



# UNIT IV

## File System

**Dr S.Rajarajan**  
**SoC**

**School of Computing**  
**SASTRA**

# File Concept

- Computers can store information on various storage media, such as **magnetic disks**, **magnetic tapes**, and **optical disks**.
- Operating system provides a **uniform logical view** of stored information.
- The operating system **abstracts from the physical** properties of its storage devices to define a logical storage unit, the **file**.
- Files are **mapped** by the operating system onto **physical devices**.
- These storage devices are usually **nonvolatile**, so the **contents are persistent** between system reboots

# What is a file ?

- A file is a **named collection** of **related information** that is recorded on secondary storage.
- From a user's perspective, a **file is the smallest allotment** of logical secondary storage.
- Commonly, **files represent programs and data**.
- Data files may be **numeric, alphabetic, alphanumeric, or binary**.
- 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**,

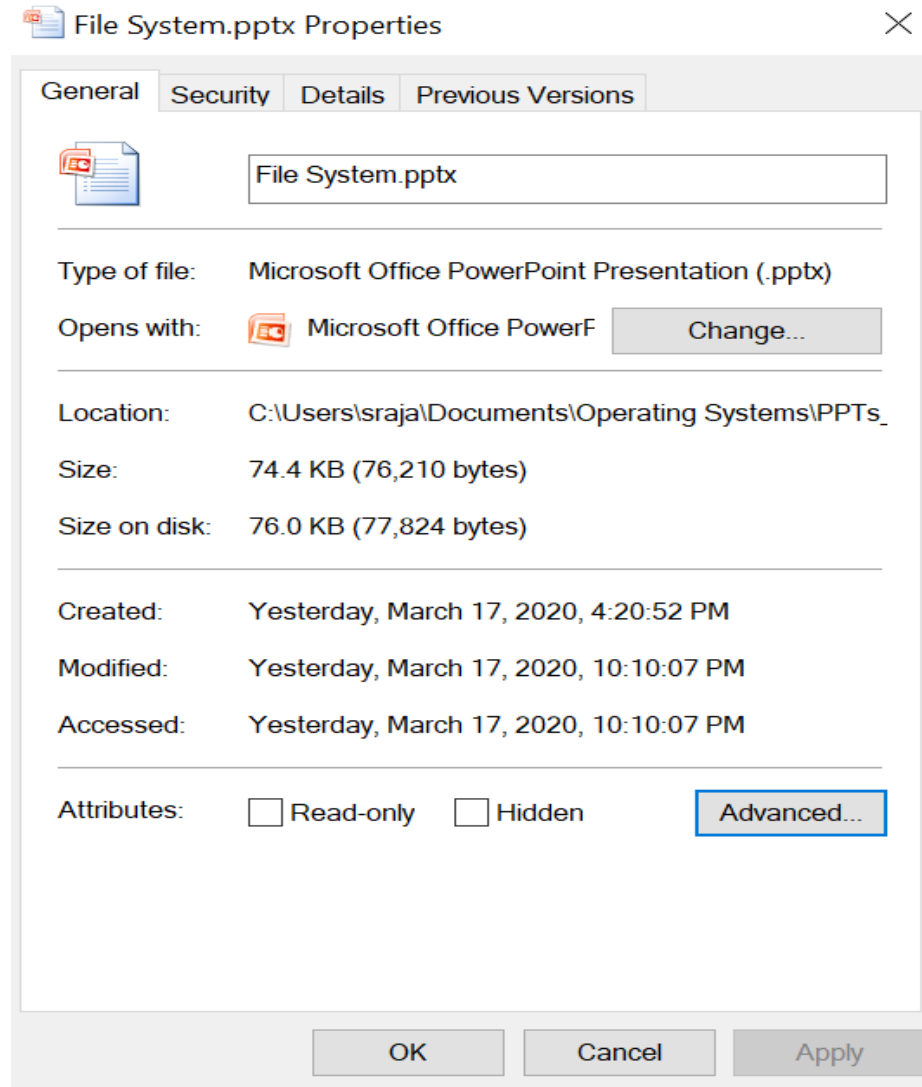
# File Attributes

- A file is **named**, for the convenience of its human users, and is
- A name is usually a **string of characters**, such as `example.c` referred to by its name.
- Some systems are **case sensitive**.
- When a file is named, it becomes independent of the process, the user, and even the system that created it.
- **Anyone can operate the file** by referring its name.

- A file's **attributes** vary from one operating system to another but typically consist of these:
- **Name.** The symbolic file name
- **Identifier.** This unique tag, usually a number, identifies the file.
- **Type.** This information is needed for systems that support different types of files.
- **Location.** This information is a pointer to a device and to the location of the file.
- **Size.** The current size of the file (in bytes, words, or blocks) and possibly the maximum allowed size are included in this attribute.
- **Protection.** Access-control information determines who can do reading, writing, executing, and so on.

- **Time, date, and user identification.** This information may be kept for creation, last modification, and last use.

# File properties in Windows



# 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 operating system can provide **system calls** to **create**, **write**, **read**, **reposition(seek)**, **delete**, and **truncate( erase file)** files.
- Most of the file operations mentioned involve **searching the directory** for the entry associated with the named file.
- To avoid this constant searching, many systems require that an **open() system call** be made before a file is first used.



- The operating system keeps a table, called the **open-file table**, containing information about all open files.
- When a file operation is requested, the file is specified via **an index into this table**, so no searching is required.
- When the **file is no longer** being actively used, it is **closed by the process**, and the operating system **removes its entry** from the open-file table.
- Implementation of the **open()** and **close()** operations is **more complicated** in an environment where **several processes** may open the file **simultaneously**.

- Several pieces of information are associated with an open file.
  - File pointer
  - File open count
  - Disk location of file
  - Access rights
- Some operating systems provide facilities for **locking** an open file (or sections of a file).
- File locks allow one process to lock a file and **prevent other processes** from gaining access to it.
- File locks are useful for files that are **shared by several processes**.

# Types of locks

- A **shared lock** is akin to a reader lock in that several processes can acquire the lock concurrently.
- An **exclusive lock behaves like a writer lock**; only one process at a time can acquire such a lock.
- operating systems may provide either **mandatory or advisory** file-locking mechanisms.
- If a lock is **mandatory**, then once a process acquires an exclusive lock, the operating system will prevent any other process from accessing the locked file.
- If a lock is **advisory**, then the operating system may or may not prevent any other process from accessing the locked file.

# 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, perl, asm	source code in various languages
batch	bat, sh	commands to the command interpreter
markup	xml, html, tex	textual data, documents
word processor	xml, rtf, docx	various word-processor formats
library	lib, a, so, dll	libraries of routines for programmers
print or view	gif, pdf, jpg	ASCII or binary file in a format for printing or viewing
archive	rar, zip, tar	related files grouped into one file, sometimes compressed, for archiving or storage
multimedia	mpeg, mov, mp3, mp4, avi	binary file containing audio or A/V information

# File Structure

- File types also can be used to indicate the **internal structure** of the file.
- Further, certain files must conform to a required structure that is understood by the operating system.
- Some operating systems extend this idea into a **set of system-supported file structures**, with sets of **special operations** for manipulating files with those structures.
- Some operating systems impose (and support) a **minimal number** of file structures.

- This approach has been adopted in UNIX, Windows, and others.
- UNIX considers each file to be a sequence of **8-bit bytes**; **no interpretation** of these bits is made by the operating system.
- This scheme provides **maximum flexibility** but **little support**.
- Each **application program** must include **its own code to interpret** an input file as to the appropriate structure.
- However, all operating systems **must support** at least one structure—that of **an executable file**—so that the system is able to **load and run programs**.