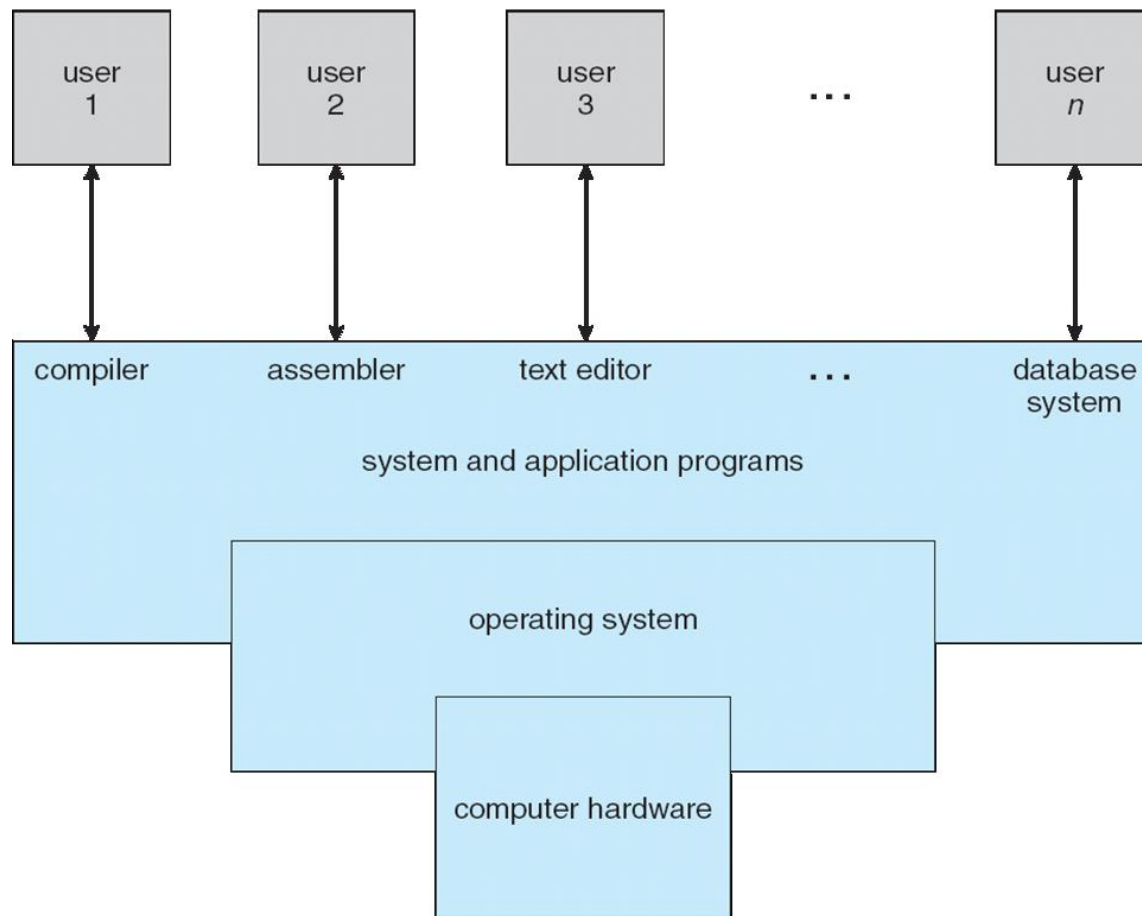


# Operating System

# What is operating system

- ❖ An operating system is a program on which application programs are executed and acts as an communication bridge (interface) between the user and the computer hardware.
- ❖ An operating system is a software which performs all the basic tasks like
  - ❖ File Management
  - ❖ Memory Management
  - ❖ Process Management
  - ❖ Handling Input and Output
  - ❖ Controlling Peripheral devices such as disk drives and printers.



# Services/Functions

## **Program Execution: Introduction**

- ❖ Operating systems handle many kinds of activities from user programs to system programs like Printer Spooler, Name Servers, File Server, etc.
  - ❖ Each of these activities is encapsulated as a process.
  - ❖ A process includes the complete execution context (Code to Execute, Data to Manipulate, Registers, OS Resources in use).
-

## **Program Execution: Activities**

- ❖ Following are the major activities of an operating system with respect to program management:
    - ❖ Loads a program into memory.
    - ❖ Executes the program.
    - ❖ Handles program's execution.
    - ❖ Provides a mechanism for process synchronization.
    - ❖ Provides a mechanism for process communication.
    - ❖ Provides a mechanism for deadlock handling.
-

## I/O Operation

- ❖ An I/O subsystem comprises of I/O devices and their corresponding driver software.
- ❖ Drivers hide the complexity of specific hardware devices from the users.
- ❖ An OS manages the communication between user and device drivers.
- ❖ I/O operation means read or write operation with any file or any specific I/O device.
- ❖ Operating system provides the access to the required I/O device when required.

## **File System Manipulation (1/2)**

### **File System Manipulation (1/2)**

- ❖ A file represents a collection of related information.
  - ❖ Computers can store files on the disk (secondary storage), for long-term storage purpose.
  - ❖ Examples of storage media include Magnetic Tape, Magnetic Disk and Optical Disk drives like CD, DVD.
  - ❖ Each of these media has its own characteristics like Speed, Capacity, Data Transfer Rate and Data Access methods.
-



## File System Manipulation (2/2)

- ❖ A file system is normally organized into directories for easy navigation and usage.
- ❖ These directories may contain files and other directions.
- ❖ Following are the major activities of an operating system with respect to file management:
  - Program needs to read a file or write a file.
  - The OS gives the permission to the program for operation on file.
  - Permission varies from read-only, read-write, denied and so on.
  - OS provides an interface to the user to create/delete files.
  - OS provides an interface to the user to create/delete directories.
  - OS provides an interface to create the backup of file system.

## Communication

- ❖ In case of distributed systems which are a collection of processors that do not share memory, peripheral devices, or a clock, the operating system manages communications between all the processes.
- ❖ Multiple processes communicate with one another through communication lines in the network.
- ❖ The OS handles routing and connection strategies, and the problems of conflict and security.

## Communication

- ❖ Following are the major activities of an operating system with respect to communication:
  - Two processes often require data to be transferred between them
  - Both the processes can be on one computer or on different computers, but are connected through a computer network.
  - Communication may be implemented by two methods, either by Shared Memory or by Message Passing.

## Error handling

- ❖ Errors can occur anytime and anywhere.
  - ❖ An error may occur in CPU, in I/O devices or in the memory hardware.
  - ❖ Following are the major activities of an operating system with respect to error handling:
    - The OS constantly checks for possible errors.
    - The OS takes an appropriate action to ensure correct and consistent computing.
-

# **Resource Management**

- ❖ In case of multi-user or multi-tasking environment, resources such as main memory, CPU cycles and files storage are to be allocated to each user or job.
  - ❖ Following are the major activities of an operating system with respect to resource management:
    - The OS manages all kinds of resources using schedulers.
    - CPU scheduling algorithms are used for better utilization of CPU.
-

# **Protection**

- ❖ Considering a computer system having multiple users and concurrent execution of multiple processes, the various processes must be protected from each other's activities.
- ❖ Protection refers to a mechanism or a way to control the access of programs, processes, or users to the resources defined by a computer system.
- ❖ Following are the major activities of an operating system with respect to protection:
  - The OS ensures that all access to system resources is controlled.
  - The OS ensures that external I/O devices are protected from invalid access attempts.
  - The OS provides authentication features for each user by means of passwords.

# Types of Operating System

## **Types of Operating System**

1. Batch OS
2. Time-Sharing OS
3. Distributed OS
4. Network OS
5. Real-Time OS

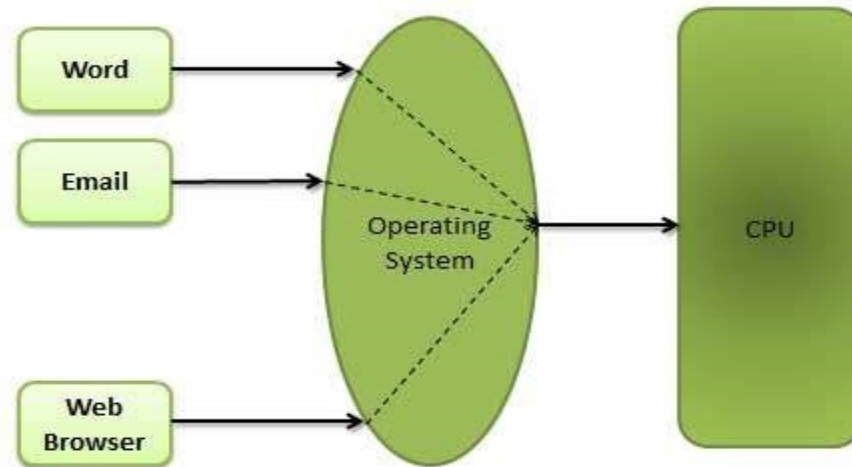


# Multitasking

Multitasking is when multiple jobs are executed by the CPU simultaneously by switching between them. Switches occur so frequently that the users may interact with each program while it is running. An OS does the following activities related to multitasking –

- ▣ The user gives instructions to the operating system or to a program directly, and receives an immediate response.
- ▣ The OS handles multitasking in the way that it can handle multiple operations/executes multiple programs at a time.
- ▣ Multitasking Operating Systems are also known as Time-sharing systems.
- ▣ These Operating Systems were developed to provide interactive use of a computer system at a reasonable cost.
- ▣ A time-shared operating system uses the concept of CPU scheduling and multiprogramming to provide each user with a small portion of a time-shared CPU.
- ▣ Each user has at least one separate program in memory.





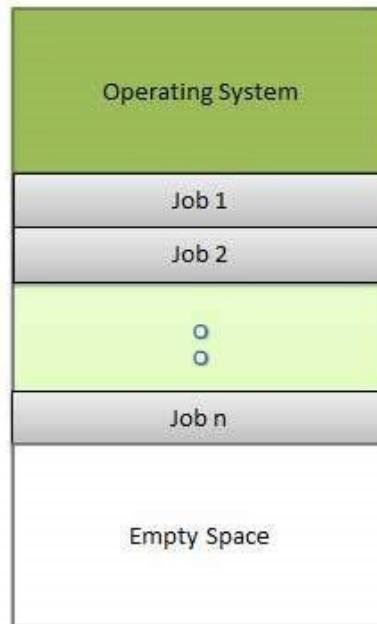
- A program that is loaded into memory and is executing is commonly referred to as a **process**.
- When a process executes, it typically executes for only a very short time before it either finishes or needs to perform I/O.
- Since interactive I/O typically runs at slower speeds, it may take a long time to complete. During this time, a CPU can be utilized by another process.
- The operating system allows the users to share the computer simultaneously. Since each action or command in a time-shared system tends to be short, only a little CPU time is needed for each user.

# Multiprogramming

Sharing the processor, when two or more programs reside in memory at the same time, is referred as **multiprogramming**. Multiprogramming assumes a single shared processor. Multiprogramming increases CPU utilization by organizing jobs so that the CPU always has one to execute.

The following figure shows the memory layout for a multiprogramming system.

The following figure shows the memory layout for a multiprogramming system.



An OS does the following activities related to multiprogramming.

- The operating system keeps several jobs in memory at a time.
- This set of jobs is a subset of the jobs kept in the job pool.
- The operating system picks and begins to execute one of the jobs in the memory.

## Advantages

- High and efficient CPU utilization.
- User feels that many programs are allotted CPU almost simultaneously.

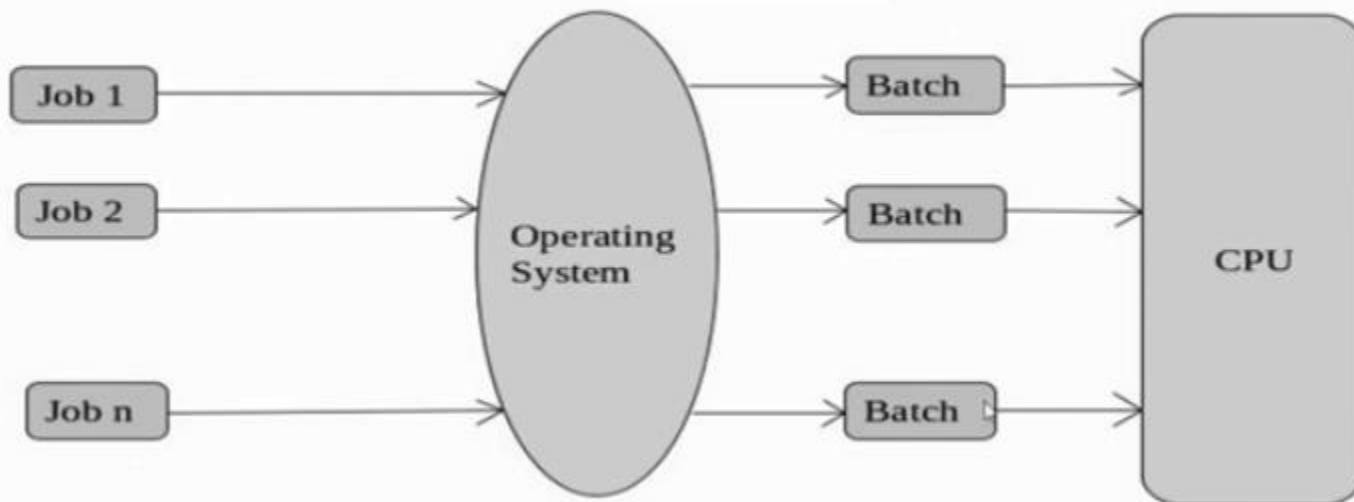
## Disadvantages

- CPU scheduling is required.
- To accommodate many jobs in memory, memory management is required.

# 1. Batch Operating System

- ❖ The users of a batch operating system do not interact with the computer directly.
- ❖ Each user prepares his job on an off-line device like punch cards and submits it to the computer operator.
- ❖ To speed up processing, jobs with similar needs are batched together and run as a group.
- ❖ The programmers leave their programs with the operator and the operator then sorts the programs with similar requirements into batches.

## 1. Batch Operating System



# 1. Batch OS: Advantages/ Disadvantages

## ❖ **Advantages of Batch Operating System:**

- It is very difficult to guess or know the time required for any job to complete. Processors of the batch systems know how long the job would be when it is in queue
- Multiple users can share the batch systems
- The idle time for the batch system is very less
- It is easy to manage large work repeatedly in batch systems

## ❖ **Disadvantages of Batch Operating System:**

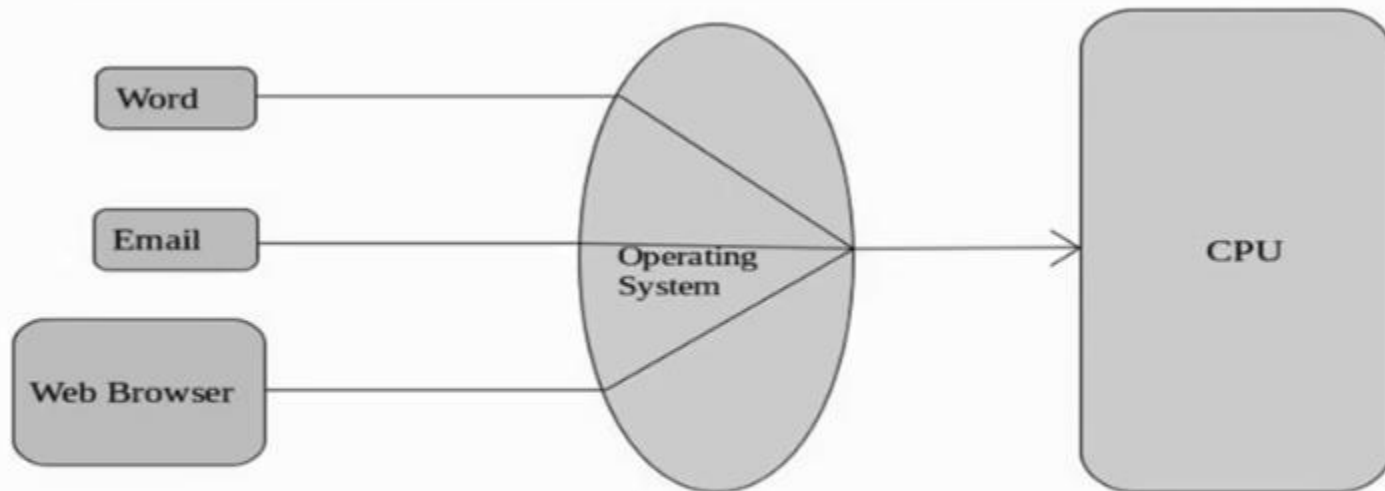
- The computer operators should be well known with batch systems
- Batch systems are hard to debug
- It is sometimes costly
- The other jobs will have to wait for an unknown time if any job fails

## 2. Time-Sharing Operating Systems

- ❖ Each task is given some time to execute so that all the tasks work smoothly.
  - ❖ Each user gets the time of CPU as they use a single system.
  - ❖ These systems are also known as Multitasking Systems.
  - ❖ The task can be from a single user or different users also.
  - ❖ The time that each task gets to execute is called quantum.
  - ❖ After this time interval is over OS switches over to the next task.
-



## 2. Time-Sharing Operating Systems



❖ **Examples of Time-Sharing OSs are:**

- Multics, Unix, etc.

## 2. Time-Sharing OS: Advantages/ Disadvantages

### ❖ **Advantages:**

- Each task gets an equal opportunity
- Fewer chances of duplication of software
- CPU idle time can be reduced

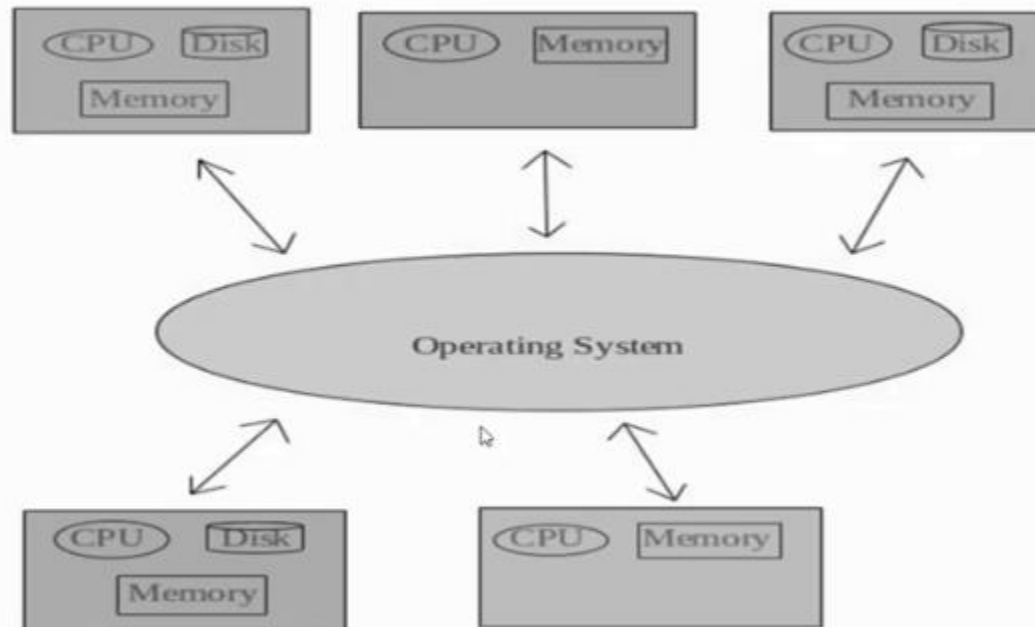
### ❖ **Disadvantages:**

- Reliability problem
- One must have to take care of the security and integrity of user programs and data
- Data communication problem

## 3. Distributed Operating System

- ❖ Distributed systems use multiple central processors to serve multiple real-time applications and multiple users.
- ❖ Data processing jobs are distributed among the processors accordingly.
- ❖ Various autonomous interconnected computers communicate with each other using a shared communication network.
- ❖ Independent systems possess their own memory unit and CPU.
- ❖ These are referred to as loosely coupled systems or distributed systems.
- ❖ These system's processors differ in size and function.
- ❖ Remote access is enabled within the devices connected in that network.

### 3. Distributed Operating System



### 3. Distributed OS: Advantages

- ❖ Failure of one will not affect the other network communication, because all the systems are independent from each other.
- ❖ Electronic mail increases the data exchange speed.
- ❖ Since resources are being shared, computation is highly fast and durable.
- ❖ Load on host computer reduces.
- ❖ These systems are easily scalable as many systems can be easily added to the network.
- ❖ Delay in data processing reduces.

### **3. Distributed OS: Disadvantages**

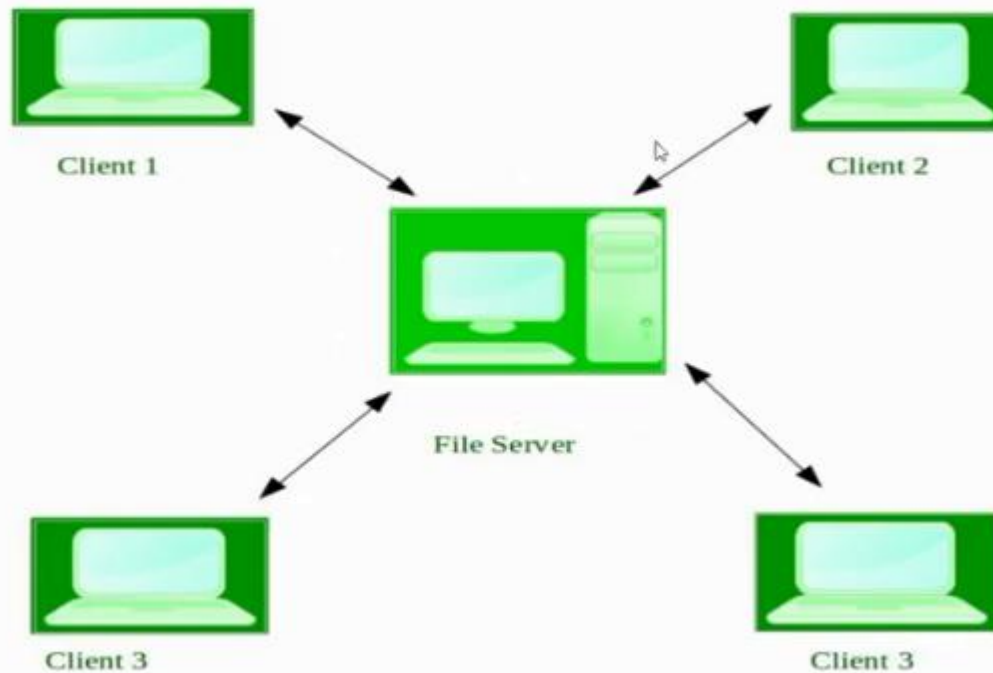
- ❖ Failure of the main network will stop the entire communication
- ❖ To establish distributed systems the language which is used are not well defined yet
- ❖ These types of systems are not readily available as they are very expensive. Not only that the underlying software is highly complex and not understood well yet



## 4. Network Operating System

- ❖ A Network Operating System (NOS) runs on a server and provides the server the capability to manage data, users, groups, security, applications, and other networking functions.
- ❖ The primary purpose of the network operating system is to allow shared file and printer access among multiple computers in a network, typically a local area network (LAN), a private network or to other networks.
- ❖ Examples of network operating systems include Microsoft Windows Server 2003, Microsoft Windows Server 2008, UNIX, Linux, Mac OS X, Novell NetWare, and BSD.

## 4. Network Operating System





## 4. Network OS: Advantages & Disadvantages


### ❖ **Advantages:**

- Centralized servers are highly stable.
- Security is server managed.
- Upgrades to new technologies and hardware can be easily integrated into the system.
- Remote access to servers is possible from different locations and types of systems.

### ❖ **Disadvantages:**

- High cost of buying and running a server.
- Dependency on a central location for most operations.
- Regular maintenance and updates are required.

## 5. Real Time operating System

- ❖ A real-time system is defined as a data processing system in which the time interval required to process and respond to inputs is so small that it controls the environment.
- ❖ The time taken by the system to respond to an input and display of required updated information is termed as the response time.
- ❖ So in this method, the response time is very less as compared to online processing.
- ❖ Real-time systems are used when there are rigid time requirements on the operation of a processor or the flow of data and real-time systems can be used as a control device in a dedicated application.
- ❖ A real-time operating system must have well-defined, fixed time constraints, otherwise the system will fail.
- ❖ For example, Scientific experiments, medical imaging systems, industrial control systems, weapon systems, robots, air traffic control systems, 

## 5. Real Time operating System: Types

### ❖ **Hard real-time systems:**

- Hard real-time systems guarantee that critical tasks complete on time.
- In hard real-time systems, secondary storage is limited or missing and the data is stored in ROM.
- In these systems, virtual memory is almost never found.

### ❖ **Soft real-time systems:**

- Soft real-time systems are less restrictive.
- A critical real-time task gets priority over other tasks and retains the priority until it completes.
- Soft real-time systems have limited utility than hard real-time systems.
- For example, multimedia, virtual reality, Advanced Scientific Projects like undersea exploration and planetary rovers, etc.

# File system in Operating system

# What is File?


- ❖ A file is a named collection of related information that is recorded on secondary storage such as
  - Magnetic Disks
  - Magnetic Tapes
  - Optical Disks.
- ❖ In general, a file is a sequence of bits, bytes, lines or records whose meaning is defined by the files creator and user.

# File Structure

- ❖ A File Structure should be according to a required format that the operating system can understand.
  - A file has a certain defined structure according to its type.
  - A text file is a sequence of characters organized into lines.
  - A source file is a sequence of procedures and functions.
  - An object file is a sequence of bytes organized into blocks that are understandable by the machine.
  - When operating system defines different file structures, it also contains the code to support these file structure.
  - Unix, MS-DOS support minimum number of file structure



# Files Types

- ❖ File type refers to the ability of the operating system to distinguish different types of file such as **text files**, **source files**, and **binary files** etc.
- ❖ Many operating systems support many types of files.
- ❖ Operating system like **MS-DOS** and **UNIX** have the following types of files:
- ❖ **Ordinary Files:**
  - These files contain user information.
  - These may have Text, Databases or Executable program.
  - The user can apply various operations on such files like **Add**, **Modify**, **Delete** or even **Remove** the entire file. 

# Files Types

## ❖ Directory Files:


- These files contain list of file names and other information related to these files.

## ❖ Special Files:

- These files are also known as device files.
- These files represent physical device like **Disks, Terminals, Printers, Networks, Tapedrive** etc.
- **These files are of two types:**
  - ✓ **Character Special Files:** data is handled character by character as in case of terminals or printers.
  - ✓ **Block Special Files:** data is handled in blocks as in the case of disks and tapes.



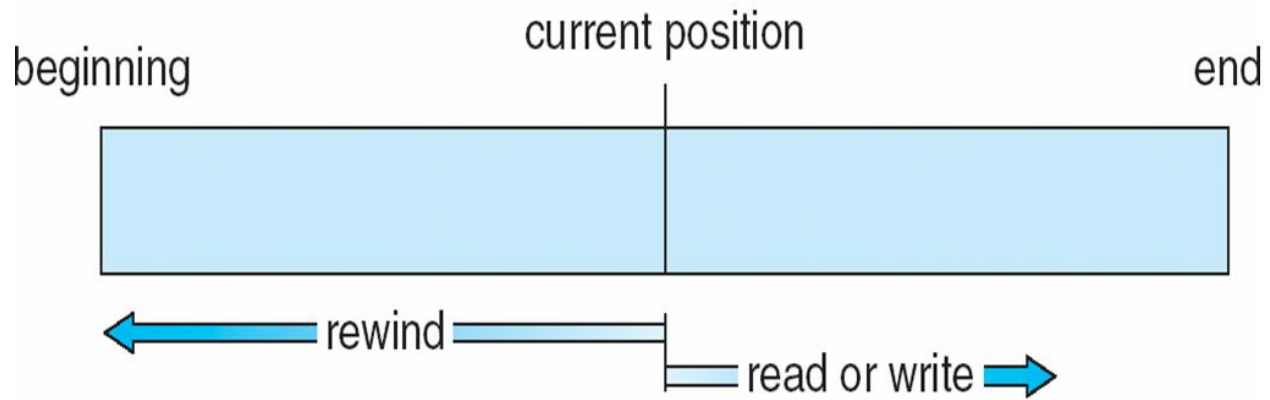
## **File Access Mechanisms**

- ❖ File access mechanism refers to the manner in which the records of a file may be accessed.
- ❖ There are several ways to access files: 
  - Sequential Access
  - Direct/Random Access
  - Indexed Sequential Access

## Sequential Access

- ❖ A sequential access is that in which the records are accessed in some sequence, i.e., the information in the file is processed in order, one record after the other.
- ❖ This access method is the most primitive one.
- ❖ **Example:** Compilers usually access files in this fashion.

# Sequential-access File



- Advantages
- 1 A sequential file is simple to organize and maintain.
- 2 Locating a record requires only the record key.
- 3 It is less expensive than direct access files as comparatively.
- 4 It is the most efficient and economical way to organize file if the number of records to be processed is large.

# Disadvantages

- 1 Transactions must be stored before processing
- 2 Finding data is time consuming as for locating an N record , all preceding records from A to M are searched.
- 3 Data redundancy is high as same file may be required to sort on different key fields.
- 4 Direct access to any record is not possible.

## Direct/Random Access

- ❖ Random access file organization provides, accessing the records directly.
- ❖ Each record has its own address on the file with by the help of which it can be directly accessed for reading or writing.
- ❖ The records need not be in any sequence within the file and they need not be in adjacent locations on the storage medium.

# Advantages

- 1 Immediate access to any is possible.
- 2 Immediate updating of several files as a result of a single transaction.
- 3 Records can also be processed sequentially, if needed.
- 4 Sorting is not necessary prior to processing.
- 5 Direct access is the fastest method of accessing records.

# Disadvantages

- 1 May be less efficient in the use of storage space than sequentially organized file.
- 2 File organization and programming is relatively complex.
- 3 Expensive hardware and programming is relatively costly.



## **Indexed Sequential Access**

- ❖ This mechanism is built up on base of sequential access.
- ❖ An index is created for each file which contains pointers to various blocks.
- ❖ Index is searched sequentially and its pointer is used to access the file directly.

# **Space Allocation**

- ❖ Files are allocated disk spaces by operating system.
- ❖ Operating systems deploy following three main ways to allocate disk space to files:
  - Contiguous Allocation
  - Linked Allocation
  - Indexed Allocation

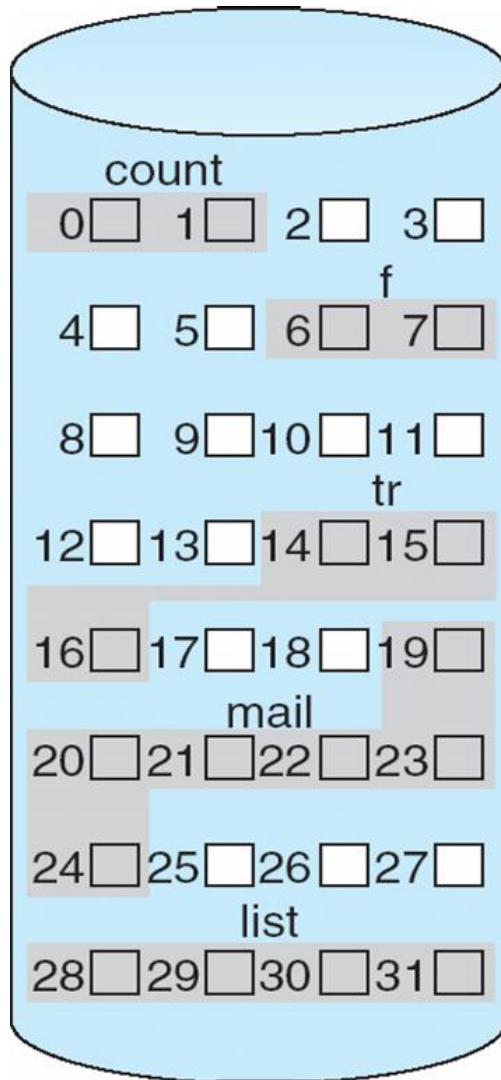
## Contiguous Allocation

- ❖ Each file occupies a contiguous address space on disk.
- ❖ Assigned disk address is in linear order.
- ❖ Easy to implement.
- ❖ External fragmentation is a major issue with this type of allocation technique.

# Allocation Methods - Contiguous

- **Contiguous allocation** – each file occupies set of contiguous blocks
- Blocks are allocated  $b, b+1, b+2, \dots$ 
  - Best performance in most cases
  - Simple – only starting location (block #) and length (number of blocks) are required (directory)
- Easy to implement
- Read performance is great. Only need one seek to locate the first block in the file. The rest is easy.
- Accessing file is easy
  - Minimum disk head movement
  - Sequential and direct access

# Contiguous Allocation of Disk Space



directory

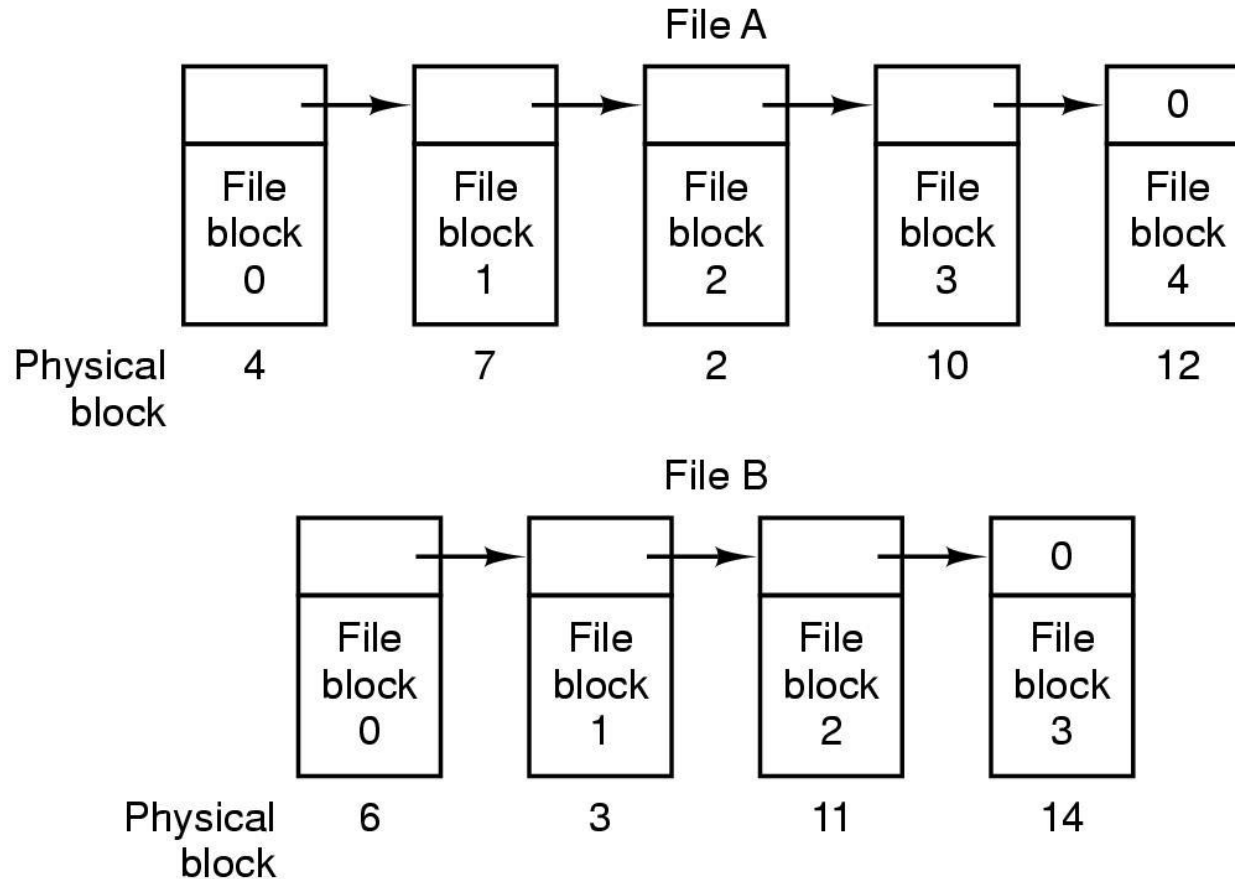
file	start	length
count	0	2
tr	14	3
mail	19	6
list	28	4
f	6	2

- Problems
  - Finding space for file
    - Satisfy the request of size  $n$  from the list of holes
    - External fragmentation
      - Need for **compaction routine**
      - **off-line** (**downtime**) or **on-line**
  - Do not know the file size a priori
    - Terminate and restart
    - Overestimate
    - Copy it in a larger hole
    - Allocate new contiguous space (Extent)

## Linked Allocation

- ❖ Each file carries a list of links to disk blocks.
- ❖ Directory contains link / pointer to first block of a file.
- ❖ No external fragmentation
- ❖ Effectively used in sequential access file.
- ❖ Inefficient in case of direct access file.

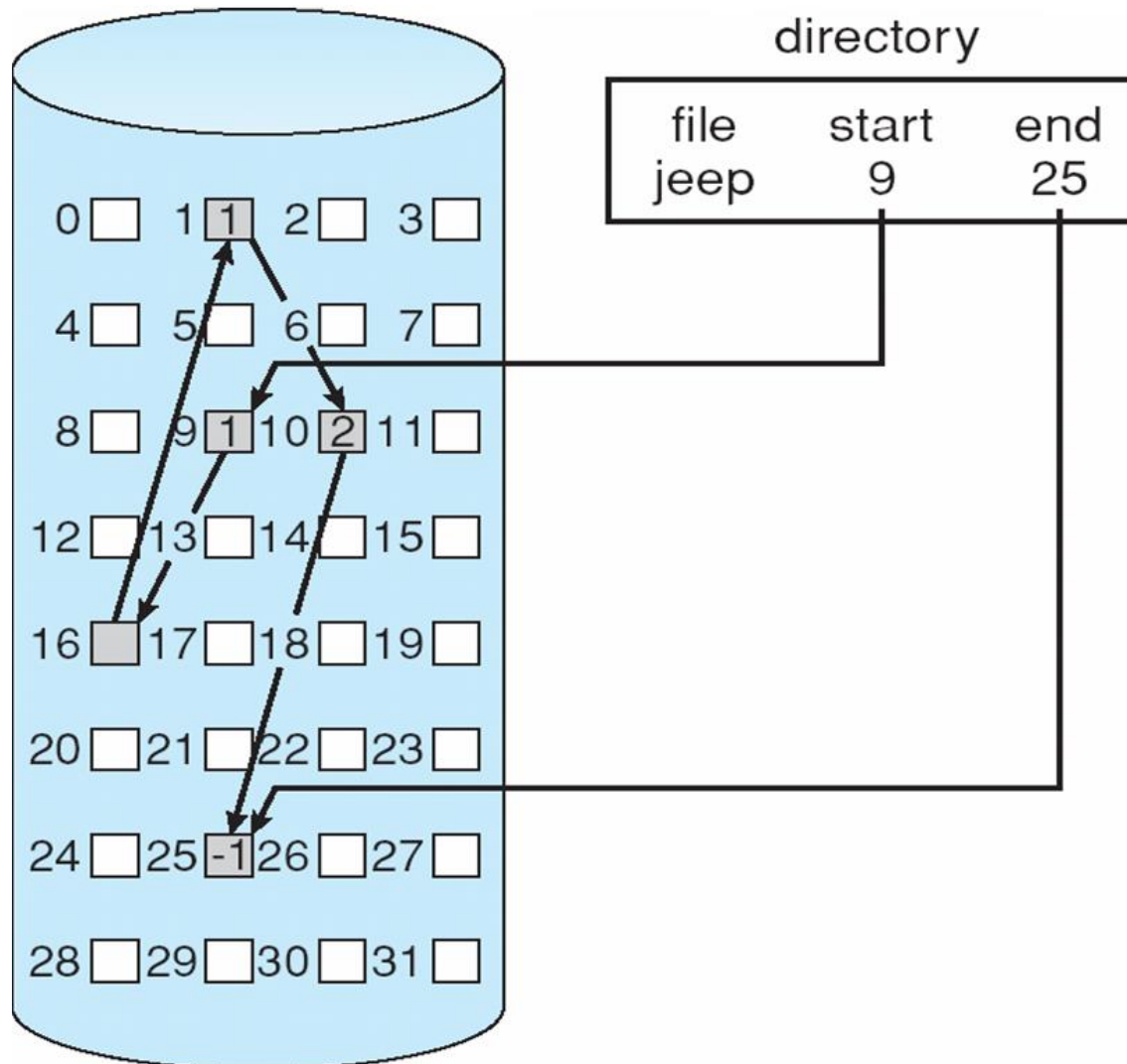
# Linked List Allocation



Storing a file as a linked list of disk blocks.



# Linked Allocation



# Linked Allocation

- Free blocks are arranged from the free space management
- No external fragmentation
- Files can continue to grow

## Disadvantage

1. Effective only for sequential access  
Random/direct access (i-th block) is difficult

2. Space wastage

If block size 512 B

Disk address 4B

Effective size 508B

3. Reliability

Lost/damaged pointer

Bug in the OS software and disk hardware failure

4. Poor performance

### **Solution: Clusters**

- **Improves disk access time (head movement)**
- **Decreases the link space needed for block**
- **Internal fragmentation**

## **Indexed Allocation**

- ❖ Provides solutions to problems of contiguous and linked allocation.
- ❖ An index block is created having all pointers to files.
- ❖ Each file has its own index block which stores the addresses of disk space occupied by the file.
- ❖ Directory contains the addresses of index blocks of files.

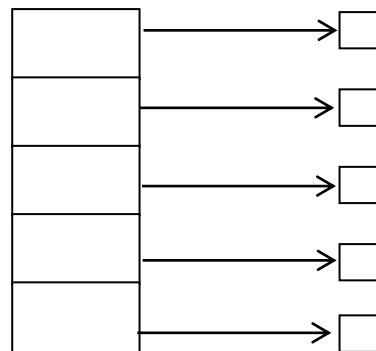
# Allocation Methods - Indexed

- **Indexed allocation**

- Each file has its own **index block**(s) of pointers to its data blocks

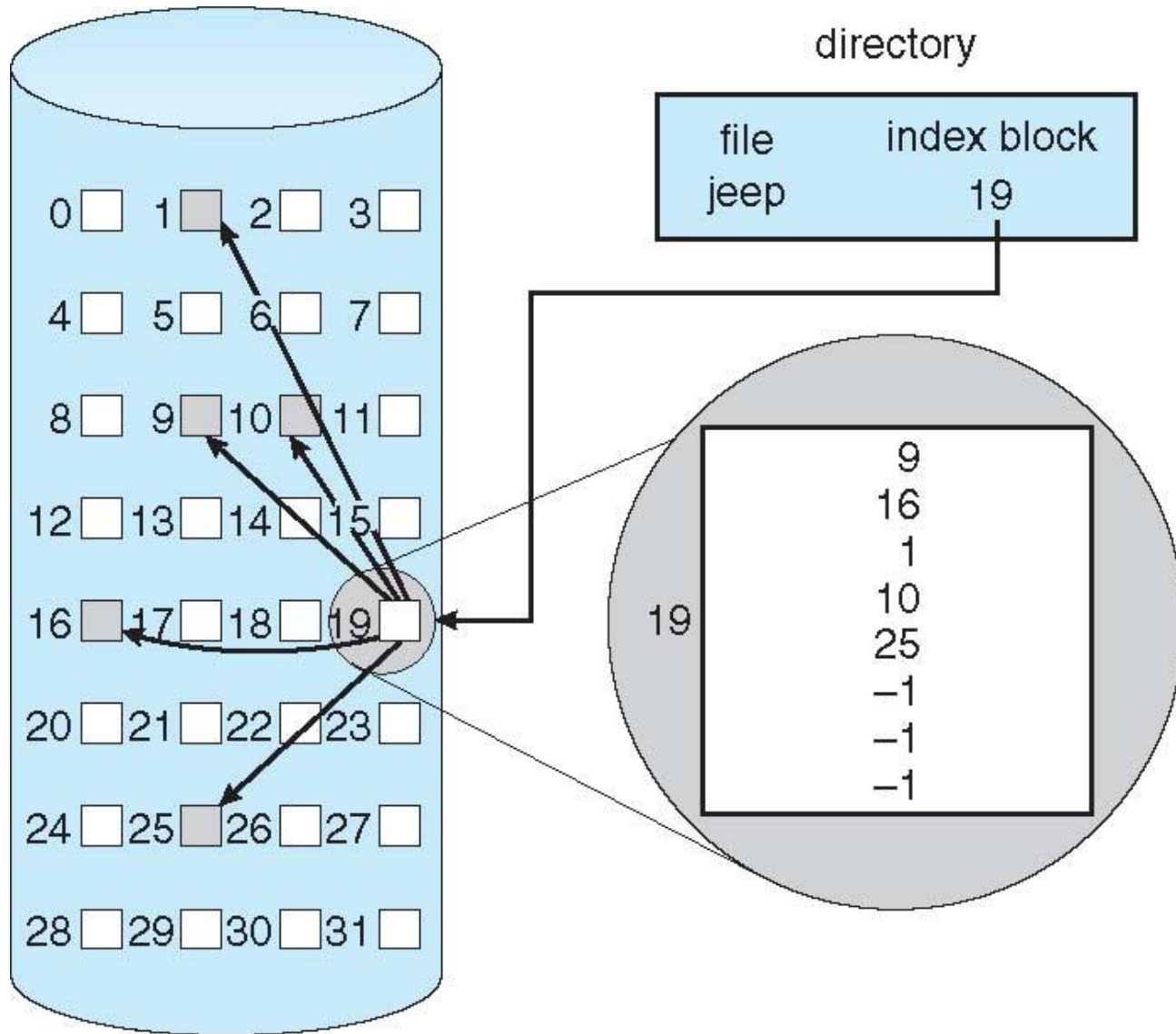
- Directory contains address of the index block

- Logical view



index table

# Example of Indexed Allocation



# Indexed Allocation

- Efficient random access without external fragmentation,
- Size of index block
  - One data block
- Overhead of index block
  - Wastage of space
  - Small sized files

# Advantages

- 1 It provides faster access to records.
- 2 Sequential files can overcome their limitation of going through entire file for a record with use of indexes

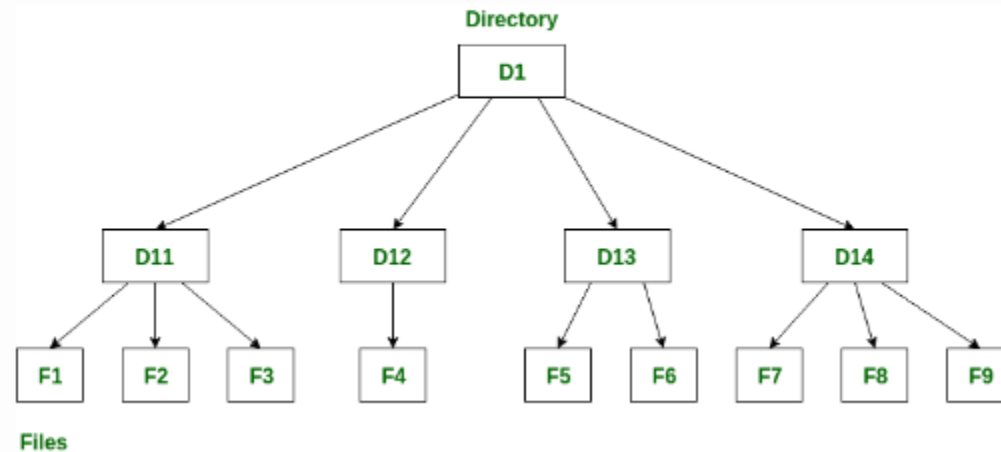
# Disadvantages

- 1 It requires more space to store the indexes.



# Operating System | Structures of Directory

A **directory** is a container that is used to contain folders and file. It organises files and folders into hierarchical manner.

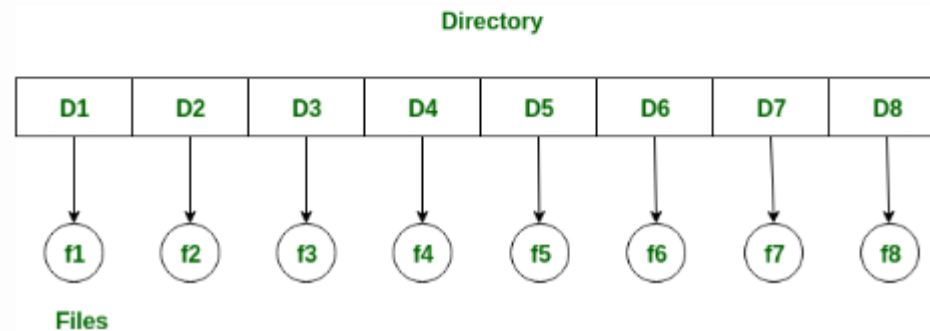


There are several logical structures of directory, these are given as below.

### 1. **Single-level directory –**

Single level directory is simplest directory structure. In it all files are contained in same directory which make it easy to support and understand.

A single level directory has a significant limitation, however, when the number of files increases or when the system has more than one user. Since all the files are in the same directory, they must have the unique name. If two users call their dataset test, then the unique name rule is violated.



## **Advantages:**

- Since it is a single directory, so its implementation is very easy.
- If files are smaller in size, searching will faster.
- The operations like file creation, searching, deletion, updating are very easy in such a directory structure.

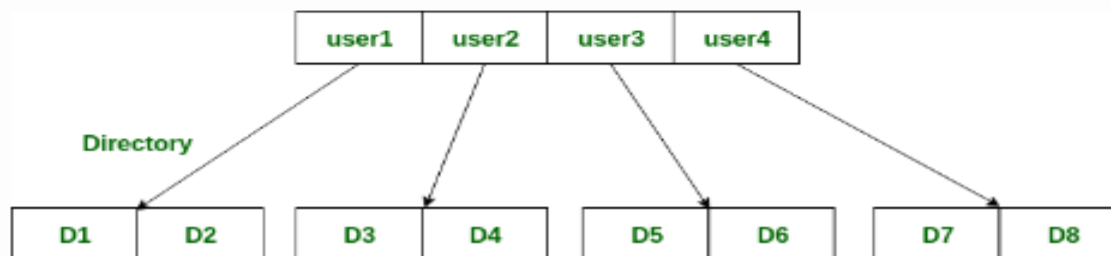
## **Disadvantages:**

- ◊ There may chance of name collision because two files can not have the same name.
- ◊ Searching will become time taking if directory will large.
- ◊ In this can not group the same type of files together.

## 2. Two-level directory –

As we have seen, a single level directory often leads to confusion of files names among different users. the solution to this problem is to create a separate directory for each user.

In the two-level directory structure, each user has there own *user files directory (UFD)*. The UFDs has similar structures, but each lists only the files of a single user. system's *master file directory (MFD)* is searches whenever a new user id=s logged in. The MFD is indexed by username or account number, and each entry points to the UFD for that user.



### **Advantages:**

- We can give full path like /User-name/directory-name/.
- Different users can have same directory as well as file name.
- Searching of files become more easy due to path name and user-grouping.

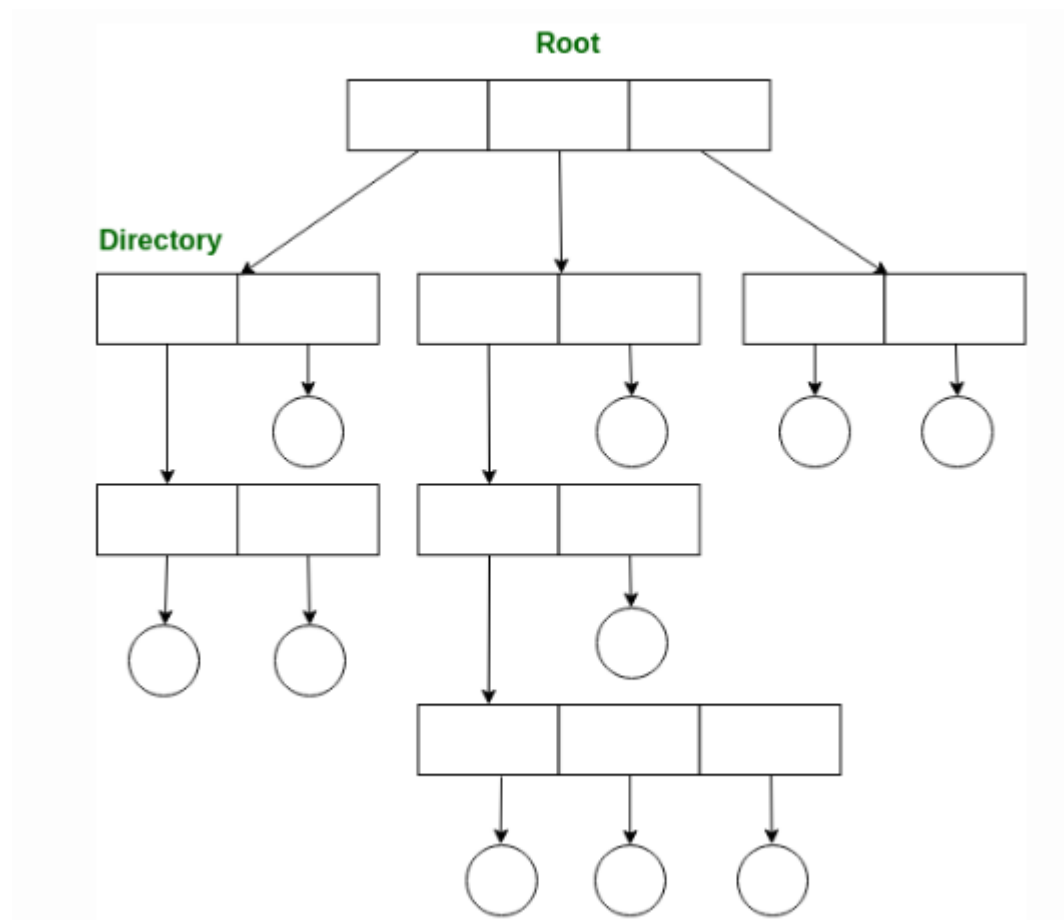
### **Disadvantages:**

- A user is not allowed to share files with other users.
- Still it not very scalable, two files of the same type cannot be grouped together in the same user.

### 3. **Tree-structured directory -**

Once we have seen a two-level directory as a tree of height 2, the natural generalization is to extend the directory structure to a tree of arbitrary height.

This generalization allows the user to create their own subdirectories and to organise their files accordingly.





A tree structure is the most common directory structure. The tree has a root directory, and every file in the system have a unique path.

**Advantages:**

- Very generalize, since full path name can be given.
- Very scalable, the probability of name collision is less.
- Searching becomes very easy, we can use both absolute path as well as relative.

**Disadvantages:**

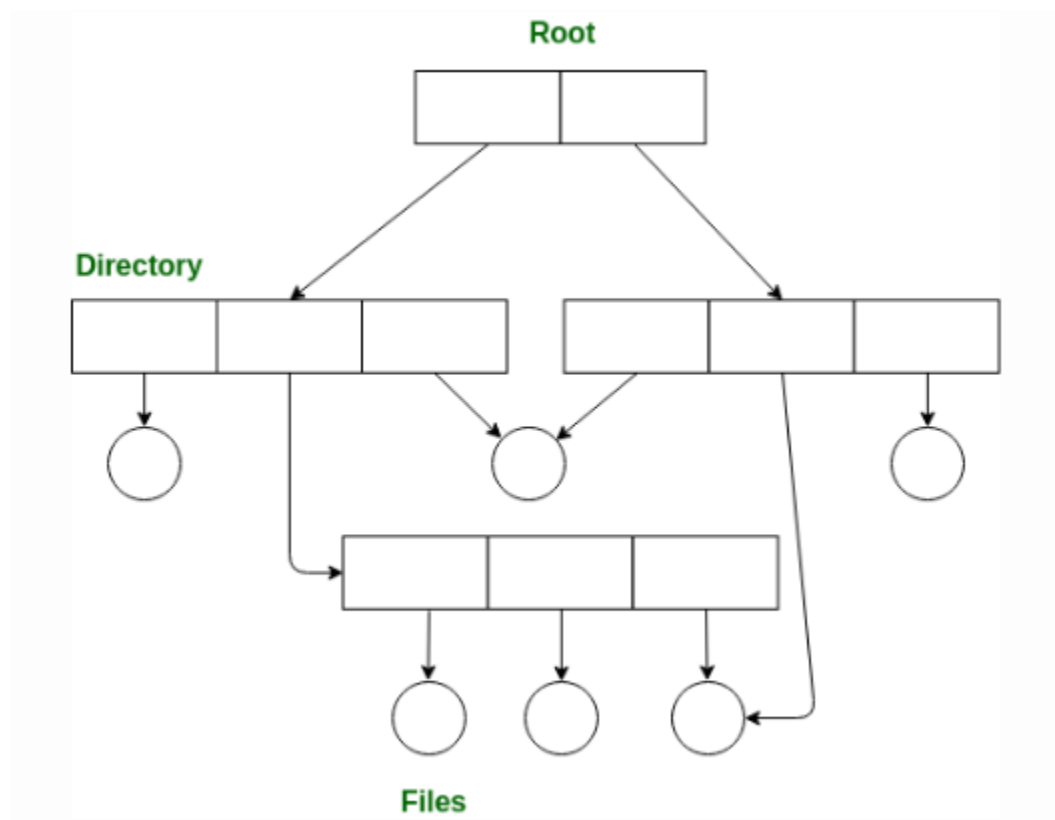
- Every file does not fit into the hierarchical model, files may be saved into multiple directories.
- We can not share files.
- It is inefficient, because accessing a file may go under multiple directories.

## 4. Acyclic graph directory -

An acyclic graph is a graph with no cycle and allows to share subdirectories and files. The same file or subdirectories may be in two different directories. It is a natural generalization of the tree-structured directory.

It is used in the situation like when two programmers are working on a joint project and they need to access files. The associated files are stored in a subdirectory, separated them from other projects and files of other programmers since they are working on a joint project so they want to the subdirectories into there own directories. The common subdirectories should be shared. So here we use Acyclic directories.

It is the point to note that shared file is not the same as copy file if any programmer makes some changes in the subdirectory it will reflect in both subdirectories.



**Advantages:**

- We can share files.
- Searching is easy due to different-different paths.

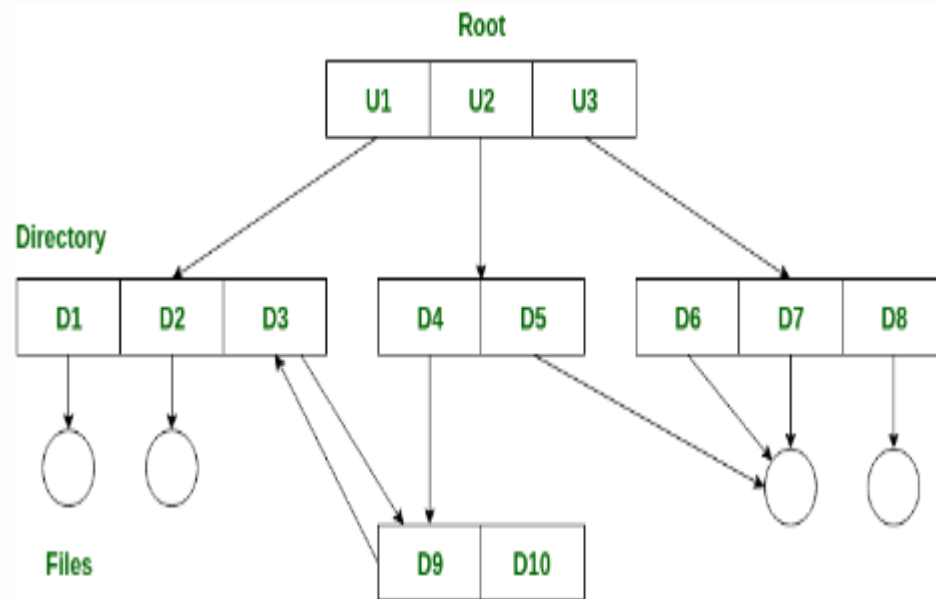
**Disadvantages:**

- We share the files via linking, in case of deleting it may create the problem,
- If the link is softlink then after deleting the file we left with a dangling pointer.
- In case of hardlink, to delete a file we have to delete all the reference associated with it.

## 5. General graph directory structure -

In general graph directory structure, cycles are allowed within a directory structure where multiple directories can be derived from more than one parent directory.

The main problem with this kind of directory structure is to calculate total size or space that have been taken by the files and directories.



**Advantages:**

- It allows cycles.
- It is more flexible than other directories structure.

**Disadvantages:**

- It is more costly than others.
- It needs garbage collection.

## Protection in File System

In computer systems, a lot of user's information is stored, the objective of the operating system is to keep safe the data of the user from the improper access to the system.

Protection can be provided in number of ways. For a single laptop system, we might provide protection by locking the computer in a desk drawer or file cabinet. For multi-user systems, different mechanisms are used for the protection.



### **Types of Access :**

The files which have direct access of the any user have the need of protection. The files which are not accessible to other users doesn't require any kind of protection. The mechanism of the protection provide the facility of the controlled access by just limiting the types of access to the file. Access can be given or not given to any user depends on several factors, one of which is the type of access required. Several different types of operations can be controlled:

- **Read -**  
Reading from a file.
- **Write -**  
Writing or rewriting the file.
- **Execute -**  
Loading the file and after loading the execution process starts.
- **Append -**  
Writing the new information to the already existing file, editing must be end at the end of the existing file.

- **Delete -**

Deleting the file which is of no use and using its space for the another data.

- **List -**

List the name and attributes of the file.

Operations like renaming, editing the existing file, copying; these can also be controlled. There are many protection mechanism. each of them mechanism have different advantages and disadvantages and must be appropriate for the intended application.

## Access Control :

There are different methods used by different users to access any file. The general way of protection is to associate *identity-dependent access* with all the files and directories an list called access-control list (ACL) which specify the names of the users and the types of access associate with each of the user. The main problem with the access list is their length. If we want to allow everyone to read a file, we must list all the users with the read access. This technique has two undesirable consequences:

Constructing such a list may be tedious and unrewarding task, especially if we do not know in advance the list of the users in the system.

Previously, the entry of the any directory is of the fixed size but now it changes to the variable size which results in the complicates space management. These problems can be resolved by use of a condensed version of the access list. To condense the length of the access-control list, many systems recognize three classification of users in connection with each file:

- **Owner -**

Owner is the user who has created the file.

- **Group -**

A group is a set of members who has similar needs and they are sharing the same file.

- **Universe -**

In the system, all other users are under the category called universe.

The most common recent approach is to combine access-control lists with the normal general owner, group, and universe access control scheme. For example: Solaris uses the three categories of access by default but allows access-control lists to be added to