Auxiliar 10: Administración de Memoria

José Astorga

PPT adaptada de las de Pablo Jaramillo

1. Tablas de Paginamiento

Tablas de Paginamiento

Las tablas de paginamiento son en esencia un mapa entre páginas virtuales y páginas reales que permiten registrar cuál página de cierto proceso pertenece a cuál página real. Esto se hace para permitir que todos los procesos en ejecución manejen sus propias páginas internamente sin considerar las otras que existen, permitiendo al SO cargar páginas a memoria cuando son necesarias (en demanda) y descargandolas a disco cuando se necesita más espacio sin tener que comunicarle al proceso el nuevo número de la página cuando esta vuelva a ser necesitada.

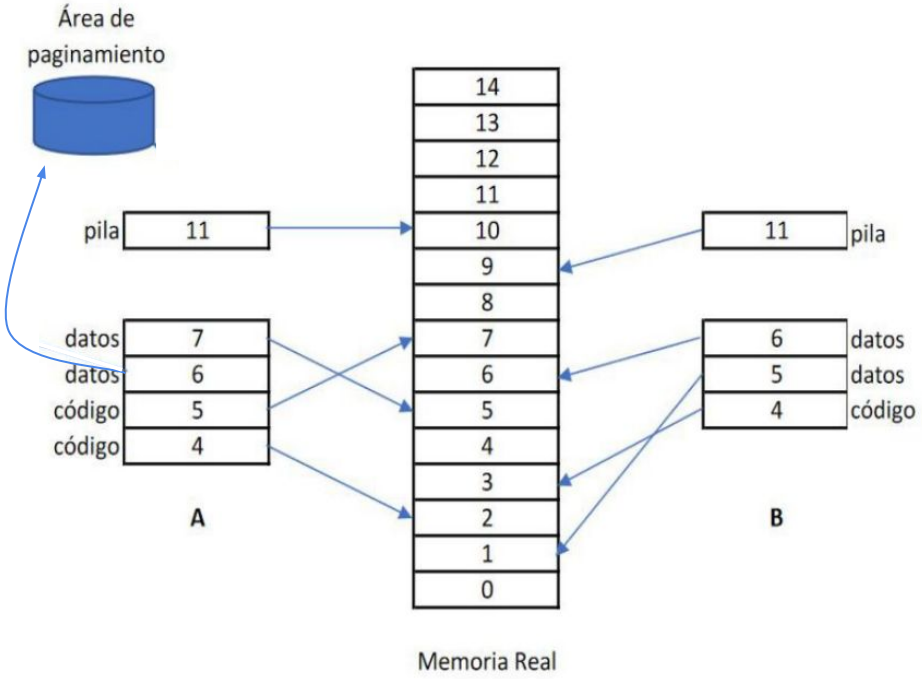
Tablas de Paginamiento - Ejercicio

El siguiente diagrama muestra la asignación de páginas en un sistema Unix que ejecuta los procesos A y B. Las páginas son de 4 KB. El núcleo utiliza la estrategia “copy-on-write” para implementar fork.

a. Construya la tabla de páginas del proceso A después de que invoca sbrk pidiendo 10 KB adicionales.

b. Considere que B invocó fork. Construya la tabla del proceso hijo justo después de que este modifique la página 11. No construya la tabla del padre.

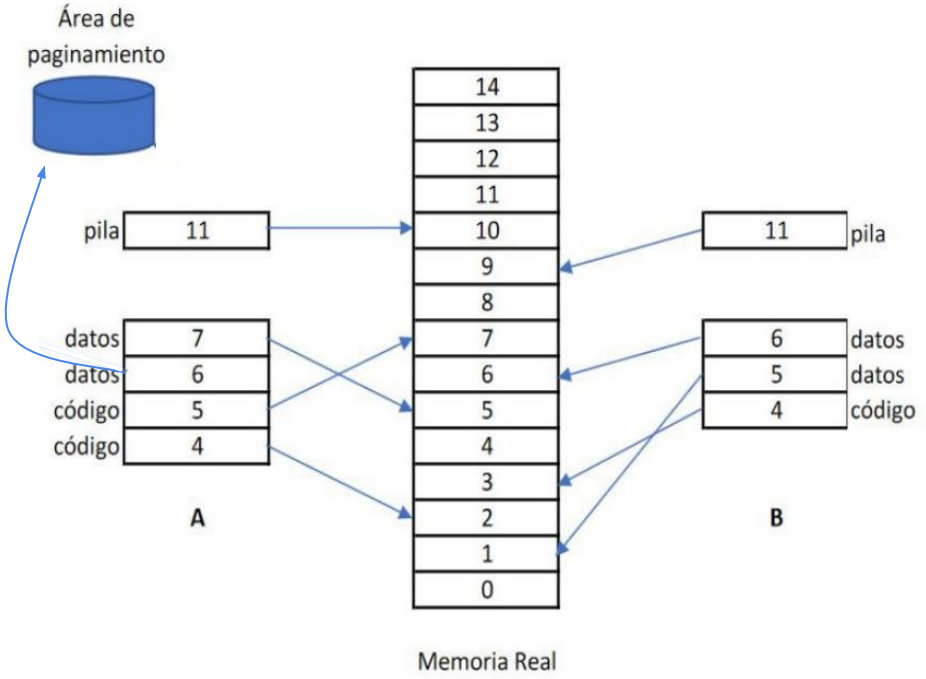
En cada caso indique la página virtual, real y atributos de validez y escritura.



Tablas de Paginamiento - Ejercicio (a)

Proceso A

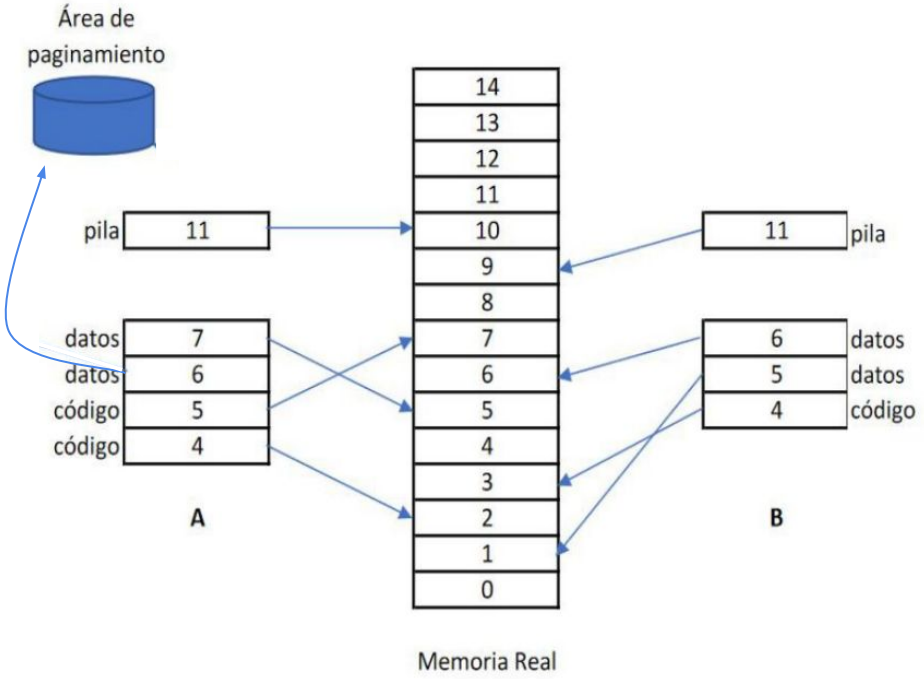
	Virtual	Real	V	W
	1	-	0	
	2	-	0	
	3	-	0	
código	4	2	1	0
código	5	7	1	0
datos	6	disco	0	0
datos	7	5	1	1
	8	-	0	
	9	-	0	
	10	-	0	
pila	11	10	1	1



Tablas de Paginamiento - Ejercicio (a)

Proceso A + 10KB

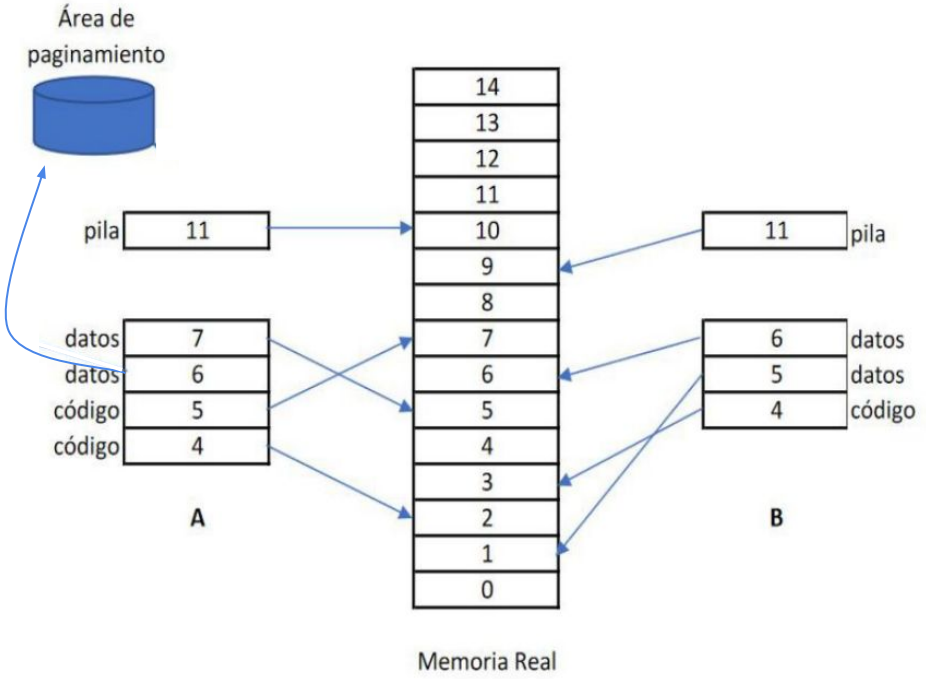
	Virtual	Real	V	W
	1	-	0	
	2	-	0	
	3	-	0	
código	4	2	1	0
código	5	7	1	0
datos	6	disco	0	0
datos	7	5	1	1
	8	8	1	1
	9	4	1	1
	10	0	1	1
pila	11	10	1	1



Tablas de Paginamiento - Ejercicio (b)

Proceso B

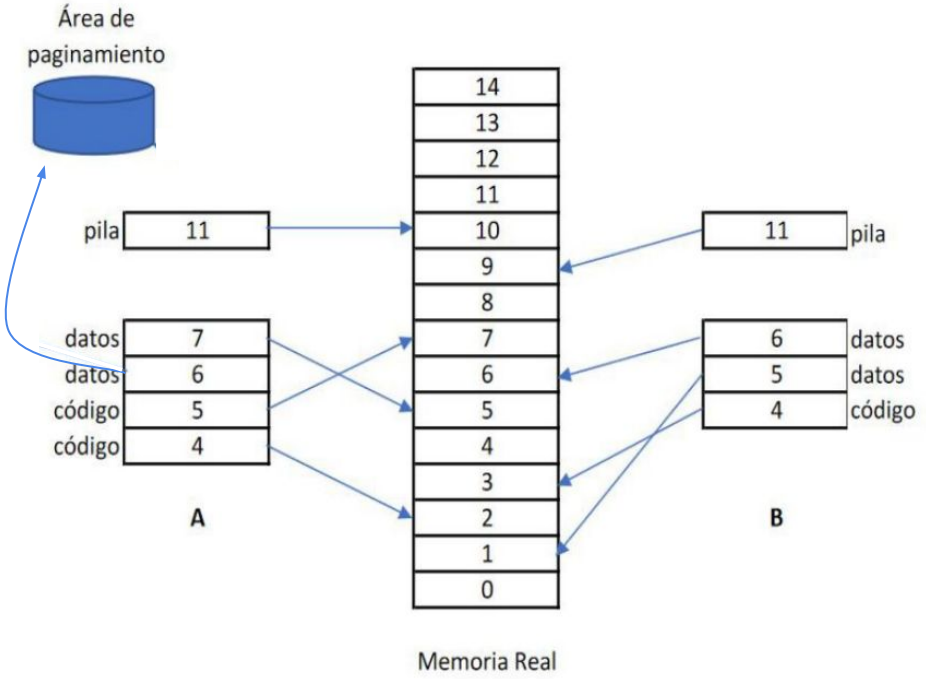
	Virtual	Real	V	W
	1	-	0	
	2	-	0	
	3	-	0	
código	4	3	1	0
datos	5	1	1	1
datos	6	6	1	1
	7		0	
	8		0	
	9		0	
	10		0	
pila	11	9	1	1



Tablas de Paginamiento - Ejercicio (b)

Proceso B Fork

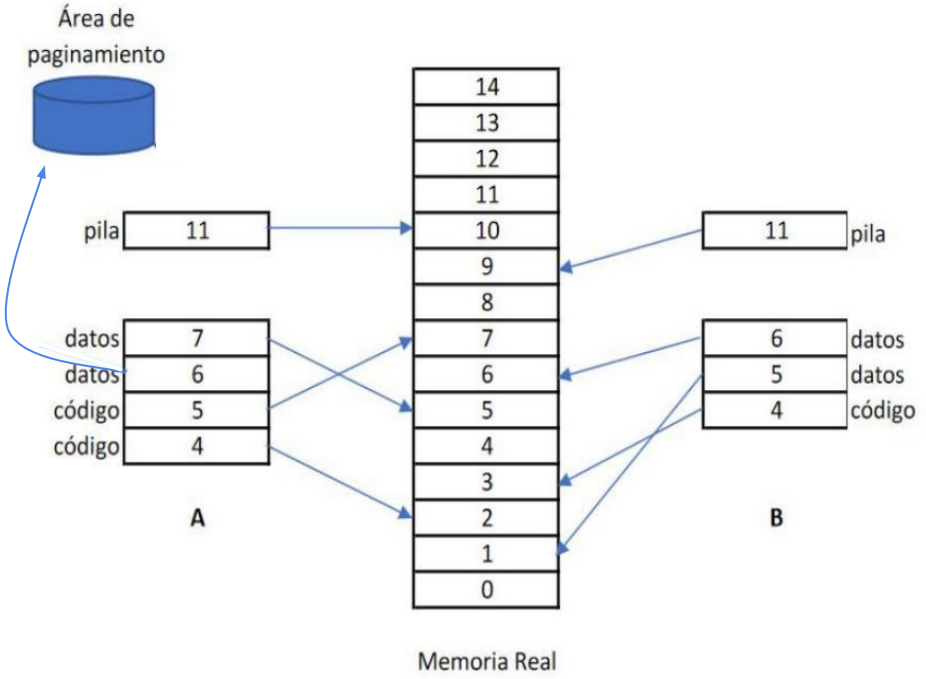
	Virtual	Real	V	W	COW
	1	-	0		
	2	-	0		
	3	-	0		
código	4	3	1	0	0
datos	5	1	1	0	1
datos	6	6	1	0	1
	7		0		
	8		0		
	9		0		
	10		0		
pila	11	9	1	0	1



Tablas de Paginamiento - Ejercicio (b)

Proceso B Fork + modificación en la pág 11

	Virtual	Real	V	W	COW
	1	-	0		
	2	-	0		
	3	-	0		
código	4	3	1	0	0
datos	5	1	1	0	1
datos	6	6	1	0	1
	7		0		
	8		0		
	9		0		
	10		0		
pila	11	14	1	1	0



2. Estrategia del Reloj

Estrategia del Reloj

Una estrategia de paginamiento en demanda que cada vez que ocurre un page-fault cicla un puntero por números de páginas en memoria para descargar páginas de la memoria no utilizadas recientemente tal que se evite descargar páginas recientemente cargadas en la memoria.

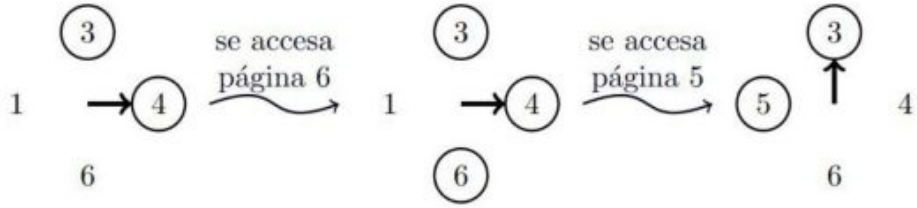
Esta estrategia es simple de implementar y tiene 0 sobrecosto cuando hay suficiente memoria.

No obstante, esta tiene problemas cuando escasea la memoria, pues entra en lo que se conoce como “thrashing”. El thrashing es cuando ocurren demasiados page faults y el computador pasa la mayor parte del tiempo manejando el disco en lugar de la CPU constantemente reemplazando una página de memoria por una que buscó desde el disco.

Estrategia del Reloj

Acá pueden apreciar una serie de estados para una estrategia del reloj tras realizar 2 accesos en un sistema de memoria para reemplazar páginas.

- Las páginas con el bit de referencia son las que tienen un círculo. La flecha representa el puntero.
- Cuando se acceda a una página que esté en el reloj se enciende el bit R de esta página, el puntero NO se mueve. (Caso del acceso a la página 6)

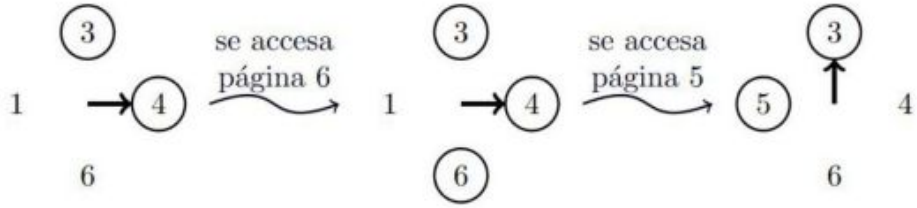


- Cuando se accede a una página que no está en el reloj se avanza el puntero apagando los bits R de cada página en el camino hasta encontrar una página con el bit R apagado y se reemplaza esta con la nueva página con el bit R encendido, luego se avanza el puntero 1 posición (Caso del acceso a la página 5)

Estrategia del Reloj

Acá pueden apreciar una serie de estados para una estrategia del reloj tras realizar 2 accesos en un sistema de memoria para reemplazar páginas.

- Las páginas con el bit de referencia son las que tienen un círculo. La flecha representa el puntero.
- Cuando se acceda a una página que esté en el reloj se enciende el bit R de esta página, el puntero NO se mueve. (Caso del acceso a la página 6)

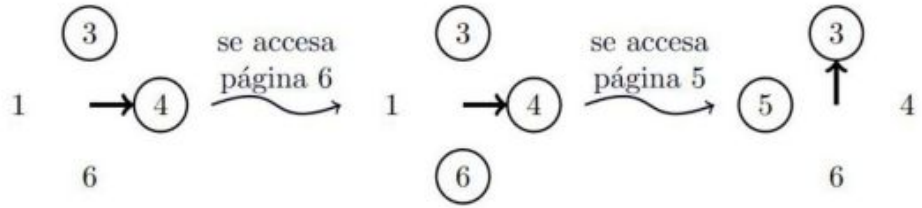


- Cuando se accede a una página que no está en el reloj se avanza el puntero apagando los bits R de cada página en el camino hasta encontrar una página con el bit R apagado y se reemplaza esta con la nueva página con el bit R encendido, luego se avanza el puntero 1 posición (Caso del acceso a la página 5)

Estrategia del Reloj - Ejercicio

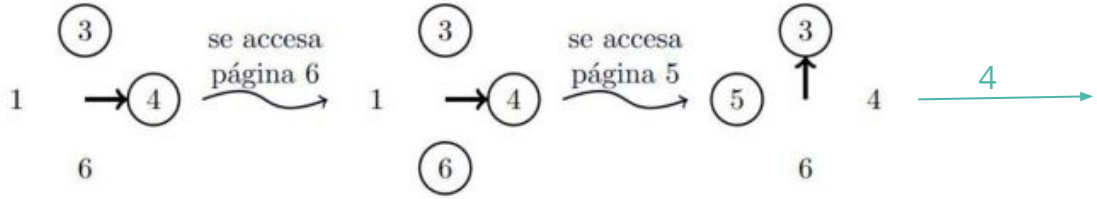
Acá pueden apreciar una serie de estados para una estrategia del reloj tras realizar 2 accesos en un sistema de memoria para reemplazar páginas.

Continue el diagrama pasando por la siguiente secuencia de acceso a páginas de memoria:
4, 7, 5, 3, 4, 1, 5



Estrategia del Reloj - Ejercicio

Continue el diagrama pasando por la siguiente secuencia de acceso a páginas de memoria: 4, 7, 5, 3, 4, 1, 5

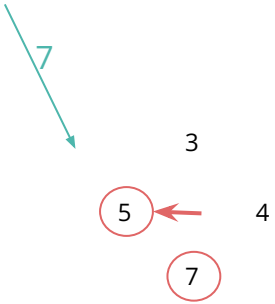


Estrategia del Reloj - Ejercicio

Continue el diagrama pasando por la siguiente secuencia de acceso a páginas de memoria: 4, 7, 5, 3, 4, 1, 5

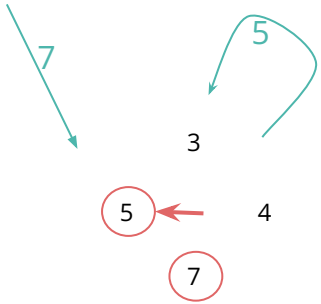
Estrategia del Reloj - Ejercicio

Continue el diagrama pasando por la siguiente secuencia de acceso a páginas de memoria: 4, 7, 5, 3, 4, 1, 5



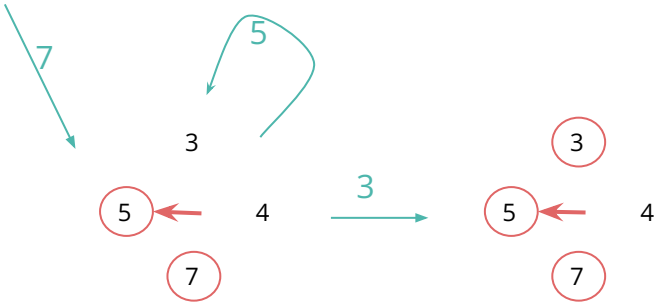
Estrategia del Reloj - Ejercicio

Continue el diagrama pasando por la siguiente secuencia de acceso a páginas de memoria: 4, 7, 5, 3, 4, 1, 5



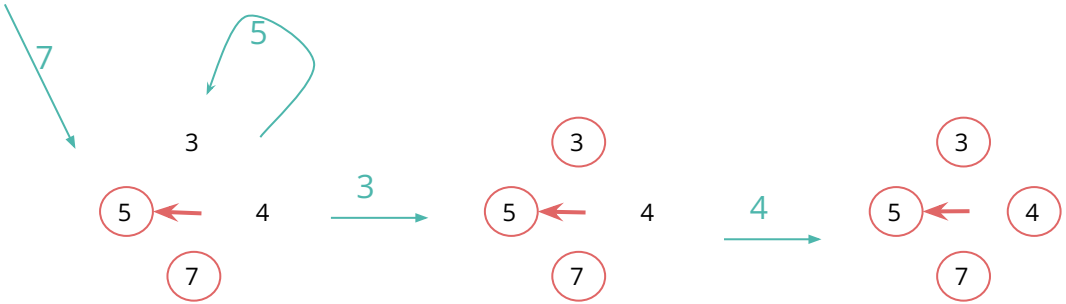
Estrategia del Reloj - Ejercicio

Continue el diagrama pasando por la siguiente secuencia de acceso a páginas de memoria: 4, 7, 5, 3, 4, 1, 5



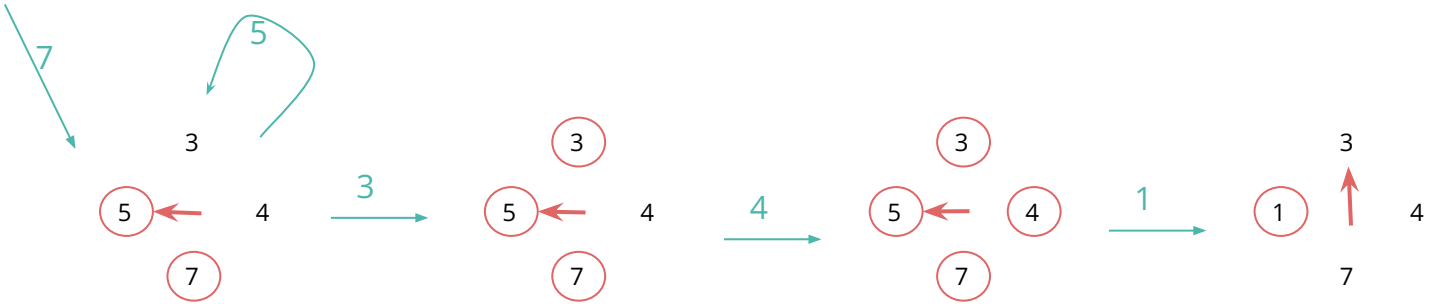
Estrategia del Reloj - Ejercicio

Continue el diagrama pasando por la siguiente secuencia de acceso a páginas de memoria: 4, 7, 5, 3, 4, 1, 5



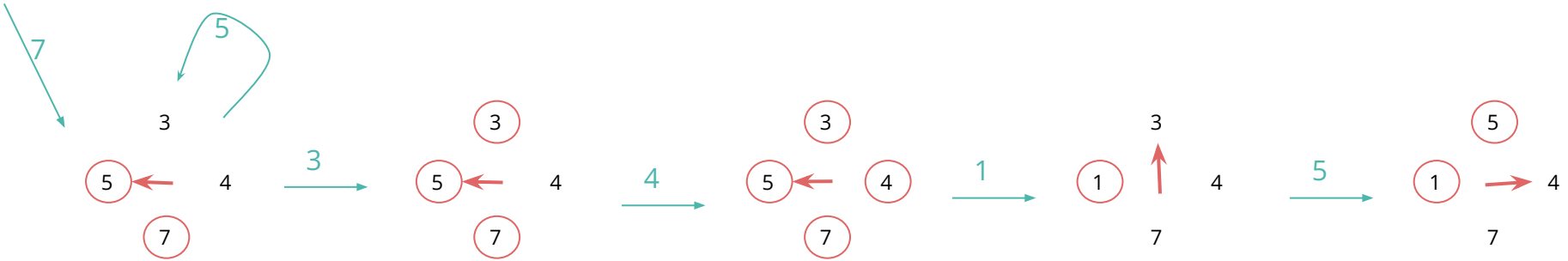
Estrategia del Reloj - Ejercicio

Continue el diagrama pasando por la siguiente secuencia de acceso a páginas de memoria: 4, 7, 5, 3, 4, 1, 5



Estrategia del Reloj - Ejercicio

Continue el diagrama pasando por la siguiente secuencia de acceso a páginas de memoria: 4, 7, 5, 3, 4, 1, 5



3. Working Set

Working Set

¿Que es la estrategia del Working Set?

Es una estrategia de paginamiento en demanda para administrar cuales páginas de un proceso el core deja en memoria basado en el supuesto de que las páginas fueron utilizadas en el pasado.

Es común que si usaste una página en un proceso que la vuelvas a usar próximamente.

Esta estrategia puede evitar el thrashing utilizando Swapping pero tiene un sobrecosto inherente por calcular los working sets.

- Page-Fault: Cuando se intenta acceder a una página que no está en memoria, causa que se obtenga la página y se lleve a memoria.

- Working Set: Conjunto de páginas de un proceso accedidas en un intervalo de tiempo.

- Bit R: "Referencia" o "Referenced", asociado a una página de un proceso para un intervalo de tiempo, es un indicador booleano que indica si esta página ha sido referenciada en un intervalo de tiempo. Se usa para calcular el Working Set.

- Bit D: "Dirty", asociado a una página de un proceso para un intervalo de tiempo, es un indicador booleano que indica si la página ha sufrido modificaciones durante dicho intervalo de tiempo. Si este valor es 0 entonces se puede descargar esta página de memoria sin copiarla a disco.

Working Set - Ejercicio

El siguiente es un diagrama de accesos a páginas de un proceso ejecutado en un sistema Unix utilizando la estrategia del Working Set. Donde:

- r es un acceso de lectura, w es un acceso de escritura.
- Las letras A, B, ..., G representan intervalos de tiempo en los cuales se calcularon working sets.
- Los números 0, 1, ..., 6 representan las páginas del proceso.

6		r	rr	ww	r	w	
5	rw	r		rrr		r	r
4		r					r
3	r		rrr		www	ww	
2	rrr	r	r	rr	wr	ww	r
1	rr	rr	r	rw			
0		ww	r		r	w	
	A	B	C	D	E	F	G

Working sets

- A: 1, 2, 3, 5
- B: 0, 1, 2, 4, 5, 6
- C: 0, 1, 2, 3, 6
- D: 1, 2, 5, 6
- E: 0, 2, 3, 6
- F: 0, 2, 3, 5, 6
- G: 2, 4, 5

Working Set - Ejercicio (a)

6		r	rr	ww	r	w	
5	rw	r		rrr		r	r
4		r					r
3	r		rrr		www	ww	
2	rrr	r	r	rr	wr	ww	r
1	rr	rr	r	rw			
0		ww	r		r	w	
	A	B	C	D	E	F	G

Working sets
- A: 1, 2, 3, 5
- B: 0, 1, 2, 4, 5, 6
- C: 0, 1, 2, 3, 6
- D: 1, 2, 5, 6
- E: 0, 2, 3, 6
- F: 0, 2, 3, 5, 6
- G: 2, 4, 5

a) Asumiendo que el bit D de todas las páginas comienza en 0. Indique cuales accesos entre los períodos C a F podrían producir page-faults. Utilice coordenadas (A, 1, 1er. acceso).

Working Set - Ejercicio (a)

6		r	rr	ww	r	w	
5	rw	r		rrr		r	r
4		r					r
3	r		rrr		www	ww	
2	rrr	r	r	rr	wr	ww	r
1	rr	rr	r	rw			
0		ww	r		r	w	
	A	B	C	D	E	F	G

Working sets
- A: 1, 2, 3, 5
- B: 0, 1, 2, 4, 5, 6
- C: 0, 1, 2, 3, 6
- D: 1, 2, 5, 6
- E: 0, 2, 3, 6
- F: 0, 2, 3, 5, 6
- G: 2, 4, 5

a) Asumiendo que el bit D de todas las páginas comienza en 0. Indique cuales accesos entre los períodos C a F podrían producir page-faults. Utilice coordenadas (A, 1, 1er. acceso).

Respuesta:
- (C, 3, 1er acceso)
- (D, 5, 1er acceso)
- (E, 3, 1er acceso)
- (E, 0, 1er acceso)
- (F, 5, 1er acceso)

Working Set - Ejercicio (b)

6		r	rr	ww	r	w	
5	rw	r		rrr		r	r
4		r					r
3	r		rrr		www	ww	
2	rrr	r	r	rr	wr	ww	r
1	rr	rr	r	rw			
0		ww	r		r	w	
	A	B	C	D	E	F	G

b) Indique el valor del atributo Referenced para todas las páginas al inicio y al final del intervalo E.

Inicio:

Página	bit R
6	
5	
4	
3	
2	
1	
0	

Fin:

Página	bit R
6	
5	
4	
3	
2	
1	
0	

Working Set - Ejercicio (b)

6		r	rr	ww	r	w	
5	rw	r		rrr		r	r
4		r					r
3	r		rrr		www	ww	
2	rrr	r	r	rr	wr	ww	r
1	rr	rr	r	rw			
0		ww	r		r	w	
	A	B	C	D	E	F	G

b) Indique el valor del atributo Referenced para todas las páginas al inicio y al final del intervalo E.

Inicio:

Página	bit R
6	0
5	0
4	0
3	0
2	0
1	0
0	0

Fin:

Página	bit R
6	1
5	0
4	0
3	1
2	1
1	0
0	1

Working Set - Ejercicio (c)

c) Suponga que al inicio del intervalo E tiene el atributo Dirty de la página 5 es 0. Explique si el acceso (D, 5, 1er acceso produjo o no un page fault)

6		r	rr	ww	r	w	
5	rw	r		rrr		r	r
4		r					r
3	r		rrr		www	ww	
2	rrr	r	r	rr	wr	ww	r
1	rr	rr	r	rw			
0		ww	r		r	w	
	A	B	C	D	E	F	G

Working Set - Ejercicio (c)

c) Suponga que al inicio del intervalo E tiene el atributo Dirty de la página 5 es 0. Explique si el acceso (D, 5, 1er acceso produjo o no un page fault)

(Visualización Bit D de página 5)

6		r	rr	ww	r	w	
5	rw	r		rrr		r	r
4		r					r
3	r		rrr		www	ww	
2	rrr	r	r	rr	wr	ww	r
1	rr	rr	r	rw			
0		ww	r		r	w	
	A	B	C	D	E	F	G

Working Set - Ejercicio (c)

c) Suponga que al inicio del intervalo E tiene el atributo Dirty de la página 5 es 0. Explique si el acceso (D, 5, 1er acceso produjo o no un page fault)

(Visualización Bit D de página 5)

6		r	rr	ww	r	w	
5	0 rw	r		rrr	0	r	r
4		r					r
3	r		rrr		www	ww	
2	rrr	r	r	rr	wr	ww	r
1	rr	rr	r	rw			
0		ww	r		r	w	
	A	B	C	D	E	F	G

Working Set - Ejercicio (c)

c) Suponga que al inicio del intervalo E tiene el atributo Dirty de la página 5 es 0. Explique si el acceso (D, 5, 1er acceso produjo o no un page fault)

(Visualización Bit D de página 5)

6		r	rr	ww	r	w	
5	0 rw 1	r		rrr	0	r	r
4		r					r
3	r		rrr		www	ww	
2	rrr	r	r	rr	wr	ww	r
1	rr	rr	r	rw			
0		ww	r		r	w	
	A	B	C	D	E	F	G

Working Set - Ejercicio (c)

c) Suponga que al inicio del intervalo E tiene el atributo Dirty de la página 5 es 0. Explique si el acceso (D, 5, 1er acceso produjo o no un page fault)

(Visualización Bit D de página 5)

6		r	rr	ww	r	w	
5	0 rw 1	r 1		rrr	0	r	r
4		r					r
3	r		rrr		www	ww	
2	rrr	r	r	rr	wr	ww	r
1	rr	rr	r	rw			
0		ww	r		r	w	
	A	B	C	D	E	F	G

Working Set - Ejercicio (c)

c) Suponga que al inicio del intervalo E tiene el atributo Dirty de la página 5 es 0. Explique si el acceso (D, 5, 1er acceso produjo o no un page fault)

(Visualización Bit D de página 5)

6		r	rr	ww	r	w	
5	0 rw 1	r 1		0 rrr	0	r	r
4		r					r
3	r		rrr		www	ww	
2	rrr	r	r	rr	wr	ww	r
1	rr	rr	r	rw			
0		ww	r		r	w	
	A	B	C	D	E	F	G

Working Set - Ejercicio (c)

c) Suponga que al inicio del intervalo E tiene el atributo Dirty de la página 5 es 0. Explique si el acceso (D, 5, 1er acceso produjo o no un page fault)

(Visualización Bit D de página 5)

6		r	rr	ww	r	w	
5	0 rw 1	r 1		A disco 0 rrr	0	r	r
4		r					r
3	r		rrr		www	ww	
2	rrr	r	r	rr	wr	ww	r
1	rr	rr	r	rw			
0		ww	r		r	w	
	A	B	C	D	E	F	G

Sí hubo page fault. Puesto que de no haber habido entonces D debería ser 1.

La escritura (A, 5, 2do acceso) dejó el bit D de la página 5 en 1. La única forma de que D sea 0 es que se haya descargado esta página y copiado a disco en el intervalo C y cargada por el page-fault producido por (D, 5, 1er. acceso), estableciendo el bit D como 0.

Working Set - Ejercicio (d)

Compare las dos estrategias de paginamiento en demanda vistas en el curso según la tabla:

	Reloj	Working Set
Sobrecosto en tiempo de ejecución cuando sobra la memoria física		
Page-faults cuando hay penuria de memoria pero hay solo 1 proceso en ejecución		
Page-faults cuando hay penuria de memoria y hay muchos procesos en ejecución		

Working Set - Ejercicio (d)

Compare las dos estrategias de paginamiento en demanda vistas en el curso según la tabla:

	Reloj	Working Set
Sobrecosto en tiempo de ejecución cuando sobra la memoria física	0 Sobrecosto	Sobrecosto fijo: Calcular el working set
Page-faults cuando hay penuria de memoria pero hay solo 1 proceso en ejecución		
Page-faults cuando hay penuria de memoria y hay muchos procesos en ejecución		

Working Set - Ejercicio (d)

Compare las dos estrategias de paginamiento en demanda vistas en el curso según la tabla:

	Reloj	Working Set
Sobrecosto en tiempo de ejecución cuando sobra la memoria física	0 Sobrecosto	Sobrecosto fijo: Calcular el working set
Page-faults cuando hay penuria de memoria pero hay solo 1 proceso en ejecución	Muchos, incluso puede producer thrashing	Muchos, incluso puede producer thrashing
Page-faults cuando hay penuria de memoria y hay muchos procesos en ejecución		

Working Set - Ejercicio (d)

Compare las dos estrategias de paginamiento en demanda vistas en el curso según la tabla:

	Reloj	Working Set
Sobrecosto en tiempo de ejecución cuando sobra la memoria física	0 Sobrecosto	Sobrecosto fijo: Calcular el working set
Page-faults cuando hay penuria de memoria pero hay solo 1 proceso en ejecución	Muchos, incluso puede producer thrashing	Muchos, incluso puede producer thrashing
Page-faults cuando hay penuria de memoria y hay muchos procesos en ejecución	Muchos, incluso puede producer thrashing	Una cantidad razonable, pues se recurre a swapping

4. TLB

Translation Lookaside Buffer (TLB)

Una memoria caché especial diseñada para traducir páginas virtuales a páginas reales.

Los procesos cada uno nombra sus propias páginas tal que cada proceso pueda funcionar independientemente de los otros, y el acceso a sus páginas es manejado por el SO. Esto pues cada proceso puede tener un límite de páginas reales a la vez, por lo que cada proceso conocerá sus páginas y el SO las pone en memoria cuando se necesitan.

La TLB guarda pares virtual-real, para correlacionar (traducir) un número de página virtual a uno real. Como es caché tiene un límite pequeño por lo que pueden ocurrir desaciertos (no se encuentra en el caché). Cuando no se encuentra una traducción en la TLB se debe buscar está en la memoria.

TLB: Translation Lookaside Buffer

TLB - Ejercicio (P2.a C3 2012-1)

Considere las siguientes implementaciones de un diccionario.

```
typedef struct { char key[8]; char data[24]; } Entry;
```

`Entry dict[1000];`	`Entry *dict[1000];`

En la primera implementación toda la información se encuentra contigua en la memoria. Mientras que para la segunda las entradas pueden estar muy dispersas en la memoria puesto que el heap que maneja malloc está fragmentado. Considere que este diccionario funciona con búsqueda secuencial, y páginas de 4kB

a) Estime para ambas implementaciones el peor caso del número de fallas en la TLB en una búsqueda.

TLB - Ejercicio (P2.a C3 2012-1)

Considere las siguientes implementaciones de un diccionario.

```
typedef struct { char key[8]; char data[24]; } Entry;
```

`Entry dict[1000];`	`Entry *dict[1000];`

En la primera implementación toda la información se encuentra contigua en la memoria. Mientras que para la segunda las entradas pueden estar muy dispersas en la memoria puesto que el heap que maneja malloc está fragmentado. Considere que este diccionario funciona con búsqueda secuencial, y páginas de 4kB

a) Estime para ambas implementaciones el peor caso del número de fallas en la TLB en una búsqueda.

- Caso dict[1000]: Como la estructura ocupa (8+24)x1000 B = 32 Kb, esta ocupará 32/4 = 8 páginas contiguas en memoria. En el peor caso la búsqueda tendrá que acceder a las 8 páginas, por lo que la TLB tendrá a lo más 8 desaciertos.
- Caso *dict[1000]: Los datos residen en un heap fragmentado, en el peor caso todas las entradas del diccionario se encuentran dispersas en 1000 páginas distintas. Por lo tanto, en el peor caso se tendrán que visitar todas estas, por lo que la TLB tendrá a lo más 1000 desaciertos.

TLB - Ejercicio (P2.a C3 2012-1)

Considere las siguientes implementaciones de un diccionario.

```
typedef struct { char key[8]; char data[24]; } Entry;
```

`Entry dict[1000];`	`Entry *dict[1000];`

En la primera implementación toda la información se encuentra contigua en la memoria. Mientras que para la segunda las entradas pueden estar muy dispersas en la memoria puesto que el heap que maneja malloc está fragmentado. Considere que este diccionario funciona con búsqueda secuencial, y páginas de 4kB

b) ¿Cuántos accesos adicionales a la memoria significa cada falla en la TLB considerando un microprocesador Intel x86?

TLB - Ejercicio (P2.a C3 2012-1)

Considere las siguientes implementaciones de un diccionario.

```
typedef struct { char key[8]; char data[24]; } Entry;
```

`Entry dict[1000];`	`Entry *dict[1000];`

En la primera implementación toda la información se encuentra contigua en la memoria. Mientras que para la segunda las entradas pueden estar muy dispersas en la memoria puesto que el heap que maneja malloc está fragmentado. Considere que este diccionario funciona con búsqueda secuencial, y páginas de 4kB

b) ¿Cuántos accesos adicionales a la memoria significa cada falla en la TLB considerando un microprocesador Intel x86?

- En un Intel x86 la búsqueda en memoria se hace en un “directorio de tablas” por lo que cada falla en la TLB nos fuerza a acceder al directorio (primer acceso) y luego un segundo para leer la tabla de páginas.

FIN