

Ejercicios Tema 1: Conceptos fundamentales



Programación Concurrente

Francisco Javier Pérez Martínez

1 de octubre de 2021

Ejercicios

- Usando las condiciones de Bernstein, construye una tabla como la anterior para el siguiente código:

```
S1 cuad := x * x;
S2 m1 := a * cuad;
S3 m2 := b * x;
S4 z := m1 + m2;
S5 y := z + c;
```

Para que se puedan ejecutar concurrentemente dos conjuntos de instrucciones S_i y S_j se debe cumplir que ninguno escriba lo que el otro lee o escribe.

- $L(S_i) \cap E(S_j) = \emptyset$
- $E(S_i) \cap L(S_j) = \emptyset$
- $E(S_i) \cap E(S_j) = \emptyset$

Aplicando estas reglas, obtenemos la siguiente tabla:

	S1	S2	S3	S4	S5
S1	-	N	S	S	S
S2	-	-	S	N	S
S3	-	-	-	N	S
S4	-	-	-	-	N
S5	-	-	-	-	-

- Supongamos que una variable se actualiza con dos funciones que se ejecutan concurrentemente, ¿qué valor tendría x al acabar el programa?

<pre>1 x = 0 2 3 def incrementa_5_A(): 4 i = 0 5 while i < 5: 6 i = i + 1 7 x = x + 1</pre>	<pre>def incrementa_5_B(): i = 0 while i < 5: i = i + 1 x = x + 1</pre>
--	--

La variable x tendría un valor entre 5 y 10, un mínimo de 5 debido a que el bucle se itera 5 veces y hasta 10 porque al ejecutarse concurrentemente y compartir una variable común los 2 procesos podría darse el caso que a la hora de leer uno de los dos lea el mismo dato y éste se sobrescriba.

a) Plantea una traza que haría que x valiese 5 al final de la ejecución. ¿Qué rango de valores crees que podría acabar teniendo x ?

- **XiA** = Valor de x inicialmente en la función A.
- **XiB** = Valor de x inicialmente en la función B.
- **XfA** = Valor de x finalmente en la función A.
- **XfB** = Valor de x finalmente en la función B.

it	XiA	XiB	XfA	XfB
0	0	0	1	1
1	1	1	2	2
2	2	2	3	3
3	3	3	4	4
4	4	4	5	5

Ambas funciones A y B acceden al mismo valor de X. En la 1^o iteración con $x = 0$; la función A le suma uno ($x := x + 1$) y la función B realiza lo mismo, pero al valor de x inicial (0), por lo que ambas funciones ahora tienen $x = 1$. En la siguiente iteración ocurriría igual por lo que hasta finalizar el bucle, x se iría sobrescribiendo y su valor final sería de 5.

b) ¿De qué problemas de los nombrados en esta sesión adolece el programa?

Problemas de *exclusión mutua* y *condición de sincronización*.

Cuestiones breves

1. **¿Cuál es la ventaja de la concurrencia en los sistemas monoprocesador?**

Utilizando sistemas monoprocesador optimizamos el uso de los recursos como p.e. el aprovechamiento de la CPU.

2. **¿Cuáles son las diferencias entre programación concurrente, paralela y distribuida?**

- La *programación concurrente* es un conjunto de instrucciones que se pueden ejecutar al mismo tiempo.
- La *programación paralela* es un tipo de programación concurrente para sistemas multiprocesador.
- La *programación distribuida* es un tipo de programación paralela para sistemas distribuidos.

3. **¿Cuáles son los dos problemas principales inherentes a la programación concurrente?**

Los problemas de exclusión mutua y de condición de sincronización.

4. **¿Qué es una sección crítica?**

La región donde se encuentra un recurso que comparten dos o más procesos.

5. **¿Cuáles son las características de un programa concurrente?**

Las características de un programa concurrente son su orden parcial y su indeterminismo.

6. **¿Qué se entiende por un programa concurrente correcto?**

Un programa que cumple con las siguientes reglas:

- Propiedad de *seguridad*: controla exclusión mutua, condiciones de sincronización e interbloqueo.
- Propiedades de *viveza*: controla el interbloqueo activo y la inanición.