



Asignatura:

Sistemas Operativos III

Tema:

HOWTO Y VIDEO (CREACION DE SCRIPT)

Participante:

Santiago M. Duval Contreras

Matricula:

2015-3246

Facilitador:

José Doñe

Fecha:

4/5/2020

HOW-TO? | Creando Scripts y Automatizando con Crontabs.

HOW-TO? | Creando Scripts y Automatizando con Crontabs.

En este documento veremos los pasos requerido para crear scripts y automatizarlos con la herramienta crontabs en ClearOS en Oracle VirtualBox.

Link a demostración audiovisual: <https://youtu.be/9pnHsz1mEDg>

Requerimientos:

Tener instalado el programa VirtualBox y tener el OS de ClearOS instalado con los requisitos de hardware virtual que sean requeridos.

¿Qué es un Shell Script?

Un núcleo está en el núcleo de un ordenador. Hace posible la comunicación entre el hardware y el software. Mientras que el núcleo es la parte más interna de un sistema operativo, un shell es el más externo. Un shell en un sistema operativo Linux toma la entrada de usted en forma de comandos, lo procesa y, a continuación, da una salida. Es la interfaz a través de la cual un usuario trabaja en los programas, comandos y scripts. Un terminal que lo ejecuta accede a un shell.

Al ejecutar el terminal, el Shell emite un símbolo del sistema (normalmente \$), donde puede escribir la entrada, que luego se ejecuta cuando pulsa la tecla Intro. La salida o el resultado se visualizan posteriormente en el terminal. El Shell envuelve el delicado interior de un sistema operativo que lo protege de daños accidentales. De ahí el nombre Shell.

Crear un script para hacer una Calculadora.

El script para crear una calculadora en ClearOS usando la cli es el siguiente.

```
#!/bin/bash
clear
sum=0
i="y"

echo "Introduce la primera cifra"
read n1
echo "Ahora la segunda cifra"
read n2
while [ $i = "y" ]
do
echo "1.Sumar"
echo "2.Restar"
echo "3.Multiplicar"
echo "4.Dividir"
echo "Introduce el numero de operación seleccionada (1 al 4)"
read ch
case $ch in
1)sum=`expr $n1 + $n2`
echo "Resultado de la suma ="$sum;;
2)sum=`expr $n1 - $n2`
echo "Resultado de la resta = "$sum;;
3)sum=`expr $n1 \* $n2`
echo "Resultado de la multiplicación = "$sum;;
4)sum=`expr $n1 / $n2`
echo "Resultado de la división = "$sum;;
*)echo "Tu selección es incorrecta";;
esac
echo "Quieres continuar (y/n)) ?"
read i
if [ $i != "y" ]
then
exit
fi
done
```

Crear un script para crear Usuarios.

```
#!/bin/bash
echo "Introduzca el nombre del usuario x que desee crear"
read user
adduser $user && echo "#####edite el password#####" && passwd $user
```

Crear un script para crear Grupos.

```
#!/bin/bash
echo "Introduzca el nombre del grupo x que desee crear"
read grupo
if groupadd $grupo; then
    echo "Completado"
else
    echo "error"
fi
```

Crear un script para hacer un Backup.

```
#!/bin/bash
#
srcdir="/home/S20153246/Desktop"
destdir="/home/S20153246/Desktop/ladoA/"
dt=$(date +%Y-%m-%d_%H-%M-%S)
filename="backup-$dt.zip"
#
zip -r $destdir$filename $srcdir
```

Crear un script para transferir Data.

```
#!/bin/bash
echo "Introduzca el nombre del archivo"
read archivo
scp -r /home/S20153246/Desktop/ladoA/$archivo S20153246@192.168.1.39:/home/S20153246/Desktop/ladoB/
```

Crear un script para administrar servicios.

```
#!/bin/bash
clear
sum=0
i="y"

echo "Introduzca el servicio a Administrar"
read service
while [ $i = "y" ]
do
echo "Que quieres hacer?"
echo "1. Start Servicio"
echo "2. Stop Servicio"
echo "3. Status Servicio"
echo "4. Restart Servicio"
read ser
case $ser in
    1)systemctl start $service.service
        echo "El servicio ha sido iniciado correctamente";;
    2)systemctl stop $service.service
        echo "El servicio $service ha sido detenido";;
    3)systemctl status $service.service
        echo "Aqui el status del servicio $service";;
    4)systemctl restart $service.service
        echo "El servicio $service ha sido reiniciado";;
esac
echo "Quieres continuar (y/n)) ?"
read i
if [ $i != "y" ]
then
    exit
fi
done
```

HOW-TO? | Creando Scripts y Automatizando con Crontabs.

Automatizar un Script con crontabs.

La utilidad de software cron es un programador de trabajos basado en el tiempo; En sistemas operativos de computadoras similares a Unix. Los usuarios que configuran y mantienen entornos de software utilizan cron para programar trabajos (comandos o scripts de shell) para ejecutarse periódicamente a horas, fechas o intervalos fijos. Por lo general, automatiza el mantenimiento o la administración del sistema, aunque su naturaleza de uso general lo hace útil para cosas como descargar archivos de Internet y descargar correo electrónico a intervalos regulares.

Si queremos utilizar crontabs primero debemos de tener el programa en ejecución utilizando el comando **systemctl start crontab**, una vez iniciado podemos utilizar el comando nano para editar el siguiente archivo: **nano /etc/crontab**:

```
SHELL=/bin/bash
PATH=/sbin:/bin:/usr/sbin:/usr/bin
MAILTO=root

# For details see man 4 crontabs

# Example of job definition:
# .----- minute (0 - 59)
# | .----- hour (0 - 23)
# | | .----- day of month (1 - 31)
# | | | .----- month (1 - 12) OR jan,feb,mar,apr ...
# | | | | .----- day of week (0 - 6) (Sunday=0 or 7) OR sun,mon,tue,wed,thu,fri,sat
# | | | | |
# * * * * * user-name  command to be executed
```

Podemos agregar scripts si utilizamos el ejemplo de definición del archivo crontab, por ejemplo: *** * * * * root /home/backup.sh**, una vez añadida esta línea al archivo de crontab, guardamos y reiniciamos el servicio de crontab con el comando **systemctl restart crontab**.