



Universidad de
los Andes >

**FACULTAD
DE INGENIERÍA
Y CIENCIAS
APLICADAS**

Informe Tarea 2 Sistemas Operativos

Integrantes: Eduardo Saldivia
Emilio Arias
Profesor: Claudio Alvarez

17 de Octubre del 2019

modo de uso

para ejecutar el programa en la línea de comando debe escribirse
./virtmem -n npages -f nframes -a rand|fifo -p [patrón de acceso a memoria]
donde:

-npages es el número de páginas

-nframes es el número de marcos

-rand|fifo es el algoritmo de selección de víctima, los comandos válidos son:

-RAND

-FIFO

-patrón de acceso de memoria es uno de los tres patrones implementados, los comandos válidos son:

-pattern1

-pattern2

-pattern3

Algoritmos

Algoritmo 1: Patrón lineal que recorre de inicio a fin las páginas, dándoles valores a estas, para luego leer de inicio a fin todos los valores. cantidad total de usos de páginas: $npages * 2$

Algoritmo 2: Patrón aleatorio que consiste en entrar 5000 veces a una posición aleatoria de página. Luego, en el caso que esta sea par, el algoritmo escribe en ella. Mientras que en el caso que sea impar, el algoritmo lee el bloque elegido y hace escritura en el antecesor. cantidad total de usos de páginas: 7500

Algoritmo 3: Consiste de una iteración de 100 veces, donde a una variable "a" se le asigna un valor aleatorio, para luego, hacer otra iteración, pero esta vez de 20 veces, donde a una variable "b" se le asigna otro valor aleatorio, y luego, se le actualiza el valor la página en la posición "a" por el valor de la página en la posición "b". cantidad total de usos de páginas: 4000

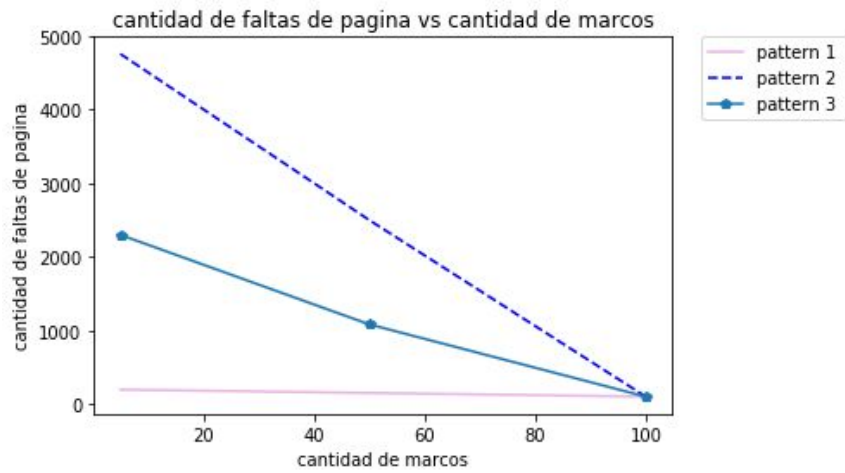


Gráfico faltas de página vs número de marcos (FIFO)

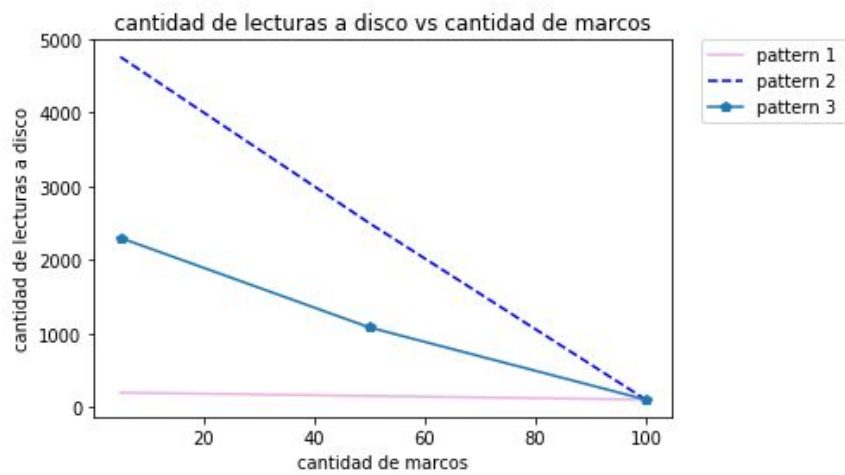


Gráfico lecturas vs número de marcos (FIFO)

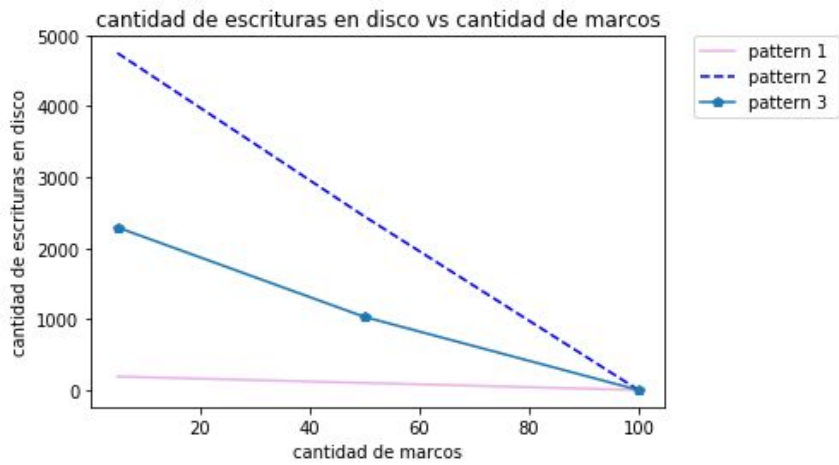


Gráfico escrituras vs número de marcos (FIFO)

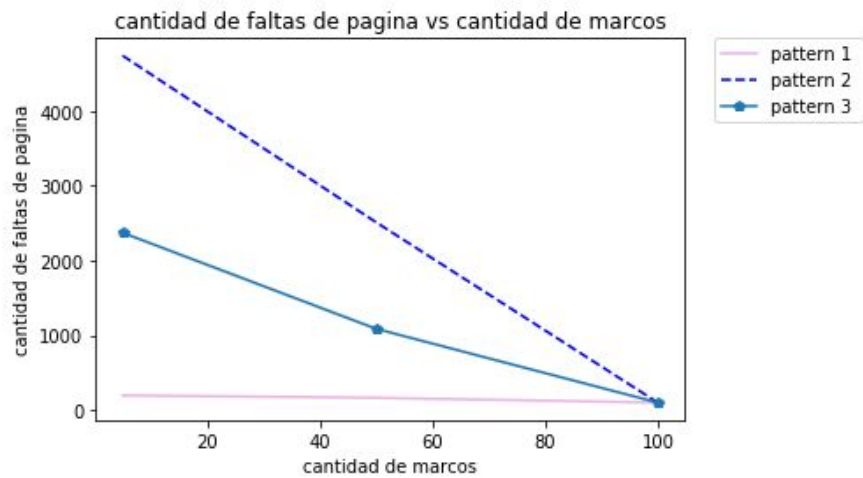


Gráfico faltas de página vs número de marcos (Rand)

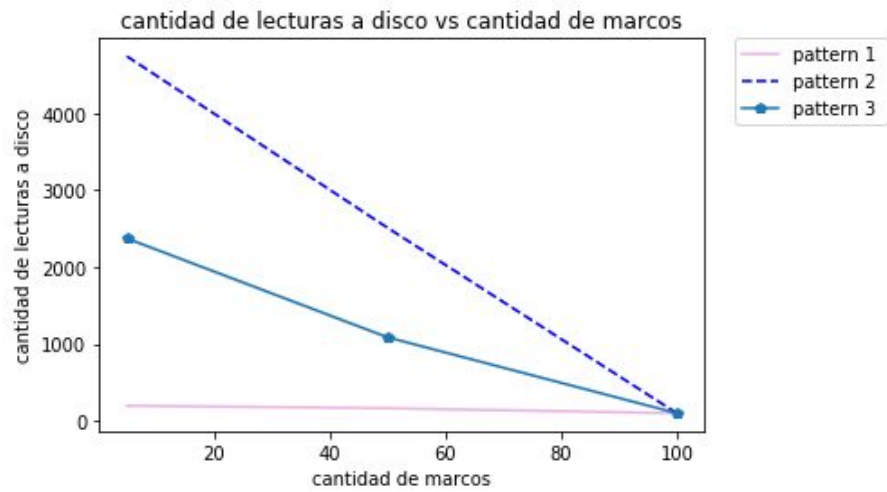


Gráfico lecturas vs número de marcos (Rand)

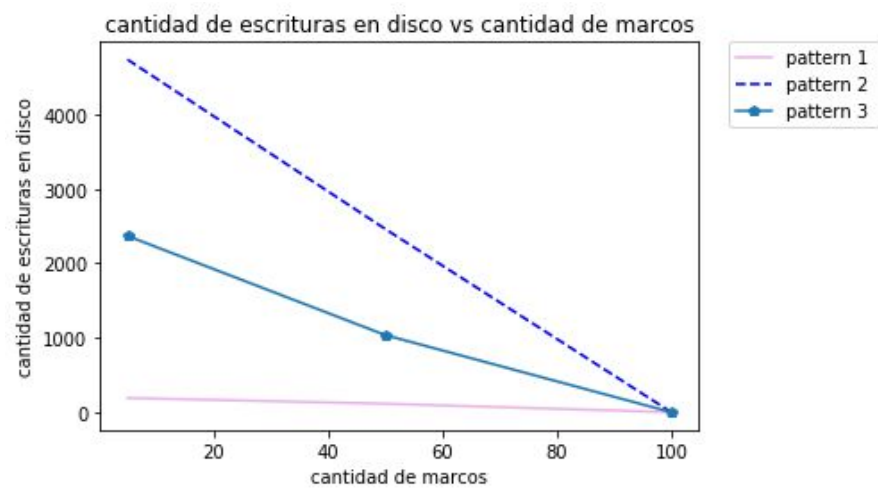


Gráfico escrituras vs número de marcos (Rand)

Hipótesis

Algoritmo 1: Como el algoritmo recorre todas las páginas por orden, debería haber tantas faltas de página como llamados a página haga.

Algoritmo 2: El rendimiento debería ser un poco mejor que en el 1, debido a que como entra a las páginas de manera aleatoria, algunas veces podría tener suerte y no tener falta de página, pero a grandes rasgos se comporta parecido al algoritmo 1.

Algoritmo 3: Debería tener tantas faltas de página como un poco más de la mitad o menos, que llamados a páginas. Esto debido a que dentro de los ciclos se actualizan los valores de tal manera que la variable b va a tener faltas de páginas mientras que la variable a va a tener casos que están cargados.

Conclusión

Concluimos que el 3er algoritmo se comporta mejor que los otros dos respecto a la cantidad de faltas de páginas debido a su método de actuar, pero esto no es sorpresa, ya que el patrón fue diseñado para simular una situación más real, en el sentido que existen páginas que son llamadas más veces que otras para usar un programa, por lo que, al ser cargadas una vez, no incurrir en más faltas de páginas una vez cargadas.

También, se concluyó que cuando se iteraba con 100 números de marcos, se obtuvo exactamente 100 faltas de páginas, esto es debido a que cuando se tiene el mismo número de marcos, que de páginas, estas al ser cargadas, incurrir en faltas de páginas una sola vez ya que luego se tienen todas las páginas cargadas en memoria por lo que si se vuelven a necesitar, ya están listas. Por lo tanto, si la cantidad de nframes es igual a la de npages, entonces se obtiene la misma cantidad de faltas de página.