

COMANDOS LINUX

ADOLFO SANZ DE DIEGO

OCTUBRE 2017

1 ACERCA DE

1.1 AUTOR

- Adolfo Sanz De Diego
 - Blog: asanzdiego.blogspot.com.es
 - Correo: asanzdiego@gmail.com
 - GitHub: github.com/asanzdiego
 - Twitter: twitter.com/asanzdiego
 - LinkedIn: in/asanzdiego
 - SlideShare: slideshare.net/asanzdiego

1.2 LICENCIA

- Copyright:
 - Antonio Sarasa Cabezuelo

1.3 FUENTE

- Las slides y sus fuentes las podéis encontrar en:
 - <https://github.com/asanzdiego/curso-intro-linux-web-sql-2017>

2 COMANDOS BÁSICOS

2.1 AYUDA COMANDOS

- El comando `man nombrecomando` muestra el manual del comando. Una vez dentro para salir hay que pulsar la tecla q.
- También podemos probar con `nombrecomando -h` o `nombrecomando --help` o con `info nombrecomando`.

2.2 CAMBIO DE DIRECTORIO

- Para cambiar de directorio se usa el comando `cd directorio_destino`.
- Usamos `cd ..` para ir al directorio superior.
- Usamos `cd` sin argumentos para volver a la carpeta personal.

2.3 SITUACIÓN ACTUAL

- El comando `pwd` imprime la ruta del directorio en el que nos encontramos en este momento.

2.4 LISTADO

- El comando `ls` muestra los nombres de los ficheros y subdirectorios contenidos en el directorio en el que se está. Sólo se obtienen los nombres de los ficheros, sin ninguna otra información.

2.5 LISTAS OCULTOS

- El comando `ls -a` muestra todos los ficheros incluyendo algunos que están ocultos para el usuario (aquellos que comienzan por un punto).
- Observar que el fichero punto `.` indica el directorio actual y el doble punto `..` el directorio padre, que contiene al actual.

2.6 LISTADO ORDENADOS

- `ls -c` muestra ordenando por día y hora de creación.
- `ls -t` muestra ordenando por día y hora de modificación.
- `ls -r` muestra el directorio y lo ordena en orden inverso.
- `ls subdir` muestra el contenido del subdirectorio `subdir`.

2.7 OTROS LISTADOS

- `ls --color` muestra el contenido del directorio coloreado: verde para los ejecutables, azul las carpetas, fucsia las imágenes, rojo los comprimidos, ...
- `ls -l` muestra toda la información de cada fichero incluyendo: protecciones, tamaño y fecha de creación o del último cambio introducido, ...

2.8 COMBINACIÓN DE OPCIONES

- Las opciones anteriores pueden combinarse como por ejemplo `ls -cr` que muestra el directorio ordenando inversamente por fechas.

2.9 FILTRADO

- Muchos comandos admite los caracteres de sustitución:
 - * que representa cualquier conjunto o secuencia de caracteres
 - y ? que representa cualquier carácter pero sólo uno.
- Por ejemplo `ls *.gif` muestra todos los nombres de ficheros que acaben en `.gif` y `ls file?` muestra todos los ficheros cuyos nombres empiecen por `file` y tengan un nombre de cinco caracteres.

2.10 CREACIÓN DE SUBDIRECTORIOS

- El comando `mkdir` permite crear un nuevo subdirectorio. La sintaxis es `mkdir subdir1` donde `subdir` es el nombre del directorio que se va a crear.

2.11 BORRADO DE SUBDIRECTORIOS

- El comando `rmdir` borra uno o más directorios del sistema siempre que estos subdirectorios estén vacíos. La sintaxis es `rmdir subdir1` donde `subdir` es el nombre del directorio que se va a eliminar.

2.12 COPIA DE FICHEROS

- El comando `cp` permite hacer la copia de un fichero. La sintaxis del comando es `cp file1 file2` que indica que hace una copia de `file1` y le llama `file2`. Si `file2` no existía, lo crea con los mismos atributos de `file1`, y en caso de existir su contenido es sustituido por el de `file1`. El fichero `file2` estará en el mismo directorio que `file1`.

2.13 MOVER/RENOMBRAR FICHEROS

- El comando `mv` permite mover o renombrar un fichero. La sintaxis es `mv file1 file2` y realiza la misma función que `cp` pero eliminando el fichero original. Desde la visión del usuario, se cambia el nombre a `file1` por `file2`.
- Si los nombres que aparecen son de directorios entonces el comando `mv namedir1 namedir2` cambia el nombre del subdirectorio `namedir1` por `namedir2`.

2.14 ENLACES

- Un mismo fichero puede estar repetido con más de un nombre y poder acceder a él desde más de un directorio. Esto último se denomina enlaces múltiples a un fichero, y se crean con el comando **ln**: `ln file1 file2` Así el fichero file1 tiene dos nombres: file1 y file2.

2.15 BORRADO DE FICHEROS

- El comando `rm` elimina uno o más ficheros de un directorio en el cual tengamos permiso de escritura. La sintaxis es `rm file1 file2`

2.16 PERMISOS

- Los permisos de cada fichero se pueden ver con el comando ls -l.
- Estos permisos son:
 - **r**: de lectura (o listar en directorios)
 - **w**: de escritura (o crear y borrar ficheros en directorios)
 - **x**: de ejecución (o buscar y utilizar ficheros en directorios)

2.17 CAMBIAR PERMISOS

- Para cambiar los permisos de un fichero se emplea el comando **chmod [quien] operacion permiso file** donde:
 - quien: indica a quien afecta el permiso que se desea cambiar:
 - **u**: para el usuario propietario del archivo,
 - **g**: para el grupo del usuario propietario del archivo,
 - **o**: para los otros usuarios
 - **a**: para todos los anteriores. (valor por defecto).
 - oper: indica si el permiso se da usando un + o se quita usando un -.

2.18 CAMBIAR DUEÑO

- El comando **chown** se emplea para cambiar de propietario a un determinado conjunto de ficheros: chown newowner file1 file2
...
 - Sólo lo puede emplear el actual propietario de los mismos.
 - Los nombres de propietario se encuentran almacenados en el fichero **/etc/passwd**.

2.19 CAMBIO DE GRUPO

- El comando `chgrp` se emplea para cambiar el grupo al que pertenece un fichero : `chgrp newgroup file1 file2...`
- Los grupos de usuarios están almacenados en el fichero `/etc/group`.

2.20 VISUALIZACIÓN CON CAT

- El comando `cat filename` permite visualizar el contenido de uno o más ficheros de forma no formateada, y copiar uno o más ficheros como apéndice de otro ya existente.

2.21 CONCATENAR FICHEROS

- `cat file1 file2 >file3`: El contenido de los ficheros file1 y file2 es almacenado en file3.
- `cat file1 file2 >>file3`: el contenido de file1 y file2 es añadido al final de file3.
- `cat >file1`: Acepta lo que se introduce por el teclado y lo almacena en file1 (se crea file1). Para terminar se emplea d

2.22 VISUALIZACIÓN CON MORE

- El comando `more filename` permite visualizar el contenido de un fichero pantalla por pantalla.
- Para pasar de pantalla se utiliza la barra espaciadora o la tecla enter.
- Para salir se pulsa d o q.

2.23 VISUALIZACIÓN CON LESS

- El comando `less filename` permite visualizar el contenido de un fichero pantalla pantalla.
- Para pasar de pantalla se utilizan las flechas arriba y abajo.
- Para salir se pulsa d o q.

2.24 BÚSQUEDA EN FICHEROS

- El comando `grep 'conjunto de caracteres' file1 file2 file3` busca una palabra, clave o frase en un conjunto de directorios, indicando en cuáles de ellos la ha encontrado.
- Se pueden utilizar expresiones regulares de la forma: `grep [-opcion] expresión_regular [referencia...]`

2.25 OPCIONES GREP

- **c**: escribe el número de las líneas que satisface la condición.
- **i**: no se distinguen mayúsculas y minúsculas.
- **l**: escribe los nombres de los ficheros que contienen líneas buscadas.
- **n**: cada línea es precedida por su número en el fichero.

2.26 MAS OPCIONES GREP

- **s**: no se vuelcan los mensajes que indican que un fichero no se puede abrir.
- **v**: se muestran sólo las líneas que no satisfacen el criterio de selección.

2.27 EJEMPLOS GREP

- `grep '^d' text` recupera las líneas que comienzan por d.
- `grep '^[^d]' text` recupera las líneas que no comienzan por d.
- `grep -v '^C' file1 > file2` quita las líneas de file1 que comienzan por C y lo copia en file2.

2.28 REDIRECCIONES

- Se puede redirigir la salida de un comando usando los operadores:
 - (>) redirige la salida estándar hacia el fichero indicado y en caso de no existir se crea.
 - (<) redirige la entrada estándar desde un determinado fichero
 - (>>) redirige la salida estándar hacia otro fichero, pero añadiendo dicha salida al final de ese fichero, sin sobreescribir el contenido original.

2.29 EJEMPLOS REDIRECCIONES:

- `date >> archivo` el fichero archivo contendrá información sobre todas las veces que hemos entrado en el sistema.
- `cat file1 file2 > file3` añade al fichero file2 al final de file1 y al conjunto lo llama file3

2.30 TUBERÍAS

- Una tubería (|) permite comunicar la salida estándar de un comando con la entrada estándar de otro.
- Por ejemplo `ls | mail juan` envía a juan una lista de los ficheros del sistema.
- Con el operador de tubería se pueden empalmar tantos comandos como se desee.

2.31 EJECUCIÓN SEGUNDO PLANO

- El carácter & al final permite realizar una ejecución en segundo plano recuperando inmediatamente el control del terminal. Para ello se añade el carácter & al final del comando de ejecución.

2.32 PARAR UN PROCESO

- Para detener la ejecución de un proceso se puede utilizar el comando `kill` `númerodeproceso`.

2.33 CONTINUAR EN SEGUNDO PLANO

- Cuando se sale del sistema si hay algún proceso ejecutándose en segundo plano se para salvo que se use el comando `nohup nombreprograma`. En este caso todas las salidas del programa se dirigen a un fichero llamado `nohup.out`.

2.34 CAMBIAR LA PRIORIDAD

- Para darle al programa la prioridad mínima habría que invocarlo con `nice -19 nombreprograma &`.
- Para darle al programa la prioridad máxima habría que invocarlo con `nice --20 nombreprograma &`.

2.35 LISTADO PROCESOS

- El comando `top` muestra una lista de los procesos que se están ejecutando.

2.36 AGRUPACIÓN DE FICHEROS

- El comando `tar -cvf nombre_archivo.tar fichero1 fichero2 ...` agrupa varios ficheros en uno solo “archivo” tar

2.37 COMPRESIÓN DE FICHEROS

- El comando **gzip fichero** comprime fichero (que es borrado) y se crea un fichero con nombre fichero.gz.

2.38 AGRUPACIÓN Y COMPRESIÓN

- El comando `tar -czfv archivo.tar.gz ficheros` empaqueta y comprime ficheros.

2.39 DESCOMPRIMIR FICHEROS

- El comando `tar -xzvf archivo.tar.gz` descomprime `archivo.tar.gz`

3 OTROS COMANDOS

3.1 ESPACIO CARPETAS

- El comando **du** permite conocer el espacio en bloques ocupado en el disco por un determinado directorio y todos los subdirectorios que cuelgan de él.
- Para obtener la información en bytes se debe emplear la opción -h: **du -h**

3.2 ESPACIO PARTICIONES

- El comando **df** informa del espacio usado por las particiones del sistema que se encuentren montadas.
- Para obtener la información en bytes se debe emplear la opción **-h**: **df -h**

3.3 FECHA

- El comando `date` muestra el día y la hora.

3.4 CALENDARIO

- El comando `cal` muestra el calendario. Tiene diversas opciones.
Por ejemplo `cal 1945` mostraría el calendario del año 1945.

3.5 USUARIOS CONECTADOS

- El comando `who` muestra los usuarios están conectados al ordenador en ese momento, en qué terminal están y desde qué hora.

3.6 QUIEN SOY

- El comando `whoami` te indica quien eres.

3.7 TIEMPO SIN APAGAR

- El comando `uptime` muestra el tiempo que lleva encendido nuestro ordenador.

3.8 NOMBRE DE LA MÁQUINA

- El comando `hostname` muestra el nombre de la máquina.

3.9 HARDWARE

- El comando `Ishw` muestra todas las características del hardware.

3.10 DISPOSITIVOS PCI

- El comando `lspci` muestra los diferentes dispositivos PCI.

3.11 DISPOSITIVOS USB

- El comando `lsusb` muestra los dispositivos tengo en los bus USB

3.12 COMANDOS USADOS

- El comando `history` muestra los comandos usados por el usuario en orden cronológico.

3.13 PERMISOS POR DEFECTO

- El comando `umask` muestra los permisos con los que el usuario creara sus archivos por defecto

3.14 VERSIÓN KERNEL

- El comando `uname -r` muestra la versión del kernel

3.15 CERRAR PROGRAMAS

- El comando `xkill` permite cerrar un programa bloqueado.

3.16 GRUPOS

- El comando `groups` muestra los grupos del sistema a los que pertenece un usuario.

3.17 LIMPIAR TERMINAL

- El comando `clear` limpia el terminal.

3.18 PRIMERAS LÍNEAS

- El comando `head` muestra las primeras líneas de un fichero

3.19 ULTIMAS LÍNEAS

- El comando `tail` muestra las últimas líneas de un fichero