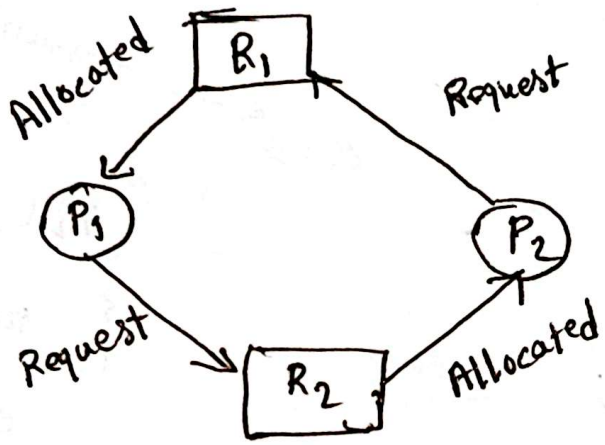
8th class 20-10-2024

Deadlock: Deadlock occurs when a process or thread enters a waiting state because a requested system resource is held by another waiting process which in turn is waiting for another resource held by another waiting process.



Process Synchronization:

Independent Process: are those process that can neither affect other process or be affected by other processes when executing in the system.

Cooperating process: are those process that can affect or are affected by other process when executing in the system.

These processes may share resources, data, variable, code with each other.

Problem :

c = 5 ← resource (shareable)

```
P1()
{ read (c);
  c++ ;
  write (c);
}
```

```
P2()
{ read (c);
  c++ ;
  write (c);
}
```

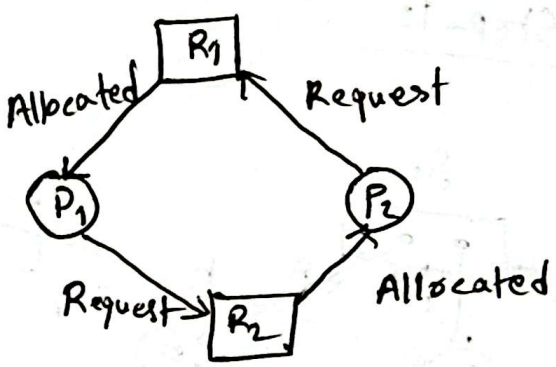
Race condition: A race condition occurs when two or more processes can access shared data and they try to change it at the same time.

9th class

21-10-2024

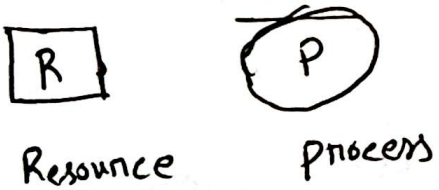
Deadlock: Necessary Condition

1. Mutual exclusion (No shareable resources)
2. Hold & wait
3. No preemption
4. Circular wait.

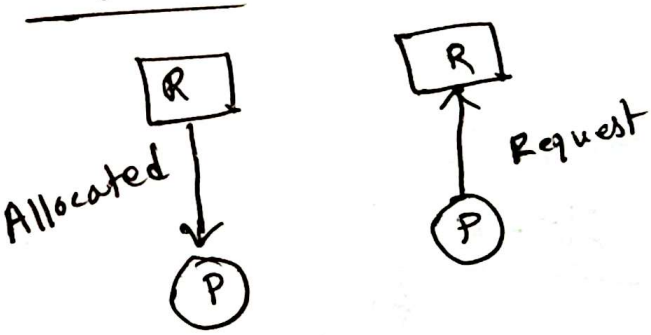


Resource Allocation Graph :

Vertex



Edge



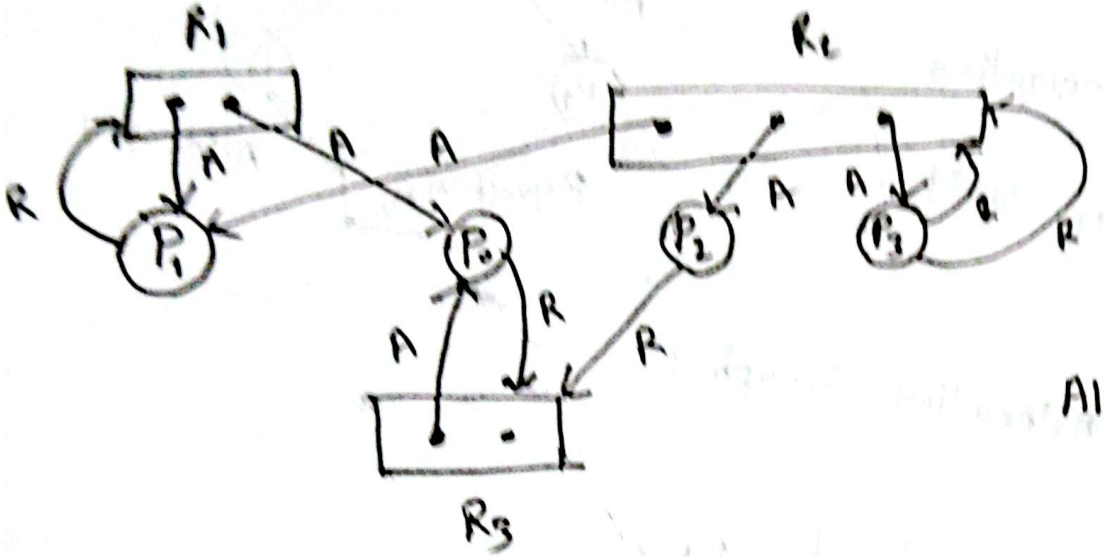
Single instance :

Cytomis®

Multiple Instance :

R_2

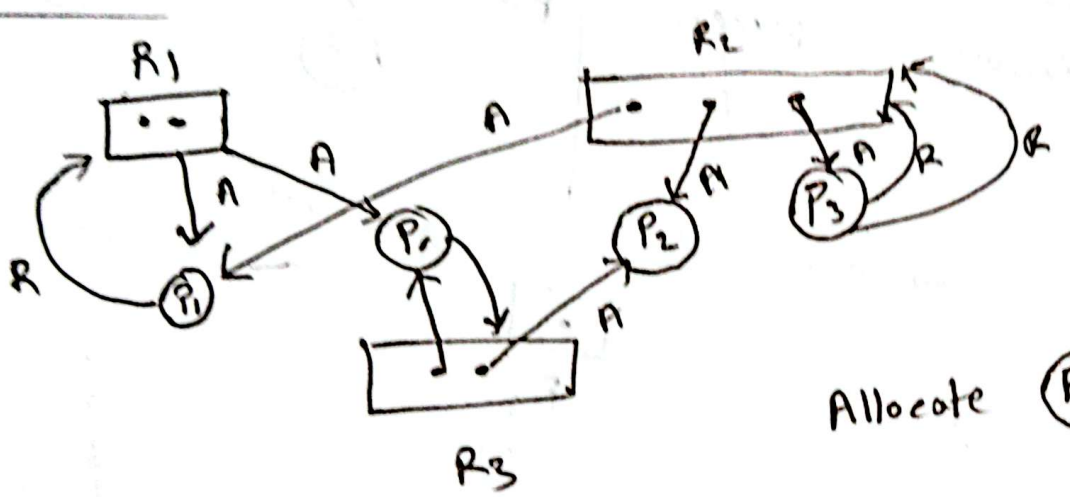
Step-1:



Allocate R_3 into P_2

Executed : P_2,

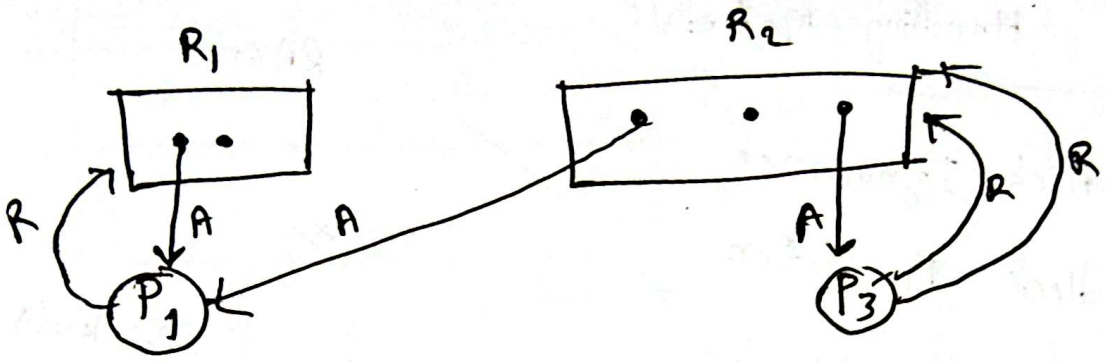
Step-2 :



Allocate R_3 into P_0

Executed : P_2, P_0

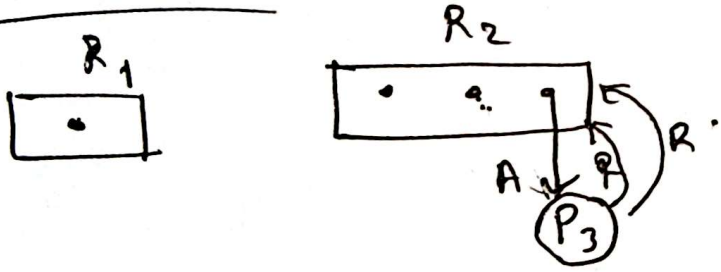
Step - 3 :

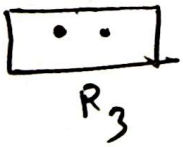


Allocate (R1) into (P1)

Executed : P2, P0, P1

Step - 4 :





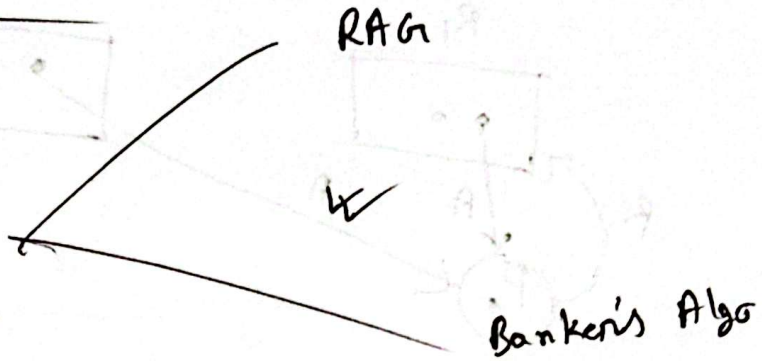
Allocate ~~two~~ (R2) into two instance of (P3)

Executed : P2, P0, P1, P3

10th class

28-10-24

Deadlock Handling Methods :



(I) Deadlock Ignorance

(II) Deadlock Prevention

(III) Deadlock Avoidance

(IV) Deadlock detection & Recovery.

Q. Deadlock handling methods, where to use, how to use (Theory in brief)

Banker's Algorithm

Here $R_1 = 10$

$R_2 = 5$

$R_3 = 3$

Process	Max need	Allocated resources	current resource need	Available resources
	R_1 R_2 R_3	R_1 R_2 R_3	R_1 R_2 R_3	R_1 R_2 R_3
P_0	7 5 3	0 1 0	7 4 3	3 3 2
P_1	3 2 2	2 0 0	1 2 2	5 3 2
P_2	9 0 2	3 0 2	6 0 0	7 4 3
P_3	2 2 2	2 1 1	0 1 1	7 5 3
P_4	4 3 3	0 0 2	4 3 1	10 5 5
		total : 7 2 5		10 5 7

Safe Sequence:

P_1, P_3, P_0, P_2, P_4

H.W → (RAG ⇒ Bankers previous class)

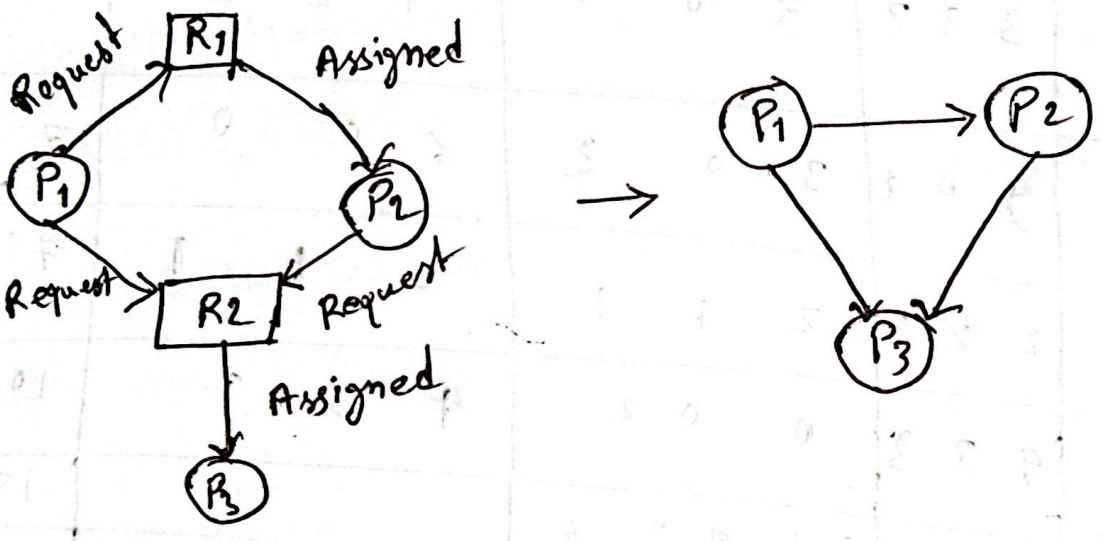
Bankers algorithm → (code in Lab)

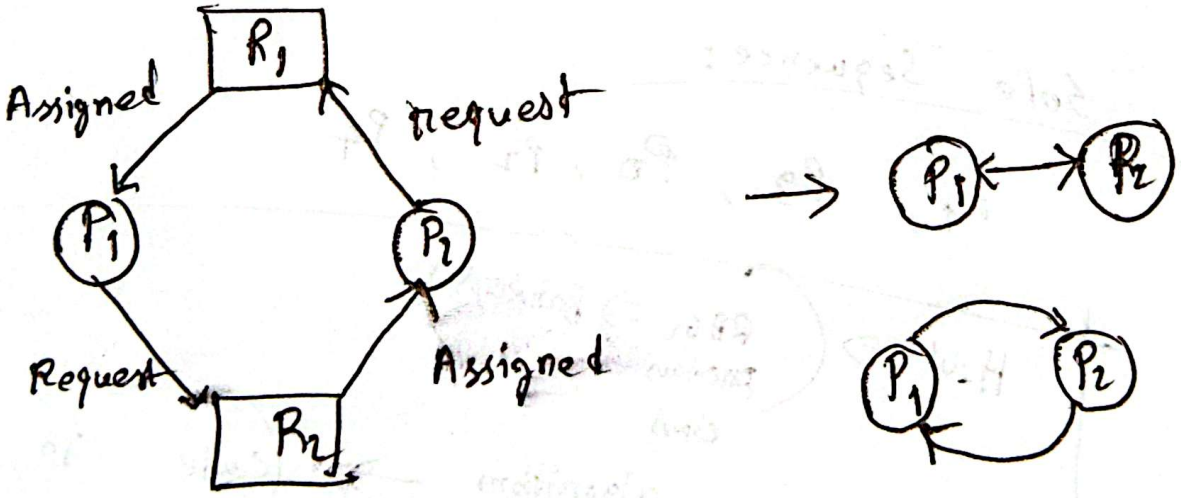
03-11-2024

11th class

Deadlock Detection & Recovery

⊛ Wait for graph (single instance)





⊛ Process Termination:

(i) Abort all deadlock process

(ii) Abort one process at a time until the deadlock cycle is eliminated.

⊛ Resource Preemption:

Selecting a victim:

(i) As in process termination, we must determine the order of preemption to minimize the cost.

(ii) Rollback: Rollback the process which was selected as a victim. Rollback the process to some safe state or abort the process and restart it.

(iii) Starvation: It may happen that the same process is always picked as victim. As a result, the process never complete its task, we must ensure that a process can be picked as a victim for a finite number of time.

04-11-2024

12 th class

Q. Which type of algorithm is used in which of type of operating system?

(Advantage & disadvantage of)

Critical Section (Self study) Important for exam (gate smasher)

A process has three types of section

→ Entry (Starting)

→ Critical Section

→ Exit (Ending)

Q. What is critical section? with example (code can be written as example)

Synchronization Algorithm - 3 condition

⇒ Mutual exclusion (process complete other process start)

⇒ Progress

⇒ bounded wait

Semaphone: is an integer variable that is used to solve the critical problem by using two atomic operation wait and signal that are used in process synchronization.

Entry Section → (wait)

Critical Section

Exit Section → (Signal())

```
wait (s)
{ while (s<=0) { }
  s = s-1;
}
```

```
Signal (s)
{
  s = s+1;
}
```

```
do { wait(s)
     critical section
     signal()
} while (true)
```

10-11-2024

13th class

Memory Management Function: The memory management function keeps track of the status of each memory location, either allocated or free. It determines how memory is allocated among processes, deciding which gets memory. When memory is allocated, it determines which memory location will be assigned. It tracks when memory is unallocated and updates the status.

Swapping : Hard-disc → Ram

Swapp out : Ram to Harddisc

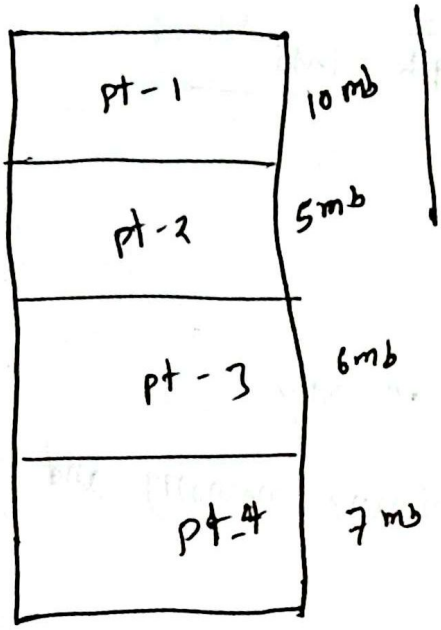
Q. What is swapping operating system? Use of swapping?

Q. What is contiguous memory allocation and how it works?

What is non-contiguous memory allocation and how it works?

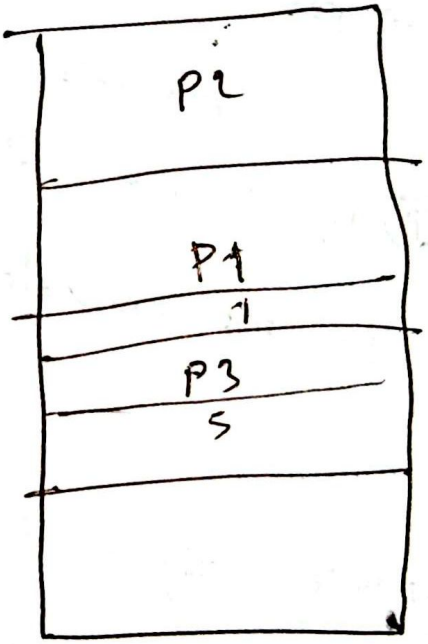
Memory allocation:

- Contiguous
- Non-contiguous



* Contiguous → 1 process only 1 part
* Non contiguous → 1 process can be assigned many part

P1 → 5mb
P2 → 4 mb
P3 → 15 mb
P4 → 13 mb



Q. What is fragmentation ?
(External & Internal fragmentation)

Example

19th class

12-11-2024

Page Replacement Algorithm:

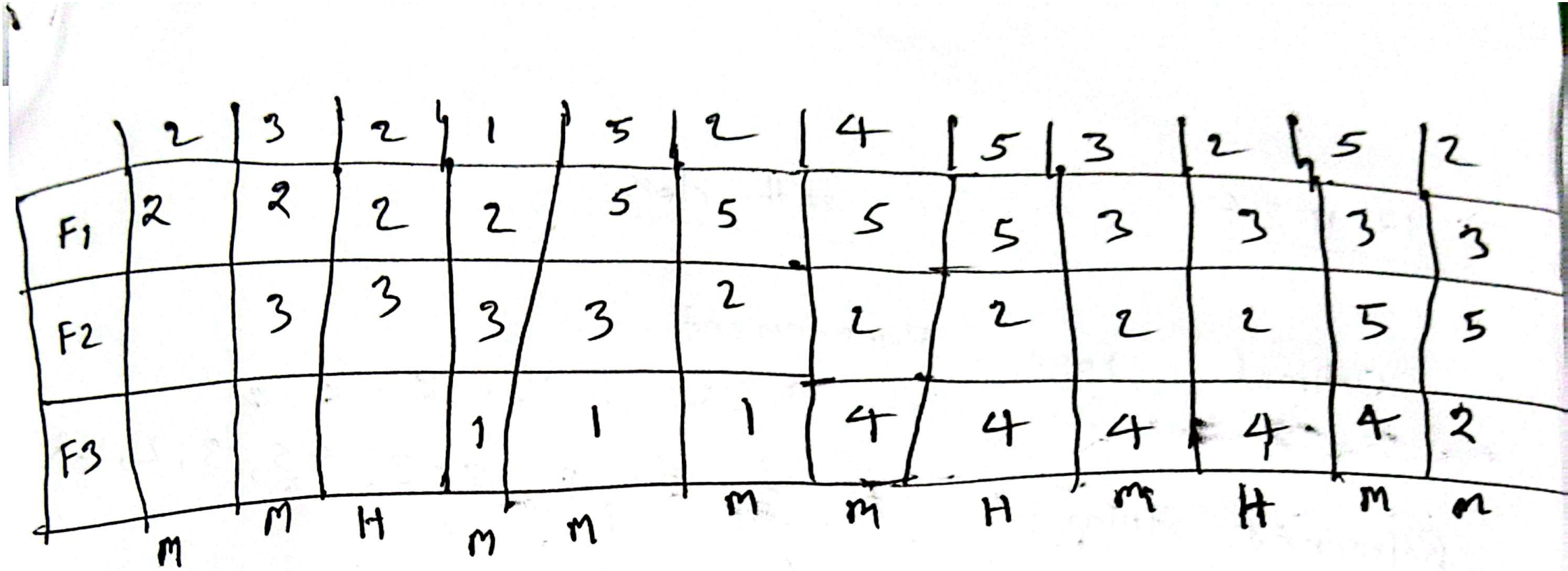
Hit : Page in the primary memory

Miss : (I) Page is not in the primary memory and space is available

(II) Page is not in the primary memory and space is not available.

FIFO

Reference String : 2, 3, 2, 1, 5, 2, 4, 5, 3, 2, 5, 2. How find the Hit and Miss Ratio.



Hit Ratio : $\frac{3}{12} = \frac{1}{4}$

Miss Ratio : $\frac{9}{12} = \frac{3}{4}$

17-11-24

15th class

Optimal Page Replacement.

Reference String: 2, 3, 2, 1, 5, 2, 4, 5, 3, 2, 5, 2

Now find out hit and miss ratio.

No. of frames → 3

S/n:

	2	3	2	1	5	2	4	5	3	2	5	2
F_1	2	2	2	2	2	2	4	4	4	4	4	4
F_2		3	3	3	3	3	3	3	3	2	2	2
F_3				1	5	5	5	5	5	5	5	5
	M	M	H	M	M	H	M	H	H	M	H	H

Hit → 6

Miss → 6

Hit ratio: $\frac{6}{12} = \frac{1}{2}$

Miss ratio: $\frac{6}{12} = \frac{1}{2}$

Least Recently used (Algorithm) LRU

Reference string: 2, 3, 2, 1, 5, 2, 4, 5, 3, 2, 5, 2

No of frames 3.

Also find out miss and hit ratio.

	2	3	2	1	5	2	4	5	3	2	5	2
F1	2	2	2	2	2	2	4	4	3	3	3	3
F2		3	3	3	5	5	5	5	5	5	5	5
F3				1	1	1	1	1	1	2	2	2
	M	M	H	M	M	M	M	H	M	M	H	H

HR → $\frac{5}{12}$

MR → $\frac{7}{12}$

Hit → 5

Miss → 7

02-P2-24

Asif Sir

Operating System

Disk Scheduling: (FCFS)

A disk contains 200 tracks (0 - 199)

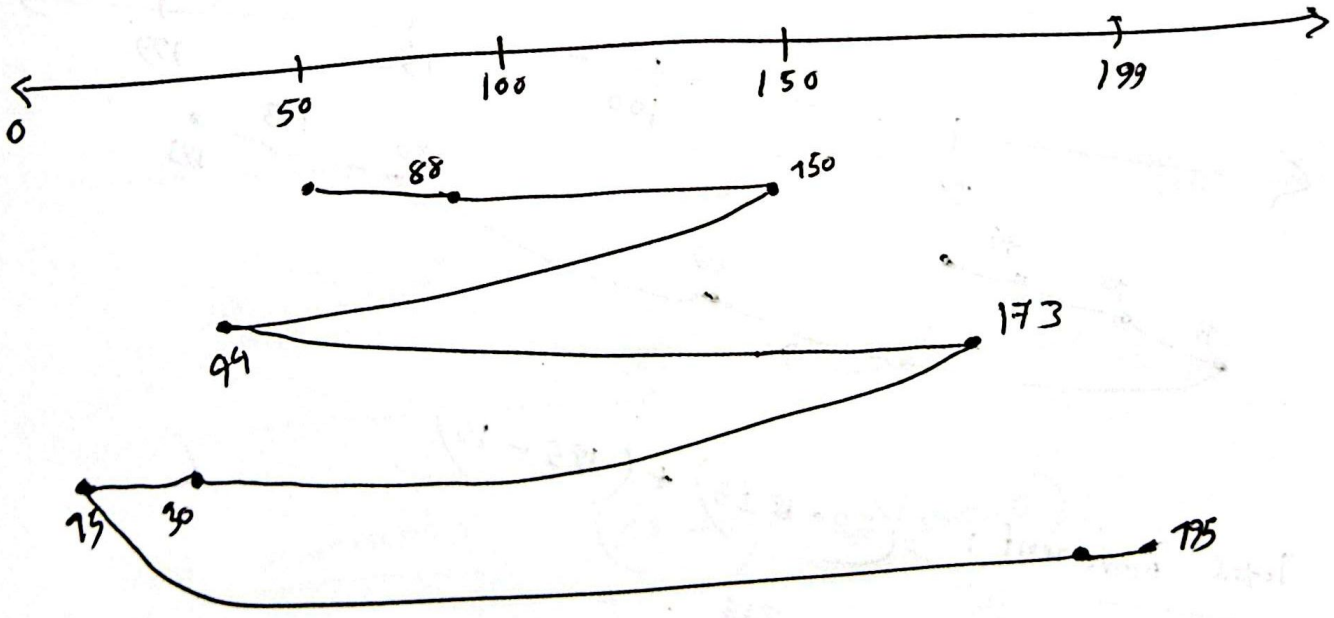
Request queue contains track no:

88, 150, 44, 173, 30, 15, 195

current position of R/W head = 50

Calculate total number of track movement.

Sln:



Total movement

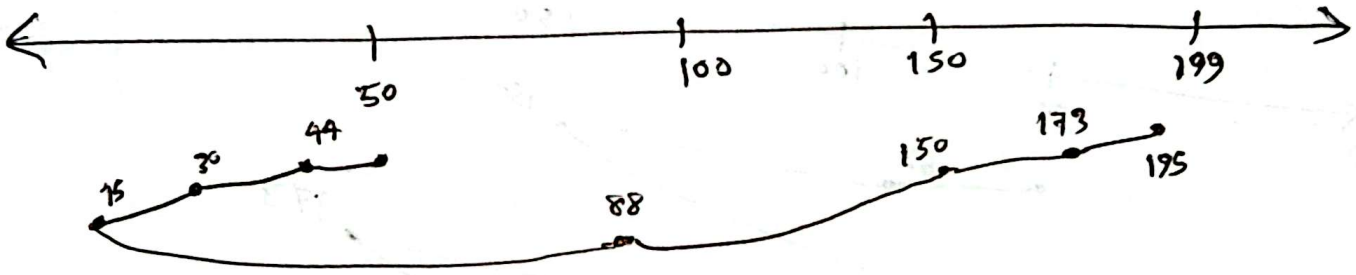
$= (150 - 50) + (150 - 44) + (173 - 44) +$

$(173 - 15) + (195 - 15)$

$= 573$

SSTF (Shortest seek time first)

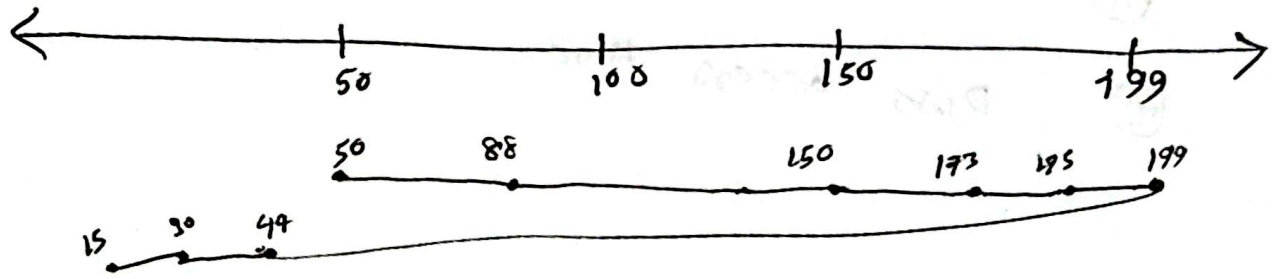
Question → Previous one



Total movement : $(50 - 15) + (195 - 15)$

$= 215$

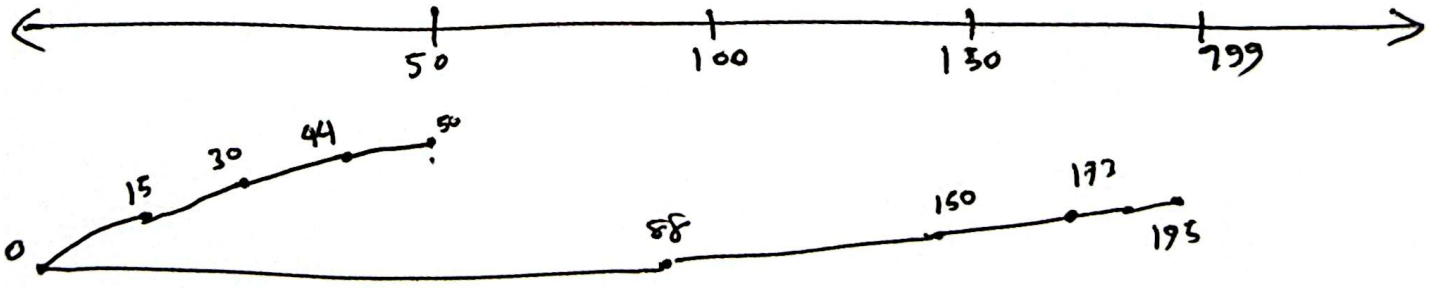
Scan (Increasing)



Total movement: (199 - 50) + (199 - 15)

= 333

Scan (Decreasing)



Total movement: (50 - 0) + (185 - 0)

= 50 + 185 = 245

Theory (self study)

(A) Disk Architecture

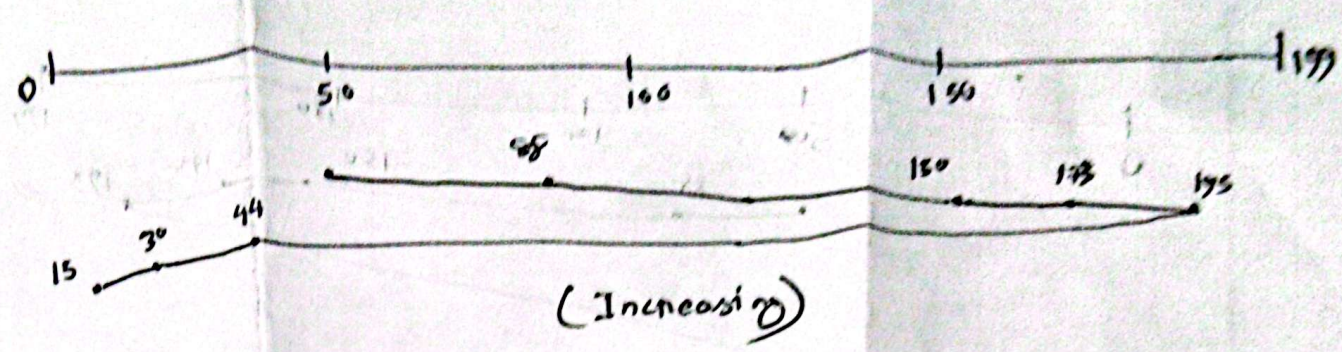
(N) Disk access time

Sub: Operating System

Date: 08 / 10 / 2024

Sun Mon Tue Wed Thu Fri

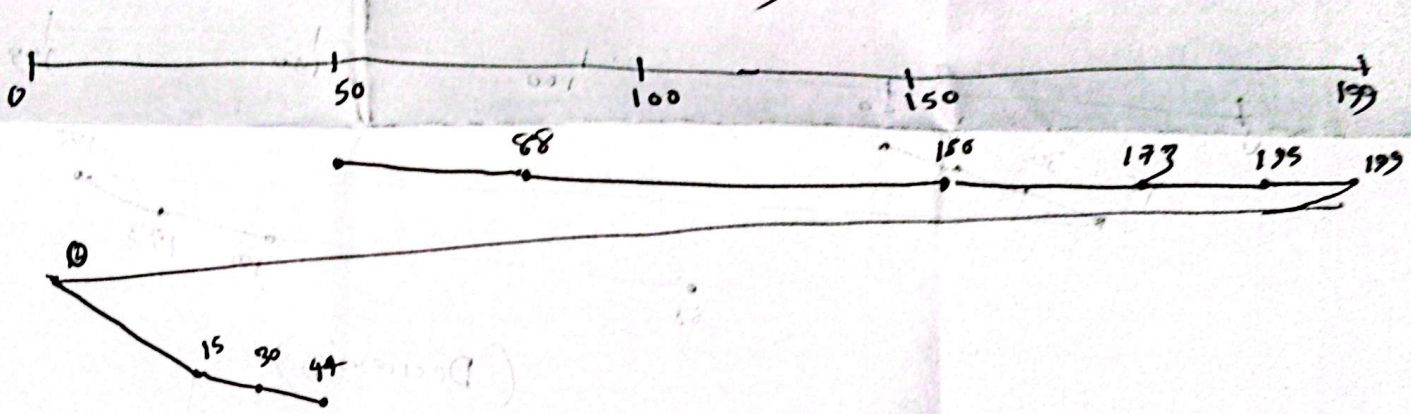
Look :

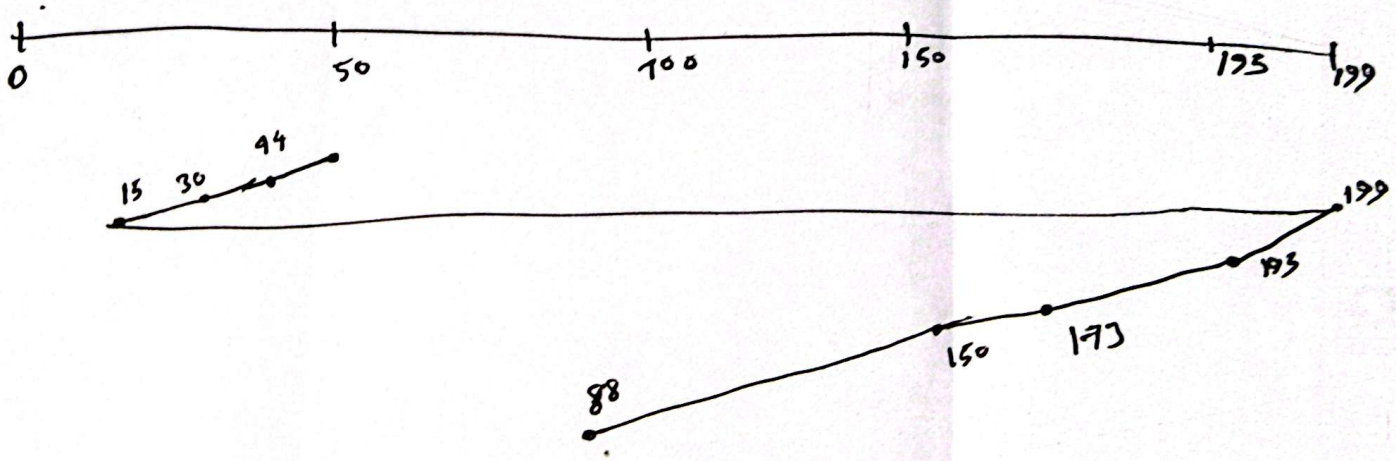


(Increasing)

C-SCAN :

(Increasing)

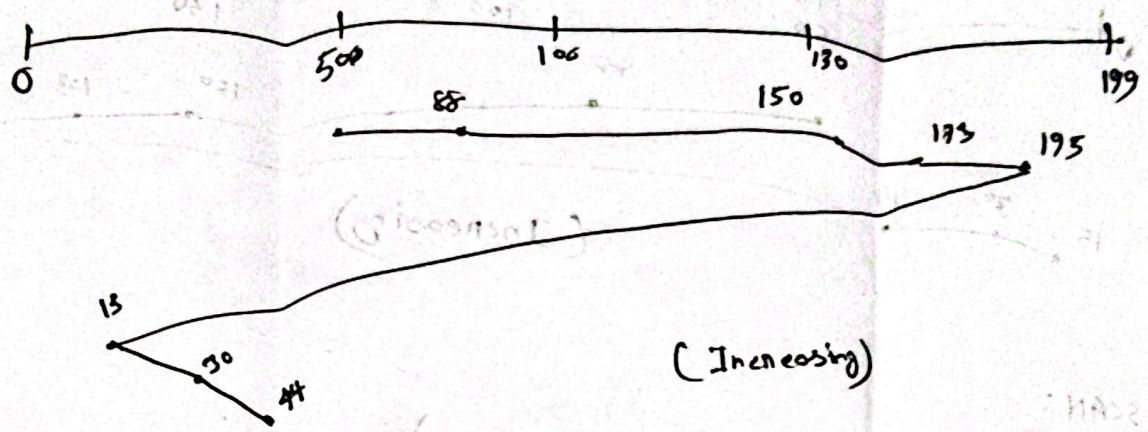




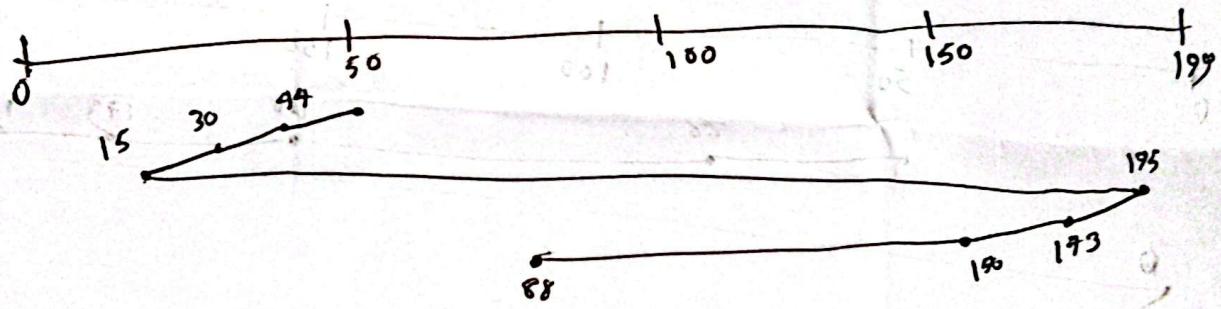
(Decreasing)

Sub: _____ Date: Sat Sun Mon Tue Wed Thu Fri

C Look :



(Increasing)



(Decreasing)

Theory :

OS

class Note by Sami

Second chance / clock

Page Replecement Algorithm

Reference String: 2, 3, 2, 1, 5, 2, 4, 5, 3, 2, 5, 2

	2	3	2	1	5	2	4	5	3	2	5	2	7	5
F_1	2	2	2	2	2	2	2	2	3	3	3	3	7	7
F_2		3	3	3	5	5	5	5	5	5	5	5	5	5
F_3				1	1	1	4	4	4	2	2	2	2	2
	M	M	H	M	M	H	M	H	M	M	H	H	M	H

Ref_string: 0, 4, 1, 4, 2, 4, 3, 4, 2, 4, 0, 4, 1, 4, 2, 4, 3, 4

2	8
7	8
5	5
2	2
H	

	0	4	1	4	2	4	3	4	2	4	0	4	1	4	2	4	3	4
F_1																		
F_2																		
F_3																		