Université Paris 7
M2 II
Programmation répartie

TP 2

Lecteurs / Ecrivains

Le but est d'écrire une application distribuée client/serveur permettant de modifier et de consulter une base de données. Le serveur pourra traiter plusieurs demandes concurrentes.

On représentera la base de données par un tableau d'entiers et on ne s'occupera pas de vérifier que l'utilisateur a bien le droit de consulter ou de modifier la base.

Le serveur attend les requêtes des clients sur le port 1027, le client peut faire soit une demande de consultation soit une demande de modification. A la réception d'une requête le serveur lance, suivant le cas, une thread lectrice ou rédactrice. La synchronisation entre les threads se fait par des sémaphores.

1. Réaliser cette application distribuée en utilisant la politique de priorité suivante: *Priorité des lecteurs sur les rédacteurs.* Si un lecteur se présente alors que des lecteurs sont en train de lire la base de données, le lecteur peut consulter la base. Les lecteurs peuvent donc occuper indéfiniment la base de données. Il y a famine des écrivains.

2. Réaliser cette application distribuée en utilisant la politique de priorité suivante: *Priorité égale pour les rédacteurs et les lecteurs.* Si un lecteur utilise la base de données, tous les lecteurs nouveaux peuvent y accéder jusqu'à l'arrivée d'un rédacteur. A partir de ce moment tous les nouveaux arrivant attendent. Quand le dernier lecteur a fini, il réveille l'écrivain en attente. Quand cet écrivain a terminé, il réveille le premier processus en attente, si plusieurs lecteurs se suivent dans la file d'attente ils accedent aux fichiers.

3. Réaliser cette application distribuée en utilisant la politique de priorité suivante: *Priorité des rédacteurs sur les lecteurs.* Dés qu'un rédacteur réclame l'accès à la base de données, il doit l'obtenir le plus tôt possible sans réquisition (i.e. sans arrêer les utilisateurs actuels de la base de données).