TP3 : Threads POSIX

RS : Réseaux et Systèmes

Deuxième année

★ Exercice 1: Création et attente de threads

▷ **Question 1:** Écrivez un programme ayant le comportement suivant :

1. Des threads sont créés (leur nombre étant passé en paramètre lors du lancement du programme) ;
2. Chaque thread affiche un message (par example « hello world ! ») ;
3. Le thread « principal » attend la terminaison des différents threads créés.

★ Exercice 2: Identification des threads

▷ **Question 1:** Modifiez le programme de la question précédente pour que chaque thread affiche :

— son PID (avec `getpid()`) ;

— La valeur opaque retournée par `pthread_self`, par exemple avec :
`printf("%p\n", (void *) pthread_self());`

★ Exercice 3: Passage de paramètres et exclusion mutuelle

▷ **Question 1:** Modifiez le programme de la question précédente pour passer son numéro d'ordre à chaque thread. Chaque thread doit ensuite l'afficher. Vérifiez que le numéro d'ordre affiché par chaque thread est bien différent (corrigez votre programme le cas échéant).

▷ **Question 2:** Déclarez une variable globale `somme` initialisée à 0. Chaque thread doit, dans une boucle, ajouter 1 000 000 fois son numéro d'ordre à cette variable globale (on veut bien faire 1 000 000 additions par thread, pas juste une). Affichez la valeur obtenue après la terminaison de tous les threads.

▷ **Question 3:** Avec 5 threads (numérotés de 0 à 4), on devrait obtenir (0+1+2+3+4)*1 000 000 = 10 000 000. Corrigez votre programme s'il n'affiche pas systématiquement ce résultat.

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