

Apellidos:	Nombre:
Grupo: <input type="checkbox"/> 1 <input type="checkbox"/> 2 <input type="checkbox"/> 3 <input type="checkbox"/> PCEO	DNI:

Instrucciones:

- Todas las respuestas deben escribirse con **bolígrafo**.
- Se valorará la **exactitud, completitud y brevedad** de todas las respuestas, que deberán ser **RAZONADAS**. No tendrán validez las respuestas no justificadas.
- Esta parte vale 3 puntos y, por tanto, representa el 30 % de la nota del examen final de teoría.

1. (2,10 puntos) Supongamos un ordenador con 2 GiB de RAM, memoria virtual paginada con direcciones virtuales de 36 bits, físicas de 32 bits, tamaño de página de 4.096 bytes y palabras de 32 bits. El SO emplea una tabla de páginas invertida como la vista en clase, con la función hash (de dispersión)  $h(PID, p) = (\frac{PID}{2} + p) \bmod 8$ , siendo  $PID$  el identificador del proceso que hace el acceso y  $p$  la página a la que quiere acceder. El algoritmo de reemplazo es NRU. Se pide:
- (0,10 puntos) Explica en qué campos se descomponen las direcciones virtuales y físicas.
  - (0,25 puntos) Dibuja cómo se organizaría la tabla de páginas invertida, indicando las distintas estructuras (tablas) que se emplean, y para cada una de ellas, incluyendo aquellos campos que sean necesarios a partir de la información del enunciado y justificando convenientemente su inclusión.
  - (0,25 puntos) Supón que las 3 primeras entradas en la tabla de traducción acaban de ser ocupadas por accesos de lectura, escritura y escritura por parte de los procesos 123, 321, 333 respectivamente, y en ese orden, a los que se le han asignado los marcos de página 1000, 2000 y 3000 respectivamente. Supón que en los 3 casos la función de dispersión devolvió el valor «0» y que no ha ocurrido ningún tic de reloj aún. Dibuja, justificándolo adecuadamente, cómo quedaría la tabla de páginas invertida tras los mismos (usa «X» como contenido de aquellos campos para los que no dispongas de información).
  - (1,10 puntos) Partiendo de la situación del apartado anterior, muestra cómo queda la tabla de páginas invertida tras cada una de las referencias a memoria por parte del proceso con PID 768 que se muestran en la tabla siguiente. Supón que el SO ha asignado inicialmente a este proceso 4 marcos de página (del 6660 al 6663), que utiliza una política de asignación dinámica con reemplazo global (el SO podría asignar más marcos de página al proceso si fuese necesario) y que los únicos marcos de página ocupados hasta ahora son los 3 mencionados en el apartado anterior (1000, 2000 y 3000).

Orden	Dirección	L/E
1	0x000004A44	E
2	0x211077540	E
Tic		
3	0x000404880	L
4	0x000004010	L
5	0xF47FF07F0	L
6	0x001015721	E

- (0,40 puntos) Supón que el sistema operativo hubiese utilizado un esquema de tabla de páginas de 3 niveles, donde se utiliza el mismo número de bits para PT1, PT2 y PT3. Considerando la forma en la que ha quedado la tabla de páginas invertida en el apartado 1.d, ¿cuántas tablas de páginas de 3 niveles se requerirían? ¿Cuántas traducciones habría en cada una de ellas? Justifica tus respuestas y dibuja cómo quedaría la tabla de 3 niveles que más traducciones tiene.

Nota: Para el apartado 1.d no es necesario que dibujes las tablas enteras cada vez, basta con que indiques qué entradas cambian en cada tabla y su contenido.

2. (0,40 puntos) Supón un reloj programable funcionando en modo onda cuadrada cuyo oscilador de cuarzo opera a 10 MHz, y cuyo registro de carga almacena el valor  $100.000_{10}$ . Para controlar la hora del día, el SO emplea el esquema basado en un contador de marcas de 64 bits visto en clase. Se pide:
- a) (0,20 puntos) ¿Qué valor almacenará el registro contador descendente del reloj programable el día 13-01-2020, a las 15 horas, 32 minutos, 25 segundos, 72 milisegundos?
  - b) (0,20 puntos) Si el instante de referencia se establece en las 00:00:00 horas del día 01-01-2020, la máquina arrancó el día 09-01-2020 a las 13:09:13, y ahora son las 17:06:18 horas del día 13-01-2020, ¿qué valor almacenará el registro contador de marcas en este instante?
- 

3. (0,50 puntos) Supón un disco con 4 cabezas y 8.192 cilindros, que se encuentra dividido en 2 zonas, la zona 0 con 2.048 sectores por pista y la zona 1 con 1.536 sectores por pista, ambas zonas con el mismo número de cilindros. El disco gira a una velocidad de 6.000 RPM, una búsqueda tarda 3 ms. por cada cilindro desplazado y las cabezas se encuentran inicialmente sobre el cilindro 0. Se pide:
- a) (0,15 puntos) Calcula el tiempo medio de acceso a cualquiera de los sectores del cilindro 1.121.
  - b) (0,35 puntos) Indica, de forma razonada, la posición en el disco (zona, cilindro, cabeza, sector en la pista) del sector 35.308.602 (empezando a contar todo desde 0).
-

# Soluciones

## Ejercicio 1:

La solución de cada apartado se muestra a continuación:

- a) Al ser las páginas de 4.096 bytes, el desplazamiento dentro de una página tiene un tamaño de 12 bits, por lo que los 36 bits de cada dirección virtual se descomponen de la siguiente manera:

24 bits	12 bits
Nº de página virtual	Desplazamiento

Las direcciones físicas, al tener 32 bits, se descomponen así:

20 bits	12 bits
Nº de marco	Desplazamiento

- b) Como vimos en clase, la tabla de páginas invertida se compone de dos tablas: la tabla de dispersión y la tabla de traducción.

El número de entradas de la tabla de dispersión viene determinado por la función de dispersión. En nuestro caso, esta función devuelve el resto de dividir el número de página virtual más la mitad del PID entre 8, por lo que, a lo sumo, podrá devolver 8 valores distintos, uno por cada entrada de la tabla de dispersión. Cada una de estas entradas almacenará el número de entrada de la tabla de traducción a la que apunte o «-1» si no está en uso (no apunta a ninguna entrada).

En el caso de la tabla de traducción, el número de entradas viene dado por el número de marcos de página disponibles en la memoria principal. Según el enunciado, el ordenador tiene 2 GB de memoria RAM y el tamaño de página es de  $4.096 = 2^{12}$  bytes, como ya hemos comentado. Por lo tanto, en memoria hay:

$$\frac{2^{31}}{2^{12}} = 2^{19}$$

marcos de página, que será también el número de entradas de la tabla de traducción.

Además, a partir de la información que nos da el enunciado, cada entrada de la tabla de traducción tiene los siguientes campos:

- Bit de validez, para saber si el contenido de la entrada es o no relevante.
- Número de la página virtual que se encuentra en el marco.
- PID del proceso al que pertenece la página.
- Número de marco donde se encuentra la página.
- Bit de uso o referencia (1 bit), para saber si la página ha sido usada o no. Este bit lo usará el algoritmo NRU.
- Bit de modificación (1 bit), para conocer si se ha modificado o no el contenido de la página y poder guardarla en disco en el caso de que sea expulsada. También lo usará el algoritmo NRU.
- El siguiente elemento de la lista en caso de colisiones.

Con toda esta información, la tabla de páginas invertida queda de la siguiente manera:

TD		Estado inicial							
		Tabla de traducción							
	P <sub>tr</sub>		V	NPV	PID	NMP	R	M	Sig
0	-1	0	0						
1	-1	1	0						
2	-1	2	0						
3	-1	3	0						
4	-1	4	0						
5	-1	5	0						
6	-1	6	0						
7	-1	7	0						
		8	0						
		9	0						
		10	0						
		...	...						
		$2^{19} - 1$	0						

Podemos observar cómo en la tabla de dispersión aparece un «-1» en cada entrada, indicando que no apunta a nada, y en la tabla de traducción todos los bits de validez están a «0», indicando que no hay ninguna traducción disponible.

- c) A partir de la información que nos proporcionan, la tabla de páginas invertidas, tras los 3 accesos que nos indican, quedaría de la siguiente manera:

TD		Estado tras los 3 accesos del apartado 1.c							
		Tabla de traducción							
	P <sub>tr</sub>		V	NPV	PID	MF	R	M	Sig
0	0	0	1	X	123	1000	1	0	1
1	-1	1	1	X	321	2000	1	1	2
2	-1	2	1	X	333	3000	1	1	-1
3	-1	3	0						
4	-1	4	0						
5	-1	5	0						
6	-1	6	0						
7	-1	7	0						
		8	0						
		9	0						
		10	0						
		...	...						
		$2^{19} - 1$	0						

- d) Veamos ahora cómo evoluciona la tabla de páginas invertida con cada acceso a memoria. En negrita vamos a mostrar las modificaciones realizadas en la tabla.

- Referencia: 0x000004A44 (E) → NPV=4, hash=4. Como no hay ninguna página del proceso inicialmente en memoria, este acceso produce un fallo. Asignamos el marco 6660 a la página, y las tablas quedan así:

Tras acceso a la página 4

TD		Tabla de traducción							
	Ptr		V	NPV	PID	MF	R	M	Sig
0	0	0	1	X	123	1000	1	0	1
1	-1	1	1	X	321	2000	1	1	2
2	-1	2	1	X	333	3000	1	1	-1
3	-1	<b>3</b>	<b>1</b>	<b>0x4</b>	<b>768</b>	<b>6660</b>	<b>1</b>	<b>1</b>	<b>-1</b>
<b>4</b>	<b>3</b>	4	0						
5	-1	5	0						
6	-1	6	0						
7	-1	7	0						
		8	0						
		9	0						
		10	0						
		...	...						
		$2^{19} - 1$	0						

- Referencia: 0x211077540 (E) → NPV=0x211077, hash=7. La función de dispersión nos lleva a la entrada 7 de la tabla de dispersión. Como la entrada está vacía, se produce un fallo de página. Asignamos el siguiente marco libre a la página, el 6661, y almacenamos su información en la entrada 4 de la tabla de traducción:

Tras acceso a la página 211077

TD		Tabla de traducción							
	Ptr		V	NPV	PID	MF	R	M	Sig
0	0	0	1	X	123	1000	1	0	1
1	-1	1	1	X	321	2000	1	1	2
2	-1	2	1	X	333	3000	1	1	-1
3	-1	3	1	0x4	768	6660	1	1	-1
4	3	<b>4</b>	<b>1</b>	<b>0x211077</b>	<b>768</b>	<b>6661</b>	<b>1</b>	<b>1</b>	<b>-1</b>
5	-1	5	0						
6	-1	6	0						
<b>7</b>	<b>4</b>	7	0						
		8	0						
		9	0						
		10	0						
		...	...						
		$2^{19} - 1$	0						

- Se produce un tic de reloj, lo que pone a 0 los bits R:

### Tras tic de reloj

TD		Tabla de traducción							
	Ptr		V	NPV	PID	MF	R	M	Sig
0	0	0	1	X	123	1000	<b>0</b>	0	1
1	-1	1	1	X	321	2000	<b>0</b>	1	2
2	-1	2	1	X	333	3000	<b>0</b>	1	-1
3	-1	3	1	0x4	768	6660	<b>0</b>	1	-1
4	3	4	1	0x211077	768	6661	<b>0</b>	1	-1
5	-1	5	0						
6	-1	6	0						
7	4	7	0						
		8	0						
		9	0						
		10	0						
		...	...						
		2 <sup>19</sup> - 1	0						

- Referencia: 0x000404880 (L) → NPV=0x404, hash=4. La función de dispersión nos lleva a la entrada 4 de la tabla de dispersión, la cual apunta a la entrada 3 de la tabla de traducción. Esta entrada está ocupada por la página 4 del proceso. Como no hay coincidencia en el número de página, se produce un fallo de página y también una colisión. Usamos la siguiente entrada libre de la tabla de traducción, la 5, para guardar la información de la página 404, a la que asignamos el siguiente marco libre, el 6662:

### Tras acceso a la página 404

TD		Tabla de traducción							
	Ptr		V	NPV	PID	MF	R	M	Sig
0	0	0	1	X	123	1000	0	0	1
1	-1	1	1	X	321	2000	0	1	2
2	-1	2	1	X	333	3000	0	1	-1
3	-1	3	1	0x4	768	6660	0	1	<b>5</b>
4	3	4	1	0x211077	768	6661	0	1	-1
5	-1	<b>5</b>	<b>1</b>	<b>0x404</b>	<b>768</b>	<b>6662</b>	<b>1</b>	<b>0</b>	<b>-1</b>
6	-1	6	0						
7	4	7	0						
		8	0						
		9	0						
		10	0						
		...	...						
		$2^{19}-1$	0						

- Referencia: 0x000004010 (L) → NPV=4, hash=4. La función de dispersión nos lleva a la entrada 4 de la tabla de dispersión la cual, a su vez, nos lleva a la entrada 3 de la tabla de traducción. En esta entrada se encuentra información de la página 4 a la que estamos accediendo, por lo que se produce un acierto que, simplemente, pone a 1 el bit R:

Tras acceso a la página 4

TD		Tabla de traducción							
	Ptr		V	NPV	PID	MF	R	M	Sig
0	0	0	1	X	123	1000	0	0	1
1	-1	1	1	X	321	2000	0	1	2
2	-1	2	1	X	333	3000	0	1	-1
3	-1	3	1	0x4	768	6660	<b>1</b>	1	5
4	3	4	1	0x211077	768	6661	0	1	-1
5	-1	5	1	0x404	768	6662	1	0	-1
6	-1	6	0						
7	4	7	0						
		8	0						
		9	0						
		10	0						
		...	...						
		2 <sup>19</sup> - 1	0						

- Referencia: 0xF47FF07F0 (L) → NPV=0xF47FF0, hash=0. La función de dispersión nos lleva a la entrada 0 de la tabla de dispersión que se encuentra en uso por las tres referencias a memoria que nos indicaban en el apartado 1.c. Siguiendo la lista, y puesto que ninguna de estas se corresponde con el proceso 768, se llegaría al final de la misma y se detectaría el fallo de página. Asignamos a la página el último marco libre de los inicialmente asignados, el 6663, y usamos la entrada 6 de la tabla de traducción para almacenar la información, ajustando convenientemente el puntero de la entrada 2:

Tras acceso a la página F47FF0

TD		Tabla de traducción							
	Ptr		V	NPV	PID	MF	R	M	Sig
0	0	0	1	X	123	1000	0	0	1
1	-1	1	1	X	321	2000	0	1	2
2	-1	2	1	X	333	3000	0	1	<b>6</b>
3	-1	3	1	0x4	768	6660	1	1	5
4	3	4	1	0x211077	768	6661	0	1	-1
5	-1	5	1	0x404	768	6662	1	0	-1
6	-1	<b>6</b>	<b>1</b>	<b>0xF47FF0</b>	<b>768</b>	<b>6663</b>	<b>1</b>	<b>0</b>	<b>-1</b>
7	4	7	0						
		8	0						
		9	0						
		10	0						
		...	...						
		2 <sup>19</sup> - 1	0						

- Referencia: 0x001015721 (E) → NPV=0x1015, hash=5. La función de dispersión nos lleva a la entrada 5 de la tabla de dispersión, que está vacía. Por lo tanto, se produce un fallo de página. Puesto que todos los marcos asignados inicialmente al proceso están ocupados, hay marcos de página libre (solo 7 de ellos están ocupados) y el sistema operativo emplea una política de asignación dinámica con reemplazo global, se asignaría al proceso un nuevo marco de página de entre todos los disponibles, por ejemplo el 6664. Se utilizará la entrada 7 en la tabla de traducción:

TD		Tabla de traducción							
	Ptr		V	NPV	PID	MF	R	M	Sig
0	0	0	1	X	123	1000	0	0	1
1	-1	1	1	X	321	2000	0	1	2
2	-1	2	1	X	333	3000	0	1	6
3	-1	3	1	0x4	768	6660	1	1	5
4	3	4	1	0x211077	768	6661	0	1	-1
<b>5</b>	<b>7</b>	5	1	0x404	768	6662	1	0	-1
6	-1	6	1	0xF47FF0	768	6663	1	0	-1
7	4	<b>7</b>	<b>1</b>	<b>0x1015</b>	<b>768</b>	<b>6664</b>	<b>1</b>	<b>1</b>	<b>-1</b>
		8	0						
		9	0						
		10	0						
		...	...						
		2 <sup>19</sup> - 1	0						

- e) En este caso, se requeriría una tabla de páginas por proceso. Puesto que el enunciado menciona 4 procesos (con PIDs 123, 321, 333 y 768), se necesitarían 4 tablas de páginas de 3 niveles. Las correspondientes a los procesos 123, 321 y 333 tendrían 1 sola traducción cada una (las correspondientes a los accesos mencionados en el apartado 1.c), mientras que la correspondiente al proceso 768 tendría 5 traducciones (una por cada uno de los 5 accesos listados en la tabla del enunciado).

Cada una de las tablas que se usasen en la tabla de páginas de 3 niveles tendría 256 entradas ( $2^8$ , dado que se utilizan 8 bits para indexar cada uno de los niveles). La tabla de páginas del proceso 768 quedaría tal y como se muestra en la figura 1.

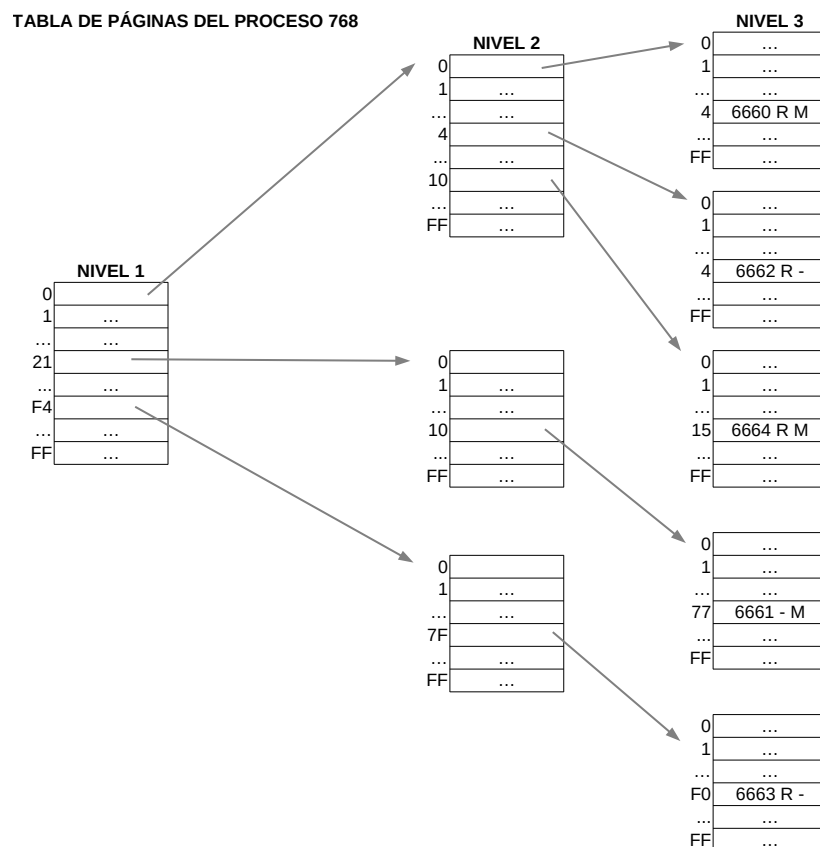


Figura 1: Tabla de páginas de 3 niveles del proceso 768.

## Ejercicio 2:

La solución de cada apartado se muestra a continuación:

- a) A partir de los datos que nos dan podemos calcular el número de tics que se producen cada segundo:



$$\frac{10 \times 10^6}{100.000} = 100 \text{ tics por segundo.}$$

De esta forma, cada segundo se producen 100 tics de reloj, o lo que es lo mismo, cada tic de reloj necesita  $\frac{1}{100} = 0,01$  segundos = 10 ms.

Por lo tanto, para calcular el valor del registro contador descendente en este momento debemos concentrarnos en los 72 ms que nos indican. Durante este tiempo se producen 7 tics de reloj ( $7 \times 10 = 70$  ms) y durante los 2 ms restantes, el contador descendente se habrá reducido 20.000 veces ( $\frac{2 \times 100.000}{10}$ ). El contador descendente quedará, pues, con valor 80.000.

- b) El registro contador de marcas de 64 bits almacena el número de tics desde el instante de referencia. A partir de los datos que nos dan, calculamos el número total de segundos desde el instante de referencia como:  $((12 \text{ días} \times 24) + 17 \text{ horas}) \times 60 + 6 \text{ min.}) \times 60 + 18 \text{ seg.} = 1.098.378 \text{ seg.}$  Multiplicando por el número de tics por segundo (100) obtenemos el valor del contador de marcas: 109.837.800

### Ejercicio 3:

La solución de cada apartado se muestra a continuación:

- a) El cilindro pedido (1.121) está en la zona 0 del disco, con lo que sus pistas tendrán 2.048 sectores. Para calcular el tiempo medio de acceso ( $t_{ma}$ ) a un sector del cilindro 1.121, calcularemos cada uno de los tres componentes que lo determinan, es decir el tiempo de búsqueda ( $t_{busc}$ ), el tiempo medio de latencia ( $t_{mlat}$ ) y el tiempo de transferencia ( $t_{trans}$ ):

$$t_{busc} = N_{cilindros} \times 3 = (1.121 - 0) \times 3 = 3.363 \text{ ms}$$

$$t_{mlat} = \frac{30}{6.000} = 5 \text{ ms}$$

$$t_{trans} = \frac{\frac{60}{2.048}}{6.000} = 0,005 \text{ ms}$$

Con lo que nos queda que:

$$t_{ma} = 3.363 + 5 + 0,005 = 3.368,005 \text{ ms}$$

- b) Para calcular la posición exacta del sector dado (35.308.602) deberemos determinar en primer lugar en qué zona del disco se encuentra. La zona 0 incluirá desde el sector 0 al sector 33.554.431 ( $(4 \times 4.096 \times 2.048) - 1$ ). Por lo tanto, se encuentra en la zona 1.

Expresamos, a continuación, el número de sector buscado como un desplazamiento desde el primer sector de la zona 1 (el sector 0 de la zona 1). Para ello, le restamos al número de sector buscado (35.308.602) el número del primer sector de la zona 1 (33.554.432): 1.754.170.

Para calcular en qué cilindro se encuentra dentro de la zona 1, dividimos entre el tamaño del cilindro en sectores y nos quedamos con el cociente de la división:  $1.754.170 \text{ div } (4 \times 1.536) = 285$ .

A partir del resto de la división anterior (3.130), y dividiendo entre el tamaño de la pista en sectores, obtenemos la cabeza con la que se operaría sobre dicho sector (cociente de esta división) y la posición en la pista correspondiente (resto de esta división). Cabeza:  $3.130 \text{ div } 1.536 = 2$ ; Sector dentro de la pista:  $3.130 \text{ mod } 1.536 = 58$ .

Así pues, la posición es (1, 285, 2, 58).