

File System Interface

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

2

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

3

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

4

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
 - Locking (mandatory, advisory)

5

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

6

Access Methods

- Sequential Access

read next
write next
reset
no read after last write
(rewrite)

- Direct Access

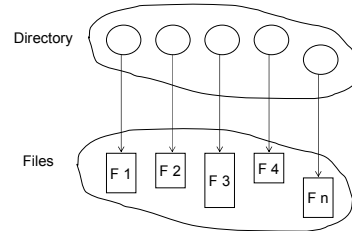
read n
write n
position to n
read next
write next
rewrite n

n = relative block number

7

Directory Structure

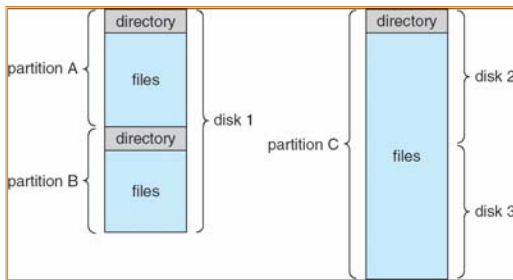
- A collection of nodes containing information about all files



Both the directory structure and the files reside on disk
Backups of these two structures are kept on tapes

8

A Typical File-system Organization



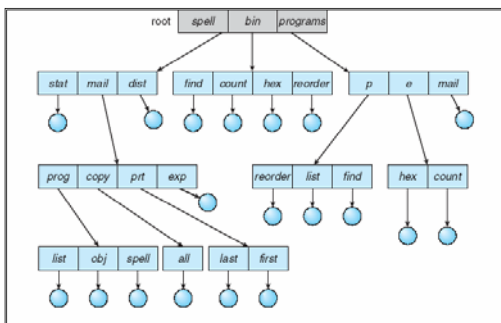
9

Operations Performed on Directory

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

10

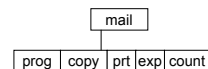
Tree-Structured Directories



11

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`

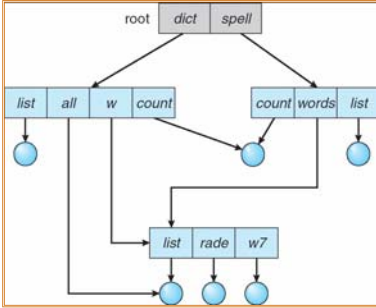


Deleting "mail" \Rightarrow deleting the entire subtree rooted by "mail"

12

Acyclic-Graph Directories

- Have shared subdirectories and files



13

Acyclic-Graph Directories (Cont.)

- Two different names (aliasing)
 - If *dict* deletes *count* \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

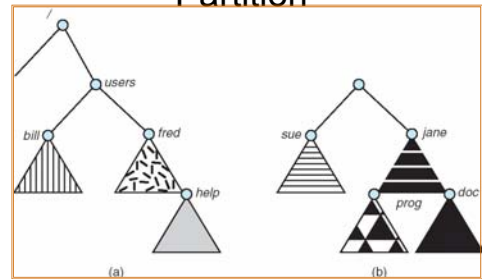
14

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**

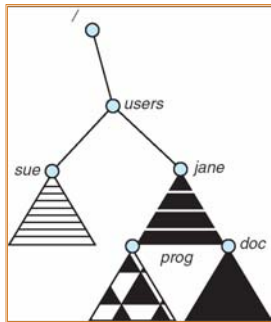
15

(a) Existing. (b) Unmounted Partition



16

Mount Point



17

Protection

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

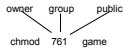
18

Access Lists and Groups

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

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

