

Curso de administración de sistemas GNU/Linux
Módulo de Bash scripting
Construcciones *if-then-else-fi* y
comparaciones.

Por Jorge Fuertes

<http://jorgefuertes.com>

©2009 Jorge Fuertes Alfranca
Revisado a 15 de mayo de 2009

Índice

1. Introducción	3
2. Operadores comparativos	3
2.1. Comparaciones numéricas	3
2.2. Comparaciones alfanuméricas	4
3. Ejercicios	5
3.1. Enunciados	5
3.2. Soluciones	7
4. Sobre esta unidad didáctica	16
4.1. Notas y advertencias	16
4.2. Derechos	16
4.3. Agradecimientos	16
4.4. Revisiones	16

1. Introducción

Las construcciones *if* sirven para reaccionar de distinta forma dependiente de una condición. Por ejemplo, en pseudo código, diríamos:

```
pon valor=1
si valor es mayor que 0
entonces
    decir "El valor es mayor que cero."
en otro caso
    decir "El valor no es mayor que cero."
fin
```

Como se puede ver a simple vista lo que hacemos es poner el número **1** dentro de la variable *valor*. Después miramos si dicha variable es mayor que cero, y de ser así pasamos a ejecutar el bloque *entonces* y si no lo es, hipotéticamente, pasaríamos a ejecutar el bloque *en otro caso*.

Esto puesto en idioma *Bash* es así:

```
#!/bin/bash

let VALOR=1
if [ $VALOR -gt 0 ]
then
    echo "El valor es mayor que cero."
else
    echo "El valor no es mayor que cero."
fi
```

Nota: Son importantes los espacios antes y después de los corchetes.

Cambia poco sobre el pseudocódigo. Salvando que las sentencias son en inglés, sólo hay que tener cuidado con los corchetes que encierran la comparación, con el operador de comparación en sí y con que, en *Bash*, cuando escribimos en una variable la llamamos sólo por su nombre, pero cuando queremos leer su contenido debemos especificar un signo de dólar al inicio de la misma, por ejemplo "\$VALOR".

2. Operadores comparativos

Son los operadores que podemos utilizar en las sentencias "*if*" para realizar las comparaciones:

2.1. Comparaciones numéricas

Operadores para comparaciones entre número o variables con contenido numérico.

Op:	Comparación:	Traducción:	Ejemplo:
-lt	lesser than	menor que	if [\$VAR -lt 5]
-le	lesser than o equal to	menor o igual que	if [\$VAR -le 7]
-gt	greater than	mayor que	if [\$VAR -gt 5]
-ge	greater than or equal to	mayor o igual que	if [\$VAR -ge 4]
-eq	equal to	igual a	if [\$VAR -eq 8]
-ne	Not equal to	no es igual a	if [\$VAR -ne 7]
-z	is zero	es cero	if [-z \$VAR]

2.2. Comparaciones alfanuméricas

Operadores para comparaciones entre cadenas de texto.

Op:	Comparación:	Traducción:	Ejemplo:
==	equal to	igual que	if ["\$VAR" == "hola"]
!=	not equal to	distinto de	if ["\$VAR" != "hola"]
<	lesser than	alfabéticamente anterior a	if ["\$VAR" < "hola"]
>	greater than	alfabéticamente posterior a	if ["\$VAR" > "hola"]
-z	zero lenght	está vacía	if [-z "\$VAR"]

Nota: Son importantes los espacios antes y después de los corchetes.

3. Ejercicios

3.1. Enunciados

Cree un script o guión de *Bash* para cada uno de los ejercicios, y llámelo *ejercicio-num.sh*, siendo *num* el número de ejercicio:

1. Pida dos números por pantalla y que diga cual de ellos es menor¹.
2. Pida el nombre al usuario y vuelvalo a pedir si dicho nombre está vacío y si no lo está que diga "Hola " y el nombre.
3. Programe un script que pida un número y que diga "Bien: mayor de diez." si dicho número es menor que 10, dé un error por pantalla.
4. Pida un número y diga si es mayor, menor o igual que 10.
5. El ejecutar el script de este ejercicio debe aparecer en pantalla el número de mes en el que nos encontramos y si pertenece al primer, segundo, tercer o último cuarto del año.
6. Debe aparecer en pantalla cuantos segundos y cuantos años han transcurrido desde 1970.

¹para pedir entradas al usuario deberá utilizar la orden `read`

7. Haga un programa que hable con el usuario. Dicho programa deberá saber contestar a las expresiones hola, adiós, que tal, y que fecha es.

8. **Medio:** Dado un fichero con una lista de nombres que usted deberá crear, haga un guión que cree un directorio `/root/otros-usuarios` y dentro de este un directorio para cada uno de los nombres, con un fichero en él con nombre `nombredelalista-bienvenido.txt`, y que contenga la frase `"Hola nombre, bienvenido."`.

9. **Medio:** Pregunte al usuario fecha de nacimiento, a partir de ella dígame el número de segundos, minutos y horas que ha vivido.

10. **Avanzado:** Programe una pequeña calculadora que permita restar, sumar, dividir y multiplicar dos números. Dicha calculadora debe decir las operaciones con palabras, por ejemplo: `"10 multiplicado por 2 es igual a 20."`.

11. **Avanzado:** Haga lo mismo en el ejercicio 7, pero el programa debe entender también la frase `"mi nombre es x"`, siendo `x` un nombre cualquiera, y contestar con `"Hola, x. ¿Como estás?"`. Después consiga que si se contesta `"bien"` a esa frase el programa diga `"Me alegro, x"` pero que si el usuario introduce `"bien"` sin haber sido preguntado acerca de ello, el programa responda `"¿Bien? ¿Bien qué?"`.

12. **Avanzado:** Retoque el programa anterior para que reconozca todas las palabras tanto en mayúsculas como en minúsculas, con espacios delante o detrás o sin ellos.

3.2. Soluciones

1. Pida dos números por pantalla y que diga cual de ellos es menor².

```
#!/bin/bash

echo -n "Introduzca un número: "
read NUM1
echo -n "Introduzca otro número: "
read NUM2

if [ $NUM1 -gt $NUM2 ]
then
    echo "$NUM2 es menor que $NUM1"
else
    echo "$NUM1 es menor que $NUM2"
fi
```

2. Pida el nombre al usuario y vuélvalo a pedir si dicho nombre está vacío y si no lo está que diga "Hola " y el nombre.

```
#!/bin/bash

echo -n "Introduzca su nombre: "
read nombre

if [ -z $nombre ]
then
    echo -n "introduzca de nuevo su nombre: "
    read nombre
fi

echo "Hola $nombre"
```

3. Programe un script que pida un número y que diga "Bien: mayor de diez." si dicho número es menor que 10, dé un error por pantalla.

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

²para pedir entradas al usuario deberá utilizar la orden **read**

```

echo -n "Introduzca un número: "
read NUM

if [ $NUM -lt 10 ]
then
    echo "ERROR: Menor que 10."
elif [ $NUM -gt 10 ]
then
echo "Bien: Mayor que 10."
fi

```

4. Pida un número y diga si es mayor, menor o igual que 10.

```

#!/bin/bash

echo "Introduzca Valor"
read VAL1

if [ $VAL1 -eq 10 ]
then
    echo " $VAL1 Es igual a 10"
elif [ $VAL1 -lt 10 ]
then
    echo "$VAL1 Es menor a 10"
elif [ $VAL1 -gt 10 ]
then
    echo "$VAL1 Es mayor a 10"
else
    echo "Ni es igual, ni menor, ni mayor que 10."
fi

```

5. El ejecutar el script de este ejercicio debe aparecer en pantalla el número de mes en el que nos encontramos y si pertenece al primer, segundo, tercer o último cuarto del año.

```

#!/bin/bash

FECHA="$(date +%m'|bc)"
echo "Estamos en el mes nº $FECHA."

if [ $FECHA -le 3 ]
then
    echo "es el primer cuarto del año"
elif [ $FECHA -le 6 ]
then
    echo "es el segundo cuarto del año"
elif [ $FECHA -le 9 ]
then
    echo "es el tercer cuarto del año"

```

```

        else
            echo "es el último cuarto del año"
        fi

```

6. Debe aparecer en pantalla cuantos segundos y cuantos años han transcurrido desde 1970.

```

#!/bin/bash

VAR1=$(date +%s")
echo "Han pasado $VAR1 seg desde 1970"

ANO=$(date +%Y")
ANOS=$(expr $ANO - 1970)
echo "Han transcurrido $ANOS desde 1970."

```

7. Haga un programa que hable con el usuario. Dicho programa deberá saber contestar a las expresiones hola, adiós, que tal, y que fecha es.

```

#!/bin/bash

clear
echo -n "Introduce una expresión: "
read DATO

if [ "$DATO" == "hola" ]
then
    echo "Hola ¿Qué tal?"
elif [ "$DATO" == "adiós" ]
then
    echo "Hasta luego."
elif [ "$DATO" == "qué tal" ]
then
    echo "Bien, ¿Y tú?"
elif [ "$DATO" == "qué fecha es" ]
then
    echo "Estamos a $(date +%d-%m-%Y')."
else
    echo "¡No entiendo!"
fi

```

8. **Medio:** Dado un fichero con una lista de nombres que usted deberá crear, haga un guión que cree un directorio `/root/otros-usuarios` y dentro de este un directorio para cada uno de los nombres, con un fichero en él con nombre `nombredelalista-bienvenido.txt`, y que contenga la frase `"Hola nombre, bienvenido."`.

```

#!/bin/bash

```

```

DIRUSUS="/root/otros-usuarios"
NOMBRES="nombres23.txt"

if [ -f $NOMBRES ]
then
  for NOMBRE in $(cat $NOMBRES)
  do
    echo -n "Creando directorio de $NOMBRE..."
    mkdir -p $DIRUSUS/$NOMBRE
    echo "OK"
    echo "Hola $NOMBRE, bienvenido." > \
      $DIRUSUS/$NOMBRE/$NOMBRE-bienvenido.txt
  done
else
  echo "No existe el fichero $NOMBRES."
fi

echo "Finalizado."

```

9. **Medio:** Pregunte al usuario fecha de nacimiento, a partir de ella dígame el número de segundos, minutos y horas que ha vivido.

```

#!/bin/bash

clear

echo -n "Año de nacimiento: "
read YEAR
echo -n "Mes de nacimiento: "
read MES
echo -n "Día de nacimiento: "
read DIA

SEG=$(date --date $YEAR-$MES-$DIA +%s)
HASTAHOY=$(date +%s)
MISSEG=$(expr $HASTAHOY - $SEG)

MIN=$(expr $MISSEG / 60)
HORAS=$(expr $MIN / 60)

echo "Nacido en $DIA-$MES-$YEAR."
echo "Has vivido $HORAS horas. $MIN minutos. $MISSEG segundos."

```

10. **Avanzado:** Programe una pequeña calculadora que permita restar, sumar, dividir y multiplicar dos números. Dicha calculadora debe decir las operaciones con palabras, por ejemplo: "10 multiplicado por 2 es igual a 20".

```

#!/bin/bash

```

```

# Limpiar pantalla:
clear

# Pedir datos al usuario:
echo -n "Introduce el primer numero: "
read num1

echo -n "Introduce operacion (+, -, *, /):"
read oper
if [ -z $oper ]
then
    echo "ERROR: Debe introducir una operación."
    exit 1
fi

echo -n "Introduce el segundo numero: "
read num2

opertexto=$(echo "$oper"| \
    sed "s/\*/multiplicado por/"| \
    sed "s/\+/mas/"| \
    sed "s/\-/menos/"| \
    sed "s/\//dividido por/")

# Calculadora:
res=$(echo "scale=2; $num1 $oper $num2"|bc)
echo "Operación: $num1 $opertexto $num2 es igual a $res"
# FIN DE PROGRAMA
exit

# Hasta aquí el programa optimizado.
# Una solución válida con IF sería:

# Operación de calculadora:
res=$(echo "scale=2; $num1 $oper $num2"|bc)

# Cambiar signo por texto:
if [ "$oper" == "+" ]
then
    echo "$num1 mas $num2 es igual a $res"
fi

if [ "$oper" == "-" ]
then
    echo "$num1 menos $num2 es igual a $res"
fi

if [ "$oper" == "*" ]

```

```

then
    echo "$num1 multiplicado por $num2 es igual a $res"
fi

if [ "$oper" == "/" ]
then
    echo -e "$num1 dividir por $num2 es igual a $res"
fi

```

11. **Avanzado:** Haga lo mismo en el ejercicio 7, pero el programa debe entender también la frase "*mi nombre es x*", siendo *x* un nombre cualquiera, y contestar con "*Hola, x. ¿Como estás?*". Después consiga que si se contesta "*bien*" a esa frase el programa diga "*Me alegre, x*" pero que si el usuario introduce "*bien*" sin haber sido preguntado acerca de ello, el programa responda "*¿Bien? ¿Bien qué?*".

```

#!/bin/bash

# Limpiar pantalla:
clear

echo -n "Introduce una expresión: "
read dato

if [ "$dato" == "hola" ]
then
    echo "hola majo"
elif [ "$dato" == "adiós" ]
then
    echo "hasta luego"
elif [ "$dato" == "qué tal" ]
then
    echo "bien y tú"
elif [ "$dato" == "qué fecha es" ]
then
    echo "Estamos a $(date +%d-%m-%Y)."
elif [ "$dato" == "bien" ]
then
    echo "¿Bien? ¿Bien qué?"
else
    echo $dato|grep "mi nombre es" &> /dev/null
    if [ $? -eq 0 ]
    then
        nombre=$(echo $dato|cut -f4 -d" ")
        echo -n "Hola $nombre ¿cómo estás?: "
        read dato2
        dato2=$(echo $dato2|tr "A-Z" "a-z")
        if [ "$dato2" == "bien" ]
        then
            echo "Me alegre, $nombre"

```

```

        fi
    else
        echo "Lo siento, mi no entenderrrr."
    fi
fi

```

12. **Avanzado:** Retoque el programa anterior para que reconozca todas las palabras tanto en mayúsculas como en minúsculas, con espacios delante o detrás o sin ellos.

```

#!/bin/bash

# Limpiar pantalla:
clear

echo -n "Introduce una expresión: "
read dato

# *** A Ñ A D I M O S   E S T O : ***
dato=$(echo $dato|tr "A-Z" "a-z")

if [ "$dato" == "hola" ]
then
    echo "hola majo"
elif [ "$dato" == "adiós" ]
then
    echo "hasta luego"
elif [ "$dato" == "qué tal" ]
then
    echo "bien y tú"
elif [ "$dato" == "qué fecha es" ]
then
    echo "Estamos a $(date +%d-%m-%Y)."
elif [ "$dato" == "bien" ]
then
    echo "¿Bien? ¿Bien qué?"
else
    echo $dato|grep "mi nombre es" &> /dev/null
    if [ $? -eq 0 ]
    then
        nombre=$(echo $dato|cut -f4 -d" ")
        echo -n "Hola $nombre ¿cómo estás?: "
        read dato2
        dato2=$(echo $dato2|tr "A-Z" "a-z")
        if [ "$dato2" == "bien" ]
        then
            echo "Me alegro, $nombre"
        fi
    else
        echo "Lo siento, mi no entenderrrr."
    fi
fi

```

```
        fi
    fi
```

13. Como ejercicio adicional se ha mejorado entre toda la clase el programa que habla con el usuario, permitiendo una conversación constante y más fluida:

```
#!/bin/bash
clear

cosa=""
RESP=""
NOMBRE=""

while [ "$cosa" != "adios" ]
do
if [ -z "$pregunta" ]
then
echo "¿Qué me cuentas? ('adiós' para terminar): "
else
echo "$pregunta ('adios' para terminar): "
fi
read cosa
echo $cosa|grep -i "mi nombre es" &> /dev/null
if [ $? -eq 0 ]
then
NOMBRE=$(echo $cosa|cut -f4 -d" ")
pregunta="Hola, ${NOMBRE}. ¿Cómo estas?"
else
cosa=$(echo $cosa|tr "A-Z" "a-z")
case $cosa in
"hola")
echo "Hola"
;;
"que tal")
echo "Muy bien"
;;
"que fecha es")
echo "La fecha de hoy es $(date +%d-%m-%Y)"
;;
"bien")
if [ -z "$pregunta" ]
then
echo "¿Bien? ¿Bien qué?"
else
echo "Me alegro ${NOMBRE}."
pregunta=""
fi
;;
"adios")
```

```
echo "Vale, hasta otra."  
exit  
;;  
*)  
echo "Lo siento, no te entiendo."  
esac  
fi  
done
```

4. Sobre esta unidad didáctica

4.1. Notas y advertencias

Debian: Esta guía está basada en el sistema *Debian GNU/Linux*, podría haber pequeños cambios si se aplica a otras distribuciones de *GNU*, pero en su mayor parte funcionará bien con la excepción de lo referido al sistema de paquetería de programas, los comandos que empiezan por *apt*, ya que otras *distros* no basadas en *Debian* podrían incorporar sistemas diferentes para el manejo de sus paquetes.

4.2. Derechos

Esta guía se cede bajo contrato Coloriuris. Sólo puede ser utilizada previa aceptación del contrato de cesión sito en:

- <http://www.coloriuris.net/contratos/ef5af6aaa441ab9c213273fade56dca1>

Dicho contrato garantiza que estoy cediendo los derechos de uso y modificación sin ánimo de lucro.

4.3. Agradecimientos

El autor quiere reflejar su agradecimiento a todas las páginas de Internet que ponen a disposición de todo el mundo sus contenidos, así como a todo aquél que publica artículos, manuales y experiencias en Internet, ya que eso favorece a la difusión del conocimiento y al desarrollo humano. *La información quiere ser libre.*

Un agradecimiento muy especial a toda la comunidad del Software Libre. Sin ellos el autor viviría en la oscuridad: Programadores, traductores, asociaciones, hacktivistas, webmasters, etc...

También quiero agradecer muy especialmente su ayuda a mis alumnos y lectores, por tomarse la molestia de comunicarme las erratas y por darme ideas para mejorar los ejercicios.

4.4. Revisiones

El autor irá eventualmente publicando revisiones de esta unidad en su página personal, y estará encantado de recibir sugerencias y dudas en la misma o en su email:

- <http://jorgefuertes.com>.
- cursos@jorgefuertes.com.

Por supuesto se puede contactar con el autor para contratarle para hacer nuevas unidades, adaptaciones, modificaciones, cursos, etc...