



Instituto Tecnológico Las Américas
(ITLA)

Sistemas Operativos 3 (SO3)

Daniel Alejandro Moreno Martínez

Matrícula: 2010-2946



How to

!!! How to: Creación de Scripts !!!

Script

¿Qué es un Shell script en Linux?

Shell es un programa de usuario, o su entorno provisto para la interacción del usuario. Shell es un intérprete de lenguaje de comando que ejecuta comandos de lectura de entrada por medio del dispositivo estándar (teclado) o desde un archivo.

Shell no es parte del núcleo del sistema, pero utiliza el núcleo del sistema para ejecutar programas, crear archivos, etc. Varios shell disponibles con Linux, son: **BASH (Bourne-Again Shell)**, **CSH (C Shell)**, **KSH (Korn Shell)** y **TCSH**.

Un Script es un archivo de texto, que contiene una serie de comandos para shell, que el sistema ejecuta ordenadamente, de arriba abajo.

Para editarlos, tan solo hace falta un editor de textos, como Emacs, o Vi. Se guardan con extensión: **.sh** y se ejecutan desde la Shell mediante el comando: **source nombre de script.sh**. Los scripts, se comportan de igual manera que los comandos de la shell.

Un Script es un archivo de texto, que contiene una serie de comandos para shell, que el sistema ejecuta ordenadamente, de arriba abajo.

Para editarlos, tan solo hace falta un editor de textos, como Emacs, o Vi. Se guardan con extensión: **.sh** y se ejecutan desde la Shell mediante el comando: **source nombre de script.sh**. Los scripts, se comportan de igual manera que los comandos de la shell.

¿Cómo Funciona un Script en Linux?



La Shell lee la entrada estándar de la terminal, y manda la salida estándar y los errores estándar a la terminal al menos que se le haya dicho que lo mande a otro lugar.

La shell es de línea orientada, no procesa los comandos hasta que pulse **<Enter>** para indicar el final de una línea. Usted puede corregir lo que ha escrito mientras digita. Diferentes shell proporcionan diferentes instalaciones para la edición de sus comandos, pero por lo general reconocen **<BKSP>** o **<Supr>** como la pulsación de tecla para borrar los caracteres anteriores.

Cuando se presiona **< Enter >**, el shell interpreta la línea que ha introducido, antes de ejecutar los comandos en esa línea. Los pasos que se ejecuta a través son los siguientes:

- **La shell se divide** la línea en fichas. Una ficha es un comando, variable, u otro símbolo reconocido por la shell. Se continúa la construcción de una secuencia de ficha, hasta que llega a una palabra reservada (un comando de shell interno que regula el flujo de control de un script de shell), nombre de la función, o el operador (un símbolo que denota una pipa, una condición lógica, un separador de comandos, o alguna otra operación que no puede llevarse a cabo hasta que se evaluó el comando anterior).

- **La Shell organiza** las fichas en tres categorías:
 - **I / O redirección;** comandos que determinan la entrada o salida de un programa y a donde se dirigen. Por ejemplo, en la siguiente línea de comandos, el texto `> listfile` se interpreta como una redirección de la salida, que luego se aplica el comando anterior: `> Ls-al listfile`
 - **Asignación** de variables, la shell puede reconocer los comandos que asignan un valor a una variable.
 - **Varios comandos**, otras fichas son revisados para ver si son alias. La primera palabra está revisada. Si se trata de un alias, se sustituye por el significado original del alias, si no es un alias, o si es seguido por un espacio en blanco antes de la siguiente palabra, el proceso de comprobación de alias se repite hasta que ya no quedan más palabras (o hasta que un alias se ha detectado que no es seguido por un espacio).
- **Puede ser que lo comando** hayan sido ejecutados, ya sea como comandos de la shell interno (que hacen que el propio shell tome alguna acción) o, si no son comandos internos, como los programas externos (si la shell puede localizar un archivo ejecutable de ese nombre).

¿Cómo los identifica el sistema?

Los archivos script suelen ser identificados por el sistema a través de uno de los siguientes encabezamientos en el contenido del archivo, conocido como **shebang**:

#!/bin/bash ; #!/bin/ksh ; #!/bin/csh

Aunque en entornos UNIX la mayoría de los scripts son identificados por dicho encabezamiento, también pueden ser identificados a través de la extensión **".sh"**, siendo ésta quizá menos importante que el encabezamiento, ya que casi todos los sistemas no necesitan dicha extensión para ejecutar el script, por lo tanto, esta suele ser

añadida por tradición, o más bien, es útil para que el usuario pueda identificar estos archivos a través de una interfaz de línea de comandos sin necesidad de abrirlo.

Difieren de los programas de aplicación, debido a que los últimos son más complejos; además, los scripts son más bien un programa que le da instrucciones a otros más avanzados.

¿Cómo ejecutar un script?

Una vez guardado el archivo con el nombre deseado, tenemos dos formas de ejecutarlos:

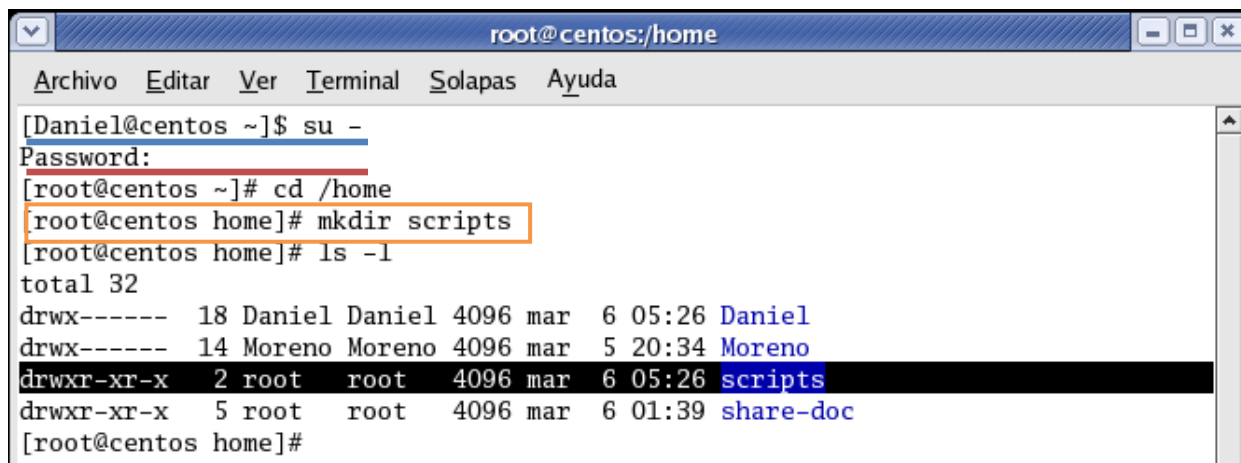
- **Ejecutar el comando `source fichero`.** En esta forma de ejecutarlo no tenemos que darle permiso de ejecutar. El cual carga el fichero en la memoria del **Bash y lo ejecuta.**
- La segunda forma, a diferencia del anterior no implica darle permiso de ejecución al fichero, esto podemos hacerlo con el comando **`chmod+X +nombre de fichero`**. Poniéndole este permiso, podremos ejecutarlo siempre y cuando este en algunos de los directorios indicados en la variables de entorno PATH. **Luego escribimos `sh`** y el nombre del fichero. **Donde `sh` es la orden encargada de ejecutar el script.**

Crear un script para hacer una calculadora

Después de haber adquirido **los conocimientos básicos** sobre Shell script podemos proceder a comenzar a verlos en función; en este caso se va a crear un script para que sea capaz de sumar, restar, multiplicar y dividir.

Existen 2 formas de hacer los scripts, una es mediante cualquier editor del Shell como: nano, cat, vi, etc, y la otra forma es haciéndolo en un editor de palabras como **KWrite**, **gedit Text Editor**, etc.

En nuestro caso lo haremos de la primera forma. Lo primero que debemos hacer es abrir una terminal.



```
root@centos:/home
Archivo  Editar  Ver  Terminal  Solapas  Ayuda
[Daniel@centos ~]$ su -
Password:
[root@centos ~]# cd /home
[root@centos home]# mkdir scripts
[root@centos home]# ls -l
total 32
drwx----- 18 Daniel Daniel 4096 mar  6 05:26 Daniel
drwx----- 14 Moreno Moreno 4096 mar  5 20:34 Moreno
drwxr-xr-x  2 root  root  4096 mar  6 05:26 scripts
drwxr-xr-x  5 root  root  4096 mar  6 01:39 share-doc
[root@centos home]#
```

Aquí, lo primero que hicimos fue pasar al **modo root** con el comando **su -**

Luego nos pide que introduzcamos el **password del root**.

Después, procedemos a crear el directorio **scripts** que es donde vamos a guardar los **scripts** que crearemos.

A continuación veremos algunas variables autodefinidas por shell-

\$ ---- indica la posición del argumento cuando el script es llamado.

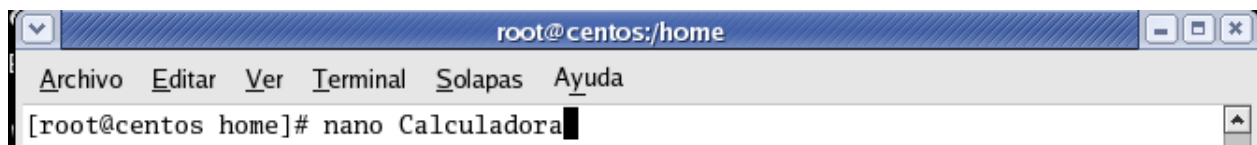
\$# --- Esta variable contiene el número de argumentos que hemos pasado al script.

\$0 --- Esta variable contiene el nombre del shell-script que se ejecuta.

\$* --- Esta variable muestra una cadena con los parámetros que hemos pasado al script.

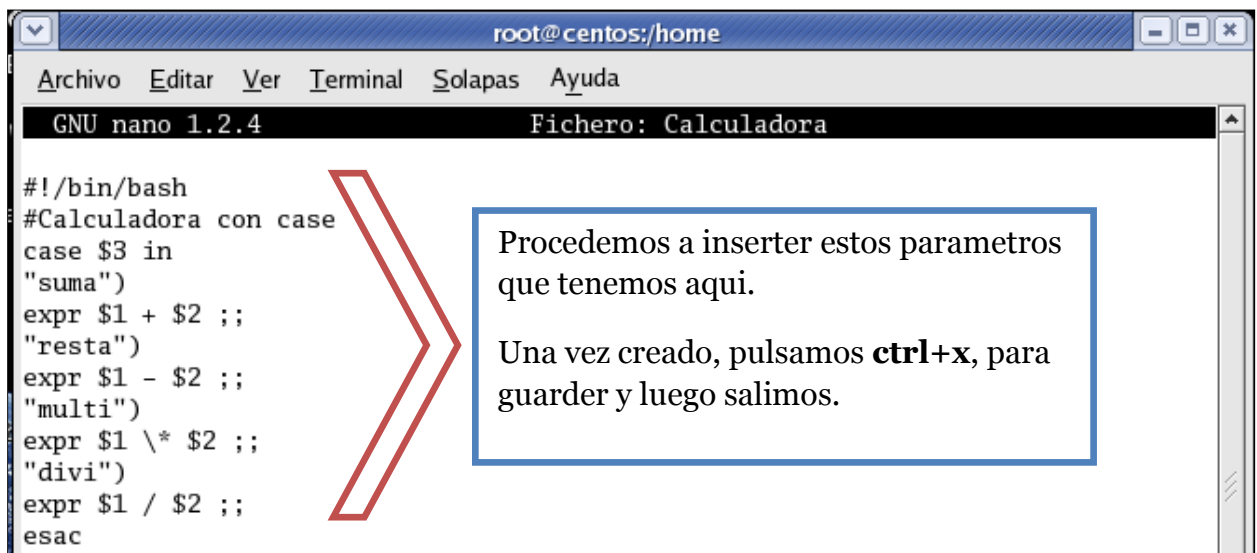
\$? --- Esta variable contiene la salida del comando anterior donde se encuentre dentro del script.

Una vez dentro del directorio scripts que fue el que creamos anteriormente, **procedemos a crear el fichero o script**, como dijimos antes esto se puede lograr con cualquier editor de fichero disponible, en este caso utilizaremos **nano**, usamos el comando **nano Calculadora**, porque el fichero a crear será una calculadora.



```
root@centos:/home
Archivo Editar Ver Terminal Solapas Ayuda
[root@centos home]# nano Calculadora
```

Después de haber presionado la tecla **ENTER** tenemos lo siguiente:

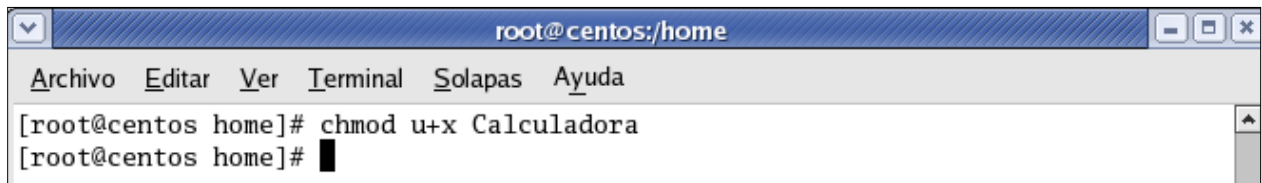


```
root@centos:/home
Archivo Editar Ver Terminal Solapas Ayuda
GNU nano 1.2.4 Fichero: Calculadora
#!/bin/bash
#Calculadora con case
case $3 in
"suma")
expr $1 + $2 ;;
"resta")
expr $1 - $2 ;;
"multi")
expr $1 \* $2 ;;
"divi")
expr $1 / $2 ;;
esac
```

Procedemos a insertar estos parametros que tenemos aqui.

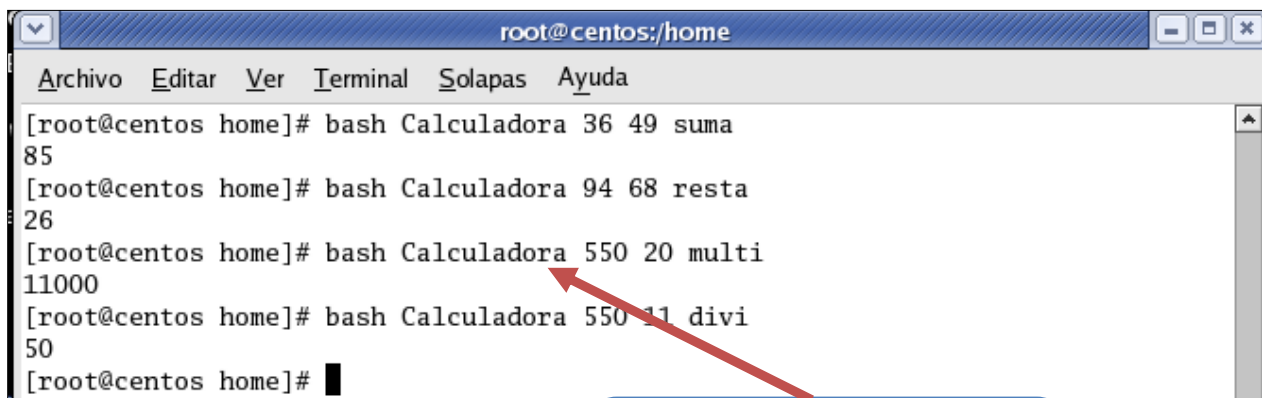
Una vez creado, pulsamos **ctrl+x**, para guardar y luego salimos.

Por último vamos a darle **permisos de ejecución**, con el comando **chmod u+x Calculadora**



```
root@centos:/home
Archivo Editar Ver Terminal Solapas Ayuda
[root@centos home]# chmod u+x Calculadora
[root@centos home]#
```

Al ejecutar nuestra nueva calculadora con la mitad de **líneas gracias a case**, tenemos que **case** nos permite dar *múltiples valores a una variable* y actuar de la forma que sea, según el valor que introduzcamos.

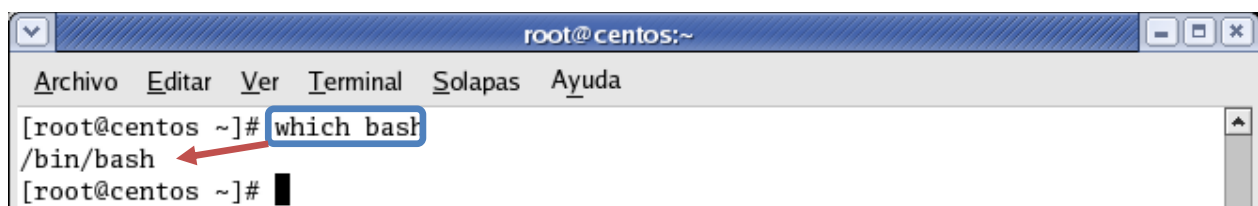


```
root@centos:/home
Archivo Editar Ver Terminal Solapas Ayuda
[root@centos home]# bash Calculadora 36 49 suma
85
[root@centos home]# bash Calculadora 94 68 resta
26
[root@centos home]# bash Calculadora 550 20 multi
11000
[root@centos home]# bash Calculadora 550 11 divi
50
[root@centos home]#
```

Aquí podemos observar nuestra calculadora funcionando correctamente.

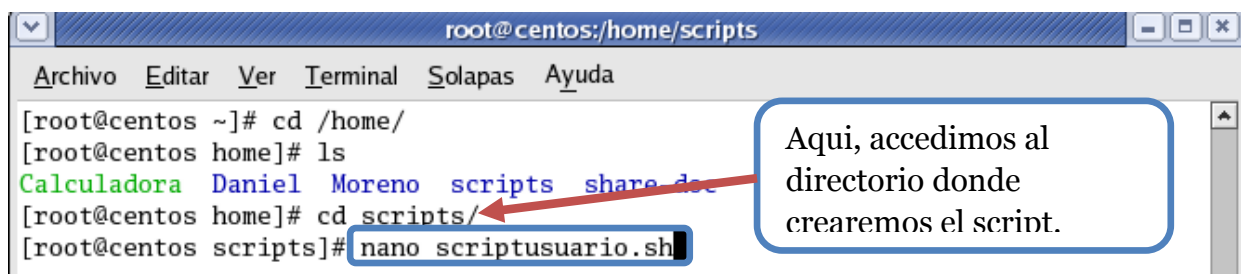
Script para crear usuarios

Para hacer un script que nos permita crear usuarios necesitaremos dos comandos importantes, los cuales son: **useradd** o **adduser**. Otra cosa que debemos tener en cuenta es que **bash utiliza nuestro kernel**. Con el comando **which bash** nos damos cuenta del bash que utiliza nuestro kernel. Procedemos a abrir una terminal.



```
root@centos:~  
Archivo Editar Ver Terminal Solapas Ayuda  
[root@centos ~]# which bash  
/bin/bash  
[root@centos ~]#
```

Después de esto podemos empezar a crear nuestro script. En este caso utilizaremos **nano para crear el script**, y el nombre del archivo será **scriptusuario.sh**.



```
root@centos:/home/scripts  
Archivo Editar Ver Terminal Solapas Ayuda  
[root@centos ~]# cd /home/  
[root@centos home]# ls  
Calculadora Daniel Moreno scripts share doc  
[root@centos home]# cd scripts/  
[root@centos scripts]# nano scriptusuario.sh
```

Aquí, accedimos al directorio donde crearemos el script.

Para crear el script escribimos:

#!/bin/bash: este es el bash que estamos utilizando.

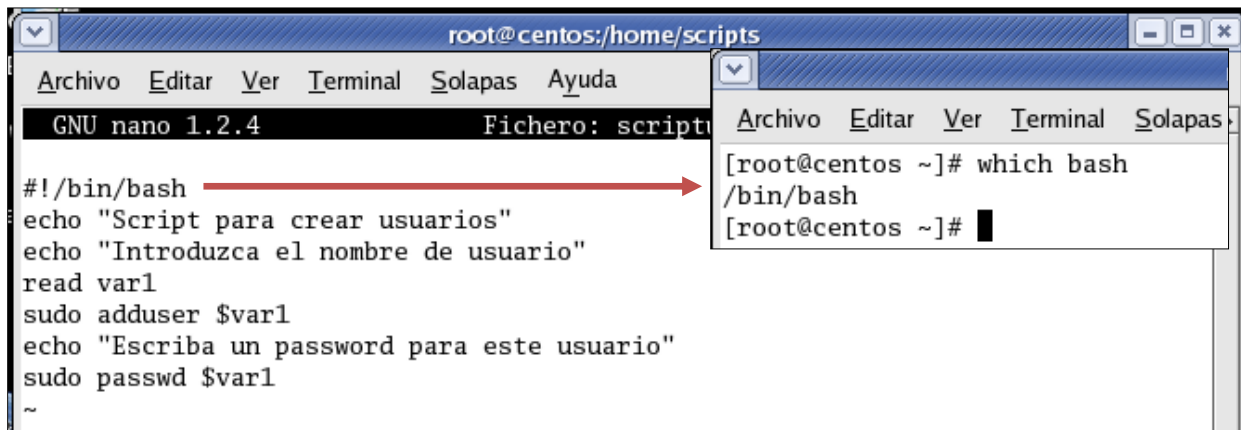
Script Funcional: es la etiqueta de nuestro script.

Escribe el nombre de usuario: que es un mandato.

Var 1: es una variable.

Echo: nos dará print de mensaje que pongamos entre comillas.

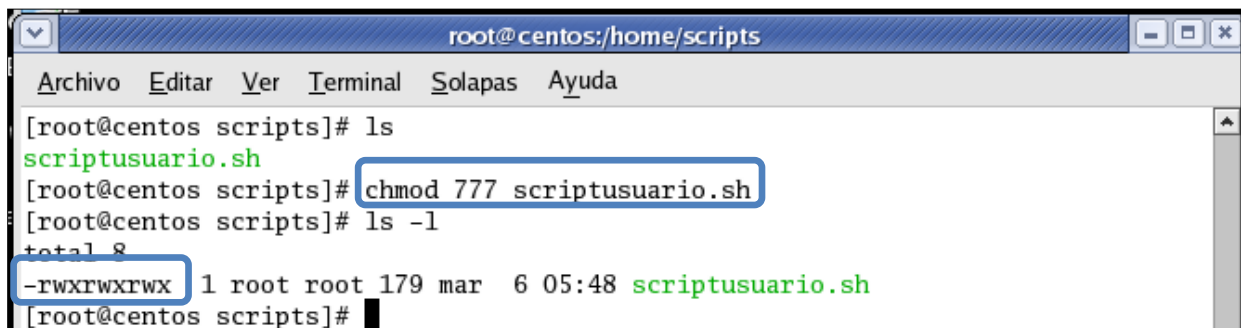
Read: leerá los datos que introduzca el usuario y los guardará en las variables.



```
root@centos:/home/scripts
GNU nano 1.2.4 Fichero: scriptu
#!/bin/bash
echo "Script para crear usuarios"
echo "Introduzca el nombre de usuario"
read var1
sudo adduser $var1
echo "Escriba un password para este usuario"
sudo passwd $var1
~

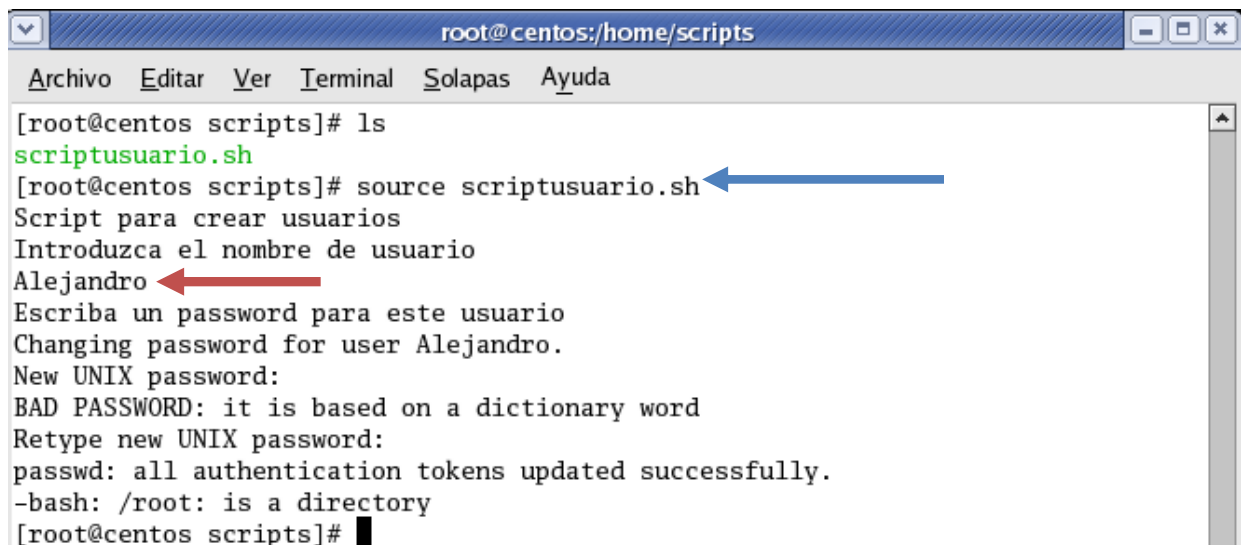
[root@centos ~]# which bash
/bin/bash
[root@centos ~]#
```

Luego guardamos los cambios presionando **ctrl+x** y **cerramos**, ahora procedemos a dar los permisos necesarios para la ejecución del script.



```
root@centos:/home/scripts
Archivo Editar Ver Terminal Solapas Ayuda
[root@centos scripts]# ls
scriptusuario.sh
[root@centos scripts]# chmod 777 scriptusuario.sh
[root@centos scripts]# ls -l
total 8
-rwxrwxrwx 1 root root 179 mar  6 05:48 scriptusuario.sh
[root@centos scripts]#
```

Utilizamos el comando **source** para ejecutar el script.

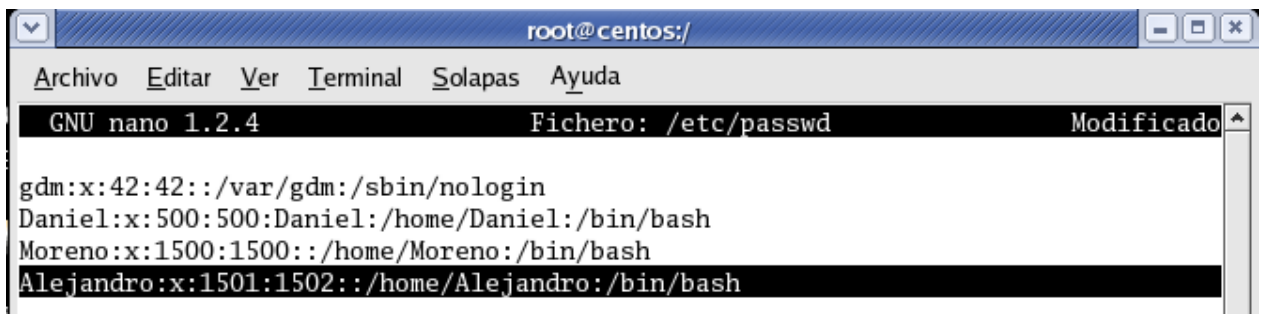


```
root@centos:/home/scripts
Archivo Editar Ver Terminal Solapas Ayuda
[root@centos scripts]# ls
scriptusuario.sh
[root@centos scripts]# source scriptusuario.sh
Script para crear usuarios
Introduzca el nombre de usuario
Alejandro
Escriba un password para este usuario
Changing password for user Alejandro.
New UNIX password:
BAD PASSWORD: it is based on a dictionary word
Retype new UNIX password:
passwd: all authentication tokens updated successfully.
-bash: /root: is a directory
[root@centos scripts]#
```

Como podemos ver hemos creado el usuario, y hemos asignado una contraseña. Para comprobar que nuestro usuario ha sido creado nos dirigimos al siguiente archivo: **/etc/passwd**.



```
root@centos:/  
Archivo Editar Ver Terminal Solapas Ayuda  
[root@centos /]# nano /etc/passwd
```

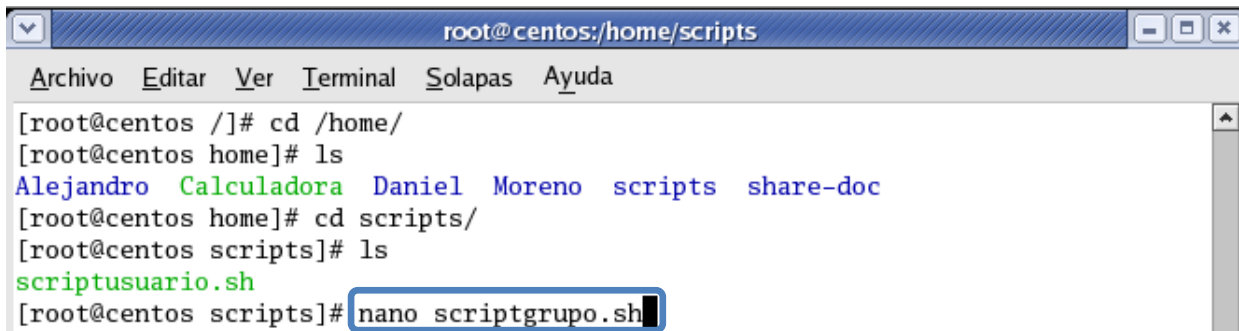


```
root@centos:/  
Archivo Editar Ver Terminal Solapas Ayuda  
GNU nano 1.2.4 Fichero: /etc/passwd Modificado  
gdm:x:42:42:./var/gdm:/sbin/nologin  
Daniel:x:500:500:Daniel:/home/Daniel:/bin/bash  
Moreno:x:1500:1500:./home/Moreno:/bin/bash  
Alejandro:x:1501:1502:./home/Alejandro:/bin/bash
```

Script para crear grupos

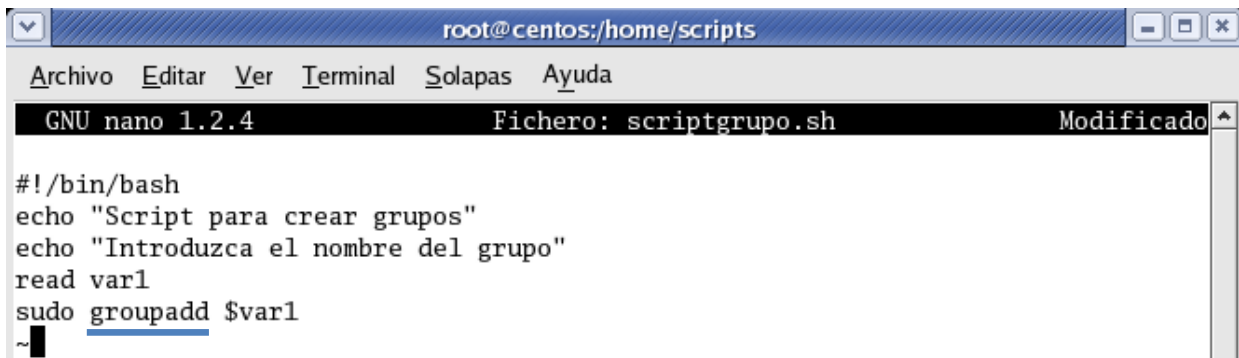
Los pasos para crear un grupo a través de un script son similares a la práctica anterior.

En primer lugar nos ubicamos en el directorio que queremos **crear el script** y editamos el archivo **del script**, en nuestro caso utilizaremos **nano**, y el script se llamara **scriptgrupo.sh**.



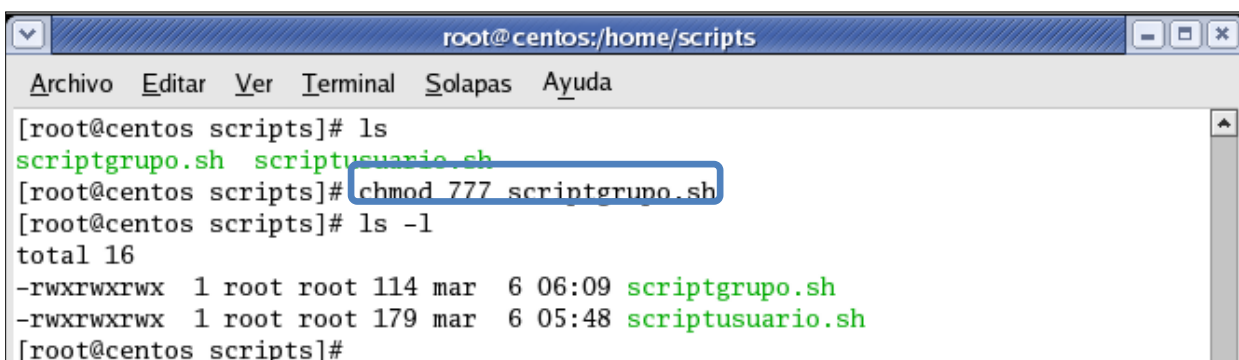
```
root@centos:/home/scripts
Archivo Editar Ver Terminal Solapas Ayuda
[root@centos /]# cd /home/
[root@centos home]# ls
Alejandro Calculadora Daniel Moreno scripts share-doc
[root@centos home]# cd scripts/
[root@centos scripts]# ls
scriptusuario.sh
[root@centos scripts]# nano scriptgrupo.sh
```

Utilizaremos el comando **groupadd** para crear los grupos, y **usermode** para agregar usuarios al grupo creado.



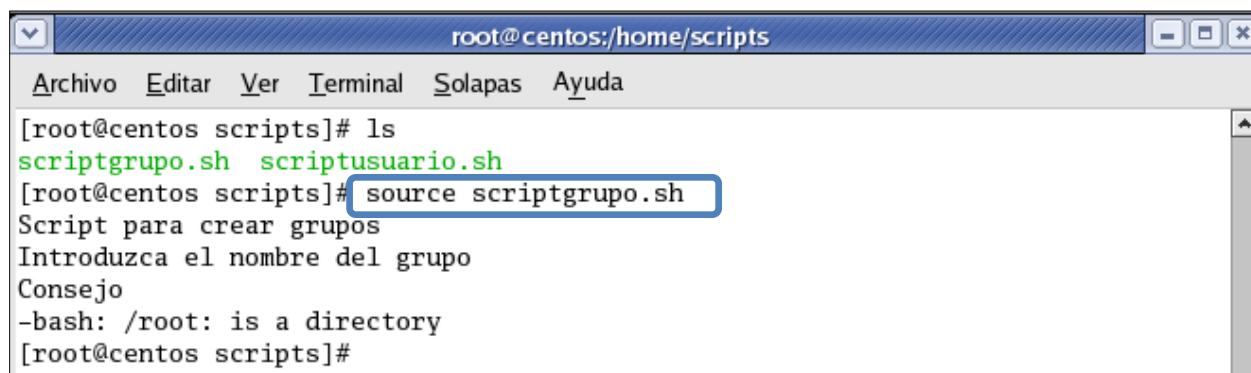
```
root@centos:/home/scripts
GNU nano 1.2.4 Fichero: scriptgrupo.sh Modificado
#!/bin/bash
echo "Script para crear grupos"
echo "Introduzca el nombre del grupo"
read var1
sudo groupadd $var1
~
```

Añadimos los permisos necesarios al script creado:



```
root@centos:/home/scripts
Archivo Editar Ver Terminal Solapas Ayuda
[root@centos scripts]# ls
scriptgrupo.sh scriptusuario.sh
[root@centos scripts]# chmod 777 scriptgrupo.sh
[root@centos scripts]# ls -l
total 16
-rwxrwxrwx 1 root root 114 mar  6 06:09 scriptgrupo.sh
-rwxrwxrwx 1 root root 179 mar  6 05:48 scriptusuario.sh
[root@centos scripts]#
```

Luego utilizamos el comando **source** para ejecutar el **script** y ver su funcionalidad.

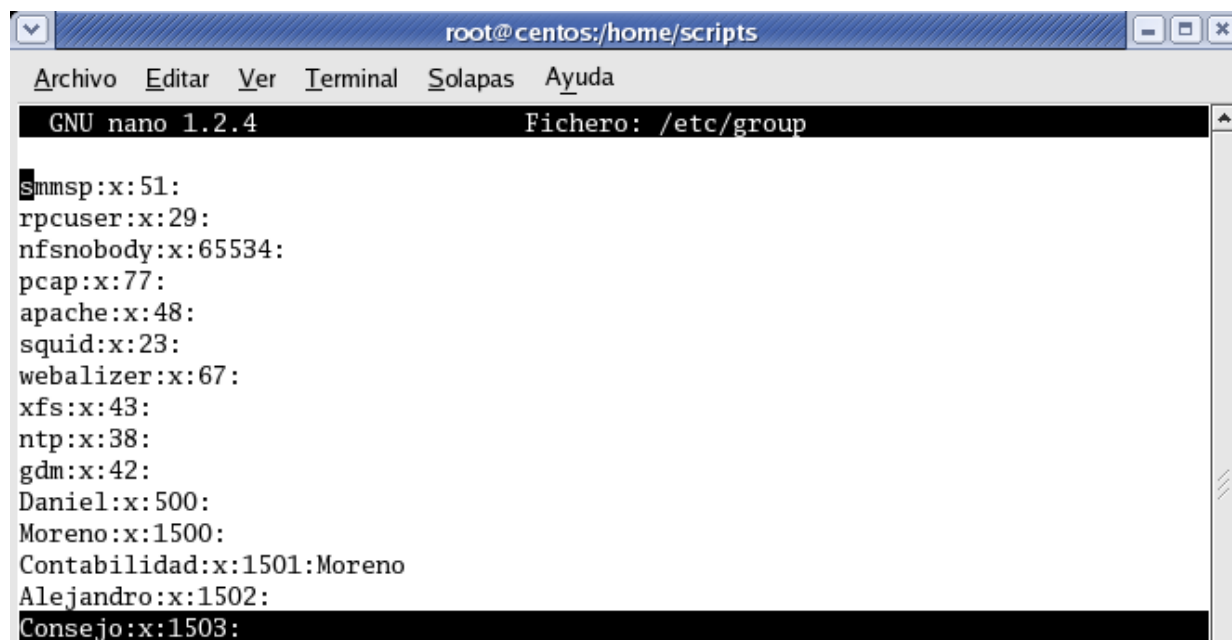


```
root@centos:/home/scripts
Archivo Editar Ver Terminal Solapas Ayuda
[root@centos scripts]# ls
scriptgrupo.sh scriptusuario.sh
[root@centos scripts]# source scriptgrupo.sh
Script para crear grupos
Introduzca el nombre del grupo
Consejo
-bash: /root: is a directory
[root@centos scripts]#
```

Nos dirigiremos al archivo **/etc/group**, para verificar que el grupo ha sido creado.



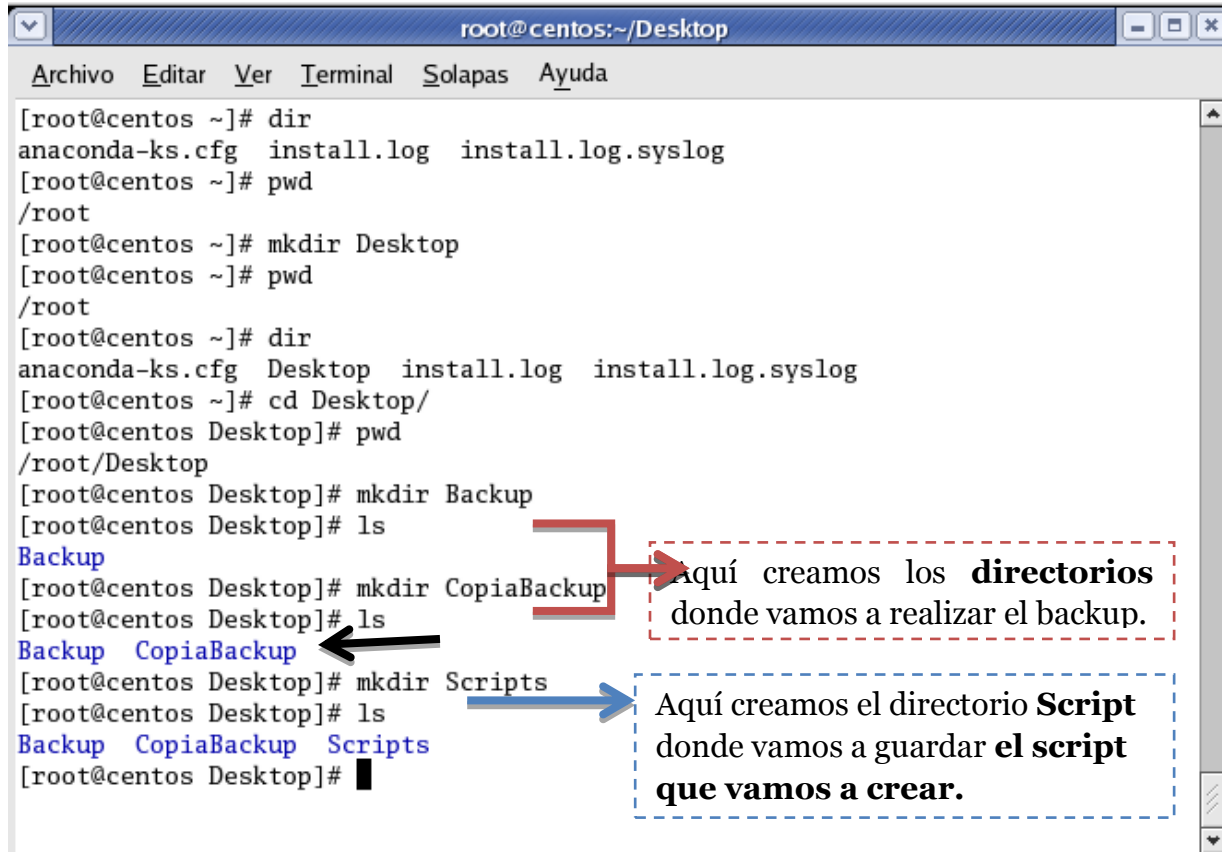
```
root@centos:/home/scripts
Archivo Editar Ver Terminal Solapas Ayuda
[root@centos scripts]# nano /etc/group
```



```
root@centos:/home/scripts
Archivo Editar Ver Terminal Solapas Ayuda
GNU nano 1.2.4 Fichero: /etc/group
smmsp:x:51:
rpcuser:x:29:
nfsnobody:x:65534:
pcap:x:77:
apache:x:48:
squid:x:23:
webalizer:x:67:
xfs:x:43:
ntp:x:38:
gdm:x:42:
Daniel:x:500:
Moreno:x:1500:
Contabilidad:x:1501:Moreno
Alejandro:x:1502:
Consejo:x:1503:
```

Script para hacer backup de datos

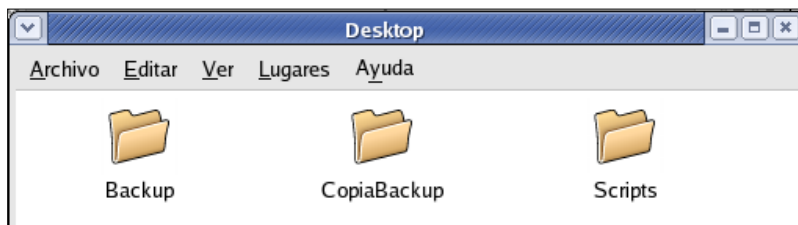
El primer paso es crear un **directorio con un fichero**, el cual será el lugar donde se realizara el **backup**, y otro directorio en donde se realizara la copia del mismo.



```
root@centos:~/Desktop
Archivo  Editar  Ver     Terminal  Solapas  Ayuda
[root@centos ~]# dir
anaconda-ks.cfg  install.log  install.log.syslog
[root@centos ~]# pwd
/root
[root@centos ~]# mkdir Desktop
[root@centos ~]# pwd
/root
[root@centos ~]# dir
anaconda-ks.cfg  Desktop  install.log  install.log.syslog
[root@centos ~]# cd Desktop/
[root@centos Desktop]# pwd
/root/Desktop
[root@centos Desktop]# mkdir Backup
[root@centos Desktop]# ls
Backup
[root@centos Desktop]# mkdir CopiaBackup
[root@centos Desktop]# ls
Backup  CopiaBackup
[root@centos Desktop]# mkdir Scripts
[root@centos Desktop]# ls
Backup  CopiaBackup  Scripts
[root@centos Desktop]#
```

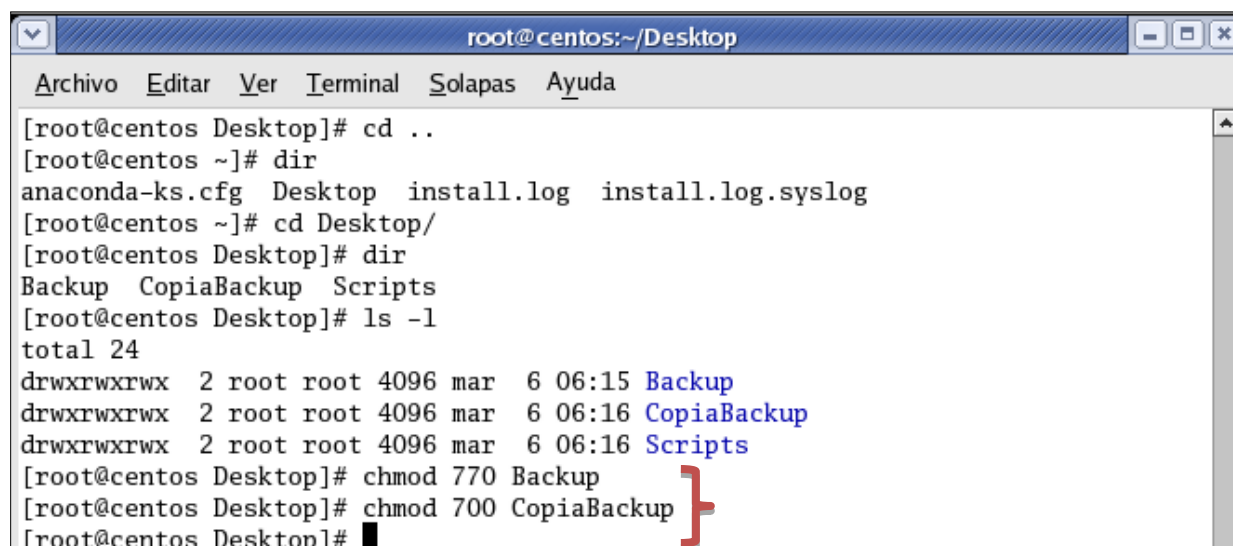
Aquí creamos los **directorios** donde vamos a realizar el backup.

Aquí creamos el directorio **Script** donde vamos a guardar **el script que vamos a crear**.



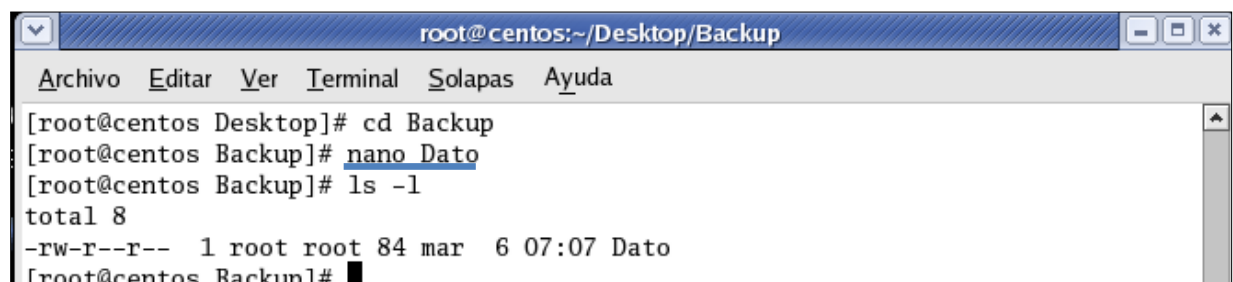
Estos son los directorios que hemos creado!!!

Añadimos los permisos necesarios a los directorios creados.



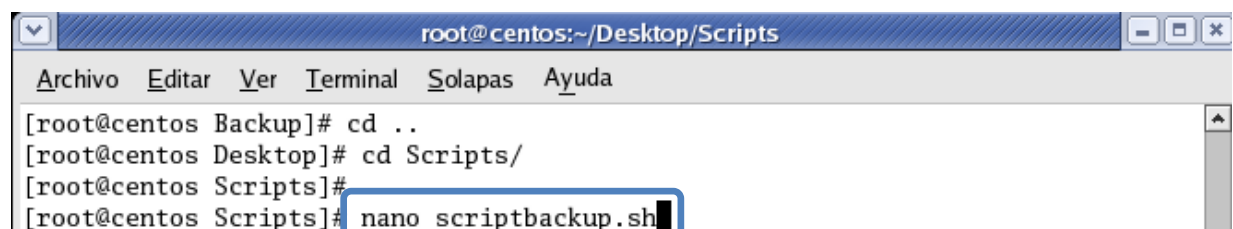
```
root@centos:~/Desktop
Archivo Editar Ver Terminal Solapas Ayuda
[root@centos Desktop]# cd ..
[root@centos ~]# dir
anaconda-ks.cfg Desktop install.log install.log.syslog
[root@centos ~]# cd Desktop/
[root@centos Desktop]# dir
Backup CopiaBackup Scripts
[root@centos Desktop]# ls -l
total 24
drwxrwxrwx 2 root root 4096 mar  6 06:15 Backup
drwxrwxrwx 2 root root 4096 mar  6 06:16 CopiaBackup
drwxrwxrwx 2 root root 4096 mar  6 06:16 Scripts
[root@centos Desktop]# chmod 770 Backup
[root@centos Desktop]# chmod 700 CopiaBackup
[root@centos Desktop]#
```

Vamos a crear un archivo en el directorio **Backup**, y este archivo será el que actuara como dato para hacer el backup.

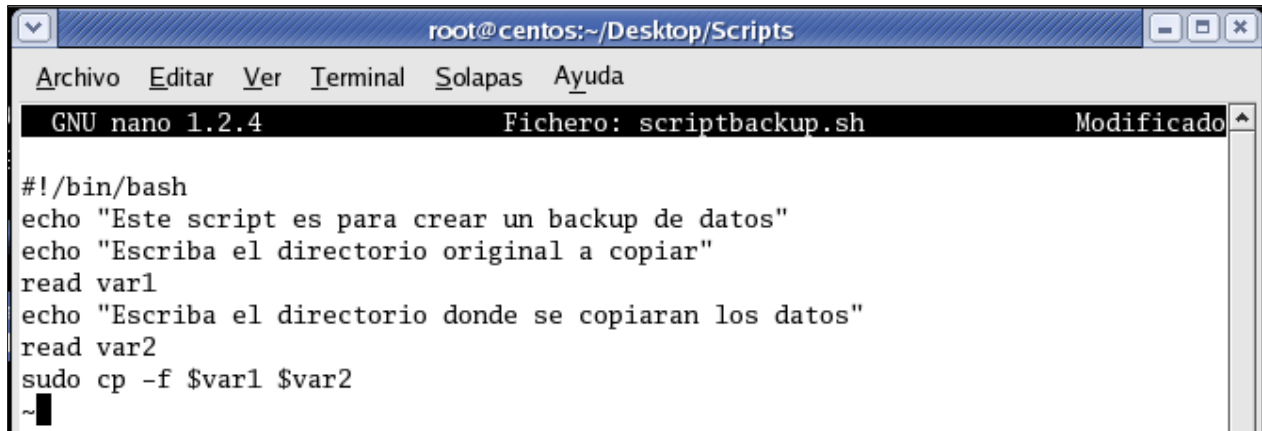


```
root@centos:~/Desktop/Backup
Archivo Editar Ver Terminal Solapas Ayuda
[root@centos Desktop]# cd Backup
[root@centos Backup]# nano Dato
[root@centos Backup]# ls -l
total 8
-rw-r--r-- 1 root root 84 mar  6 07:07 Dato
[root@centos Backup]#
```

Procedemos a la **creación del script**.



```
root@centos:~/Desktop/Scripts
Archivo Editar Ver Terminal Solapas Ayuda
[root@centos Backup]# cd ..
[root@centos Desktop]# cd Scripts/
[root@centos Scripts]#
[root@centos Scripts]# nano scriptbackup.sh
```



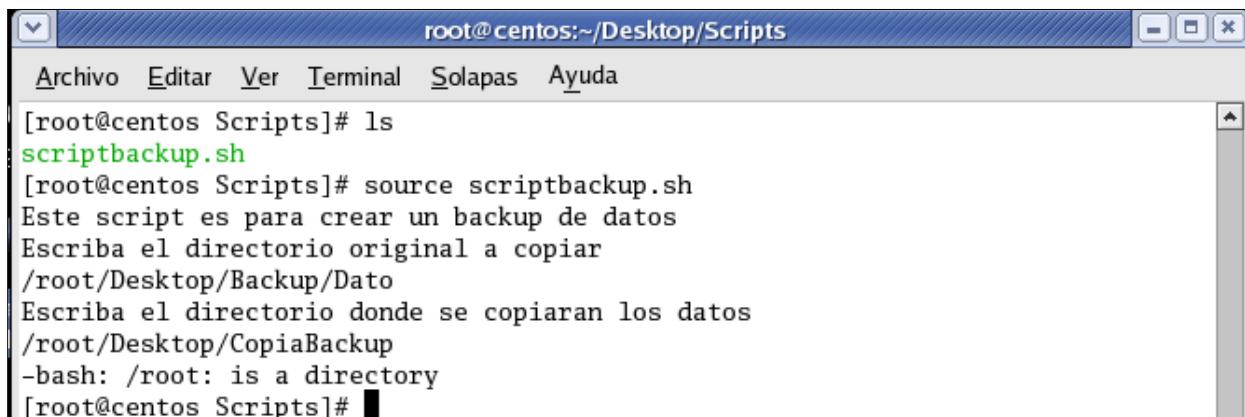
```
root@centos:~/Desktop/Scripts
Archivo Editar Ver Terminal Solapas Ayuda
GNU nano 1.2.4           Fichero: scriptbackup.sh           Modificado
#!/bin/bash
echo "Este script es para crear un backup de datos"
echo "Escriba el directorio original a copiar"
read var1
echo "Escriba el directorio donde se copiaran los datos"
read var2
sudo cp -f $var1 $var2
~
```

El script nos pedirá el directorio donde se encuentran los datos a copiar, y también nos pedirá el directorio destino donde serán copiados.

Los datos que vamos a copiar en esta práctica pertenecen a la siguiente ruta:

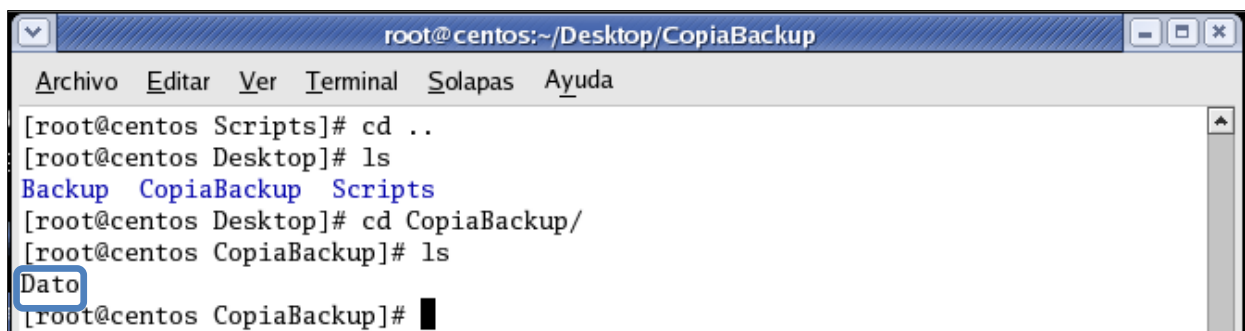
/root/Desktop/Backup/Datos.

Y la ruta donde se copiaran esos datos es: **/root/Desktop/CopiaBackup/.**



```
root@centos:~/Desktop/Scripts
Archivo Editar Ver Terminal Solapas Ayuda
[root@centos Scripts]# ls
scriptbackup.sh
[root@centos Scripts]# source scriptbackup.sh
Este script es para crear un backup de datos
Escriba el directorio original a copiar
/root/Desktop/Backup/Dato
Escriba el directorio donde se copiaran los datos
/root/Desktop/CopiaBackup
-bash: /root: is a directory
[root@centos Scripts]#
```

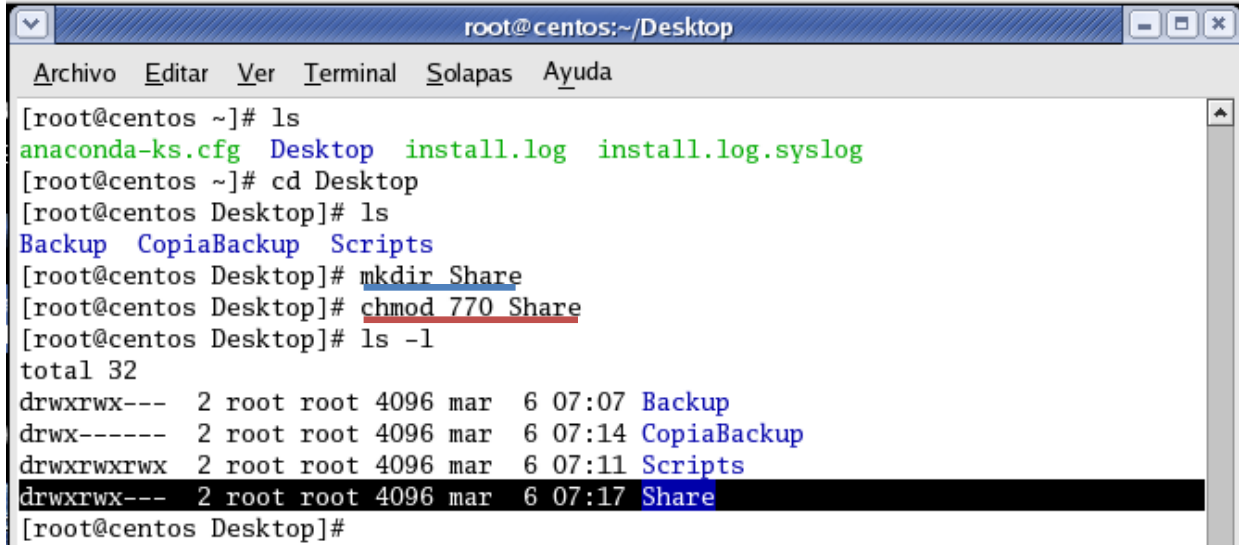
Ahora nos dirigimos al directorio que elegimos para hacer el **backup**, y vemos los datos copiados y guardados en este directorio.



```
root@centos:~/Desktop/CopiaBackup
Archivo Editar Ver Terminal Solapas Ayuda
[root@centos Scripts]# cd ..
[root@centos Desktop]# ls
Backup CopiaBackup Scripts
[root@centos Desktop]# cd CopiaBackup/
[root@centos CopiaBackup]# ls
Dato
[root@centos CopiaBackup]#
```

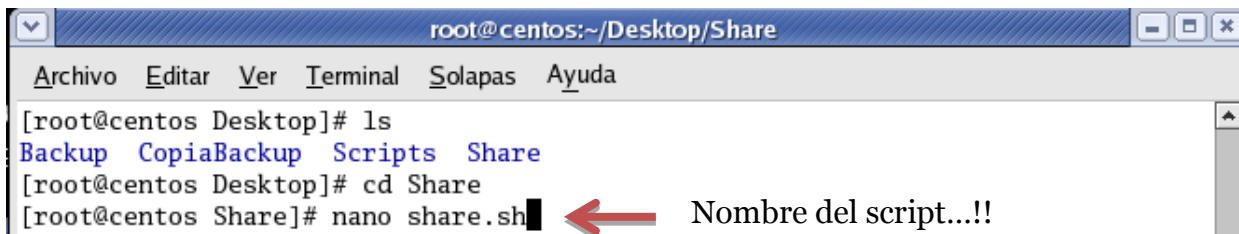

Script para transferir datos entre PCs

Para **hacer un script** que nos permita transferir **un directorio a otra PC** lo primero que tenemos que hacer es crear un directorio que será el compartido entre las dos PCs dándole en primer lugar los permisos necesarios al directorio que se va a transferir.



```
root@centos:~/Desktop
Archivo Editar Ver Terminal Solapas Ayuda
[root@centos ~]# ls
anaconda-ks.cfg Desktop install.log install.log.syslog
[root@centos ~]# cd Desktop
[root@centos Desktop]# ls
Backup CopiaBackup Scripts
[root@centos Desktop]# mkdir Share
[root@centos Desktop]# chmod 770 Share
[root@centos Desktop]# ls -l
total 32
drwxrwx--- 2 root root 4096 mar  6 07:07 Backup
drwx----- 2 root root 4096 mar  6 07:14 CopiaBackup
drwxrwxrwx 2 root root 4096 mar  6 07:11 Scripts
drwxrwx--- 2 root root 4096 mar  6 07:17 Share
[root@centos Desktop]#
```

Luego nos ubicamos en el directorio, **y procedemos a crear el script.**



```
root@centos:~/Desktop/Share
Archivo Editar Ver Terminal Solapas Ayuda
[root@centos Desktop]# ls
Backup CopiaBackup Scripts Share
[root@centos Desktop]# cd Share
[root@centos Share]# nano share.sh ← Nombre del script...!!
```

Utilizaremos los siguientes parámetros al momento de crear el script:

- **SOURCEDIR:** para establecer el origen de nuestro directorio a transferir
- **DESTDIR:** para establecer el destino de nuestro directorio de nuestra otra pc.
- **rsync:** para sincronizar las carpetas.
- **progress:** para enseñar el progreso de la transferencia
- **avze:** opciones del comando rsync.


```
root@centos:~/Desktop/Share
Archivo Editar Ver Terminal Solapas Ayuda
GNU nano 1.2.4 Fichero: share.sh Modificado
#!/bin/bash
echo "Script para transferir datos entre PCs"

read -p "Directorio:" SOURCEDIR
echo
read -p "Directorio:" DESTDIR
echo
echo
echo "Espere a que los datos se transfieran"

sudo rsync --progress -avze ssh $SOURCEDIR $DESTDIR
```

Asignamos **los permisos** necesarios al script.

```
root@centos:~/Desktop/Share
Archivo Editar Ver Terminal Solapas Ayuda
[root@centos Share]# ls
share.sh
[root@centos Share]# chmod 770 share.sh
[root@centos Share]#
```

```
root@centos:~/Desktop/Share
Archivo Editar Ver Terminal Solapas Ayuda
[root@centos Share]# mkdir ShareTest
[root@centos Share]# mkdir Test
[root@centos Share]# ls
share.sh ShareTest Test
[root@centos Share]#
[root@centos Share]# ls -l
total 24
-rwxrwx--- 1 root root 233 mar  6 07:24 share.sh
drwxr-xr-x 2 root root 4096 mar  6 07:26 ShareTest
drwxr-xr-x 2 root root 4096 mar  6 07:26 Test
[root@centos Share]#
```

Vemos los archivos que se van a transferir...

```

Aplicaciones Acciones mar 6 de mar, 07:53
root@centos:~/Desktop/Share
Archivo Editar Ver Terminal Solapas Ayuda

[root@centos Share]# ls
192.168.46.148 share.sh ShareTest Test
[root@centos Share]# source share.sh
Script para transferir datos entre PCs
Directorio:/root/Desktop/Share

Directorio:192.168.46.148:/home/Martinez/Desktop

Espera a que los datos se transfieran
root@192.168.46.148's password:
building file list ...
9 files to consider
Share/
Share/192.168.46.148/
Share/192.168.46.148/Share/
Share/192.168.46.148/Share/ShareTest/
Share/192.168.46.148/Share/Test/
Share/192.168.46.148/Share/share.sh
 235 100% 0.00kB/s 0:00:00 (1, 66.7% of 9)
Share/ShareTest/
Share/Test/
Share/share.sh
 235 100% 229.49kB/s 0:00:00 (2, 100.0% of 9)

sent 600 bytes received 60 bytes 69.47 bytes/sec
total size is 470 speedup is 0.71
[root@centos Share]#

```

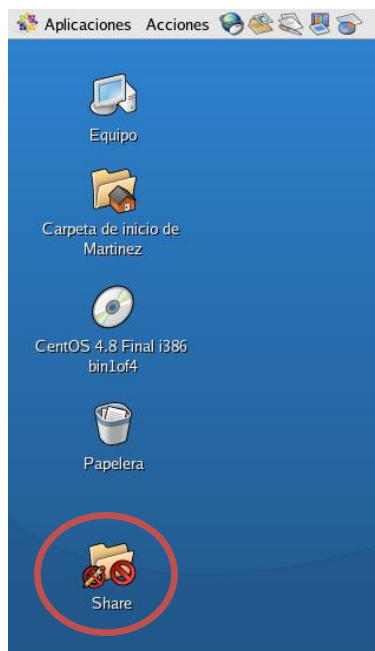
Ejecutamos el script con el comando **source**.

Directorio desde donde vamos a compartir.

Directorio al cual vamos a transferir. Para esto debemos **saber la dirección IP de la maquina donde** se van a transferir los datos. En este caso esa **dirección IP es 192.168.56.104**, y el directorio se escribirá de la siguiente manera: **192.168.56.104:/root/share-doc**.

Password PC destino.

Ahora vamos a la maquina en la cual se copiaron los archivos y podremos ver los datos:



Ahora procederemos a darle permisos la carpeta compartida para así poder explorar los archivos que están dentro de ella.

```

Martinez@localhost:/home/Martinez/Desktop
Archivo Editar Ver Terminal Solapas Ayuda

[root@localhost Desktop]# dir
Share
[root@localhost Desktop]# chmod 777 Share
[root@localhost Desktop]# clear

[root@localhost Desktop]# ls -l
total 8
drwxrwxrwx 5 root root 4096 mar 6 2012 Share
[root@localhost Desktop]# chmod 777 Share -R
[root@localhost Desktop]#

```

