

# Examen de Programación Concurrente y Distribuida

## 3º Curso de Grado en Ingeniería Informática

### ANTES DE COMENZAR:

- Apague el móvil y quítelo de encima del pupitre.
- Ponga su nombre en todos los folios que tenga. Cada pregunta debe responderse en un folio distinto.

1. Indique como el algoritmo del matón realiza la elección del proceso coordinador para sincronizar sistemas distribuidos. **(0,5 Puntos)**
2. Justifique si el siguiente algoritmo para el control de la concurrencia cumple las condiciones requeridas.

**Inicialmente, turno vale 0**

```

process P0
  repeat
    1. c0 := 1;
    2. while c1=1 and turno=1 do;
    3. Sección Crítica
    4. c0 := 0;
    5. turno:= 1;
    6. Resto0
  forever

process P1
  repeat
    1. c1 := 1;
    2. while c0=1 and turno=0 do;
    3. Sección Crítica
    4. c1 := 0;
    5. Turno:=0;
    6. Resto1
  forever
  
```

Para referirse a una determinada secuencia de instrucciones, use los números de instrucción. **(0,75 Puntos)**

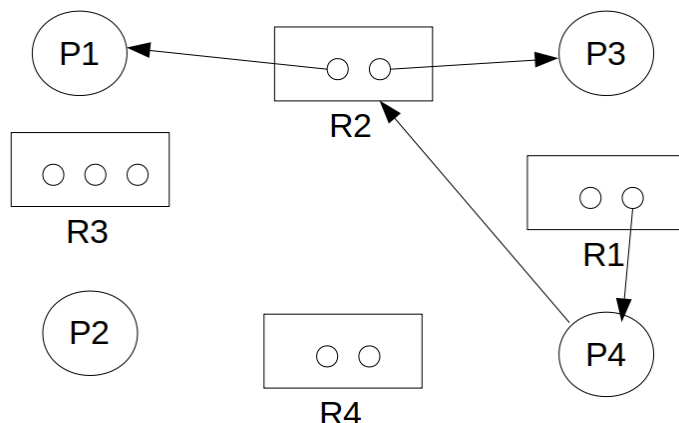
3. Usando semáforos, haga que, de forma cíclica, los procesos accedan a la sección crítica en la siguiente secuencia: P1, P1, P1, P2, P3, P3 P1, P1, P1, P2, P3, P3 .... **(1,5 Puntos)**

**No se considera válida la solución si no se inicializan los semáforos correctamente**

```

Program uuudtt
var
process P1          process P2          process P3          begin
begin              begin                begin                cobegin
  repeat            repeat                repeat                P1;P2;P3;
    Sección Crítica  Sección Crítica        Sección Crítica        coend
    Resto1           Resto2                 Resto3                 end
  forever           forever               forever
end                end                    end
  
```

4. Tenemos un sistema operativo con 4 procesos, que en un momento dado presenta el siguiente estado:



y se sabe que las necesidades máximas de los procesos son:

	N. Máximas			
	R1	R2	R3	R4
P1	1	1	1	0
P2	0	0	2	1
P3	1	2	0	0
P4	1	1	0	1

- a) Indique si se cumplen las condiciones para la existencia de un interbloqueo.
- b) Si estamos usando un algoritmo para evitar los interbloques, ¿debería concederse a P3 un ejemplar del recurso de R1?. Justifique la respuesta. **(1,25 Puntos)**.

5. En una estación de lavado de coches hay tres túneles de lavado dispuestos de forma paralela. A dicha estación llegan para lavarse turismos y furgonetas. Cualquiera de los vehículos podrá usar cualquiera de los túneles, pero si hay una furgoneta en el túnel central no podrá haber otra furgoneta en ninguno de los túneles laterales y viceversa.

- 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*). **(3 Puntos)**.
- b) Solucionar el problema anterior usando **Canales**. 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 puede usar **return**.

**ANEXO 1. Estructura de los procesos para el problema 5**

```
program Febrero22;
const
    nC=20;
    nF=20;

process type TCoche(id:integer);
begin

end;

process type TFurgo(id:integer);
begin

end;

var
    i,j: integer;
    Coche: array[1..nC] of TCoche;
    Furgo: array[1..nF] of TFurgo;

begin
    cobegin
        for i := 1 to nC do Coche[i](i);
        for j := 1 to nF do Furgo[j](j);
    coend
end.
```