

Práctica 1

La biblioteca se llama “mitiempo”, implementa un cronómetro para medir el tiempo de ejecución de funciones en nuestro programa (y el tiempo de ejecución total), todas las funciones devuelven un int y en caso de error devuelven -1, el tiempo esta medido en microsegundos. Se compila con la biblioteca dinámica por defecto (make), para compilar con la biblioteca estática hay que usar “make static”. La biblioteca dinámica se llama “libmitiempo_S.so” y la estática “libmitiempo_s.a”.

El correcto uso de esta biblioteca sería:

Ejecutar primero la función init() que lee del archivo “config.txt” y se establece que reloj del sistema usará.

A continuación se debe usar la función start() para que empiece la cuenta global del cronómetro, la cuenta global no parara hasta que se llame a stop().

La función pause() devuelve el tiempo transcurrido desde la invocación de un start()/resume() y para la cuenta global pero está no finaliza.

La función resume() continua la cuenta de tiempo global, que se había pausado previamente con pause().

Por último la función stop() para la cuenta global definitivamente y devuelve su valor.

La aplicación de ejemplo de use realiza una multiplicación de matrices en la que estableces tanto el alto y ancho de las mismas, las matrices operando se rellenan con números aleatorios comprendidos en el intervalo [0-100].

Argumentos de la aplicación:

- H: establece el alto de la matriz resultante(C).
- W: establece el ancho de la matriz resultante(C).
- w: establece el ancho(A) y alto(B) de las matrices operando.
- v: muestra las matrices.

Ejemplo: ./mulMatrixDynamic.elf -H 1024 -W 1024 -w 512 -v

El archivo salida.txt se puede utilizar para volcar la salida de la aplicación.

Ejemplo: `./mulMatrixDynamic.elf -H 1024 -W 1024 -w 512 -v > salida.txt`

Comparación de tiempos optimizados:

El tamaño de las matrices es de A:1024x512 B:512x1024 y C:1024x1024, se compilado con la biblioteca dinámica.

-o0:

Elapsed time to process the multiplication: 4069645 us

-o1:

Elapsed time to process the multiplication: 3441401 us

-o2:

Elapsed time to process the multiplication: 3475660 us

-o3:

Elapsed time to process the multiplication: 3455206 us

Observamos que la optimización en líneas generales con mejores resultados es la de -o1, el resto (-o2, -o3) no mejora significativamente los tiempos en comparación con -o1.