

TD – Systèmes Distribués

Datations et états globaux

Licence Informatique 3^{ème} Année – UPPA

Spécification du système distribué (exercice 1 et 2)

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

Processus P1	Processus P2	Processus P3	Processus P4
1. $z = \text{receive}(\text{P2})$	1. $x = 10$	1. $z = \text{receive}(\text{P2})$	1. $z = \text{receive}(\text{P2})$
2. $z = z * 2$	2. $\text{send}(x, \text{P1})$	2. $z = z + 6$	2. $z = z + 4$
3. $y = \text{receive}(\text{P3})$	3. $\text{send}(x, \text{P3})$	3. $\text{send}(z, \text{P1})$	3. $y = \text{receive}(\text{P3})$
4. $z = z + y$	4. $\text{send}(x, \text{P4})$	4. $\text{send}(z, \text{P4})$	4. $z = z + y$
5. $\text{send}(z, \text{P2})$	5. $z = \text{receive}(\text{P1})$		5. $\text{send}(z, \text{P2})$
	6. $y = z / 2$		
	7. $z = \text{receive}(\text{P4})$		
	8. $y = z + y$		

L'opération $\text{send}(\text{nb}, \text{Px})$ envoie la valeur de l'entier nb au processus Px .

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

Exercice 1 : Horloges et datations

1. Dessiner le chronogramme correspondant à l'exécution en parallèle des 4 processus.
2. Dater chacun des événements en utilisant la méthode de l'horloge de Lamport.
3. Dater chacun des événements en utilisant la méthode de l'horloge de Mattern.
4. Donner l'ordre total global défini par la datation via la méthode de l'horloge de Lamport.

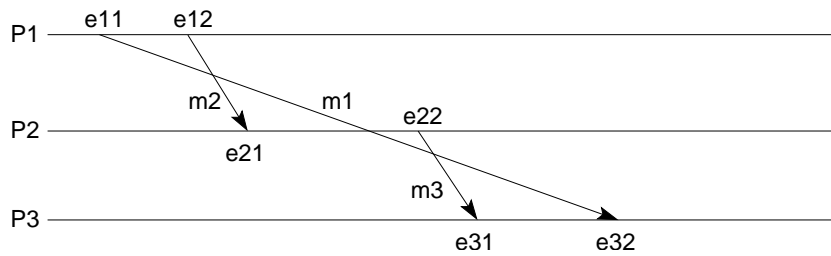
Exercice 2 : Coupures et états

1. Soit la coupure C1 définie par la frontière contenant les événements numérotés 3 de tous les processus et la frontière C2 définie par l'événement 5 de P1, le 4 de P2, le 4 de P3 et le 2 de P1. Ces coupures sont-elles cohérentes ? Justifier le en datant les coupures.
2. Déterminer ("à la main") l'état global associé aux coupures C1 et C2.

Exercice 3 : Protocole d'ordre causal

Un protocole d'ordre causal est un protocole qui assure que les messages reçus sur un *même* site sont délivrés en respectant les dépendances causales entre les événements d'émission de ces messages.

Pour un message m , on notera e_m son événement d'émission, r_m son événement de réception et d_m l'événement de délivrement du message, c'est-à-dire l'événement correspondant au moment où le message sera réellement délivré au processus récepteur (le délivrement peut être décalé dans le futur par rapport à la réception).



1. Mettre en évidence le non-respect des dépendances causales en émission pour le chronogramme ci-dessus (ce chronogramme ne correspond pas à celui du système décrit à la page précédente). Placer les événements de délivrement des messages sur le chronogramme en respectant ces dépendances causales.
2. Déterminer la relation générale entre les événements associés à 2 messages pour que l'ordre causal de leur émission soit respecté lors de leur délivrement.
3. Les horloges de Mattern ou de Lamport permettent-elles de détecter le non-respect des dépendances causales en émission et de bien ordonner les délivrances des messages ?
4. Proposer une nouvelle méthode de datation permettant d'assurer l'ordre causal.