

Práctica 2
GNU/Linux

Fundamentos de programación

Integrantes:

Arreola Santander Ricardo
Martínez Castañeda Nosha Alejandro
Miranda Rayón Juan Manuel
Ramírez Bartolo Ignacio
Sánchez Wilhelm Jeremías

Profesor: Jorge Alberto Hernández Nieto

Grupo 19

Equipo 4

Fecha: 22/octubre/2020

UNAM. Facultad de Ingeniería

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

Sistema operativo:

Un sistema operativo es un conjunto de programas que permite manejar la memoria, disco, medios de almacenamiento de información y los diferentes periféricos o recursos de nuestra computadora, como son el teclado, el mouse, la impresora, la placa de red, entre otros.

Dentro de las tareas que realiza el sistema operativo, en particular, se ocupa de gestionar la memoria de nuestro sistema y la carga de los diferentes programas, para ello cada programa tiene una prioridad o jerarquía y en función de la misma contará con los recursos de nuestro sistema por más tiempo que un programa de menor prioridad.

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.
- Envía mensajes a cada aplicación o usuario (o a un operador del sistema) sobre el estado de funcionamiento y los errores que se hayan podido producir.
- En los equipos que pueden proporcionar procesamiento en paralelo, un sistema operativo puede manejar la forma de dividir el programa para que se ejecute en más de un procesador a la vez.

Existen dos tipos de sistemas operativos:

Sistema operativo para PC y Sistema operativo para teléfonos móviles

Los sistemas operativos para PC son muy variados y hay muchos, pero los más utilizados son el Windows, el Mac y el LINUX.

Cada parte del sistema operativo está dividido de acuerdo al tipo de funcionalidad y objetivo que tiene, así también a las áreas del ordenador a las cuales están enfocadas.

Algunos elementos son:

1. Gestión de procesos

Al momento de ejecutar un programa en nuestra computadora, deberá de utilizar procesos para reservar la utilización de los recursos necesarios para su funcionamiento.

2. Gestión de la memoria

La memoria es utilizada por los distintos programas, el sistema operativo se encarga de la correcta utilización de este recurso, pues debe de ser el encargado de saber que partes están siendo utilizadas y asignar espacios cuando se requiera para su liberación.

3. Sistema de entrada y salida

Es el medio por el cual el usuario va a interactuar con los componentes físicos de la máquina. También se encargará de gestionar la comunicación entre los dispositivos, controlar los posibles errores y detectar las interrupciones que se puedan dar.

4. Sistema de archivos

Se refiere a las distintas técnicas para almacenar la información dentro de los discos duros que aloja el ordenador, estas técnicas o formas de almacenamiento se llaman sistema de archivos.

5. Sistemas de protección

Se refiere al control de acceso de los programas estableciendo límites de seguridad, establece quienes pueden tener acceso a determinados recursos estableciendo políticas.

6. Sistema de comunicaciones

Esto se refiere al uso específico de interfaces de red, permitiendo la comunicación entre distintos sistemas, también hace referencia a los medios que son proporcionados para hacer posible la comunicación entre aplicaciones que se ejecutan tanto localmente como remotamente a través de conexiones virtuales.

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.

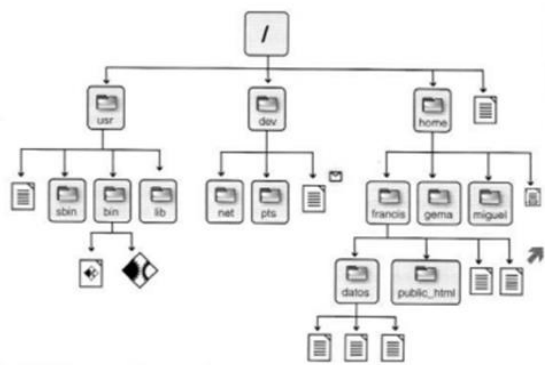


Figura 2: Una parte del sistema de archivos jerárquico en GNU/Linux.

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” qué 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* en este caso *Ubuntu en Windows*.

O bien en el ícono de aplicaciones en la línea de “buscar” escribir “Ubuntu” 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.

Ejemplo (comando **ls**).

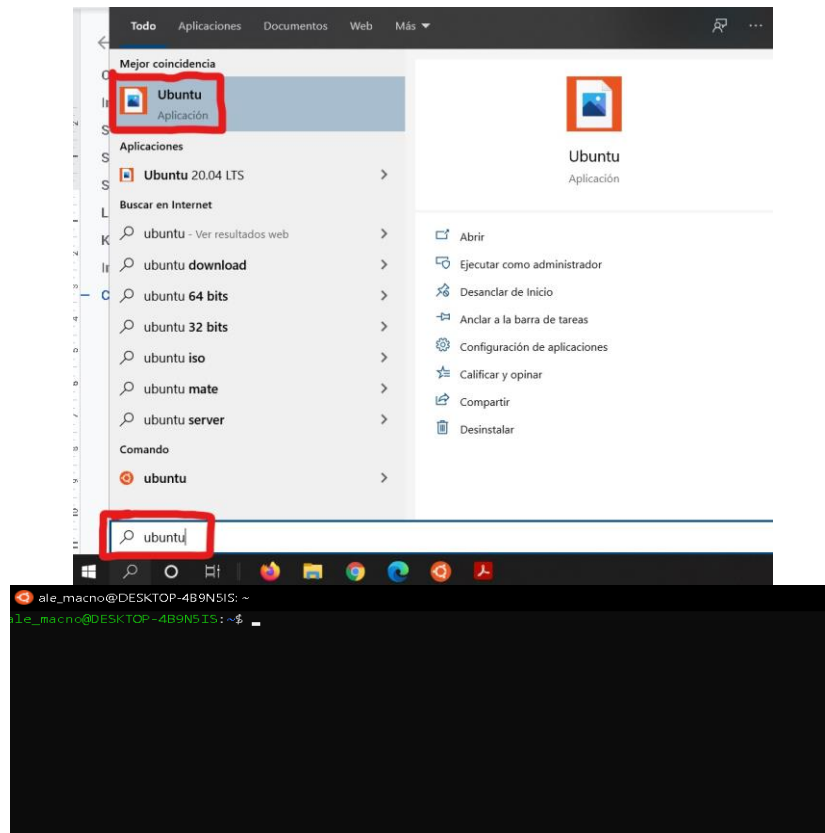


Figura 3: Una parte del terminal de Linux en Ubuntu.

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`

Antes de revisar otros comandos, es importante aprender a “navegar” por el sistema de archivos de Linux en modo texto. Basándonos en la Figura 2 de esta práctica, si deseamos ver la lista de los archivos del directorio usr, podemos escribir el comando: `ls /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”.

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 escribe el siguiente comando: `ls ..` o `ls ../`

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 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.

Ejemplo (comando 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.

Ejemplo (comando 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 “tareas” se escribe el siguiente comando: **mkdir tareas**

Ejemplo (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 ..**

Ejemplo (comando pwd)

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

Ejemplo (comando 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 indicar 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**

Ejemplo (comando clear)

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

Ejemplo (comando cp)

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.

Ejemplo (comando mv)

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 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**

Ejemplo (comando rm)

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.

Actividad

Usando los comandos de Linux realizar las siguientes acciones:

1.- Crear una carpeta

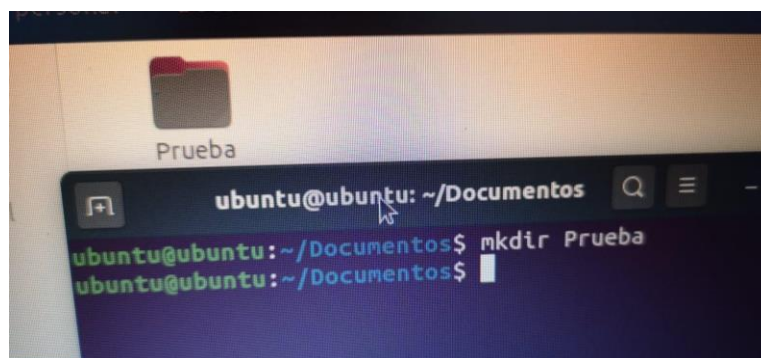
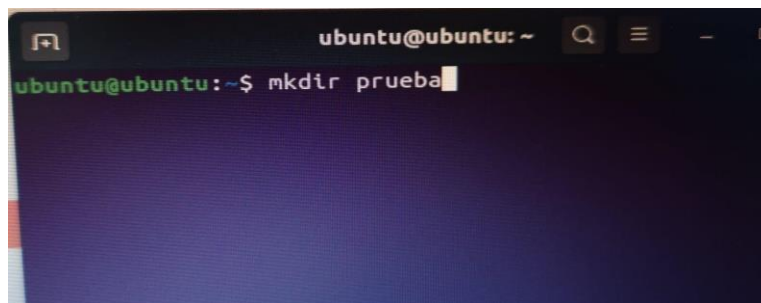
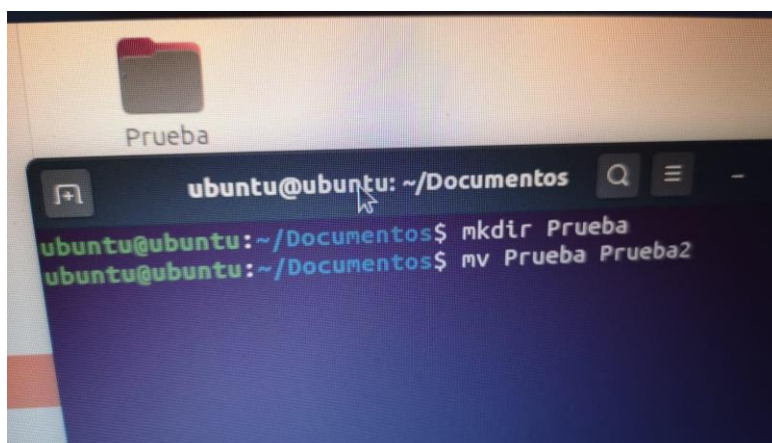


Figura 3 y 4. Se observa como el comando `mkdir` sirve para crear un documento, en este caso se creó una carpeta llamada prueba.

2.- Renombrarla



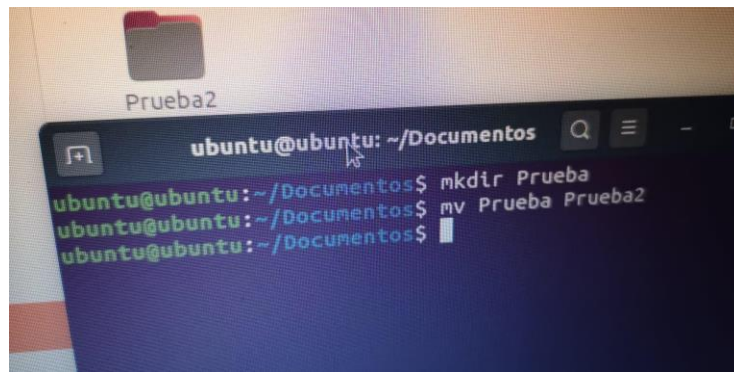


Figura 5 y 6. Al principio la carpeta se llama Prueba y luego de aplicar el comando mv, se cambia la carpeta que se especifique primero para luego escribir el nuevo nombre que se le quiere cambiar a la carpeta.

3.- Revocar el permiso de escritura a un archivo (crear un documento desde alguna aplicación de notas).

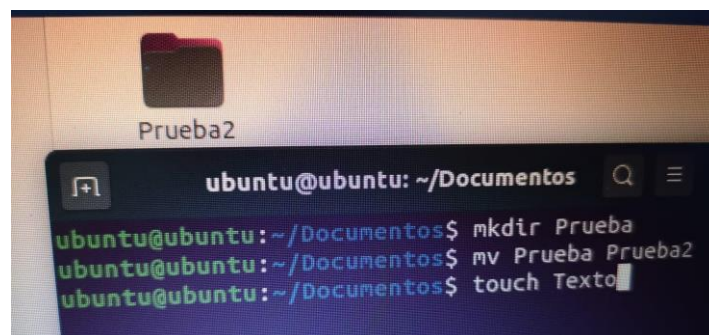


Figura 7. Primero se tiene que crear un documento como notas, y se hace con el comando touch.

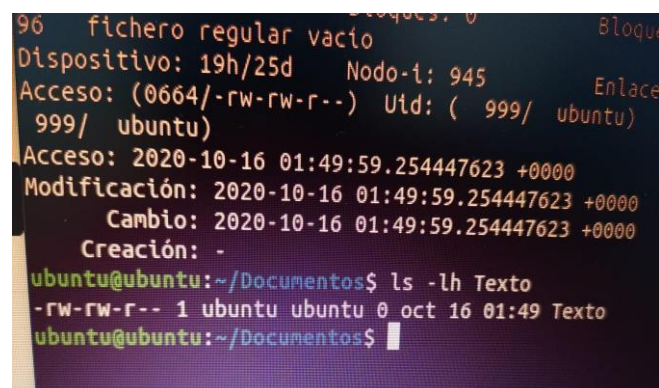


Figura 8. Con el comando ls -lh se procede a poner el nombre del documento para conocer sus permisos.

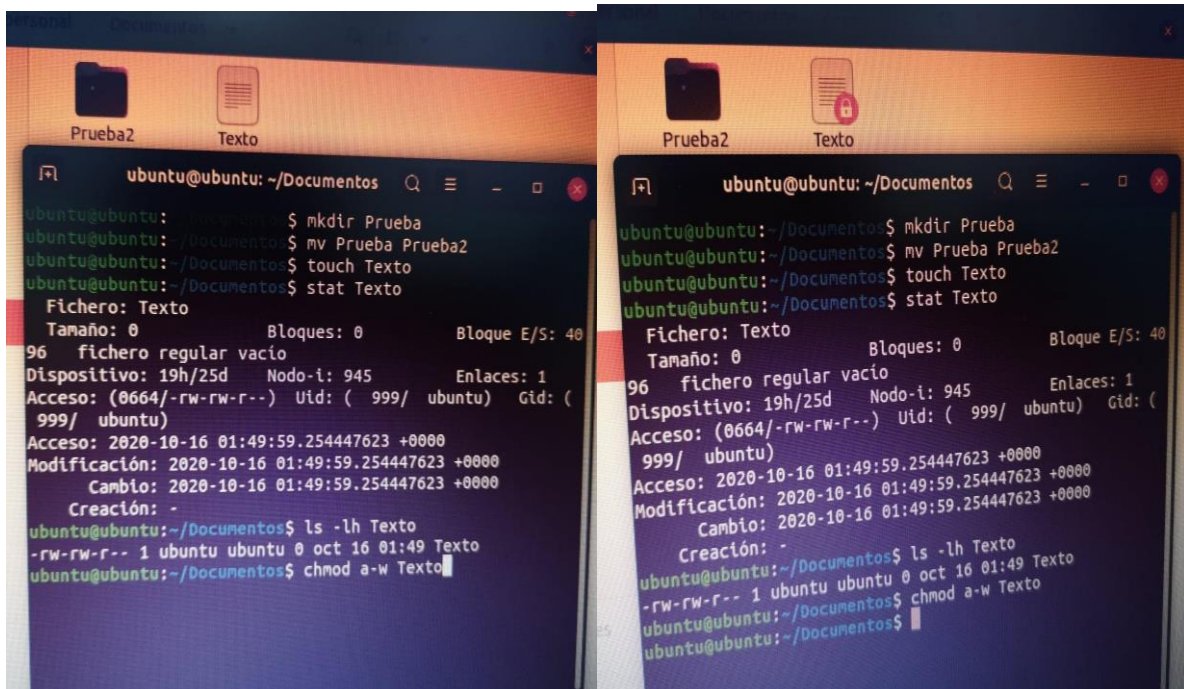


Figura 9 y 10. Con el comando `chmod` se procede a restringir el permiso de escritura con la letra `w` al documento llamado `texto`. Se sabrá que está bloqueado si aparece un candado junto al documento.

4.- Copiar una carpeta de Documentos a Escritorio.

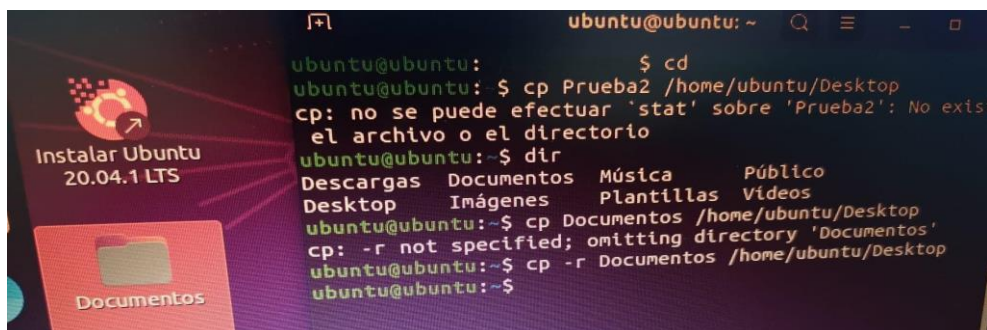
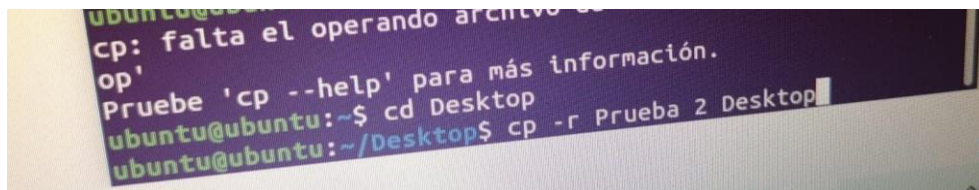


Figura 11 y 12. Con el comando `cp -r` se copia el documento que se especifica y luego para copiarlo al escritorio se escribe `/home/ubuntu/Desktop` (dado que está en inglés), para que se pueda pegar la carpeta al escritorio.

5.- Crear un enlace simbólico a la carpeta.

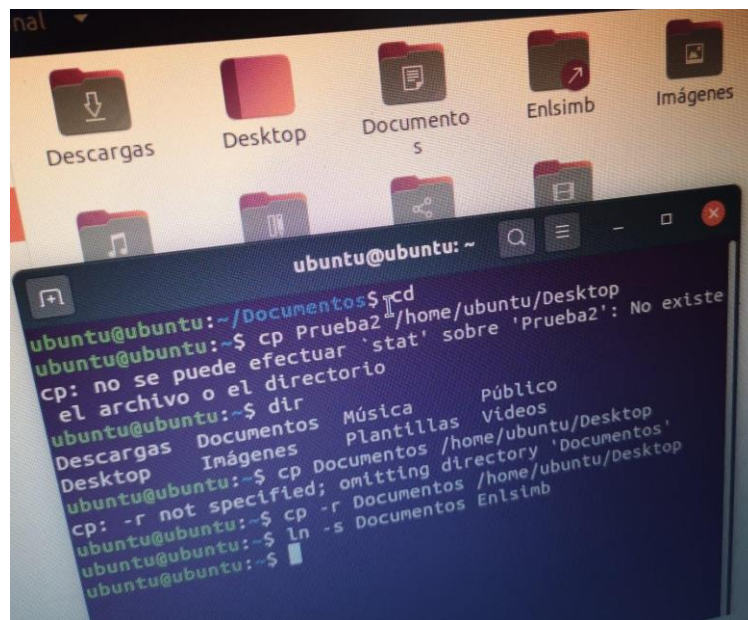
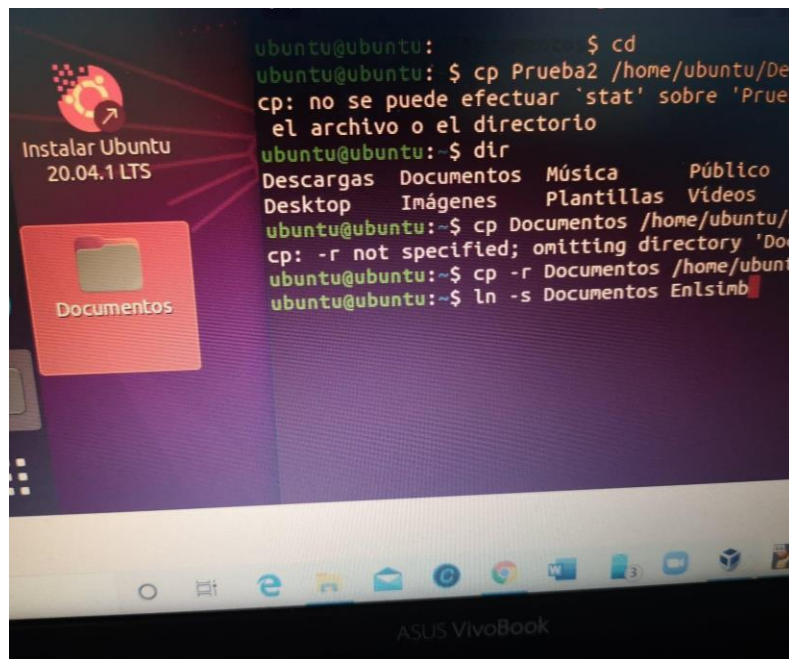
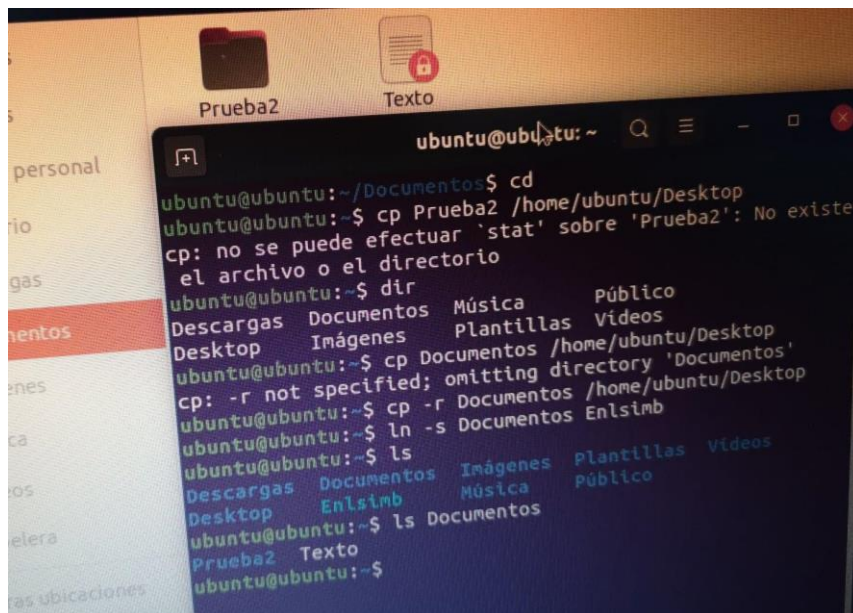


Figura 13 y 14. Con el comando `ln -s` se procede a crear un enlace simbólico de un documento que se especifique. Estos enlaces simbólicos tienen gran similitud con los accesos directos de Windows.

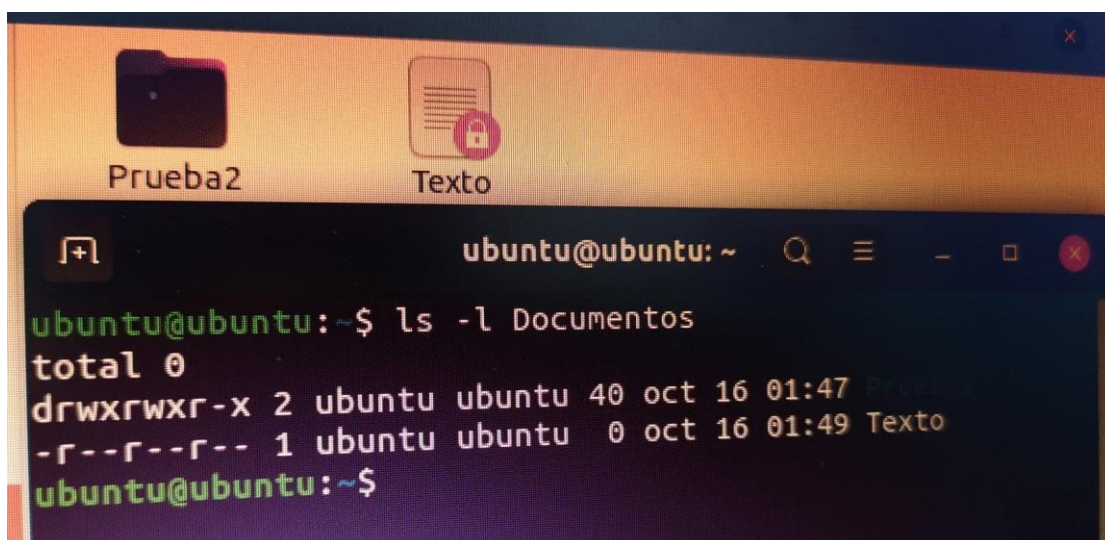
6.- Listar los elementos de una carpeta.



```
ubuntu@ubuntu:~/Documentos$ cd /home/ubuntu/Desktop
ubuntu@ubuntu:~$ cp Prueba2 /home/ubuntu/Desktop
cp: no se puede efectuar 'stat' sobre 'Prueba2': No existe el archivo o el directorio
ubuntu@ubuntu:~$ dir
Descargas  Documentos  Música      Público
Desktop    Imágenes    Plantillas  Videos
ubuntu@ubuntu:~$ cp Documentos /home/ubuntu/Desktop
cp: -r not specified; omitting directory 'Documentos'
ubuntu@ubuntu:~$ cp -r Documentos /home/ubuntu/Desktop
ubuntu@ubuntu:~$ ln -s Documentos Enlsimb
ubuntu@ubuntu:~$ ls
Descargas  Documentos  Imágenes  Plantillas  Videos
Desktop    Enlsimb     Música    Público
ubuntu@ubuntu:~$ ls Documentos
Prueba2  Texto
ubuntu@ubuntu:~$
```

Figura 15. Con el comando ls se procede a lista los elementos de una carpeta, en este caso se listan los elementos de la carpeta Documentos, que contiene la Prueba 2 y el documento llamado Texto.

7.- Mostrar los permisos de la carpeta Documentos.



```
ubuntu@ubuntu:~$ ls -l Documentos
total 0
drwxrwxr-x 2 ubuntu ubuntu 40 oct 16 01:47 Prueba2
-r--r--r-- 1 ubuntu ubuntu  0 oct 16 01:49 Texto
ubuntu@ubuntu:~$
```

Figura 16. Con el comando ls -l se muestran todos los permisos que tiene la carpeta que se especifique, en este caso Documentos, donde se ve que la carpeta Prueba2 tiene permisos de lectura y escritura, mientras que el documento solo tiene permisos de lectura, dado que se le quito los permisos de escritura en el paso 3.

8.- Eliminar la copia de la carpeta ubicada en Documentos.

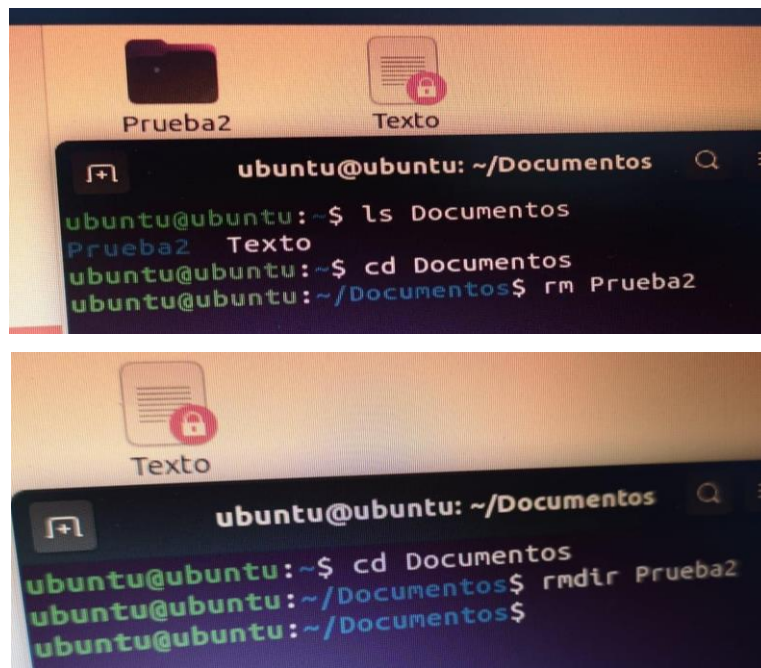


Figura 17 y 18. Con el comando rmdir se procede a eliminar la carpeta que se encontraba en Documentos.

9.- Mostrar en qué parte de la estructura de archivos se encuentra el usuario.

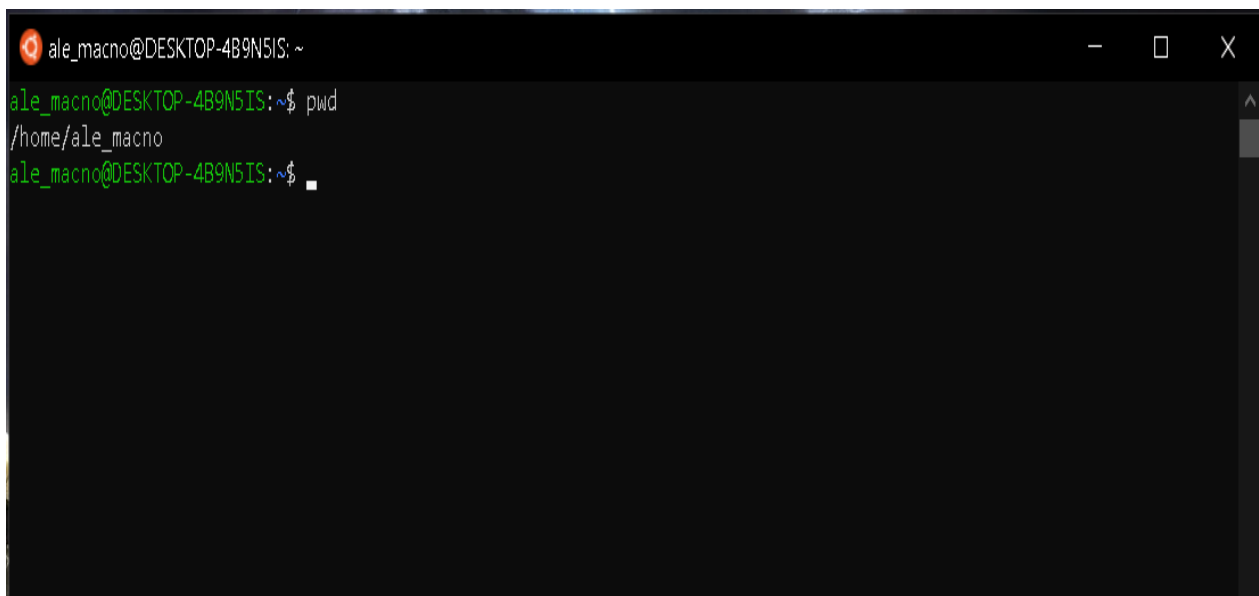


Figura 19. El comando pwd muestra en que parte y la ruta del directorio en donde nos encontramos

10.-Bonus: <https://lamiradadelreplicante.com/2015/04/19/chuu-chuu-dejen-paso-al-tren-de-linux/>

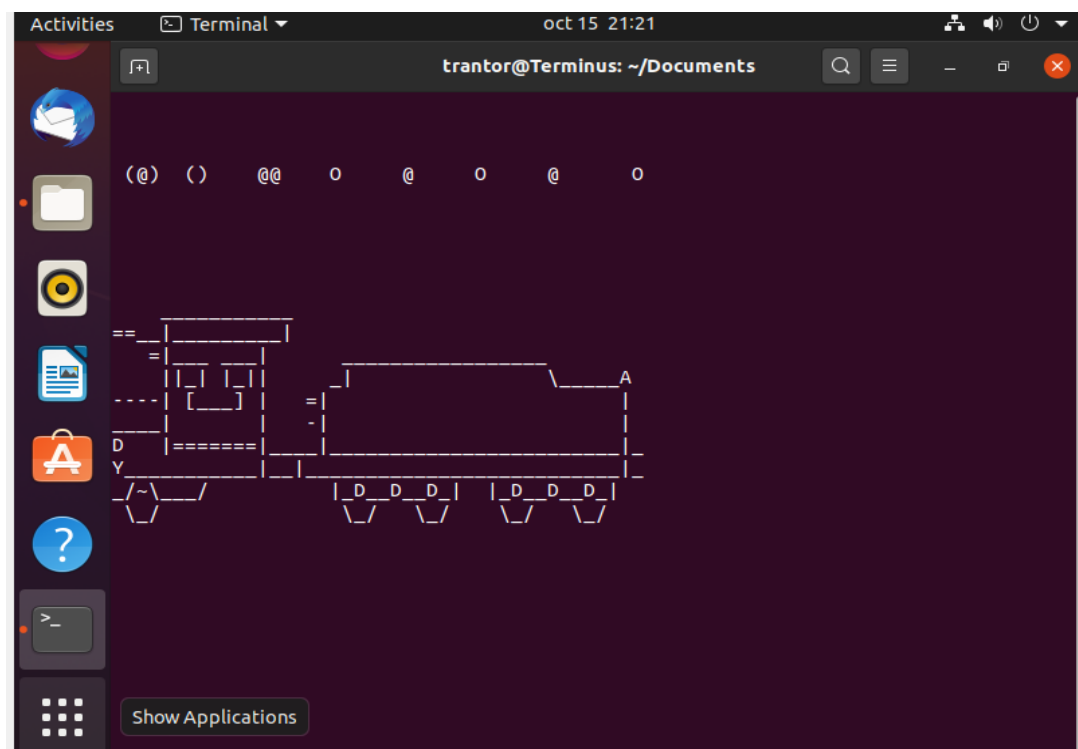


Figura 20. Programa instalado que demuestra la versatilidad de la terminal, al mostrar un tren en movimiento hecho de puros caracteres.

Conclusion

Al realizar esta práctica en el entorno de GNU/Linux por medio de Ubuntu, pudimos ver cómo es instalar un nuevo sistema operativo, acercarnos a lo que es esta plataforma, conocer un poco de sus comandos básicos, y jugar un poco con ellos, y ver unas cuantas de sus funciones que más adelante ayudarán a la creación de nuevos programas, además de lo cual, y como extra y personalizamos un poco la pantalla principal, y para ser un primer acercamiento, consideramos estuvo bien.

A Pesar de que Linux no es un sistema ocupado por mucha gente tiene ventajas ante otros sistemas, así como desventajas, sin embargo, para la programación es de utilidad debido a la facilidad en la que se puede trabajar hoy en día.

De igual manera muestra la versatilidad que el sistema operativo puede llegar a tener para poder realizar tareas sin tener que hacerlas directamente para poder ejecutarlas, si no hacerlas a base de un terminador en el que se pueda realizar con ayuda de unos códigos que, a pesar de que si lleguen a ser algo difíciles, si tienen lógica si la sintaxis que manejamos está en idioma inglés, como rm que significa remove (borrar en español), mostrando que no solamente los comandos tienen significados sin sentido, por lo cual estos significados pueden ayudar cuando se redacten las instrucciones que se quieren obtener.

Esto demuestra lo importante que puede llegar a ser el sistema operativo Linux, dado que igual puede serle útil para otras personas que no estén acomodados con otro sistema operativo, siendo Linux el indicado para ellos, basándose en las necesidades que cada usuario necesite para ser adaptado y personalizado, y eso que muchas empresas se encargan de ocupar Linux para poder satisfacer sus necesidades y cumplir sus objetivos.

A comparación con Windows, Linux si ayuda en todo el mundo para fines empresariales y científicos, dado que en los trabajos se encargan de emplear Linux ya que permite trabajar en todo tipo de entornos y situaciones, mostrando la gran importancia que puede tener ese sistema operativo para el mundo, así como eficientes pueden ser sus comandos para eso mismo.

Comentarios

Ignacio Ramírez Bartolo

Al realizar la práctica, se pudo determinar cómo funcionan los comandos que nos puede proveer el sistema operativo Linux, así también se puede notar de que hay variantes de distribución Linux como Ubuntu o Fedora. De igual manera, se demostró que, a pesar de que Linux es un sistema operativo que no todo el mundo usa, si muestra que puede tener ventajas que la hacen diferente a otros sistemas operativos como Windows o MacOS, y eso se comprueba en los comandos que se pueden hacer en la terminal, algo que muchas personas ni saben que se puede hacer por la complejidad que se requiere para ponerlas a llevar a cabo, pero de igual manera si es algo que puede ayudar si se logra dominar.

Martínez Castañeda Nosha Alejandro

Encontramos que es importante utilizar los comando en la terminal para navegar por el sistema de Linux, así como lo hacemos en Windows en biblioteca de archivos, solo que, con comandos, lo que también facilita el manipular la terminal es que solo necesitamos el teclado sin necesidad de utilizar el mouse. Pudiendo así manipular todos nuestros archivos y carpetas desde una simple terminal.

Juan Manuel Miranda Rayón

Ubuntu no es como lo recordaba, se ha hecho es mucho más fácil e intuitivo, se me hizo similar la “terminal” a “cmd” que se ocupa en Windows donde se hacer exactamente esto que se realizó de crear un directorio moverlo, copiarlo, borrarlo, darle atributos y así. Ayuda a conocer nuevos comandos con sintaxis distintas a las habituales que enriquece nuestro conocimiento a la programación.

Arreola Santander Ricardo

La realización de esta práctica ayuda a conocer cómo funciona el sistema operativo de una computadora, en este caso principalmente con el sistema operativo de Linux, se aprenden sus áreas de desarrollo, su forma de trabajar, el cómo instalarlo y cómo utilizarlo. La práctica también ayuda a aprender los comandos básicos que se pueden utilizar para iniciar en el área de la programación, es una buena opción para iniciar con los lenguajes de programación.

En la práctica se explica paso por paso como utilizar el programa, cada punto se muestra explicado con imágenes que ayudan a guiar el desarrollo correcto de la práctica.

Sánchez Wilhelm Jeremías

El sistema operativo GNU/Linux es un conjunto de proyectos orientado al código abierto y multitarea, y es uno de los ejemplos más prominentes en cuanto al software libre. Todo esto se pudo observar al momento de realizar la práctica, en la cual se pudo apreciar la facilidad del uso del sistema operativo mediante el uso de una terminal, en la que, a base de varios comandos, se pueden realizar y editar varios aspectos del sistema, todo a base de texto y sin necesidad de una interfaz.

De esta manera es entonces que se puede ver que este sistema operativo es de gran utilidad por la facilidad que le da al usuario para modificar múltiples partes del sistema sin verse limitado, comparado con otros sistemas operativos mucho más limitados para las necesidades que puedan surgir para el usuario.

Referencias bibliográficas

Ó. Huguet Soriano and S. Doménech Gómez, "8.1. Introducción a Linux", *Mural.uv.es*, 2020. [Online]. Available: http://mural.uv.es/oshuso/81_introduccion_a_linux.html. [Accessed: 16- Oct- 2020].

P. Delgado, "Integración de sistemas. Linux y su sistema gestor de ficheros (descripciones).", *Todobytes.es*, 2020. [Online]. Available: <http://todobytes.es/2014/09/integracion-de-sistemas-linux-y-su-sistema-gestor-deficheros-descripciones/>. [Accessed: 16- Oct- 2020].