西安交通大学
软件学院

操作系统原理

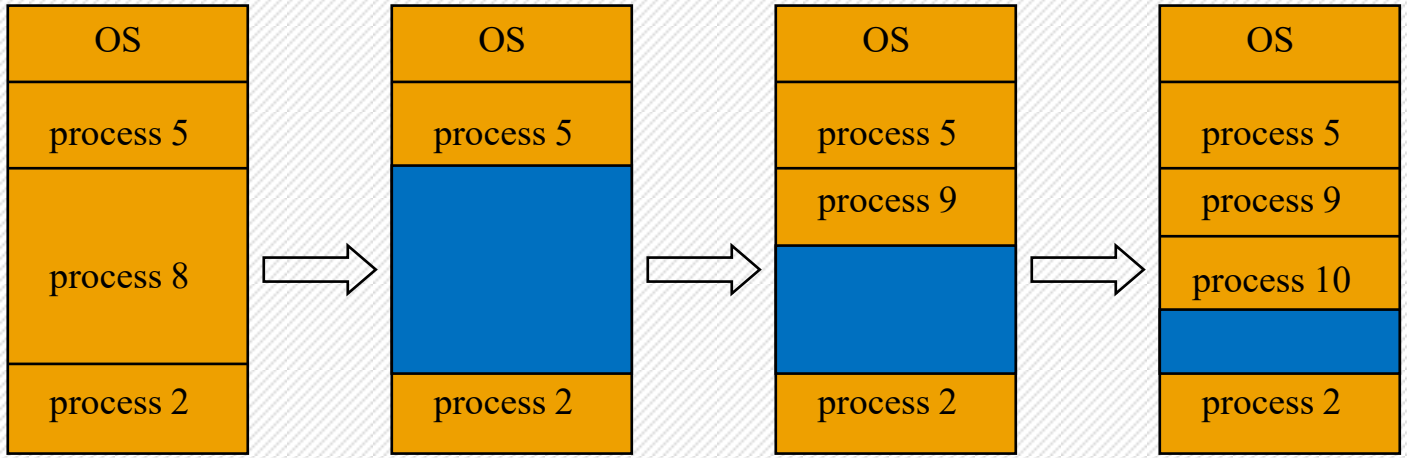
Operating System Principle

田丽华

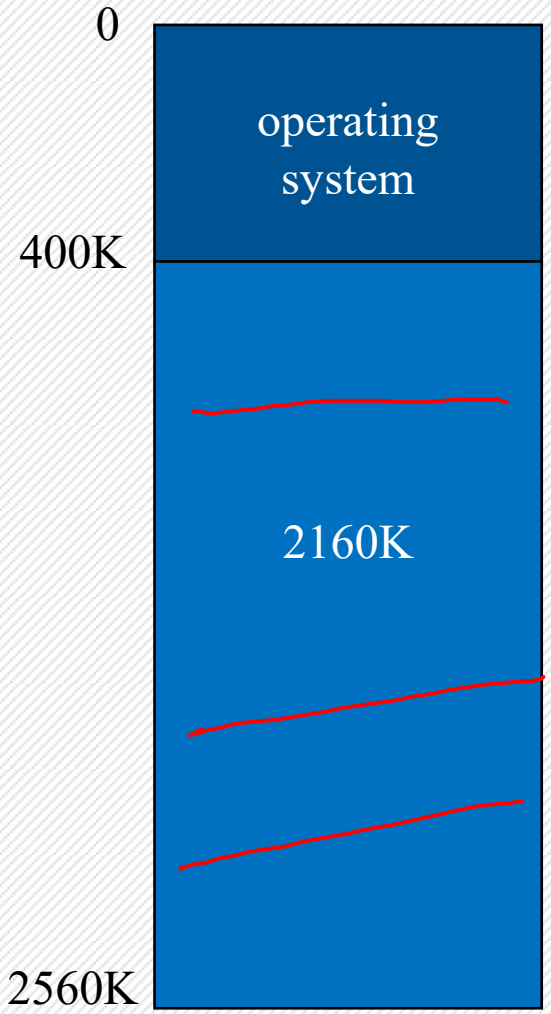
8-3 动态分区分配

Contiguous Allocation (Cont.)

- Multiple-partition allocation（多分区分配）
- 分区的划分是动态的,不是预先确定的
  - *Hole* – block of available memory; holes of various size are scattered throughout memory.（分区—可用的内存块，不同大小的分区分布在整个内存中。）
  - When a process arrives, it is allocated memory from a hole large enough to accommodate it.（当一个进程到来的时候，它将从一个足够容纳它分区中分配内存。）
  - Operating system maintains information about（操作系统包含以下信息）:a) allocated partitions （分配的分区） b) free partitions (hole)（空的分区）



Contiguous Allocation (Cont.)



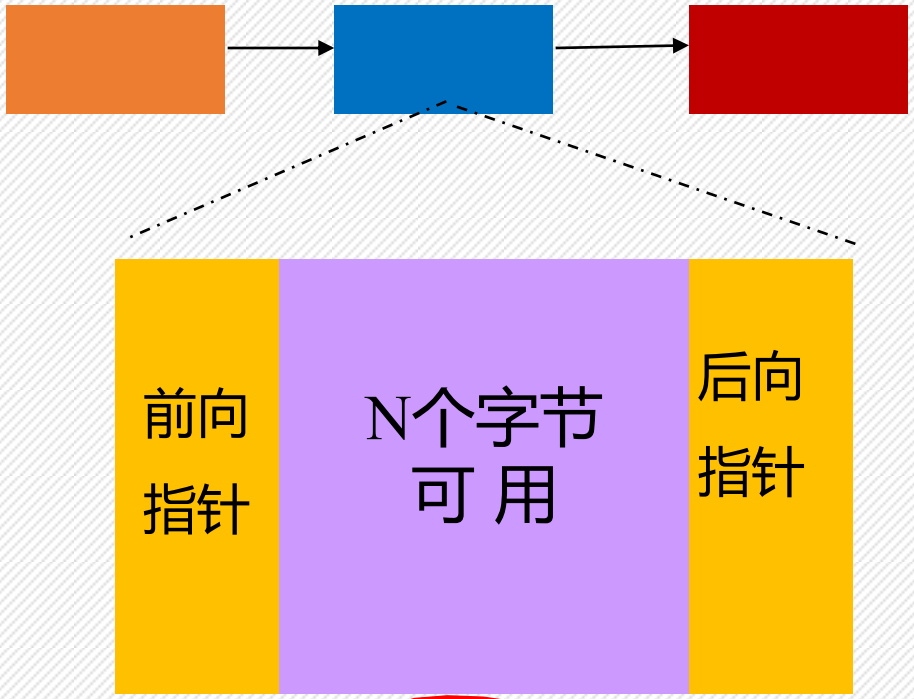
Job queue

process	memory	time
P_1	600K	10
P_2	1000K	5
P_3	300K	20
P_4	700K	8
P_5	500K	15

空闲分区的管理

序号	大小	起址	状态
1	48K	116K	空闲
2	252K	260K	空闲
3	---	---	空表目
4	---	---	空表目
5	---	---	空表目

空闲分区表



空闲分区链

可变分区分配

可变分区分配

分区分配算法

寻找某个空闲分区，其大小需大于或等于程序的要求。若是大于要求，则将该分区分割成两个分区，其中一个分区为要求的大小并标记为“占用”，而另一个分区为余下部分并标记为“空闲”。分区的先后次序通常是从内存低端到高端。

分区释放算法

需要将相邻的空闲分区合并成一个空闲分区。(这时要解决的问题是：合并条件的判断)

Dynamic Storage-Allocation Problem

- How to satisfy a request of size n from a list of free holes.
  (怎样从一个空的分区序列中满足一个申请需要。)

01 First-fit（首先适应）：Allocate the *first* hole that is big enough.（分配最先找到的合适的分区。）

02 Best-fit（最佳适应）：Allocate the *smallest* hole that is big enough; must search entire list, unless ordered by size. Produces the smallest leftover hole.（搜索整个序列，找到适合条件的最小的分区进行分配。）

03 Worst-fit（最差适应）：Allocate the *largest* hole; must also search entier list. Produces the largest leftover hole.（搜索整个序列，寻找最大的分区进行分配。）

- First-fit and best-fit better than worst-fit in terms of speed and storage utilization.（在速度和存储的利用上，首先适应和最佳适应要比最差适应好。）

首次适应算法（First Fit）

从空闲分区表的第一个表目开始查找，把找到的第一个满足要求的空闲区分配给作业，目的在于减少查找时间。通常将空闲分区表（空闲区链）中的空闲分区要按地址由低到高进行排序。

特点1

分配和释放的时间性能较好，较大的空闲分区可以被保留在内存高端。

特点2

随着低端分区不断划分而产生较多小分区，每次分配时查找时间开销会增大。

特点3

在系统不断地分配和回收中，必定会出现一些不连续的小的空闲区，称为外碎片。虽然可能所有碎片的总和超过某一个作业的要求，但是由于不连续而无法分配。

最佳适应算法（Best Fit）

- 从全部空闲区中找出能满足作业要求的、且最小的空闲分区.
- 能使碎片尽量小
- 为提高查找效率，空闲分区表（空闲区链）中的空闲分区要按从小到大进行排序，自表头开始查找到第一个满足要求的自由分区分配
- 特点

从个别来看，外碎片较小，
但从整体来看，
会形成较多无法利用的碎片。

特点1

特点2

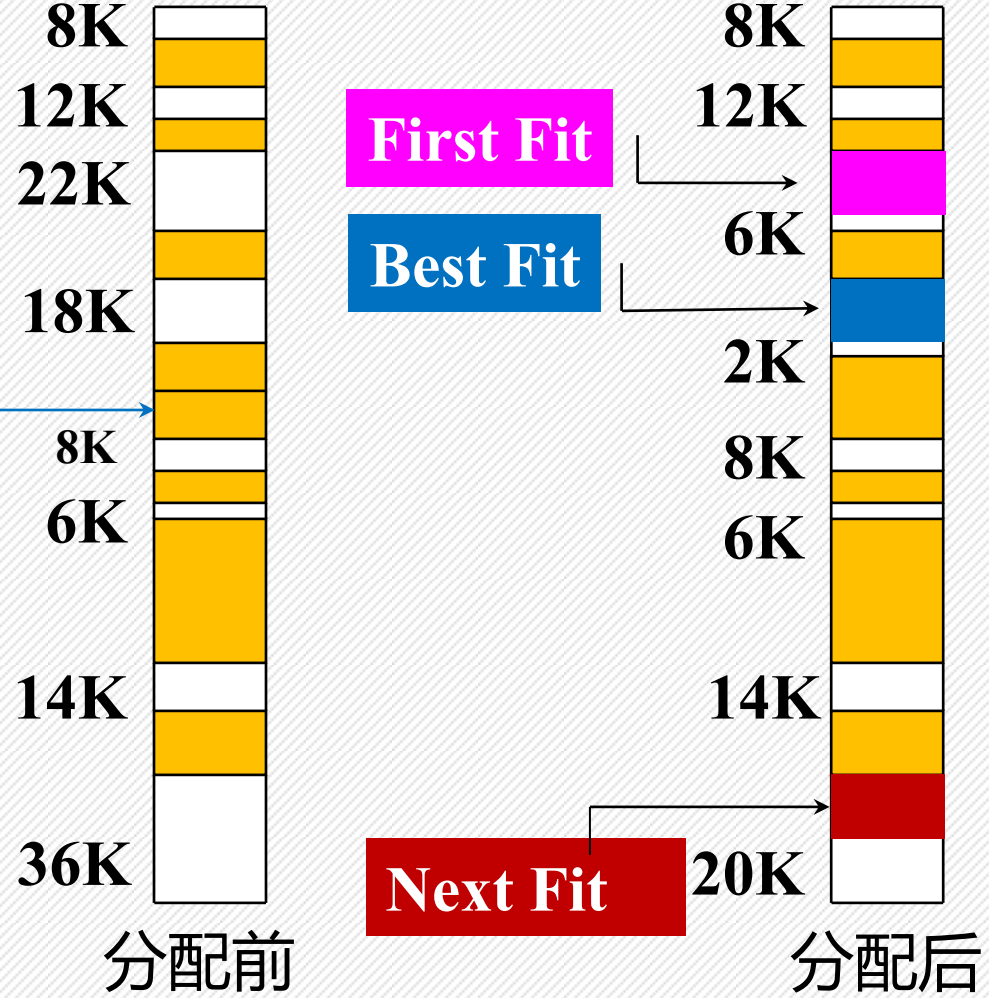
较大的空闲分区可以被保留。

动态分区分配算法举例

例：需分配一个16KB分区

Allocated block
已分配区域

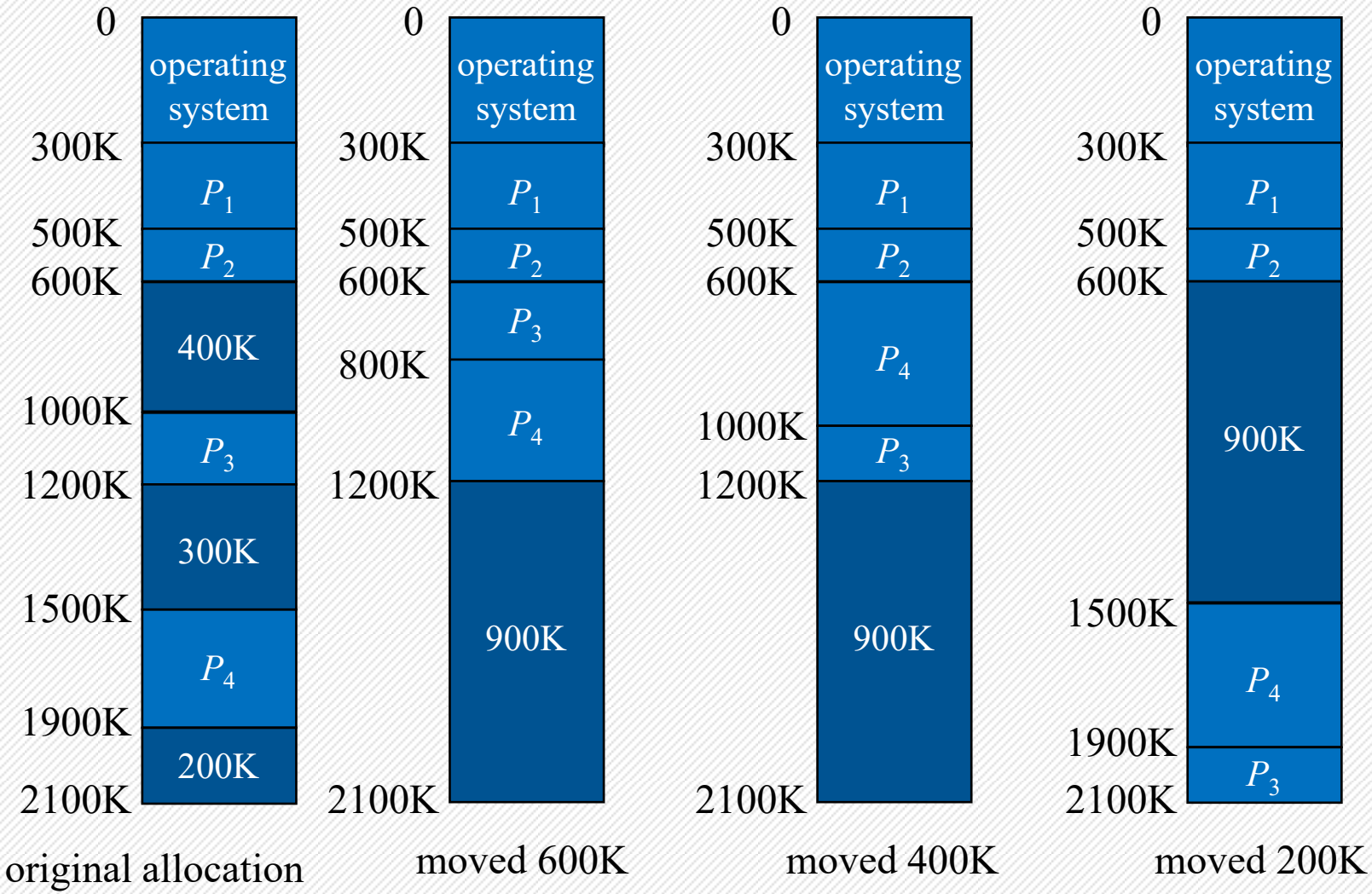
Free block
空闲区域



动态分区分配算法举例

- 可变式分区的碎片问题:在系统不断地分配和回收中，必定会出现一些不连续的小的空闲区，称为外碎片。虽然可能所有碎片的总和超过某一个作业的要求，但是由于不连续而无法分配。
- 解决碎片的方法是拼接（或称紧凑），即向一个方向（例如向低地址端）移动已分配的作业，使那些零散的小空闲区在另一方向连成一片。
- 分区的拼接技术，一方面要求能够对作业进行动态重定位，另一方面系统在拼接时要耗费较多的时间。

Contiguous Allocation (Cont.)



Fragmentation

- External fragmentation（外碎片） – total memory space exists to satisfy a request, but it is not contiguous.
- Internal fragmentation（内碎片） – allocated memory may be slightly larger than requested memory; this size difference is memory internal to a partition, but not being used.
- Reduce external fragmentation by compaction（通过紧缩来减少外碎片）Shuffle memory contents to place all free memory together in one large block.（把一些小的空闲内存结合成一个大的块。）
  - Compaction is possible *only* if relocation is dynamic, and is done at execution time.（只有重定位是动态的时候，才有可能进行紧缩，紧缩在执行时期进行）
  - I/O problem（I/O问题）
    1. Latch job in memory while it is involved in I/O.（当I/O的时候，把工作锁定在内存中。）
    2. Do I/O only into OS buffers.（只对操作系统的缓冲区进行I/O。）