

****1. What is an Operating System? Explain its goals. (10 Marks)**

Answer:

An **Operating System (OS)** is a program that acts as an intermediary between user applications and computer hardware. It controls and coordinates the use of hardware resources among various applications and users.

Goals of an Operating System:

1. **Execute user programs** – It allows users to run applications easily.
2. **Make the system convenient to use** – Provides a user-friendly interface (CLI/GUI).
3. **Efficient use of hardware** – Manages CPU, memory, and I/O devices efficiently.
4. **Resource management** – Allocates system resources properly.
5. **Process coordination** – Enables multiple programs to run smoothly.

Thus, the OS makes the computer system usable and efficient.

****2. Explain the structure of a Computer System. (10 Marks)**

Answer:

A computer system consists of four main components:

1. Hardware

Provides basic computing resources such as:

- CPU
- Memory

- Input/Output Devices

2. Operating System

- Controls and coordinates hardware use.
- Acts as an intermediary between hardware and applications.

3. Application Programs

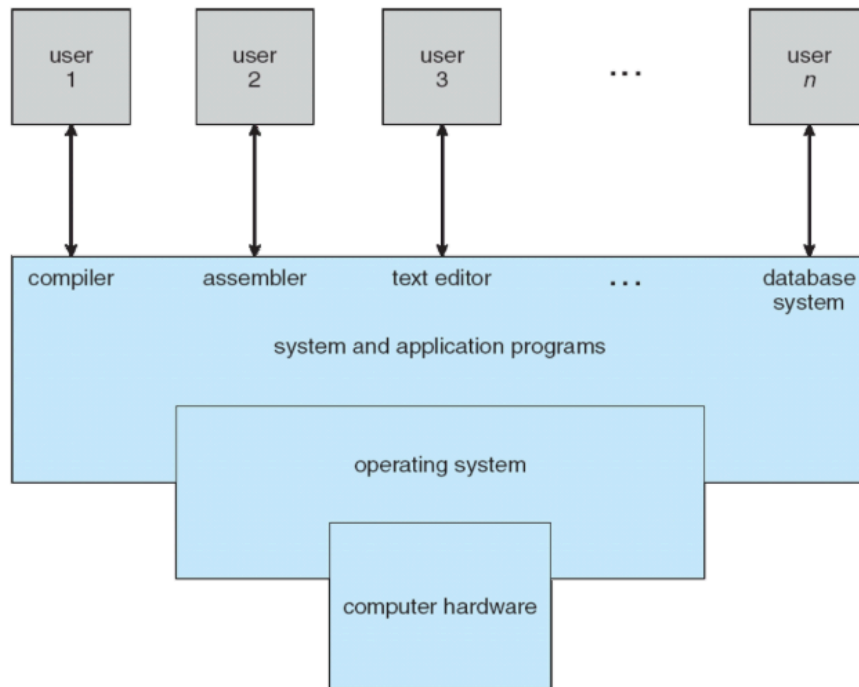
- Define how system resources are used.
- Examples: Word processors, compilers, browsers, database systems.

4. Users

- People, machines, or other computers.

Diagram Representation:

Computer System Structure(6)



3. What is Kernel? Explain its importance. (10 Marks)

Answer:

The **kernel** is the most important program in an operating system. It is the one program that is always running on the computer.

Importance of Kernel:

1. Manages CPU.
2. Manages memory.
3. Controls hardware devices.

4. Handles system calls.
5. Manages processes.

Everything else (system programs and applications) depends on the kernel. Without the kernel, the operating system cannot function.

4. What is a System Call? Explain with examples. (10 Marks)

Answer:

A **system call** is a programmatic way in which a computer program requests a service from the kernel of the operating system.

When a user program needs hardware access (like file reading), it uses system calls.

Examples:

- `fork()` – Creates a new process.
- `exec()` – Executes a new program.
- `read()` – Reads data from a file.
- `write()` – Writes data to a file.

System calls act as an interface between user programs and the operating system.

****5. What happens when we run a program? (10 Marks)**

Answer:

When a program is executed:

Step 1: Compilation

A compiler translates high-level code into an executable file (.exe).

Step 2: Loading

The OS loads the executable file (code and data) from disk into memory (RAM).

Step 3: Execution

The CPU:

- Fetches instruction using the **Program Counter (PC)**.
- Loads data into registers.
- Decodes instruction.
- Executes instruction.
- Stores result in memory.

The OS initializes registers and manages execution.

6. Explain Process and CPU Management by OS. (10 Marks)

Answer:

A **process** is a running program.

OS Responsibilities:

1. Creates processes.
2. Manages process states.
3. Performs CPU scheduling.
4. Shares CPU among multiple processes (Time-sharing).
5. Provides illusion that each process has its own CPU (Virtualization).

Thus, OS manages CPU efficiently and allows multitasking.

7. Explain Memory Management in Operating System. (10 Marks)

Answer:

The OS manages the memory of each process, including:

- Code
- Data
- Stack
- Heap

Key Concepts:

1. Each process thinks it has its own memory.
2. Uses **virtual addresses** starting from 0.
3. OS translates virtual addresses into physical addresses.
4. Ensures memory isolation between processes.

This prevents one process from interfering with another.

8. Explain Device Management in Operating System. (10 Marks)

Answer:

The OS manages external devices using **device drivers**.

Device Driver:

A program that communicates with hardware devices.

Functions:

1. Issues commands to hardware.
2. Handles interrupts.
3. Manages disk, network card, keyboard, etc.
4. Organizes persistent data using a filesystem.

Thus, OS acts as a controller for all hardware devices.

***9. What is an Interrupt? Explain its types, functions and handling mechanism. (10 Marks)

Answer:

Definition of Interrupt

An **interrupt** is a signal sent to the CPU that stops the current process and transfers control to an interrupt service routine.

Types of Interrupts

❶ Hardware Interrupt

- Generated by hardware devices.

Example: When a user presses a key, the keyboard sends an interrupt signal to the CPU.

❷ Software Interrupt (Trap)

- Generated by software programs.
- Also called a **trap**.

Example: When a program calls `read()` or `write()`, it generates a software interrupt.

Functions of Interrupts

Interrupts perform several important functions:

1. **Transfer control** to the interrupt service routine.
2. **Save the current state** of the CPU (program counter, registers).
3. Allow the OS to respond quickly to hardware events.
4. Enable multitasking and time-sharing.
5. Prevent CPU from continuously checking devices (efficient operation).

Interrupt Handling Mechanism

Interrupt handling is the method used by the operating system to respond to an interrupt.

When an interrupt occurs, the OS first **saves the CPU state** (program counter and registers). Then it identifies the type of interrupt using:

- **Polling** – checking devices one by one.
- **Vectored interrupt system** – directly finding the correct interrupt service routine (ISR).

After executing the appropriate ISR, the OS restores the saved state and resumes the interrupted program.

10. Explain Design Goals of an Operating System. (10 Marks)

Answer:

The main design goals are:

1. Convenience

- Make system easy to use.

- Provide user interface (CLI/GUI).

2. Efficiency

- Proper CPU utilization.
- Efficient memory usage.
- Optimal device management.

3. Abstraction

- Hide hardware complexity from users.

4. Process Isolation

- Prevent processes from interfering with each other.

These goals ensure the system is reliable, secure, and efficient.