

Contrôle de Crypto/SD

Exercice 1

Pour chacune des affirmations suivantes dites si elle est vraie ou fausse dans un système distribué. Les deux mots canal et lien sont synonymes dans cette évaluation.

- | | |
|--|-------------|
| a/ Les processeurs partagent leur mémoire | Vrai – Faux |
| b/ Les messages sont globalement gérés en FIFO | Vrai – Faux |
| c/ Chaque nœud dispose de ses propres variables. | Vrai – Faux |
| d/ Un nœud peut traiter plusieurs messages simultanément | Vrai – Faux |

Exercice 2

Algorithme Réveil (Sur un réseau de topologie quelconque)

Messages M Constante Voisins : ensemble des canaux

Variables C : canal ; prem_reception : booléen initialisé à VRAI

Spontanément :

Si prem_reception Alors prem_reception \leftarrow Faux ; Envoyer M à Voisins FinSi

A la réception de M sur le canal C :

Si prem_reception et $(\text{Voisins} \setminus \{C\} \neq \emptyset)$ Alors

premiere_reception \leftarrow FAUX ; Envoyer M à Voisins $\setminus \{C\}$

FinSi

Les questions suivantes portent sur l'algorithme Réveil. Seules les réponses argumentées seront considérées.

Question 1 : Cet algorithme peut-il avoir plusieurs initiateurs ?

Question 2 : Si une action spontanée a été exécuté, quel est le nombre minimal de message que peut recevoir un nœud non initiateur ?

Question 3 : Si une action spontanée a été exécuté, quel est le nombre maximal de messages que peut envoyer un nœud ?

Exercice 3

Dans un réseau R, chaque nœud dispose d'une Constante ListeRessources qui liste l'ensemble des ressources (scanner, imprimante, Lecteur de carte SD, ...) disponible sur ce nœud.

Une fonction avec un résultat booléen appelée `TestPrésence (Res, ListeRessources)` permet de tester si la ressource `Res` est présente dans la liste `ListeRessources`. `TestPrésence (Res, ListeRessources)` répond vrai si et seulement si La valeur de `Res` est présente dans la liste `ListeRessources`.

On souhaite écrire un algorithme distribué avec un unique initiateur pour compter le nombre de nœuds du réseau disposant de la ressource `Res`.

Ecrivez un algorithme distribué pour résoudre ce problème.

Vous pouvez par exemple vous inspirer de l'algorithme de circulation de jeton unique (CIJU) sur un graphe quelconque, écrit en TD. L'idée étant qu'au fur et à mesure de la circulation du jeton on conserve le nombre de nœud traversé possédant la ressource `Res`. A la fin de la circulation, l'initiateur aura la réponse souhaitée.