



UNIVERSIDAD DE LOS ANDES  
Facultad de Ingeniería y Ciencias Aplicadas  
Sistemas Operativos y Redes 2018-20

## Tarea 2: Memoria Virtual (Paginación bajo demanda)

Integrantes:

Hugo de la Fuente, María Pía Sánchez

## 1. Introducción

La paginación bajo demanda es un sistema de paginación que busca disminuir los tiempos de respuesta y la cantidad de programas en memoria. Para lograr esto en esta tarea, se utilizarán distintos algoritmos de reemplazo de páginas, con los cuales se logrará simular su funcionamiento con procesos de distinto comportamiento, para posteriormente comparar los resultados obtenidos.

## 2. Descripción de los Algoritmos

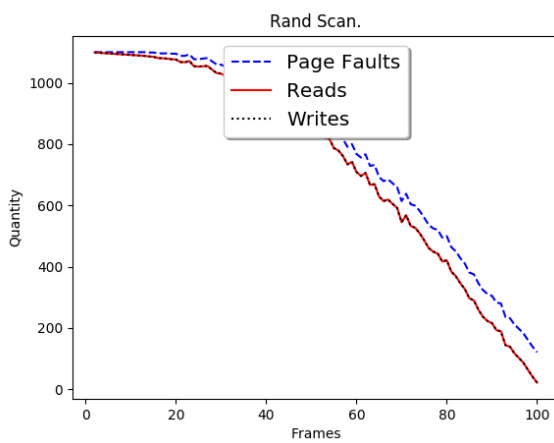
Para realizar los reemplazos de página, se utilizaron siguientes algoritmos:

1. **Random.** En este algoritmo, cuando se requiere reemplazar una página, se elige un marco al azar y luego la cambia.
2. **FIFO.** Este algoritmo es asociado con el tiempo en que una página fue cargada en memoria. Luego, cuando se tiene que reemplazar una página, siempre se elige la más antigua cargada en memoria. Para ello, se utilizó un *queue* para tener encolados los marcos que han sido utilizados.
3. **Custom: LIFO.** Este algoritmo, a diferencia del anterior, siempre reemplaza la última página cargada en memoria. Para ello, se utilizó un *stack* para tener apilados los marcos que han sido utilizados.

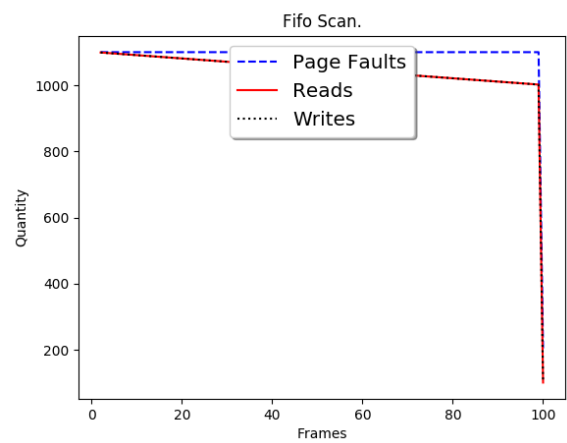
### 3. Resultados Obtenidos

Los resultados obtenidos para cada algoritmo, según los tres programas (*scan*, *sort* y *focus*), fueron los siguientes:

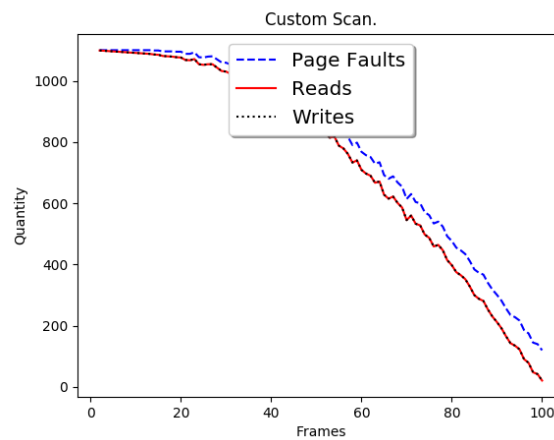
#### 1. Scan:



(a) Random



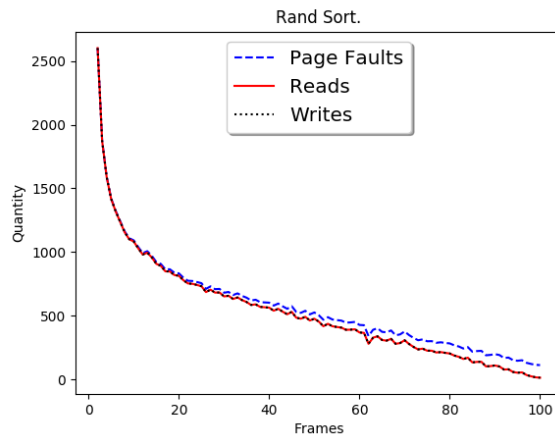
(b) FIFO



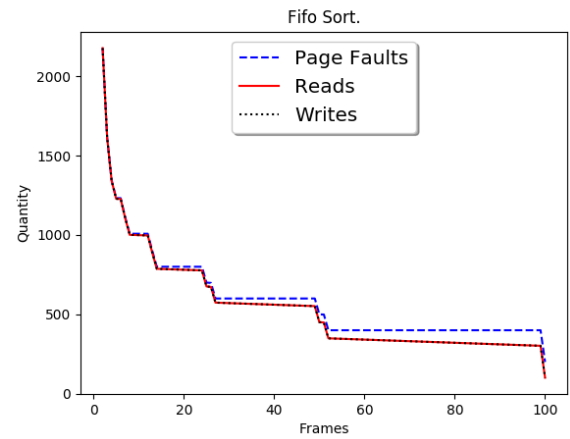
(c) Custom

Figura 1: Programa *Scan*

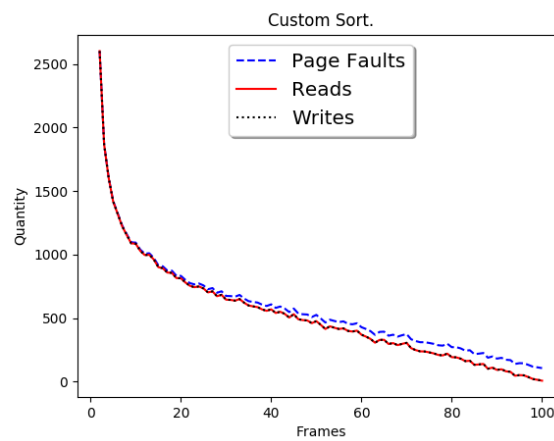
## 2. Sort:



(a) Random



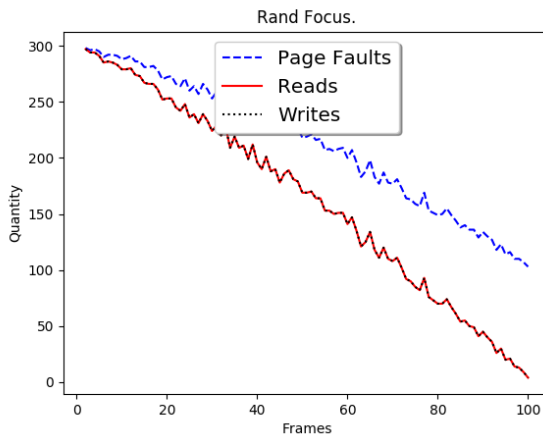
(b) FIFO



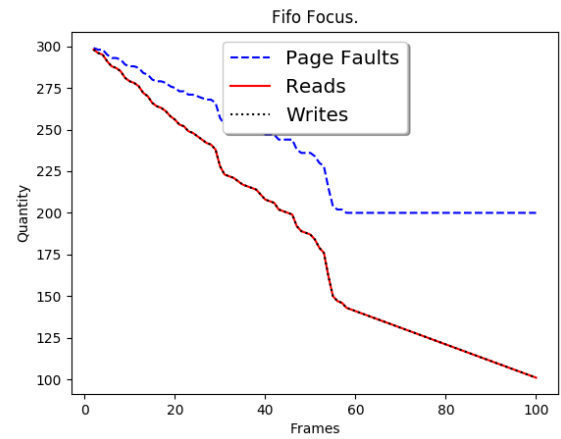
(c) Custom

Figura 2: Programa *Sort*

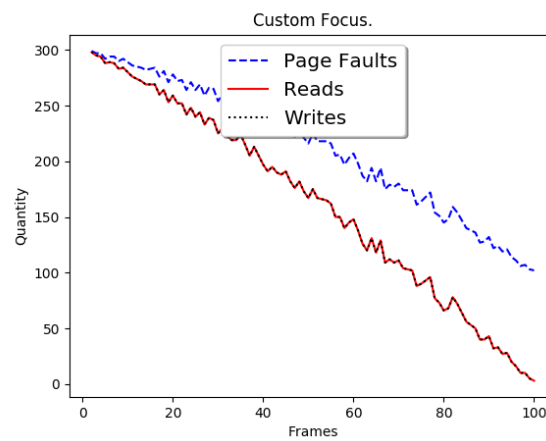
### 3. Focus:



(a) Random



(b) FIFO



(c) Custom

Figura 3: Programa *Focus*

## 4. Análisis de Resultados

Según los gráficos obtenidos, se puede observar que para el programa *Scan* el comportamiento de faltas de página del algoritmo *Random* y *Custom* son bastante parecidos, ya que a mayor cantidad de marcos, menor la falta de página con un decrecimiento parabólico, pero con crecimientos y decrecimientos locales. Esto ocurre también en las lecturas y escrituras, las cuales poseen la misma curva. Sin embargo, el crecimiento de *FIFO* al principio es lineal, y se divide en dos partes, una de decrecimiento lento, hasta llegar a los últimos marcos, el cual el decrecimiento sería bastante rápido, ocurriendo sin crecimientos y decrecimientos locales. Con esto se puede observar que la anomalía de Belady no ocurre utilizando el algoritmo *FIFO* para este programa.

En el caso del programa *Sort*, se puede observar un decrecimiento exponencial en los algoritmos para las faltas de página, las lecturas y escrituras. Sin embargo, se puede observar que en el algoritmo *FIFO*, esto ocurre de manera más escalonada que en los demás y sin anomalías de Belady, como en el algoritmo *Random* y *Custom*.

En el programa *Focus* se puede apreciar un decrecimiento lineal por parte del algoritmo *Random* y *Custom* en sus faltas de páginas, lecturas y escrituras, pero con anomalías de Belady. Sin embargo el algoritmo *FIFO* posee una curva decreciente que, a partir de aproximadamente los 50 marcos, poseen un decrecimiento rápido y que a partir del marco 54, su decrecimiento es lineal para las lecturas y escrituras, mientras que las faltas de páginas se mantienen constante, y todo esto sin anomalías de Belady.



## 5. Bibliografía

1. Link Paginación por demanda y Fallos de Páginas