

FUNDAMENTOS DE INFORMÁTICA¹

(12 de septiembre 2013; 2ª PARTE: ejercicios; 5 puntos)

1. La siguiente secuencia de valores corresponde a un fichero codificado por medio del algoritmo Lempel-Ziv

(0,0,0) (001) (2,2,1) (4,4,0) (10, 9, 1) (19,15,1) (25,23,0)

- a) Obtener la cadena de bits que codifica (dar el resultado en hexadecimal)
b) Suponiendo que se codificase una cadena de 32 KB con 512 tripletas y cada una de éstas ocupase 16 bits ¿Cuál sería el factor de compresión?

Solución:

- a) Teniendo en cuenta el significado de los tres símbolos que se incluyen en cada tripleta, se tiene que:

(0,0,0)	(0,0,1)	(2,2,1)	(4,4,0)	(10, 9, 1)	(19, 15, 1)	(25, 23, 0)
0	1	01 1	1011 0	010111011 1	101110110010111 1	101110111 10111011001011 0

Es decir, la cadena de bits es:

0 1 01 1 1011 0 010111011 1 101110110010111 1
10111011110111011001011 0

En hexadecimal será:

0101	1101	1001	0111	0111	1011	1011	0010	1111	1011	1011	1101	1101	1001	0110
5	D	9	7	7	B	B	2	F	B	B	D	D	9	6

Respuesta: **5D977BB2FBBDD96**

- b) Suponiendo que se codificase una cadena de 32 KB con 512 tripletas cada una de ellas ocupase 16 bits ¿Cuál sería el factor de compresión?

Capacidad del fichero original (antes de comprimir): $C_a = 32 \text{ KB}$

Capacidad del fichero comprimido: $C_d = 512 \cdot 16 \text{ bits} = 512 \cdot 2 \text{ Bytes} = 1024 \text{ Bytes} = 1 \text{ Kbyte}$

Luego el factor de conversión es:

$$f_c = \frac{C_a}{C_d} = \frac{32 \text{ KB}}{1 \text{ KB}} = 32$$

Puntuación: 1a → 0,25 puntos; 1b → 0,5; 1c → 0,25
2a → 0,8; 2b → 0,1; 2c → 0,2; 2d → 0,2; 2e → 0,2
3 → 1
4a → 0,3; 4b → 0,3; 4c → 0,3; 4d → 0,3; 4e → 0,3
Mala presentación: puede bajar hasta 0,5 puntos.

2. Hacer un programa en código máquina para CODE-2, a ubicarse a partir de la dirección de memoria **A000**, que encuentre el número de valores negativos dentro de una zona determinada de su memoria, y proporcione el resultado por el puerto de salida **OP1**. Las direcciones iniciales y finales de la zona las debe dar el usuario a través del puerto de entrada **IP01**.

Para la realización del programa, considere los siguientes pasos:

- Organigrama
- Asignación de registros y memoria
- Redactar el programa en nemónicos (código máquina).
- Escribir el programa completo en hexadecimal
- Obtener el tiempo que tardaría en ejecutarse el programa en función del número de posiciones de la zona de memoria a analizar, suponiendo que la frecuencia de reloj de CODE-2 fuese de 10 GHz

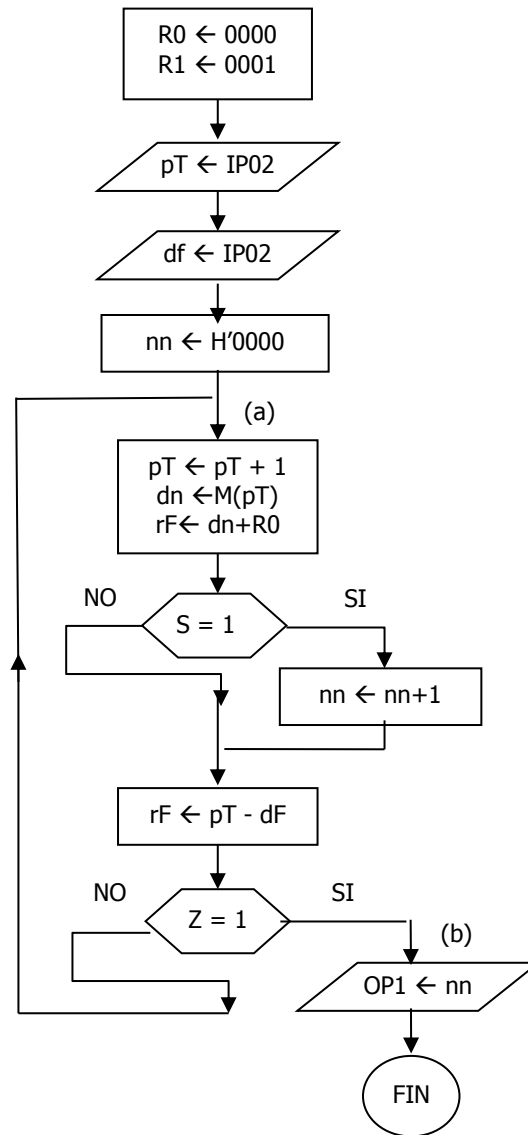
SOLUCIONES

a. Organigrama.

Utilizamos las siguientes variables:

- Puntero de memoria dentro de la zona donde se encuentran los números: **pT**. Su valor inicial será la dirección de comienzo de la zona de datos (**di**).
- Dirección final de la zona de números: **df**
- Nº de números negativos: **nn**. Su valor inicial debe ser 0.
- Dato nuevo a considerar: **dn**. Este dato se irá leyendo iterativamente de la memoria.

En la figura se encuentra el organigrama.



b. Asignación de registros y de memoria.

- El programa se carga a partir de la dirección M(0000)

Rtro.	Contenido	Valor inicial
r0	Valor 0	0000
r1	Valor 1	0001
r3	Puntero de la lista de datos (pT)	En IP01
r4	Dirección final de la lista de datos (df)	En IP01
r5	Dato nuevo en consideración (dn)	
r6	Número de negativos (nn)	0000

a) Programa en nemónicos y código máquina. Puede verse en la siguiente tabla:

Dcc. de salto	Dirección	Nemónico	Hexadecimal	Comentarios
	0000	LLI R0,00	2000	$R0 \leftarrow H'0000$
	0001	LLI R1,01	2101	$R1 \leftarrow H'0001$
	0002	IN R3, IP02	4302	$R3 \leftarrow IP02$ (dirección inicial)
	0003	IN R4, IP02	4402	$R4 \leftarrow IP02$ (dirección final)
	0004	ADDS RD, R3, R0	6D30	$RD \leftarrow R3$
	0005	LLI R6,00	0600	$R6 \leftarrow H'0000$ (nº de mínimos inicial)
(a)	0006	ADDS R3,R3,R1	6331	$pT \leftarrow pT + 1$

	0007	ADDS RD, R3, R0	6D30	$r0 \leftarrow rD$
	0008	LD R5, [00]	0500	$dn \leftarrow M(pT)$
	0009	ADDS R5,R5,R0	7F56	activar biestables indicadores
	000A	LLI RD, 11	2D11	dirección de salto (b)
	000B	BS	C200	saltar a (b) si el n° es negativo
(c)	000C	SUB RF,R3,R4	7F34	$pT-dF$
	000D	LLI RD, 14	2D14	dirección de salto (d)
	000E	BZ	C100	saltar a (d) si $pT = dF$ (se ha llegado al final)
	000F	LLI RD, 06	2D06	dirección de salto (a)
	0010	BR	C000	Salto incondicional a (a)
(b)	0011	ADDS R6,R6,R1	6650	Actualizar el n° de datos negativos : $r6 \leftarrow r5$
	0012	LLI RD, 0C	2D0C	dirección de salto (c)
	0013	BR	C000	Salta a (c)
(d)	0014	OUT OP01,R6	5601	Salida del número de negativos (nn)
	0015	HALT	F000	Fin

b) Tiempo que tardaría en ejecutarse el programa

Llamando N al número de datos que hay en la zona de memoria a considerar (df-di), en la siguiente tabla se indica el número de ciclos que consume cada instrucción

Nemónico	Nº de ciclos	Nº de veces que se ejecuta
LLI R0,00	6	1
LLI R1,01	6	1
IN R3, IP02	8	1
IN R4, IP02	8	1
ADDS RD, R3, R0	7	1
LLI R6, 00	6	1
ADDS R3,R3,R1	7	N
ADDS RD, R3, R0	7	N
LD R5, [00]	8	N
ADDS R5,R5,R0	7	N
LLI RD, 11	6	N
BS	6	N
SUBS RF,R3,R4	7	N
LLI RD, 14	6	N
BZ	6	N
LLI RD, 06	6	N
BR	6	N
ADDS R6,R5,R0	7	N
LLI RD, 0C	6	N
BR	6	N
OUT OP01,R6	8	1
HALT	6	1

Obsérvese que como el tiempo de ejecución del programa hay que calcularlo “en el peor de los casos” se ha supuesto que en todos los casos el valor del número es negativo.

Entonces el número total de ciclos consumidos en la ejecución del programa será:

$$NC = 41 + 91 \cdot N + 14 = 91 \cdot N + 55$$

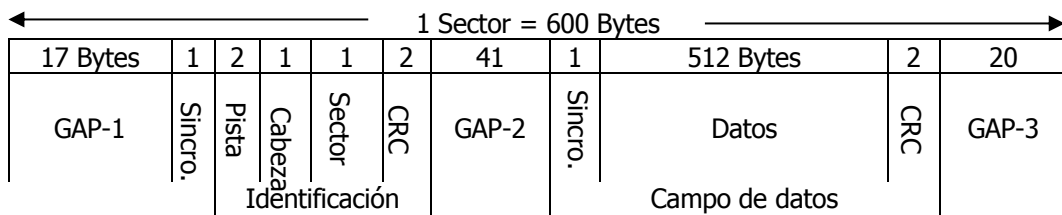
Por otra parte, el periodo de reloj (tiempo de ciclo) es:

$$T = \frac{1}{F} = \frac{1}{10 \times 10^9} = 100 \mu s$$

con lo que el tiempo de ejecución del programa sería:

$$t = T \cdot NC = 100 \mu s \cdot (91N + 55) = (9,1 \cdot N + 5,5) ms$$

3. Se dispone de una unidad de disco en el que el formato de grabación de los sectores es el que se indica en la siguiente figura.



Suponiendo que la pista 800, se encuentra a una distancia del centro de $1\frac{1}{2}$ " (3,81 cm) tiene un total de 16 sectores, girando el disco cuando se encuentra en dicha pista la cabeza lectora/grabadora a 7.200 rpm; obtener:

- El caudal de datos (en Megabits/s) de los datos útiles para el usuario de esa pista.
- La densidad lineal de grabación en dicha pista (en bits/cm).
- El tiempo máximo (en microsegundos, μs) que puede utilizar la controladora del disco para comprobar si el código de error CRC de la identificación es válido para proceder a leer el campo de datos (*Pista*: este tiempo coincide con el tiempo de recorrido del GAP-2).

a) El caudal de datos (en bits/s) de los datos útiles para el usuario de esa pista.

Cada sector de la pista considerada tiene 16 sectores, y cada sector tiene 512 Bytes = 0,5 KB de datos útiles para el usuario, con lo que la capacidad de información útil para el usuario en una pista es:

$$C_{usuario} = 16 \cdot 0,5 KB = 8KB$$

Esta información se tiene que transmitir a la caché de la controladora del disco en el tiempo que dura una rotación del mismo. Llamando T_r al periodo de rotación se verifica que:

$$T_r = \frac{1}{\omega_r} = \frac{1}{7200 r/m} = \frac{1}{120 r/s} = 0,0083 s = 8,3 ms$$

Con lo que el caudal de datos, r , deberá ser:

$$r = \frac{8 KB}{8,3 \cdot 10^{-3} s} = 0,96 \cdot 1000 KB/s = 960 KB/s$$

RESULTADO: 960 KB/s

b) La densidad lineal de grabación en dicha pista (en bits/cm).

El número de bits que tiene la pista es:

$$C_{pista} = n_{sectores} \cdot n_{bits/sector} = 16 \cdot (17 + 1 + 2 + 1 + 1 + 2 + 41 + 1 + 512 + 2 + 20) Bytes \cdot 8 \frac{bits}{Byte} = 76.800 bits$$

Por otra parte la longitud de esa pista es:

$$l_{pista} = 2 \cdot \pi \cdot r = 2 \cdot 3,1416 \cdot 1,5 = 9,4248 \text{ pulgadas}$$

Luego la densidad lineal será:

$$\lambda = \frac{C_{pista}}{l_{pista}} = \frac{76.800 bits}{9,4248''} = 8.149 bpi = \frac{8.149 \frac{bits}{pulgada}}{2,54 \frac{cm}{pulgada}} = 3.212,2 bits/cm$$

RESULTADO: 3.212,4 bits/cm

c) El tiempo máximo (en microsegundos, μs) que puede utilizar la controladora del disco para comprobar si el código de error CRC de la identificación es válido para proceder a leer el campo de datos (Pista: este tiempo coincide con el tiempo de recorrido del GAP-2).

Como, una vez leída la cabecera de un sector, el disco sigue girando; para dar tiempo a hacer la comprobación de los CRC se incluye en el formato el GAP-2, constituido por 41 Bytes. Es decir, el tiempo máximo que debe tardar la controladora de disco en hacer la comprobación ha de coincidir con el tiempo en que se tardan en recorrer esos 41 Bytes. Teniendo en cuenta el periodo de giro del motor, T_r (calculado anteriormente) el tiempo que se invierte en recorrer un Byte será:

$$t_{Byte} = \frac{T_r}{C_{pista}} = \frac{8,3 ms}{9.600 Bytes} = 0,87 \frac{\mu s}{Byte}$$

Es decir, el tiempo en recorrer los 41 Bytes de GAP-2 será:

$$t_{GAP} = n_{Bytes} \cdot t_{Byte} = 41 \cdot 0,87 = 35,6 \mu s$$

RESULTADO: 35,6 μs

4. En un computador se presentan cuatro procesos, cuyas prioridades, instantes de llegada y millones de instrucciones a ejecutar por el procesador se indican en la tabla. El procesador utilizado es de una frecuencia de reloj de $F = 0,5$ GHz y, por término medio, emplea dos ciclos en ejecutar cada instrucción. El sistema operativo invierte (5 ms) para realizar la planificación cuando concluye un proceso o cuando debe realizar cambios de contexto.

Proceso	Prioridad (1 la mayor)	Instante de llegada (ms)	Nº de instrucciones (millones)	Tiempo de CPU (ms)
P1	3	0	7,5	
P2	4	15	5	
P3	2	25	3,75	
P4	1	40	8,75	

- a) Obtener los tiempos de ejecución de cada proceso (*incluirlos en la última columna de la tabla*).

Obtener el coeficiente de respuesta del proceso P3 en los siguientes 4 casos:

- b) Planificación **FCFS** (*First Come, First Served*) y sistema operativo no apropiativo.

Resultado	
-----------	--

- c) Planificación por prioridades y sistema operativo no apropiativo

Resultado	
-----------	--

- d) Planificación por prioridades y sistema operativo apropiativo (es decir, el SO interrumpe a un proceso en ejecución cuando llega otro más prioritario)

Resultado	
-----------	--

- e) Planificación **SPN** (*Shortest Process Next*) y sistema operativo no apropiativo.

Resultado	
-----------	--

SOLUCIONES:

- a) **Obtener los tiempos de ejecución de cada proceso (*incluirlos en la última columna de la tabla*).**

El tiempo de procesador será el producto del nº de instrucciones (N_I) del proceso por lo que tarda en ejecutarse cada una de ellas. Y el tiempo de ejecución de cada instrucción se puede estimar como el producto del número de ciclos medio de cada instrucción (N_C) por el periodo de reloj ($T=1/F$); es decir:

$$t = \frac{N_I \cdot N_C}{F}$$

En nuestro caso, $N_c = 2$ y $F = 0,5$ GHz; con lo que, llamando a nm el nº de millones de instrucciones, se tiene que:

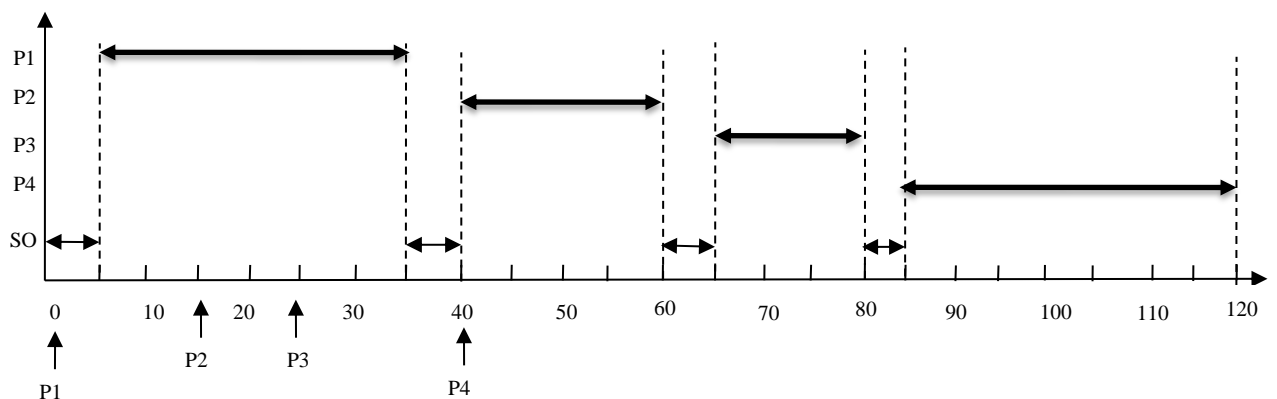
$$t = \frac{nm \cdot 10^6 \cdot 2}{0,5 \cdot 10^9} = 4 \cdot nm \text{ milisegundos}$$

Es decir, tenemos los tiempos de procesador que se indican en la última columna de la siguiente tabla:

Proceso	Prioridad (1 la mayor)	Instante de llegada (ms)	Nº de instrucciones (millones)	Tiempo de CPU (ms)
P1	3	0	7,5	30
P2	4	15	5	20
P3	2	25	3,5	15
P4	1	40	8,75	35

b) **Obtener el coeficiente de respuesta del proceso P3 suponiendo planificación FCFS (First Come, First Served) y sistema operativo no apropiativo.**

La prioridad es $P1 \rightarrow P2 \rightarrow P3 \rightarrow P4$; es decir, cuando hay varios procesos pendientes de ejecutar, se ejecuta primero el que de ellos esté antes en la lista anterior.



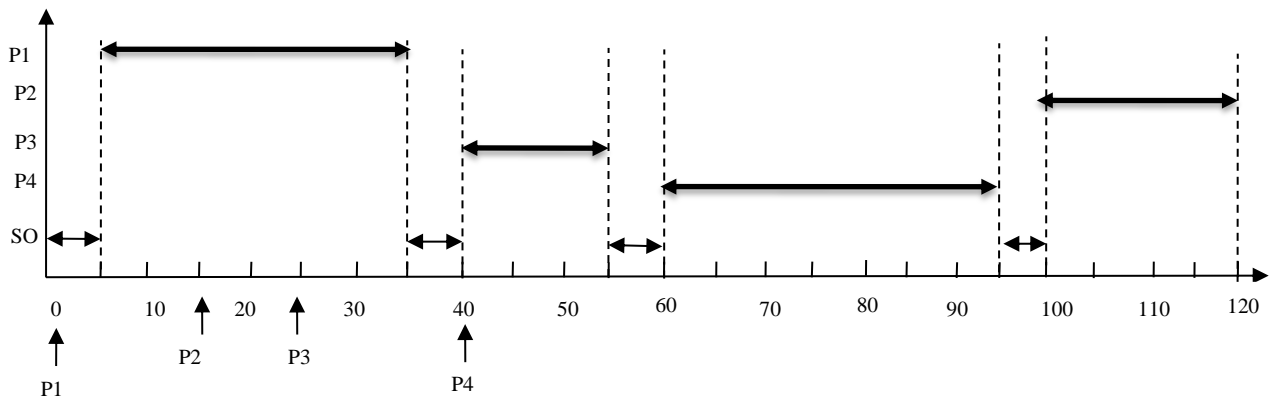
Por definición el tiempo de respuesta (t_r), es el tiempo comprendido entre el instante que llega el proceso (t_0) y el tiempo en que finaliza el mismo (t_f); y el coeficiente de respuesta (R) es el tiempo de respuesta dividido por el tiempo de procesamiento (t_{CPU}); es decir, para el proceso B se tiene:

$$R_{P3} = \frac{t_{r3}}{t_{CPU,3}} = \frac{t_{f3} - t_{03}}{t_{CPU,3}} = \frac{80 - 25}{15} = 3,67$$

Resultado	3,67
-----------	------

- c) **Obtener el coeficiente de respuesta del proceso P3 suponiendo planificación por prioridades y sistema operativo no apropiativo.**

Ahora la prioridad es $P4 \rightarrow P3 \rightarrow P1 \rightarrow P2$; es decir, cuando hay varios procesos pendientes de ejecutar, se ejecuta primero el que de ellos esté antes en la lista anterior.

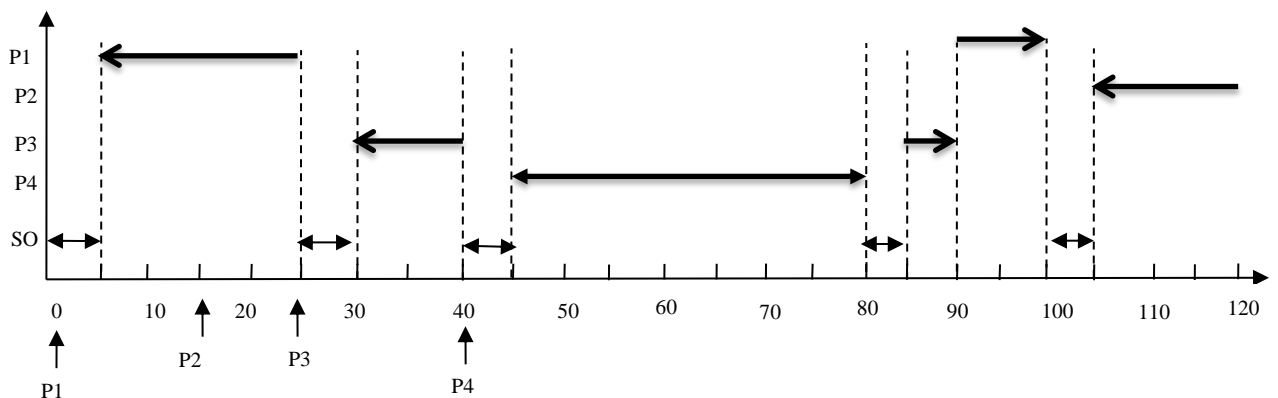


$$R_{P3} = \frac{t_{r3}}{t_{CPU,3}} = \frac{55 - 25}{15} = 2$$

Resultado	2
-----------	---

- d) **Obtener el coeficiente de respuesta del proceso P3 suponiendo planificación por prioridades y sistema operativo apropiativo (es decir, el SO interrumpe a un proceso en ejecución cuando llega otro más prioritario)**

La prioridad sigue siendo $P4 \rightarrow P3 \rightarrow P1 \rightarrow P2$; pero ahora en el momento que llegue un nuevo proceso con prioridad mayor que el que se esté ejecutando, el SO interrumpe a éste para dar paso al más prioritario.

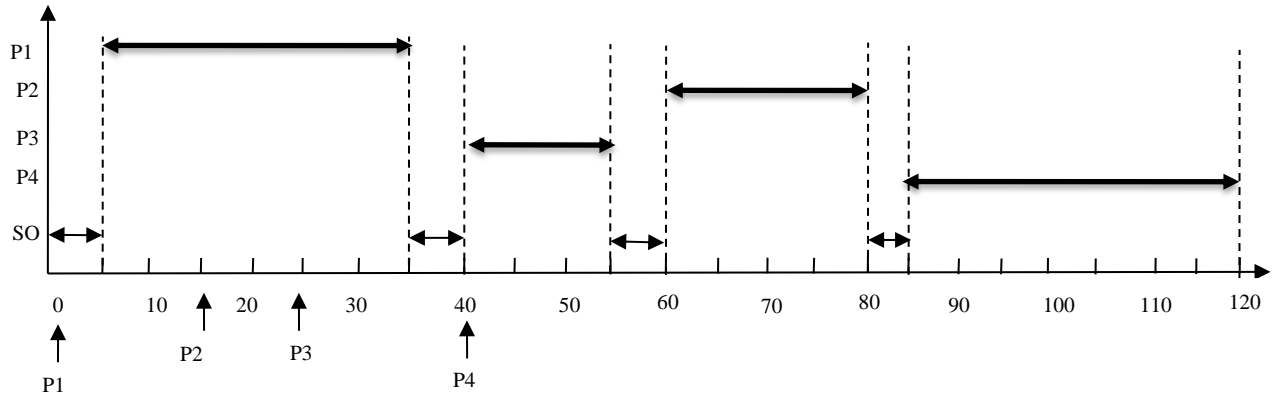


$$R_{P3} = \frac{t_{r3}}{t_{CPU,3}} = \frac{90 - 25}{15} = 4,3$$

Resultado	4,3
-----------	-----

e) Obtener el coeficiente de respuesta del proceso P3 suponiendo planificación SPN (*Shortest Process Next*) y sistema operativo no apropiativo.

Ahora la prioridad es P3 → P2 → P1 → P4; es decir, cuando hay varios procesos pendientes de ejecutar, se ejecuta primero el que de ellos esté antes en la lista anterior.



$$R_{P3} = \frac{t_{r3}}{t_{CPU,3}} = \frac{55 - 25}{15} = 2$$

Resultado	2
-----------	---