

Organización y gestión de la memoria

Diego Feroldi
Arquitectura del Computador

26/10/2023

Introducción

- ▶ Las primeras computadoras contaban con muy poca memoria física.
- ▶ PDP-1 con una memoria principal de hasta 144 KB para almacenar el sistema operativo y los programas de usuario.
- ▶ En 1961 se propuso un método para realizar automáticamente la gestión entre la memoria principal y secundaria: **memoria virtual**.
- ▶ La **memoria principal** es una memoria utilizada para almacenar programas mientras se ejecutan. Ejemplo: DRAM.
- ▶ La **memoria secundaria** es una memoria no volátil. Ejemplos: discos de estado sólido, cintas magnéticas, CDs, DVDs, memorias USB, etc.

Memoria virtual

- ▶ Es una abstracción de la memoria principal.
- ▶ La memoria virtual es un mecanismo eficiente que proporciona a cada proceso un espacio de direcciones grande, uniforme y privado.
- ▶ La memoria virtual proporciona tres capacidades importantes:
 1. Utiliza la memoria principal de manera eficiente manteniendo solo las áreas activas en la memoria principal y transfiriendo datos entre el disco y la memoria según sea necesario.
 2. Simplifica la gestión de la memoria al proporcionar a cada proceso un espacio de direcciones uniforme.
 3. Protege el espacio de direcciones.

Memoria virtual

- ▶ La memoria virtual se puede definir como una técnica que utiliza la memoria principal como un “*buffer*” de almacenamiento para el almacenamiento secundario.
- ▶ La idea consiste en separar los conceptos de **espacio de direcciones** y **posiciones de memoria**.
- ▶ Ejemplo: Consideremos una computadora de los años 60 con una memoria de 4 KB y palabras de 16 bits.
- ▶ El número de palabras direccionables virtualmente depende únicamente del número de bits que tiene una dirección y es **independiente** de la cantidad de memoria física que tiene la máquina.

Memoria virtual

- ▶ La motivación principal para utilizar memoria virtual es permitir el uso compartido eficiente y seguro de la memoria entre varios programas.
- ▶ La segunda motivación es permitir que un solo programa de usuario exceda el tamaño de la memoria primaria.

Memoria virtual

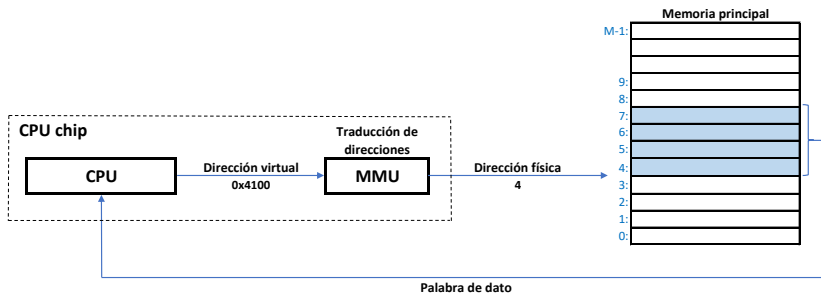


Figura: Esquema básico de memoria virtual.

Memoria virtual

- ▶ Una **dirección lógica** es una dirección de memoria tal como la ve una aplicación.
- ▶ En la instrucción `movq 0x404028, %rax` hay una dirección lógica: `0x404028`.
- ▶ ¡Un programador tiene la ilusión de que él es el único usuario de la memoria!
- ▶ En este contexto, **dirección virtual** es sinónimo de **dirección lógica**.

Segmentación

- ▶ Un método para implementar memoria virtual es el denominado **segmentación**.
- ▶ Se basa en proporcionar muchos espacios de direcciones totalmente independientes, llamados **segmentos**.
- ▶ Los segmentos son una forma de organizar y administrar el espacio de direcciones de la memoria virtual.
- ▶ Representan divisiones lógicas dentro de ese espacio de direcciones y cada segmento puede tener sus propios permisos, tamaño y características.

Segmentación

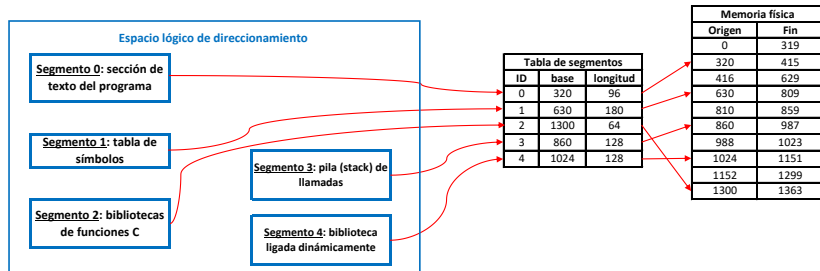
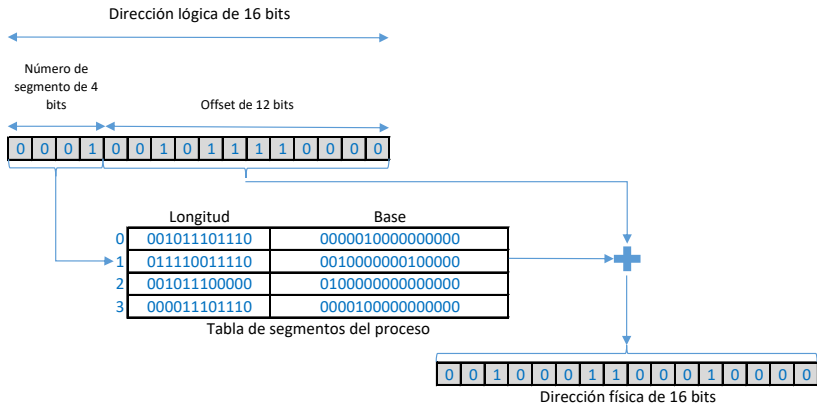


Figura: Esquema de segmentación (ID es el identificador del segmento).

Ejemplo con segmentación

Consideremos una dirección de $n + m$ bits, donde los n bits más a la izquierda son el número de segmento y los m bits más a la derecha son el desplazamiento.

En nuestro ejemplo, $n = 4$ y $m = 12$. Por lo tanto, el tamaño máximo del segmento es $2^{12} = 4096$.



Paginación

- La **dirección virtual** se convierte mediante una combinación de hardware y software en una dirección física.

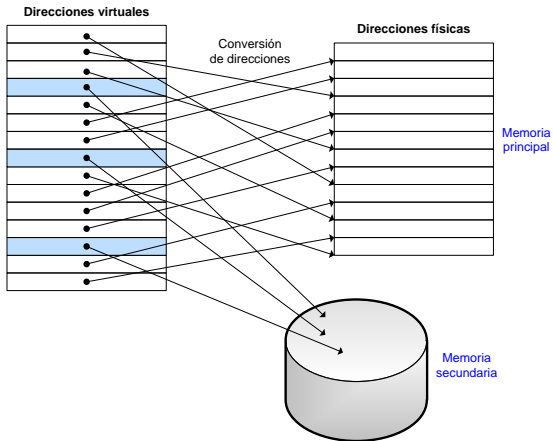


Figura: Conversión de direcciones.

Tabla de paginación

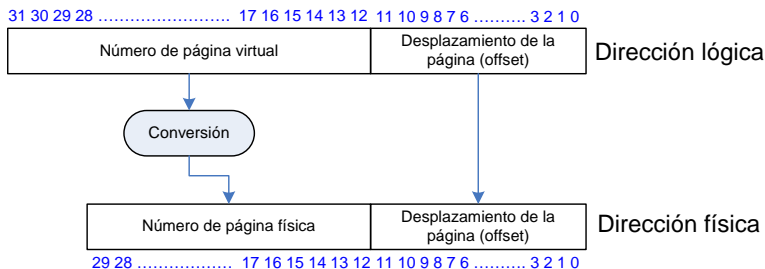
- ▶ El procesador usa una **tabla de páginas** para traducir direcciones virtuales a direcciones físicas.
- ▶ La tabla de páginas (o tabla de paginación) contiene una entrada para cada **página virtual**.

Número de página virtual	V	Dirección de página física
0x7fff	0	
0x7ffe	0	
0x7ffd	0	
0x7ffc	0	
0x7ffb	1	0x0001
0x7ffa	0	
⋮		⋮
0x0004	0	
0x0003	1	0x7ffe
0x0002	0	
0x0001	0	
0x0000	1	0x0000

Figura: Tabla de páginas.

Paginación (Ejemplo)

- ▶ Supongamos direcciones virtuales de 32 bits y páginas de 4 KB = 4096 bytes = 2^{12} bytes.
- ▶ Además, supongamos que tenemos una memoria física de 1 GB = 2^{30} bytes.



Paginación (Ejemplo)

	V Número de página físico	
0xffff0	0	0x3aa00
0xffff1	1	0x31800
0xffff2	0	0x320a0
0xffff3	0	0x3021f
0xffff4	0	0x31c09
0xffff5	1	0x36020
0xffff6	0	0x32056
0xffff7	1	0x3ff00
0xffff8	1	0x3ff0c
0xffff9	0	0x3fe00

Tamaño de página

- ▶ El tamaño de la página es a menudo un parámetro que puede elegir el sistema operativo entre varios tamaños soportables por la arquitectura.
- ▶ Determinar el mejor tamaño de página requiere equilibrar varios factores en competencia.
- ▶ **fragmentación interna:** Con n procesos y un tamaño de página de p bytes, se desperdiciarán $(n \times p)/2$ bytes debido a la fragmentación interna.
- ▶ Por otro lado, páginas pequeñas significan que los programas necesitarán muchas páginas, por lo tanto, una tabla de páginas grande.

Translation Lookaside Buffer (TLB)

- ▶ El procesador puede mantener las últimas entradas de la tabla de páginas en la memoria caché.
- ▶ El procesador primero trata encontrar la traducción en la TLB.
- ▶ Una TLB se organiza como una memoria asociativa y, por lo general, contiene de 16 a 512 entradas.
- ▶ Cada entrada de TLB contiene un número de página virtual, su correspondiente número de página física y banderas.
- ▶ Si la TLB “acierta”, devuelve el número de página física correspondiente.
- ▶ Las TLB suelen tener una tasa de aciertos superior al 99 %.

TLB: Ejemplo

Supongamos ahora que además de la tabla de páginas tenemos una TLB como se muestra a continuación:

V	Número de página virtual	Número de página físico
0	0xefff0	0x0aa00
1	0xffff5	0x36020
1	0x0fff4	0x4800a
0	0xaaff3	0x7a001

- ▶ Dirección virtual: 0xffff5a06 → Número de página virtual: es 0xffff5 → Dirección física: 0xffff5a06.
- ▶ Dirección virtual: 0xffff8407 → Número de página virtual: 0xffff8 pero no se encuentra en la TLB → hay que buscar en la tabla de páginas.

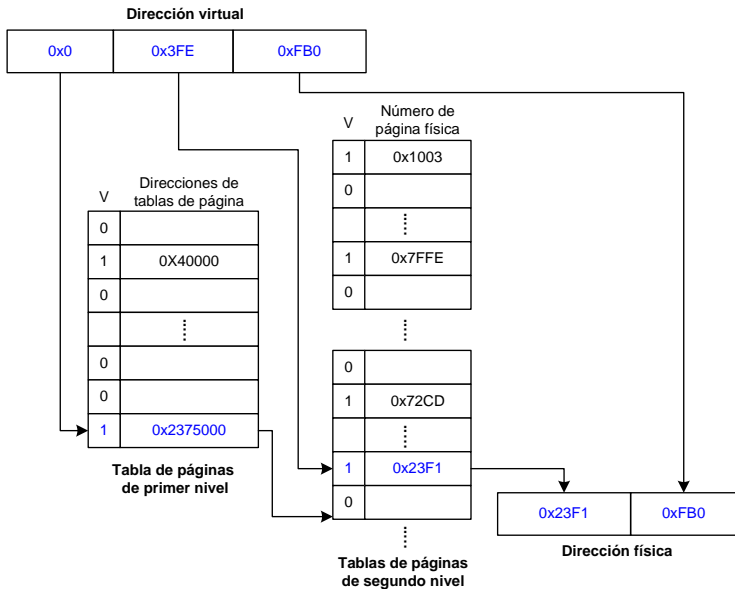
Tablas de páginas multinivel

- ▶ Las tablas de páginas pueden ocupar una gran cantidad de memoria física.
- ▶ Por ejemplo, la tabla de páginas para una memoria virtual de 32 bits con páginas de 4 KB necesitaría $\frac{2^{32} \text{ bytes}}{2^{12} \text{ bytes}} = 2^{20}$ entradas.
- ▶ Si cada entrada tiene 4 bytes, la tabla de páginas ocupa $2^{20} \times 2^2 \text{ bytes} = 2^{22} \text{ bytes} = 4 \text{ MB}$.
- ▶ Las tablas de páginas se pueden dividir en múltiples niveles.
- ▶ Las estructuras de tablas de páginas más complejas comprenden cuatro niveles.

Paginación multinivel

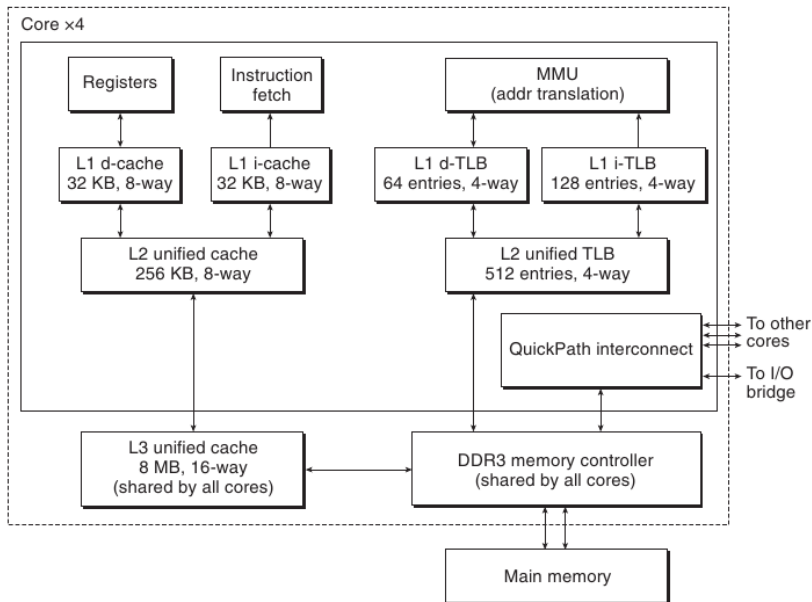
- ▶ El número de página virtual de 19 bits se divide en 9 y 10 bits, para indicar el número de la tabla de páginas y el desplazamiento de la tabla de páginas.
- ▶ La tabla de páginas de primer nivel tiene $2^9=512$ entradas.
- ▶ Cada una de estas 512 entradas apunta a una tabla de segundo nivel.
- ▶ A su vez, cada una de las tablas de segundo nivel tiene $2^{10} = 1024$ entradas.
- ▶ Si cada una de las entradas de la tabla de páginas de primer y segundo nivel es de 4 bytes y solo dos tablas de páginas de segundo nivel están presentes en la memoria física a la vez $\rightarrow (512 \times 4 \text{ bytes}) + 2 \times (1024 \times 4 \text{ bytes}) = 10 \text{ KB}$ de memoria física (mucho menor a los 2 MB).
- ▶ Sin embargo, agrega otro acceso a la memoria para la traducción cuando “falla” la TLB.

Paginación multinivel: ejemplo



Caso de estudio: Intel Core i7

Processor package



Caso de estudio: Intel Core i7

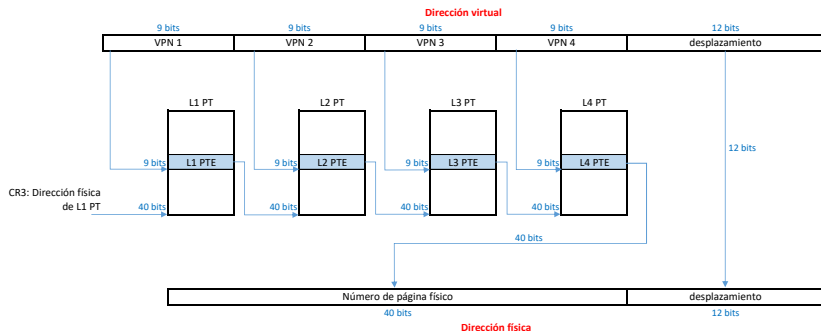


Figura: Esquema de traducción de direcciones virtuales del Intel Core i7 con tabla de páginas multinivel.