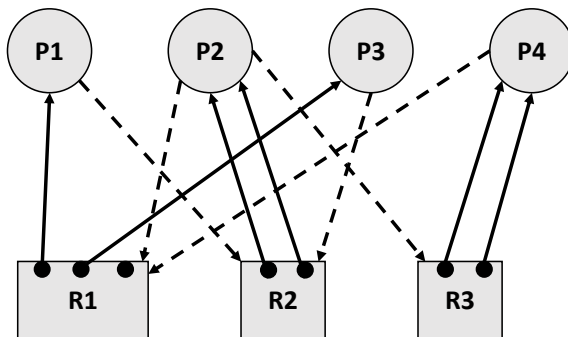


## Semaine 9 – Questions sur l'interblocage (D-E-P) – Solutions

- Un système dispose de 3 R1, 2 R2 et 2 R3. Supposez que 4 processus P1, P2, P3 et P4 partagent ces ressources et que l'état actuel d'attribution des ressources est :
  - P1 détient une ressource de type R1 et demande une ressource de type R2;
  - P2 détient 2 ressources de type R2 et demande une ressource de type R1 et une ressource de type R3;
  - P3 détient 1 ressource de type R1 et demande une ressource de type R2;
  - P4 détient 2 ressources de type R3 et demande une ressource de type R1.

Construisez le **graphe d'allocation des ressources**. Y a-t-il un interblocage ? Si oui, quels sont les processus concernés ?

Réponse :



Non, car il existe une séquence saine : P4, P2, P1, P3.

- Un système gère 4 processus (P1, P2, P3 et P4) et 3 types de ressources (R1, R2 et R3). L'état courant du système est :

Processus	Alloc			Req		
	R1	R2	R3	R1	R2	R3
P1	1	0	2	0	1	0
P2	2	0	1	0	y	4
P3	1	1	0	1	0	3
P4	1	1	1	0	0	1

A		
R1	R2	R3
1	1	x

Où Alloc indique les ressources allouées à chaque processus, Req donne pour chaque processus, les ressources nécessaires mais non encore obtenues et A indique les ressources disponibles.

a) Supposez que l'algorithme du banquier est appliqué pour éviter les interblocages. Pour quelles plus petite valeur de x et plus grande valeur de y l'état courant est sûr ?

1.  $Req(P1) \leq A$

Oui, on marque P1

$A1 = [2 \ 1 \ 2+x]$

2.  $Req(P4) \leq A1$

Oui, on marque P4

$A2 = [3 \ 2 \ 3+x]$

3.  $Req(P3) \leq A2$

Oui, on marque P3

$A3 = [4 \ 3 \ 3+x]$

4.  $Req(P2) \leq A3$

Oui, on marque P2 Si  $y = 3$  et  $x = 1$

b) Supposez toujours que l'algorithme du banquier est appliqué pour éviter les interblocages,  $x=2$ ,  $y=1$  et que le processus P3 demande 2 ressources de type R3. Est-ce que le système va accepter cette demande ? Justifiez votre réponse.

Réponse : Les nouvelles matrices :

Alloc		
R1	R2	R3
1	0	2
2	0	1
1	1	2
1	1	1

Req		
R1	R2	R3
0	1	0
0	1	4
1	0	1
0	0	1

A		
R1	R2	R3

1	1	0
---	---	---

1. Req(P1) <= A  
Oui, on marque P1  
A1 = [2 1 2]
2. Req(P3) <= A1  
Oui, on marque P3  
A2 = [3 2 4]
3. Req(P4) <= A2  
Oui, on marque P4  
A3 = [4 3 5]
4. Req(P2) <= A3  
Oui, marque P2, la demande est acceptée car l'état atteint est sûr.

3. Expliquez les quatre conditions de Coffman.

Réponse :

Objectif : s'assurer qu'au moins, l'une des quatre conditions nécessaires à l'interblocage, **n'est jamais satisfaite**.

Condition	Comment l'empêcher
Exclusion mutuelle	Impossible de l'empêcher car les ressources sont en général à usage exclusif (imprimantes, mémoires, processeurs, etc.).
Détention et attente	Chaque processus demande à la fois toutes les ressources dont il a besoin. Cependant, <ul style="list-style-type: none"> <li>- il est difficile de connaître à l'avance les besoins des processus,</li> <li>- il y a un risque de famine.</li> </ul>
Pas de réquisition	L'état de la ressource est sauvegardé avant de la préempter puis restaurer ultérieurement. Cette solution est envisageable pour certaines ressources (processeurs, mémoires, etc.) mais pas pour d'autres (imprimantes).
Attente circulaire	Une solution consiste à : <ul style="list-style-type: none"> <li>- ordonner les ressources,</li> <li>- imposer, à chaque processus, la règle suivante : Un processus peut demander une ressource RJ seulement si toutes les</li> </ul>

	ressources qu'il détient sont inférieures à $R_j$ selon l'ordre établi.
--	---

4. Dans le dîner des philosophes, comment peut-on faire pour que la condition **Détention et attente** de Coffman ne soit jamais satisfaite et ainsi prévenir l'interblocage? Expliquez votre réponse.

Réponse :

Chaque philosophe qui a faim entre dans sa section critique, si ses deux fourchettes ne sont pas libres, il quitte alors sa section critique.