

Comandos Básicos

Claudio Jiménez Castillo

Grupo de Usuarios de Linux
Universidad Carlos III de Madrid.



16 de Noviembre de 2005

Contenido del curso

- ▶ Inicio y fin de sesión.
- ▶ Obtención de ayuda.
- ▶ Comandos Básicos.
- ▶ Editores de texto.
- ▶ Utilidades de Linux.
- ▶ Referencias.
- ▶ Dudas.

Inicio y fin de sesión

```
login: usuario
```

```
passwd: *****
```

(los asteriscos no aparecen en la pantalla)

- ▶ Recuerde que Linux distingue entre mayúsculas y minúsculas.
- ▶ Para cerrar sesión teclee: exit, logout o ^D (Ctrl+D)
- ▶ Para entrar a la sesión de otro usuario sin salir de la nuestra, teclee:

```
su <usuario>
```

(Si no se especifica usuario el sistema sobreentiende que se

Obtención de ayuda

- ▶ `man <comando>`: nos muestra las páginas de ayuda de ese comando.
- ▶ `info <comando>`: nos muestra las páginas de información de ese comando.
- ▶ `apropos <palabra>`: devuelve la página del manual en la que aparezca dicha palabra.
- ▶ `help` : muestra los comandos de una shell o cshell.
- ▶ `<comando> -h` o `<comando> --help`: muestra una pequeña ayuda del comando.

Comandos básicos

- ▶ Estructura de directorios.
- ▶ Directorios.
- ▶ Listar ficheros.
- ▶ Contenido de los ficheros.
- ▶ Crear y borrar ficheros.
- ▶ Mover y copiar ficheros
- ▶ Enlaces simbólicos.
- ▶ Comodines.
- ▶ Redirección de la Entrada y la Salida.
- ▶ Permisos.
- ▶ Procesos.

Estructura de directorios

Linux tiene un sistema de directorios jerárquico, empieza en el denominado directorio raíz o /, y a partir de ahí, se subdivide:

- ▶ /etc : Archivos de configuración
- ▶ /home : Directorios "principales" de los usuarios (/home/usuario), el root o superusuario esta en /root.
- ▶ /media : Directorio donde se montan los dispositivos (en versiones modernas de Debian), por ejemplo: /media/cdrom, para el cdrom.

En el directorio / existen más directorios, estos son solo unos ejemplos.

Directorios

- ▶ Para ir al directorio home: `cd` o `cd ~`
- ▶ Para ir al directorio home de otro usuario: `cd ~<usuario>`
- ▶ Para ir a un directorio: `cd <directorio>`
- ▶ Para referirse al directorio donde estoy: `./`
- ▶ Para referirse al directorio padre: `../`
- ▶ Para referirse al directorio raiz: `/`
- ▶ Para saber en que directorio estoy: `pwd`

Listar ficheros

- ▶ Para ver los ficheros usaremos el comando `ls`:
 - ▶ `ls <directorio>`: Muestra los ficheros del directorio, si se omite el directorio, se ejecuta sobre el directorio actual.
 - ▶ `ls -a`: Muestra también los ficheros ocultos.
 - ▶ `ls -l`: Muestra información sobre los ficheros (tamaño, permisos,...).
 - ▶ `ls -F`: Identifica tipos de ficheros.
 - ▶ `ls -R`: Muestra los ficheros de forma recursiva.
 - ▶ `ls --help`: Muestra las opciones del comando `ls`
- ▶ `ls` admite varios modificadores juntos.

Contenido de los ficheros

- ▶ `file <fichero>`: Nos indica el tipo de fichero que es.
- ▶ `cat <fichero>`: Nos muestra el contenido de un fichero.
- ▶ `more <fichero>`: Nos muestra el contenido del fichero página a página.
- ▶ `less <fichero>`: Nos muestra el contenido del fichero y nos permite desplazarnos libremente por él.

Crear y borrar ficheros

- ▶ `mkdir <directorio>`: Crea un directorio.
- ▶ `rmdir <directorio>`: Elimina un directorio vacío.
- ▶ `touch <fichero>`: Si no existe lo crea, si existe le establece la fecha actual.
- ▶ `rm <fichero>`: Borra el fichero especificado.
- ▶ `rm -r <directorio>`: Borra el directorio especificado y todo lo que contenga.

Mover y copiar ficheros

- ▶ `mv <fichero> <destino-fichero>`: Mueve el fichero al destino y lo renombra si el nombre del fichero es diferente. Si existe lo sobrescribe.
- ▶ `mv <fichero> <destino-directorio>`: Mueve el fichero al directorio de destino.
- ▶ `cp <fichero> <destino-fichero>`: Copia el fichero al destino y lo renombra si el nombre del fichero es diferente. Si existe lo sobrescribe.
- ▶ `cp <fichero> <destino-directo>`: Copia el fichero al directorio de destino.

Enlaces Simbólicos

Los enlaces simbólicos son un tipo de ficheros que vienen a ser lo que los informáticos conocen como punteros.

Es decir, son ficheros que apuntan a otro, de forma que cuando ejecutamos, modificamos, o leemos un enlace simbólico, en realidad lo que estamos ejecutando, modificando o viendo es otro fichero. Sin embargo si borramos un enlace simbólico, el fichero original sigue donde estaba.

El comando para crear enlaces simbólicos tiene esta sintaxis:

```
ln -s <archivo/directorio origen><enlace destino>
```

Comodines

Para moverse por los archivos y trabajar con ellos tenemos la posibilidad de usar caracteres comodín:

- ▶ *: Se sustituye por cualquier número de caracteres mayor o igual que 0.
Ej: `ls ca*sa` mostraría todos los archivos que empiecen por `ca` acaben en `sa`, `casa` incluido si existiera el archivo.
- ▶ ?: Se sustituye por un caracter cualquiera.
Ej: `ls /etc/???:` mostraría todos los archivos de 3 caracteres en el directorio `/etc/`
- ▶ <tab>: El tabulador (usando `bash`) autocompleta los nombres de archivos, directores e incluso los comandos si se tienen

Redirección de la Entrada y la Salida

Hay varias maneras de redireccionar la entrada/salida muy útiles a la hora de usar Linux:

- ▶ `<comando> > <fichero>`: Redirige la salida del comando a un fichero. Si no existe lo crea y si existe lo sobrescribe.
- ▶ `<comando> >> <fichero>`: Redirige la salida del comando a un fichero. Si no existe lo crea y si existe añade al final del archivo.
- ▶ `<comando> < <fichero>`: Toma como entrada para el comando el contenido del fichero.
- ▶ `<comando> | <comando2>`: Toma como entrada para el comando2 la salida del comando1.

Permisos

- ▶ Los permisos son una de las cosas básicas de seguridad en un sistema tipo UNIX. Los usuarios tienen distintos permisos sobre un fichero para leerlo, modificarlo o ejecutarlo.
- ▶ Es la forma que hay de mantener la privacidad entre los archivos de los distintos usuarios, de permitir a cierto grupo de usuarios modificar un archivo, y de asegurarse de que sólo el administrador puede hacer ciertas cosas, por ejemplo.
- ▶ De esta forma, en el supuesto caso de que un usuario se infectara con un virus, ese virus tan sólo podría estropear lo que ese usuario tenga permisos para estropear, pero de ninguna manera el sistema completo.

Permisos: Leer una línea de permisos

`drwxrw-r--`

(r: leer, w: escribir x: ejecutar, listado si es un directorio)

Esto es una línea de permisos y se divide en 4 segmentos:

1. El primer caracter nos dice que tipo de archivo es, en este caso, un directorio.
2. Los tres caracteres siguientes, nos dicen los permisos del dueño del fichero.
3. Los tres caracteres siguientes, nos dicen los permisos de los usuarios que pertenecen al grupo al que pertenece el fichero.
4. Los tres últimos caracteres nos indican los permisos del resto de usuarios.

Permisos: Modificar permisos(l)

Hay dos maneras de modificar los permisos.

- ▶ Modo 1: `chmod <permisos> <fichero>`

Para calcular <permisos> se suman los valores de cada tipo de permisos (r=4, w=2, x=1).

Ej: 764, esto le da permisos de rwx al dueño, rw al grupo y r al resto de usuarios: `drwxrw-r--`.

- ▶ Modo 2: `chmod <u/g/o>=<+/-><r/w/x> <fichero>`

- ▶ <o/g/u>: u=usuario; g=grupo; o=otros.

- ▶ <+/->: +=añadir; -=quitar.

- ▶ <r/w/x>: r=lectura; w=escritura; x=ejecución.

Ej: `chmod o=-r <fichero>` Quita permisos de lectura al

fichero para usuarios no pertenecientes al grupo ni al dueño

Permisos: Modificar dueño y grupo

- ▶ Dueño: `chown <usuario> <fichero>`
- ▶ Grupo: `chgrp <grupo> <fichero>`
- ▶ Dueño y grupo: `chown <usuario>:<grupo> <fichero>`

Procesos

- ▶ `ps`: Muestra la lista de procesos.
- ▶ `pstree`: Muestra el árbol de procesos.
- ▶ `top`: Aplicación para ver nuestros procesos.
- ▶ `kill`: Elimina procesos.
 - ▶ `kill <pid>`: Envía una señal para terminar el proceso con dicho pid.
 - ▶ `kill -9 <pid>`: Envía una señal para "matar" el proceso con dicho pid.
 - ▶ `killall <proceso>`: Envía una señal para terminar todos los procesos con ese nombre.
- ▶ `xkill`: Es un comando para matar un programa del entorno gráfico al hacer clic sobre él.

Editores de texto

- ▶ **vi**: Es el editor de texto más universal en Unix, esta presente en casi cualquier sistema Unix. Su potencia y pequeño tamaño lo hacen uno de los mejores editores de texto. En Linux también podemos encontrar **vim** (Vi IMproved).
- ▶ **emacs**: No es tan solo un editor, su gran cantidad de extensiones le dan capacidad para hacer muchísimas otras funciones, como lector de correo o news, cliente de irc o incluso tetris.
- ▶ **joe**: Es un editor bastante amigable y fácil de aprender.
- ▶ **nano**: Es un clon GNU del editor pico que puede encontrarse en las distribuciones de Red Hat.

Utilidades de Linux

- ▶ Comunicaciones.
- ▶ Compresión.
- ▶ Varias.
- ▶ Divertidas.
- ▶ Juegos.

Utilidades de Linux: Comunicaciones

- ▶ `telnet`: inicia una sesión remota en la máquina indicada.
- ▶ `ssh`: inicia una sesión remota cifrada en la máquina indicada.
- ▶ `ftp`: inicia una sesión de transferencia de ficheros.
- ▶ `talk`: habilita un terminal para hablar con un usuario.

Utilidades de Linux: Compresión (I)

Hay varias herramientas de compresión para Linux, las de uso más generalizado son `gzip` y `bzip2` que se usan acompañadas del empaquetador `tar`. Otra herramienta de compresión de la que disponemos, aunque su uso en el ámbito de Linux es mucho menor, es `zip`, más asociado con los sistemas operativos de Microsoft. A continuación explicaremos brevemente como funciona cada comando:

- ▶ `tar`: Empaquetador
 - ▶ Empaquetar: `tar -cvf <fichero.tar> <ficheros>`
 - ▶ Desempaquetar: `tar -xvf <fichero.tar>`

Utilidades de Linux: Compresión (II)

- ▶ **gzip: Compresor**
 - ▶ Comprimir: `gzip <fichero>`
 - ▶ Descomprimir: `gzip -d <fichero.gz>`
- ▶ **bzip2: Compresor**
 - ▶ Comprimir: `bzip2 <fichero>`
 - ▶ Descomprimir: `bzip2 -d <fichero.bz2>` o `bunzip2 <fichero.bz2>`
- ▶ **zip: Compresor y empaquetador**
 - ▶ Comprimir: `zip <fichero.zip> <ficheros>`
 - ▶ Descomprimir: `unzip <fichero.zip>`

Utilidades de Linux: Varias

- ▶ `date`: Muestra la fecha actual.
- ▶ `find`: Busca un archivo.
- ▶ `grep`: Busca un patrón en un texto.
- ▶ `uptime`: Muestra el tiempo que lleva nuestra máquina encendida, el número de usuarios logeados actualmente y la carga media del procesador.
- ▶ `w, who`: ver quien ha iniciado sesión en nuestra máquina (física y remotamente).
- ▶ `lynx, links`: navegadores en modo texto.
- ▶ `df`: ver el espacio libre en los discos.

Utilidades de Linux: Divertidas

- ▶ `figlet`: Muestra un texto grande en ascii.
- ▶ `cowsay`: Muestra una vaca diciendo algo en ascii.
- ▶ `bb`: Demostración de la capacidad de las bibliotecas ascii art.
- ▶ `sl`: Muestra un tren de lado a lado de la pantalla.
- ▶ `fortune`: Muestra una frase aleatoria.
- ▶ `apt-get moo`: Muestra una vaca en ascii.

Utilidades de Linux: Juegos

- ▶ overkill: Juego de disparos usando Ascii-art.
- ▶ tuxracer: Juego de Carreras con Tux como protagonista.
- ▶ csmash: Juego de ping-pong en 3D.
- ▶ wesnoth: Juego de estrategia basado en turnos.
- ▶ freeciv: Clon del Civilization.
- ▶ trophy: Juego de carreras de coches en 2D.

Referencias

- ▶ G.L.U.P.: Guía Linux para el usuario.
- ▶ Proyecto Lucas: <http://lucas.hispalinux.org>
- ▶ Web del GUL-uc3m: <http://www.gul.uc3m.es>
- ▶ Lista de Correo de GUL-uc3m: gul@gul.uc3m.es (hay que darse de alta en la web del GUL-uc3m)

Dudas

Agradecimientos

- ▶ A Jesus Espino por pasarme las transparencias en las que me he basado.
- ▶ A la gente del GUL que ha hecho posible estas charlas.
- ▶ A ustedes por venir.