

Curso de sistemas GNU/Linux
Bash scripting
Construcciones ”**case**”

Por Jorge Fuertes
<http://jorgefuertes.com>

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

Índice

1. Introducción	3
1.1. ¿Qué es case?	3
1.2. Ejemplo en pseudocódigo	3
1.3. Una construcción típica	3
2. Ejercicios	4
2.1. Enunciados	4
2.2. Soluciones	6
3. Sobre esta unidad didáctica	11
3.1. Notas y advertencias	11
3.2. Derechos	11
3.3. Agradecimientos	11
3.4. Revisiones	11

1. Introducción

1.1. ¿Qué es case?

Las construcciones *case* permiten, en base a un valor o a un variable, tomar acciones dependiendo de dicho valor, a veces de una forma más sencilla que con una construcción *if*.

Para ello se define la variable de cabecera y una serie de valores (o casos) que de corresponderse con la variable provocarán la ejecución de las sentencias que le siguen.

Básicamente las construcciones *case* sirven para lo mismo que las *if*, pero pueden resultar mucho más claras y sencillas de utilizar en determinadas ocasiones.

1.2. Ejemplo en pseudocódigo

En pseudocódigo sería algo así:

```
en_caso_de_que VARIABLE valga
  100
    decir La variable vale 100.
  200
    decir La variable vale 200.
  300
    decir La variable vale 300.
  hola
    decir La variable vale hola.
  cualquier_otra_cosa
    decir La variable no corresponde con ninguna de las anteriores.
fin
```

Si seguimos el programa visualmente, veremos que si la variable es igual a cualquier de los casos contemplados (100, 200, 300, hola) el programa "dirá" lo que vale, y que si no corresponde con ninguno dirá la última frase, que no corresponde con ningún valor contemplado, lo que equivaldría a un "else" en una construcción *if*. Eso se consigue, en *case*, con un asterisco (*).

1.3. Una construcción típica

El programa anterior transcrito a código real de *Bash* quedaría así:

```
#!/bin/bash

echo -n "Introduzca valor: "
read $VARIABLE

case $VARIABLE in
  100)
    echo "La variable vale 100."
  ;;
  200)
```

```

        echo "La variable vale 200."
    ;;
300)
    echo "La variable vale 200."
    ;;
hola)
    echo "La variable vale hola."
    ;;
*)
    echo "La variable no corresponde con ninguna de las anteriores."
    ;;
esac

echo "Fin de programa."

```

Observaremos que en primer lugar pedimos una entrada al usuario y después reaccionamos a ella con la construcción *case*. Si la entrada de usuario corresponde a alguno de los casos, se ejecutarán las sentencias entre dicho caso y el cierre con doble punto y coma (";;"), lo que puede parecer algo extraño pero que resulta útil si se hace una de estas construcciones en la línea de comandos.

Por último el comodín (*) capturará cualquier caso no contemplado y ejecutará sus sentencias correspondientes, de la misma forma que hubiese concordancia con el valor de *\$VARIABLE*.

La construcción *case* queda cerrada con la palabra *esac*, que no es sino *case* al revés, de la misma forma que los *if* se cierran con *fi*.

2. Ejercicios

2.1. Enunciados

Cree un script o guión de *Bash* para cada uno de los ejercicios, y llámelo *ej-case-num.sh*, siendo *num* el número de ejercicio¹:

1. Pida al usuario un número resultado de una tirada de un dado (del 1 al 6) y diga en pantalla el resultado con letras en vez de con números.
2. Copie y modifique el programa anterior pero ahora debe decir también que hay en la cara contraria del dado. El programa deberá contemplar si se introduce un número no válido, dar un error y volver a pedir constantemente un número hasta que sea válido (de 1 a 6).

¹En Español, la construcción "siendo num tal cosa" quiere decir que hay que sustituir *num* por lo que se dice a continuación, no que haya que poner literalmente *num*.

3. Utilizando *case*, cree una calculadora. El programa debe pedir el primer número, la operación a realizar con texto (sumar, restar, multiplicar o dividir) y el segundo número y realizar la operación con una precisión decimal de tres dígitos.

4. Elabore un programa que pida una letra minúscula y diga si la mayúscula correspondiente es recta (A, E, F, H, I, K, L, M, N, T, V, W, X, Y, Z), curva (C, O, S, U, Q) o ambas cosas (B, D, G, J, P, R).

5. Utilizando *case*, haga un programa que hable con el usuario. Dicho programa deberá saber contestar a las expresiones hola, adiós, que tal, y que fecha es. El programa debe terminar, despidiéndose del usuario, al ordenarle "*salir*".

6. Escriba un programa que diga, sin preguntar nada, que día de la semana es hoy, en texto y utilizando *case*.

2.2. Soluciones

1. Pida al usuario un número resultado de una tirada de un dado (del 1 al 6) y diga en pantalla el resultado con letras en vez de con números.

```
#!/bin/bash

clear

read -p "Tirada de dado: " -n 1 DADOS
echo

case $DADOS in
    1)
        echo "Has sacado un UNO."
        ;;
    2)
        echo "Has sacado un DOS."
        ;;
    3)
        echo "Has sacado un TRES."
        ;;
    4)
        echo "Has sacado un CUATRO."
        ;;
    5)
        echo "Has sacado un CINCO."
        ;;
    6)
        echo "Has sacado un SEIS."
        ;;
    *)
        echo "ERROR: El dado sólo puede ir de 1 a 6."
        ;;
esac
```

2. Copie y modifique el programa anterior pero ahora debe decir también que hay en la cara contraria del dado. El programa deberá contemplar si se introduce un número no válido, dar un error y volver a pedir constantemente un número hasta que sea válido (de 1 a 6).

```
#!/bin/bash

clear

while [ 1 ]
do
```

```

read -n1 -p "Introduzca un número (0 para salir): " DADO
echo

case $DADO in
    0)
        echo "Adiós."
        exit
        ;;
    1)
        echo "La tirada del dado es uno."
        echo "La cara contraria es seis."
        ;;
    2)
        echo "La tirada del dado es dos."
        echo "La cara contraria es cinco."
        ;;
    3)
        echo "La tirada del dado es tres."
        echo "La cara contraria es cuatro."
        ;;
    4)
        echo "La tirada del dado es cuatro."
        echo "La cara contraria es tres."
        ;;
    5)
        echo "La tirada del dado es cinco."
        echo "La cara contraria es dos."
        ;;
    6)
        echo "La tirada del dado es seis."
        echo "La cara contraria es uno."
        ;;
    *)
        echo "ERROR: El dado sólo puede ser 1 a 6."
        ;;
esac
done

```

- Utilizando *case*, cree una calculadora. El programa debe pedir el primer número, la operación a realizar con texto (sumar, restar, multiplicar o dividir) y el segundo número y realizar la operación con una precisión decimal de tres dígitos.

```

#!/bin/bash

clear

echo -n "INTRODUZCA UN NUMERO: "
read NUM1

```

```

echo -n "INTRODUZCA LA OPERACION: "
read OPR
echo -n "INTRODUZCA OTRO NUMERO: "
read NUM2

case $OPR in
    "SUMA")
        OP="+"
        ;;
    "RESTA")
        OP="-"
        ;;
    "MULTIPLICAR")
        OP="*"
        ;;
    "DIVIDIR")
        OP="/"
        ;;
esac

echo El resultado es $(echo "scale=3; ${NUM1} ${OP} ${NUM2}"|bc).

```

4. Elabore un programa que pida una letra minúscula y diga si la mayúscula correspondiente es recta (A, E, F, H, I, K, L, M, N, T, V, W, X, Y, Z), curva (C, O, S, U, Q) o ambas cosas (B, D, G, J, P, R).

```

#!/bin/bash

clear

read -p "Introduce una letra: " letra
letra=$(echo $letra|tr "a-z" "A-Z")

case $letra in
    C|O|S|U|Q)
        echo "Corresponde a curva"
        ;;
    B|D|G|J|P|R)
        echo "Corresponde a recta y a curva"
        ;;
    A|E|F|H|I|K|L|M|N|T|V|W|X|Y|Z)
        echo "Corresponde a recta"
        ;;
esac

```

5. Utilizando *case*, haga un programa que hable con el usuario. Dicho programa deberá saber contestar a las expresiones hola, adiós, que tal, y que fecha es. El programa debe terminar, despidiéndose del usuario, al ordenarle "salir".

```

#!/bin/bash

clear

while [ 1 ]
do
    read -p "Introduce una expresión: " pal
    pal=$(echo $pal|tr "A-Z" "a-z")

    case $pal in
        "adios")
            echo "Hasta luego"
            exit
            ;;
        "hola")
            echo "Buenas dias"
            ;;
        "que tal")
            echo "Podia ser mejor"
            ;;
        "que fecha es")
            echo "$(date +%D)"
            ;;
        *)
            echo "No entiendo"
            ;;
    esac
done

```

6. Escriba un programa que diga, sin preguntar nada, que día de la semana es hoy, en texto y utilizando case.

```

#!/bin/bash

clear

DIA=$(date +%u)

case $DIA in
    1)
        echo "Hoy es Lunes."
        ;;
    2)
        echo "Hoy es Martes."
        ;;
    3)
        echo "Hoy es Miércoles."
        ;;
    4)

```

```
                echo "Hoy es Jueves."  
;;  
5)              echo "Hoy es Viernes."  
;;  
6)              echo "Hoy es Sábado."  
;;  
7)              echo "Hoy es Domingo."  
;;  
esac
```

3. Sobre esta unidad didáctica

3.1. 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.

3.2. 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.

3.3. 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.

3.4. 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...