**Ejercicios de Gestión de Memoria**

1) Considere el espacio de direccionamiento lógico de 8 páginas de 1024 bytes cada una,

mapeado en una memoria física de 32 frames.

1. • ¿Cuántos bits hay en la dirección lógica?
2. • ¿Cuántos bits hay en la dirección física?
3. • En un esquema de paginación sin memoria virtual, ¿Como tendría que ser la relación entre los tamaños de las direcciones calculadas anteriormente? (dirección lógica menor, igual o mayor que dirección física)
4. 8 Paginas = 2^3 (con 3 bit selecciono la página)

Cada página tiene 1024 bytes = 2^10 (con los 10 bits me desplazo en la página)

3 bits para seleccionar la Página + 10 bits para seleccionar el desplazamiento dentro de la página

**Total bits dirección logica=13 bits**

1. 32 Frames = 2^5 = 5 bits para elegir los marcos Físicos

Cada Frame/página tiene 1024 bytes = 2^10 (con los 10 bits me desplazo en la página)

5 (seleccionar Frame)+ 10 (Seleccionar byte de la página) = 15 bits

**Total bits dirección física=15 bits.**

1. Deberían ser menor la dirección lógica que las direcciones físicas (o también igual)

**2)** Dado un sistema con una administración de memoria paginada simple (o sea, sin memoria

virtual), con 256 KiB de memoria real, 20 bits de direccionamiento lógico y páginas de 4 KiB, se

pide:

1. Tamaño máximo posible de un programa a ejecutar en el sistema (ignorando el espacio ocupado por el SO).
2. Cantidad máxima en bytes de fragmentación interna (por proceso) y externa que podría existir.
3. El tamaño máximo es de 256K porque es el tamaño máximo de memoria física que podría ocupar.
4. Datos

256 x 1024 = 262144 bytes de memoria física.

Páginas de 4K=4096 =2^12 (uso 12 bits para desplazarme por una página de 4K)

20 bits de direccionamiento lógico

Fragmentación externa: No tiene

Fragmentación interna: 3999 bytes por proceso (4K -1)

**\* “1 KiB” es el equivalente a 1024 bytes (1KB, por el contrario, formalmente representa 1000 bytes)**

**3)** Dada la siguiente tabla de segmentos, determine las direcciones físicas:

|  |  |  |
| --- | --- | --- |
| **Segmento** | **Base** | **Limite** |
| 0 | 219 | 600 |
| 1 | 2300 | 14 |
| 2 | 90 | 100 |
| 3 | 1327 | 580 |

|  |  |
| --- | --- |
| **Dirección Lógica (segmento, desplazamiento)** | **Dirección Física** |
| 0, 430 | Base + Despl. = 219+430= **649** |
| 1, 10 | 2300+10 =**2310** |
| 2, 500 | **Segmentation Fault** |

1. Verifico que el Desplazamiento sea menor que el Limite
2. Sumo el Desplazamiento a la Base del segmento correspondiente.

**SEGMENTO 0**

Base= 219

Limite=600

4) Considere una máquina con direcciones de 18 bits. Los primeros 2 bits se utilizan para identificar el segmento, y los últimos 16, el offset dentro del segmento. Asumir que las bases y los límites de la tabla de segmentos están configuradas de la siguiente forma:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Segmento** | **Base** | **Limite/Largo** | **Protección** | **Respuesta** |
| 0 | 00000h | 0ABCDh | Readonly | **Segmentation Fault (c)** |
| 1 | 1B000h | 007FFh | Execonly | **No puedo escribir (b)** |
| 2 | 1B800h | 00FFFh | Readwrite | **Escribo (a)** |
| 3 | 30000h | 01234h | Readwrite |  |

Responder en forma clara y concisa que sucede cuando el proceso intenta escribir en cada una de las siguientes direcciones virtuales: a) 20000h , b) 10000h, c) 0BEEFh, d) 00ACEh.

1. La dirección virtual 20000h lo convierto a binario me da 2 de Segmento (2bits) y 0000 desplazamiento (16bits).

Cant. de Bits Segmento + Cant. de bits desplazamiento = 18bits

La base + desplazamiento <= que Base+ Limite

1B800h + 0000h <= 1B800h +00FFFh.

1B800h<=1C7FFh

OK

1. (10000h) Segmento 1 – desplazamiento 0000

1B000h+ 0000 <= 1B000h+007FFh

1B000h<= 1B7FFh (OK)

Verifico que en ese segmento puedo realizar opción requerida

1. (0BEEFh) Segmento 0 – Desplazamiento BEEFh

000000h + 0BEEF <= 00000h+0ABCDh

La base + desplazamiento <= que Base+ Limite

0BEEF <= 0ABCDh (NO CUMPLE)

1. (00ACEh) Segmento 0 – Desplazamiento 0ACEh

00000h+ 0ACEh <= 00000h+ABCDh

0ACEh <= ABCDh

OK

Pero no permite Escribir porque es Read Only

5) En un esquema de memoria virtual, con una máquina que tiene 128 KiB de RAM y páginas de 8 KiB. ¿Cuál sería el tamaño mínimo (en bits) de dirección si queremos que un proceso pueda direccionar hasta 1 MiB?

Páginas de 8K = 8x1024= 8x2^10 (o sea 10 bits de desplazamiento)

128KiB= 128 x1024 = 131072 = 2^17

Para direccionar 1M-> 2^20

6) Suponga un esquema de memoria virtual con paginación por demanda. Se están ejecutando los programas A, B y C con longitudes totales de 2 KiB, 1.5 KiB y 3 KiB respectivamente. La longitud de la página es de 0.5 KiB.

a) ¿Cuál es la cantidad máxima de páginas que podría tener un programa si las instrucciones tienen direcciones de 16 bits y la memoria física es de 32 KiB?

b) Proponga un formato para las entradas de la tabla de páginas de un proceso y la tabla de administración de memoria libre/ocupada del sistema, suponiendo ahora que el tamaño de la memoria es 64Kb. ¿Que tamaño ocuparía en memoria la segunda tabla?

c) Determinar el contenido de las tablas de páginas para los programas A, B y C, sabiendo que el estado de los primeros 9 frames de memoria Fisica es el siguiente:

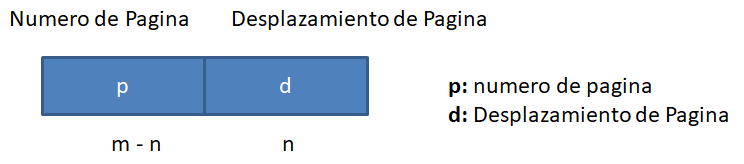
|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| A0 | B0 | C5 | A1 | A2 | C3 | C1 | B2 | C4 |

1. Tamaño de Pagina = 0,5 Kib = 0,5x1024 = 512 = 2^9

O sea 9 bits para el desplazamiento

16 bits de memoria lógica

Numero de pagina/ selector de página = 16 – 9 =7



O sea 2^7 paginas= **128 Paginas**

1. Sigo teniendo 128 paginas aunque la memoria sea de 64KiB

Por lo que tengo Tabla de Paginas por Programa de 128 entradas

c)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| A0 | B0 | C5 | A1 | A2 | C3 | C1 | B2 | C4 |

Tabla de Paginas Programa A

PA= 2K =2048 = 4 Paginas

|  |  |
| --- | --- |
| **Marco** | **Pagina** |
| 0 | 0 |
| 3 | 1 |
| 4 | 2 |
| X | 3 |

Tabla de Paginas Programa B

PB=1,5K=1536 B = 3 Paginas

|  |  |
| --- | --- |
| **Marco** | **Pagina** |
| 1 | 0 |
| X | 1 |
| 7 | 2 |

Tabla de Paginas Programa C

PC=3K= 3072 B = 6 Paginas

|  |  |
| --- | --- |
| **Marco** | **Pagina** |
| X | 0 |
| 6 | 1 |
| X | 2 |
| 5 | 3 |
| 8 | 4 |
| 2 | 5 |

7) Un esquema de memoria virtual tiene un tamaño de página de 1024 bytes y la memoria física tiene 4 marcos de página. La Tabla de páginas de un proceso es:

|  |  |
| --- | --- |
| **Página Virtual** | **Marco** |
| 0 | 3 |
| 1 | 1 |
| 2 | -- |
| 3 | -- |
| 4 | 2 |
| 5 | -- |
| 6 | 0 |
| 7 | -- |

¿Cuáles son las direcciones físicas de las siguientes direcciones virtuales (expresadas en

decimal): 0, 3728, 1024, 1025, 4099, 7800?

**Respuesta:**

Tengo 4 marcos = 2^2

Paginas de 1024 bytes = 2^10 bytes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Lógica** | **Página** | **Marco** | **Desplazamiento** | **M. Física** | |
| 0 | 0 | 3 | 0 | 3\*1024+0 | 3072 |
| 3728 | 3 | - | 656 | - | Page Fault |
| 1024 | 1 | 1 | 0 | 1\*1024+0 | 1024 |
| 1025 | 1 | 1 | 1 | 1\*1024+1 | 1025 |
| 4099 | 4 | 2 | 3 | 2\*1024+3 | 2051 |
| 7800 | 7 | - | 632 | - | Page Fault |

8) Un proceso de 8 páginas está ejecutando en un sistema de memoria virtual, con asignación fija de 4 marcos por proceso y alcance local. Si se considera que la memoria está inicialmente vacía, determinar el número de fallos de página que ocurren al referenciarse las siguientes páginas: 0, 1, 7, 2, 3, 2, 7, 1, 0, 3, 0, 2, 3, 1; bajo los siguientes algoritmos:

* Óptimo, FIFO, LRU, Clock
* Proponga para el algoritmo Clock un formato para la entrada de la tabla de páginas, sabiendo que los procesos pueden leer/escribir/ejecutar siempre y cuando los permisos de la página lo permitan, con una memoria física de 64 KiB y frames de 2 KiB.

9) Se tiene una PC con procesador Intel de 32 bits de direccionamiento que acepta el uso de páginas de 8KB de tamaño. Dicha PC cuenta, además, con un sistema operativo que utiliza paginación bajo demanda donde la política de asignación de frames es fija, siendo de 4 frames por proceso. La política de sustitución de páginas es local.

Se está ejecutando un proceso de 159KB de tamaño con la siguiente asignación actual de frames:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Puntero (si aplica)** | **Marco** | **Página (numerada desde cero)** | **Uso** | **Modificado** | **Instante de Referencia** |
|  | 1 | 14 | 1 | 1 | 28 |
|  | 3 | 17 | 1 | 0 | 3 |
|  | 5 | 19 | 1 | 1 | 15 |
| -- | 8 | -- | -- | -- | -- |

A continuación se detallan las próximas referencias a memoria (con sus respectivos modos de acceso) que el proceso realizará:

100(Lectura) – 122950(Escritura) – 98306(Lectura) – 139264(Escritura) – 122880(Lectura) –

155650(Escritura) - 172100(Lectura) - 100(Lectura)

Se pide:

Para los algoritmos LRU y Clock modificado:

a) Indicar el estado de las páginas en memoria luego de cada referencia, así como también los page faults producidos y las páginas que fueron escritas a disco. No es necesario calcular los instantes de referencia.

b) Dejando a un lado el costo de ejecución e implementación de un algoritmo de sustitución. Indicar cuál de los dos algoritmos presenta un mejor rendimiento con la secuencia de referencias dada. ¿Qué criterio tiene en cuenta para tomar esa decisión?.

**10)** Se tienen dos procesos, P1 y P2 que generan respectivamente las siguientes secuencias de referencias a memoria:

P1: 10 11 0 3 4 11 0 3 4 11 0 3 4

P2: 10 11 12 13 14 15 16 17 18

Si se le asignan 4 frames a cada proceso, y sabiendo que el mecanismo de sustitución de páginas es LRU, y que se le agrega una TLB con capacidad de 4 páginas, con algoritmo de sustitución FIFO, Indique:

a) En cada caso, ¿Cuántos accesos a la TLB, a memoria y a disco se producen?

b) ¿En alguno de los casos sirve tener una caché? Justifique. (Tenga en cuenta el concepto de localidad)

**11)** Suponga un sistema operativo con un administrador de memoria paginada cuyo tamaño de página es de 8KB. Se conoce que el sistema operativo funciona sobre un procesador cuyo tamaño de palabra es de 64 bits.

A) Calcule el tamaño máximo que puede llegar a alcanzar la tabla de página de un proceso, suponiendo que cada entrada de la tabla de páginas ocupa 128 bits.

B) Calcule cuántas páginas puede llegar a ocupar como máximo la tabla de páginas.

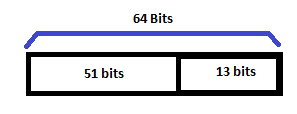
C) Dado un proceso en ejecución que requiere 7 Mbytes, calcule cuántas entradas de la tabla de páginas están siendo empleadas por dicho proceso en el sistema descrito anteriormente.

D) Calcule la cantidad de memoria Física total direccionable.

Respuesta:

1. Paginas de 8K = 2^13 bits

Tamaño de la palabra del Procesador es de 64Bits

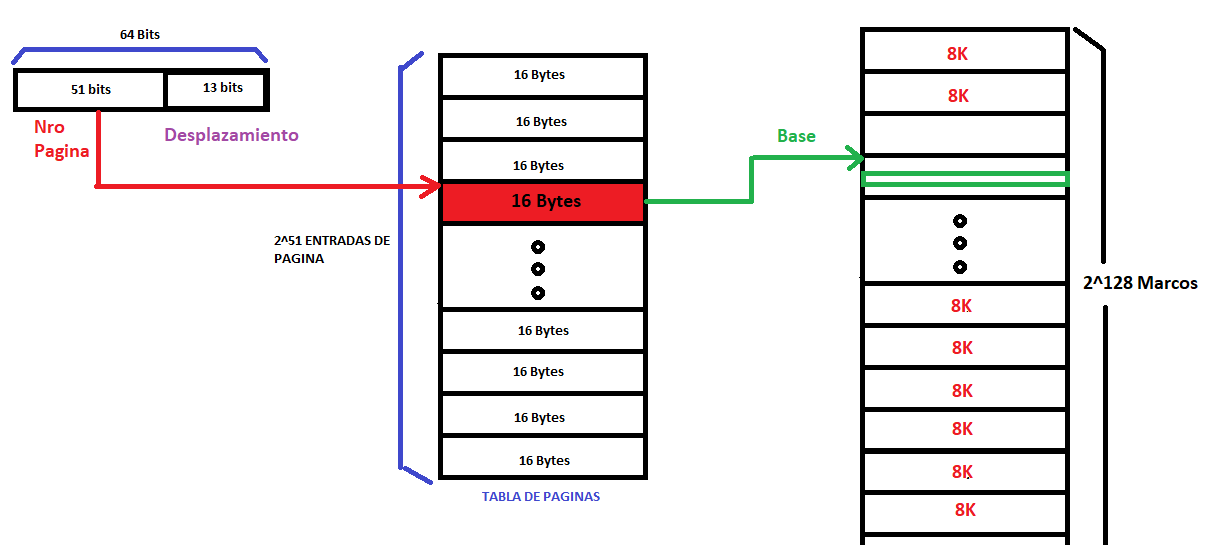


Cantidad de Marcos es = 64 – 13 = 2^51 Marcos Fisicos

Entonces el tamaño de la tabla de paginas es de 2^51 Entradas x 16 Bytes= 2^51x2^4=

**2^55 bytes = 32 Petabibyte**

1. Si una entrada de la tabla de Paginas tiene 128 bits = 16 bytes



Tamaño total de la Tabla de páginas = 2^55

Tamaño de una página = 8K

Entonces el tamaño de la tabla de páginas es de = 2^55 / 2^13 = 2^(55-13) = 2^42 Paginas =

**4T de Páginas**

1. Dado un proceso de 7MBytes, calcule cuantas entradas de la tabla de paginas están siendo utilizadas

Tamaño del Proceso / Tamaño de una pagina = Cantidad de Paginas utilizadas = Cantidad de entradas de la tabla de paginas

7 MB / 8K = 7 x 2^20 / 2^13 = 7x 2 ^7 = **896 paginas ocupa el proceso en memoria o necesita 896 entradas de pagina de la tabla de paginas**

1. Tamaño de la memoria direccionable

Si tengo 128 bits para una entrada de la tabla de paginas = 16Bytes

Con 128 bits puedo direccionar 2^128 Marcos de Memoria Fisica

Si cada Marco se corresponde con el tamaño de una pagina (8K)

2^128 Marcos 2 ^13 = **2 ^ 131 Bytes = Mucho!!!!!**

**Ejercicio 12**

Sea un sistema con gestión de memoria paginada. La capacidad máxima de direccionamiento virtual es de 4 GB y la memoria principal es de 256 MB. El desplazamiento dentro de una página y de un marco se realiza con 12 bits.

1. ¿Cuál es el tamaño de una página? ¿Y de un marco?

2. Indique el formato de una dirección de memoria virtual y de memoria principal.

3. ¿Cuál sería el número máximo de páginas que puede tener un proceso?

4. ¿Cuál sería el formato de una entrada en la tabla de páginas?

5. Indique la problemática que se plantea con respecto al tamaño de la tabla de páginas y proporcione una solución.

6. Según la solución del apartado anterior, diseñe una arquitectura para el nuevo sistema de gestión de memoria. Respete el tamaño del campo desplazamiento.

**Respuesta:**

**1. En ambos casos el campo desplazamiento es de 12 bits, con los que se pueden direccionar 212 posiciones de memoria = 4 KB, suponiendo direccionamiento a nivel de byte.**

**2. Antes de dar respuesta a la cuestión planteada es necesario establecer el tamaño de las direcciones de memoria virtual y de memoria principal. El tamaño de la memoria virtual para un proceso es de 4 GB = 2^32 bytes/proceso. El tamaño de la memoria principal para todos los procesos en el sistema es de 256 MB = 2^28 bytes. Si en ambos casos el campo desplazamiento es de 12 bits, el formato es:**

**Formato de una dirección de memoria virtual: Página (20 bits) | Desplazamiento (12 bits). Formato de una dirección de memoria principal: Marco (16 bits) | Desplazamiento (12 bits).**

**3. El número máximo de páginas que puede tener un proceso vendrá dado por la capacidad de direccionamiento del campo P de una dirección virtual. Con 20 bits pueden identiﬁcarse 2^20 páginas. Por tanto el número máximo de páginas que puede tener un proceso es 1 Mpáginas.**

**4. La tabla de páginas contiene los marcos de memoria principal, 16 bits, que identiﬁcan la parte alta de una dirección de memoria. El formato de una entrada en la tabla de páginas consta al menos del campo Marco (16 bits).**

**5. Solución de paginación simple con el esquema de memoria virtual con paginación paginada. La tabla de páginas podría tener un tamaño de: 2^20 entradas (referenciadas por las páginas del proceso) multiplicado por el tamaño de cada entrada, al menos 16 bits (2 bytes). Esto da un valor de, al menos, 220 \* 2 bytes = 2 MB.**

**6. En el caso de modificar la arquitectura para utilizar un esquema de paginación paginada, la dirección virtual estaría formada por los campos: Directorio (x bits) | Página (y bits) | Desplazamiento (12 bits), donde x+y = 20 bits. Además se tendría un tabla de directorio por proceso que apuntaría a tablas de páginas con el siguiente formato:**

**Formato tabla de directorio: Dir. TMP (28 bits). Formato tabla de páginas: Marco (16 bits).**

**Ejercicio 13**

Se ejecuta el proceso P1. La unidad de gestión de memoria utiliza la tabla de segmentos mostrada en la tabla1, donde se referencia para cada segmento de memoria virtual, la dirección de inicio en memoria principal y el límite de cada segmento.

Tabla 1: Tabla de segmentos del proceso P1.

|  |  |  |
| --- | --- | --- |
| **Segmento** | **Base (Hex)** | **Limite (Hex)** |
| 0 | 0F00 0000 | FFFF |
| 1 | 0000 7980 | 0600 |
| 2 | 0600 2000 | 0700 |
| 3 | 0300 0000 | FFFF |
| 4 | 0301 0000 | C000 |

Según esta información, responda a las siguientes cuestiones:

1. Dibuje el mapa del espacio de direccionamiento virtual y físico del proceso P1.

2. ¿Cuál es la correspondencia entre la dirección de un segmento de memoria virtual y una dirección memoria principal?

3. ¿Qué direcciones de memoria principal se corresponden las siguientes referencias a direcciones virtuales?

a) 0x0004 0202

b) 0x0004 D898

c) 0x0003 000A

d) 0x0000 0509

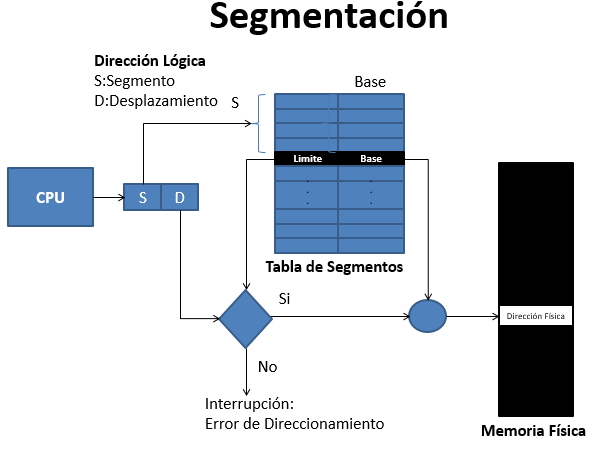
e) 0x0001 06FF

f) 0x0002 0701

g) 0x0005 0001

**Respuesta**

1. Esquema Segmentación



3.

a) 0x0004 0202 Segmento: 0x4. La base del segmento es: 0x0301 0000 y el límite 0xC000.

Desplazamiento: 0x0202 < límite del segmento. Se suma la base del segmento con el desplazamiento para obtener la dirección en memoria principal: 0x0301 0000 + 0202 = 0x0301 0202.

b) 0x0004 D898 Segmento: 0x4 < . La base del segmento es: 0x0301 0000 y el límite 0xC000. Desplazamiento: 0xD898 < límite del segmento es FALSO. Se produce una excepción por intento de acceso a una zona de memoria fuera de sus límites.

c) 0x0003 000A Segmento: 0x3 < La base del segmento es: 0x0300 0000 y el límite 0xFFFF. Desplazamiento: 0x000A < límite del segmento. Se suma la base del segmento con el desplazamiento para obtener la dirección en d) 0x0000 0509 La base del segmento es: 0x0F00 0000 y el límite 0xFFFF. Desplazamiento: 0x0509 < límite del segmento. Se suma la base del segmento con el desplazamiento para obtener la dirección en memoria principal: 0x0F00 0000 + 0509 = 0x0F00 0509.

e) 0x0001 06FF Segmento. La base del segmento es: 0x0000 7980 y el límite 0x0600. Desplazamiento: 0x06FF < límite del segmento es FALSO. Se produce una excepción por intento de acceso a una zona de memoria fuera de los límites del segmento.

f) 0x0002 0701 Segmento: 0x2. La base del segmento es: 0x0600 2000 y el límite 0x0700. Desplazamiento: 0x0701 < límite del segmento es FALSO. Se produce una excepción por intento de acceso a una zona de memoria fuera de los límites del segmento.

g) 0x0005 0001 Segmento: Es FALSO. Se produce una excepción por intento de acceso a una zona de memoria fuera de los límites del espacio de direccionamiento virtual asignado al proceso.

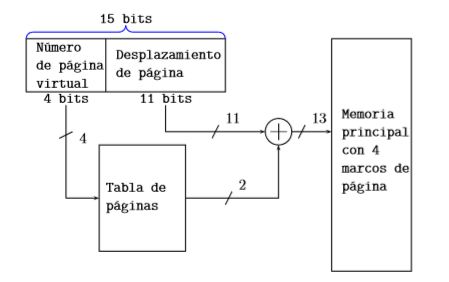
**Ejercicio 14**

Diseñar un esquema de transformación de direcciones virtuales de acuerdo con las siguientes especiﬁcaciones:

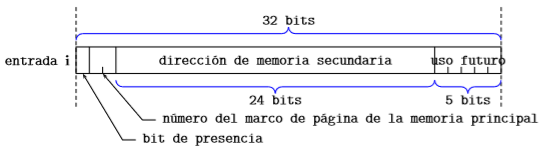
Espacio virtual direccionable = 32K palabras. Tamaño de la memoria principal = 8K palabras. Tamaño de las páginas = 2K palabras. Bus de direcciones de la memoria secundaria = 24 bits.

**Respuesta**

**Las páginas son de 2K palabras. La memoria principal admite 8K 2K = 4 páginas. La memoria virtual tiene 32K 2K = 16 páginas. Como 32K = 2^15 =⇒ las direcciones (virtuales) tienen 15 bits, de los que 11 corresponden al desplazamiento dentro de la página (2K palabras) y los 4 restantes al número de la página virtual (16 páginas virtuales).**

****

**La tabla de páginas hay que mantenerla en memoria principal ya que hace las traducciones desde las direcciones virtuales a las direcciones físicas. Sus dimensiones son: 16 entradas (o registros), uno por cada página virtual, donde se guarda el número del la página física, lo que se llama el marco de página. Además debe contener la dirección donde está la página en la memoria secundaria, por si la página no está en la memoria principal saber de dónde traerla.**

****

Ejercicio 1

**Ejercicio 15**

Se tiene un equipo con un esquema basado solo en paginación. Las direcciones virtuales son de 32 bits y las páginas son de 4 Kbytes. La memoria física del ordenador tiene 16Mbytes (cada posición es de 1 byte).

a. ¿En cuántos marcos de página se divide la memoria física?

b. ¿Cuántos bits de la dirección virtual se utilizan para identificar la página?

c. ¿Cuántos bits son necesarios para direccionar un marco de página?

d. Suponiendo que tenemos una tabla de página de un sólo nivel: ¿Cuántas entradas tiene

la tabla de páginas?

e. Grafique el esquema de paginación simple según los datos calculados.

a) Si el tamaño de página es 4K, el tamaño de marco también. Si hay

16M de memoria física:

**a)** Si el tamaño de página es 4K, el tamaño de marco también. Si hay

16M de memoria física:

16x 1024 x1024 = **4096 Marcos de Pagina**

4 x1024

**b)** Si una página es de 4K, hacen falta 12 bits como desplazamiento (u offset) en la dirección virtual (dado que 2 a la 12 = 4096). Por tanto, los bits de la dirección virtual que se utilizan para identificar la página son 32 - 12 = **20 bits.**

**c)** Para direccionar un marco hacen falta **12 bits**, considerando la memoria instalada (4096 marcos). (dado que 2 a la 12 = 4096).

**d)** Con una Tabla de Páginas de un nivel, el número de entradas es el número de páginas de memoria virtual: 2 a la 20=**1M de entradas (1.048.576 de entradas).**

**e)**

