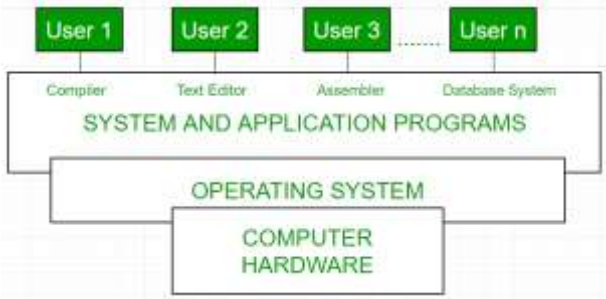
Unit – 01 : Introduction to operating system

- An Introduction to O.S & its services,
- Evolution of O.S,
- Various types of O.S(Batch Processing, Multiprogramming, Multiprocessing, Multitasking).
- Concepts of Process – files – system calls – Interrupt – shell.
- Introduction to Unix, Shell commands.

1. Define operating system. List the objective of an operating system.
2. Discuss in details generation of operating system.
3. What are difference between batch OS and multi-programming?
4. Differentiate between multi-programming OS and multi-tasking.
5. List the difference between mainframe and desktop OS.
6. Define LINUX virtualization.
7. List the component of LINUX kernel modules.
8. Describe about the network structure ok LINUX system in data.
9. Discuss the features and benefits of UNIX operating system in details.
10. Write short notes on :
    a. UNIX Shell
    b. System call
    c. Interrupt in OS

Introduction :

- An operating system is an important part of almost every computer system.
- A computer system can be divided roughly into four components :
  1. Hardware
  2. Operating System
  3. Application Programs
  4. Users



What is an Operating System ?

- An operating system is a system software.
- It works as as interface between user and computer hardware.
- It co-ordinate between different devices.
- If schedule multiple tasks as per priority.
- It manages computer hardware and provide common service to run user programs.
- Every computer have atleast one OS to run other application programs like MS-office, Chrome etc.
- Without OS a computer can't be work.

Functions of Operating System

1. **Booting :** It boots the computer system.
2. **Throughput:** Number of tasks per unit time.
3. **Resource manager :** It performs basic computer tasks & manage the various peripheral devices like mouse, keyboard, monitor, printer etc.
4. **Interface :** It provides a user interface like command line interface(CLI) & graphical user interface (GUI).
5. **Memory management :** It has to decide which process needs memory space and how much. OS also has to allocate and deallocate the memory space.
6. **Security/Privacy Management:** Privacy is also provided by the Operating system by means of passwords so that unauthorized applications can't access programs or data.

Goal of an Operating System :

- The Primary goal of an OS is to make the computer system convenient to use.
- A Secondary goal is to use the computer hardware in an efficient manner.

Generations of Operating System :

- Operating Systems have evolved over the years.
- So, their evolution through the years can be mapped using generations of OS.
- There are four generations of operating systems.

First Generation (1945-1955)

- When the first computer was developed in 1945, it was created without any operating system.
- In this generation there is no operating system, so the computer system is given instructions which must be done directly.

Second Generation (1955 - 1965)

- The first operating system was created in the early 1950s and was known as **GMOS.**
- **General Motors** has developed OS for the **IBM** computer.
- The second-generation operating system was based on a single stream batch processing system because it collects all similar jobs in batches and then submits the jobs to the operating system using a punch card to complete all jobs in a machine.

The Third Generation (1965 - 1980)

- During the late 1960s, operating system designers were very capable of developing a new operating system that could simultaneously perform multiple tasks in a single computer program called multiprogramming.
- UNIX is a powerful Operating System initially developed by Ken Thompson, Dennis Ritchie at AT&T Bell laboratories in 1970.

The Fourth Generation (1980 - Present Day)

- The fourth generation of operating systems is related to the development of the personal computer.
- Microsoft created the first **window** operating system in 1975 & **MS-DOS** in 1981.
- And then, Windows released various operating systems such as Windows 95, Windows 98, Windows XP and the latest operating system, Windows 11.
- Apple is another popular operating system built in the 1980s, and this operating system was developed by Steve Jobs, a co-founder of Apple.
- They named the operating system Macintosh OS or Mac OS.

Types of an Operating System

There are five types of an operating system

1. Batch OS
2. Multi-programming OS
3. Multi-tasking/Time sharing OS
4. Multi-processing OS
5. Real Time OS

Batch Processing :

- Batch processing is a technique in which an OS collects the programs and data together in a batch before processing.
- A series of jobs are executed without any human intervention in Batch processing system.
- It is also called as Simple Batch System.
- It is slower in processing than other operating system.

Multi-programming :

- When two or more programs reside in memory at the same time, it is known as multiprogramming.
- Multiprogramming increases CPU utilization by organizing jobs so that the CPU always in execution.
- It is faster than Batch Processing operating system.

Multi-tasking :

- In multitasking OS multiple jobs are executed by the CPU simultaneously.
- Multitasking Operating Systems are also known as Time-sharing operating system.
- A time-shared operating system uses the concept of CPU scheduling.
- Each user has at least one separate program in memory.

Multiprocessing Systems

- In multiprocessing OS two or more CPUs control the functions of the computer.
- Each CPU contains a copy of the OS, and these copies communicate with one another.
- The use of multiple processors allows the computer to perform calculations faster.
- These operating systems include Windows NT, 2000, XP, and Unix.

Real-Time Operating System (RTOS)

- Real-time operating system (RTOS) is an OS intended to serve real time application.
- It is time-bound system that can be defined as fixed time constraints.
- In this type of system, processing must be done inside the specified time. Otherwise, the system will fail.
- There are two types of RTOS :

Hard Real Time :

- In Hard RTOS, the deadline is handled very strictly which means that given task must start executing on specified scheduled time, and must be completed within the assigned time duration.
  Example: Medical critical care system, Aircraft systems, etc.

Soft Real Time:

- Soft Real time RTOS, accepts some delays by the Operating system.
  Example: Online Transaction, online application & Various types of Multimedia applications.

Difference between Multiprogramming and Multi-processing OS :

Multiprogramming	Multi-tasking
In multiprogramming, multiple programs are ready to access CPU.	In Multitasking, a single CPU is used to process multiple tasks.
The process resides in the main memory.	The process resides in the same CPU.
Based on the context switching mechanism.	Based on the time-sharing mechanism.
It is useful for reducing CPU idle time and increasing throughput as much as possible.	Running multiple processes at the same time so, increasing CPU utilization & throughput.
It takes more time to executes the process.	It takes less time to executes the process.

Difference between Multi-tasking and Multi-processing OS :

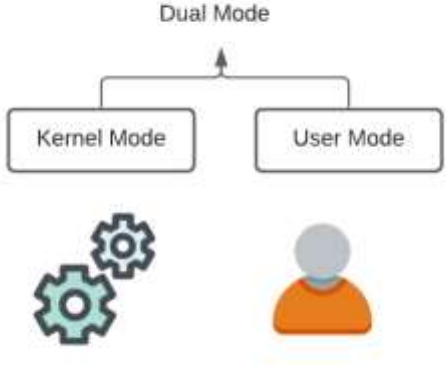
Multi-tasking	Multiprocessing
The execution of more than one task simultaneously is known as multitasking.	The availability of more than one processor per system is known as multiprocessing.
The number of CPU is one.	The number of CPUs is more than one.
It takes moderate amount of time.	It takes less time for job processing.
In this, one by one job is being executed at a time.	Here, more than one process can be executed at a time.
Throughput is moderate.	Throughput is maximum.
Its efficiency is moderate.	Its efficiency is maximum.
It is of two types: Single user multitasking and Multiple user multitasking.	It is of two types: Symmetric Multiprocessing and Asymmetric Multiprocessing.

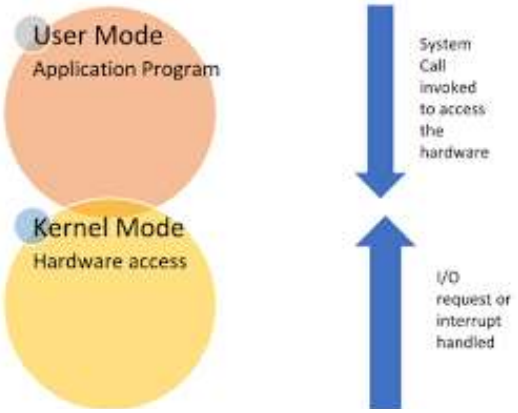
What is UNIX OS ?

- UNIX is the most popular operating system.
- It is a multi-user OS , which means that more than one person can work on the same computer at the same time .
- UNIX also supports multi-tasking.
- The first version of UNIX was introduced during the 1970s.
- It was designed by Ken Thompson at AT & T Bell Laboratories & its code is written by Denis Ritche.
- The source code of UNIX OS was written in C language.
- Unix is portable, that is fit for any hardware & used everywhere.

Dual-Mode/Multi-Mode Operation in Operating System

- There are two modes of operation in the operating system to make sure it works correctly. These are user mode and kernel mode.





User Mode :

- The system is in user mode when the operating system is running a user application such as handling a text editor.
- The transition from user mode to kernel mode occurs when the application requests the help of operating system or an interrupt or a system call occurs.
- The mode bit is set to 1 in the user mode. It is changed from 1 to 0 when switching from user mode to kernel mode.

Kernel Mode :

- The system starts in kernel mode when it boots and after the operating system is loaded, it executes applications in user mode.
- The mode bit is set to 0 in the kernel mode. It is changed from 0 to 1 when switching from kernel mode to user mode.

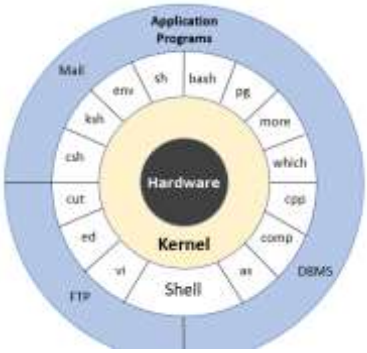
What is shell in Unix OS ?

- A shell is a program whose primary purpose is to read commands and run other programs.
- The shell is the outermost layer of the operating system.
- The shell manages the interaction between user and the operating system.
- This communication is carried out either interactively (input from the keyboard) or as a shell script.

Unix Shell Commands

There are three types of Unix Shell Commands :

1. Basic.
2. Intermediate.
3. Advanced.



Basic Unix Shell Commands

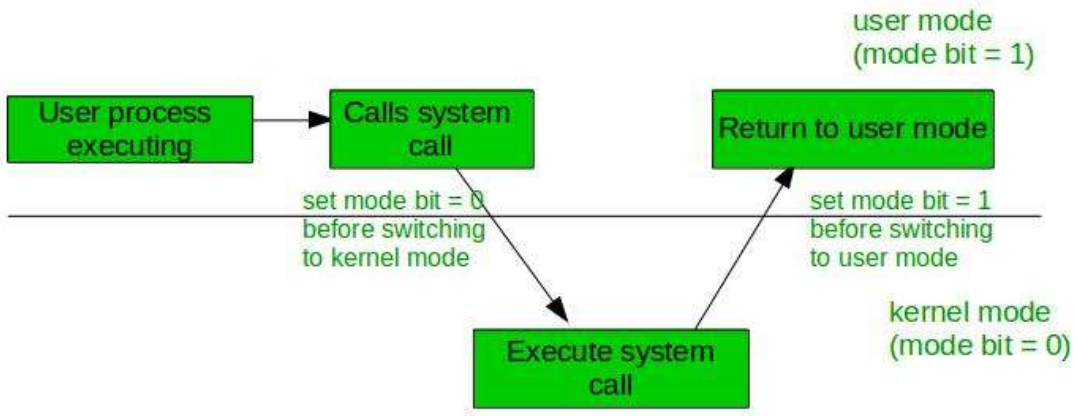
- **Listing files (ls) –** 'ls' command is used to list out all the files in a directory.
- **Creating & Viewing Files –** 'cat command can be used to create the file or view the contents of the file.
- **Deleting Files –** 'rm' command is used to delete a file from the directory.
- **Moving and Re-naming files –** 'mv' command is used for moving a file from one location to another. This command can also be used for renaming the file as the source file gets deleted and a new file gets created.
- **Making directories –** 'mkdir' Unix also provides us with the command to make our own directory. It is just like making our own folder where all relevant files can be stored.

What is Interrupt in OS?

- An interrupt is an event that alters the sequence of processor execution.
- It is a signal emitted by hardware or software when a process needs immediate attention.
- In I/O devices, one of the bus control lines is dedicated for this purpose and is called the Interrupt Service Routine (ISR).
- There are two types of interrupts :
  - ❖ Hardware interrupts
  - ❖ Software interrupts.

System call :

- A system call is a method to request a service from the kernel of the operating system on which it is running.
- The interface between a process and an operating system is provided by system calls.
- In general, system calls are available in assembly language or a high-level language like **C** or **Pascal**.
- 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.



Types of System Calls :

There are mainly five types of system calls. These are explained in detail as follows –

1. **Process Control :** These system calls deal with processes such as process creation, process termination etc.
2. **File Management :** These system calls are responsible for file manipulation such as creating a file, reading a file, writing into a file etc.
3. **Device Management :** These system calls are responsible for device manipulation such as reading from device buffers, writing into device buffers etc.
4. **Information Maintenance :** These system calls handle information and its transfer between the operating system and the user program.
5. **Communication :** These system calls are useful for interprocess communication. They also deal with creating and deleting a communication connection.

Types of System Calls	Windows	Linux
Process Control	CreateProcess() ExitProcess() WaitForSingleObject()	fork() exit() wait()

Unit – 02 : Process Management

- An Introduction to process,
- process state & Transition,
- Process control Block,
- Process Context,
- Context switch.

Questions to be discussed :

1. What is process ?
2. Drow a process transition diagram and explain the various states.
3. What is process control block ?
4. Define process context and context switching.

What is a Process?

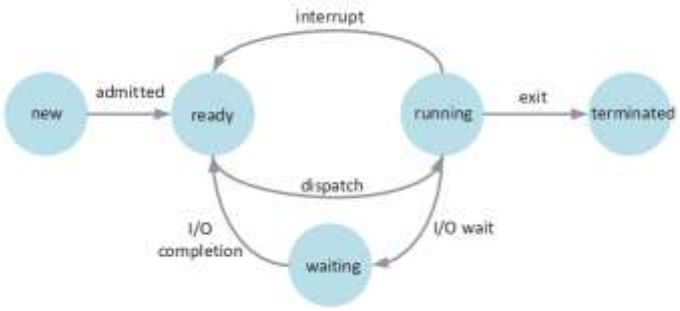
- A process is a program in execution.
- A process is an instance of a program.
- In other words a running program is known as process.
- A process is an 'active' entity.
- Process has a high resource requirement, it needs resources like CPU, memory address, I/O during its lifetime.

Difference between Program and Process :

Program	Process
A program is a set of instructions designed to complete a specific task.	Process is an instance of an executing program.
Program is a passive entity.	Process is a active entity.
It resides in the secondary memory.	It is created during execution and loaded into the main memory.
Program exists at a single place and continues to exist until it is deleted.	Process exists for a limited span of time as it gets terminated after the completion of task.
Program is a static entity.	Process is a dynamic entity.
Program does not have any resource requirement, it only requires memory space for storing the instructions.	Process has a high resource requirement, it needs resources like CPU, memory address, I/O during its lifetime.

Different Process States/Process transition diagram :

- Process transition diagram represents the present activity of the program when it is running on OS.
- Processes in the operating system can be in any of the following states :



- **NEW -** The process is being created.
- **READY -** The process is waiting to be assigned to a processor.
- **RUNNING -** Instructions are being executed.
- **WAITING -** The process is waiting for some event to occur(such as an I/O request).
- **TERMINATED -** The process has finished execution.

What is Scheduler ?

- It is a system software which handles the process scheduling in various ways.
- Their main task is to select the jobs to be submitted into the system and decide which process to run.
- There are three types of process scheduler :
  1. Long-Term Scheduler
  2. Short-Term Scheduler
  3. Medium-Term Scheduler

Long Term or job scheduler :

- It create the new process & submit to the 'Ready State'.
- It controls Degree of Multi-programming,
- It is also called a job scheduler.
- Long-Term Scheduler changes the process state from **New** to **Ready**.
- Speed is less than the short-term scheduler.

Short term or CPU scheduler :

- It is responsible for selecting process from ready state for scheduling it on the running state.
- Short-term scheduler only selects the process to schedule it doesn't load the process on running.
- **Short-Term Scheduler** is also known as **CPU Scheduler.**
- Speed is very fast as compared to long-term scheduler.
- Here is when all the scheduling algorithms are used.

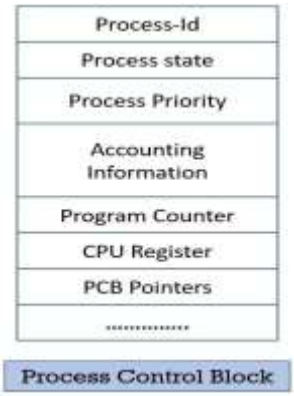
Medium-term scheduler :

- It is responsible for suspending and resuming the process.
- It mainly does swapping (moving processes from main memory to disk and vice versa).
- It is helpful in maintaining a perfect balance between the I/O bound and the CPU bound.
- It also reduces the degree of multiprogramming.

Comparision between Schedulers :

Long-Term Scheduler	Short-Term Scheduler	Medium-Term Scheduler
It is a job scheduler	It is a CPU scheduler	It is a process swapping scheduler.
Speed is lesser than short term scheduler	Speed is fastest among other two	Speed is in between both short and long term scheduler.
It controls the degree of multiprogramming	It provides lesser control over degree of multiprogramming	It reduces the degree of multiprogramming.
It selects processes from pool and loads them into memory for execution	It selects those processes which are ready to execute	It can re-introduce the process into memory and execution can be continued.

Process Control Block(PCB) :



- A PCB is a data structure used by OS to store all process related information.
- The OS creates a process control block when a process is created.
- Each process is represented in OS by a Process Control Block.
- It is also called Task control block.
- PCB contains all the information about the each process.
- It is a data structure, that contains the following information about the process.

Context Switching :

- A context switching is the mechanism to store and restore the state of process.
- When interrupt occurs OS needs to change the current task after saving current state of process so that it can be resumed later.
- The atate of process saves Process Control Block(PCB).
- The whole mechanism of context switch is done by scheduler.
- Context switching is an essential part of a multitasking operating system.

Process Context

- Process switching is a type of context switching where we switch one process with another process.
- It involves switching of all the process resources with those needed by a new process.
- Each time a process is removed from the processor then sufficient information of its current operating state must be stored such that when it is again scheduled to run on the processor.
- This operational state of process is known as its context and the act of removing the process from the processor is known as a process context or process switch.

Unit – 03 : Process Schedulling

- Process Scheduling : (Pre-emptive & Non-pre-emptive Algorithms),
  - FCFS (First Come First Served) Algorithm,
  - SJF (Shortest Job First),
  - Priority Scheduling,
  - Round Robin Scheduling.
- Performance criteria of scheduling Algorithm,
- CPU utilization, throughput, Turnaround time, waiting time, Response time.
- Overview of Inter-process communication
- Race condition
- critical section,
- Semaphores.

Questions to be discussed :

1. Define process scheduling. What is the goal of process scheduling.
2. Explain different types of schedulling algorithm in brief.
3. What is preemptive and non-preemtive scheduling ?
4. Explain FCFS scheduling algorithm in brief.
5. Explain Turn Around Time(TAT), Waiting Time(WT) and response time(RT).
6. Assume the following jobs have arrive in the order 1,2,3,4 and 5.

Job	Arrival Tiime	Brust Time	Priority
1	0	15	2
2	2	3	1
3	5	5	5
4	6	8	4
5	7	12	3

Draw a Gantte-chart and calculate Turn Around Time and average Waiting Time for

a) FCFS

b) Preemtive priority algorithm.

7. Assume the following jobs have arrive in the order 1,2,3,4 and 5.

Process	P1	P2	P3	P4	P5
Brust Time	7	9	2	1	3
Priority	3	2	1	4	5

   a. Draw a Gantte -Chart illustrating the execution of there processes using FCFS, SJF, preemptive priority and RR(quantum=1) schedulling.
   b. Calculate the waiting time and Turn around time for each process.
8. Define inter-process communication in brief with examples.
9. Write short notes on the following :
   a. Race condition
   b. Critical section
   c. Semaphores

What is Process Scheduling ?

- The act of determining which process is in the ready state, and should be moved to the running state is known as Process Scheduling.
- A Process Scheduler schedules different processes to be assigned to the CPU based on particular scheduling algorithms.
- There are some popular process scheduling algorithms which are given below :
  1) First-Come, First-Served (FCFS) Scheduling
  2) Shortest-Job-Next (SJN) Scheduling
  3) Priority Scheduling
  4) Round Robin(RR) Scheduling

Goal of process schedulling :

- Maximize CPU utilization
- Maximum throughput
- Minimize waiting time
- CPU scheduling should be fair

What is preemptive and non-preemtive scheduling.

Non Pre-emptive Scheduling :

- Non-preemptive algorithms are designed so that once a process enters the running state, it cannot be preempted until it completes its allotted time.

Pre-emptive Scheduling :

- Preemptive scheduling is based on priority where a scheduler may preempt a low priority running process anytime when a high priority process enters into a ready state.

Process Time :

- Arrival Time
- Burst Time/Service Time
- Turn Around Time,
- Completion Time
- Waiting Time and response time :

Turn Around Time, Waiting Time and response time :

Turn around time (TAT)

- Turn around time (TAT) is the time interval from the time of submission of a process to the time of the completion of the process.
- It can also be considered as the sum of the time periods spent waiting to get into memory or ready queue, execution on CPU and executing input/output.
- Turn Around Time(T.A.T) is the Time Difference between completion time and arrival time.

Turn Around Time = Completion Time - Arrival Time

TAT = C.T – A.T

Waiting time(WT)

- Waiting time is the total time spent by the process in the ready state waiting for CPU.
- Waiting Time(W.T) is the Time Difference between turn around time and burst time.

Waiting Time = Turnaround time - Burst Time

i.e; **WT = TAT – B.T**

Response time(RT)

- Response time is the time spent when the process is in the ready state and gets the CPU for the first time.

Important :

- **Arrival Time :** Time between new to ready queue.
- **Completion Time(C.T) :** Time at which process complete its execution.
- **Burst Time :** Time required by a process for CPU execution.
- **Turn Around Time(T.A.T) :** It is the time difference between completion time and arrival time.

  Turn Around Time = Completion Time – Arrival Time
- **Waiting Time(W.T):** Time Difference between turn around time and burst time.

  Waiting Time = Turn Around Time – Burst Time
- **Throughput** : Number of processes completed per unit time is called throughput.

FCFS scheduling algorithm.

- First come first serve scheduling algorithm based on **arrival time**.
- This algorithm is **non-preemptive**.
- The job which comes first in the ready queue will get the CPU first.
- FCFS scheduling may cause the problem of starvation.

Advantages of FCFS

- Simple
- Easy

Disadvantages of FCFS

- Due to the non-preemptive nature, the problem of starvation may occur.
- It is easy to implement, but it is poor in performance.

Shortest Job First (SJF) Scheduling

- SJF scheduling algorithm based on **burst time**.
- This algorithm is **non-preemptive**.
- In SJF, the process with the lowest burst time is going to be scheduled next.
- This algorithm is very difficult to implement in the system.

Advantages of SJF

1. Maximum throughput
2. Minimum average waiting and turnaround time

Disadvantages of SJF

1. May suffer with the problem of starvation
2. It is not implementable.

Round Robin Scheduling Algorithm

- Round Robin scheduling algorithm is based on **time quantum**.
- This is the **preemptive version** of first come first serve scheduling.
- This algorithm is designed for **Time Sharing** system.
- A certain time slice is defined in the system which is called time **quantum**.
- Each process present in the ready queue is assigned the CPU for that time quantum, if the execution of the process is completed during that time then the process will **terminate** else the process will go back to the **ready queue** and waits for the next turn to complete the execution.

Priority Scheduling

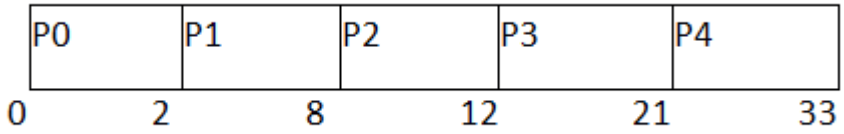
- Priority scheduling algorithm based on **priority** of process.
- In Priority scheduling, there is a priority number assigned to each process.
- In some systems, the lower the number, the higher the priority while, in the others, the higher the number, the higher will be the priority.
- There are two types of priority scheduling algorithm exists.
  1) **Preemptive** priority scheduling
  2) **Non Preemptive** Priority scheduling.

Q. Find the waiting time of given process using FCFS scheduling algorithm.

Process No	Arrival Time(A.T)	Burst Time(B.T)
0	0	2
1	1	6
2	2	4
3	3	9
4	6	12

Solution :

Gantte chart :



P. No	A.T	B.T	C.T	TAT	WT
0	0	2	2	2	0
1	1	6	8	7	1
2	2	4	12	10	6
3	3	9	21	18	9
4	6	12	33	27	15

$$\text{Avg. waiting time} = \frac{0+1+6+9+15}{5} = \frac{31}{5} = 6.2$$

Describe in brief independent and co-operative process.

There are two types of process in operating system :

1. **Independent**
2. **Cooperating.**

Independent Processes

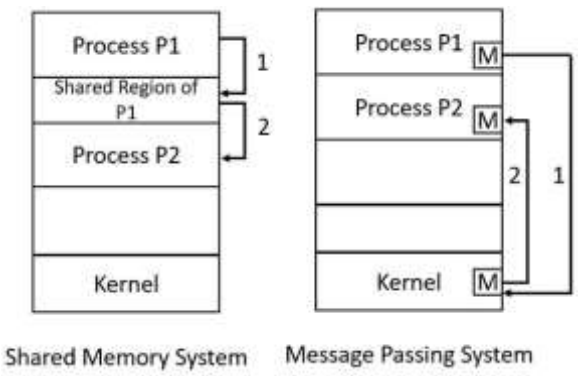
- A process is independent if it can't affect or be affected by other processes executing in the system.
- These process does not share any data with any process.

Cooperating Processes

- A process is cooperating if it can affect or be affected by other processes executing in the system.
- These processes share their data with other process.
- Cooperating processes require an IPC mechanism that will allow them to exchange data/information.

Inter Process Communication (IPC)

- Inter Process Communication (IPC) is a mechanism, where the operating systems allow various processes to communicate with each other.
- The communication between these processes can be seen as a method of co-operation between them.
- Processes can communicate with each other through two ways :
  1. Shared Memory
  2. Message passing



What is a race condition?

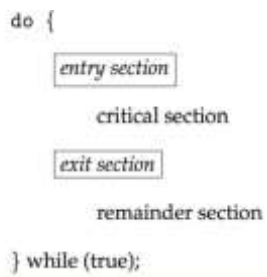
- A race condition is an undesirable situation that occurs when two processes attempt to access the same resource at the same time.
- A race condition is a situation that may occur inside a critical section.
- Race conditions in critical sections can be avoided if the critical section is treated as an atomic instruction.

Critical Section(CS):

- A CS is a segment of code which can be accessed by a signal process at a specific point of time.
- This section consists of shared data resources that required to be accessed by other processes.
- The entry to the critical section is handled by the wait() function.
- The exit from a critical section is controlled by the signal() function.
- In the critical section, only a single process can be executed. Other processes, waiting to execute their critical section, need to wait until the current process completes its execution.

Mutual Exclusion :

- Mutual Exclusion is a special type of binary semaphore which is used for controlling access to the shared resource.
- Not more than one process can execute in its critical section at one time.



Solutions To The Critical Section

- In Process Synchronization, critical section plays the main role so that the problem must be solved.
- Here are some widely used methods to solve the critical section problem.
  - Peterson Solution
  - Synchronization Hardware
  - Mutex Locks
  - Semaphore Solution

Semaphores in Process Synchronization

- A semaphore S is an integer variable that is used to solve critical section problem.
- It is only accessed through two standard atomic operation wait(P) & signal(V).
- It was proposed by Dijkstra in 1965.
- It is a very significant technique to manage concurrent processes by using a simple integer value, which is known as a semaphore.

Semaphores are of two types :

Binary Semaphore :

- This is also known as mutex lock.
- It can have only two values – 0 and 1. Its value is initialized to 1.
- It is used to implement the solution of critical section problem with multiple processes.

Counting Semaphore :

- Its value can range over an unrestricted domain.
- It is used to control access to a resource that has multiple instances.

Unit – 04 : Deadlock

- Introduction to Deadlock,
- Necessary condition for Dead-lock,
- Method for Handling Deadlock,
- Brief overview of Deadlock prevention,
- Deadlock Avoidance (Banker's Algorithm),
- Deadlock Detection & Recovery.

Questions to be discussed :

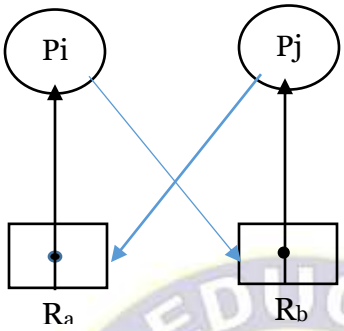
1. What is deadlock ? How to deadlock are detected ?
2. What are the necessary condition for deadlock describe in brief.
3. What is starvation ? Explain in brief method for handling deadlock.
4. How to prevent deadlock explain in details ?
5. Write short notes on :
   a. Banker's algorithm
   b. Deadlock detection & recovery
6. Considering a system with five processes P_0 through P_4 and three resources of type A, B, C. Resource type A has 10 instances, B has 5 instances and type C has 7 instances. Suppose at time t_0 following snapshot of the system has been taken :

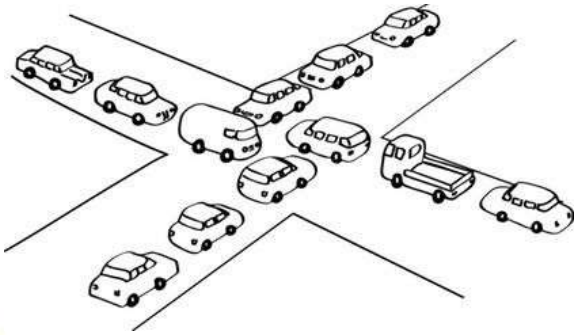
Process	Allocation	Max	Available
	A B C	A B C	A B C
P_0	0 1 0	7 5 3	3 3 2
P_1	2 0 0	3 2 2	
P_2	3 0 2	9 0 2	
P_3	2 1 1	2 2 2	
P_4	0 0 2	4 3 3	

Is the system in a safe state? If Yes, then what is the safe sequence?

What is Deadlock ?

- Two or more processes are said to be in deadlock if and only if they wait for happening of an events which would never happen.
- Deadlock is a situation where a set of processes are blocked because each process is holding a resource and waiting for another resource acquired by some other process.
- In other words, we can say that deadlock is infinite waiting.
- Deadlock happens when Mutual exclusion, hold and wait, No preemption and circular wait occurs simultaneously.



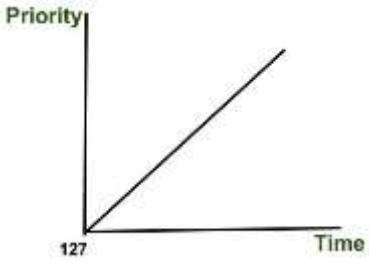


What is Starvation ?

- Starvation is the long time waiting.
- It is the problem that occurs when high priority processes keep executing and low priority processes get blocked for indefinite time.
- In starvation resources are continuously utilized by high priority processes.
- Problem of starvation can be resolved using Aging.
- In Aging priority of long waiting processes is gradually increased.

What is Aging ?

- Aging is a scheduling technique used to avoid starvation.
- It is a technique of gradually increasing the priority of processes that wait in the system for a long time.



Difference between deadlock & starvation.

Deadlock	Starvation
Deadlock is infinite waiting.	Starvation is long time waiting.
It is a situation in which more than one process is blocked.	It is a process in which the low priority processes are postponed.
Resources are blocked by a set of processes in a circular fashion.	Resources are continuously used by high-priority process.
It is prevented by avoiding anyone necessary condition required for a deadlock.	It can be prevented by aging.
In a deadlock, none of the processes get executed.	In starvation, higher priority processes execute while lower priority processes are postponed.
Deadlock is also called circular wait.	Starvation is also called lived lock.

What are the necessary condition for occuring deadlocks.

A deadlock condition can arise if the following four conditions hold simulteniously in a system :

Mutual Exclusion

- ME states that only one process can be inside the critical section at any time.
- If any other processes require the critical section, they must wait until it is free.

Hold and Wait

- There must exist a process that is holding at least one resource and is waiting to acquire additional resources that are held by other process.

No Preemption

- Resource cannot be preempted; that is a resource can be released by the process, after that process has completed its task.

Circular Wait

- There must exist a set of waiting processes such that P_0 is waiting for a resource that is held by P_1, P_1 is waiting for a resource that is held by P_2, ... P_{n-1} is waiting for a resource that is held by P_n and P_n is waiting for a resource that is held by P_0.

Method for handling deadlock :

There are three different methods to handling deadlock problems :

Handling Deadlock

1. Prevention
2. Avoidance (Banker's Algorithm)

2. Detection & Recovery

4. Ignorancce (Ostrich algorithm)

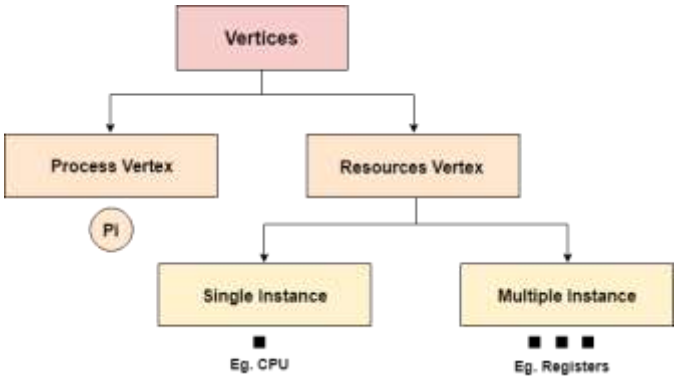
Note : To ensure that deadlocks never occur, the system can either a deadlock prevention or a deadlock avoidance scheme.

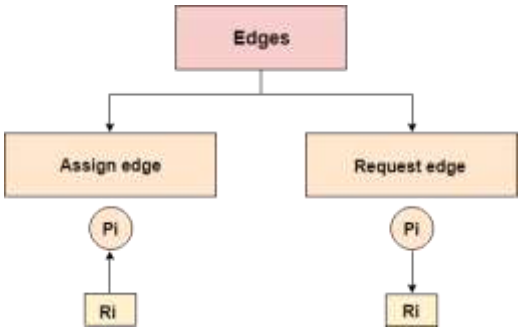
Deadlock Prevention :

- As we know that, for occurring deadlock, each of the four necessary conditions must hold.
  - Mutual exclusion
  - Hold & Wait
  - No Preemption
  - Circular Wait
- By ensuring that atleast one of these conditions cannot hold, then we can prevent the occurrence of deadlock.
- Prevention is the set of methods for ensuring that atleast one of the necessary conditions cannot hold.

Resource Allocation Graph :

- It is the pictorial representation of the state of the system.
- It describe the complete information about all the processes which are holding some resources or waiting for some resources.
- It also contains the information about all the resources whether they are available or being used by the processes.
- In Resource allocation graph, the process is represented by a Circle while the Resource is represented by a rectangle.





Safe state :

- A system is said to be safe if the needs of all the processes should be satisfy with the available resources in the same order.
- The order in which the needs of the process are satisfy is called safe sequence.
- There may be a multiple safe sequence for a given state of the system at a time.

Unsafe States ;

- A system is said to be unsafe if the needs of all the processes should not be satisfy with the available resources.

Deadlock Avoidance

- In deadlock avoidance, the request for any resource will be granted if the resulting state of the system doesn't cause deadlock.
- The state of the system will continuously be checked for safe and unsafe states.
- It requires prior information about the needs & total resources.
- Deadlock avoidance can be done with Banker's Algorithm.
- This algorithm examines the resource allocations so that there can never be a circular wait condition.

Banker's Algorithm :

- Bankers's Algorithm is used in deadlock avoidance.
- The name was choosen because this algorithm could be used in banking system to ensure that the bank never allocates its available cash such that it can no longer satisfy the needs of all its customers.
- It checks for the safe state :
  - If after granting request system remains in the safe state then it allows the request.
  - If after granting request the system is unsafe state then it doesn't allow the request.

Let n be the number of process in the system and m be the number of resource type then we need the following data structure :

Available = Total - Allocation

Need = Maximum - Allocation

Example :

Concider a five processes has 30 resources. Using Banker's algorithm check whether the system is in safe state, if in safe state then find the safe sequence. Suppose at a given time the system has given scenario :

Process	Maximum	Allocation	Need	Available
P_1	10	4		
P_2	5	2		
P_3	15	7		
P_4	20	10		
P_5	8	4		

Safe Sequence : < P_2 P_5 P_1 P_3 P_4>

Example :

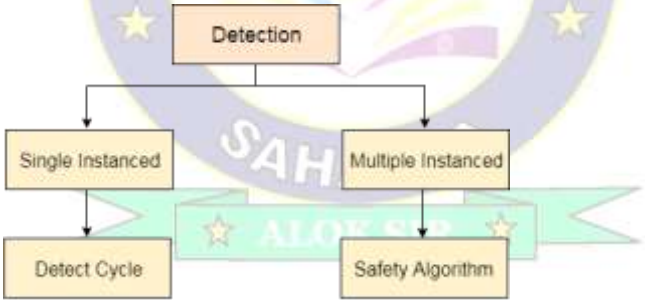
Considering a system with five processes P_0 through P_4 and three resources of type A, B, C. Resource type A has 10 instances, B has 5 instances and type C has 7 instances. Suppose at time t_0 following snapshot of the system has been taken:

Process	Allocation	Max	Available
	A B C	A B C	A B C
P_0	0 1 0	7 5 3	3 3 2
P_1	2 0 0	3 2 2	
P_2	3 0 2	9 0 2	
P_3	2 1 1	2 2 2	
P_4	0 0 2	4 3 3	

Safe Sequence : $< P_1\ P_3\ P_4\ P_0\ P_2 >$

Deadlock Detection and Recovery :

- In this approach, The OS doesn't apply any mechanism to avoid or prevent the deadlocks.
- Therefore the system considers that the deadlock will definitely occur.
- The OS periodically checks the system for any deadlock.
- In case, it detects the deadlock then the OS will recover using some recovery techniques.
- In order to recover the system from deadlocks, either OS considers resources or processes.



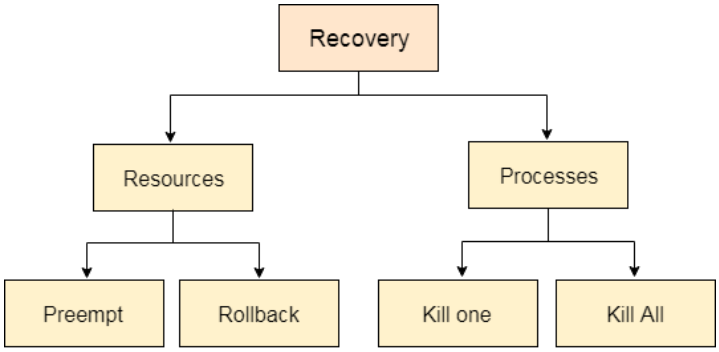
- In single instance resource types, if a cycle is being formed in the system then there will definitely be a deadlock.
- In multiple instances resource type graph, detecting a cycle is not enough.
- We have to apply the safety algorithm on the system by converting the resource allocation graph into the allocation matrix and request matrix.

For Resource

- Preempt the resource
- Rollback to a safe state

For Process

- Kill a process
- Kill all process



Unit-05 : Memory Management

- Introduction to Memory management,
- Contiguous allocation,
  - Fixed Partition,
  - Dynamic partition,
- Non-contiguous allocation –
  - paging,
  - segmentation.
- Introduction to Virtual-memory management –
- Demand paging,
- Cop-on-write,
- page replacement,
- Allocation of frames,
- Thrashing.

Questions to be discussed :

1. Describe memory management. How memory management is required ?
2. Compare between internal and external fragmentation.
3. What are the difference between static & dynamic memory allocation ?
4. Define logical address and physical address.
5. Differentiate between paging and segmentation ?
6. What are the difference between paging and segmentation ?
7. What is virtual memory ? Discuss the benefits of virtual memory technique in details.
8. What is demand paging ? Describe the process of how can demand paging can be implemented with virtual memory.
9. What is page faults ? Explain in detail the step involve in handling a page fault.
10. What is thrashing ? What is the cause of thrashing ? Also discuss how does the system detect thrashing.
11. Explain different types of page replacement policy.
    a) First In First Out (FIFO)
    b) Least Recently Used (LRU)
    c) Optimal Page Replacement
12. Concider the reference string :

    7, 0, 1, 2, 0, 3, 0,4, 2, 3, 0, 3, 2, 1, 2, 0, 1, 7, 0, 1 for a memory with three frames. Trace FIFO, Optimal and LRU page replacement algorithms.
13. Write short notes on following :
    a) Cop-on write
    b) Demand paging

What is memory management ?

- The task of dividing the memory among different processes is called memory management.
- It is the function responsible for allocating and managing main memory of computer.
- It keeps track of each and every memory location, either it is allocated to some process or free.
- It checks how much memory is to be allocated to processes.
- It decides which process will get memory at what time.
- It tracks whenever some memory gets freed or unallocated then it updates the status.
- The main aim of memory management is to achieve efficient utilization of memory.

Why Memory Management is required ?

- To minimize fragmentation issues.
- To proper utilization of main memory.
- To keep track of used memory space by processes.
- To maintain data integrity while executing of process.
- Allocate and de-allocate memory before and after process execution.

Function of memory management :

- Allocation
- Protection
- Free space management
- Deallocation

Goals :

- Effective utilization of memory space.
- Run larger program in smaller memory.

There are two types of memory management techniques :

1. Contiguous(process stored in continuous manner)
2. Non-contiguous(process stored at different-different location)

Memory management techniques

Contiguous

- Partitioning
- Overlays

Non-contiguous

- Paging
- Segmentation
- Vertual memory

Difference between Contiguous & Noncontiguous Memory Allocation :

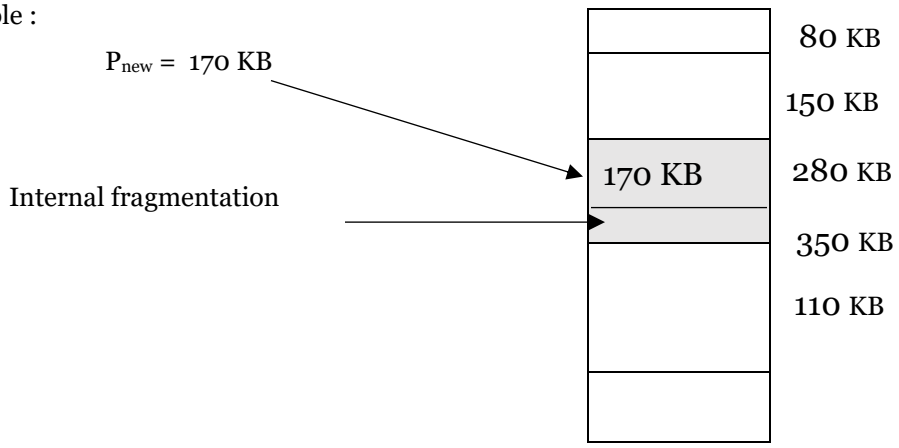
Contiguous Memory Allocation	Non-Contiguous Memory Allocation
It allocates consecutive blocks of memory.	It allocates non-consecutive blocks of memory.
Memory space present at same location.	Memory space present at different location.
Faster in Execution.	Slower in Execution.
It is easier for the OS to control.	It is difficult for the OS to control.
Both Internal fragmentation and external fragmentation occurs.	Only External fragmentation occurs.

What is fragmentation ?

- Wastage of memory space within the computer is called fragmentation.
- There are two types of fragmentation :
  1. Internal fragmentation
  2. External fragmentation

Internal fragmentation :

- It happens when the process assigned is smaller than the memory.
- It is the difference between allocated memory to a process and required memory.
- If the memory block assigned to process is bigger then some portion of memory is left unused and it cannot be used by another process.
- The wastage of memory in this situation is known as internal fragmentation.
- Example :



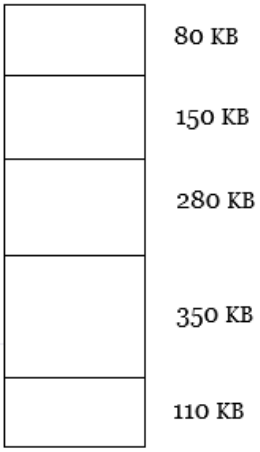
External fragmentation :

- If total memory space is enough to satisfy a request of a process , but we can't accommodate it into the memory because it is not contiguous, so it cannot be used.
- The wastage of memory in this situation is known as external fragmentation.

Example :

P_{new} = 360 KB

Here, external fragmentation occurs because,
A new process of size 360 KB can not able to
Accommodate while total 870 KB memory size
is available.



Compare between internal and external fragmentation.

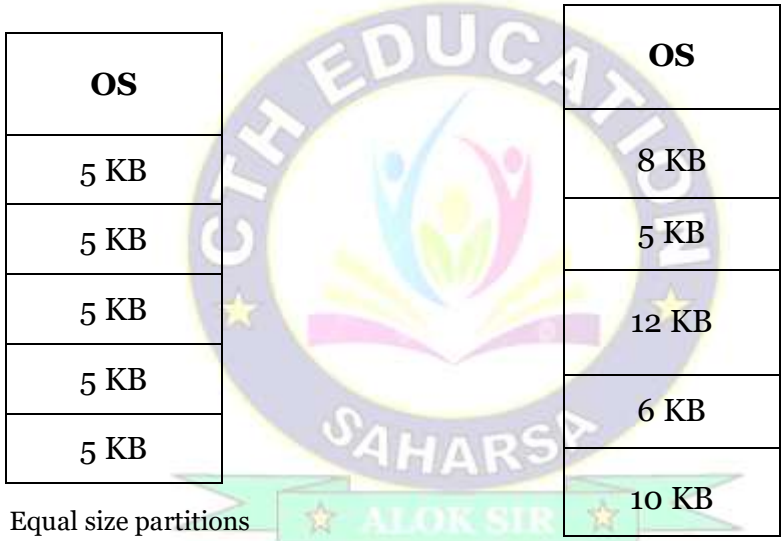
Internal fragmentation(IF)	External fragmentation(EF)
The difference between allocated memory and required memory is called IF.	The unused spaces are too small to accommodate a new process, is called EF.
It happens when the process is smaller than the memory.	It happens when the process is not satisfy any big enough free memory to accommodate it.
It occurs with fixed partitioning.	It occurs with dynamic partitioning.
It occurs in worst fit memory allocation method.	It occurs in best fit and first fit memory allocation method.
The solution of internal fragmentation is best-fit allocation algorithm.	Solution of external fragmentation is compaction.

There are two types of contiguous memory allocation technique :

1. Fixed (static) partitioning
2. Variable (dynamic) partitioning

Fixed partitioning :

- It is a contiguous memory management technique.
- In this method the number of partitions are fixed.
- Size of partition may or may not be same.
- It is also known as static partitioning.
- It is the oldest and simplest technique.
- It is a contiguous allocation so, spanning is not allowed.
- Here partitions are made before execution or during system configure.



Equal size partitions
No. of partitions = fixed

Unequal size partitions
No. of partitions = fixed

Advantages of Fixed Partitioning :

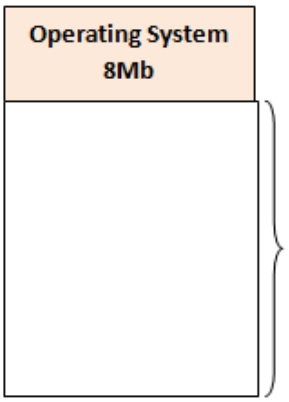
- Easy to implement

Disadvantages of Fixed Partitioning :

- Internal Fragmentation
- External Fragmentation
- Limit process size
- Limitation on Degree of Multiprogramming:

Variable Partitioning :

- It is a contiguous memory management technique in which the main memory is not divided into partitio ns before coming process in memory.
- In this method the memory is allocated to the process according to their need.
- It is also known as dynamic partitioning.
- Here, internal fragmentation is not occurs but it suffer from external fragmentation.
- It also provides the concept of compaction to remove external fragmentation.
- In compaction the spaces that are free and the spaces which not allocated to the process are combined and single large memory space is made.



Advantages of variable partitioning :

- No Internal Fragmentation
- No restriction on Degree of Multiprogramming
- No Limitation on the size of the process

Disadvantages of variable vartitioning :

- Difficult Implementation
- External Fragmentation

Difference between fixed & variable partitioning :

Fixed partitioning	Variable partitioning
Memory is divided into fixed sized partitions.	Memory is not divided into fixed sized partitions.
It is also known as static partitioning.	It is also known as dynamic partitioning.
Only one process can be placed in partition.	Memory is allocated according to need of process.
It does not utilize main memory effectively.	It utilizes the main memory effectively.
Internal and external fragmentation occurs.	Only external fragmentation occurs.
Limitation on degree of multi-programming.	No limitation on degree of multiprogramming.
It is more easier to implement.	It is difficult to implement.
There is limitation on size of process.	There is no limitation on size of process.

Partition allocation policies :

- There are different placement/aloocation Algorithm :
  1. First Fit
  2. Best Fit
  3. Worst Fit
  4. Next Fit

First Fit :

- Allocate the process in the first free big-enough of the partition.
- It scans memory from the beginning and choose the first available block that is big enough.

Example :

P_{new} =100 KB

80 KB
150 KB
280 KB
350 KB
110 KB

Best Fit :

- Allocate the process to the partition which is the first smallest sufficient partition among the free available partition.
- It searches the entire list of holes to find the smallest hole whose size is greater than or equal to the size of the process.

Example :

P_{new} =100 k

80 k
150 k
280 k
350 k
110 k

Worst Fit :

- Allocate the process to the partition which is the largest sufficient among the freely available partitions available in the main memory.
- It searches the entire list of holes to find the largest hole and allocate it to process.
- It is opposite to the best-fit algorithm.

Example :

P_{new} =100 k

80 k
150 k
280 k
350 k
110 k

Next Fit :

- Next fit is similar to the first fit but it will search for the first sufficient partition from the last allocation point.

Example :

P_{new} =100 k

80 k
150 k
280 k(Used)
350 k
110 k

What is Overlays ?

- The main problem in fixed partitioning is the size of a process has to be limited.
- To solve this problem, earlier people have used some solution which is called as Overlays.
- So overlay is a technique to run a larger program in smaller size of memory by keeping only those instructions and data that are needed at any given time.
- Divide the program into modules in such a way that not all modules need to be in the memory at the same time.
- The concept of overlays is that whenever a process is running it will not use the complete program at the same time, it will use only some part of it.
- Then overlays concept says that whatever part you required, you load it and once the part is done, then you just unload it.

What is address space ?

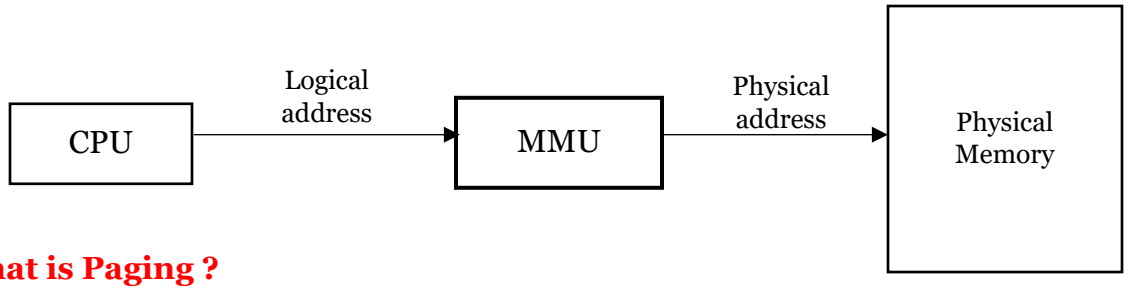
- Each & every instruction or data are stored at a particular address in memory.
- The set of all addresses is known as the address space.
- The address space is the set of addresses generated by programs.
- There are two types of address space :
    1. Logical address space
    2. Physical address space

Difference between logical & physical address :

LOGICAL ADDRESS	PHYSICAL ADDRESS
It is generated by CPU.	Actual location in main memory unit.
User can access the logical address.	User can never access physical address.
The user can use the logical address to access the physical address.	The user can indirectly access physical address but not directly.
Logical address can be change.	Physical address will not change.
Generated by the CPU	Computed by MMU
It is also known as virtual address.	It is also known as real address.
Logical Address Space is set of all logical addresses generated by CPU.	Physical Address is set of all physical addresses mapped to the corresponding logical addresses.

Memory Management Unit(MMU) :

- MMU is a hardware device.
- It convert the logical address into the physical address during the program execution.
- It is located within the Central Processing Unit.
- A memory management unit is also known as a paged memory management unit.



What is Paging ?

- Paging is the concept of non-contiguous memory management technique.
- The main purpose of paging is to divide each process in equal size pages & load them into the main memory.
- In paging process address space is broken into blocks of the same size pages.
- The main memory will also be divided in the form of frames.
- In this mechanism the page size and frame size must be same.

Advantages and Disadvantages of Paging :

- Paging reduces external fragmentation, but still suffer from internal fragmentation.
- Paging is simple to implement and assumed as an efficient memory management technique.
- Due to equal size of the pages and frames, swapping becomes very easy.
- Page table requires extra memory space, so may not be good for a system having small RAM.

What is Segmentation ?

- Segmentation is the concept of non-contiguous memory management technique.
- In this, the process is divided into the variable size segments.
- The details about each segment are stored in a table called as segment table.
- Segment table contains base address & length of the segments.

Advantages of Segmentation

- No internal fragmentation
- Average Segment Size is larger than the actual page size.
- It is easier to relocate segments than entire address space.
- The segment table is of lesser size as compare to the page table in paging.

Disadvantages

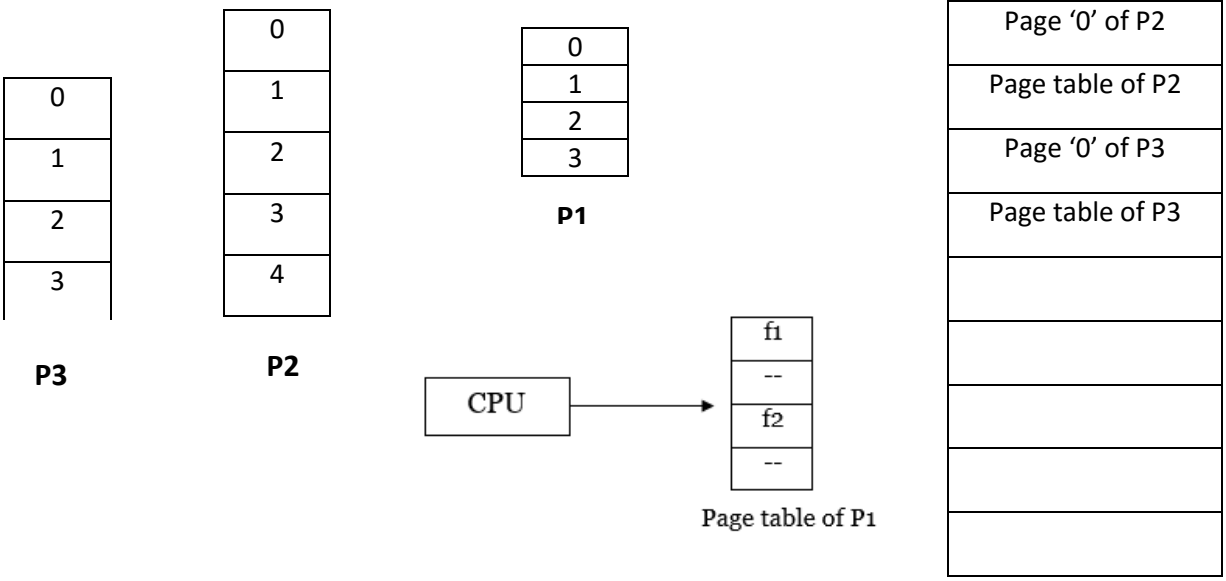
- It can have external fragmentation.
- It is difficult to allocate contiguous memory to variable sized partition.
- Costly memory management algorithms.

Difference between paging and segmentation :

Paging	Segmentation
Program is divided into fixed size called pages.	Program is divided into variable size called frames.
For paging operating system is accountable.	For segmentation compiler is accountable.
Paging is faster.	Segmentation is slow.
Internal fragmentation occurs in paging.	Here, External fragmentation is occurs.
In paging, logical address is split into page number and page offset.	Here, logical address is split into segment number and segment offset.
Paging is invisible to the user.	Segmentation is visible to the user.
In paging, processor needs page number, offset to calculate absolute address.	In segmentation, processor uses segment number, offset to calculate full address.

Virtual memory :

- It provides the illusion to the programmer that a process whose size is bigger than the size of main memory can also be executed.
- It is a storage scheme that provides user an illusion of having a very big main memory.
- Virtual memory increase the degree of multiprogramming.
- In this concept, load only required pages of process in the main memory.
- It is implemented by demand paging.
- This concept is handled by operating system.
- Now a days, virtual memory concept uses almost every computer system.



Advantage :

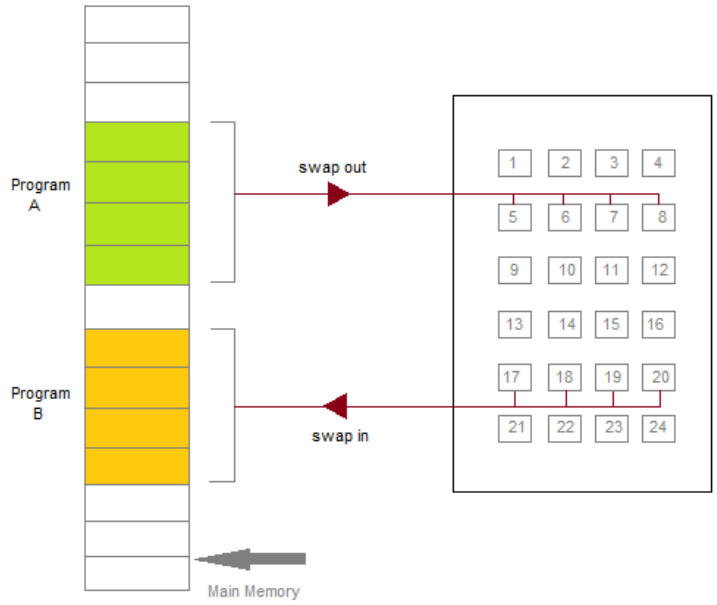
- Illusion of large main memory.
- Increase degree of multiprogramming.

Disadvantage

- It may increase page fault
- The system becomes slower since swapping takes time.
- It takes more time in switching between applications.

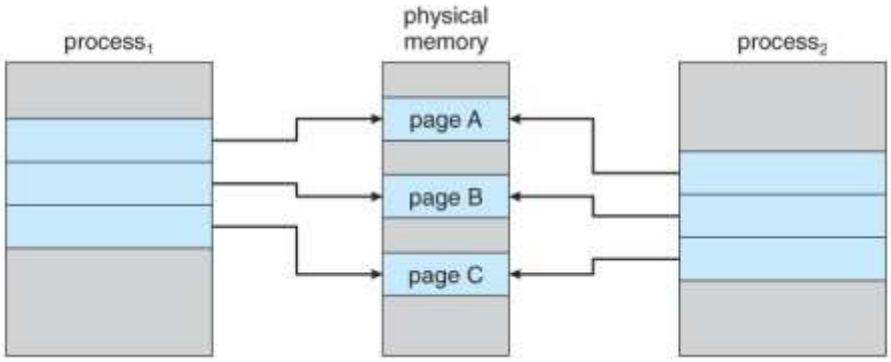
Demand paging :

- This concept tells that not load any page until it is required or request by CPU.
- The process of loading the page into memory on demand (when page fault occurs) is known as demand paging.
- If the CPU tries to access page that is currently not available in the main memory, it generates a request.
- The OS will search required page in the logical address space.
- The required page will be brought from logical address space to physical address space.



Copy on Write(COW) :

- Copy on Write is a resource management technique.
- In OS, fork() system call creates a duplicate process of the parent process which is called as the child process.
- When a parent process creates a child process then both of these processes initially will share the same pages in memory.
- These shared pages will be marked as copy-on-write which means that if any of these processes will try to modify the shared pages then only a copy of these pages will be created and the modifications will be done.



Page replacement

- It is a process of swapping out an existing page from the frame of a main memory and replacing it with the required page.
- Page replacement algorithms help to decide which page must be swapped out from the main memory.
- Various page replacement algorithms are :
  1. FIFO Page Replacement Algorithm
  2. LRU Page Replacement Algorithm
  3. Optimal Page Replacement Algorithm

Note : A good page replacement algorithm is one that minimizes the number of page faults.

FIFO Page Replacement Algorithm :

- This algorithm works on the principle of "**First in First out**".
- It replaces the oldest page that has been present in the main memory for the longest time.

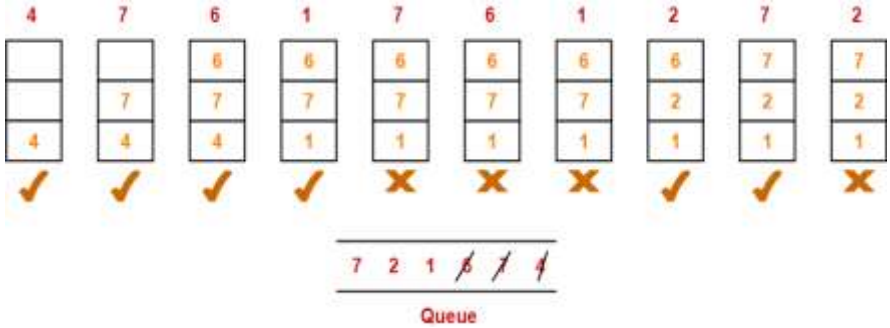
Problem-01: A system uses 3 page frames for storing process pages in main memory. It uses the FIFO page replacement policy. Assume that all the page frames are initially empty. What is the total number of page faults that will occur while processing the page reference string given below-

4 , 7, 6, 1, 7, 6, 1, 2, 7, 2

Also calculate the hit ratio and miss ratio.

Solution :

Total number of references = 10



Total number of page faults occurred = 6

Calculating Hit ratio :

Total number of page hits = Total number of references – Total number of page faults

= 10 – 6 = 4

Hit ratio = Total number of page hits / Total number of references

= 4 / 10

= 0.4 or 40%

Calculating Miss ratio :

Miss ratio = 1 – Hit ratio

= 1 – 0.4

= 0.6 or 60%

Optimal Page Replacement Algorithm :

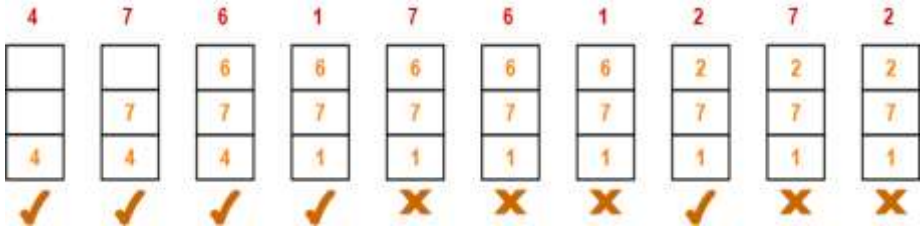
- This algorithm replaces the page that will not be referred by the CPU in future for the longest time.
- It is the best known algorithm and gives the least number of page faults.
- It is practically impossible to implement this algorithm.

Problem-02 : A system uses 3 page frames for storing process pages in main memory. It uses the Optimal page replacement policy. Assume that all the page frames are initially empty. What is the total number of page faults that will occur while processing the page reference string given below-

4 , 7, 6, 1, 7, 6, 1, 2, 7, 2

Also calculate the hit ratio and miss ratio.

Solution : Total number of references = 10



Total number of page faults occurred = 5

- Hit ratio = 0.5 or 50%
- Miss ratio = 0.5 or 50%

LRU Page Replacement Algorithm :

- This algorithm works on the principle of "**Least Recently Used**".
- It replaces the page that has not been referred by the CPU for the longest time.

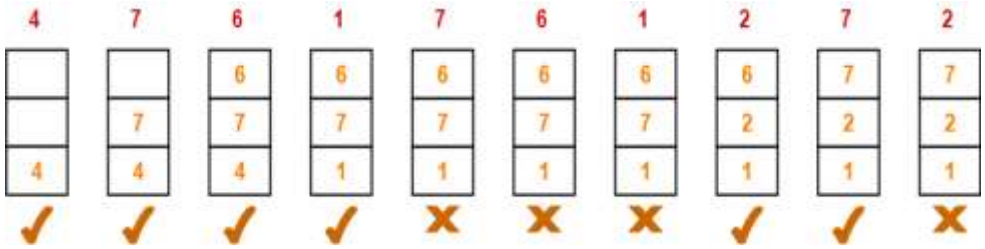
Problem-03 : A system uses 3 page frames for storing process pages in main memory. It uses the LRU page replacement policy. Assume that all the page frames are initially empty. What is the total number of page faults that will occur while processing the page reference string given below-

4 , 7, 6, 1, 7, 6, 1, 2, 7, 2

Also calculate the hit ratio and miss ratio.

Solution :

Total number of references = 10



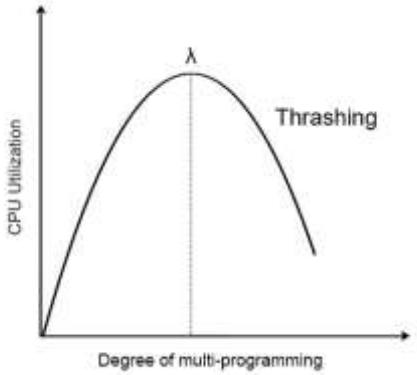
Total number of page faults occurred = 6

In the similar manner as above-

- Hit ratio = 0.4 or 40%
- Miss ratio = 0.6 or 60%

What is thrashing ?

- A high page fault rate is called thrashing.
- It is also an undesirable state of the system like deadlock.
- Thrashing occurs due to increasing of degree of multiprogramming.
- Lack of frames in main memory cause thrashing.
- To prevent thrashing we must provide processes with as many frames as they need "right now".



Causes of Thrashing :

- High degree of multiprogramming.
- Lacks of Frames.

Recovery of Thrashing :

- Do not allow the system to go into thrashing by instructing the long-term scheduler not to bring the processes into memory after the threshold.
- If the system is already thrashing then instruct the mid-term scheduler to suspend some of the processes so that we can recover the system from thrashing.

Unit – 06 : Files and protection

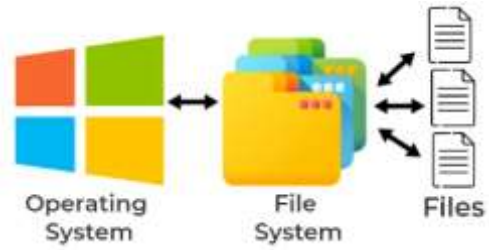
- Introduction to file systems –
- File system design,
- Access methods –
  - sequential,
  - Direct,
- File allocation methods
- OS-security :
  - Authentication,
  - Access control,
  - Access Rights,
  - System logs,
  - Protection.

Questions to be discussed :

1. What is file ? Explain file system in details.
2. Discuss about different operations performed on file.
3. Explain different types of file access methods in OS.
4. Discuss about file allocation methods in brief.
5. Write short notes on :
   a. Authentication
   b. Protection

What is the file ?

- A file is a collection of related information with specific name.
- The information within the file is recorded on secondary storage such as magnetic disks & magnetic tapes etc.
- In general, a file is a sequence of bits, bytes, or records whose meaning is defined by the file creator and user.
- Every File has a logical location where they are located for storage and retrieval.
- The file can be explained as the smallest unit of storage on a computer system.
- The user can perform file operations like open, close, read, write, and modify.
- Different types of files are Text file, Image file, Source file, Object file, Executable file etc.



Text file :

- A text file is a sequence of characters organized into lines or pages.

Image file :

- An image file is a sequence of visual information.

Source file :

- The source file has subroutines and functions that are compiled later.

Object file :

- An object file is a sequence of bytes used by the linker.

Executable file :

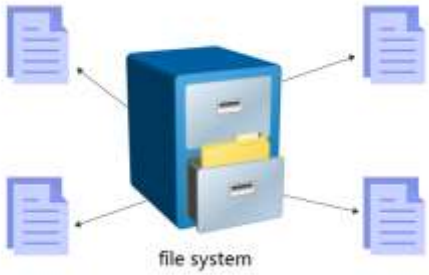
- In an executable file, the binary code that is loaded in the memory for execution is stored.
- It is stored in an exe type file.

Operations performed on file :

- To define a file properly, we need to consider the operations that can be performed on files.
- The operating system provides system call to create, write, read, reposition, delete and truncate files.
- There are six basic operations performed on file :
  1. Creating a file
  2. Writing a file
  3. Reading a file
  4. Repositioning within a file
  5. Deleting a file
  6. Truncating a file

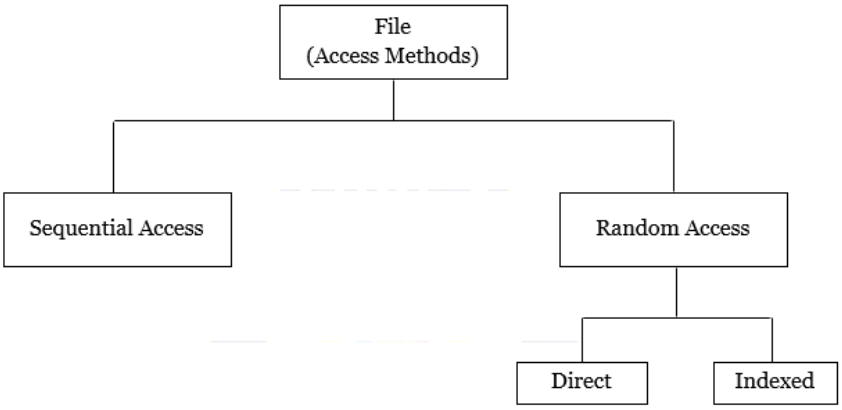
What is File Systems ?

- A File System is a data structure.
- It stores data and information on storage devices making them easily retrievable.
- Different OS's use different file systems, but all have similar features.
- In other words, a file system organizes the data blocks into files, directories, and file information.
- File system is the part of the operating system which is responsible for file management.
- The File system takes care of the following issues
  - File Structure
  - Recovering Free space
  - Disk space assignment to the files
  - tracking data location



File Access Methods :

- When a file is used, information is read and accessed from computer memory.
- There are several ways to access this information of the file.
- Some systems provide only one access method but other support many access methods.
- There are three ways to access a file into a computer system :
  1. Sequential-Access,
  2. Direct Access,
  3. Index sequential Method.

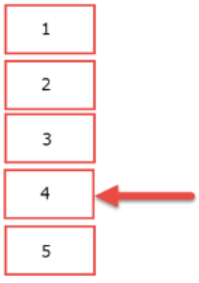


Sequential Access :

- It is the simplest access method.
- Information in the file is accessed in the order one after another.
- Example, editor and compiler usually access the file in this fashion.
- Such a method is reasonable for magnetic tape.

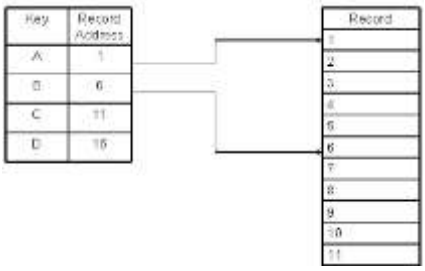
Direct Access :

- Direct access method also known as relative access method.
- It allows to read and write record rapidly in no particular order.
- The direct access is based on the disk model.
- For direct access, the file is viewed as a sequence of block or record.



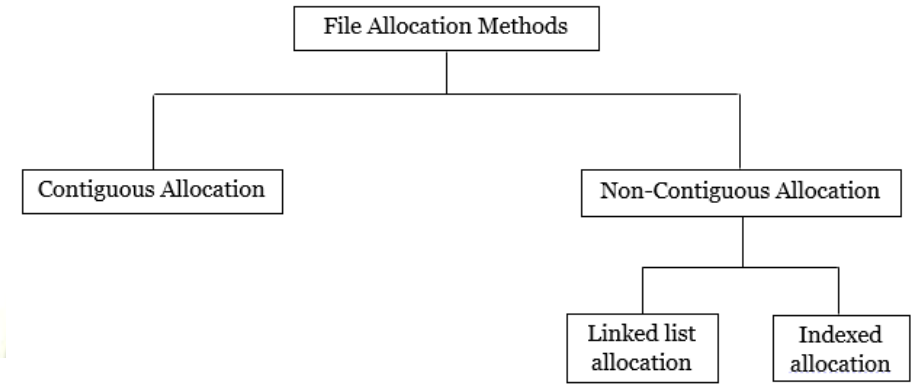
Index sequential method :

- These methods construct an index for the file.
- An index in the back of a book, contains the pointer to the various blocks.
- To find a record in the file, we first search the index, and then by the help of pointer we access the file directly.



File Allocation Methods :

- The allocation methods define how the files are stored in the disk blocks.
- There are three main disk space or file allocation methods.
  1. Contiguous Allocation
  2. Linked Allocation
  3. Indexed Allocation



Contiguous Allocation :

- In this scheme, each file occupies a contiguous set of blocks on the disk.
- If a file requires n blocks and is given a block x as the starting location, then the blocks assigned to the file will be: x, x+1, x+2,.....x+n-1.
- The directory entry for a file with contiguous allocation contains
  - Address of starting block
  - Length of the allocated portion.

Linked List Allocation :

- In this scheme, each file is a linked list of disk blocks which need not be contiguous.
- The disk blocks can be scattered anywhere on the disk.
  The directory entry contains a pointer to the starting and the ending file block.
- Each block contains a pointer to the next block occupied by the file.

Indexed Allocation :

- In this scheme, a special block known as the Index block contains the pointers to all the blocks occupied by a file.
- Each file has its own index block.
- The i^{th} entry in the index block contains the disk address of the i^{th} file block.

OS Security :

- If an illegal user runs a computer application then, the data stored may be damaged.
- Security means protect against unauthorized access.
- It providing safety for computer system resources like software, CPU, memory, disks, etc.
- The processes of protecting the operating system from dangers is called OS security.
  - Authentication,
  - Access control,
  - Access Rights,
  - System logs,
  - Protection.

Authentication :

- It is allow authorized users to access the computer and to deny access to unauthorized users.
- OS generally authenticates users using 3 ways: Passwords, User Card, and Biometrics.

Access control :

- It is a method of limiting access to a system or to physical or virtual resources.
- It is a process by which users can access granted certain resources or information.

Access Rights :

- Access Rights are the permissions an individual user to read, write, modify, delete or otherwise access a computer file, change configurations or settings, or add or remove applications.

System logs :

- A log file is a computer-generated data file that contains information about usage patterns.
- The system log (syslog) contains a record of the operating system (OS) events that indicates how the system processes and drivers were loaded.

Protection :

- It is a mechanism which controls the access of programs, processes, or users to the resources defined by a computer system.
- We can take protection as a helper to multi programming operating system, so that many users might safely share a common logical name space such as directory or files.