WHAT IS AN OPERATING SYSTEM

<1> An operating system is a software that operates the computer system.

<2> It is a manger of the computer system.

<3> An operating system is a software i.e. system software.

<4> An operating system is a software/system software a set of program that manages the resources of computer system such as I/O device, CPU, memory, file, disk etc.

<5> Every general purpose computer must have an operating system to run other program.

<6> It is used to run and develop other program.

<7> An operating system is a software that enables the computer hardware to communicate and operate with the computer.

<8> An operating system is like a traffic cop (means police). It makes sure that different program and users are running at the same time without interference with each other.

<9> An operating system is a platform for developing the application program.

<10> It is also responsible for security, ensuring that unauthorized user does not access the system.

<11> Operating system perform some basic tasks such as recognizing input from the keyboard, sending output to display information, keeping tracks of files

Example- LINUX, UNIX, Win'2000

Multitasking O.S - An Operating System that is capable of allowing multiple software process to run at the same time or allows more than one program to run concurrently.

Example- LINUX, UNIX, Win'2000

Multithreading O.S - An Operating System that allows different parts of a software program to run concurrently.

Example- LINUX, UNIX, Win'2000

EXAMPLES OF OPERATING SYSTEM

There are many types of operating system. The most common is :-

(a) Microsoft Operating System → They include from recent to the oldest.

- Win' 2007
- Win' VISTA
- Win' XP professional Edition
- Win' XP Home Edition
- Win' 2000
- Win' ME
- Win' 98 Second Edition
- Win' 98 First Edition
- Win' NT
- Win' 95
- Win' 3.x version

Non-Microsoft Operating System → There are other types of operating system not made by Microsoft. The greatest problem of these operating systems are that they lie not as many application program as written on them. Ex-

UNIX → UNIX; pronounced as UNICS (Uniplexed Information and Computing Services) developed by MULTICS (Multiplexed Information and Computing Services) team at Bell Laboratory (Bell Lab's) starting in 1960's by many of the same people who helped created the C programming language. It can be difficult to learn.

LINUX → LINUX, developed by 'LINUX Torvaldes' and further elaborated by a number of developer throughout the world. It is similar to UNIX operating system but it is free and GUI O.S.

FUNCTIONS OF OPERATING SYSTEM

I. MAJOR FUNCTION OF O.S ARE-

(i) acts as an extended machine → An operating system acts as an extended machine by translating our command into machine language instructions and retranslates the output back into the user understandable language.

(ii) acts as a resource manager → An operating system acts as a resource manager by controlling and

locating various hardware and software resources to different user in an optimal and efficient mode.

The task of resource management become essential in multiuser operating system where different user compete for the same resources.

(iii) acts as a constant application program Interface →

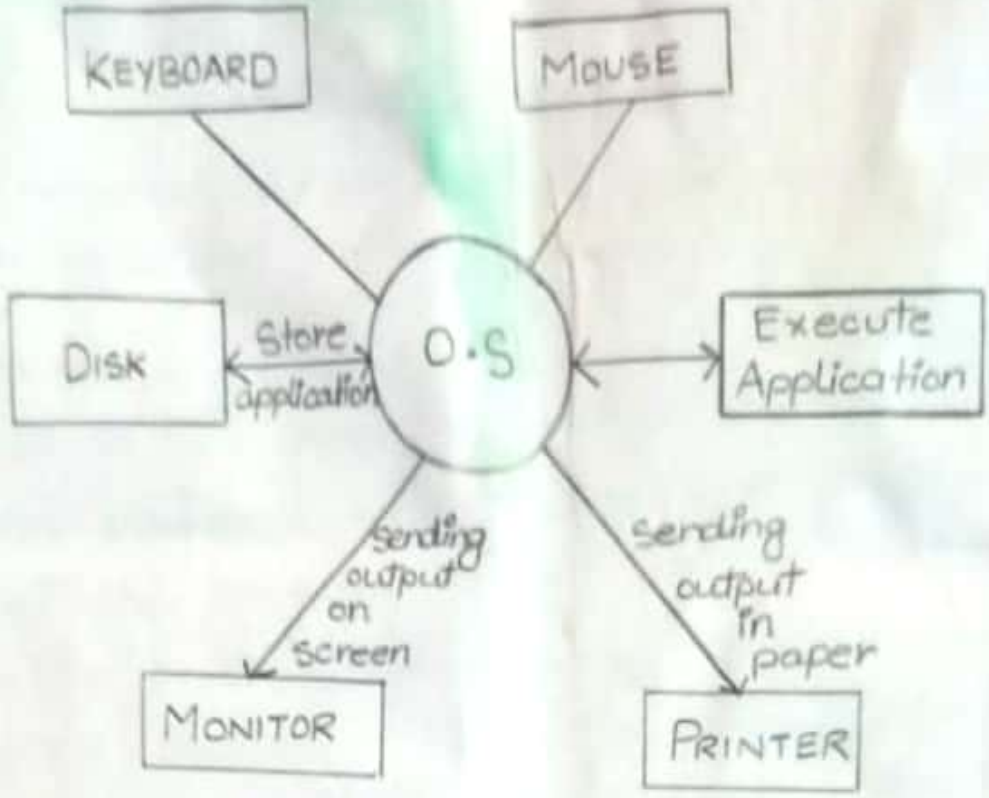
An operating system acts as a constant application program Interface that allow us to develop an application on a computer and execute it on another computer. It doesn't produces any dissimilarities in the configuration of computer which is different as the application remains at the same time.

II. OTHER FUNCTIONS OF OS ARE-

(i) System tools used to monitor computer performance, debug problems or maintain parts of the system.

(ii) A set of libraries or a function program may used to perform specific task specially relating to interfacing with other computer components.

III. BASIC FUNCTIONS OF O.S ARE -



(i) perform Input - output operation

(ii) provides security by ensuring control access to resources.

(iii) acts as an command interpreter to translate.

(iv) control the device driver attached to the computer and control the files (remote and local) places on the computer.

(v) control the allocation and reallocation of memory.

(vi) decides the priority of job.

(vii) monitor various job running on computer for their security.

GENERAL FUNCTIONS OF O.S ARE-

(i) Process Management

(ii) Memory Management

(iii) I/O Management

(iv) File Management etc.

INTERACTION OF OPERATING SYSTEM WITH HARDWARE & USER PROGRAMMING

(1) A computer is made up of hardware, system software and application software(program).

(2) Hardware →

(i) It is the essential part of the computer system

(ii) It represents the physical and tangible components of the computer system.

(iii) The part which can be touched and seen.

(iv) It includes - Input devices, Output devices, storage devices, CPU, Electronic circuits consisting of resistors, wires, IC etc.

(v) The hardware is categorized into -

(a) physical devices

(b) Microarchitecture

(c) Machine Language

(a) physical devices → physical device consists of ICs, power supply and wires.

(b) Microarchitecture → It is the functional grouping of physical devices that consists of CPU, register and data path

Register → It is a hardware block that store bit of information.

Data path → Flow of data into ALU and is controlled by software called Microprogram that is located in ROM and directly control the physical devices.

Machine Language → also called Binary language is the language of 0's and 1's i.e. directly understandable by CPU.

(3) System software → System software (programs) manages the routine operation of the system.
Example - O.S, compiler, Editor etc.

(4) User program/Application program → It is for the users to solve the problem and build application program.
Example - Airline Reservation System
Hotel management System

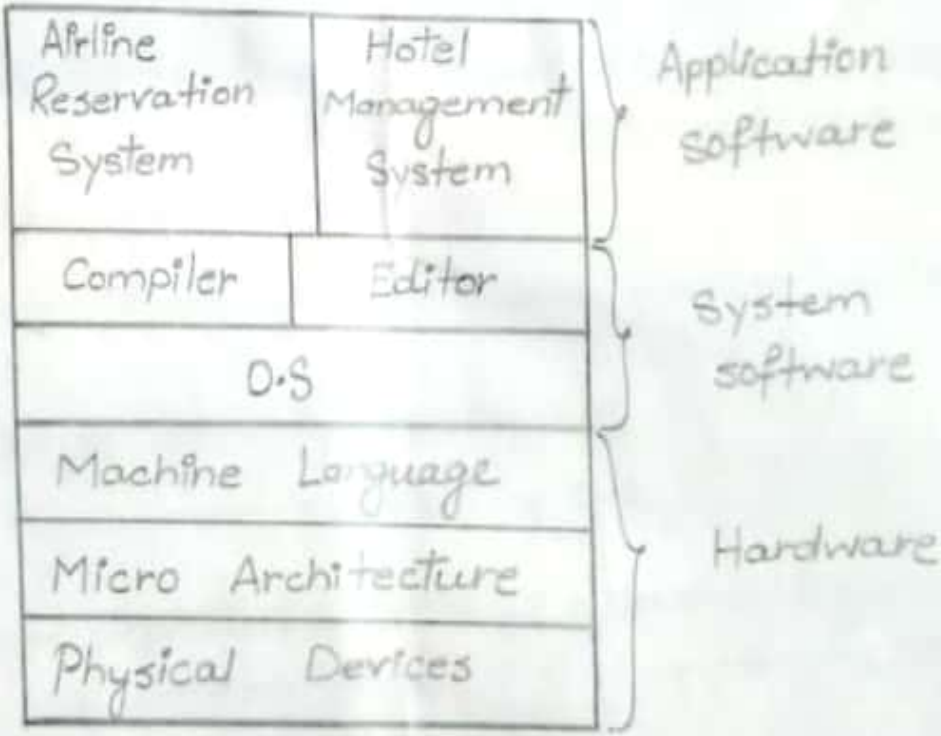
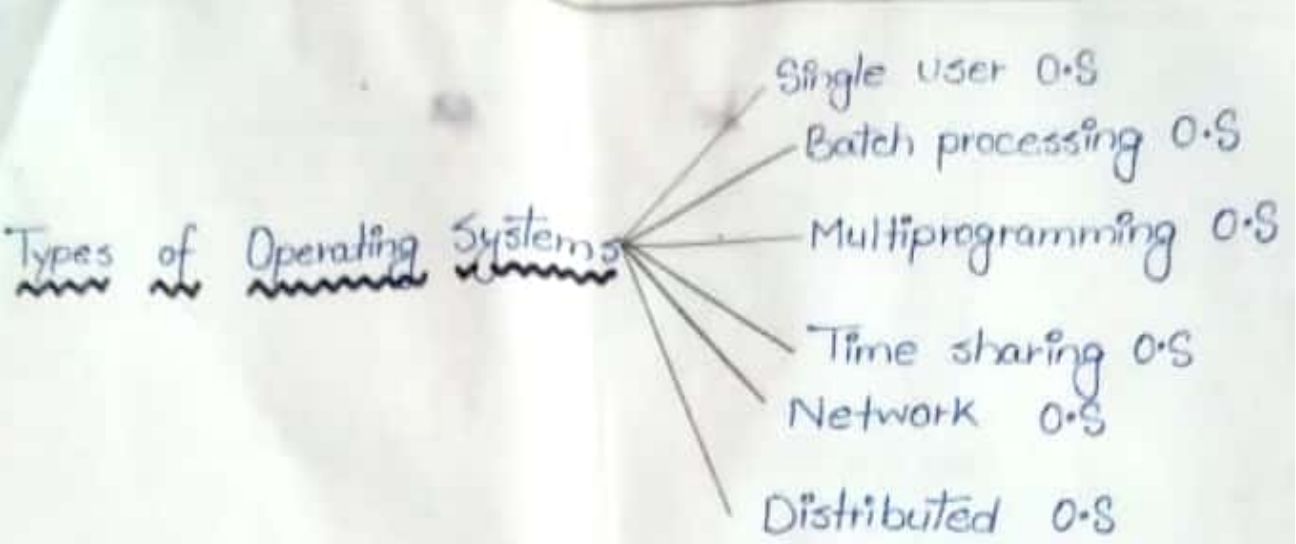


Fig shows. Elements of computer from which we see that O.S provide interface to the user and communicate or interact with the hardware



SINGLE USER OPERATING SYSTEM

(1) Single user operating system allow a single user to access the computer at a time. These computer have a single processor and execute a single program.

(2) The resources such as CPU and I/O devices constantly available to the user in a single user operating system for operating the system.

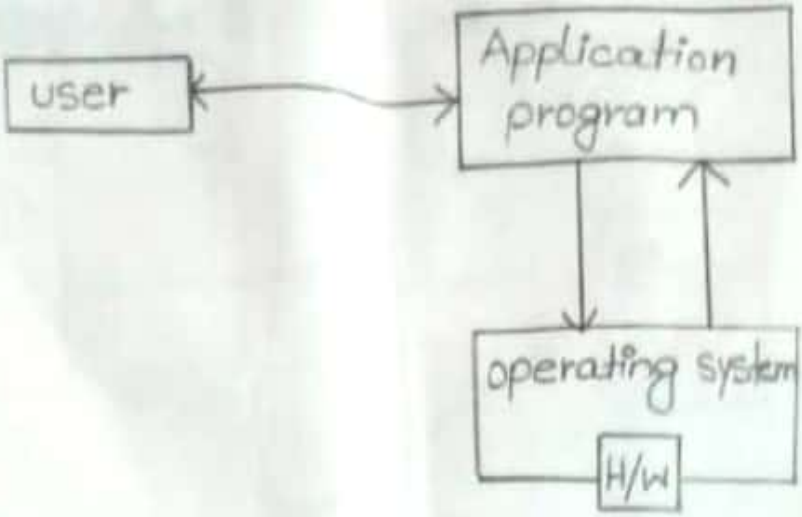
(3) Single user operating system vare categorized/divided into :-

i) Single user singletasking O.S
ii) Single user multitasking O.S

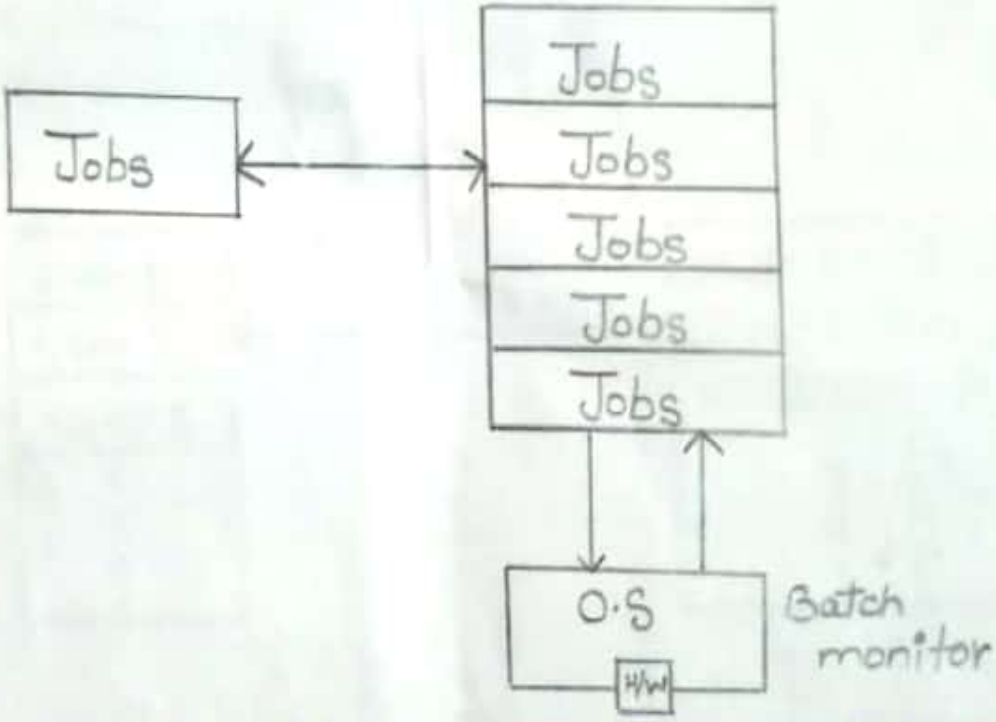
i) Single user singletasking O.S ⇒ It allows a single user to execute one program at a time. Example - Ms DOS and PALM Operating System.

ii) Single user multitasking O.S ⇒ It allows a single user to operate multiple program at the same time. Example - Macintosh O.S, all windows O.S

NOTE - Single user operating system execute an application of a user through hardware interaction and gives the result back to the user.



BATCH PROCESSING OPERATING SYSTEM

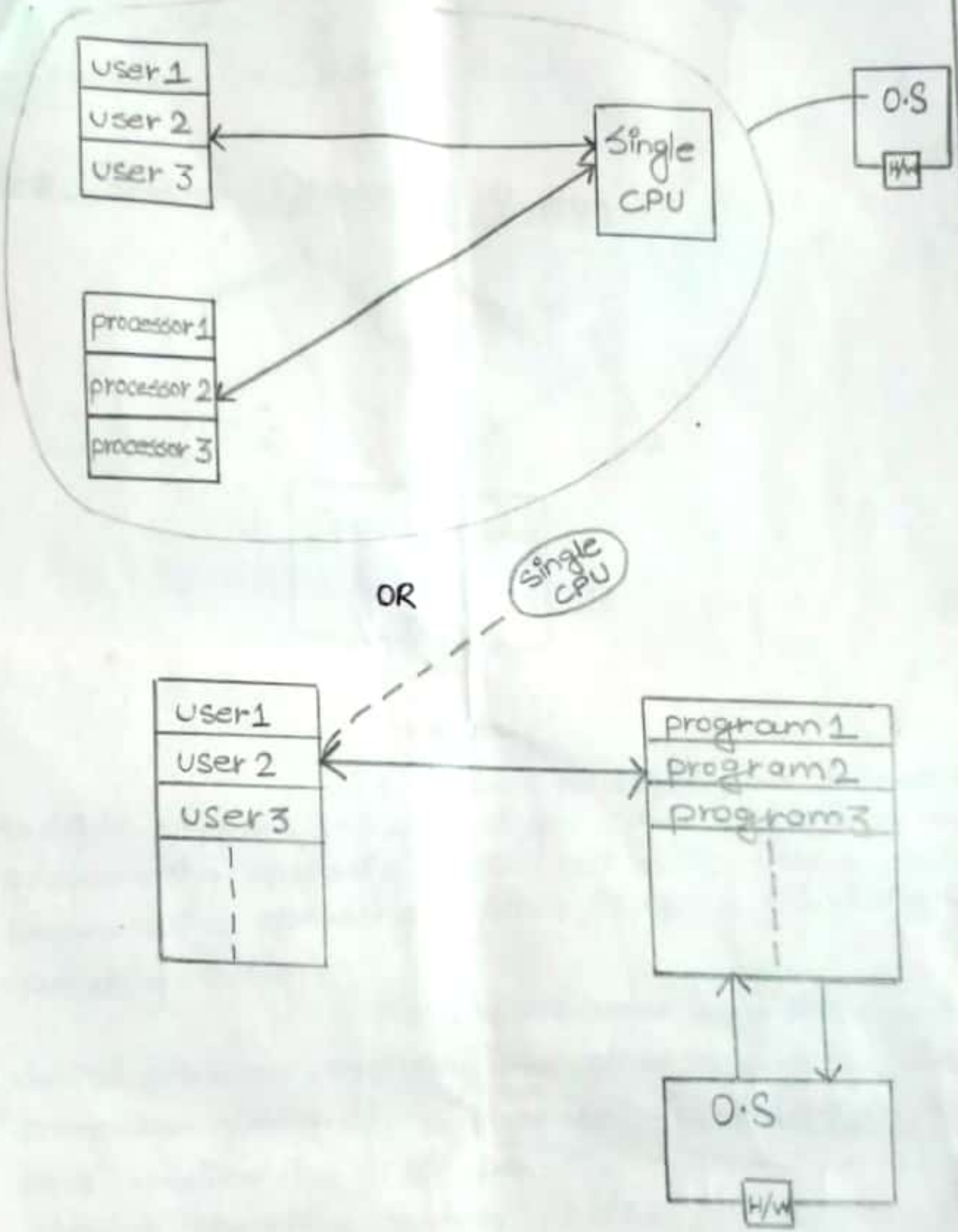


(1) The user submits jobs to a JCL (Job Control Language). The jobs are collected into a batch subsequently placed on an input queue at the computer where they will be run.

(2) Batch monitor executes the batches based on the process of job scheduling that sequences different jobs on FCFS (First-Come, First-Served) basis.

(3) Examples – UNIX, Windows 2000 Server.

PROGRAMMING OPERATING SYSTEM

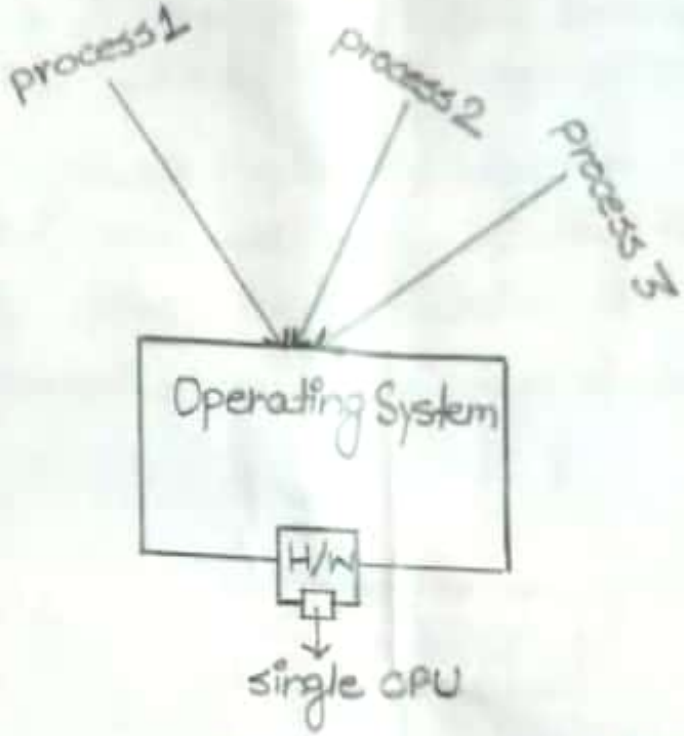


<1> It allows multiple user to execute multiple program in a single C.P.U.

e multiprogramming O.S execute different processes currently using a single CPU by implementing the concept of CPU slicing.

⇒ Example - UNIX, Win'NT, Amoeba.

TIME SHARING OPERATING SYSTEM



It allows multiple user to execute multiple program using a single C.P.U so it is a multiprogramming operating system but note that a multiprogramming operating system is not a time sharing operating system.

The various user are sharing the central processor, memory and other resources of the computer system in a manner facilitated, controlled and monitor by the operating system. The time sharing operating system provide time slice for each process. The CPU switching between the program is so fast that gives the response time for user.

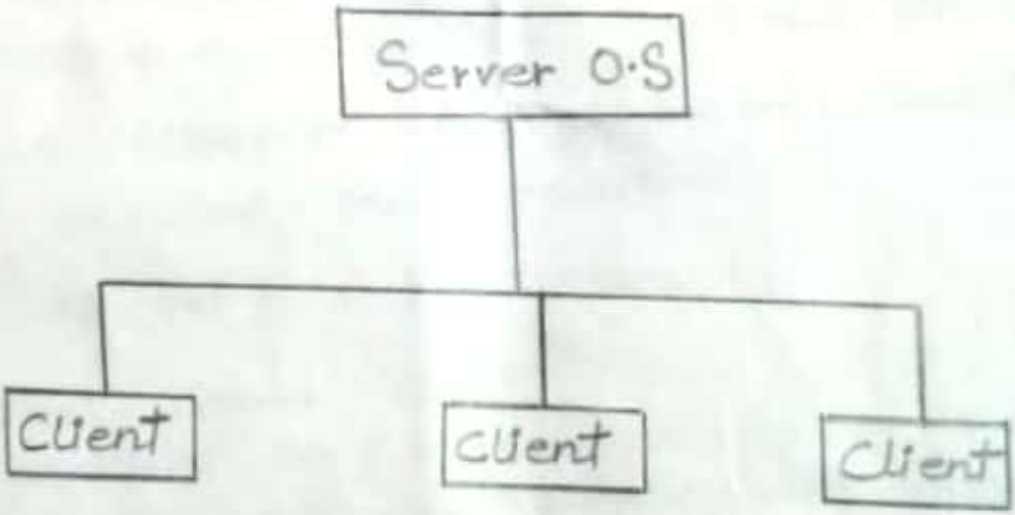
WORK OPERATING SYSTEM

A network operating system is a collection of physically interconnected computers.

(2) The operating system of each of the interconnected computer must connected in addition to its own stand-alone functionality, provision for handling communication and transfer of data among the other computers with which it is connected.

(3) In a network operating system, the users are aware of the existence of multiple machines and copy files from one machine to other.

(4) Each machine has its own local operating system.



(4) Examples - Win' NT 2000 server
Win' XP professional
UNIX
NOVELL etc.

DISTRIBUTED OPERATING SYSTEM

(1) It is a type of operating system.

(2) In a distributed operating system, processing user requests are carried out independently in more than one location.

(3) In a distributed system, the work load is spread between two or more computer link together by a communication network such as telephone line, buses, wires, lease line etc.

(4) The processor in distributed Operating system has its own resources.

(5) The processor in distributed operating system doesnot share clock, memory and pheripheral devices.

(6) Example - Amoeba, NOVELL Netware etc.

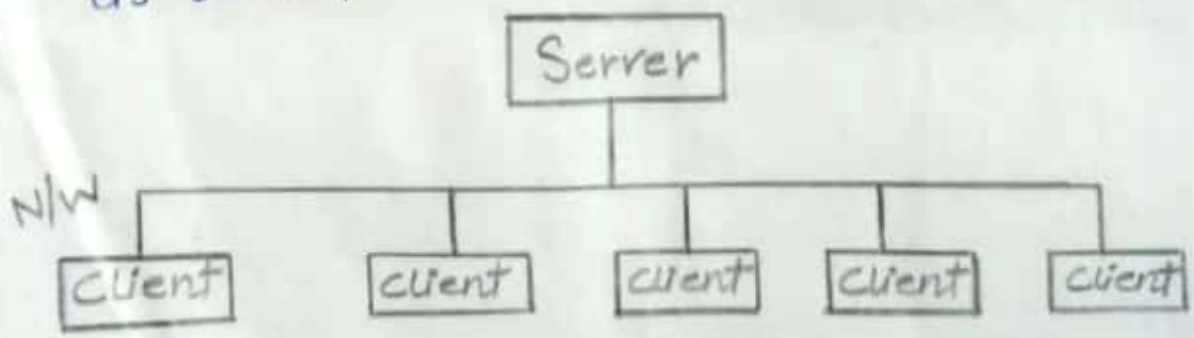
(7) It is based on two paradigms/model -

a) Client - Server Model

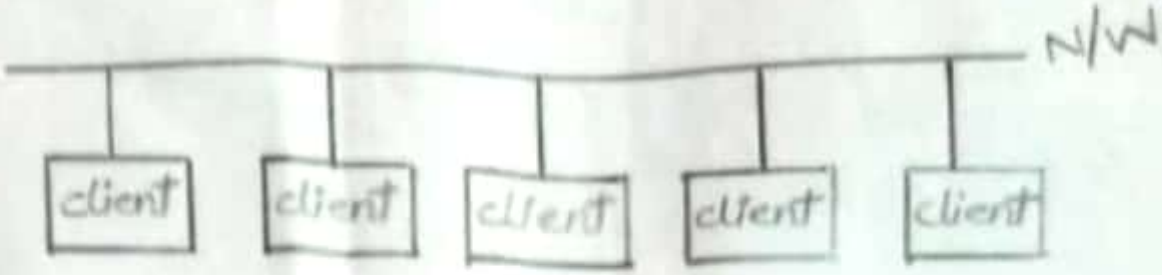
b) Peer - to - Peer Model

a) Client - Server Model

The client sent a resource request (software, file, data etc.) Ex- Printer to the server and the server in turn provides the requested resource as a response back to the client.



Peer- to - peer Model
All computer behaves as peer (समान) as well as client. These client communicate with each other for exchange of the resource.



STRUCTURE OF OPERATING SYSTEM

Process Management

CPU executes a large number of program while its main concerned is the execution of user program, CPU is also needed for certain number of activities. These activities are called processes. Operating system is responsible for performing the following activities in connection with processes. This is called "process management" (process to manage.) Example—

(1) creation and deletion of processes,
(2) suspension/delay and resumption of processes etc.

Process

The notion (धारणा) of the process is centralised to the understanding of operating system. There are quite few definition presented in the literature but they are not perfect 'definition' as they appear.

Definition

The term 'process' was first used by designer of MULTICS in 1960. Since, then the term process used some what interchangeably with 'task' or 'job'.

(1) A program in execution (To perform to complete).
(2) A process is an instance of program.
(3) A process is the unit of work in the system.
(4) The entity assign to the processor.
(5) The Dispatchable unit.
(6) A process is a framework of set of sequential step for performing task.
(7) animated spirit of a procedure in a sequence.

Process States

A process state is indicative of the activity which is currently performed. Example-

w state: process is being created.
ady state: The process waiting for a chance to be allocated to C.P.U time. etc...

PROCESS STATES TRANSITION

(1) A process goes through various state for performing its task.

(2) The transition of a process from one state to another occur depending on the flow of execution of the process.

(3) It is not necessary for a process to undergo all the states.

(4) Various process states are-

a) New state ⇒ Process is being created.

b) Ready state ⇒ The process waiting for a chance to be allocated to CPU time.

c) Running state ⇒ When a process is being executed.

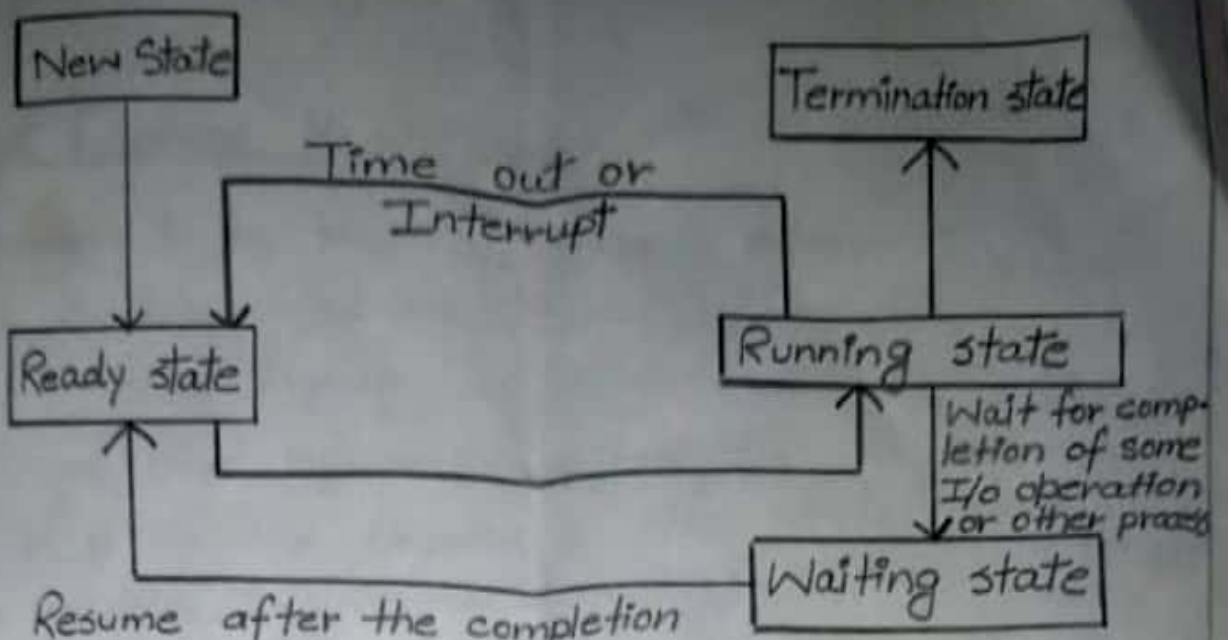
d) Waiting state ⇒ It is also called Block state. When a process waits for the termination of an I/o operation or another process.

e) Terminate state ⇒ The process has finished execution.

BLOCK DIAGRAM OF PROCESS TRANSITION:-

OR

TRANSITION OF A PROCESS FROM ONE STATE TO ANOTHER



A new process is admitted into a data structure called ready queue or ready state.

A queue is a data structure that stores all data in first-in-first-out manner. The process is inserted first in the ready queue and then it is sent for execution.

Each process assign a time slice for execution. The C.P.U executes the process at the front end of the ready queue and send to the running state. A time out occurs if the process is not completed within the time slice.

When a process is interrupted, the running process is sent back to the ready queue. When a process waits for the completion of an I/o task or another process, it is sent to the waiting state, this process state is marked as waiting. Then after completion of I/o task, the process goes back to the ready state.

When execution of a process is, the process goes to terminated state.

PROCESS CONTROL BLOCK (PCB)

(1) A process in an operating system is represented by a data structure called process control block.

(2) It is also known as process descriptor.

(3) It get the information about the process.

(4) A PCB is the location in the main memory, where various information regarding a process is stored.

(5) Each process has a single PCB.

(6) When the process is terminated, the PCB is released from the memory.

(7) The information contained in PCB about the specific process includes -

i) Current state ⇒ current state of process i.e. whether it is ready state, running state, waiting state or whatever.

ii) process ID ⇒ unique identification of the process in order to track "which is which" information.

iii) Link to parent process ⇒ A new process (i.e. child can be created from an existing process.) The existing process is called parent process.

iv) Link to child ⇒ (If it exists)

v) Priority of a process ⇒ The priority of a process (a part of CPU scheduling information.)

vi) A register save area.

vii) The processor it is running on.

The PCB is certain store that allows the operating system to locate key (बहुत जरूरी) information about the process.

Thus a PCB is a data structure or descriptor that defines the process of the O.S.

OPERATION ON PROCESS OR

O.S SERVICES FOR PROCESS CONTROL

(1) Operating system services are set of routines required for operating and managing the application.

(2) In an Operating system, categories services are available such as process control, file management, device management, memory management etc.

(3) operating system services are invoked/started by system call, system call provide an interface between process and operating system.

(4) O.S system call for process management or operations available in various O.S are such as

a) Create system call/operation.
b) Fork/Join system call/operation.
c) Execute system call/operation.
d) Delete system call/operation.
e) Wait/suspend system call/operation.
f) Resume system call/operation.
g) Abort system call/operation.

a) Create system call ⇒ create a new process with a blank PCB. The Operating system generates process ID and assigned it to the new process.

ther attributes (field of PCB) are copied to the PCB of the new process and process is transferred to the ready queue. A process can be created by another process.

b) Fork/Join system call ⇒ Fork system call making easy creation of a process by bifurcation of an existing process.

In other words, fork system call creates a duplicate process. The new process is the child of the existing process from which the fork system call has been invoked/started. The values in the various field of the existing processID of the PCB are copied to the PCB of the new process.

c) Execute system call ⇒ Execute a process from the running state.

d) Delete system call ⇒ Delete system call terminates a process. A process can be deleted by itself or from other process. After the deletion of a process, the Operating System releases and reclaim the resources such as file or memory space allocated to the deleted process. It also releases the PCB of the deleted process.

e) Wait system call ⇒ It is also known as suspend system call. The wait system call suspend a running process and waits for the completion of an I/O operation or another process. It is also called locking of a process.

Resume system call ⇒ It is also called wake-up system call. It is used for restoring a block process. A process cannot involved the resume system call itself because it must be in the running state to invoke a system call. A suspended process needs for another process for restoring.

8) Abort system call ⇒ It is similar to delete system call except the abort is invoked for forcefully (अत्यधिक) termination of a process. While executing the abort system call, the operating system perform similar task as delete but it save the information regarding the process. Operating system displays the information such as reason for aborting the process and values of the register at the time of aborting in a file on the computer system.

CRITICAL SECTION

<1> A process is in critical section if it executes code that manipulates shared data and resources.

<2> In other word, critical section is that overlapping portion of each process where shared data and resources are accessed.

<3> Critical section problem is to design a protocol to achieve cooperation among processes.

Each process should seek permission to enter its critical section.

<4> The code section that implements this request is called entry section.

<5> An exit section may follow critical section.

Remainder section → Remaining codes comes under remainder section.

General Structure of a Typical Process

```
Repeat {
        Entry section;
        Critical section;
        Exit section;
        Remainder section;
        until false.
}
```

(7) There are three necessary and sufficient requirement for critical section solution. These are as follows:

(a) Mutual Exclusion
(b) Bounded Waiting
(c) Progress

(a) Mutual Exclusion ⇒ It states that if one process is executing in its critical section, no other process can execute in that critical section.

(b) Bounded Waiting ⇒ It states that if a process makes a request to enter its critical section and before that request is granted there is limit number of time, other processes are allowed to enter that critical section.

(c) Progress ⇒ It states that a process cannot stop other processes for entering their critical section if it is not executing in its critical section.

V.V.I

RULES OF MUTUAL EXCLUSION

OR

MUTUAL EXCLUSION PRINCIPLE

(1) Resources like memory and certain input output cannot be used by more than one process at a time.

(2) The principle of mutual exclusion states that a process has unique access to shared resources in critical section.

(3) For a given resource among competing processes, one process can be in its critical section at an instant of time.

For Example- If a process is using a printer then no other process can use it until first process is over-come with its work.

SEMAPHORES

Semaphore is a variable with an integer value on which three basic operations are defined.

(1) Initialization - to non-negative value.

(2) Wait operation → Decrement semaphore value. A process executing wait is blocked if its value become negative.

Pseudocode for wait operation

```
wait(s)
while(s<=0)
{
//no-operation
    s--;
}
```

signal operation → Increment semaphore value.
A process blocked by wait operation is unblocked if value become positive.

Pseudocode for signal operation

```
signal(s)
{
  s++;
}
```

V.V.I

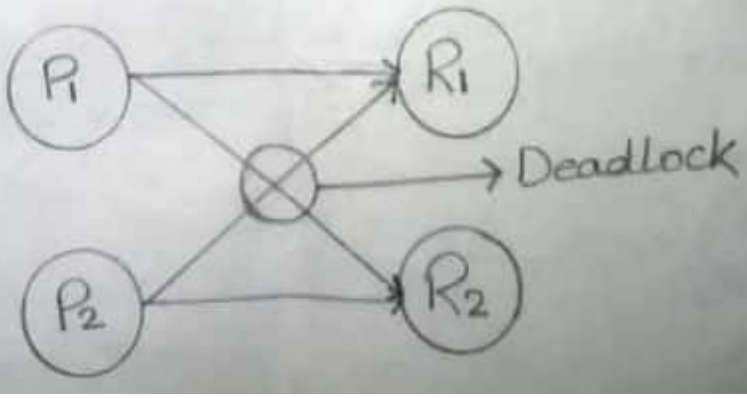
DEADLOCK

(1) In a multiprogramming O.S, resources such as I/O devices, CPU cycle and memory are used by multiple number of processes. There can be multiple number of instances of a resource.

(2) In a computer system deadlock occurs when two or more process are attempt to access the resource that the other process has locked. Hence, cannot be shared as a result. Each process wait for a resource that the other has locked and neither transaction can finished.

(3) A deadlock is a situation where two or more competing action are waiting for the other to finish and this neither ever does.

For Example-



resources, a CD-writer drive (i.e. R_1) and a printer (i.e. R_2) attached to a computer. To process P_1 and P_2 exist in the ready queue. Both the process P_1 and P_2 wants to read a document, write it back to the CD and get modified document printed. Both the processes P_1 and P_2 require both the resources, the CD-writer drive (i.e. R_1) and printer (i.e. R_2). The process P_1 acquire the CD-writer drive (i.e. R_1) and lock it in non-sharable mode. The process P_2 acquire the printer (i.e. R_2) and lock it in non-sharable mode. The process P_1 waits for obtaining the printer and process P_2 waits for obtaining the CD-writer drive. This led to a situation which is called a deadlock.

Necessary Conditions for Deadlock

Following are the conditions under which deadlock can occur:-

<1> Mutual Exclusion
<2> Hold-and-wait
<3> No-Preemption
<4> Circular wait

<1> Mutual Exclusion ⇒ One or more resources should be require in exclusive mode. A process 'P' has locked a resource. This means no other process can acquire the resource 'R'.

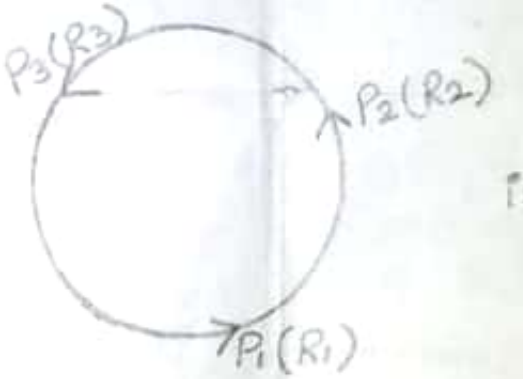
In other word, mutual exclusion means that a resource can be utilized by one process at a time.

Hold-and-wait ⇒ The process involved in a deadlock acquires atleast one process in exclusive mode and wait for atleast another resource that has been acquired by another process in exclusive mode.

<3> No-Preemption ⇒ processes do not releases the resource allocated to them without utilizing the resource. For Example ⇒ process 'P_1' hold the resource 'R_1' exclusively waits for the resource 'R_2' which is locked by process 'P_2' in exclusive mode.

The concept of no-preemption denotes that operating system cannot revoke the resource from the process forcefully.

<4> Circular wait ⇒



i = 1, 2, 3, 4 ..... n

A close chain or circular relationship exist among the processes involved in a deadlock.

For Example- There are three processes P_1, P_2 and P_3. The process P_1 is waiting for a resource that has been exclusively acquired by the process P_2, the process P_2 is waiting for a resource which is hold exclusively by the process P_3. The process P_3 is waiting for a resource which is acquired exclusively by the process P_1.

Deadlock Prevention

Deadlock prevention techniques allocate the resources in such a manner that atleast one of the four necessary condition of deadlock cannot occur. It denies atleast one of the condition acquired for occurrence of a deadlock. Deadlock prevention technique do not utilize a resource properly.

We have already discussed that there are four conditions to be satisfied for the occurrence of a deadlock. These conditions are:-

<1> Mutual Exclusion
<2> Hold-and-wait
<3> No-Preemption
<4> Circular wait

<1> Mutual Exclusion ⇒ The mutual exclusion principle is usually difficult to denied/displaced with, so one or more of the remaining conditions are to be prevented.

The mutual exclusion condition indicates locking of resources in exclusive mode, also known as non-sharable mode. An operating system can avoid blocking the resources in sharable mode if visible.

For Example- Two processes needs to read data from a file. The file is locking in sharable mode by two processes simultaneously. Locking of resource in sharable mode prevent waiting for a resource.

Some resources cannot be shared by multiple processes. For Example- Two process cannot simultaneously share a scanner.

• Hold-and-wait ⇒ The second condition for the occurence of deadlock is hold-and-wait. Operating system can avoid this by adopting a strategy that a process must request for all resources of the same time. This strategy can be implemented in two ways:-

i) A process can request all the resources it need when execution begin.

ii) Secondly, a process must release all the resources it presently hold before requiring any other required resource.

For Example- A process 'P' require resources R_1, R_2, R_3 and R_4 in exclusive mode in the sequence. R_3 and R_1 simultaneously at the beginning of the execution. R_1 and R_2 resources, then R_3 simultaneously and lastly R_1, R_4 simultaneously.

When applying the first strategy, the process 'P' acquires all the four resources simultaneously before it start execution.

When applying the second strategy, the process 'P' need not acquire all the four resources at the beginning. The process 'P' acquire the resources R_1 and R_2 at the time of start of execution. The process 'P' request the operating system for acquiring the resources R_1, R_2, R_3. The process 'P' works and releases all i.e. R_1, R_2, R_3 and send request for R_1 and R_4.

(3) No-Preemption ⇒ The third condition is no preemption. An operating system can adopt a policy to avoid this condition. Operating system stop the resources if it is responsible for deadlock and starting for further requesting the resource.

(4) Circular wait ⇒ The fourth condition of occurrence of deadlock is circular wait. In this approach, system resources are divided into different classes j (where $j = 1, 2, \ldots\ldots n$).

Deadlock are prevented by requiring all processes to request and acquired the resources in strictly increasing order of specified system resource class. Once a process acquire a resource belonging to the class j (i.e. C_j), it can only request resource of C_{j+1} or higher thereafter.

VNI

DEADLOCK AVOIDANCE

(1) In deadlock avoidance, operating system requires additional information about which process requires which resource in advance to avoid a deadlock.

(2) The fundamental concept is that operating system only entertain those request for resource by a process that will not lead to a deadlock.

(3) Banker algorithm is a theoretical approach to avoid deadlock in resource scheduling.

(4) Banker algorithm adopts a strategy. The name banker depicts (comes from) the similarity of concept of the field of banking.

It is usually explained using an analog
i) The customer are process.
ii) Money is the resource.
iii) Bank is the Operating system.
iv) Each customer has credit limit.

There are two part of Banker algorithm
I. Safety algorithm
II. Resource Request algorithm

(I.) Safety Algorithm
Safety algorithm determines whether or not a system is in a safe state.

Step 1: Let Work and Finish, be arrays of length m and n respectively.
Initialize: Work = Available and
Finish[i] = False; for i=1, 2, ... n

Step 2: Find an i such that both
Finish[i] = False
Needi ≤ Work

Step 3: If no such i exists goto step 4; otherwise
Work = Work - Allocationi
Finish[i] = True
goto step 2

Step 4: if Finish[i] = True ∀, or, for all i
then the system is in safe state.

(II.) Resource Request Algorithm
The resource request algorithm determines whether a resource request can be safely granted.
Step 1: Let Requesti be the request array for process Pi.

Step 2: If Requesti > Needi that means it get errors because resource claim is maximum.

Step 3: If Requesti > Available then block process Pi for resource request that available resource.

Step 4: Here the system pretend to allocate the requested resources to the process Pi by modifying the state as follows:

Available = Available - Requesti

Needi = Needi - Requesti

Allocationi = Allocationi + Requesti

If the resulting resource - allocation state is safe, the transaction is completed and process Pi is allocated its resources. However, if the new state is unsafe, then Pi must wait for Requesti and the old resource - allocation state is restored.

DEADLOCK DETECTION AND RECOVERY

(1) Another approach of deadlock management is detection and recovery.

(2) In this approach, the system is not prevented from occurring a deadlock.

(3) Operating system periodically searches and analyses whether a deadlock occurs

(4) This approach of deadlock management has two phases - Detection and Recovery of a deadlock.

(5) The algorithm to detect deadlock in an operating system can use data structure similar to that

of data structure(array) used in the context of deadlock avoidance.

(6) In this algorithm, data structure 'Allocation', 'Claim', 'Available' and 'Flag' are used.

(7) Flag is the array or vector of length equals to (=) the number of processes in the system.

(8) Flag is used to mark and unmark a process.

Flag[x] = True, means process Px is marked.

Flag[y] = False, means process Py is unmarked.

(9) An operating system verifies after a certain interval whether a deadlock has occurred.

(10) This algorithm find a sequence of execution of processes that doesnot lead in unsafe state.

Step 1: Initialize : Flag[i] = False ; for i = 1, 2, ... n.
; denotes the number of processes.

Step 2: Find a process Pi such that
Flag[i] = False and Claim$_i$ ≤ Available$_i$; otherwise
goto step 6 if no such process found.

Step 3: update Available vector because of termination of process. Operating system revoke to control all the resources held by Pi process.

Available$_i$ = Available$_i$ + Allocation

Step 4: Mark the process Pi, Flag[i] = True

Step 5: goto step 2

Step 6: Flag[i] = True, process Pi is marked and no deadlock is occurred.

Operating system can adopt a strategy for deadlock recovery. It states that all the processes involved in deadlock can be terminated forcefully.

SCHEDULING
OR
PROCESSOR SCHEDULING

In a multiprogramming Operating system, the method or procedure for switching the CPU among multiple processes is called CPU scheduling or processor scheduling.

The CPU scheduler is a part of Operating system which is responsible for CPU scheduling. Whenever CPU becomes Idle, the CPU scheduler selects a process from the ready queue and send it to the CPU for execution.

For Example- Process A is in running state and need to perform I/O related task. process A doesnot need the CPU while performing the I/O task.

PROCESSOR SCHEDULAR

Processor scheduler change the state of process A from running to waiting state and enables process B to use the CPU.

For Example- In the process of CPU scheduling, two processes A and B share the CPU. When the process A uses the CPU, the process B perform I/O operation and when the CPU is accessed by process B, the process A perform I/O operation.

DULAR

Schedular is a part of Operating System which is responsible for entire process scheduling and transition of process state.

LEVELS/TYPES OF SCHEDULAR

<1> Long term schedular
<2> Medium term schedular
<3> Short term schedular.

<1> Long-term schedular ⇒ Long term schedular works with a new process which should be initiated in the system. Once initiated, this schedular takes the newly submitted jobs and convert them into processes which are further put in the ready queue for processing.

<2> Medium-term schedular ⇒ Medium term schedular decides whether to introduce a process to the waiting state from the running state or put up back a process from waiting state to the ready state.

<3> Short-term schedular ⇒ Schedular decision are made on which process in the ready state should be dispatched to the CPU/running state for execution.

This schedular is the most important and complex among all the scheduling levels.

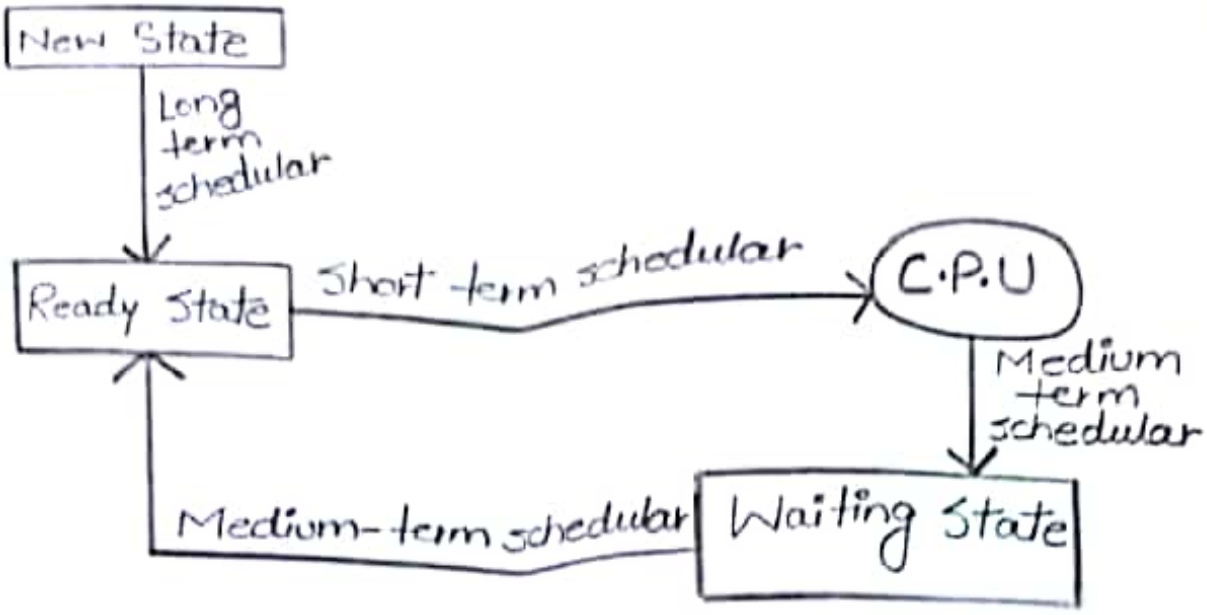


Fig. Schedular

SCHEDULING CRITERIA

Some performance criteria that are frequently used by schedular to maximize system performance are as follows:-

1) CPU Utilization ⇒ The key idea is that if the CPU is busy all the time, the utilization factor of all the components of the system will also be high.

2) Throughput ⇒ It refers to the amount of work completed in a unit of time.

One way to measure throughput is by means of the number of processes that are completed in a unit of time. The higher the number of processes, the more work apparently being done by the system.

naround Time ⇒ The amount of time to complish the execution of a process is known as turnaround time of the process. It is always minimized.

Turnaround time = Processing time/Execution time + Waiting time.

4.) Waiting Time ⇒ The amount of time a process waits in the ready queue to get a chance for execution is known as waiting time of the process.

Waiting time = Turnaround time - Processing/Execution time

5.) Response Time ⇒ It is the amount of time, it takes to start responding, not the time taken to output that response.

SCHEDULING ALGORITHMS

1. CPU scheduling deals with the problem of deciding which of the processes in the ready queue to be allocated the CPU.
2. There are several scheduling algorithm for taking decision.
3. A major division among scheduling algorithm is that whether they support preemptive or non-preemptive scheduling:-

Nonpreemptive scheduling ⇒

i) A scheduling discipline is nonpreemptive, if once a process has given to C.P.U, the C.P.U cannot be taken away from that process.

ii) Ex → DOS, Window 3.1 etc.

iii) The control of the C.P.U is in the hand of process

iv) In nonpreemptive, system jobs are made to wait by longer time, but the treatment of the process is fairer.

(B) Preemptive scheduling ⇒

i) A scheduling discipline is preemptive if the C.P.U can be taken away.

ii) The control of the C.P.U is in the hand of Operating System.

iii) Ex → Win' 95 upto is preemptive.

iv) Preemptive scheduling is useful in high priority process which require immediate response.

Several types of CPU scheduling algorithm are:-

(1) FCFS (First-Come, First-Served) Scheduling Algorithm.

(2) SJF (Shortest-Job-First) Scheduling Algorithm.

(3) SRTN (Shortest Remaining Time Next) Scheduling Algorithm.

(4) Priority scheduling

(5) RR (Round-Robin) Scheduling Algorithm

(First-Come, First-Served) Scheduling Algorithm

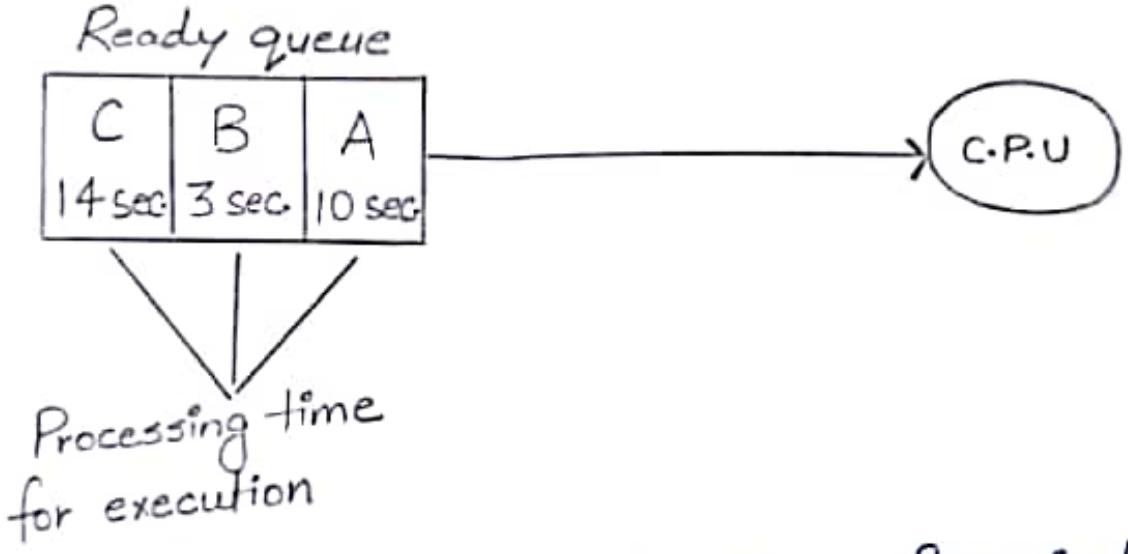
FCFS algorithm is the simplest and most state forward of all the CPU scheduling algorithm.

(ii) A process is allocated CPU time according to the arrival time of a process. The process that comes first is allocated the CPU first.

(iii) A FCFS scheduling is non-preemplive which usually results in poor performance.

(iv) This results in poor scheduling performance because a process requiring very short time waits for the completion of a process that requires a comparatively longer time.

Example-



Turnaround time of process A = Waiting time + Processing time
= 0 + 10
= 10 sec.

Turnaround time of process B = Waiting time + Processing time
= 10 + 3
= 13 sec.

Turnaround time of process C = Waiting time + Processing time
= 13 + 14
= 27 sec.

In FCFS scheduling, various processes are scheduled for execution regardless of their accepted execution time.

The process A comes first in the ready queue followed by the processes B and C.

The process are executed in a sequence. First, A is executed, then B and then C.

(2) SJF (Shortest - Job - First) Scheduling Algorithm

(i) It is faster than FCFS.

(ii) In SJF, the process with the least estimated execution time is selected from the ready queue for execution.

(iii) This algorithm is non-preemptive in nature.

(iv) It check always for shorted executed time.

(v) In SJF scheduling, various processes are scheduled for execution according to their estimated execution time.

Example

Ready Queue

Turnaround time of process B = Waiting time + Processing time
= 0 + 3
= 3 sec.

Turnaround time of process A = Waiting time + Processing time

= 3 + 10

= 13 sec.

Turnaround time of process C = Waiting time + Processing time

= 13 + 14

= 27 sec.

The process A, B, C requires 10, 3 and 14 sec. respectively.

The process B with minimum estimated execution time is scheduled for execution first followed by A and C.

(3) SRTN (Shortest Remaining Time Next) Scheduling Algorithm

(i) It is a scheduling discipline in which the next scheduling entity, a job or processes is on the basis of the shortest remaining execution time.

(ii) SRTN scheduling may be implemented either in the non-preemptive or preemptive variety.

(iii) The Non-preemptive of SRTN is called SJF (Shortest - Job - First) scheduling.

(iv) This scheduling searches the shorted remaining execution time. The difference between two cases lies in the condition that lead in evacuating the schedular.

out preemption, SRTN schedular control to e process.

In the preemption version, whenever an event occurs, that makes a new process to the ready queue and control directly to the operating system

(vii) SRTN is probably an optimal scheduling discipline in terms of minimizing the average waiting time of a given work load.

(viii) It is the preemption version of SJF (Shortest - Job - First) Scheduling with fixed interval time for each process.

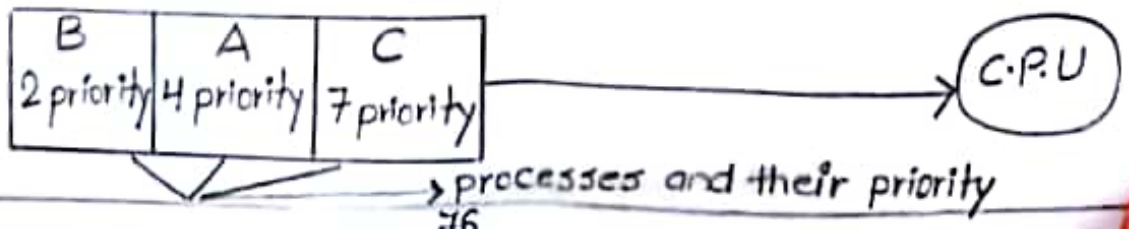
(4) Priority Scheduling

(i) In priority scheduling, a priority is associated with all processes. Processes are executed in a sequence according to their priority.

(ii) The CPU time is allocated to the process with the highest priority.

(iii) If the priority of two or more process are equal then the process that has been inserted first into the ready queue is selected for execution.

In other word, FCFS scheduling is performed.



priority scheduling algorithm, various processes be scheduled for execution according to their priority such as

A = 4 priority
B = 2 priority
C = 7 priority

(v.) The process C with the highest priority is selected for execution first followed by process A and B.

(5) RR (Round-Robin) Scheduling Algorithm

(i.) The RR scheduling algorithm implement the time sharing concept.

(ii.) Each process is allow to use C.P.U for a predefine unit of time known as time-slice or interval time with the help of a interval time hardware.

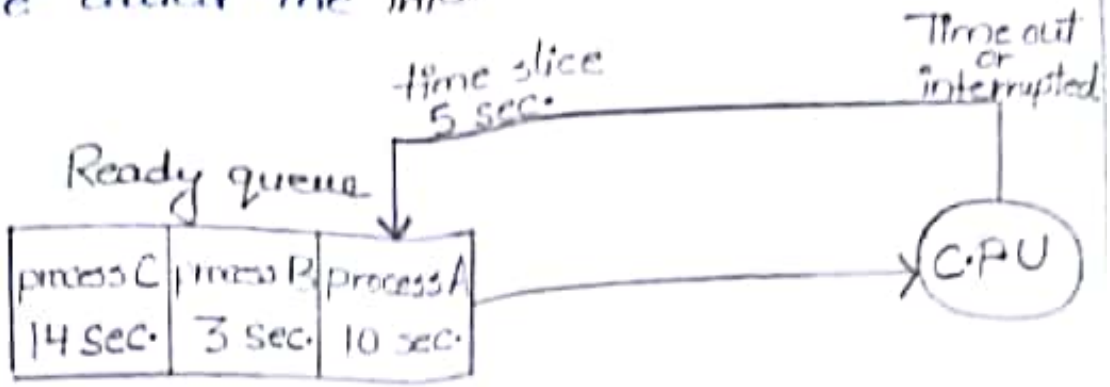
(iii.) If the running process is completed within the time slice, the next process in the ready queue is allocated to the C.P.U.

(iv.) In case, if the running process remain incomplete, it is stored back to the ready queue and the next process in the ready queue is selected for the execution.

(v.) In Round-Robin scheduling algorithm, the ready queue is a FCFS queue. When the time out occurs, the partly done processes are inserted at the end of the ready queue.

Is similar to FCFS but preemptive in nature.
Timer is used to mark the end of a time slice called the interval timer.

Example-



Processing ↓

process C	process C	process A	process C	process B	process A
4 sec	5 sec	5 sec.	5 sec	3 sec	5 sec

Waiting time of process A = 8 sec
Processing time of process A = 10 sec
Turnaround time of process A = Waiting time + Processing time
= 8 + 10
= 18 sec.

Waiting time of process B = 5 sec.
Processing time of process B = 3 sec.
Turnaround time of process B = Waiting time + Processing time
= 5 + 3
= 8 sec.

g time of process C = 13 sec.

essing time of process C = 14 sec.

ound time of process C = Waiting time + Processing time

= 13 + 14

= 27 sec.

In figure, there are three process A, B and C. Each process have a time slice 5 sec. The operating system transfer the control of the C.P.U to the next process after 5 sec. automatically.

INTERVAL TIMER

Timer interruption is a technique that is closely related to preemption. When a process gets the CPU, a timer may be set to a specified interval. If the process is still using the CPU at the end of the interval, then it is preempted. Both timer interruption and preemption force a process to yield the CPU before its CPU burst is complete.

MANAGEMENT

e management system of an operating system takes care of file related activities such as accessing, naming, sharing and protection of file.

FILES

(i) A file is the smallest unit containing the information on a topic in a PC.

(ii) A file is a collection of related information.

(iii) Each file has its name, data and its attributes.

(iv) A file name is uniquely identifier for the user to use it and access it.

(v) Data is the contents of a file.

(vi) File is always stored in the secondary storage device such as harddisk.

Example- ABC.txt
xyz.bmp

DIRECTORY

(i) containers of the files.

(ii) Whenever we saved a typed letter in our PC and give it a nature, operating system (i.e DOS) will save it on our hard disk or secondary storage in the form of electronic file into an electronic folder or drawer called directory.

(iii) It works as a electronic drawer for the files.

TYPES OF FILES

File name extension (i.e the three characters appearing after the dots (.)) used in a file name to show the

e of file. These are described below:-

- Text file(.txt)
- Backup file(.bak)
- Help file(.hlp)
- Document file(.doc)
- Graphics file(.bmp)
- Ms access file(.mdb)
- Powerpoint file(.ppt)
- Excel file(.xls)
- C file(.c)
- C++ file(.cpp)

- Text file(.txt)

A text file is a file which is created so that it can be used by any word processing program or software.

eg. XYZ.txt

- Backup file(.bak)

Extension name of a backup file is .bak. This indicates that the file contents of the named file are created as a separate copy for safety purpose. In case, the original file is deleted or get corrupt, we can open the backup file.

eg. XYZ.bak

- Help file(.hlp)

In every software package, for example MS-Office or Word Star, there is a help file which contain information to help the user for learning specific commands.

eg. ws.hlp

- Document file(.doc)

Every word processor create a file which contain some special symbols along with the typed text. It contains left space, right space, top margin, formatting etc.

eg. Kriti.doc

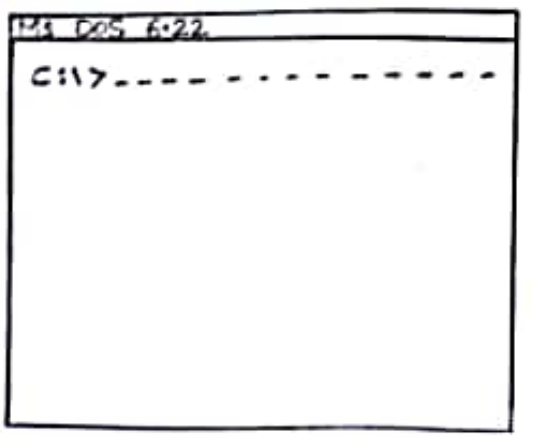
aphics file(.bmp)
hen we draw graphics or pictures using special graphic software such as MS-Paint, Corel draw etc. we use graphics file.
eg. Kriti.bmp

There are several other types of file available for the user for different task such as

- Ms PowerPoint (.ppt) → for the presentation of document
- Ms Excel (.xls) → used as a electronic separate sheet.
- C file(.c)
- C++ file(.cpp)
- Ms Access file(.mdb)
- foxfro file (.dbf)

Ms DOS (Microsoft Disk Operating System)

(i) It is a command line user interface.
(ii) Ms DOS 1.0 was released in 1981 from IBM computer.
(iii) The latest version of MsDos is MsDos 6.22 comes in 1994.
(iv) It still can be accessed from every version of Microsoft Window by clicking start → Run and typing "command" or by typing "cmd" in Windows NT, Windows 2000, Win' XP etc.



Interpreter work in it ←

MANDS

instructions processed by DOS command interpreter.

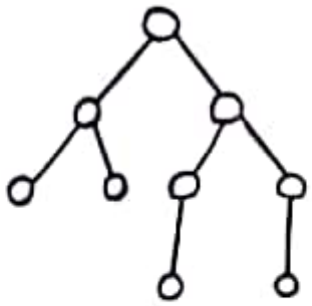
There are two types of command
(i) Internal command
(ii) External command

(i) Internal command
Internal command in DOS are such programs which get loaded in the memory of a PC automatically along with DOS at the time of booting.
Ex- dir, cd, md, Type, Del, Ren, edit, copy etc.

(ii) External command
External command files that do reside on disk.
Ex- command.com
Format.com
Xcopy etc.

PATH

A list of directories that DOS must go through to find a directory or files.



There are two types of path
(i) Absolute path
(ii) Relative path

(i) Absolute path → A path that start from the root directory is called absolute path.

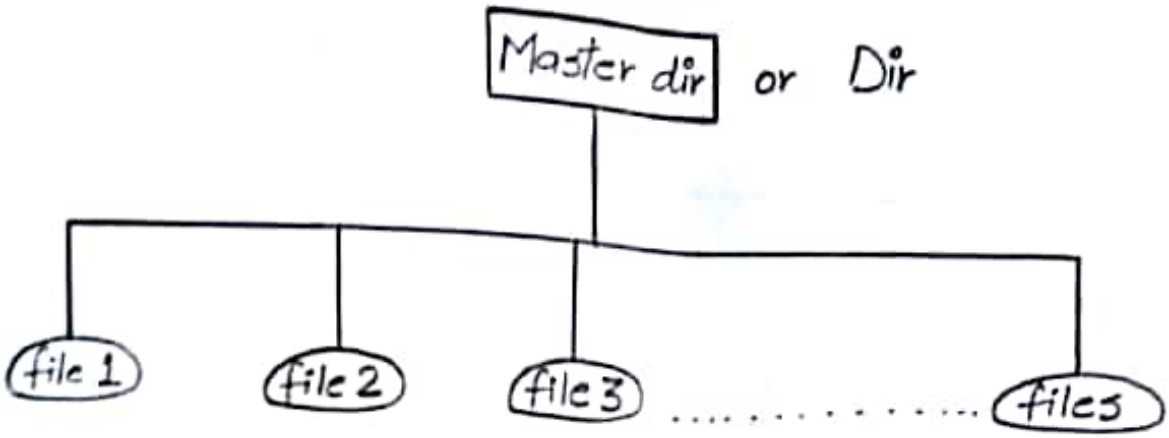
(ii) Relative path → A path that start from the drive's current directory is called relative path.

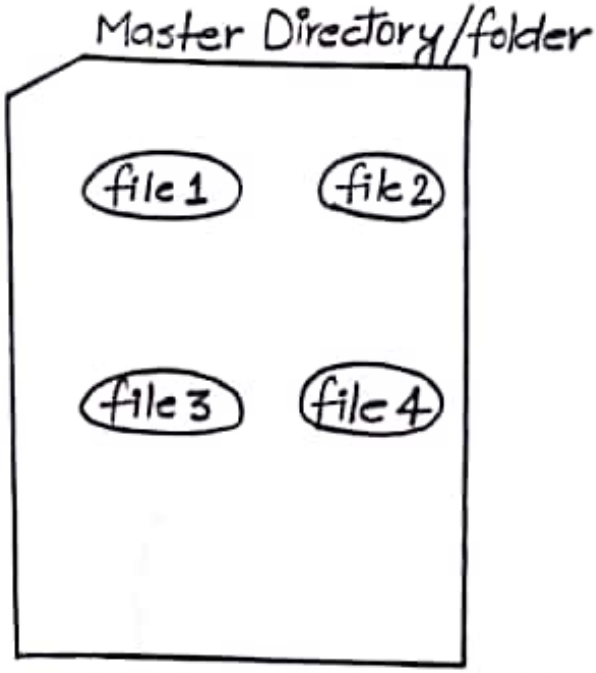
ctory Structure

Three types of Directory structure are

(i) Single level Directory
(ii) Two level Directory
(iii) Hierarchical Directory

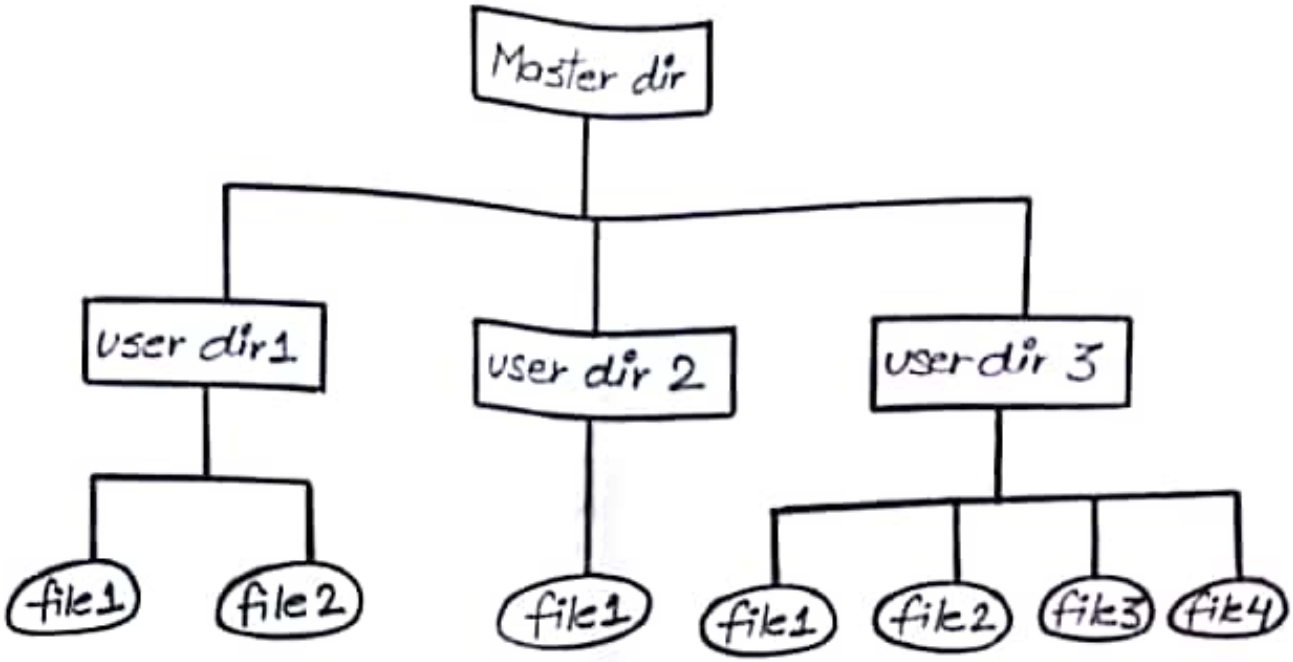
(i) Single level Directory

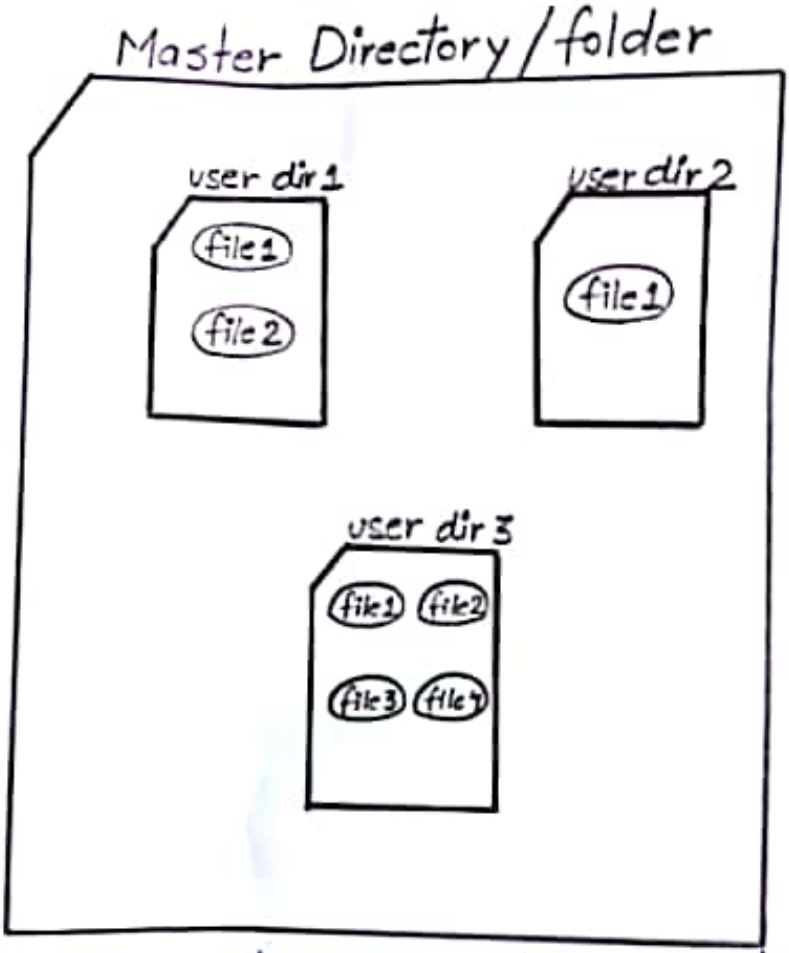




In single level directory structure, there is only one directory which contains certain number of files.

(ii) Two level Directory

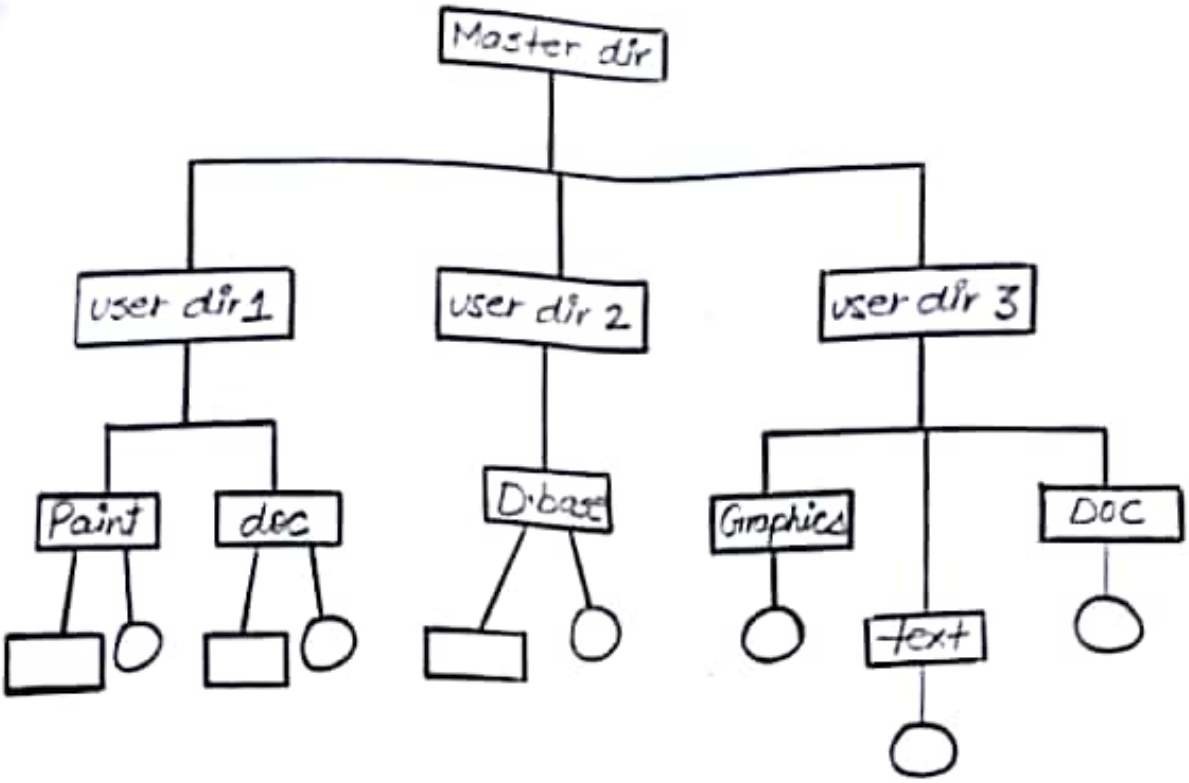


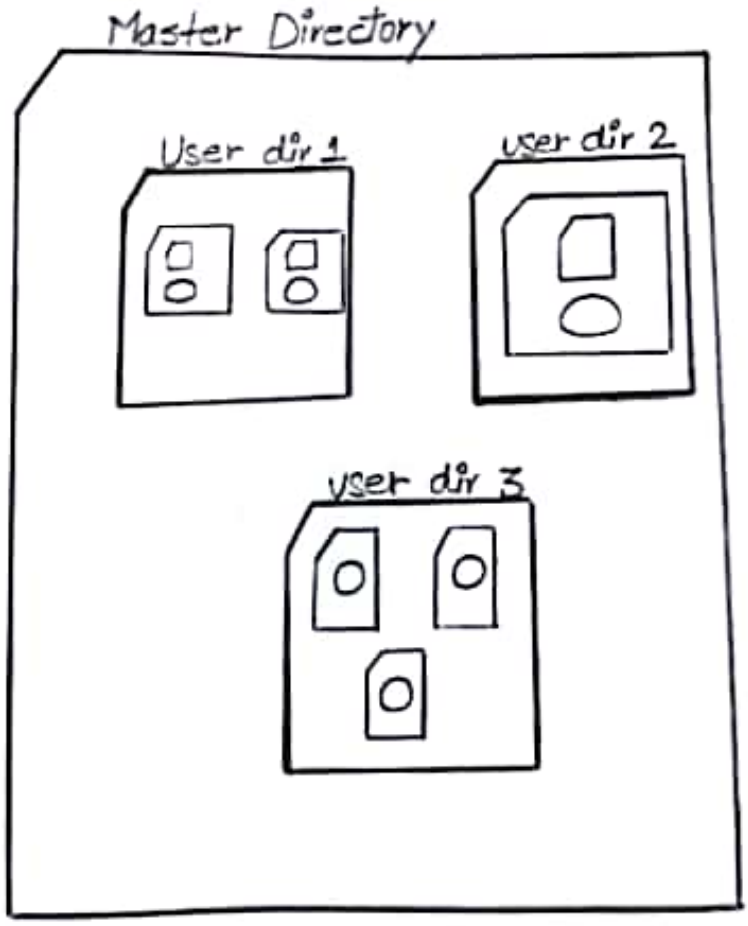


1) In two level directory structure, the master directory contains number of subdirectories.

2) In two level directory structure, the root directory contains number of branches with files as leaves.

Hierarchical Directory





The hierarchical directory structure works like a tree structure which has roots, branches and subbranches with leaves.

2) In case of hierarchical directory structure, the subdirectory of master directory also contain the directories.

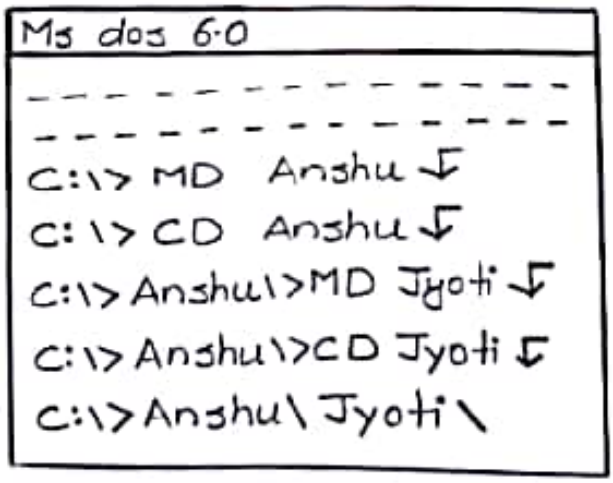
3) After two level directory structure, if any directory level is created then it is said to be hierarchical structure.

File Operation

The user manipulate file, various operation or commands. These operation vary from system to system and user operation on a file are by means of command language. In a DOS operating system, we use some operation such as creating a file, renaming a file, deleting a file etc.

File Management Commands

Start → Run Typed → cmd/command



directory बनाने
enter directory

MD (Make Directory)

It is used to make directory.

C:\> MD kriti ↵

or

MD Dirname

CD (Change Directory)

(i) It is used to go to current directory.

Ex- C:\> CD Dirname ↵

or

C:\> CD kriti ↵

(ii) It is used to return that directory.

RD (Remove Directory)

It is used to remove that directory.

Ex- C:\> RD Dirname ↵

or

C:\> RD kriti ↵

Cls

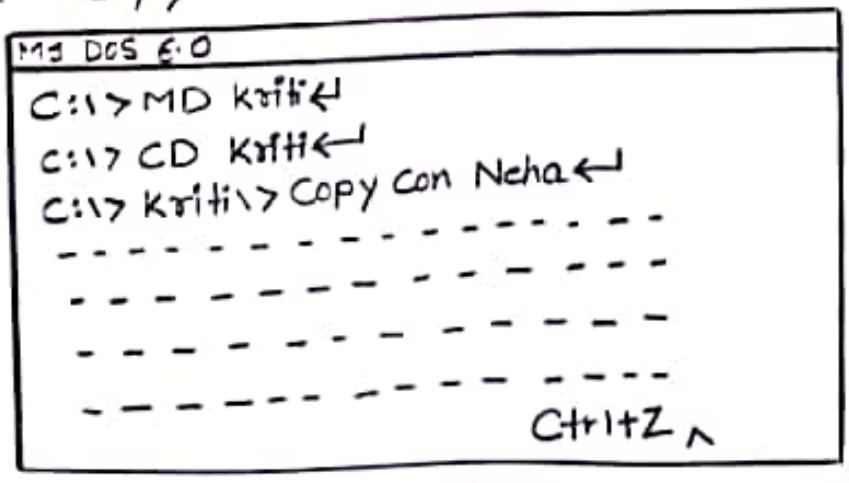
It is used to clear the screen

C:\> CLS ↵

Copy Con

It is used to create a file.

Syntax- Copy Con Filename



ii+z → It is used to store the file.

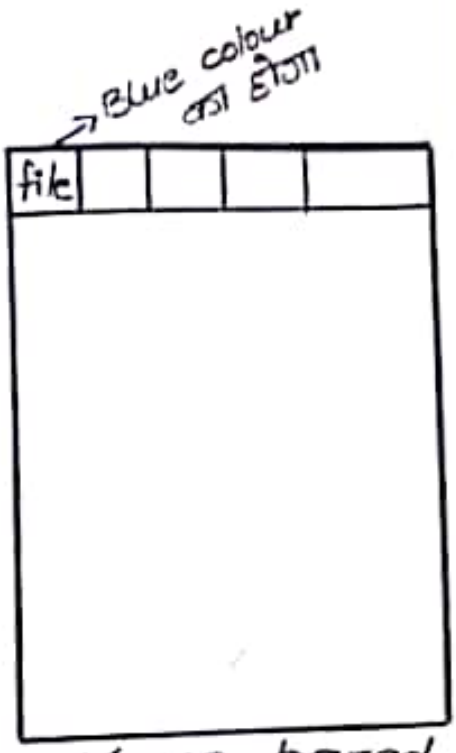
Edit command

(i) It is a menudriven text editor.

(ii) It is used to make corrections in a file.

Syntax edit filename ↵

It is not mouse-based



Mouse-based

Ren command

It is used to rename the file.

Syntax Ren oldfilename newfilename

Example C:\>kriti\>Ren Neha Priya ↵

Copy command

copy command copies the contents of a file (c-alled the source file) and produces a new file (called the destination, or target file).

Syntax

copy filename1 filename2

Example C:\>kriti\>Copy Priya Ruchi ↵

command

It is used to see the contents of file.

Syntax - Type filename↵

Example - Type Priya↵

Type Ruchi↵

Move command

Move command moves one or more files to another drive, directory or drive and directory.

Syntax move source_filename destination_filename

Example move Ruchi Pooja↵

Del/Delete/Erase command

It is used to delete a file from the disk default to drive.

Syntax delete filename↵

Example

C:\>Kriti\> delete Pooja↵

Undelete command

undelete file are used only for external command.

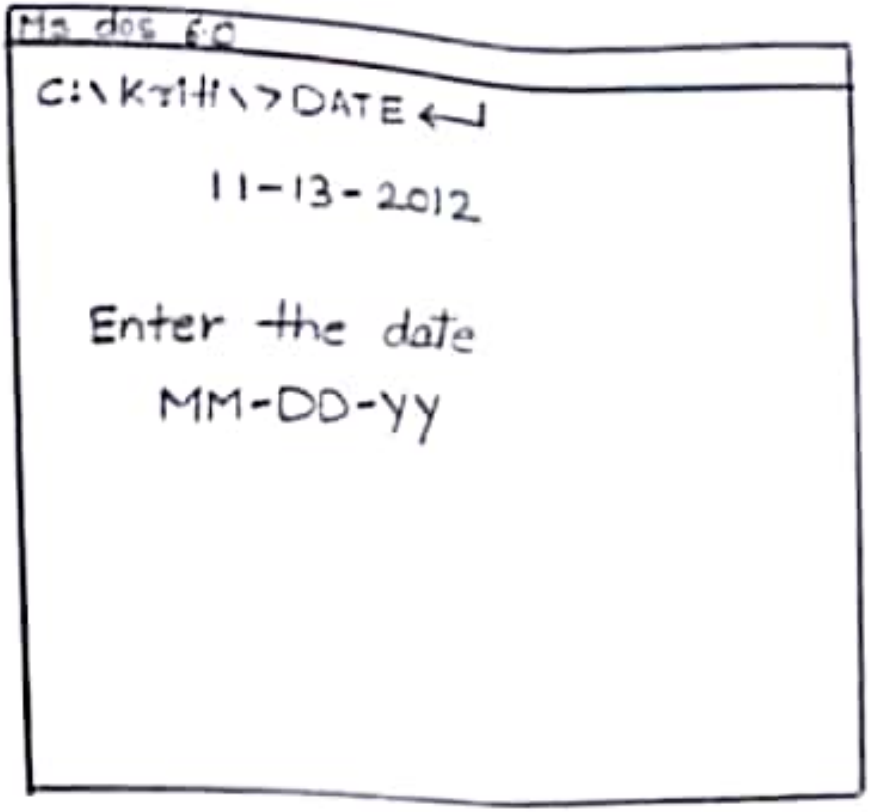
Syntax undelete filename↵

Example undelete Pooja↵

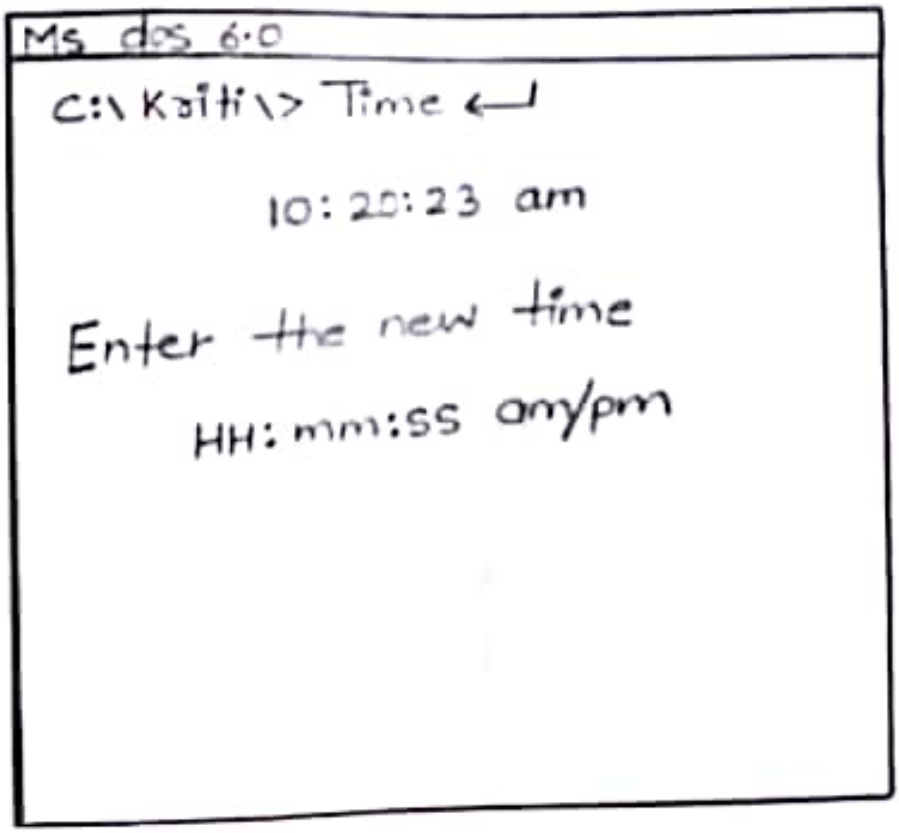
It is used to review the file if it is deleting.

Date command

It is used to see the date and also change the date.



Time command

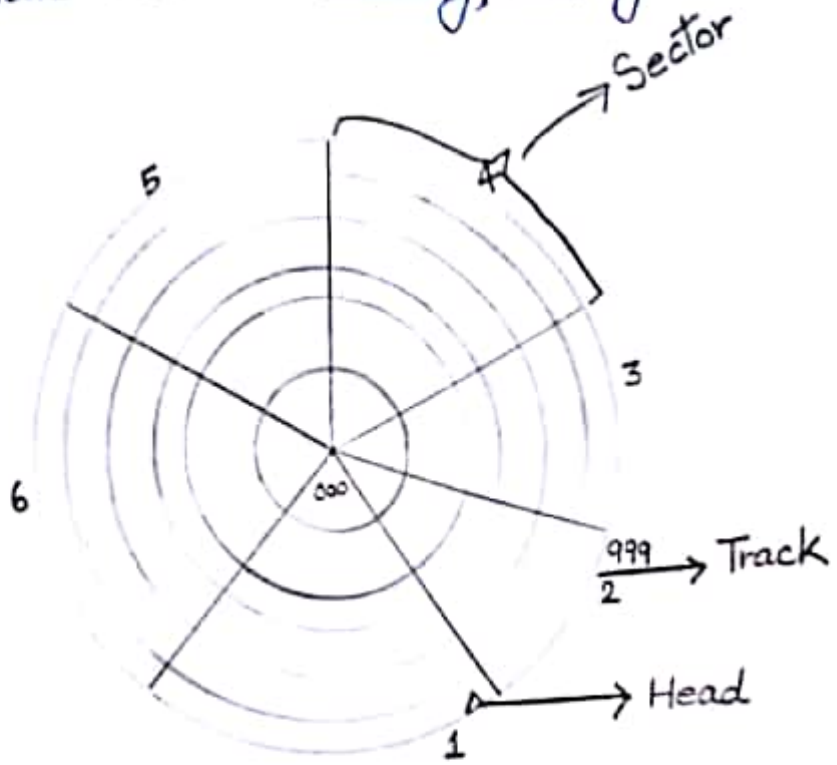


ORGANISATION

Disk is the secondary storage media i.e. data store and external devices such as hard disk, magnetic disk, floppy disk, CD-ROM where the data is stored in the sector and provides to user program via I/O Request.

DISK MANAGEMENT

Disk management is an important function of operating system. These include disk scheduling, management of swap area etc.



Track

For data recording, the surface of a disk is divided into a number of concentric circle called track.

Sector

Each track is further subdivided into number of segments called sector.

A disk is a device used for mass storage of data needed for direct access. This way, a disk can be used to retrieve any record of data at very high speed.

Seek Time

It is the time taken from read/write over requested track.

Seek Time = Time of one track to the Time of another track

Latency Time

It is the time taken for the required sector to come under the head.

Latency Time = Time taken from one particular sector to time taken from same particular sector.

NOTE: Another name of Latency time is Search time.

If the head is fixed, seek time = 0

∴ Access time = Latency time

i.e. $A \cdot T = L \cdot T$

Movable head, $A \cdot T = S \cdot T + L \cdot T$

DISK SCHEDULING

(17) When more than one process is running concurrently, several requests are generated by these processes for I/O operation. These are issued to operating system at the same time.

erating system maintain a queue for this purpose, taking out the I/O request from the queue and processing it one by one.

The queue needs to be scheduled. Various scheduling are given to perform this function -

(a) FIFO (First - Come - First - Served) Scheduling.
(b) SSTF (Shortest - Seek - Time - First) Scheduling.
(c) SCAN (elevator algorithm) Scheduling.
(d) C - SCAN (Circular SCAN) Scheduling.

(a) FIFO (First - Come - First - Served) Scheduling

As the name suggest, the request that arises first will be served first. Being the simplest form of disk scheduling, it processes the request in sequential order i.e. the order in which the I/O request arrive.

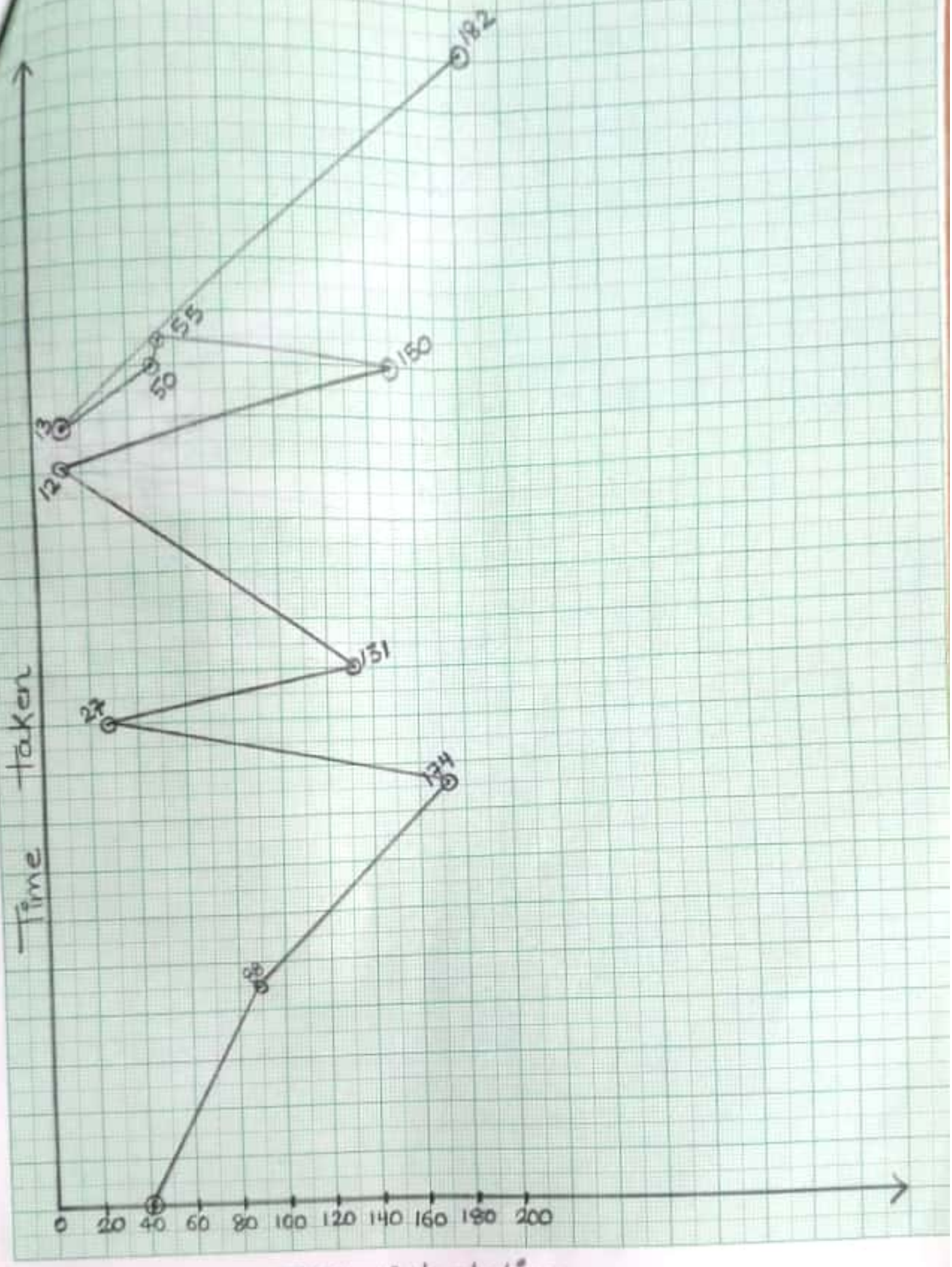
For Example - A disk of size 200 track receives the request for the block of I/O operation. Number of the blocks/tracks on the disk are 88, 174, 27, 131, 12, 150, 55, 50, 13 and 182.

Disk arm is initially at track 40. It will first move from 40 to 88, then 88 to 174, then 174 to 27 and so on..... covering 948 track in total.

disk is on track 40

Requested Track	Track covered
88	88-40 = 48
174	174-88 = 86
27	174-27 = 147
131	131-27 = 104
12	131-12 = 119
150	150-12 = 138
55	150-55 = 95
50	55-50 = 5
13	50-13 = 37
182	182-13 = 169
	Total no° of track covered = 948

(Shortest - Seek - Time - First) Scheduling



FCFS Scheduling

(Shortest - Seek - Time - First) Scheduling

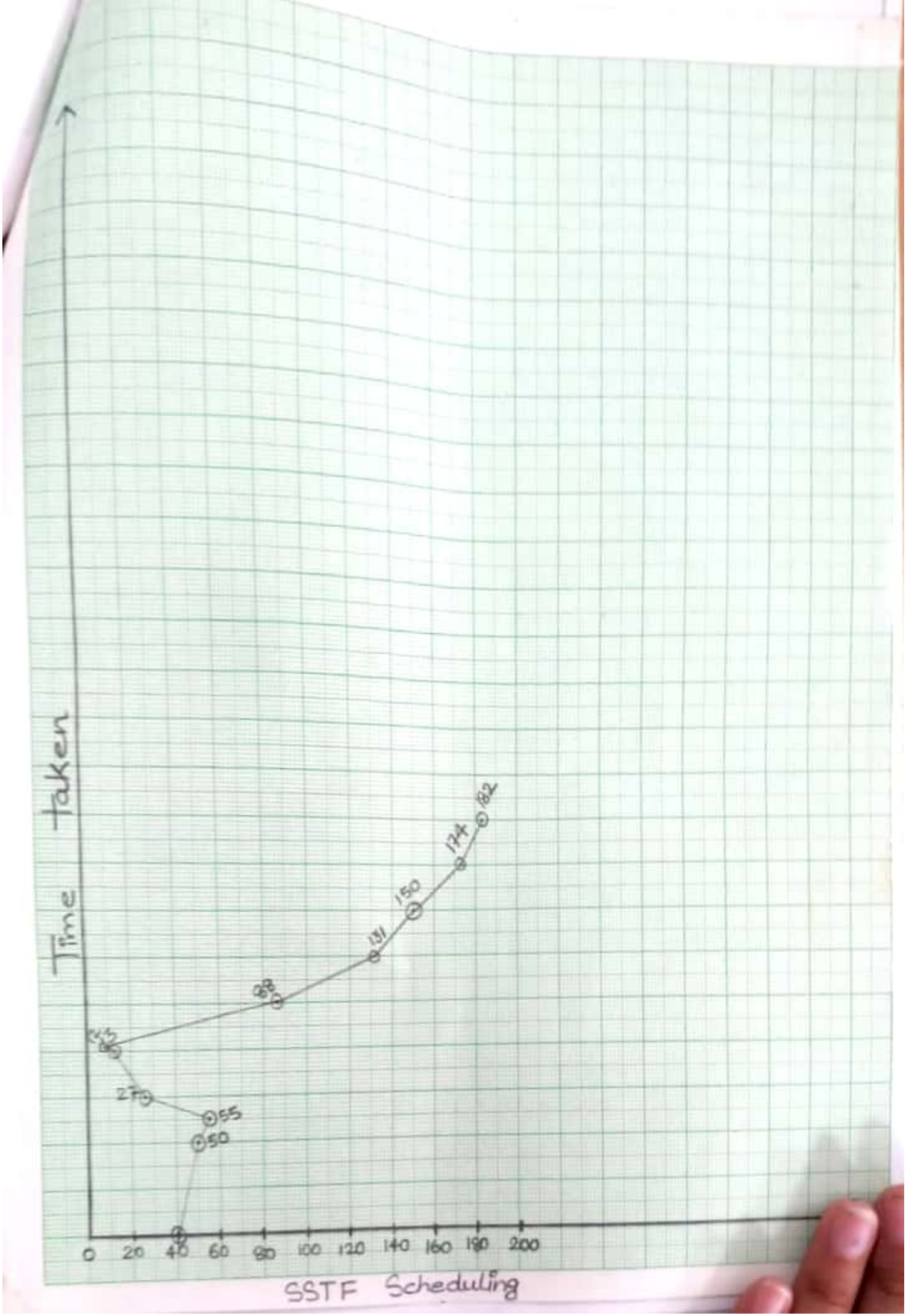
In this scheduling, disk arm (head) will those I/O operation that require minimum track to be covered. Only that I/O request is served first i.e. close to the disk arm.

SSTF policy helps in achieving the minimum seek time by serving request that require minimum seek time. That means further following same mechanism for - 88, 174, 27, 131, 12, 150, 55, 50, 13 and 182.

Disk arm is initially at track 40. It will first move from 40 to 50, then 50 to 55, then 55 to 27, then 27 to 13, then 13 to 12, then 12 to 88, then 88 to 131 and so on..... covering 228 track in total.

disk is on track 40

Requested Track	Track covered
88	40 - 50 = 10
174	50 - 55 = 5
27	55 - 27 = 28
131	27 - 13 = 14
12	13 - 12 = 1
150	12 - 88 = 76
55	88 - 131 = 43
50	131 - 150 = 19
13	150 - 174 = 24
182	174 - 182 = 8
	Total number of track covered = 228



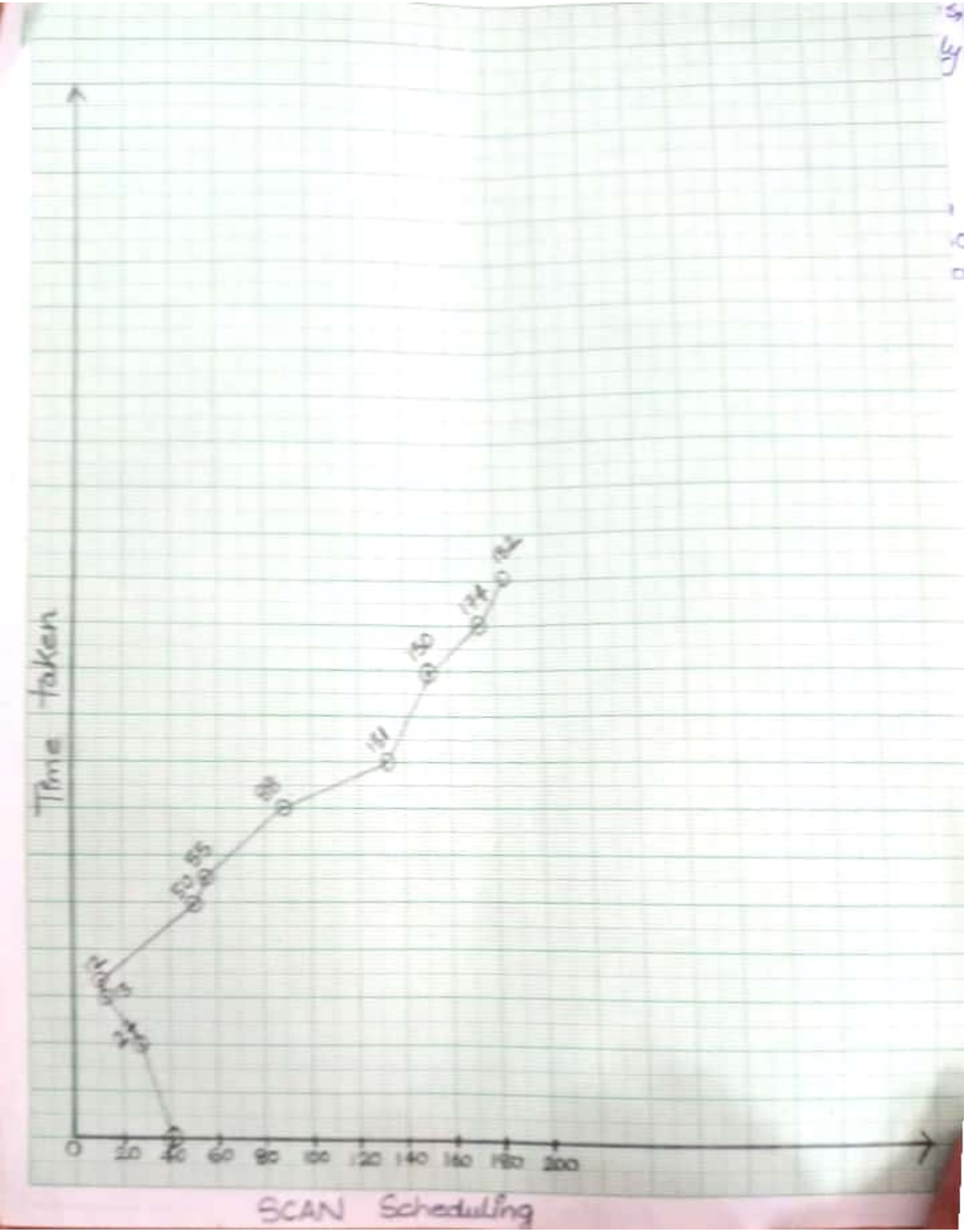
AN Scheduling

In SCAN Scheduling, disk arm is req-ed to move from beginning of the disk and then owards the end of the disk satisfying the entire request that exist in the route of the disk arm. SCAN policy behaves like SSTF (Shortest - Seek - Time - First) if it starts moving fro-m the lower track number.

Disk arm is initially at track 40. It will first move from 40 to 27, then 27 to 13, then 13 to 12, then 12 to 50, then 50 to 55, then 55 to 88, then 88 to 131, then 131 to 150, then 150 to 174 and finally 174 to 182 covering 198 track in total.

disk is on track 40

Requested Track	Track covered
88	40 - 27 = 13
174	27 - 13 = 14
27	13 - 12 = 1
131	12 - 50 = 38
12	50 - 55 = 5
150	55 - 88 = 33
55	88 - 131 = 43
50	131 - 150 = 19
13	150 - 174 = 24
182	174 - 182 = 8
	Total number of track covered = 198



d) C-SCAN (Circular SCAN) Scheduling

It is an important improvement over scan scheduling. In this policy, disk arm start from end where more number of request are present, while moving towards its end satisfies all the request that come in the route. When the disk arm reaches it quickly return to reach its end, then it quickly reaches to the other end without fulfilling any request in the way. That are 88, 174, 27, 131, 12, 150, 55, 50, 13, 182.

Disk arm is initially at track 40. It will first move from 40 to 50, 50 to 55, then 55 to 88, then 88 to 131, then 131 to 150, then 150 to 174, then 174 to 182, then 182 to 12, then 12 to 13 and finally 13 to 27 covering 227 track in total.

disk is on track 40

Requested track	Track covered
88	40 - 50 = 10
174	50 - 55 = 5
27	55 - 88 = 33
131	88 - 131 = 43
12	131 - 150 = 19
150	150 - 174 = 24
55	174 - 182 = 8
50	182 - 12 = 70
13	12 - 13 = 1
182	27 - 13 = 14
	Total number of track covered = 227

STORAGE MANAGEMENT

(1) Other than the CPU, main memory is an important resource of a computer system which must be properly managed for overall system management.

(2) The memory management of an operating system takes care of its requirement.

(3) Its job is to keep track of which part of memory are in use and which part are not in use.

(4) To allocate memory to processes which they need it and deallocate them they are done.

LOGICAL ADDRESS (VIRTUAL ADDRESS)

The address generated by the CPU is known as logical or virtual address.

OR

The address which are used in the program are called logical address.

PHYSICAL ADDRESS

The real or physical is the actual main memory provided in the systems, it is directly addressed by CPU called physical address

OR

Any location in the main memory is uniquely identified by an address. This is called physical address of main memory or RAM.

GICAL ADDRESS SPACE

The set of all logical addresses is called logical address space.

PHYSICAL ADDRESS SPACE

The set of all physical addresses is called physical adress space.

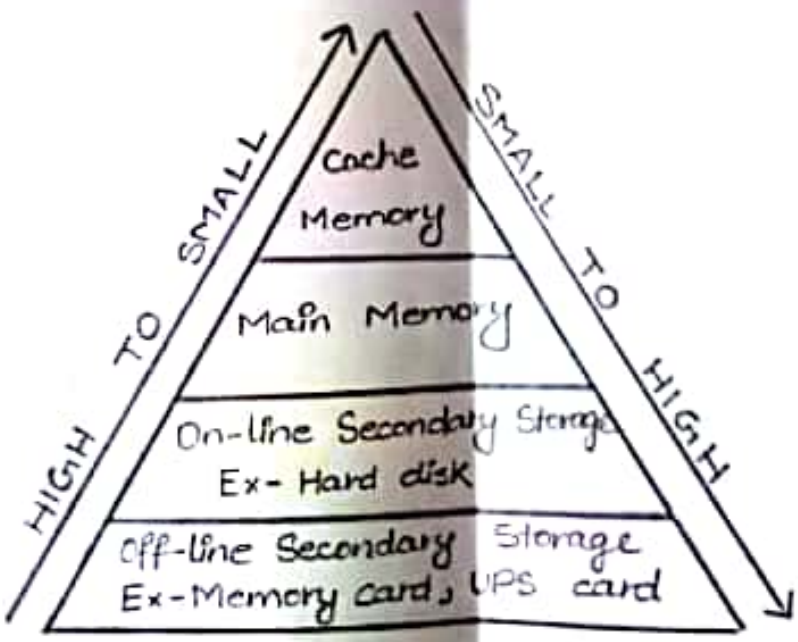
MEMORY MAPPING

The procedure for conversion of logical to physical address is called memory mapping.

MMU (MEMORY MANAGEMENT UNIT)

The hardware unit that convert a logical address to physical adress is called MMU (Memory Management Unit).

STORAGE HIERARCHY



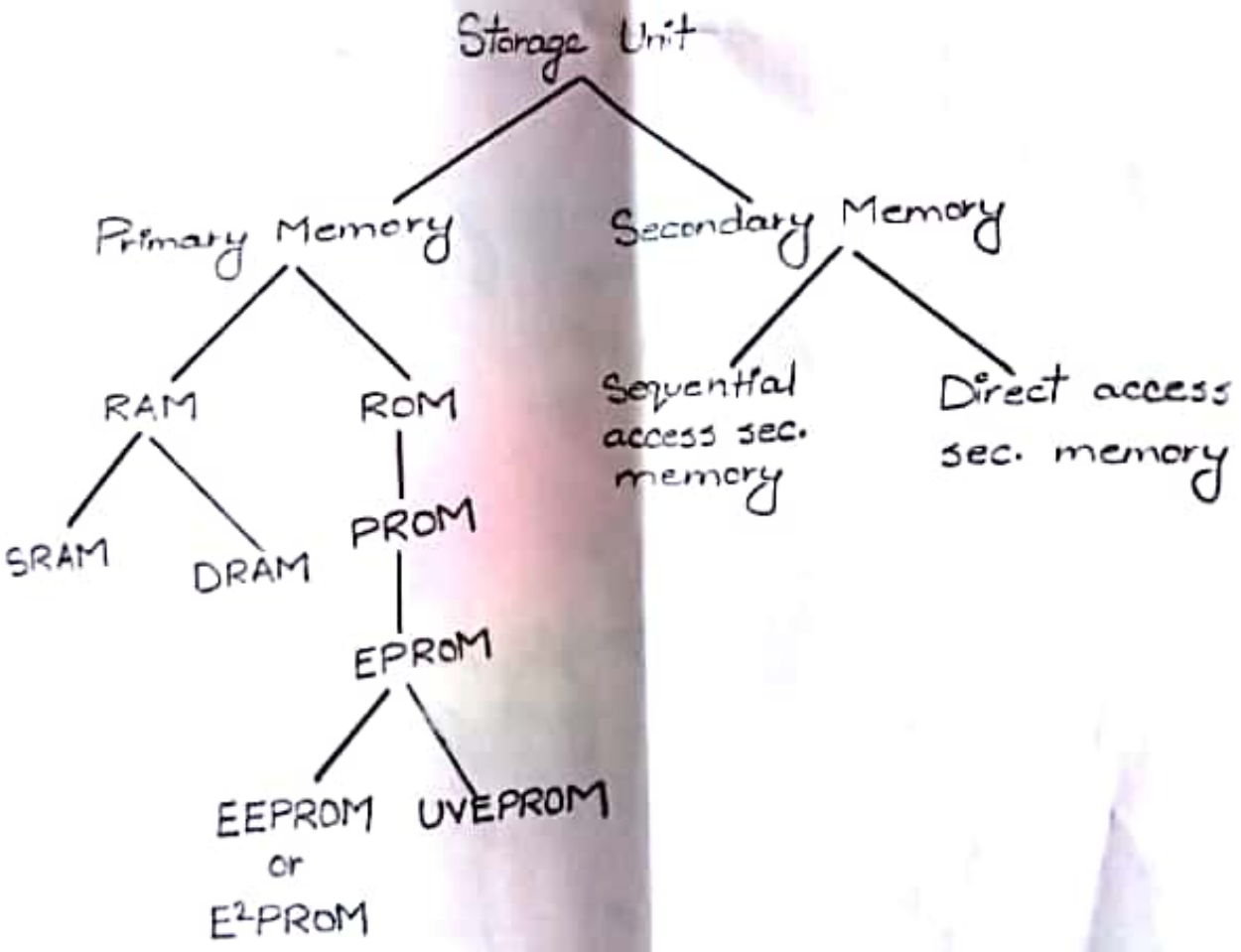
It includes cache memory, main memory and secondary storage.

2) As we move up the pyramid, we encounter storage element which has faster access time, higher cost per bit store and smaller capacity.

3) As we move down the pyramid, we encounter storage element, which have slower access time, lower cost per bit store and larger capacity.

4) Cache Memory → Main Memory → On-line Secondary storage → Off-line Secondary storage.

STORAGE ORGANISATION



STORAGE UNIT

It is used to store the data.

It is also the part of hardware.

A computer uses two types of storage:-

a) Main Memory, and

b) Secondary Memory

a) Main Memory

Main Memory temporarily stored instruction or data to be executed by the computer. The Primary storage hold information only when the computer system is on. As soon as the computer system switched off or reset, the information held in the primary storage is disappeared.

b) Secondary Memory

It is used to take care the limitation of primary memory. It is the permanent storage device which retain(reserve) information even when the computer system is switched off or reset.

Primary Memory is of two types

I. RAM (Random Access Memory)

II. ROM (Read Only Memory)

I. RAM

RAM is temporarily r/w memory of a computer system and faster than secondary memory.

ROM

Programme held in ROM are called firmware. They are stored permanently and are read for use when the computer is switched on. Users cannot write into a ROM. Its contents are written at manufacturing time i.e. POST (Power on Self Testing). ROM store fonctions such as square root, cosine, exp, log, sine etc.

PROM (Programmable Read Only Memory)

Programme held in ROM are called PROM.

EPROM (Erasable Programmable Read Only Memory)

Once a information stored in ROM or PROM, chip cannot be altered or erased. However, another type of erasable programmable read only memory is developed which erase the stored information and restore the new information or programme.

There are two types of EPROM:-

i) EEPROM or E^2PROM

ii) UVEPROM

i) EEPROM (Electrically Erasable Programmable Read Only Memory)

Erased with the help of high voltage electricity.

ii) UVEPROM (Ultra-violet Erasable Programmable Read Only Memory)

Erased by ultra-violet light.

RAM is of two types:
I. SRAM (Static RAM)
II. DRAM (Dynamic RAM)

Static RAM	Dynamic RAM
(i) Static RAMs retain stored information only as long as the power supply is on.	(i) Dynamic RAM loses its stored information in a very few seconds (nanoseconds) even though the power supply is on.
(ii) It has very high speed.	(ii) It has low speed.
(iii) It consume more power.	(iii) It consume less power.
(iv) It is costlier	(iv) It is cheaper

b) Secondary Memory

It is of two types:

I. Sequential access secondary memory.

II. Direct access secondary memory.

I. Sequential access Secondary memory:-

A sequential access storage device is one in which the arrival at the location desire may be proceeded sequentially through other location.

Ex- Magnetic Tape.

II. Direct access Secondary memory:-

A direct access storage device is one in which the arrival is literally available at random i.e. it is available in any order.

Ex- • Optical disk such as CD-ROM, DVD, WORM etc.

agnetic disk such as Floppy disk, Hard disk etc.
Removable disk such as pen-drive, memory card etc.

MEMORY ALLOCATION

(1) The procedure of assigning memory to a process is called memory allocation.

(2) There are two types of memory allocation:
- → Contiguous memory allocation
- → Non-contiguous memory allocation.

(3) In contiguous memory allocation, process is loaded in one or more contiguous blocks of memory. It is sufficient that contiguous memory allocation loaded in contiguous form. Its (process) are not loaded in scattered or fragmented order.

(4)

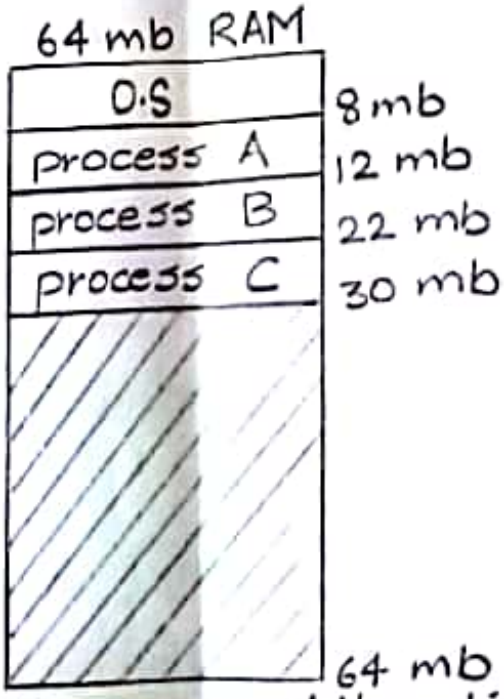


Fig. Contiguous Memory Allocation

In the above figure, main memory as 64 mb and the processes are loaded in contiguous form (not in scattered or fragmented order.)

In non-contiguous memory allocation, a process can be loaded in scattered way. In other word, a process may not be loaded in contiguous memory allocation.

(6)

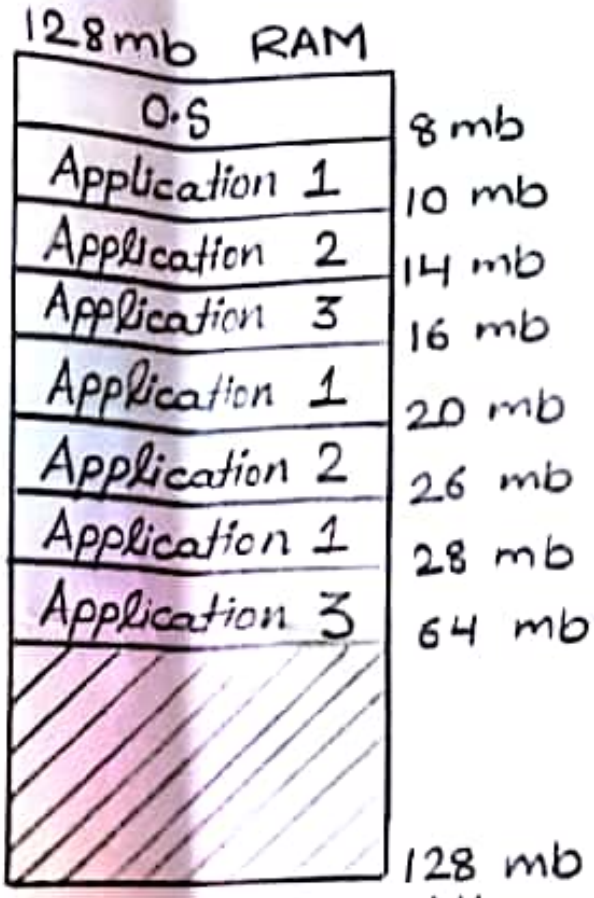


Fig. Non-contiguous Memory Allocation

In the above figure, main memory as 128 mb and the processes are loaded in scattered or fragmented order.

Single User Contiguous Storage Allocation

(7) The single user contiguous storage allocation is used in uniprogramming system (in which one job is processed by the system at a time) and multitasking system (in which more than one job is processed by the system at a time.)

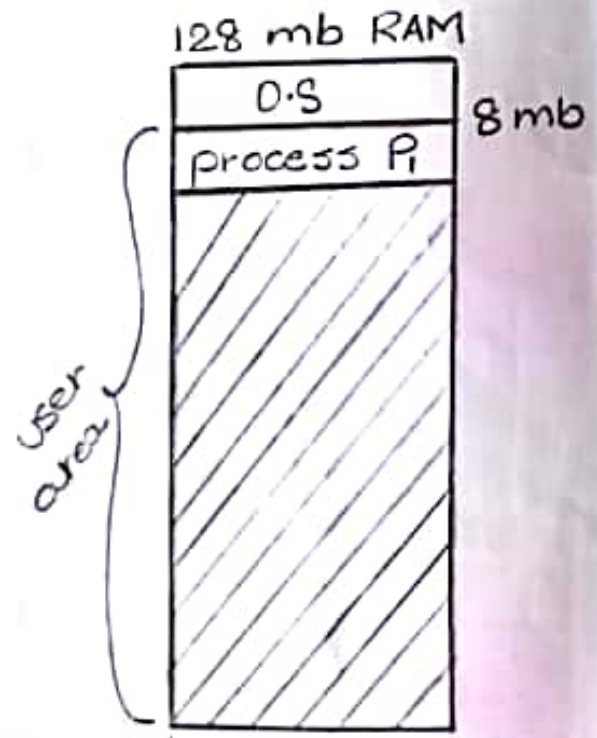
operating system load and all the system resource are exclusively available for the single user.

(3) In this memory management strategy, the operating system resides in one part of the memory and an entire remaining part of the memory is available for user. The currently active user process.

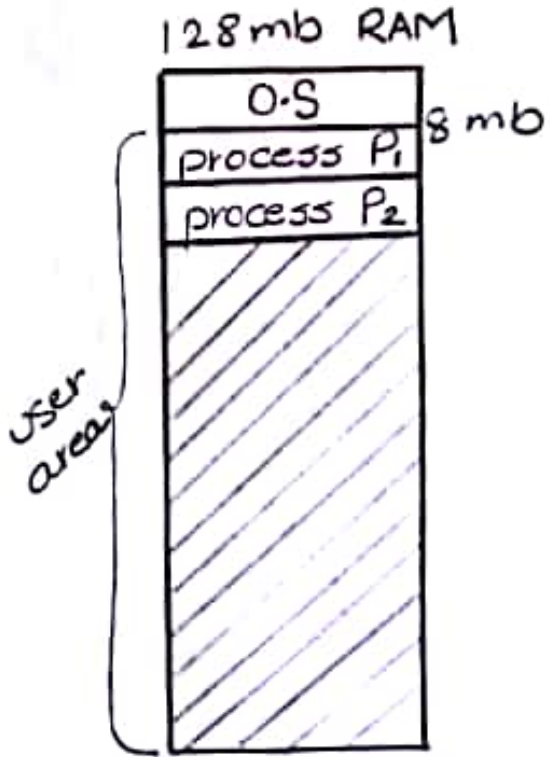
(4) In this scheme, operating system load the program to be executed from the disk into the user area of the memory and executes it.

(5) When the processes finish, the operating system cleans up the user area of the memory and load the next program to be executed.

(6) The two figure shows single user contiguous memory allocation



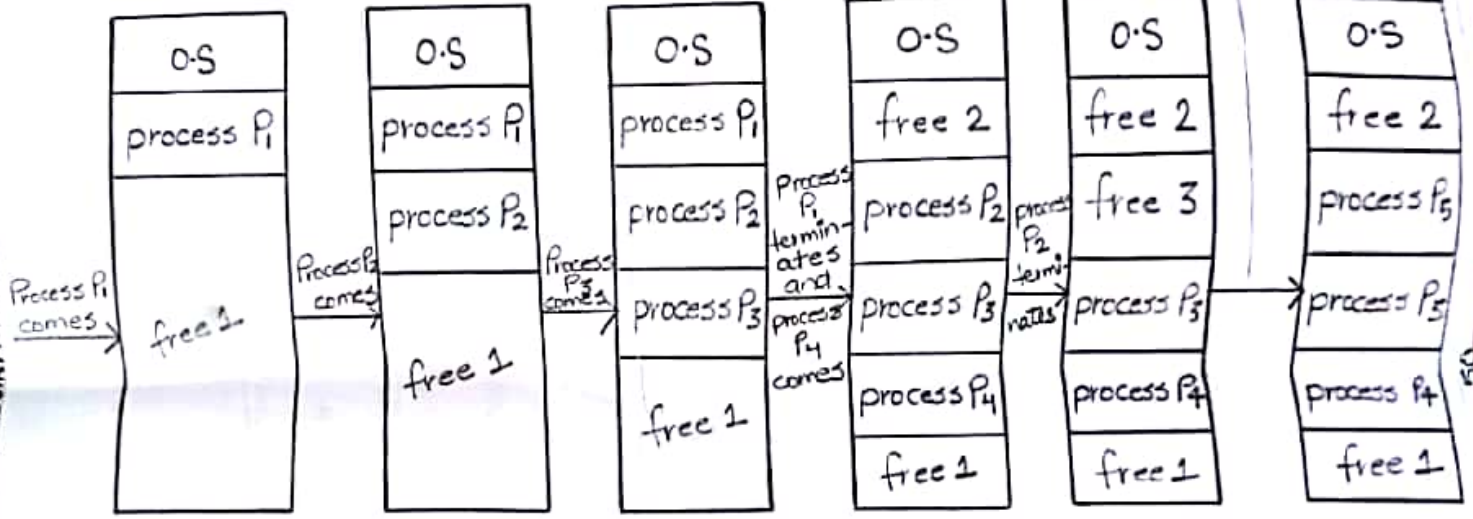
Single tasking/uniprogramming
fig.1



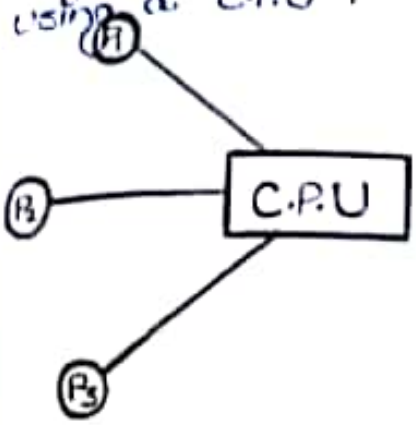
Multitasking
fig.2

Variable Portion Multiprogramming

process P_5 comes but not fit in free 1 then it goes to free 2 but not fit in free 2 then it goes to free 3 and fit in free 3.



Multiprogramming operating system allows multiple users to execute multiple programs using a C.P.U.



(2) It is the memory management scheme in which all memory is available for the user space.

(3) When a process arrives, only as much memory as needed by it is allocated to its keeping. The rest available are to satisfy future request.

(4) Operating System maintain a table track of which part of memory keep to use and free to be occupied.

(5) When a new process arrives, the operating system searches for a free block i.e. large enough for this process.

(6) If the free block is too large, it is split into two parts. One part is used for memory requirement of the process and is allocated to the process. The other part contain the remaining memory for the process.

When a process terminates, it releases the portion allocated to it.

UAL MEMORY

Conventional memory management scheme discussed until now, suffer from the following two main limitation. These are:

(i) A process cannot be loaded and has keep waiting for its executing to start until sufficient free memory for loading the entire process scheme become available. This may delay process's turnaround time.

(ii) A process cannot be loaded in a system whose memory delay a process's turnaround time.

(2) Three - basic concept for the realisation of virtual memory are:

(a) On-line Secondary storage.

(b) Swapping

(c) Demand Paging

(a) On-Line Secondary Storage ⇒ It is a secondary storage device i.e. hard disk whose capacity is much larger than the main memory and which is always kept on-line to the system. High speed disk storage is used for this purpose.

(b) Swapping ⇒ Swapping is the process of transferring a block of data from on-line secondary storage to the main memory and then main-memory to on-line secondary storage.

Transferring of data from on-line secondary to main memory is known as a swapping-in of data and transferring of data from main memory to on-line secondary is known as swap-out of data.

Demand Paging: In a virtual memory system, all the processes are partitions into pages (logical address space) and resides in on-line secondary storage. Pages are swapped-in and out of the main memory.

A program is initially stored in the on-line secondary storage. When a page in the program is required, it is swapped-in to the main memory. It is called demand paging because until a page is not required, it is not loaded.

(3) Based on the way, virtual memory is realised.

(4) Virtual memory is often described hierarchy of two storage system:

(i) On-line disk storage
(ii) Main memory (RAM)

The Operating system manages the two storage system in such a way that the users of the system feels that they have accessed a single, large, direct addressable and fast main memory.

(5) This is from the point of view of application program, virtual (realise/imagine/feeling) size of the available main memory is to be unlimited.

(6) Virtual memory is storing data and instruction of a program in the on-line secondary storage and when they required, they are loaded in the main memory.

In other word, the part of a program is required at any instant is loaded in the main memory while the rest of the program is in the secondary requirement.

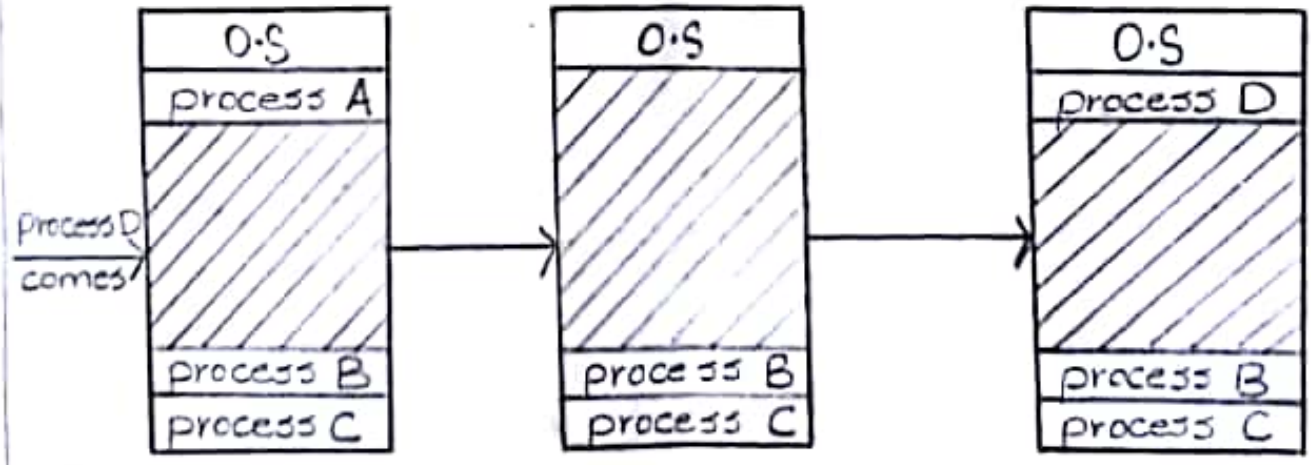
SWAPPING

(1) Swapping is the technique to remove process from the main memory and storing it in the on-line secondary memory or secondary memory usually hard-disk.

(2) Swapping should be very fast so that it doesnot become a large overhead to execute a process.

(3) A process is swapped-out to make room in the main memory for the processes.

(4) The area of the disk where the swap-out processes are stored is called as the swap area.



process A → swap-out
process D → swap-in

EXPLANATION

(i) In the above figure, the concept of swapping is the multitasking system. It occupy a part of the memory. Various user processes use rest of the memory known as user program area of the memory.

(ii) Initially, memory is allocated to process A, B, C.

(iii) process 'D' is required to be loaded in the memory for execution.

(iv) The available space is not sufficient for the process 'D'.

(v) process 'A' is swapped-out of the memory and is stored in on-line secondary storage or secondary memory (i.e. hard disk.)

(vi) To create space for process 'D', process 'A' is loaded into on-line secondary storage.

PAGING

(1) Paging is a method of non-contiguous memory allocatio.

(2) In paging, logical address space is divided into fixed-size blocks, known as "pages".

(3) Physical address space is also divided into fixed-size blocks, known as "frame."

(4) A page is mapped into a frame.

(5) Individual pages and frames are recognized by unique number known as page number and frame number respectively.

(6) Size of a page is equals to (=) the size of a frame.

(7) Page Number denotes unique identification. Rest of logical address denotes Offset.

(8) Frame Number denotes unique identification. Rest of physical address denotes Offset.

(9) Example-

Logical Address (page) of 13 bit.

Page no.	Offset				
011	10	10	10	10	10

The higher 13-10 i.e. 3 bit denotes the page number and rest of the 10 bit denotes Offset.

Frame no.	Offset				
011	10	10	10	10	10

frame	00
frame	01
frame	10
frame	11

The logical address 011 1010101010 denotes the page number 011, the higher order left 3 bit.

<10> Operating system maintain a table to convert a logical address into its corresponding physical address. This table is called MPT (Memory Page Table:)

page no.	frame no.	presence bit
000	00	0
001	01	1
010	10	0
011	11	1

Fig. Memory Page Table

<11> There are two fields in the page table. The first field stored the corresponding frame number and the next field is called presence bit which shows page is available or not.

Ex- page 6 is stored in the 13th frame.

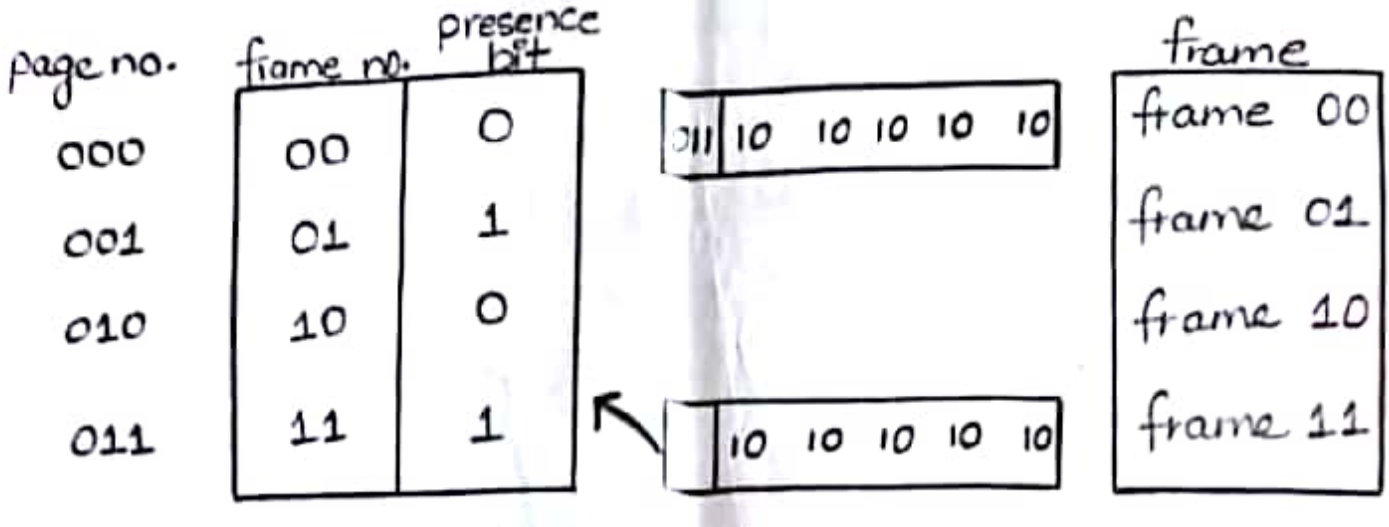


Fig. Paging system

PAGE FAULT

(1) In demand paging, M·M·U (Memory Management Unit) also maintain an index table for address translation (Logical to physical and physical to logical) known as "Page Table". The page Table stores the information in the main memory.

(2) When a page is required, the operating system scan the page table to find out the page. If the page is found, operating system continue processing. In case the page is not present in the main memory, page fault occurs.

(3) A page fault is not a fault or errors rather it indicates a situation that the page requested by a program is not currently available in the main memory.

When a page fault occurs, M·M·U swap-in the requested page from the in-line secondary memory into the main memory.

In case there is no free space in the main memory, MMU finds free space if not available. MMU swap-out an unused page from the main memory and stored unused page in the on-line secondary memory for creating space in the main memory. Then the required page is loaded in the main memory and finally page table is updated.

Page Replacement Algorithms

A page replacement algorithm is the logic or policy regarding how to select a page to be swapped-out-from the main memory to create space for the page known as requested page, which caused a page fault.

There are several page replacement algorithm. These are as follows:

(1) First-in-First-out (FIFO) Replacement Algorithm.
(2) Optimal page Replacement Algorithm.
(3) Least Recently Used (LRU) Replacement Algorithm.

(1) First-in-First-out (FIFO) Replacement Algorithm

The First-in-First-out Replacement Algorithm is the simplest of all the page replacement algorithm. It conveys a basic idea that when a page fault occurs, the oldest page in the main memory is to be swapped-out of the main memory to create room for the required page that needs to be executed.

For Example: Page number 14, 21 and 18 are present in the memory and page number 21 was loaded first followed by 14 and 18.

If the page number 30 is required to be accessed by the C.P.U, it results in Page fault because page number 30 is not available in the main memory.

The page loaded first ie. page number 21 is swapped-out of the main memory and stored in the on-line secondary memory to create room for the page number 30. Then 30 is loaded in the main memory.

(2) Optimal page Replacement Algorithm (will not been)

(1) Optimal page Replacement Algorithm is considered as the best possible page replacement algorithm.

(2) In virtual memory, theoretically it is difficult to implement.

(3) According to the page replacement policy, the page in the main memory which is not been referred or requested for the longest time is swapped-out from the main memory and creates room for the requested page.

For Example: page 14, 21 and 18 are present in the main memory. page 82 is required to be loaded resulting in a page fault since page 82 is not in the main memory at present.

page 14 is not required till 2000 instruction. page 21 is not required till 1000 instruction and, page 18 is not required till 1500 instruction.

This means that page 14 is one that will not been accessed by the CPU for the largest time. Page 14 is swapped-out from the main memory and creates room for the page 82.

(3) Least Recently Used (LRU) Replacement Algorithm (Has not Been)

(1) LRU algorithm uses information about the pages accessed in recent past to predict the near-future.

(2) LRU algorithm states that when a page fault occurs, the page that has not been referred to for the largest time is swapped-out of the main memory to create space for the requested page that has caused the page fault.

(3) Implementation of LRU can be done in various ways. One of the common method is to apply the LRU scheme for virtual memory management which is done by using an array.

The Array stores the information about the page present in the main memory.

The front end of the array stores the page accessed recently.

The rear end (back end) of the array stores the pages that has not been accessed for the longest period of time.

However, a page that is present in the main memory is accessed. The information about the page is shifted to the front end of the array.

If a page fault occurs, the

page indicated by the rear end of the array is swapped-out of the main memory and requested page is swapped-in the main memory.

Information about the page which is swapped-in is stored in the front end of the array.

For Example→ The array stores the following four page number

12, 56, 27, 61.

page 12 is in the front end of the array. This indicates that page 12 has been accessed recently.

page 61 is at the rear end of the array. This indicates that page 61 has not been referred for the longest time.

page 27 is accessed. No page fault occurs because page 27 is present in the main memory. Information about the page 27 is shifted to the front end of the array. The array becomes

{27, 12, 56, 61}

page 42 is referred to access. A page fault occurs, since page 42 is not in the main memory.

page 61 is at the rear end of the array which indicates that it has not been accessed for the longest time. So, page 61 is swapped-out from the main memory. The array becomes

{42, 27, 12, 56}

SEGMENTATION

(1) Memory segmentation is a technique of non-contiguous memory allocation.

(2) In segmentation, logical address is divided into certain unequal size of chunks known as segment.

(3) A logical address consists of two parts
   i) Segment number
   ii) Offset

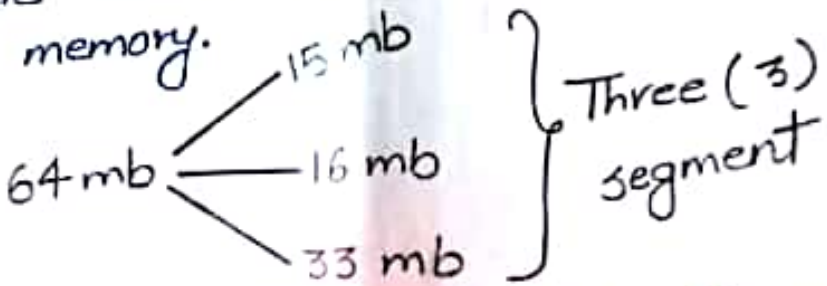
(4) Example- logical address

Segment no.	Offset

3 : 1000

3:1000 indicate that 1000 offset and segment is 3.

(5) An application can consist of three modules. Each module is loaded in distinct segment in the main memory.



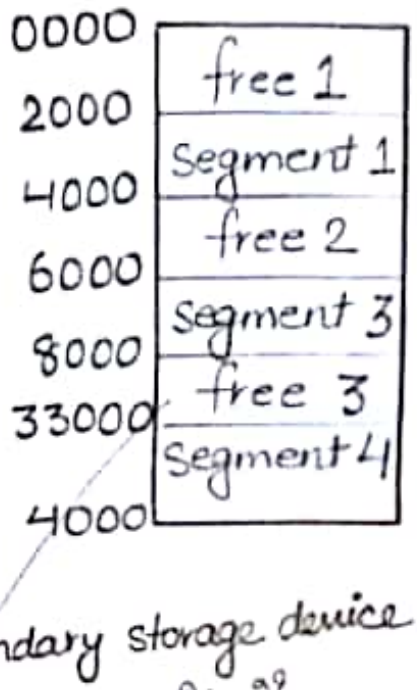
(6) Segmentation table is used for the validation of an address.

(7) A segmentation table has two field
   i) Base address of the segment.
   ii) Size of the segment.

(8) Base address of the segment is the base address of the location of a particular segment in the main memory from where segment stores.

(9) Segment Table

Segment no.	Base address	Size of segment
0	1000	800
1	6000	1000
2	3000	1200
3	4000	2000
4	2000	4000



0,2 secondary storage device में है।

According to this figure, various memory segments in the main memory and there corresponding entries in the segment table.

For Example- Segment 3 has loaded in the main memory.

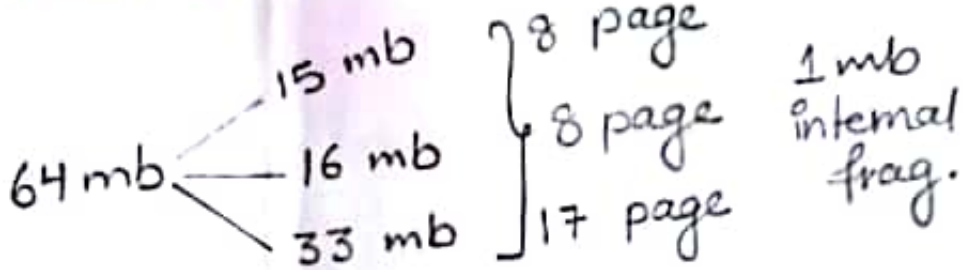
PAGING AND SEGMENTATION SYSTEM

(1) Segmentation of memory can be implemented with paging scheme.

(2) Unequal size of segment can be represented of equal size.

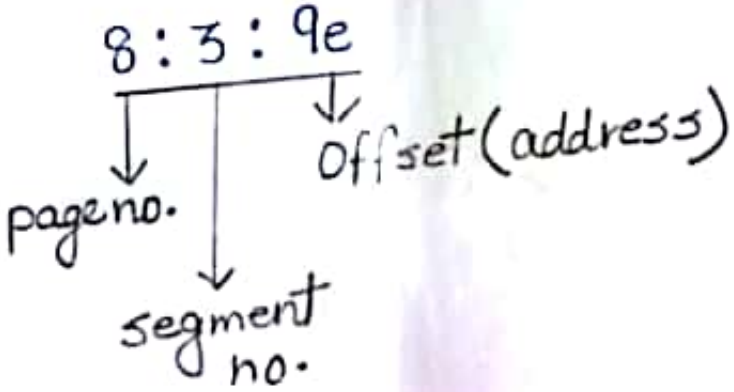
(3) For Example- In a logical address space of 64 mb, there are 32 page of 2 mb each. A program is divided into three segment of size 15mb, 16 mb, 33mb each.

These three segment require 8, 8, 17 pages where 1 mb is an internal fragmentation.



(4) A logical address has three part
i) segment no.
ii) page no.
iii) offset

Example-



VICE MANAGEMENT

1) Device Management- control pheripheral devices by sending their command in their own proprietory language (अपनी रूप भाषा में)

2) Device Management manages all the devices such as input devices, output device and other processing device installed with the system.

CONTROL OF VARIOUS DEVICES

1) Compiter perform two main function:
I/O and processing data.

2) Input unit - By input (some kind of data) such as numeric or character are inserted into the computer system through a program in order to being processed for specific result.

3) Output unit - This process of inputting data is then produced outside known as output.

4) Various devices such as I/O device, processing device are controlled and managed by two aspects:
a) I/O subsystem
b) Device Driver

a) I/O subsystem ⇒ I/O subsystem is a part of operating system that control and manages all types of input-output operation performed by I/O devices. Controlling I/O command and devices is the main aspect of operating system i.e. done by I/O subsystem

b) Device Driver ⇒ The software routine that knowns how to deal with the device is called a device driver.

The Operating system contain all the ...er for the peripherals attached to the computer for ...mmunication. When a new peripheral is added, the device driver is install into them.

NEED OF HARDWARE (H/W) AND SOFTWARE (S/W) RULE

Any Operating System is made up of linker, loader, locator (address definition), file manager, debugger, translator which gets hardware requirement such as keyboard controllers, Bus controllers, serial controllers, PS/2, USB controllers, DMA controllers, Network (N/W) controllers, Interrupt controller; Display controller (VGA), multimedia controller. It is the important issue for any device management which gets certain need of hardware (H/W) and software (S/W) rules such as

I/O subsystem -

(1) It is a part of operating system.
(2) It control all types of input-output operation performed by device.
(3) It is an essential part of operating system which control I/O i.e hardware.

Interrupt Driven -

(1) Processor transfer the I/O operation generated by a program -to I/O subsystem, processor continues with processing of program. When I/O subsystem completes the execution of I/O operation, it interrupts. This is called Interrupt driven or pull down data transfer.

Interrupt are the most important rules of an Operating system. Interrupt stop the execution of a program to perform other task such as numerical computation. This is done when a process need to perform an I/O operation. If an interrupt occur, the CPU stores the current status of the process in the program register(pr) and stops the execution time.

DEVICE DRIVER

(1) The software that the system user communicate with device such as printer, mouse, keyboard etc.. is called a device driver.

(2) A program that allow a hardware pheripheral known as a device to communicate with a computer.

(3) The system software acts as an interface between an application and hardware devices.

(4) A program that interface between windows and a device like printer, scanner etc.

(5) Hardware devices such as sound cards, printers, CD-ROM drive must be installed proper driver to run.

(6) Each device from each manufacturer will habits its own device driver otherwise each device is difficult to run.

(7) The driver, a program that determine how an operating system will communicate with a pheripheral device.

(8) It is a software that is essential requirement for each hardware or devices attached in the system.

(9) Device Driver such as visual display driver, sound drivers, hardware driver, basic driver(Keyboards mouse, disks I/O)