

# Entrega 9

## Sistemas Operativos II

Mellino, Natalia

Farizano, Juan Ignacio

### Ejercicio 1

Bit de referencia											
1	2	3	4	1	2	5	1	2	3	4	5
1 (1,1) (0,0) (0,0)	1 (1,1) 2 (1,0) (0,0)	1 (1,1) 2 (1,0) 3 (0,0)	1 (1,1) 2 (1,0) 3 (1,1) 4 (0,0)	1 (1,1) 2 (1,0) 3 (1,1) 4 (1,0)	1 (1,1) 2 (1,0) 3 (1,1) 4 (1,0)	5 (1,1) 2 (0,0) 3 (0,1) 4 (0,0)	5 (0,1) 1 (0,0) 3 (0,1) 4 (0,0)	5 (0,1) 1 (1,0) 3 (0,1) 2 (1,0)	5 (0,1) 1 (1,0) 3 (0,1) 2 (1,0)	4 (0,1) 1 (1,0) 3 (1,1) 2 (1,0)	4 (1,0) 5 (1,0) 3 (1,1) 2 (1,0)
M	M	M	M	H	H	M	M	M	H	M	M
3 aciertos (Hits) 9 fallos (Misses)											

### Ejercicio 2

Por lo general esta tarea es realizada por el hardware. Suponemos que específicamente la MMU cuando se solicita una lectura o escritura de memoria.

### Ejercicio 3

Esta decisión se debe a un mecanismo que se emplea para hacer que el sistema busque siempre tener espacio disponible en memoria en vez de esperar a que surja la necesidad de reemplazar un marco marcado como escrito, resultando en una mayor latencia. Entonces, para hacer esto el sistema operativo busca las páginas sucias más proclives a ser llevadas a disco (siempre que sea posible) y va actualizando la imagen en disco, borrando el bit de escritura.

### Ejercicio 4

- No es posible. Al no haber ningún mecanismo de memoria compartida, no existe la posibilidad de que dos o más procesos carguen la misma página en memoria, porque no la están compartiendo en un principio. Y no es posible que un solo proceso cargue dos veces la misma página de la swap, ya que va a estar bloqueado esperando que la página se cargue a memoria.
- Si, es posible. Como es el sistema el que decide enviar una página a swap, un proceso puede intentar utilizar esta página antes de que se efectivice la escritura (realizada por el sistema) ocasionando que se traiga del disco una versión anterior de la página.

### Ejercicio 5

- A diferencia de paginación, rastrear la memoria libre/usada resulta más sencillo.

- Es posible ceder grandes bloques de memoria contiguos para los procesos que lo necesiten, como por ejemplo los controladores de DMA.
- No se produce fragmentación externa ya que los bloques liberados tienen la posibilidad de fundirse.
- Poco desperdicio de memoria (a lo sumo la mitad de un bloque, ya que si fuera más pequeño, se pediría un bloque más pequeño).
- El Slab Allocator contribuye a reducir la fragmentación interna ya que este actúa cuando hay pedidos de memoria demasiado pequeños (menor a una página).