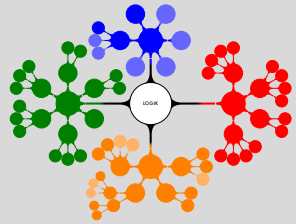


Tema 1

Resolución de la primera relación de problemas. SCD-GIIM

Sistemas Concurrentes y Distribuidos (SCD)

Asignatura *de GIIM (3er curso)*

Fecha 30 septiembre 2021

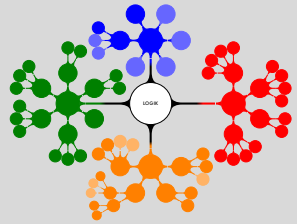
Manuel I. Capel
manuelcapel@ugr.es
Departamento de Lenguajes y Sistemas Informáticos
Universidad de Granada

P1

1. Los posibles valores de la variable son: **x = 2, 3 y 4**
   - Cada uno de los dos procesos `P1`, `P2` hace 2 lecturas:`L11, L12, L21, L22` y 2 escrituras
   - Cada proceso incrementa (+1) **x**, 2 veces partiendo de 0: el valor final de x $\not<$ 2
   - Se hacen 4 incrementos de **x** : el valor final de x $\not>$ 4

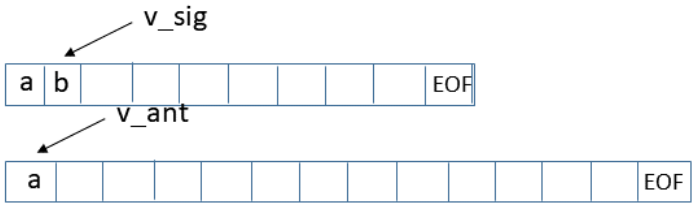
x	P1	P2	x	P1	P2	x	P1	P2
0	L11	-	0	L11	-	0	L11	-
0	-	L21	0	-	L21	1	E11	-
1	E11	-	1	E11	-	1	-	L21
1	-	E21	1	-	E21	2	-	E21
1	L12	-	1	L12	-	2	L12	-
1	-	L22	2	E12	-	3	E12	-
2	E12	-	2	-	L22	3	-	L22
2	-	E22	**3**	-	E22	**4**	-	E22

Manuel I. Capel
manuelcapel@ugr.es



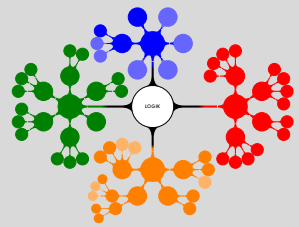
P2

- Los datos del primer archivo han de ser escritos en el segundo conservando el orden secuencial de aparición
- La escritura de un elemento procedente del primer archivo puede solaparse en el tiempo con la lectura del siguiente
- Hayq ue evitar una *condición de carrera* en el acceso a la variable compartida que contenga el último dato leído

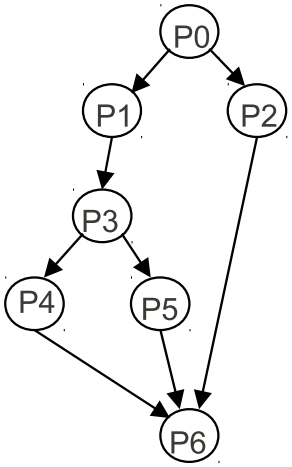



```
process Correcto ;
  var v_ant, v_sig : T ;
  begin
    v_sig := leer(f) ;
    while not fin(f) do begin
      v_ant := v_sig ;
      cobegin escribir(g,v_ant); v_sig := leer(f) ; coend
    end
  end
```


Manuel I. Capel
manuelcapel@ugr.es



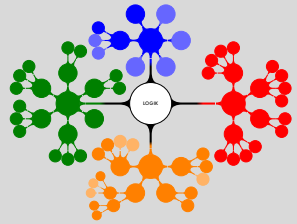
P3



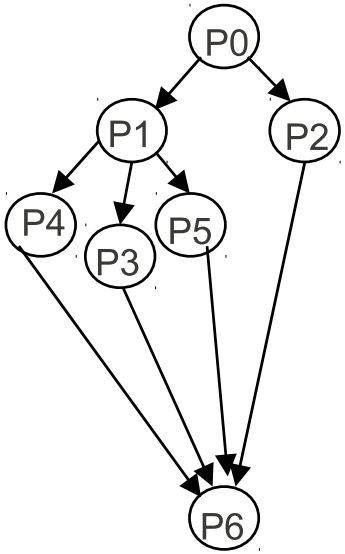

```
begin
  P0 ; fork P2 ;
  P1 ; P3 ; fork P5 ; P4 ;
  join P2 ; join P5 ;
  P6 ;
end
```

```
begin
  P0 ;
  cobegin
    begin
     P1 ; P3 ;
     cobegin P4 ; P5 ; coend
    end
    P2 ;
  coend
  P6 ;
end
```


Manuel I. Capel
manuelcapel@ugr.es



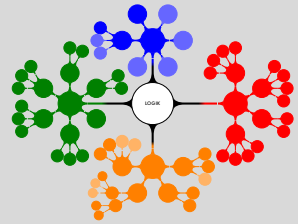
P3



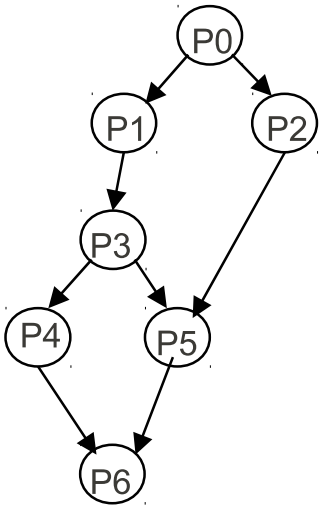

```
begin
  P0 ; fork P2 ;
  P1 ; fork P3 ;  fork P5 ;
  P4
  join P2 ; join P3 ; join P5 ;
  P6 ;
end
```

```
begin
  P0 ;
  cobegin
   begin
     P1 ;
     cobegin P3 ; P4 ; P5 ; coend
   end
   P2 ;
  coend
  P6 ;
end
```


Manuel I. Capel
manuelcapel@ugr.es



P3

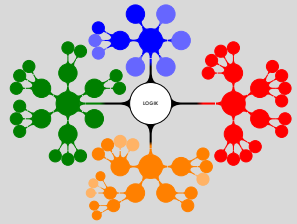



```
begin
  P0 ; fork P2 ;
  P1 ;
  P3 ; fork P4 ;
  join P2 ;
  P5;
  join P4 ;
  P6 ;
end
```

```
begin
  P0 ;
  cobegin
    begin P1 ; P3 ; end
    P2 ;
  coend
  cobegin P4 ; P5 ; coend
  P6 ;
end
```

```
begin
  P0 ;
  cobegin
    begin
      P1 ; P3 ; P4 ;
    end ;
    P2 ;
  coend
  P5 ; P6 ;
end
```


Manuel I. Capel
manuelcapel@ugr.es

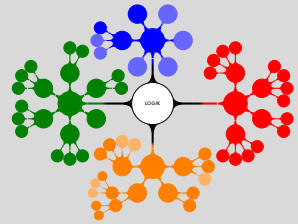


P5

- Suponemos una variable ficticia **OUT** que se crea como resultado de la instrucción `write(n)` **(2)** que contiene el valor impreso (éste pasa así a formar parte del estado)
- En el estado inicial se cumple `n == k`
- Sólo serán correctos los entrelazamientos de instrucciones atómicas del programa que sean compatibles con el estado final: `OUT + n == k +1`
- Los posibles entrelazamientos son: (a) 1,2,3, (b) 2,1,3 y (c) 2,3,1.

(a)			(b)			(c)		
inst.	n	OUT	inst.	n	OUT	inst.	n	OUT
-	k	-	-	k	-	-	k	-
n:=n+1	k+1	-	write(n)	k	k	write(n)	k	k
write(n)	k+1	k+1	n:= n+1	k+1	k	n:= 0	0	k
n:=0	0	k+1	n:=0	0	k	n:= n+1	1	k

Manuel I. Capel
manuelcapel@ugr.es



P6

1. Calculo del tiempo para 2 instancias concurrentes del procedimiento **Sort** n que se ejecutan en 1 solo procesador:

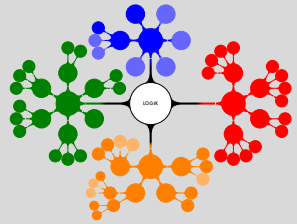
$$P1 = 2 \cdot T_s(n) + T_m(2n) = (n^2 - n) + 2n = n^2 + n$$

(tiempo de ejecución de la versión totalmente secuencial para ordenar 2n elementos: $S = 2 \cdot n^2 - n$

2. Cada instancia de **Sort** se ejecuta en un procesador distinto:

$$P2 = T_s(n) + T_m(2n) = (1/2 \cdot n^2 - 1/2 \cdot n) + 2n = 1/2 \cdot n^2 + 3/2 \cdot n$$

Manuel I. Capel
manuelcapel@ugr.es

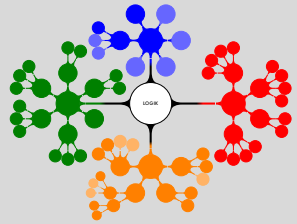


P7

- Se podría paralelizar calculando de forma independiente las : filas, columnas, ..., de la matriz resultado
- Utilizamos 3 procesos concurrentes **CalcularFila (i:1..3)**:

```
var a, b, c : array [1..3,1..3] of real ;
process CalcularFila[ i : 1..3 ] ;
   var j, k : integer ;
  begin
    for j := 1 to 3 do begin
     c[i,j] := 0 ;
     for k := 1 to 3 do
       c[i,j] := c[i,j] + a[i,k]*b[k,j] ;
    end
end
```


Manuel I. Capel
manuelcapel@ugr.es



P8

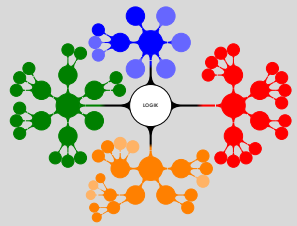
- Se utilizará un vector de valores lógicos
- Dicho vector ha de inicializarse de 1 sola vez antes de la siguiente iteración de los bucles
- solución: al terminar el proceso 9 se inicializará el vector

```
{ compartido entre todas las tareas }
var finalizado : array [1..9] of boolean := (false,..,
    false) ;
```

```
procedure EsperarPor( i : integer )
  begin
    while not finalizado[i] do  begin; end
end
```

```
procedure Acabar( i : integer )
  var j : integer ;
  begin    if i < 9 then
             finalizado[i] := true ;
           else for j := 1 to 9 do
              finalizado[j] := false ;
end
```


Manuel I. Capel
manuelcapel@ugr.es

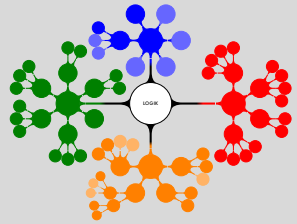


P10

Se resuelve aplicando directamente el axioma de asignación, basado en la sustitución textual de $\{P\}$ por $\{P\}^x_e$ en la precondición de los triples :

1. $\{i < 10\}\ i = 2 * i + 1\ \{\ i < 21\ \}$ puesto que: $\{i < 21\}^i_{2*i+1} \equiv \{2 * i + 1 < 21\} \equiv \{i < 10\}$
2. $\{i > 0\}\ i = i - 1; \{\ i > -1\}$
3. $\{i > j\}\ i = i + 1;\ \{i > j + 1\}\ j = j + 1\ \{\ i > j\}$
4. $\{F\}\ a = a + 7;\ \{\ V\ \}$
5. $\{V\}\ i = 3;\ \{i == 3\}\ j = 2 * i\ \{\ j == 6\}$
6. $\{V\}\ c = a + b; \{c == a + b\}\ c = c/2\ \{c == \frac{a+b}{2}\}$

Manuel I. Capel
manuelcapel@ugr.es

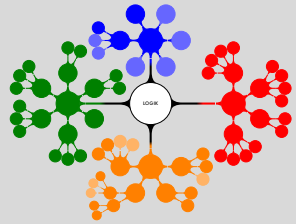


P11

$i, x, a \in \mathbb{Z}$

1. $\{i > 0\} i = i - 1; \{i + 1 > 0\} \Rightarrow \{i \geq 0\}$
2. $\{x \geq 7\}\ x = x + 3; \{x \geq 10\} \Rightarrow \{x \geq 9\}$
3. $\{i < 9\}\ i = 2 * i + 1; \{i < 19\} \Rightarrow \{i \leq 20\}$
4. $\{a > 0\}\ a = a - 7; \{a > -7\} \nRightarrow \{a > -6\}$

Manuel I. Capel
manuelcapel@ugr.es

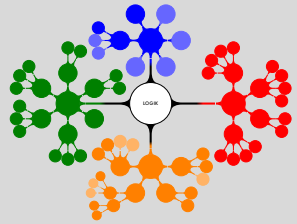


P12

El triple $\{P\}\ C\ \{Q\}$ es demostrable,

1. $\{P\}\ C\ \{Q \vee P\}$ también lo es por debilitamiento de la poscondición
2. $\{P \wedge D\}\ C\ \{Q\}$ también lo es por fortalecimiento de la precondición
3. $\{P \vee D\}\ C\ \{Q\}$ No lo es porque se debilita la precondición
4. $\{P\}\ C\ \{Q \vee D\}$ lo mismo que (1)
5. $\{P\}\ C\ \{Q \wedge P\}$ No lo es porque se fortalece la poscondición

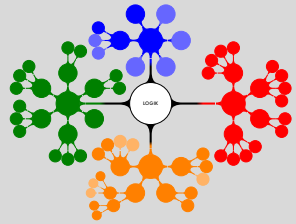
Manuel I. Capel
manuelcapel@ugr.es



P13

1. $\{P \wedge D\}\ C\ \{Q\}$
2. $\{P \vee D\}\ C\ \{Q\}$ No se puede demostrar porque se debilita la precondición
3. $\{P\}\ C\ \{Q \vee D\}$
4. $\{P\}\ C\ \{Q \vee P\}$

Manuel I. Capel
manuelcapel@ugr.es



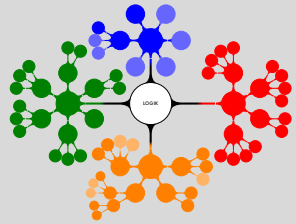
P14

`int x = 5, y = 2; cobegin < x = x + y >;< y = x * y > coend;,`

a. Considerando operaciones atómicas (con los símbolos <,>)
   1. $\{x == 5 \wedge y == 2\}\ < x = x + y >;< y = x * y >\ \{x == 7 \wedge y == 14\}$
   2. $\{x == 5 \wedge y == 2\}\ < y = x * y >;< x = x + y >\ \{x == 15 \wedge y == 10\}$

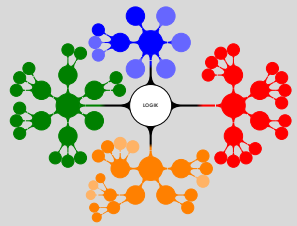
b. Sin considerarlas operaciones atómicas (quitando los símbolos <,>)
   1. Los valores de (a) y además $\{x == 7 \wedge y == 10\}$

Manuel I. Capel
manuelcapel@ugr.es



P15

Manuel I. Capel
manuelcapel@ugr.es



Regla no de interferencia de predicado $\{P\}$ con acción atómica $\{P \wedge pre(a)\} < a > \{P\}$ El triple

$\{x \geq 2\} < x = x - 2 >; \{x \geq 0\}$ interfiere con:

$\{x \geq 0\} < x = x + 3 > \{x \geq 3\}$	**Sí**	$\{x \geq 3\} < x = x - 2 > \{x \geq 1\} \not\Rightarrow \{x \geq 3\}$
$\{x \geq 0\} < x = x + 3 > \{x \geq 0\}$	**No**	$\{x \geq 2\} \wedge \{x \geq 0\} < x = x - 2 >; \{x \geq 0\}$
$\{x \geq 7\} < x = x + 3 > \{x \geq 10\}$	**Sí**	$\{x \geq 7\} < x = x - 2 > \Rightarrow \{x \geq 5\} \not\Rightarrow \{x \geq 7\}$
$\{y \geq 0\} < y = y + 3 > \{y \geq 3\}$	**No**	Las variables x e y son disjuntas
$\{x$ *es impar*$\} < y = x + 1 > \{y$ *es par*$\}$	**No**	$\{x \in \dot{2} + 1\} \wedge \{x \geq 2\} < x = x - 2 >;$ $\{x + 2\} \in \{\dot{2} + 1\} \Rightarrow \{x \in \dot{2} + 1\}$

P16

$$\{x == 0\}$$

$$-- \textit{ inits cobegin}$$

$$\{x == 0 \lor x == b \lor x == c \lor x == b + c\}$$

$$< x = x + a > \quad \|$$

$$\{x == a \lor x == a + b \lor x == a + c \lor x == a + b + c\}$$

$$\{x == 0 \lor x == a \lor x == c \lor x == a + c\}$$

$$< x = x + b > \quad \|$$

$$\{x == b \lor x == b + a \lor x == b + c \lor x == a + b + c\}$$

$$\{x == 0 \lor x == b \lor x == a \lor x == a + b\}$$

$$< x = x + c >$$

$$\{x == c \lor x == c + b \lor x == c + a \lor x == a + b + c\}$$

$$-- \textit{ inits coend}$$

$$-- \textit{ aplicando regla de la concurrencia}$$

$$\{x == a + b + c\}$$

Manuel I. Capel
manuelcapel@ugr.es