

Sistemas Operativos y Redes

Tarea 2: Parte 1

Integrantes:

Nicolás Hurtado

Ángeles Aldunate

Introducción:

En esta tarea se nos pidió trabajar sobre un código parcialmente completo que le faltaba implementar la función para manejar errores de paginación. El programa actual sobre un disco virtual que cuenta con información que se quiere ser cargada en memoria. Este programa requiere que se le especifique la cantidad de páginas a cargar en el disco, la cantidad de marcos en la memoria, el algoritmo con el cual se quiere manejar los errores de paginación y por último el programa a utilizar. Los tres programas que se encuentran incluidos son sort, focus y scan. Para temas de paginación, funcionan exactamente igual, cada uno cargara la cantidad de páginas especificadas en la memoria una sola vez en la cantidad de marcos posibles, y la forma que son cargados en el marco dependen del algoritmo dado.

Lo que tuvimos que completar del código, fue implementar la función para manejar errores de paginación, la cual puede ser llevada a cabo por tres algoritmos diferentes: FIFO, Random y Custom.

El algoritmo FIFO (First in, First out) opera de la manera que la primera página en entrar (O cargar) será la primera página en salir (Ser escrita en el disco para luego ser sobrescrita por otra). Es decir, en un principio se cargarán las paginas especificadas en marcos diferentes. Cuando todos los marcos disponibles estén llenos, se comenzará a reemplazar el primer marco cargado, es decir se escribirá en el disco la página escrita en él en el marco y luego se cargará la nueva página en ese marco y así sucesivamente se ira reemplazando el marco con la página más antigua (la primera en entrar).

El algoritmo random opera cargando las paginas en algún marco al azar. Si es que hay una página ya cargada en el marco, esta será escrita al disco y se cargará desde el disco la nueva página en ese marco.

El algoritmo custom es uno creado por nosotros, este funciona de forma similar al algoritmo random, el número de marco se genera al azar, pero cuando tenemos una página del disco de numero par, el número de marco generado se volverá a generar hasta que este también lo sea. En el caso de que la página del disco sea de número impar, el número de marco generado también tendrá que ser impar.

Para probar las diferencias entre los algoritmos implementados para manejar los errores de páginas, iremos probando los tres algoritmos, en los tres programas. Se mantendrá el número de páginas a cargar como 100 y el número de marcos disponibles se ira variando desde 2 a 100. Luego se agruparán los efectos de los algoritmos y los marcos que se tiene sobre los programas.

Hipótesis:

1.-Siempre comenzamos con nada en la memoria y todo está en el disco. Además, los tres programas cargan la cantidad de páginas especificadas una sola vez, entonces todas las páginas que se quieren cargar causaran error de paginación, es decir que la cantidad de errores de paginación será el mismo número que el de páginas que se quiere cargar. Consideramos que esto debería ocurrir sin importar que algoritmo se utilice.

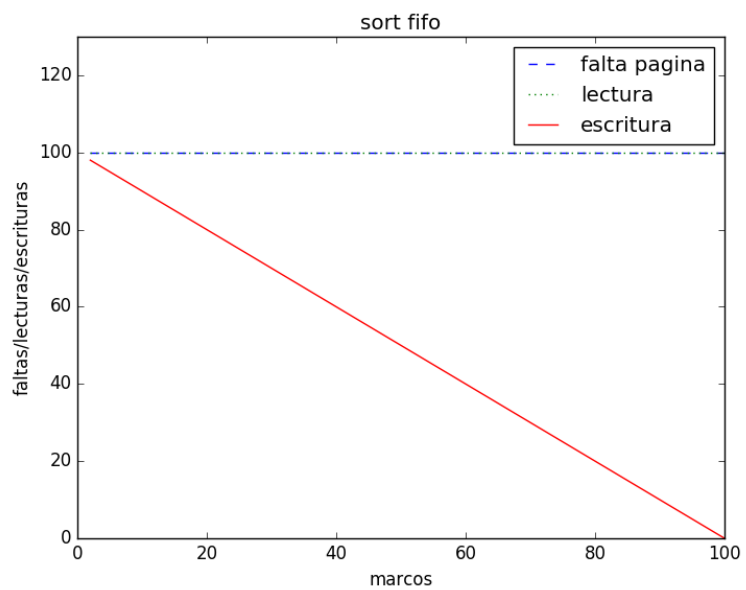
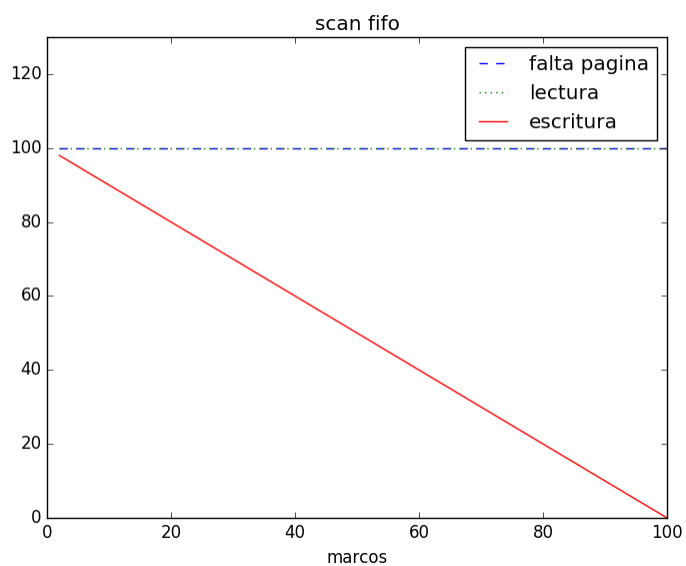
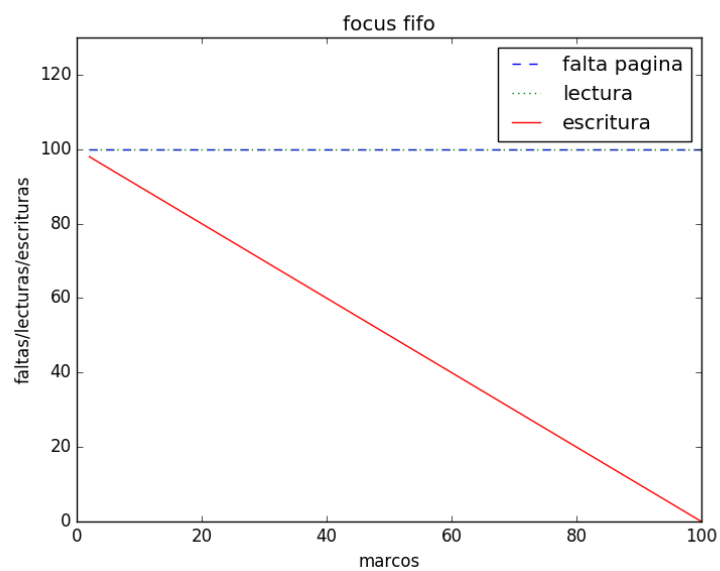
2.-Como consideramos que el número de errores de página es el mismo número de páginas que se quiere cargar, entonces por cada error de página se hará una lectura al disco de la página correspondiente que se quiere cargar. Esto quiere decir que el número de lecturas también es el mismo número que la cantidad de páginas a cargar, sin importar que algoritmo se utiliza.

3.-Para el algoritmo FIFO suponemos que como las páginas serán escritas secuencialmente una por una y luego sobrescritas una por una, quiere decir que al principio se cargarían páginas en marcos libres, luego cuando no queden más, se comenzaran a sobrescribir desde el principio y así sucesivamente. Las escrituras de páginas al disco solo ocurren cuando se sobrescribe un marco, por lo que el número de escrituras debería estar dado por la diferencia del número de páginas a cargar y el número de marcos. Si es que el número de marcos es igual o superior al número de páginas a cargar, el número de escrituras será cero.

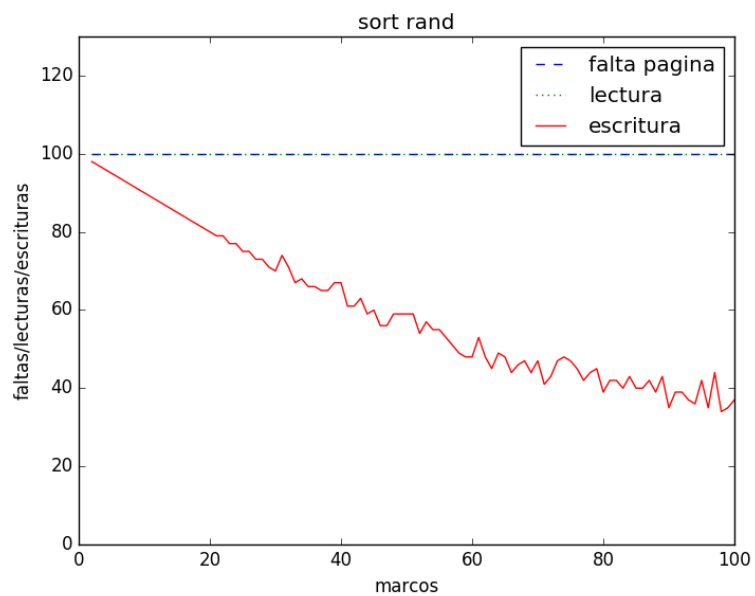
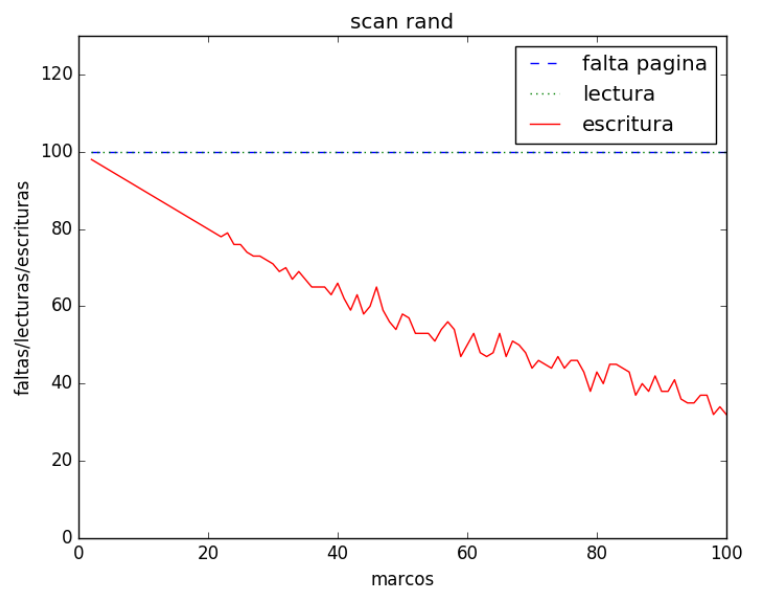
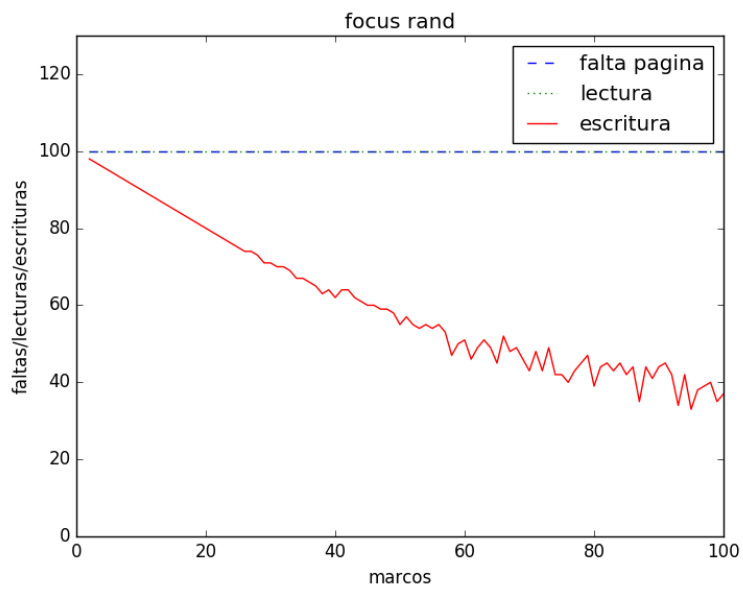
4.-Para el algoritmo Random y Custom, suponemos que como las páginas serán escritas secuencialmente una por una en marcos al azar (o parcialmente en el caso del custom), la cantidad de escrituras al disco dependerá de que justo ocurra que se quiera escribir una página en un marco ya utilizado. Al principio como todos los marcos no están utilizados, no puede haber sobrescritura, pero todos los siguientes errores de paginación pueden ir a parar en el mismo marco, entonces el número de escrituras debería ser un número al azar que va desde la cantidad de páginas a cargar e iría decreciendo si se aumenta la cantidad de marcos. Es decir, como se escogen al azar, hay menos probabilidad de sobrescribir muchas veces el mismo marco, por lo que entre aumenta el número de marcos, disminuye la probabilidad de sobrescribir marcos, es decir, disminuye la probabilidad de escrituras al disco.

Resultados:

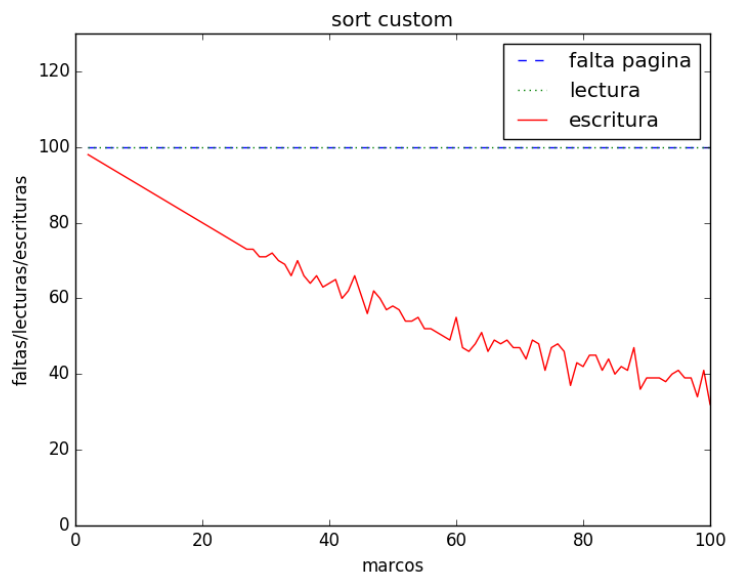
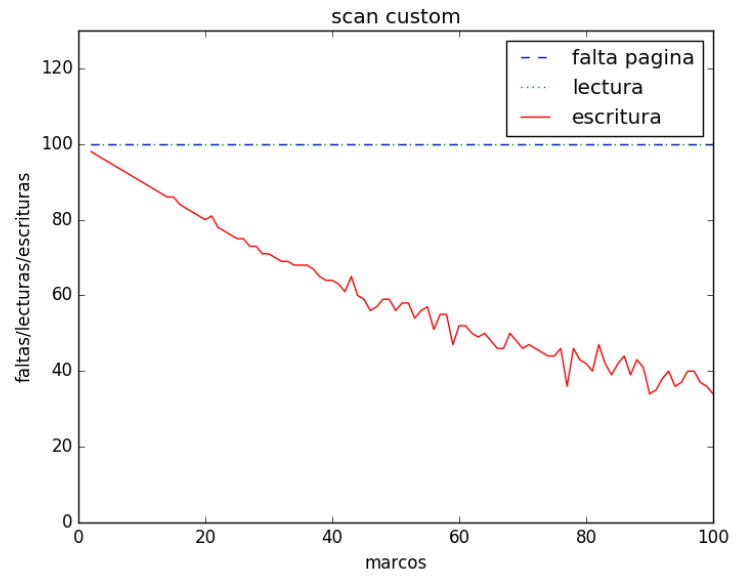
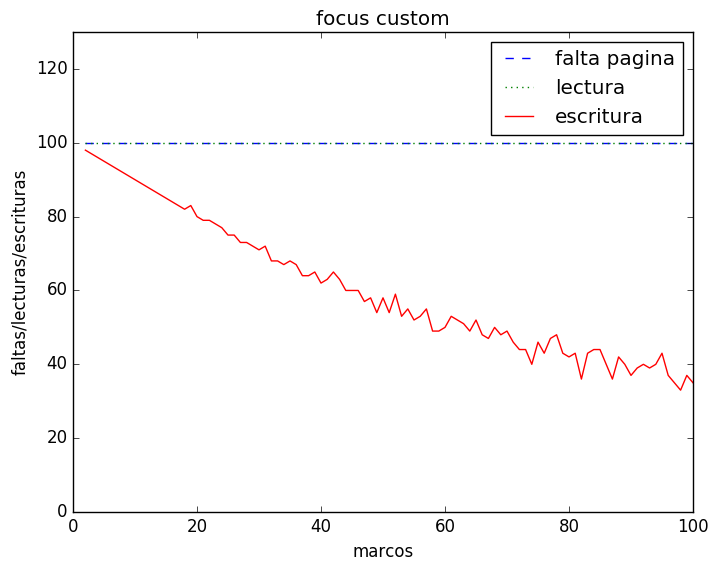
Gráficos de algoritmo FIFO:



Gráficos de algoritmo Random:



Gráficos de algoritmo Custom:



Conclusión:

Con nuestros resultados, podemos observar que el número de errores de paginación y de lecturas, siempre se mantuvo como el número de páginas a cargar, esto prueba nuestras dos primeras hipótesis como correctas. Luego, con los gráficos de los tres programas cuando se utilizó el algoritmo FIFO, podemos ver el número de escrituras fue la diferencia entre el número de páginas a cargar y el número de marcos disponibles, lo que también confirma nuestra tercera hipótesis. Además, con los gráficos de los tres programas cuando se utilizó el algoritmo Random y Custom, el número de escrituras estuvo entre el número de páginas a escribir y fue bajando a medida que se tuvieran más marcos disponibles, confirmando nuestra última hipótesis.

Finalmente, tomando en cuenta todos los gráficos, podemos ver que los algoritmos Random y Custom causaron menos escrituras en el disco que el algoritmo FIFO, por lo que los hace más eficientes en ese sentido.