

TD n° 4 : Threads

Objectif : Manipuler des threads en langage C++

Exercice 1 – Threads et passage d'arguments

Écrire un programme dans lequel le thread principal crée un thread qui dort un nombre de secondes passé en argument lors de sa création. Le thread principal devra attendre la terminaison de ce dernier.

Exercice 2 – Threads et attente de terminaison

Écrire un programme dans lequel le thread principal crée 2 threads, chargés d'afficher chacun des mots de la phrase « Hello world ! ». On veillera à s'assurer de l'affichage dans le bon ordre des mots à l'écran et ce, quelle que soit la manière dont sont ordonnancés les threads.

Exercice 3 – Processus vs. Threads

1) Quel résultats seront produits par le programme suivant ?

```
#include <iostream>
#include <sys/types.h>
#include <unistd.h>
#include <sys/wait.h>
#include <stdlib.h>
using namespace std;
int i;
int main()
{
    pid_t pid;
    i = 0;
    pid = fork();
    if (pid == -1)
        perror("Impossible de créer le processus fils");
    else if (pid == 0)
    {
        i += 10;
        cout << "Bonjour fils : " << i << endl;
        i += 20;
        cout << "Bonjour fils : " << i << endl;
        exit(0);
    }
    else
    {
        i += 1000;
        cout << "Bonjour père : " << i << endl;
        i += 2000;
        cout << "Bonjour père : " << i << endl;
        wait(NULL);
    }
    return 0;
}
```

2) Proposez une variante avec un thread dans laquelle la variable `i` est partagée par tous les fils d'exécution et vaut 3030 en fin d'exécution.