

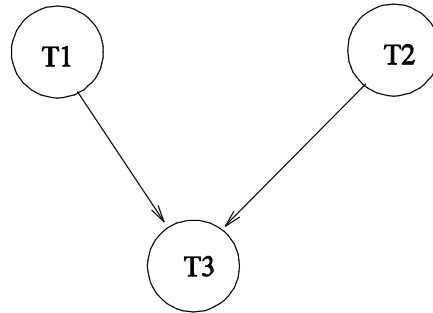
## Corrigé Interrogation n°2

### Exercice1 (/5):

A/ Complétez le code pour réaliser le graphe de précédence suivant en utilisant les sémaphores.

Shared semaphore S=2 ; (0.5)

ProcA() { (0.5)  T1 ; V(S) ;  }	ProcB () { (0.5)  T2 ; V(S) ;  }	ProcC () { (0.5) P(S) ; P(S) ; T3 ;  }
--	---	---



B/ Pour la suite des processus suivants donner l'ordre d'exécution selon les quatre variantes (Les autres processus arrivent simultanément dans cet ordre lorsque Lect1 est en train de lire) : Lect1 Lect2 Redact1 Lect3 Redact2 Lect4

Sol :

FIFO : L1||L2 ;R1 ;L3 ;R2 ;L4 ; (0.5)

Prior-Lect : L1||L2||L3||L4 ;R1 ;R2 ; (0.5)

Prior-Redact ; L1;R1;R2;L2||L3||L4; (0.5)

Prior-relative-Redact:L1;R1;L2||L3||L4;R2 (0.5)

C/ Quel est l'inconvénient majeure de la synchronisation par wait() et signal()?(1)

Si le processus effectuant un signal s'exécute avant le processus bloqué sur wait, alors ce dernier sera bloqué indéfiniment !

### Exercice2 (/5) :

I) Soit un système d'allocation de ressources décrit par le tableau suivant :

Processus	Alloc			MAX			Besoins			Disp			
	A	B	C	A	B	C	A	B	C	A	B	C	
P1	1	0	1	2	0	2	1	0	1	1	2	0	(0.25)
P2	1	2	0	2	4	2	1	2	2	1	2	0	(0.25)
P3	0	1	1	1	4	1	1	3	0	1	3	0	(0.52)
P4	1	0	1	2	1	1	1	1	0	1	1	0	(0.52)

1) Calculer la matrice Bes (besoins courants) et le vecteur Disp (disponibilité) si le nombre d'instance initialement prévu par ressource est: A=4, B=5, C=3.

2) Le système est-il sain ? Donner une séquence d'exécution saine et est-elle unique?

<P4, P1, P2, P3> (1)elle est unique (0.25)! Seul P4 peut commencer la séquence Bes[4,\*]≤Disp (110<120), si P4 se termine alors Seul P1 peut continuer la séquence (101<221), si P1 se termine alors Seul P2 peut continuer la séquence (122<322) et enfin P3 ! (0.5)

Processus	Alloc			MAX			Besoins			Disp			
	A	B	C	A	B	C	A	B	C	A	B	C	
P1	1	0	1	2	0	2	1	0	1	1	2	0	(0.25)
P2	1	2	0	2	4	2	1	2	2	1	2	0	(0.25)
P3	0	1	1	1	4	1	1	3	0	1	3	0	(0.52)
P4	1	0	1	2	1	1	1	1	0	1	1	0	(0.52)

P1	4	0	4	2	0	2	<del>1</del> 0 <del>1</del>	<del>1</del> <del>2</del> 0
P2	4	2	0	2	4	2	<del>1</del> <del>2</del> <del>2</del>	2 2 1
P3	0	1	1	1	4	1	1 3 0	3 2 2
P4	4	0	4	2	1	1	<del>1</del> <del>1</del> 0	4 4 2
								4 5 3

3) La requête de P4 pour une instance de B serait-elle acceptée selon l'algorithme du Banquier ? Justifier.

Elle est valide et satisfaisable.  $Req[P4,B] \leq Bes[P4,B]$  et  $Req[P4,B] \leq Disp[B]$  **(0.25)**

Comme P4 commence l'unique séquence saine alors toutes ses requêtes **valides et satisfiables** seront acceptées, car l'état virtuel obtenu serait sain ! **(0.75)**

4) Si P1 demandait instance de A ?

Elle est valide et satisfaisable.  $Req[P1,A] \leq Bes[P1,A]$  et  $Req[P1,A] \leq Disp[A]$  **(0.25)**

Si virtuellement on acceptait la requête de P1 pour A, P4 ne pourrait plus commencer la séquence (qui est unique !) et donc l'état virtuel ne serait pas sain et la requête de P1 serait refusée ou retardée ! **(0.75)**