MEMORIA PRÁCTICA 4 DE TÉCNICAS DE OPTIMIZACIÓN

Apartado a. Medir los tiempos de ejecución del algoritmo

El algoritmo a implementar es el siguiente:

```
for (int i=0; i < 4224; i++){
        for (int j=0; j < 4224; j++){
            A[j][i] = A[j][i] + B[i][j] + C[j][i]*D[i][j];
        }
    }
```

Opción -O0	Opción -O2
417.418 ms	189.103 ms

Apartado b. Implementar “tiling” o “blocking” sobre el código anterior, para uno y dos niveles de memoria caché. Medir tiempo de ejecución del bucle

El algoritmo a implementar es el siguiente:

```
for(int ii=0; ii < 4224; ii = ii+16){
        for(int jj=0; jj < 4224; jj = jj+8){
            for(int i = ii; i < ii + 16; i++){
                for(int j = jj; j < jj + 8; j++){
                    A[j][i] = A[j][i] + B[i][j] + C[j][i] * D[i][j];
                }
            }
        }
    }
```

Opción -O0	Opción -O2
133.937 ms	50.847 ms

Apartado c. Para los resultados obtenidos en los casos de ejecución anteriores, realizar la comparación de tiempos de ejecución con las distintas opciones de optimización en la compilación (defecto, O0 y O2)

Si observamos los tiempos de cómputo, comprobamos que al aplicar la opción -O2 se produce un tiempo inferior al obtenido con la opción -O0. Esto es debido a que la opción -O2 aplica una optimización; la opción -O0 no aplica dicha optimización.

Apartado d. Justificar resultados, aportando características hardware del equipo donde se han realizado las pruebas

Si comparamos los tiempos de cómputo del algoritmo original con los obtenidos al implementar “tiling”, comprobamos que el tiempo de cómputo aplicando tiling es menor a los obtenidos aplicando el algoritmo original. Esto es debido a que la técnica de optimización “tiling” divide la memoria caché en bloques y almacena la información en dichos bloques, de modo que utiliza cada bloque hasta que no sea necesario el uso de dicho bloque.

Las características hardware del equipo empleado son las siguientes:

Procesador del equipo	Intel Core i7-8565U
Entorno de desarrollo	Ubuntu 20.04.3
Herramienta de desarrollo	C++
Compilador empleado	G++ version 9.3.0

Características de la memoria caché L1

data	4x32 KB (4 bloques de 32 KB)
line size	64 B
Associativity	8-way set

Numero de palabras por bloque: 16
Calculamos el número de líneas por vía:

$$\frac{32\,KB}{8} = \frac{4\,KB}{via} \rightarrow \frac{\frac{4\,KB}{via}}{\frac{64\,B}{via}} = \frac{64\,lineas}{via}$$