Examen de Programación Concurrente y Distribuida
3º Curso de Grado en Ingeniería Informática
Febrero. Curso 2023-24. Convocatoria I

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.**
- **No se corregirá si**
  - **La caligrafía no es legible, tanto por calidad como por tamaño.**
  - **El código que se presenta no está ordenado adecuadamente.**

1. Explique brevemente el funcionamiento del algoritmo de Ricart y Agrawala para solucionar el problema de la exclusión mutua de forma distribuida. **(0,50 Puntos)**

2. Indique si el siguiente algoritmo cumple las condiciones requeridas para solucionar el problema de la exclusión mutua mediante el uso de la instrucción hardware **exchange**. **(0,75 Puntos)**

```
Program P
var
 m:integer

process P0
var
 r0:integer;
begin
r0=0;
repeat
  repeat
    exchange(r0,m);
  until r0 = 1;
  «SECCIÓN CRÍTICA»
  exchange(r0,m)
  Resto0
forever
end
```

```
process P1
var
 r1:integer;
begin
r1=1;
repeat
  repeat
    exchange(r1,m)
  until r1 = 1;
  «SECCIÓN CRÍTICA»
  exchange(r1,m)
  Resto0
forever
end
```

```
begin
 m=1;
 cobegin
   P0;
   P1;
 coend
end.
```

3. Indique el grafo de precedencia que corresponde con el siguiente programa: **(1 Puntos)**

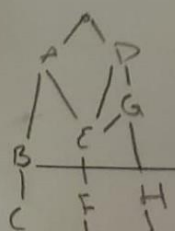
```
Program P
var
   s1: semaphore;
   s2: semaphore;
   s3: semaphore;
```

```
process P1
begin
   A;
   release(S1)
   B;
   C;
   release(S3)
end
```

```
process P2
begin
   acquire(S1)
   acquire(S1)
   D;
   release(S2)
   acquire(S1)
   E;
   F;
   release(S3)
end
```

```
process P3
begin
   acquire(S2)
   acquire(S2)
   G;
   release(S1)
   H;
   acquire(S3)
   acquire(S3)
   I;
end
```

```
begin
   initial(s1,2);
   initial(s2,1);
   initial(s3,0);
   cobegin
      P1;
      P2;
      P3;
   coend
end
```



4. Usando semáforos, haga que, de forma cíclica, los procesos accedan a la sección crítica en la siguiente secuencia: P1, P2, P1, P2, P3 P1, P2, P1, P2, P3 ….. **(1,25 Puntos)**

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

```
Program ududt
var
process P1                process P2                process P3
begin                     begin                     begin
 repeat                    repeat                    repeat
  Sección Crítica           Sección Crítica           Sección Crítica
  Resto1                    Resto2                    Resto3
 forever                   forever                   forever
end                       end                       end

                                                    begin
                                                     cobegin
                                                      P1;P2;P3;
                                                     coend
                                                    end
```

5. Tenemos un sistema operativo con 5 procesos y 4 recursos del que conocemos la siguiente información:

E=(3,2,3,2)

	N. Máximas			
	R1	**R2**	**R3**	**R4**
P1	0	1	0	2
P2	2	0	0	1
P3	0	1	2	0
P4	1	1	1	0
P5	2	1	0	0

ASIGNADOS			
R1	**R2**	**R3**	**R4**
0	1	0	1
1	0	0	0
0	0	1	0
0	1	0	0
2	0	0	0

A partir de dicha situación, llegan las siguientes peticiones:

1. P4 solicita R3
2. P2 solicita R4
3. P5 solicita R2
4. P1 solicita R4
5. P2 solicita R1
6. P4 solicita R1
7. P3 solicita R3
8. P3 solicita R2

De ocurrir interbloqueo, qué solicitud lo provoca?. Justifique la respuesta. **(1 Punto)**

Examen de Programación Concurrente y Distribuida Febrero de 2024

PROBLEMA

En un puesto de control del Rally Dakar hay tres comisarios para sellar el paso. Por el control pasan motos y coches. Las motos pueden sellar con cualquier comisario, pero los coches necesitan dos comisarios a la vez. Los coches tendrán prioridad, pero si hay un coche sellando, el comisario libre podrá atender a una moto.

Cada vez que se pongan 5 sellos, habrá que esperar a que repongan la tinta. Para ello, **al entrar el quinto vehículo** se avisará a un asistente, que deberá traer la tinta. Una vez el asistenta traiga la tinta se podrá seguir sellando.

6. 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)**.

7. Solucionar el problema anterior usando **Buzones**. 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 Febrero24;
const
    nC=20;
    nM=20;

process type TCoche(id:integer);
begin

end;

process type TMoto(id:integer);
begin

end;

process Asistente;
begin

end;

var
    i,j: integer;
    Coche: array[1..nC] of TCoche;
    Moto: array[1..nM] of TMoto;

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