Rokaya kheirdeen 1018306

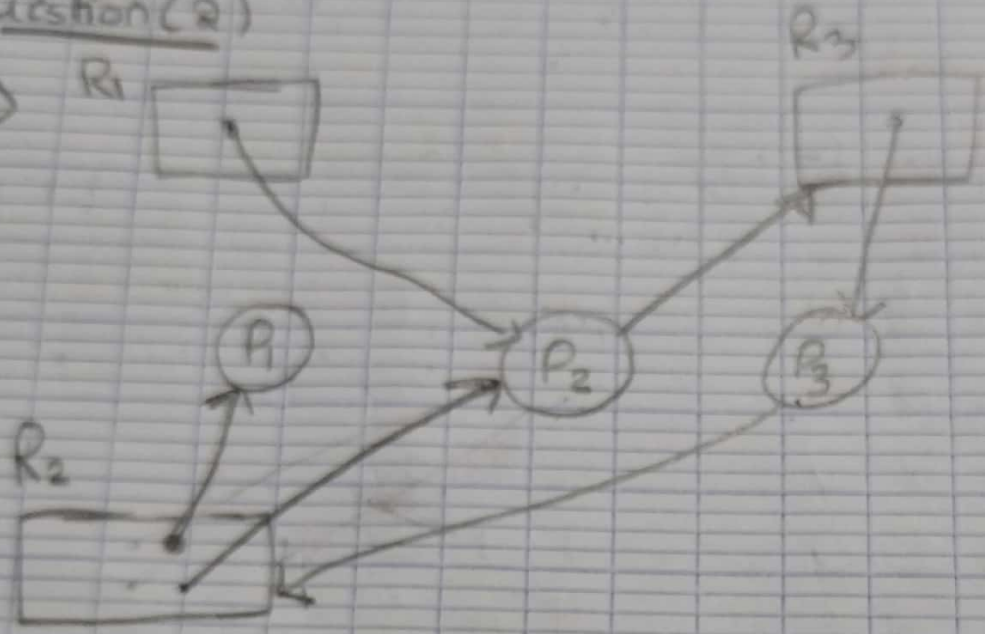
Question 1

$S_1=0,\ S_2=0,\ S_3=0,\ S_4=0$.

P_1	P_2	P_3	P_4
wait (S_1)			
statement(1)	statement 3;	statement 5;	statement 7;
	signal (S_3)	Signal (S_4)	
wait(S_4)		wait(S_3)	wait (S_2)
statement 2;	statement 4;	statement 6	statement 8;
	signal(S_1)	signal(S_2)	

Question (2)

a)



b) $R_2 \rightarrow P_2 \rightarrow R_3 \rightarrow P_3$

c) P_3 can be executed, because R_2 is available

P_2 can be executed, because R_3 is available

Then there is no deadlock

Question (3)

Allocation	A (5 instances)	B (3 instances)	C (6 instances)
	A B C	A B C	Available
P_0	0 1 0	4 3 4	
P_1	1 0 0	2 2 5	
P_2	2 0 0	2 1 3	
P_3	0 1 0	1 1 2	
P_4	0 0 1	4 0 1	

a) Available

2 1 5

b) Need

P_0	4	2	4 ✓
P_1	1	2	5 ✓
P_2	0	1	3 ✓
P_3	1	0	2 ✓
P_4	4	0	0 ✓

c)

	work
	2 1 5
P_2	4 1 5
P_3	4 2 5
P_0	4 3 5
P_1	5 3 5
P_4	5 3 6

⇒ Safe

Order ($P_2 \rightarrow P_3 \rightarrow P_0 \rightarrow P_1 \rightarrow P_4$)

d) (1, 1, 1) by P_2 req. > need.

↳ can't be granted by P_2 1 1 3 > 0 1 3

e) (2, 2, 2) P_0 need 4 2 4

(2, 2, 2) < 4, 2, 4 req. < need ✓

req. & Available

2 2 2 < 2 1 5 ⇒ req. > Avaible

So The request can't be granted.

f) 1, 0, 2 (10 2)

Need of P_3: 1 0 2 ✓

Available: 2 1 5 ✓

New allocation → old request + Allocation

new Available → old Available − request

new need → old need − req.

we need to check P_3.

	Allocation			Need			Available	
	A	B	C	A	B	C		
P_0	0	1	0	P_0 4	2	4 ✓	2 1 5	
P_1	1	0	0	P_1 1	2	5 ✓	1 0 2	
P_2	2	0	0	P_2 0	1	3 ✓	1 1 3	
P_3	1	1	2	P_3 0	0	0 ✓		
P_4	0	0	1	P_4 4	0	0		

Work

	1	1	3
P_2	3	1	3
P_3	4	2	5
P_0	4	3	5
P_1	5	3	5
P_4	5	3	6 ✓