

## CREACION DE SCRIPT

### ¿Qué es el Shell?

El Shell es un intérprete de órdenes, pero el Shell no es solamente eso; los intérpretes de órdenes de Linux son auténticos lenguajes de programación. Como tales, incorporan sentencias de control de flujo, sentencias de asignación, funciones, etc.

Los programas de Shell no necesitan ser compilados como ocurre en otros lenguajes. En este caso, el propio Shell los ejecuta línea a línea. A estos programas se les conoce con el nombre de Shell scripts y son los equivalentes a los archivos por lotes de otros sistemas operativos.

Nota Importante: Para crear un script no necesitamos compilar su código para poder ejecutarlo, con lo que solo necesitamos escribir ASCII en un fichero, y para eso podemos editar cualquiera de los editores, escribimos las órdenes y la guardamos con un nombre en cualquier parte visible y fácil de encontrar en el sistema.

Aplicando esta información a la práctica vamos a realizar varios script, como primer ejemplo realizaremos el script de creación de usuario.

### Script de creación de usuario

Primero con un editor de texto en este caso (**nano**) vamos a editar un script llamado **script\_usuario**

```
root@ubuntuserver:/home/usuario # nano script_usuario_
```



Luego dentro de la edición, colocaremos la línea de comando para agregar usuario **adduser nombre**

```
GNU nano 2.2.2      Archivo: script_usuario      Modificado
adduser guillermo

[ Archivo nuevo ]
^G Ver ayuda  ^O Guardar  ^R Leer Fich ^Y RePág.  ^K Cortar Tex ^C Pos actual
^X Salir      ^J Justificar ^W Buscar   ^U Pág. Sig. ^U PegarTxt  ^T Ortografía
```

Luego de grabar el archivo tendremos lo siguiente, con el comando **ls -l** veremos nuestro script colocado.

```
root@ubuntuserver:/home/userver# ls -l
total 8
drwxr-xr-x 2 root root 4096 2012-02-12 22:14 prueba-nfs
-rw-r--r-- 1 root root  19 2012-03-06 19:51 script_usuario
root@ubuntuserver:/home/userver#
```

Ahora utilizando la opción **sh** o shell y el nombre del script colocado se ejecutara el mismo

```
root@ubuntuserver:/home/userver# ls -l
total 8
drwxr-xr-x 2 root root 4096 2012-02-12 22:14 prueba-nfs
-rw-r--r-- 1 root root  19 2012-03-06 19:51 script_usuario
root@ubuntuserver:/home/richard# sh script_usuario
Añadiendo el usuario 'guillermo' ...
Añadiendo el nuevo grupo 'guillermo' (1002) ...
Añadiendo el nuevo usuario 'guillermo' (1002) con grupo 'guillermo' ...
Creando el directorio personal '/home/guillermo' ...
Copiando los ficheros desde '/etc/skel' ...
Introduzca la nueva contraseña de UNIX:
Vuelva a escribir la nueva contraseña de UNIX:
passwd: password updated successfully
Changing the user information for guillermo
Enter the new value, or press ENTER for the default
    Full Name []: Ricardo guillermo
    Room Number []:
    Work Phone []: 809-446-9527
    Home Phone []:
    Other []:
¿Es correcta la información? [S/n] S
root@ubuntuserver:/home/userver#
```

Para verificar basta con presentar la información del usuario creado.

```
root@ubuntuserver:/home/userver# finger guillermo
Login: guillermo                Name: Ricardo guillermo
Directory: /home/guillermo      Shell: /bin/bash
Office Phone: 809-446-9527
Never logged in.
No mail.
No Plan.
root@ubuntuserver:/home/userver#
```

Es necesario resalta que la ejecución del script debe ser dada por los privilegios correspondientes editados por chmod y chown, en el ejemplo siguiente le damos privilegios al usuarios root y al grupo root para la ejecución, edición y lectura, pero a los demás no le entregamos ningún privilegio

```
root@ubuntuserver:/home/userver# ls -l
total 8
drwxr-xr-x 2 root root 4096 2012-02-12 22:14 prueba-nfs
-rw-r--r-- 1 root root 19 2012-03-06 19:51 script_usuario
root@ubuntuserver:/home/userver# chmod 770 script_usuario
root@ubuntuserver:/home/userver# _
```

## Script de creación de grupo

Ya conociendo la forma de la creación y ejecución de los script podemos realizar el siguiente script con el comando **groupadd script-prueba** editando un archivo al que daremos como nombre Script-prueba.

```
root@ubuntuserver:/home/userver# nano script-prueba
```

Luego insertaremos el comando **groupadd script-prueba** para la creación del grupo.



The screenshot shows the nano 2.2.2 text editor interface. The title bar at the top indicates the file being edited is 'script-prueba'. The main editing area contains the text 'groupadd script-prueba'. At the bottom, a status bar shows '[ 2 líneas leídas ]' and a list of keyboard shortcuts for various editor functions such as Ver ayuda, Guardar, Leer Fich, RePág., Cortar Tex, Pos actual, Salir, Justificar, Buscar, Pág. Sig., PegarTxt, and Ortografía.

Ahora también daremos los privilegios correspondientes solo al usuario richard

```
root@ubuntuuserver:/home/userver# ls -l
total 8
drwxr-xr-x 2 root root 4096 2012-02-12 22:14 prueba-nfs
-rw-r--r-- 1 root root  24 2012-03-07 02:21 script-prueba
root@ubuntuuserver:/home/userver# chown richard script-prueba
root@ubuntuuserver:/home/userver# chmod 700 script-prueba
root@ubuntuuserver:/home/userver#
```

Luego ejecutaremos el script

```
root@ubuntuuserver:/home/userver# sh script-prueba
root@ubuntuuserver:/home/userver# _
```

Y por último verificaremos la creación en el directorio con el comando **nano /etc/group**

```
GNU nano 2.2.2      Archivo: /etc/group
postfix:x:114:
postdrop:x:115:
dovecot:x:116:
haldaemon:x:117:
kvm:x:118:
libvirt:x:119:userver
postgres:x:120:
smbshare:x:121:userver
sane:x:122:
winbindd_priv:x:123:
landscape:x:124:
tomcat6:x:125:
richard:x:1000:
admin:x:126:userver
permisos:x:1001:
guillermo:x:1002:
script-prueba:x:1003:

^G Ver ayuda ^O Guardar ^R Leer Fich ^V RePág. ^K Cortar Tex ^C Pos actual
^X Salir ^J Justificar ^W Buscar ^U Pág. Sig. ^U PegarTxt ^T Ortografía
```

Ya aquí tenemos nuestro grupo creado como se editó en el script.

En esta ocasión vamos a crear un script para realizar backup de datos, primero crearemos la carpetas, tanto la de backup como a la cual se le realizara **mkdir /home/userver/backup** y **/home/userver/prueba**

```
root@ubuntuserver:/home/userver# mkdir /home/userver/backup
root@ubuntuserver:/home/userver#
```

Ya aquí podemos ver los directorios creados he incluso el directorio **prueba-nfs** anteriormente utilizado en prácticas anteriores y que utilizaremos mas adelante.

```
root@ubuntuserver:~# ls -l /home/userver
total 12
drwxr-xr-x 2 root root 4096 2012-03-07 02:50 backup
drwxr-xr-x 2 root root 4096 2012-03-07 02:55 prueba
drwxr-xr-x 2 root root 4096 2012-02-12 22:14 prueba-nfs
root@ubuntuserver:~# ls -l /home/userver/prueba
total 4
-rw-r--r-- 1 root root 27 2012-03-07 02:55 prueba-script
root@ubuntuserver:~# _
```



Luego de creados nuestros directorios vamos a utilizar el editor **nano** para crear un elemento de prueba de backup

```
root@ubuntuserver:~# nano /home/userver/prueba/prueba-script_
```

```
GNU nano 2.2.2 Archivo: /home/userver/prueba/prueba-script
prueba de script de backup

[ 1 línea leída ]
^G Ver ayuda ^O Guardar ^R Leer Fich ^Y RePág. ^K Cortar Tex ^C Pos actual
^X Salir ^J Justificar ^W Buscar ^U Pág. Sig. ^U PegarTxt ^T Ortografía
```

Luego de guardado, vamos a crear nuestro script y a dar los privilegios para edición, ejecución y lectura al grupo root y al usuario root.

Para ello con el comando **nano script-backup**, editaremos el siguiente script **ln /home/richard/prueba/prueba-script /home/userver/backup**, donde **ln** es para crear un enlace entre ficheros, en este caso vinculamos **/home/userver/prueba/prueba-script** (archivo origen de los datos) y **/home/userver/backup** (archivo destino de las copias).

```
GNU nano 2.2.2      Archivo: script-backup

ln /home/userver/prueba/prueba-script /home/userver/backup

[ 2 líneas leídas ]
^G Ver ayuda ^O Guardar ^R Leer Fich ^Y RePág. ^K Cortar Tex ^C Pos actual
^X Salir ^J Justificar ^W Buscar ^U Pág. Sig. ^U PegarTxt ^T Ortografía
```

Luego de creado nuestro script entregamos los privilegios con el comando **chmod 770 script-backup**, donde solo hemos dado los privilegios de edición, ejecución y lectura al usuario y grupo root

```
[ 1 línea escrita ]

root@ubuntuserver:~# ls -l
total 4
-rw-r--r-- 1 root root 45 2012-03-07 03:11 script-backup
root@ubuntuserver:~# chmod 770 script-backup
```

Luego de esto solo basta la ejecución de nuestro script utilizando shell con el comando **sh script-backup** y se realizara la sentencia colocada, que en este caso es el backup al archivo correspondiente.

```
root@ubuntuserver:~# ls -l
total 4
-rwxrwx--- 1 root root 60 2012-03-07 04:05 script-backup
root@ubuntuserver:~# sh script-backup
root@ubuntuserver:~# ls -l /home/userver/backup
total 4
-rw-r--r-- 2 root root 27 2012-03-07 02:55 prueba-script
root@ubuntuserver:~#
```

## Creación de script para servicio nfs

Vamos a proceder a recordar el estado actual de nuestro servicio nfs, primero confirmando la carpeta a compartir con el comando `ls -l /home/userver/prueba-nfs`

```
root@ubuntuuserver:~# ls -l /home/userver /
total 12
drwxr-xr-x 2 root root 4096 2012-03-07 04:05 backup
drwxr-xr-x 2 root root 4096 2012-03-07 02:55 prueba
drwxr-xr-x 2 root root 4096 2012-02-12 22:14 prueba-nfs
root@ubuntuuserver:~# ls -l /home/userver/prueba-nfs
total 0
root@ubuntuuserver:~#
```

Luego de ello vamos a verificar nuestro directorio de con el comando `nano /etc/exports`, donde podemos notar la carpeta a compartir, la red permitida y sus privilegios.

```
GNU nano 2.2.2      Archivo: /etc/exports
# /etc/exports: the access control list for filesystems which may be exported
# to NFS clients.  See exports(5).
#
# Example for NFSv2 and NFSv3:
# /srv/homes          hostname1(rw,sync,no_subtree_check) hostname2(ro,sync,no_sub$
#
# Example for NFSv4:
# /srv/nfs4           gss/krb5i(rw,sync,fsid=0,crossmnt,no_subtree_check)
# /srv/nfs4/homes     gss/krb5i(rw,sync,no_subtree_check)
#
/home/richard/prueba-nfs 192.168.1.0/24 (rw,sync,no_root_squash,no_subtre$

^G Ver ayuda  ^O Guardar  ^R Leer Fich ^Y RePág.   ^K Cortar Tex ^C Pos actual
^X Salir      ^J Justificar ^W Buscar   ^U Pág. Sig. ^U PegarTxt  ^T Ortografía
```

Vamos a realizar un cambio de permisos a la carpeta con el comando **chmod 777 /home/userver/prueba-nfs**, para así entregarle todos los privilegios a cualquier usuario.

```
root@ubuntuuserver:~# chmod 777 /home/userver/prueba-nfs
root@ubuntuuserver:~#
```

Para la aplicación de todos los cambio como es recomendable siempre, realizaremos un reinicio a kernel nfs, con el comando **/etc/init.d/nfs-kernel-server restart**

```
root@ubuntuuserver:~# chmod 777 /home/userver/prueba-nfs
root@ubuntuuserver:~# /etc/init.d/nfs-kernel-server restart
* Stopping NFS kernel daemon [ OK ]
* Unexporting directories for NFS kernel daemon... [ OK ]
* Exporting directories for NFS kernel daemon...
exportfs: No options for /home/userver/prueba-nfs 192.168.1.0/24: suggest 192.16
8.1.0/24(sync) to avoid warning
exportfs: /etc/exports [1]: Neither 'subtree_check' or 'no_subtree_check' specif
ied for export "192.168.1.0/24:/home/userver/prueba-nfs".
    Assuming default behaviour ('no_subtree_check').
    NOTE: this default has changed since nfs-utils version 1.0.x

exportfs: No host name given with /home/userver/prueba-nfs (rw,sync,no_root_squa
sh,no_subtree_check), suggest *(rw,sync,no_root_squash,no_subtree_check) to avoi
d warning

* Starting NFS kernel daemon [ OK ]
root@ubuntuuserver:~# _
```

Ahora bien en nuestro cliente vamos a crear el script para la el auto montaje con el servidor nfs, esto lo realizaremos de la siguiente manera.

- 1- En nuestro cliente verifiquemos la conexión con el servido. Comando **ping 192.168.1.3**

```
* Starting NFS kernel daemon [ OK ]
root@ubuntuserver:~# ifconfig
eth0      Link encap:Ethernet  direcciónHW 00:0c:29:3c:2a:a1
          Direc. inet:192.168.1.3  Difus.:192.168.1.255  Másc:255.255.255.0
          Dirección inet6: fe80::20c:29ff:fe3c:2aa1/64 Alcance:Enlace
          ACTIVO DIFUSIÓN FUNCIONANDO MULTICAST MTU:1500 Métrica:1
          Paquetes RX:13774 errores:0 perdidos:0 overruns:0 frame:0
          Paquetes TX:220 errores:0 perdidos:0 overruns:0 carrier:0
          colisiones:0 long.colatX:1000
          Bytes RX:874182 (874.1 KB)  TX bytes:38673 (38.6 KB)

lo        Link encap:Bucle local
          Direc. inet:127.0.0.1  Másc:255.0.0.0
          Dirección inet6: ::1/128 Alcance:Anfitrión
          ACTIVO BUCLE FUNCIONANDO MTU:16436 Métrica:1
          Paquetes RX:1116 errores:0 perdidos:0 overruns:0 frame:0
          Paquetes TX:1116 errores:0 perdidos:0 overruns:0 carrier:0
          colisiones:0 long.colatX:0
          Bytes RX:166929 (166.9 KB)  TX bytes:166929 (166.9 KB)

virbr0    Link encap:Ethernet  direcciónHW ba:15:c2:91:3c:ef
          Direc. inet:192.168.122.1  Difus.:192.168.122.255  Másc:255.255.255.0
          Dirección inet6: fe80::b815:c2ff:fe91:3cef/64 Alcance:Enlace
          ACTIVO DIFUSIÓN FUNCIONANDO MULTICAST MTU:1500 Métrica:1
          Paquetes RX:0 errores:0 perdidos:0 overruns:0 frame:0
          Paquetes TX:102 errores:0 perdidos:0 overruns:0 carrier:0
          colisiones:0 long.colatX:0
          Bytes RX:0 (0.0 B)  TX bytes:19649 (19.6 KB)

root@ubuntuserver:~#
```

```
ggomez@ggomez-virtual-machine:~$ ping 192.168.1.3
PING 192.168.1.3 (192.168.1.3) 56(84) bytes of data.
64 bytes from 192.168.1.3: icmp_req=1 ttl=64 time=8.63 ms
64 bytes from 192.168.1.3: icmp_req=2 ttl=64 time=0.491 ms
64 bytes from 192.168.1.3: icmp_req=3 ttl=64 time=0.406 ms
64 bytes from 192.168.1.3: icmp_req=4 ttl=64 time=0.397 ms
^C
--- 192.168.1.3 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3000ms
rtt min/avg/max/mdev = 0.397/2.483/8.638/3.553 ms
ggomez@ggomez-virtual-machine:~$
```

2- Luego confirmamos la conexión con el directorio a compartir

```
gjjomez@gjjomez-virtual-machine:~$ showmount -e 192.168.1.3
Export list for 192.168.1.3:
/home/gjjomez/prueba-nfs (everyone)
gjjomez@gjjomez-virtual-machine:~$
```

3- Ahora crearemos el script para montar la conexión con el servidor, esto lo realizaremos con el comando **nano script-nfs**

```
GNU nano 2.2.4      Archivo: script-nfs      Modificado
mount 192.168.1.3:/home/gjjomez/prueba-nfs /home/gjjomez/Escritorio/nfs

^G Ver ayuda  ^O Guardar  ^R Leer Fich ^Y RePág.    ^K Cortar Tex ^C Pos actual
^X Salir      ^J Justificar ^W Buscar    ^V Pág. Sig.  ^U PegarTxt   ^T Ortografia
```

Luego modificamos quienes pueden realizar el script con el comando **chmod 770 script-nfs**

```
gjjgomez@gjjgomez-virtual-machine:~$ showmount -e 192.168.1.3
Export list for 192.168.1.3:
/home/gjjgomez/prueba-nfs (everyone)
gjjgomez@gjjgomez-virtual-machine:~$ nano script-nfs
gjjgomez@gjjgomez-virtual-machine:~$ sudo su
[sudo] password for richard:
root@gjjgomez-virtual-machine:/home/gjjgomez# chmod 770 script-nfs
root@gjjgomez-virtual-machine:/home/gjjgomez# █
```

Luego ponemos a correr nuestro script con el comando **sh script-nfs** y ya con esto tendremos nuestro script para montar la conexión con el nfs.

```
root@gjjgomez-virtual-machine:/home/gjjgomez# sh script-nfs
root@gjjgomez-virtual-machine:/home/gjjgomez#
```