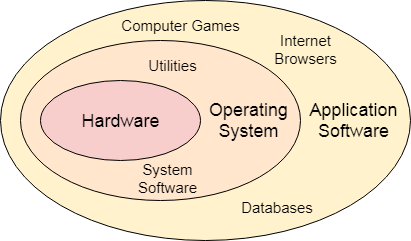
**UNIT-I**

**CHAPTER –2**

What Operating Systems Do,Operating-System Functions,Operating-System Operations, Resource Management, Security and Protection, Virtualization, Distributed Systems, Kernel Data Structures.

**What Operating Systems Do or Role of the Operating Systems**

* **Definition: An Operating System (OS) is an interface between computer user and computer hardware.**
* An operating system is 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.
* An operating system is a program that acts as an interface between the user and the computer hardware and controls the execution of all kinds of programs.
* Some popular Operating Systems include Linux, Windows, OS X, VMS, OS/400, AIX, z/OS, etc.

 or 

**Fig: Operating System**

**The goals of an Operating system are:**

* Execute user programs and make solving user problems **easier**.
* Make the computer system **convenient** to use.
* Use the computer hardware in an **efficient** manner.

**Functions of an operating System**

**Following are some of important functions of an operating System:**

* Memory Management
* Processor Management
* Device Management
* File Management
* Security
* Control over system performance
* Job accounting
* Error detecting aids
* Coordination between other software and users

**Memory Management:**

Memory management refers to management of Primary Memory or Main Memory. Main memory is a large array of words or bytes where each word or byte has its own address.

Main memory provides a fast storage that can be accessed directly by the CPU. For a program to be executed, it must in the main memory**.**

**An Operating System does the following activities for memory management −**

* Keeps tracks of primary memory, i.e., what part of it are in use by whom, what part are not in use.
* In multiprogramming, the OS decides which process will get memory when and how much.
* Allocates the memory when a process requests it to do so.
* De-allocates the memory when a process no longer needs it or has been terminated.

**Processor Management**

In multiprogramming environment, the OS decides which process gets the processor when and for how much time. This function is called **process scheduling**.

**An Operating System does the following activities for processor management −**

* Keeps tracks of processor and status of process. The program responsible for this task is known as **traffic controller**.
* Allocates the processor (CPU) to a process.
* De-allocates processor when a process is no longer required.

**Device Management**

An Operating System manages device communication via their respective drivers.

**It does the following activities for device management −**

* Keeps tracks of all devices. Program responsible for this task is known as the **I/O controller**.
* Decides which process gets the device when and for how much time.
* Allocates the device in the efficient way.
* De-allocates devices.

**File Management**

A file system is normally organized into directories for easy navigation and usage. These directories may contain files and other directions.

**An Operating System does the following activities for file management −**

* It keeps track of where information is stored, user access settings and status of every file, and more… These facilities are collectively known as the file system.

**Other Important Activities**

Following are some of the important activities that an Operating System performs −

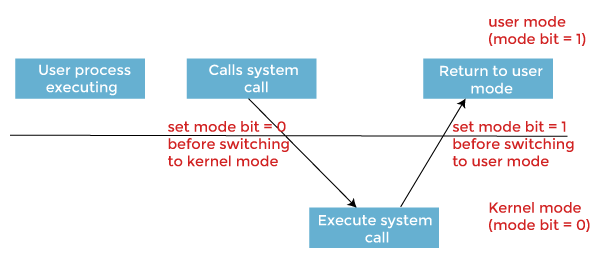
* **Security** − By means of password and similar other techniques, it prevents unauthorized access to programs and data.
* **Control over system performance** − Recording delays between request for a service and response from the system.
* **Job accounting** − Keeping track of time and resources used by various jobs and users.
* **Error detecting aids** − Production of dumps, traces, error messages, and other debugging and error detecting aids.
* **Coordination between other software’s and users** − Coordination and assignment of compilers, interpreters, assemblers and other software to the various users of the computer systems.

**Operations of an operating System**

* An error in one program can adversely affect many processes, it might modify data of another program, or also can affect the operating system. For example, if a process stuck in the infinite loop then this infinite loop could affect the correct operation of other processes.
* **Suppose,** when the computer system is run by user applications like creating a text document or using any application program, then the system is in user mode. When the user application requests for a service from the operating system or an interrupt occurs or [system call](https://www.geeksforgeeks.org/operating-system-introduction-system-call/), then there will be a transition from user to kernel mode to fulfil the requests.
* In order to ensure the proper execution of the operating system, we must be able to distinguish between the execution of operating-system code and user defined code. The approach taken by most computer systems is to provide hardware support that allows us to differentiate among various modes of execution. At the very least, we need two separate modes of operation: **user mode** and **kernel mode** (also called **supervisor mode**, **system mode**, or **privileged mode**).
* In monitor mode, the CPU can use all instructions and access all areas of memory.
* In user mode, the CPU is restricted to unprivileged instructions and a specified area of memory.

**Example**

* With the mode bit, we can distinguish between a task executed on behalf of the operating system and one executed on behalf of the user.



* When the computer system executes on behalf of a user application, the system is in ***user mode***.
* However, when a user application requests a service from the operating system via a system call, it must transition from ***user*** to ***kernel mode*** to fulfill the request. As we can say, this architectural enhancement is useful for many other aspects of system operation.
* At system boot time, the hardware starts in ***kernel mode***.
* The operating system is then loaded and starts user applications in ***user mode***.
* Whenever a trap or interrupt occurs, the hardware switches from ***user mode*** to ***kernel mode***, changing the mode bit's state to 0.
* Thus, whenever the operating system gains control of the computer, it is in ***kernel mode***.
* The system always ***switches to user mode***by setting the mode bit to 1 before passing control to a user program.

**Kernel Data Structures**

* ***Kernel is a computer program that is a core or heart of an operating system.***
* Kernel is the core part of an OS (Operating system); hence it has full control over everything in the system. Each operation of hardware and software is managed and administrated by the kernel.
* It acts as a bridge between applications and data processing done at the hardware level. It is the central component of an OS.
* It is the part of the OS that always resides in computer memory and enables the communication between software and hardware components.

We briefly describe several fundamental data structures used extensively in operating systems.

* The following are Different types of Kernel data structures are:
* **Array:** An array is a simple data structure in which each element can be accessed directly.
* **In a singly linked list**, each item points to its successor.
* **In a doubly linked list**, a given item can refer either to its predecessor or to its successor.
* **In a circularly linked list**, the last element in the list refers to the first element, rather than to null.
* **A stack** is a sequentially ordered data structure that uses the last in, first out (LIFO) principle for adding and removing items, meaning that the last item placed onto a stack is the first item removed. The operations for inserting and removing items from a stack are known as push and pop, respectively.
* **A queue**, in contrast, is a sequentially ordered data structure that uses thefirst in, first out (FIFO) principle: items are removed from a queue in the orderin which they were inserted.
* **A tree** is a data structure that can be used to represent data hierarchically. Datavalues in a tree structure are linked through parent–child relationships. In ageneral tree, a parent may have an unlimited number of children.
* **In a binary tree**, a parent may have at most two children, which we term the left childand the right child.
* **A binary search tree** additionally requires an orderingbetween the parent’s two children in which le f t child <= right child.
* A **hash function**takes data as its input, performs a numeric operation on this data, and returns a numeric value. This numeric value can then be used as an index into a table (typically an array) to quickly retrieve the data. Whereas searching for a data item through a list of size *n* can require up to *O*(*n*) comparisons in the worst case, using a hash function for retrieving data from table can be as good as *O*(1) in the worst case, depending on implementation details.

**Resource Management**

* **The operating system acts as a manager of the all resources and allocates them to specific programs and users, whenever necessary to perform a particular task. Therefore the operating system is the resource manager i.e. it can manage the resource of a computer system internally.**
* **Resource management is the dynamic allocation and de-allocation by an operating system of processor cores, memory pages, and various types of bandwidth to computations that compete for those resources. The objective is to allocate resources so as to optimize responsiveness subject to the finite resources available.**

**Security and Protection**

* **Protection is a method used in operating systems that manages threats within the system to maintain the proper functioning of the system.**
* **Security is a method used in operating systems that handles the threats from outside of the system to maintain the proper functioning of the system.**
* Protection and security requires that computer resources such as CPU, software’s, memory etc. are protected. This extends to the operating system as well as the data in the system. This can be done by ensuring integrity, confidentiality and availability in the operating system. The system must be protecting against unauthorized access, viruses, worms etc.

**Threats to Protection and Security:**

A threat is a program that is malicious in nature and leads to harmful effects for the system. Some of the common threats that occur in a system are −

**Virus**

Viruses are generally small snippets of code embedded in a system. They are very dangerous and can corrupt files, destroy data, crash systems etc. They can also spread further by replicating themselves as required.

**Trojan Horse**

A trojan horse can secretly access the login details of a system. Then a malicious user can use these to enter the system as a harmless being and wreak havoc.

**Trap Door**

A trap door is a security breach that may be present in a system without the knowledge of the users. It can be exploited to harm the data or files in a system by malicious people.

**Worm**

A worm can destroy a system by using its resources to extreme levels. It can generate multiple copies which claim all the resources and don't allow any other processes to access them. A worm can shut down a whole network in this way.

**Denial of Service**

These type of attacks do not allow the legitimate users to access a system. It overwhelms the system with requests so it is overwhelmed and cannot work properly for other user.

**Protection and Security Methods:**

The different methods that may provide protect and security for different computer systems are −

**Authentication**

This deals with identifying each user in the system and making sure they are who they claim to be. The operating system makes sure that all the users are authenticated before they access the system. The different ways to make sure that the users are authentic are:

* **Username/ Password**

Each user has a distinct username and password combination and they need to enter it correctly before they can access the system.

* **User Key/ User Card**

The users need to punch a card into the card slot or use they individual key on a keypad to access the system.

* **User Attribute Identification**

Different user attribute identifications that can be used are fingerprint, eye retina etc. These are unique for each user and are compared with the existing samples in the database. The user can only access the system if there is a match.

**One Time Password**

These passwords provide a lot of security for authentication purposes. A onetime password can be generated exclusively for a login every time a user wants to enter the system. It cannot be used more than once. The various ways a onetime password can be implemented are −

* **Random Numbers**

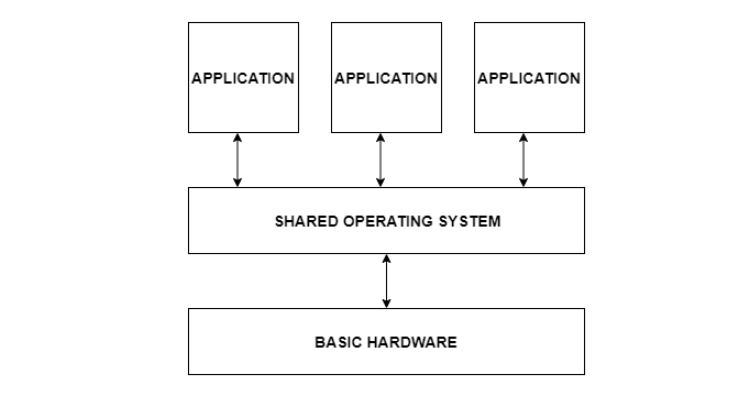
The system can ask for numbers that correspond to alphabets that are pre arranged. This combination can be changed each time a login is required.

* **Secret Key**

A hardware device can create a secret key related to the user id for login. This key can change each time.

**Virtualization**

* **Virtualization in operating system changes a normal operating system so that it can run different types of applications that may be handled on a single computer system by many users.**
* **The operating system may appear different to each user and each of them may believe they are interacting with the only operating system i.e. this does not interfere with user experience.**
* Operating system virtualization can also be used to migrate a process from one instance of the operating system to another. However, all the processes in the system are isolated and there operations are strictly monitored so there are no discrepancies in the system.
* A diagram representing operating system virtualization is as follows −



**Advantages of Virtualization:**

Some of the advantages of virtualization are −

* Virtualization allows a finite number of hardware resources to be easily distributed to multiple processes that require them.
* Improved security can be obtained by using virtualisation. This happens because each process inhabits its own instance of the operating system and works independently.
* Operating system virtualization is very useful for establishing a virtual hosting environment.
* There is only a little overhead involved for operating system virtualization and so it is very beneficial.

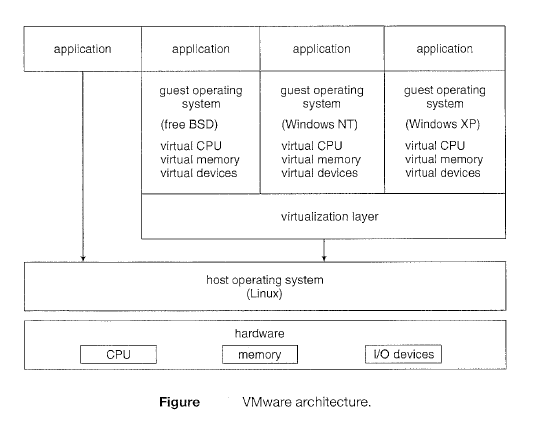
**Disadvantages of Virtualization:**

Some of the disadvantages of virtualization are −

* Specialized experts are required to implement and manage a virtualized system. This results in need for virtualization experts and increased costs.
* There are many upfront costs involved in virtualization. These include the cost for virtualization software as well as the additional hardware required. The costs also depends on the existing system network.

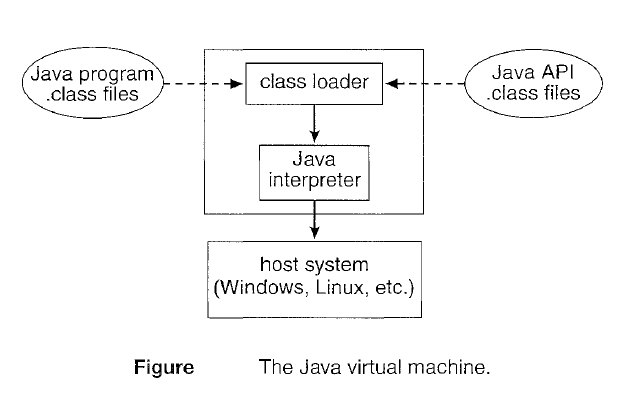
**Example 1 : VMware Workstation**

* It is a popular commercial application that abstracts Intel X86 and compatible hardware into isolated virtual machines.
* VMware Workstation runs as an application on a host operating system such as Windows or Linux and allows this host system to concurrently run several different guest operating systems as independent virtual machines.
* The architecture of such a system is shown in Figure.
* In this scenario, Linux is running as the host operating system; and FreeBSD, Windows NT, and Windows XP are running as guest operating systems. The virtualization layer is the heart of VMware, as it abstracts the physical hardware into isolated virtual machines running as guest operating systems. Each virtual machine has its own virtual CPU, memory, disk drives, network interfaces, and so forth.

****

**Example 2 : The Java Virtual Machine**

* Java is a popular object-oriented programming language introduced by Sun
* Microsystems in 1995. In addition to a language specification and a large API library, Java also provides a specification for a Java virtual machine-or JVM.
* Java objects are specified with the class construct; a Java program consists of one or more classes. For each Java class, the compiler produces an architecture-neutral byte code output (.class) file that will run on any implementation of the JVM.
* The JVM is a specification for an abstract computer. It consists of a class loader and a Java interpreter that executes the architecture-neutral byte codes, as diagrammed in Figure.
* The class loader loads the compiled . class files from both the Java program and the Java API for execution by the Java interpreter. After a class is loaded, the verifier checks that the . class file is valid Java bytecode and does not overflow or underflow the stack It also ensures that the bytecode does not perform pointer arithmetic, which could provide illegal memory access. If the class passes verification, it is run by the Java interpreter.

****

**Distributed Systems**

* **Distributed Operating System** is one of the important type of operating system. Multiple central processors are used by Distributed systems to serve multiple real-time applications and multiple users.
* Accordingly, Data processing jobs are distributed among the processors.
* Processors communicate with each other through various communication lines (like high-speed buses or telephone lines).
* These are known as **loosely coupled systems** or distributed systems. Processors in this system may vary in size and function. They are referred as sites, nodes, computers, and so on.

**Advantages**

* With resource sharing facility, a user at one site may be able to use the resources available at another.
* Speedup the exchange of data with one another via electronic mail.
* Failure of one site in a distributed system doesn’t affect the others, the remaining sites can potentially continue operating.
* Better service to the customers.
* Reduction of the load on the host computer.
* Reduction of delays in data processing.