File Operations

J P Misra

Topics

- File Concept
- Access Methods
- Directory Structure
- File System Mounting
- File Sharing
- Protection

File Concept

- Collection of related information and it is logical storage unit.
- OS maps files on physical device
- Types:
  - Data
    - numeric
    - character
    - binary
  - Program
    - Source
    - Object
    - Executables

File Structure

- None - sequence of words, bytes
- Simple record structure
  - Lines
  - Fixed length
  - Variable length
- Complex Structures
  - Formatted document
  - Relocatable load file
- Can simulate last two with first method by inserting appropriate control characters.

File Attributes

- **Name** – only information kept in human-readable form
- Identifier – unique tag usually number identifies files with in the file system (in non human readable form).
- **Type** – needed for systems that support different types.
- **Location** – pointer to file location on device.
- **Size** – current file size.
- **Protection** – controls who can do reading, writing, executing.
- **Time**, **date**, **and user identification** – data for protection, security, and usage monitoring.
- Information about files are kept in the directory structure, which is maintained on the disk.

File Operations

- Create
  - Space in the file system must be found
  - An entry for the new file must be made in the directory.
  - Directory entry consist of file name, location & other information about the file.
- Write
  - Make a system call specifying both name of the file and the information to be written.
  - Search for the location of the file
  - System must keep a write pointer to keep the location of the file where the next write is to take place.

File Operations

- Read
  - Make a system call specifying the name of the file and where the next block of the file should be put.
  - Maintain a read pointer to where the next read is to take place.
- Reposition within file – file seek
  - Current file position is set to a given value.
  - Reposition with in a file does not need to invoke any actual I/O
- Delete
  - Search the directory for the file name
  - Release all the file space and erase the directory entry.

File Operations

- Truncate
  - Erase the content of a file but keep its attributes.
  - Only the file length attribute will change.
- Other common operations
  - Append and rename
- Open(F_i) – search the directory structure on disk for entry F_i, and move the content of entry to memory.
- Close (F_i) – move the content of entry F_i in memory to directory structure on disk.

Implementation in Unix

- Uses 2 levels of internal tables
  - A per process table
    - Tracks all open files
    - Stores the information regarding the use of the file by the process (file pointer indicating the location in the file that the next read or write will affect)
  - System wide table
    - Contains process independent information (location of the file, access dates and file size)
    - It has an open count associated with each file, indicating the number of processes that have the file open.

Information associated with open file table

- File pointer
  - This pointer is unique to each process operating on file.
  - So it must be kept separate from the on disk file attributes
- File open count
  - If file open count is 0 then only close the file
- Disk location of the file
- Access rights

File Types – Name, Extension

file type	usual extension	function
executable	exe, com, bin or none	read 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, rrf, doc	various word-processor formats
library	lib, a, so, dll, mpeg, mov, rm	libraries of routines for programmers
print or view	arc, zip, tar	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	binary file containing audio or A/V information

Access Methods

- **Sequential Access**

  read next *:* reads the next portion and automatically advances a file pointer

  write next *:* appends to the end of fileand file pointer is advanced to new end of file (eof)

  reset*:* file pointer is reset to beginning

  Direct Access: file is considered as numbered block or record

  read n

  write n

  position to n

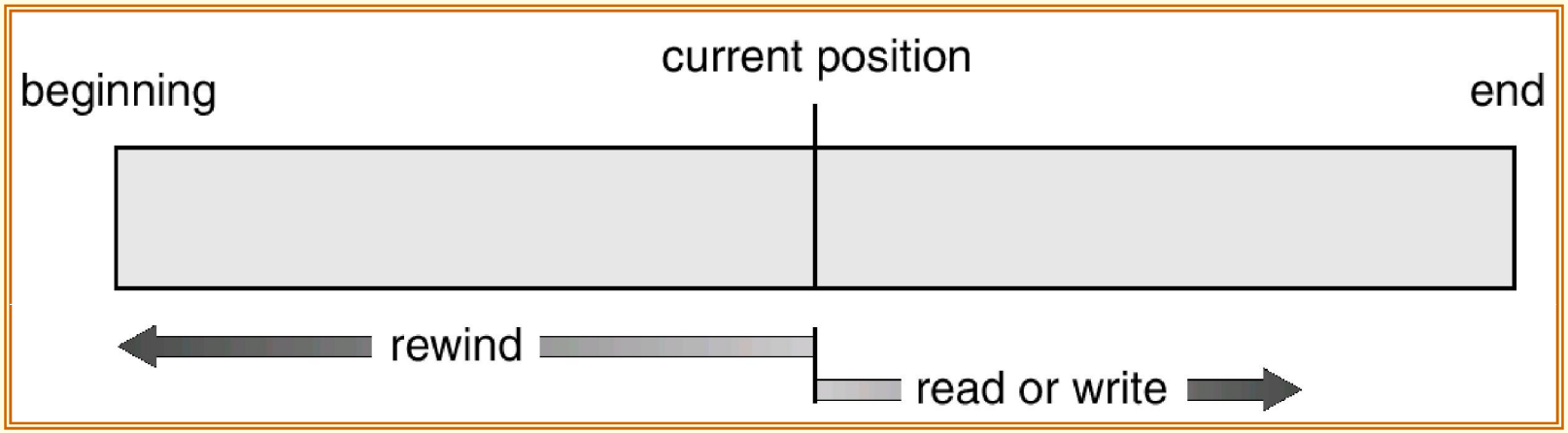
  read next

  write next

  rewrite n

  n = relative block number

Sequential-access File



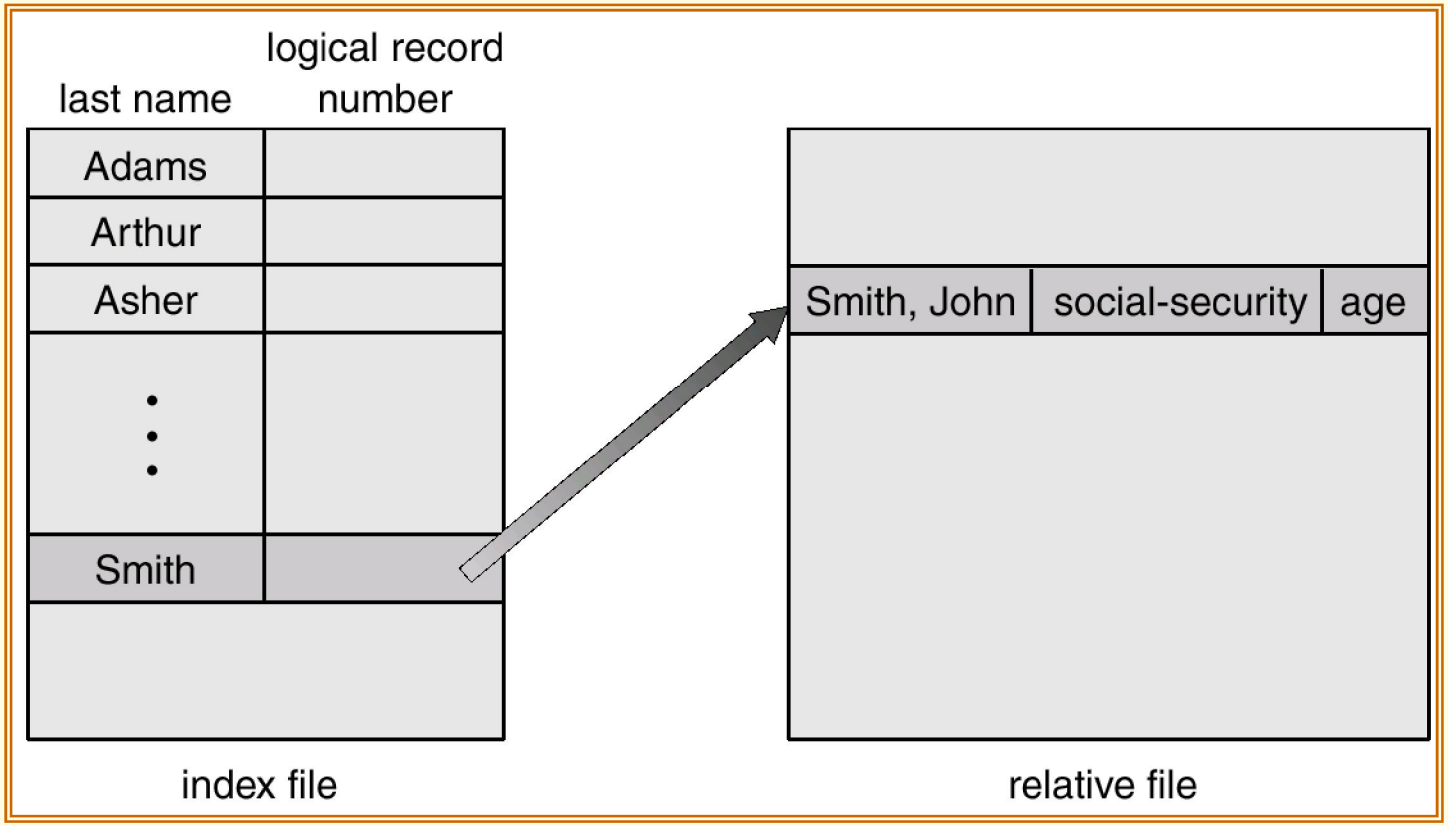
Simulation of Sequential Access on a Direct-access File

sequential access	implementation for direct access
reset	*cp* = 0;
read next	*read cp*; *cp* = *cp*+1;
write next	*write cp*; *cp* = *cp*+1;

Example of Index and Relative Files

- Consider a file containing 10digit product code and 6 digit indicating cost of product requiring a total of 16 byte for one product
- Let disk store 1024 byte /block. 64 records can be stored per block
- If we need 120000 records it will require 2000block
- By keeping the file sorted on product code, we can define index consisting of first product code of each block
- The index would have 2000 entries each of 10 digit size.
- The index can be easily searched to find which block is containing desired record

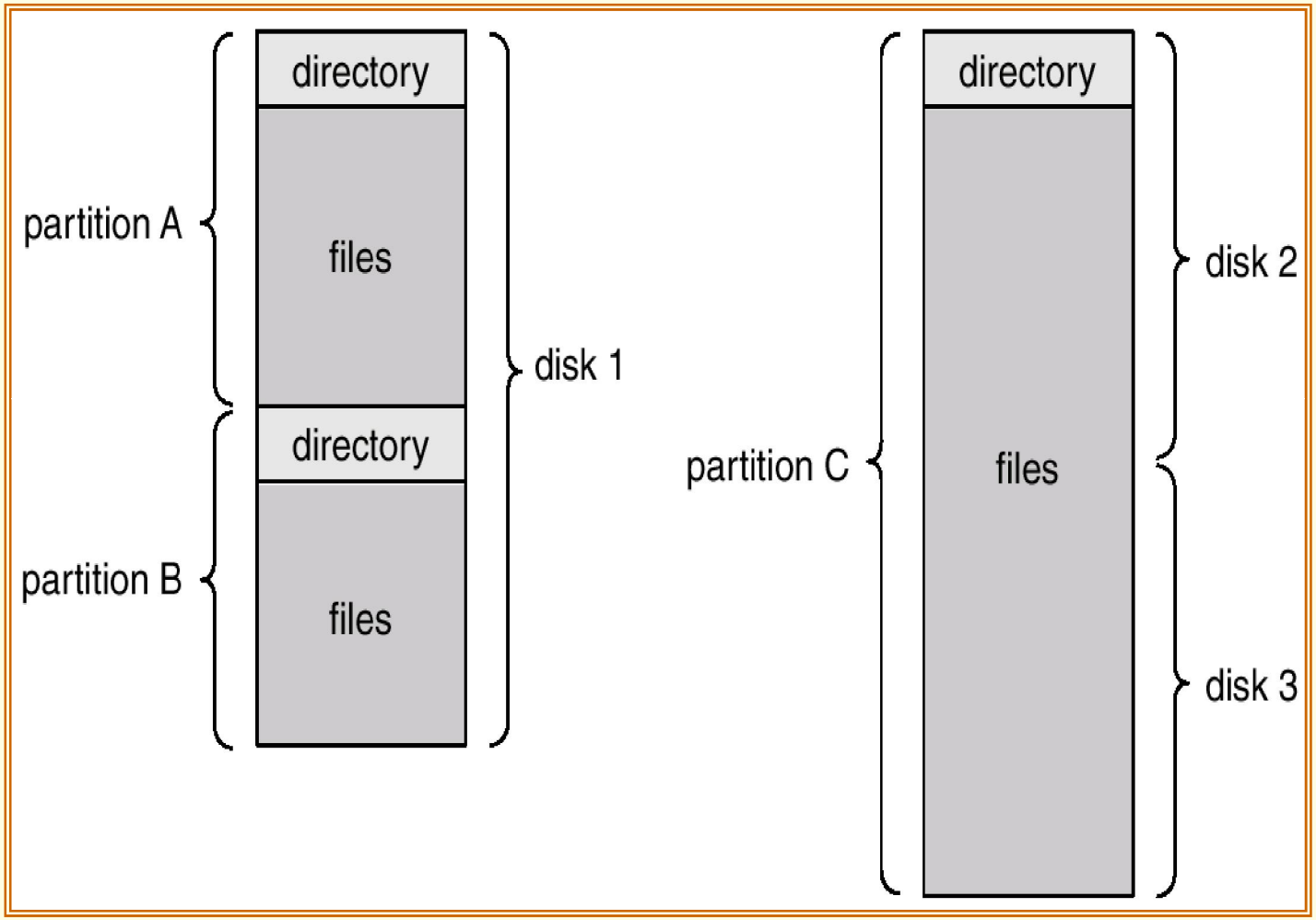
Example of Index and Relative Files



Directory & disk structure

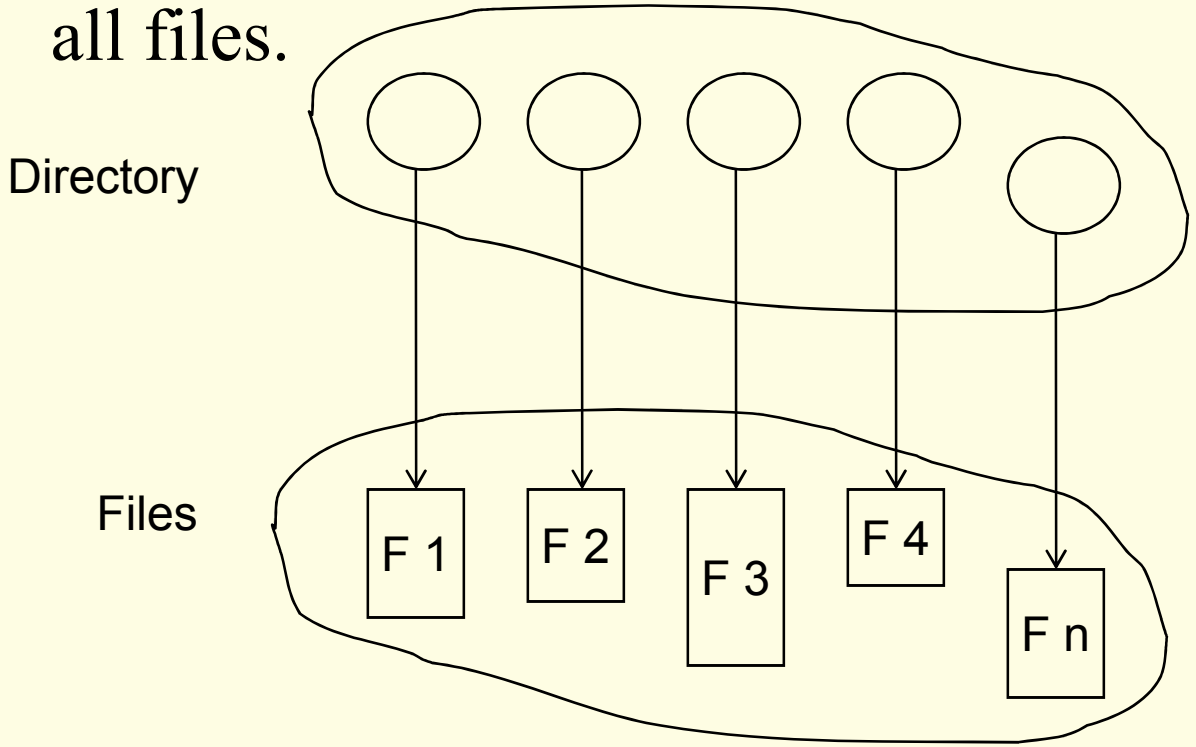
- A computer can have thousands of files
- Files are stored on random accees storage device including HDD, CD, SSD etc
- The entire device can be used for a single file system or it can be sub deivided or multiple devices can be viewed as single device and can be used
- A devise can be partitioned
- Any entity containing a file system is known as volume.
- Volume may be subset of device, whole device or multiple device

A Typical File-system Organization



Directory Structure

- A collection of nodes containing information about all files.



Both the directory structure and the files reside on disk.
Backups of these two structures are kept on tapes.

Information in a Device Directory

- Name
- Type
- Address
- Current length
- Maximum length
- Date last accessed (for archival)
- Date last updated (for dump)
- Owner ID
- Protection information

Operations Performed on Directory

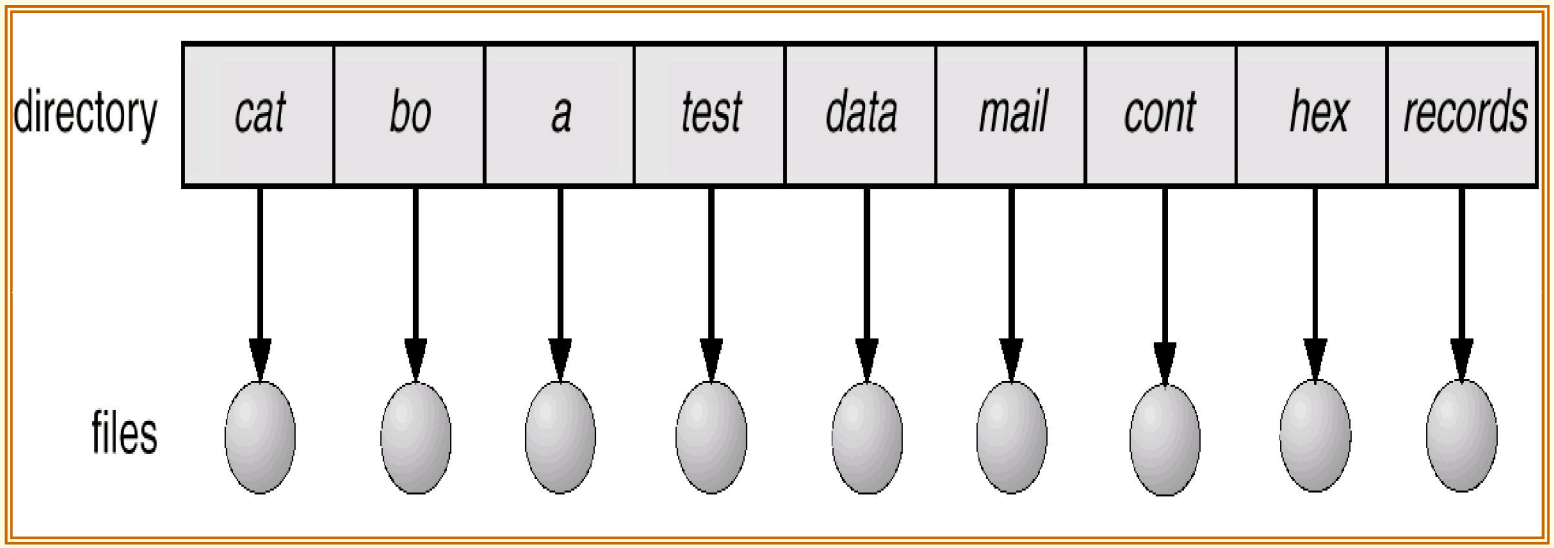
- Search for a file
  - Search a file in the directory
- Create a file
- Delete a file
- List a directory
  - List the files in the directory
- Rename a file
  - Rename may also allow its position with in the directory structure to be changed
- Traverse the file system

Organize the Directory (Logically) to Obtain

- **Efficiency** – locating a file quickly.
- **Naming** – convenient to users.
  - Two users can have same name for different files.
  - The same file can have several different names.
- **Grouping** – logical grouping of files by properties, (e.g., all Java programs, all games, …)

Single-Level Directory

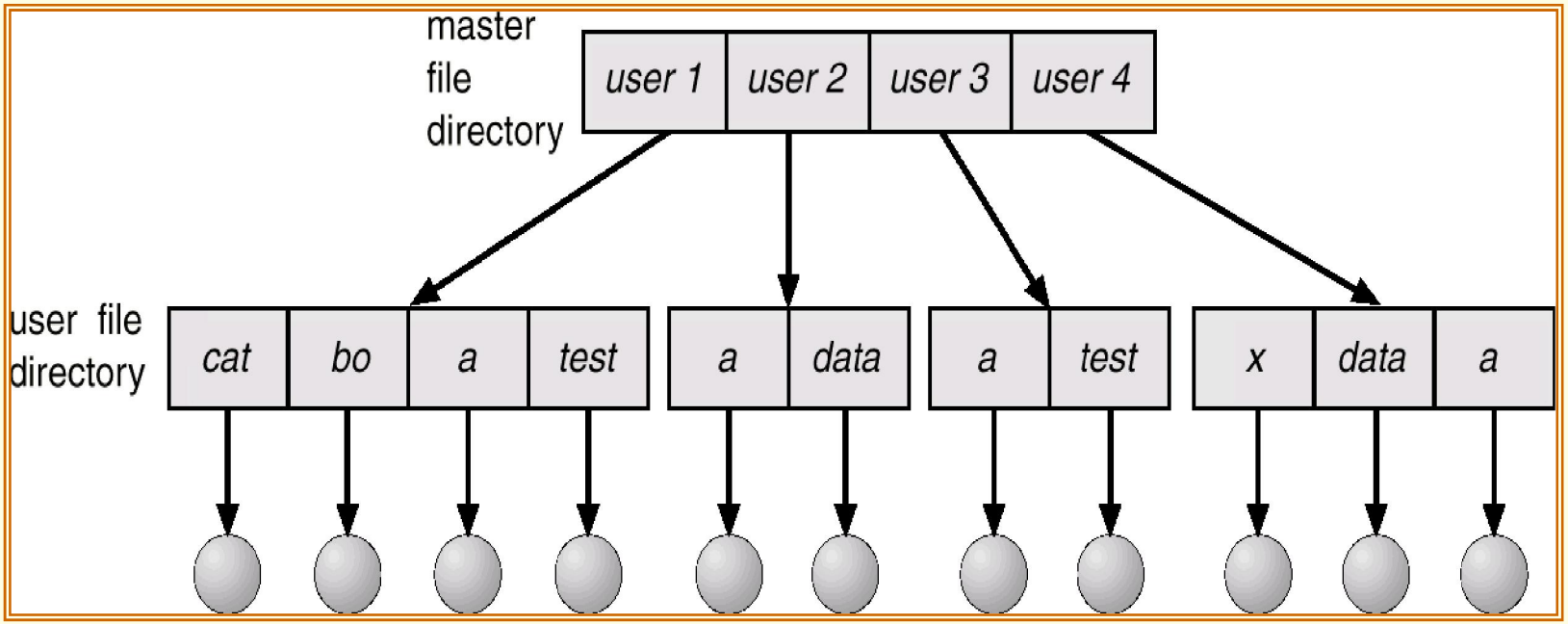
- A single directory for all users.



Naming problem

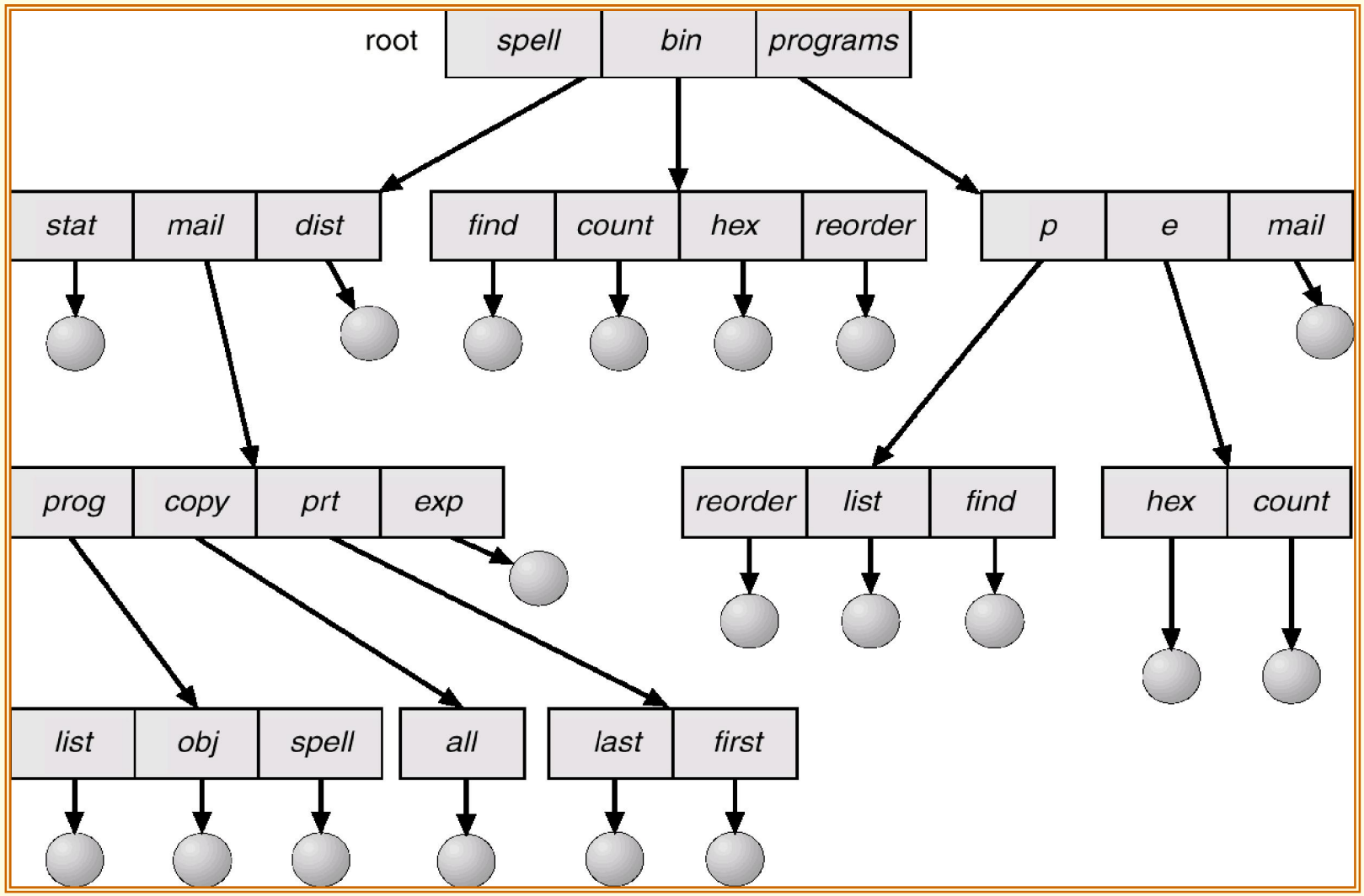
Grouping problem

Two-Level Directory



- Path name
- Can have the same file name for different user
- Efficient searching
- No grouping capability

Tree-Structured Directories



Tree-Structured Directories (Cont.)

- Efficient searching
- Grouping Capability
- Current directory (working directory)
  - **cd** /spell/mail/prog
  - **type** list

Tree-Structured Directories (Cont.)

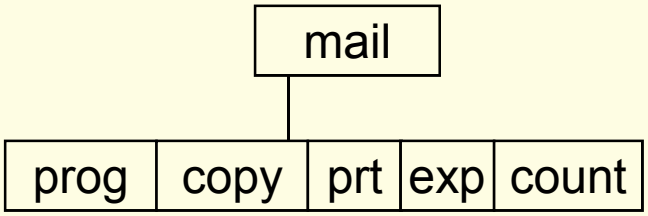
- **Absolute** or **relative** path name
- Creating a new file is done in current directory.
- Delete a file

  rm <file-name>
- Creating a new subdirectory is done in current directory.

  mkdir <dir-name>

  Example: if in current directory **/mail**

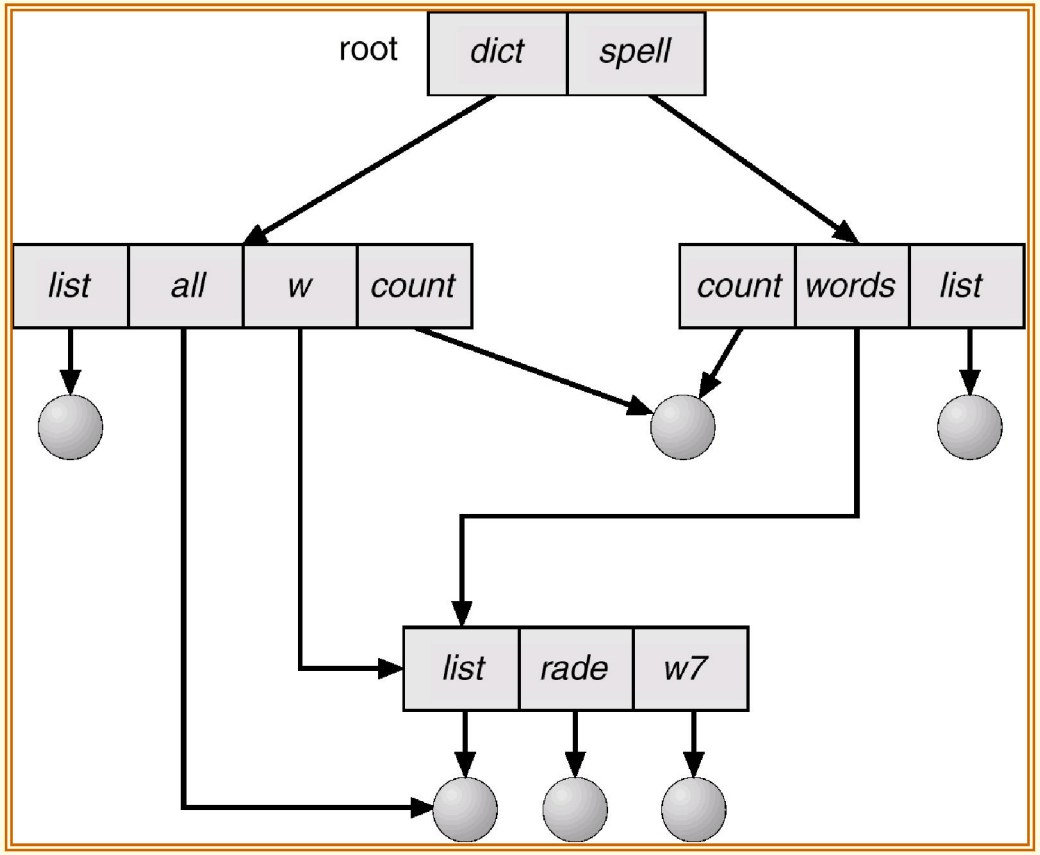
  mkdir count



Deleting “mail” $\Rightarrow$ deleting the entire subtree rooted by “mail”.

Acyclic-Graph Directories

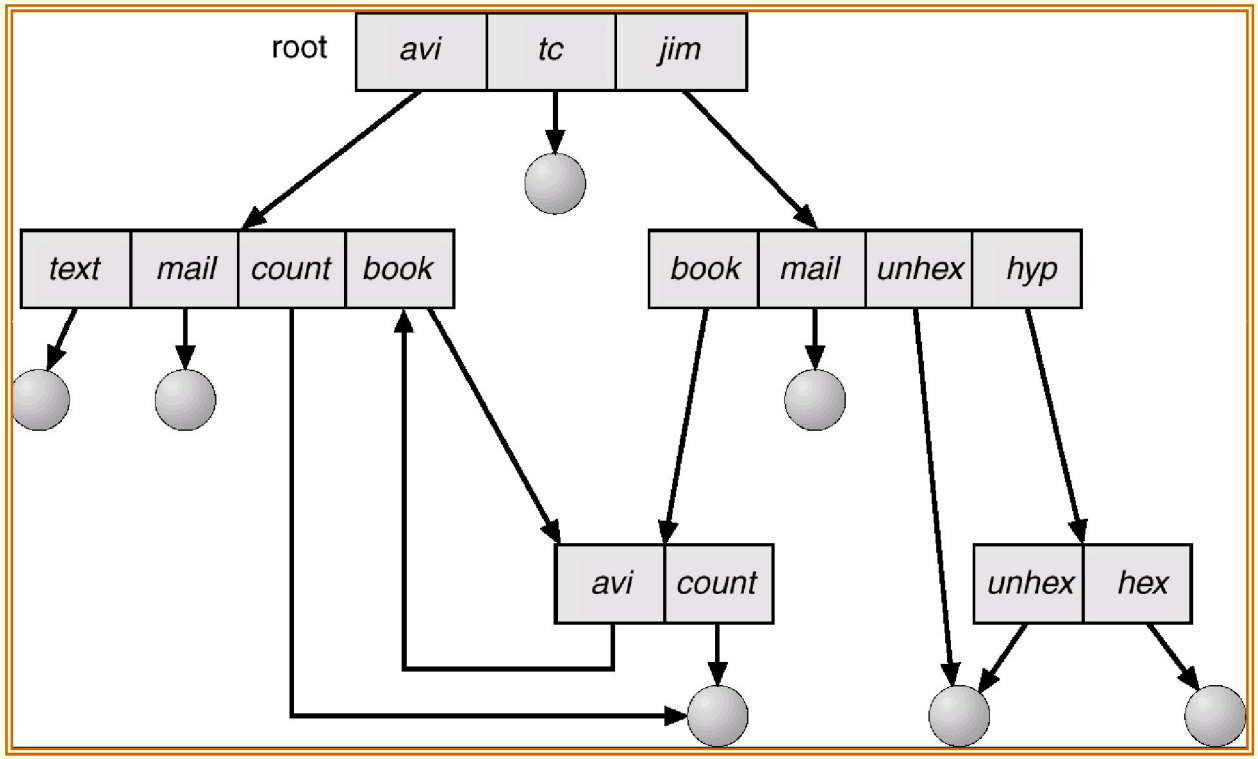
- Have shared subdirectories and files.



Acyclic-Graph Directories (Cont.)

- Two different names (aliasing)
- Can implement shared files and subdirectories by using link.
- OS ignores this link when traversing directory trees to preserve the a cyclic structure of the system.
- Can duplicate all information in both shared directory
  - Inconsistency if the file is modified
  - Careful during deletion (dangling pointers)

General Graph Directory



General Graph Directory (Cont.)

- How do we guarantee no cycles?
  - Allow only links to file not subdirectories.
  - Garbage collection.
  - Every time a new link is added use a cycle detection algorithm to determine whether it is OK.

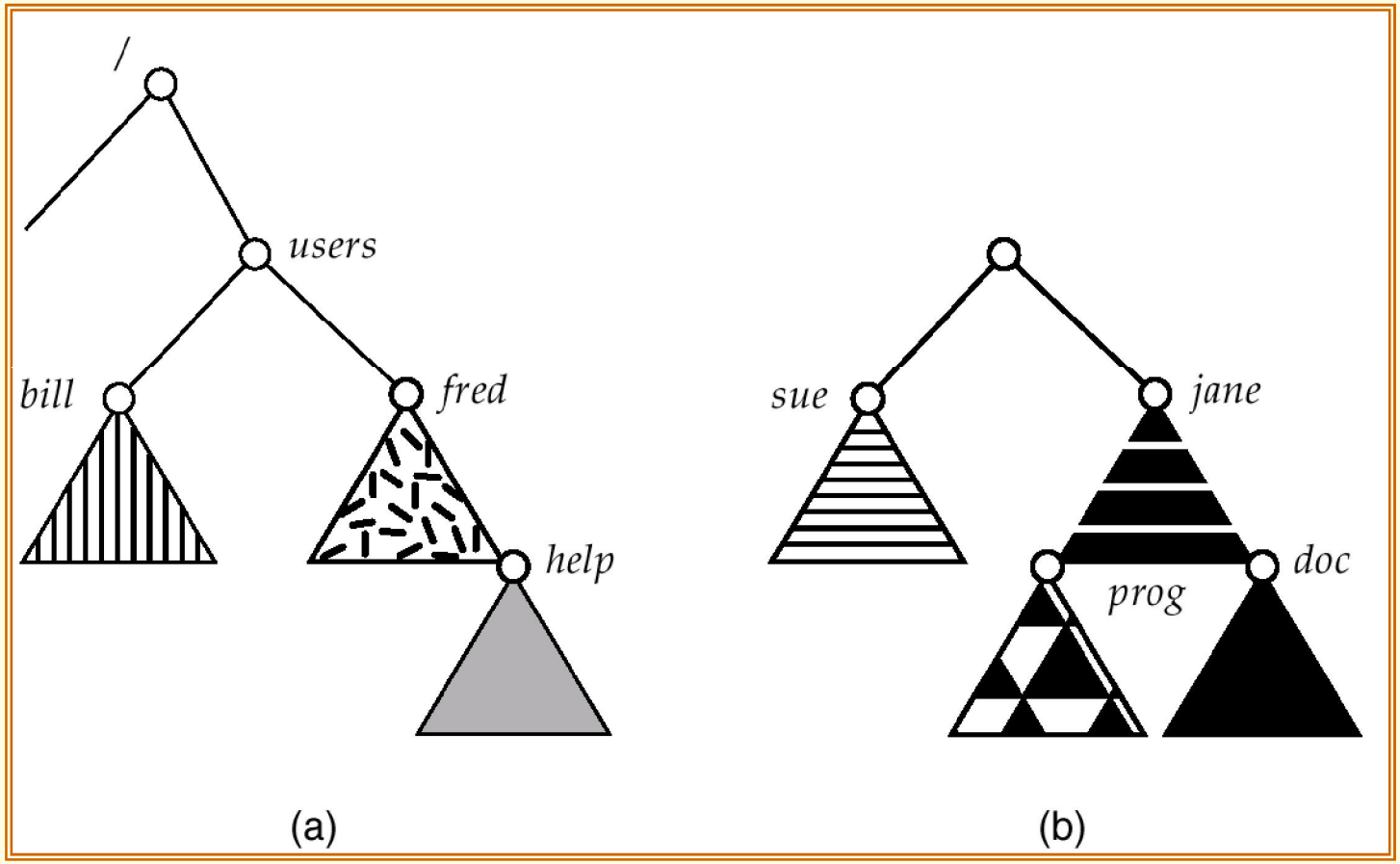
File System Mounting

- A file system must be **mounted** before it can be accessed.
- File is mounted at **mount point**
  - Typically a mount point is an empty directory at which the mounted system will be attached.

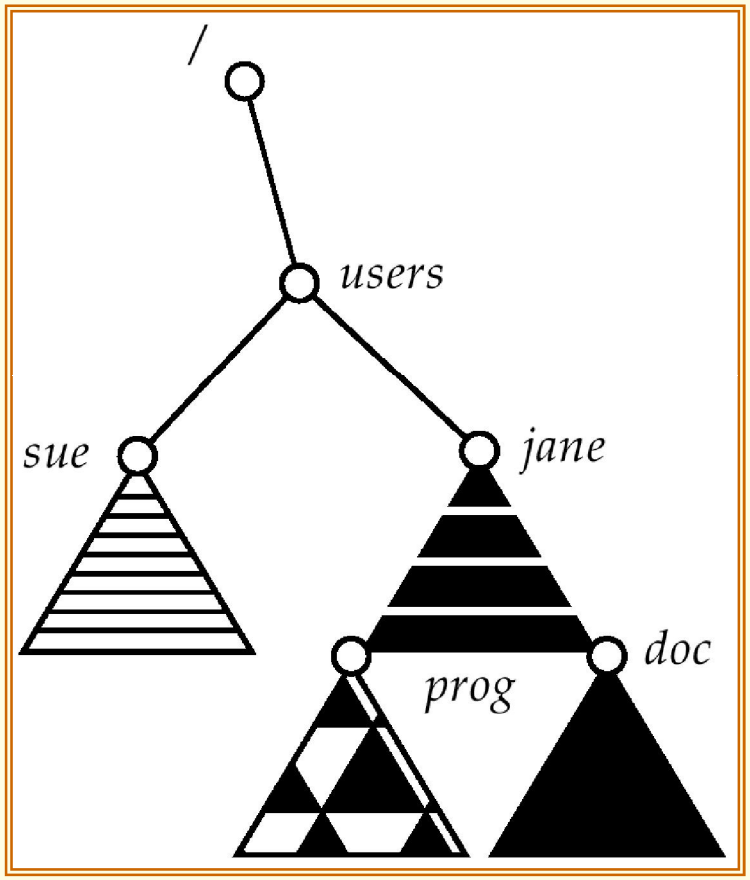
File System Mounting

- OS verifies that the device contain a valid file system (it does so by asking the device driver to read the device directory and verify that the directory has the expected format)
- OS notes in its directory structure that a file system is mounted at the specified mount point.
- Example shows how file mounting work in Linux

(a) Existing. (b) Unmounted Partition



Mount Point



File Sharing

- Sharing of files on multi-user systems is desirable.
  - Most OS implement sharing using owner and group
  - System implement user attributes by managing a list of user names and associated user identifiers
- Sharing may be done through a *protection* scheme.
- On distributed systems, files may be shared across a network.
- Network File System (NFS) is a common distributed file-sharing method.

Protection

- File owner/creator should be able to control:
  - what can be done
  - by whom

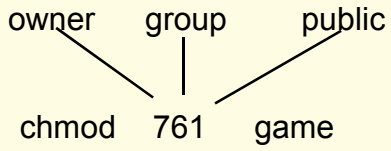
- Types of access
  - Read read from the file
  - Write write / rewrite the file
  - Execute Load the file into memory and execute
  - Append write new information at the end of the file
  - Delete delete the file and free the memory for reuse
  - List List the name and attributes of the file
  - Rename, copy, editing

Access Lists and Groups

- Mode of access: read, write, execute
- Three classes of users

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

- Ask manager to create a group (unique name), say G, and add some users to the group.
- For a particular file (say *game*) or subdirectory, define an appropriate access.



Thanks