

## TD n°6 : Threads

### **SOLUTIONS (ne pas distribuer aux étudiants)**

#### **Exercice 1 – Threads et passage d'arguments**

##### **Solution :**

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

void *ma_fonction_thread(void *arg)
{
    int nbsec = (int)(arg);
    printf("Je suis un thread et j'attends %d secondes\n", nbsec);
    sleep(nbsec);
    printf("Je suis un thread et je me termine");
    pthread_exit(NULL);
}

int main(void)
{
    int ret;
    pthread_t my_thread;
    int nbsec=5;

    /* on crée le thread */
    ret = pthread_create(&my_thread, NULL, ma_fonction_thread, (void *)nbsec);
    if (ret != 0)
    {
        fprintf(stderr, "Erreur de création du thread");
        exit (1);
    }
    pthread_join(my_thread, NULL); /* on attend la fin du thread */
    return 0;
}
```

#### **Exercice 2 – Threads et valeur de retour**

##### **Solution :**

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

void *ma_fonction_thread(void *arg)
{
    int resultat;
    printf("Je suis un thread. Veuillez entrer un entier :\n");
    scanf("%d", &resultat);
    pthread_exit((void*)resultat); /* termine le thread proprement */
}
```

```
int main(void)
{
    int ret;
    pthread_t my_thread;
    /* on crée le thread */
    ret = pthread_create(&my_thread, NULL, ma_fonction_thread, (void*)NULL);
    if (ret != 0)
    {
        fprintf(stderr, "Erreur de création du thread");
        exit (1);
    }
    pthread_join(my_thread, (void*)&ret); /* on attend la fin du thread */
    printf("Entier retourné dans le main, ret = %d\n", ret);
    return 0;
}
```

### Exercice 3 – Processus vs. Threads

#### *Solution :*

1)

```
Bonjour père : 1000
Bonjour père : 3000
Bonjour fils : 10
Bonjour fils : 30
```

2)

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

int i;

void addition()
{
    i += 10;
    printf("Bonjour thread, i est égal à %d\n", i);
    i += 20;
    printf("Bonjour thread, i est égal à %d\n", i);
}

int main()
{
    pthread_t thread_id;

    i = 0;

    /* Crée un thread */
    if (pthread_create(&thread_id, NULL, (void *(*))addition, NULL) == -1)
    {
        perror("Impossible de créer le thread");
    }

    /* Addition dans le processus principal */
    i += 1000;
    printf("Bonjour file d'exécution principale , i est égal à %d\n", i);
    i += 2000;
```

```
printf("Bonjour file d'exécution principale, i est égal à %d\n", i);

/* Attend que le thread se termine */
pthread_join(thread_id, NULL);

return 0;
}
```