Assignment
CSE323

Name: Munna Biswas
SID : 221-15-5261
Section: 61-J

Answer to the question no-1

a) (i)

	Allocation				Max need				Connected				Available			
Process	A	B	C	D	A	B	C	D	A	B	C	D	A	B	C	D
P_1	0	2	3	0	1	4	5	0	1	2	2	0	7	5	8	10
P_2	3	1	0	1	5	2	0	2	2	1	0	1	7	7	11	10
P_3	1	0	1	0	1	2	1	2	0	2	0	2	10	8	12	11
P_4	0	0	0	1	0	1	0	1	0	1	0	0	11	8	12	12
P_5	2	1	1	0	3	2	3	1	1	1	2	1	11	8	12	12
P_6	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 that,

$A = 14$ | $C = 13$

$B = 10$ | $D = 12$

$\therefore$ safe sequence $\Rightarrow P_1 \rightarrow P_2 \rightarrow P_3 - P_4 \rightarrow P_5 \rightarrow P_6$

(ii) Given that,

$P_0(13, 5, 3, 1)$, $P_1(0, 1, 0, 0)$

Now, if P_0 request $\leq$ need (1)

$(13, 5, 3, 1) \nleq (1, 2, 2, 0)$

$\therefore$ Not safe, request exceeds need (P_0)

For P_i : Check if request $\leq$ need (P_1)
$(0,1,0,0) \leq (2,1,0,1)$ safe

Granting true request :
1) available update : (7, 4, 8, 10)
2) updated allocation : (3, 2, 0, 1)
3) Needed update : (2, 0, 0, 1)

So, P_0 can not be granted but P_1 can be granted.

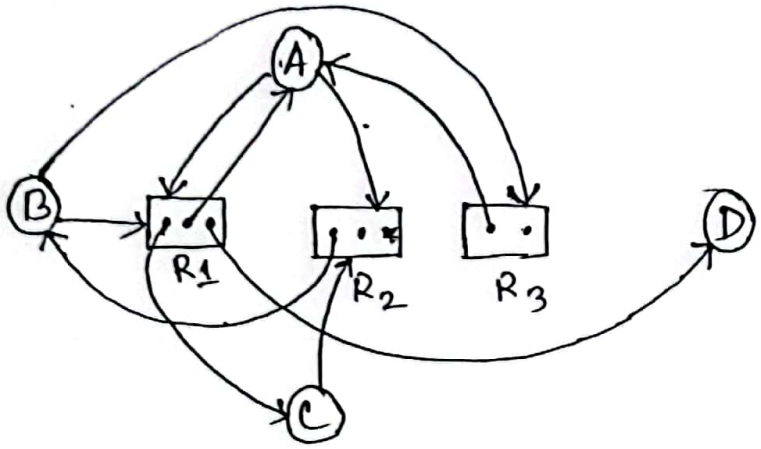
b) Here, Available resources :
R_1 (Laptop) → 3 ; R_2 (Routers) → 2 ; R_3 (Hard Drives) → 2

Now,
Team A : Allocation : 1 Laptop, 1 hard drive
Request : 1 Laptop, 1 Routers

Team B : Allocation : 1 Routers
Request : 1 laptop, 1 hard drive

Team C : Allocation : 1 Laptop
Request : None

: Resource Allocation Graph :



(C) - The resource Allocation graph represents safe state:

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

- As there is no cycle, so each resource has enough instance to satisfy the requests. Here P_2 can finish because it has two requests, where the P_3 need R_4, which can be finish as well. After that P_1 can finish and then P_4 also. P_5 can finish because R_2 is available. Then P_0 also can finish at the last and all the processes are executed.

Answer to the question no-2

a)

-Dynamic partitioning scheme:

-Memory is allocated dynamically which is based on process requests and after that creating partitions of required sizes.

Advantages:

1. Flexible for process of different sizes.
2. Efficient memory utilization.

Disadvantages:

1. Overhead in managing memory allocation.
2. External fragmentation.

b) The resource behind the laggy issue in performance described below:

1. Memory Leaks: The process fail to release memory.
2. Insufficient memory: i) Excessive disk shows the system.
   ii) The system spends to much time in swapping.

3. Memory Fragmentation: Non-contiguous blocks prevent longer allocation.

- Solutions for performance improvement:
  1. Perform memory compaction to reduce fragmentation.
  2. Detect and fix memory leaks.
  3. uses effective memory allocation Algorithm

Answer to the question no-3

Given that,
physical memory : 3. frame

Ref: 1, 2, 3, 4, 1, 2, 5, 1, 2, 3, 4, 5, 6, 2, 1, 3, 7, 7, 0, 9

LRU:

Q.	1	2	3	4	1	2	5	1	2	3	4	5	6	2	1	3	7	7	0	9	
f_1	1	1	1	4	4	4	5	5	5	3	3	3	6	6	6	3	3	3	3	9	
f_2		2	2	2	1	1	1	1	1	1	4	4	4	2	2	2	7	7	7	7	
f_3			3	3	3	2	2	2	2	2	2	5	5	5	1	1	1	1	0	0	
	F	F	F	F	F	F	F	H	H	F	F	F	F	F	F	F	F	H	F	F	

Hit ratio $= \frac{3}{20} \times 100\% = 15\%$

Fault Ratio $= \frac{17}{20} \times 100 = 85\%$

6

Optimal:

Q	1	2	3	4	1	2	5	1	2	3	4	5	6	2	1	3	7	7	0	9
F_1	1	1	1	1	1	1	1	1	1	3	4	4	4	4	4	3	3	3	3	9
F_2		2	2	2	2	2	2	2	2	2	2	2	2	2	1	1	1	1	0	0
F_3			3	4	4	4	5	5	5	5	5	5	6	6	6	6	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 is more effcient.

6)-Stradigies to improve LRU efficiency:

1. Second chance Algorithm.
2. clock Algorithm
3. Adaptive Algorithm.

-Stradigies to improve optimal Algorithm efficiency:

Its motivates to forecast future page uses despite the fact that they are impracticable.

7.

Practical constraints and Trade off:

a. Hardware support

b. Complexity vs performance

c. Prediction challenges

In conclusion page replacement efficiency can be increased by utlizing optimals principles and improving LRU.