1. **Operating system act as a resource management justify ?**

Let us understand how the operating system works as a Resource Manager.

Now-a-days all modern computers consist of processors, memories, timers, network interfaces, printers, and so many other devices.

The operating system provides for an orderly and controlled allocation of the processors, memories, and I/O devices among the various programs in the bottom-up view.

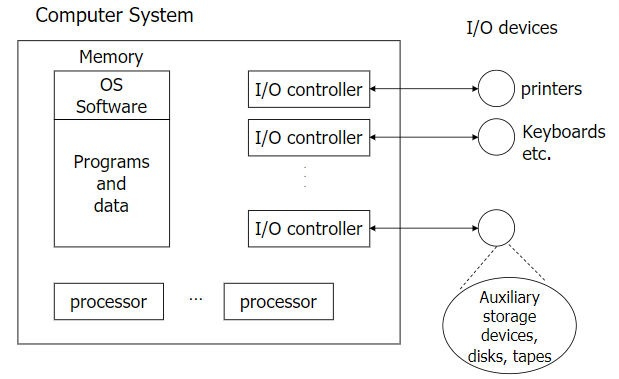
Operating system allows multiple programs to be in memory and run at the same time.

Resource management includes multiplexing or sharing resources in two different ways: in time and in space.

In time multiplexed, different programs take a chance of using CPU. First one tries to use the resource, then the next one that is ready in the queue and so on. For example: Sharing the printer one after another.

In space multiplexing, Instead of the customers taking a chance, each one gets part of the resource. For example − Main memory is divided into several running programs, so each one can be resident at the same time.

The diagram given below shows the functioning of OS as a resource manager −



1. **Diffrence between multiprogramming and multi user os**

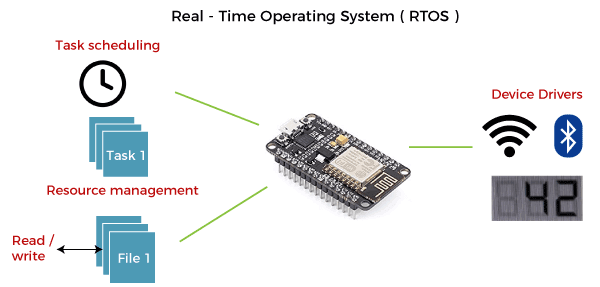
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ****S.No.**** | ****Multiprogramming**** | ****Multitasking**** | ****Multithreading**** | ****Multiprocessing**** |
| **1.** | **In multiprogramming, multiple programs execute at a same time on a single device.** | **In Multitasking, a single resource is used to process multiple tasks.** | **Multithreading is an extended form of multitasking.** | **In multiprocessing, multiple processing units are used by a single device.** |
| **2.** | **The process resides in the main memory.** | **The process resides in the same CPU.** | **More than one thread processed on a single CPU.** | **The process switches from one to another CPU as multiple processing units are used.** |
| **3.** | **It uses batch OS. The CPU is utilized completely while execution.** | **It is time sharing as the task assigned switches regularly.** | **The tasks are always further divided into sub tasks.** | **It carries multiple processors to execute the task.** |
| **4.** | **The processing is slower, as a single job resides in the main memory while execution.** | **Multitasking follows the concept of context switching.** | **It allows a single process to get multiple code segments.** | **A large amount of work can be done in a short period of time.** |

|  |
| --- |
|  |
|  |
|  |
|  |
|  |

**4.What do you mean by real time system**

****What do you mean by Real-Time Operating System?****

****A real-time operating system (RTOS)**** is a special-purpose operating system used in computers that has strict time constraints for any job to be performed. It is employed mostly in those systems in which the results of the computations are used to influence a process while it is executing. Whenever an event external to the computer occurs, it is communicated to the computer with the help of some sensor used to monitor the event. The sensor produces the signal that is interpreted by the operating system as an interrupt. On receiving an interrupt, the operating system invokes a specific process or a set of processes to serve the interrupt.



This process is completely uninterrupted unless a higher priority interrupt occurs during its execution. Therefore, there must be a strict hierarchy of priority among the interrupts. The interrupt with the highest priority must be allowed to initiate the process , while lower priority interrupts should be kept in a buffer that will be handled later. Interrupt management is important in such an operating system.

Real-time operating systems employ special-purpose operating systems because conventional operating systems do not provide such performance.

****The various examples of Real-time operating systems are:****

* MTS
* Lynx
* QNX
* VxWorks etc.

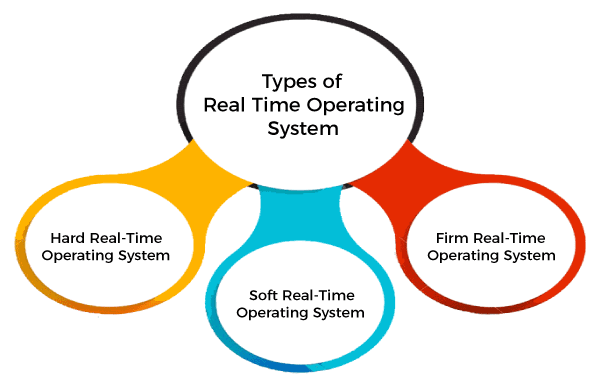
****Applications of Real-time operating system (RTOS):****

RTOS is used in real-time applications that must work within specific deadlines. Following are the common areas of applications of Real-time operating systems are given below.

* Real-time running structures are used inside the Radar gadget.
* Real-time running structures are utilized in Missile guidance.
* Real-time running structures are utilized in on line inventory trading.
* Real-time running structures are used inside the cell phone switching gadget.
* Real-time running structures are utilized by Air site visitors to manipulate structures.
* Real-time running structures are used in Medical Imaging Systems.
* Real-time running structures are used inside the Fuel injection gadget.
* Real-time running structures are used inside the Traffic manipulate gadget.
* Real-time running structures are utilized in Autopilot travel simulators.

### Types of Real-time operating system

Following are the three types of RTOS systems are:



****Hard Real-Time operating system:****

In Hard RTOS, all critical tasks must be completed within the specified time duration, i.e., within the given deadline. Not meeting the deadline would result in critical failures such as damage to equipment or even loss of human life.

****For Example,****

Let's take an example of airbags provided by carmakers along with a handle in the driver's seat. When the driver applies brakes at a particular instance, the airbags grow and prevent the driver's head from hitting the handle. Had there been some delay even of milliseconds, then it would have resulted in an accident.

Similarly, consider an on-stock trading software. If someone wants to sell a particular share, the system must ensure that command is performed within a given critical time. Otherwise, if the market falls abruptly, it may cause a huge loss to the trader.

****Soft Real-Time operating system:****

Soft RTOS accepts a few delays via the means of the Operating system. In this kind of RTOS, there may be a closing date assigned for a particular job, but a delay for a small amount of time is acceptable. So, cut off dates are treated softly via means of this kind of RTOS.

****For Example,****

This type of system is used in Online Transaction systems and Livestock price quotation Systems.

****Firm Real-Time operating system:****

In Firm RTOS additionally want to observe the deadlines. However, lacking a closing date might not have a massive effect, however may want to purposely undesired effects, like a massive discount within the fine of a product.

****For Example****, this system is used in various forms of Multimedia applications.

****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.

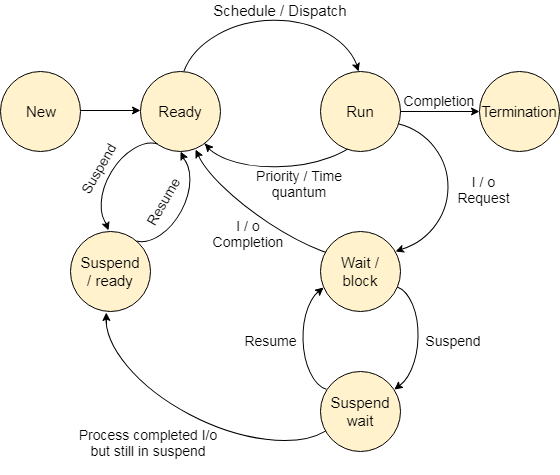
****Disadvantages of Real-time operating system:****

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.

1. **Explain the process states with a diagram**

****State Diagram****



The process, from its creation to completion, passes through various states. The minimum number of states is five.

The names of the states are not standardized although the process may be in one of the following states during execution.

### 1. New

A program which is going to be picked up by the OS into the main memory is called a new process.

### 2. Ready

Whenever a process is created, it directly enters in the ready state, in which, it waits for the CPU to be assigned. The OS picks the new processes from the secondary memory and put all of them in the main memory.

The processes which are ready for the execution and reside in the main memory are called ready state processes. There can be many processes present in the ready state.

### 3. Running

One of the processes from the ready state will be chosen by the OS depending upon the scheduling algorithm. Hence, if we have only one CPU in our system, the number of running processes for a particular time will always be one. If we have n processors in the system then we can have n processes running simultaneously.

### 4. Block or wait

From the Running state, a process can make the transition to the block or wait state depending upon the scheduling algorithm or the intrinsic behavior of the process.

When a process waits for a certain resource to be assigned or for the input from the user then the OS move this process to the block or wait state and assigns the CPU to the other processes.

### 5. Completion or termination

When a process finishes its execution, it comes in the termination state. All the context of the process (Process Control Block) will also be deleted the process will be terminated by the Operating system.

### 6. Suspend ready

A process in the ready state, which is moved to secondary memory from the main memory due to lack of the resources (mainly primary memory) is called in the suspend ready state.

If the main memory is full and a higher priority process comes for the execution then the OS have to make the room for the process in the main memory by throwing the lower priority process out into the secondary memory. The suspend ready processes remain in the secondary memory until the main memory gets available.

### 7. Suspend wait

Instead of removing the process from the ready queue, it's better to remove the blocked process which is waiting for some resources in the main memory. Since it is already waiting for some resource to get available hence it is better if it waits in the secondary memory and make room for the higher priority process. These processes complete their execution once the main memory gets available and their wait is finished.

## Operations on the Process

### 1. Creation

Once the process is created, it will be ready and come into the ready queue (main memory) and will be ready for the execution.

### 2. Scheduling

Out of the many processes present in the ready queue, the Operating system chooses one process and start executing it. Selecting the process which is to be executed next, is known as scheduling.

### 3. Execution

Once the process is scheduled for the execution, the processor starts executing it. Process may come to the blocked or wait state during the execution then in that case the processor starts executing the other processes.

### 4. Deletion/killing

Once the purpose of the process gets over then the OS will kill the process. The Context of the process (PCB) will be deleted and the process gets terminated by the Operating system.

1. **Differentiate between process and thread**

| **Comparison Basis** | **Process** | **Thread** |
| --- | --- | --- |
| Definition | A process is a program under execution i.e an active program. | A thread is a lightweight process that can be managed independently by a scheduler. |
| Context switching time | Processes require more time for context switching as they are more heavy. | Threads require less time for context switching as they are lighter than processes. |
| Memory Sharing | Processes are totally independent and don’t share memory. | A thread may share some memory with its peer threads. |
| Communication | Communication between processes requires more time than between threads. | Communication between threads requires less time than between processes . |
| Blocked | If a process gets blocked, remaining processes can continue execution. | If a user level thread gets blocked, all of its peer threads also get blocked. |
| Resource Consumption | Processes require more resources than threads. | Threads generally need less resources than processes. |
| Dependency | Individual processes are independent of each other. | Threads are parts of a process and so are dependent. |
| Data and Code sharing | Processes have independent data and code segments. | A thread shares the data segment, code segment, files etc. with its peer threads. |
| Treatment by OS | All the different processes are treated separately by the operating system. | All user level peer threads are treated as a single task by the operating system. |
| Time for creation | Processes require more time for creation. | Threads require less time for creation. |
| Time for termination | Processes require more time for termination. | Threads require less time for termination. |

1. **Explain inter process communication**

The process of communication refers to the transmission or passage of information or message from the sender through a selected channel to the receiver overcoming barriers that affect its pace.

The process of communication is a cyclic one as it begins with the sender and ends with the sender in the form of feedback. It takes place upward, downward and laterally throughout the organization.

The process of communication as such must be a continuous and dynamic interaction, both affecting and being affected by many variables.

Communication process consists of certain steps where each step constitutes the essential of an effective communication.

The following is a brief analysis of the important steps of the process of communication.

## The Different Elements in The Process of Communication

We will now learn about the different elements in the process of communication.

### Sender

The very foundation of communication process is laid by the person who transmits or sends the message. He is the sender of the message which may be a thought, idea, a picture, symbol, report or an order and postures and gestures, even a momentary smile. The sender is therefore the initiator of the message that need to be transmitted. After having generated the idea, information etc. the sender encodes it in such a manner that can be well-understood by the receiver.

### Message

Message is referred to as the information conveyed by words as in speech and write-ups, signs, pictures or symbols depending upon the situation and the nature and importance of information desired to be sent. Message is the heart of communication. It is the content the sender wants to covey to the receiver. It can be verbal both written and spoken; or non-verbal i.e. pictorial or symbolic, etc.

### Encoding

Encoding is putting the targeted message into appropriate medium which may be verbal or non-verbal depending upon the situation, time, space and nature of the message to be sent. The sender puts the message into a series of symbols, pictures or words which will be communicated to the intended receiver. Encoding is an important step in the communication process as wrong and inappropriate encoding may defeat the true intent of the communication process.

### Channel

Channel(s) refers to the way or mode the message flows or is transmitted through. The message is transmitted over a channel that links the sender with the receiver. The message may be oral or written and it may be transmitted through a memorandum, a computer, telephone, cell phone, apps or televisions.



Since each channel has its advantages and disadvantages, the choice of proper selection of the channel is paramount for effective communication.

### Receiver

Receiver is the person or group who the message is meant for. He may be a listener, a reader or a viewer. Any negligence on the part of the receiver may make the communication ineffective. The receiver needs to comprehend the message sent in the best possible manner such that the true intent of the communication is attained. The extent to which the receiver decodes the message depends on his/her knowledge of the subject matter of the message, experience, trust and relationship with the sender.

The receiver is as significant a factor in communication process as the sender is. It is the other end of the process. The receiver should be in fit condition to receive the message, that is, he/she should have channel of communication active and should not be preoccupied with other thoughts that might cause him/her to pay insufficient attention to the message.

### Decoding

Decoding refers to interpreting or converting the sent message into intelligible language. It simply means comprehending the message. The receiver after receiving the message interprets it and tries to understand it in the best possible manner.

### Feedback

Feedback is the ultimate aspect of communication process. It refers to the response of the receiver as to the message sent to him/her by the sender. Feedback is necessary to ensure that the message has been effectively encoded, sent, decoded and comprehended.

It is the final step of the communication process and establishes that the receiver has received the message in its letter and spirit. In other words, the receiver has correctly interpreted the message as it was intended by the sender. It is instrumental to make communication effective and purposeful.

Consider the following points related to the feedback involved in the process of communication −

It enhances the effectiveness of the communication as it permits the sender to know the efficacy of his message.

It enables the sender to know if his/her message has been properly comprehended.

The analysis of feedbacks helps improve future messages. Feedback, like the message, can be verbal or nonverbal and transmitted through carefully chosen channel of communication.

We can represent the above steps in a model as the model of communication process.

### Types of Feedback

Kevin Eujeberry, the world famous leadership exponent mentioned the four types of feedback. The types are as follows −

Negative Feedback or corrective comments about past behavior

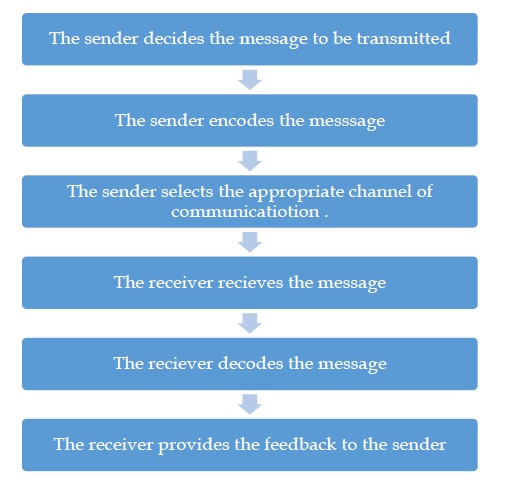
Positive Feedback or affirming comments about future behavior

Negative *feedforward* or corrective comments about future behavior

Positive *feedforward* or affirming comments about future behavior

## The Model of Communication Process

Let us now see the model of communication process −



The process of communication, however, is not as smooth or barrier-free as it seems. From its transmission to receipt, the message may get interfered or disturbed with at any stage by many factors which are known as barriers to effective communication. One of the factors is poor choice of communication method. In addition to a poor choice of communication method, other barriers to effective communication include noise and other physical distractions, language problems, and failure to recognize nonverbal signals. We will discuss these barriers of communication in a subsequent chapter.

## Summary

The process of communication refers to the transmission or passage of information or message from the sender through a selected channel to the receiver overcoming barriers that affect its pace.

The different elements in the process of communication −

Sender

Message

Encoding

Channel

Receiver

Decoding

Feedback

Kevin Eujeberry, the world famous leadership exponent mentioned the four types of feedback. The types are as follows −

Negative Feedback or corrective comments about past behavior

Positive Feedback or affirming comments about future behavior

Negative feedforward or corrective comments about future behavior

Positive feedforward or affirming comments about future behavior

**8. Explain memory allocation strategies**

# Memory Allocation

13th August 2019 by [Neha T](https://binaryterms.com/author/author) [4 Comments](https://binaryterms.com/static-and-dynamic-memory-allocation.html" \l "comments)

****Memory allocation**** is an action of assigning the physical or the virtual memory address space to a process (its instructions and data). The two fundamental methods of memory allocation are static and dynamic memory allocation.

The static memory allocation method assigns the memory to a process, ****before its execution****. On the other hand, the dynamic memory allocation method assigns the memory to a process, ****during its execution****.

In this section, we will be discussing what is memory allocation, its types (static and dynamic memory allocation) along with their advantages and disadvantages. So let us start.

### Content: Static and Dynamic Memory Allocation

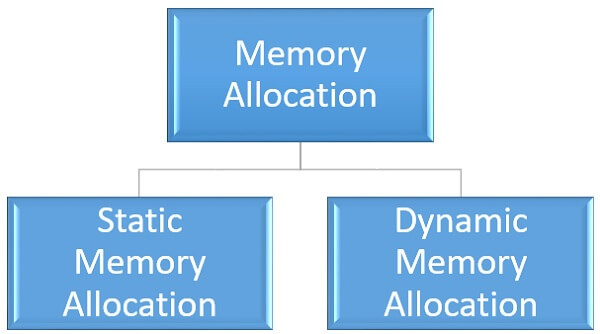
1. [Memory Allocation](https://binaryterms.com/static-and-dynamic-memory-allocation.html" \l "MemoryAllocation)
2. [Types of Memory Allocation](https://binaryterms.com/static-and-dynamic-memory-allocation.html" \l "TypesofMemoryAllocation)
3. [Advantages and Disadvantages of Static and Dynamic Memory Allocation](https://binaryterms.com/static-and-dynamic-memory-allocation.html" \l "AdvantagesandDisadvantagesofStaticandDynamicMemoryAllocation)
4. [Key Takeaways](https://binaryterms.com/static-and-dynamic-memory-allocation.html" \l "KeyTakeaways)

## Memory Allocation

To get a process executed it must be first placed in the memory. Assigning space to a process in memory is called memory allocation. Memory allocation is a general aspect of the term ****binding.****

Let us understand binding with the help of an example. Suppose, there is an ****entity**** in a program, with a set of attributes. Now, a ****variable**** of this entity will have values for this ****set of attributes****. For storing these values, we must have memory allotted to these attributes.

So, the act of assigning the memory address to the attribute of the variable is called ****memory allocation****. And the act of specifying/binding the values to the attributes of the variable is called ****binding****. This action of binding must be performed before the variable is used during the execution of the program.

We have two types of memory allocation or we can say two methods of binding, static and dynamic binding.  


## Types of Memory Allocation

### 1. Static Memory Allocation

Static memory allocation is performed when the compiler compiles the program and generates object files. The linker merges all these object files and creates a single executable file. The loader loads this single executable file in the main memory, for execution. In static memory allocation, the size of the data required by the process must be known ****before**** the execution of the process initiates.

If the data sizes are not known before the execution of the process, then they have to be guessed. If the data size guessed is larger than the required, then it leads to ****wastage**** of memory. If the guessed size is smaller, then it leads to inappropriate execution of the process.

The static memory allocation method does not need any memory allocation operation during the execution of the process. All the memory allocation operation required for the process is done before the execution of the process has started. So, it leads to ****faster**** execution of a process.

Static memory allocation provides more ****efficiency**** when compared to dynamic memory allocation.

### 2. Dynamic Memory Allocation

Dynamic memory allocation is performed while the program is in execution. Here, the memory is allocated to the entities of the program when they are to be used for the ****first time**** while the program is running.

The actual size, of the data required, is known at the run time so, it allocates the ****exact**** memory space to the program thereby, reducing the memory wastage.

Dynamic memory allocation provides ****flexibility**** to the execution of the program. As it can decide what amount of memory space will be required by the program. If the program is large enough then a dynamic memory allocation is performed on the different parts of the program, which is to be used currently. This reduces memory wastage and improves the performance of the system.

Allocating memory dynamically creates an overhead over the system. Some allocation operations are performed repeatedly during the program execution creating more overheads, leading in ****slow**** execution of the program.

Dynamic memory allocation does not require special support from the operating system. It is the responsibility of the programmer to design the program in a way to take advantage of dynamic memory allocation method.

Thus the dynamic memory allocation is flexible but slower than static memory allocation.

### Advantages of static and dynamic memory allocation

#### Static Memory Allocation

1. Static memory allocation provides an ****efficient****way of assigning the memory to a process.
2. All the memory assigning operations are done before the execution starts. So, there are no ****overheads**** of memory allocation operations at the time of execution of the program.
3. Static memory allocation provides ****faster****execution, as at the time of execution it doesn’t have to waste time in allocation memory to the program.

#### Dynamic Memory Allocation

1. Dynamic memory allocation provides a ****flexible****way of assigning the memory to a process.
2. Dynamic memory allocation ****reduces****the memory ****wastage****as it assigns memory to a process during the execution of that program. So, it is aware of the exact memory size required by the program.
3. If the program is large then the dynamic memory allocation is performed on the different parts of the program. Memory is assigned to the part of a program that is currently in use. This also reduces memory wastage and indeed improves ****system performance****.

### Disadvantages of static and dynamic memory allocation

#### Static Memory Allocation

1. In static memory allocation, the system is ****unaware****of the memory requirement of the program. So, it has to guess the memory required for the program.
2. Static memory allocation leads to memory ****wastage****. As it estimates the size of memory required by the program. So, if the estimated size is larger, it will lead to memory ****wastage****else if the estimated size is smaller, then the program will execute ****inappropriately****.

#### Dynamic Memory allocation

1. Dynamic memory allocation method has an ****overhead****of assigning the memory to a process during the time of its execution.
2. Sometimes the memory allocation actions are repeated several times during the execution of the program which leads to more ****overheads****.
3. The overheads of memory allocation at the time of its execution ****slowdowns****the execution to some extent.

### Key Takeaways

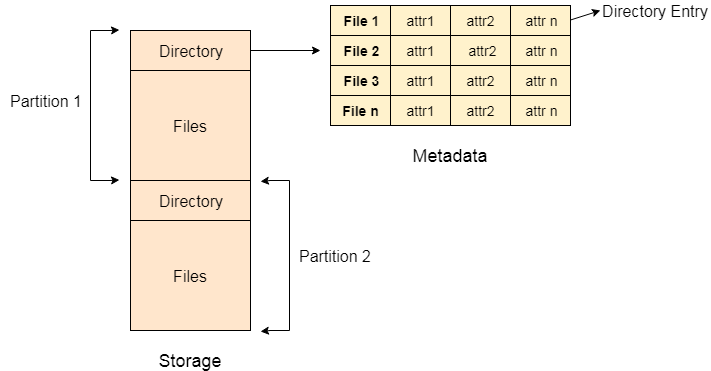
* Memory allocation specifies the memory address to a program or a process.
* Memory allocation has two methods ****static****memory allocation and ****dynamic****memory allocation.
* Static memory allocation provides ****efficiency****as it assigns the memory to a process ****before****its execution has started. So it doesn’t have any ****overhead****of memory allocation operation during the execution of the program which leads to ****faster**** execution of the program.
* In static memory allocation, the required memory size must be known ****prior****to the execution of the program.
* Static memory allocation assigns the assumed amount of memory space to a process as it is ****unaware****of the amount of memory required by the program. This leads to the wastage of memory.
* Dynamic memory allocation is performed ****during****the time of execution of a program. So it allocates the exact amount of memory to the program avoiding memory wastage.
* Dynamic memory allocation has the ****overheads****of memory allocation operation during the execution of the program which ****slowdowns****the execution of the program.
* Dynamic memory allocation provides ****flexibility****during memory allocation, as if the program is large enough then it performs memory allocation operations on different parts of the programs and reduces memory wastage

1. **What do you mean by directory structure in operating system ?**

Directory can be defined as the listing of the related files on the disk. The directory may store some or the entire file attributes.

To get the benefit of different file systems on the different operating systems, A hard disk can be divided into the number of partitions of different sizes. The partitions are also called volumes or mini disks.

Each partition must have at least one directory in which, all the files of the partition can be listed. A directory entry is maintained for each file in the directory which stores all the information related to that file.



A directory can be viewed as a file which contains the Meta data of the bunch of files.

Every Directory supports a number of common operations on the file:

1. File Creation
2. Search for the file
3. File deletion
4. Renaming the file
5. Traversing Files
6. Listing of files
7. **define os system.explain its function**

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.

An operating system is software that enables applications to interact with a computer's hardware. The software that contains the core components of the operating system is called the **kernel**.

The primary purposes of an **Operating System** are to enable applications (spftwares) to interact with a computer's hardware and to manage a system's hardware and software resources.

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 mobile phones, personal computers, mainframe computers, automobiles, TV, Toys etc.

## Definitions

We can have a number of definitions of an Operating System. Let's go through few of them:

An Operting System is the low-level software that supports a computer's basic functions, such as scheduling tasks and controlling peripherals.

We can refine this definition as follows:

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.

Following is another definition taken from Wikipedia:

An operating system (OS) is system software that manages computer hardware, software resources, and provides common services for computer programs.

Following are some of important functions of an operating System.

* 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

## 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 −

Keeps track of information, location, uses, status etc. The collective facilities are often known as **file system**.

Decides who gets the resources.

Allocates the resources.

De-allocates the resources.

## 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 softwares and users** − Coordination and assignment of compilers, interpreters, assemblers and other software to the various users of the computer systems.

1. **Explain the diffrence between non premptive and preempetive scheduling with FCFS and RR algorithm.**

## What is Preemptive Scheduling?

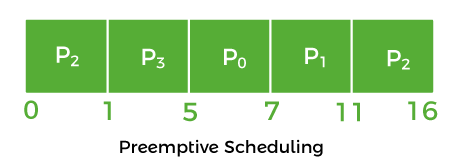
Preemptive scheduling is a method that may be used when a process switches from a running state to a ready state or from a waiting state to a ready state. The resources are assigned to the process for a particular time and then removed. If the resources still have the remaining CPU burst time, the process is placed back in the ready queue. The process remains in the ready queue until it is given a chance to execute again.

When a high-priority process comes in the ready queue, it doesn't have to wait for the running process to finish its burst time. However, the running process is interrupted in the middle of its execution and placed in the ready queue until the high-priority process uses the resources. As a result, each process gets some CPU time in the ready queue. It improves the overhead of switching a process from running to ready state and vice versa, increasing preemptive scheduling flexibility. It may or may not include SJF and Priority scheduling.

****For example:****

Let us take the example of Preemptive Scheduling. We have taken ****P0, P1, P2,**** and ****P3**** are the four processes.

|  |  |  |
| --- | --- | --- |
| **Process** | **Arrival Time** | **CPU Burst time (in millisec.)** |
| ****P0**** | 3 | 2 |
| ****P1**** | 2 | 4 |
| ****P2**** | 0 | 6 |
| ****P3**** | 1 | 4 |



* Firstly, the process ****P2**** comes at time ****0****. So, the CPU is assigned to process ****P2****.
* When process ****P2**** was running, process P3 arrived at time 1, and the remaining time for process ****P2 (5 ms)**** is greater than the time needed by process ****P3 (4 ms)****. So, the processor is assigned to P3.
* When process ****P3**** was running, process ****P1**** came at time ****2****, and the remaining time for process ****P3 (3 ms)**** is less than the time needed by processes ****P1 (4 ms)**** and ****P2 (5 ms)****. As a result, ****P3**** continues the execution.
* When process ****P3**** continues the process, process ****P0**** arrives at time ****3****. P3's remaining time ****(2 ms)**** is equal to ****P0's**** necessary time ****(2 ms)****. So, process ****P3**** continues the execution.
* When process ****P3**** finishes, the CPU is assigned to ****P0****, which has a shorter burst time than the other processes.
* After process ****P0**** completes, the CPU is assigned to process ****P1**** and then to process ****P2****.

### Advantages and disadvantages of Preemptive Scheduling

There are various advantages and disadvantages of Preemptive scheduling. The advantages and disadvantages of non-preemptive scheduling are as follows:

****Advantages****

1. It is a more robust method because a process may not monopolize the processor.
2. Each event causes an interruption in the execution of ongoing tasks.
3. It improves the average response time.
4. It is more beneficial when you use this method in a multi-programming environment.
5. The operating system ensures that all running processes use the same amount of CPU.

****Disadvantages****

1. It requires the use of limited computational resources.
2. It takes more time suspending the executing process, switching the context, and dispatching the new incoming process.
3. If several high-priority processes arrive at the same time, the low-priority process would have to wait longer.

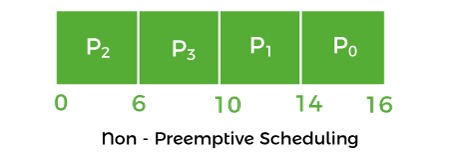
## What is Non-Preemptive Scheduling?

Non-preemptive scheduling is a method that may be used when a process terminates or switches from a running to a waiting state. When processors are assigned to a process, they keep the process until it is eliminated or reaches a waiting state. When the processor starts the process execution, it must complete it before executing the other process, and it may not be interrupted in the middle.

When a non-preemptive process with a high CPU burst time is running, the other process would have to wait for a long time, and that increases the process average waiting time in the ready queue. However, there is no overhead in transferring processes from the ready queue to the CPU under non-preemptive scheduling. The scheduling is strict because the execution process is not even preempted for a higher priority process.

****For example:****

Let's take the above preemptive scheduling example and solve it in a non-preemptive manner.



* The process ****P2**** comes at ****0****, so the processor is assigned to process ****P2**** and takes ****(6 ms)**** to execute.
* All of the processes, ****P0, P1****, and ****P3****, arrive in the ready queue in between. But all processes wait till process ****P2**** finishes its CPU burst time.
* After that, the process that comes after process ****P2****, i.e., ****P3****, is assigned to the CPU until it finishes its burst time.
* When process ****P1**** completes its execution, the CPU is given to process ****P0****.

### Advantages and disadvantages of Non-preemptive Scheduling

There are various advantages and disadvantages of non-preemptive scheduling. The advantages and disadvantages of non-preemptive scheduling are as follows:

****Advantages****

1. It provides a low scheduling overhead.
2. It is a very simple method.
3. It uses less computational resources.
4. It offers high throughput.

****Disadvantages****

1. It has a poor response time for the process.
2. A machine can freeze up due to bugs.

## Main Differences between the Preemptive and Non-Preemptive Scheduling



Here, you will learn the main differences between Preemptive and Non-Preemptive Scheduling. Various differences between the Preemptive and Non-Preemptive Scheduling are as follows:

1. In preemptive scheduling, the CPU is assigned to the processes for a particular time period. In contrast, the CPU is assigned to the process until it removes and switches to the waiting state.
2. When a process with a high priority appears in the ready queue frequently in preemptive scheduling, the process with a low priority must wait for a long period and can starve. In contrast, when the CPU is assigned to the process with the high burst time, the processes with the shorter burst time can starve in non-preemptive scheduling.
3. When a higher priority process comes in the CPU, the running process in preemptive scheduling is halted in the middle of its execution. On the other hand, the running process in non-preemptive scheduling doesn't interrupt in the middle of its execution and waits until it is completed.
4. Preemptive scheduling is flexible in processing. On the other side, non-preemptive is strict.
5. Preemptive scheduling is quite flexible because critical processes are allowed to access the CPU because they come in the ready queue and no matter which process is currently running. Non-preemptive scheduling is tough because if an essential process is assigned to the ready queue, the CPU process is not be interrupted.
6. In preemptive scheduling, CPU utilization is more effective than non-preemptive scheduling. On the other side, in non-preemptive scheduling, the CPU utilization is not effective as preemptive scheduling.
7. Preemptive scheduling is very cost-effective because it ensures the integrity of shared data. In contrast, it is not in the situation of non-preemptive scheduling.

## Head-to-head Comparison between the Preemptive and Non-Preemptive Scheduling

Here, you will learn the head-to-head comparison between preemptive and non-preemptive scheduling. The main differences between preemptive and non-preemptive scheduling are as follows:

|  |  |
| --- | --- |
| **Preemptive Scheduling** | **Non-Preemptive Scheduling** |
| The resources are assigned to a process for a long time period. | Once resources are assigned to a process, they are held until it completes its burst period or changes to the waiting state. |
| Its process may be paused in the middle of the execution. | When the processor starts the process execution, it must complete it before executing the other process, and it may not be interrupted in the middle. |
| When a high-priority process continuously comes in the ready queue, a low-priority process can starve. | When a high burst time process uses a CPU, another process with a shorter burst time can starve. |
| It is flexible. | It is rigid. |
| It is cost associated. | It does not cost associated. |
| It has overheads associated with process scheduling. | It doesn't have overhead. |
| It affects the design of the operating system kernel. | It doesn't affect the design of the OS kernel. |
| Its CPU utilization is very high. | Its CPU utilization is very low. |
| Examples: Round Robin and Shortest Remaining Time First | FCFS and SJF are examples of non-preemptive scheduling. |

1. **Define deadlocks with real life example.Explain the characteristics of Deadlocks.Explain how deadlocks can be prevented ?**

Every process needs some resources to complete its execution. However, the resource is granted in a sequential order.

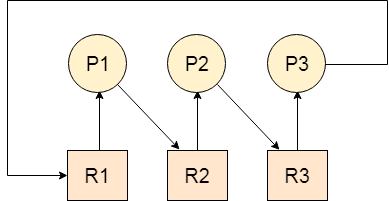
1. The process requests for some resource.
2. OS grant the resource if it is available otherwise let the process waits.
3. The process uses it and release on the completion.

A Deadlock is a situation where each of the computer process waits for a resource which is being assigned to some another process. In this situation, none of the process gets executed since the resource it needs, is held by some other process which is also waiting for some other resource to be released.

Let us assume that there are three processes P1, P2 and P3. There are three different resources R1, R2 and R3. R1 is assigned to P1, R2 is assigned to P2 and R3 is assigned to P3.

After some time, P1 demands for R1 which is being used by P2. P1 halts its execution since it can't complete without R2. P2 also demands for R3 which is being used by P3. P2 also stops its execution because it can't continue without R3. P3 also demands for R1 which is being used by P1 therefore P3 also stops its execution.

In this scenario, a cycle is being formed among the three processes. None of the process is progressing and they are all waiting. The computer becomes unresponsive since all the processes got blocked.

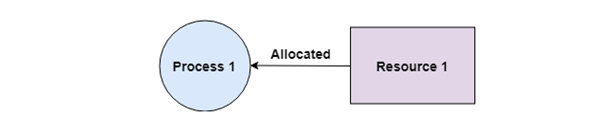


A deadlock happens in operating system when two or more processes need some resource to complete their execution that is held by the other process.

A deadlock occurs if the four Coffman conditions hold true. But these conditions are not mutually exclusive. They are given as follows −

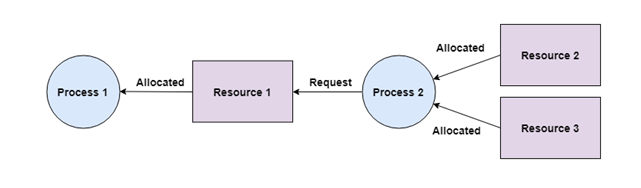
## Mutual Exclusion

There should be a resource that can only be held by one process at a time. In the diagram below, there is a single instance of Resource 1 and it is held by Process 1 only.



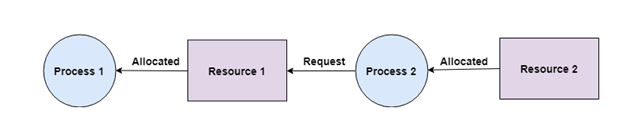
## Hold and Wait

A process can hold multiple resources and still request more resources from other processes which are holding them. In the diagram given below, Process 2 holds Resource 2 and Resource 3 and is requesting the Resource 1 which is held by Process 1.



## No Preemption

A resource cannot be preempted from a process by force. A process can only release a resource voluntarily. In the diagram below, Process 2 cannot preempt Resource 1 from Process 1. It will only be released when Process 1 relinquishes it voluntarily after its execution is complete.



## Circular Wait

A process is waiting for the resource held by the second process, which is waiting for the resource held by the third process and so on, till the last process is waiting for a resource held by the first process. This forms a circular chain. For example: Process 1 is allocated Resource2 and it is requesting Resource 1. Similarly, Process 2 is allocated Resource 1 and it is requesting Resource 2. This forms a circular wait loop.

1. **What are the different file allocation method? Explain.**

# **[File Allocation Methods in Operating System](https://www.tutorialandexample.com/file-allocation-methods)**

****File Allocation Methods****

There are different kinds of methods that are used to allocate disk space. We must select the best method for the file allocation because it will directly affect the system performance and system efficiency. With the help of the allocation method, we can utilize the disk, and also files can be accessed.

There are various types of file allocations method:

1. Contiguous allocation
2. Extents
3. Linked allocation
4. Clustering
5. FAT
6. Indexed allocation
7. Linked Indexed allocation
8. Multilevel Indexed allocation
9. Inode

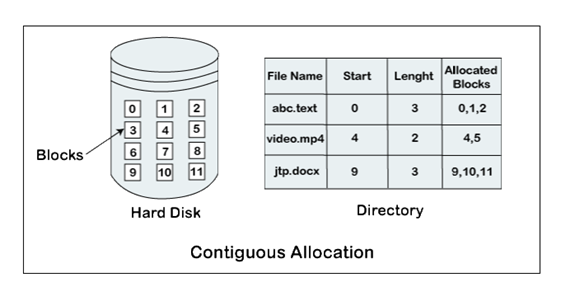
There are different types of file allocation methods, but we mainly use three types of file allocation methods:

1. Contiguous allocation
2. Linked list allocation
3. Indexed allocation

These methods provide quick access to the file blocks and also the utilization of disk space in an efficient manner.

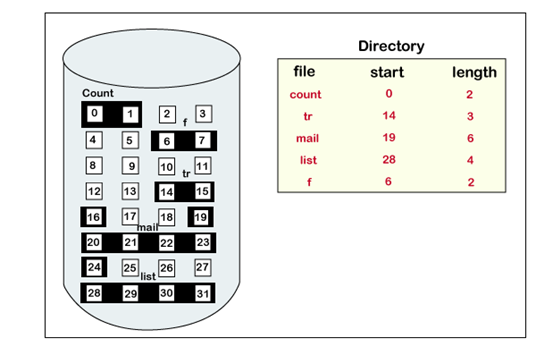
****Contiguous Allocation: -****Contiguous allocation is one of the most used methods for allocation. Contiguous allocation means we allocate the block in such a manner, so that in the hard disk, all the blocks get the contiguous physical block.

We can see in the below figure that in the directory, we have three files. In the table, we have mentioned the starting block and the length of all the files. We can see in the table that for each file, we allocate a contiguous block.



****Example of contiguous allocation****

We can see in the given diagram, that there is a file. The name of the file is ‘mail.’ The file starts from the 19th block and the length of the file is 6. So, the file occupies 6 blocks in a contiguous manner. Thus, it will hold blocks 19, 20, 21, 22, 23, 24.



### **Advantages of Contiguous Allocation**

The advantages of contiguous allocation are:

1. The contiguous allocation method gives excellent read performance.
2. Contiguous allocation is easy to implement.
3. The contiguous allocation method supports both types of file access methods that are sequential access and direct access.
4. The Contiguous allocation method is fast because, in this method number of seeks is less due to the contiguous allocation of file blocks.

### **Disadvantages of Contiguous allocation**

The disadvantages of contiguous allocation method are:

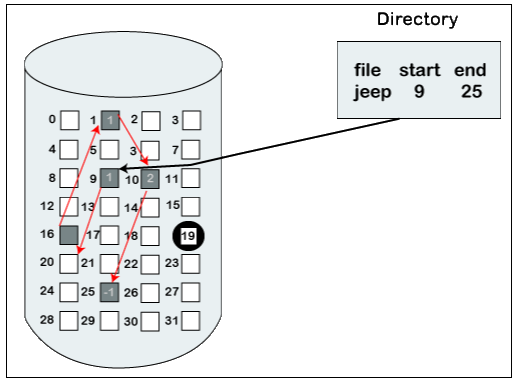
1. In the contiguous allocation method, sometimes disk can be fragmented.
2. In this method, it is difficult to increase the size of the file due to the availability of the contiguous memory block.

### **Linked List Allocation**

The linked list allocation method overcomes the drawbacks of the contiguous allocation method. In this file allocation method, each file is treated as a linked list of disks blocks. In the linked list allocation method, it is not required that disk blocks assigned to a specific file are in the contiguous order on the disk. The directory entry comprises of a pointer for starting file block and also for the ending file block. Each disk block that is allocated or assigned to a file consists of a pointer, and that pointer point the next block of the disk, which is allocated to the same file.

****Example of linked list allocation****

We can see in the below figure that we have a file named ‘jeep.’ The value of the start is 9. So, we have to start the allocation from the 9th block, and blocks are allocated in a random manner. The value of the end is 25. It means the allocation is finished on the 25th block. We can see in the below figure that the block (25) comprised of -1, which means a null pointer, and it will not point to another block.



### **Advantages of Linked list allocation**

There are various advantages of linked list allocation:

1. In liked list allocation, there is no external fragmentation. Due to this, we can utilize the memory better.
2. In linked list allocation, a directory entry only comprises of the starting block address.
3. The linked allocation method is flexible because we can quickly increase the size of the file because, in this to allocate a file, we do not require a chunk of memory in a contiguous form.

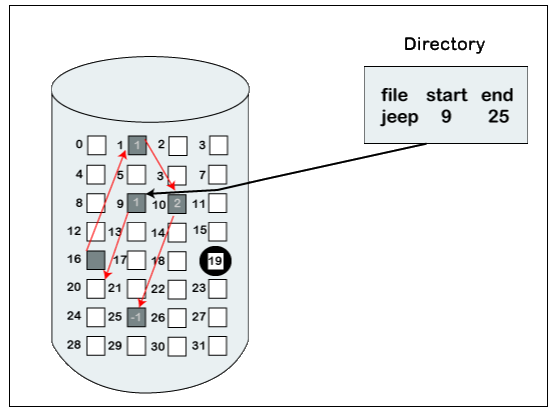
### **Disadvantages of Linked list Allocation**

There are various disadvantages of linked list allocation:

1. Linked list allocation does not support direct access or random access.
2. In linked list allocation, we need to traverse each block.
3. If the pointer in the linked list break in linked list allocation, then the file gets corrupted.
4. In the disk block for the pointer, it needs some extra space.

### **Indexed Allocation**

 The Indexed allocation method is another method that is used for file allocation. In the index allocation method, we have an additional block, and that block is known as the index block. For each file, there is an individual index block. In the index block, the ith entry holds the disk address of the ith file block. We can see in the below figure that the directory entry comprises of the address of the index block.



### **Advantages of Index Allocation**

The advantages of index allocation are:

1. The index allocation method solves the problem of external fragmentation.
2. Index allocation provides direct access.

### **Disadvantages of Index Allocation**

The disadvantages of index allocation are:

1. In index allocation, pointer overhead is more.
2. We can lose the entire file if an index block is not correct.
3. It is totally a wastage to create an index for a small file.

A single index block cannot hold all the pointer for files with large sizes.

To resolve this problem, there are various mechanism which we can use:

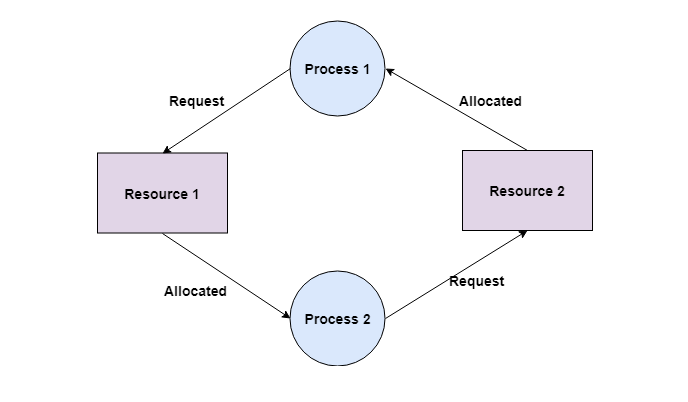
1. Linked scheme
2. Multilevel Index
3. Combined Scheme
4. ****Linked Scheme: -****In the linked scheme, to hold the pointer, two or more than two index blocks are linked together. Each block contains the address of the next index block or a pointer.
5. ****Multilevel Index: -****In the multilevel index, to point the second-level index block, we use a first-level index block that in turn points to the blocks of the disk, occupied by the file. We can extend this up to 3 or more than 3 levels depending on the maximum size of the file.
6. ****Combined Scheme: -****In a combined scheme, there is a special block which is called an information node (Inode). The inode comprises of all the information related to the file like authority, name, size, etc. To store the disk block addresses that contain the actual file, the remaining space of inode is used. In inode, the starting pointer is used to point the direct blocks. This means the pointer comprises of the addresses of the disk blocks, which consist of the file data. To indicate the indirect blocks, the next few pointers are used. The indirect blocks are of three types, which are single indirect, double indirect, and triple indirect.

****Inode****

In the UNIX operating system, every file is indexed with the help of Inode. An Inode is a block that is created at the time when the file system is designed.

There are various types of information included in Inode:

1. Attributes of the file, such as timestamp, permissions, details, ownership, etc.
2. The total number of direct blocks that comprise of the pointer to the starting blocks.
3. In Inode, there is a single indirect pointer. It is used to point an index block. If using direct blocks, entire file cannot be indexed, then, in that case, we use a single indirect pointer.
4. Inode also contains a double indirect pointer. This pointer is used to point a disk block.
5. In Inode there is another pointer, which is a triple index pointer. This pointer is also used to point a disk block.



**BONOUS**

**Components of os**

There are various components of an Operating System to perform well defined tasks. Though most of the Operating Systems differ in structure but logically they have similar components. Each component must be a well-defined portion of a system that appropriately describes the functions, inputs, and outputs.

There are following 8-components of an Operating System:

1. Process Management
2. I/O Device Management
3. File Management
4. Network Management
5. Main Memory Management
6. Secondary Storage Management
7. Security Management
8. Command Interpreter System

Following section explains all the above components in more detail:

## Process Management

A process is program or a fraction of a program that is loaded in main memory. A process needs certain resources including CPU time, Memory, Files, and I/O devices to accomplish its task. The process management component manages the multiple processes running simultaneously on the Operating System.

A program in running state is called a process.

The operating system is responsible for the following activities in connection with process management:

* Create, load, execute, suspend, resume, and terminate processes.
* Switch system among multiple processes in main memory.
* Provides communication mechanisms so that processes can communicate with each others
* Provides synchronization mechanisms to control concurrent access to shared data to keep shared data consistent.
* Allocate/de-allocate resources properly to prevent or avoid deadlock situation.

## I/O Device Management

One of the purposes of an operating system is to hide the peculiarities of specific hardware devices from the user. I/O Device Management provides an abstract level of H/W devices and keep the details from applications to ensure proper use of devices, to prevent errors, and to provide users with convenient and efficient programming environment.

Following are the tasks of I/O Device Management component:

* Hide the details of H/W devices
* Manage main memory for the devices using cache, buffer, and spooling
* Maintain and provide custom drivers for each device.

## File Management

File management is one of the most visible services of an operating system. Computers can store information in several different physical forms; magnetic tape, disk, and drum are the most common forms.

A file is defined as a set of correlated information and it is defined by the creator of the file. Mostly files represent data, source and object forms, and programs. Data files can be of any type like alphabetic, numeric, and alphanumeric.

A files is a sequence of bits, bytes, lines or records whose meaning is defined by its creator and user.

The operating system implements the abstract concept of the file by managing mass storage device, such as types and disks. Also files are normally organized into directories to ease their use. These directories may contain files and other directories and so on.

The operating system is responsible for the following activities in connection with file management:

* File creation and deletion
* Directory creation and deletion
* The support of primitives for manipulating files and directories
* Mapping files onto secondary storage
* File backup on stable (nonvolatile) storage media

## Network Management

The definition of network management is often broad, as network management involves several different components. Network management is the process of managing and administering a computer network. A computer network is a collection of various types of computers connected with each other.

Network management comprises fault analysis, maintaining the quality of service, provisioning of networks, and performance management.

Network management is the process of keeping your network healthy for an efficient communication between different computers.

Following are the features of network management:

* Network administration
* Network maintenance
* Network operation
* Network provisioning
* Network security

## Main Memory Management

Memory is a large array of words or bytes, each with its own address. It is a repository of quickly accessible data shared by the CPU and I/O devices.

Main memory is a volatile storage device which means it loses its contents in the case of system failure or as soon as system power goes down.

The main motivation behind Memory Management is to maximize memory utilization on the computer system.

The operating system is responsible for the following activities in connections with memory management:

* Keep track of which parts of memory are currently being used and by whom.
* Decide which processes to load when memory space becomes available.
* Allocate and deallocate memory space as needed.

## Secondary Storage Management

The main purpose of a computer system is to execute programs. These programs, together with the data they access, must be in main memory during execution. Since the main memory is too small to permanently accommodate all data and program, the computer system must provide secondary storage to backup main memory.

Most modern computer systems use disks as the principle on-line storage medium, for both programs and data. Most programs, like compilers, assemblers, sort routines, editors, formatters, and so on, are stored on the disk until loaded into memory, and then use the disk as both the source and destination of their processing.

The operating system is responsible for the following activities in connection with disk management:

* Free space management
* Storage allocation

Disk scheduling

## Security Management

The operating system is primarily responsible for all task and activities happen in the computer system. The various processes in an operating system must be protected from each other’s activities. For that purpose, various mechanisms which can be used to ensure that the files, memory segment, cpu and other resources can be operated on only by those processes that have gained proper authorization from the operating system.

Security Management refers to a mechanism for controlling the access of programs, processes, or users to the resources defined by a computer controls to be imposed, together with some means of enforcement.

For example, memory addressing hardware ensure that a process can only execute within its own address space. The timer ensure that no process can gain control of the CPU without relinquishing it. Finally, no process is allowed to do it’s own I/O, to protect the integrity of the various peripheral devices.

## Command Interpreter System

One of the most important component of an operating system is its command interpreter. The command interpreter is the primary interface between the user and the rest of the system.

Command Interpreter System executes a user command by calling one or more number of underlying system programs or system calls.

Command Interpreter System allows human users to interact with the Operating System and provides convenient programming environment to the users.

Many commands are given to the operating system by control statements. A program which reads and interprets control statements is automatically executed. This program is called the shell and few examples are Windows DOS command window, Bash of Unix/Linux or C-Shell of Unix/Linux.

## Other Important Activities

An Operating System is a complex Software System. Apart from the above mentioned components and responsibilities, there are many other activities performed by the Operating System. Few of them are listed below:

**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 softwares and users** − Coordination and assignment of compilers, interpreters, assemblers and other software to the various users of the computer systems.