

Practica 1 – Eficiencia

Ejercicio 7

Características del computador usado:

- Procesador: Intel(R) Core(TM) i7-2600 CPU @ 3.40GHz
- Ram: 8 GB
- SO: Ubuntu 16.04 (Linux version 4.4.0-96-generic)
- Compilador: gcc version 5.4.0 20160609 (Ubuntu 5.4.0-6ubuntu1~16.04.4)

NOTA: En la compilacion de las pruebas realizadas no se han utilizado flags de optimización ni de otro tipo.

Algoritmo

/*

A con k filas y m columnas,

B con m filas y n columnas.

C es la matriz que resulta del producto AB.

C presenta k filas y n columnas.

$$C[i][j] = A[i][0] * B[0][j] + A[i][1] * B[1][j] + A[i][2] * B[2][j] + \dots + A[i][m-1] * B[m-1][j]$$
*/

#include <iostream>

#include <iostream>

#include <ctime> // Recursos para medir tiempos

#include <cstdlib> // Para generación de números pseudoaleatorios

using namespace std;

```
void calcular(int a[500][500],int b[500][500],int c[500][500],int k,int n, int m){
    // Generamos la matriz C.
    for(int i=0; i<k; ++i)
        for(int j=0; j<n; ++j)
            for(int z=0; z<m; ++z)
                c[i][j] += a[i][z] * b[z][j];
}
```

```
void imprimir(int a[500][500],int b[500][500],int c[500][500],int k,int n, int m){
    cout<<"\nMatriz A: "<<endl;
    for(int i=0; i<k; ++i)
```

Angel Barrilao Benshrir

```
{
    for(int j=0; j<m; ++j)
    {
        cout<<a[i][j]<<" ";
    }
    cout<<endl;
}

cout<<"\nMatriz B: "<<endl;
for(int i=0; i<m; ++i)
{
    for(int j=0; j<n; ++j)
    {
        cout<<b[i][j]<<" ";
    }
    cout<<endl;
}

cout<<"\nMatriz C: "<<endl;
for(int i=0; i<k; ++i)
{
    for(int j=0; j<n; ++j)
    {
        cout<<c[i][j]<<" ";
    }
    cout<<endl;
}
cout<<endl<<endl;
}

int main(int argc, char * argv[])
{
    srand((int)time(0));

    int A[500][500], B[500][500], C[500][500];
    int k, m, n;
    k=atoi(argv[1]);
    m=atoi(argv[2]);
    n=atoi(argv[3]);

    /*cout<<"Lectura de matriz A."<<endl;
    cout<<"Filas de A: "; cin>>k;
```

Angel Barrilao Benshrir

```
cout<<"Columnas de A: "; cin>>m;
cout<<endl;*/
for(int i=0; i<k; ++i)
    for(int j=0; j<m; ++j)
    {
        A[i][j]=((rand() % 100) +1);
    }

/*cout<<"Lectura de matriz B."<<endl;
cout<<"Filas de B: "<<m<<" (Tiene que ser el mismo que en la matriz A)."<<endl;
cout<<"Columnas de B: "; cin>>n;
cout<<endl;*/
for(int i=0; i<m; ++i)
    for(int j=0; j<n; ++j)
    {
        B[i][j]=((rand() % 100) +1);
    }

//Inicializamos la matriz a cero
    for(int i=0; i<k; ++i)
        for(int j=0; j<n; ++j)
        {
            C[i][j]=0;
        }

clock_t tini; // Anotamos el tiempo de inicio
    tini=clock();
        calcular(A,B,C,k,n,m);
    clock_t tfin; // Anotamos el tiempo de finalización
    tfin=clock();

//imprimir(A,B,C,k,n,m);

// Mostramos resultados
    cout << k*n << "\t" << (tfin-tini)/(double)CLOCKS_PER_SEC << endl;
}
```

Script

```
#!/bin/csh
@ inicio = 2
@ fin = 30000
@ incremento = 2
set ejecutable = mult_matrices
set salida = tiempos_mult_matrices.dat

@ i = $inicio
echo > $salida
while ( $i <= $fin )
    echo Ejecución tam = $i
    echo `.{ $ejecutable } $i $i $i` >> $salida
    @ i += $incremento
end
```

Eficiencia

Teórica

$$\sum_{i=0}^k \left[\sum_{j=0}^n \left[\sum_{z=0}^m 5 \right] \right]$$

Empírica

$$f(x) = a \cdot x \cdot x \cdot x + b$$

