

# GNU / Linux



Lo que todo linuxero debe saber

1

## Acciones básicas

- Arrancar el sistema
- Hacer login
  - En el shell (startX)
  - Desde las X: xdm, gdm
- Cambiar entre consolas: [Alt-F1] o [Ctrl-Alt-F1]
  - Consolas o terminales: tty1 (teletypes)
  - También [Alt - ->]
- Hacer logout: [Ctrl-Alt-Supr], logout.
- Apagar el sistema: Shutdown, init0, halt.

2

## Acciones básicas

- Ejecutar un comando:
  - `$ hostname`
  - `$ hostname -opción`
  - `$ hostname --help` (opciones estilo GNU)
- Cambiar la password: `$ passwd`

3

## Acciones básicas

- ¿Quién hay en el sistema?
  - `$ who`: muestra quién hay en el sistema
  - `$ whoami`: muestra quién soy yo en el sistema (username)
  - `$ w`: muestra quién hay en el sistema y qué están haciendo
  - `$ last`: muestra los últimos login de los usuarios (last usuario, para un usuario)

4

## Buscar ayuda

- Páginas man:
  - \$ *man w*
- Documentación en Linux
  - En /usr/docs o en /usr/share/doc
  - Howto's y mini Howto's
  - Proyecto LTDP
  - Proyecto Lucas

5

## Buscar ayuda

- La orden *apropos*: Busca la herramienta o programa que podemos usar para realizar una cierta acción
  - \$ *apropos sort, apropos audio, apropos console*
  - Otra forma es usar las páginas man. Com -k busca en las cabeceras de las páginas de ayuda: *man -k sort*
- ¿Para qué sirve un programa determinado?
  - \$ *whatis who*
  - Opción *help*: \$ *whoami -help*
    - ? (probar también -h o -?)

6

# GNU / Linux



## El shell I (acciones básicas)

7

## El shell

- Interfaz de comandos
- Tipos de shell: bash, csh, ksh, sh.
- Prompt: \$(usuario) , #(root)
- Dejar que el shell complete la orden: *Tab*
  - \$ *ls /mn [Tab]*
- Histórico de órdenes: ?
  - Búsqueda en el histórico de órdenes: [Ctrl+r] y la orden

8

## Teclas especiales en el terminal

- Tab: completa y sugiere un comando incompleto
- Cursores: navegar por el historial
- Shift+Av. página o Re. Página, muestra pantallas en el terminal que no se ven
- Ctrl-l: limpia la pantalla
- Ctrl-r: Busca comandos en el historial

9

## GNU / Linux



Sistemas de ficheros I  
Ordenes básicas de manipulación de ficheros

10

## **Sistema de ficheros**

- Estructura jerárquica del sistema de ficheros en UNIX
- Linux trata todo (directorios, dispositivos, puertos,...) como ficheros
- Tipos de ficheros
  - Ficheros
  - Directorios
  - Enlaces (links)
  - Ficheros de bloque

11

## **Sistema de ficheros**

- Descripción del árbol de directorios
  - /: raíz
  - /home: homedirs de los usuarios
  - /root: homedir de root
  - /mnt: montaje de las particiones
  - /usr: aplicaciones instaladas
  - /bin: ficheros ejecutables (path)
  - /sbin: ficheros ejecutables de administración (path)

12

## Sistema de ficheros

- Descripción del árbol de directorios
  - /etc: ficheros de configuración
  - /dev: dispositivos
  - /boot: arranque del sistema
  - /proc: registro de configuración
  - /tmp: ficheros temporales
  - /var: aloja los registros (logs) del sistema y algunos cachés (proxu squid)

13

## Comandos sobre el árbol de directorios

- *cd* (Change directory)
  - Rutas relativas y absolutas: *tree*
  - *pwd* y directorio home (sólo *cd*)
- *ls* (*-a*, *-l*, *-x*, *--color*)
- *mkdir* (make directory)
  - *-p* crea directorios padre si no existen
- Para crear un archivo debo usar un editor como Vi o Emacs

14

## Comandos sobre el árbol de directorios

- *rm* (-i, -f, -r) y *rmdir*
- *cp* (-r para copiar directorio y todo su contenido)
- *mv* -rf (también renombra)
- Uso de comodines: \* , ? [i,o]

15

## GNU / Linux



Sistemas de ficheros II: acciones avanzadas

16



## Enlaces a ficheros

- Hard links:
  - Mismo sistema de ficheros y mismo inodo
  - Si borro el fichero borro el enlace
- Links simbólicos:
  - Pueden estar en diferentes máquinas incluso
  - Diferente inodo
  - Si borro el enlace no borro el fichero
- Orden *ln* (-s)
- La orden *find* (-exec)

17

## Compartiendo ficheros

- Grupos de usuarios: un usuario pertenece al menos a un grupo (el login group)
  - \$ *groups grupo*
  - \$ *groups usuario*
- Cuando creo un fichero por defecto yo soy el propietario y pertenece a su vez a mi login group
- Determinar el propietario de un fichero: \$ *ls -l*
- Para cambiar el propietario o el grupo
  - \$ *chown usuario fichero*
  - \$ *chgroup grupo fichero*

18

## Permisos sobre ficheros

- Ver los permisos: `$ ls -l`
- Cambiar permisos: `$ chmod`
  - Letras: r,w,x / u, g, o, a
  - Numéricos: 777
- Hacer un fichero privado (eliminando el grupo y otros: `$ chmod go= fichero`
- Hacer un fichero ejecutable: `+x`
- Permisos por defecto: `$ umask 022`
- Bit s y bit g.

19

## Encontrar un fichero

- Para encontrar un fichero que tenga en su nombre la palabra audio:
  - `$ locate audio` o `$ locate *audio`
  - `$ locate /.` encuentra todos los ficheros ocultos del sistema
  - NOTA: no distingue mayúsculas

20

## Encontrar un fichero

- La orden *find*
  - *\$ find /home -name Desktop*
  - *\$ find /home -iname desktop* (no distingue mayúsculas)
  - *\$ find /home -name 'D\*'*
  - *\$ find /home -name '\*.tex'*

21

## Encontrar un fichero

- Otros ejemplos:
  - *\$ find ~ -size +100k* (ficheros mayores de 100k)
  - *\$ find ~ -empty* (ficheros vacíos)
  - *# find ~ -user root* (ficheros propiedad de root)
- Ejecutar comandos con los ficheros encontrados
  - *\$ find ~ -empty -exec rm -f {} \;*
  - *-ok* es igual pero pide confirmación
- Redireccionar errores:
  - *\$ find / -name 'et\*' >~/file 2>/dev/null*

22

## Encontrar un fichero

- Cómo saber el número de ficheros en un directorio:
  - `$ ls -a |wc -l`
- Cómo saber el número de ficheros en un árbol:
  - N° de ficheros y directorios: `$ find /usr/share | wc -l`
  - N° de ficheros: `$ find /usr/share \! -type d | wc -l` (`\!` -type d para excluir la lista de directorios y no contarlos)
  - N° de directorios: `$ find /usr/share \! -type f | wc -l` (`\!` -type f para excluir la lista de ficheros y no contarlos)

23

## Encontrar un fichero

- Localizar una orden:
  - `$ which cd`
  - Consejo: mirar siempre en `/bin`, `/sbin`, `/usr/bin`, `/usr/sbin`,... en general en el PATH
- Ver el tipo de un fichero
  - `$ type fichero`
  - `$ file fichero`

24

## Filtros

- La orden grep
- La orden sort
- La orden cut

25

## Ver el contenido de un fichero

- *cat*
  - Puede crear ficheros: *cat > file*
  - Puede agregar algo a un fichero sin abrirlo:*cat >>file.existe*
- ? *more*
- *less*
- *echo*

26

## Tratamiento de ficheros

- La orden split: fracciona un archivo en partes.
  - \$ `split b2m prueba.bmp prueba.bmp.` (prueba de 3M)
    - ? Crea `prueba.bmp.aa` (2M) `prueba.bmp.ab` (1M)
  - Para juntarlos:
    - ? `cat prueba.bmp.* > prueba.bmp`
    - ? `Rm prueba.bmp.*`

27

## Comparar ficheros

- La orden cmp: compara dos ficheros y ve si son iguales
  - \$ `cmp file1 file2`
- La orden diff y sdiff también comparan pero dan más información
  - \$ `sdiff file1 file2 | less`

28

## Compresión de ficheros

- La orden `gzip`:
  - `$ gzip file1`, y obtendremos `file1.gz`, borrando `file1`
  - `$ gzip -r directorio` (comprime los archivos de dentro)
  - `$ gunzip file1.gz` para descomprimir (también extensión `z`, `Z`, `zip`,...)
- Otros compresores:
  - `bzip2`
  - `zip`
  - `Compress`

29

## Empaquetar en un fichero

- La orden `tar`:
  - `$ tar -cvf file.tar file`
  - `$ tar -czvf file.tar.gz file` (`tar+gzip`)
  - `$ tar -xzf file.tar.gz` (`descomprime y desempaqueta`)

30

## Sistema de ficheros y particiones

- La orden mount / umount
  - \$ *mount -t fat /dev/hda1 /mnt/windows*
  - \$ *umount /dev/cdrom*
- El fichero /etc/fstab
  - Estructura
  - Opciones:
    - ? User
    - ? Auto
    - ? ro

31

## Sistema de ficheros y particiones

- La orden df: informa del tamaño de los sistemas de ficheros y su uso
  - \$ *df -h* (h, human readable)
- La orden du: muestra lo que ocupa cada fichero del directorio donde me encuentre
  - \$ *du -hm* (megas) o *-k(kb)*
  - \$ *du -sm /usr/local* (muestra un resumen, sólo la última línea, es decir, lo que ocupa todo el directorio).

32



## Disquetes

- Formatear un disquete
  - `$ mke2fs /dev/fd0`
- Formatear con formato fat:
  - `$ umount /floppy; mkfs -t fat /dev/fd0`
  - `$ mformat /dev/fd0`

33

## GNU / Linux



El shell II (acciones avanzadas)

34

## El shell

- Ejecutar una lista de comandos
  - \$ *clear ; ls*
- Tuberías (pipes)
  - \$ *who | w -l*
  - \$ *man | less*

35

## El shell

- Redireccionamiento de entrada y de salida
  - Entradas y salidas estándar y otras entradas y salidas
- Redireccionar la entrada a un fichero
  - \$ *apropos < Keywords*, donde *Keywords* es un fichero con palabras clave (órdenes)
- Redireccionar la salida a un fichero
  - \$ *ls / > ~/filedir*
- Anexar la salida a un fichero existente
  - \$ *ls /etc >> ~/filedir*
- Redireccionar errores: *2>/dev/null*

36

## El shell

- Redireccionar los mensajes de error a un fichero
  - `$ ls / 2> ~file.error`
- Redireccionar la salida de un comando y tomarlo como entrada de otro es al fin y al cabo una tubería

37

## Histórico de órdenes

- Para ver el histórico de órdenes:
  - `$ history`
  - `$ history | grep apropos`
  - `$ !3`, ejecuta la 3ª orden de éste histórico
- Guardado en `~/.bash_history`

38

## Personalizar el shell

- Cambiar el prompt del shell:
  - `$ PS1=' tus deseos son órdenes: '`, con comillas porque contiene espacios
  - `\w` indica el directorio actual
  - `\u` el usuario actual
  - `\h` el nombre el equipo
  - `$ echo $PS1`

39

## Personalizar el shell

- Crear alias
  - `$ alias bye="exit"`
  - `$ alias ap="apropos shell bash"`
- Variable PATH:
  - `$ echo $PATH`
- Se guardan en:
  - `~/.bashrc` (o `/etc/bashrc`) para todos los shell excepto el login shell
  - `~/.bash_profile` para el login shell

40

## Personalizar el shell

- Si los quiero unificar poner esta línea en el `~/.bash_profile`
  - `If [ -f ~/.bashrc ]; then . ~/.bashrc; fi`
- El archivo `/etc/profile`: cuando hago login cargo todo lo que esté en este fichero. Es válido para cualquier usuario del sistema, mientras que `.bash_profile` es distinto para cada usuario.

41

## Variables del shell

- Puedo definir variables para cualquier shell que arranque (ejecutando `bash`). Para ello debo exportarla.
  - `$ export` (vería todas las variables exportadas)
- Exportar variables:
  - `$ BAR=bar; FOO=foo`
  - `$ export BAR`
  - `$ echo $BAR $FOO`
  - `$ bash; $ echo $BAR $FOO`

42

## Variables del shell

- En teoría debería poner *set FOO=foo (unset)*
- Otra forma sería declararlas y exportarlas en el *.bash\_profile*, y estarían dispuestas para un login posterior.
- Puedo crear a su vez alias de órdenes:
  - *\$ alias 'lh = ls -lh'*
  - *\$ alias*
  - *\$ unalias lh*