

Examen de rattrapage

Module de Algorithmique et Systèmes
d'exploitation Distribués

Durée : 01H30

Corrigé

Exercice 1 :

Question 1 : Rappelez brièvement qu'est ce qui motive le recours aux architectures parallèles.

Réponse :

Besoins d'application : Plusieurs applications nécessitent une capacité de traitement importante qui ne peut être fournie que par une architecture parallèle (ex : prévisions météorologiques).

Limite de l'architecture mono-processeur.

Aptitude de certaines application au traitement parallèle (ex : calcul numérique matriciel).

(2.5 points)

Question 2 : La gestion des ressources dans un système réparti peut se faire selon deux méthodes : allocateur unique ou plusieurs allocateurs. Quels sont les avantages et les inconvénients de chaque méthode ?

Réponse :

	<i>Avantages</i>	<i>Inconvénients</i>
<i>Allocateur unique</i>	L'état des ressources est centralisé, il n'y a pas de problème de cohérence.	La panne du site bloquerait le système.
<i>Plusieurs allocateurs</i>	En cas de panne d'un allocateur, le système continue de fonctionner avec les autres allocateurs.	Trafic très important entre les sites allocateurs pour reconstituer l'état global des ressources du système.

(2.5 points)

Question 3 : Dans quels cas peut-on utiliser l'algorithme du banquier d'évitement des interblocages dans un système distribué ?

Réponse :

Cas 1 : Existence d'un allocateur unique dans tout le système.

Cas 2 : Existence de plusieurs allocateurs, mais chaque allocateur gère exclusivement les ressources de son site.

(2.5 points)

Question 4 : Quel est le coût, en nombre de messages, pour l'entrée en Section Critique en utilisant l'algorithme de Lamport et l'algorithme de Ricart/Agrawala ? Justifiez.

Réponse :

Coût de l'algorithme de Lamport : $3(N-1)$ messages, N étant le nombre de site.*

$N-1$ messages de type Request.

$N-1$ messages de type ACK

$N-1$ messages de type Release

Coût de l'algorithme de Ricart-Agrawala : $2*(N-1)$ messages, N étant le nombre de site.
 $N-1$ messages de type Request.
 $N-1$ messages de type ACK

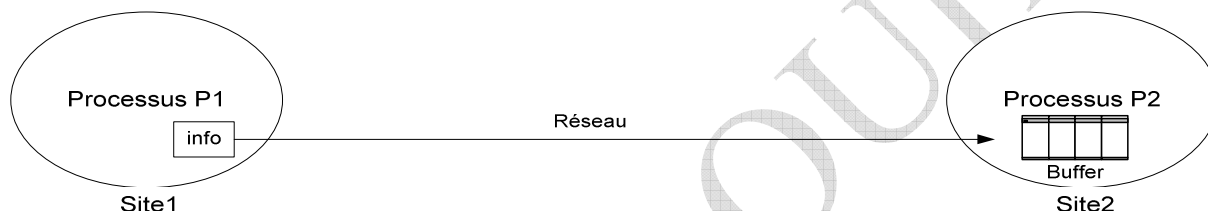
Les messages de type Release n'existent pas dans l'algorithme de Ricart – Agrawala .

(2.5 Points).

Exercice 2 :

On considère un système distribué constitué de deux processus P1 et P2 situés sur deux sites différents qui sont respectivement Site1 et Site2. P1 produit et envoie cycliquement une information « info » vers P2 à travers le réseau. Le processus P2 effectue lui aussi un travail cyclique qui consiste à consommer les informations parvenues à son buffer. Cependant, on impose la règle suivante : P2 doit consommer les informations dans le même ordre que leur envoi par P1.

On suppose que la voie de communication est fiable, c'est-à-dire qu'il n'y a pas de perte de messages, cependant les messages délivrés ne sont pas forcément dans l'ordre FIFO.



Travail à faire : On vous demande de proposer une méthode pour respecter la règle énoncée. Pour chacun des deux cas suivants, vous devez expliquer clairement le principe de la méthode proposée, de décrire précisément vos déclarations et d'écrire les codes des processus P1 et P2 :

Cas 1 : Le buffer de P2 est illimité.

On utilise le principe des horloges scalaires (logiques) : chaque information envoyée par P1 est estampillée avec la valeur d'horloge $H1$. Du côté de P2, on utilise aussi une horloge logique $H2$ qui s'incrémente entre deux consommations. On ne consomme une information que si $H1=H2$.

<pre> Processus P1 Entier : H1 ; Début H1 :=0 ; cycle Produire « info » ; H1=H1+1 ; Envoyer(P2, « info », H1) ; fncycle Fin.</pre>	<pre> Processus P2 Entier : H2 ; Début H2 :=1 ; Cycle 'rechercher dans le buffer l'information ayant l'estampille H1 = H2 ; Si cette information existe Alors Consommer(« info ») ; H2 :=H2+1 Finsi fncycle Fin.</pre>
--	--

(06 points)

Cas 2 : Le buffer de P2 est constitué d'une seule case.

On n'est pas obligé d'estampiller les messages de P1. Par contre, on doit bloquer P1 tant que sa dernière information qui a été envoyée n'a pas été consommé par P2. L'échange de message se fait donc dans les deux sens.

<p><i>Processus P1</i></p> <p><i>Début</i></p> <p><i>Produire « info » ;</i> <i>Envoyer(P2, « info ») ;</i> <i>cycle</i></p> <p><i>attendre message de P2</i> <i>Produire « info » ;</i> <i>Envoyer(P2, « info ») ;</i> <i>fincycle</i></p> <p><i>Fin.</i></p>	<p><i>Processus P2</i></p> <p><i>Début</i></p> <p><i>Cycle</i></p> <p><i>Tantque buffervide</i> <i>Faire</i> <i>Fait ;</i> <i>Consommer(« info ») ;</i> <i>Envoyer(P1,« poursuivre production »)</i></p> <p><i>fincycle</i></p> <p><i>Fin.</i></p>
---	--

(4 points)

Dr Mourad LOUKAM