



# Práctica 1:

## Introducción al sistema operativo Linux y Windows

### Sistemas Operativos – 2CM17

#### Integrantes:

Mora Ayala José Antonio

Ramírez Cotonieto Luis Fernando

Torres Carrillo Josehf Miguel Ángel

Tovar Jacuinde Rodrigo

#### Profesor:

Cortés Galicia Jorge



# INTRODUCCIÓN

Dentro de esta primera practica se abordaran practicas relacionadas con temas y aplicaciones previas que hemos visto a lo largo de la carrera, así como con las previas lecturas dictaminadas por la Tarea numero 1, la cual tiene relación a los Sistemas Operativos, clasificación, funcionamientos y la historia de los mismos, así como conceptos básicos que debemos entender para poder adentrarnos de una forma correcta y mas completa al mundo de la programación y el desarrollo. Comenzaremos con la definición de un Sistema Operativo el cual es un programa (software) que cuando arrancamos o iniciamos el ordenador se encarga de gestionar todos los recursos del sistema informático permitiendo así la comunicación entre el usuario y el ordenador.

Los recursos que gestiona son tanto del hardware (partes físicas, disco duro, pantalla, teclado, etc.) como del software (programas e instrucciones). Los otros programas, que no son SO, se llaman aplicaciones. Resumiendo, un SO es el encargado de gestionar el software y el hardware de un ordenador o computadora. En su sentido más general, un SO es un software que permite a un usuario ejecutar otras aplicaciones en un dispositivo informático (ordenador, Smartphone, Tablet, etc.). La mayoría de la gente usa el sistema operativo que viene en su ordenador o móvil, pero es posible actualizarlo o incluso cambiar el sistema operativo por otro diferente.

## ¿PARA QUÉ SIRVE EL SISTEMA OPERATIVO?

Los sistemas operativos utilizan imágenes y botones para poder comunicarnos con el ordenador de forma sencilla y fácil para decirle lo que queremos hacer en cada momento a nuestro ordenador. Su función principal es la de darnos las herramientas necesarias para poder controlar nuestra computadora y poder hacer uso de ella, de la forma más cómoda y sencilla posible.

Las funciones básicas del Sistema Operativo son administrar los recursos del ordenador, coordinar el hardware y organizar archivos y directorios en los dispositivos de almacenamiento de nuestro ordenador.

Algunas cosas más concretas que puede realizar un Sistema Operativo son:

- El sistema operativo es la única gran pieza de software que ejecuta los programas y se encarga de todo lo demás.

Por ejemplo, el sistema operativo controla los archivos y otros recursos a los que pueden acceder estos programas una vez ejecutados.

- Cuando tenemos múltiples programas que se pueden ejecutar al mismo tiempo, el sistema operativo determina qué aplicaciones se deben ejecutar en qué orden y cuánto tiempo.
- Gestiona el intercambio de memoria interna entre múltiples aplicaciones.
- Se ocupa de la entrada y la salida de los datos desde y hacia los dispositivos de hardware conectados, tales como discos duros, impresoras, puertos de comunicación, teclado, etc.

### Tipos de Sistemas Operativos

Ha nivel de usuario de un PC, podemos decir que hay dos tipos de Sistemas Operativos: SO para PC y SO para Teléfonos Móviles. Pero también hay una amplia familia de sistemas operativos categorizados según los tipos de ordenadores que controlan y el tipo de aplicaciones que admiten. Las 4 categorías para este tipo de SO son: **SO en tiempo real, SO Usuario Único Tarea Única, SO Usuario Único Multitarea y SO Multiusuario.**

- Sistema operativo en tiempo real (RTOS): los sistemas operativos en tiempo real se utilizan para controlar maquinaria, instrumentos científicos y sistemas industriales. Una parte muy importante de un RTOS es administrar los recursos de la computadora para que una operación particular se ejecute exactamente en la misma cantidad de tiempo, cada vez que ocurre.
- Usuario único, tarea única: como su nombre lo indica, este sistema operativo está diseñado para administrar la computadora de modo que un usuario pueda hacer una cosa a la vez.
- Usuario único, multitarea: este es el tipo de sistema operativo que la mayoría de la gente usa en sus computadoras de escritorio y portátiles en la actualidad.

Windows de Microsoft y las plataformas MacOS de Apple son ejemplos de sistemas operativos que permitirán que un solo usuario tenga varios programas en funcionamiento al mismo tiempo.

- Multiusuario: un sistema operativo multiusuario permite que muchos usuarios diferentes aprovechen los recursos de la computadora simultáneamente.

## DESARROLLO EXPERIMENTAL

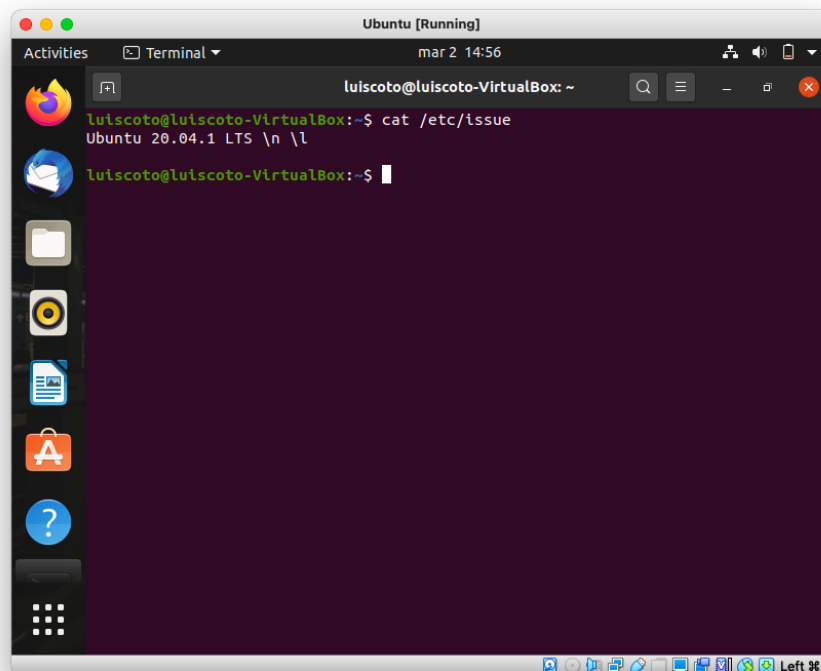
Competencias.

- El alumno aprende a familiarizarse con los sistemas operativos Linux y Windows mediante la exploración de sus ambientes de trabajo para desarrollar aplicaciones en lenguaje C.
- El alumno analiza el sistema operativo Linux y Windows mediante la comparación de sus características principales para diferenciarlos en su ambiente de trabajo.

## SECCIÓN LINUX:

1. Verifique la existencia del sistema operativo Linux y su correcto funcionamiento con el entorno gráfico (si el entorno gráfico no funciona inicie sesión en modo texto).
2. Reporte la distribución que usted está utilizando. Mencione que otras distribuciones de Linux existen y cuáles son las diferencias respecto a la distribución que usted está utilizando.

Estamos usando la distribución de Ubuntu en su versión 20.04 ya que es el OS basado en Linux más utilizadas por la facilidad de uso además de ser amigable para principiantes en su instalación y uso



Algunas otras distribuciones de linux son:

**Fedora:** Distribución gratuita creada y mantenida por la empresa Red Hat que utiliza el sistema de paquetería RPM (Red Hat Package Manager). Tiene tres versiones diferentes para escritorio, servidores y sistemas en la nube, y destaca por su seguridad gracias al sistema SELinux ("Security-Enhanced Linux")

**Arch Linux:** Una distribución modular en la que empiezas desde cero y tienes que ir añadiéndole los componentes que quieras. No es muy apta para principiante, y utiliza pacman, su propio gestor de paquetes. Se trata de una Rolling Release, lo que quiere decir que todos sus componentes van actualizándose sin necesidad de instalar versiones nuevas del sistema operativo.

**Linux Mint:** Basado en Ubuntu, es uno de los más recomendados para todos aquellos que tocan Linux por primera vez. Su entorno de escritorio, Cinnamon, tiene muchas similitudes con el de Windows, y es también uno de los más personalizables

**Peppermint OS:** Distribución rápida y ligera basada en Ubuntu con entorno de escritorio LXDE. Utiliza la tecnología Prism de Mozilla para integrarse con las aplicaciones basadas en la nube, utilizando las webapps como si fueran nativas. Se presenta como una alternativa a otros sistemas basados en la nube como Chrome OS

**Kali Linux:** Distribución basada en Debian con una inmensa colección de herramientas para proteger nuestros equipos. Utiliza un kernel personalizado con parches de seguridad y tiene soporte para la arquitectura ARM.

**Puppy Linux:** Una minúscula distribución que puede llevarse en un USB o CD, pero sorprendentemente rápida al cargarse enteramente en la memoria RAM del ordenador. Se carga en 30 o 40 segundos y ocupa sólo 100 MB.

**Debian:** Muy estable y 100% libre, Debian destaca por su sistema de paquetería .deb y su gestión de paquetes APT. Es una de las distribuciones más importantes de GNU/Linux, ya que en ella se basan gigantes como Ubuntu.

**Red Hat Enterprise Linux:** Distribución comercial de Linux desarrollada por Red Hat. Ofrece una estabilidad y flexibilidad punteras, lo que la coloca como una de las más recomendadas para empresas y servidores.

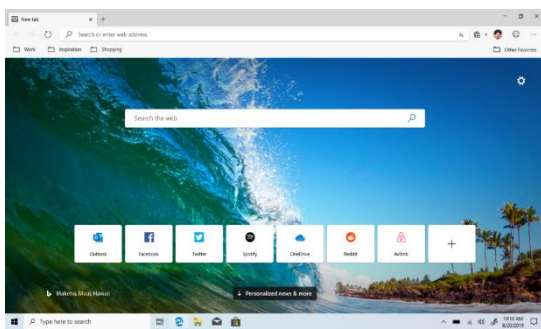
**CentOS:** Nació como un derivado gratuito de la distribución comercial Red Hat Enterprise Linux (RHEL) destinada al uso empresarial. Recientemente unió las fuerzas con el propio Red Hat, y sigue siendo una apuesta segura para los que busquen un código de gran calidad.

3. Revise el entorno Linux e identifique los componentes de menú con los que cuenta su sistema operativo. Entre en cada uno de los componentes del menú y explore algunas de sus funcionalidades. Reporte que funcionalidades encontró, además haga un comparativo de estas funcionalidades con respecto a las existentes en el sistema operativo Windows que usted utilice.

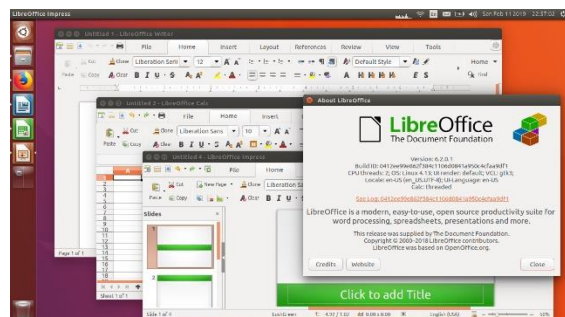
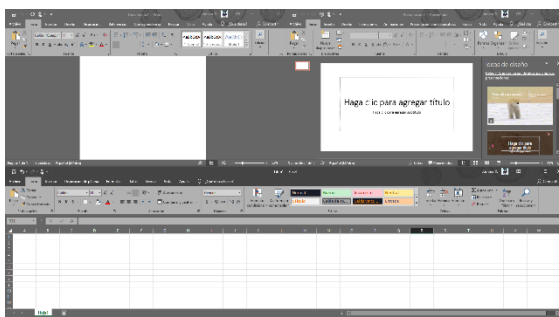
Las diferencias con los sistemas son bastantes, tanto que a simple vista cambia completamente el entorno, principalmente algunas características que debemos notar es que las barras de tarea están ubicadas y distribuidas de manera totalmente diferente



Viene instalado el explorador de internet Mozilla Firefox por defecto mientras que en Windows actualmente se opta por el nuevo Microsoft Edge



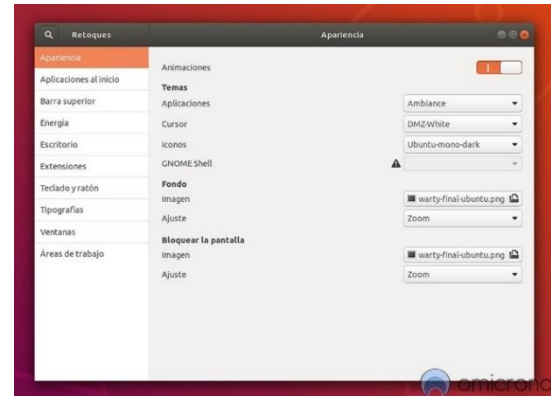
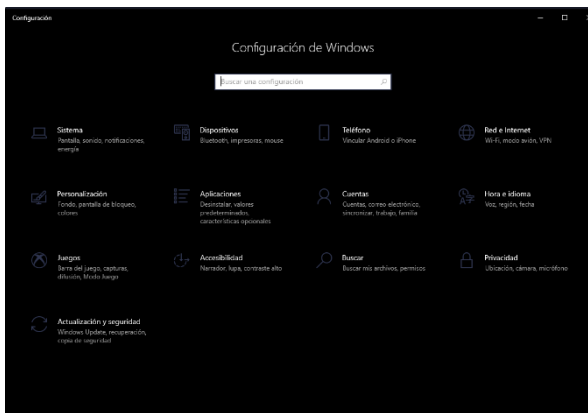
Las paqueterías de oficinas son diferentes en Ubuntu tenemos libre office mientras que en Windows tenemos Microsoft office



Los menús donde tenemos las aplicaciones son distintos el de Ubuntu recuerda un poco a lo que fue Windows 8 al tener un menú que se extiende por toda la pantalla



Los menús de configuraciones tienen una distribución distinta entre sí, siendo Ubuntu un poco más lite que Windows en el sentido de que no tiene tantos apartados



Otras características diferenciadas es que sus paletas de colores, y distribuciones en distintas aplicaciones son tan opuestas en uno a la otra, los gestores de archivos, la cantidad de parámetros configurables en ambos sistemas y el software tan orientado a cada sistema los hace bastante únicos además de que un gran punto a favor de Ubuntu es que es totalmente gratuito, a diferencia de Windows de que si queremos tener todas las funcionalidades desbloqueadas como personalización tenemos que adquirir una licencia

Otras diferencias que no son notables a nivel visual pero si internamente son que el código de Ubuntu es open source esto quiere decir que cualquiera puede modificarlo a su antojo sin ninguna restricción, las actualizaciones se hacen de manera manual, es decir cada cierto tiempo tienes que reinstalar Ubuntu a la nueva versión, para ello hay versiones que duran un año completo evitando estos problemas, tiene una mejor seguridad con respecto a Windows ya que su código desde sus comienzos fue creado atendiendo principios de seguridad y la optimización en Ubuntu es enormemente mayor a otros OS siendo más ligero y rápido al abrir aplicaciones o procesamiento de programas pesados



4. Abra una sesión de consola (Terminal) y reporte para que se utilizan cada uno de los siguientes comandos:

<b>ls:</b>	<b>cd:</b>	<b>cp:</b>
<b>chmod:</b>	<b>cat:</b>	<b>mv:</b>
<b>vi:</b>	<b>grep:</b>	<b>mkdir:</b>
<b>pwd:</b>	<b>rm:</b>	<b>rmdir:</b>
<b>clear:</b>	<b>ps:</b>	<b>whoami:</b>

para ello utilice el manual en línea (man) con el que cuenta linux. La forma de usarlo es tecleando el siguiente comando en la Terminal:

**man nombre\_del\_comando\_a\_buscar, por ejemplo: man ls, man ps.**

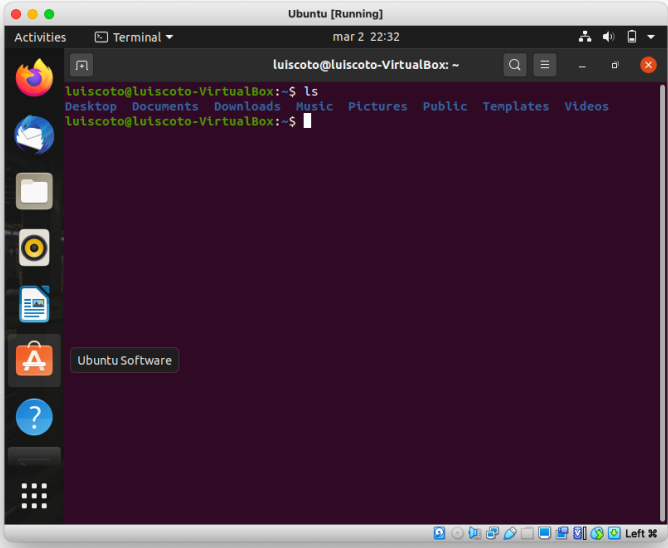
5. Pruebe desde la terminal los siguientes comandos (tenga cuidado al ejecutar los comandos de borrado de archivos y directorios):

<b>ls:</b>	<b>cd nombre_directorio:</b>	<b>cp [ruta1]archivo1 [ruta2][archivo 2]:</b>
<b>ls -l</b>	<b>cat nombre_archivo:</b>	<b>Mv [ruta1]archivo1 [ruta2][archivo 2]:</b>
<b>ls -la</b>	<b>ls -la   more:</b>	<b>mkdir nombre_directorio:</b>
<b>pwd:</b>	<b>rm nombre_archivo:</b>	<b>rmdir nombre_directorio:</b>
<b>clear:</b>	<b>ps:</b>	<b>whoami:</b>

Observe los resultados de la ejecución de cada comando.

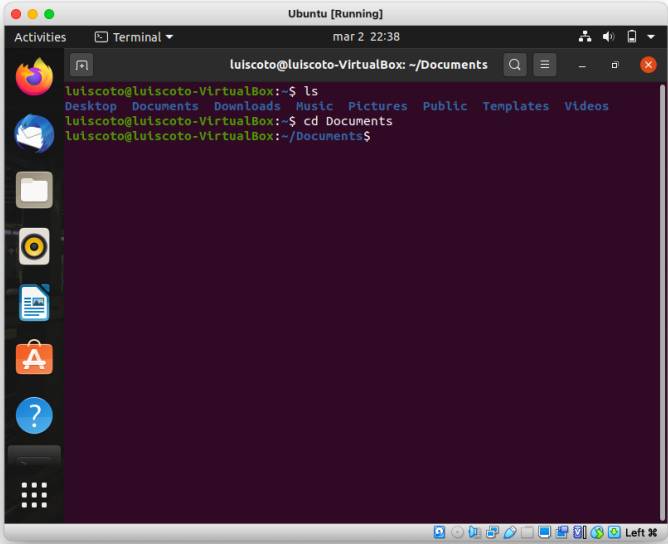


- I. **ls:** Se utiliza para listar los archivos y directorios en el directorio de trabajo actual.



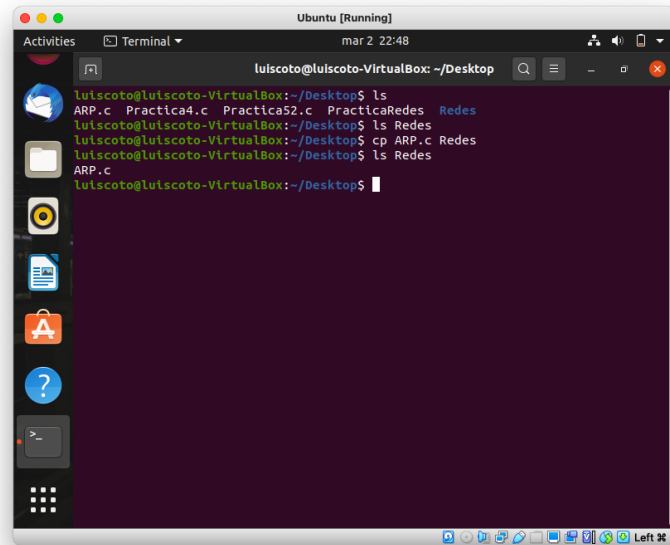
A terminal window titled "Ubuntu [Running]" with a date and time of "mar 2 22:32". The prompt is "luiscoto@luiscoto-VirtualBox: ~". The command "ls" has been entered, and the output lists the following directories: Desktop, Documents, Downloads, Music, Pictures, Public, Templates, and Videos. The prompt is now "luiscoto@luiscoto-VirtualBox:~\$".

- II. **cd:** Nos cambia el directorio de trabajo.



A terminal window titled "Ubuntu [Running]" with a date and time of "mar 2 22:38". The prompt is "luiscoto@luiscoto-VirtualBox: ~/Documents". The command "ls" has been entered, and the output lists the following directories: Desktop, Documents, Downloads, Music, Pictures, Public, Templates, and Videos. The prompt is now "luiscoto@luiscoto-VirtualBox:~/Documents\$".

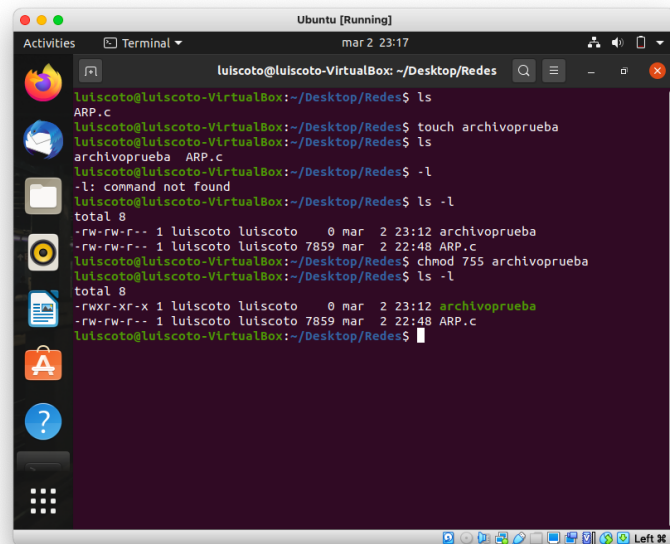
- III. **cp:** Nos permite copiar un archivo a otra carpeta existente



```
luiscoto@luiscoto-VirtualBox: ~/Desktop
luiscoto@luiscoto-VirtualBox:~/Desktop$ ls
ARP.c  Practica4.c  Practica52.c  PracticaRedes  Redes
luiscoto@luiscoto-VirtualBox:~/Desktop$ ls Redes
luiscoto@luiscoto-VirtualBox:~/Desktop$ cp ARP.c Redes
luiscoto@luiscoto-VirtualBox:~/Desktop$ ls Redes
ARP.c
luiscoto@luiscoto-VirtualBox:~/Desktop$
```

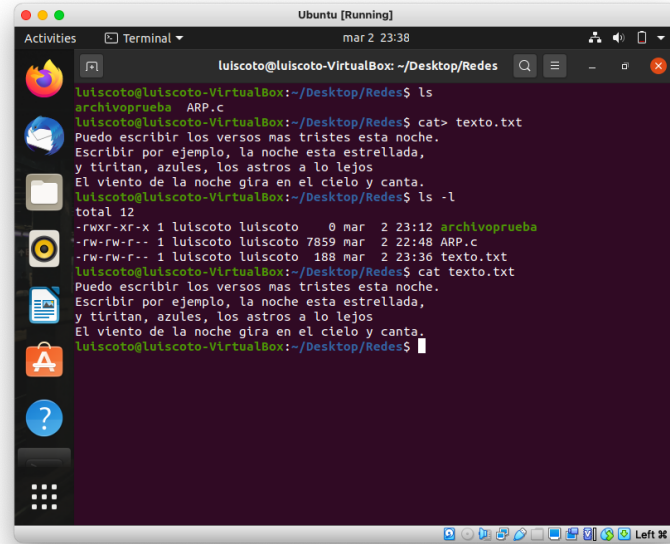
IV. **chmod**: Cambia los permisos de archivos o directorios, siendo:

- 0: Sin permisos      1: Ejecución      2: Escritura      3: Lectura y escritura  
4: Lectura      5: Lectura y ejecución      6: Lectura y escritura



```
luiscoto@luiscoto-VirtualBox: ~/Desktop/Redes
luiscoto@luiscoto-VirtualBox:~/Desktop/Redes$ ls
ARP.c
luiscoto@luiscoto-VirtualBox:~/Desktop/Redes$ touch archivoprueba
luiscoto@luiscoto-VirtualBox:~/Desktop/Redes$ ls
archivoprueba  ARP.c
luiscoto@luiscoto-VirtualBox:~/Desktop/Redes$ -l
-l: command not found
luiscoto@luiscoto-VirtualBox:~/Desktop/Redes$ ls -l
total 8
-rw-rw-r-- 1 luiscoto luiscoto  0 mar  2 23:12 archivoprueba
-rw-rw-r-- 1 luiscoto luiscoto 7859 mar  2 22:48 ARP.c
luiscoto@luiscoto-VirtualBox:~/Desktop/Redes$ chmod 755 archivoprueba
luiscoto@luiscoto-VirtualBox:~/Desktop/Redes$ ls -l
total 8
-rwxr-xr-x 1 luiscoto luiscoto  0 mar  2 23:12 archivoprueba
-rw-rw-r-- 1 luiscoto luiscoto 7859 mar  2 22:48 ARP.c
luiscoto@luiscoto-VirtualBox:~/Desktop/Redes$
```

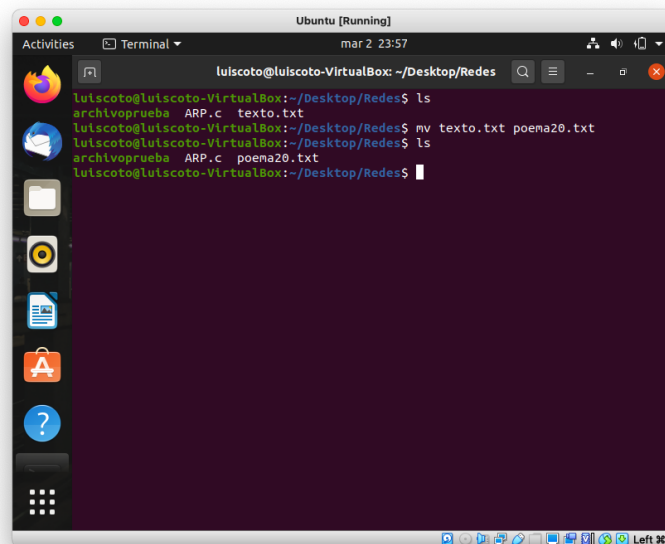
V. **cat**: Te permite crear, fusionar o imprimir archivos en la pantalla de salida estándar o en otro archivo



A screenshot of a terminal window titled "Ubuntu [Running]" with a date and time of "mar 2 23:38". The terminal shows a user named "luisquito" in a "VirtualBox" environment, working in the directory "~/Desktop/Redes". The user runs the following commands:

```
luisquito@luisquito-VirtualBox: ~/Desktop/Redes$ ls
archivoprueba ARP.c
luisquito@luisquito-VirtualBox: ~/Desktop/Redes$ cat > texto.txt
Puedo escribir los versos mas tristes esta noche.
Escribir por ejemplo, la noche esta estrellada,
y tiritan, azules, los astros a lo lejos
El viento de la noche gira en el cielo y canta.
luisquito@luisquito-VirtualBox: ~/Desktop/Redes$ ls -l
total 12
-rwxr-xr-x 1 luisquito luisquito  0 mar  2 23:12 archivoprueba
-rw-rw-r-- 1 luisquito luisquito 7859 mar  2 22:48 ARP.c
-rw-rw-r-- 1 luisquito luisquito 188 mar  2 23:36 texto.txt
luisquito@luisquito-VirtualBox: ~/Desktop/Redes$ cat texto.txt
Puedo escribir los versos mas tristes esta noche.
Escribir por ejemplo, la noche esta estrellada,
y tiritan, azules, los astros a lo lejos
El viento de la noche gira en el cielo y canta.
luisquito@luisquito-VirtualBox: ~/Desktop/Redes$
```

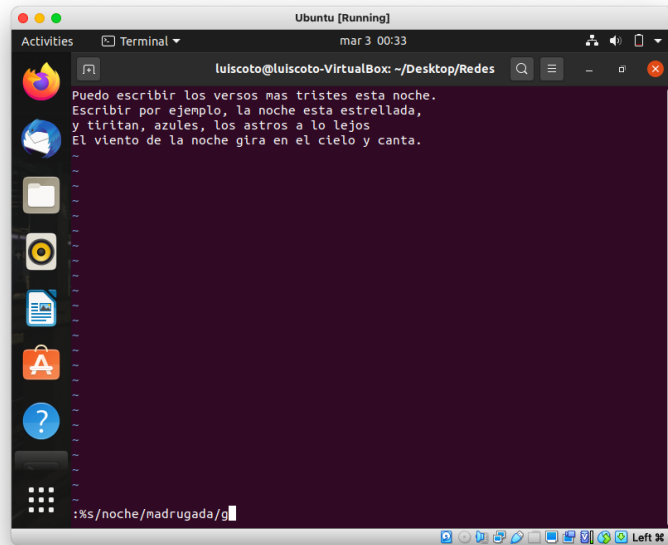
VI. **mv**: Sirve para mover o renombrar archivos o directorios del sistema de archivos.



A screenshot of a terminal window titled "Ubuntu [Running]" with a date and time of "mar 2 23:57". The terminal shows the same user and environment as the previous screenshot. The user runs the following commands:

```
luisquito@luisquito-VirtualBox: ~/Desktop/Redes$ ls
archivoprueba ARP.c texto.txt
luisquito@luisquito-VirtualBox: ~/Desktop/Redes$ mv texto.txt poema20.txt
luisquito@luisquito-VirtualBox: ~/Desktop/Redes$ ls
archivoprueba ARP.c poema20.txt
luisquito@luisquito-VirtualBox: ~/Desktop/Redes$
```

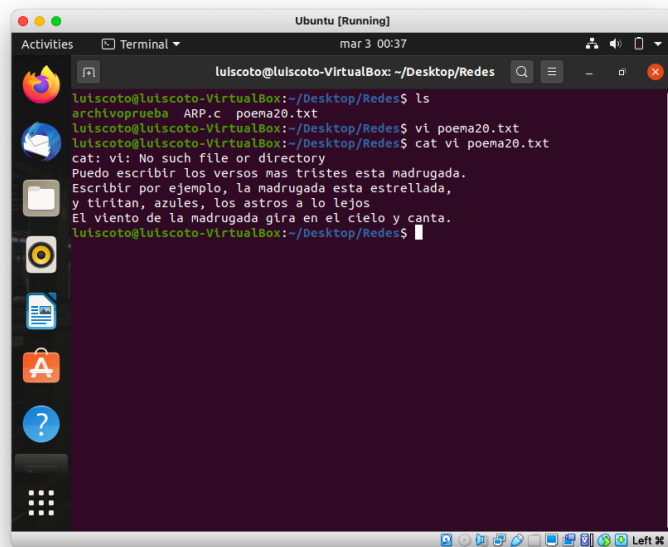
VII. **vi**: Es el editor clásico de todos los sistemas UNIX que se utiliza en modo texto



A screenshot of a terminal window titled "Ubuntu [Running]" with the date "mar 3 00:33". The user is "luiscoto" and the host is "luiscoto-VirtualBox". The current directory is "~/Desktop/Redes". The terminal displays a poem in Spanish:

```
Puedo escribir los versos mas tristes esta noche.  
Escribir por ejemplo, la noche esta estrellada,  
y tiritan, azules, los astros a lo lejos  
El viento de la noche gira en el cielo y canta.
```

The prompt is `:%s/noche/madrugada/g`.

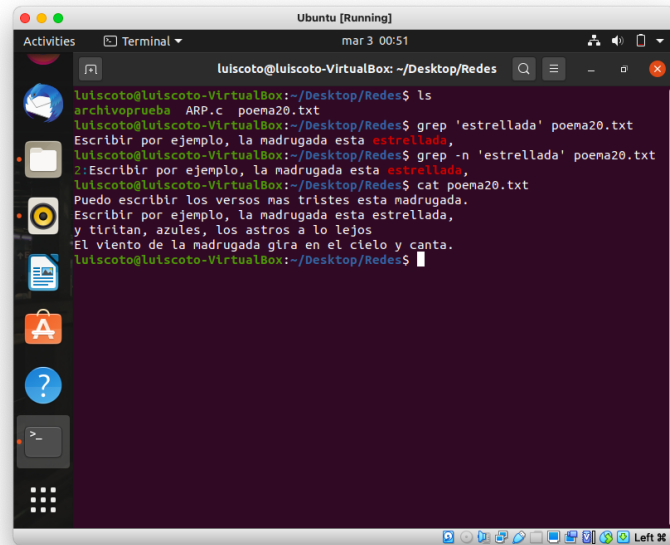


A screenshot of a terminal window titled "Ubuntu [Running]" with the date "mar 3 00:37". The user is "luiscoto" and the host is "luiscoto-VirtualBox". The current directory is "~/Desktop/Redes". The terminal shows the following commands and output:

```
luiscoto@luiscoto-VirtualBox:~/Desktop/Redes$ ls  
archivoprueba ARP.c poema20.txt  
luiscoto@luiscoto-VirtualBox:~/Desktop/Redes$ vi poema20.txt  
luiscoto@luiscoto-VirtualBox:~/Desktop/Redes$ cat vi poema20.txt  
cat: vi: No such file or directory  
Puedo escribir los versos mas tristes esta madrugada.  
Escribir por ejemplo, la madrugada esta estrellada,  
y tiritan, azules, los astros a lo lejos  
El viento de la madrugada gira en el cielo y canta.
```

The prompt is `luiscoto@luiscoto-VirtualBox:~/Desktop/Redes$`.

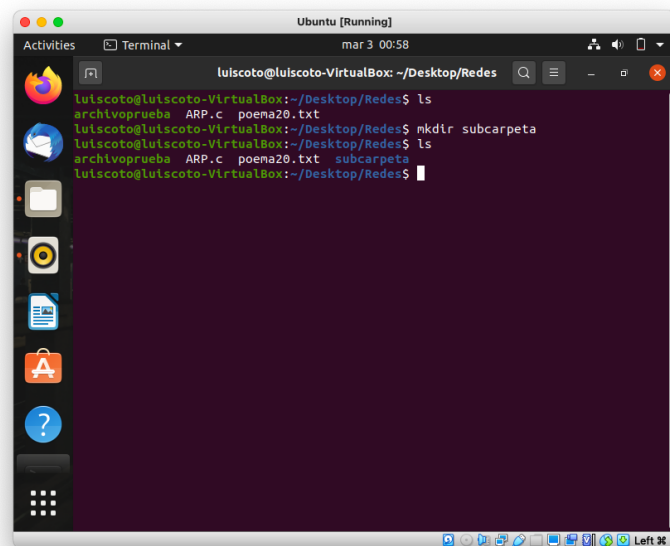
- VIII. **grep:** Toma una expresión regular de la línea de comandos, lee la entrada estándar o una lista de archivos, e imprime las líneas que contengan coincidencias para la expresión regular.



A terminal window titled 'Ubuntu [Running]' with a date of 'mar 3 00:51'. The user is logged in as 'luisccot@luisccot-VirtualBox' in the directory '~/Desktop/Redes'. The terminal shows the following commands and output:

```
luisccot@luisccot-VirtualBox: ~/Desktop/Redes$ ls
archivoprueba ARP.c poema20.txt
luisccot@luisccot-VirtualBox: ~/Desktop/Redes$ grep 'estrellada' poema20.txt
Escribir por ejemplo, la madrugada esta estrellada,
luisccot@luisccot-VirtualBox: ~/Desktop/Redes$ grep -n 'estrellada' poema20.txt
2:Escribir por ejemplo, la madrugada esta estrellada,
luisccot@luisccot-VirtualBox: ~/Desktop/Redes$ cat poema20.txt
Puedo escribir los versos mas tristes esta madrugada.
Escribir por ejemplo, la madrugada esta estrellada,
y tiritan, azules, los astros a lo lejos
El viento de la madrugada gira en el cielo y canta.
luisccot@luisccot-VirtualBox: ~/Desktop/Redes$
```

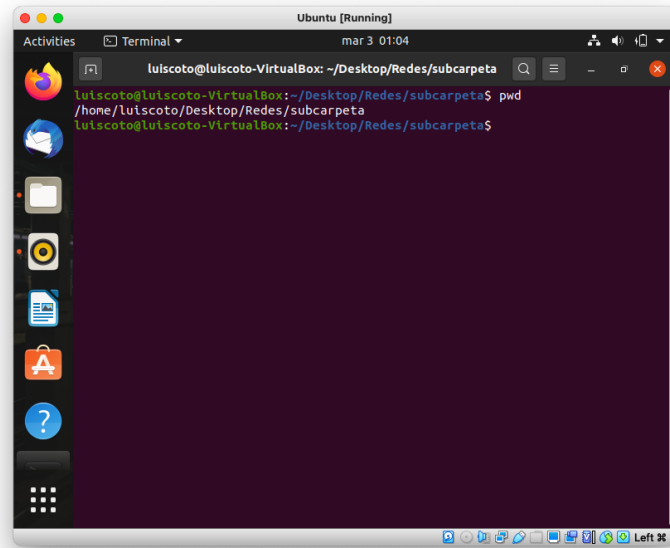
IX. **mkdir**: Crea un nuevo directorio



A terminal window titled 'Ubuntu [Running]' with a date of 'mar 3 00:58'. The user is logged in as 'luisccot@luisccot-VirtualBox' in the directory '~/Desktop/Redes'. The terminal shows the following commands and output:

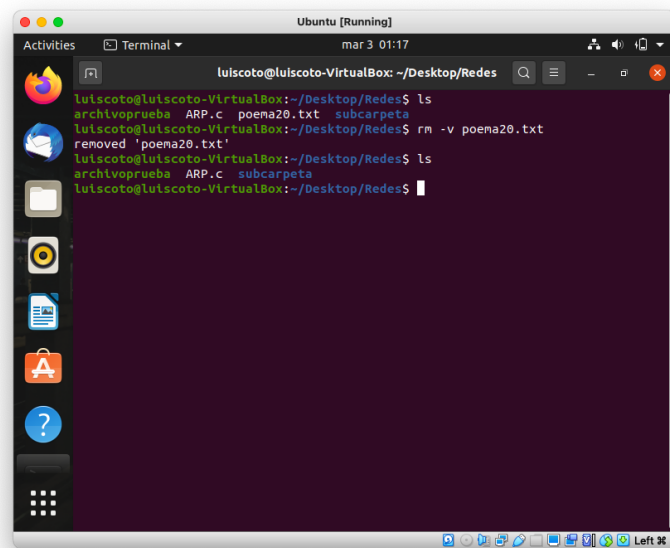
```
luisccot@luisccot-VirtualBox: ~/Desktop/Redes$ ls
archivoprueba ARP.c poema20.txt
luisccot@luisccot-VirtualBox: ~/Desktop/Redes$ mkdir subcarpeta
luisccot@luisccot-VirtualBox: ~/Desktop/Redes$ ls
archivoprueba ARP.c poema20.txt subcarpeta
luisccot@luisccot-VirtualBox: ~/Desktop/Redes$
```

X. **pwd**: Te devuelve la ruta en la que estas situado, se suele utilizar para saber en que parte de la estructura de directorios te encuentras.

A terminal window titled 'Ubuntu [Running]' with a date of 'mar 3 01:04'. The prompt is 'luiscoto@luiscoto-VirtualBox: ~/Desktop/Redes/subcarpeta'. The user enters 'pwd' and the output is '/home/luiscoto/Desktop/Redes/subcarpeta'. The prompt then changes to 'luiscoto@luiscoto-VirtualBox: ~/Desktop/Redes/subcarpeta\$'.

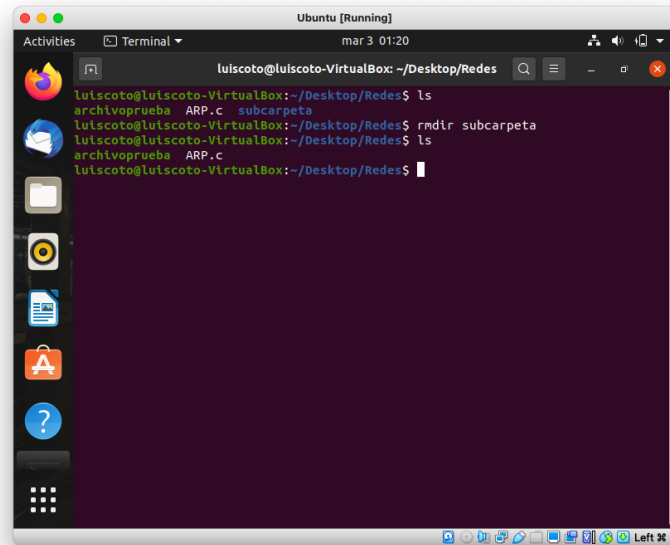
```
luiscoto@luiscoto-VirtualBox: ~/Desktop/Redes/subcarpeta$ pwd
/home/luiscoto/Desktop/Redes/subcarpeta
luiscoto@luiscoto-VirtualBox: ~/Desktop/Redes/subcarpeta$
```

XI. **rm** : Nos sirve para eliminar archivos y directorios del sistema de archivos.

A terminal window titled 'Ubuntu [Running]' with a date of 'mar 3 01:17'. The prompt is 'luiscoto@luiscoto-VirtualBox: ~/Desktop/Redes'. The user enters 'ls' and the output is 'archivoprueba ARP.c poena20.txt subcarpeta'. The user then enters 'rm -v poena20.txt' and the output is 'removed \'poena20.txt\''. The user enters 'ls' again and the output is 'archivoprueba ARP.c subcarpeta'. The prompt then changes to 'luiscoto@luiscoto-VirtualBox: ~/Desktop/Redes\$'.

```
luiscoto@luiscoto-VirtualBox: ~/Desktop/Redes$ ls
archivoprueba ARP.c poena20.txt subcarpeta
luiscoto@luiscoto-VirtualBox: ~/Desktop/Redes$ rm -v poena20.txt
removed 'poena20.txt'
luiscoto@luiscoto-VirtualBox: ~/Desktop/Redes$ ls
archivoprueba ARP.c subcarpeta
luiscoto@luiscoto-VirtualBox: ~/Desktop/Redes$
```

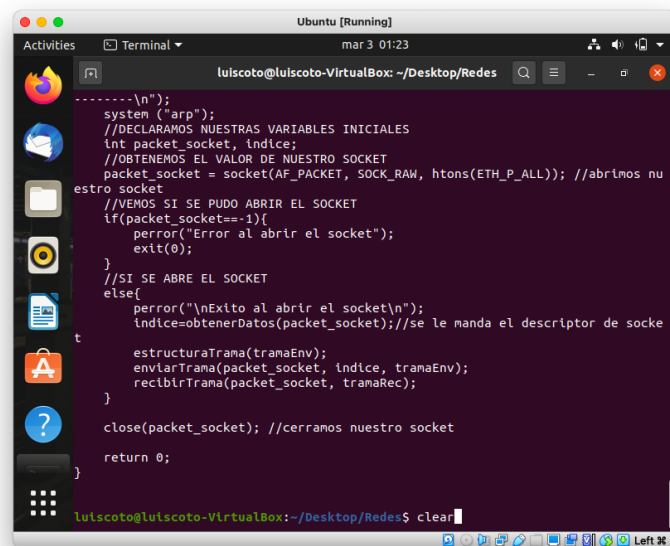
XII. **rmdir**: Sirve para borrar directorios o subdirectorios vacíos de nuestro sistema de archivos.



A terminal window titled 'Ubuntu [Running]' with a date of 'mar 3 01:20'. The prompt is 'luiscolo@luiscolo-VirtualBox: ~/Desktop/Redes'. The user enters 'ls', showing 'archivoprueba ARP.c subcarpeta'. Then they enter 'mkdir subcarpeta'. Then they enter 'ls' again, showing 'archivoprueba ARP.c'.

```
luiscolo@luiscolo-VirtualBox:~/Desktop/Redes$ ls
archivoprueba  ARP.c  subcarpeta
luiscolo@luiscolo-VirtualBox:~/Desktop/Redes$ mkdir subcarpeta
luiscolo@luiscolo-VirtualBox:~/Desktop/Redes$ ls
archivoprueba  ARP.c
```

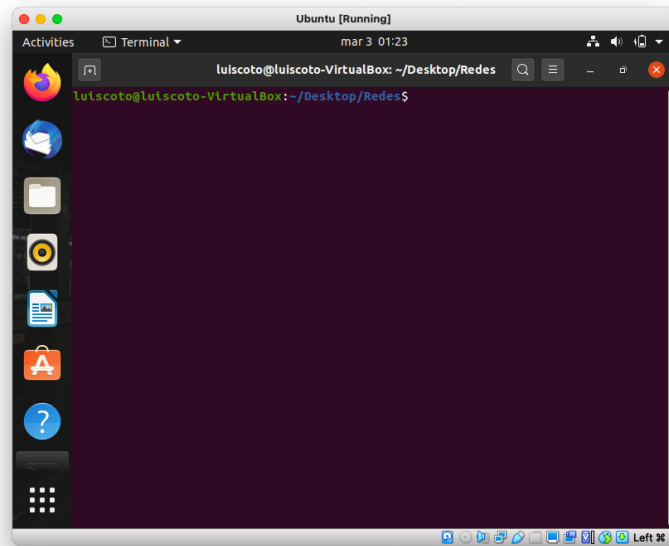
XIII. **clear:** Nos borra el contenido de la pantalla



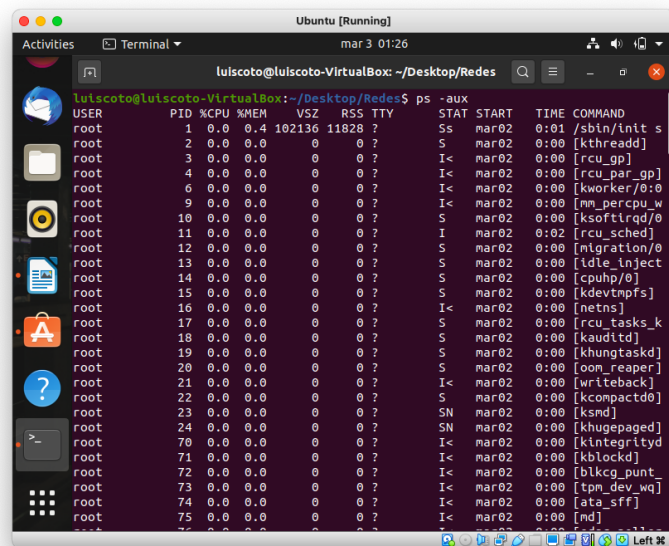
A terminal window titled 'Ubuntu [Running]' with a date of 'mar 3 01:23'. The prompt is 'luiscolo@luiscolo-VirtualBox: ~/Desktop/Redes'. The user pastes a C code snippet. The code includes a system call for 'arp', declares variables, creates a socket, and has functions for sending and receiving packets. At the bottom, the user enters 'clear'.

```
luiscolo@luiscolo-VirtualBox:~/Desktop/Redes$
-----\n");
system("arp");
//DECLARAMOS NUESTRAS VARIABLES INICIALES
int packet_socket, indice;
//OBTENEMOS EL VALOR DE NUESTRO SOCKET
packet_socket = socket(AF_PACKET, SOCK_RAW, htons(ETH_P_ALL)); //abrimos nu
estro socket
//VEMOS SI SE PUDO ABRIR EL SOCKET
if(packet_socket==1){
    perror("Error al abrir el socket");
    exit(0);
}
//SI SE ABRE EL SOCKET
else{
    perror("\nExito al abrir el socket\n");
    indice=obtenerDatos(packet_socket);//se le manda el descriptor de socke
t
    estructuraTrama(tramaEnv);
    enviarTrama(packet_socket, indice, tramaEnv);
    recibirTrama(packet_socket, tramaRec);
}
close(packet_socket); //cerramos nuestro socket
return 0;
}
luiscolo@luiscolo-VirtualBox:~/Desktop/Redes$ clear
```

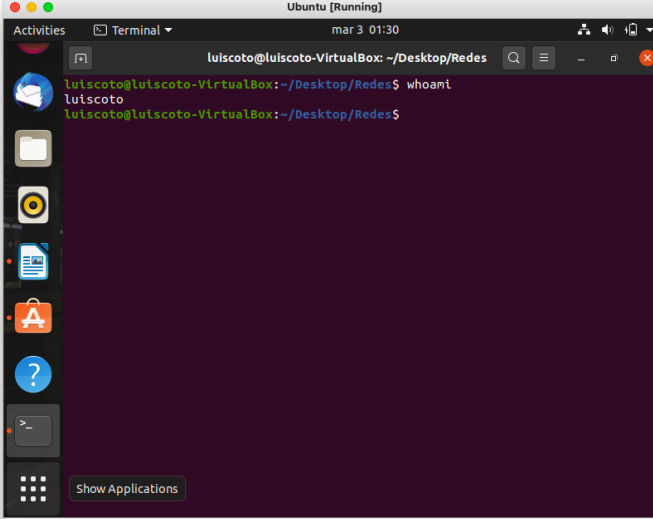




- XIV. **ps**: Nos permite visualizar el estado de los procesos que se estan llevando en ese momento

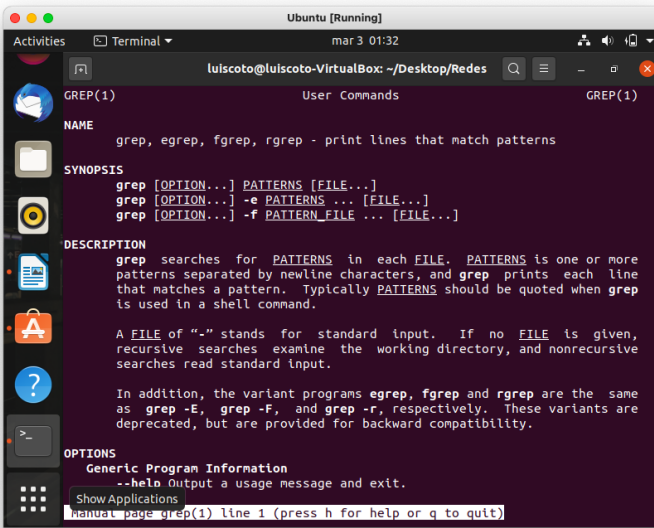


- XV. **whoami:** Es un comando simple, utilizado para imprimir el nombre de usuario efectivo del usuario actual cuando se invoca, que se entiende como el nombre de el usuario en sesión.



```
luisquito@luisquito-VirtualBox: ~/Desktop/Redes$ whoami
luisquito
luisquito@luisquito-VirtualBox: ~/Desktop/Redes$
```

6. Algunos de los comandos anteriores pueden tener una serie de opciones para su ejecución (p.e. ls), utilice nuevamente el comando `man nombre_comando` para ver detalles de información sobre cada uno de los comandos anteriores. Observe las diversas opciones que se utilizan en dichos comandos.



```
GREP(1)                                User Commands                                GREP(1)
NAME
    grep, egrep, fgrep, rgrep - print lines that match patterns
SYNOPSIS
    grep [OPTION...] PATTERNS [FILE...]
    grep [OPTION...] -e PATTERNS ... [FILE...]
    grep [OPTION...] -f PATTERN_FILE ... [FILE...]
DESCRIPTION
    grep searches for PATTERNS in each FILE. PATTERNS is one or more
    patterns separated by newline characters, and grep prints each line
    that matches a pattern. Typically PATTERNS should be quoted when grep
    is used in a shell command.
    A FILE of "-" stands for standard input. If no FILE is given,
    recursive searches examine the working directory, and nonrecursive
    searches read standard input.
    In addition, the variant programs egrep, fgrep and rgrep are the same
    as grep -E, grep -F, and grep -r, respectively. These variants are
    deprecated, but are provided for backward compatibility.
OPTIONS
    Generic Program Information
    --help output a usage message and exit.
    manual page grep(1) line 1 (press h for help or q to quit)
```

- guaje C que  
ello utilice el  
\_salida para

Código del programa

```
#include <stdio.h>

int main(){
    printf("Hola mundo");
}
```

## Ejecución en Ubuntu

```
jomiantc@JomianTC:~/Escritorio/Programas$ gcc Hola.c -o Hola.exe
jomiantc@JomianTC:~/Escritorio/Programas$ ./Hola.exe
Hola mundojomiantc@JomianTC:~/Escritorio/Programas$
```

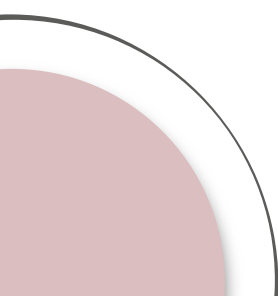
8. Realice un programa que muestre la siguiente salida:

El dato de entrada al programa es el número de asteriscos de la línea horizontal media en la figura con la mayor cantidad de ellos (es decir la figura del rombo), en este caso el dato de entrada es 7 (línea horizontal de asteriscos en rojo). Este dato se pedirá al usuario y podrá ser cualquier valor dentro de un rango que usted establezca, el cual permita construir correctamente la figura completa en la pantalla de la computadora. A partir de este único dato, se construirá la figura con la restricción de que sólo deberá imprimirse un asterisco a la vez. Adicionalmente, dé la opción de guardar la figura en un archivo. Utilice solo bibliotecas estándares de ANSI C. Compile y ejecute su programa.

```

      *           *           *
    **         ***         **
   ***       ****         ***
  ****     *****       ****
*****  *****  *****  *****
***  ***  ***  ***  ***  ***  ***
  ***     *****       ****
    **         ***         **
      *           *           *

```



## Código del programa

```
#include <stdio.h>
#include <stdlib.h>

int main() {

    int a, b, line;
    FILE *f;

    f = fopen("maripoxa.txt","w");

    if(f==NULL){

        printf("no se ha podido abrir \n");
        exit(1);

    }

    printf("elija un numero del 5 hasta 60: ");
    scanf("%d", &line);

    for (a = 1; a <= (line-1)/2; a++){

        for (b = 0; b < (line-1)/2; b++)
            fprintf(f, " ");

        for (b = 1; b <= a; b++)
            fprintf(f, "*");

        for (b = 1; b <= (line+1)-(2*a); b++)
            fprintf(f, " ");

        for (b = 1; b <=a; b++)
            fprintf(f, "*");

        for (b = 1; b <=a-1; b++)
            fprintf(f, "*");

        for (b = 1; b <= (line+1)-(2*a); b++)
            fprintf(f, " ");

        for (b = 1; b <=a; b++)
            fprintf(f, "*");
        fprintf(f, "\n");

    }

    for(a=1; a<= line*3; a++)
        fprintf(f, "*");
    fprintf(f, "\n");

    for(a=1; a<=(line-1)/2; a++){

        for (b = 0; b < (line-1)/2; b++)
            fprintf(f, " ");

        for (b = ((line+1)/2)-1; b >= a; b--)
            fprintf(f, "*");

        if(line % 2 == 0){

            for (b = 1; b <=(2*a)+1; b++)
                fprintf(f, " ");

        }

    }
```

```
    else{

        for (b = 1; b <=2*a; b++)
            fprintf(f, " ");

    }

    for (b = ((line+1)/2)-1; b >= a; b--)
        fprintf(f, "*");

    for (b = ((line+1)/2)-1; b >= a+1; b--)
        fprintf(f, "*");

    if(line % 2 == 0){

        for (b = 1; b <=(2*a)+1; b++)
            fprintf(f, " ");

    }
    else{

        for (b = 1; b <=2*a; b++)
            fprintf(f, " ");

    }

    for (b = ((line+1)/2)-1; b >= a; b--)

        fprintf(f, "*");
        fprintf(f, "\n");

    }

    fclose(f);

    FILE *archivo = fopen("maripoxa.txt", "r");
    int c;

    while ((c = fgetc(archivo))!= EOF){
        putchar(c);
    }

    return 0;
}
```

[illegible]

```

1 | * * *
2 | **          ***          **
3 | ***         *****         ***
4 | ****        ********        ****
5 | *****     *************     *****
6 | ****        ********************     *****
7 | *****     ********************     *****
8 | *****     ********************     *****
9 | *****     ********************     *****
10| *****
11| *****     ********************     *****
12| *****     ********************     *****
13| *****     ********************     *****
14| *****     ********************     *****
15| *****     ********************     *****
16| ****        ********************     ****
17| ***         *****          ***
18| **          ***             **
19| *           *              *

```

9. Realice un programa que calcule la serie de Fibonacci de un número dado como entrada. Compile y ejecute su programa.

Código del programa

```
#include<stdio.h>

int main(){

    int n,x=0,y=1,z=1;
    printf("Digita el numero de elementos para la serie de Fibonacci: \n");
    scanf("%i",&n);
    printf("%i, ",z);

    for (int i = 0; i < n; i++){

        z = x + y;
        x = y;
        y = z ;

        printf(" %d , ", z );
    }

    return 0;
}
```

Ejecución en Ubuntu

```
jomiantc@JomianTC:~/Escritorio/Programas$ gcc Fibonacci.c -o Fibonacci.exe
jomiantc@JomianTC:~/Escritorio/Programas$ ./Fibonacci.exe
Digita el numero de elementos para la serie de Fibonacci:
14
1, 1 , 2 , 3 , 5 , 8 , 13 , 21 , 34 , 55 , 89 , 144 , 233 , 377 , 610 , jomiantc@JomianTC:~/Escritorio/Programas$
```

10. Realice un programa que determine si un grupo de paréntesis están balanceados utilizando pilas. Deberán considerarse tres tipos de paréntesis: ( ), { }, [ ]
- Código del programa

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>

struct nodo {
    char simbolo;
    struct nodo *sig;
};

struct nodo *raiz=NULL;

void insertar(char x){

    struct nodo *nuevo;
    nuevo = (struct nodo *)malloc(sizeof(struct nodo));
    nuevo->simbolo = x;

    if (raiz == NULL){
        raiz = nuevo;
        nuevo->sig = NULL;
    }
    else{
        nuevo->sig = raiz;
        raiz = nuevo;
    }
}

char extraer(){

    if (raiz != NULL){
        char informacion= raiz->simbolo;
        struct nodo *bor = raiz;
        raiz = raiz->sig;
        free(bor);

        return informacion;
    }

    else{
        return -1;
    }
}

void liberar(){

    struct nodo *reco = raiz;
    struct nodo *bor;

    while (reco != NULL){
        bor = reco;
        reco = reco->sig;
        free(bor);
    }
}
```



```
int vacia(){
    if (raiz == NULL)
        return 1;
    else
        return 0;
}

void cargarFormula(char *formula){
    printf("Ingrese la formula:");
    gets(formula);
}

int verificarBalanceada(char *formula){
    int f;
    for (f=0;f<strlen(formula);f++){
        if (formula[f]=='(' || formula[f]=='[' || formula[f]=='{'){
            insertar(formula[f]);
        }
        else{
            if (formula[f]==')'){
                if (extraer()!='('){
                    return 0;
                }
            }
            else{
                if (formula[f]==']'){
                    if (extraer()!='['){
                        return 0;
                    }
                }
                else{
                    if (formula[f]=='}'){
                        if (extraer()!='{'){
                            return 0;
                        }
                    }
                }
            }
        }
    }
}
```

```

        if (vacía()){
            return 1;
        }

        else{
            return 0;
        }
    }

    int main(){

        char formula[100];

        cargarFormula(formula);

        if (verificarBalanceada(formula)){
            printf("La formula esta correctamente balanceada");
        }

        else{
            printf("La formula no esta correctamente balanceada");
        }

        liberar();

        return 0;
    }

```

## Ejecución en Ubuntu

```

jomiantc@JomianTC:~/Escritorio/Programas$ gcc Formula.c -o Formula.exe
Formula.c: In function 'cargarFormula':
Formula.c:78:5: warning: implicit declaration of function 'gets'; did you mean 'fgets'? [-Wimplicit-f
unction-declaration]
   78 |     gets(formula);
      |     ^~~~
      |     fgets
/usr/bin/ld: /tmp/ccphXiJf.o: en la función `cargarFormula':
Formula.c:(.text+0x15a): aviso: the `gets' function is dangerous and should not be used.
jomiantc@JomianTC:~/Escritorio/Programas$ ./Formula.exe
Ingrese la formula: {[()]}
La formula esta correctamente balanceadajomiantc@JomianTC:~/Escritorio/Programas$

```

```

jomiantc@JomianTC:~/Escritorio/Programas$ ./Formula.exe
Ingrese la formula:{5+2[2/(7+2+1)-5]*14}
La formula esta correctamente balanceadajomiantc@JomianTC:~/Escritorio/Programas$

```

11. Realice un programa que evalúe una expresión aritmética delimitada completamente con paréntesis balanceados (por ejemplo:  $((5-2)+4)*10$ ). Compile y ejecute su programa. Restricción: No utilizar transformación a notación prefija o posfija.

Código del programa

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>

char prueba[30], parentesis[10], corchete[10], llaves[10], arreglo[10];
int i = 0, j = 0, f = 0, k = 0, l = 0, m = 0, aux = 1, val1 = 0, val2 = 0;
int val3 = 0, igual = 0, op = 0, total = 0, bad = 0, check = 0;

struct nodo {
};

struct nodo *raiz=NULL;

void insertar(char x){
}

char extraer(){
}

void liberar(){
}

int vacia(){
}

void cargarFormula(char *formula){
}

int verificarBalanceada(char *formula){
}

void circulo(){
    if (prueba[f-1]=='('){
        j = f;
        do{
            parentesis[i] = prueba[j];
            i++;
            j++;
        } while (prueba[j] != ')');
        i = 0;
    }
}

void cuadrado(){
    if (prueba[f-1]=='['){
        j = f;
        do{
            corchete[i] = prueba[j];
            if (prueba[j]=='('){

```

```

void cuadrado(){
    if (prueba[f-1]=='['){
        j = f;
        do{
            corchete[i] = prueba[j];
            if (prueba[j]=='('){
                do{
                    j++;
                } while (prueba[j+1] != ')');
            }
            i++;
            j++;
        } while (prueba[j] != ']');
        i = 0;
    }
}

void pentagono(){
    if (prueba[f-1]=='{'){
        j = f;
        do{
            llaves[i] = prueba[j];
            if (prueba[j]=='['){
                do{
                    j++;
                } while (prueba[j+1] != ']');
            }
            i++;
            j++;
        } while (prueba[j] != '}');
        i = 0;
    }
}

```

```

void operacion (){
    switch (op){
        case 1:
            if (aux == 2){
                igual = val1 + val2;
            }
            if (aux == 3){
                igual = igual + val3;
            }
            aux = 3;
            break;
        case 2:
            if (aux == 2){
                igual = val1 - val2;
            }
            if (aux == 3){
                igual = igual - val3;
            }
            aux = 3;
            break;
        case 3:
            if (aux == 2){
                igual = val1 / val2;
            }
            if (aux == 3){
                igual = igual / val3;
            }
            aux = 3;
            break;
        case 4:
            if (aux == 2){
                igual = val1 * val2;
            }
            if (aux == 3){
                igual = igual * val3;
            }
            aux = 3;
            break;
    }
}

```

```

void insertarEntero(){

    switch(aux){

        case 1:
            val1 = arreglo[f]-'0';
            aux++;
            break;

        case 2:
            val2 = arreglo[f]-'0';
            aux++;
            break;

        case 3:
            val3 = arreglo[f]-'0';
            aux=3;
            break;

    }

}

void comprobarOperacion(){

    if (aux == 3 && val3 == 0 && igual == 0){

        aux = 2;

    }

    if(arreglo[f-1]=='+' && val2 != 0){

        op = 1;
        operacion();

    }

    if(arreglo[f-1]=='-' && val2 != 0){

        op = 2;
        operacion();

    }

    if(arreglo[f-1]=='/' && val2 != 0){

        op = 3;
        operacion();

    }

    if(arreglo[f-1]=='*' && val2 != 0){

        op = 4;
        operacion();

    }

}

```

```

void operacionCirculo(){

    strcpy(arreglo, parentesis);

    for (f = 0; f < strlen(parentesis); f++){

        if (parentesis[f] == '0' || parentesis[f] == '1' ||
            parentesis[f] == '2' || parentesis[f] == '3' ||
            parentesis[f] == '4' || parentesis[f] == '5' ||
            parentesis[f] == '6' || parentesis[f] == '7' ||
            parentesis[f] == '8' || parentesis[f] == '9'){

            insertarEntero();

        }

        comprobarOperacion();

    }

    total = igual;

}

void operacionCuadrado(){

    strcpy(arreglo, corchete);

    for (f = 0; f < strlen(corchete); f++){

        if (corchete[f] == '0' || corchete[f] == '1' ||
            corchete[f] == '2' || corchete[f] == '3' ||
            corchete[f] == '4' || corchete[f] == '5' ||
            corchete[f] == '6' || corchete[f] == '7' ||
            corchete[f] == '8' || corchete[f] == '9'){

            insertarEntero();

        }

        if (corchete[f] == '('){

            if (aux == 3){ val3 = total;    aux = 3;}
            if (aux == 2){ val2 = total;    aux++;}
            if (aux == 1){ val1 = total;    aux++;}
            bad = 1;
            comprobarOperacion();

        }

        if(bad == 0){

            comprobarOperacion();

        }

        bad = 0;

    }

    total = igual;

}

```

```

void operacionPentagono(){

    strcpy(arreglo, llaves);

    for (f = 0; f < strlen(llaves); f++){

        if (llaves[f] == '0' || llaves[f] == '1' ||
            llaves[f] == '2' || llaves[f] == '3' ||
            llaves[f] == '4' || llaves[f] == '5' ||
            llaves[f] == '6' || llaves[f] == '7' ||
            llaves[f] == '8' || llaves[f] == '9'){

            insertarEntero();

        }

        if (llaves[f] == '['){

            if (aux == 3){ val3 = total;    aux = 3;}
            if (aux == 2){ val2 = total;    aux++;}
            if (aux == 1){ val1 = total;    aux++;}
            bad = 1;
            comprobarOperacion();
        }
        if(bad == 0){

            comprobarOperacion();

        }

        bad = 0;

    }

    printf("La operacion es igual a:  %d\n", igual);
    total = igual;

}

void reset(){

    aux = 1;
    op = 1;
    igual = 0;
    val3 = 0;

}

```

```

int main(){

    char formula[100];

    cargarFormula(formula);

    if (verificarBalanceada(formula)){

    }

    else{

    }

    if (check == 1){

        strcpy(prueba, formula);

        for (f = 0; f < strlen(prueba); f++){

            pentagono();

            cuadrado();

            circulo();

        }

        operacionCirculo();

        reset();

        operacionCuadrado();

        reset();

        operacionPentagono();

    }

    liberar();
    return 0;

}

```

Las funciones que fueron minimizadas con el editor de texto son las mismas que el ejercicio anterior fueron cortadas con el fin de reducir el tamaño del archivo

El esquema resumido del código sin la comprobación de balance sería el siguiente

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>

char prueba[30], parentesis[10], corchete[10], llaves[10], arreglo[10];
int i = 0, j = 0, f = 0, k = 0, l = 0, m = 0, aux = 1, val1 = 0, val2 = 0;
int val3 = 0, igual = 0, op = 0, total = 0, bad = 0;

void circulo(){
}

void cuadrado(){
}

void pentagono(){
}

void operacion (){
}

void insertarEntero(){
}

void comprobarOperacion(){
}

void operacionCirculo(){
}

void operacionCuadrado(){
}

void operacionPentagono(){
}

void reset(){
}

int main( ){
    for (f = 0; f < strlen(prueba); f++){
        pentagono();
        cuadrado();
        circulo();
    }

    operacionCirculo();

    reset();

    operacionCuadrado();

    reset();

    operacionPentagono();
}
```



## Ejecución en Ubuntu

```
jomiantc@JomianTC:~/Escritorio/Programas$ gcc Ecuacion.c -o Ecuacion.exe
Ecuacion.c: In function 'cargarFormula':
Ecuacion.c:76:5: warning: implicit declaration of function 'gets'; did you mean 'fgets'? [-Wimplicit-function-declaration]
   76 |     gets(formula);
      |     ^~~~~
/usr/bin/ld: /tmp/ccxk5drq.o: en la función 'cargarFormula':
Ecuacion.c:(.text+0x15a): aviso: the 'gets' function is dangerous and should not be used.
jomiantc@JomianTC:~/Escritorio/Programas$ ./Ecuacion.exe
Ingrese la formula:{5+2-[(7*2)-3]+2}
La formula esta correctamente balanceada
La operacion es igual a: -2
jomiantc@JomianTC:~/Escritorio/Programas$
```

```
jomiantc@JomianTC:~/Escritorio/Programas$ ./Ecuacion.exe
Ingrese la formula:{9*[2+(4*2)-3]+7}
La formula esta correctamente balanceada
La operacion es igual a: 70
jomiantc@JomianTC:~/Escritorio/Programas$
```

12. Guarde sus programas (fuente y ejecutable) en memoria usb.

## SECCIÓN WINDOWS

1. Inicie sesión en Windows.
2. Abra una consola.
3. Investigue para que sirve cada uno de los siguientes comandos y ejecútelos en la consola (tenga cuidado al ejecutar los comandos de borrado de archivos y directorios).
4. Reporte la función de cada uno de los anteriores comandos.

### DIR

El comando dir imprime la unidad en la que estas alojado

El número de serie del volumen de la unidad

Y también dependiendo de la ubicación donde estemos alojados por ejemplo en Desktop nos desplegara todos los archivos que tenemos en dicho directorio incluyendo la fecha en la que fue creado, el tamaño del archivo en bytes, el nombre del archivo además de su extensión

```
C:\Windows\system32\cmd.exe

C:\Users\JomianTC\Desktop>DIR
El volumen de la unidad C es SYSTEM
El número de serie del volumen es: 2417-1823

Directorio de C:\Users\JomianTC\Desktop

02/03/2021  08:42 p. m.    <DIR>          .
02/03/2021  08:42 p. m.    <DIR>          ..
08/05/2020  12:15 p. m.             2,622,754 Leeme.exe
                1 archivos             2,622,754 bytes
                2 dirs  259,810,050,048 bytes libres

C:\Users\JomianTC\Desktop>
```

## IPCONFIG

El comando IPCONFIG imprime todas las conexiones activas en ese momento por ejemplo tenemos la conexión de Ethernet es decir la conexión de internet por cable, las conexiones inalámbricas lan es decir conexiones en local, la conexión con nuestro Wi-Fi que imprime nuestra dirección IPv4, las mascara de subred y la puerta de enlace, también muestra la conexión de la red Bluetooth

```
C:\Windows\system32\cmd.exe

C:\Users\JomianTC\Desktop>IPCONFIG

Configuración IP de Windows

Adaptador de Ethernet Ethernet:

    Estado de los medios. . . . . : medios desconectados
    Sufijo DNS específico para la conexión. . :

Adaptador de LAN inalámbrica Conexión de área local* 1:

    Estado de los medios. . . . . : medios desconectados
    Sufijo DNS específico para la conexión. . :

Adaptador de LAN inalámbrica Conexión de área local* 10:

    Estado de los medios. . . . . : medios desconectados
    Sufijo DNS específico para la conexión. . :

Adaptador de LAN inalámbrica Wi-Fi:

    Sufijo DNS específico para la conexión. . :
    Vínculo: dirección IPv6 local. . . : fe80::d8f2:bd5a:fda7:5a9a%18
    Dirección IPv4. . . . . : 192.168.0.8
    Máscara de subred . . . . . : 255.255.255.0
    Puerta de enlace predeterminada . . . . : 192.168.0.1

Adaptador de Ethernet Conexión de red Bluetooth:

    Estado de los medios. . . . . : medios desconectados
    Sufijo DNS específico para la conexión. . :

C:\Users\JomianTC\Desktop>
```

## CLS

Este comando limpia nuestra ventana de comandos dejando solo la dirección de la carpeta en la que estamos

```
Adaptador de LAN inalámbrica Wi-Fi:

    Sufijo DNS específico para la conexión. . . :
    Vínculo: dirección IPv6 local. . . : fe80::d8f2:bd5a:fda7:5a9a%18
    Dirección IPv4. . . . . : 192.168.0.8
    Máscara de subred . . . . . : 255.255.255.0
    Puerta de enlace predeterminada . . . . . : 192.168.0.1

Adaptador de Ethernet Conexión de red Bluetooth:

    Estado de los medios. . . . . : medios desconectados
    Sufijo DNS específico para la conexión. . . :

C:\Users\JomianTC\Desktop>CLS

C:\Windows\system32\cmd.exe

C:\Users\JomianTC\Desktop>
```

## VER

imprime la versión actual de Microsoft Windows

```
C:\Windows\system32\CMD.exe

C:\Users\JomianTC\Desktop>VER

Microsoft Windows [Versión 10.0.18363.418]

C:\Users\JomianTC\Desktop>
```

## TREE

imprime una estructura de árbol de las carpetas en el directorio donde estamos alojados

```
C:\Windows\system32\CMD.exe

C:\Users\JomianTC\Desktop>TREE
Listado de rutas de carpetas para el volumen SYSTEM
El número de serie del volumen es 2417-1823
C:.
├──Codigo
├──pRACTICA 1
└──PROGRAMAS

C:\Users\JomianTC\Desktop>_
```

## CD - Nombre\_Directorio

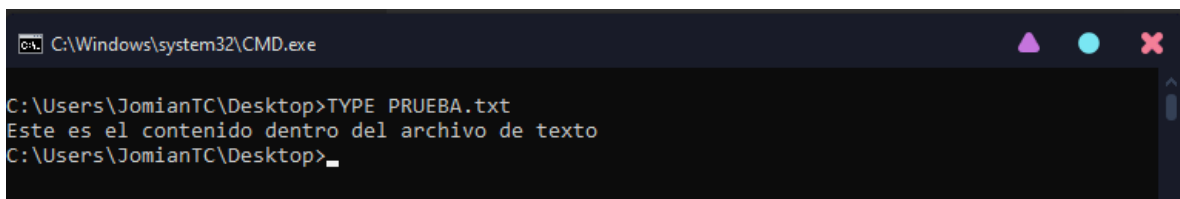
Nos permite acceder a la carpeta del nombre del directorio que hayamos seleccionado



```
C:\Windows\system32\CMD.exe
C:\Users\JomianTC>CD Desktop
C:\Users\JomianTC\Desktop>CD pPROGRAMAS
C:\Users\JomianTC\Desktop\pPROGRAMAS>
```

## TYPE - Nombre\_Archivo

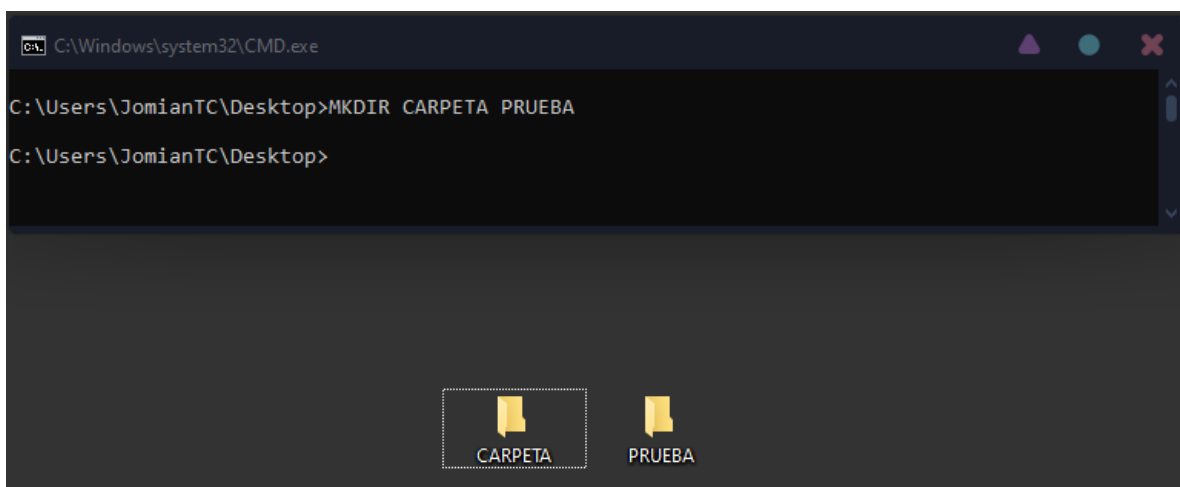
Nos permite ver el contenido del fichero que hayamos seleccionado, siempre y cuando sea un fichero plano (un archivo de texto, por ejemplo)



```
C:\Windows\system32\CMD.exe
C:\Users\JomianTC\Desktop>TYPE PRUEBA.txt
Este es el contenido dentro del archivo de texto
C:\Users\JomianTC\Desktop>_
```

## MKDIR - Nombre\_Directorio

Crea una carpeta en el directorio donde nosotros queramos asignándole un nombre

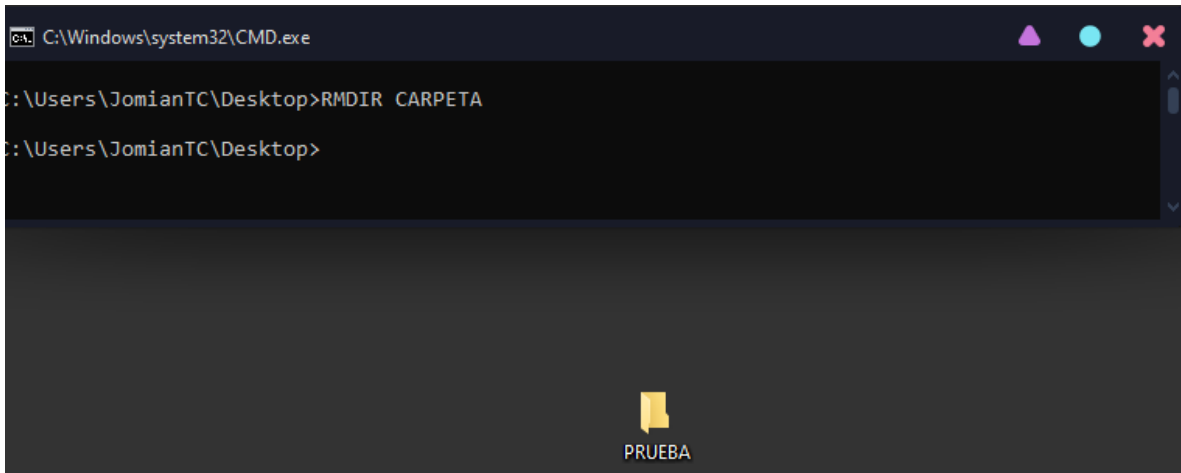


```
C:\Windows\system32\CMD.exe
C:\Users\JomianTC\Desktop>MKDIR CARPETA PRUEBA
C:\Users\JomianTC\Desktop>
```

Below the command prompt, a file explorer view shows the newly created 'CARPETA' and 'PRUEBA' folders on the desktop.

## RMDIR - Nombre\_Directorio

Elimina la carpeta en el directorio que hayamos elegido con su contenido



```
C:\Windows\system32\CMD.exe

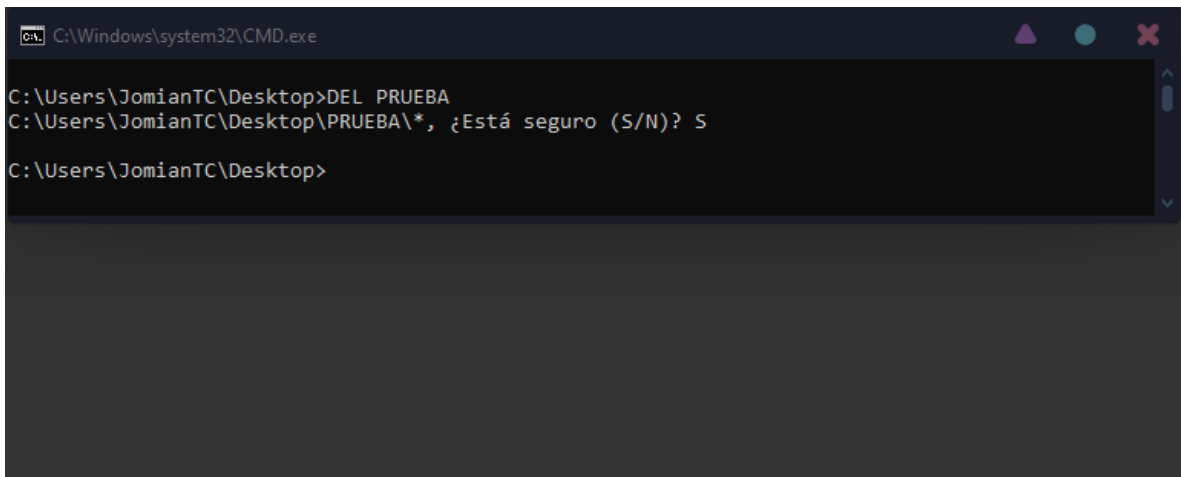
C:\Users\JomianTC\Desktop>RMDIR CARPETA

C:\Users\JomianTC\Desktop>
```

The screenshot shows a Windows Command Prompt window with a dark background. The title bar indicates the path 'C:\Windows\system32\CMD.exe'. The command prompt shows the user is in the directory 'C:\Users\JomianTC\Desktop'. The command 'RMDIR CARPETA' has been entered and executed. Below the command prompt, a folder icon labeled 'PRUEBA' is visible on the desktop.

## DEL - Nombre\_Archivo

Elimina el archivo en el directorio donde estas o el contenido si el nombre que escribiste es una carpeta



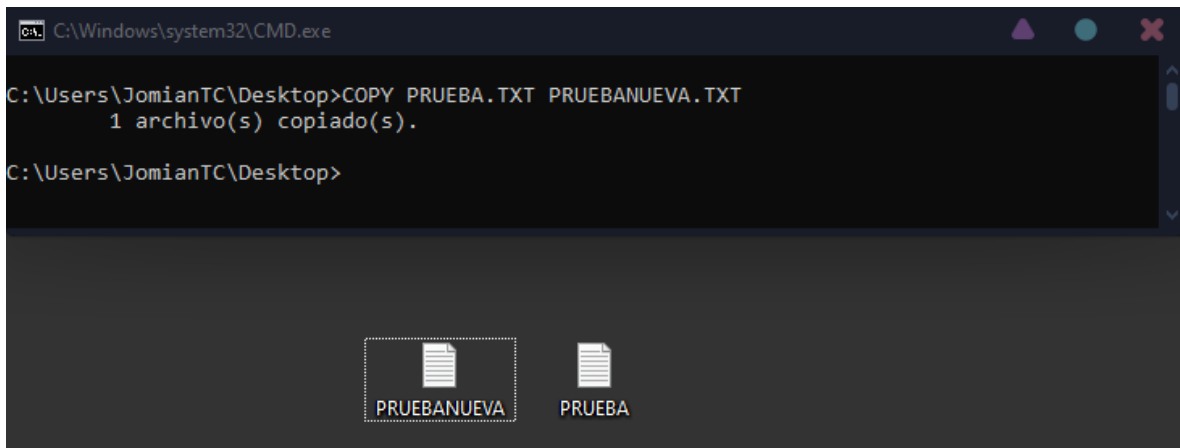
```
C:\Windows\system32\CMD.exe

C:\Users\JomianTC\Desktop>DEL PRUEBA
C:\Users\JomianTC\Desktop\PRUEBA\*, ¿Está seguro (S/N)? S
C:\Users\JomianTC\Desktop>
```

The screenshot shows a Windows Command Prompt window with a dark background. The title bar indicates the path 'C:\Windows\system32\CMD.exe'. The command prompt shows the user is in the directory 'C:\Users\JomianTC\Desktop'. The command 'DEL PRUEBA' has been entered and executed. The prompt then asks for confirmation: 'C:\Users\JomianTC\Desktop\PRUEBA\\*, ¿Está seguro (S/N)?'. The user has responded with 'S' (Yes). The command prompt then returns to the prompt 'C:\Users\JomianTC\Desktop>'.

## **COPY - Nombre\_Archivo Nombre\_NuevoArchivo**

Sirve para copiar un archivo y ponerle un nuevo nombre al archivo copiado

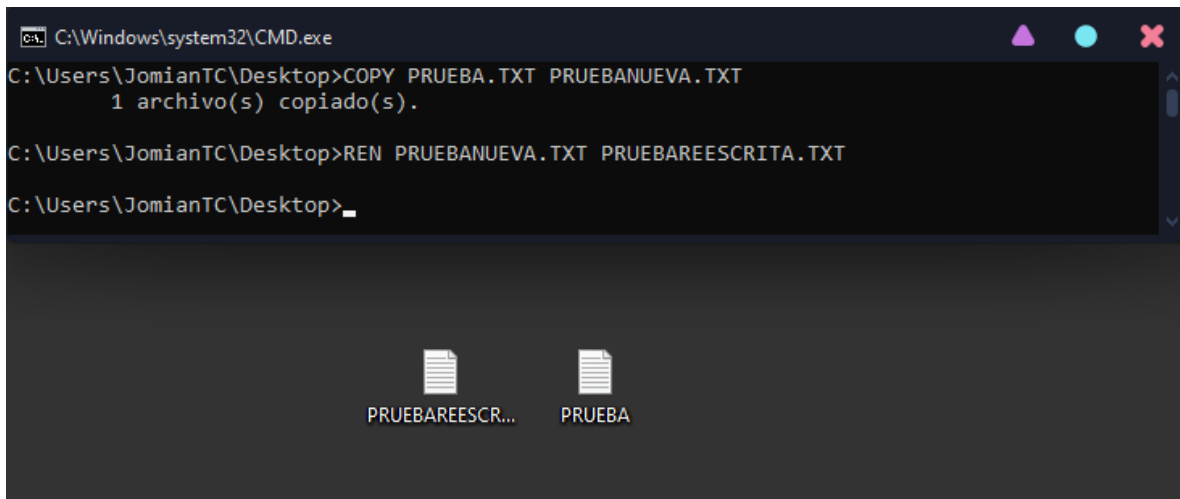


The screenshot shows a Windows Command Prompt window titled "C:\Windows\system32\CMD.exe". The command prompt is open at the directory "C:\Users\JomianTC\Desktop". The user has entered the command "COPY PRUEBA.TXT PRUEBANUEVA.TXT", and the system has responded with "1 archivo(s) copiado(s)". The prompt is now waiting for the next command. Below the command prompt, two file icons are visible on the desktop: "PRUEBANUEVA" and "PRUEBA".

```
C:\Windows\system32\CMD.exe  
  
C:\Users\JomianTC\Desktop>COPY PRUEBA.TXT PRUEBANUEVA.TXT  
1 archivo(s) copiado(s).  
  
C:\Users\JomianTC\Desktop>
```

## **REN - Nombre\_Archivo Nombre\_NuevoArchivo**

Renombra el archivo con un nuevo nombre que nosotros elegimos

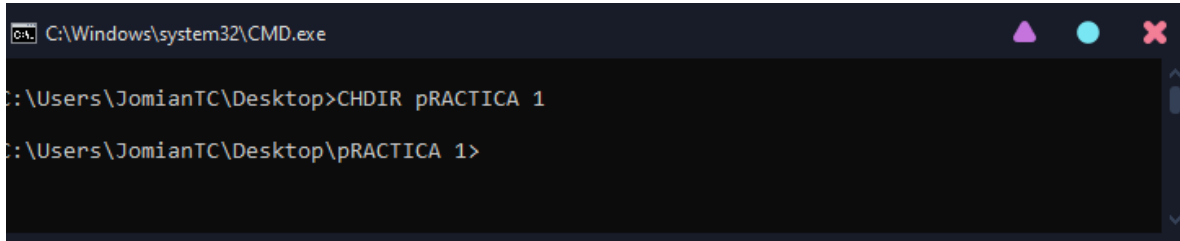


The screenshot shows a Windows Command Prompt window titled "C:\Windows\system32\CMD.exe". The command prompt is open at the directory "C:\Users\JomianTC\Desktop". The user has entered the command "COPY PRUEBA.TXT PRUEBANUEVA.TXT", and the system has responded with "1 archivo(s) copiado(s)". The user then enters the command "REN PRUEBANUEVA.TXT PRUEBAREESCRITA.TXT". The prompt is now waiting for the next command. Below the command prompt, two file icons are visible on the desktop: "PRUEBAREESCR..." and "PRUEBA".

```
C:\Windows\system32\CMD.exe  
  
C:\Users\JomianTC\Desktop>COPY PRUEBA.TXT PRUEBANUEVA.TXT  
1 archivo(s) copiado(s).  
  
C:\Users\JomianTC\Desktop>REN PRUEBANUEVA.TXT PRUEBAREESCRITA.TXT  
  
C:\Users\JomianTC\Desktop>
```

## CHDIR - Nombre\_Directorio

Tiene la misma función que CD permite ir o imprimir la dirección que queremos o estamos respectivamente



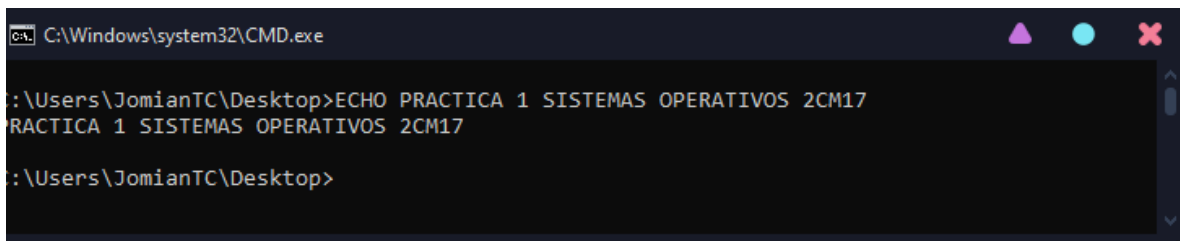
```
C:\Windows\system32\CMD.exe

C:\Users\JomianTC\Desktop>CHDIR pRACTICA 1

C:\Users\JomianTC\Desktop\pRACTICA 1>
```

## ECHO - Cadena

Imprime la cadena que nosotros escribimos después del comando o también lo podemos desactivar para quitar nuestra barra de dirección actual



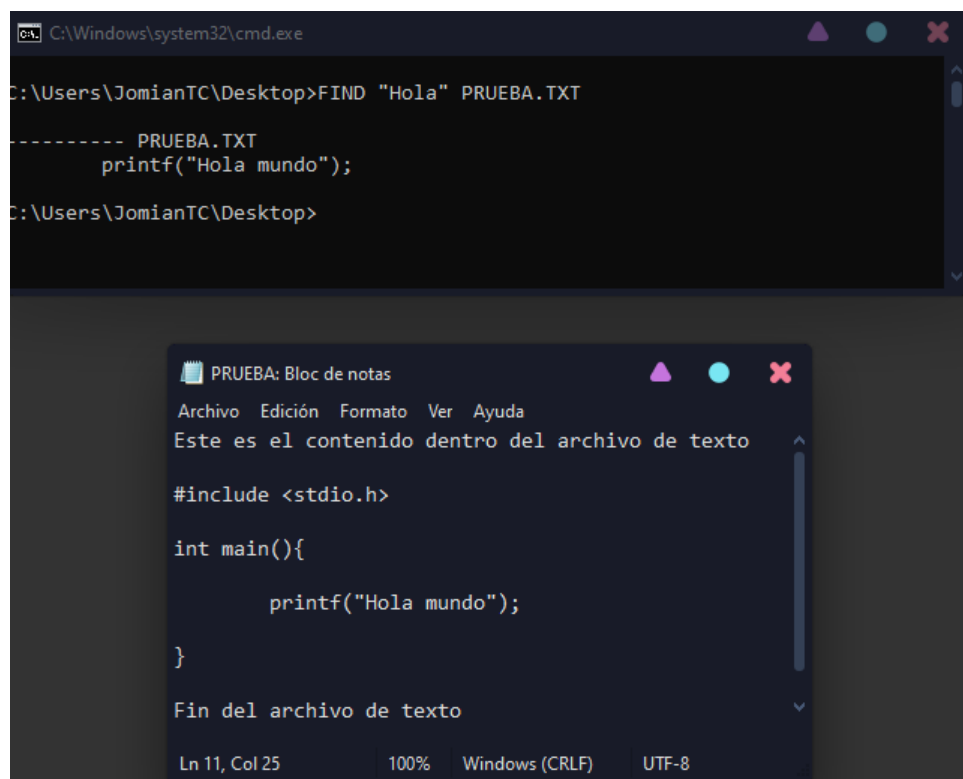
```
C:\Windows\system32\CMD.exe

C:\Users\JomianTC\Desktop>ECHO PRACTICA 1 SISTEMAS OPERATIVOS 2CM17
PRACTICA 1 SISTEMAS OPERATIVOS 2CM17

C:\Users\JomianTC\Desktop>
```

## FIND - Cadena\_Buscar Nombre\_Archivo

Imprime la línea en la cual se encuentra el texto que nosotros queremos buscar en el archivo que especificamos



```
C:\Windows\system32\cmd.exe

C:\Users\JomianTC\Desktop>FIND "Hola" PRUEBA.TXT

----- PRUEBA.TXT
      printf("Hola mundo");

C:\Users\JomianTC\Desktop>
```

PRUEBA: Bloc de notas

Archivo Edición Formato Ver Ayuda

Este es el contenido dentro del archivo de texto

```
#include <stdio.h>

int main(){

    printf("Hola mundo");

}

Fin del archivo de texto
```

Ln 11, Col 25 100% Windows (CRLF) UTF-8

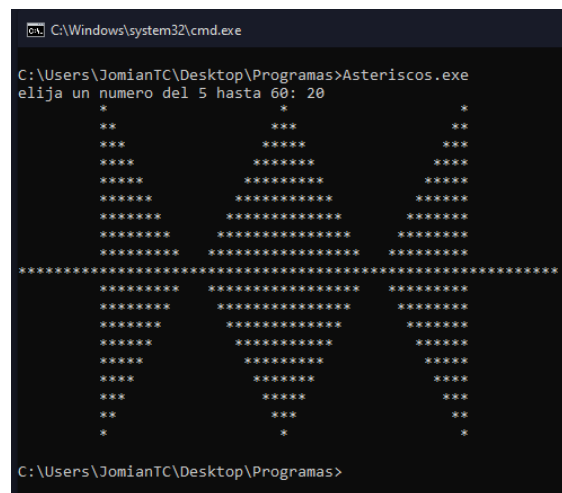


5. Abra la carpeta de su memoria usb donde tiene los programas que desarrolló en la sección de Linux y ejecútelos. Dé sus observaciones sobre la ejecución de los mismos.
6. Busque el directorio donde está instalado Dev C.
7. Desde la consola ubíquese en el directorio de Dev C, y cambie al directorio "bin"
8. Desde ese directorio compile en la consola como se indica en el punto 10 de la sección Linux cada uno de los programas creados en esa sección y que guardó en la memoria usb, ejecútelos y observe su funcionamiento.

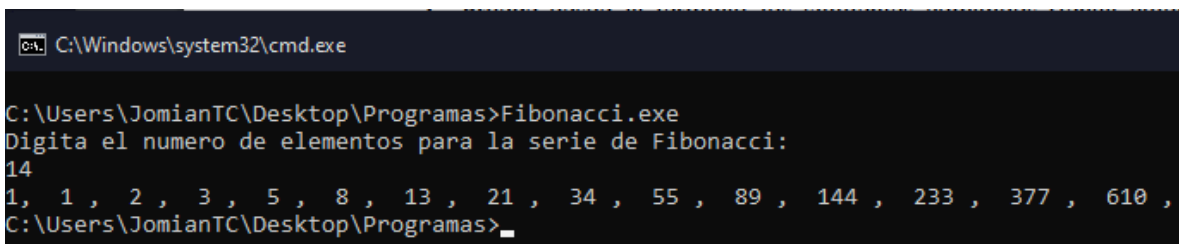
### Programa Hola mundo

```
C:\Users\JomianTC\Desktop\Programas>Hola.exe
Hola mundo
C:\Users\JomianTC\Desktop\Programas>
```

### Programa Asteriscos



### Programa Fibonacci



## Programa Balance de corchetes

```
C:\Windows\system32\cmd.exe

C:\Users\JomianTC\Desktop\Programas>Formula.exe
Ingrese la formula: {[()]}
La formula esta correctamente balanceada
C:\Users\JomianTC\Desktop\Programas>_
```

```
C:\Windows\system32\cmd.exe - Formula.exe

C:\Users\JomianTC\Desktop\Programas>Formula.exe
Ingrese la formula:{5+2[2/(7+2+1)-5]*14}
La formula esta correctamente balanceada_
```

## Programa Expresiones aritméticas

```
C:\Windows\system32\cmd.exe - Ecuacion.exe

C:\Users\JomianTC\Desktop\Programas>gcc Ecuacion.c -o Ecuacion

C:\Users\JomianTC\Desktop\Programas>Ecuacion.exe
Ingrese la formula:{5+2-[(7*2)-3]+2}
La formula esta correctamente balanceada
La operacion es igual a: -2
```

```
C:\Windows\system32\cmd.exe

C:\Users\JomianTC\Desktop\Programas>Ecuacion.exe
Ingrese la formula:{9*[2+(4*2)-3]+7}
La formula esta correctamente balanceada
La operacion es igual a: 70

C:\Users\JomianTC\Desktop\Programas>_
```

9. Reporte las diferencias y similitudes entre los comandos de Linux y Windows, así como la compilación y ejecución de los programas realizados.

Como pudimos observar las diferencias entre estos sistemas en cuanto a la compilación y ejecución de los programas anteriormente escritos no fue nada diferente a excepción de que la compilación en Ubuntu dio mensajes de avisos por funciones que no pertenecían al compilador de C si no al de C++ pero que de igual forma funcionan en C sin ningún problema, nada mas resulto diferente, los programas arrojaron los mismos resultados en igualdad de circunstancias, además de que en Ubuntu es mas sencillo compilar ya que tiene un compilador integrado para lenguaje C cosa que en Windows no, en cuanto los comandos sus objetivos son los mismo pero la forma de escribirlos es muy diferente, por la costumbre de usar Windows como sistema operativo principal al momento de escribir estos comandos es algo nuevo y tuvimos muchos fallos de escritura al momento de ejecutarlos, pero no hay mucha diferencia en los comandos con excepción de la sintaxis que como es de gran obviedad es muy distinta siendo ambas igualmente fáciles de aprender y de utilizar

## **CONCLUSIONES**

Me parece que la practica asignada presento un buen reto para el equipo y a nivel personal, pues me hizo volver a temas que había dejado un poco abandonados por cuestiones académicas de tener que comenzar a manejar otro tipo de lenguajes de programación, de igual forma esta practica me hizo darme cuenta del enorme panorama que hay presente con Linux así como la gran diferencia que tiene con Windows, pues personalmente he estado trabajando siempre con Windows y ya estoy mucho más familiarizado con dicho Sistema Operativo, aunque me agrado mucho como es que Linux te proporciona un manual integrado en su misma consola que te proporciona información acerca de los distintos comandos que es posible usar, la realización del código presentó en ciertos momentos algunas trabas, principalmente de lógica, lo cual hizo que el equipo se esforzara aún más para poder proporcionar la salida esperada y por último la lectura de los documentos proporcionados para la elaboración de los mapas me pareció de suma utilidad, pues me proporcionan más conocimiento acerca del equipo con el cual trabajo día a día y el cual en un futuro seguirá siendo mi principal herramienta de trabajo, es por eso que me parece muy importante tener conocimiento del funcionamiento interno que este posee.

***-Mora Ayala José Antonio***

Existe una gran variedad de sistemas operativos en la actualidad que nos facilitan el aprovechamiento de la computadora, la práctica nos permitió visualizar las diferencias en dos de ellos, windows y linux, en mi caso particular llevo tiempo sin relacionarme con windows y ocupando linux de una manera un tanto más seguido, pero centrandome principalmente en macOS; los anteriores ejercicios me obligaron a desempolvar todo y visualizar cómo se realizan los procesos que podemos visualizar de maneras sencillas, directo desde la terminal con los comandos correspondientes. A pesar de estar trabajando paralelamente y de la preferencia común por windows, la capacidad que tiene linux para configurar el sistema a las necesidades del usuario me parece increíble. La práctica y la lectura del libro me brindó mucho aprendizaje y despertó un enorme interés en continuar investigando y probando diversas distribuciones de los sistemas operativos, por mi cuenta.

***-Ramírez Cotonieto Luis Fernando***

La practica me ayudo mucho a descubrir mejor lo diferentes que son los sistemas operativos entre si además de que también tuve la oportunidad de instalar probar y ejecutar programas en Linux en el OS Ubuntu, es un sistema muy distinto eficiente y veloz a la hora de ejecutar aplicaciones y programas sin tener problemas de rendimiento también observe que como es puramente un OS de código abierto todos los programas que tiene instalados por defecto lo son, haciendo que todo el contenido de sistema sea gratuito, a la hora de ejecutar los comandos me pudo dar cuenta de que no es muy diferente de Windows en esencia, escribimos el comando con sus parámetros si los necesitas y los ejecuta al pie de la letra al igual que Windows, teniendo ambos sus pros y contras frente el uno al otro, sabiendo el potencial de Ubuntu tiene muchas posibilidades de volverse mi OS favorito

***-Torres Carrillo Josehf Miguel Angel***

Después de realizar esta práctica me doy cuenta de las grandes funciones que tienen los diversos sistemas operativos que existen y para que suele ser más ocupado cada uno, por ejemplo en mi situación personal muy pocas veces había tenido acercamiento a Linux porque no me terminaba de acostumbrar a sus comando y a su modo de manejo, pero con esta práctica e investigando sus funciones me resulta un poco más cómodo su manejo, aunque aún no se todo al cien por ciento se me hace una muy buena herramienta ya que me da más herramientas que las que me proporciona Windows. Y después de lograr acoplarme al nuevo sistema operativo presento un nuevo reto recordar los temas de base de datos, por lo que tuve que dar un repaso para lograr saber cómo resolver los programas, tal vez no todos se lograron hacer de la manera más eficiente, pero ayudo a recordar cómo es que se ocupan las estructuras y la implementación en C

***-Tovar Jacuinde Rodrigo***