

# Programmation Concurrente

## Threads POSIX

### Série 3

#### Exercice 1

Soit le programme multi-threadé ci-dessous créant 16 threads où chaque thread affiche son numéro :

```
#include <stdio.h>
#include <stdlib.h>
#include <pthread.h>

#define NUM_THREADS 16

void *thread(void *thread_id) {
    int id = *((int *) thread_id);
    printf("Hello from thread %d\n", id);
    return NULL;
}

int main() {
    pthread_t threads[NUM_THREADS];
    for (int i = 0; i < NUM_THREADS; i++) {
        int code = pthread_create(&threads[i], NULL, thread, &i);
        if (code != 0) {
            fprintf(stderr, "pthread_create failed!\n");
            return EXIT_FAILURE;
        }
    }
    return EXIT_SUCCESS;
}
```

#### Questions

- A votre avis, l'implémentation de ce programme est-elle correcte ? Vérifiez votre réponse en exécutant le programme **plusieurs fois**.
  - Si ce n'est pas le cas :
    - Expliquez pourquoi le comportement est incorrect.
    - Modifiez le code afin qu'il produise le comportement attendu.

## Exercice 2

On aimerait implémenter un programme créant un thread pour calculer la somme d'un tableau d'entiers. Ce thread doit calculer la somme des entiers du tableau et le programme principal (`main`) s'occupera d'afficher le résultat.

Les points suivants sont à respecter :

- Aucun argument n'est passé au thread.
- Le thread ne retourne aucune valeur.
- Les entiers sont entrés sur la ligne de commande (chacun séparé par un espace).
- La somme est stockée dans une variable globale.
- N'oubliez pas de vérifier qu'aucune des fonctions utilisées n'échoue ; en cas d'erreur, veuillez afficher le résultat sur le canal d'erreur (`stderr`).
- Ecrivez un `makefile` pour compiler votre code et faire le ménage (cible `clean`).

## Exercice 3

Nous aimerions implémenter une variante du programme de l'exercice 2. Celui-ci sera fonctionnellement identique, mais ne devra utiliser **aucune variable globale**.

Vous pouvez par contre passer un argument à votre thread et celui-ci peut retourner une valeur.