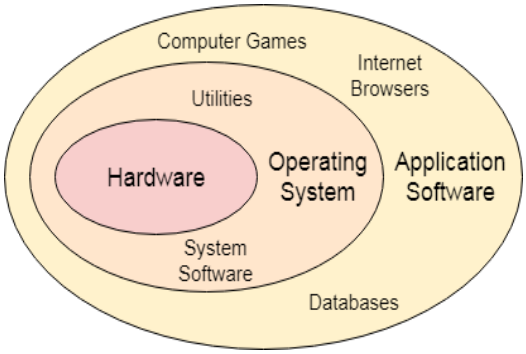
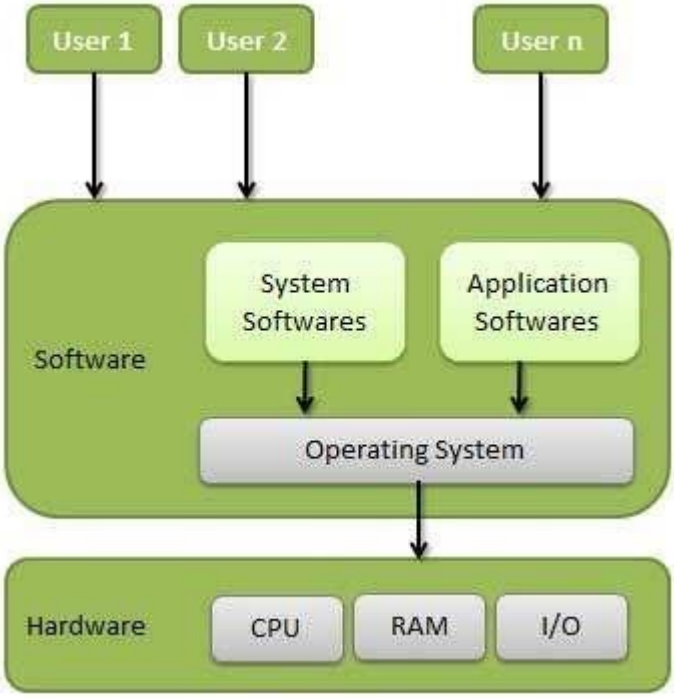
Operating System (OS) definition:

- An **Operating System** (OS) is a program that acts as an interface between a computer user and computer hardware. An operating system (OS) is system software that manages computer hardware, software resources,and provides common services for computer programs.
- 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.
- Some popular Operating Systems include Linux Operating System, Windows Operating System, VMS, OS/400, AIX, z/OS, etc. Today, Operating systems is found almost in every device like mobilephones, personal computers, mainframe computers, automobiles, TV, Toys etc.



Generic architecture diagram of an Operating System which is as follows:



Operating System Generations

Operating systems have been evolving over the years. We can categories this evaluation based on different generations which is briefed below:

0th Generation

The term 0th generation is used to refer to the period of development of computing when Charles Babbage invented the Analytical Engine and later John Atanasoff created a computer in 1940. The hardware component technology of this period was electronic vacuum tubes. There was no Operating System available for this generation computer and computer programs were written in machine language. This computers in this generation were inefficient and dependent on the varying competencies of the individual programmer as operators.

First Generation (1951-1956)

The first generation marked the beginning of commercial computing including the introduction of Eckert and Mauchly's **UNIVAC** I in early 1951, and a bit later, the IBM 701.

System operation was performed with the help of expert operators and without the benefit of an operating system for a time though programs began to be written in higher level, procedure-orientedlanguages, and thus the operator's routine expanded. Later mono-programmed operating system was developed, which eliminated some of the human intervention in running job and provided programmers with a number of desirable functions. These systems still continued to operate under the control of a human operator who used to follow a number of steps to execute a program. Programming language like **FORTRAN** was developed by John W. Backus in 1956.

Second Generation (1956-1964)

The second generation of computer hardware was most notably characterized by transistors replacing vacuum tubes as the hardware component technology. The first operating system **GMOS** was developed by the IBM computer. GMOS was based on single stream batch processing system, because it collects all similar jobs in groups or batches and then submits the jobs to the operating system using a punch card to complete all jobs in a machine. Operating system is cleaned after completing one job and then continues to read and initiates the next job in punch card.

Researchers began to experiment with multiprogramming and multiprocessing in their computing services called the time-sharing system. A noteworthy example is the Compatible **Time-Sharing System (CTSS),** developed at MIT during the early 1960s.

Third Generation (1964-1979)

The third generation officially began in April 1964 with IBM's announcement of its System/360 familyof computers. Hardware technology began to use **integrated circuits (ICs**) which yielded significant advantages in both speed and economy.

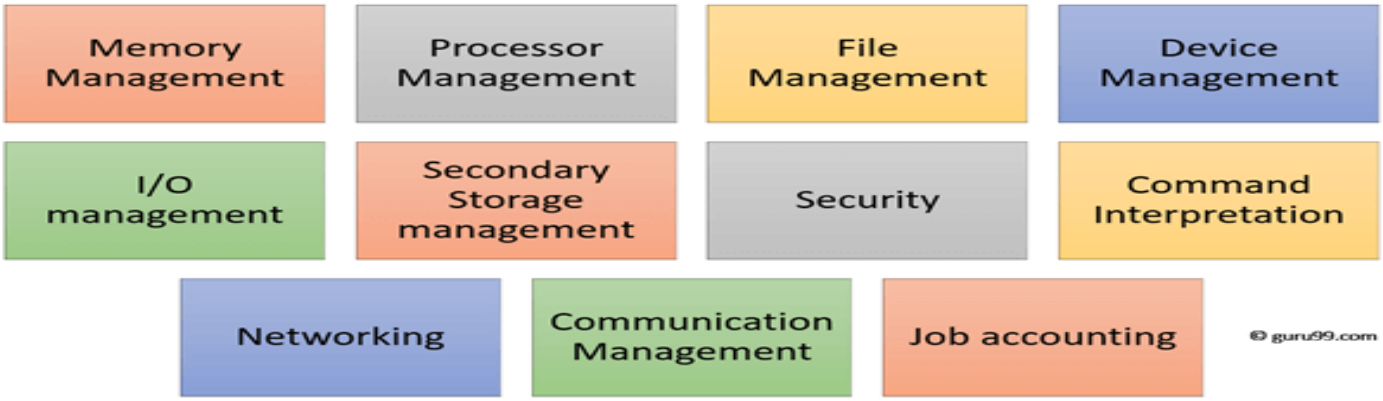
Operating system development continued with the introduction and widespread adoption of multiprogramming. The idea of taking fuller advantage of the computer's data channel I/O capabilities continued to develop.

Another progress which leads to developing of personal computers in fourth generation is a new development of minicomputers with DEC PDP-1.

Fourth Generation (1979 – Present)

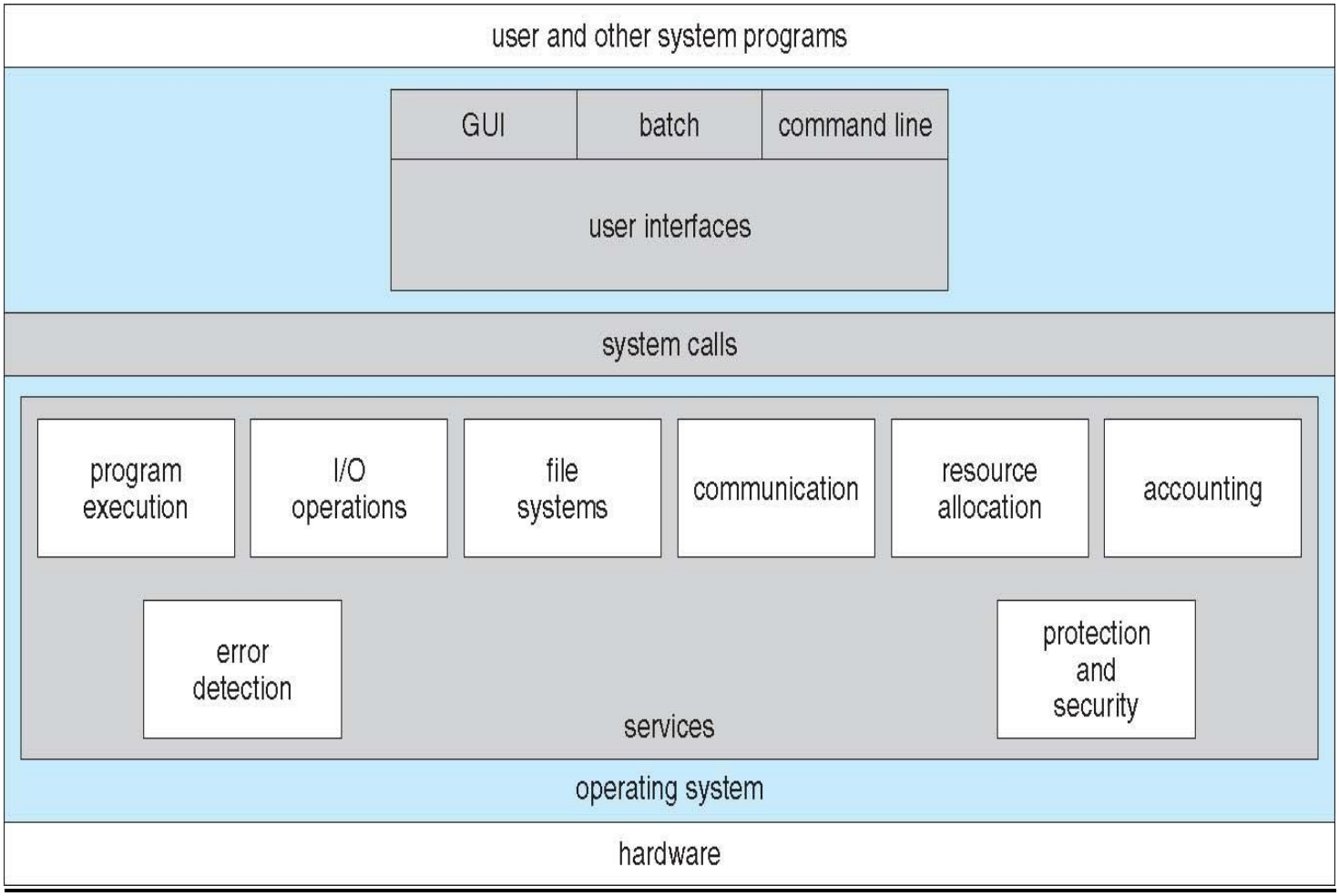
The fourth generation is characterised by the appearance of the personal computer and the workstation. The component technology of the third generation, was replaced by very large-scale integration (VLSI). Many Operating Systems which we are using today like Windows, Linux, MacOS etc developed in the fourth generation.

Operating System: some of important functions/Services/Responsibilities :



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

A view of Operating System services



Various types of operating systems

An **Operating System (OS)** is a software that acts as an interface between computer hardware components and the user. Every computer system must have at least one operating system to run other programs. Applications like Browsers, MS Office, Notepad Games, etc., need some environment to run and perform its tasks.
Types of Operating System (OS) :
Following are the popular types of OS (Operating System):

- Batch Operating System
- Multitasking/Time Sharing OS
- Multiprocessing OS
- Real Time OS
- Distributed OS
- Network OS
- Mobile OS

1.Batch Operating System

Some computer processes are very lengthy and time-consuming. To speed the same process, a job with a similar type of needs are batched together and run as a group. The user of a batch operating system never directly interacts with the computer. In this type of OS, every user prepares his or her job on an offline device like a punch card and submit it to the computer operator.

Advantages of Batch OS

- The use of a resident monitor improves computer efficiency as it eliminates CPU time between two jobs.

Disadvantages of Batch OS

1. Starvation: Batch processing suffers from starvation.

2.Multi-Tasking/Time-sharing Operating systems

Time-sharing operating system enables people located at a different terminal(shell) to use a single computer system at the same time. The processor time (CPU) which is shared among multiple users is termed as time sharing.

Advantages of Time-Sharing Operating System

- The time-sharing operating system provides effective utilization and sharing of resources.
- This system reduces CPU idle and response time.

Disadvantages of Time-Sharing Operating System

- Data transmission rates are very high in comparison to other methods.
- Security and integrity of user programs loaded in memory and data need to be maintained as many users access the system at the same time.

3.Real time OS

A real time operating system time interval to process and respond to inputs is very small. Examples: Military Software Systems, Space Software Systems are the Real time OS example.

Advantages of Real-time operating system:

- Easy to layout, develop and execute real-time applications under the real-time operating system.
- In a Real-time operating system, the maximum utilization of devices and systems.

Disadvantages of Real-time operating system:

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

4.Distributed Operating System

Distributed systems use many processors located in different machines to provide very fast computation to its users.

Advantages of Distributed Operating System

- The distributed operating system provides sharing of resources.
- This type of system is fault-tolerant.

Disadvantages of Distributed Operating System

- Protocol overhead can dominate computation cost.

5.Network Operating System

Network Operating System runs on a server. It provides the capability to serve to manage data, user, groups, security, application, and other networking functions.

Advantages of Network Operating System

- In this type of operating system, network traffic reduces due to the division between clients and the server.
- This type of system is less expensive to set up and maintain.

Disadvantages of Network Operating System

- In this type of operating system, the failure of any node in a system affects the whole system.
- Security and performance are important issues. So trained network administrators are required for network administration.

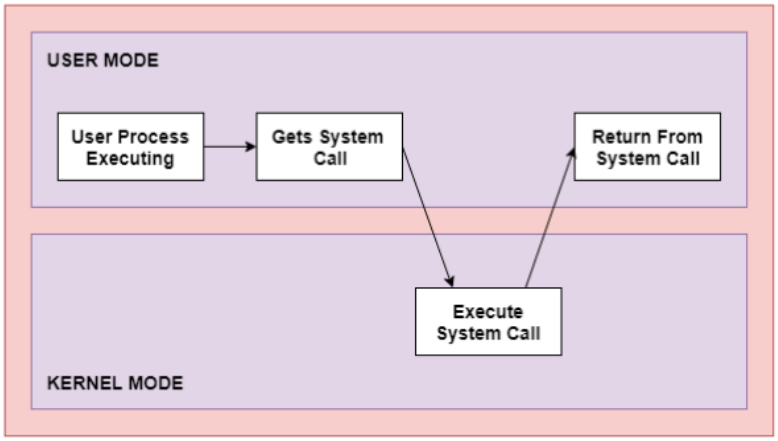
6.Mobile OS

Mobile operating systems are those OS which is especially that are designed to power smartphones, tablets, and wearables devices. Some most famous mobile operating systems are Android and iOS, but others include BlackBerry, Web, and watch OS.

system calls

system calls are privileged machine/assembly language instructions. System calls are usually made when a process in user mode requires access to a resource. Then it requests the kernel to provide the resource via a system call.

A figure representing the execution of the system call is given as follows −



As can be seen from this diagram, the processes execute normally in the user mode until a system call interrupts this. Then the system call is executed on a priority basis in the kernel mode. After the execution of the system call, the control returns to the user mode and execution of user processes can be resumed.

In general, system calls are required in the following situations −

- If a file system requires the creation or deletion of files. Reading and writing from files also require a system call.
- Creation and management of new processes.
- Network connections also require system calls. This includes sending and receiving packets.
- Access to a hardware device such as a printer, scanner etc. requires a system call

Types of system calls

- **Process control**
- **File management**

- **Device management**
- **Information maintenance**
- **Communications**
- **Protection**

Types of system calls in windows and Unix Operating systems given below.

	Windows	Unix
Process Control	CreateProcess()	fork()
	ExitProcess()	exit()
	WaitForSingleObject()	wait()
File Manipulation	CreateFile()	open()
	ReadFile()	read()
	WriteFile()	write()
	CloseHandle()	close()
Device Manipulation	SetConsoleMode()	ioctl()
	ReadConsole()	read()
	WriteConsole()	write()
Information Maintenance	GetCurrentProcessID()	getpid()
	SetTimer()	alarm()
	Sleep()	sleep()
Communication	CreatePipe()	pipe()
	CreateFileMapping()	shmget()
	MapViewOfFile()	mmap()
Protection	SetFileSecurity()	chmod()
	InitlializeSecurityDescriptor()	umask()
	SetSecurityDescriptorGroup()	chown()

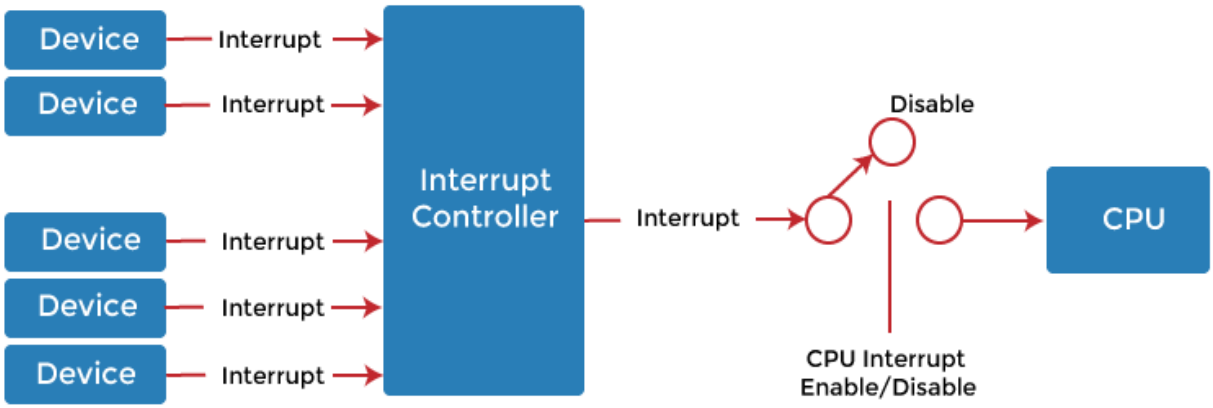
Interrupts

An interrupt is a signal from a device attached to a computer or from a program within the computer that requires the operating system to suspend current execution and transfer to requested routine.

Interrupt systems are nothing but while the CPU can process the programs if the CPU needs any IO operation. Then, it is sent to the queue and it does the CPU process. Later on Input/output (I/O) operation is ready.

The I/O devices interrupt the data which is available and does the remaining process; like that interrupts are useful. If interrupts are not present, the CPU needs to be in idle state for some time, until the IO operation needs to complete. So, to avoid the CPU waiting time interrupts are useful.

While the processor is handling the interrupts, it must inform the device that its request has been recognized to stop sending the interrupt request signal. Also, saving the registers so that the interrupted process can be restored in the future increases the delay between the time an interrupt is received and the start of the execution of the ISR. This is called ***Interrupt Latency***.



Processor handle interrupts

Whenever an interrupt occurs, it causes the CPU to stop executing the current program. Then, comes the control to interrupt handler or interrupt service routine.

These are the steps in which ISR handles interrupts. These are as follows −

Step 1 − When an interrupt occurs let assume processor is executing i'th instruction and program counter will point to the next instruction (i+1)th.

Step 2 − When an interrupt occurs the program value is stored on the process stack and the program counter is loaded with the address of interrupt service routine.

Step 3 − Once the interrupt service routine is completed the address on the process stack is popped and placed back in the program counter.

Step 4 − Now it executes the resume for (i+1)th line.

Types of interrupts: There are two types of interrupts which are as follows −

1.Hardware interrupts

The interrupt signal generated from external devices and i/o devices are made interrupt to CPU when the instructions are ready.

For example − In a keyboard if we press a key to do some action this pressing of the keyboard generates a signal that is given to the processor to do action, such interrupts are called hardware interrupts.

Hardware interrupts are classified into two types which are as follows −

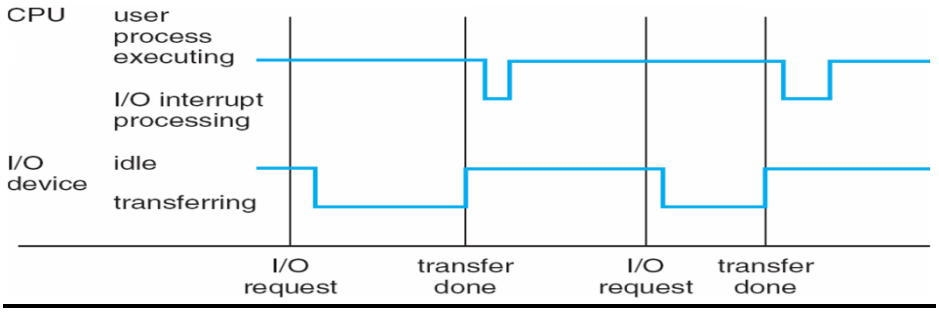
- **Maskable Interrupt** − The hardware interrupts that can be delayed when a highest priority interrupt has occurred to the processor.
- **Non Maskable Interrupt** − The hardware that cannot be delayed and immediately be serviced by the processor.

2.Software interrupts

The interrupt signal generated from internal devices and software programs need to access any system call then software interrupts are present.

Software interrupt is divided into two types. They are as follows −

- **Normal Interrupts** − The interrupts that are caused by the software instructions are called software instructions.
- **Exception** − Exception is nothing but an unplanned interruption while executing a program. For example − while executing a program if we got a value that is divided by zero is called an exception.
- **Time line diagram of interrupts**

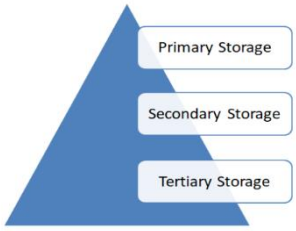


There are different types of interrupt handlers.

- ***First Level Interrupt Handler*** (FLIH) is a hard interrupt handler or fast interrupt handler. These interrupt handlers have more jitter while process execution, and they are mainly maskable interrupts.
- ***Second Level Interrupt Handler*** (SLIH) is a soft interrupt handler and slow interrupt handler. These interrupt handlers have less jitter.

Storage systems

- Computer storage has components that store computer data. The different storage types in the storage hierarchy are as follows: Storage systems organized in hierarchy
  - Speed
  - Cost
  - Volatility



1.Primary Storage

This is also known as the main memory and is the memory directly accessible by the CPU. All the instructions are executed in the main memory by CPU and the data required by these instructions is also stored in main memory.

Main memory primarily consists of the RAM which is volatile in nature. It is also quite small compared to secondary memory and expensive as well.

2.Secondary Storage

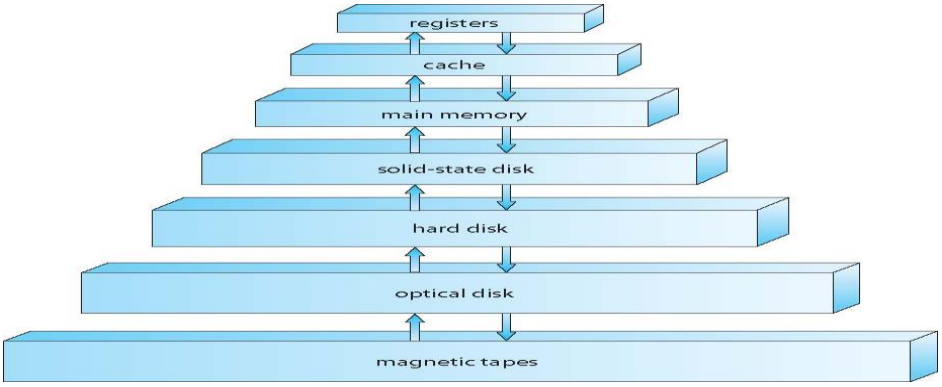
Secondary or external storage is not directly accessible by the CPU. The data from secondary storage needs to be brought into the primary storage before the CPU can use it.

Secondary storage is nonvolatile i.e., the data stored is not lost when power is switched off. Mostly hard drives are used as secondary storage.

3.Tertiary Storage

This is a third level of storage that mostly contains data that needs to be archived. This is because it is quite slow. Data is normally retrieved from tertiary storage to primary storage when it needs to be viewed. Tertiary storage is mostly formed of tapes and disks.

Storage systems organized in hierarchy- Storage structure



Characteristics of Storage

The different characteristics of storage devices are as follows:

Volatility

Volatile memory needs power to work and loses its data when power is switched off. However, it is quite fast so it is used as primary memory.

Non - volatile memory retains its data even when power is lost. So, it is used for secondary memory.

Mutability

Mutable storage is both read and write storage and data can be overwritten as required. Primary storage typically contains mutable storage and it is also available in secondary storage nowadays.

Accessibility

Storage access can be random or sequential. In random access, all the data in the storage can be accessed randomly and roughly in the same amount of time. In sequential storage, the data needs to accessed in sequential order i.e. one after the other.

Addressability

Each storage location in memory has a particular memory address. The data in a particular location can be accessed using its address.

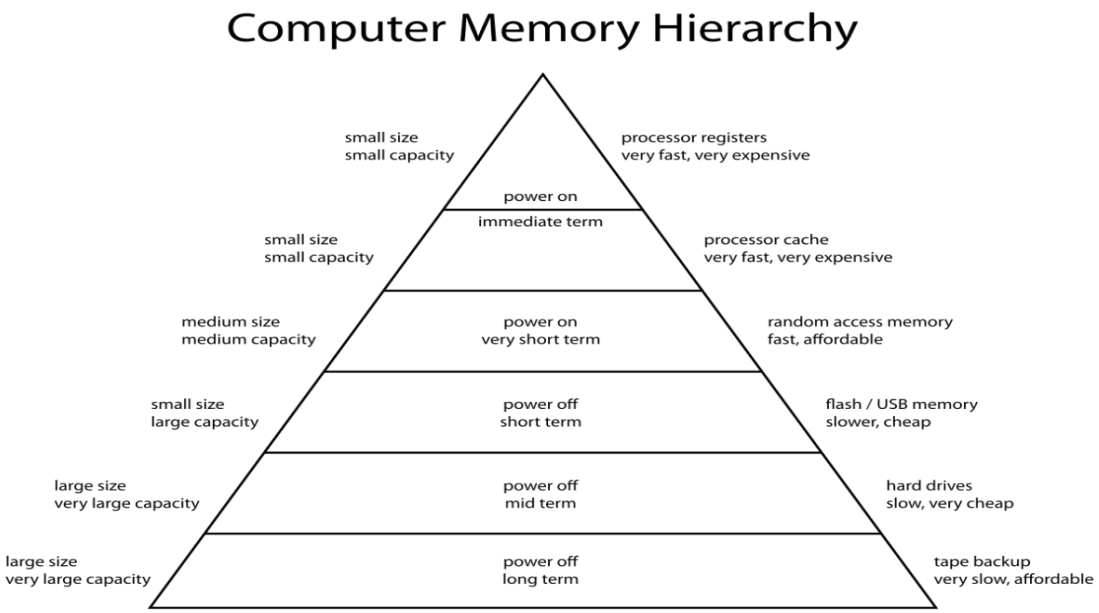
Capacity

The capacity of any storage device is the amount of data it can hold. This is usually represented in the form of bits or bytes.

Performance

Performance can be described in terms of latency or throughput.

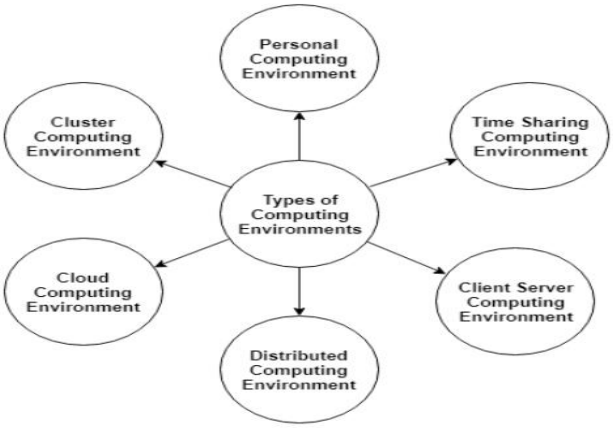
- **Latency** is the time required to access the storage. It is specified in the form of read latency and write latency.
- **Throughput** is the data reading rate for the memory. It can be represented in the form of megabytes per second.



Computing-Environments

A computer system uses many devices, arranged in different ways to solve many problems. This constitutes a computing environment where many computers are used to process and exchange information to handle multiple issues.

The different types of Computing Environments are −



- **Personal Computing Environment**
  In the personal computing environment, there is a single computer system. All the system processes are available on the computer and executed there. The different devices that constitute a personal computing environment are laptops, mobiles, printers, computer systems, scanners etc.
- **Time Sharing Computing Environment**
  The time-sharing computing environment allows multiple users to share the system simultaneously. Each user is provided a time slice and the processor switches rapidly among the users according to it. Because of this, each user believes that they are the only ones using the system.
- **Client Server Computing Environment**

In client server computing, the client requests a resource and the server provides that resource. A server may serve multiple clients at the same time while a client is in contact with only one server. Both the client and server usually communicate via a computer network but sometimes they may reside in the same system.

- **Distributed Computing Environment**
  A distributed computing environment contains multiple nodes that are physically separate but linked together using the network. All the nodes in this system communicate with each other and handle processes in tandem. Each of these nodes contains a small part of the distributed operating system software.
- **Cloud Computing Environment**
  The computing is moved away from individual computer systems to a cloud of computers in cloud computing environment. The cloud users only see the service being provided and not the internal details of how the service is provided. This is done by pooling all the computer resources and then managing them using a software.
- **Cluster Computing Environment**
  The clustered computing environment is similar to parallel computing environment 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 which then work parallel to each other.

Multiprogramming and Multi-Tasking

A multiprogramming operating system may run many programs on a single processor computer. If one program must wait for an input/output transfer in a multiprogramming operating system, the other programs are ready to use the CPU. As a result, various jobs may share CPU time. However, the execution of their jobs is not defined to be at the same time period.

When a program is being performed, it is known as a **"Task", "Process"**, and **"Job"**. Concurrent program executions improve system resource consumption and throughput as compared to serial and batch processing systems.

Types of the Multiprogramming Operating System

There are mainly two types of multiprogramming operating systems. These are as follows:

1. **Multitasking Operating System**
2. **Multiuser Operating System**

Multitasking Operating System

A multitasking operating system enables the execution of two or more programs at the same time. The operating system accomplishes this by shifting each program into and out of memory one at a time. When a program is switched out of memory, it is temporarily saved on disk until it is required again.

Multiuser Operating System

A multiuser operating system allows many users to share processing time on a powerful central computer from different terminals. The operating system accomplishes this by rapidly switching between terminals, each of which receives a limited amount of processor time on the central computer. The operating system changes among terminals so quickly that each user seems to have continuous access to the central computer. If there are many users on a system like this, the time it takes the central computer to reply can become more obvious.

Working of the Multiprogramming Operating System

Multiple users can accomplish their jobs simultaneously in the multiprogramming system, and it can be stored in the main memory. When one program is engaged in I/O operations, the CPU may deliver time to various programs while sitting in idle mode.

When one application is waiting for an I/O transfer, another is ready to use the processor at all times, and numerous programs may share CPU time. All jobs are not run simultaneously, but there could be numerous jobs running on the processor at the same time, and parts of other processes being executed first, then another segment, etc. As a result, the overall goal of a multiprogramming system is to keep the CPU busy until some tasks are available in the job pool. Thus, the numerous programs can run on a single processor computer, and the CPU is never idle.

Examples of Multiprogramming Operating System

There are various examples of multiprogramming operating systems, including download apps, transfer data, MS-Excel, Google Chrome, Firefox browser, and many more apps. Other examples are Windows O/S, UNIX O/S, Microcomputers such as XENIX, MP/M, and ESQview.

Advantages and Disadvantages of Multiprogramming Operating System

There are various advantages and disadvantages of the multiprogramming operating system. Some of the advantages and disadvantages are as follows:

Advantages

There are various advantages of the multiprogramming operating system. Some of the advantages are as follows:

1. It provides less response time.
2. It may help to run various jobs in a single application simultaneously.
3. It helps to optimize the total job throughput of the computer.
4. Various users may use the multiprogramming system at once.
5. Short-time jobs are done quickly in comparison to long-time jobs.
6. It may help to improve turnaround time for short-time tasks.
7. It helps in improving CPU utilization and never gets idle.
8. The resources are utilized smartly.

Disadvantages

There are various disadvantages of the multiprogramming operating system. Some of the disadvantages are as follows:

1. It is highly complicated and sophisticated.
2. The CPU scheduling is required.
3. Memory management is needed in the operating system because all types of tasks are stored in the main memory.

4. The harder task is to handle all processes and tasks.
5. If it has a large number of jobs, then long-term jobs will require a long wait.

What is Multiprogramming?

A multiprogramming operating system can run several programs on a single processor machine. If a single application has to wait for I/O transfer in a multiprogramming operating system, other programs are always ready to use the CPU. As a result, numerous jobs can share the CPU's time. However, in a multiprogramming operating system, it is not predefined that their jobs would be executed simultaneously.

If a program is in the process of being executed, it is referred to as **a "Process", "Job", or "Task"**. Simultaneous program executions help increase system resource utilization and system throughput performance compared to serial and batch processing systems.

Advantages and disadvantages of multiprogramming

There are various advantages and disadvantages of multiprogramming. Some of them are as follows:

Advantages

1. It provides high CPU utilization.
2. It has a shorter response time.
3. It can assign priority to the jobs.

Disadvantages

1. Its scheduling implementation is not easy.
2. It required more management.

What is Multitasking?

Multitasking means working on multiple tasks simultaneously, such as using your computer while listening to music. Also, using a browser, search for something on the internet and create a word document that is your assignment. It appears that all of the tasks are taking place at the same time. It is not all of the tasks happening simultaneously; the processor moves between them at such a fast pace that we believe they are happening simultaneously.

Multitasking is similar to multiprogramming in that the CPU is assigned to a process for a specified period of time, i.e., **'Time quantum or time slice'**, after which the CPU 'Context switches' to another process. It runs various programs at the same time.

The PC requires a huge memory to execute multitasking (**RAM or ROM**). Its primary goal is to improve the timing of the CPU's response. Users can engage with the system during multitasking, for example, by typing a letter while the printing process is running.

Multitasking is a highly complicated system. It is based on the time slice principle, which assigns a fixed amount of time to each activity to be completed. It is especially useful when a program requires a high level of parallelism. It provides a set amount of time for each program to run.

Advantages and disadvantages of multitasking

There are various advantages and disadvantages of multitasking. Some of them are as follows:

Advantages

1. It provides logical parallelism.
2. It provides a shorter response time.
3. It provides CPU utilization.

Disadvantages

1. It couldn't be executed on a slow-speed processor.
2. It needs a large amount of storage memory to do the work.

Differences between **Multiprogramming and Multitasking**

Features	Multiprogramming	Multitasking
Basic	It allows multiple programs to utilize the CPU simultaneously.	A supplementary of the multiprogramming system also allows for user interaction.
Mechanism	Based on the context switching mechanism.	Based on the time-sharing mechanism.
Objective	It is useful for reducing/decreasing CPU idle time and increasing throughput as much as possible.	It is useful for running multiple processes at the same time, effectively increasing CPU and system throughput.
Execution	When one job or process completes its execution or switches to an I/O task in a multi-programmed system, the system momentarily suspends that process. It selects another process from the process scheduling pool (waiting queue) to run.	In a multiprocessing system, multiple processes can operate simultaneously by allocating the CPU for a fixed amount of time.
CPU Switching	In a multiuser environment, the CPU switches between programs/processes quickly.	In a single-user environment, the CPU switches between the processes of various programs.
Timing	It takes maximum time to execute the process.	It takes minimum time to execute the process