

Examen de Programación Concurrente y Distribuida

3º Curso de Grado en Ingeniería Informática

Febrero. Curso 2017-18

1. Explique brevemente qué problemas plantea el paso de parámetros en las llamadas a procedimiento remoto (RPC) y cuales son sus posibles soluciones. **(0,5 Puntos)**.
2. Dado el siguiente conjunto de instrucciones, utilice las condiciones de Bernstein para establecer el grafo de precedencias que le corresponde. **(0,75 Puntos)**

$$\begin{aligned}
 S1: & A = 1; \\
 S2: & B = 3; \\
 S3: & C = A - D - K; \\
 S4: & E = B * B / M; \\
 S5: & F = E + G;
 \end{aligned}$$
3. Usando semáforos, haga que, de forma cíclica, el proceso P1 acceda dos veces de forma consecutiva a a sección crítica, seguido por un acceso del proceso P2, es decir, la secuencia de entrada de la sección crítica deberá ser: P1, P1, P2, P1, P1, P2, P1, P1, P2 **(2 Puntos)**

<pre> Program dosporuno var process P1 begin repeat Sección Crítica Resto1 forever end </pre>	<pre> process P2 begin repeat Sección Crítica Resto2 forever end begin cobegin P1;P2; coend end </pre>
--	--

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

<pre> process P1 repeat Sección Crítica Resto1 forever </pre>	<pre> process P2 repeat Sección Crítica Resto2 forever </pre>
---	---

5. En una cooperativa de fruta se procesan fresas y arándanos. Para tal fin se han instalado dos robots seleccionadores y un robot empaquetador.

- El robot *Seleccionador1* se encarga de escoger las fresas de una cinta (que siempre tiene fresas) y colocarlas en una caja. Cuando la caja está completa (asumimos un tiempo aleatorio de entre 1 a 3 segundos para ello), las coloca en la bandeja1 que tiene capacidad máxima para tres cajas. Si la bandeja está llena, el robot espera a que quede sitio en la bandeja.
- El robot *Seleccionador2* se encarga de escoger los arándanos de una cinta (que siempre tiene arándanos) y colocarlas en una caja. Cuando la caja está completa (asumimos un tiempo aleatorio de entre 2 a 3 segundos para ello), las coloca en la bandeja2 que tiene capacidad máxima para tres cajas. Si la bandeja está llena, el robot espera a que quede sitio en la bandeja.
- El robot *Empaquetador* se encarga de coger una caja de la bandeja que más tenga, embalarlas y colocarlas en una cinta de salida, que siempre está disponible (asumimos un tiempo aleatorio de entre 1 a 2 segundos para toda la tarea).

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**. **(2,5 Puntos)**

6. 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	0	2	1	3
P2	1	1	2	2
P3	0	0	2	1
P4	2	2	2	1
P5	2	2	1	2

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

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

Si P4 solicita un ejemplar de R1, ¿se debería conceder si estamos usando el algoritmo del banquero para evitar los interbloqueos?. Justifique la respuesta. **(1 Puntos)**.

ANEXO 1. Estructura de los procesos para el problema 5

```
program Cooperativa;
```

```
process Seleccionador1;  
begin
```

```
    repeat
```

```
        sleep(1+random(3)); // Montando caja de fresa
```

```
    forever
```

```
end;
```

```
process Seleccionador2;  
begin
```

```
    repeat
```

```
        sleep(2+random(2)); // Montando caja de arándanos
```

```
    forever
```

```
end;
```

```
process Empaquetador;  
begin
```

```
    repeat
```

```
        sleep(1+random(2)); // embalando caja y soltándola en la cinta
```

```
    forever
```

```
end;
```

```
begin
```

```
    cobegin
```

```
        Seleccionador1;
```

```
        Seleccionador2;
```

```
        Empaquetador;
```

```
    coend
```

```
end.
```