

Práctica 1 – Variables compartidas

1. Para el siguiente programa concurrente suponga que todas las variables están inicializadas en 0 antes de empezar.

Indique cual/es de las siguientes opciones son verdaderas:

- a) En algún caso el valor de x al terminar el programa es 56.
- b) En algún caso el valor de x al terminar el programa es 22.
- c) En algún caso el valor de x al terminar el programa es 23.
- d) X puede obtener un valor incorrecto por interferencias.

| | | |
|--|--|---------------------------------------|
| P1:: If (x = 0) then y:= 4*2; x:= y + 2; | P2:: If (x > 0) then x:= x + 1; | P3:: x:= (x*3) + (x*2) + 1; |
|--|--|---------------------------------------|

2. Dado un numero N verifique cuantas veces aparece ese número en un arreglo de longitud M. Realice el algoritmo en forma concurrente utilizando <> y <await B; S>. Escriba las condiciones que considere necesarias.

3. En base a lo visto en la clase 3 de teoría.

- a) Indicar si el siguiente código funciona para resolver el problema de Productor/Consumidor con un buffer de tamaño N. En caso de no funcionar, debe hacer las modificaciones necesarias.

| | |
|--|---|
| int cant = 0; int pri_ocupada = 0; int pri_vacia = 0; int buffer[N]; | |
| Productor:: { while (true) { <i>produce elemento</i> <await (cant < N); cant++> buffer[pri_vacia] = <i>elemento</i> ; pri_vacia = (pri_vacia + 1) mod N; } } | Consumidor:: { while (true) { <await (cant > 0); cant-- > <i>elemento</i> = buffer[pri_ocupada]; pri_ocupada = (pri_ocupada + 1) mod N; <i>consume elemento</i> } } |

- b) Modificar el código para que funcione para C consumidores y P productores.