Republic of the Philippines
BOHOL ISLAND STATE UNIVERSITY
Magsija, Balilihan, 6342, Bohol, Philippines
Office of the Director of Instruction
Balance | Integrity | Stewardship | Uprightness

OPERATING SYSTEM FUNDAMENTALS

I. **An operating system (OS)** is system software that manages a computer's hardware and software resources, serving as the essential interface between the user and the device. It performs fundamental tasks like allocating the CPU and memory, handling input/output, managing storage, and controlling peripheral devices. Without an OS, programs would need to include complex code for low-level hardware operations, making the system difficult to use and inefficient. Common examples include Microsoft Windows, Apple macOS, Google Android, and Apple iOS.

Key Functions of an OS

- Resource Management: The OS allocates and manages essential resources such as the CPU, memory, storage, and network connections to different running programs.
- Process Management: It manages the execution of programs, creating, suspending, resuming, and coordinating processes to ensure they get the resources they need.
- Memory Management: The OS controls how memory is used by different programs and users, often using virtual memory to expand memory capacity.
- Device Management: It handles communication with peripheral devices like printers and disk drives, ensuring they function correctly.
- User Interface (UI): The OS provides a way for users to interact with the computer, either through a Graphical User Interface (GUI) (with icons and windows) or a Command-Line Interface (CLI) (with text commands).
- Application Interface: It provides Application Programming Interfaces (APIs) (APIs), which are sets of tools that allow application programs to request services from the operating system.

Components of an OS

- Kernel: The core component that handles essential functions like memory management, process management, and input/output at the hardware level.
- Shell: The outer layer of the OS that manages the user's interaction, taking commands from the user (via a UI) and translating them for the kernel.

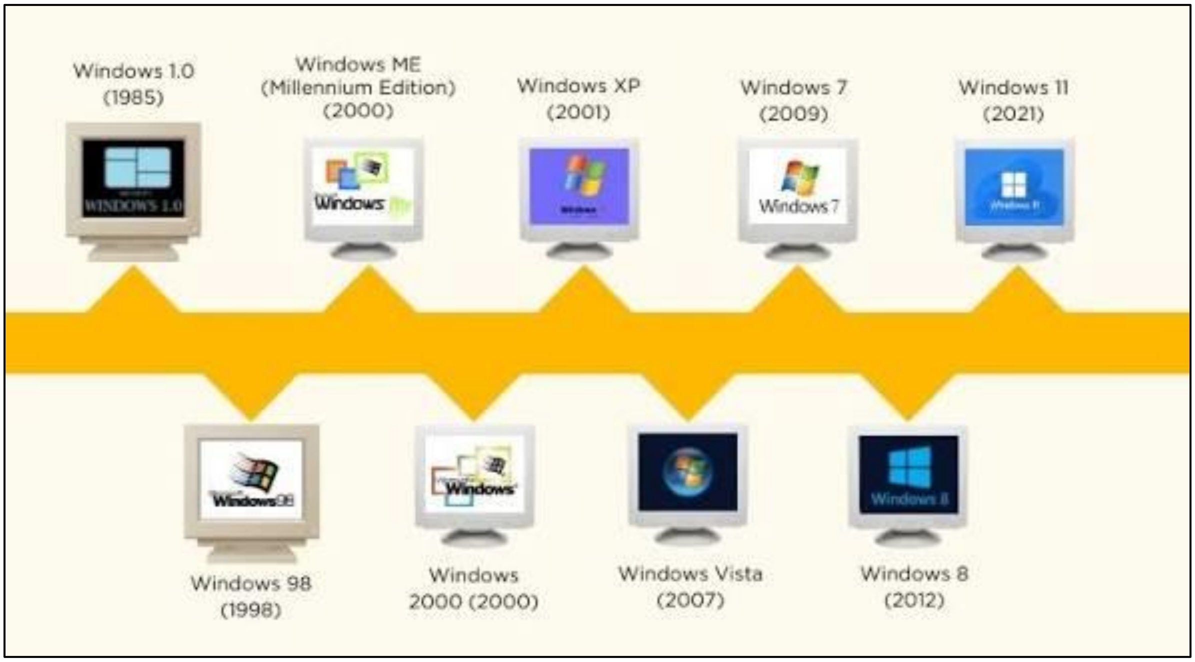
Examples of Operating Systems

- Desktop/Laptop: Microsoft Windows, Apple macOS, Linux.

- Mobile: Google Android (for phones and tablets), Apple iOS (for iPhones and iPads), iPadOS, watchOS.

II. History of Operating Systems

The history of operating systems (OS) is a story of continuous innovation, driven by a need for efficiency and user-friendliness. Starting with manual, single-task processes in the 1940s, OS technology has evolved to enable multiprocessing, time-sharing, graphical user interfaces (GUI), and mobile
connectivity.



Early computing (1940s–1950s): No operating systems

In the first generation of computers, there were no operating systems.

- Manual operation: Programmers had direct control over the hardware and scheduled their own machine time.

- Physical input: Instructions were entered manually using techniques like plugboards, paper tape, and punch cards. These computers ran only one program at a time.

The birth of the OS (1950s–1960s): Batch processing

As computers became more complex and expensive, the goal shifted to maximizing processor utilization.

- Resident monitors: These programs automatically loaded and executed jobs from a storage medium like magnetic tape, eliminating the need for manual setup between tasks.
- Batch processing: Similar jobs were grouped into "batches" and run sequentially. This increased throughput but still had no user interaction during execution.
- First OS: The first working OS was GM-NAA I/O, developed in 1956 by General Motors for its IBM 704 computer.

Efficiency and interaction (1960s–1970s): Multiprogramming and time-sharing

The development of minicomputers and integrated circuits led to more efficient resource usage.

- Multiprogramming: This technique kept multiple jobs in memory at once. While one job was waiting for an input/output (I/O) operation, the CPU would switch to another job, reducing idle time.
- Time-sharing: A logical extension of multiprogramming, this allowed multiple users to share a single computer by giving each user a small slice of CPU time. This enabled interactive computing and led to early multi-user systems.
- UNIX: Developed at Bell Labs in the 1970s, UNIX was a portable, multi-user, and multitasking OS. Its simple design, emphasis on consistency, and availability led to its widespread use and influence on many future operating systems, including Linux and macOS.

The PC revolution (1980s): GUIs and single-user OS

The introduction of microcomputers for personal use shifted the focus from efficiency to user experience.

- MS-DOS: When IBM released its first PC in 1981, it ran on PC-DOS, a version of Microsoft's MS-DOS. This simple, text-based, single-user system became the standard for PCs and a commercial success.

- Graphical User Interfaces (GUIs): The Xerox Alto introduced the first GUI in the 1970s, but it was commercialized by Apple and Microsoft.
    - Macintosh OS (1984): Apple made the GUI and mouse-driven interaction mainstream with its Macintosh computer.
    - Microsoft Windows (1985): Initially a graphical shell on top of MS-DOS, Windows was released as a standalone OS in 1995 with the release of Windows 95.

Connectivity and open source (1990s–2000s): Linux and networking

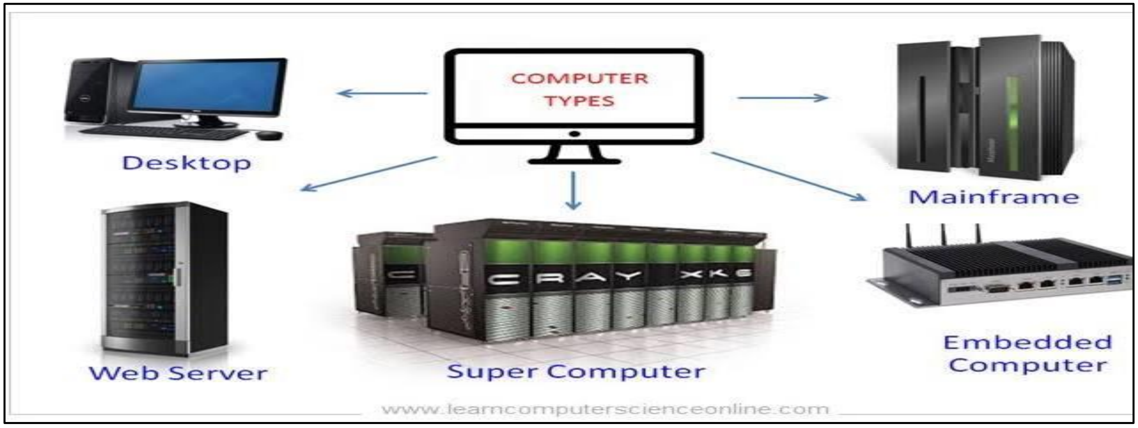
The growing importance of networks and the internet drove new OS developments.

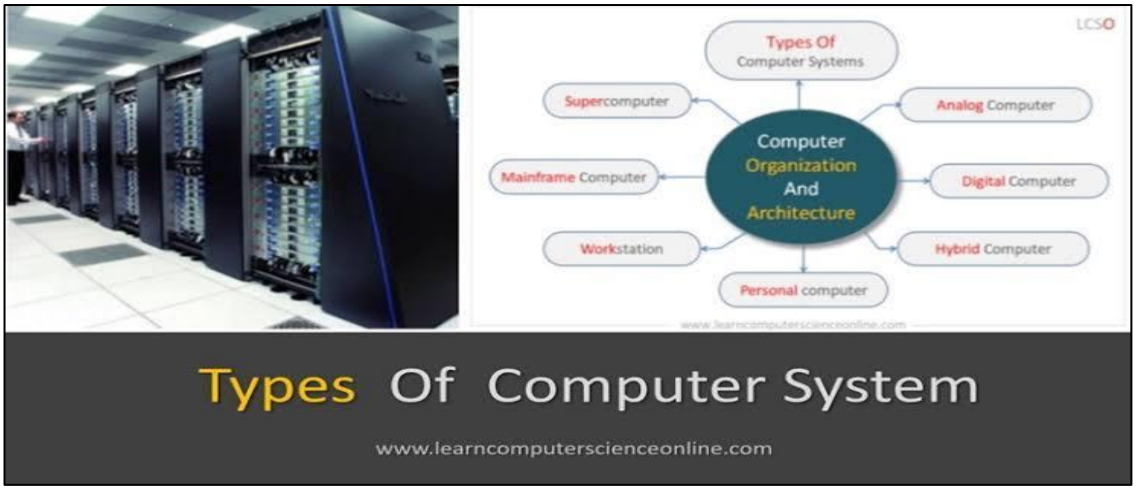
- Linux (1991): Created by Linus Torvalds, Linux is a free, open-source, and highly customizable UNIX-like OS. It gained popularity on servers and is now used on a wide range of devices.
- Windows 9x series and NT: Windows 95 and later versions continued to improve the user experience with features like plug-and-play. Microsoft's server-oriented Windows NT (and its successors) provided greater reliability and security.
- macOS: Based on a UNIX core (Darwin), Apple's macOS combines a powerful, stable foundation with an intuitive GUI.

The mobile and cloud era (2000s–Present): Mobility and virtualization

Today's OSs are defined by connectivity, security, and portability across diverse devices.

- Mobile operating systems: Apple's iOS and Google's Android emerged as the dominant mobile platforms in the late 2000s, bringing powerful, touch-based computing to smartphones and tablets.
- Virtualization and cloud computing: A modern OS can now run under the control of a hypervisor, which allows multiple virtual operating systems to run on a single physical machine. This is the foundation of cloud computing and has reshaped how enterprises deploy software.
- AI integration: The latest frontier involves AI-powered operating systems that learn from user behavior to optimize performance and offer personalized experiences.





III. **Other types of computer systems** include microcomputers (like desktops, laptops, and tablets), minicomputers, mainframes, supercomputers, workstations, servers, and embedded systems. They can also be categorized by their operating principles as digital, analog, or hybrid computers.

Here's a breakdown of some key types:

By Size and Purpose

- **Microcomputers:** These are small, general-purpose computers for individual use, including desktops, laptops, and tablets.
- **Minicomputers:** A mid-range computer designed for multiple users and to perform input, processing, output, and storage activities, with capabilities between a mainframe and a microcomputer.

- **Mainframe Computers:** Large, powerful, and expensive computers used by major corporations like airlines and banks for processing massive amounts of data and serving thousands of users simultaneously.
- **Supercomputers:** The most powerful computers, designed for extremely high-speed computations and large-scale scientific and engineering problems.
- **Workstations:** High-performance computers designed for complex tasks, often used by professionals for engineering, graphics, or software development.
- **Servers:** Computers that provide resources or services to other computers on a network, such as data storage or web page hosting.
- **Embedded Systems:** Small, dedicated computer systems designed for specific functions within a larger device, like in cars, appliances, or industrial control systems.

By Working Principle

- **Digital Computers:** Process data in discrete, digital form (binary 0s and 1s), which is the basis for most modern computers like smartphones and PCs.
- **Analog Computers:** Process continuously changing data, such as physical quantities, and are used in applications like speedometers or mercury thermometers.
- **Hybrid Computers:** Combine aspects of both analog and digital computers, processing continuous and discrete data, which is useful in specialized applications like petrol pumps that convert measurements to price.

IV. Bootable Device

A bootable device is a storage device, such as a hard drive, SSD, USB drive, or CD/DVD, that contains the operating system and necessary boot files and drivers to start a computer. When a computer starts up, it reads instructions from a bootable device to load the operating system or a bootable program, allowing the system to become functional. Examples include a computer's internal hard drive that holds Windows or a USB drive with diagnostic tools.

How it Works

1. **Startup Process**: When you turn on a computer, the system must initiate a startup process.
2. **BIOS/UEFI**: The computer's BIOS or UEFI firmware loads the operating system from the designated bootable device.

3. **Loading Boot Files**: The bootable device provides the essential bootloader program and drivers to load the operating system into the computer's memory.

4. **Full Functionality**: Once the OS is loaded, the computer can perform its normal functions.

Common Examples of Bootable Devices

- **Internal Drives**:

The computer's primary hard drive (HDD) or solid-state drive (SSD) is typically the most common bootable device, containing the installed operating system.

- **USB Flash Drives**:

Can be created to install or repair operating systems, or to run diagnostic software from an external source.

- **CD/DVD Drives**:

Bootable discs containing operating systems or specialized software are less common now but are still used for certain tasks.

- **Network Boot (PXE)**:

Some systems can boot over a network, using a technology called PXE (Preboot Execution Environment).

Why Bootable Devices are Important

- **Operating System Installation**: They are used to install a new operating system on a computer.

- **Troubleshooting**: A bootable USB or CD can be used to run diagnostic tools or access system recovery options when the main operating system is not working.

- **Portable Environments**: They allow you to run a full operating system or specific applications from a portable device, even on another computer.

V. Creating a Bootable Device

To create a bootable device, such as a USB drive, you will need a USB drive, the desired operating system's ISO file (e.g., Windows, Linux), and a bootable USB creation tool like Rufus or the official Microsoft Media Creation Tool. Insert the USB drive, launch the tool, select the ISO file, and follow the prompts to format and write the image to the USB, creating a device that can boot and install an operating system or run recovery tools.

Using Rufus (for various operating systems)

Rufus is a popular third-party tool for creating bootable USB drives for a variety of operating systems.

1. **Download Rufus**: Download the Rufus tool from its official website (rufus.ie).
2. **Launch Rufus**: Run the downloaded executable file.
3. **Insert USB Drive**: Plug in a USB flash drive that is large enough to hold the ISO file.
4. **Select ISO File**: In the "Boot selection" section, click "Select" to find and choose the ISO image of the operating system you want to install.
5. **Configure Settings**: Rufus will automatically select most settings. You may need to choose between MBR for BIOS or GPT for UEFI.
6. **Start the Process**: Click the "Start" button. You will receive a warning that all data on the USB drive will be destroyed, so back up any important files first.
7. **Complete**: Wait for the process to finish, indicated by a "Ready" status.