

# **SISTEMAS OPERATIVOS**

## **SEGUNDA PRUEBA DE EVALUACIÓN A DISTANCIA (PED2)**

**Laura Almón Manzano  
DNI: 53191030-B  
Centro Asociado de Pontevedra**

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

**I) La sobrepaginación aumenta el porcentaje de uso del procesador.**

**FALSO.**

Si se producen fallos de página constantemente, es decir, si se produce sobrepaginación, el uso del procesador disminuirá debido a que los procesos se bloquearían porque tendrían que esperar por la lectura y escritura de páginas a memoria secundaria.

**II) 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.**

Se denomina buffering de páginas a la estrategia consistente en mantener una reserva de marcos libres implementada en la forma de una lista de marcos libres, ya que el tiempo de tratamiento de un fallo de página se reduce si se dispone de un marco libre de memoria principal donde cargar la página referenciada por la dirección virtual que ha producido el fallo de página.

**2. 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.**

- 4 marcos de página: 16 fallos de página

[illegible][illegible]

- 5 marcos de página: 14 fallos de página

[illegible]

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

Si se aumentase el número de marcos de página, mejoraría la tasa de fallos de página. Cuanto mayor sea el número de marcos de página, menor sería la tasa de fallos de página, pues las páginas permanecerían más tiempo cargadas en memoria, y habría más páginas diferentes cargadas en memoria, lo cual favorecería a más aciertos.

### **3. 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 necesario para la gestión de las operaciones de E/S se puede organizar en tres capas:

\* SUBSISTEMA DE E/S:

- Asignación y liberación de dispositivos dedicados. Algunos dispositivos, llamados dispositivos dedicados, solo pueden ser utilizados simultáneamente por un único proceso. Es responsabilidad del subsistema de E/S determinar si una petición de E/S de un proceso sobre un dispositivo dedicado puede ser aceptada o debe ser rechazada.

- Bloqueo de procesos que solicitan una operación de E/S. Cuando al subsistema de E/S le llega, a través de la interfaz de llamadas al sistema, una petición de un proceso para la realización de una operación de E/S, éste puede pasar o no al estado bloqueado al proceso que solicitó la operación de E/S.

- Planificación de la E/S. Con el objetivo de disminuir el tiempo de espera promedio para la finalización de una operación de E/S y distribuir equitativamente el uso de los recursos entre todos los procesos, el subsistema de E/S realiza una planificación del uso de los dispositivos de E/S.

- Invocación del driver de dispositivo apropiado. Cuando el subsistema de E/S tiene que realizar alguna operación de E/S, a petición de un proceso, debe invocar al driver adecuado. El subsistema de E/S traduce la cadena de caracteres que es utilizada por los procesos para hacer referencia a los dispositivos, en la dirección de inicio del driver correspondiente.

- Almacenamiento temporal de datos de E/S o buffering. El subsistema de E/S se encarga de asignar buffers (áreas en memoria principal que almacenan temporalmente datos mientras éstos se transfieren entre un proceso y un dispositivo, o viceversa) para el almacenamiento temporal de los datos que se leen o se escriben en las operaciones de E/S sobre los dispositivos.

- Proporcionar un tamaño de bloque uniforme a los niveles superiores de software. Así las capas superiores de software pueden trabajar con tamaños de bloques lógicos independientemente del tamaño de los bloques físicos del dispositivo de E/S.

- Gestión de los errores producidos en una operación de E/S. El subsistema de E/S se encarga de decidir las acciones a tomar cuando se producen errores (errores de programación y errores en el dispositivo) en una operación de E/S.

Además, el subsistema de E/S debe proporcionar una interfaz para los drivers de los dispositivos.

**\* DRIVERS DE LOS DISPOSITIVOS DE E/S:**

- Comprobar que los parámetros de la función invocada por el subsistema de E/S son correctos y que la operación de E/S solicitada se puede realizar. En caso contrario devuelve un error.

- Traducir los parámetros de dicha función en parámetros específicos del dispositivo.

- Comprobar si el dispositivo de E/S está ocupado atendiendo alguna petición anterior de E/S. Si el dispositivo está ocupado, coloca la petición en una cola. Si no está atendiendo ninguna petición, comprueba si se encuentra preparado, ya que quizás el driver deba activar e inicializar el dispositivo.

- Generar un conjunto de órdenes para el controlador del dispositivo dependiendo de la petición de E/S solicitada por el subsistema de E/S. Dichas órdenes son cargadas en los registros del controlador. El driver puede comprobar que el controlador acepta una orden y que se encuentra listo para aceptar la siguiente.

- Una vez transmitidas todas las órdenes al controlador, el driver debe esperar a que el controlador las ejecute. Si el tiempo de espera estimado es importante, entonces el driver se bloquea usando algún mecanismo de sincronización hasta que el controlador finalice. Cuando la operación de E/S se complete, el controlador activará una interrupción. El manejador o rutina de servicio de dicha interrupción despertará al driver, si éste se bloqueó.

- Comprobar que no se han producido errores en la operación de E/S. En dicho caso, quizás transfiera al subsistema de E/S el resultado de la operación de E/S. Si se ha producido algún error y el driver sabe cómo resolverlo, realiza la acción correspondiente. Si no sabe cómo resolverlo o la solución que plantea no surte efecto, entonces informa del error al subsistema de E/S para que tome las medidas que considere oportunas.

- Examinar la cola de peticiones de E/S pendientes. Si existe alguna, procede a atenderla. Si la cola está vacía, el driver se bloqueará en espera de la llegada de nuevas peticiones.

**\* MANEJADORES DE INTERRUPCIONES:**

Las acciones específicas que realiza un manejador de interrupciones dependen de cada tipo de interrupción.

Si el driver de 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 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 de 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) 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.**

Una dirección lógica consta de dos campos: el número de segmento de  $s$  bits y el desplazamiento dentro del segmento de  $d$  bits. El tamaño  $s$  del campo número de segmento se obtiene a partir del número total de segmentos  $N_s$  de que consta un proceso  $X$ , resolviendo la siguiente desigualdad:

$$\min_s \left\{ N_s \leq 2^s \right\}$$

El número de segmento  $s$  es igual a:

$$N_s = 5 \leq 2^3 = 8 \Rightarrow s = 3$$

Por su parte el tamaño mínimo  $d$  del campo desplazamiento dentro de un segmento se obtiene a partir del tamaño  $S_s$  del segmento expresado en unidades direccionables, resolviendo la siguiente desigualdad:

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

- Segmento nº 0:

$$7230 \leq 2^{13} \Rightarrow d = 13 \text{ bits} \Rightarrow m = 3 + 13 = 16 \text{ bits}$$

- Segmento nº 1:

$$8191 \leq 2^{13} \Rightarrow d = 13 \text{ bits} \Rightarrow m = 3 + 13 = 16 \text{ bits}$$

- Segmento nº 2:

$$1024 \leq 2^{10} \Rightarrow d = 10 \text{ bits} \Rightarrow m = 3 + 10 = 13 \text{ bits}$$

- Segmento nº 3:

$$356 \leq 2^9 \Rightarrow d = 9 \text{ bits} \Rightarrow m = 3 + 9 = 12 \text{ bits}$$

- Segmento nº 4:

$$4200 \leq 2^{13} \Rightarrow d = 13 \text{ bits} \Rightarrow m = 3 + 13 = 16 \text{ bits}$$

El tamaño del campo desplazamiento viene fijado por el segmento de longitud más grande (el segmento 0 con 13 bits), por lo tanto el tamaño de todas las direcciones lógicas del proceso será de  $3 + 13 = 16$  bits.



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

**I)  $11AE_{16}$**

$$11AE_{16} = 1 \times 16^3 + 1 \times 16^2 + 10 \times 16^1 + 14 \times 16^0 = 4526$$

Esta dirección lógica pasada a binario es:

0001000110101110

Los 3 primeros bits corresponden al número de segmento y los 13 bits restantes al campo desplazamiento. Pasando a decimal el valor de ambos campos se obtiene la dirección:

(0, 4526)

Esta dirección hace referencia a la palabra 4526 dentro del segmento 0. Como el desplazamiento no supera el tamaño del segmento ( $4526 < 7230$ ), se trata de una dirección lógica válida.

**II)  $6190_{16}$**

$$6190_{16} = 6 \times 16^3 + 1 \times 16^2 + 9 \times 16^1 + 0 \times 16^0 = 24976$$

Esta dirección lógica pasada a binario es:

0110000110010000

Los 3 primeros bits corresponden al número de segmento y los 13 bits restantes al campo desplazamiento. Pasando a decimal el valor de ambos campos se obtiene la dirección:

(3, 400)

Esta dirección hace referencia a la palabra 400 dentro del segmento 0. En este caso, el desplazamiento supera el tamaño del segmento ( $400 > 356$ ), por lo que se trata de una dirección lógica no válida.

**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 1 MiB. Se pide:**

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

Una 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.

Para hallar el tamaño  $n$  en bits de una dirección física, hay que resolver la siguiente desigualdad:

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

$$C_{MP} = \frac{2^{20} \text{ bytes}}{1 \text{ byte} / \text{palabra}} = 2^{20} \text{ palabras} \Rightarrow n = 20 \text{ bits}$$

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:

$$\min_d \left\{ S_P \leq 2^d \right\}$$

$$S_P = \frac{2^{10} \text{ bytes}}{1 \text{ byte} / \text{palabra}} = 2^{10} \text{ palabras} \Rightarrow d = 10 \text{ bits}$$

El tamaño  $f$  del campo número de marco de página se obtiene así:

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

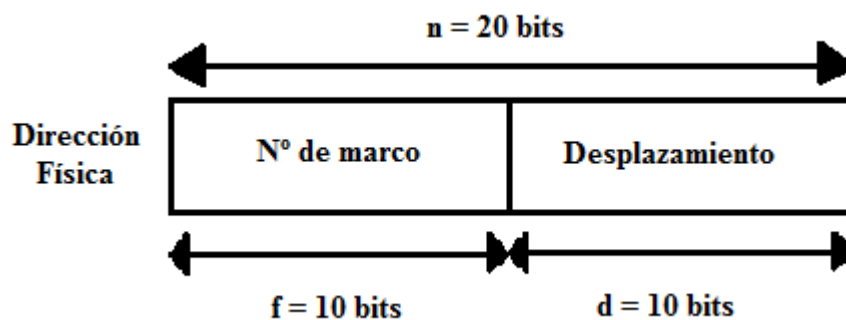
El tamaño  $m$  en bits de una dirección virtual se halla resolviendo la siguiente desigualdad:

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

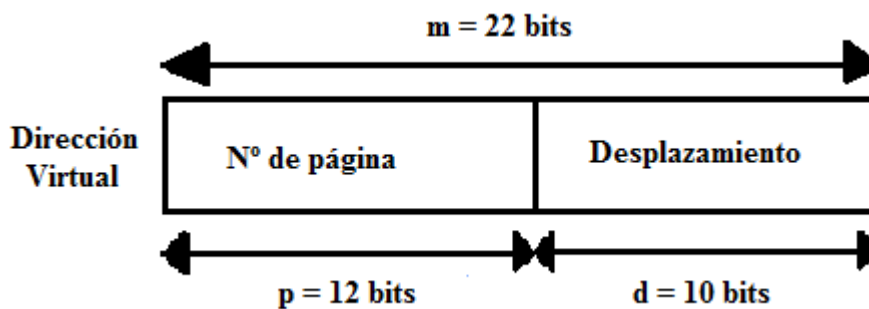
$$C_A = \frac{2^{22} \text{ bytes}}{1 \text{ byte} / \text{palabra}} = 2^{22} \text{ palabras} \Rightarrow m = 22 \text{ bits}$$

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 obtiene así:

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



Formato de una dirección física.



Formato de una dirección virtual.

**b) 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 proceso de mayor tamaño que se puede ejecutar es el que ocupa toda la memoria virtual, que son:

$$4 \text{ MiB} = \frac{2^{22} \text{ bytes}}{2^{10} \text{ bytes} / \text{página}} = 2^{12} \text{ páginas}$$

La tabla de páginas debe tener una entrada por cada página del proceso, por lo tanto, constará de  $2^{12}$  entradas. En un sistema cuya política de gestión de memoria es del tipo demanda de página, una entrada de la tabla de páginas debe tener como mínimo 2 campos: número de marco de página que, como hemos visto en el apartado anterior, ocupa 10 bits, y un bit de validez. Por lo tanto, el tamaño mínimo de una entrada es de  $10 + 1 = 11$  bits. Así pues, la capacidad mínima que debe tener la tabla de páginas del proceso de mayor tamaño que se puede ejecutar en el sistema es:

$$2^{12} \text{ entradas} \times 11 \text{ bits} / \text{entrada} = 45056$$

El tanto por ciento de la memoria principal que ocuparía esta tabla es:

$$\frac{11 \times 2^{12}}{8 \times 2^{20}} \times 100 = \frac{1100}{2048} = 0.54 \%$$