

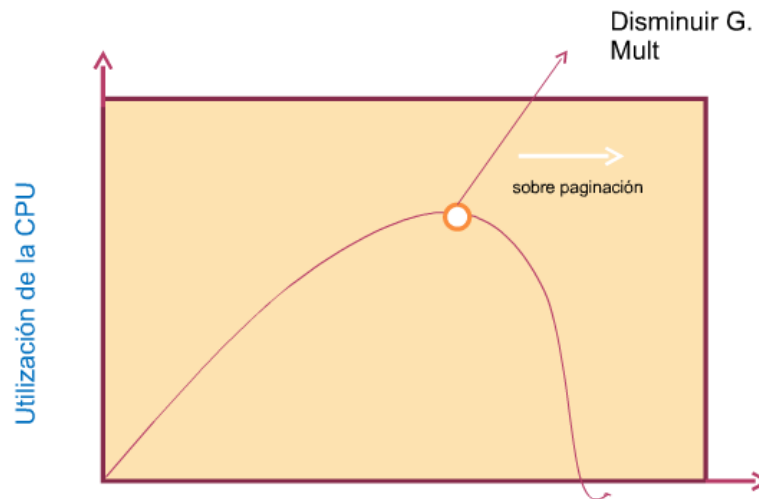
SISTEMAS OPERATIVOS

Segunda prueba de evaluación a distancia (PED2)

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

I) (1 p) La sobrepaginación aumenta el porcentaje de uso del procesador.

Falso: El porcentaje de uso del procesador aumenta cuando el grado de multiprogramación aumenta hasta llegar a un punto en el que empieza a producirse sobrepaginación. Superado este punto, disminuye el uso de CPU



Grado de Multiprogramación

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: La política de buffering de páginas consiste en mantener un conjunto de marcos de página libres que conservan la información de la página que contuvieron la última vez, de tal modo que en caso de fallo de página, busca la página requerida primero en este "buffer".

La estrategia que carga un cierto número de páginas de un proceso antes de iniciar o continuar su ejecución se denomina "Paginación por adelantado"

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.

Con 4 marcos:

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

Se producen 16 fallos de página.

Con 5 marcos:

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

Se producen 14 fallos de página.

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

Al aumentar el número de marcos de página el número de fallos de página siempre será menor o igual ya que se dispondrán de más espacios donde almacenar las referencias y esto aumentará la probabilidad de acierto en las siguientes lecturas.

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

El software del núcleo de un sistema operativo, se puede organizar en tres capas: **subsistema de E/S, drivers de los dispositivos de E/S y manejadores de las interrupciones.**

Cada capa posee una interfaz bien definida para comunicarse con las capas adyacentes. Además cada capa realiza unos servicios determinados.

Subsistema de E/S:

El subsistema de E/S es el componente del sistema operativo que se encarga de efectuar todas aquellas tareas necesarias para la realización de las operaciones de E/S que son comunes a todos los dispositivos e independientes de los mismos. Es decir, el subsistema de E/S, gestiona la parte independiente del dispositivo de todas las operaciones de E/S. Entre las tareas que se encarga de realizar el subsistema de E/S se encuentran las siguientes:

- *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.*
- *Almacenamiento temporal de datos de E/S o buffering*
- *Proporcionar un tamaño de bloque uniforme a los niveles superiores de software.*
- *Gestión de los errores producidos en una operación de E/S.*

Drivers de dispositivos de E/S

El funcionamiento y las características de los dispositivos de E/S varían considerablemente de un dispositivo a otro. Debido a estas diferencias el sistema operativo necesita disponer de un código específico para poder controlar cada dispositivo de E/S.

Un driver de dispositivo contiene el código que permite a un sistema operativo controlar un determinado tipo de dispositivo de E/S. Así, se requiere un driver para el ratón, otro driver para el teclado, otro driver para el disco duro, etc. Los drivers son facilitados por los fabricantes de los dispositivos para cada SO. Un driver se diseña

teniendo en cuenta las especificaciones de la interfaz de drivers del subsistema de E/S de cada sistema operativo y las características del dispositivo.

Un driver de dispositivo interactúa con el subsistema de E/S y con el controlador de E/S que controla el dispositivo. Un driver suministra al subsistema de E/S el conjunto de funciones que se pueden realizar sobre el dispositivo, tales como lectura o escritura. Además un driver puede invocar a ciertas rutinas o procedimientos del núcleo. Los procedimientos a los que tiene acceso un driver quedan definidos por la interfaz de drivers del subsistema de E/S.

El driver de un dispositivo interactúa con el controlador de E/S, cargando en sus registros diferentes órdenes para que las efectúe sobre el dispositivo, comprobando su estado e inicializándolo si es necesario. Cuando el subsistema de E/S invoca a una función de un driver, éste debe realizar un conjunto de acciones. Las acciones que realiza un driver y el orden en que las realiza, pueden variar dependiendo del tipo de dispositivo y sistema operativo.

Manejadores de las interrupciones.

Cuando finaliza una operación de E/S en un dispositivo y éste se encuentra preparado para procesar otra, el controlador de E/S que lo supervisa genera una interrupción. En un computador que implemente un sistema de interrupciones vectorizadas cada interrupción suministra un número denominado número del vector de interrupción que se utiliza como índice en una tabla, denominada tabla de vectores de interrupción. Esta tabla usualmente se encuentra almacenada en las posiciones más bajas de memoria. Cada entrada de esta tabla se denomina vector de interrupción, que entre otras informaciones contiene la dirección de comienzo del manejador (handler) de la interrupción.

Los manejadores de interrupciones forman parte del núcleo del sistema operativo y son extremadamente dependientes del hardware, Por este motivo, a la hora de portar un sistema operativo a una nueva arquitectura, esta parte del núcleo siempre tiene que ser reescrita.

Con objeto de que el rendimiento del computador sea óptimo, los manipuladores de interrupciones tienen una alta prioridad de ejecución. Además su código suele ser pequeño y rápido de ejecutar, Las acciones 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 mediante el uso del mismo mecanismo de sincronización (semáforo, mensajes,...) que utilizó el driver para bloquearse.

En el caso de que no se realice DMA un manejador de una interrupción también se puede encargar de transferir datos desde un registro del controlador del dispositivo a un buffer en el espacio del núcleo, o viceversa en función de si se trata de una operación de lectura o escritura, respectivamente. Esta función, en ocasiones, la puede realizar el propio driver en lugar del manejador.

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 (s), desplazamiento (d)], determinar el tamaño en bits de cada uno de estos campos.

Puesto que disponemos de 5 segmentos, el campo nº de segmento tendrá un tamaño de $s=3$ bits. Por otra parte, el tamaño mínimo d del campo desplazamiento depende de la longitud S_s de cada segmento expresada en palabras y se determina resolviendo la desigualdad:

$$\min_d \{ S_s \leq 2^d \}$$

El segmento 0 tiene una longitud de 7230 bits / 8 = 903,75 bytes $\leq 2^{10}$. Luego una dirección lógica asociada al segmento 0 tiene un tamaño de $s + d = 13$ bits

El segmento 1 tiene una longitud de 8191 bits / 8 = 1023,875 bytes $\leq 2^{10}$. Luego una dirección lógica asociada al segmento 1 tiene un tamaño de $s + d = 13$ bits

El segmento 2 tiene una longitud de 1024 bits / 8 = 128 bytes $\leq 2^7$. Luego una dirección lógica asociada al segmento 2 tiene un tamaño de $s + d = 10$ bits

El segmento 3 tiene una longitud de 356 bits / 8 = 44,5 bytes $\leq 2^6$. Luego una dirección lógica asociada al segmento 3 tiene un tamaño de $s + d = 9$ bits

El segmento 4 tiene una longitud de 4200 bits / 8 = 525 bytes $\leq 2^{10}$. Luego una dirección lógica asociada al segmento 4 tiene un tamaño de $s + d = 13$ bits

Considerando que el tamaño del campo d viene fijado por el segmento de longitud más grande, con lo que el campo desplazamiento más grande sería de 10 bits y en consecuencia el tamaño de todas las direcciones sería de $3 + 10 = 13$ bits.

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

i) $11AE_{16}$,

$11AE_{16} = 1000110101110_2$

Los tres bits más significativos (100) indican el segmento al que pertenece (4)

El resto de bits 0110101110_2 indican el desplazamiento en binario = 430 en decimal.

Dirección en decimal = BASE segmento 4 + Despl = 24576 + 430 = 25006

ii) 6190_{16} ,

$6190_{16} = 110000110010000_2$

Los tres bits más significativos (110) indican el segmento al que pertenece (6) como no existe este segmento se produciría un error de direccionamiento.

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.

La dirección física se descompone en los campos número de marco de página de f bits y desplazamiento dentro del marco de d bits. Por su parte una dirección

virtual se descompone en número de página de p bits y desplazamiento dentro de la página de d bits.

- Dirección física:

Del dato de la capacidad de la memoria principal C_{MP} expresada en palabras se puede determinar el tamaño n en bits de una dirección física, para ello hay que resolver la siguiente desigualdad:

$$\min_n \{ C_{MP} \leq 2^n \}$$

Del enunciado se sabe que $C_{MP} = 1 \text{ MiB} = 2^{20}$ bytes, luego $n=20$ bits

Por otra parte del tamaño de una página S_p expresado en palabras se puede determinar el tamaño d en bits del campo desplazamiento tanto de una dirección física como de una dirección virtual, para ello hay que resolver la siguiente desigualdad:

$$\min_d \{ S_p \leq 2^d \}$$

Del enunciado se sabe que $S_p = 1 \text{ KiB} = 2^{10}$ bytes, luego **$d=10$ bits**

El tamaño f del campo número de marco de página se puede obtener, por ejemplo, de la siguiente forma:

$$f = n - d = 20 - 10 = 10 \text{ bits}$$

Por tanto la dirección física consta de un total de $n=20$ bits de los cuales $f=10$ corresponden al marco de página y $d=10$ corresponden al desplazamiento dentro del marco.

- Dirección virtual:

Del dato del tamaño C_A del espacio de direcciones virtuales se puede determinar el tamaño m en bits de una dirección virtual, para ello hay que resolver la siguiente desigualdad:

$$\min_m \{ C_A \leq 2^m \}$$

Del enunciado se sabe que $C_A = 4 \text{ MiB} = 2^{22}$ bytes, luego **$m=22$ bits**

El tamaño del campo desplazamiento de una dirección virtual tiene el mismo tamaño que el de una dirección física. Por su parte, el tamaño p del campo número de página se puede obtener, por ejemplo, de la siguiente forma:

$$p = m - d = 22 - 10 = 12 \text{ bits}$$

Por tanto la dirección virtual consta de un total de $m=22$ bits de los cuales $p=12$ corresponden al número de página y $d=10$ corresponden al desplazamiento dentro de la página.

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 número de páginas N_p de que consta el proceso máximo se calcula de la siguiente forma:

$$N_p = \text{ceil} \left(\frac{C_{MP}}{S_p} \right) = \text{ceil} \left(\frac{2^{20}}{2^{10}} \right) = 2^{10} = 1024 \text{ páginas}$$

Es decir, la tabla de páginas T_p debe tener 2^{10} entradas.

Por otro lado, una entrada de la tabla de páginas contiene el campo marco de página y los bits r , m y v . El campo marco de página de la dirección física calculado en el apartado (a) es $f=10$ bits, por lo que si sumamos los bits r , m y v se tendría un total de 13 bits por entrada: $E=13$.

Por tanto, la capacidad máxima de la tabla de páginas:

$$C_{Tp}=(T_p \times E)=2^{10} \times 13= 13312 \text{ bits}=1664 \text{ bytes}$$

El porcentaje P de memoria principal que ocupa esta tabla es de:

$$P = \frac{C_{Tp}}{C_{Mp}} = \frac{1664 \text{ bytes}}{2^{20} \text{ bytes}} = \frac{1664 \text{ bytes}}{1048576 \text{ bytes}} * 100 = 0,15869140625 = 15,87\%$$