

# Sistemas Operacionales

1<sup>st</sup> Cristian Orlando  
Rincón Bonilla

2<sup>nd</sup> Devisson David  
Vásquez Galeano

3<sup>th</sup> Santiago Andres  
Hernandez Beltran

4<sup>th</sup> Jury Natalia  
Quiceno Valencia

**Abstract**—Este documento contiene la solución de los ejercicios de la semana 7, cuyo objetivo es desarrollar herramientas relacionadas con los temas propuestos que son manejo de múltiples hilos.

**Index Terms**—java, python, threads

## I. TERCERA ENTREGA

Se revisarán tópicos específicos los cuales son:

A. *Elabore un programa para una pequeña aplicación en Java sobre la máquina virtual Linux, con cuatro clases (main, limpiar, generar y consumir), en donde desde el main ejecute la clase, limpie y sincronice dos hilos, uno para generar una variable llamada vacío que tenga el valor “true”, pero que el hilo consumidor se quede esperando y, cuando cambie a “false” en el hilo generador, entonces dicha variable sea tomada por el hilo consumidor.*

B. *Monte una red interna entre las dos máquinas virtuales.*

C. *Haga una transferencia de archivos entre las máquinas virtuales usando el comando scp.*

D. *Monte un sistema de archivos NFS.*

E. *Elabore una copia de seguridad de las carpetas / etc. Envíe al tutor el log de compresión.*

## II. SOLUCIÓN

A. *Elabore un programa para una pequeña aplicación en Java sobre la máquina virtual Linux, con cuatro clases (main, limpiar, generar y consumir), en donde desde el main ejecute la clase, limpie y sincronice dos hilos, uno para generar una variable llamada vacío que tenga el valor “true”, pero que el hilo consumidor se quede esperando y, cuando cambie a “false” en el hilo generador, entonces dicha variable sea tomada por el hilo consumidor.*

- Código fuente del programa
- Documentación
- Demo

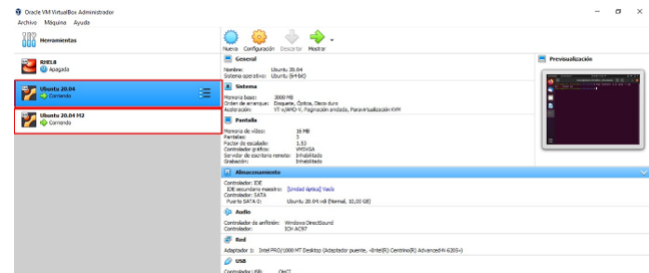


Fig. 1. Máquinas virtuales conectadas por red

B. *Monte una red interna entre las dos máquinas virtuales. "fig. 1"*

C. *Haga una transferencia de archivos entre las máquinas virtuales usando el comando scp. "fig. 2,3,4,5,6 y 7"*

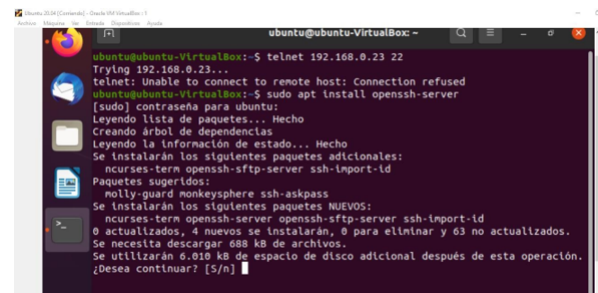


Fig. 2. Instalamos el servidor ssh

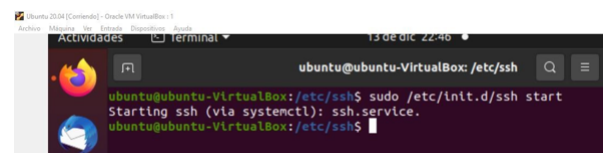


Fig. 3. Arrancamos el servidor ssh

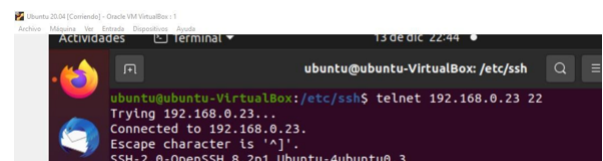


Fig. 4. Probamos que este habilitado el puerto de transferencia (1)

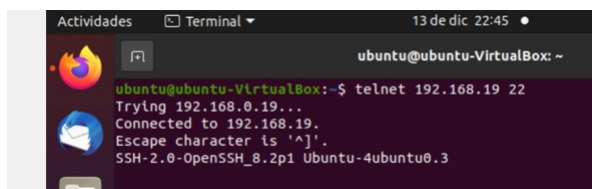


Fig. 5. Probamos que este habilitado el puerto de transferencia (1)

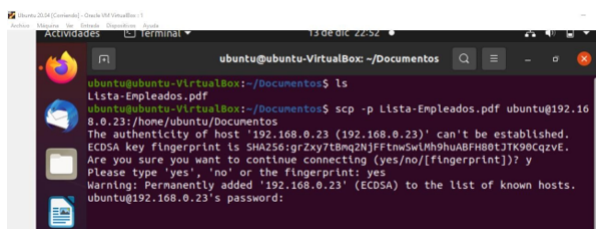


Fig. 6. Transferimos el archivo Lista-Empleados.pdf del server Ubuntu 20.04 (192.168.19) al server Ubuntu 20.04 M2 (192.168.23).

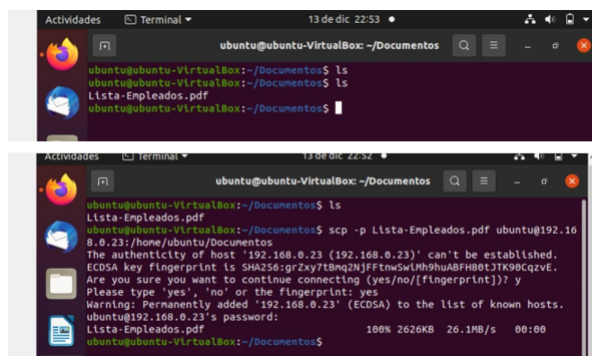


Fig. 7. Transferimos el archivo Lista-Empleados.pdf del server Ubuntu 20.04 (192.168.19) al server Ubuntu 20.04 M2 (192.168.23).

#### D. Monte un sistema de archivos NFS.

1) Para montar el sistema de archivos necesitamos dos máquinas mínimo, un cliente y un servidor:

- Servidor: 192.168.19
- Cliente: 192.168.23

```
Creating config file /etc/idnmapd.conf with new version
Añadiendo el usuario del sistema 'statd' (UID 122) ...
Añadiendo un nuevo usuario 'statd' (UID 122) con grupo 'nogroup' ...
No se crea el directorio personal '/var/lib/nfs'.
Created symlink /etc/systemd/system/multi-user.target.wants/nfs-client.target →
/lib/systemd/system/nfs-client.target.
Created symlink /etc/systemd/system/remote-fs.target.wants/nfs-client.target →
/lib/systemd/system/nfs-client.target.
nfs-utils.service is a disabled or a static unit, not starting it.
Configurando nfs-kernel-server (1:1.3.4-2.1ubuntu5.5) ...
Created symlink /etc/systemd/system/multi-user.target.wants/nfs-server.service →
/lib/systemd/system/nfs-server.service.
Job for nfs-server.service canceled.

Creating config file /etc/exports with new version
Creating config file /etc/default/nfs-kernel-server with new version
Progress: 100% [#####]
```

Fig. 8. Instalamos los drivers para manejar el sistema de nfs

```
mount 192.168.19:/home /mnt/nfs/home
```

7) Montamos el directorio compartido con el comando:  
Podemos usar el comando `df -h` para revisar que el directorio se montó correctamente. en cuyo caso debería aparecer en pantalla algo similar a:

```
Archivo Editor Ver Buscar Terminal Ayuda
root@natallia-Virtual-Machine: /home/natallia
root@natallia-Virtual-Machine: /home/natallia# exportfs -a
root@natallia-Virtual-Machine: /home/natallia# systemctl restart nfs-kernel-server
root@natallia-Virtual-Machine: /home/natallia#
```

Fig. 10.

```
Filesystem Size Used Avail Use
% Mounted on /dev/sda 20G 783M 18G 5% /
192.168.19:/home 20G 785M 18G 5% /mnt/nfs/home
```

E. Elabore una copia de seguridad de las carpetas / etc. Envíe al tutor el log de compresión.

- Logs de información
- Logs de errores

#### REFERENCES

- [1] Oracle.com. “VirtualBox” [online] <https://www.oracle.com/lad/virtualization/solutions/t/oracle-vm-virtualbox/> [Accedido 28 Junio 2022].
- [2] mirantis.com. “Crear servidores con VirtualBox” [online] <https://www.mirantis.com/kubernetes-lab/how-to-create-a-server-on-virtualbox/> [Accedido 28 Junio 2022].

2) Instalamos los drivers para manejar el sistema de nfs:

```
1 yum install nfs-utils nfs-utils-lib
```

```
1 chkconfig nfs on service rpcbind start service nfs start
```

3) Inicializamos los programas necesarios:

4) Seleccionamos el directorio que se va a compartir con el servidor cliente: Exportamos el directorio agregando la siguiente línea al archivo `/etc/exports`

```
1 /home 192.168.19 { rw, sync, no_root_squash, no_subtree_check }
```

Aplicamos la exportación del directorio con el siguiente comando:

```
1 exportfs -a
```

5) Configurar NFS del lado del cliente: Realizamos los pasos de instalación vistos en el paso D.2.

```
1 mkdir -p /mnt/nfs/home
```

6) Creamos el directorio compartido con el comando:

```
root@natallia-Virtual-Machine: /home/natallia# mkdir /var/nfs/compartido
root@natallia-Virtual-Machine: /home/natallia# ls -la /var/nfs/compartido
total 8
drwxr-xr-x 2 root root 4096 jun 28 16:03 .
drwxr-xr-x 3 root root 4096 jun 28 16:03 ..
```

Fig. 9.