

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

I) (1 p) La sobrepaginación aumenta el porcentaje de uso del procesador. Esto es falso: la sobrepaginación implica que el sistema operativo tendrá que atender al tratamiento de fallos de página con mayor frecuencia, lo que aumenta la sobrecarga del sistema y produce que los procesos de usuario puedan usar menos tiempo de procesador.

II) (1 p) Se denomina buffering de páginas a la estrategia consistente en cargar un cierto número de páginas de un proceso antes de iniciar o continuar su ejecución.

Falso: a la estrategia descrita se la denomina prepaging, no buffering de páginas, que se usa para reducir el tiempo dedicado a la atención a los fallos de página

2. (2 p) Un sistema con memoria virtual mediante demanda de páginas utiliza el algoritmo LRU para la sustitución de páginas. Un proceso genera la siguiente secuencia de referencias a páginas de memoria:

1 3 2 4 1 5 7 4 3 2 8 9 4 5 4 9 1 8 3 2

a) Determinar cuántos fallos de página se producen cuando se dispone de 4 o 5 marcos de página para este proceso.

Se supone que para implementar el algoritmo LRU se utiliza una lista enlazada que se gestiona como una pila. En las tablas siguientes se muestra el contenido de la lista gestionada como una pila utilizada para implementar el algoritmo LRU en el caso de utilizar 4 y 5 marcos. Cada referencia que produce un fallo de página se respresenta con una F y los aciertos con una A.

1	3	2	4	1	5	7	4	3	2	8	9	4	5	4	9	1	8	3	2	
				4	1	5	7	4	3	2	8	9	4	5	4	9	1	8	3	2
			2	2	4	1	5	7	4	3	2	8	9	4	5	4	9	1	8	3
		3	3	3	2	4	1	5	7	4	3	2	8	9	9	5	4	9	1	8
	1	1	1	1	3	2	4	1	5	7	4	3	2	8	8	8	5	4	9	1
F	F	F	F	A	F	F	A	F	F	F	F	F	F	A	A	F	F	F	F	

Con 4 marcos de página se producen 16 fallos de página.

1	3	2	4	1	5	7	4	3	2	8	9	4	5	4	9	1	8	3	2	
						5	7	4	3	2	8	9	4	5	4	9	1	8	3	2
				4	1	1	5	7	4	3	2	8	9	4	5	4	9	1	8	3
			2	2	4	4	1	5	7	4	3	2	8	9	9	5	4	9	1	8
		3	3	3	2	2	4	1	5	7	4	3	2	8	8	8	5	4	9	1
	1	1	1	1	3	3	2	2	1	5	7	4	3	2	2	2	8	5	4	9
F	F	F	F	A	F	F	A	F	F	F	F	A	F	A	A	F	A	F	F	

Con 5 marcos de página se producen 14 fallos de página, un 12,5% menos que con 4 marcos.

b) Explicar razonadamente si mejoraría la tasa de fallos de página si se aumentase el número de marcos de página a N, siendo $N > 5$.

Dado que el algoritmo LRU no presenta la anomalía de Belady (comportamiento por el cual puede aumentar el número de fallos de página al aumentar el número de marcos), si se aumenta el número de marcos de página la tasa de fallos de página mejorará.

3. (2 p) Explique razonadamente las funciones que realizan las capas de software de E/S del núcleo de un sistema operativo.

Existen tres capas de software: el subsistema de E/S, los drivers de dispositivos y los manejadores de interrupciones.

El *subsistema de E/S* es el componente del SO que se encarga de efectuar las tareas necesarias para las operaciones de E/S comunes a todos los dispositivos e independientes de los mismos, como: asignación y liberación de dispositivos dedicados, bloqueo de procesos que solicitan una operación de E/S, planificación de la E/S, invocación del driver de dispositivo apropiado, buffering, proporcionar un tamaño de bloque uniforme a los niveles superiores de software, y la gestión de errores producidos en operaciones E/S.

Un *driver de dispositivo* proporciona al subsistema de E/S el conjunto de funciones que se pueden realizar sobre el dispositivo, tales como lectura o escritura. El driver de dispositivo interactúa con el controlador de E/S, y carga en sus registros órdenes para que las efectúe sobre el dispositivo, comprobando su estado e inicializándolo si fuera necesario. Por otra parte comprueba errores en la operación de E/S y notifica al subsistema de E/S una vez que esta ha finalizado.

Los *manejadores de interrupciones*, que forman parte del núcleo del SO y son extremadamente dependientes del hardware. Cuando un dispositivo termina la operación E/S su controlador E/S genera una interrupción que ha de ser atendida por el correspondiente manejador. Las instrucciones específicas que realiza un manejador de interrupciones dependen de cada tipo de interrupción. Si el driver del dispositivo se bloqueó en espera de que el controlador de E/S estuviera preparado para procesar otra petición de E/S, entonces una acción que debe realizar un manejador de interrupción es el desbloqueo del driver del dispositivo. También se puede encargar de transferir datos desde un registro del controlador del dispositivo a un buffer en el espacio del núcleo o viceversa según sea operación de lectura o escritura, respectivamente, aunque en ocasiones esto lo realiza el driver.

4. En un computador con una capacidad de memoria principal de 64 kibipalabras se utiliza gestión de memoria mediante segmentación. La tabla de segmentos (todos los datos numéricos están en decimal) es la siguiente:

Nº de segmento	Base	Longitud
0	0	7230
1	16384	8191
2	32768	1024
3	8192	356
4	24576	4200

Se pide:

a) (1 p) Supuesto que una dirección lógica tiene el mismo tamaño en bits que una dirección física y que consta de los campos [nº de segmento, desplazamiento], determinar el tamaño en bits de cada uno de estos campos.

Cmp = 64 Kibipalabras = 2^{16} palabras, luego el número de bits de una dirección física ha de tener 16 bits.

Para determinar el número de bits del campo número de segmento s , relovemos la siguiente desigualdad: $s = \min_s \{N_s \leq 2^s\}$, donde N_s es el número de segmentos. Como $N_s = 5$, se necesitan al menos $s = 3$ bits.

El tamaño mínimo del campo d desplazamiento depende de la longitud de cada segmento, pero tendrá una longitud máxima de 13 bits, porque el tamaño de la dirección lógica no puede ser mayor que el tamaño de la dirección física, luego $16 - 3 = 13$ bits. Ya que el enunciado del problema expresa que una dirección lógica tiene el mismo tamaño en bits que una dirección física, y dado que el tamaño del campo s número de segmento es común a todos los segmentos, el tamaño del campo d desplazamiento es de 13 bits para todos los segmentos (con independencia de si se necesitan más o menos bits para direccionar todas las direcciones de cada segmento en particular).

b) (1 p) Determinar a qué direcciones físicas expresadas en decimal corresponden las siguientes direcciones lógicas expresadas en hexadecimal:

- 11AE16**, $(11AE)_{16} = (0001000110101110)_2$ los 3 bits más significativos 000 corresponden al número de segmento, 0. los 13 bits restantes en decimal son 4526 que siendo menor que el desplazamiento del segmento 0 dan lugar a la dirección física válida (0,4526)
- 6190**, $(6190)_{16} = (0110000110010000)_2$ los 3 bits más significativos 011 corresponden al número de segmento válido 3. los 13 bits restantes en decimal son $(0000110010000)_2 = (400)_{10}$ que al ser mayores que el desplazamiento del segmento 3, dan lugar a una dirección física inválida.

5. La política de gestión de memoria de un cierto sistema es del tipo demanda de página. El tamaño de una página es de 1 KiB, el tamaño máximo de la memoria virtual es de 4 MiB y el tamaño de la memoria física es de 1 MiB. Se pide:

a) (1 p) Determinar el tamaño de cada uno de los campos de una dirección virtual y de una dirección física.

Para hallar el tamaño n de una dirección física, Cmp = 1 MiB = 2^{20} bytes. Suponemos un tamaño de palabra de 1 byte (el enunciado no da el dato) lo que resulta en 2^{20} palabras

direccionables, con lo que la dirección física tendrá un tamaño máximo de 20 bits.

El campo f número de marco lo calculamos resolviendo la desigualdad $f = \min_f \{N_{MP} \leq 2^f\}$ donde N_{MP} es el número de marcos de página en que se divide la memoria principal, que se calcula de la siguiente manera: $N_{MP} = \text{floor}(\frac{C_{MP}}{S_P} = \frac{2^{23} \text{ bits}}{2^{13} \text{ bits}}) = 2^{10} \text{ marcos}$. Luego f tiene un tamaño de 10 bits. Al ser el tamaño de la dirección la suma de los campos s y d (desplazamiento), el campo d vale $n-f = 20 - 10 = 10$ bits.

Para dirección virtual partimos del tamaño máximo de la memoria virtual C_A

$4 \text{ MiB} = 2^{22} \text{ bytes} = 2^{22} \text{ palabras}$, si consideramos como tamaño de la palabra 1 byte. Luego el tamaño de la dirección virtual m será como máximo de 22 bits. El campo desplazamiento tiene el mismo tamaño que en la dirección física, 10 bits. Entonces ya que $m = p + d$, p o campo número de página es igual a $m - p = 22 - 10 = 12$ bits.

b) (1 p) Determinar la capacidad mínima que debe tener la tabla de páginas del proceso de mayor tamaño que se puede ejecutar en el sistema. ¿Qué tanto por ciento de la memoria principal ocuparía dicha tabla?

El tamaño de mayor tamaño ejecutable en el sistema ocuparía toda la memoria virtual, es decir 4 MiB o 2^{20} bytes . Su tabla de páginas tiene tantas entradas como número de

páginas tenga el proceso $N_P = \text{ceil}(\frac{C_A}{S_P} = \frac{2^{23} \text{ bits}}{2^{13} \text{ bits}}) = \frac{2^{20} \text{ bytes}}{2^{10} \frac{\text{bytes}}{\text{página}}} = 2^{10} \text{ páginas}$. Luego la tabla

de páginas del proceso habría de tener 2^{10} entradas. Una entrada de la tabla de páginas contiene el campo marco de página, de 10 bits, y los bits r , m y v . Luego el tamaño E de una entrada es: $E = 10 + 1 + 1 + 1 = 13$ bits.

Así que el tamaño de la tabla de páginas será

$T_{TP} = 13 \frac{\text{bits}}{\text{entrada}} * 2^{10} \text{ entradas} = 13312 \text{ bits} = 1664 \text{ bytes}$ Y el porcentaje de memoria principal

que ocupa es $(\frac{T_{TP}}{C_{MP}}) * 100 = (\frac{1664}{2^{20}}) * 100 = 0,158 \% \text{ de la memoria principal.}$