

Ejercicios de memoria: políticas y VM en GNU/Linux

Sistemas Operativos

14/05/2020

1. Un programa realiza la siguiente traza de accesos a memoria. Los accesos subrayados indican una escritura en esa página (i.e.: no sólo se setea el bit de uso si no que también se setea el bit de dirty).

1 7 2 4 5 3 5 1 3 5 6

Se tienen 5 páginas físicas.

Indicar el contenido de la memoria en cada acceso, contar la cantidad de fallos y de aciertos y la cantidad de lecturas y escrituras en el disco si se emplea:

- A. Segunda oportunidad mejorado
- B. Algoritmo óptimo

Nota: el libro no es del todo claro con respecto al algoritmo de segunda oportunidad mejorado. Usar la siguiente referencia: <http://courses.cs.tamu.edu/bart/cpsc410/Supplements/Slides/page-rep3.pdf> (dónde además se dan otros ejemplos de LRU y el reloj). Es muy recomendable utilizar el esquema indicado allí para rastrear correctamente el puntero (o manecilla) y los bits.

2. Comparar los resultados del ejercicio anterior con el de la plancha anterior. ¿el algoritmo óptimo siempre será mejor que FIFO?
3. La figura representa el estado actual de un sistema de memoria manejado con el Buddy System. Puede verse que hay dos bloques de tamaño 128 páginas, uno de 512 páginas y ninguno de tamaño 256 páginas. Se muestra también la ubicación de cada bloque libre en la memoria RAM.

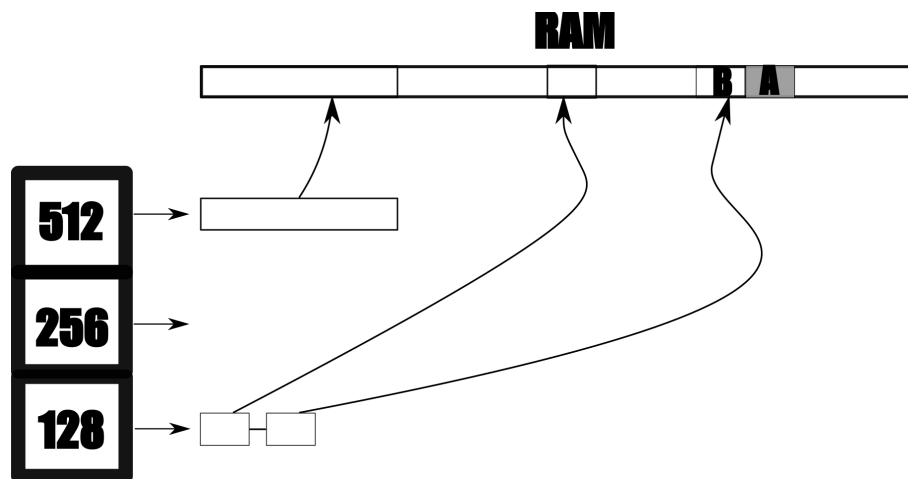


Figura 1: Estado actual del sistema Buddy

- (a) Mostrar el estado resultante si se libera el bloque "A" de tamaño 128. Nota: el bloque "B" comienza en la dirección $16 \times 256 \times PageSize$.
 - (b) Mostrar el estado resultante si, luego de haber liberado "A", se pide un bloque de tamaño 128.
4. Un swap-in y un swap-out concurrentes pueden ocasionar que se traiga del disco una versión anterior de la página ¿qué consecuencias puede traer eso? Explicar con un ejemplo.