



COMANDOS EN LINUX

TODO LO NECESARIO PARA USAR LINUX

KIKE GANDIA
CONTANDO BITS

1. Introducción

En este documento tienes recogido una gran cantidad de comandos de Linux que te permitirán manejarte con un alto nivel de soltura dentro de los sistemas operativos Linux/GNU. Una vez adquieras este documento, podrás volverlo a descargar gratis ya que con el tiempo se irá ampliando incluyendo comandos mucho más técnicos y específicos que iremos viendo en el canal.

2. Comandos para navegar entre directorios

En primer lugar, veremos los comandos básicos para movernos por los directorios de Linux:

- **pwd**: para encontrar la ruta del directorio (carpeta) donde te encuentres en ese momento. El comando devolverá una ruta absoluta, como por ejemplo `/home/username`.
 - Sintaxis: `"pwd"`
- **cd**: comando usado para entrar o salir de un directorio.
 - Sintaxis: `"cd nombre_directorio"`
 - Variaciones:
 - `"cd .."`: para volver al directorio anterior (carpeta padre).
 - `"cd -"`: para volver al directorio dónde estábamos anteriormente.

NOTA: Linux distingue entre mayúsculas y minúsculas por lo que el nombre del directorio debe escribirse de forma exacta.

- **ls**: este comando se utiliza para listar los archivos contenidos en un directorio, es decir, ver que archivos se encuentran dentro de una carpeta.
 - Sintaxis: `"ls -opciones"`
 - Variaciones más útiles:
 - `"ls -a"`: muestra todos los archivos del directorio en el que nos encontremos, incluyendo los que estén ocultos (comienzan por un `."`).
 - `"ls -t"`: ordena los archivos listados por fecha de modificación.
 - `"ls -X"`: ordena los archivos listados por extensión.
 - `"ls -l"`: muestra toda la información de los archivos (usuario, permisos, tamaño, fecha y hora de creación...)
 - `"ls -R"`: muestra el contenido de los subdirectorios de forma recursiva.
 - `"ls -S"`: ordena los archivos listados por tamaño de archivo.
 - `"ls -1"`: muestra el listado de archivos en una única columna.
 - `"ls --color"`: muestra los archivos por un color distinto en función al tipo de archivos.

3. Comandos para gestionar archivos

A continuación, veremos los comandos para gestionar los archivos en Linux:

- **mkdir**: se utiliza para crear nuevas carpetas, es la abreviación de *"make directory"*.
 - Sintaxis: *"mkdir ruta/nombre_archivo"*
 - Variaciones más útiles:
 - *"mkdir -p archivo/archivo_nuevo/archivo"*: opción para crear directorios entre dos directorios ya existentes
- **rmdir**: abreviación de *"remove directory"*, sirve para borrar directorios vacíos, si el directorio contiene algún archivo se utilizará el comando *"rm -r"* (ver más abajo).
- **touch**: comando para la creación de archivos en blanco de la extensión que deseemos.
 - Sintaxis: *"touch ruta/nombre_archivo.extension"*
- **cat**: es la abreviatura de concatenate y se utiliza para listar el contenido de un archivo en la salida estándar del sistema, ósea en la terminal.
 - Sintaxis: *"cat archivo"*
 - Variaciones más útiles:
 - *"cat > nombredearchivo"* crea un nuevo archivo.
 - *"cat nombredearchivo1 nombredearchivo2>nombredearchivo3"* une dos archivos (1 y 2) y almacena la salida de ellos en un nuevo archivo (3)
- **cp**: se utiliza para copiar archivos del directorio actual al directorio diferente especificado.
 - Sintaxis: *"cp archivo ruta"*
- **mv**: se utiliza para mover archivos de un directorio a otro o para cambiarles el nombre.
 - Sintaxis: *"mv archivo ruta"*
 - Variaciones cambio de nombre:
 - *"mv nombre_antiguo nombre_nuevo"*
- **rm**: se utiliza para eliminar cualquier archivo.
 - Sintaxis: *"rm archivo"*
 - Variaciones más útiles:
 - *"rm -r"*: para eliminar directorios junto a su contenido. Una vez eliminados los archivos con estos comandos, no se podrán recuperar así que hay que usarlo con cuidado.
- **wc**: devuelve el número de líneas, palabras o caracteres de un archivo.
 - Sintaxis: *"wc archivo1.ext"*
 - Variaciones:

- “wc -w *archivo1.ext*”: muestra el número de palabras.
 - “wc -m *archivo1.ext*”: muestra el número de caracteres.
 - “wc -l *archivo1.ext*”: muestra el número de líneas.
- **tail**: para ver las últimas líneas de cualquier archivo de texto. El número de líneas por defecto son 10.
 - Sintaxis: “tail *archivo1.ext* *archivo2.ext*”
 - Variaciones:
 - “tail -n *archivo*”: muestra el número de líneas que se estipulen en el parámetro n.
- **head**: para ver las primeras líneas de cualquier archivo de texto. El número de líneas por defecto son 10.
 - Sintaxis: “head *archivo1.ext* *archivo2.ext*”
 - Variaciones:
 - “head -n *archivo*”: muestra el número de líneas que se estipulen en el parámetro n.
- **diff**: compara el contenido de dos archivos línea a línea. Después genera las líneas que no coinciden. Muy útil para el control de versiones.
 - Sintaxis: “diff *archivo1.ext* *archivo2.ext*”

4. Comandos de búsqueda

En este apartado, veremos los comandos para realizar búsquedas en Linux:

- **locate**: utilizado para localizar un archivo.
 - Sintaxis: `"locate archivo"`
 - Variación:
 - `"locate -i nombrearchivo"`: para que la búsqueda no distinga entre mayúsculas y minúsculas.
 - `"locate -i palabra1*palabra2"`: para que la búsqueda saque todos los resultados que contengan las palabras estipuladas.
- **find**: similar al comando locate, pero con la diferencia de que este se usa para buscar archivos dentro de un directorio concreto.
 - Sintaxis: `"find ruta -name archivo"`
- **grep**: utilizado para hacer búsquedas de texto dentro de un archivo en concreto. Devuelve todas las líneas que contengan la palabra buscada. También suele usarse como filtro gracias al operador "|".
 - Sintaxis: `"grep palabra archivo"`
 - Variaciones cambio de nombre:
 - `"grep -i palabra archivo"`: la búsqueda no distingue entre mayúsculas y minúsculas.
 - `"grep -r palabra archivo"`: la búsqueda será recursiva.
 - `"grep -v palabra archivo"`: la búsqueda mostrará las líneas que NO coincidan con el patrón/palabra estipulada.
 - `"grep -n palabra archivo"`: la búsqueda muestra el número de líneas.

5. Comandos de sistema

A continuación, veremos los comandos básicos para la administración de los sistemas de Linux:

- **sudo**: este comando te permite realizar tareas que requieren permisos administrativos o root. No es recomendable usarlo como usuario diario.
 - Sintaxis: "sudo"
- **df**: para obtener un informe sobre el espacio del disco del sistema, que se muestra en porcentaje y KB.
 - Sintaxis: "df"
 - Variaciones:
 - "df -m": para ver el tamaño en MB.
- **du**: cuánto espacio ocupa un archivo o un directorio.
 - Sintaxis: "du"
- **chmod**: comando utilizado para cambiar los permisos de lectura, escritura y ejecución de archivos y directorios. Abreviación de "change mode".
 - Sintaxis: "chmod *-opciones permisos nombre_archivo*"
 - Variaciones:
 - **chmod -rwx archivo**: todos los permisos a la vez
 - r: permisos de lectura.
 - w: permisos de escritura.
 - x: permisos de ejecución.
 - **Permisos**:
 - 0 = --- = sin acceso
 - 1 = --x = ejecución
 - 2 = -w- = escritura
 - 3 = -wx = escritura y ejecución
 - 4 = r-- = lectura
 - 5 = r-x = lectura y ejecución
 - 6 = rw- = lectura y escritura
 - 7 = rwx = lectura, escritura y ejecución
- **chown**: Abreviación de "change owner". comando que nos ofrece la posibilidad de modificar el dueño o propietario de un archivo o carpeta en el sistema de archivos.
 - Sintaxis: "chown *user filename*"
- **jobs**: este comando muestra todos los trabajos actuales junto con sus estados.
 - Sintaxis: "jobs"
- **ps ux**: comando para ver los procesos que se están ejecutando en el sistema, así como su PID (Process ID)
 - Sintaxis: "ps ux"
- **kill**: comando para cerrar manualmente un proceso o programa.

- Sintaxis: “kill -*opciones proceso*”
- Variaciones:
 - “kill -9 pid”: obliga a los programas a detenerse inmediatamente y no se guarda el progreso.
 - “kill -15 pid”: obliga a los programas a detenerse con un margen de tiempo y guarda el progreso.
- **wget**: para descargar archivos de Internet.
 - Sintaxis: “wget *url_descarga*”
- **top**: inicia un administrador de tareas similar al de Windows. Mostrará una lista de los procesos en ejecución y la cantidad de CPU que utiliza cada proceso.
 - Variaciones:
 - htop: versión “colorida” de top
- **zip**: para comprimir tus archivos en un archivo zip.
 - Sintaxis: “zip -*opciones archivo archivos_para_comprimir*”
- **unzip**: para extraer los archivos comprimidos de un archivo zip.
 - Sintaxis: “unzip *archivo.zip*”
- **tar**: guarda archivos en un “tarball” similar a un archivo .zip.
 - Sintaxis: “tar -*opciones archivo*”
 - Variaciones:
 - “tar -vcf *archivo*”: El comando tar permite empaquetar varios archivos en uno solo, sin comprimirlos.
 - “tar -vxf *archivo*”: desempaquetar archivos, es decir para extraer todos los archivos individuales contenidos en un archivo tar.
 - -c: está opción es para comprimir.
 - -v: está opción es para extraer.
- **shutdown**: comando para apagar el equipo desde la terminal.
- **reboot**: comando para reiniciar el equipo desde la terminal.

6. Comandos para repositorios

Los siguientes comandos nos permitirán mantener y gestionar nuestros repositorios de Linux:

- **sudo apt-get update:** Actualizar los repositorios de nuestra distribución
- **sudo apt-get upgrade:** Actualizar los paquetes y programas que tenemos instalados.
- **sudo apt-get install:** Instalar uno o varios paquetes en nuestro sistema operativo.
- **sudo apt-get full-upgrade:** Actualizamos la totalidad de paquetes que tenemos instalados.
- **sudo apt-get remove:** Desinstalar uno o varios paquetes manteniendo sus archivos de configuración.
- **sudo apt-get purge:** Desinstalar los paquetes y los archivos de configuración de las utilidades desinstaladas.
- **sudo apt-get autoremove:** Para desinstalar las dependencias de programas que en su día eran necesarias, pero ahora ya no son necesarias.

7. Comandos de red

A continuación, veremos los comandos básicos para gestionar la configuración de red en Linux:

- **ping:** para verificar tu estado de conexión a un servidor web.
- **ifconfig:** este comando es utilizado para arrancar la interface de la tarjeta de red, pararla y realizar todas las configuraciones sobre dicha tarjeta. Muy útil para ver la IP de nuestro dispositivo, así como la máscara de red.
- **Dhclient -r:** libera la IP actual y obtener una nueva desde el servidor DHCP de tu sistema.
- **tracpath / traceroute:** muestran la ruta de red hasta un destino especificado mostrando los saltos hasta llegar al host.
- **Mtr:** combina el comando ping con el *tracpath* en un solo comando.
- **netstat -lptu:** es una herramienta de línea de comandos que muestra las conexiones de red (entrantes y salientes), tablas de enrutamiento y una serie de estadísticas de interfaz de red
 - muestra una lista con los puertos abiertos junto con procesos a los que corresponden.

8. Operadores lógicos

Por último, veremos los operadores básicos de Linux:

- **;**: para ejecutar comandos unos detrás de otros:
 - Sintaxis: *"comando1;comando2;comando3"*
- **&&**: Ejecutar el comando2 solo si se ejecuta el comando1. Si las dos expresiones son verdaderas entonces los dos comandos devuelven verdadero
 - Sintaxis: *"comando1 && comando2"*
- **||**: Ejecutar el comando2 solo si no se ejecuta el primero. La expresión global es verdadera si al menos uno de los comandos devuelve verdadero.
 - Sintaxis: *"comando1 || comando2"*
- **|**: el comando2 se ejecutará sobre la salida del comando1.
 - Sintaxis: *"comando1 | comando2"*

9. Conclusión.

Cualquier duda podéis contactar conmigo vía email en **contandobitscontacto01@gmail.com** o en mi Instagram **@contandobits_**.

NOTA: este PDF estará en constante actualización así que... ¡estate atento al canal!