



UNIVERSIDAD NACIONAL AUTONOMA DE MÉXICO



FACULTAD DE INGENIERIA.

FUNDAMENTOS DE PROGRAMACIÓN.

Semestre:2022-2.

Profesor: Hugo Zuñiga Barragan.

Alumno: Vargas Palencia Oscar Adrian.

Practica: 2

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.

Actividades:

- ✓ Iniciar sesión en un sistema operativo GNU/Linux y abrir una “terminal”
- ✓ Utilizar los comandos básicos para navegar por el sistema de archivos.
- ✓ Emplear comandos para manejo de archivos.

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

En la actualidad existen diversos sistemas operativos; por ejemplo, para equipos de cómputo están Windows, Linux, Mac OS entre otros. Para el caso de dispositivos móviles se encuentran Android, IOS, Windows Phone, etcétera. Cada uno de ellos tiene diferentes versiones y distribuciones que se ajustan a los diversos equipos de cómputo y comunicación en los que trabajan.

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.

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

Procedimiento:

Comenzamos ingresando a la siguiente pagina la cual nos va a servir de mucha utilidad en esta práctica.

<https://bellard.org/jslinux/vm.html?url=alpine-x86.cfg&mem=192>

1-Comenzamos usando el comando ls para listar los elementos del sistema de archivos. de Linux.

```
Loading...

Welcome to JS/Linux (i586)

Use 'vlogin username' to connect to your account.
You can create a new account at https://vfsync.org/signup .
Use 'export_file filename' to export a file to your computer.
Imported files are written to the home directory.

localhost:~# ls
bench.py  hello.c  hello.js  readme.txt
localhost:~#
```

2-usamos el comando ls all para en listar todos los archivos del sistema y no uno en especifico como el caso anterior.

```
Loading...

Welcome to JS/Linux (i586)

Use 'vlogin username' to connect to your account.
You can create a new account at https://vfsync.org/signup .
Use 'export_file filename' to export a file to your computer.
Imported files are written to the home directory.

localhost:~# ls
bench.py  hello.c  hello.js  readme.txt
localhost:~# ls-all
sh: ls-all: not found
localhost:~# ls -all
total 40
drwxr-xr-x  5 root  root    237 Jan  9  2021 .
drwxrwxrwx 21 root  root    461 Feb 21 18:09 ..
-rw-----  1 root  root     18 Feb 21 18:09 .ash_history
drwx-----  3 root  root     61 Jul  5  2020 .cache
drwx-----  5 root  root    124 Jul  5  2020 .mozilla
drwxr-xr-x  4 root  root    202 Jul  6  2020 .wine
-rw-r--r--  1 root  root    114 Jul  5  2020 bench.py
-rw-r--r--  1 root  root     76 Jul  3  2020 hello.c
-rw-r--r--  1 root  root     22 Jun 26  2020 hello.js
-rw-r--r--  1 root  root    151 Jul  5  2020 readme.txt
localhost:~#
```

3- ahora usamos el comando pwd para saber en que carpeta nos encontramos.

```
Loading...
Welcome to JS/Linux (i586)

Use 'vlogin username' to connect to your account.
You can create a new account at https://vfsync.org/signup .
Use 'export_file filename' to export a file to your computer.
Imported files are written to the home directory.

localhost:~# ls
bench.py  hello.c  hello.js  readme.txt
localhost:~# ls -all
sh: ls-all: not found
localhost:~# ls -all
total 40
drwxr-xr-x  5 root    root    237 Jan  9  2021 .
drwxrwxrwx 21 root    root    461 Feb 21 18:09 ..
-rw-----  1 root    root     18 Feb 21 18:09 .ash_history
drwx-----  3 root    root     61 Jul  5  2020 .cache
drwx-----  5 root    root    124 Jul  5  2020 .mozilla
drwxr-xr-x  4 root    root    202 Jul  6  2020 .wine
-rw-r--r--  1 root    root    114 Jul  5  2020 bench.py
-rw-r--r--  1 root    root     76 Jul  3  2020 hello.c
-rw-r--r--  1 root    root     22 Jun 26  2020 hello.js
-rw-r--r--  1 root    root    151 Jul  5  2020 readme.txt
localhost:~# pdw
sh: pdw: not found
localhost:~# pwd
/root
localhost:~#
```

4-Posteriormente hacemos una búsqueda con un carácter y el comando ls -all "carácter" de esta forma nos va a mostrar los archivos que inicien con ese carácter en la carpeta root.

```
bench.py  hello.c  hello.js  readme.txt
localhost:~# ls -all
sh: ls-all: not found
localhost:~# ls -all
total 40
drwxr-xr-x  5 root    root    237 Jan  9  2021 .
drwxrwxrwx 21 root    root    461 Feb 21 18:09 ..
-rw-----  1 root    root     18 Feb 21 18:09 .ash_history
drwx-----  3 root    root     61 Jul  5  2020 .cache
drwx-----  5 root    root    124 Jul  5  2020 .mozilla
drwxr-xr-x  4 root    root    202 Jul  6  2020 .wine
-rw-r--r--  1 root    root    114 Jul  5  2020 bench.py
-rw-r--r--  1 root    root     76 Jul  3  2020 hello.c
-rw-r--r--  1 root    root     22 Jun 26  2020 hello.js
-rw-r--r--  1 root    root    151 Jul  5  2020 readme.txt
localhost:~# pdw
sh: pdw: not found
localhost:~# pwd
/root
localhost:~# ls h*
hello.c  hello.js
localhost:~# ls b*
sh: lsb*: not found
localhost:~# ls b*
bench.py
localhost:~# ls be*
bench.py
localhost:~# ls ba*
ls: ba*: No such file or directory
localhost:~#
```

5-Ahora usamos el comando `mkdir curso` para crear una carpeta en la carpeta `root` y comprobamos con el comando `ls -all`, donde podemos ver la hora cuando fue creado la carpeta.

```
localhost:~# mkdir curso
localhost:~# ls -all
total 44
drwxr-xr-x  6 root    root    259 Jan  9  2021 .
drwxrwxrwx 21 root    root    461 Feb 26 22:02 ..
-rw-----  1 root    root      20 Feb 26 22:02 .ash_history
drwx-----  3 root    root      61 Jul  5  2020 .cache
drwx-----  5 root    root    124 Jul  5  2020 .mozilla
drwxr-xr-x  4 root    root    202 Jul  6  2020 .wine
-rw-r--r--  1 root    root    114 Jul  5  2020 bench.py
drwxr-xr-x  2 root    root     37 Feb 26 22:02 curso
-rw-r--r--  1 root    root     76 Jul  3  2020 hello.c
-rw-r--r--  1 root    root     22 Jun 26  2020 hello.js
-rw-r--r--  1 root    root    151 Jul  5  2020 readme.txt
localhost:~#
```

6- Ahora usamos el comando `cd curso` para entrar en esta carpeta, si todo salió bien ingresamos a la carpeta y GNU nos va a mostrar que ya estamos en ella y usamos el comando `ls -all` para que nos muestre que hay adentro pero como la acabamos de encontrar no va a haber nada.

```
localhost:~# cd curso
localhost:~/curso# ls -all
total 8
drwxr-xr-x  2 root    root     37 Feb 26 22:02 .
drwxr-xr-x  6 root    root    259 Jan  9  2021 ..
localhost:~/curso#
```

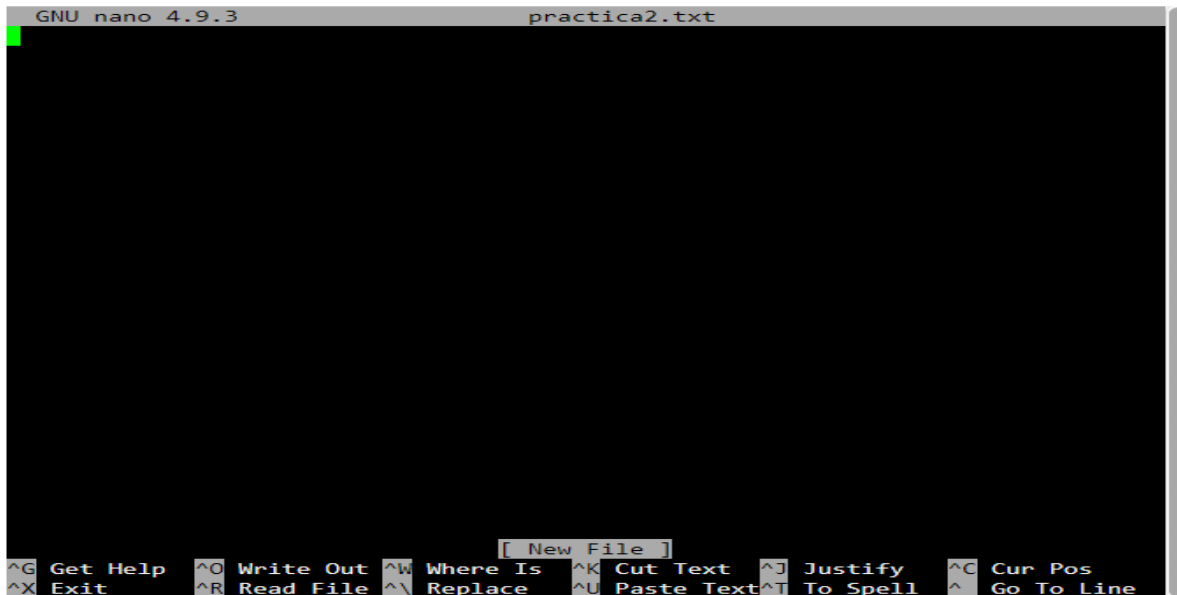
7-Para salir de esta carpeta usamos el comando `cd ..` y el sistema nos va a mostrar que nos encontramos en el sistema madre.

```
drwxr-xr-x  6 root    root    259 Jan  9  2021 ..
localhost:~/curso# cd ..
localhost:~#
```

8-Usamos el comando `pwd` para ver en que carpeta estamos.y usamos el comando `cd curso` para ingresar a la carpeta y posteriormente volvemos a usar el comando `pwd` para ver en que carpeta nos encontramos.

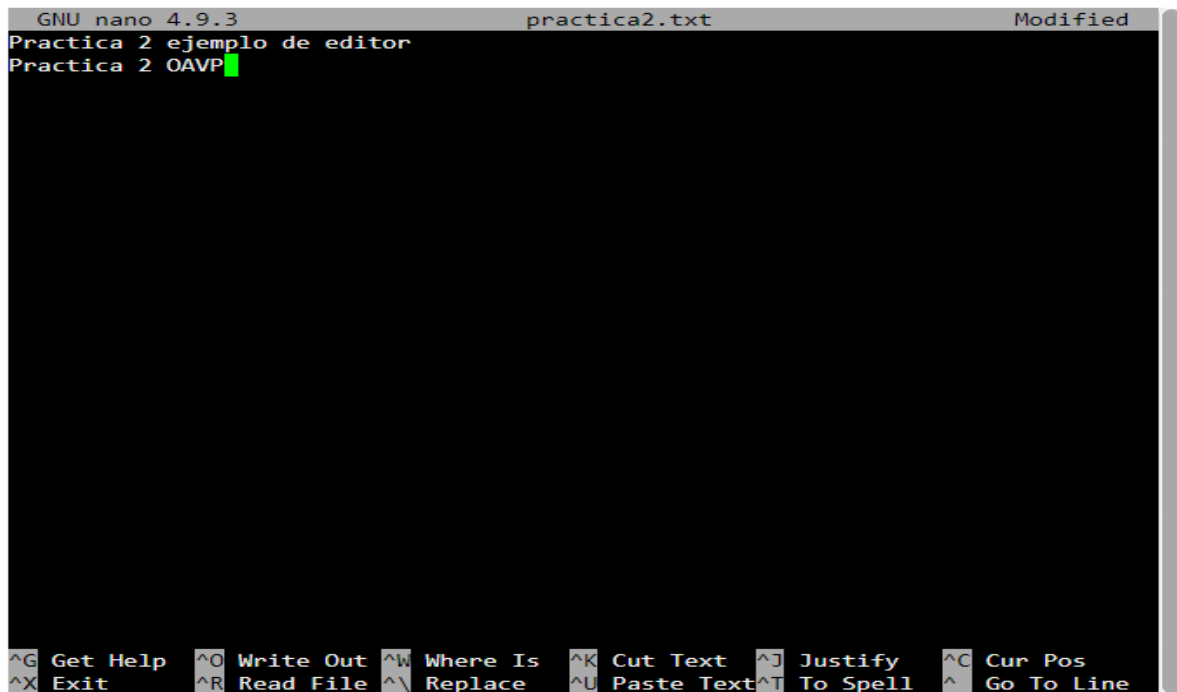
```
localhost:~/curso# cd ..
localhost:~# pwd
/root
localhost:~# cd curso
localhost:~/curso# pwd
/root/curso
localhost:~/curso#
```

9-Usamos el comando nano practica2.txt para escribir en la carpeta, ya dentro nos aparece las opciones que tenemos.



```
GNU nano 4.9.3 practica2.txt
[ New File ]
^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos
^X Exit      ^R Read File ^\ Replace  ^U Paste Text ^T To Spell  ^_ Go To Line
```

10- Ahora escribimos algo dentro del archivo y nos salimos del editor con control + x.



```
GNU nano 4.9.3 practica2.txt Modified
Practica 2 ejemplo de editor
Practica 2 OAVP
^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos
^X Exit      ^R Read File ^\ Replace  ^U Paste Text ^T To Spell  ^_ Go To Line
```

11-ahora usamos el comando ls -all dentro de la carpeta curso y dentro ya vamos a poder ver el archivo que creamos.

```
localhost:~/curso# ls -all
total 12
drwxr-xr-x  2 root    root        67 Feb 26 22:18 .
drwxr-xr-x  6 root    root       259 Jan  9  2021 ..
-rw-r--r--  1 root    root        45 Feb 26 22:22 practica2.txt
localhost:~/curso#
```

12-Ahora creamos otra carpeta con el comando mkdir tareas y nos dirigimos a ella con el comando cd y revisamos que nos encontremos en esta carpeta.

```
localhost:~/curso# mkdir tareas
localhost:~/curso# cd tareas
localhost:~/curso/tareas# pwd
/root/curso/tareas
localhost:~/curso/tareas#
```

13- Ahora para borrar una carpeta usamos el comando rmdir tareas y revisamos que se hay efectuado el cambio con el comando ls -all.

```
localhost:~/curso/tareas# cd ..
localhost:~/curso# rmdir tareas
localhost:~/curso# ls -all
total 12
drwxr-xr-x  2 root    root        67 Feb 26 22:18 .
drwxr-xr-x  6 root    root       259 Jan  9  2021 ..
-rw-r--r--  1 root    root        45 Feb 26 22:22 practica2.txt
localhost:~/curso#
```

14-Ahora regresamos hasta la carpeta madre con el comando `cd ..` y revisamos que están creadas las carpetas y procedemos a borrarlas, donde nos va a dar un error ya que la carpeta `curso` se encuentra vacía.

```
localhost:~/curso/tareas# cd ..
localhost:~/curso# rmdir tareas
localhost:~/curso# ls -all
total 12
drwxr-xr-x  2 root    root          67 Feb 26 22:18 .
drwxr-xr-x  6 root    root        259 Jan  9  2021 ..
-rw-r--r--  1 root    root          45 Feb 26 22:22 practica2.txt
localhost:~/curso# cd ..
localhost:~# ls -all
total 44
drwxr-xr-x  6 root    root        259 Jan  9  2021 .
drwxrwxrwx 21 root    root       461 Feb 26 22:31 ..
-rw-----  1 root    root       117 Feb 26 22:31 .ash_history
drwx-----  3 root    root        61 Jul  5  2020 .cache
drwx-----  5 root    root       124 Jul  5  2020 .mozilla
drwxr-xr-x  4 root    root       202 Jul  6  2020 .wine
-rw-r--r--  1 root    root       114 Jul  5  2020 bench.py
drwxr-xr-x  2 root    root         67 Feb 26 22:18 curso
-rw-r--r--  1 root    root         76 Jul  3  2020 hello.c
-rw-r--r--  1 root    root         22 Jun 26  2020 hello.js
-rw-r--r--  1 root    root       151 Jul  5  2020 readme.txt
localhost:~# rmdir curso
rmdir: 'curso': Directory not empty
localhost:~#
```

15- Ahora nos dirigimos a la carpeta `curso` y en esta usamos el comando `cp practica2.txt` seguida del nombre del archivo que va a ser la copia. donde si no marca un error todo salió bien. y checamos con el comando `ls -all` para saber que todo esta bien.

```
localhost:~# cd curso
localhost:~/curso# ls -all
total 12
drwxr-xr-x  2 root    root          67 Feb 26 22:18 .
drwxr-xr-x  6 root    root        259 Jan  9  2021 ..
-rw-r--r--  1 root    root          45 Feb 26 22:22 practica2.txt
localhost:~/curso# cp practica2.txt practica.txt
localhost:~/curso# ls -all
total 16
drwxr-xr-x  2 root    root          96 Feb 26 22:18 .
drwxr-xr-x  6 root    root       259 Jan  9  2021 ..
-rw-r--r--  1 root    root          45 Feb 26 22:34 practica.txt
-rw-r--r--  1 root    root          45 Feb 26 22:22 practica2.txt
localhost:~/curso#
```


16- Ahora con el comando moved(mv) cambiamos el archivo en caso de un error. Si no nos da un mensaje de error todo salió bien.

```
localhost:~/curso# mv practica.txt practica3.txt
localhost:~/curso# ls -all
total 16
drwxr-xr-x  2 root    root          97 Feb 26 22:18 .
drwxr-xr-x  6 root    root        259 Jan  9  2021 ..
-rw-r--r--  1 root    root         45 Feb 26 22:22 practica2.txt
-rw-r--r--  1 root    root         45 Feb 26 22:34 practica3.txt
localhost:~/curso#
```

17-Ahora con el comando rm borramos el archivo y procedemos a comprobar con el comando ls -all que todo salió bien.

```
localhost:~/curso# rm practica3.txt
localhost:~/curso# ls -all
total 12
drwxr-xr-x  2 root    root          67 Feb 26 22:18 .
drwxr-xr-x  6 root    root        259 Jan  9  2021 ..
-rw-r--r--  1 root    root         45 Feb 26 22:22 practica2.txt
localhost:~/curso#
```

18- en caso de querer limpiar la pantalla usamos el comando clear y la pantalla quedara sin ningún comando antes escrito.

```
localhost:~#
```

18- Usamos el comando `find . -name` para encontrar nuestro archivo deseado, y en que carpeta se encuentra.

```
localhost:~# find . -name practica2.txt
./curso/practica2.txt
localhost:~#
```

19-Si usamos el mismo comando pero ahora con este cambio `find .. -name` ya que `..` significa que lo busque en la carpeta madre, y este va a tardar mas que el anterior pero nos dará desde la carpeta madre donde se encuentra.

```
localhost:~# find . -name practica2.txt
./curso/practica2.txt
localhost:~# ^C
localhost:~# find .. -name practica2.txt
../root/curso/practica2.txt
localhost:~#
```

20-Ahora con el comando `ls` notamos que hay un archivo en lenguaje C.

```
localhost:~# find .. -name practica2.txt
../root/curso/practica2.txt
localhost:~# ls
bench.py  curso  hello.c  hello.js  readme.txt
localhost:~#
```

21- Ahora ingreemos con el comando `nano hello.c` y podemos ver que tiene un programa que se encarga de dar un mensaje

```
GNU nano 4.9.3 hello.c
#include <stdio.h>
int main(void)
{
    printf("hello world\n");
    return 0;
}

[ Read 7 lines ]
^G Get Help  ^O Write Out  ^W Where Is  ^K Cut Text  ^J Justify  ^C Cur Pos
^X Exit      ^R Read File  ^\ Replace  ^U Paste Text ^T To Spell  ^_ Go To Line
```

22- procedemos a salirnos y a hacer el comando clear y el comando ls -all para ver las carpetas que tenemos.

```
localhost:~# ls -all
total 44
drwxr-xr-x   6 root    root      259 Jan  9  2021 .
drwxrwxrwx  21 root    root      461 Feb 26 22:53 ..
-rw-----   1 root    root      359 Feb 26 22:53 .ash_history
drwx-----   3 root    root        61 Jul  5  2020 .cache
drwx-----   5 root    root      124 Jul  5  2020 .mozilla
drwxr-xr-x   4 root    root      202 Jul  6  2020 .wine
-rw-r--r--   1 root    root      114 Jul  5  2020 bench.py
drwxr-xr-x   2 root    root        67 Feb 26 22:18 curso
-rw-r--r--   1 root    root        76 Jul  3  2020 hello.c
-rw-r--r--   1 root    root        22 Jun 26  2020 hello.js
-rw-r--r--   1 root    root      151 Jul  5  2020 readme.txt
localhost:~#
```

23-Ahora para ejecutar un programa usamos el comando gcc hello.c -o hola, el compilador lo traducirá y nos dará el resultado.y lo comprobamos con ls -all y nos daremos cuenta que esta en verde el nombre del archivo porque este si se puede ejecutar.

```
localhost:~# gcc hello.c -o hola
localhost:~# ls -all
total 64
drwxr-xr-x   6 root    root      280 Jan  9  2021 .
drwxrwxrwx  21 root    root      461 Feb 26 22:56 ..
-rw-----   1 root    root      387 Feb 26 22:56 .ash_history
drwx-----   3 root    root        61 Jul  5  2020 .cache
drwx-----   5 root    root      124 Jul  5  2020 .mozilla
drwxr-xr-x   4 root    root      202 Jul  6  2020 .wine
-rw-r--r--   1 root    root      114 Jul  5  2020 bench.py
drwxr-xr-x   2 root    root        67 Feb 26 22:18 curso
-rw-r--r--   1 root    root        76 Jul  3  2020 hello.c
-rw-r--r--   1 root    root        22 Jun 26  2020 hello.js
-rwxr-xr-x   1 root    root    18284 Feb 26 22:56 hola
-rw-r--r--   1 root    root      151 Jul  5  2020 readme.txt
localhost:~#
```

24-Ahora para ejecutar el programa usamos el comando ./ seguido del nombre del archivo.

```
localhost:~# ./hola
hello world
localhost:~#
```

25-Si queremos ver el lenguaje de la maquina para el hello world usamos el comando nano seguido del nombre del programa.

[illegible]

26-procedemos a salirnos y limpiar la pantalla

```
localhost:~#
```

Conclusión.

En esta práctica aprendimos diversos comandos con ayuda de la interfaz online de GNU/Linux los cuales usamos para la gestión de carpetas y archivos en GNU, desde crearlos hasta moverlos, copiarlos y removerlos con ayuda de puros comandos, en lo personal esta práctica me saco un poco de mi zona de confort por el hecho de que esta interfaz no era muy grafica y es diferente a lo que estamos acostumbrados, en el proceso de la practica hubo varias dudas conmigo y con mis compañeros de clase donde solo por usar un carácter diferente causaba un error en la interfaz así que debíamos ser claros con la sintaxis y el orden de los caracteres ya que solo por poner un espacio de mas o de menos causaba un error y teníamos que corregir esto, en el caso del comando “nano” este fue complicado entenderlo ya que teníamos que usar combinaciones de teclas para guardar, salir entre otras acciones las cuales si no estábamos atentos a esta explicación para entenderlo correctamente, pero al final todo salió como lo esperábamos aprendiendo los comandos en GNU y la aplicación de estos en la gestión de archivos.