

COMPUTACIÓN CONCURRENTES

TAREA 1

Prof. Manuel Alcántara Juárez
manuelalcantara52@ciencias.unam.mx

Alejandro Tonatiuh Valderrama Silva José de Jesús Barajas Figueroa
at.valderrama@ciencias.unam.mx jebarfig21@ciencias.unam.mx

Luis Fernando Yang Fong Baeza Ricchy Alain Pérez Chevanier
fernandofong@ciencias.unam.mx alain.chevanier@ciencias.unam.mx

Fecha de Entrega: 17 de Octubre de 2021 a las 23:59

1. Ejercicios

1. Considera el siguiente código que representa un contador:

```
public class Counter implements Runnable {  
    public static final int ROUNDS = 5;  
    private int counter = 0;  
  
    public void run() {  
        for(int i=0; i < Counter.ROUNDS; i++){  
            counter++;  
        }  
    }  
}
```

Suponiendo que dos hilos se ejecutan concurrentemente sobre el mismo objeto de tipo `Counter`. Muestre la historia de una ejecución en la cual el resultado final del contador sea:

- a) 10 b) 8 c) 5 d) 1 e) 0

En caso de que no existiera dicha ejecución, demuéstralo formalmente.

HINT: Por contradicción. Supón que existe, y ve en reversa para ver como pudo suceder.

2. Considera el siguiente código de un banco encargado de hacer una transferencia:

```
public boolean transferir(double cantidad, Cuenta destino) {  
    if (this.balance < cantidad) return false;  
    destino.balance += cantidad;  
    this.balance -= cantidad;  
    return true;  
}
```

- a) Demuestra que existe una condición de carrera.
- b) ¿Tiene una carrera por los datos?
3. Considera un problema computacional que puede resolverse secuencialmente en $n^3 * \log_{10}(n)$ con una unidad de tiempo de 2 nanosegundo ($10^{-9}s$). De esta manera, si $n = 1000$, el cómputo tomaría $1000^3 * \log_{10}(1000) \times 2ns = 6s$. Supón que has creado un algoritmo concurrente que trabaja de manera completamente eficiente, es decir, el cómputo total tarda $\frac{n^3 * \log_{10}(n)}{p}$ (p es el número de hilos). ‘¿Qué tan grande es la entrada que se puede manejar en:

- a) 1 segundo b) 1 minuto c) 1 semana

Si tenemos los siguientes hilos disponibles:

- a) 8 b) 1000 c) 1,000,000

4. Da un ejemplo de un programa que:
- a) Tenga una condición de carrera y carrera por los datos.
- b) Tenga una condición de carrera pero no carrera por los datos.
- c) No tenga condición de carrera pero si carrera por los datos.
- d) No tenga ni condición de carrera ni carrera por los datos.
5. Calcula el speedup que tendría un programa con 100 procesadores si al medirlo en forma secuencial se tarda 188 segundos y con dos procesadores tarda 104 segundos.
6. Usa la ley de Amdahl para resolver las siguientes preguntas
- a) Supón que un programa tiene un método M que no puede ser paralelizado y que este método cuenta el 45 % del tiempo de la ejecución del programa. ¿Cuál es el límite de speedup general que se puede lograr ejecutando el programa en una máquina con n procesos?
- b) Supón que el método M cuenta el 35 % del tiempo de la ejecución del programa. Sea s_n el speedup con n procesos. Tu jefe te dice que debes duplicar este speedup: la versión nueva del programa debe tener un speedup $s'_n \geq 2 * s_n$. Tu buscas a un programador para reemplazar M con una version mejorada, k veces más rápida. ¿Qué valor de k es requerido?
- c) Supón que el método M se puede acelerar tres veces. ¿Qué fracción de todo el tiempo de ejecución debe contar M para que se pueda doblar el speedup del programa?
7. Tu amigo Juanito acaba de paralizar un código para su trabajo con la finalidad de que se ejecute más rápido. Por el momento, solo cuenta con una computadora con dos núcleos, la cual produjo un speedup S_2 . Juanito quiere saber cuántos nucleos adicionales tendría que comprar para alcanzar el mejor desempeño sin desperdiciar recursos.
- Ayuda a tu amigo y utiliza la Ley de Amdahl para derivarle una fórmula S_n (speedup con n procesadores) en términos de n y S_2 .

- Supongamos que vuelves a encontrar a Juanito y te dice logró optimizar su programa 10x haciendo únicamente el 35 % de su código paralelo. ¿Lo que dice es verdad? Justifica tu respuesta.
8. Propón un algoritmo de exclusión mutua para 3 hilos basado en turnos (round robin).
 - a) Demuestra que cumple con la propiedad de exclusión.
 - b) ¿Es libre de deadlock?
 - c) ¿Es libre de hambruna?
 9. Menciona al menos 2 ventajas de usar hilos heredando de la clase **Thread** y dos ventajas de implementar la interfaz **Runnable**. ¿En qué casos usarías qué forma?
 10. Tu eres uno de los P prisioneros recién arrestados durante una manifestación en pro de las ciencias de la computación. El guardia, un computólogo transtornado , hace el siguiente anuncio:

Escúchenme bien porque no lo repetiré. Hoy por la noche podrán juntarse y planear una estrategia, pero después de hoy, cada uno estará en celdas aisladas y no tendrán comunicación con nadie. He puesto un cuarto con un apagador, el cual puede estar en un estado de prendido o apagado, sin embargo, no está conectado a nada.

De vez en cuando, seleccionaré a un prisionero aleatoriamente para entrar al cuarto con el apagador. Ese prisionero podría cambiar el estado del apagador, de prendido a apagado o viceversa, o podría dejarlo tal y como lo encontró. Me aseguraré que para cualquier P , eventualmente cada uno de ustedes visitará el cuarto al menos N veces pero sin ningún orden en particular. Cuando ustedes lo deseen, podrían declarar la siguiente frase: “todos hemos visitado el cuarto con el apagador al menos una vez”. Si su afirmación es correcta, ¡los liberaré!, de lo contrario serán comida de cocodrilos. Escojan sabiamente y que comience el juego.

- a) Desarrolla una estrategia ganadora para cuando sabes que el estado inicial del apagador comienza en apagado.
- b) Desarrolla una estrategia ganadora para cuando no sabes el estado inicial del apagador

Hint: No todos los prisioneros necesitan hacer lo mismo.