COMANDOS

DE LINUX

- **pwd (de *print working directory* o imprimir directorio de trabajo)**, es un comando que imprime nuestra ruta o ubicación al momento de ejecutarlo, así evitamos perdernos si estamos trabajando con múltiples directorios y carpetas.
  Su sintaxis sería:

  $ pwd

- **ls (de listar)**, permite listar el contenido de un directorio o fichero**.**
  La sintaxis es:

  $ ls directorio

  - Si omites "directorio" mostrará el contenido del directorio actual.
  - Puedes usar "***-a***" para mostrar los archivos ocultos
  - Puedes usar "***-l***" para mostrar los usuarios, permisos y la fecha de los archivos

  Así como para todos los comandos Linux, estas opciones pueden combinarse, terminando en algo como:

  $ ls -la directorio

- **cd (de *change directory* o cambiar directorio)**, es como su nombre indica el comando que necesitarás para acceder a una ruta distinta a la que te encuentras.

  Por ejemplo, si estas en el directorio "/home" y deseas acceder a "/home/libros", seria:

  $ cd libros

  Si estás en "/home/libros" y deseas subir un nivel (es decir ir al directorio padre, "/home"), ejecutarías:

  $ cd ..

- **touch crea un archivo vacío**, si el archivo existe actualiza la hora de modificación.

Para crear el archivo “prueba1.txt” seria:

$ touch prueba1.txt

- **mkdir (de *make directory* o crear directorio)**, crea un directorio nuevo tomando en cuenta la ubicación actual.

  Por ejemplo, si estas en “/home/profesor” y deseas crear el directorio “ejercicios”, sería:

  $ mkdir ejercicios

  Puedes usar la opción “-p” que permite crear un árbol de directorios completo que no existe

  $ mkdir -p ejercicios/uno/dos/tres

- **cp (de *copy* o copiar)**, copia un archivo o directorio origen a un archivo o directorio destino.

  Por ejemplo, para copiar el archivo “prueba.txt” ubicado en “/home/profesor” a un directorio de respaldo (que ha de existir previamente), podemos usar:

  $ cp prueba.txt /home/respaldo/prueba.txt

  En la sintaxis **siempre se especifica primero el origen y luego el destino**. Si indicamos un nombre de destino diferente, **cp** copiará el archivo o directorio con el nuevo nombre.

  • Con la opción “**-r** ” se copia no sólo el directorio especificado sino todos sus directorios internos de forma *recursiva.*

  Suponiendo que deseamos hacer una copia del directorio “/home/profesor/ejercicios” que a su vez tiene las carpetas “ejercicio1” y “ejercicio2” en su interior, en lugar de ejecutar un comando para cada carpeta, ejecutamos:

  $ cp -r /home/profesor/ejercicios /home/respaldos/

- **mv (de *move* o mover)**, mueve un archivo a una ruta específica, y a diferencia de ***cp***, lo elimina del origen finalizada la operación. Por ejemplo:

$ mv prueba.txt /home/respaldos/prueba2.txt

Al igual que ***cp*,** en la sintaxis se especifica primero el origen y luego el destino. Si indicamos un nombre de destino diferente, **mv** moverá el archivo o directorio con el nuevo nombre.

- **rm (de *remove* o remover),** es el comando necesario para borrar un archivo o directorio. Para borrar el archivo "prueba.txt" ubicado en "/home/profesor", ejecutamos:

$ rm /home/profesor/prueba.txt

Este comando también presenta varias opciones:

- La opción " ***-r*** " borra todos los archivos y directorios de forma recursiva.
- Por otra parte, "***-f*** " borra todo sin pedir confirmación.

Estas opciones pueden combinarse causando un borrado recursivo y sin confirmación del directorio que se especifique. Para realizar esto en el directorio "respaldos" ubicado en el "/home", usaríamos:

$ rm -rf /home/respaldos

Este comando es muy peligroso, por lo tanto, es importante que nos documentemos bien acerca de los efectos de estas opciones en nuestro sistema para así evitar consecuencias nefastas.

- **rmdir (de *remove directory* o remover directorio)**, elimina un directorio siempre que éste esté vacío (en caso contrario dará un error). Para eliminar "/home/respaldos" usaríamos:

$ rmdir /home/respaldos

- **chown (de *change owner* o cambiar propietario)**, permite ceder la propiedad a otro usuario y/o asignárselo a un grupo al que pertenezcamos. Para ceder la propiedad del archivo "prueba.txt" al usuario "alumno" escribiríamos:

$ chown alumno prueba.txt

- La opción "*-R*" aplica el cambio a todos los archivos y directorios de forma recursiva.

  Para ceder la propiedad al usuario "alumno" y al grupo "practicas" del directorio "/home/ejercicios" escribiríamos:

$ chown -R alumno:practicas /home/ejercicios

***NOTA**: También existe el comando **chgrp** para cambiar de grupo a un documento o carpeta*

- **chmod (de *change mode* o cambiar modo)** permite cambiar los permisos de un archivo o directorio. Este comando en concreto tiene varias sintaxis permitidas de la que explicaremos la más sencilla: **chmod [opciones] modo[,modo] fichero**

  Para ello tenemos que tener claros los distintos grupos de usuarios:

  - **u**: usuario dueño del fichero
  - **g**: grupo de usuarios asociado al fichero
  - **o**: todos los otros usuarios
  - **a**: todos los tipos de usuario (dueño, grupo y otros)

  También hay que saber la letra que abrevia cada tipo de permiso:
  - **r**: permiso de lectura
  - **w**: permiso de escritura
  - **x:** permiso de ejecución (exploración en directorios)

  Asigna permisos de lectura, escritura y ejecución para los usuarios "otros" a todos los archivos de la carpeta:

$ chmod o=rwx *

Asigna todos los permisos a todos los usuarios para el archivo “fichero.txt”:

$ chmod a=rwx fichero.txt

Quita todos los permisos para los usuario del grupo y los usuarios otros.

$ chmod go= *

Da todos los permisos al dueño del fichero, a los del grupo del dueño le asigna permisos de lectura y escritura y a los otros usuarios les quita todos los permisos. La opción “*-R*” aplica el cambio a todos los archivos y directorios de forma recursiva.

$ chmod -R u=rwx,g=rw,o= /home/ejercicios

NOTA. Un espacio después de la coma “ , “ en los distintos nodos de permisos que se indiquen hace fallar el comando

Da permisos únicamente de lectura a todos los tipos de usuario.

$ **chmod a=r ***

De un modo parecido a lo que acabamos de ver, también se pueden añadir o quitar permisos con los operadores **+** y **-**. Para ello se indica el tipo de usuario y el permiso que se resta o añade. Algo como esto:

Esto quita todos los permisos a todos los tipos de usuario.

$ **chmod a-wrx ***

Este comando asigna permisos de lectura a todos los usuarios y permisos de escritura al dueño del archivo y el grupo del dueño.

$ chmod a+r,gu+w *

- **du (de *disk usage* o uso de disco)**, estima el espacio ocupado por el directorio y todos sus subdirectorios.

- Con el parámetro "**-h**" para facilitar la lectura de las unidades de medida (Gigas, Mengas, Kb).

  Para analizar la ocupación de los contenidos almacenados en "/home/ejercicios" usaríamos:

  $ du -h /home/ejercicios

  Si sólo nos interesa conocer el resumen de todo el directorio añadiríamos el parámetro "-s":

  $ du -hs /home/ejercicios

- **df (de *disk filesystem* o sistema de archivos)** muestra el espacio de disco utilizado en cada volumen o partición montado.

  - La opción " **-T** " muestra el sistema de archivo también.

  - Con " **-h** " para utilizar múltiplos en las unidades de medida (Gigas, megas, etc):

    $ df -Th

- **find (de encontrar)**. Busca el archivo o carpeta que le indiques:

  $ find /home -name prueba.txt

  El comando anterior buscaría en todos los sitios las carpetas y archivos que se llamen "prueba.txt" a partir de la carpeta "/home"

  Si no estamos muy seguros del nombre podemos indicárselo con comodines.

  Supongamos que el nombre de lo que buscamos contiene "ejercicio", en la misma carpeta de antes:

$ find /home -name *ejercicio*

Tiene otras opciones avanzadas que permiten discriminar por tipo de archivo, fechas de creación/acceso/modificación, tamaño, propietario y/o permisos, entre otros.

- **clear (de limpiar)**, es un sencillo comando que limpiara nuestra terminal por completo dejándola como recién abierta.
  Para ello ejecutamos:

  $ **clear**

- ***man* muestra una documentación completa del comando proporcionado.** Para *clear*, por ejemplo:

  $ **man clear**

Página para practicar comandos
https://bellard.org/jslinux/vm.html?cpu=riscv64&url=buildroot-riscv64.cfg&mem=256