



Sistemas Operativos

Administración de Memoria

Rodrigo Urrea

Agenda Auxiliar

P1.

Tablas de
paginamiento

P2.

Estrategia del
reloj

P3.

Estrategia del
Working Set

P1.

Tablas de paginamiento

Tablas de Paginamiento

- Mapeo entre **página virtual** (proceso) y **página real** (memoria).
- Incluye **bits de atributos** que indican propiedades de la página, algunos de ellos son:
 - **V** indica si la página es o no **válida**. Acceder a una página no válida gatilla un *page fault*.
 - **W** indica si es posible o no **escribir** en la página. Al escribir en un página con W en 0 se produce un *page fault*.

Tablas de Paginamiento

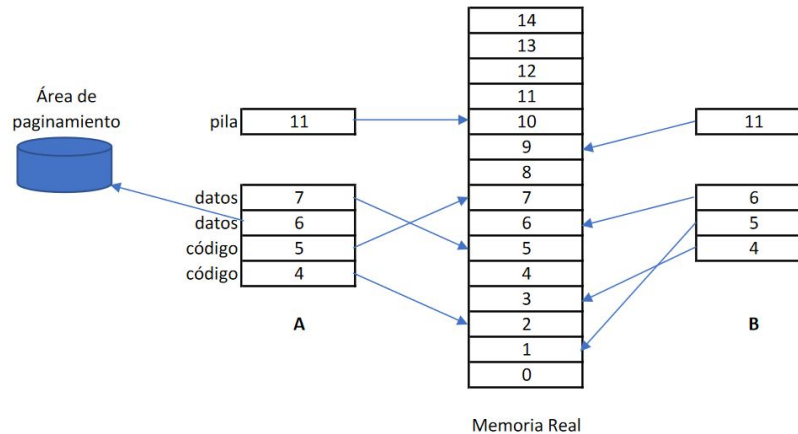
- La llamada a sistema **sbrk** permite solicitar memoria (usada por *malloc*).
- Al pedir k KB usando **sbrk** el sistema entrega p páginas al proceso tal que se cumpla:

$$p * S \leq k, \text{ donde } S \text{ es el tamaño de 1 página.}$$

- La llamada a sistema **fork** permite bifurcar el proceso actual en 2 sub-procesos (padre e hijo).
- Ambos procesos comparten las páginas, pero en todas se coloca el atributo W en 0 y el atributo COW (copy on write) en 1. Si se escribe en alguna se duplica la página.

Tablas de Paginamiento

- 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.



Tablas de Paginamiento

- Genere las tablas de paginamiento de los procesos A y B. En las tablas indique página virtual, página real y atributos de validez y escritura

Tablas de Paginamiento

- Genere las tablas de paginamiento de los procesos A y B. En las tablas indique página virtual, página real y atributos de validez y escritura

Proceso A

Pág. Virtual	Pág. Real	V	W
0	-	0	-
1	-	0	-
2	-	0	-
3	-	0	-
4	2	1	0
5	7	1	0
6	(disco)	0	0
7	5	1	1
8	-	0	-
9	-	0	-
10	-	0	-
11	10	1	1

Proceso B

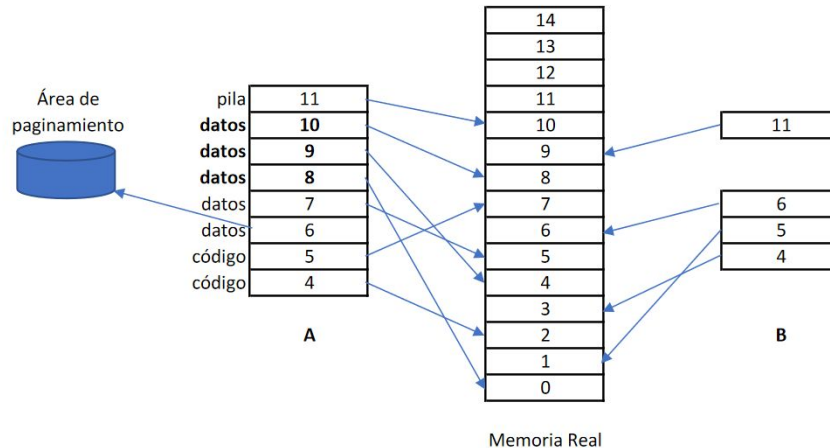
Pág. Virtual	Pág. Real	V	W
0	-	0	-
1	-	0	-
2	-	0	-
3	-	0	-
4	3	1	0
5	1	1	1
6	6	1	1
7	-	0	-
8	-	0	-
9	-	0	-
10	-	0	-
11	9	1	1

Tablas de Paginamiento

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

Tablas de Paginamiento

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



Proceso A

Pág. Virtual	Pág. Real	V	W
0	-	0	-
1	-	0	-
2	-	0	-
3	-	0	-
4	2	1	0
5	7	1	0
6	(disco)	0	0
7	5	1	1
8	0	1	1
9	4	1	1
10	8	1	1
11	10	1	1

Tablas de Paginamiento

- Considere que el proceso B invocó fork. Construya la tabla de páginas para el proceso hijo justo después de que este modificó la página 11.

Tablas de Paginamiento

- Considere que el proceso B invocó fork. Construya la tabla de páginas para el proceso hijo justo después de que este modificó la página 11.

Proceso hijo de B

Pág. Virtual	Pág. Real	V	W
0	-	0	-
1	-	0	-
2	-	0	-
3	-	0	-
4	3	1	0
5	1	1	0
6	6	1	0
7	-	0	-
8	-	0	-
9	-	0	-
10	-	0	-
11	11	1	1

P2.

Estrategia del Reloj

Estrategia del Reloj

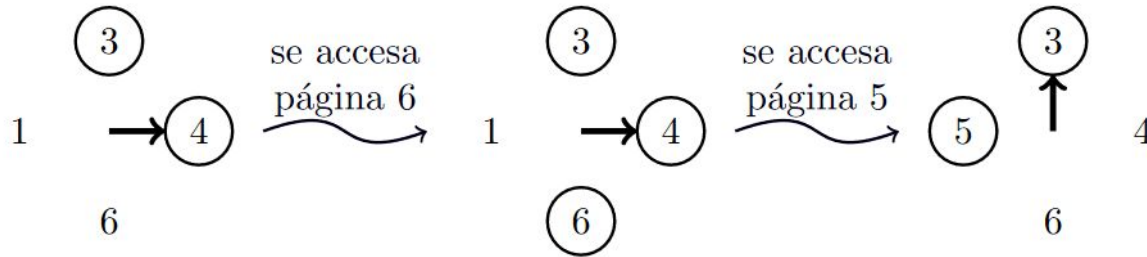
- Estrategia de paginamiento en demanda que sirve para determinar cuál de las páginas actualmente en memoria debe ser enviada a disco para almacenar una nueva página en la memoria.
- Al momento de querer leer una página p :
 - Si p se encuentra en memoria, poner su atributo R en 1.
 - Si no, buscar en el **reloj** una página para intercambiar. Sea q la página que actualmente apunta la manecilla del reloj:
 - Si q tiene el atributo R en 1, pasar R a 0 y seguir.
 - Si q tiene el atributo R en 0, **reemplazar q por p** .

Estrategia del Reloj

- El problema de esta estrategia es que produce el fenómeno conocido como **Trashing**, este se produce cuando ocurren **muchos page fault** (más frecuente al trabajar con **múltiples procesos**) lo que produce que el **disco se sature** de llamados (llegando a un 100% de uso), pero que la **CPU no se utilice** (llegando a un 0% de uso).

Estrategia del Reloj

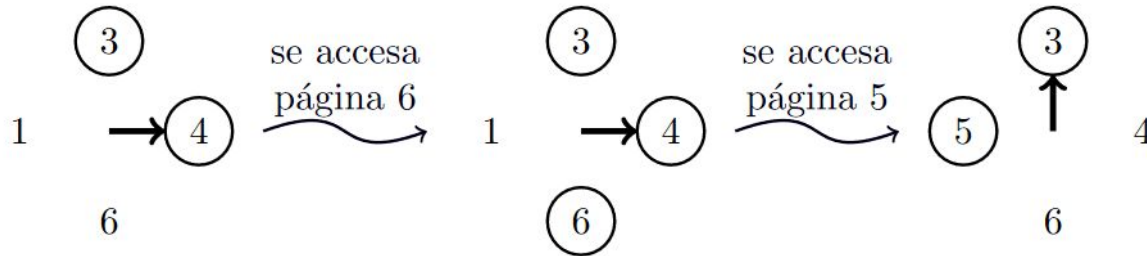
- La siguiente figura muestra los estados sucesivos de la memoria al realizar 2 accesos en un sistema de memoria virtual que usa la estrategia del reloj para el reemplazo de páginas.



- Las páginas que tienen el bit de referencia en 1, aparecen encerradas en una circunferencia. La posición del puntero está señalada por la flecha.

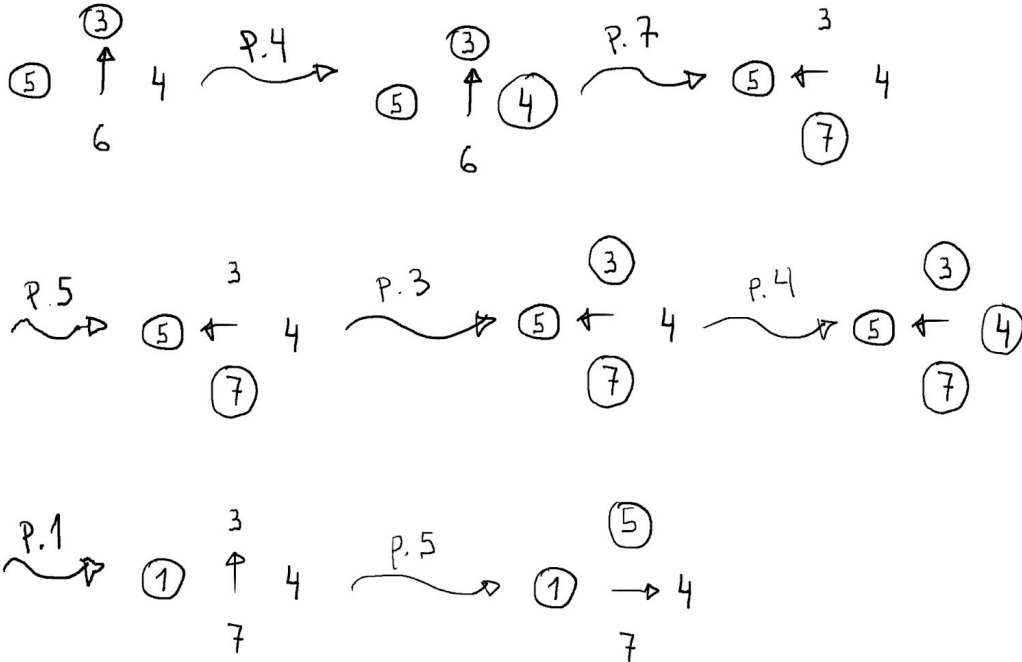
Estrategia del Reloj

- En la figura se observa una primera transición de estados cuando se accesa la **página 6 (residente)** y una segunda transición al acceder la **página 5 (no residente)**. A continuación se accesan las siguientes páginas de memoria: **4 7 5 3 4 1 5**.
- Siguiendo el mismo esquema de la figura, muestre los estados de la memoria después de realizar cada uno de los accesos indicados.



Estrategia del Reloj

Accesos: ~~4~~ ~~7~~ ~~5~~ ~~3~~ ~~4~~ ~~1~~ ~~5~~



P3.

Estrategia del Working Set

Estrategia del Working Set

- Estrategia de reemplazo de páginas que combina la estrategia del reloj con swapping.
- Busca solucionar el problema de trashing que se produce al usar la estrategia del reloj.
- Sigue la lógica de que si uno usó anteriormente la página es probable que vuelva a usarla en un futuro cercano.
- Tiene un sobrecosto debido al cálculo de los working sets.

Estrategia del Working Set

- El **working set** es el conjunto de páginas en un tiempo **t** que fueron accedidas (su atributo **R** en 1) en el instante **t-1**. Las páginas que pertenecen al working set tienen su atributo **WS** en 1. Este atributo se actualiza periódicamente.
- Se define el atributo **D** (dirty) que indica si una página ha sido o no modificada, con el fin de evitar guardar innecesariamente páginas que no han sido modificadas. Al modificar una página se coloca su atributo **D** en 1.

Estrategia del Working Set

P2.c Control 3 2012/1

- El siguiente diagrama muestra con una *r* las lecturas en memoria y con una *w* las escrituras para un proceso que se ejecuta en un sistema Unix que utiliza la estrategia del working set.
- Las letras A, B, C, etc. denotan los intervalos para los que se calcula el working set. Los números 0, 1, 2, etc. denotan las páginas del proceso. Suponga que al inicio de A todas las páginas tienen el atributo Dirty en falso.

Estrategia del Working Set

P2.c Control 3 2012/1

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

- Indique para los períodos C a F qué accesos pueden producir page-faults. Utilice coordenadas del estilo (G, 4, 1er. acceso).

Estrategia del Working Set

P2.c Control 3 2012/1

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

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

Estrategia del Working Set

P2.c Control 3 2012/1

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

- Indique el valor del atributo Referenced para todas las páginas al inicio del intervalo E y al final de ese intervalo.

Estrategia del Working Set

P2.c Control 3 2012/1

- Indique el valor del atributo Referenced para todas las páginas al inicio del intervalo E y al final de ese intervalo.

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

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

Estrategia del Working Set

P2.c Control 3 2012/1

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

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

Estrategia del Working Set

P2.c Control 3 2012/1

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

- Suponga que al inicio de E el atributo Dirty de la página 5 es falso. Explique si el acceso (D, 5, 1er. acceso) produjo o no un page-fault.
- **R.** Hubo page fault. De no haber existido, el bit D (dirty) debería ser verdadero, dado que hubo una escritura en (A, 5, 2do acceso). Después de esa escritura, la página se llevó a disco, y fue cargada por el page fault de (D, 5, 1er acceso), estableciendo su bit D en 0.

Estrategia del Working Set

P2.c Control 3 2012/1

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

- Indique el valor del atributo Dirty para cada página al inicio del intervalo E y al final de ese intervalo, suponiendo que no hubo ningún page-fault en el período A-E.

Estrategia del Working Set

P2.c Control 3 2012/1

- Indique el valor del atributo Dirty para cada página al inicio del intervalo E y al final de ese intervalo, suponiendo que no hubo ningún page-fault en el período A-E.

Página	D
6	1
5	1
4	0
3	0
2	0
1	1
0	1

Página	D
6	1
5	1
4	0
3	1
2	1
1	1
0	1

Estrategia del Working Set

P1.iv Control 3 2012/1

- Compare las dos estrategias de paginamiento en demanda vistas en el curso desde el punto de vista de:

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

Estrategia del Working Set

P1.iv Control 3 2012/1

- Compare las dos estrategias de paginamiento en demanda vistas en el curso desde el punto de vista de:

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

Estrategia del Working Set

P1.iv Control 3 2012/1

- Compare las dos estrategias de paginamiento en demanda vistas en el curso desde el punto de vista de:

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

Estrategia del Working Set

P1.iv Control 3 2012/1

- Compare las dos estrategias de paginamiento en demanda vistas en el curso desde el punto de vista de:

	Reloj	Working set
Sobrecosto en tiempo de ejecución cuando la memoria física sobra	0 sobrecosto	Sobrecosto fijo: Calcular el working set
Page faults cuando hay penuria de memoria pero hay un solo proceso en ejecución	Muchos, incluso se puede producir thrashing	Muchos, incluso se puede producir thrashing
Page faults cuando hay penuria de memoria y hay muchos procesos en ejecución	Muchos, incluso se puede producir thrashing	Un número razonable, se recurre a swapping

P-Extra.

Fallas en la TLB

P2.a Control 3 2012/1

En una aplicación se requiere implementar un diccionario. Se consideran 2 implementaciones:

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 memoria. En la segunda, las entradas del diccionario pueden quedar muy dispersas y desordenadas en la memoria debido a que el heap que maneja malloc está fragmentado. Suponga que las búsquedas en el diccionario son secuenciales. Estime para ambas implementaciones el peor caso del número de fallas en la TLB (Translation Lookaside Buffer) al hacer una búsqueda. ¿Cuántos accesos adicionales a la memoria significaría cada falla en la TLB considerando un microprocesador Intel x86? Considere páginas de 4kB.

- **Caso `Entry dict[1000]`:** La estructura ocupa 1000×32 bytes = 32000 bytes ≈ 8 páginas contiguas en memoria. En el peor caso, la búsqueda visita las 8 páginas, por lo que la TLB tendrá a lo más 8 desaciertos.
- **Caso `Entry *dict[1000]`:** Como los datos residen en un heap fragmentado, en el peor caso las 1000 entradas del diccionario se encuentran dispersas en 1000 páginas distintas. Por lo tanto, en el peor caso la búsqueda visita 1000 páginas, por lo que la TLB tendrá a lo más 1000 desaciertos.

P2.a Control 3 2012/1

En una aplicación se requiere implementar un diccionario. Se consideran 2 implementaciones:

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 memoria. En la segunda, las entradas del diccionario pueden quedar muy dispersas y desordenadas en la memoria debido a que el heap que maneja malloc está fragmentado. Suponga que las búsquedas en el diccionario son secuenciales. Estime para ambas implementaciones el peor caso del número de fallas en la TLB (Translation Lookaside Buffer) al hacer una búsqueda. ¿Cuántos accesos adicionales a la memoria significaría cada falla en la TLB considerando un microprocesador Intel x86? Considere páginas de 4kB.

- En relación a los accesos a memoria en un Intel x86, cada falla en la TLB incurre en 2 accesos adicionales a memoria: el primero para acceder al directorio de tablas y el segundo para leer la tabla de páginas.



Sistemas Operativos

Administración de Memoria

Rodrigo Urrea