

TD – Multithreading

ING2 – Programation Système et Réseau

Année 2020 – 2021



Remarque : Tous les codes doivent être compilés avec l'option **-lpthread**.

Introduction

- 1 | Ecrivez un programme ayant le comportement suivant :
 - Des threads sont créés (leur nombre étant passé en paramètre lors du lancement du programme);
 - Chaque thread affiche un message (par exemple "hello world!");
 - Le thread "principal" attend la terminaison des différents threads créés.☐
- 2 | Modifiez le programme de l'exercice précédente pour que chaque thread affiche :
 - son PID (avec `getpid()`);
 - La valeur opaque retournée par `pthreadself`, par exemple avec :
`printf("%p\n", (void *) pthreadself());`
 - son numéro d'ordre passé comme argument. Vérifiez que le numéro d'ordre affiché par chaque thread est bien différent (corrigez votre programme le cas échéant).☐
- 3 | Implémentez un programme avec quatre threads. Créez une fonction `imprimerNumThread()` qui affiche trois fois, pour chaque thread, son numéro de thread. Que se passe-t-il quand tous les threads appellent la fonction en même temps? Assurez une exécution en ordre en utilisant un mécanisme de synchronisation. ☐

Calcul matriciel

- 4 | Implémentez un programme qui calcule le carré d'une matrice comportant 4 éléments par 4 threads. La matrice sera déclarée en tant que variable globale. Chaque thread ne devra s'occuper que d'une case. ☐

Chaîne de caractères

5

Implémentez un programme qui crée deux threads qui partagent une chaîne de caractères :

- Le premier thread dans une boucle demande à l'utilisateur de saisir une chaîne de caractères (tant que la chaîne n'est pas "sortir").
- Le deuxième thread affiche la chaîne courante toutes les 4 secondes.

□