

TP N° 2 (Exclusion mutuelle dans les systèmes répartis)

L'objectif de ce TP est l'implantation de processus communiquant par passage de messages en se basant l'algorithme de Lamport.

Rappel de l'algorithme d'exclusion mutuelle en distribué :

- Chaque processus envoie aléatoirement un certain nombre de messages à destination des autres processus.
- Dater chaque message à l'aide d'une estampille scalaire modifiée.
- Afficher uniquement les messages de façon à simuler une délivrance causale.
- Implanter l'algorithme de Lamport :

· Objectif : Chaque site gère une file d'attente dans laquelle il place, dans l'ordre induit par la valeur de leurs estampilles, toutes les requêtes pour entrer en section critique (y compris les siennes) et leurs estampilles. En fait, la file des requêtes sera répliquée sur chaque site, si bien que chaque site peut décider de la possibilité d'entrer en section critique sur la base des informations qu'il possède.

· Hypothèse :

- Cet algorithme suppose que les canaux de communication entre les différents sites respectent la propriété FIFO.
- Horloges logiques scalaire HL_i sur chaque site i
- Trois types de messages diffusés à tous :
 - Requête: envoyé quand site veut SC
 - Réponse : envoyé pour répondre à requête
 - Libération : envoyé par site qui sort de SC
- Une file $FILE_i$ de messages ordonnés par leur estampille EL_m par site i

· Algorithme :

si un site S_i veut entrer en SC :

- il place sa requête dans $FILE_i$ et envoie message *requête* à tous les autres
- Si entre en SC quand reçu *réponse* et quand sa requête est en tête de $FILE_i$

Quand un site S_j reçoit *requête* de S_i :

- met à jour HL_j (en utilisant HL_j et EL_m)
- place *requête* dans $FILE_j$ et envoie sa réponse *réponse* à S_i

Quand S_i sort de SC il envoie message *libération* à tous les sites et retire sa *requête* de $FILE_i$

si le site S_j reçoit un message *libération* de S_i il met à jour son horloge et enlève *requête* de $FILE_j$

Travail à effectuer

L'objectif étant de créer 3 processus, de les dérouler en une boucle infinie utiliser la fonction random (man rand) pour que chaque site décide aléatoirement s'il change d'état ou non et s'il souhaite rentrer en section critique.

Si oui, modifier les horloges locales et faire envoyer automatiquement les messages de requêtes aux autres sites conformément à l'algorithme donné.

Implanter une file d'attente locale qui sera mise à jour en fonction des messages reçus.