

Programmation Système

Threads

Juliusz Chroboczek
Khouloud Zine Elabidine

10 décembre 2012

Exercice 1 (Introduction aux *threads*). Écrivez un programme qui crée un *thread* qui affiche la liste des entiers de 1 à 100 pendant que le programme principal affiche la liste des lettres de *A* à *Z*. Pourquoi est-il important d'appeler `pthread_join` à la fin ?

Exercice 2 (Exclusion mutuelle). Écrivez un programme qui crée un *thread*. Le programme principal et le fils partageront une variable initialisée à 0, qu'ils incrémenteront¹ chacun un million de fois ; vérifiez que la variable vaut deux millions à la fin. Pourquoi faut-il protéger la variable par un *mutex* ?

Exercice 3 (Variables de condition).

1. Modifiez le programme précédent pour que le *thread* décrémente la variable partagée au lieu de l'incrémenter. Vérifiez que la variable vaut 0 à la fin.
2. On demande maintenant que la variable ne soit jamais négative (pensez à votre compte bancaire). Modifiez la fonction principale du *thread* pour que, lorsque la variable vaut 0, il relâche le *mutex*, fasse un appel à `sched_yield`, puis essaie de nouveau de décrémente la variable. Mesurez le temps d'exécution de votre programme à l'aide de la commande `time`.
3. Modifiez le programme précédent pour qu'il utilise une variable de condition pour synchroniser les deux *threads*. Le temps d'exécution a-t-il changé.

1. Incrémenter une variable, c'est lui ajouter 1.