



Introduction to Operating Systems - Chapter 1

Created @August 24, 2025 11:11 PM

Class IN 3311 Operating Systems

What is an Operating System?

An Operating System (OS) is system software that manages computer hardware and software resources and provides common services for computer programs.

- An **Operating System (OS)** is software that helps you **use your computer easily**.
- It sits **between you and the hardware** (CPU, memory, keyboard, etc.).
- Example: Without an OS, you would have to control hardware directly, which is very hard.

Purpose: Acts as an interface between the user and the computer hardware

Functionalities of an OS

- **Process Management** – Runs programs and manages tasks. (e.g., playing music while browsing the web).
- **Memory Management** – Shares memory among running programs. (e.g., making sure Chrome and Word don't overwrite each other's data).
- **File Management** – Stores and organizes data in files/folders.
- **Device Management** – Controls hardware like printers, monitors, keyboards.
- **Security** – Protects data and prevents unauthorized access.
- **User Interface** – Lets you interact (Windows GUI, Linux terminal, etc.).

Characteristics of an OS

- **Multi-user** → Many people can use the system.
- **Multitasking** → Can run many apps at once.
- **Portability** → Can run on different hardware.
- **Security** → Protects files and apps.
- **Efficiency** → Uses CPU and memory wisely.
- **Scalability** → Can handle bigger workloads (small PC → large server).

Evolution of Operating Systems

The evolution of operating systems (OS) has progressed from simple, hardware-focused systems to complex, user-friendly platforms that integrate AI and cloud computing.

- **1st Gen (1940s–50s):** No OS, programs run manually.

As we know that before 1940s, there was no use of OS . Earlier, people are lacking OS in their computer system so they had to manually type instructions for each tasks in machine language(0-1 based language) . And at that time , it was very hard for users to implement even a simple task. And it was very time consuming and also not user-friendly . Because not everyone had that much level of understanding to understand the machine language and it required a deep understanding.

- **2nd Gen (1950s–60s):** Batch systems (jobs run one after another).

With the growth of time, batch processing system came into the market .Now Users had facility to write their programs on punch cards and load it to the computer operator. And then operator make different batches of similar types of jobs and then serve the different batch(group of jobs) one by one to the CPU .CPU first executes jobs of one batch and then jump to the jobs of other batch in a sequence manner.

- **3rd Gen (1960s–80s):** Multiprogramming & time-sharing.

Multiprogramming was the first operating system where actual revolution began. It provide user facility to load the multiple program into the memory and provide a specific portion of memory to each program. When one program is waiting for any I/O operations (which take much

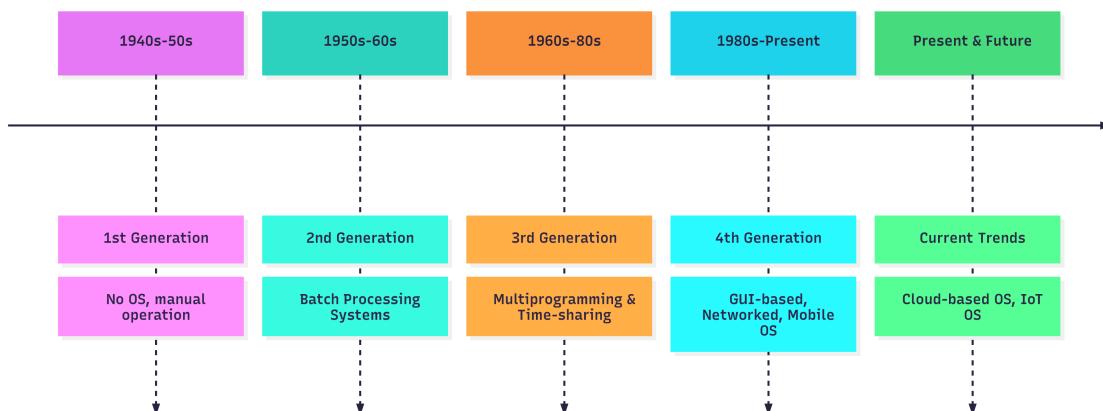
time) at that time the OS give permission to CPU to switch from previous program to other program(which is first in ready queue) for continuous execution of program with **interrupt**.

- **4th Gen (1980s–Now):** GUI-based (Windows, macOS) + Mobile OS.

With the growth of time, Graphical User Interfaces (GUIs) came. First time OS became more user-friendly and changed the way of people to interact with computer. GUI provides computer system visual elements which made user's interaction with computer more comfortable and user-friendly. User can just click on visual elements rather than typing commands. Here are some feature of GUI in Microsoft's windows icons, menus and windows.

- **Now:** Cloud OS, IoT OS (runs on smart devices).

Evolution of Operating Systems

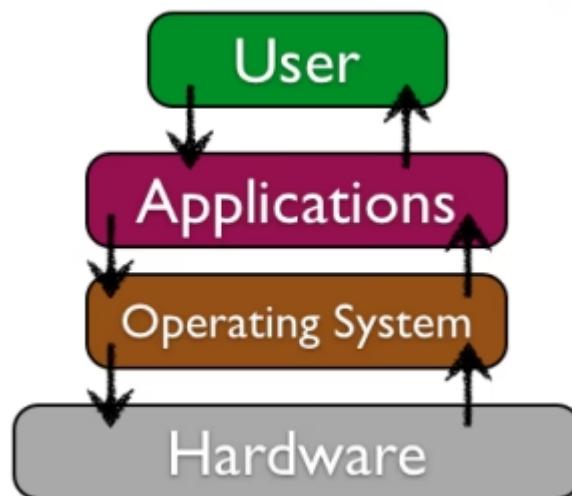


Hardware Concepts (Basic Parts OS Handles)

- **CPU** → Brain, executes instructions.
- **Memory (RAM, Cache, Registers, Disk)** → Stores data temporarily or permanently.
- **I/O Devices** → Mouse, keyboard, printer, etc.
- **System Bus** → Highway that connects CPU, memory, and devices.
- **Interrupts** → Signals that grab CPU's attention (e.g., keyboard press).

OS Interaction with Hardware & Applications

- You → App → OS → Hardware → Output back to you.



- Example:
 - Open a photo → App asks OS → OS fetches from disk → CPU processes → Display on monitor.

Examples of Operating Systems

- **Desktop:** Windows, macOS, Linux
- **Mobile:** Android, iOS
- **Server:** Unix, Windows Server
- **Embedded** (inside devices): FreeRTOS, VxWorks

Program, Job, and Process

- **Program** → A program is non-active set of instructions stored on disk.
- **Job** → When program is **selected to run** (waiting/executing).
A program becomes a job from the moment it is selected for execution until it has finished running and becomes a program again.
- **Process** → Program that is **currently running** in memory.
A process is a program in execution. It is a program that has started but has not finished.

Example:

"MS Word" on disk = **Program**

You double-click it = **Job**

It is running and you are typing = **Process**

In short:

An **OS is like the manager** of a computer. It manages apps, memory, files, devices, and makes everything work together smoothly.