

Fundamentos de los Sistemas Operativos (FSO)

Departamento de Informàtica de Sistemes y Computadoras (DISCA)

Universitat Politècnica de València

Bloque Temático 4: Gestión de Memoria

Seminario de la Unidad Temática 11

SUT11:

Problemas de Memoria Virtual

fSO

DISCA

- Ejercicio 1: Paginación sin MV
- Ejercicio 2: Paginación con MV
- Ejercicio 3: Algoritmos de reemplazo
- Ejercicio 4: Ámbito del reemplazo

Ejercicio 1: Paginación sin MV

Un procesador maneja direcciones lógicas de 16 bits y soporta paginación. Se fabrica en tres versiones, con tamaños de página de 256, 1024 y 4096 bytes. En cada modelo, el Sistema Operativo ha de gestionar la paginación según sus características.

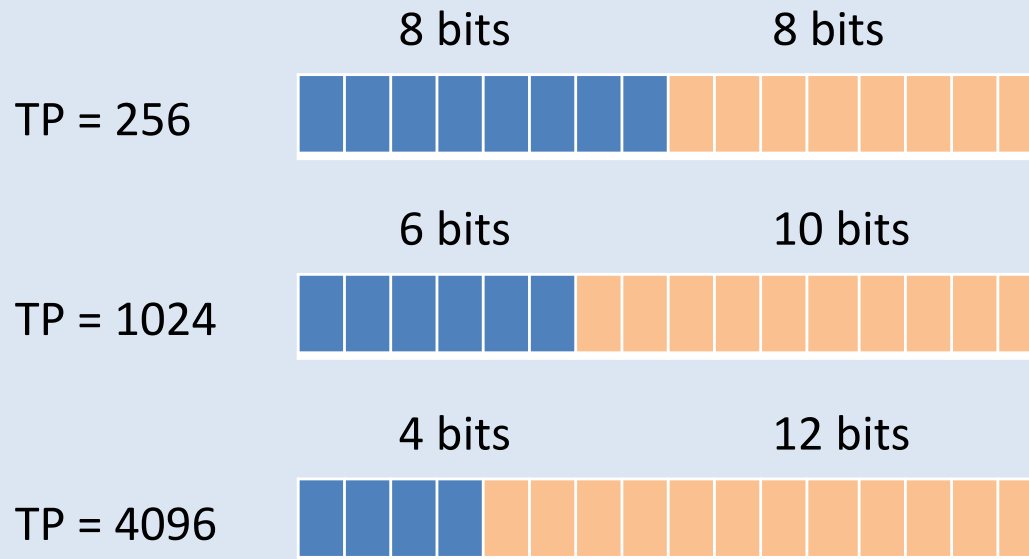
Un ejecutable para este procesador contiene **2800** bytes de instrucciones a partir de la dirección 0x1000, **1198** bytes de datos a partir de 0x3000 y prevé **2048** bytes iniciales para la pila a partir de 0x9000.

Calcule el número de entradas de la tabla de páginas y el número de páginas iniciales que ocupará el proceso para cada modelo de procesador

1. 256 bytes
2. 1024 bytes
3. 4096 bytes

Tamaño de página (bytes)	Número de entradas de la tabla	Número de páginas ocupadas por el proceso
256		
1024		
4096		

Ejercicio 1: Paginación sin MV



región	tamaño (bytes)	dirección base
código	2800	0x1000
variables	1198	0x3000
pila	2048	0x9000

Tamaño de página (bytes)	Número de entradas de la tabla	Número de páginas ocupadas por el proceso
256	$2^8 = 256$	$2800 \text{ div } 256 + 1198 \text{ div } 256 + 2048 \text{ div } 256 = \mathbf{24}$
1024	$2^6 = 64$	$2800 \text{ div } 1024 + 1198 \text{ div } 1024 + 2048 \text{ div } 1024 = \mathbf{7}$
4096	$2^4 = 16$	$2800 \text{ div } 4096 + 1198 \text{ div } 4096 + 2048 \text{ div } 4096 = \mathbf{3}$

Ejercicio 1. Apartado 2

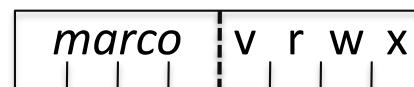
región	tamaño (bytes)	base (hex)
código	2800	1000
variables	1198	3000
pila	2048	9000

Construya *en binario* la tabla de páginas inicial del proceso cuando ejecuta el programa en el caso de tamaño de página de 4096 bytes.

Considere que la memoria física es de 64 KB y que en el momento de la carga del proceso sólo están ocupados los dos marcos superiores. El SO asigna marcos por orden creciente de dirección física.

Primero ubica el código, luego las variables y, finalmente, la pila.

Los descriptores de páginas tienen la siguiente estructura:



Si una página no existe en el mapa, se ha de indicar con $r=w=x=0$

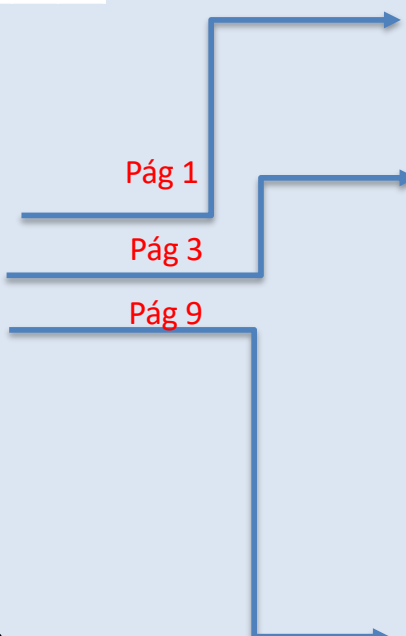
0		
1		
2		
3		
4		
5		
6		
7		
8		
9		
A		
B		
C		
D		
E		
F		

Ejercicio 1. Apartado 2

TP = 4096

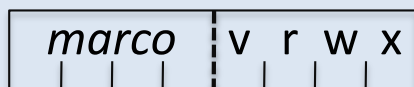


región	tamaño (bytes)	dirección base
código	2800	0x1000
variables	1198	0x3000
pila	2048	0x9000

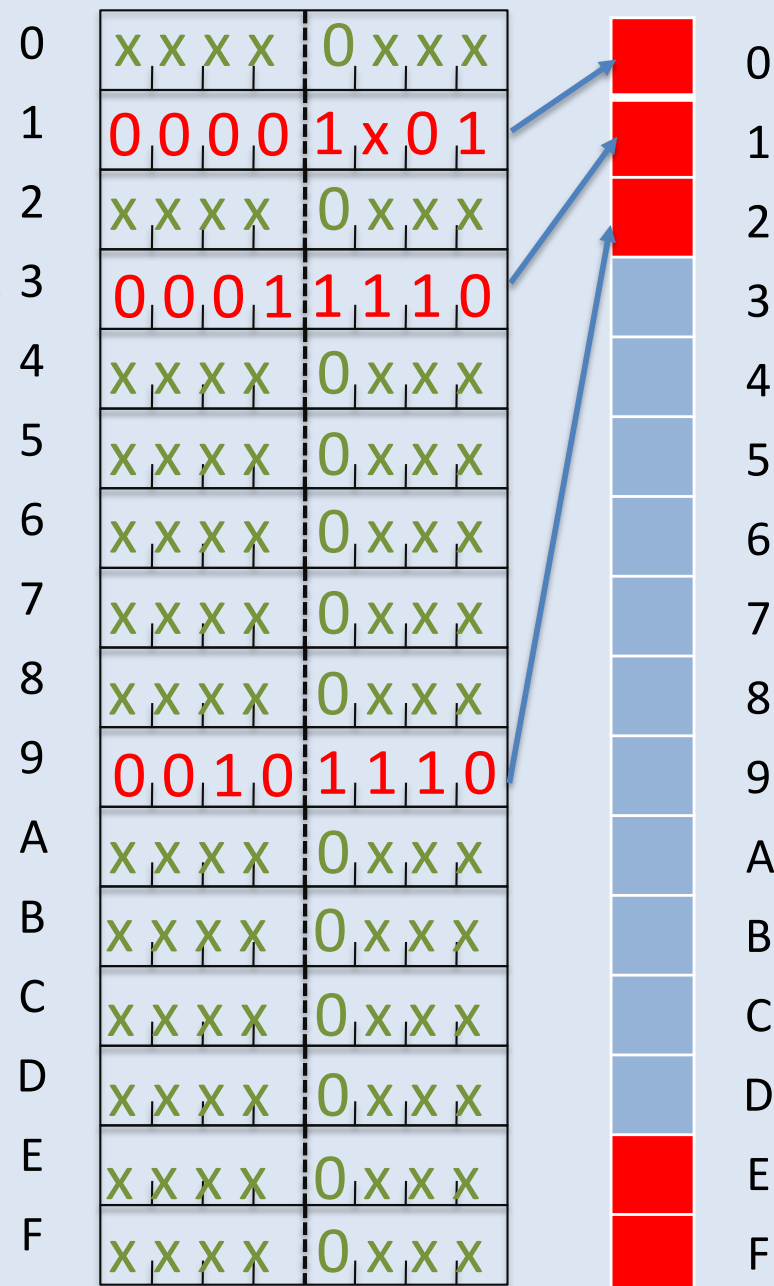


La memoria física es de 64 KB
 Los dos marcos superiores
 están ocupados al principio
 El SO asigna marcos por orden
 creciente de dirección física

Primero ubica el código,
 luego las variables y,
 finalmente, la pila



Si una página no existe en el mapa,
 se ha de indicar con r=w=x=0



Ejercicio 1. Apartado 3

Calcule, si es posible, las direcciones físicas que genera la MMU para cada uno de los accesos siguientes. Si no es posible la traducción, indique la razón

región	tamaño (bytes)	base
código	2800	0x1000
variables	1198	0x3000
pila	2048	0x9000

Tipo de acceso	Dirección lógica	Dirección física	Acceso legal o ilegal?
Lectura de instrucción	0x1000		
Lectura de instrucción	0x1004		
Lectura de instrucción	0x2000		
Lectura de instrucción	0x3000		
Lectura de variable	0x3010		
Lectura de variable	0x9010		
Escritura de variable	0x1000		
Escritura de variable	0x5000		

Ejercicio 1. Apartado 3

Calcule, si es posible, las direcciones físicas que genera la MMU para cada uno de los accesos siguientes. Si no es posible la traducción, indique la razón

región	tamaño (bytes)	base
código	2800	0x1000
variables	1198	0x3000
pila	2048	0x9000

Tipo de acceso	Dirección lógica	Dirección física	Acceso legal o ilegal?
Lectura de instrucción	0x1000	0x0000	Legal
Lectura de instrucción	0x1004	0x0004	Legal
Lectura de instrucción	0x2000		No existe página 2
Lectura de instrucción	0x3000		Página de variables. No ejecución
Lectura de variable	0x3010	0x1010	Legal
Lectura de variable	0x9010	0x2010	Legal
Escritura de variable	0x1000		Ilegal. Sin permiso de escritura
Escritura de variable	0x5000		No existe página 5

Ejercicio 2: Paginación con MV

fSO

Considere el procesador del ejercicio 1, con páginas de 4 KB, y un sistema operativo **con memoria virtual**. El proceso comienza sin ningún marco asignado y el sistema operativo asigna marcos libres por este orden: marco 0, marco 1, marco 2, etc.

región	tamaño (bytes)	base
código	2800	0x1000
variables	1198	0x3000
pila	2048	0x9000

Complete la tabla suponiendo que los accesos se producen POR EL ORDEN en que están escritos.

Acceso	Dirección lógica	Dirección física	Fallo de página?	Acceso legal?
Lectura de instrucción	0x1000			
Lectura de instrucción	0x1004			
Lectura de variable	0x97FC			
Lectura de instrucción	0x1008			
Escritura de variable	0x97F8			
Lectura de instrucción	0x5000			

¿Cuántos marcos ocupa el proceso justo después del quinto acceso?

¿Podría continuar el proceso después del último acceso?

Ejercicio 2: Paginación con MV

fSO

Considere el procesador del ejercicio 1,
páginas de 4 KB,
con memoria virtual.

Orden de asignación creciente: marco 0,
marco 1, marco 2, etc.

región	tamaño (bytes)	base
código	2800	0x1000
variables	1198	0x3000
pila	2048	0x9000

Acceso	Dirección lógica	Dirección física	Fallo de página?	Acceso legal?
Lectura de instrucción	0x1000	0x0000	Sí (asigna marco 0)	Sí
Lectura de instrucción	0x1004	0x0004	No	Sí
Lectura de variable	0x97FC	0x17FC	Sí (asigna marco 1)	Sí
Lectura de instrucción	0x1008	0x0008	No	Sí
Escritura de variable	0x97F8	0x17F8	No	Sí
Lectura de instrucción	0x5000	---	Sí	No

¿Cuántos marcos ocupa el proceso justo después del quinto acceso?

2 marcos (código y pila)

¿Podría continuar el proceso después del último acceso?

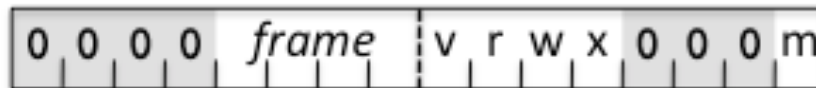
Después de un acceso ilegal a memoria el Sistema Operativo normalmente termina el proceso. No seguiría ejecutándose.

Ejercicio 2. Apartado 2

Describa *en hexadecimal* la evolución de la tabla de páginas del proceso.

Considere que la memoria física máxima es de 64 KB y que en el momento del inicio del proceso sólo están ocupados los dos marcos superiores. El sistema asigna marcos libres por orden creciente de dirección física.

Los descriptores de páginas ocupan 16 bits y tienen la siguiente estructura:



Si una página no existe en el mapa, se ha de indicar con $r=w=x=0$

Estado inicial

0	0 ? 0 0
1	0 ? 5 0
2	0 ? 0 0
3	0 ? 6 0
4	0 ? 0 0
5	0 ? 0 0
6	0 ? 0 0
7	0 ? 0 0
8	0 ? 0 0
9	0 ? 6 0
A	0 ? 0 0
B	0 ? 0 0
C	0 ? 0 0
D	0 ? 0 0
E	0 ? 0 0
F	0 ? 0 0

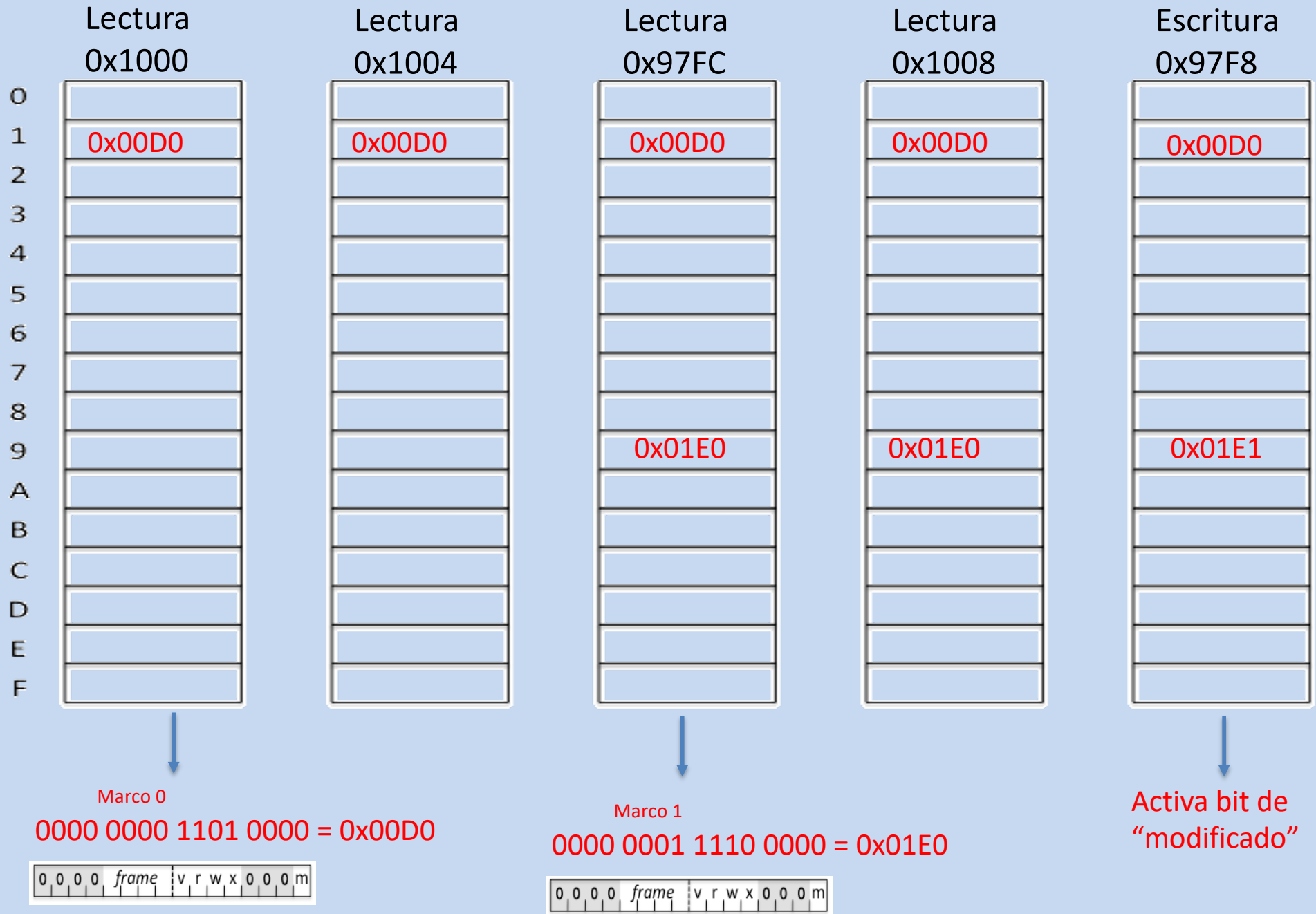
Ejercicio 2. Apartado 2

fSO

	Lectura 0x1000	Lectura 0x1004	Lectura 0x97FC	Lectura 0x1008	Escritura 0x97F8
0					
1					
2					
3					
4					
5					
6					
7					
8					
9					
A					
B					
C					
D					
E					
F					

Ejercicio 2. Apartado 2

fSO



En un computador con **32 MB de memoria principal**, con **memoria virtual paginada** con páginas de 4 KB, el sistema operativo asigna al proceso A **4 marcos de memoria principal (del 0 al 3)**, que inicialmente se encuentran vacíos.

Se pide:

- Describa el **formato de una dirección física de memoria**.
- Si el proceso A genera la siguiente secuencia de direcciones lógicas (mostradas en hexadecimal):

02D4B8, 02D4B9, 02D4EB, 02D4EB, 02D86F, F0B621, F0B815, F05963, F0B832, F0BA23, D946C3, D9B1A7, D9B1A1, F0BA25, 02D4C7, 628A31, F0B328, D9B325, D73425

Indique, para **los algoritmos FIFO y LRU**, cuántos **fallos de página** se producen y el estado final de la memoria, anotando la página que contiene cada uno de los marcos asignados al proceso.

Ejercicio 3. Algoritmos de reemplazo

TP = 4K



Secuencia de referencias a páginas: **02D, F0B, F05, F0B, D94, D9B, F0B, 02D, 628, F0B, D9B, D73**

FIFO

Marco	02D	F0B	F05	F0B	D94	D9B	F0B	02D	628	F0B	D9B	D73
0												
1												
2												
3												

LRU

Marco	02D	F0B	F05	F0B	D94	D9B	F0B	02D	628	F0B	D9B	D73
0												
1												
2												
3												

Ejercicio 3. Algoritmos de reemplazo

TP = 4K



Secuencia de referencias a páginas: **02D, FOB, F05, FOB, D94, D9B, FOB, 02D, 628, FOB, D9B, D73**

FIFO

Marco	02D	FOB	F05	FOB	D94	D9B	FOB	02D	628	FOB	D9B	D73
0	02D	02D	02D	02D	02D	D9B	D9B	D9B	D9B	D9B	<u>D9B</u>	D73
1		FOB	F0B	<u>F0B</u>	F0B	F0B	<u>F0B</u>	02D	02D	02D	02D	02D
2			F05	F05	F05	F05	F05	F05	628	628	628	628
3					D94	D94	D94	D94	D94	FOB	F0B	F0B

LRU

Marco	02D	FOB	F05	FOB	D94	D9B	FOB	02D	628	FOB	D9B	D73
0	02D	02D	02D	02D	02D	D9B	D9B	D9B	D9B	D9B	<u>D9B</u>	D9B
1		FOB	F0B	<u>F0B</u>	F0B	F0B	<u>F0B</u>	F0B	F0B	<u>F0B</u>	F0B	F0B
2			F05	F05	F05	F05	F05	02D	02D	02D	02D	D73
3					D94	D94	D94	D94	628	628	628	628

- Existen dos alternativas en el ámbito de los algoritmos de reemplazo:
 - **Reemplazo local:**
 - Un proceso selecciona la víctima entre sus propias páginas ubicadas en marcos, es decir, únicamente se producen reemplazos en marcos que ya tiene asignados; no puede elegir marcos de otro proceso.
 - El número de marcos asignados a un proceso no cambia
 - **Reemplazo global:**
 - Un proceso selecciona para reemplazar marcos del conjunto de marcos total de la memoria principal
 - La selección de la página a reemplazar puede recaer sobre otro proceso distinto al que produce el fallo de página, es decir, se pueden seleccionar marcos de otro proceso.

Ejercicio 4: Reemplazo global y local

En un sistema de memoria virtual con páginas de 1024 bytes, el SO ha asignado 6 marcos (del 0 al 5) a dos procesos A y B. En el instante $t=10$, las tablas de páginas de ambos procesos son las siguientes:

		marco	Bit validez	contador
Tabla de páginas de A	0		i	
	1		i	
	2	2	v	10
	3	5	v	3
	4		i	
	5	4	v	5
	6		i	
	7		i	

		marco	Bit validez	contador
Tabla de páginas de B	0		i	
	1		i	
	2		i	
	3	1	v	2
	4		i	
	5		i	
	6		i	
	7		i	

A continuación, los procesos emiten la siguiente serie de direcciones. Considere que todos los accesos son legales:

A,100; A,4000; B,100; A,7000; B,2100; B,1028; A,5800; A,100

Calcule qué páginas contiene cada uno de los marcos y la dirección física en que se traducen el primer y el último acceso de la serie en los dos casos siguientes:

- 1) El SO reemplaza páginas según el algoritmo LRU de ámbito **global**
- 2) El SO reemplaza páginas según el algoritmo LRU de ámbito **local**. El proceso A dispone de cuatro marcos y el proceso B de 2

Ejercicio 4: Reemplazo global y local

A,100; A,4000; B,100; A,7000; B,2100; B,1028; A,5800; A,100

Reemplazo global:

Instante	10	11	12	13	14	15	16	17	18
dirección		A,100	A,4000	B,100	A,7000	B,2100	B,1028	A,5800	A,100
Pagina/ marco		A0	A3	B0	A6	B2	B1	A5	A0
0	-----								
1	B3,2								
2	A2,10								
3	-----								
4	A5,5								
5	A3,3								

Ejercicio 4: Reemplazo global y local

A,100; A,4000; B,100; A,7000; B,2100; B,1028; A,5800; A,100

Reemplazo global:

Instante	10	11	12	13	14	15	16	17	18
dirección		A,100	A,4000	B,100	A,7000	B,2100	B,1028	A,5800	A,100
Página/ marco		A0	A3	B0	A6	B2	B1	A5	A0
0	-----	A0,11	A0,11	A0,11	A0,11	A0,11	A0,11	A5,17	A5,17
1	B3,2	B3,2	B3,2	B3,2	A6,14	A6,14	A6,14	A6,14	A6,14
2	A2,10	A2,10	A2,10	A2,10	A2,10	A2,10	B1,16	B1,16	B1,16
3	-----	-----	-----	B0,13	B0,13	B0,13	B0,13	B0,13	B0,13
4	A5,5	A5,5	A5,5	A5,5	A5,5	B2,15	B2,15	B2,15	B2,15
5	A3,3	A3,3	<u>A3,12</u>	A3,12	A3,12	A3,12	A3,12	A3,12	A0,18

Ejercicio 4: Reemplazo global y local

A,100; A,4000; B,100; A,7000; B,2100; B,1028; A,5800; A,100

Reemplazo local:

Instante	10	11	12	13	14	15	16	17	18
dirección		A,100	A,4000	B,100	A,7000	B,2100	B,1028	A,5800	A,100
Pagina/ marco		A0	A3	B0	A6	B2	B1	A5	A0
0	-----								
1	B3,2								
2	A2,10								
3	-----								
4	A5,5								
5	A3,3								

Ejercicio 4: Reemplazo global y local

A,100; A,4000; B,100; A,7000; B,2100; B,1028; A,5800; A,100

Reemplazo local:

Instante	10	11	12	13	14	15	16	17	18
dirección		A,100	A,4000	B,100	A,7000	B,2100	B,1028	A,5800	A,100
Pagina/ marco		A0	A3	B0	A6	B2	B1	A5	A0
0	-----	A0,11	A0,11	A0,11	A0,11	A0,11	A0,11	A0,11	<u>A0,18</u>
1	B3,2	B3,2	B3,2	B3,2	B3,2	B2,15	B2,15	B2,15	B2,15
2	A2,10	A2,10	A2,10	A2,10	A2,10	A2,10	A2,10	A5,17	A5,17
3	-----	-----	-----	B0,13	B0,13	B0,13	B1,16	B1,16	B1,16
4	A5,5	A5,5	A5,5	A5,5	A6,14	A6,14	A6,14	A6,14	A6,14
5	A3,3	A3,3	A3,12	A3,12	A3,12	A3,12	A3,12	A3,12	A3,12