**Informe Caso 2 Infraestructura computacional**

**Integrantes:**

* **Samuel Josué Freire Tarazona 202111460**
* **Juan Felipe Garcia 202014961**
* **Juan Sebastián Pinzón 202013144**

**Uso:**

Para usar el programa tiene que correr la clase llamada “MainTest”. Luego de correr esto, se le va a solicitar el nombre del archivo de entrada. Este archivo debe estar guardado en la carpeta data que viene en el proyecto de java. Porfavor escribir el tipo del archivo, es decir, escribir el “.txt” al final del nombre del archivo. Luego, se le va a pdeir que escriba 1, si quiere solo el modo 1, o 2 si quiere el modo 2. Luego si selecciono solo el modo 1. La respuesta se va aguardar en el archivo, de la carpeta data llamado: “Modo1.txt”. Por otro lado, si selecciono el Modo 2. Se le va apreguntar si quiere utilizar el archivo que genero del modo 1. Cabe aclarar que, si selecciona el modo 1, se va a realizar sí o sí. Sin embargo, en este punto se le da la opción de escoger si quiere utilizar el archivo que genero el modo 1. En dado, caso escribir “Modo1.txt”. O, por otro lado, utilizar otro archivo. Caso en el cual, requiere escribir el archivo que quiera y al final poner el “.txt”. La respuesta se genera en un print, que dice el número de fallos de página en consola.

**Informe:**

1. **Algoritmo referencia de paginas**
   1. Descripción del algoritmo usado para generar las referencias de página (modo uno)

Para desarrollar, este apartado, es necesario entender antes cual era el contexto del problema. En este caso, se solicitaba un programa el cual emulara la ejecución de un programa o pedazo de código de acuerdo a las partes vistas en clase. Ahora bien, sabiendo esto, podemos empezar a detallar la solución de generación de referencias. En primer lugar, se necesita un archivo de entrada, el cual cuenta con los detalles de las matrices las cuales se van a necesitar para sumar. En este, pedazo, lo que realizamos es la lectura línea por línea del archivo. En este punto, cabe aclarar cuales partes cuenta el archivo de entrada. Dentro de estas partes, están el número de fila de las matrices, el numero de columnas, el tamaño de página, el tamaño de entero y el numero de marcos. Estas propiedades el sistema serán necesarias para la siguiente parte. Luego, de haber aclarado esto vamos a detallar la referencia. Ahora, sabiendo ya el tamaño de las matrices, el cual surge de multiplicar el numero de filas con el numero de columnas. Vamos a realizar un ciclo, el cual nos asegure que va a rellenar o guardar, en “memoria virtual”, las tres matrices. Este ciclo va a seguir tanto se completen las páginas. Ahora dentro de este ciclo, existen otros dos anidados, los cuales se van a “recorrer” las matrices. Aquí dentro, se recorrer cada fila completa de las tres matrices. Cada elemento se va a una página “virtual”, es decir, una lista en nuestra solución. Ahora, si se llena la página esta se guarda en otra lista de página, o la “memoria virtual”, que resulta ser una lista de listas. Ahora no solo se guarda si se termina una página, sino también cuando se alcanza el total de elementos de las tras matrices. Cuando se termina la página se reinicia el ciclo, es decir, se quiere rellenar otra página por decir así. Luego, también, otro evento importante es cuando se llena una fila se reinicia el conteo de filas. Además, de cada que se termina todas las filas de una matriz se aumenta un contador de matrices y se reinicia el contador de filas. Cabe aclarar que al realizar este almacenamiento en idea de row-major order. Siempre se busca que se recorra cada fila de izquierda a derecha. Lo cual es fundamental en nuestra operación. Cabe aclarar, que la información que está en cada página, es decir, la lista, son los elemento s de la matriz que más adelante se referencian. Además, de esto cada que se guarda un elemento en la lista de listas, o memoria virtual, se guarda su referencia en un diccionario, cuyas llaves son el elemento de la matriz, debidamente diferenciado, y su valor es la posición de la página, es decir, el numero de página virtual, además de su posición dentro de la matriz o el desplazamiento de <acuerdo al tamaño del entero. Luego de haber realizado esto, digamos que ya tenemos las tres matrices en memoria virtual. Ahora el siguiente paso, es traer las referencias a estas matrices. Para esto se vuelve a realizar row-major order. Esto para recorrer cada matriz de izquierda a derecha de arriba abajo. Esto se realiza de manera en un doble ciclo de igual manera y cada elemento es llamado por las tres matrices para saber su ubicación o referencia. Esto usando el diccionario de referencias, detallado anteriormente, así que se llama a cada elemento y se guarda su junto a su ubicación en memoria virtual. Dando esto como resultado el archivo de referencias en el modo 1. El cual en nuestra solución se guarda en la carpeta data, en el archivo llamado “Modo1.txt”.

* 1. Descripción de las estructuras de datos usadas para simular el comportamiento del sistema de paginación y cómo usa dichas estructuras (cuándo se actualizan, con base en qué y en qué consiste la actualización).

Se usaron 3 estructuras, un Array que contiene las páginas, otro array con índices correspondientes a cada una de las páginas que contiene varios arráis de 8 enteros que simulan los 8 bits usados en el algoritmo de envejecimiento. Finalmente se usó un mapa cuya función es aquella de una tabla de páginas guardando referencias de cada elemento de la matriz a su página correspondiente. El array usado para el algoritmo de envejecimiento se actualiza cada milisegundo según los accesos a las páginas. En el momento de un fallo de página el thread (Thread 1) encargado consulta el array mencionado anteriormente para buscar el elemento menor, con este reiniciaría el contador y remplazaría la página de su mismo índice con la requerida. El array usado en el algoritmo de envejecimiento al ser actualizado corre todos los “Bits” a la derecha borrando el “Bit” que está ubicado a la derecha del todo, luego si fue accedido entre milisegundos el “Bit” más izquierdo se convierte en 1. Se utiliza otro Array en el cual se guardan los accesos a cada página 1 para acceso y 0 para no acceso, este se reinicia a 0 cada milisegundo.

* 1. Esquema de sincronización usado. Justifique brevemente dónde es necesario usar sincronización y por qué

Ahora hablando del sistema de sincronización. Se trataron 2 esquemas de sincronización. El primero de ellos para lograr solucionar el encuentro, o que los dos Threads no entren al mismo tiempo. Es decir, el T1, que se encarga del sistema de paginación, no entre al mismo tiempo que el T2, es decir, el que se encarga del sistema de envejecimiento. Luego, también se trató otro sistema de sincronización para lograr solucionar el señalamiento, es decir, procurar el orden de ejecución de los threads en el sistema. En este caso, la idea es que el T2 se ejecute mínimo una vez antes del T1. Luego, de haber definido los dos sistemas de sincronización que se usaron, vamos a detallar como fue su implementación o nuestra solución. En primer lugar, el sistema de encuentro fue implementado en la clase Frames. La cual como se ha dicho antes, es la que tiene todo el segundo modo en si misma si se quiere pensar. En este punto se implementó la idea de synchronized. La cual permite meter una estructura de encuentro a lo largo de esta estructura. Esta característica natural de java fue puesta a lo largo de varios métodos que necesitábamos que estuviera bloqueados entre sí para no dañar su concurrencia. En este caso, se aplicó a varios métodos lo cuales, por ejemplo, se encontraban modificando la tabla de páginas que en nuestro caso se llama la lista ‘Framespag’. Uno de ellos es el método insertpage, la cual se encarga de insertar la página en alguno de los marcos de página o páginas en ejecución. Otro método es el reiniciar contador, que es el que se encarga de manejar el reinicio de contado o de envejecer los contadores de acuerdo con el algoritmo. En este punto, era necesario que no se pudieran acceder a los dos métodos al mismo tiempo. Ya que, en el primer método modificamos el contador del ciclo de cada marco de página, y en el segundo necesitamos saber a cuál marco de página se accedió en ese ciclo. Por lo que era necesario que no se accediera al mismo tiempo de los dos métodos. Por último, otor método que tenía esta característica, era ‘inPagina’, el cual busca verificar si la página virtual ya se encuentra en alguno de los marcos de página. Cabe aclarar que esta sincronización al manejarse en el mismo objeto es decir la entidad Frames, todos los métodos sincronizados se bloquean entre si lo que era necesario para que no hubiera errores de información. Por ejemplo, si necesitara ver si una página fue accedida, no pueda estar acecido a una página al mismo tiempo, necesitaría idealmente acceder en otro momento en el que no esté modificando esta estructura de páginas. Luego, el otro método usado para resolver el señalamiento fue Wait(), notify(). Este método desincronización fue usado realmente muy poco en la implementación. Pero a pesar, de esto, es la sincronización más importante de todo el código. Este método fue implementado dentro de dos métodos. El primer de ellos es el método ‘insertPage’. En este método, se implementó un wait, luego de verificar que no existan marcos de página disponibles, con un condicional, era necesario reemplazar una de las ya existentes. Sin embargo, antes de verificar cual estaba disponible por el algoritmo de envejecimiento, se dormía, o se le hacia un wait. Esto generaba que el thread o el proceso se durmiera hasta que el otro thread no lo levantar de cierta manera. Ahora, sabiendo esto el notify fue implementado en el método ‘reinicarContador’. En este método luego de realizar el algoritmo de envejecimiento, se realiza el notify, para levantar al otro thread. De esta manera, nos asegurábamos de que el t2 se ejecutara mínimo una vez antes de alguna ejecución del T1. Esto generaba que cuando el T1 necesite reemplazar alguna página ya ocupada, el marco que escoja por el algoritmo sea correcto y no sea erróneo entre ejecución. Por lo tanto, los sistemas de sincronización usado fueron wait(), notify() y synchronized; resolviendo su respectivo problema de concurrencia.

* 1. Tabla de datos

Casos bases con 4 marcos probados 20 veces por configuración:

Se llevo a cabo una serie de pruebas por configuración en los cuales se usaron los tamaños de matriz básicos, es decir: 8x8, ,5x3,16x16, 13x7,32x32,64x64 y 80x50. El número de marcos fue el mismo para todos el cual fue 4 y la variación que se llevó a cabo fue en los tamaños de página en los cuales se usaron los tamaños: 100,256,350, 512,750, 1024,1500 y 2048. Los resultados se ven en la columna de numero de fallos la cual hace referencia a los fallos en esa configuración especifica. Cabe resaltar que cada configuración fue probada entre 20 y 30 veces para asegurar los resultados.

Caso = Te = 4; MP = 4 (fijos)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  |  |  |
| TP | TE(Tamaño Entero) | NF | NC | MP(Marcos Pagina) | Numero Fallos |
| 100 | 4 | 8 | 8 | 4 | 10 |
| 256 | 4 | 8 | 8 | 4 | 3 |
| 350 | 4 | 8 | 8 | 4 | 3 |
| 512 | 4 | 8 | 8 | 4 | 2 |
| 750 | 4 | 8 | 8 | 4 | 2 |
| 1024 | 4 | 8 | 8 | 4 | 1 |
| 1500 | 4 | 8 | 8 | 4 | 1 |
| 2048 | 4 | 8 | 8 | 4 | 1 |
| 100 | 4 | 5 | 3 | 4 | 2 |
| 256 | 4 | 5 | 3 | 4 | 1 |
| 350 | 4 | 5 | 3 | 4 | 1 |
| 512 | 4 | 5 | 3 | 4 | 1 |
| 750 | 4 | 5 | 3 | 4 | 1 |
| 1024 | 4 | 5 | 3 | 4 | 1 |
| 1500 | 4 | 5 | 3 | 4 | 1 |
| 2048 | 4 | 5 | 3 | 4 | 1 |
| 100 | 4 | 16 | 16 | 4 | 33 |
| 256 | 4 | 16 | 16 | 4 | 12 |
| 350 | 4 | 16 | 16 | 4 | 11 |
| 512 | 4 | 16 | 16 | 4 | 6 |
| 750 | 4 | 16 | 16 | 4 | 5 |
| 1024 | 4 | 16 | 16 | 4 | 3 |
| 1500 | 4 | 16 | 16 | 4 | 3 |
| 2048 | 4 | 16 | 16 | 4 | 2 |
| 100 | 4 | 13 | 7 | 4 | 13 |
| 256 | 4 | 13 | 7 | 4 | 5 |
| 350 | 4 | 13 | 7 | 4 | 4 |
| 512 | 4 | 13 | 7 | 4 | 3 |
| 750 | 4 | 13 | 7 | 4 | 2 |
| 1024 | 4 | 13 | 7 | 4 | 2 |
| 1500 | 4 | 13 | 7 | 4 | 1 |
| 2048 | 4 | 13 | 7 | 4 | 1 |
| 100 | 4 | 32 | 32 | 4 | 125 |
| 256 | 4 | 32 | 32 | 4 | 48 |
| 350 | 4 | 32 | 32 | 4 | 38 |
| 512 | 4 | 32 | 32 | 4 | 24 |
| 750 | 4 | 32 | 32 | 4 | 19 |
| 1024 | 4 | 32 | 32 | 4 | 12 |
| 1500 | 4 | 32 | 32 | 4 | 11 |
| 2048 | 4 | 32 | 32 | 4 | 6 |
| 100 | 4 | 26 | 28 | 4 | 90 |
| 256 | 4 | 26 | 28 | 4 | 37 |
| 350 | 4 | 26 | 28 | 4 | 28 |
| 512 | 4 | 26 | 28 | 4 | 20 |
| 750 | 4 | 26 | 28 | 4 | 14 |
| 1024 | 4 | 26 | 28 | 4 | 11 |
| 1500 | 4 | 26 | 28 | 4 | 8 |
| 2048 | 4 | 26 | 28 | 4 | 5 |
| 100 | 4 | 64 | 64 | 4 | 494 |
| 256 | 4 | 64 | 64 | 4 | 192 |
| 350 | 4 | 64 | 64 | 4 | 144 |
| 512 | 4 | 64 | 64 | 4 | 96 |
| 750 | 4 | 64 | 64 | 4 | 68 |
| 1024 | 4 | 64 | 64 | 4 | 48 |
| 1500 | 4 | 64 | 64 | 4 | 35 |
| 2048 | 4 | 64 | 64 | 4 | 24 |
| 100 | 4 | 80 | 50 | 4 | 480 |
| 256 | 4 | 80 | 50 | 4 | 189 |
| 350 | 4 | 80 | 50 | 4 | 140 |
| 512 | 4 | 80 | 50 | 4 | 96 |
| 750 | 4 | 80 | 50 | 4 | 67 |
| 1024 | 4 | 80 | 50 | 4 | 49 |
| 1500 | 4 | 80 | 50 | 4 | 34 |
| 2048 | 4 | 80 | 50 | 4 | 26 |

Caso = Te = 8; MP = 8 (fijos)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  |  |  |
| TP | TE(Tamaño Entero) | NF | NC | MP(Marcos Pagina) | Numero Fallos |
| 100 | 8 | 8 | 8 | 8 | 8 |
| 256 | 8 | 8 | 8 | 8 | 3 |
| 350 | 8 | 8 | 8 | 8 | 3 |
| 512 | 8 | 8 | 8 | 8 | 2 |
| 750 | 8 | 8 | 8 | 8 | 2 |
| 1024 | 8 | 8 | 8 | 8 | 1 |
| 1500 | 8 | 8 | 8 | 8 | 1 |
| 2048 | 8 | 8 | 8 | 8 | 1 |
| 100 | 8 | 5 | 3 | 8 | 4 |
| 256 | 8 | 5 | 3 | 8 | 2 |
| 350 | 8 | 5 | 3 | 8 | 2 |
| 512 | 8 | 5 | 3 | 8 | 1 |
| 750 | 8 | 5 | 3 | 8 | 1 |
| 1024 | 8 | 5 | 3 | 8 | 1 |
| 1500 | 8 | 5 | 3 | 8 | 1 |
| 2048 | 8 | 5 | 3 | 8 | 1 |
| 100 | 8 | 16 | 16 | 8 | 31 |
| 256 | 8 | 16 | 16 | 8 | 12 |
| 350 | 8 | 16 | 16 | 8 | 9 |
| 512 | 8 | 16 | 16 | 8 | 6 |
| 750 | 8 | 16 | 16 | 8 | 5 |
| 1024 | 8 | 16 | 16 | 8 | 3 |
| 1500 | 8 | 16 | 16 | 8 | 3 |
| 2048 | 8 | 16 | 16 | 8 | 2 |
| 100 | 8 | 13 | 7 | 8 | 23 |
| 256 | 8 | 13 | 7 | 8 | 9 |
| 350 | 8 | 13 | 7 | 8 | 7 |
| 512 | 8 | 13 | 7 | 8 | 5 |
| 750 | 8 | 13 | 7 | 8 | 3 |
| 1024 | 8 | 13 | 7 | 8 | 3 |
| 1500 | 8 | 13 | 7 | 8 | 2 |
| 2048 | 8 | 13 | 7 | 8 | 2 |
| 100 | 8 | 32 | 32 | 8 | 123 |
| 256 | 8 | 32 | 32 | 8 | 48 |
| 350 | 8 | 32 | 32 | 8 | 36 |
| 512 | 8 | 32 | 32 | 8 | 24 |
| 750 | 8 | 32 | 32 | 8 | 17 |
| 1024 | 8 | 32 | 32 | 8 | 12 |
| 1500 | 8 | 32 | 32 | 8 | 9 |
| 2048 | 8 | 32 | 32 | 8 | 6 |
| 100 | 8 | 26 | 28 | 8 | 182 |
| 256 | 8 | 26 | 28 | 8 | 69 |
| 350 | 8 | 26 | 28 | 8 | 51 |
| 512 | 8 | 26 | 28 | 8 | 35 |
| 750 | 8 | 26 | 28 | 8 | 24 |
| 1024 | 8 | 26 | 28 | 8 | 18 |
| 1500 | 8 | 26 | 28 | 8 | 12 |
| 2048 | 8 | 26 | 28 | 8 | 9 |
| 100 | 8 | 64 | 64 | 8 | 492 |
| 256 | 8 | 64 | 64 | 8 | 192 |
| 350 | 8 | 64 | 64 | 8 | 142 |
| 512 | 8 | 64 | 64 | 8 | 96 |
| 750 | 8 | 64 | 64 | 8 | 66 |
| 1024 | 8 | 64 | 64 | 8 | 48 |
| 1500 | 8 | 64 | 64 | 8 | 33 |
| 2048 | 8 | 64 | 64 | 8 | 24 |
| 100 | 8 | 80 | 50 | 8 | 1000 |
| 256 | 8 | 80 | 50 | 8 | 375 |
| 350 | 8 | 80 | 50 | 8 | 280 |
| 512 | 8 | 80 | 50 | 8 | 188 |
| 750 | 8 | 80 | 50 | 8 | 130 |
| 1024 | 8 | 80 | 50 | 8 | 94 |
| 1500 | 8 | 80 | 50 | 8 | 65 |
| 2048 | 8 | 80 | 50 | 8 | 47 |

Caso = Te = 2; MP = 2 (fijos)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  |  |  |
| TP | TE(Tamaño Entero) | NF | NC | MP(Marcos Pagina) | Numero Fallos |
| 100 | 2 | 8 | 8 | 2 | 192 |
| 256 | 2 | 8 | 8 | 2 | 192 |
| 350 | 2 | 8 | 8 | 2 | 54 |
| 512 | 2 | 8 | 8 | 2 | 2 |
| 750 | 2 | 8 | 8 | 2 | 2 |
| 1024 | 2 | 8 | 8 | 2 | 1 |
| 1500 | 2 | 8 | 8 | 2 | 1 |
| 2048 | 2 | 8 | 8 | 2 | 1 |
| 100 | 2 | 5 | 3 | 2 | 1 |
| 256 | 2 | 5 | 3 | 2 | 1 |
| 350 | 2 | 5 | 3 | 2 | 1 |
| 512 | 2 | 5 | 3 | 2 | 1 |
| 750 | 2 | 5 | 3 | 2 | 1 |
| 1024 | 2 | 5 | 3 | 2 | 1 |
| 1500 | 2 | 5 | 3 | 2 | 1 |
| 2048 | 2 | 5 | 3 | 2 | 1 |
| 100 | 2 | 16 | 16 | 2 | 768 |
| 256 | 2 | 16 | 16 | 2 | 768 |
| 350 | 2 | 16 | 16 | 2 | 768 |
| 512 | 2 | 16 | 16 | 2 | 768 |
| 750 | 2 | 16 | 16 | 2 | 768 |
| 1024 | 2 | 16 | 16 | 2 | 768 |
| 1500 | 2 | 16 | 16 | 2 | 54 |
| 2048 | 2 | 16 | 16 | 2 | 2 |
| 100 | 2 | 13 | 7 | 2 | 273 |
| 256 | 2 | 13 | 7 | 2 | 51 |
| 350 | 2 | 13 | 7 | 2 | 2 |
| 512 | 2 | 13 | 7 | 2 | 2 |
| 750 | 2 | 13 | 7 | 2 | 1 |
| 1024 | 2 | 13 | 7 | 2 | 1 |
| 1500 | 2 | 13 | 7 | 2 | 1 |
| 2048 | 2 | 13 | 7 | 2 | 1 |
| 100 | 2 | 32 | 32 | 2 | 3072 |
| 256 | 2 | 32 | 32 | 2 | 3072 |
| 350 | 2 | 32 | 32 | 2 | 3072 |
| 512 | 2 | 32 | 32 | 2 | 3072 |
| 750 | 2 | 32 | 32 | 2 | 3072 |
| 1024 | 2 | 32 | 32 | 2 | 3072 |
| 1500 | 2 | 32 | 32 | 2 | 3072 |
| 2048 | 2 | 32 | 32 | 2 | 3072 |
| 100 | 2 | 26 | 28 | 2 | 2184 |
| 256 | 2 | 26 | 28 | 2 | 2184 |
| 350 | 2 | 26 | 28 | 2 | 2184 |
| 512 | 2 | 26 | 28 | 2 | 2184 |
| 750 | 2 | 26 | 28 | 2 | 2184 |
| 1024 | 2 | 26 | 28 | 2 | 2184 |
| 1500 | 2 | 26 | 28 | 2 | 2052 |
| 2048 | 2 | 26 | 28 | 2 | 408 |
| 100 | 2 | 64 | 64 | 2 | 12288 |
| 256 | 2 | 64 | 64 | 2 | 12288 |
| 350 | 2 | 64 | 64 | 2 | 12288 |
| 512 | 2 | 64 | 64 | 2 | 12288 |
| 750 | 2 | 64 | 64 | 2 | 12288 |
| 1024 | 2 | 64 | 64 | 2 | 12288 |
| 1500 | 2 | 64 | 64 | 2 | 12288 |
| 2048 | 2 | 64 | 64 | 2 | 12288 |
| 100 | 2 | 80 | 50 | 2 | 12000 |
| 256 | 2 | 80 | 50 | 2 | 12000 |
| 350 | 2 | 80 | 50 | 2 | 12000 |
| 512 | 2 | 80 | 50 | 2 | 12000 |
| 750 | 2 | 80 | 50 | 2 | 12000 |
| 1024 | 2 | 80 | 50 | 2 | 12000 |
| 1500 | 2 | 80 | 50 | 2 | 12000 |
| 2048 | 2 | 80 | 50 | 2 | 12000 |

Caso = Te = 2; MP = 4 (fijos)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  |  |  |
| TP | TE(Tamaño Entero) | NF | NC | MP(Marcos Pagina) | Numero Fallos |
| 100 | 2 | 8 | 8 | 4 | 4 |
| 256 | 2 | 8 | 8 | 4 | 2 |
| 350 | 2 | 8 | 8 | 4 | 2 |
| 512 | 2 | 8 | 8 | 4 | 1 |
| 750 | 2 | 8 | 8 | 4 | 1 |
| 1024 | 2 | 8 | 8 | 4 | 1 |
| 1500 | 2 | 8 | 8 | 4 | 1 |
| 2048 | 2 | 8 | 8 | 4 | 1 |
| 100 | 2 | 5 | 3 | 4 | 1 |
| 256 | 2 | 5 | 3 | 4 | 1 |
| 350 | 2 | 5 | 3 | 4 | 1 |
| 512 | 2 | 5 | 3 | 4 | 1 |
| 750 | 2 | 5 | 3 | 4 | 1 |
| 1024 | 2 | 5 | 3 | 4 | 1 |
| 1500 | 2 | 5 | 3 | 4 | 1 |
| 2048 | 2 | 5 | 3 | 4 | 1 |
| 100 | 2 | 16 | 16 | 4 | 18 |
| 256 | 2 | 16 | 16 | 4 | 6 |
| 350 | 2 | 16 | 16 | 4 | 5 |
| 512 | 2 | 16 | 16 | 4 | 3 |
| 750 | 2 | 16 | 16 | 4 | 3 |
| 1024 | 2 | 16 | 16 | 4 | 2 |
| 1500 | 2 | 16 | 16 | 4 | 2 |
| 2048 | 2 | 16 | 16 | 4 | 1 |
| 100 | 2 | 13 | 7 | 4 | 8 |
| 256 | 2 | 13 | 7 | 4 | 3 |
| 350 | 2 | 13 | 7 | 4 | 2 |
| 512 | 2 | 13 | 7 | 4 | 2 |
| 750 | 2 | 13 | 7 | 4 | 1 |
| 1024 | 2 | 13 | 7 | 4 | 1 |
| 1500 | 2 | 13 | 7 | 4 | 1 |
| 2048 | 2 | 13 | 7 | 4 | 1 |
| 100 | 2 | 32 | 32 | 4 | 64 |
| 256 | 2 | 32 | 32 | 4 | 24 |
| 350 | 2 | 32 | 32 | 4 | 20 |
| 512 | 2 | 32 | 32 | 4 | 12 |
| 750 | 2 | 32 | 32 | 4 | 11 |
| 1024 | 2 | 32 | 32 | 4 | 6 |
| 1500 | 2 | 32 | 32 | 4 | 5 |
| 2048 | 2 | 32 | 32 | 4 | 3 |
| 100 | 2 | 26 | 28 | 4 | 46 |
| 256 | 2 | 26 | 28 | 4 | 20 |
| 350 | 2 | 26 | 28 | 4 | 15 |
| 512 | 2 | 26 | 28 | 4 | 11 |
| 750 | 2 | 26 | 28 | 4 | 8 |
| 1024 | 2 | 26 | 28 | 4 | 5 |
| 1500 | 2 | 26 | 28 | 4 | 3 |
| 2048 | 2 | 26 | 28 | 4 | 3 |
| 100 | 2 | 64 | 64 | 4 | 248 |
| 256 | 2 | 64 | 64 | 4 | 96 |
| 350 | 2 | 64 | 64 | 4 | 73 |
| 512 | 2 | 64 | 64 | 4 | 48 |
| 750 | 2 | 64 | 64 | 4 | 35 |
| 1024 | 2 | 64 | 64 | 4 | 24 |
| 1500 | 2 | 64 | 64 | 4 | 19 |
| 2048 | 2 | 64 | 64 | 4 | 12 |
| 100 | 2 | 80 | 50 | 4 | 240 |
| 256 | 2 | 80 | 50 | 4 | 96 |
| 350 | 2 | 80 | 50 | 4 | 71 |
| 512 | 2 | 80 | 50 | 4 | 49 |
| 750 | 2 | 80 | 50 | 4 | 34 |
| 1024 | 2 | 80 | 50 | 4 | 26 |
| 1500 | 2 | 80 | 50 | 4 | 18 |
| 2048 | 2 | 80 | 50 | 4 | 14 |

Caso = Te = 8; MP = 4 (fijos)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | |  |  |  |  |
| TP | TE(Tamaño Entero) | NF | NC | MP(Marcos Pagina) | Numero Fallos |
| 100 | 8 | 8 | 8 | 4 | 18 |
| 256 | 8 | 8 | 8 | 4 | 6 |
| 350 | 8 | 8 | 8 | 4 | 5 |
| 512 | 8 | 8 | 8 | 4 | 3 |
| 750 | 8 | 8 | 8 | 4 | 3 |
| 1024 | 8 | 8 | 8 | 4 | 2 |
| 1500 | 8 | 8 | 8 | 4 | 2 |
| 2048 | 8 | 8 | 8 | 4 | 1 |
| 100 | 8 | 5 | 3 | 4 | 4 |
| 256 | 8 | 5 | 3 | 4 | 2 |
| 350 | 8 | 5 | 3 | 4 | 2 |
| 512 | 8 | 5 | 3 | 4 | 1 |
| 750 | 8 | 5 | 3 | 4 | 1 |
| 1024 | 8 | 5 | 3 | 4 | 1 |
| 1500 | 8 | 5 | 3 | 4 | 1 |
| 2048 | 8 | 5 | 3 | 4 | 1 |
| 100 | 8 | 16 | 16 | 4 | 66 |
| 256 | 8 | 16 | 16 | 4 | 24 |
| 350 | 8 | 16 | 16 | 4 | 20 |
| 512 | 8 | 16 | 16 | 4 | 12 |
| 750 | 8 | 16 | 16 | 4 | 11 |
| 1024 | 8 | 16 | 16 | 4 | 6 |
| 1500 | 8 | 16 | 16 | 4 | 5 |
| 2048 | 8 | 16 | 16 | 4 | 3 |
| 100 | 8 | 13 | 7 | 4 | 25 |
| 256 | 8 | 13 | 7 | 4 | 11 |
| 350 | 8 | 13 | 7 | 4 | 9 |
| 512 | 8 | 13 | 7 | 4 | 5 |
| 750 | 8 | 13 | 7 | 4 | 3 |
| 1024 | 8 | 13 | 7 | 4 | 3 |
| 1500 | 8 | 13 | 7 | 4 | 2 |
| 2048 | 8 | 13 | 7 | 4 | 2 |
| 100 | 8 | 32 | 32 | 4 | 258 |
| 256 | 8 | 32 | 32 | 4 | 96 |
| 350 | 8 | 32 | 32 | 4 | 74 |
| 512 | 8 | 32 | 32 | 4 | 48 |
| 750 | 8 | 32 | 32 | 4 | 36 |
| 1024 | 8 | 32 | 32 | 4 | 24 |
| 1500 | 8 | 32 | 32 | 4 | 19 |
| 2048 | 8 | 32 | 32 | 4 | 12 |
| 100 | 8 | 26 | 28 | 4 | 184 |
| 256 | 8 | 26 | 28 | 4 | 71 |
| 350 | 8 | 26 | 28 | 4 | 53 |
| 512 | 8 | 26 | 28 | 4 | 37 |
| 750 | 8 | 26 | 28 | 4 | 26 |
| 1024 | 8 | 26 | 28 | 4 | 20 |
| 1500 | 8 | 26 | 28 | 4 | 14 |
| 2048 | 8 | 26 | 28 | 4 | 11 |
| 100 | 8 | 64 | 64 | 4 | 1026 |
| 256 | 8 | 64 | 64 | 4 | 384 |
| 350 | 8 | 64 | 64 | 4 | 288 |
| 512 | 8 | 64 | 64 | 4 | 192 |
| 750 | 8 | 64 | 64 | 4 | 135 |
| 1024 | 8 | 64 | 64 | 4 | 96 |
| 1500 | 8 | 64 | 64 | 4 | 68 |
| 2048 | 8 | 64 | 64 | 4 | 48 |
| 100 | 8 | 80 | 50 | 4 | 1002 |
| 256 | 8 | 80 | 50 | 4 | 375 |
| 350 | 8 | 80 | 50 | 4 | 282 |
| 512 | 8 | 80 | 50 | 4 | 189 |
| 750 | 8 | 80 | 50 | 4 | 132 |
| 1024 | 8 | 80 | 50 | 4 | 96 |
| 1500 | 8 | 80 | 50 | 4 | 67 |
| 2048 | 8 | 80 | 50 | 4 | 49 |

* 1. Gráficas y tablas con configuraciones extra variando tamaño de página y marcos asignados

A continuación, se realizaron más pruebas con distintas configuraciones en este caso se variaron dos factores con respecto al índice anterior, el primer cambio que se realizó fueron la variación de marcos, se usaron 3 distintos números de marcos el primero fueron 4 marcos el cual es el que se usó en el índice anterior, el segundo fueron 8 marcos y por último 2 marcos. El otro cambio que se realizó en estas pruebas fueron nuevos tamaños de página en este caso se agregaron los siguientes tamaños: 100, 350, 750 y 1500 por lo que se duplicaron los tamaños de página en comparación al anterior índice. Aparte de realizar las pruebas con estas nuevas configuraciones se realizaron graficas por resultados estas van asociadas al número de marcos usados y sus ejes son FallosDePagina X TamañoDePagina X TamañoDeMatriz, las pruebas y graficas mencionadas anteriormente se encuentran a continuación:

**4 Marcos y 4 Tamaño de entero:**

Fallos:

Tamaño de la matriz x Tamaño de pagina

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Size** | **8x8** | **5x3** | **16x16** | **13x7** | **32x32** | **26x28** | **64x64** | **80x50** |
| 100 | 10 | 2 | 33 | 13 | 125 | 90 | 494 | 480 |
| 256 | 3 | 1 | 12 | 5 | 48 | 37 | 192 | 189 |
| 350 | 3 | 1 | 11 | 4 | 38 | 28 | 144 | 140 |
| 512 | 2 | 1 | 6 | 3 | 24 | 20 | 96 | 96 |
| 750 | 2 | 1 | 5 | 2 | 19 | 14 | 68 | 67 |
| 1024 | 1 | 1 | 3 | 2 | 12 | 11 | 48 | 49 |
| 1500 | 1 | 1 | 3 | 1 | 11 | 8 | 35 | 34 |
| 2048 | 1 | 1 | 2 | 1 | 6 | 5 | 24 | 26 |

**8 Marcos y 8 Tamaño de entero:**

Fallos:

Tamaño de la matriz x Tamaño de pagina

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Size** | **8x8** | **5x3** | **16x16** | **13x7** | **32x32** | **26x28** | **64x64** | **80x50** |
| 100 | 8 | 4 | 31 | 23 | 123 | 182 | 492 | 1000 |
| 256 | 3 | 2 | 12 | 9 | 48 | 69 | 192 | 375 |
| 350 | 3 | 2 | 9 | 7 | 36 | 51 | 142 | 280 |
| 512 | 2 | 1 | 6 | 5 | 24 | 35 | 96 | 188 |
| 750 | 2 | 1 | 5 | 3 | 17 | 24 | 66 | 130 |
| 1024 | 1 | 1 | 3 | 3 | 12 | 18 | 48 | 94 |
| 1500 | 1 | 1 | 3 | 2 | 9 | 12 | 33 | 65 |
| 2048 | 1 | 1 | 2 | 2 | 6 | 9 | 24 | 47 |

**2 Marcos y 2 Tamaño de entero:**

Fallos:

Tamaño de la matriz x Tamaño de pagina

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Size** | **8x8** | **5x3** | **16x16** | **13x7** | **32x32** | **26x28** | **64x64** | **80x50** |
| 100 | 192 | 1 | 768 | 273 | 3072 | 2184 | 12288 | 12000 |
| 256 | 192 | 1 | 768 | 51 | 3072 | 2184 | 12288 | 12000 |
| 350 | 54 | 1 | 768 | 2 | 3072 | 2184 | 12288 | 12000 |
| 512 | 2 | 1 | 768 | 2 | 3072 | 2184 | 12288 | 12000 |
| 750 | 2 | 1 | 768 | 1 | 3072 | 2184 | 12288 | 12000 |
| 1024 | 1 | 1 | 768 | 1 | 3072 | 2184 | 12288 | 12000 |
| 1500 | 1 | 1 | 54 | 1 | 3072 | 2052 | 12288 | 12000 |
| 2048 | 1 | 1 | 2 | 1 | 3072 | 408 | 12288 | 12000 |

**4 Marcos y 2 Tamaño de entero:**

Fallos:

Tamaño de la matriz x Tamaño de pagina

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Size** | **8x8** | **5x3** | **16x16** | **13x7** | **32x32** | **26x28** | **64x64** | **80x50** |
| 100 | 4 | 1 | 18 | 8 | 64 | 46 | 248 | 240 |
| 256 | 2 | 1 | 6 | 3 | 24 | 20 | 96 | 96 |
| 350 | 2 | 1 | 5 | 2 | 20 | 15 | 73 | 71 |
| 512 | 1 | 1 | 3 | 2 | 12 | 11 | 48 | 49 |
| 750 | 1 | 1 | 3 | 1 | 11 | 8 | 35 | 34 |
| 1024 | 1 | 1 | 2 | 1 | 6 | 5 | 24 | 26 |
| 1500 | 1 | 1 | 2 | 1 | 5 | 3 | 19 | 18 |
| 2048 | 1 | 1 | 1 | 1 | 3 | 3 | 12 | 14 |

**4 Marcos y 8 Tamaño de entero:**

Fallos:

Tamaño de la matriz x Tamaño de pagina

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Size** | **8x8** | **5x3** | **16x16** | **13x7** | **32x32** | **26x28** | **64x64** | **80x50** |
| 100 | 18 | 4 | 66 | 25 | 258 | 184 | 1026 | 1002 |
| 256 | 6 | 2 | 24 | 11 | 96 | 71 | 384 | 375 |
| 350 | 5 | 2 | 20 | 9 | 74 | 53 | 288 | 282 |
| 512 | 3 | 1 | 12 | 5 | 48 | 37 | 192 | 189 |
| 750 | 3 | 1 | 11 | 3 | 36 | 26 | 135 | 132 |
| 1024 | 2 | 1 | 6 | 3 | 24 | 20 | 96 | 96 |
| 1500 | 2 | 1 | 5 | 2 | 19 | 14 | 68 | 67 |
| 2048 | 1 | 1 | 3 | 2 | 12 | 11 | 48 | 49 |

Una vez terminadas las pruebas y examinando las gráficas y resultados se ve que con 2 marcos existen los fallos de página empiezan a abundar desde que el tamaño de la matriz es 16x16, en cambio la que tiene menor número de fallos de página vendría siendo la configuración que tiene 8 marcos, en este caso tiene menor número de fallos, pero la diferencia no es tan amplia con la de 4 marcos, en la mayoría de configuraciones cuando su tamaño de página y su tamaño de matriz son iguales y solo varia la cantidad de marcos la diferencia en fallos de página se diferencia en 2 si es que existe una diferencia entre estos dos.

* 1. ¿Qué pasaría si las matrices fueran almacenadas siguiendo column-major order?

En el caso de que se cambiara la forma de paginar de Row-major order a Column-major order y no se cambiara la forma de recorrer la matriz en el algoritmo, debido a que este recorre de izquierda a derecha pero las páginas están de arriba abajo aumenta la posibilidad de un fallo de página, para entender lo anteriormente mencionado se puede ver un ejemplo: imaginemos una matriz NxN y una página a la cual le caben 2n enteros, al estar la página en columna-major order solo podrá guardar dos columnas lo que implica 2 elementos por fila, a la hora de ejecución cada dos elementos encontrara un fallo de página lo que equivale a n/2 fallos de página por fila o n\*\*2/2 por matriz, a diferencia de row-major order en el cual fallaría después de dos filas lo que fallaría ½ por fila y n/2 por matriz. A partir de lo anteriormente mencionado se ve claramente el cambio que sufriría el algoritmo en caso de que la forma de recorrer la matriz no fuese cambiada, por lo que se puede decir que aumentarían mucho los fallos de pagina y posiblemente el tiempo de ejecución.

* 1. ¿Cómo varía el número de fallas de página si el algoritmo estudiado es el de multiplicación de matrices?

A la hora de hablar de cambiar el algoritmo de estudio por uno de multiplicación, no debería importarnos realmente el numero dentro de las matrices, sino como se va a recorrer y que generar esto junto a las referencias en el número de fallos de página. Antes de definir cuál sería este efecto es necesario recordar que en matemática la multiplicación de matrices es diferente, a la suma. Esto se debe a que para encontrar el primer elemento de la matriz c. En la suma, solo sería sumar los dos primeros elementos de las otras dos matrices. Sin embargo, en la multiplicación necesitarías multiplicar cada elemento de la primera fila de la primera matriz con toda la primera columna de la segunda matriz. Y así sucesivamente hasta rellenar toda la tercera matriz. Ahora sabiendo esto, es necesario hablar de que sucedería en el sistema de paginación. En este punto, necesitaríamos muchísimos más elementos en un mismo momento que el que tenemos actualmente. Esto sucede por lo que se habló antes. Al haber más elementos, realmente esta parte no están importante. En este punto, realmente lo que importa es que de la segunda matriz vamos a buscar es por columnas. Este genera que las referencias que se llaman, en este punto, por lo general van a perecer o existen más probabilidades de que estén en páginas virtuales diferentes. Por lo que lo más probable es que cada vez que queramos acceder a una referencia de la segunda matriz, vamos a necesitar hacer algún reemplazo de página o se va a generar un fallo de página en la mayoría de los casos. Por lo que, al hacer esto en la segunda matriz al estar buscando por columnas va a generar varios fallos de página. Por lo tanto, si el algoritmo es cambiado a multiplicación de matrices, los fallos de página van a aumentar casi por el doble.

* 1. Escriba su interpretación de los resultados: ¿tienen sentido? Justifique su respuesta.

Como se puede ver a lo largo de las pruebas que se realizaron anteriormente, específicamente en la sección de casos de prueba. Podemos evidenciar varias cosas. Todas estas conclusiones tienen que ver con tiempo de ejecución, con el numero de fallos de página. En primer lugar, una de las conclusiones que podemos ver es en cuanto tiempo de ejecución. Esta conclusión es que a medida que se van aumentando los tamaños de las matrices, el tiempo de ejecución del programa va aumentando. Esto quiere decir que entre mas grande sea la matriz mas grande es el tiempo de ejecución. Esto se debe a varios aspectos de la implementación. El mas importante de ellos, es encontrado en el modo 2. Específicamente en el T1, que se encarga de cargar las referencias. Esto se debe a que este thread debe cargar uno a uno las referencias y asignarlas a memoria virtual. Esto sucede ya que, al ser tres matrices grandes, se requiere recorrer un archivo de referencias muy grandes una a una e ir ejecutando funciones relacionadas con el otro thread. Por lo que, en este punto, ya podemos concluir que se demorara mínimo 2 milisegundos por referencia, es decir, que si la matriz es muy grande este proceso va a tomar la mayoría de tiempo de ejecución. De esta manera aumentando caso a caso. Luego, la otra conclusión tiene que ver con el numero de fallos de página. Esto se debe a que este número tiene relación con dos conceptos. En primer lugar, y mas importante, es el número de marcos. Esto se debe a que entre menor sea este numero directamente aumenta el numero de fallos de página. Realmente, el numero de marcos es el que genera que le numero de fallas aumente de manera exponencial. Luego, otra conclusión relacionada al numero de fallas, es que entre mas paginas virtuales se necesiten para almacenar las tres matrices, también va a generar aún más fallos de página a un caso donde estas sean menores. Por lo tanto, las conclusiones principales fueron. En primer lugar, que a menor número de marcos de página mayo numero de fallos de página. Luego, que, a mayor tamaño de matrices, mayor tiempo de ejecución va requerir y, por último, entre más paginas virtuales se necesiten, es decir, el tamaño de página virtual es pequeño, mas fallos de página va a ver también. Además, de esto podemos concluir o definir r que realmente los resultados obtenidos, son correctos con los que se esperaría de cierta manera. Ya que, acorde a lo que se concluyó, era lo que teóricamente también se esperaba. Por lo que por ejemplo lo fallos aumenten a mediad de que las matrices son mas grandes, incluso cuando se mantiene los tamaños de página entre caso. O que los fallos disminuyan en cuanto el tamaño se disminuye y viceversa. Son resultados que se esperan. Esto se debe a que, entre mas referencias halla mas paginas virtuales va a ver que asignar o reemplazar. O si los tamaños de pagina son menores menos elementos caben en cada pagina se van a necesitar mas páginas. Por lo tanto, cada resultado va acorde a lo explicado anteriormente, dado los efetos que generan los casos admitidos.