Chapter 10: File-System Interface

Chapter 10: File-System Interface

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

Objectives

- To explain the function of file systems
- To describe the interfaces to file systems
- To discuss file-system design tradeoffs, including access methods, file sharing, file locking, and directory structures
- To explore file-system protection

File Concept

- Contiguous logical address space
- Types:
  - Data
    - text
    - binary
  - Program

File Attributes

- Name – only information kept in human-readable form
- Identifier – unique tag 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 for F_i, and copy the content to memory
- *Close (F_i)*
  - move the content of F_i in memory to disk

Open Files

- Several pieces of data are needed to manage open files:
  - File pointer
    - Pointer to last read/write location
  - 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

Open File Locking

- Provided by some OSes and file systems
- Mediates access to a file
- Mandatory or advisory:
  - Mandatory
    - Access is denied depending on locks held and requested
    - Windows
  - Advisory
    - Processes can find status of locks and decide what to do
    - Unix/Linux

File Types – Name, Extension

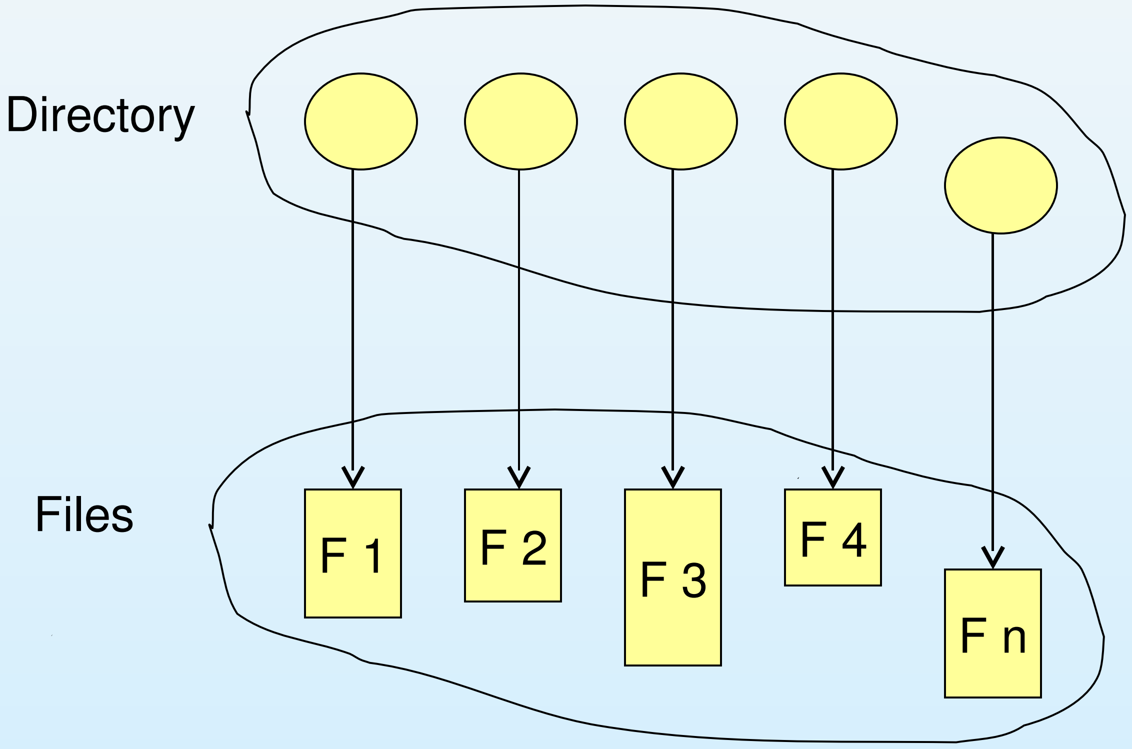
file type	usual extension	function
executable	exe, com, bin or none	ready-to-run machine-language program
object	obj, o	compiled, machine language, not linked
source code	c, cc, java, pas, asm, a	source code in various languages
batch	bat, sh	commands to the command interpreter
text	txt, doc	textual 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 compressed, for archiving or storage
multimedia	mpeg, mov, rm, mp3, avi	binary file containing audio or A/V information

File Structure

- OS view
  - None - sequence of words, bytes
- Simple record structure
  - Lines
  - Fixed length
  - Variable length
- Complex Structures
  - Formatted document
  - Relocatable load file
- Who decides:
  - Program

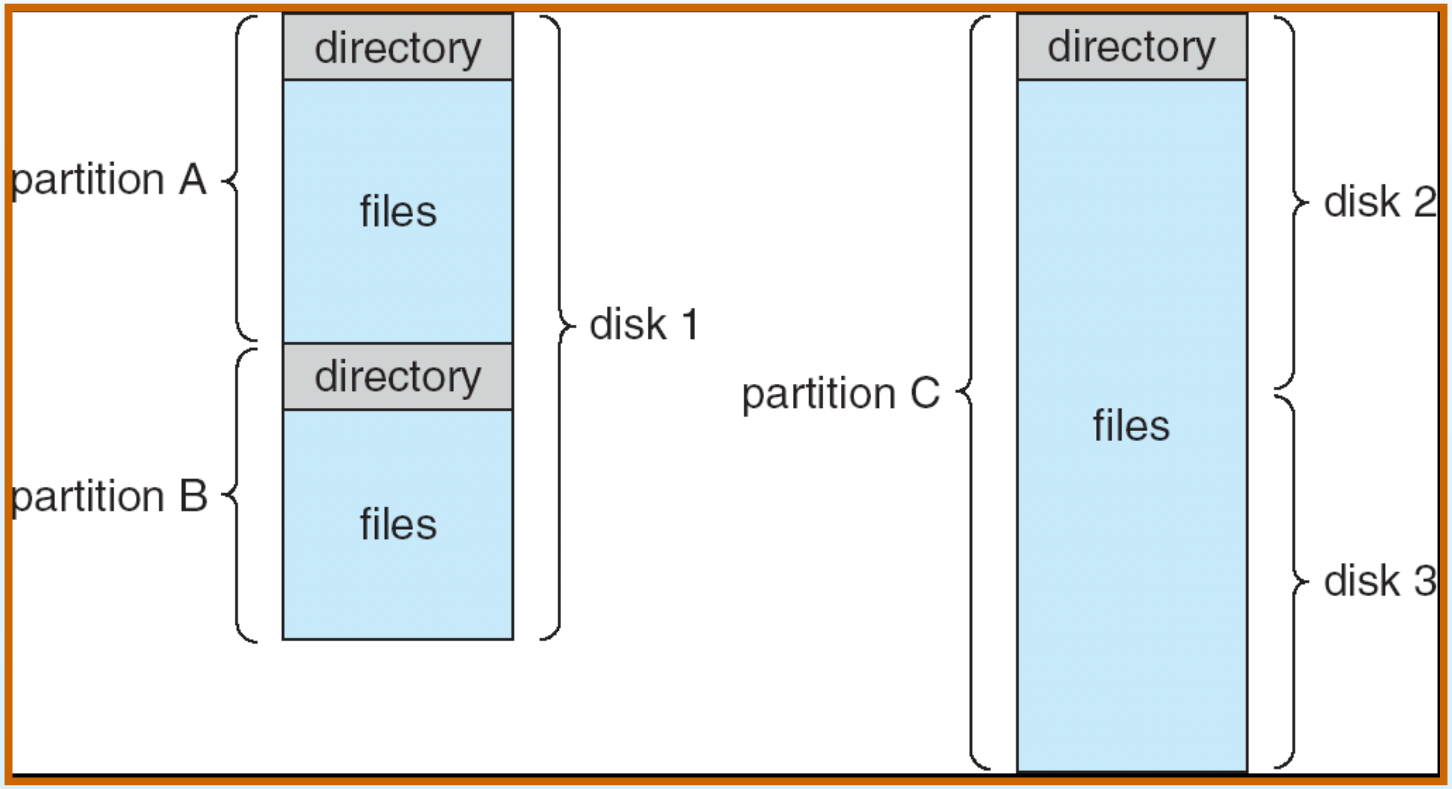
Directory Structure

- **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

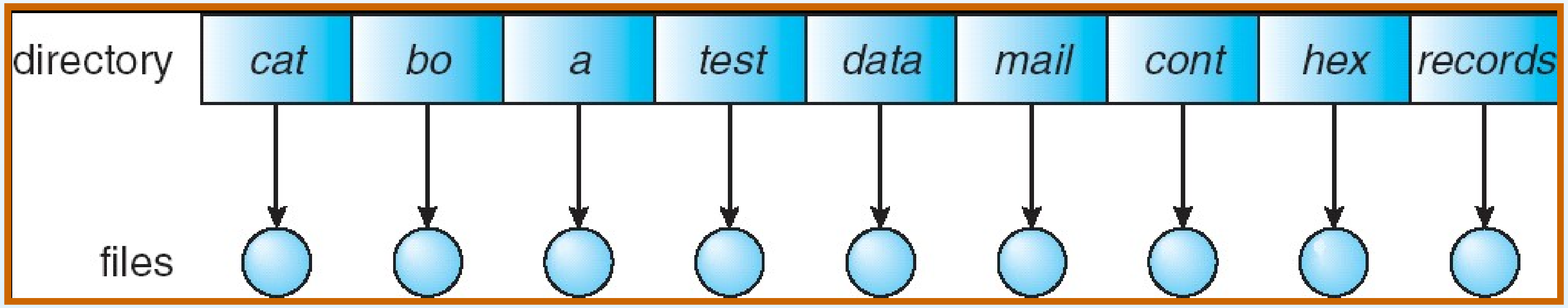
- Search for a file
- Create a file
- Delete a file
- List a directory
- Rename a file

Organize the Directory (Logically) to Obtain

- Efficiency
  - Locating a file quickly
- Naming
  - 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, …)

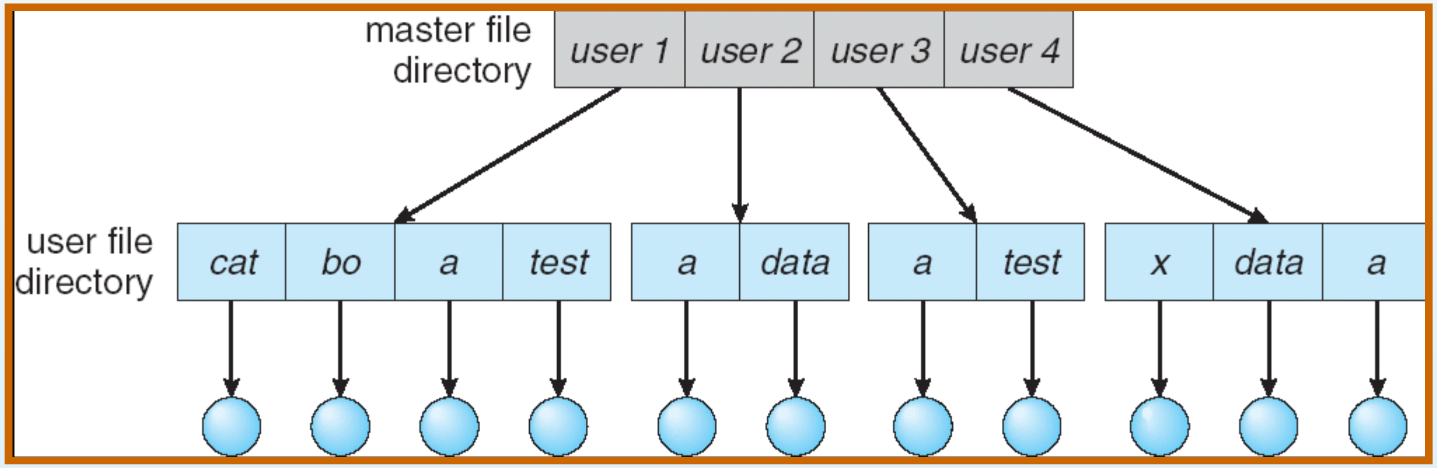
Single-Level Directory

- A single directory for all users



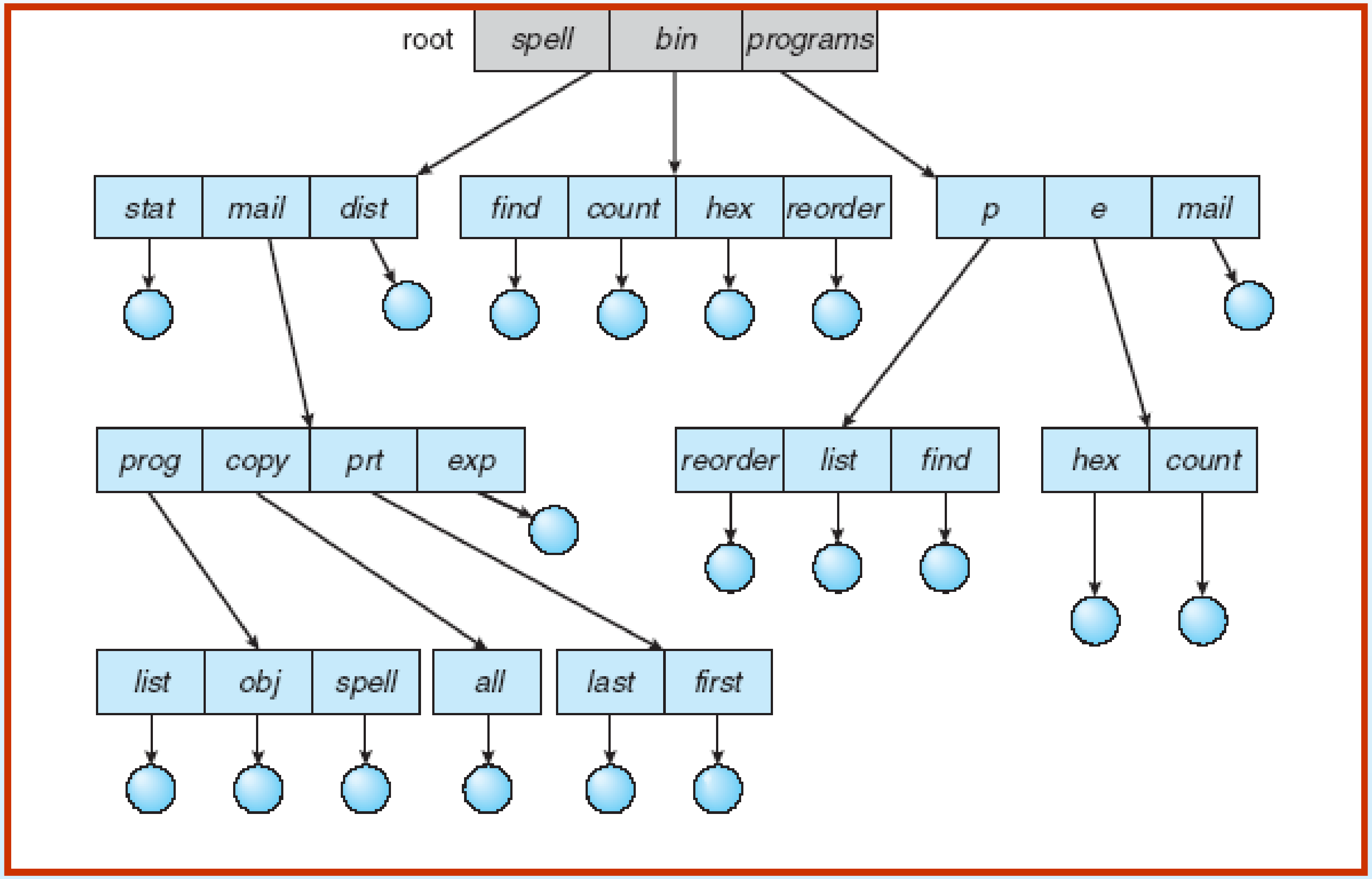
Two-Level Directory

- 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
  - cat list

Tree-Structured Directories (Cont)

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

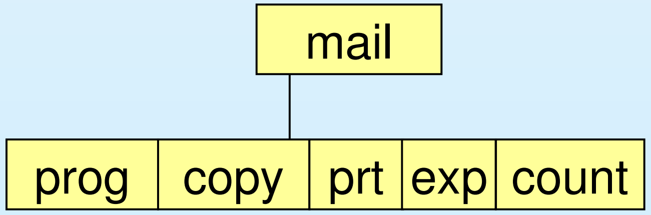
  rm <file-name>

- Creating a new subdirectory is done in current directory

  mkdir <dir-name>

  Example: if in current directory /mail

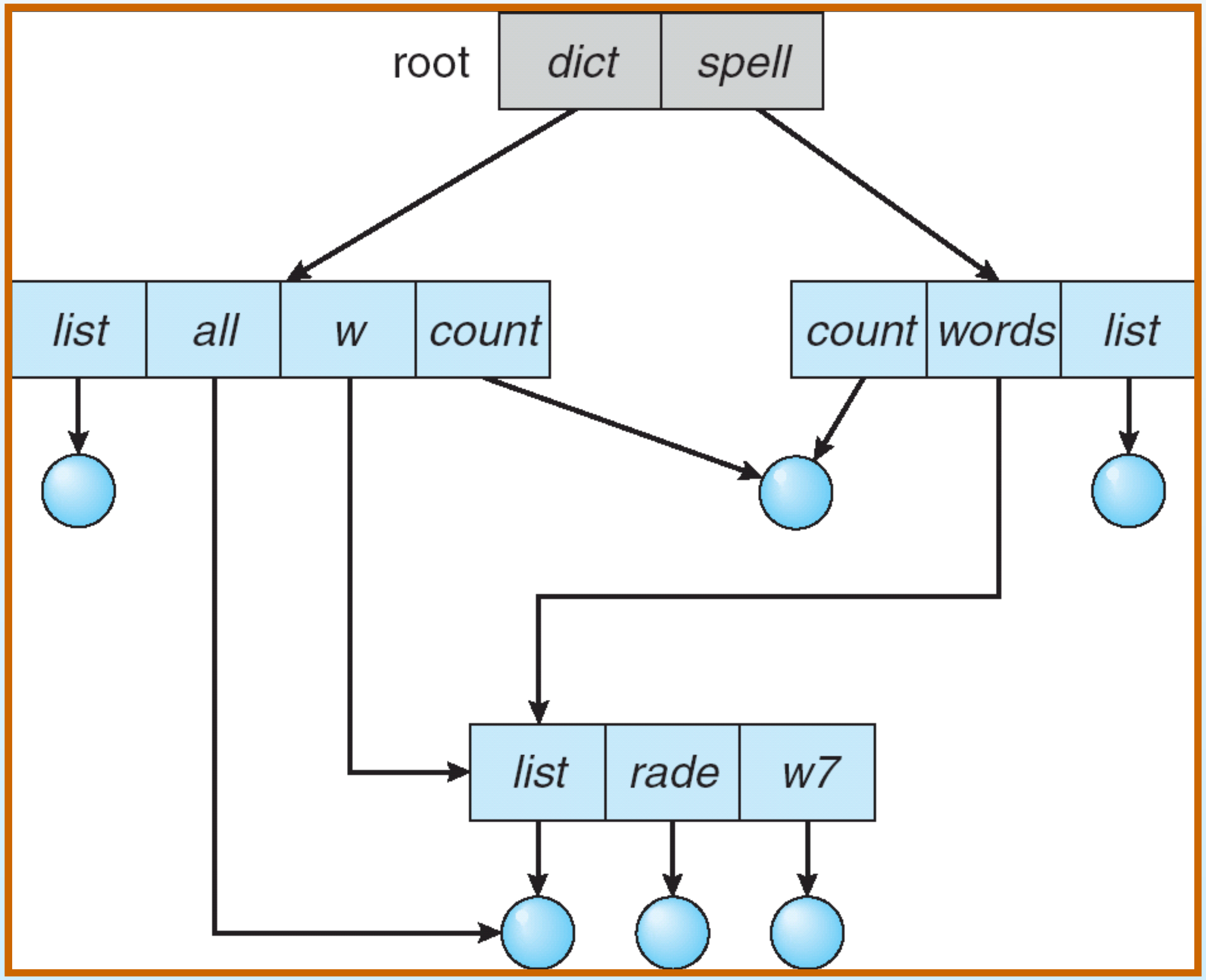
  mkdir count



Deleting "mail" $\Rightarrow$ deleting the entire subtree rooted by "mail"

Acyclic-Graph Directories

- Have shared subdirectories and files



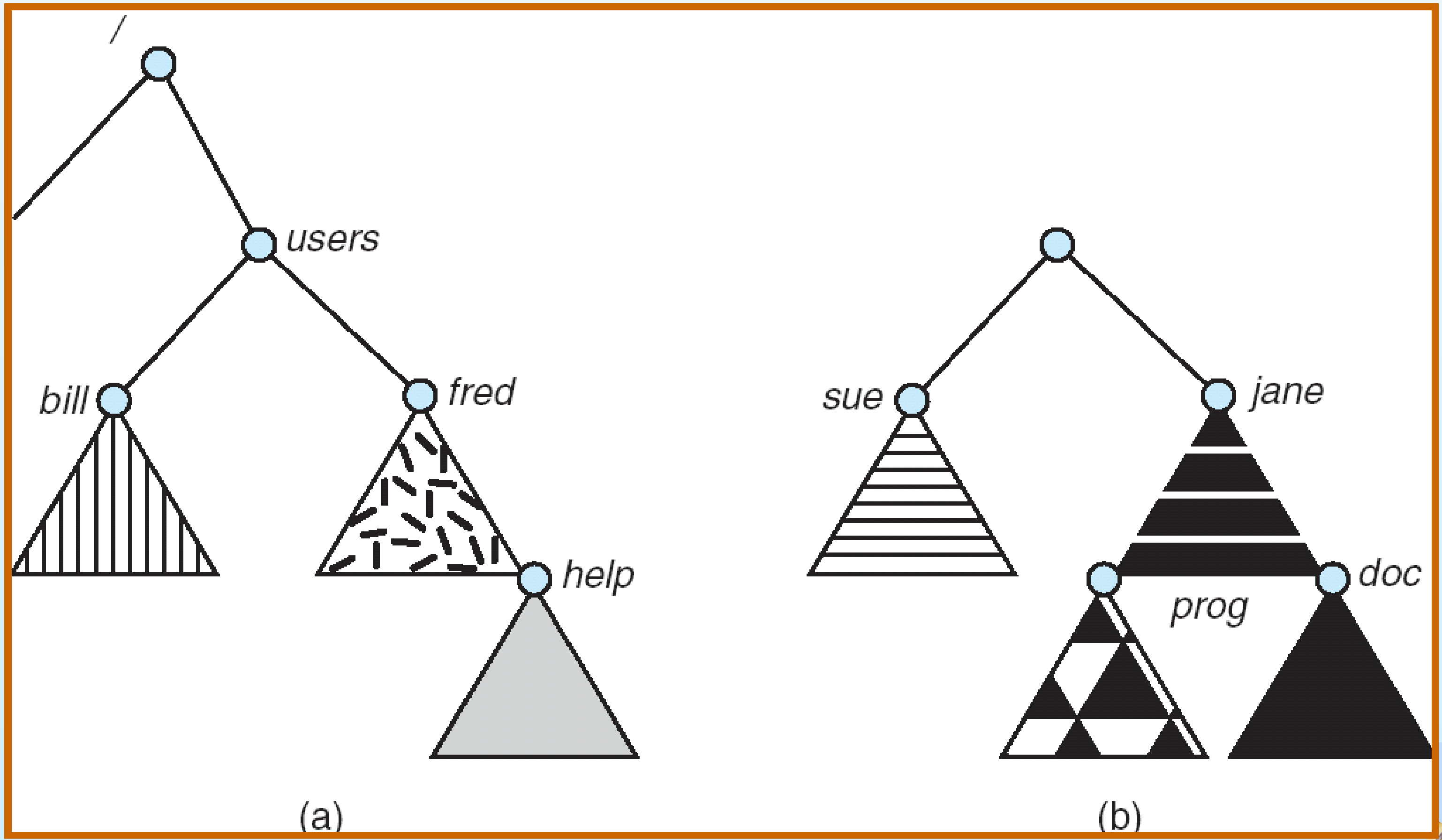
Acyclic-Graph Directories (Cont.)

- Two different names (aliasing)
- If *dict* deletes *all* ⇒ dangling pointer

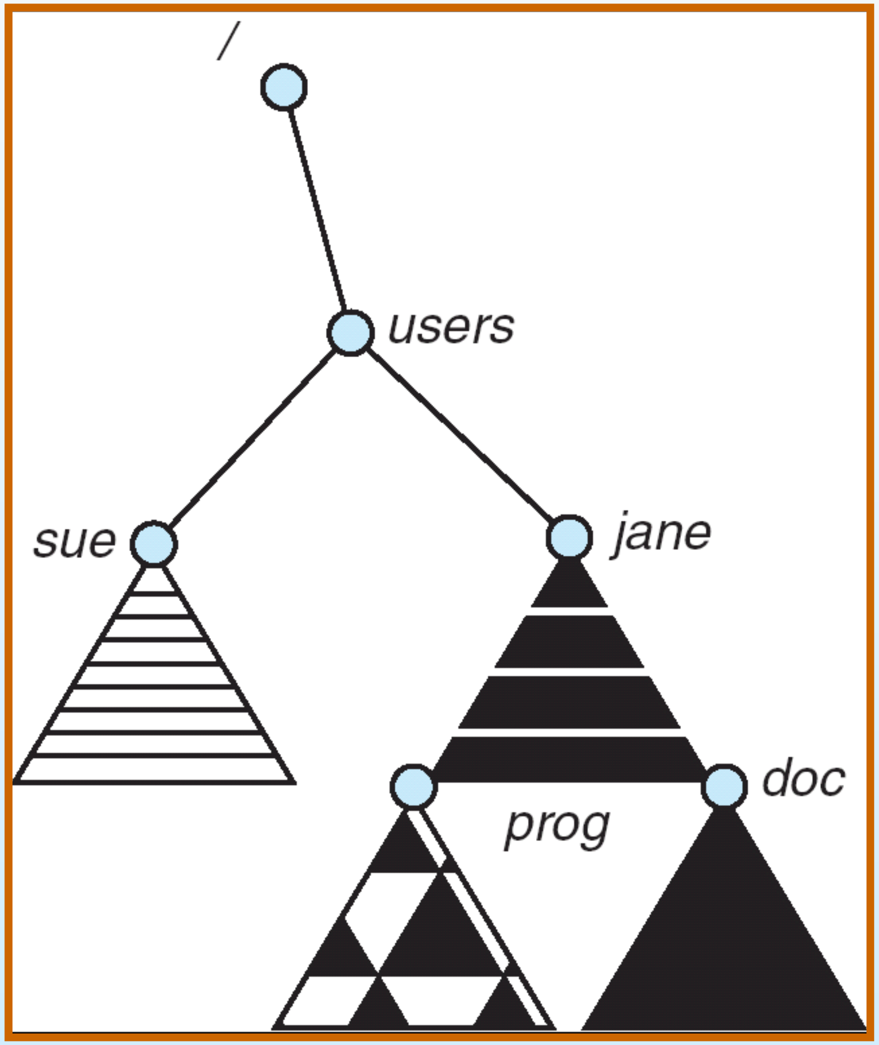
  Solutions:
  - Doesn't care
  - Entry-hold-count solution
- New directory entry type
  - Link – another name (pointer) to an existing file
  - Resolve the link – follow pointer to locate the file

File System Mounting

- A file system must be mounted before access



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 – Remote File Systems

- Uses networking to allow file system access between systems
  - Manually via programs like FTP
  - Automatically, seamlessly using distributed file systems
  - WebDAV
- Clients mount remote file systems from servers
  - Server can serve multiple clients
  - Client and user-on-client identification is insecure or complicated
  - NFS is standard UNIX client-server file sharing protocol
  - CIFS is standard Windows protocol
  - Standard OS file calls are translated into remote calls

Protection

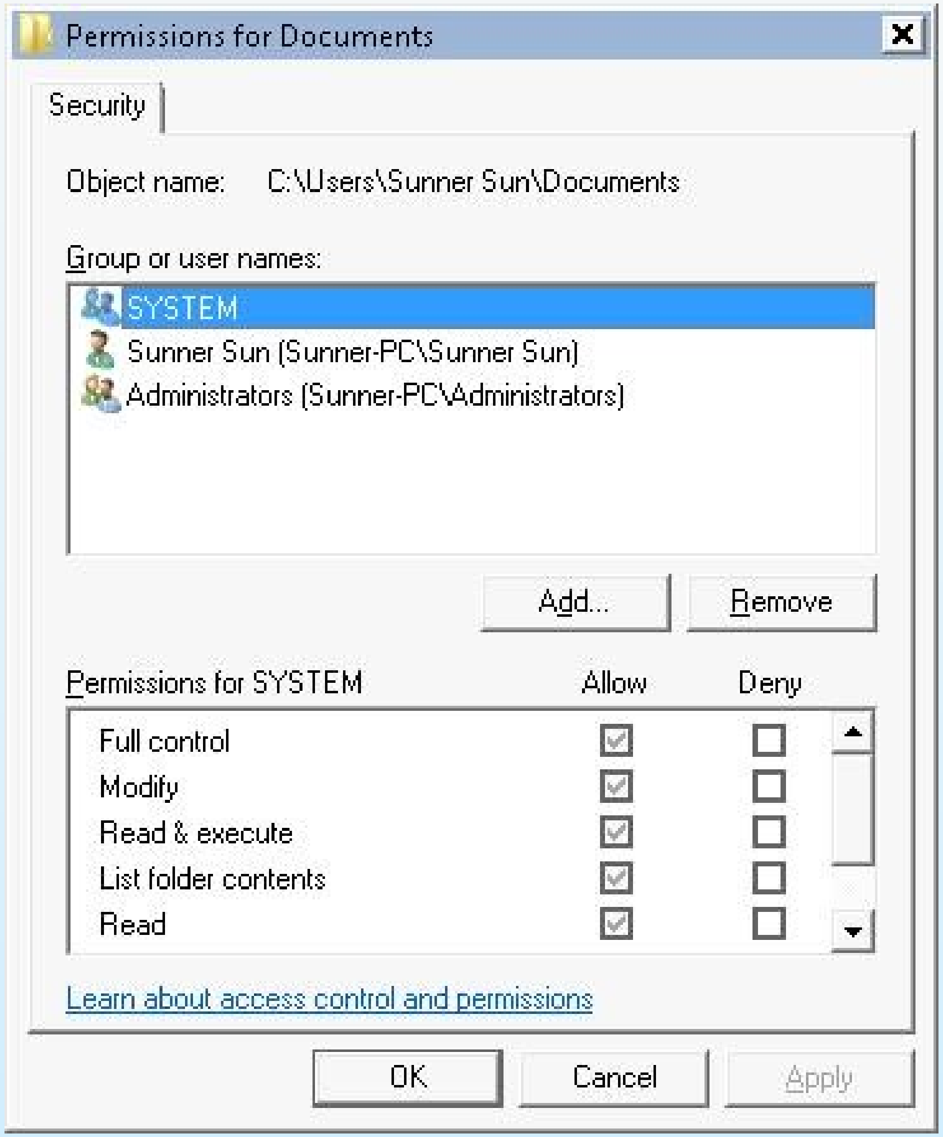
- File owner/creator should be able to control:
  - what can be done
  - by whom
- Types of access
  - Read
  - Write
  - Execute
  - Append
  - Delete
  - List

Access Lists and Groups

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

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

Windows XP Access-control List Management


Permissions for Documents

Security

Object name: C:\Users\Sunner Sun\Documents

Group or user names:

- SYSTEM
- Sunner Sun (Sunner-PC\Sunner Sun)
- Administrators (Sunner-PC\Administrators)

Add... Remove

Permissions for SYSTEM	Allow	Deny
Full control	✓	☐
Modify	✓	☐
Read & execute	✓	☐
List folder contents	✓	☐
Read	✓	☐

Learn about access control and permissions

OK Cancel Apply

End of Chapter 10