

# Practica 2: GNU/Linux

## Objetivos:

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.

## Desarrollo:

La práctica nos da una pequeña introducción a lo que son los sistemas operativos:

“El Sistema Operativo es el conjunto de programas y datos que administra los recursos tanto de hardware como de software de un sistema de cómputo y/o comunicación.”

Algunos ejemplos más comunes de sistemas operativos, son los ya conocidos Windows, Linux y Mac Operative System, y para dispositivos móviles como Android y iPhone Operative System.

La practica también nos dio una breve explicación del uso de Linux, que es un sistema operativo tipo Unix de libre distribución para computadoras personales, servidores y estaciones de trabajo, además de distintos tipos de software libres, los cuales no necesitan de una licencia especial y son gratuitos, lo cual tiene muchas ventajas, porque podemos hacer muchas modificaciones del sistema en el momento que nosotros deseemos.

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.

Para comenzar el uso de Linux mediante comandos necesitamos la terminal-consola, y una vez teniendo abierta esta consola, podemos ya empezar a trabajar e introducir comandos, se tienen de distintos tipos de comandos, pero los más usados y más comunes son:

## Comando ls:

1. `ls / ls .` : El comando `ls` permite listar los elementos que existen en alguna ubicación del sistema de archivos de Linux.
2. `ls -l` :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:
3. `ls /:` Es posible listar los elementos que existen en cualquier ubicación del sistema de archivos. Si queremos ver los archivos que se encuentran en a raíz
4. `ls /home` :Para ver los usuarios del equipo local, revisamos el directorio `home` que parte de la raíz (`/`)
5. `ls -l /home` :Tanto las opciones como los argumentos se pueden combinar para generar una ejecución más específica
6. `man ls`: 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.
7. `ls /usr`: Antes de revisar otros comandos, es importante aprender a “navegar” por el sistema de archivos de Linux en modo texto, si deseamos ver la lista de los archivos del directorio `usr`. Esto es, el argumento se inicia con `/` indicando que es el directorio raíz, seguido de `usr` que es el nombre del directorio. Cuando especificamos la ubicación de un archivo partiendo de la raíz, se dice que estamos indicando la “ruta absoluta” del archivo. Existe otra forma de especificar la ubicación de un archivo, esto es empleando la “ruta relativa”.
8. `ls ..` o `ls ../` Si bien el punto (`.`) es para indicar la ubicación actual, el doble punto (`..`) se utiliza para referirse al directorio “padre”. De esta forma si deseamos listar los archivos que dependen de mi directorio padre

Se pueden utilizar varias referencias al directorio padre para ir navegando por el sistema de archivos, de tal manera que se realice la ubicación de un archivo a través de una ruta relativa. De la Figura 2, si nuestra cuenta depende de `home`, la ruta relativa para listar los archivos de del directorio `usr` es:

```
ls ../../usr
```

Con los primeros dos puntos se hace referencia al directorio `home`, con los siguientes dos puntos se refiere al directorio raíz, y finalmente se escribe el nombre del directorio `usr`.

Comando touch:

La principal función del comando touch es crear archivos de texto, (cabe destacar, que en Linux, no es necesario agregar el nombre de la extensión de un archivo, pero es recomendable para saber el formato del archivo) la forma en que se escribe es:

```
touch nombredelarchivo[.txt]
```

En el laboratorio, se creó un archivo para probar esta función, sin la necesidad de tener un nombre específico, esto para practicar lo visto en clase.

Comando mkdir:

El comando mkdir permite crear carpetas, para el traslado o guardado de archivos, en el laboratorio, usamos el comando mkdir para crear una carpeta en la que posteriormente ubicaremos el archivo que creamos usando el comando touch, llamada tareas. La forma en la que se escribe es:

```
mkdir nombre_carpeta
```

Comando cd :

El comando cd permite desplazarse a una carpeta, esto sirve para ubicarse y facilitar el desplazamiento. En el laboratorio, nos desplazamos dentro de la carpeta tareas. La forma en la que se escribe es la siguiente:

```
cd nombre_carpeta
```

Comando pwd:

El comando pwd permite conocer la ubicación actual del usuario. Útil cuando aún no se conoce la interfaz de Linux apropiadamente. Se escribe de la siguiente forma:

```
pwd
```

Comando find:

El comando find permite buscar un archivo, pero es necesario especificarle la ubicación general y el nombre del archivo que se desea buscar. Se escribe en la forma:

```
find . -name cadena_buscar
```

En la que name representa el nombre del archivo a buscar y cadena\_ la ubicación en la que se desea buscar, la cadena.

Comando clear:

El comando clear sirve para ver la consola de Linux limpia, no borra todo lo anteriormente hecho, solo despeja la pantalla para poder tener una buena organización de nuevo. Para usarlo solo se debe escribir clear.

Comando cp:

El comando cp sirve para copiar un archivo en otro, esto, se hace especificando el archivo que se desea copiar, la ubicación y el archivo en el que se va a copiar, se puede incluir la ubicación de la carpeta padre haciendo uso de los dos puntos ,

(..)

La forma en la que se escribe es:

```
cp archivo_origen archivo_destino
```

Comando mv

El comando mv sirve para relocalizar archivo en una nueva ubicación, o en otra anteriormente creada, es necesario especificar localización y archivo solamente. Su sintaxis es la siguiente:

```
mv ubicación_origen/archivo ubicación_destino
```

Comando rm

El comando rm sirve, principalmente para remover un archivo completamente, es de gran utilidad si se desea quitar el archivo de Linux, este comando no lo moverá, sino que lo eliminará. Permite borrar archivos o carpetas. Su sintaxis es:

```
rm nombre_archivo rm nombre_carpeta
```

Todos estos comandos fueron experimentados por nosotros en la clase, para comprobar su uso y aprenderá usarlos correctamente

## Conclusiones:

Utilizamos una interfaz muy útil, pero a la vez un poco complicada de manejar, ya que no estamos acostumbrados a su uso, además de que para redactar una acción, lo tenemos que hacer de una manera más abstracta y literal, complicando así el ser intuitivo. Pero de igual manera, una de sus ventajas es que es una gran herramienta para que los alumnos se empiecen a acostumbrar a los lenguajes de programación y al uso de comandos, que, a pesar de ser mas un sistema operativo que un lenguaje de programación en sí, las estructuras son un poco parecidas; además de que gracias a esta práctica podemos ver conceptos como ubicaciones y navegación de una manera más abstracta y menos gráfica.

## Bibliografía

Práctica numero 2, Facultad de Ingeniería, en Laboratorios A y B (2017), Sitio Web: <http://lcp02.fib.unam.mx/>