



華中農業大學  
HUAZHONG AGRICULTURAL UNIVERSITY

勤讀力耕  
立己達人

# 操作系统原理

**Principle of Operating System**

主讲：倪福川

E-mail: [fcni\\_cn@mail.hzau.edu.cn](mailto:fcni_cn@mail.hzau.edu.cn)

TEL:18602714039

信息学院 大数据科学系



# 第八章 磁盘存储器的管理

- 第一节 外存的组织方式
- 第二节 文件存储空间的管理



## 8.1 外存的组织方式

外存分配方式即文件的物理组织形式。文件的物理结构与外存分配方式有关，采用不同的分配方式时，将形成不同的文件物理结构。

### 分配方式

### 文件物理结构

- 连续分配——顺序式文件结构
- 链接分配——链接式文件结构
- 索引分配——索引式文件结构



## 8.1.1、连续组织方式

### 1、连续分配方式原理：

- 为每一个文件分配一组相邻接的盘块；
- 把逻辑文件中的记录顺序的存储到邻接的各物理盘块中，此时的文件结构称为顺序文件结构。

目录

file	star	length
cou	0	2
tr	14	3
list	18	4
f	6	2





## ●2、外部碎片问题

- 随着空间的分配和回收，磁盘空间会出现一些小的、难以再利用的连续区，称为外存的碎片。
- “碎片”的解决方法——“紧凑”

## ●3、连续分配的优缺点

- 顺序访问容易，并可实现直接存取；
- 顺序访问速度快；（磁头的移动距离最少）
- 缺点：**要求有连续的存储空间（定期做紧凑处理）、必须事先知道文件的长度。



## 8.1.2、链接组织方式

- 将文件装到多个**离散**的盘块中，是离散的分配方式。

- **原理：**

通过在每个盘块上的链接指针，将同属于一个文件的多个离散的盘块链接成一个链表，形成**链接式文件结构**。物理文件称为**链接文件**。

- **类型：**

- 隐式链接
- 显式链接



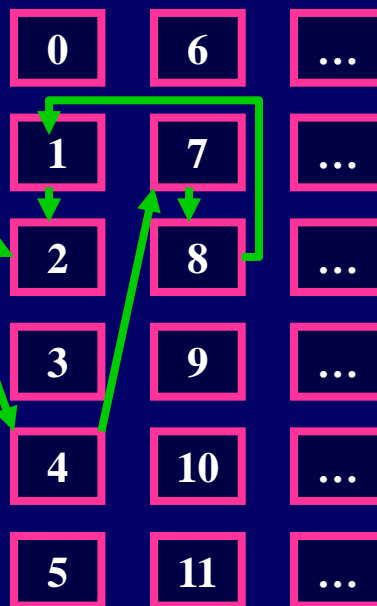
## ●1、隐式链接

- 在文件的每个目录项中，都含有指向链接文件第一盘块和最后一个盘块的指针。
- 每个盘块中都有指向下一个盘块的指针。
- 特点：只适合于顺序访问，随机访问效率极低。



### 目录

文件名:jeep  
起始块号:4  
结束块号:2  
.....



### 隐式链接





## ● 2、显式链接

- 把用于链接文件各物理块的指针，显式地存放在内存的一张“**链接表**”中。该表在整个磁盘只设置一张。即**文件分配表**（FAT）。序号为盘块号0..n-1
- **FCB**（文件控制块）：每个文件的首盘块号作为文件地址记录在FCB中。



FCB

物理地址:4

.....

物理块号	FAT
0	
1	2
2	EOF
3	
4	7
5	
6	
7	8
8	1
.....	.....

显式链接



## 8.1.3、FAT和NTFS技术

### ● 微软的文件系统：

- **FAT12**——早期MS-DOS系统
- **FAT16**——MS-DOS
- **FAT32**——Win95、Win98
- **NTFS**——Win NT、Win2000、Win xp

上述文件分配方式基本类似于**显式链接**



- “卷”概念的引入：
- 源于早期的MS-DOS的FAT文件系统。
- 支持将一个物理磁盘分成四个逻辑磁盘，每个逻辑盘就是一个卷（分区）。C、D、E、F四个卷。
- 每个卷都是一个能被单独格式化的使用的逻辑单元。都划出单独的区域存放自己的目录和FAT表，以及自己的逻辑驱动器字母。

## ● 1) 以盘块为基本分配单位

在早期的MS-DOS系统中。采用显式链接的方式。每个磁盘分区都有FAT表。文件的首盘块号放在自己的FCB中，FAT的每个表项中存放下一个盘块号。

例如：

下面计算该文件系统中，磁盘的最大容量。  
FAT表项为12位。FAT表中最多有 4096 个表项。  
盘块大小512B，每个磁盘分区的容量为2MB(4096\*512B)  
一个物理磁盘共四个分区，总容量为8MB

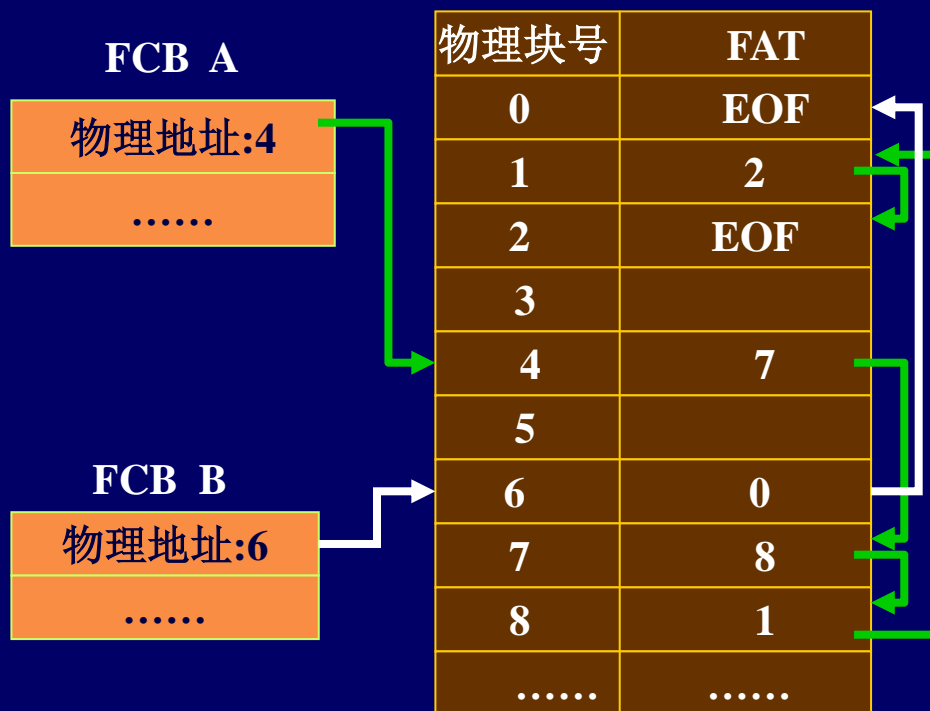




## 例1: MS-DOS的文件物理结构

文件A占用盘块号: 4、7、8、1、2

文件B占用盘块号: 6、0



MS-DOS的文件物理结构



## 2) 簇的基本概念

- 为了适应不断增大的磁盘容量的需要，以**簇**为基本单位进行分配。
- 簇是一组连续的扇区。簇的大小一般为 **$2n$** 个盘块。
- **MS-DOS**中，簇的容量：**一个、2个、4个、8个**扇区
- 簇为一个扇区时，磁盘容量：**8MB**
- 簇为两个扇区时，磁盘容量：**16MB**
- 簇为四个扇区时，磁盘容量：**32MB**
- 簇为八个扇区时，磁盘容量：**64MB**

**簇的优点：适应不断增大的磁盘容量。缺点：簇内碎片**



## 2、FAT16

- FAT12表最多只允许4096个表项，即最多分为4096个簇。解决办法：增加FAT表的宽度，以增加FAT表的表项数目。
- FAT16——FAT表宽度16位。最多允许65536个表项。  
( $2^{16}=65536$ ) 即最多可有65536个簇。
- 把具有16位表宽的FAT表称为FAT16。
- 簇中的盘块数可为4、8、16、32、64，因此，最大分区空间： $2^{16} * 64 * 512 = 2048\text{MB}$
- 缺点：簇内碎片太大。
- FAT12和FAT16不支持长文件名，受8+3格式限制。





### 3、FAT32

- 用**32位**表示**FAT表**的表项宽度。最多允许 **$2^{32}$** 个表项
- **FAT32**的每个簇都固定为**4KB**。
- **FAT32**分区的最大空间 **$4KB * 2^{32} = 2TB$**
- **FAT32**主要应用于**Win98**及后续**Windows**系统。
- 优点：簇较小，簇内碎片减少。支持的磁盘容量大
- 缺点：
  - 1、文件分配表太大，运行速度慢。
  - 2、有最小管理空间限制。卷至少有**65537**个簇，所以**不支持**容量小于**512MB**的分区。
  - 3、单个文件长度不能大于**4GB**
  - 4、不能保持向下兼容。



## 4、NTFS

### ● 1) NTFS新特性

- 适用于Win2000、xp、2003等。
- 使用了64位磁盘地址。支持 $2^{64}$ 字节磁盘分区。
- 支持长文件名，255个字符。
- 具有容错功能。
- 提供了数据的一致性。
- 提供了文件加密、文件压缩。



## ● 2) NTFS的磁盘组织

- 以簇作为空间分配和回收的基本单位。一个簇包含2n个盘块，大多数情况下，簇大小为4kB。
- 支持扇区大小不是512字节的非标准磁盘。

## ● 3) NTFS的文件组织

- 以卷为单位，将一个卷中的所有文件信息、目录信息、可用的未分配空间信息，以文件记录的方式记录在主控文件表中（MFT）。每个文件占一行，称为该文件的元数据，也称文件控制字。元数据大小1KB
- 文件较小时，文件的所有信息都存储在元数据中
- 文件较大时，文件的一部分属性存储在元数据中，其他属性，如文件内容放在其他簇中，并用指针连接。



## ● 4) NTFS的不足之处

- 只能被WinNT识别。NTFS可存取FAT系统文件，但是，反之不可以。即缺乏兼容性。



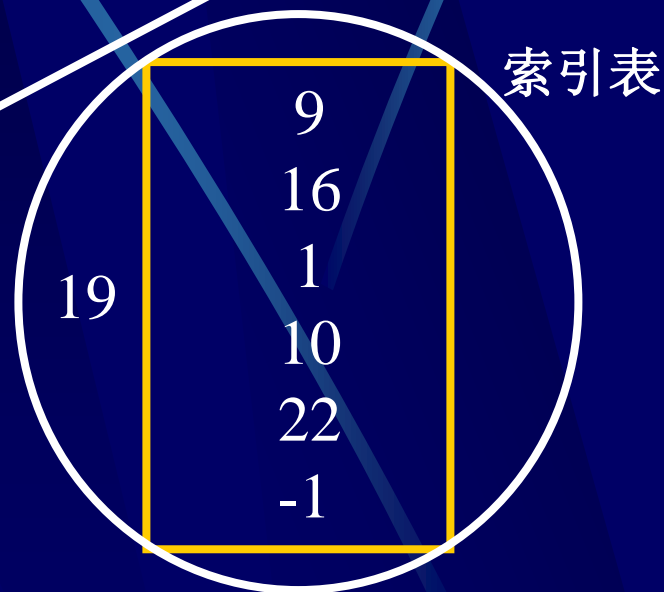
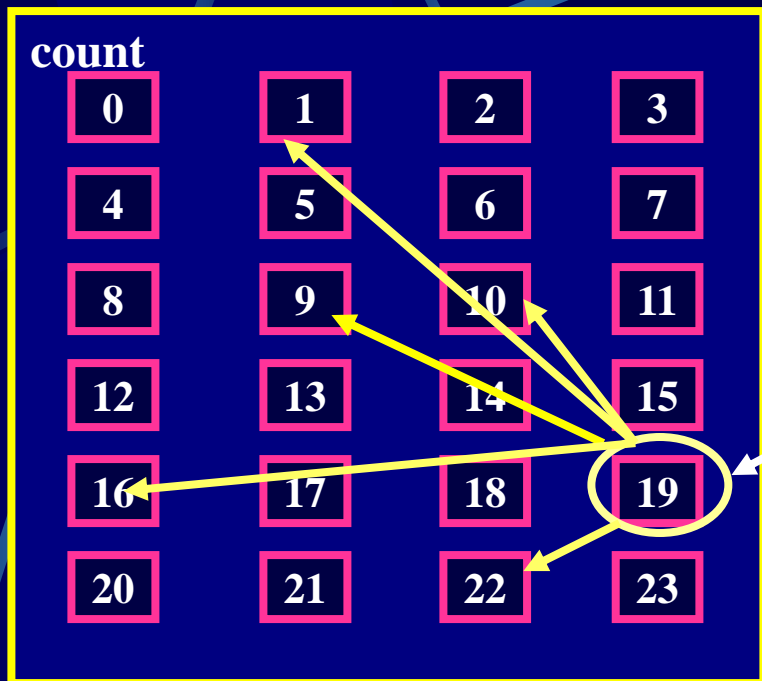
## 8.1.4、索引组织方式

- 1、链接分配方式的缺点：
  - 不能支持高效的直接存取
  - FAT需要占用较大的内存空间
- 2、单级索引分配
  - **基本思想**：将每个文件对应的盘块号集中存放
  - **方法**：为每个文件分配一个索引块（表），该文件的所有盘块号记录在内。



## 目录

File	块序号
Jeep	19



- 索引分配方式的优缺点：
  - 是一种离散分配方式，不会产生外部碎片
  - 支持直接访问
  - 缺点：花费较多的外存空间



### ● 3、多级索引分配

- 基本思想：

为大文件分配磁盘空间时，可形成多个索引块，需建立索引块的索引，放到一个索引块中，形成两级索引分配方式。

- 如果文件非常大，还可用三级、四级索引分配方式。



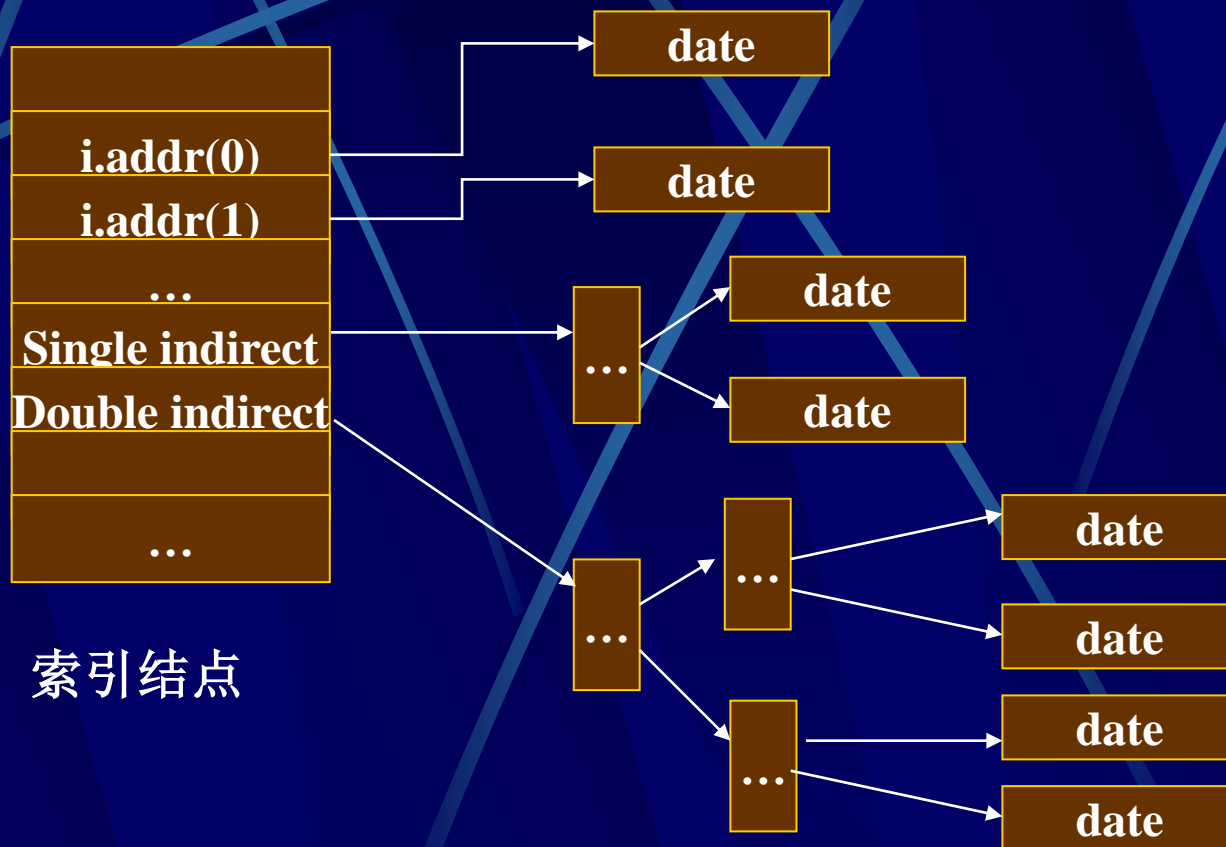
## 两级索引分配





## • 4、混合索引分配

- 多种索引分配方式相结合而形成的一种分配方式。既采用了直接地址，又采用了一级、两级或多级索引分配方式。
- 已在**UNIX**系统中采用。索引结点中共设**13**个地址项。其中*i.addr(0)~i.addr(9)*存放直接地址，其它地址项提供间接地址。



索引结点

混合索引方式



## 8.2 文件存储空间的管理

分配方式：连续分配、离散分配

存储空间的基本分配单位是以磁盘块（扇区）为单位，而非字节。

- 空闲表法和空闲链表法
- 位示图法
- 成组链接法



## 8.2.1、空闲表法和空闲链表法

### ● 1、空闲表法——连续分配方式

- 为外存上的所有空闲区建立一张空闲表

序号	第一空闲盘块号	空闲盘块数	盘块号
1	2	4	2, 3, 4, 5
2	9	3	9, 10, 11
3	15	5	15, 16, 17, 18, 19
4	—	—	

### ● 存储空间的分配与回收

- 分配：首次适应算法、循环首次适应算法
- 回收：考虑邻接的前后空闲区拼接合并



## ●2、空闲链表法——离散分配方式

### ●空闲盘块链

将磁盘上的所有空闲空间，以盘块为单位拉成一条链。

分配盘块时从链首开始；回收盘块时插入链尾

### ●空闲盘区链

将磁盘上所有空闲盘区（每个盘区可包含若干个盘块）拉成一条链。

每个盘区含有：指示下一个盘区的指针；  
本盘区大小的信息

分配盘区常采用首次适应算法；回收时要和相邻空闲盘区合并



## 8.2.2、位示图法

### ● 1、位示图

- 利用二进制的一位来表示磁盘中一个盘块的使用情况。由所有盘块所对应的位构成一个集合，称为位示图。用  $m \times n$  个位数构成位示图。

### ● 2、盘块的分配（分三步进行）

- 顺序扫描位示图，找到一个或一组值为“0”的位；
- 将找到的位转换成与之相对应的盘块号：

$$b = n(i-1) + j$$

- 修改位示图，令  $\text{map}[i,j]=1$



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	1	1	0	0	0	1	1	1	0	0	1	0	0	1	1	0
2	0	0	0	1	1	1	1	1	1	0	0	0	0	1	1	1
3	1	1	1	0	0	0	1	1	1	1	1	1	0	0	0	0
4																
⋮																
⋮																
16																

位示图



### • 3、盘块的回收（分两步进行）

- 将回收的盘块号转换成位示图中的行号和列号：

$$i = (b-1) \text{ DIV } n + 1 \quad j = (b-1) \text{ MOD } n + 1$$

- 修改位示图，令  $\text{map}[i,j]=0$

### ● 优点

- 从位示图中很容易找到一个或一组相邻接的空闲盘块；
- 位示图占用空间小，可常驻内存；
- 常用于微型机和小型机中，如CP/M





## 8.2.3、成组链接法

- 结合了空闲表法和空闲链表法而形成的一种空闲盘块管理方法，克服了表太长的缺点。（UNIX采用）
- 1、空闲盘块的组织
  - 空闲盘块号栈—超级块
  - 每100个空闲盘块作为一组
  - 将第一组的盘块总数和所有的盘块号，记入空闲盘块号栈中
  - 每组含有的盘块数和所有盘块记入前一组的末盘块
  - 最后一组只有99个盘块，以0作为结束标志



华中农业大学

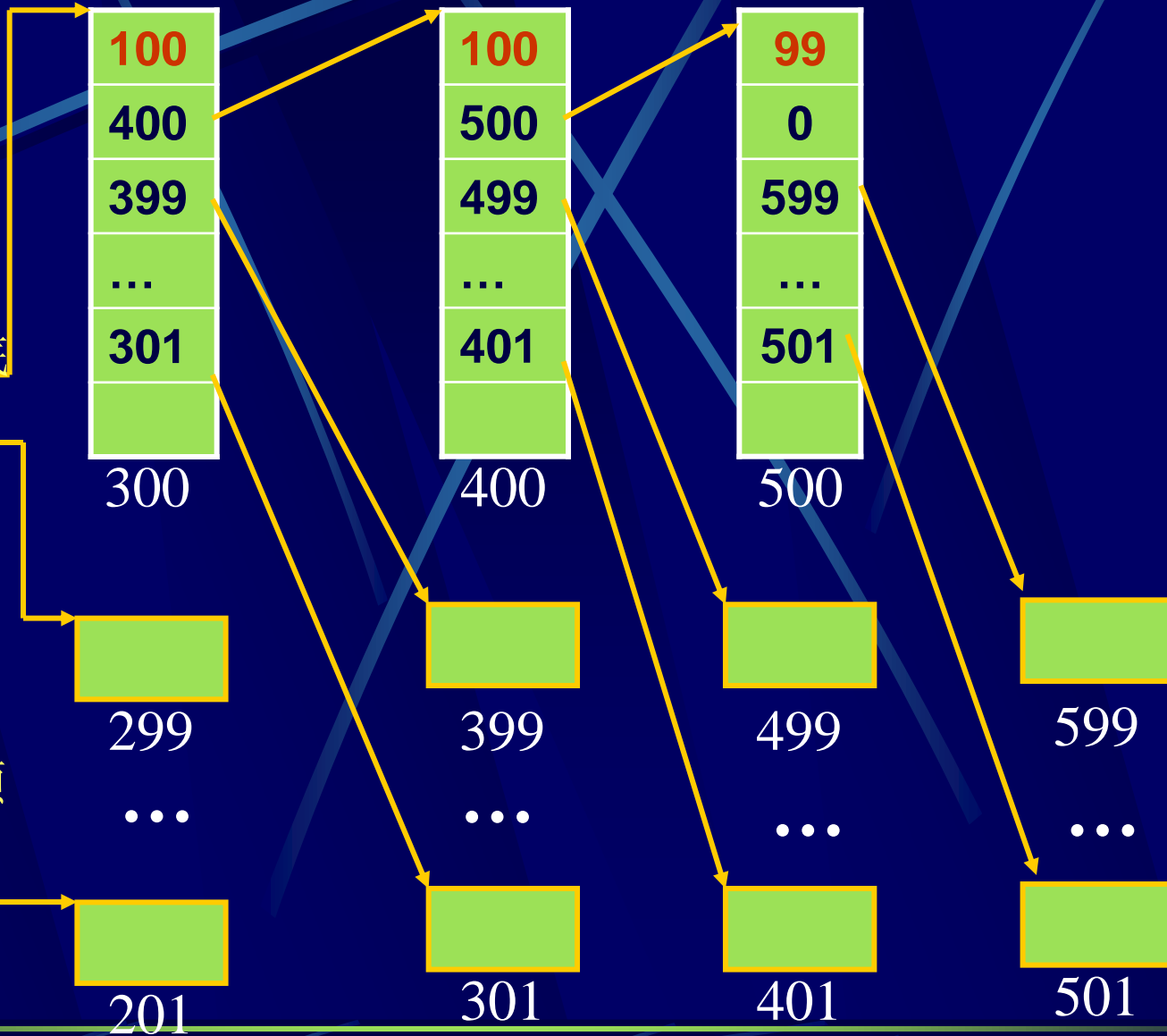
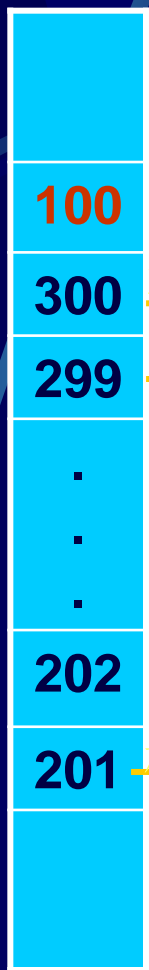
HUAZHONG AGRICULTURAL UNIVERSITY

空闲盘  
块号栈

总数  
s.free  
0  
1

栈底

栈顶



第一组

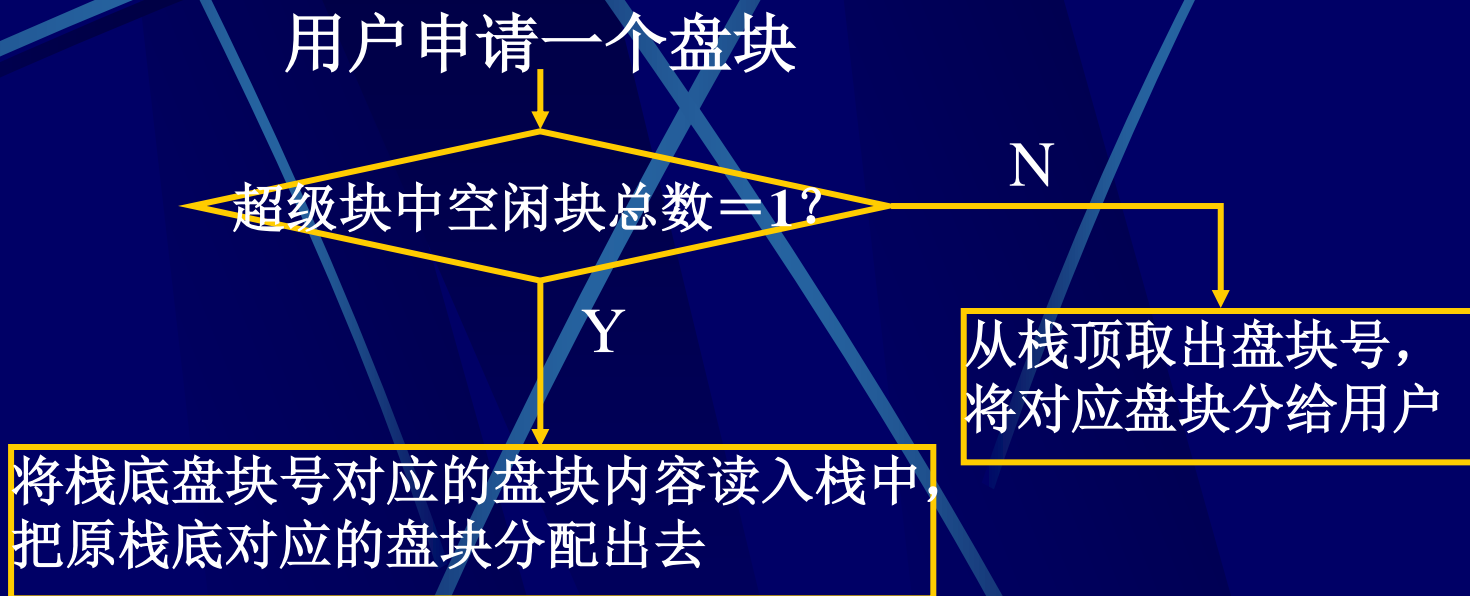
第二组

第三组

最后一组



## 2、空闲盘块的分配



## 3、空闲盘块的回收

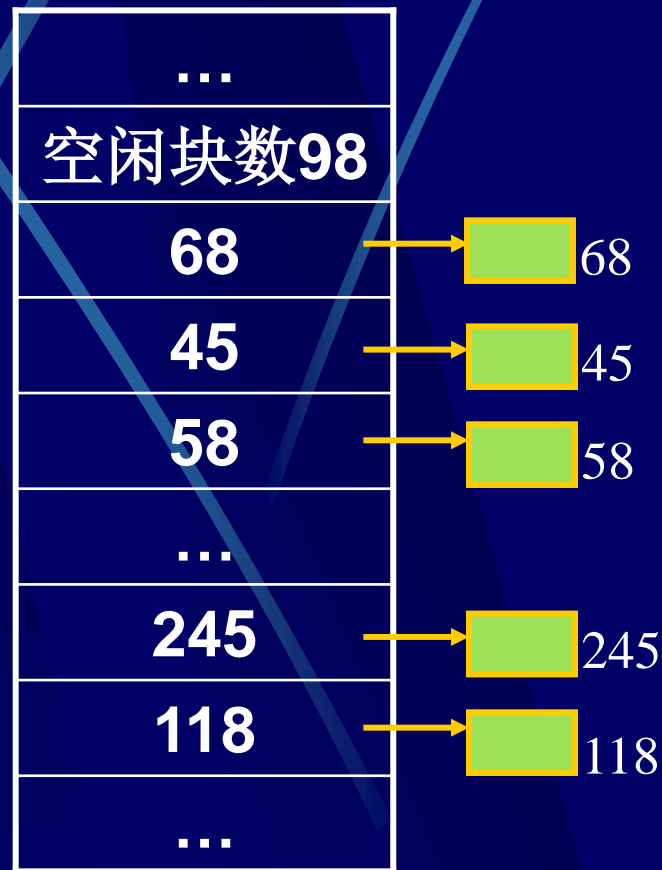
将回收盘块的盘块号记入空闲盘块号栈顶部，总数+1  
当栈中原空盘块总数=100时（栈满），将现有栈中的  
100个盘块号，记入新回收盘块中，再将其盘块号作为  
新栈底。



Unix系统中超级块如图示，  
问：

(1) 当用户释放了78#，89#，108#，204#物理块后，超级块中的变化情况如何？

(2) 当用户又申请了五个物理块，超级块中的变化情况如何？





## 6.6.3 磁盘容错技术

### ● 磁盘数据的不安全因素

- 人为因素：有意或无意的（错误）行为；
- 系统因素：系统的某些（个）部分出现异常；磁盘故障
- 自然因素：时间的推移，磁盘上的数据溢出或消失。

### ● 可以采取的措施

- 通过存取控制机制，防止人为因素；
- 通过磁盘容错技术，来防止磁盘故障；
- 通过“后备系统”，来防止自然因素。

### ● 容错技术（系统容错技术）：

- 通过增加冗余的磁盘驱动器、控制器等方法，来提高磁盘系统可靠性的一种技术。



## ● 容错技术分类——三类

低级磁盘容错技术SFT-I、中级磁盘容错技术SFT-II、  
基于集群技术的容错功能

### 1、低级磁盘容错技术SFT-I

- 防止因磁盘表面发生缺陷所引起的数据丢失。
- 1) 双份目录和双份文件分配表
  - 在不同磁盘或磁盘的不同区域中，分别建立主目录、主文件分配表；备份目录和文件分配表
  - 当主目录、主文件分配表损坏时，系统自动启用备份目录和文件分配表。





## 2) 热修复重定向和写后读校验

- 这是对少量坏的磁盘块（扇区）的补救措施
  - **热修复重定向**：磁盘容量的一部分（2%-3%），用于存放当发现磁盘有缺陷时的待写数据，并对写入的数据进行登记，以便访问
  - **写后读校验**：每次从内存缓冲区向磁盘中写入一个数据块后，又立即读出，并送至另一缓冲区中，进行数据比较。两者一致，继续写下一盘块，否则重写。重写后若仍不一致，则认为盘块有缺陷，数据写入**热修复重定向区**。

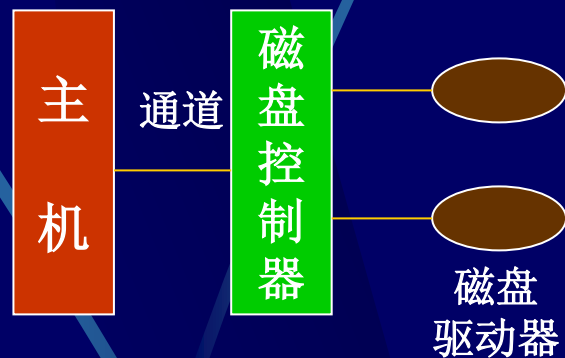


## 2、中级磁盘容错技术SFT-II

- 用于防止由磁盘驱动器和控制器的故障所导致的系统不能正常工作。

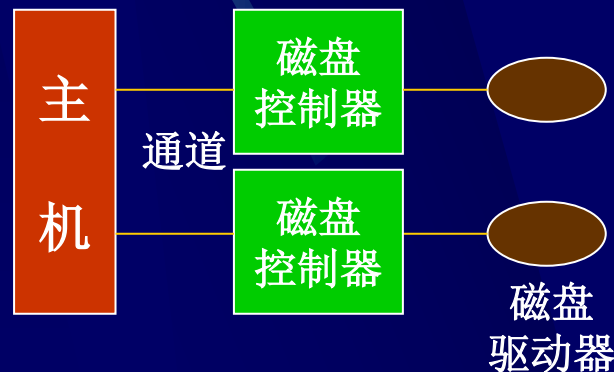
### • 1) 磁盘镜像

- 在同一磁盘控制器下，增设一个完全相同的磁盘驱动器。
- 每次向主磁盘写入数据后，都需要再写入到备份磁盘上。



### • 2) 磁盘双工

- 将两台磁盘驱动器分别接到两个磁盘控制器上，同时使这两台磁盘机镜像成对。
- 某个通道或控制器发生故障，另一通道上的磁盘仍正常工作







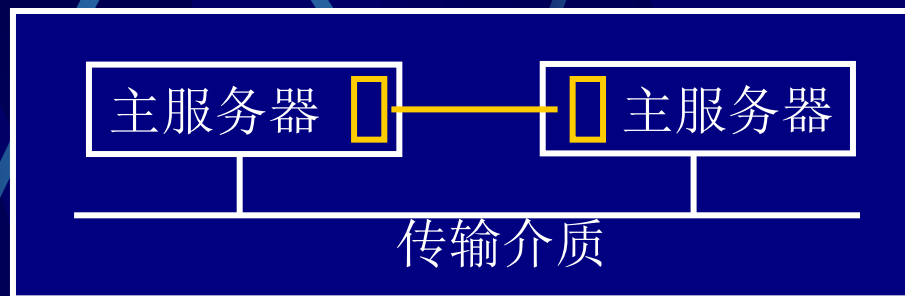
### ● 3、基于集群技术的容错功能

- **集群：** 由一组互连的自主计算机组成统一的计算机系统，给人们的感觉，它们是一台机器。
- **集群系统的主要工作模式：**

- 热备份模式
- 互为备份模式
- 公用磁盘模式

#### ● 1) 双机热备份模式

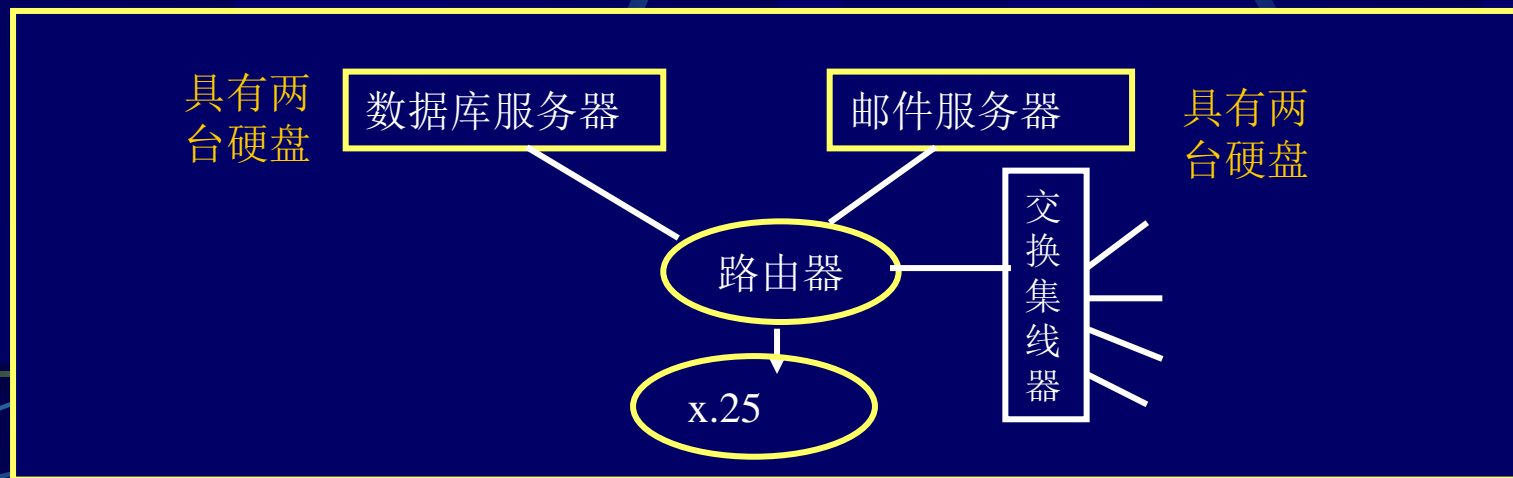
- 主服务器、备份服务器
- 平时，主服务器运行，备份服务器监视运行
- 主服务器发生故障，备份服务器接替主服务器，主服务器修复后，角色再次转换。





## ● 2) 双机互为备份模式

- 两台服务器均为在校服务器，各自完成自己的任务。
- 每台服务器内都配置两台硬盘，一个装载系统程序，一个接收另一台服务器发来的备份数据。
- 某台发生故障，由另一台负责其服务；修好后，再把服务功能返回。





### ● 3) 公用磁盘模式

- 将多台计算机连接到一台公用的磁盘系统中。
- 公用磁盘被划分为若干卷，每台计算机使用一个卷。
- 某台计算机发生故障，系统将选择另一台替代计算机，它对故障机的卷拥有所有权，并承担故障机的任务。



## 6.7 数据一致性控制

- **数据一致性**：不同文件中所存储的同一个数据相一致。
  - 保证数据一致性的软件
  - 相应的支持硬件
    - 稳定存储器（磁盘双工方式来实现）
- **6.7.1 事务**
  - 1、事务的定义：访问和修改各种数据项的一个程序单位
  - 事务也是一系列相关读和写操作。



- **托付操作**：对分布在不同位置的同一数据进行读和写操作全部完成时，以托付操作来终止事务
- **夭折操作**：只要有一个读、写操作失败，便执行夭折操作。并将事务内已被修改的数据项新值恢复成旧值。
- **事务操作具有“原子性”。**
- **2、事务记录**
  - **1、事务记录的数据结构：**  
事务名、数据项名、旧值、新值
  - 事务记录也称运行记录。存放在稳定存储器中
  - 事务记录表中的每一记录，描述了事务运行中的重要操作：**修改操作、开始事务、托付事务、夭折事务等。**



### ● 3、恢复算法

- 利用事务记录表，系统能处理任何故障而不致使故障造成稳定存储器中信息的丢失。 **恢复算法：**
  - (1) **Undo<Ti>** 把事务Ti修改过的数据恢复为旧值
  - (2) **Redo<Ti>** 把事务Ti修改过的数据设置为新值
- 如果系统发生故障，应对以前发生的事务进行清理。
  - 一类：各类操作已完成的事务：在事务记录表中，既包含<Ti开始>记录，又包含<Ti托付>记录。利用**Redo<Ti>** 过程，把所有已被修改的数据设置为新值。
  - 另一类：在Log表中，只有<Ti开始>，无<Ti托付>利用**Undo<Ti>** 过程，把已修改的数据恢复为旧值





## 6.7.2 检查点

- **检查点的作用：**使对事务记录表中的事务记录的清理工作经常化。即每隔一定时间，**做下述工作：**
  - 将驻留在内存中的当前事务记录表中的所有记录输出到稳定存储器中
  - 将驻留在内存中的所有已修改数据输出到稳定存储器中
  - 将事务记录表中的<检查点>记录输出到稳定存储器中
  - 每当出现一个<检查点>记录时，系统便执行恢复操作，利用**redo**和**undo**过程实现恢复功能。



## 6.7.3 并发控制

- 在多用户系统下，可能有多个用户在同时执行事务
- 事务的原子性，决定了各事务对数据项的修改是互斥的。即**顺序性的特征**。
- 保证事务处理的顺序性的同步机制——锁
  - 1、利用互斥锁实现顺序性
  - 2、利用互斥锁和共享锁实现顺序性





## ● 1、利用互斥锁实现顺序性

- **互斥锁**：仅允许一个事务对对象执行读或写操作
- 为每个共享对象设置一把互斥锁。当一事务Ti要去访问某对象时，应先获得该对象的互斥锁。然后才能对该对象执行读或写的操作。

## ● 2、利用互斥锁和共享锁实现顺序性

- **共享锁**：允许多个事务对相应对象执行读操作，不允许其中任何一个事务对对象执行写操作。
- 实现方式类似于“**读者——写者**”问题



## 6.7.4 重复数据的数据一致性问题

- 把关键文件和数据结构复制多份，存储在不同地方来保证数据的安全性。即重复数据的数据一致性。
  - 主文件、备份文件应相一致。
  - 不同处的空闲盘块表中数据应相一致



## 1、重复文件的一致性

### ● 以Unix为例。文件目录的不同

文件名	i结点
文件1	17
文件2	22
文件3	12
文件4	84

不允许有重复文件的目录

文件名	i结点		
文件1	17	19	40
文件2	22	72	91
文件3	12	30	29
文件4	84	15	66

允许有重复文件的目录

当一个文件被修改后，可查找文件目录，得到其他几个文件拷贝的索引结点号，在索引结点中找到各拷贝的位置进行修改。

或者，为新修改的文件建立几个拷贝，用新拷贝替换原来的文件拷贝。



## 2、盘块号一致性的检查

- 空闲盘块表记录空盘块的编号，FAT记录已分配盘块的使用情况。
- 保证盘块数据结构的一致性，用软件方法构成一个计数器表。

盘块号 计数器组	0	1	2	3	4	5	6	.....	n
空盘块号计数器	1	1	0	1	0	1	1		0
数据盘块号计数器	0	0	1	0	1	0	0		1

正常情况盘块号

每个盘块占一个表项，N为盘块总数。正常情况下，每列数据的值应为互补的。



非正常情况:

## 1、丢失了空闲盘块号

盘块号 计数器组	0	1	2	3	4	5	6	.....	n
空盘块号计数器	1	1	0	1	0	1	1		0
数据盘块号计数器	0	0	0	0	1	0	0		1

丢失了空闲盘块号

## 2、空闲盘块号重复出现

盘块号 计数器组	0	1	2	3	4	5	6	.....	n
空盘块号计数器	1	1	0	1	2	1	1		0
数据盘块号计数器	0	0	1	0	0	0	0		1

空闲盘块号重复出现



### 3、数据盘块号重复出现

盘块号 计数器组	0	1	2	3	4	5	6	.....	n
空盘块号计数器	1	1	0	1	0	0	1		0
数据盘块号计数器	0	0	1	0	1	2	0		1

数据盘块号重复出现



### 3、链接数一致性检查

- 对于Unix中的共享文件，索引结点号会在文件目录中出现多次。如，5个用户进程共享该文件，则出现5次。
- 该共享文件的索引结点中有一个链接计数count
- 正常情况下，这两个数字应一致。
- 检测一致性的方法：计数器表
  - 每个文件作为一个表项，含有索引结点号的计数值
  - 检查时，从根目录开始，遇到该索引结点号，则在计数器表中相应表项上加1，直至检查完毕
  - 用索引结点号的计数值与count比较，一致，则正确