|  | TP : découverte des threads en C sous Windows |
| --- | --- |
| Objectifs | |
| Découverte des threads en C | |
| Prérequis | Ressources à disposition / Environnement |
| Cours langage C | Environnement : PC sous Windows  Environnement de développement intégré (EDI) : CodeBlocks ou autre choix. |

1. Travail demandé
2. Exercice 1

Créer un projet en C dans CodeBlocks (ou autre IDE sous Windows) et copier dans le main le programme ThreadEx1.c proposé.

Examiner le code, compiler et tester le programme puis répondre aux questions suivantes :

* 1. **À quoi sert le type ThreadData ?**

Le type ThreadData sert à initialiser les données de début et de fin

* 1. **Quel type de variable représente LPVOID ?**

C’est un type où on définit un pointeur sur n’importe quoi.

* 1. **Même question pour DWORD.**

C’est un mot de 32 bits soit 4 octets non signé

* 1. **Que fait la fonction CreateThread ?**

Elle crée un thread pour un processus

* 1. **Parmi les arguments de la fonction CreateThread, 2 ont des valeurs particulières. Que représentent-ils ?**

SumThreadFunction est un pointeur et data est l’adresse du paramètre de SumThreadFunction

* 1. **Que fait la fonction WaitForSingleObject ?**

**Elle attend que le thread soit créé**

* 1. **Que fait la fonction GetExitCodeThread ?**

Il récupère la valeur de ThreadHandle et la range dans exitCode

* 1. **Lors de l’exécution du programme, expliquer pourquoi l’affichage est curieux et différent à chaque exécution.**

Parce que la fonction main et SumThreadFunction s’exécute en parallèle

* 1. **Qu’est-ce qu’un thread ?**

Il s’agit d’une fonction qui s’exécute en parallèle du main

1. Exercice 2

Modifier le programme de l’exercice 1 de façon à ce que la somme des entiers passés en paramètre au thread se fasse avec un pas (step) différent de 1, valeur qui devra être passée en paramètre.

Compiler et tester votre programme.

* 1. **Pourquoi le paramètre d’un thread, comme SumThreadFunction, est de type LPVOID ?**

Pour permettre de passer un pointeur sur n’importe quel type, donc sur le type de données qu’on souhaite.

1. Exercice 3
   1. **Un thread (sous Windows) peut-il retourner une valeur d’un type différent de DWORD ?**

On souhaite modifier le programme précédent de façon à ce qu’on puisse compter puis afficher dans le main le nombre de valeurs additionnés dans le thread.

**1ère solution**: utiliser une variable globale.

* 1. **Pourquoi est-il déconseillé d’utiliser une variable globale ?**

Non, un thread (la fonction) ne peut renvoyer qu’une variable de type DWORD.

En trop gros nombre, les variables globales complexifient la lecture de l’organisation du code.

Comme elles sont accessibles dans toutes fonctions, leur usage est dangereux. Par exemple, on peut modifier une variable globale sans s’en rendre compte, en croyant modifier une variable local.

**2ème solution** : utiliser une variable locale au main dont il faudra passer l’adresse (un pointeur) au thread.

* 1. **On dit que les threads d’un même programme partagent le même espace d’adressage. Expliquer ce que cela signifie.**