Comandos Básicos 2

cp

cp -> Nos permite copiar archivos.

- Ejemplo: /etc/hosts home/user || ORGEN -> DESTINO

Podemos dar una ruta al DESTINO, ejemplo: cp /etc/hosts /tmp/pepe2

En caso de que la carpeta del DESTINO no exista, lo copiará con ese nombre en el mismo lugar donde está el ORGEN.

Opciones del comando cp

cp -r -> Nos permite copiar el directorio de manera recursiva, es decir, que copie todo el contenido (directorio, subdirectorios, archivos, etc).

cp -a -> Tiene integrado la función recursiva, y tiene la función "-p" le permite mantener los permisos. Complemente un montón de opciones.

- Definición de Linux -> Copia todos los archivos, pero preservando como sea posible la mayor cantidad de estructuras y atributos de los archivos originales en la copia. Prácticamente igual que el -r.

mv

mv -> Nos permite mover archivos/carpetas o renombrarlos.

- Ejemplo renombrar: midir directorio|| ORGEN -> DESTINO (Sin indicar un path)
- Ejemplo mover: directorio home/user|| ORGEN -> DESTINO (Indicando un path)

mkdir

mkdir -> Nos permite crear directorios. Ejemplo: mkdir directorio

Opciones del comando mkdir

mkdir -p -> Nos permite crear directorios emparentados con un directorio "'padre", en caso de de que no existe el "padre" lo crea, creando padre-hijo

- Ejemplo: mkdir –p directorio/subdirectorio || En caso de no existir el "directorio", lo crea y así permite la creación del subdirectorio. Caso que el mkdir solo no permite al no existir el "padre" del subdirectorio
- Ejemplo anidamiento de Directorios: mkdir -p directorio/{A1,B1,C2} || En caso que no exista algún directorio/subdirectorio lo crea, y permite anidar en "directorio" los subdirectorios A1,B2,etc al "mismo nivel" dentro del directorio
- También puedo volver a anidar dentro de este directorio ya creado, ejemplo: mkdir -p directorio/{A1}/{A1.1,A1.2,A1.3}.
- También podemos hacer muchas combinaciones más.

tree

tree -> Nos permite listar directorios en forma de “árbol genealógico”

rm y rmdir

rm -> Nos permite borrar contenidos, puede dar error en caso que sea un directorio. NO SE PUEDE RECUPERAR

- Ejemplo: rm archivo.txt

rmdir -> Nos permite borrar directorios, utilizando paths completos. NO SE PUEDE RECUPERAR.

- Ejemplo: rmdir Juan/Lucia -> Indicar todo el path, sino dará error.

Opciones del comando rmdir y rm

rmdir -r -> Nos permite borrar de manera recursiva un directorio, no hace falta indicar todo el path, solo el comienzo.

rm -f -> Nos permite borrar de manera forzada (a toda costa) cualquier tipo de archivo, esto anula el “error” que da al intentar borrar un directorio.

touch

touch -> Permite crear un archivo de 0 bytes, si ya existe nos permite cambiar la fecha y hora (timestamp).

Opciones del comando touch

touch --date=20210531 (Año, mes, día) -> Cambia la fecha a una especifica.

echo

echo -> Muestra un mensaje en pantalla – Permite almacenar un mensaje en un archivo.

- Ejemplo: echo “Hola mundo” -> Imprime en consola el mensaje “Hola mundo”
- Ejemplo: echo “Hola” && echo “Mundo” -> Imprime en diferentes líneas el mensaje.
- Ejemplo: ech “Hola” ;; echo “Mundo” -> En caso de haber error en cualquier de los comandos, no le importe e imprime el siguiente. En este caso el primer echo va a tirar error por estar mal escrito, pero se va a ejecutar el 2do echo, cosa que con los && no pasa.
- Ejemplo: echo “Listado de Archivos” && ls

Opciones del comando echo

echo -n -> Elimina el salto de línea que produce el echo.

- Ejemplo: Echo -n “Ingrese su nombre: ” && read NOMBRE

Operadores > >> <

> >> <: A esto se le llama redireccionamiento, vamos a poder mandar contenido a archivos para que se sobrescriba o agregue al final, como también tomarlo como entrada.

- Ejemplo: echo “Hola” > archivo.txt || Escribe en el archivo txt el contenido de echo. En caso de haber contenido previo LO BORRA y lo reemplaza.
- Ejemplo: echo ls > archivo.txt || Escribe en el archivo txt el contenido de ls. En caso de haber contenido previo LO BORRA y lo reemplaza.
- Ejemplo: echo “Hola mundo” >> archivo.txt || Escribe en el archivo txt el contenido de echo SIN BORAR Y REEMPLAZAR el contenido previo.
- Ejemplo: echo “Esto es mi listado” > archivo.txt && ls >> archivo.txt || Combinación de ambos
- Ejemplo: cat < archivo.txt || El < permite la entrada del contenido de archivos, directorios, etc. En Bash tiene más sentido.

which

which X-> Permite buscar donde se encuentra almacenado el contenido de comandos. Dependiendo el orden que le da el PATH, ya que buscará el primero que encuentre.

- Ejemplo: which ls -> /bin/ls

nl

nl -> Nos muestra el número de línea de un archivo

cut

cut -> Con este comando nos permite el cortar el contenido de un archivo con un separador.

Ejemplo: cut -d ‘:’ -f1 /etc/passwd || -d es el delimitador, el ‘:’ indica donde tiene que cortar, y el -f1 le indicamos la columna que deseamos cortar y mostrar, luego le indicamos el path.

Ejemplo: cut -d ‘:’ -f1-3 /etc/passwd || El -f1-3 indica un rango.

Ejemplo: cut -d ‘:’ -f1-3,6 /etc/passwd || El -f1-3,6 el guion indica un rango, las comas indican columnas especificas

man

man -> Nos trae el manual del comando

Opciones del comando man

man -k X-> Nos trae todas las opciones posibles que existen para los manuales con el nombre X. Para buscar con un numero especifico hacemos un **man (NUMERO) X**

whatis

whatis X -> Nos trae todas las opciones posibles que existen para los manuales con el nombre X

apropos

apropos X -> Nos permite buscar con una palabra aproximada el manual que podemos estar buscando, nos sirve por si nos olvidamos el nombre.

info

info X -> La diferencia con man es que podemos navegar el manual, ingresando a vínculos que tiene el mismo manual, cosa que el man no permite.

Anotaciones

Si no tengo una aplicación guardada dentro de nuestro $PATH, puedo agregar mi ruta del directorio actual (Clase 5) o, para ejecutarlo, agregarlo un "./NOMBRE"

read: Crea una variable y permite almacenar algo en ella, ejemplo:

- Echo -n "Ingrese su nombre: " && read NOMBRE