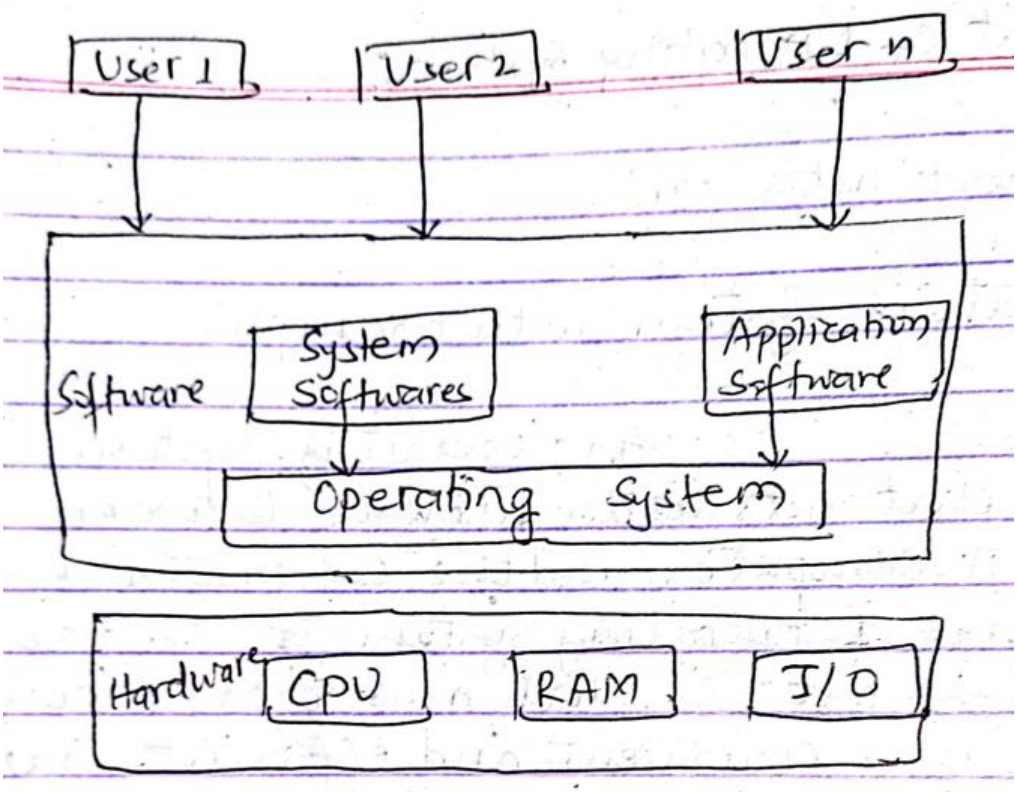
Unit-1 Introduction to Operating System

An operating system (OS) is a fundamental software that manages and controls computer hardware and software resources. It serves as an intermediary between users and the computer's hardware, allowing users to interact with the system through a user-friendly interface. The primary purpose of an operating system is to provide a stable and efficient environment for running applications and executing tasks.

The operating system handles various essential functions, such as managing memory, scheduling tasks for the CPU, handling input and output operations, and providing security measures to protect the system from unauthorized access and malware. It also enables communication between different hardware components and ensures that they work together cohesively. In essence, the operating system is the core software that enables a computer to function and allows users to perform tasks effectively and efficiently.



Functionalities of Operating System:

- **Process Management:** The OS manages all the running programs (processes) on your computer. It decides which process gets to use the CPU, how much time it gets, and handles switching between different processes so that everything runs smoothly.

- **Memory Management:** The OS takes care of the computer's memory, ensuring that each program gets the space it needs to run and preventing one program from interfering with another.

- **File Management:** It helps you organize and store your files effectively. The OS manages files and folders, allowing you to create, delete, copy, and move them around.

- **Device Management:** The OS communicates with various hardware devices like printers, scanners, and keyboards, making sure they work correctly and allowing you to use them without worrying about the technical details.

- **User Interface:** The OS provides a graphical user interface (GUI) that lets you interact with the computer easily. You can click on icons, open applications, and manage files with a mouse or touchpad.

- **Security:** It ensures the security of your computer and data by managing user accounts, permissions, and access rights. The OS also protects against viruses and other malicious software.

- **Networking:** The OS enables your computer to connect to the internet and other devices on a network. It handles data transmission and ensures smooth communication.

- **Interrupt Handling:** The OS deals with sudden events like hardware issues or input from devices, called interrupts. It manages these interruptions and ensures they are appropriately addressed.

- **Time Management:** The OS keeps track of time and maintains a system clock, allowing you to schedule tasks and events accurately.

- **Error Handling:** When something goes wrong, the OS tries to handle errors gracefully. It may display error messages or attempt to recover from a crash to avoid data loss.

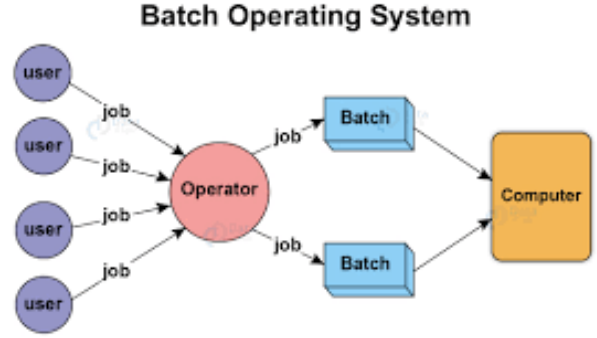
Different types of OS:

a) Based on Processing method:

The processing method refers to how the OS manages and executes tasks and processes on a computer system. It involves the techniques used to handle various tasks and allocate system resources, such as CPU time, memory, and input/output devices, to different processes or users.

1) Batch Operating System:

Batch operating systems process a group of similar jobs (programs or tasks) together in batches without requiring user intervention. Users submit their jobs to the system, and the operating system executes them in sequence. The system processes each job without any manual intervention, and the results are typically collected at the end of the batch processing.



Advantages:

- Batch systems can maximize CPU and I/O utilization since they execute jobs one after another without idle time between tasks. This leads to better resource utilization and higher system throughput.

-Batch processing is ideal for scenarios where the same job needs to be performed regularly. It is commonly used in tasks like payroll processing, report generation, and large-scale data processing.

-Since jobs are executed automatically in batches, there is minimal need for user interaction during the execution phase. This makes system management easier, especially in environments with a high volume of similar tasks.

Disadvantages:

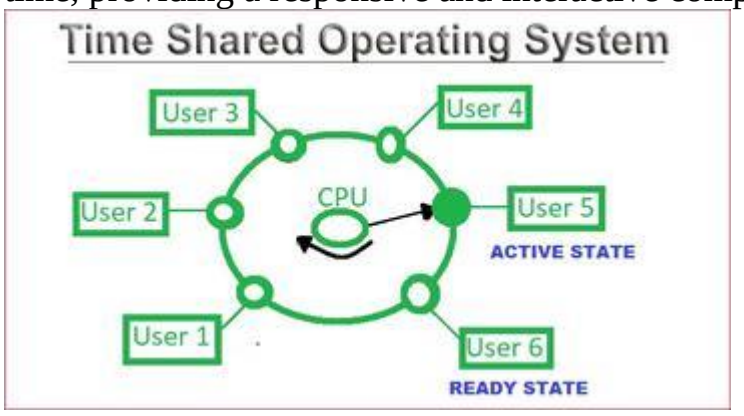
-Batch systems do not provide immediate feedback to users, as jobs are processed in batches. Users might have to wait until the current batch completes before getting results or being able to submit new jobs.

- Batch systems are not well-suited for interactive or real-time applications where immediate user input or response is required. They are best used for tasks that can be executed without user intervention.

- Debugging and error handling can be challenging in batch systems. If errors occur during batch processing, users might have to wait until the entire batch is completed to identify and resolve the issues.

2) **Time sharing Operating System:**

A time-sharing operating system, also known as a multi-user operating system, enables multiple users to share a single computer simultaneously. The CPU time is divided into small time slices, and each user gets a portion of CPU time during which their tasks or processes are executed. Time-sharing allows users to interact with the system in real-time, providing a responsive and interactive computing experience.



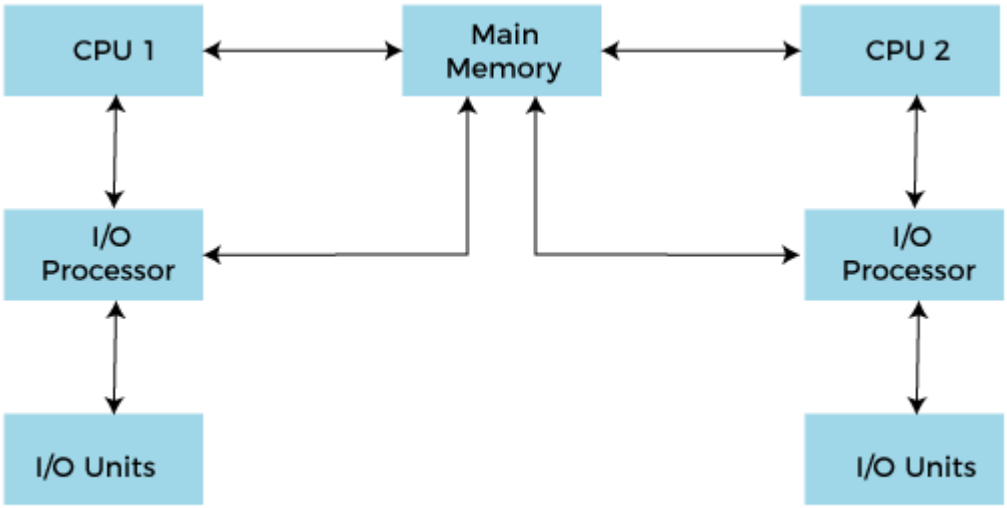
Advantages:

- Time-sharing OS provides real-time interaction with the computer, allowing users to execute commands and run applications interactively.

- Multiple users can work on the same computer simultaneously, efficiently utilizing the system's resources.
- The OS uses scheduling algorithms to ensure each user gets a fair share of CPU time, preventing resource monopolization.

Disadvantages:
- If too many users are active, the system may become slow due to frequent context switching, leading to decreased overall performance.
-Multiple users sharing the same system may risk unauthorized access to others' data or resources, necessitating robust security measures.
- In heavily loaded systems, users may compete for resources, potentially impacting the performance of individual tasks.

3) **Multiprocessing Operating System:**
A multiprocessing operating system utilizes multiple processors (CPUs) to execute tasks concurrently. Instead of relying on a single CPU, multiprocessing systems distribute tasks across multiple processors, enabling parallel processing and increasing overall system performance.



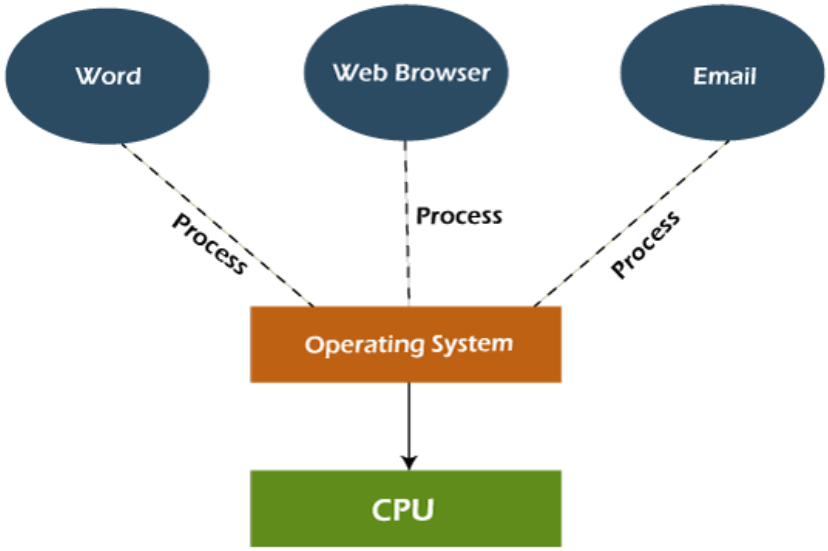
Working of Multiprocessor System

Advantages:
-By utilizing multiple processors, multiprocessing OS achieves faster execution and higher system performance, leading to reduced task completion times.
- Multiprocessing allows the system to process numerous jobs simultaneously, making it ideal for environments with heavy computational workloads.
- In case one processor fails, the system can continue functioning with the remaining processors, ensuring high availability and better fault tolerance.

Disadvantages:
- Designing and managing a multiprocessing system can be complex due to issues like data synchronization, load balancing, and inter-processor communication.
-Multiprocessing systems require additional hardware, making them more expensive to implement and maintain compared to single-processor systems.
- Adding more processors may not always result in a proportional performance boost due to overhead and communication bottlenecks, posing scalability limitations.

4) **Multitasking Operating System:**

A multitasking operating system allows a single CPU to handle multiple tasks simultaneously by rapidly switching between them. It gives the illusion of parallel execution, as the CPU allocates small time slices to each task in a round-robin fashion, ensuring that each task gets a fair share of CPU time.



Advantages:

-Multitasking enables optimal use of CPU time by allowing multiple tasks to share the processor, leading to improved overall system efficiency.

- Users can run multiple applications simultaneously, enhancing productivity and allowing seamless multitasking.

- The OS handles task switching and scheduling, simplifying the management of multiple running applications.

Disadvantages:

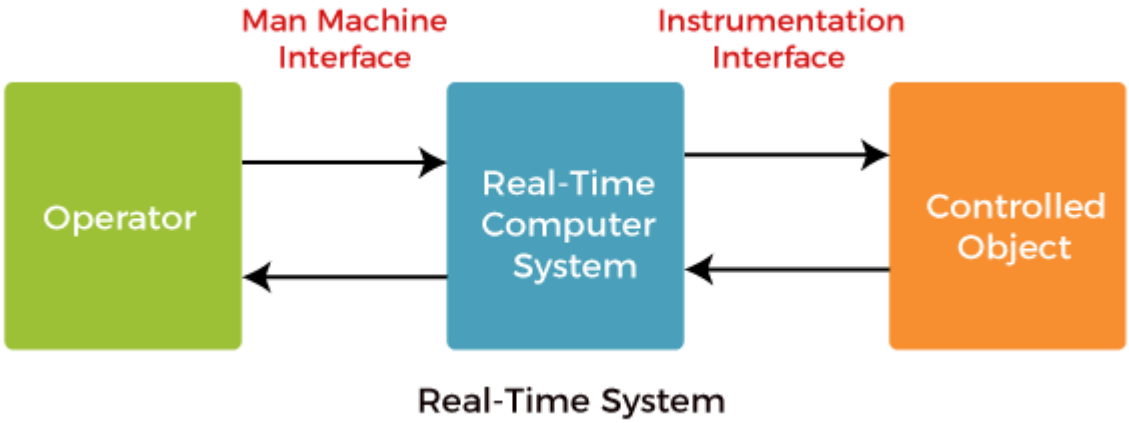
- Frequent task switching introduces some overhead, which can slow down individual tasks and affect overall system performance.

- If one application misbehaves or crashes, it can affect the stability of other running tasks, potentially leading to system instability.

- Running multiple tasks simultaneously may require more RAM, and if the system has limited memory, it could lead to performance issues**.**

5) **Real Time Operating System:**

A real-time operating system (RTOS) is designed to process tasks within strict time constraints. It is crucial for applications that require immediate and predictable responses to external events, often in real-time scenarios like robotics, aerospace, industrial control systems, and medical devices.



Real-Time System

Advantages:

-RTOS ensures critical tasks are completed on time, crucial for time-sensitive applications.

- It prioritizes tasks based on deadlines, maintaining system stability and predictability.

- RTOS allows precise control over processes and timing, essential for accurate operation in critical environments.

Disadvantages:

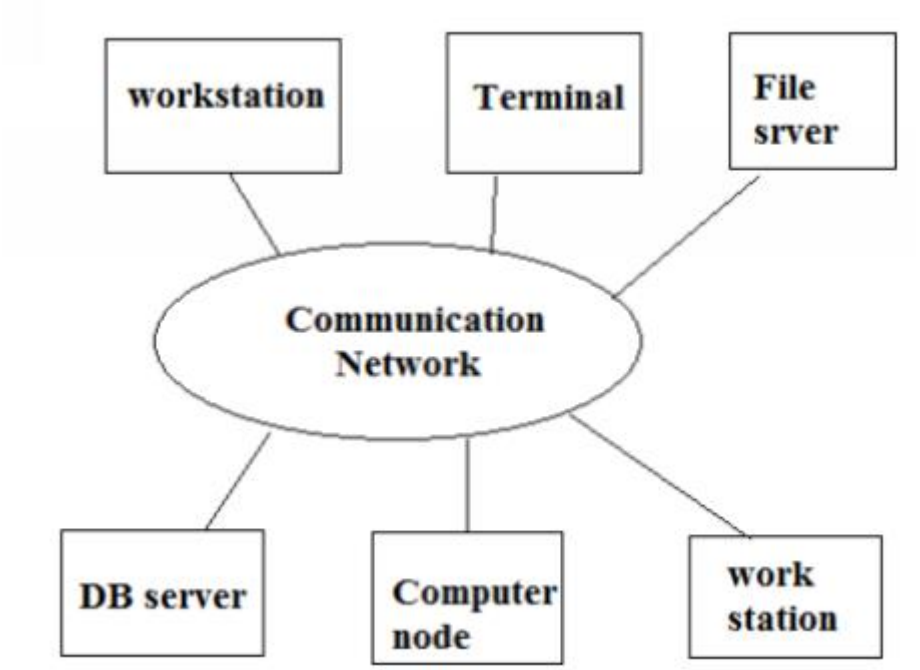
- Real-time systems may not be as flexible to adapt to changing application needs.

- Designing and implementing RTOS can be complex due to precise timing requirements.

- RTOS often requires specialized hardware and tools, making it more expensive to develop and maintain.

6) **Distributed Operating System:**

A distributed operating system manages a network of interconnected computers and enables them to work together as a single, integrated system. The distributed OS provides transparency to users, making it appear as if the entire network functions as a unified computing environment.



Advantages:

- Distributed systems allow efficient sharing of resources like processing power, storage, and data among multiple interconnected machines.
- Adding more machines to the network can enhance performance and capacity, making distributed systems easily scalable.
- Distributed systems can continue operating even if one or more machines fail, ensuring high availability and reliability.

Disadvantages:

- Managing a distributed system involves dealing with issues like communication, synchronization, and data consistency across multiple machines, which can be complex.
- Distributed systems are vulnerable to security breaches due to increased communication and interconnectedness.
- Network communication introduces latency and may affect the overall system performance compared to a centralized system.

(Tips to Remember: BTMMRD)

Assignment:

1) What is the main purpose of a batch operating system? Provide an example of a repetitive task that can be efficiently performed using batch processing.
2) Differentiate between Hard RTOS and Soft RTOS.
3) Describe two advantages of using a multiprocessing operating system. Why is it preferred for applications requiring high computational performance?
4) What is the primary benefit of a multitasking operating system? How does it allow users to run multiple applications simultaneously on a single CPU?
5) In a few sentences, explain what a distributed operating system does. Mention one advantage and one potential challenge associated with managing a distributed system.

b) Based on User Interface:

i. Command Line Interface (CLI):

- In a CLI, users interact with the operating system by typing commands into a text-based terminal.
- The system responds with text-based output, and users need to know specific commands and their syntax to perform tasks.
- Examples of CLI-based operating systems include UNIX, Linux, and MS-DOS.

ii. Graphical User Interface (GUI):

- GUI-based operating systems provide a visual interface with icons, windows, and menus to interact with the system.
- Users can perform tasks by clicking on icons, dragging and dropping files, and using intuitive graphical elements.
- Examples of GUI-based operating systems include Windows, macOS, and most modern Linux distributions.

iii. Web User Interface (WebUI):

- WebUI-based operating systems use a web browser as the primary interface for interacting with the system.
- Users access applications and services through web-based portals, and tasks are performed within the browser environment.
- Examples of WebUI-based operating systems include Chrome OS and some cloud-based operating systems.

iv. Touch-based User Interface:

- Touch-based operating systems are designed for devices with touchscreens, such as smartphones and tablets.
- Users interact with the system using gestures, swipes, taps, and pinches to navigate and perform tasks.
- Examples of touch-based operating systems include Android, iOS, and Windows 10 (for tablets).

v. Voice User Interface (VUI):

- VUI-based operating systems use voice commands and natural language processing to interact with the system.
- Users can issue spoken commands to perform tasks and access information.
- Examples of VUI-based operating systems include some smart speaker operating systems like Amazon Echo (Alexa) and Google Home (Google Assistant).

c) Based on Number or mode of users:

i. Single-User Operating System:

- A single-user operating system is designed to support only one user at a time.
- It allows the user to have full control over the system and all its resources.
- Examples of single-user operating systems include most desktop and laptop operating systems like Windows, macOS, and Linux distributions used on personal computers**.**

ii. Multi-User Operating System:

- A multi-user operating system allows multiple users to access and use the system simultaneously.
- Each user has their own account, and the OS provides mechanisms for resource sharing and user isolation.

- Examples of multi-user operating systems include UNIX and Linux servers, which can handle multiple remote connections from different users.

End of Unit-1