

Introduction to Operating Systems

Introduction to Operating Systems

Course-level overview – concise, organized, and ready for study or quick reference.

1. What Is an Operating System (OS)?

Aspect	Description
Definition	System software that manages hardware resources and provides services for application programs.
Primary Goals	• Efficient resource utilization • Fair and secure sharing of resources • Simplicity for programmers (abstraction)
Typical Users	End-users, developers, system administrators, embedded devices.
Examples	Windows, macOS, Linux, Android, iOS, RTOS (e.g., FreeRTOS, VxWorks).

2. Core Functions of an OS

1. Process Management

- Creation, scheduling, termination of processes/threads.
- Context switching and inter-process communication (IPC).

2. Memory Management

- Allocation/deallocation of RAM.
- Virtual memory, paging, segmentation, swapping.

3. File System Management

- Organization, storage, retrieval, and protection of data on disks, SSDs, etc.
- Directory structures, naming conventions, metadata.

4. Device Management

Introduction to Operating Systems

- Drivers translate generic OS calls into hardware-specific actions.
- Buffering, caching, and I/O scheduling.

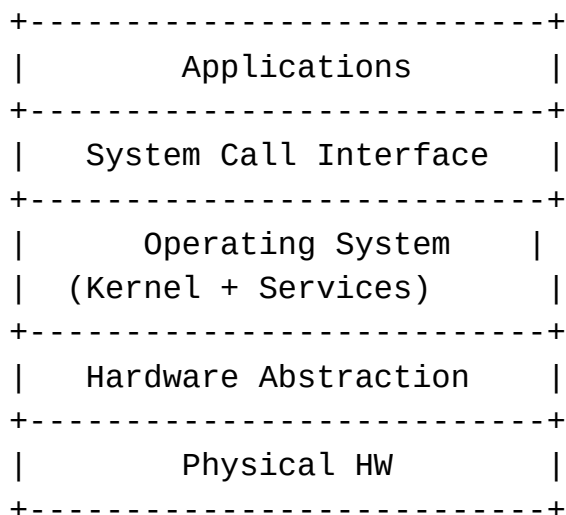
5. Security & Protection

- Authentication, authorization, access control lists (ACLs).
- Isolation of processes (user vs. kernel mode), encryption, auditing.

6. User Interface (UI)

- CLI (command-line interface) – e.g., Bash, PowerShell.
 - GUI (graphical UI) – windows, icons, menus, pointers.
-

3. System Architecture



3.1 Kernel vs. User Space

Layer	Role
Kernel Space	Privileged code; handles core tasks (CPU scheduling, memory, device drivers).
User Space	Runs applications and most system services; cannot directly access hardware.

3.2 Types of Kernels

Type	Characteristics
Monolithic	All OS services run in kernel mode (e.g., Linux).

Introduction to Operating Systems

Type	Characteristics
Microkernel	Minimal kernel (CPU, memory, IPC); other services run in user mode (e.g., Minix, QNX).
Hybrid	Mix of monolithic and microkernel ideas (e.g., Windows NT).
Exokernel	Exposes hardware resources directly; OS policies implemented in libraries.

4. Process & Thread Concepts

Term	Definition
Process	Instance of a program in execution; has its own address space.
Thread	Lightweight execution unit within a process; shares address space.
PCB (Process Control Block)	Data structure holding process state (PID, registers, scheduling info).
Context Switch	Saving current PCB, loading next PCB – enables multitasking.
Scheduling	Algorithms that decide which process/thread runs next.
States	New ? Ready ? Running ? Waiting (Blocked) ? Terminated.
Common	