
IF4 DIST - TP

Exclusion mutuelle dans les systèmes répartis

Objectif :

Implantation de processus communiquant par passage de messages en vous basant sur le programme que vous trouverez à l'adresse www.esiee.fr/~grandpit : "*TP-DIST.c*" (un client "*TP-DIST-CL.c*" est aussi disponible pour tester rapidement le fonctionnement des serveurs).

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 : satisfaire les demandes de différents sites dans l'ordre des demandes d'entrée en section critique après accord de tous les autres sites
 - Principe : distribution d'une file d'attente associée à un mécanisme d'estampillage des messages. Basé sur un jeton circulant dans un anneau logique de processus.
 - Hypothèse :
 - Canaux de comm. FIFO
 - Horloges logiques scalaires 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
 - S_i entre en SC quand reçoit *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 Si sort de SC il envoie message *libération* à tous les sites et retire sa *requête* de FILEi

Si le site Sj reçoit un message *libération* de Si il met à jour son horloge et enlève *requête* de FILEj

Programme fourni :

Il faut compléter le programme TP.c. qui devra être exécuté sur 3 machines (la machine courante et deux machines distantes).

Pour le tester vous pouvez le compiler puis ensuite : vous connecter sur 2 autres machines, par exemple si machine1.esiee.fr est la machine locale, faite *ssh machine2.esiee.fr* dans une autre console et *ssh machine3.esiee.fr* dans la troisième console.

Puis sur machine1 faire *./TP 7000 machine1.esiee.fr machine2.esiee.fr machine3.esiee.fr* Ce programme attend que le même programme soit lancé sur machine2 et machine3.

Il faut donc faire *./TP 7000 machine1.esiee.fr machine2.esiee.fr machine3.esiee.fr* sur machine2 et de même sur machine3.

Quand les trois sont lancés, ils exécutent chacun une boucle infini (il pourraient faire autre choses qu'attendre, voir code) et scrute à chaque tour de boucle si une demande de connexion distante à eu lieu. Si oui, le programme accepte la connexion et lit le message envoyé (par hypothèse toute demande de connexion est suivie de l'envoi d'un message) pour ensuite l'afficher.

Remarque : 7000 est le numéro de port du serveur de la machine1, 7001 sera le numéro du port du serveur de la machine2 etc. Seule l'adresse de base est à donner (ici 7000).

Quand les 3 sites sont lancés il est donc possible d'utiliser le programme CL.c pour se connecter à l'un des serveurs et lui envoyer un message qu'il affichera.

Exemple : *./CL machine3 7002 toto 3*

Ceci effectuera 3 connexions sur le port 7002 de la machine3

Travail à effectuer

Il faut donc compléter ce programme. Dans la boucle infini utiliser la fonction random (man rand) pour que chaque site décide aléatoirement si il change d'état ou non et si il souhaite rentrer en section critique. Si oui, modifier les horloges locales et faire envoyer automatiquement les messages de requêtes aux autres site conformément à l'algorithme donné. Implanter une file d'attente locale qui sera mise à jour en fonction des messages reçus.