# OPERATING SYSTEMS - INTRODUCTION

# SOFTWARE

## SYSTEM SOFTWARE

- *System software is a type of computer program designed to manage and control computer hardware and application software.*
- Operating Systems: Windows, macOS, Linux, Android, iOS
- Device Drivers: Printer Driver, Sound Card Driver, GPU Driver
- Utilities: Disk Cleanup, Antivirus, File Compression Tools (WinRAR)
- Firmware: BIOS, UEFI(**Unified Extensible Firmware Interface)**

## APPLICATION SOFTWARE

- *Application software is designed to perform specific tasks for the user. It runs on top of the system software.*
- *Office Tools : Microsoft Word, Excel, PowerPoint, Google Docs*
- *Web Browsers: Chrome, Firefox, Safari, Edge*
- *Media Players: VLC Media Player, Windows Media Player*
- *Communication: Zoom, Skype, WhatsApp, MS Teams*
- *Educational Software : Duolingo, MATLAB etc.*

# 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.

✓ Interface between user and hardware

✓ Manages system resources (CPU, memory, I/O)

✓ Executes and controls programs

✓ Ensures system security and efficiency

# FUNCTIONS OF AN OPERATING SYSTEM

- Process Management - The OS handles the execution of programs and processes, including starting, stopping, and managing their execution. It allows multiple programs to run concurrently by sharing the CPU and other resources. Eg: Running a browser, a music player, and a download manager at the same time.
- Resource Management - The OS manages the computer's resources, including the CPU, memory, storage devices, and input/output devices. It allocates these resources to different programs and processes, ensuring efficient and fair usage. Eg: Ensuring a background virus scan doesn’t slow down a video call.
- Memory management - Allocates and deallocates memory space as needed by programs. Keeps track of each byte in memory and who is using it. Ensures processes do not interfere with each other’s memory (protection).

# FUNCTIONS OF AN OPERATING SYSTEMS

- File system Management - The OS provides a way to organize, store, retrieve, and manage files on the computer's storage devices. Eg: Reading, writing, and saving files on your hard drive or SSD.
- Device management - Manages communication between hardware devices and the system. Uses device drivers to operate peripherals (keyboard, mouse, printer, etc.). Eg: Printing a document while listening to music.
- User interface - The OS provides a user interface (GUI or command-line) that allows users to interact with the computer and its applications.
- Security and access control - Ensures that unauthorized users do not access the system. Protects system data and user information such as user login and file permission settings.

# TYPES OF OPERATING SYSTEMS

They are classified based on various factors like functionality, purpose, and the environment they operate in.

- Mainframe Systems
- Desktop Systems
- Multiprocessor Systems
- Distributed Systems
- Clustered Systems
- Real-Time Systems
- Handheld Systems

# MAINFRAME SYSTEMS

- Used for large-scale data processing - They are used by large organizations to process massive amounts of data and perform complex tasks such as transaction processing, enterprise resource planning, and census data processing.
- Supports many users concurrently
- High I/O throughput
- Terminal-based access – A dumb terminal with no processing capability connected to the mainframe. Mainframes often used time-sharing to allow multiple users to access the system concurrently through their terminals.
- Examples include processing credit card transactions, managing customer accounts, and handling online orders.  Airlines use mainframes for booking and managing flights.  Well-suited for processing large batches of data, such as payroll, billing cycles, and inventory management.

- **Common Mainframe Operating Systems**

**1. z/OS (by IBM)**

- **Most widely used mainframe OS** today.
- Designed for IBM's **zSeries** mainframes.
- Supports **virtualization, multi-user, batch and transaction processing**.
- Compatible with modern software (e.g., Java, C++). Used by banks, insurance companies, and governments.

**2. z/VM (Virtual Machine)**

- Hypervisor operating system (a type of operating system that allows multiple operating systems to run concurrently on a single physical machine, essentially acting as a virtual machine manager (VMM)) for **running multiple virtual machines** on one IBM mainframe.
- Each virtual machine can run its own OS (like z/OS, Linux).
- Ideal for **cloud environments** and **test/dev platforms**.

**3. z/VSE (Virtual Storage Extended)**

- Lightweight, efficient OS for **smaller mainframe tasks**.
- Good for **batch processing** and simple transaction systems.

**4. Linux on IBM Z**

- IBM supports **Linux distributions** (e.g., Red Hat, SUSE, Ubuntu) on mainframes.
- Offers flexibility for **modern workloads**, cloud computing, and **open-source** environments.

5. TPF (Transaction Processing Facility)

- Specialized OS for high-volume transaction systems.
- Extremely fast and efficient.
- Used by airlines, credit card networks, telecom, etc.
- Designed for environments needing millions of transactions per second.

# DESKTOP SYSTEMS

- Designed for single-user environments
- GUI-based, interactive systems
- Supports multitasking Examples: Windows, Linux, macOS
- These are the most common type of systems used in homes and offices for general purposes.

# MULTIPROCESSOR SYSTEMS

- A computer system with multiple CPUs (or cores) to enhance processing power and efficiency. It allows multiple tasks to be executed concurrently by distributing workloads across different processors, thus improving system performance and throughput.
- Features and Benefits:
- Increased Processing Power:
- Multiple processors work together to handle complex tasks and execute multiple processes simultaneously, leading to faster overall processing.
- Improved Throughput:
- By distributing tasks across processors, the system can handle more jobs concurrently, increasing the number of tasks completed in a given time.

- Fault Tolerance:
- If one processor fails, the system can continue to operate, potentially with reduced performance, but without a complete shutdown, due to redundancy.
- Resource Sharing:
- Multiple processors share resources like memory, input/output devices, and the system bus.

- Types of Multiprocessor Operating Systems:
- Symmetric multiprocessing (SMP):
- All processors are treated equally, and the operating system can schedule tasks on any available processor.
- Asymmetric multiprocessing (ASMP):
- Certain processors are assigned specific tasks or functions, while others handle general processing.

# DISTRIBUTED SYSTEMS

- A distributed operating system (DOS) manages a network of independent computers, making them appear as a single, cohesive system to users and applications. It enables multiple computers to work together, sharing resources and distributing workloads, which enhances performance and reliability.
- Examples: Cloud platforms, web services

- Solaris: A Unix-based operating system known for its distributed computing capabilities.
- Mach: A microkernel-based operating system that can be used as a foundation for distributed systems.
- Locus: An early distributed operating system that provided a transparent distributed file system.
- **Google's Fuchsia OS:** An OS that might work across several devices each having distributed processing capability.
- **Plan 9 from Bell Labs:** Distributed OS.
  It has the reputation for being able to make the entire network of computers appear as one system.
- **Inferno:** Also developed by **Bell Labs**. Designed for **distributed and portable** environments. Can run on many different platforms (embedded systems, PCs).

# CLUSTERED SYSTEMS

- A **Clustered Operating System** is an OS that manages **two or more interconnected computers (nodes)** to work together as a **single system**. These computers are often connected via a **high-speed network** and are used to improve **performance, availability,** and **scalability**.

❖**Features:**

- High Availability- If one node fails, another can take over (failover support).
- Load Balancing - Distributes workload evenly across nodes.
- Scalability - More nodes can be added to improve performance.
- Resource Sharing - Nodes share storage, CPU, and memory resources efficiently.
- Central Management  Appears as a single system to users/administrators.

## ❖Types

- High-Availability (HA) Clusters - Focused on reliability and uptime (e.g., for databases or websites).
- Load-Balancing Clusters - Spread the workload across systems (e.g., for web servers).
- High-Performance (HPC) Clusters - Designed for scientific computing, simulations, or big data tasks.

## ❖Examples

- Microsoft Windows Server Failover Clustering, Red Hat Cluster Suite, Oracle Real Application Clusters (RAC) etc.

# REAL-TIME SYSTEMS

- A **Real-Time Operating System (RTOS)** is an operating system designed to **respond to events or inputs immediately, within a guaranteed time limit.**

❖Features

- Deterministic timing - Always responds within a predictable time (hard deadlines).
- Prioritized tasks - Critical tasks are given higher priority over others.
- Minimal latency - Very short delay between input and response.
- Multitasking - Manages multiple tasks, often with strict time constraints.
- Small footprint - Lightweight and optimized for embedded systems.

❖Types

- Hard RTOS - Missing a deadline is considered system failure. Strict time limits. Common examples include flight control systems, missile guidance systems, and nuclear reactor control systems.
- Soft RTOS - Deadlines are important, but missing one occasionally is acceptable. Examples are Online games, multimedia applications etc.

❖Examples

- FreeRTOS - Embedded devices, IoT systems
- VxWorks - Aerospace(navigation systems), medical equipment (pacemakers)
- RTLinux - Real-time version of Linux for industrial applications
- QNX - Cars(engine control), telecom, industrial control (automation)
- TinyOS - Sensor networks, wireless embedded systems

# HANDHELD SYSTEMS

- A **Handheld Operating System** is an OS designed specifically to run on **portable devices** like:
- Smartphones
- Tablets
- Wearables (e.g., smartwatches)
- Mobile gaming devices
- These systems are optimized for **touch input, limited hardware resources, mobility,** and **battery efficiency.**

## ❖Features

- Touchscreen support - Designed for finger/stylus interaction
- Battery efficiency - Optimized for long usage on small batteries
- Connectivity support - Built-in support for Wi-Fi, Bluetooth, GPS, 4G/5G
- Lightweight design - Uses minimal system resources (RAM, CPU)
- App ecosystem - Supports app installation via stores (e.g., Play Store, App Store)
- Sensors integration - Works with gyroscope, accelerometer, camera, etc.

❖Examples

- Android - Most popular OS for smartphones and tablets
- iOS - Apple's iPhones and iPads
- HarmonyOS - Huawei phones, tablets, smart devices
- Wear OS - Smartwatches (by Google)
- watchOS - Apple Watches