Chapter 11: File System Interface

Zhi Wang
Florida State University

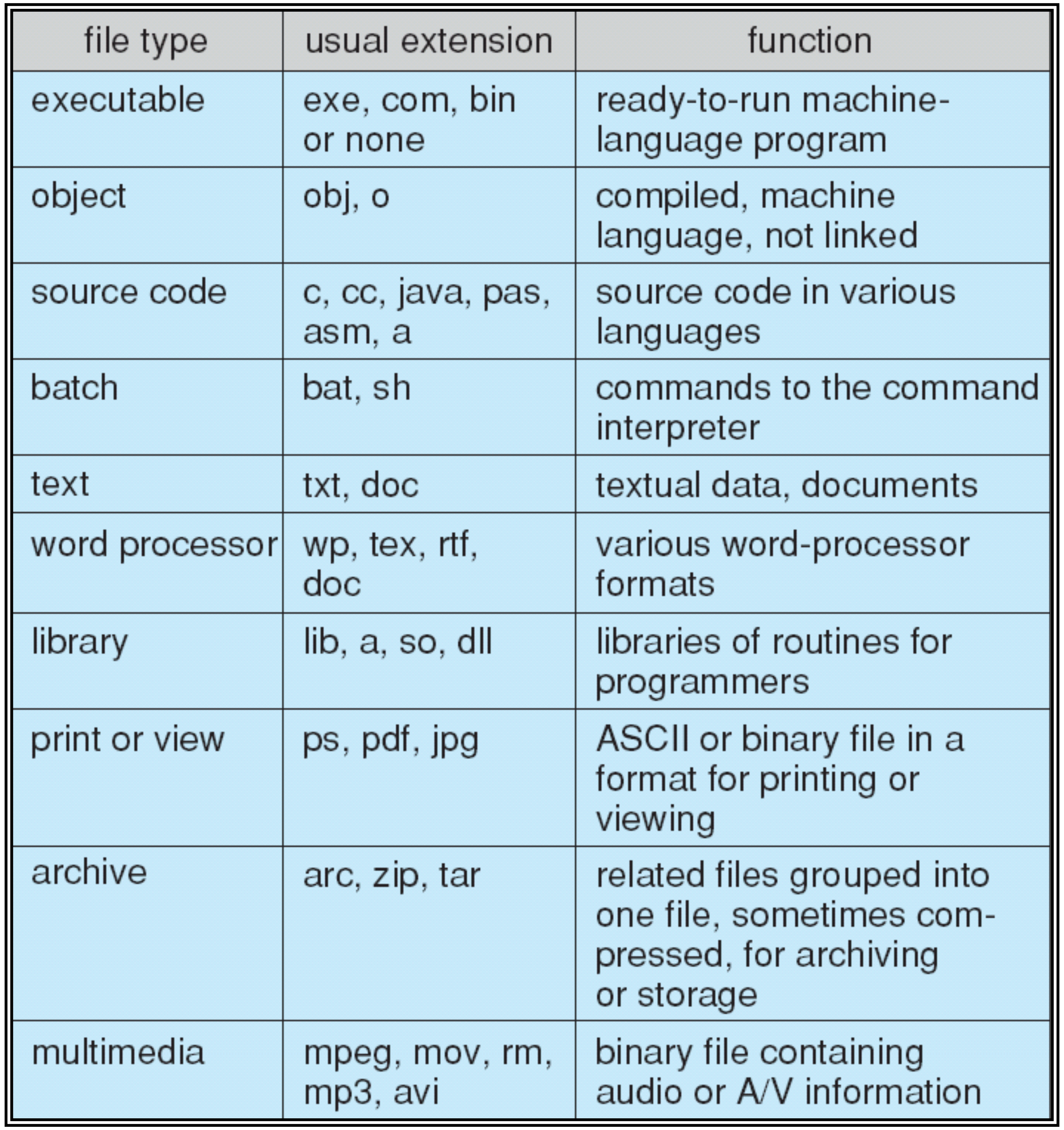
Content

- File concept
- Access methods
- Directory structure
- File-system mounting
- File sharing
- Protection

File Concept

- **File** is a contiguous logical address space for storing information
  - database, audio, video, web pages…
- There are different types of file:
  - data: numeric, character, binary
  - program

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

File Structure

- A file can have different structures, determined by OS or program
  - **no structure**: a stream of bytes or words
    - linux files
  - **simple record structure**
    - lines of records, fixed length or variable length
    - e.g., database
  - **complex structures**
    - e.g., word document, relocatable program file
  - simple and complex structure can be encoded in the first method

File Attributes

- OS keeps file **attributes** in the file **directory** structure, which is maintained on the disk
  - **name**: the name of the file
    - only information kept in human-readable form
  - **identifier**: an unique tag (number) identifies file within file system
  - **type**: the type of the file
    - needed for systems that support different types
  - **location**: pointer to file location on device
  - **size**: current file size
  - **protection**: attributes control who can do reading, writing, executing
  - **time, date, and user identification**: data for protection, security, and usage monitoring

File Operations

- OS provides file operations to
  - create, open, and close
  - read/write
  - reposition within file
  - delete
  - truncate

Open Files

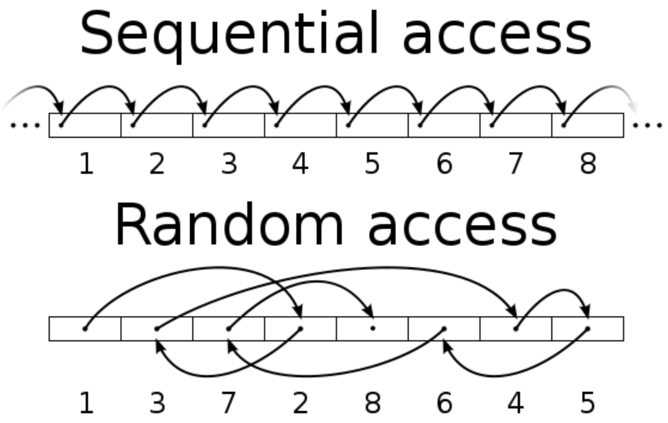
- To open a file, the OS need:
  - file position: pointer to last read/write location
    - file position is **per-process** that has the file open
  - file-open count: the number of times a file is open
    - to allow removal of data from open-file table when last processes closes it
  - disk location: cache of data access information
  - access rights: per-process access mode information
- Some file systems provide file lock to mediates access to a file
  - **mandatory lock**: access is denied depending on locks held and requested
  - **advisory lock**: processes can find status of locks and decide what to do

File Locking Example – Java API

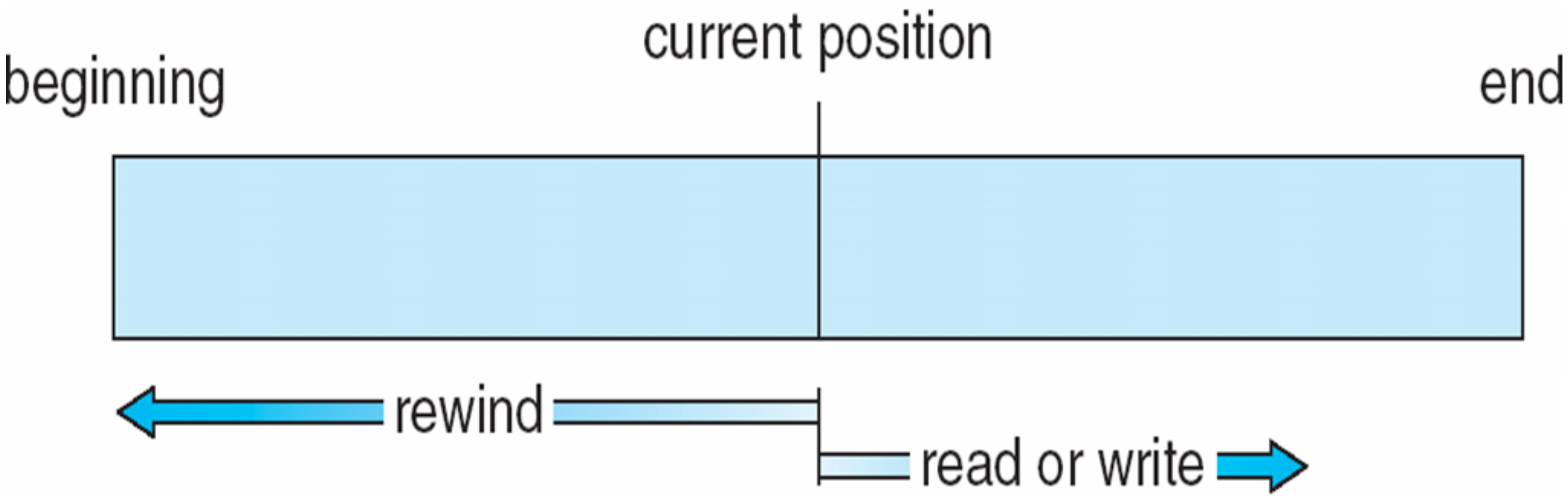
```
FileLock sharedLock = null;
FileLock exclusiveLock = null;
RandomAccessFile raf = new RandomAccessFile("file.txt", "rw");

// get the channel for the file
FileChannel ch = raf.getChannel();

// this locks the first half of the file - exclusive
exclusiveLock = ch.lock(0, raf.length()/2, true);
/** Now modify the data . . . */
exclusiveLock.release();

// this locks the second half of the file - shared
sharedLock = ch.lock(raf.length()/2+1, raf.length(), false);
/** Now read the data . . . */
sharedLock.release();
```

Access Methods

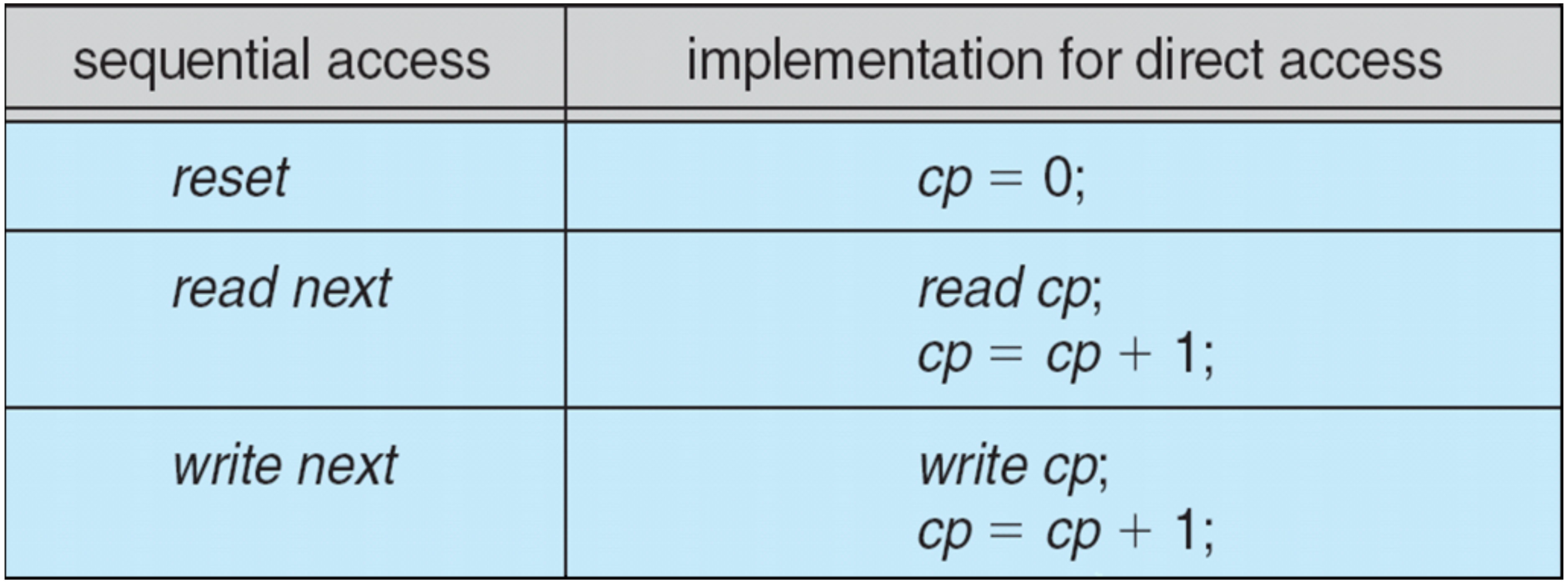
- Sequential access
  - a group of elements is access **in a predetermined order**
  - for some media types, the only access mode (e.g., tape)
- Direct access
  - access an element at an **arbitrary position** in a sequence in (roughly) **equal time**, independent of sequence size
    - it is possible to emulate random access in a tape, but access time varies
  - sometime called random access



Sequential-access File



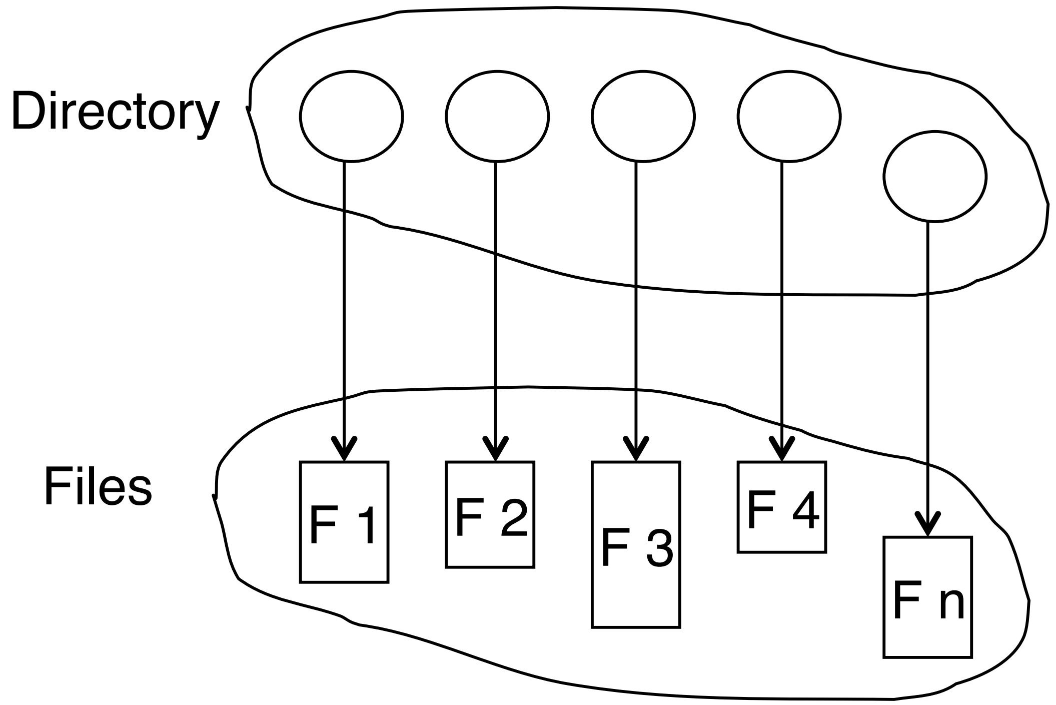
Sequential Access on Direct-access File



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

Directory Structure

- Directory is a collection of nodes containing information about all files

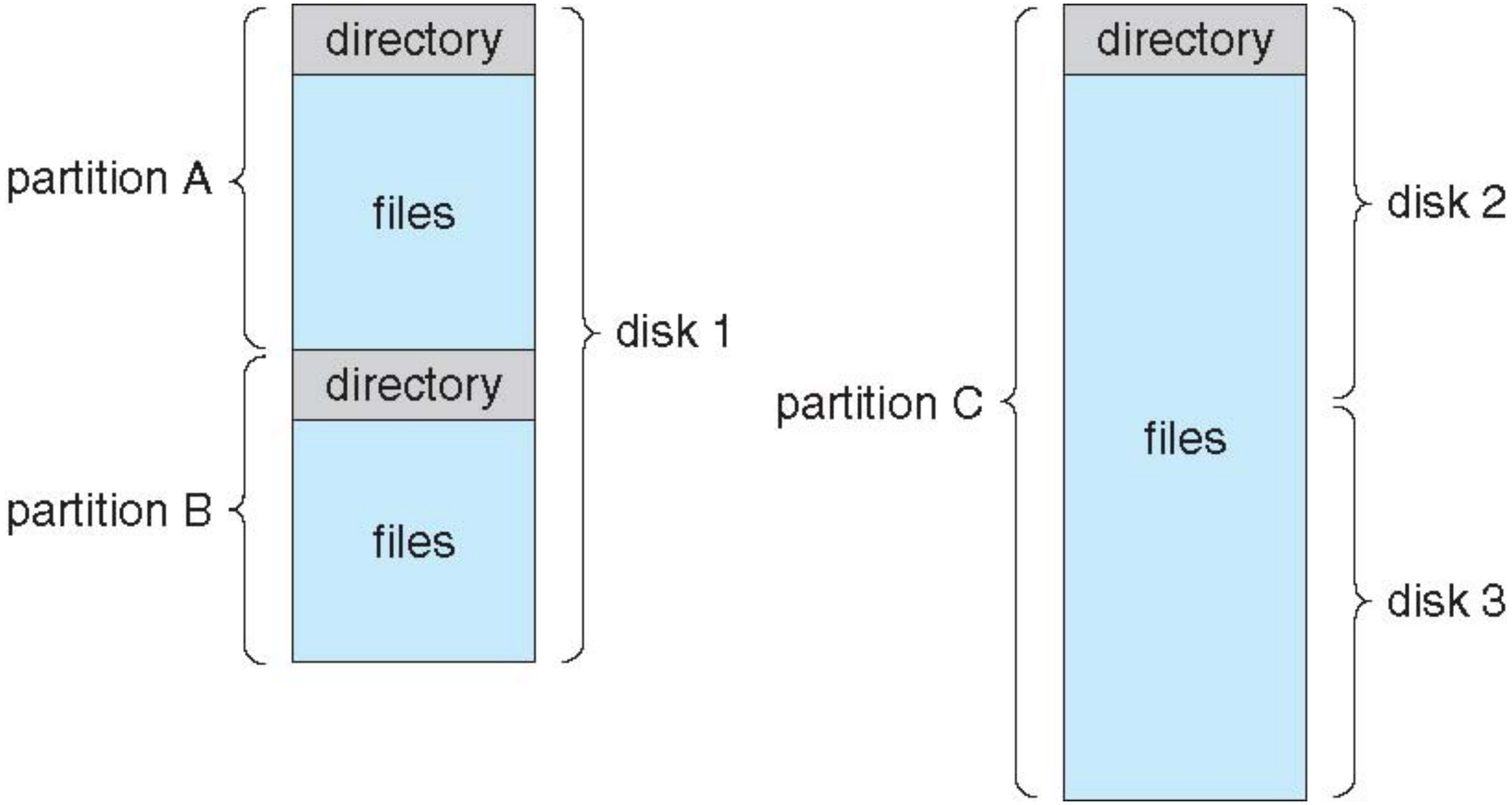


both the directory structure and the files reside on disk

Disk Structure

- Disk can be subdivided into **partitions**
  - partitions also known as **minidisks**, **slices**
  - different partitions can have different file systems
    - a partition containing file system is known as a **volume**
    - each volume tracks file system info in the volume's table of contents
    - a file system can be general purpose or special purpose
  - disk or partition can be used **raw** (without a file system)
    - applications such as database prefer raw disks

A Typical File-system Organization



Operations Performed on Directory

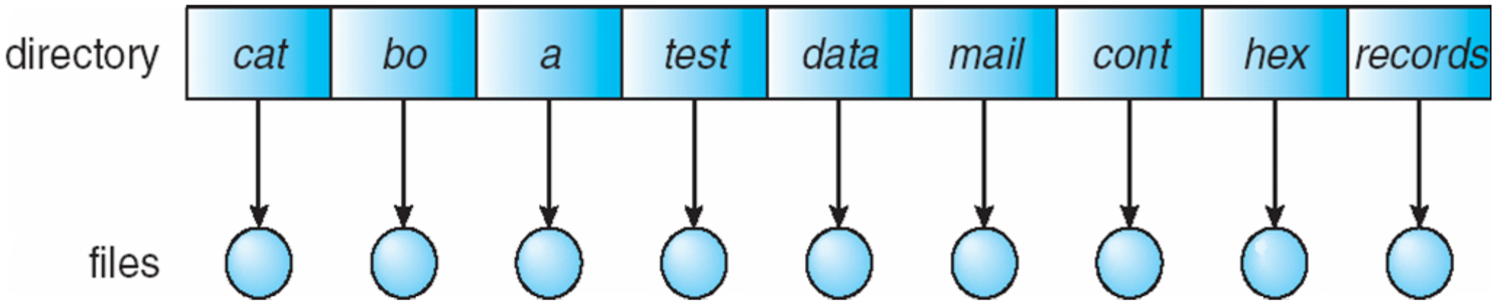
- Create/delete/rename a file
- List a directory
- Search for a file
- Traverse the file system
- …

Directory Organization

- Organize directories to achieve
  - **efficiency**: to locate a file quickly
  - **naming**: organize the directory structure to be convenient to users
    - two users can have same name for different files
    - the same file can have several different names
  - **grouping**: provide a way to logically group files by properties
    - e.g., all Java programs, all games, …
  - …

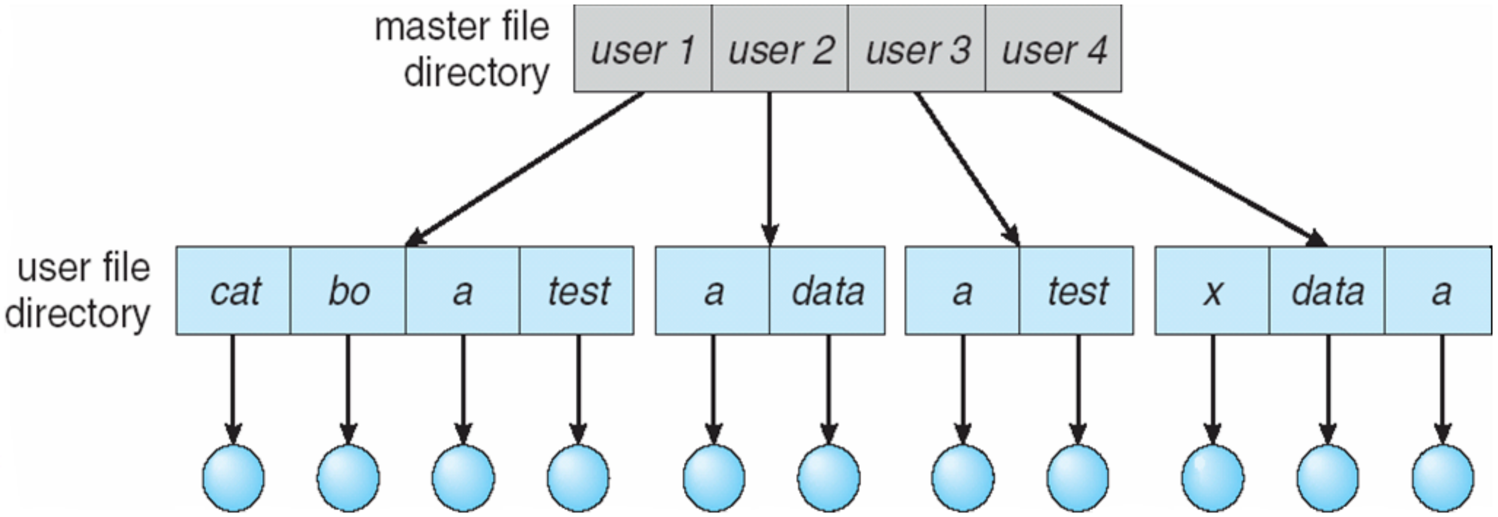
Single-Level Directory

- A single directory for all users
  - naming problems and grouping problems



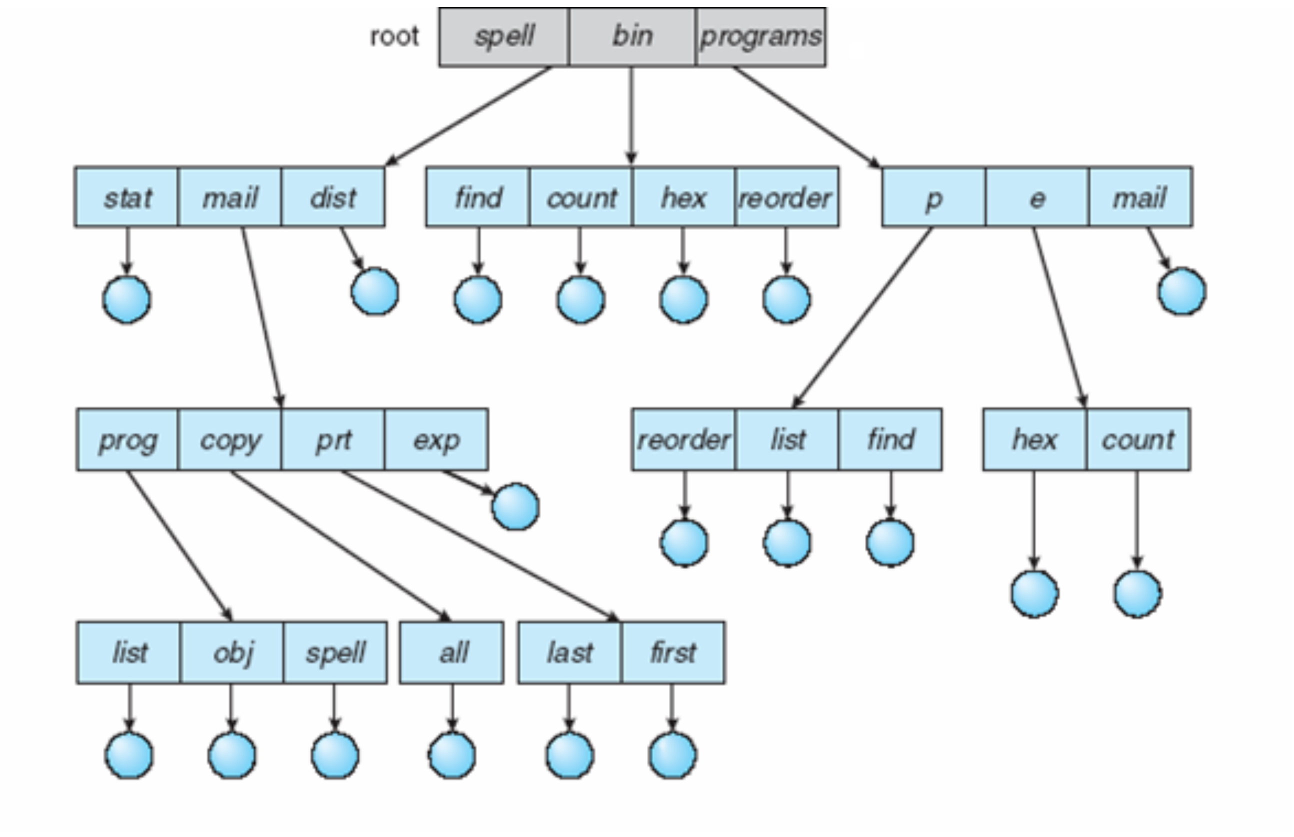
Two-Level Directory

- Separate directory for each user
  - different user can have the same name for different files
  - efficient to search, cannot group files



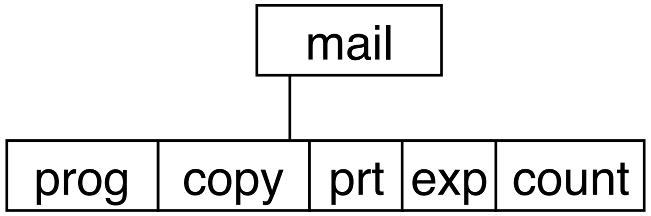
Tree-Structured Directories

- Files organized into trees
  - efficient in searching, can group files, convenient naming



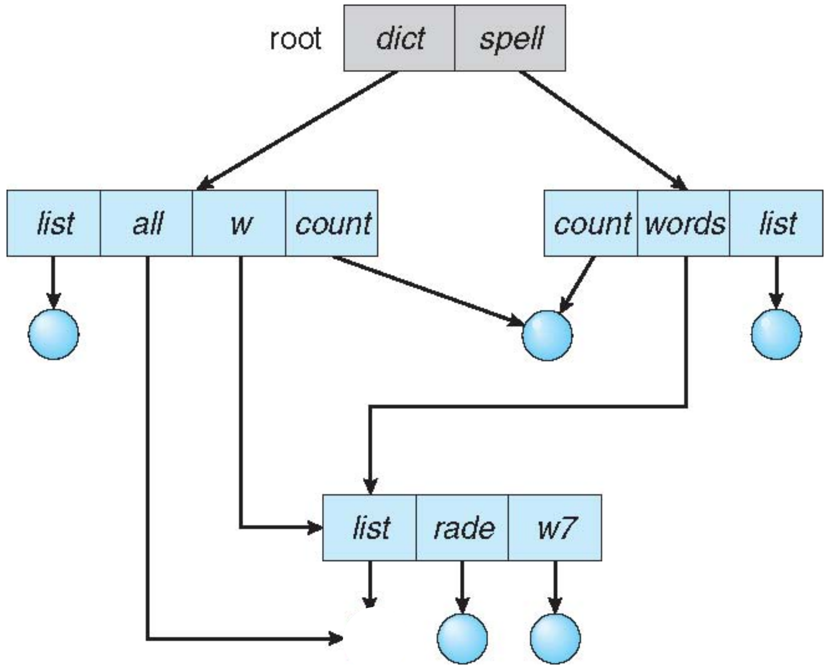
Tree-Structured Directories

- File can be accessed using **absolute** or **relative** path name
  - absolute path name: /home/alice/..
  - relative path is relative to the **current directory** (*pwd*)
    - creating a new file, delete a file, or create a sub-directory
    - e.g., if current directory is /mail, a **mkdir count** will create /mail/count



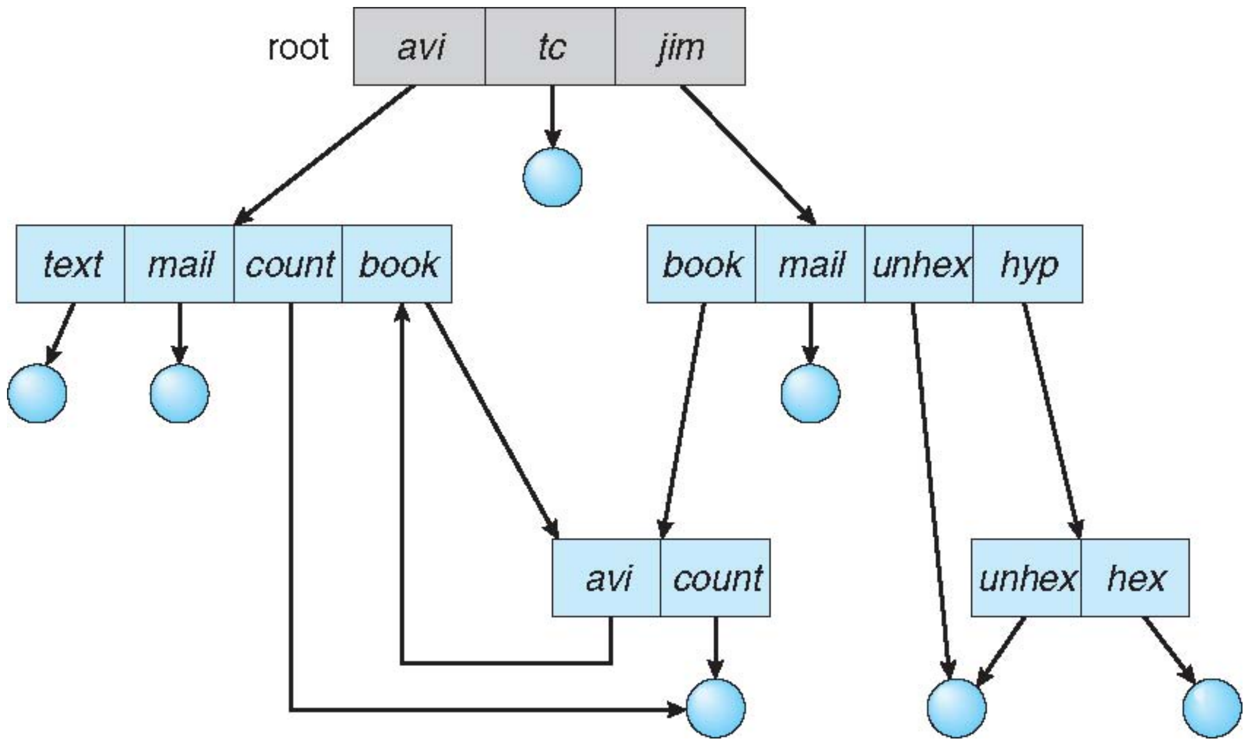
Acyclic-Graph Directories

- Organize directories into acyclic-graphs
  - allow links to a directory entry/files for **aliasing** (no longer a tree)
- Dangling pointer problem:
  - e.g., if delete /dict/all, /dict/w/list and /spell/words/list are dangling pointers
  - Solution: **backpointers/reference counter**
    - backpointers record all the pointers to the entity, a variable size record
    - count # of links to it and only (physically) delete it when counter is zero



General Graph Directory

- Allowing arbitrary links may generate cycles in the directory structure
- Solution
  - allow only links to files, but not directories
  - allow cycles, but use **garbage collection** to reclaim disk spaces
  - every time a new link is added use a **cycle detection** algorithm

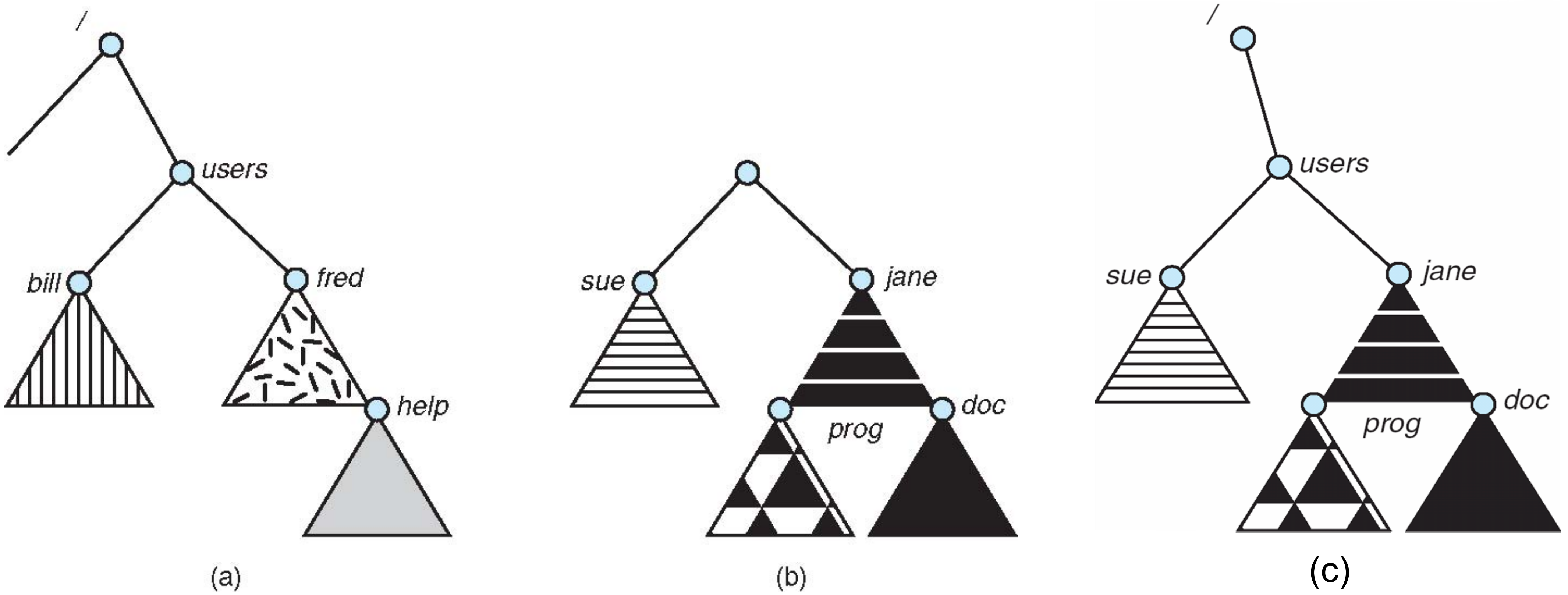


File System Mounting

- A file system must be **mounted** before it can be accessed
    - mounting link a file system to the system, usually forms a **single name space**
    - the location of the file system being mounted is call the **mount point**
    - a mounted file system makes the old directory at the mount point **invisible**

File System Mounting

- **a**: existing file system
- **b**: an unmounted partition
- **c**: the partition mounted at **/users**



File Sharing

- Sharing of files on multi-user systems is desirable
  - sharing must be done through a protection scheme
    - **User ID**s identify users, allowing protections to be per-user
    - **Group ID**s allow users to be in groups, permitting group access rights
- On distributed systems, files may be shared across a network
  - Network File System (NFS) is a common distributed file-sharing method

Remote File Sharing

- Use 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 FS from servers
  - a server can serve multiple clients
  - client and user-on-client identification is complicated
    - server cannot assume the client is trusted
    - standard OS file calls are translated into remote calls
  - **NFS** is standard UNIX file sharing protocol, **CIFS** is standard for Windows

Protection

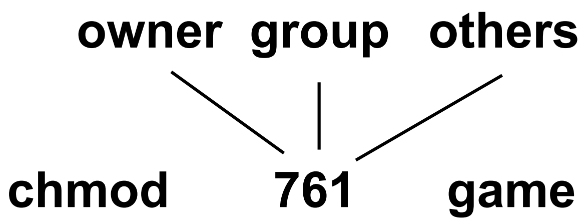
- File owner/creator should be able to control:
  - what can be done
  - by whom
- Types of access
  - read, write, append
  - execute
  - delete
  - list

Unix Access Control

- Three modes of access: **read**, **write**, **execute** (encoded in three bits)
- Three classes of users: **owner**, **group**, and **others**

		RWX
a) owner access:	7	1 1 1
b) group access:	6	1 1 0
c) others access:	1	0 0 1

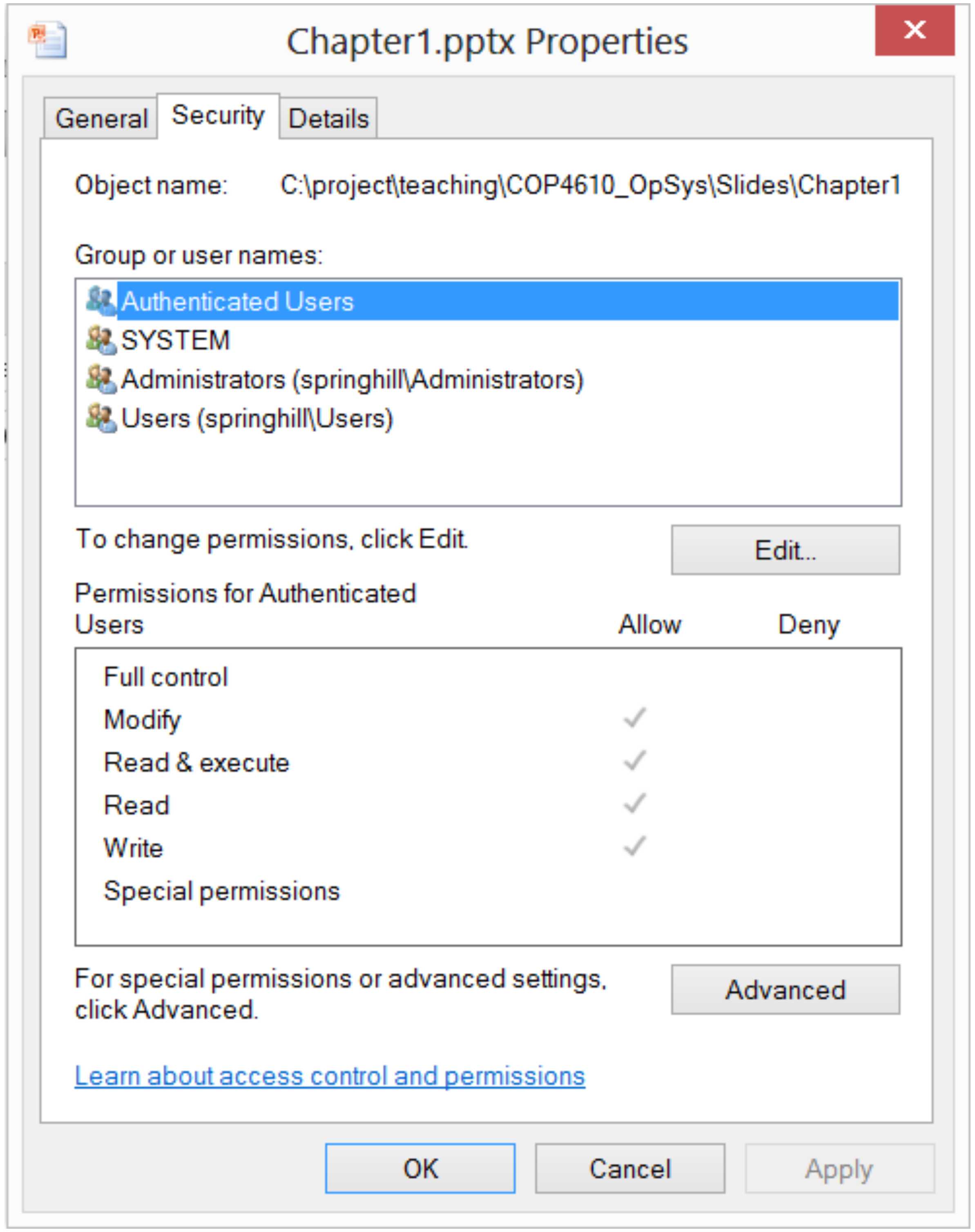
- To grant access to users, create a group and change its access mode
  - in Linux, use **chmod** and **chgrp**



owner group others

chmod 761 game

Windows 8 File Access-Control



Chapter1.pptx Properties

Object name: C:\project\teaching\COP4610_OpSys\Slides\Chapter1

Group or user names:

- Authenticated Users
- SYSTEM
- Administrators (springhill\Administrators)
- Users (springhill\Users)

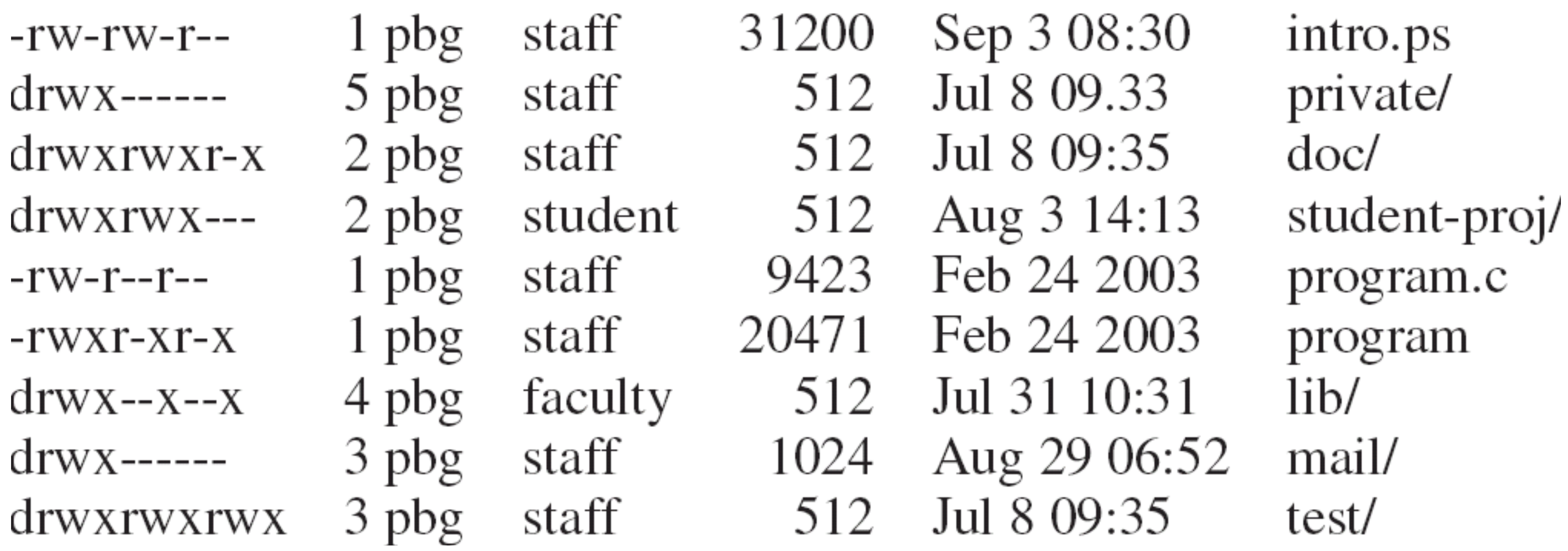
To change permissions, click Edit.

Permissions for Authenticated Users	Allow	Deny
Full control		
Modify	✓	
Read & execute	✓	
Read	✓	
Write	✓	
Special permissions		

For special permissions or advanced settings, click Advanced.

Learn about access control and permissions

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 10