Examen de Programación Concurrente y Distribuida

3º Curso de Grado en Ingeniería Informática

Septiembre. Curso 2021-22

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. Dado el siguiente conjunto de instrucciones, utilice las condiciones de Bernstein para establecer el **grafo de precedencias** que le corresponde, y su correspondiente bloque **cobegin/coend**. **(1 Puntos)**

```
S1: A = G1 + G2;
S2: B = G1 * G2;
S3: C = A - G3;
S4: B = G3 * G4 / D1;
S5: C = B - D1;
S6: F = C + G;
```

2. Dado el siguiente programa, corríjalo, usando la menor cantidad posible de semáforos, para que se cumpla el grafo de precedencias que se indica. No olvide inicializar los semáforos que use. **(1,5 Punto)**

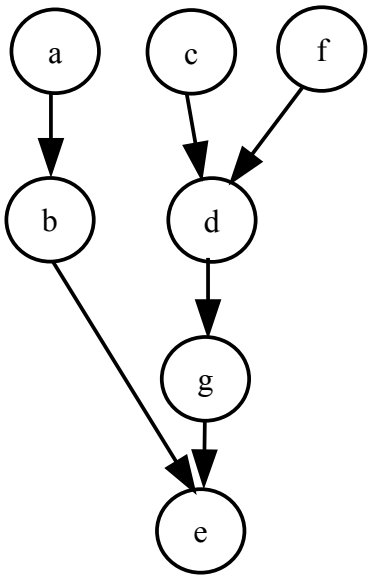
```
Program P

process P1
begin
    a;
    b;
end

process P2
begin
    c;
    d;
    e;
end

process P3
begin
    f;
    g;
end

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



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

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

```
Program udtdu
var
process P1
begin
 repeat
   Sección Crítica
   Resto1
 forever
end

process P2
begin
 repeat
   Sección Crítica
   Resto2
 forever
end

process P3
begin
 repeat
   Sección Crítica
   Resto3
 forever
end

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

4. Un taller de reparaciones tiene 4 puestos de intervención. Los vehículos que llegan al taller pasarán a cualquiera de los puestos que estén libres. Una vez que el vehículo está siendo atendido, se diagnosticará si la avería es leve o grave. (se decidirá aleatoriamente, con una probabilidad del 20% de ser grave). Si la avería se diagnostica como grave, en los otros puestos no se admitirán nuevos vehículos (al quedarse libres) hasta que la avería grave haya sido reparada.

   El tiempo de reparación de la avería leve es de 1 a 2 segundos, y el tiempo de reparación de la avería grave es de 4 a 6 segundos.

   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 Septiembre22;
const
    np=20;

process type Vehiculo(id:integer);
begin

end;

var
    i: integer;
    PA: array[1..np] of Vehiculo;

begin
    cobegin
         for i := 1 to np1 do PA[i](i);
    coend
end.
```