# (ISLM-225)

## Introduction
## to
## Operating System

# Distributed OS and File System

Department of Information Science & Library Management (ISLM)
University of Rajshahi, Rajshahi-6205, Bangladesh

**Unit-1: Operating System Overview:** Introduction to OS. Operating system functions, evaluation of O.S., Different types of O.S.: batch, multi-programmed, time-sharing, real-time, distributed, parallel.

**Unit-2: System Structure:** Computer system operation, I/O structure, storage structure, storage hierarchy, different types of protections, operating system structure (simple, layered, virtual machine), O/S services, system calls.

**Unit-3: Installing and Configuring OS:** Introduction to Installation and Media Types, Performing a Custom OS Installation, Run Levels and the Startup/Shutdown Sequence, Logging In and Out of a Operating System.

**Unit-4: Process Management:** Processes- Concept of processes, process scheduling, operations on processes, co-operating processes, interprocess communication, Threads- overview, benefits of threads, user and kernel threads., CPU scheduling, process synchronization, deadlocks- system model, deadlock characterization, methods for handling deadlocks, deadlock prevention, deadlock avoidance, deadlock detection, recovery from deadlock.

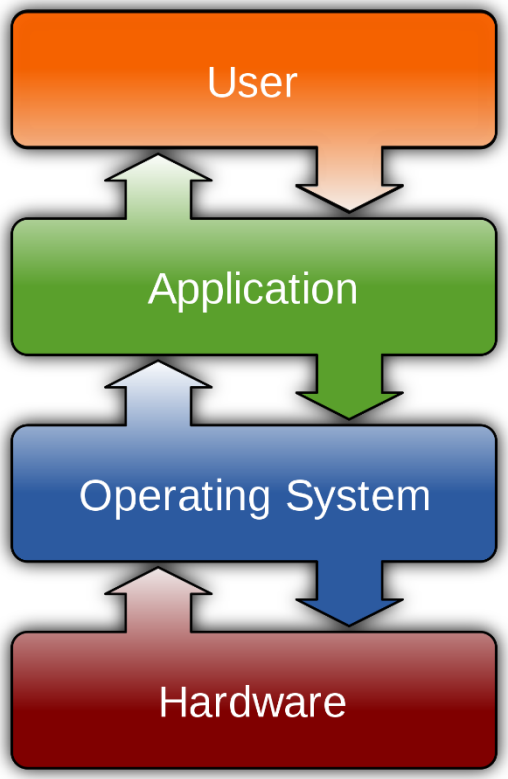
**Unit-5: Storage Management:** Memory Management- background, logical vs. physical address space, swapping, contiguous memory allocation, paging, segmentation, segmentation with paging, Virtual Memory- background, demand paging, performance, page replacement, page replacement algorithms (FCFS, LRU), allocation of frames, thrashing, File Systems, I/O Management, Disk Management.

**Unit-6: Distributed OS and File System:** Motivation, Types of Network-based OS, Network structure, Distributed File System- Background, Naming and transparency, Remote File Access, State full and Stateless services. Distributed Synchronization: Event Ordering, Mutual Exclusion, Atomicity, Concurrency Control, Deadlock Handling, Election algorithm and Reaching agreement.

# What is Operating System (OS)?

•An OS is basically, a program that acts as an interface between the system hardware and the user.

•Moreover, it handles all the interactions between the software and the hardware.



# What is Distributed OS (DOS)?

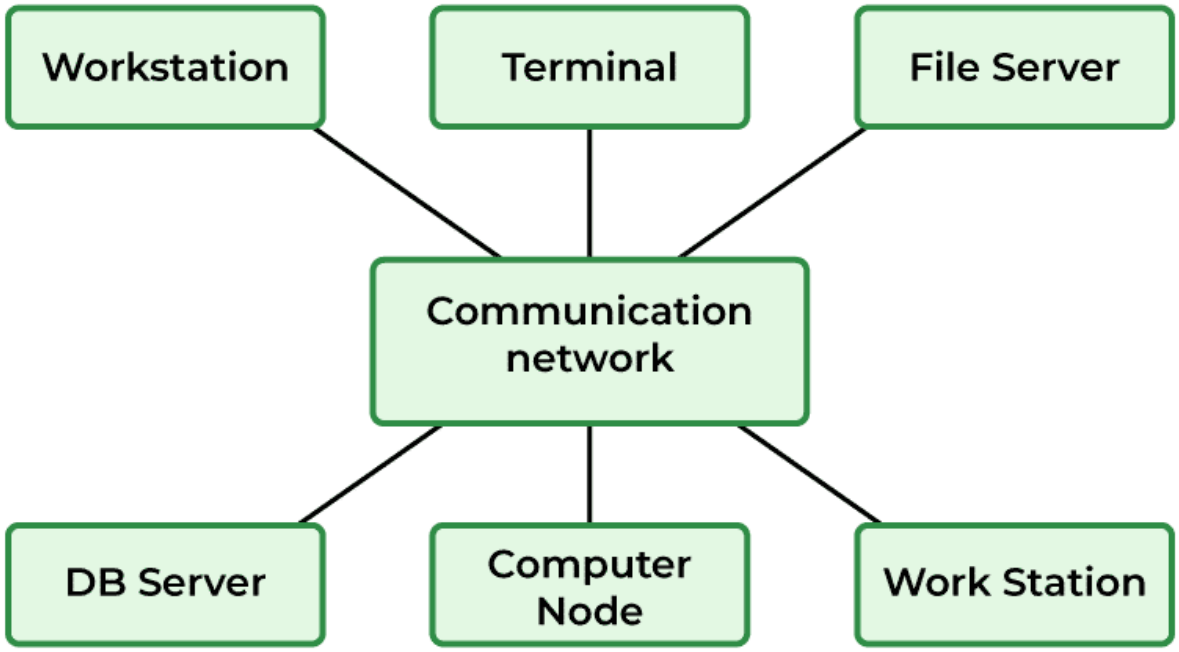
- A distributed operating system is an important type of operating system.
- A distributed operating system is one in which several computer systems connected through a single communication channel.
- Moreover, these systems have their individual processors and memory.
- Furthermore, these processors communicate through high-speed buses or telephone lines.
- These individual systems that connect through a single channel are considered as a single unit.
- We can also call them loosely coupled systems.
- The individual components or systems of the network are nodes.

# What is Distributed OS (DOS)?

- In a Distributed OS, multiple CPUs are utilized, but for end-users, it appears as a typical centralized operating system.
- It enables the sharing of various resources such as CPUs, disks, network interfaces, nodes, and computers across different sites, thereby expanding the available data within the entire system.
- Effective communication channels like high-speed buses and telephone lines connect all processors, each equipped with its own local memory and other neighboring processors.
- Due to its characteristics, a distributed operating system is classified as a loosely coupled system.
- It encompasses multiple computers, nodes, and sites, all interconnected through LAN/WAN lines.

# What is Distributed OS (DOS)?

The diagram below illustrates the structure of a distributed operating system:



# Examples of DOS

**Few examples of a distributed OS are as follows:**

- AIX operating system for IBM RS/6000 computers.
- Solaris operating system for SUN multiprocessor workstations.
- Mach/OS is a multitasking and multithreading UNIX compatible operating system.
- OSF/1 operating system

# Real-life Examples of DOS

**1. Web search**

We Have different web pages, multimedia content, and scanned documents that we need to search. The purpose of web search is to index the content of the web. So to help us, we use different search engines like google, Yahoo, Bing, etc. These search engines use distributed architecture.

**2. Banking system**

Suppose there is a bank whose headquarters is in Dhaka. That bank has branch offices in cities like Rajshahi, Sylhet, Faridpur, and Natore. You can operate your bank by going to any of these branches. How is this possible? It's because whatever changes you make at one branch office are reflected at all branches. This is because of the distributed system.

**3. Massively multiplayer online games**

Nowadays, you can play online games where you can play games with a person sitting in another country in a **real-time** environment. How's it possible? It is because of distributed architecture.

# Types of DOS

There are mainly three types of DOS, they are as follows:

**1. Client-Server Systems**

This strongly connected OS is appropriate for multiprocessors and homogenous multicomputer. It functions as a centralized server, handling and approving all requests originating from client systems.

**2. Peer-to-Peer Systems**

This loosely coupled system is implemented in computer network applications, consisting of multiple processors without shared memories or clocks. Each processor possesses its own local memory, and communication between processors occurs through high-speed buses or telephone lines.

**3. Middleware:**

It facilitates interoperability among applications running on different operating systems. By employing these services, applications can exchange data with each other, ensuring distribution transparency.

# Features/Characteristics of DOS

**1. Resource Sharing**

The main important feature of this system is that it allows users to share resources. Moreover, they can share resources in a secure and controlled manner. Resources can be of any type. For example, some common resources which are shared can be printers, files, data, storage, web pages, etc.

**2. Openness**

This means that the services which the system provides are openly displayed through interfaces. Moreover, these interfaces provide only the syntax of the services. For example, the type of functions, their return types, parameters, etc. These interfaces use Interface Definition Languages (IDL).

**3. Concurrency**

It means that several tasks take place at different nodes of the system simultaneously. Moreover, these tasks can also interact with each other. It results in increasing the efficiency of the system.

# Features/Characteristics of DOS

**4. Scalability**

It refers to the fact that the efficiency of the system should not change when more nodes are added to the system. Moreover, the performance for the system with 100 nodes should be equal to the system with 1000 nodes.

**5. Fault Tolerance**

It means that the user can still work with the system in the case, hardware, or software fails.

**6. Transparency**

It is the most important feature of the system. The main goal of a distributed OS is to hide the fact that the resources are being shared. Furthermore, transparency means that the user should not know that the resources he is using are shared. Moreover, for the user, the system should be a separate individual unit.

# Advantages of Distributed OS

- The load on the system decreases.
- If one system stops it will not affect the other.
- The system shares a workload that makes calculations easy.
- The size of the system can be set according to requirements.

# Disadvantages of Distributed OS

- The cost for set up is more.
- Failure of the main system will affect the whole system.
- Programming is complex.

# Examples of DOS

**There are various examples of the distributed operating system. Some of them are as follows:**

- **Solaris**: It is designed for the SUN multiprocessor workstations
- **OSF/1**: It's compatible with Unix and was designed by the Open Foundation Software Company.
- **Micros:** The MICROS operating system ensures a balanced data load while allocating jobs to all nodes in the system.
- **DYNIX**: It is developed for the Symmetry multiprocessor computers.
- **Locus**: It may be accessed local and remote files at the same time without any location hindrance.
- **Mach:** It allows the multithreading and multitasking features.

# Applications of DOS

- **Network Applications**: DOS is used by many network applications, including the Web, peer-to-peer networks, multiplayer web-based games, and virtual communities.
- **Telecommunication Networks**: DOS is useful in phones and cellular networks. A DOS can be found in networks like the Internet, wireless sensor networks, and routing algorithms.
- **Parallel Computation:** DOS is the basis of systematic computing, which includes cluster computing and grid computing, and a variety of volunteer computing projects.
- **Real-Time Process Control:** The real-time process control system operates with a deadline, and such examples include aircraft control systems.

# Distributed File System (DFS)

- A distributed file system (DFS) is a file system that enables clients to access file storage from multiple hosts through a computer network as if the user was accessing local storage.
- Files are spread across multiple storage servers and in multiple locations, which enables users to share data and storage resources.
- A DFS can be designed so geographically distributed users, such as remote workers and distributed teams, can access and share files remotely as if they were stored locally.

# The challenges associated DFS compared to traditional file systems

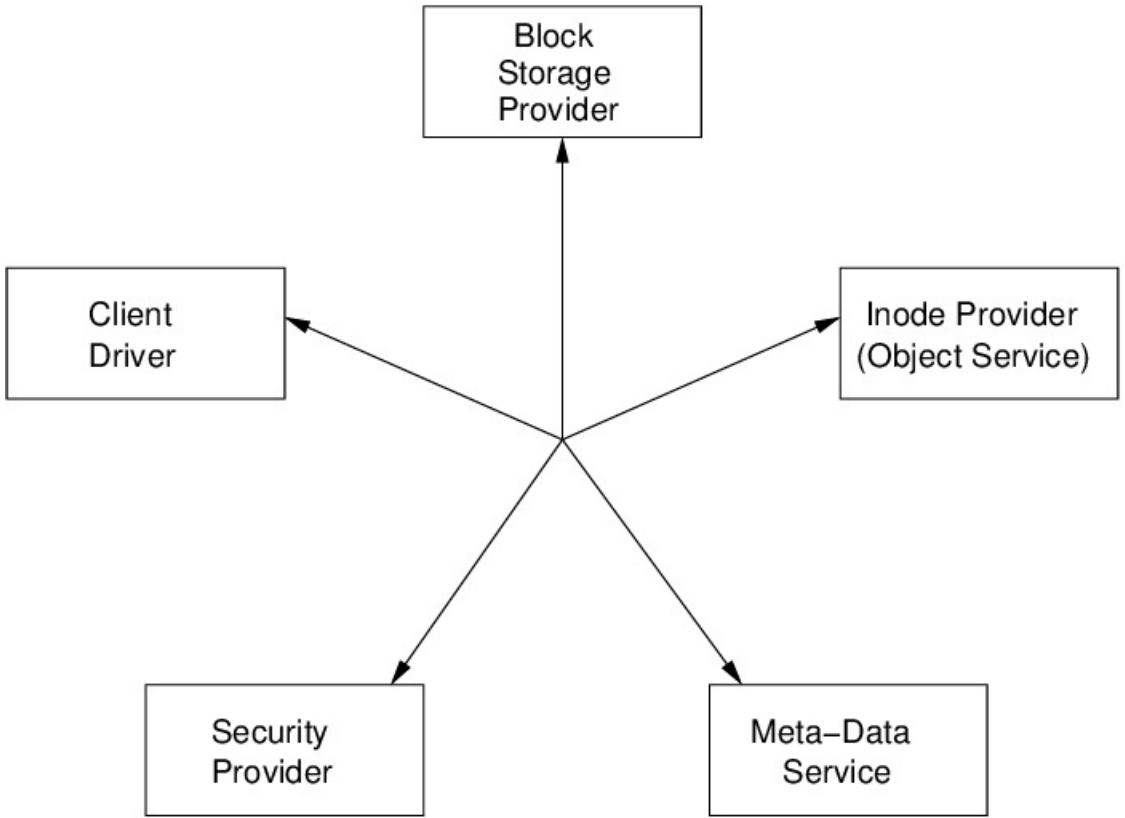
- Data redundancy and inconsistency.
- Difficulty in accessing data.
- Data isolation
- Integrity problems
- Unauthorized access is not restricted.
- It coordinates only physical access.

# Components of DFS

The components of DFS are as follows −

- Block Storage provider
- Client Driver
- Security provider
- Meta- Data Service
- Object service.

These components are pictorially represented below −



# What Are the Different Types of DFS?

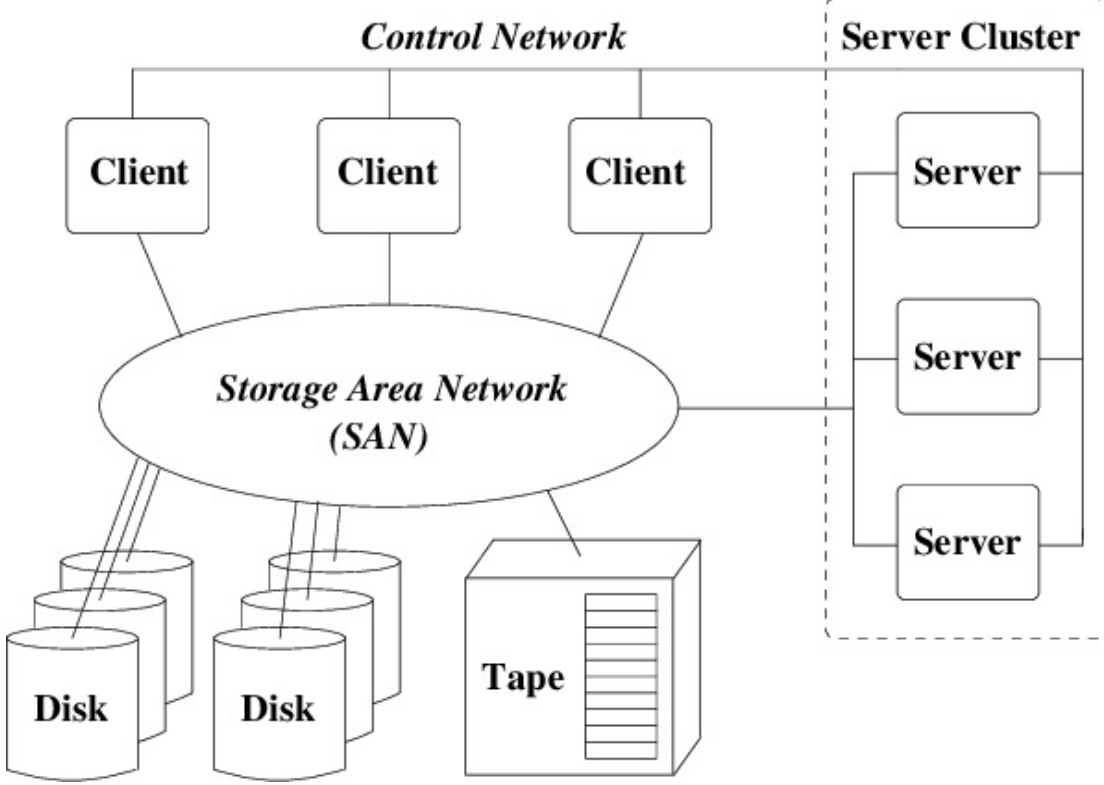
These are the most common DFS implementations:

- Windows Distributed File System
- Network File System (NFS)
- Server Message Block (SMB)
- Google File System (GFS)
- Lustre
- Hadoop Distributed File System (HDFS)
- GlusterFS
- Ceph
- MapR File System

# How a DFS works?

- A DFS clusters together multiple storage nodes and logically distributes data sets across multiple nodes that each have their own computing power and storage.
- The data on a DFS can reside on various types of storage devices, such as solid-state drives and hard disk drives.
- Data sets are replicated onto multiple servers, which enables redundancy to keep data highly available.
- The DFS is located on a collection of servers, mainframes or a cloud environment over a local area network (LAN) so multiple users can access and store unstructured data.
- If organizations need to scale up their infrastructure, they can add more storage nodes to the DFS.

# DFS Architecture




- Clients access data on a DFS using namespaces.
- Organizations can group shared folders into logical namespaces.
- A namespace is the shared group of networked storage on a DFS root.
- These present files to users as one shared folder with multiple subfolders.
- When a user requests a file, the DFS brings up the first available copy of the file.

# Features of DFS

- **Location independence.** Users do not need to be aware of where data is stored. The DFS manages the location and presents files as if they are stored locally.
- **Scalability.** To scale a DFS, organizations can add file servers or storage nodes.
- **High availability.** The DFS should continue to work in the event of a partial failure in the system, such as a node failure or drive crash. A DFS should also create backup copies if there are any failures in the system.
- **Security.** Data should be encrypted at rest and in transit to prevent unauthorized access or data deletion.

# Features of DFS

•**Transparency.** Transparency keeps the details of one file system away from other file systems and users. There are multiple types of transparency in DFS which are as follows:

- ***Structural transparency.*** *Data appears as if it's on a user's device. Users are unable to see how the DFS is configured, such as the number of file servers or storage devices.*
- ***Access transparency.*** *Users can access files that are located locally or remotely. Files can be accessed no matter where the user is, as long as they are logged in to the system. If data is not stored on the same server, users should not be able to tell, and applications for local files should also be able to run on remote files.*
- ***Replication transparency.*** *Replicated files that are located on different nodes of the file system, such as on another storage system, are hidden from other nodes in the system. This enables the system to create multiple copies without affecting performance.*
- ***Naming transparency.*** *Files should not change when moving among storage nodes.*

# Benefits of Using DFS

- **Flexibility in storage management** − In DFS, storage management is very flexible and we can easily modify it according to our need.
- **Load sharing advantage** − Load sharing can be done with optimal results using the DFS. Load sharing is one of the best benefits of DFS.
- **Security Integration** − If we want to implement security then it can be easily done in the DFS.
- **Graphical way of Administration** − Graphical view of administration window is available here, which reduces cost in administration training.
- **High Availability** − High availability is also one of the best benefits of DFS. It keeps all the important data available all the time.

# Mutual Exclusion in Synchronization

•During concurrent execution of processes, processes need to enter the critical section at times for execution.

•It might so happen that because of the execution of multiple processes at once, the values stored in the critical section become inconsistent.

•In other words, the values depend on the sequence of execution of instructions – also known as a race condition.

•The primary task of process synchronization is to get rid of race conditions while executing the critical section.

•This is primarily achieved through mutual exclusion.

# Mutual Exclusion in Synchronization

•Mutual exclusion is a property of process synchronization which states that “no two processes can exist in the critical section at any given point of time”.

•Any process synchronization technique being used must satisfy the property of mutual exclusion, without which it would not be possible to get rid of a race condition.