

TD-TP3 : Threads POSIX

Exercice 1: Creation et attente de threads

Question 1: Ecrivez un programme ayant le comportement suivant :

1. Des threads sont crees (leur nombre etant passe en parametre lors du lancement du programme) ;
2. Chaque thread affiche un message (par exemple hello world !) ;
3. Le thread principal attend la terminaison des differents threads crees

Exercice 2: Identification des threads

Question 1: Modifiez le programme de la question precedente pour que chaque thread affiche :

- son PID (avec getpid()) ;
- La valeur opaque retournee par pthread_self, par exemple avec `printf("%p\n", (void *) pthread_self ())` ;

Exercice 3: Passage de parametres et exclusion mutuelle

Question 1: Modifiez le programme de la question precedente pour passer son numero d'ordre a chaque thread. Chaque thread doit ensuite l'afficher. Verifiez que le numero d'ordre affiche par chaque thread est bien different (corrigez votre programme le cas echeant).

Question 2: Declarez une variable globale *somme* initialisee a 0. Chaque thread doit, dans une boucle, ajouter 1 000 000 fois son numero d'ordre a cette variable globale (on veut bien faire 1 000 000 additions par thread, pas juste une). Affichez la valeur obtenue apres la terminaison de tous les threads. .

Question 3: Avec 5 threads (numeroes de 0 a 4), on devrait obtenir $(0+1+2+3+4)*1\,000\,000 = 10\,000\,000$. Corrigez votre programme s'il n'affiche pas systematiquement ce resultat.

Question 4: Modifiez votre programme pour ne plus utiliser de variables globales. Il faut donc passer les adresses des differentes variables necessaires en argument aux threads, dans une structure.