



**Universidad Nacional
Autónoma de México**
Facultad de Ingeniería
División de Ciencias Básicas



Fundamentos de Programación (1122)

Profesor: M.I. Marco Antonio Martínez Quintana
Semestre 2021-1

Practica No. 02

GNU/Linux

Grupo: 1129

No. de Equipo de cómputo empleado: No aplica

No. de Lista o Brigada: No aplica

No. de Lista: 42

Nombre: Adolfo Román Jiménez

GNU/Linux

OBJETIVO:

Conocer la importancia del sistema operativo de una computadora, así como sus funciones. Explorar un sistema operativo GNU/Linux con el fin de conocer y utilizar los comandos básicos en GNU/Linux.

INTRODUCCIÓN:

El Sistema Operativo es el conjunto de programas y datos que administra los recursos tanto de hardware (dispositivos) como de software (programas y datos) de un sistema de cómputo y/o comunicación. Y ayuda a la comunicación de usuario con computadora.

Existen diversos sistemas operativos; para equipos de cómputo están Windows, Linux, Mac OS entre otros. Para el caso de dispositivos móviles se encuentran Android, IOS, Windows Phone entre otros.

Sistema Operativo Linux

Linux es un sistema operativo tipo Unix de libre distribución para computadoras personales, servidores y estaciones de trabajo.

El sistema está conformado por el núcleo (kernel) y un gran número de programas y bibliotecas. Muchos programas y bibliotecas han sido posibles gracias al proyecto GNU, por lo mismo, se conoce a este sistema operativo como GNU/Linux.

Software libre

Un software libre es aquel que se puede adquirir de manera gratuita, es decir, no se tiene que pagar algún tipo de licencia a alguna casa desarrolladora de software por el uso del mismo.

Además, que un software sea libre implica también que el software viene acompañado del código fuente, es decir, se pueden realizar cambios en el funcionamiento del sistema si así se desea.

Linux se distribuye bajo la Licencia Pública General de GNU por lo tanto, el código fuente tiene que estar siempre accesible y cualquier modificación o trabajo derivado debe tener esta licencia.

Licencia GNU

La Licencia Pública General de GNU o GNU General Public License (GNU GPL) es una licencia creada por la Free Software Foundation en 1989 y está orientada principalmente a proteger la libre distribución, modificación y uso de software.

Su propósito es declarar que el software cubierto por esta licencia es software libre y protegerlo de intentos de apropiación que restrinjan esas libertades a los usuarios.

Kernel de GNU/Linux

El kernel o núcleo de linux se puede definir como el corazón del sistema operativo. Es el encargado de que el software y el hardware del equipo se puedan comunicar. Sus componentes son los que se mencionaron en la introducción de esta práctica.



Figura 1: Capas que componen al sistema operativo GNU/Linux.

De la figura 1, se puede observar que entre el kernel y las aplicaciones existe una capa que permite al usuario comunicarse con el sistema operativo y en general con la computadora, a través de programas que ya vienen instalados con la distribución de Linux (Debian, Ubuntu, Fedora, etc.) y trabajan ya sea en modo gráfico o en modo texto. Uno de estos programas es el Shell.

La estructura de Linux para el almacenamiento de archivos es de forma jerárquica; por lo que la carpeta o archivo base es "root" (raíz) la cual se representa con una diagonal (/). De este archivo raíz, parten todos los demás. Los archivos pueden ser carpetas (directorios), de datos, aplicaciones, programas, etc.

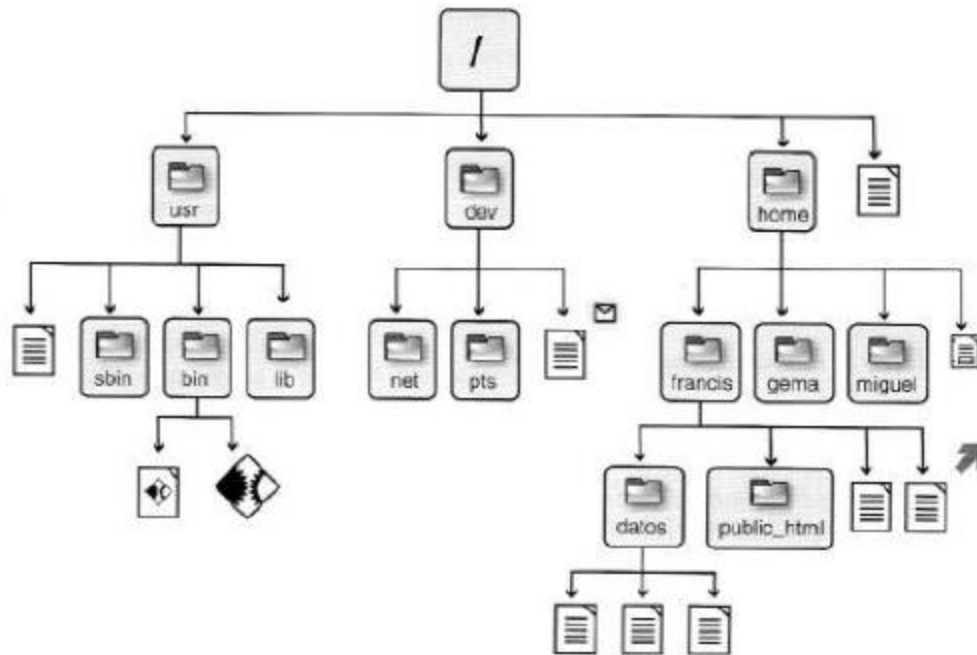


Figura 2: Una parte del sistema de archivos jerárquico en GNU/Linux.

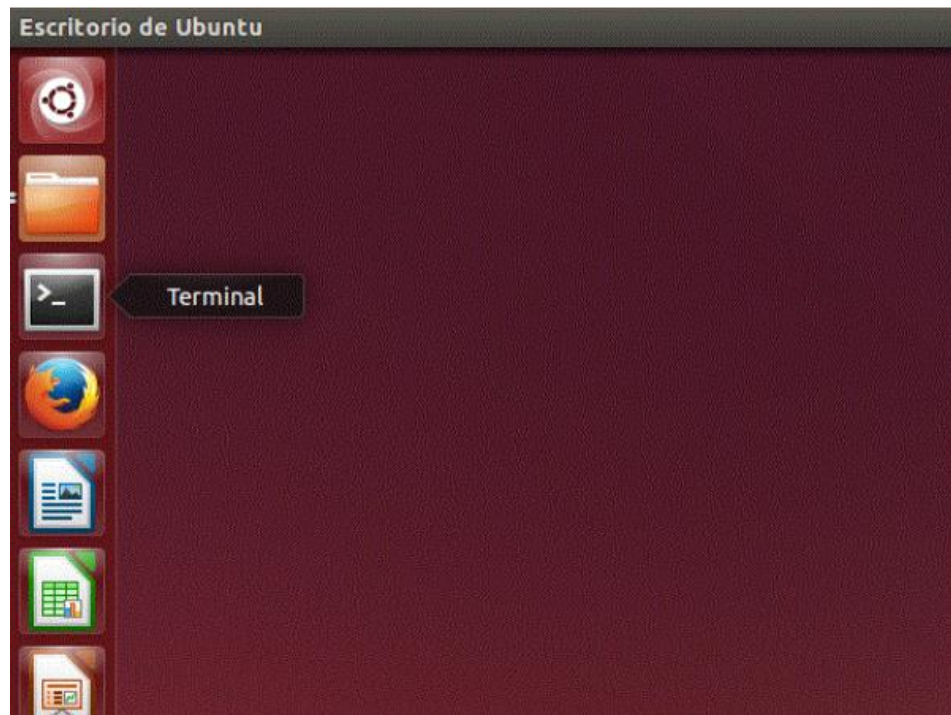
Interfaz de línea de comandos (CLI) o shell de GNU/Linux

El Shell de GNU/Linux permite introducir órdenes (comandos) y ejecutar programas en el sistema operativo. Todas las órdenes de UNIX/Linux son programas que están almacenados en el sistema de archivos y a los que llamamos comandos, por lo tanto, todo en GNU/Linux se puede controlar mediante comandos.

Comandos básicos

Para trabajar en Linux utilizando comandos, se debe abrir una “terminal” o “consola” que es una ventana donde aparece la “línea de comandos” en la cual se escribirá la orden o comando. La terminal permite un mayor grado de funciones y configuración de lo que queremos hacer con una aplicación o acción en general respecto a un entorno gráfico.

El proceso de abrir una terminal varía dependiendo del entorno gráfico. Por lo general hay un área de “aplicaciones” donde se selecciona terminal o consola.



O bien en el ícono de aplicaciones en la línea de “buscar” escribir “terminal” si es que no está a la vista el ícono de terminal.



Una vez teniendo una terminal abierta, estamos listos para introducir comandos.



La sintaxis que siguen los comandos es la siguiente:

comando [-opciones] [argumentos]

Esto es, el nombre del comando, seguido de algunas banderas (opciones) para modificar la ejecución del mismo y, al final, se puede incluir un argumento (ruta, ubicación, archivo, etcétera) dependiendo del comando. Tanto las opciones como los argumentos son opcionales.

Los componentes de un sistema operativo, de forma general, son:

- Gestor de memoria,
- Administrador y planificador de procesos,
- Sistema de archivos y
- Administración de E/S.

Fedora Linux – Terminal

Comandos:

- `ls`

(muestra los archivos dentro del folder donde se ubica)

El comando `ls` permite listar los elementos que existen en alguna ubicación del sistema de archivos de Linux. Por defecto lista los elementos que existen en la ubicación actual; Linux nombra la ubicación actual con un punto (.) por lo que

```
ls
```

y

```
ls .
```

realizan exactamente lo mismo.

El comando `ls` realiza acciones distintas dependiendo de las banderas que utilice, por ejemplo, si se utiliza la opción `l` se genera un listado largo de la ubicación actual:

```
ls -l
```

Es posible listar los elementos que existen en cualquier ubicación del sistema de archivos, para ello hay que ejecutar el comando especificando como argumento la ubicación donde se desean listar los elementos. Si queremos ver los archivos que se encuentran en la raíz, usamos:

```
ls /
```

Para ver los usuarios del equipo local, revisamos el directorio *home* que parte de la raíz (/):

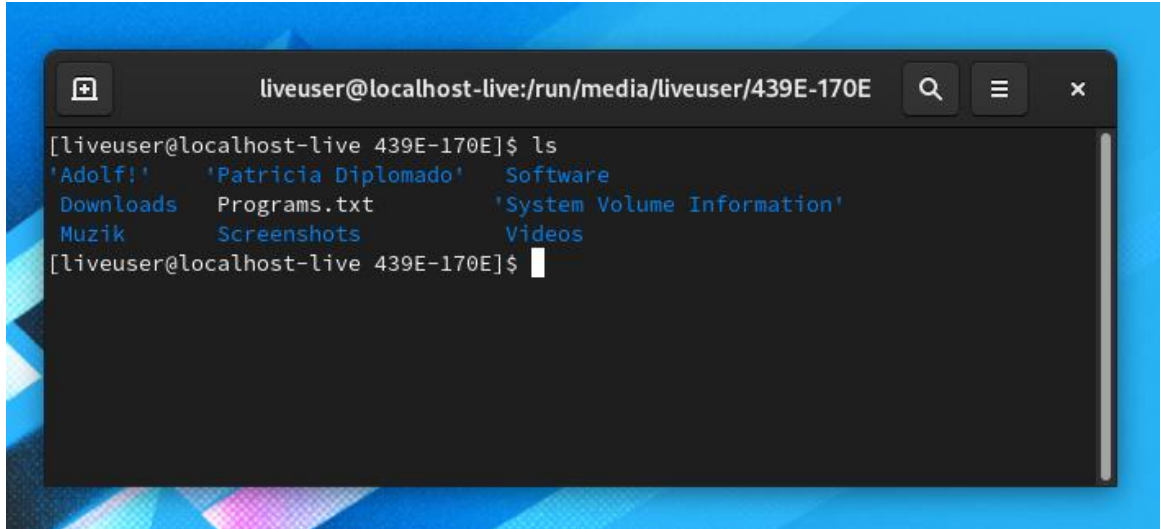
```
ls /home
```

Tanto las opciones como los argumentos se pueden combinar para generar una ejecución más específica:

```
ls -l /home
```

GNU/Linux proporciona el comando *man*, el cual permite visualizar la descripción de cualquier comando así como la manera en la que se puede utilizar.

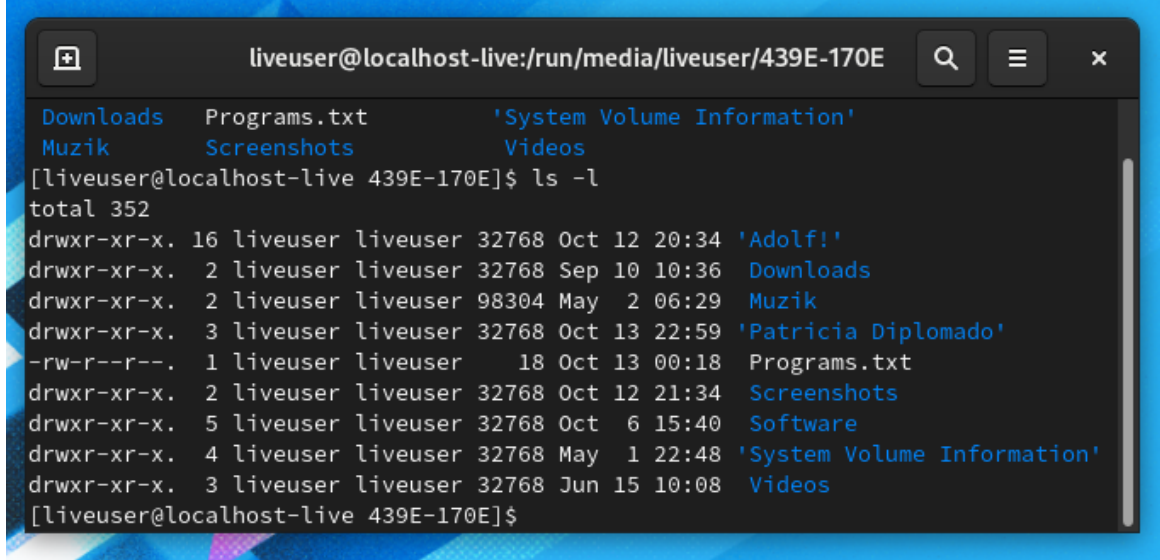
```
man ls
```

```
liveuser@localhost-live:/run/media/liveuser/439E-170E
[liveuser@localhost-live 439E-170E]$ ls
'Adolf!'      'Patricia Diplomado'  Software
Downloads    Programs.txt          'System Volume Information'
Muzik        Screenshots           Videos
[liveuser@localhost-live 439E-170E]$
```

- `ls -l`

(hace un listado con detalles del de los archivos dentro del folder donde se ubica)



```
liveuser@localhost-live:/run/media/liveuser/439E-170E
[liveuser@localhost-live 439E-170E]$ ls -l
total 352
drwxr-xr-x. 16 liveuser liveuser 32768 Oct 12 20:34 'Adolf!'
drwxr-xr-x.  2 liveuser liveuser 32768 Sep 10 10:36 Downloads
drwxr-xr-x.  2 liveuser liveuser 98304 May  2 06:29 Muzik
drwxr-xr-x.  3 liveuser liveuser 32768 Oct 13 22:59 'Patricia Diplomado'
-rw-r--r--.  1 liveuser liveuser   18 Oct 13 00:18 Programs.txt
drwxr-xr-x.  2 liveuser liveuser 32768 Oct 12 21:34 Screenshots
drwxr-xr-x.  5 liveuser liveuser 32768 Oct  6 15:40 Software
drwxr-xr-x.  4 liveuser liveuser 32768 May  1 22:48 'System Volume Information'
drwxr-xr-x.  3 liveuser liveuser 32768 Jun 15 10:08 Videos
[liveuser@localhost-live 439E-170E]$
```

- `ls /`

(muestra archivos de la carpeta *root* desde cualquier ubicación)


```
liveuser@localhost-live:/run/media/liveuser/439E-170E  🔍  ≡  ✕  
drwxr-xr-x.  5 liveuser liveuser 32768 Oct  6 15:40  Software  
drwxr-xr-x.  4 liveuser liveuser 32768 May  1 22:48 'System Volume Information'  
drwxr-xr-x.  3 liveuser liveuser 32768 Jun 15 10:08  Videos  
[liveuser@localhost-live 439E-170E]$ ls /  
bin  dev  home  lib64      media  opt   root  sbin  sys  usr  
boot etc  lib   lost+found  mnt    proc  run   srv   tmp   var  
[liveuser@localhost-live 439E-170E]$
```

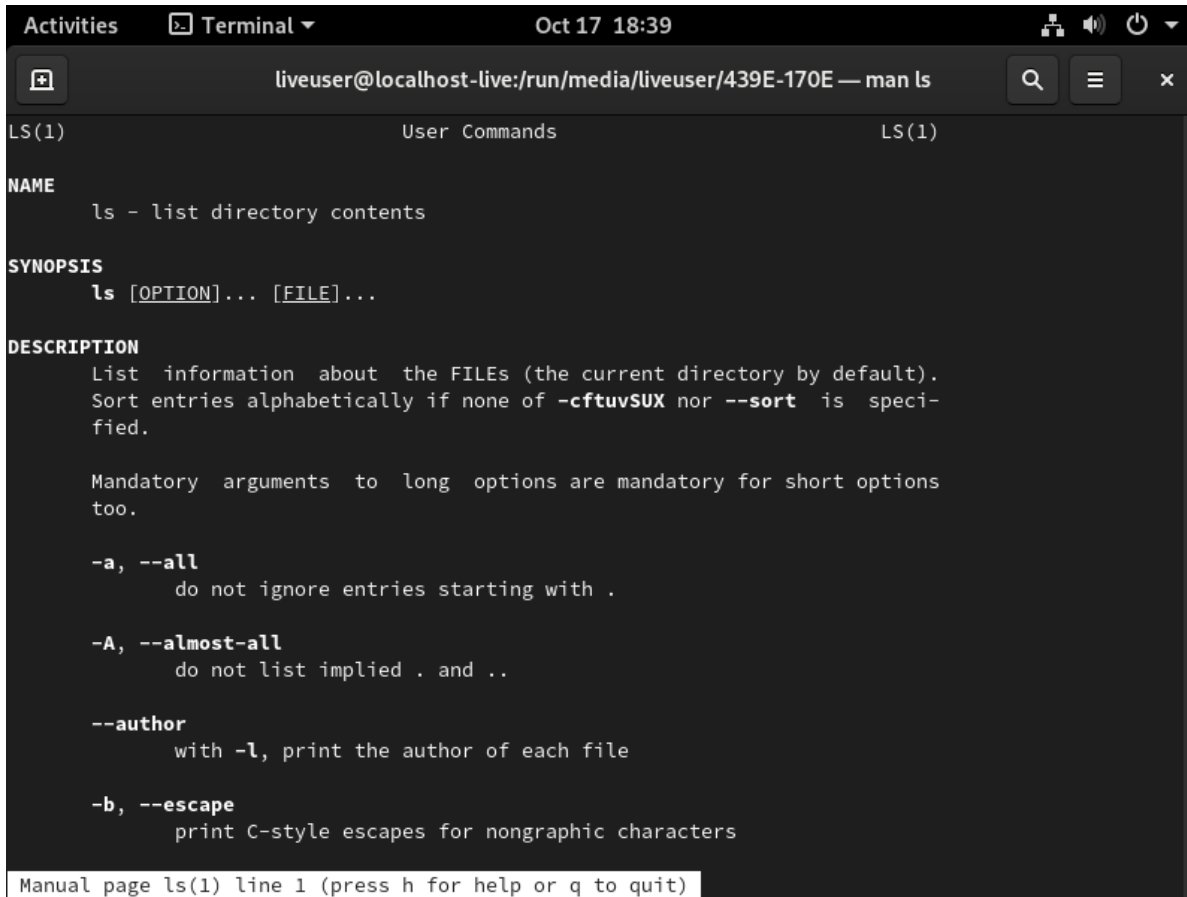
- *ls /home*

(muestra archivos de la carpeta home en *root* desde cualquier ubicación)

```
liveuser@localhost-live:/run/media/liveuser/439E-170E  🔍  ≡  ✕  
drwxr-xr-x.  3 liveuser liveuser 32768 Jun 15 10:08  Videos  
[liveuser@localhost-live 439E-170E]$ ls /  
bin  dev  home  lib64      media  opt   root  sbin  sys  usr  
boot etc  lib   lost+found  mnt    proc  run   srv   tmp   var  
[liveuser@localhost-live 439E-170E]$ ls /home  
liveuser  
[liveuser@localhost-live 439E-170E]$
```

- *man ls*

(El comando *man* seguido del argumento, describe las opciones del comando usado como argumento)



The screenshot shows a terminal window titled "liveuser@localhost-live:/run/media/liveuser/439E-170E — man ls". The terminal displays the manual page for the `ls` command. The content is as follows:

```
LS(1)                                User Commands                                LS(1)

NAME
    ls - list directory contents

SYNOPSIS
    ls [OPTION]... [FILE]...

DESCRIPTION
    List information about the FILES (the current directory by default).
    Sort entries alphabetically if none of -cftuvSUX nor --sort is speci-
    fied.

    Mandatory arguments to long options are mandatory for short options
    too.

    -a, --all
        do not ignore entries starting with .

    -A, --almost-all
        do not list implied . and ..

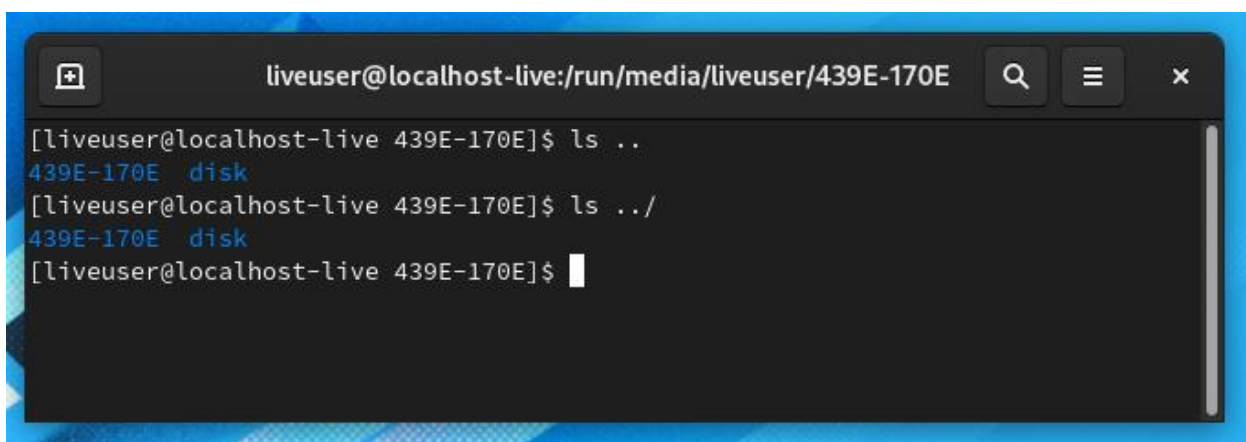
    --author
        with -l, print the author of each file

    -b, --escape
        print C-style escapes for nongraphic characters

Manual page ls(1) line 1 (press h for help or q to quit)
```

- *ls .. o ls ../*

(Muestra los archivos del directorio padre donde la carpeta se encuentra ubicada)



The screenshot shows a terminal window titled "liveuser@localhost-live:/run/media/liveuser/439E-170E". The terminal displays the following commands and output:

```
[liveuser@localhost-live 439E-170E]$ ls ..
439E-170E disk
[liveuser@localhost-live 439E-170E]$ ls ../
439E-170E disk
[liveuser@localhost-live 439E-170E]$
```

- `ls ../../../../etc`

(Nos permite formar la lista de un directorio padre de forma relativa al directorio donde nos encontramos)

```

[liveuser@localhost-live 439E-170E]$ ls ../../../../run
alsactl.pid      firewallld      lvm             ppp             systemd
auditd.pid       gdm             mcelog-client  pptp            tmpfiles.d
avahi-daemon     gluster        mcelog.pid     radvd           tpm2-tss
blkid            gssproxy.pid   mdadm          rpcbind         udev
chrony           gssproxy.sock  media          rpcbind.sock    udisks2
console          hplip          motd           rpc.statd.pid   user
criu             httpd          motd.d         samba           utmp
cryptsetup       initctl        mount          sepermit        vmware
cups            initramfs     NetworkManager setrans         vpnc
dbus            libgpod       openvpn-client sm-notify.pid   xtables.lock
dmeventd-client  libvirt       openvpn-server spice-vdagentd
dmeventd-server  lock          plymouth      sssd.pid
faillock         log           podman        sudo

[liveuser@localhost-live 439E-170E]$ ls /run
alsactl.pid      firewallld      lvm             ppp             systemd
auditd.pid       gdm             mcelog-client  pptp            tmpfiles.d
avahi-daemon     gluster        mcelog.pid     radvd           tpm2-tss
blkid            gssproxy.pid   mdadm          rpcbind         udev
chrony           gssproxy.sock  media          rpcbind.sock    udisks2
console          hplip          motd           rpc.statd.pid   user
criu             httpd          motd.d         samba           utmp
cryptsetup       initctl        mount          sepermit        vmware
cups            initramfs     NetworkManager setrans         vpnc
dbus            libgpod       openvpn-client sm-notify.pid   xtables.lock
dmeventd-client  libvirt       openvpn-server spice-vdagentd
dmeventd-server  lock          plymouth      sssd.pid
faillock         log           podman        sudo

```

- `touch`

El comando `touch` permite crear un archivo de texto, su sintaxis es la siguiente:

```
touch nombre_archivo[.ext]
```

En GNU/Linux no es necesario agregar una extensión al archivo creado, sin embargo, es recomendable hacerlo para poder identificar el tipo de archivo creado.

(Aunque en la práctica dice que el comando `touch` crea archivos de texto, en el internet especifica que únicamente genera archivos que no tienen información y que no necesariamente son archivos que terminan en `.txt`)

```
liveuser@localhost-live:/run/media/liveuser/439E-170E
dmeventd-client  libvirt      openvpn-server  spice-vdagentd
dmeventd-server  lock         plymouth        sssd.pid
faillock         log          podman          sudo
[liveuser@localhost-live 439E-170E]$ man touch
[liveuser@localhost-live 439E-170E]$ touch adolf
[liveuser@localhost-live 439E-170E]$ ls
adolfo  Downloads  'Patricia Diplomado'  Screenshots  'System Volume Informa
tion'
'Adolfo!'  Muzik      Programs.txt          Software      Videos
[liveuser@localhost-live 439E-170E]$
```

- *mkdir*

El comando *mkdir* permite crear una carpeta, su sintaxis es la siguiente:

```
mkdir nombre_carpeta
```

Para crear una carpeta en nuestra cuenta, que tenga como nombre “tarefas” se escribe el siguiente comando:

```
mkdir tareas
```

(Genera un directorio en la ubicación)

```
liveuser@localhost-live:/run/media/liveuser/439E-170E
'Adolfo!'  Muzik      Programs.txt          Software      Videos
[liveuser@localhost-live 439E-170E]$ mkdir Practica
[liveuser@localhost-live 439E-170E]$ ls
adolfo  Muzik      Programs.txt  'System Volume Information'
'Adolfo!'  'Patricia Diplomado'  Screenshots  Videos
Downloads  Practica          Software
[liveuser@localhost-live 439E-170E]$
```

- `cd`

(Significa “*change directory*” y permite ingresar a una carpeta deseada utilizando `cd` (nombre de carpeta) en la ubicación del directorio, cuando se quiere ir a la carpeta padre en el primer nivel se puede utilizar el comando `cd ..`)

El comando `cd` permite ubicarse en una carpeta, su sintaxis es la siguiente:

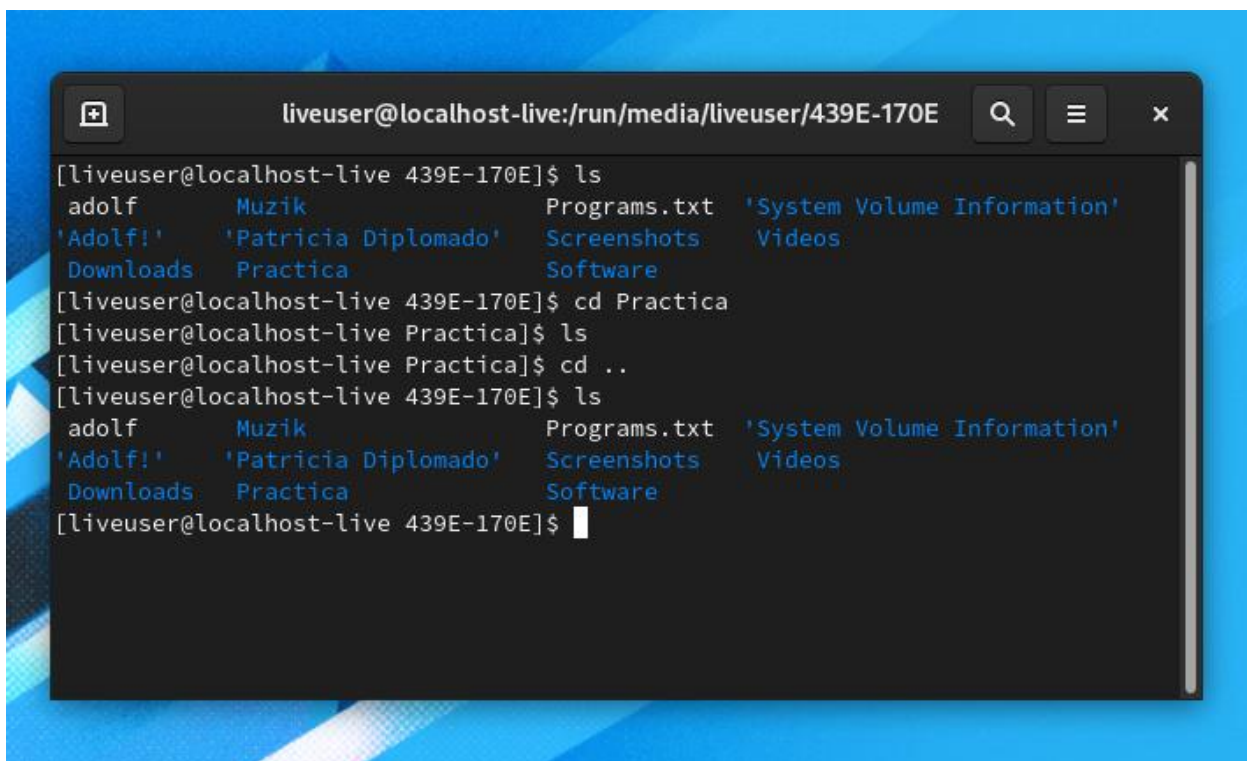
```
cd nombre_carpeta
```

Por lo que si queremos situarnos en la carpeta “tareas” creada anteriormente, se escribe el comando:

```
cd tareas
```

Ahora, si deseamos situarnos en la carpeta de inicio de nuestra cuenta, que es la carpeta padre, escribimos el comando:

```
cd ..
```



The screenshot shows a terminal window titled "liveuser@localhost-live:run/media/liveuser/439E-170E". The terminal output is as follows:

```
[liveuser@localhost-live 439E-170E]$ ls
adolf      Muzik      Programs.txt 'System Volume Information'
'Adolf!'   'Patricia Diplomado' Screenshots  Videos
Downloads  Practica   Software

[liveuser@localhost-live 439E-170E]$ cd Practica
[liveuser@localhost-live Practica]$ ls
[liveuser@localhost-live Practica]$ cd ..
[liveuser@localhost-live 439E-170E]$ ls
adolf      Muzik      Programs.txt 'System Volume Information'
'Adolf!'   'Patricia Diplomado' Screenshots  Videos
Downloads  Practica   Software

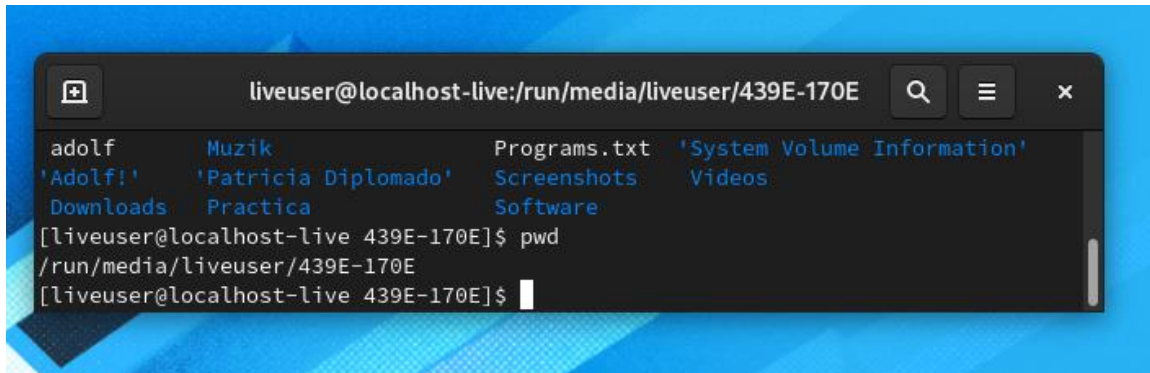
[liveuser@localhost-live 439E-170E]$
```

- *pwd*

(Permite conocer la ruta actual)

El comando *pwd* permite conocer la ubicación actual(ruta), su sintaxis es la siguiente:

```
pwd
```



The screenshot shows a terminal window titled 'liveuser@localhost-live:/run/media/liveuser/439E-170E'. The terminal content includes a directory listing and the execution of the 'pwd' command. The directory listing shows files and folders like 'adolf', 'Muzik', 'Programs.txt', 'System Volume Information', 'Adolf!', 'Patricia Diplomado', 'Screenshots', 'Videos', 'Downloads', 'Practica', and 'Software'. The 'pwd' command is entered and executed, returning the path '/run/media/liveuser/439E-170E'.

```
adolf      Muzik      Programs.txt 'System Volume Information'
'Adolf!'   'Patricia Diplomado' Screenshots  Videos
Downloads  Practica   Software
[liveuser@localhost-live 439E-170E]$ pwd
/run/media/liveuser/439E-170E
[liveuser@localhost-live 439E-170E]$
```

- *find*

El comando *find* permite buscar un elemento dentro del sistema de archivos, su sintaxis es la siguiente:

```
find . -name cadena_buscar
```

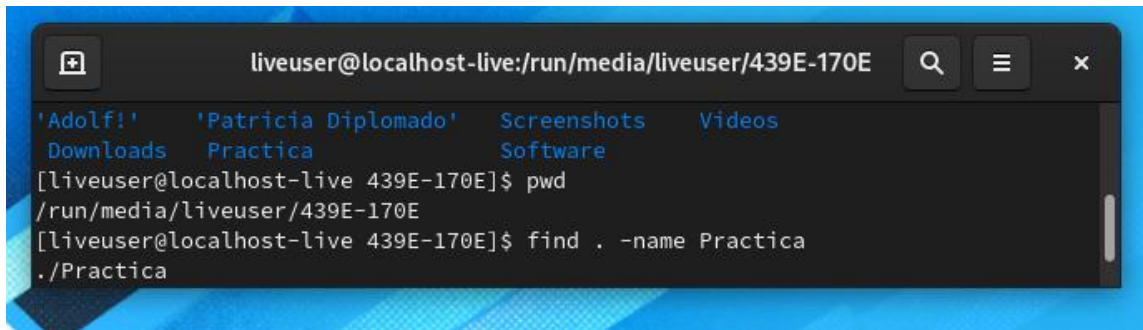
Al comando *find* hay que indicarle en qué parte del sistema de archivos va a iniciar la búsqueda. En el ejemplo anterior la búsqueda se inicia en la posición actual (uso de `.`). Además, utilizando la

bandera `-name` permite determinar la cadena a buscar (comúnmente es el nombre de un archivo).

Si queremos encontrar la ubicación del archivo *tareas*, se escribe el siguiente comando:

```
find . -name tareas
```


(Permite encontrar un archivo o directorio en una jerarquía)



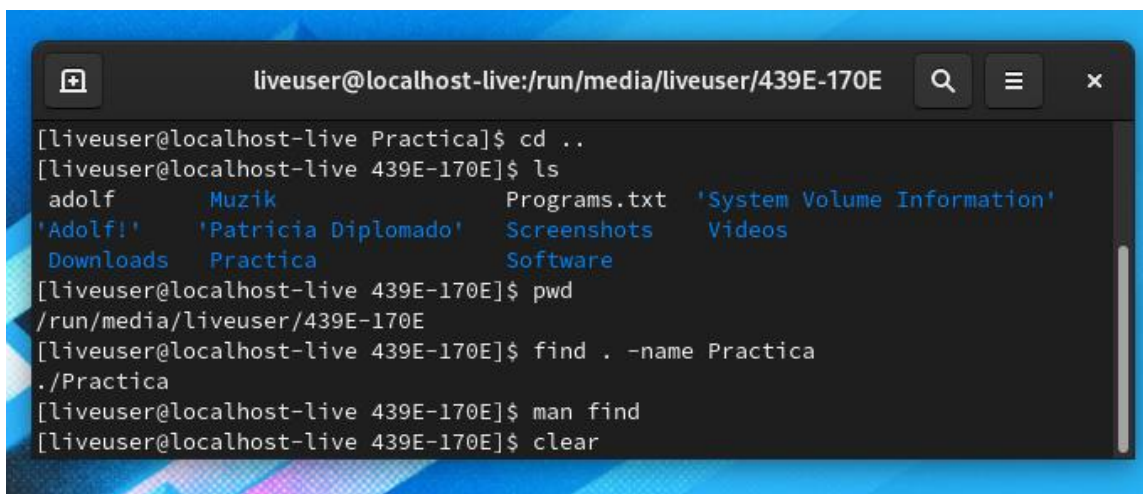
```
liveuser@localhost-live:/run/media/liveuser/439E-170E
'Adolf!' 'Patricia Diplomado' Screenshots Videos
Downloads Practica Software
[liveuser@localhost-live 439E-170E]$ pwd
/run/media/liveuser/439E-170E
[liveuser@localhost-live 439E-170E]$ find . -name Practica
./Practica
```

- *clear*

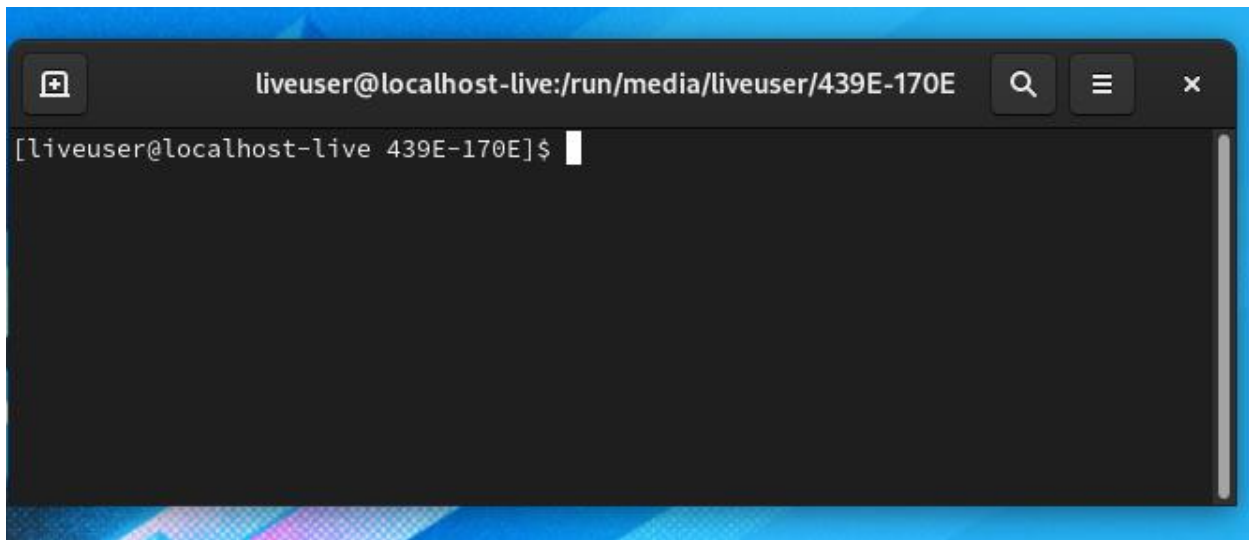
(Limpia la pantalla)

El comando *clear* permite limpiar la consola, su sintaxis es la siguiente:

clear



```
liveuser@localhost-live:/run/media/liveuser/439E-170E
[liveuser@localhost-live Practica]$ cd ..
[liveuser@localhost-live 439E-170E]$ ls
adolf      Muzik      Programs.txt 'System Volume Information'
'Adolf!'   'Patricia Diplomado' Screenshots Videos
Downloads Practica    Software
[liveuser@localhost-live 439E-170E]$ pwd
/run/media/liveuser/439E-170E
[liveuser@localhost-live 439E-170E]$ find . -name Practica
./Practica
[liveuser@localhost-live 439E-170E]$ man find
[liveuser@localhost-live 439E-170E]$ clear
```

- *cp*

(Permite copiar un archivo hacia el mismo directorio)

El comando `cp` permite copiar un archivo, su sintaxis es la siguiente:

```
cp archivo_origen archivo_destino
```

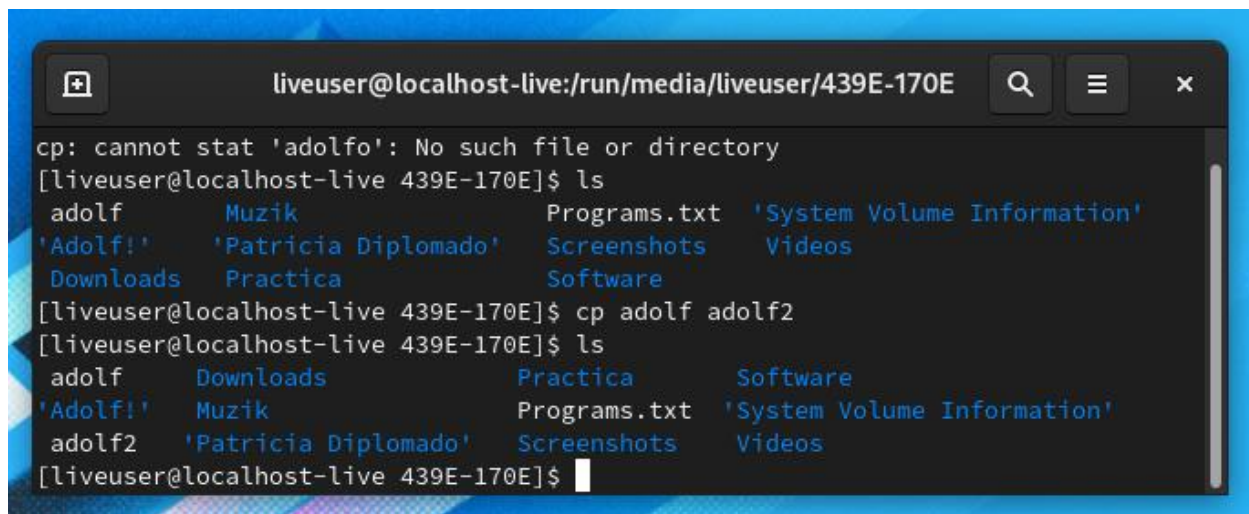
Si queremos una copia del archivo `datos.txt` con nombre `datosViejos.txt` en el mismo directorio, entonces se escribe el comando

```
cp datos.txt datosViejos.txt
```

Ahora, si requerimos una copia de un archivo que está en la carpeta padre en la ubicación actual y con el mismo nombre, entonces podemos emplear las rutas relativas de la siguiente forma:

```
cp ../archivo_a_copiar .
```

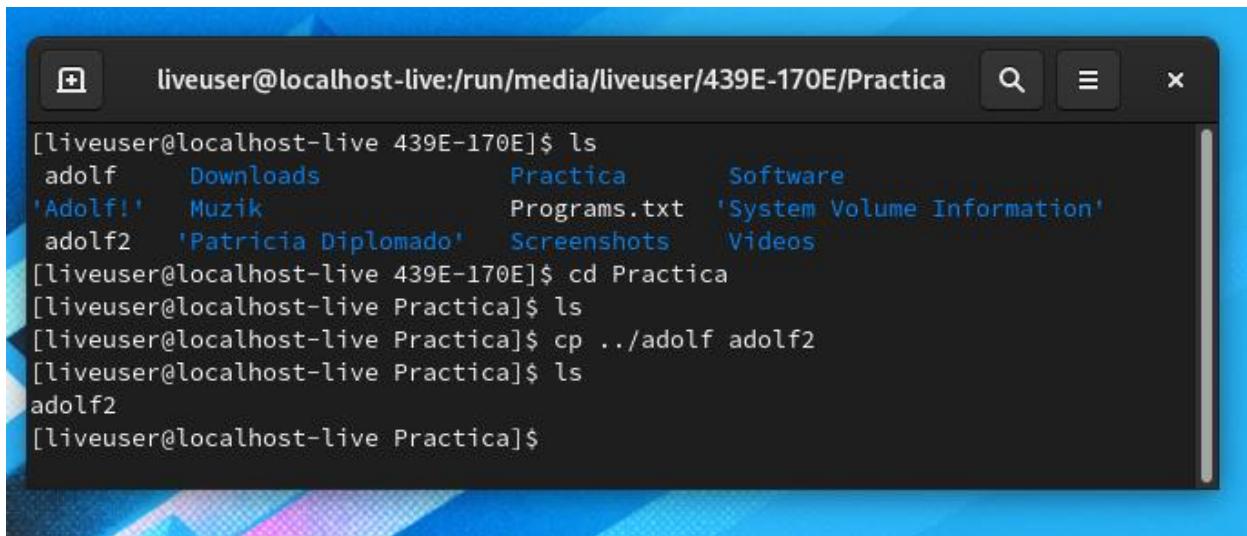
Es muy importante indicar como archivo destino al punto (.) para que el archivo de copia se ubique en el directorio actual.



```
liveuser@localhost-live:/run/media/liveuser/439E-170E
cp: cannot stat 'adolfo': No such file or directory
[liveuser@localhost-live 439E-170E]$ ls
 Adolf!  Muzik      Programs.txt  'System Volume Information'
Downloads  Practica    Screenshots   Videos
adolfo
[liveuser@localhost-live 439E-170E]$ cp adolfo adolfo2
[liveuser@localhost-live 439E-170E]$ ls
 Adolf!  Downloads  Practica      Software
adolfo  Muzik      Programs.txt  'System Volume Information'
adolfo2 'Patricia Diplomado' Screenshots   Videos
[liveuser@localhost-live 439E-170E]$
```

- `cp ../`

(nos permite copiar un archivo ubicado en el directorio padre hacia el nuevo.)

A terminal window titled 'liveuser@localhost-live:/run/media/liveuser/439E-170E/Practica'. The terminal shows the following commands and output:

```
[liveuser@localhost-live 439E-170E]$ ls
adolf    Downloads    Practica     Software
'Adolf!' Muzik        Programs.txt 'System Volume Information'
adolf2   'Patricia Diplomado' Screenshots  Videos

[liveuser@localhost-live 439E-170E]$ cd Practica
[liveuser@localhost-live Practica]$ ls
adolf
adolf2

[liveuser@localhost-live Practica]$ cp ../adolf adolf2

[liveuser@localhost-live Practica]$ ls
adolf
adolf2

[liveuser@localhost-live Practica]$
```

- `mv`

(Mueve archivos hacia otras direcciones)

El comando `mv` mueve un archivo de un lugar a otro, en el sistema de archivos; su sintaxis es la siguiente:

```
mv ubicación_origen/archivo ubicación_destino
```

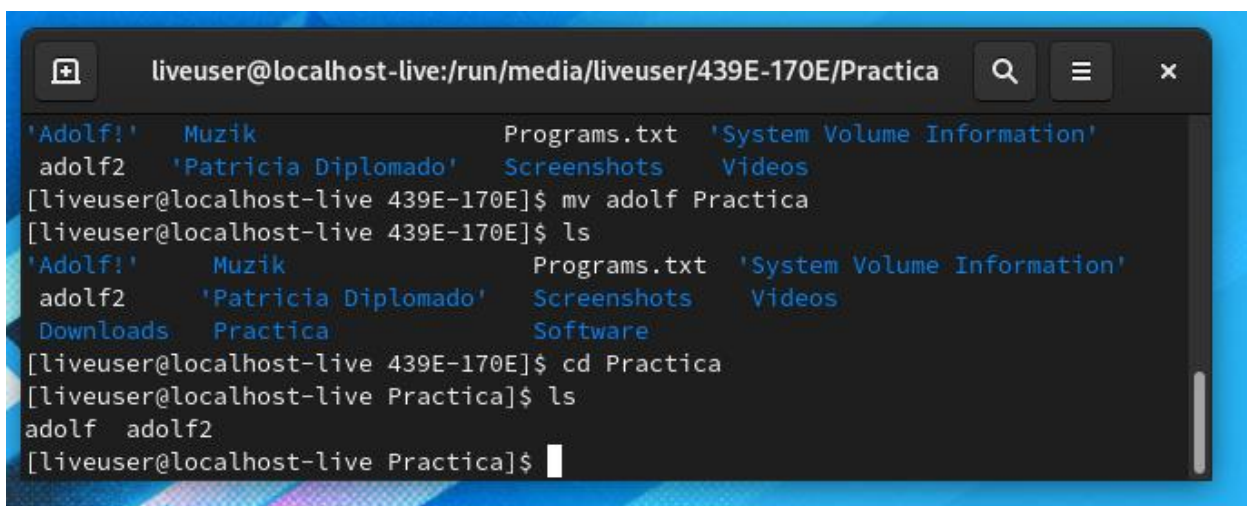
El comando mueve el archivo desde su ubicación origen hacia la ubicación deseada(destino).

Si queremos que un archivo que está en la carpeta padre, reubicarlo en el directorio actual y con el mismo nombre, entonces podemos emplear las rutas relativas de la siguiente forma:

```
mv ../archivo_a_reubicar .
```

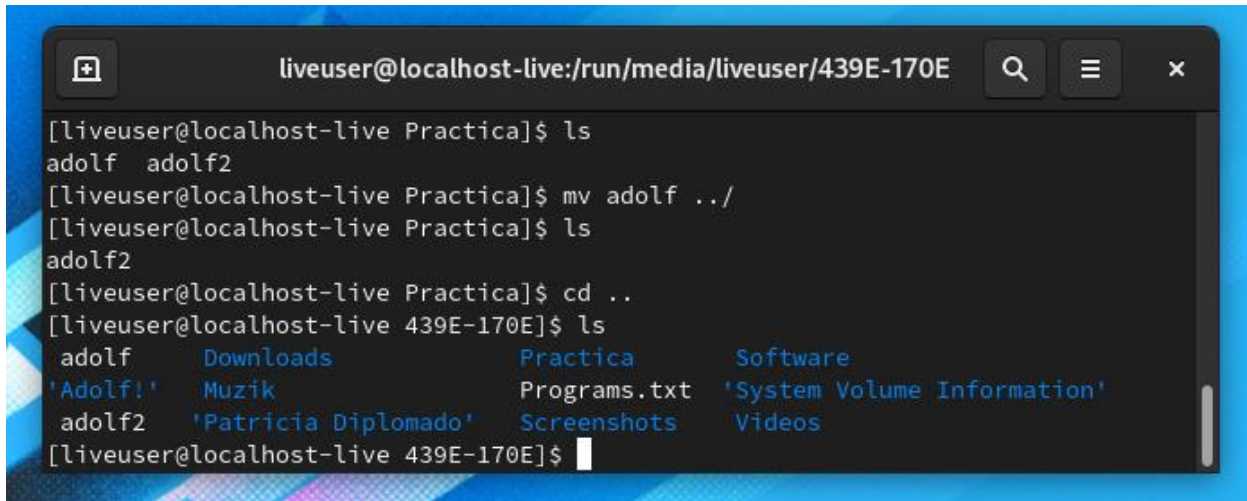
Este comando también puede ser usado para cambiar el nombre de un archivo, simplemente se indica el nombre actual del archivo y el nuevo nombre:

```
mv nombre_actual_archivo nombre_nuevo_archivo
```



```
liveuser@localhost-live:/run/media/liveuser/439E-170E/Practica
'Adolf!'  Muzik          Programs.txt  'System Volume Information'
adolff2   'Patricia Diplomado'  Screenshots  Videos
[liveuser@localhost-live 439E-170E]$ mv adolf Practica
[liveuser@localhost-live 439E-170E]$ ls
'Adolf!'  Muzik          Programs.txt  'System Volume Information'
adolff2   'Patricia Diplomado'  Screenshots  Videos
Downloads Practica          Software
[liveuser@localhost-live 439E-170E]$ cd Practica
[liveuser@localhost-live Practica]$ ls
adolff adolff2
[liveuser@localhost-live Practica]$
```

Para mover a carpeta padre es la misma lógica



```
liveuser@localhost-live:/run/media/liveuser/439E-170E
[liveuser@localhost-live Practica]$ ls
adolf adolf2
[liveuser@localhost-live Practica]$ mv adolf ../
[liveuser@localhost-live Practica]$ ls
adolf2
[liveuser@localhost-live Practica]$ cd ../
[liveuser@localhost-live 439E-170E]$ ls
adolf      Downloads      Practica      Software
'Adolf!'"  Muzik         Programs.txt  'System Volume Information'
adolf2     'Patricia Diplomado' Screenshots   Videos
[liveuser@localhost-live 439E-170E]$
```

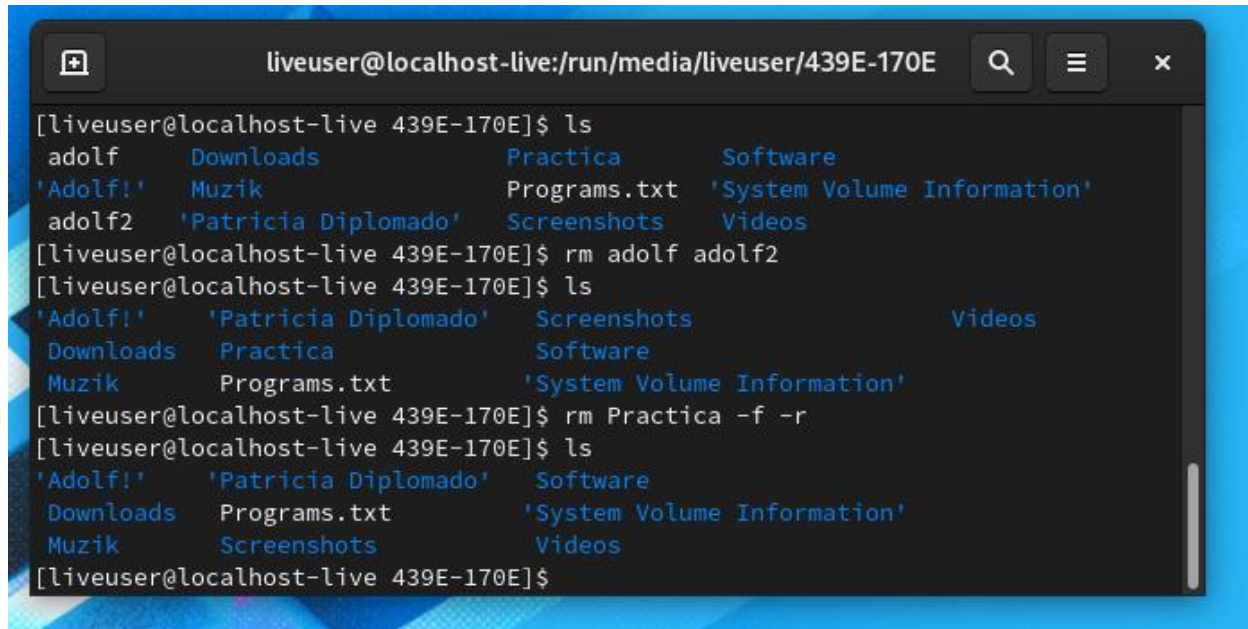
- *rm*

(Remueve archivos, se usa *-f* para forzar la acción o *-r* para que esta acción igualmente sea recursiva)

El comando *rm* permite eliminar un archivo o un directorio, su sintaxis es la siguiente:

```
rm nombre_archivo  
rm nombre_carpeta
```

Cuando la carpeta que se desea borrar contiene información, se debe utilizar la bandera *-f* para forzar la eliminación. Si la carpeta contiene otras carpetas, se debe utilizar la opción *-r*, para realizar la eliminación recursiva.



```
liveuser@localhost-live:/run/media/liveuser/439E-170E  🔍  ☰  ✕  
[liveuser@localhost-live 439E-170E]$ ls  
adolf      Downloads      Practica      Software  
'Adolf!'  
adolf2     'Patricia Diplomado'  Screenshots  Videos  
[liveuser@localhost-live 439E-170E]$ rm Adolf Adolf2  
[liveuser@localhost-live 439E-170E]$ ls  
'Adolf!'  
'Patricia Diplomado'  Screenshots      Videos  
Downloads      Practica      Software  
Muzik      Programs.txt      'System Volume Information'  
[liveuser@localhost-live 439E-170E]$ rm Practica -f -r  
[liveuser@localhost-live 439E-170E]$ ls  
'Adolf!'  
'Patricia Diplomado'  Software  
Downloads      Programs.txt      'System Volume Information'  
Muzik      Screenshots      Videos  
[liveuser@localhost-live 439E-170E]$
```

Conclusión:

En esta practica aprendimos a usar los comandos principales de la terminal en Linux para poder movernos a través del sistema operativo a través de un ambiente que no es un GUI.

Aprendi distintos comandos como ls, rm, mkdir, touch, clear, etc y durante el proceso también aprendimos como buscar la forma de utilizar el comando apropiadamente buscando su información con el comando man seguido del argumento a utilizar.

Fue interesante saber como usar otros comandos además de los que existen en el cmd de Windows.

Referencias:

- Facultad de ingeniería - UNAM. (2018, 6 abril). Manual de prácticas del laboratorio de Fundamentos de programación. Recuperado 04 de enero de 2021, de: http://odin.fib.unam.mx/salac/practicasFP/MADO-17_FP.pdf

35 Linux commands every user should know: <https://www.hostinger.com/tutorials/linux-commands>