William Mendez – 202012662

**Caso 2 InfraComp**

1. ***Descripción del algoritmo usado en la opción 1***

Para generar el archivo se siguen los siguientes pasos:

* Se piden los valores de tp, te, nf, nc y tr al usuario.
* Se calculan nr y np.

Texto

Descripción generada automáticamente

* Se escriben los datos anteriores en el archivo.

Texto

Descripción generada automáticamente

* Se crean 3 matrices de String vacías para A,B y C. Además, se generan contadores para la matriz, la fila y la columna actual.

Texto

Descripción generada automáticamente

* Se recorre por cada página de la memoria (determinada por np) y por cada entero que cabe en una página y se poblan las 3 matrices con un String de la forma <Letra de la matriz>:[<filaActual>.<columnaActual>],<paginaEnMemoria>,<desplazamientoEnMemoria>\n. Esto teniendo en cuenta que en cada recorrido los contadores deben subir y mantener la coherencia de su valor (si la columna actual es igual al número de columnas esta se reinicia y se le suma uno al contador de filas, y lo mismo con la cantidad de filas y la matriz actual)

Texto

Descripción generada automáticamente

* Luego se escribe según el tipo de recorrido en el archivo por cada fila i y columna j de las matrices el String almacenado en las matrices A,B y C en las posiciones i,j.

Texto

Descripción generada automáticamente

1. ***Descripción de las estructuras usadas para simular el comportamiento del sistema de paginación***

En la opción al leer el archivo su información se almacena en un HashMap memoriaFisica que representa justamente a la memoria física de la forma:

Texto

Descripción generada automáticamente

Donde la llave en el número de página virtual y el valor es un entero de 31 bits que representa el uso de la página con respecto al tiempo (No se usan 32 bits porque en la futura comparación para encontrar el mínimo se interpreta a cualquier entero de 32 bits como negativo). Además, durante la ejecución se asegura que la cantidad de entradas de memoriaFisica sea menor o igual a la cantidad de marcos de página de la ejecución para asegurar que cada entrada represente una página que estaría presente en la memoria física del computador.

Además, para realizar el envejecimiento de las páginas se usa un thread aparte cuya vida se limita a el tiempo en el que se realiza algún recorrido y su método run se encuentra en el siguiente punto.

1. ***Esquema de sincronización usado***

En la implementación planteada el recurso crítico y con el que se comunican los dos procesos es el HashMap memoriaFisica, por lo que en cada modificación necesaria se genera un monitor sobre este recurso. Las ocasiones donde se opera esta memoria son:

* Petición de un dato en memoria física para asignar 1 a la lectura y si es necesario realizar el swap.

Texto, Escala de tiempo

Descripción generada automáticamente

* Actualización del envejecimiento cada 1ms, como la función del thread requiere que este modifique la memoria física sincronizamos este proceso con un monitor en la memoria.

Texto

Descripción generada automáticamente

1. ***Tablas de datos***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 4x4, TE=1, Recorrido 1 | | | | |
|  | 4 | 8 | 16 | 32 |
| MP=36 | 12 | 6 | 3 | 3 |
| MP=18 | 12 | 6 | 3 | 3 |
| MP=06 | 12 | 6 | 3 | 3 |
| MP=03 | 12 | 6 | 3 | 3 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 4x4, TE=1, Recorrido 2 | | | | |
|  | 4 | 8 | 16 | 32 |
| MP=36 | 12 | 6 | 3 | 2 |
| MP=18 | 12 | 6 | 3 | 2 |
| MP=06 | 48 | 6 | 3 | 2 |
| MP=03 | 48 | 24 | 3 | 2 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 4x4, TE=2, Recorrido 1 | | | | |
|  | 4 | 8 | 16 | 32 |
| MP=36 | 24 | 12 | 6 | 3 |
| MP=18 | 24 | 12 | 6 | 3 |
| MP=06 | 24 | 12 | 6 | 3 |
| MP=03 | 24 | 12 | 6 | 3 |

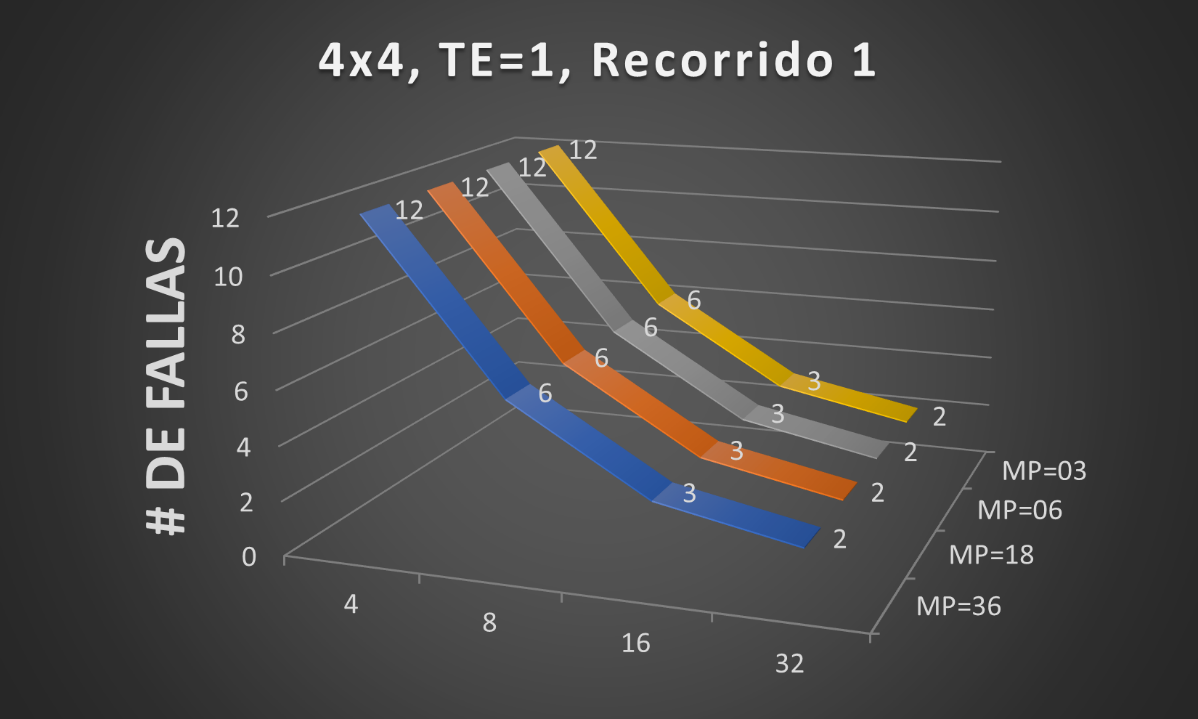
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 4x4, TE=2, Recorrido 2 | | | | |
|  | 4 | 8 | 16 | 32 |
| MP=36 | 24 | 12 | 6 | 3 |
| MP=18 | 24 | 12 | 6 | 3 |
| MP=06 | 48 | 48 | 6 | 3 |
| MP=03 | 48 | 48 | 24 | 3 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 4x4, TE=4, Recorrido 1 | | | | |
|  | 4 | 8 | 16 | 32 |
| MP=36 | 48 | 24 | 12 | 6 |
| MP=18 | 48 | 24 | 12 | 6 |
| MP=06 | 48 | 24 | 12 | 6 |
| MP=03 | 48 | 24 | 12 | 6 |

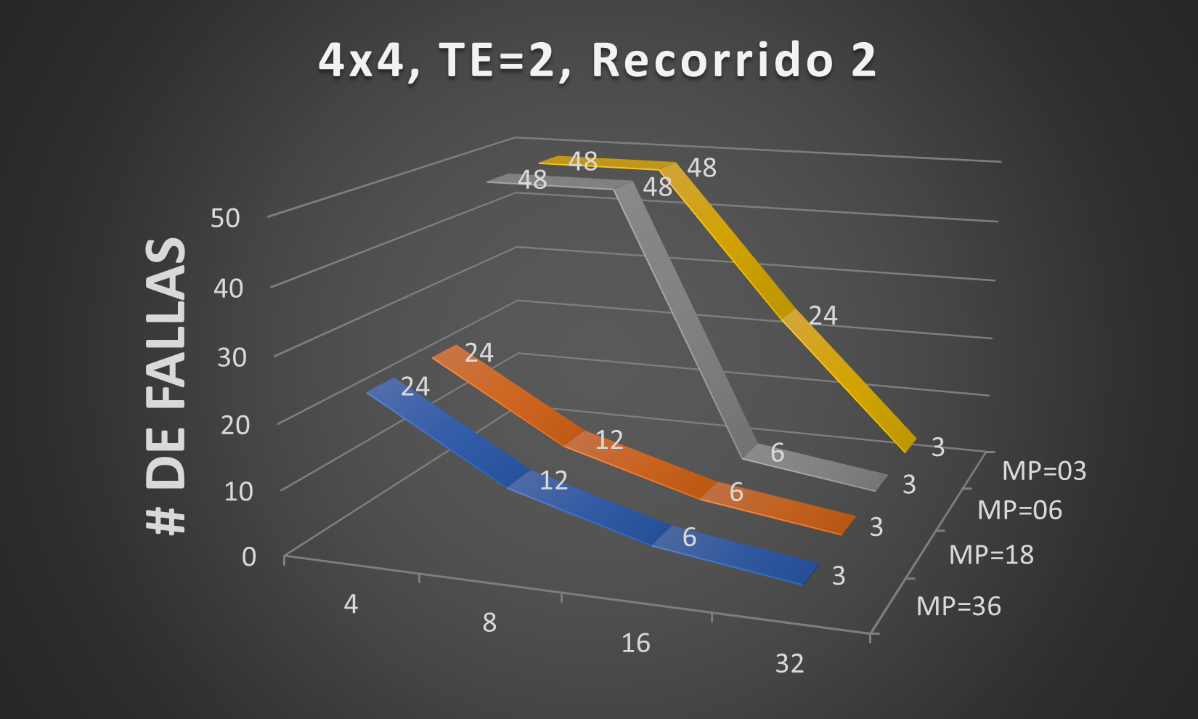
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 16x16, TE=4, Recorrido 1 | | | | | | |
|  | 4 | 8 | 16 | 32 | 64 | 128 |
| MP=36 | 768 | 384 | 195 | 96 | 48 | 24 |
| MP=18 | 768 | 384 | 192 | 96 | 48 | 24 |
| MP=06 | 768 | 384 | 192 | 96 | 48 | 24 |
| MP=03 | 768 | 385 | 192 | 96 | 49 | 25 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 16x16, TE=4, Recorrido 2 | | | | | | |
|  | 4 | 8 | 16 | 32 | 64 | 128 |
| MP=36 | 768 | 696 | 674 | 531 | 433 | 24 |
| MP=18 | 768 | 768 | 768 | 768 | 768 | 235 |
| MP=06 | 768 | 768 | 768 | 768 | 768 | 284 |
| MP=03 | 768 | 768 | 768 | 768 | 768 | 386 |

1. ***Gráficas***

Gráfico

Descripción generada automáticamenteDiagrama

Descripción generada automáticamenteGráfico

Descripción generada automáticamenteGráfico

Descripción generada automáticamente

Interfaz de usuario gráfica

Descripción generada automáticamenteGráfico, Gráfico de líneas

Descripción generada automáticamente

1. ***Interpretación de resultados***

En las gráficas generadas con los datos del programa podemos notar unos valores casi perfectamente constantes en los recorridos de tipo 1 de todos los valores de tamaño de enteros, por lo que podemos concluir que, para lecturas secuenciales, la cantidad de marcos de páginas solo nos afecta en cuanto esta sea menor a la cantidad de matrices que se acceden, ya que si esto se cumple en cada página se reemplazaran de forma secuencial. En los pocos casos en los que notamos unos valores que se alejan de los valores constantes en el recorrido 1 por una o dos fallas podemos concluir que se hace visible el no determinismo del uso de este algoritmo que puede generar malas predicciones de la mejor página para reemplazar.

Además, se puede notar que el crecimiento de las fallas en este recorrido al tener un tamaño de página muy pequeño muestra que para que el algoritmo cumpla su propósito es mejor que las páginas tengan un mayor tamaño independientemente de cuantas son, pues el desperdicio de memoria que esto pueda conllevar se repone con el poco tiempo que tardaría si no debe operar un fallo de página con cada lectura.

Para el recorrido 2, una lectura no secuencial de los datos vemos que la cantidad de fallas que se generan con el algoritmo para tamaños de páginas muy pequeñas es igual a las del recorrido 1 sin embargo no notamos una gran mejoría en el número de fallas hasta que no aumentamos el tamaño de una página a cantidades enormes o aumentamos la cantidad de marcos de página a para que todos los datos necesarios quepan en la RAM.

De todo lo anterior podemos concluir que a nivel de una RAM verdadera que maneja este algoritmo sería mejor aumentar la cantidad de marcos de página, pues en un contexto real con muchos procesos ejecutándose de forma simultánea la cantidad querríamos que cada proceso tenga un marco de página para almacenar las páginas virtuales que necesite.