

## TP Processus légers « Thread »

### 1 But du TP

Le but de ce TP est d'écrire un programme en langage C qui utilise les processus légers et de vérifier que le système d'exploitation répartit ces threads sur les processeurs de l'ordinateur (architecture SMP bien sur !)

### 2 Calcul de PI

Pour calculer PI, nous allons calculer l'aire de la fonction  $f(x) = \frac{1}{(1+x^2)}$  pour un intervalle donné. On sait que la primitive de  $f(x)$  est la fonction  $\arctan(x)$  donc le calcul de l'aire, c'est-à-dire la somme de tout les trapèzes (au sens de Riemann...) entre 0 et un très grand nombre va nous donner une approximation de pi ce qui donne en langage C :

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

main (int argc, char **argv)
{
    register double width, sum;
    register int intervals, i;

    if (argc != 2)
    {
        printf ("Le programme prend un seul argument le pas de calcul...\n");
        exit (1);
    }
    else
    {
        /* récupère le pas */
        intervals = atoi (argv[1]);
        width = 1.0 / intervals;

        /* calcul de l'aire */
        sum = 0;
        for (i = 0; i < intervals; ++i)
        {
            register double x = (i + 0.5) * width;
            sum += 4.0 / (1.0 + x * x);
        }
        sum *= width;

        printf ("Estimation de pi is %f\n", sum);

        return (0);
    }
}
```

Analyser le programme. Comment le lance-t-on ? Donner un exemple.

### **3 Calcul de PI en utilisant les processus légers**

En se basant sur le programme précédant, transformer le, en y introduisant une fonction qui calcule la somme et qui prend en argument le début de l'intervalle de calcul.

Ecrire les programmes :

1. qui utilise cette fonction comme unique processus léger.
2. qui utilise cette fonction avec deux processus légers (un qui calcule la somme des nombres pairs, l'autre des somme des impairs)

Vérifier que les processus sont distribués sur les deux coeurs du microprocesseurs et que le programme 2 est plus rapide que le 1 !