

BASH 03.02 - Shell script

[Información sobre el documento.....1](#)

[1. - Introducción.....2](#)

[2. - Actividades.....3](#)

[Pasando parámetros y leyendo parámetros.....3](#)

[Explicaciones.....3](#)

[Mostrando información de un fichero.....4](#)

[Explicaciones básicas.....4](#)

[Comprobando parámetros.....5](#)

[Explicaciones básicas.....5](#)

[Información de ficheros.....6](#)

[Explicaciones básicas.....6](#)

[Información de ficheros y permisos.....7](#)

[Explicaciones básicas.....7](#)

[Log del usuario.....8](#)

[Información del script.....8](#)

Información sobre el documento

El objetivo de este documento es enseñar, compartir conocimientos para facilitar el aprendizaje. Este documento es mejorable, y será actualizado si es preciso.

Este documento puede ser utilizado para uso personal, no comercial, como se presenta, respetando una serie de **condiciones de uso basadas en Licencia Creative Commons**, como aparece en el logotipo, que se describe a continuación



- El **documento se proporciona como está**, por tanto *no se pueden realizar modificaciones*, ni en el **formato** ni en el **contenido**, ni trabajos derivados, sin la autorización expresa del autor.
- Se debe **mencionar al autor** del mismo, por supuesto sin modificar los enlaces o imágenes introducidas por éste en el documento.
- **No se puede utilizar** este material **con fines lucrativos, comerciales o cualquier uso que pueda proporcionar, directa o indirectamente, un beneficio económico de terceros**, sin la autorización expresa del autor.
- Si se desea publicar el documento en algún sitio web, se debe hacer a través de un **enlace al documento en el sitio del autor** (www.educatica.es o cursos.educatica.es)

Si se desea publicar como un recurso dentro de un sitio web, sin utilizar un enlace al material en el sitio web del autor, se debe solicitar autorización expresa y, en cualquier caso, referenciar el sitio web del autor (www.educatica.es).

Espero sea de provecho ;)



1. - Introducción

El objetivo de este documento es practicar en la creación de shell scripts bash.

En este documento vamos a trabajar con:

1. El paso de valores a un script a través de parámetros.
2. Las variables de parámetro y variables especiales.
3. Las sentencias condicionales IF y el comando test.

Para sacar el máximo partido al documento, se recomienda leer el apartado donde se presentan los contenidos en el material didáctico proporcionado en el sitio web de [educatica](http://educatica.es) o de terceros autores.



2. - Actividades

Pasando parámetros y leyendo parámetros

Con el editor de textos que prefieras, crea un nuevo fichero llamado **muestraParametros.sh** en el directorio de scripts de tu usuario. El fichero deberá contener el siguiente código:

```

1 #!/bin/bash
2 #
3 # Autor: Javier Pedrero Martín - javier@educatica.es
4 # Fecha: Febrero 2019
5 #
6 # ....
7
8 echo "Nombre del comando/script: $0"
9 echo "Número de parámetros pasados: $#"
```

```

10 echo
11 echo "Parámetro 1: $1"
12 echo "Parámetro 2: $2"
13 echo "Parámetro 3: $3"
14 echo "Parámetro 4: $4"
15 echo "Parámetro 5: $5"
16 echo "Parámetro 6: $6"
17 echo "Parámetro 7: $7"
18 echo "Parámetro 8: $8"
19 echo "Parámetro 9: $9"
20 echo
21 echo "Cadena con todos los parámetros a partir del 1: $*"
22 echo "Array de parámetros, cada uno una cadena: @$@"
23 echo
24 echo "Valor de retorno del último comando: $?"
25 echo "PID del último proceso lanzado en segundo plano: $!"
26 echo "PID del proceso actual: $$"
27 echo "Parámetros pasados al último script o comando ejecutado: $_"
28 echo "Parámetros especiales de bash: $-"
```

1. Antes de ejecutar el script, trata de averiguar o intuir que hace cada línea y qué representa cada variable que se utiliza. Recuerda, las variables vienen precedidas por un carácter \$.
2. Desde una terminal, configura el script para que pueda ser ejecutado y ejecútalo como hemos hecho hasta ahora. Comprueba el resultado que te da.
3. Ahora ejecuta el script pero pasándole parámetros, comparando las salidas que muestra.

```
./muestraParametros hola que tal
```

Explicaciones

Un script puede recibir información por parámetro, igual que cualquier comando que ejecutemos. Como programadores, debemos decidir

1. Si vamos a utilizar el parámetro. Puede que nuestro script no utilice parámetros, y no los usará aunque el usuario los pase. Como hemos hecho hasta ahora.
2. Si es obligatorio, necesitamos, que nos pasen un parámetro para poder continuar. En caso de no recibir el parámetro podemos cancelar la ejecución del script.



3. Si necesitamos comprobar que el parámetro es correcto. Si el parámetro pasado no es válido, podemos decidir si cancelar la ejecución o realizar otra acción por defecto.

Mostrando información de un fichero

Crea un script llamado muestraInfoFichero.sh en tu directorio de scripts. Deberás seguir las pautas dadas para la creación de cualquier script:

- Interprete que usará el script
- Resumen básico de la utilidad del script
- Información de autor y fecha

Este script deberá realizar las siguientes tareas:

1. Mostrar información de permisos, propietario y grupo del fichero cuya ruta se pasa como primer parámetro del script.
 1. Será tan sencillo como usar el comando ls con la opción -l y la ruta del fichero.

```
ls -ld <rutaFichero>
```

Una vez creado el script, deberás configurarlo para que pueda ser ejecutado y ejecutarlo desde la terminal.

Prueba a ejecutarlo de las siguientes formas y comprueba que pasa

1. Sin pasarle ningún parámetro.
2. Pasándole como parámetro la ruta de un fichero que exista.
3. Pasándole como parámetro la ruta de un fichero que no exista

Explicaciones básicas

El script funcionará correctamente solo en el caso 2, cuando le pases como parámetro la ruta de un fichero que exista.

```
javier@educatica:~$ ls -l muestraParametros.sh
-rwxrwxr-x 1 javier javier 775 feb 11 19:50 muestraParametros.sh
```

Si le pasamos como parámetro la ruta de un fichero que no existe, el comando ls -ld se ejecutará correctamente, pero como el fichero no existe mostrará un mensaje de error.

```
javier@educatica:~$ ls -l rutaNoValida
ls: no se puede acceder a 'rutaNoValida': No existe el archivo o el directorio
```

Si no se le pasa parámetro, la variable \$1 estará vacía y por tanto la shell la sustituye por una cadena vacía. Esto provoca que se muestre el directorio actual, que es lo que sucede cuando se ejecuta ls -l sin parámetros.

```
ls -l
```

Comprobando parámetros

Crea un script llamado **muestraInfoFichero02.sh** en tu directorio de scripts. Deberás seguir las pautas dadas para la creación de cualquier script: Interprete que usará el script, resumen básico de la utilidad del script e información de autor y fecha.

Este script deberá realizar las siguientes tareas:

1. Deberá recibir por parámetro la ruta de un fichero.
 1. Si no se pasa parámetro, es decir el primer parámetro no tiene contenido, se deberá mostrar un mensaje de error y salir. Para salir puedes ejecutar el comando exit con un número distintos de 0 por parámetro por ejemplo exit 1.
2. Comprobar si el fichero cuya ruta se ha pasado por parámetro existe.
 1. Si el fichero cuya ruta se ha pasado no existe, se deberá dar un mensaje de error y salir.
3. Mostrar información de permisos, propietario y grupo del fichero cuya ruta se pasa como primer parámetro del script.
 1. Será tan sencillo como usar el comando ls con la opción -l y la ruta del fichero.

```
ls -ld <rutaFichero>
```

Una vez creado el script, deberás configurarlo para que pueda ser ejecutado y ejecutarlo desde la terminal.

Prueba a ejecutarlo de las siguientes formas y comprueba que pasa

1. Sin pasarle ningún parámetro.
2. Pasándole como parámetro la ruta de un fichero que exista.
3. Pasándole como parámetro la ruta de un fichero que no exista

Explicaciones básicas

Deberás utilizar el comando if para comprobar si se pasa parámetro o no. Consulta la ayuda del comando if para ver como funciona. La sintaxis básica del comando es:

```
if <condicion>; then
sentencias
fi
```

<condición> es una comprobación de condición que se realiza ejecutando un comando. Si el resultado del comando es 0, es decir todo salió bien, entonces se cumple la condición. Si retorna cualquier otro valor, entonces la condición no se cumplirá.

Para realizar comprobaciones, por ejemplo con cadenas, valores numéricos o ficheros, podemos utilizar el comando test. Lee la ayuda del comando test.

El comando test se puede simplificar sustituyéndolo por su forma reducida [EXPRESION]. Por ejemplo, para comprobar si una cadena está vacía podemos usar:

```
if [ -z $variable]; then
..
fi
```

Información de ficheros

Crea un script llamado **muestraInfoFichero03.sh** en tu directorio de scripts. Deberás seguir las pautas dadas para la creación de cualquier script:

- Interprete que usará el script
- Resumen básico de la utilidad del script
- Información de autor y fecha

Este script deberá realizar las siguientes tareas:

1. Deberá recibir por parámetro la ruta de un fichero.
 1. Si no se pasa parámetro, es decir el primer parámetro no tiene contenido, se deberá mostrar un mensaje de error y salir. Para salir puedes ejecutar el comando `exit` con un número distinto de 0 por parámetro, por ejemplo **exit 1**.
2. Mostrar información de permisos concretos del fichero:
 1. Si el fichero no existe, mostrar un mensaje indicando que el fichero no existe.
 2. Si es un directorio mostrar un mensaje indicando que es un directorio.
 3. Si es un fichero normal mostrar un mensaje indicando que es un fichero.
 4. Si es un enlace simbólico mostrar un mensaje indicando que es un enlace simbólico.
 5. Si somos el propietario del fichero mostrar un mensaje indicando que somos el propietario del fichero.
 6. Si pertenecemos al grupo del fichero, mostrar un mensaje indicando que pertenecemos al grupo del fichero.
 7. Si tenemos permiso de lectura sobre el fichero, mostrar un mensaje indicando que tenemos permisos de lectura sobre el fichero.
 8. Si tenemos permiso de escritura sobre el fichero, mostrar un mensaje indicando que tenemos permisos de escritura sobre el fichero.
 9. Si es un fichero y tenemos permiso de ejecución sobre el fichero, mostrar un mensaje indicando que tenemos permisos de ejecución sobre el fichero.
 10. Si es un directorio y tenemos permiso de acceso sobre el fichero, mostrar un mensaje indicando que tenemos permisos de acceso sobre el fichero.

Explicaciones básicas

Deberás utilizar el comando `if` de nuevo, pero tendrás que mirar la ayuda del comando `test` para tener en cuenta las opciones de comparación en la expresión que se comprueba.

```
If [ EXPRESION ]; then
sentencias
fi
```

Información de ficheros y permisos

Crea un script llamado **muestralInfoFichero04.sh** cuyo contenido sea el mismo que el del script **muestralInfoFichero03.sh** en tu directorio de scripts.

Mantén el código del script, puesto que vamos a añadir unas líneas al final para realizar las siguientes tareas extra:

1. Muestra en pantalla el mensaje “Fichero “ seguido del nombre del fichero con el que estamos trabajando.
2. Muestra en pantalla el mensaje “Propietario del fichero “ seguido del nombre del propietario.
 1. **Comandos y tuberías:** Esta información la puedes obtener usando tuberías con los comandos con los que hemos estado trabajando en clase.
3. Muestra en pantalla el mensaje “Grupo del fichero “ seguido del nombre del grupo.
 1. **Comandos y tuberías.**
4. Muestra en pantalla el mensaje “Permisos propietario: “ seguido de los permisos concretos que tiene el propietario del fichero.
 1. **Comandos y tuberías.**
5. Muestra en pantalla el mensaje “Permisos grupo: “ seguido de los permisos concretos que tiene el grupo del fichero.
 1. **Comandos y tuberías.**
6. Muestra en pantalla el mensaje “Permisos otros: “ seguido de los permisos concretos que tiene el resto de usuarios sobre el fichero.
 1. **Comandos y tuberías.**
7. Muestra en pantalla el mensaje “Tamaño: “ seguido del tamaño del fichero en un formato comprensible por humanos.
 1. **Comandos y tuberías.**

Explicaciones básicas

En este caso podemos utilizar el comando echo con la opción -n para que el cursor no baje a una nueva línea y vuelva al comienzo de la misma. De esta forma, la salida del siguiente comando se mostrará donde esté el cursor, que será a continuación de la cadena mostrada, en la misma línea.

Para mostrar información concreta de permisos, propietario y grupo de un fichero, vamos a usar tuberías y los principales comandos de filtrado de salida para obtener la información que necesitamos. Recuerda que hemos trabajado con varios, pero en este caso los fundamentales serán grep, cut y tr.

Log del usuario

Copia el script llamado **userInfo02.sh** como **userInfo03.sh** y vamos a modificarlo para añadir nuevas características.

Este script deberá realizar las mismas tareas que el script `userInfo02.sh` solo que vamos a obtener el nombre del usuario por parámetro y además comprobaremos que el usuario existe y que su directorio personal también existe.

1. Comprueba que se pasa como parámetro el nombre del usuario.
 1. Si no se pasa parámetro se deberá mostrar un mensaje de error y salir con un código de error -1.
2. Comprueba que el usuario existe.
 1. Si el usuario no existe se deberá mostrar un mensaje de error y salir con un código de error -2.
3. Crea una nueva variable llamada `homeDir` cuyo contenido será la ruta del directorio personal del usuario que se ha insertado por parámetro. Vamos a suponer que el directorio personal de cualquier usuario está en el directorio `/home` y tiene como nombre el del usuario.
4. Comprobar que el directorio personal del usuario existe.
 1. Si no existe, se deberá mostrar un mensaje de error y salir con un código de error -3.
5. Si las comprobaciones han sido correctas, entonces mostraremos la información del script
 1. Mostrará en pantalla el siguiente mensaje: "Nombre del usuario: " seguido del nombre del usuario insertado por teclado.
 2. Mostrará en pantalla información de identificación del usuario cuyo nombre se ha insertado por teclado. Esta información de identificación estará compuesta por el nombre del usuario, el UID, el grupo principal al que pertenece con su GID y los grupos a los que pertenece el usuario. Hay un comando que nos da toda esta información.
 3. Muestre el siguiente mensaje en pantalla "Directorio personal del usuario " seguido de la ruta del directorio personal del usuario insertado por teclado. Esta información está almacenada en la variable `homeDir` creada anteriormente.
 4. Muestre en pantalla información de permisos del directorio personal del usuario insertado por teclado, no de su contenido.
 5. Muestre en pantalla el contenido del directorio personal del usuario insertado por teclado, sin información extendida de permisos.

Una vez creado el script, deberás configurarlo para que pueda ser ejecutado y ejecutarlo desde la terminal.

Información del script

Ten en cuenta que la condición de un `if` puede ser la ejecución de un comando. Si el comando termina con código de retorno 0, es decir sin errores, se cumplirá la condición. En caso contrario, código de retorno distinto de 0, sería como una condición que no se cumple.

