**Operating System (OS)**

An **Operating System (OS)** is a collection of software that manages computer hardware resources and provides common services for computer programs.

An operating system acts as an interface between the software and different parts of the computer or the computer hardware.

It controls and monitors the execution of all other programs that reside in the computer, which also includes application programs and other system software of the computer.

Commonly Used OS: Windows, UNIX, LINUX, BOSS, SOLARIS

**Functions of OS**

**1. Memory Management:**

* An Operating system manages the allocation and deallocation of memory to various processes and ensures that the other process does not consume the memory allocated to one process.
* It allocates the memory to a process when the process requests and deallocates the memory when the process has terminated or is performing an I/O operation.

**2. Processor Management:**

* In a multi-programming environment, the OS decides the order in which the processes should access the processor, and how much time each process has, this is also called **Process Scheduling**.

**3. Device Management:**

* An OS manages device communication via its respective drivers.
* It keeps track of all the connected devices, receives requests from these devices, performs a specific task, and communicates back.

**4. User Interface:**

* The user interacts with the computer system through the OS. Hence it also acts as an interface between the user and the computer hardware.
* This interface which is used by users to interact with the applications and machine hardware is offered through a set of commands or a GUI (Graphical User Interface).

**5. Security:**

* The Operating System uses password protection and other similar techniques to protect user data.
* It also prevents unauthorized access to programs and user data.

**6. Job Accounting:**

* The operating system Keeps track of time and resources used by various tasks and using this information the OS decides the order of applications running and how much time should be allocated.

**A diagram of a operating system

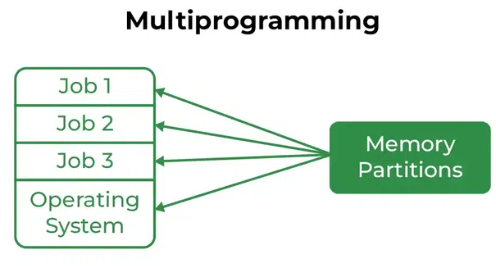
Description automatically generatedTypes of OS**

**1.Batch Operating System:**

This type of operating system does not interact with the computer directly.

There is an operator which takes similar jobs having the same requirement and groups them into batches.

It is the responsibility of the operator to sort jobs with similar needs.

**2. Multi-Programming OS**

Multiprogramming Operating Systems can be simply illustrated as more than one program is present in the main memory and any one of them can be kept in execution.

This is basically used for better execution of resources.

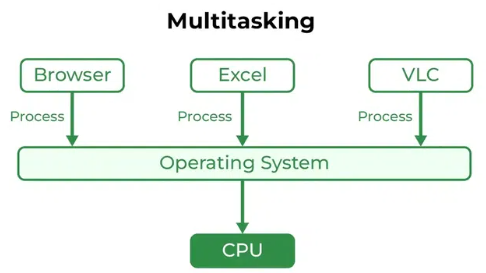
**A diagram of a computer

Description automatically generated**

**3. Multi-Processing OS**

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

It betters the throughput of the System.

**4. Multi-Tasking OS**

Multitasking Operating System is simply a multiprogramming Operating System with having facility of a Round-Robin Scheduling Algorithm.

It can run multiple programs simultaneously.

A diagram of a computer system

Description automatically generated**5. Time Sharing OS:**

It is the OS in which each task is given some time to execute so that all the tasks work smoothly.

The time that each task gets to be executed is called quantum. After this time interval is over OS switches over to the next task.

**A diagram of a computer network

Description automatically generated**

**6. Distributed OS:**

In this type of OS various autonomous interconnected computers communicate with each other using a shared network.

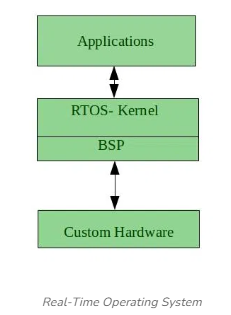
These are referred to as **loosely coupled systems**.

**A diagram of a computer network

Description automatically generated7. Networking OS:**

These systems run on a server and provide the capability to manage data, users, groups, security, applications, and other networking functions.

These are referred to as **tightly coupled systems**.

**8. Real-Time Operating System (RTOS):**

These types of OSs serve real-time systems. The time interval required to process and respond to inputs is very small.

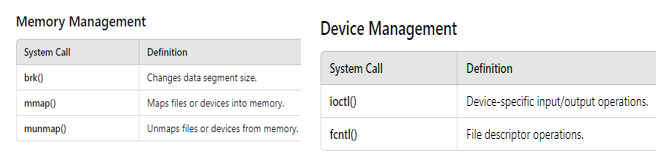
Real-time systems are used when there are time requirements that are very strict like missile systems, air traffic control systems, robots, etc.

Types of RTOS: Hard RTOS, Soft RTOS

**System Call**

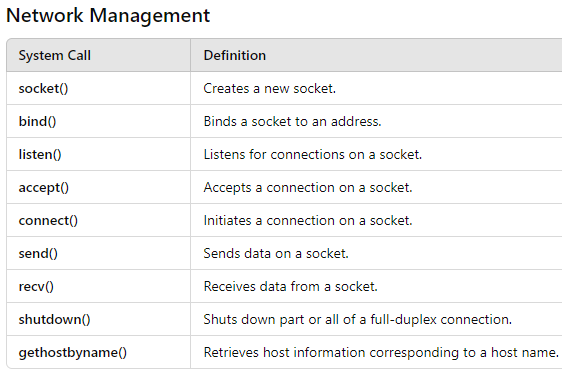
* It is a programmatic way in which a program requests a service from the kernel (central component) of OS.
* System call provides the services of the operating system to the user programs via Application Program Interface (API).
* A system call is initiated by the program executing a specific instruction, which triggers a switch to kernel mode, allowing the program to request a service from the OS. The OS then handles the request, performs the necessary operations, and returns the result back to the program.

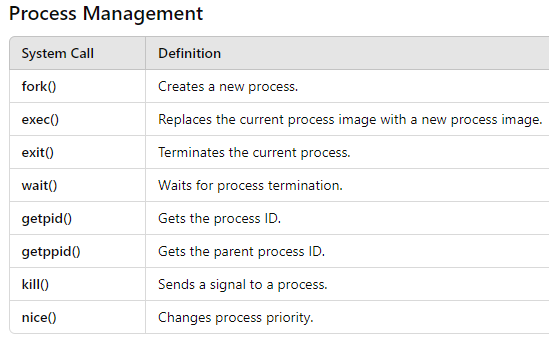
**Examples of System Call:**

****

**A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated**

****

**Process**

* A **process** is a program in execution.
* **For example**, when we write a program in C or C++ and compile it, the compiler creates binary code. The original code and binary code are both programs. When we run the binary code, it becomes a **process**.
* The OS is responsible for managing the start, stop, and scheduling of processes, which are basically programs running on the system and this management of processes is called **process management**.
* Every process goes through different states throughout the life cycle during the process execution, which is known as **process states**.
* Processes generally has following **7 states**:
  1. **New State**: This is the first state of the process life cycle. When creation of process is taking place; the process is in a **new state**.
  2. **Ready State**: When the process creation is completed, the process comes into a **ready state**. During this state, the process is loaded into the main memory and will be placed in the queue of processes which are waiting for the CPU allocation. When the process is in the creation process is in a new state and when the process gets created process is in the ready state.
  3. **Running State**: Whenever the CPU is allocated to the process from the ready queue, the process state changes to **Running**.
  4. **Block or Wait State**: When the process is executing the instructions, the process might require carrying out a few tasks that might not require CPU. If the process requires performing Input-Output task or the process needs some resources that are already acquired by other processes, during such conditions process is brought back into the main memory, and the state is changed to **Blocking or Waiting for the state**. Process is placed in the queue of processes that are in waiting or block state in the main memory.
  5. **Terminated or Completed**: When the entire set of instructions is executed, and the process is completed. The process is changed to a **terminated or completed state**. During this state the **PCB** of the process is also deleted.
  6. **Suspend Ready**: Whenever the main memory is full, the process which is in a ready state is swapped out from main memory to secondary memory. The process is in a ready state when goes through the transition of moving from main memory to secondary memory, the state of that process is changed **to Suspend Ready State**. Once the main memory has enough space for the process, the process will be brought back to the main memory and will be in a ready state.
  7. **Suspend Wait or Suspend Blocked**: Whenever the process that is in waiting for state or block state in main memory gets to swap out to secondary memory due to main memory being completely full, the process state is changed to **Suspend wait or Suspend blocked state.**

**Inter Processor Communication (IPC):**

IPC in OS is a way by which multiple processes can communicate with each other. Shared memory in OS, message queues, FIFO and so on are some of the ways to achieve **IPC** in OS.

A close-up of a sign

Description automatically generatedIPC provides a mechanism to exchange data and information across multiple processes, which might be on single or multiple computers connected by a network.

**Process Control Block (PCB):**

A Process Control Block (PCB) is a data structure used by the operating system to store all the information about a process. It is essential for process management and allows the OS to keep track of the state of processes.

When the process makes a transition from one state to another, the operating system must update information in the process’s PCB.

https://www.linkedin.com/pulse/50-commonly-asked-operating-system-interview-topic-wise-lokeswari/