



Universidad
de Huelva

DEPARTAMENTO DE INGENIERÍA ELECTRÓNICA,
SISTEMAS INFORMÁTICOS Y AUTOMÁTICA

PROGRAMACIÓN

Curso 2003/2004

2 de febrero de 2004

Ejercicio 1 (1 punto)

Sea el siguiente programa:

```
PROGRAM ejercicio1;
VAR x: INTEGER;

PROCESS p1;
BEGIN
  y := 1;
END;

PROCESS p2;
BEGIN
  x := x + y;
END;

PROCESS p3;
BEGIN
  x := x + y;
END;

BEGIN
  x := 0; y := 0;
  COBEGIN
    p1; p2; p3
  COEND
  writeln('El valor de x es', x);
END.
```

- Si nuestro objetivo es que el resultado final sea "El valor de x es 2", que problemas presenta el programa. Razona la respuesta.
- Soluciona los problemas con Semáforos para conseguir ese resultado.

Ejercicio 2 (1 punto)

El paso de mensajes mediante invocación remota (estilo Ada) permite la comunicación, en ambas direcciones, y la sincronización entre procesos. Suponiendo que dos procesos P1 y P2 deben hacer uso de ella, implementa el pseudocódigo para todas las posibles alternativas (ver figura) de comunicación y sincronización, teniendo en cuenta que es el proceso P1 el que define los puntos de entrada.

$P_1 \rightarrow P_2$

$P_1 \leftarrow P_2$

$P_1 \leftrightarrow P_2$

$P_1 \leftrightarrow P_2$



Universidad
de Huelva

DEPARTAMENTO DE INGENIERÍA ELECTRÓNICA,
SISTEMAS INFORMÁTICOS Y AUTOMÁTICA

PROGRAMACIÓN

Curso 2003/2004

2 de febrero de 2004

Ejercicio 3

Supongamos un proceso *Productor* que genera los números del 1 al 1000, como muestra el pseudocódigo, y los coloca en un Buffer, para que dos procesos consumidores puedan retirarlos. De tal manera que uno, *Consumidor5*, sólo retirará los valores múltiplos de 5, y el otro, *ConsumidorX*, retirará el resto (los no múltiplos de 5). Tal como muestra el pseudocódigo, ambos procesos utilizan el procedimiento *retirar*, donde el primer parámetro será la variable que recibirá el dato retirado y el segundo el indicativo de si el dato a retirar debe ser múltiplo de 5 (*true*) o no (*false*).

<pre>PROCESS Productor; VAR i : INTEGER; BEGIN FOR i:=1 TO 1000 DO Buffer.colocar(i); END;</pre>	<pre>PROCESS Consumidor5; VAR i,x : INTEGER; BEGIN FOR i:=1 TO 200 DO BEGIN Buffer.retirar(x, true); writeln(x); END; END;</pre>	<pre>PROCESS ConsumidorX; VAR i,x : INTEGER; BEGIN FOR i:=1 TO 800 DO BEGIN Buffer.retirar(x, false); writeln(x); END; END;</pre>
---	---	--

Se pide, teniendo en cuenta que el array máximo permitido es de 5 posiciones, resolver la problemática del problema productor-consumidor basado en el código anterior con:

a) Monitores (2 puntos)

b) Recursos (2 puntos)