

Exemple v1- Détection des inter-blocages

1. Init: Travail=Disponible; Fin[i]=Allocation[i]==0;
2. Trouver !Fin[i] && Requête[i] ≤ Travail Sinon terminé.
3. Travail+=Allocation[i]; Fin[i]=true; goto 2;

Allocation	A	B	C
P0	0	1	0
P1	2	0	0
P2	3	0	3
P3	2	1	1
P4	0	0	2

Requête	A	B	C
P0	0	0	0
P1	2	0	2
P2	0	0	0
P3	1	0	0
P4	0	0	2

Disponible		
A	B	C
0	0	0

Travail	A	B	C
Init	0	0	0
It1	0	1	0
It2	3	1	3
It3	5	1	3
It4	7	2	4
It5	7	2	6

Fin	P0	P1	P2	P3	P4
Init	F	F	F	F	F
It1	V	F	F	F	F
It2	V	F	V	F	F
It3	V	V	V	F	F
It4	V	V	V	V	F
It5	V	V	V	V	V

Exemple v2- Détection des inter-blocages

1. Init: Travail=Disponible; Fin[i]=Allocation[i]==0;
2. Trouver !Fin[i] && Requête[i] ≤ Travail Sinon terminé.
3. Travail+=Allocation[i]; Fin[i]=true; goto 2;

Allocation	A	B	C
P0	0	1	0
P1	2	0	0
P2	3	0	3
P3	2	1	1
P4	0	0	2

Requête	A	B	C
P0	0	0	0
P1	2	0	2
P2	0	0	1
P3	1	0	0
P4	0	0	2

Disponible		
A	B	C
0	0	0

Travail	A	B	C
Init	0	0	0
It1	0	1	0

Fin	P0	P1	P2	P3	P4
Init	F	F	F	F	F
It1	V	F	F	F	F

Exemple v1- Évitement des inter-blocages

1. Init: Travail=Disponible; Fin[all]=false;
2. Trouver !Fin[i] && Besoin[i] ≤ Travail Sinon terminé.
3. Travail+=Allocation[i]; Fin[i]=true; goto 2

	Allocation			Max			Besoin		
	A	B	C	A	B	C	A	B	C
P0	0	1	0	7	5	3	7	4	3
P1	2	0	0	3	2	2	1	2	2
P2	3	0	2	9	0	2	6	0	0
P3	2	1	1	2	2	2	0	1	1
P4	0	0	2	4	3	3	4	3	1

Disponible		
A	B	C
3	3	2

Travail	A	B	C
Init	3	3	2
IT1-P1	5	3	2
IT2-P3	7	4	3
IT3-P0	7	5	3
IT4-P2	10	5	5
IT5-P4	10	5	7

Fin	P0	P1	P2	P3	P4
Init	F	F	F	F	F
IT1-P1	F	V	F	F	F
IT2-P3	F	V	F	V	F
IT3-P0	V	V	F	V	F
IT4-P2	V	V	V	V	F
IT5-P4	V	V	V	V	V

Exemple v2- Évitement des inter-blocages

1. Init: Travail=Disponible; Fin[all]=false;
2. Trouver !Fin[i] && Besoin[i] ≤ Travail Sinon terminé.
3. Travail+=Allocation[i]; Fin[i]=true; goto 2

Est-ce qu'une requête (1,0,2) par le processus P1 peut être accordée?

	Allocation			Max			Besoin		
	A	B	C	A	B	C	A	B	C
P0	0	1	0	7	5	3	7	4	3
P1	3	0	2	3	2	2	0	2	0
P2	3	0	2	9	0	2	6	0	0
P3	2	1	1	2	2	2	0	1	1
P4	0	0	2	4	3	3	4	3	1

Disponible		
A	B	C
2	3	0

Travail	A	B	C
Init	2	3	0
IT1-P1	5	3	2
IT2-P3	7	4	3
IT3-P0	7	5	3
IT4-P2	10	5	5
IT5-P4	10	5	7

Fin	P0	P1	P2	P3	P4
Init	F	F	F	F	F
IT1-P1	F	V	F	F	F
IT2-P3	F	V	F	V	F
IT3-P0	V	V	F	V	F
IT4-P2	V	V	V	V	F
IT5-P4	V	V	V	V	V

Réponse: Oui

Exemple v3- Évitement des inter-blocages

1. Init: Travail=Disponible; Fin[all]=false;
2. Trouver !Fin[i] && Besoin[i] ≤ Travail Sinon terminé.
3. Travail+=Allocation[i]; Fin[i]=true; goto 2

Est-ce qu'une requête (3,3,0) par le processus P4 peut être accordée?

	Allocation			Max			Besoin		
	A	B	C	A	B	C	A	B	C
P0	0	1	0	7	5	3	7	4	3
P1	2	0	0	3	2	2	1	2	2
P2	3	0	2	9	0	2	6	0	0
P3	2	1	1	2	2	2	0	1	1
P4	3	3	2	4	3	3	1	0	1

Disponible		
A	B	C
0	0	2

Travail	A	B	C
Init	0	0	2

Fin	P0	P1	P2	P3	P4
Init	F	F	F	F	F

Réponse: Non

Exemple v4- Évitement des inter-blocages

1. Init: Travail=Disponible; Fin[all]=false;
2. Trouver !Fin[i] && Besoin[i] ≤ Travail Sinon terminé.
3. Travail+=Allocation[i]; Fin[i]=true; goto 2

Est-ce qu'une requête (0,2,0) par le processus P0 peut être accordée?

	Allocation			Max			Besoin		
	A	B	C	A	B	C	A	B	C
P0	0	3	0	7	5	3	7	2	3
P1	2	0	0	3	2	2	1	2	2
P2	3	0	2	9	0	2	6	0	0
P3	2	1	1	2	2	2	0	1	1
P4	0	0	2	4	3	3	4	3	1

Disponible		
A	B	C
3	1	2

Travail	A	B	C
Init	3	1	2
IT1-P3	5	2	3
IT2-P1	7	2	3
IT3-P0	7	5	3
IT4-P2	10	5	5
IT5-P4	10	5	7

Fin	P0	P1	P2	P3	P4
Init	F	F	F	F	F
IT1-P3	F	F	F	V	F
IT2-P1	F	V	F	V	F
IT3-P0	V	V	F	V	F
IT4-P2	V	V	V	V	F
IT5-P4	V	V	V	V	V

Réponse: Oui