



海南大学  
HAINAN UNIVERSITY



# 《操作系统原理及安全》

## 第9章 磁盘存储器管理

教师：秦小立

学院：网络空间安全学院

邮箱：xlqin@hainanu.edu.cn

办公地点：学院309



# 课程知识导图

OS

第1章 操作系统引论

第2章 进程的描述与控制

第3章 处理机调度与死锁

第4章 进程同步

第5章 存储器管理

第6章 虚拟存储器

第7章 输入/输出系统

第8章 文件管理

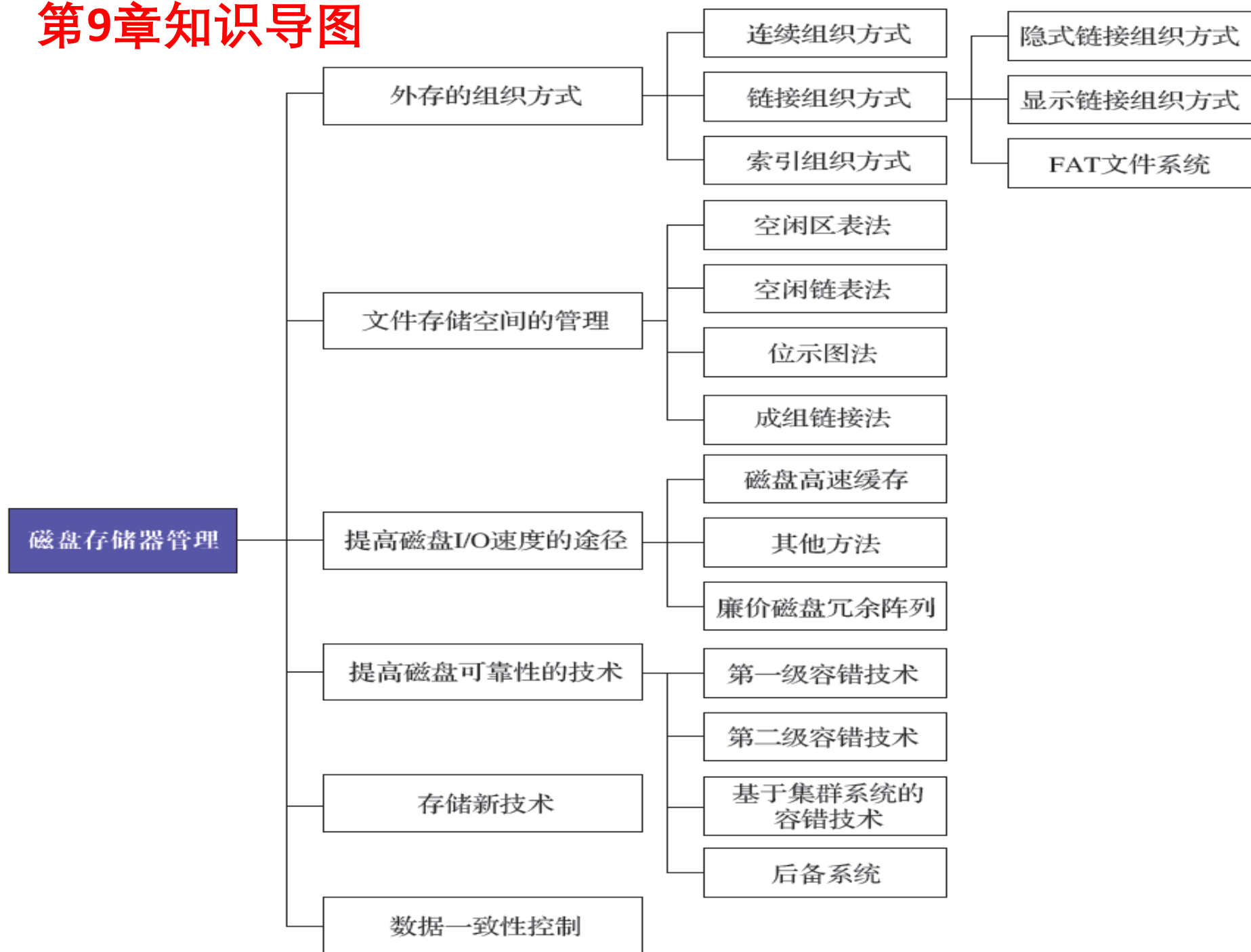
第9章 磁盘存储器管理

第10章 多处理机操作系统

第11章 虚拟化和云计算

第12章 保护和安全

# 第9章知识导图





## 第9章 磁盘存储器管理

### 9.1 外存的组织方式

### 9.2 文件存储空间的管理

### 9.3 提高磁盘I/O速度的途径

### 9.4 提高磁盘可靠性的技术

### 9.5 存储新技术

### 9.6 数据一致性控制

## 9.1 外存的组织方式

文件在逻辑上可以看作是连续的；但物理存储上却有不同  
的存储方式；

文件的物理结构直接与外存的组织方式有关。

不同的外存组织方式，将形成不同的文件物理结构。

- 连续组织方式：形成的文件物理结构是顺序式文件结构。
- 链接组织方式：形成的文件物理结构是链接式文件结构。
- 索引组织方式：形成的文件物理结构是索引式文件结构。

现代OS，可能会采用多种类型的外存组织方式。

## 9.1.1连续组织方式

**定义：** 为每个文件分配一组相邻接（连续）的磁盘块

### 优点：

- 顺序访问容易 – 仅需要起始块号和长度
- 顺序访问速度快：是几种外存组织方式中最快的一种
- 支持随机访问

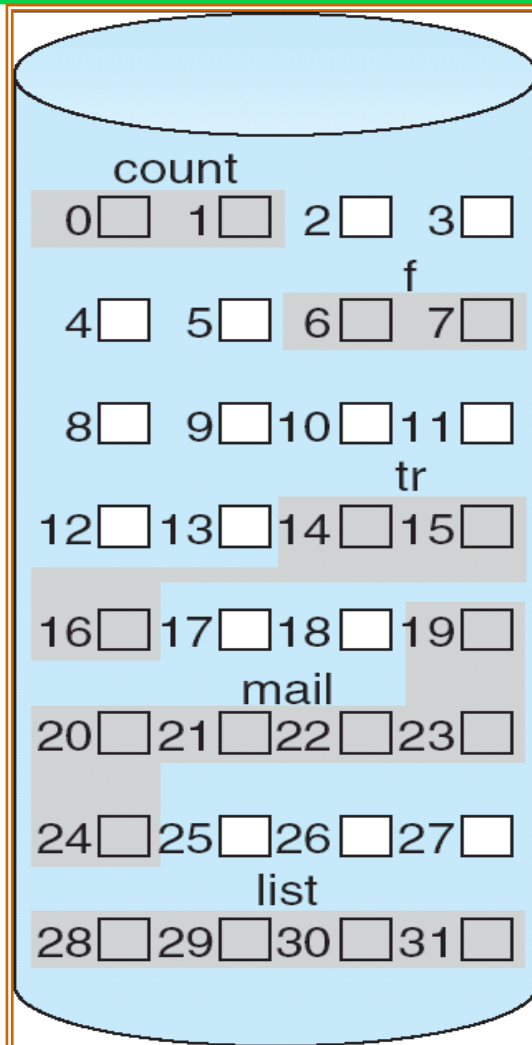
### 缺点：

- 要求分配连续的存储空间，会产生许多外碎片，因此外存空间利用率低
- 必须事先知道文件的长度
- 不能灵活地删除和插入记录
- 文件不能动态增长



## 9.1.1连续组织方式

### 磁盘空间的连续组织方式



directory

file	start	length
count	0	2
tr	14	3
mail	19	6
list	28	4
f	6	2

## 9.1.2 链接组织方式

**定义：** 为文件分配不连续的盘块，通过链接指针将一个文件的所有盘块链接在一起，所形成的物理文件称为[链接文件](#)

### 优点：

- 消除了磁盘的外碎片，提高了外存的利用率
- 对插入、删除和修改记录都非常容易
- 文件可动态增长

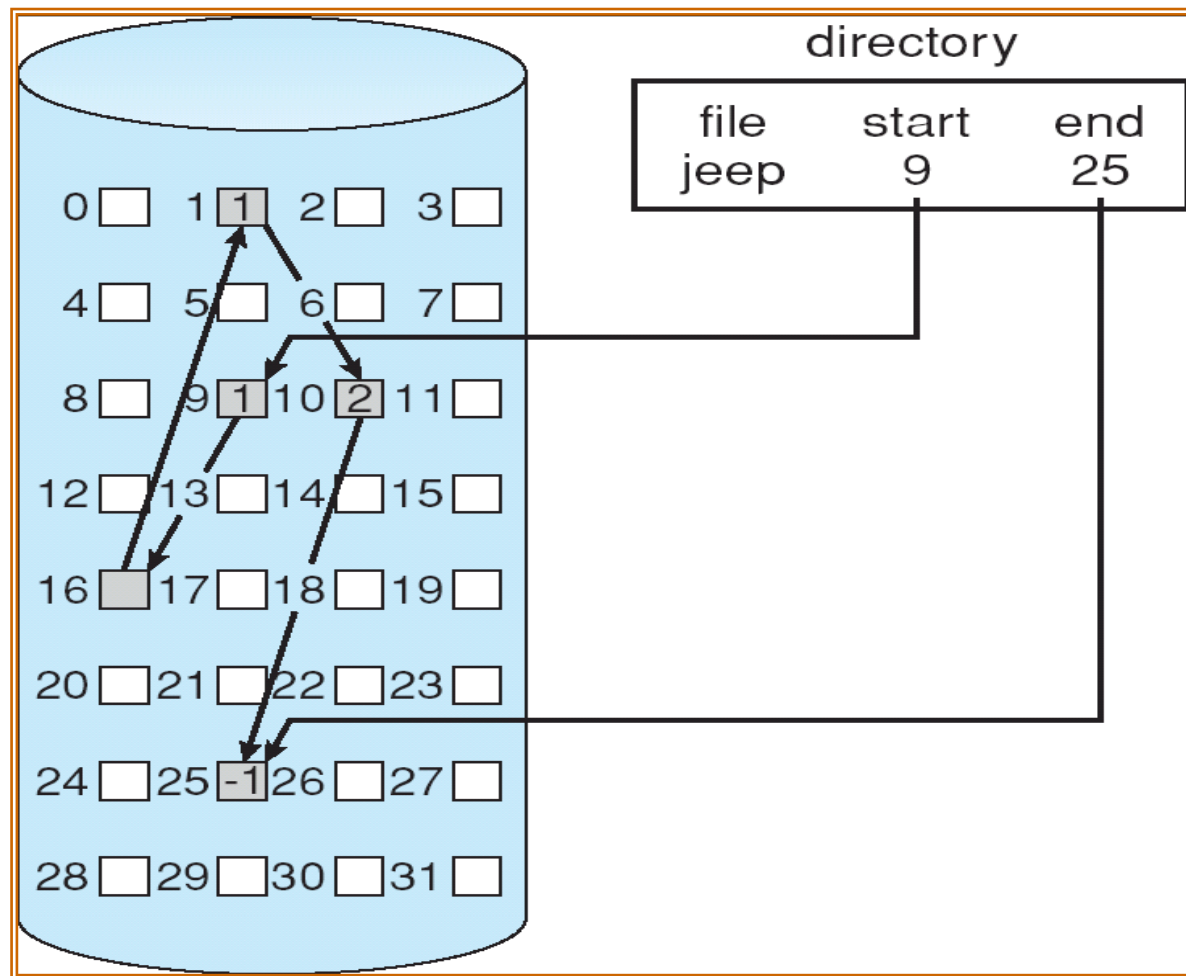
### 链接方式

- 隐式链接：指针存放在每个盘块中，只适合于顺序访问
- 显式链接：指针显式地存放在内存的文件分配表FAT中



## 9.1.2 链接组织方式

### 隐式链接示例





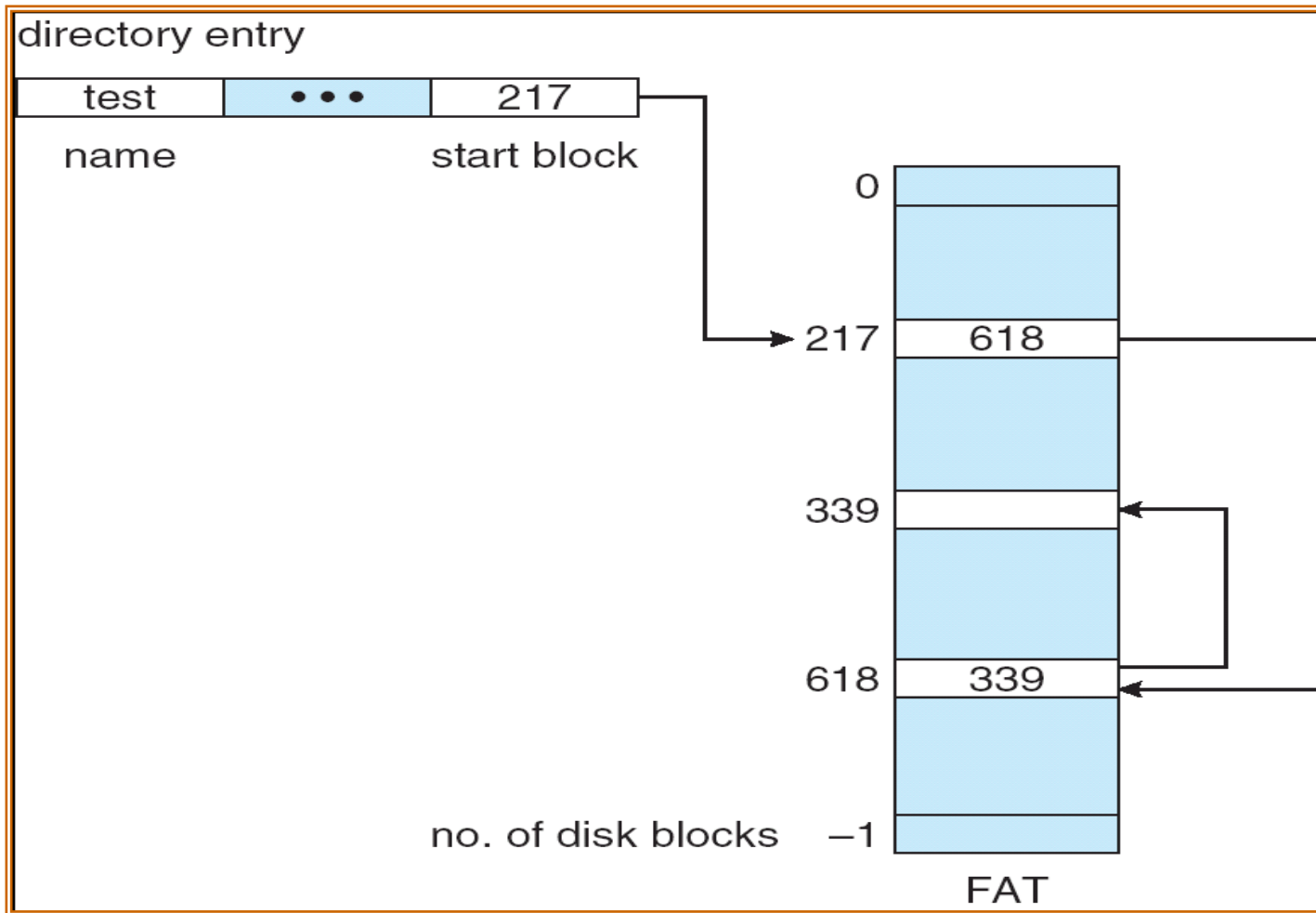
## 9.1.2 链接组织方式

### 显式链接：FAT

- 特点:**链接指针统一存储在**一张文件分配表FAT**中，并将文件的首个物理块地址登记在它的目录项（也称文件控制块FCB）中。
- FAT:**在一个文件卷(逻辑盘)中仅设置一张FAT(**实际上配有两个文件分配表:FAT1和FAT2。FAT1用于日常工作，FAT2备用，因此，在FAT1损坏时，可用FAT2表修补**)，其每个表项的序号为对应的物理块号，而表项中的内容则是分配给文件的下一个物理块的指针，即下一个盘块号。
- 优点:**FAT常驻内存，检索速度快，可解决隐式链接查找时需多次访问磁盘的问题。

## 9.1.2 链接组织方式

### 显式链接示例：FAT







## 9.1.2 链接组织方式

### FAT文件系统

**FAT文件系统**是微软最早在MS-DOS开始使用的文件系统

FAT (File Allocation Table)

■ FAT12      ■ FAT16      ■ FAT32      ■ FAT64 (exFAT)

卷	布局	类型	文件系统	状态	容量
	简单	基本		状态良好 (EFI 系统分区)	100 MB
 (C:)	简单	基本	NTFS	状态良好 (启动, 故障转储, 主分区)	111.57 GB
 (G:)	简单	基本	exFAT	状态良好 (主分区)	119.04 GB
 (I:)	简单	基本	FAT32	状态良好 (主分区)	14.31 GB

## 9.1.2 链接组织方式

### FAT文件系统 (2)

#### 1、以盘块为单位的FAT文件系统

➤ **FAT12**以盘块为基本分配单位， $2^{12}=4096$ 个表项，每个分区有两张相同的FAT表，FAT1和FAT2

➤ 例如，对1.2MB的软盘，每个盘块大小为512B

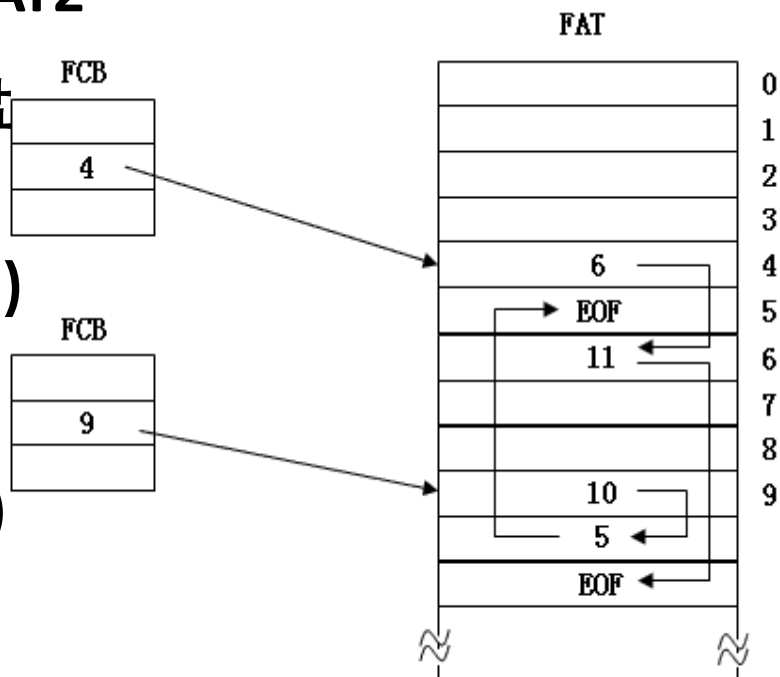
■ FAT共有2.4K个表 (1.2MB/512B)

■ 每个FAT表项占12b (1.5B)

■ 因此FAT表占用3.6KB(2.4K\*1.5B)

➤ 最大管理单个磁盘容量8M

➤ 只支持8+3格式文件名;



## 9.1.2 链接组织方式

### FAT文件系统 (3)

#### 2、以簇为单位的FAT文件系统

- 簇是一组相邻的扇区，在FAT中被视作一个虚拟扇区
- 簇的大小一般是 $2^n$ 个盘块( $n=1,2,4,8,\dots$ )
- 一个簇应包含扇区的数量与磁盘容量大小直接相关

在MS-DOS中，簇的容量可以有2个(1KB)、4个(2KB)或8个扇区(4KB)等;当簇包含8个盘块，磁盘分区最大容量为16MB。

- 簇的优点:可减少FAT表中的项数（在相同磁盘容量下，FAT表中的项数与簇大小成反比），FAT表占更少存储空间，提高文件系统效率。



## 9.1.2 链接组织方式

### FAT文件系统 (3)

以簇为单位的FAT文件系统

➤ FAT16支持一个簇有64个盘块，其FAT表的表项数最大为 $2^{16}$ ，则最大可以管理的分区空间为 $2048\text{MB}(512*64*2^{16})=2\text{G}$

➤ 优点：

□ 可以减少FAT中的项数，使FAT占用更少的存储空间，并且可以减少访问FAT的存取开销

➤ 缺点：簇的容量越大，簇内碎片越大

□ FAT16对FAT12改善有限，磁盘容量依旧受限严重，且只支持8+3格式文件名；Win95以后的系统对FAT16进行了扩展，文件名长度可达255个字符。





## 9.1.2 链接组织方式

### 4、FAT 32

- 两份FAT表；
- 每个簇(物理块)固定为4KB~32KB；
- FAT表的表项占据32位；FAT表最大表项数 $2^{32}$ 项；
- 单个文件不能大于4G；
- FAT32管理的单个最大磁盘空间： $512B * 2^{32} = 2TB$ 。





## 9.1.2 链接组织方式

### 4、FAT 32

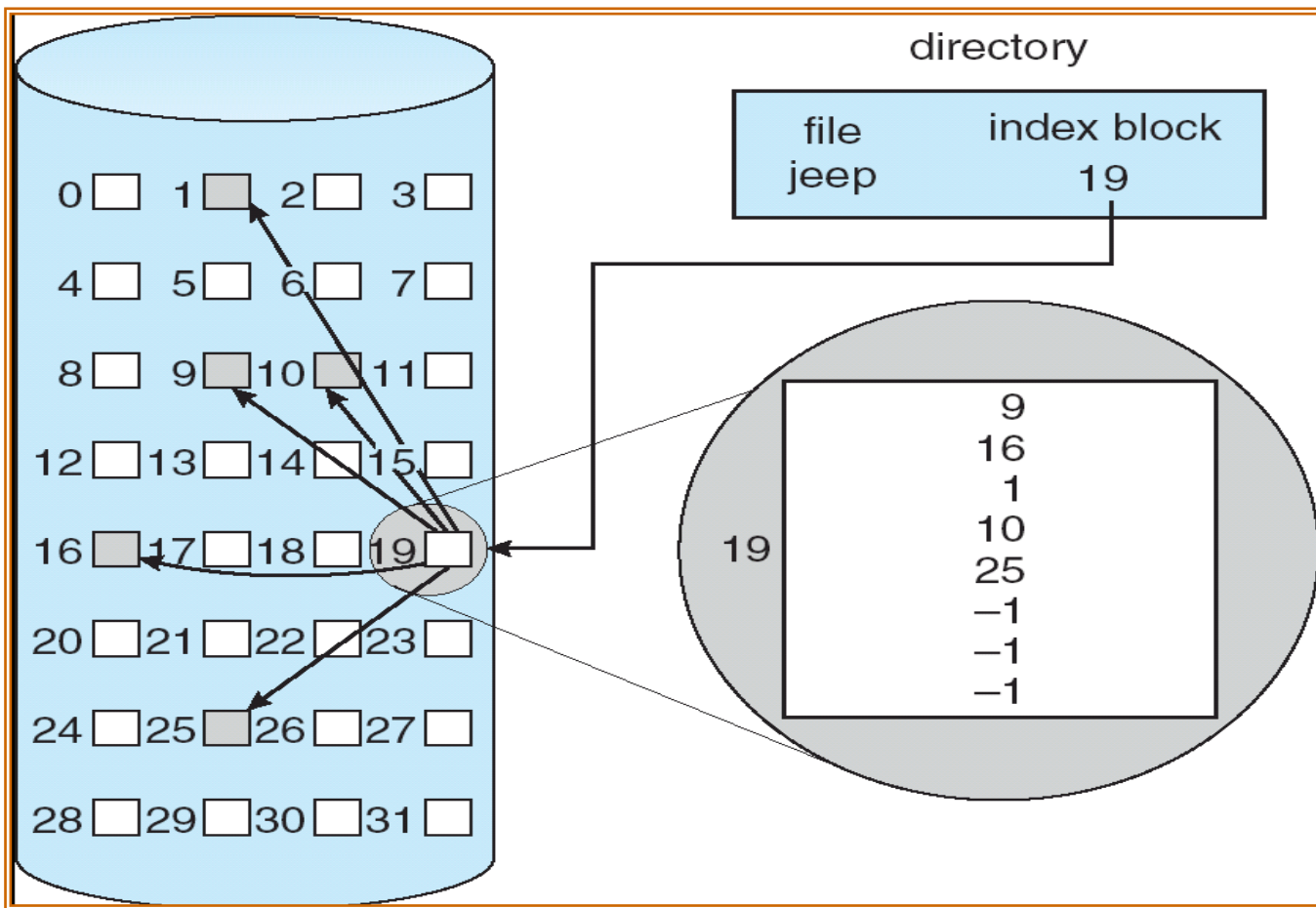
- 两份FAT表；
- 每个簇(物理块)固定为4KB~32KB；
- FAT表的表项占据32位；FAT表最大表项数 $2^{32}$ 项；
- 单个文件不能大于4G；
- FAT32管理的单个最大磁盘空间： $512B * 2^{32} = 2TB$ 。

## 9.1.3索引组织方式

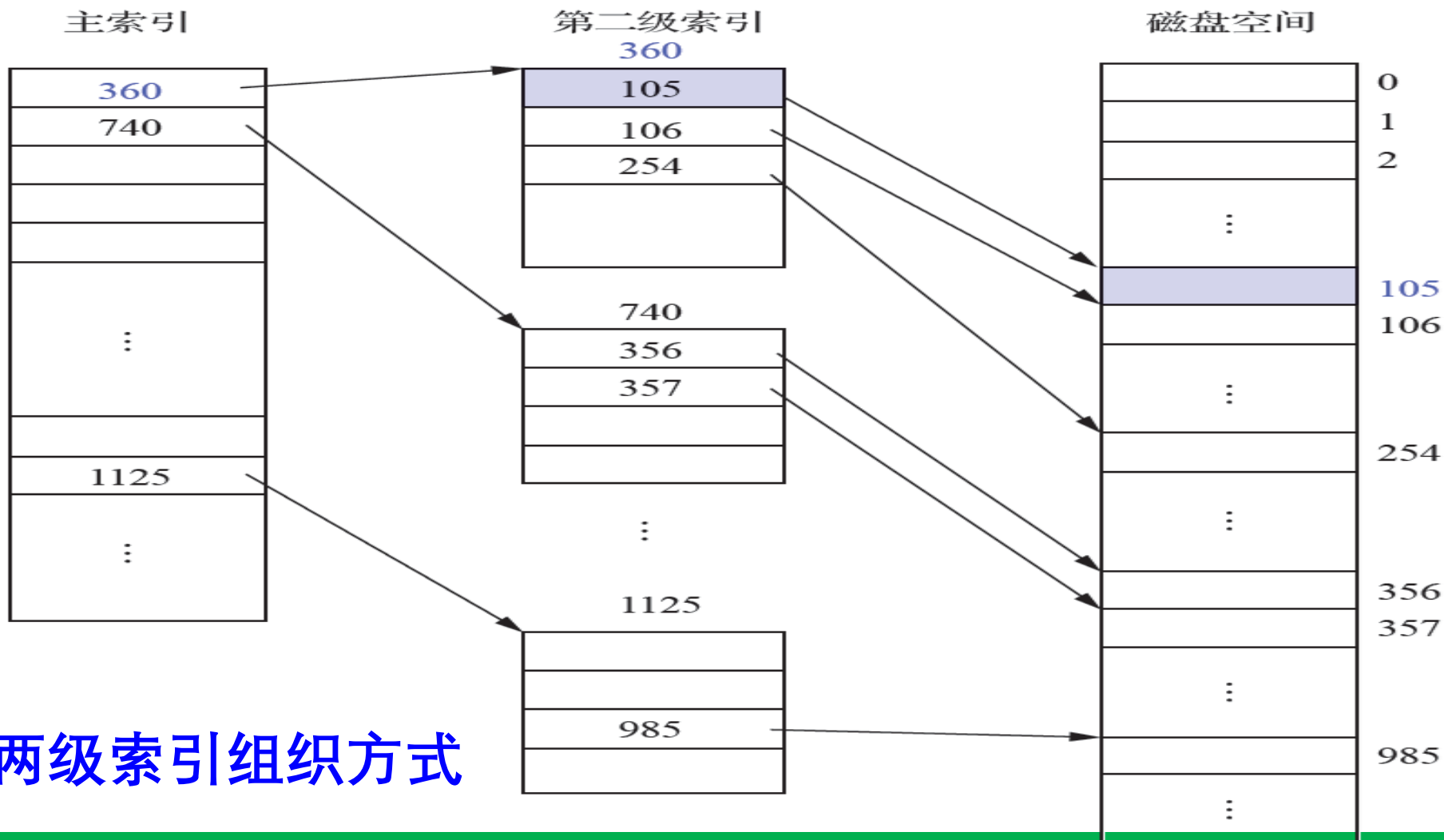
- 链接方式的缺点
  - 不能支持高效的直接存取 ➤ FAT表需占用较大的内存空间
- 索引方式：把所有的磁盘块号放在一个索引块(表)中
- 优点：
  - 支持随机访问 ➤ 动态存取没有外碎片，但索引块的负担较重
- 缺点：
  - 需要索引表
  - 对小文件，其索引块的利用率低

## 9.1.3 索引组织方式

### 索引组织方式示例

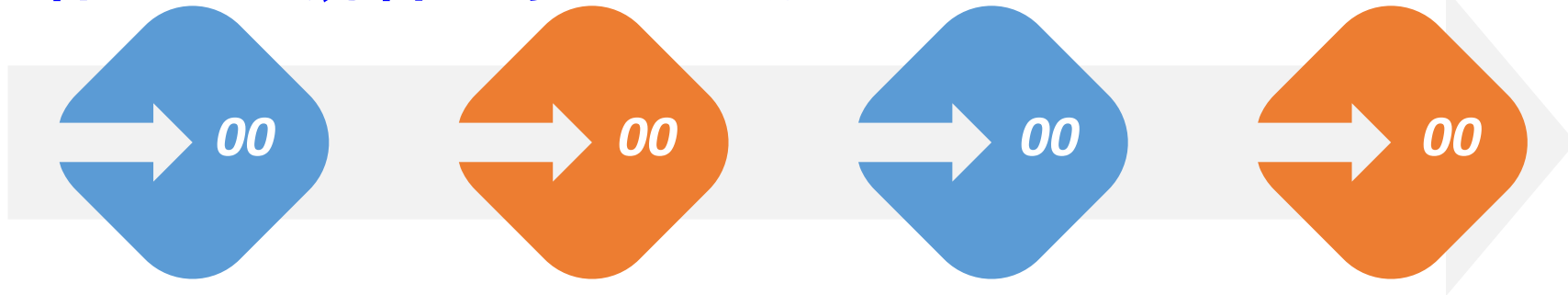


## 9.1.3 索引组织方式



## 9.1.3 索引组织方式

### 增量式（混合）索引组织方式



对于小文件（如大小为1KB-10KB或4KB-40KB）

- 可直接寻址（直接从FCB获得文件的盘块地址）

对于中文件（如大小为11KB-256KB或5KB-4MB）

- 可一次间址（从FCB中找到该文件的索引表）

对于大型和特大型文件

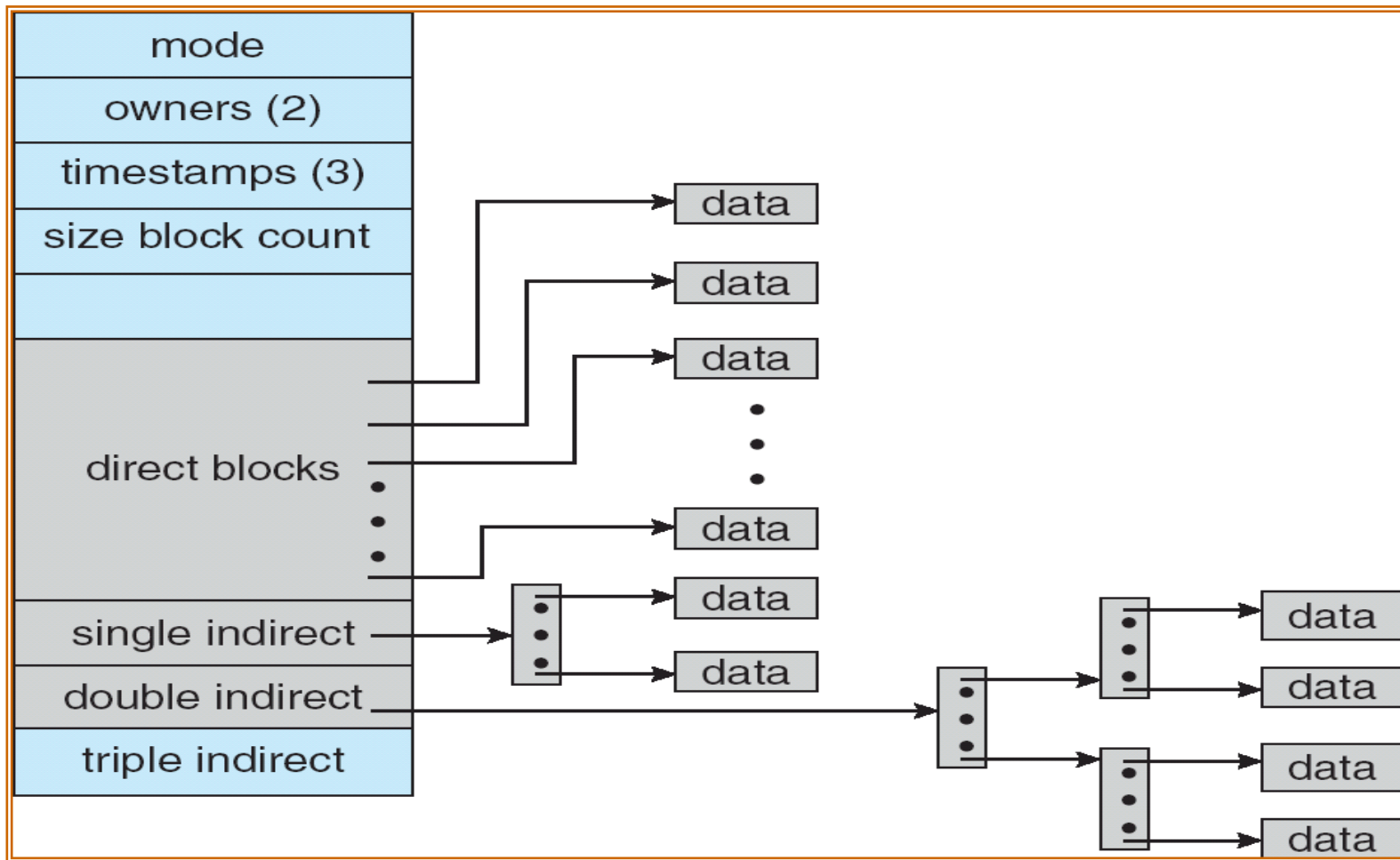
- 可二次或三次间址（两级或三级索引组织方式）

**增量式索引组织方式（混合索引组织方式）**

- 既采用直接寻址方式，又采用单级和多级索引组织方式（间接寻址方式）

## 9.1.3 索引组织方式

### 混合索引方式(UNIX iNode)







内容导航:

## 第9章 磁盘存储器管理

9.1 外存的组织方式

9.2 文件存储空间的管理

9.3 提高磁盘I/O速度的途径

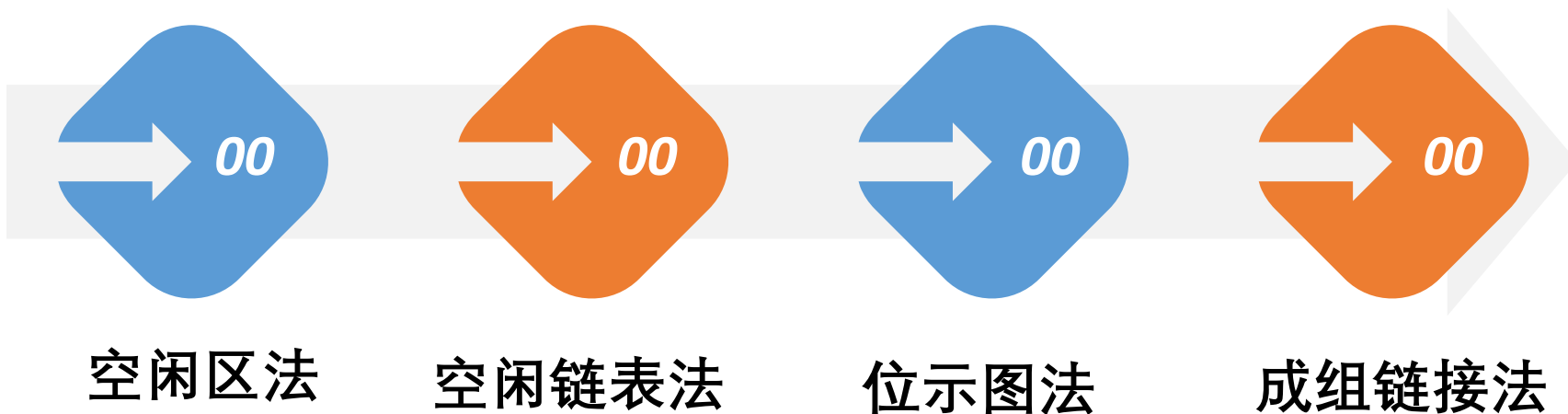
9.4 提高磁盘可靠性的技术

9.5 存储新技术

9.6 数据一致性控制

## 9.2 文件存储空间管理

- 文件存储空间的基本分配单位是盘块 (block)
- 常用的管理方法



## 9.2.1 空闲区表法

- 系统为外存上所有空闲区建立一张空闲表，属于连续分配方式
- 分配与回收
  - **分配**类似于内存的分区动态分配，如首次适应算法和最佳适应算法等
  - **回收**时，也类似于内存回收方法

序号	第一空闲盘块号	空闲盘块数
1	2	4
2	9	3
3	15	5
...	...	...

## 9.2.1 空闲区表法

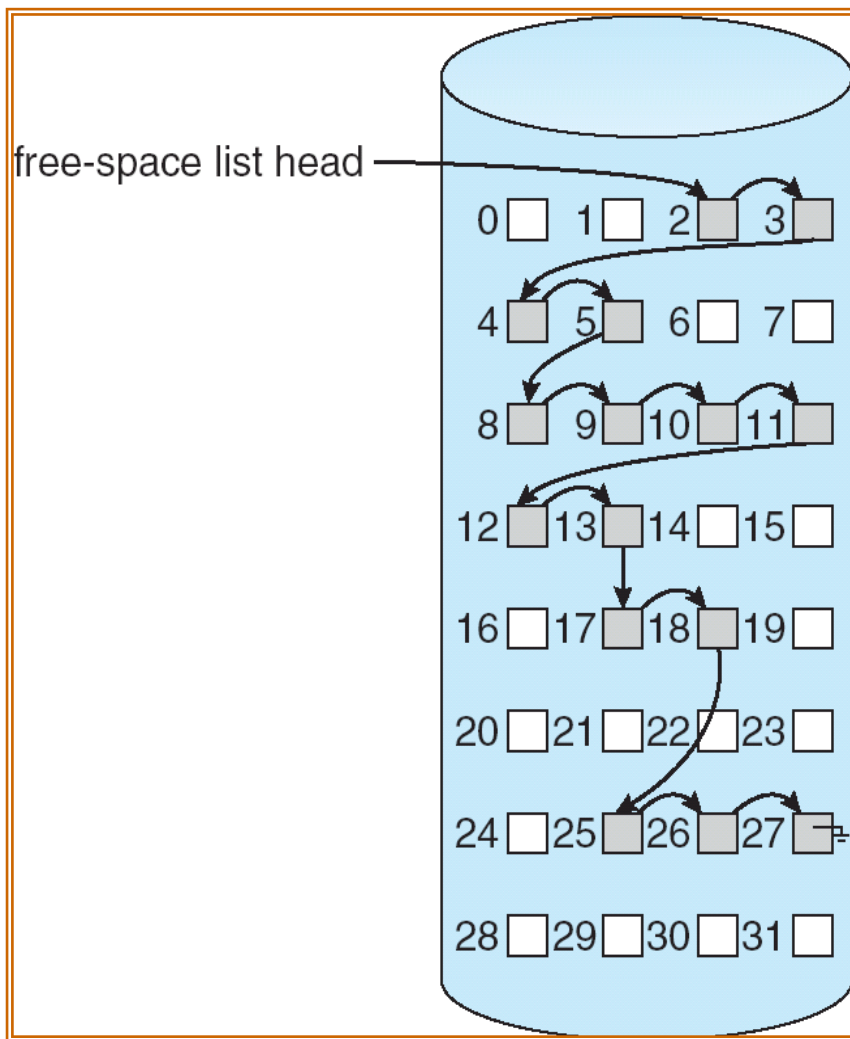
### 空闲链表法

#### 空闲盘块链

- 以盘块为单位链接起来
- 优点：分配和回收简单
- 缺点：效率较低

#### 空闲盘区链

- 将所有空闲盘区（每个盘区可包含若干个盘块）链接起来
- 分配盘区的方法与内存分区的动态分配方法类似
- 优点：效率较高
- 缺点：分配和回收比较复杂



## 9.2.2 位示图法

### 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
  - 当值为0时，表示对应的盘块空闲
  - 当值为1时，表示已分配
- 磁盘上的所有盘块都有一个二进制位与之对应，这样，由所有盘块所对应的位构成一个集合，称为**位示**

	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
...																
16																



## 9.2.2 位示图法

### 盘块的分配和回收

#### 盘块的分配

- 顺序扫描位示图，从中找出一个或一组其值为0的二进制位
- 将所找到的一个或一组二进制位，转换成与之相应的盘块号。  
假定找到的其值为0的二进制位，位于位示图的第*i*行、第*j*列，则其相应的盘块号为： $b=n(i-1)+j$
- 修改位示图，令 $\text{map}[i,j]=1$

#### 盘块的回收

- 将回收盘块的盘块号**b**转换成位示图中的行号和列号：
  - $i=(b-1)\text{div } n+1$     $j=(b-1)\text{mod } n+1$
- 修改位示图，令 $\text{map}[i,j]=0$

## 9.2.3 成组链接法



### 空闲盘块的**组织**

- 文件区中的所有空闲盘块，被分成若干个组
- 将每一组含有的盘块总数 $N$ 和该组所有的盘块号，记入其前一组的第一个盘块的 $S.free(0) \sim S.free(99)$ 中。
- 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，作为当前可供分配的空闲盘块号
- 最末一组只有99个盘块

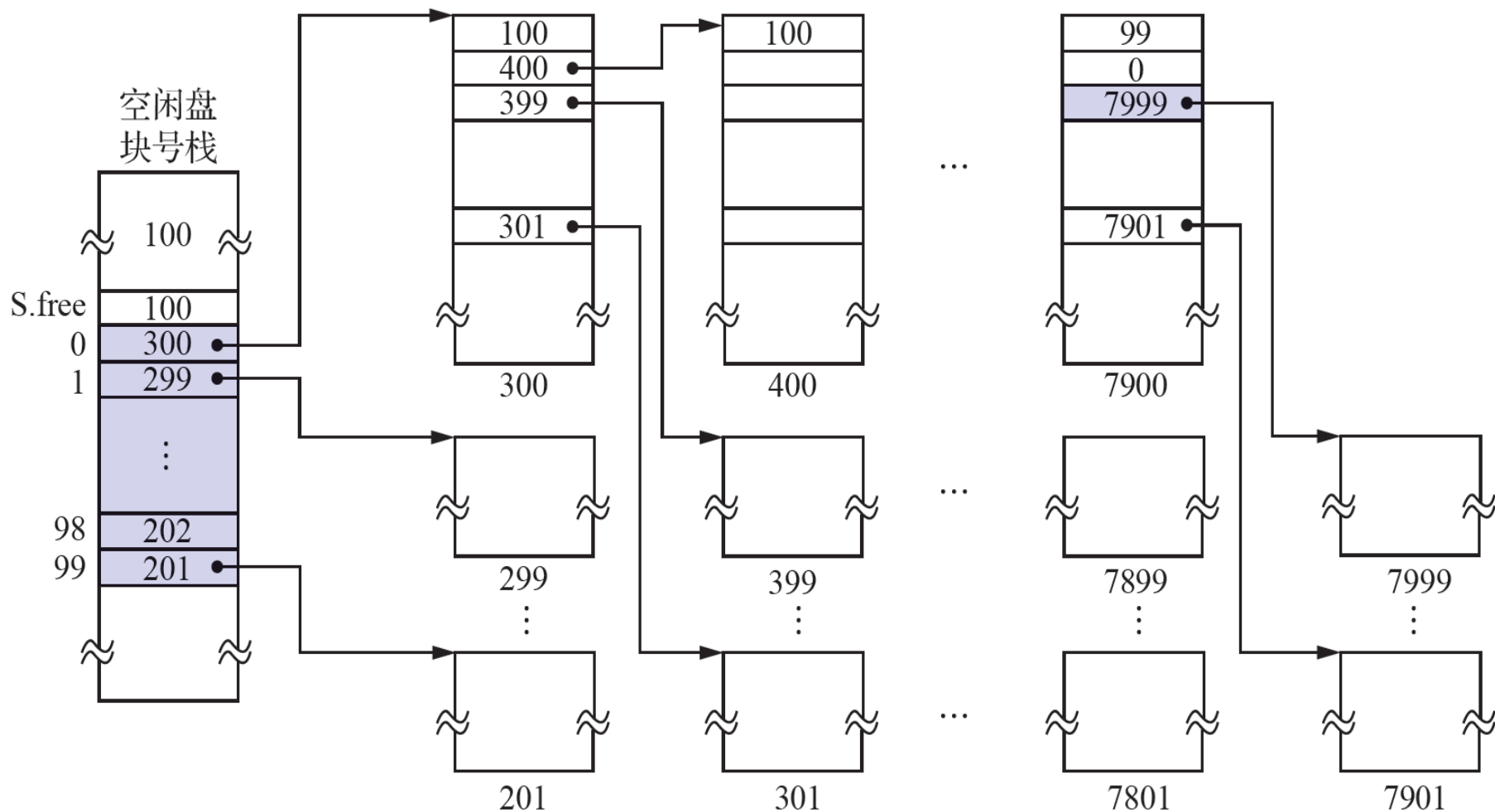


### 空闲盘块的分配与回收



## 9.2.3 成组链接法

### 成组链接法示例





内容导航:

## 第9章 磁盘存储器管理

9.1 外存的组织方式

9.2 文件存储空间的管理

**9.3 提高磁盘I/O速度的途径**

9.4 提高磁盘可靠性的技术

9.5 存储新技术

9.6 数据一致性控制

## 9.3 提高磁盘I/O速度的途径



### 提高文件的访问速度

- 改进文件的目录结构以及检索目录的方法
- 选取好的文件存储结构
- 提高磁盘I/O速度



### 磁盘I/O速度远低于内存的访问速度，**已成为瓶颈**

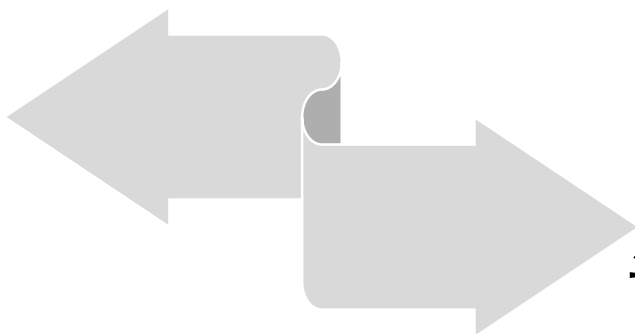
- 磁盘高速缓存
- 其他方法
  - 提前读、延迟写、优化物理块的分布、虚拟盘等

## 9.3.1 磁盘高速缓存

### 磁盘高速缓存



在内存中为磁盘盘块设置一个缓冲区，在该缓冲区中保存了某些盘块的副本



访问磁盘时，先查看内容是否已在磁盘高速缓存中，如果在，直接从缓存中获取；如果不在，需启动磁盘将需要的盘块读入内存，并送到缓存中

## 9.3.1 磁盘高速缓存

### 设计磁盘高速缓存需考虑的问题



#### 数据交付方式

- 数据交付：直接将磁盘高速缓存中的数据送到内存
- 指针交付：只将指向磁盘高速缓存中某区域的指针交付给请求者进程



#### 置换算法

- 较常用算法：**LRU**、Clock以及最少使用页面置换算法



#### 周期性地写回磁盘

- 根据LRU算法，经常被访问的盘块数据可能会一直保留在缓存中，长期不会被写回磁盘
- UNIX系统增设了一个修改程序在后台运行，周期性写回磁盘

## 9.3.2提高磁盘I/O速度的其他方法

01

### 提前读

- 在读当前块的同时要求将下一个盘块（提前读的盘块）中的数据也读入缓冲区

02

### 延迟写

- 缓冲区中的数据延迟写回磁盘，已被广泛采用

03

### 优化物理块的分布




- 为文件分配盘块时，优化盘块的分布，如将同属于一个文件的盘块安排在同一条磁道上或相邻的磁道上，减少磁头的移动距离

04

### 虚拟盘

- 利用内存空间仿真磁盘（虚拟盘，RAM盘）

## 9.3.3 廉价磁盘冗余阵列

-  由多个小磁盘所组成的一个容量很大的**廉价磁盘冗余阵列RAID**
-  RAID利用一台磁盘阵列控制器来统一管理和控制一组（几台到几十台）磁盘驱动器，进而组成一个大型磁盘系统
  - 大幅度增加了磁盘的容量
  - 提高了磁盘的I/O速度
  - 提高了磁盘系统的可靠性
-  **优点：**
  - 可靠性高
  - 磁盘I/O速度快
  - 性价比高

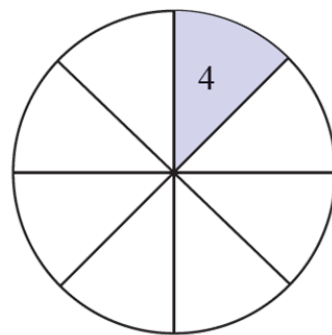
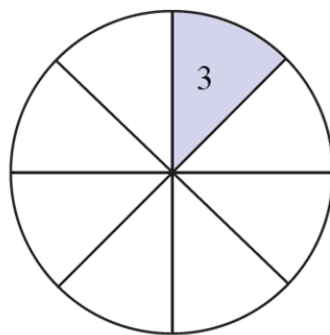
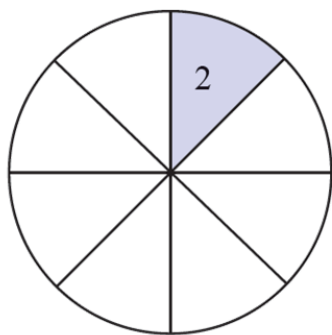
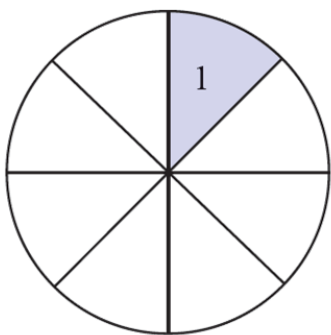


## 9.3.3 廉价磁盘冗余阵列

### RAID的技术

#### 并行交叉存取

- 系统将每一盘块中的数据分为若干个子盘块数据，再把每一子盘块的数据分别存储到不同磁盘中的相同位置上
- 当要将一个盘块的数据传送到内存时，采取并行传输方式，将各个盘块中的子盘块数据同时向内存中传输
- 传输时间大大减少



## 9.3.3 廉价磁盘冗余阵列

### RAID的分级

RAID被分成了多个不同级别。

➤ RAID0~RAID7

➤ RAID 0:

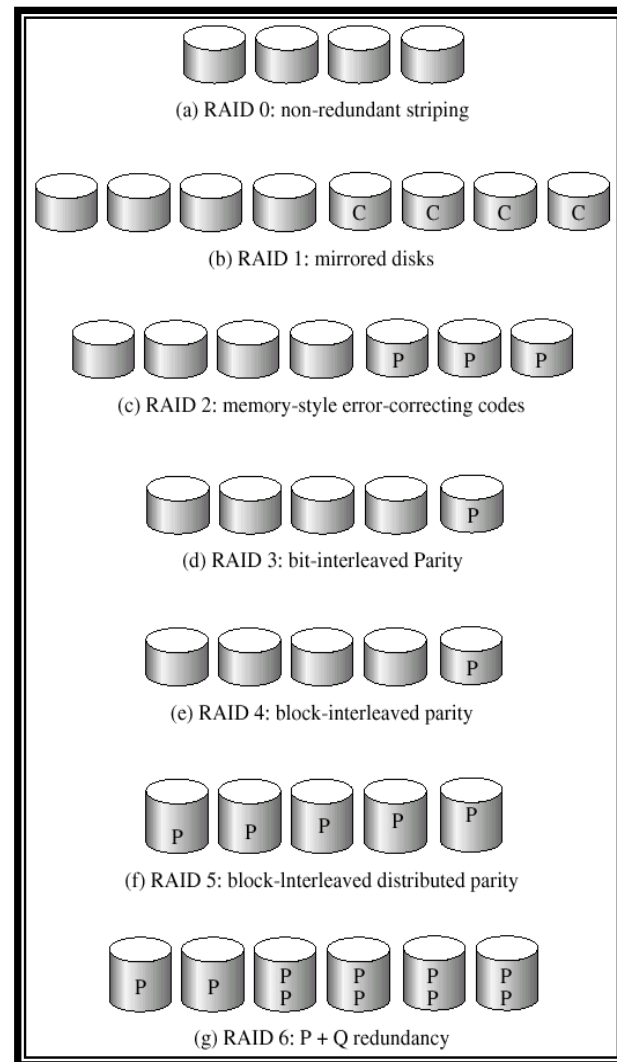
- 数据分散在多个磁盘上；
- 提高读写性能
- 条状分散技术；

➤ RAID 1:

- 磁盘镜像；
- 提高可靠性。

➤ RAID 5:

- 分散+校验；
- 校验信息分散在各个磁盘。





内容导航:

## 第9章 磁盘存储器管理

9.1 外存的组织方式

9.2 文件存储空间的管理

9.3 提高磁盘I/O速度的途径

**9.4 提高磁盘可靠性的技术**

9.5 存储新技术

9.6 数据一致性控制

## 9.4 提高磁盘可靠性的技术



### 容错技术

- 通过在系统中设置冗余部件来提高系统可靠性的一种技术
- 广泛采用磁盘容错技术



### 磁盘容错技术分为三个级别


- 第一级，低级磁盘容错技术
- 第二级，中级磁盘容错技术
- 第三级，基于集群系统的容错技术（系统容错技术SFT）

## 9.4.1 第一级容错技术SFT-I

SFT-1是最基本的磁盘容错技术，**主要用于防止因磁盘表面缺陷所造成的数据丢失。**

- OS 双份目录和双份文件分配表FAT
  - 一份为主文件目录及主FAT
  - 另一份为备份文件目录及备份FAT
- OS 热修复重定向
  - 系统将磁盘容量的很小一部分（如2%~3%）作为热修复重定向区，用于存放发现磁盘有缺陷时的待写数据
- OS 写后读校验
  - 在每次向磁盘中写入一个数据块后立即将它读出来，并送至另一缓冲区中
  - 将该缓冲区的内容与内存缓冲区中写后仍保留的数据进行比较，若两者一致，则认为此次写入成功；否则重写

## 9.4.2 第二级容错技术SFT-II(1)

 **SFT-II**主要用于防止系统因磁盘驱动器和磁盘控制器故障而无法正常工作。



为了避免磁盘驱动器发生故障而丢失数据，须在同一磁盘控制器下增设一个完全相同的磁盘驱动器



在每次向主磁盘写入数据后，都需要将数据再写到备份磁盘上，以使两个磁盘上具有完全相同的位像图



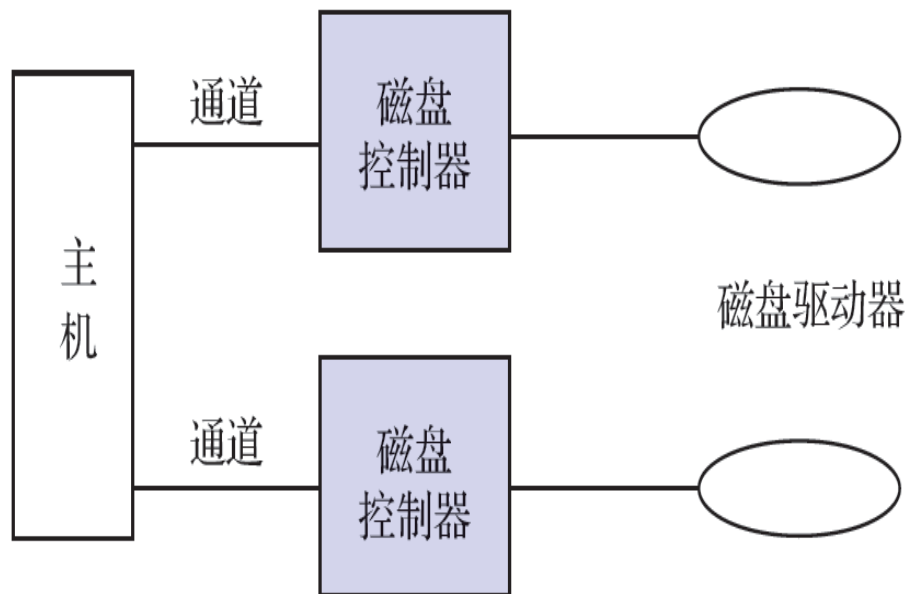
虽有容错功能，但磁盘利用率降至原来的50%

## 9.4.2 第二级容错技术SFT-II(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，同样使这两个磁盘控制器镜像成对
- 文件服务器同时将数据写到两个处于不同控制器下的磁盘上，使两者有完全相同的位像图
- 由于每个磁盘都有自己的独立通道，可同时（并行）将数据写入磁盘，或从磁盘中读出数据



## 9.4.3 基于集群系统的容错技术

所谓**集群**，是指由一组互连的自主计算机组成统一的计算机系统。



双机热备份模式



双机互为备份模式



公用磁盘模式



## 9.4.3 基于集群系统的容错技术

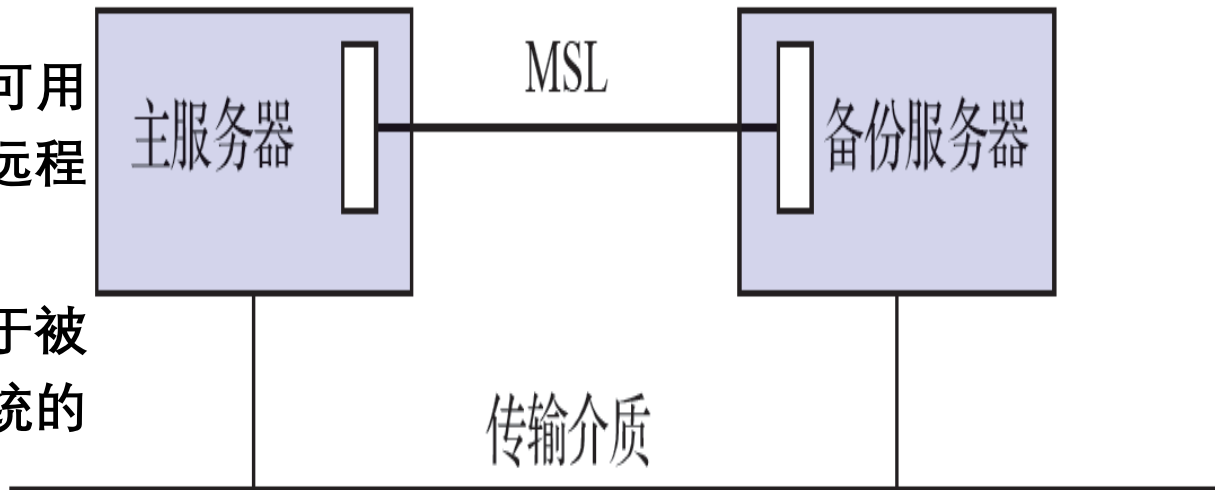
### 双机热备份

系统备有两台服务器，处理能力通常是完全相同的，**一台作为主服务器，另一台作为备份服务器。**

➤ 平时，主服务器运行，备份服务器则时刻监视着主服务器的运行情况，一旦主服务器出现故障，备份服务器便立即接替主服务器的工作而成为系统中的新的主服务器修复后的原来的主服务器作为备份服务器

➤ 优点：提高了系统的可用性，易于实现，支持远程热备份

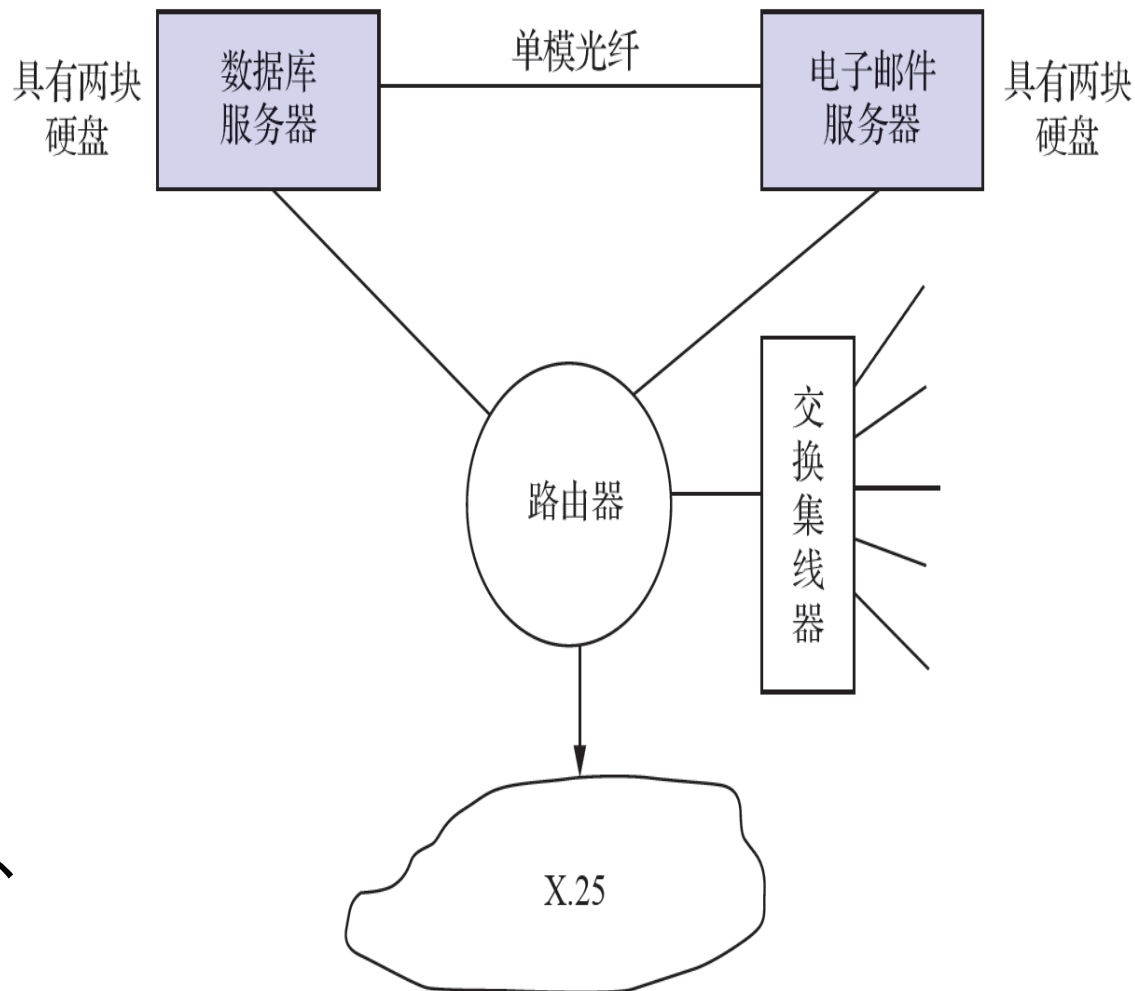
➤ 缺点：备份服务器处于被动等待状态，整个系统的使用效率只有50%



## 9.4.3 基于集群系统的容错技术

### 双机互为备份模式

- 两台服务器在平时均为在线服务器，各自完成自己的任务，两者互为备份。
- 优点是两台服务器都可用于处理任务，因而系统效率较高。
- 这种模式现已从两台机器扩大到4台、8台、16台甚至更多。



## 9.4.3 基于集群系统的容错技术

### 公用磁盘模式

- 将多台计算机连接到一台公用磁盘上，该磁盘被划分为若干个卷，每台计算机使用一个卷
- 如果某台计算机发生故障，则系统将重新进行配置，即根据某种调度策略来选择另一台机器进行替代
- **优点**是消除了信息的复制时间，因而减少了网络和服务器开销。

## 9.4.4 后备系统

**后备系统：**保存暂时不需要但仍然有用的数据

常用后备系统的设备



磁带机



硬盘



光盘驱动器



内容导航:

## 第9章 磁盘存储器管理

9.1 外存的组织方式

9.2 文件存储空间的管理

9.3 提高磁盘I/O速度的途径

9.4 提高磁盘可靠性的技术

**9.5 存储新技术**

9.6 数据一致性控制

## 9.5.1 传统存储系统

### 直连式存储DAS

- 通过总线适配器直接将硬盘等存储介质连接到主机上的存储方式，也称为主机连接存储
- 在存储设备和主机之间通常没有任何网络设备的参与

### 网络附加存储NAS

- 一种提供文件级别访问接口的网络存储系统架构，通常采用NFS、SMB/CIFS等网络文件共享协议进行文件存取

### 存储区域网络SAN

- 一种通过光纤交换机等高速网络设备在服务器和磁盘阵列等存储设备间搭设专门的存储网络

## 9.5.2 新型存储系统



### 分布式存储系统

- C/S (client/server) 架构
- P2P (peer-to-peer) 架构

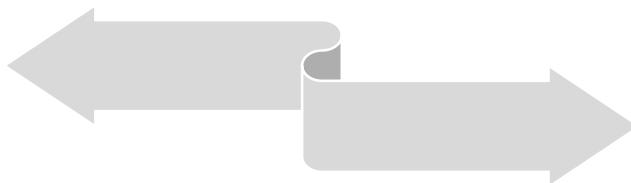


### 云存储系统

- 由第三方运营商提供的在线存储系统



## 9.5.3 硬盘新技术



### 机械磁盘（HDD）技术革新：

- 传统磁记录（CMR）技术；
- 叠瓦式（CMR）技术；
- 热辅助磁记录（HAMR）技术；
- 双磁头驱动臂技术（Mach.2）。

**固态硬盘（SSD）：**用固态电子存储芯片阵列所制成的硬盘。

- 基于闪存的固态硬盘；
- 基于DRAM的固态硬盘；
- 基于Xpoint类的固态磁盘。





内容导航:

## 第9章 磁盘存储器管理

9.1 外存的组织方式

9.2 文件存储空间的管理

9.3 提高磁盘I/O速度的途径

9.4 提高磁盘可靠性的技术

9.5 存储新技术

**9.6 数据一致性控制**

## 9.6 数据一致性控制

所谓**数据一致性**问题，是指保存在多个文件中的同一个数据，在任何情况下都必须能保证相同。

- 例如，当我们发现某种商品的进价有错时，我们必须同时修改流水账、付费账、分类账以及总账等一系列文件中该商品的价格，如此才能保证数据的一致性。
- 但如果在修改途中系统突然发生故障，则会造成各个账目中该数据的不一致性，进而就会使得多个账目不一致。

---

现代OS中都配置了能保证数据一致性的软件。

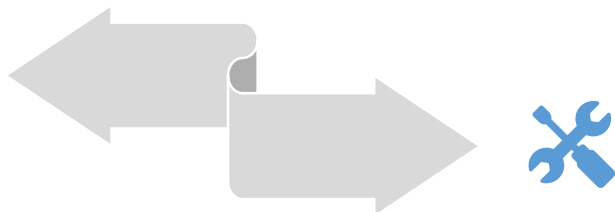
---

## 9.6.1 事务 (1)



### 事务的定义

- 事务是用于访问和修改各种数据项的一个程序单位
- 事务也可被看作一系列读/写操作
- 事务操作具有原子性



### 事务记录

- 是用于记录在事务运行过程中与数据项修改相关的全部信息数据结构，又被称为运行记录
- **事务名**：用于标志该事务的唯一名字。
- **数据项名**：它是被修改数据项的唯一名字。
- **旧值**：修改前数据项的值。
- **新值**：修改后数据项将具有的值。

## 9.6.1事务 (2)

### 恢复算法



系统利用**事务记录表**可以处理任何故障而不致使故障造成非易失性存储器中信息的丢失

- (1)  $\text{undo} \langle T_i \rangle$  : 把所有被事务 $T_i$ 修改过的数据恢复为修改前的值
- (2)  $\text{redo} \langle T_i \rangle$  : 把所有被事务 $T_i$ 修改过的数据设置为新值

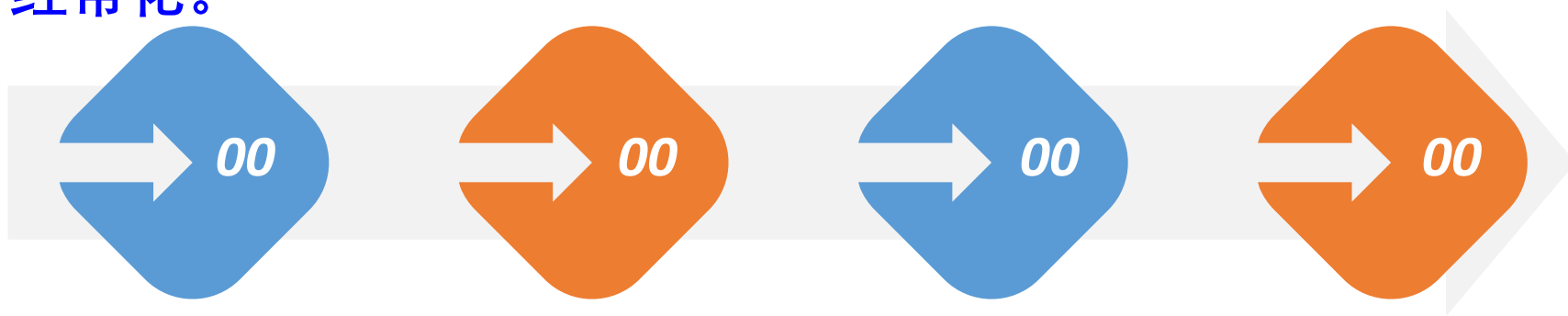


如果系统发生故障，系统通过查找**事务记录表**，对以前所发生的事务进行清理，分为两类处理：

- 一类是所包含的各类操作都已完成的事务，系统会利用 $\text{redo} \langle T_i \rangle$ 过程把所有已被修改的数据设置成新值
- 另一类是所包含的各个操作并未全部完成的事务，系统会利用 $\text{undo} \langle T_i \rangle$ 过程将所有已被修改的数据恢复为修改前的值

## 9.6.2 检查点

引入**检查点**的主要目的，使事务记录表中事务记录的清理工作经常化。



- 首先，将驻留在易失性存储器（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；
- 其次，将驻留在易失性存储器中的所有已修改数据输出到稳定存储器中；
- 再次，将事务记录表中的〈检查点〉记录输出到稳定存储器中；
- 最后，每当出现一个〈检查点〉记录，系统便执行一次恢复操作。

## 9.6.3 并发控制



由于事务具有原子性，各个事务的执行必然会按某种次序依次执行，即各事务对数据项的修改是互斥的（称为顺序性）



用于实现事务顺序性的技术称为并发控制

- 利用互斥锁实现顺序性
- 利用互斥锁和共享锁实现顺序性

## 9.6.4 重复数据的一致性问题的



主文件或数据结构中的数据应与各备份文件中对应的数据保持一致；应保证不同处的同一数据结构中（**可能已被修改的**）数据的一致性。

- **重复文件的一致性**：当有重复文件时，如果一个文件复制被修改，则必须同时修改其他几个文件复制，以保证各相应文件中数据的一致性。
- **链接数一致性检查**：共享文件的索引节点数与链接计数 count 应一致。

# 第9章总结

## 磁盘存储器管理



本章学习  
结束





内容导航:

## 第9章 磁盘存储器管理

9.1 外存的组织方式

9.2 文件存储空间的管理

9.3 提高磁盘I/O速度的途径

9.4 提高磁盘可靠性的技术

9.5 存储新技术

9.6 数据一致性控制



## 第9章 磁盘存储器管理

### 作业

#### 1、课后作业

- 简答：1、5、7
- 计算题：13、15、16

#### 2、课后练习

- 简答：4、6
- 计算题：16
- 综合：21



# 课程思政小讲堂

## 新一轮科技变革之际

### 国产OS如何依托工业互联网实现“换道超车”？

2019年，国际政治关系的变化导致华为技术有限公司（以下简称华为）面临紧迫的Android（安卓）系统停供风险。在此背景下，华为毅然推出自主研发的操作系统——鸿蒙系统（HarmonyOS），以尽量减少Android系统停供所造成的损失。鸿蒙系统一经问世便引发产业界高度关注，这充分反映出大家对优秀国产操作系统的期盼。

近年来，我国软件产业的规模和效益同步提升，企业实力和创新能力强显著增强。工业和信息化部数据显示，2018年，软件著作权登记量突破110万件，同比增长48%；2019年，软件业务收入已达71 768亿元，同比增长15.4%。由此可知，软件业的发展为国产操作系统的研发提供了基础性的技术支持。



## 课程思政小讲堂

### 新一轮科技变革之际

### 国产OS如何依托工业互联网实现“换道超车”？

随着云计算、大数据、人工智能等新一代信息技术与实体经济深度融合，工业互联网、工业大数据等新模式、新业态快速发展，工业已成为世界各国重点布局的领域。我国作为全世界唯一拥有联合国产业分类中所列全部工业门类的国家，在工业领域发展操作系统可谓具有独特优势。国内大量处在工业2.0、3.0和4.0不同发展阶段的企业通过“企业上云”，加速实现数字化和网络化，伴随着工业互联网基础设施的快速迭代升级，国产操作系统不仅具有广阔的市场空间，也将推动传统制造业加速向数字化、网络化和智能化转变，为国产操作系统“换道超车”提供重大历史机遇。



## 课程思政小讲堂

### 新一轮科技变革之际

#### 国产OS如何依托工业互联网实现“换道超车”？

坚信，通过①以工业互联网发展为动力推动国产操作系统在工业领域的应用和普及，②以开源为抓手提升国产操作系统技术水平，③以创新为引擎促进形成发展的内生动力，④打造以国产操作系统为核心的软硬件生态体系，国产操作系统一定能够在新一轮科技变革之际，依托工业互联网实现“换道超车”！



海南大学  
HAINAN UNIVERSITY



祝大家考试顺利!