

UNIVERSIDAD DE LAS FUERZAS ARMADAS ESPE

DEPARTAMENTO CIENCIAS DE LA COMPUTACIÓN

SISTEMAS OPERATIVOS



**Consultas**

**NRC:** 14912

**Carrera:** Ingeniería de Software

**Nombre:** Gustavo Aguas y Sebastian Paucar

**Docente:** Ing. Fuertes Diaz Walter Marcelo Dr.

SANGOLQUI - ECUADOR

2024

# Capítulo 1

## Laboratorios

### 1.1. Laboratorio 1

#### 1.1.1. Tema: Instalación, configuración y creación de una máquina virtual

##### Objetivos:

- Aprender a crear máquinas virtuales
- Crear nuestra primera máquina virtual

##### Recursos:

- PC
- Windows 10
- VirtualBox
- Ubuntu - .ISO

### Desarrollo

Para realizar la practica de laboratorio tenemos que seguir los siguientes pasos:

- 1. Descargar la imagen de Ubuntu Desktop:** Acceder al sitio web oficial de Ubuntu y descargar la imagen de Ubuntu Desktop versión 24.04 LTS
- 2. Instalar Ubuntu Desktop:** Proceder con la instalación del sistema operativo utilizando la imagen descargada. Seguir cuidadosamente los pasos de instalación proporcionados por el asistente de instalación.
  - 2.1** Seleccionamos Normal Installation
  - 2.2** Seleccionamos el idioma que vamos a manejar y damos click en Install Ubuntu.
  - 2.3** Por defecto nuestra localización Geográfica sera Guayaquil.
  - 2.4** Ingresamos los datos que nos piden para continuar con la instalación.

### 3. Configurar Ubuntu Desktop

### 3.1 Abrimos Virtual Box



Figura 1.1: Virtual Box

3.2 En la parte superior seleccionamos la opción Maquina y después la opción nueva, colocamos el nombre que deseamos para nuestra máquina virtual.

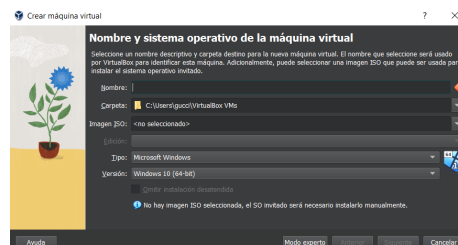


Figura 1.2: Creación Máquina Virtual

3.3 Revisamos las especificaciones de nuestro computador, para seguir con la creación de nuestra máquina virtual.

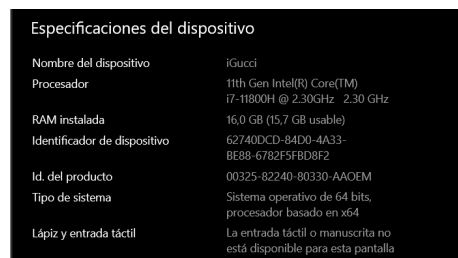


Figura 1.3: Especificaciones del Computador

3.4 Al revisar nuestras especificaciones seguimos con la creación de la máquina, en la cual tenemos que señalar la memoria base y los procesadores.

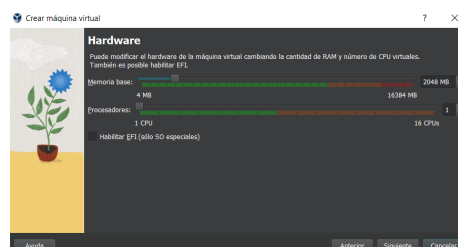


Figura 1.4: Hardware

3.5 Después de terminada la instalación y configuración ya podremos usar la máquina virtual de Ubuntu.

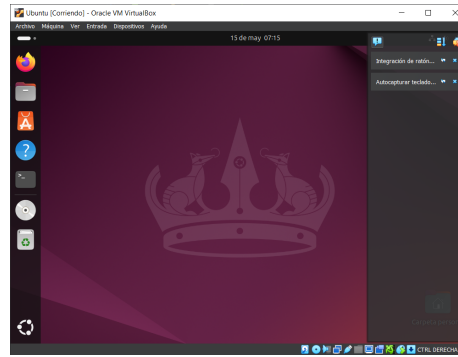


Figura 1.5: Ubt

## Conclusiones

- La creación de una máquina virtual en Ubuntu es un proceso eficiente y beneficioso para diversos fines. Al crear una máquina virtual, se logra un entorno virtualizado que permite ejecutar diferentes sistemas operativos y aplicaciones en una sola máquina física. Esto brinda flexibilidad, escalabilidad y seguridad.
- Tener en cuenta las especificaciones de cada computador para el correcto funcionamiento de la máquina virtual que se va a crear.

## Referencias

Download Ubuntu Desktop — Download — Ubuntu. (s.f.). Ubuntu. <https://ubuntu.com/download/desktop>

## 1.2. Laboratorio 2

### 1.2.1. Comandos básicos de Linux

#### Objetivos:

- Al final la sesión el estudiante aprenderá a obtener ayuda en línea para obtendrá ayuda en línea para aprehender cualquier comando que necesite.
- Al finalizar esta clase el estudiante será capas de navegar por todo el filesystem , prender y apagar la maquina.

#### Recursos:

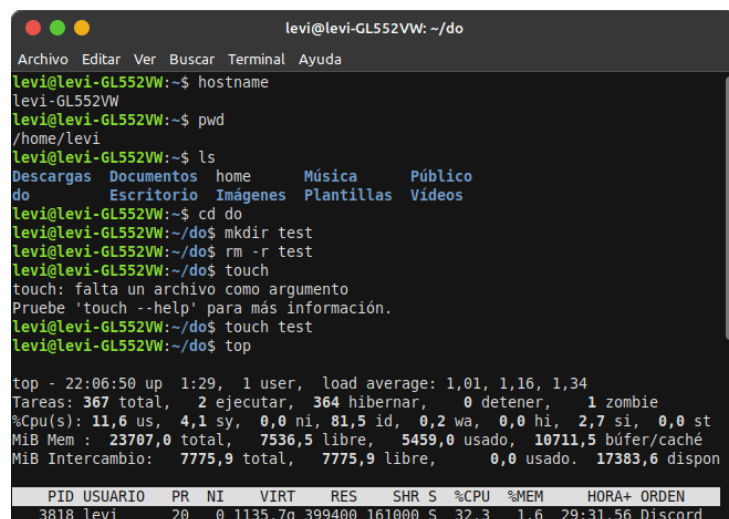
- PC
- Windows 10
- VirtualBox
- Ubuntu - .ISO

## Desarrollo

En esta practica de laboratorio aprenderemos los comandos basicos de Linux con la finalidad de adentrarnos y familiarizarnos.

### 1. Comandos para tareas básicas de Linux

- hostname: Manual en linea.
- pwd: Muestra el directorio actual.
- clear: limpiar pantalla
- ls: Lista el contenido de una carpeta.
- cd: Cambia de directorio.
- cat: Muestra el contenido de un archivo.
- mkdir: Crea un nuevo directorio.
- rm: Elimina archivos o directorios.
- history: Muestra el historial de comandos ejecutados.
- tar: Archiva o descomprime archivos.
- sudo: Ejecuta un comando con privilegios de superusuario.
- top: Muestra los procesos que están consumiendo más recursos del sistema.
- df: Muestra el espacio libre y utilizado en los sistemas de archivos.
- ping: Verifica la conectividad de red con un host remoto.
- ls -a: Lista archivos y directorios, incluyendo los ocultos.
- info: Proporciona información adicional sobre el sistema y comandos.
- help: Muestra información de ayuda sobre el comando.
- date: Muestra la fecha y la hora actuales.
- who: Muestra información sobre los usuarios conectados.
- whoami: Muestra el nombre de usuario actual.



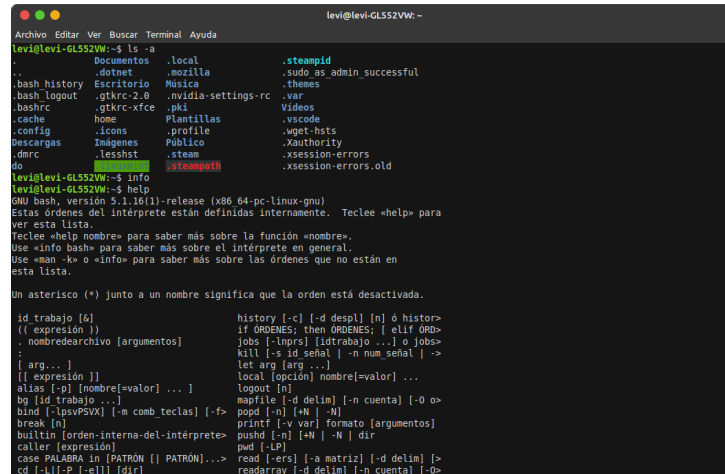
```

levi@levi-GL552VW: ~/do
Archivo Editar Ver Buscar Terminal Ayuda
levi@levi-GL552VW:~$ hostname
levi-GL552VW
levi@levi-GL552VW:~$ pwd
/home/levi
levi@levi-GL552VW:~$ ls
Descargas  Documentos  home        Música      Público
do         Escritorio  Imágenes    Plantillas  Videos
levi@levi-GL552VW:~$ cd do
levi@levi-GL552VW:~/do$ mkdir test
levi@levi-GL552VW:~/do$ rm -r test
levi@levi-GL552VW:~/do$ touch
touch: falta un archivo como argumento
Pruebe 'touch --help' para más información.
levi@levi-GL552VW:~/do$ touch test
levi@levi-GL552VW:~/do$ top
top - 22:06:50 up 1:29, 1 user, load average: 1,01, 1,16, 1,34
Tareas: 367 total, 2 ejecutar, 364 hibernar, 0 detener, 1 zombie
%Cpu(s): 11,6 us, 4,1 sy, 0,0 ni, 81,5 id, 0,2 wa, 0,0 hi, 2,7 si, 0,0 st
MiB Mem : 23707,0 total, 7536,5 libre, 5459,0 usado, 10711,5 búfer/caché
MiB Intercambio: 7775,9 total, 7775,9 libre, 0,0 usado, 17383,6 dispon

  PID USUARIO PR NI  VIRT  RES  SHR S  %CPU  %MEM  HORA+ ORDEN
3818 levi    20  0 1135,7g 399400 161000 S   32,3   1,6  29:31.56 Discord

```

Figura 1.6: Comando basicos de linux



```

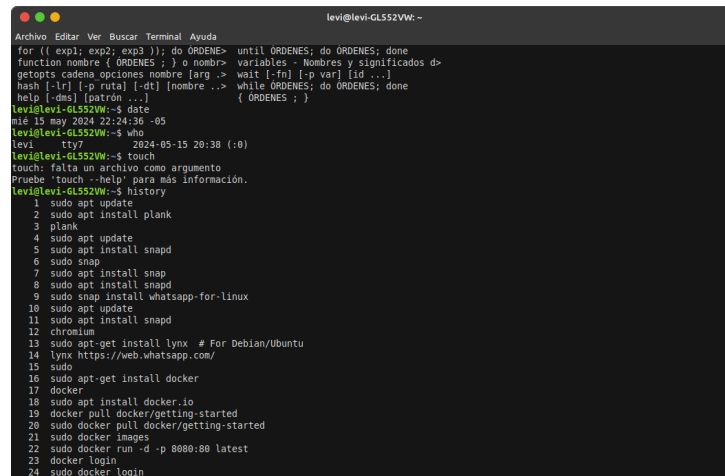
levi@levi-GL552VW:~$ ls -la
.          .Documents  .local      .steampid
..         .dotnet     .mozilla    .sudo_as_admin_successful
.bash_history  Escritorio  Música      .themes
.bash_logout  gkrcc-2.0  .nvidia-settings-rc  .var
.bashrc       gkrcc-xfce .pk          Videos
.cache        home       Plantillas  .vscode
.config       .icons    .profile    .wget-hsts
Descargas     Imágenes   Público     .Xauthority
.dnrc         .lessshat .steam      .xsession-errors
dp            .steampath .xsession-errors.old

levi@levi-GL552VW:~$ info
levi@levi-GL552VW:~$ help
GNU bash, versión 5.1.16(1)-release (x86_64-pc-linux-gnu)
Estas órdenes del intérprete están definidas internamente. Teclee «help» para
ver esta lista.
Teclee «help nombre» para saber más sobre la función «nombre».
Use «info bash» para saber más sobre el intérprete en general.
Use «man «>»» o «info» para saber más sobre las órdenes que no están en
esta lista.
Un asterisco (*) junto a un nombre significa que la orden está desactivada.

id trabajo [6]          history [-c] [-d despl] [n] ó histor>
(( expresión ))        if ORDENES; then ORDENES; [ elif ORD>
.nombrederechivo [argumentos]  jobs [-lnprs] [idtrabajo ...] o jobs>
[ arg... ]              kill [-s id señal | -n num_señal | ->
let arg [arg ...]
[[ expresión ]]         local [opción] nombre[=valor] ...
alias [-p] [nombre[=valor] ... ]  logout [n]
eg [id.trabajo ...]      mapfile [-d delim] [-n cuenta] [-O o>
bind [-lpsvPSVX] [-m comb teclas] [-f>  popd [-n] [+N | -N]
break [n]               printf [-v var] formato [argumentos]
builtin [orden-interna-del-intérprete>  pushd [-n] [+N | -N | dir
caller [expresión]       pwd [-LP]
case PALABRA in [PATRON] [...] ...>  read [-ers] [-a matriz] [-d delim] [>
cd [-L|-P [-e]] [dir]    readarray [-d delim] [-n cuenta] [->

```

Figura 1.7: Comando basicos de linux



```

levi@levi-GL552VW:~$ who
levi          tty7          2024-05-15 20:38 (+0)

levi@levi-GL552VW:~$ touch
touch: falta un archivo como argumento
Pruebe 'touch --help' para más información.

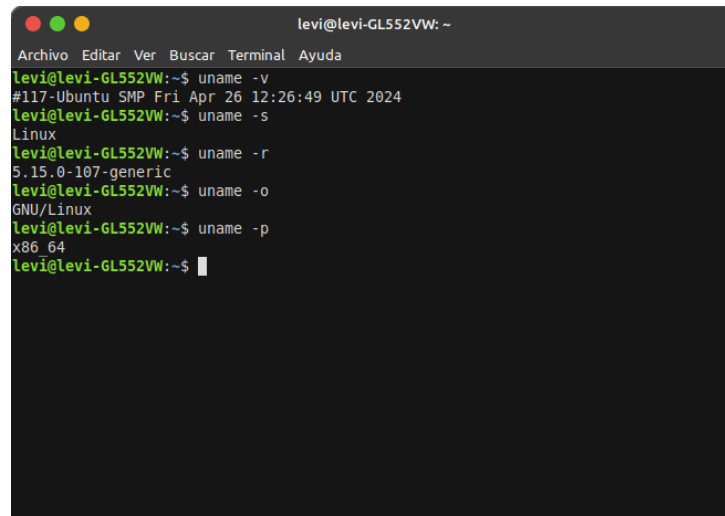
levi@levi-GL552VW:~$ history
1 sudo apt update
2 sudo apt install plank
3 plank
4 sudo apt update
5 sudo apt install snapd
6 sudo snap
7 sudo apt install snap
8 sudo apt install snapd
9 sudo snap install whatsapp-for-linux
10 sudo apt update
11 sudo apt install snapd
12 chromium
13 sudo apt-get install lynx # For Debian/Ubuntu
14 lynx https://web.whatsapp.com/
15 sudo
16 sudo apt-get install docker
17 docker
18 sudo apt install docker.io
19 docker pull docker/getting-started
20 sudo docker pull docker/getting-started
21 sudo docker images
22 sudo docker run -d -p 8080:80 latest
23 docker login
24 sudo docker login

```

Figura 1.8: Comando basicos de linux

## 2. Comandos para ver las versiones:

- **uname -v:** Este comando devuelve la versión del kernel (núcleo) del sistema operativo. Proporciona detalles sobre la versión del kernel que está actualmente en ejecución.
- **uname -s:** Devuelve el nombre del sistema operativo.
- **uname -r:** Proporciona la versión del kernel. Muestra la versión específica del kernel que está en ejecución.
- **uname -o:** Este comando devuelve el nombre del sistema operativo
- **uname -p:** Proporciona el tipo de procesador o arquitectura de la máquina.



```

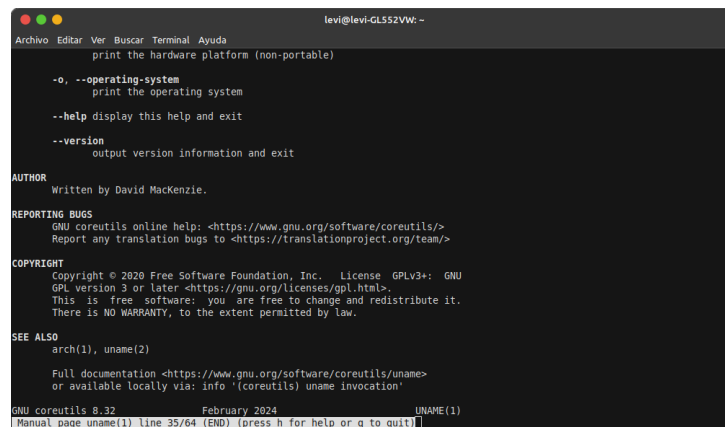
levi@levi-GL552VW: ~
Archivo Editar Ver Buscar Terminal Ayuda
levi@levi-GL552VW:~$ uname -v
#117-Ubuntu SMP Fri Apr 26 12:26:49 UTC 2024
levi@levi-GL552VW:~$ uname -s
Linux
levi@levi-GL552VW:~$ uname -r
5.15.0-107-generic
levi@levi-GL552VW:~$ uname -o
GNU/Linux
levi@levi-GL552VW:~$ uname -p
x86_64
levi@levi-GL552VW:~$

```

Figura 1.9: Comandos en Linux

### 3. Comando para el manual de texto

- **man uname:** Se utiliza para mostrar el manual de la utilidad uname. La utilidad uname proporciona información sobre el sistema operativo en el que se está trabajando. Al ejecutar `man uname`, accedemos a la página del manual que describe en detalle cómo usar este comando y qué opciones puedes utilizar con él.



```

levi@levi-GL552VW: ~
Archivo Editar Ver Buscar Terminal Ayuda
print the hardware platform (non-portable)

-o, --operating-system
    print the operating system

--help display this help and exit

--version
    output version information and exit

AUTHOR
    Written by David Mackenzie.

REPORTING BUGS
    GNU coreutils online help: <https://www.gnu.org/software/coreutils/>
    Report any translation bugs to <https://translationproject.org/team/>

COPYRIGHT
    Copyright © 2020 Free Software Foundation, Inc.  License GPLv3+: GNU
    GPL version 3 or later <https://gnu.org/licenses/gpl.html>.
    This is free software: you are free to change and redistribute it.
    There is NO WARRANTY, to the extent permitted by law.

SEE ALSO
    arch(1), uname(2)

    Full documentation <https://www.gnu.org/software/coreutils/uname>
    or available locally via: info '(coreutils) uname invocation'

GNU coreutils 8.32      February 2024      UNAME(1)
[Manual page uname(1) line 35/64 (END) (press h for help or q to quit)]

```

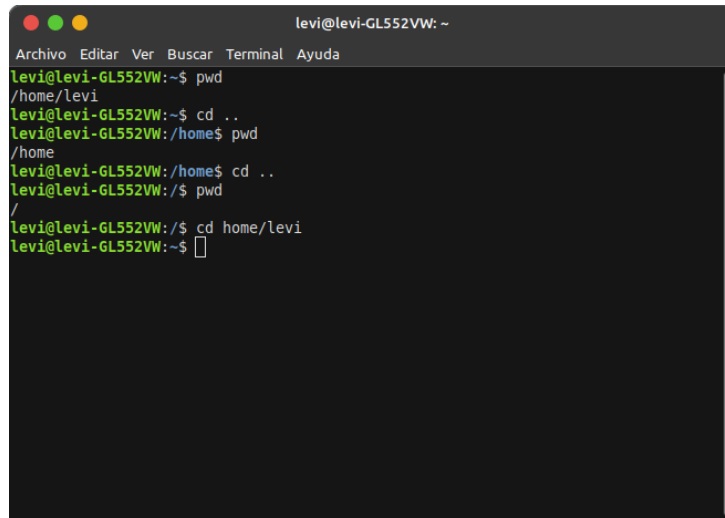
Figura 1.10: Comando man uname

### 4. Comandos para navegación de directorios

Como se logra observar en la figura 1.13, se están ejecutando una serie de comandos para navegar por el sistema de archivos. A continuación se detalla lo que se está haciendo en cada paso:

- `'pwd'`: Muestra el directorio actual, que es `'/home/levi'`.
- `'cd ..'`: Cambia al directorio padre, por lo que el directorio actual se convierte en `'/home'`.
- `pwd`: Muestra el directorio actual, que es `/home`.
- `'cd ..'`: Cambia al directorio padre, por lo que el directorio actual se convierte en `'/'` (raíz del sistema de archivos).
- `'pwd'`: Muestra el directorio actual, que es `'/'` (raíz del sistema de archivos).

- `'cd /home/levi/':` Cambia al directorio `'/home/levi/'`.
- El prompt cambia a `levi@levi-GL552VW: $`, lo que indica que el directorio actual es `/home/levi`.



```
levi@levi-GL552VW: ~  
Archivo Editar Ver Buscar Terminal Ayuda  
levi@levi-GL552VW:~$ pwd  
/home/levi  
levi@levi-GL552VW:~$ cd ..  
levi@levi-GL552VW:/home$ pwd  
/home  
levi@levi-GL552VW:/home$ cd ..  
levi@levi-GL552VW:/$ pwd  
/  
levi@levi-GL552VW:/$ cd home/levi  
levi@levi-GL552VW:~$
```

Figura 1.11: Navegación de directorios

Adicionalmente, se puede ver la estructura de directorios de manera gráfica con el comando `'tree'`. A continuación se realiza una breve descripción de los comandos de `tree`:

- **`sudo apt-get install tree`:** Comando que instala el comando `tree`, ya que este comando no viene instalado por defecto en ubuntu.
- **`tree`:** Imprime la estructura de los directorios y archivos en forma de árbol lo que facilita la visualización de la jerarquía de los directorios del sistema operativo. Además, el comando `tree` permite listar los directorios de los dispositivos externos. Al ejecutar el comando `tree` sin opciones, se mostrará la estructura de directorios a partir del directorio actual.
- **`tree -L 1`:** Muestra la estructura de directorios del directorio actual hasta un nivel de profundidad específico, en este caso, un nivel. La opción `'-L'` seguida de un número indica el nivel de profundidad que se desea mostrar en el árbol de directorios. Al utilizar `'-L 1'`, se mostrará únicamente el contenido del directorio actual, sin incluir subdirectorios. Esto es útil para obtener una vista rápida de la estructura de directorios en el directorio actual sin adentrarse en subdirectorios.
- **`tree -L 1 | more`:** Muestra la estructura de directorios del directorio actual hasta un nivel de profundidad específico. Al utilizar `'|'` (pipe) y el comando `more`, la salida se visualiza de manera controlada, permitiendo desplazarse por el contenido página por página. Esto es útil para obtener una vista rápida y organizada de la estructura de directorios en el directorio actual sin adentrarse en subdirectorios.



```

levi@levi-GL552VM: ~
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
├── Sin título.canvas
├── Web
│   ├── Dockerfile.file
│   └── Writer
│       └── Ejemplo.odt
├── Escritorio
├── home
├── Imágenes
│   ├── Captura de pantalla de 2024-05-15 18-37-55.png
│   ├── Captura de pantalla de 2024-05-15 18-48-37.png
│   ├── Captura de pantalla de 2024-05-15 18-41-35.png
│   ├── Captura de pantalla de 2024-05-15 18-46-06.png
│   ├── Captura de pantalla de 2024-05-15 18-52-01.png
│   ├── Captura de pantalla de 2024-05-15 18-53-09.png
│   ├── Captura de pantalla de 2024-05-15 20-59-51.png
│   ├── Captura de pantalla de 2024-05-15 21-06-25.png
│   ├── directorios.png
│   ├── manuname.png
│   └── uname.png
├── Música
├── Plantillas
├── Público
└── Videos

18 directories, 29 files
levi@levi-GL552VM:~$ tree -L 1
├── Descargas
├── do
├── Documentos
├── Escritorio
├── home
├── Imágenes
├── Música
├── Plantillas
├── Público
└── Videos

9 directories, 1 file
levi@levi-GL552VM:~$ tree -L 1 | more
├── Descargas
├── do
├── Documentos
├── Escritorio
├── home
├── Imágenes
├── Música
├── Plantillas
├── Público
└── Videos

9 directories, 1 file
levi@levi-GL552VM:~$

```

Figura 1.12: Vista del comando 'tree -L 1'

- **history:** Muestra el historial de comandos previamente ejecutados en una sesión.

## 6. Comando para apagar la maquina virtual

- **shutdown -h now**

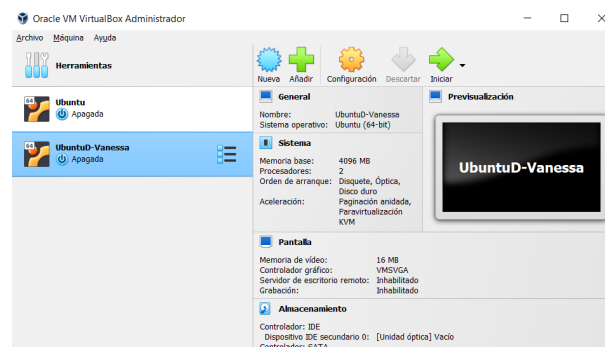


Figura 1.13: Comando para apagar la maquina virtual

## Conclusiones

- Los comandos básicos de Linux son fundamentales para la administración y manipulación de archivos y directorios en un sistema operativo, estos comandos permiten a los usuarios realizar tareas como crear, copiar, mover y eliminar archivos, así como navegar por la estructura de directorios del sistema.

- Los comandos `uname -v`, `s`, `r`, o `p` son herramientas útiles para obtener información específica sobre el sistema operativo y el kernel en un sistema Linux.
- El comando `man uname` es una herramienta útil para obtener información detallada sobre el comando `uname` y aprovechar al máximo sus funcionalidades en un sistema Linux.

## Referencias

Equipo editorial de IONOS. (2023, 22 agosto). Comandos de Linux. IONOS Digital Guide.

## 1.3. Laboratorio 3

### 1.3.1. Gestión de archivos y directorios en Linux

#### Objetivos:

- Identificar el sistema de archivos Linux
- Manipular archivos y directorios

#### Recursos:

- PC
- MS-Windows
- VirtualBox
- Ubuntu - .ISO

### Desarrollo

La gestión de archivos en Linux es un aspecto fundamental del sistema operativo, ya que proporciona herramientas y comandos para crear, remover, renombrar, editar, visualizar y eliminar archivos y directorios. Con su estructura jerárquica de directorios, Linux permite a los usuarios y administradores organizar la información de manera eficiente.

### Gestion de archivos

1. **Comando mkdir:** Con este comando podemos crear el directorio donde se van a almacenar los archivos de la práctica.

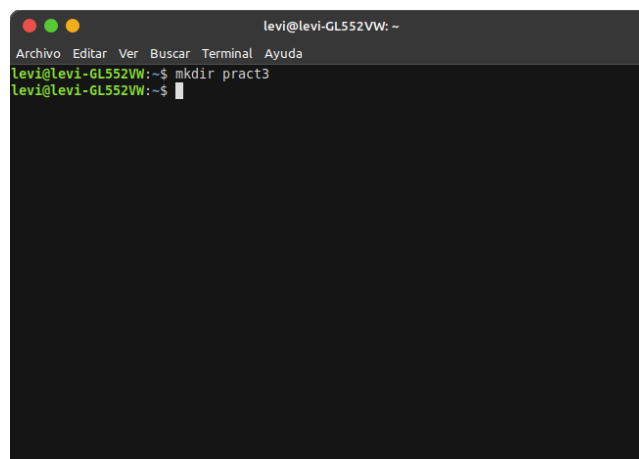
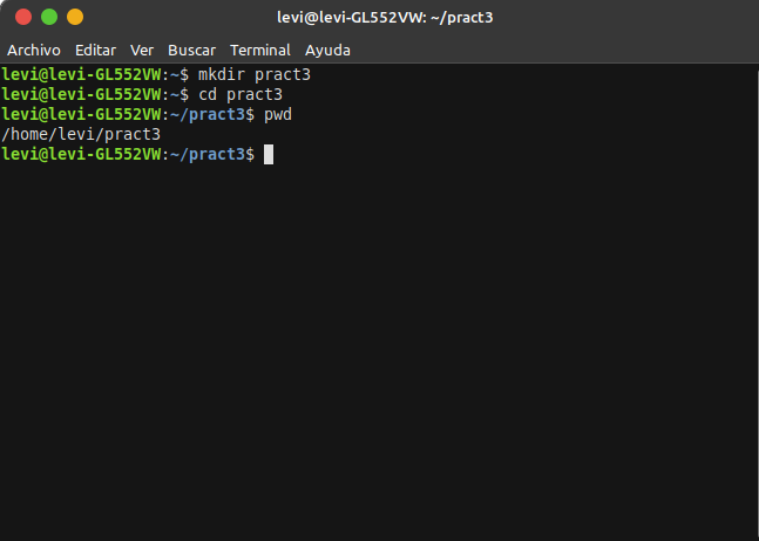


Figura 1.14: Comando mkdir

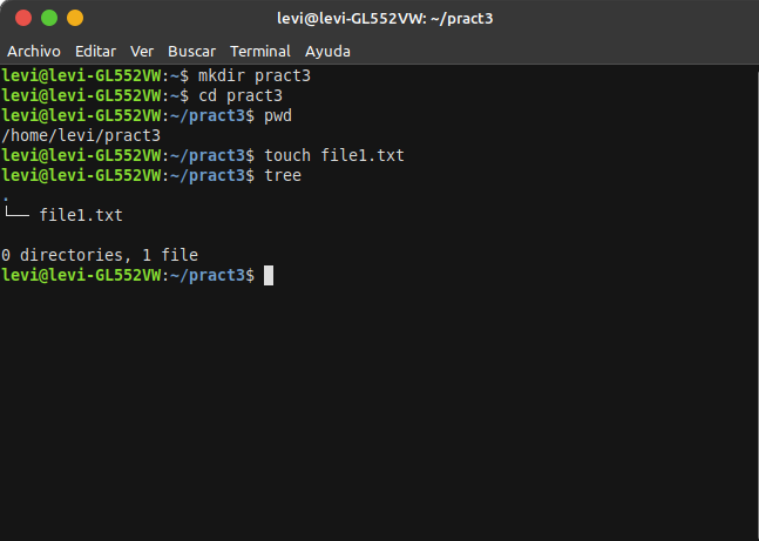
2. **Comando cd:** Con este comando podemos acceder al directorio que hemos creado.

A terminal window titled 'levi@levi-GL552VW: ~/pract3' with a menu bar (Archivo, Editar, Ver, Buscar, Terminal, Ayuda). The terminal shows the following commands and output:

```
levi@levi-GL552VW:~$ mkdir pract3
levi@levi-GL552VW:~$ cd pract3
levi@levi-GL552VW:~/pract3$ pwd
/home/levi/pract3
levi@levi-GL552VW:~/pract3$
```

Figura 1.15: Comando cd y pwd

3. **Comando touch:** Con este comando podemos crear un archivo el cual va a estar vacío.

A terminal window titled 'levi@levi-GL552VW: ~/pract3' with a menu bar (Archivo, Editar, Ver, Buscar, Terminal, Ayuda). The terminal shows the following commands and output:

```
levi@levi-GL552VW:~$ mkdir pract3
levi@levi-GL552VW:~$ cd pract3
levi@levi-GL552VW:~/pract3$ pwd
/home/levi/pract3
levi@levi-GL552VW:~/pract3$ touch file1.txt
levi@levi-GL552VW:~/pract3$ tree
.
├─ file1.txt

0 directories, 1 file
levi@levi-GL552VW:~/pract3$
```

Figura 1.16: Comando touch

4. **Comando ls-l:** Este comando se utiliza para mostrar el contenido de un directorio.

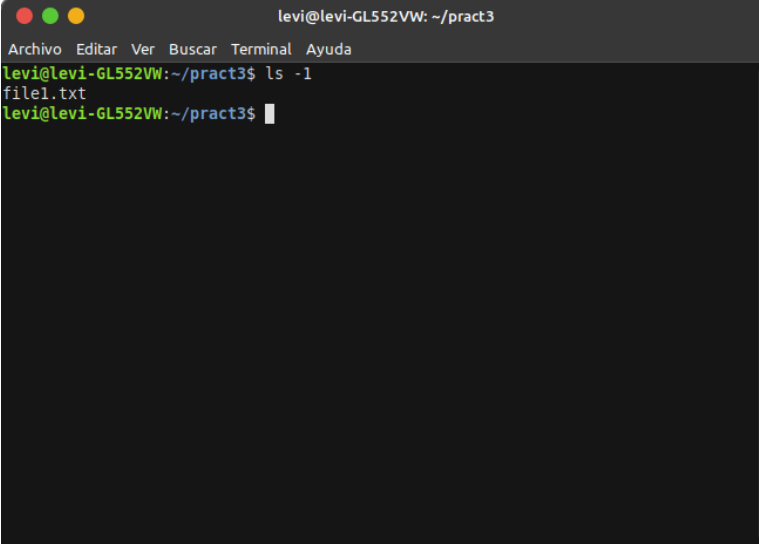
A terminal window titled 'levi@levi-GL552VW: ~/pract3' with a menu bar (Archivo, Editar, Ver, Buscar, Terminal, Ayuda). The prompt is 'levi@levi-GL552VW:~/pract3\$'. The command 'ls -l' is entered, and the output 'file1.txt' is displayed on the next line. The prompt returns to 'levi@levi-GL552VW:~/pract3\$'.

Figura 1.17: Comando ls -l

5. **Comando cat:** Este comando nos permite realizar el redireccionamiento de informacion al archivo que creamos

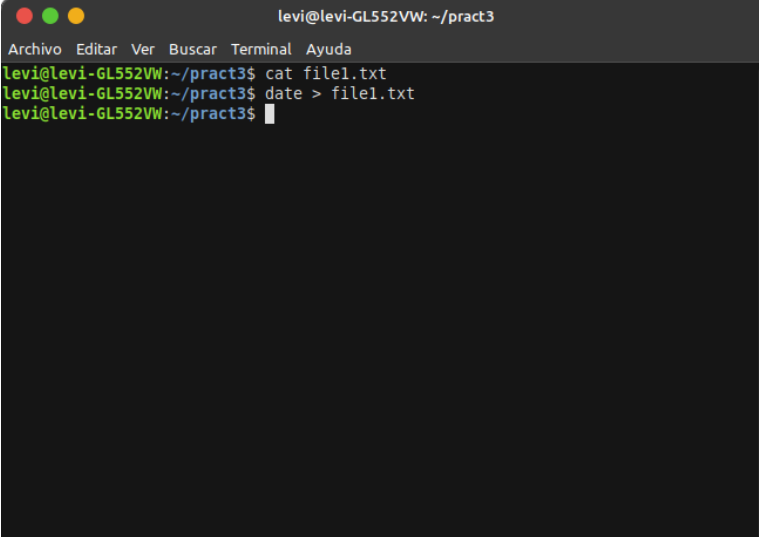
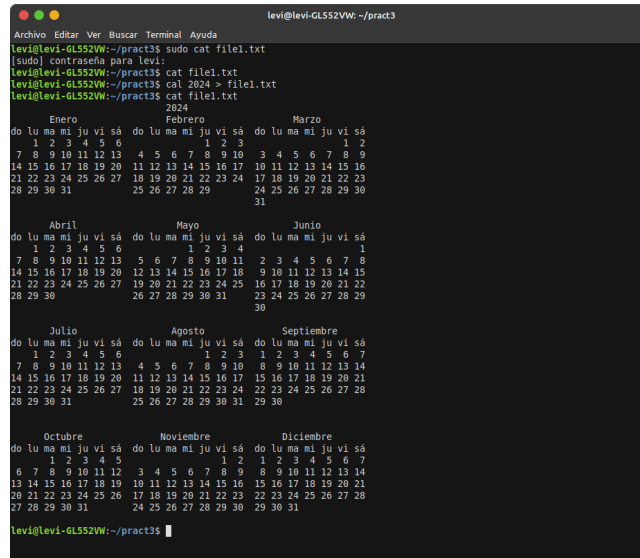
A terminal window titled 'levi@levi-GL552VW: ~/pract3' with a menu bar (Archivo, Editar, Ver, Buscar, Terminal, Ayuda). The prompt is 'levi@levi-GL552VW:~/pract3\$'. The command 'cat file1.txt' is entered. The prompt returns to 'levi@levi-GL552VW:~/pract3\$'. The command 'date > file1.txt' is entered. The prompt returns to 'levi@levi-GL552VW:~/pract3\$'.

Figura 1.18: Comando cat y redireccionamiento de data al file1.txt

6. **Comando cal:** Permite observar la fecha que deseamos, pero para utilizar primero tenemos que instalarnos con el siguiente comando: `sudo apt-get -y install ncal`



```

levi@levi-GL552VW: ~/pract3
levi@levi-GL552VW:~/pract3$ sudo cat file1.txt
[sudo] contraseña para levi:
levi@levi-GL552VW:~/pract3$ cat file1.txt
levi@levi-GL552VW:~/pract3$ cal 2024 > file1.txt
levi@levi-GL552VW:~/pract3$ cat file1.txt

      2024
Enero
do lu ma mi ju vi sa
1 2 3 4 5 6
7 8 9 10 11 12 13
14 15 16 17 18 19 20
21 22 23 24 25 26 27
28 29 30 31

Febrero
do lu ma mi ju vi sa
1 2 3 4
5 6 7 8 9 10
11 12 13 14 15 16 17
18 19 20 21 22 23 24
25 26 27 28 29

Marzo
do lu ma mi ju vi sa
1 2
3 4 5 6 7 8 9
10 11 12 13 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29 30
31

Abril
do lu ma mi ju vi sa
1 2 3 4 5 6
7 8 9 10 11 12 13
14 15 16 17 18 19 20
21 22 23 24 25 26 27
28 29 30

Mayo
do lu ma mi ju vi sa
1 2 3 4
5 6 7 8 9 10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28 29 30 31

Junio
do lu ma mi ju vi sa
1
2 3 4 5 6 7 8
9 10 11 12 13 14 15
16 17 18 19 20 21 22
23 24 25 26 27 28 29
30

Julio
do lu ma mi ju vi sa
1 2 3 4 5 6
7 8 9 10 11 12 13
14 15 16 17 18 19 20
21 22 23 24 25 26 27
28 29 30 31

Agosto
do lu ma mi ju vi sa
1 2 3
4 5 6 7 8 9 10
11 12 13 14 15 16 17
18 19 20 21 22 23 24
25 26 27 28 29 30 31

Septiembre
do lu ma mi ju vi sa
1 2 3 4 5 6 7
8 9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30

Octubre
do lu ma mi ju vi sa
1 2 3 4 5
6 7 8 9 10 11 12
13 14 15 16 17 18 19
20 21 22 23 24 25 26
27 28 29 30 31

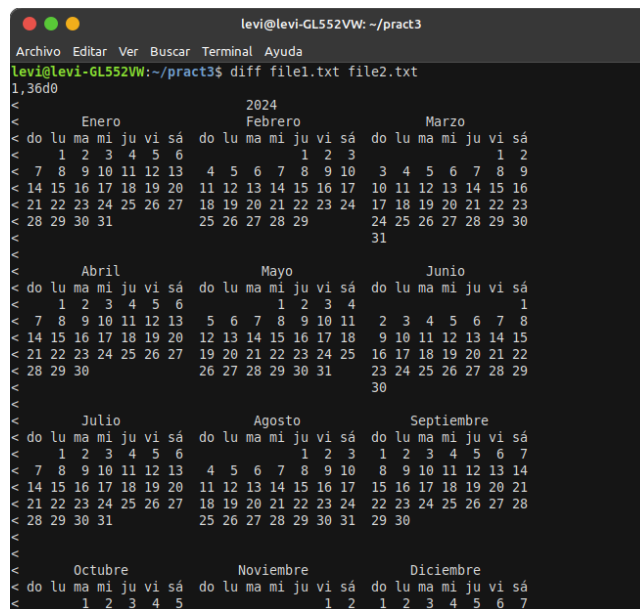
Noviembre
do lu ma mi ju vi sa
1 2
3 4 5 6 7 8 9
10 11 12 13 14 15 16
17 18 19 20 21 22 23
24 25 26 27 28 29 30

Diciembre
do lu ma mi ju vi sa
1 2 3 4 5 6 7
8 9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30 31

```

Figura 1.19: Comando cal

7. **Comando diff:** Este comando permite comparar el contenido de los archivos.



```

levi@levi-GL552VW: ~/pract3
levi@levi-GL552VW:~/pract3$ diff file1.txt file2.txt
1,360d0
<
<      2024
< Enero
< do lu ma mi ju vi sa
< 1 2 3 4 5 6
< 7 8 9 10 11 12 13
< 14 15 16 17 18 19 20
< 21 22 23 24 25 26 27
< 28 29 30 31
<
<
<
<      2024
< Febrero
< do lu ma mi ju vi sa
< 1 2 3 4
< 5 6 7 8 9 10
< 11 12 13 14 15 16 17
< 18 19 20 21 22 23 24
< 25 26 27 28 29
<
<
<
<      2024
< Marzo
< do lu ma mi ju vi sa
< 1 2
< 3 4 5 6 7 8 9
< 10 11 12 13 14 15 16
< 17 18 19 20 21 22 23
< 24 25 26 27 28 29 30
< 31
<
<
<
<
<      2024
< Abril
< do lu ma mi ju vi sa
< 1 2 3 4 5 6
< 7 8 9 10 11 12 13
< 14 15 16 17 18 19 20
< 21 22 23 24 25 26 27
< 28 29 30
<
<
<
<      2024
< Mayo
< do lu ma mi ju vi sa
< 1 2 3 4
< 5 6 7 8 9 10 11
< 12 13 14 15 16 17 18
< 19 20 21 22 23 24 25
< 26 27 28 29 30 31
<
<
<
<      2024
< Junio
< do lu ma mi ju vi sa
< 1
< 2 3 4 5 6 7 8
< 9 10 11 12 13 14 15
< 16 17 18 19 20 21 22
< 23 24 25 26 27 28 29
< 30
<
<
<
<
<      2024
< Julio
< do lu ma mi ju vi sa
< 1 2 3 4 5 6
< 7 8 9 10 11 12 13
< 14 15 16 17 18 19 20
< 21 22 23 24 25 26 27
< 28 29 30 31
<
<
<
<      2024
< Agosto
< do lu ma mi ju vi sa
< 1 2 3
< 4 5 6 7 8 9 10
< 11 12 13 14 15 16 17
< 18 19 20 21 22 23 24
< 25 26 27 28 29 30 31
<
<
<
<      2024
< Septiembre
< do lu ma mi ju vi sa
< 1 2 3 4 5 6 7
< 8 9 10 11 12 13 14
< 15 16 17 18 19 20 21
< 22 23 24 25 26 27 28
< 29 30
<
<
<
<
<      2024
< Octubre
< do lu ma mi ju vi sa
< 1 2 3 4 5
< 6 7 8 9 10 11 12
< 13 14 15 16 17 18 19
< 20 21 22 23 24 25 26
< 27 28 29 30 31
<
<
<
<      2024
< Noviembre
< do lu ma mi ju vi sa
< 1 2
< 3 4 5 6 7 8 9
< 10 11 12 13 14 15 16
< 17 18 19 20 21 22 23
< 24 25 26 27 28 29 30
<
<
<
<      2024
< Diciembre
< do lu ma mi ju vi sa
< 1 2 3 4 5 6 7
< 8 9 10 11 12 13 14
< 15 16 17 18 19 20 21
< 22 23 24 25 26 27 28
< 29 30 31

```

Figura 1.20: Comando diff

- 8. Comando cp:** Este comando permite copiar el contenido de los archivos.

## Conclusiones

- En este laboratorio se observa la importancia de la gestión de archivos y directorios en Linux ya que es fundamental para organizar y administrar información.
- Los comandos mencionados anteriormente son herramientas importantes que permiten manipular y controlar los archivos y directorios de manera eficiente.

## Referencias

Marco. (s.f.). *MANEJO DE ARCHIVOS y DIRECTORIOS a TRAVÉS DE COMANDOS*. [https://www.investigacion.frc.utn.edu.ar/labsis/publicaciones/apunte\\_linux/mmadi.html](https://www.investigacion.frc.utn.edu.ar/labsis/publicaciones/apunte_linux/mmadi.html)

10 – Comandos básicos para gestión de archivos. (s.f.). <https://www.valenciatech.com/10-comandos-basicos-pa>

## 1.4. Laboratorio 4

### 1.4.1. Programación en Shell Script

#### Objetivos:

- El estudiante comprendera en que consiste este lenguaje de de programación.
- El estudiante automatizara procesos programados con shell script.
- El estudiante podrá crear comandos Linux propios.

#### Recursos:

- PC
- MS-Windows
- VirtualBox
- Ubuntu - .ISO

### Desarrollo

Shell Script es un lenguaje de programación interprete de comandos que tiene como objetivo automatizar procesos mediante la creación secuencias de código contenidos dentro de un programa llamado script. Los scripts de shell pueden ser utilizados para realizar una amplia gama de tareas, como el procesamiento de archivos, la administración del sistema, la manipulación de datos y la automatización de flujos de trabajo.

#### Tipos de shell script

- **C Shell:** El shell C es un intérprete de mandatos interactivo y un lenguaje de programación de mandatos. Utiliza una sintaxis que es similar al lenguaje de programación C.
- **Korn Shell:** El shell Korn (mandato ksh) es compatible con las versiones anteriores del shell Bourne (mandato bsh) y contiene la mayoría de las características del shell Bourne así como algunas de las mejores características del shell C.
- **BASH Shell (Bourne Again Shell):** Es una mejora del Bourne Shell y es el shell predeterminado en la mayoría de las distribuciones de Linux.

#### Tipos de editores

- **Vin:** Vim es la version mejorada del editor de texto Vi. La principal característica tanto de Vim como de Vi consiste en que disponen de diferentes modos entre los que se alterna para realizar ciertas operaciones. La aternacia de estos modos ofrecen una gran versatilidad a la hora de editar el codigo que nos mejora enormemente la eficiencia de la edicion del texto .
- **Nano:** GNU nano es un editor de textos de comandos muy extendido que se incluye en la mayoría de las distribuciones de Linux. La interfaz es comparable a la de los editores de texto con interfaz gráfica, lo que hace que nano sea una opción muy apreciada por quienes consideran que los comandos de vi o emacs no son intuitivos.

### Requerimientos



- Editor de texto plano nano.
- Crear el programa con extensión sh.
- Ejecutar el programa.

#### Ejercicio 1 Primer programa

```
#!/bin/bash
#Program: Hello World
#Author: Gustavo Aguas y Sebastian Paucar
#Date: Mayo 22th, 2024
echo "Hello World"
echo "I love Linux"
echo "Version: "
# End Program
```

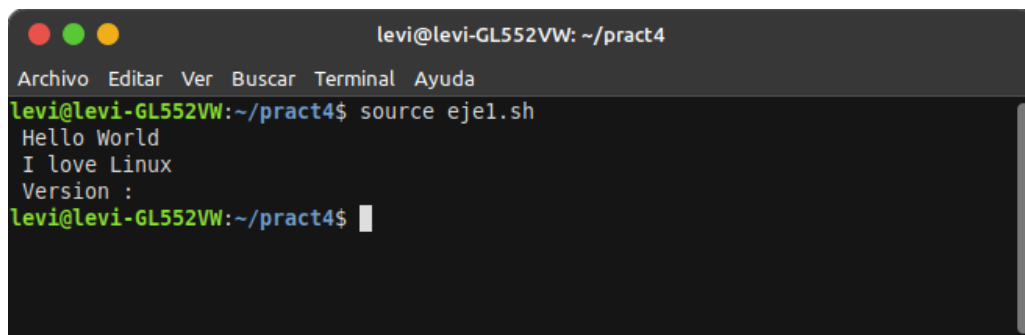


Figura 1.21: Programa Shell: Hello World

#### Ejercicio 2 Permitir visualizar el árbol de directorios de Linux y el estado de archivos de mi carpeta

##### The Kernel version

```
#!/bin/bash
#Program: The kernel version
#Author: Gustavo Aguas y Sebastian Paucar
#Date: Nov 22th, 2023
clear
echo "The kernel information"
echo "The kernel version"; uname -v
echo "The kernel release": uname -r
echo "The kernel distro"; uname -o
echo "The kernel procesor" ; uname -p
echo "Bye,I love Linux"
```

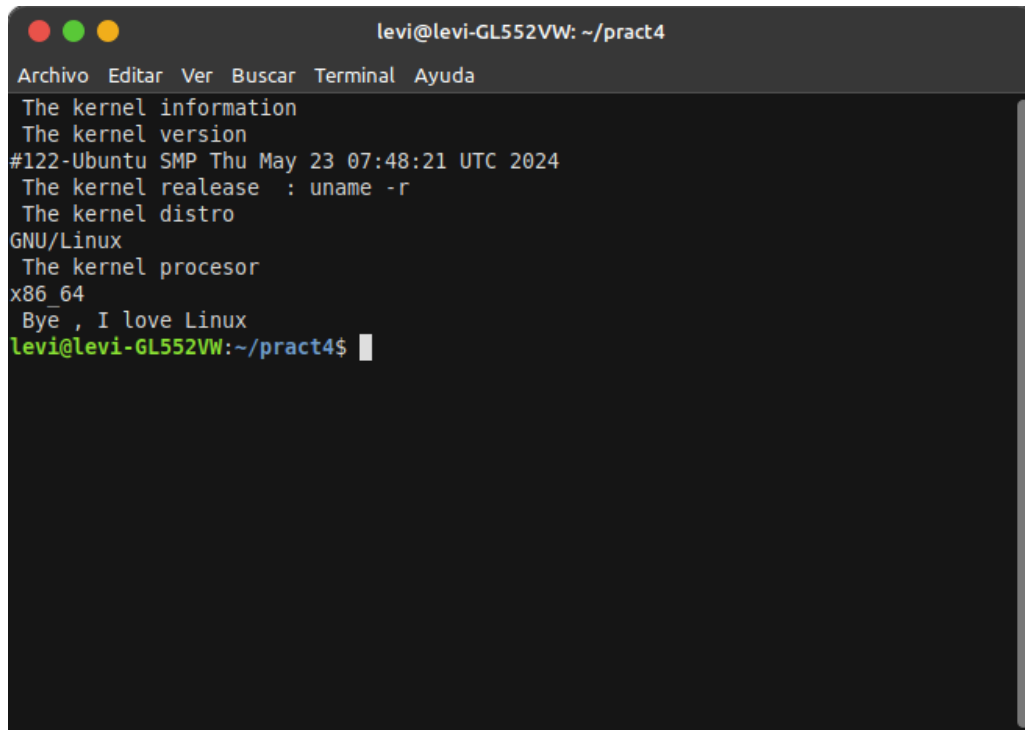
A screenshot of a terminal window with a dark background. The title bar at the top shows three colored circles (red, green, yellow) and the text 'levi@levi-GL552VW: ~/pract4'. Below the title bar is a menu bar with the options 'Archivo', 'Editar', 'Ver', 'Buscar', 'Terminal', and 'Ayuda'. The terminal content displays the following text: 'The kernel information', 'The kernel version', '#122-Ubuntu SMP Thu May 23 07:48:21 UTC 2024', 'The kernel realease : uname -r', 'The kernel distro', 'GNU/Linux', 'The kernel procesor', 'x86\_64', and 'Bye , I love Linux'. The prompt 'levi@levi-GL552VW:~/pract4\$' is shown at the bottom with a cursor.

Figura 1.22: Programa Shell: The kernel version

## Conclusiones

- Los scripts de shell permiten a los usuarios crear secuencias de comandos para llevar a cabo tareas específicas de manera eficiente.
- Es importante comprender los conceptos básicos de la programación en shell, como las estructuras de control, las variables y los comandos integrados, para poder aprovechar al máximo esta herramienta.

## 1.5. Laboratorio 5

### 1.5.1. Tema: Permisos de archivos y directorios

#### Objetivos:

- Comprender que los archivos y directorios requieren de permisos de escritura(r), lectura(w) y ejecución(x) para su utilización.
- Aprender a asignar/quitar permisos

#### Desarrollo

La estructura de un archivo se compone de la siguiente manera:

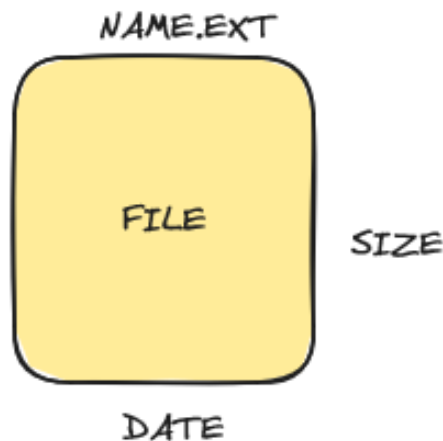


Figura 1.23: Estructura de un archivo

Los tipos de archivos a los que se les puede asignar permisos son los archivos, directorios y link, se los puede identificar por `-d` y `-l` respectivamente:

- `-rwx-rwx-r--permisos1.sh -¿file`
- `d r-x rwx rwx America -¿directorios`
- `l r-x rwx rwx America -¿link`

#### Tipos de permisos:

- `r`: read
- `w`: write
- `x`: execution

**Tipos de archivos:**

- -: file
- d: directory
- l: link
- p: pipe
- s: socket
- b: block

**Tipos de propietarios:**

- u: user
- g: group
- o: others
- a: all

**Comandos Linux para otorgar permisos:**

- chmod: Permite otorgar permichos a los archivos y directorios pero solo del u propietario.
- chgrp:
- chown

**Formas de otorgar permisos:**

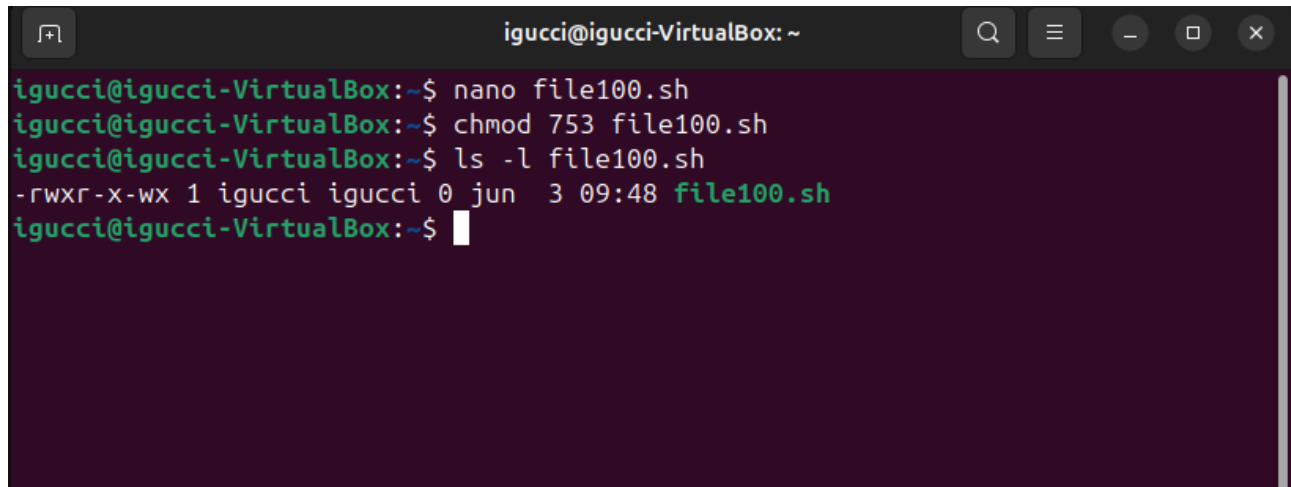
Numérica (Absoluta)

Número	r (read)	w (write)	x (execute)
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

Cuadro 1.1: Tabla de permisos en notación octal para archivos en Linux.

**Ejercicio1: Otorgue permisos de r al u/g, permiso de w al u/o, permiso de ejecucion a todos**

- chmod 753 file100
- 111 101 011



```

igucci@igucci-VirtualBox: ~
igucci@igucci-VirtualBox:~$ nano file100.sh
igucci@igucci-VirtualBox:~$ chmod 753 file100.sh
igucci@igucci-VirtualBox:~$ ls -l file100.sh
-rwxr-x-wx 1 igucci igucci 0 jun  3 09:48 file100.sh
igucci@igucci-VirtualBox:~$

```

Figura 1.24: Ej 1

**Ejercicio2:** Otorgue permisos de lectura y escritura al archivo file101 solamente a usuarios propietarios y al grupo. Compruebe si puede modificar el archivo

- `chmod 660 file101`
- `110 110 000`

**Ejercicio3:** Dado el archivo file102 se despliegan los permisos 764. Cuales son los permisos asignados?

- `111 110 100`
- `rwX rw- r-`

**Ejercicio4:** Dado el archivo file103 otorgue permisos de lectura y ejecucion a todos

- `chmod 555 file103`
- `101 101 101`

### Permisos Simbólicos

Símbolo	Descripción
u	user
g	group
o	others
a	all
r	read
w	write
x	execution
=	asignación
+	incremento
-	eliminación

Cuadro 1.2: Significado de los símbolos para otorgar permisos en Linux Shell.

**Ejercicio5: chmod a+x file104**

- añade el permiso de ejecución (x) para el propietario, el grupo y otros, manteniendo los permisos existentes de lectura (r) y escritura (w).

**Ejercicio6: chmod go-x file105**

- elimina el permiso de ejecución (x) para el grupo (g) y otros (o), mientras mantiene intactos los permisos de lectura (r) y escritura (w) que ya existían. No afecta a los permisos del propietario del archivo.

**Ejercicio8: chmod u-x, go-r file106**

- u-x: Elimina el permiso de ejecución (x) para el propietario (u).
- go-r: Elimina el permiso de lectura (r) para el grupo (g) y otros (o).

**Ejercicio9: chmod u-w, go-rwx file107**

- u-w: Elimina el permiso de escritura (w) para el propietario (u).
- go-rwx: Elimina todos los permisos (lectura r, escritura w y ejecución x) para el grupo (g) y otros (o).

**Ejercicio10: chmod a+r, u+w file108**

- a+r: Agrega el permiso de lectura (r) para todos los usuarios.
- u+w: Agrega el permiso de escritura (w) solo para el propietario.

**Ejercicio11: chmod u-rw, g+w, o+x file109**

- u-rw: Elimina los permisos de lectura y escritura para el propietario.
- g+w: Agrega el permiso de escritura para el grupo.
- o+x: Agrega el permiso de ejecución para otros.

**1.5.2. Funciones en Linux Shell**

Las funciones se encargan de:

- Depurar
- Organizar
- Modular

**Ejemplo 1 de funciones en bash**

Calculadora de funciones

Listing 1.1: Calculadora con funciones

```
#!/bin/bash
```

```
#Author: Gustavo Aguas y Sebastian Paucar
```

```
#Date: 03/06/2024

#Program: Calculadora con funciones

suma(){
    suma=$(echo "scale=2;$num1+$num2"|bc)
    echo "La suma es: $suma"
}

resta(){
    resta=$(echo "scale=2;$num1-$num2"|bc)
    echo "La resta es: $resta"
}

multiplicar(){
    multiplicar=$(echo "scale=2;$num1*$num2"|bc)
    echo "La multiplicacion: $multiplicar"
}

division(){
    divi=$(echo "scale=2;$num1/$num2"|bc)
    echo "La division: $divi"
}

potencia(){
    potencia=$(echo "scale=2;$num1^$num2"|bc)
    echo "La potencia es: $potencia"
}

echo "Ingrese el primer numero"
read num1;

echo "Ingrese el segundo numero"
read num2;

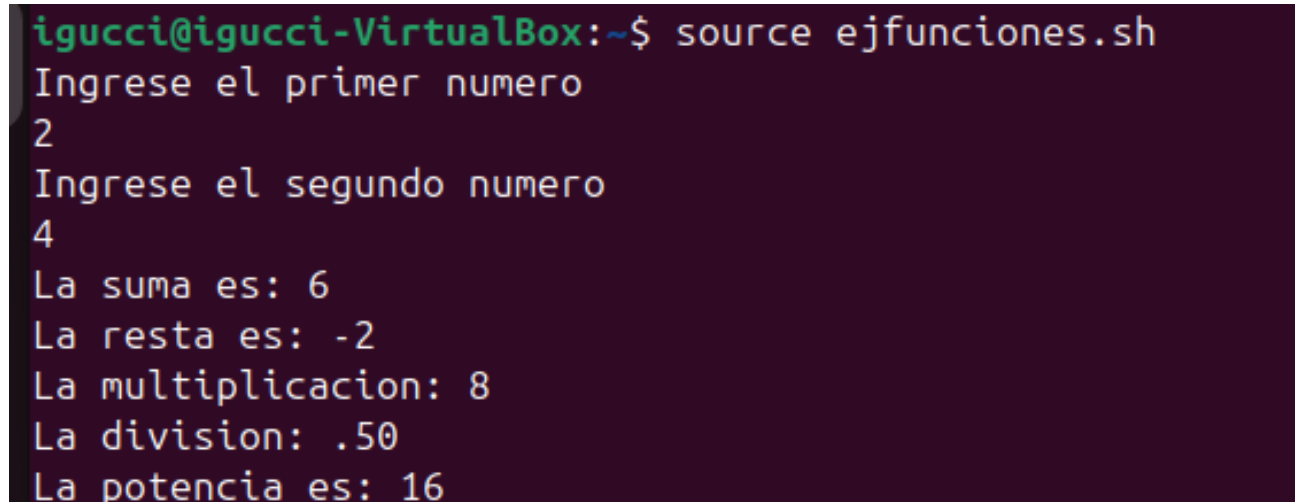
suma
resta
multiplicar
division
potencia

#I LOVE LINUX
```

## Ejemplo 2 de funciones en bash

Listing 1.2: Calculadora con funciones

```
#!/bin/bash
```



```
igucci@igucci-VirtualBox:~$ source ejfunciones.sh
Ingrese el primer numero
2
Ingrese el segundo numero
4
La suma es: 6
La resta es: -2
La multiplicacion: 8
La division: .50
La potencia es: 16
```

Figura 1.25: Ej Funciones

```
#Author: Gustavo Aguas

#Date: 03/06/2024

#Program: Calculadora con funciones

suma(){
    suma=$(echo "scale=2;$num1+$num2"|bc)
    echo "La suma es: $suma"
}

resta(){
    resta=$(echo "scale=2;$num1-$num2"|bc)
    echo "La resta es: $resta"
}

multiplicar(){
    multiplicar=$(echo "scale=2;$num1*$num2"|bc)
    echo "La multiplicacion: $multiplicar"
}

division(){
    divi=$(echo "scale=2;$num1/$num2"|bc)
    echo "La division: $divi"
}

potencia(){
    potencia=$(echo "scale=2;$num1^$num2"|bc)
    echo "La potencia es: $potencia"
}
```



```
echo "Ingrese el primer numero"
read num1;

echo "Ingrese el segundo numero"
read num2;

suma
resta
multiplicar
division
potencia
```

## Conclusiones

- El conocer como otorgar permisos a ciertos directorios y archivos de Linux es de vital importancia para la seguridad de el sistema operativo donde se este trabajando, por lo que es una funcionalidad importante que incluye Ubuntu.
- La programación de funciones ayuda a quitar redundancias en instrucciones de en la programación de codigo, por lo que es útil para utilizar en mismo proceso en varias instrucciones del código.