

INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES DE OCCIDENTE

Departamento de Electrónica, Sistemas e Informática

INGENIERÍA EN SISTEMAS COMPUTACIONALES



PROGRAMACIÓN CON MEMORIA DINÁMICA TAREA 2. APUNTADORES A FUNCIONES

Autor: Gallegos, José Isaac

Presentación: 10 pts.
Funcionalidad: 60 pts.
Pruebas: 20 pts.

4 de junio de 2018. Tlaquepaque, Jalisco,

- Las figuras deben tener un número y descripción.
- Las figuras, tablas, diagramas y algoritmos en un documento, son material de apoyo para transmitir ideas.
- Sin embargo deben estar descritas en el texto y hacer referencia a ellas. Por ejemplo: En la Figura 1....
- Falta describir las pruebas (escenario, y resultados de la experimentación).
- Cuando se tienen resultados que se pueden comparar, se recomienda hacer uso de diagramas o tablas que permitan observar el resultado de los diversos casos y contrastas los resultados (en el tiempo por ejemplo).

Objetivo de la actividad

El objetivo de la tarea es que el alumno aplique los conocimientos y habilidades adquiridos en el tema de apuntadores a funciones y la distribución de tareas mediante el uso de hilos para la resolución de problemas utilizando el lenguaje ANSI C.

Descripción del problema

Existen diversas técnicas para generar una aproximación del valor del número irracional **Pi**. En este caso utilizaremos la serie de Gregory y Leibniz.

$$\pi = 4 \left(\sum_{n=1}^{\infty} \left(\frac{(-1)^{(n+1)}}{(2n-1)} \right) \right)$$
$$= \left(1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \dots \right)$$

Procedimiento

1. Codificar una solución secuencial (sin el uso de hilos) que calcule el valor de Pi, su solución debe basarse en la serie de Gregory y Leibniz para calcular los primeros diez dígitos decimales de Pi. Para esto, utilice los primeros tomando los primeros 50,000,000 términos de la serie.
2. Utilice las funciones definidas en la librería **time.h** (consulte diapositivas del curso) para medir el tiempo (en milisegundos) que requiere el cálculo del valor de **Pi**. Registre el tiempo.
3. Parametrice la solución que se implementó en el paso 1.
4. Utilice hilos para repartir el trabajo de calcular el valor de **Pi**. Pruebe su solución con los siguientes casos: 2 hilos, 4 hilos, 8 hilos y 16 hilos.
5. Tomar el tiempo en milisegundos que toma el programa para calcular el valor de **Pi** en cada uno de los casos mencionados en el paso 4.
6. Registre los tiempos registrados para cada caso en la siguiente tabla:

No. de Hilos	Tiempo (milisegundos)
1	
2	
4	
8	
16	

Descripción de la entrada

El usuario eberá indicar al programa cuantos hilos quiere utilizar para el calcular el valor de **Pi**.

Descripción de la salida

En un renglón imprimirá el valor calculado de **Pi**, con exactamente 10 dígitos decimales. En el siguiente renglón mostrará el número de milisegundos que se requirió para realizar el cálculo.

Ejemplo de ejecución:

```
Hilos? 4
Pi: 3.1415926535
Tiempo: 24487 ms
```

SOLUCIÓN DEL ALUMNO, PRUEBAS Y CONCLUSIONES

Código fuente de la versión secuencial (sin el uso de hilos)

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

int main(){
    setvbuf(stderr, NULL, _IONBF, 0);
    setvbuf(stdout, NULL, _IONBF, 0);
    //Declaración de variables
    float Pi=0, inicio, fin;
    unsigned long long i;

    inicio=clock(); //Captura el tiempo de inicio

    i=500000000000;
    while(i){
        if((i&1)==0) //pregunta si es par
            Pi-=(float)4/(2*i-1);
        else
            Pi+=(float)4/(2*i-1);
        i--;
    }

    fin=clock(); //Captura el tiempo de finalizado
    printf("\nPi = %.10f", Pi); //imprime pi
    printf("\nTiempo de ejecución: %f segundos",
        (fin-inicio)/(float)CLOCKS_PER_SEC); //imprime el tiempo
}
```

}Código fuente de la versión paralelizada

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

#define NHILOS 2
typedef struct{
    long long p;
    float suma;
    int nhilos;
}argumento;
DWORD WINAPI SerieGregoryLeibniz(void*);
float SumaDeHilos(argumento array[]);
int main(){
    setvbuf(stderr, NULL, _IONBF, 0);
    setvbuf(stdout, NULL, _IONBF, 0);

    int nhilos=1;
```

```

float inicio, fin;
long long repeticiones;
repeticiones=50000000000;

inicio=clock();
HANDLE *hilos;
hilos=(HANDLE*)malloc(sizeof(HANDLE)*nhilos);
argumento *array;
array=(argumento*)malloc(sizeof(argumento)*nhilos);
for(int i=0; i<nhilos; i++){
    array[i].p=repeticiones-i;
    array[i].suma=0;
    array[i].nhilos=nhilos;
    hilos[i]=CreateThread(NULL, 0,SerieGregoryLeibniz,
                          (void *)&(array[i]),0,NULL);
}
for(int i=0; i<nhilos; i++)
    WaitForSingleObject(hilos[i], INFINITE);
fin=clock();
printf("\nPi = %0.10f", SumaDeHilos(array));
printf("\nTiempo de ejecución: %f segundos",
        (fin-inicio)/(float)CLOCKS_PER_SEC);
}
DWORD WINAPI SerieGregoryLeibniz(void *arg){
    argumento *p=arg;
    while(p->p>0){
        if(((p->p)&1)==0)
            (p->suma)-=(float)4/(2*(p->p)-1);
        else
            (p->suma)+=(float)4/(2*(p->p)-1);
        p->p=p->nhilos;
    }
    return 0;
}
float SumaDeHilos(argumento array[]){
    float suma=0;
    for(int i=0; i<array->nhilos; i++){
        suma+=array[i].suma;
    }
    return suma;
}

```

Ejecución

Hilos	Sin hilos
Pi:	3.1415927410
Tiempo:	814 144 ms

Hilos	2
Pi:	3.1415901184
Tiempo:	449 664 ms

Hilos	4
Pi:	3.1415758133
Tiempo:	1 032 969 ms

Hilos	8
Pi:	3.1415944099
Tiempo:	636 455 ms

Hilos	16
Pi:	3.1415891647
Tiempo:	519 038 ms

Conclusiones:

Esta tarea me ayudó mucho a comprender el funcionamiento de hilos, ya que habían quedado muchas dudas del tema. Ahora tengo una mejor idea de lo que es trabajar con hilos y su funcionalidad. A la par con esto, tuve que hacer frente a ciertos problemas (como el hecho que se me llenara el stack con una función recursiva) que me han ayudado a entender de mejor manera el manejo de la memoria y conocer el tamaño y características de algunos tipos de datos que casi no utilizaba (como unsigned, long y double).

Me quedan aún varias dudas ¿Por qué el valor de la sumatoria cambia dependiendo de cuantos hilos se utilicen, a pesar de que al imprimir cada uno de los términos que se van añadiendo aparentemente son iguales? ¿Porque los tiempos de ejecución varían tanto? Como se observa en las tablas con 2 hilos tuvo el menor tiempo de ejecución y con 4 fue cuando más tardó. Espero que estas dudas puedan aclararse durante el transcurso del curso.