Memoria Virtual

Caso de Estudio 2 - Infraestructura Computacional

Juan Andrés Romero - 202013449

Juan Alegría - 202011282

Estructuras de datos usadas para simular el comportamiento del sistema de paginación

- ConcurrentHashMap: Esta estructura simula la tabla de páginas de la memoria, en esta se guardan las páginas con sus estados y atributos (Carga en memoria, Bits de referencia y Modificación).

  El hashmap se actualiza cada vez que se hace referencia o modificación a una página, dependiendo de la secuencia dada por el proceso. La actualización consiste en obtener la página del hashmap, actualizarla con base en la secuencia y volverla a poner en el mapa. Sucede lo mismo cuando el algoritmo de reemplazo carga o descarga páginas de la RAM y cuando son reiniciadas por la clase ReinicioPaginas (Reinicio por ciclos de reloj cada 20ms).

- ArrayList: Esta lista representa la memoria RAM real donde se cargan las páginas.

  Se actualiza cuando se cargan o descargan páginas de la memoria real. Esto consiste en añadir o borrar elementos de la ArrayList con base en la carga de páginas o la decisión del algoritmo de reemplazo.

- Página: Creamos una nueva clase que contiene los datos de la página:

  -Número de página. (int)

  -Estado de carga. (Boolean)

  -Bits R y M. (Boolean)

Se actualiza dependiendo de la secuencia de referencias del proceso o de la decisión del algoritmo de reemplazo. Solo se modifica el valor booleano de los bits o el estado de carga.

Esquema de sincronización usado

Se utilizaron métodos sincronizados con exclusión mutua para evitar que dos threads modifiquen la misma estructura de datos al mismo tiempo. De igual manera, se implementó un ConcurrentHashMap, el cual tiene la característica de que todos sus métodos básicos (get, put, replace, etc) son de tipo de exclusión mutua a nivel de par llave, valor. Con lo cual múltiples threads pueden utilizar el HashMap al mismo tiempo, pero solo se excluyen mutuamente si ambos están utilizando el mismo elemento (par llave, valor).

Datos recopilados

Caso Simulado	**Número de fallas de página**
referencias8_16_75.txt	32
referencias5_16.txt	65
referencias20_16.txt	14
referencias8_32_75.txt	97
referencias5_32.txt	251

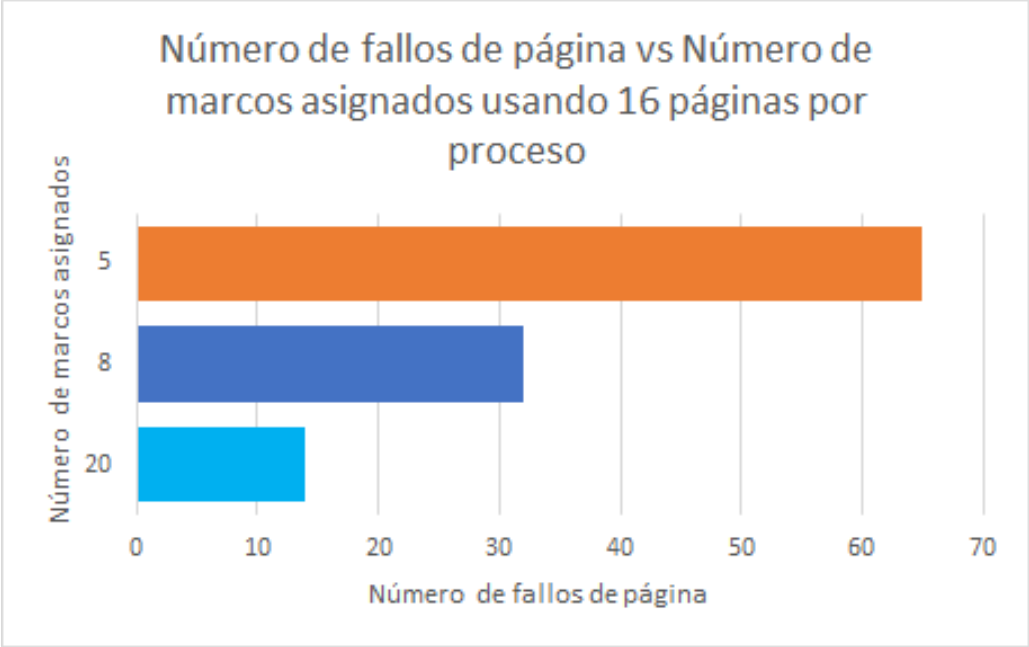
referencias20_32.txt	42
referencias8_64_75.txt	278
referencias5_64_75.txt	359
referencias20_64_75.txt	86
referencias8_128_75.txt	405
referencias5_128_75.txt	437
referencias20_128_75.txt	260

Razón de las variaciones entre los resultados de mismas ejecuciones del programa

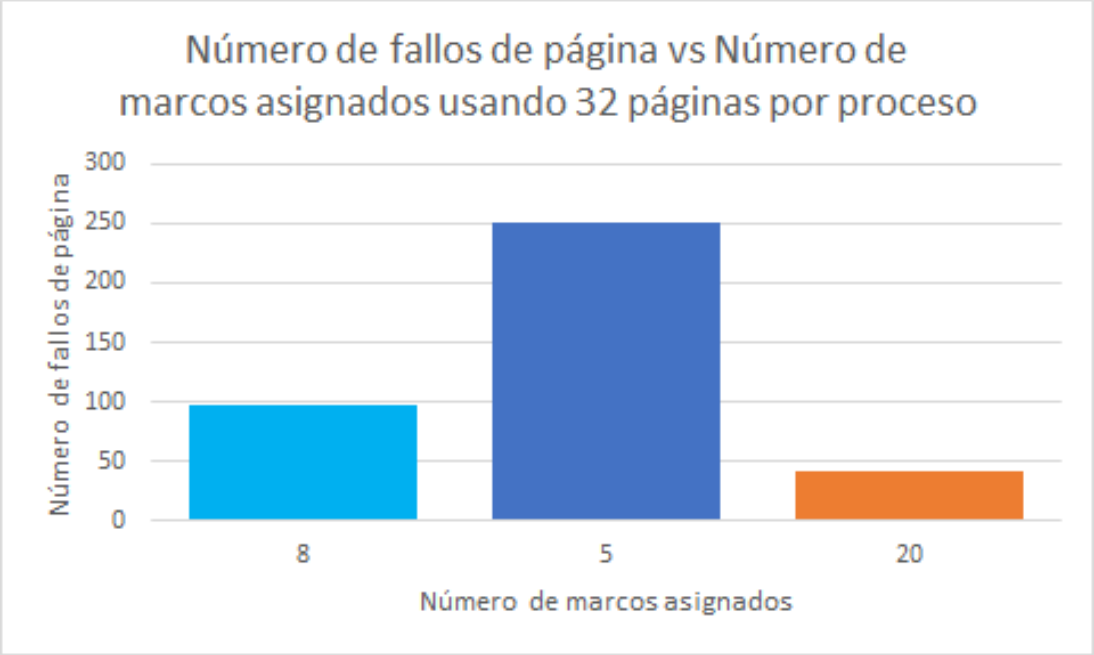
El número de fallas contabilizadas varía entre las ejecuciones de un programa debido a la concurrencia y las variaciones en milisegundos de la ejecución de estas operaciones. Además, el acceso a las fuentes de datos, tanto virtuales como reales, causa pequeñas diferencias entre la ejecución y el tiempo de espera que tienen los threads durante su ejecución.

Gráficas del comportamiento del sistema por cada tamaño de programa

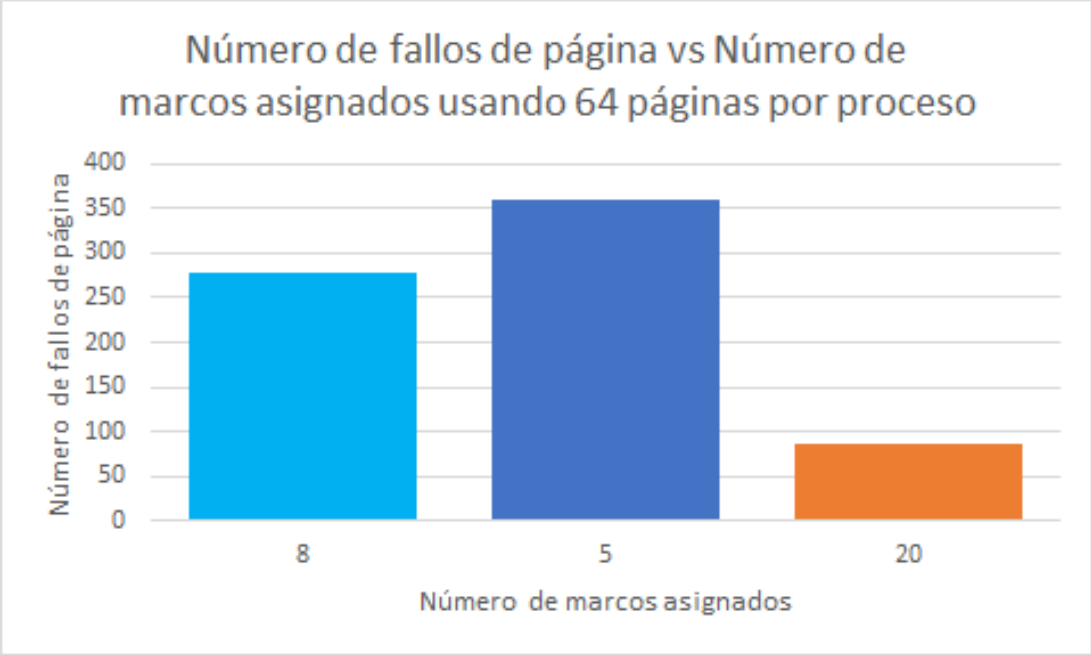
16 páginas por proceso:



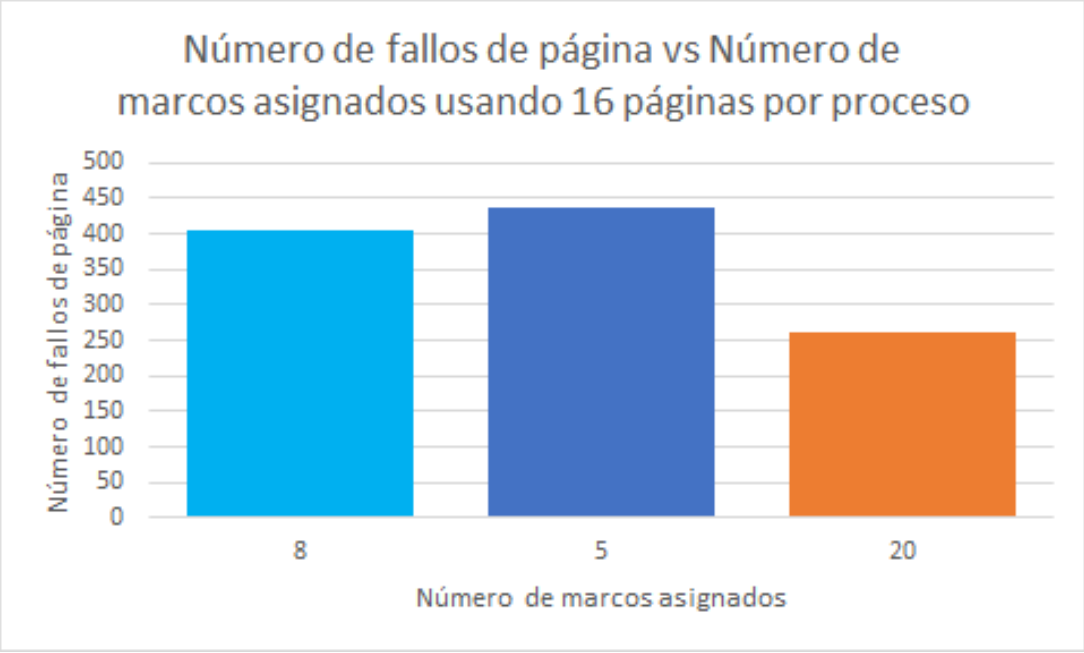
32 páginas por proceso:



64 páginas por proceso:



128 páginas por proceso:



Conclusiones e interpretaciones

Los resultados obtenidos tienen sentido, y se pueden apreciar dos patrones. El primero sucede cuando se tiene un número de páginas del proceso fijo, pero cambia la cantidad de marcos asignados. En este patrón entre más marcos pueda utilizar el proceso, menos fallos de página sucederán, mientras que si este número es menor, muchos más fallos ocurrirán. El segundo patrón comprende un número de páginas del proceso variable, mientras que la cantidad de marcos asignados es fija. Para este caso, entre más número de páginas sean utilizadas (referenciadas o modificadas), mayor será el número de fallos de página y viceversa.