**Ejercicio 1.** Escriba un guion que acepte dos argumentos. El primero será el nombre de un directorio y el segundo será un valor entero. El funcionamiento del guion es el siguiente: deberán anotarse en un archivo denominado archivosSizN.txt aquellos archivos del directorio dado como argumento y que cumplan la condición de tener un tamaño menor al valor aportado en el segundo argumento. Se deben tener en cuenta las comprobaciones sobre los argumentos, es decir, debe haber dos argumentos, el primero deberá ser un directorio existente y el segundo un valor entero.

|  |  |
| --- | --- |
| #! /bin/bash |  |
|  | if[ $# == 2 ]; then |
|  | if[ -d $1 ] && [ $2 -eq $2 ];then |
|  | find $1 -size -$2 > archivosSizN.txt |
|  | else |
|  | echo "Los argumentos son invalidos" |
|  | fi |
|  | else |
|  | echo "El numero de argumentos no es valido" |
|  | fi |

**Ejercicio 2.** Escriba un guion que acepte el nombre de un directorio como argumento y muestre como resultado el nombre de todos y cada uno de los archivos del mismo y una leyenda que diga "Directorio", "Enlace" o "Archivo regular", según corresponda. Incluya la comprobación necesaria sobre el argumento, es decir, determine si el nombre aportado se trata de un directorio existente.

|  |  |
| --- | --- |
| #! /bin/bash |  |
|  | if [ -d $1 ]; then |
|  | for archivo in `ls $1` |
|  | do |
|  | if [ -d $1$archivo ];then |
|  | printf "Directorio: %\n" $archivo |
|  | elif [ -L $1$archivo ];then |
|  | printf "Enlace: %/n" $archivo |
|  | else |
|  | printf "Archivo: %\n" $archivo |
|  | fi |
|  | done |
|  | else |
|  | printf "Argumento no valido " |
|  | fi |

**Ejercicio 3**. Escriba un guion en el que, a partir de la pulsación de una tecla, detecte la zona del teclado donde se encuentre. Las zonas vendrán determinadas por las filas. La fila de los números 1, 2, 3, 4, ... será la fila 1, las teclas donde se encuentra la Q, W, E, R, T, Y,... serán de la fila 2, las teclas de la A, S, D, F, ... serán de la fila 3 y las teclas de la Z, X, C, V, ... serán de la fila 4. La captura de la tecla se realizará mediante la orden read. (vea sección 6 en página 6).

#!/bin/bash

printf "Pulsa una tecla\n"

read tecla

case $tecla in

[º0-9\'¡]) echo "Primera fila" ;;

[qwertyuiop\`+]) echo "Segunda fila" ;;

[asdfghjklñ´ç]) echo "Tercera fila" ;;

[\<zxcvbnm,.-]) echo "Cuarta fila" ;;

\*) echo "Otra linea" ;;

fi

**Ejercicio 4.** Escriba un guion que acepte como argumento un parámetro en el que el usuario indica el mes que quiere ver, ya sea en formato numérico o usando las tres primeras letras del nombre del mes y muestre el nombre completo del mes introducido. Si el número no está comprendido entre 1 y 12 o las letras no son significativas del nombre de un mes, el guion deberá mostrar el correspondiente mensaje de error.

#!/bin/bash

printf "Elige un mes del 1 al 12\n"

read mes

case $mes in

1) echo "Enero" ;;

2) echo "Febrero" ;;

3) echo "Marzo" ;;

4) echo "Abril" ;;

5) echo "Mayo" ;;

6) echo "Junio" ;;

7) echo "Julio" ;;

8) echo "Agosto" ;;

9) echo "Septiembre" ;;

10) echo "Octubre" ;;

11) echo "Noviembre" ;;

12) echo "Diciembre" ;;

\*) echo "Elige una opcion dentro del rango" ;;

esa

**Ejercicio 5.** Escriba un guion que solicite un número hasta que su valor esté comprendido entre 1 y 10. Deberá usar la orden while y, para la captura del número, la orden read (ver página 6).

#!/bin/bash

num=0

while [ $num -lt 1 ] || [ $num -gt 10 ]

do

printf "Mete un numero del 1 al 10\n"

read num

done

Ejercicio 6. Copie este ejercicio y pruébelo en su sistema para ver su funcionamiento. ¿Qué podemos modificar para que el giro se vea más rápido o más lento? ¿Qué hace la opción -e de las órdenes echo del guion?

Para que el giro sea más rápido solo hay que cambiar el valor de la variable INTERVAL, que vale 1, y poner un valor más pequeño.

-La orden -e de los echo es para poder interpretar las barras invertidas como caracter

**Ejercicio 7.** Escriba un guion que admita como argumento el nombre de un tipo de shell (por ejemplo, csh, sh, bash, tcsh, etc.) y nos dé un listado ordenado alfabéticamente de los usuarios que tienen dicho tipo de shell por defecto cuando abren un terminal. Dicha información del tipo de shell asignado a un usuario se puede encontrar en el archivo /etc/passwd y para poder filtrar la información que nos interesa nos será útil la orden siguiente:

cut -d':' -f1

que aplicada de forma encauzada con cualquier orden para mostrar el contenido del citado archivo, cortará por la columna 1 y hasta que aparezca el delimitador ":".

#!/bin/bash

for linea in `cat /etc/passwd`

do

usuario=`echo $linea|cut -d: -f1`

shell=`echo $linea|cut -d: -f7`

valor="/bin/$1"

if [[ $valor == `echo $shell` ]]

then echo $usuario >> temp.txt

fi

done

cat temp.txt|sort

rm temp.txt

**Ejercicio 8.** Dos órdenes frecuentes de Unix son tar y gzip. La orden tar permite almacenar/extraer varios archivos de otro archivo. Por ejemplo, podemos almacenar el contenido de un directorio en un archivo con

tar -cvf archivo.tar directorio (la opción -x extrae los archivos de un archivo .tar).

#!/bin/bash

#Nombre del script: cpback.sh

destino="CopiasSeguridad"

if [[ `test -d $destino && echo 0 || echo 1` == 1 ]]

then

mkdir $destino

fi

if [ $# -ge 1 ]

then

suma=0

for i in $@

do

if [ `test -e $i && echo 0 || echo 1` == 1 ]

then

suma=$[$suma + 1]

fi

done

if [ $suma -eq 0 ]

then tar -cvf "./$destino/copia`date +"%Y%m%d"`.tar" $@

gzip "./$destino/copia`date +"%Y%m%d"`.tar"

else

echo "Alguno de los parametros dados no existe"

fi

else

echo "Numero de paramentros incorrecto"

fi

**Ejercicio 9.** Hacer un script en Bash denominado newdirfiles con los siguientes tres argumentos:

* <dirname> Nombre del directorio que, en caso de no existir, se debe crear para alojar en el los archivos que se han de crear.
* <num\_files> Número de archivos que se han de crear.
* <basefilename> Será una cadena de caracteres que represente el nombre base de los archivos.
* Ese guión debe realizar lo siguiente:
* Comprobar que el número de argumentos es el correcto y que el segundo argumento tenga un valor comprendido entre 1 y 99.
* Crear, en caso de no existir, el directorio dado en el primer argumento a partir del directorio donde se esté situado y que posea permisos de lectura y escritura para el usuario $USER.
* Dentro del directorio dado en el primer argumento, crear archivos cuyos contenidos estarán vacíos y cuyos nombres lo formarán el nombre dado como tercer argumento y un número que irá desde 01 hasta el número dado en el segundo argumento.