

Sistemas operativos

Segunda prueba de Evaluación a Distancia (PED2)

Curso 2016-2017



Centro asociado de Cádiz

Juan Manuel Merino López

31657846-W

Solución ejercicio 1

- I) **Falsa.** Es un algoritmo de detección de interbloqueos, trabaja con una descripción matricial de los recursos existentes, de los recursos disponibles y de las peticiones y asignaciones de recursos.

- II) **Falsa.** La numeración de los sectores de una pista de un disco duro no se realizan de forma contigua sino intercalada en función de un determinado factor de intercalado o entrelazado. El motivo es que el controlador del disco no es suficientemente rápido para poder leer/escribir dos sectores adyacentes en una pista según pasan en una sola vuelta por la cabeza lectura/escritura, pero sí que los puede leer/escribir en una misma vuelta si los dos sectores que se desean leer/escribir están separados por uno o varios sectores que no se van a leer/escribir.

- III) **Falsa.** A la estrategia de buffering de páginas, se le denomina así ya que la lista de marcos libres también hace las veces de un buffer de marcos libres. La lista de marcos libres se utiliza como una memoria caché software de páginas. Así que la estrategia de buffering de páginas no consiste en almacenar en el área de intercambio una copia de un conjunto de páginas de un proceso.

- IV) **Verdadera.** Es una de las ventajas de la segmentación simple. Cada parte de un programa tiene asignado un segmento independiente, la protección y la compartición se realiza a nivel de segmento. Esta es una ventaja importante con respecto a la paginación, donde no se puede asegurar si en una misma página no estarán contenidas diferentes partes (código y datos) del programa.

Solución ejercicio 2

- a) Para calcular x , y y z , instancias de los recursos R_1 , R_2 y R_3 respectivamente, habría que sumar, según los datos del enunciado, los recursos asignados y los recursos disponibles.

$$R_E = R_D + R_A = (3 \ 3 \ 2) + (3 \ 2 \ 8) = (6 \ 5 \ 10)$$

Luego:

$$(x \ y \ z) = (6 \ 5 \ 10)$$

- b) Para saber si en un cierto instante T , llamemos S_1 es seguro o no, hay que comprobar si existe algún proceso P_i que cumpla la condición siguiente:

$$(N_i - A_i) \leq R_D \quad i = 1, 2, 3, 4, 5$$

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

$$N - A = \begin{pmatrix} 4 & 4 & 4 \\ 1 & 4 & 6 \\ 3 & 2 & 5 \\ 4 & 4 & 8 \\ 5 & 1 & 10 \end{pmatrix} - \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 2 \\ 0 & 0 & 3 \\ 1 & 0 & 3 \\ 1 & 1 & 0 \end{pmatrix} = \begin{pmatrix} 3 & 4 & 4 \\ 1 & 3 & 4 \\ 3 & 2 & 2 \\ 3 & 4 & 5 \\ 4 & 0 & 10 \end{pmatrix}$$

La primera fila, asociada al P_1 , no cumple que cada uno de sus elementos es menor o igual que los elementos disponibles:

$$(3 \ 4 \ 4) \leq (3 \ 3 \ 2)$$

La segunda fila, asociada al P_2 , no lo cumple:

$$(1 \ 3 \ 4) \leq (3 \ 3 \ 2)$$

La tercera fila, asociada al P_3 , si lo cumple:

$$(3 \ 2 \ 2) \leq (3 \ 3 \ 2)$$

La cuarta fila, asociada al P_4 , tampoco lo cumple:

$$(3 \ 4 \ 5) \leq (3 \ 3 \ 2)$$

La quinta fila asociada al proceso P_5 , no lo cumple:

$$(4 \ 0 \ 10) \leq (3 \ 3 \ 2)$$

Luego vemos que al proceso P_3 se le puede conceder todos los recursos que necesita para completarse aunque los solicite todos a la vez. Supongamos que P_3 se ha completado, el estado pasa a ser S_2 :

$$S_2 = \left\{ N = \begin{pmatrix} 4 & 4 & 4 \\ 1 & 4 & 6 \\ 0 & 0 & 0 \\ 4 & 4 & 8 \\ 5 & 1 & 10 \end{pmatrix} A = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 2 \\ 0 & 0 & 0 \\ 1 & 0 & 3 \\ 1 & 1 & 0 \end{pmatrix} R_E = (6 \ 5 \ 10) R_D = (3 \ 3 \ 5) \right\}$$

El número de instancias de cada recurso que todavía necesita cada proceso se obtiene restando la matriz N y la matriz A, y comprobando si cumplen con la siguiente condición:

$$(N_i - A_i) \leq R_D$$

$$N - A = \begin{pmatrix} 4 & 4 & 4 \\ 1 & 4 & 6 \\ 0 & 0 & 0 \\ 4 & 4 & 8 \\ 5 & 1 & 10 \end{pmatrix} - \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 2 \\ 0 & 0 & 0 \\ 1 & 0 & 3 \\ 1 & 1 & 0 \end{pmatrix} = \begin{pmatrix} 3 & 4 & 4 \\ 1 & 3 & 4 \\ 0 & 0 & 0 \\ 3 & 4 & 5 \\ 4 & 0 & 10 \end{pmatrix}$$

La fila uno, P_1 , no lo cumple:

$$(3 \ 4 \ 4) \leq (3 \ 3 \ 5)$$

La fila dos, P_2 , si lo cumple:

$$(1 \ 3 \ 4) \leq (3 \ 3 \ 5)$$

La fila cuatro, P_4 , no lo cumple:

$$(3 \ 4 \ 5) \leq (3 \ 3 \ 5)$$

La fila cinco, P_5 , no lo cumple:

$$(4 \ 0 \ 10) \leq (3 \ 3 \ 5)$$

Se observa que la fila 2, asociada al P_2 , si cumple, luego se puede asignar a ese proceso todos los recursos que necesita para completarse. Si P_2 se completa, el estado del sistema pasa a ser S_3 :

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

Se vuelve a comprobar en este estado el número de instancias de cada recurso que todavía necesita, restando N-A, y comprobando de nuevo que se cumple:

$$(N_i - A_i) \leq R_D \quad i = 1, 4, 5$$

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

La fila uno, si lo cumple:

$$(3 \quad 4 \quad 4) \leq (3 \quad 4 \quad 7)$$

La fila cuatro, si lo cumple:

$$(3 \quad 4 \quad 5) \leq (3 \quad 4 \quad 7)$$

La fila cinco, no lo cumple:

$$(4 \quad 0 \quad 10) \leq (3 \quad 4 \quad 7)$$

Tanto el proceso P_1 como el P_4 cumplen la condición, se puede ver que tanto si se completa uno u otro dejaría recursos libres suficientes para que P_5 se complete. Supongamos que se completa primero P_1 , entonces el estado pasaría a ser S_4 :

$$S_4 = \left\{ N = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 4 & 4 & 8 \\ 5 & 1 & 10 \end{pmatrix} A = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 3 \\ 1 & 1 & 0 \end{pmatrix} R_E = (6 \quad 5 \quad 10) R_D = (4 \quad 4 \quad 7) \right\}$$

Se vuelve a comprobar en este estado el número de instancias de cada recurso que todavía necesita, restando N-A, y comprobando de nuevo que se cumple:

$$(N_i - A_i) \leq R_D \quad i = 4, 5$$

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

Observamos que la fila cuatro, si cumple la condición:

$$(3 \quad 4 \quad 5) \leq (4 \quad 4 \quad 7)$$

Así que se completa al concederle todos los recursos que necesita para completarse aunque los solicite todos a la vez. Al completarse P_4 , el estado pasa a ser, S_5 :

$$S_5 = \left\{ N = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 5 & 1 & 10 \end{pmatrix} A = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 0 \end{pmatrix} R_E = (6 \ 5 \ 10) R_D = (5 \ 4 \ 10) \right\}$$

Se vuelve a comprobar en este estado de nuevo que se cumple:

$$(N_i - A_i) \leq R_D \quad i = 5$$

A simple vista, restando la matriz N y la matriz A , podemos comprobar que se cumple la condición para el proceso P_5 :

$$(4 \ 0 \ 10) \leq (5 \ 4 \ 10)$$

Luego se puede asignar a este proceso todos los recursos que necesita para completarse.

En definitiva se ha demostrado que el estado S_1 es un estado seguro, ya que a partir de él es posible completar la ejecución de todos los procesos.

- c) Para aplicar el algoritmo del banquero, se parte del estado S_1 , y se simula que se concede la petición en este caso una petición del proceso P_3 de una instancia del recurso R_2 y se actualiza al estado del sistema, S_1' :

$$S_1' = \left\{ N = \begin{pmatrix} 4 & 4 & 4 \\ 1 & 4 & 6 \\ 3 & 2 & 5 \\ 4 & 4 & 8 \\ 5 & 1 & 10 \end{pmatrix} A = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 2 \\ 0 & 1 & 3 \\ 1 & 0 & 3 \\ 1 & 1 & 0 \end{pmatrix} R_E = (6 \ 5 \ 10) R_D = (3 \ 3 \ 2) \right\}$$

Para saber si este nuevo estado S_1' es seguro, se comprueba lo siguiente:

$$(N_i - A_i) \leq R_D \quad i = 1, 2, 3, 4, 5$$

$$N - A = \begin{pmatrix} 4 & 4 & 4 \\ 1 & 4 & 6 \\ 3 & 2 & 5 \\ 4 & 4 & 8 \\ 5 & 1 & 10 \end{pmatrix} - \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 2 \\ 0 & 1 & 3 \\ 1 & 0 & 3 \\ 1 & 1 & 0 \end{pmatrix} = \begin{pmatrix} 3 & 4 & 4 \\ 1 & 3 & 4 \\ 3 & 1 & 2 \\ 3 & 4 & 5 \\ 4 & 0 & 10 \end{pmatrix}$$

La primera fila, asociada al P_1 , no cumple que cada uno de sus elementos es menor o igual que los elementos disponibles:

$$(3 \ 4 \ 4) \leq (3 \ 3 \ 2)$$

La segunda fila, asociada al P_2 , no lo cumple:

$$(1 \ 3 \ 4) \leq (3 \ 3 \ 2)$$

La tercera fila, asociada al P_3 , si lo cumple:

$$(3 \ 1 \ 2) \leq (3 \ 3 \ 2)$$

La cuarta fila, asociada al P_4 , tampoco lo cumple:

$$(3 \ 4 \ 5) \leq (3 \ 3 \ 2)$$

La quinta fila asociada al proceso P_5 , no lo cumple:

$$(4 \ 0 \ 10) \leq (3 \ 3 \ 2)$$

Luego vemos que al proceso P_3 se le puede conceder todos los recursos que necesita para completarse aunque los solicite todos a la vez. Supongamos que P_3 se ha completado, el estado pasa a ser S_2 , que sería el mismo caso que se vió en el apartado **b)** de este mismo ejercicio, por lo que vemos que **SI** se admitiría la petición del proceso P_3 de una instancia del recurso R_2 .

Solución Ejercicio 3

Utilizando una estrategia de reemplazamiento de páginas global y un algoritmo de reemplazamiento LRU, donde es candidata a ser reemplazada la página que lleva más tiempo sin ser referenciada, para los procesos A y B, quedaría de la siguiente forma:

Una cosa a tener en cuenta es que se repite la página 7 en cada proceso, pero obviamente son páginas distintas, así que si está cargada la página 7 del proceso A, si se llamara a la 7 del proceso B, evidentemente habría un fallo.

Para implementar el algoritmo LRU se utiliza una lista enlazada que se gestiona como una pila.

Las primeras 5 referencias producen fallo ya que los marcos están inicialmente vacíos además de ser todas referencias distintas. Los números de página se colocan en la pila y la página se carga en el marco correspondiente empezando desde el marco 1 al marco 5. La sexta referencia (página 8 del proceso B), también produce un fallo de página. La página que se sustituye es aquella cuyo número aparece en la base de la pila, en este caso la 7 del proceso A, ya que hace más tiempo que fue referenciada. El número de página 3 del proceso B se coloca en la cima de la pila y el número 7 desaparece de la pila. La página 8 del proceso B es cargada en el marco que contenía la página 7 del proceso A, es decir, el marco 1.

Procediendo de forma análoga se obtiene que la secuencia de referencias indicada en este caso se puede observar en la siguiente figura, donde A, son los aciertos y F son los fallos. El número de fallos de página que se producen son **13 fallos**, como se puede ver.

	7	3	15	7	17	8	13	6	15	7	11	5	15	7	11	5	9	5	9	5	
					17	8	13	6	15	7	11	5	15	7	11	5	9	5	9	5	Pila
				7	7	17	8	13	6	15	7	11	11	11	11	11	5	5	5	5	
			15	15	15	7	17	8	13	6	15	7	7	7	7	7	11	11	11	11	
		3	3	3	3	15	7	17	8	13	6	15	15	15	15	15	7	7	7	7	
	7	7	7	7	7	3	15	7	17	8	13	6	6	6	6	6	15	15	15	15	
	7	7	7	7	7	8	8	8	8	8	11	11	11	11	11	11	11	11	11	11	Marco 1
		3	3	3	3	3	13	13	13	13	13	5	5	5	5	5	5	5	5	5	Marco 2
			15	15	15	15	15	6	6	6	6	6	6	6	6	6	9	9	9	9	Marco 3
				7	7	7	7	7	15	15	15	15	15	15	15	15	15	15	15	15	Marco 4
					17	17	17	17	17	7	7	7	7	7	7	7	7	7	7	7	Marco 5
	F	F	F	F	F	F	F	F	F	F	F	F	A	A	A	A	F	A	A	A	

Figura aplicación del algoritmo del ejercicio 3.