Tarea 5

1 Include & Static

El uso de `#include <file>` es para llamar a un header del sistema. Es decir, los busca en la lista estándar del sistema.

El uso de `#include "file"` es para llamar a un header creado por nosotros. Los busca en la misma carpeta donde está el archivo a compilar.

Las variables del tipo `static` tienen la propiedad de preservar su valor, aunque estén fuera del scope (función, archivo, etc.) de donde fueron declaradas y no pueden ser inicializadas después de su primer inicialización.

2 Investigación

2.1 Basic Linear Algebra Subprograms

BLAS provee rutinas que facilitan operaciones entre vectores y matrices. El **nivel 1** hace operaciones escalares y vectoriales. El **nivel 2** hace operaciones entre vectores y matrices. El **nivel 3** hace operaciones entre matrices. Estas operaciones usan datos de punto flotante o números complejos. Usan paralelización de las rutinas.

2.2 CBLAS

CBLAS es la implementación de BLAS en C.

2.3 Linear Algebra Package

LAPACK tiene rutinas para resolver sistemas de ecuaciones lineales, mínimos cuadrados de un sistema lineal, problemas de eigenvalores, problemas de descomposición de valores singulares y factorizaciones de matrices (LU, Cholesky, QR, SVD, Schur). El objetivo de LAPACK es ser una librería eficiente para el uso de multi-core, esto lo logran a través de llamadas a BLAS, particularmente, al tercer nivel de BLAS.

2.4 Automatically Tuned Linear Algebra Software

ATLAS es un proyecto en aplicar técnias empíricas en orden para dar un desempeño portable, principalmente, de BLAS y un poco de LAPACK.

3 Código

El código se encuentra en la carpeta y se llama `daxpy.c`.

```
v1 <- sample(1:40)[1:3]
cat(v1, file = "v1.txt")
v2 <- sample(1:40)[1:3]
cat(v2, file = "v2.txt")
```

Los vectores son: 36, 19, 33 y 2, 27, 20

```
gcc -Wall dot_product.c funciones.c -o programa.out -lblas
./programa.out 3
```

```
## ------------
## vector :
## vector[0]=36.00000
## vector[1]=19.00000
## vector[2]=33.00000
## ------------
## vector :
## vector[0]=2.00000
## vector[1]=27.00000
## vector[2]=20.00000
## ------------
## resultado: 1245.000000
```

4 ddot

Funciona para calcular el producto punto de dos vectores. Sus parametros:

- **N**: es un entero que dice el número de elementos de un vector.
- **DX** y **DY**: arreglos a multiplicar.
- **INCX** y **INCY**: tamaño del espacio entre los elementos de los arreglos.

5 daxpy

```
gcc -Wall daxpy.c funciones.c -o programa1.out -lblas
./programa1.out 3
```

```
## ------------
## vector :
## vector[0]=36.00000
## vector[1]=19.00000
## vector[2]=33.00000
## ------------
## vector :
## vector[0]=2.00000
## vector[1]=27.00000
## vector[2]=20.00000
## ------------
## vector[0]=34.40000
```

```
## vector[1]=44.10000
## vector[2]=49.70000
```