

Examen de Programación Concurrente y Distribuida

3º Curso de Grado en Ingeniería Informática

Febrero. Curso 2018-19

1. Explique de forma escueta los diferentes modelos de comunicación de procesos en los Sistemas Distribuidos. **(0,5 Puntos)**
2. Justifique si el siguiente algoritmo para el control de la concurrencia cumple las condiciones requeridas. **(0,75 Puntos)**

```

process P0
repeat
  while c <> 1 do;
    Sección Crítica
    c := c * -1;
  Resto0
forever

```

```

process P1
repeat
  while c <> -1 do;
    Sección Crítica
    c := c * -1;
  Resto1
forever

```

Donde inicialmente $c = 1$

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

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

Program unodosuno

var

process P1

begin

repeat

Sección Crítica

Resto1

forever

end

process P2

begin

repeat

Sección Crítica

Resto1

forever

end

process P3

begin

repeat

Sección Crítica

Resto2

forever

end

begin

cobegin

P1;P2;P3;

coend

end

4. En una estación de I.T.V trabajan dos inspectores y un supervisor. A dicha estación acuden dos tipos distintos de vehículos:

- Vehículos tipo A. Son aquellos que quieren pasar una inspección periódica. Estos pueden ser atendidos indistintamente por cualquier inspector o por el supervisor.
- Vehículos tipo B. Son aquellos que quieren validar una modificación técnica. Estos necesitan ser atendidos **simultáneamente** por el supervisor y uno cualquiera de los inspectores.

Los vehículos tipo B tendrán prioridad sobre los del tipo A, es decir, los del tipo A no podrán pasar si hay de tipo B esperando, aunque esté libre alguno de los tres.

Cada vehículo será representado por un hilo.

- 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)**
- Solucionar el problema anterior usando **buzones**. **(3 Puntos)**

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

Se sabe que las necesidades máximas de los procesos son:

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

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

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

A partir de ese momento llegan las siguientes solicitudes por parte de los procesos:

- P4 Solicita 1 ejemplar de R2
- P3 Solicita 1 ejemplar de R3
- P2 Solicita 1 ejemplar de R4
- P2 Solicita 1 ejemplar de R3
- P1 Solicita 1 ejemplar de R4
- P3 Solicita 1 ejemplar de R3
- P1 Solicita 1 ejemplar de R1
- P5 Solicita 1 ejemplar de R2

¿Llega el sistema a interbloquearse?. En caso afirmativo, ¿qué solicitud lo provoca?. Justifique la respuesta usando la técnica adecuada. **(1,25 Puntos)**.

ANEXO 1. Estructura de los procesos para el problema 4

```
program ITV;

const
    nPA=15;
    nPB=15;

process type TIPOA(id:integer);
begin
    { PROTOCOLO OCUPACION }
    writeln('Taladrando tablero doble ..');
    { PROTOCOLO LIBERACION }
end;

process type TIPOB(id:integer);
begin
    { PROTOCOLO OCUPACION }
    writeln('Taladrando tablero simple en ..');
    { PROTOCOLO LIBERACION }
end;

var
    i,j: integer;
    VehiculoA: array[1..nPA] of TIPOA;
    VehiculoB: array[1..nPB] of TIPOB;

begin
    cobegin
        for i := 1 to nPA do VehiculoA[i](i);
        for j := 1 to nPB do VehiculoB[j](j);
    coend
end.
```