DATE : / /
PAGE NO. :

Unit-01

Introduction and History of operating System (Processor Management)

* Operating System : operating system is a program, that controls the execution of an application program and acts as an interface b/w user & computer hardware.

* Objective of Operating System :
→ Hide the detailing of the machine/system from the user.
→ Provide convenient interface of the computer system for user.
→ Manage all the resources, task, processes

Ex: Windows, Mac OS, Linux etc.

* Generations of operating system

[1] 1st Generation [1945-1955] : Vacume tubes & plugboards
⇒ Calculating Engines with mechanical relay were build at that time, However, the mechanical relay were very slow and were later replaced with the vacume tubes.

[2] 2nd Generation [1955-1965] : Monitrome computer (batches)
→ Transistors leads to the development of the computer system.
→ The batch system was introduce to reduce the waste time in computer.

[3] 3rd Generation [1965-1980] : Mini computer (multiprogramming
→ In this generation used IC's & provide performance.

[4] 4th Generation [1980-present] : Personal computer
→ Personal computer were easy to create with the development of large scale IC's

(*) Characteristic of OS

(1) Memory Management : Primary & secondary memory

(2) Processor Management : CPU scheduling

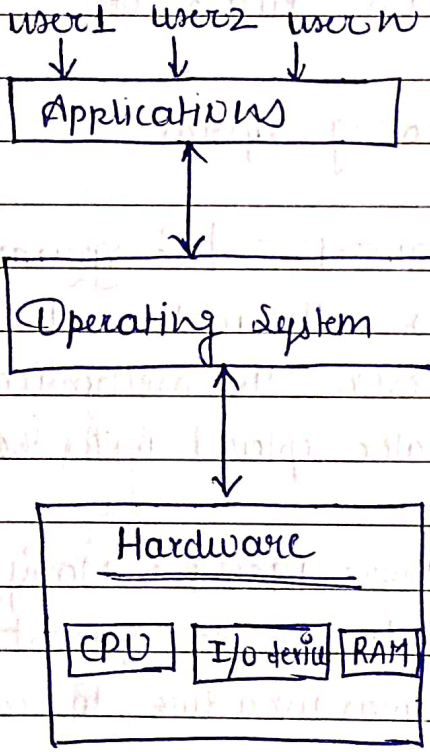
(3) Device Management : In which we check ~~the device~~ which procss is allocated to device.

(4) File Management : hard disk (HDD), SSD

(5) Security Management : provide the security from unauthorised access.

(6) Job Scheduling : In every system task is scheduled and after time expires the session.

(*) Block diagram of operating system



POORNIMA

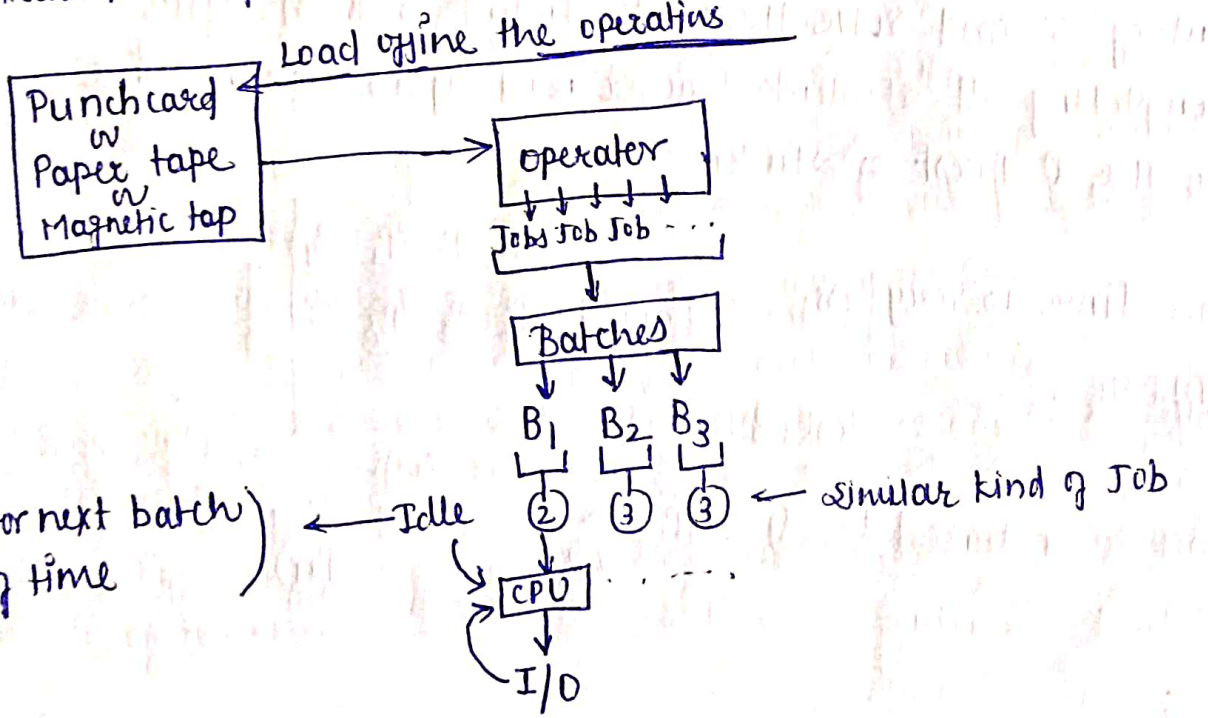
Date	Unit No.	Lecture No.	Faculty	Subject Name	Subject Code	Main Topics:-

★ Types of Operating System

[1] Batch OS
[2] Multiprogrammed OS
[3] Multitasking OS
[4] Real time OS
[5] Distributed OS
[6] clustered OS
[7] Embedded OS

[1] Batch Operating System

⇒ In the system we make batch of ~~similler~~ similar kind of Jobs to executed the operations.



Main Ideas, Questions & Summary:

Library / Website Ref.:-

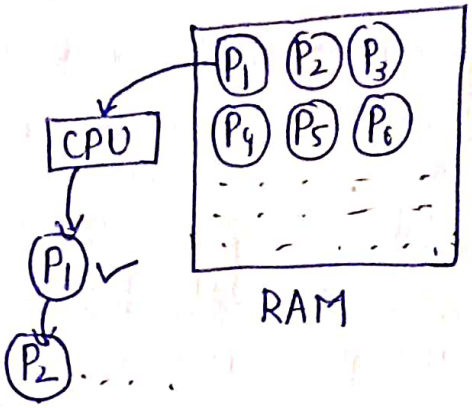
[2] Multiprogrammed operating systems

⇒ Multiprogrammed OS is not preamptive, it means in the RAM ~~Atmost~~ programs are exist, and CPU solved every problem and after solve complete problem it will pass to the next.

Disadvantage :

① Response time : Response time for the last problem is very high.

② IDLENESS



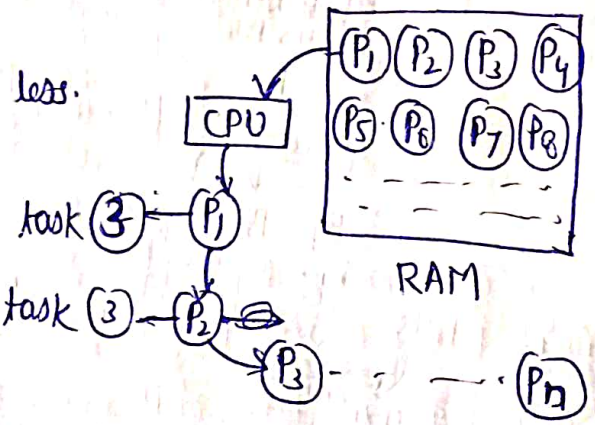
[3] Multitasking operating systems / Time sharing OS

⇒ Multitasking OS is preamptive. It means CPU will give the fix time ~~Sedule~~ Schedule for every problems.

Ex. In a class, teacher (CPU) solve the ~~problem of every~~ 3 problem out of 5 and solve the next student problem. After completing all student teacher will again schedul time to the ~~F~~ First ~~s~~ student.

① Responce time is very less.

② Not IDLENESS



POORNIMA

Date	Unit No.	Lecture No.	Faculty	Subject Name	Subject Code	Main Topics:-

[4] Real time operating system

① Hard Real time OS : Very restrict for time, No delay

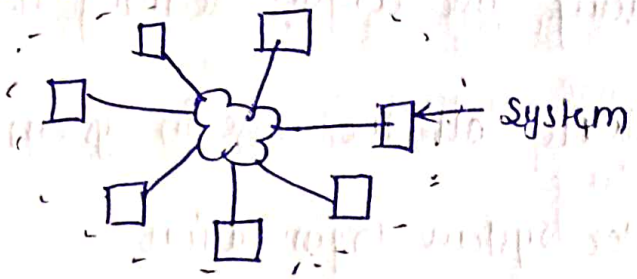
② Soft Real time OS : Less restricted than Hard Real time OS.

[5] Distributed Environment

⇒ All the OS are distributed in environment and connected with network

[6] Clustered environment

⇒ Connection of all system in the same environment with the server. These are connected to local network.



→ Load bearing is the advantage of clustered Environment OS.

→ Scalability

→ fall tolarance

[7] Embedded ~~system~~ operating Systems

⇒ It will work with the fixed functionality like microwave, washing machine AC etc.

→ An embedded operating system is a computer operating system designed for use in embedded computer systems.

→ These operating systems are designed to be small, resource efficient, dependable and reduce many features that aren't required by specialized application.

Main Ideas, Questions & Summary:

Library / Website Ref.:-

Q. What is an OS, what it does?

Ans ⇒ User View

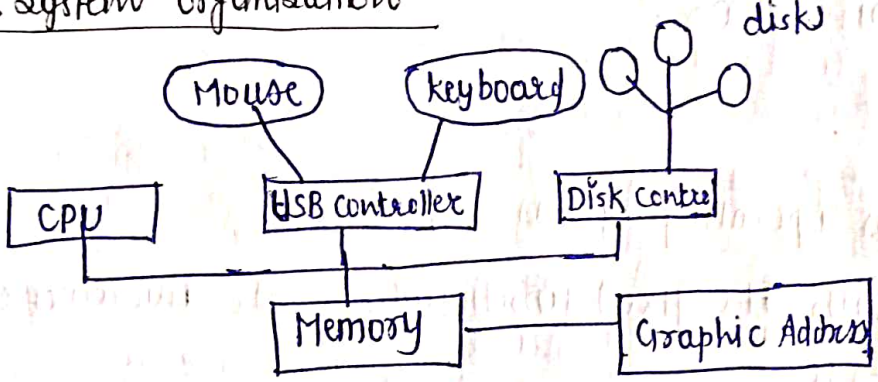
- It varies according to the user and the interface being used.
- If we are using PC then system is optimized according to the single user experience.
- The OS design to maximize the resource utilization to use the whole CPU time.

System view

- OS can be named as resource allocator.
- To handle a problem computer has many resources CPU time, processor, file storage, memory. etc.
- Facing the conflict deciding the best case is the working of OS.

In OS dones all the time in system [kernal]

★ Computer System Organisation



Bootstrap Process

For a computer to be started at it powered up & rebuted. It needs a program to initiated. Bootstrap program must know how to ~~program~~ load OS. & start executing. For this bootstrap program must locate in the memory & OS kernal.

Firmware → electrically programmable ROM

POORNIMA

Date	Unit No.	Lecture No.	Faculty	Subject Name	Subject Code	Main Topics:-

- **Storage Structure :**

⇒ There are two type of storage
① Internal – RAM : Internal Instruction
② External – ROM

- Input/output structure

⇒ The large portion of OS is dedicted to I/O vehicles because the peyormance of the system need to be more eyicient. ~~Vehicles~~, the vehicles, the devices are of different types.

- **Computer System Architecture**

[1] Single Processor computer : In which have single CPU. It will work serially.
[2] Multiprocessor computer : In which have two or more CPU It will work parallelly.
[3] Clustered Processor computer : In whiche multiple devices/systems and connected to multiple CPU. It will wark parakely & simultansely.

(*) Structures of Operating System

(1) Layered approach : OS is divided into No. of layers where innermost layer is (L_0) is kernal & outermost layer is (L_n) UI. Each layer can only uses the services of its lower layer.

(2) Monolithic : The Entire OS is running in kernal space & and set of system was define to provide services such as memory management, scheduling etc.

(3) Microkernal Structure : Only bare minimum services of all tasks are executed in user space.
→ Better security
→ Easy to port

(4) Exokernal architecture / structure : user can access the raw hardware because the OS removes the all abstractions b/w user & hardware.
→ Benyits : Utilization of hardware

(5) Virtual Machines : Provide interface b/w guest OS & OS. Each guest feels like it has unlimited access to hardware.

DATE : / /
PAGE NO. :

★ Processor Management :

⊛ Interprocess communication (IPC)

⇒ A process can be two types

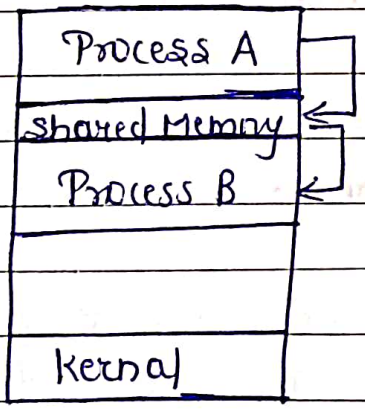
- Independent process
- Co-operating process

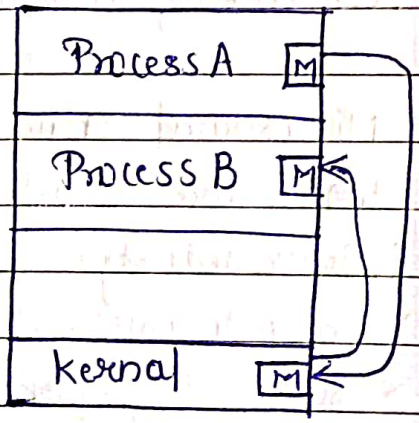
→ An Independent process is not affected by the execution of other processes while a co-operating processes can be affected by other executing processes.

→ IPC is a mechanism that allows processes to communicate with each other and synchronise their actions.

→ Processes can communicate with each other through

① Shared Memory ② Message passing





DATE : / /
PAGE NO. :

(*) Mutual Exclusion

⇒ During concurrent execution of processes, processes need to enter the (section of program shared across processes) critical section at a time of execution.

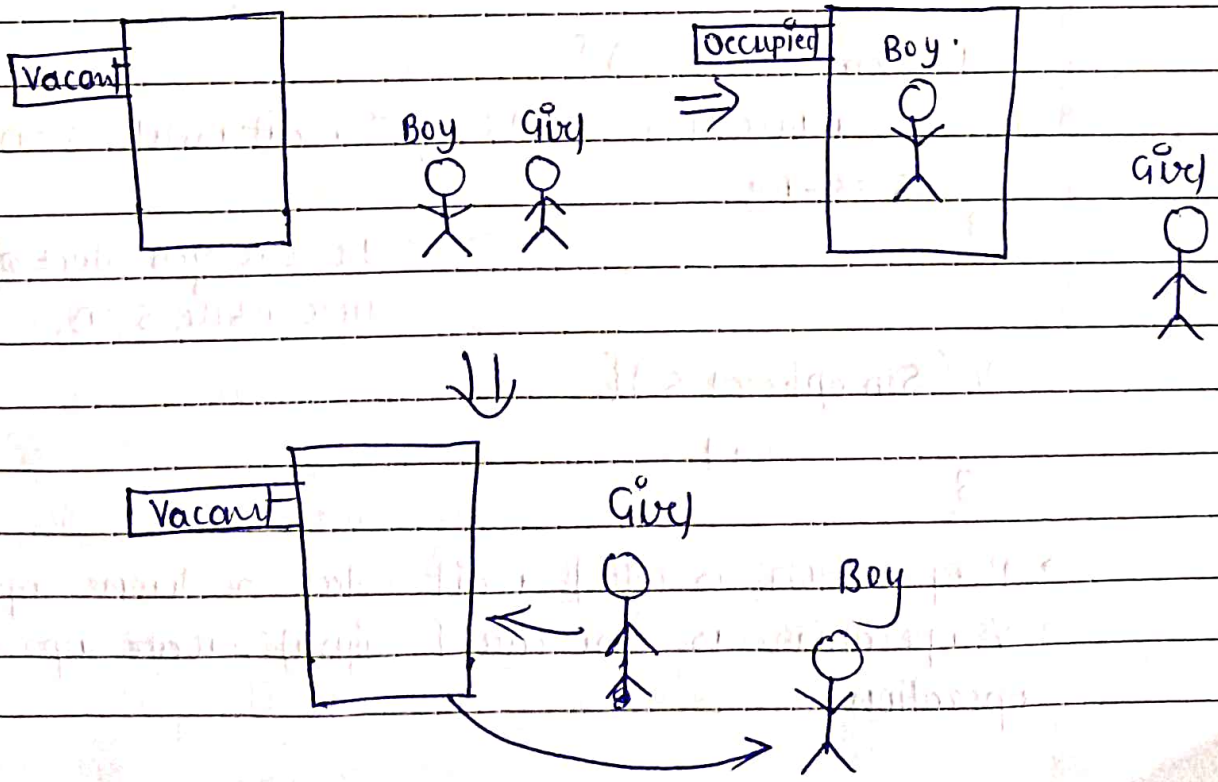
→ It might happens that because of the execution of multiple processes at a same time, so, value stored in critical section is inconsistant.

→ This process is called Race condition.

→ The primary task of process synchronization is to get rid of race condition while executing the critical section.

→ This is achieved through Mutual exclusion.

→ Mutual exclusion is a property of procss synchronization.



DATE : / /
PAGE NO. :

⊛ Semaphores in Process Synchronization

→ Semaphores was proposed by Dijakstra. It is very significant technique to manage concurrent processes by using simple integer value which is known as a semaphore.

→ Semaphores is an integer variable that will shared b/w threads.

→ There are two types of semaphores :

① Binary Semaphores : This is also known as mutex lock. It can have only two values 0 & 1. value is initialized to 1.
It is used to solve / implement the solution of critical section problems with multiple processes.

② Counting Semaphores : Its value can range over an unrestricted domain. It is used to control access to a resource that has multiple instances.

```
P(Semaphores s){
    while (S==0); /* wait until S=0 */
    S=S-1;
}

V(Semaphores s){
    S=S+1
}
```

The code gets stucked Here while S=0

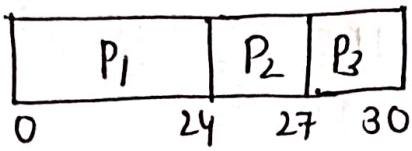
→ P operation is called wait, sleep or down operation.
→ V operation is also called signal, wake-up, or up operation.

★ Scheduling ~~Scheduling~~ Algorithm

[1] FCFS (first come first serve) ← Non-premptive (staxvation problem occurs)

⇒ The process come first to the system will serve first.

$P_1 = 24$
$P_2 = 3$
$P_3 = 3$



~~Total time = 30~~

Note: CPU burst time is the ~~the~~ total time taken by the process.

[1] Waiting time : ($P_1 = 0$, $P_2 = 24$, $P_3 = 27$) Avg. waiting time $= \frac{0+24+27}{3}$

$= 17$ sec

[2] Turn-around time : ($P_1 = 24$, $P_2 = 27$, $P_3 = 30$)

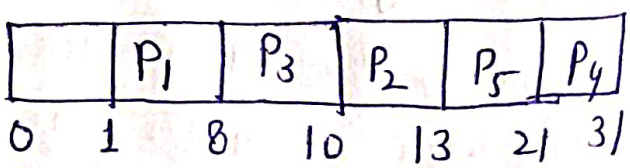
$$\text{Avg TAT} = \frac{24+27+30}{3} = \frac{81}{3} = 27 \text{ sec}$$

[3] Round trip time : The length of time taken by a data packet to be sent to a destination including the time it takes for an acknowledgement of that packet to be received back at the origin place.

[4] Response time :

[2] SJF (Shortest Job First)

PID	Arrival time	Burst time	completion time ↓	Turn around time	Waiting time
1	1	7	8	7	0
2	3	3	13	10	7
3	6	2	10	4	2
4	7	10	31	24	14
5	9	8	21	12	4



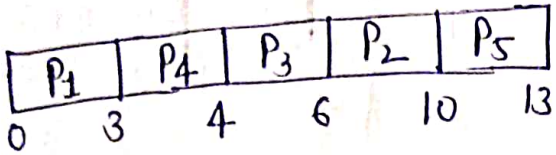
Avg waiting time $= \frac{27}{5}$

POORNIMA

Date	Unit No.	Lecture No.	Faculty	Subject Name	Subject Code	Main Topics:-

[3] Priority ~~Scheduling~~ Scheduling

Process	Burst Time	Priority
P_1	3	1
P_2	4	3
P_3	2	2
P_4	1	1
P_5	3	4

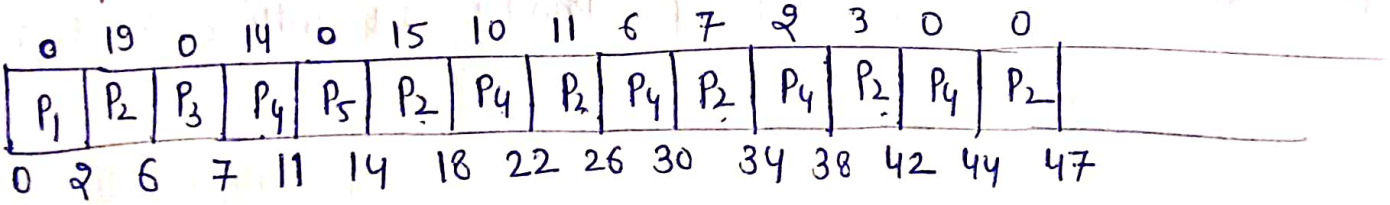


[4] Round Round Scheduling

Eg: $P_1 = 2$ $P_4 = 18$ quanta = 4

$P_2 = 23$ $P_5 = 3$

$P_3 = 1$



$P_1 = 0 = 0$

$P_2 = 44 - 20 = 24$

$P_3 = 8 - 6 = 2$

$P_4 = 42 - 16 = 26$

$P_5 = 11 = 11$

62

$\frac{62}{5} = 12.4$ Ans

DATE : / /
PAGE NO. :

(*) Scheduling Algorithms

→ The purpose of scheduling Algorithms :

1. Maximum CPU utilization
2. Fare allocation of CPU
3. Maximum throughput
4. Maximum turnaround time
5. Minimum waiting time
6. Minimum response time.

★ Fair share scheduling

A	25%	A 33.3%	A 50%
B	25%	B 33.3%	B 50%
C	25%	C 33.3%	
D	25%		

★ Multiple feedback queue

	Burst time	priority
P_1	23	1
P_2	10	2
P_3	5	1
P_4	8	2
P_5	12	1

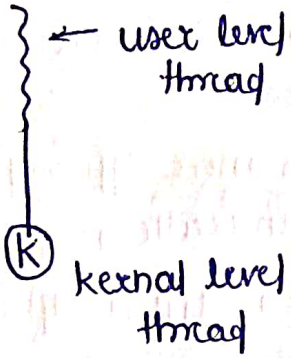
Round Robin scheduling
FCFS scheduling
Priority scheduling

POORNIMA

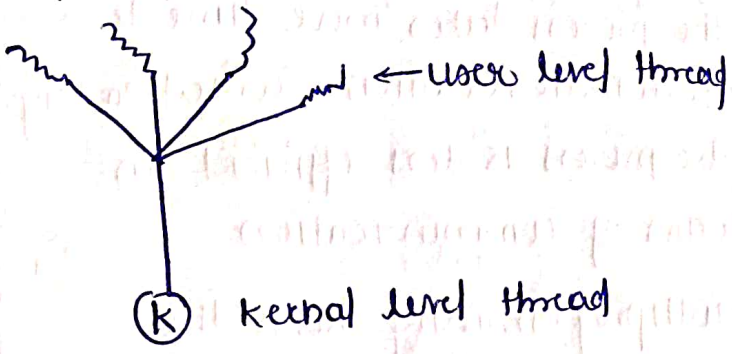
Date	Unit No.	Lecture No.	Faculty	Subject Name	Subject Code	Main Topics:-

★ Multithreading Models

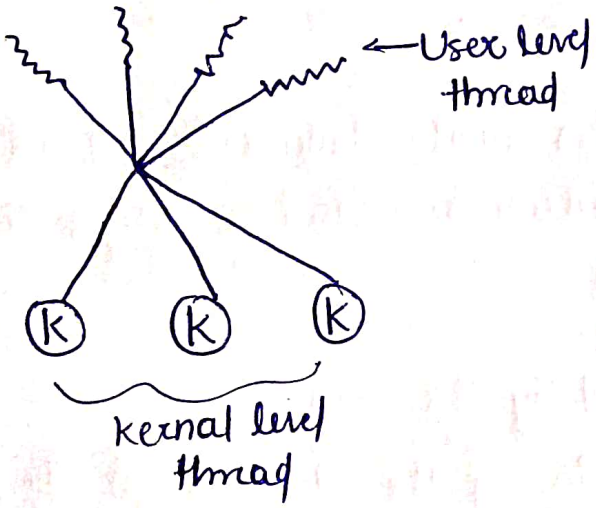
[1] 1 to 1



[2] 1 to M



[2] M to M



★ Threads v/s Process

Process	Thread
① Process means any program is in execution.	① Thread means a segment of a process.
② The process takes more time to terminate, creation & context switching.	② The thread ~~less~~ takes less time to terminate, creation & context switching.
③ The process is less efficient in terms of communication	③ Thread is more efficient in terms of communication.
④ Multiprogramming holds the concept of multi-process.	④ Thread is lightweight as each thread in process shares code data and resources.
⑤ The process is isolated.	
⑥ Changes is parent process do not ~~affect~~ affect child processes.	

★ Threads (Advantage & disadvantage)

→ Responsiveness
→ Faster contex switching
→ Effective utilization of multiprocessor
→ Resource sharing
→ Communication
→ Enhanced throughput of the system

★ Process (Advantage & disadvantage)

→

POORNIMA

Date	Unit No.	Lecture No.	Faculty	Subject Name	Subject Code	Main Topics:-

* Bootstrap Loader

⇒ It is a small program which can be activated immediately after a microcontroller has been powered up.

→ In order to load & execute another program in a well defined manner.

Note: Bootstrap Loader is a part of BIOS.

* BIOS (Basic input/output system)

⇒ It is a program in computer's µp it uses to start computer systems after it is powered on. It provides the data flow b/w computers, OS & attached devices.

Q. BIOS same as bootstrap loader ?

★ Scheduling queue

① Job queue :
② Ready queue :
③ Device queue :

★ Schedulers

① Long term schedulers : It bring all the jobs to the ready state.
② Short term schedulers : It select any jobs from the ready queue & dispatch the CPU for the execution
③ Mid term schedulers : It take care of the swap out processes. & controls

Main Ideas, Questions & Summary:

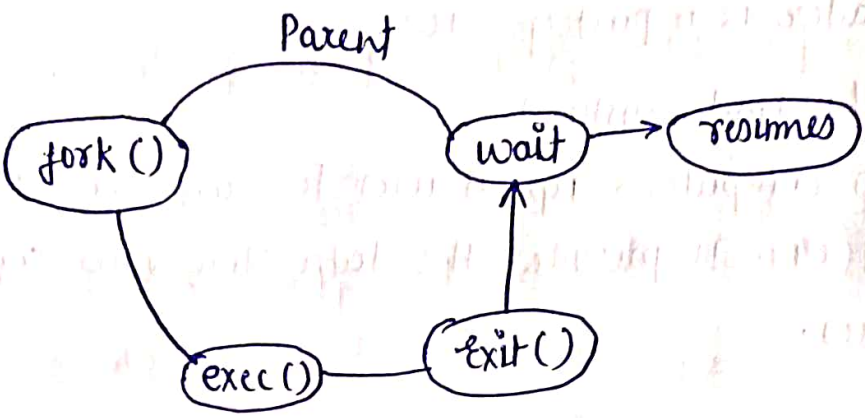
Library / Website Ref.:-

⊛ Dispatcher

→ It is responsible for loading process selected by short term scheduler on the CPU.

⊛ Process Creation

The process may create several new process via create system call system call during the case of execution



⊛ Process Termination

⇒ The process terminates when it finishes final statement and ask them. by using system exit call. The process returned an integer to the parent process via wait call.

All the Resources will be reallocated by the OS.

⊛ Parent may terminate the execution of the children for the following reason.

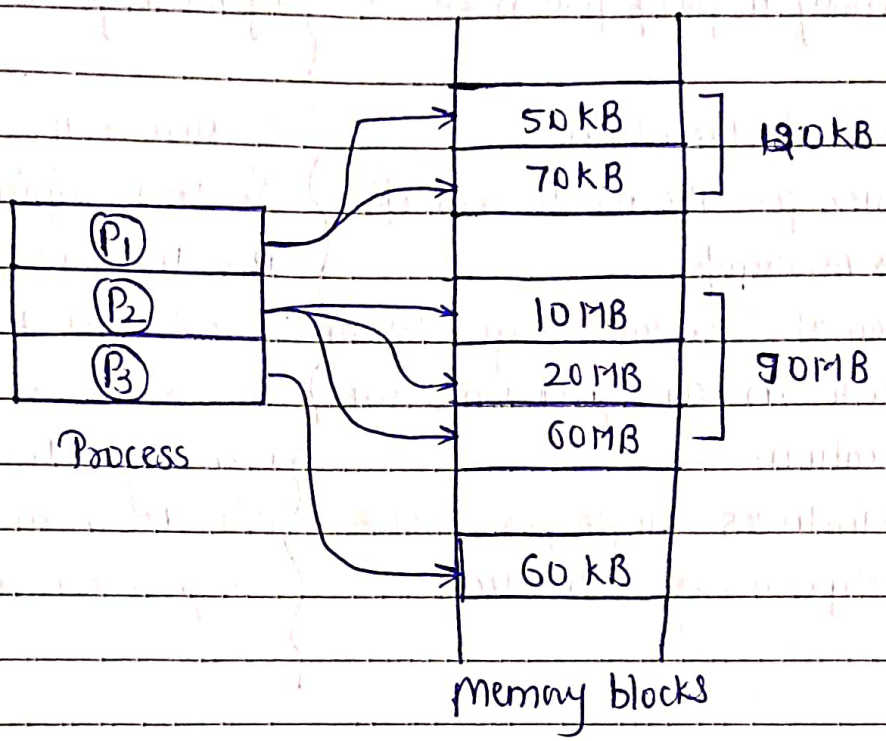
→ It Exceeds the usage of some resource.
→ The task assign to the process is not longer require.
→ The parent exit on OS not allowing the process to continue if parent is terminated.

DATE : / /
PAGE NO. :

Unit - 02 : Memory Management

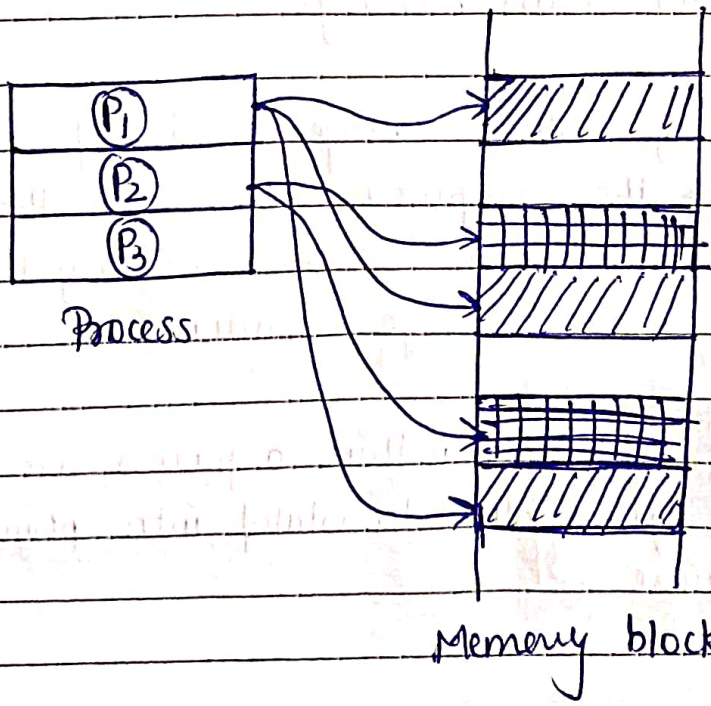
⊛ Contiguous Memory Allocation

⇒ Contiguous Memory allocation is basically a method in which a single contiguous section/part of memory is allocated to a process or file needing it.



⊛ Non-Contiguous Memory Allocation

⇒ It's basically a method in which allocates memory space present in different locations to the process as per requirement.



DATE : / /
PAGE NO. :

- Difference b/w Contiguous and Non-contiguous Memory Allocation

Contiguous Memory Allocation	Non-contiguous Memory Allocation
① It allocates consecutive blocks of memory to file/process.	① It allocates separate blocks of memory to file/process.
② Faster in Execution	② Slower in Execution
③ Easier for the OS to control	③ It is difficult for the OS to control Overhead.
Overhead	
④ Internal & External Fragmentation occurs in contiguous memory allocation.	④ External Fragmentation occurs in Non-contiguous memory allocation.
⑤ It includes single partition & multipartition allocation.	⑤ It includes paging & segmentation.

★ Virtual Memory

⇒ Virtual Memory is a storage allocation scheme in which secondary memory can be addressed as though it were part of the main memory.

⇒ The size of virtual storage is limited by addressing scheme of the computer system and the amount of secondary memory is available not by the actual number of main storage locations.

① ⇒ All memory references within a process are logical addresses that are dynamically translated into physical addresses at run time.

POORNIMA

Date	Unit No.	Lecture No.	Faculty	Subject Name	Subject Code	Main Topics:-

★ **Paging** : It is the memory management technique that permits physical address space of a process ~~of a~~ to ~~Access~~ non-contiguous. It includes frames and pages.

→ Frames are fixed size block of physical memory.

→ Pages are fixed size blocks of logical memory.

Size of the page is generally 2kb, 4kb etc.

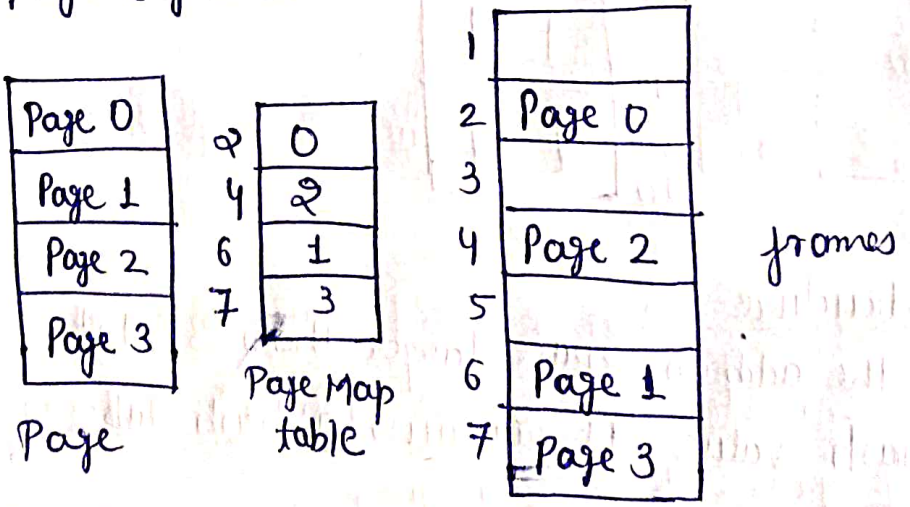
⇒ Every page contain page No.(MSB) & page offset(LSB).

⇒ In a system with virtual memory, programs use virtual address so that the can access large memory.

Physical Address = frames No. + page offset

Logical address = Page No. + page offset

⇒ The data structure called page Map table keep wrap of the relation b/w page & frame.



Main Ideas, Questions & Summary:

Library / Website Ref.:-

fault =>

→ We need to change a page table, we need to change only PTBR (Page table base register)

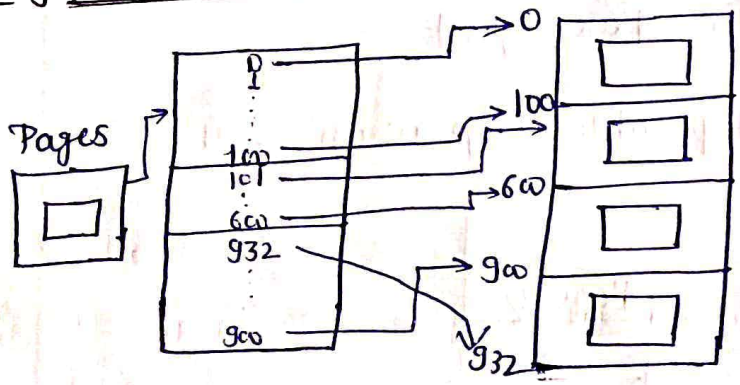
→ Two factor of PTBR
  → Retriving the Address
  → Removal the offset

⇒ Solution of the two factor delay is translation look aside buffer [TLB]
→ It contain key & value. It contains page table entries which is most recently used.

There are two process
→ Hit TLB : The page you want to find in TLB.
→ TLB Miss : The page you not found in TLB.

- Structure of page table

[1] Hierachical structure



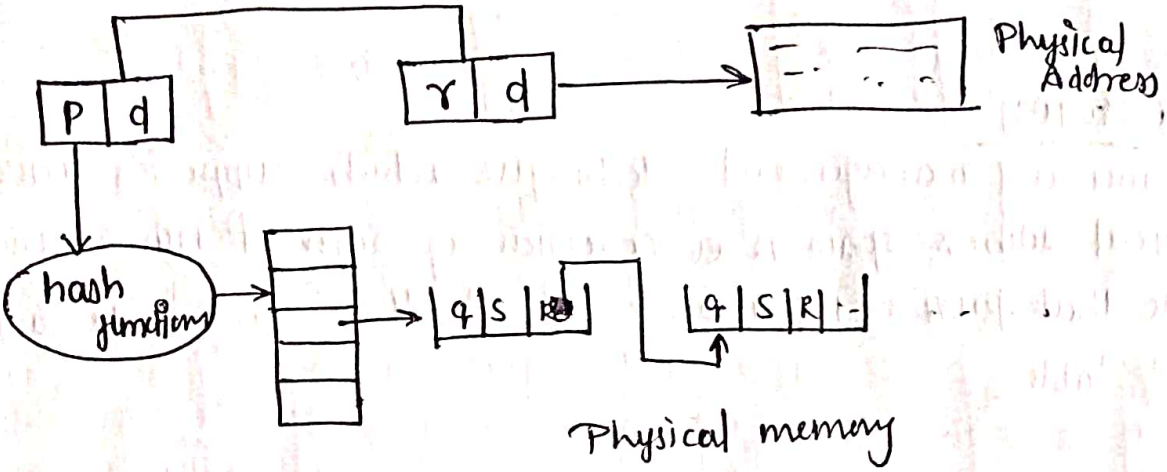
[2] Hashed Structure

⇒ It handle the address space larger than 32 bits. It is a table with a hash value. It contains a hash table which has three field.

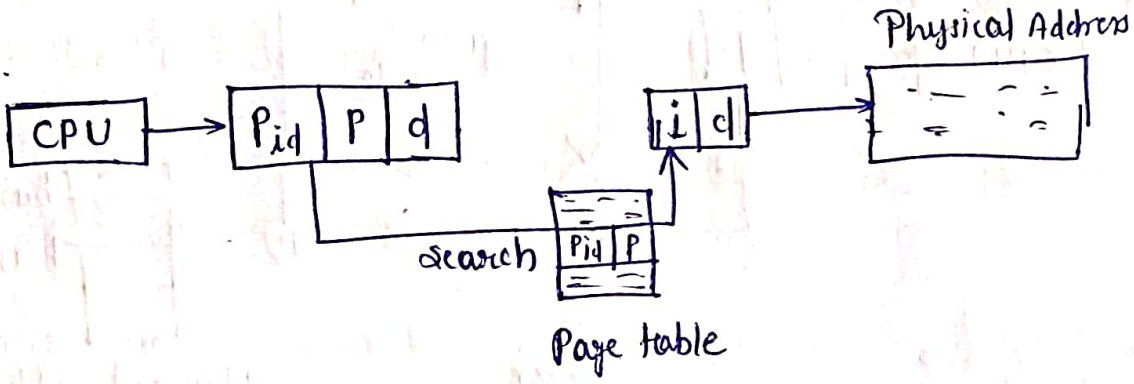
① virtual page number
② value of mapped pages & frames
③ Pointer to the next element in the linked list.

POORNIMA

Date	Unit No.	Lecture No.	Faculty	Subject Name	Subject Code	Main Topics:-



[3] Inverted structure



⇒ It contain the entry of one real page.

→ Each entry consist of virtual address of page stored in real memory location.

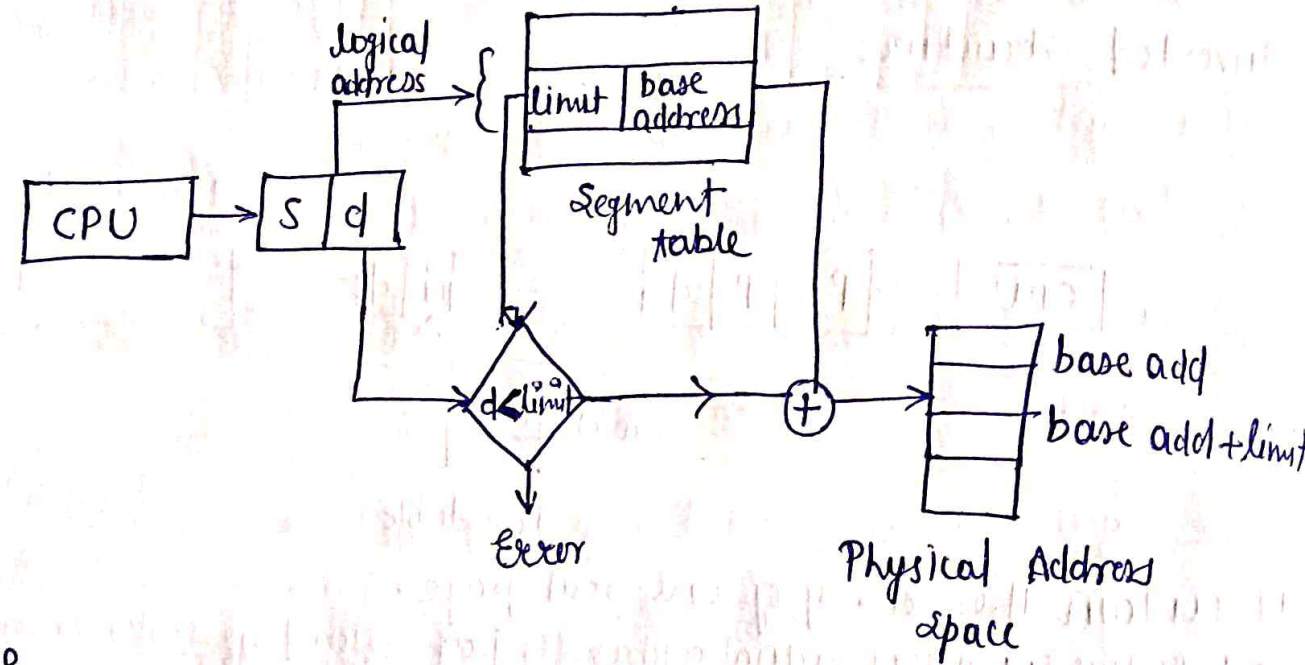
→ It increases the amount of time needed but decreases the memory required.

⊛ Segmentation

⇒ It's a memory management technique which support's user view.

→ The logical address space is a collection of segment name and his length

→ A Table that provide information about the segment is called Segment table.



⊛ Advantage

① It consume less space in compare to page table.

⊛ **Page Replacement** : When requested page is not available in main memory & available space is not sufficient ~~and~~ for allocation of requested space.

- **Demand Paging**

⇒ The process of loading the page into memory on demand (whenever page fault occures) is known as demand paging.

★ **Page Replacement Policies**

- **Page Fault :** It happens when running program accesses a memory page that is mapped into the virtual address space but is not loaded in physical memory.

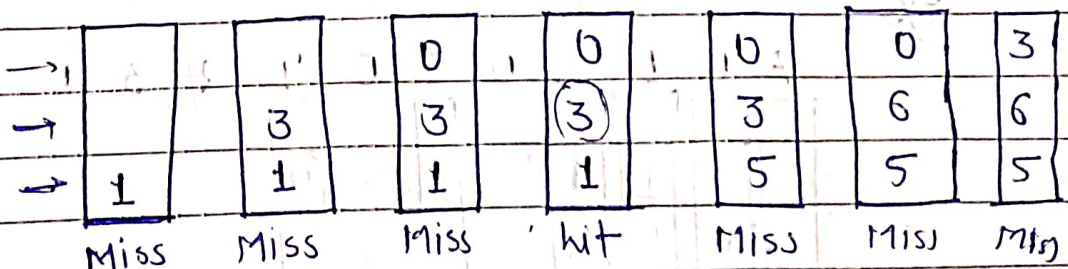
- **Page Replacement Algorithms**

[1] FIRST In FIRST OUT (FIFO)

⇒ This is the simplest page replacement algorithm. the OS keep tracks of all pages in the Memory in a queue.

Ex. Page Reference : 1 3 0 3 5 6 3

Slots = 3



Total page fault = 6

DATE : / /
PAGE NO. :

→ Belady's Anomaly proves that it is possible to have more page fault when increasing the No. of page frame while using FIFO.

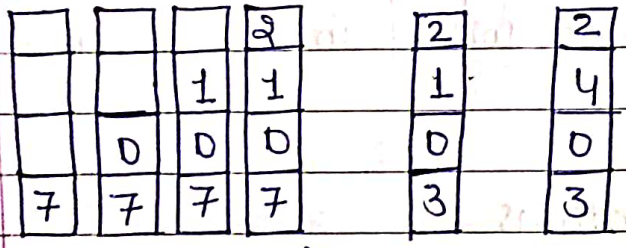
Ex. If reference string : 3, 2, 1, 0, 3, 2, 4, 3, 2, 1, 0, 4 with 3 slots. We get 9 page fault

but if we increase slots to 4, we get 10 page fault

2. Optimal Page replacement

⇒ In this algo, pages are replaced which would not be used for the longest duration of time in the future.

Ex. 7 0 1 2 0 3 0 4 2 3 0 3 2 3



1 is not used at long time in future

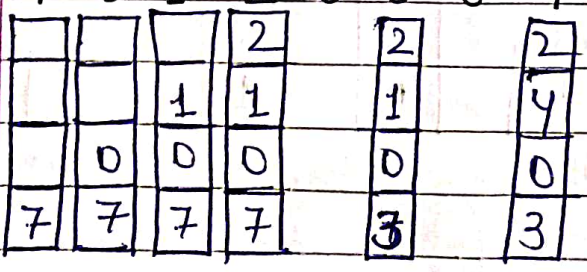
because 7 is not used at long time in future

No. of page fault = 6

3. Least Recently used (LRU)

⇒ Page will be replaced which is least recently used

Ex. 7 0 1 2 0 3 0 4 2 3 0 3 2 3



Total page fault = 6

DATE : / /
PAGE NO. :

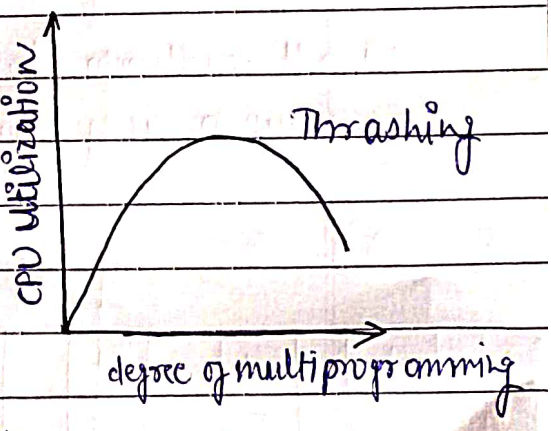
★ Thrashing

→ thrash is a poor performance of virtual memory (or paging) system when the same pages are being loaded repeatedly due to a lack of main memory to keep them in memory.

→ thrashing occurs when a computer's virtual memory resources are overused, leading to a constant state of paging and page faults, inhibiting most application level processing.

- Swapping : Whenever a page fault happens, the OS will try to fetch that page from secondary memory and try to swap it with one of the pages in RAM. This process is called swapping.

→ Thrashing occurs when page fault & swapping occurs frequently at higher rate.



- Algorithms during thrashing

(1) Global Page Replacement
→ Not suitable method

(2) Local Page Replacement
→ Alternative of Global page replacement

DATE : / /
PAGE NO. :

- Causes of Thrashing

① If CPU utilization is too low, we increases the degree of multiprogramming by introducing a new system.

② CPU utilization is plotted against degree of multiprogramming.

③ First, degree of Multiprog. ∝ CPU Utilization but thrashing sets in and CPU utilization drops sharply.

④ So, at this point, to increase CPU utilization and to stop thrashing, we must decrease the degree of multiprog.

- How to Eliminate Thrashing

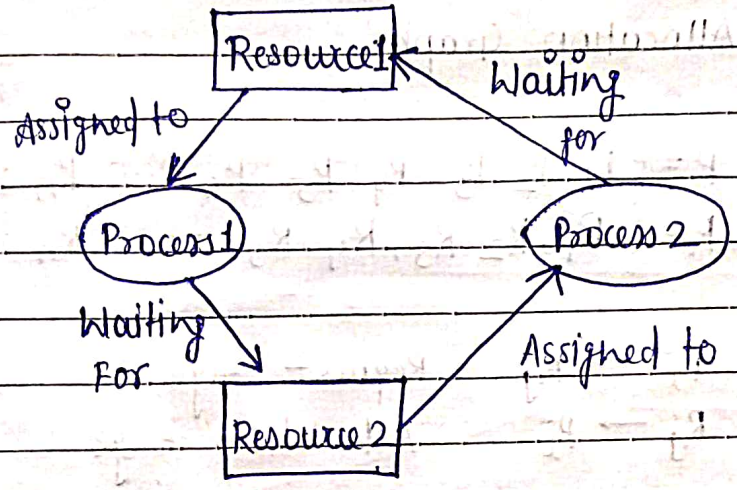
→ Adjust the swap file size
→ Increase the amount of RAM
→ Decrease the No. of applications
→ Replace programs.

DATE : / /
PAGE NO. :

Unit-3 : Deadlock & Device Management

★ Deadlock

⇒ Deadlock is a situation where a set of processes are blocked because each process is holding a resource & waiting for another resource acquired by some other process.



- Necessary condition for deadlock

① Mutual Exclusion : Two or more resources are non-shareable (only one process can use at a time)

② Hold & Wait : A process is holding at least one resource and waiting for resources.

③ No-Preemption : A resource cannot be taken from a process unless the process the resources.

④ Circular Wait : A set of processes are waiting for each other in circular form.

DATE : / /
PAGE NO. :

- Methods for handling deadlock

① Deadlock prevention or avoidance
② Deadlock detection & Recovery
③ Ignore the problem altogether

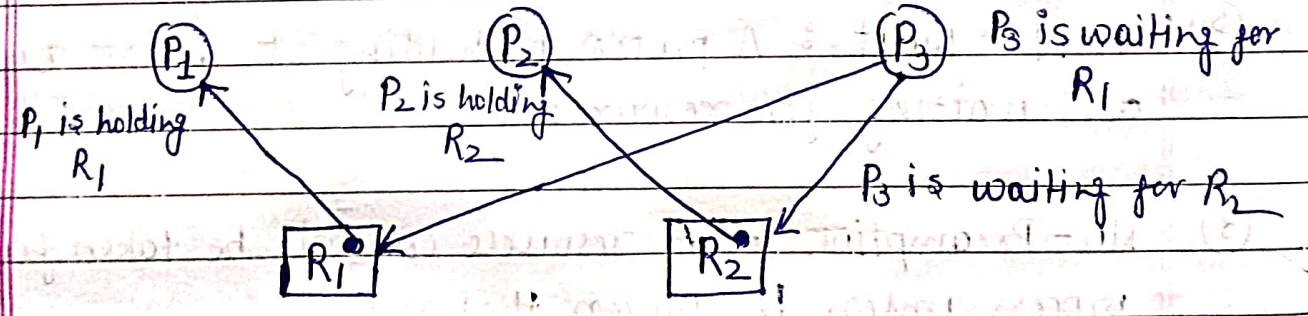
- Resources Allocation Graph:

Vertices : $P = \{P_1, P_2, P_3, P_4, P_5, P_6 \cdots P_n\}$
$R = \{R_1, R_2, R_3, R_4, R_5 \cdots R_n\}$

Edges : $P_i \rightarrow R_j$ Request Edge
$R_j \rightarrow P_i$ Allocation edge

$\dashrightarrow$ claim edge

Ex: $\{R_1 \rightarrow P_1\}, \{R_2 \rightarrow P_2\}, \{P_3 \rightarrow R_1\}, \{P_3 \rightarrow R_2\}$



Single instance resource without deadlock

- Deadlock Prevention or avoidance

→ To deadlock prevention, we have to eliminate all the four conditions:

① Mutual exclusion ② Hold & wait
③ No Promption ④ circular wait

DATE : / /
PAGE NO. :

- Deadlock Prevention & Avoidance
- Deadlock Detection & Recovery

→ Deadlock Detection

① If resources have a single instance.
⇒ In this case, we can run an algorithm to check for the cycle in the Resource allocation Graph.
The presence of a cycle in the graph is a sufficient condition for deadlock.

② If there are multiple instances of resources
⇒ Detection of the cycle is necessary but not sufficient condition for deadlock detection. In this case, the system may or may not be in deadlock varies according to different situations.

→ Deadlock Recovery

① Killing the process.
② Resource preemption

★ Bankers Algorithm's (Deadlock Avoidance Algorithm)

→ Bankers Algorithms also used for Deadlock detection.

Safety Algorithm

1) Let Work and Finish be vectors of length 'm' and 'n' respectively.
   Initialize : Work = Available
   Finish[i] = false , for i = 1, 2, 3, 4 ---- n

2) Find an i such that both
   a) Finish[i] = false
   b) $Need_i$ <= Work
   if no such i exists goto step(4)

3) Work = Work + Allocation[i]
   Finish[i] = true
   goto step ②

4) If Finish[i] = true for all i
   then the system is in a safe state

* Resource-Request Algorithm

1) If $Request_i$ <= $Need_i$
   Goto step (2) ; otherwise raise an error condition.

2) If $Request_i$ <= Available
   Goto step ③ ; otherwise P_i must wait

3) Available = Available - $Request_i$
   Allocation = $Allocation_i$ + $Request_i$
   $Need_i$ = $Need_i$ - $Request_i$

POORNIMA

Date	Unit No.	Lecture No.	Faculty	Subject Name	Subject Code	Main Topics:-

Ex.

Process	Allocation	Max	Available (Work)	Need
	A B C	A B C	A B C	A B C
P_0	0 1 0	7 5 3	3 3 2	7 4 3
P_1	2 0 0	3 2 2		1 2 2
P_2	3 0 2	9 0 2		6 0 0
P_3	2 1 1	2 2 2		0 1 1
P_4	0 0 2	4 3 3		4 3 1

Apply Safety Algorithm

step ①:

Work = Available

Work = | 3 | 3 | 2 |

Finish = | f | f | f | f | f |

step ②:

for i=0

$Need_0$ = 7, 4, 3

Finish[0] is false and Need <= Work

743 <= 332 X

for i=1 ↙ go to step 2 again

$Need_1$ = 1 2 2

Finish[1] is false and Need <= Work

122 <= 332 ✓ → go to step ③

Main Ideas, Questions & Summary:

Library / Website Ref.:-

$$Work = Work + Allocation \quad (332 \quad 200)$$

Work = 5 3 2

Finish =	f	f	f	f	f

for i=2

$Need_2 = 6, 0, 0$

Finish[2] is false and ($Need_2 <= work$: 600 <= 532) X

So P_2 must wait, go to step (2)

for i=3

Need = 0,1,1

0,1,1 <= 5,3,2 (Need <= Work) ✓

P_3 is safe sequence

go to step-(4)

Work = work + allocation

= 532 + 211 = 743

for i=4

Need = 4,3,1

431 <= 743 (Need ≤ Work) ✓

go to step-(4)

Work = 743 + 002 = 745

P_4 is safe sequence

for i=0

Need = 743

743 <= 745 (Need ≤ Work) ✓

go to step-(2)

Work = 745 + 0,1,0 = 755

P_0 is safe sequence

for i=2

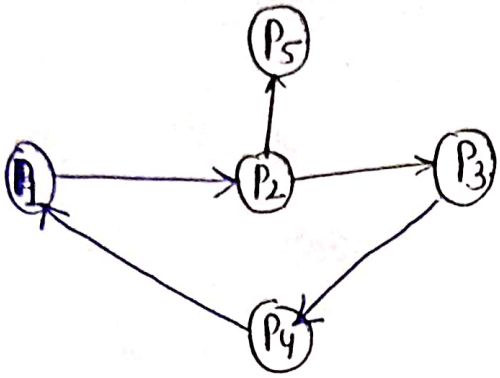
Need = 6,0,0

600 <= 755 (Need ≤ Work)

So P_2 is also safe sequence

The safe sequence is P_1, P_3, P_4, P_0, P_2 Ans

✱ Wait for graph



For single instance of each resource type

→ For several Instances : wait for graph is not applicable for ~~several~~ single instances of each resource type. It will same as bankers algorithm.

Compare this algo. with bankers algorithm

step-① let Work and finish be the vectors of length m & n respectively

Initialize : Work = Available

Finish[i] = false , for i = 1, 2, 3, 4, ——— n

if (allocation$_i$ ≠ 0)

if (allocation$_i$ = 0)

Finish[i] = true

step-②: Find an index i such that

a) Finish[i] = false

b) Request$_i$ = Work

If no such condition exist then go to step-④

step-③: Work = work + allocation[i]

Finish[i] = true

go to step-②

POORNIMA

Date	Unit No.	Lecture No.	Faculty	Subject Name	Subject Code	Main Topics:-

step-(4): If Finish[i] == false for some i ~~$1 \leq i \leq n$~~

$$0 \leq i < n$$

then system is in deadlock state, and the process P_i is the deadlock.

(2) Let System recover from it ~~of the~~ automatically

breaking deadlock
- Process terminator
  - aboart one
  - abort all at once
- Preempt resources

Q. How to find the deadlock for multiple system server.

- Resource Pramption

(1) select the victim

(2) Rollback

(3) starvation: We ~~process~~ again & again prompt the resource from where the prous is in deadlock state.

POORNIMA

Date	Unit No.	Lecture No.	Faculty	Subject Name	Subject Code	Main Topics:-

Device Management

⇒ Device Management in OS controlling the IP/OP devices like disk ~~management~~, microphone, keypad.
A process may required various resources including memory, file, access & access to disk devices & others.

⇒ The system has multiple devices & in order to handle these physical or virtual devices, the OS req. separated program known as device controller.

→ Types of Devices

[1] Block devices : These devices provides main interfaces to all disk devices are used to store files on a system.
Direct access to a block device is Availble.

[2] character devices : These devices control other devices.
Eg. mouse, keyboard, printer, USB etc.

[3] Network devices : User can not directly transfer data to network devices, they must connect directly by opening the connection to kernal by network subsystem.

→ Types of peripheral Devices

[1] Dedicated : These type of devices are ~~devic~~ dedicated for only single task.
Eg : Mouse

[2] Shared : These type of devices are for multiple task
Eg. Pendrive, disk, SSD. etc.

[3] Virtual : That are the hybrid of two or more devices.
Eg. processer

Main Ideas, Questions & Summary:

Library / Website Ref.:-

- Categories of devices
  - → I/O devices
  - → storage devices
    1. serial access devices
    2. completely direct access
    3. direct access storage devices

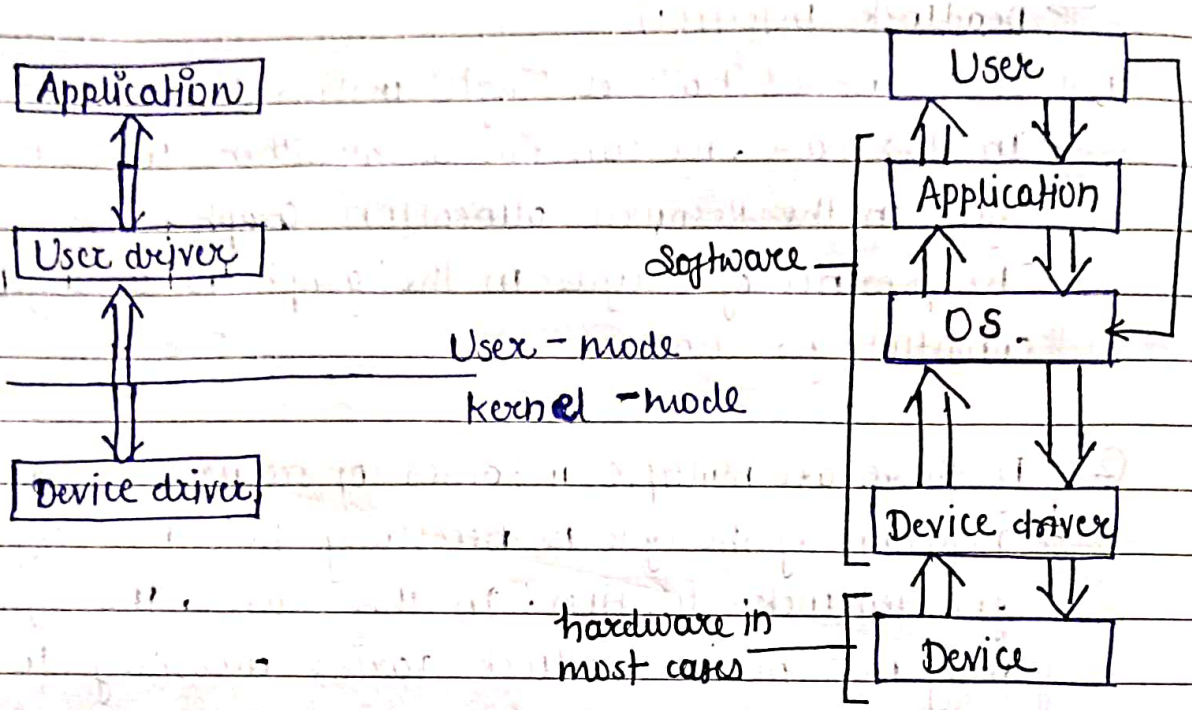
- **Bad Block** : The particular section of a devices that becomes defective is known as bad block.
  This part is no more reliable for storing data.

- Handling of Bad block
  1. Manually : In which we find the bad block. This process may take long time.
  2. sector slipping : In which we skip the defected part in sectors of the disk.
  3. Sector sparing : In which we copy the all data is spear and play.

DATE : / /
PAGE NO. :

⊛ Device Driver & its purpose

⇒ Device Drivers are essential for a computer system to work properly because without a device driver the particular hardware fails to work accordingly.



- Types of Device Driver

① Kernal - mode Device Driver

⇒ It includes some generic hardware that loads with the OS as part of the OS there are BIOS, motherboard, processor etc. These include the minimum system requirement device drivers for each OS.

② User - mode Device drivers

Ex: The user needs any plug & play action that comes under this.

③ Virtual Device Drivers

⇒ Sometimes we use the same hardware virtually at that time virtual drivers control / manages the dataflow from the different applications used by different users to the same hardware.

DATE : / /
PAGE NO. :

★ Disk Scheduling Algorithm

→ Terminologies :

- Seek time : It is time taken to local the disk arc to a specific track. Seek time is minimum than get optimal solution.

- Rotational Latency : To rotate into aqusition, so that it can access/ read / write heads.

- Transfer time : Time taken to transfer data from hard disk to main memory.

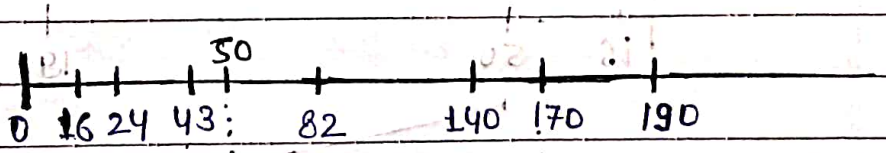
{ Disk access time = Seek time + Rotational latency + Transfer time }

⊛ Disk Scheduling Algorithm's

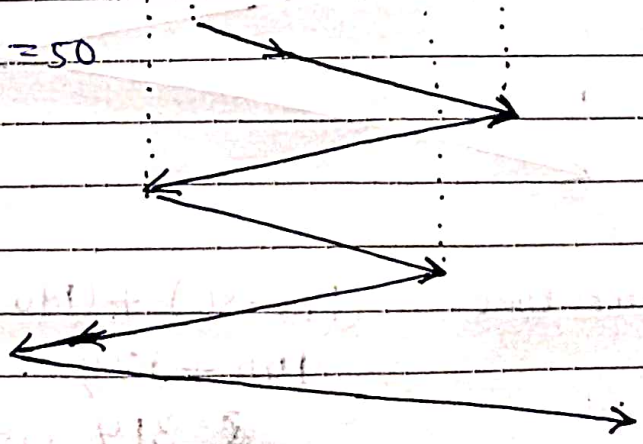
[1] FCFS (First Come First Serve)

Track NO : 82, 170, 43, 140, 24, 16, 190

disk arm :



start disk arm = 50



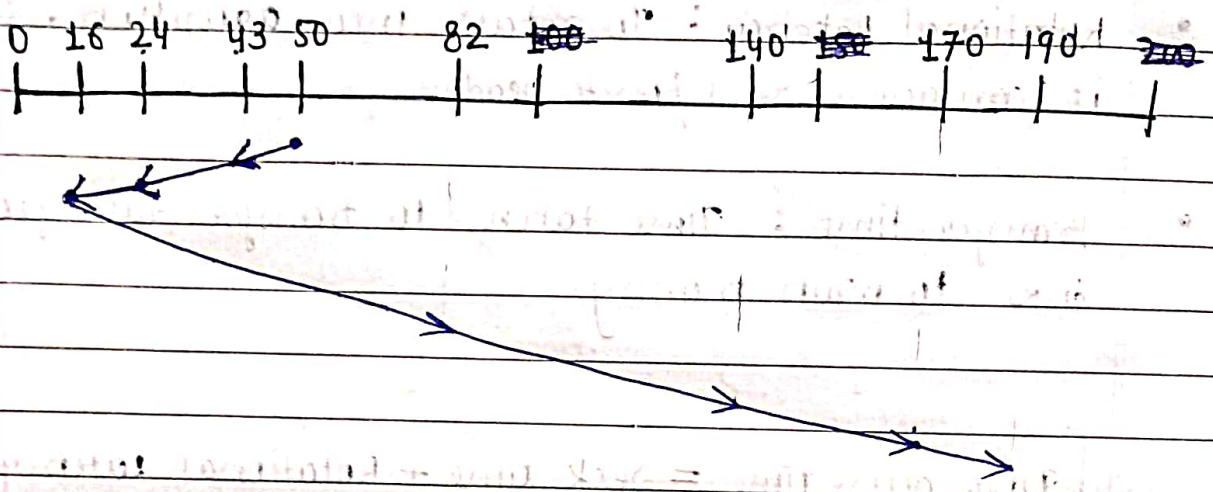
DATE : / /
PAGE NO. :

So, the seek time = (82-50) + (170-82) + (170+43) + (140-
+ (140-24) + (24-16) + (190-16) = 642 Ans

② SSTF (Shortest Seek time First)

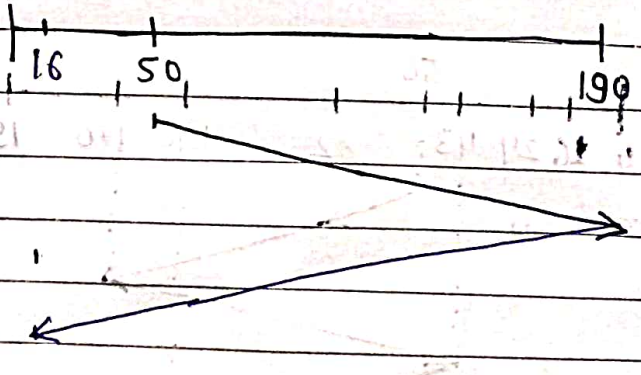
Ex.

order of request : (82, 170, 43, 140, 24, 16, 190)
current pos. of w/R head = 50



Total seek time : (50-43) + (43-24) + (24-16) + (82-16)
+ (140-82) + (170-140) + (190-170)
= 208 Ans

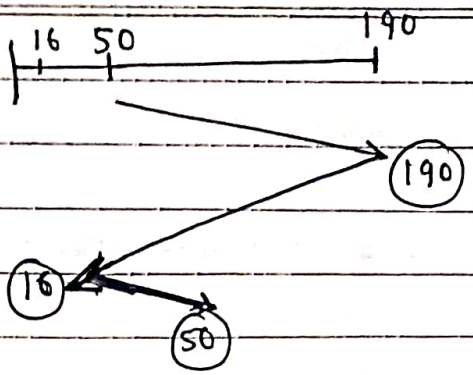
③ SCAN (towards the larger value)



Total seek time = (190-50) + (190-16)
= 140 + 174
= 314 Ans

DATE : / /
PAGE NO. :

(4) CSCAN (circular scan)



(5) LOOK SCAN (towards the larger value)
(6) CLOOK SCAN (circular Look scan)
(7) RSS (Random scheduling)
(8) LIFO (Last in first Out)
(9) N-step SCAN (N-step Look algorithm)
(10) FSCAN

POORNIMA

Unit-04 : File concepts

Date	Unit No.	Lecture No.	Faculty	Subject Name	Subject Code	Main Topics:-

• File

⇒ Logically storing of data is File.

→ When we logically save the data onto the physical device, we need to map physical & logical container, the result in into the problem internal Fragmentation.

→ The solution of this problem is packing.

→ No of logical records into physical block

Grouping

File attribute: size, location, identifier, name, location, details, date & time.

Operations: copy, paste, delete, truncate.

	extension	Functionality
Executable File	.exe	Ready to run machine language program.
Source code file	.c, .cpp, .py, .html	Run the source code of programming language
Text Files	.txt, .doc	Contain the text of the file
Archieve file	.zip, .jar, .tar, .arc	compress / lopped file and decrease file size.
Image files	.jpg, .jpeg, .png	contain pictures

Main Ideas, Questions & Summary:

Library / Website Ref.:-

Audio files	.mp3,	Store ~~Video~~ Audio files
Video files	.mp4	store video files
Tabular data	.xls, xml, .excel .csv	store tabular data

File structure: offset

Internal on offset is located with the file, disk system have well defined block size; according to the size of sector.

The files mapped on physical devices are non-volatile. So the contents are percistant ~~while~~ reboot or power failure.

★ File Access method

1. Sequential Access

All the record are process one after another, a reads operation reads next portion of file & automatically advances a file pointer which tracks I/P & O/P locations.

2. Direct Access Whole disk is viewed as numbered sequence of block, so we can access (block 7, block 8 ---- block n).

3. Other Access

⇒ It involves the index & contains pointers

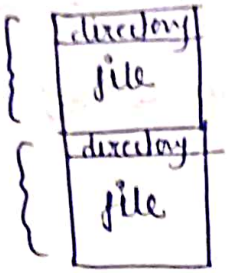
POORNIMA

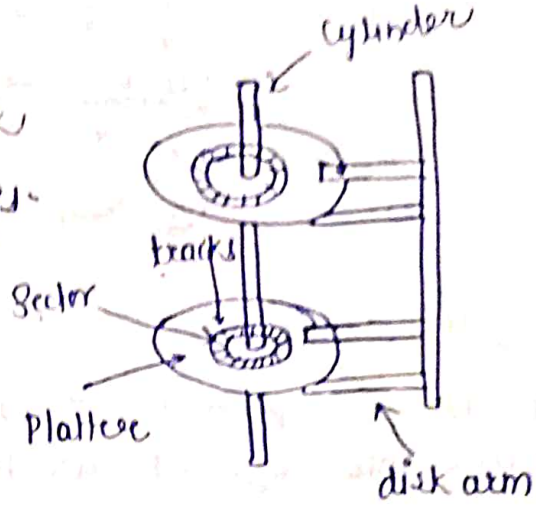
Date	Unit No.	Lecture No.	Faculty	Subject Name	Subject Code	Main Topics:-

★ ~~Directory~~ Structure

Directory

⇒ There are terabytes of files stored on a disk
So, to manage all the files we need directories.

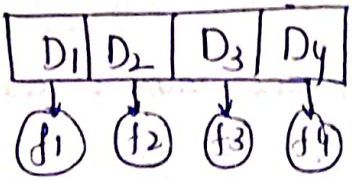




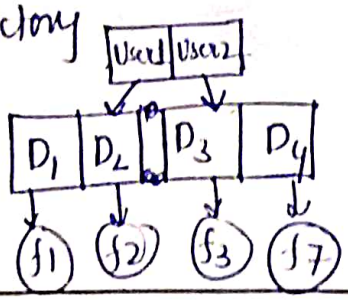
• Operations performs on directry
① Search, ② rename, ③ list a directory ④ create ⑤ delete.

⊛ Common schemes for defining logical structure of directory

① Single level directry: All the files are kept at one place.
→ Searching is easy



② Two level directory



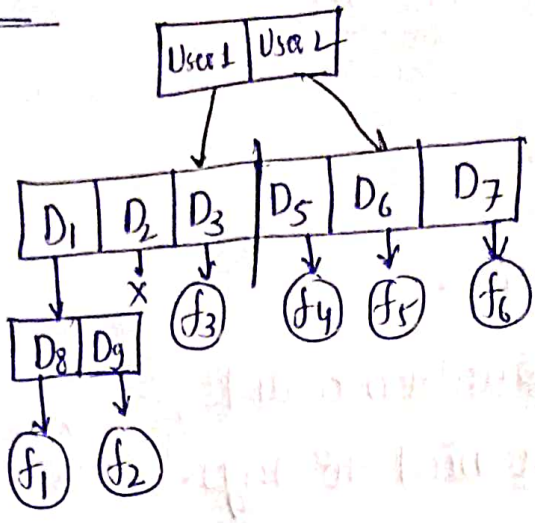
→ Searching complex
→ User can not share files

Main Ideas, Questions & Summary:

Library / Website Ref.:-

③ Tree Structure

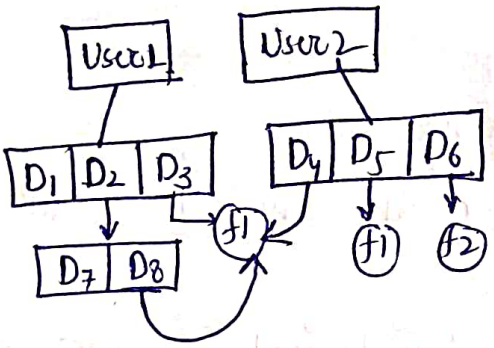
→ Scalable



④ Acyclic graph

⇒ It is a graph with no cycle. It allows sub directory with same name into different directory.

⇒ you ~~should~~ should have backup of data.



⑤ Cyclic graph / General Graph

⇒ cycle are allowed

→ multiple directly are allowed for more than 1 parent directory.

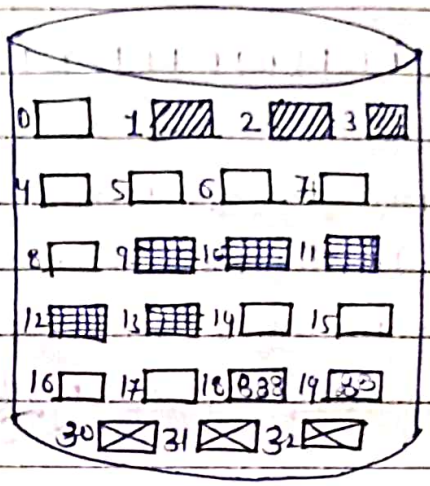
★ FILE ALLOCATION METHODS :

① Continuous Allocation

⇒ A single continuous set of blocks is allocated to a file at the time of file creation.

→ static allocation

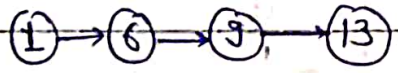
File Name	start block	Length
File A	2	3
File B	9	5
File C	18	2
File D	30	2

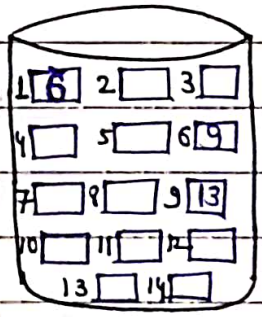


② Linked Allocation (Non-contiguous Allocation)

⇒ Each block contains a pointer to the next block in the chain.

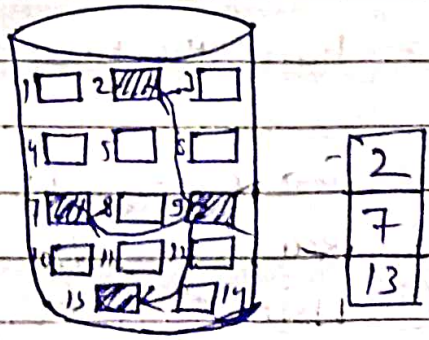
→ dynamic allocation





③ Indexed Allocation

⇒ The file allocation table contains a separate one-level index for each file : The index has one entry for each block allocated to the file.



DATE : / /
PAGE NO. :

★ Disk Free Space Management

① Bit tables

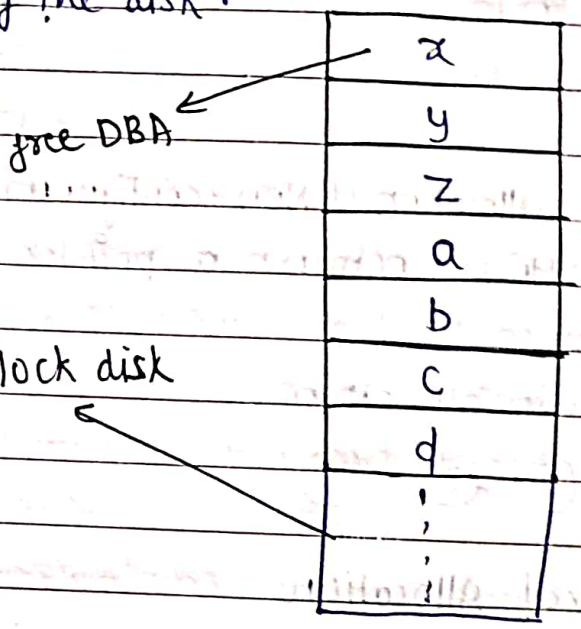
⇒ This method uses a vector containing one bit for each blocks on the disk. Each entry for a 0 corresponds to a free block and each 1 corresponds to a block in use.

Ex.

0001101011100 1

② Free Block List

⇒ In this method each block is assigned a number sequentially and the list of the numbers in a reserved block of the disk.



★ Protection in File System

⇒ The files which have direct access of the any user have the need of protection.

Several different types of operations can be controlled :

1. Read
2. Write
3. Execute
4. Append
5. Delete
6. List

DATE : / /
PAGE NO. :

Unit-05 - UNIX & LINUX OS

★ Real time Operating system

⇒ A Real time operating system, commonly known as an RTOS, is a software component that rapidly switches b/w tasks, giving the impression that multiple programs are being executed at the same time on a single processing core.

It must be accepted and processed in a short time or within certain deadlines.

Such applications are industrial control, telephone switching equipment, flight control & real time simulations.

In RTOS, processing time is measured in tenths of seconds.

Three types of RTOS

① Hard Real time OS : These OS guarantee critical task in time limit.
② Soft Real time OS: Some relaxation in time limit.
③ Firm Real time OS: It follow deadlines as well. In spite of its small impact, missing a deadlines can have unintended consequences, including a reduction in the quality of the product.
Ex. Multimedia applications

- Advantage :
① Maximum consumption
② Task shifting
③ Focus on application
④ Error free
⑤ Memory allocation

- Disadvantage
① Limited tasks
② Use heavy system resources
③ complex Algorithm
④ Device Drivers & Interrupt Signals

DATE : / /
PAGE NO. :

★ Mobile Operating System

⇒ A mobile operating system is an operating system that helps to run other application software on mobile devices. It is a same kind of software as Linux & windows. but how they are light and simple to some extent.

The OS found on smartphones include Symbian OS, iphone OS, RIM's Blackberry,

→ Popular platforms of the Mobile OS

① Android OS : Based on Linux kernel & open source software developed by Google. The first android device was launched in 2008.

② Bada (Samsung Electronics)

③ BlackBerry OS

④ iphone os / ios