

EJERCICIO 1 [4puntos]. Sea una instrucción máquina que ocupa una palabra de memoria, en su ejecución se referencian N direcciones de memoria diferentes, y el hardware la ha implementado como reinicializable (es decir tras una falta de página se reinicia desde el principio, frente al caso de que prosiguiera por el punto en que se generó la falta de página). Explique cuál es el número mínimo de marcos de página que el SO debe asignar a un proceso para que tal instrucción se ejecute correctamente, y justifique su respuesta.

Puede darse el caso de que las N direcciones aludidas en esta instrucción y la palabra donde se almacena la propia instrucción estén cada una en una página distinta, luego hará falta asignar al proceso un mínimo de N+1 marcos; si tiene menos de ese nº de marcos asignados la instrucción no terminaría nunca.

EJERCICIO 2 [4puntos]. En el mecanismo de “Intercambio”: Explique porqué no elegir a un proceso zombi para pasarlo de memoria a disco.

No serviría de nada puesto que no tiene memoria asignada, ya ha liberado todos los recursos que tuviera asignados y solo existe su `task_struct` para almacenar el valor `exit_status` que devuelve al padre en su instrucción `exit(exit_status)`

EJERCICIO 3 [4puntos]. Memoria virtual con paginación: Explicar porqué la dirección de comienzo de la tabla de páginas del proceso actual ha de almacenarse en un registro (RBTP o Registro Base de la Tabla de Páginas) mientras que la dirección de comienzo de la tabla de ubicación en disco (TUD) puede estar en una dirección de memoria principal.

En el acceso a cada dirección aludida a memoria supondría un retraso excesivo acceder a dirección de memoria donde está la dirección de comienzo de la tabla de páginas. Sin embargo, cuando hay una falta de página habrá que leer la página de disco, y en ese tiempo resulta despreciable un acceso a memoria más. Esta es la razón principal. Otra razón es que el valor del RBTP se utiliza un mayor nº de veces que el valor de la dirección de comienzo de la TUD.

EJERCICIO 4 [4puntos]. Memoria virtual con paginación: explicar porqué el algoritmo de sustitución basado en el Conjunto de Trabajo (WS) con ancho de ventana V (parámetro que suponemos ajustado con un valor adecuado) consigue que no se produzcan los fenómenos de hiperpaginación (saturación y pérdida de control) (No es necesario que defina el algoritmo).

Porque asegura que el conjunto de trabajo está en memoria, con lo que las páginas que realmente necesita el proceso están presentes y el nº de faltas de página será muy pequeño.

Responda en la hoja de respuestas asignando A=VERDADERO; B=FALSO

- | | |
|----|--|
| 1) | F* En un sistema informático ideado para que, aparte del SO, se ejecute solo un proceso cada vez carece de utilidad la memoria virtual (Reflexión en libro de Candela 4.30, pag 202) |
| 2) | F*El algoritmo de sustitución FFT (Frecuencia de Faltas de Página) es más costoso de implementar que el basado en el Conjunto de Trabajo (WS) |

- 3)** V*En el algoritmo de sustitución FFT (Frecuencia de Faltas de Página), en el caso de que el intervalo entre faltas de página sea $> L$ (parámetro del algoritmo) no hace falta buscar un marco más para asignar al proceso.
- 4)** V*En el algoritmo de sustitución basado en el Conjunto de Trabajo (WS) con ancho de ventana V , cuando se aplica el algoritmo en el momento de tiempo t , una página que no se ha referenciado en el intervalo $(t-V, t]$ es eliminada de memoria principal
- 5)** F*En el algoritmo de sustitución basado en el Conjunto de Trabajo (WS) con ancho de ventana V , una página que lleva en memoria un tiempo $> V$ es eliminada de memoria principal
- 6)** F*En el algoritmo de sustitución basado en el Conjunto de Trabajo (WS) con ancho de ventana V , aumentar el valor de V tiene siempre como repercusión que aumenta el tamaño de memoria asignada al proceso.

Estamos ideando un algoritmo de sustitución local y asignación fija en que cada página tiene asociado un contador (inicialmente vale 0 cuando la página es cargada en memoria) que es incrementado en 1 por la MMU cuando se referencia una dirección de la página; se elige para la sustitución la página del proceso que tiene un menor valor del contador.

- 7)** V*Puede darse el caso de que se elija como víctima a la página objeto de la última referencia.
- 8)** F*Siempre se elige como víctima a la página objeto de la última referencia.
- 9)** F*Asegura mejor que el LRU que se mantienen en memoria las páginas con más probabilidad de ser utilizadas en un futuro inmediato.

Sea un proceso que tiene asignados tres marcos de página inicialmente vacíos, y genera la siguiente secuencia de referencias a páginas: 1, 2, 3, 2, 1, 4, 1, 3, 2, 1, 3

- 10)** V*Con el algoritmo FIFO, cuando se hace referencia a la página 4 se elige como víctima la 1 (es decir, la 4 es almacenada en el marco hasta ahora ocupado por la 1)
- 11)** F*Con el algoritmo LRU, cuando se hace referencia a la página 4 se elige como víctima la 1
- 12)** V*El nº de faltas de página generados por el algoritmo LRU es 6(contando las tres primeras lecturas de páginas a memoria a partir de los tres marcos vacíos)

13) V*Entre los algoritmos de sustitución local y asignación fija (LRU, FIFO, Reloj y Optimo) el FIFO es el que más faltas de página provoca en promedio.

Supongamos que a cierto proceso se le han asignado m marcos de página inicialmente vacíos, y en su ejecución genera una cadena de referencias de páginas de longitud p , siendo n el nº de páginas distintas de dicha cadena. Independientemente del método de sustitución elegido.....

- 14)** V*se producen como mínimo n faltas de página
- 15)** F*puede haber más de p faltas de página

16) V*Es el SO (y no la MMU) quien da valor al campo “Número de Marco” de la tabla de páginas

17) F*Es la MMU (y no el SO) quien da valor al campo “Permisos” de la tabla de páginas

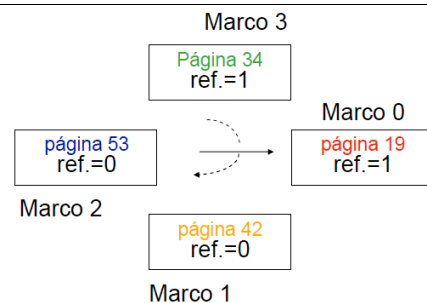
Sobre el mecanismo de Intercambio o Swapping (sin memoria virtual):

- 18)** V*Tiene como objetivo permitir que existan más procesos que los que caben simultáneamente en memoria principal.
- 19)** F*Un proceso bloqueado no puede ser elegido para pasar de memoria a disco.

Algoritmo del Reloj. Tenemos un proceso con 4 marcos de página asignados como puede verse en la figura, el puntero del algoritmo marca actualmente el marco 0 (página 19).

- 20)** F*A partir de la situación de la figura, para elegir un marco víctima el algoritmo elegirá el Marco 0
- 21)** V*A partir de la situación de la figura, para elegir un marco víctima el algoritmo elegirá el Marco 1

22) F* El bit de referencia del marco elegido como víctima queda a 0



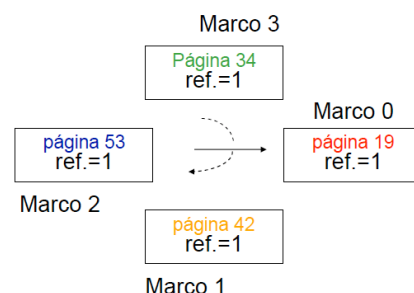
Algoritmo del Reloj. Tenemos un proceso con 4 marcos de página asignados como puede verse en la figura, el puntero del algoritmo marca actualmente el marco 0 (página 19).

23) V*A partir de la situación de la figura, para elegir un marco víctima el algoritmo elegirá el Marco 0

24) F*A partir de la situación de la figura, para elegir un marco víctima el algoritmo elegirá el Marco 3

25) V*Tras la elección realizada en el enunciado anterior, el bit de referencia del Marco 1 queda a 0

26) V*Tras la elección realizada en el enunciado anterior, el bit de referencia del Marco 2 queda a 0



27) V*La segmentación puede generar fragmentación externa

28) F*La segmentación paginada puede generar fragmentación externa

29) V*La gestión de la protección es más complicada en paginación que en segmentación

30) V*La gestión de la compartición es más complicada en paginación que en segmentación

31) F*En paginación sin memoria virtual, pero donde tenemos implementado el mecanismo de Intercambio (Swapping), carece de utilidad mantener el bit de Modificación asociado a cada página.

En el ciclo de vida de las páginas de un proceso...

32) V*Cuando se elige como víctima para la sustitución un marco que contiene una página de código, ésta página no ha de escribirse en disco puesto que cuando sea necesaria en el futuro se leerá del archivo ejecutable.

33) F*Ante la primera falta de página de una página correspondiente a “Datos con valor inicial”, el SO no ha de leerla de disco.

34) V*Ante la primera falta de página de una página perteneciente a la pila, el SO no ha de leerla de disco.

Suponga un sistema de gestión de memoria que utiliza paginación a dos niveles.

Las direcciones son de 16 bits y tienen la siguiente estructura: 5 bits para paginación a primer y segundo nivel respectivamente y 6 bits para desplazamiento.

El espacio de direcciones virtuales de un proceso (ver figura) está constituido por: 128 bytes (B) de código, 64 B de datos, la pila ocupa 64B y está situada al final del espacio de direcciones, y el resto está sin usarse.

El tamaño de la palabra de memoria es de 1B.

Cada entrada de una tabla de páginas ocupa 2B.

En cierto momento tenemos la siguiente asignación de memoria:

Código0 está en el marco 8

Código1 está en el marco 12

Datos está en el marco 14

Pila está en el marco 7

La Pág.nº0 de las Tablas de Páginas a 2º nivel está en el marco 41

La Pág.nº31 de las Tablas de Páginas a 2º nivel está en el marco 56

La Tabla de Páginas a primer nivel está en el marco 63

	desde	hasta
Código 0	0	63
Código 1	64	127
Datos	128	191
...		
Pila	65471	65535

debe ser 65472

35) V*El tamaño de página es de 64B.

36) V*En una página caben 32 entradas de tablas de página.

37) F*La dirección virtual 200 actualmente está en uso.

38) F*La dirección virtual 14 pertenece a Datos

39) V*La primera entrada de la Tabla de Páginas a primer nivel contiene el valor 41

40) F*La última entrada de la Tabla de Páginas a primer nivel contiene el valor 7

41) V*La segunda Tabla de Páginas a 2º nivel no está cargada en memoria.

42) F*La siguiente dirección virtual expresada en binario tiene desplazamiento=0

1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

43) F*La dirección virtual 65534 (expresado en base 2 a continuación) corresponde a la dirección real (NºMarco=7, Desplazamiento=30)

1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Se ha rectificado a falsa. Los 6 bits de la dcha :111110 corresponden al decimal 62, por lo que este enunciado es falso.

44) V*La dirección virtual 195 genera una falta de página

0	0	0	0	0	0	0	0	1	1	0	0	0	0	1	1
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

45) F*La dirección virtual 60 corresponde a la dirección real (NºMarco=5, Desplazamiento=60)

46) V*La dirección real (NºMarco=8, Desplazamiento=10) corresponde a la dirección virtual 10

47) F*La dirección real (NºMarco=7, Desplazamiento=0) corresponde a la dirección virtual 65471

ITEM ANULADO. El enunciado es falso. Véase que en la figura que encabeza el enunciado de este grupo de item hay que corregir 65471 por 65472. Traduciendo la dirección virtual 65471 se produce una falta de página, por lo que el enunciado realmente es falso. Pero si uno se basa en los datos de la figura puede concluir que es verdadera.

48) F*Si el proceso necesitara una página más para Datos habría que construir una nueva Tabla de Páginas a 2º nivel.

49) F*Se se compactara la memoria principal llevando el contenido de todos los marcos en uso a partir del nº0 en adelante la velocidad de acceso a memoria se incrementaría.