

TP Programmation répartie

Thread – Création - Terminaison

Exercice 1 :

Ecrire un programme qui génère deux *threads*. Ces deux *threads* attendent 2 secondes utiliser `sleep()` affichent leur *pid* et la valeur d'un entier passé en argument (de la fonction).

Exercice 2 :

Rajouter au programme précédent l'appel qui permet au processus principal d'attendre la terminaison des threads avant de se terminer lui-même. Le code de retour de ces deux *threads* doit être récupéré et affiché par le processus principal. Dans les exercices qui suivent, le processus principal devra systématiquement appeler cette fonction et se terminer après ses threads.

Exercice 3 :

Créez 2 threads. Vous devez passer 3 arguments aux threads : un entier, un *float*, une chaîne de caractère.

Exercice 4 :

Lancer 10 threads au travers d'une boucle *for*. Les threads doivent être numérotés de 1 à 10. La valeur de « *i* » de la boucle *for* doit être passée en argument et affichée par le thread lui-même.

Les valeurs de « *i* » sont-elles cohérentes ? Sinon proposer une solution (que vous devez tester) pour afficher correctement le numéro du thread.

Exercice 5 :

Testez l'utilisation de `exit()` ou de `exec()` dans un thread ou le processus principal.