

## Série 04

### La Gestion Des Threads Sous Linux

*Un thread (ou fil d'exécution en français) est une partie du code d'un programme (une fonction), qui se déroule parallèlement à d'autre parties du programme. Toutes les fonctions et types de données relatifs aux threads sont déclarés dans le fichier d'entête <pthread.h>*

#### Création de threads

```
int pthread_create ( pthread_t      *thread ,
                     pthread_attr_t *attr,
                     void          *nomfonction,
                     void          *arg );
```

#### Les 4 paramètres sont:

1. pointeur vers une variable **pthread\_t**, où sera stocké l'identifiant du nouveau thread;
2. pointeur vers un objet d'attribut de thread qui contrôle les détails de l'interaction du thread avec le reste du programme (**NULL** signifie des attributs par défaut) ;
3. pointeur vers le code que doit exécuter le thread;
4. Argument transmis à la fonction de thread lorsque celui-ci commence à s'exécuter.

Un appel à `pthread_create` se termine immédiatement et le thread original continue à exécuter l'instruction suivant l'appel. Pendant ce temps, le nouveau thread débute l'exécution de la fonction de thread. Linux ordonne les deux threads de manière asynchrone et votre programme ne doit pas faire d'hypothèse sur l'ordre d'exécution relatif des instructions dans les deux threads.

Soit le programme **thread-create.c** suivant :

```
#include <stdio.h>
#include <pthread.h>
int i;
```

```
void addition()
{
    i = i + 10;
    printf ("Je suis le thread fils, la valeur de i= %d\n", i);
    i = i + 20;
    printf ("Je suis le thread fils, la valeur de i=%d\n", i);}
```

```
main()
{
    pthread_t num_thread;
    i = 0;
if (pthread_create (&num_thread, NULL, (void *(*)( ))addition, NULL) == -1)
    perror ("pb pthread_create\n");
i = i + 1000;
printf ("Je suis le thread principal, la valeur de i= %d\n", i);
i = i + 2000;
printf ("Je suis le thread principal, la valeur de i= %d\n", i);
}
```

Commentez les traces générées après avoir compiler en utilisant la commande:

```
gcc -o thread-create thread-create.c -lpthread
```

#### Exercice 1 Attendre la fin d'un thread

Ecrivez un programme qui a pour rôle :

1. un certain nombre de threads sont créés (leur nombre étant passé en paramètre lors du lancement du programme) ;
2. Chaque thread affiche le message 'Je suis un thread' ;
3. Le thread principal doit attendre la terminaison des différents threads créés