

SISTEMAS OPERATIVOS
SEGUNDA PRUEBA
DE
EVALUACIÓN A DISTANCIA
(PED2)

Curso 2016-2017

Nombre: Pablo Francisco Guerra de Diego

DNI: 48.895.423-E

Centro: UNED de Cádiz

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

- I) (1 p) El *algoritmo de Coffman* se utiliza en la estrategia de interbloqueos conocida como *evitación o predicción de interbloqueos*.
Falso, porque se utiliza en la estrategia de tratamiento de interbloqueos conocida como *detección y recuperación de interbloqueos*.
- II) (1 p) La numeración de los sectores de una pista de un disco duro se realiza de forma contigua: 0, 1, 2,...
Falso, se realiza de forma intercalada en función de un determinado *factor de intercalado o entrelazado*. Esto se debe a que el controlador del disco no es lo suficientemente rápido para poder leer/escribir dos sectores adyacentes en una pista y en una sola vuelta cuando pasa por la cabeza de lectura/escritura.
- 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, pues consiste en el almacenamiento temporal de los datos en posiciones de memoria del espacio del núcleo y no en el área de intercambio que pertenece a la memoria secundaria.
- IV) (1 p) La *segmentación simple* facilita la protección y compartición de las diferentes partes de un programa.
Verdadero, pues cada parte de un programa, bien sea una rutina o módulo principal, los conjuntos de subrutinas o módulos auxiliares, y las estructuras de datos, tienen asignados segmentos independientes, la protección y compartición se realiza a nivel de segmento. Con esto nos aseguramos que no haya contenidas diferentes partes (código y datos) en una misma entidad lógica (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:

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

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 .
b) (1.2 p) Determinar si 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 .

$$a) R_E = R_A + R_D = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 2 \\ 0 & 0 & 3 \\ 1 & 0 & 3 \\ 1 & 1 & 0 \end{bmatrix} + (3 \quad 3 \quad 2) = (3 \quad 2 \quad 8) + (3 \quad 3 \quad 2) = (6 \quad 5 \quad 10)$$

Por tanto $x = 3$, $y = 5$, $z = 10$

$$b) S_1 = \left\{ \begin{array}{l} \mathbf{N} = \begin{bmatrix} 4 & 4 & 4 \\ 1 & 4 & 6 \\ 3 & 2 & 5 \\ 4 & 4 & 8 \\ 5 & 1 & 10 \end{bmatrix} \quad \mathbf{A} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 2 \\ 0 & 0 & 3 \\ 1 & 0 & 3 \\ 1 & 1 & 0 \end{bmatrix} \quad \mathbf{R}_E = (6 \ 5 \ 10) \quad \mathbf{R}_D = (3 \ 3 \ 2) \end{array} \right\}$$

Para determinar si el estado S_1 es seguro o no, hay que comprobar si alguna fila i de $\mathbf{N} - \mathbf{A}$ existe, es decir, que algún proceso P_i cumpla la siguiente condición:

$$(\mathbf{N}_i - \mathbf{A}_i) \leq \mathbf{R}_D \quad i = 1, 2, 3$$

El resultado de la resta $\mathbf{N} - \mathbf{A}$ es el siguiente:

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

La primera fila está asociada al proceso P_1 y como vemos no cumple la condición:

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

La segunda fila está asociada al proceso P_2 y tampoco cumple la condición:

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

La tercera fila está asociada al proceso P_3 y sí que cumple la condición:

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

La cuarta fila está asociada al proceso P_4 , no cumple la condición:

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

La quinta fila está asociada al proceso P_5 y tampoco cumple la condición:

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

Por tanto al proceso P_3 se le pueden conceder todos los recursos que necesita para completarse aunque los solicite todos a la vez. Supongamos que el proceso P_3 se ha completado, pasaríamos al siguiente estado:

$$S_2 = \left\{ \begin{array}{l} \mathbf{N} = \begin{bmatrix} 4 & 4 & 4 \\ 1 & 4 & 6 \\ 0 & 0 & 0 \\ 4 & 4 & 8 \\ 5 & 1 & 10 \end{bmatrix} \quad \mathbf{A} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 2 \\ 0 & 0 & 0 \\ 1 & 0 & 3 \\ 1 & 1 & 0 \end{bmatrix} \quad \mathbf{R}_E = (6 \ 5 \ 10) \quad \mathbf{R}_D = (3 \ 3 \ 5) \end{array} \right\}$$

La fila de $\mathbf{N} - \mathbf{A}$ asociada al proceso P_3 se haría 0:

$$\begin{bmatrix} 3 & 4 & 4 \\ 1 & 3 & 4 \\ 0 & 0 & 0 \\ 3 & 4 & 5 \\ 4 & 0 & 10 \end{bmatrix}$$

Por lo tanto podemos determinar que **el estado en el instante T es seguro**, lo cual no quiere decir que en posteriores instantes ($T+1$, $T+2$, ...) tenga que continuar siéndolo.

- c) Como comprobamos en el apartado anterior, partimos de un estado inicial de asignación seguro:

$$S_0 = \left\{ N = \begin{bmatrix} 4 & 4 & 4 \\ 1 & 4 & 6 \\ 3 & 2 & 5 \\ 4 & 4 & 8 \\ 5 & 1 & 10 \end{bmatrix} \quad A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 2 \\ 0 & 0 & 3 \\ 1 & 0 & 3 \\ 1 & 1 & 0 \end{bmatrix} \quad R_E = (6 \ 5 \ 10) \quad R_D = (3 \ 3 \ 2) \right\}$$

Se simula la asignación al proceso P_3 de la instancia del recurso R_2 que ha solicitado y se actualiza el estado del sistema a un estado ficticio denotado como S' :

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

A continuación se comprueba si S' es seguro:

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

Comprobamos si el proceso P_1 cumple la condición:

$$(3 \ 4 \ 4) \leq (3 \ 3 \ 2) \quad \text{No la cumple}$$

Comprobamos si el proceso P_2 cumple la condición:

$$(1 \ 3 \ 4) \leq (3 \ 3 \ 2) \quad \text{No la cumple}$$

Comprobamos si el proceso P_3 cumple la condición:

$$(3 \ 1 \ 2) \leq (3 \ 3 \ 2) \quad \text{Sí la cumple}$$

Una vez que hemos determinado que al menos un proceso cumple la condición, se concede la petición al proceso, en este caso P_3 , y se actualiza el estado del sistema:

$$S_{0+1} = S'$$

Contemplando todo el proceso anterior, podemos determinar que la petición de una instancia del recurso R_2 por parte del proceso P_3 es admitida

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.

Determinamos, según el enunciado, que la secuencia global de referencias a páginas es la siguiente:

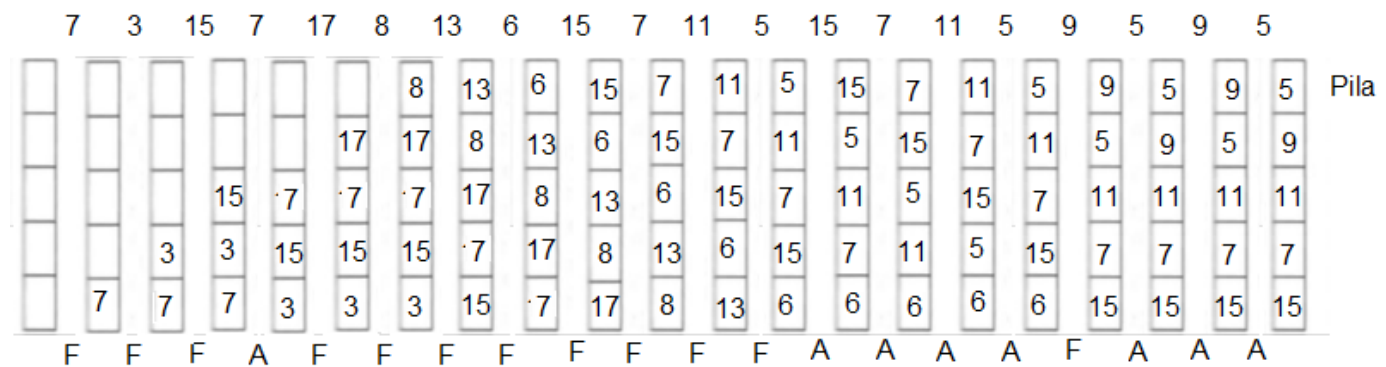


Figura 1: Pila de referencias a páginas

Por tanto, revisando la pila de referencias a páginas de la Figura 1, llegamos a la conclusión de que el número de fallos de página producidos son 12.