

## **TRABAJO PRÁCTICO N° 1 - Parte 1**



### **Parte 1 - Temas: Herramientas: uso de consola, edición, compilación y ejecución de un programa básico en lenguaje en C con variables simples. Operadores**

Observación: Durante el cursado de la asignatura, en clases prácticas se trabajará en la terminal que viene incorporada en Visual Studio Code para compilar los programas. Solamente en esta primera clase veremos cómo se opera con la terminal propia del sistema operativo (en el laboratorio todas las pc poseen sistema operativo Linux).

#### **Para recordar:**

#### **Comandos básicos de Linux**

La sintaxis general de los comandos es: **comando opciones argumentos**. Recordar el uso del espacio entre comando, opciones y argumentos. Las opciones generalmente van precedidas por un guión (-).

<b>Comando</b>	<b>¿Para qué sirve?</b>	<b>Ejemplo</b>
man	Manual de ayuda.	man passwd
info	Ídem al anterior.	info mkdir
ls	Lista los archivos existentes en el directorio	ls -l
cat	Muestra, concatena contenidos de archivos.	cat archivo1 archivo2 > nuevo
more	Muestra el contenido de un archivo en forma paginada.	more archivo1
cp	Copia archivos.	cp archivo1 archivo2
mv	Mover y renombrar archivos y directorios.	mv archivo1 archivo2
rm	Borra archivos y directorios.	rm archivo1
mkdir	Crea directorio	mkdir directorio1
rmdir	Borra el directorio, si está vacío.	rmdir directorio1
cd	Cambia el directorio activo.	cd directorio1/directorio2
passwd	Cambiar la clave de acceso	passwd
tree	Lista el contenido del directorio en forma de árbol	tree

## Lenguaje C, Estructuras de control, Datos simples.

### ETAPAS A SEGUIR PARA LA CREACIÓN DE UN PROGRAMA

Para escribir el programa, se debe utilizar un editor de textos, por ejemplo: notepad++, sublime text o **visual code studio** (recomendado por la cátedra). Una vez finalizada la escritura del programa, se debe guardar dicho archivo. El archivo generado constituye el código fuente del programa. En el momento de guardar el archivo, se debe elegir un nombre que **no posea espacios ni símbolos especiales y debe poseer la extensión .c**, por ejemplo un nombre válido sería el siguiente: nombre\_archivo.c.

Para compilar el código escrito, linkear el código objeto (este último se genera a partir del código fuente) con las librerías y generar un archivo ejecutable, se utilizará el **compilador de C, gcc**.

Para realizar esta tarea deberá escribir la siguiente instrucción y luego presionar enter:

**gcc nombre\_archivo\_fuente -o nombre\_archivo\_ejecutable**

Donde **nombre\_archivo\_fuente** es el nombre del archivo creado y guardado con el editor de textos, con extensión .c y **nombre\_archivo\_ejecutable** es el nombre que tendrá el archivo ejecutable generado. Cuando se asigne un nombre a este último archivo, **NO** debe tener exactamente el mismo nombre y extensión que el archivo creado con el editor de textos. Dado que, si fuese así, se corre el riesgo de sobrescribir el archivo fuente con el archivo ejecutable, por lo tanto se perdería todo el código escrito.

Por ejemplo, para el problema 1 del Trabajo Práctico, si se guarda el código escrito con el nombre Ejerc1.c, para compilarlo se debe escribir lo siguiente y luego presionar enter:

**gcc Ejerc1.c -o Ejerc1**

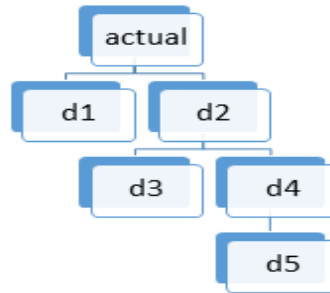
Después de hacer esta acción, puede suceder que aparezca en pantalla unos mensajes, indicando que el programa fuente posee errores en el código escrito, en este caso se deberá editar nuevamente el archivo fuente para realizar las correcciones correspondientes, luego se debe guardar nuevamente el archivo con los cambios realizados y volver a compilar el programa, hasta que no aparezca en pantalla ningún mensaje de error.

Luego para ejecutar el programa obtenido, se debe escribir: ./Ejerc1 y a continuación presionar enter.

## Problemas resueltos

### Comandos básicos de Linux

1. Crear la siguiente estructura de directorios a partir del directorio actual de trabajo (por defecto debería ser la carpeta estudiante):



2. Utilizando el editor de textos nano, escribir la frase "hola mundo" en un archivo llamado **comandos1.txt** y guardarlo en el subdirectorio **actual**.
3. Mover el archivo **comandos1.txt** al subdirectorio **d4**.
4. Cambiar el nombre del archivo **comandos1.txt** a **secuencia1.txt**.
5. Mover el subdirectorio **d2** al directorio **d1**.
6. Copiar el archivo **secuencia1.txt** al subdirectorio actual.
7. Eliminar los subdirectorios que se encuentran debajo del subdirectorio actual.

La secuencia de Comandos utilizada es la siguiente, siendo la ubicación el subdirectorio **actual**:

```

mkdir actual
cd actual
mkdir d1
mkdir d2
mkdir d2/d3
mkdir d2/d4
mkdir d2/d4/d5
nano comandos1.txt

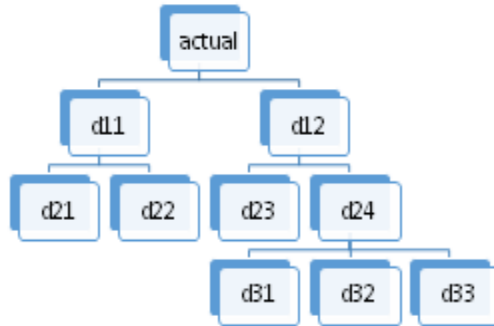
mv comandos1.txt d2/d4
mv d2/d4/comandos1.txt d2/d4/secuencia1.txt
mv d2 d1
cp d1/d2/d4/secuencia1.txt ./
rm -r *
```

## Problemas para resolver

### Comandos básicos y Editor de texto en Linux

*Para la resolución de los siguientes problemas, debe utilizar los comandos de Linux y el Editor de texto nano:*

1. Utilizando comandos de Linux, crear la siguiente estructura de directorios:



2. Posicionarse en el directorio d2, utilizando el editor de textos nano, escribir el siguiente código:

```
#include <stdio.h>
int main() {
    printf("Hola mundo \n");
    return 0;
}
```

Guardar el archivo con el nombre **ejemplo1.c**

3. Cambiar el nombre del subdirectorio d21 por el nombre d\_21.
4. Luego mover el subdirectorio d24 al subdirectorio d22.
5. Copiar el archivo ejemplo1.c al subdirectorio d22 pero con el nombre holamundo.c.
6. Copiar el archivo holamundo.c al subdirectorio d\_21 y cambiarle el nombre por holaargentina.c
7. Por medio del editor nano, modificar el contenido de **holaargentina.c** para que quede de la siguiente manera:

```
#include <stdio.h>
int main() {
    printf("Hola Argentina \n");
    return 0;
}
```

8. Borrar todos los subdirectorios que se encuentren debajo del subdirectorio d2.
9. Ejecutar el comando **tree** y luego el comando **tree /** ¿Qué diferencias encuentra en lo que se muestra?
10. Compile el archivo holamundo.c para generar un archivo ejecutable a.out
11. Ejecute el programa a.out
12. Con nano modifique el programa holamundo.c para que quede de la siguiente manera
 

```
#include <stdio.h>

int main() {
    printf("Hola Tucuman \n");
    return 0;
}
```
13. Compile el archivo holamundo.c nuevamente para generar un archivo ejecutable pero esta vez que tenga en nombre holatucuman.out
14. Ejecute el programa holatucuman.out
15. ¿Qué conclusiones puede obtener acerca del proceso de compilación?.

## Lenguaje C, Estructuras de control, Datos simples.

*Para los presentes ejercicios **recordar que debe:***

- ❖ *Indentar el código para una mejor comprensión*
- ❖ *Usar el standard C*
- ❖ *El código fuente se almacena en archivos con extensión .C*
- ❖ *Usar VSCode para esta parte del práctico.*

*Diseñar el algoritmo y codificar los programas en lenguaje C que permitan resolver los siguientes problemas:*

**ES OBLIGATORIO UTILIZAR NOMBRES DE VARIABLES SIGNIFICATIVOS**

1. Realizar un programa en C en el que se declare una variable entera, se asigne un valor que Ud. elija y se la presente por pantalla.
2. Leer dos números enteros, calcular y presentar por pantalla el producto, la suma y la media aritmética (promedio) de los números ingresados.
3. Leer cuatro números enteros, calcular y presentar por pantalla la media aritmética (promedio) de los números ingresados.
4. Realizar un programa que muestre por pantalla el contenido de dos variables llamadas "peso" y "altura" (el valor de cada una será definido por ud). La presentación de pantalla deberá respetar el siguiente formato: "El peso de Carlos es .... Kg, mientras que su altura es ... m".
5. Realizar un programa que permita calcular la longitud de una circunferencia, donde la

misma se calcula como:

$$\ell = 2\pi \cdot r$$

Siendo pi una constante que vale 3,14 . Elegir el valor de r.

6. Ingresar dos números reales, luego presentar por pantalla el valor de la suma y el producto de los números ingresados. **(Resuelto)**