103 - Comandos GNU y Unix

1 - Trabajar en linea de comandos

Es un prompt del shell de usuario para introducir instrucciones.

Si el prompt finaliza en $ significa que es un usuario común, si termina en # es root.

Shell Bash

Es el estándar de la mayoría de las distros Linux.

Algunos comando importantes podrían ser (los que piden en el temario):

Comando	Finalidad	Ejemplo
alias	Crea un alias para otro comando	alias rm='rm -i'
exec	Un comando invocato con exec, sustituye la sesión actual de shell	exec teleinit 1
echo	imprime un texto o variable	echo $PATH
env	Sin argumentos muestras las variables de entorno y contenidos	env DISPLAY =:1.0 xclock
export	Define una variable de entorno para la sesión	export PATH=$PATH:/usr/....
pwd	Muestra el directorio actual	pwd o echo $PWD
set	Muestra y define los valores de las variables de entorno Linux	set | NOMBRE='Guillem'
unset	elimina una variable en la sesión	unset NOMBRE
type	Muestra que tipo de comando es	type cd or type ls
which	Muestra el path dónde está instalado el paquete ejecutado	which firefox
man	Lanza los manuales del sistema de cada comando	man ls
uname	Muestra información básica del sistema, la mas usada -a	uname -a
history	Lanza el listado de comandos ejecutados en el sistema, se almacena en .bash_history	history

Variables

No deben tener espacios y para crearlas es tan sencillo como:

```
nombre_variable=valor
```

```
# Para mostrar el valor usaremos el $
echo $nombre_variable
```

Hay dos tipos:

- Locales: sólo accesibles para la sesión actual
- Exportadas: accesibles en la actual y la siguiente sesión, definidas en bashrc o profile de usuario o de sistema

Variables predefinidas:

Variable	Definición
DISPLAY	Determina en que display X el programa debe mostrar sus ventanas
HISTFILE	Ruta al historial de comandos de usuario ($HOME/.bash_history)
HOME	Ruta del directorio personal del usuario
LOGNAME	Nombre que el usuario usó para entrar al sistema
PATH	Lista de directorios en los cuales se buscarán los programas
PWD	El directorio actual
SHELL	El shell usado
TERM	Tipo de emulador, es distinto si se usa en X que en consola TTY
$!	PID del último proceso ejecutado
$$	PID de la shell actual
$?	Devuelve 0 si el comando tiene éxito, si no 1 (ls | echo $?)
~	Directorio personal del usuario actual
~ziko	Directorio personal del usuario ziko

Comandos secuenciales

Para ejecutar comandos uno tras otro usamos:

```
comando1 ; comando2 ; comando3
```

Para ejecutar varios comandos si el anterior fue correcto:

```
comando1 && comando2 && comando3
```

Para ejecutar comandos si el anterior fue fallido

```
comando1 || comando2 || comando3
```

Referencias y manuales

Tras escribir las primeras letras de un comando o directorio, con la tecla tabulador podremos completar la palabra o el path.

Para consultar el manual de cada comando usaremos el comando man, estos manuales tienen la siguiente organización:

- Nombre --> asunto de la página del manual y breve descripción
- Sinopsis --> la sintaxis del comando
- Opciones --> Revisión de todas las opciones y sus funciones
- Archivos --> Archivos relacionados con el asunto
- Vea también --> otras páginas relacionadas

Por defecto los manuales están guardados en los path:

- /usr/man
- /usr/share/man

Podemos añadir mas rutas configurando /usr/lib/man.conf o /etc/man.conf

Impresión de manuales

```
# Imprimir como texto sin formato redirigiendo la salida de man con el comando groff
zcat /usr/man/man1/find.1.gz | groff -man -Tps > find.ps

# Para imprimir directamente
zcat /usr/man/man1/find.1.gz | groff -man -Tps | lprint
```

2 - Procesar cadenas de texto por medio de filtros

Comando	Descripción
cat	Se usa para mostrar el contenido de archivos y puede actuar como redireccionador
cut	delimita un archivo en columnas, para separar por campo usamos -d carácter delimitador -f informa de la posición del campo. Ej: cut -d':'1,3 /etc/passwd
uniq	muestra el contenido de archivos quitando lineas repetidas, con -u muestra sólo las líneas que no se repiten
tac	igual que cat, pero muestra el contenido de atrás adelante
head	muestra el inicio de los archivos, por defecto 10 primeras líneas pero con la opción -n podemos indicar cuantas líneas y con -c indicamos el número de caracteres que se mostrarán
tail	igual que head pero muestra el final, tiene las mismas opciones, pero añade la opción -f para que muestre continuamente el final cuando se agrega texto (útil para logs por ejemplo)

Comando	Descripción
wc	cuenta lineas (-l), palabra (-w) o caracteres (-c), sin argumentos muestra las 3 opciones
nl	número de líneas como el comando cat -b. Con el argumento -ba se numeran todas las líneas. Con -bt se numeran las líneas que no están en blanco
expand	sustituye dos o mas espacios simples por una tabulación
hexdump	muestra archivos binarios, con -c es mas legible
od	se usa para convertir entre diferentes formatos, hexadecimal, binario, etc
split	divide un archivo en otros menores, con -l se indica el número de líneas de cada parte, con -b el tamaño de cada parte. Ej: split -b 1024k archivo parte *y creará parte_aa, parte_ab, ... para volver a concatenarlo cat parte** > archivo_entero
paste	concatena archivos lado a lado en forma de columnas. Ej paste texto1 texto2 y mostraría linea 1 del 1 linea 1 del 2.
join	como el paste pero especificando los campos. Ej join -1 CAMPO -2 CAMPO archivo 1 archivo 2.
sort	ordena alfabéticamente, con -n lo hace numéricamente y con -r invierte el resultado
fmt	configura un texto para determinado número de caracteres por linea, el estándar es 75. Con -w indicamos en número de caracteres por línea, con -s divide lineas grandes y -u un espacio entre palabras y dos entre sentencias
pr	divide el archivo para la impresión, el estándar es 66 líneas (-l) de 72 carácteres (-w)
tr	convierte caracteres. ej echo abc | tr '[a-z]''[A-Z]', podemos sustituir espacios por otro carácter con echo 'Frase con espacios' | tr ' '_'
zcat	descomprime ficheros a una salida estándar
xzcat	comprime y descomprime ficheros en formato xz
bzcat	comprime y descomprime fichero en formato bzip2
md5sum	Imprime o chequea MD5 (128 bit) checksums
sha256sum	Imprime o chequea 256 (256 bit) checksums
sha512sum	Imprime o chequea 512 (512 bit) checksums

3 - Administración básica de archivos

Directorios y archivos

Pueden accederse tanto por su camino absoluto como relativo.

El punto es el directorio actual y los dos puntos el directorio superior.

El comando ls lo usamos para listar archivos y contenido de un directorio.

Lo que muestra el comando ls -l de izquierda a derecha:

- Tipo de archivo y permiso
- Número de hardlinks para el archivo.
- Dueño
- Grupo al que pertenece
- Tamaño en bytes
- Fecha de la última modificación
- Hora de la última modificación
- Nombre del archivo

Con el comando file podemos saber que tipo de archivo es.

Con el comando stat podemos ver todos los detalles de acceso y modificación de un fichero

Manipulación de archivos y directorios

El comando cp se usa para copiar archivos y como argumentos tiene:

- -i: modo interactivo, va preguntando
- -p: copia también los atributos del archivo original
- -r: copia recursivamente el contenido del directorio origen

Con el comando mv, movemos los ficheros, con -i irá preguntando.

Touch nos permite modificar la fecha de creación de un archivo a los valores actuales del sistema. Para sólo modificar la fecha, opción -m, para la hora -a y otros valores -t.

Navegamos por el sistema con el comando cd

mkdir nos permite crear una carpeta. La opción -p nos creará toda la ruta de directorios.

rmdir borrará un directorio, pero debe estar vacío

rm borra fichero y con la opción -rf borrará ficheros y directorios recursivamente, con lo que se vuelve mejor opción para borrar directorios que el anterior rmdir.

Empaquetar archivos

tar y cpio para agrupar en un archivo a varios, opciones básicas:

- c --> crea
- v --> muestra cada archivo
- f --> especifica el path
- x --> descomprime

```
# Comprimir una carpeta:
tar -cvf nombre_agrupado /directorio
```

```
# Descomprimir el archivo comprimido:
tar -xvf nombre_agrupado
```

gzip y bzip2, usaremos gzip/bzip2 archivo para comprimir, y creará un fichero .gz o .bz2, la diferencia radica en que gzip es más rápido y bzip comprime mas.

Podemos comprimir directamente al pasar a tar con: tar czvf nombre_paquete.gz /directorio, de esta manera usaremos gzip. Para usar bzip cambiaremos las opciones por cjvf.

Para descomprimir usamos gunzip y bunzip2 y también puede redireccionarse con el comando tar con las opciones z y j.

Carácteres comodín (file globbing)

- * : sustituye a cualquier secuencia de carácteres.
- ? : sustituye un carácter
- [] : indica una lista de caracteres, por ejemplo [abc]
- {} : indica una lista de términos sperados por una coma, ej. ls /dev/{hda, fd0}
- ! : antes de un carácter comodín lo elimina de la operación.
- \ : antes de cualquier comodín no realizan sustitución.

Búsqueda de archivos

El comando principal es find directorio criterio, el directorio es donde se debe iniciar la búsqueda, se hará recursivamente y el criterio es la regla de búsqueda a realizar, defino los criterios en una tabla:

Criterio	Definición
-type	Define el tipo: archivo (f), directorio (d) o enlace (l)
-name nombre	Nombre del archivo
-user usuario	Dueño del archivo
-size -/+n	Archivos con tamaño superior o inferior a n
-atime -/+n	Accedido antes o después de n (n*24 horas)
-ctime -/+n	Archivo creado antes o después de n
-mtime -/+n	Archivo modificado antes o después de n
-amin -/+n	Archivo accedido antes o después de n (n es cantidad de minutos)
-cmin -/+n	Archivo creado antes o después de n
-mmin -/+n	Archivo modificado antes o después de n
-newer archivo	El archivo buscado se creó o modificó después de archivo.
-perm modo	El archivo buscado tiene permiso igual a modo (r,w,l)
-perm -modo	El archivo buscado tiene todos los permisos de modo
-perm +modo	El archivo buscado tiene cualquiera de los permisos de modo

Criterio	Definición
-exec	permiten pasar un comando a ejecutar en cada una de las lineas sin preguntar
-ok	idem que exec, pero pregunta

```
# Todos los enlaces /usr/lib creados hace menos de 24 horas
find /usr/lib -type l -ctime -1

# Archivos con permiso setuid
find / -type f -perm -4000

# Directorios con permiso setguid
find / -type d -perm -2000

# Directorios con permiso sticky bit
find / -type d -perm -1000

# Buscar archivos con un tamaño superrior a 100M
find / -type f -size +100M

# Buscar archivos con un tamaño inferior a 1M
find / -type f -size -1M

# Buscar archivos modificados en los últimos 3 días
find / -type f -mtime -3

# Buscar ficheros con creación o modificación inferior a 5 minutos
find / -mmin -5

# Busca y elimina todos los ficheros avi
find / -type -f -name "*.avi" -exec rm -rf "{}" \;
```

4 - Flujos pipes (tuberías) y redireccionamientos de salida

- Descriptor entrada --> STDIN (0) --> suele ser el teclado u otro periférico de entrada
- Descriptor salida --> STDOUT (1) --> finalización OK
- Descriptor de errores --> STDERR (2) --> finalización fallida

Redireccionamiento

- > --> redirecciona a una salida, creando un fichero nuevo
- >> --> Redirecciona a una salida, añadiendo al final del fichero
- < --> Redireccionamos un fichero a la entrada estándar.
- 2> --> Redirecciona los errores (stderr)
- &> --> Redireccciona los errores y la salida estándar (stderr + stdout)

Tubería (pipe)

Con el símbolo | envíamos la salida de un comando a la entrada del siguiente.

Sustitución de comandos

Es posible usar la salida de un comando como argumento para otro usando las comillas invertidas ``

```
ls -l `cat /etc/ld.so.con`
```

Con el comando xargs pasamos los datos que recibe de stdin como argumento para otro comando, este funciona de intermediario.

5 - Crear, monitorear y finalizar procesos

Un proceso es un programa en ejecución, cada proceso tiene un PID (número único de identificación)

Monitorear procesos

Tenemos diferentes comandos para poder verlos:

- ps: muestra los procesos activos de manera detallada
  - ps -ef
  - ps aux
  - ps -eo user, pid, ... (escoje columnas)
  - ps -u apache, root (para realizar filtros)
- top: monitorea continuamente los procesos, uso de memoria y CPU
- pstree: muestra procesos activos en árbolps genealógico (en desuso)
- kill PID: envía la señal SIGTERM con valor 15, que pide la finalización del proceso (kill -9 PID, se carga el proceso sin solicitar, lo mata)
- killall: funciona igual que kill pero usa el nombre del proceso en lugar del PID

Tareas en primer y segundo plano

Podemos iniciar una tarea en segundo plano añadiendo & al final de la linea del comando.

Con el comando jobs podemos ver todos los procesos que tenemos actualmente en segundo plano.

```
sleep 100 &
sleep 300 &
jobs
>> [1]-  Running                 sleep 100 &
>> [2]+  Running                 sleep 300 &
```

Para detenerlos tenemos dos opciones:

- Matar el proceso con kill (es un proceso normal con su PID)

- Devolverlo a primer plano y detenerlo.

Para devolver a primer plano usaremos el comando fg (nº de job)

Y para enviar un proceso a segundo plano de nuevo bg (nº de job)

6 - Modificar la prioridad de ejecución de un proceso

Como todos los SOs multitarea se pueden atribuir prioridades a los procesos, estas prioridades se definen con números denominados NI (números nice), usados para modificar la prioridad de CPU y balancear su uso. Por defecto se inicia con prioridad 0.

Los NI van desde la mas baja (19) a la mas alta prioridad (-20), pero sólo root puede cambiar a prioridades por debajo de 0.

Para iniciar un comando con prioridad cambiada usamos nice.

Para modificar la prioridad de un proceso ya arrancado, usaremos renice.

Con renice también podemos modificar todos los procesos del grupo o usuario.

```
nice -n [-20 a 19] comando
renice -n [-20 a 19] -p PID
renice [-20 a +19] -g/u nombre_grupo/usuario
```

7 - Buscar en archivos de texto usando expresiones regulares

Muchos programas soportan el uso de expresiones regulares.

Carácter	Finalidad
^	Inicio de línea
$	Final de línea
.	Cualquier carácter
*****	Cualquier secuencia de 0 o más caracteres
<	Inicio de palabra
>	Fin de palabra
b	Límite de palabra
****	Evita el significado del sistema
{4}	4 veces
{4,}	4 veces o mas

Carácter	Finalidad
{4,10}	Entre 4 y 10
[ ]	Conjunto de caracteres
[^]	Niega el conjunto de caracteres

GREP

Opciones mas importantes de grep

Modificador	Descripción
-c	Cuenta las líneas
-i	Ignora mayúsculas y minúsculas
-f	Usa la expresión regular incluida en el archivo indicado
-n	Busca solamente en la linea indicada
-v	Muestra lo que NO coincida
-w	Expresión EXACTA
-o	Patrones (expresiones regulares)

```
# Este ejemplo captura las IPs
grep -Eo '[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}'

# Podría llegar a acortarse como si de aritmética se tratara
grep -Eo '([0-9]{1,3}\.){3}[0-9]{1,3}'

# Borrar lineas con comentarios
grep '^#' /etc/lilo.conf

# Esta linea monstrara lo que contenga hda o hdb
grep 'hd[ab]' /etc/lilo.conf
```

Podemos encontrar diferentes expresiones regulares en:

http://regular-expressions.info/
https://regex101.com/

Podemos encontrar algunas variaciones de grep:

- egrep: es equivalente a grep -E, puede usar operadores como pipe | que actrúna como un o. Ej. egrep 'invención|invenciones'
- fgrep: actua como grep -F con lo que deja de interpretar las expresiones regulares

SED

Sed se usa para buscar y sustituir estándares en texto y mostrando el resultado en pantalla, su sintáxis es " sed opciones 'comando y expresión regular' archivo ". No modifica el archivo de origen.

```
# Muestra el contenido sin las lineas iniciadas por #
sed -e '/^#/d' /etc/lilo.conf
     |      |
  Opcion  Comando
```

Opciones de SED

Modificador	Descripción
-e	Ejecuta la siguiente expresión y comando
-f	lee expresiones y comandos del archivo
-n	no muestra las lineas que no correspondan con la expresión

Comandos de SED

Comando	Acción
s	sustituir
d	borrar linea
r	inserta el contenido del archivo indicado en la ocurrencia de la expresión
w	escribe la salida en el archivo indicado
g	sustituye todas las ocurrencias de la expresión en la línea actual

8 - Vi

Lo mejor en este caso es adjuntar directamente el manual de referencia rápida.

s globales de sustitución

sustitución/opción
Sustituye con las opciones el patrón por sustitución entre las líneas x e y

s (*substitute*) sustituye solo la primera ocu-
a cadena.

Rango de líneas x e y entre las que se efectuará la sustitución
Si se trata de todo el fichero, el rango puede sustituirse por este meta–carácter

Meta–carácter básico de sustitución de los caracteres de la expresión
Meta–carácter si el set nomagic está activo
Meta–secuencia que es sustituida por el carácter indicado en la n–ésima subexpresión incluida entre '**\(**' y '**\)**'
Convierte a mayúsculas el carácter
Convierte a minúsculas el carácter
Convierte a mayúsculas hasta encontrar el carácter '**\E**' ó '**\e**'
Convierte a minúsculas hasta encontrar el carácter '**\E**' ó '**\e**'

Sustituye todas las ocurrencias de una línea
Las líneas son impresas en pantalla
Sustituciones condicionadas. **Y** confirma la sustitución, otra tecla como no

Cambia 'esto' por 'Esto'
Cambia 'esto' por 'ESTO'
uel\)/\U\1E \u\2
Cambia 'esto aquel' por 'ESTO Aquel'

Repite el último comando **:s** de sustitución

es con ficheros

Escribe el fichero

Comandos: (macros y abreviaturas)

Estos comandos pueden ser definidos a través de la variable EXINIT de entorno.

```
EXINIT='<comando>|<comando>|…'
<comando>:  set options
            map …
            ab …
export EXINIT (en Bourne shell)
```

Alternativamente pueden ser guardadas en el fichero **.exrc** del directorio del usuario.

"	Línea de comentarios
abbr	Abreviaturas: :abbr uci Unidad de Cuidados Intensivos
map	Macros: :map v:!clear^M
set opción	Opciones

Opciones

Existen dos tipos de opciones que se (des)activan con el comando **set**:

- Booleanas (on/off)
- Numéricas (que necesitan un valor)

:set opción	Activa la opción
:set noopción	Desactiva la opción
ai **(noai)**	:set ai (autoindent), habilita la autoindentación. Se vuelve al principio de línea con Ctrl-D
ht	:set ht=5 (hartabs), espacio de tabulación
nu **(nonu)**	Numera las líneas (solo informativamente)
ts	:set ts=5 (tabstop), espacio de tabulación
wm	:set wm=5 (wrapmargin), hace un retorno automático a partir del quinto último carácter de la línea

Vi

Guía de Referen Rápida