

Examen de Programación Concurrente y Distribuida

3º Curso de Grado en Ingeniería Informática

Febrero. Curso 2013-14

CUESTIONES

1. Indique las propiedades que debe cumplir una transacción para que se considere atómica. **(0,5 Puntos)**
2. Justifique si el siguiente algoritmo para el control de la concurrencia cumple las condiciones requeridas. **(0,75 Puntos)**

```
process P0
1. repeat
2.   c0 := 1;
3.   c0 := c1+1;
4.   while c1<>0 and c1<c0 do;
5.     Sección Crítica
6.   c0 := 0;
7.   Resto0
8. forever
```

```
process P1
1. repeat
2.   c1 := 1;
3.   c1 := c0+1;
4.   while c0<>0 and c0<=c1 do;
5.     Sección Crítica
6.   c1 := 0;
7.   Resto1
8. forever
```

Para referirse a una determinada secuencia de instrucciones, use los número de instrucción, precedidas del nombre del proceso.

3. Usando la instrucción hardware *testset*, garantice la exclusión mutua para los procesos P1 y P2. **(0,75 Puntos)**

```
process P1
repeat

    Sección Crítica

    Resto1
forever
```

```
process P2
repeat

    Sección Crítica

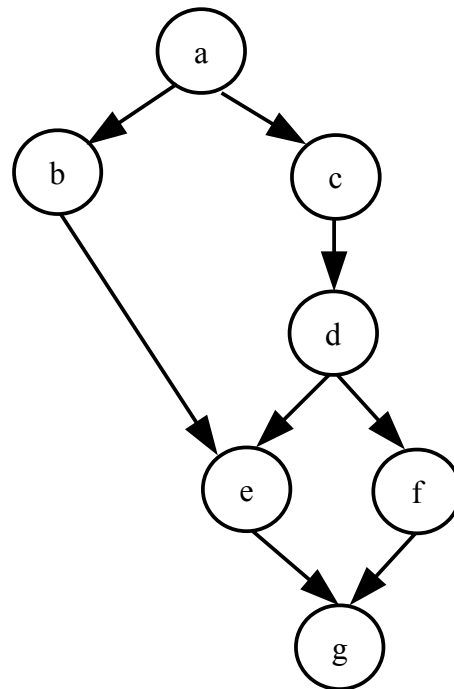
    Resto2
forever
```

4. Dado el siguiente programa, corríjalo, usando como máximo cuatro semáforos, para que se cumpla el grafo de precedencias que se indica. **(1,5 Punto)**

```

Program P
process P1
begin
  a;
  d;
end
process P2
begin
  c;
  f;
  g;
end
process P3
begin
  b;
  e;
end
begin
  conbegin
    P1; P2; P3;
  coend
end

```



PROBLEMAS

5. **(1 Puntos)**. Tenemos un sistema operativo con 5 procesos, y cuatro recursos que presentan los siguientes ejemplares: (2,3,3,2).

Se sabe que las necesidades máximas de los procesos son:

	R1	R2	R3	R4
P1	1	2	2	1
P2	2	1	3	0
P3	2	2	1	2
P4	0	2	1	0
P5	2	1	2	2

y que en un momento dado, los recursos asignados son:

	R1	R2	R3	R4
P1	1	0	2	0
P2	0	1	0	0
P3	0	0	0	0
P4	0	1	0	0
P5	1	0	0	0

Si P3 solicita un ejemplar de R4, ¿se debería conceder si estamos usando un algoritmo de evitación de los interbloqueos?. Justifique la respuesta.

6. En un sistema existen dos impresoras distintas (una láser y una de inyección) y un escáner. En este sistema se ejecutan dos tipos diferentes de procesos, los que únicamente necesitan la impresora (tipo P1), y los que necesitan el escáner y una de las impresoras al mismo tiempo (tipo P2).

Los procesos tipo P1, que sólo necesitan la impresora, prefieren usar la láser, pero si está libre la de inyección también la usarán. Si tienen que esperar lo harán en la que tenga la cola de espera más corta. Una vez en la cola, ya no cambiarán de cola, aunque la otra impresora quede libre.

Los procesos tipo P2, que necesitan el escáner y la impresora siempre usarán la impresora de inyección, y toman los recursos de forma incremental, es decir, primero adquieren el escáner y al poseerlo intentan acceder a la impresora.

- a) Solucionar el problema anterior usando **monitores**. Se asume una semántica de la operación `resume` tipo “desbloquear y espera urgente” (la habitual de *Pascal-FC*). **(2,5 Puntos)**
- b) Solucionar el problema anterior usando **buzones**. Considerar en esta ocasión que solo existen procesos del tipo P1 y que no existe escáner. La solución debe ser correcta para un sistema distribuido, donde los procesos estén en máquinas distintas. **(3 Puntos)**.

NOTAS:

- Para simplificar el código se usarán llaves { y } en lugar de las instrucciones **begin** y **end** para marcar los bloques de código.
- Si fuese necesario que un procedimiento devuelva un valor se permite el uso de la instrucción **return**.

ANEXO 1. Estructura de los procesos para el problema 6

```
program Edicion;

const
    np1=5;
    np2=5;

process type P1(id:integer);
begin
    repeat
        { PROTOCOLO OCUPACION }
        writeln('Proceso tipo P1 ',id,' accediendo');
        { PROTOCOLO LIBERACION }
    forever
end;

process type P2(id:integer);
begin
    repeat
        { PROTOCOLO OCUPACION }
        writeln('Proceso tipo P2 ',id,' accediendo');
        { PROTOCOLO LIBERACION }
    forever
end;

var
    i,j: integer;
    PR1: array[1..np1] of P1;
    PR2: array[1..np2] of P2;

begin
    cobegin
        for i := 1 to np1 do PR1[i](i);
        for j := 1 to np2 do PR2[j](j);
    coend
end.
```