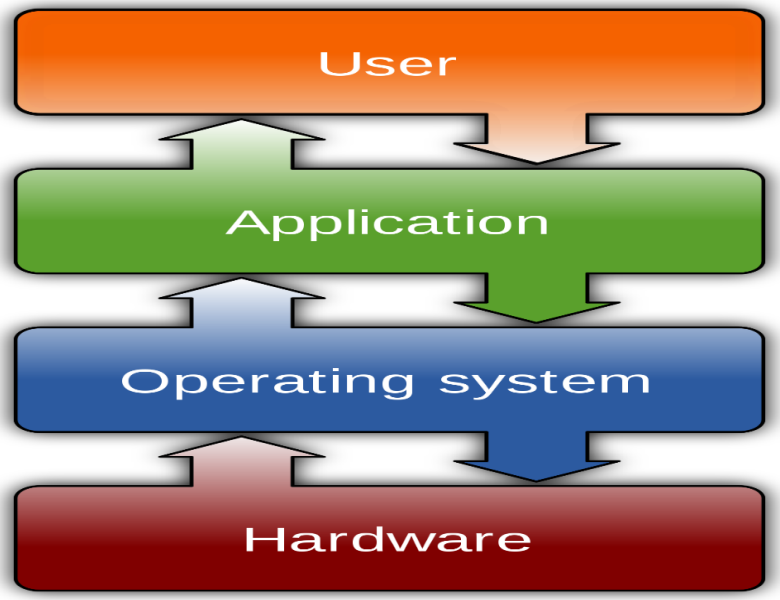
Definition of OS

Definition1:

The interface between user and hardware is called Operating System

Definition2:

An operating system (OS) is system software that manages computer hardware, software resources, and provides common services for computer programs.



Types of OS:

SINGLE USER OS -EX : MS-DOS, MS-WINDOWS ETC

MULTI USER 0S- EX:UNIX & LINUX ETC

Categorized of OS

Operating systems can be categorized into several types based on their functionality, usage, and design. Here are some common types of operating systems:

1. **Batch Operating System:**
   - Processes are grouped together and executed in batches.
   - Users submit their jobs and do not interact with the computer during job execution.
   - Examples: early mainframe systems like UNIVAC and IBM OS/360.
2. **Multiprogramming Operating System:**
   - Allows multiple programs to run concurrently.
   - The operating system switches between programs to maximize CPU usage.
   - Examples: UNIX, Linux, and Windows.
3. **Time-Sharing Operating System:**
   - Similar to multiprogramming but designed to provide interactive use.
   - Users share the computer's resources, and the OS switches between tasks quickly.
   - Examples: UNIX, Linux.
4. **Real-Time Operating System (RTOS):**

- Designed for systems that require immediate or near-immediate response.
- Common in embedded systems, industrial control systems, and robotics.
- Examples: VxWorks, FreeRTOS.

5. **Distributed Operating System:**
   - Manages a group of independent computers and makes them appear as a single computer.
   - Enables resource sharing and communication across a network.
   - Examples: Amoeba, Google's Chrome OS.
6. **Network Operating System (NOS):**
   - Manages network resources and provides services to other computers (nodes) on the network.
   - Examples: Novell NetWare, Microsoft Windows Server.
7. **Mobile Operating System:**
   - Designed for mobile devices like smartphones and tablets.
   - Examples: Android, iOS.
8. **Embedded Operating System:**
   - Designed to operate embedded systems, which are dedicated to specific tasks.
   - Examples: VxWorks, QNX.
9. **Single-User, Single-Task Operating System:**
   - Designed for a single user and can handle only one task at a time.
   - Examples: MS-DOS (Microsoft Disk Operating System).
10. **Multi-user Operating System:**

- Allows multiple users to use the computer simultaneously.
- Examples: UNIX, Linux, Windows Server editions.

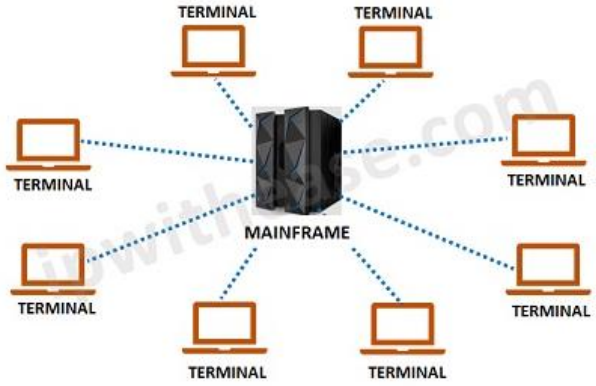
MAINFRAME SYSTEMS

- z/OS, a widely used mainframe operating system, is designed to offer a stable, secure, and continuously available environment for applications running on the mainframe.
- As a control program, **z/Virtual Machine** (z/VM) is a **hypervisor** because it runs other operating systems in the virtual machines it creates.
- **z/Virtual Storage Extended** (z/VSE) is popular with users of smaller mainframe computers. Some of these customers eventually migrate to z/OS when they grow beyond the capabilities of z/VSE.
- Several (non-IBM) Linux distributions can be used on a mainframe.
- The **z/Transaction Processing Facility** (z/TPF) operating system is a special-purpose system that is used by companies with very high transaction volume, such as credit card companies and airline reservation systems

Mainframe Systems

- Reduce setup time by batching similar jobs
- Automatic job sequencing – automatically transfers control from one job to another. First rudimentary operating system.
- Resident monitor
  - initial control in monitor
  - control transfers to job
  - when job completes control transfers pack to monitor

What is Mainframe?



Here are some key characteristics and features of mainframe operating systems:

1. **Batch Processing:** Mainframes are often used for batch processing, where large volumes of data are processed in scheduled batches. The operating system facilitates the orderly execution of batch jobs, managing the input, processing, and output in a sequential manner.
2. **Multiuser and Multitasking:** Mainframe operating systems support multiple users simultaneously, allowing several users to access and use the system resources concurrently. They also support multitasking, enabling the execution of multiple tasks or processes at the same time.
3. **Resource Management:** These operating systems efficiently allocate and manage system resources such as CPU time, memory, and input/output devices to ensure optimal performance and responsiveness.
4. **Security:** Security is a critical aspect of mainframe operating systems. They include robust security features to protect sensitive data, control access to resources, and ensure the integrity of the system.

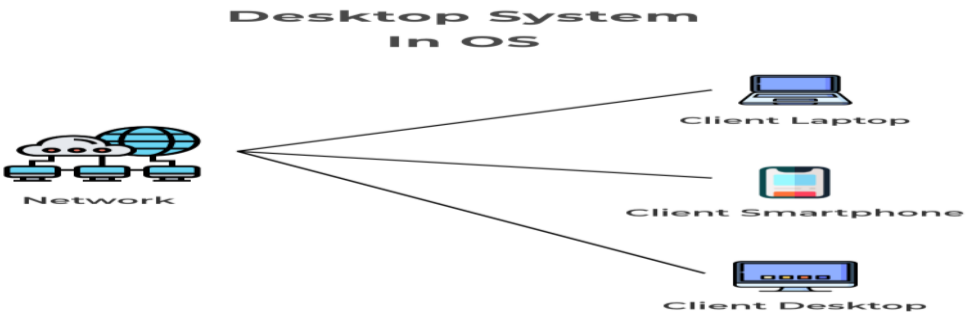
5. **Reliability and Availability:** Mainframes are designed for high reliability and availability. The operating systems include features such as fault tolerance, backup systems, and redundancy to minimize downtime and ensure continuous operation.
6. **Transaction Processing:** Mainframes often handle large volumes of online transaction processing (OLTP). The operating systems are optimized for efficient transaction processing, ensuring data integrity and consistency in real-time.
7. **Database Management:** Mainframe operating systems include features for managing large databases efficiently. They provide tools and utilities for data storage, retrieval, and manipulation.
8. **Compatibility:** Mainframe operating systems are designed to be compatible with a wide range of hardware and software configurations. They support various programming languages and application environments.
9. **Networking:** Mainframes are typically part of extensive networked environments. Mainframe operating systems include networking capabilities to facilitate communication and data exchange between mainframes and other computing systems.

Some examples of mainframe operating systems include IBM z/OS (previously known as OS/390 and MVS), IBM z/VSE, and Unisys MCP (Master Control Program).

DESKTOP SYSTEM

What is desktop system operating system?

Desktop operating system is the environment through which the user controls a personal computer (Notebook, desktop, PC). For example, Windows, Max OS, and various Linux distributions (Ubuntu, Linux Mint, CentOS, Debian, and so on). Operating system. Personal Computer (PC)



The most widely used ones include:

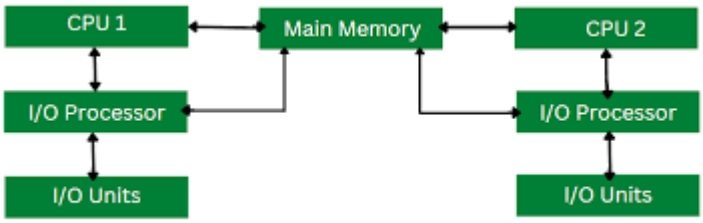
1. **Microsoft Windows:** Windows is one of the most popular desktop operating systems globally. Versions like Windows 10 and Windows 11 are commonly used on PCs and laptops.
2. **macOS:** Developed by Apple Inc., macOS is the operating system for Macintosh computers. It is known for its sleek design and is exclusive to Apple hardware.
3. **Linux:** Linux is an open-source operating system kernel that is widely used in various distributions (distros). Popular desktop Linux distributions include Ubuntu, Fedora, Debian, and Linux Mint.

Components of a Desktop System

- **Central Processing Unit (CPU):** The CPU is the brain of a desktop system, responsible for executing instructions and performing calculations. It processes data and carries out tasks based on the instructions provided by software programs. The CPU's performance is measured by its clock speed, number of cores, and cache size.
- **Random Access Memory (RAM):** RAM is a type of volatile memory that temporarily stores data and instructions for the CPU to access quickly. It allows for efficient multitasking and faster data retrieval, significantly impacting the overall performance of the system. The amount of RAM in a desktop system determines its capability to handle multiple programs simultaneously.
- **Storage Devices:** Desktop systems utilize various storage devices to store and retrieve data. Hard Disk Drives (HDDs) are the traditional storage medium, offering large capacities but slower read/write speeds. Solid-State Drives (SSDs) are a newer technology that provides faster data access, enhancing the system's responsiveness and reducing loading times.
- **Graphics Processing Unit (GPU):** The GPU is responsible for rendering images, videos, and animations on the computer screen. It offloads the graphical processing tasks from the CPU, ensuring smooth visuals and enabling resource-intensive applications such as gaming, video editing, and 3D modeling. High-performance GPUs are essential for users who require demanding graphical capabilities.
- **Input and Output Devices:** Desktop systems are equipped with various input and output devices. Keyboards and mice are the primary input devices, allowing users to interact with the system and input commands. Monitors, printers, speakers, and headphones serve as output devices, providing visual or auditory feedback based on the system's output.

MULTI PROCESSOR SYSTEM

Multiprocessor operating systems are used in operating systems to boost the performance of multiple CPUs within a single computer system. Multiple CPUs are linked together so that a job can be divided and executed more quickly. When a job is completed, the results from all CPUs are compiled to provide the final output.



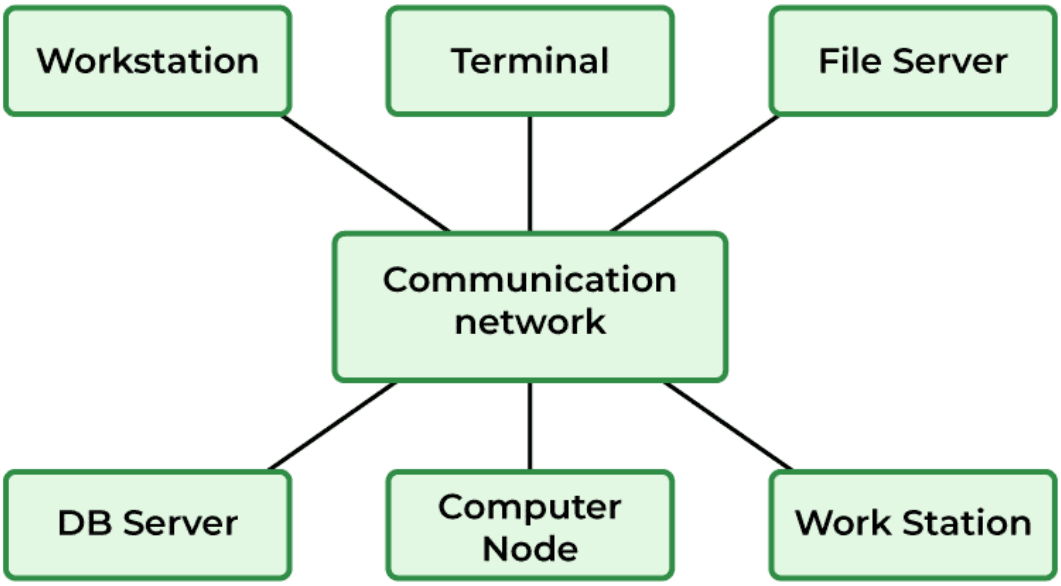
some key aspects of a multiprocessor system in terms of the operating system:

1. **Task Scheduling:**
   - In a multiprocessor system, the OS must schedule tasks to run on multiple processors concurrently.
   - Load balancing is essential to distribute tasks evenly across processors, ensuring efficient utilization of resources.
2. **Process Management:**
   - The OS is responsible for creating, managing, and terminating processes. This involves allocating resources to processes and handling inter-process communication (IPC) mechanisms.
   - Processes may run concurrently on different processors, and the OS must coordinate their execution.
3. **Memory Management:**
   - Memory management in a multiprocessor system involves allocating and deallocating memory across multiple processors.
   - Cache coherence is a critical issue in multiprocessor systems, and the OS must ensure that data is consistent across different caches.
4. **I/O Management:**
   - Input and output operations must be managed efficiently in a multiprocessor environment.
   - The OS coordinates I/O operations and ensures that multiple processors can access I/O devices without conflicts.
5. **Synchronization and Mutual Exclusion:**
   - Synchronization mechanisms are crucial to prevent race conditions and ensure the correct execution of concurrent processes.
   - Mutual exclusion is needed to avoid conflicts when multiple processors attempt to access shared resources simultaneously.
6. **Inter-Processor Communication:**
   - The OS facilitates communication between processors through mechanisms such as message passing or shared memory.
   - Coordination and synchronization between processors are critical to maintaining the integrity of the system.
7. **Fault Tolerance:**
   - Multiprocessor systems may incorporate fault tolerance mechanisms to ensure system reliability.
   - The OS may implement strategies such as redundancy and error recovery to handle hardware failures.
8. **Scalability:**
   - Multiprocessor systems should be designed to scale efficiently as the number of processors increases.
   - The OS must be capable of adapting to changes in the system configuration dynamically.
9. **Power Management:**
   - Efficient power management is crucial in modern multiprocessor systems to optimize energy consumption.
   - The OS may implement power-saving features and dynamically adjust processor frequencies based on system load.
10. **Parallel Programming Support:**

- The OS may provide tools and libraries to support parallel programming, making it easier for developers to create applications that can take advantage of multiple processors.

DISTRIBUTED SYSTEM

A Distributed Operating System refers to a model in which applications run on multiple interconnected computers, offering enhanced communication and integration capabilities compared to a NETWORK OS.
The diagram below illustrates the structure of a distributed operating system:



Distributed Operating System

A distributed operating system is an essential type of operating system. Distributed systems use many central processors to serve multiple real-time applications and users. As a result, data processing jobs are distributed between the processors.

It connects multiple computers via a single communication channel. Furthermore, each of these systems has its own processor and memory. Additionally, these **CPUs** communicate via high-speed buses or telephone lines. Individual systems that communicate via a single channel are regarded as a single entity. They're also known as **loosely coupled systems**.

Distributed Operating System:

- A distributed operating system extends the capabilities of a single-machine operating system to manage resources and provide services across a network of computers.
- It provides a consistent and integrated environment for users and applications, even though the resources are distributed.

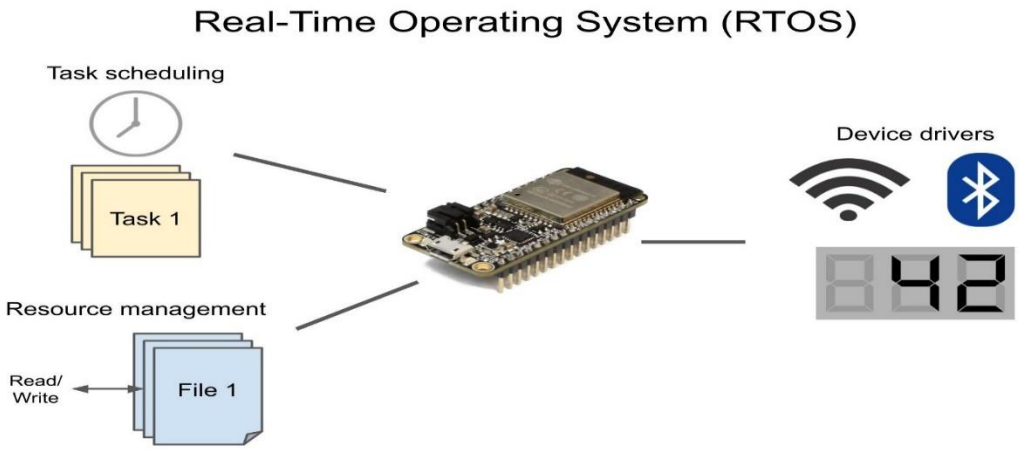
Examples of Distributed OSSolaris – Made for multiprocessor SUN workstations. OSF/1 – Created by the very Open Foundation Software Company and is Unix compatible. Micros – While allocating particular jobs to all nodes present in the system, the MICROS OS ensures a balanced data load.

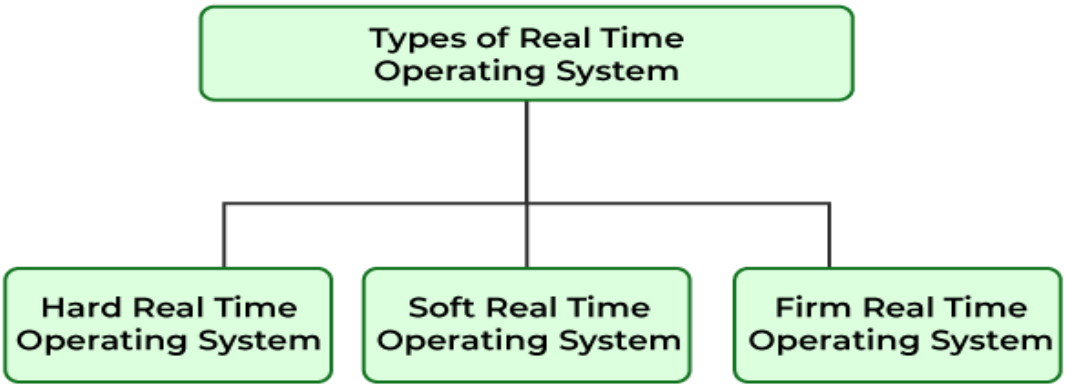
REAL TIME SYSTEM

RTOSes are designed to handle multiple processes at one time, ensuring that these processes respond to events within a predictable time limit. Processing in an RTOS occurs within defined time constraints and monitors the priority of tasks. An RTOS is also able to make changes to task priority.

A real-time operating system (RTOS) is an operating system with two key features: predictability and determinism. In an RTOS, repeated tasks are performed within a tight time boundary, while in a general-purpose operating system, this is not necessarily so.

Examples of real-time operating systems are airline traffic control systems, Command Control Systems, airline reservation systems, Heart pacemakers, Network Multimedia Systems, robots, etc.





1. **Hard Real-Time Operating System:** These operating systems guarantee that critical tasks are completed within a range of time.
   For example, a robot is hired to weld a car body. If the robot welds too early or too late, the car cannot be sold, so it is a hard real-time system that requires complete car welding by the robot hardly on time., scientific experiments, medical imaging systems, industrial control systems, weapon systems, robots, air traffic control systems, etc.

2. **Soft real-time operating system:** This operating system provides some relaxation in the time limit.
   For example – Multimedia systems, digital audio systems, etc. Explicit, programmer-defined, and controlled processes are encountered in real-time systems. A separate process is changed by handling a single external event. The process is activated upon the occurrence of the related event signaled by an interrupt.
   Multitasking operation is accomplished by scheduling processes for execution independently of each other. Each process is assigned a certain level of priority that corresponds to the relative importance of the event that it services. The processor is allocated to the highest-priority processes. This type of schedule, called, priority-based preemptive scheduling is used by real-time systems.

3. **Firm Real-time Operating System**: RTOS of this type have to follow deadlines as well. In spite of its small impact, missing a deadline can have unintended consequences, including a reduction in the quality of the product. Example: Multimedia applications.