

1. Explique razonadamente si las siguientes afirmaciones son verdaderas o falsas:

I) (1 p) El algoritmo de Coffman se utiliza en la estrategia de interbloques conocida como evitación o predicción de interbloques.

**Falso. El algoritmo de Coffman se utiliza en la estrategia de interbloques conocida como detección y recuperación de interbloques.**

II) (1 p) La numeración de los sectores de una pista de un disco duro se realiza de forma contigua: 0, 1, 2,...

**Falso. Debido a que el controlador de disco no puede leer ni escribir dos sectores contiguos, éstos se numeran de forma intercalada, de acuerdo a un determinado factor.**

III) (1 p) La estrategia del buffering de páginas consiste en almacenar en el área de intercambio una copia de un conjunto (buffer) de páginas de un proceso.

**Falso. Consiste en utilizar como buffer de páginas la propia lista de marcos libres, ya que no se borran, si no que se marcan como libres simplemente hasta que se reemplazan. Así, si hiciera falta alguna, se aprovecha que está ahí en lugar de tener que leerla de la memoria secundaria.**

IV) (1 p) La segmentación simple facilita la protección y compartición de las diferentes partes de un programa.

**Verdadero. Puesto que se realiza a nivel de segmento.**

2. En un computador con  $x$  instancias de un recurso  $R_1$ , y instancias de un recurso  $R_2$  y  $z$  instancias de un recurso  $R_3$  se están ejecutando los procesos  $P_1, P_2, P_3, P_4$  y  $P_5$ . En un cierto instante de tiempo  $T$  la matriz  $N$  de recursos máximos necesitados, la matriz  $A$  de recursos asignados y el vector de recursos disponibles  $R_D$  son:

$$\begin{array}{ccc}
 4 & 4 & 4 \\
 1 & 4 & 6 \\
 N = ( & 3 & 2 & 5 & ) \\
 4 & 4 & 8 \\
 5 & 1 & 10 \\
 \end{array}
 \quad
 \begin{array}{ccc}
 1 & 0 & 0 \\
 0 & 1 & 2 \\
 A = ( & 0 & 0 & 3 & ) \\
 1 & 0 & 3 \\
 1 & 1 & 0 \\
 \end{array}
 \quad
 R_D = ( 3 \quad 3 \quad 2 )$$

En cada matriz se ha asociado la fila  $i$  al proceso  $P_i$  ( $i = 1, 2, 3, 4$  y  $5$ ) y la columna  $j$  al recurso  $R_j$  ( $j = 1, 2$  y  $3$ ). Se pide:

a) (0.6 p) Calcular  $x, y$  y  $z$ .

$$R_{A_j} = \sum_{k=1}^5 A_{kj} \quad R_{A_1} = 1+0+0+1+1=3 \quad R_{A_2} = 0+1+0+0+1=3 \quad R_{A_3} = 0+2+3+3+0=8 \\
 R_A = (3 \ 2 \ 8) \quad R_E = R_D + R_A = (3 \ 3 \ 2) + (3 \ 2 \ 8) = (6 \ 5 \ 10)$$

$$\mathbf{x=6 \quad y=5 \quad z=10}$$

b) (1.2 p) Determinar si el estado en el instante  $T$  es seguro.

Llamamos  $S_1$  al estado en el instante  $T$ :

$$\begin{array}{ccc}
 4 & 4 & 4 \\
 1 & 4 & 6 \\
 S_1 = \{ N = ( & 3 & 2 & 5 & ) \\
 4 & 4 & 8 \\
 5 & 1 & 10 \\
 \end{array}
 \quad
 \begin{array}{ccc}
 1 & 0 & 0 \\
 0 & 1 & 2 \\
 A = ( & 0 & 0 & 3 & ) \\
 1 & 0 & 3 \\
 1 & 1 & 0 \\
 \end{array}
 \quad
 R_E = ( 6 \ 5 \ 10 ) \quad R_D = ( 3 \ 3 \ 2 ) \}$$

Y calculamos  $N-A$ :

$$\begin{array}{ccc}
 4 & 4 & 4 \\
 1 & 0 & 0 \\
 \end{array}
 \quad
 \begin{array}{ccc}
 3 & 4 & 4 \\
 \end{array}$$

$$\begin{array}{r}
 1 \ 4 \ 6 \\
 3 \ 2 \ 5 \\
 4 \ 4 \ 8 \\
 5 \ 1 \ 10
 \end{array}
 -
 \begin{array}{r}
 0 \ 1 \ 2 \\
 0 \ 0 \ 3 \\
 1 \ 0 \ 3 \\
 1 \ 1 \ 0
 \end{array}
 =
 \begin{array}{r}
 1 \ 3 \ 4 \\
 3 \ 2 \ 2 \\
 3 \ 4 \ 5 \\
 4 \ 0 \ 10
 \end{array}$$

Ahora comprobamos si se cumple  $(N_i - A_i) \leq R_D$ , que solo se cumple para  $N_3$ :

$$\begin{array}{l}
 (N_3 - A_3) \leq R_D \\
 (3 \ 2 \ 2) \leq (3 \ 3 \ 2)
 \end{array}$$

Por lo que si suponemos que proceso  $P_3$  puede terminar, actualizamos el estado del sistema:

$$\begin{array}{r}
 4 \ 4 \ 4 \\
 1 \ 4 \ 6 \\
 0 \ 0 \ 0 \\
 4 \ 4 \ 8 \\
 5 \ 1 \ 10
 \end{array}
 A =
 \begin{array}{r}
 1 \ 0 \ 0 \\
 0 \ 1 \ 2 \\
 0 \ 0 \ 0 \\
 1 \ 0 \ 3 \\
 1 \ 1 \ 0
 \end{array}
 R_E =
 \begin{array}{r}
 6 \ 5 \ 10 \\
 3 \ 3 \ 5
 \end{array}
 R_D =
 \begin{array}{r}
 3 \ 4 \ 4 \\
 1 \ 3 \ 4 \\
 0 \ 0 \ 0 \\
 3 \ 4 \ 5 \\
 4 \ 0 \ 10
 \end{array}$$

$$\begin{array}{r}
 4 \ 4 \ 4 \\
 1 \ 4 \ 6 \\
 0 \ 0 \ 0 \\
 4 \ 4 \ 8 \\
 5 \ 1 \ 10
 \end{array}
 -
 \begin{array}{r}
 1 \ 0 \ 0 \\
 0 \ 1 \ 2 \\
 0 \ 0 \ 0 \\
 1 \ 0 \ 3 \\
 1 \ 1 \ 0
 \end{array}
 =
 \begin{array}{r}
 3 \ 4 \ 4 \\
 1 \ 3 \ 4 \\
 0 \ 0 \ 0 \\
 3 \ 4 \ 5 \\
 4 \ 0 \ 10
 \end{array}$$

$$\begin{array}{l}
 (N_2 - A_2) \leq R_D \\
 (1 \ 3 \ 4) \leq (3 \ 3 \ 5)
 \end{array}$$

Suponemos que termina  $P_2$ .

$$\begin{array}{r}
 4 \ 4 \ 4 \\
 1 \ 0 \ 0
 \end{array}$$

$$S_3 = \{ N = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 4 & 4 & 8 \\ 5 & 1 & 10 \end{pmatrix} \quad A = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 3 \\ 1 & 1 & 0 \end{pmatrix} \quad R_E = \begin{pmatrix} 6 & 5 & 10 \end{pmatrix} \quad R_D = \begin{pmatrix} 3 & 4 & 7 \end{pmatrix} \}$$

$$N - A = \begin{pmatrix} 4 & 4 & 4 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 4 & 4 & 8 \\ 5 & 1 & 10 \end{pmatrix} - \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 3 \\ 1 & 1 & 0 \end{pmatrix} = \begin{pmatrix} 3 & 4 & 4 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 3 & 4 & 5 \\ 4 & 0 & 10 \end{pmatrix}$$

$$(N_1 - A_1) \leq R_D \\ (3 \ 4 \ 4) \leq (3 \ 4 \ 7)$$

$$(N_4 - A_4) \leq R_D \\ (3 \ 4 \ 5) \leq (3 \ 4 \ 7)$$

Aquí vemos que podrían terminar dos procesos ( $P_1$  y  $P_4$ ). Aunque en realidad tendrían que hacerlo secuencialmente, suponemos que terminan ambos a la vez:

$$S_3 = \{ N = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 5 & 1 & 10 \end{pmatrix} \quad A = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 0 \end{pmatrix} \quad R_E = \begin{pmatrix} 6 & 5 & 10 \end{pmatrix} \quad R_D = \begin{pmatrix} 5 & 4 & 10 \end{pmatrix} \}$$

$$\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

$$\begin{array}{r}
 \begin{array}{ccc} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{array} \\
 N-A=( \begin{array}{ccc} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 5 & 1 & 10 \end{array} ) - ( \begin{array}{ccc} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 0 \end{array} ) = ( \begin{array}{ccc} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 4 & 0 & 10 \end{array} )
 \end{array}$$

Y comprobamos que el último proceso que queda podría terminar:

$$\begin{array}{l}
 (N_5 - A_5) \leq R_D \\
 (4 \ 0 \ 10) \leq (5 \ 4 \ 10)
 \end{array}$$

**Por lo que al haber encontrado una secuencia de asignación de recursos que garantiza que los procesos pueden finalizar sin producirse un interbloqueo, hemos demostrado que el estado en el instante T es seguro.**

*c) (1.2 p) Aplicando el algoritmo del banquero determinar si el sistema puede admitir una petición del proceso  $P_3$  de una instancia del recurso  $R_2$ .*

1. Partimos de  $S_0$ , estado que sabemos seguro por el apartado anterior:

$$\begin{array}{r}
 \begin{array}{ccc} 4 & 4 & 4 \\ 1 & 4 & 6 \\ 3 & 2 & 5 \end{array} \\
 S_0 = \{ N=( \begin{array}{ccc} 4 & 4 & 8 \\ 5 & 1 & 10 \end{array} ) \ A=( \begin{array}{ccc} 1 & 0 & 0 \\ 0 & 1 & 2 \\ 0 & 0 & 3 \end{array} ) \ R_E=( \begin{array}{ccc} 6 & 5 & 10 \end{array} ) \ R_D=( \begin{array}{ccc} 3 & 3 & 2 \end{array} ) \}
 \end{array}$$

2. Simulamos que concedemos la instancia de  $R_2$  a  $P_3$ :

$$\begin{array}{r}
 \begin{array}{ccc} 4 & 4 & 4 \\ 1 & 4 & 6 \\ 3 & 2 & 5 \end{array} \\
 S' = \{ N=( \begin{array}{ccc} 4 & 4 & 8 \\ 5 & 1 & 10 \end{array} ) \ A=( \begin{array}{ccc} 1 & 0 & 0 \\ 0 & 1 & 2 \\ 0 & 1 & 3 \end{array} ) \ R_E=( \begin{array}{ccc} 6 & 5 & 10 \end{array} ) \ R_D=( \begin{array}{ccc} 3 & 2 & 2 \end{array} ) \}
 \end{array}$$

3. Comprobamos si  $S'$  es seguro:

$$\begin{array}{r}
4 \ 4 \ 4 \quad 1 \ 0 \ 0 \quad 3 \ 4 \ 4 \\
1 \ 4 \ 6 \quad 0 \ 1 \ 2 \quad 1 \ 3 \ 4 \\
N-A = ( \begin{array}{ccc} 3 & 2 & 5 \end{array} ) - ( \begin{array}{ccc} 0 & 1 & 3 \end{array} ) = ( \begin{array}{ccc} 3 & 1 & 2 \end{array} ) \\
4 \ 4 \ 8 \quad 1 \ 0 \ 3 \quad 3 \ 4 \ 5 \\
5 \ 1 \ 10 \quad 1 \ 1 \ 0 \quad 4 \ 0 \ 10
\end{array}$$

Y al comprobar si se cumple  $(N_i - A_i) \leq R_D$ , se cumple para  $N_3$ :

$$\begin{array}{l}
(N_3 - A_3) \leq R_D \\
(3 \ 1 \ 2) \leq (3 \ 2 \ 2)
\end{array}$$

Al suponer que termina  $P_3$  tendríamos el estado  $S_2$  del apartado anterior, por lo que ya sabemos que  $S'$  es seguro.

**Así que, aplicando el algoritmo del banquero, el sistema sí puede admitir una petición del proceso  $P_3$  de una instancia del recurso  $R_2$ .**

3. (3 p) Un sistema operativo en colaboración con el hardware gestiona la memoria principal mediante paginación por demanda. Se dispone de 5 marcos de página en la memoria principal para cargar páginas de procesos. El sistema operativo utiliza una estrategia de reemplazamiento de página de tipo global y un algoritmo de reemplazamiento LRU. Se tienen dos procesos A y B. La secuencia de referencias a páginas para el proceso A es:

7 15 17 13 15 11 15 11 9 9

La secuencia de referencias a páginas para el proceso B es:

3 7 8 6 7 5 7 5 5 5

Se supone que la primera página que se carga es la primera del proceso A, después la primera del proceso B, a continuación la segunda de A, después la segunda de B, y así sucesivamente. Además se supone que no llegan nuevos procesos al sistema mientras se ejecutan A y B. Determinar el número de fallos de página que se producen.

Utilizaré la implementación del algoritmo LRU que emplea un contador hardware. Además, para distinguir entre los dos setes, llamaré a uno 7a y al otro 7b:

1. 7a : Fallo de página. Se carga en el primer marco de página.
2. 3 : Fallo de página. Se carga en el segundo marco de página.
3. 15: Fallo de página. Se carga en el tercer marco de página.
4. 7b: Fallo de página. Se carga en el cuarto marco de página.
5. 17: Fallo de página. Se carga en el quinto marco de página.
6. 8: Fallo de página. Se reemplaza la página 7a por tener el campo contador más pequeño (1).
7. 13: Fallo de página. Se reemplaza la página 3 por tener el campo contador más pequeño (2).
8. 6: Fallo de página. Se reemplaza la página 15 por tener el campo contador más pequeño (3).
9. 15: Fallo de página. Se reemplaza la página 7b por tener el campo contador más pequeño (4).
10. 7b: Fallo de página. Se reemplaza la página 17 por tener el campo contador más pequeño (5).
11. 11: Fallo de página. Se reemplaza la página 8 por tener el campo contador más pequeño (6).
12. 5: Fallo de página. Se reemplaza la página 13 por tener el campo contador más pequeño (7).
13. 15: Está cargada (se actualiza el contador).
14. 7b: Está cargada (se actualiza el contador).
15. 11: Está cargada (se actualiza el contador).
16. 5: Está cargada (se actualiza el contador).
17. 9: Fallo de página. Se reemplaza la página 6 por tener el campo contador más pequeño (8).

18. 5: Está cargada (se actualiza el contador).

19. 9: Está cargada (se actualiza el contador).

20. 5: Está cargada (se actualiza el contador).

1	2	3	4	5	6	7	8	9	10
$7a_1$	$7a_1$	$7a_1$	$7a_1$	$7a_1$	$8_6$	$8_6$	$8_6$	$8_6$	$8_6$
	$3_2$	$3_2$	$3_2$	$3_2$	$3_2$	$13_7$	$13_7$	$13_7$	$13_7$
		$15_3$	$15_3$	$15_3$	$15_3$	$15_3$	$6_8$	$6_8$	$6_8$
			$7b_4$	$7b_4$	$7b_4$	$7b_4$	$7b_4$	$15_9$	$15_9$
				$17_5$	$17_5$	$17_5$	$17_5$	$17_5$	$7b_{10}$

11	12	13	14	15	16	17	18	19	20
$11_{11}$	$11_{11}$	$11_{11}$	$11_{11}$	$11_{15}$	$11_{15}$	$11_{15}$	$11_{15}$	$11_{15}$	$11_{15}$
$13_7$	$5_{12}$	$5_{12}$	$5_{12}$	$5_{12}$	$5_{16}$	$5_{16}$	$5_{18}$	$5_{18}$	$5_{20}$
$6_8$	$6_8$	$6_8$	$6_8$	$6_8$	$6_8$	$9_{17}$	$9_{17}$	$9_{19}$	$9_{19}$
$15_9$	$15_9$	$15_{13}$	$15_{13}$	$15_{13}$	$15_{13}$	$15_{13}$	$15_{13}$	$15_{13}$	$15_{13}$
$7b_{10}$	$7b_{10}$	$7b_{10}$	$7b_{14}$						

**Se producen 13 fallos de página.**