MCA 32
OPERATING SYSTEMS
UNIT : #V
FILE SYSTEM

Khalid Raza
DEPARTMENT OF COMPUTER SCIENCE
Jamia Millia Islamia (Central University), New Delhi, India.
✉ kraza@jmi.ac.in | Web site: www.kraza.in

CONTENTS

- 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

BACKGROUND

- **File system** - the most visible aspect of an OS.
- Provides mechanism for on-line storage and access to both data and programs.
- Consists of two distinct parts: a collection of *files*, each storing related data, and a *directory structure*, which organizes and provides information about all the files.
- File systems live on devices, which is the main objective of current unit.

FILE CONCEPT

- Computers store information on various storage media,
  - Magnetic disks, magnetic tapes, and optical disks.
- For convenient to be used, OS provides a uniform logical view of information storage.
- The OS abstracts from the physical properties of its storage devices to define a logical storage unit, the *file*.
- Files are mapped by the OS onto physical devices.
- *A file is a named collection of related information recorded on secondary storage.*
- From a user's perspective, *a file is the smallest allotment of logical secondary storage*; i.e., data cannot be written to secondary storage unless they are within a file.
- Files may be free form, such as text files, or may be formatted rigidly.
- In general, a file is a sequence of bits, bytes, lines, or records. The concept of a file is extremely general.

…FILE CONCEPT

- Many different types of information may be stored in a file
  - source programs, object programs, executable programs, numeric data, text, payroll records, graphic images, sound recordings, and so on.
- **File Attributes:** *A* file's attributes vary from one OS to another but typically consist of :
  - **Name:** Symbolic file name in human readable form
  - **Identifier:** Unique tag, usually a number, identifies the file within the file system.
  - **Type:** Needed for systems that support different types of files.
  - **Location:** A pointer to location of file on that device.
  - **Size:** Current size of the file (in bytes, words, or blocks)
  - **Protection:** Access-control (read, write, executing, etc.)
  - **Time, date, and user identification:** Kept for creation, last modification, and last use.

…File Concept (File Operations)

- A file is an abstract data type.
- To define a file properly, we need to consider the operations that can be performed on files.
- The OS can provide system calls to create, write, read, reposition, delete, and truncate files.
- *Six basic file operations*:
  - Creating a file.
  - Writing a file.
  - Reading a file.
  - Repositioning within a file.
  - Deleting a file.
  - Truncating a file.

7

…File Concept (File Types)

- If an OS recognizes the type of a file, it can then operate on the file in reasonable ways.
  - E.g., a common mistake occurs when a user tries to print binary-object that produces garbage.
- A common way for implementing file types is to include the type as part of the file name.
- *The name is split into two parts*:
  - File name and
  - File *extension,*
- E.g., *resume.doc, Server.java,* and *ReaderThread.*c
- The system uses the extension to indicate the type of the file and the type of operations that can be performed on that file.
- E.g., only a file with a *.com, .exe,* or *.bat* extension can be *executed.*

8

…File Concept (File Types)

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

9

File Structure

- File types also used to indicate the internal structure of the file.
  - E.g. source and object files have structures that match the expectations of the programs that read them.
- Certain files must conform to a required structure that is understood by the OS.
  - E.g., OS requires that an executable file have a specific structure so that - where in memory to load and location of first instruction can be determined.
- Some OS extend this idea into a set of system-supported file structures.

10

…File Structure

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

11

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

12

Open File Locking

- Provided by some OS and file systems
- 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

13

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

14

Access Methods

- **Sequential Access**

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

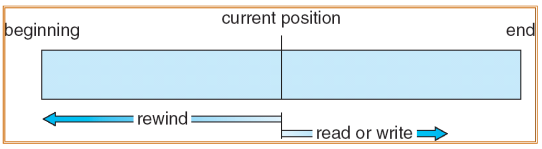
- **Direct Access**

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

n = relative block number

15

Sequential-access File



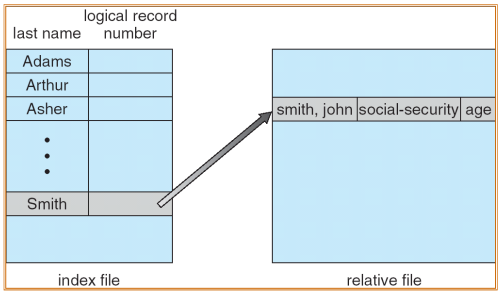
16

Simulation of Sequential Access on a Direct-access File

sequential access	implementation for direct access
reset	*cp* = 0;
read next	*read cp*; *cp* = *cp* + 1;
write next	*write cp*; *cp* = *cp* + 1;

17

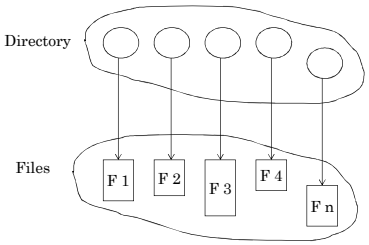
Example of Index and Relative Files



18

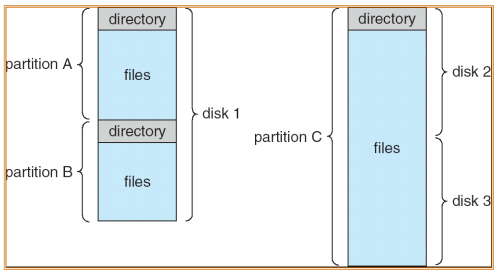
Directory Structure

- 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

A Typical File-system Organization



Operations Performed on Directory

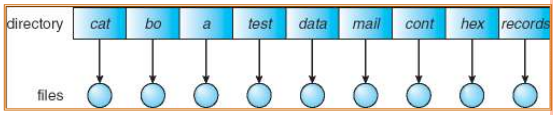
- Search for a file
- Create a file
- Delete a file
- List a directory
- Rename a file
- Traverse the file system

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

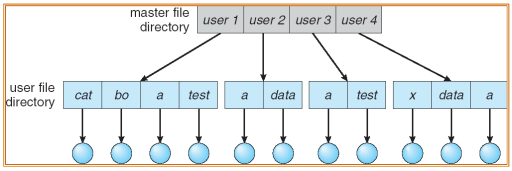
- A single directory for all users



- Naming problem
  Grouping problem

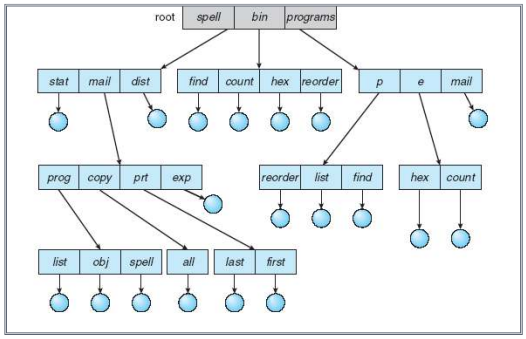
Two-Level Directory

- 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

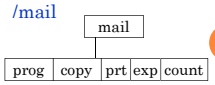
- Efficient searching
- Grouping Capability
- Current directory (working directory)
  - cd /spell/mail/prog
  - type list
- **Absolute** or **relative** path name
- Creating a new file is done in current directory
- Delete a file

  rm <file-name>
- Creating a new subdirectory is done in current directory

  mkdir <dir-name>

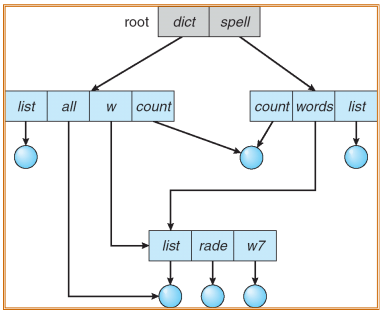
  Example: if in current directory /mail

  mkdir count



ACYCLIC-GRAPH DIRECTORIES

- Have shared subdirectories and files

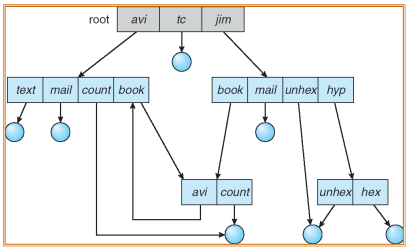


…ACYCLIC-GRAPH DIRECTORIES

- Two different names (aliasing)
- If *dict* deletes *list* ⇒ 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

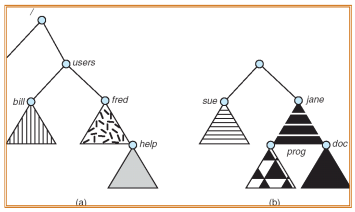
GENERAL GRAPH DIRECTORY



- 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

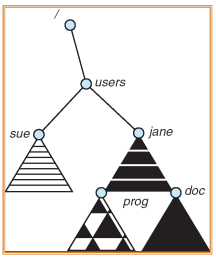
FILE SYSTEM MOUNTING

- 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

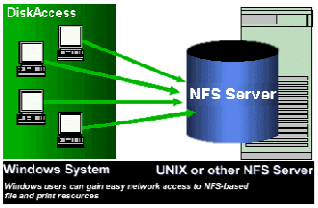
…File System Mounting



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

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**
  - Semi automatically via the **world wide web**
- **Client-server** model allows clients to 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 operating system file calls are translated into remote calls
- Distributed Information Systems **(distributed naming services)** such as LDAP, DNS, NIS, Active Directory implement unified access to information needed for remote computing

File Sharing – Failure Modes

- Remote file systems add new failure modes, due to network failure, server failure
- Recovery from failure can involve state information about status of each remote request
- Stateless protocols such as NFS include all information in each request, allowing easy recovery but less security

File Sharing – Consistency Semantics

- **Consistency semantics** specify how multiple users are to access a shared file simultaneously
  - Similar to process synchronization algorithms
    - Tend to be less complex due to disk I/O and network latency (for remote file systems)
  - Andrew File System (AFS) implemented complex remote file sharing semantics
  - Unix file system (UFS) implements:
    - Writes to an open file visible immediately to other users of the same open file
    - Sharing file pointer to allow multiple users to read and write concurrently
  - AFS has session semantics
    - Writes only visible to sessions starting after the file is closed

Protection

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

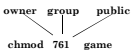
37

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	⇒	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.

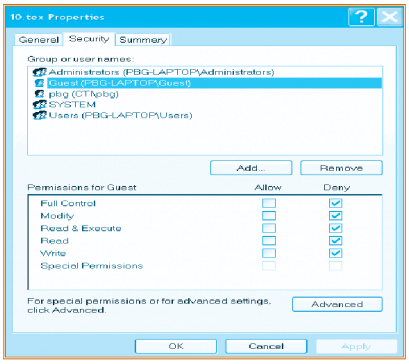


Attach a group to a file

chgrp G game

38

Windows XP Access-control List Management

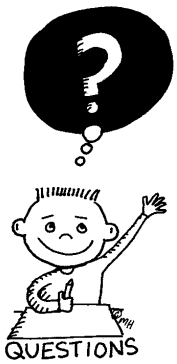


39

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/

40



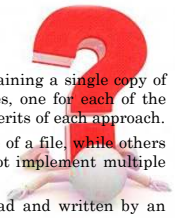
41

Conclusions

- File Concept
- Access Methods
- Directory Structure
- File-System Mounting
- File Sharing
- Protection
- File system protection in Windows & Unix

42

Self Assessment Questions



1. Some systems provide file sharing by maintaining a single copy of a file; other systems maintain several copies, one for each of the users sharing the file. Discuss the relative merits of each approach.
2. Why do some systems keep track of the type of a file, while others leave it to the user and others simply do not implement multiple file types? Which system is "better?"
3. In some systems, a subdirectory can be read and written by an authorized user, just as ordinary files can be.

   a. Describe the protection problems that could arise.

   b. Suggest a scheme for dealing with each of these protection problems.
4. Consider a system that supports 5,000 users. Suppose that you want to allow 4,990 of these users to be able to access one file.

   a. How would specify this protection scheme in UNIX?

   b. Can you suggest another protection scheme that can be used more effectively for this purpose than the scheme provided by UNIX?

References

- Abraham Silberschatz, Peter B. Galvin, Operating system concepts, 8th Ed., Wiley India.
- Andrew S. Tanenbaum, "Modern Operating System", PHI.
- Other freely available web resources.