

Exercice N° : 01 (10 pts : 3+3+2+2)

Soit 4 sites interconnectés entre eux via des canaux et qui exécutent les séquences de pseudo-code suivantes :

Site P1		Site P2		Site P3		Site P4	
Evènement	Action	Evènement	Action	Evènement	Action	Evènement	Action
e ₁₁	z:= receive(P2)	e ₂₁	x := 2	e ₃₁	z:=receive(P2)	e ₄₁	z:= receive(P2)
e ₁₂	z:= z + 1	e ₂₂	send(x, P1)	e ₃₂	z:= z+1	e ₄₂	z:= z+1
e ₁₃	y:= receive(P3)	e ₂₃	send(x, P3)	e ₃₃	send(z, P1)	e ₄₃	y:= receive(P3)
e ₁₄	z:= y+1	e ₂₄	send(x, P4)	e ₃₄	send(z, P4)	e ₄₄	z:= z+y
e ₁₅	send(z, P2)	e ₂₅	z:=receive(P1)			e ₄₅	send(z, P2)
		e ₂₆	y:= z+1				
		e ₂₇	z:= receive(P4)				

L'opération send(nb, Px) envoie la valeur de l'entier nb au processus Px.

L'opération nb = receive(Px) attend un message contenant un entier de la part du processus Px. L'entier reçu est placé dans nb.

1. Donner le chronogramme correspondant à l'exécution parallèle des 4 sites.
2. Dater chacun des événements en utilisant l'horloge de Lamport (Sachant que les horloges locales sont initialisées à 0).
3. Donner l'ordre total des évènements.
4. Donner la valeur finale de la variable **z** pour chacun des sites.

Exercice N° : 02 (10 pts :2+2+2+2+2)

Soit l'algorithme de Naimi et Trehel suivant :

Lors d'un appel à acquérir :
 danslafile_i := vrai ;
 si père_i ≠ nil alors
 envoyer requête(i) à père_i ;
 père_i := nil ;
 fsi
 attendre(jetonprésent_i) ;

Lors de la réception de requête(j):
 si père_i ≠ nil alors envoyer requête(j) à père_i ;
 sinon si danslafile_i alors suivant_i := j ;
 sinon
 jetonprésent_i := faux ;
 envoyer jeton à j ;
 fsi
 fsi
 père_i := j ;

Lors d'un appel à Libérer :
 danslafile_i := faux ;
 si suivant_i ≠ nil alors
 envoyer jeton à suivant_i ;
 jetonprésent_i := faux ;
 suivant_i := nil ;
 fsi

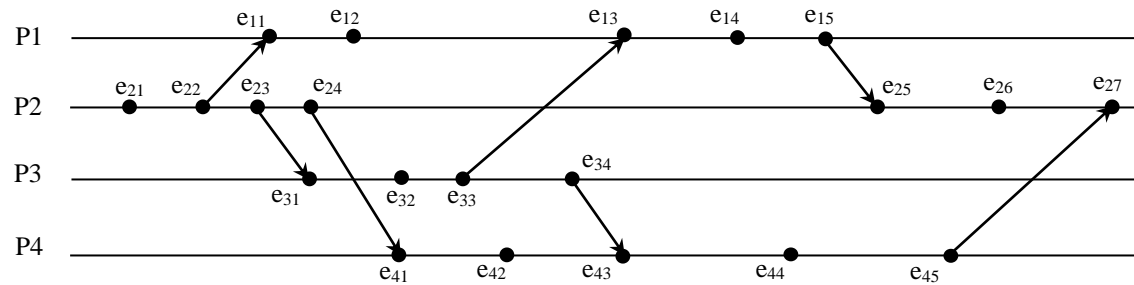
Lors de la réception du jeton:
 jetonprésent_i := vrai ;

1. Comment un site sait-il qu'il est en tête de la file ?
2. Quels sont les sites qui appartiennent à la file et ceux qui appartiennent à l'arborescence ?
3. Quel est l'état de la file s'il existe un site i tel que : danslafile_i = faux et jetonprésent_i = vrai ?
4. Etudier la complexité de l'algorithme dans le pire des cas ensuite dans le meilleur des cas (en expliquant c'est quoi le pire des cas et c'est quoi le meilleur des cas)
5. Dérouler l'algorithme avec 4 sites S1, S2, S3 et S4 dans le cas où S1 est choisi initialement comme racine et que S2 demande l'entrer à la SC, ensuite S3 demande l'utilisation de la SC pendant que S2 utilise la sienne (en donnant l'arborescence à chaque étape).

Les Algorithmes Répartis (Corrigé type)

Exercice N° : 01 (10 pts)

1) 3 pts



2) 3 pts

Evt	e11	e12	e13	e14	e15	e21	e22	e23	e24	e25	e26	e27	e31	e32	e33	e34	e41	e42	e43	e44	e45
Da te	3	4	7	8	9	1	2	3	4	10	11	12	4	5	6	7	5	6	8	9	10

3) 2 pts

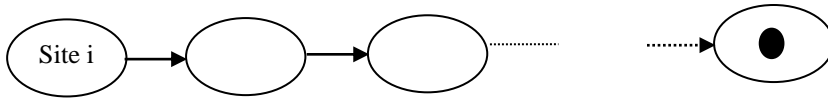
$e_{21} \ll e_{22} \ll e_{11} \ll e_{23} \ll e_{12} \ll e_{24} \ll e_{31} \ll e_{32} \ll e_{41} \ll e_{33} \ll e_{42} \ll e_{13} \ll e_{34} \ll e_{14} \ll e_{43} \ll e_{15} \ll e_{44} \ll e_{25} \ll e_{45} \ll e_{26} \ll e_{27}$

4) 2 pts

Site	P1	P2	P3	P4
Valeur de z	4	6	3	6

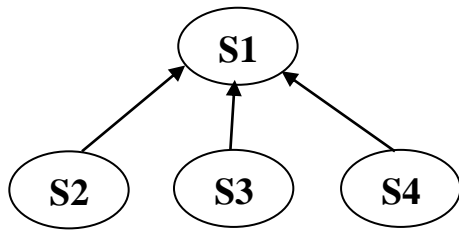
Exercice N° : 02 (10 pts : 2 + 2 + 2 + 2 + 2)

- 1) Le site possède le jeton (jetonprésenti = vrai)
- 2) Les sites qui appartiennent à la file : les sites demandeurs et le site dans la SC
Les sites qui appartiennent à l'arborescence : tous les sites
- 3) La file est vide
- 4) Complexité :
Le pire des cas : $(n - 1)$ message + jeton

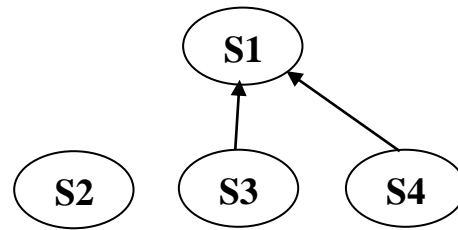


Le meilleur des cas : 0 message
Le site possède déjà le jeton

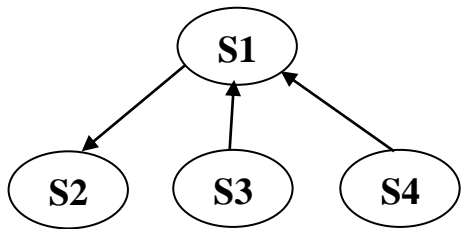
5) Déroulement



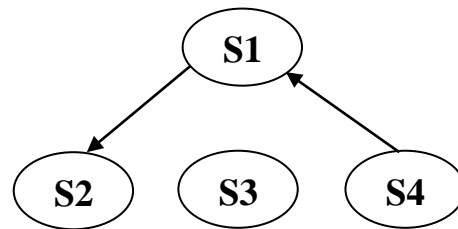
(1)



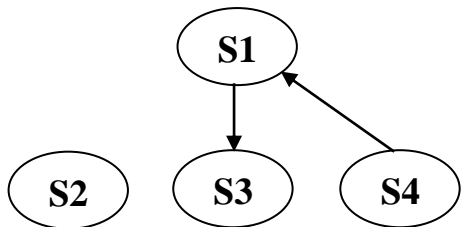
(2)



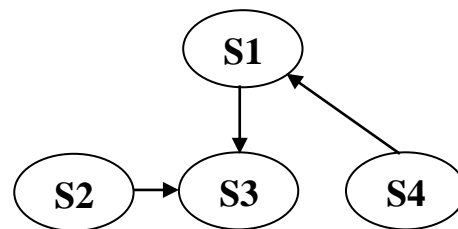
(3)



(4)



(5)



(6)