

- ◆ 单一连续分配
- ◆ 固定分区分配
- ◆ 动态分区分配
- ◆ 已使用的分区怎么回收?



回收后相邻空间要合并;
更新空闲分区表和/
空闲分区链记录;

空闲分区表: 记录各分区的分配与回收状态

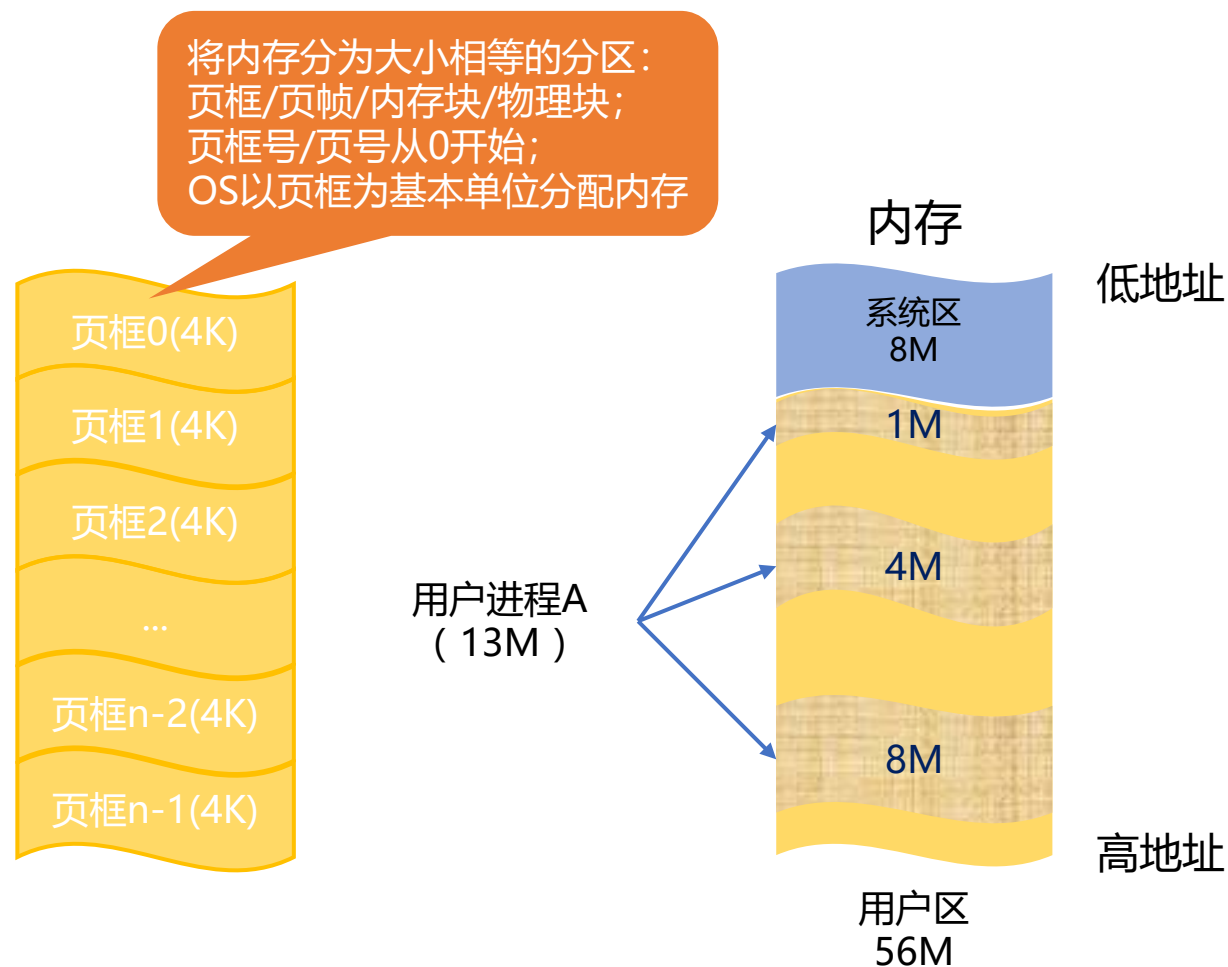
分区号	大小(MB)	起始地址	状态
5	12	45	空闲
3	44	13	空闲

1.什么是内存管理?

• 内存管理方式

◆ 非连续分配管理方式

- ◆ 基本分页存储管理方式
- ◆ 基本分段存储管理方式
- ◆ 段页式管理方式



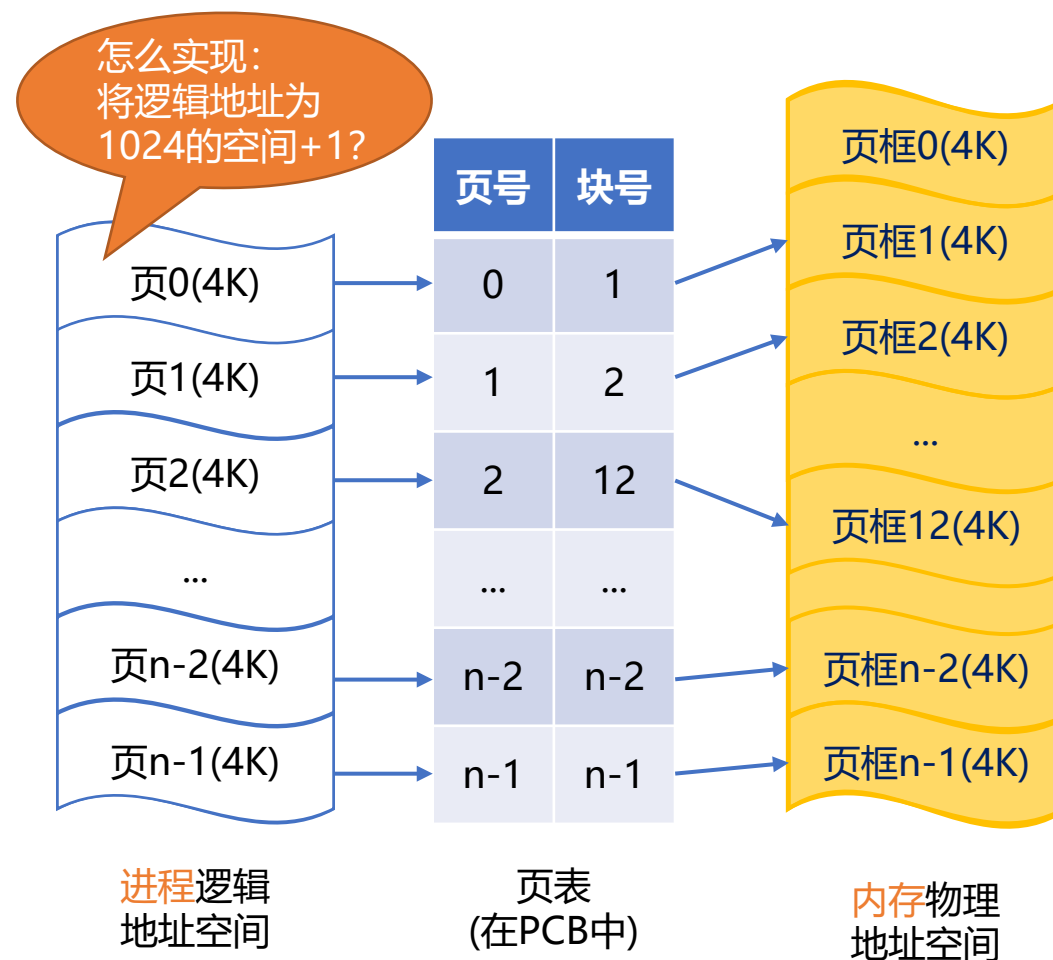
1.什么是内存管理?

• 内存管理方式

◆ 非连续分配管理方式

◆ 基本分页存储管理方式

- ◆ 页/页面、页框、块
- ◆ 页表
- ◆ 基本地址变换机构



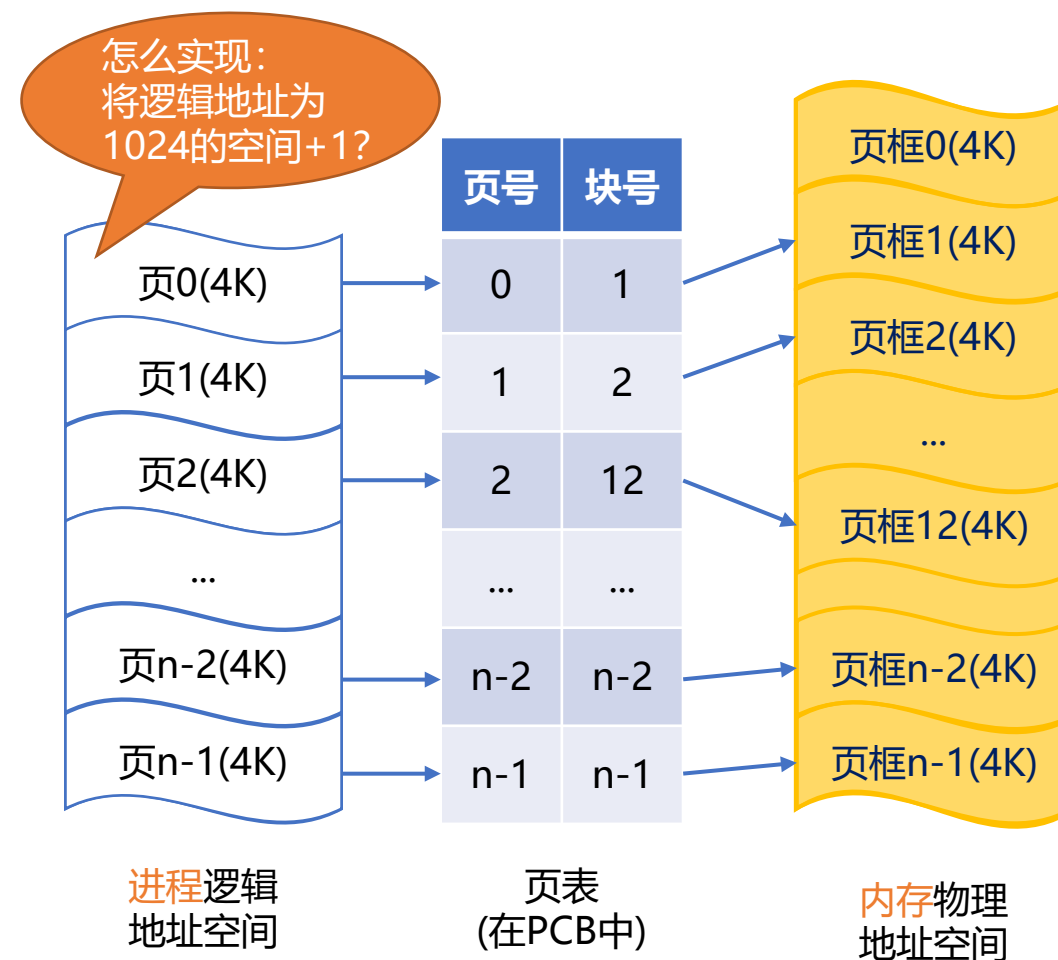
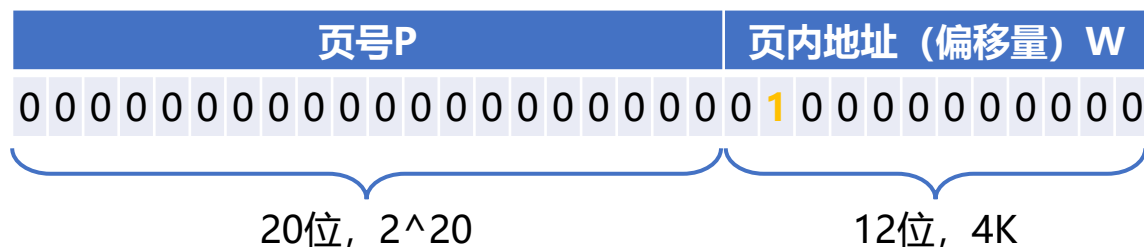
1.什么是内存管理?

• 内存管理方式

◆ 非连续分配管理方式

◆ 基本地址变换机构

- ◆ 物理地址 = (页号 → 块号) + 偏移量
- ◆ 页号 $P = \text{逻辑地址} A / \text{页面长度 (大小)} L$
- ◆ 偏移量 $W = \text{逻辑地址} A \% \text{页面长度} L$
- ◆ $P = A \gg 12; W = A \& 4095$



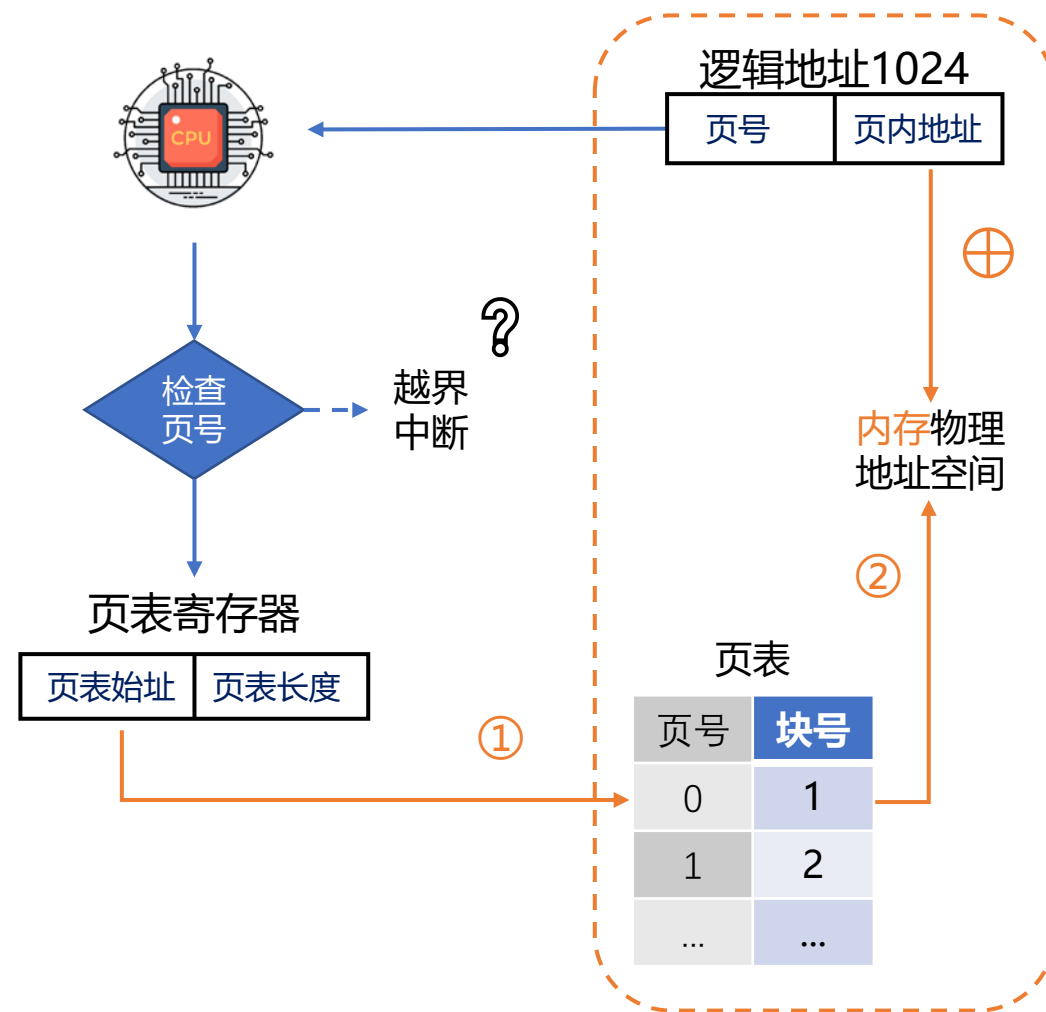
1.什么是内存管理?

• 内存管理方式

◆ 非连续分配管理方式

◆ 基本地址变换机构

- ◆ 页式管理中地址空间是一维的
- ◆ 每次访存都需要地址转换，必须足够快
- ◆ 页表不能太大，会降低内存利用率



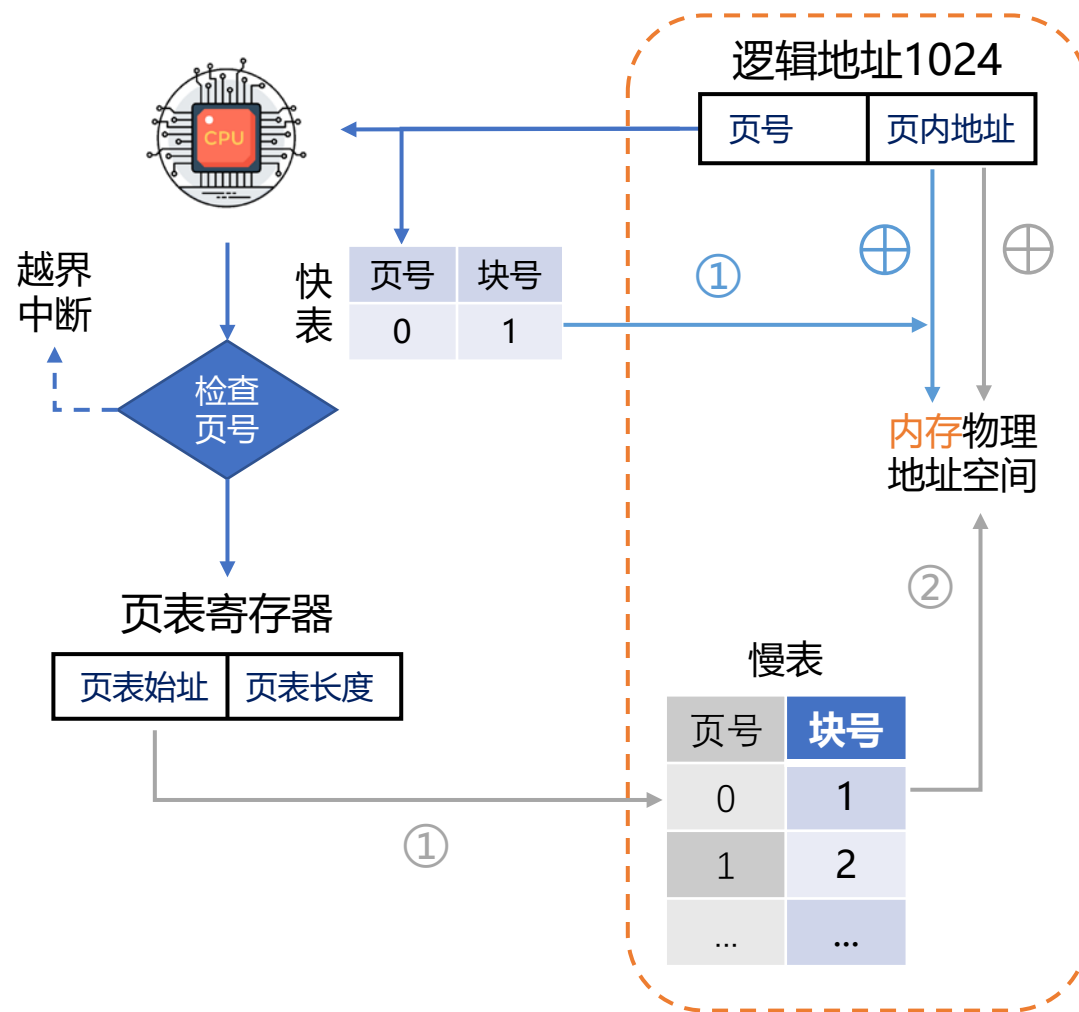
1.什么是内存管理?

• 内存管理方式

◆ 非连续分配管理方式

◆ 具有快表的地址变换机构

- ◆ 直接将页号与快表页号比较
- ◆ 匹配成功，取块号+偏移量形成地址
- ◆ 匹配失败，访问主存页表，并同步到快表（局部性原理）



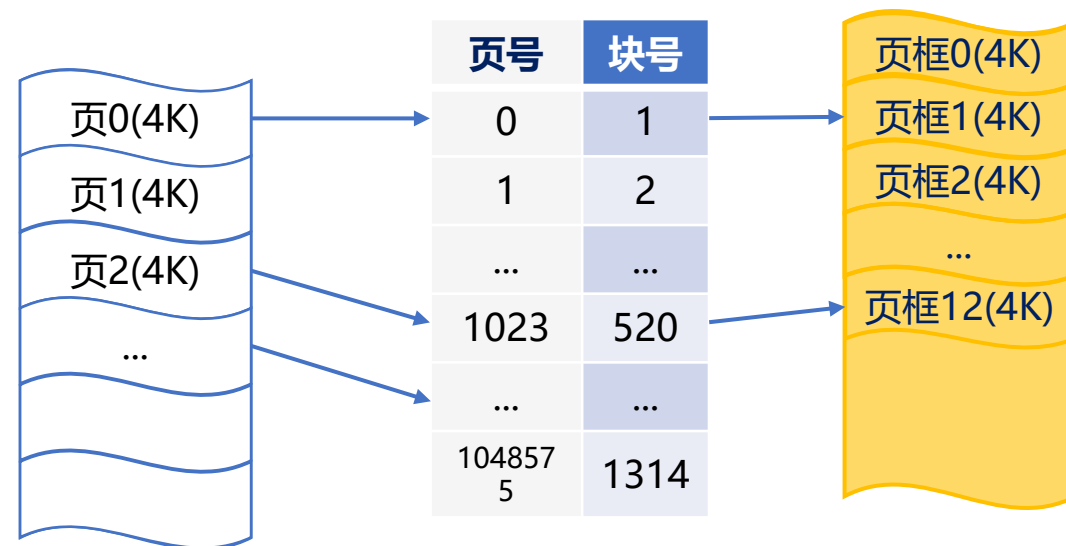
1.什么是内存管理?

• 内存管理方式

◆ 非连续分配管理方式

◆ 两级页表

- ◆ 页表连续存放, 占用大量连续空间
- ◆ 一段时间内只需要访问部分特定页面
- ◆ 页表项分组/分页离散存储
- ◆ 建页目录表管理离散页表



最多 2^{20} 个页表项, 每个占4B
 页表占用内存空间(页框数):
 $2^{10} = 2^{20} * 4B / 2^{12}$

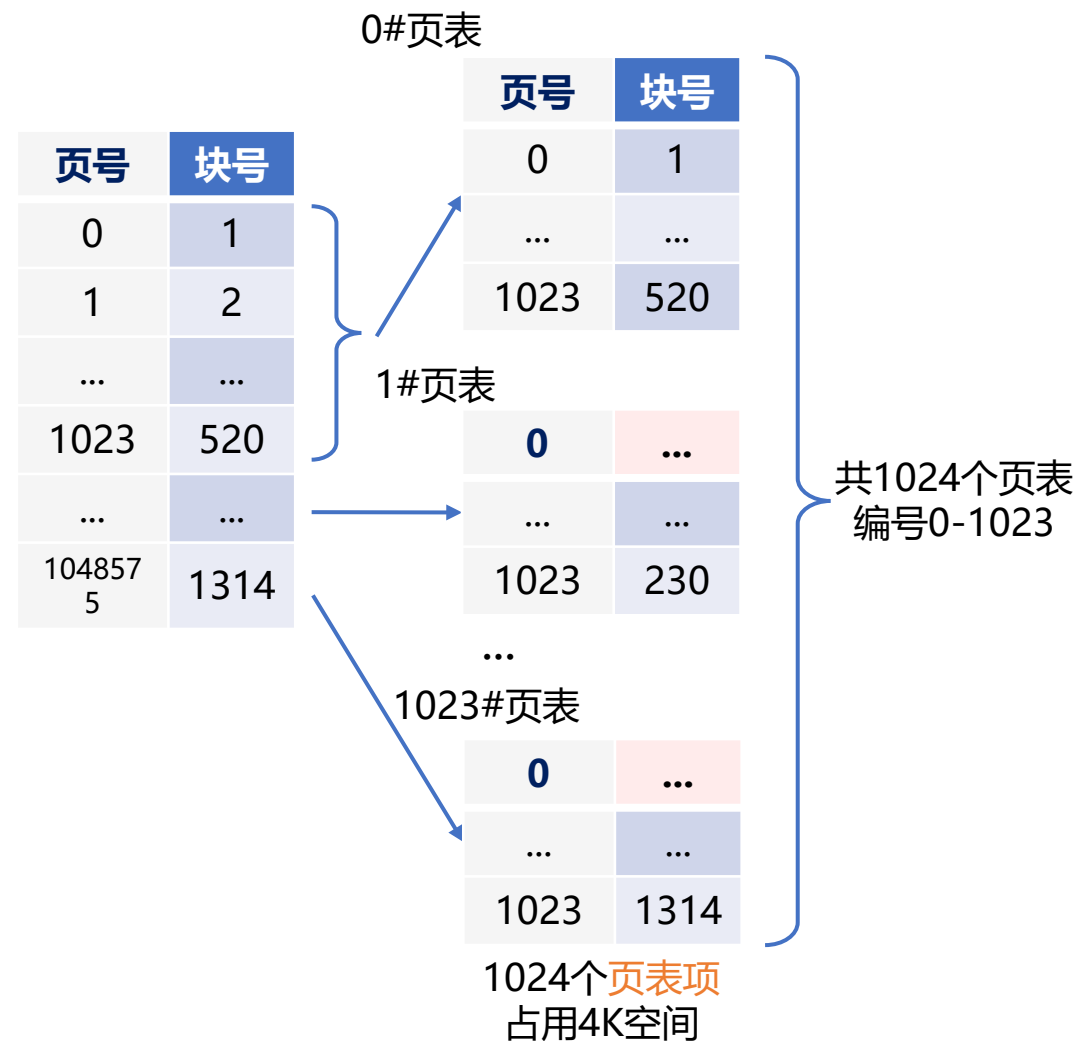
1.什么是内存管理?

• 内存管理方式

◆ 非连续分配管理方式

◆ 两级页表

- ◆ 页表连续存放，占用大量连续空间
- ◆ 一段时间内只需要访问部分特定页面
- ◆ 页表项分组/分页离散存储
- ◆ 建页目录表管理离散页表



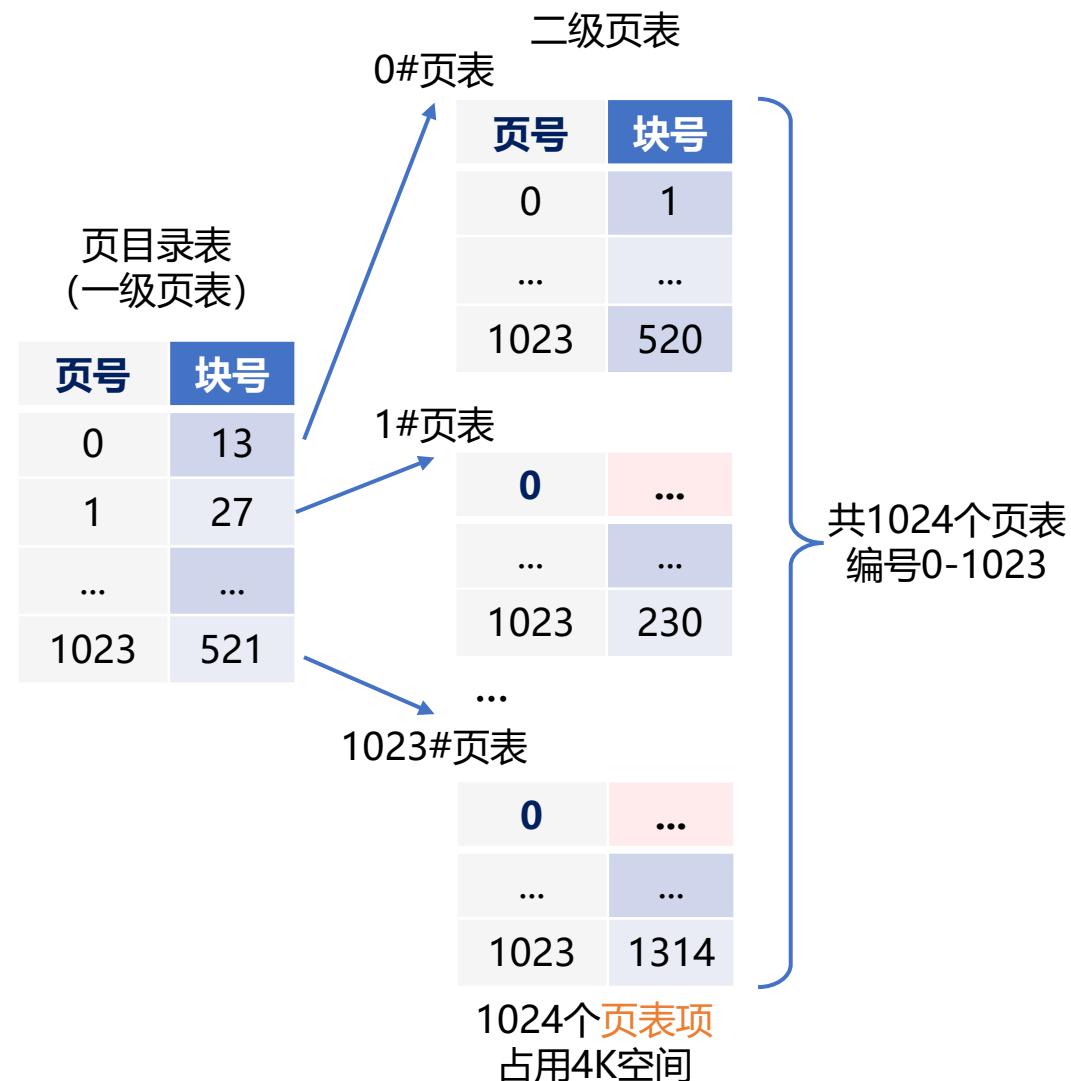
1.什么是内存管理?

• 内存管理方式

◆ 非连续分配管理方式

◆ 两级页表

- ◆ 页表连续存放，占用大量连续空间
- ◆ 一段时间内只需要访问部分特定页面
- ◆ 页表项分组/分页离散存储
- ◆ 建页目录表管理离散页表



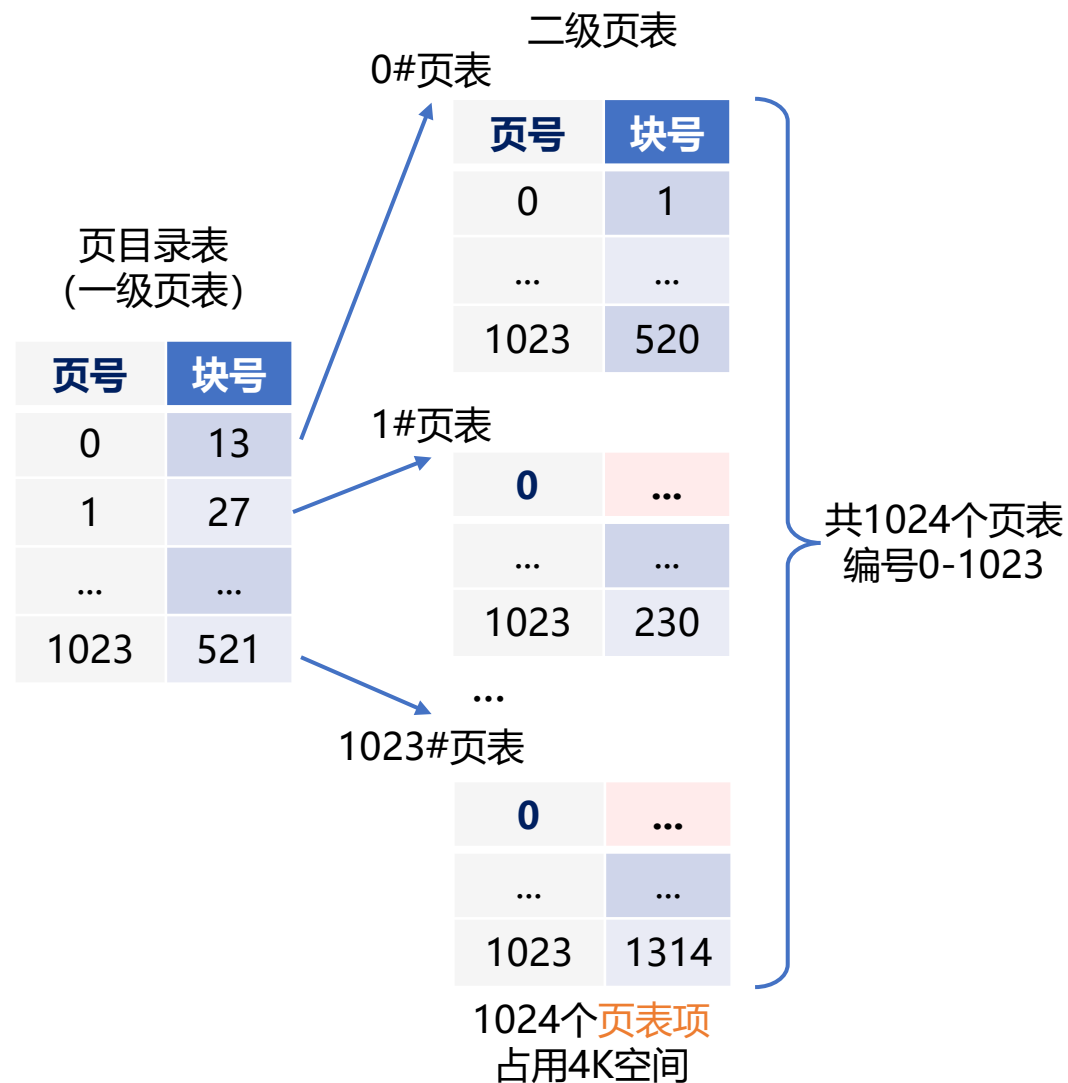
1.什么是内存管理?

• 内存管理方式

◆ 非连续分配管理方式

◆ 两级页表

- ◆ 将逻辑地址拆分成三部分
- ◆ 从PCB中读取页目录表始址
- ◆ 根据一级页号查出二级页表位置
- ◆ 根据二级页号查内存块号，加偏移量计算物理地址

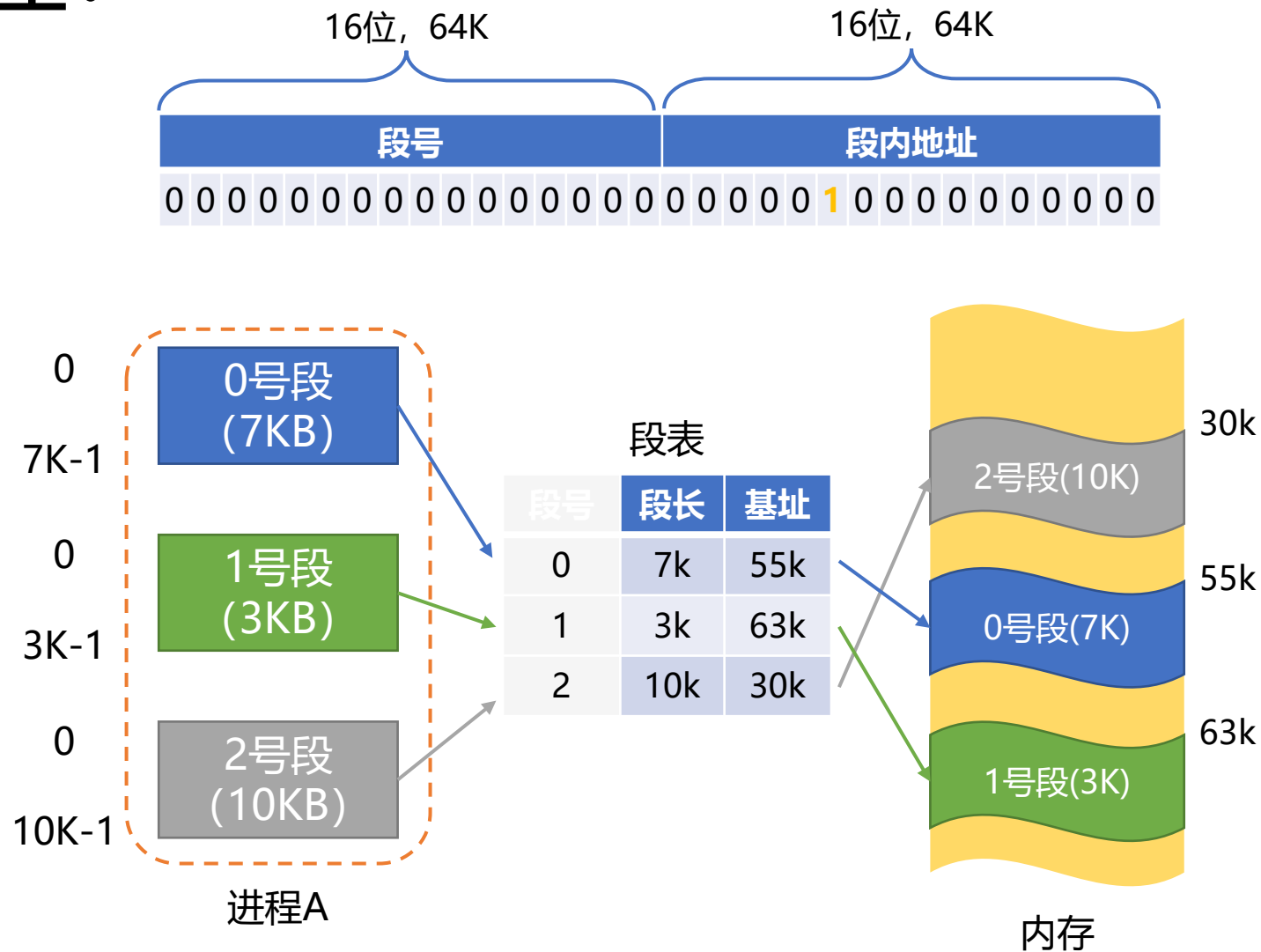


1.什么是内存管理?

• 内存管理方式

◆ 非连续分配管理方式

- ◆ 基本分页存储管理方式
- ◆ 基本分段存储管理方式
 - ◆ 分段
 - ◆ 段表
 - ◆ 地址变换机构
 - ◆ 段的共享与保护
- ◆ 段页式管理方式



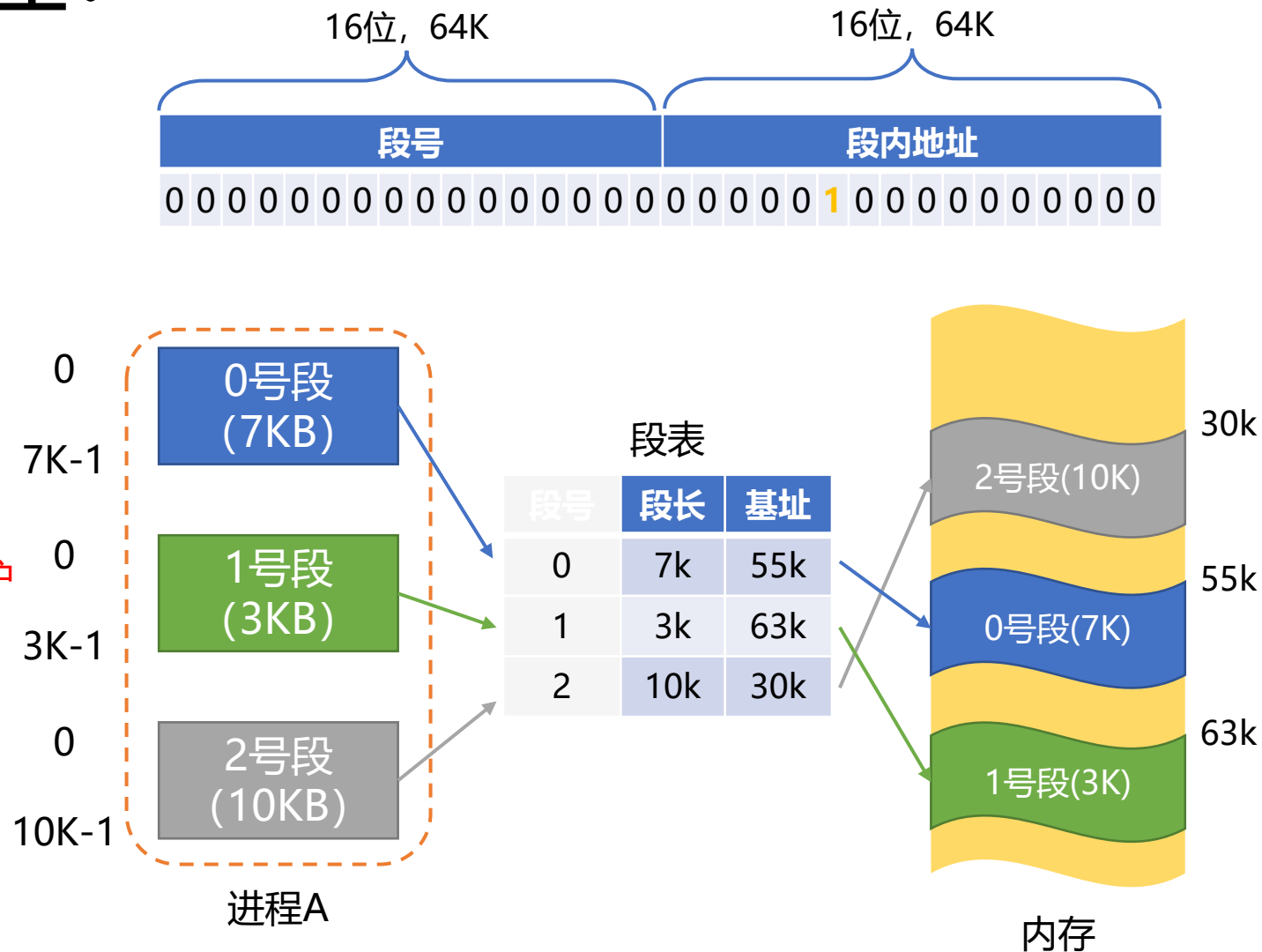
1.什么是内存管理?

• 内存管理方式

◆ 非连续分配管理方式

◆ 分页与分段方式对比

- ◆ 页->物理单位
- ◆ 段->逻辑单位
- ◆ 分页->一维地址空间
- ◆ 分段->二维地址空间
- ◆ 分段更容易信息共享和保护



1.什么是内存管理?

• 内存管理方式

◆ 非连续分配管理方式

- ◆ 基本分页存储管理方式
- ◆ 基本分段存储管理方式
- ◆ 段页式管理方式

	优点	缺点
分页管理	内存利用率高, 不会产生外部碎片, 少量内部碎片	不好按照逻辑模块实现信息共享和保护
分段管理	容易按逻辑模块实现信息共享和保护	段长较大时, 不便分配空间; 会产生外部碎片

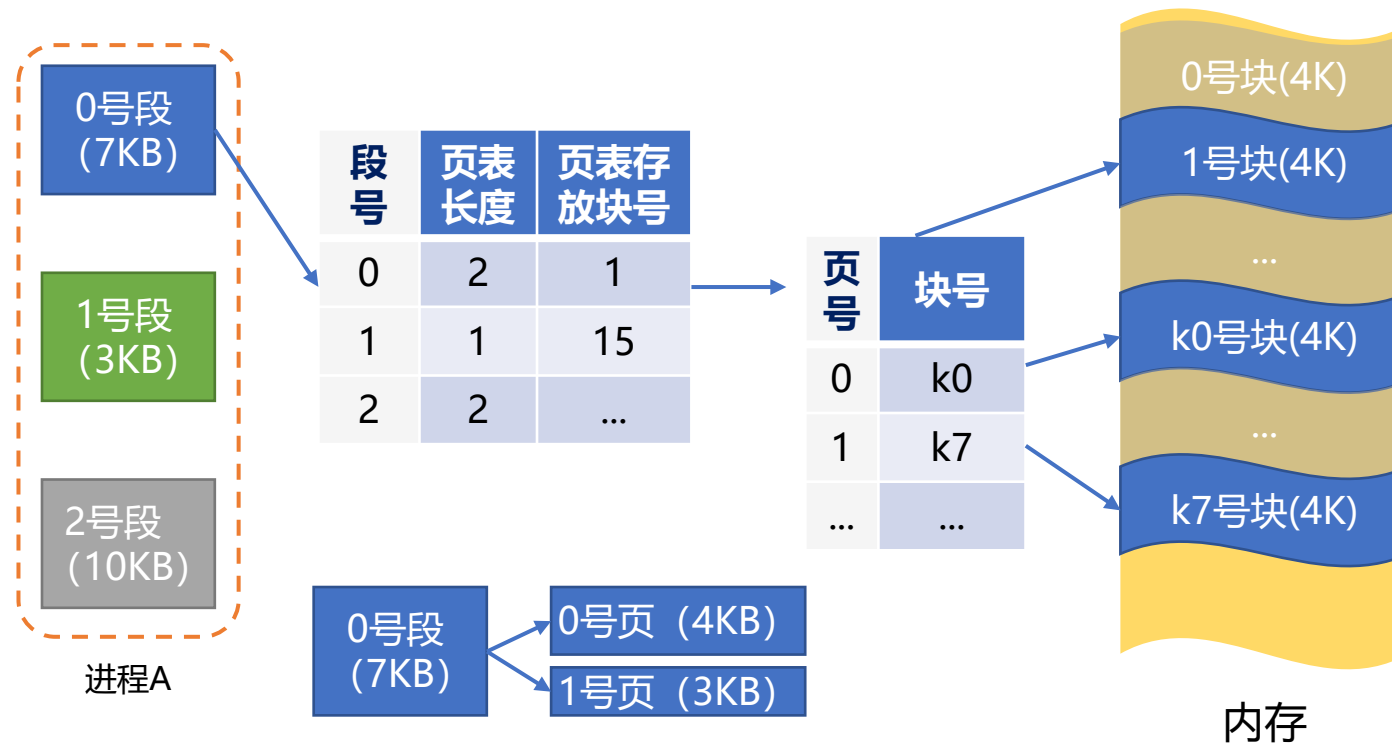
1.什么是内存管理?

• 内存管理方式

◆ 非连续分配管理方式

◆ 段页式管理方式

- ◆ 先分段，再分页
- ◆ 1个进程->1个段表
- ◆ 1个段表项->1个页表
- ◆ 1个页表->多个物理块



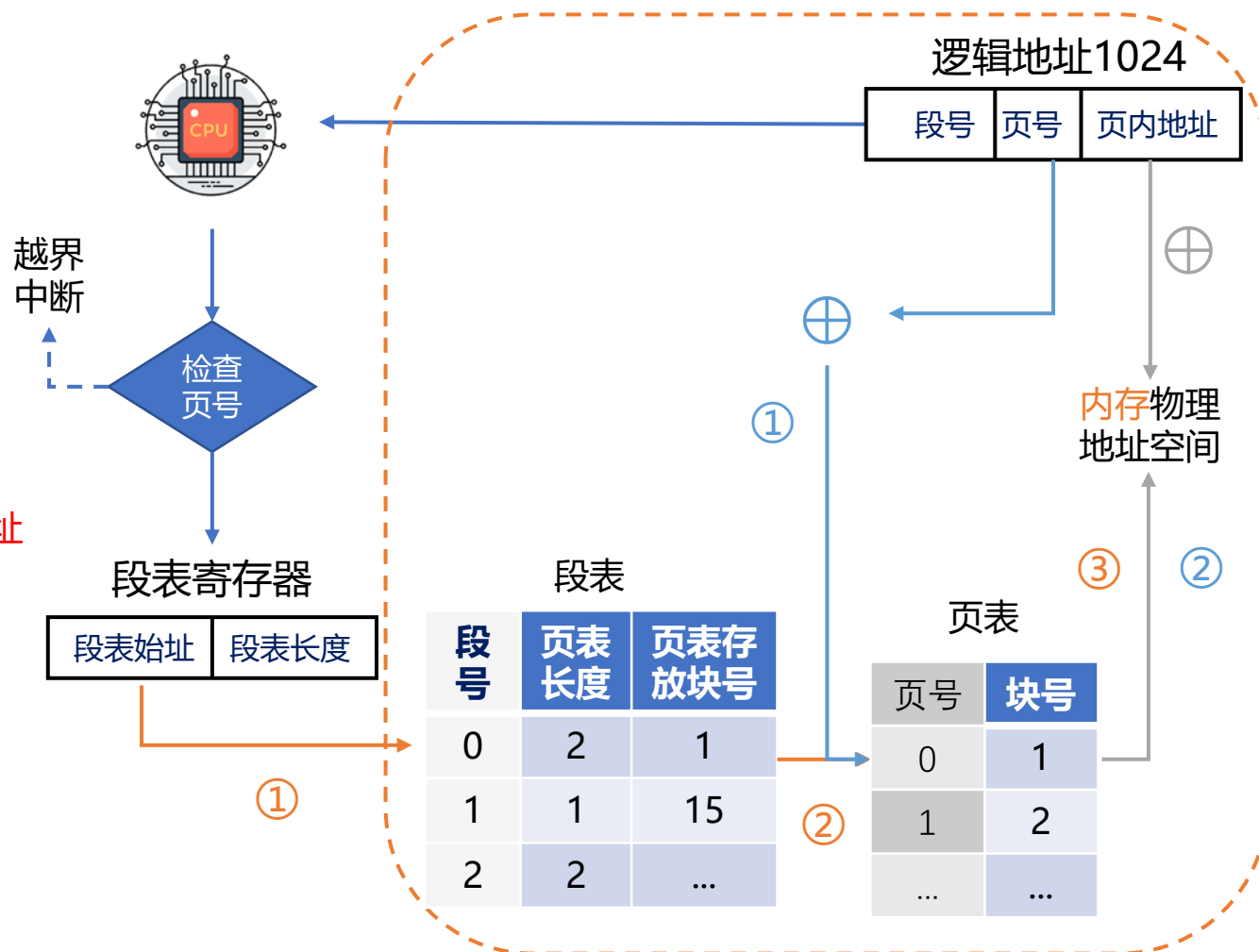
1.什么是内存管理?

• 内存管理方式

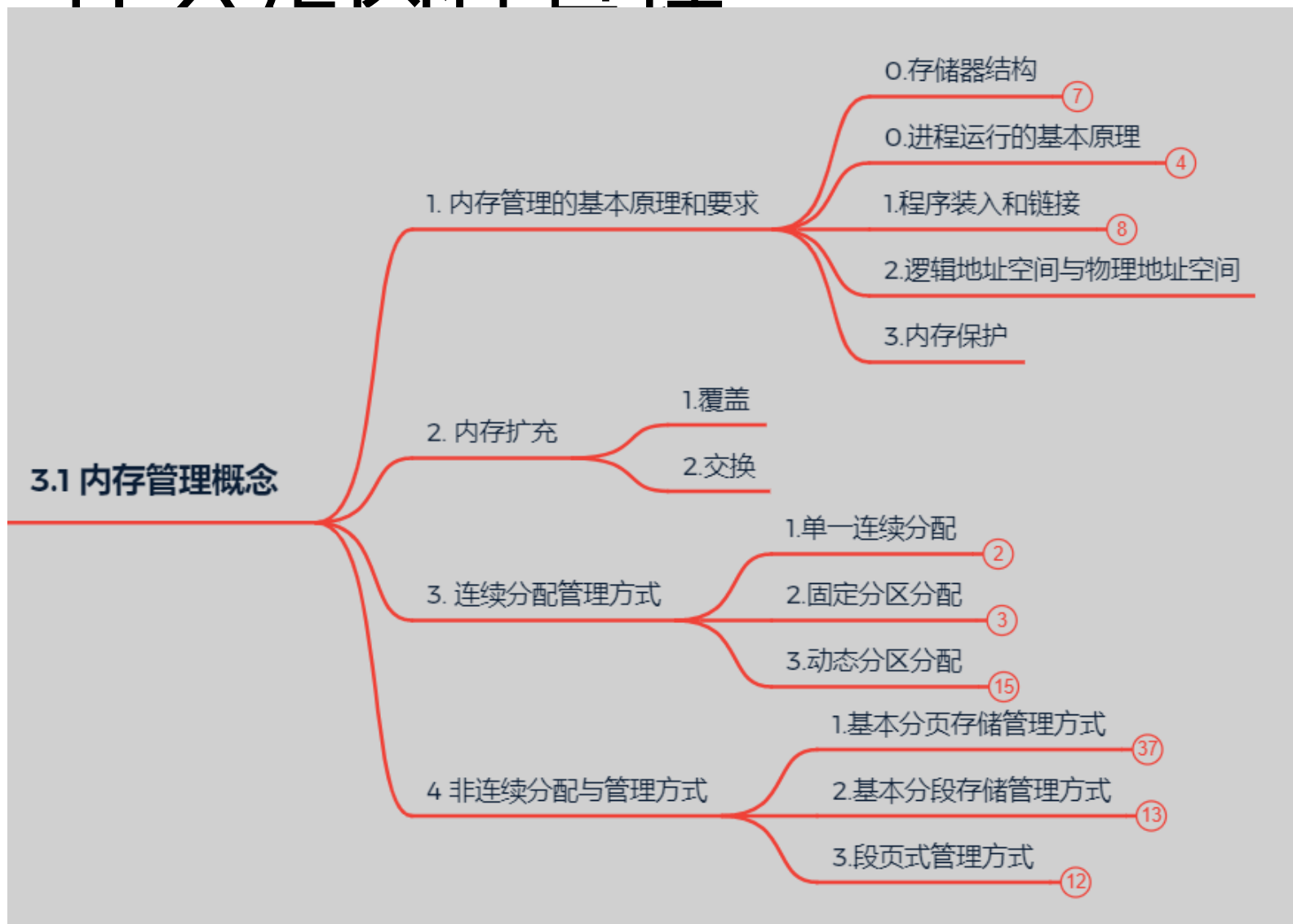
◆ 非连续分配管理方式

◆ 段页式管理方式

- ◆ 段表始址+段号找到段表项
- ◆ 根据页表长度检查页号越界情况
- ◆ 页表地址+页号找到页表项
- ◆ 内存块号+页内地址得到物理地址



小结：什么是内存管理？



目录

1. 什么是内存管理?

2. 虚拟内存管理有什么不同?

2.虚拟内存管理有什么不同？

- ◆ 虚拟内存的基本概念
- ◆ 请求分页管理方式
- ◆ 页面置换算法
- ◆ 页面分配策略

2. 虚拟内存管理有什么不同？

• 虚拟内存的概念

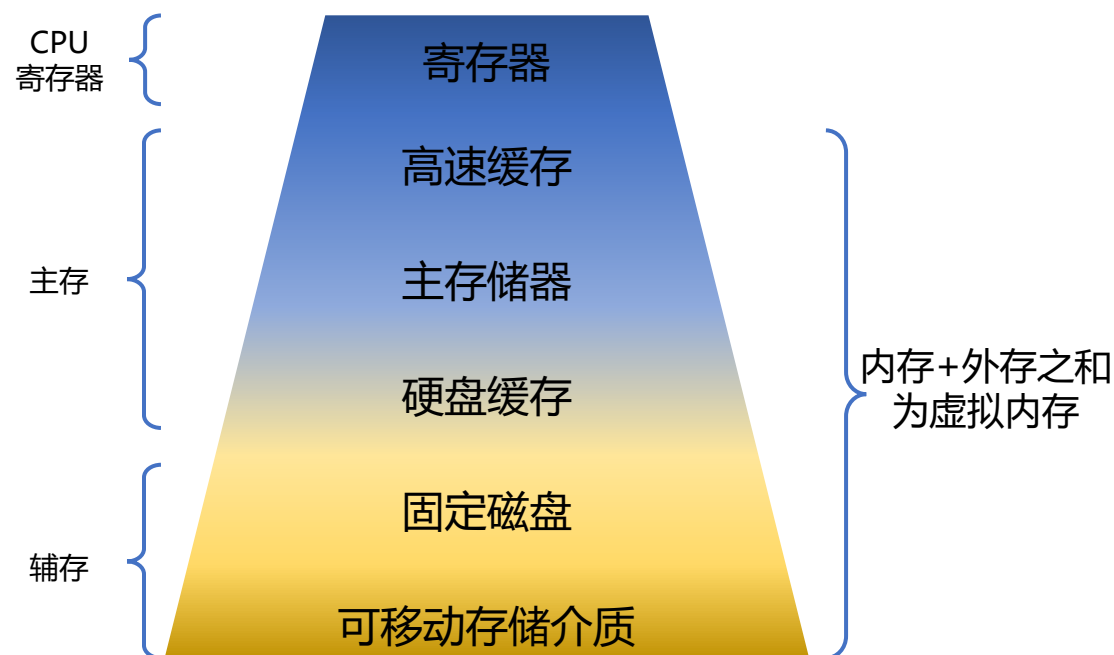
• 具有请求调入和置换功能，从逻辑上对内存容量加以扩充的一种存储器系统

◆ 局部性原理

- ◆ 时间局部性
- ◆ 空间局部性

◆ 虚拟内存的特征

- ◆ 多次性
- ◆ 对换性
- ◆ 虚拟性



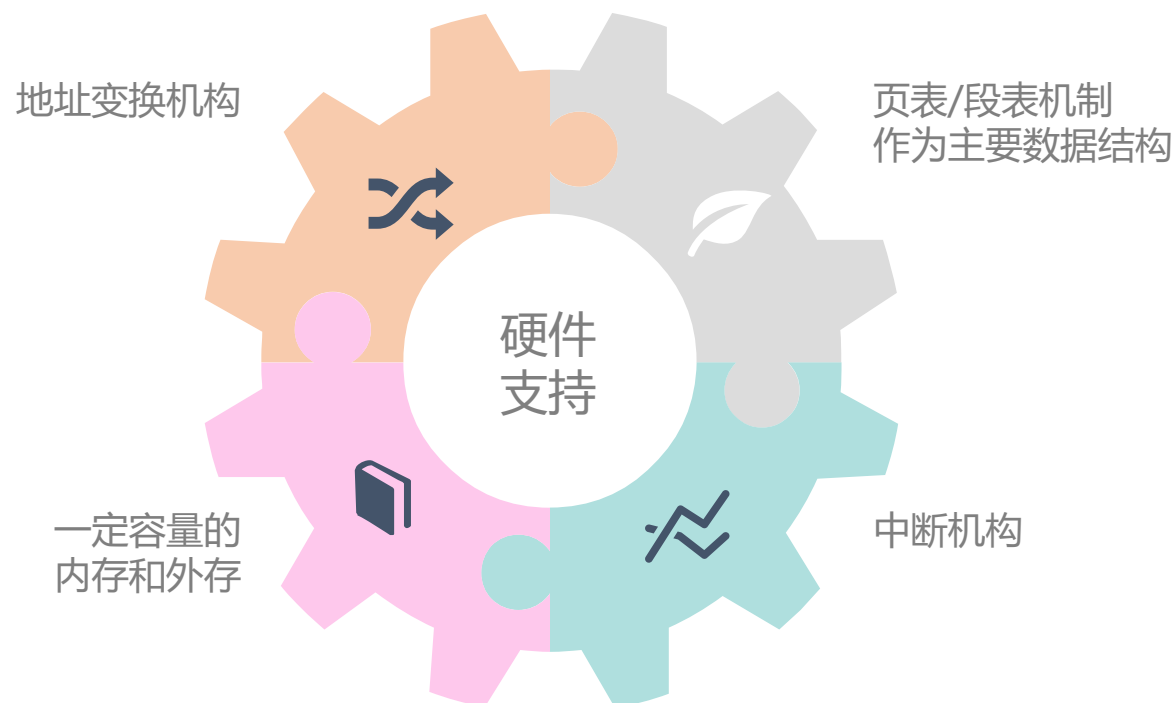
2. 虚拟内存管理有什么不同？

- 虚拟内存的概念

- 具有请求调入和置换功能，从逻辑上对内存容量加以扩充的一种存储器系统

- ◆ 虚拟内存的实现

- ◆ 请求分页存储管理
- ◆ 请求分段存储管理
- ◆ 请求段页式存储管理



2. 虚拟内存管理有什么不同?

• 请求分页管理方式

◆ 页表机制

- ◆ 状态位P
- ◆ 访问字段A
- ◆ 修改位M
- ◆ 外存地址

◆ 缺页中断机构

◆ 地址变换机构

基本分页
存储管理页表

页号	块号
0	a
1	b
2	c

请求分页
存储管理页表

页号	内存块号	状态位	访问字段	修改位	外存地址
0	无	0	0	0	x
1	b	1	13	1	y
2	c	1	8	1	z

2. 虚拟内存管理有什么不同？

• 请求分页管理方式

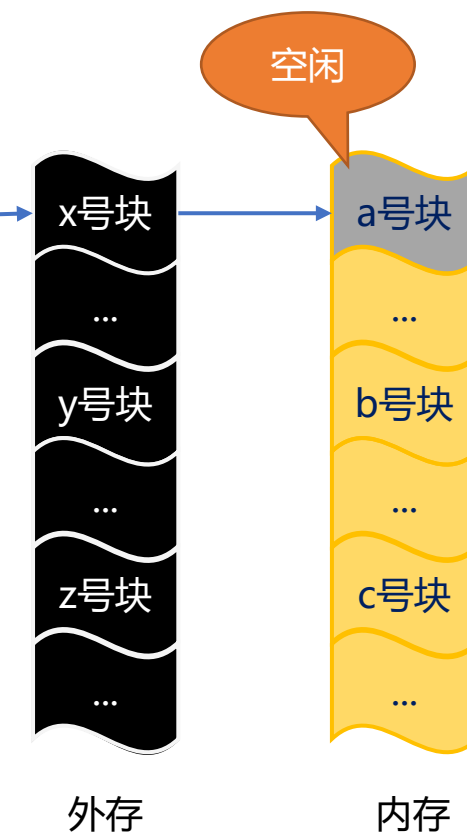
◆ 页表机制

◆ 缺页中断机构

◆ 地址变换机构

请求分页
存储管理页表

页号	内存块号	状态位	访问字段	修改位	外存地址
0	无	0	0	0	x
1	b	1	13	1	y
2	c	1	8	1	z



2. 虚拟内存管理有什么不同?

• 请求分页管理方式

◆ 页表机制

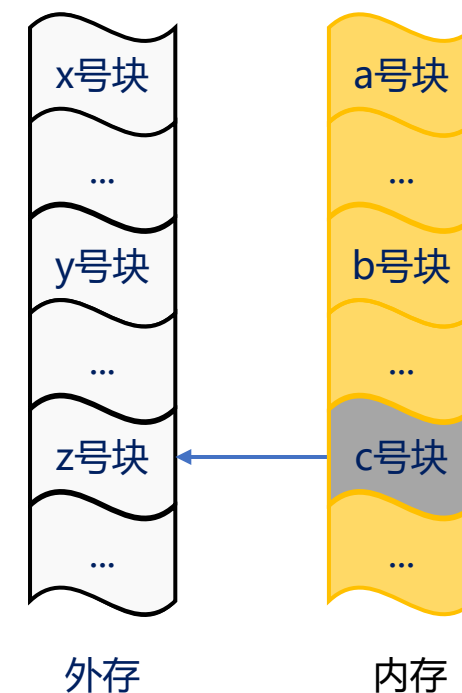
◆ 缺页中断机构

◆ 地址变换机构

请求分页
存储管理页表

页号	内存块号	状态位	访问字段	修改位	外存地址
0	无	0	0	0	x
1	b	1	13	1	y
2	c	1	8	1	z

页面置换算法
选择淘汰



2. 虚拟内存管理有什么不同?

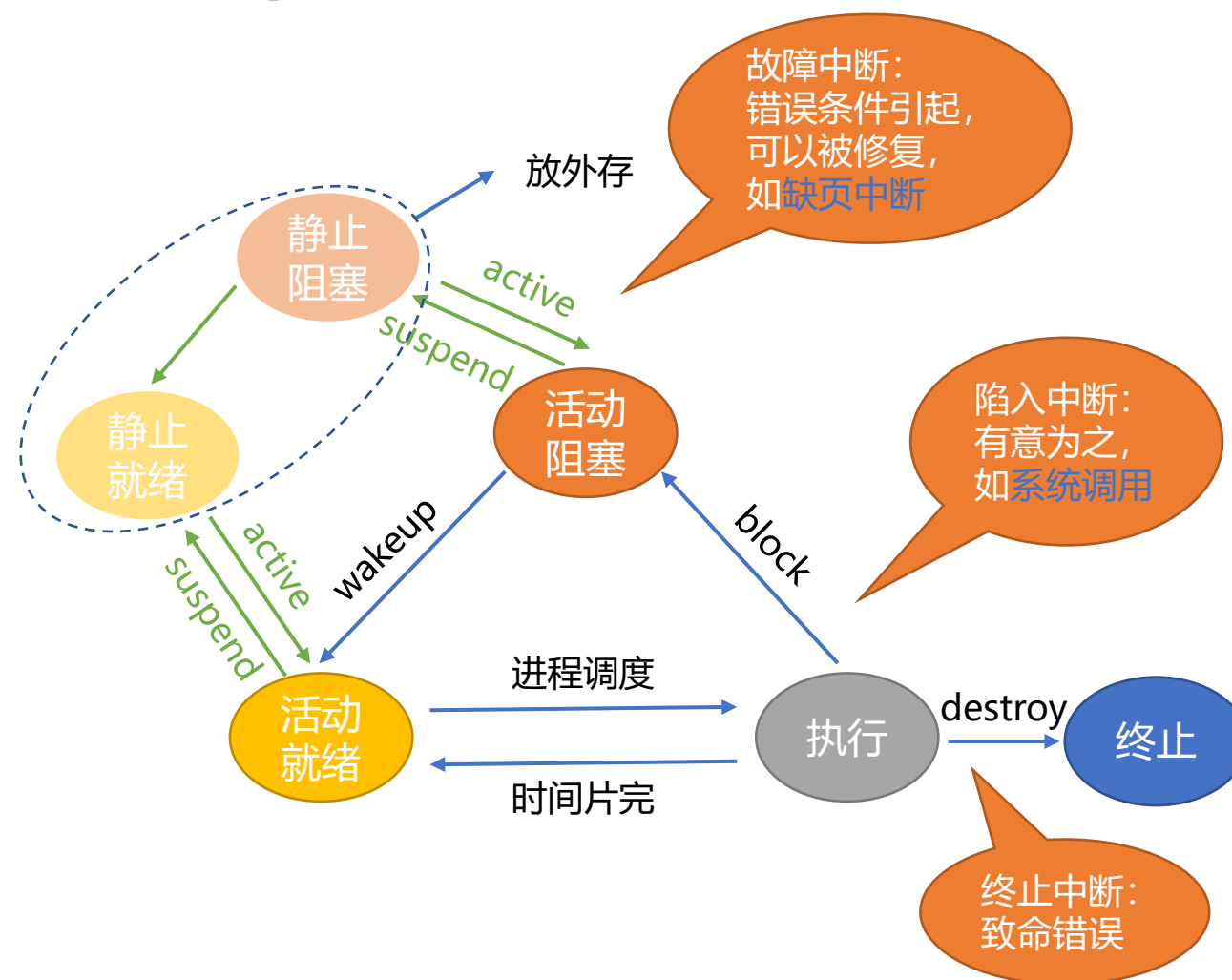
• 请求分页管理方式

◆ 页表机制

◆ 缺页中断机构

- ◆ 内中断(CPU内部)
 - ◆ 陷入、故障、终止
- ◆ 外中断(CPU外部)
 - ◆ I/O中断请求
 - ◆ 人工干预

◆ 地址变换机构



2. 虚拟内存管理有什么不同？

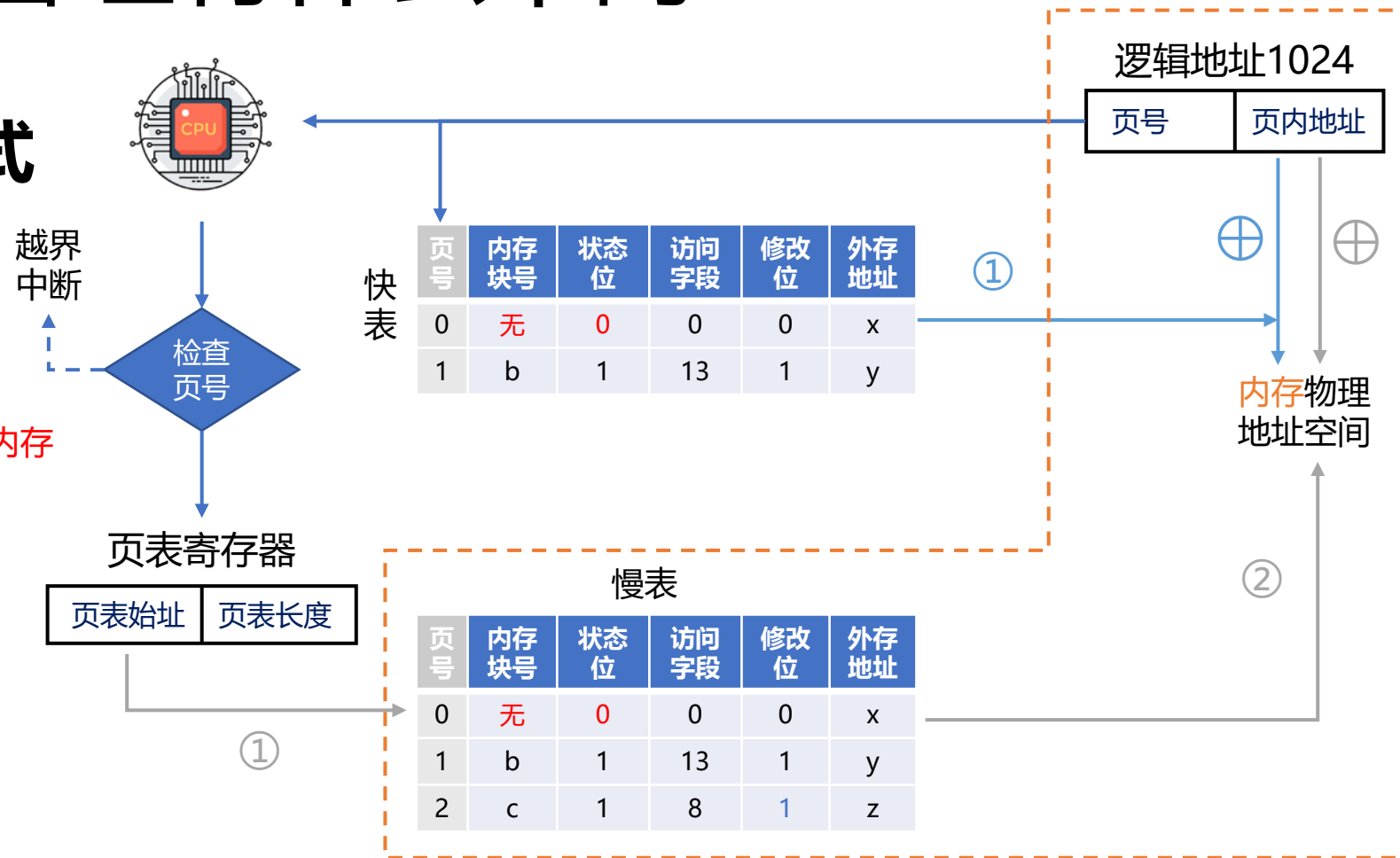
• 请求分页管理方式

◆ 页表机制

◆ 缺页中断机构

◆ 地址变换机构

- ◆ 请求调页，判断是否在内存
- ◆ 可能需要页面置换
- ◆ 新增/修改页表项
- ◆ 热点表项同步到快表



2.虚拟内存管理有什么不同？

• 页面置换算法

保障顺序上的公平：
每次选择淘汰最早进入内存的页面
Belady异常，性能差



先进先出置换算法FIFO



最佳置换算法OPT

保障最低缺页率：
每次选择淘汰最不可能再次被使用的页面
无法实现

页号	内存块号	状态位	访问字段	修改位	外存地址
----	------	-----	------	-----	------

访问位：
0未访问；
1已访问

保障性能和开销均衡：
为页面设置访问位(0/1)，并链接成循环队列，进程访问页面后置为1。淘汰时为1置为0并跳过，为0时淘汰。
最多需要两轮扫描



时钟置换算法NRU



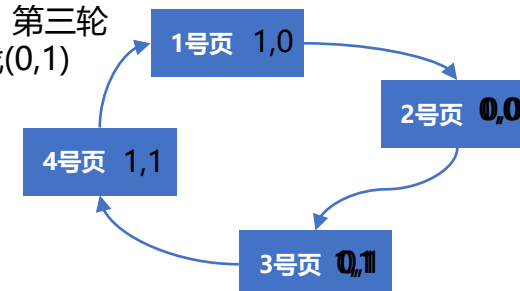
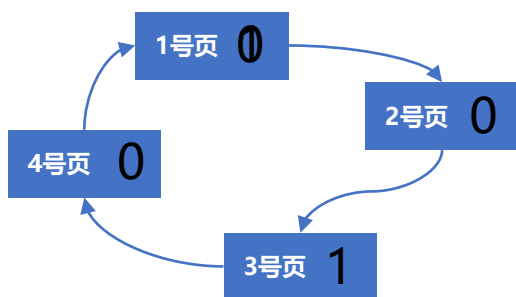
最近最久置换算法LRU

保障时间和距离上的公平：
每次选择淘汰最久最近未使用的页面
需要硬件支持，开销大



改进型时钟置换算法

额外考虑是否修改，保障最少I/O操作：
增加修改位(0/1)，第一轮找(0,0)，第二轮找(0,1)并修改访问位为0，第三轮找(0,0)，第四轮找(0,1)



2.虚拟内存管理有什么不同？

• 页面分配策略

◆ 驻留集（驻留在主存中页面数）大小

分配空间小，进程数量多，CPU时间利用效率就高

进程在主存中页数少，错页率就高

进程在主存页数多，错页率并无明显改善

◆ 页面分配策略

固定分配局部置换

可变分配全局置换

可变分配局部置换

	局部置换	全局置换
固定分配	✓	-
可变分配	✓	✓

2. 虚拟内存管理有什么不同?

• 页面分配策略

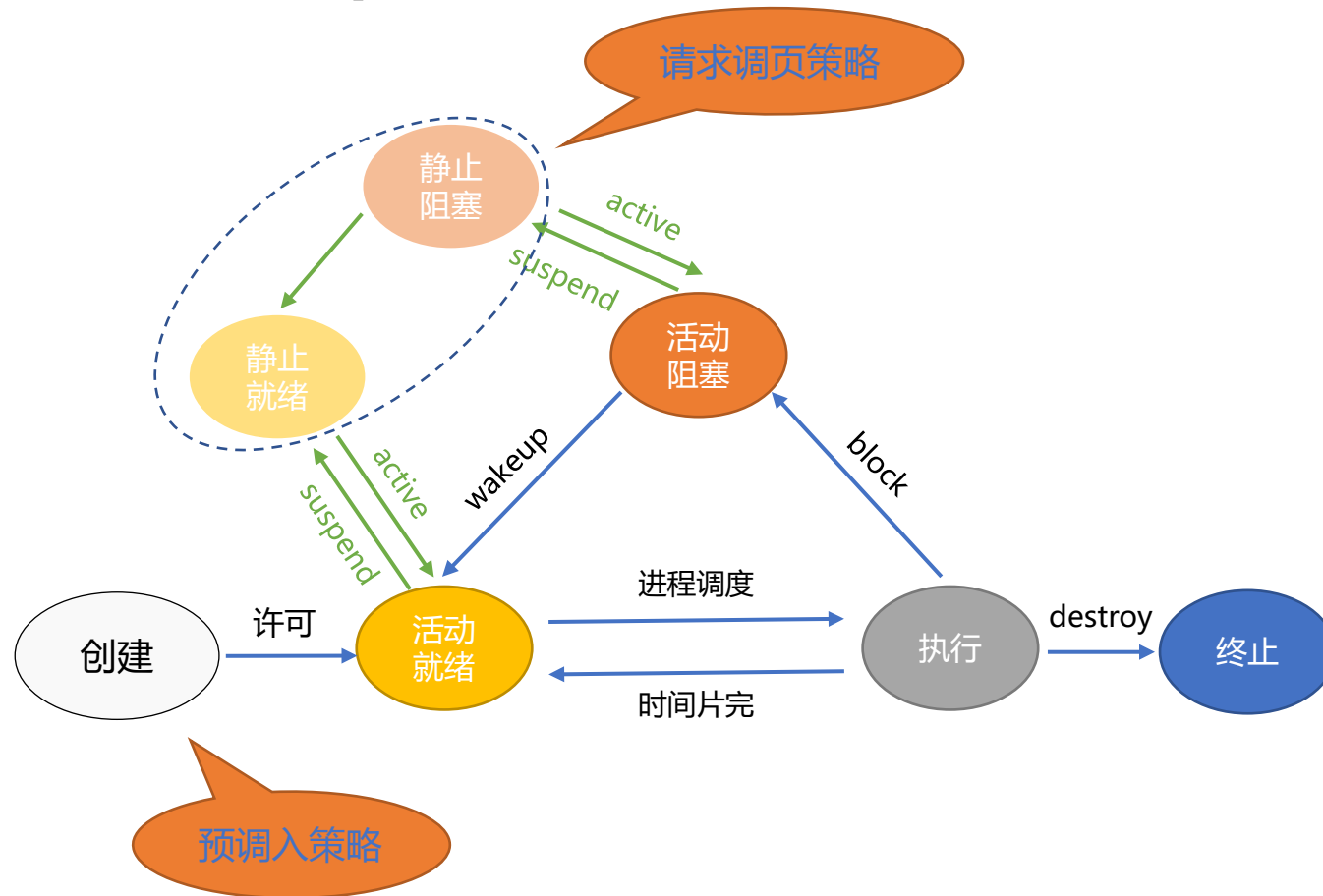
◆ 调入页面的时机

◆ 预调页策略

- ◆ 一次性调入若干相邻页面
- ◆ 多用于进程首次调入

◆ 请求调页策略

- ◆ 运行时发现缺页时调入
- ◆ I/O开销较大

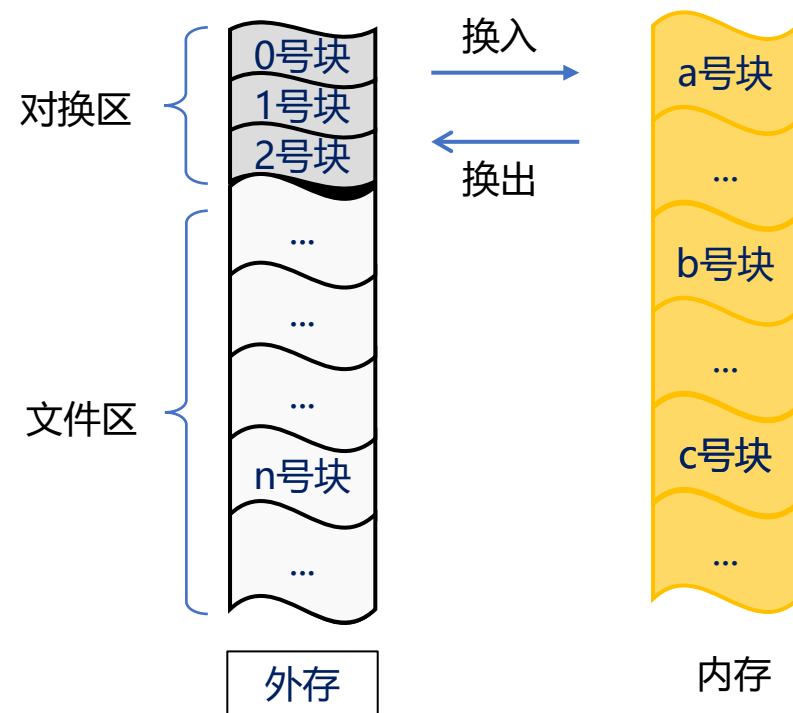


2. 虚拟内存管理有什么不同?

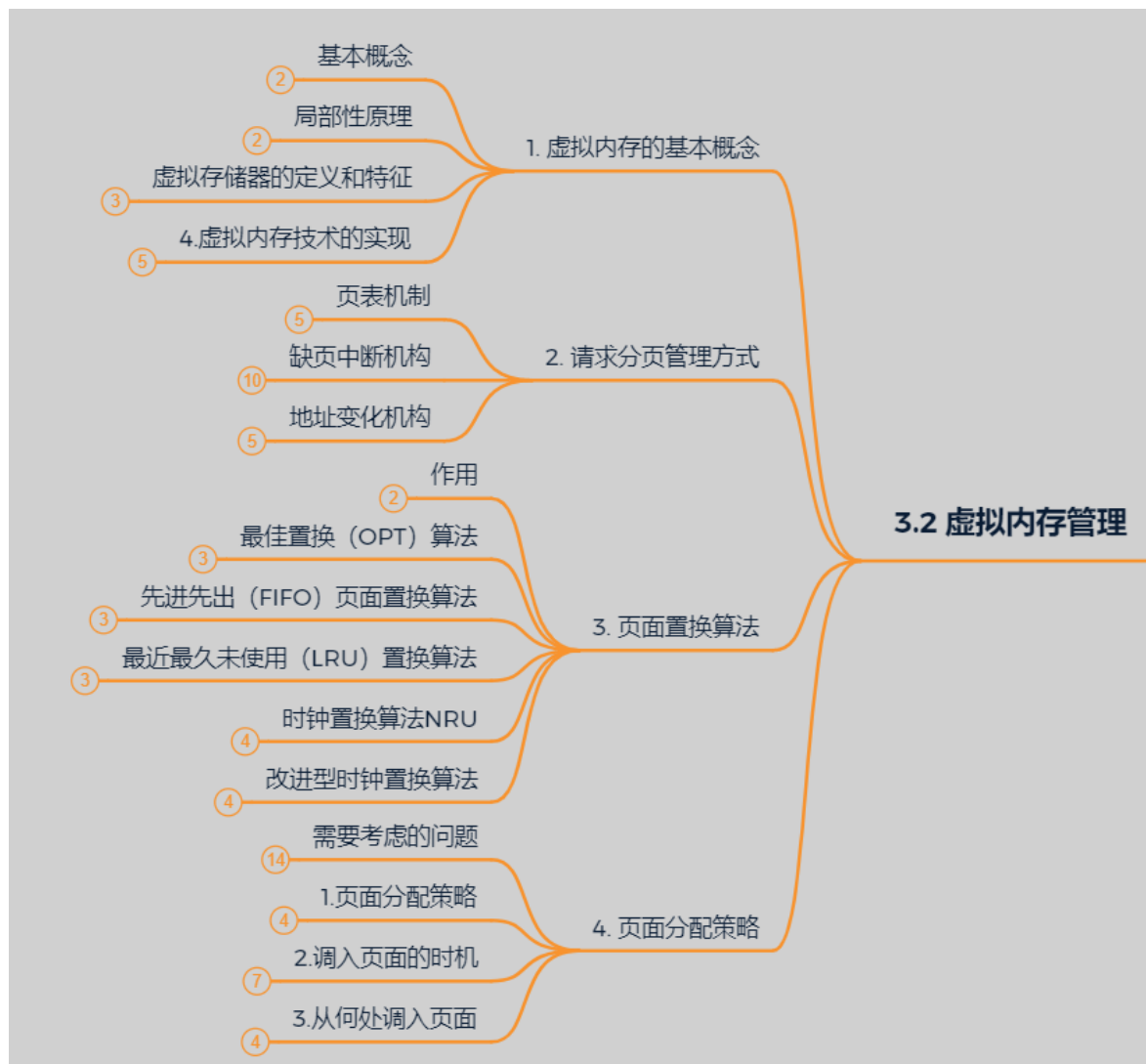
• 页面分配策略

◆ 从何处调页

系统拥有足够的对换区空间
系统缺少足够的对换区空间
UNIX方式



小结：虚拟内存管理有什么不同？





马士兵教育

www.mashibing.com



扫码加马老师微信