

UD3 COMANDOS BÁSICOS

Introducción al BASH

Es el programa que recibe la orden, normalmente introducida por el usuario y la convierte en instrucciones para el S.O.

```
// BUSCAR RUTA REAL
```

La ventana es un intérprete de comandos, en este caso, del tipo bash.

Al abrir el bash, aparece en la primera línea el llamado “prompt” (indicador de mandatos) que consta del nombre del usuario, el nombre del equipo y la localización actual en el árbol de directorios (path). El prompt puede configurarse diferente.

Detrás del prompt se introducen los comandos.

Ficheros y directorios

Para trabajar eficientemente con la shell hay que conocer la estructura de directorios y ficheros en Linux.

Revisar los directorios más importantes nombrados en la UD2.

Comandos

Los comandos están formados por varios elementos:

1. la palabra de comando
2. los argumentos, que pueden ser parámetros u opciones.

Los comandos se ejecutan solo cuando presiona Intro. Hasta entonces se puede editar la línea de comandos sin problemas.

Las opciones van precedidas de un **guion**. (Ej: ls -l)

Si un comando ocupa varias líneas se separa cada línea con “\”

Para ejecutar varios comandos en una línea se usa el “;”

Con el comando exit salimos de la shell.

Comandos de manipulación de archivos y directorios.

CARACTERES COMODÍN

- *** → sustituye a una cantidad de letras
- ?** → sustituye a una letra cualquiera
- []** → una palabra con alguna de las letras del grupo
- [!]** → no tengan las letras del grupo

INFORMACIÓN:

whatis comando → pequeña descripción de un comando y los man que podemos consultar. Ej: `whatis chmod`

man comando → muestra el manual de esa orden Ej: `man ls` (para salir q)

info → muestra todos los manuales, con intro vamos entrando.

comando --help → ayuda

ls → lista el contenido de un directorio

ls [argumentos][directorio]

-l → lista detallada

-a → para ver los ficheros ocultos

-F → diferencia entre ficheros y directorios (/)

Ejemplo: `ls -lF /bin` `ls ..` `ls b[ion]*` `ls b[i-o]`

pwd → muestra la ruta del directorio actual

cd [opcion(es)] directorio → cambia al directorio que se marca. Si solo se escribe `cd` se desplaza al directorio home.

file → muestra el tipo de archivo: ascii, directorio...

mkdir o md → crea un nuevo directorio

rmdir o rd → borra directorios si están vacíos. Opción R: recursivo.

cp [opcion(es)] fichero_original fichero_copia → Crea una copia de `fichero_original` en `fichero_copia`

i: espera una confirmación antes de sobrescribir un `fichero_copia` existente

r: copia recursiva incluyendo subdirectorios

mv [opcion(es)] fichero_original fichero_destino → Hace una copia de archivo origen en `fichero_destino` y a continuación borra el fichero original.

b: crea una copia de seguridad antes de mover el fichero al destino.

i: espera, dado el caso, una confirmación antes de sobrescribir un `fichero_destino` existente.

rm [opcion(es)] fichero(s) → borra el fichero(s) indicados.

i: solicita información para borrar cada archivo

r: borra de forma recursiva, incluidos subdirectorios.

touch → crea un archivo de texto.

COMANDOS DE VISUALIZACIÓN Y EDITORES

cat [opcion(es)] fichero(s) → Permite ver el contenido de 1 o más ficheros.

n: numera las líneas. Ej: cat -n ~/hola

less [opcion(es)] fichero(s) → visualiza uno o varios ficheros página a página. Instrucciones:

flechas → moverse por el fichero

intro → moverse línea a línea.

espacio → moverse página a página

q → salir

head [opcion(es)] fichero(s) → Muestra las primeras líneas de un fichero.

head **-N** fichero → muestra N líneas, sino 10

tail [opcion(es)] fichero(s) → Muestra el final de un fichero

tail **-N** fichero → muestra las N últimas líneas del fichero, si no se marca número, 10.

tail **+N** fichero → muestra desde la línea N hasta el final del fichero.

EDITOR VI:

Hay muchos editores, pero se encuentra en todas las distribuciones.

El editor vi tiene dos modos de funcionamiento e inicialmente se encuentra en modo comando:

Modo edición: “todo lo que escribo aparece en pantalla”. Se accede con alguna de las siguientes teclas: i, I, a, A, o, O

Modo comando: Se accede pulsando la tecla ESC y las teclas tienen un significado especial:

- **\$**: mueve el cursor hasta el final de la línea
- **j**: mueve una línea hacia abajo
- **:w** : graba el fichero
- **:q**: salir sin guardar
- **:wq** : salir y guardar
- **: !q**: forzar a salir sin guardar

Crear o editar un archivo: vi archivo texto Si el archivo existe lo abre, sino lo crea.

Agregar texto: Se puede añadir texto en diferentes partes del fichero, eliminarlo o cambiarlo.

Para entrar en el modo edición se puede hacer escribiendo:

- **a**: se agrega texto después del carácter donde esta
- **i**: se agrega texto antes del carácter donde está.

Borrar texto: con los siguientes comandos,

- **x**: se borra el carácter donde está el cursor
- **dd**: elimina la línea completa

Modificar línea:

- **cc**: permite modificar una línea completa.

BUSCAR

grep [opcion(es)] palabra_o_texto fichero(s) → Busca una palabra en 1 o más ficheros.

Si se encuentra muestra la línea donde está y el nombre del fichero.

Permite usar caracteres comodines.

Para buscar más de una palabra debe ir entre "".

i: ignora la distinción entre mayúsculas y minúsculas.

n: numera las líneas donde encontró la(s) palabra(s).

l: da el nombre de los ficheros donde encontró, pero no muestra las líneas.

r: recursiva, para buscar en todos los subdirectorios.

Ej:

grep pepe * → busca las líneas que contienen la palabra pepe de entre todos los ficheros de este directorio.

grep -r 7 * → busca las líneas donde aparece el número 7 de entre todos los ficheros de este directorio y de todos sus subdirectorios.

whereis comando → Muestra la ruta del ejecutable del comando, de su código fuente y del archivo que contiene su manual o man.

Ej:

whereis ls → devuelve ls: /bin/ls /usr/share/man/man1/ls.1.gz

find directorio_inicial -name nombre_archivo → Busca archivos a partir del directorio inicial cuyo nombre responda a nombre_archivo

Ej:

find . -name pe* → busca a partir del directorio actual todos los archivos que comiencen por pe

find / -name pe* → busca en el directorio raíz todos los archivos que comiencen por pe

diff [opciones] fichero1 fichero2 → Compara el contenido de 2 ficheros, mostrando las líneas distintas.

-q: indica sí existen diferencias, pero no muestra cada una de las líneas.

Ej: diff hola adios → compara el contenido de los dos ficheros.

Comandos de información del sistema

df [opciones] [directorio] → Permite ver el espacio ocupado y libre de un directorio.

H: muestra los tamaños en GB, MB, ...

T: muestra el tipo de formato de la unidad

Ej: `df -TH /`

du → Indica el tamaño de todos los subdirectorios de donde nos encontramos

h: muestra el tamaño en K, M, ...

a: muestra el tamaño de cada uno de los archivos que lo componen

free: muestra la cantidad de RAM libre y ocupada, así como la caché y la swap.

date: muestra la hora y fecha del sistema (se puede cambiar desde aquí)

calc: mostrar un mes en pantalla.

Ej: `calc mes año:` muestra ese mes de ese año.

Redireccionamientos

Agrupamiento de órdenes: podemos ejecutar varias órdenes en la misma línea, una detrás de otra, separándolas con `;`

Redireccionamiento de salida: la salida por defecto de las órdenes es la pantalla.

Para redireccionar la salida se usa el símbolo `>` y `>>`

Redireccionamiento de entrada: la entrada por defecto es el teclado.

Para redireccionar la entrada se usa el símbolo `<`.

Tuberías: Une la salida de una orden con la entrada de otra.

Se usa el símbolo `|` (alt gr + 1)

Alias y enlaces

ALIAS

Se usan para simplificar las órdenes.

alias → muestra todos los alias cargados actualmente.

Para crear un nuevo alias: **alias nombre="comando"** (este alias solo se crea para la **sesión actual**)

Para borrar un nuevo alias: **unalias nombre** (solo para la **sesión actual**)

ENLACES

Linux asocia a cada archivo un número, con el cual lo identifica internamente. Este número es el **i-nodo**.

ls -i → se ven los i-nodos

TIPOS DE ENLACES

- **Duros:** Se crea un nombre en un directorio que físicamente está en el mismo sitio que otro. Cualquier modificación utilizando uno de los nombres se realiza en la misma zona de memoria física. Ambos ficheros tendrán el mismo i-nodo.

Si se borra alguno de los nombres enlazados al fichero, sigue existiendo mientras exista uno de los nombres.

No se pueden hacer enlaces duros de un directorio.

Crear un enlace duro entre fichero1 y fichero2 (ambos tendrán el mismo número de i-nodo) → **ln fich1 fich2**

Para ver si un fichero tiene enlaces hay que mirar la columna links del comando ls en formato largo.

Además, con **ls -i**, que se ven los i-nodos se puede comprobar que los ficheros enlazados tienen el mismo.

- **Simbólicos:** Son un tipo de archivos especiales que representan a otro archivo. El contenido de un enlace simbólico es la ruta correspondiente al archivo o directorio con el que está enlazado. Se puede crear un enlace simbólico, aunque el archivo que representa no exista.

ln -s nombre_archivo nombre_enlace

Nombre_archivo, es la ruta correspondiente al archivo o directorio que va a ser representado en el enlace.

Debe ser una trayectoria absoluta.

Los enlaces simbólicos se pueden diferenciar con el comando **ls -l** ya que se produce una salida parecida a esta:

```
lrwxrwxrwx 1 cla01 group 7Mar 24 10:44 motd ->/etc/motd
```

donde el tipo de archivo l indica enlace simbólico y la trayectoria que sigue al nombre del archivo es la del archivo con que esta enlazado.

Se puede utilizar **ls -L** para ver el fichero o directorio enlazado y sus propiedades.

Variables de entorno

Las variables de entorno son cadenas de caracteres de la forma NOMBRE=valor, donde NOMBRE puede ser cualquier cadena de caracteres que **no incluya el carácter \$ y no tenga espacios en blanco** y valor, puede ser cualquier cadena de caracteres incluyendo espacios.

Existen variables definidas en el sistema, y pueden verse con el comando set

Cuando se va a utilizar el valor contenido en una variable o valor de la variable de entorno:

\$nombre_variable

Control de tareas y procesos

Linux es un sistema multitarea y multiusuario, por lo que es capaz de atender varios programas de diferentes usuarios a la vez.

Un proceso o tarea es una petición al sistema de un programa en ejecución.

Linux asigna a cada proceso un número de identificación llamado **PID**.

Hay algunos procesos característicos:

- La shell de presentación de un usuario es un proceso que está en ejecución mientras el usuario no termine la sesión de trabajo.
- Los demonios son procesos del sistema que actúan sin que el usuario lo solicite. Por ejemplo: el demonio que atiende las peticiones de login desde un terminal es getty, el demonio encargado del sistema de impresión es lpsched.

La variable de entorno \$ contiene el PID del proceso en ejecución.

Ej: echo "El \$\$ es el shell" → El 114 es el Shell

ps → Se usa para obtener información sobre el estado de los procesos que están activos e iniciados por uno mismo. (con ps aux muestra todos los procesos sin tener en cuenta el usuario)

Algunos de los indicadores que muestra el comando son:

STAT: estado del proceso (0 – en marcha, S – dormido, R –ejecutándose, T – parado , ...)

UID: identificación del usuario propietario PID, PPID (PID del padre),

C: estimación del uso de CPU

TTY: terminal que controla el proceso

EJECUCIÓN EN PRIMER PLANO: Cada comando que se ejecuta es considerado por el sistema como un programa independiente o proceso. Es necesario que el comando que se está ejecutando en primer plano finalice para poder ejecutar otro en primer plano.

Por defecto los procesos se ejecutan en primer plano.

EJECUCIÓN EN SEGUNDO PLANO: No es necesario esperar a que termine para ejecutar otro comando. No veo como se ejecuta y hay que tener cuidado si la salida es por pantalla.

Para que un comando se ejecute en segundo plano hay que poner **&** al final de la orden:

Ej: `ls /etc > ~/listado.txt &`

jobs: permite ver todos los procesos o tareas que se están ejecutando. (nº tarea (%) = PID)

Ctrl + C: termina las tareas en primer plano

Ctrl + Z: pausa las tareas en primer plano

fg % nºtarea: reanuda la tarea en primer plano

bg % nºtarea: reanuda la tarea en segundo plano.

kill [opcion] PIDs → mata el proceso que se le indique. Si no deja kill -9 PID

killall nombre → igual que kill pero es necesario el nombre del proceso en vez del PID.