

UNIVERSITE Mohamed Khider BISKRA
1^{ère} année Master. Département d'informatique
Matière : Systèmes distribués

Série d'exercices n° 4
Exclusion mutuelle distribuée.

Exercice n°1

1. Quels sont les avantages et les inconvénients de la gestion de l'exclusion mutuelle par un coordinateur ?
2. Si un site ne reçoit pas un OK, quels types de problèmes peuvent être causés? Proposer une solution.
3. Est-ce que l'algorithme centralisé produit la famine ?
4. Faire une comparaison entre l'algorithme centralisé et l'algorithme de permission.
5. Dans quels cas l'algorithme de permission est non recommandé ?

Exercice n°2

Cet exercice porte sur le problème de l'exclusion mutuelle dans un système repartit. On s'intéresse à une ressource qui peut être partagée par n sites mais qui n'est accessible que par un seul à un moment donné. On doit écrire un protocole d'accès à cette ressource assurant les conditions d'exclusion mutuelle en se basant sur l'algorithme de Ricart-Agrawala. Chaque site désirant la ressource va demander la permission à tous les autres. On aura besoin d'estampiller les messages avec des marques d'horloges tel qu'aucune horloge globale n'est disponible. Par conséquent, on départage les conflits en étiquetant chaque demande par heure on l'a faite. Les demandes les plus anciennes sont les plus prioritaires.

On décrit un mécanisme d'estampillage basé sur chaque émission ou réception d'un message :

Lors de la réception d'un message <M,h>	Lors de l'envoi des données M
$H_i = \max(H_i, h) + 1;$ Traiter(M) ;	$H_i = H_i + 1;$ Envoyer(M, H_i);

1. Quelles sont les variables locales gérées par chaque site ?
2. Donner leurs valeurs initiales ?
3. Expliquer le comportement d'un site demandeur d'une ressource critique en précisant les changements sur variables nécessaires.
4. Même question lorsque le site détenteur de la ressource termine de l'utiliser.
5. Décrire le comportement d'un site lorsqu'il reçoit un message.
6. Donner la complexité de l'algorithme et ses avantages.

Exercice n°3

Soit l'algorithme de Chang-Roberts suivant pour l'élection dans un système distribué.

Initialement, un (ou plusieurs) processus commence(nt) une élection en envoyant un message;

- Un tel processus se marque comme participant à l'élection.
- L'arrivée d'un message à un processus non marqué provoque sa participation à l'élection.
- Le processus participant à l'élection (marqué) qui reçoit son propre numéro est élu et doit diffuser son identité aux autres.

L'objectif de cet algorithme est donc d'élire parmi tous les sites, celui qui possède l'identificateur max. Autrement, les messages d'élection venant des sites d'identificateur inférieur sont arrêtés.

Site i

Déclaration: Var état ∈ {passif, candidat, élu, battu} ; **booleen** est_initiateur;

Procédure Initialisation:

Procédure Detecter_Panne: est_initiateur ← vrai;

Procédure Lancer_Election:

Si est_initiateur **alors**

état ← candidat; envoyer <Election, Myid> au succ. **Finsi**

Procédure Reception<Leader,id>:

si Myid ≠ id **alors** id élu ← id; état ← battu; envoyer <Leader,id> au succ.;

sinon terminaison **finsi**

Procédure Reception<Election,id>:

.....

Ecrire les procédures de l'initialisation et réception d'un message d'élection en prévenance du site id.

Exercice n°4

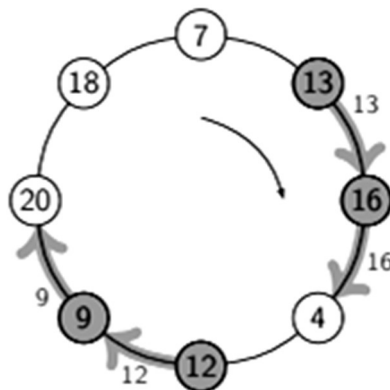
Soit un système distribué de n sites connectés via une topologie du réseau quelconque (graphe) où les canaux de communication sont fiables et synchrones. Chaque site possède un ID unique. Selon l'algorithme de BULLY, lorsqu'un site S_i détecte une panne d'un coordinateur, il initie l'algorithme d'élection en exécutant l'Initiateur (S_i). Cet algorithme va provoquer l'exécution de l'algorithme NonInitiateur(S_j) par d'autres sites S_j . Les demandes d'élection sont diffusées seulement aux sites dont l'ID est plus grand. Un site répond à ceux de numéros inférieurs au sien. Autrement, un site qui ne reçoit aucune réponse constate qu'il est élu. Soient les procédures suivantes :

Initialisation	Lancement_élection par S_i	Reception_mgs(elected, S_j) par S_i
Pour tout i faire Demande_Elect_ i ← faux; Leader ← maxEntier; Elu ← faux;	Demande_Elect_ i ← vrai; Pour (tout $j > i$) faire envoyer(S_i , Election); Armer délai de garde T ; // (*)	Demande_Elect_ i ← faux; Leader = S_j ;

1. Quels sont les types des messages échangés durant le déroulement de cet algorithme ?
2. A quoi sert l'instruction (*) dans la procédure **Lancement_élection** exécutée par le site S_i ?
3. Quel est le comportement de S_i lorsqu'il envoie un message (S_i , Election)?
4. Le délai de garde T (*) influence-t-il ce comportement ? Expliquer.
5. Quelles sont les tâches d'un NonInitiateur(S_i)?
6. Ecrire les procédures suivantes :
 - a. **Réception_mgs(S_j , Election) par S_i** : elle décrit le comportement d'un site S_i lorsqu'il reçoit un message de type Election en prévenance de S_j .
 - b. **Déclenchement_délai_T(S_i)** : elle décrit le comportement d'un site S_i lorsque le délai de garde T (*) est écoulé.
7. Quelle est la complexité de l'algorithme de BULLY au pire des cas où le site de plus petit identificateur lance l'élection ? Justifier en précisant les types des messages échangés.
8. Comparer les deux algorithmes d'élection de Bully et de Chang et Robert en termes de complexité au pire des cas.

Exercice n°5

Soit I un système distribué qui se compose de 8 sites, connectés sous un anneau où $I = \{16, 4, 12, 9, 20, 18, 7, 13\}$. Init est l'ensemble des initiateurs d'élection qui peuvent être un ou plusieurs.



- a. Supposons que $Init = \{9, 12, 13, 16\}$, donner les valeurs de la variable locale est_initiateur de tous les sites.
- b. Faire fonctionner l'algorithme de Chang Robert sur I avec $Init$. Donner les valeurs des différentes variables des différentes étapes.