

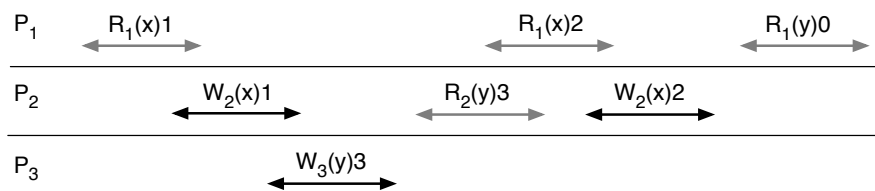
# Systemes et algorithmique repartis

ENSEEIH/DIMA  
1h30, documents autorises

14 novembre 2019

## 1 Questions de cours (6 points)

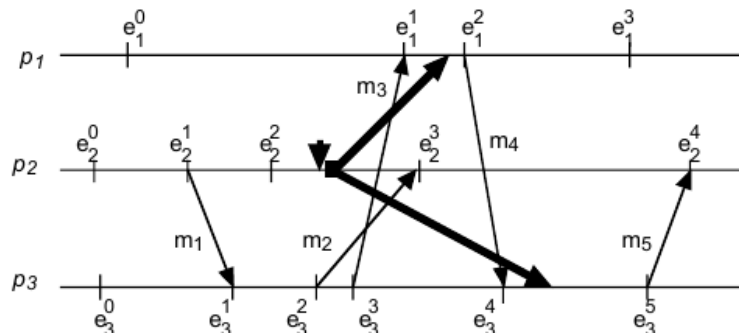
- On considere un systeme de trois processus muni d'horloges vectorielles. Soit deux evenements  $e$  et  $e'$  respectivement dates  $(2, 1, 5)$  et  $(6, 3, 5)$ . Quelle est la relation de precedance causale entre  $e$  et  $e'$  ?
- Representer graphiquement un scenario de deroulement du systeme conduisant a la production de  $e$  et  $e'$ .
- Meeme question sur la precedance causale si  $e$  et  $e'$  sont respectivement dates  $(2, 1, 5)$  et  $(6, 3, 4)$ .
- Representer graphiquement un scenario pour ce second cas.
- (2 points) On considere le scenario suivant d'accès aux copies de deux variables  $x$  et  $y$ , initialement nulles, et dupliques sur les sites  $P_1, P_2$  et  $P_3$ . La fleche placee en dessous du texte de l'operation represente la plage d'execution de l'operation.



Cette execution est-elle linearisable ? Verifie-t-elle la coherence sequentielle ? La coherence causale ? La coherence FIFO ? Justifiez chacune de vos reponses : seule la justification sera notee.

## 2 Prise de cliché (4 points)

On considere le systeme de trois processus decrit ci-dessous. Le processus  $p_2$  lance l'algorithme de Chandy-Lamport pour l'enregistrement d'etat, au point indique sur la figure. Les deux marqueurs qu'il emet sont indiquees par les fleches epaissees. La communication est FIFO entre deux sites. Compléter l'execution de l'algorithme en indiquant les messages echanges. Quel est l'etat enregistré par l'algorithme (etat des sites et messages en transit) ?



### 3 Détecteurs de défaillance (3 points)

On souhaite implanter un détecteur de défaillance parfait. On dispose d'une primitive  $send(from, to, msg)$  qui permet d'envoyer un message à un processus.

1. On suppose que le réseau est fiable et *synchrone* (existence d'une borne supérieure connue  $\delta$  sur le temps de transfert d'un message). Est-il possible d'implanter un détecteur de défaillance parfait  $P$ ? Si oui, donner son algorithme en pseudo-code. Si non, démontrer l'impossibilité.
2. On suppose que le réseau est fiable et synchrone, mais on ne connaît pas la valeur de la borne supérieure de transfert. Est-il possible d'implanter un détecteur de défaillance finalement parfait  $\diamond P$ ? Si oui, donner son algorithme en pseudo-code. Si non, démontrer l'impossibilité.
3. On suppose que le réseau est fiable et asynchrone. Est-il possible d'implanter un détecteur de défaillance parfait  $P$ ? Si oui, donner son algorithme en pseudo-code. Si non, démontrer l'impossibilité.

### 4 Algorithme de gestion de groupes (7 points)

On souhaite réaliser un protocole de gestion de groupes de sites s'appuyant sur un séquenceur. Le séquenceur est un site particulier du groupe qui est chargé d'une part de réaliser la diffusion des messages applicatifs aux membres du groupe, d'autre part de maintenir la liste courante des membres du groupe. Des sites peuvent quitter et rejoindre le groupe; un site peut tomber en panne (panne franche, *fail stop*), et se réintégrer après réparation. On suppose que la communication est fiable et que le temps de transmission d'un message est borné par une constante connue  $T$ . La primitive de communication que l'on veut réaliser est une diffusion fiable totalement ordonnée : un message diffusé parvient à tous les sites corrects (non en panne), ou à aucun; les messages sont délivrés dans le même ordre par tous les sites.

Pour réaliser la gestion de groupes et la diffusion, on dispose d'un système élémentaire de communication avec une primitive  $send(m, dest)$  qui envoie le message  $m$  au site destinataire désigné par  $dest$ , et une primitive  $receive(m, exp)$  qui attend un message et fournit en retour un message et l'identité de l'expéditeur.

Chaque site possède une variable  $seq$  indiquant l'identité du séquenceur. On répondra aux premières questions en supposant que le groupe est correctement initialisé, et que le service de communication est fiable et que le séquenceur n'est jamais défaillant. Ces points seront étudiés ensuite.

1. Expliquer comment le séquenceur peut maintenir la liste des sites présents. Donner l'algorithme des opérations qui permettent à un site de rejoindre ou de quitter le groupe.
2. Donner un algorithme permettant de réaliser la diffusion fiable totalement ordonnée.
3. Même question, mais en supposant maintenant que le système de communication peut perdre des messages (mais toujours avec l'hypothèse du délai de transmission borné : soit un message arrive avec un délai inférieur à  $T$ , soit il n'arrive pas).
4. Proposer un mécanisme pour initialiser un groupe de sites, et en particulier déterminer un séquenceur unique.
5. Comment détecter la défaillance du séquenceur?
6. Proposer un algorithme de traitement de la défaillance du séquenceur.