

Tema II: EL SHELL. COMANDOS BÁSICOS DE LINUX

ÍNDICE:

Introducción.

Sintaxis de los comandos.

Alias.

Directorios de Linux.

Algunos comandos sencillos de Linux.

- *date (fecha)*
- *cal (calendario)*
- *who (usuario)*
- *whoami (sesión y terminal)*
- *man (manuales)*
- *clear (limpiar)*
- *directorio personal*
- *ls (listados)*
- *mkdir (crear directorios)*
- *rmdir (borrar directorios)*
- *cd (cambiar de directorio)*
- *pwd (directorio actual)*
- *cp (copiar directorios)*
- *mv (mover directorios)*
- *rm (borrar directorios)*
- *file (características de ficheros)*
- *exit (salir)*
- *Comandos de impresión*

Cualquier usuario de **Linux** acabará antes o después relacionándose con el sistema empleando el modo texto, y no los entornos gráficos. Este modo se basa en la ejecución de una serie de comandos, que son interpretados por un programa o **shell**. **Linux** dispone de varios de estos programas pero el más habitual es conocido como **bash** o **Bourne Shell**. Si **Linux** se ha arrancado en modo texto el sistema arranca de forma directa el **shell** y queda a la espera de introducción de nuevos comandos. Si se ha arrancado en modo gráfico se puede acceder al **shell** de dos formas:

- Se puede acceder al **shell** del sistema presionando alguna de las siguientes combinaciones de teclas (hay que tener en cuenta que en el Linex de las aulas no funcionan las combinaciones de teclas y hay que entrar ejecutando el programa `gnome-terminal`):

> **<ctrl>+<alt>+<F1>**

> **<ctrl>+<alt>+<F2>**

> **<ctrl>+<alt>+<F3>**

> **<ctrl>+<alt>+<F4>**

> **<ctrl>+<alt>+<F5>**

> **<ctrl>+<alt>+<F6>**

Esto hace que el sistema salga del modo gráfico y acceda a alguna de las seis consolas virtuales de **Linux**, a las cuales también se puede acceder cuando se arranca en modo de texto. Para volver al modo gráfico hay que presionar alguna de las siguientes combinaciones de teclas (igual que antes, este procedimiento no funciona en el Linex de las aulas):

> **<ctrl>+<alt>+<F7>**

> **<ctrl>+<alt>+<F8>**

- La segunda forma es más cómoda y menos radical permitiendo acceder al shell desde el mismo entorno gráfico. Para esto hay que abrir un programa llamado terminal o consola, por ejemplo: **kconsole** (en el entorno **KDE**), **xterm**, **gnome-terminal** (en **GNOME**), etc.

Introducción

Existen una serie de nociones básicas que hay que tener en cuenta a la hora de introducir los comandos. En primer lugar citaremos las siguientes:

- Los comandos hay que teclearlos exactamente.
- Las letras mayúsculas y minúsculas se consideran como diferentes.
- En su forma más habitual, el sistema operativo utiliza un signo de **\$** como **prompt** para indicar que está preparado para aceptar comandos, aunque este carácter puede ser fácilmente sustituido por otro u otros elegidos por el usuario. En el caso de que el usuario acceda como administrador este signo se sustituye por **#**.
- Cuando sea necesario introducir el nombre de un fichero o directorio como argumento a un comando, **Linux**, permite escribir las primeras letras del mismo y realiza un autorrellenado al presionar la tecla del **tabulador**. Si no puede distinguir entre diversos casos rellenará hasta el punto en el que se diferencien. Por ejemplo, supongamos una carpeta con los siguientes directorios:

Programas

Documentos_proyecto

Documentos_privados

Al escribir **cd Pr<tab>** **Linux** rellenará el resto del contenido hasta escribir **cd Programas**. Por el contrario al escribir **cd D<tab>** escribirá **cd Documentos_pr**.

Sintaxis de los comandos

Los comandos tienen la siguiente sintaxis:

programa arg1 arg2 ... argn

Se observa que, en la "línea de comandos", se introduce el programa seguido de uno o varios argumentos. Así, el intérprete ejecutará el programa con las opciones que se hayan escrito.

Cuando se quiere que el comando sea de varias líneas, se separa cada línea con el carácter barra invertida (\). Además, cuando se quiere ejecutar varios comandos en la misma línea, los separa con punto y coma (;). Por ejemplo:

```
# make modules ; make modules_install
```

En los comandos, también se puede utilizar los siguientes comodines:

El asterisco (*) es equivalente a uno o más caracteres en el nombre de un archivo. Ejm: *ls *.c* lista todos los archivos con extensión c.

El signo de interrogación (?) es equivalente a un único carácter. Ej.: *ls curso.te?* lista el archivo *curso.tex* completando el último carácter.

Un conjunto de caracteres entre corchetes es equivalente a cualquier carácter del conjunto. Ej.: *ls curso_linux.[aeiou]x* lista *curso_linux.tex* seleccionando la e del conjunto.

Alias

Un "alias" es un nombre alternativo para un comando. Así, en lugar de escribir el comando propiamente dicho, escribiríamos el alias de dicho comando.

Un alias se puede definir por varios motivos, por ejemplo:

Dar nombres familiares a comandos comunes:

```
alias md='mkdir'
```

Crearía un alias para el comando *mkdir*, similar al de DOS.

Dar nombres a comandos largos:

```
alias tbz2='tar -cv --use-compress-program=bzip2 -f'
```

Crearía un alias para el comando *tar* para que use el compresor *bzip2* en lugar de *gzip*.

Para no tener que escribir todos los alias siempre que entremos al sistema, escribiremos dicho alias en el archivo */.bash_profile*

Directorios en Linux:

Bajo Linux el espacio de archivos que es visible a los usuarios está basado en una estructura de árbol, con la raíz en la parte superior (/). El directorio más alto, /, es conocido como directorio raíz.

/

___bin

___dev

___etc

____ **home**
 ____ **Seccion01**
 ____ **Seccion02**
____ **lib**
____ **usr**
____ **proc**
____ **root**
____ **sbin**
____ **var**

Por lo general, la mayor parte del sistema operativo reside en dos sistemas de archivos: el sistema de archivos raíz, conocido como */*, y el sistema de archivos montado bajo */usr*. El directorio **bin** contiene programas ejecutables conocidos como binarios. Estos programas son archivos de sistema esenciales. Muchos de los comandos como *ls*, son en realidad programas que se encuentran en este directorio.

El directorio **sbin** también se utiliza para guardar archivos binarios de sistema. La mayoría de los archivos de este directorio se usa para propósitos de la administración del sistema.

El directorio **etc.** contiene muchos de los archivos de configuración del sistema Linux. Son los archivos que personalizan Linux. P. ej. el archivo de contraseñas, *passwd*, se encuentra aquí.

Las bibliotecas compartidas que emplean los programas cuando se ejecutan están guardadas en el directorio */lib*. Mediante el uso de estas bibliotecas compartidas, muchos programas pueden utilizar el mismo código, y estas bibliotecas se pueden guardar en un lugar común, reduciendo el tamaño de los programas al momento de su ejecución.

El directorio **/dev** contiene archivos especiales conocidos como archivos de dispositivo, los cuales se usan para tener acceso a todo tipo de hardware del sistema. Por ejemplo, el

archivo */dev/mouse* se utiliza para la lectura de entrada del mouse.

/proc es un sistema de archivos virtual. Se usa para leer de la memoria información de procesos.

El directorio **/home** es el directorio base para los directorios personales de los usuarios.

El directorio **/var** guarda los archivos que tienden a cambiar de tamaño a lo largo del tiempo. Por lo general, los diversos archivos de bitácora de sistema se encuentran bajo este directorio. El directorio */var/spool* y sus subdirectorios se utilizan para guardar datos que son de manera transitiva, como el correo y las noticias recién recibidas o puestas en cola para transmisión a otro sitio.

El directorio **/usr** y sus subdirectorios contienen algunos de los programas más importantes del sistema. Por lo general, los subdirectorios de */usr* contienen los grandes paquetes de software que se instalan.

Algunos Comandos Sencillos de LINUX

date Muestra por pantalla el día y la hora.

cal 1949 Muestra el calendario del año 1949.

cal 05 1949 Muestra el calendario de mayo de 1949.

who Indica qué usuarios tiene el ordenador en ese momento, en qué terminal están y desde qué hora.

whoami Indica cuál es la terminal y la sesión en la que se está trabajando.

man comando: Todos los manuales de **Linux** están dentro del propio sistema operativo, y este comando permite acceder a la información correspondiente al comando **comando**. Por ejemplo con **man who** aparecerá por pantalla y de forma formateada por páginas, la explicación del comando **who**. Se puede navegar a través de estas páginas con los cursores del teclado, y presionando **q** para salir.

man ls: muestra la información sobre el comando de visualización de la pantalla.

man dir: muestra la información sobre el comando que contiene los directorios.

man mkdir: muestra la información sobre la creación de archivos.

man rmdir: muestra la información sobre borrar ficheros.

man pico: muestra la información sobre el uso de este editor de texto.

Clear Este comando limpia la consola.

Directorio Personal. Como se ha visto anteriormente el directorio personal es un directorio con un determinado nombre asignado a un usuario. Los directorios personales habitualmente son subdirectorios de **/home** (en algunos casos se utiliza **mnt**, u otro subdirectorio de orden inferior). Generalmente el nombre coincide con el del **nombre de usuario**, aunque puede no ser así, y varios usuarios pueden estar trabajando en el mismo directorio. Cada usuario de **Linux** puede crear una estructura en árbol de subdirectorios y archivos tan compleja como desee bajo su directorio personal pero normalmente nunca fuera de él.

Comando ls (listado del contenido de directorios): Una de las acciones más habituales a la hora de trabajar es mostrar el contenido de un directorio, para lo cual existen herramientas gráficas. No obstante el **shell** incluye un programa con este mismo fin: **ls**. **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.

ls -a Muestra todos los ficheros incluyendo algunos que ordinariamente están ocultos para el usuario (aquellos que comienzan por un punto). Se recuerda que el fichero punto **.** indica el directorio actual y el doble punto **..** el directorio padre, que contiene, al actual.

ls -l Esta es la opción de lista larga: muestra toda la información de cada fichero incluyendo: protecciones, tamaño y fecha de creación o del último cambio introducido,...

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 indicado como **subdir**.

Las opciones anteriores pueden combinarse. Por ejemplo:

ls -cr Muestra el directorio ordenando inversamente por fechas.

El comando **ls** admite los caracteres de sustitución o metacaracteres (*****) y (**?**). El carácter ***** representa cualquier conjunto o secuencia de caracteres. El carácter **?** Representa cualquier carácter, pero sólo uno. Por ejemplo:

ls *.gif Muestra todos los nombres de ficheros que acaben en **.gif**, por ejemplo **dib1.gif**, **a.gif**, etc..

ls file? Muestra todos los ficheros cuyos nombres empiecen por **file** y tengan un nombre de cinco caracteres, por ejemplo: **file1**, **file2**, **filea**, etc.

Comando mkdir (Creación de subdirectorios): este comando (**make directory**) permite a cada usuario crear un nuevo subdirectorio, y se emplea de la siguiente forma:

mkdir subdir, donde **subdir** es el nombre del directorio que se va a crear.

Comando rmdir (Borrado de subdirectorios): este comando borra uno o más directorios del sistema (**remove directory**), siempre que estos subdirectorios estén **vacíos**. Por ejemplo:

rmdir subdir, donde **subdir** es el nombre del directorio que se va a eliminar.

Comando cd: (Cambio de directorio): este comando permite cambiar de directorio a partir del directorio actual de trabajo. Por ejemplo, **cd /home/Pedro** En este ejemplo pasamos del directorio actual de trabajo al nuevo directorio **/home/Pedro**, que será desde ahora nuestro nuevo directorio.

cd directory Nos traslada al subdirectorio **directory** (que deberá existir como subdirectorio en el directorio actual).

cd .. Retrocedemos un nivel en la jerarquía de directorios. Por ejemplo, si estamos en **/home/Pedro** y usamos este comando, pasaremos al escalafón inmediatamente superior de la jerarquía de directorios, en este caso a **/home**. **Nota:** al contrario que en **MS-DOS** en **Linux** no existe la forma **cd..** sin espacio entre **cd** y los dos puntos.

cd Nos sitúa nuevamente en el directorio personal del usuario.

Comando pwd (Situación actual): el comando **pwd (print working directory)** visualiza o imprime la ruta del directorio en el que nos encontramos en este momento. Este comando es uno de los pocos que no tiene opciones y se utiliza escribiendo simplemente **pwd**.

Comando cp (Copia de ficheros): este comando tiene la siguiente forma:

cp file1 file2 y hace una copia de **file1** y le llama **file2**. Si **file2** no existía, lo crea con los mismos atributos de **file1**. Si **file2** existía antes, su contenido queda destruido y es sustituido por el de **file1**. El fichero **file2** estará en el mismo directorio que **file1**. Tanto **file1** como **file2** indican el nombre de un archivo, que puede incluir la ruta al mismo si alguno de ellos no se encuentra en el directorio actual. Otra posibilidad es:

cp file1 file2 namedir que hace copias de **file1** y **file2** en el directorio **namedir**.

cp -i: pregunta si deseas sobrescribir ficheros.

Comando mv (Traslado y cambio de nombre de ficheros): este comando tiene una forma similar al anterior, **mv file1 file2**.

El comando **mv** realiza la misma función que el anterior (**cp**) pero además destruye el fichero original. En definitiva traslada el contenido de **file1** a **file2**; a efectos del usuario lo que ha hecho es cambiar el nombre a **file1**, llamándole **file2**.

De igual forma, **mv file1 file2 namedir** traslada uno o más ficheros (**file1**, **file2**,...) al directorio **namedir** conservándoles el nombre. El comando, **mv namedir1 namedir2** cambia el nombre del

subdirectorio ***namedir1*** por ***namedir2***.

mv -i: con esta variación al mover o renombrar algún fichero el comando pide una confirmación si ya hay un fichero destino.

Hay que recalcar que el comando **mv** sirve así mismo para cambiar el nombre de los ficheros.

Comando rm (Borrado de ficheros): este comando tiene la forma siguiente:

rm file1 file2.

Este comando elimina uno o más ficheros de un directorio en el cual tengamos permiso de escritura. Con este comando resulta facilísimo borrar ficheros inútiles, y desgraciadamente, también los útiles. Por eso es conveniente y casi imprescindible emplear la opción **-i**, de la forma siguiente:

rm -i file1 file2

Con esta opción, **Linux** pedirá confirmación para borrar cada fichero de la lista, sobre si realmente se desea su destrucción o no. Se recomienda usar siempre este comando con esta opción para evitar el borrado de ficheros útiles. Por ejemplo, si se teclea, **rm -i superfluo** aparecerá en pantalla el aviso siguiente:

remove superfluo? y habrá que contestar **y** (yes /sí) o **n** (not / no).

En este comando se pueden utilizar los caracteres de sustitución (***** y **?**), como por ejemplo, **rm fich*** que borraría todos los ficheros del directorio actual que comiencen por **fich**.

El comando **rm *** borrará todos los ficheros del directorio actual, mientras que **rm -i *** realiza una labor análoga, pero con previa confirmación.

rm -i: pide la confirmación de borrado.

rm -r: borra los árboles / estructura de directorios.

Comando file (Características de un fichero): este comando realiza una serie de comprobaciones en un fichero para tratar de clasificarlo. Su formato es: **file fich**. Tras su ejecución este comando muestra el tipo del fichero e información al respecto del mismo.

Exit: Termina la sesión de Linux en una terminal

Comandos de impresión.

Comando lpr

El comando **lpr** se emplea para imprimir una serie de ficheros. Si se emplea sin argumentos imprime el texto que se introduzca a continuación en la impresora por defecto. Por el contrario,

lpr nombre_fichero imprime en la impresora por defecto el fichero indicado.