L36- FILE DIRECTORIES

File-System Interface

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

Objectives

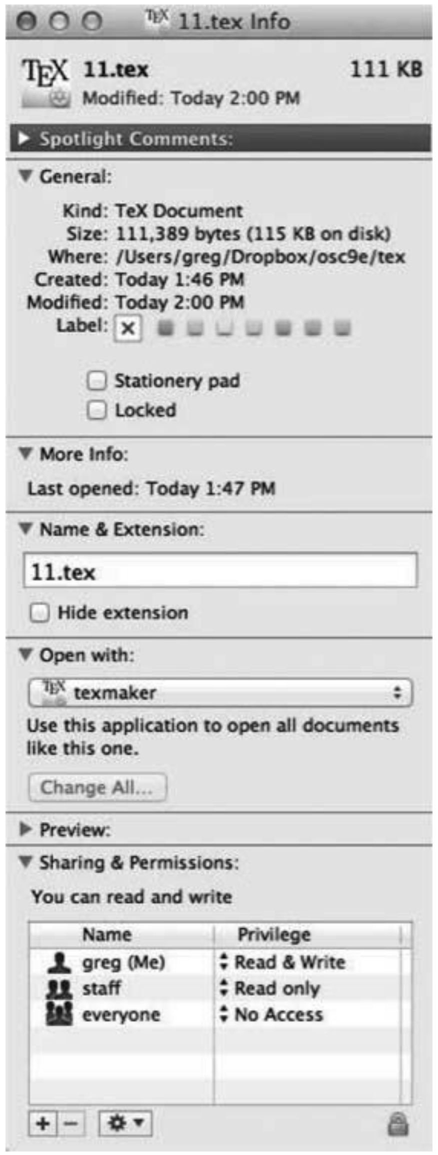
- To explain the function of file systems
- To describe the interfaces to file systems
- To discuss file-system design tradeoffs, including access methods, file sharing, file locking, and directory structures
- To explore file-system protection

File Concept

- Contiguous logical address space
- Types:
  - Data
    - numeric
    - character
    - binary
  - Program
- Contents defined by file’s creator
  - Many types
    - Consider **text file, source file, executable file**

File Attributes

- **Name** – users identify a file with name.
- **Identifier** – unique number identifies file within file system
- **Type** – Format of data inside, application that can access it.
- **Location** – pointer to file location on device
- **Size** – amount of storage the file consumes
- **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



11.tex Info

11.tex 111 KB
Modified: Today 2:00 PM

▶ Spotlight Comments:

▼ General:

Kind: TeX Document
Size: 111,389 bytes (115 KB on disk)
Where: /Users/greg/Dropbox/osc9e/tex
Created: Today 1:46 PM
Modified: Today 2:00 PM
Label:

☐ Stationery pad
☐ Locked

▼ More Info:

Last opened: Today 1:47 PM

▼ Name & Extension:

11.tex

☐ Hide extension

▼ Open with:

texmaker

Use this application to open all documents like this one.

Change All...

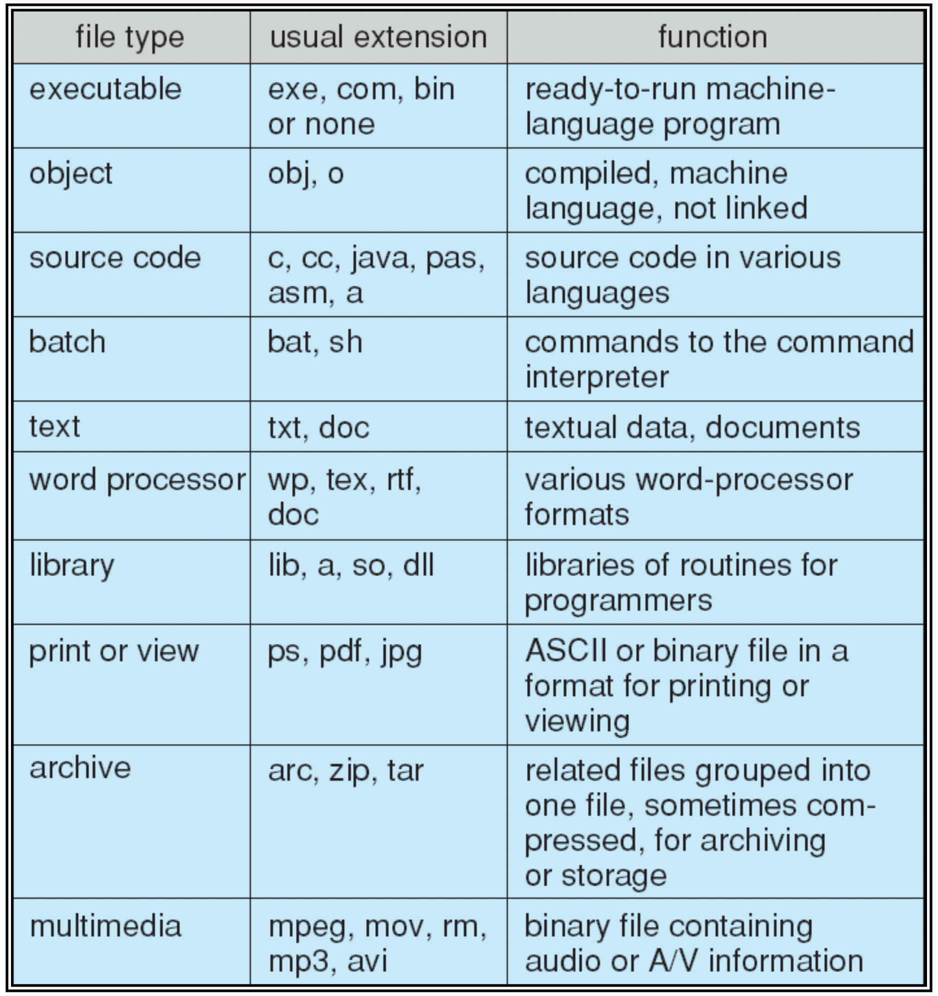
▶ Preview:

▼ Sharing & Permissions:

You can read and write

Name	Privilege
greg (Me)	Read & Write
staff	Read only
everyone	No Access

File Types & 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 Operations

- **Create**
- **Write –** at **write pointer** location
- **Read –** at **read pointer** location
- **Reposition within file - seek**
- **Delete**
- **Truncate**
- **Open(F)** – search the directory structure on disk for entry **F**, and move the content of entry to memory
- **Close (F)** – move the content of entry **F** in memory to directory structure on disk

File Open Operation

- Several pieces of data are needed to manage open files:
  - **Open-file table**: tracks 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

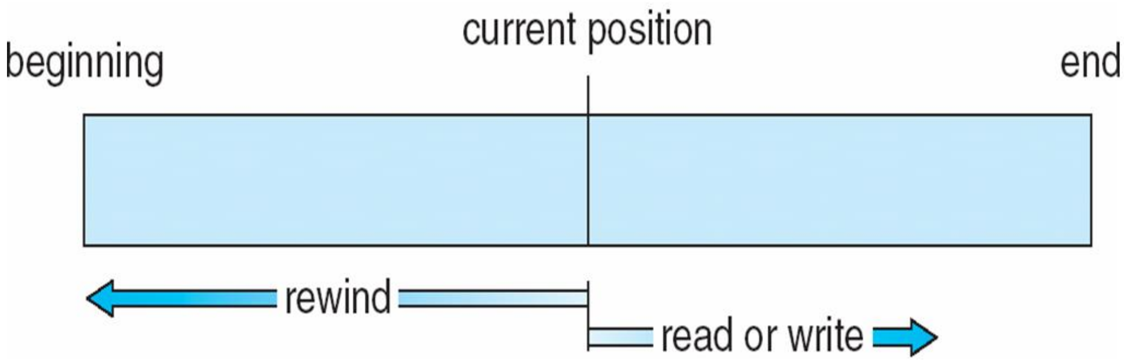
Open File Locking

- **Shared lock** similar to reader lock – several processes can acquire concurrently
- **Exclusive lock** similar to writer lock
- Mediates access to a file
- Mandatory or advisory:
  - **Mandatory** – access is denied depending on locks held and requested
  - **Advisory** – processes can find status of locks and decide what to do

Access Methods

- **Sequential Access**

```
read next
write next
reset
```



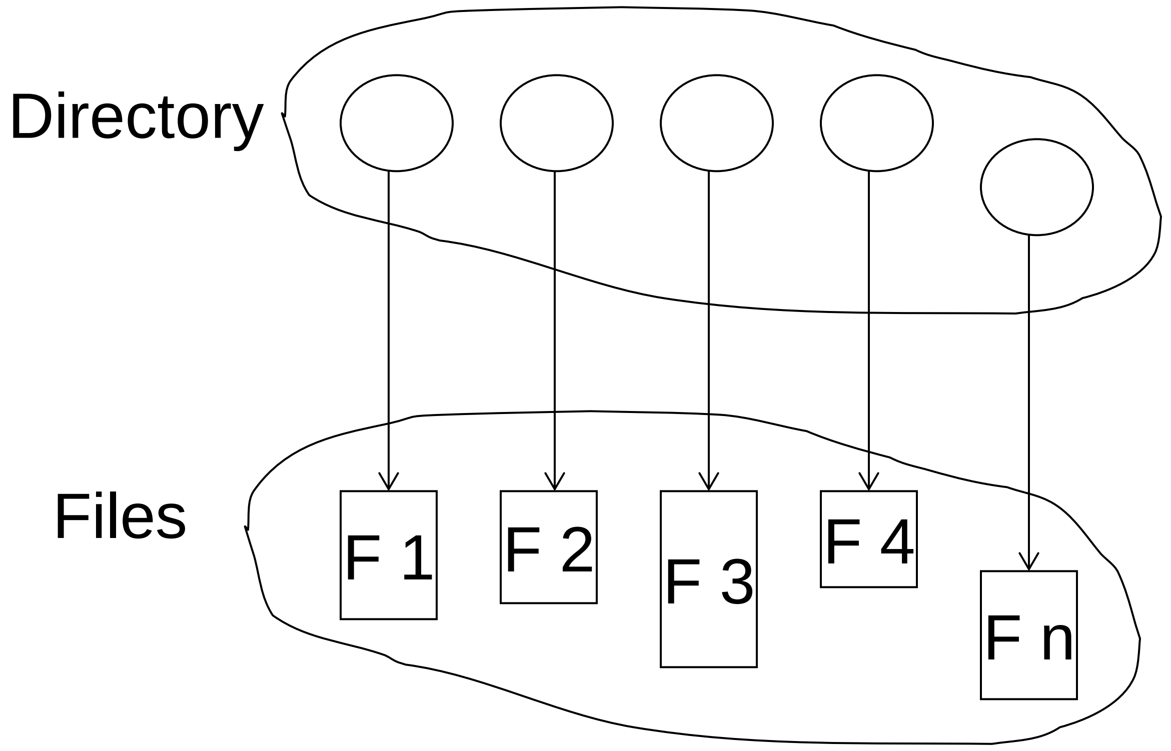
- **Direct Access**

```
write n
position to n
read next
write next
rewrite n
```

n = relative block number

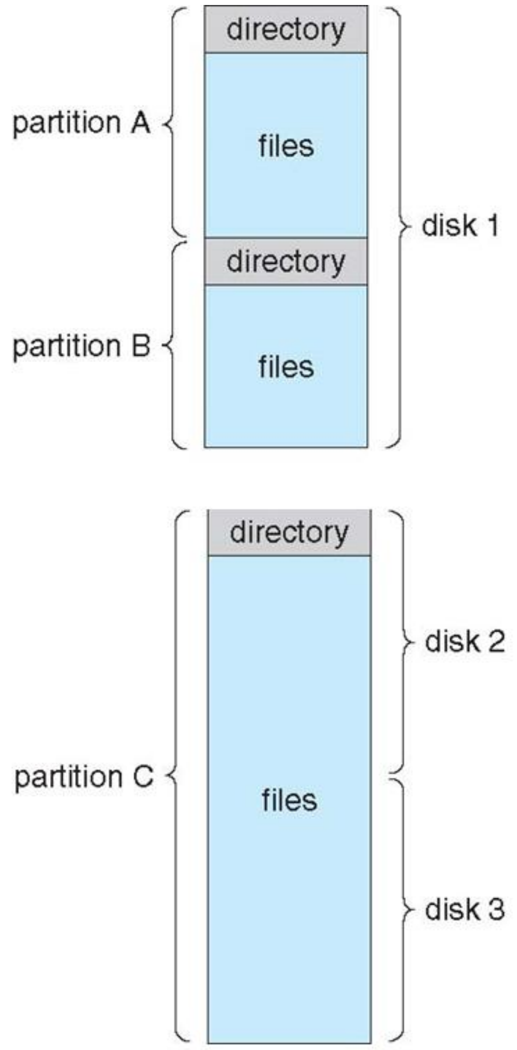
Directory Structure

- ❖ A collection of nodes containing information about all files
- ❖ Both the directory structure and the files reside on disk



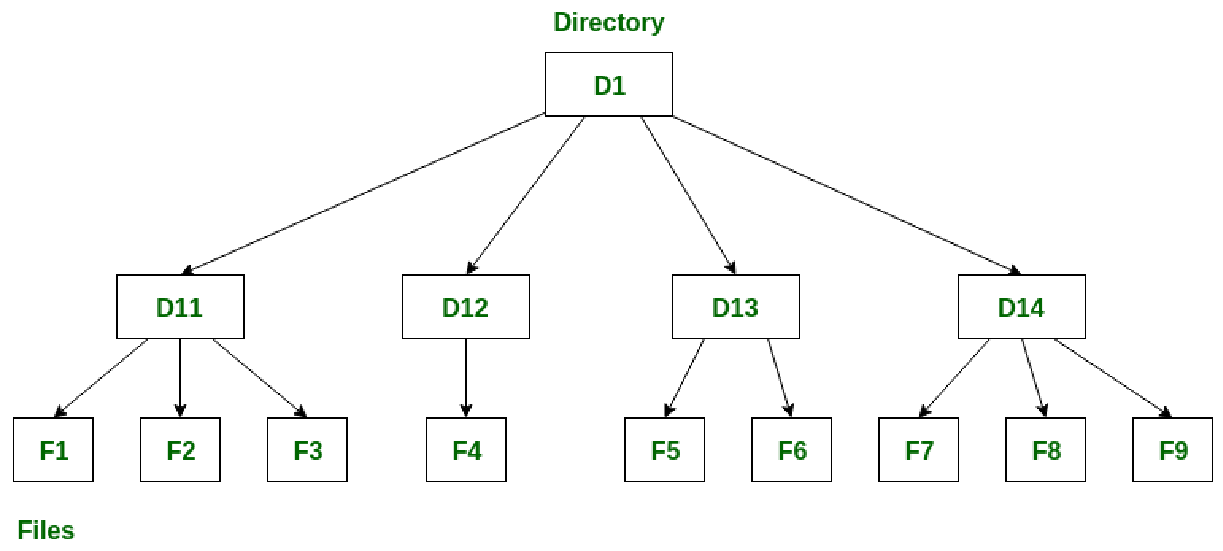
Disk Structure

- 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
- Each partition contains a file system known as a **volume** that tracks that file system's info in **device directory** or **volume table of contents**



Operations Performed on Directory

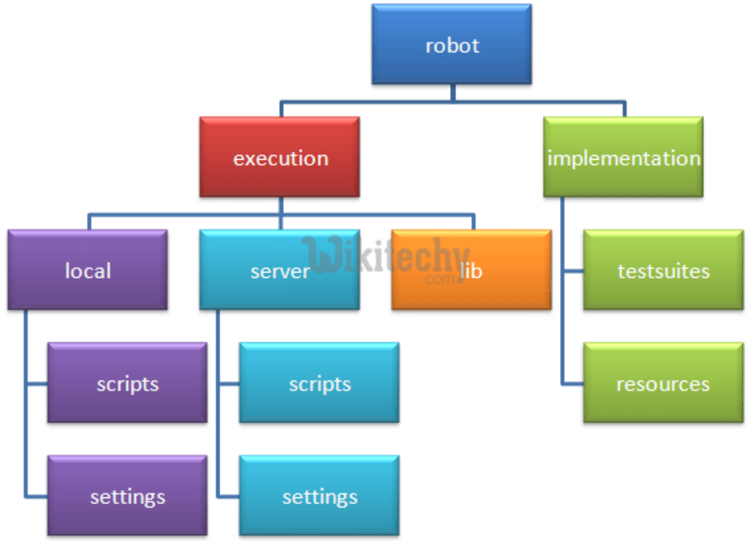
- ❖ Search for a file
- ❖ Create a file
- ❖ Delete a file
- ❖ List a directory
- ❖ Rename a file
- ❖ Traverse the file system



Directory Organization

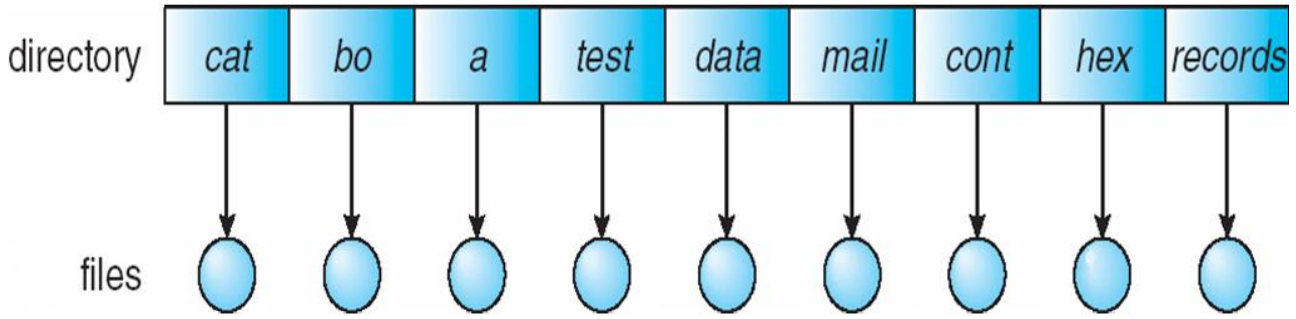
- Efficiency – locating a file quickly
- Naming – convenient to users
  - Two users can have same name for different files
  - The same file can have several different names
- Grouping – logical grouping of files by properties,

  (e.g., all programs, all games, …)



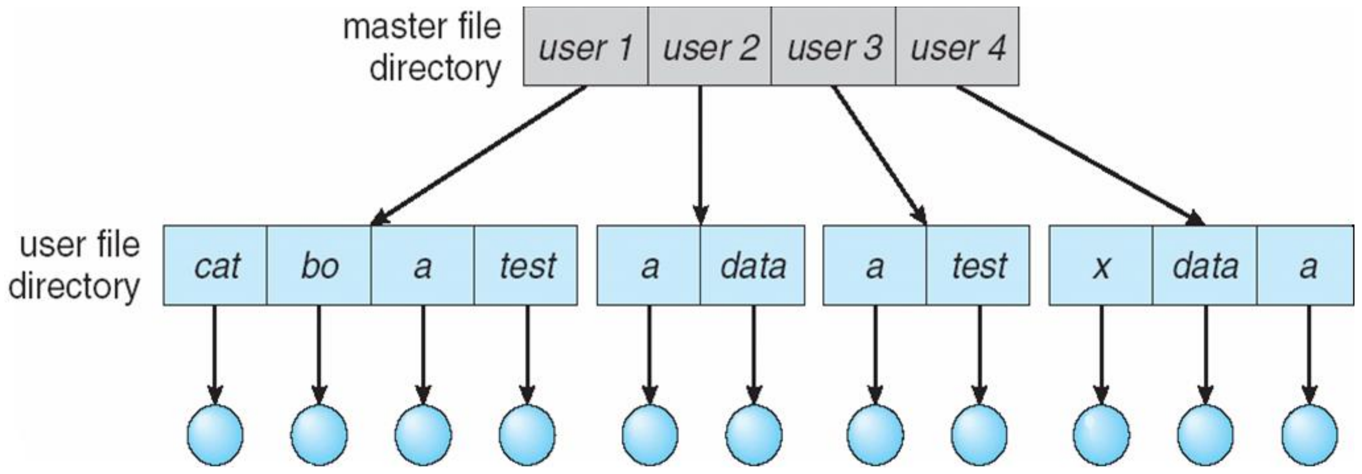
Single-Level Directory

- A single directory for all users
- Naming problem
- Grouping problem

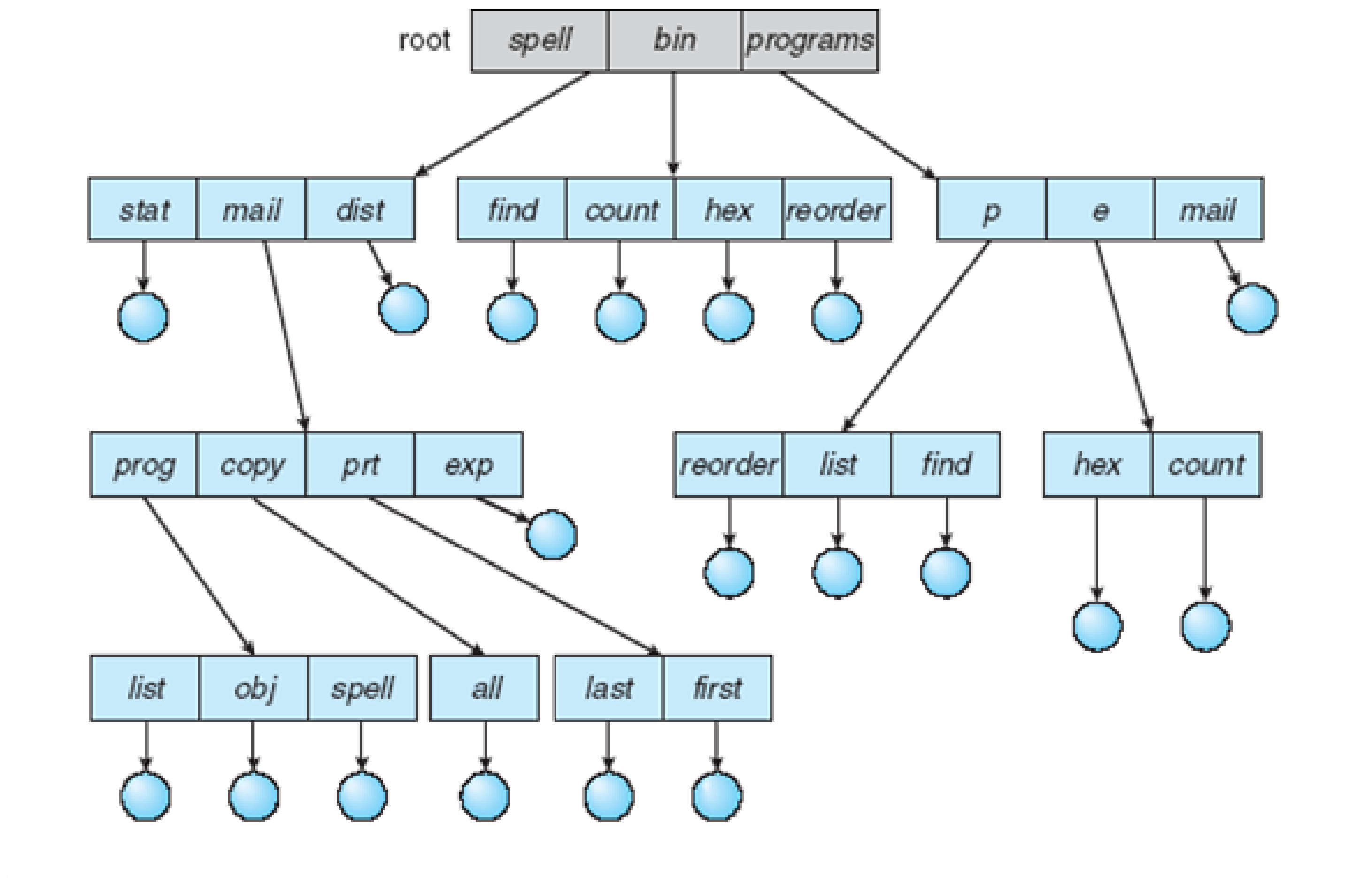


Two-Level Directory

- ❖ Separate directory for each user
- ❖ Path name
- ❖ Can have the same file name for different user

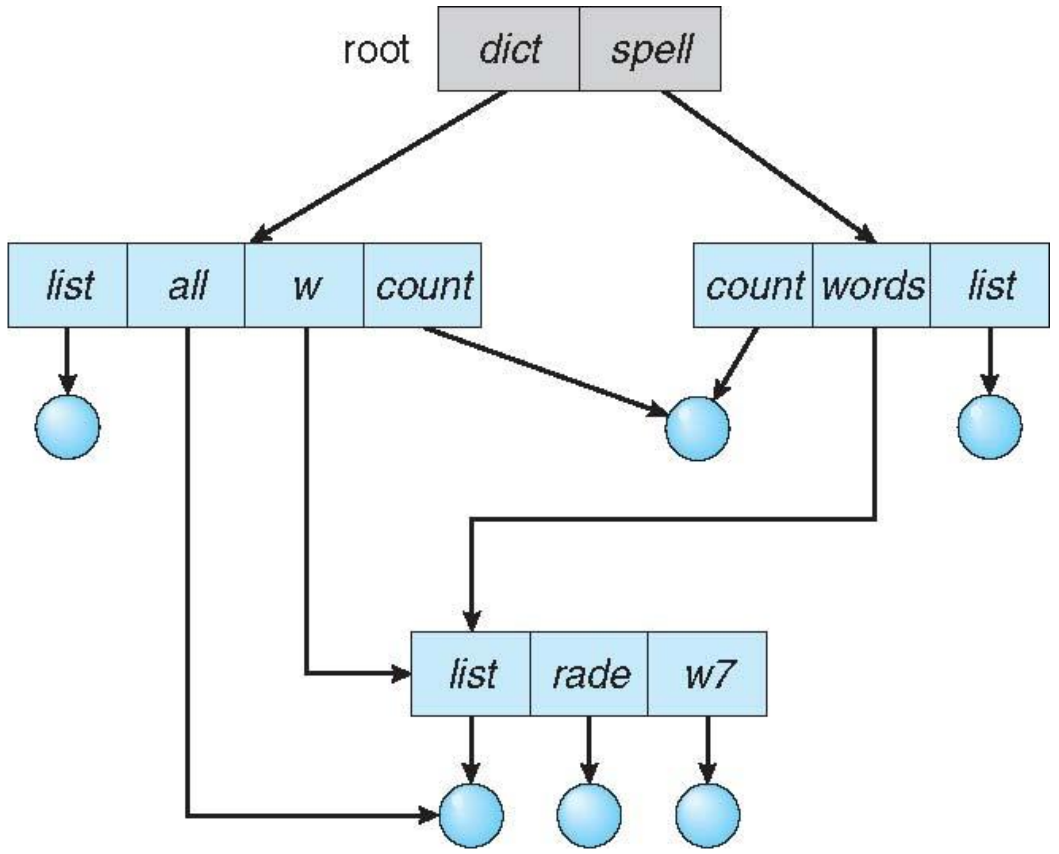


Tree-Structured Directories

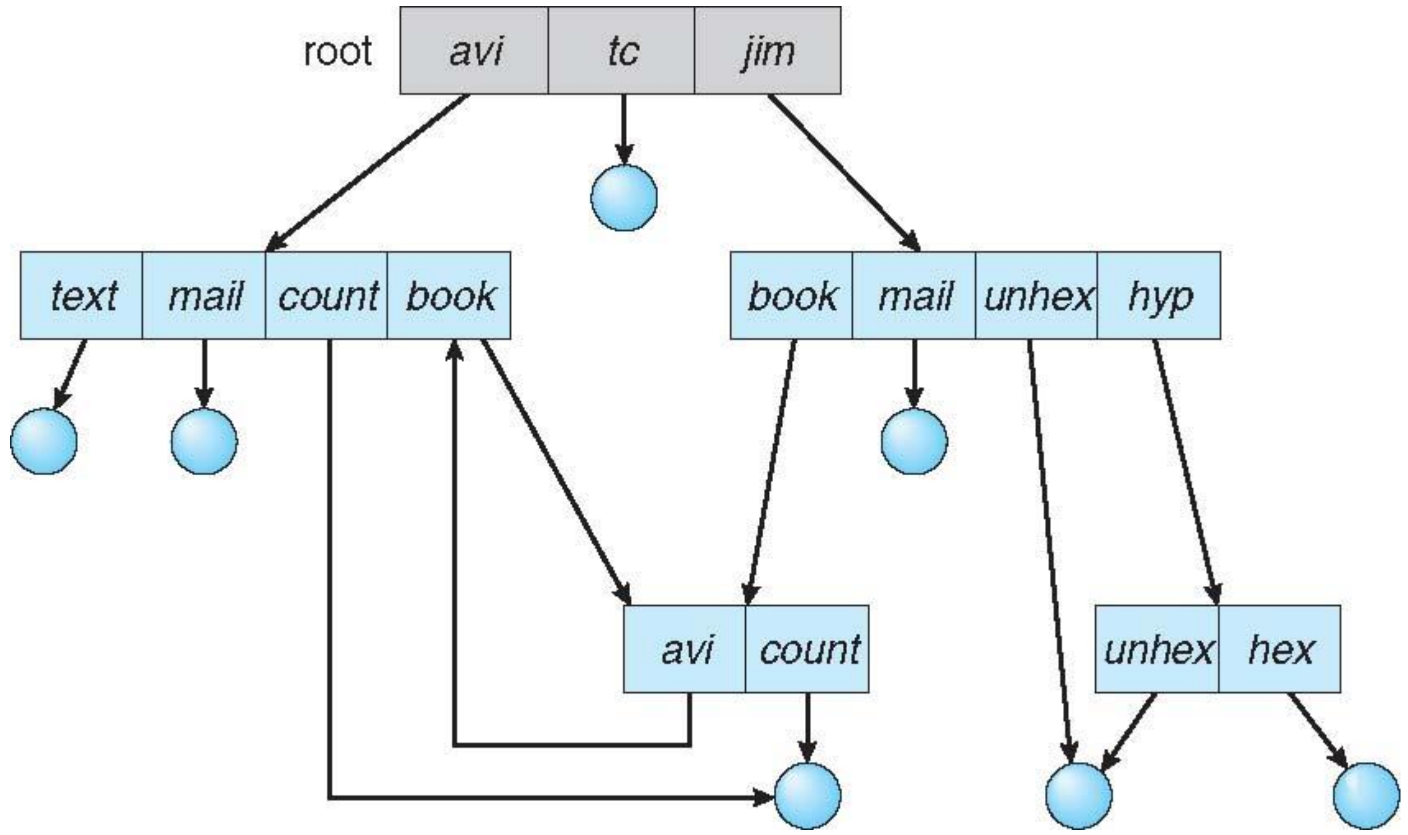


Acyclic-Graph Directories

- Have shared subdirectories and files



General Graph Directory



Thank You