Operating Systems

What is an Operating System

An operating system acts as an intermediary between the user of a computer and computer hardware.

The purpose of an operating system is to provide an environment in which a user can execute programs conveniently and efficiently.

An operating system is software that manages computer hardware.

operating system is the one program running at all times on the computer (usually called the kernel), with all else being application programs.

• Mainframe OS – optimize utilization

• PC – business applications, complex games

• Operating system goals:

• Execute user programs and make solving user problems easier

• Make the computer system convenient to use

• Use the computer hardware in an efficient manner

Functionalities of Operating System

- **Resource Management:** When parallel accessing happens in the OS means when multiple users are accessing the system the OS works as Resource Manager, Its responsibility is to provide hardware to the user. It decreases the load in the system.
- **Process Management:** It includes various tasks like **scheduling and termination** of the process. It is done with the help of CPU scheduling techniques.
- **Storage Management:** The **file system** mechanism used for the management of the storage. **NIFS**, **CIFS**, **CFS**, **NFS**, etc. are some file systems. All the data is stored in various tracks of Hard disks that are all managed by the storage manager. It included **Hard Disk**.
- **Memory Management:** Refers to the management of primary memory. The operating system has to keep track of how much memory has been used and by whom. It has to decide which process needs memory space and how much. OS also has to allocate and deallocate the memory space.

- **Security/Privacy Management:** Privacy is also provided by the Operating system using passwords so that unauthorized applications can't access programs or data. For example, Windows uses **Kerberos** authentication to prevent unauthorized access to data.

Computer system can be divided into four components:

• Hardware – provides basic computing resources
CPU, memory, I/O devices
• Operating system
Controls and coordinates use of hardware among various applications and users
• Application programs – define the ways in which the system resources are used to solve the computing problems of the users
Word processors, compilers, web browsers, database systems, video games
• Users
People, machines, other computers

The process operating system as User Interface:

1. User
2. System and application programs
3. Operating system
4. Hardware

User
Users want convenience, ease of use and good performance
System:
• OS is a resource allocator
Manages all resources
Decides between conflicting requests for efficient and fair resource use
• OS is a control program
Controls execution of programs to prevent errors and improper use of the Computer

Purposes of an Operating System

- It controls the allocation and use of the computing System's resources among the various user and tasks.
- It provides an interface between the computer hardware and the programmer that simplifies and makes it feasible for coding and debugging of application programs.

- **Bootstrap** program is loaded at power-up or reboot
- Typically stored in ROM or EEPROM, generally known as firmware
- Initializes all aspects of system

- Loads operating system kernel and starts execution

Storage Structure

Main memory – only large storage media that the CPU can access directly

• Random access

• Typically volatile

Secondary storage – extension of main memory that provides large nonvolatile storage capacity

Hard disks – rigid metal or glass platters covered with magnetic recording material

• Disk surface is logically divided into tracks, which are subdivided into sectors

• The disk controller determines the logical interaction between the device and the computer

Solid-State Drives – faster than hard disks, nonvolatile

• Various technologies

• Becoming more popular

Storage Hierarchy

Storage systems organized in hierarchy

• Speed

• Cost

• Volatility

Caching – copying information into faster storage system; main memory can be viewed as a cache for secondary storage

Device Driver for each device controller to manage I/O

• Provides uniform interface between controller and kernel

Operating-System Operations

Interrupt driven (hardware and software)

• Hardware interrupt by one of the devices

• Software interrupt (exception or trap):

Dual-Mode and Multimode Operation

• Distinguish between the execution of operating-system code and user defined code

• User mode (1) and kernel mode (0)

- Mode bit provided by hardware

Provides ability to distinguish when system is running user code or kernel code
Some instructions designated as privileged, only executable in kernel mode
Ex: instruction to switch to user mode is a privileged instruction; I/O control; timer mgmt

- Increasingly CPUs support multi-mode operations
- i.e. virtual machine manager (VMM) mode for guest VMs

Timer

- Timer to prevent infinite loop / process hogging resources
- Timer is set to interrupt the computer after some time period
- Keep a counter that is decremented by the physical clock.
- Operating system set the counter (privileged instruction)
- When counter zero generate an interrupt
- Set up before scheduling process to regain control or terminate program that exceeds allotted time

Process Management

- A process is a program in execution. It is a unit of work within the system. Program is a passive entity, process is an active entity.

Process Management Activities

- OS is responsible for the following activities:
- Creating and deleting both user and system processes
- Suspending and resuming processes
- Providing mechanisms for process synchronization
- Providing mechanisms for process communication
- Providing mechanisms for deadlock handling

OS services:

- User interface - Almost all operating systems have a user interface (UI). Varies between Command-Line (CLI), Graphics User Interface (GUI), Batch
- Program execution - The system must be able to load a program into memory and to run that program, end execution, either normally or abnormally (indicating error)
- I/O operations - A running program may require I/O, which may involve a file

or an I/O device

• File-system manipulation - Programs need to read and write files and directories, create and delete, search them, list file.
• Communications – Processes may exchange information, on the same computer or between computers over a network
Communications via shared memory or through message passing
• Error detection – OS needs to be constantly aware of possible errors
May occur in the CPU and memory hardware, in I/O devices, in user program
For each type of error, OS should take the appropriate action to ensure correct and consistent computing
Debugging facilities can greatly enhance the users and programmers abilities
• Ensuring the efficient operation of the system itself via resource sharing
• Resource allocation - When multiple users or multiple jobs running concurrently, resources must be allocated to each of them
Many types of resources - CPU cycles, main memory, file storage, I/O devices.
• Accounting - To keep track of which users use how much and what kinds of computer resources
• Protection and security - concurrent processes should not interfere with each other
Protection involves ensuring that all access to system resources is controlled
Security of the system from outsiders requires user authentication, extends to defending external I/O devices from invalid access attempts

Types of System Calls

Process Control
• end, abort
• load, execute
• create process, terminate process
• get/set process attributes
• wait for time
File Management
• create file, delete file
• open, close file
• read, write, reposition
• get and set file attributes
Device Management
• request device, release device

- read, write, reposition
- 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 and set process, file, or device attributes

Communications

- create, delete communication connection
- send, receive messages if message passing model to host name or process name

From client to server

- Shared-memory model create and gain access to memory regions
- transfer status information
- attach and detach remote devices
- Protection
- Control access to resources
- Get and set permissions
- Allow and deny user access

Operating System Structure

A common approach is to partition the task into small components,
or modules, rather than have one monolithic system

1) Simple Structure
2) Layered Approach
3) Microkernels
4) Modules
5) Hybrid Systems