

Sistemas Operativos

Informe de Primer Parcial.

Joan Alexis Córdoba Narváez – A00232548.

RETOS CMDCHALLENGE.COM.

Cuando ingresé a la página, el sitio se encargó de darme automáticamente los retos.

Fueron los siguientes:

```
hello_world/  
current_working_directory/  
list_files/  
list_files_adv/  
nested_dirs/
```

You are off to a great start, keep on going!

```
# Print "hello world".  
# Hint: There are many ways to print text on  
# the command line, one way is with the 'echo'  
# command.  
#  
# Try it below and good luck!  
#  
bash(0)> echo "hello world"  
hello world  
# [ ] [ ] [ ] Correct!
```

```
[root@localhost ~]# echo "hello world"  
hello world  
[root@localhost ~]#
```

```
# You have a new challenge!  
# Print the current working directory.  
#  
bash(0)> pwd  
/var/challenges/current_working_directory  
# [ ] [ ] [ ] Correct!
```

```
[root@localhost ~]# pwd  
/root  
[root@localhost ~]# _
```

```
# You have a new challenge!
# List names of all the files in the current
# directory, one file per line.
#
bash(0)> ls
README
# [ ] [ ] [ ] Correct!
```

```
[root@localhost ~]# ls
anaconda-ks.cfg
[root@localhost ~]# _
```

```
# each pathname that is a directory, an asterisk ( '*') after each that is executable,
# an at sign ('@') after each symbolic link. Output one file per line.
# Hint: directories are suffixed with '/', executables with '*'.
# You will need to restrict the number of columns too.
#
bash(0)> ls -af
./
../
.hidden-dir/
.hidden-file
README
normal-file.dat
runthis.sh*
symlink-file.dat@
# [ ] [ ] [ ] Correct!
```

```
[root@localhost ~]# ls -af
./ ../ anaconda-ks.cfg .bash_logout .bash_profile .bashrc .cshrc .tcshrc
[root@localhost ~]# _
```

```
# There is a file:
# ./.../ /. .the flag.txt
# Show its contents on the screen.
#
bash(0)> cat ./**/*.txt
you got it!
# [ ] [ ] [ ] Correct!
```

```
[root@localhost ~]# cat **/*.txt
# ~/.bash_logout

[root@localhost ~]# _
```

Estas fueron las imágenes de los retos solucionados tanto en la web como en CentOS.

SCRIPT REMOTO.

Se muestra a continuación, la ejecución en consola del Script requerido para hacer una descarga remota a dos estaciones. Dentro del Script se ha incluido la contraseña para hacer que este sea automático. Además se ha hecho uso de SPAWN, EXPECT y SEND que se encargan de la interacción con la entrada y salida de datos facilitando la realización de un Script automático.

```
192.168.85.129 192.168.85.131 192.168.85.129
[root@localhost tmp]# ./slop.sh
spawn ssh operativos@192.168.85.131
operativos@192.168.85.131's password:
Last login: Mon Mar  6 03:00:20 2017 from 192.168.85.129
[operativos@localhost ~]$ cd /tmp
[operativos@localhost tmp]$ wget http://www.gutenberg.org/files/16467/16467.txt
--2017-03-06 03:09:59-- http://www.gutenberg.org/files/16467/16467.txt
Resolviendo www.gutenberg.org (www.gutenberg.org)... 152.19.134.47, 2610:28:3090:3000:0:bad:cafe:47
Conectando con www.gutenberg.org (www.gutenberg.org)[152.19.134.47]:80... conectado.
Petición HTTP enviada, esperando respuesta... 200 OK
Longitud: 543858 (531K) [text/plain]
Grabando a: "16467.txt"

100%[=====>] 543.858 140KB/s en 3,8s

2017-03-06 03:10:03 (140 KB/s) - "16467.txt" guardado [543858/543858]

[operativos@localhost tmp]$ exit
logout
Connection to 192.168.85.131 closed.
spawn ssh operativos@192.168.85.129
operativos@192.168.85.129's password:
Last login: Mon Mar  6 03:03:06 2017
[operativos@localhost ~]$ cd /tmp
[operativos@localhost tmp]$ wget http://www.gutenberg.org/files/16467/16467.txt
--2017-03-06 03:10:30-- http://www.gutenberg.org/files/16467/16467.txt
Resolviendo www.gutenberg.org (www.gutenberg.org)... 152.19.134.47, 2610:28:3090:3000:0:bad:cafe:47
Conectando con www.gutenberg.org (www.gutenberg.org)[152.19.134.47]:80... conectado.
Petición HTTP enviada, esperando respuesta... 200 OK
Longitud: 543858 (531K) [text/plain]
Grabando a: "16467.txt"

100%[=====>] 543.858 151KB/s en 3,5s
```

El Script es el siguiente:

```
#!/usr/bin/expect
#Declaro la lista de IPs
set IPUNO "192.168.85.131"
set IPDOS "192.168.85.129"

#Creo la conexión SSH con primer PC
spawn ssh operativos@$IPUNO
expect "operativos@$IPUNO's password:"
send "operativos\r"
expect "$"

#Entro a la carpeta tmp
send "cd /tmp\r"
expect "$"

#Descargo el libro
send "wget http://www.gutenberg.org/files/16467/16467.txt\r"
expect "$"

#Cierro Conexión SSH
send "exit\r"
expect oef

#Creo la conexión SSH con segundo PC
spawn ssh operativos@$IPDOS
expect "operativos@$IPDOS's password:"
send "operativos\r"
expect "$"

#Entro a la carpeta tmp
send "cd /tmp\r"
expect "$"

#Descargo el libro
send "wget http://www.gutenberg.org/files/16467/16467.txt\r"
expect "$"

#Cierro Conexión SSH
send "exit\r"
expect oef
```

Se muestra a continuación, un antes y un después de las dos estaciones en su respectiva carpeta *tmp*.

```
192.168.85.129 192.168.85.131 192.168.85.129
[root@localhost tmp]# ls -l
total 12
-rwx-----. 1 root root 836 mar 5 19:11 ks-script-uXKum6
-rw-----. 1 root root 0 mar 5 19:01 yum.log
-rw-----. 1 root root 4741 mar 6 01:50 yum_save_tx.2017-03-06.01-50.xQBYGr.yumtx
[root@localhost tmp]# ls -l
total 544
-rw-rw-r--. 1 operativos operativos 543858 ago 7 2005 16467.txt
-rwx-----. 1 root root 836 mar 5 19:11 ks-script-uXKum6
-rw-----. 1 root root 0 mar 5 19:01 yum.log
-rw-----. 1 root root 4741 mar 6 01:50 yum_save_tx.2017-03-06.01-50.xQBYGr.yumtx
[root@localhost tmp]# █

192.168.85.129 192.168.85.131 192.168.85.129
[operativos@localhost tmp]$ ls -l
total 8
-rwx-----. 1 root root 836 mar 5 16:47 ks-script-iDCDun
-rwxrwxrwx. 1 root root 805 mar 6 03:07 slop.sh
drwx-----. 2 root root 6 mar 6 01:57 vWMnscu
-rw-----. 1 root root 0 mar 5 16:38 yum.log
[operativos@localhost tmp]$ ls -l
total 540
-rw-rw-r--. 1 operativos operativos 543858 ago 7 2005 16467.txt
-rwx-----. 1 root root 836 mar 5 16:47 ks-script-iDCDun
-rwxrwxrwx. 1 root root 760 mar 6 03:09 slop.sh
drwx-----. 2 root root 6 mar 6 01:57 vWMnscu
-rw-----. 1 root root 0 mar 5 16:38 yum.log
[operativos@localhost tmp]$ █
```

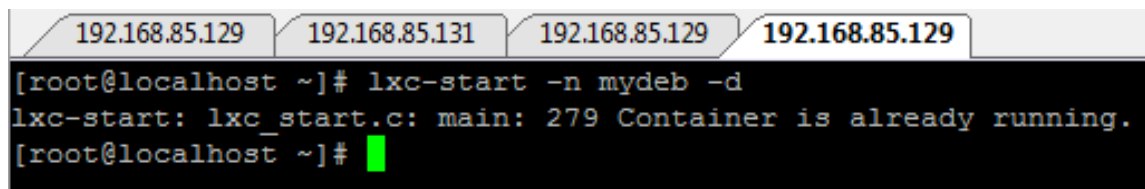
CONTENEDOR DEBIAN – LXC

Para la creación del contenedor Debian, he seguido una guía encontrada en la siguiente URL: <http://www.tecmint.com/install-create-run-lxc-linux-containers-on-centos/>.

La secuencia de comando fue la siguiente:

```
37 yum install epel-release
38 yum install debootstrap perl libvirt -y
39 yum install lxc lxc-templates -y
40 systemctl status lxc.service
41 systemctl start lxc.service
42 systemctl start libvirtd
43 systemctl status lxc.service
44 lxc-checkconfig
45 ls -alh /usr/share/lxc/templates/
46 lxc-create -n mydeb -t debian
47 lxc-start -n mydeb -d
48 lxc-ls --active
49 lxc-ls
50 man lxc
51 lxc-console -n mydeb
```

Al final, al estar en funcionamiento el contenedor, verifiqué que el contenedor estuviese corriendo:



```
192.168.85.129 192.168.85.131 192.168.85.129 192.168.85.129
[root@localhost ~]# lxc-start -n mydeb -d
lxc-start: lxc_start.c: main: 279 Container is already running.
[root@localhost ~]#
```

NOTAS FINALES: Para la realización de este parcial, se han usados dos estaciones virtuales con CentOS 7 v16.11 (Minimal) a 64bits.

MultyPutty Manager.

Putty.