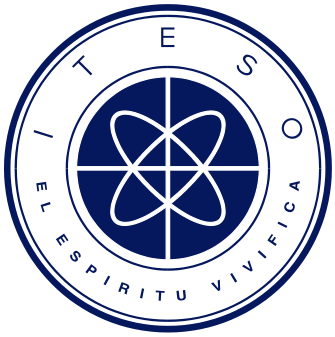
INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES DE OCCIDENTE

Departamento de Electrónica, Sistemas e Informática

INGENIERÍA EN SISTEMAS COMPUTACIONALES



PROGRAMACIÓN CON MEMORIA DINÁMICA

TAREA 2. APUNTADORES A FUNCIONES

Autor: Rodríguez Murguía Francisco Isaías

Presentación: 5 pts.
Funcionalidad: 60 pts.
Pruebas: 20 pts.

11 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).

Instrucciónes para entrega de tarea

Esta tarea, como el resto, es ***IMPRESCINDIBLE*** entregar los entregables de esta actividad de la siguiente manera:

- **Reporte:** vía *moodle* en **un archivo PDF**.
- **Código:** vía su repositorio **Github**.

La evaluación de la tarea comprende:

- 10% para la presentación
- 60% para la funcionalidad
- 30% para las pruebas

Es necesario responder el apartado de conclusiones, pero no se trata de llenarlo con paja. Si no se aprendió nada al hacer la práctica, es preferible escribir eso. Si el apartado queda vacío, se restarán puntos al porcentaje de presentación.

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 suo 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.

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

$$= \left(1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \cdots\right)$$

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 50,000’000,000 términos de la seria.

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 implemento 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	183000
2	231000
4	274000
8	149000
16	106000

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>
#include<string.h>
double p;

void calcularPi();

int main(){
	setbuf(stdout, NULL);
	fflush(stdout);
	clock_t init = clock();
	int timer;
	calcularPi();
	clock_t end = clock();
	timer= ((end-init)/CLOCKS_PER_SEC)*1000;
	printf("Tiempo %d ms", timer);
	return 0;
}

void calcularPi(){
	long long int i;
	for (i=1;i<50000000000;i++){
		p+=(i &1 ? 1.0 : -1.0 )/(2*i-1);
	}
	p=p*4;
	printf("Pi= %0.10lf\n",p);
}
```

Código fuente de la versión paralelizada

<<Copie y pegue su código fuente aquí.>>

```
#include<stdio.h>
#include<stdlib.h>
#include<time.h>
#include<string.h>
#include<windows.h>
#define MAX 50000000000

typedef struct {
	long long int sup;
	long long int inf;
	double   res;
}lims;

double p=0;

DWORD WINAPI calcularPi(void *);

int main(void)
```

```
{
	setbuf(stdout, NULL);
	fflush(stdout);
	int time;
	int i;
	int num;
	long long int l;

	HANDLE array[16];
	lims lim[16];

	printf("Cuantos hilos necesitas?\n");
	scanf("%d", &num);
	l = MAX/num;

	clock_t init = clock();

	for(i=0;i<num;i++){
		lim[i].inf=l*i+1;
		lim[i].sup=l*(i+1);
		lim[i].res=0;
		array[i]=CreateThread(NULL,0,calcularPi,(void *)&lim[i],0,NULL);
	}

	for(i=0;i<num;i++){
			WaitForSingleObject(array[i],INFINITE);
		}


	for(i=0;i<num;i++){
			p=p+lim[i].res;
			}

	clock_t end = clock();
	printf("Pi : %0.10lf\n",p);
	time = ((end-init)/CLOCKS_PER_SEC)*1000;
	printf("El tiempo de calculo fueron %d ms", time);

	return 0;
}

DWORD WINAPI  calcularPi(void * param){
	lims *pLim= (lims*)param;
	long long int i;

	for (i=pLim->inf; i<pLim->sup; i++){
		pLim->res+=((i-1) &1 ? -1.0 : 1.0 )/(2*i-1);
	}
	pLim->res=(pLim->res)*4;
	return 0;
}
```

Ejecución

```
Pi= 3.1415926536
Tiempo 183000 ms
--------------------------------
Process exited after 183.2 seconds with return value 0
Presione una tecla para continuar . . .
```

Secuencial (1 solo hilo)

```
Cuantos hilos necesitas?
2
Pi: 3.1415926537
El tiempo de calculo fueron 231000 ms
--------------------------------
Process exited after 233.1 seconds with return value 0
Presione una tecla para continuar . . .
```

```
Cuantos hilos necesitas?
4
Pi : 3.1415926539
El tiempo de calculo fueron 274000 ms
--------------------------------
Process exited after 276.8 seconds with return value 0
Presione una tecla para continuar . . .
```

```
Cuantos hilos necesitas?
8
Pi : 3.1415926544
El tiempo de calculo fueron 149000 ms
--------------------------------
Process exited after 152 seconds with return value 0
Presione una tecla para continuar . . .
```

```
Cuantos hilos necesitas?
16
Pi : 3.1415926557
El tiempo de calculo fueron 106000 ms
--------------------------------
Process exited after 108.7 seconds with return value 0
Presione una tecla para continuar . . .
```

Conclusiones (obligatorio):

- ✓ Lo que aprendí con esta práctica. Lo que ya sabía.

A usar mejor los hilos y a hacer sumatorias en código. Ya sabia usar la sintaxis de los hilos, así como el uso de apuntadores.

- ✓ Lo que me costó trabajo y como lo solucioné.

Me costo trabajo usar los hilos ya que no los entendí tan bien como debería, pero cuando vi los códigos de ejemplo en Moodle ya entendí mejor.

- ✓ Lo que no pude solucionar.

No hubo nada que no pudiera solucionar.