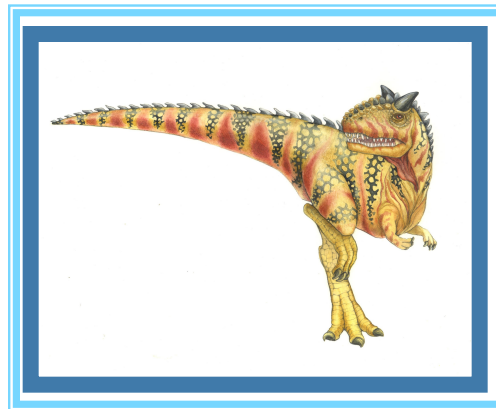


Chapter 10: File-System Interface





File System

- Two main parts:
 - A collection of files
 - A directory structure





File Concept

- Contiguous logical address space

- Types:
 - Data
 - ▶ numeric
 - ▶ character
 - ▶ binary
 - Program





File Structure

- None - sequence of words, bytes
- Simple record structure
 - Lines
 - Fixed length
 - Variable length
- Complex Structures
 - Formatted document
 - Relocatable load file





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 (i.e., seek)**
 - **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



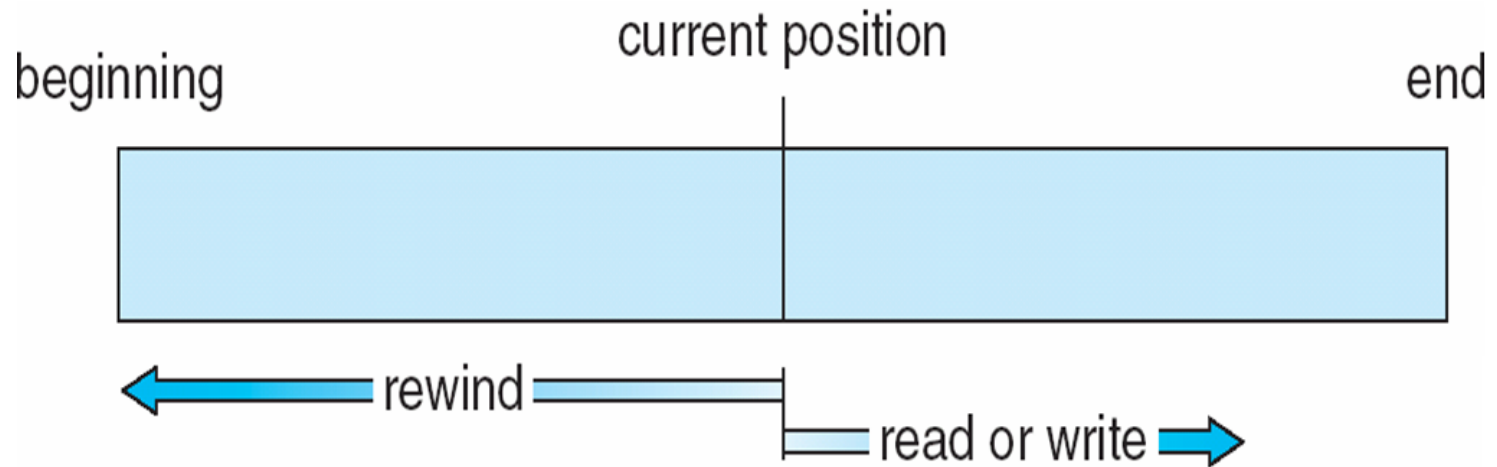


File Types – Name, Extension

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



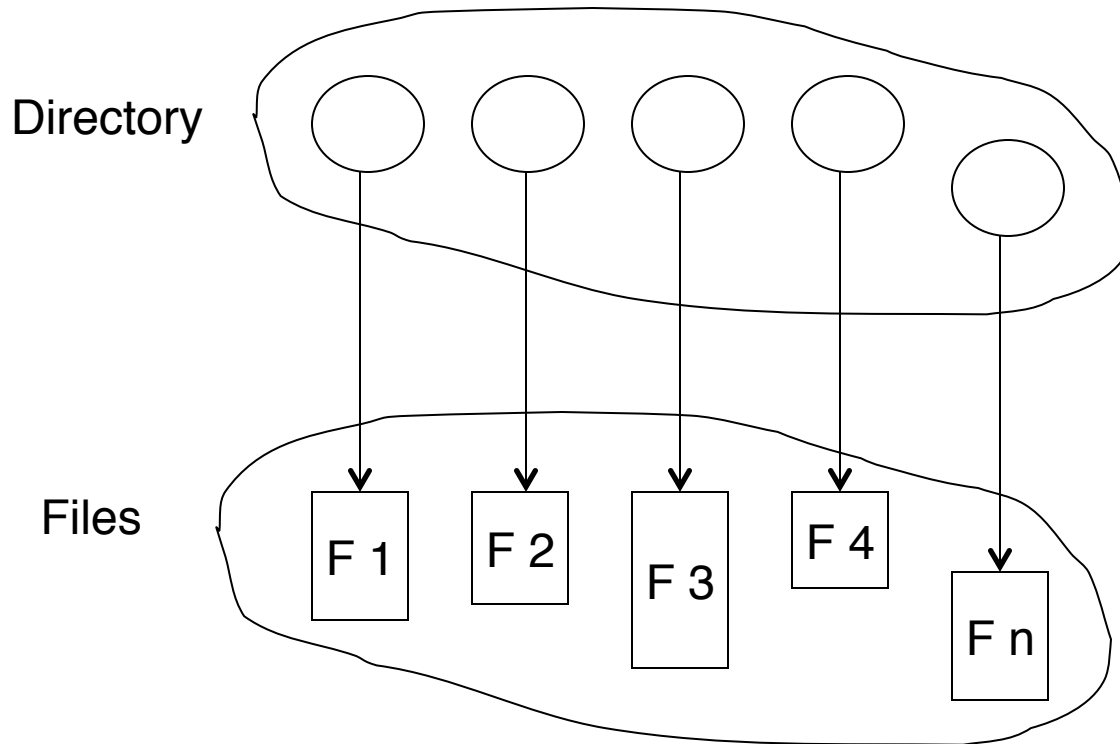
Sequential-access File





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





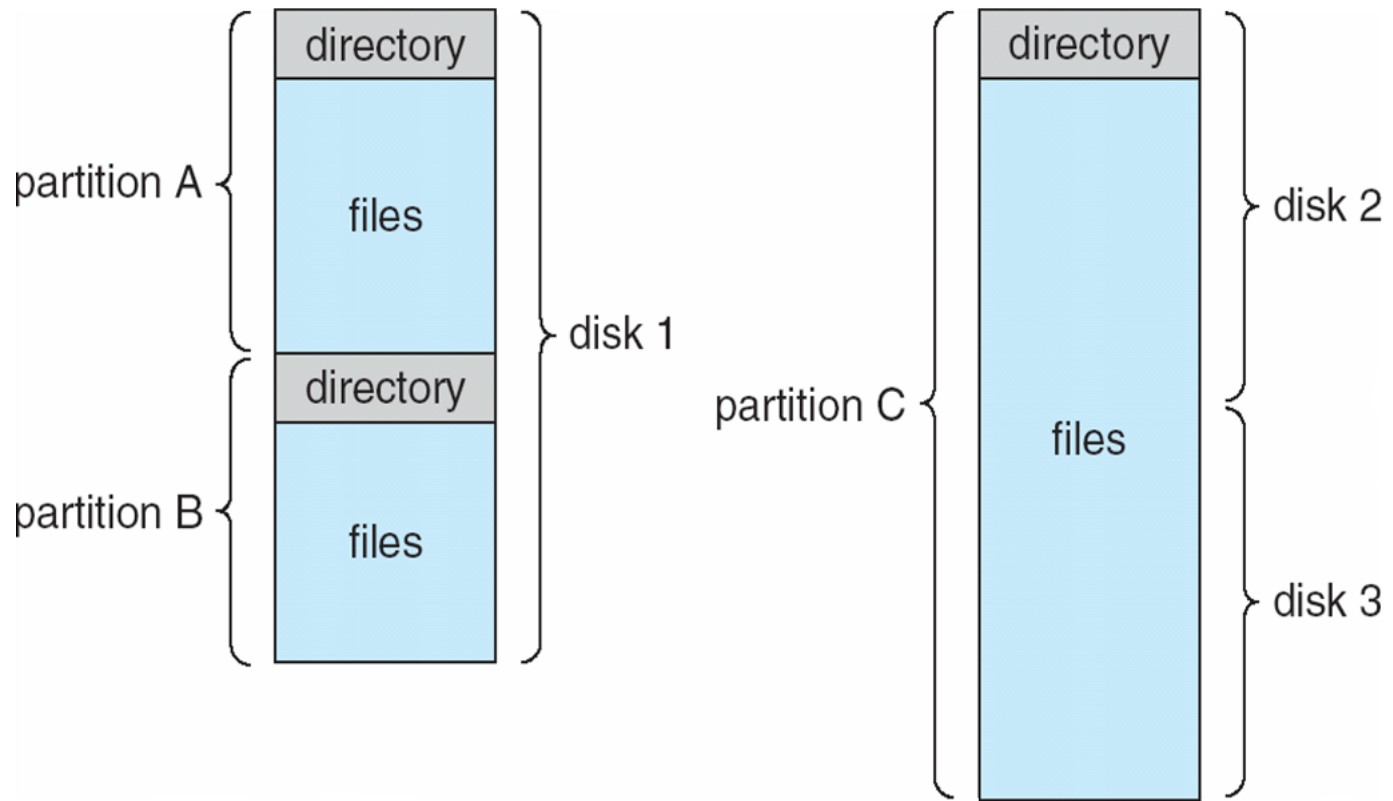
Disk Structure

- Disk can be subdivided into **partitions**
- Disks or partitions can be **RAID** protected against failure
- Disk or partition can be used **raw** – without a file system, or **formatted** with a file system
- Partitions also known as minidisks, slices
- Entity containing file system known as a **volume**
- Each volume containing file system also tracks that file system's info in **device directory** or **volume table of contents**
- As well as **general-purpose file systems** there are many **special-purpose file systems**, frequently all within the same operating system or computer





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

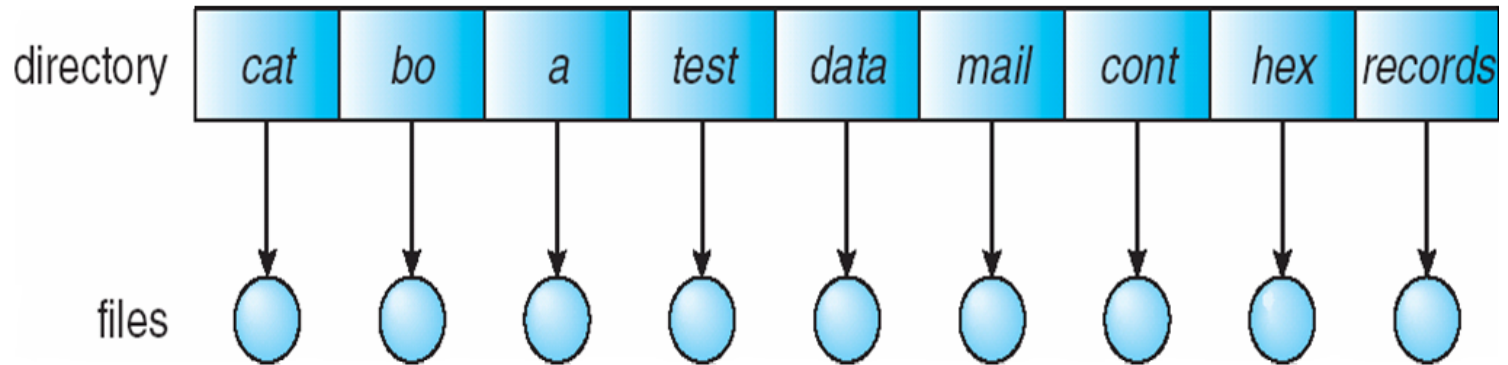
- Efficiency – locating a file quickly
- Naming – convenient to users
 - Two users can have same name for different files
 - The same file can have several different names
- Grouping – logical grouping of files by properties, (e.g., all Java programs, all games, ...)





Single-Level Directory

- A single directory for all users



Naming problem

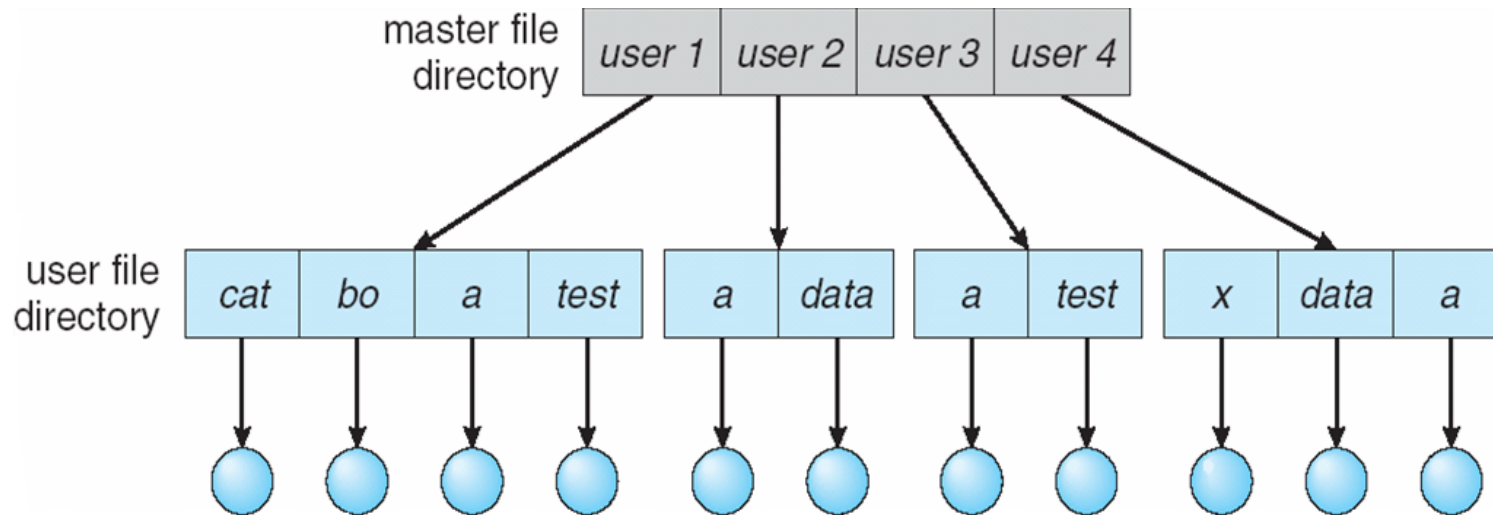
Grouping problem





Two-Level Directory

- Separate directory for each user

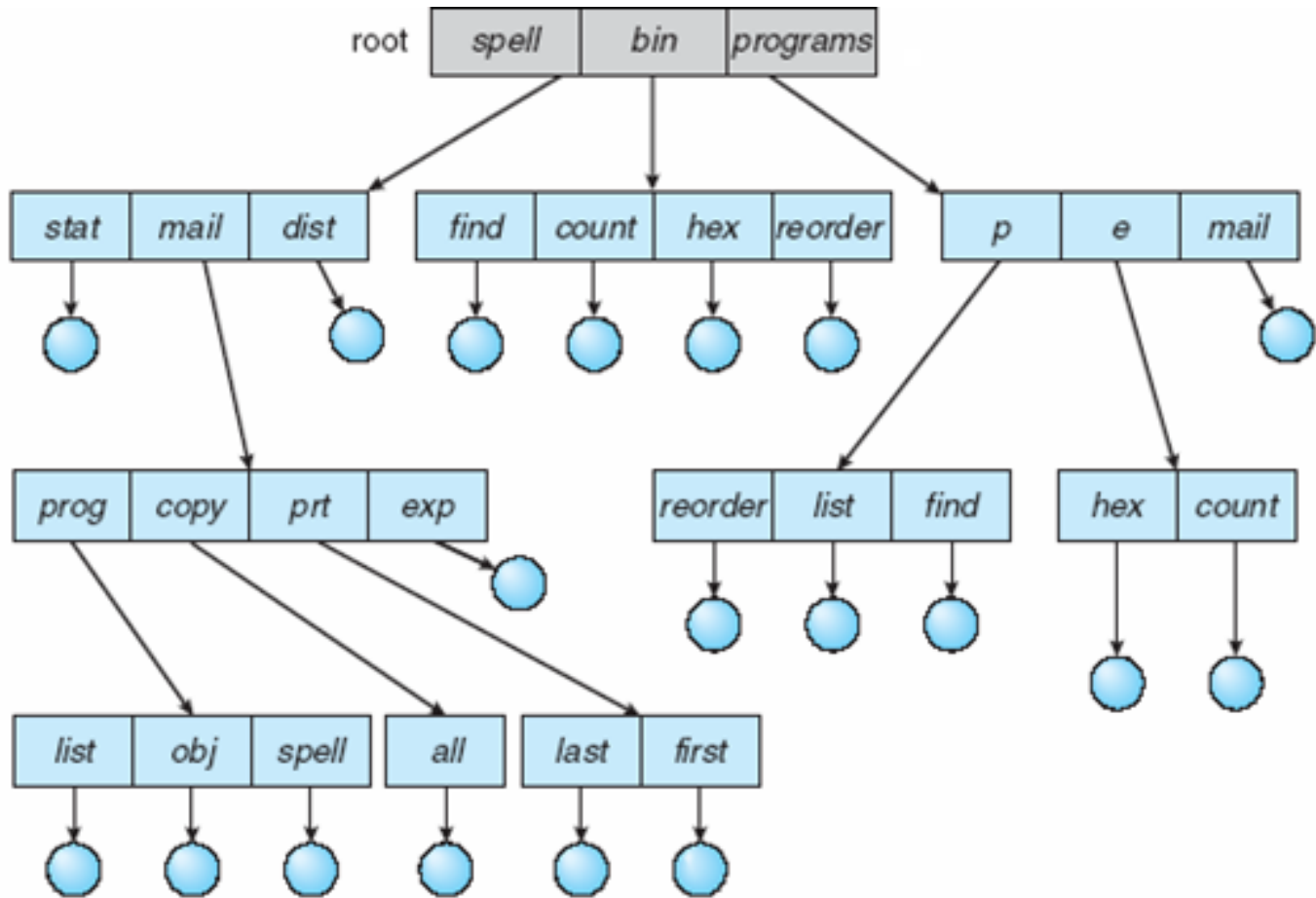


- Path name
- Can have the same file name for different user
- Efficient searching
- No grouping capability





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

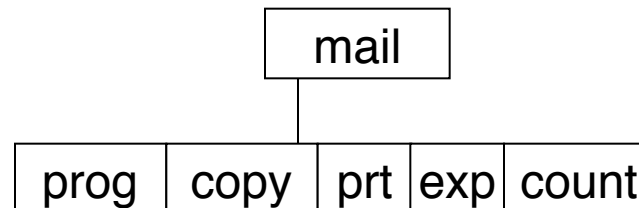
`rm <file-name>`

- Creating a new subdirectory is done in current directory

`mkdir <dir-name>`

Example: if in current directory `/mail`

`mkdir count`



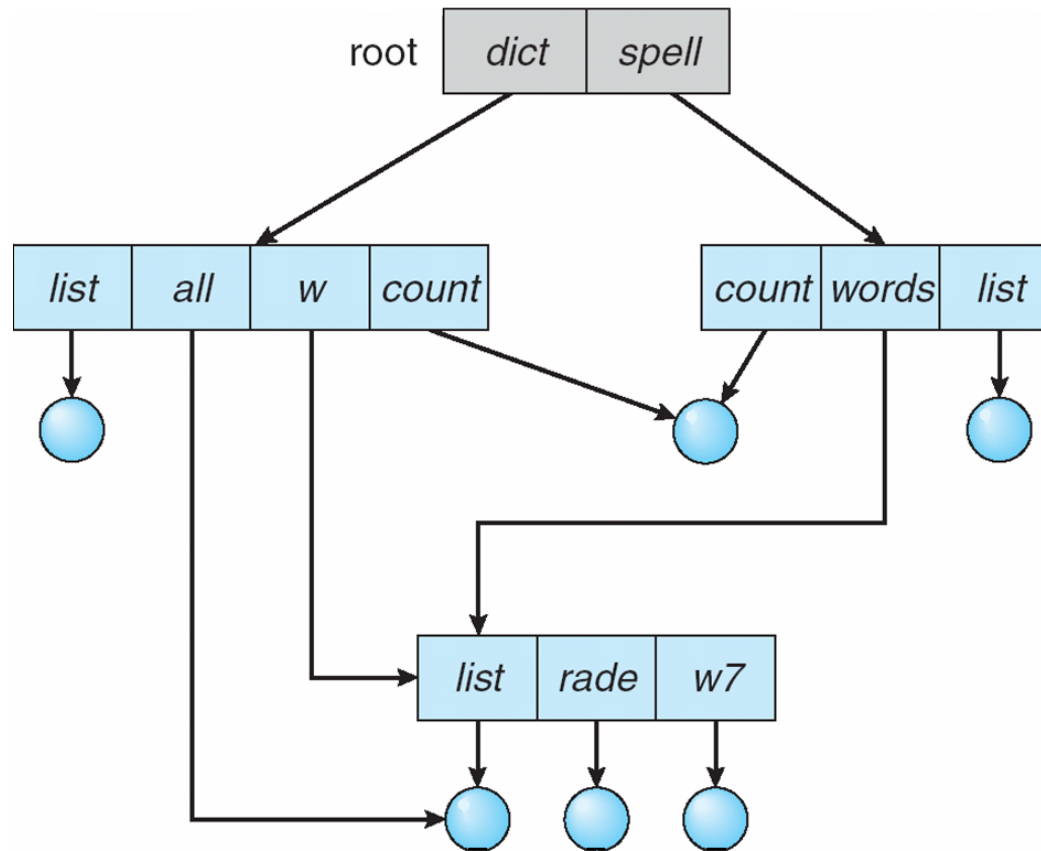
Deleting “mail” ⇒ deleting the entire subtree rooted by “mail”





Acyclic-Graph Directories

- Have shared subdirectories and files



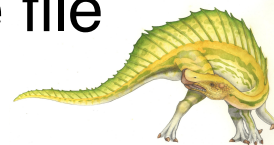


Acyclic-Graph Directories (Cont.)

- Two different names (aliasing)
- If *dict* deletes *list* \Rightarrow dangling pointer

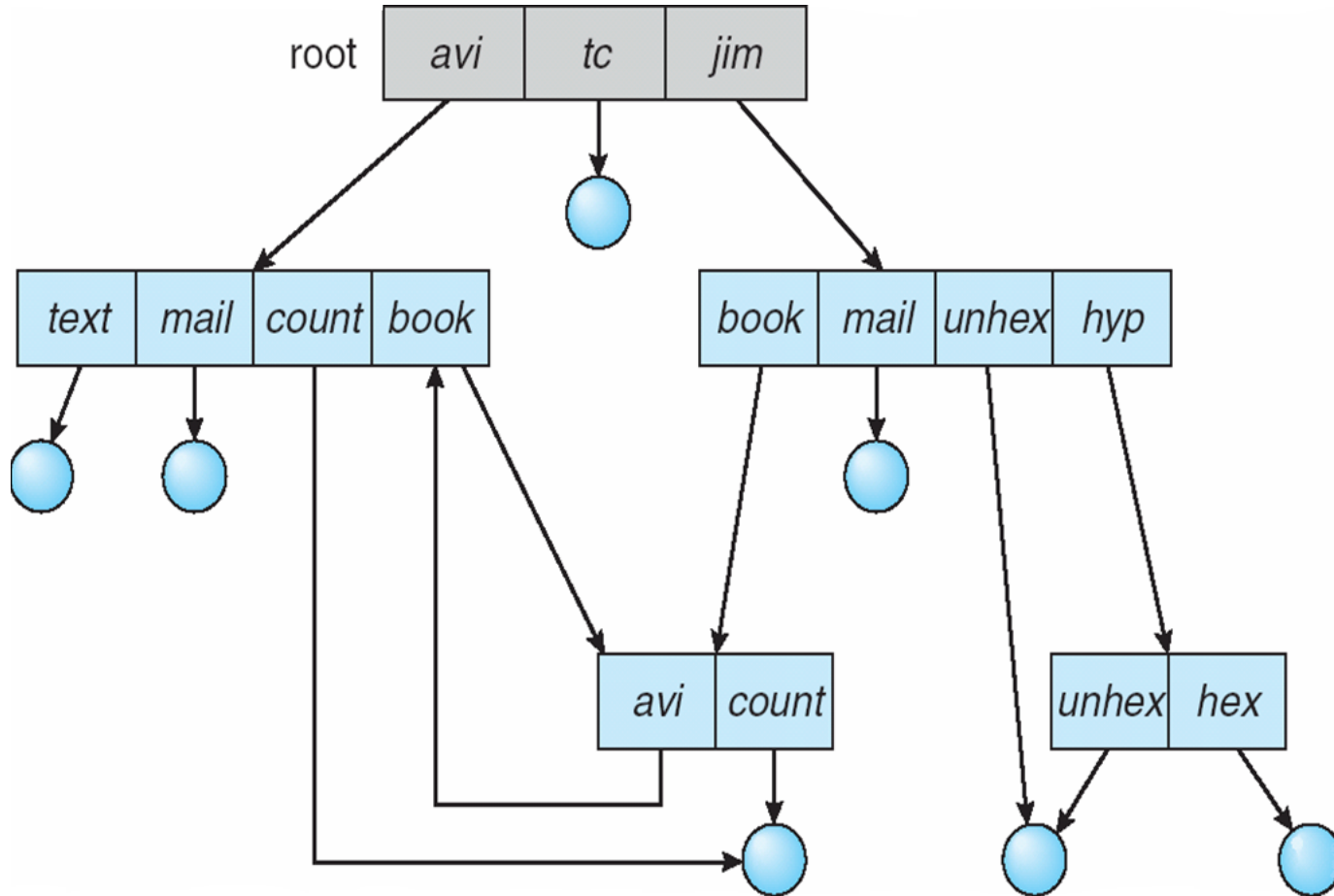
Solutions:

- Backpointers, so we can delete all pointers
Variable size records a problem
- Backpointers using a daisy chain organization
- Entry-hold-count solution
- New directory entry type
 - **Link** – another name (pointer) to an existing file
 - **Resolve the link** – follow pointer to locate the file





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





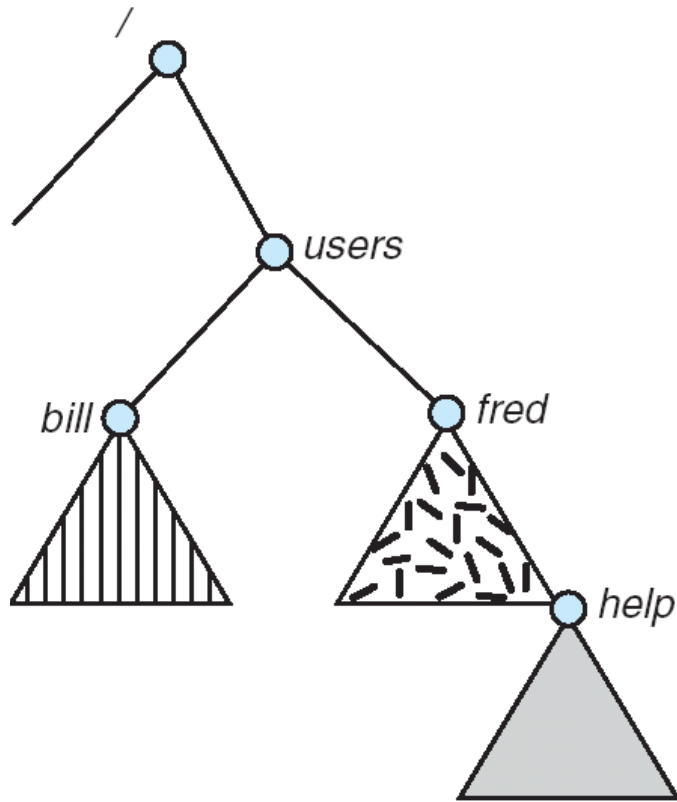
File System Mounting

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

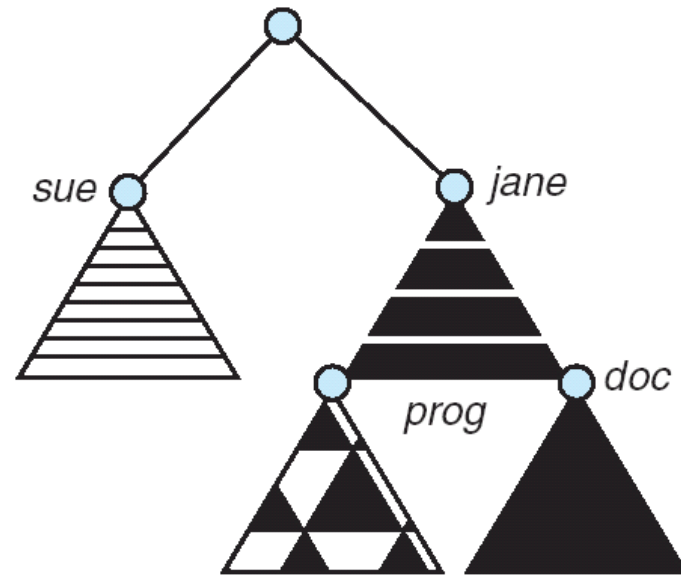




(a) Existing. (b) Unmounted Partition



(a)

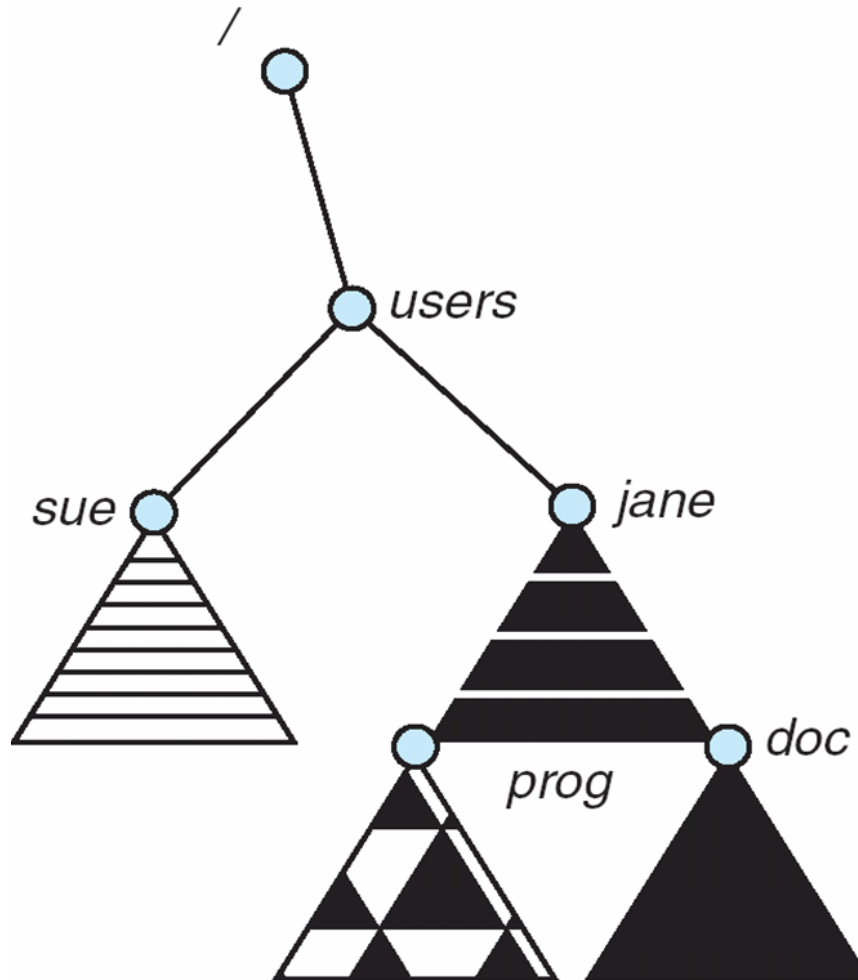


(b)





Mount Point





File Sharing

- Sharing of files on multi-user systems is desirable
- 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





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**



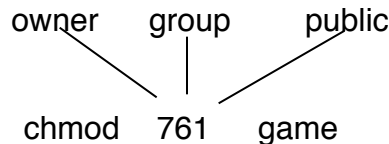


Access Lists and Groups

- 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





Windows XP Access-control List Management

10.tex Properties

General Security Summary

Group or user names:

- Administrators (PBG-LAPTOP\Administrators)
- Guest (PBG-LAPTOP\Guest)**
- pbg (CTI\pbg)
- SYSTEM
- Users (PBG-LAPTOP\Users)

Add... Remove

Permissions for Guest

	Allow	Deny
Full Control	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Modify	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Read & Execute	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Read	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Write	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Special Permissions	<input type="checkbox"/>	<input type="checkbox"/>

For special permissions or for advanced settings, click Advanced.

Advanced

OK Cancel Apply





A Sample UNIX Directory Listing

```
-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/
```



End of Chapter 10

