

Laboratorio 06

Instrucciones.

Esta actividad se realizará individualmente. Al finalizar los períodos de laboratorio o clase, deberá dejar constancia de sus avances en Canvas, según indique su catedrático. Al finalizar la actividad, adjuntar los archivos .pdf y .cpp para solucionar los ejercicios:

- Realice el diagrama de flujo del algoritmo generado.
- Desarrolle el programa solución en C++.
- Incluir video corto con narración de funcionamiento del programa.

Link Carpeta con Videos: <https://drive.google.com/drive/folders/1YH-LriCTajOqanbob7aa-2TWI5ORq8Ex?usp=sharing>

Ejercicio 01

Desarrolle un programa que utilice Pthreads de C++, que permita calcular/determinar los números primos entre 0 y un valor numérico máximo, el cual debe ser ingresado por el usuario (por teclado). El usuario también deberá ingresar la cantidad de hilos a utilizar en el programa. Cada hilo verificará n valores, donde n es:

$$n = \frac{\text{valor máximo ingresado por teclado}}{\text{cantidad de hilos ingresados por teclado}}$$

Se debe calcular e imprimir el valor de la suma total de los números primos encontrados, en la rutina principal, debe implementar paso de parámetros por medio de estructuras.

Si el resultado de n no es un valor exacto (entero), el último hilo deberá realizar la verificación de los valores restantes. En el siguiente ejemplo se verifica cuáles son los números primos entre 0-25, y se utilizan 2 hilos para realizar la comprobación:

```
-----
Ingrese el valor máximo: 25
Ingrese el numero de threads para realizar el cálculo: 2

Cantidad valores a evaluar por thread: 12 y residuo: 1
Buscando primos entre 0 y 25...
-----
```

Como $n = 25 / 2 = 12$, en principio, cada hilo debería verificar 12 valores cada uno:

- HILO 0: los valores entre 0-11
- HILO 1: los valores entre 12 y 24. Debido a que 24 es menor al valor ingresado, el último hilo debe comprobar los valores restantes hasta alcanzar el valor máximo (es decir, también verificará si el valor 25 es primo).

Cada pthread comprueba los números primos en el rango de valores correspondientes y calcula la suma de los primos en ese hilo:

```
-----  
THREAD 0  
Lim inf: 0  
Lim sup (no incluido): 12  
2  
3  
5  
7  
11  
Suma hilo: 28  
-----  
THREAD 1  
Lim inf: 12  
Lim sup (no incluido): 26  
13  
17  
19  
23  
Suma hilo: 72  
Suma TOTAL: 100
```

Ejercicio 02

Desarrolle un programa que utilice Pthreads de C++, que realice el cálculo de valor de convergencia para la siguiente serie geométrica:

$$\sum_{n=0}^{\infty} \frac{3}{2^n}$$

a.

Condiciones:

- El valor máximo de “n” a evaluar en la serie debe ser ingresado por teclado.
- Se utilizará 1 pthread para evaluar cada valor de n en la serie. Es decir, la cantidad de hilos utilizados para evaluar la serie debe ser igual a n.
- La sumatoria e impresión del resultado se debe efectuar en la rutina principal. Debe implementar paso de parámetros por medio de pthread_join.

Ejercicio 03 (20 pts.)

Utilice el ejemplo de hello world con Pthreads y realice un ciclo for para crear hilos y un ciclo for para hacer su respectivo join. Cada hilo debe de imprimir "Hello world thread No. X" colocando el número del hilo en lugar de la "X".

Basándose en el mismo ejemplo, realice un ciclo for que cree el hilo y haga su respectivo join dentro del mismo ciclo. Cada hilo debe de imprimir "Hello world thread No. X" colocando el número del hilo en lugar de la "X".

Responder las siguientes preguntas:

1. ¿Cuál fue la diferencia entre la impresión del primer programa y el segundo?
En el primer programa, todos los hilos se crean primero y luego se hace el join, lo que permite que los hilos trabajen en paralelo. En el segundo programa, cada hilo se crea y espera a que termine inmediatamente, lo que significa que los hilos se ejecutan uno tras otro.
2. ¿A qué se debió el comportamiento descrito en la respuesta anterior?
 - En el primer caso, los hilos se ejecutan en paralelo porque el `pthread_join` se ejecuta después de la creación de todos los hilos.
 - En el segundo caso, se espera a que cada hilo termine antes de crear el siguiente, lo que provoca una ejecución secuencial.