



LINUX

Cuarta Parte

Abril 2023

Comandos más usuales de Linux

Comando: mkdir

Propósito: Crea uno o más directorios

Sintáxis: mkdir [opciones] dir1 [dir2.....dirn]

Opciones posibles:

- p permite crear una cadena de directorios completa.
- m permite especificar los permisos de nuevos directorios (complementar con el comando chown).



Comandos más usuales de Linux

Comando: rmdir

Propósito: Borra uno o más directorios, debiendo estar **vacíos** para ser borrados.

Sintaxis: `rmdir dir1 [dir2 dir3 dir4dirn]`

Posibilidades:

- p permite borrar una cadena de directorios completa, para lo cual los directorios deben estar vacíos.



Comandos más usuales de Linux

Comando: `wc`

Propósito: Cuenta líneas, palabras, caracteres en los archivos nombrados ó en las entradas estándar si no se da nombre de archivo.

Sintaxis: `wc [opciones] [nombres]`

Posibilidades de uso:

- l cuenta únicamente la cantidad de líneas.
- w cuenta únicamente la cantidad de palabras
- c cuenta únicamente la cantidad de caracteres



Comandos más usuales de Linux

Comando: find

Propósito: Busca archivos en el filesystem que concuerden con un criterio de selección, ejecutando sobre los mismos una determinada acción (específica).

El comando find recorre el árbol recursivamente.

Sintaxis: find camino expresión

Posibilidades de uso:

-name nomDelArchivo

Es verdadero si el patrón “nomdelArchivo” coincide con el nombre del archivo encontrado.



Comandos más usuales de Linux

-links n

(continuación uname)

Indica el número de links que debe tener el archivo

-atime n

verdadera si hace más de n días que el archivo fue accedido

-mtime n

Es verdadero si el archivo ha sido modificado hace más de n días.

-user nombre

Es verdadero si el dueño del archivo encontrado coincide con “nombre”



Comandos más usuales de Linux

-group nombre

(continuación comando find)

Es verdadero si el grupo al que corresponde el archivo encontrado coincide con “nombre”.

-size n (c)

Es verdadera si el archivo tiene una longitud n de bloques. Si el valor de n viene seguido de una “c” la indicación es en caracteres.

-print

Si está o existe es siempre verdadera y hace que se imprima el camino del archivo localizado.



Comandos más usuales de Linux

-exec comando { } \;

(continuación comando find)

Ejecuta el comando “comando” sobre los archivos encontrados, los “{ }” representa el nombre del archivo, es un requerimiento de la sintaxis y el “;” (también una imposición de sintaxis) es una separación necesaria de las sucesivas ejecuciones de comandos generadas por cada archivo encontrado.

-perm num

Verdadero si el archivo encontrado tiene los permisos iguales a “num”; num expresado en octal y es indicación absoluta.



Comandos más usuales de Linux

Posibilidades de uso:

(continuación comando find)

```
find / -name planilla -print
```

Es el uso más común del comando find y lo que hace es mostrar por pantalla el directorio donde se encuentra el archivo “planilla”. La barra “/” indica que debe iniciar la búsqueda desde la raíz.

```
find . -name planilla -print
```

Idem anterior pero la búsqueda comienza en el directorio que estamos trabajando.



Comandos más usuales de Linux

(continuación comando find)

```
find / -name core -exec rm {} \;
```

Comienza la búsqueda de los archivos “core” a partir de la raíz y los borra.

```
find / -perm 777 -print
```

A partir de la raíz busca todos los archivos que tengan permisos 777 y muestra su nombre por pantalla. Esta línea es ideal para conocer cuáles son los archivos que tienen bajos niveles de seguridad.



Comandos más usuales de Linux

Comando: grep

Propósito: Busca un patrón en uno o más archivos, ejecutando las acciones indicadas por las opciones. Cuando no se indica acción entonces muestra por pantalla las líneas que contienen el patrón.

Sintaxis:

grep [opciones] expresión archivo(s)

Posibilidades de uso:

- v muestra todas las líneas excepto las que tienen el patrón.
- c cuenta las líneas que tienen el patrón.



Comandos más usuales de Linux

(continuación)

- l Muestra sólo los nombres de archivos que contiene el patrón, uno en cada línea.
- n Impresión de las líneas que contienen el patrón es precedidas por el número de línea en el archivo.
- b la impresión de las líneas que contiene el patrón es precedida por el desplazamiento de la línea en bytes desde el comienzo del archivo.
- s suprime los mensaje de error producidos por la no existencia o imposibilidad de leer un archivo. Es útil cuando se especifica varios.



Comandos más usuales de Linux

(continuación)

-i hace que el patrón coincida independientemente de si está en minúscula o mayúscula.

grep carga new*

Linux busca el patrón “carga” en todos los archivos cuyo nombre comience con “new” en el directorio de trabajo, e imprime las líneas que lo contienen.

grep '[V v] er' new

Linux busca el patrón “Ver” o “ver” en todos los archivos cuyo nombre comience con “new” en el directorio de trabajo, e imprime todas las líneas que lo contienen.



Comandos más usuales de Linux

(continuación de usos)

grep -i ver new

Busca el patrón “ver” , con mayúsculas o minúsculas, en todos los archivos cuyos nombres comienzan con “new”, e imprime todas las líneas que lo contienen. El patrón puede tener una, algunas o todas sus letras en mayúsculas o minúsculas, mientras que el ejemplo anterior sólo es para la primer letra



Comandos más usuales de Linux

Comando: `df` (disk free)

Propósito: permite mostrar por pantalla la cantidad de espacio libre y ocupado de los archivos (filesystems) a los que el usuario que lo invoca tiene permiso de lectura.

Sintaxis: `df`



Comandos más usuales de Linux

Comando: df (continuación)
Posibilidades de uso:

\$df Por defecto muestra 6 columnas, sin opción, size es en bloques de 1KB:

Filesystem	1k-blocks	Used	Available	Use%	Mounted on
/dev/sda1	588352	210220	335124	39%	/boot

Sistema de archivo

Tamaño, según opción

Lo usado

Lo disponible

Lo usado en %

Donde está montado



Comandos más usuales de Linux

Comando: df (continuación)

Posibilidades de uso:

\$ df -h formato humano

\$df -m formato en megabytes

\$df -k por defecto

EJ:

\$df -h

Filesystem	size	used	avail	used %	Monted on
devtmpfs	7.8G	0	7.8G	0%	/dev



Contiene los files especiales del dispositivo para todos los dispositivos, está montado en /dev



Sesión de trabajo

- ☐ Una vez que el usuario se conecta al sistema el usuario tiene abierta su ``sesión de trabajo“.
- ☐ Podrá hacer tareas simples como leer el correo electrónico o tareas más complejas como escribir y ejecutar programas para resolver algún problema.



Tipo de comando: Sesión de trabajo

Comando: logname

Propósito: Muestra el nombre con el cual el usuario ingresó al sistema

Sintaxis: logname



Tipo de Comando: Sesión de trabajo

Comando: who

Propósito: El comando who permite conocer datos de las sesiones activas: puede listar o mostrar los nombres de los usuarios que están trabajando en el sistema actualmente, su terminal, el tiempo que han estado conectado (la hora de ingreso o login) y el nombre del host desde el que se han conectado.

Sintaxis: who [opciones]



Tipo de Comando: Sesión de trabajo

Comando: who

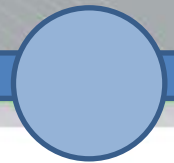
(continuación)

Posibilidades de uso: (opciones)

- u lista únicamente los usuarios presentes
- H muestra el encabezado de las columnas en la salida

Por ejemplo si ejecutamos :
who -uH





Tipo de Comando: Sesión de trabajo

Ejecutamos:
who -uH

(continuación)

Salida:

NAME	LINE	TIME	IDLE	PID	COMMENT
Juan	ttpy3	Ago 15 10:02	.	4578	

Login usuario

Hora en la que
entró usuario

Número que identifica
un proceso

Punto indica que la terminal
ha tenido actividad en el
último minuto

Nombre de la línea de entrada, como
aparece en /dev



Tipo de Comando: Sesión de trabajo

Comando: who

(continuación)

IDLE indica el tiempo en horas y minutos de inactividad registradas en la línea "line". Si han transcurrido más de 24 horas desde la última actividad, aparece la palabra old. Este campo es útil para averiguar si un usuario está trabajando o no en la terminal.

Otras posibilidades de uso: (opciones)

- t parecido a -u pero muestra el estado de la terminal.
- q opción rápida incluye las demás. Muestra nombres y la cantidad de usuarios que han registrado su ingreso.
- m muestra información



Tipo de Comando: Sesión de trabajo

Comando: whoami

Propósito: Permite visualizar bajo cuál login se está trabajando. Es decir imprime el identificador de usuario efectivo.

Sintaxis: whoami



Tipo de Comando: Sesión de trabajo

Comando: `tty`

Propósito: Permite visualizar el nombre del fichero de la terminal de la entrada estándar. Es decir muestra el pathname de la terminal en el sistema como dispositivo.

Sintaxis: `tty [opciones]`

`tty` sin opciones, muestra el nombre del fichero asociado al terminal actual.



Tipo de Comando: Sesión de trabajo

Comando: `tty`

(continuación)

Sintaxis: `tty [opciones]`

Ejemplo:

`$ tty`

`/dev/tty3` estamos trabajando en la terminal 3 del sistema.

`$echo "help" > /dev/tty3` aparecerá help en su pantalla



Tipo de Comando: Sesión de trabajo

Comando: clear

Propósito: limpia la pantalla

Sintaxis: clear



Tipo de Comando: Sesión de trabajo

Comando: write

Propósito: manda un mensaje a la pantalla de un usuario conectado al mismo sistema.

Comunicación bidireccional única (dúplex).

Sintaxis: write usuario [tty]



Tipo de Comando: Sesión de trabajo

Comando: write

(continuación)

Sintaxis: write usuario [tty]

Ejemplo: (los dos usuarios deben estar conectados)

```
$write Pedro
```

```
Hola, Pedro
```

```
^d                (apretar tecla ctrl + d)
```

El usuario Pedro verá en su pantalla:

```
Messeg from  juan tty0 [ jun3  08:10:31]
```

```
Hola,Pedro
```

```
EOF
```

Se contesta tipiendo: write juan



Tipo de Comando: Sesión de trabajo

Comando: talk

Propósito: permite establecer comunicación bidireccional simultánea (fullduplex) por consolas entre dos usuarios conectados a la misma o diferente máquina. Talk particiona las pantalla para tener la vista propia y recibir la vista del otro usuario.

Sintaxis: talk usuario [tty3]



Tipo de Comando: Sesión de trabajo

Comando: wall

Propósito: permite enviar un mensajes a todos los usuarios del sistema. Mensaje masivo

Sintaxis: wall mensaje

Por ejemplo: el usuario root alerte a los usuarios.

```
$ Wall
```

```
Hola a todos, mensaje del administrados
```

```
^d
```



Tipo de Comando: Sesión de trabajo

Comando: cal

Propósito: se utiliza para mostrar el calendario del mes y año indicado.

Sintaxis: cal [opciones] [mes] [año]

Si no se coloca argumentos, la salida es la del mes y año en curso.

Si se coloca sólo el año, se imprimen todos los meses del mismo.

Si se coloca sólo el mes, es el mes del corriente año el que se imprime en pantalla.



Tipo de Comando: Sesión de trabajo

Comando: cal

(continuación)

Sintaxis: cal [opciones] [mes] [año]

Tener presente que el año se indica completo. Si se coloca 18, se entiende que es el año 18 y no 2018. El año debe estar entre 1 y 9999, el mes entre 1 y 12, y la suficiente cantidad de caracteres para que identifiquen el mes (inglés).



Tipo de Comando: Sesión de trabajo

Comando: cal

(continuación)

Sintaxis: cal [opciones] [mes] [año]

Opciones:

- 1 sólo muestra el mes actual
- 3 muestra mes previo, actual y el siguiente
- s muestra el domingo como primer día de la semana
- m muestra el lunes como primer día de la semana.
- j muestra fechas julianas (días ordenados, numerados desde el 1 de Enero)
- y muestra un calendario para el año actual.



Tipo de Comando: Sesión de trabajo

Comando: `uname`

Propósito: Muestra información del sistema y del ordenador.

Sintaxis.

`uname [opción]`

Algunas de las opciones más comunes son:

`uname -a`: Muestra toda la información de nuestro sistema y nuestro ordenador.

`uname -r`: La versión del Kernel de nuestro sistema.



Tipo de Comando: Sesión de trabajo

Comando: `uname`

(continuación)

`uname -v`: La fecha de publicación del Kernel de nuestro sistema.

`uname -n`: Nombre de nuestro ordenador (el que asignamos al instalar el SO).

`uname -m`: Arquitectura de nuestro procesador (i386, i486, i586, i686, etc).

`uname -s`: Nos muestra el nombre del kernel.

`uname -o`: Nos muestra el nombre del sistema operativo.



Tipo de Comando: Sesión de trabajo

Ejemplos:

(continuación)

\$uname -a

Linux sasanchez 2.6.28- 16-generic #55-Ubuntu SMP
Tue Oct 15 19:36:48 UTC 2009 i 686 GNU /Linux

El mismo ejemplo fragmentando la información:

uname -s

Linux

uname -n nombre de nuestro ordenador (asignamos al
instalar el SO), es el host del nodo de la red.

sasanchez



Tipo de Comando: Sesión de trabajo

Comando: uname
sintaxis)

(continuación

\$uname -r

2.6.28- 16-generic

\$uname -v

#55-Ubuntu SMP Tue Oct 15 19:36:48 UTC 2009

\$uname -m

i 686

\$uname -o

GNU /Linux



Tipo de Comando: Sesión de trabajo

Comando: `uname`

(continuación sintaxis)

Si queremos conocer dos o más datos del sistema o de nuestro ordenador a la vez, por ejemplo, queremos que se muestre el nombre de nuestro ordenador junto con la versión del Kernel, no tenemos más que combinarlos para ello combinamos opciones, así:

```
$uname -nr
```



BIBLIOGRAFIA

Título: Advanced Linux programming

Autor: mark L. Mitchell, Alex Samuel, Jeffrey Oldham

Editorial: New Riders Publishing

ISBN: 0735710430

Disponible gratuitamente en [http: /www.advancedlinuxprogramming.com/](http://www.advancedlinuxprogramming.com/)

Título: Programación en Linux

Autor: Kurt Wall

Editorial: Prentice Hall

ISBN: 0-7897-2215-1



GRACIAS

