

# TEMA 8 - GESTIÓN DE MEMORIA

**Sistemas informáticos**  
Alba Alejandre Voces

# CONCEPTOS BÁSICOS

**MEMORIA:** Es el dispositivo de almacenamiento de programas en ejecución junto con sus datos asociados.

**POSICIÓN DE MEMORIA:** Cada una de las celdas que forman la memoria y donde se almacena un dato. Generalmente este dato ocupa 1 byte, pero pueden ser una cantidad mayor.

**DIRECCIÓN DE MEMORIA:** Número que indica dónde se encuentra la posición de memoria y que sirve para acceder al dato. Se numeran desde 0 y en orden creciente.

# CONCEPTOS BÁSICOS

**DIRECCIONABLE POR BYTE:** Cada posición de memoria dispone de un byte para almacenar información y cada byte tiene una dirección de memoria para acceder al dato que contiene.

**REFERENCIA DE MEMORIA:** Posición a la que quiere acceder la CPU cuando busca un dato en memoria.

**INTERACCIÓN:** Secuencia de referencias a direcciones específicas de memoria. La CPU lee información (lectura) de la memoria o la almacena en ella (escritura).

# ADMINISTRADOR DE MEMORIA

LLEVAR CONTROL MEMORIA LIBRE

ASIGNAR MEMORIA A PROCESOS LIBRES

CONTROL DE MEMORIA USADA

LIBERAR MEMORIA

# ESPACIO DE DIRECCIONES

**Protección:** sólo acceso a zonas de memoria del propio proceso

**Reubicación:** zonas de memoria diferentes en diferentes ejecuciones

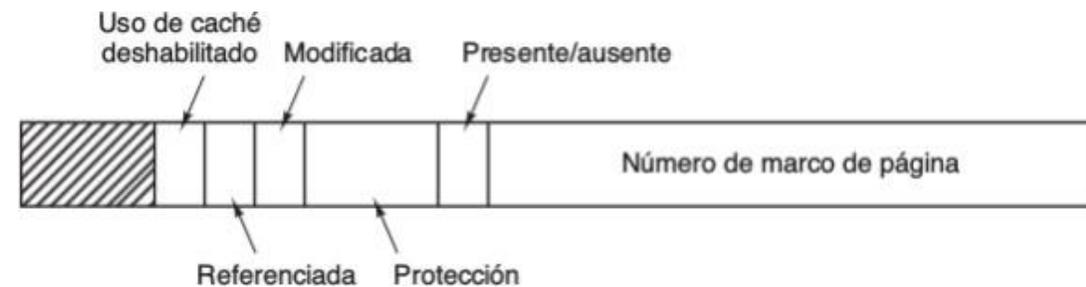
**Espacio de direcciones lógico:** direcciones que referencian un proceso

**Espacio de direcciones físico:** conjunto de direcciones físicas en memoria

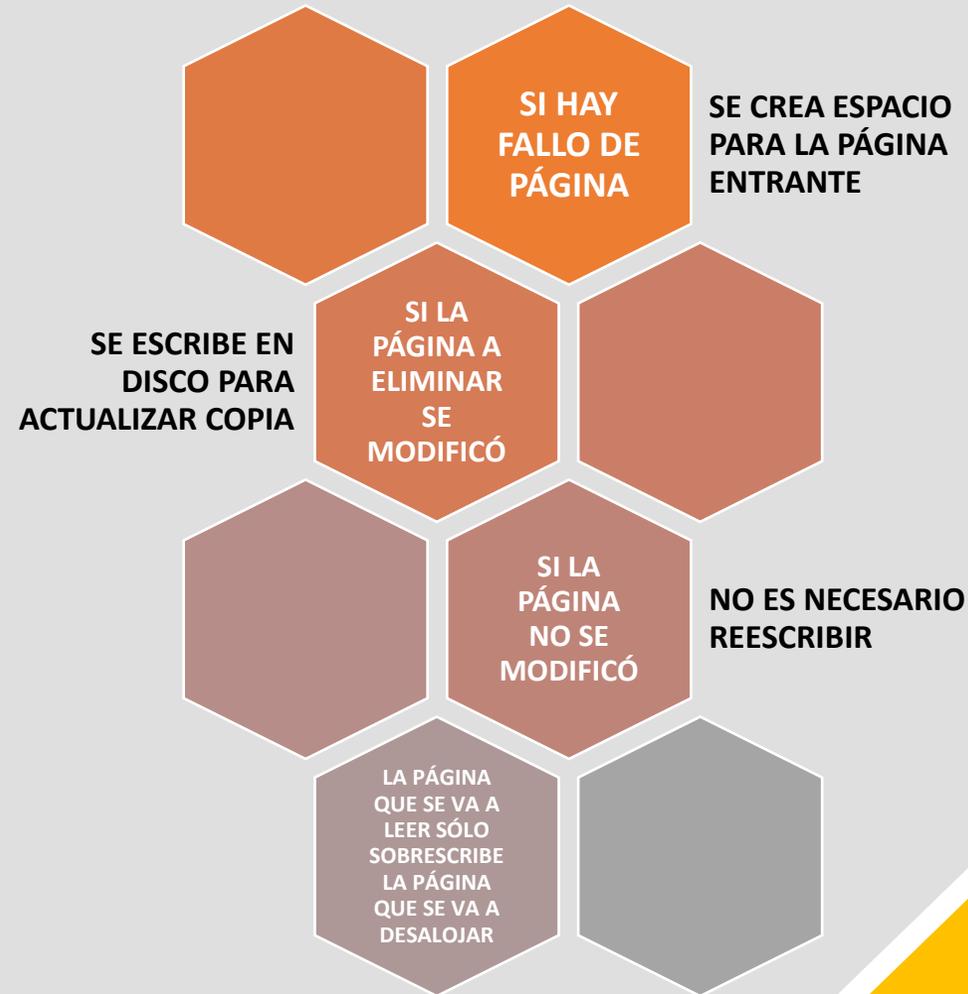
# MEMORIA VIRTUAL PAGINACIÓN

Surge por la demanda de más memoria

- Los procesos hacen referencia a un conjunto de **direcciones virtuales** divididas en **páginas**
- La CPU referencia una dirección virtual que la **MMU** (Memoria Management Unit o Unidad de Administración de Memoria) traduce a la dirección física



# Algoritmo reemplazo de página



# Algoritmo reemplazo de página óptimo

LA PÁGINA CON LA  
ETIQUETA MÁS ALTA  
DEBE ELIMINARSE

PROBLEMA DEL  
ALGORITMO: NO SE  
PUEDE REALIZAR

# NRU

La mayor parte de los sistemas con memoria virtual tienen dos bits de estado asociados a cada página

- R se establece a 1 cada vez que se hace referencia a la página (lectura o escritura)
- M se establece a 1 cuando se escribe en la página, es decir, se modifica

# NRU

Cuando ocurre un fallo de página, el sistema operativo inspecciona todas las páginas y las divide en 4 categorías con base en los valores actuales de sus bits R y M:

- Clase 0: no ha sido referenciada, no ha sido modificada.
- Clase 1: no ha sido referenciada, ha sido modificada
- Clase 2: ha sido referenciada, no ha sido modificada
- Clase 3: ha sido referenciada, ha sido modificada

El algoritmo NRU (*Not Recently Used*) elimina una página al azar de la clase de menor numeración que no esté vacía

# FIFO

El Sistema Operativo mantiene una lista con todas las páginas que están actualmente en memoria ordenadas por orden de llegada

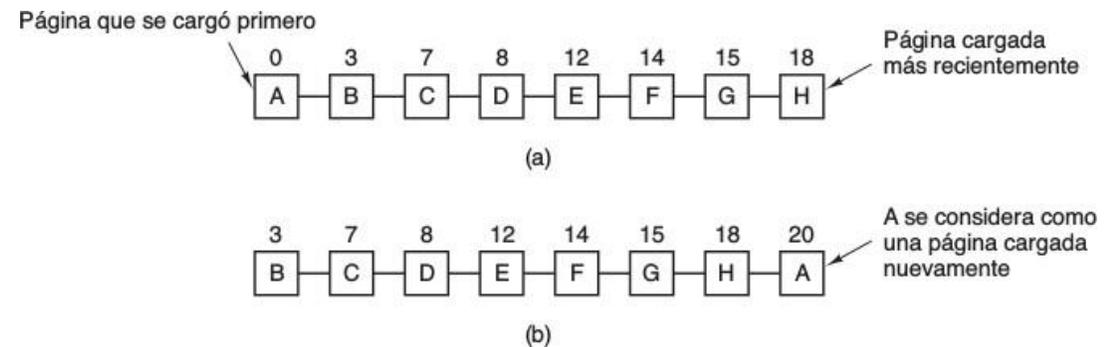
Si hay fallo de página se elimina la página que lleva más tiempo en la lista y la nueva se agrega al principio

**PROBLEMA:** puede reemplazar una página muy referenciada

# Segunda oportunidad

Modificación algoritmo FIFO en la que se soluciona el problema de la página muy referenciada inspeccionando el bit R

- Si es 0, la página es antigua y no se ha utilizado, se sustituye de inmediato
- Si es 1, el bit se borra, la página se pone al final de la lista de páginas y se carga su tiempo



# LRU

En el algoritmo LRU (Least Recently Used) cuando ocurre un fallo de página se descarta la que no se haya utilizado durante la mayor longitud de tiempo

- PROBLEMA: muy costoso
- SOLUCIÓN: Contador 64 bits (C) que se incrementa después de cada instrucción. Tras cada referencia a memoria, el valor de C se almacena en la entrada de la tabla de páginas para la página que se acaba de referenciar. Cuando hay un fallo, el SO examina los contadores para encontrar el menor, siendo la página de uso menos reciente

# Conjunto de trabajo

Es el conjunto de páginas que utiliza un proceso en un momento dado

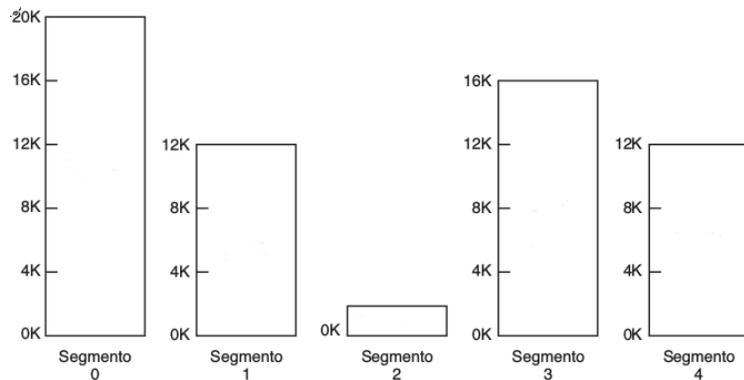
- Los procesos inician sin ninguna página en memoria
- La primera instrucción en CPU genera un fallo de página
- SO tiene que traer la página que contiene primera instrucción
- Al repetirse suficientes veces el proceso tiene la mayoría de páginas que necesita

Este proceso se denomina **Paginación bajo demanda**

# PÁGINAS COMPARTIDAS

- Es más eficiente compartir las páginas para evitar tener 2 copias de la misma en memoria al mismo tiempo
- No todas las páginas se pueden compartir, sólo las de **lectura**
- Cuando dos o más procesos comparten código puede ocurrir un problema con las páginas compartidas si el planificador decide borrar todas las páginas de un proceso

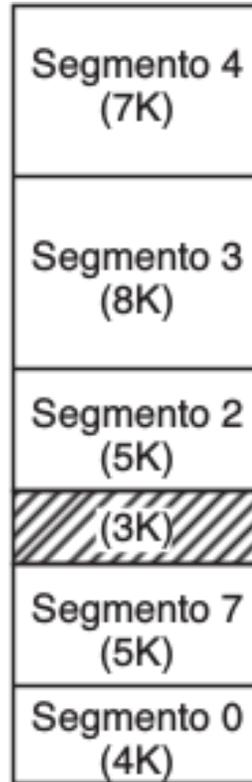
# SEGMENTACIÓN



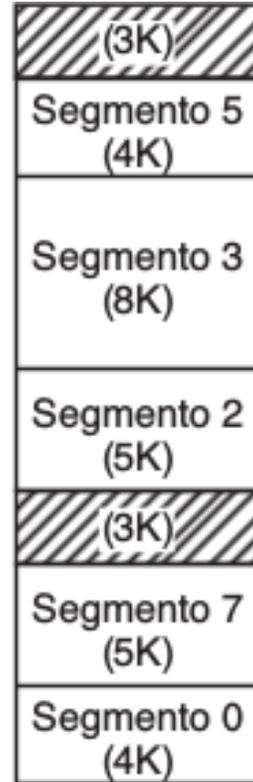
- Sistema de memoria que permite disponer de espacio de direcciones dinámicos
- Programa dividido en **segmentos** de tamaño diferente que pueden crecer y decrecer durante la ejecución del programa
- Cada **segmento** es secuencia de direcciones independientes de otros segmentos
- Dirección de memoria segmentada número de segmento y dirección dentro del segmento



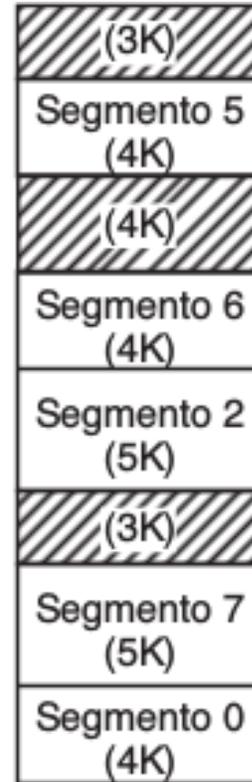
(a)



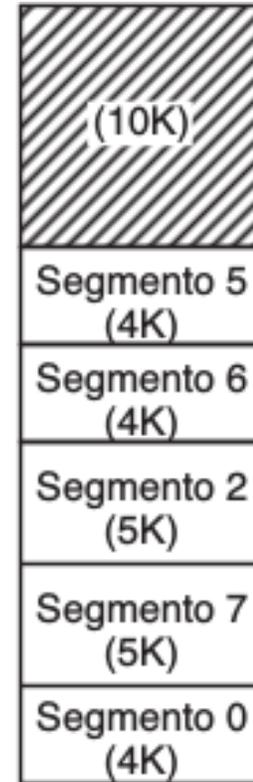
(b)



(c)



(d)



(e)

# Entrada de la tabla de segmentos

Bit de residencia de página	Dirección en almacenamiento secundario (si la página no está en almacenamiento principal)	Bit de protección				Longitud del segmento	Dirección base del segmento (si está en almacenamiento real)
<i>r</i>	<i>a</i>	<i>R</i>	<i>W</i>	<i>E</i>	<i>A</i>	<i>l</i>	<i>s'</i>

*r* = 0 si la página no está en almacenamiento real  
*r* = 1 si la página está en almacenamiento real

Bits de protección: (1 - si, 0 - no)

R - acceso para lectura  
W - acceso para escritura  
E - acceso para ejecución  
A - acceso para adición

# Traducción de direcciones virtuales con segmentación

- Un proceso en ejecución hace referencia a una dirección de memoria virtual:  $v=(s,d)$
- El SO busca  $s$  en tabla de segmentos y examina  $r$  para ver si está en memoria
- Comprueba el desplazamiento  $d$  si es menor o igual que la longitud del segmento  $l$
- Revisa bits de protección para asegurar si está permitida la operación
- El SO determina el segmento  $s$  a partir de la dirección  $S$ . Dirección real  $r = S+d$

# Estrategias colocación segmentos en la memoria

PRIMER  
AJUSTE

SIGUIENTE  
AJUSTE

MEJOR  
AJUSTE

PEOR  
AJUSTE

# SEGMENTACIÓN PAGINADA

Sólo las páginas que se necesitan son cargadas en memoria

Segmentos con tamaño múltiplo de páginas

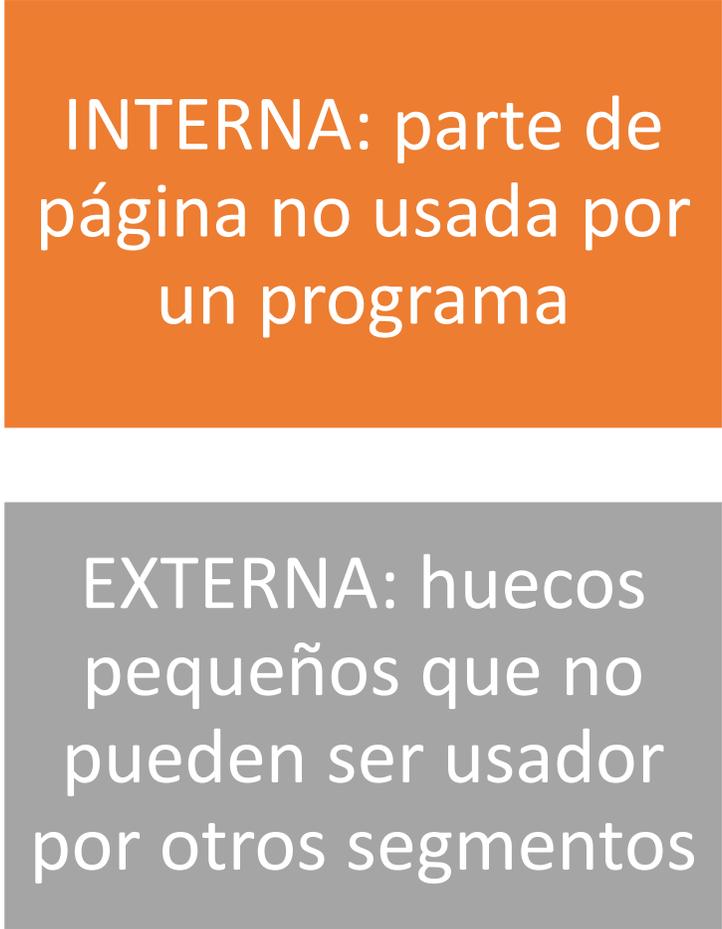
No todas las páginas de un segmentos tienen que estar en memoria a la vez

Las páginas contiguas en memoria virtual no son contiguas en memoria física

# *Traducción de direcciones virtuales en segmentación paginada*

Número de segmento $s$	Número de página $p$	Desplazamiento $d$	Dirección virtual $v = (s,p,d)$
------------------------	----------------------	--------------------	---------------------------------

# FRAGMENTACIÓN



The diagram illustrates memory fragmentation. On the left, the word 'FRAGMENTACIÓN' is written in large black letters. A vertical line extends from this word to the right, where two rectangular boxes are stacked vertically. The top box is orange and contains the text 'INTERNA: parte de página no usada por un programa'. The bottom box is gray and contains the text 'EXTERNA: huecos pequeños que no pueden ser usados por otros segmentos'. The entire content is framed by a thick gray border, with a yellow triangular shape in the bottom right corner.

INTERNA: parte de página no usada por un programa

EXTERNA: huecos pequeños que no pueden ser usados por otros segmentos