计算机操作系统

Operating Systems

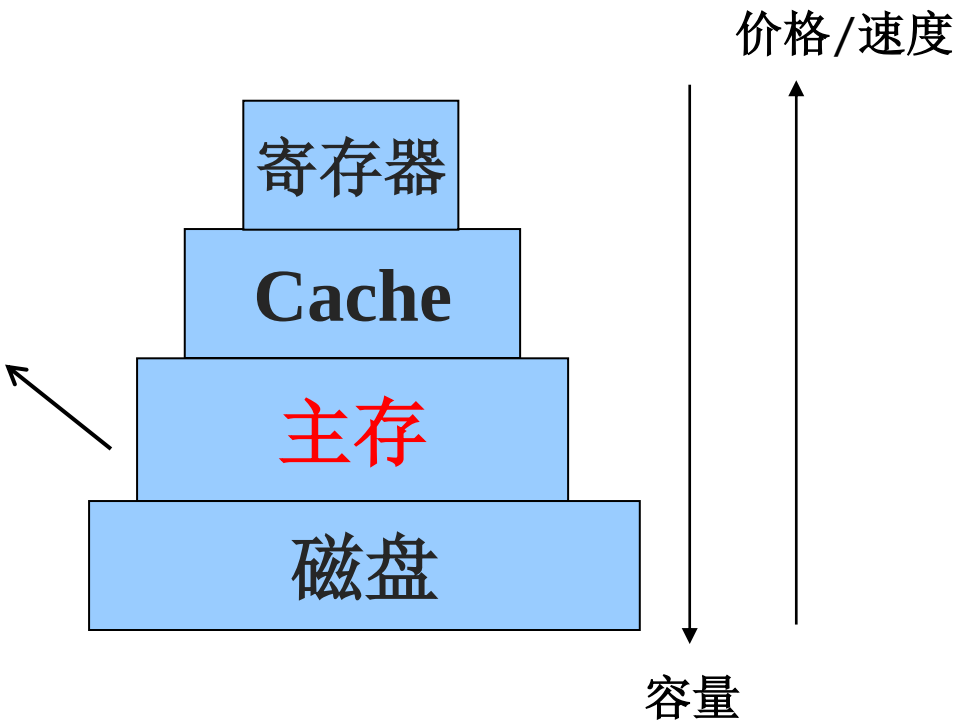
李　琳

第四章 存储器管理

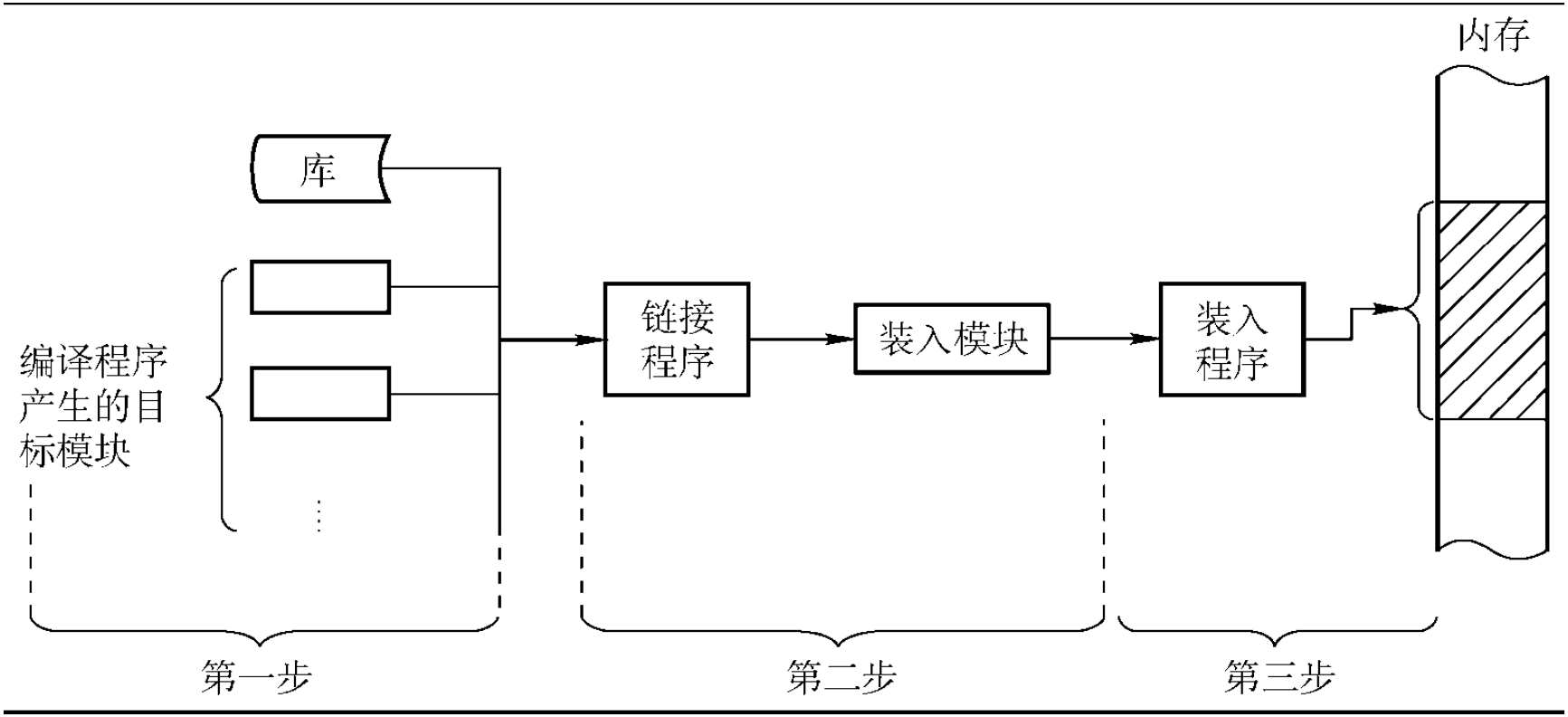
计算机存储体系

- 由存储单元（字节或字）组成的一维连续的地址空间，简称内存空间。
- 用来存放当前正在运行程序的代码及数据，是程序中指令本身地址所指的、亦即程序计数器所指的存储器
- 性能“瓶颈”：关键、紧张
- 帕金森定律：

  内存多大，程序多长



4.1 程序的装入和链接

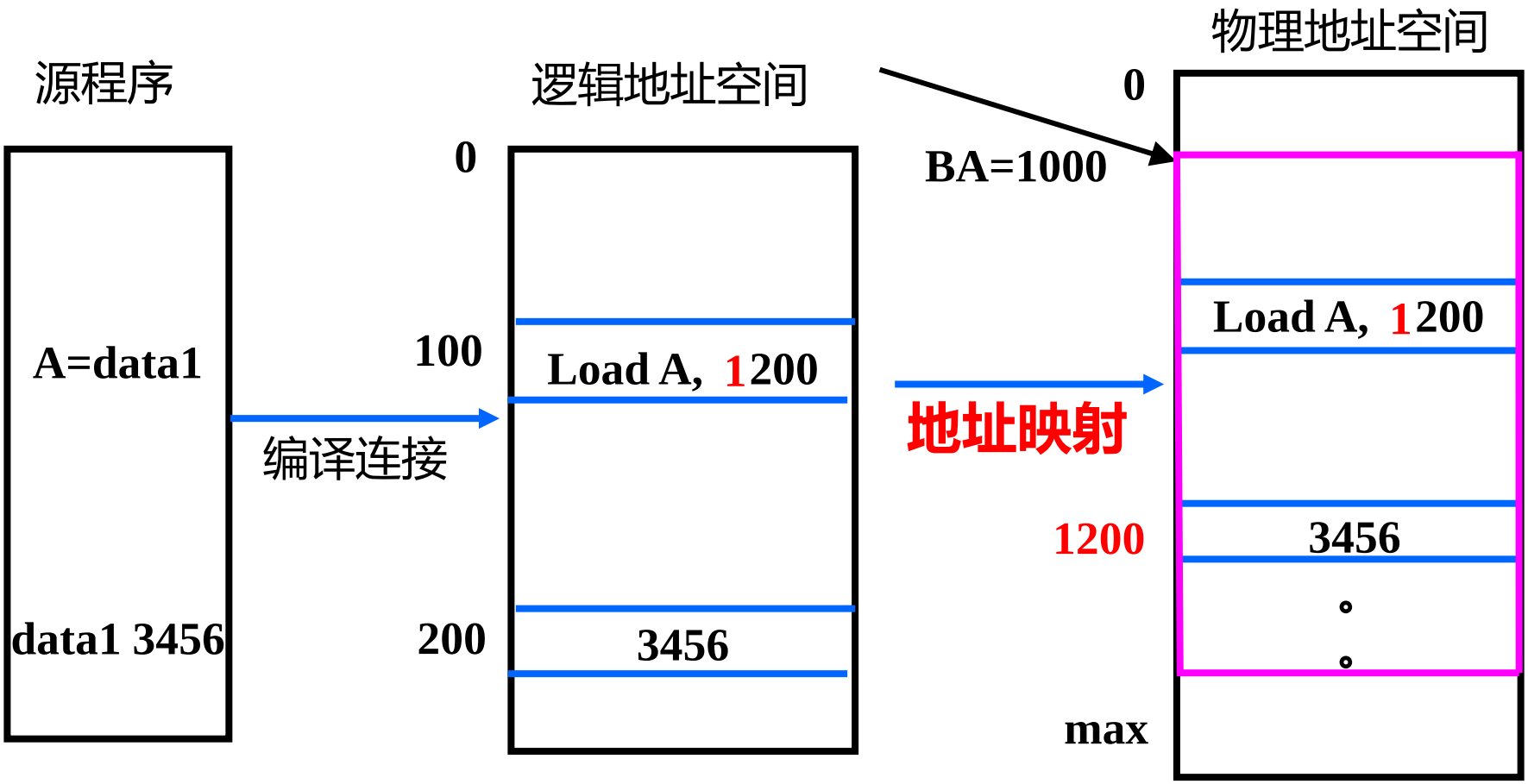


.cpp .h 用户程序 → .obj .lib静态链接库 → .exe .dll 动态链接库

4.1 程序的装入和链接

4.1.1 地址映射

- 逻辑地址（相对地址，虚地址）：用户程序经过编译和链接后形成的机器代码
- 物理地址（绝对地址，实地址）：内存中存储单元的地址，可直接寻址



4.1 程序的装入和链接

4.1.1 地址映射

- 地址重定位（地址映射）

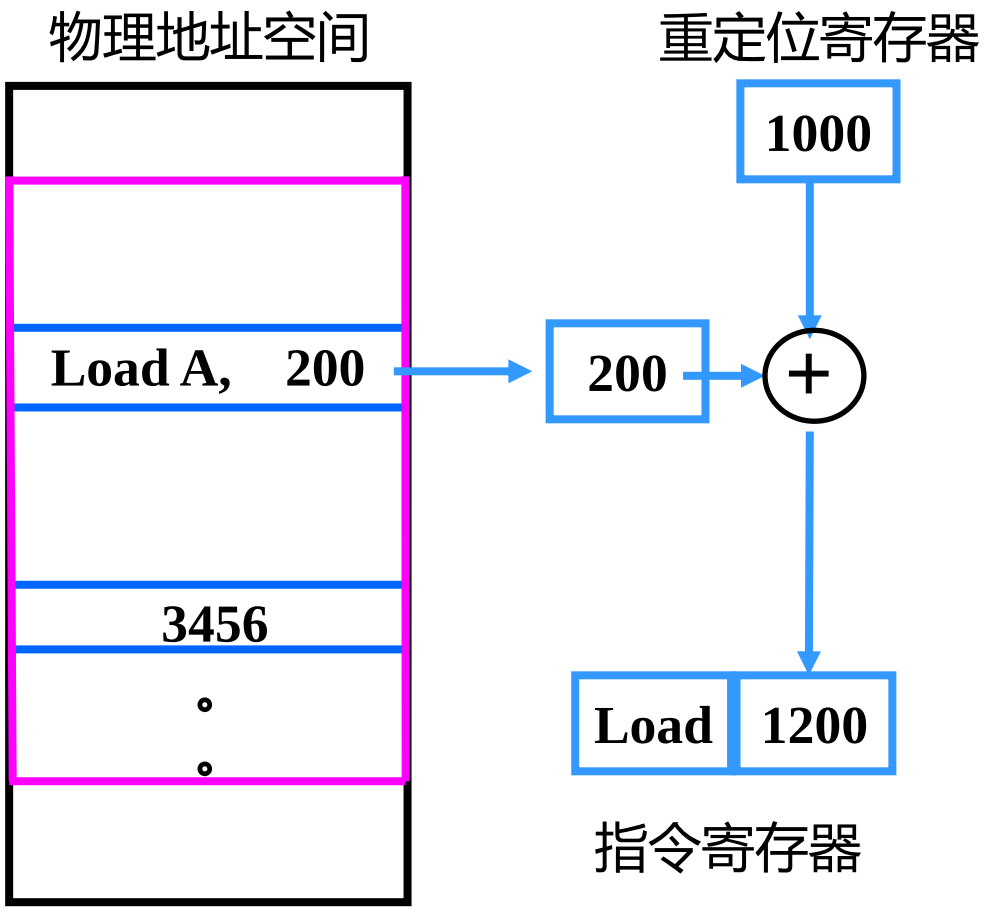
为了保证CPU执行指令时可正确访问存储单元，需将用户程序中的逻辑地址转换为运行时由机器直接寻址的物理地址，这一过程称为地址映射

- 静态地址重定位

当用户程序被装入内存时，一次性实现逻辑地址到物理地址的转换，一般在装入内存时由软件(OS)完成

- 动态地址重定位

在程序运行过程中要访问数据时再进行地址变换，硬件上需要寄存器的支持



4.1 程序的装入和链接

4.1.2 程序的装入

- 绝对装入方式
  - ✓编程时，直接确定内存位置
  - ✓不灵活，不支持多道程序设计技术
- 可重定位装入方式—静态地址重定位
  - ✓虚拟技术：编程时使用逻辑地址，运行时使用物理地址；
  - ✓装入时，进行地址重定位；
  - ✓灵活，支持多道程序设计技术；
- 动态运行时装入方式—动态地址重定位
  - ✓编程时，使用逻辑地址，运行时使用物理地址；
  - ✓运行时，进行地址重定位；
  - ✓灵活;支持多道程序设计技术；支持程序“搬家”

4.1 程序的装入和链接

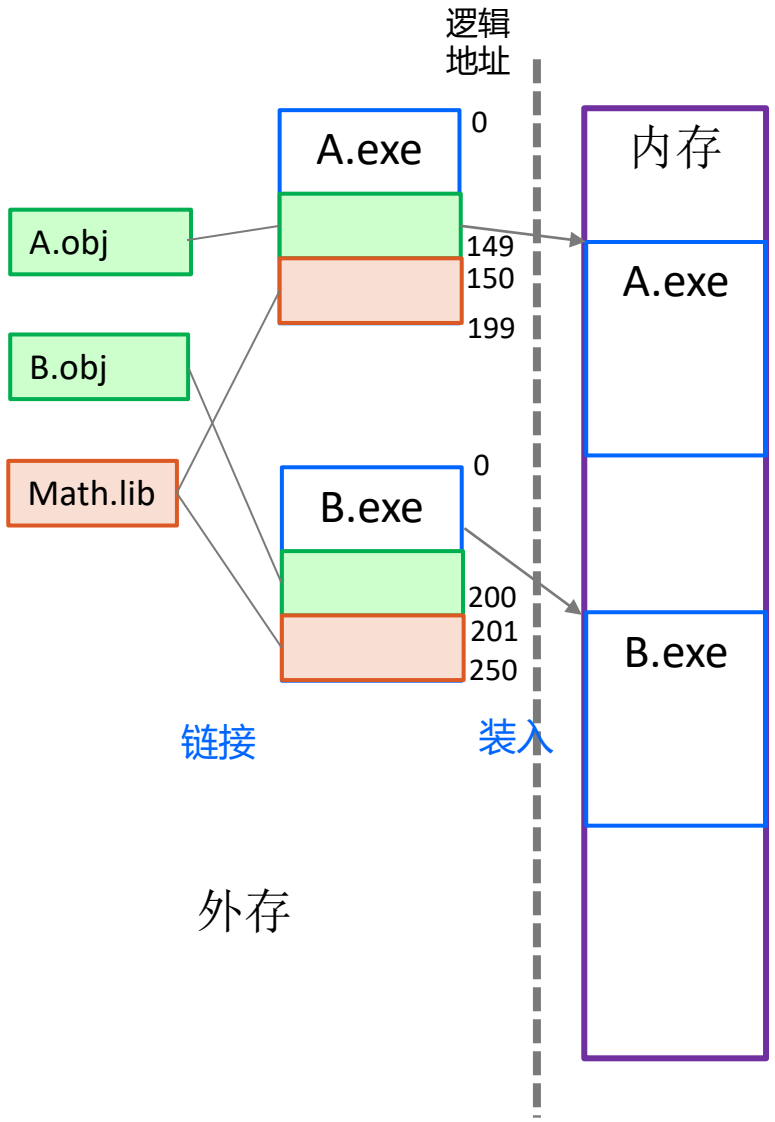
4.1.3 程序的链接

- 静态链接

  编程时，链接所有模块；
- 装入时动态链接

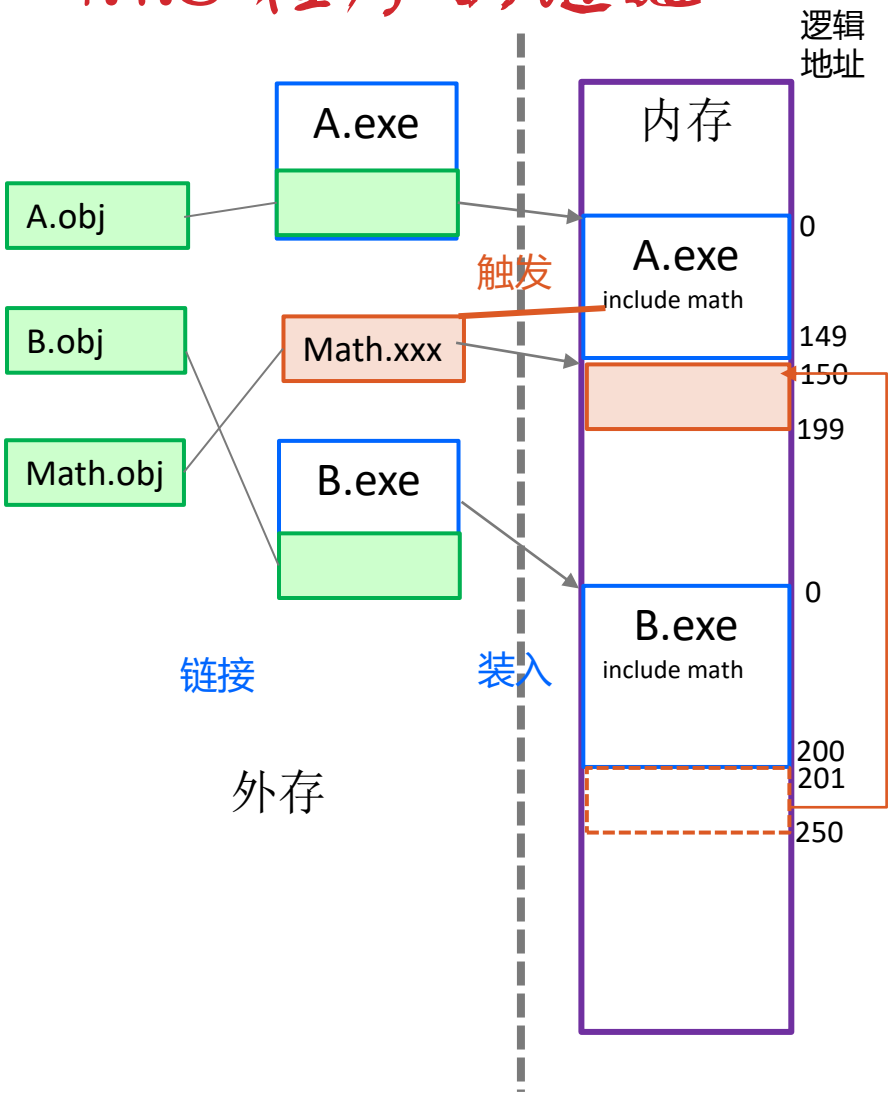
  装入时，链接所有模块；
- 运行时动态链接

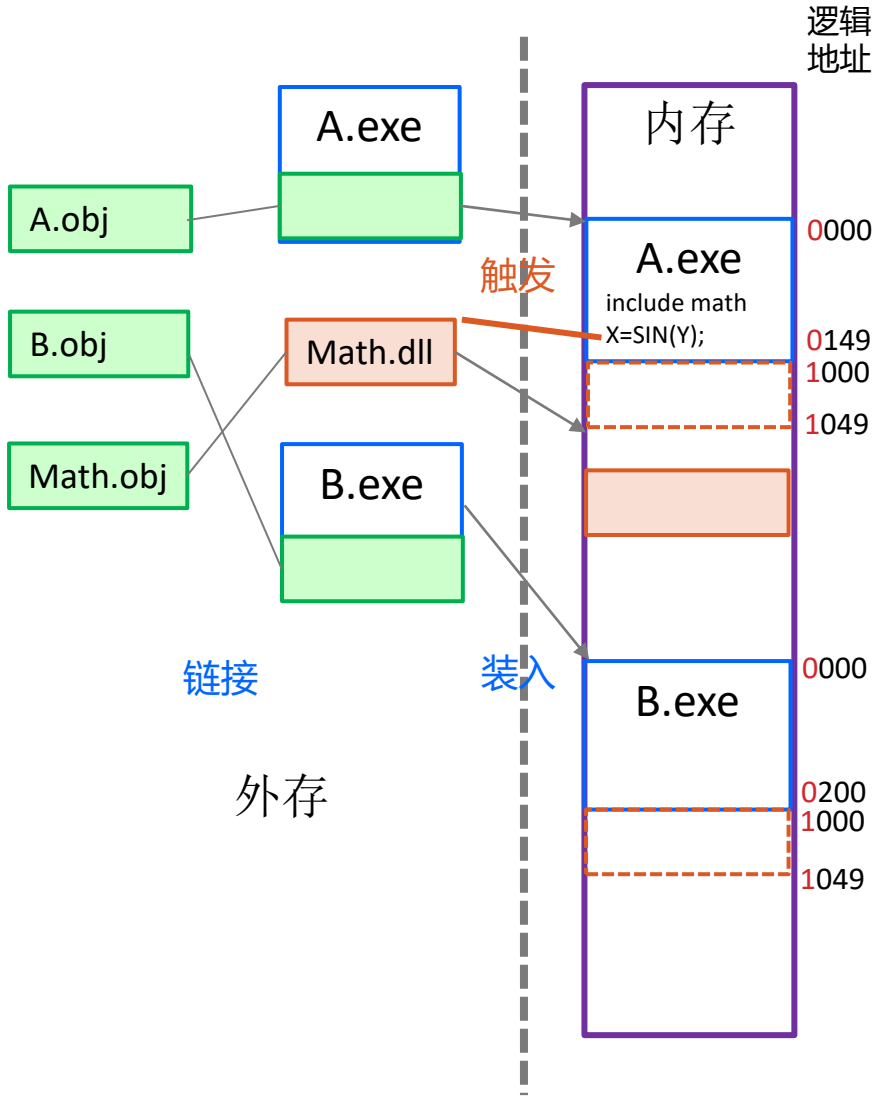
  运行时，根据需要链接模块；



4.1 程序的装入和链接

4.1.3 程序的链接





试一试

- 在虚拟内存管理中，地址变换机构将逻辑地址变换为物理地址，形成该逻辑地址的阶段是（ ）

  A. 编辑 B. 编译 C. 链接 D. 装载

- 要保证一个进程在主存中被改变了存放位置后仍能正确执行，则对主存空间应采用（ ）技术

  A、静态分配 B、静态重定位 C、动态分配 D、动态重定位

- 对重定位存储管理方式，以下哪个是正确的（ ）。

  A．在整个系统中设置一个重定位寄存器

  B．为每道程序设置一个重定位寄存器

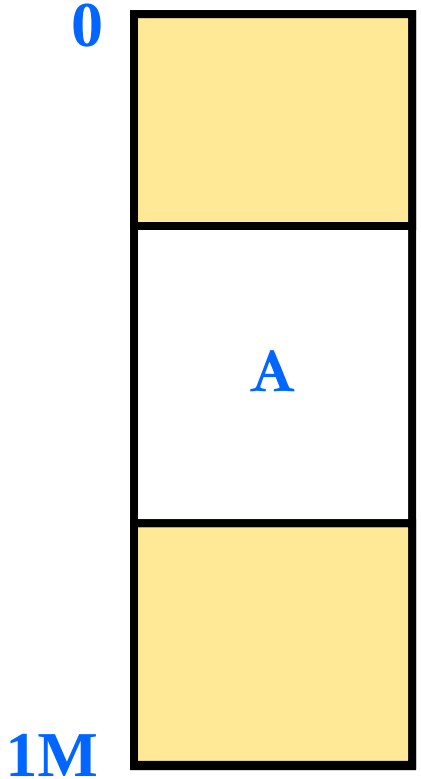
  C．为每道程序设置两个重定位寄存器

  D．为每道程序和数据都设置一个重定位寄存器

4.2 连续分配方式

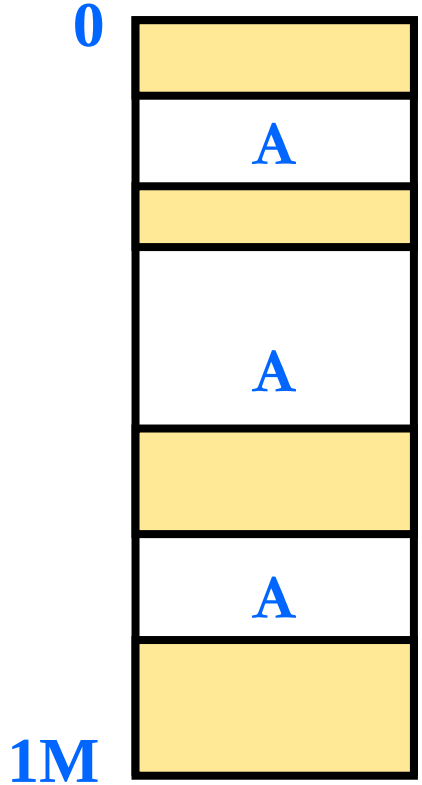
连续分配方式

为程序分配一段连续存储空间



离散分配方式

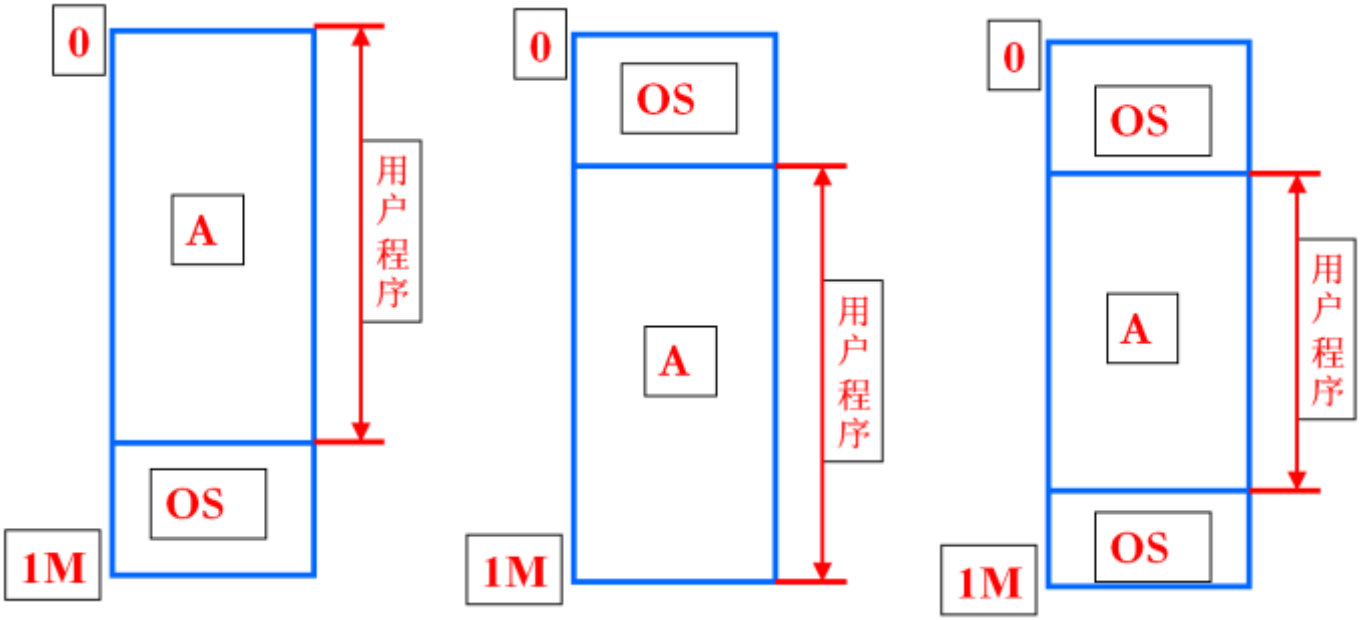
为程序分配若干段连续的存储空间



4.2 连续分配方式

4.2.1 单一连续分配存储管理

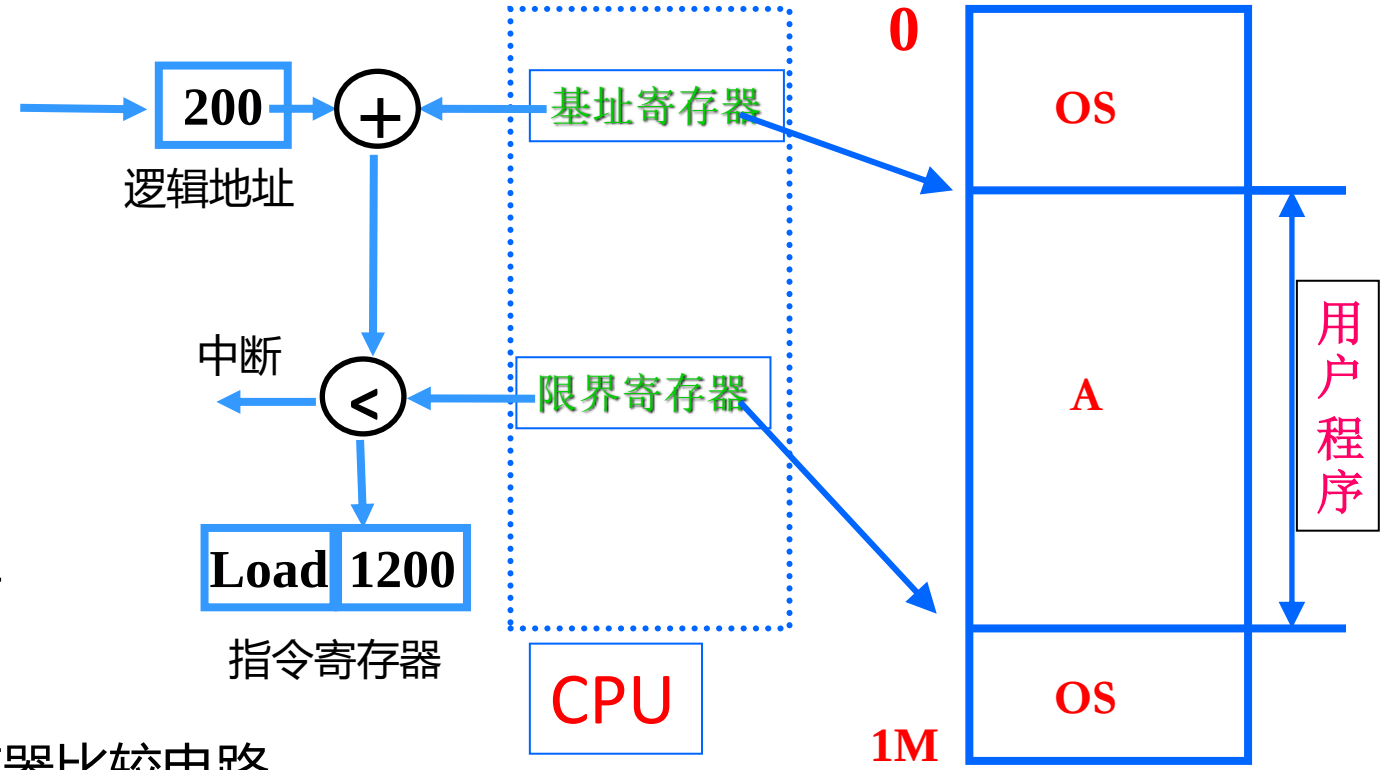
- 应用背景
  - ✓单用户系统。
  - ✓在一段时间内，只有一个进程在内存
- 基本思想
  - ✓内存分为两个区域：一个供操作系统使用，一个供用户使用



4.2 连续分配方式

4.2.1 单一连续分配存储管理

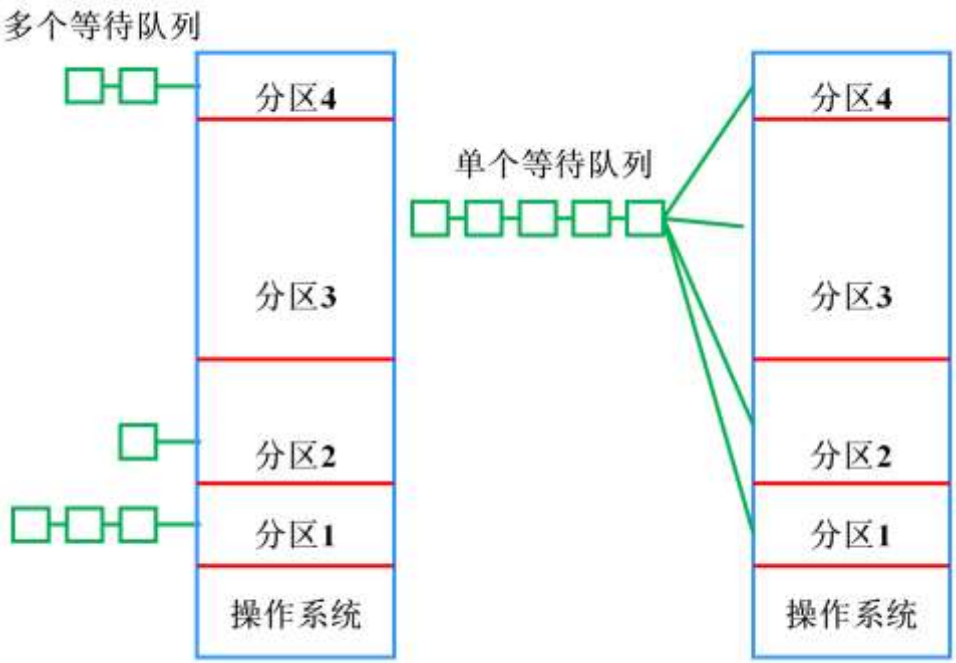
- 分配
  - ✓不需要分配
  - ✓load（）
- 回收
  - ✓不需要回收
  - ✓覆盖
- 地址映射
  - ✓绝对装入
  - ✓三种方式都可
- 地址保护
  - ✓增加限界寄存器比较电路
  - ✓越界检查



4.2 连续分配方式

4.2.2 固定分区分配存储管理

- 应用背景
  - ✓多道程序设计系统
- 基本思想
  - ✓将用户区划分为若干固定大小的内存区域
  - ✓每个分区可放一个程序运行
  - ✓分区大小可以相同，也可以不同
  - ✓OS初始化阶段定义



4.2 连续分配方式

4.2.2 固定分区分配存储管理

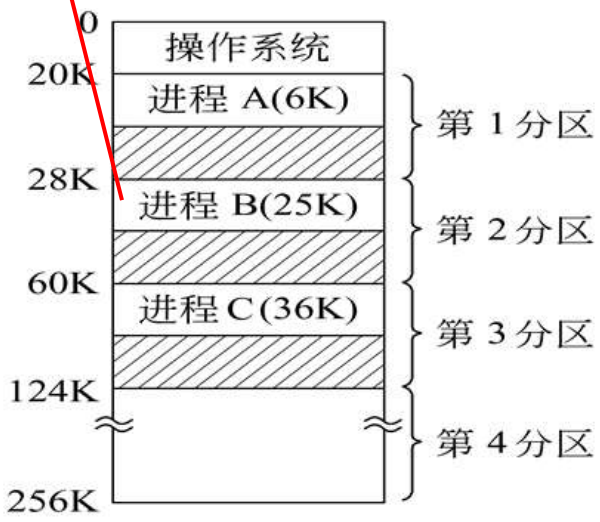
- 分配
  - ✓搜索分区表，改状态
- 回收
  - ✓搜索分区表，改状态
- 地址映射
  - ✓静态重定位
  - ✓动态重定位
- 地址保护
  - ✓增加限界寄存器比较电路
  - ✓越界检查

思考：分区大小选择会对内零头产生什么影响？

内零头：分配给进程，而进程未用到的内存部分

区号	分区长度	起始地址	状态
1	8K	20K	已分配
2	32K	28K	已分配
3	64K	60K	已分配
4	132K	124K	未分配

(a) 分区说明表

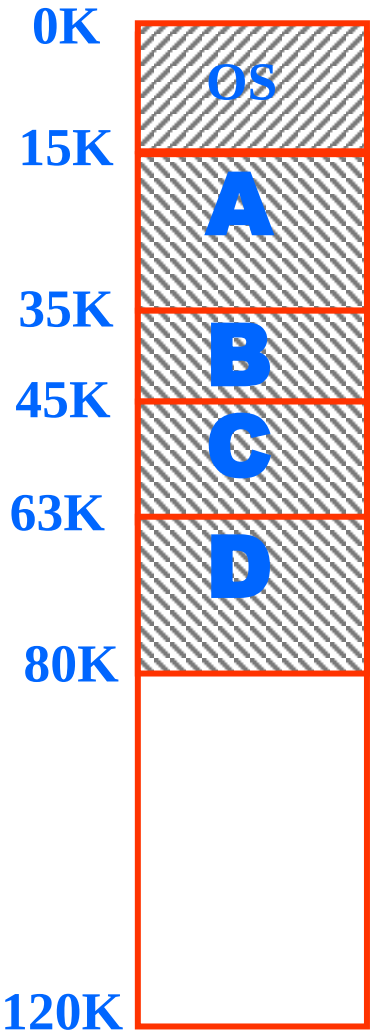


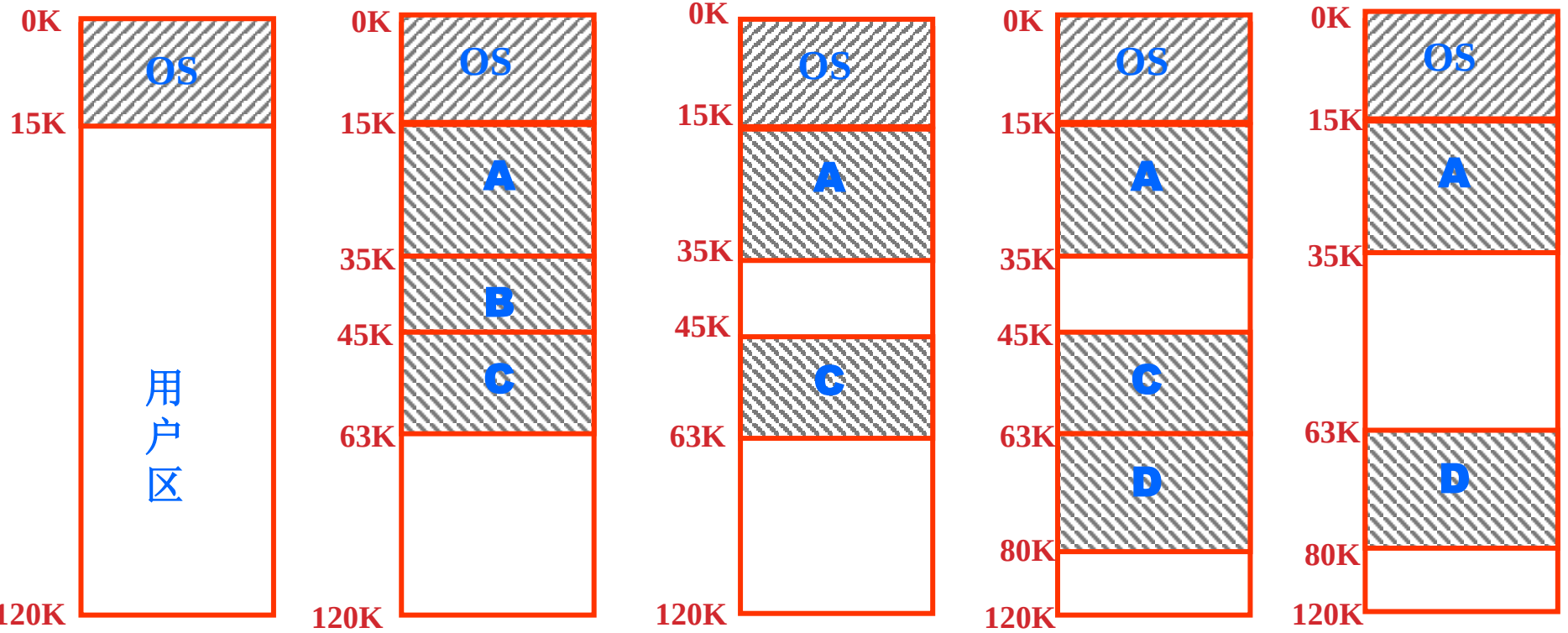
(b) 内存状态

4.2 连续分配方式

4.2.2 动态分区分配存储管理

- 应用背景
  - ✓多道程序设计系统
- 基本思想
  - ✓OS初始化阶段，将用户区划分为一个大的空白分区
  - ✓每个程序运行时，划分出一个大小合适的分区
  - ✓程序执行结束，其所在的分区撤消





分区表的变化

始址	长度	标志
15K	105K	未分配

始址	长度	标志
15K	20K	A
35K	10K	B
45K	18K	C
63K	57K	未分配

始址	长度	标志
15K	20K	A
35K	10K	未分配
45K	18K	C
63K	57K	未分配

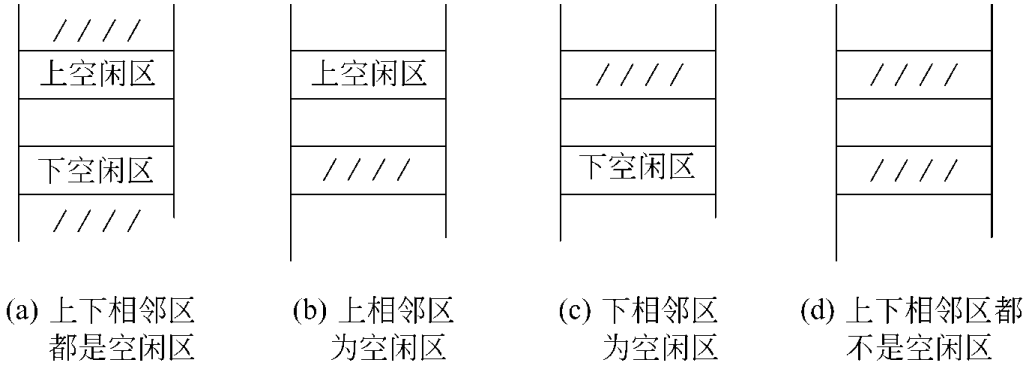
始址	长度	标志
15K	20K	A
35K	10K	未分配
45K	18K	C
63K	17K	D
80K	40K	未分配

始址	长度	标志
15K	20K	A
35K	28K	未分配
63K	17K	C
80K	40K	未分配

4.2 连续分配方式

4.2.2 动态分区分配存储管理

- 分配
  - ✓大部分情况下一分为二
  - ✓大部分是一个增加分区表表项的过程
- 回收
  - ✓可能合二、合三为一
  - ✓可能是一个减少分区表表项的过程
- 地址映射
  - ✓静态重定位
  - ✓动态重定位
- 地址保护
  - ✓增加限界寄存器比较电路
  - ✓越界检查

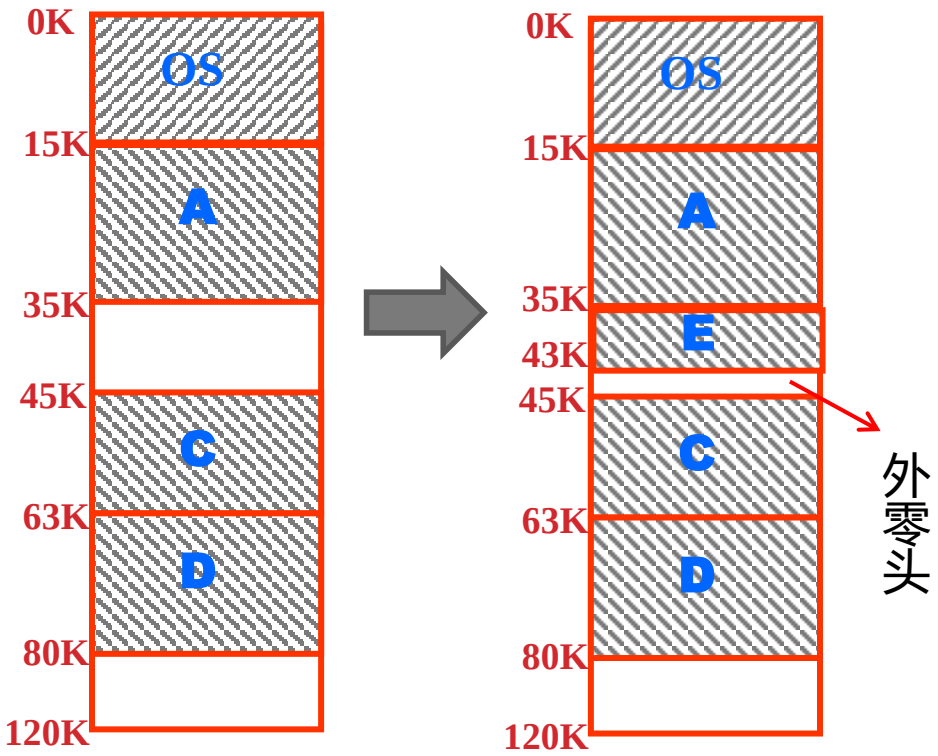


(a) 上下相邻区都是空闲区　(b) 上相邻区为空闲区　(c) 下相邻区为空闲区　(d) 上下相邻区都不是空闲区

4.2 连续分配方式

4.2.2 动态分区分配存储管理

- 外零头现象（碎片）
  - ✓经过一段时间的分配回收后，内存中存在很多很小的空闲块。它们每一个都很小，不足以满足分配要求；但其总和满足分配要求。这些空闲块被称为碎片造成存储资源的浪费

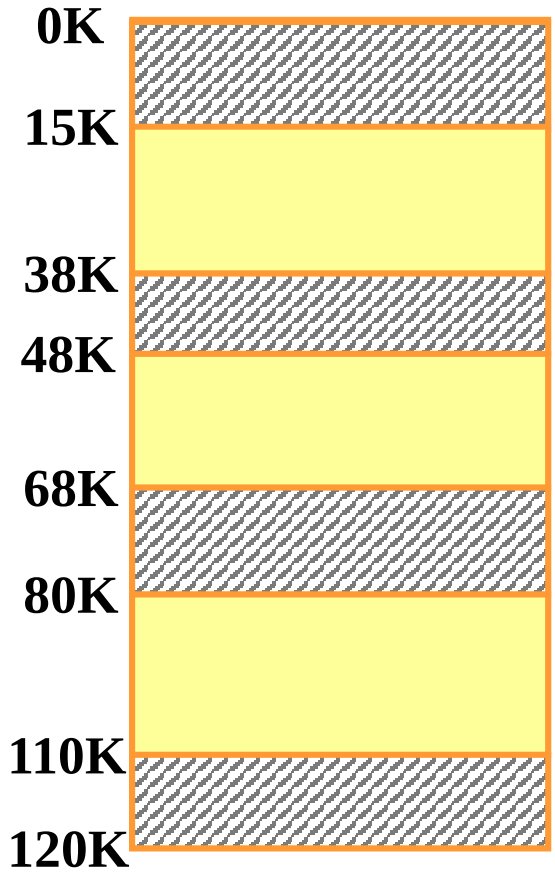


外零头和内零头的区别?

4.2 连续分配方式

4.2.2 动态分区分配存储管理

- 分配算法：首次适应算法



空闲区表

始址	长度	标志
15K	23K	未分配
48K	20K	未分配
80K	30K	未分配
		空
		空

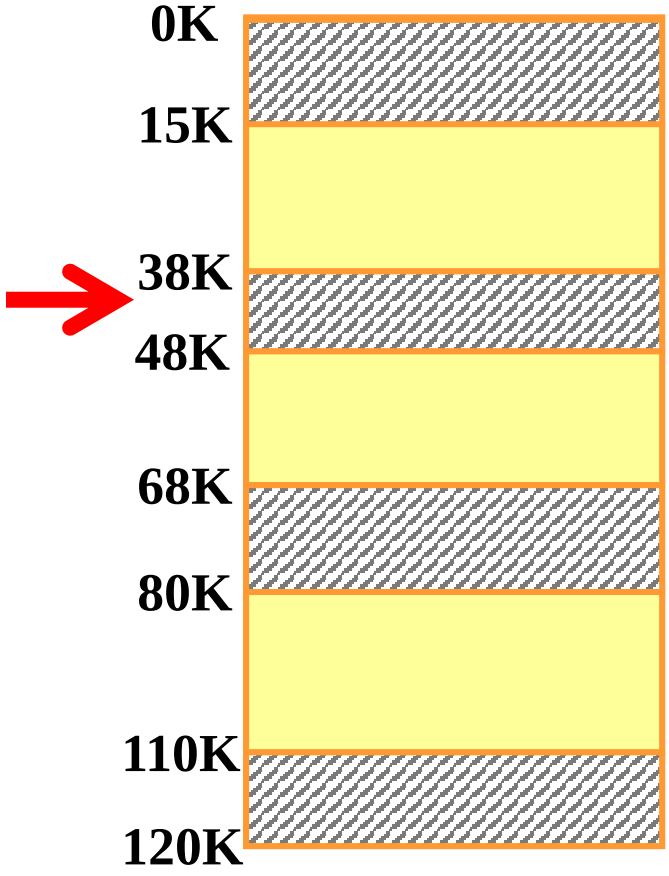
新增任务，要求内存27K

算法倾向于优先利用内存中低址部分的空闲分区，从而保留了高址部分的大空闲区

4.2 连续分配方式

4.2.2 动态分区分配存储管理

- 分配算法：循环首次适应算法



空闲区表

始址	长度	标志
15K	23K	未分配
48K	20K	未分配
80K	30K	未分配
		空
		空

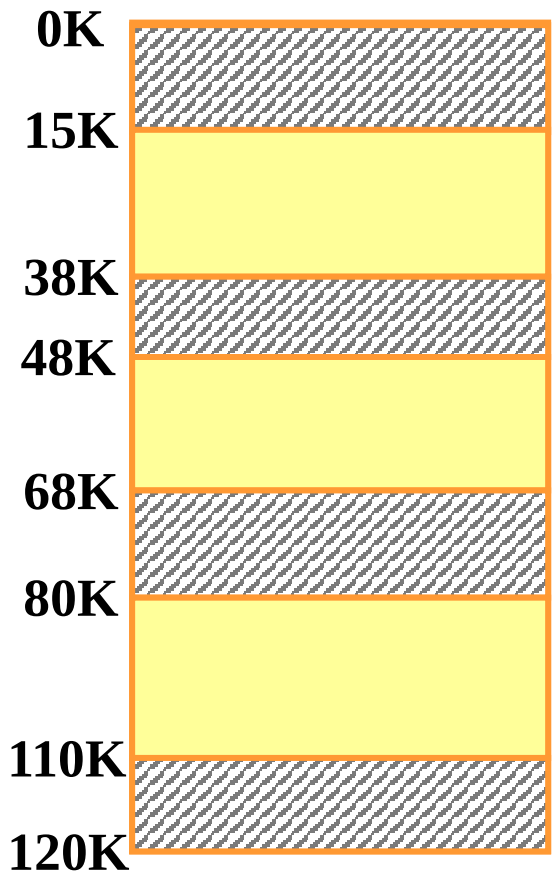
新增任务，要求内存18K

算法倾向于扫描所有空闲区，从而很难保留一个较大的空闲区

4.2 连续分配方式

4.2.2 动态分区分配存储管理

- 分配算法：最佳适应算法



空闲区表

始址	长度	标志
15K	23K	未分配
48K	20K	未分配
80K	30K	未分配
		空
		空

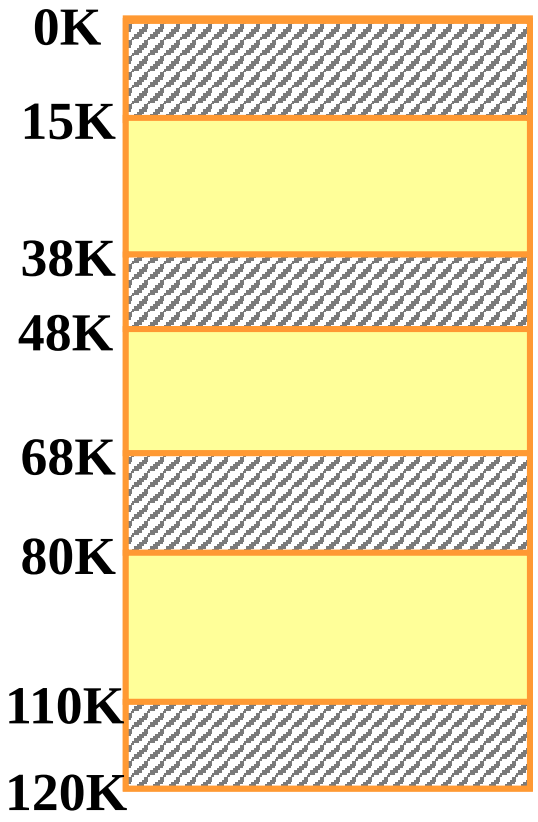
新增任务，要求内存22K

算法**倾向于切分大小最接近的分区，**因而易于造成碎片

4.2 连续分配方式

4.2.2 动态分区分配存储管理

- 分配算法：最差适应算法



空闲区表

始址	长度	标志
15K	23K	未分配
48K	20K	未分配
80K	30K	未分配
		空
		空

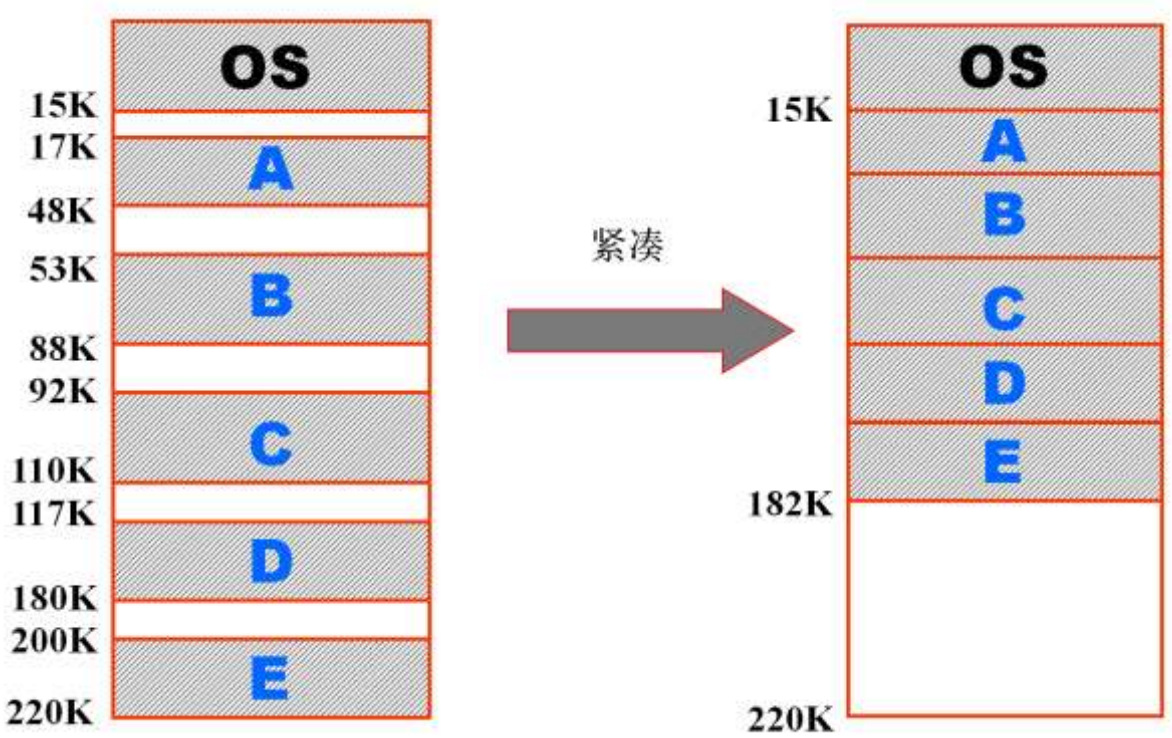
新增任务，要求内存12K

算法倾向于切分最大的分区，因而碎片情况会很少

4.2 连续分配方式

4.2.2 动态分区分配存储管理

- 紧凑方法



问题1： 在内存管理的什么阶段使用紧凑（分配、回收、地址映射）？

问题2： 使用紧凑方法的系统的地址映射应该采用什么方式？

问题3： 如何紧凑，才能保证算法效率最高？

试一试

- 每次分配时总是从低地址到高地址顺序查找空闲区表，找到第一个能满足作业长度要求的空闲区，此种分配算法称为（　）

  A、首次适应分配算法　　B、最坏适应分配算法

  C、最优适应分配算法　　D、随机适应分配算法

- 在下列存储管理方案中，不适用于多道程序设计的是（　　）

  A、可变分区分配　　B、固定分区分配

  C、单一连续分配　　D、分页存储管理

- 在可变式分区存储管理中，某作业完成后要收回其主存空间，该空间可能与相邻空闲区合并，在修改空闲区表时使空闲区数不变且空闲区起始地址不变的情况是（　）

  A、有上邻空闲区但无下邻空闲区

  B、有上邻空闲区也有下邻空闲区

  C、无上邻空闲区但有下邻空闲区

  D、无上邻空闲区也无下邻空闲区

填空题 1分

设置

某基于动态分区存储管理的计算机，其主存容量为55MB（初始为空闲），采用最佳适配算法，分配和释放的顺序为：分配15MB，分配30MB，释放15MB，分配8MB，分配6MB，此时主存中最大空闲分区的大小是 [填空1]

正常使用填空题需3.0以上版本雨课堂

作答