|  |  |  |
| --- | --- | --- |
|  | **Carátula para entrega de prácticas** | |
| Facultad de Ingeniería | | Laboratorio de docencia |

Laboratorios de computación salas A y B

|  |  |
| --- | --- |
| *Profesor:* | Rodríguez Espino Claudia |
| *Asignatura:* | Fundamentos de programación |
| *Grupo:* | 1102 |
| *No de Práctica(s):* | 6 |
| *Integrante(s):* | Martínez Soto Mariana |
|  |  |
|  |  |
| *Semestre:* | 1 |
| *Fecha de entrega:* | Viernes 22 de Septiembre |
| *Observaciones:* | -Lectura de la práctica  -Ejemplos |
|  |  |

CALIFICACIÓN: \_\_\_\_\_\_\_\_\_

**Guía práctica de estudio 06: Entorno de C (editores, compilación y ejecución)**

**Objetivo:** Conocer y usar los ambientes y herramientas para el desarrollo y ejecución de programas en Lenguaje C, como editores y compiladores en diversos sistemas operativos.

**En clase:** Leímos la práctica de manera grupal, definimos los comandos y como utilizarlos en el editor de GNU /LINUX. Antes de eso comenzamos nuestra sesión no como de costumbre sino usando Linux como nuestros sistema operativo, de esta manera trabajamos en su editor, algo así como trabajar en cmd de Windows el cual se encuentra en la parte superior izquierda del escritorio, en aplicaciones y en herramientas. Nos dimos a la tarea de realizar sencillos ejemplos como crear un archivo.c como editarlo y como compilarlo fuera y dentro del *vi* además de ejecutarlo. Para la creación de uno debemos usar el comando touch + nombre del archivo y su extensión, para modificar el nombre del archivo ejecutable del código es necesario gcc nombre del archivo.c -o nombre del archivo nuevo. Esto facilita bastante la ejecución del programa que hemos creado pues todos los archivos ejecutables se guardan en automático como a.c lo que hace que al ejecutar uno se ejecuten todos a la vez y provoque una "saturación".

Al estar dentro, editando nuestro código fuente (en este caso lo hicimos al estilo lenguaje c) podemos elegir la manera de "navegarlo" ya sea por modo de comandos o por modo escritura, para cambiarlo solo es necesario presionar "Esc" + i si se quiere ir al modo escritura y así comenzar a editar nuestro código fuente, ya que al entrar el modo automático es el modo comando. Para guardar los cambios que hemos hecho se necesita presionar "Esc" + w y para salir "Esc" + q. Si se quiere ejecutar fuera del editor de nuestro código solo hace falta poner "./nombre del archivo ejecutable.out"

Con estas herramientas previas nos dimos a la tarea de realizar las siguientes actividades:

1REALIZAR LA SUMA DE DOS NÚMEROS

2 MODIFICAR EL PROGRAMA SUMA DE DOS NÚMEROS CAMBIANDO EL PROCESO DE SUMA A RESTA Y GUARDAR EL ARCHIVO COMO RESTA.C

3 CON EL PROGRAMA SUMA C PRACTICAR LOS COMANDOS DETALLADOS EN LA PARTE "MODO COMANDO" Y "MODO ULTIMA LINEA"

4 COMPILAR ARCHIVO SUMA CON GCC (COMPILER COLLECTION)

5 VERIFICAR LA CREACIÓN DEL ARCHIVO suma.out (fácilmente esto se observa al presionar el comando *ls*)

6. REALIZAR LA COMPLILACIÓN DE SUMA

9. CORRER PROGRAMA EJECUTABLE SUMA

Este es el código en c del programa suma:

#include <stdio.h>

int a,b,c;

main()

{

printf("dame un numero");

scanf("%d", &a);

printf("dame otro numero");

scanf("%d", &b);

c=a+b;

printf("Este es tu resultado %d" ,c);

}

Este es el código del programa resta:

#include <stdio.h>

int a,b,c;

main()

{

printf("dame un numero");

scanf("%d", &a);

printf("dame otro numero");

scanf("%d", &b);

c=a-b;

printf("Este es tu resultado %d" ,c);

}

Conclusiones:

La importancia de dominar los comandos en los sistemas operativos es esencial para poder emplear la programación sin necesidad de un lenguaje de programación (el programa en sí) para que podamos correr nuestro archivo ejecutable. Con este panorama me doy cuenta que se puede programar de muchas maneras y ser más prácticos en ciertos aspectos. Lo hace más cómodo y nos permite entender más la estructura del código en sí.