

SISTEMAS OPERATIVOS

Solución PED2 (Enero 2013)

Solución Ejercicio 1

- I) Los enlaces simbólicos consumen más espacio que los enlaces duros ya que ocupan bloques de datos mientras que los enlaces duros ocupan solo entradas de directorios. En conclusión la afirmación es **FALSA**.
- II) La afirmación es **VERDADERA** ya que una de las principales ventajas de la paginación simple es que permite compartir entre varios procesos un código común, con el consiguiente ahorro de memoria principal que esto supone. Si el código de un programa es reentrantante puede ser compartido por varios procesos, de tal forma que solo es necesario tener cargadas en memoria principal una copia de las páginas del código del programa independientemente del número de instancias de dicho programa (procesos) que se ejecuten.
- III) La afirmación es **VERDADERA** ya que efectivamente la capacidad de un disco duro disminuye cuando se formatea a bajo nivel. Esta reducción de la capacidad es debida a tres motivos. En primer lugar, existe una pequeña separación física entre pistas contiguas, así como entre sectores contiguos. Estas separaciones físicas ocupan un espacio que no se puede aprovechar para almacenar datos. En segundo lugar, debido a que durante el proceso de fabricación y de utilización de un disco pueden aparecer sectores defectuosos, es necesario reservar espacio para sectores de reserva. En tercer lugar, debe tenerse en cuenta que la cabecera y el ECC de un sector consumen espacio. En general, tras el formateo a bajo nivel la capacidad del disco queda reducida en torno a un 20 %.
- IV) Al usar memoria virtual el espacio de direcciones virtuales de un proceso no es necesario que esté cargado completamente en la memoria principal para que el proceso pueda ser ejecutado, ello permite tener cargados en memoria principal muchos más procesos que sino se utiliza memoria virtual y en consecuencia aumentar el grado de multiprogramación del sistema. En consecuencia la afirmación es **FALSA**.

Solución Ejercicio 2

Supuesto que el área de datos tiene una capacidad suficiente, el tamaño máximo que puede tener un archivo en este sistema de archivos está limitado por el número de bloques de datos del archivos que pueden ser direccionados directa o indirectamente a través de su nodo-i asociado.

De acuerdo con el enunciado en un nodo-i asociado a un archivo se almacenan las direcciones de sus 8 primeros bloques de datos y la dirección de un bloque de indirección simple, en el cual recuérdese se almacenan direcciones de bloques de datos.

El número de direcciones N_D de bloques de datos que caben en un bloque de indirección simple se calcula de la siguiente forma:

$$N_D = \text{floor} \left(\frac{S_B}{n} \right)$$

Donde S_B es el tamaño de un bloque de datos y n es el número n de bits que ocupa la dirección física de un bloque de disco. Ésta última se obtiene resolviendo la siguiente desigualdad:

$$\min_n \{N_B \leq 2^n\}$$

Donde N_B es el número de bloques de datos existentes. Como $N_B = 256 = 2^8$ bloques entonces $n = 8$ bits.

Sustituyendo valores y operando se obtiene:

$$N_D = \text{floor}\left(\frac{16 \text{ bytes}}{8 \text{ bits}}\right) = \text{floor}\left(\frac{16 \text{ bytes}}{1 \text{ byte}}\right) = 16 \text{ direcciones}$$

Luego en un bloque de indirección simple pueden almacenar un máximo de 16 direcciones de bloques.

En consecuencia el número total de direcciones de bloques accesibles desde un nodo- i son $8 + 16 = 24$ direcciones. Por lo tanto el tamaño máximo de un archivo queda limitado a **24 bloques**, es decir, a $24 (\text{bloques}) \cdot 16 (\text{bytes/bloque}) = \mathbf{384 \text{ bytes}}$.

Solución Ejercicio 3

La dirección Dir_F de la última palabra que se lee en la memoria principal expresada en decimal se obtiene de la siguiente forma:

$$\text{Dir}_F = \text{Dir}_{F0} + N_P - 1$$

Donde Dir_{F0} es la dirección inicial expresada en decimal y N_P es el número de palabras leídas.

Del enunciado se sabe que la dirección física inicial expresada en hexadecimal es AA00_{16} . Pasando a decimal se obtiene 43520.

Por otra parte N_P se calcula dividiendo el número de bytes leídos por el número de bytes que ocupa una palabra:

$$N_P = \text{floor}\left(\frac{1000 (\text{bytes})}{2 (\text{bytes/palabra})}\right) = 500 \text{ palabras}$$

Sustituyendo valores y operando se obtiene que dirección Dir_F de la última palabra que se lee en la memoria principal expresada en decimal es:

$$\text{Dir}_F = 43520 + 500 - 1 = 44019$$

Pasando a hexadecimal se obtiene finalmente que la dirección pedida es **ABF3**₁₆

Solución Ejercicio 4

- a) En primer lugar hay que calcular el número de páginas N_P en que se descompone el espacio de direcciones lógicas del proceso A:

$$N_P = \text{ceil}\left(\frac{C_X}{S_P}\right)$$

Donde C_X es el tamaño del espacio de direcciones lógicas del proceso A y S_P es el tamaño de una página. Sustituyendo valores y operando se obtiene que:

$$N_P = \text{ceil}\left(\frac{50}{4}\right) = \text{ceil}(12,5) = 13 \text{ páginas}$$

Cada entrada de la tabla de páginas del proceso A estará asociada a una página i ($i = 0, 1, 2, \dots, 12$) del espacio de direcciones lógicas del proceso y almacenará, entre otras informaciones, el marco de página j de la memoria principal donde se encuentra cargada la página i . Luego la tabla de páginas del proceso A constará de **13 entradas**.

Examinando el mapa se observa que los primeros 13 marcos vacíos son $j=1, 3, 23, 24, 26, 27, 28, 29, 30, 32, 34, 36$ y 38 . Luego serán en estos marcos donde se carguen las páginas $i=0, \dots, 12$ del proceso A. En conclusión el contenido de la tabla de páginas del proceso A una vez que éste se haya cargado en memoria será el siguiente:

	Marco j	Otros campos
$i=0$	1	
$i=1$	3	
$i=2$	23	
$i=3$	24	
$i=4$	26	
$i=5$	27	
$i=6$	28	
$i=7$	29	
$i=8$	30	
$i=9$	32	
$i=10$	34	
$i=11$	36	
$i=12$	38	

Puesto que el espacio de direcciones virtuales del proceso A ocupa 12,5 páginas, la última página del proceso ($i = 12$) solo estará ocupada hasta la mitad, la otra mitad estará libre. Luego la magnitud de la fragmentación interna es de **2 KiB**.

[illegible]