



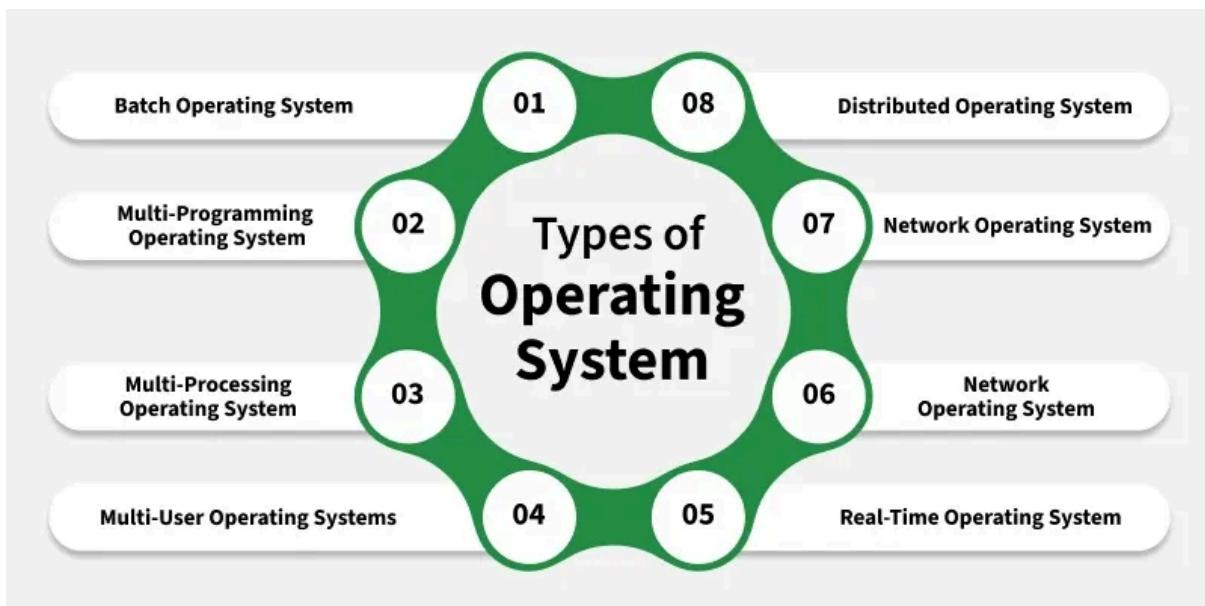
Types of Operating Systems - chapter 1 (part II)

Created @August 25, 2025 12:33 AM

Class IN 3311 Operating Systems

An **operating system** (OS) is software that manages computer hardware and software resources.

It acts as a bridge between users and the computer, ensuring smooth operation. Different types of OS serve different needs some handle one task at a time, while others manage multiple users or real-time processes.



Batch Operating System

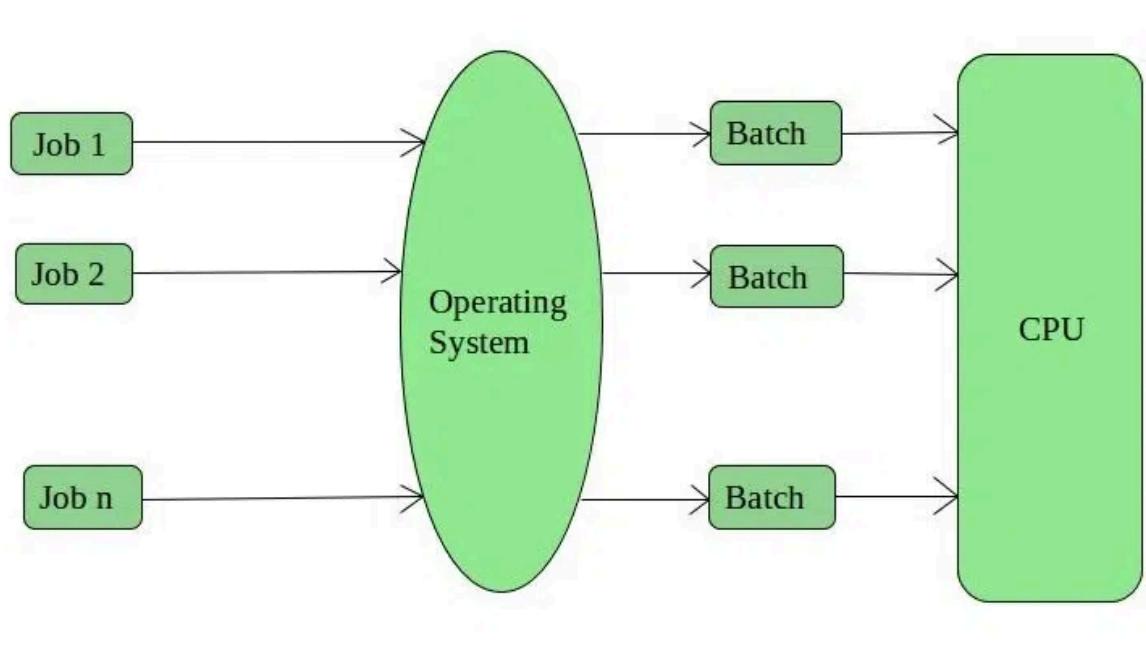
A Batch Operating system is designed to handle large groups of similar jobs efficiently. It does not interact with the computer directly but instead processes

jobs that are grouped by an operator. These jobs are queued and executed one after the other, without user interaction during the process.

What is a Batch-Processing Operating System?

- The batch-processing operating system was very popular in the **1970s**. In a batch operating system, the jobs were performed in batches.
- This means Jobs having similar requirements are grouped and executed as a group to speed up processing.
- Users using batch operating systems do not interact with the computer directly. Each user prepares their job using an offline device for example a punch card and submits it to the computer operator.

Once the programmers have left their programs with the operator, they sort the programs with similar needs into batches.



Advantages and disadvantages of Batch Operating System

Feature	Advantages of Batch Operating System	Disadvantages of Batch Operating System
Job Management	Efficient for multiple users, making it cost-effective.	Unpredictable job completion if one job fails.

System Utilization	Minimizes idle time by processing jobs continuously.	Inefficient CPU utilization when a job is waiting for I/O.
Task Handling	Ideal for handling large, repetitive tasks like payroll and billing.	Not suitable for interactive tasks due to lack of real-time feedback.
Performance	Improves throughput by handling high volumes of jobs at once.	Increased response time due to sequential job processing.

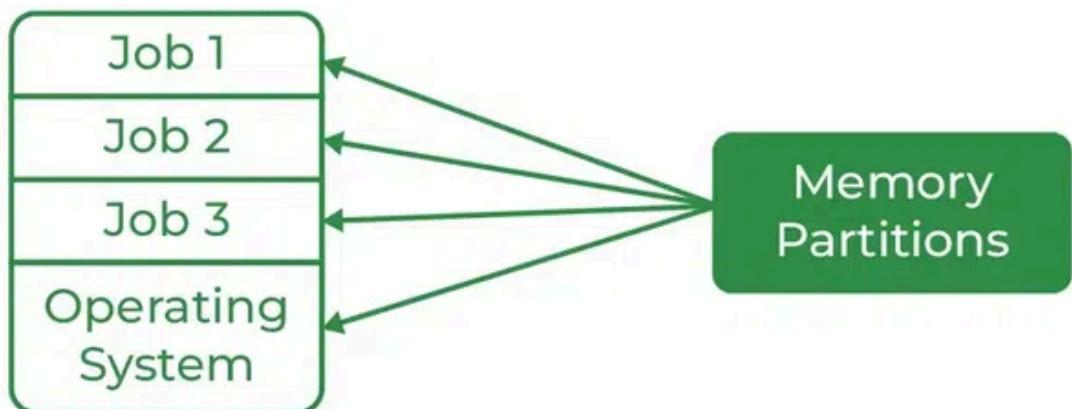
Examples: Payroll systems, Bank statements

Multi-Programming Operating System

In a Multi-Programming operating system, multiple programs run in memory at the same time. The CPU switches between programs, utilising its resources more effectively and improving overall system performance.

In multiprogramming system, multiple programs are to be stored in memory and each program has to be given a specific portion of memory which is known as process. The operating system handles all these process and their states.

Multiprogramming



Advantages of Multi-programming Operating System

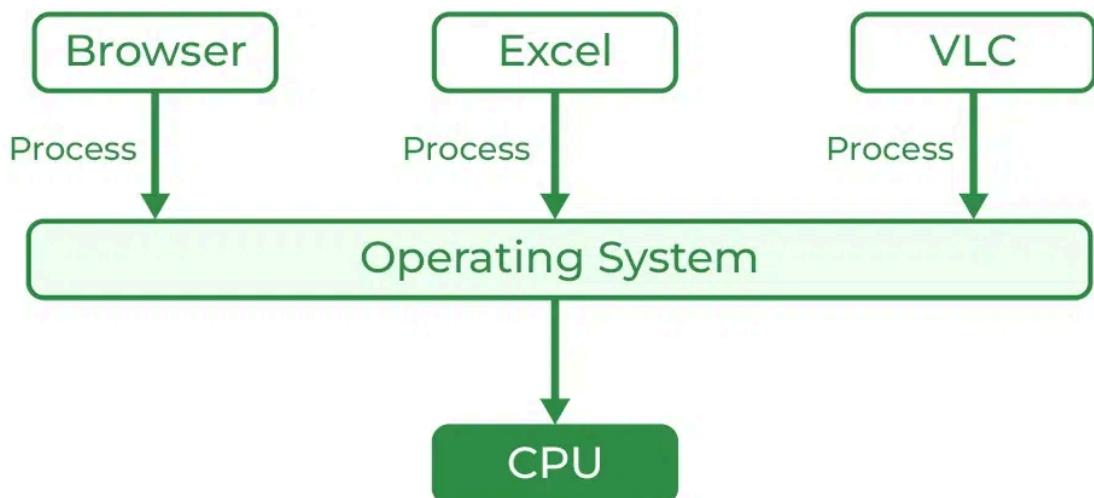
- CPU is better utilized and the overall performance of the system improves.
- It helps in reducing the response time.

Multi-tasking/Time-sharing Operating Systems

A multitasking OS is a type of Multiprogramming system with every process running in a round-robin manner. Each task is given some time to execute so that all the tasks work smoothly. Each user gets the time of the 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. The time that **each task gets to execute is called a quantum**. After this time interval is over, the OS switches over to the next task.

Multitasking



Advantages and Disadvantages of Time-Sharing Operating Systems

Advantages	Disadvantages
Each task gets an equal opportunity.	Reliability problems.
Fewer chances of duplication of software.	Security and integrity of user programs and data must be ensured.
CPU idle time can be reduced.	Data communication problems.
Resource Sharing: Users share CPU, memory, and peripherals, reducing hardware cost and increasing efficiency.	High Overhead: Context switching, scheduling, and multi-user support add overhead.
Improved Productivity: Multiple users work concurrently, reducing waiting time.	Complexity: Requires advanced software to manage multiple users, increasing risk of bugs.

Advantages	Disadvantages
Improved User Experience: Provides real-time, interactive environment better than batch processing.	Security Risks: Multiple users sharing resources increases the chance of breaches, requiring strong access controls.

Example: IBM VM/CMS, TSO (Time Sharing Option) Windows Terminal Services

Multi-Processing Operating System

A Multi-Processing Operating System is a type of Operating System in which more than one CPU is used for the execution of resources. It improves the throughput of the System.

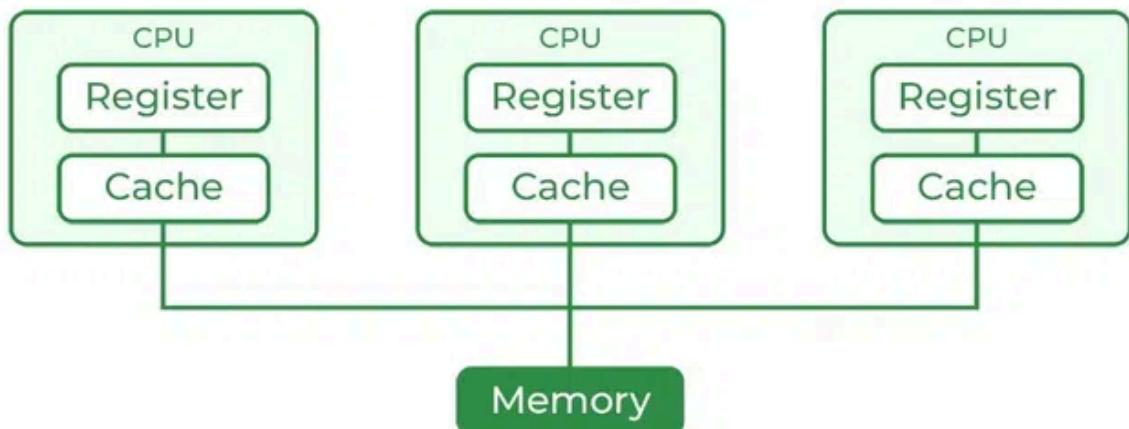
What is Multiprocessing?

- Multiprocessing is a system that has two or more processors. In this, **CPUs** are added to increase the computing speed of the system.
- Because of Multiprocessing, many processes are executed simultaneously. This approach enhances performance by leveraging parallelism to handle more tasks at once.
- Multiprocessing is further classified into two categories: Symmetric Multiprocessing and Asymmetric Multiprocessing.

Asymmetric multiprocessing vs Symmetric multiprocessing

- **Asymmetric multiprocessing:** a specific task is allocated to a particular processor, not every processor can perform all the tasks. Typically, there are slave processors present that execute tasks based on the Master processor (who controls the system) assigning the tasks to the slave processors.
- **Symmetric multiprocessing:** all the processors have equal access to all resources and are capable of executing any task allocated to them. They share a common memory space which helps to communicate processors and data sharing between them.

Multiprocessing



Advantages of a Multi-Processing Operating System

- It increases the throughput of the system as processes can be parallelized.
- As it has several processors, so, if one processor fails, we can proceed with another processor.

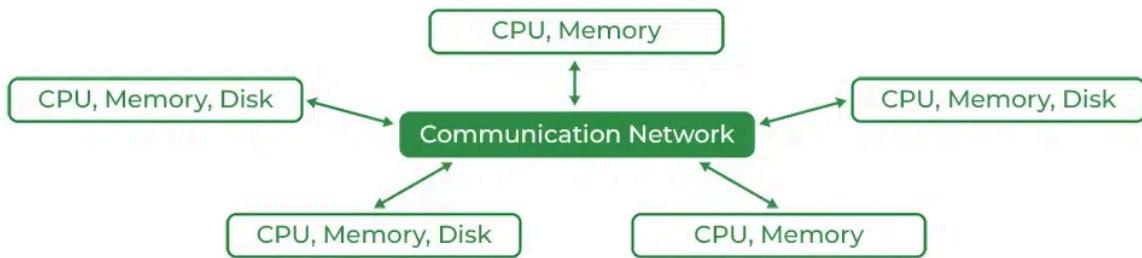
Distributed Operating System

Distributed Operating Systems are a recent advancement in the world of computer technology and are being widely accepted all over the world, and that too, at a great pace.

Various autonomous interconnected computers communicate with each other using a shared communication network. Independent systems possess their own memory unit and CPU.

These systems' processors differ in size and function. The major benefit of working with these types of operating systems is that it is always possible that one user can access the files or software which are not present on his system but on some other system connected within this network, i.e., remote access is enabled within the devices connected to that network.

Architecture of Distributed OS



Advantages and Disadvantages of Distributed Operating Systems

Advantages	Disadvantages
Failure of one system does not affect others, as all systems are independent.	Failure of the main network will stop the entire communication.
Electronic mail increases the speed of data exchange.	The language for establishing distributed systems is not yet well-defined.
Shared resources make computation fast and durable.	These systems are very expensive and not readily available.
Load on the host computer is reduced.	The underlying software is highly complex and not well understood.
Easily scalable – new systems can be added to the network.	—
Reduces delay in data processing.	—

Issues With Distributed Operating Systems

- Networking causes delays in the transfer of data between nodes of a distributed system. Such delays may lead to an inconsistent view of data located in different nodes and make it difficult to know the chronological order in which events occurred in the system.
- Control functions like scheduling, resource allocation and deadlock detection have to be performed in several nodes to achieve computation speedup and provide reliable operation when computers or networking components fail.
- Messages exchanged by processes present in different nodes may travel over public networks and pass through computer systems that are not controlled by the distributed operating system. An intruder may exploit this

feature to tamper with messages or create fake messages to fool the authentication procedure and masquerade as a user of the system.

Example: *LOCUS, MICROS, Amoeba*