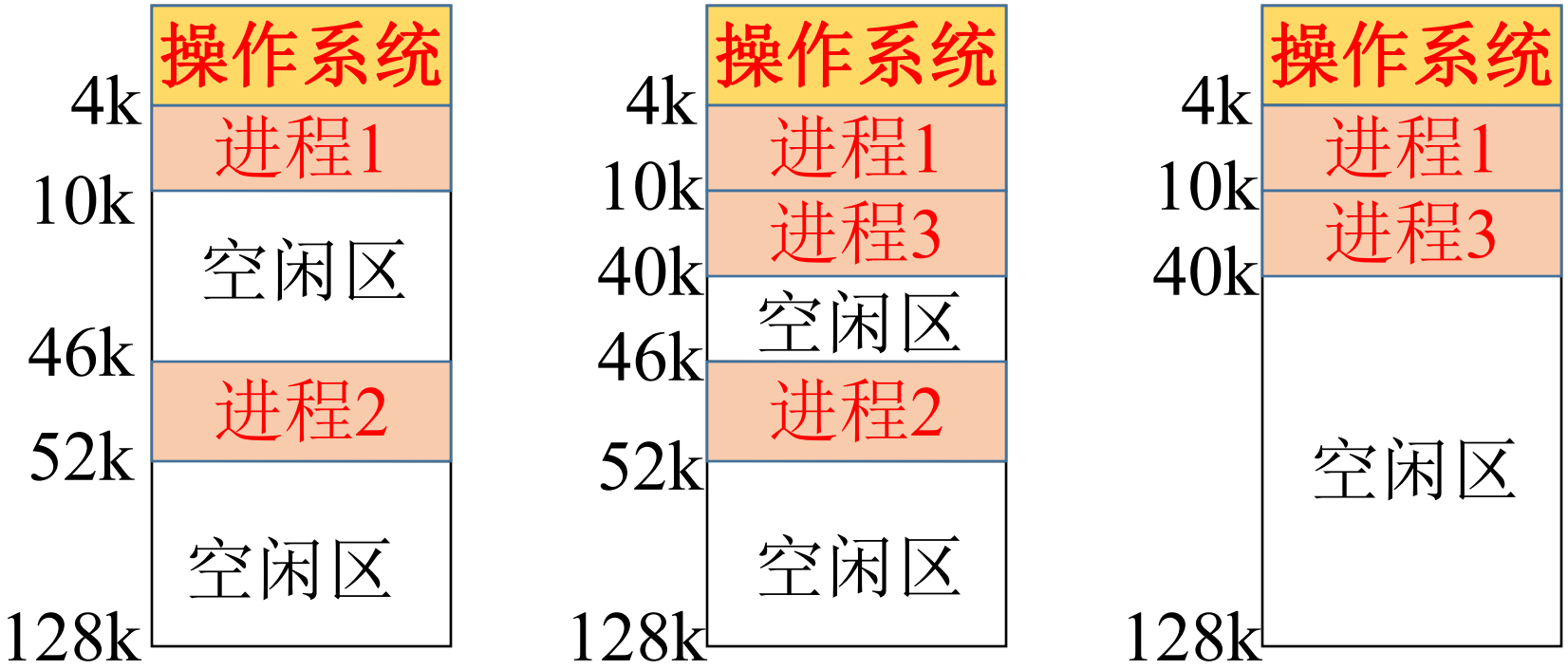
可变分区存储管理

- 按进程的内存需求来动态划分分区
- 创建一个进程时，根据进程所需主存量查看主存中是否有足够的空闲空间
  - 若有，则按需要量分割一个分区
  - 若无，则令该进程等待主存资源
- 由于分区大小按照进程实际需要量来确定，因此分区个数是随机变化的

可变分区方式的内存分配示例



可变分区方式的主存分配表

- 已分配区表与未分配区表，采用链表

起址	长度	标志
4k	6k	J1
46k	6k	J2
		空
		空
⋮	⋮	⋮

(a)已分配区情况表

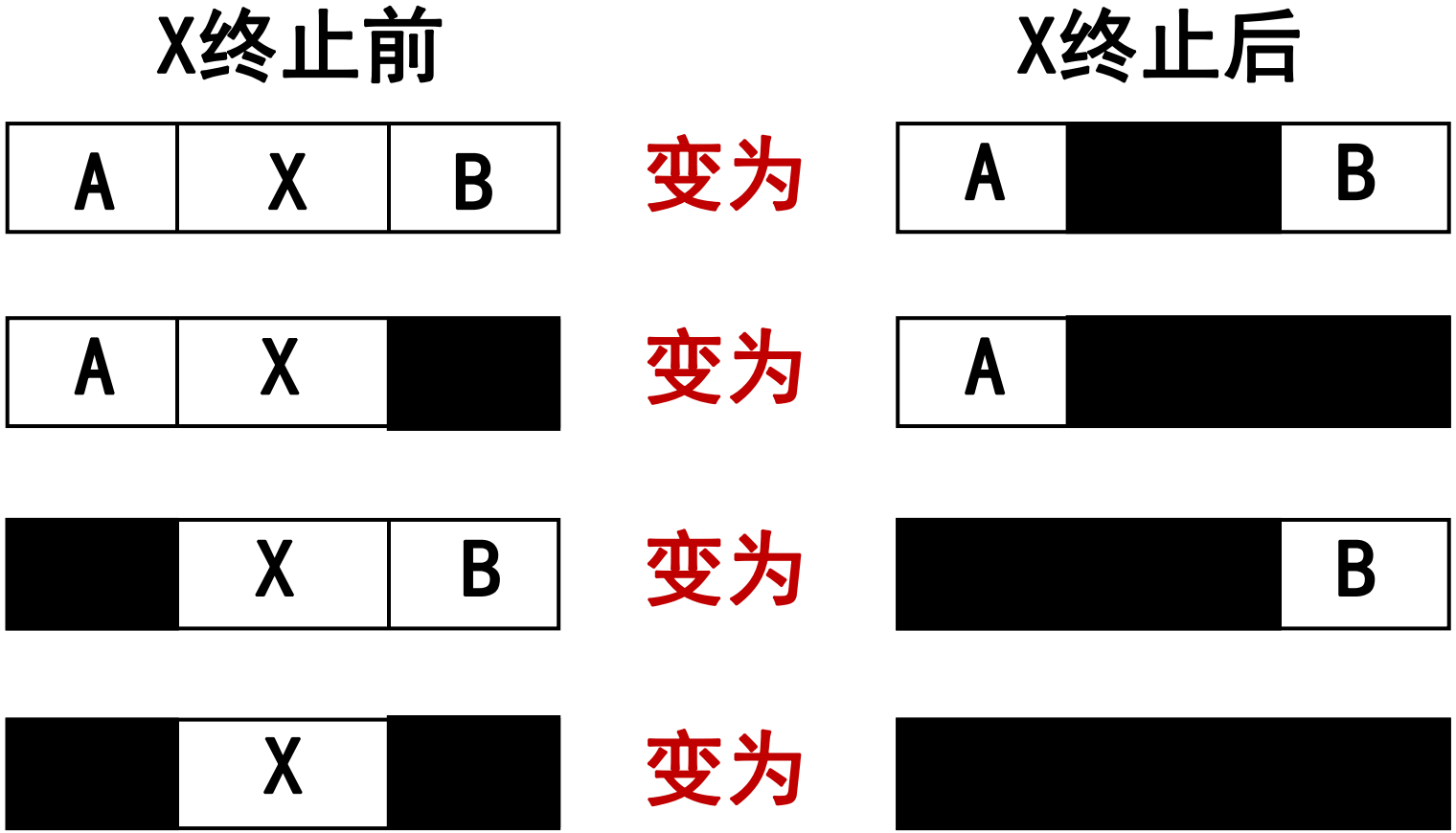
起址	长度	标志
10k	36k	未分配
52k	76k	未分配
		空
		空
⋮	⋮	⋮

(b)未分配区情况表

可变分区方式的内存分配

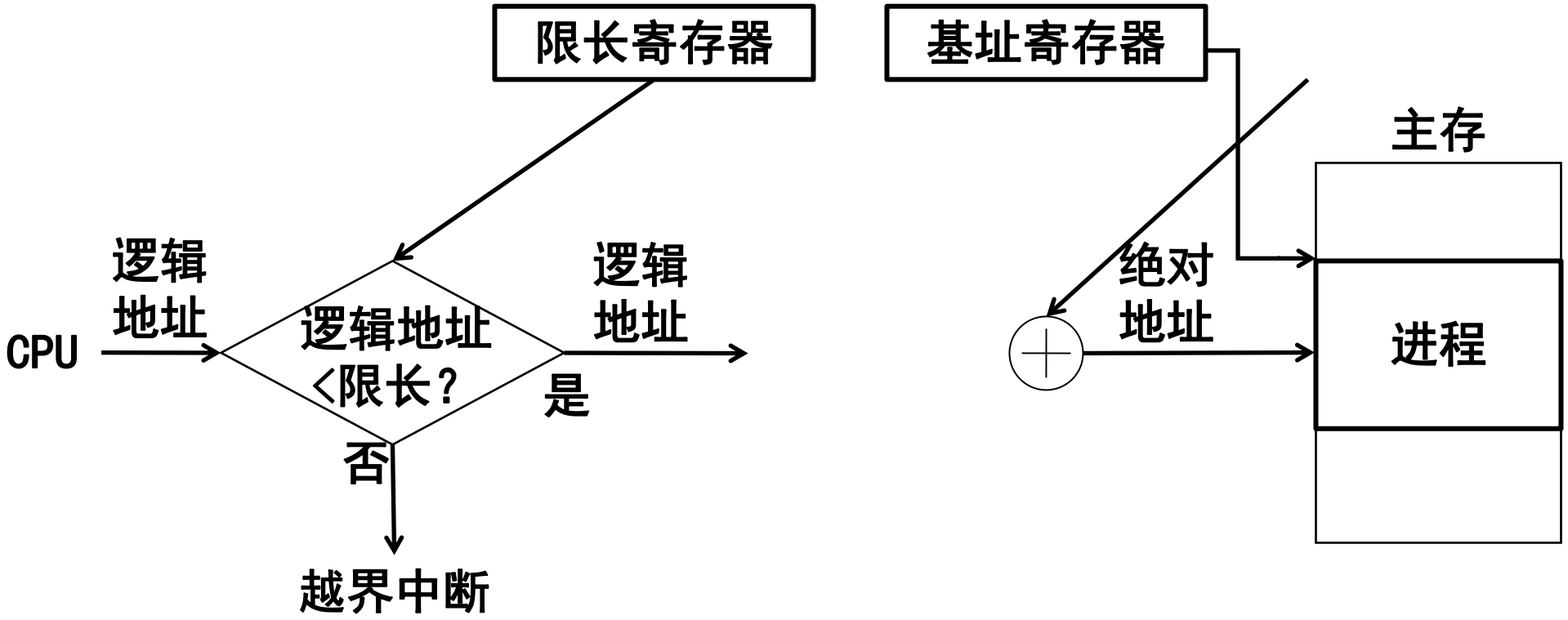
- 最先适应分配算法
- 邻近适应分配算法
- 最优适应分配算法
- 最坏适应分配算法

可变分区方式的内存回收



地址转换与存储保护

- 硬件实现机制与动态重定位

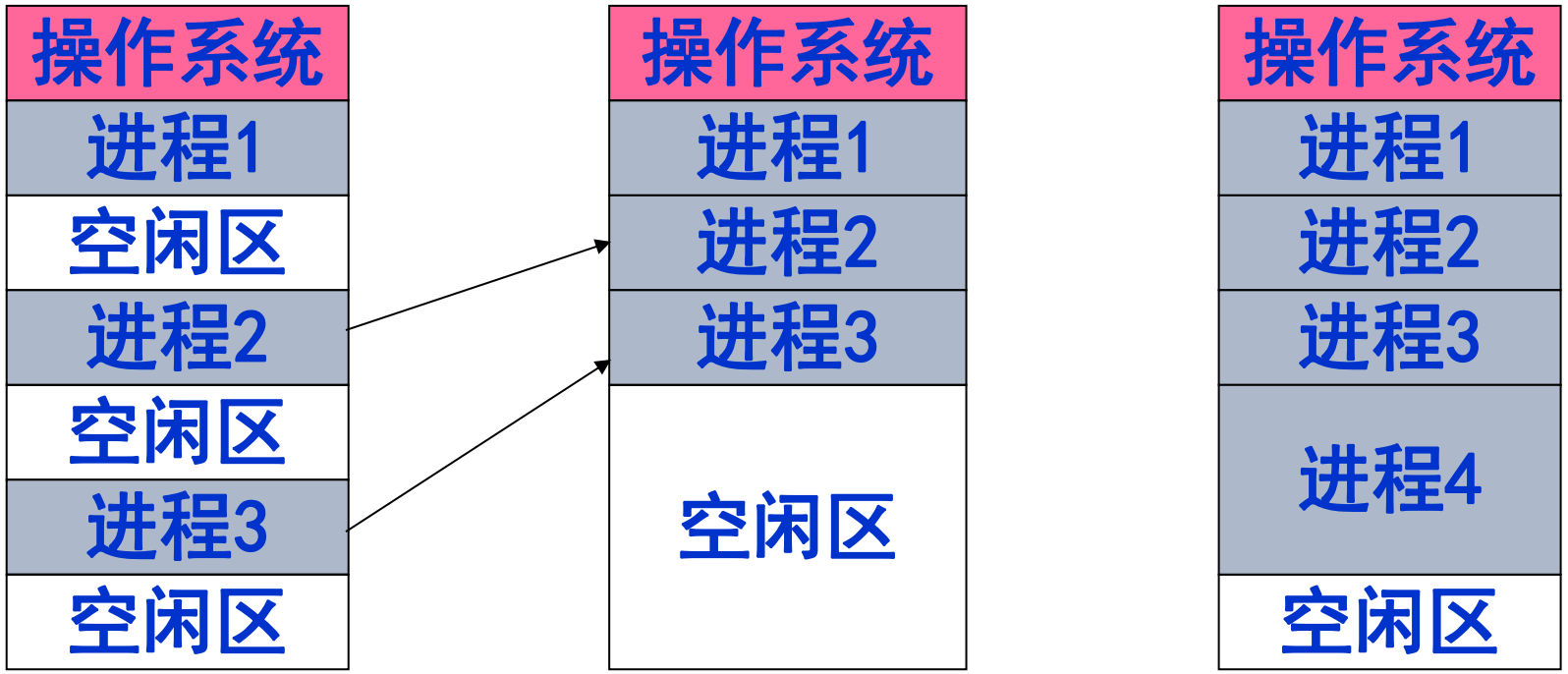


可变分区方式的内存零头

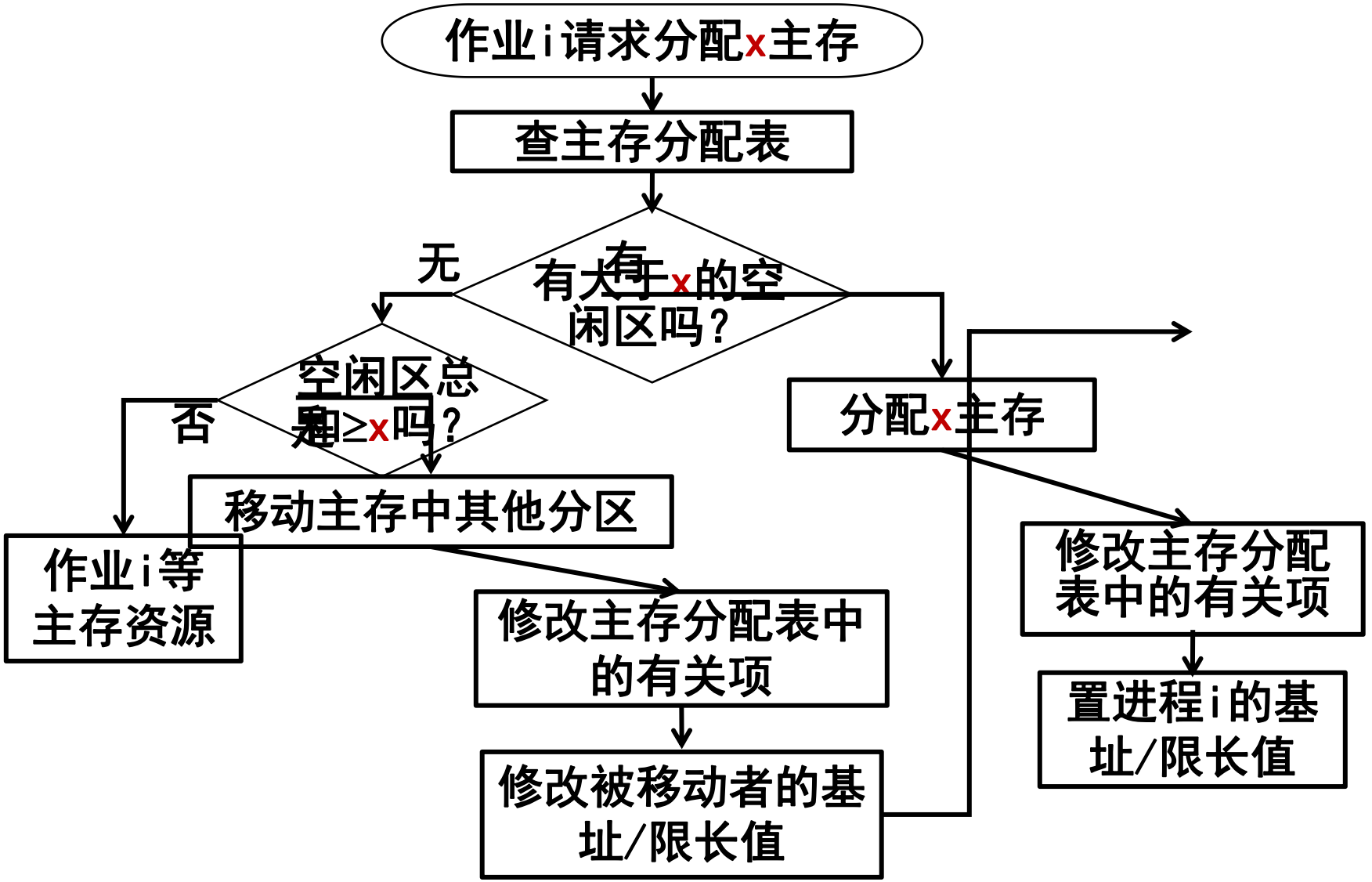
- 固定分区方式会产生**内存内零头**
- 可变分区方式也会随着进程的内存分配产生一些小的不可用的内存分区，称为**内存外零头**
- 最优适配算法最容易产生外零头
- 任何适配算法都**不能避免**产生外零头

移动技术（程序浮动技术）

- 移动分区以解决内存外零头
- 需要动态重定位支撑



移动技术的工作流程


作业i请求分配x主存
查主存分配表
有大于x的空闲区吗？
有
无
空闲区总和≥x吗？
是
否
分配x主存
移动主存中其他分区
作业i等主存资源
修改主存分配表中的有关项
修改主存分配表中的有关项
置进程i的基址/限长值
修改被移动者的基址/限长值