

Algorithmique et Programmation Parallèle

TD 2 MT – Conditions, sémaphores

Exercice 1 : Découverte condition

Nous considérons 2 threads A et B. A doit afficher une valeur calculée par lui-même et ainsi qu'une valeur calculée par B : les 2 valeurs affichées doivent être identiques.

Le fichier `condition.c` essaie d'implémenter ce comportement mais n'est pas correct.

1. Après compilation, décrire le comportement obtenu (effectuer plusieurs essais) et expliquer le problème.

Pour corriger le programme, 2 implémentations sont demandées :

2. En utilisant les variables mutex et conditionnelle déjà présentes.
3. En réécrivant le programme avec les sémaphores.

On souhaite désormais effectuer ce travail en boucle tel que dans `boucle.c`.

4. Rajouter les synchronisations nécessaires pour obtenir le résultat correct (les valeurs de A et de B doivent être identiques à chaque tour de boucle).

Exercice 2 : Max d'un tableau (version sémaphore)

Reprenez l'exercice 3 du TD précédent (« Recherche parallèle du max d'un tableau ») en remplaçant l'utilisation du *mutex* par l'utilisation d'un *semaphore*.

Exercice 3 : Barrière multithread

La bibliothèque `pthread` fournit un objet `pthread_barrier_t` permettant de définir un point de synchronisation entre threads. On se propose ici d'implémenter nous-même une fonctionnalité similaire en utilisant des variables de condition.

Analyser le fichier source `barriere.c` et le compiler. Implémenter la structure de barrière ainsi que les fonctions d'initialisation et d'attente.