MEMORIA PRACTICA 1 DE TÉCNICAS DE OPTIMIZACIÓN

Ejercicio 1. Realizar una implementación del programa de multiplicación de 2 matrices de dimensiones 2048x2048

Se ha realizado la implementación del programa mediante código C++.

El código a implementar es el siguiente:

```
for(int i=0; i < 2048; i++){
        for(int j=0; j < 2048; j++){
                for(int k = 0; k < 2048; k++){
                        C[i][j] = C[i][j] + A[i][k] * B[k][j]
                }
        }
}
```

Ejercicio 2. Caracterizar el equipo donde se ejecutará el programa

Las características del equipo empleado son las siguientes:

PROCESADOR	Intel Core i7-8565U
ENTORNO DE DESARROLLO	Ubuntu 20.04.3
HERRAMIENTA DE DESARROLLO	C++
COMPILADOR	G++ 9.3.0
MEMORIA CACHÉ L1	
DATA	4x32 KB
LINE SIZE	64 B
ASSOCIATIVITY	8-way set
NUMERO DE LINEAS POR VIA	$\frac{32\,KB}{8} = 4\,\frac{Kb}{via} \rightarrow \frac{4\,\frac{KB}{via}}{64\,\frac{B}{via}} = 64\,\frac{lineas}{via}$
MEMORIA CACHÉ L2	
DATA	4x256 KB
LINE SIZE	64 B
ASSOCIATIVITY	8-way set
NUMERO DE LINEAS POR VIA	$\frac{256\,KB}{8} = \frac{32\,KB}{via} \rightarrow \frac{32\,\frac{KB}{via}}{64\,\frac{B}{via}} = 5\,\frac{lineas}{via}$
MEMORIA CACHÉ L3	
DATA	8 MB
LINE SIZE	64 B
ASSOCIATIVITY	16-way set

NUMERO DE LINEAS POR VIA	$\frac{8\,MB}{16} = 500\,\frac{KB}{via} \rightarrow \frac{500\,\frac{KB}{via}}{64\,\frac{B}{via}} = 800\,\frac{lineas}{via}$

Ejercicio 3. Medir los tiempos del bucle de ejecución de las dos matrices, inicialmente con vector de iteraciones (i,j,k)

Se ha tardado un tiempo de ejecución de 124’533 ms

Ejercicio 4. Medir los tiempos de ejecución con vector de iteraciones (i,k,j)

Se ha tardado un tiempo de ejecución de 24’187.5 ms

Ejercicio 5. Realizar comparativa de tiempos para la compilación con las opciones -O0 y -O2. Identifica la opción por defecto e indica vuestras conclusiones generales sobre el compilador

	-O0	-O2
Vector (i,j,k)	124’616 ms	56’763.1 ms
Vector (i,k,j)	24’402.3 ms	5’579.78 ms

Por defecto, se ejecuta la opción -O0.

Sobre el compilador, al ejecutar con la opción -O0 no se aplica una optimización, hecho que cambia para la opción -O2. Esto demuestra que el tiempo de ejecución de la opción -O2 sea inferior al tiempo de ejecución de la opción -O0.

Ejercicio 6. Si se obtienen tiempos distintos con ambos vectores de iteraciones, realizar una justificación exhaustiva del resultado explicando a tu parecer lo que está sucediendo internamente en el procesador y la jerarquía de memoria

Como ya sabemos, la memoria caché trabaja cargando los datos secuenciales que cree que vamos a utilizar, y en caso de que esos datos no sean usados o la CPU reclame algún dato que no esté cargado en memoria caché, debe acceder a memoria principal para obtenerlo, proceso que ocupa mucho más tiempo.

Con los tiempos obtenidos, podemos observar cómo en el caso del orden (i,k,j) se tarda mucho menos tiempo que con el orden (i,j,k). Esto se debe a que con (i,k,j) se hacen menos accesos a memoria principal pues los accesos a las celdas de las matrices corresponden al orden establecido de índices, de manera que la memoria caché ya tiene cargado el vector entero que reclamarán los bucles a la hora de recorrer las matrices, produciéndose muchos menos fallos de caché.