

Laboratorio 06

Ángel de Jesús Mérida Jiménez 23661
Roberto Samuel Nájera Marroquín 23781

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:

- Desarrolle el programa solución en C++.
- Incluir video corto con narración de funcionamiento del programa.

Ejercicio 02 (20 pts.)

- a. Investigue qué es, cómo usar y para qué sirve `pthread_join()`. Deje evidencia de su investigación creando una infografía.

https://www.canva.com/design/DAGQ7Qktdl0/bHALfK4VsmmzCB3oHqVp-A/view?utm_content=DAGQ7Qktdl0&utm_campaign=designshare&utm_medium=link&utm_source=editor

Utilice el ejemplo de hello world con Pthreads y:

- b. Modifique el programa para realizar un ciclo for para crear hilos y un ciclo for (separado) 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".
- c. 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?

R// En el primer programa, todos los hilos se crean primero, y luego el programa espera a que todos terminen. Es posible que los mensajes de los hilos no sigan el mismo orden en que fueron creados, ya que los hilos se ejecutan concurrentemente. En el segundo programa, cada hilo se crea y espera su finalización antes de crear el siguiente. Esto garantiza que los mensajes aparezcan en el orden en que los hilos fueron creados (del 0 al 9).

2. ¿A qué se debió el comportamiento descrito en la respuesta anterior?

R// En el primer programa, los hilos se ejecutan en paralelo o concurrencia debido a que todos se crean antes de hacer el "join". Como no se espera a que un hilo termine antes de crear el siguiente, todos los hilos pueden empezar su ejecución casi al mismo tiempo y correr en paralelo (dependiendo de la implementación del sistema operativo). En la versión del segundo programa, el "pthread_join()" inmediatamente después de "pthread_create()" hace que el hilo principal espere a que el hilo actual termine antes de crear el siguiente. Esto provoca que los hilos se ejecuten secuencialmente uno por uno, en lugar de en paralelo. Por lo tanto, el comportamiento es más similar a una ejecución secuencial que a una concurrente.

Enlace al vídeo explicativo:

<https://youtu.be/1uNj-eG4LYs?feature=shared>