File Systems (1)

Dr. Jun Zheng

CSE325 Principles of Operating Systems

11/15/2019

NEW MEXICO TECH
SCIENCE · ENGINEERING · RESEARCH · UNIVERSITY

File Systems

- Most visible aspect of an OS
- Implement an abstraction (files) for secondary storage
  - A file is a named collection of related information that is recorded on secondary storage.
  - Data can NOT be written to secondary storage unless they are within a file.
- Organize files logically (directories)
- Permit sharing of data between processes, users, and machines
- Protect data from unwanted access (security)

File Structure

- A file has a certain defined **structure** which depends on its types:
  - A text file is a sequence of characters organized into lines.
  - A source file is a sequence of subroutines and function.
  - An object file is a sequence of bytes organized into blocks understandable by the system's linker.
  - An executable file is a series of code sections that the loader can bring into memory and execute.

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

- **Create**
- **Write –** at **write pointer** location
- **Read –** at **read pointer** location
- **Reposition within file - 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

Access Methods

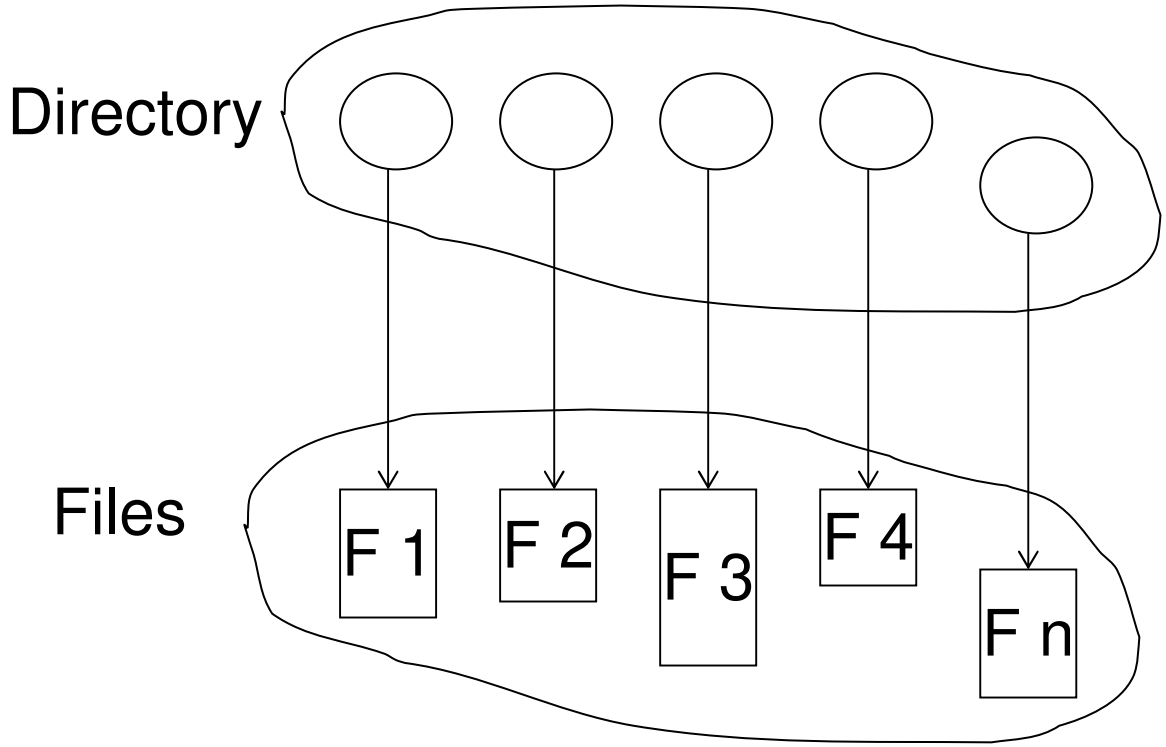
- Sequential access
  - Read all bytes/records from the beginning
  - Cannot jump around
    - May rewind or back up, however
  - Convenient when medium was magnetic tape
  - Often useful when whole file is needed
- Random access
  - Bytes (or records) read in any order
  - Essential for database systems
  - Read can be …
    - Move file marker (seek), then read or …
    - Read and then move file marker

Directories

- Naming is nice, but limited
- Humans like to group things together for convenience
- File systems allow this to be done with *directories*

Directory Structure

A collection of nodes containing information about all files



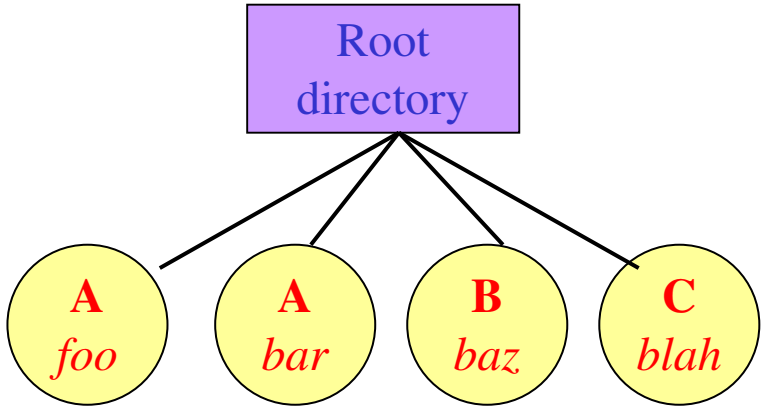
Both the directory structure and the files reside on disk

Directory Organization

The directory is organized logically to obtain

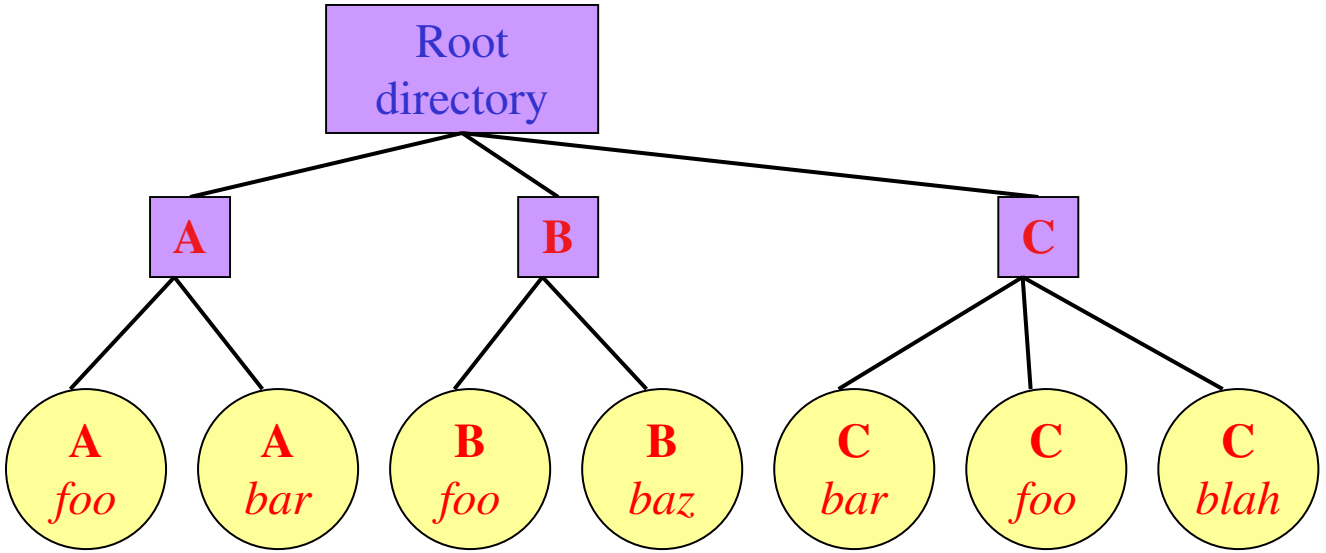
- Efficiency – locating a file quickly
- Naming – convenient to users
  - Two users can have same name for different files
  - The same file can have several different names
- Grouping – logical grouping of files by properties, (e.g., all Java programs, all games, ...)

Single-Level Directory



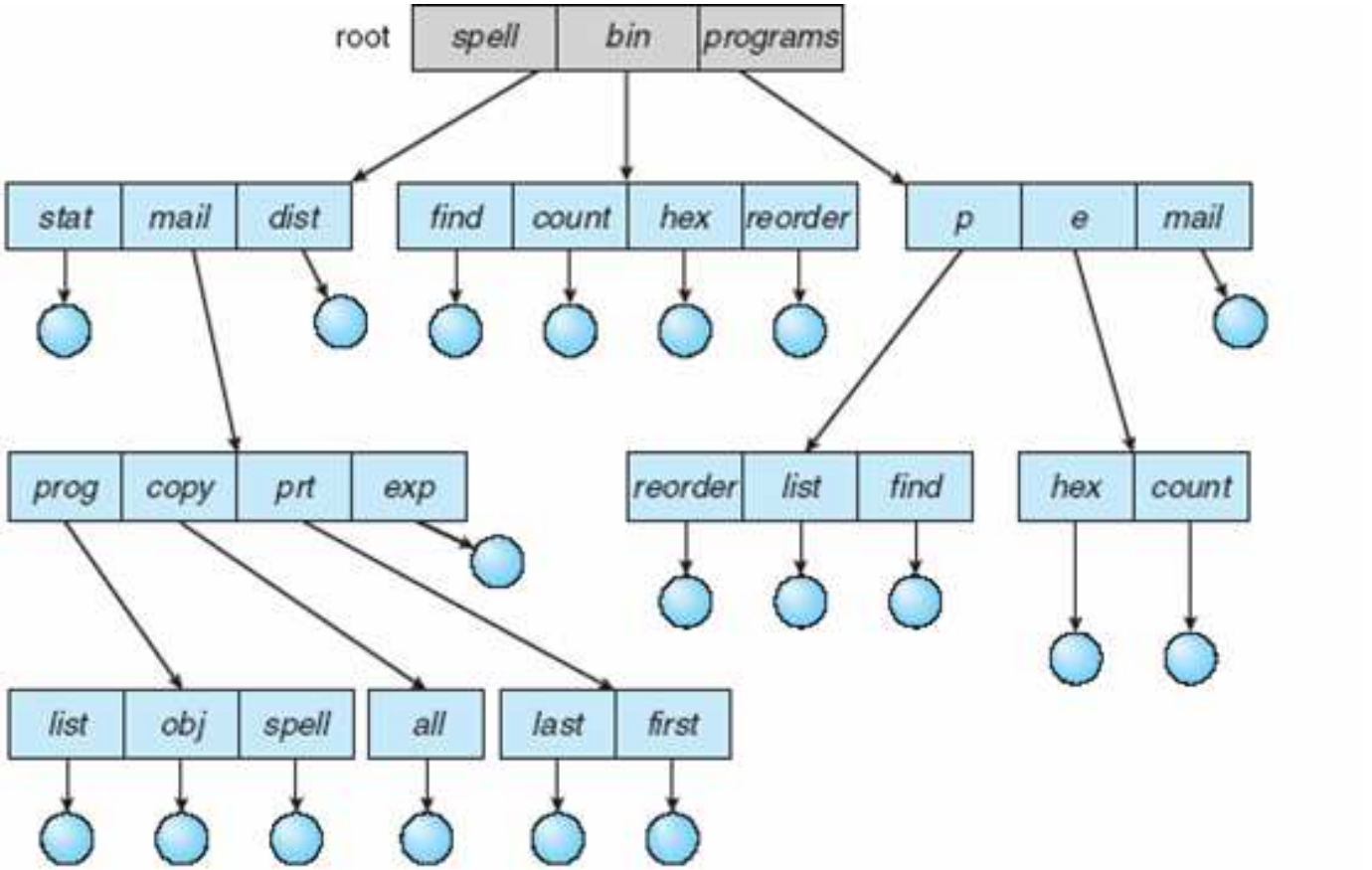
- One directory in the file system
- Example directory
  - Contains 4 files (*foo*, *bar*, *baz*, *blah*)
  - owned by 3 different people: A, B, and C (owners shown in red)
- Problem: what if user B wants to create a file called *foo*?
- Naming problem, grouping problem

Two-Level Directory



- Solves naming problem: each user has her own directory
- Multiple users can use the same file name
- By default, users access files in their own directories
- Extension: allow users to access files in others' directories
- No grouping capability

Tree-Structured Directories



Tree-Structured Directories

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

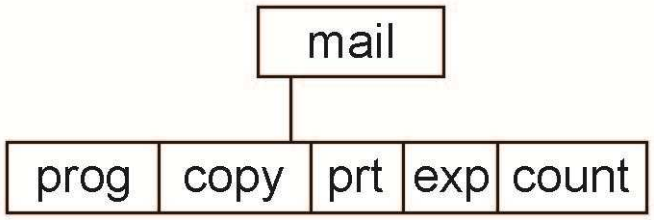
```
rm <file-name>
```

- 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"

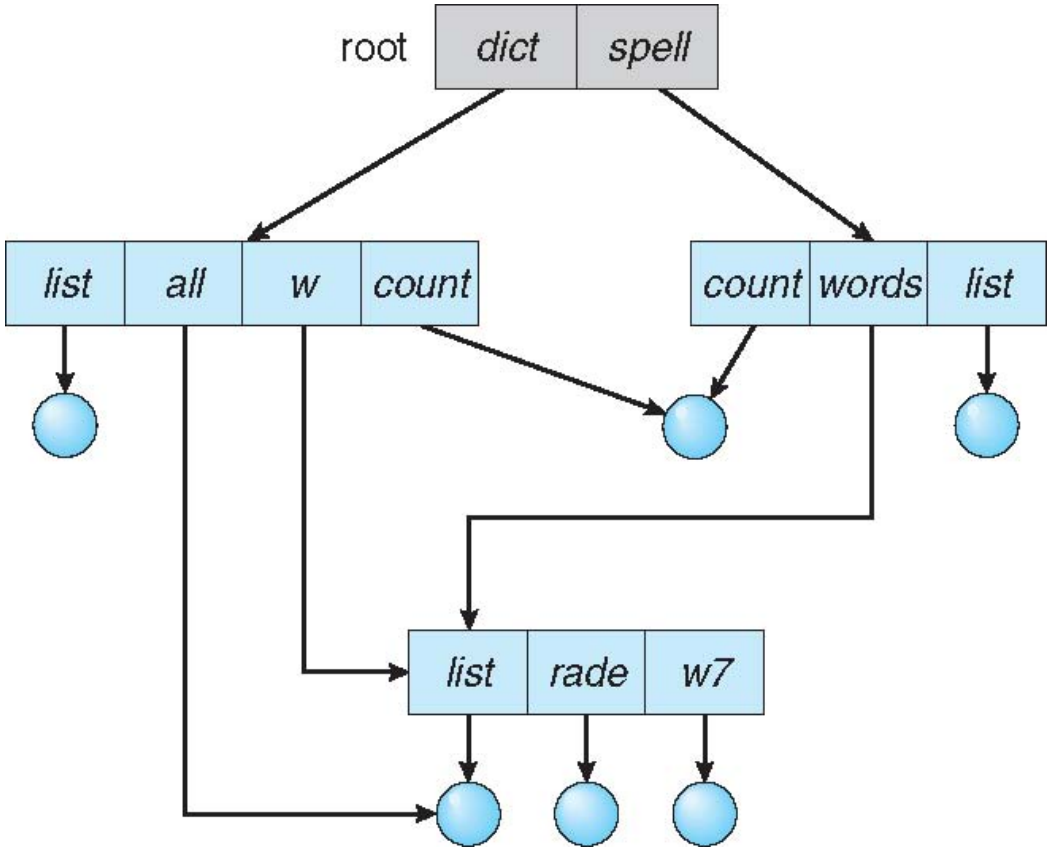
NEW MEXICO TECH
SCIENCE · ENGINEERING · RESEARCH · UNIVERSITY

Tree-Structured Directories

- Efficient searching
- Grouping Capability
- Current directory (working directory)

Acyclic-Graph Directories

Have shared subdirectories and files



Acyclic-Graph Directories

- Two different names (aliasing)
- New directory entry type
  - **Link** – another name (pointer) to an existing file
  - **Resolve the link** – follow pointer to locate the file
- If ***dict*** deletes ***count*** ⇒ dangling pointer
  - How to solve this problem?

Solutions to Dangling Pointer Problem

- Backpointers, so we can delete all pointers
  Variable size records a problem
- Entry-hold-count solution