Operating System

An operating system manages computer hardware and software resources.

Processes represent running programs in memory.

Threads are lightweight execution units inside processes.

Scheduling determines process execution priority.

Round-robin scheduling allocates fixed time slots to processes.

Multitasking allows multiple processes to run simultaneously.

Interrupts signal processors to pause and handle events.

System calls allow applications to request OS services.

Memory management allocates RAM to active processes.

Paging divides memory into fixed-size blocks.

Segmentation divides memory into variable-sized sections.

Virtual memory extends RAM using disk space.

Deadlocks occur when processes wait indefinitely for resources.

Semaphores help synchronize processes.

Mutex locks provide exclusive access to shared data.

File systems organize data storage on drives.

Directory structures store file metadata.

I/O management handles input and output devices.

Device drivers enable hardware communication.

Boot loaders initialize the operating system.

Kernel space manages core system operations.

User space isolates applications for safety.

Security policies protect system access.

Authentication verifies user identity.

Authorization manages access permissions.

Access control lists store security privileges.

System logs record operational events.

Resource allocation assigns CPU, memory, and devices.

Context switching swaps processes during multitasking.

IPC enables inter-process communication.

Pipes transfer data between processes.

Sockets enable network communication.

Shells allow users to interact with the OS.

Command interpreters execute user commands.

Backup and recovery protect data integrity.

Power management optimizes energy consumption.

Real-time operating systems support time-critical tasks.

Embedded OS runs on compact hardware.

Mobile operating systems manage smartphones.

Windows, Linux, and macOS are common OS types.

Open-source operating systems allow public code modification.

Kernel modes determine privilege levels.

Monolithic kernels include all services in one unit.

Microkernels minimize core OS components.

Hybrid kernels combine both approaches.

Cloud computing relies on scalable virtualized OS instances.

Virtual machines emulate full operating environments.

Containers isolate applications with shared kernels.

Operating systems remain essential for modern computing.