TP – Développement Multitâche
Découverte des Threads

Exercice 1

Voici le code suivant :

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

static void *task_a (void *p_data)
{
  puts ("Hello world A\n");
  (void) p_data;
  return NULL;
}

int main (void)
{
  pthread_t ta;

  puts ("main init");

  pthread_create (&ta, NULL, task_a, NULL);
  sleep(10);
  puts ("main end");

  return 0;
}
```

Ecrire un programme où un thread qui affiche un message en boucle toute les secondes. Le thread principal attend 5 secondes avant de se terminer (fonction sleep()). Que se passe-t-il lorsque le thread principal se termine ?
Quelle est la différence avec l'appel système fork() ? Expliquez cette différence.

Exercice 2

Ecrire un programme ayant deux threads qui affichent un nombre aléatoire (voir annexe) toutes les secondes pour l'un et 2 secondes pour l'autre. La fin du programme se fait par appui sur une touche (voir annexes).
Modifiez ensuite votre programme pour que les deux threads utilisent la même fonction tout en respectant les temps d'affichage indiqués précédemment (leur passer un paramètre).

Exercice 3

Ecrire un programme avec 100 threads dont la fonction :
• incrémente un nombre global indiquant le nombre de threads actifs.
• tire une valeur au hasard comprise entre 1 et 5.
• prévient l'utilisateur par un message signifiant le nombre de secondes qu'il va attendre ainsi que son numéro (celui qu'il a incrémenté) (exemple : "Le thread n°?? va attendre pendant ?? secondes...").
• attend pendant x secondes (tiré au hasard juste avant).
• puis termine en affichant un message de fin avec son numéro (exemple : "Le thread n°?? a fini !!").
Le thread principal attendra la fin de tous les autres threads.

ANNEXES

Fonction de lecture de caractère : int getchar()
Cette fonction permet de lire un caractère depuis le flux d'entrée. Celui-ci est inséré dans le flux d'entrée après appui sur la touche <ENTRÉE>. Cette fonction est bloquante et ne rend pas la main à moins d'avoir frapper la
touche <ENTRÉE>. La valeur retournée par la fonction est un unsigned char casté en un int représentant le code ASCII de la touche frappée.
Exemple 1 : frapper <ENTRÉE> vous donnera la valeur 10, qui est le retour à la ligne.
Exemple 2 : frapper la lettre 'w' puis <ENTRÉE> vous donnera la valeur 119 puis la valeur 10.

Génération de nombres aléatoires : srand() et rand()
La fonction srand() permet d'initialiser le générateur de nombres aléatoires. Elle n'est à appeler qu'une seule fois dans votre programme. En effet, si vous ne procédez pas ainsi la fonction rand() sortira toujours les mêmes nombres, et nous ne sommes donc plus dans de l'aléatoire. Examinez le morceau de code qui suit pour comprendre comment cela fonctionne...

```
#include <stdio.h>
#include <time.h>
#include <stdlib.h>
int main()
{
int val;
srand(time(NULL)); //Permet d'initialiser le générateur pseudo-aléatoire.
printf("Valeur tiree au hasard : %d",rand()%10);
return NULL;
}
```