

OS Services

To Users

- UI : Interfaces for user interaction with the system; either text-based graphical or touch.
- Program Execution: Loads programs into memory, runs them, and also termination
- I/O Operations: Handles input/output devices through system calls instead of direct user access.

OS Services

To Users

- File-System Manipulation: Operations like reading, writing, creating, deleting files and managing permissions.
- Communication: Allows processes to exchange data using shared memory or message passing.
- Error Detection: Monitors and handles hardware, software, and user errors.

OS STRUCTURE

OS Services

To System

- Resource Allocation: Manages CPU, memory, and I/O devices to ensure fair and efficient usage.
- Accounting: Tracks resource usage to analyze performance or enable billing.
- Protection & Security: Prevents unauthorized access and ensures processes don't interfere with each other.

Interfaces to OS

- Command : Shells that accept text commands and execute them.
- GUI: Graphical environment where users interact using icons, windows, and menus.
- Touch Interface: Used in mobile.
- API Interface: Programming interface for accessing OS functions in code.
- System Calls: Low-level functions provided by OS to perform tasks like file handling, process control, etc.

Design & Implementation

- Mechanism vs Policy: Mechanisms define how something is done; policies define what should be done.
- Implementation : mostly written in C for performance and portability.
- Linkers, Loaders: Linkers combine object files into executables; loaders load them into memory to run.
- Static vs Dynamic Linking: Static links libraries at compile time; dynamic links them at run time .
- SYSGEN : The process of tailoring the OS to specific hardware during installation.

OS Architecture Models

- Monolithic Structure
- Layered Structure
- Microkernel
- Modular
- Virtual Machines

OS STRUCTURE

HOW OS STARTS UP

- Bootstrap Program
- Firmware
- Boot Loader
- Kernel Load
- Init System / Services Start

Portability Challenges

- Binary/ABI Incompatibility: Programs compiled for one OS/hardware may not run on another.
- Virtual Machines: abstract the hardware to allow running apps on any platform.
- Cross-platform APIs: help write code once and run on many systems with minor changes.