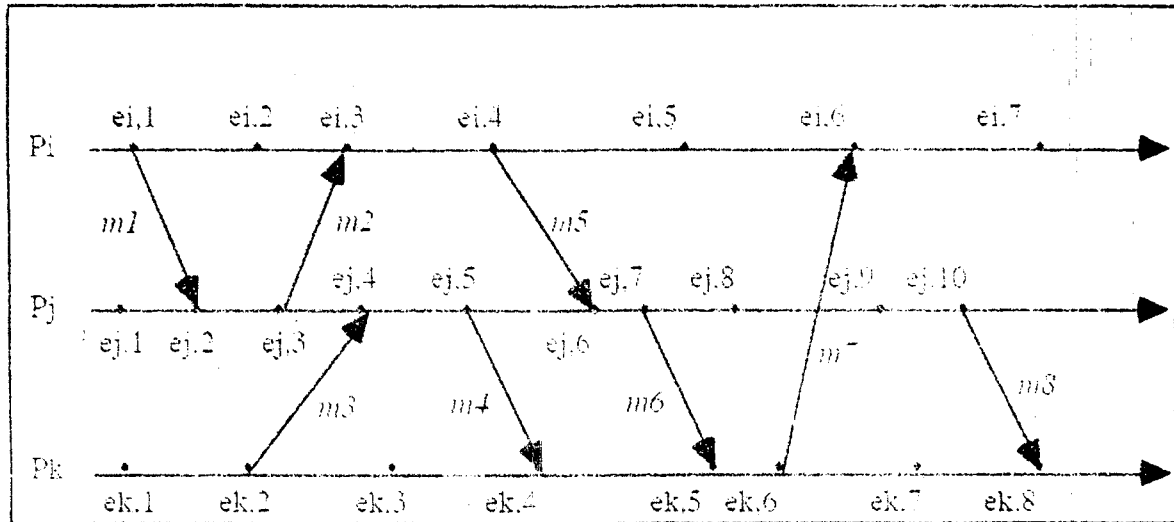
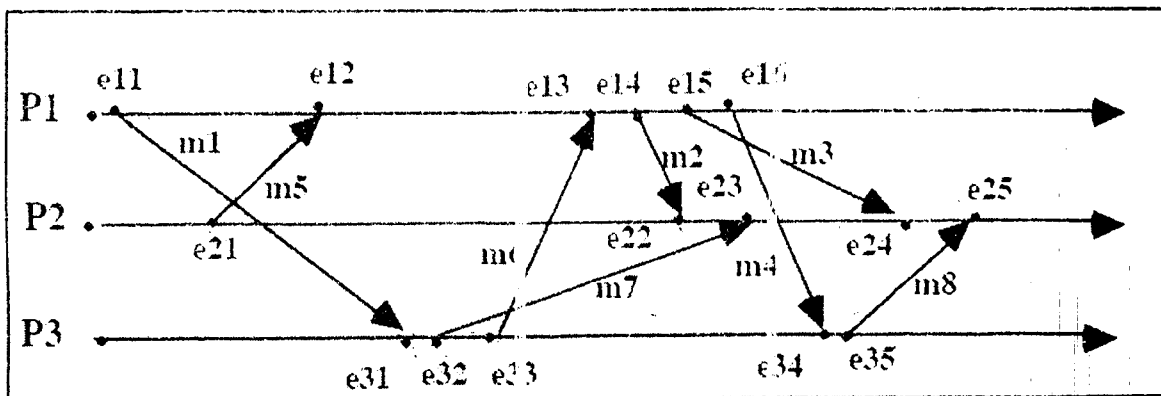


Exercice 1 : On considère le diagramme espace-temps d'un système réparti asynchrone suivant :

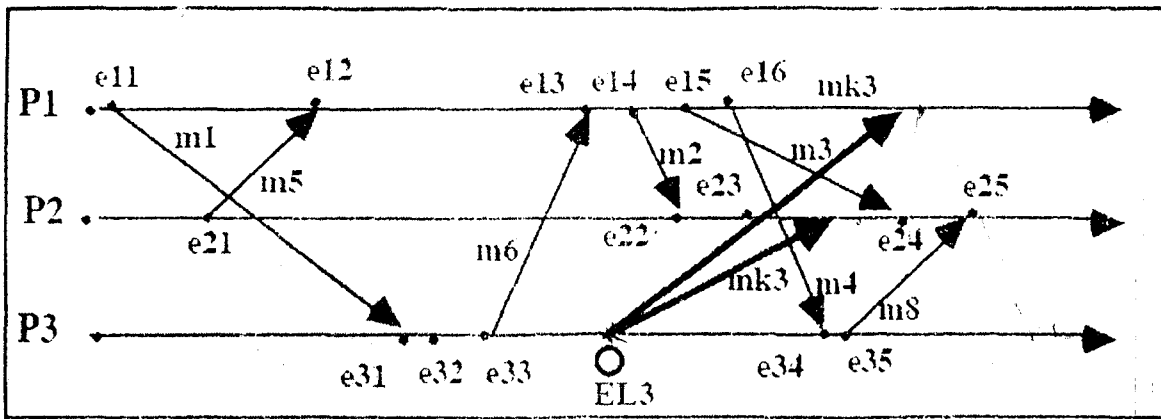


- I. Comparer les événements suivants en mettant la relation qui convient (\rightarrow ou \parallel) :
 $e_{i,2}$ et $e_{k,4}$; $e_{k,1}$ et $e_{j,9}$; $e_{k,2}$ et $e_{i,7}$; $e_{k,3}$ et $e_{i,5}$; $e_{k,3}$ et $e_{j,10}$
- II. Donner l'histoire des événements suivants :
 $e_{i,2}$; $e_{k,1}$; $e_{j,9}$; $e_{i,7}$; $e_{k,3}$; $e_{j,10}$
- III. Est-ce que les relations suivantes sont vérifiées :
 $e_{k,1} \parallel e_{j,9}$; $e_{k,2} \parallel e_{i,7}$; $e_{k,3} \parallel e_{i,5}$

Exercice 2 : Estampiller les événements du diagramme ci-dessous à l'aide des horloges vectorielles. Dites si les relations suivantes sont vérifiées ou non : $e_{23} \parallel e_{32}$; $e_{24} \parallel e_{34}$. Quels sont les messages dont la réception ne respecte pas l'ordre causal ?



Exercice 3 : Soit l'exécution d'un calcul réparti représentée sur la figure suivante :

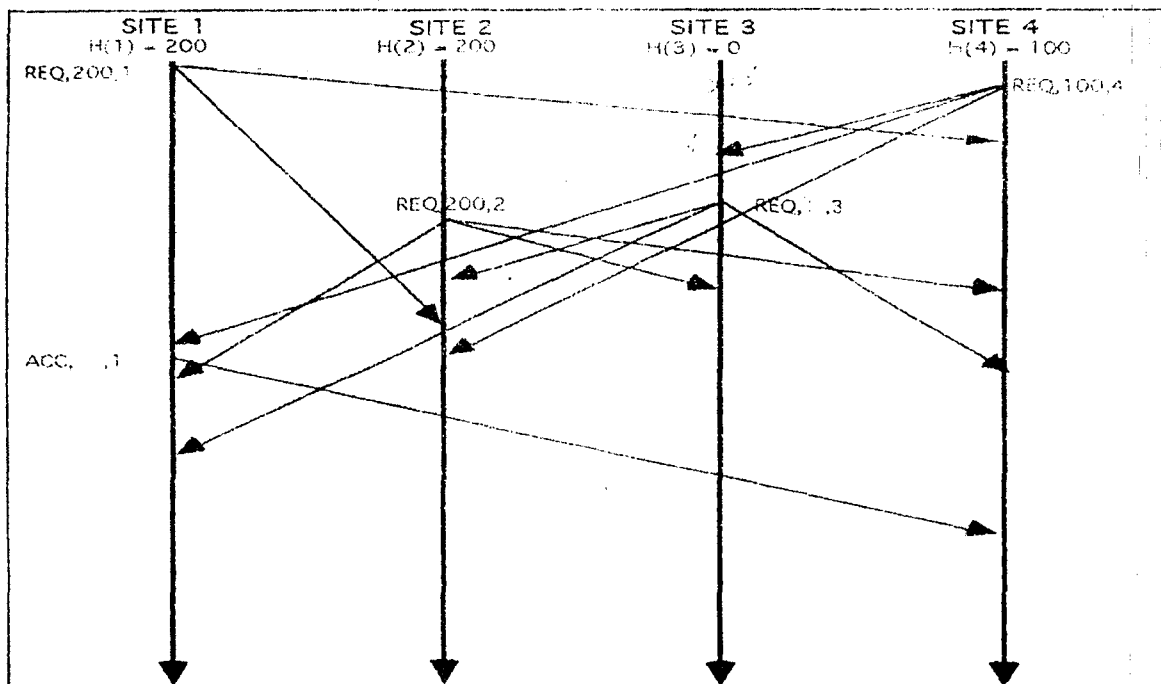


On suppose que P3 lance un calcul d'état global après e33 selon l'algorithme Chandy-Lamport. Sur la figure les mi représentent les messages de données du calcul réparti, mki les marqueurs et Eli l'état local du processus Pi. Compléter le diagramme.

Exercice 4 : On considère un système réparti asynchrone composé de quatre sites site1, site2, site3 et site4. Les quatre sites envoient des demandes d'entrer en section critique comme indiqué sur la figure ci-dessous. On vous demande de gérer cette situation :

- i. Avec l'algorithme de Lamport
- ii. Avec l'algorithme de Ricart et Agrawala

On suppose que les messages complémentaires pour chaque algorithme sont envoyés dès que c'est logiquement possible (il n'y a pas d'attente locale et l'ordre local respecte l'ordre des demandes reçues). On rappelle que les canaux doivent être FIFO pour Lamport et non pour Ricart et Agrawala. On supposera pour ce dernier algorithme seulement que le canal C34 n'est pas FIFO et que l'accusé de réception ACC vers le site 4 est doublé par le message de demande REQ. (3).



Propriétés

- La correction de l'algorithme repose sur les observations suivantes:

- la propriété FIFO de chacun des canaux de communication et la synchronisation des horloges, implique que si un site a reçu un message d'accord (type REPONSE) du site j , toute requête antérieure de ce même site lui est nécessairement arrivée: toute demande lui arrivant de provenance de ce site sera postérieure à la sienne. Ainsi, si un site a reçu l'accord de tous les sites et que sa demande est la plus ancienne, aucune demande antérieure ne lui parviendra d'un autre site: les autres devant nécessairement attendre que la demande en tête de sa file de requêtes (c'est-à-dire le site élu sorte de section critique et envoie un message de type LIBERATION).

- Le nombre de messages échangés pour traiter complètement (entrée et sortie) une phase de section critique requiert $3*(N-1)$ messages ($N-1$ messages de chacun des types).

Exemple

Dans cet exemple impliquant trois sites, les sites S_1 et S_2 veulent entrer en section critique alors que leurs horloges scalaires ont respectivement 3 et 2 comme valeur. Les messages envoyés par S_1 à S_2 et S_3 sont donc estampillés 3.1. Les messages envoyés par S_2 à S_1 et S_3 sont quant à eux estampillés 2.2.

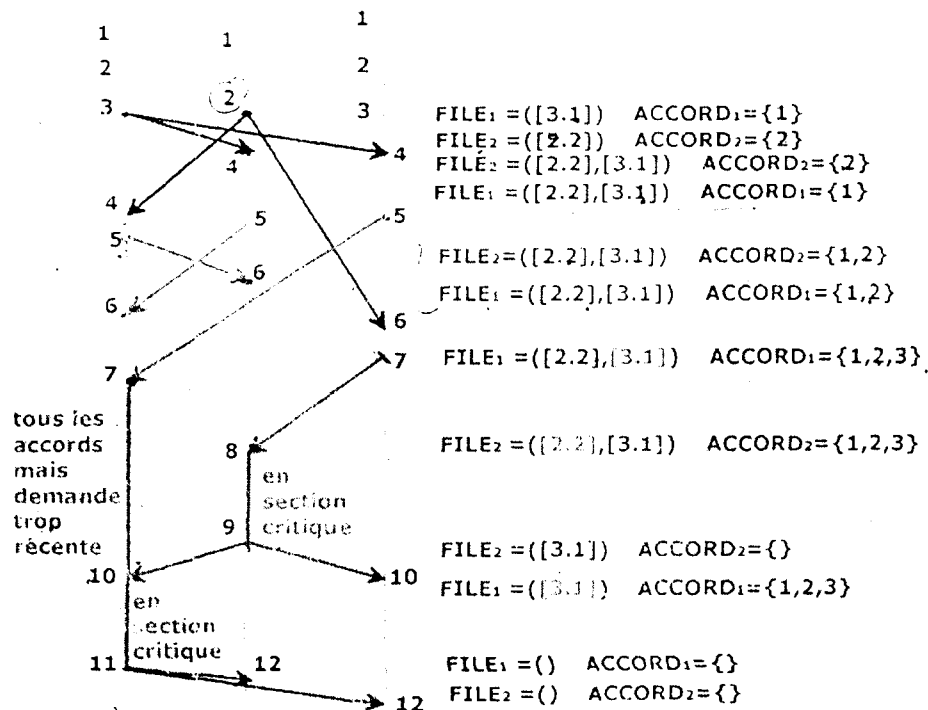
Dans la figure ci-dessous

- les envois de messages de type REQUETE correspondent aux flèches bleues,
- les envois de messages de type REPONSE correspondent aux flèches vertes,
- les envois de messages de type LIBERATION correspondent aux flèches rouges.

On y a matérialisé les files des requêtes et les ensembles de sites dont la réponse est arrivée pour les sites demandeurs (on n'a pas fait apparaître ces informations pour le site S_3 car ce site n'a pas demandé à entrer en section critique).

On peut observer que sur le site S_1 , au temps 7.1, les réponses des deux autres sites ont été reçues mais la demande de S_1 n'est pas la première dans sa file des requêtes: il ne peut donc rentrer immédiatement en section critique et doit attendre de pouvoir retirer la demande qui se trouve devant la sienne. Cette période est matérialisée en rouge. Les sections critiques sont matérialisées en vert.

Pour voir une version animée de cet exemple, cliquez sur la figure.



Exclusion mutuelle : algorithme de Lamport

Cet algorithme, proposé en 1978, vise à satisfaire les demandes des différents sites dans l'ordre où elles sont formulées. Cela suppose évidemment l'existence d'un ordre dans les requêtes: des horloges logiques scalaires sont utilisées à cet effet.

L'algorithme

Cet algorithme suppose que les canaux de communication entre les différents sites respectent la propriété FIFO. Chacun des composants du système utilise une horloge scalaire qu'il synchronise lors de la réception de messages.

Trois types de messages (estampillés lors de leur envoi) sont utilisés et chacun des messages sera systématiquement envoyé à tous les autres participants:

- **REQUETE** : un tel message est envoyé lorsqu'un site veut entrer en section critique ;
- **REPOSE** : un tel message est envoyé par un site qui reçoit un message du type précédent ;
- **LIBERATION** : un tel message est envoyé par un site lorsqu'il sort de section critique.

Chaque site gère une file d'attente dans laquelle il place, dans l'ordre induit par la valeur de leurs estampilles, toutes les requêtes pour entrer en section critique (y compris les siennes) et leurs estampilles. En fait la file des requêtes sera répliquée sur chaque site, si bien que chaque site peut décider de la possibilité d'entrer en section critique sur la base des informations qu'il possède.

L'algorithme est le suivant :

On notera HL_i l'horloge de S_i , EL_m l'estampille du message m et $FILE_i$ la file d'attente du site S_i (messages rangés par estampilles croissantes).

- si le site S_j veut entrer en section critique, il place sa requête dans sa file d'attente $FILE_j$ et envoie un message de type **REQUETE** à tous les autres sites. Il pourra entrer en section critique lorsqu'il aura effectivement reçu un accord (message de type **REPOSE**) de tous les autres sites ET que sa requête sera en tête de $FILE_j$;
- si le site S_j reçoit un message du type **REQUETE** en provenance du site S_i , il met à jour son horloge HL_j (selon la méthode habituelle à partir des valeurs de HL_i et de EL_m), il place la requête dans sa file d'attente $FILE_j$ et envoie à S_i un message de type **REPOSE** ;
- lorsque le site S_j sort de section critique, il envoie à tous les autres sites un message de type **LIBERATION** et retire sa requête de $FILE_j$;
- si le site S_j reçoit du site S_i un message de type **LIBERATION**, il met à jour son horloge et enlève de $FILE_j$ le message de type **REQUETE**.