

运行分三步：

1. 编译时期将程序分成多个段
2. 在内存划分出空闲区域
3. 磁盘读写，将程序载入内存
4. 设定PCB表，将其关联起来

哈工大计算机科学与技术学院

软件基础教研室

操作系统

Operating Systems

程序分段是  
编译时期完成的

LDT表

1	1000
0	3000

## L21 内存分区与分页

### Memory Partition and Paging

进程切换的时候，  
对应的映射表也需要切换

LDTR寄存器

授课教师：李治军

lizhijun\_os@hit.edu.cn

综合楼411室

LDTR寄存器存储  
LDT表的起始地址

CS:40 = 0:40 → 3040

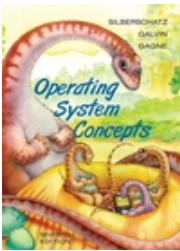
然后CS相当于存储了一个索引  
通过其实地址和索引，可以找到对应的表项  
表项中存储的有这个段的起始地址。  
将其实地址和偏移相加，就可以得到物理地址

---

# 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

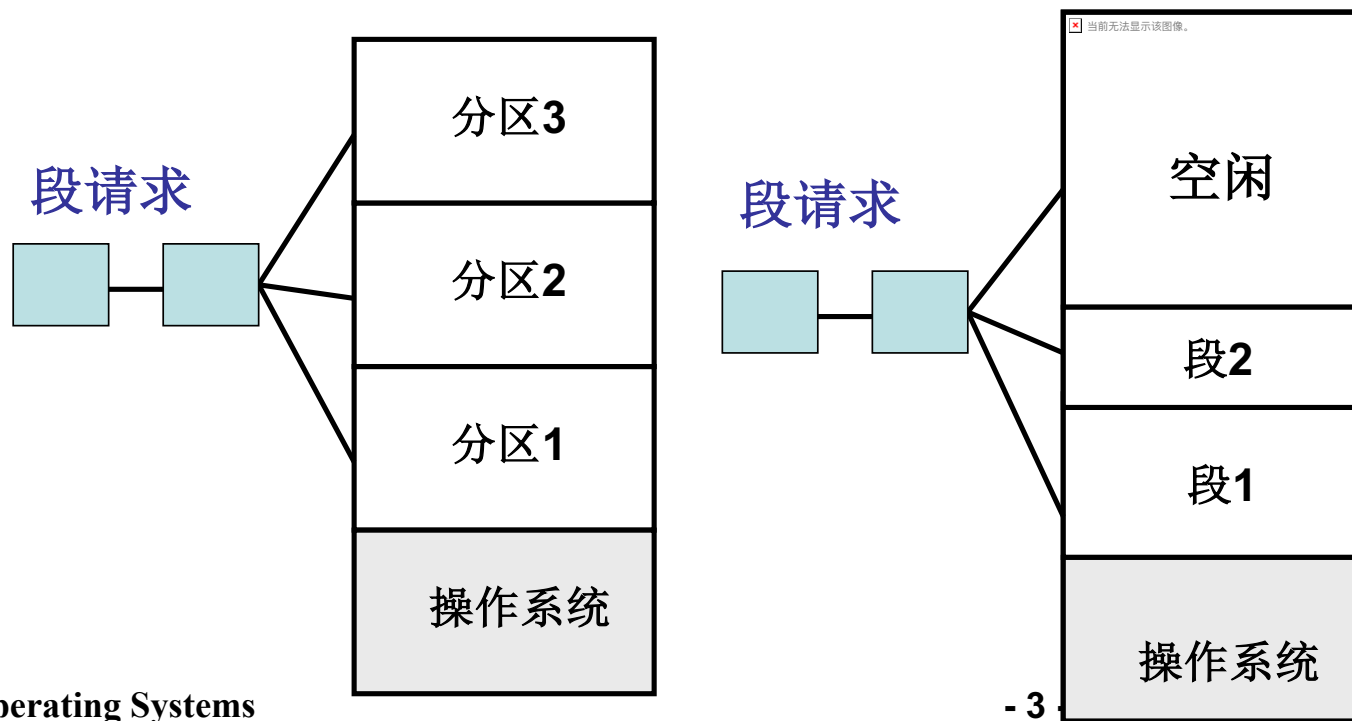
使用数据结构和对应的算法实现



# 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

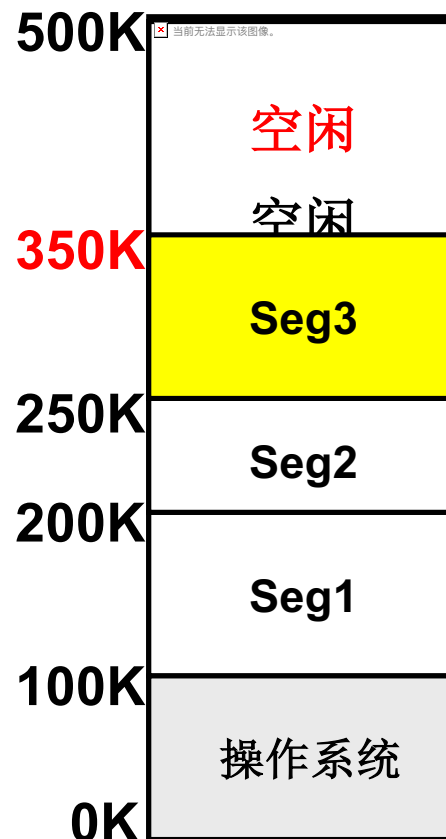
始址	长度
250K	250K

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



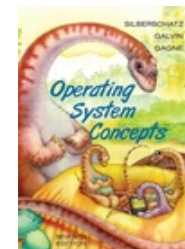
■ 段内存请求: reqSize = 100K

空闲分区表

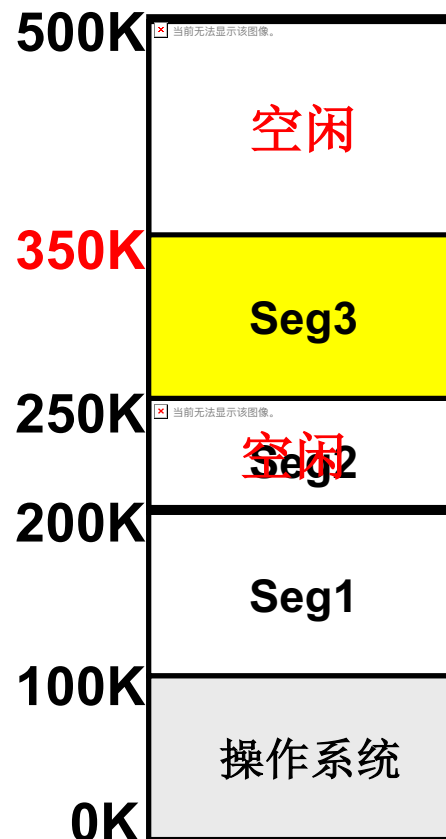
始址	长度
250K	250K

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

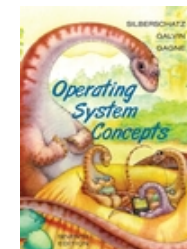
■ 最佳适配: (200,50)

■ 最差适配: (350,150)

始址	长度
350K	150K
200K	50K

最差适配, 碎片比较均匀

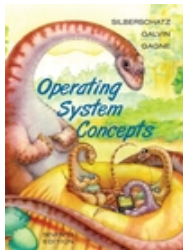
最佳适配, 会导致比较多的小碎片



---

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？(B)

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

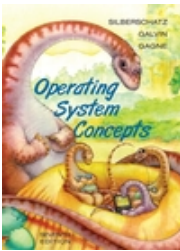




---

# 引入分页：解决内存分区导致的内存效率问题

分段对应虚拟内存  
虚拟内存对应到物理内存使用分页的技术

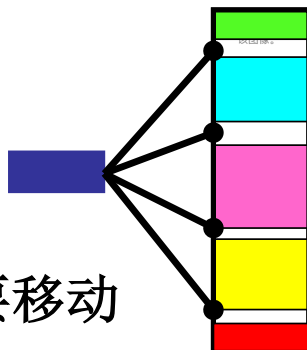


# 可变分区造成的问题

## ■ 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



空闲分区表

始址	长度
350K	150K
200K	50K

- 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



物理内存希望分页  
用户希望分段  
两部分都支持

## 从连续到离散...

页框7

页框6

页框5

页框4

页框3

页框2

页框1

页框0

<input checked="" type="checkbox"/> 当前无法显示该图像。
段0: 页3
段0: 页0
<input checked="" type="checkbox"/> 当前无法显示该图像。
段0: 页2
<input checked="" type="checkbox"/> 当前无法显示该图像。
段0: 页1
<input checked="" type="checkbox"/> 当前无法显示该图像。

### ■ 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

按页分  
配

问题：此时需要内存紧缩吗？最大的内存浪费是多少？

碎片只会在页内存在  
一个段最多也就只会浪费4k

mem\_map 4k一块



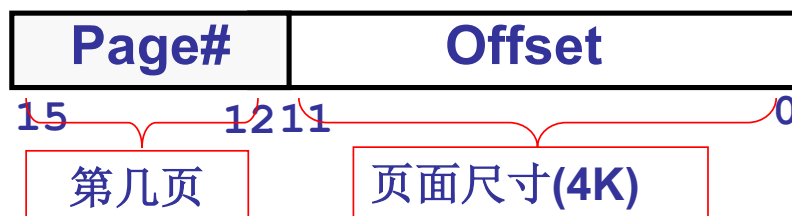
# 页已经载入了内存，接下来的事情...

■ 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

这些操作都是由  
硬件做的MMU

`mov [0x2240], %eax`



逻辑地址

0x02

0x240

页表指针

PCB中应  
有此值

页号	页框号	保护
0	5	R
1	1	R/W
2	3	R/W
3	6	R

3

240

物理地址: 0x3240

权限检查

CR3寄存器

每个进程有自己的段表  
也有自己的页表

页表基址存储在CR3寄存器中

页框7

页框6

页框5

页框4

页框3

页框2

页框1

页框0

段0: 页3

段0: 页0

段0: 页2

段0: 页1

