1878

Western

UNIVERSITY · CANADA

Chapter 13 – File-System Interface

Spring 2023

Overview

- Overview
- File Concept
- Access Methods
- Directory Structure
- Protection

Overview

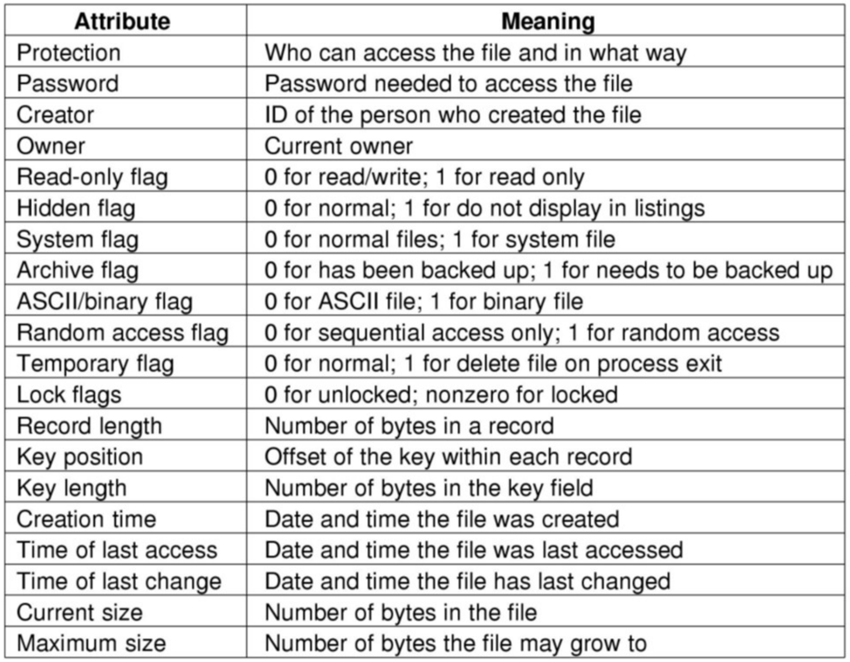
- The most visible aspect of a general-purpose OS is the file system
- The file system contains
  - **Files** (including metadata about each file)
  - **Directories** to organize the files
- The file system is stored on storage devices as described in Chapter 11
- The operating system provides an abstraction between the physical parts of the storage devices and the logical representation of the data in files

File Concept

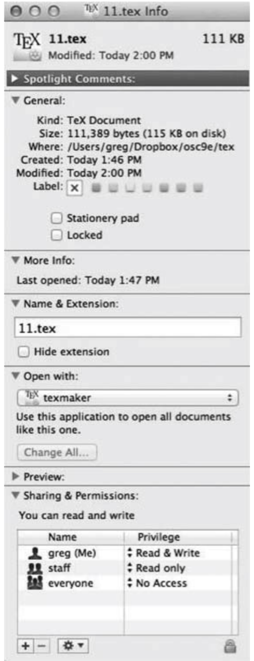
- A **file** is a named collection of related information
  - From the user's perspective, this is the most basic unit of data. All data is in a file.
- Files can be
  - Binary (e.g. executable files, compressed files)
  - Alphanumeric or text based (e.g. source files, text files)
- Since files are so fundamental, Linux uses files to represent attributes of the operating system or the system state under the `/proc` directory

File Concept

- Files have **attributes**. This metadata helps users identify information about the file



Attribute	Meaning
Protection	Who can access the file and in what way
Password	Password needed to access the file
Creator	ID of the person who created the file
Owner	Current owner
Read-only flag	0 for read/write; 1 for read only
Hidden flag	0 for normal; 1 for do not display in listings
System flag	0 for normal files; 1 for system file
Archive flag	0 for has been backed up; 1 for needs to be backed up
ASCII/binary flag	0 for ASCII file; 1 for binary file
Random access flag	0 for sequential access only; 1 for random access
Temporary flag	0 for normal; 1 for delete file on process exit
Lock flags	0 for unlocked; nonzero for locked
Record length	Number of bytes in a record
Key position	Offset of the key within each record
Key length	Number of bytes in the key field
Creation time	Date and time the file was created
Time of last access	Date and time the file was last accessed
Time of last change	Date and time the file has last changed
Current size	Number of bytes in the file
Maximum size	Number of bytes the file may grow to



11.tex Info

11.tex 111 KB
Modified: Today 2:00 PM

▶ Spotlight Comments:

▼ General:

Kind: TeX Document
Size: 111,389 bytes (115 KB on disk)
Where: /Users/greg/Dropbox/osc9e/tex
Created: Today 1:46 PM
Modified: Today 2:00 PM
Label: ☒

□ Stationery pad
□ Locked

▼ More Info:

Last opened: Today 1:47 PM

▼ Name & Extension:

11.tex

□ Hide extension

▼ Open with:

texmaker

Use this application to open all documents like this one.

▶ Preview:

▼ Sharing & Permissions:

You can read and write

Name	Privilege
greg (Me)	Read & Write
staff	Read only
everyone	No Access

File Concept

- Basic file operations
  - Create – Allocate space on disk and create a new file
  - Read – at a read location defined by a process
  - Write – at a write location defined by a process
  - Reposition within a file (seek) – at a location defined by a process
  - Delete – Erase the file and the metadata from disk
  - Truncate – Erase the file contents but not the metadata from disk
  - Open/Close – Move the contents of the file in or out of memory

File Concept

- Other file operations
  - Rename – Update the name of the file in the metadata
  - Append – Write to the end of the file
- Most operations can be achieved by combining the above operations
  - E.g. To "Copy a file": create a new file and open it for writing, open another file for reading, read the contents of the other file, write the data to the new file, then close both files

File Concept

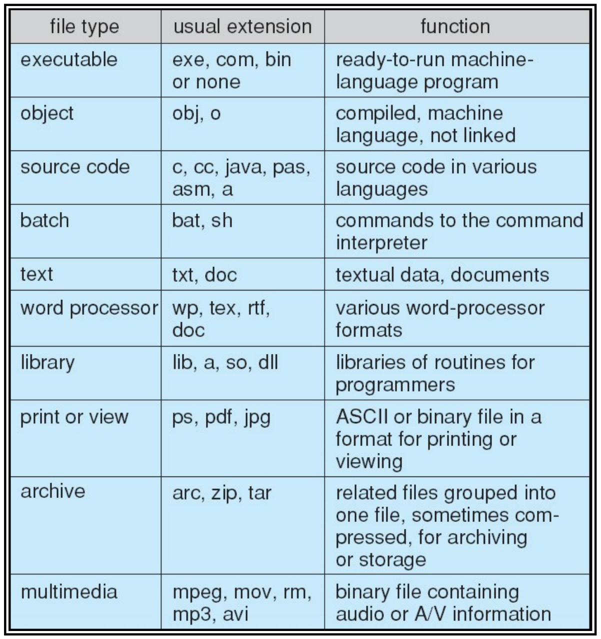
- The operating system maintains an **open-file table** to keep track of all the files that are "open" by processes at any given time
  - What file is open?
  - Read or write?
  - Which process(es) has opened the file?
  - What position in the file is the process(es)?

File Concept

- Some operating systems track **file locking**
  - Shared lock – Multiple processes can read from the file
  - Exclusive lock – Exactly one process can write to the file
- What happens in a lock conflict?
  - Mandatory locking – Prevent processes from sharing an exclusive lock by blocking it until the lock is released. Used in Windows
  - Advisory locking – Leave it up to the user processes to obtain the lock safely. Used in Linux
- Refer to chapters 6, 7, and 8 for how locking must be handled

File Concept

- Some applications use file extensions to identify files. The operating system can help by associating default applications to certain extensions.



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 Concept

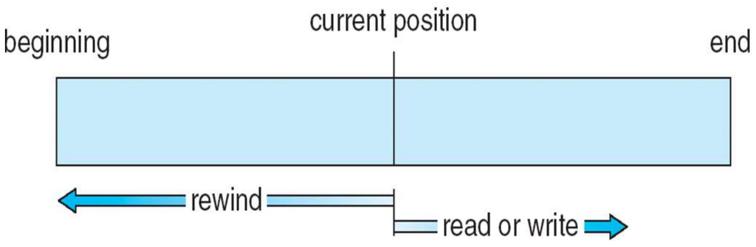
- Some files have a rigid **structure**, others are completely free-form. The operating system doesn't care but the applications that use the files do.
  - Free-form – No structure. Sequence of characters (text) or bytes (binary)
  - Simple record structure – Fixed or variable length lines
  - Complex record structure – Control characters can be added at intentional places to a simple record structure to make a complex structure. Other approaches use XML, json, proprietary formats, etc.
- If the operating system had to keep track of all the different structure types, this would add overhead. Adding new structures (e.g. new file formats) would mean an update to the operating system every time to support them.

Access Methods

- A file must be loaded into memory before it can be used. The file can then be read with
  - Sequential access
  - Direct access
  - Other access methods

Access Methods

- Sequential access – This is the most common method
- Operations required
  - Read next
  - Write next
  - Reset
  - Skip forward or backward (on most systems)



Access Methods

- Direct access – Read and write records in any order (typically used with knowledge about where the file's blocks are on disk). Ideal, for example, with databases who know exactly what portion of the data needs to be read

- Operations required

  - Read block N

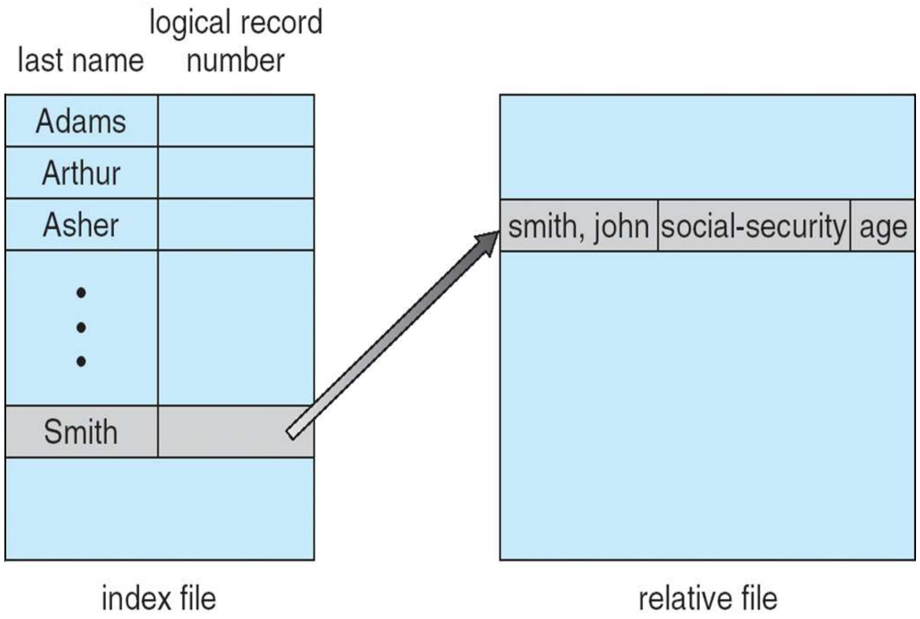
  - Write to block N

- Since files are scattered in blocks across the disk, N is usually a number of blocks relative to the first block. A translation table maps logical blocks to physical blocks

- Sequential access could easily be simulated with direct access

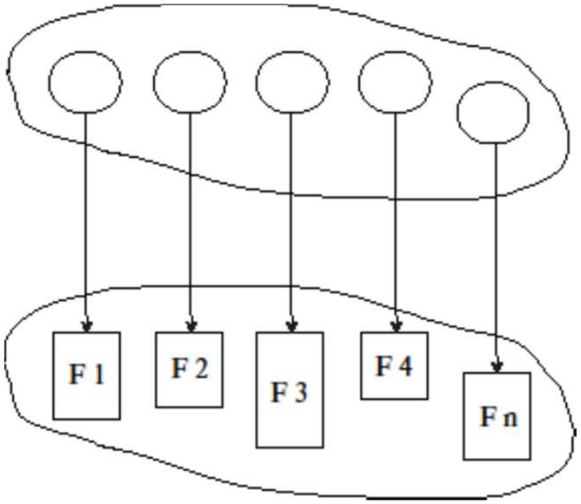
Access Methods

- Other access methods
  - Use indexes to blocks for quick lookup. Hold the indexes in memory



Directory structure

- Files are grouped in directories
  - The directory typically contains all the filenames and their identifiers
  - The identifier is used to look up all the other attributes



Directory Structure

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

Directory Structure

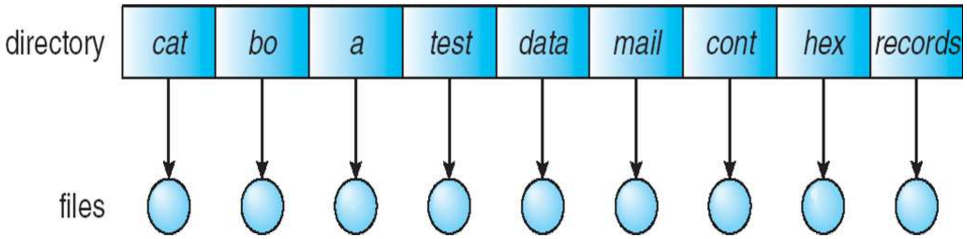
- Directories are organized logically
  - Efficiency – Quickly find the files the user wants
  - Naming – Easily identify the files the user wants
  - Grouping – Group similar files together logically (e.g. Driver files, game files, etc.)

Directory Structure

- Directory structures
  - Single
  - Two-level
  - Tree
  - Acyclic-Graph

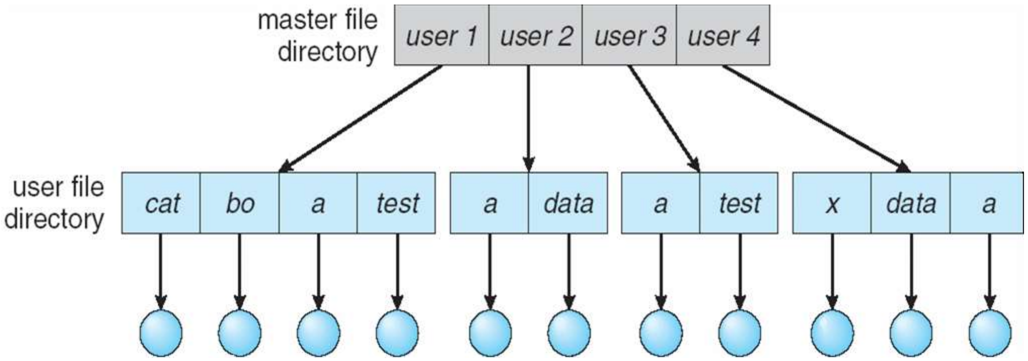
Directory Structure

- Single-level directory
  - Simplest to understand and implement, but it is the least useful
  - Naming problem – All files on the system must have a unique name
  - Grouping problem – It is not possible to group similar files together



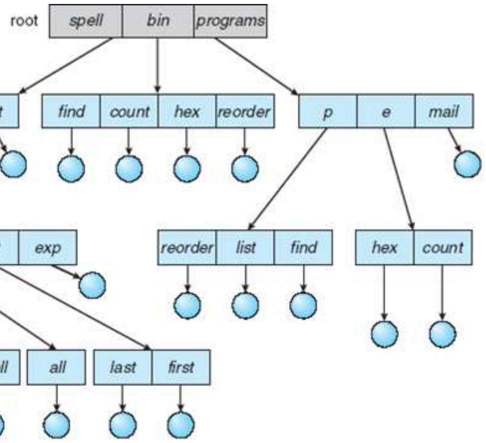
Directory Structure

- Two-level directory
  - Resolves the naming problem but grouping is still a problem



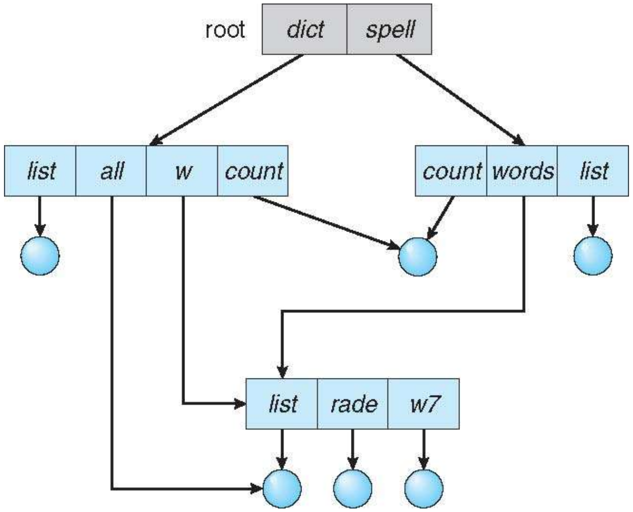
Directory Structure

- Tree-Structured directories
  - Start with a **root** directory and maintain files and subdirectories below it
  - Use a flag (eg. 0) to signify a file and another flag (eg. 1) to signify a directory



Directory Structure

- Acyclic-Graph directories
  - Relax the tree structure to permit file and directory sharing
  - Links and shortcuts allow files to point to existing files

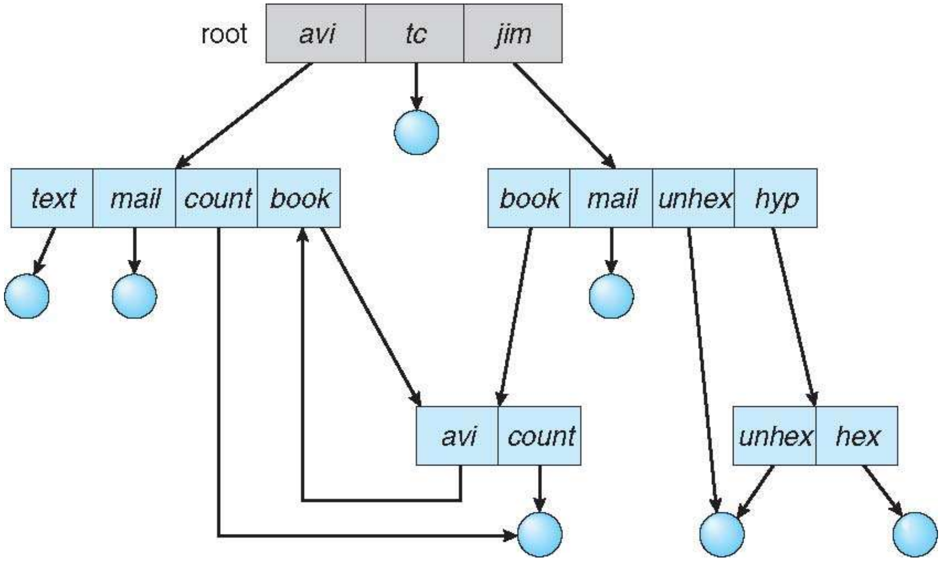


Directory Structure

- Acyclic-Graph directories
  - Directory traversal must account for potential cycles
    - Allow links to files only and not directories
    - Limit the number of directories that can be searched
  - Deletes need to account for potential cycles
    - Keep a count of incoming references. Use garbage collection to remove unreferenced files
  - Check for potential cycles whenever a link is added

Directory Structure

- Acyclic-Graph directories



Directory Structure

- Regardless of the structure, if the directory is not specified, file operations occur on the current working directory
    - Creating or deleting a file or directory occurs in the current directory
    - Deleting the current directory could remove all files and directories in the subtree
        - Linux, for example, requires the directory to be empty before it can be deleted. The `rm -r` command can be used to manually override this restriction
        - Backups should be used in case of accidental data loss

Protection

- Files need to be protected
    - From physical damage
        - We use techniques discussed in Chapter 11
    - From improper access
        - Authentication – A valid username and password should be provided
        - Authorization – If a user is authenticated, is the user permitted to use a file?
        - Encryption – If the drive is copied or removed, the files should still be protected

Protection

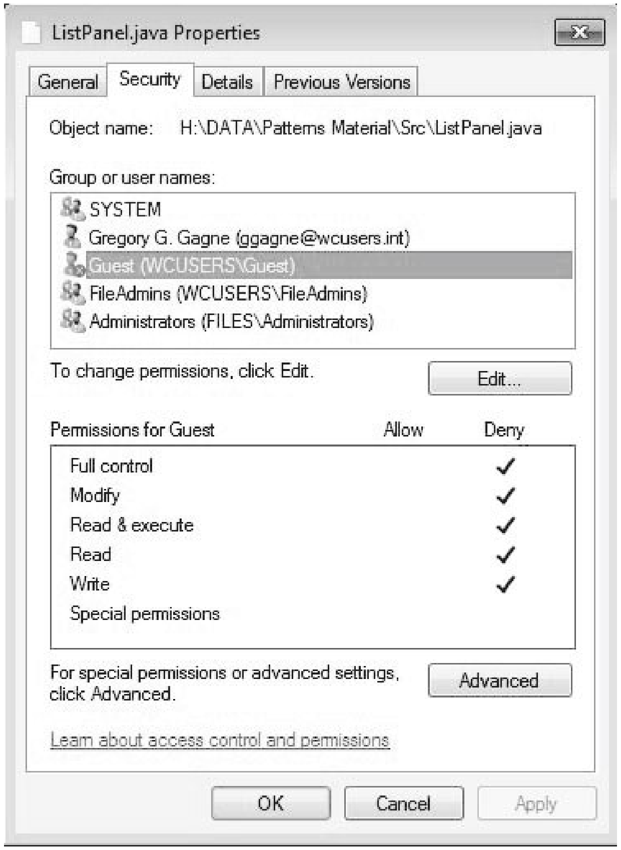
- Common protection mechanisms
  - Read – Can the contents of the file be read?
  - Write – Can data be written or re-written to the file?
  - Execute – Can the contents of the file be loaded into memory and executed?
  - Append – Can data be appended to the file?
  - Delete – Can the file be deleted and space reclaimed?
  - List – Can the name and attributes be listed?
  - Attribute change – Can the attributes (metadata) be changed?

Protection

- Most operating systems employ protection based on the identity of the user
  - **Access control lists (ACLs)** map users to permissions
- ACLs can be very large. Classifications allow Linux to condense access
  - Owner – The user who created the file
  - Group – A set of users who are sharing the file
  - Other – All other users
- More complex permissions can still be achieved with ACLs.
- Windows uses ACLs through the GUI

Protection

```
-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/
```



Object name: H:\DATA\Patterns Material\Src\ListPanel.java

Group or user names:

- SYSTEM
- Gregory G. Gagne (ggagne@wcusers.int)
- Guest (WCUSERS\Guest)
- FileAdmins (WCUSERS\FileAdmins)
- Administrators (FILES\Administrators)

To change permissions, click Edit.

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

For special permissions or advanced settings, click Advanced.

1878

Western

UNIVERSITY · CANADA