

说明：本视频对应王道书 4.3.3

在前面的视频中，我们学习了王道书“4.1.6 文件的物理结构”，解释了“文件数据如何在外存中存放”的问题。

而王道书“4.3.3 外存空闲空间管理”主要介绍“外存中没有存放文件数据的空闲空间该如何管理”的问题。因此在本课程中，我们把 4.1.6、4.3.3 连起来学习。

建议：学完本视频，可以接着阅读王道书 4.3.3。

本节内容

文件存储空 间管理

知识总览



操作系统需要对磁盘块进行哪些管理

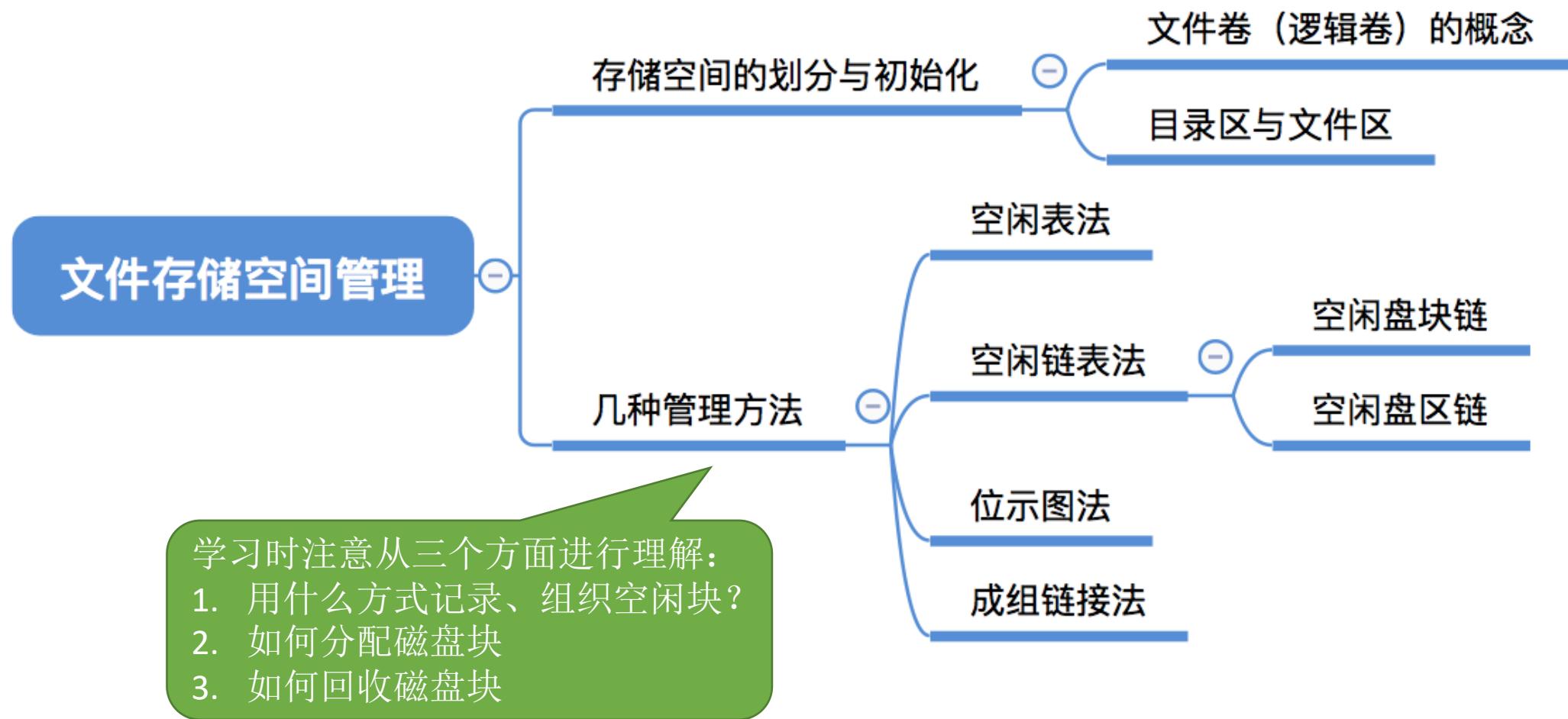
对非空闲磁盘块的管理（存放了文件数据的磁盘块）

对空闲磁盘块的管理

“文件存储空间管理”要探讨的问题

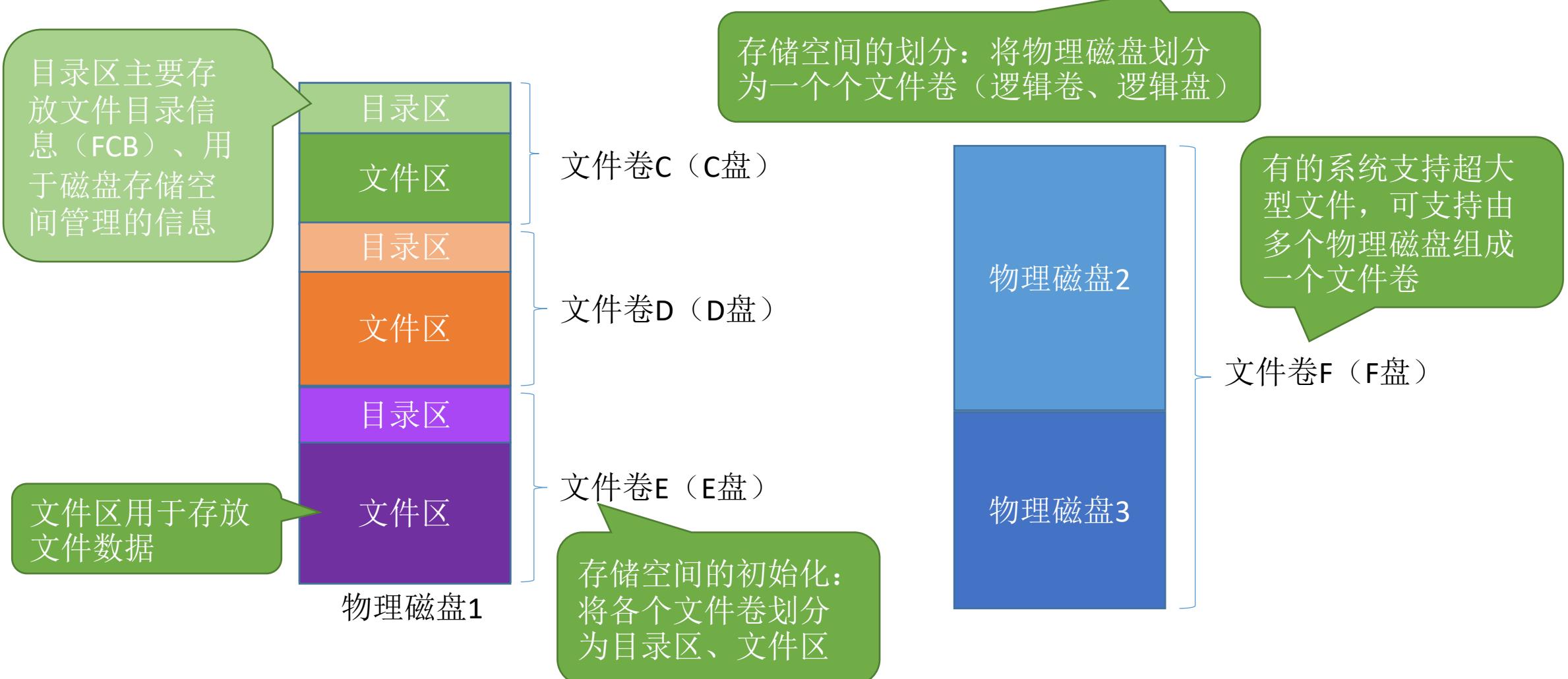
“文件的物理结构/文件分配方式”要探讨的问题——连续分配、链接分配、索引分配

知识总览



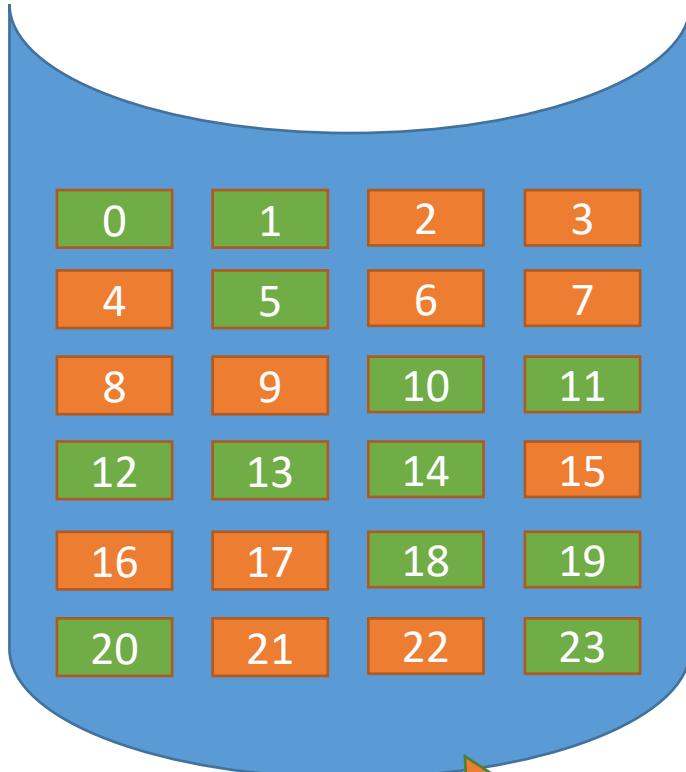
存储空间的划分与初始化

安装 Windows 操作系统的时候，一个必经步骤是——为磁盘分区（C: 盘、D: 盘、E: 盘等）



存储空间管理——空闲表法

适用于“连续分配方式”



| 第一个空闲盘块号 | 空闲盘块数 |
|----------|-------|
| 0 | 2 |
| 5 | 1 |
| 13 | 2 |
| 18 | 3 |
| 23 | 1 |

空闲盘块表

Eg: 新创建的文件
请求3个块, 采用
首次适应算法

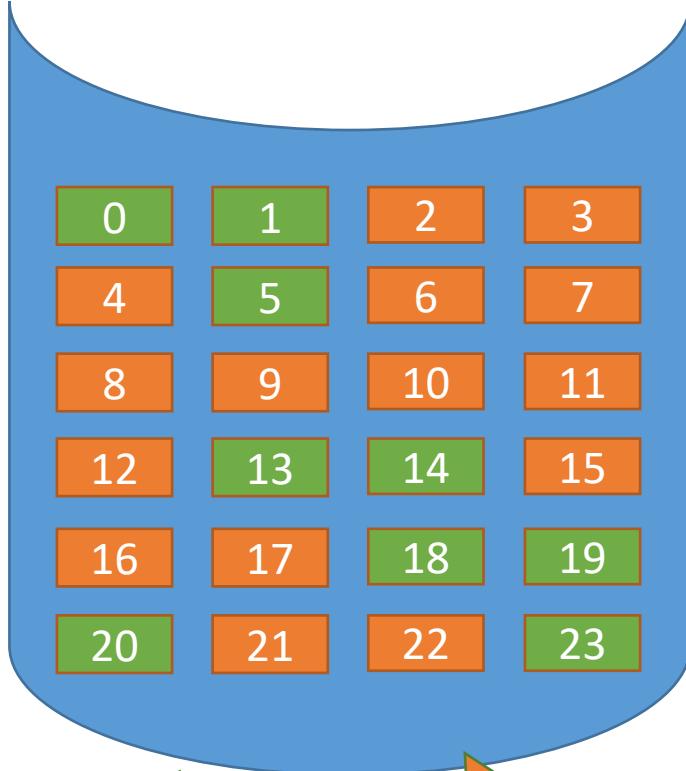
如何分配磁盘块: 与内存管理中的动态分区分配很类似, 为一个文件
分配连续的存储空间。同样可采用首次适应、最佳适应、最坏适应等
算法来决定要为文件分配哪个区间。

绿色为
空闲块

橙色为
非空闲
块

存储空间管理——空闲表法

适用于“连续分配方式”



绿色为
空闲块

橙色为
非空闲
块

| 第一个空闲盘块号 | 空闲盘块数 |
|----------|-------|
| 0 | 2 |
| 5 | 1 |
| 13 | 8 |
| 23 | 1 |

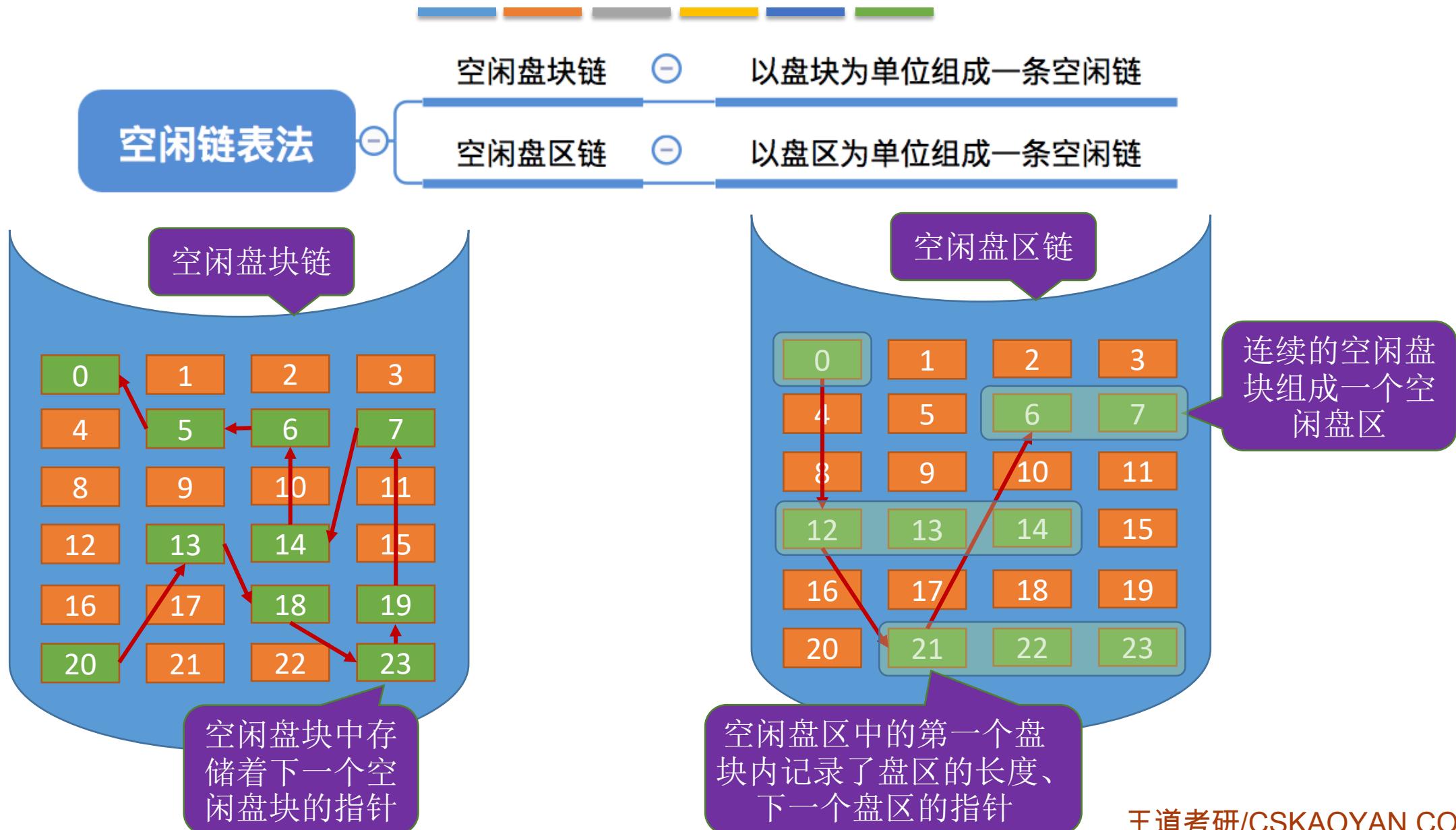
空闲盘块表

如何分配磁盘块：与内存管理中的动态分区分配很类似，为一个文件分配连续的存储空间。同样可采用首次适应、最佳适应、最坏适应等算法来决定要为文件分配哪个区间。

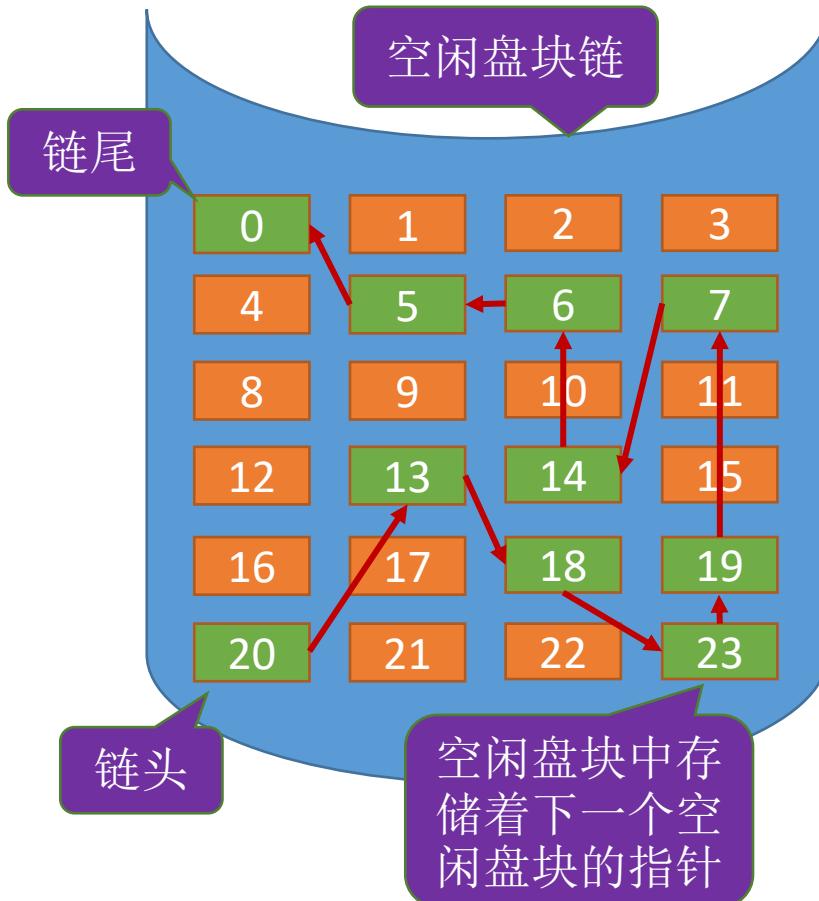
如何回收磁盘块：与内存管理中的动态分区分配很类似，当回收某个存储区时需要有四种情况——①回收区的前后都没有相邻空闲区；②回收区的前后都是空闲区；③回收区前面是空闲区；④回收区后面是空闲区。总之，回收时需要注意表项的合并问题。

情况② Eg: 假设此时删除了某文件，系统回收了它占用的 15、16、17 号块

存储空间管理——空闲链表法



存储空间管理——空闲链表法



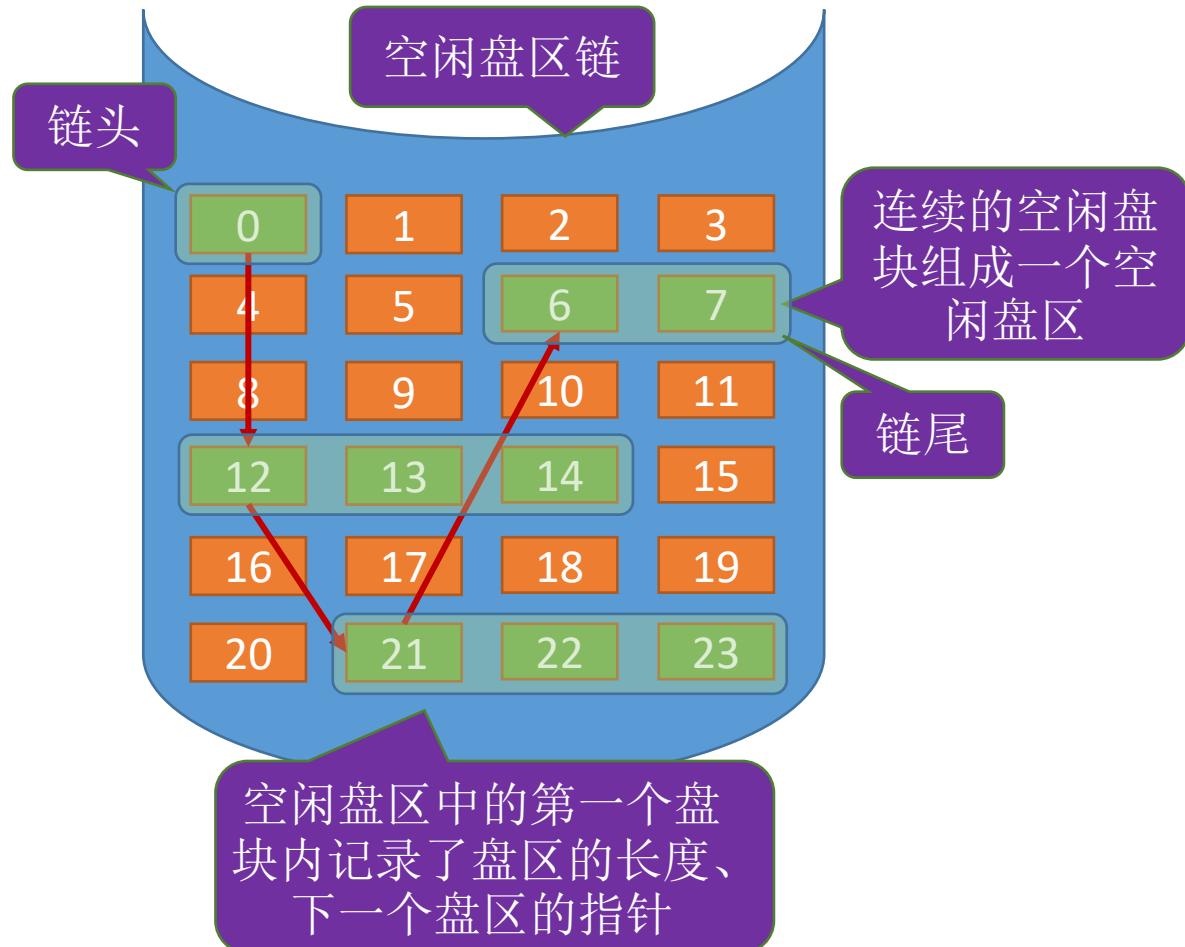
操作系统保存着链头、链尾指针。

如何分配：若某文件申请 k 个盘块，则从链头开始依次摘下 k 个盘块分配，并修改空闲链的链头指针。

如何回收：回收的盘块依次挂到链尾，并修改空闲链的链尾指针。

适用于离散分配的物理结构。为文件分配多个盘块时可能要重复多次操作

存储空间管理——空闲链表法



操作系统保存着**链头、链尾指针**。

如何分配：若某文件申请 K 个盘块，则可以采用首次适应、最佳适应等算法，从链头开始检索，按照算法规则找到一个大小符合要求的空闲盘区，分配给文件。若没有合适的连续空闲块，也可以将不同盘区的盘块同时分配给一个文件，注意分配后可能要修改相应的链指针、盘区大小等数据。

如何回收：若回收区和某个空闲盘区相邻，则需要将回收区合并到空闲盘区中。若回收区没有和任何空闲区相邻，将回收区作为单独的一个空闲盘区挂到链尾。

离散分配、连续分配都适用。为一个文件分配多个盘块时效率更高

$$(0,1) \rightarrow b=16*0+1=1$$

$$(1,10) \rightarrow b=16*1+10=26$$

$$b=13 \rightarrow i=13/16=0, j=13\%16=13$$

$$b=31 \rightarrow i=31/16=1, j=31\%16=15$$

| | | | |
|----|----|-------|----|
| 0 | 1 | 2 | 3 |
| 4 | 5 | 6 | 7 |
| 8 | 9 | 10 | 11 |
| 12 | 13 | 14 | 15 |
| 16 | 17 | 18 | 19 |
| 20 | 21 | 22 | 23 |
| 24 | 25 | 26 | 27 |
| 28 | 29 | 30 | 31 |
| 32 | 33 | | |

存储空间管理——位示图法

| 字号 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|-----|---|---|-----|---|---|---|---|---|---|---|----|----|----|----|----|----|
| 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 2 | 1 | 1 | ... | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | |

位示图：每个二进制位对应一个盘块。在本例中，“0”代表盘块空闲，“1”代表盘块已分配。位示图一般用连续的“字”来表示，如本例中一个字的字长是16位，字中的每一位对应一个盘块。因此可以用（字号，位号）对应一个盘块号。当然有的题目中也描述为（行号，列号）

重要重要重要：要能自己推出盘块号与（字号，位号）相互转换的公式。

注意题目条件：盘块号、字号、位号到底是从0开始还是从1开始
如本例中盘块号、字号、位号从0开始，若n表示字长，则...

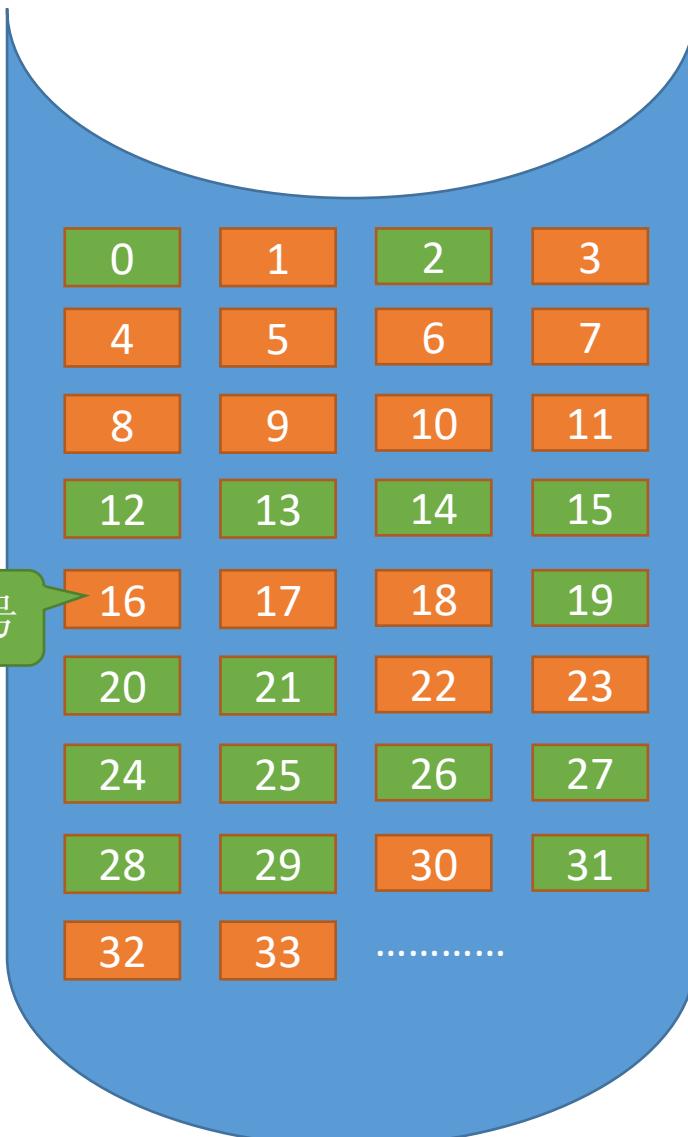
（字号，位号）=（i, j）的二进制位对应的 盘块号 $b = ni + j$

b 号盘块对应的字号 $i = b/n$ ， 位号 $j = b\%n$

注：王道课本来
给出的是盘块号、
字号、位号都从
1 开始的算法。

存储空间管理——位示图法

连续分配、离散分配都适用



| 字号 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|-----|---|---|-----|---|---|---|---|---|---|---|----|----|----|----|----|----|
| 0 | 0 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 |
| 2 | 1 | 1 | ... | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | |

位示图：每个二进制位对应一个盘块。在本例中，“0”代表盘块空闲，“1”代表盘块已分配。

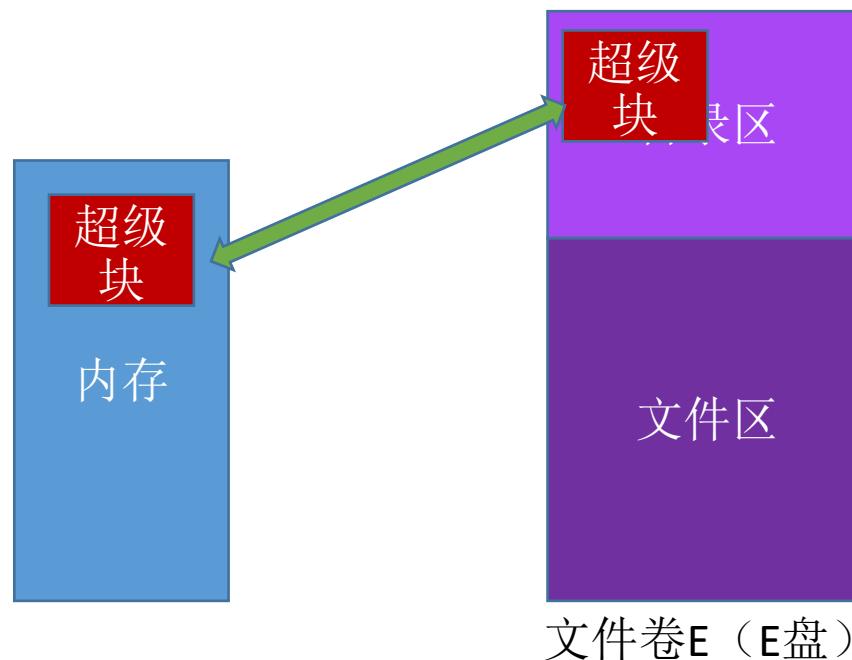
如何分配：若文件需要K个块，①顺序扫描位示图，找到K个相邻或不相邻的“0”；②根据字号、位号算出对应的盘块号，将相应盘块分配给文件；③将相应位设置为“1”。

如何回收：①根据回收的盘块号计算出对应的字号、位号；②将相应二进制位设为“0”

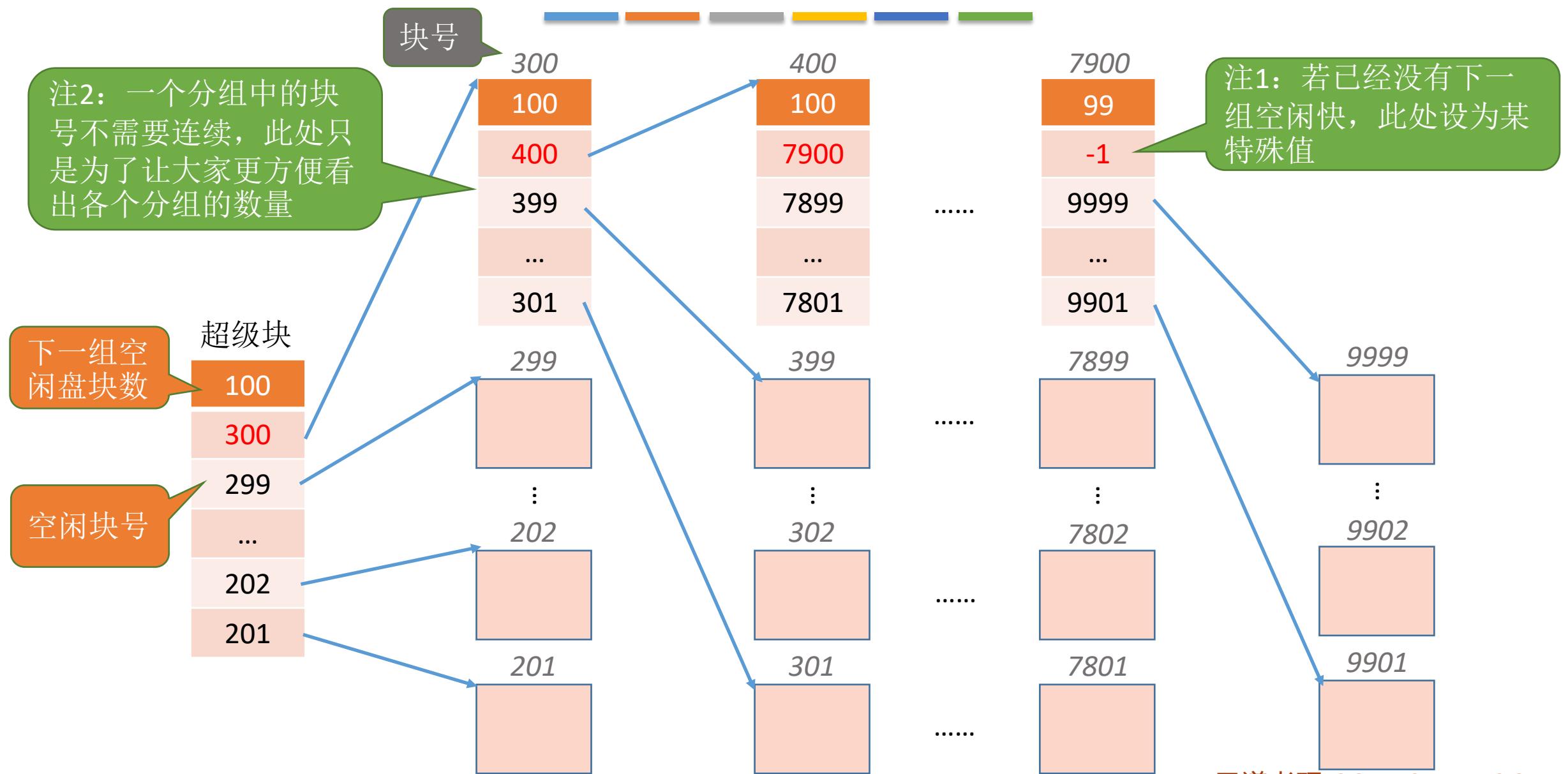
存储空间管理——成组链接法

空闲表法、空闲链表法不适用于大型文件系统，因为空闲表或空闲链表可能过大。UNIX系统中采用了成组链接法对磁盘空闲块进行管理。

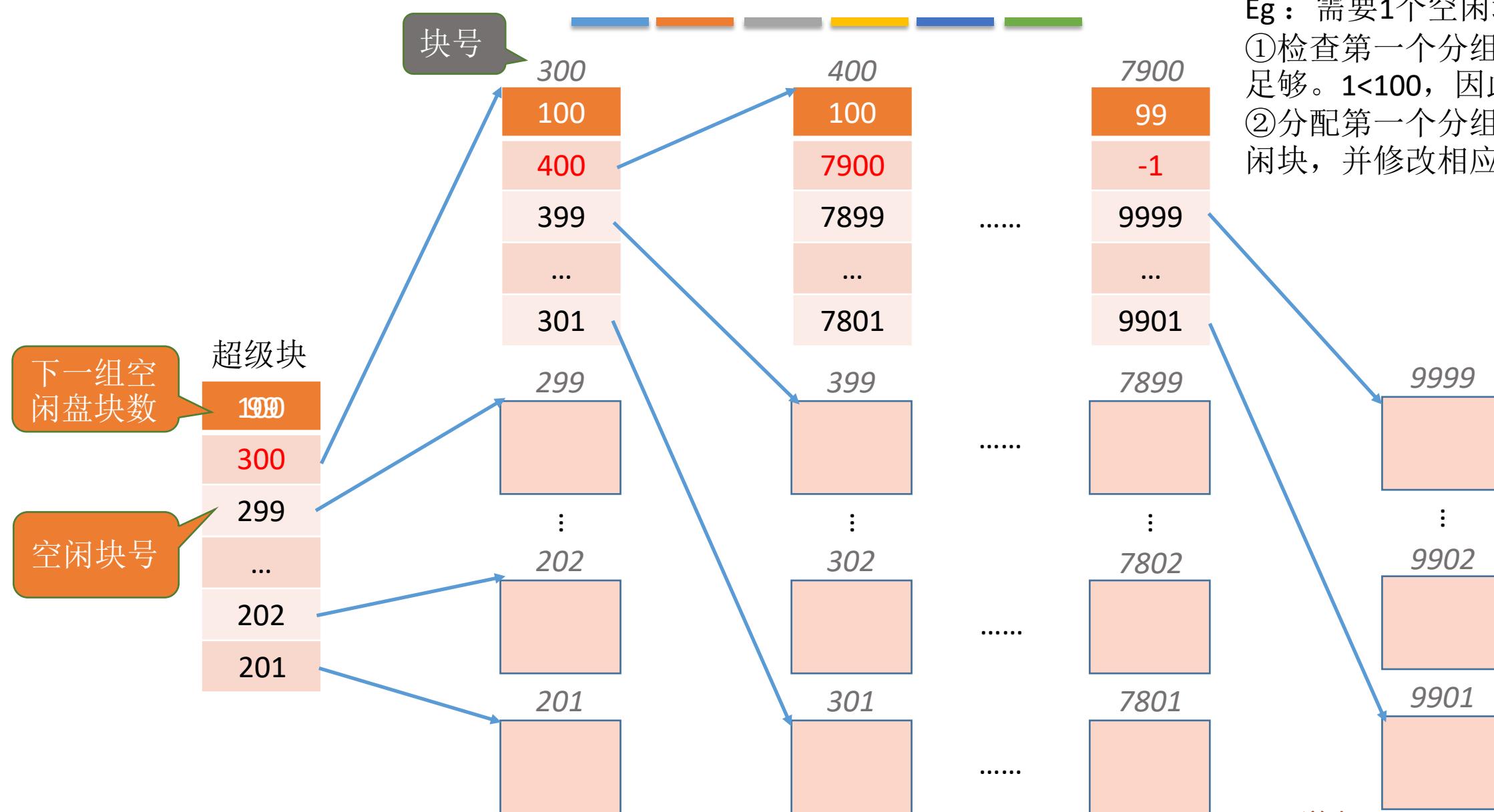
文件卷的目录区中专门用一个磁盘块作为“超级块”，当系统启动时需要将超级块读入内存。并且要保证内存与外存中的“超级块”数据一致。



存储空间管理——成组链接法



存储空间管理——成组链接法

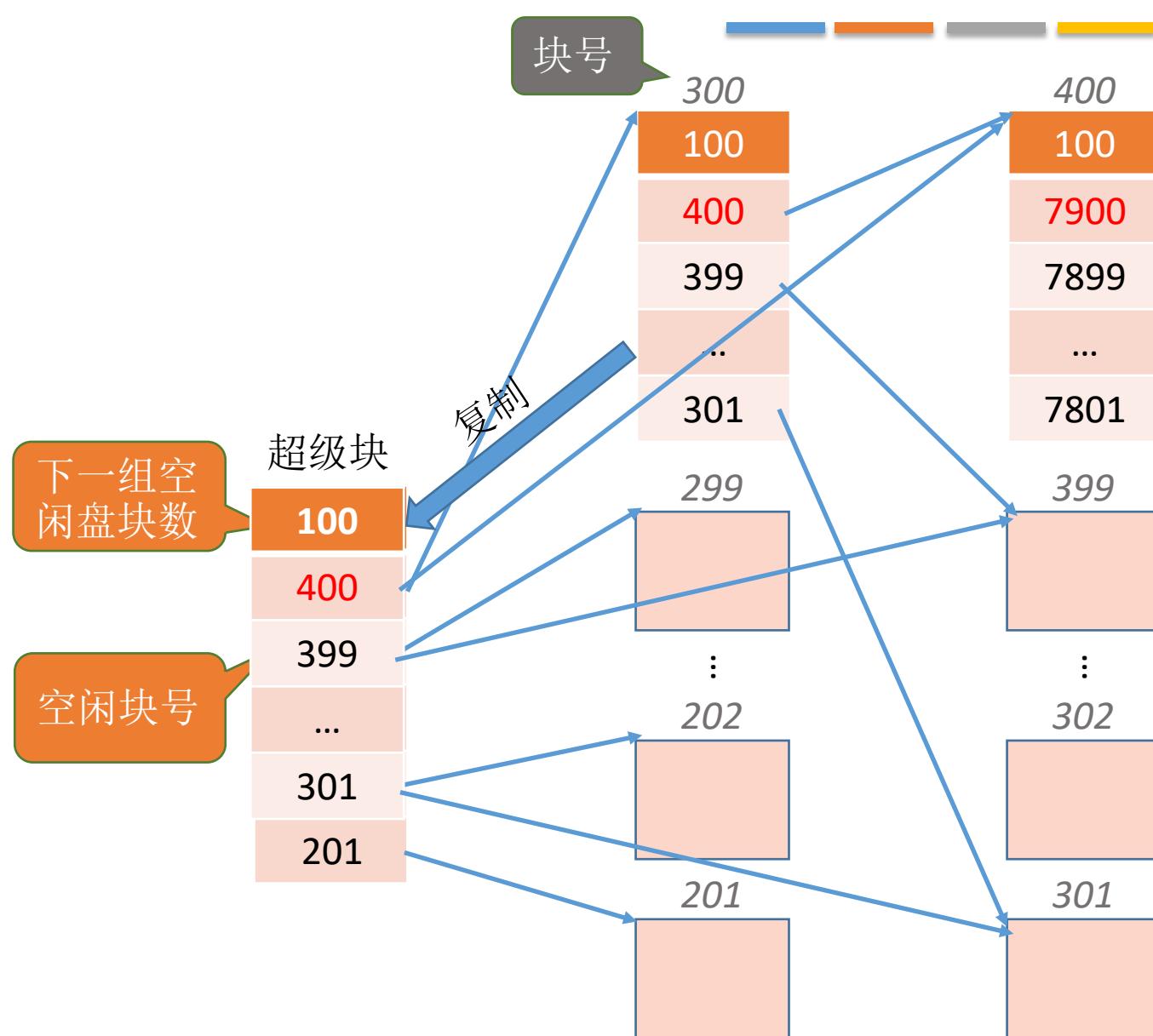


如何分配？

Eg: 需要1个空闲块

- ①检查第一个分组的块数是否足够。 $1 < 100$ ，因此是足够的。
 - ②分配第一个分组中的1个空闲块，并修改相应数据

存储空间管理——成组链接法

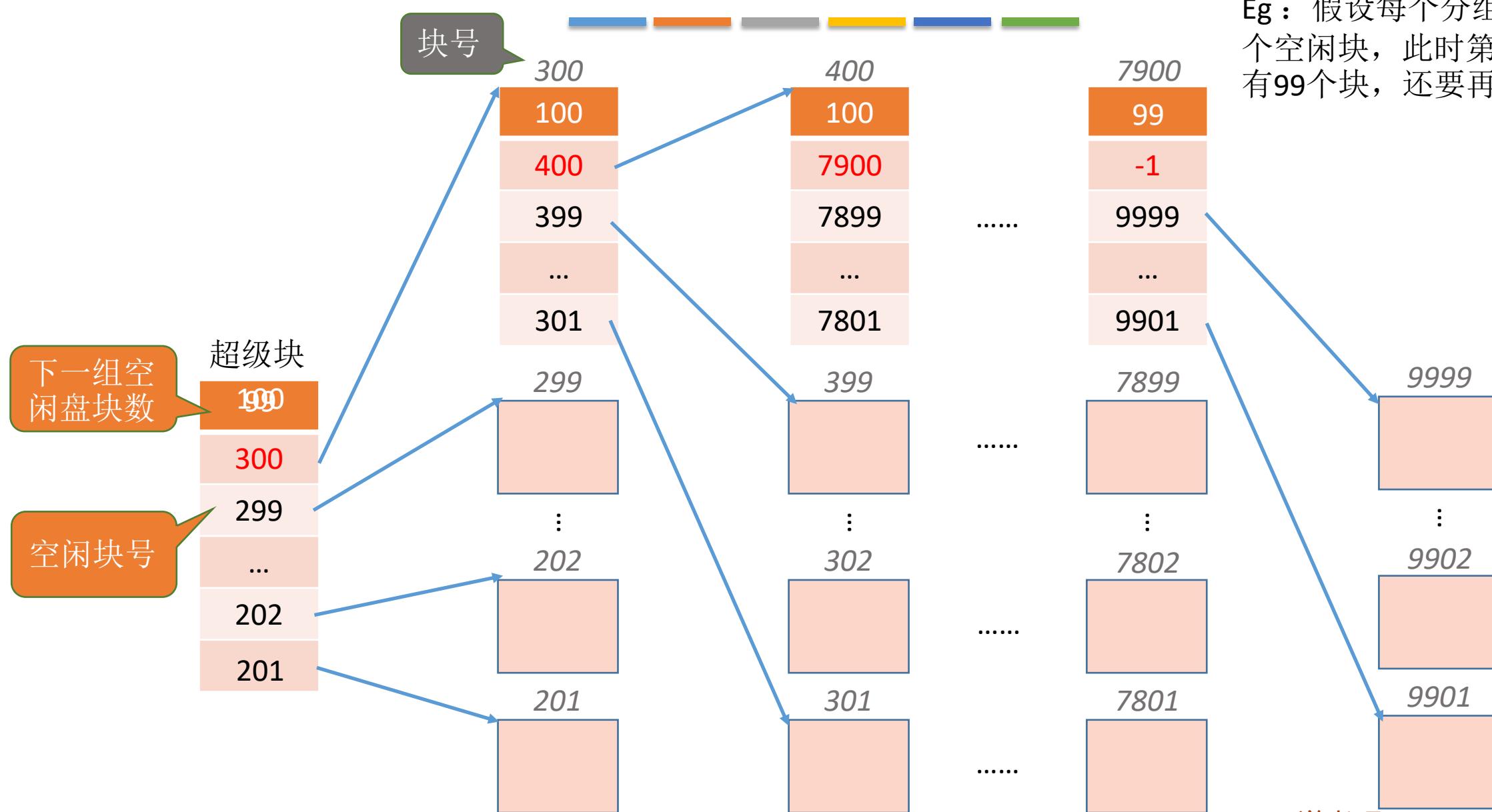


如何分配?

Eg: 需要100个空闲块

①检查第一个分组的块数是否足够。 $100=100$, 是足够的。
②分配第一个分组中的100个空闲块。但是由于300号块内存放了再下一组的信息, 因此300号块的数据需要复制到超级块中。

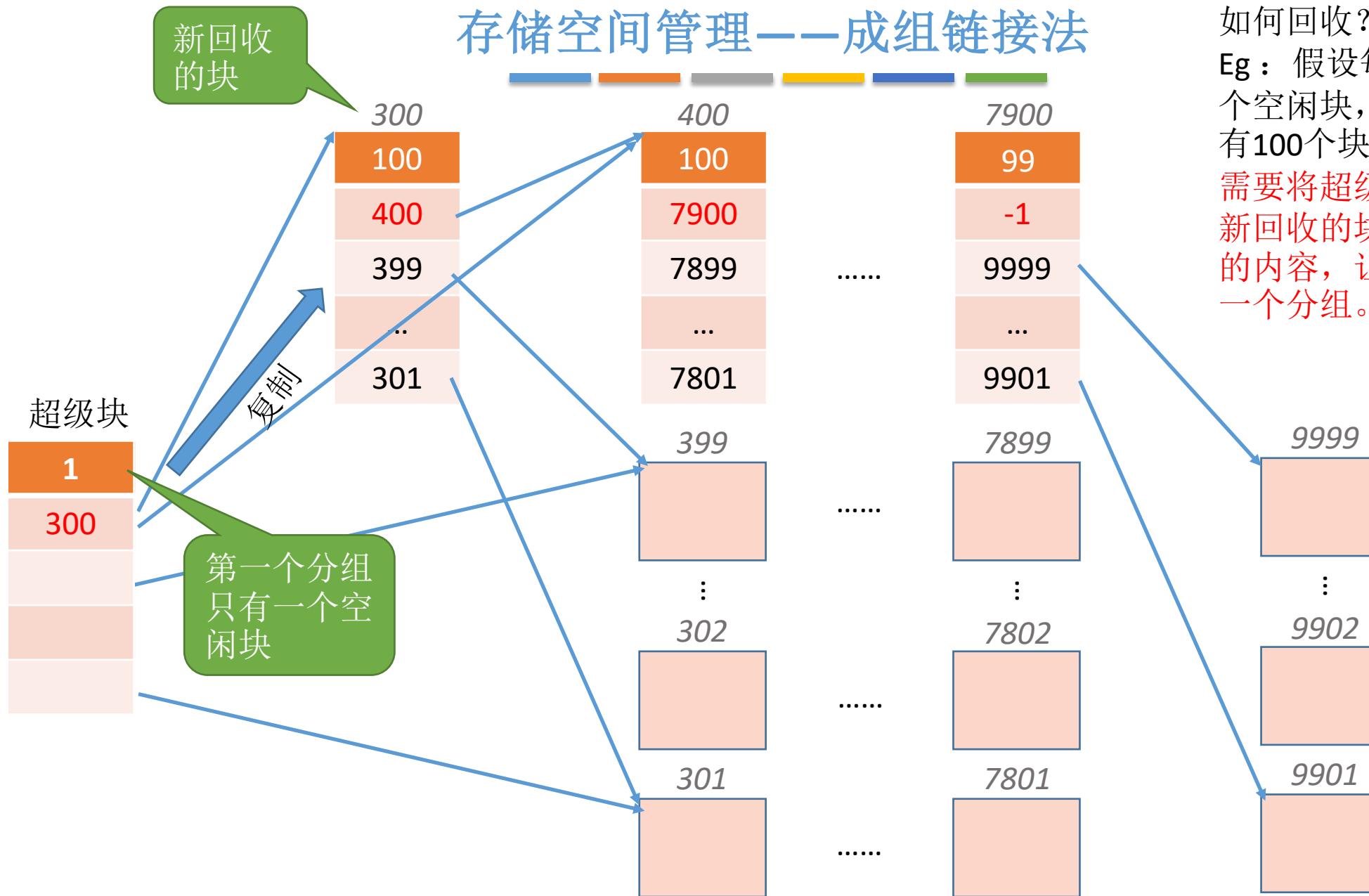
存储空间管理——成组链接法



如何回收?

Eg: 假设每个分组最多为100个空闲块, 此时第一个分组已有99个块, 还要再回收一块

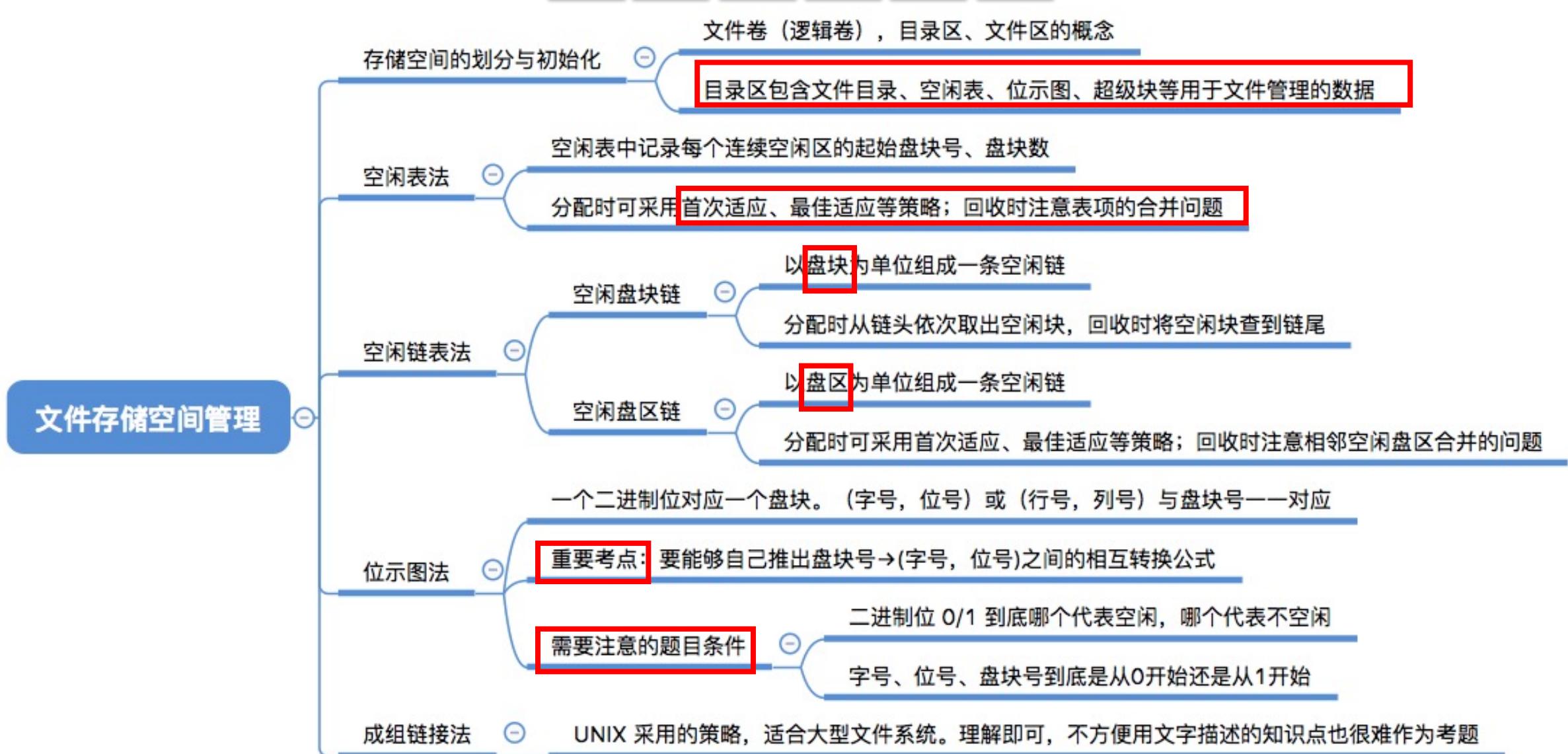
存储空间管理——成组链接法



如何回收?

Eg: 假设每个分组最多为100个空闲块, 此时第一个分组已有100个块, 还要再回收一块。需要将超级块中的数据复制到新回收的块中, 并修改超级块的内容, 让新回收的块成为第一个分组。

知识点回顾与重要考点





@王道论坛



@王道计算机考研备考

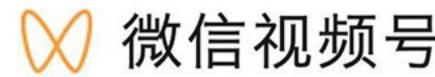
@王道咸鱼老师-计算机考研
@王道楼楼老师-计算机考研



@王道计算机考研



@王道计算机考研



@王道计算机考研



微信公众号平台

@王道在线