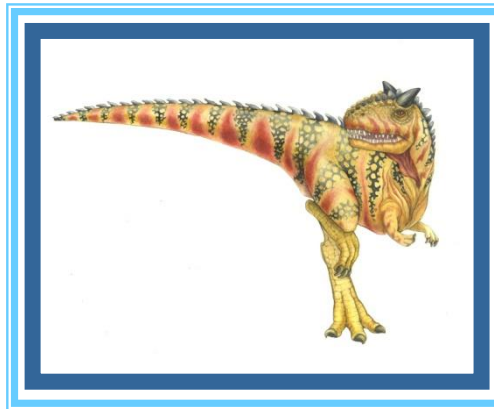


Chapter 10:

File-System Interface





File System

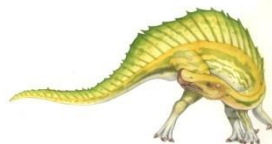
- For most users
 - File system is the most visible aspect of an operating system
- OS provides the mechanism for
 - Storage of and access to both data and programs
 - That too, for all the users of the computer system
- file system consists of two distinct parts
 - A collection of files, each storing related data
 - A directory structure, which organizes all the files in the system





File Concept

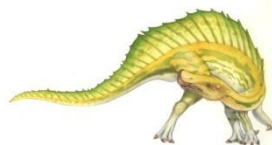
- ❑ Computers can store information on various storage media
 - ❑ Magnetic disks, magnetic tapes, and optical disks
- ❑ Operating system provides a uniform logical view
 - ❑ abstracts from the physical properties of its storage devices
- ❑ File: A named collection of related information that is recorded on secondary storage
- ❑ User's perspective: The smallest allotment of logical secondary storage
- ❑ File Type
 - ❑ Text: numeric, character, binary
 - ❑ Source: a sequence of functions with executable statements
 - ❑ Executable: Code sections that the loader can bring into memory and execute
- ❑ Information in a file is defined by its creator





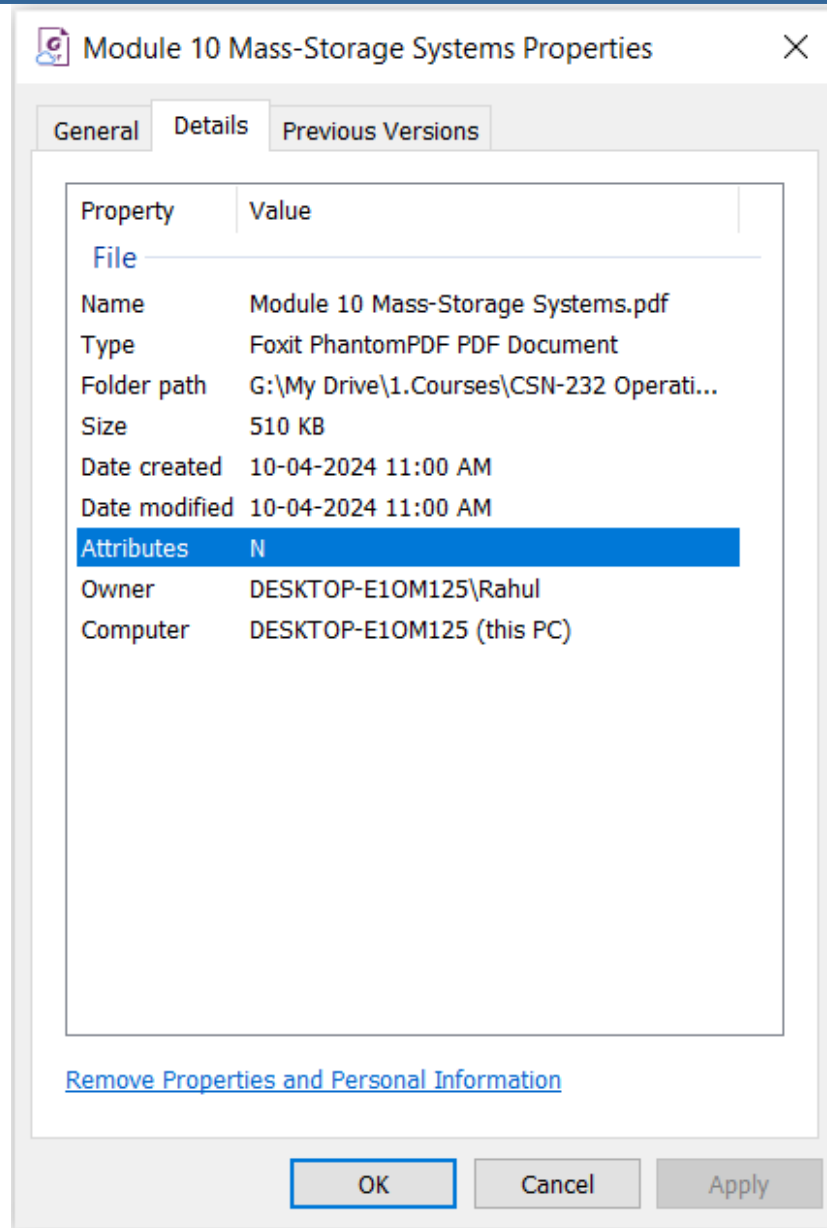
File Attributes

- ❑ **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 info Windows 10





File Operations

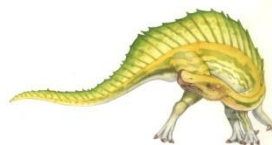
- File is an **abstract data type**
- **Create:** space in the file system, entry in the directory
- **Write** – at **write pointer** location
- **Read** – at **read pointer** location
- **Reposition within file** - **file-position pointer is repositioned**
- **Delete:** search the directory for the named file
- **Truncate:** erase the contents of a file
- **$Open(F_i)$** – 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





Open Files

- 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

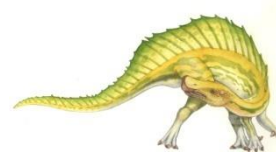
- Provided by some operating systems and file systems
 - Similar to reader-writer locks
 - **Shared lock** similar to reader lock – several processes can acquire concurrently
 - **Exclusive lock** similar to writer lock
- Mandatory or advisory:
 - **Mandatory** – access is denied depending on locks held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





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 com- pressed, for archiving or storage
multimedia	mpeg, mov, rm, mp3, avi	binary file containing audio or A/V information

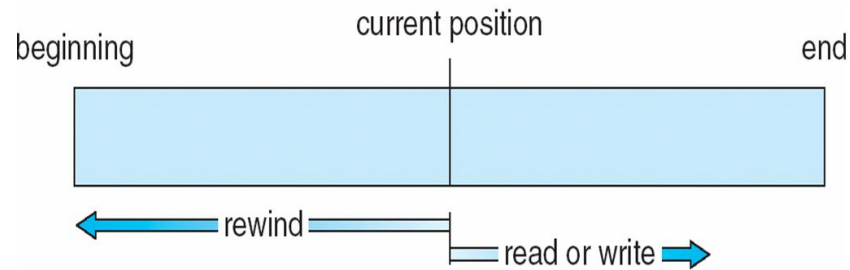




Access Methods

□ Sequential Access

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



□ Direct Access – file is fixed length **logical records**

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

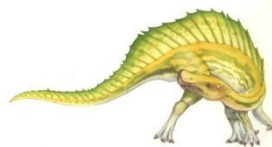
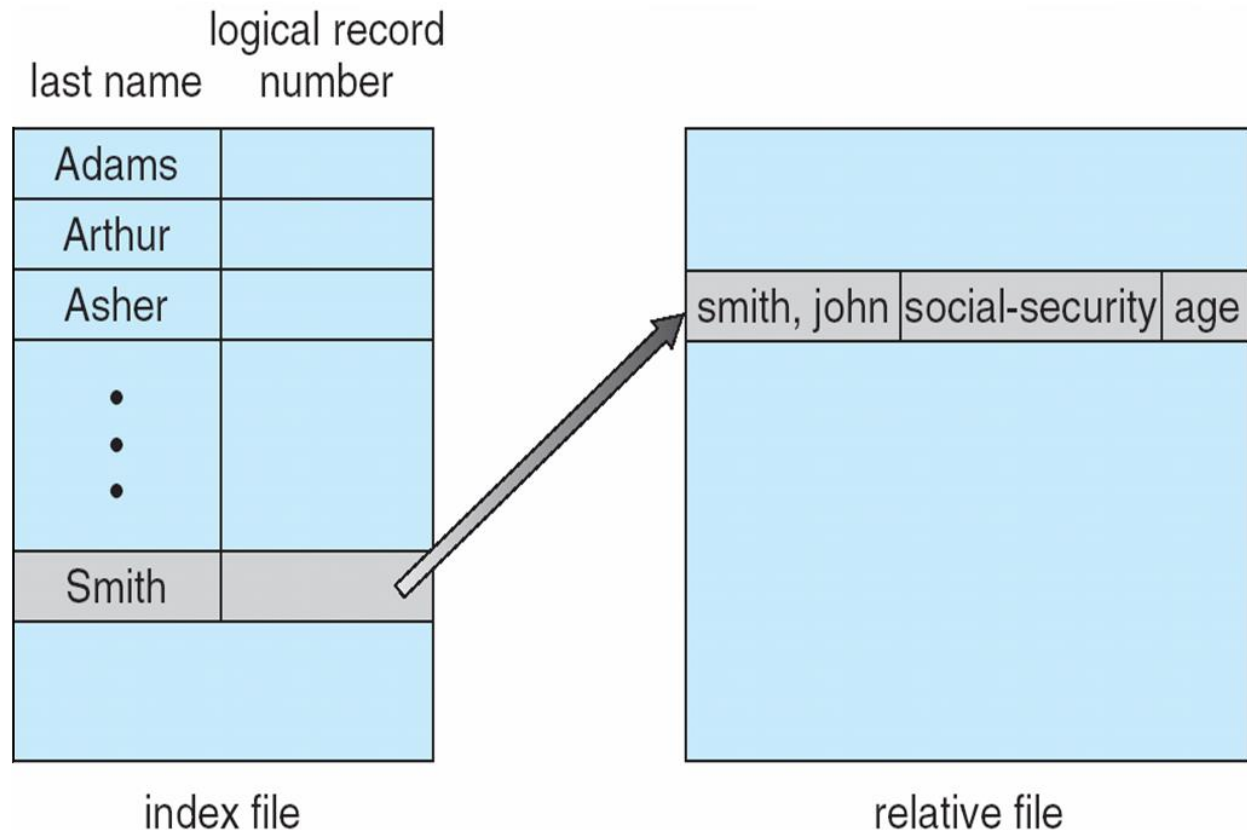
n = **relative block number**





Other Access Methods

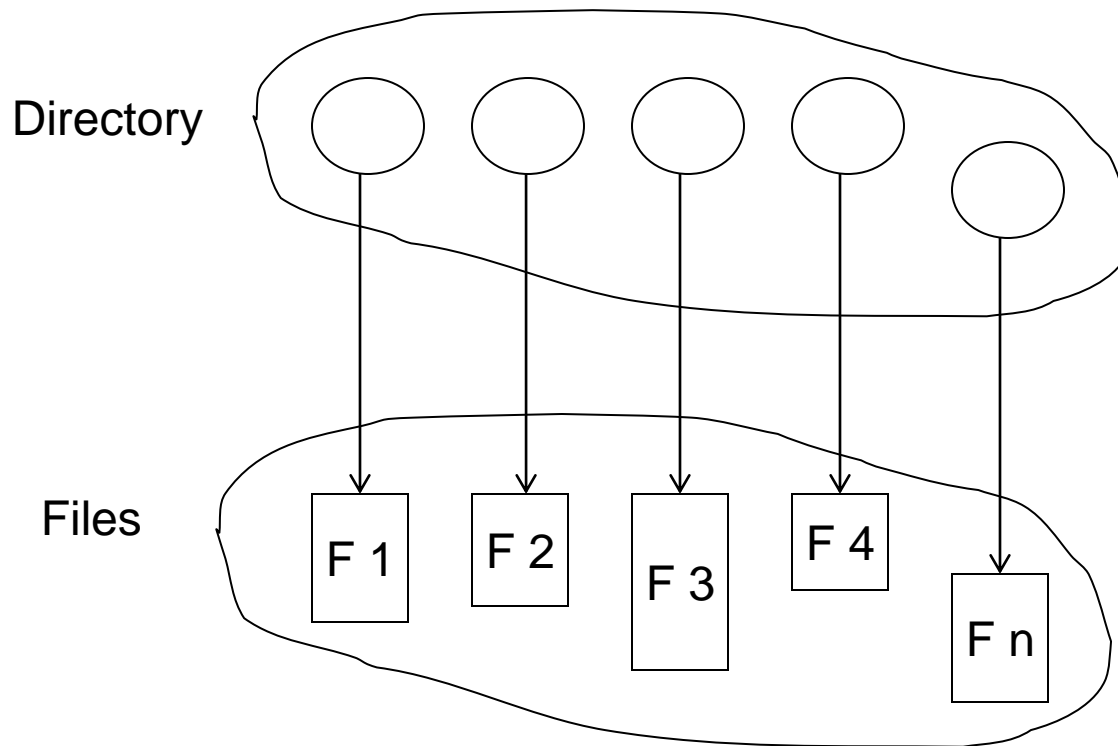
- Can be built on top of base methods
- General involve creation of an **index** for the file
- Keep index in memory for fast determination of location of data to be operated on



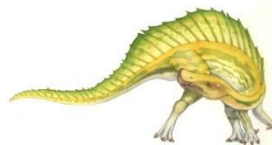


Directory Structure

- A collection of nodes containing information about all files

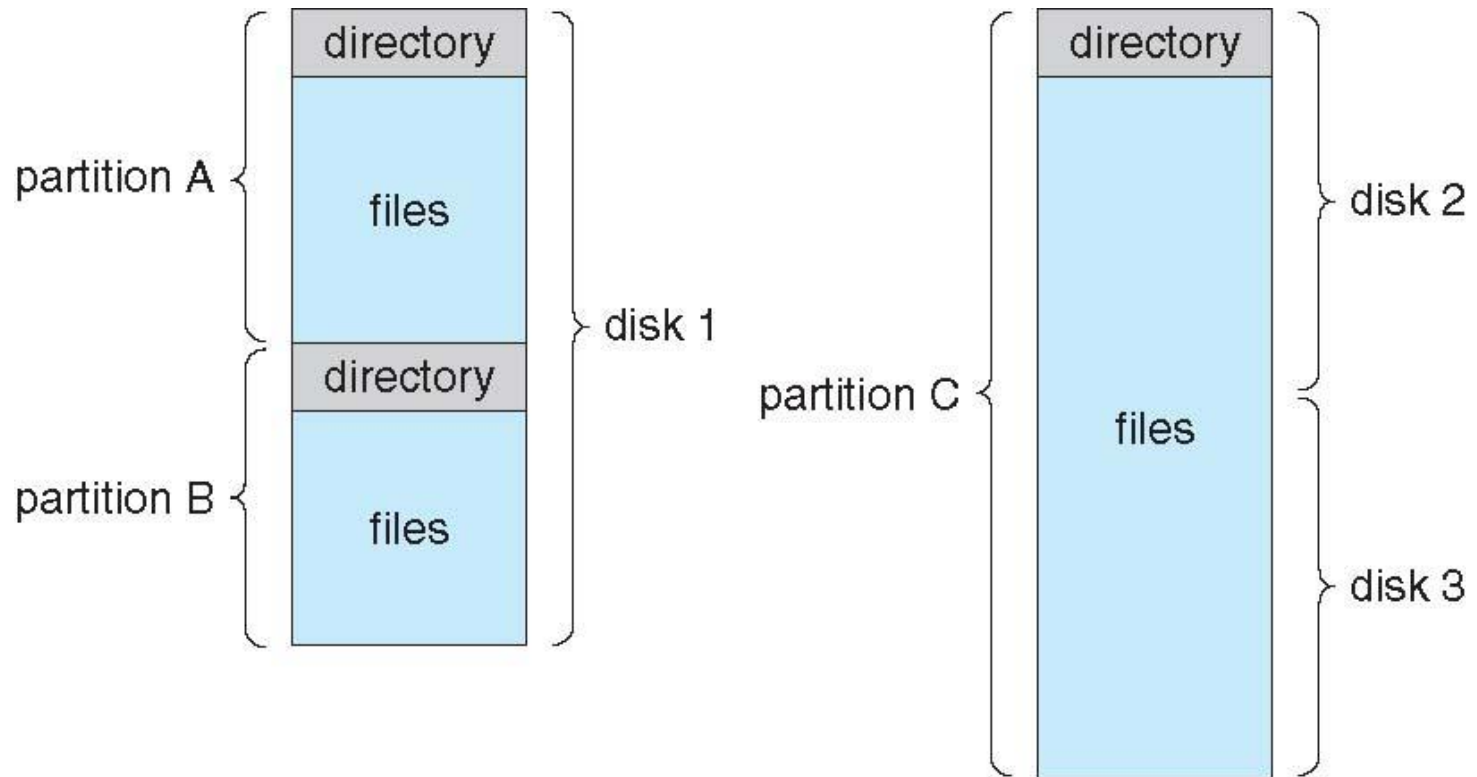


Both the directory structure and the files reside on disk





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
- Traverse the file system





Directory Organization

The directory is organized logically to obtain

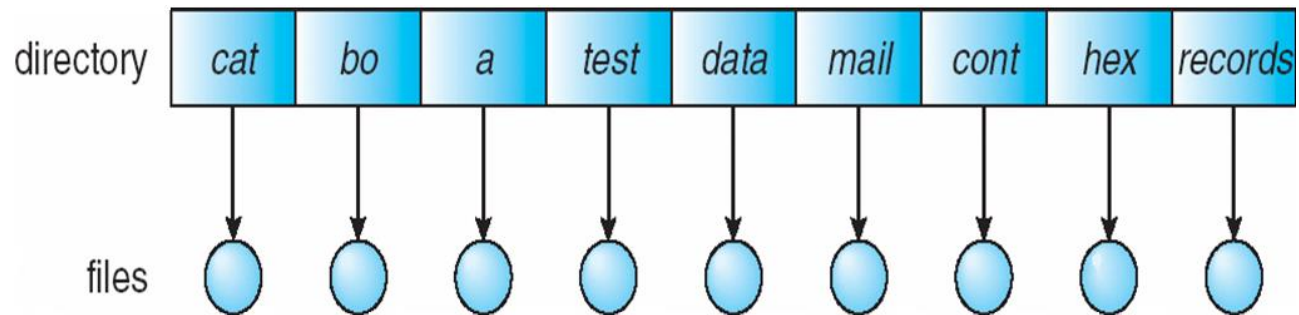
- Efficiency – locating a file quickly
- Naming – convenient to users
 - Two users can have same name for different files
 - The same file can have several different names
- Grouping – logical grouping of files by properties, (e.g., all Java programs, all games, ...)



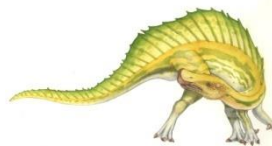


Single-Level Directory

- A single directory for all users



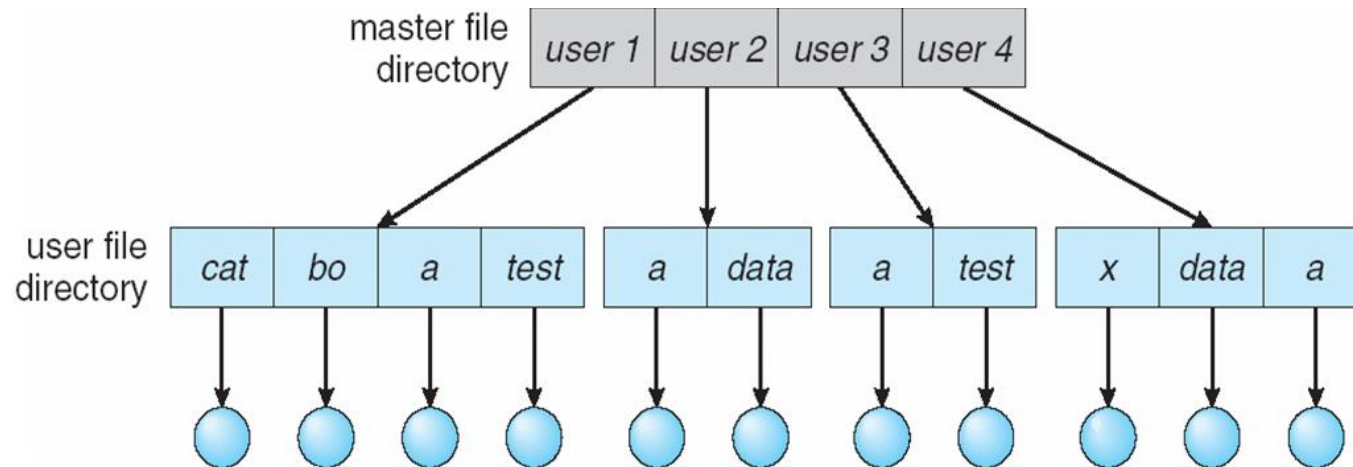
- Naming problem
- Grouping problem



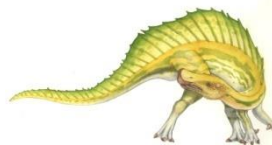


Two-Level Directory

- Separate directory for each user

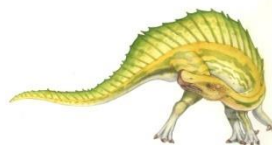
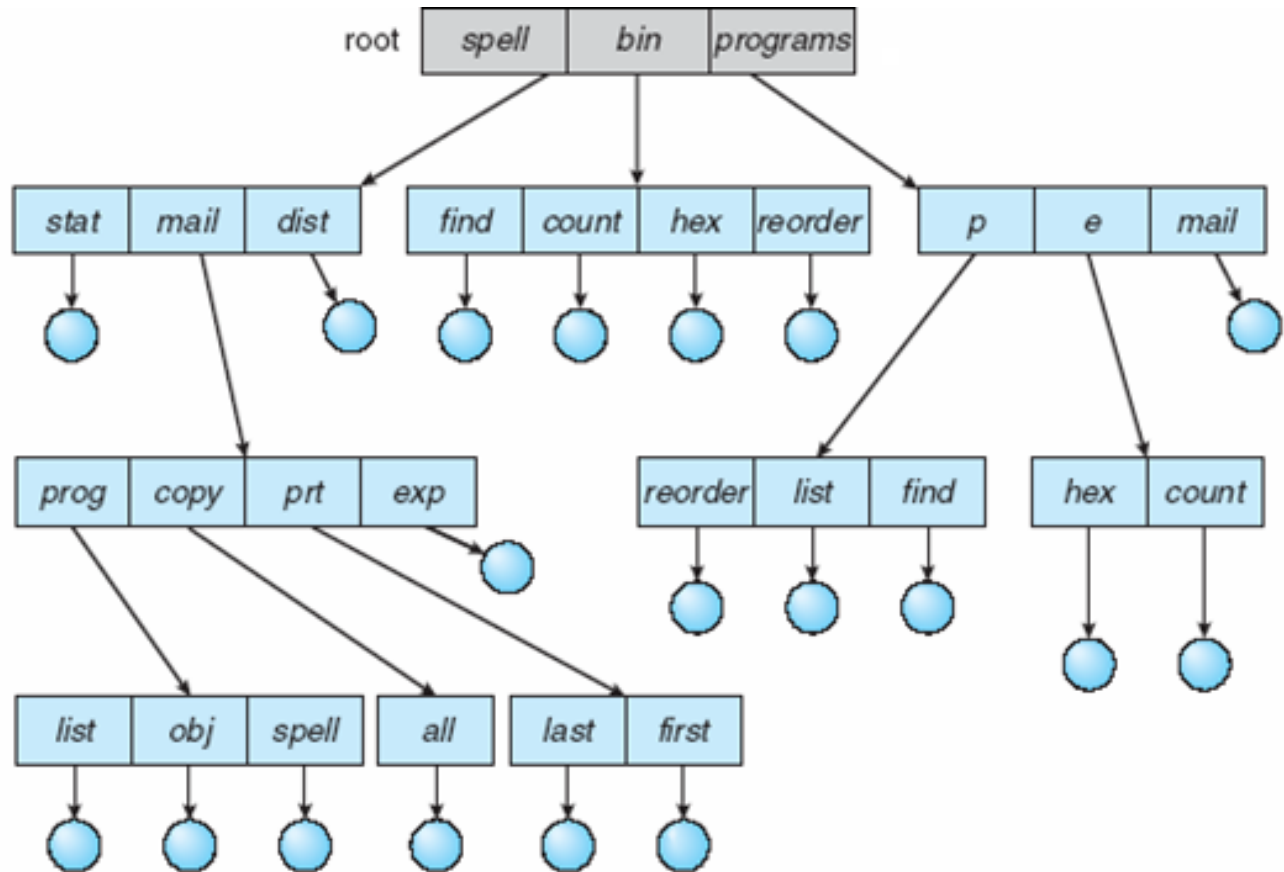


- Path name
- Can have the same file name for different user
- Efficient searching
- No grouping capability





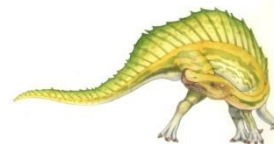
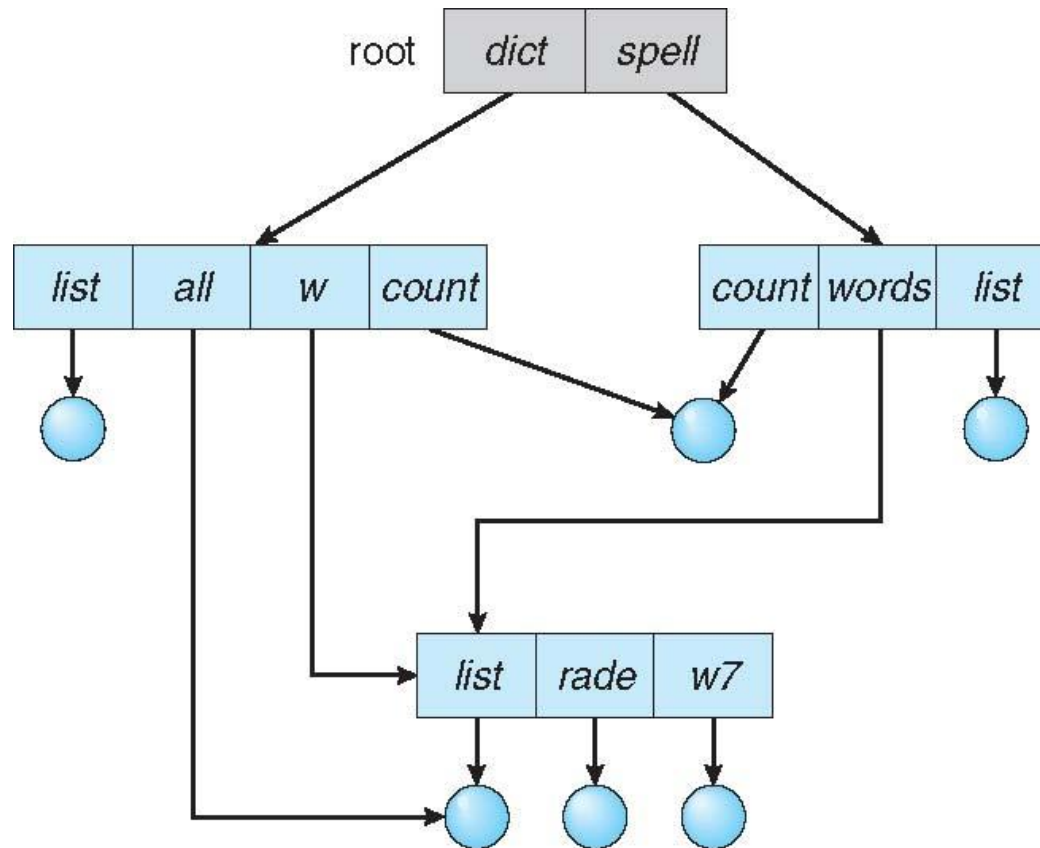
Tree-Structured Directories





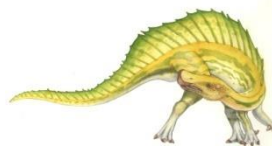
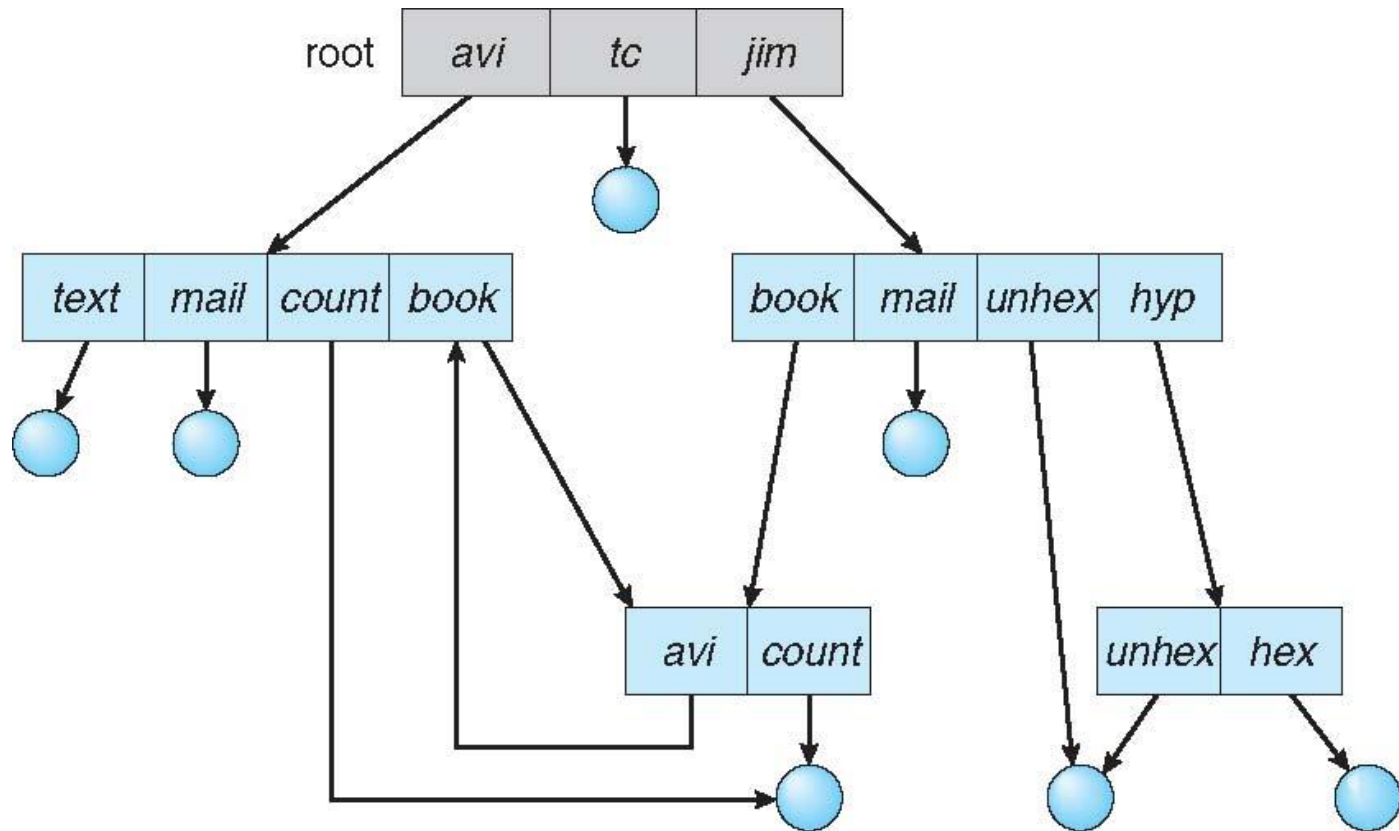
Acyclic-Graph Directories

- Have shared subdirectories and files





General Graph Directory





File Sharing

- Sharing of files on multi-user systems is desirable
- Sharing may be done through a **protection** scheme
- If multi-user system
 - **User IDs** identify users, allowing permissions and protections to be per-user
 - **Group IDs** allow users to be in groups, permitting group access rights
 - Owner of a file / directory
 - Group of a file / directory





Protection

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



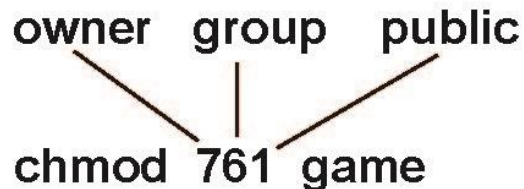


Access Lists and Groups

- Mode of access: read, write, execute
- Three classes of users on Unix / Linux

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



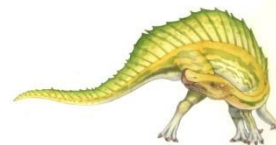
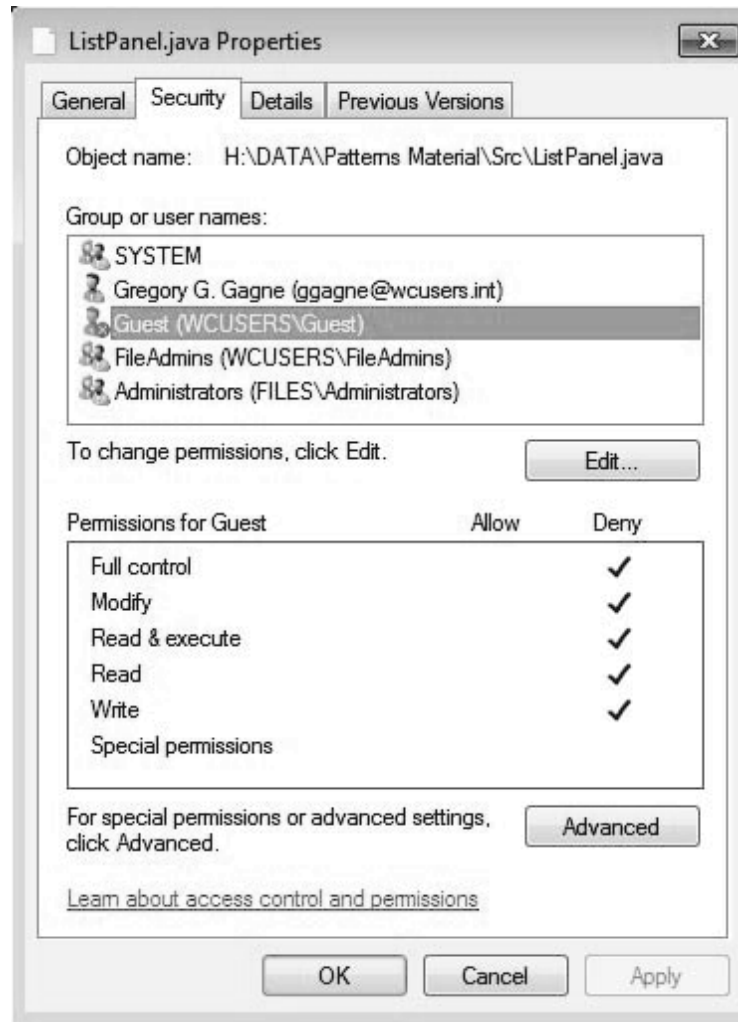
Attach a group to a file

chgrp G game





Windows 7 Access-Control List Management





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/



End of Chapter 11

