

Lecture 14

File Systems

INT107 Computing Platform Technology - 2016

Source:

- *Operating System Concepts with Java*, 7th edition, A. Silberschatz et.al., 2007

Outline

- File
 - Types
 - Structure
 - Attributes
 - Operations
 - Access Methods
- Directory
 - Operations
 - Organization
 - Mounting
- Link
- Protection

File Concept

- Contiguous logical address space
- Types:
 - Data
 - numeric
 - character
 - binary
 - Program

File Types – Name, Extension

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 com- pressed, for archiving or storage
multimedia	mpeg, mov, rm, mp3, avi	binary file containing audio or A/V information

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
- Who decides:
 - Operating system
 - Program

File Attributes

- **Name** – only information kept in human-readable form
- **Identifier** – unique tag (number) identifies file within file system
- **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

- File is an **abstract data type**
- **Create**
- **Write**
- **Read**
- **Reposition within file**
- **Delete**
- **Truncate**
- $Open(F_i)$ – 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

Open Files

- Several pieces of data are needed to manage open files:
 - File pointer: pointer to last read/write location, per process that has the file open
 - File-open count: counter of number of times a file is open – to allow removal of data from open-file table when last processes closes it
 - Disk location of the file: cache of data access information
 - Access rights: per-process access mode information

Access Methods

- **Sequential Access**

- read next

- write next

- reset

- no read after last write
(rewrite)

- **Direct Access**

- read n

- write n

- position to n

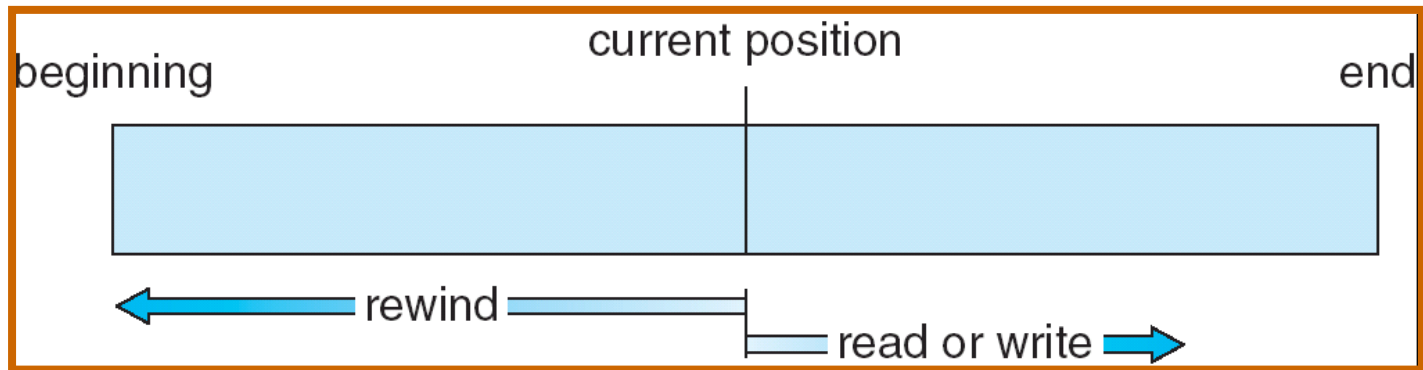
- read next

- write next

- rewrite n

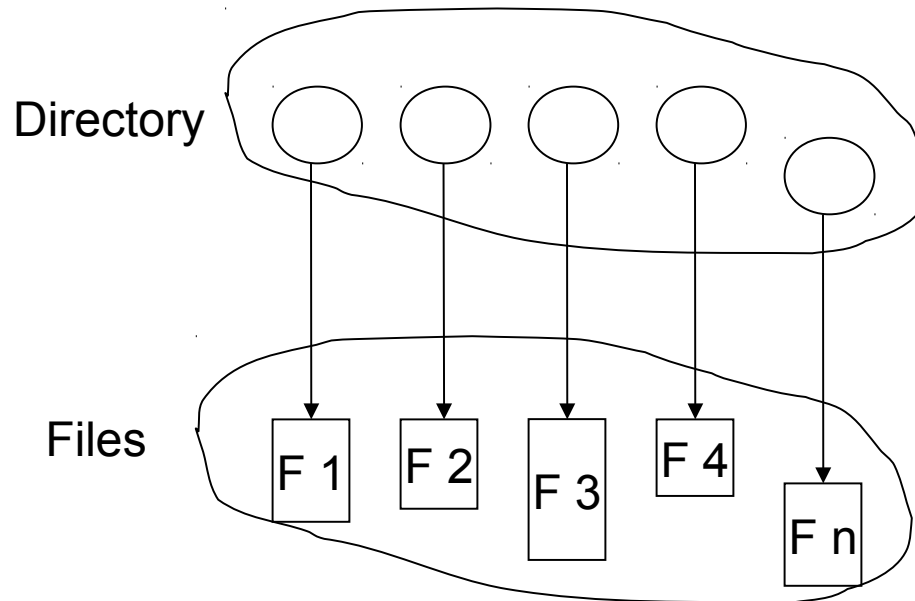
n = relative block number

Sequential-access File



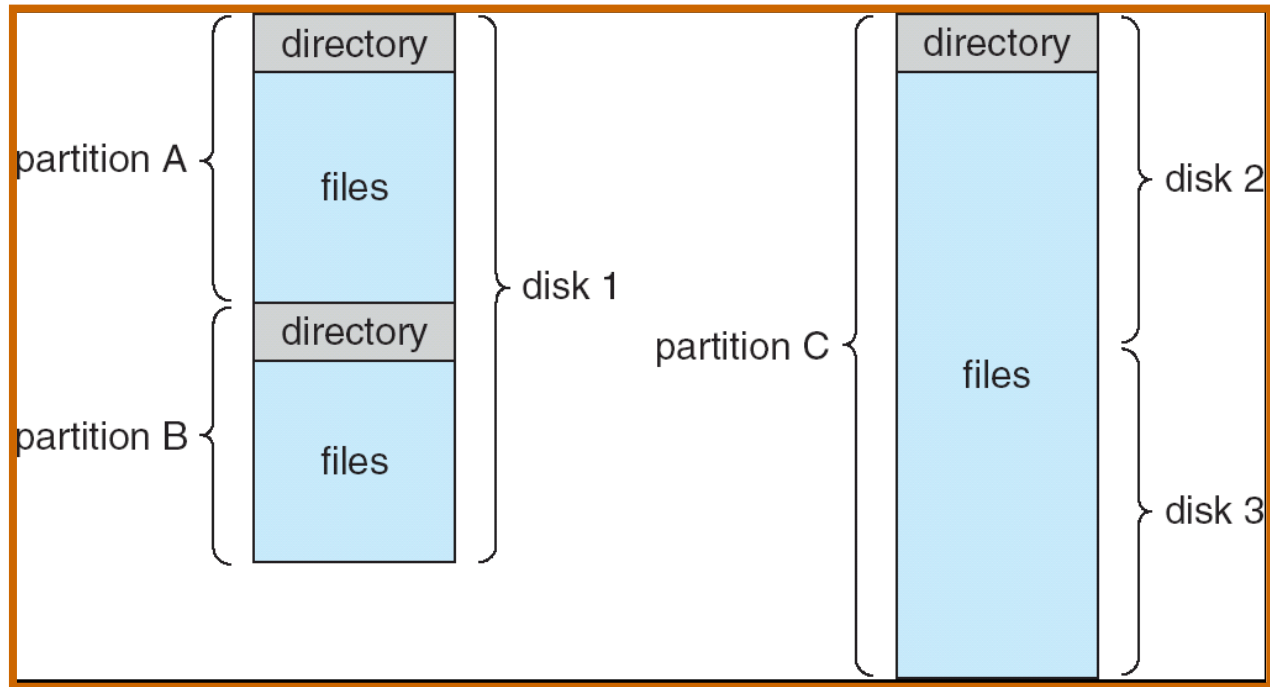
Directory Structure

- A collection of nodes containing information about all files



Both the directory structure and the files reside on disk

A Typical File-system Organization



Operations Performed on Directory

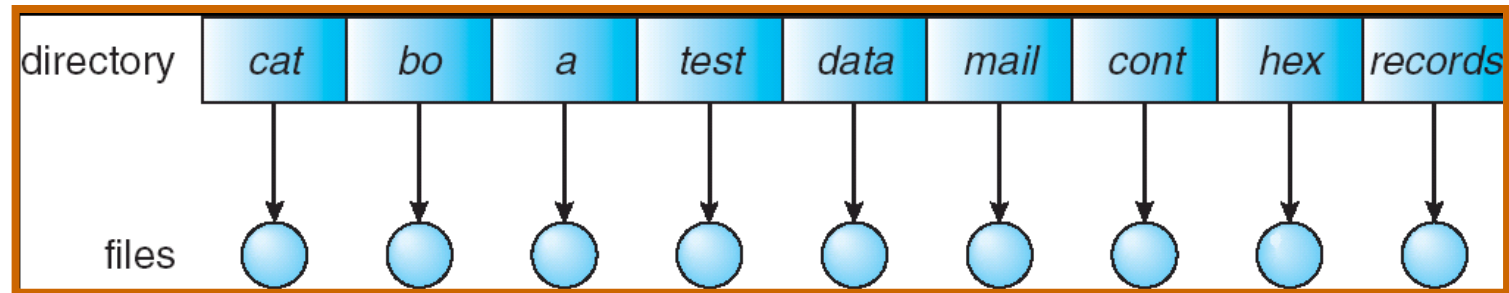
- Search for a file
- Create a file
- Delete a file
- List a directory
- Rename a file
- Traverse the file system

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

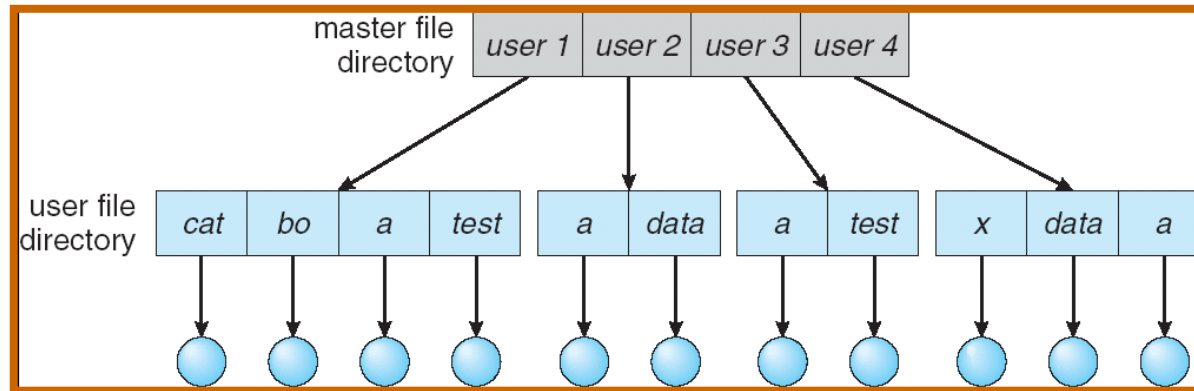


Naming problem

Grouping problem

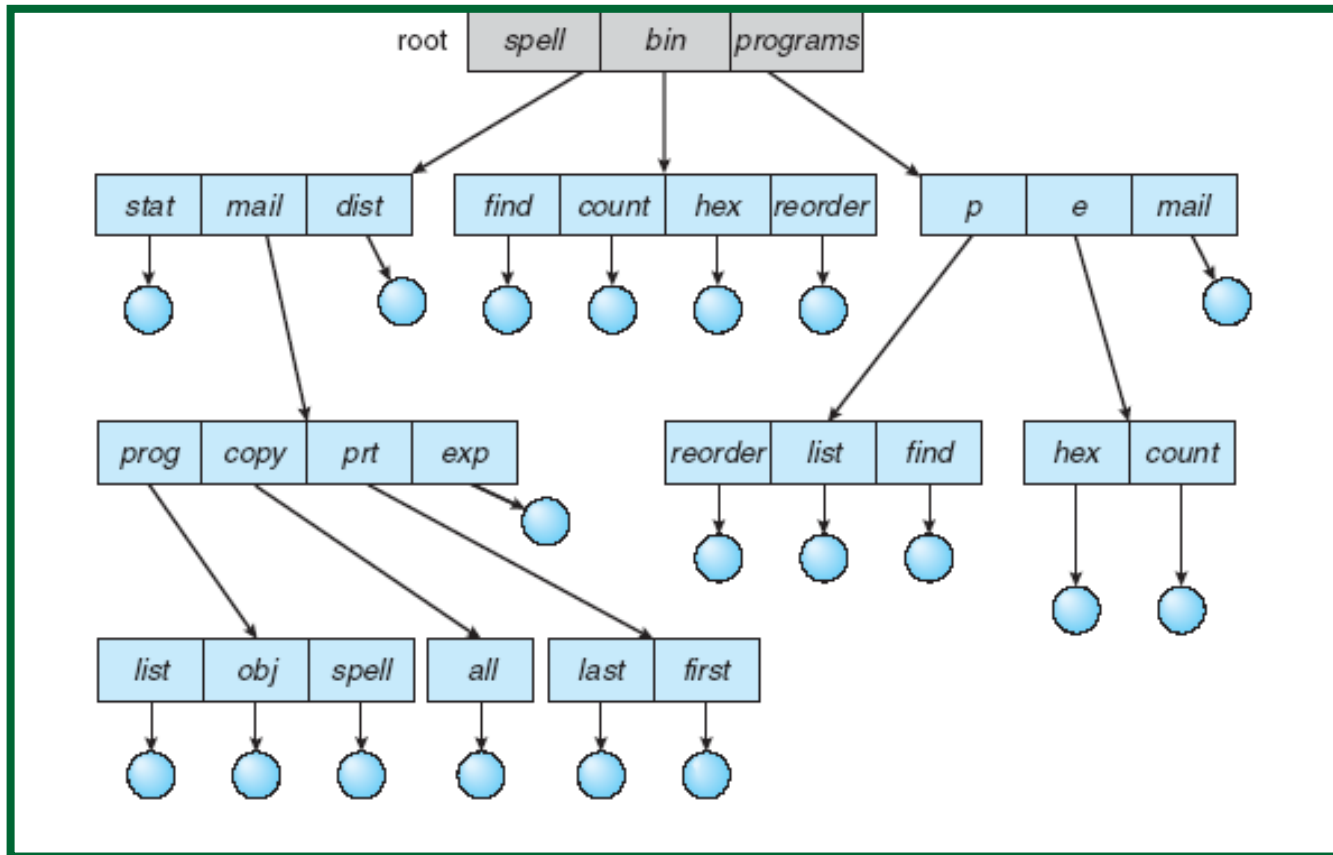
Two-Level Directory

- Separate directory for each user



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

Tree-Structured Directories



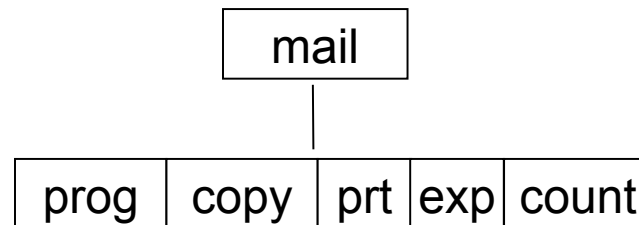
Tree-Structured Directories (Cont)

- 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`

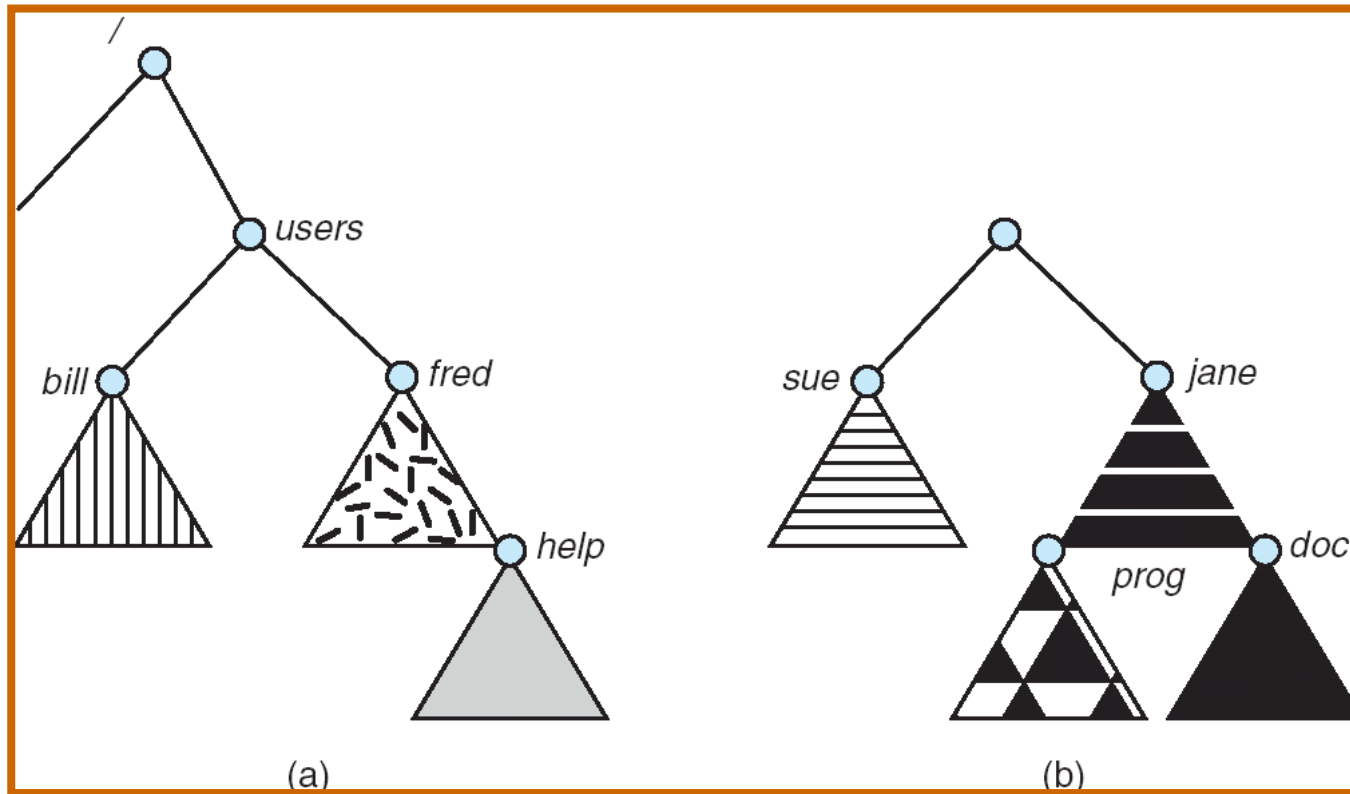


Deleting “mail” => deleting the entire subtree rooted by “mail”

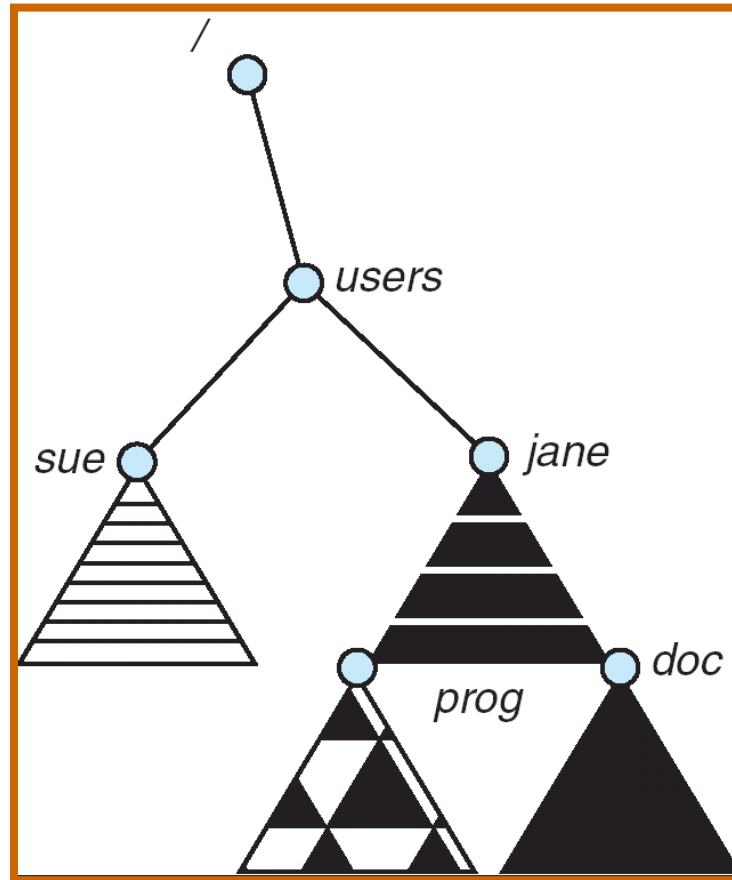
File System Mounting

- A file system must be **mounted** before it can be accessed
- A unmounted file system (i.e. Fig. 11-11(b)) is mounted at a **mount point**

(a) Existing. (b) Unmounted Partition



Mount Point

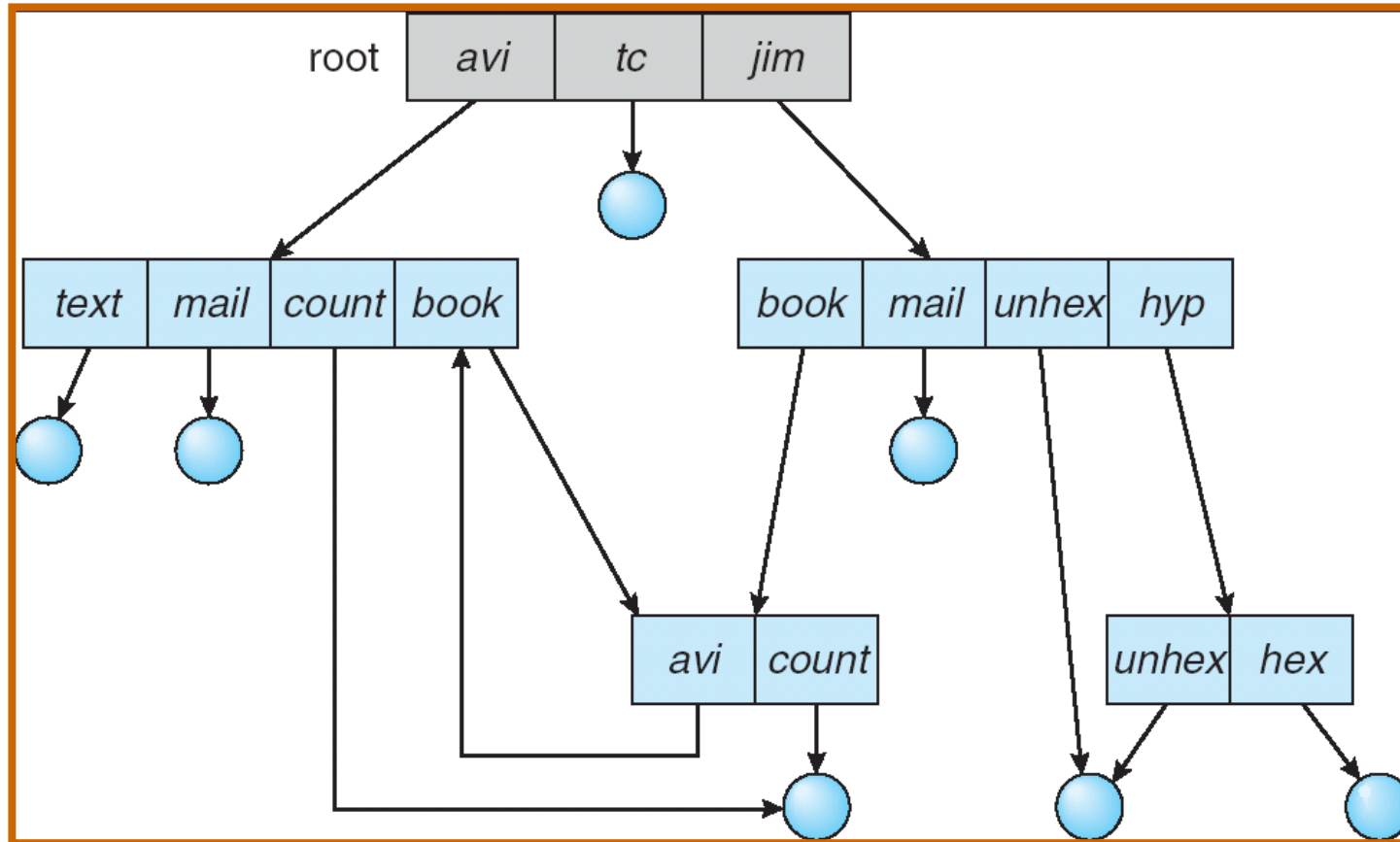


Link

- Symbolic link
 - ❑ Store path name of the real file.
 - ❑ Easy handle of deletion of link.
 - ❑ Can link to files on other machines, via network address.
 - ❑ Extra overhead to find the actual file.
- Unsymbolic link or Hard link
 - ❑ Store pointer to the data structure of the real file.
 - ❑ Use link count to handle deletion.

Source: A.S. Tanenbaum, Modern Operating Systems, 2nd ed., Prentice Hall, 2001

General Graph Directory



File Sharing – Multiple Users

- **User IDs** identify users, allowing permissions and protections to be per-user
- **Group IDs** allow users to be in groups, permitting group access rights

Protection

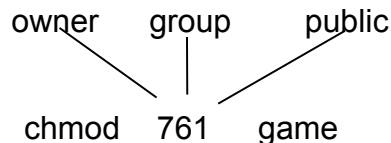
- File owner/creator should be able to control:
 - ❑ what can be done
 - ❑ by whom
- Types of access
 - ❑ **Read**
 - ❑ **Write**
 - ❑ **Execute**
 - ❑ **Append**
 - ❑ **Delete**
 - ❑ **List**

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.



Attach a group to a file
chgrp G game