

SISTEMAS OPERATIVOS

Segunda Prueba de Evaluación a Distancia (PED 2)

Alumno: Oscar Rubio Fauria

Centro Asociado: Cádiz

Curso: 2014 – 15

Ejercicio 1.

I) **FALSA.**

(pg 239 del libro base)

Dentro del apartado 6.6.7. Ventajas e inconvenientes de la Paginación Simple

“La principal desventaja del uso de la paginación simple es que produce fragmentación interna. El espacio lógico de direcciones de un proceso se descompone en páginas de un determinado tamaño, normalmente el resultado de dividir el tamaño del espacio lógico entre el tamaño de páginas no da como resultado un número entero, sino uno decimal. Esto significa que la parte final del espacio de direcciones del proceso se ubicará en una página que no ocupará por completo. Es decir, suele suceder que se desaprovecha parte del espacio de la última página asignada al proceso. Por lo tanto, en el peor de los casos la fragmentación interna será casi una página. Puede demostrarse que en promedio la fragmentación interna es del orden de media página por proceso.”

En consecuencia cuanto mayor sea el tamaño de página utilizado mas grande será la fragmentación interna, por tanto la afirmación es falsa.

II) **FALSA.**

(pg 323 del libro base) Definición de ECC

“Código de corrección de errores (Error Correcting Code, ECC). Es un elemento que forma parte de la estructura de un sector y que contiene información que permite al controlador del disco detectar y corregir posibles errores de lectura.”

(pg 327 del libro base, primer parrafo)

“Para identificar si un sector está dañado, el controlador utiliza el ECC del sector. Cuando el controlador del disco realiza una operación de escritura sobre un sector, el ECC del sector se calcula usando una función de comprobación de redundancia ciclica (CRC) que usa como parámetro de entrada todos los bytes almacenados en el área de datos del sector. Cuando se requiere la lectura de dicho sector en una operación de lectura, el controlador calcula de nuevo el ECC y lo compara con el valor de ECC almacenado en el sector. Si el nuevo ECC difiere del almacenado, entonces significa que el sector esta dañado.”

En conclusión del libro base se entiende que cada sector de una pista tiene su propio ECC y cada uno debe ser diferente, ya que si no fuera así cuando uno estuviera dañado, lo estarían todos. Por tanto, la afirmación es falsa.

III) **FALSA.**

(pg 227 del libro base) 6.6.2. Traducción de direcciones

“En la técnica de gestión de memoria mediante paginación, la traducción de direcciones lógicas a direcciones físicas requiere que el sistema operativo cargue en el hardware cierta información sobre la tabla de páginas del proceso actualmente en ejecución.

La información que carga depende de los componentes hardware disponibles para realizar la traducción de direcciones:

- Registro base. Se utiliza para almacenar la dirección física de comienzo de la tabla de páginas.
- Banco de registros. Se utiliza para almacenar una copia completa de la tabla de páginas.
- Buffer de traducción de vista lateral (Translation Lookaside Buffer, TLB). Se utiliza para almacenar algunas entradas de la tabla de páginas.”

Por consiguiente al ser necesario almacenar en hardware cierta información sobre la tabla de paginas la afirmacion es falsa.

IV) FALSA.

(pg 175 del libro base)Ultimo parrafo dentro del apartado 5.5.1 Estado de la asignacion de recursos: estados seguros y estados inseguros.

“Un estado se dice que es seguro si posibilita al menos una secuencia de asignación de recursos a los procesos que garantiza que éstos se pueden ejecutar hasta su finalización sin que se produzca interbloqueo, incluso aunque los procesos soliciten simultáneamente todos los recursos que van a necesitar. Un estado inseguro es aquél que no garantiza que no se pueda producir un interbloqueo.”

La parte subrayada contradice la afirmación, por consiguiente es falsa.

Ejercicio 2.

a)

Proceso	Instancias asignadas	Instancias máximas necesitadas
A	2	5
B	1	2
C	6	8

La **matriz N de recursos máximos necesitados por cada proceso**. Esta matriz tiene 3 filas, una asociada a cada proceso existente en el sistema. Además posee 1 columna, ya que solo existe un tipo de recurso. El valor de cada elemento, es el número de instancias máximas necesitadas por cada proceso del recurso.

$$\mathbf{N} = \begin{pmatrix} 5 \\ 2 \\ 8 \end{pmatrix} \begin{array}{l} \text{----- las instancias máximas necesitadas por el proceso A} \\ \text{----- las instancias máximas necesitadas por el proceso B} \\ \text{----- las instancias máximas necesitadas por el proceso C} \end{array}$$

La **matriz A de recursos asignados a cada proceso**. Esta matriz tiene 3 filas, una asociada a cada proceso existente. Además posee 1 columna, ya que solo existe un tipo de recurso.

$$A = \begin{pmatrix} 2 \\ 1 \\ 6 \end{pmatrix} \begin{array}{l} \text{----- las instancias asignadas del recurso por el proceso A} \\ \text{----- las instancias asignadas del recurso por el proceso B} \\ \text{----- las instancias asignadas del recurso por el proceso C} \end{array}$$

El **vector Re de recursos existentes**. El sistema posee un tipo de recurso. Contiene el número de instancias existentes del recurso.

$$\mathbf{Re} = (10)$$

El **vector Ra de instancias de recursos asignados**. Es el número total de instancias asignadas del recurso. La suma de todas las instancias asignadas de los 3 procesos existentes. Se suman las filas de una misma columna de la matriz A en este caso solo hay una columna porque solo hay un recurso.

$$\mathbf{Ra} = (2+1+6) = (9)$$

El **vector Rd de recursos disponibles** se obtiene restando el vector de recursos existentes menos el vector de recursos asignados.

$$\mathbf{Rd} = \mathbf{Re} - \mathbf{Ra} = (10) - (9) = (1)$$

b)

Se va denotar el instante en el momento T como S1:

$$S1 = \left\{ N = \begin{pmatrix} 5 \\ 2 \\ 8 \end{pmatrix} ; A = \begin{pmatrix} 2 \\ 1 \\ 6 \end{pmatrix} ; Re = (10) ; Ra = (9) ; Rd = (1) \right\}$$

En este caso el número de instancias de cada recurso que todavía necesita cada proceso se obtiene restando la matriz N y la matriz A.

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

Para saber si este estado S1 es seguro o no, simplemente hay que comprobar si existe alguna fila i de $N - A$, es decir, algún proceso P_i que cumpla la condición:

$$(N_i - A_i) \leq R_d \quad i = 1, 2, 3$$

Fila 1, Proceso A: $(3) \leq (1)$ (no cumple la condición)

Fila 2, Proceso B: $(1) \leq (1)$ (cumple la condición)

Fila 3, Proceso C: $(2) \leq (1)$ (no cumple la condición)

Al proceso B se le pueden conceder todos los recursos que necesita para completarse aunque los solicite todos a la vez. Supóngase que el proceso B se ha completado, el estado del sistema pasa a ser S2:

$$S2 = \left\{ N = \begin{pmatrix} 5 \\ 0 \\ 8 \end{pmatrix} ; A = \begin{pmatrix} 2 \\ 0 \\ 6 \end{pmatrix} ; Re = (10) ; Ra = (8) ; Rd = (2) \right\}$$

Como el proceso B se ha completado, la fila 2 de las matrices N y A son nulas.

El número de instancias de cada recurso que todavía necesita cada proceso se obtiene restando la matriz N y la matriz A.

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

Comprobar si existe alguna fila i de $N - A$, es decir, algún proceso P_i que cumpla la condición:

$$(N_i - A_i) \leq R_d \quad i=1,2,3$$

Fila 1, Proceso A: $(3) \leq (2)$ (no cumple la condición)

Fila 2 es nula

Fila 3, Proceso C: $(2) \leq (2)$ (cumple la condición)

Se observa que la tercera fila asociada al proceso C cumple la condición. Se pueden asignar al proceso C todos los recursos que necesita para completarse. Suponemos que el proceso C se ha completado, el estado del sistema pasa a ser S3:

$$S3 = \left\{ N = \begin{pmatrix} 5 \\ 0 \\ 0 \end{pmatrix} ; A = \begin{pmatrix} 2 \\ 0 \\ 0 \end{pmatrix} ; Re = (10) ; Ra = (2) ; Rd = (8) \right\}$$

Como el proceso C se ha completado, la fila 3 de las matrices N y A son nulas.

El número de instancias de cada recurso que todavía necesita cada proceso se obtiene restando la matriz N y la matriz A.

$$N - A = \begin{pmatrix} 5 \\ 0 \\ 0 \end{pmatrix} - \begin{pmatrix} 2 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 3 \\ 0 \\ 0 \end{pmatrix}$$

Comprobar si existe alguna fila i de $N - A$, es decir, algún proceso P_i que cumpla la condición:

$$(N_i - A_i) \leq R_d \quad i=1,2,3$$

Fila 1, Proceso A: $(3) \leq (8)$ (cumple la condición)

Fila 2 es nula

Fila 3 es nula

Se observa que la primera fila asociada al proceso A cumple la condición. Se pueden asignar al proceso A todos los recursos que necesita para completarse.

El estado S1 en el instante T es seguro, ya que hemos demostrado que a partir de él es posible completar la ejecución de todos los procesos.

c)

El número de instancias asignadas del recurso para el proceso A pasaría de ser 2 a ser 3

Proceso	Instancias asignadas	Instancias máximas necesitadas
A	3	5
B	1	2
C	6	8

Se va denotar el instante en el momento T como S1:

$$S1 = \left\{ N = \begin{pmatrix} 5 \\ 2 \\ 8 \end{pmatrix} ; A = \begin{pmatrix} 3 \\ 1 \\ 6 \end{pmatrix} ; R_e = (10) ; R_a = (10) ; R_d = (0) \right\}$$

En este caso el número de instancias de cada recurso que todavía necesita cada proceso se obtiene restando la matriz N y la matriz A.

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

Para saber si este estado S1 es seguro o no, simplemente hay que comprobar si existe alguna fila i de $N - A$, es decir, algún proceso P_i que cumpla la condición:

$$(N_i - A_i) \leq R_d \quad i = 1,2,3$$

Fila 1, Proceso A: $(2) \leq (0)$ (no cumple la condición)

Fila 2, Proceso B: $(1) \leq (0)$ (no cumple la condición)

Fila 3, Proceso C: $(2) \leq (0)$ (no cumple la condición)

Como ninguna de las filas cumple la condición, **el estado S1 en el instante T es un estado inseguro.**

Ejercicio 3

a)

(pag 279 – 280) 7.4.4. Algoritmo de reemplazamiento FIFO

“Para implementar este algoritmo el sistema operativo debe mantener una lista enlazada o cola FIFO. Cada entrada de la lista contiene el número de página de una página del conjunto de páginas candidatas a ser reemplazadas. Cuando una página se carga en memoria su número de página se coloca al final de la lista. Así, el número de página que se encuentra al principio de la lista indica la página que lleva más tiempo cargada en memoria y será la elegida para ser reemplazada. Cuando una página k que ocupaba el marco j es reemplazada por una página i entonces su número se elimina del principio de la lista y se coloca al final el número de la página que ha sido cargada en el marco j reemplazando a la página k.”

Posibles situaciones:

1. Si los marcos están vacíos produce un fallo de página, el número de página se coloca en la cola FIFO y la página se carga en el primer marco que es el j.
2. Si los marcos están parcialmente llenos, si el número de página coincide con alguno de los que están en el marco produce un acierto y no se modifica nada.
3. Si los marcos están parcialmente llenos, si el número de página no coincide con ninguno de los que están en el marco se produce un fallo de página, se introduce el número de página en la cola FIFO y la página se carga en el primer marco libre.
4. Si los marcos están todos ocupados, en caso de que el número de página no coincida con ninguno de los ocupados por los marcos se produce un fallo de página, se introduce el número de página en la cola FIFO y se elimina de la cola el número que más tiempo lleva. Además, el marco que contenía ese número se ve modificado y almacena el número de página que ha producido el fallo.
5. Si los marcos están todos ocupados, en caso de que el número de página coincida con alguno de los ocupados por los marcos se produce un acierto y no se modifica nada.

Cadena de referencias de páginas que produce la ejecución del proceso X es:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
4	8	9	7	8	6	7	8	6	5	8	6	5	4	5	6	5	6	4	6	4

Pila:

		9	7	7	6	6	8	8	5	5	5	5	4	4	6	6	6	6	6	6
	8	8	9	9	7	7	6	6	8	8	8	8	5	5	4	4	4	4	4	4
4	4	4	8	8	9	9	7	7	6	6	6	6	8	8	5	5	5	5	5	5

Resultado: (F: Fallo; A: Acierto)

F	F	F	F	A	F	A	F	A	F	A	A	A	F	A	F	A	A	A	A	A
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Marco j:

4	4	4	7	7	7	7	7	7	5	5	5	5	5	5	5	5	5	5	5	5
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Marco k:

	8	8	8	8	6	6	6	6	6	6	6	6	4	4	4	4	4	4	4	4
--	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Marco l:

		9	9	9	9	9	8	8	8	8	8	8	8	8	6	6	6	6	6	6
--	--	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

9 FALLOS y 12 ACIERTOS

b)

(pag 275-277) 7.4.2. Algoritmo de reemplazamiento LRU

“Una posible implementación del algoritmo LRU es mediante el uso de una lista enlazada que se gestiona como una pila. Cada entrada de la lista contiene el número de página de una página del conjunto de páginas candidatas a ser reemplazadas. Cada vez que una página es referenciada su número es colocado al principio de la lista. Es decir, la lista funciona como una pila. Así al principio de la lista se coloca el número de página referenciado mas recientemente. Mientras que al final de la lista se coloca el número de página referenciado menos recientemente, es decir, la candidata para ser sustituida. Cada vez que el procesador referencia a una direccion virtual, la lista debe ser reordenada, lo cual consume un tiempo importante.”

Posibles situaciones: (son las mismas que el algoritmo FIFO pero con una variante en la situación 2 y 5)

1. Si los marcos están vacios produce un fallo de pagina, el numero de pagina se coloca en la cola y la pagina se carga en el primer marco que es el j.
2. Si los marcos están parcialmente llenos, si el numero de pagina coincide con alguno de los que están en el marco produce un acierto y la pila se reordena de tal manera que el numero de pagina es colocado al principio de la lista.
3. Si los marcos están parcialmente llenos, si el numero de pagina no coincide con ninguno de los que están en el marco se produce un fallo de pagina, se introduce el numero de pagina en la cola y la pagina se carga en el primer marco libre.
4. Si los marcos están todos ocupados, en caso de que el numero de pagina no coincida con ninguno de los ocupados por los marcos se produce un fallo de pagina, se introduce el numero de pagina en la cola y se elimina de la cola el numero que mas tiempo lleva. Además, el marco que contenía ese numero se ve modificado y almacena el numero de pagina que ha producido el fallo.

5. Si los marcos están todos ocupados, en caso de que el número de página coincida con alguno de los ocupados por los marcos se produce un acierto y la pila se reordena de tal manera que el número de página es colocado al principio de la lista.

Cadena de referencias de páginas que produce la ejecución del proceso X es:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
4	8	9	7	8	6	7	8	6	5	8	6	5	4	5	6	5	6	4	6	4

Pila:

		9	7	8	6	7	8	6	5	8	6	5	4	5	6	5	6	4	6	4
	8	8	9	7	8	6	7	8	6	5	8	6	5	4	5	6	5	6	4	6
4	4	4	8	9	7	8	6	7	8	6	5	8	6	6	4	4	4	5	5	5

Resultado: (F: Fallo; A: Acierto)

F	F	F	F	A	F	A	A	A	F	A	A	A	F	A	A	A	A	A	A	A
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Marco j:

4	4	4	7	7	7	7	7	7	5	5	5	5	5	5	5	5	5	5	5	5
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Marco k:

	8	8	8	8	8	8	8	8	8	8	8	8	4	4	4	4	4	4	4	4
--	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Marco l:

		9	9	9	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6
--	--	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

7 FALLOS y 14 ACIERTOS