Operating System

Tuesday, June 24, 2025 9:20 PM

What is an Operating System (OS)?

Definition:

An **Operating System (OS)** is **system software** that acts as an **interface between the user and the hardware**. It **manages hardware resources** and provides **services for application software**.

Key Functions of an Operating System:

Function	Description
1. Process Management	Manages execution of processes, multitasking, and process scheduling.
2. Memory Management	Allocates and deallocates memory to programs.
3. File System Management	Manages data storage, directories, and file permissions.
4. Device Management	Controls and communicates with hardware devices via drivers.
5. User Interface (UI)	Provides CLI (Command Line Interface) or GUI (Graphical User Interface).
6. Security & Access Control	Protects against unauthorized access to data and resources.
7. Networking	Enables computers to communicate over a network.
8. Error Detection	Identifies and handles errors during program execution.

Types of Operating Systems:

Type	Description	Examples
Batch OS	Executes batches of jobs with no user interaction	Early IBM systems
Time-Sharing OS	Multiple users access the system simultaneously	UNIX
Distributed OS	Manages multiple computers as a single system	Amoeba, Plan 9
Real-Time OS	Processes data in real-time with strict timing constraints	VxWorks, RTLinux
Mobile OS	Designed for smartphones and tablets	Android, iOS
Network OS	Supports servers and shared network resources	Novell NetWare, Windows Server

OS Components

Component	Role
Kernel	Core part that directly interacts with hardware
Shell	Interface for user commands (CLI or GUI)
File System	Organizes and manages data storage
Device Drivers	Software that lets the OS talk to hardware
System Utilities	Programs that support OS functions (e.g., disk cleanup)

⚙ Examples of Operating Systems

OS	Type	Popular For
Windows	General-purpose	PCs and enterprise systems
macOS	General-purpose	Apple computers
Linux	Open-source	Servers, developers, education
Android	Mobile	Smartphones, tablets
iOS	Mobile	iPhones, iPads

VS Linux vs Windows

Feature/Aspect	Linux	Windows
Type	Open-source OS	Proprietary OS (licensed by Microsoft)
Cost	Free (most distributions)	Paid (license required)
Source Code Access	Open and customizable	Closed-source (not available to public)
User Interface	CLI + GUI (GNOME, KDE, etc.)	Primarily GUI (user-friendly)
Command Line Use	Heavily used by power users/admins	Optional, rarely needed for common users
Performance	Lightweight, good on older systems	Heavier, requires more system resources
Security	Highly secure, fewer viruses	More vulnerable to malware and viruses
Software Support	Limited proprietary software (e.g., MS Office)	Wide support for commercial applications
Gaming Support	Limited (improving with Steam Proton)	Excellent (direct support for most games)
System Updates	User has full control over updates	Automatic updates (can be forced)
Customization	Highly customizable	Limited customization
File System	ext3, ext4, Btrfs	NTFS, FAT32
Community Support	Large open-source community	Microsoft customer support, forums
Use Case	Developers, servers, cybersecurity	General users, businesses, gaming

Examples	Ubuntu, Fedora, Debian, Arch	Windows 10, Windows 11

Summary Notes

Linux

- Open-source, free, and secure.
- Best for **developers, servers, and advanced users**.
- Requires familiarity with **commands** and **manual configurations**.

Windows

- Paid, user-friendly, and widely supported.
- Best for **general users, businesses, and gamers**.
- Offers **easy GUI navigation** and mainstream software compatibility.

Study Notes Summary

What is OS?

- A system software that connects **user ↔ hardware**.
- Provides a platform to run applications.

Functions

- Manages: CPU, Memory, Devices, Files, Security.
- Offers GUI or CLI for user interaction.

Types

- Batch, Time-sharing, Distributed, Real-time, Mobile, Network OS.

Examples

- Windows, Linux, macOS, Android, iOS.

Components

- Kernel, Shell, File System, Drivers, Utilities.

What is OS Architecture?

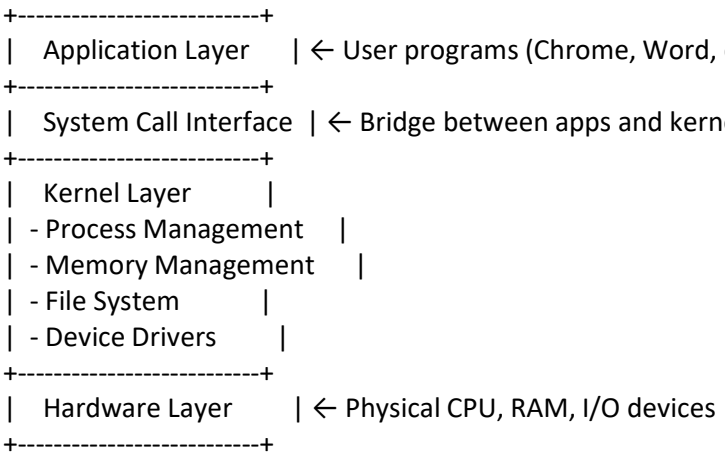
Operating System Architecture refers to the **design structure** of how the OS components are organized and how they interact with hardware and user applications. It defines the **layers** and **modules** that manage system resources.

Types of OS Architectures

Architecture Type	Description
1. Monolithic Kernel	All OS services run in kernel space as one large block.
2. Microkernel	Only essential services run in kernel space; others in user space.
3. Layered Architecture	OS is divided into layers; each layer interacts only with its adjacent layers.

4. Modular Kernel	Hybrid of monolithic and microkernel, allowing loadable modules.
5. Hybrid Architecture	Combines features of monolithic and microkernel for performance and flexibility.

📊 Visual: Layered OS Architecture (Example)



🔍 Components of OS Architecture

Component	Role
Process Manager	Handles process creation, scheduling, and termination
Memory Manager	Manages allocation/deallocation of RAM
File System	Manages data storage, retrieval, and access
Device Drivers	Interface for hardware (keyboard, mouse, printer)
Security Manager	Controls access and ensures system protection
User Interface	CLI or GUI to interact with the system

☑ Study Notes Summary

🌐 Types of OS Architectures:

- **Monolithic**: All-in-one kernel (e.g., early Linux).
- **Microkernel**: Minimal kernel; services in user space (e.g., Minix).
- **Layered**: Organized in layers (e.g., THE OS).
- **Modular**: Kernel with plug-and-play modules (e.g., modern Linux).
- **Hybrid**: Mix of monolithic + microkernel (e.g., Windows, macOS).

🧱 OS Architecture Components:

- **Application Layer**
- **System Call Interface**
- **Kernel (Core Services)**
- **Hardware Layer**