

操作系统

Operating Systems

L21 内存分区与分页

Memory Partition and Paging

授课教师：李治军

lizhijun_os@hit.edu.cn

综合楼411室

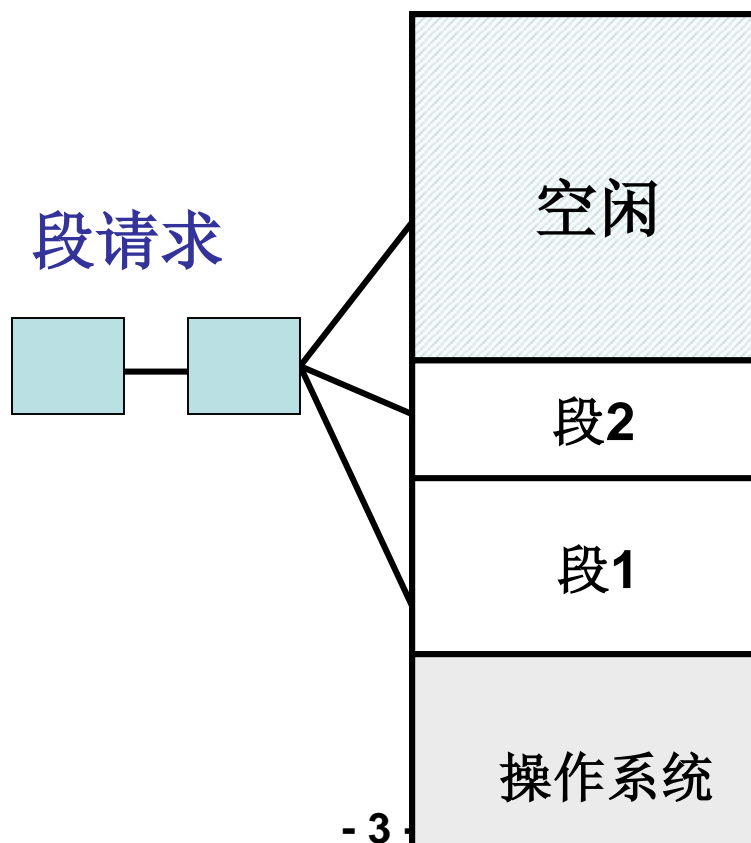
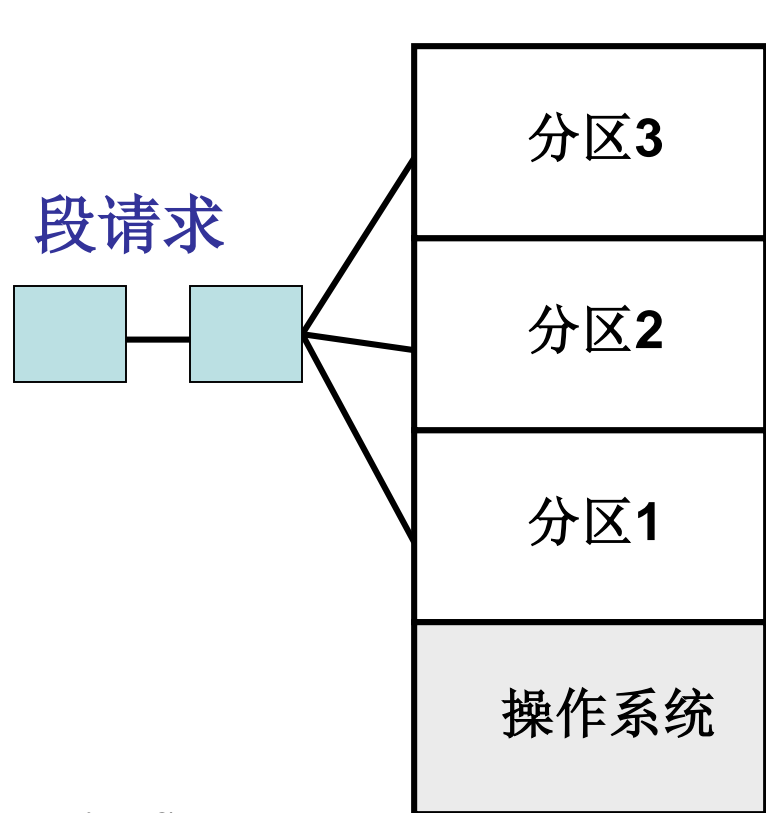
接下来的问题是内存怎么割？

这样就可以将程序的各个段载入到相应的内存分区中了

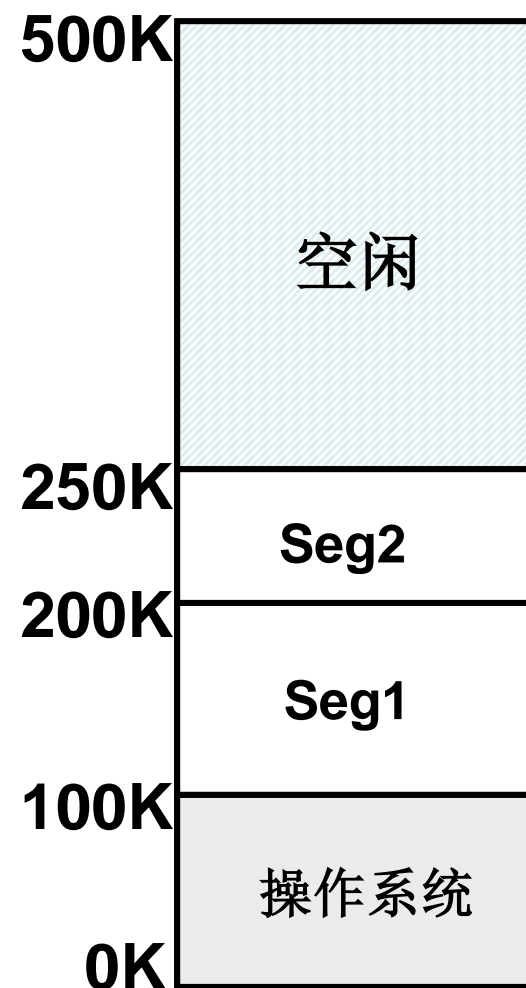


固定分区 与 可变分区

- 给你一个面包，一堆孩子来吃，怎么办？
 - 等分，操作系统初始化时将内存等分成 k 个分区
 - 但孩子有大有小，段也有大有小，需求不一定



可变分区的管理过程 — 核心数据结构



空闲分区表

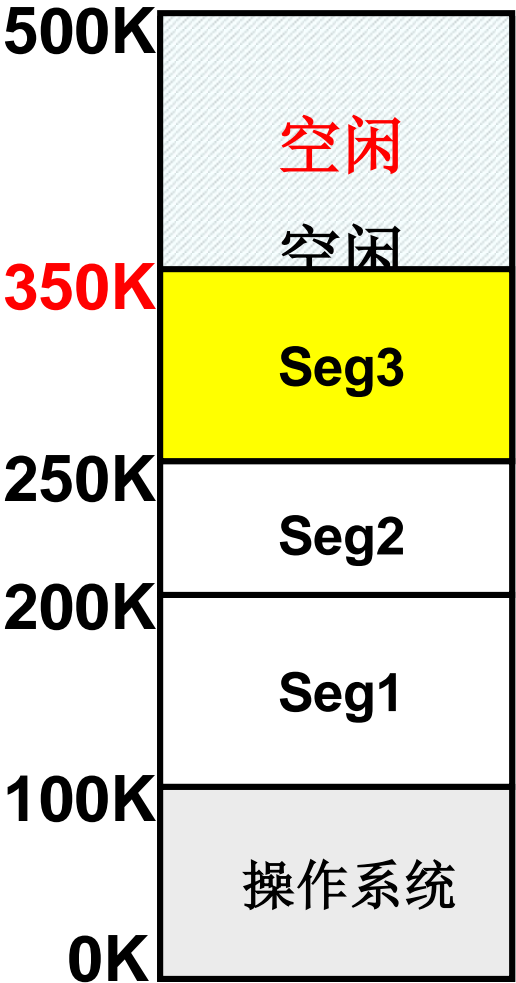
始址	长度
250K	250K

已分配分区表

始址	长度	标志
0K	100K	OS
100K	100K	Seg1
200K	50K	Seg2



可变分区的管理—请求分配



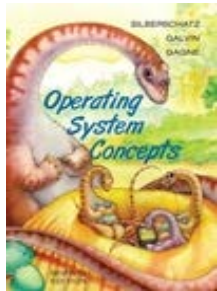
■ 段内存请求: reqSize = 100K

空闲分区表

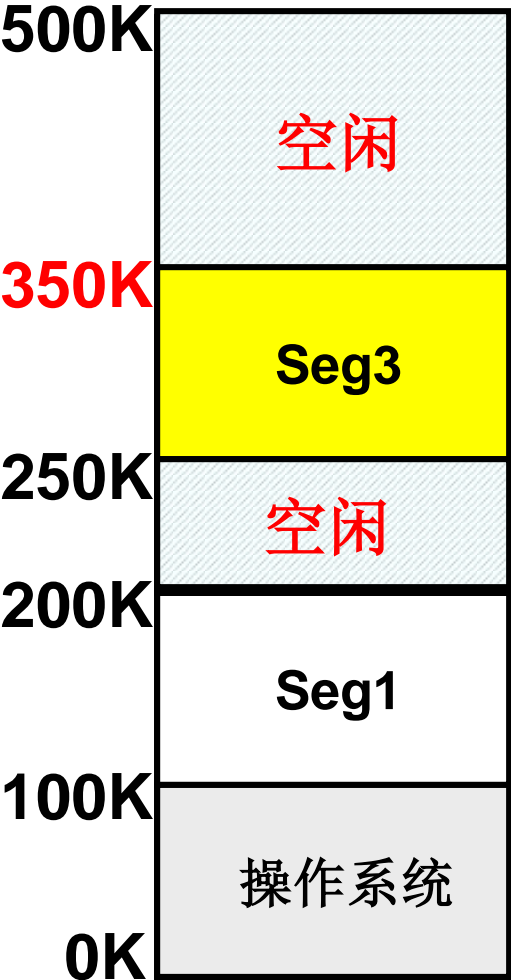
始址	长度
250K	250K

已分配分区表

始址	长度	标志
0K	100K	OS
100K	100K	Seg1
200K	50K	Seg2
250K	350K	Seg3



可变分区的管理—释放内存



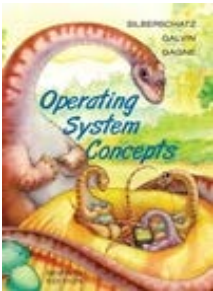
■ 段2不再需要，释放内存

空闲分区表

始址	长度
350K	150K
200K	50K

已分配分区表

始址	长度	标志
0K	100K	OS
100K	100K	Seg1
200K	100K	Seg2
250K	100K	Seg3



可变分区的管理—再次申请

■ 又一个段提出内存请求: **reqSize=40K**, 怎么办?

■ 有2个空闲分区, 选哪一个?

■ 首先适配: (350,150) 空闲分区表

■ 最佳适配: (200,50)

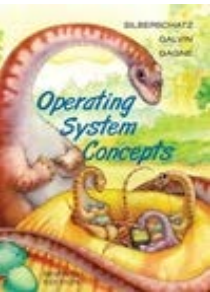
■ 最差适配: (350,150)

始址	长度
350K	150K
200K	50K

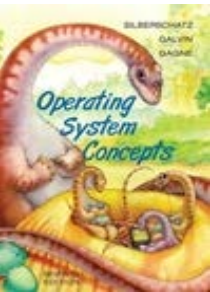


问题：如果某操作系统中的段内存请求很不规则，有时候需要很大的一个内存块，有时候又很小，此时用哪种分区分配算法最好？()

- A. 最先适配
- B. 最佳适配
- C. 最差适配
- D. 没有区别



引入分页：解决内存分区导致的内存效率问题



可变分区造成的问题

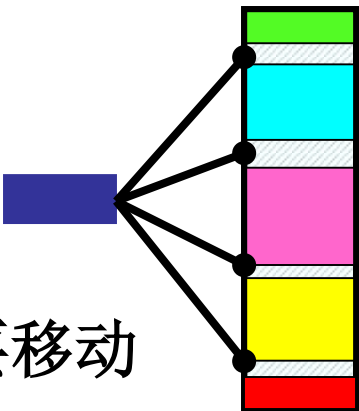
■ 发起请求reqSize=160K怎么办？

■ 总空闲空间>160，但没有一个空闲分区>160，怎么办？

■ 这就是内存碎片

■ 将空闲分区合并，需要移动
1个段(复制内容)：内存紧缩

■ 内存紧缩需要花费大量时间，如
果复制速度1M/1秒，则1G内存
的紧缩时间为1000秒≈17分钟



空闲分区表

始址	长度
350K	150K
200K	50K

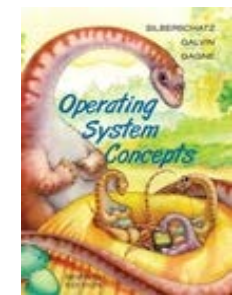


从连续到离散...

- 让给面包没有谁都不想要的碎末
 - 将面包切成片，将内存分成页
 - 针对每个段内存请求，系统一页一页的分配给给这个段

问题：此时需要内存紧缩吗？最大的内存浪费是多少？

页框7	
页框6	段0：页3
页框5	段0：页0
页框4	
页框3	段0：页2
页框2	
页框1	段0：页1
页框0	



页已经载入了内存，接下来的事情...

■ 页0放在页框5中，页0中的地址就需要重定位

页中的仍然是逻辑地址



```
mov [0x2240], %eax
```

