



# 第10章 文件系统





# 文件系统

- **文件系统**：操作系统中与管理文件有关的软件和数据集合。
- 文件系统的组成
  - 管理文件所需的数据结构：
    - 如目录：用于组织系统内所有文件并提供文件信息
  - 被管理的文件集合
    - 每个文件存储相关数据
  - 相应的管理软件
    - 存储设备空间的分配与回收
    - 目录与文件管理
    - 用户接口
    - 共享与保护



# 目录

- 10.1 文件的基本概念
- 10.2 文件访问方法
- 10.3 文件目录结构
- 10.4 文件系统的安装
- 10.5 文件共享
- 10.6 文件保护

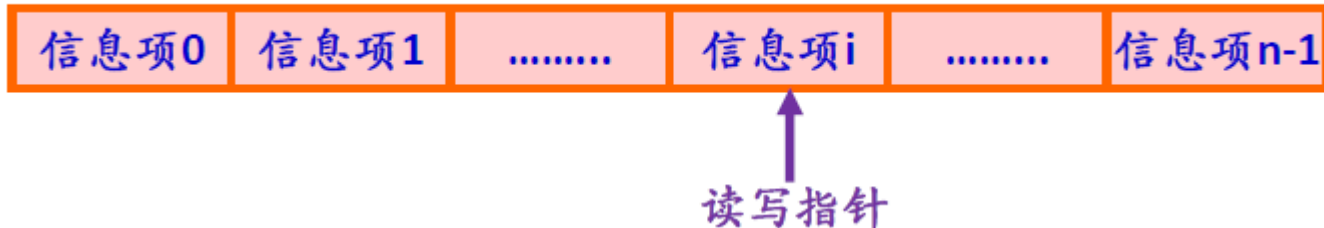




# 1.文件的定义

## ■ 文件

- 是记录在外部存储上，具有名字的，在逻辑上有完整意义的信息项的序列。
- 信息项：构成文件内容的基本单位（单个字节，或多个字节），各信息项之间具有顺序关系
- 文件内容的意义：由文件建立者和使用者解释
- 从用户角度看，文件是逻辑外存的最小分配单元。
- 是文件系统的基本数据单位，即数据保存在文件中





## 2. 文件分类

### ■ 文件的分类

#### ■ 按用途

- 系统文件、库文件、用户文件

#### ■ 按保护级别

- 只读文件、读写文件、执行文件、不受保护文件

#### ■ 按信息流向

- 输入文件、输出文件、输入输出文件

#### ■ 按数据形式

- 源文件、目标文件、可执行文件



# 3. 文件属性

## ■ 文件的属性，因OS而异

- 名称：符号文件名，唯一的，且按照方便人类读取的形式保存
- 标识符：文件系统中对应文件的唯一标签，数字形式为主，不易于人类读取
- 类型：支持不同类型文件的系统所需要
- 位置：在设备上的文件位置指针
- 大小：当前文件的大小，以及可能允许的最大尺寸
- 保护：访问控制信息确定谁能读、写和执行
- 时间、日期、用户标识：文件创建、最后修改、最后使用等信息可被保存，用于保护、安全和使用监控

## ■ 所有关于文件的信息都保存于**目录结构**中，并在外存上长期存储



## 4. 文件操作

- 文件是一个抽象数据类型，定义必须的操作
  - Create：在文件系统中找到空间，在目录中创建条目
  - Write：维护写位置指针，并通过系统调用向相应位置写入内容
  - Read：维护读位置指针，并通过系统调用读取相应内容
  - Seek：搜索目录条目，并设置文件指针为给定值
  - Delete：给定文件，删除文件，释放外存空间，删除目录条目
  - Truncate：删除文件内容，但保留文件属性
  - .....
- 为避免频繁搜索文件，系统提供统一的系统调用
  - Open：在磁盘上搜索目录结构，并将所搜索到的文件目录内容移入内存
  - Close：将对应文件的目录内容移出内存，回到磁盘目录结构中



```
/* File copy program. Error checking and reporting is minimal. */
```

```
#include <sys/types.h>
```

```
/* include necessary header files */
```

```
#include <fcntl.h>
```

```
#include <stdlib.h>
```

```
#include <unistd.h>
```

## 用基本文件操作构造其他操作示例

```
int main(int argc, char *argv[]);
```

```
/* ANSI prototype */
```

```
#define BUF_SIZE 4096
```

```
/* use a buffer size of 4096 bytes */
```

```
#define OUTPUT_MODE 0700
```

```
/* protection bits for output file */
```

```
int main(int argc, char *argv[])
```

```
{
```

```
    int in_fd, out_fd, rd_count, wt_count;
```

```
    char buffer[BUF_SIZE];
```

```
    if (argc != 3) exit(1);
```

```
/* syntax error if argc is not 3 */
```

```
/* Open the input file and create the output file */
```

```
    in_fd = open(argv[1], O_RDONLY);
```

```
/* open the source file */
```

```
    if (in_fd < 0) exit(2);
```

```
/* if it cannot be opened, exit */
```

```
    out_fd = creat(argv[2], OUTPUT_MODE);
```

```
/* create the destination file */
```

```
    if (out_fd < 0) exit(3);
```

```
/* if it cannot be created, exit */
```





```
/* Copy loop */
while (TRUE) {
    rd_count = read(in_fd, buffer, BUF_SIZE); /* read a block of data */
    if (rd_count <= 0) break; /* if end of file or error, exit loop */
    wt_count = write(out_fd, buffer, rd_count); /* write data */
    if (wt_count <= 0) exit(4); /* wt_count <= 0 is an error */
}

/* Close the files */
close(in_fd);
close(out_fd);
if (rd_count == 0) /* no error on last read */
    exit(0);
else
    exit(5); /* error on last read */
}
```



# 打开文件Open()

- 在打开文件过程中需要维护许多的数据
  - 打开文件表(Open-file Table):
    - 跟踪打开的文件
    - 当涉及多进程访问环境, 系统会维护进程表 (局部) + 系统表 (全局)
  - 文件指针:
    - 指向最近的读写位置, 每个进程都维护自己的文件指针
  - 打开文件计数:
    - 文件被打开次数的计数, 允许当最后一个进程关闭文件时从文件打开表中移除相关数据
  - 文件的磁盘位置:
    - 便于数据访问, 避免每次修改操作时从磁盘中获取相关信息
  - 访问权限:
    - 为每个进程提供访问模式



## 5. 文件类型

- OS需要识别和支持文件类型，方便以合理方式操作文件。

file type	usual extension	function
executable	exe, com, bin or none	ready-to-run machine-language program
object	obj, o	compiled, machine language, not linked
source code	c, cc, java, pas, asm, a	source code in various languages
batch	bat, sh	commands to the command interpreter
text	txt, doc	textual data, documents
word processor	wp, tex, rtf, doc	various word-processor formats
library	lib, a, so, dll	libraries of routines for programmers
print or view	ps, pdf, jpg	ASCII or binary file in a format for printing or viewing
archive	arc, zip, tar	related files grouped into one file, sometimes compressed, for archiving or storage
multimedia	mpeg, mov, rm, mp3, avi	binary file containing audio or A/V information



## 6. 文件的结构

- 文件结构指文件的组织形式，文件有两种形式的结构：
  - 逻辑结构：
    - 又称文件组织，是**从用户观点出发所看到的文件组织形式**。
  - 物理结构：
    - 又称文件的存储结构，是**文件在外存上的存储组织形式**。它与存储设备特性、外存分配方式有关。
    - 数据块是逻辑存储单元，扇区是物理存储单元。



# 文件的逻辑结构

- 文件的逻辑结构又称文件组织，是从用户观点出发所看到的文件组织形式。
- 可分为两类：
  - 记录式文件：是一种有结构文件，由一组相关记录组成。又分为：
    - 等长记录文件：又称定长记录文件，是指文件中所有记录的长度相等。
    - 变长记录文件：是指文件中各记录长度不相等。
  - 流式文件：是一种无结构文件，由字符序列构成。

<https://zhuanlan.zhihu.com/p/537716594>



# 文件的物理结构

(和外存分配方式结合学习)

- 文件的物理结构与存储介质的存储特性及外存分配方式有关。
  - 文件存储设备通常划分为大小相等的物理块，物理块是分配及传输信息的基本单位。**物理块的大小与设备有关**，但**与逻辑记录的大小无关**，因此一个物理块中可以存放若干个逻辑记录，一个逻辑记录也可以存放在若干个物理块中。
- 为有效地利用外存设备和便于系统管理，**一般**也把文件信息划分为与物理存储块**大小相等**的逻辑块。



# 物理结构类型

- 常见的文件物理结构有以下几种形式：
  - 顺序结构
  - 链接结构
  - 索引结构





# 顺序结构

- 顺序结构又称连续结构，它将一个在逻辑上连续的信息存放在外存连续的物理块中。
- 以顺序结构存放的文件称为顺序文件或连续文件。
- 特点：顺序存取速度较快；对等长记录文件**支持随机访问**。但因要求连续存放，**会产生碎片**，同时也不利于文件的动态扩充。





# 链接结构

- 链接结构又称串联结构，它将一个逻辑文件的信息存放在外存**不连续物理块**中，且在每个物理块中设置一个指向下一个物理块的指针。
- 采用链接结构存放的文件称为链接文件或串联文件。
- 特点：可**解决碎片问题**，便于文件动态增长。但只能**顺序访问**，因而查找效率较低，指针占用存储空间。



# 索引结构

- 索引结构：将一个逻辑文件的信息存放于外存的若干个物理块中，并为每个文件**建立一个索引表**，其中的每个表目存放文件信息所在的逻辑块号和与之对应的物理块号。
- 采用索引结构存放的文件称为**索引文件**。
- 特点：既可以顺序访问也可以随机访问，但增加了存储空间开销，且要**两次访问外存**。



# 目录

- 10.1 文件的基本概念
- 10.2 文件访问方法
- 10.3 文件目录结构
- 10.4 文件系统的安装
- 10.5 文件共享
- 10.6 文件保护



## 10.2 文件访问方法

- 常用的文件访问方法有：
  - 顺序访问法
  - 直接访问法
  - 索引访问法

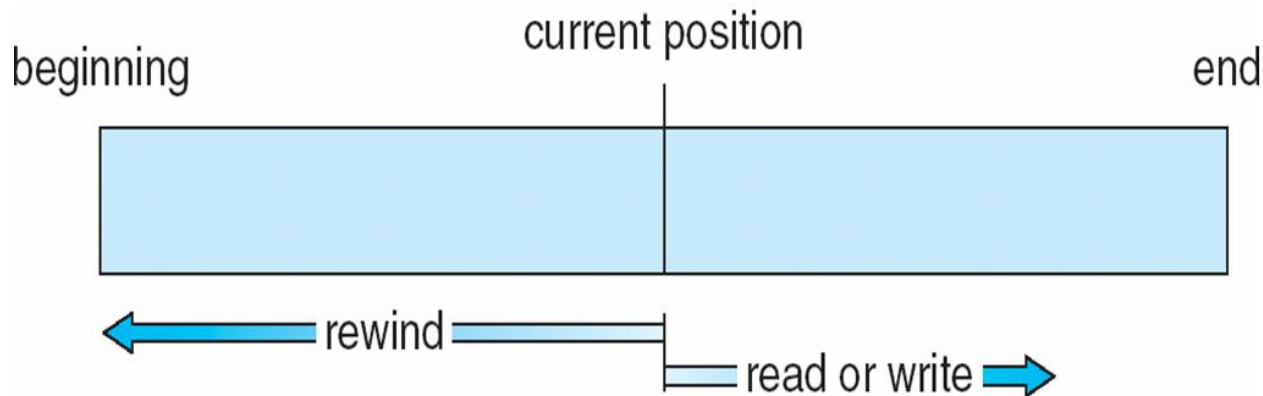


# 1. 顺序访问法

- 也称顺序存取法，是按照文件信息的逻辑顺序依次存取。
- 在记录式文件中，顺序存取反映为按记录的排列顺序来存取；在流式文件中，顺序存取反映为当前读写指针的变化。
- 对定长记录的顺序文件，若知道当前记录地址，则很易确定下一个记录地址。

$$rptr = rptr + L$$

- 其中L为文件记录的长度，rptr为读写指针。





## 2. 直接访问法

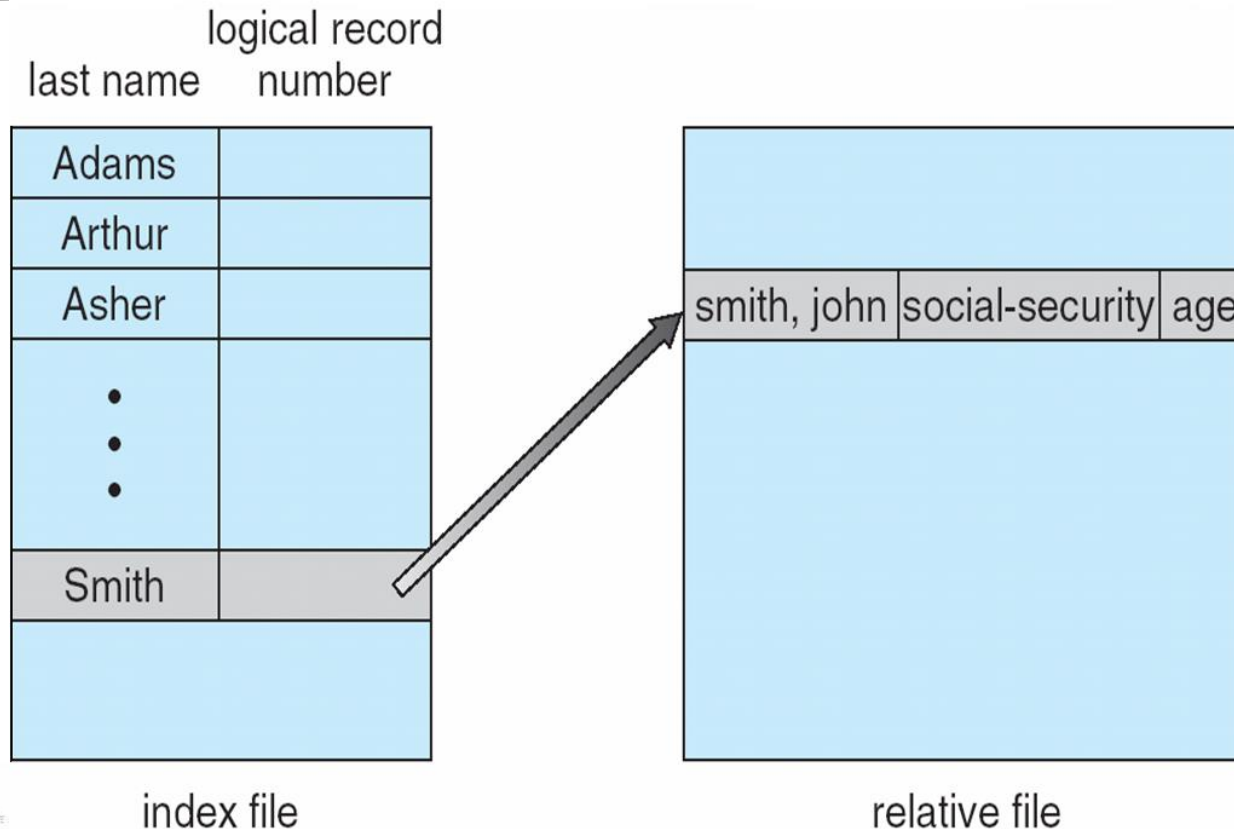
- 又称随机存取法，允许按记录的编号来存取任何一个记录，或根据读写命令把存取指针移到指定位置来读写。
- 对于定长记录的顺序文件，若知道文件的起始地址和记录长度，则第*i*个记录 ( $i = 0, 1, 2, \dots$ ) 的首地址为
  - $$rptr = addr + i \times L$$
  - 其中addr是该文件的首地址，L为记录长度。
- 直接访问法可模拟顺序访问法

sequential access	implementation for direct access
<i>reset</i>	$cp = 0;$
<i>read next</i>	$read\ cp;$ $cp = cp + 1;$
<i>write next</i>	$write\ cp;$ $cp = cp + 1;$



### 3. 索引访问法

- 索引访问法实质上也是直接存取法，它根据文件记录中数据项（通常称为关键字），经过某种方法计算处理，转换成相应的物理地址后进行存取。
- 也可通过关键字建立索引，按照索引来进行访问。





# 存储设备、存取方法和物理结构

- 文件存取方法与文件的物理结构、存储设备有关

存储设备	磁 盘			磁 带
物理结构	顺序结构	链接结构	索引结构	顺序结构
存取方法	顺序、直接	顺序	顺序、直接	顺序





# 目录

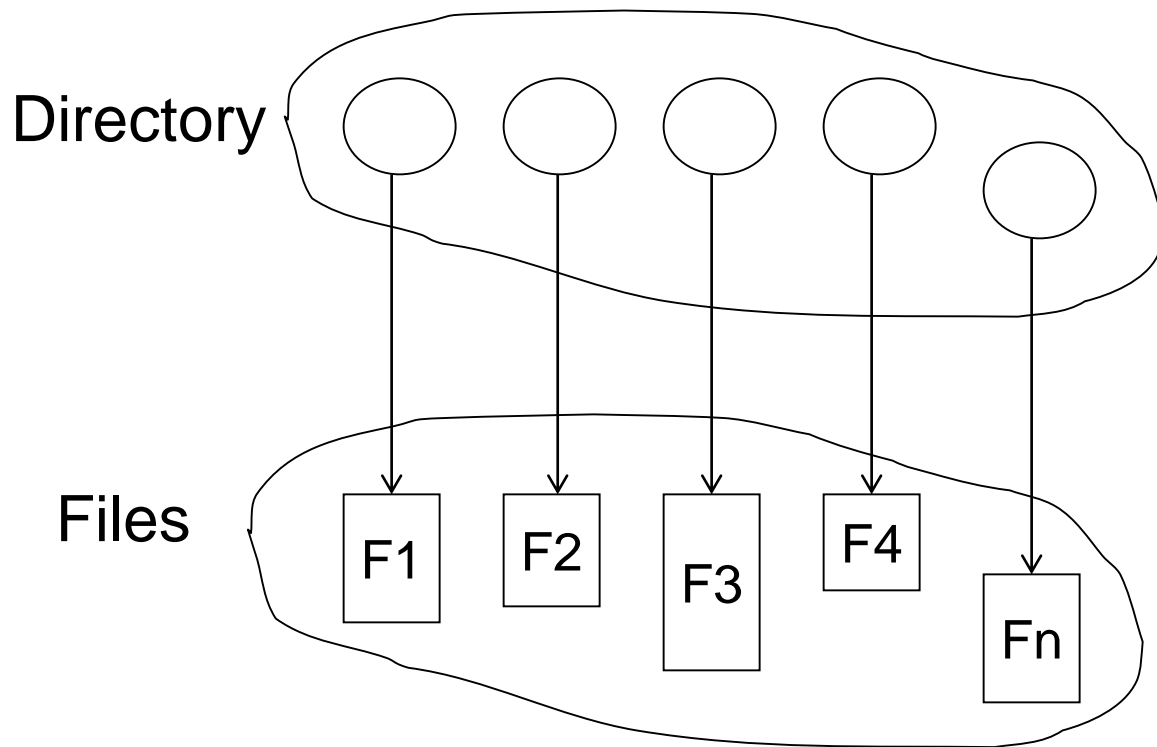
- 10.1 文件的基本概念
- 10.2 文件访问方法
- 10.3 文件目录结构
- 10.4 文件系统的安装
- 10.5 文件共享
- 10.6 文件保护





# 文件目录

- 计算机系统中的文件种类繁多，数量庞大，为了有效地管理这些文件，方便用户查找所需的文件，应对它们加以适当的组织。文件的组织可以通过目录来实现。



目录结构和文件都驻留在磁盘上



# 1. 文件控制块与文件目录

- 从文件管理的角度看，文件由文件说明和文件体两部分组成。
- 文件体即文件本身。文件说明是保存文件属性信息的数据结构，又称为文件控制块（FCB, File Control Block）。
- 文件与FCB一一对应，FCB是文件存在的标志。
- FCB的有序集合称为文件目录，即一个FCB就是一个文件目录项。
- 通常，将文件目录以文件的形式存储在磁盘上，称为目录文件。



# 文件控制块的内容

- 文件名：标识一个文件的符号名。
- 文件类型：如文本文件。
- 文件结构：说明文件的逻辑结构和物理结构。
- 文件的物理位置：指示文件在外存上的存储位置。包括设备名、存储地址及文件长度等。
- 存取控制信息：指示文件的存取权限。
- 管理信息：包括文件建立的日期及时间、上次存取日期及时间、当前文件使用状态信息。



# DOS的文件控制块

- 文件名及扩展名占11字节；属性占1字节，包括只读、隐藏、系统、卷标、子目录及归档；上次存取时间及日期各占2个字节；文件起始簇号占2个字节；文件长度占4个字节。
- FCB的长度为32字节。

文件名
扩展名
文件属性
备用
上次访问时间
上次访问日期
文件起始簇号
文件长度



# 文件目录及目录文件

- **文件目录**：文件控制块的集合。即文件控制块是一个目录项。
- **目录文件**：将文件目录以文件的形式存放在磁盘上，即文件的内容为目录信息。





## 2. 目录管理

- 目录可视为符号表，可将文件名称转成目录条目
- 对目录管理的要求如下：
  - (1) 实现“按名存取”。
  - (2) 提高对目录的检索速度。
  - (3) 文件共享。
  - (4) 允许文件重名。



# 目录上的基本操作

- 目录上的基本操作
  - 搜索文件
  - 创建文件
  - 删除文件
  - 查看目录
  - 重命名文件
  - 遍历文件系统





# 目录组织的目标

- 高效性 —— 快速定位一个文件
- 命名 —— 方便用户使用
  - 两个用户对于不同文件可以起相同的名字
  - 相同的文件也可以有多个不同的名字
- 分组 —— 文件可根据不同属性进行逻辑分组



# 3. 目录结构

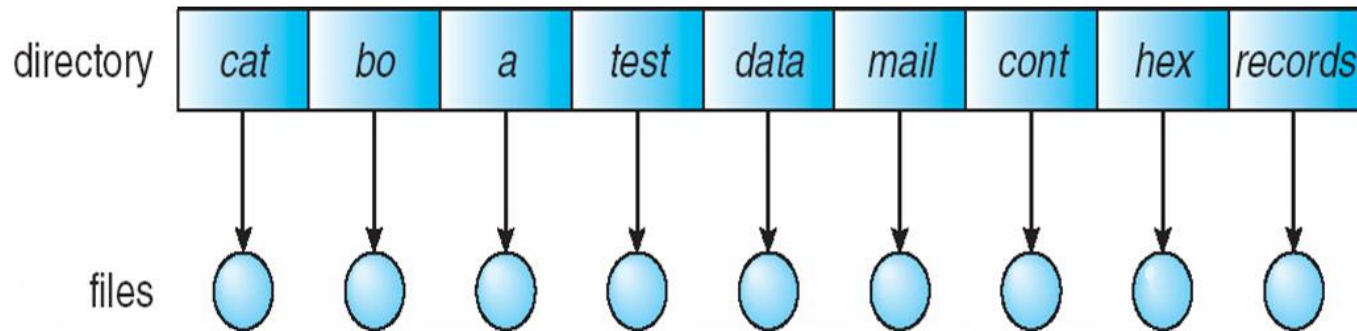
- 常用的文件目录结构有：
  - 单级目录结构
  - 二级目录结构
  - 多级目录 结构





# (1) 单级目录结构

- 单级目录结构又称一级目录结构。在这种结构中，整个文件系统只建立一个目录，从实现上，每个文件占据一个目录的项。





# 单级目录结构的操作

## ■ 建立

- 先应确定该文件名在目录中是否惟一，若惟一则找出一个空表目，将新文件的相关信息填入其中。

## ■ 删除

- 先从目录表中找到文件的目录项，从中找到该文件的物理地址，对其占用空间进行回收，然后再清除其所占用的目录项。

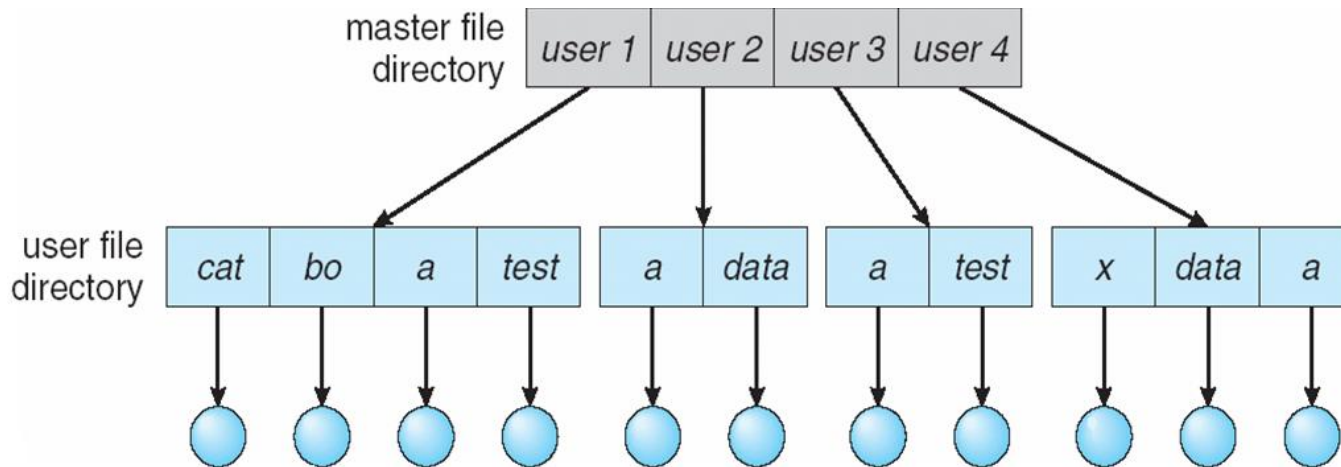
## ■ 特点

- 易于实现，管理简单；
- 当文件数增多时，查找时间较长
- 不支持重名问题



## (2) 二级目录结构

- 二级目录结构将文件目录分成：
  - 主文件目录MFD：记录用户名及相应用户文件目录所在的存储位置。
  - 用户文件目录UFD：记录该用户文件的有关信息。





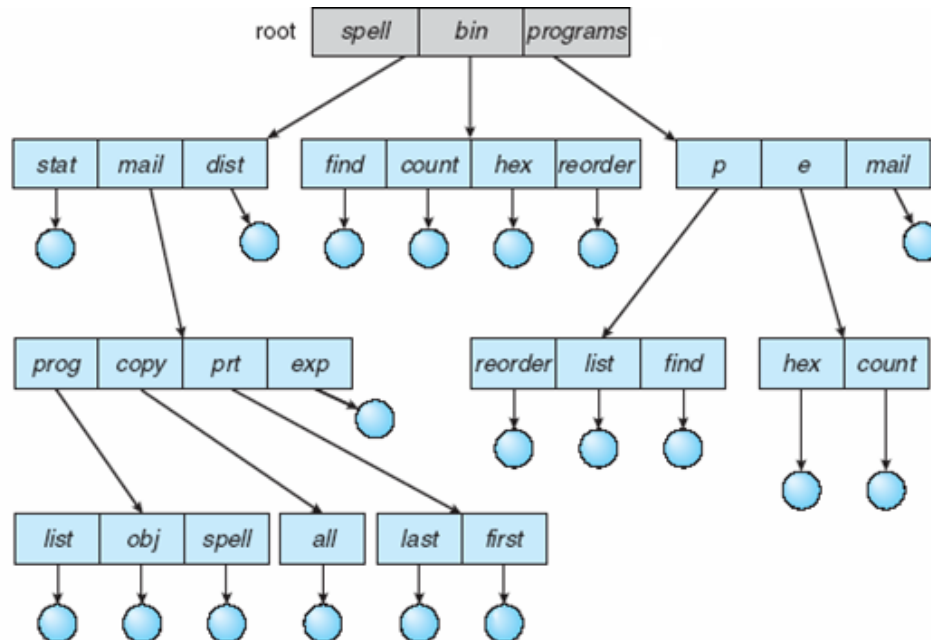
# 二级目录结构的特点

- 优点：有效实现了用户之间的隔离
  - 重命名问题
    - 不同MFD下，文件名可重复
    - 相同UFD下，文件名不可重复
- 缺点：隔离不适合合作共享
  - 用户文件的共享：需了解完整路径，支持额外语法
    - /MFD/UFD/Filename,
    - [partition name]:[directory name]:[Filename]
  - 系统文件的共享：加载器、编译器、库函数、工具程序等
    - 访问方式：给定一条命令，即文件名称，而非整条路径
    - 方法1：将所有系统文件拷贝到每个UFD
    - 方法2：建立特殊目录，建立搜索路径



### (3)多级目录结构

- 多级目录结构是二级目录结构层次关系的推广，也称为**树形目录结构**。
- 在多级目录结构中，第一级目录称为根目录（树根），目录树中的非叶节点均为目录文件（又称子目录），叶结点为文件。





# 多级目录结构 (Cont.)

## ■ 路径名

在树形目录结构中，从根目录到任何数据文件，都只有一条惟一的通路。

- 从树根开始的路径名称为**绝对路径名**(absolute path name)。由从根目录出发到所找文件的通路上所有各级子目录名和该文件名用分隔符连接起来构成。

- /spell/mail/prog/filename

- 从当前目录开始直到所找文件为止所构成的路径名，称为**相对路径名**(relative path name);

- ./prog/filename (当前目录为/spell/mail)





# 多级目录结构 (Cont.)

- 当前目录(Current Directory)。
- 可为每个进程设置一个“当前目录”，又称为“工作目录”。进程对各文件的访问都相对于“当前目录”而进行。此时各文件所使用的路径名，只需从当前目录开始，逐级经过中间的目录文件，最后到达要访问的数据文件。
- 有两个特殊目录：
  - “..”：表示给定目录的父目录
  - “.”：表示当前目录



# 多级目录结构 (Cont.)

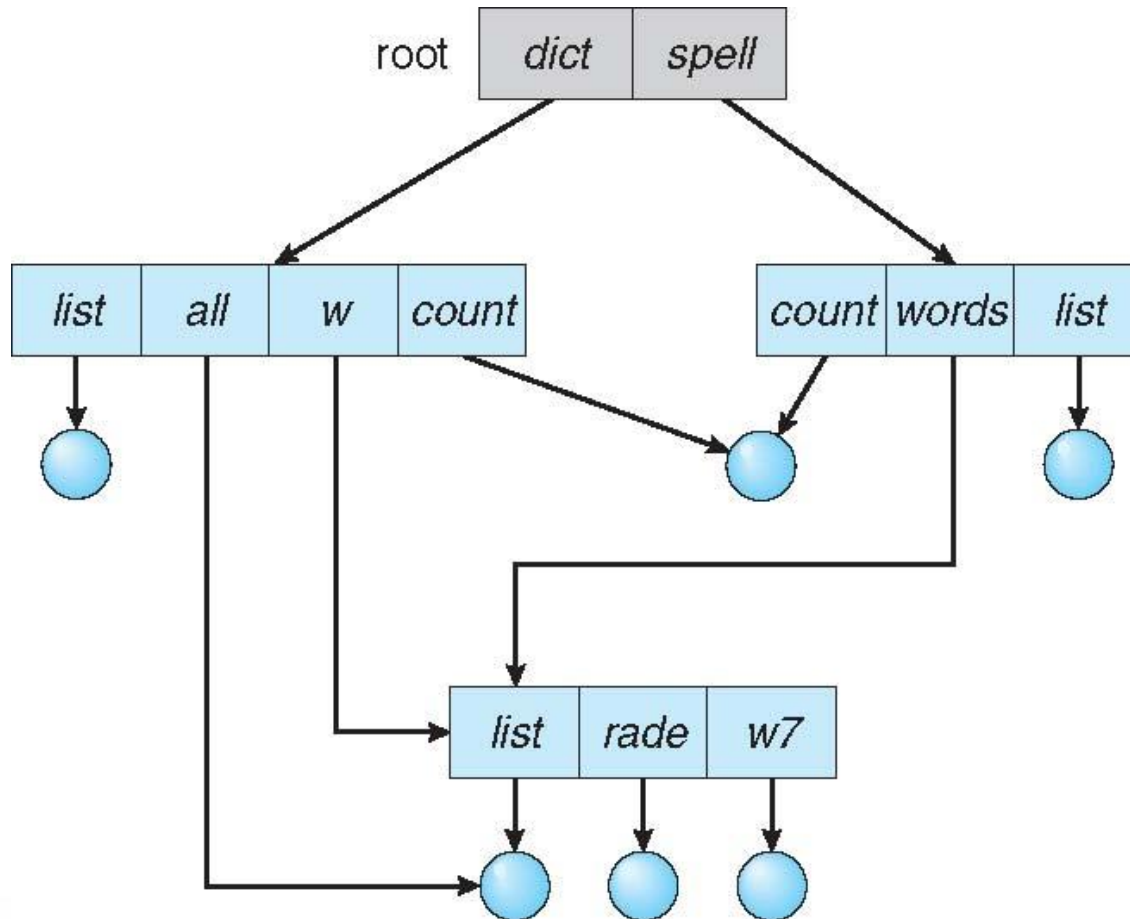
## ■ 增加和删除目录

- (1) 不删除非空目录。当目录(文件)不空时，不能将其删除，必须先删除目录中的所有文件，使之成为空目录，然后再予以删除。如果目录中还包含有子目录，还必须采取递归调用方式来将其删除。MS-DOS
- (2) 可删除非空目录。要删除一目录时，如果在该目录中还包含有文件，则目录中的所有文件和子目录也被删除。UNIX rm



## (4) 无环图目录

- 树型不允许共享文件和目录，无环图允许共享子目录和文件





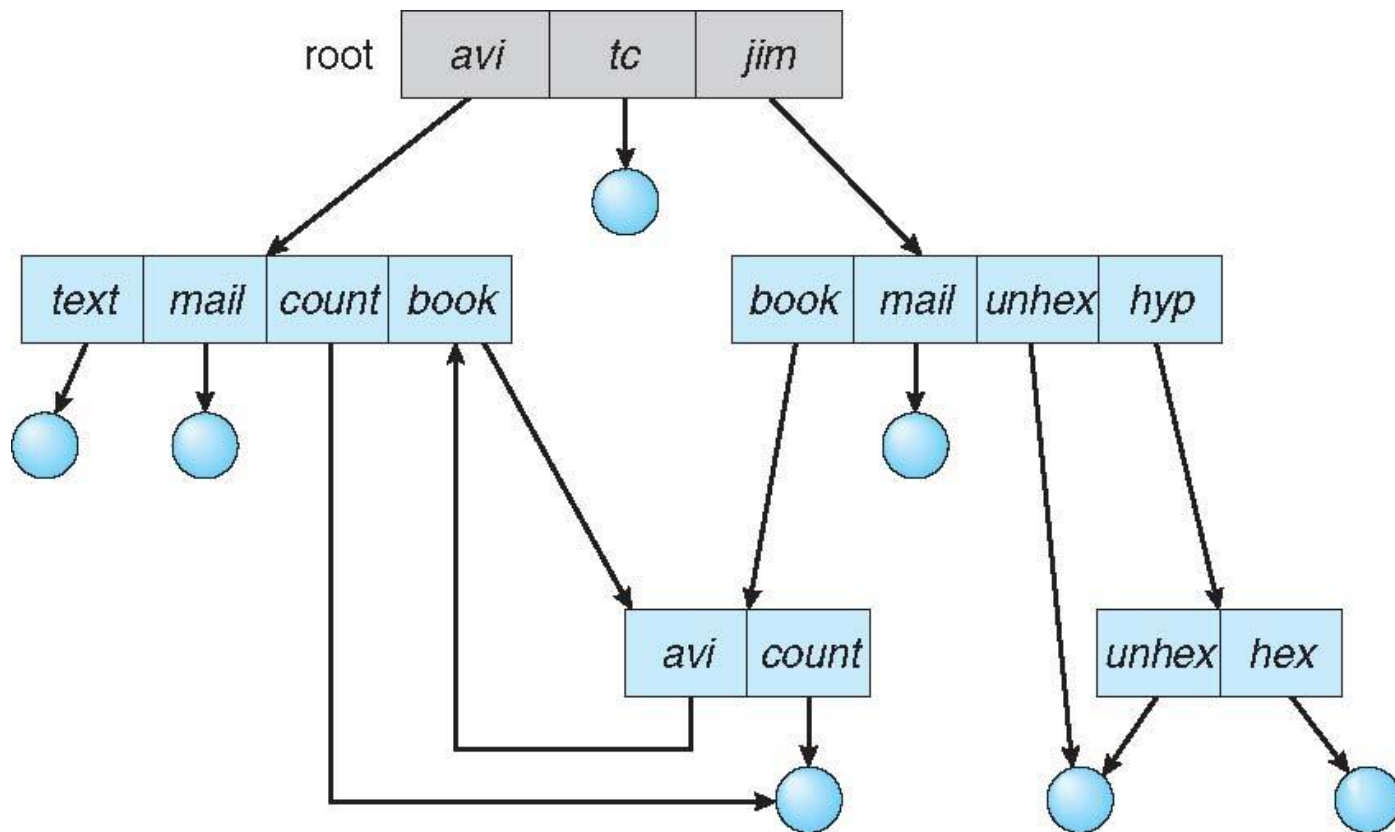
# 无环图目录 (Cont.)

- 支持别名
- 共享方法：Link,
  - 链接：创建一个指向已存在文件的指针
  - 解析：沿指针定位文件
- 共享文件的删除
  - 问题：
    - 容易产生指针悬挂：从/dict/conunt下删除文件
  - 方法一：对于符号链接实现的共享
    - 只删除链接指针，不删除原始文件
    - 若原文件被删除了，指针依然悬挂，则需要检查机制
  - 方法二：保留文件，直到删除所有引用
    - 需引入引用列表，或引用计数



## (5) 通用图目录

- 当对已存在的树状结构不断增加链接，就形成了图结构



- 如何避免循环搜索，限制搜索深度
- 如何删除：建立垃圾收集机制 or 遍历目录时规避链接



# 目录

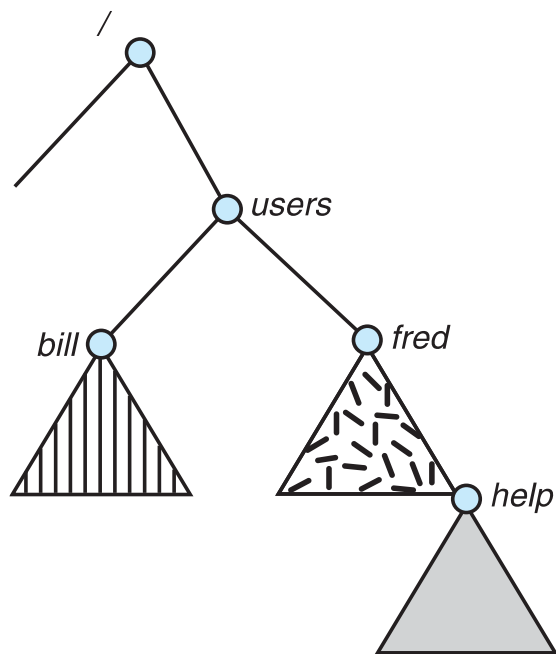
- 10.1 文件的基本概念
- 10.2 文件访问方法
- 10.3 文件目录结构
- 10.4 文件系统的安装
- 10.5 文件共享
- 10.6 文件保护





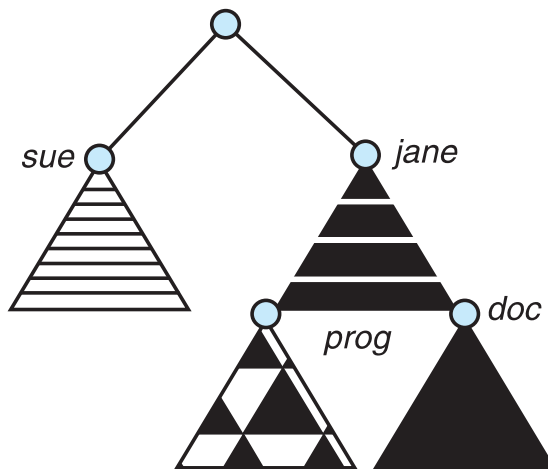
## 10.4 文件系统的安装

- 一个文件系统在访问前需要安装挂载，mount



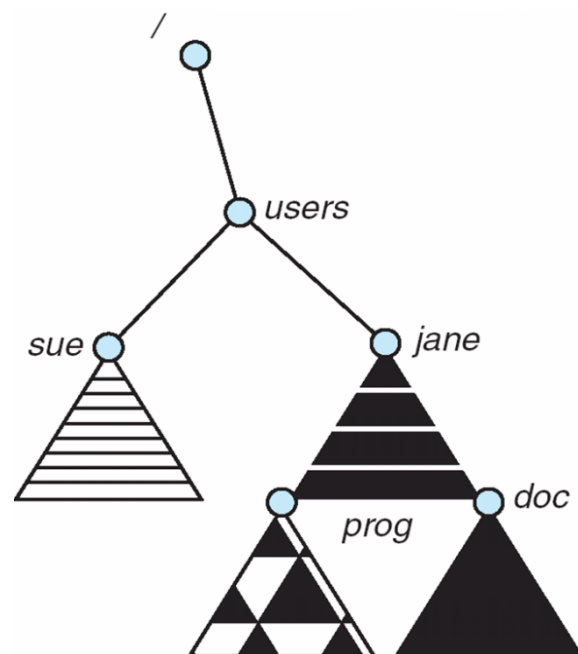
(a)

已有文件系统



(b)

未安装的卷



安装点



# 目录

- 10.1 文件的基本概念
- 10.2 文件访问方法
- 10.3 文件目录结构
- 10.4 文件系统的安装
- 10.5 文件共享
- 10.6 文件保护







## 10.5 文件共享

### ■ 目标：

- 多用户系统共享文件
- 文件共享时需要考虑保护机制
- 对于分布式系统，文件可以通过网络共享，如NFS



# (1) 多用户

- 为了实现文件的共享和保护，多用户系统通常采用了文件用户（所有者）和组的概念。
  - UserID
  - GroupID
  - 文件Owner
  - 文件Group





## (2) 远程文件系统

- 使用网络实现系统间文件访问的方法
  - 手工程序实现，如FTP
  - 自动化的，无缝实现：分布式文件系统DFS
  - 半自动化的通过万维网实现
  - 云计算技术实现云文件共享



## (2) 远程文件系统

- 所有的文件系统都包含故障模式Failure Mode
  - 本地文件系统故障原因：磁盘、目录结构、磁盘管理信息、磁盘控制器、线缆、适配器故障、目录卷丢失等。
  - 远程文件系统引入了新故障模式
    - 网络故障、服务器故障
- 故障恢复
  - 涉及每个远程请求状态的状态信息
  - 无状态DFS， 有状态 DFS



## (3) 共享中的一致性语义

- 描述多个用户如何同时访问共享文件
  - 规定了一个用户数据被修改，何时另一个用户可见？
  - 进程间同步算法不适合解决此问题
    - 磁盘I/O和网络延迟所导致，复杂算法无法有效工作
  - Andrew File System (AFS)会话语义
    - 写操作，对于打开同一文件的其他用户不可见
    - 一旦文件关闭，写操作仅对后来打开的会话可见，已打开的文件实例不反映变化
  - Unix file system (UFS)
    - 写操作对其他打开同一文件的用户是立即可见的
    - 允许多用户在读写时共享文件指针
  - 不可变共享文件语义
    - 一旦一个文件由创建者声明为共享，他就不能被修改。
    - 不可变文件：名称不可重用、内容不可改变



# 目录

- 10.1 文件的基本概念
- 10.2 文件访问方法
- 10.3 文件目录结构
- 10.4 文件系统的安装
- 10.5 文件共享
- 10.6 文件保护

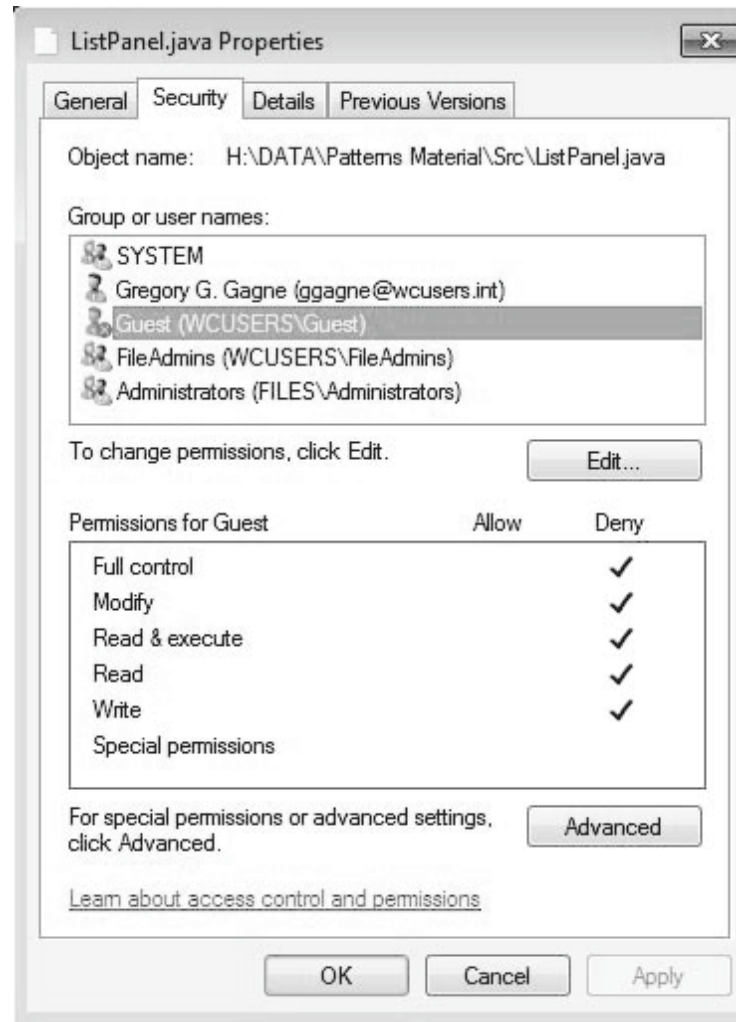


## 10.6 文件保护

- 文件保护：防止文件被破坏。包括：
  - 问题1：防止系统崩溃所造成的文件破坏；
  - 问题2：防止文件所有者和其他用户有意或无意的非法操作所造成的文件破坏。
- 防止问题一：
  - 定时转储
  - 多副本
- 防止问题二
  - 访问控制——基本思想是建立三元组：（用户、对象、存取权限）



# Windows 7 Access-Control List Management







# Unix

- 访问模式: read, write, execute
- Unix / Linux三类用户

			RWX
a) owner access	7	⇒	1 1 1
			RWX
b) group access	6	⇒	1 1 0
			RWX
c) public access	1	⇒	0 0 1

- 修改权限: chmod
- 修改组关系: chgrp



# Unix 目录列表

-rw-rw-r--	1 pbg	staff	31200	Sep 3 08:30	intro.ps
drwx-----	5 pbg	staff	512	Jul 8 09:33	private/
drwxrwxr-x	2 pbg	staff	512	Jul 8 09:35	doc/
drwxrwx---	2 pbg	student	512	Aug 3 14:13	student-proj/
-rw-r--r--	1 pbg	staff	9423	Feb 24 2003	program.c
-rwxr-xr-x	1 pbg	staff	20471	Feb 24 2003	program
drwx--x--x	4 pbg	faculty	512	Jul 31 10:31	lib/
drwx-----	3 pbg	staff	1024	Aug 29 06:52	mail/
drwxrwxrwx	3 pbg	staff	512	Jul 8 09:35	test/