Chapter Six

File Systems

Fundamental concepts

❑ ***Data means:***

➢ Facts and statistics are collected together for reference or analysis.

➢ The quantities or characters or symbols, on which operations are performed by a computer are known as ***data***.

➢ Which may be stored and transmitted in the form of **electrical signals and recorded on magnetic, optical, or mechanical recording media**.

➢ *A file management system:* is a type of software that **manages data files** in a computer system.

➢ It has limited capabilities and is designed to manage individual or group files, such as **special office documents and records**.

➢ The data may be ***numbers, characters, or binary information***.

Cont. ...

❑ *Metadata:* means *"data about data"*

➢ *Metadata* is data that provides information about other data.

➢ *Three* distinct *types* of metadata exist: **descriptive metadata, structural metadata, and administrative metadata.**

❑ *Descriptive metadata***:** describes a resource for purposes such as discovery and identification.

It can include elements such as **title, abstract, author, and keywords**.

❑ *Structural metadata***:** metadata about containers of data indicates how compound objects are put together. **For example** how pages are ordered from chapters. It describes the **types, versions, relationships**, and other characteristics of **digital materials**.

❑ *Administrative metadata***:** provides information to help **manage resources**, such as **when and how it was created**, file type and other technical information, and **who can access it**.

➢ *Metadata* is traditionally used in the **card catalogs of libraries, museums, digital audio files, websites, traffic analysis, etc.**

Cont. ...

❑ *A file* is a collection of data or information that has a name called the file name.

➢ Almost all information stored in a computer must be in **a file**.

➢ *A file* is an object on a computer that stores data, information, settings, or commands used with a computer program.

➢ All computer applications need to store and retrieve information.

➢ While a **process is running**, it can store a **limited amount of information** within its own address space.

➢ However, the storage capacity is **restricted** to the size of the virtual address space.

➢ A **second problem** with keeping information within a process address space is that when the process terminates, the information is lost.

➢ For many applications, the information must be **retained for weeks, months, or even forever.**

Cont. ...

➢ A **third problem** is that it is frequently necessary for multiple processes to access the information at the same time.

➢ Thus we have **three essential requirements** for long-term information storage:

1. It must be possible to store a very large amount of information.
2. The information must survive the termination of the process using it.
3. Multiple processes must be able to access the information concurrently.

➢ Magnetic disks have been used for years for this **long-term storage** and **support** of **two operations:**

   ➢ **Read block k**
   ➢ **Write block k**

➢ **Files** are logical units of information created by processes.

➢ A disk usually **contains thousands or even millions** of them, each one independent of the others. Processes can read existing files and create new ones if need be.

File Operations

➢ File exists to **store information** and allow it to be **retrieved** later.

➢ Different system provides different operations to allow **storage** and **retrieval**.

➢ **Below is a discussion of the most common system calls relating to files.**

❑ *Create:* The file is created with no data. The purpose of the call is to announce that the files are coming and to set some of the attributes.

❑ *Delete:* When the file is no longer needed, it has to be deleted to free up the disk space.

❑ *Open:* Before using a file, a process must open it. The purpose of open calls is to allow the system to fetch the attributes for rapid access on later calls.

❑ *Close:* When all the accesses are finished, the attributes and disk address are no longer needed, so the file should be closed to free up internal table space.

❑ *Read:* Data are read from a file. Usually, the bytes come from the current position. The caller must specify how much data is needed and must also provide a buffer to put them in.

Cont. …

❑ *Write:* Data are written to the file again, usually, at the current position. If the current position is the end of the file, the file size increases.

❑ *Append:* This call is a restricted form of writing. It can **only add data to the end of the file**. Systems that provide a minimal set of system calls do not generally have append, but many systems provide multiple ways of doing the same things, and these systems sometimes have append.

❑ *Seek:* For random access files, a method is needed to specify from where to take the data. One common approach is a system call, seek that repositions the file pointer to a specific place in the file. After this call has been completed, data can be read from, or written to that position.

❑ *Rename:* It frequently happens that a user needs to change the name of an existing file. This system call makes that possible. It is not always strictly necessary, because the file can usually be copied to a new file with the new name, and the old file is then deleted.

File Organization

➢ File organization refers to the **way data is stored in a file**.

➢ File organization is very important because it determines the **methods of access, efficiency, flexibility, and storage devices to use.**

➢ The **methods** of organizing files on a storage media. This includes:
  ✓ **Sequential / Serial, Random or Direct, Indexed- Sequential**

❑ *Sequential File Organization:*

➢ Records are stored and accessed in a particular order stored using a key field. Retrieval requires searching sequentially through the entire file record by record to the end., better file searching methods like the **binary search technique** can be used to reduce the time used for searching a file. Records in a file are stored and accessed one after the other.

Cont. ...

❑ *Random or Direct File Organization:*

➢ Records are stored randomly but accessed directly.

➢ To access a file stored randomly, a record key is used to determine where a record is stored on the storage media.

➢ **Magnetic and optical disks** allow data to be stored and accessed randomly.

Advantages: Quick retrieval of records.

Indexed-Sequential File Organization:

➢ Almost similar to the sequential method but only that an index is used to enable the computer to locate individual records on the storage media.

➢ **For example,** on a **magnetic drum**, records are stored sequentially on the tracks. However, each record is assigned an **index** that can be used to access it directly.

Content and Structure of Directories:

➢ A **directory** is a collection of nodes containing information about all files. Both the directory structure and the files reside on disk.

➢ **The following operations are performed on directories:**
  ✓ Search for a file
  ✓ Create a file
  ✓ Delete a file
  ✓ List a directory
  ✓ Rename a file
  ✓ Traverse the file system

➢ **The directory is organized logically to obtain:**

✓ Efficiency- locating a file quickly.

✓ Naming is convenient to users.
  ✓ Two users can have the 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,…)
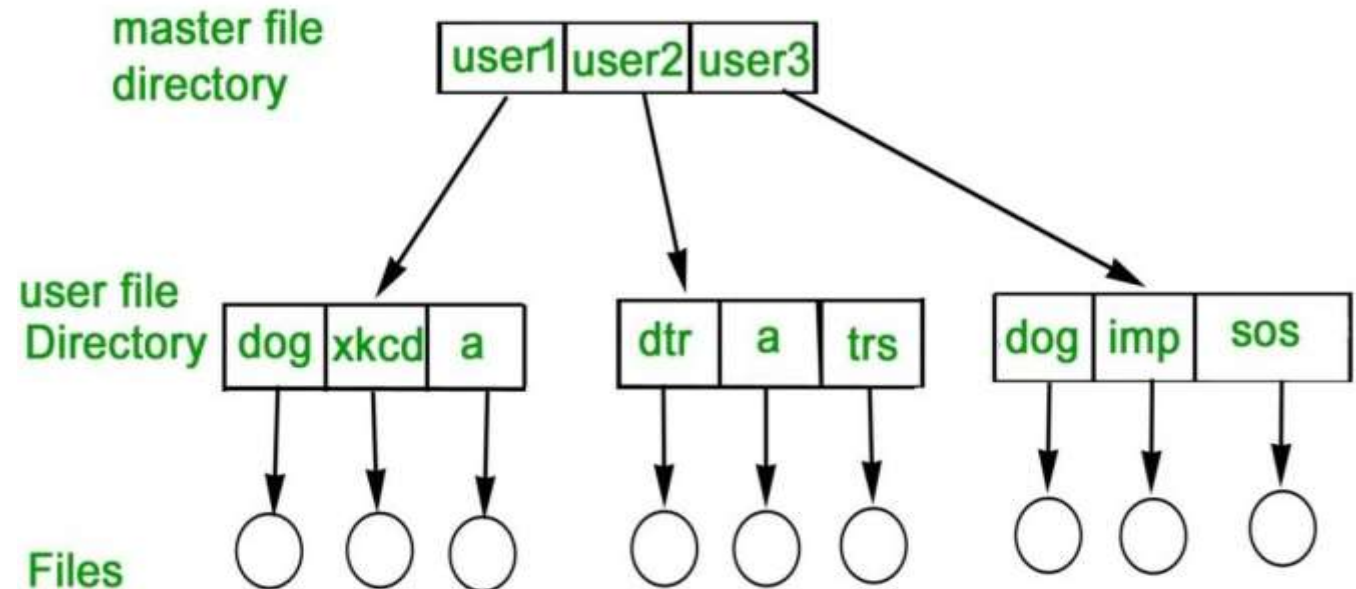
Types of directory

❑ *Single-level directory:*

➢ A single directory for all users

➢ Naming problem

➢ No grouping
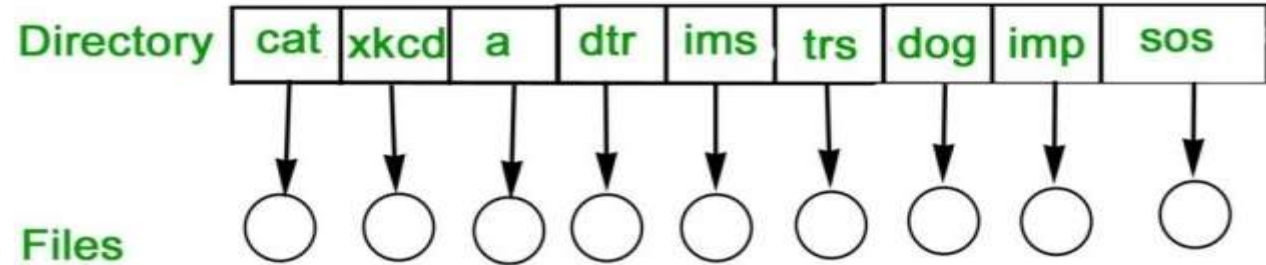
❑ *Two-level directory:*

➢ Separate directory for each user

➢ Path name Can have
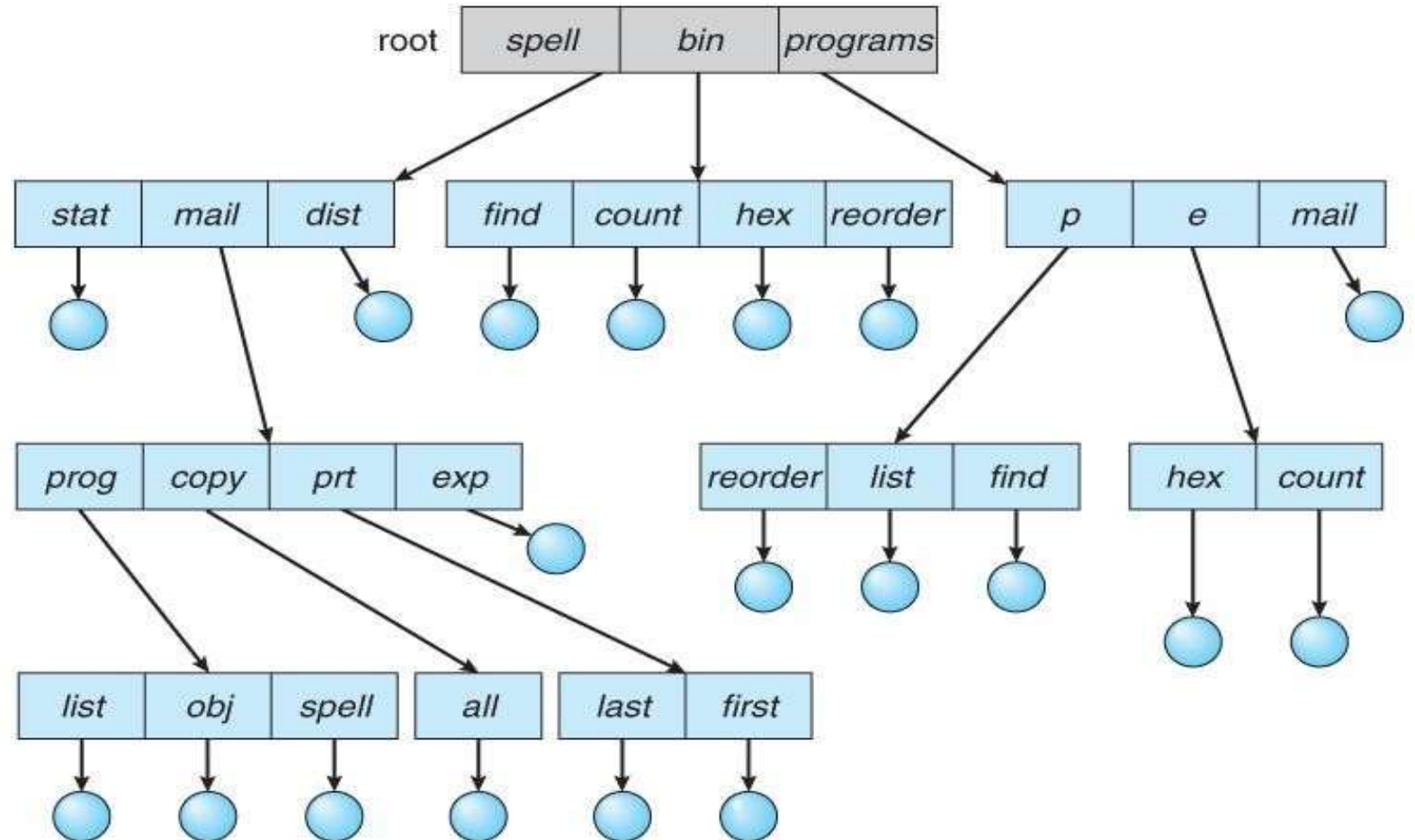   the same file name for
   different user.

➢ Efficient searching

11

Cont....

❑ ***Tree-structured directories***

➢ Efficient searching

➢ Grouping capability

➢ Current directory

  (Working directory)
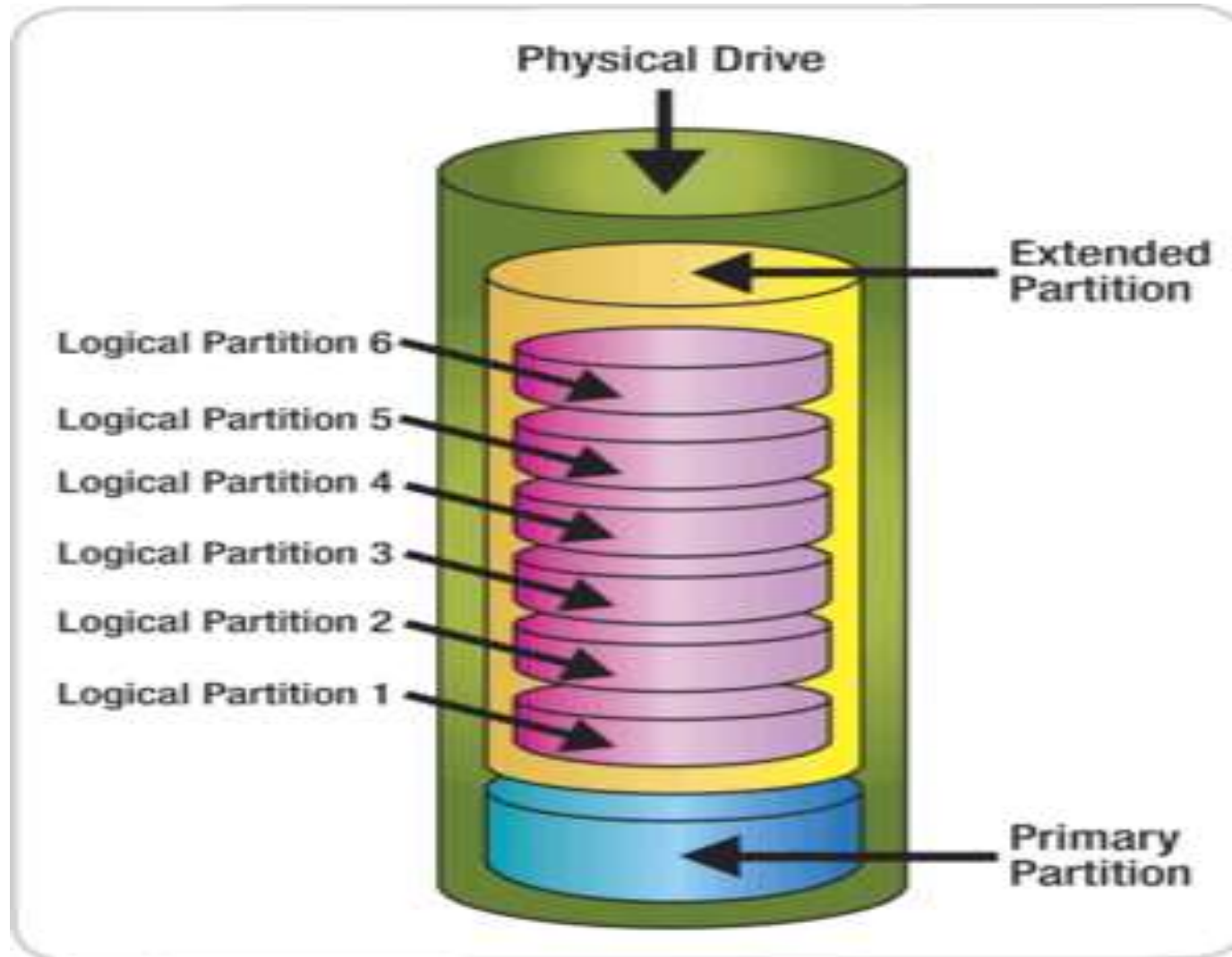
  eg. spell/mail/prog/obj.

File System Techniques:

❑ ***Partitioning:*** A partition is a **logical division** of a **hard disk** that is treated as a separate unit by operating systems and file systems.

➢ The operating systems and file systems can manage information on each partition as if it were a distinct hard drive.

➢ This allows the drive to operate as several smaller sections to improve efficiency, although it reduces usable space on the hard disk because of additional overhead from multiple operating systems.

➢ A disk manager partition manager allows system administrators to create, resize, delete, and manipulate partitions, while a partition table logs the location and size of the partition.

➢ Each partition appears to the operating system as a distinct logical disk, and the operating system reads the partition table before any other part of the disk.

Cont. ...

➢ Once a partition is created, it is formatted with a file system such as:
  - ✓ **NTFS on Windows drives**
  - ✓ **BSD partition**
  - ✓ **FAT32 and exFAT for removable drives**
  - ✓ **Solaris x86**
  - ✓ **HFS Plus on Mac computers**
  - ✓ **DOS partition**
  - ✓ **Ext4 on Linux etc.**

➢ Data and files are then written to the file system on the partition.

➢ When users boot the operating system in a computer, a critical part of the process is to give control to the first sector on the hard disk.

➢ This includes the partition table that defines how many partitions will be formatted on the hard disk, the size of each partition, and the address where each disk partition begins.

➢ The sector also contains a program that reads the boot sector for the operating system and gives it control so that the rest of the operating system can be loaded into RAM.

Cont. ...

Physical Drive

Extended Partition

Logical Partition 6

Logical Partition 5

Logical Partition 4

Logical Partition 3

Logical Partition 2

Logical Partition 1

Primary Partition

Fig: Disk Partitioning

Mounting and Unmounting

❑ ***Mounting:*** Mounting is a process by which the operating system makes files and directories on a storage device (such as a hard disk drive, CD-ROM, or network share) available for users to access via the computer file system.

➢ In general, the process of mounting comprises the operating system acquiring access to the storage medium, recognizing, reading, and processing file system structure and metadata on it, before registering them to the virtual file system component.

➢ The exact location in VFS where the newly-mounted medium got registered is called **the mount point**, when the mounting process is completed, the user can access files and directories on the medium from there.

❑ ***Unmounting:*** An opposite process of mounting is called unmounting, in which the operating system cuts off all user access to files and directories on the mount point, writes the remaining queue of user data to the storage device, refreshes file system metadata, then relinquishes access to the device, making the storage safe for removal.

➢ Normally, when the computer is shutting down, every mounted storage will undergo an unmounting process to ensure that all queued data got written, and to preserve the integrity of the system structure on the media.

Virtual File System

- A **virtual file system** is programming that forms an interface between an operating system's kernel and a more concrete file system.

- The VFS serves as an abstraction layer that gives applications access to different types of file systems and local and network storage devices.

- VFS on UNIX provides an object-oriented way of implementing file systems. VFS allows the same system call interface (the API) to be used for different types of file systems.

- The API is to the VFS interface, rather than only a specific type of file system.
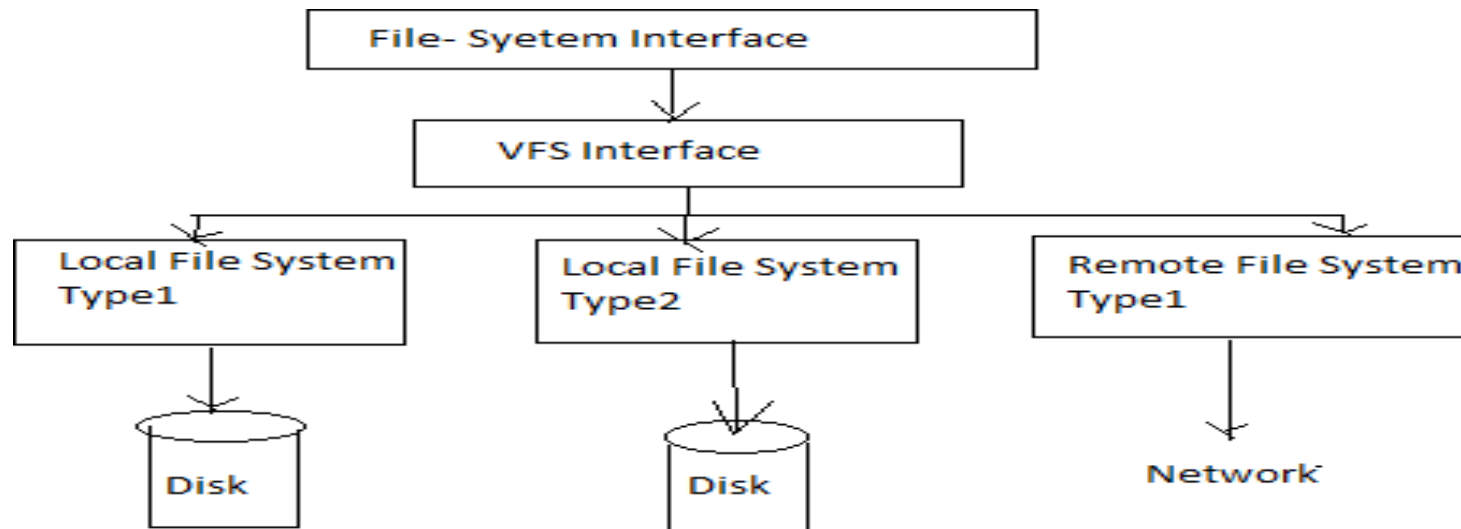
Memory –Mapped Files

➢ A **memory-mapped file** is a feature for all modern operating systems. It requires coordination between the memory manager and the I/O subsystem.

➢ You can tell the OS that some file is the backing store for a certain portion of the process memory. To understand that there is a virtual memory.

Process A

View 1

View 2

Process B

View 1

View 2

View 3

Memory Mapped file

Block A

Block B

Block C
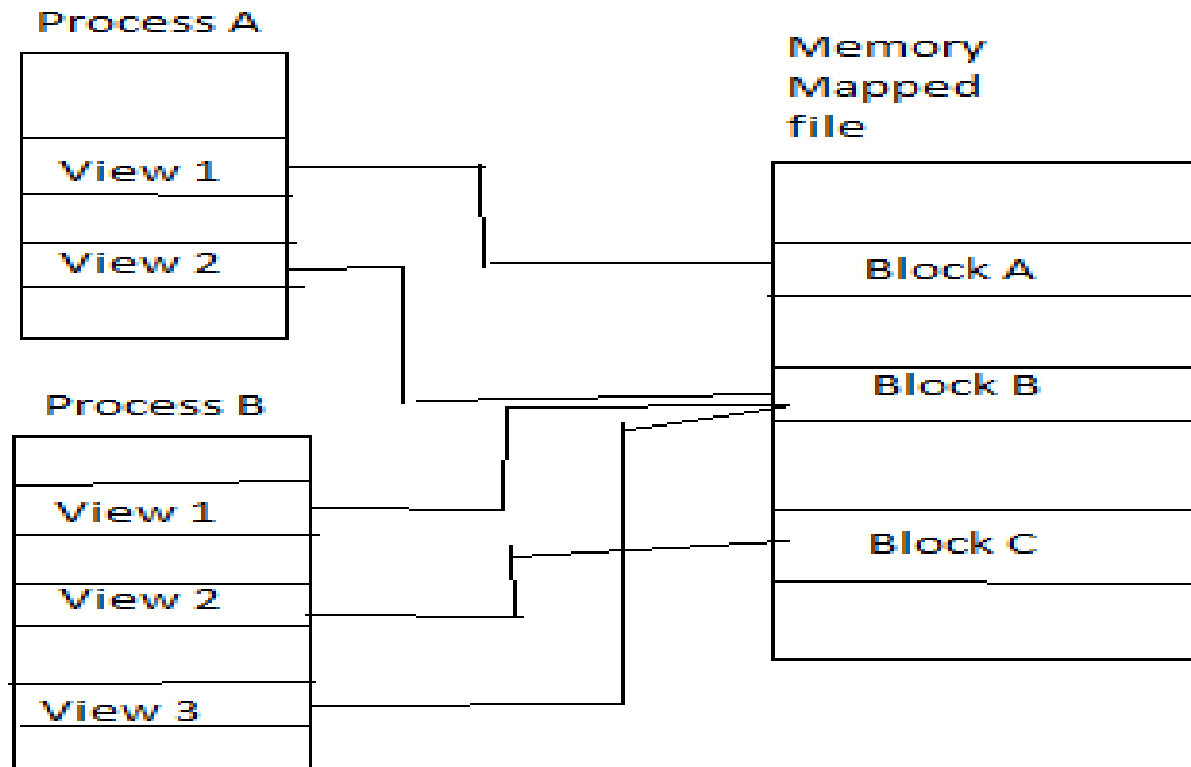
Cont. ...

➢ **Memory-mapped files offer** a unique memory management feature that allows applications to access files on disk in the same way they access dynamic memory through pointers.

➢ With this capability, you can map a view of all or part of a file on disk to a specific range of addresses within your processor address space.
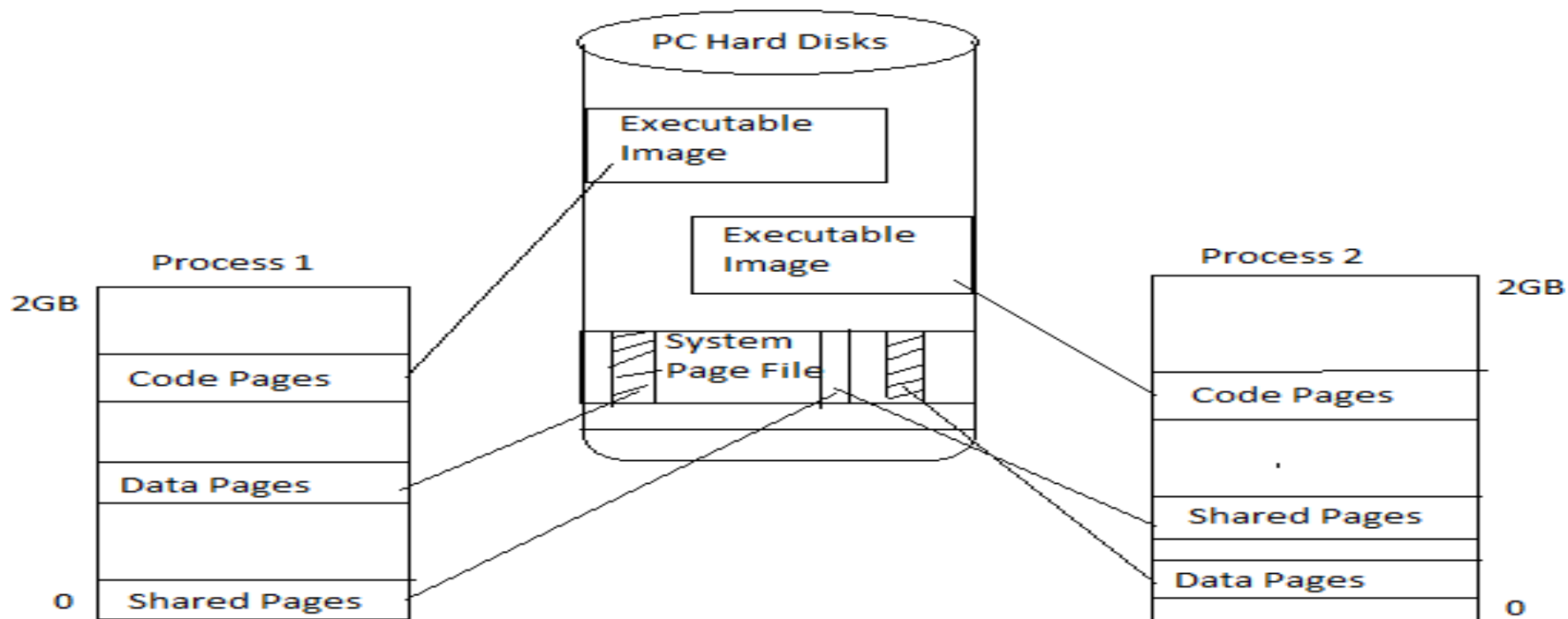
Special Purpose File Systems

➢ The most familiar file systems make use of an underlying data storage device that offers access to an array of fixed-size blocks, sometimes called **sectors**.

➢ The file system software is responsible for organizing these sectors into files and directories and keeping track of **which sectors belong to which files** and which are not being used.

  ✓ File systems typically have directories that associate file names with files, usually by connecting the file name to an index into a file allocation table of some sort, such as the FAT in an MS-DOS file system, or an inode in a UNIX-like file system.

  ✓ **Disk File System** – FAT, NTFS,ext2, ISO9660

  ✓ **Database File System** – File, Topic, another structure, and SQL, WinFS.

  ✓ **Transactional File System** – Transactions Sending and Receiving.

  ✓ **Special Purpose File System(utility)** – Dynamically arrangement, UNIX, processes.

Naming

- ➤ **A filename** (also written as two words, file name) is a name used to **uniquely identify a computer file stored in file systems.**

- ➤ Different file systems impose different restrictions on file name lengths and the allowed characters within filenames.

- ➤ Filenames consist of **long filenames, foreign letters, commas, dots, and space characters** as they appear in software displaying filenames.

- ❑ **Filename may include one or more of these components:**
    - ✓ **Host (or server)** – network device that contains the file.
    - ✓ **Device (or drive)** – hardware device or drive.
    - ✓ **Directory (or path**) – directory tree.
    - ✓ **File –** the base name of the file.
    - ✓ **Type (or format)** – the content type of the file.
    - ✓ **Version** – revision or generation number of the file.

- ➤ The components required to identify a file vary across operating systems, as does the syntax and format for a valid filename. **Example: c:\directory\mufile.txt**

Searching

➢ **Searching** is just trying to find the information you need.

➢ Searching a file means finding a file where the file is stored in computer memory.

❑ **Searching can be done in two ways:**
  - ✓ Linear search
  - ✓ Binary search

❑ **Linear Search:** This is the simplest method of searching.

➢ In this method, the element to be found is sequentially searched in the list.

➢ This method can be applied to a sorted or unsorted list.

❑ **Binary Search:** Binary search method is very fast and efficient.

➢ This method requires that the list of elements be in sorted order.

➢ In this method, to search an element we compare it with the element present at the center of the list.

➢ If it matches then the search is successful.

File access method

➢ **File access methods** define **how data is accessed and modified within a file**.

➢ The file access method is a way of accessing and manipulating data stored in a file.

➢ It determines how data is read and written in computer storage devices.

➢ There are different file access methods with their own set of strengths and limitations.

❑ **The three main types of file access methods are :**

❑ *Sequential access:* **Sequential access** reads and writes data in a linear order.

❑ *Direct/Random access:* **Direct access** involves accessing data directly by its physical location in the file. **Random access** allows direct access to specific data within the file.

❑ *Indexed Sequential Access:* The **indexed access** method involves accessing files through an index or directory that contains a list of file names and their corresponding locations on the disk.

➢ This method is suitable for applications that need to access files by their names or attributes, such as file managers or search engines.

➢ The **indexed access** method provides a fast and efficient way to locate and access files.

Backup Strategies

➢ In information technology, a backup, or the process of backup, refers to the copying into an archived file of computer data so it may be used to restore the original after a data loss event.

❑ **Backup has two distinct purposes:**

    ✓ **The primary purpose** is to recover data after its loss, be it by data deletion or corruption.

    ✓ **The secondary purpose** of backups is to recover data from an earlier time, according to a user-defined data retention policy, typically configured within a backup application for how long copies of data are required.

➢ Data backup is an essential part of data center operations, but it's important to understand what makes a backup strategy successful.

➢ Most people say that it is necessary to have a second copy of data in case the original copy fails.

Cont. ...

➢ **A good backup strategy** is going to create that **second copy**, but it is more crucial that, when file recovery is needed, the data can be found quickly.
- ✓ CD's or DVD's
- ✓ Flash memory
- ✓ Hard Disk Drive
- ✓ Backup software
- ✓ Cloud storage
- ✓ Compression
- ✓ Duplication
- ✓ Cache etc.

:

End of Chapter Six

Thank You!!

Questions ?