	Carátula para entrega de prácticas
Facultad de Ingeniería	Laboratorio de docencia

Laboratorios de computación Salas A y B

Profesor: Marco Antonio Martinez Quintana

Asignatura: Fundamentos de Programacion

Grupo: 3

No de Práctica(s): 2

Integrante(s): Lopez Martinez Diana

No. de Equipo de cómputo empleado:

No. de Lista o Brigada: 28

Semestre: 1

Fecha de entrega: 15/10/2020

Observaciones:

CALIFICACIÓN: _____

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.

Comandos básicos

Para trabajar en Linux utilizando comandos, se debe abrir una “terminal” o “consola” que 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.

Una vez teniendo una terminal abierta, estamos listos para introducir comandos.

La sintaxis que siguen los comandos es la siguiente:

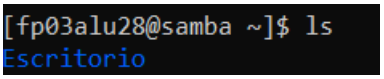
comando [-opciones] [argumentos]

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)

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

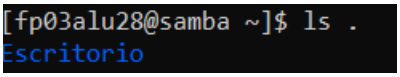
ls



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

```
[fp03alu28@samba ~]$ ls -1
Escritorio
```

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 a raíz, usamos:

ls /

```
[fp03alu28@samba ~]$ ls /
bin   database  etc   lib    lost+found  mnt  proc  run   sitio     srv  tmp   usr  webserver
boot  dev       home  lib64  media       opt  root  sbin  software  sys  users  var
```

Para ver los usuarios del equipo local, revisamos el directorio home que parte de la raíz (/):

ls /home

```
[fp03alu28@samba ~]$ ls /home
administrador  mena  MNR  squid
```

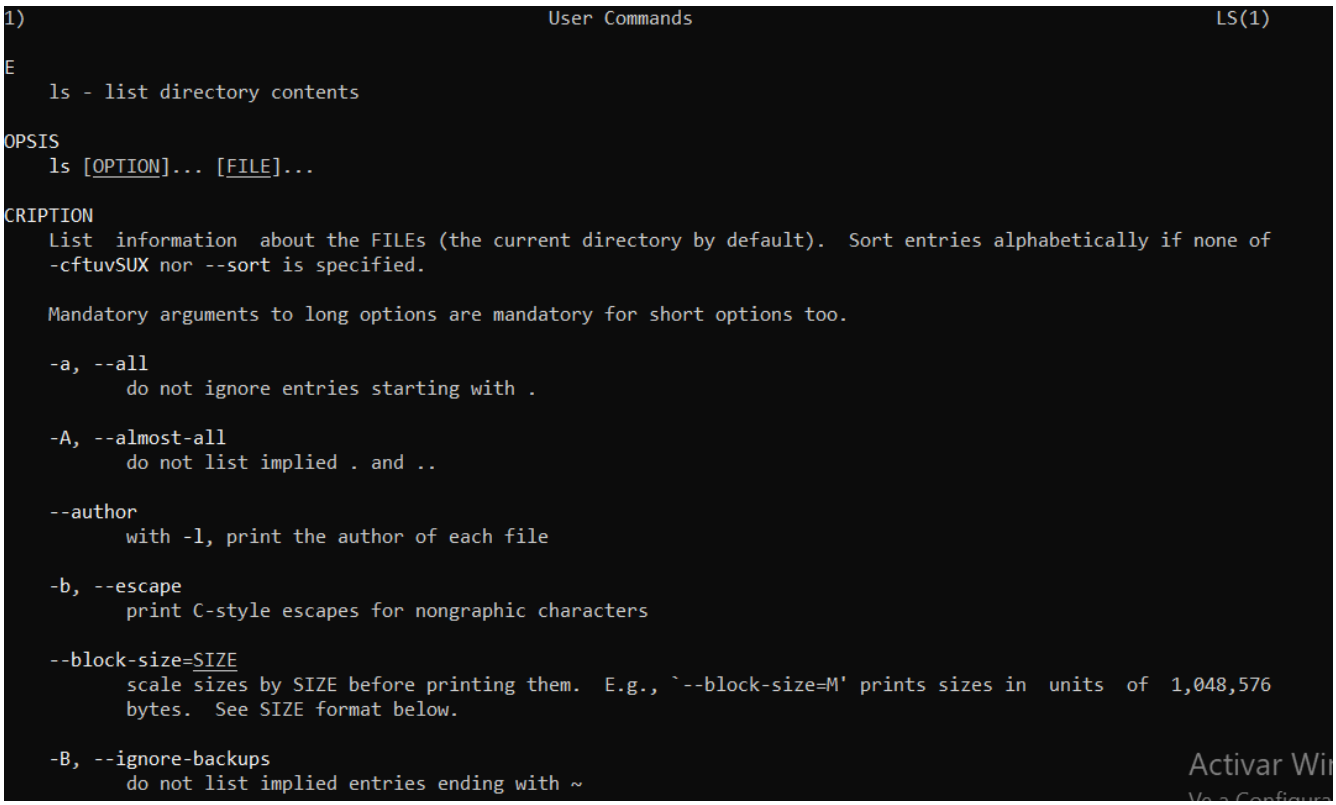
Tanto las opciones como los argumentos se pueden combinar para generar una ejecución más específica:

ls –l /home

```
[fp03alu28@samba ~]$ ls -1 /home
administrador
mena
MNR
squid
```

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 ..

```
[fp03alu28@samba ~]$ ls ..
fp03alu01  fp03alu06  fp03alu11  fp03alu16  fp03alu21  fp03alu26  fp03alu31  fp03alu36  fp03alu41  fp03alu46  fp03alu51  fp03alu56
fp03alu02  fp03alu07  fp03alu12  fp03alu17  fp03alu22  fp03alu27  fp03alu32  fp03alu37  fp03alu42  fp03alu47  fp03alu52  fp03alu57
fp03alu03  fp03alu08  fp03alu13  fp03alu18  fp03alu23  fp03alu28  fp03alu33  fp03alu38  fp03alu43  fp03alu48  fp03alu53
fp03alu04  fp03alu09  fp03alu14  fp03alu19  fp03alu24  fp03alu29  fp03alu34  fp03alu39  fp03alu44  fp03alu49  fp03alu54
fp03alu05  fp03alu10  fp03alu15  fp03alu20  fp03alu25  fp03alu30  fp03alu35  fp03alu40  fp03alu45  fp03alu50  fp03alu55
```

o

ls ../

```
[fp03alu28@samba ~]$ ls ../
fp03alu01  fp03alu06  fp03alu11  fp03alu16  fp03alu21  fp03alu26  fp03alu31  fp03alu36  fp03alu41  fp03alu46  fp03alu51  fp03alu56
fp03alu02  fp03alu07  fp03alu12  fp03alu17  fp03alu22  fp03alu27  fp03alu32  fp03alu37  fp03alu42  fp03alu47  fp03alu52  fp03alu57
fp03alu03  fp03alu08  fp03alu13  fp03alu18  fp03alu23  fp03alu28  fp03alu33  fp03alu38  fp03alu43  fp03alu48  fp03alu53
fp03alu04  fp03alu09  fp03alu14  fp03alu19  fp03alu24  fp03alu29  fp03alu34  fp03alu39  fp03alu44  fp03alu49  fp03alu54
fp03alu05  fp03alu10  fp03alu15  fp03alu20  fp03alu25  fp03alu30  fp03alu35  fp03alu40  fp03alu45  fp03alu50  fp03alu55
```

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

```
[fp03alu28@samba ~]$ 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]

```
[fp03alu28@samba ~]$ touch nombre_archivo[.txt]
```

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

```
[fp03alu28@samba ~]$ touch mkdir nueva_carpeta
```

Para crear una carpeta en nuestra cuenta, que tenga como nombre “tareas” se escribe el siguiente comando:

mkdir tareas

```
[fp03alu28@samba ~]$ 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

```
[fp03alu28@samba ~]$ cd tareas
```

Ahora, si deseamos situarnos en la carpeta de inicio de nuestra cuenta, que es la carpeta padre, escribimos el comando:

cd ..

```
[fp03alu28@samba tareas]$ cd ..
```

Ejemplo (comando pwd)

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

pwd

```
[fp03alu28@samba ~]$ pwd
/users/fp03/fp03alu28
```

Ejemplo (comando find)

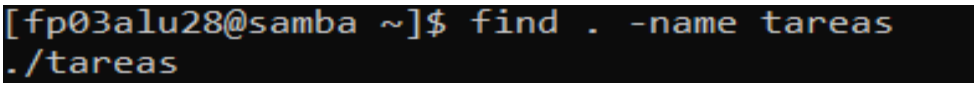
El comando find permite buscar un elemento dentro del sistema de archivos, su sintaxis es la siguiente:

find . –name cadena_buscar

Al comando find hay que indicarle 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)

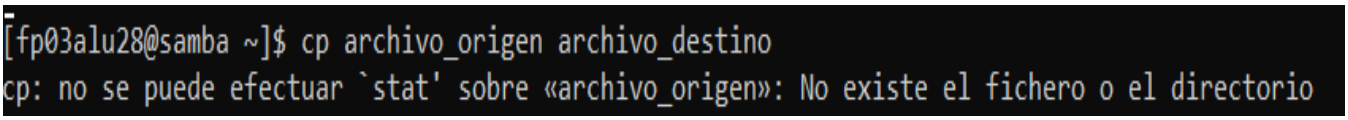
El comando clear permite limpiar la consola, su sintaxis es la siguiente:

clear

Ejemplo (comando cp)

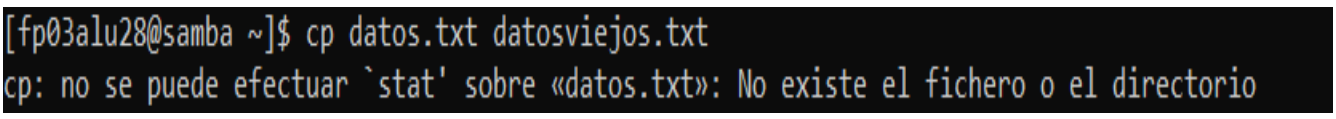
El comando cp permite copiar un archivo, su sintaxis es la siguiente:

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 ycon el mismo nombre, entonces podemos emplear las rutas relativas de la siguiente forma:

cp ../archivo_a_copiar .

```
[fp03alu28@samba ~]$ cp datos.txt datosviejos.txt
cp: no se puede efectuar `stat' sobre «datos.txt»: No existe el fichero o el directorio
```

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 que 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 .

```
[fp03alu28@samba ~]$ 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

```
[fp03alu28@samba ~]$ 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.

```
[fp03alu28@samba ~]$ rm nombre_archivo
rm: no se puede borrar «nombre_archivo»: No existe el fichero o el directorio
```

```
[fp03alu28@samba ~]$ rm nombre_carpeta
rm: no se puede borrar «nombre_carpeta»: Es un directorio
```

Conclusión

Es de crucial importancia saber manejar en su totalidad todos los comandos de lynux y de cualquier otro lenguaje de programación, pues de esa forma podemos obtener mayor control sobre las funciones del sistema operativo y así programar de la mejor calidadad.