

Principes des Systèmes Répartis
Examen

10 décembre 1998

Durée: 3 heures, documents autorisés

*L'examen comporte 6 problèmes indépendants. Lire l'ensemble des énoncés avant de commencer à répondre. La longueur des énoncés n'est pas signe de difficulté mais elle est nécessaire à une bonne spécification des problèmes. La **clarté**, la **précision** et la **concision** des réponses, ainsi que leur **présentation matérielle**, seront des éléments importants d'appréciation.*

Problème 1 (4 points)

On considère un système de communications par messages entre deux sites p et q , dans lequel la communication est fiable (tout message arrive à destination), mais n'est pas FIFO. On souhaite pouvoir envoyer des messages particuliers dits 'balai' qui ont la propriété suivante (en désignant par m un message quelconque et par m_b un message balai :

- Si m_b est envoyé par p avant m , alors m_b est délivré à q avant m ;
- Si m_b est envoyé par p après m , alors m_b est délivré à q après m .

Donner un algorithme permettant d'assurer ces propriétés, en utilisant des horloges logiques. On utilisera les notations suivantes :

- Le système de communication initial (non FIFO) a deux primitives : $send(mess, dest)$: envoi du message $mess$ vers la destination $dest$; $receive(mess)$: réception du message $mess$ (on écrira la procédure de traitement de l'événement 'réception d'un message').
- Le système de communication que l'on veut construire a deux primitives : $transmit(mess, b, dest)$: envoi du message $mess$ vers la destination $dest$; b est *true* si $mess$ est un message balai, *false* pour un message ordinaire ; $deliver(mess, b)$: délivrance d'un message destiné au site sur lequel s'exécute cette primitive ; b a la même signification que pour $transmit$.

Quel est à votre avis l'utilité des messages balai ?

Problème 2 (4 points)

On considère un ensemble de processus communiquant au moyen de messages point à point. La communication est fiable (tout message arrive à destination).

- a) Rappeler la définition de l'ordre causal pour la délivrance des messages.
- b) On impose les règles suivantes pour les communications :
- Pour chaque message m , le processus destinataire envoie au processus émetteur un acquittement $ack(m)$ indiquant que m est bien arrivé.
 - Un processus n'envoie pas de nouveaux messages tant qu'il n'a pas reçu l'acquittement du dernier message qu'il a envoyé (sauf pour son premier message).

Montrer que ces règles assurent que les messages sont délivrés dans l'ordre causal.

c) Donner le principe d'un algorithme réalisant la communication point à point causale en utilisant les horloges vectorielles. Quels sont les avantages et les inconvénients de cette méthode par rapport à la méthode de la question b) ?

Problème 3 (2 points)

a) On rappelle qu'un état S' d'un système réparti est dit *accessible* à partir d'un état S (et on écrit $S \rightsquigarrow S'$) s'il existe un chemin de S vers S' dans le treillis des états cohérents. On rappelle qu'une propriété globale P d'un système réparti est dite *stable* si

$$\forall S : P(S) \text{ et } S \rightsquigarrow S' \Rightarrow P(S')$$

Rappeler le principe de la détection d'une propriété stable au moyen de l'algorithme de Chandy-Lamport.

b) On dit qu'une propriété globale P d'un système réparti est *inévitablement stable* si

$$\exists S : P(S) \text{ et } S \rightsquigarrow S' \Rightarrow P(S')$$

Peut-on détecter une telle propriété au moyen de l'algorithme de Chandy-Lamport ?

Problème 4 (4 points)

On considère un calcul réparti dans lequel tout processus peut être soit actif (il exécute son programme), soit passif (il n'exécute pas d'instructions). Un processus actif peut envoyer des messages aux autres processus ; ces messages sont de deux types : rouge et vert. Lorsqu'un processus reçoit un message rouge, il devient passif (s'il ne l'était pas déjà). Un processus actif peut passer spontanément (sans recevoir de message) dans l'état passif, mais un processus passif ne peut passer à l'état actif que sur réception d'un message vert (et un processus actif qui reçoit un message vert reste actif). Les canaux de communication entre processus sont tous FIFO. Initialement, au moins un processus est dans l'état actif.

a) Le calcul est dit *terminé* quand tous les processus sont dans l'état passif, et qu'aucun processus ne peut redevenir actif. Énoncer une propriété globale du système (état des processus et des canaux) qui caractérise la terminaison. Cette propriété est-elle stable ?

b) Écrire un algorithme permettant de détecter la terminaison, avec les hypothèses indiquées (sans utiliser d'anneau virtuel).

Problème 5 (3 points)

On considère un système asynchrone, dans lequel les processus peuvent avoir des défaillances de type panne franche, sans reprise (la défaillance est définitive).

a) Rappeler la définition et le principe de réalisation d'un protocole de diffusion fiable (*reliable broadcast*), en supposant disponible une primitive d'envoi de message point à point fiable (tout message arrive à destination).

b) On suppose que l'on dispose d'un détecteur de défaillances de type W (complétude faible, exactitude faible). Écrire l'algorithme d'un détecteur de défaillances de type S (complétude forte, exactitude faible) en supposant disponible un protocole de diffusion fiable.

Problème 6 (3 points)

On rappelle la définition du consensus : étant donné n processus $\{p_1, \dots, p_n\}$, il s'agit de construire un protocole défini comme suit.

A un instant initial (supposé défini pour tout processus), chaque processus p_i propose une valeur v_i . A l'issue du protocole, chaque processus p_i décide une valeur d_i . Les propriétés suivantes doivent être vérifiées :

1. deux processus corrects ne peuvent pas décider des valeurs différentes ;
2. un processus correct décide une seule fois, et la valeur qu'il décide est nécessairement un des v_i ;
3. tout processus correct décide au bout d'un temps fini.

On suppose que :

- les seules pannes possibles sont des pannes franches des processus ; un processus défaillant cesse de fonctionner et son arrêt est définitif,
- le système de communication est fiable (tout message parvient à son destinataire),
- le système est synchrone (le rapport des vitesses d'exécution de deux processus quelconques est borné, ainsi que le délai de transmission d'un message).

a) Donner dans ce cas un algorithme réalisant le consensus (exécuté par tout processus p_i).

b) Le consensus est dit *uniforme* quand on remplace la propriété 1 (accord) par la propriété d'accord uniforme : deux processus (corrects ou non) ne peuvent pas décider des valeurs différentes. Quel est l'intérêt du consensus uniforme ? Votre réponse à la question a) réalise-t-elle le consensus uniforme ? Si non, lui apporter les modifications nécessaires.