Introduction to OS

Jayesh Deep Dubey
Technical Officer
C-DAC Patna

What is OS ?

- An operating system (OS) is a software that manages computer hardware resources and provides common services for computer programs.
- It is a program that acts as an intermediary between a user of a computer and the computer hardware.
- Operating system tasks:
  - Execute user programs and make solving user problems easier
  - Make the computer system convenient to use
  - Use the computer hardware in an efficient manner

How is OS different from other application software?

1) **Function**: An operating system is a fundamental software that manages and controls the hardware resources of a computer, whereas other application software (like Microsoft Word or Google Chrome) are designed to perform specific tasks or functions that the user needs to do
2) **Complexity**: Operating systems are generally more complex than other application software because they need to be able to manage a wide range of hardware resources and provide services to other programs running on the computer. They also need to be designed to work with different types of hardware and software configurations.

How is OS different from other application software?

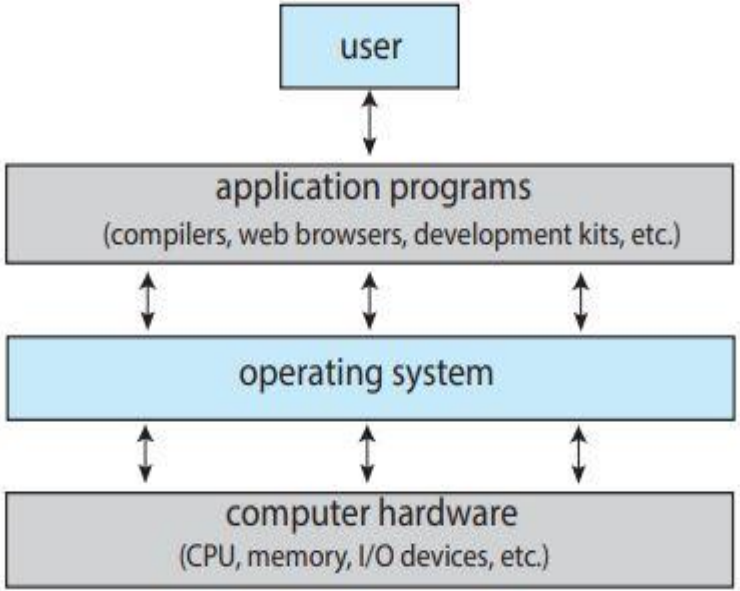
3) **Control:** The operating system has a higher level of control over the computer than other application software. It has the ability to allocate system resources, manage processes, and control access to the computer's hardware, whereas other software is limited to running within the parameters set by the operating system

Goals of Operating System

- Convenience
- Efficiency
- Portability
- Reliability
- Scalability
- Robustness

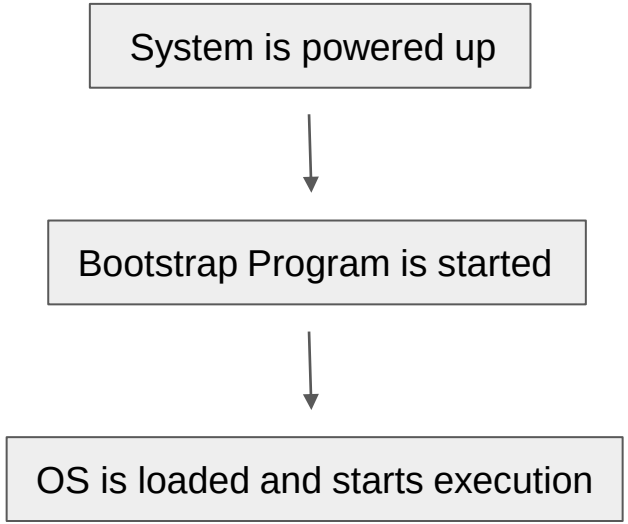
Abstract view of the components of a computer system

.



System Startup Process

.



Operating System Services

- User interface
- Program execution
- I/O operations
- File-system manipulation
- Communications
- Error detection
- Resource allocation
- Accounting
- Protection and security

Components of Operating System

Process Management

- A program in execution is called as a process
- A process needs certain resources (CPU time, memory, files, and I/O devices) to accomplish its task.
- A system consists of a collection of processes, some of which are operating-system processes (those that execute system code) and the rest of which are user processes (those that execute user code)

Components of Operating System

Process Management

- The operating system is responsible for the following activities in connection with process management:
  - Creating and deleting both user and system processes
  - Scheduling processes on the CPUs
  - Suspending and resuming processes
  - Providing mechanisms for process synchronization
  - Providing mechanisms for process communication

Components of Operating System

Memory Management

- The primary goal of memory management is to allocate memory to running programs and ensure that they can access the required memory space.
- Memory is typically organized as a large array of words or bytes, each with its own unique address. This allows the CPU and I/O devices to quickly access data stored in memory by specifying the memory address of the data they need to access.
- There are two main types of memory in a computer system: primary memory and secondary memory.

Components of Operating System

Memory Management

- Primary memory (RAM) is a volatile type of memory that stores data temporarily while the computer is running. It is used to hold the data and instructions that the CPU is currently processing. When the computer is turned off, the data in RAM is lost.
- Secondary memory is a non-volatile type of memory that stores data for long-term use. Unlike primary memory, the data in secondary memory is not lost when the computer is turned off.

Components of Operating System

Memory Management

- The operating system is responsible for the following activities in connection with memory management:
  - Keeping track of which parts of memory are currently being used and which process is using them
  - Allocating and deallocating memory space as needed
  - Deciding which processes (or parts of processes) and data to move into and out of memory

Components of Operating System

I/O System Management

The primary goal of I/O system management is to ensure that all I/O operations are performed efficiently and reliably.

The operating system provides a variety of services and mechanisms to achieve goal of I/O system management:

- **Device Driver**: A device driver is a software component that allows the operating system to communicate with a specific hardware device. Only the device driver knows the peculiarities of the specific device to which it is assigned

Components of Operating System

I/O System Management

- **Interrupts:** The operating system uses interrupts to handle I/O operations. Interrupts are signals that are sent to the CPU to indicate that an I/O operation is complete or that an I/O device requires attention.
- **Buffering:** The operating system uses buffering to store data temporarily when transferring data between I/O devices and memory. This improves the efficiency of I/O operations by reducing the number of times the CPU must access the I/O device

Components of Operating System

I/O System Management

- **Spooling:** The operating system uses spooling to manage print jobs, allowing multiple print jobs to be processed simultaneously without requiring the user to wait for each job to complete

Components of Operating System

File Management

- File management in an operating system refers to the set of functions and tools that are used to create, organize, store, access, and manipulate files and directories on a computer's storage devices.
- A file is a collection of related information defined by its creator.A file can be any type of data, such as a document, image, video, audio, or program code.

Components of Operating System

File Management

- The operating system is responsible for the following activities in connection with file management:
  - Creating and deleting files
  - Creating and deleting directories to organize files
  - Supporting primitives for manipulating files and directories
  - Mapping files onto mass storage
  - Backing up files on stable (nonvolatile) storage media

Components of Operating System

Network Management

- This component enables the computer to connect to and communicate with other devices on a network.
- The OS provides various tools and protocols for managing the network, which include**:**
  - Network configuration
  - Network monitoring
  - Network security
  - Network file sharing

Components of Operating System

Protection and Security

- Protection and security in operating systems (OS) refer to the measures taken to prevent unauthorized access to system resources and ensure the confidentiality, integrity, and availability of data
- Key elements of protection and security in OS:
  - **Access control**: These mechanisms include user authentication, authorization, permissions, and privileges.
  - **Encryption**: OS can use encryption to protect sensitive data from unauthorized access

Components of Operating System

Protection and Security

- Key elements of protection and security in OS:
  - **Firewalls**: OS can use firewalls to prevent unauthorized network access. Firewalls can block incoming and outgoing traffic based on predefined rules and policies
  - **Antivirus software**: OS can use antivirus software to detect and remove malicious software, such as viruses, worms, and Trojans
  - **Backup and recovery**:OS can provide backup and recovery features to protect against data loss and ensure business continuity in case of system failure or disaster
  - **Auditing and monitoring**: OS can provide auditing and monitoring capabilities to track system activities and detect potential security breaches

Types of OS

Embedded System OS

- An embedded system OS (operating system) is a specialized software system designed to manage the resources of an embedded system efficiently and reliably.
- An embedded system is a computer system designed to perform a specific task or function
- Examples of embedded system OS include FreeRTOS, VxWorks, eCos, and QNX. These OSes provide a range of features and capabilities to support various embedded system applications.

Types of OS

Embedded System OS

- Characteristics of an embedded system OS:
  - Real-time performance
  - Small footprint
  - Reliability
  - Customizability

Types of OS

Real Time OS

- A real-time operating system (RTOS) is a specialized software system designed to manage the resources of a computer system, such as memory, processing power, and input/output devices, in a way that meets specific timing constraints.
- RTOS must respond to events or inputs within a fixed and predictable time frame, usually measured in microseconds or milliseconds.
- RTOS is commonly used in applications that require real-time performance, such as industrial control systems, medical devices, aerospace and defense systems, and automotive systems,

Types of OS

Real Time OS

- RTOS typically provides the following features:
  - Real time Task scheduling
  - Fast and efficient Interrupt handling
  - Device Drivers
- Examples of Real time OS includes RTLinux, QNX, FreeRTOS

Types of OS

Mobile OS

- A mobile operating system (OS) is a software system that runs on mobile devices, such as smartphones, tablets, and smartwatches
- Mobile OSes are designed to be lightweight, efficient, and optimized for mobile devices with limited resources, such as memory, processing power, and battery life
- Examples of Mobile OS: Android, iOS

Types of OS

Server Machine OS

- A server operating system (OS) is a software system that runs on a server machine and provides a platform for hosting and managing applications and services that can be accessed by other devices on a network
- Server OSes are designed to be reliable, scalable, and secure, and typically include a range of features, such as:
  - **Network Services**: File sharing, Print services, and Email services
  - **Virtualization**: A server OS can support virtualization, allowing multiple virtual machines to run on a single physical server

Types of OS

Server Machine OS

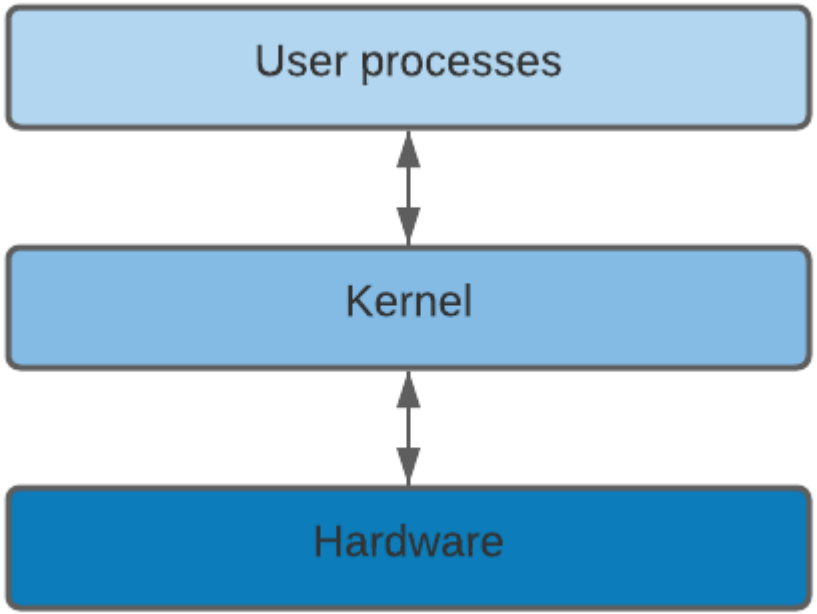
- Features of Server OS:
  - **Security**: A server OS includes security features, such as firewalls, intrusion detection and prevention, and authentication and authorization mechanisms, to protect the server and the data it stores
  - **Remote Administration**: A server OS allows administrators to manage the server remotely, using tools such as remote desktop or SSH
- Examples of Server OS: Windows Server, Linux, macOS Server

Kernel

- Kernel is the central component that provides essential services and manages the system's resources.
- The kernel is responsible for managing system resources such as CPU, memory, input/output (I/O) devices, and network connections
- The kernel also handles system calls, which are requests made by applications for access to resources or services provided by the operating system

Kernel

.

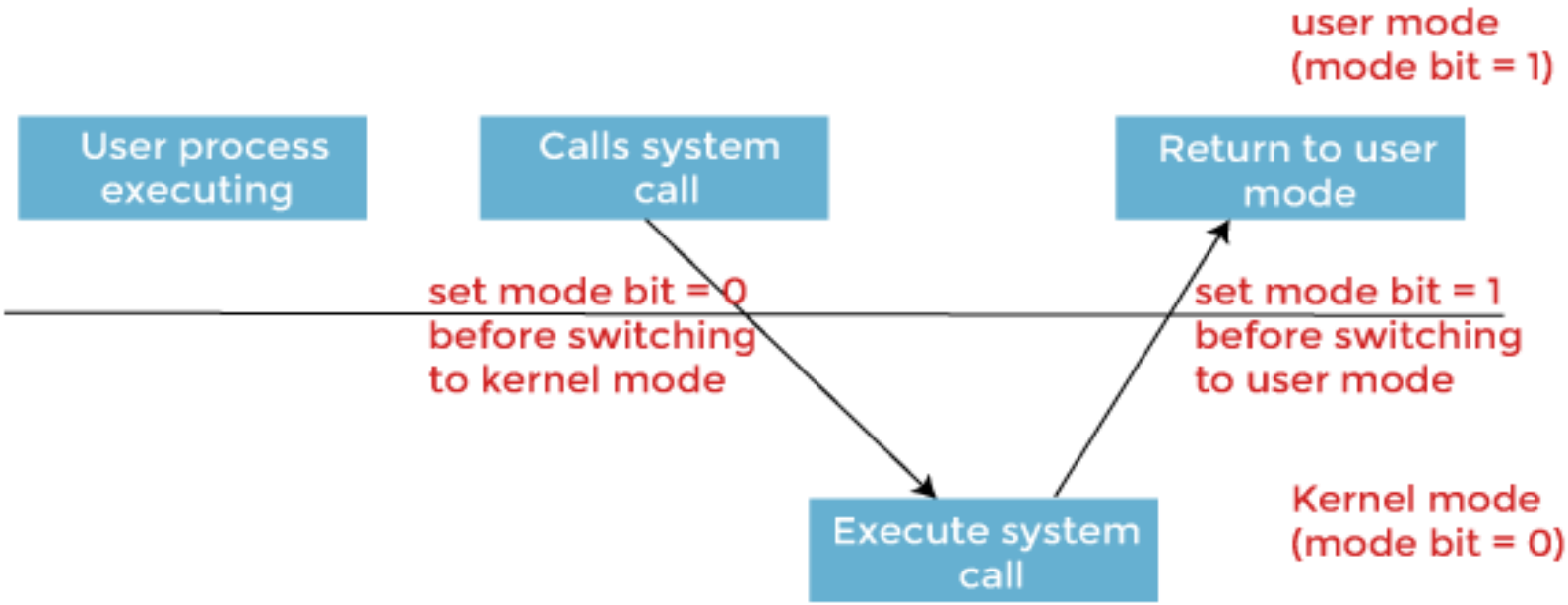


Dual Mode of Operation

- Dual mode of operation refers to a computer system's ability to switch between two different modes of operation: **user mode** and **kernel mode**.
- In **user mode**, the system runs applications, and the applications have limited access to system resources.The user mode is designed to be a safe environment.
- **Kernel mode** is used for tasks that require direct access to system resources, such as managing hardware devices, handling interrupts, and performing system calls.Kernel mode has unrestricted access to system resources.

Dual Mode of Operation

.



System Calls

- System calls are a way for applications to request services from the operating system (OS).
- These services can include accessing hardware devices, managing files, creating and managing processes, and performing other privileged operations that are not available in user space.
- System calls provide a standardized interface between applications and the operating system, enabling applications to perform tasks that require access to system resources.

System Calls

- System calls can be grouped roughly into six major categories:
  - Process control
  - File management
  - Device management
  - Information maintenance
  - Communications
  - Protection

System Calls

- Process Control
  - create process, terminate process
  - load, execute
  - get process attributes, set process attributes
  - wait event, signal event
  - allocate and free memory
- File management
  - create file, delete file
  - open, close
  - read, write
  - get file attributes, set file attributes

System Calls

- Device management
  - request device, release device
  - read, write
  - get device attributes, set device attributes
  - logically attach or detach devices
- Information maintenance
  - get time or date, set time or date
  - get system data, set system data
  - get process, file, or device attributes
  - set process, file, or device attributes

System Calls

- Communications
  - create, delete communication connection
  - send, receive messages
  - transfer status information
  - attach or detach remote devices
- Protection
  - get file permissions
  - set file permissions

Interrupts

- Interrupts are signals sent to a processor by external devices or software, which cause the processor to temporarily halt its current task and switch its attention to the interrupting device or software.
- Interrupts are a way for external devices or software to communicate with the processor and request its attention for a particular task or event
- They enable a multitasking operating system to switch between multiple running programs and perform various background tasks.

Interrupts

- Interrupts can be classified into several types:
  - **Hardware interrupts**: They are generated by external hardware devices, such as keyboards, mouse, or network adapters
  - **Software interrupts:** They are triggered by software, usually by calling a system call.
  - **Exceptions**: interrupts generated by the processor itself in response to exceptional conditions, such as division by zero or a memory access violation.

Interrupt Handling

- When an interrupt is received by a processor, it follows a specific sequence of steps to handle the interrupt:
  1. Save the current state
  2. Determine the source of the interrupt
  3. Disable further interrupts
  4. Call the interrupt handler
  5. Return from the interrupt

Thank You !!