

Tutorías con Grupos Reducidos (TGR)

Parte 5: Memoria Virtual

ESTRUCTURA DE COMPUTADORES

Grupo de Arquitectura de Computadores (GAC)

Índice

- 1 Introducción
- 2 Sistemas paginados
- 3 Sistemas segmentados
- 4 Sistemas segmentados con paginación
- 5 Ejercicios

Índice

- 1 **Introducción**
- 2 Sistemas paginados
- 3 Sistemas segmentados
- 4 Sistemas segmentados con paginación
- 5 Ejercicios

Memoria Virtual

- 1 Gestión de memoria automática que da al programador la **ilusión** de que su espacio de direccionamiento no está limitado por el espacio de memoria principal reservado a su programa (**espacio físico**), sino por el rango de direcciones que permite el sistema (**espacio virtual**).
- 2 **Espacio virtual**: Conjunto de direcciones que puede direccionar un proceso.
- 3 **Espacio físico**: Espacio de memoria principal reservado para el proceso.

Organización

- 1 Los sistemas de memoria virtual agrupan las direcciones virtuales en **bloques**.
- 2 Estos bloques son las unidades de:
 - 1 Transferencia de información entre la memoria secundaria y la principal.
 - 2 Traducción (sólo se traduce el número de bloque).

Clasificación

- 1 **Sistemas Paginados:** bloques del mismo tamaño llamados páginas.
- 2 **Sistemas segmentados:** los bloques o segmentos son de tamaño distinto.
- 3 **Sistemas segmentados con paginación:** los bloques (segmentos) son de tamaño desigual pero múltiplo de un tamaño unidad (página).

Índice

- 1 Introducción
- 2 Sistemas paginados**
- 3 Sistemas segmentados
- 4 Sistemas segmentados con paginación
- 5 Ejercicios

Sistemas paginados

- 1 Tanto el espacio virtual como el físico se dividen en bloques de un tamaño fijo P , potencia de 2, llamados páginas.
- 2 Las direcciones quedan divididas en dos campos:
 - 1 Número/Nivel de página.
 - 2 Desplazamiento.

TLB: Translation-lookaside Buffer

- 1 El TLB almacena los pares (página virtual, página física) más recientemente referenciados junto con los bits de gestión que se requieran.
- 2 Su éxito está justificado por el principio de localidad.
- 3 Tamaño típicos de la TLB: 32 a 512 entradas.

Algoritmo de carga y reemplazamiento

1 Carga bajo demanda:

- ▶ La página virtual se carga en la memoria principal cuando es referenciada y produce un fallo de página.
- ▶ Método simple y no sobrecarga el canal de paginación.

2 Precarga:

- ▶ Ante un fallo de página, se carga en memoria principal la página fallada junto con otra/s adicional/es.
- ▶ Método predictivo: se intenta precargar páginas que son susceptibles de ser referenciadas en un futuro próximo.

1 Algoritmo de reemplazo: controlado por el S.O. (LFU, NUR...)

Inconvenientes

- Las páginas tienen un tamaño fijo y arbitrario: no tienen relación con la estructura lógica del programa.
- Puede haber datos en una página no relacionados con los demás. Ineficiencia por el principio de localidad.

Índice

- 1 Introducción
- 2 Sistemas paginados
- 3 Sistemas segmentados**
- 4 Sistemas segmentados con paginación
- 5 Ejercicios

Sistemas segmentados

- 1 Sistema de memoria virtual que considera bloques o **segmentos** de tamaño desigual.
- 2 Los bloques se definen en función de la estructura lógica del código y datos del programa (procedimientos, funciones, arrays, matrices).
- 3 La mayoría de la características (conceptos) de los sistemas segmentados son similares a las de los sistemas paginados.

Nota:

El algoritmo de reemplazamiento es más complejo

Sistemas segmentados

- 1 Sistema de memoria virtual que considera bloques o **segmentos** de tamaño desigual.
- 2 Los bloques se definen en función de la estructura lógica del código y datos del programa (procedimientos, funciones, arrays, matrices).
- 3 La mayoría de la características (conceptos) de los sistemas segmentados son similares a las de los sistemas paginados.

Nota:

El algoritmo de reemplazamiento es más complejo

Fallos de segmento

- 1 Es necesario determinar si hay espacio de memoria principal para ubicar el segmento virtual fallado (este problema no existe en los sistemas paginados).
 - ▶ Lista de segmentos reservados.
 - ▶ Lista de segmentos libres (LAVS), normalmente en orden ascendente de su posición.
- 2 Algoritmos de ubicación: first-fit, best-fit, worst-fit, binary-buddy.
- 3 Si no hay segmentos disponibles: **compactación** o **reemplazo**.

Índice

- 1 Introducción
- 2 Sistemas paginados
- 3 Sistemas segmentados
- 4 Sistemas segmentados con paginación**
- 5 Ejercicios

Sistemas segmentados con paginación

- 1 Hoy en día pocos sistemas utilizan segmentación pura debido a los problemas para efectuar los reemplazos.
- 2 Enfoque híbrido: segmentos paginados.
- 3 Simplifica el reemplazo: no se precisa que la memoria de un segmento sea contigua y no es preciso cargar los segmentos enteros en memoria.

Índice

- 1 Introducción
- 2 Sistemas paginados
- 3 Sistemas segmentados
- 4 Sistemas segmentados con paginación
- 5 Ejercicios**

Ejercicio 2

Enunciado:

Considera un computador con **memoria virtual paginada** con un esquema de traducción directa en un nivel, donde el **espacio virtual es de 128MB**, el **espacio físico es de 1MB** y el **tamaño de página es de 4KB**. ¿Qué tamaño en bytes tiene la tabla de páginas si esta contiene la información mínima para hacer la traducción (no hay ningún bit de control)?

- Halla el tamaño de las páginas si queremos que la tabla de páginas ocupe sólo 1KB.

Ejercicio 2

Solución (planteamiento):

- $|V| = 128\text{MB} = 2^{27}$ bytes \implies **27 bits dirección virtual.**
- $|M| = 1\text{MB} = 2^{20}$ bytes \implies **20 bits dirección física.**
- Páginas de 4KB $\implies 2^{12}$ bytes \implies **12 bits de desplazamiento.**

Ejercicio 2

Solución (a):



Tamaño TP = Número de páginas virtuales \times Tamaño entrada =

$$= \frac{2^{27} \text{ bytes de espacio virtual}}{2^{12} \text{ bytes/página}} \times \log_2\left(\frac{2^{20} \text{ bytes de espacio físico}}{2^{12} \text{ bytes/página}}\right) =$$

$$= 2^{15} \text{ entradas} \times 8 \text{ bits/entrada} = 2^{15} \text{ bytes} = \boxed{32 \text{ KB}}$$

Ejercicio 2

Solución (b):



$$\left(\frac{2^{27}}{2^p}\right) \text{ entradas} \times \log_2\left(\frac{2^{20}}{2^p}\right) \text{ bits/entrada} \times \left(\frac{1}{2^3}\right) \text{ bytes/bit} = 2^{10} \text{ B}$$

$$\left(\frac{2^{14}}{2^p}\right) \times \log_2\left(\frac{2^{20}}{2^p}\right) = 1$$

$$2^{(14-p)} \times (20 - p) = 1$$

$$p=16 \implies 2^p = 2^{16} = 64\text{KB}$$

Ejercicio 3

Enunciado:

Supongamos un computador con un **espacio virtual de 64 bytes**, una **memoria principal de 32 bytes** y un **tamaño de página de 8 bytes**. El computador utiliza un esquema de memoria virtual paginada con traducción en un nivel, con el auxilio de una **TLB de tres entradas**. El algoritmo de reemplazo de las páginas físicas y de las entradas de la TLB es de tipo FIFO.

El sistema también posee una **memoria caché de 8 bytes**, organizada por conjuntos. En concreto dispone de dos conjuntos con dos líneas por conjunto. Dentro de cada conjunto se utiliza un algoritmo de reemplazo de tipo LRU.

Si el procesador emite lista de direcciones virtuales mostrada, indicar y explicar cómo evoluciona la tabla de páginas, el TLB y el directorio caché partiendo de la situación inicial indicada en la figura siguiente:

Ejercicio 3

Enunciado:

Referencias: **1D**, **21**, **1C**, **0C**, **15** y **2A**.

Tabla
de paginas

R	Pag. Fis.
0	---
1	00
2	---
3	01
4	---
5	---
6	---
7	10

TLB

Pag. Virt.	Pag. Fis.
111	10
001	00
011	01

Cache

LRU	V	DC	ZA
1	1	000	
Conjunto 0	1	001	

LRU	V	DC	ZA
0	1	010	
Conjunto 1	1	101	

Ejercicio 3

Solución (planteamiento):

- $|V| = 64$ bytes \implies **6 bits de dirección virtual.**
- $|M| = 32$ bytes \implies **5 bits de dirección física.**
- Páginas de 8 bytes \implies **3 bits de desplazamiento.**
 - ▶ $6 - 3 = 3$ bits para indexar la tabla de páginas.
 - ▶ $5 - 3 = 2$ bits para el número de página física.
- Caché de 8B asociativa por conjuntos de 2 vías con 2 conjuntos \implies **líneas de 2B.**
 - ▶ 1 bit para seleccionar el byte, 1 bit para seleccionar el conjunto, 3 bits de etiqueta.

Ejercicio 3

Solución (Estado tras el primer acceso **1D**):

Tabla de paginas

R	Pag. Fis.
0	---
1	00
2	---
3	01
4	---
5	---
6	---
7	10

TLB

Pag. Virt.	Pag. Fis.
111	10
001	00
011	01

Cache

LRU	V	DC	ZA
0	1	000	
Conjunto 0	1	011	

LRU	V	DC	ZA
0	1	010	
Conjunto 1	1	101	

Ejercicio 3

Solución (Estado tras el segundo acceso **21**):

Tabla de paginas

R	Pag.	Fis.
0	0	---
1	1	00
2	0	---
3	1	01
4	1	11
5	0	---
6	0	---
7	1	10

TLB

Pag. Virt.	Pag. Fis.
100	11
001	00
011	01

Cache

LRU	V	DC	ZA
1	1	110	
1	1	011	

Conjunto 0

LRU	V	DC	ZA
0	1	010	
1	1	101	

Conjunto 1

Ejercicio 3

Solución (Estado tras el tercer acceso **1C**):

Tabla de paginas

R	Pag.	Fis.
0	0	---
1	1	00
2	0	---
3	1	01
4	1	11
5	0	---
6	0	---
7	1	10

TLB

Pag. Virt.	Pag. Fis.
100	11
001	00
011	01

Cache

LRU	V	DC	ZA
0	1	110	
1	1	011	

Conjunto 0

LRU	V	DC	ZA
0	1	010	
1	1	101	

Conjunto 1

Ejercicio 3

Solución (Estado tras el cuarto acceso **0C**):

Tabla de paginas

R	Pag.	Fis.
0	0	---
1	1	00
2	0	---
3	1	01
4	1	11
5	0	---
6	0	---
7	1	10

TLB

Pag. Virt.	Pag. Fis.
100	11
001	00
011	01

Cache

LRU	V	DC	ZA
1	1	001	
Conjunto 0	1	011	

LRU	V	DC	ZA
0	1	010	
Conjunto 1	1	101	

Ejercicio 3

Solución (Estado tras el quinto acceso **15**):

Tabla
de paginas

R	Pag.	Fis.
0	0	---
1	1	00
2	1	10
3	1	01
4	1	11
5	0	---
6	0	---
7	0	---

TLB

Pag. Virt.	Pag. Fis.
100	11
010	10
011	01

→

Cache

LRU	V	DC	ZA
0	1	001	
1	1	101	

Conjunto 0

LRU	V	DC	ZA
1	1	010	
0	0	101	

Conjunto 1

Ejercicio 3

Solución (Estado tras el sexto acceso **2A**):

Tabla
de paginas

R	Pag.	Fis.
0	0	---
1	0	---
2	1	10
3	1	01
4	1	11
5	1	00
6	0	---
7	0	---

TLB

Pag. Virt.	Pag. Fis.
100	11
010	10
101	00

Cache

LRU	V	DC	ZA
0	0	001	
1	1	101	

Conjunto 0

LRU	V	DC	ZA
0	1	010	
1	1	000	

Conjunto 1

Ejercicio 4

Enunciado:

Un computador tiene un **esquema de memoria virtual paginada en dos niveles** direccionable a nivel de byte. Tanto la tabla de páginas del primer nivel como las del segundo tienen la misma estructura: caben exactamente en una página y constan de 128 entradas de 16 bits, siendo el bit más significativo el de residencia y los 8 menos significativos el número de página física. Además, la tabla de páginas del primer nivel se encuentra siempre residente en la última página física.

El computador tiene una **caché de 4KB** indexada por direcciones físicas y líneas de 4 bytes en la que cada entrada del directorio caché consta de un bit de validez más la etiqueta de la línea correspondiente. El tamaño total del directorio caché es de 1KB.

Sabiendo que el byte de la página física X con desplazamiento Y contiene el valor **$(X+Y) \bmod 256$** ; indica:

Ejercicio 4

Enunciado:

- a) Tamaño del espacio virtual y del espacio físico.
- b) Dirección física correspondiente a la dirección virtual **415A**.
- c) Indica el tamaño de los campos en que se divide la dirección física desde el punto de vista de la caché y cuántas líneas tiene cada conjunto de la misma.
- d) Si esta caché tuviera una tasa de fallos del 10%, un tiempo de acierto de 2ns y su tiempo medio de acceso fuera de 5ns, ¿Cuál sería su **penalización de fallo**?

Ejercicio 4

Solución (planteamiento):

- 128 entradas $\implies \log_2(128) = \mathbf{7 \text{ bits}}$ para indexar la entrada.
- 128 entradas $\times 2\text{B} \implies \mathbf{256\text{B}}$ por página.
- 256B por página $\implies \log_2(256) = \mathbf{8 \text{ bits}}$ para el desplazamiento.
- Dirección virtual:

Número PV		Δ
TP1	TP2	
7 bits	7 bits	8 bits

- Dirección física:

PF	Δ
8 bits	8 bits

Ejercicio 4

Solución (a):

- $|V| = 2^{22}$ bytes = **4MB espacio virtual.**
- $|F| = 2^{16}$ bytes = **64KB espacio físico.**

Ejercicio 4

Solución (b):

- Descomposición de la dirección **415A**:

TP1	TP2	Δ
$(0000000)_2 = 0_{16}$	$(1000001)_2 = 41_{16}$	$(01011010)_2 = 5A_{16}$

- La última página física es la 255 (FF), entonces la dirección $FF|00$ contiene según el enunciado:

Dirección	FF00	FF01
Contenido	FF	00

Ejercicio 4

Solución (b):

- En el segundo nivel el índice $(1000001)_2 = 41_{16}$ nos lleva, con entradas de 2B, a 00|82:

Dirección	0082	0083
Contenido	82	83

- Número de página física $(83_{16}) + \text{desplazamiento } (5A_{16}) \implies \boxed{835A}$

Ejercicio 4

Solución (c):

- $$\text{Líneas caché} = \frac{4096 \text{ bytes}}{4 \text{ bytes/línea}} = 1024$$

- Descomposición de la dirección en la caché:

Etiqueta	Índice	Δ
7 bits	7 bits	2 bits

- $$\text{Líneas por conjunto} = \frac{1024 \text{ líneas}}{128 \text{ conjuntos}} = \boxed{8 \text{ líneas}}$$

Ejercicio 4

Solución (d):



TMA = Tiempo de acierto + Tasa de Fallos \times Penalización de fallo



$$5ns = 2ns + 10/100 \times PF$$



$$PF = \frac{(5 - 2)ns}{0,1} = \boxed{30 \text{ ns}}$$

Ejercicio 5

Enunciado:

*Un computador con una **memoria física de 4 MB** tiene un espacio virtual segmentado con un esquema de traducción directa de un nivel que es direccionable a nivel de byte. El sistema permite ubicar los segmentos a partir de cualquier posición arbitraria en memoria física y no hay un tamaño mínimo para los segmentos. La tabla de segmentos ocupa exactamente 6 KB. Cada entrada tiene una longitud de 6 bytes. Los bits de control y protección (no incluyen la longitud del segmento) son los 6 bits más significativos de cada entrada, siendo en concreto el bit de residencia el más significativo. Tras ellos sigue la longitud del segmento. Finalmente, la dirección del segmento en memoria física se encuentra en los bits menos significativos de cada entrada. El registro base de la tabla de segmentos (RBTS) es 0 y la posición de memoria X contiene el valor **$(X + 2) \bmod 256$** .*

Ejercicio 5

Enunciado:

- a) Determina el tamaño del espacio virtual.
- b) Calcula la dirección física asociada a la dirección virtual **0x04081516**.

Ejercicio 5

Solución (a):

- $6KB/6B = 1024$ **entradas en la tabla de segmentos**
 - ▶ 2^{10} segmentos \implies **10 bits** para identificar el segmento
- $4MB \implies$ **22 bits** dirección física
- Entrada en la tabla de segmentos ($6B = 48bits$):

Control	Longitud	Dir. Base
6 bits	20 bits	22 bits

- 10 bits de número de segmento virtual + 20 bits de desplazamiento dentro del segmento \implies 2^{30} bytes = 1 GB

Ejercicio 5

Solución (b):

- Descomposición de la dirección virtual:

Núm SV	Δ
040	81516

- Dirección de la entrada para el segmento:

$$0 + 40 \times 6 = 180$$

- Contenido de la memoria a partir de 0x180:

Dirección	180	181	182	183	184	185
Contenido	82	83	84	85	86	87

Ejercicio 5

Solución (b):

- Descomposición del contenido de la entrada:

Longitud	Dir. Base
A0E12	058687

- Dirección buscada:

$$058687 + 81516 = \boxed{0D9B9D}$$