

Programmation Système TD n° 5

Notions abordées :

- **Threads POSIX** (`pthread_create`, `pthread_join`, `pthread_exit`)
- **Mutex** (`pthread_mutex_lock`, `pthread_mutex_unlock`)

Exercice 1

- Écrire un programme qui affiche "main init", crée 2 threads : l'un affiche "Hello World A", l'autre affiche "Hello World B", et affiche "main end". Quelles sont les sorties possibles de ce programme ?
- Modifier le programme précédent de façon à ce que les 2 threads se terminent avant que le programme affiche "main end".
- Modifier le programme précédent de façon à ce que les 2 threads réalisent respectivement 3 et 2 fois l'affichage de "Hello World A" ou "Hello World B". Donner un exemple de sortie pour ce programme.
- Synchroniser les 2 threads de façon à ce que la sortie du programme soit : "Hello World A", "Hello World B", "Hello World A", "Hello World B", "Hello World A".

Exercice 2

Écrire un programme qui initialise un compteur à 0, crée 2 threads : chacun de ces threads incrémente le compteur 10000000 fois, attend que les 2 threads se terminent et affiche la valeur finale du compteur. Quelle sera la valeur affichée si les 2 threads ne sont pas synchronisées ? Synchroniser les threads de façon à ce que la valeur affichée soit bien 20000000.

Exercice 3

On souhaite trouver toutes les occurrences d'un élément dans un tableau donné. Pour cela, si le tableau est long (plus qu'une constante `TAILLE_MIN`), on peut le couper en deux parties et effectuer la recherche indépendamment dans les deux moitiés, en confiant chaque partie à un nouveau thread travaillant en parallèle. Écrire un programme effectuant la recherche de cette manière. Le programme devra afficher un message pour chaque occurrence trouvée.

Exercice 4

Écrire un programme qui décrit un système de producteurs/consommateurs partageant un buffer borné à une seule place. Il s'agit pour les producteurs et les consommateurs de se synchroniser sur le buffer : Un producteur doit attendre que le buffer soit vide pour mettre la valeur produite dans le buffer. Un consommateur doit attendre que le buffer soit plein pour consommer la valeur.