



ADMINISTRACION de SISTEMAS

Linux Shell

Operaciones Lógicas y Evaluación de Condiciones

Estas variables se mantienen a todo lo largo de una sesión aparecen incorporadas en el fichero

.bash_profile

Si no existe se usa el fichero **/etc/profile**

Para mostrar las variables de entorno establecidas actualmente usaremos el comando:

env

```
#!/bin/bash
```

```
if comando
```

```
then #tiene que ir en una linea diferente de if o usa ;
```

```
comando 1
```

```
comando 2
```

```
...
```

```
[else # la ejecución del comando devuelve 1 (=false)
```

```
comando 3
```

```
comando 4
```

```
...]
```

```
fi
```

```
#!/bin/bash
# cadena.sh
dir=/home/blanca
#Sacar el contenido del directorio si existe
# "2>/dev/null" oculta el mensaje de error si no existe
if cd $dir 2>/dev/null; then
    echo "Estas en $dir."
else
    echo "No puedes ir al directorio $dir."
fi
```

```
#!/bin/bash
# cadena.sh
if echo "$VAR" | grep -q txt      # if [[ $VAR = *txt* ]]
then
    echo "$VAR contiene la subcadena \"txt\""
fi
```

Comando test o []

```
#!/bin/bash
```

```
if test condicion-true
```

```
#Otra forma if [ condicion-true ] importante blancos
```

```
then #tiene que ir en una linea diferente de if o usa ;
```

```
comando 1
```

```
comando 2
```

```
...
```

```
[else # Cuando la condición no es verdad
```

```
comando 3
```

```
comando 4
```

```
...]
```

```
fi
```

Operadores del comando test para comparaciones numéricas

test n1 **-eq** n2, cierto si n1 y n2 son iguales

test n1 **-ne** n2, cierto si n1 y n2 son distintos

test n1 **-ge** n2, cierto si n1 es mayor o igual que n2

test n1 **-gt** n2, cierto si n1 es mayor que n2

test n1 **-le** n2, cierto si n1 es menor o igual que n2

test n1 **-lt** n2, cierto si n1 es menor que n2

```
#!/bin/bash
```

```
# soy_root.sh: ¿Soy root o no?
```

```
# root tiene $UID con valor 0.
```

```
if test $UID -eq 0 # ¿Es root de verdad?
```

```
then
```

```
    echo "root".
```

```
else
```

```
    echo "Usted es un usuario normal."
```

```
fi
```

```
exit 0
```

Operadores del comando test para el tratamiento de cadenas

`test cadena1 = cadena2`

cierto si cadena1 es igual que cadena2

`test cadena1 != cadena2`

cierto si cadena1 es distinta que cadena2

`test cadena` #cierto si cadena no es la cadena nula

`test -z cadena` #cierto si la longitud de la cadena es 0

```
#!/bin/bash
# soy_root2.sh: ¿Soy root o no?
# O... `id -nu` en lugar de $(id -nu)
# Otra posibilidad es usar el comando whoami
if [ $(id -nu) = "root" ]
then
    echo "root".
else
    echo "Usted es un usuario normal."
fi
exit 0
```

Operadores del comando test para el tratamiento de archivos. Devuelve true si ...

- e archivo, cierto si existe
- f archivo, cierto si es un archivo normal (no un directorio o fichero de dispositivo)
- s archivo, cierto si no es tamaño cero
- d archivo, cierto si es un directorio

Operadores del comando test para el tratamiento de archivos. Devuelve true si ...

- r archivo, cierto si tiene permiso de lectura (por el usuario que ejecuta la prueba)
- w archivo, cierto si tiene permiso de escritura (para el usuario que ejecuta la prueba)
- x archivo, cierto si tiene permiso de ejecución (para el usuario que ejecuta

```
#!/bin/bash
```

```
# fichero_pass.sh
```

```
file=/etc/passwd
```

```
if [[ -e $file ]]      # if test -e $file
```

```
then
```

```
echo "El fichero password existe."
```

```
fi
```

Estructuras de Control Repetitivas

Bucle **while**, se ejecutarán mientras la condición especificada sea cierta

while {*comando*}

do

comando_1

....

comando_n

done

```
#!/bin/bash
N=0
ACUM=0
while test $N -le 9
  do
    echo "Introduce el valor $N"
    read VALOR
    #expr me permite realizar operaciones aritmeticas
    ACUM=`expr $ACUM + $VALOR`
    N=`expr $N + 1 `
  done
echo "Acumulado = $ACUM"
```

Bucle **for**, se ejecutará una vez por cada valor que haya en el conjunto de valores

for *variable* [in *conjunto de valores*]

do

comando_1

....

comando_n

done

```
#!/bin/bash
#envio_mail.sh
for USER in usu1 usu2 usu3;
do
    echo "enviado mensaje a: $USER";
    mail $USER < msg
done
```

En la sentencia **for**

si se omite **in conjunto de valores**

los comandos que aparezcan entre do y done serán ejecutados una vez por cada parámetro posicional pasado

```
#!/bin/bash
# crear-fich.sh
for A
do
    # Crea los ficheros pasados como argumentos
    > $A
done
```

Otra sintaxis para el bucle **for**

for ((variable=valor-inicial; variable<=valor-final; incremento))

do

comando_1

....

comando_n

done