

RESUMEN DE LOS PRINCIPALES COMANDOS DE LINUX

Orden	OPCIONES	DESCRIPCIÓN
cal		Imprime el calendario del mes actual.
cat	-u, -v	muestra por pantalla el contenido de un fichero que recibe como parámetro
cd		cambia de directorio
chgrp		Cambia el grupo al que pertenece un archivo o directorio
chmod	+-xwr, g o u	Este comando asigna o quita permisos a los ficheros o directorios.
cp	-i, -p, -r	copia un fichero o directorio desde un origen a un destino
date		fecha y hora actuales
df		Informa del espacio total ocupado por cada sistema de ficheros.
du	-h	Permite conocer el espacio ocupado por un disco o directorio
file		Muestra información detalla del fichero que recibe como argumento
find	-name,-print	encuentra los archivos en el camino que se le indique como parámetro
finger		muestra información relativa a los usuarios conectados al sistema
grep		busca un patrón dentro de una cadena
gzip	-d	Permite comprimir el fichero que se le pasa como parámetro
head		muestra las primeras líneas de un fichero
kill	-9	mata un proceso activo en el sistema
ln	-f, -s	crea un enlace a un fichero
ls	-l, -a, -q, -d, -m	listado de los ficheros que se encuentran en el directorio que se le pasa por parámetro
mail		escribe un mensaje a un usuario de nuestro sistema
man		muestra información sobre el comando que recibe como parámetro
mkdir		crea un directorio
more		muestra de forma pagina el contenido de un fichero
mv		mueve o renombra un fichero a una dirección o nombre especificado
passwd		cambia la palabra de paso del usuario que la ejecuta
ps	-a,-e,-f	muestra los procesos activos en este momento en el sistema
pwd		muestra el directorio activo en este momento
rm	-f, -i, -r	borra el archivo que se le pasa como parámetro
rmdir	-p	borra directorios
sed		Editor de textos en línea.
su		permite al usuario convertirse en superusuario
tail		muestra el final de un fichero
tar	-x -p -v -f -c	Permite crear o extraer de un fichero tar
touch		crea un fichero vacío
wall		escribe un mensaje para difundirlo a todos los usuarios locales
wc	-l, -c, -w	cuenta líneas, caracteres o palabras dentro de un fichero

PROGRAMACIÓN SE SHELL-SCRIPTS

LECTURA Y ESCRITURA DE DATOS.

La forma de declarar una variables de entorno de Linux es

\$nombre_variable = valor

y la forma de ver su contenido es

Secho \$nombre_variable

Para que el valor de una variable, que va a ser utilizada en un shell-scripts de Linux, sea introducida por teclado, se usa la orden read.

read variable.

SENTENCIAS DEL LENGUAJE DE PROGRAMACIÓN

if

```
if expresión then
ordenes
else
ordenes
fi
```

Si se desea introducir otro operador if a continuación de else se utiliza la forma abreviada (elif)

```
if expresión
then
ordenes
elif expresión
then
ordenes
if
```

• La orden test

La orden test ofrece valores al operador if. La orden test puede ser sustituida por corchetes []. Los argumentos que se usan son:

-f fichero:	Devuelve cierto si el fichero existe es un fichero normal.
-r fichero:	Devuelve cierto si el fichero existe y tenemos permiso de lectura.
-w fichero:	Devuelve cierto si el fichero existe y se puede modificar.
-x fichero:	Devuelve cierto si el fichero existe y podemos ejecutarlo
-d fichero.	Devuelve cierto si es directorio.

Operadores relacionales:

-eq Igual	-lt Menor que
-ne Distintos	-le Menor o Igual que
-gt Mayor que	-ge Mayor o igual que

Los operadores relacionales nos permiten construir expresiones compuestas.

-a operador Y
-o operador O
! operador NOT

• Operador Case

```
case $variable in
valor1)
ordenes
;;
valorn)
ordenes
;;
*)
ordenes
esac
```

- **Operador for**

```
for variable in valores
do
    ordenes
done
```

- **Operador while**

```
while expresión
do
    ordenes
done
```

- **Orden exit**

La orden exit finaliza la ejecución del shell-script y puede tomar como argumento un valor que devolverá al Shell.

La variable \$? devuelve 0 si la orden que se acaba de ejecutar se ha ejecutado correctamente, en este caso si la orden ls es correcta, \$? devuelve un 0 y es el valor que comparamos con \$EXIT0.

.

- **Los parámetros \$# y \$***

\$# contiene el número de parámetros pasados y \$* los valores de todos los parámetros.

Para obtener el valor de cada parámetro individualmente se utilizan \$0 , que contiene el nombre del shell-script, \$1, \$2, hasta \$9 que son los parámetros que recibe el programa.