COMPILADORES E INTÉRPRETES

Generación y optimización de código

Práctica 1:

1. Las medidas de tiempo de ejecución se harán utilizando la función *gettimeofday()*.
   a. Lee el manual de dicha función.
   b. Si queremos medir el tiempo en una parte del código haremos lo siguiente:

```
#include <sys/time.h>
…
struct timeval inicio, final;
double tiempo;
…
gettimeofday(&inicio,NULL);
            {…} /*Código a medir*/
gettimeofday(&final,NULL);
tiempo = (final.tv_sec-inicio.tv_sec+(final.tv_usec-inicio.tv_usec)/1.e6);
```

   c. Comprueba la influencia del tiempo empleado por la propia función *gettimeofday*.
   d. Mide el tiempo de ejecución con dos ejemplos simples: uno con muchas operaciones aritméticas, y otro con muchas operaciones de entrada/salida. Haz varias medidas.

2. Usaremos el compilador gcc.
   a. Lee el manual de dicho compilador.
   b. El ensamblador que genera es el de la arquitectura IA32 de Intel. Su manual está en: http://www.intel.com/Assets/PDF/manual/253665.pdf. Hojéalo. Tienes información interesante en el siguiente resumen: https://www.cs.umd.edu/users/meesh/webpages/cmsc311/links/handouts/ia32.pdf y en la siguiente chuleta: https://www.cs.swarthmore.edu/~kwebb/cs31/f18/IA32_Cheat_Sheet.pdf
   c. Considera el siguiente código ineficiente que realiza el producto de dos matrices cuadradas. Comprueba que funciona correctamente.

```
#include <stdio.h>
#define Nmax 500

void producto(float x, float y, float *z) {
     *z=x*y; }

main() {
float A[Nmax][Nmax], B[Nmax][Nmax], C[Nmax][Nmax], t, r;
int i,j,k;

for(i=0;i<Nmax;i++)         /* Valores de las matrices */
     for(j=0;j<Nmax;j++) {
            A[i][j]=(i+j)/(j+1.1);
            B[i][j]=(i-j)/(j+2.1); }
for(j=0;j<Nmax;j++)         /* Producto matricial */
     for(i=0;i<Nmax;i++) {
            t=0;
            for (k=0;k<Nmax;k++) {
                  producto(A[i][k],B[k][j],&r);
                  t+=r; }
            C[i][j]=t; }  }
```

d. Comprueba la generación de este código usando el proyecto Compiler Explorer: https://godbolt.org/
e. La opción –E realiza solamente el preprocesado. Comprueba cómo se sustituyen las constantes.
f. La opción –S genera el código en ensamblador. Compruéba la sintáxis de este ensamblador, como se implementan los lazos, como se hacen las llamadas a funciones y como son las operaciones en punto flotante.
g. La opción –c genera el código objeto. Compruébalo. Un fichero objeto se puede enlazar con gcc incluyendo el fichero .o directamente en la línea comando.
h. El enlazado estático se puede hacer con la opción –static de gcc. Los ficheros compilados de esta manera son autónomos al quedar incorporadas a su código las librerías. Comprueba el tamaño del ejecutable.
i. Compíla con las opciones -O0, -O1, -O2, -O3, -Os. Compara el tamaño de los códigos objeto de cada compilación. Compara los tiempos de ejecución. Compara los códigos en ensamblador. Sube un breve informe en pdf solamente sobre este apartado en el entregable del CV.

3. Considera la operación de suavizado de una imágen en gris de tamaño NxM con una vecindad de 5x5. Esta operación consiste en aplicar el promedio de los píxeles vecinos de acuerdo con una ventada de tamaño 5x5 centrada en cada uno de los píxeles de la imágen.

$$I'(i,j) = \frac{1}{25} \sum_{m=-2}^{2} \sum_{n=-2}^{2} I(i+m, j+n)$$

donde *I(i,j)* es el valor del píxel original en la posición (*i*,*j*), *I'(i,j)* es el valor suavizado de dicho píxel. El rango *m* y *n* va de *-2* a *+2*, lo que cubre la vecindad de *5x5* alrededor del píxel (*i*,*j*), y *25* es el número total de píxeles en esa vecindad. Para los píxeles de las fronteras se considerarán solamente los píxeles de la vecindad que estén dentro de la imagen y se sustiuirá el *25* por el número de vecinos en cada caso. El código en C tendrá 4 lazos anidados que recorran los índices *i*, *j*, *m* y *n*. Busca el orden de anidamiento más eficiente de estos índices en términos de tiempo de ejecución de las 24 permutaciones posibles. Hazlo para compilaciones con la opción -O1 y con la optimización **–O1 –funroll-loops**. Compara los códigos en ensamblador obtenidos en dichas compilaciones. Analiza los tiempos de ejecución de ambas versiones para diferentes tamaños de la imágen. Sube un breve informe al CV en formato pdf.