

Manual script 3

Creación de grupos y usuarios:

INDICE:

Introducción:	1
Uso del script:.....	1
1. Ejecución del script:	1
Archivo de entrada:	1
2. Estructura de código:	3
Funciones:	3

Introducción:

Este script de Bash está diseñado para facilitar la creación de grupos y usuarios en un sistema Linux. Proporciona una interfaz sencilla para procesar un archivo de entrada que contiene información sobre los grupos y la cantidad de usuarios a crear para cada grupo.

Uso del script:

1. Ejecución del script:

Para ejecutar el script, debes de abrir la terminal, navegar al directorio donde se encuentra el script y ejecutar el comando «./script3.sh <archivo>» siendo <archivo> el archivo que contiene la información de grupos y usuarios.

(en este caso el archivo se llama “grupos”):

```
esteban@DESKTOP-8VR3M0M: ~/enlace-scripts/practicas/practica3$ bash script3.sh grupos
```

Archivo de entrada:

El script necesita un archivo de entrada que contenga en cada línea texto con el siguiente formato: **nombreGrupo:cantidadUsuarios**

En donde “nombreGrupo” es el nombre del grupo que se creará y “numeroUsuarios” la cantidad de usuarios que se deben crear para ese grupo.

En este caso, el archivo contiene lo siguiente:

```
GNU nano 6.2 grupos
asir2:3
peepo:2
```

Durante la ejecución, el script:

- Verificará la cantidad correcta de argumentos proporcionados.
- Comprobará si el archivo de entrada es un fichero regular y existe.
- Creará grupos si no existen.
- Creará usuarios para cada grupo, eliminando previamente los usuarios existentes si es necesario, con el formato '**<grupo>-<numeroUsuario>**' (ej: asir2-1 , asir2-2, etc.)

Si ejecutamos el script con el ejemplo que indicamos anteriormente tendremos los siguiente:

```
esteban@DESKTOP-8VR3M0M: ~/enlace-scripts/practicas/practica3$ bash script3.sh grupos|
-----
Se ha creado el grupo asir2
-----
Se ha creado el usuario 'asir2-1'
-----
Se ha creado el usuario 'asir2-2'
-----
Se ha creado el usuario 'asir2-3'
-----
Se ha creado el grupo peepo
-----
Se ha creado el usuario 'peepo-1'
-----
Se ha creado el usuario 'peepo-2'
-----
esteban@DESKTOP-8VR3M0M: ~/enlace-scripts/practicas/practica3$ |
```

Como podemos ver, se cumple lo del archivo “grupos”; se crearon los grupos ‘asir2’ y ‘peepo’, tres usuarios para el grupo ‘asir2’ y dos para el grupo ‘peepo’.

Si **volvemos a ejecutar el script** con los usuarios ya creados, tendremos lo siguiente:

```
Usuario asir2-1 eliminado junto a su directorio home.
-----
Se ha creado el usuario 'asir2-1'
-----
Usuario asir2-2 eliminado junto a su directorio home.
-----
Se ha creado el usuario 'asir2-2'
-----
Usuario asir2-3 eliminado junto a su directorio home.
-----
Se ha creado el usuario 'asir2-3'
-----
Usuario peepo-1 eliminado junto a su directorio home.
-----
Se ha creado el usuario 'peepo-1'
-----
Usuario peepo-2 eliminado junto a su directorio home.
-----
Se ha creado el usuario 'peepo-2'
-----
esteban@DESKTOP-8VR3M0M: ~/enlace-scripts/practicas/practica3$ |
```

Esta vez antes de crear el usuario, se eliminará. El grupo se mantiene si ya existe, por lo que no hay mensaje.

2. Estructura de código:

El script está organizado en secciones mediante el uso de funciones, esto para facilitar la comprensión y mantenimiento del código.

Funciones:

- **compruebaArgumentos():** Verifica la cantidad correcta de argumentos.

```
# Verifica si se proporciona el número correcto de argumentos
compruebaArgumentos(){
    if [ $1 -ne 1 ];then
        echo -e "$rojo-----$reset"
        echo "ERROR. Debes ingresar el nombre del archivo."
        echo -e "$rojo-----$reset"
        exit 1
    fi
}
```

- **compruebaArchivo():** Verifica si el archivo de entrada es válido.

```
# Verifica si el archivo de entrada es un fichero regular y existe
compruebaArchivo(){
    if [ ! -f $1 ];then
        echo -e "$rojo-----$reset"
        echo "No es un fichero regular o no existe."
        echo -e "$rojo-----$reset"
        exit 2
    fi
}
```

- **creaUsuario():** Crea usuarios para un grupo específico.
Verifica que el usuario no exista, buscando el nombre en el archivo **/etc/passwd**, si existe le elimina. Si no existe, lo crea directamente.

```
# Crea usuarios para un grupo específico
creaUsuario(){
    for ((i=1;i<=$1;i=i+1));do
        nuevoUser="$2-$i"

        # Elimina el usuario existente junto con su directorio home
        if cat /etc/passwd | grep -e "^$nuevoUser:" &>/dev/null ;then
            sudo userdel -r "$nuevoUser" &>/dev/null
            echo -e "$rojo-----$reset"
            echo "Usuario $nuevoUser eliminado junto a su directorio home."
            echo -e "$rojo-----$reset"
        fi

        # Crea un nuevo usuario y lo agrega al grupo correspondiente
        sudo useradd "$nuevoUser" -g "$2" &> /dev/null
        echo -e "$verde-----$reset"
        echo "Se ha creado el usuario '$nuevoUser'"
        echo -e "$verde-----$reset"
    done
}
```

- **creaGrupo():** Crea un grupo si no existe.
Verifica si el grupo no existe, buscándolo en el archivo **/etc/group**. Si no existe, lo crea.

```
# Crea un grupo si no existe
creaGrupo(){
    if ! cat /etc/group | grep -e "^$1:" &>/dev/null;then
        sudo groupadd "$1"
        echo -e "$cyan-----$reset"
        echo "Se ha creado el grupo $1"
        echo -e "$cyan-----$reset"
    fi
}
```

Finalmente, las funciones principales que crean los grupos y usuarios se ejecutan por medio de un “while”, que recorre todas las líneas del archivo ingresado y asigna los valores correspondientes al grupo y al usuario a una variable, para así entregárselas a la función.

```
# Ejecución de funciones

# Verifica la cantidad de argumentos proporcionados al script
compruebaArgumentos $#

# Verifica si el archivo de entrada es válido
compruebaArchivo $1

# Procesa cada línea del archivo de entrada
while read linea;do
    grupo=$(echo $linea | cut -d ':' -f 1)
    usuario=$(echo $linea | cut -d ':' -f 2)

    # Crea el grupo si no existe
    creaGrupo $grupo

    # Crea usuarios para el grupo
    creaUsuario $usuario $grupo
done < $1 # Lee las líneas del archivo ingresado ($1)
```