An operating system is an interface between the user and hardware, responsible for process execution, resource allocation, CPU management, file management, and providing a convenient and efficient environment for executing programs.

Functions of Operating System: Memory Management, Processor Management, User Interfaces, File Management, Device Management, Security.

Some names of operating systems: MS Windows, Chrome OS, macOS, Android, and Ubuntu.

Multitasking: A system that works on more than one task at a time by rapidly switching between various tasks is known as multitasking.

Multiprocessing: A system that allows multiple processors in a computer to process two or more different portions of the same program simultaneously.

Multiprogramming: in OS is the ability to execute multiple programs concurrently by managing CPU and I/O resources efficiently.

Types of Operating Systems

1. Batch Operating System: Jobs with similar requirements are grouped into batches and processed sequentially without user interaction.

2. Multiprogramming OS: Multiprogramming OS runs multiple programs simultaneously by managing system resources efficiently. It switches between programs so the CPU is always busy, improving performance and reducing idle time.

3. Multiprocessing Operating System: A multiprocessing operating system uses multiple CPUs to perform tasks simultaneously. Each CPU works on different tasks at the same time, which enhances performance and reliability.

4. Multi-Tasking Operating: A Multi-Tasking Operating System is a logical extension of a multiprogramming system that allows multiple tasks or processes to run concurrently on a single CPU. It achieves this by rapidly switching between tasks, giving the illusion that all are running simultaneously.

5. Multithreading OS: Multithreading is a feature that allows a single process to have multiple threads of execution running concurrently. Each thread can perform a different task, but they share the same resources such as memory, which makes communication between threads easier.

6. Real-Time Operating System (RTOS): A Real-Time Operating System ensures tasks are completed within a fixed time frame, crucial for time-sensitive applications like medical devices and automotive systems.

The kernel is the central component of an operating system responsible for handling, managing, and controlling all operations.

Types of Kernel: Monolithic Kernel, Microkernel, Hybrid Kernel, Nano Kernel, Exokernel.

**Monolithic Kernel**: A monolithic kernel is a kernel which includes all operating system code in a single executable image.

**Microkernel**: Microkernel is the kernel which runs minimal performance-affecting services for the operating system. In the microkernel operating system, all other operations are performed by the processor.

**Hybrid Kernel**: A hybrid kernel is a combination of microkernel and monolithic kernel.

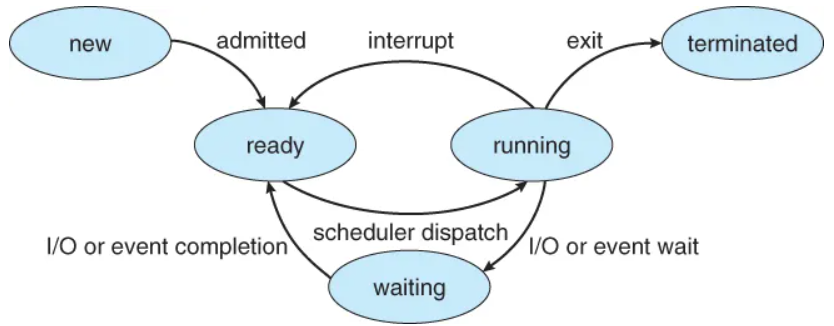
**Nano/Exo Kernels**: Optimized versions of microkernels that focus on minimalism and efficiency, often eliminating unnecessary services and allowing for a more streamlined operation.

A program is a set of instructions written in a programming language that a computer can execute to perform a specific task.

Process: A program under execution, indicated by the program counter (PC), represented by a Process Control Block (PCB).

**Process vs. Program**

| **Aspect** | **Process** | **Program** |
| --- | --- | --- |
| **Definition** | An active entity (running instance) | A passive entity (set of instructions) |
| **State** | Changes dynamically | Static |
| **Example** | A running browser | The browser executable file |
| **Real-Life Example** | A recipe being actively followed to cook a dish | A written recipe |

****

**Context Switching:**  The process of saving the state of a running process and loading the state of another process to ensure smooth multitasking.

**Process Scheduling**:

1. **Arrival Time**: When the process enters the ready queue.
2. **Completion Time**: When the process finishes execution.
3. **Burst Time**: Time required for CPU execution.
4. **Turn Around Time**: Time from arrival to completion (Completion Time - Arrival Time).
5. **Waiting Time (WT)**: Time difference between turnaround time and burst time (Waiting Time = Turnaround Time - Burst Time).

Thread: A lightweight process and the basic unit of CPU utilization, allowing a process to perform multiple tasks simultaneously.

**Characteristics**:

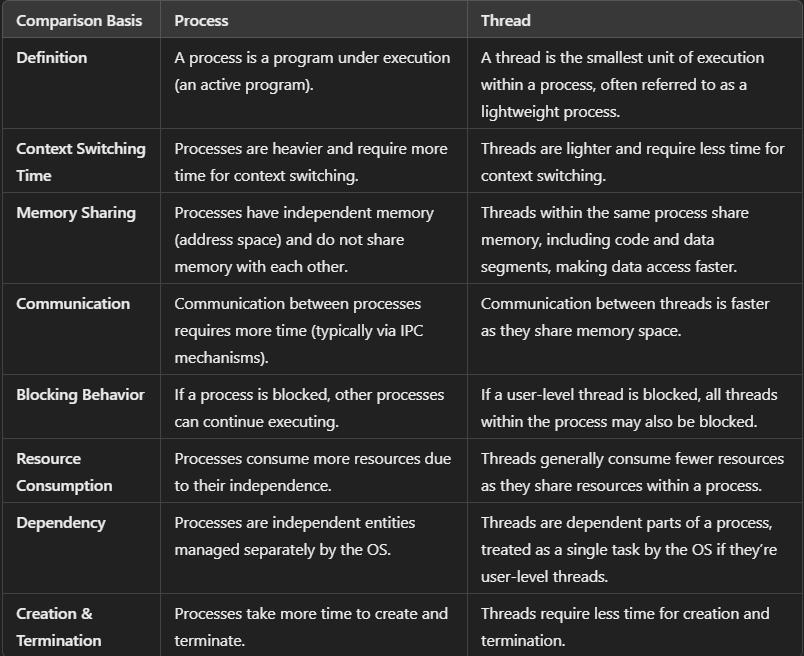
* + Each thread has its own program counter, register set, and stack.
  + Threads share resources (code, data, files, signals) with other threads in the same process.

**Note**: A new thread (child process) can be created using the fork() system call, with n calls generating 2n−12^n - 12n−1 child processes.

**Types of Threads**:

* **User Threads**: Implemented by users.
* **Kernel Threads**: Implemented by the operating system.

# Process vs Thread



Scheduling algorithms:

1. **First Come First Serve (FCFS)**: Schedules processes based on arrival times.
2. **Shortest Job First (SJF)**: Schedules processes with the shortest burst time first.
3. **Shortest Remaining Time First (SRTF)**: Preemptive SJF that schedules based on the shortest remaining time.
4. **Round Robin (RR)**: Each process receives a fixed time slice in a cyclic manner.
5. **Priority Based Scheduling (Non-Preemptive)**: Schedules processes by priority, with ties resolved by arrival time.
6. **Highest Response Ratio Next (HRRN)**: Schedules based on the highest response ratio, preventing starvation (Response Ratio = (Waiting Time + Burst Time) / Burst Time).
7. **Multilevel Queue Scheduling (MLQ)**: Processes are placed in different priority queues; higher priority queues are scheduled first.
8. **Multilevel Feedback Queue (MLFQ)**: Allows processes to move between queues based on CPU usage characteristics

**Critical Section**: The portion of the code in the program where shared variables are accessed and/or updated.

**Synchronization Tools:**

1. **Semaphore**: A protected variable used to lock resources; indicates resource status. Binary Semaphore, Counting Semaphore. **wait(), signal().**
2. **Mutex**: Provides mutual exclusion; only one thread can access a resource at a time, ensuring synchronization between producers and consumers.

### Deadlocks:

A situation where processes are blocked because each holds a resource and waits for another.

**Necessary Conditions for Deadlock:** Mutual Exclusion, Hold and Wait, No Preemption, Circular Wait.

**Methods for Handling Deadlock:** Deadlock Prevention/Avoidance, Deadlock Detection and Recovery, Ignore the Problem.

Banker's algorithm is used to avoid deadlock. It is one of the deadlock-avoidance methods. It is named as Banker's algorithm on the banking system where a bank never allocates available cash in such a manner that it can no longer satisfy the requirements of all of its customers.

Memory Management Memory management is a critical function of the OS, ensuring that memory is allocated efficiently and securely among processes.

***Types of Memory Management****:*

* **Paging**: Divides memory into fixed-size pages and divides the process into the same size. Pages can be loaded into any free frame in the physical memory.
* **Segmentation**: Divides processes into variable-size segments based on the logical divisions of the program (e.g., code, data, stack).
* **Virtual Memory**: Allows processes to use more memory than physically available by swapping pages between disk and RAM.

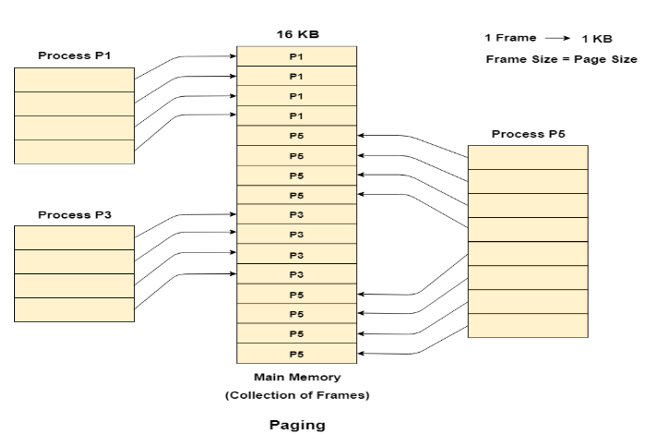
Allocation Schemes: **First Fit, Best Fit**, **Worst Fit**.

Best fit does not necessarily give the best results for memory allocation.

The cause of external fragmentation is the condition in Fixed partitioning and Variable partitioning saying that the entire process should be allocated in a contiguous memory location. Therefore Paging is used.

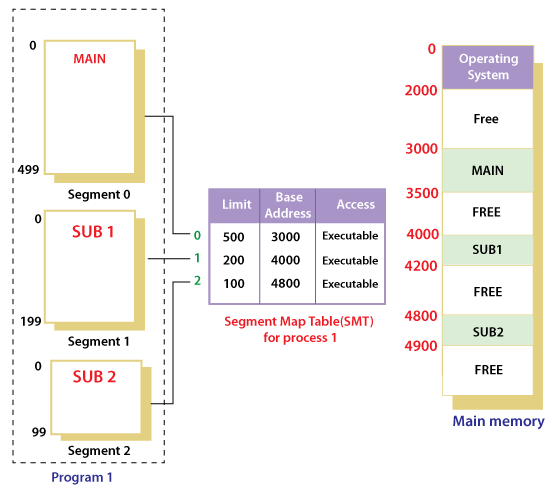
Paging is a memory management scheme in operating systems where processes are divided into pages, and memory is divided into frames. Pages from a process are loaded into frames for execution, and the page size matches the frame size.

The Memory Management Unit (MMU) translates a logical address (generated by the CPU) into a physical address (actual memory location). A logical address consists of two parts: **Page Number**, O**ffset**.



Segmentation is a memory management technique where memory is divided into variable-sized parts called segments. Each segment is allocated to a process and typically contains similar types of functions, such as storing the main function in one segment and library functions in another. A segment table holds the **base address** (starting point) and **limit** (size) of each segment.

The CPU maps the segment number to the segment table. The base address is added to the offset to generate the physical address.



A page fault is an interrupt raised when a program accesses a memory page not loaded in physical memory.

# ****Page Replacement Algorithms:****

1. **First In First Out (FIFO)**: This simplest algorithm uses a queue to track pages, replacing the oldest page first.
2. **Optimal Page Replacement**: This algorithm replaces pages that will not be used for the longest time in the future. It’s impractical as future requests are unknown.
3. **Least Recently Used (LRU)**: This algorithm replaces the least recently used page.

Belady’s Anomaly: It shows that increasing the number of page frames can lead to more page faults. (especially in the FIFO)

# ****Disk Scheduling Algorithms:****

1. **FCFS**: Services requests in the order they arrive.
2. **SSTF**: Executes requests with the shortest seek time first.
3. **SCAN**: Moves in one direction, servicing requests until it reaches the end, then reverses.
4. **CSCAN**: Similar to SCAN but scans all the way back to the start after reaching the end.
5. **LOOK**: Services requests only to the last one in front of the head, then reverses direction.
6. **CLOOK**: Similar to CSCAN, but only goes to the last request before returning to the opposite

Real-time System: A real-time system is used in the case when rigid-time requirements have been placed on the operation of a processor. It contains well-defined and fixed time constraints.

Re-entrancy: **Re-entrancy is a simple function in which various clients can use and shares a single copy of program during a similar period**

Demand Paging: A memory management scheme that loads pages into physical memory only when they are needed, swapping them to disk when not in use to optimize memory usage.

Virtual Memory (Imp): Virtual memory is a very useful memory management technique which enables processes to execute outside of memory. This technique is especially used when an executing program cannot fit in the physical memory.

RAID: RAID stands for Redundant Array of Independent Disks. It is used to store the same data redundantly to improve overall performance. There are 7 RAID levels.

Logical vs. Physical Address Space: Logical address space specifies the address that is generated by the CPU. On the other hand, physical address space specifies the address that is seen by the memory unit.

Fragmentation: Fragmentation is a phenomenon of memory wastage. It reduces capacity and performance because space is used inefficiently.

**Internal Fragmentation**: Internal fragmentation occurs when allocated memory is not fully utilized, leading to wasted space within memory blocks.

**External Fragmentation**: External fragmentation occurs when free memory is split into small, non-contiguous blocks, making it difficult to allocate large memory chunks.

Spooling: A process that temporarily gathers data to be used and executed by a device, program, or system, often associated with printing. It queues output from multiple applications into a disk file, managing print jobs efficiently.

Starvation: Starvation is a resource management problem. In this problem, a waiting process does not get the resources it needs for a long time because the resources are being allocated to other processes.

Aging: Aging is a technique used to avoid starvation in the resource scheduling system.

# Advantages of Multithreaded Programming:

Enhance the responsiveness to the users.

Resource sharing within the process.

Economical.

Completely utilize the multiprocessing architecture.

Thrashing: Thrashing is a phenomenon in virtual memory schemes when the processor spends most of its time in swapping pages, rather than executing instructions.

A socket is an endpoint for sending or receiving data across a computer network, allowing communication between applications.

A pipe is a mechanism used for inter-process communication that allows two or more related processes to connect and exchange data through message passing.

IPC stands for Inter Process Communication is a mechanism that allows processes to communicate with each other. **Shared memory, message passing.**

Zombie process: A process that has finished the-execution but still has an entry in process table.

Orphan process: A process whose parent process no more exist.

Device drivers are software components that allow the operating system to communicate with and control hardware devices.

A bootloader is a small program that loads the operating system into memory and starts its execution.

Context Switching is a process of saving the context of one process and loading the context of other process.

Direct Mapping**:** Each memory block maps to one specific cache line, leading to potential conflicts.

Associative Mapping: Any memory block can go into any cache line, reducing conflicts but needing more complex hardware.

The Producer-Consumer Problem involves a producer generating data and placing it in a shared buffer while a consumer retrieves data from that buffer, requiring synchronization to prevent overfilling and underflowing the buffer.

Data Binding**:** Data binding is a programming technique that establishes a connection between user interface elements and data sources, allowing automatic synchronization of data. Changes in the data source reflect in the UI and vice versa