Meet Raut
S21 84

Assignment 2

Q.1)

Ans a) Deadlock avoidance and prevention.

Deadlock avoidance	Deadlock prevention
i) It involves dynamically examining resource allocation request to ensure deadlock do not occur	i) It focuses on structually negating one of the four necessary cond^n for deadlock: Mutual exclusion, hold and wait, no preemption, circular wait
ii) This allows processes to request resource while ensuring system is in safe state.	ii) This allows designing the system in such a way that deadlock can't occur by eliminating one or more of necessary cond^n
iii) Algo can be complex and may require overhead	iii) Algo are generally less complex
iv) It leads to better resource utilisation as resources are dynamically being allocated	iv) It may not optimise the resource utilization since it relies on defined rules.

⑥ RAG and banker's algo.

Resource Allocation Graph	Banker's Algorithm
i) It is deadlock prevention method used in operating system to represent resource allocation	i) It is a deadlock avoidance method to allocate resource available dynamically by checking its
ii) RAG works by representing processes and resources as nodes in a graph with edges denoting resource	ii) Banker's algo operates by maint the max demand of each proces and the available resources in the sy
iii) The system overload is reduced but the time delay may increase due to checking safety cond^n for every proces	iii) System overload may occur as dynamic allocation is carried out but it may increase the system sta
iv) RAG may not optimize the utilization of resource as process may need to wait for safety cond^n check	iv) Since the resources are alloc dynamically, bankers algo optimizes the resources utilizatio

(Q.2)

Ans $Need[i,j] = Max[i,j] - Allocation[i,j]$

Processes	Need			
	A	B	C	D
P_0	2	1	0	3
P_1	1	0	0	1
P_2	0	2	0	0
P_3	4	1	0	2
P_4	2	3	2	0

Need < Available

$i=0$: Need = 2,1,0,3
Available = 0,3,0,1
Condⁿ is false
∴ Finish[0] = F

$i=1$ Need 1 ≤ Work
(1,0,0,1) ≤ (0,3,0,1)
Condⁿ is false
Finish[1] = F

$i=3$ Need 2 ≤ Work
(0,2,0,0) ≤ (0,3,0,1)
Condⁿ True
Finish[3] = T

i = 4 Need 3 ≤ Work
(4,1,0,2) ≤ (3,4,2,2)
Cond^n false

i = 5 Need 0 ≤ Work
(4,1,0,2) (3,4,2,2)
Cond^n is false

Now, Work = Allocated (0) + Work = (3,0,1,4) + (7,6,3,4)
Work = (10,6,4,8)
Release process

For process P_1:

Need 1 ≤ Work ⇒ cond^n true
Finish [1] = T
Work = (12,8,5,8)

Now, For process P_3:
Is Need 3 ≤ Work
(4,1,0,2) (12,8,5,8)

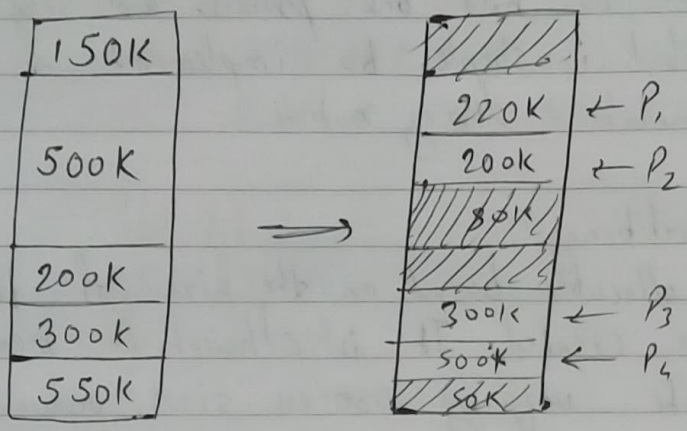
$P_2 \rightarrow P_4 \rightarrow P_0 \rightarrow P_1 \rightarrow P_3$

Q.3)
Ans. a) Partitioning: Virtual memory management brings process into main memory for executing by the processor involves virtual memory based on segmentation and paging.

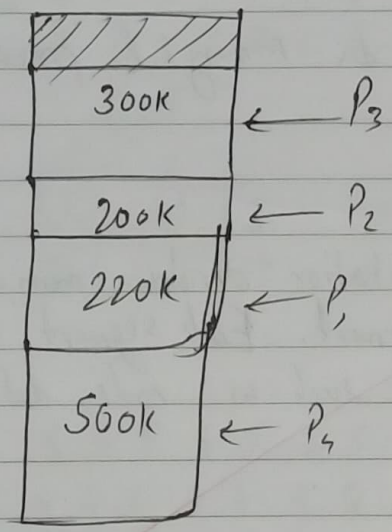
Types of memory partitioning:

Q.4)

Ans. Next Fit:



Best fit:



I) Fixed partitioning:

In this, memory is divided into fixed size partitions. Each partition can hold one process, and size of partition can. This method is simple to implement & there is no fragmentation during runtime

II) dynamic partitioning:

It allocates based on the size of the process partitions are created. It is efficient in memory usage can accomodate varying process sizes. reduces usage of memory.

III) paging:

It divides memory & processes into fixed sizes block called frames.

IV) Segmentation:

Segmentation divides memory & processes. It has variable size segment. Each segment represents a logical unit of a program. such as codes, data or stack.

Q.5) FIFO

Ans.

F_1	1	1	1	4	4	4	6	6	6	6	9	9	9	9	9	9	9	9	9	2
F_2		2	2	2	5	5	5	7	7	7	7	7	7	7	7	5	5	5	5	5
F_3			3	3	3	3	1	1	1	8	8	8	8	8	8	8	4	4	4	4
				H	H					H	H			H	H	H			H	H

Total hit: 9
Total fault: 13
Page size: 22

LRU:

F_1	1	1	1	4	4	4	4	4	4	7	7	7	7	7	7	7	7	5	5	5	5
F_2		2	2	2	5	5	5	1	1	1	8	8	8	8	8	8	8	4	4	4	4
F_3			3	3	3	3	3	6	7	6	6	6	6	9	9	9	9	9	9	9	2
					H	H					H	H			H	H	H			H	H

Total hit = 9
Total fault = 13

Optimal:

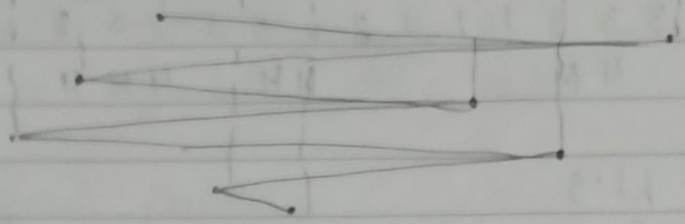
F_1	1	1	1	1	5	5	5	5	5	5	5	5	5	9	9	9	9	5	5	5	5	5
F_2		2	2	4	4	4	4	4	4	4	8	8	8	8	8	8	8	8	4	4	4	4
F_3			3	3	3	3	3	1	6	7	7	7	7	7	7	7	7	7	7	7	7	2
					H	H					H	H		H		H	H			H	H	

Total hit = 9
Total fault = 13

(Q.6)

Ans. FCFS

0 14 37 53 65 67 98 122 124 183 199

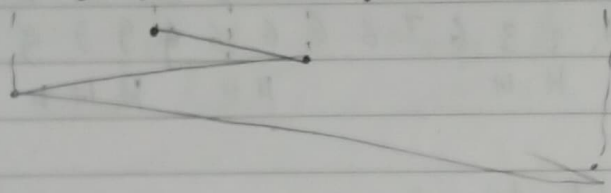


Total seek time = 45 + 85 + 146 + 85 + 108 + 110 + 59 + 2

= 640

SSTF:

0 14 37 53 65 67 98 122 124 183 199

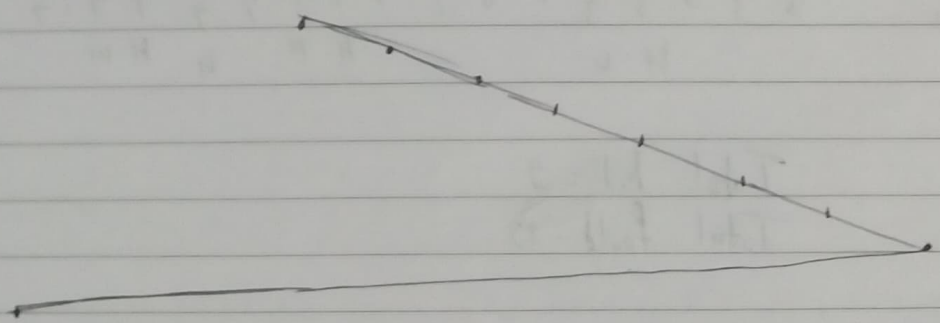


Total seek time = 12 + 2 + 30 + 23 + 84 + 24 + 2 + 59

= 220

SCAN:

0 14 37 53 65 67 98 122 124 183 199



Total seek time = 12 + 2 + 31 + 24 + 2 + 59 + 16 + 162 + 23

= 331