1. ls: Lista el contenido del directorio, al usarse muestra los directorios y los archivos disponibles dentro de la carpeta en la que se esta trabajando, también puede usarse para listar los directorios de un directorio en especifico usando la dirección del directorio como referencia.

Ejemplo:

sintaxis: ls usr/bin

en este caso lo que haría es mostrar el contenido del directorio bin, estando en el directorio principal.

Si se emplea solo el comando “ls” mostrara el contenido del directorio actual.

1. cd: Permite cambiar de directorio, tanto volver a directorios anteriores como ir a unos en el siguiente nivel, como saltar de directorio en directorio directamente usando la ruta específica del directorio al que se quiere saltar.

Sintaxis: cd /tmp, cd..

Ejemplo:

En el primer caso lo que hace el comando es ir al directorio llamado tmp,

En el segundo caso lo que hace es retroceder de directorio por ejemplo estando en el directorio tmp, nos devolvería al directorio anterior.

1. pwd: Muestra la ruta del directorio actual, la sintaxis de este comando no tiene variaciones ya que solo muestra la ruta del directorio en el cual se esta trabajando.

Sintaxis: pwd

Ejemplo : pwd

1. mkdir: Crea un directorio, se puede usar para crear un directorio directamente en el directorio actual como para crear uno en una ruta especifica, se debe especificar un nombre y la ruta en la que se quiere crear el directorio.

Sintaxis: mkdir [Nombre del directorio] [Ruta]

ejemplo: estando en la carpeta escritorio se quiere crear un directorio dentro de un directorio existente, la carpeta se llamarà hola, dentro de la carpeta mundo, lo siguiente es encontrar la ruta, se veria algo asi:

mkdir hola /escritorio/mundo

1. rmdir: Borra un directorio vacío, puedes usarlo para remover directorios en el mismo directorio de trabajo como para borrar directorios que se encuentren en otra ubicación del sistema usando la ruta especifica de este.

/s y /q son parámetros extras que se pueden asignar al comando, “/S” quita todos los directorios y archivos contenido de el directorio especificado, además eliminar dicho directorio.

“/q” lo que hace es que no se va a pedir una confirmación a la hora de eliminar un árbol de directorios con “/s”,

Si se quiere eliminar contenido de una unidad de almacenamiento que no sea en la que se esta trabajando, se puede especificar la unidad para eliminar los datos, se agrega como parámetro antes de dar la ruta del directorio que se quiere eliminar.

Sintaxys: RMDIR /S /Q [<unidad>:]<ruta>

Ejemplo: quiero eliminar la carpeta de imágenes,

Rmdir /s /q <C:> users/imagenes

1. touch: El comando touch se usa principalmente para manipular el tiempo de acceso y modificación de los archivos mediante el uso de varios parámetros

Sintaxis: touch [opciones] [nombre\_archivo]

Las opciones son variadadas, por ejemplo:

Touch -a archivo.txt : modifica la ultima hora de acceso a la hora actual

Touch -m archivo.txt : modifica el tiempo de utima modificación del archivo

Touch – a -m archivo.txt : modifica la hora de acceso y de modificación del archivo a la actuar

1. cp: comando cp permite copiar archivos, vas al directorio del archivo que quieres copiar, e introduce el comando, el nombre del archivo y la ruta en la que quieres que se guarde el archivo nuevo.

Sintaxis : cp [Origen] [Destino]

Ejemplo: cp texto.txt /home/usuario/carpeta\_de\_destino/

En este ejemplo, creamos un nuevo archivo en la carpeta de destino que tiene el mismo nombre que el archivo de origen. Si quieres **copiar varios archivos al mismo tiempo**, escríbelos uno detrás de otro dentro de la misma línea de comando.

Ejemplo: cp texto.txt texto2.txt texto3.txt /home/usuario/carpeta\_de\_destino/

1. mv: El comando linux mv se utiliza para mover archivos y directorios de una ubicación a otra, también es utilizado para renombrar tanto archivos como directorios y la forma de usar este comando es muy parecida al uso del [comando linux cp](http://www.comoinstalarlinux.com/comandos-linux-cp/).

Sintaxis para mover un archivo:

mv archivo\_fuente directorio\_destino

Sintaxis para renombrar un archivo:

mv archivo\_de\_origen archivo\_de\_objetivo

1. rm: el comando rm permite borrar archivos y directorios

sintaxis: rm nombre\_archivo

ejemplo: rm holamundo

este comando elimina un archivo llamado hola-mundo, dentro del mismo directorio en el que se esta trabajando, también permite especificar una ruta para elminar un directorio de estando fuera del directorio de trabajo.

1. nano o vim: el comando vim tiene tres modos de trabajo, el modo comando:

El ‘ Modo Comando’ permite al usuario navegar por el documento así como introducir comandos a ejecutar dentro del propio archivo (buscar, reemplazar, guardar…), ya que en este modo el programa no interpreta las teclas del teclado por los caracteres que representan, si no por las funciones preestablecidas o comandos asignados a cada tecla.

En el ‘ Modo ex’ manipularemos los archivos. Para entrar a este modo deberemos escribir “ : ” seguido directamente por el nombre del comando ‘ex’ que queramos usar. Tras esto, vi volverá automáticamente al modo comando.

En el Modo inserciónsimplemente añadiremos texto al fichero. Mientras estemos en este modo, podremos salir al modo comando directamente pulsando ‘Esc’.

Este comando tiene muchos parámetros agregables a la ejecución del mismo, a continuación dejo un link con mayor información.

<https://openwebinars.net/blog/vim-manual-de-uso-basico/>

1. cat: permite concatenar y mostrar el contenido de archivos. Deriva de “concatenar” y se utiliza para visualizar, unir y crear archivos.

Sintaxis: cat parámetros nombre\_archivo

Por ejemplo, “cat ejemplo.txt” muestra el contenido de “ejemplo.txt”.

1. head: El comando head, es un comando utilizado para mostrar las primeras líneas de cualquier fichero.

Sintaxis: head /ruta\_del\_fichero/nombre\_del\_fichero

Ejemplo:head /var/log/syslog

Nos muestra las primeras 10 lineas del archivo syslog.

1. tail: el comando tail permite ver el contenido de las ultimas 10 lineas de un archivo, también se puede ocupar para saber si un archivo esta dañado, ya que los datos se escriben hacia abajo, nos permite comprobar si se han añadido nuevos dartos a un archivo.

Sintaxis: tail [parámetros] <archivos>

Se puede usar el comando tail sin la necesidad de agregar parámetros, ya también el apartado de parámetros permite elegir una ruta de fichero para leer.

1. grep: se trata de un comando de búsqueda global para las líneas que encajen con la expresión regular e imprimirlas (Global Regular Expresion Print en sus siglas en ingles), se los comandos grep se usan para buscar patrones y cadenas , esto permite filtrar información incluso en archivos de registro de gran tamaño

sintaxis: grep [Opciones] “Cadena de búsqueda” [ARCHIVO]

ejemplo: si queremos buscar el termino “test” en un archivo llamado ejemplo.txt, entonces la sintaxis se vería algo asi:” grep test ejemplo.txt ”

1. find: podemos realizar búsquedas, utilizando parámetros, en una amplia variedad de condiciones, ya sea encontrar ficheros modificados por fecha, o bien por tamaño, permisos, usuarios, grupos, o tipos de archivo.

Sintaxis: find /ruta/nombre\_archivo

Ejemplo: buscar por nombre en directorio en el que se esta trabajando:

Find / -name hola.txt

Ejemplo de buscar por ruta, directorio de reinaldo dentro del directorio user:

Find /user/reinaldo -name hola.txt

existen muchas otras formas de usar el comando, agrego link para info mas completa:

<https://www.ochobitshacenunbyte.com/2019/09/26/17-ejemplos-practicos-del-comando-find-en-gnu-linux/>

1. chmod: permite cambiar los permisos de acceso de un archivo o de un directorio.

Sintaxis: chmod [opción] modo archivo

Este comando tiene muchos parámetros manipulables por lo que dejare un link donde se profundiza mas en el tema, el contenido es algo extenso pasando por propiedad del archivo, permisos de escritura, lectura, ejecución etc… [https://www.infomaniak.com/es/asistencia/faq/111/cambiar-los-derechospermisos-de-un-archivo#:~:text=El%20CHMOD%20(abreviatura%20de%20cambio,archivo%20o%20de%20un%20directorio.](https://www.infomaniak.com/es/asistencia/faq/111/cambiar-los-derechospermisos-de-un-archivo%23:~:text=El%20CHMOD%20(abreviatura%20de%20cambio,archivo%20o%20de%20un%20directorio.)

1. chown: “change owner” por sus siglas en ingles, permite cambiar el propietario de un archivo o directorio en sistemas tipo UNIX, puede especificarse tanto el nombre del usuario, así como el identificador de usuarios (UID) y el identificador de grupo (GID).

Antes de usar el comando chown, tendríamos que confirmar la información del usuario y del grupo. Puedes obtener esta información mediante el uso de **cd** y navegando al directorio requerido.

Sintaxis: chown user filename (s)

Ejemplp: chown reinaldo ejemplodechonw.txt

(se puede especificar mas de un archivo a la vez)

1. ln: Con el comando ln de Linux (ln es la abreviatura de “link”, enlace en inglés), se consigue crear enlaces a un archivo o carpeta. Para ello, utilizamos la línea de comandos **de** Linux, que se abre y maneja mediante una ventana del terminal. ln es uno de los [comandos en Linux](https://www.ionos.es/digitalguide/servidores/configuracion/comandos-de-linux-la-lista-fundamental/) más importantes y utilizados. Con este comando de la lista de comandos, es posible generar enlaces simbólicos o duros.

Sintaxis: ln /ruta\_del\_archivo /ruta del destino en el que se quiere guardar el link

Ejemplo ln /home/reinaldo/videos/ejemplo.mp4

1. du: útil para buscar archivos y directorios que ocupan grandes cantidades de espacio en disco.

Sintaxis: du parámetros /directorio de referencia

Ejemplo: du ̴ / documentos

También se puede usar para fichar varios directorios o archivos de la misma manera

Ejemplo: du ̴ / documentos ̴/videos

1. df: da información sobre la utilzacion del disco duro en los diferentes sistemas de archivos montados en el sistema, para la mayoría de los sistemas basados en GNU/Linux, quedarse sin espacio es algo grave, ya que muchos programas en general utilizan el directorio /tmp para guardar información mientras se ejecutan.

Sintaxis: df /parámetros /sistema de archivos

Si no se da información acerca del sistema de archivos entonces dará como resultado información acerca de todos los sistemas de archivos montados y en funcionamiento dentro del sistema.

21. kill:

22. grep:

23. tar:

estudiar bien los daemons de linux