

# 第51 - 52讲 文件目录管理、 文件共享和记录组块方式



## §5.3 File Directories



电子科技大学  
University of Electronic Science and Technology of China

# File Directories

- ❖ **Directory itself is a file owned by the operating system.**
- ❖ **Provides mapping between file names and the files themselves.**



# File Directories

**Contains information about files.**

## **1. Basic Information**

- **File Name, File Type, File Organization**

## **2. Addr. Information**

- **Volume, Starting Addr., Size Used, Size Alloc.**

## **3. Access Contr. Information**

- **Owner, Access Info., Permitted Actions**

## **4. Usage information**

- **Date Created, ID. of Creator, Date Last Read Access, ID of Last Reader, Date of Last Modified, ID of Last Modifier, Date of Last Backup, Current Usage**



# Structure for a Directory

- ❖ 可见，不同系统保存文件目录的方式不同。有的系统将其部分信息保存在文件头部，只将一些必要信息如文件名、文件大小、外存中的存储位置等保存在文件目录中
- ❖ 对目录文件的典型操作：
  - Search：按文件名检索文件目录
  - Create file：建立文件时，必须为之建立一个新目录
  - Delete file：删除文件时，必须同时删除其文件目录
  - List Directory：用白登需要列文件目录及文件的



# Simple Structure for a Directory

- ❖ List of entries, one for each file.
- ❖ Sequential file with the name of the file serving as the key.
- ❖ Provides no help in organizing the files.
- ❖ Forces user to be careful not to use the same name for two different files.



# Two-level Scheme for a Directory

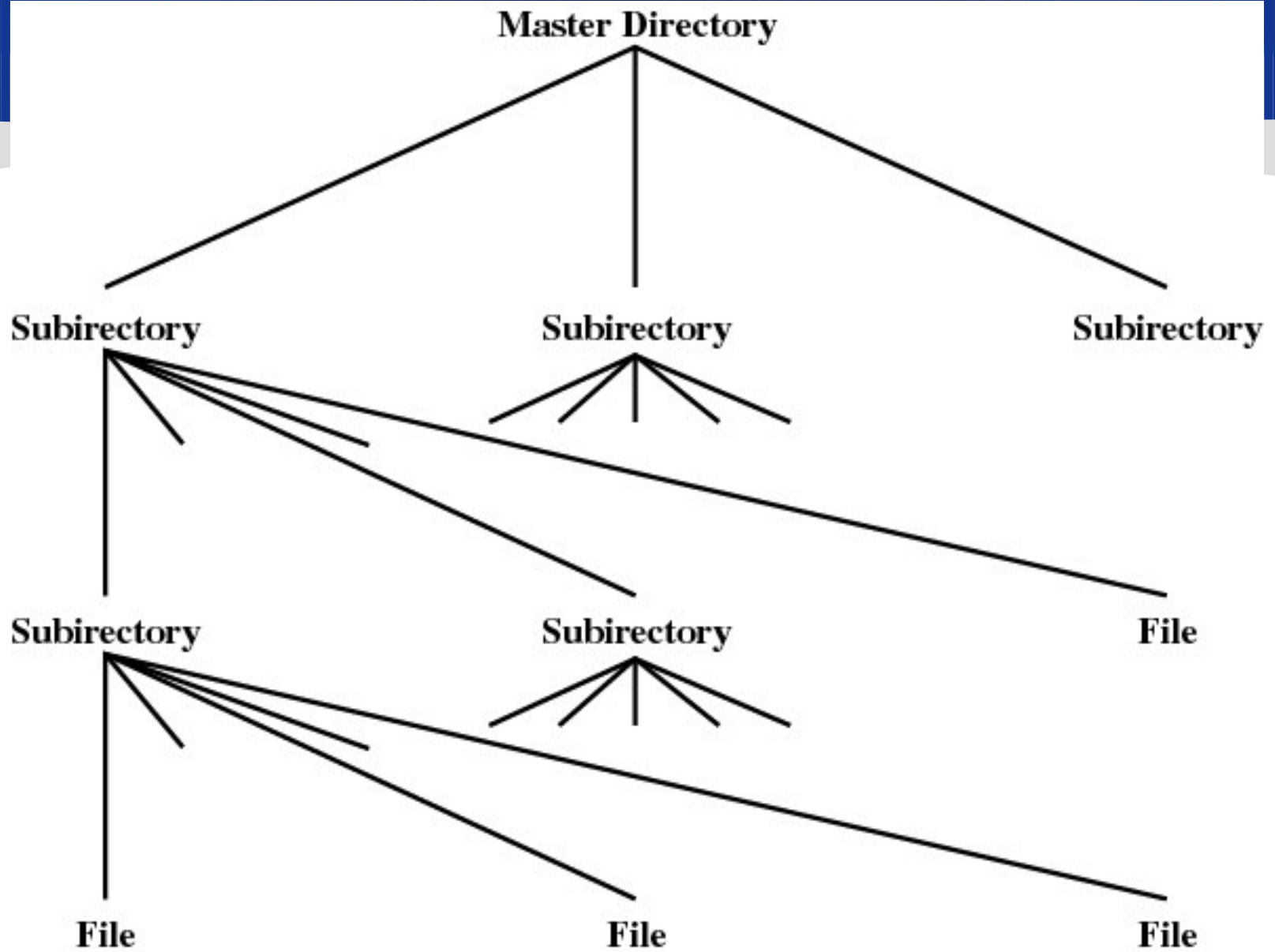
- ❖ One directory for each user and a master directory.( 用户目录和主目录 )
- ❖ Master directory contains entry for each user.
  - Provides address and access control information.
- ❖ Each user directory is a simple list of files for that user.
- ❖ Still provides no help in structuring collections of files.



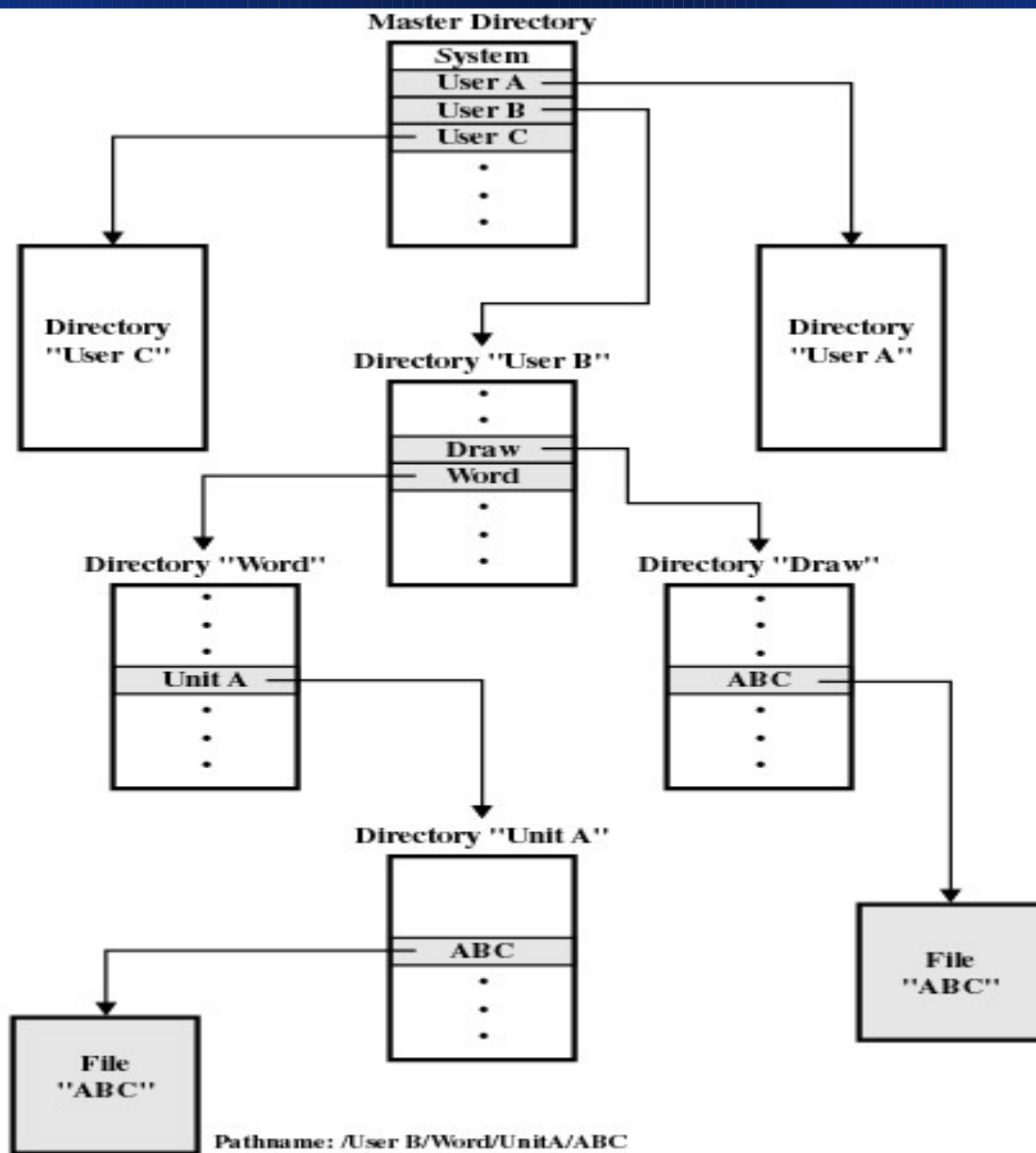
# Hierarchical, or Tree-Structured Directory

- ❖ Master directory with user directories underneath it. (主目录下建立若干用户目录)
- ❖ Each user directory may have subdirectories and files as entries. (每个用户目录下允许再创建子目录及文件目录)





**Figure 12.4** Tree-Structured Directory



**Figure 12.5 Example of Tree-Structured Directory**

# Hierarchical, or Tree-Structured Directory

- ❖ Files can be located by following a path from the root, or master, directory down various branches.
  - This is the pathname for the file.
- ❖ Can have several files with the same file name as long as they have unique path names.



# Hierarchical, or Tree-Structured Directory

- ❖ **Current directory is the working directory.**  
(当前目录即工作目录)
- ❖ **Files are referenced relative to the working directory.** (相对工作目录的文件路径为相对路径)



## §5.4 File Sharing



电子科技大学  
University of Electronic Science and Technology of China

# File Sharing

❖ In multiuser system, allow files to be shared among users.

❖ Two issues

- Access rights
- Management of simultaneous access. (并行存取)



# Access Rights

## ❖ None

- User may not know of the existence of the file.
- User is not allowed to read the user directory that includes the file.

## ❖ Knowledge (探知)

- User can only determine that the file exists and who its owner is.

# Access Rights

## ❖ Execution

- The user can load and execute a program but cannot copy it.

## ❖ Reading

- The user can read the file for any purpose, including copying and execution.

## ❖ Appending

- The user can add data to the file but cannot modify or delete any of the file's contents.





# Access Rights

## ❖ Updating

- The user can modify, delete, and add to the file's data. This includes creating the file, rewriting it, and removing all or part of the data.

## ❖ Changing protection

- User can change access rights granted to other users.

## ❖ Deletion

- User can delete the file.



# Access Rights

## ❖ Owners

- Has all rights previously listed.
- May grant rights to others using the following classes of users.
  - Specific user
  - User groups
  - All for public files



# Simultaneous Access

- ❖ User may lock entire file when it is to be updated.
- ❖ User may lock the individual records during the update.
- ❖ Mutual exclusion and deadlock are issues for shared access.



## §5.5 Record Blocking



电子科技大学  
University of Electronic Science and Technology of China

# Record Blocking

❖ 记录是存取文件的逻辑单位，而数据块是 I/O 的基本单位，记录必须组织成数据块以便于 I/O

❖ 问题：

- 数据块的长度固定还是可变？
- 根据记录的平均长度如何设置数据块的长度？



# 数据块的长度固定还是可变？

大多数系统采用固定长度的数据块，以简化 I/O 操作、Buffer 的分配及辅存中数据块的组织管理



# 如何设置数据块的长度？

- ❖ 数据块越大，一次 I/O 传输的记录就越多
- ❖ 大数据块适合文件顺序访问，因为这样可以减少 I/O 次数，加快处理速度
- ❖ 当随机访问文件，或访问局部性很差时，传输的部分记录不会使用，效率低。
- ❖ 另外，大数据块需要更大的 I/O buffer，增加其管理复杂度



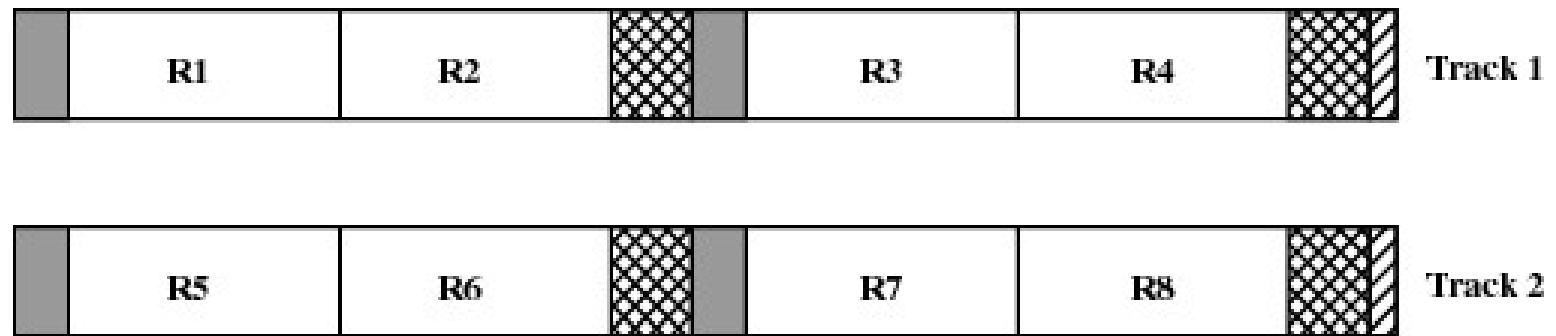
# Fixed blocking （固定组块法）

- ❖ 数据块由固定长度的若干条记录组成
- ❖ 块内可能会存在一些被浪费空间





# Fixed Blocking



Fixed Blocking

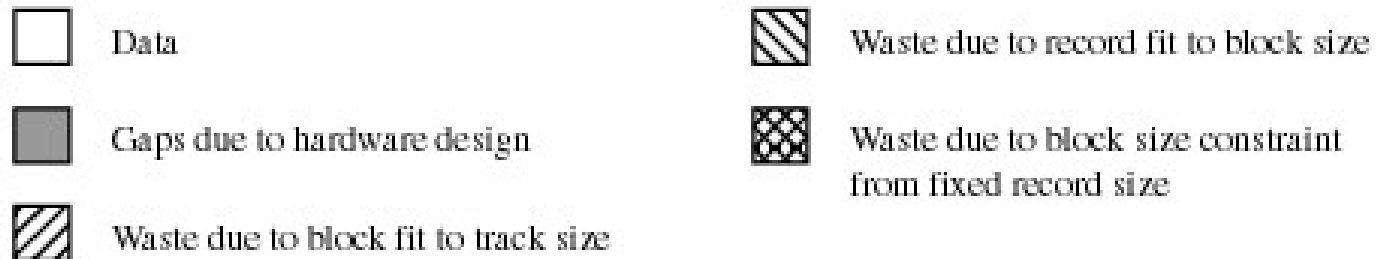
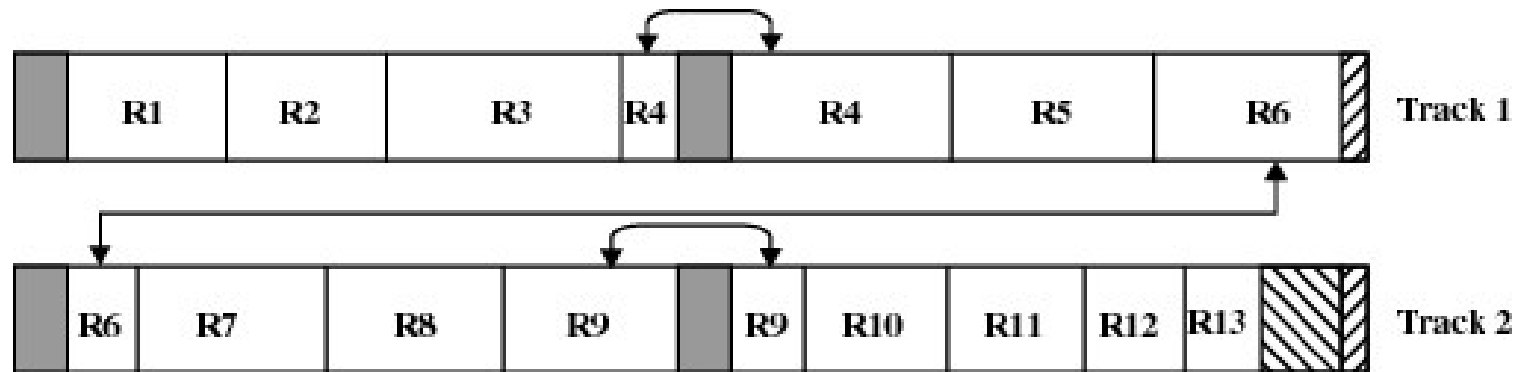


Figure 12.6 Record Blocking Methods [WIED87]

# Variable Blocking: Spanned (可变长跨块组块法)

- ❖ 数据块由变长记录组成，一条记录可以跨越两个数据块，可用指针记载一条跨块存储记录所在的下一数据块
- ❖ 块内无浪费空间

# Variable Blocking: Spanned



Variable Blocking: Spanned

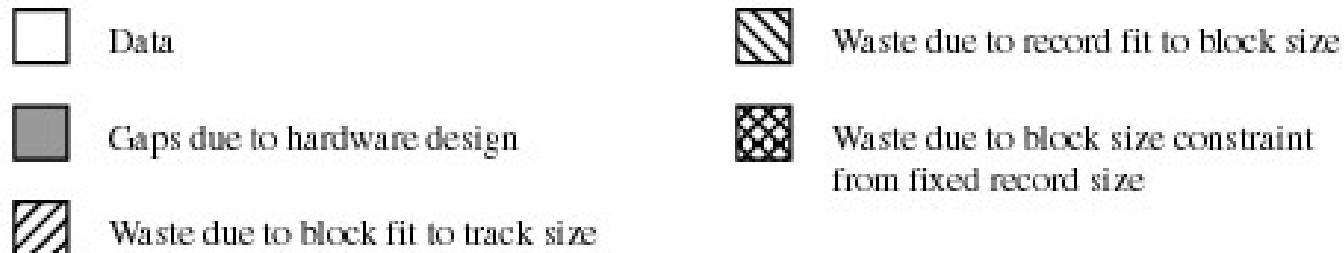


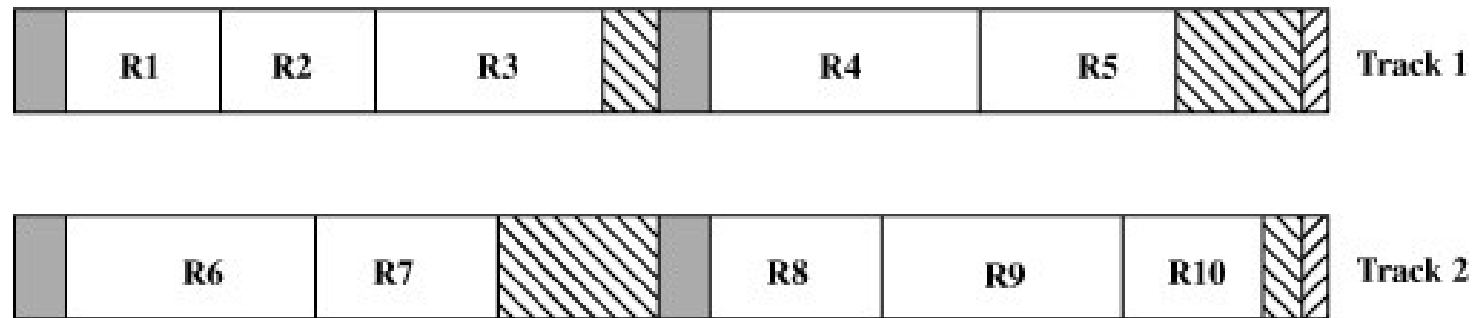
Figure 12.6 Record Blocking Methods [WIED87]

# Variable Blocking Unspanned ( 可变长非跨块组块法 )

- ❖ 数据块由变长记录组成
- ❖ 不允许一条记录跨越两个数据块
- ❖ 块内可能存在被浪费空间



# Variable Blocking Unspanned



Variable Blocking: Unspanned

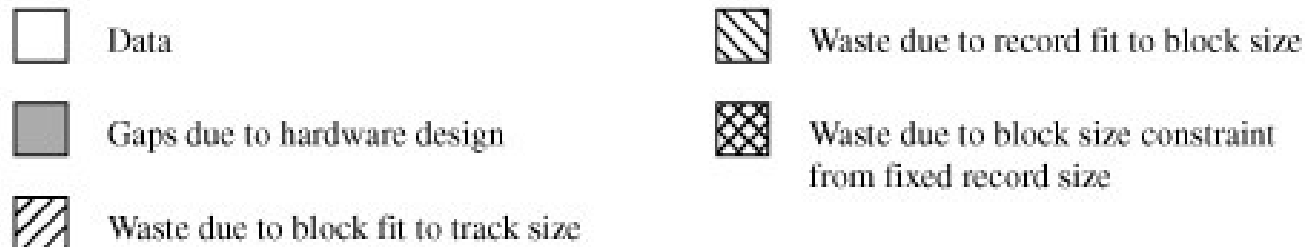


Figure 12.6 Record Blocking Methods [WIED87]