

# Arquitectura y organización de computadoras.

## IA-32 PRACTICA-3



Alberto Amigo Alonso.(T2)

Sergio Delgado Álvarez.

## 1.Introducción:

Durante las distintas pruebas de ejecución de nuestro programa realizadas sobre distintas máquinas hemos obtenido diferentes resultados sobre el tiempo que tardan en calcularse las DFT en rutinas escritas en lenguaje ANSI C y en ensamblador IA32 con la extensión SSE. Ésta última nos proporciona ocho registros de 128 bits sobre los que podemos escribir dos datos en punto flotante de precisión doble con un mecanismo de operaciones de alto rendimiento, lo que resulta especialmente útil para representar números complejos. Hemos podido comprobar que estos resultados pueden llegar a variar mucho en función de la máquina en la que ejecutamos el mismo código, llegando desde el orden de 900 ms en los computadores habilitados en los laboratorios a 100 ms en el personal, verificando así los conocimientos aprendidos en la asignatura sobre la importancia que pueden llegar a tener aspectos como el diseño de la CPU o las políticas con que se implementan las memorias.

## 2.Resultados:

Ejemplo:

En más detalle y a modo de ejemplo, los resultados exactos obtenidos en una de las simulaciones han sido:

**Parte C:** 262.945 ms

**Parte IA32:** 205.349 ms

lo que nos conduce a la conclusión esperada de que una función escrita en ANSI C se ejecuta más lentamente que otra con la misma funcionalidad implementada en ensamblador con un buen algoritmo diseñado específicamente para resolver ese problema.

## 3.Explicación de los resultados:

Llegados a este punto, nos podría surgir la siguiente cuestión:

¿Es realmente tanto más rápido como esperábamos?

La respuesta más natural es "no", pues nos imaginaríamos el programa compilado en ANSI C como un conjunto de instrucciones generadas sin mejora de implementación alguna y con multitud de cargas y almacenamientos en memoria para salvar la gran cantidad de datos que se producirían por generar 1024 números complejos que no podrían estar permanentemente en los registros EXX.

Buscando respuestas ante esta cuestión hemos desensamblado la función "dftc" con la opción "disas" del depurador GDB y hemos podido comprobar que GCC, el compilador utilizado, incluye en sus últimas versiones implementación para la extensión SSE y hace uso en la medida de lo posible de los registros XMM que ésta pone a disposición del programador (como se muestra en la imagen).

```
movsd (%rax),%xmm3
movsd %xmm3,-0x20(%rbp)
cvtsi2sdl -0x8(%rbp),%xmm0
movsd 0x410(%rip),%xmm1
mulsd %xmm0,%xmm1
cvtsi2sdl -0x4(%rbp),%xmm0
mulsd %xmm1,%xmm0
callq 0x400660 <cos@plt>
mulsd -0x20(%rbp),%xmm0
movsd %xmm0,-0x20(%rbp)
mov -0x4(%rbp),%eax
cldq
```

#### 4. Complemento:

Para maximizar el aprendizaje, también hemos aprovechado esta práctica para comprobar la importante mejora que supone el mecanismo estudiado en las clases teóricas del desenrollado de bucles que incluye el "nivel de refinamiento" de compilación -O1 de GCC, obteniendo un tiempo de ejecución del cálculo con el algoritmo en ANSI C de 186.335 ms, siendo aproximadamente un 70% más rápido.

NOTA: El cálculo del tiempo que tardan en ejecutarse los algoritmos está probado con rutinas que realizan las mismas operaciones (calcular cada dft con datos del array val\_complex y almacenar los resultados en res\_complex), por lo que son bien comparables.