

Universidad Tecnológica de Panamá
Facultad de Ingeniería en Sistemas Computacionales
Departamento de Arquitectura y Redes de Comunicación

Laboratorio SHELL SCRIPTS

Prof. Aris Castillo de Valencia

OBJETIVO:

- El objetivo de este documento es practicar en la creación de shell scripts bash.

MATERIALES

- Computadora, con maquina virtual y sistema operativo Linux

INFORMACION

Un **script** es un **archivo de ordenes** o pequeño **programa** con el que poder automatizar ciertas tareas o acciones en un ordenador. Intentaré explicarlo de forma más sencilla.

Un **script** suele ser un archivo de texto plano, en el que se insertan las órdenes que se quieren realizar. En Ubuntu (o **cualquier distribución Linux**) podríamos hacer un script con comandos de consola para poder, por ejemplo, actualizar el sistema, instalar alguna aplicación, hacer un **backup de archivos**, o todo a la vez. El contenido de un **script** que nos hiciese un **backup** de nuestras carpetas importantes y lo guardase en un **pendrive** podría ser del siguiente modo:

Antes de empezar voy a crear un usuario y a darle permisos root

```
[root@localhost-live liveuser]# sudo useradd jorge
```

```
[root@localhost-live liveuser]# sudo passwd jorge
```

Pedirá una contraseña y se la colocamos.

Luego aplicamos los permisos de la siguiente manera.

```
[root@localhost-live liveuser]# sudo visudo
```

jorge ALL=(ALL) ALL

```
GNU nano 6.4 /etc/sudoers.tmp
##      user      MACHINE=COMMANDS
##
## The COMMANDS section may have other options added to it.
##
## Allow root to run any commands anywhere
root    ALL=(ALL)      ALL
jorge ALL=(ALL) ALL
## Allows members of the 'sys' group to run networking, software,
## service management apps and more.
# %sys ALL = NETWORKING, SOFTWARE, SERVICES, STORAGE, DELEGATING, PROCESSES, LO>

## Allows people in group wheel to run all commands
%wheel  ALL=(ALL)      ALL

## Same thing without a password
# %wheel      ALL=(ALL)      NOPASSWD: ALL

## Allows members of the users group to mount and unmount the
## cdrom as root
# %users  ALL=/sbin/mount /mnt/cdrom, /sbin/umount /mnt/cdrom

^G Help      ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location
^X Exit      ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^_ Go To Line
```

Ctrl+O para guardar y ctrl+X para salir.

```
#!/bin/bash
cd /media/pendrive
tar -cvzf documentos.tar.gz /home/usuario/Documentos/*
tar -cvzf videos.tar.gz /home/usuario/Videos/*
tar -cvzf imagenes.tar.gz /home/usuario/Imagenes/*
tar -cvzf descargas.tar.gz /home/usuario/Descargas/*
```

```
[jorge@localhost-live liveuser]$ sudo nano script.sh
```

```
jorge@localhost-live:/home/liveuser — sudo nano script.sh
GNU nano 6.4 script.sh Modified
#!/bin/bash
cd /home/Desktop/copia
tar -cvzf documentos.tar.gz /home/jorge/Documents/*
tar -cvzf videos.tar.gz /home/jorge/Videos/*
tar -cvzf imagenes.tar.gz /home/jorge/Pictures/*
tar -cvzf descargas.tar.gz /home/jorge/Downloads/*
```

Ctrl+O guardar y ctrl+X salir

```
[jorge@localhost-live liveuser]$ sudo cat script.sh
#!/bin/bash
cd/home/Desktop/copia
tar -cvzf documentos.tar.gz /home/jorge/Documents/*
tar -cvzf videos.tar.gz /home/jorge/Videos/*
tar -cvzf imagenes.tar.gz /home/jorge/Pictures/*
tar -cvzf descargas.tar.gz /home/jorge/Downloads/*
```

De este modo, podríamos guardar y comprimir todo el contenido de nuestras carpetas importantes con un solo click o ejecutandolo en consola así: **./script.sh**

Hay que aplicarle permisos de ejecución de la siguiente manera.

```
[jorge@localhost-live liveuser]$ sudo ./script.sh
sudo: ./script.sh: command not found
[jorge@localhost-live liveuser]$ sudo chmod +x script.sh
```

Ejecutamos ./script.sh

```
[jorge@localhost-live liveuser]$ sudo ./script.sh
./script.sh: line 2: cd/home/Desktop/copia: No such file or directory
tar: Removing leading `/' from member names
tar: /home/jorge/Documents/*: Cannot stat: No such file or directory
tar: Exiting with failure status due to previous errors
tar: Removing leading `/' from member names
tar: /home/jorge/Videos/*: Cannot stat: No such file or directory
tar: Exiting with failure status due to previous errors
tar: Removing leading `/' from member names
tar: /home/jorge/Pictures/*: Cannot stat: No such file or directory
tar: Exiting with failure status due to previous errors
tar: Removing leading `/' from member names
tar: /home/jorge/Downloads/*: Cannot stat: No such file or directory
tar: Exiting with failure status due to previous errors
```

Verificamos que se hayan creado las copias

```
[jorge@localhost-live liveuser]$ sudo cd Desktop
[sudo] password for jorge:
[jorge@localhost-live liveuser]$ ls
ls: cannot open directory '.': Permission denied
[jorge@localhost-live liveuser]$ sudo ls
descargas.tar.gz  Documents      Music          Scripts        Videos
Desktop           Downloads      Pictures       script.sh      videos.tar.gz
documentos.tar.gz imagenes.tar.gz Public         Templates
```

Los scripts no sólo usan **comandos Linux**, hay de muchos tipos y para todos los sistemas operativos. En Windows, por ejemplo, el tipo de scripts más común son los archivos ***.bat**.

En Linux lo más común es el **Shell Script**, scripts con comandos de **consola Linux**. Pero los scripts también pueden ser **scripts PHP**, python, java, etc.

```
cd $HOME && touch script.sh && chmod +x script.sh
```

```
cd $HOME && echo '#!/bin/bash' > script.sh && echo '# -*- ENCODING: UTF-8 -*-' >> script.sh
```

```
#!/bin/bash
# -*- ENCODING: UTF-8 -*-
echo "<° Linux es lo mejor"
exit
```

```
[jorge@localhost-live liveuser]$ $HOME
bash: /home/jorge: Is a directory
[jorge@localhost-live liveuser]$ sudo nano script2.sh
```

```
GNU nano 6.4 script2.sh
#!/bin/bash
#-*-ENCODING:UTF-8-*-
echo":) Linux es lo mejor"
exit
```

Para ejecutar el script:

```
./script.sh
```

Damos permisos de ejecución

```
[jorge@localhost-live liveuser]$ sudo chmod +x script2.sh
```

```
[jorge@localhost-live liveuser]$ sudo ./script2.sh
./script2.sh: line 3: echo:) Linux es lo mejor: command not found
```

PROCEDIMIENTO

1. Verificar conectividad con un sitio web:

Primero debe entrar a un editor, escribir los comandos y guardar el archivo con extensión sh

```
#!/bin/bash
ping -c 3 www.google.com > /dev/null
if [ $? != 0 ]
then
echo "Google is down"
fi
```

- Después en la línea de comandos debe cambiar el script a modo ejecución:
chmod +x nombredelscript
- Después puede ejecutar el script colocando:

./nombredelscript.sh

```
[root@localhost-live liveuser]# nano conectividad.sh
```

```
GNU nano 6.4 conectividad.sh
#!/bin/bash
ping -c 3 www.google.com > dev/null
if [ $? -ne 0 ]; then
echo "google is down"
else
echo "google is working correctly"
fi
```

```
[root@localhost-live liveuser]# chmod +x conectividad.sh
```

```
./conectividad.sh: line 6: echogoogle is working correctly:
```

2. Verificar que un servicio/proceso se está ejecutando

Si queremos saber si existen procesos de algún servicio o programa ejecutándose podríamos usar este script (*en este caso verifica que el demonio de **Apache** esté corriendo*):

```
#!/bin/sh
SERVICE='httpd'
if ps ax | grep -v grep | grep $SERVICE > /dev/null
then
echo "El servicio $SERVICE esta ejecutandose"
else
echo "El servicio $SERVICE esta detenido"
fi
```

```
[root@localhost-live liveuser]# nano servicios.sh
```

```
jorge@localhost-live:/home/liveuser — nano servicios.sh
GNU nano 6.4 servicios.sh
#!/bin/bash
SERVICE='httpd'
if ps ax | grep -v grep | $SERVICE > /dev/null
then
echo "El servicio $SERVICE esta ejecutandose"
else
echo "El servicio $SERVICE esta detenido"
fi
```

```
[root@localhost-live liveuser]# chmod +x servicios.sh
[root@localhost-live liveuser]# ./servicios.sh
AH00558: httpd: Could not reliably determine the server's fully qualified domain
name, using fe80::acd9:f9ab:c27f:3bbe%enp0s3. Set the 'ServerName' directive gl
obally to suppress this message
./servicios.sh: line 5: echoEl servicio httpd esta ejecutandose: command not fou
nd
[root@localhost-live liveuser]#
```

3. Pedir confirmación antes de ejecutar un script

A veces es útil hacer que el usuario confirme la ejecución de un lote de sentencias, es decir, el típico mensaje que pide al usuario escribir yes o no. Esto lo podemos hacer así:

```
#!/bin/bash
while true; do
echo
read -p "esta seguro de hacer lo que sea que vaya a hacer " yn
case $yn in
yes ) break;;
no ) exit;;
* ) echo "por favor responda yes o no";;
esac
done
echo "si se ejecuta esto es que aceptaste"
```

```
[root@localhost-live liveuser]# nano confirmacion.sh
```

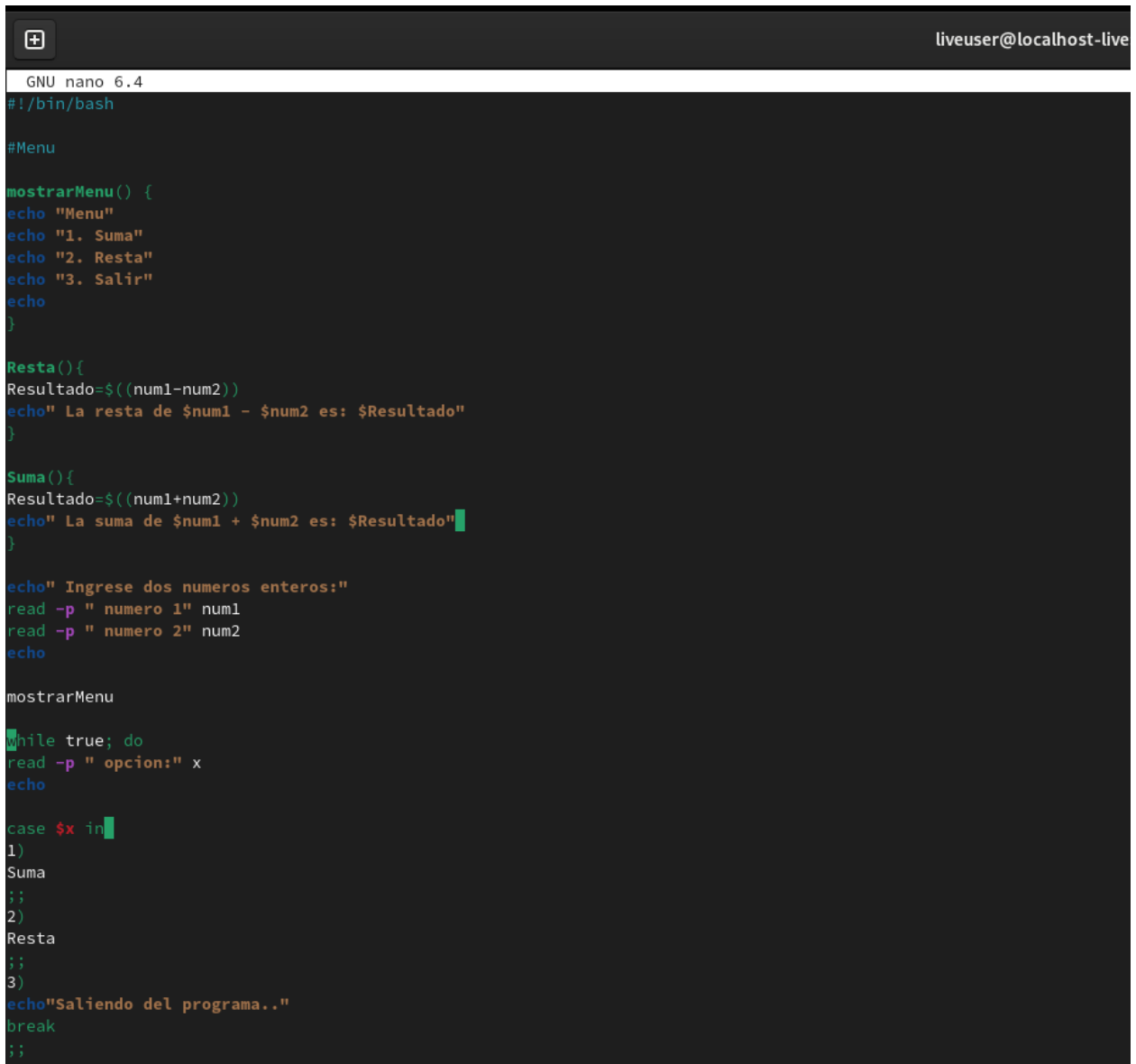
```
GNU nano 6.4                                confirmacion.sh                                Modified
#!/bin/bash
while true; do
echo
read -p "estas seguro de hacer lo que vayas a hacer" yn
case $yn in
yes) break;;
no) exit;;
* ) echo"Responda yes o no";;
esac
done
echo"elegiste que si"
```

```
[root@localhost-live liveuser]# chmod +x confirmacion.sh
```

```
[root@localhost-live liveuser]# ./confirmacion.sh
estas seguro de hacer lo que vayas a haceryes
./confirmacion.sh: line 11: echoelegiste que si: c
```

4. Realice un Shell Script sobre operaciones matemáticas, en donde el usuario ingrese dos números enteros y mediante un menú escoja que operación desea realizar.

```
[root@localhost-live liveuser]# nano Operacion
```



The screenshot shows a terminal window with the title bar 'liveuser@localhost-live'. The terminal content shows the nano 6.4 editor editing a file. The script defines functions for displaying a menu, performing subtraction, and performing addition. It prompts the user to enter two integers and then enters a loop where it displays the menu and processes the user's choice. The script is as follows:

```
GNU nano 6.4
#!/bin/bash

#Menu

mostrarMenu() {
echo "Menu"
echo "1. Suma"
echo "2. Resta"
echo "3. Salir"
echo
}

Resta(){
Resultado=$((num1-num2))
echo" La resta de $num1 - $num2 es: $Resultado"
}

Suma(){
Resultado=$((num1+num2))
echo" La suma de $num1 + $num2 es: $Resultado"
}

echo" Ingrese dos numeros enteros:"
read -p " numero 1" num1
read -p " numero 2" num2
echo

mostrarMenu

while true; do
read -p " opcion:" x
echo

case $x in
1)
Suma
;;
2)
Resta
;;
3)
echo"Saliendo del programa.."
break
;;

```

```
[root@localhost-live liveuser]# chmod +x Operacion
```

```
[root@localhost-live liveuser]# ./Operacion
./Operacion: line 23: echo Ingrese dos numeros enteros:: command not found
numero 13
numero 25

Menu
1. Suma
2. Resta
3. Salir

opcion:1

./Operacion: line 20: echo La suma de 3 + 5 es: 8: command not found
Menu
1. Suma
2. Resta
3. Salir

opcion:3

./Operacion: line 42: echoSaliendo del programa...: command not found
```

INFORME

1. Enliste comandos de red de Linux con sus funciones.
2. Muestre capturas que comprueben la realización de los scripts
3. Conclusión y recomendaciones

REFERENCIA

<https://www.javatpoint.com/linux-networking-commands>

<https://blog.desdelinux.net/scripts-shell-utiles-en-cualquier-programa/>

<https://iterm2.com/documentation-scripting-fundamentals.html>