Principle of operating systems

by

G. M. Al-Arafat. Shaown

An Operating System (OS) is a collection of software that manages computer hardware resources and provides common services for computer programs. When you start using a Computer System then it's the Operating System (OS) which acts as an interface between you and the computer hardware. The operating system is really a low level **Software** which is categorized as a **System Software** and supports a computer's basic functions, such as memory management, tasks scheduling and controlling peripherals etc.

This simple and easy tutorial will take you through step by step approach while learning Operating System concepts in detail.

What is Operating System?

An Operating System (OS) is an interface between a computer user and computer hardware. An operating system is a software which performs all the basic tasks like file management, memory management, process management, handling input and output, and controlling peripheral devices such as disk drives and printers.

Generally, a **Computer System** consists of the following components:

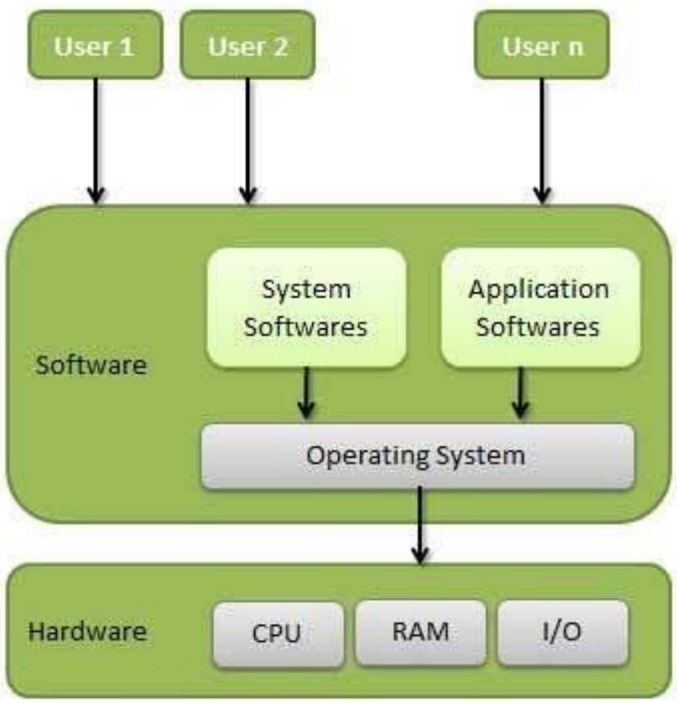
- **Computer Users** are the users who use the overall computer system.
- **Application Softwares** are the softwares which users use directly to perform different activities. These softwares are simple and easy to use like Browsers, Word, Excel, different Editors, Games etc. These are usually written in high-level languages, such as Python, Java and C++.
- **System Softwares** are the softwares which are more complex in nature and they are more near to computer hardware. These software are usually written in low-level languages like assembly language and includes **Operating Systems** (Microsoft Windows, macOS, and Linux), Compiler, and Assembler etc.
- **Computer Hardware** includes Monitor, Keyboard, CPU, Disks, Memory, etc.

So now let's put it in simple words:

If we consider a Computer Hardware is body of the Computer System, then we can say an Operating System is its soul which brings it alive ie. operational. We can never use a Computer System if it does not have an Operating System installed on it.

Architecture View:

We can draw a generic architecture diagram of an Operating System which is as follows:



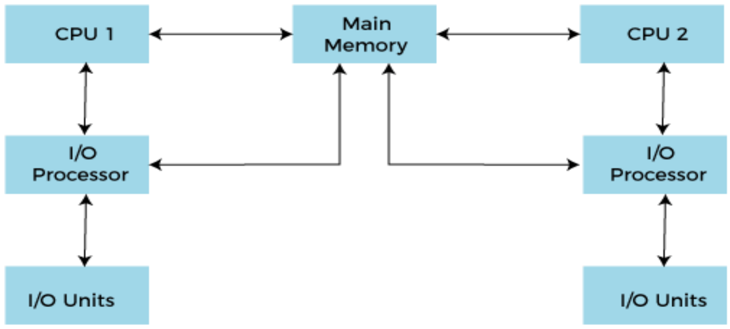
Multiprocessor Operating system

Multiprocessor system means, there are more than one processor which work parallel to perform the required operations.

It allows the multiple processors, and they are connected with physical memory, computer buses, clocks, and peripheral devices.

The main objective of using a multiprocessor operating system is to increase the execution speed of the system and consume high computing power.

For Example: UNIX Operating system is one of the most widely used multiprocessing systems.



Working of Multiprocessor System

Advantages

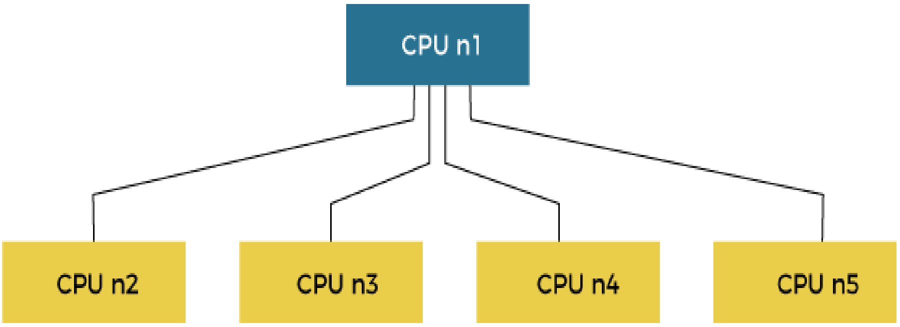
The advantages of multiprocessor systems are as follows −

- If there are multiple processors working at the same time, more processes can be executed parallel at the same time. Therefore the throughput of the system will increase.
- Multiprocessor systems are more reliable. Due to the fact that there are more than one processor, in case of failure of any one processor will not make the system come to a halt. Although the system will become slow if it happens but still it will work.
- Electricity consumption of a multiprocessor system is less than the single processor system. This is because, in single processor systems, many processes have to be executed by only one processor so there is a lot of load on it. But in case of multiple processor systems, there are many processors to execute the processes so the load on each processor will be comparatively less so electricity consumed will also be less.

Fields

The different fields of multiprocessor operating systems used are as follows −

- **Asymmetric Multiprocessor** − Every processor is given seeded tasks in this operating system, and the master processor has the power for running the entire system. In the course, it uses the master-slave relationship.



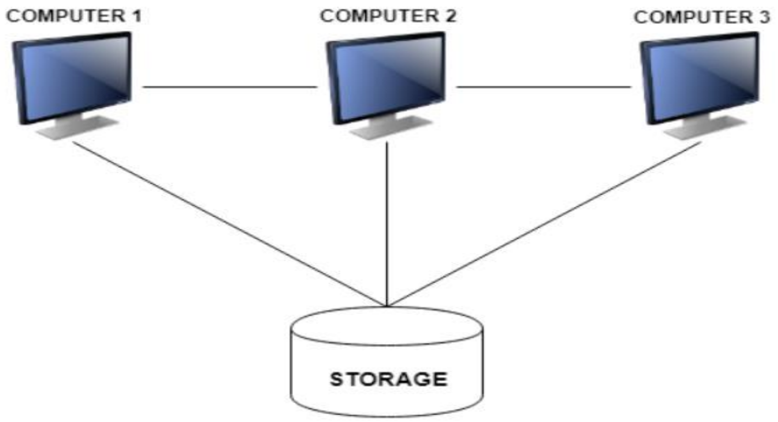
Asymmetric Multiprocessor System

- **Symmetric Multiprocessor** − In this system, every processor owns a similar copy of the OS, and they can make communication in between one another. All processors are connected with peering relationship nature, meaning it won’t be using master & slave relation.

Clustered systems:

Clustered systems are similar to parallel systems as they both have multiple CPUs. However a major difference is that clustered systems are created by two or more individual computer systems merged together. Basically, they have independent computer systems with a common storage and the systems work together.

A diagram to better illustrate this is –



Types of Cluster Systems:

Primarily, there are two types of Cluster Systems:

- **Asymmetric Cluster:** In this type of clustering, all the nodes run the required applications, and one node is in hot standby mode. The Hot standby node is used for monitoring the server till it fails, when it fails then it takes its place.
- **Symmetric Cluster:** In this type of clustering, all the nodes run applications and monitor other nodes at the same time. This clustering is more efficient than Asymmetric clustering as it doesn't have any hot standby key.

Advantages of Clustered Systems

The difference benefits of clustered systems are as follows −

Performance

Clustered systems result in high performance as they contain two or more individual computer systems merged together. These work as a parallel unit and result in much better performance for the system.

Fault Tolerance

Clustered systems are quite fault tolerant and the loss of one node does not result in the loss of the system. They may even contain one or more nodes in hot standby mode which allows them to take the place of failed nodes.

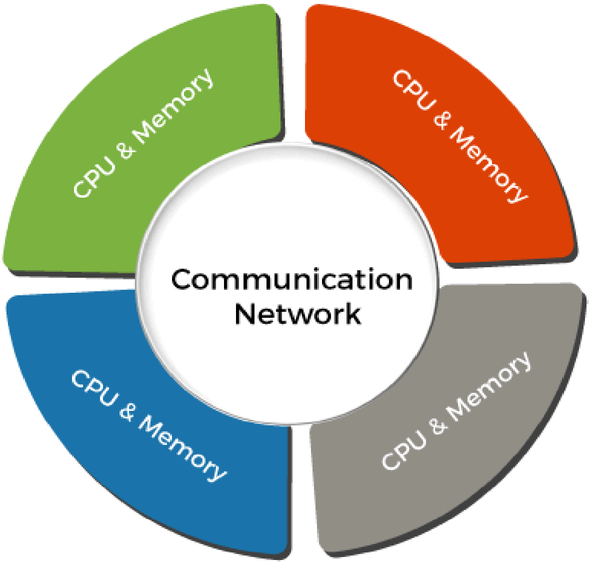
Scalability

Clustered systems are quite scalable as it is easy to add a new node to the system. There is no need to take the entire cluster down to add a new node.

Distributed Operating System

A distributed operating system (DOS) is an essential type of operating system. Distributed systems use many central processors to serve multiple real-time applications and users. As a result, data processing jobs are distributed between the processors.

It connects multiple computers via a single communication channel. Furthermore, each of these systems has its own processor and memory. Additionally, these CPUs communicate via high-speed buses or telephone lines. Individual systems that communicate via a single channel are regarded as a single entity. They're also known as loosely coupled systems.



Types of Distributed Operating System:

- Client-Server Systems
- Peer-to-Peer Systems

Client-Server System:

This type of system requires the client to request a resource, after which the server gives the requested resource. When a client connects to a server, the server may serve multiple clients at the same time. Client-Server Systems are also referred to as "Tightly Coupled Operating Systems". This system is primarily intended for multiprocessors and homogenous multicomputer. Client-Server Systems function as a centralized server since they approve all requests issued by client systems.

Peer-to-Peer System

The nodes play an important role in this system. The task is evenly distributed among the nodes. Additionally, these nodes can share data and resources as needed. Once again, they require a network to connect.

The Peer-to-Peer System is known as a "Loosely Couple System". This concept is used in computer network applications since they contain a large number of processors that do not share memory or clocks. Each processor has its own local memory, and they interact with one another via a variety of communication methods like telephone lines or high-speed buses.

Handheld Operating System:

Handheld operating systems are present in all handheld devices like Smartphones and tablets. It is also called a Personal Digital Assistant. The popular handheld device in today's market is android and iOS. These operating systems need a high processing processor and also embedded with different types of sensor.

Features of Handheld Operating System:

- Its work is to provide real-time operations.
- There is direct usage of interrupts.
- Input/Output device flexibility.
- Configurability.

Types of Handheld Operating Systems:

Types of Handheld Operating Systems are as follows:

- Palm OS
- Symbian OS
- Linux OS
- Windows
- Android

Advantages of Handheld Operating System:

Some advantages of a Handheld Operating System are as follows:

- Less Cost.
- Less weight and size.
- Less heat generation.
- More reliability.

Disadvantages of Handheld Operating System:

Some disadvantages of Handheld Operating Systems are as follows:

- Less Speed.
- Small Size.
- Input / Output System (memory issue or less memory is available).

Real-time systems:

Real-time operating systems (RTOS) are used in environments where a large number of events, mostly external to the computer system, must be accepted and processed in a short time or within certain deadlines. such applications are industrial control, telephone switching equipment, flight control, and real-time simulations. With an RTOS, the processing time is measured in tenths of seconds. This system is time-bound and has a fixed deadline. The processing in this type of system must occur within the specified constraints. Otherwise, This will lead to system failure.

Examples of the real-time operating systems: Airline traffic control systems, Command Control Systems, Airlines reservation system, Heart Pacemaker, Network Multimedia Systems, Robot etc.

There are two types of real-time operating systems:

1. Hard Real-Time operating system:

These operating systems guarantee that critical tasks be completed within a range of time.

For example, a robot is hired to weld a car body. If the robot welds too early or too late, the car cannot be sold, so it is a hard real-time system that requires complete car welding by robot hardly on the time.

2. Soft real-time operating system:

This operating system provides some relaxation in the time limit.

For example – Multimedia systems, digital audio systems etc. Explicit, programmer-defined and controlled processes are encountered in real-time systems. A separate process is changed with handling a single external event. The process is activated upon occurrence of the related event signaled by an interrupt.

Advantages of Real-time operating system:

The benefits of real-time operating system are as follows-:

- Easy to layout, develop and execute real-time applications under the real-time operating system.
- The real-time working structures are extra compact, so those structures require much less memory space.
- In a Real-time operating system, the maximum utilization of devices and systems.
- Focus on running applications and less importance to applications that are in the queue.
- Since the size of programs is small, RTOS can also be embedded systems like in transport and others.
- These types of systems are error-free.
- Memory allocation is best managed in these types of systems.
  Disadvantages of Real-time operating system:

The disadvantages of real-time operating systems are as follows-

- Real-time operating systems have complicated layout principles and are very costly to develop.
- Real-time operating systems are very complex and can consume critical CPU cycles.

Services of Operating System

- Program Execution
- Input Output Operations
- File Management
- Error Handling
- Resource Management
- Communication between Processes
- Security and Protection

Program Execution:

It is the Operating System that manages how a program is going to be executed. It loads the program into the memory after which it is executed. The order in which they are executed depends on the CPU Scheduling Algorithms. A few are FCFS, SJF, etc. When the program is in execution, the Operating System also handles deadlock i.e. no two processes come for execution at the same time.

Input Output Operations:

Operating System manages the input-output operations and establishes communication between the user and device drivers. Device drivers are software that is associated with hardware that is being managed by the OS so that the sync between the devices works properly. It also provides access to input-output devices to a program when needed.

File Management:

The operating system helps in managing files also. If a program needs access to a file, it is the operating system that grants access. These permissions include read-only, read-write, etc. It also provides a platform for the user to create, and delete files. The Operating System is responsible for making decisions regarding the storage of all types of data or files, i.e, floppy disk/hard disk/pen drive, etc. The Operating System decides how the data should be manipulated and stored.

Error handling:

Errors can occur anytime and anywhere. An error may occur in CPU, in I/O devices or in the memory hardware. Following are the major activities of an operating system with respect to error handling −

- The OS constantly checks for possible errors.
- The OS takes an appropriate action to ensure correct and consistent computing.

Resource Management:

In case of multi-user or multi-tasking environment, resources such as main memory, CPU cycles and files storage are to be allocated to each user or job. Following are the major activities of an operating system with respect to resource management −

- The OS manages all kinds of resources using schedulers.
- CPU scheduling algorithms are used for better utilization of CPU.

Communication between Processes:

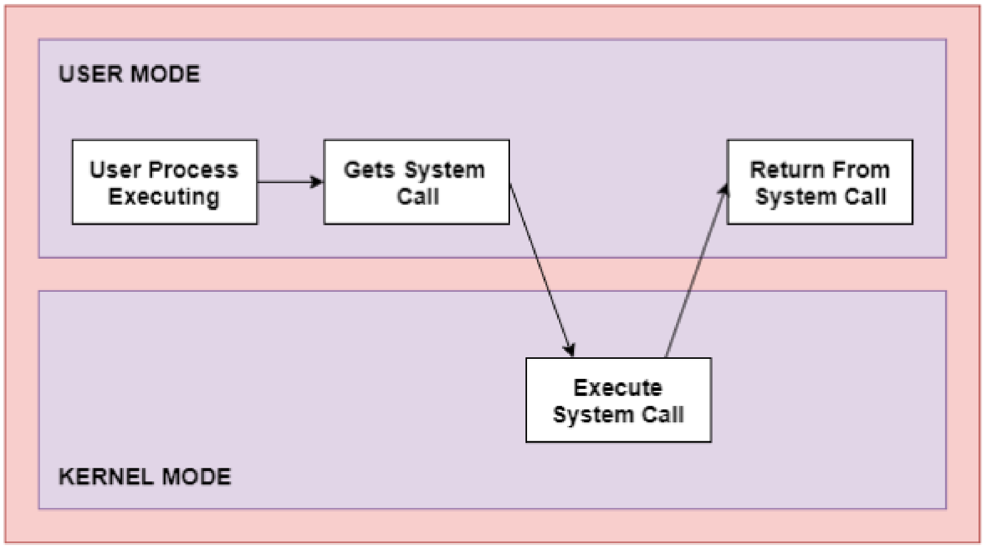
The Operating system manages the communication between processes. Communication between processes includes data transfer among them. If the processes are not on the same computer but connected through a computer network, then also their communication is managed by the Operating System itself.

Security and Protection:

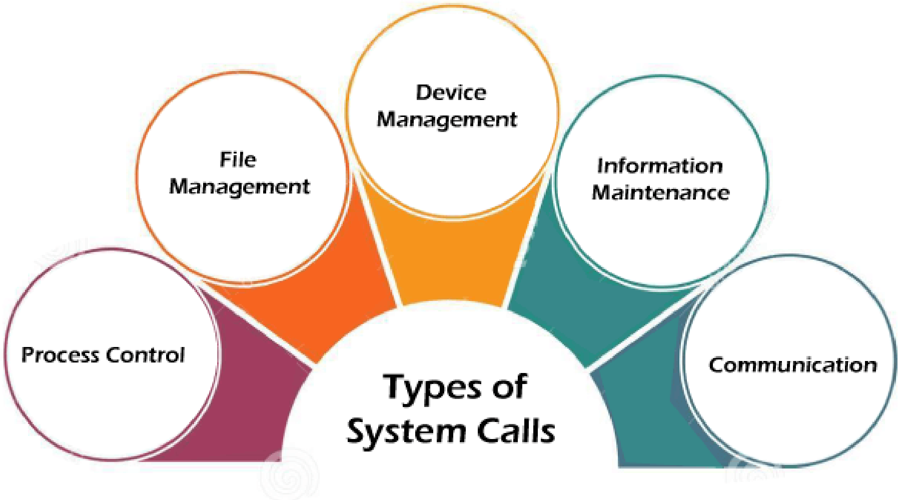
Operating device affords safety to the statistics and packages of a person and protects any interference from unauthorized users. The safety feature counters threats, which are published via way of individuals out of doors the manage of the running device.

System calls:

A system call is a way for programs to interact with the operating system. A computer program makes a system call when it makes a request to the operating system's kernel. System call provides the services of the operating system to the user programs via Application Program Interface (API).



Types of System Calls:



- Process Control
- File Management
- Device Management
- Information Maintenance
- Communication

Process Control:

Process control is the system call that is used to direct the processes. Some process control examples include creating, load, abort, end, execute, process, terminate the process, etc.

File Management:

File management is a system call that is used to handle the files. Some file management examples include creating files, delete files, open, close, read, write, etc.

Device Management:

These system calls are responsible for device manipulation such as reading from device buffers, writing into device buffers etc.

Information Maintenance:

These system calls handle information and its transfer between the operating system and the user program.

Communication:

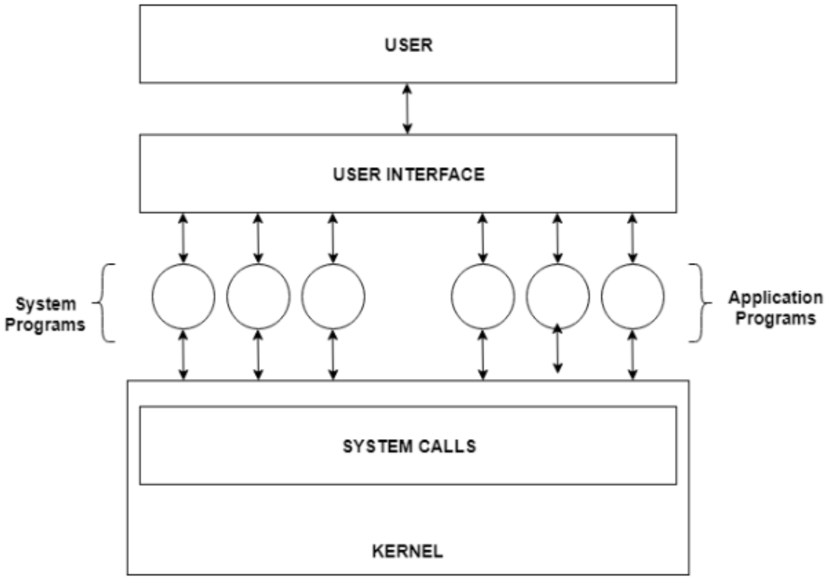
Communication is a system call that is used for communication. There are some examples of communication, including create, delete communication connections, send, receive messages, etc.

System programs:

System programs provide an environment where programs can be developed and executed. In the simplest sense, system programs also provide a bridge between the user interface and system calls. In reality, they are much more complex. For example, a compiler is a complex system program.

System Programs Purpose:

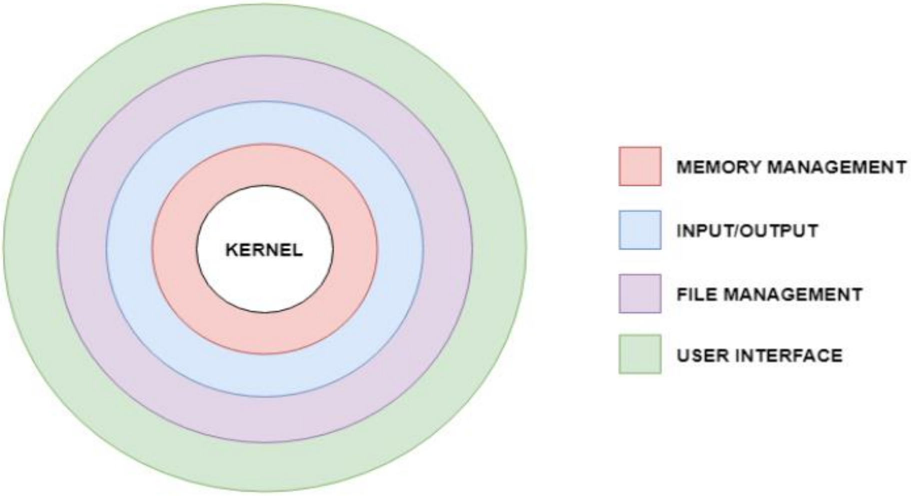
The system program serves as a part of the operating system. It traditionally lies between the user interface and the system calls. The user view of the system is actually defined by system programs and not system calls because that is what they interact with and system programs are closer to the user interface.



Operating System Design and Implementation:

An operating system is a construct that allows the user application programs to interact with the system hardware. Operating system by itself does not provide any function but it provides an atmosphere in which different applications and programs can do useful work.

There are many problems that can occur while designing and implementing an operating system. These are covered in operating system design and implementation.



Layered Operating System Design

Operating System Design Goals:

It is quite complicated to define all the goals and specifications of the operating system while designing it. The design changes depending on the type of the operating system i.e if it is batch system, time shared system, single user system, multi user system, distributed system etc.

There are basically two types of goals while designing an operating system. These are −

User Goals:
The operating system should be convenient, easy to use, reliable, safe and fast according to the users. However, these specifications are not very useful as there is no set method to achieve these goals.

System Goals:
The operating system should be easy to design, implement and maintain. These are specifications required by those who create, maintain and operate the operating system. But there is not specific method to achieve these goals as well.

Operating System Implementation:

The operating system needs to be implemented after it is designed. Earlier they were written in assembly language but now higher level languages are used. The first system not written in assembly language was the Master Control Program (MCP) for Burroughs Computers.

Advantages of Higher Level Language:

There are multiple advantages to implementing an operating system using a higher level language such as: the code is written more fast, it is compact and also easier to debug and understand. Also, the operating system can be easily moved from one hardware to another if it is written in a high level language.

Disadvantages of Higher Level Language:

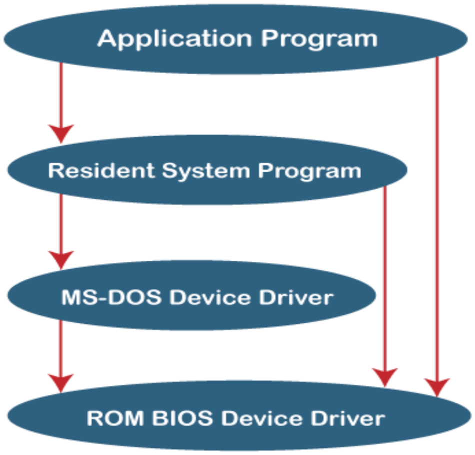
Using high level language for implementing an operating system leads to a loss in speed and increase in storage requirements. However in modern systems only a small amount of code is needed for high performance, such as the CPU scheduler and memory manager. Also, the bottleneck routines in the system can be replaced by assembly language equivalents if required.

Operating system structure:

The operating system may be implemented with the assistance of several structures. The structure of the operating system is mostly determined by how the many common components of the OS are integrated and merged into the kernel.

Simple Structure:

Such OS's are small, simple, and limited, with no well-defined structure. There is a lack of separation between the interfaces and levels of functionality. The MS-DOS is the best example of such an operating system. Application programs in MS-DOS can access basic I/O functions. If one of the user programs fails on these OSs, the complete system crashes. Below is the diagram of the MS-DOS structure that may help you understand the simple structure.

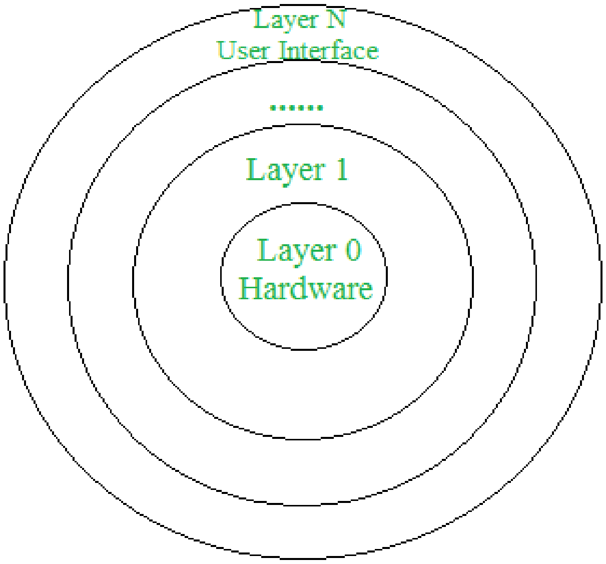


MS-DOS Structure

Layered structure:

An OS can be broken into pieces and retain much more control on system. In this structure the OS is broken into number of layers (levels). The bottom layer (layer 0) is the hardware and the topmost layer (layer N) is the user interface. These layers are so designed that each layer uses the functions of the lower level layers only. This simplifies the debugging process as if lower level layers are debugged and an error occurs during debugging then the error must be on that layer only as the lower level layers have already been debugged.

UNIX is an example of this structure.



Virtual Machine:

Virtual Machine abstracts the hardware of our personal computer such as CPU, disk drives, memory, NIC (Network Interface Card) etc, into many different execution environments as per our requirements, hence giving us a feel that each execution environment is a single computer.

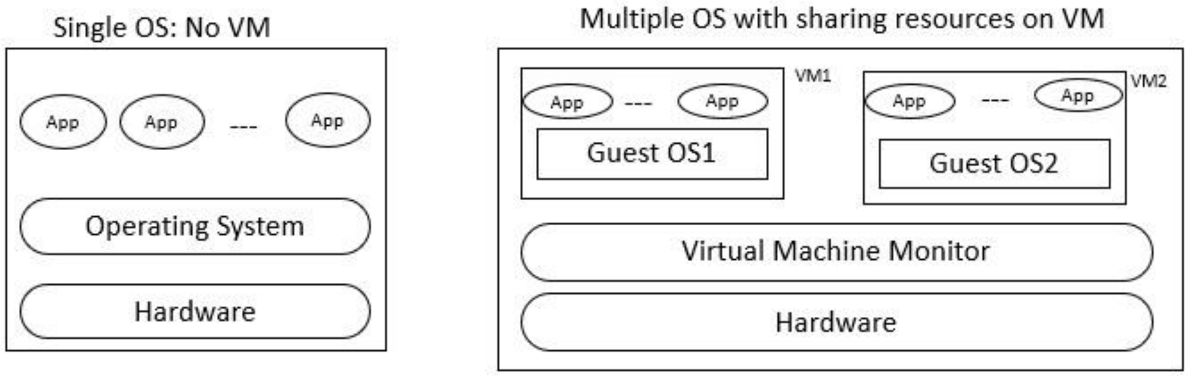
For example, VirtualBox.

<u>Characteristics of virtual machines</u>:

The characteristics of the virtual machines are as follows −

- Multiple OS systems use the same hardware and partition resources between virtual computers.
- Separate Security and configuration identity.
- Ability to move the virtual computers between the physical host computers as holistically integrated files.

The below diagram shows you the difference between the single OS with no VM and Multiple OS with VM −



Advantages:

- There are no protection problems because each virtual machine is completely isolated from all other virtual machines.
- Virtual machine can provide an instruction set architecture that differs from real computers.
- Easy maintenance, availability and convenient recovery.

Disadvantages:

- When multiple virtual machines are simultaneously running on a host computer, one virtual machine can be affected by other running virtual machines, depending on the workload.
- Virtual machines are not as efficient as a real one when accessing the hardware.