Tema 05 - Linux: Scripts

Scripts

1. Introducción

Los scripts se utilizan para automatizar o realizar tareas sin tener que repetirlas una y otra vez, ya que en el propio fichero del script introduciremos todos los comandos necesarios para realizar la tarea.

Para crear el script nos basta con el editor de texto que venga en nuestra terminal de Linux. Para crear un script es necesario:

1. Crear un fichero de texto en la ruta donde deseamos tener el script.
2. Guardar el fichero con extensión "**.sh**"
3. Para que se reconozca como un script debemos escribir como primera línea "**#!/bin/bash**" (sin las comillas).

2. Variables

Como cualquier otro lenguaje de programación, necesitamos variables que nos servirán para guardar datos en la memoria del ordenador hasta el momento que los necesitemos.

Para asignar el valor a una variable simplemente simplemente debemos usar el signo =:

nombre_variable=valor_variable

¡Importante! → no dejar espacios ni antes ni después del =.

Para recuperar el valor de dicha variable sólo hay que anteponer el símbolo de dolar $ antes del nombre de la variable:

$nombre_variable

2. Variables

Nombre de las variables

Las variables pueden tomar prácticamente cualquier nombre, sin embargo, existen algunas restricciones:

- Sólo puede contener caracteres alfanuméricos y guiones bajos
- El primer carácter debe ser una letra del alfabeto o “_” (este último caso se suele reservar para casos especiales).
- No pueden contener espacios.
- Las mayúsculas y las minúsculas importan, “a” es distinto de “A”.
- Algunos nombres son usado como variables de entorno y no los debemos utilizar para evitar sobreescribirlas (por ejemplo: PATH).

3. Bucle for

```
for VARIABLE in LISTA_VALORES;
do
    COMANDO 1
    COMANDO 2
    ...
    COMANDO N
done
```

La lista de valores puede ser:

- **un rango numérico:**

for VARIABLE in 1 2 3 4 5 6 7 8 9 10;

for VARIABLE in {1..10};

- **una serie de valores:**

for VARIABLE in file1 file2 file3;

- **o el resultado de la ejecución de un comando:**

for VARIABLE in $(ls /bin | grep -E

3. Bucle for

Por ejemplo

```
#!/bin/bash

for numero in {1..20};

do

  echo Este es el número: $numero

done
```

4. Condicional if

```
if [ CONDICIÓN ];
then
  COMANDO 1 si se cumple la condición
else
  COMANDO 2 si no se cumple la condición
fi
```

```
if [ CONDICIÓN 1 ];
then
  COMANDO 1 si se cumple la condición 1
elif [ CONDICIÓN 2 ];
then
  COMANDO 2 si se cumple la condición 2
else
  COMANDO 3 si no se cumple la condición 2
fi
```

4. Condicional if

Condicionales

operador	significado
-lt	menor que (<)
-gt	mayor que (>)
-le	menor o igual que (<=)
-ge	mayor o igual que (>=)
-eq	igual (==)
-ne	no igual (!=)

4. Condicional if

Por ejemplo

```
#!/bin/bash
num1=$1  # la variable toma el primer valor que le pasamos al script
num2=$2  # la variable toma el segundo valor que le pasamos al script
if [ $num1 -gt $num2 ];
then
  echo $num1 es mayor que $num2
else
  echo $num2 es mayor que $num1
fi
```

5. Iteraciones: Bucle while

```
while [ condition ];do
    commands
done
```

Por ejemplo

```
#!/bin/bash

number=10

while [ $number -gt 5 ];do

    echo $number
    number=$(($number-1))

done
```

5. Iteraciones: Bucle until

```
until [ condition ];do
    commands
done
```

Por ejemplo

```
#!/bin/bash

number=10

while [ $number -gt 5 ];do

    echo $number
    number=$(($number-1))

done
```

5. Iteracion: case

```
case [ expresion ] in
    caso1)
            commands
    ;;
    caso2)
            commands
    ;;
    *)
            commands
;;
esac
```

Por ejemplo

```
#!/bin/bash

echo “Teclee una opción:”
read opc

case $opc in
   1)
            echo “1”
   ;;
   2)
            echo “2”
   ;;
   *)
            echo “Otro”
   ;;
esac
```

6. Manipulación de cadenas de texto

Extraer subcadena

Mediante ${cadena:posicion:longitud} podemos extraer una subcadena de otra cadena. Si omitimos :longitud, entonces extraerá todos los caracteres hasta el final de cadena.

Por ejemplo en la cadena string=abcABC123ABCabc:

- *echo ${string:0} : abcABC123ABCabc*
- *echo ${string:0:1} : a (primer carácter)*
- *echo ${string:7} : 23ABCabc*
- *echo ${string:7:3} : 23A (3 caracteres desde posición 7)*
- *echo ${string: -4} : Cabc (la cuarta posición desde atrás hasta el final)*
- *echo ${string: -4:2} : Ca (dos caracteres desde la cuarta posición desde atrás)*

6. Manipulación de cadenas de texto

Borrar subcadena

Hay diferentes formas de borrar subcadenas de una cadena:

- ${cadena#subcadena} : borra la coincidencia más corta de subcadena desde el principio de cadena
- ${cadena##subcadena} : borra la coincidencia más larga de subcadena desde el principio de cadena

Por ejemplo, en la cadena string=abcABC123ABCabc:

- *echo ${string#a*C} : 123ABCabc*
- *echo ${string##a*C} : abc*

6. Manipulación de cadenas de texto

Reemplazar subcadena

También existen diferentes formas de reemplazar subcadenas de una cadena:

- ${cadena/buscar/reemplazar} : Sustituye la primera coincidencia de buscar con reemplazar
- ${cadena//buscar/reemplazar} : Sustituye todas las coincidencias de buscar con reemplazar

Por ejemplo, en la cadena string=abcABC123ABCabc:

- *echo ${string/abc/xyz} : xyzABC123ABCabc.*
- *echo ${string//abc/xyz} : xyzABC123ABCxyz.*

7. Operaciones aritméticas

- **+, -:** suma, resta

```
num=10

echo $((num + 2))
```

- ***, /, % :** multiplicación, división, resto (módulo)

```
echo $((num * 2))

echo $((num / 2))
echo $((num % 2))
```

- **VAR++ VAR– :** post-incrementa, post-decrementa

```
echo $((num++))

echo $num
```

- **++VAR –VAR :** pre-incrementa, pre-decrementa

```
echo $((++num))

echo $num
```

8. Funciones

En Linux podemos crear funciones:

```
function nombre_funcion(){
        commands
        return  *: sólo devuelve números
}
```

Por ejemplo

```
#!/bin/bash

function saludo() {
   echo “Hola mundo !!! ”
}

saludo
```

Para llamar a la función:

```
nombre_funcion
```

Tarea 01: Scripts iniciales I

1. Dado por teclado tres caracteres, imprímelos en orden inverso.
2. Dados por teclado tres números enteros diferentes, imprimir los números en orden descendente.
3. Dado por teclado los datos categoría y el sueldo de un trabajador, calcule el aumento correspondiente teniendo en cuenta la siguiente tabla. Imprima la categoría del trabajador y su nuevo sueldo.

Categoría	Aumento
1	15%
2	10%
3	8%
4	7%

1. Crea un programa que lea dos números de manera REPETITIVA. El programa termina cuando ambos números son iguales.
2. Crea un programa que lee un número por pantalla y te dice qué día de la semana es.

Tarea 02: Scripts iniciales II

1. Crea un programa que simule el juego del “Piedra, papel o tijera”. Si ambos jugadores empatan se repite, en caso contrario el programa termina y muestra al jugador ganador.
2. Modifica el anterior ejercicio para que muestre el nombre del jugador que gane primero 3 partidas.
3. Crea un script que reciba un número entero por teclado y dice si es positivo, negativo o cero.
4. Crea un script que te diga los 10 primeros múltiplos de 2.
5. Modifica el anterior script para que, de esos 10 números, te diga sólo los que son múltiplos también de 3.
6. Crea un script que lea palabras y las guarde al final de un fichero, hasta que se escriba “:q”