[C] Threads

Introduction

Le langage C supporte l'utilisation des **threads** qui sont des tâches sous-jacentes d'un processus permettant d'effectuer plusieurs tâches en parallèle.

Manuel

pthread_create

La fonction **pthread_create** va permettre d'exécuter une fonction dans un nouveau thread.

pthread_join

La fonction **pthread_join** va permettre d'attendre que le thread soit terminé avant de continuer à exécuter la suite du programme.

Exemple

Voici un exemple d'utilisation de thread très simpliste :

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

// Cette fonction sera exécutée par le thread
void *fonctionThread(void *arg) {
    int thread_num = *((int *)arg); // Récupération du numéro du thread

    // Affichage du numéro du thread
    printf("Je suis le thread n°%d\n", thread_num);
```

```
    // Il est nécessaire de renvoyer une valeur
    return NULL;
}

int main() {
    pthread_t thread1, thread2;
    int num1 = 1, num2 = 2;

    // Création du premier thread
    if (pthread_create(&thread1, NULL, fonctionThread, (void *)&num1) != 0) {
        fprintf(stderr, "Erreur lors de la création du thread 1.\n");
        return 1;
    }

    // Création du deuxième thread
    if (pthread_create(&thread2, NULL, fonctionThread, (void *)&num2) != 0) {
        fprintf(stderr, "Erreur lors de la création du thread 2.\n");
        return 1;
    }

    // Attente de la fin d'exécution des threads
    if (pthread_join(thread1, NULL) != 0) {
        fprintf(stderr, "Erreur lors de l'attente du thread 1.\n");
        return 1;
    }
    if (pthread_join(thread2, NULL) != 0) {
        fprintf(stderr, "Erreur lors de l'attente du thread 2.\n");
        return 1;
    }

    printf("Les threads ont terminé leur exécution.\n");

    return 0;
}
```

Revision #1
Created 2023-12-26 16:04:44 UTC by Elieroc
Updated 2023-12-27 08:59:26 UTC by Elieroc