

SISTEMAS OPERATIVOS

Segunda Prueba de Evaluación a Distancia (PED 2)

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

I) Cuanto mayor sea el tamaño de página utilizado más pequeña será en promedio la fragmentación interna.

Cuanto mayor sea el tamaño de página es más probable que existan procesos con tamaños del espacio de direcciones inferiores al tamaño establecido, provocando así una mayor fragmentación interna. Por tanto, la afirmación es FALSA.

II) El contenido del ECC de un sector de disco es siempre el mismo para todos los sectores de una misma pista.

El contenido del ECC de un sector depende del contenido del área de datos de dicho sector. Por tanto, la afirmación es FALSA.

III) En la técnica de gestión de memoria mediante paginación simple la traducción de direcciones lógicas a direcciones físicas es realizada por el sistema operativo sin contar con ningún apoyo del hardware.

La traducción de direcciones lógicas a direcciones físicas requiere que el sistema operativo cargue en el hardware cierta información sobre la tabla de páginas. Tanto la información cargada como el proceso de traducción dependen de los componentes hardware disponibles. Por tanto, la afirmación es FALSA.

IV) Si el sistema operativo se encuentra en un estado inseguro con respecto a la asignación de recursos entonces está garantizado que se va a producir un interbloqueo.

Un estado inseguro es aquel que no garantiza que no se pueda cumplir un interbloqueo, así que aunque el sistema operativo se encuentre en un estado inseguro, se puede dar la situación de que no se produzca un interbloqueo. Por tanto, la afirmación es FALSA.

2.- Considérense los procesos A, B y C que comparten un recurso del que existen 10 instancias. En la siguiente tabla se muestra el número de instancias asignadas y el número máximo de instancias necesitadas por cada proceso en un cierto instante de tiempo T.

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

Se pide:

- a) Determinar para el instante T la matriz N de recursos máximos necesitados por cada proceso, la matriz A de recursos asignados a cada proceso, el vector R_E de recursos existentes y el vector R_D de recursos disponibles.

Dado que nuestros procesos solo comparten un recurso, las matrices dispondrán de una única columna. Así, las matrices N y A son:

$$N = \begin{pmatrix} 5 \\ 2 \\ 8 \end{pmatrix} \quad y \quad A = \begin{pmatrix} 2 \\ 1 \\ 6 \end{pmatrix}$$

Sumando los elementos de una misma columna de la matriz A se obtiene el vector R_A que indica el número de instancias de cada tipo de recurso asignadas. Como los procesos A, B y C solamente comparten un recurso, R_A tiene un solo elemento:

$$R_A = (9)$$

El vector R_D de recursos disponibles se obtiene restando el vector R_E de recursos existentes menos el vector R_A de recursos asignados. Como del único recurso que comparten los procesos A, B y C existen 10 instancias, entonces:

$$R_E = (10)$$

$$R_D = R_E - R_A = (1)$$

- b) Haciendo uso de las matrices y vectores obtenidos en el apartado anterior determinar razonadamente si el estado del sistema en ese instante T es seguro.

Sea S el estado del sistema en el instante T, cuyas características se encuentran recogidas en N, A, R_A , R_E y R_D . El número de instancias de cada recurso que todavía necesita cada recurso se obtiene calculando la matriz N-A.

$$S = \left\{ N = \begin{pmatrix} 5 \\ 2 \\ 8 \end{pmatrix}; A = \begin{pmatrix} 2 \\ 1 \\ 6 \end{pmatrix}; R_E = (10); R_D = (1) \right\}$$

Para saber si S es seguro o no, basta con comprobar si existe alguna fila de N-A que cumpla que cada uno de sus elementos es menor o igual que los elementos de R_D .

$$N - A = \begin{pmatrix} 3 \\ 1 \\ 2 \end{pmatrix} \quad y \quad R_D = (1)$$

Como se puede apreciar, solo la segunda fila cumple la condición exigida. Por ello, al proceso B se le pueden conceder todos los recursos que necesita para completarse. Si suponemos que el proceso B se ha completado, el estado del sistema pasa a ser S' :

$$S' = \left\{ N = \begin{pmatrix} 5 \\ 0 \\ 8 \end{pmatrix}; A = \begin{pmatrix} 2 \\ 0 \\ 6 \end{pmatrix}; R_E = (10); R_D = (2) \right\}$$

Comprobamos si este nuevo estado cumple la condición de ser seguro:

$$N - A = \begin{pmatrix} 3 \\ 0 \\ 2 \end{pmatrix} \quad y \quad R_D = (2)$$

La tercera fila cumple la condición exigida. Por ello, al proceso C se le pueden conceder todos los recursos que necesita para completarse. Si suponemos que el proceso C se ha completado, el estado del sistema pasa a ser S'' :

$$S'' = \left\{ N = \begin{pmatrix} 5 \\ 0 \\ 0 \end{pmatrix}; A = \begin{pmatrix} 2 \\ 0 \\ 0 \end{pmatrix}; R_E = (10); R_D = (8) \right\}$$

Comprobamos si este nuevo estado cumple la condición de ser seguro:

$$N - A = \begin{pmatrix} 3 \\ 0 \\ 0 \end{pmatrix} \quad y \quad R_D = (8)$$

En efecto, ahora la primera fila también cumple la condición. Por ello, al proceso A se le pueden conceder todos los recursos que necesita para completarse.

Por tanto, se ha demostrado que el estado S es un **estado seguro**, ya que a partir de él es posible completar la ejecución de todos los procesos.

- c) **Supóngase que en dicho instante T el proceso A solicita otra instancia del recurso. Determinar si el sistema quedaría en un estado seguro si se atiende la petición del proceso A.**

En este caso, el estado inicial S quedaría definido de la siguiente forma:

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

Al no quedar ningún recurso disponible, podemos afirmar que S es un **estado inseguro**.

3.- Considérese un sistema con memoria virtual en el que se utiliza la técnica de paginación por demanda. En este sistema se han asignado a un cierto proceso X tres marcos de página para su ejecución. La cadena de referencias de página que produce la ejecución del proceso X es:

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

Determinar el número de fallos de página que se producen para los siguientes casos:

a) Se utiliza el algoritmo de reemplazamiento de páginas FIFO.

NOTA: Para implementar este algoritmo voy a utilizar una cola FIFO.

Sea i la **página referenciada** y sean j , k y l los **marcos de página**.

La primera referencia ($i=4$) produce un **fallo de página**, ya que inicialmente los marcos de página están vacíos. El número de página se coloca en la cola FIFO y la página se carga en el marco j .

La segunda referencia ($i=8$) produce un **fallo de página**. El número de página se coloca en la cola FIFO y la página se carga en el marco k .

La tercera referencia ($i=9$) produce un **fallo de página**. El número de página se coloca en la cola FIFO y la página se carga en el marco l .

La cuarta referencia ($i=7$) produce un **fallo de página**. La página que se sustituye es aquella cuyo número aparece al principio de la cola, en este caso $i=4$, ya que es la que lleva más tiempo cargada en la memoria. Así, el número de página $i=7$ se coloca al final de la cola y el número $i=4$ se elimina de la cola. Además, la página $i=7$ es cargada en el marco j , que es el que contenía a la página $i=4$.

La quinta referencia ($i=8$) produce un **acierto de página**, por lo que no hay que hacer ninguna modificación de ningún tipo.

La sexta referencia ($i=6$) produce un **fallo de página**. Procediendo de forma análoga a la explicada en la cuarta referencia, el número de página $i=6$ se coloca al final de la cola y el número $i=8$ se elimina de la cola. Además, la página $i=6$ es cargada en el marco k , que es el que contenía a la página $i=8$.

La séptima referencia ($i=7$) produce un **acierto de página**.

La octava referencia ($i=8$) produce un **fallo de página**. El número de página $i=8$ se coloca al final de la cola y el número $i=9$ se elimina de la cola. Además, la página $i=8$ es cargada en el marco l , que es el que contenía a la página $i=9$.

La novena referencia ($i=6$) produce un **acierto de página**.

La décima referencia ($i=5$) produce un **fallo de página**. El número de página $i=5$ se coloca al final de la cola y el número $i=7$ se elimina de la cola. Además, la página $i=5$ es cargada en el marco j , que es el que contenía a la página $i=7$.

La undécima referencia ($i=8$) produce un **acierto de página**.

La duodécima referencia ($i=6$) produce un **acierto de página**.

La decimotercera referencia ($i=5$) produce un **acierto de página**.

La decimocuarta referencia ($i=4$) produce un **fallo de página**. El número de página $i=4$ se coloca al final de la cola y el número $i=6$ se elimina de la cola. Además, la página $i=4$ es cargada en el marco k , que es el que contenía a la página $i=6$.

La decimoquinta referencia ($i=5$) produce un **acierto de página**.

La decimosexta referencia ($i=6$) produce un **fallo de página**. El número de página $i=6$ se coloca al final de la cola y el número $i=8$ se elimina de la cola. Además, la página $i=6$ es cargada en el marco l , que es el que contenía a la página $i=8$.

La decimoséptima referencia ($i=5$) produce un **acierto de página**.

La decimoctava referencia ($i=6$) produce un **acierto de página**.

La decimonovena referencia ($i=4$) produce un **acierto de página**.

La vigésima referencia ($i=6$) produce un **acierto de página**.

La vigesimoprimera referencia ($i=4$) produce un **acierto de página**.

Por tanto, utilizando este algoritmo de reemplazamiento de páginas se obtienen **9 fallos de página**.

b) Se utiliza el algoritmo de reemplazamiento de páginas LRU.

NOTA: Para implementar este algoritmo voy a utilizar una pila.

Sea i la **página referenciada** y sean j , k y l los **marcos de página**.

La primera referencia ($i=4$) produce un **fallo de página**, ya que inicialmente los marcos de página están vacíos. El número de página se coloca en la pila y la página se carga en el marco j .

La segunda referencia ($i=8$) produce un **fallo de página**. El número de página se coloca en la cima de la pila y la página se carga en el marco k .

La tercera referencia ($i=9$) produce un **fallo de página**. El número de página se coloca en la cima de la pila y la página se carga en el marco l .

La cuarta referencia ($i=7$) produce un **fallo de página**. La página que se sustituye es aquella cuyo número aparece en la base de la pila, en este caso $i=4$, ya que es la que hace más tiempo que fue referenciada. Así, el número de página $i=7$ se coloca en la

cima de la pila y el número $i=4$ se elimina de la pila. Además, la página $i=7$ es cargada en el marco j , que es el que contenía a la página $i=4$.

La quinta referencia ($i=8$) produce un **acierto de página**. La pila se reordena para colocar $i=8$ en la cima de la pila.

La sexta referencia ($i=6$) produce un **fallo de página**. Procediendo de forma análoga a la explicada en la cuarta referencia, el número de página $i=6$ se coloca en la cima de la pila y el número $i=9$ se elimina de la pila. Además, la página $i=6$ es cargada en el marco k , que es el que contenía a la página $i=9$.

La séptima referencia ($i=7$) produce un **acierto de página**. La pila se reordena para colocar $i=7$ en la cima de la pila.

La octava referencia ($i=8$) produce un **acierto de página**. La pila se reordena para colocar $i=8$ en la cima de la pila.

La novena referencia ($i=6$) produce un **acierto de página**. La pila se reordena para colocar $i=6$ en la cima de la pila.

La décima referencia ($i=5$) produce un **fallo de página**. El número de página $i=5$ se coloca en la cima de la pila y el número $i=7$ se elimina de la pila. Además, la página $i=5$ es cargada en el marco j , que es el que contenía a la página $i=7$.

La undécima referencia ($i=8$) produce un **acierto de página**. La pila se reordena para colocar $i=8$ en la cima de la pila.

La duodécima referencia ($i=6$) produce un **acierto de página**. La pila se reordena para colocar $i=6$ en la cima de la pila.

La decimotercera referencia ($i=5$) produce un **acierto de página**. La pila se reordena para colocar $i=5$ en la cima de la pila.

La decimocuarta referencia ($i=4$) produce un **fallo de página**. El número de página $i=4$ se coloca en la cima de la pila y el número $i=8$ se elimina de la pila. Además, la página $i=4$ es cargada en el marco k , que es el que contenía a la página $i=8$.

La decimoquinta referencia ($i=5$) produce un **acierto de página**. La pila se reordena para colocar $i=5$ en la cima de la pila.

La decimosexta referencia ($i=6$) produce un **acierto de página**. La pila se reordena para colocar $i=6$ en la cima de la pila.

La decimoséptima referencia ($i=5$) produce un **acierto de página**. La pila se reordena para colocar $i=5$ en la cima de la pila.

La decimoctava referencia ($i=6$) produce un **acierto de página**. La pila se reordena para colocar $i=6$ en la cima de la pila.

La decimonovena referencia ($i=4$) produce un **acierto de página**. La pila se reordena para colocar $i=4$ en la cima de la pila.

La vigésima referencia ($i=6$) produce un **acierto de página**. La pila se reordena para colocar $i=6$ en la cima de la pila.

La vigesimoprimera referencia ($i=4$) produce un **acierto de página**. La pila se reordena para colocar $i=4$ en la cima de la pila.

Por tanto, utilizando este algoritmo de reemplazamiento de páginas se obtienen **7 fallos de página**.