

Bash – Los parámetros

Junio 2014

- [Introducción](#)
- [Los parámetros posicionales](#)
 - [Ejemplo 1](#)
- [Los parámetros especiales](#)
 - [Ejemplo 2](#)
- [Inicializar parámetros](#)
 - [El comando “set”](#)
 - [Ejemplos](#)
 - [El comando “shift”](#)
 - [Ejemplo 3](#)

Introducción

Es posible pasar a un script, en la línea de comando, los argumentos que necesita para su ejecución. Estos argumentos son llamados “parámetros”. Existen dos categorías de estos: los parámetros posicionales y los parámetros especiales.

Los parámetros posicionales

Estos son simplemente los argumentos pasados al script, cuando un script es invocado. Estos son almacenados en las variables reservadas 1,2,3,...9,10,11,... y pueden ser llamados con las expresiones \$1,\$2...\${10},\${11}... Nota: El shell Bourne está limitado a los parámetros del 0 al 9.

Ejemplo 1

A continuación un pequeño script que tan solo muestra ciertos argumentos pasados como parámetros en función de su posición

```
#!/bin/bash
# muestra_param.sh

echo "El 1er parámetro es: $1"
echo "El 3er parámetro es: $3"
echo "El 10mo parámetro es: ${10}"
echo "El 15 parámetro es: ${15}"
```

Tan solo hay que llamar al script y pasarle un cierto número de parámetros:

```
./muestra_param.sh 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17

El 1er parámetro es: 1
El 3er parámetro es: 3
El 10mo parámetro es: 10
El 15 parámetro es: 15
```

O también:

```
./muestra_param.sh uno 2 tres 4 5 6 7 8 9 diez 11 12 13 14 quince 16 17

El 1er parámetro es: uno
El 3er parámetro es: tres
El 10mo parámetro es: diez
El 15 parámetro es: quince
```

Si algunos parámetros contienen caracteres especiales o espacios, entonces es necesario ponerlo entre comillas:

```
./muestra_param.sh uno 2 "el 3ro" 4 5 6 7 8 9 diez 11 12 13 14 "el 15" 16 17

El 1er parámetro es: uno
El 3er parámetro es: el 3ro
El 10mo parámetro es: diez
El 15 parámetro es: el 15
```

Los parámetros especiales

Estas también son variables reservadas que permite en algunos casos efectuar operaciones sobre los mismos parámetros. Estos parámetros son los siguientes:

```
$0 Contiene el nombre del script tal como es invocado
$* El conjunto de todos los parámetros en un solo argumento
$@ El conjunto de argumentos, un argumento por parámetro
$# El número de parámetros pasados al script
$? El código de retorno del último comando
$$ El PID del shell que ejecuta el script
$! El PID del último proceso ejecutado en segundo plano
```

Ejemplo 2

A continuación un pequeño script poniendo en práctica los parámetros especiales visto líneas arriba:

```
#!/bin/bash
# muestra_param_2.sh

# Mostrar el nombre del script
echo "El nombre de mi script es: $0"
# Mostrar el número de parametros
echo "Has pasado $# paramètros"
# Lista de parámetros (un solo argumento)
for param in "$*"
do
    echo "Aquí la lista de parámetros (un solo argumento) : $param"
done
# Lista de parámetros (un parámetro por argumento)
echo "Aquí la lista de parámetros (un parámetro por argumento) : "
for param in "$@"
do
    echo -e "\tParámetro : $param"
done
# Mostrar el proceso
echo "El PID del shell que ejecuta el script es: $$"
# Ejecución de un comando en segundo plano
sleep 100 &
# Mostrar el proceso ejecutado en segundo plano
echo "El PID del ultimo comando ejecutado en segundo plano es: $!"
# Mostrar el código de retorno del último comando "echo"
echo "El código de retorno del comando anterior es: $?"
# Generación de un error
echo "Generación de un error..."
# Mostrar el comando errado
echo "ls /etc/password 2>/dev/null"
ls /etc/password 2>/dev/null
# Mostrar el código de retorno del último comando
echo "El código de retorno del comando precedente es: $?"
exit
```

Lo que resulta con la siguiente invocación:

```
./muestra_param_2.sh 1 2 3 cuatro 5 seis
```

El nombre de mi script es: ./muestra_param_2.sh

Has pasado 6 parámetros

Esta es la lista de parámetros (un solo argumento): 1 2 3 cuatro 5 seis

Esta es la lista de parámetros (un parámetro por argumento) :

Parámetro: 1

Parámetro: 2

Parámetro: 3

Parámetro: cuatro

Parámetro: 5

Parámetro: seis

El PID del shell que ejecuta el script es: 6165

El PID del último comando ejecutado en segundo plano es: 6166

El código de retorno del comando precedente es: 0

Generación de error...

ls /etc/password 2>/dev/null

El código de retorno del comando precedente es: 1

Inicializar parámetros

El comando “set”

Es posible pasar directamente parámetros al shell gracias al comando “**set**”. Un simple comando como:

```
set param1 param2 param3
```

Inicializará automáticamente los parámetros posicionales “**\$1,\$2,\$3**” con los valores “**param1,param2,param3**”, borrando de este modo los antiguos valores si existieran. Los parámetros especiales “**#,* et @**” son automáticamente actualizados.

Ejemplos

```
$ set param1 param2 param3
$ echo "Número de parámetros: $#"  
Número de parámetros: 3
$ echo "El segundo parámetro es: $2"  
El segundo parámetro es: param2
$ echo "Los parámetros son: $@"  
Los parámetros son: param1 param2 param3

$ set melocotón manzana
$ echo "Número de parámetros: $#"  
Número de parámetros: 2
$ echo "Los parámetros son: $@"  
Los parámetros son: melocotón manzana
```

Esta funcionalidad puede ser útil en el tratamiento de archivos línea por línea a fin de aislar cada palabra (campo), y formatear la salida de estos.

```
$ IFS=":"; set $(grep $USER /etc/passwd)
$ echo -e "Login :t$1\nNom :t$5\nID :t$3\nGrupo :t$4\nShell :t$7"

Login: jp
Nombre: Jean-Philippe
ID: 500
Grupo: 500
Shell: /bin/bash
```

El comando “shift”

El comando interno “**shift**” permite desplazar los parámetros. El valor del 1er parámetro (\$1) es reemplazado por el valor del 2do parámetro (\$2), el del 2do parámetro (\$2) por el del 3er parámetro (\$3), ... Es posible indicar como argumento (**shift n**) el número de posiciones que hay que desplazar los parámetros.

Ejemplo 3

A continuación veremos el empleo del comando interno “**shift**”.

```
#!/bin/bash
# desplaza_param.sh

echo
echo "Número de parámetros: $#"
```

El script muestra los parámetros de posición en tres estados:

- Estado 1 (Inicial):** Muestra los parámetros originales (\$1 a \$10).
- Estado 2 (Desplazado 1 posición):** Se ejecuta `shift`, desplazando los parámetros una posición hacia la izquierda.
- Estado 3 (Desplazado 4 posiciones):** Se ejecuta `shift 4`, desplazando los parámetros cuatro posiciones hacia la izquierda.

```
echo "El 1er parámetro es: $1"
echo "El 3er parámetro es: $3"
echo "El 6to parámetro es: $6"
echo "El 10mo parámetro es: ${10}"
echo "=====
```

Desplazar una posición con el comando \"shift\"

```
shift
echo "Número de parámetros: $#"
```

El script muestra los parámetros de posición en tres estados:

- Estado 1 (Inicial):** Muestra los parámetros originales (\$1 a \$10).
- Estado 2 (Desplazado 1 posición):** Se ejecuta `shift`, desplazando los parámetros una posición hacia la izquierda.
- Estado 3 (Desplazado 4 posiciones):** Se ejecuta `shift 4`, desplazando los parámetros cuatro posiciones hacia la izquierda.

```
echo "El 1er parámetro es: $1"
echo "El 3er parámetro es: $3"
echo "El 6to parámetro es: $6"
echo "El 10mo parámetro es: ${10}"
echo "=====
```

Desplazar cuatros posiciones con el comando \"shift 4\"

```
shift 4
echo "Número de parámetros: $#"
```

El script muestra los parámetros de posición en tres estados:

- Estado 1 (Inicial):** Muestra los parámetros originales (\$1 a \$10).
- Estado 2 (Desplazado 1 posición):** Se ejecuta `shift`, desplazando los parámetros una posición hacia la izquierda.
- Estado 3 (Desplazado 4 posiciones):** Se ejecuta `shift 4`, desplazando los parámetros cuatro posiciones hacia la izquierda.

```
echo "El 1er parámetro es: $1"
echo "El 3er parámetro es: $3"
echo "El 6to parámetro es: $6"
echo "El 10mo parámetro es: ${10}"
echo
```

Y su resultado

```
./desplaza_param.sh 1 2 3 4 5 6 7 8 9 10
```

Número de parámetros: 10

El 1er parámetro es: 1

El 3er parámetro es: 3

El 6to parámetro es: 6

El 10mo parámetro es: 10

=====

Desplazar una posición con el comando "shift"

Número de parámetros: 9

El 1er parámetro es: 2

El 3er parámetro es: 4

El 6to parámetro es: 7

El 10mo parámetro es:

=====

Desplazar cuatro posiciones con el comando "shift 4"

Número de parámetros: 5

El 1er parámetro es: 6

El 3er parámetro es: 8

El 6to parámetro es:

El 10mo parámetro es:

Este documento intitulado « [Bash – Los parámetros](#) » de [Kioskea](#) (es.kioskea.net) esta puesto a disposición bajo la licencia [Creative Commons](#). Puede copiar, modificar bajo las condiciones puestas por la licencia, siempre que esta nota sea visible.