

EJERCICIO DE REPASO DEL MÓDULO 1

PRÁCTICAS 1 a 4

Nombre: Jesús Muñoz Velasco

Escriba en cada espacio la orden u órdenes necesarias para realizar el ejercicio.

1.- A partir del directorio personal y utilizando una única orden, cree un directorio llamado repaso, dentro de él otros dos directorios llamados prueba1 y prueba2, y dentro de prueba1 otro directorio llamado ejercicio1.

```
mkdir -p repaso/prueba1/ejercicio1 repaso/prueba2
```

2.- Situados en el directorio ejercicio1, con una única orden, cree cinco archivos llamados arch1.txt, arch2.txt, arch3.txt, arch4.pdf y arch5.pdf

```
touch arch1.txt arch2.txt arch3.txt arch4.txt arch5.txt
```

3.- Desde el directorio ejercicio1, redireccione al archivo arch1.txt el resultado de buscar la palabra “expresiones” en la ayuda (help) de la orden let

```
help let | grep "expresiones" >> arch1.txt
```

4.- Desde el directorio personal, copie los ficheros de ejercicio1 cuya extensión tenga una x al directorio prueba2.

```
cp repaso/prueba1/ejercicio1/*.*x* repaso/prueba2
```

5.- Nuevamente desde el directorio personal, obtenga las 15 primeras líneas del manual de la orden find y añádalo al final del archivo arch1.txt del directorio prueba2.

```
man find | head -15 >> repaso/prueba2/arch1.txt
```

6.- Exprese la siguiente operación aritmética e indique su resultado: $2+3+4+5/2$ con y sin decimales.

```
(sin decimales) printf %[2+3+4+5/2]"\n"  
(con decimales) echo "2+3+4+5/2" | bc -l
```

7.- En la carpeta personal, construya un guión llamado repaso que admita dos argumentos, el primero será un directorio y el segundo será un archivo:

```
repaso <nombre_directorio> <nombre_archivo>
```

El guión deberá hacer lo siguiente:

- Comprobar que el número de argumentos es correcto, y si no fuera así, indicar cómo se debe ejecutar para que funcione correctamente. Esto es, una pequeña ayuda y terminar.
- Si el segundo argumento es un archivo, escribir las tres primeras líneas en otro archivo llamado salida y mostrar un mensaje informando de dicha acción. En caso contrario informar del error y salir.
- Si el primer argumento es un directorio, mostrar un mensaje con el número de archivos que contiene. En caso contrario crear un directorio con el nombre del primer argumento.
- Buscar en el directorio pasado como primer argumento los archivos que tienen permiso de lectura y añadir el resultado de la búsqueda al final del archivo salida. Además, mostrar por pantalla el siguiente mensaje: “De un total de N archivos, hay X archivos con permiso de lectura” (siendo N el número total de archivos del directorio y X los que tienen permiso de lectura).

```
1 #!/bin/bash
2 # Título: Ejercicio 7
3 # Fecha: 13/4/2023
4 # Autor: Jesús Muñoz Velasco
5 # Versión: 1.0
6 # Descripción: Ejercicio 7 de la relación de repaso
7 # Opciones: 2 argumentos: directorio y archivo
8
9 printf "\n"
10
11 if [ $# != 2 ]; then
12     printf "El número aportado de argumentos no es correcto\n"
13     printf "Se han aportado $# argumentos\n"
14     printf "Los argumentos deberán ser repaso <nombre_directorio> <nombre_archivo>\n"
15 else
16     if test -f $2; then
17         printf "Se escribirán las tres primeras líneas en otro archivo\n"
18
19         head -3 $2 > salida.fs
20     fi
21
22     if test -d $1; then      #se mostrará el número de archivos que tiene
23         printf "El directorio \"$1\" contiene $(ls $1 | wc -l) archivos\n"
24     else
25         printf "Se creará un directorio llamado $1\n"
26         #mkdir $1
27     fi
28
29     printf "\nArchivos con permiso de lectura:\n" >> salida.fs
30
31     find . -perm -444 >> salida.fs
32
33     N=$(ls . | wc -l)
34     X=$[$(find . -perm -444 | wc -l)-1]
35
36
37     printf "De un total de $N archivos hay $X archivos con permiso de lectura\n"
38 fi
39
40 printf "\n"
41
```

(Adjunto archivo)

EJERCICIO REPASO PRÁCTICAS 1 A 5 – MÓDULO 1

1.- Cree tres directorios con una única orden: curso_1920 y dentro de éste, cuatrimestre_1 y cuatrimestre_2.

```
mkdir -p curso_1920/cuatrimestre_{1,2}
```

2.- Cree un archivo cuyo nombre sea repasoMod1 en el directorio curso_1920. A continuación, edítelo y escriba lo siguiente:

DNI: número del DNI
Nombre y apellidos: nombre y apellidos
Fecha: fecha actual (dd/mm/aaaa)

```
touch curso_1920/repassoMod1.fs  
echo DNI: 76668409W\n >> curso_1920/repassoMod1.fs  
echo Nombre y apellidos: Jesús Muñoz Velasco >> curso_1920/repassoMod1.fs  
echo Fecha: `date +"%d/%m/%Y"` >> curso_1920/repassoMod1.fs
```

3.- Copie el archivo repasoMod1 del directorio curso_1920 en el directorio cuatrimestre_1.

```
cp curso_1920/repassoMod1.fs curso_1920/cuatrimestre_1
```

4.- Redireccione el resultado de ejecutar la orden que obtiene la fecha actual al fichero que ha creado en el punto 2 (en el directorio curso_1920) sin destruir su contenido.

```
echo Fecha: `date +"%d/%m/%Y"` >> curso_1920/repassoMod1.fs
```

5.- Indique qué es necesario hacer para crear una variable RUTA que contenga la ruta absoluta del directorio en el que nos encontramos. Compruebe con el comando correspondiente las variables locales de su sistema.

```
RUTA=$(pwd)  
set #para mostrar las variables
```

6.- Obtenga las 20 últimas líneas del manual de la orden find y guárdelas en un fichero llamado ayudafind, dentro del directorio cuatrimestre_2.

```
man find | tail -20 >> curso_1920/cuatrimestre_2/ayudafind.fs
```

7.- En la carpeta personal, construya un guión llamado repaso-guion que admita tres argumentos, el primero será un directorio, el segundo una letra, y el tercero una palabra:

examen-guion <nombre_directorio> <letra> <palabra>

El guión debe:

- Comprobar que se le están pasando tres argumentos, y si no fuera así, indicar cómo se debe ejecutar para que funcione correctamente. Esto es, dar una pequeña ayuda y terminar.
- Crear, en caso de no existir, el directorio dado como primer argumento.
- Comprobar que el segundo argumento sea una f o una d. Si no fuera así, indicarlo con un mensaje de error y salir.
- Encontrar la palabra dada en el tercer argumento de forma recursiva en los archivos existentes a partir del directorio indicado por el usuario (primer argumento), incluyendo el número de línea en donde se localiza el término. Almacenar el resultado de la consulta en un fichero llamado resultado dentro del directorio desde el que se ejecute el guión.
- Buscar en el directorio indicado por el usuario (primer argumento) todos los ficheros o directorios según la letra pasada como segundo argumento (f=fichero, d=directorio) Almacenar el resultado de la consulta al final del fichero resultado del apartado anterior.

```
1 #!/bin/bash
2 # Título: Ejercicio 7
3 # Fecha: 14/4/2023
4 # Autor: Jesús Muñoz Velasco
5 # Versión: 1.0
6 # Descripción: Ejercicio 7 de la relación de repaso 2
7 # Opciones: 3 argumentos: directorio, letra y palabra
8
9 if [ $# != 3 ]; then
10   printf "El número aportado de argumentos no es correcto\n"
11   printf "Se han aportado $# argumentos\n"
12   printf "Los argumentos deberán ser repaso-guion.fs <nombre_directorio> <letra> <palabra>\n"
13 else
14   test -e $1 || mkdir $1
15
16   if [[ $2 =~ ^[df]$ ]]; then
17
18     printf "Resultado de la búsqueda de la palabra \"$3\" en el directorio \"$1\":\n" >> resultado.fs
19     grep -R -n $3 $1 > resultado.fs
20
21     find $1 -type $2 >> resultado.fs
22
23   else
24     printf "El argumento $2 no se corresponde con la letra d ni f\n"
25   fi
26
27 fi
28
```

(Adjunto Archivo)