NAME : AYESHA IMRAN

ROLL NO: 2023-BSAI-011

Operating System: The Foundation of Computing

An operating system (OS) is the fundamental software that manages a computer's hardware and software resources and provides common services for computer programs. It acts as an intermediary between the user and the hardware, allowing users to interact with the computer in a user-friendly way. Without an operating system, applications would need to directly manage hardware components, a complex and error-prone task. The OS abstracts away these complexities, providing a consistent and reliable platform for software execution.

Key Functions of an Operating System:

The operating system performs a multitude of essential functions, including:

- **Process Management:** The OS manages the execution of programs, called processes. This includes creating and terminating processes, allocating resources like CPU time and memory, and scheduling their execution. It handles multitasking, allowing multiple processes to run concurrently (or appear to run concurrently), and manages inter-process communication.
- **Memory Management:** The OS controls how the computer's memory (RAM) is allocated and used. It keeps track of which parts of memory are currently in use, allocates memory to processes, and reclaims memory when it is no longer needed. Memory management techniques like virtual memory allow the OS to run programs larger than the available physical memory.
- **File System Management:** The OS organizes and manages files and directories on storage devices like hard drives. It provides a structured way to store, retrieve, and manage data. The file system defines how files are named, organized, and accessed, and handles tasks like disk space allocation and file permissions.
- **Input/Output (I/O) Management:** The OS controls communication between the computer and its peripheral devices, such as keyboards, mice, printers, and network interfaces. It provides a standardized interface for applications to interact with these devices, simplifying device driver development and ensuring compatibility.
- **Security:** The OS provides mechanisms to protect the system and user data from unauthorized access. This includes user authentication, access control, and protection against malware.
- **User Interface:** The OS provides a way for users to interact with the computer. This can be a command-line interface (CLI), where users

type commands, or a graphical user interface (GUI), with windows, icons, and menus.

Types of Operating Systems:

Operating systems come in various types, each designed for specific purposes:

- **Batch Operating Systems:** These systems process jobs in batches, without user interaction during execution. They were common in early computing environments.
- **Time-Sharing Operating Systems:** These systems allow multiple users to share the computer's resources simultaneously. Each user gets a small time slice to execute their programs.
- **Real-Time Operating Systems (RTOS):** These systems are designed for applications with strict timing requirements, such as industrial control systems and embedded systems. They guarantee timely responses to events.
- **Distributed Operating Systems:** These systems manage a network of computers, allowing them to work together as a single system.
- **Network Operating Systems:** These systems manage network resources and provide services for network users.
- **Personal Computer Operating Systems:** These are designed for personal computers, such as Windows, macOS, and Linux.
- **Mobile Operating Systems:** These are designed for mobile devices, such as Android and iOS.

Page 2: Core Concepts and Evolution of Operating Systems

Core Concepts:

Several key concepts underpin the functionality of modern operating systems:

- **Kernel:** The kernel is the core of the operating system. It is responsible for managing the system's resources and providing services to applications. It resides in memory at all times and acts as the bridge between the hardware and software.
- **System Calls:** System calls are the interface through which applications request services from the kernel. They provide a way for programs to access hardware and perform privileged operations.
- **Processes and Threads:** A process is an instance of a running program. A thread is a lightweight unit of execution within a process. Multi threading allows a single process to execute multiple parts of its code concurrently.
- **Concurrency:** Concurrency is the ability of the operating system to manage multiple tasks at the same time, even if the system has only one processor. This is achieved through techniques like time-sharing and multitasking.

- **Virtualization:** Virtualization allows multiple operating systems to run on the same physical hardware. This is achieved by creating virtual machines, each of which has its own virtual hardware.
- **Abstraction:** The OS abstracts away the complexities of the hardware, providing a simplified and consistent interface for applications. This allows developers to focus on writing application code without worrying about the specifics of the hardware.

Evolution of Operating Systems:

Operating systems have evolved significantly over the decades:

- **Early Operating Systems:** These were simple batch processing systems with limited functionality.
- **Multiprogramming:** This allowed multiple programs to reside in memory at the same time, improving CPU utilization.
- **Time-Sharing:** This allowed multiple users to interact with the computer simultaneously.
- **Personal Computing:** The development of personal computers led to the creation of user-friendly operating systems like DOS and Windows.
- **Graphical User Interfaces (GUIs):** GUIs made computers easier to use by providing a visual interface with windows, icons, and menus.
- **Networking:** The rise of the internet led to the development of network operating systems and distributed systems.
- **Mobile Computing:** The proliferation of mobile devices led to the development of mobile operating systems like Android and iOS.
- **Cloud Computing:** Cloud computing has driven the development of operating systems capable of managing resources in large, distributed data centers.

Future Trends:

Operating systems continue to evolve, driven by advancements in hardware and software. Some key trends include:

- **Increased Security:** As cyber threats become more sophisticated, operating systems are incorporating stronger security measures.
- **Cloud Integration:** Operating systems are becoming more tightly integrated with cloud services.
- **Artificial Intelligence (AI):** AI is being used to improve operating system performance and efficiency.
- **Serverless Computing:** This paradigm shifts the focus away from managing servers, allowing developers to focus on writing code.
- **Quantum Computing:** The emergence of quantum computing will require the development of new operating systems capable of managing quantum hardware.

Operating systems are a critical component of any computing system. They provide the foundation upon which all other software runs. Understanding the

basic concepts of operating systems is essential for anyone working in the field of computer science.