

Script en Linux Ubuntu

Cómo se hacen los bash script:

El shell dispone de un conjunto de comandos que vienen de serie. Son comandos muy sencillos que pueden ayudar en la programación de scripts. Vamos a listarlos, pueden usarse dentro de un script o desde el propio prompt:

!	negación
#	comentarios
:	comando nulo
.	el punto sirve para ejecutar ficheros
alias	sirve para crear alias de comandos largos: alias dir='ls -l more'
bind	gestiona la librería readline
bg	pasa procesos a background o segundo plano
break	fuerza la salida de for, while, select o until
builtin	permite modificar el comportamiento de comandos
cd	el cambio de directorio
command	ejecutar determinado comando con sus argumentos
continue	salta a la siguiente iteración en for, while, select o until
declare	declaración de variables
dirs	muestra la pila de directorios
disown de este	quita los procesos dependientes del shell para que no dependan
do, done	parte de las iteraciones
echo	mostrar datos por salida
enable	habilitar o deshabilitar los comandos builtin.
exec	ejecuta un comando en lugar del proceso actual
exit	termina el script o el shell
export	convierte las variables en globales
fc	editar comandos del historial
fg	pasar comandos a primer plano
getopts	para tomar los argumentos opcionales del script, al estilo de c
help	muestra la ayuda
history	el historial de comandos
if	para crear estructuras condicionales
jobs	muestra los procesos que tenemos en marcha
kill	envío de señales a procesos.
let	ejecución de operaciones aritméticas

local	declaración de variables locales
logout	salida del shell
popd	saca un directorio de la pila de directorios
printf	sacar datos formateados como en c
pwd	muestra el directorio actual
pushd	mete un directorio en la pila
read	lee desde la entrada estándar
readonly	previene que variables de shell sean sobrescritas
return	para terminar funciones
select	estructura de selección
set	muestra variables del shell, y puede establecer muchas opciones
shopt	establece o quita opciones de shell
shift	desplaza posiciones en el shell
source	similar al .
suspend	suspende el shell actual, suele hacerse en el su
test	para construir condicionales
time	mide el tiempo de ejecución de un comandos
times	muestra tiempos acumulados de procesos

Para crear un script lo primero es indicar el lugar donde se encuentra nuestro bash, (intérprete de comandos) por defecto esta en /bin/bash.

Esto lo hacemos del siguiente modo `#!/bin/bash`

Creamos el script 01 con el editor nano

Si el archivo no tiene permisos de ejecución se los asignamos.

```
chmod +x uno
```

y a continuación lo ejecutamos

```
. uno
```

```
-lt menor      -le menor o igual
-gt mayor      -ge mayor o igual
-eq igual      -ne no igual
```

El comando **let** nos permite operar con variables numéricas de una manera fácil y rápida.

Condicionales

En el shell podemos crear las habituales estructuras condicionales if o if-else.

Esta sería la forma de **if**:

```
if condición; then
    operaciones
fi
```

o también:

```
if condición
then
    operaciones
fi
```

Esta sería la forma del if-else

```
if condición
then
    operaciones
else
    operaciones
fi
```

```
While condición
Do
<comandos a ejecutar>
done
```

```
until condición
do
<comandos a ejecutar>
done
```

Bucle **until**, es similar a while Nos permita realizar un bucle condicional hasta que se cumpla una condición, en el momento que se cumpla la condición sale del bucle. Su funcionalidad es prácticamente la

misma que el while.

Su ejecución se detiene de forma inversa.