



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de Docencia

Laboratorios de Computación Salas A y B

Profesor: **Ing. Karina García Morales**

Asignatura: **Fundamentos de Programación (L)**

Grupo: **22**

No. de práctica(s): **Práctica 2. GNU/Linux.**

Integrante(s): **Jonathan Enrique Álvarez Hernández**

No. de lista o brigada: **2**

Semestre: **2026-1**

Fecha de entrega: **Martes 2 de Septiembre de 2025**

Observaciones:

CALIFICACIÓN: _____

GNU/Linux

Objetivo:

- El alumno identificará al sistema operativo como una parte esencial de un sistema de cómputo. Explorará un sistema operativo GNU/Linux con el fin de conocer y utilizar sus comandos básicos.

Desarrollo de la práctica:

Un 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. Además, funciona como interfaz entre la computadora y el usuario o aplicaciones.

Algunos ejemplos de sistemas operativos que existen en la actualidad son:

- Para equipos de cómputo:
 - Windows.
 - Linux.
 - Mac Os.
- Para dispositivos móviles:
 - Android.
 - IOS.
 - Windows Phone.

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

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

Comúnmente, estos componentes se encuentran en el kernel o núcleo del sistema operativo.

En cuanto a la interfaz con el usuario, las hay de tipo gráfico y de tipo texto.

[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 de éste.

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.

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.

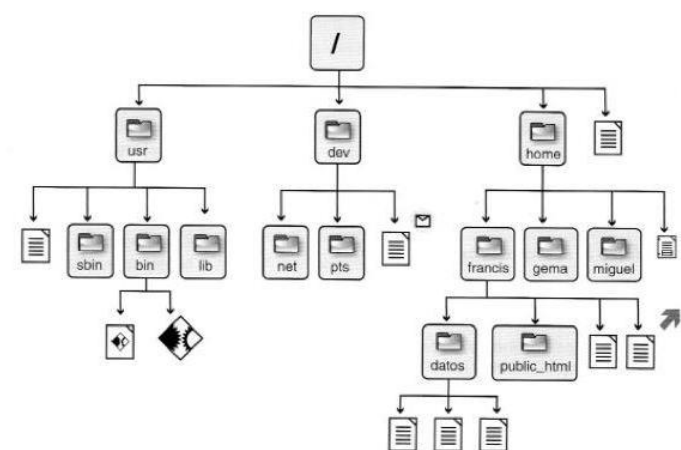


Figura 2: Una parte del sistema de archivos jerárquico en GNU/Linux.

Comandos básicos:

Comando	Descripción
<i>ls</i>	Permite listar los elementos que existen en alguna ubicación del sistema de archivos de Linux. Su sintaxis es: <ul style="list-style-type: none"> <code>ls</code>
<i>touch</i>	Permite crear un archivo de texto, su sintaxis es la siguiente: <ul style="list-style-type: none"> <code>touch nombre_archivo[.ext]</code>
<i>mkdir</i>	Permite crear una carpeta, su sintaxis es la siguiente: <ul style="list-style-type: none"> <code>mkdir nombre_carpeta</code>
<i>cd</i>	Permite ubicarse en una carpeta, su sintaxis es la siguiente: <ul style="list-style-type: none"> <code>cd nombre_carpeta</code>
<i>pwd</i>	Permite conocer la ubicación actual(ruta), su sintaxis es la siguiente: <ul style="list-style-type: none"> <code>pwd</code>
<i>find</i>	Permite buscar un elemento dentro del sistema de archivos, su sintaxis es la siguiente: <ul style="list-style-type: none"> <code>find . -name cadena_buscar</code>
<i>clear</i>	Permite limpiar la consola o terminal, su sintaxis es la siguiente: <ul style="list-style-type: none"> <code>clear</code>
<i>cp</i>	Permite copiar un archivo, su sintaxis es la siguiente: <ul style="list-style-type: none"> <code>cp archivo_origen archivo_destino</code>
<i>mv</i>	Mueve un archivo de un lugar a otro, en el sistema de archivos; su sintaxis es la siguiente: <ul style="list-style-type: none"> <code>mv ubicación_origen/archivo ubicación_destino</code>
<i>rm</i>	Permite eliminar un archivo o un directorio. Su sintaxis es la siguiente: <ul style="list-style-type: none"> <code>rm nombre_archivo</code> <code>rm nombre_carpeta</code>

Ejercicio de tarea:

1. Crea un directorio que se llame *LAB2026-1_Nombre* y entra a el.

```
Loading...

Welcome to Fedora 33 (riscv64)

[root@localhost ~]# mkdir LAB2026-1_JEAH
[root@localhost ~]# cd LAB2026-1_JEAH
[root@localhost LAB2026-1_JEAH]# pwd
/root/LAB2026-1_JEAH
[root@localhost LAB2026-1_JEAH]# cd ..
[root@localhost ~]#
```

2. Crea los directorios necesarios de las materias que estas cursando (mínimo 3) llamado *Materia*.

```
Loading...

Welcome to Fedora 33 (riscv64)

[root@localhost ~]# mkdir LAB2026-1_JEAH
[root@localhost ~]# cd LAB2026-1_JEAH
[root@localhost LAB2026-1_JEAH]# pwd
/root/LAB2026-1_JEAH
[root@localhost LAB2026-1_JEAH]# cd ..
[root@localhost ~]# cd LAB2026-1_JEAH
[root@localhost LAB2026-1_JEAH]# mkdir ÁLGEBRA
[root@localhost LAB2026-1_JEAH]# mkdir REDACCIÓN
[root@localhost LAB2026-1_JEAH]# mkdir PRODUCTIVIDAD
[root@localhost LAB2026-1_JEAH]# ls -l
ÁLGEBRA
PRODUCTIVIDAD
REDACCIÓN
[root@localhost LAB2026-1_JEAH]#
```

3. Dentro de cada directorio crea un archivo llamado *Nombre_PENDIENTES_Materia* (recuerda que puedes utilizar los comandos vistos en la práctica).
4. Muestra el contenido de cada uno de tus directorios creados dentro de *LAB2026-1_Nombre*.
5. Salir del directorio *LAB2026-1_Nombre*.

```
[root@localhost LAB2026-1_JEAH]# cd ÁLGEBRA
[root@localhost ÁLGEBRA]# touch JEAH_PENDIENTES_ÁLGEBRA.txt
[root@localhost ÁLGEBRA]# ls
JEAH_PENDIENTES_ÁLGEBRA.txt
[root@localhost ÁLGEBRA]# cd ..
[root@localhost LAB2026-1_JEAH]# cd REDACCIÓN
[root@localhost REDACCIÓN]# touch JEAH_PENDIENTES_REDACCIÓN.txt
[root@localhost REDACCIÓN]# ls
JEAH_PENDIENTES_REDACCIÓN.txt
[root@localhost REDACCIÓN]# cd ..
[root@localhost LAB2026-1_JEAH]# cd PRODUCTIVIDAD
[root@localhost PRODUCTIVIDAD]# touch JEAH_PENDIENTES_PRODUCTIVIDAD
[root@localhost PRODUCTIVIDAD]# ls
JEAH_PENDIENTES_PRODUCTIVIDAD
[root@localhost PRODUCTIVIDAD]# cd ..
[root@localhost LAB2026-1_JEAH]# cd ..
[root@localhost ~]#
```

6. Crea un directorio que se llame *Nombre_COPIA*.

```
[root@localhost ~]# mkdir JEAH_COPIA
[root@localhost ~]# ls
bench.py hello.c JEAH_COPIA LAB2026-1_JEAH
[root@localhost ~]#
```

7. Copia los archivos que creaste, al directorio *Nombre_COPIA*.
8. Muestra el contenido de tu directorio *Nombre_COPIA*.

```
[root@localhost ~]# cp LAB2026-1_JEAH/ÁLGEBRA/JEAH_PENDIENTES_ÁLGEBRA.txt JEAH_COPIA
[root@localhost ~]# cp LAB2026-1_JEAH/REDACCIÓN/JEAH_PENDIENTES_REDACCIÓN.txt JEAH_COPIA
[root@localhost ~]# cp LAB2026-1_JEAH/PRODUCTIVIDAD/JEAH_PENDIENTES_PRODUCTIVIDAD.txt JEAH_COPIA
cp: cannot stat 'LAB2026-1_JEAH/PRODUCTIVIDAD/JEAH_PENDIENTES_PRODUCTIVIDAD.txt': No such file or directory
[root@localhost ~]# cp LAB2026-1_JEAH/PRODUCTIVIDAD/JEAH_PENDIENTES_PRODUCTIVIDAD JEAH_COPIA
[root@localhost ~]# cd JEAH_COPIA
[root@localhost JEAH_COPIA]# ls
JEAH_PENDIENTES_ÁLGEBRA.txt JEAH_PENDIENTES_PRODUCTIVIDAD JEAH_PENDIENTES_REDACCIÓN.txt
[root@localhost JEAH_COPIA]#
```

9. Mueve el directorio *Nombre_COPIA* al directorio *LAB2026-1_Nombre*.

```
[root@localhost JEAH_COPIA]# cd ..
[root@localhost ~]# mv JEAH_COPIA LAB2026-1_JEAH
[root@localhost ~]# ls
bench.py hello.c LAB2026-1_JEAH
[root@localhost ~]# cd LAB2026-1_JEAH
[root@localhost LAB2026-1_JEAH]# ls
ÁLGEBRA JEAH_COPIA PRODUCTIVIDAD REDACCIÓN
[root@localhost LAB2026-1_JEAH]#
```

10. Muestra los permisos de los archivos contenidos en el directorio *LAB2026-1_Nombre* y en *Nombre_COPIA*, explica cada uno de los permisos (investigar comando *chown* y *chmod*).

```
[root@localhost ~]# ls -la LAB2026-1_JEAH
total 24
drwxr-xr-x 5 root root 146 Sep  2 12:41 .
dr-xr-x--- 4 root root 274 Dec 26  2020 ..
drwxr-xr-x 2 root root  82 Sep  2 12:46 ÁLGEBRA
drwxr-xr-x 2 root root 175 Sep  2 12:57 JEAH_COPIA
drwxr-xr-x 2 root root  83 Sep  2 12:47 PRODUCTIVIDAD
drwxr-xr-x 2 root root  84 Sep  2 12:47 REDACCIÓN
[root@localhost ~]#
```

- chown: Se usa para cambiar el propietario y/o el grupo de archivos y directorios. Su sintaxis es:
 - `chown [opciones] <nuevo_propietario>[:<nuevo_grupo>] <archivo_o_directorio>`
- chmod: Sirve para cambiar los permisos de lectura, escritura y ejecución de archivos y directorios, controlando así quién puede acceder a ellos y qué puede hacer con ellos. Su sintaxis en modo simbólico y en modo numérico (octal) son respectivamente:
 - `chmod u+w archivo.txt`
 - `chmod 644 archivo.txt`

11. Indica el directorio en el que te encuentras y con que comando lo muestras.

```
[root@localhost ~]# pwd
/root
[root@localhost ~]#
```

Se indica con el comando *pwd*.

12. Teclea el comando cal y escribe lo que muestra.

```
[root@localhost ~]# cal
September 2025
Su Mo Tu We Th Fr 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
[root@localhost ~]#
```

Se muestra un calendario del mes en curso.

13. Teclea el comando date y escribe la salida.

```
[root@localhost ~]# date
Tue Sep  2 01:24:24 PM UTC 2025
[root@localhost ~]#
```

Los datos de salida corresponden a la fecha (día y mes) junto con la hora actual.

14. Describe para que empleas el comando man.

Permite visualizar la descripción de cualquier comando, así como la manera en la que se puede utilizar.

```

[root@localhost ~]# man ls
LS(1)                                User Commands                                LS(1)
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 al-
    phabetically if none of -cftuvSUX nor --sort is specified.

    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

    --block-size=SIZE
        with -l, scale sizes by SIZE when printing them; e.g., '--block-size=M'; see SIZE
        format below

    -B, --ignore-backups

```

15. Investiga para que se utiliza el comando cat.

Se usa para mostrar el contenido de archivos de texto, crear archivos nuevos, unir varios archivos en uno solo y añadir texto a archivos existentes. Se utiliza con la sintaxis:

- cat [opciones] [archivo(s)]

Conclusiones:

Personalmente, me parece que esta práctica es un buen ejercicio de introducción a la programación más compleja que se utiliza en la práctica profesional del ingeniero. Considero que esta no fue muy complicada ya que parte de comandos precisamente básicos que ayudan a explorar las nociones básicas del sistema Linux. Sin embargo, lo anterior no entra conflicto con el hecho de que se pueda encontrar bastante aprendizaje en el hecho de realizar esta práctica ya que seguramente a lo largo del semestre se profundizará cada vez más en formas, lenguajes y sistemas de programación cada vez más complejos.

Creo que esta práctica ya entra un poco más en el terreno práctico propiamente de la programación en comparación con su antecesora, la práctica 1, donde se ofrecía como aprendizaje un panorama general de lo que implica usar la computadora en la práctica ingenieril contemporánea, así como para solucionar problemas cotidianos en la actualidad.

Fuentes de consulta:

- B. G. (2023). *Comando cat de Linux: para qué sirve y ejemplos de uso*. Hostinger. Recuperado el 2 de septiembre de 2025 de <https://www.hostinger.com/mx/tutoriales/comando-cat-linux#:~:text=el%20comando%20Cat-,%C2%BFQu%C3%A9%20es%20el%20comando%20cat%20de%20Linux?,txt%E2%80%9D>.
- Facultad de Ingeniería. (2025). Manual de prácticas del laboratorio de Fundamentos de Programación. Laboratorio de computación. Salas A y B. Universidad Nacional Autónoma de México. pp. 23-35. Recuperado el 1 de septiembre de 2025 de <http://lcp02.fi-b.unam.mx/>
- León, M. (2025). *Uso del comando Chmod y comando Chown para la gestión de servidores*. arsys.com. Recuperado el 1 de septiembre de 2025 de <https://www.arsys.es/blog/comandos-chmod-chown#:~:text=R%20755%20mi%20carpeta-,%C2%BFPara%20qu%C3%A9%20sirve%20el%20comando%20chmod?,escribir%20o%20ejecutar%20determinados%20archivos>.
- Pires, J. (2023). *Cómo Usar el Comando Chown de Linux*. HostGator. Recuperado el 2 de septiembre de 2025 de <https://www.hostgator.mx/blog/comando-chown-linux/>