

Práctica 3

Objetivo

El objetivo de esta práctica es poder ejecutar el programa *halt*, *matmult* y *sort* en nachos implementando memoria virtual con paginación por demanda pura utilizando el algoritmo de reemplazo FIFO.

Puntos a implementar:

1. Tomando como base la práctica 2 implementar el algoritmo de reemplazo FIFO, siguiendo el procedimiento

Pasos para el reemplazo de páginas

- a) Atender la excepción de fallo de página y encontrar la ubicación de la página deseada en disco (archivo de intercambio)
 - b) Encontrar un marco libre:
 - Si hay un marco libre, usarlo
 - Si no hay un marco libre, usar el algoritmo de reemplazo FIFO para elegir la página víctima
 - Escribir la página víctima en disco(solo si fue modificada), actualizar tabla de páginas y de marcos
 - c) Traer la página deseada al (nuevo) marco libre; actualizar las tablas de páginas y de marcos
 - d) Continuar con la ejecución del proceso
2. Cuando se haya elegido una víctima con el algoritmo de reemplazo se deberá verificar si esa página fue modificada para respaldarla en su archivo de intercambio(swap out), en caso contrario, no se hace nada debido a que la información que está en memoria es idéntica a la del archivo de intercambio.
4. Se debe identificar la variable del SO que tiene la cuenta de las lecturas a disco duro e incrementarla cada que se realice un intercambio hacia adentro (swap in).
5. Se debe identificar la variable del SO que tiene la cuenta de las escrituras a disco duro e incrementarla cada que se realice un intercambio hacia adentro (swap in).
6. Se debe imprimir en el resumen final de la máquina el número total de ***fallos*** de páginas, el número de ***lecturas*** y ***escrituras*** en disco al finalizar la ejecución del programa(como se muestra), donde,
- a) el número de lecturas tiene que ser igual al número de fallos y,
 - b) el número de escrituras tiene que ser menor al número de fallos, ya que solamente se deben escribir en disco aquellas páginas que hayan sido modificadas.

Machine halting!

Cadena de referencia: .

Ticks: total 23, idle 0, system 10, user 13

Disk I/O: reads 3, writes 0

Console I/O: reads 0, writes 0

Paging: faults 3

Network I/O: packets received 0, sent 0

6. Al implementar esta práctica *halt*, *matmult* y *sort* deberán ejecutarse y finalizar correctamente, mostrando el resumen de la máquina como se muestra en el punto anterior.

7. La salida que tendrá que imprimirse en terminal es el **vpn de cada página que causó el fallo de página** separadas por comas. Por ejemplo, al ejecutar el programa de *halt* la salida será

Tamaño del proceso: 1344 bytes

Cantidad de marcos que requiere para ejecutarse: 11

El archivo de intercambio ../test/halt.swp fue creado

VPN de las páginas que generaron fallo

0, 1, 10

Relación de fallos para cada programa de prueba con 32 marcos

Archivo: matmult

Marcos	FIFO
32	110

Archivo: sort

Marcos	FIFO
32	3309

Archivo: halt

Marcos	FIFO
32	3