

EJERCICIOS PARTE 3 – MAYO 2016

1. El sistema de ficheros de una tarjeta flash está basado en FAT16 con tamaño de bloques de hasta 32 KB.
 - a. ¿Cuál es tamaño máximo permitido para las tarjetas flash?
 - b. En una tarjeta de 1GB con bloques de tamaño 32 KB, ¿Cuál es el tamaño de la tabla FAT?

En una tarjeta flash con tamaño de bloque de 32KB y FAT16, la información de un directorio presente en la tarjeta flash y una fracción de la tabla FAT se representan a continuación:

Nombre Fichero	Primer Bloque	Longitud en Bytes
Image052.jpg	7	126*1024
Image053.jpg	4	828
Image054.jpg	9	80*1024

0	MBR
1	Arranque
2	Superbloque
3	14
4	EOF
5	EOF
6	5
7	3
8	EOF
9	6
10	LIBRE
11	LIBRE
12	LIBRE
13	DAÑADO
14	8
15	LIBRE
...	
....	

- c. ¿Qué bloques ocupa cada uno de los ficheros especificados en el directorio dado?
- d. ¿Qué tipo de fragmentación se produce? Explica por qué.
- e. ¿Qué cantidad de fragmentación se produce en cada uno de los ficheros especificados?
- f. Se quiere acceder a la posición 102400 (en bytes) del fichero Imagen052.jpg. Sabiendo que el sistema permite el acceso aleatorio y la tabla FAT está completamente cargada en memoria, ¿qué bloques se deberían cargar en memoria para acceder a dicha posición? Explica por qué.

SOLUCIÓN

- a. Con FAT16, los punteros de dirección son de 2 Bytes. Es posible direccionar 2^{16} bloques. Como los bloques son de hasta un máximo de 32 KB, el tamaño máximo de una tarjeta flash es de 2 GB. Nota: El problema que tiene este sistema de ficheros es que los bloques de memoria son muy grandes (32 KB), se desperdicia mucho espacio en fragmentación interna.
- b. En una tarjeta de 1GB con bloques de 32 KB se tienen 2^{15} bloques. Por cada bloque se necesitan 2 Bytes (tamaño del puntero) para mantener la información de la ubicación

del bloque. Luego, el tamaño de la tabla FAT es de 64 KB ó 2 Bloques. Nota: Posibilidad de incluir toda la tabla en una caché de sólo 64 Kb, al ser los bloques tan grandes se invierte muy poco en la tabla FAT.

c.

Nombre Fichero	Primer Bloque	Longitud en Bytes	Bloques
Image052.jpg	7	126*1024	7-3-14-8
Image053.jpg	4	828	4
Image054.jpg	9	80*1024	9-6-5

- d. Se produce fragmentación interna. Los bloques son de tamaño fijo y se pierde el espacio que no se llena del último bloque de un archivo. Es especialmente importante cuando los tamaños de los bloques son grandes, como es el caso que nos ocupa. Un fichero de 1 KB y otro de 10 KB ocuparán lo mismo 32KB.

e.

Nombre Fichero	Primer Bloque	Longitud en Bytes	Fragmentación
Image052.jpg	7	126*1024	2 KB
Image053.jpg	4	828	32KB-828B
Image054.jpg	9	80*1024	16KB

- f. La posición 102400 (en bytes) está en el bloque cuarto del archivo Image052.jpg. Por tanto, es necesario subir el bloque 8 de la tarjeta flash a memoria. El acceso es aleatorio y la tabla FAT ya estaba en memoria.

2. En un sistema informático la gestión de memoria virtual está basada en segmentación paginada, con un tamaño de página de 1024 palabras y una memoria física formada por 1024 marcos. El espacio de direcciones lógicas de cada proceso puede estar dividido hasta un máximo de 16 segmentos. El tamaño máximo de cada segmento está dado por la posición 0xFFFF en palabras.

Se pide:

- a) Tamaño máximo del espacio de direccionamiento virtual de un proceso expresado en palabras y descripción de una dirección virtual o lógica desde el punto de vista del programador y desde el punto de vista del sistema indicando, en cada caso, el número de bits que ocupa cada uno de los campos.

Solución:

La dirección virtual está formada por 24 bits. El tamaño máximo del espacio de direccionamiento virtual será, por tanto, 2^{24} palabras.

Punto de vista del programador: 4 bits para los segmentos y 20 bits para el offset del segmento.

Punto de vista del sistema: 4 bits para los segmentos + 10 bits para las páginas dentro de cada segmento + 10 bits para el offset dentro de la página.

- b) Tamaño de la memoria física expresado en palabras y descripción de la dirección física indicando el número de bits que ocupa cada uno de los campos.

Solución:

La dirección física está formada por 20 bits, 10 bits para el número de marcos (1024 marcos) y 10 bits para el desplazamiento dentro del marco.

Tamaño de la memoria física: 2^{20} palabras.

Un proceso genera la siguiente secuencia de referencias lógicas expresadas en hexadecimal y en formato (segmento, offset dentro del segmento) = (0, 0x0300):

(0, 0x0300) (0, 0x1200) (1, 0x0058) (0, 0x1112) (1, 0x1048) (2, 0x0354)

(3, 0xA0F3) (0, 0x1035) (0, 0x1036) (1, 0x2050) (1, 0x0128) (3, 0x80AF)

Sabiendo que el sistema cumple las siguientes características:

- Paginación bajo demanda
- Asignación fija de marcos
- Algoritmo de reemplazo Least Recently Used

Al proceso se le asigna 4 marcos en orden: 0x0120 (primer marco asignado), 0x0156, 0x0266 y 0x0388 (último marco asignado).

Asumiendo que las tablas de segmentos y las de páginas asociadas a cada segmento no entran dentro del cómputo de marcos asignado al proceso (ocupan espacio en la zona reservada al sistema operativo).

- c) Rellenad la tabla de referencias expresando en las celdas la parte relativa al segmento y la página de la dirección lógica o virtual. En la última fila indicad los fallos de página (I= carga inicial, R=reemplazo). En la tabla se muestra, a modo de ejemplo, el resultado del primer acceso.

marco	Direcciones lógicas referenciadas					
	(0, 0x0300)	(0, 0x1200)	(1, 0x0058)	(0, 0x1112)	(1, 0x1048)	(2, 0x0354)
0x120	(0, 0x00)	(0, 0x00)	(0, 0x00)	(0, 0x00)	(0, 0x00)	(2, 0x00)
0x156		(0, 0x04)	(0, 0x04)	(0, 0x04)	(0, 0x04)	(0, 0x04)
0x266			(1, 0x00)	(1, 0x00)	(1, 0x00)	(1, 0x00)
0x388					(1, 0x04)	(1, 0x04)
Fallos Pag.	I	I	I		R	R

	Direcciones lógicas referenciadas					
marco	(3, 0xA0F3)	(0, 0x1035)	(0, 0x1036)	(1, 0x2050)	(1, 0x0128)	(3, 0x80AF)
0x120	(2, 0x00)	(2, 0x00)	(2, 0x00)	(2, 0x00)	(1, 0x00)	(1, 0x00)
0x156	(0, 0x04)	(0, 0x04)	(0, 0x04)	(0, 0x04)	(0, 0x04)	(0, 0x04)
0x266	(3, 0x28)	(3, 0x28)	(3, 0x28)	(3, 0x28)	(3, 0x28)	(3, 0x20)
0x388	(1, 0x04)	(1, 0x04)	(1, 0x04)	(1, 0x08)	(1, 0x08)	(1, 0x08)
Fallos Pag.	R			R	R	R

- d) Según la tabla de referencias anterior, calcular las direcciones físicas para las 6 últimas referencias.

Dirección lógica (seg., offset del seg.)	Dirección física (en hexadecimal)
(3, 0xA0F3)	0x Fallo de Página
(0, 0x1035)	0x55835, (marco=0x156, offset=0x035)
(0, 0x1036)	0x55836, (marco=0x156, offset=0x036)
(1, 0x2050)	0xE2050, (marco=0x388, offset=0x050)
(1, 0x0128)	0x48128, (marco=0x120, offset=0x128)
(3, 0x80AF)	0x998AF, (marco=0x266, offset=0x0AF)

Las direcciones físicas están formadas por 20 bits (10 página + 10 offset) = 5 bytes