

PRÁCTICA 3: Variables, alias, órdenes de búsqueda y guiones (DPR)

EJERCICIO 3.1: Escriba, al menos, cinco variables de entorno junto con el valor que tienen.

```
usuario@usuario-VirtualBox:~$ printenv LANGUAGE PATH LOGNAME SHELL SHVLV
es_ES
/usr/local/sbin:
/usr/local/bin:
/usr/sbin:
/usr/bin:/sbin:
/bin:/usr/games:
/usr/local/games
usuario
/bin/bash
```

EJERCICIO 3.2: Ejecute las órdenes del cuadro e indique qué ocurre y cómo puede resolver la situación para que la variable NOMBRE se reconozca en el shell hijo.

```
usuario@usuario-VirtualBox:~$ NOMBRE=FS
usuario@usuario-VirtualBox:~$ echo $NOMBRE
FS
```

PARA HACER QUE LA VARIABLE "NOMBRE" SE MANTENGA EN EL SHELL HIJO:

```
usuario@usuario-VirtualBox:~$ export NOMBRE=FS
usuario@usuario-VirtualBox:~$ bash
usuario@usuario-VirtualBox:~$ echo $NOMBRE
FS
```

EJERCICIO 3.3: Compruebe qué ocurre en las expresiones del ejemplo anterior si se quitan las comillas dobles del final y se ponen después de los dos puntos. ¿Qué sucede si se sustituyen las comillas dobles por comillas simples?

ORDEN ORIGINAL:

```
usuario@usuario-VirtualBox:~/practica2$ echo "Los archivos que hay en el
directorio son: $(ls -l)"
Los archivos que hay en el directorio son:
total 12
drwxrwxr-x 2 usuario usuario 4096 sep 24 10:04 sources
drwxrwxr-x 2 usuario usuario 4096 sep 24 10:40 unix
drwxrwxr-x 2 usuario usuario 4096 sep 24 10:02 windows
```

ORDEN CON COMILLAS CAMBIADAS:

```
usuario@usuario-VirtualBox:~/practica2$ echo "Los archivos que hay en el
directorio son:" $(ls -l)
Los archivos que hay en el directorio son: total 12 drwxrwxr-x 2 usuario
usuario 4096 sep 24 10:04 sources drwxrwxr-x 2 usuario usuario 4096 sep
24 10:40 unix drwxrwxr-x 2 usuario usuario 4096 sep 24 10:02 windows
```

ORDEN CON COMILLA SIMPLE:

```
usuario@usuario-VirtualBox:~/practica2$ echo 'Los archivos que hay en el
directorio son: $(ls -l)'
Los archivos que hay en el directorio son: $(ls -l)
```

EJERCICIO 3.4: Pruebe la siguiente asignación: ¿Qué ha ocurrido?

```
usuario@usuario-VirtualBox:~$ numero=$numero+1
usuario@usuario-VirtualBox:~$ $numero
+1: orden no encontrada
```

LO QUE HA SUCEDIDO ES QUE LA ASIGNACIÓN REALIZADA SE HA TRANSFORMADO EN CARÁCTER, POR LO QUE NO SE PUEDE REALIZAR LA OPERACIÓN DESEADA.

EJERCICIO 3.5: Construya un guion que acepte como argumento una cadena de texto (por ejemplo, su nombre) y que visualice en pantalla la frase Holay el nombre dado como argumento.

```
#!/bin/bash
printf "Hola $1\n"

usuario@usuario-VirtualBox:~$ chmod +x ejer3.5_script.sh
usuario@usuario-VirtualBox:~$ ./ejer3.5_script.sh DanielsP
Hola DanielsP
```

EJERCICIO 3.6: Varíe el guion anterior para que admita una lista de nombres.

```
#!/bin/bash
printf "Hola $*\n"

usuario@usuario-VirtualBox:~$ ./ejer3.6_script.sh Daniel, Jose, Pablo
```

EJERCICIO 3.7: Cree tres variables llamadas VAR1, VAR2y VAR3con los siguientes valores respectivamente “hola”, “adios” y “14”.

```
usuario@usuario-VirtualBox:~$ declare VAR1=hola VAR2=adios VAR3=14
usuario@usuario-VirtualBox:~$ echo $VAR1
hola
usuario@usuario-VirtualBox:~$ echo $VAR2
adios
usuario@usuario-VirtualBox:~$ echo $VAR3
14
```

APARTADO A: Imprima los valores de las tres variables en tres columnas con 15 caracteres de ancho.

```
usuario@usuario-VirtualBox:~$ printf "%-15s %-15s %-15d\n" $VAR1 $VAR2
$VAR3
hola                adios                14
```

APARTADO B: ¿Son variables locales o globales?

```
Son variables locales.
```

APARTADO C: Borre la variable VAR2.

```
usuario@usuario-VirtualBox:~$ unset $VAR2
```

APARTADO D: Abra otra ventana de tipo terminal, ¿puede visualizar las dos variables restantes?

```
No puedo visualizarlas en una nueva ventana de terminal ya que son variables locales.
```

APARTADO E: Cree una variable de tipo vector con los valores iniciales de las tres variables.

```
usuario@usuario-VirtualBox:~$ vector=($VAR1 $VAR2 $VAR3)
```

APARTADO F: Muestre el segundo elemento del vector creado en el apartado e.

```
usuario@usuario-VirtualBox:~$ echo ${vector[1]}  
adios
```

EJERCICIO 3.8: Cree un alias que se llame **ne** (nombrado así para indicar el número de elementos) y que devuelva el número de archivos que existen en el directorio actual. ¿Qué cambiaría si queremos que haga lo mismo pero en el directorio home correspondiente al usuario que lo ejecuta?

```
usuario@usuario-VirtualBox:~$ alias ne='ls -l | wc -l'  
usuario@usuario-VirtualBox:~$ ne  
12
```

PARA EL DIRECTORIO HOME:

```
usuario@usuario-VirtualBox:~$ alias ne='ls ~ -l | wc -l'  
usuario@usuario-VirtualBox:~$ ne  
15
```

EJERCICIO 3.9: . Indique la línea de orden necesaria para buscar todos los archivos a partir del directorio home de usuario (\$HOME) que tengan un tamaño menor de un bloque. ¿Cómo la modificaría para que además imprima el resultado en un archivo que se cree dentro del directorio donde nos encontremos y que se llame archivosP?

```
usuario@usuario-VirtualBox:~$ find ~ -size -1  
/home/usuario/.config/lxsession/Lubuntu/autostart  
/home/usuario/.sudo_as_admin_successful  
/home/usuario/practica2/sources/write.odt  
/home/usuario/practica2/unix/ejercicio1  
  
usuario@usuario-VirtualBox:~$ find ~ -size -1 >> archivos
```

EJERCICIO 3.10: Indique cómo buscaría todos aquellos archivos del directorio actual que contengan la palabra “ejemplo”.

```
usuario@usuario-VirtualBox:~$ grep ejemplo ./*
```

EJERCICIO 3.11: Complete la información de findy greputilizando para ello la orden man.

INFORMACIÓN PARA FIND	
-P	Never follow symbolic links. This is the default behaviour. When find examines or prints information a file, and the file is a symbolic link, the information used shall be taken from the properties of the symbolic link itself.
-L	Follow symbolic links. When find examines or prints information about files, the information used shall be taken from the properties of the file to which the link points, not from the link itself .
-H	Do not follow symbolic links, except while processing the command line arguments. When find examines or prints information about files, the information used shall be taken from the properties of the symbolic link itself.

INFORMACIÓN PARA GREP	
-E, --extended-regexp	Interpret PATTERN as an extended regular expression (ERE, see below).
-F, --fixed-strings	Interpret PATTERN as a list of fixed strings (instead of regular expressions), separated by newlines, any of which is to be matched.
-G, --basic-regexp	Interpret PATTERN as a basic regular expression (BRE, see below). This is the default.
-P, --perl-regexp	Interpret the pattern as a Perl-compatible regular expression (PCRE). This is experimental and grep -P may warn of unimplemented features.

EJERCICIO 3.12: Indique cómo buscaría si un usuario dispone de una cuenta en el sistema.

```
usuario@usuario-VirtualBox:~$ cat /etc/passwd | grep usuario
```

EJERCICIO 3.13: Indique cómo contabilizar el número de ficheros de la propia cuenta de usuario que no tengan permiso de lectura para el resto de usuarios.

```
usuario@usuario-VirtualBox:~$ find ~ ! -perm -o+r | wc -l
876
```

EJERCICIO 3.14: Modifique el ejercicio 8 de forma que, en vez de un alias, sea un guion llamado **numE** el que devuelva el número de archivos que existen en el directorio que se le pase como argumento.

```
#!/bin/bash
#Titulo:          numE
#Fecha:           29/09/2018
#Autor:           Daniel Pérez Ruiz
#Versión:         1.0
#Descripción:     Cuenta el número de archivos del directorio
#                 dado como primer argumento
#Opciones:        Ninguna
#Uso:             numE directorio
```

```
ls -l $1 | wc -l
```

```
usuario@usuario-VirtualBox:~$ chmod +x numE.sh
usuario@usuario-VirtualBox:~$ ./numE.sh Desktop/
13
```