

3. [2pts] Considérense los directorios /drives/f y /drives/g que se encuentran montados sobre dos volúmenes con tamaño de bloque 2KBytes y 4KBytes respectivamente. El sistema de ficheros de ambos volúmenes se basa en i-nodos (indexado) donde cada i-nodo contiene 5 punteros directos a bloques de datos y uno indirecto simple. Los punteros son de tamaño 16 bits.

La gestión de bloques libres se realiza mediante un mapa de bits (cada bit representa un bloque: a 0 el bloque está libre, a 1 el bloque está ocupado). La siguiente imagen muestra un trozo del estado de los mapas de bits para cada volumen (los bits por debajo de 20 están todos a 1, indicando que los bloques por debajo del 20 están ocupados y por encima del bit 35 están todos a 0 indicando que los bloques del 36 en adelante están todos libres):

/drives/f:

	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	
...	1	0	0	1	0	1	0	1	1	0	0	0	0	1	0	1	...

/drives/g:

	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	
...	1	0	0	1	0	1	0	1	1	0	0	0	0	1	0	1	...

Se pide:

- a) Determinar el tamaño de fichero máximo posible para cada uno de los volúmenes.

Tamaño máximo de fichero en /drives/f: Tamaño máximo de fichero en /drives/g:

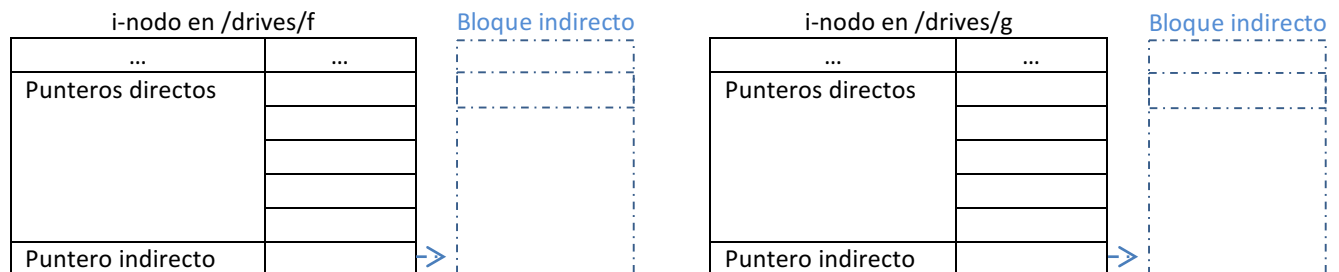
- b) Si se crea un fichero nuevo de 13 KBytes en cada volumen, determinar el estado (contenido) de los mapas de bits de ambos volúmenes y del i-nodo asociado al fichero en cada volumen tras la creación del fichero (Asignar primero los bloques con número de bloque menor. No es necesario asignar espacio al i-nodo, que reside en una zona específica del volumen. Sólo en caso necesario, asignar espacio al Bloque indirecto y consignar su contenido).

/drives/f:

	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	
...																	...

/drives/g:

	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	
...																	...



- c) Evaluar el impacto de la fragmentación interna del fichero creado en la cuestión anterior en cada volumen.

Frag. Interna del fichero en /drives/f: Frag. Interna del fichero en /drives/g:

4. [2pts] Se analiza un sistema de memoria virtual, como el estudiado en clase, direccionado a nivel de byte:

- el espacio de memoria físico abarca 4 GBytes
- las tablas se hayan paginadas en fragmentos de tamaño de una página
- todas las entradas de todas las tablas de página son de la misma longitud
- la traducción se realiza en 3 niveles, y se utiliza el siguiente esquema de dirección lógica:

8 bits (1 ^{er} nivel)	10 bits (2º nivel)	? bits (3º nivel)	? bits (offset)
--------------------------------	--------------------	-------------------	-----------------

Contesta a las siguientes cuestiones:

- a) Sabiendo que en el espacio físico hay 2^{20} marcos, determinar el tamaño de cada entrada de la tabla de páginas.
- b) Determinar el número de bits de los campos que se han indicado con “?”
- c) Si se realizara la traducción en un solo nivel, ¿cuál sería el número de entradas de la tabla de páginas?
- d) En la figura adjunta se muestra parte del contenido de la memoria física así como de la TLB. Se supondrá una TLB de 2 posiciones. La tabla raíz (nivel 1) está en el marco cero, esto es, PTBR=0. Se pide completar la siguiente tabla indicando la información que se pide para cada una de las referencias lógicas de memoria. Indíquese con “Desconocido” aquellas traducciones que no se pueden realizar con los datos proporcionados. Déjense en blanco aquellas celdas de la tabla que no procedan.

Dirección lógica	Desconocida/ Válida/Inválida	¿Fallo de TLB? (sí/no)	¿Fallo de página? (sí/no)	Dirección física en formato #pág.física:offset	Contenido (en hexadecimal)
1:2:3:4					
1:0:2:8					
4:2:1:0					
2:2:0:4					
0:2:3:12					

CONTENIDO DEL SISTEMA DE MEMORIA (PROBLEMA 4)

Notas:

Bits de control: V=bit válido, P=bit de presencia.

Los marcos ocupados por páginas de la T.P. se indican con la terna (V, P, marco_destino).

En los marcos con páginas del proceso se indica el contenido.

Los valores precedidos de 0x están expresados en hexadecimal y los que no en decimal.

Las celdas o campos sin valor conocido se hayan en blanco, o indicadas con ? ó -1.

Memoria física: (PTBR = 0)

Frame 0	
Offset	Content
0	(V=1, P=1, Frame=1)
4	(V=1, P=1, Frame=6)
8	(V=1, P=1, Frame=9)
12	(V=1, P=0, Frame=-1)
...	

Frame 5	
Offset	Content
0	0x88400000
4	0x00000000
8	0x00002208
12	0xaaab2a1b
...	

Frame 10	
Offset	Content
0	(V=1, P=1, Frame=11)
4	(V=1, P=1, Frame=12)
8	(V=1, P=0, Frame=-1)
12	(V=0, P=1, Frame=-1)
...	

Frame 1	
Offset	Content
0	(V=1, P=1, Frame=2)
4	(V=0, P=1, Frame=-1)
8	(V=1, P=1, Frame=3)
12	(V=1, P=0, Frame=-1)
...	

Frame 6	
Offset	Content
0	(V=1, P=1, Frame=7)
4	(V=1, P=0, Frame=-1)
8	(V=0, P=1, Frame=-1)
12	(V=1, P=0, Frame=-1)
...	

Frame 11	
Offset	Content
0	0x200da380
4	0x00000200
8	0x10904904
12	0x00120024
...	

Frame 2	
Offset	Content
0	(V=1, P=0, Frame=-1)
4	(V=0, P=1, Frame=-1)
8	(V=1, P=0, Frame=-1)
12	(V=0, P=0, Frame=-1)
...	

Frame 7	
Offset	Content
0	(V=1, P=1, Frame=8)
4	(V=0, P=0, Frame=-1)
8	(V=1, P=0, Frame=-1)
12	(V=1, P=0, Frame=-1)
...	

Frame 12	
Offset	Content
0	0x8188ab00
4	0x00002800
8	0x800c1a01
12	0x520022d0
...	

Frame 3	
Offset	Content
0	(V=1, P=0, Frame=-1)
4	(V=1, P=0, Frame=-1)
8	(V=1, P=1, Frame=4)
12	(V=1, P=1, Frame=5)
...	

Frame 8	
Offset	Content
0	0xc0000001
4	0xab080000
8	0x13819a8a
12	0x01010000
...	

Frame 4	
Offset	Content
0	0x36032106
4	0x80480ac5
8	0x81685485
12	0x70000600
...	

Frame 9	
Offset	Content
0	(V=0, P=0, Frame=-1)
4	(V=1, P=0, Frame=-1)
8	(V=1, P=1, Frame=10)
12	(V=1, P=0, Frame=-1)
...	

TLB:

Página lógica	Marcos (pág. física)	V
4:2:1	12	1
3:2:2	18	1