Basics of Operating System

Introduction:

An operating system serves as an interface between the program and various computer hardware or software components. The operating system is designed to control all of the computer's resources and activities. It acts as an intermediary between the user and the computer hardware, ensuring efficient and secure operation.

Examples of Operating Systems:

- Microsoft Windows
- Linux
- Mac OS
- UNIX
- Android
- iOS

Classification of the Operating System:

1. Multiprocessing:

Supports running a program on multiple CPUs within a single computer system. This improves performance and processing power.

2. Multitasking:

Allows multiple programs to run at the same time by efficiently managing CPU time and resources.

3. Multi-User:

Enables multiple users to access and run programs simultaneously on a single system by managing user permissions and resource allocation.

4. Multithreading:

Allows multiple threads (smallest units of process execution) within a single program to execute simultaneously, enhancing performance.

5. Real-Time:

Responds to inputs immediately, making it suitable for applications requiring time-sensitive processing, such as embedded systems, robotics, and medical devices.

Functions of an Operating System:

1. **Process Management** - Handles the creation, execution, and termination of processes.
2. **Memory Management** - Allocates and manages system memory efficiently.
3. **File System Management** - Organizes and controls data storage on disks.

4. **Device Management** - Controls and manages input/output devices such as keyboards, printers, and storage devices.
5. **Security and Access Control** - Protects system resources and data from unauthorized access.
6. **Networking** - Facilitates communication between computers over a network.
7. **User Interface Management** - Provides interaction between the user and the system through CLI (Command Line Interface) or GUI (Graphical User Interface).

Main Layers in an Operating System:

The software that acts as an interface between various computer parts is organized into layers. A clear benefit of layering is that each layer may be designed independently and interacts as needed.

1. Kernel:

- Links a computer's hardware with application software.
- Controls how applications in RAM access memory.
- Schedules processes and allocates CPU time and memory.

2. Memory Management:

- Manages the allocation and deallocation of memory.
- Handles virtual memory to allow programs to run with more memory than physically available.

3. Input/Output:

- Manages interactions with hardware devices like keyboards, printers, displays, and disk drives.
- Acts as a bridge between the OS and external devices.

4. File Management System:

- Also called the "file system."
- Organizes and oversees data storage on permanent storage devices such as hard drives and SSDs.

5. User Interface:

- Provides the means for user interaction with the computer.
- Types:
    - **Graphical User Interface (GUI):** Uses icons and visual indicators (e.g., Windows, macOS).
    - **Command Line Interface (CLI):** Uses text-based commands (e.g., MS-DOS, Linux Terminal).

Types of Operating Systems:

1. **Batch Operating System** - Processes jobs in batches with minimal user interaction.
2. **Time-Sharing Operating System** - Allocates a fixed time slice to each process for multitasking.

3. **Distributed Operating System** - Runs on multiple machines connected through a network to function as a single system.
4. **Embedded Operating System** - Designed for specialized devices like smartwatches, ATMs, and medical instruments.
5. **Mobile Operating System** - Optimized for mobile devices (e.g., Android, iOS).
6. **Network Operating System** - Manages network resources and allows communication between multiple computers.