Abrar Javed Sorafi
ID: 221-15-5053

Answer to the Question NO. 1

a)(i)

Process	Allocation				Maximum				Need				Available			
	A	B	C	D	A	B	C	D	A	B	C	D	A	B	C	D
P_0	0	2	3	0	1	4	5	0	1	2	2	0	7	5	8	10
P_1	3	1	0	1	5	2	0	2	2	1	0	1	7	7	11	10
P_2	1	0	1	0	1	2	1	2	0	2	0	2	10	8	11	11
P_3	0	0	0	1	0	1	0	1	0	1	0	0	11	8	12	11
P_4	2	1	1	0	3	2	3	1	1	1	2	1	11	8	12	12
P_5	1	1	0	0	2	2	1	1	1	1	1	1	13	9	13	12
	7	5	5	2									14	10	13	12

Given, A=14, B=10, C=13, D=12

P_0 ⇒ 7 5 8 10
+ 0 2 3 0
7 7 11 10

P_1 ⇒ 7 7 11 10
+ 3 1 0 1
10 8 11 11

P_2 ⇒ 10 8 11 11
+ 1 0 1 0
11 8 12 11

P_3 ⇒ 11 8 12 11
+ 0 0 0 1
11 8 12 12

P_4 ⇒ 11 8 12 12
+ 2 1 1 0
13 9 13 12

P_5 ⇒ 13 9 13 12
1 1 0 0
14 10 13 12

So the safe sequence,

$P_0 \rightarrow P_1 \rightarrow P_2 \rightarrow P_3 \rightarrow P_4 \rightarrow P_5$

(ii) Given two requests:

P0(13,5,3,1)
P1(0,1,0,0)

For P0 request:

Check if request ≤ Need(P0):

(13,5,3,1) ≰ (1,2,2,0)

Not safe, request exceeds Need

For P1 request:

check if request ≤ Need (P1):

(0,1,0,0) ≤ (2,1,0,1) i.e safe

check if request ≤ Available (P1):

(0,1,0,0) ≤ (7,5,8,10) i.e. safe

Grant the request:

1. Update Available:

(7,5,8,10) - (0,1,0,0) = (7,4,8,10)

2. Update Allocation:

(3,1,0,1) + (0,1,0,0) = (3,2,0,1)

3. Update Need:

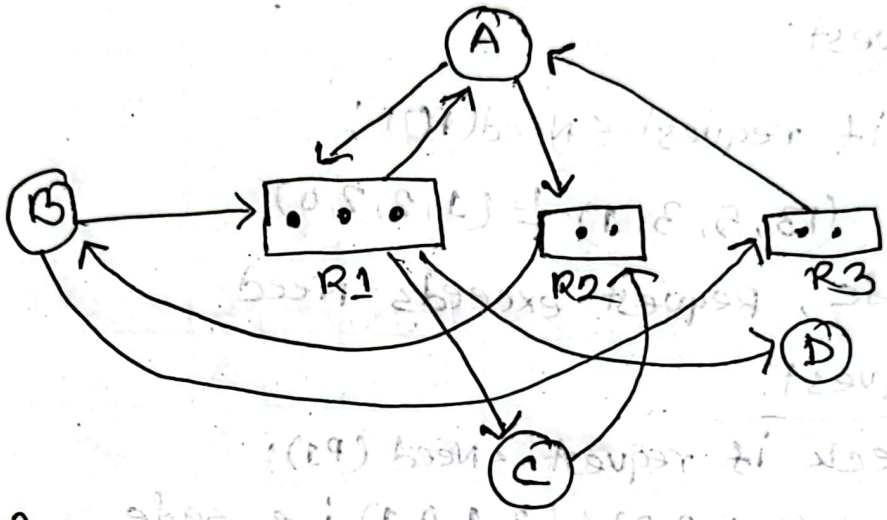
(2,1,0,1) - (0,1,0,0) = (2,0,0,1)

Therefore,

Request 1 from P0 can't be granted but request 2 from P1 can be granted.

~~Answer~~:

b) Available Resources and based on requests we construct the following graph:



c) The given resource allocation graph represents a safe state since there are no loops in the graph and each resource has enough resources to satisfy the requests of the process. The possible finishing sequence is:

P2 → P3 → P1 → P4 → P5 → P0

Explanation:

P2: P2 can finish first because it has no request.

P3: P3 can finish as it only needs R4 which is available.

P1: P1 can finish next as it only needs R4 which is availabl.

P4: P4 can finish as it only needs R3 which is available.

P5: P5 can finish after completing of P4

P0: Finall P0 can be finished.

Answer to the Question No: 2

Dynamic Partitioning scheme: Memory is allocated dynamically based on process requests creating partitions of required sizes as needed.

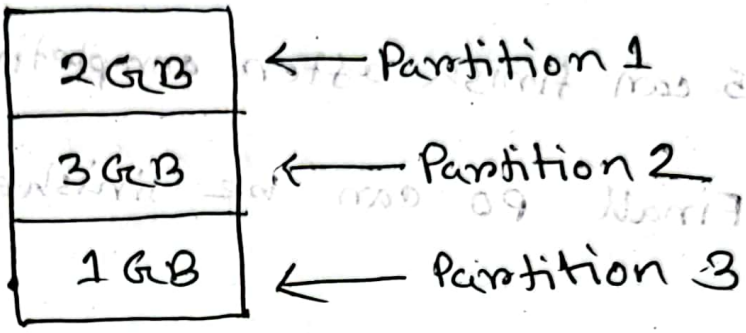
Advantages:

(i) Efficient Memory Utilization
(ii) Flexibility of process for varying sizes.

Disadvantages:

(i) External Fragmentation.
(ii) Overhead in managing memory allocation

According to the scenario the dynamic partitioning scheme:



b) Reasons behind laggy performance are:

(i) Insufficient Memory:

a. Swapping: Excessive disk I/O slows the system.

b. Thrashing: System spends too much time in swapping.

(ii) Memory Leaks: Processes fail to release memory.

(iii) Memory Fragmentation: Non-contiguous blocks prevent larger allocations.

(iv) Insufficient Memory Allocation: Poor algorithms result in suboptimal utilization.

The solutions to improve the performance are:

(i) Using effective memory allocation.

(ii) Perform memory compaction.

(iii) Detect and fix Memory leaks

(iv) Implement efficient process management

These steps could enhance the performance.

Answer to the Question No: 3

a) LRU page Replacement:

	1	2	3	4	1	2	5	1	2	3	4	5	6	2	1	3	7	7	0	9
f1	1	1	1	4	4	4	5	5	5	5	5	5	6	6	6	3	3	3	3	9
f2		2	2	2	1	1	1	1	1	3	3	3	3	2	2	2	7	7	7	7
f3			3	3	3	2	2	2	2	2	4	4	4	4	1	1	1	1	0	0
	F	F	F	F	F	F	F	H	H	F	F	H	F	F	F	F	H	F	F	F

Hit Ratio = $\frac{4}{20} \times 100\%$.

$= 20\%$.

Fault Ratio = $\frac{16}{20} \times 100\%$.

$= 80\%$.

Optimal Page Replacement:

	1	2	3	4	1	2	5	1	2	3	4	5	6	2	1	3	7	7	0	9
f1	1	1	1	1	1	1	1	1	1	3	4	4	6	6	1	1	1	1	0	0
f2		2	2	2	2	2	2	2	2	2	2	2	2	2	2	3	3	3	3	9
f3			3	4	4	4	5	5	5	5	5	5	5	5	5	5	7	7	7	7
	F	F	F	F	H	H	F	H	H	F	F	H	F	H	F	F	F	H	F	F

Hit Ratio = $\frac{7}{20} \times 100\%$ = 35%.

Fault Ratio = $\frac{13}{20} \times 100\%$ = 65%.

So Optimal Page Replacement is more efficient.

b) To improve LRU Efficiency:

(i) Second chance Algorithm: Gives pages a second chance.

(ii) Clock Algorithm: Uses a circular queue.

(iii) Adaptive Algorithm: Adjust replacement policies.

Strategies to improve Optimal Algorithm:

Although impractical, its principles can inspire algorithms that predict future page usage.

Practical Constraints and Trade-offs:

a) Hardware Support

b) Complexity vs Performance

c) Prediction challenges

So in a nutshell we can say Enhancing LRU with advanced methods and leveraging optimals principles can improve page replacement efficiency.