# Unit 2: Overview of Operating System

## Subject: Introduction to IT System

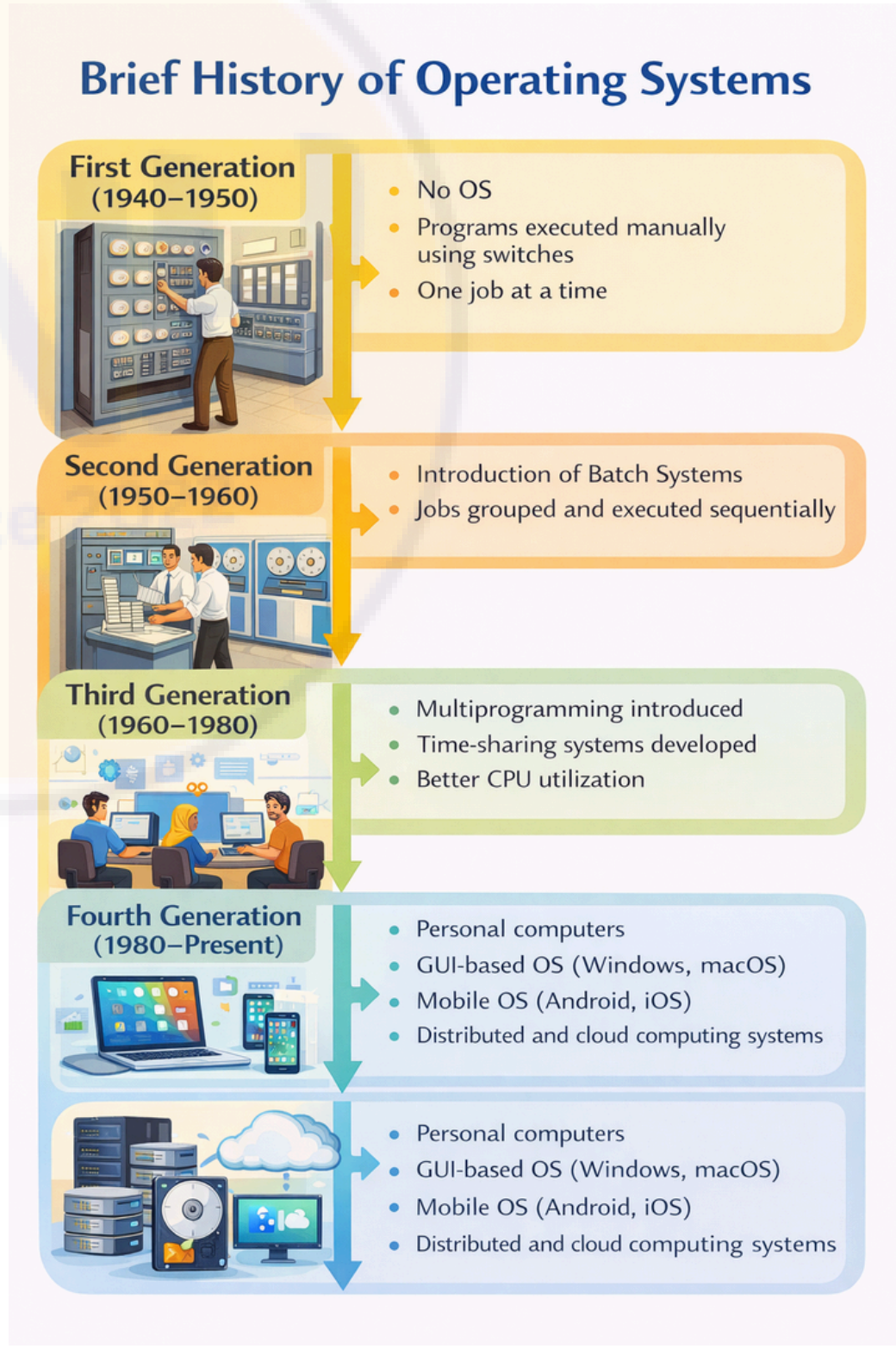
**What is an Operating System (OS)?**
An Operating System (OS) is system software that manages computer hardware and software resources and provides services for computer programs. It acts as an intermediary between users and the hardware.

**Main Objectives of OS**
- To make the computer system convenient to use
- To use hardware efficiently
- To manage system resources effectively

**Core Functions**
- Process Management: Handles creation, scheduling, and termination of processes
- Memory Management: Allocates and deallocates memory space
- File System Management: Organizes and manages files and directories
- Device Management: Controls input/output devices
- Security: Protects system and data from unauthorized access

**Brief History of Operating Systems**

1. First Generation (1940-1950)
   - No OS
   - Programs executed manually using switches
   - One job at a time
2. Second Generation (1950-1960)
   - Introduction of Batch Systems
   - Jobs grouped and executed sequentially
3. Third Generation (1960-1980)
   - Multiprogramming introduced
   - Time-sharing systems developed
   - Better CPU utilization
4. Fourth Generation (1980-Present)
   - Personal computers
   - GUI-based OS (Windows, macOS)
   - Mobile OS (Android, iOS)
   - Distributed and cloud computing systems




**Computer System Review**
A computer system consists of four main components:

1. Hardware
   - CPU (Central Processing Unit)
   - Memory (RAM)
   - Input/Output Devices
2. Operating System
   - Controls hardware
   - Acts as resource manager
3. Application Programs
   - User-level software like browsers, editors
4. Users
   - End-users or other systems

**Types of Operating Systems**

1. Batch Operating System
- Executes jobs in batches
- No interaction with user

2. Time-Sharing Operating System
- Multiple users share CPU time
- Interactive system

3. Distributed Operating System
- Uses multiple computers
- Works as a single system

4. Network Operating System
- Provides services over a network

5. Real-Time Operating System (RTOS)
- Provides immediate response
- Used in critical systems like medical devices

1. Batch Operating System – Examples
Payroll systems in large companies
Bank statement generation
Electricity bill processing systems

2. Time-Sharing Operating System – Examples
UNIX operating system
Linux operating system
Windows operating system

Real-life use:
- Multiple users working on a server
- Students using computers in a lab simultaneously

3. Distributed Operating System – Examples
Amoeba distributed operating system
Plan 9 operating system

Real-life use:
- Cloud computing systems
- Distributed databases

4. Network Operating System – Examples
Windows Server operating system
Novell NetWare operating system

Real-life use:
- Office networks
- File sharing servers
- Printer sharing systems

5. Real-Time Operating System (RTOS) – Examples
VxWorks real-time operating system
QNX real-time operating system

Real-life use:
- Air traffic control systems
- Medical monitoring machines
- Industrial robots

## Operating System Architecture

1. Monolithic Architecture
- Entire OS runs as a single program
- Fast but difficult to maintain

2. Layered Architecture
- OS divided into layers
- Each layer performs specific functions

3. Microkernel Architecture
- Only essential services in kernel
- Better security and stability

4. Hybrid Architecture
- Combination of multiple architectures
- Used in modern OS like Windows

## Batch Operating System

Definition
A Batch Operating System is a type of operating system in which similar jobs are grouped together (in batches) and executed one after another without any interaction between the user and the system during execution.

Working of Batch Operating System
- Users prepare their jobs (program + data) and submit them to the operator.
- Jobs are collected and grouped into batches based on similarity (e.g., same program type).
- These batches are placed in a job queue.
- The operating system processes each batch sequentially.
- Once execution starts, there is no interaction with the user until completion.

Key Features
- No direct communication between user and computer
- Jobs are executed automatically in sequence
- Uses spooling (Simultaneous Peripheral Operations Online)
- Suitable for repetitive and large-volume tasks
- High throughput (more jobs processed in less time)

Examples of Batch Processing
- Payroll system in companies
- Bank interest calculation
- Electricity bill generation
- Exam result processing

Advantages
- Efficient for large jobs
- Less idle time

Disadvantages
- No user interaction
- Debugging is difficult

## Multiprogrammed Batch System

Definition
A Multiprogrammed Batch System is an advanced version of the batch operating system where multiple jobs are kept in memory at the same time, and the CPU switches between them to improve efficiency.

Need for Multiprogramming

In a simple batch system, the CPU remains idle during I/O operations. To overcome this problem, multiprogramming was introduced.

Goal:

- Keep CPU always busy
- Increase system performance

Working of Multiprogrammed Batch System

- Multiple jobs are loaded into main memory.
- The CPU starts executing one job.
- If that job requires I/O operation, the CPU switches to another job.
- This process continues, ensuring CPU is never idle.

Key Features

- Multiple jobs in memory simultaneously
- CPU scheduling is used
- Switching between jobs (context switching)
- Efficient use of CPU and resources

Advantages

- Maximizes CPU Utilization
- Improves Throughput
- Better Resource Utilization

Disadvantages

- Complex Memory Management
- CPU Scheduling Required
- Possibility of Deadlock

Example

- Early mainframe systems
- Banking systems processing multiple transactions

**Time-Sharing System**

Definition

A Time-Sharing Operating System allows multiple users to share system resources simultaneously by dividing CPU time into small units called time slices (quantum).

Objective

- Provide fast response time
- Enable interactive computing
- Allow multiple users to work at the same time

Working of Time-Sharing System

- Many users are connected to the system via terminals.
- Each user is assigned a small time slice.
- CPU executes each user's task for a short time.
- Then switches to the next user (context switching).
- This happens so fast that users feel they have a dedicated system.

Key Features

- Multi-user system
- Interactive environment
- Fast response time
- Uses scheduling algorithms (like Round Robin)

Advantages
- Quick Response Time
- Interactive System
- Efficient CPU Usage
- Fair Resource Sharing

Disadvantages
- Complex System Design
- Security Issues
- System Overhead
- Dependence on CPU Speed

Examples of Time-Sharing Systems
- UNIX operating system
- Linux operating system
- Multi-user servers in colleges and offices

**Computer System Structures**

A computer system is made up of several interconnected components that work together to perform tasks efficiently. The Operating System controls and coordinates these components.

**1. CPU (Central Processing Unit)**

Definition
The CPU is the brain of the computer that performs all calculations and executes instructions given by programs.

Main Functions
- Executes program instructions
- Performs arithmetic and logical operations
- Controls the operation of all other components

Components of CPU
**1. Arithmetic Logic Unit (ALU)**
- Performs mathematical operations (addition, subtraction)
- Performs logical operations (AND, OR, NOT)

**2. Control Unit (CU)**
- Directs operations of the processor
- Fetches instructions from memory
- Decodes and executes instructions

**3. Registers**
- Small, high-speed storage inside CPU
- Store temporary data and instructions

**Batch in Operating System**

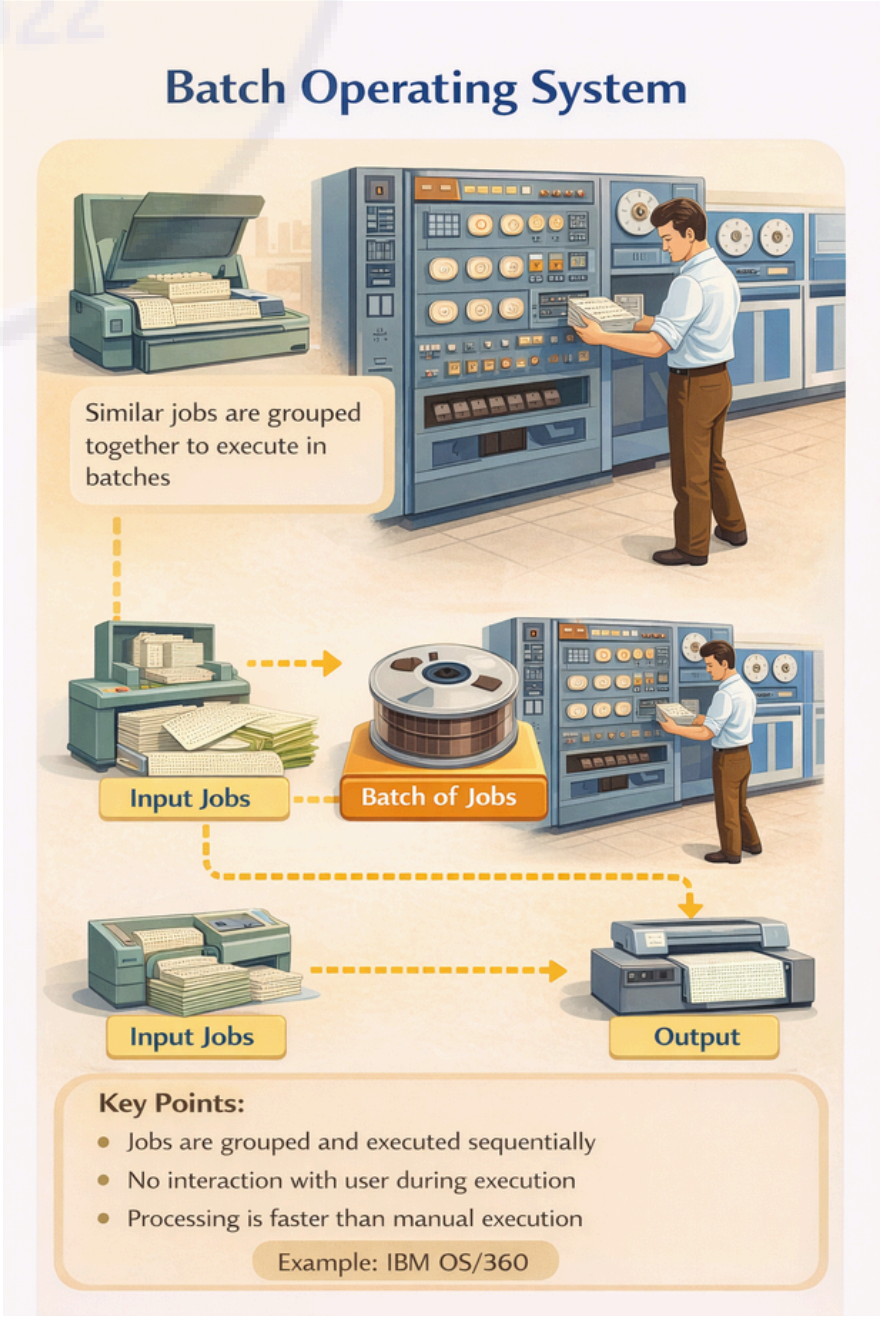
A Batch Operating System is a type of operating system where similar jobs are collected, grouped into batches, and executed one after another without any user interaction.

How It Works
- Users prepare jobs (programs along with required data).
- Jobs are submitted to the system.
- The operating system groups similar jobs into a batch.
- Jobs are executed sequentially.
- Output is generated and provided later.

Example
Early systems like IBM OS/360 used batch processing.

Practical examples:
- Payroll system processing salaries of all employees at once
- Bank transaction processing at the end of the day

Characteristics
- No user interaction during execution
- Jobs are processed sequentially
- Uses job scheduling techniques
- High throughput but longer response time



Advantages
- Efficient for handling large volumes of data
- Better utilization of CPU
- Minimizes idle time
- Suitable for repetitive tasks

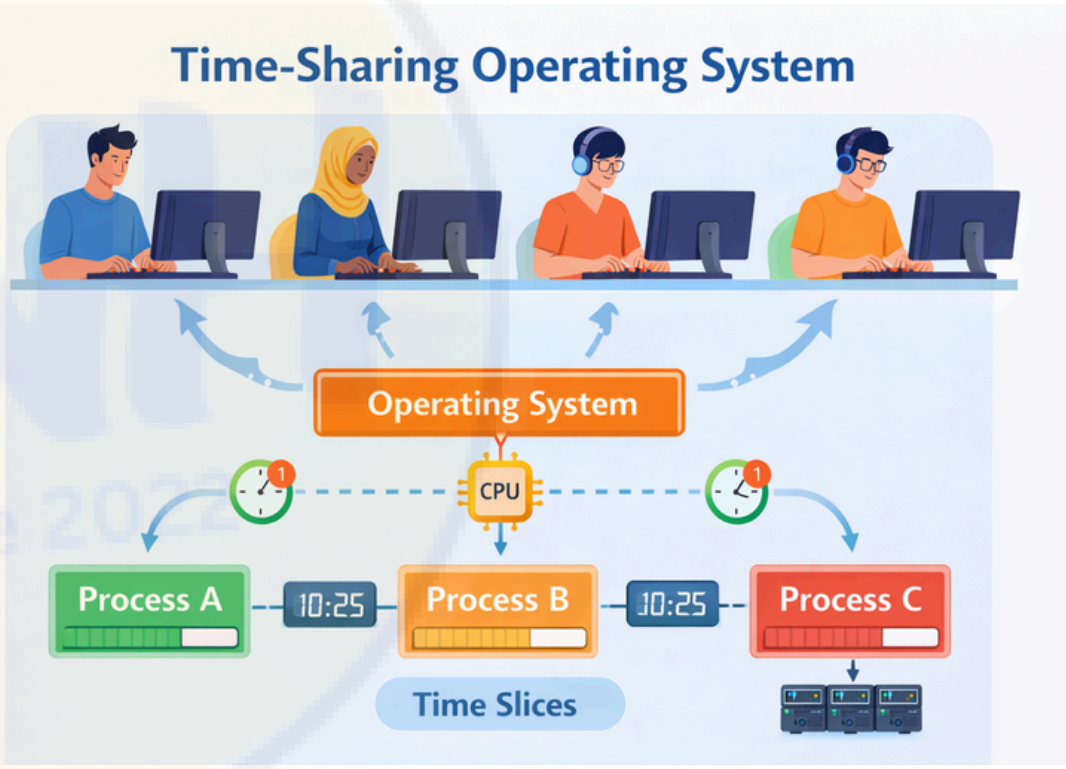
Disadvantages
- Long turnaround time
- Difficult to debug errors
- No control after job submission
- Not suitable for real-time applications

**Time-Sharing Operating System**

A Time-Sharing Operating System is an OS that allows multiple users to share the CPU simultaneously by dividing CPU time into small time slices (time quantum).

How It Works
- The CPU switches rapidly between users/processes.
- Each process gets a small amount of CPU time.
- This switching is so fast that users feel they are working simultaneously.

Example
- Systems like UNIX are based on time-sharing.




Characteristics
- Multi-user system
- Interactive in nature
- Fast response time
- Uses scheduling algorithms (like Round Robin)

Advantages
- Quick response to users
- Efficient CPU utilization
- Multiple users can work at the same time

Disadvantages
- Security issues can arise
- System becomes slow if too many users
- Requires complex scheduling

**Multiprogramming Operating System**

A Multiprogramming Operating System is an OS that keeps multiple programs in memory at the same time and executes them to maximize CPU utilization.

How It Works
- Several jobs are loaded into memory.
- The CPU executes one job.
- When that job waits for input/output, the CPU switches to another job.
- This reduces CPU idle time.

Example
- Early systems like IBM OS/360 supported multiprogramming.

Characteristics

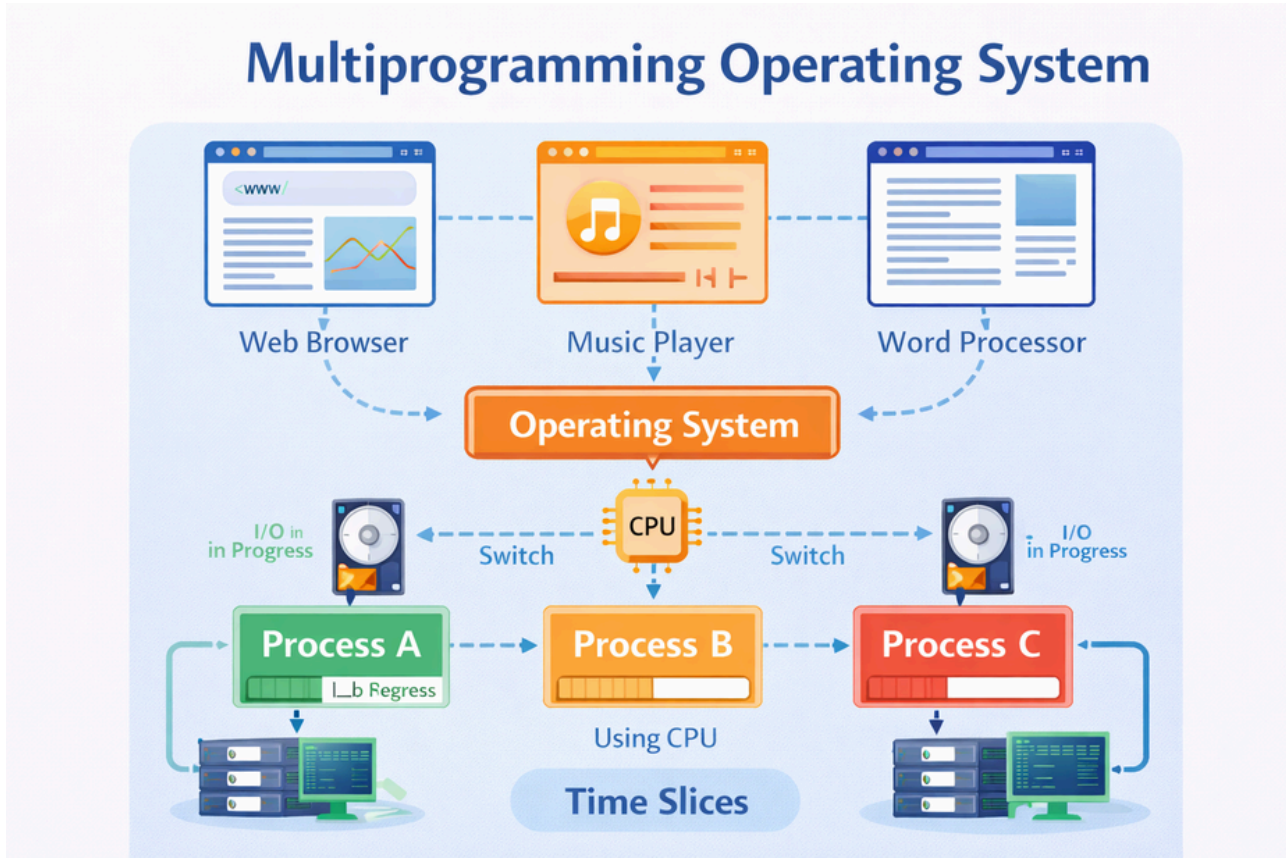
- Multiple programs in memory
- CPU switching between processes
- Non-interactive system
- Focus on CPU efficiency

Advantages

- High CPU utilization
- Increased system efficiency
- Reduced idle time

Disadvantages

- Complex memory management
- Difficult scheduling
- No direct user interaction




**Difference between Batch OS, Time-Sharing OS, and Multiprogramming OS**

| Feature | Batch Operating System | Time-Sharing Operating System | Multiprogramming Operating System |
|---|---|---|---|
| **Definition** | Executes jobs in batches without user interaction | Allows multiple users to share CPU using time | Executes multiple programs by keeping |
| **User Interaction** | No interaction | Interactive | No direct interaction |
| **Number of Users** | Single user | Multiple users | Single user (mostly) |
| **CPU Utilization** | Moderate | High | Very High |
| **Response Time** | Long | Very short | Medium |
| **Execution Method** | Sequential processing | Time slicing (switching rapidly) | Switches when one job waits (I/O) |
| **Complexity** | Simple | Complex | More complex |
| **Main Goal** | Process large volumes of jobs | Provide fast response to users | Maximize CPU utilization |
| **Example** | IBM OS/360 | UNIX | IBM OS/360 |

**Computer System Structure (Operating System)**

A Computer System Structure refers to the basic organization and components of a computer system and how they interact with each other to perform tasks.

**Basic Components of Computer System**

1. Input/Output Devices
- Used to interact with the computer
- Examples: Keyboard, Mouse, Monitor, Printer

2. Central Processing Unit (CPU)

- The brain of the computer
- Performs calculations and executes instructions

Consists of:

- Arithmetic Logic Unit (ALU) – performs calculations
- Control Unit (CU) – controls operations
- Registers – small high-speed storage

3. Memory Unit

A. Primary Memory

- Fast and directly accessed by CPU
- Examples: RAM, Cache

B. Secondary Memory

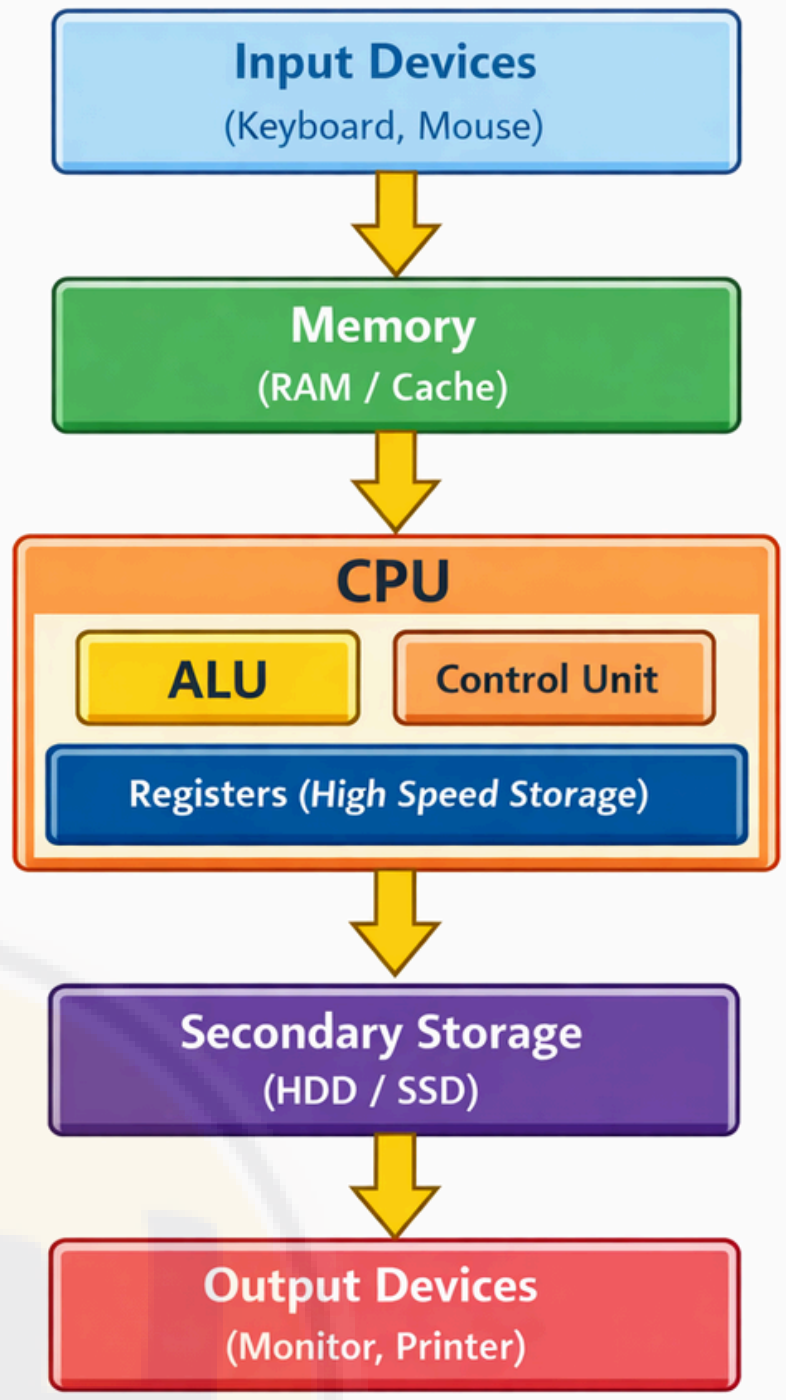
- Permanent storage
- Examples: Hard disk, SSD

4. System Bus

- Connects CPU, memory, and I/O devices

Types:

- Data Bus
- Address Bus
- Control Bus




**Computer System Organization (Working)**

- Input devices send data to the system
- Data is stored in memory
- CPU processes the data
- Results are stored or sent to output devices

**Role of Operating System**

The Operating System acts as an interface between hardware and user.

Functions:

- Process management
- Memory management
- File system management
- Device management

Examples of OS: Windows, Linux

**Operating System Structure**

The Operating System Structure refers to the way an operating system is designed and organized internally to manage hardware and software resources efficiently.

**Types of Operating System Structures**

1. Simple Structure

No clear division between components
All parts are combined together

Example: MS-DOS

Features:

- Easy to design
- Less secure
- Difficult to maintain

2. Layered Structure

- OS is divided into different layers
- Each layer performs a specific function

Working:

- Lower layer → Hardware
- Upper layer → User interface

Advantages:

- Easy to debug
- Better organization

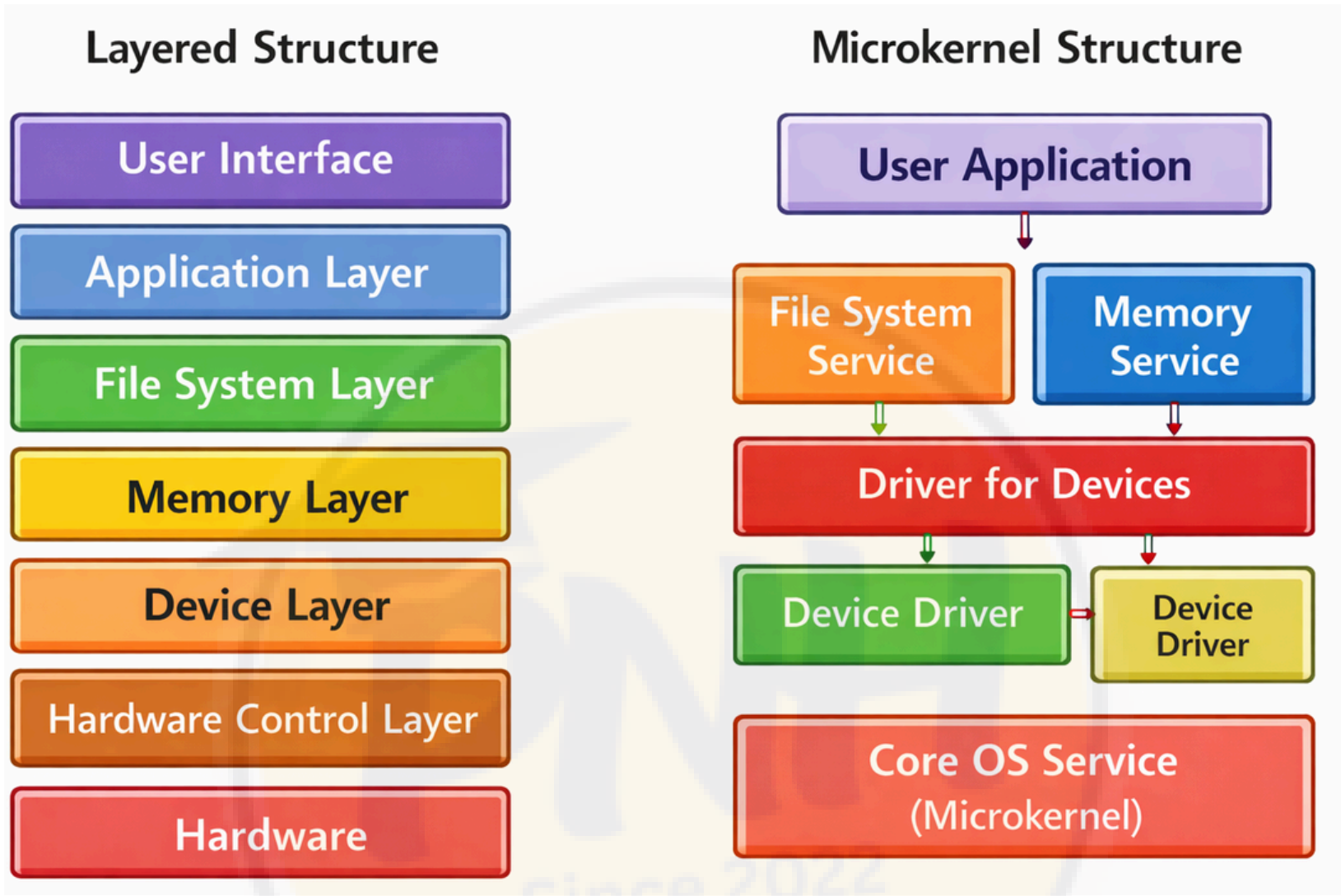
3. Microkernel Structure

- Only essential functions are kept in kernel
- Other services run in user space

- Example: MINIX

Features:

- More secure
- More stable
- Slower due to communication overhead




4. Monolithic Structure

Entire OS works as a single large program

Example: Linux

Features:

- Fast performance
- Difficult to manage
- Less modular

5. Modular Structure

- OS is divided into modules
- Modules can be loaded/unloaded

Example: Linux (modern versions)

Features:

- Flexible
- Easy to update
- Better performance than microkernel