Comandos Básicos

Jesús Espino García Jesús Lanchas Sampablo

jespino@gul.uc3m.es chechu@dei.inf.uc3m.es

Grupo de usuarios de Linux

Universidad Carlos III de Madrid.

Contenido del curso

- Inicio y fin de sesión.
- Obtención de ayuda.
- Estructura de ficheros y directorios.
- Utilidades de Linux.
- Editores de texto.
- Redirigiendo la entrada/salida.
- Referencias.
- Dudas.

Inicio y fin de sesión

```
login: pepe
passwd: ******
```

(los asteriscos no aparecen en la pantalla)

- Linux distingue entre mayúsculas y minúsculas.
- Para cerrar sesión teclee: `exit`, `logout` o `^D` (Ctrl+D)

Obtención de ayuda

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

Estructura de ficheros y directorios

- Directorios.
- Listar ficheros.
- Contenido de los ficheros.
- Crear y borrar ficheros.
- Mover y copiar ficheros.
- Comodines.
- Permisos.

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 `ls`:
  - `ls`: Muestra los ficheros del directorio.
  - `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 pagina a pagina.
- `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, si existe lo sobreescribe.
- `mv <fichero> <destino-directorio>`: Mueve el fichero al directorio de destino.
- `cp <fichero> <destino-fichero>`: Copia el fichero al destino, si existe lo sobreescribe.
- `cp <fichero> <destino-directorio>`: Copia el fichero al directorio de destino.

Comodines

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

- `*`: Se sustituye por cualquier numero 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/`

Permisos

- Leer una linea de permisos.
- Modificar permisos.
- Modificar dueño y grupo.

Permisos: Leer una linea de permisos (I)

`drwxrw-r--`

Esto es una linea de permisos y se divide en 4 segmentos.

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

Permisos: Leer una linea de permisos (II)

- `r`: Permisos de lectura.
- `w`: Permisos de escritura.
- `x`: Permisos de ejecución.

Permisos: Modificar permisos (I)

Hay dos maneras de modificar los permisos. Para ambas, se usa el comando chmod.

- Modo 1: `chmod <permisos> <fichero>`
  Para calcular `<permisos>` se suman los valores de cada tipo de permisos.
  - `r`=4
  - `w`=2
  - `x`=1

  Ej: `drwxrw-r--` = 764
  Ej: `chmod 764 <fichero>` Daría permisos de rwx al dueño, rw al grupo y r al resto de los usuarios.

Permisos: Modificar permisos (II)

- Modo 2:
  `chmod <u/g/o>=<+/-><r/w/x> <fichero>`
  - `<o/g/u>`: u=user; g=group; o=others.
  - `<+/->`: +=añadir; -=quitar.
  - `<r/w/x>`: r=lectura; w=escritura; x=ejecución.

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

Permisos: Modificar dueño y grupo

Se pueden modificar el dueño, el grupo y ambos a la vez

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

Utilidades de Linux

- Comunicaciones.
- Procesos.
- Compresión.
- Varias.
- Divertidas.
- Juegos.

Utilidades de Linux: Comunicaciones

- `telnet`: inicia una sesión remota en la maquina indicada
- `ssh`: inicia una sesión remota cifrada en la maquina indicada
- `ftp`: inicia una sesión de transferencia de ficheros
- `talk`: habilita un terminal para hablar con un usuario
- `who`: muestra los usuarios que están logeados en nuestra maquina

Utilidades de Linux: 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.

Utilidades de Linux: Compresión (I)

Hay varias herramientas de compresión para Linux, las de uso mas 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 el `zip`, mas 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>`
- `zip`: Empaquetador/Compresor:
  - 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 maquina encendida, el numero de usuarios logeados actualmente y la carga media del procesador.

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.

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.
- `realtimebattle`: Juego para programadores.
- `trophy`: Juego de carreras de coches en 2D.

Editores de texto

- `vi`: Es el editor de texto mas universal en Unix, esta presente en casi cualquier sistema Unix. Su potencia y pequeño tamaño lo hacen uno de los mejores editores de texto.
- `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 RedHat.

Redireccionando la entrada/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 sobreescribe.
- `<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.
- `<comando1> | <comando2>`: Toma como entrada para el `comando2` la salida del `comando1`.

Referencias

- G.L.U.P.: Guia Linux para el usuario.
- Proyecto Lucas: http://lucas.hispalinux.org
- Proyecto Insflug: http://www.insflug.org
- Web del Gul uc3m: http://www.gul.uc3m.es
- Lista de Correo del Gul uc3m: gul@gul.uc3m.es

Dudas