

1. Estudia y entiende los archivos definiciones.h y funciones.c de ahí, en particular investiga por qué se usan "" en la línea que tiene #include en funciones.c en lugar de usar < >. Investiga el uso de static en la definición de variables externas de funciones.c .

Definiciones.h: define el espacio para hacer un vector/matriz.

Funciones.c: rellena el arreglo/matriz creado en la función anterior, también imprime sus entradas.

"" implica que va a buscar el programa en el *working directory* actual, justo en donde estamos "parados", a diferencia, < > ya tiene un *path* definido donde buscar, está definida en la carpeta de instalación de C.

Static es una variable que permanece en memoria mientras el programa está corriendo; a diferencia una auto variable se elimina cuando la función donde fue declarada se terminó.

2. Investiga* sobre BLAS, CBLAS, LAPACK, ATLAS y las operaciones de nivel 1, nivel 2 y nivel 3 de BLAS y reporta sobre esta investigación que realizas. Es una investigación que contiene principalmente una descripción sobre los paquetes y niveles anteriores.

BLAS: son una serie de rutinas que nos permiten realizar operaciones vectoriales y Matriciales.

Op 1: estas rutinas nos permiten realizar operaciones con escalares, vectores y operaciones entre vectores.

Op 2: son rutinas para realizar operaciones entre matrices y vectores.

Op 3: son rutinas que nos permiten realizar operaciones entre matrices.

CBLAS: Al parecer es una especie de versión de BLAS para C.

LAPACK: es una serie de rutina que nos permite resolver sistemas de ecuaciones lineales, mínimos cuadrados, determinar eigenvectores y eigenvalores y problemas de valores singulares. También podemos implementar factorización de matrices como LU, Cholesky, QR, SVD, Schur y Schur generalizado.

ATLAS: Sirve para generar automáticamente librerías optimizadas de BLAS para arquitecturas en particular. Atlas incluye algunas librerías de BLAS pre-construidas y optimizadas.

LAPACK is written in Fortran 90 and provides routines for solving systems of simultaneous linear equations, least-squares solutions of linear systems of equations, eigenvalue problems, and singular value problems. The associated matrix factorizations (LU, Cholesky, QR, SVD, Schur, generalized Schur) are also provided, as are related computations such as reordering of the Schur factorizations and estimating condition numbers. Dense and banded matrices are handled, but not general sparse matrices. In all areas, similar functionality is provided for real and complex matrices, in both single and double precision.

3. En la carpeta analisis-numerico-computo-cientifico/C/BLAS/ejemplos/level1/ ejecuta el programa dot_product.c y realiza pruebas con diferentes vectores definidos por ti.

Al hacer diversas pruebas poder ver que el número al final del .out nos regresa el Número de registros a mostrar, es como una especie de head() en R. Me di cuenta aumentando/disminuyendo el número de renglones y viendo que no salía si le ponía un número más grande.

```
root@122751718bc8:/datos# ./programa.out 4
-----
vector :
vector[0]=1.00000
vector[1]=0.00000
vector[2]=-3.00000
vector[3]=0.00000
-----
vector :
vector[0]=5.00000
vector[1]=8.00000
vector[2]=9.00000
vector[3]=0.00000
-----
resultado: -22.000000
```

4. Investiga* sobre la subrutina de Fortran ddot (parámetros que recibe y la salida). La rutina DDOT hace el producto punto entre 2 vectores, puede tener hasta 5 Parámetros:

- N: es el número de elementos en el vector de entrada.
- DX: es un arreglo de doble precisión.
- DY: Es un arreglo de doble precisión.
- INCY: Es el espacio de almacenamiento entre elementos de DY.
- INCX: Es el espacio de almacenamiento entre elementos de DY.