

TD N°2 : Programmation parallèle avec Pthreads

Exercice 1

1. Ecrire un programme ayant le comportement suivant :
 - Deux threads sont créés.
 - Le premier thread affiche "Hello" et le second thread affiche "World".
 - Le thread principal attend la terminaison des deux threads créés.
 - L'ordre d'affichage ne doit pas forcément être respecté. On peut avoir comme résultat d'exécution "Hello World" comme "World Hello".
2. Ecrire un programme ayant le même comportement que dans la question 1, mais où l'ordre d'affichage doit être respecté (le résultat d'exécution doit être "Hello World").

Exercice 2

1. Ecrire un programme ayant le comportement suivant :
 - Quatre threads sont créés.
 - Chaque thread affiche le message "Hello World !", son PID (avec `getpid()`) ainsi que la valeur opaque retournée lors de sa création (son identifiant).
 - Le thread principal attend la terminaison des quatre threads créés.
2. Modifier le programme de la question précédente pour que chaque thread affiche en plus du message "Hello World !", son numéro d'ordre uniquement.

Exercice 3

Ecrire un programme qui calcule la somme des éléments d'un tableau en utilisant quatre threads (le thread principal et trois threads créés). Chaque thread créé calcule la somme pour une portion du tableau et transmet le sous-total au thread principal. Ce dernier récupère les sous-totaux et fait le total, puis l'affiche. Pour écrire le programme, il est nécessaire de prendre en considération les conditions suivantes :

- Les éléments du tableau doivent être de type "entier long".
- Le tableau doit comporter 15 éléments et doit être local au thread principal.
- La variable qui contient le total doit être locale au thread principal.