

第7章 主存管理

- 主存管理概述
- 主存管理的功能
- 分区存储管理
- 页式存储管理
- 段式及段页式存储管理



主存管理概述

引言

- 现代操作系统将主存区分为物理主存和逻辑主存2类。
- 物理主存是共享的物质基础。

1. 主存共享方式——分片共享

(1) 大小不等的区域

- ① 分区存储管理
- ② 段式存储管理

(2) 大小相等的区域

- 页式存储管理

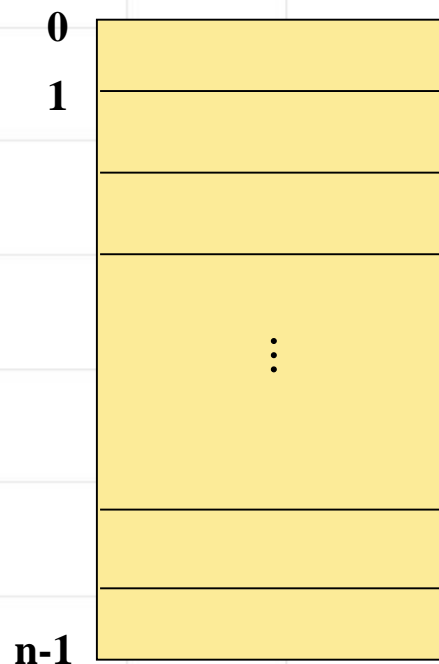
(3) 二者结合

- 段页式存储管理

2. 程序的逻辑组织

(1) 一维地址结构

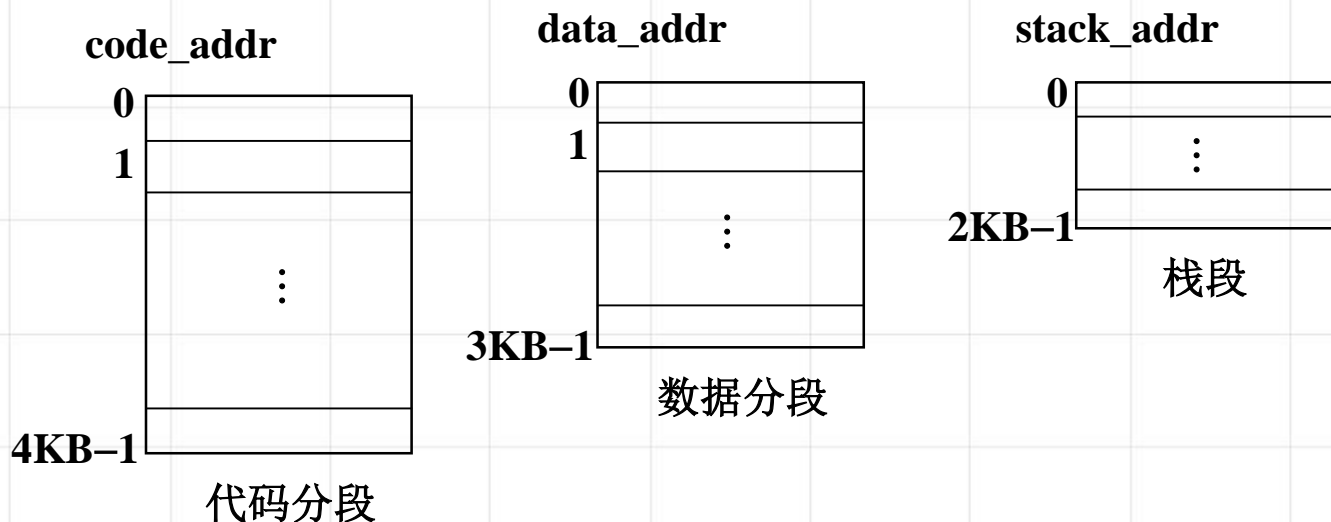
- 一个程序是一个连续、线性的地址结构；
- 确定线性地址空间中的指令地址或操作数地址只需要一个信息。



程序地址空间
一维地址结构

(2) 二维地址结构

- 一个程序由若干个分段组成，每个分段是一个连续的地址区；
- 确定线性地址空间中的指令地址或操作数地址需要两个信息，一是该信息所在的分段，另一个是该信息在段内的偏移量。



程序地址空间——二维地址结构



主存管理功能

1. 几个概念

(1) 物理地址 (绝对地址、实地址)

物理地址是计算机主存单元的真实地址，又称为绝对地址或实地址。

(2) 主存空间

物理地址的集合所对应的空间组成了主存空间。

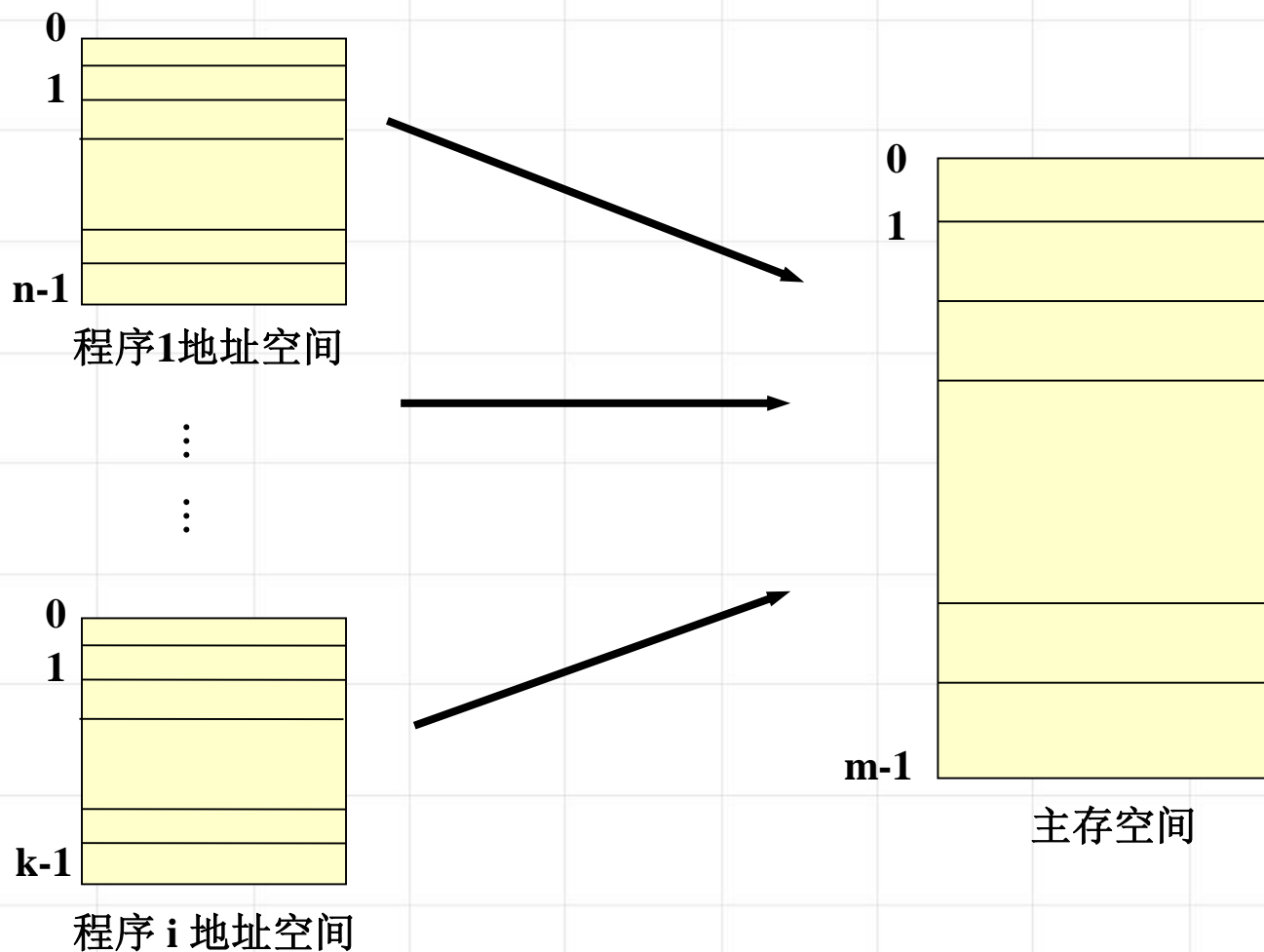
(3) 逻辑地址 (相对地址、虚地址)

用户的程序地址 (指令地址或操作数地址)均为逻辑地址。

(4) 程序地址空间

用户程序所有的逻辑地址集合对应的空间。

(5) 程序地址空间与主存空间



程序地址空间与主存空间示意图

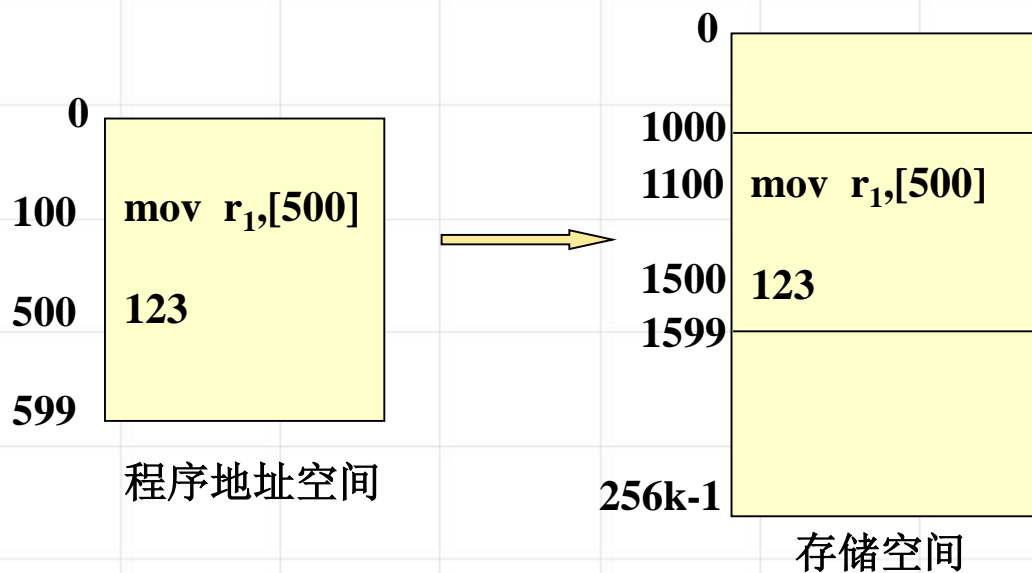
2. 主存管理功能

- 地址映射——实现逻辑地址到物理主存地址的映射
- 主存分配——在多用户之间分配物理主存
- 存储保护——对各用户区的信息提供保护措施
- 主存扩充/虚拟主存——扩充逻辑主存区

3. 地址映射

(1) 什么是地址映射

将程序地址空间中使用的逻辑地址变换成主存中的物理地址的过程，称为地址映射。



程序地址空间装入主存

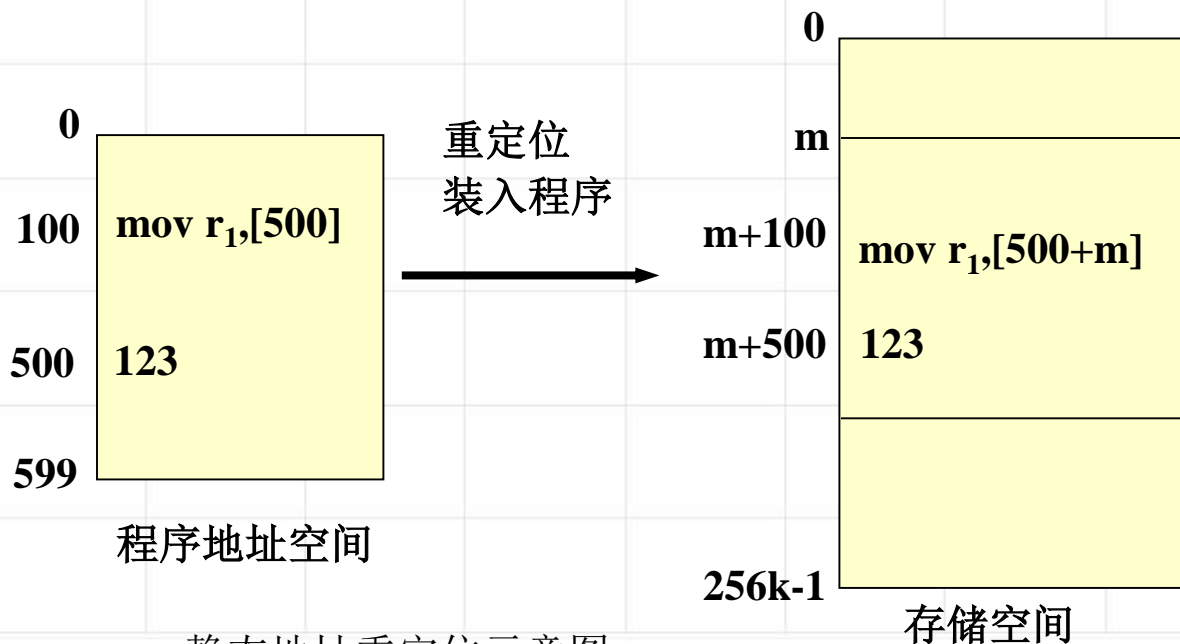
(2) 地址映射的时机和类别

① 编程或编译时确定地址映射关系

在程序编写或程序编译时确定虚、实地址之间的对应关系，结果是一个不能浮动的程序模块。

② 在程序装入时确定地址映射关系

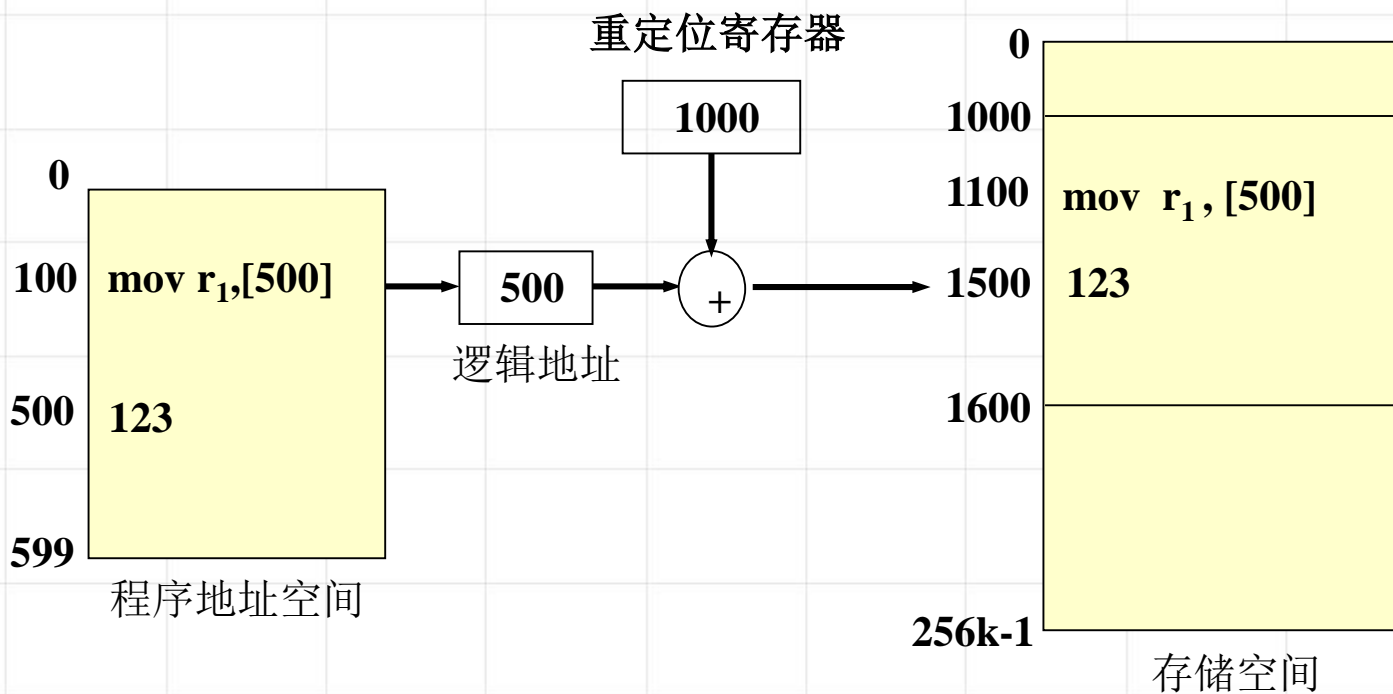
在程序装入过程中随即进行的地址变换方式称为静态地址映射。



静态地址重定位示意图

③ 在程序运行时确定地址映射关系

在程序执行期间，随着每条指令和数据的访问自动地连续地进行地址映射，这种地址变换方式称为**动态地址映射**。



动态地址重定位示意图

④ 静态地址映射与动态地址映射的区别

静态地址映射	动态地址映射
在程序装入过程中进行地址映射	在程序执行期间进行地址映射
需软件 (重定位装入程序)	需硬件地址变换机构 (重定位寄存器)
需花费较多CPU时间	地址变换快
不灵活	灵活

4. 主存分配

(1) 构造分配用的数据结构

主存资源信息块：等待队列；空闲区队列；主存分配程序

(2) 制定策略

- ① 分配策略 —— 在众多请求者中选择一个请求者的原则
- ② 放置策略 —— 在可用资源中，选择一个空闲区的原则
- ③ 调入策略 —— 决定信息装入主存的时机

预调策略：预先将信息调入主存

请调策略：当需要信息时，将信息调入主存

- ④ 淘汰策略 —— 在主存中没有可用的空闲区 (对某一程序而言)时，决定哪些信息从主存中移走，即确定淘汰已占用的内存区的原则。

(3) 实施主存分配与回收

5. 主存扩充

(1) 可行性 局部性特征

(2) 实现方法

- 程序的全部代码和数据存放在辅存中；
- 将程序当前执行所涉及的那部分程序代码放入主存中；
- 程序执行时，当所需信息不在主存，由操作系统和硬件相配合来完成主存从辅存中调入信息，程序继续执行。

(3) 虚拟存储器

① 什么是虚拟存储器

由操作系统和硬件相配合来完成主存和辅存之间的信息的动态调度。
这样的计算机系统好像为用户提供了一个其存储容量比实际主存大得多的存储器，这个存储器称为虚拟存储器。

② 虚拟存储器的核心

- 逻辑地址与物理地址分开
- 存储空间与虚地址空间分开
- 提供地址变换机构

③ 实现虚拟存储器的物质基础

- 有相当容量的辅存： 足以存放应用程序的虚地址空间
- 有一定容量的主存： 存放进入主存的多进程的信息
- 地址变换机构

6. 存储保护

(1) 什么是存储保护

在多用户环境中，主存储器按区分配给各用户程序使用。

为了互不影响，必须由硬件 (软件配合) 保证各用户程序只能在给定的存储区域内活动，这种措施叫做存储保护。

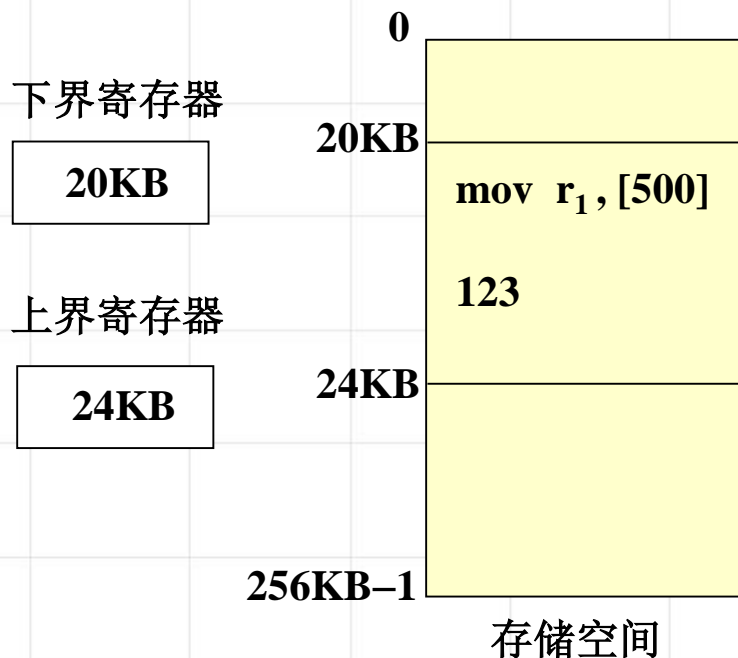
(2) 实现方法

- 界地址保护
- 存储键保护

界地址保护

① 上下界防护

例：程序大小为4KB，主存首址为20KB。



界限寄存器保护示意图

- 如何设置上下界寄存器内容？

- 如何判断是否越界？

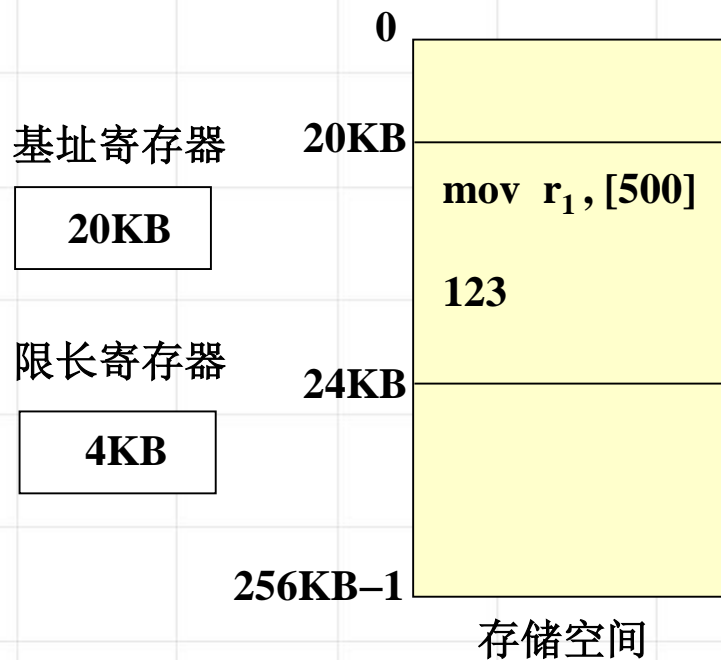
若 $20\text{KB} \leq D < 24\text{KB}$

允许访问；

否则发生越界中断

② 基址、限长防护

例：程序大小为4KB，主存首址为20KB。



界限寄存器保护示意图

- 如何设置基址、限长寄存器内容？

- 如何判断是否越界？

若 逻辑地址 $< 4\text{KB}$

允许访问；

否则发生越界中断



分区存储管理

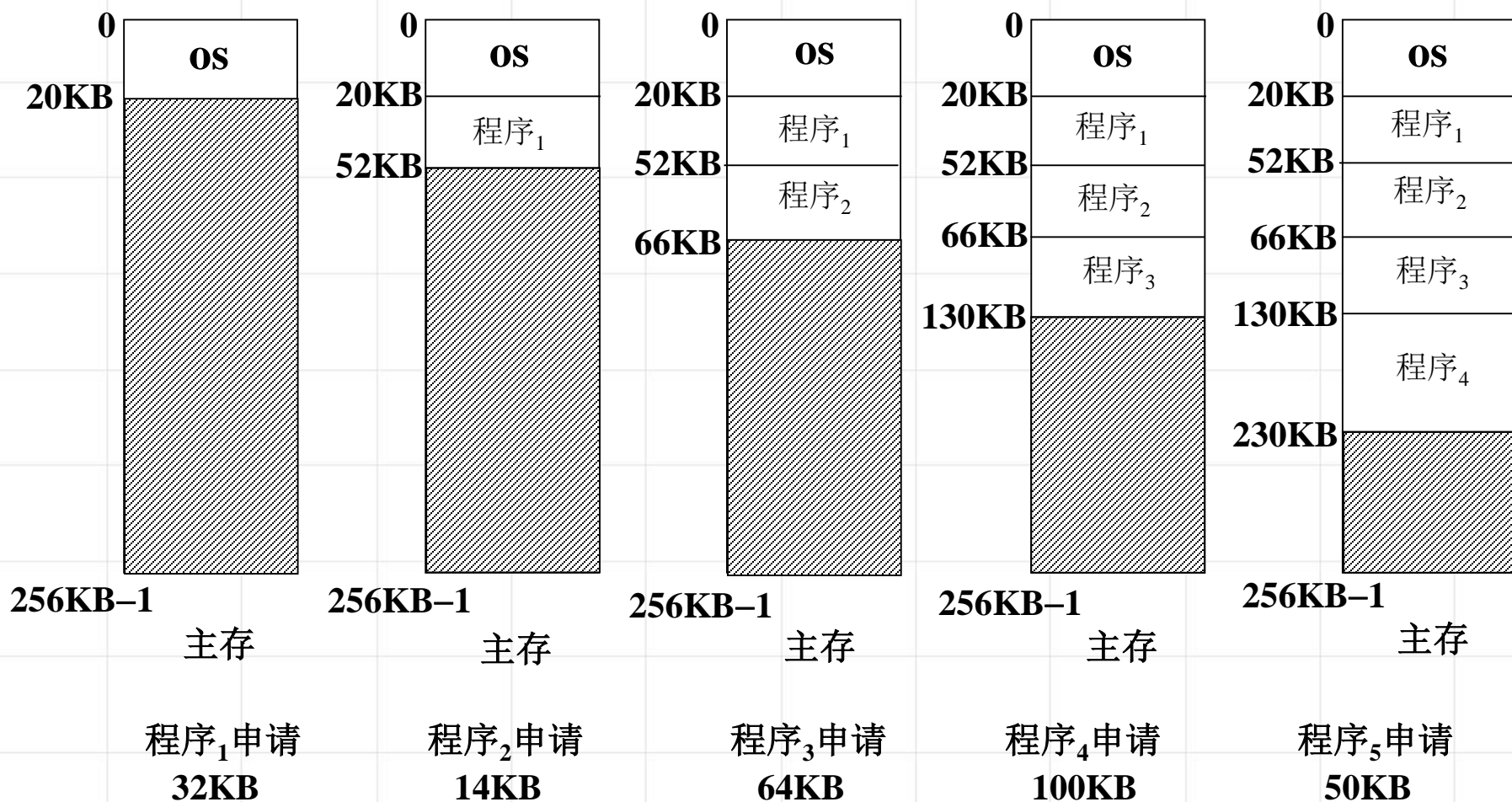
1. 动态分区分配

(1) 什么是动态分区分配

在处理程序的过程中，建立分区，依用户请求的大小分配分区。

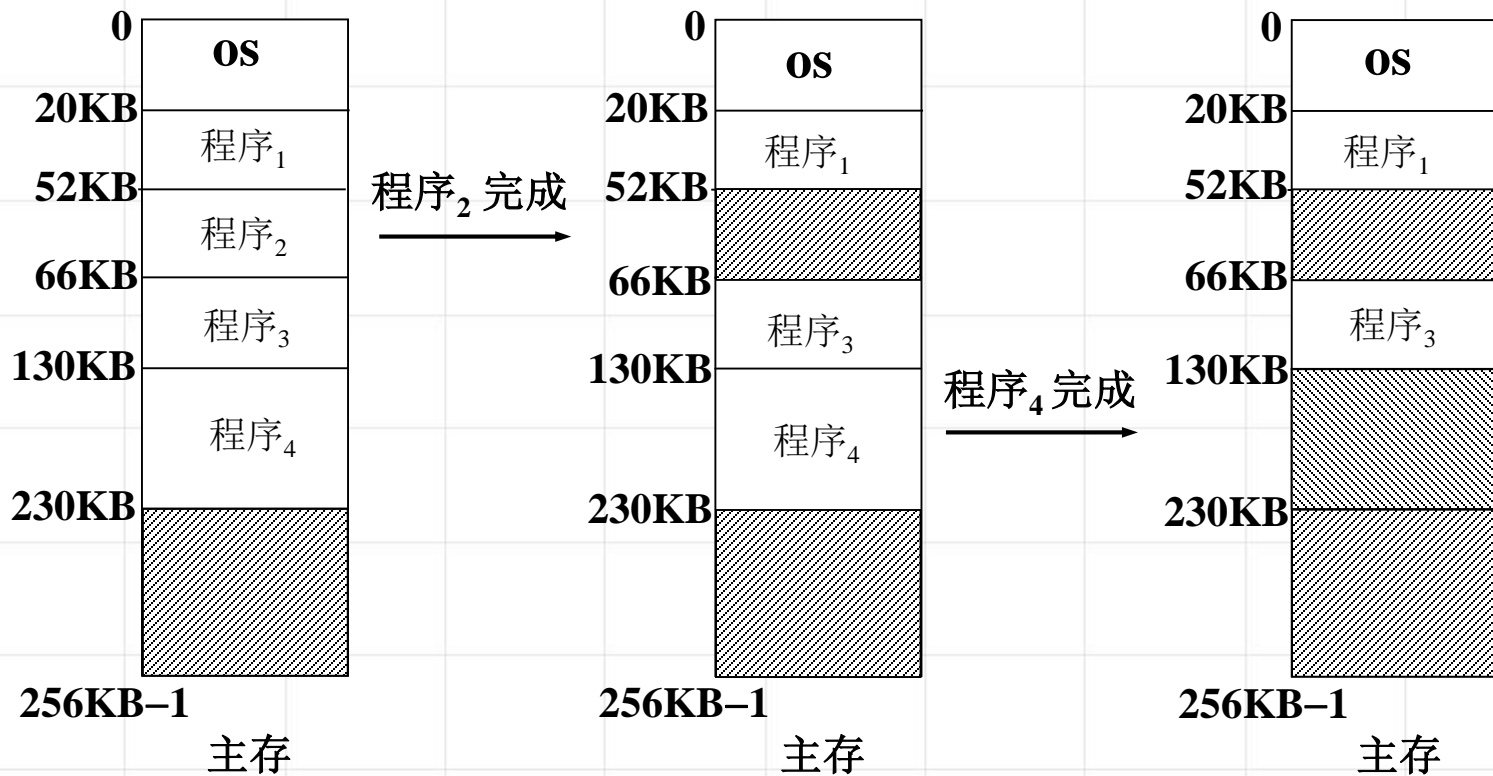
(2) 动态分区的分配、回收过程

① 动态分区的分配过程



动态分区分配示意图

② 动态分区的回收过程



动态分区分配中的存储区的释放

(3) 分区分配数据结构

① 主存资源信息块 (M_RIB)

② 分区描述器 (PD)

flag: 为 0 —— 空闲区

为 1 —— 已分配区

size: 分区大小

next: 空闲区——自由主存队列中的勾链字

已分配区——此项为零

M_RIB

等待队列头指针

空闲区队列头指针

主存分配程序入口地址

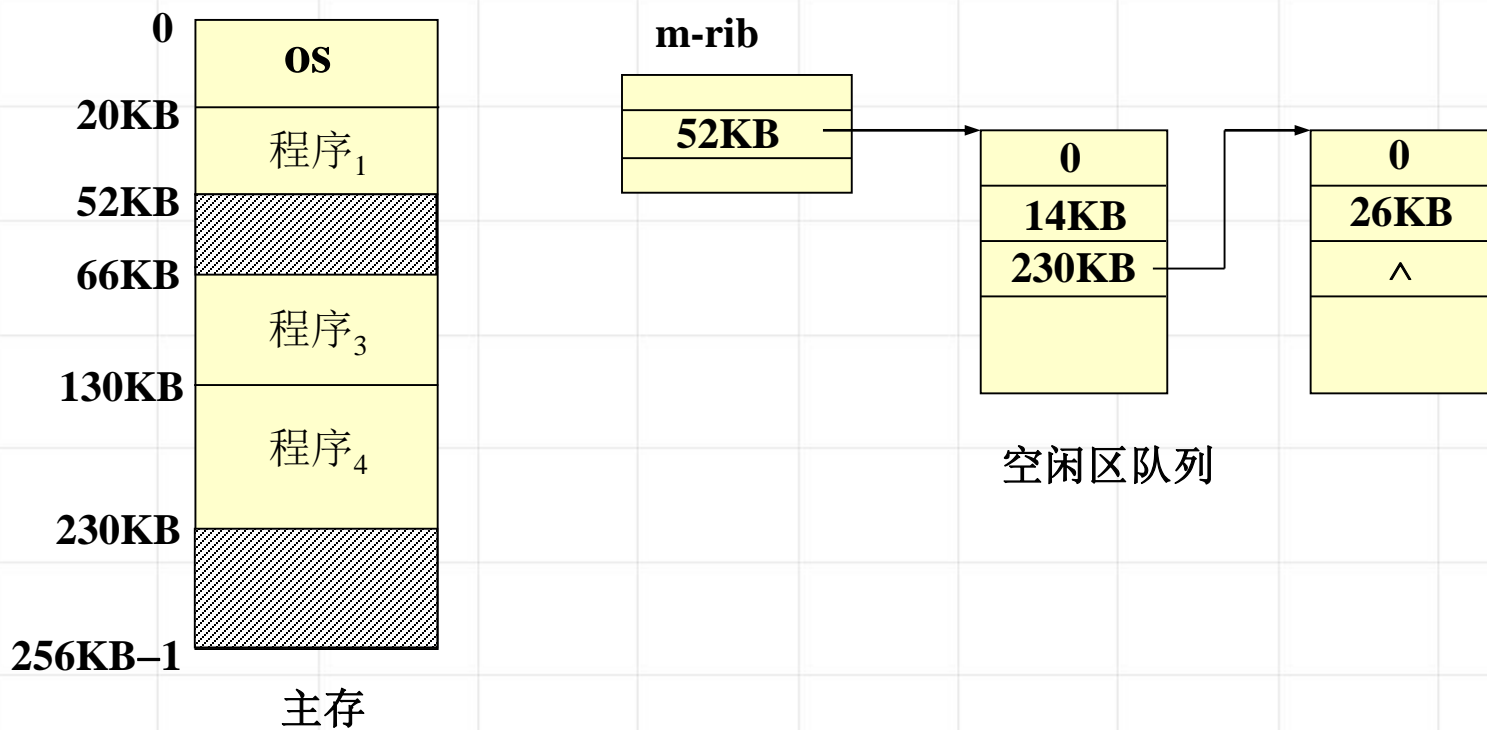
PD

分配标志	flag
------	------

大小	size
----	------

勾链字	next
-----	------

③ 空闲区队列结构



动态分区的空闲区队列结构

2. 分区的分配与回收

(1) 分区分配思路

① 寻找空闲块

依申请者所要求的主存区的大小，分区分配程序在自由主存队列中找一个满足用户需要的空闲块；

② 若找到了所需的空闲区，有两种情况

- i 空闲区与要求的大小相等，将该空闲区分配并从队列中摘除；
- ii 空闲区大于所要求的的大小，将空闲区分为两部分：一部分成为已分配区，建立已分配区的描述器；剩下部分仍为空闲区。返回所分配区域的首址；

③ 否则，告之不能满足要求。

(2) 分区回收思路

① 检查释放分区 (即为回收分区)在主存中的邻接情况

若上、下邻接空闲区，则合并，成为一个连续的空闲区

② 若回收分区不与任何空闲区相邻接

建立一个新的空闲区，并加入到空闲区队列中。

3. 放置策略

(1) 什么是放置策略

选择空闲区的策略，称为放置策略。

常用的放置策略——

首次匹配 (首次适应算法)

最佳匹配 (最佳适应算法)

最坏匹配 (最坏适应算法)

(2) 首次适应算法

① 什么是首次适应算法

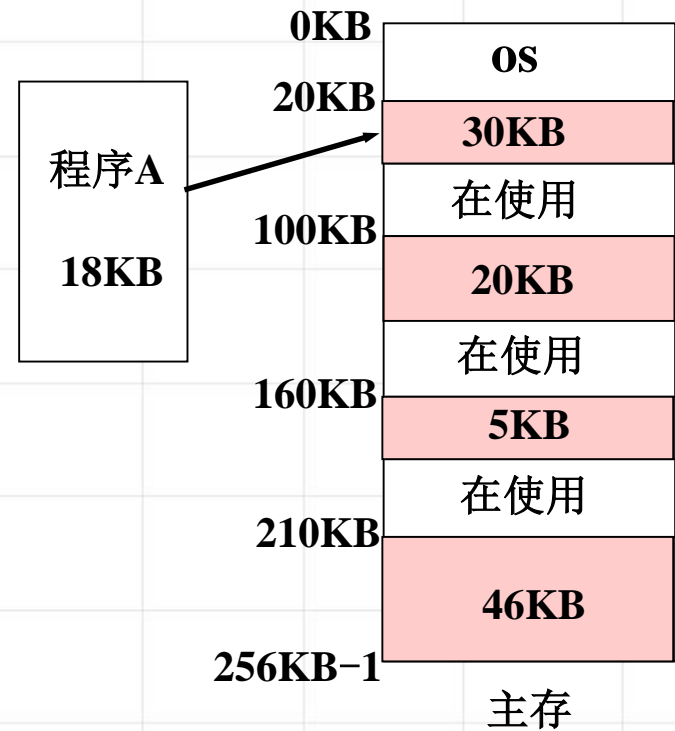
首次适应算法是将输入的程序放置到主存里第一个足够装入它的地址最低的空闲区中。

② 空闲区队列结构

空闲区地址由低到高排序

③ 首次适应算法的特点

尽可能地利用存储器中低地址的空闲区，而尽量保存高地址的空闲区。



三种放置策略的图解

首次适应算法

(3) 最佳适应算法

① 什么是最佳适应算法

最佳适应算法是将输入的程序放置到主存中与它所需大小最接近的空闲区中。

② 空闲区队列结构

空闲区大小由小到大排序

③ 最佳适应算法的特点

尽可能地利用存储器中小的空闲区，而尽量保存大的空闲区。



三种放置策略的图解
最佳适应算法

(4) 最坏适应算法

① 什么是最坏适应算法

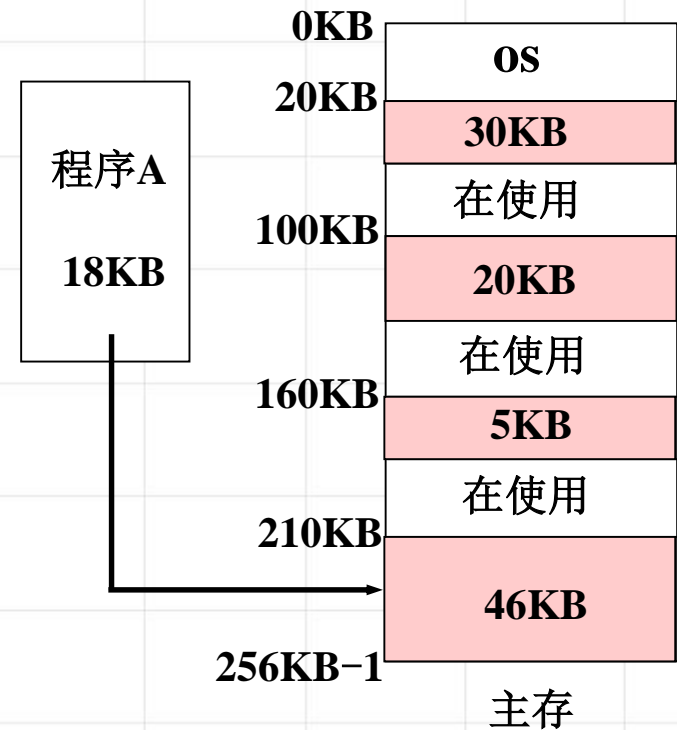
最坏适应算法是将输入的程序放置到主存中与它所需大小差距最大的空闲区中。

② 空闲区队列结构

空闲区大小由大到小排序

③ 最坏适应算法的特点

尽可能地利用存储器中大的空闲区。



三种放置策略的图解

最坏适应算法

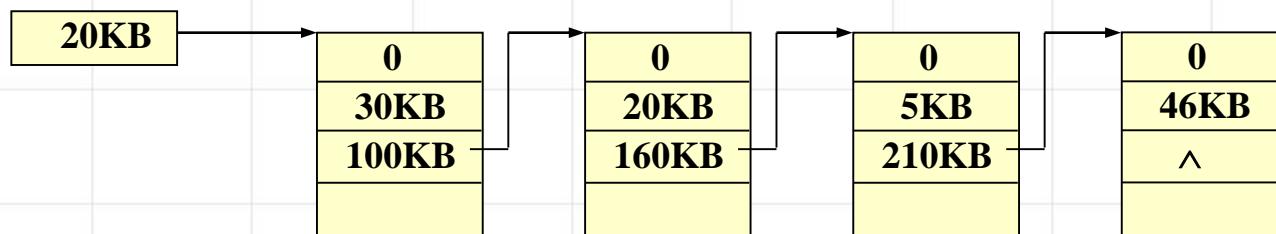
(5) 三种放置策略的讨论

① 题例

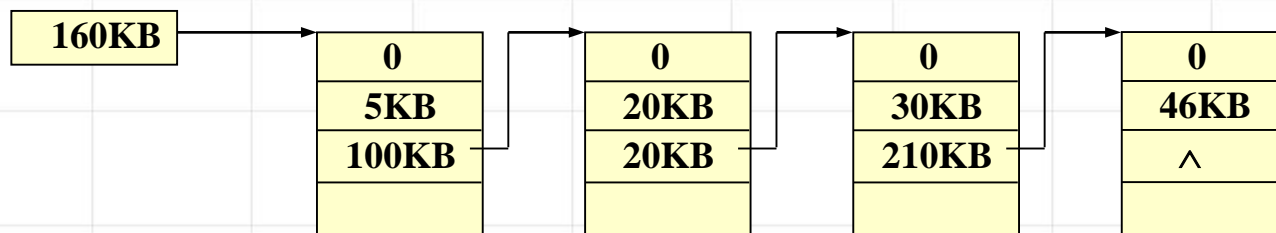
程序A要求18KB；程序B要求25KB；程序C要求30KB。

用首次适应算法、最佳适应算法、最坏适应算法来处理程序序列，看哪种算法合适。

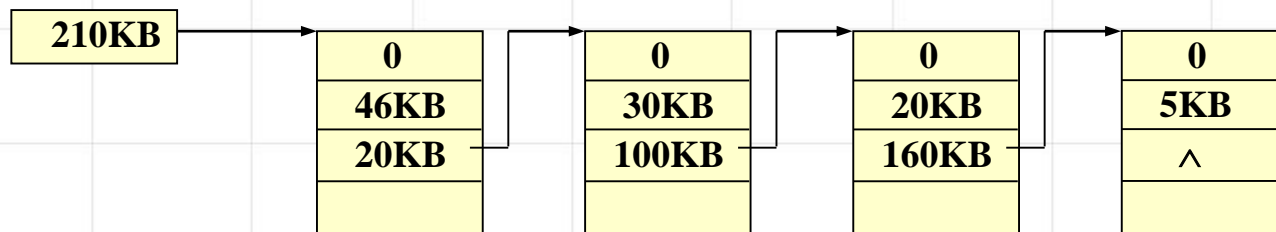
② 首次适应算法、最佳适应算法、最坏适应算法的队列结构



(a) 首次适应算法的空闲区队列



(b) 最佳适应算法的空闲区队列

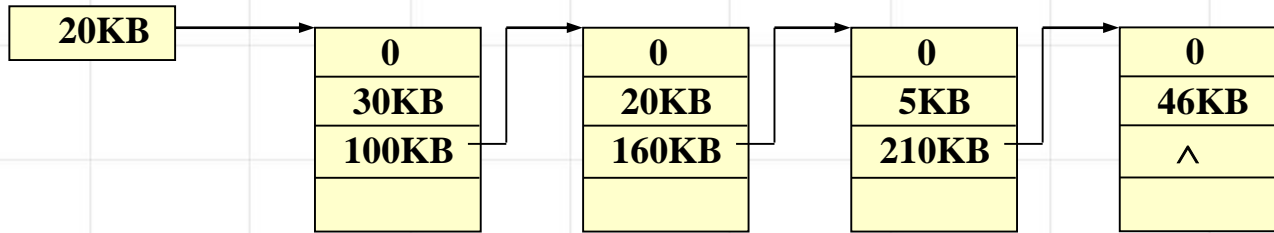


(c) 最坏适应算法的空闲区队列



主存分布示意图

i 首次适应算法



(a) 首次适应算法的空闲区队列

程序A要求18KB

程序B要求25KB

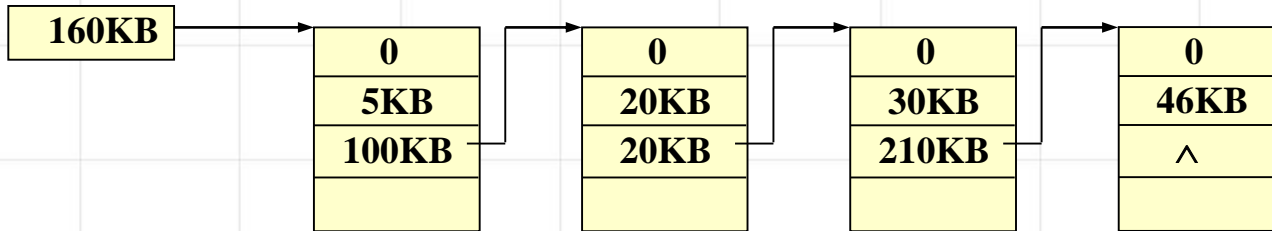
程序C要求30KB

首次适应算法对该作业序列是不合适的



主存分布示意图

ii 最佳适应算法



(b) 最佳适应算法的空闲区队列

程序A要求18KB

程序B要求25KB

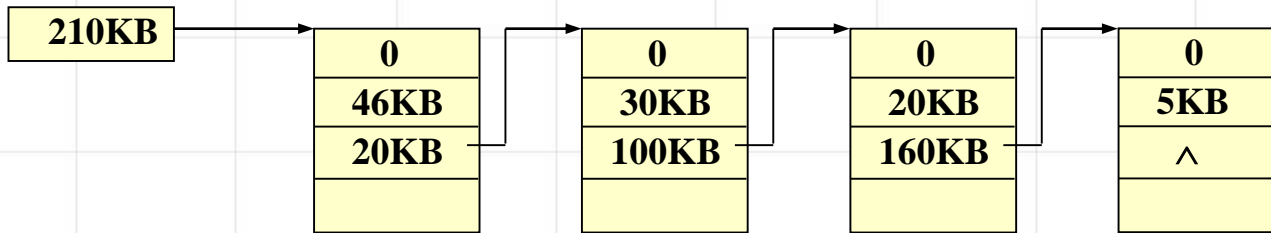
程序C要求30KB

最佳适应算法对该程序序列是合适的



主存分布示意图

iii 最坏适应算法



(c) 最坏适应算法的空闲区队列

程序A要求18KB

程序B要求25KB

程序C要求30KB

最坏适应算法对该程序序列是不合适的



主存分布示意图

4. 碎片问题及拼接技术

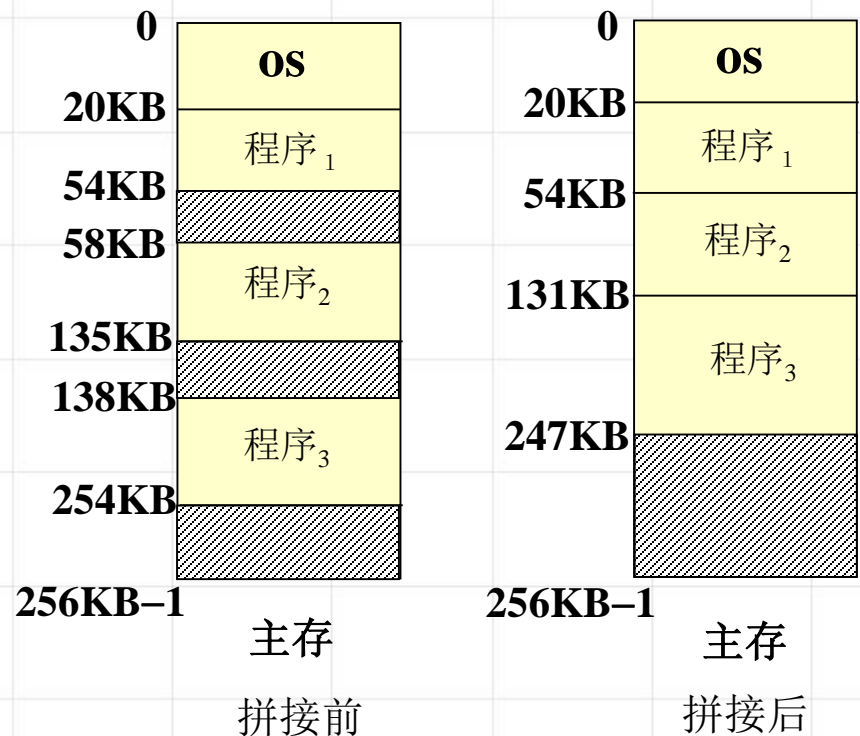
(1) 什么是碎片问题

在已分配区之间存在着的一些没有被充分利用的空闲区。

如何解决碎片问题？

(2) 什么是拼接技术

所谓拼接技术是指移动存储器中某些已分配区中的信息，使本来分散的空闲区连成一个大的空闲区。



分区分配中的存储区拼接