

Taller de Paralelismo en Kotlin

Este taller consta de 20 preguntas sobre paralelismo en Kotlin, incluyendo preguntas teóricas y un ejercicio práctico. El objetivo es que los estudiantes comprendan los conceptos fundamentales y puedan aplicar paralelismo en un programa en Kotlin.

Ejercicio de Programación en Kotlin

Este código en Kotlin calcula la suma de una lista de números enteros. La idea es que lo modifiquen para realizar el cálculo en paralelo.

```
fun sumList(numbers: List<Int>): Int {  
    var sum = 0  
    for (number in numbers) {  
        sum += number  
    }  
    return sum  
}  
  
fun main() {  
    val numbers = List(1_000_000) { (1..10).random() }  
    println("Suma total: ${sumList(numbers)}")  
}
```

Implementa el paralelismo en el código usando corrutinas o la biblioteca `ForkJoinPool` de Kotlin, y compara los tiempos de ejecución con el código original.

Preguntas del Taller

1. ¿Qué es paralelismo y cómo se diferencia de la concurrencia?
2. ¿Cuál es el propósito principal del paralelismo en sistemas informáticos?

3. Explica qué es un núcleo de procesamiento en el contexto de paralelismo.
4. ¿Qué significa que una operación sea “thread-safe” en programación paralela?
5. Define el concepto de “balanceo de carga” y su importancia en sistemas paralelos.
6. ¿Qué es una “condición de carrera” y cómo afecta a la programación paralela?
7. ¿Cuál es la diferencia entre memoria compartida y paso de mensajes en paralelismo?
8. Define el término “overhead” en el contexto de paralelismo.
9. ¿Qué es un “deadlock” y cómo puede ocurrir en sistemas paralelos?
10. ¿Cuál es el rol de un “scheduler” en la administración de tareas en paralelismo?
11. Describe cómo el paralelismo puede mejorar el rendimiento de un programa. Da un ejemplo general.
12. Explica qué es una “condición de carrera” y proporciona un ejemplo de cómo podría ocurrir.
13. ¿Por qué es importante la sincronización en programación paralela? Menciona alguna técnica de sincronización.
14. Describe los posibles efectos del “overhead” en la ejecución de tareas en paralelo.

15. Explica cómo se pueden evitar los deadlocks en un sistema paralelo.
16. ¿Cuáles son las ventajas y desventajas de utilizar memoria compartida en lugar de paso de mensajes en paralelismo?
17. ¿Qué estrategias existen para gestionar la asignación de tareas en un sistema con múltiples núcleos?
18. Describe un caso en el que el balanceo de carga sea esencial en un sistema paralelo.
19. Explica cómo la latencia y el ancho de banda afectan el rendimiento en paralelismo.
20. ¿Cuáles son los desafíos principales en la implementación de sistemas paralelos y cómo se pueden mitigar?