Introduction to Operating Systems

PRESENTED BY

anujanrcm

NARSIMHA REDDY ENGINEERING COLLEGE

July 15, 2026

Introduction to Operating Systems

An Operating System (OS) is essential software that serves as a bridge between users and computer hardware . It manages resources , executes programs , and ensures efficient operation . Without an OS, users face challenges in hardware communication , program execution , and resource sharing . The OS enhances usability , efficiency , and security , making it a critical component of modern computing . Examples include Windows , Linux , and macOS , each tailored for specific user needs and environments .

What is an Operating System ?

Definition of OS

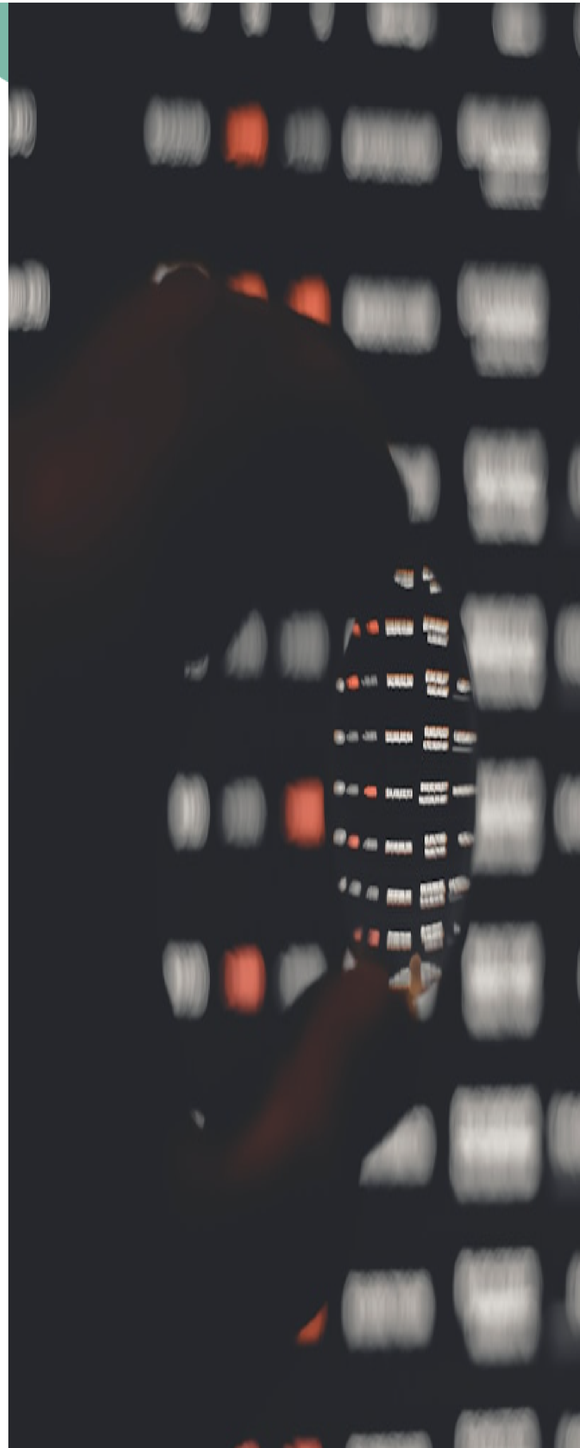
An Operating System (OS) is system software that serves as an interface between users and computer hardware , managing resources and providing essential services .

Importance of OS

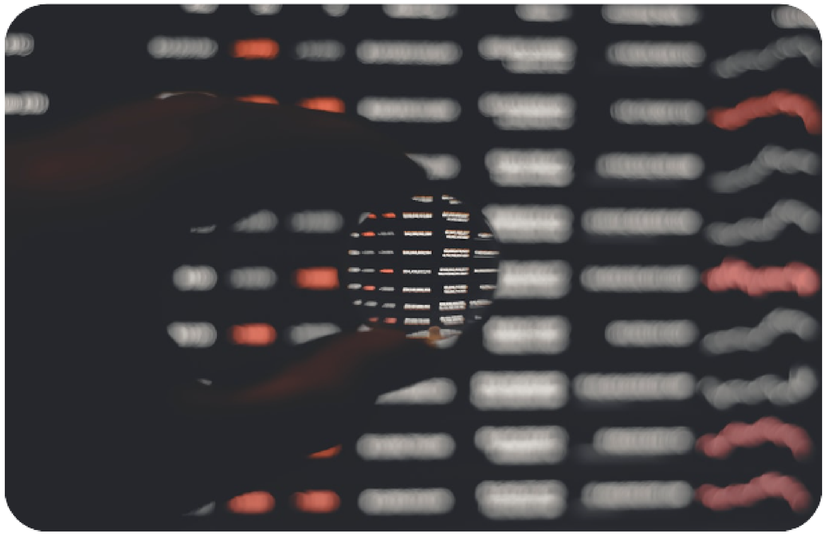
Without an OS, users cannot effectively communicate with hardware , leading to inefficient program execution and resource management , ultimately hindering usability .

Enhancing Usability

The OS enhances computer usability , efficiency , and security , making it easier for users to interact with applications and hardware seamlessly .



Objectives of an Operating System



Execute User Programs

The OS efficiently executes user programs , ensuring optimal CPU utilization and minimizing waiting time , which enhances overall system performance .

Manage Memory Effectively

An OS allocates and manages memory resources , ensuring that applications have the necessary memory while preventing conflicts and optimizing usage .

Control I/O Devices

Operating systems manage input/out devices , facilitating smooth communication between hardware a software , which is crucial for user interaction .

Overview of Different Types

Batch Operating System

Executes jobs sequentially without user interaction , improving throughput . Common in payroll processing , it reduces setup time but has long waiting times .

Multiprogrammed OS

Allows multiple programs in memory simultaneously , enhancing CPU utilization . When one process waits for I/O, the CPU switches to another , minimizing idle time .

Time -Sharing OS

Allocates CPU time among multiple users , enabling interactive computing . It provides fast responses but incurs more context switching and potential security concerns .

Batch Operating System vs. Multiprogrammed Operating System

Batch Operating System

- Executes jobs in batches without user interaction .
- High throughput due to sequential job processing .
- Long waiting times for users as jobs are processed one after another .
- Simple implementation with reduced setup time .
- Difficult debugging due to lack of real-time feedback .

Multiprogrammed Operating System

- Allows multiple programs to reside in memory simultaneously .
- CPU never remains idle , improving overall efficiency .
- Complex memory management and scheduling overhead .
- Better CPU utilization and higher throughput compared to batch systems .
- Facilitates interactive computing , allowing user engagement .

Core Components of an OS

Kernel Management

The kernel is the core of the OS, managing processes, memory, and device control, ensuring efficient resource allocation and system stability.

Process Manager Role

The process manager creates, schedules, and terminates processes, optimizing CPU usage and ensuring smooth execution of applications.

Memory Manager Function

The memory manager allocates and releases RAM effectively, maintaining system performance and preventing memory leaks or fragmentation.

Key Services Provided by OS

Program Execution

Operating systems load programs into memory and execute them, ensuring efficient CPU utilization. For instance, Windows can run multiple applications simultaneously.

I/O Operations

OS facilitates communication between software and hardware devices, enabling tasks like reading keyboard input or printing documents seamlessly.

File System Manipulation

Operating systems support essential operations such as creating, deleting renaming files, ensuring organized da management for users.

Understanding Processes in OS

1

Definition of a Process

A process is a program in execution, encompassing its own resources, including memory, CPU time, and I/O devices, essential for task completion.

2

Components of a Process

Key components include program code, data section, heap, stack, program counter, and CPU registers, all crucial for process management and execution.

3

Process Scheduling

Process scheduling involves selecting the next process for CPU usage, aiming for maximum CPU utilization and minimal waiting time for efficient performance.

Understanding Threads

Definition of Threads

A thread is the smallest unit of CPU execution within a process, allowing multiple threads to run concurrently, enhancing performance and responsiveness.

Resource Sharing

Threads within the same process share resources like memory and file descriptors, which leads to efficient resource utilization and faster communication.

User-Level vs Kernel-Level

User-Level Threads (ULT) are faster to create, while Kernel-Level Threads (KLT) offer better support for parallel execution, albeit with slightly higher overhead.

Advantages of Threads

Threads improve responsiveness, reduce resource consumption, and enhance scalability, allowing applications to utilize multiple CPU cores effectively.

Key Takeaways

In summary , operating systems are essential for managing hardware resources , executing programs , and providing user-friendly interfaces . Understanding different OS types , such as batch and real-time systems , enhances our ability to choose the right OS for specific applications . As technology evolves , staying informed about OS advancements will empower users and developers alike to optimize performance and security in computing environments .