

Programación concurrente



Universitat d'Alacant
Universidad de Alicante



Ejercicios tema 1, Conceptos fundamentales

Francisco Joaquín Murcia Gómez

25 de febrero de 2022

Índice

1. Ejercicio 1	3
1.1. Usando las condiciones de Bernstein, construye una tabla como la anterior para el siguiente código:	3
2. Ejercicio 2	4
2.1. Ejemplo: supongamos que una variable se actualiza con dos funciones que se ejecutan concurrentemente, ¿qué valor tendría x al acabar el programa?	4
2.2. Plantea una traza que haría que x valiese 5 al final de la ejecución. ¿Qué rango de valores crees que podría acabar teniendo x ?	4
2.3. ¿De qué problemas de los nombrados en esta sesión adolece el programa?	4
3. Cuestiones breves	5
3.1. ¿Cuál es la ventaja de la concurrencia en los sistemas monoprocesador?	5
3.2. ¿Cuáles son las diferencias entre programación concurrente, paralela y distribuida?	5
3.3. ¿Cuáles son los dos problemas principales inherentes a la programación concurrente?	5
3.4. ¿Qué es una sección crítica?	5
3.5. ¿Cuáles son las características de un programa concurrente?	5
3.6. ¿Qué se entiende por un programa concurrente correcto?	5

1. Ejercicio 1

- 1.1. 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.

1. $L(S_i) \cap E(S_j) = \emptyset$
2. $E(S_i) \cap L(S_j) = \emptyset$
3. $E(S_i) \cap E(S_j) = \emptyset$

Si aplicamos esto al código obtenemos la siguiente tabla:

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

2. Ejercicio 2

2.1. Ejemplo: supongamos que una variable se actualiza con dos funciones que se ejecutan concurrentemente, ¿qué valor tendría x al acabar el programa?

```
x = 0
def incrementa_5_A():
    i = 0
    while i < 5:
        i = i + 1
        x = x + 1

def incrementa_5_B():
    i = 0
    while i < 5:
        i = i + 1
        x = x + 1
```

X podría valer un rango de 5 a 10 ya que mínimo, el bucle se itera 5 veces por lo cual se sumaría 5 veces, al compartir la variable "X" puede ser que al ejecutarse concurrentemente ambas funciones accedan al mismo dato, así podría dar de 5 a 10.

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

- I: iteración.
- Xa0: variable x que lee la función A.
- Xb0: variable x que lee la función B.
- Xa1: variable x que escribe la función A.
- Xb1: variable x que escribe la función B.

I	Xa0	Xb0	Xa1	Xb1
1	0	0	1	1
2	1	1	2	2
3	2	2	3	3
4	3	3	4	4
5	4	4	5	5

En este caso la función A y B siempre están accediendo al mismo dato de X, de tal modo que en la primera iteración la X inicial vale 0, la función A le suma 1 y al mismo tiempo la función B también le suma 1 pero a la X inicial (X=0), en la siguiente iteración con X=1 pasaría lo mismo, A y B estarían machacando la variable X, de tal modo que al acabar el bucle X valdrá 5.

2.3. ¿De qué problemas de los nombrados en esta sesión adolece el programa?

- Exclusión mutua.
- Condición de sincronización.

3. Cuestiones breves

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

La velocidad de ejecución ya que hay un aprovechamiento del procesador.

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

La concurrencia son instrucciones que se pueden ejecutar al mismo tiempo, la programación paralela es lo mismo pero usa sistemas multiprocesador y la distribuida es como la paralela pero en un sistema distribuido (en una red).

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

Como ya hemos hablado antes son los de exclusión mutua y los problemas de sincronización.

3.4. ¿Qué es una sección crítica?

Es una zona donde solo puede trabajar un proceso a la vez, ya que es compartido por mas de un proceso.

3.5. ¿Cuáles son las características de un programa concurrente?

Son el orden parcial y el indeterminismo.

3.6. ¿Qué se entiende por un programa concurrente correcto?

Es un programa que cumple unas condiciones de seguridad (exclusión mutua, condición de sincronización, interbloqueo) y unas propiedades de vivacidad(interbloqueo activo, inanición).