



Boletín 6

Ejercicios sobre gestión de memoria

July 15, 2016

1. Sea un sistema que emplea un mecanismo de **paginación** para gestionar la memoria. Las páginas son de 2 KB y una entrada de la tabla de páginas ocupa 32 bits. El espacio de direccionamiento lógico es 4 GB y el espacio físico, 64 GB. Se pide:
 - a) ¿Cuál es el formato de las direcciones lógicas?
 - b) ¿Cuál es el formato de las direcciones físicas?
 - c) ¿Cuántas entradas caben en una página de la tabla de páginas?
 - d) ¿Cuánto ocupa la tabla de páginas de un proceso de 64 MB?
2. Sea un procesador con direcciones lógicas de 32 bits, con un sistema de **paginación** para gestionar la memoria, un tamaño de página de 4KB y en el que cada entrada de la tabla de páginas ocupa 4 bytes.
 - a) ¿Cuál es el formato de una dirección lógica?
 - b) Si a cada proceso se le asigna todo el espacio lógico direccionable, ¿cuánto ocuparía la tabla de páginas de cada proceso?
 - c) ¿Te parece admisible un sistema en el que cada proceso necesite una tabla de páginas del tamaño obtenido en el apartado anterior?

3. Sea un sistema que emplea un mecanismo de **paginación** para gestionar la memoria. La memoria física disponible para procesos de usuario es de 2 MB y está dividida en 1024 marcos. En un instante determinado el estado de la tabla de páginas de un proceso es el siguiente:

# página	# marco
0	0x2A5
1	0x33F
2	0x142
3	0x2BC

Se pide:

- ¿Cuál es el formato de las direcciones físicas?
 - ¿Qué dirección física genera la dirección lógica 0x175C del proceso?
 - ¿Qué dirección lógica ha generado la dirección física 0x15E6FB?
4. Sea un sistema que emplea un mecanismo de **segmentación paginada** para gestionar la memoria. Cada proceso puede tener un máximo de 32 segmentos con un máximo de 512 páginas cada uno. La memoria central tiene un tamaño de 16 MB y está organizada en 128 marcos. En un instante determinado se está ejecutando en el sistema un proceso con 3 segmentos que ocupan, respectivamente, 140 KB, 120 KB y 300 KB. La tabla de páginas de dicho proceso contiene la siguiente información:

marco
0x3A
0x3D
0x3F
0x3B
0x3E
0x3C

Se pide:

- ¿Cuál es el tamaño de una página?
 - ¿Cuál es el formato de las direcciones lógicas?
 - ¿Cuál es el formato de las direcciones físicas?
 - ¿Qué dirección física genera la dirección lógica 0x04017B04 del proceso?
 - ¿Qué dirección lógica del proceso ha generado la dirección física 0x7DA2B4?
5. En un sistema con memoria virtual, ¿cuál de los siguientes aspectos no es función del SO?
- Traducir las direcciones virtuales a físicas.
 - Asignar memoria física.
 - Gestionar que los procesos compartan memoria.

- d) Asignar memoria virtual.
6. ¿Cuál de las siguientes afirmaciones es cierta?
- Un fallo de página se traduce en una interrupción, por lo que se trata al final de la ejecución de la instrucción en curso.
 - El tamaño de página y el de marco coinciden.
 - Al aumentar el grado de multiprogramación siempre aumenta el porcentaje de utilización de la CPU.
 - Cada vez que hay un fallo de página se ejecuta el algoritmo de reemplazo.
7. En un sistema con memoria virtual, ¿cuál de los siguientes aspectos no es función del SO?
- Tratar las violaciones de dirección (accesos a direcciones no asignadas).
 - Tratar los fallos de TLB.
 - Tratar los fallos de página.
 - Tratar las violaciones de protección (accesos no permitidos a direcciones asignadas).
8. Sea un sistema que emplea un mecanismo de gestión de memoria mediante **segmentación paginada**. La memoria física disponible para procesos de usuario es de 16 MB y está dividida en 1024 marcos. Cada proceso puede tener un máximo de 8 segmentos con un máximo de 16 páginas cada uno. En dicho sistema se está ejecutando un proceso que está dividido en 2 segmentos, S0 y S1, de tamaño 30 y 40 KB, respectivamente. En un instante determinado el estado de la tabla global de páginas del proceso es el siguiente:

# página	# marco
0x2F	1
0x2E	1
0x7A	1
0x7B	1
0x7A	0

Se pide:

- ¿Cuál es el tamaño de página?
 - ¿Cuál es el formato de las direcciones lógicas?
 - ¿Qué dirección física genera la dirección lógica 0x46F8A del proceso?
 - ¿Qué fragmentación interna se está generando en ese instante?
9. Sea un sistema que emplea un mecanismo de gestión de memoria mediante **segmentación paginada**. La memoria física disponible para procesos de usuario es de 1 MB y está dividida en 1024 marcos. Cada proceso puede tener un máximo de 8 segmentos con un máximo de 16 páginas cada uno. En dicho sistema se está

ejecutando un proceso que está dividido en 3 segmentos, S0 (que puede producir como máximo una fragmentación interna de 24 bytes), S1 (que no puede generar fragmentación) y S2. En un instante determinado el estado de la tabla global de páginas del proceso es el siguiente:

# marco	V
0x1B	0
0x0A	1
0x0A	0
0x0C	1
0x1B	0
0x1B	1

La tabla de páginas de S0 comienza en la entrada 0 de la tabla, la de S1 en la 2 y la de S2 en la 5. Se pide:

- ¿Cuál es el tamaño de los tres segmentos si la fragmentación interna que está generando el proceso en ese instante es de 536 bytes? Recuerda que solo puede generar fragmentación interna la última página de cada segmento y solamente si esta está cargada en memoria central en ese instante.
 - ¿Qué dirección lógica ha generado la dirección física 0x02BFF?
 - Justo antes de referenciar esa dirección se ha accedido a una posición de la página 0 del segmento 2. Y, justo antes de ella, a una posición de la página 1 del segmento 1. Si a continuación se accede a la dirección lógica 0x02FA del proceso, ¿qué dirección física le corresponderá si se utiliza un algoritmo LRU de reemplazo local de páginas?
- En cada entrada de la tabla de páginas hay un bit que indica si la página asociada ha sido o no modificada respecto a la copia que hay de ella en el dispositivo de intercambio. Se trata del bit “modificado” o “dirty”.
 - ¿Por qué ese bit solo tiene significado cuando el correspondiente bit de presencia vale 1?
 - ¿Qué relación hay entre este bit y el rendimiento del sistema?
 - ¿Por qué es importante elegir un buen algoritmo de reemplazo de páginas?
 - Indica de forma justificada si los esquemas de paginación, segmentación y segmentación paginada generan o no fragmentación interna y/o externa. Considera tanto el caso de contemplar memoria virtual como el de no contemplarla.
 - ¿Por qué el SO gestiona mejor la memoria principal con los esquemas de paginación y de segmentación paginada que con un esquema basado en segmentación pura? Para responder piensa en las operaciones que ha de realizar el SO para asignar y liberar memoria central a los procesos.

14. Sea un sistema que emplea un mecanismo de gestión de memoria mediante **segmentación paginada**, con tamaño de página de 32 KB. Cada proceso puede tener un máximo de 8 segmentos con un máximo de 16 páginas cada uno. En dicho sistema se está ejecutando un proceso que está dividido en 2 segmentos, S0 y S1, de tamaño 90 y 50 KB, respectivamente. En un instante determinado el estado de la tabla global de páginas del proceso es el siguiente:

# marco	V
0x80	0
0x83	1
0x8F	1
0x80	1
0x8F	0

Se pide:

- ¿Qué fragmentación interna se está generando en ese instante?
 - ¿Qué dirección lógica ha generado la dirección física 0x47FDA2?
 - Justo antes de esa referencia se ha invocado la siguiente secuencia de accesos a páginas (1, 0), (0, 1), (1, 1), (1, 0), (0, 1), (1, 0), en la que cada referencia tiene el formato (segmento, página). Tras acceder a la dirección física 0x47FDA2 del apartado anterior se referencia a las siguientes direcciones lógicas: 0x8F4F4 y 0x0AD2C. ¿A qué dirección física se ha accedido con la última dirección lógica (0x0AD2C) si la política de reemplazo de página víctima sigue un algoritmo LRU?
Para resolver este apartado ten en cuenta que el proceso tiene asignados 3 marcos (0x80, 0x83 y 0x8F) para albergar las páginas que va referenciando el proceso. Y has de seguir los siguientes pasos:
 - Observa qué página y qué segmento están asignados a cada uno de los marcos del proceso justo tras generar la dirección física 0x47FDA2 del apartado anterior.
 - Calcula el segmento y página asociado a la dirección 0x8F4F4 que se referencia tras 0x47FDA2.
 - Si esa página no está cargada en memoria central, deberá aplicarse el algoritmo LRU para reemplazar una de las páginas que albergan los 3 marcos. Para ello has de tener en cuenta los pares (segmento, página) que se han referenciado previamente y que te indica el enunciado. Es decir, has de obtener una tabla parecida a la de la transparencia 133 (pero considerando que ahora tenemos solo un proceso).
 - Repite los pasos ii) y iii) para la dirección lógica 0x0AD2C.
 - Con el número de marco asignado a la dirección lógica 0x0AD2C del apartado anterior ya puedes obtener la dirección física asociada a dicha dirección.
15. ¿Qué ventajas tiene la paginación multinivel frente a un sistema de paginación con un único nivel de tablas de páginas? Consulta la información que aparece al res-

pecto en las transparencias y en el libro de Carretero que encontrarás en el Aula Virtual.

16. Cierta arquitectura de procesador implementa memoria virtual mediante paginación con direcciones lógicas de 32 bits. Utiliza páginas de 64 KB y **dos niveles de tablas de páginas**. El tamaño de las entradas de las tablas de nivel 0 –el que apunta a nuevas tablas de páginas– y de nivel 1 –el que apunta a páginas físicas– es de 64 bits ¹. Además las tablas de nivel 1 ocupan exactamente una página de memoria –64 KB– ².

Para este sistema se desea saber:

- a) Los campos en que se descompone una dirección lógica.
 - b) El máximo tamaño de memoria que pueden ocupar las tablas de páginas de un proceso ³.
 - c) Los bits de control y estado necesarios en las entradas de las tablas de página para implementar eficientemente memoria virtual, copia en escritura y protección de memoria ⁴.
 - d) Si las direcciones físicas ocupan 40 bits, el tamaño de los marcos de página y la máxima cantidad de memoria direccionable por cada proceso.
17. Cierta arquitectura de procesador implementa un sistema de memoria paginada con direcciones lógicas de 36 bits y **dos niveles de tablas de páginas**. Permite dos tamaños de páginas, siendo el menor de 16 KB. Las entradas de las tablas de páginas ocupan 64 bits, estando reservados los 14 de menor peso para bits de estado y atributos de las páginas. Las tablas de ambos niveles ocupan siempre 16 KB ⁵.
- a) Indica el tamaño de las páginas mayores, sabiendo que su número de marco se obtiene directamente de las entradas de las tablas del primer nivel ⁶.
 - b) Indica los campos en que se descompone la dirección lógica para cada uno de los tamaños de página.
 - c) Dada la dirección lógica $0x4F6803E24$, indica el desplazamiento, el índice a la tabla de nivel 1 y el índice a la tabla de nivel 2 si accede a una página de 16 KB.
 - d) Dada la dirección lógica $0x8A66B1207$, indica el desplazamiento y el índice a la tabla de nivel 1 si accede a una página grande.

¹Recuerda que una entrada de la tabla de páginas contiene información sobre el número de marco en el que se encuentra la página correspondiente a esa entrada, así como los bits de control y estado de dicha página (presencia, copia en escritura –modificación–, referenciada, R/W/X,...). Y fíjate que el tamaño de esta entrada viene expresado en bits, no en bytes.

²Esta información te permitirá obtener el número de bits de una dirección lógica necesarios para referenciar una entrada de la tabla de páginas de nivel 1. Es decir, $|p1|$, siendo $(p0, p1, d)$ el formato de una dirección lógica.

³Para obtener esta información has de tener en cuenta, entre otras cosas, a cuántas páginas se puede referenciar desde el nivel 0 de la tabla de páginas.

⁴Mira en las transparencias qué bits se utilizan para cada una de las cosas que se piden.

⁵Esto es, la tabla de primer nivel y cada una de las de segundo nivel, a las que se accede a partir de una entrada de la tabla de primer nivel, ocupan 16 KB.

⁶Es decir, cuando se utilizan páginas del tamaño mayor, se usa un único nivel de tablas de páginas.

- e) Si la implementación del procesador utiliza direcciones físicas de 40 bits, indica el tamaño máximo de memoria direccionable por cada proceso, por el procesador, el número máximo de páginas y de marcos disponibles para cada proceso.
- f) En el caso anterior, el contenido (íntegro) de la entrada 756 (0x2F4) de la tabla de páginas 312 (0x138) de un proceso es 0x000000B7552C483A⁷. Indica una dirección física generada con esa entrada y la dirección lógica que la generó.
18. Cierta arquitectura de un procesador implementa un sistema de memoria basado en **paginación multinivel**. Las direcciones lógicas son de 32 bits y las físicas, de 36 bits. Las páginas de las tablas de páginas ocupan 2 KB y las restantes páginas, 2 MB. El tamaño de una entrada de cualquiera de las tablas de páginas es de 64 bits, de los cuales los bits más **bajos** se utilizan para almacenar el número de marco. El contenido (*íntegro*) de la entrada 0x3 de la tabla de páginas del primer nivel es 0x7FA435CD02FE73A4. Esta entrada nos lleva a una tabla de páginas de segundo nivel cuya entrada número 0xB2 contiene 0x8B21F431A0827B25. Por otro lado, el contenido (*íntegro*) de la entrada 0x6 de la tabla de páginas del primer nivel es 0x68A3C285EBAC3F7E. Esta entrada nos lleva a una tabla de páginas de segundo nivel cuya entrada número 0x9A contiene 0x73E5F8AC52EB43A5.
- ¿Cuál es el formato de las direcciones lógicas?
 - ¿Cuál es el formato de las direcciones físicas?
 - Haz un dibujo que ilustre cómo se traduce una dirección lógica en una física mediante los campos pertinentes de las sucesivas entradas de las tablas de páginas. Incluye en el dibujo los datos que proporciona el enunciado sobre las entradas de las tablas de páginas.
 - ¿Cuál es el tamaño mínimo y máximo que pueden ocupar las tablas de páginas de un proceso?
 - ¿Cuál es la dirección física asociada a la dirección lógica 0xD357A54C?
 - ¿Cuál es la dirección física en la que empieza la tabla de páginas de segundo nivel asociada a la dirección del apartado anterior?
19. Cierta arquitectura de un procesador implementa un sistema de memoria basado en **paginación multinivel**. Las direcciones lógicas son de 36 bits y las físicas, de 57 bits. Las páginas de las tablas de páginas ocupan 4 KB y las restantes páginas, 2 MB. El tamaño de una entrada de cualquiera de las tablas de páginas es de 64 bits, de los cuales los bits más **bajos** se utilizan para almacenar el número de marco. El contenido (*íntegro*) de la entrada 0x3C de la tabla de páginas del primer nivel es 0xA710F24713B48ACB. Esta entrada nos lleva a una tabla de páginas de segundo nivel cuya entrada número 0xB2 contiene 0xA7AEC01B38B2141A. Por otro lado, el contenido (*íntegro*) de la entrada 0x1A de la tabla de páginas del primer nivel es 0x68A3C2857A08F2CA. Esta entrada nos lleva a una tabla de páginas de segundo nivel cuya entrada número 0x8C contiene 0x73E5F8AA7AEC01B3.

⁷Este es el contenido que aparece en la entrada 756 de la tabla de segundo nivel, a la que se accede a partir de la entrada 312 de la tabla de primer nivel. Recuerda que los 14 bits más bajos de 0x000000B7552C483A contienen información sobre el estado de la página correspondiente a esa entrada.

- (a) Justifica el formato de las direcciones lógicas.
- (b) Justifica el formato de cada una de las posibles direcciones físicas implicadas en el mecanismo de traducción.
- (c) Haz un dibujo que ilustre cómo se traduce una dirección lógica en una física mediante los campos pertinentes de las sucesivas entradas de las tablas de páginas. Indica en el dibujo cómo se obtiene exactamente la dirección de inicio de cada tabla de páginas implicada en el proceso de traducción, la dirección de inicio de cada entrada de las tablas de páginas involucradas, la dirección de inicio del marco (final) asociado a la dirección lógica y la dirección física asociada a la dirección lógica. Incluye en el dibujo los datos que proporciona el enunciado sobre las entradas de las tablas de páginas.
- (d) ¿Cuál es el tamaño mínimo y máximo que pueden ocupar las tablas de páginas de un proceso?
- (e) ¿Cuál es la dirección lógica asociada a la dirección física `0x14F5D80367C5A42`?
- (f) ¿Cuál es la dirección física en la que empieza la tabla de páginas de segundo nivel asociada a la dirección del apartado anterior?