Laboratorio de
Sistemas Operativos

Práctica 4B: Administración

Actividad 4-7:

Interfaces de red

Existen algunas aplicaciones que realizan el trabajo de conectar una computadora con Linux a una conexión de red que se encuentre disponible, sin embargo algunas ocasiones es necesario hacerlo de forma manual. A continuación aprenderás a configurar interfaces ethernet wired y wireless.

Las configuraciones manuales de las interfaces involucran entender cómo funciona el protocolo de red (IP) y el protocolo de la capa de datos (Ethernet/802.3 o los de 802.11x), por lo tanto en ocasiones puede resultar un poco complejo cuando no se tienen estos conocimientos.

Alámbricas (Wired)

Para configurar **interfaces alámbricas** se requiere primero levantar la interfaz y después solicitar una dirección de red. Los pasos son los siguientes:

1.-Ver cuáles interfaces tiene instaladas el sistema, usando el comando: **ifconfig –a**

Aquí buscamos el nombre de la interface **ethernet**, que generalmente se llama **eth0** (si tienes muchas interfaces es posible que cambie la numeración).Si tienes duda acerca del uso del comando **ifconfig** puedes ver una ayuda rápida del mismo con la bandera **--help**, o la ayuda completa con el comando **man ifconfig**

2.- Encender la interface con el comando: **ifconfig <Interfaz> up**
Podemos ver que esta encendida si buscamos la palabra **UP** en la salida que nos entrega el comando: **ifconfig <Interfaz>**

3.-Después pedimos una dirección ip al servidor DHCP con el comando: **dhclient <Interfaz>**
Algunas veces, esta aplicación no esta instalada y en su lugar usamos: **dhcpcd <Interfaz>**

En caso de que encuentres un error que te indique que no se puede renovar la dirección IP, deberás ejecutar cualquiera de los comandos anteriores con la bandera **"-r",** por ejemplo: **dhclient –r**

4.- Verificamos se nos haya asignado una dirección IP con el comando **ifconfig** en el campo **inet addr**:

ifconfig <Interfaz>

Inalámbricas

Para configurar **interfaces wireless**, se requiere levantar la interfaz, conectarnos a un punto de acceso en particular y luego solicitar dirección de red. Los pasos son los siguientes:

- Ver cuáles interfaces con capacidades wireless tiene instaladas el sistema: **iwconfig**

Las interfaces que muestren un valor diferente a "**no wireless extension"** son las interfaces que son de tipo wireless. Algunos nombres comunes son: "at0, eth1, wlan0, …".

- Identificamos el nombre de la interface y apagamos la interface para configurarla, ya que algunos drivers no soportan el cambio de la configuración mientras la interface está encendida: **ifconfig <Interfaz> down**

- Procedemos a buscar los Access Points (antenas) que tenemos al alcance y tomamos los datos del que nos interesa: **iwlist <Interfaz> scan**

Estos datos son: **essid** y **channel.**

- Configuramos la interface wireless: **iwconfig <Interfaz> essid <essidDelAp> channel <canalDelAP>**
  Si el AP tiene seguridad WEP o WAP se especifica la llave que queremos usar con el comando:
  iwconfig <Interfaz> key s:<claveWEP>

Verificamos que todo haya quedado bien configurado con el comando: **iwconfig <Interfaz>**

- Encendemos la interface y solicitamos una dirección IP, utilizando dhclient o dhcpcd:
  ifconfig <Interfaz> up

TIP*: Como podrás ver esto de repente no es tan sencillo , pero desviando la salida del escaneo de puntos de acceso a un archivo de entrada a awk para filtrar resultados y obtener solo access points válidos y se puede automatizar el proceso de conexión.*

Muy probablemente encuentres direcciones que empiecen con FE80:--- y/o 2000:--- éstas son direcciones del nuevo protocolo de red IPv6 y si la interfaz lo soporta, mínimo activará una que inicie con FE80 y si la red ya tiene implementado dicho protocolo, muy seguramente se auto-asignará una que inicie con 2000.

Actividad 8,9:

Montar dispositivos de almacenamiento

En la actualidad los sistemas Linux montan de forma automática las unidades de memoria Flash

USB o los discos duros externos; sin embargo es bueno que sepas hacerlo de forma manual. Un ejemplo de la utilidad de esto es el poder montar un archivo ISO (imagen bit por bit de un CD) en una carpeta, ahorrándote la necesidad de quemar un CDROM y aumentando la velocidad de acceso a los archivos dentro de la imagen ISO.

Memorias Flash USB.

- Primero, se conecta la memoria a la computadora. Luego se debe revisar el nombre del dispositivo que esta tomó, con el comando: **dmesg | tail**

La Imagen 2 muestra un ejemplo de la salida que deberías obtener de este comando. Como puedes ver en la última línea, la escritura en la memoria Flash se hará a través del dispositivo: /dev/sdb1

Imagen 2

Cada dispositivo de entrada o salida en Linux se considera como un archivo o sistema de archivos, incluyendo dispositivos como el teclado y la pantalla y los sockets (canales de transmisión de datos entre procesos), además de los CDROMS o discos duros o lo que pueda venir en el futuro. El directorio **/mnt** se utiliza para montar sistemas de archivos temporalmente. **/media** se utiliza para medios que pueden conectarse o desconectarse mientras la computadora está prendida, ej: CDROM, memorias flash.

Generalmente el directorio **/media** tiene /media/cd, /media/dvd y /media/fl que pueden ser utilizados para “montar” dispositivos. **“Montar” la unidad significa poder ver su contenido como si fuera un directorio más en el sistema de archivos de Linux**. En el caso en que no se tengan dichos directorios se pueden crear a conveniencia.Este es el camino que se utilizará en la práctica:

- Montarás tu memoria flash en el directorio llamado /media/memoriaFlash que vas a crear enseguida:

  su
  cd /media
  mkdir memoriaFlash
  ls

- Ahora se procede a montar la unidad:

  mount /dev/<nombreUnidadFlash> /media/memoriaFlash

  Generalmente, como se vio en la Imagen 2, el nombre de la unidad Flash tiene el formato: sda#, sdb#.

- Si todo salió bien, ahora puedes ver el contenido de la memoria Flash:

  ls /media/memoriaFlash

- Para desmontar la memoria Flash y evitar cualquier daño a los datos escritos, es necesario desmontar correctamente la memoria Flash con el comando:

  umount /media/memoriaFlash

  No es necesario borrar el directorio *memoriaFlash*, puede ser re-usado.

Archivos ISO

- Se crea un subdirectorio, siendo **super usuario**, dentro del directorio **/mnt,** donde se montará esta unidad:

  su
  cd /mnt
  mkdir archivoisomontado
  ls

- Puedes utilizar cualquier imagen de disco, por ejemplo la del disco de instalación que usaste para instalar linux en tu computadora. En esta práctica le llamaremos imagendeisoprueba.iso.

- A continuación, estando en el directorio donde se encuentra el archivo **iso**, se procede a montar el archivo **iso** con el siguiente comando:

  mount –o loop imagendeisoprueba.iso /mnt/archivoisomontado
  ls /mnt/archivoisomontado/

Trata de ejecutar el archivo: /mnt/archivoisomontado/imageniso/pinball

- Para desmontar el archivo **iso** ejecuta el comando:

  umount /mnt/archivoisomontado

Discos duros

Para montar particiones o discos duros de un sistema Linux, algunas veces es necesario conocer el tipo de **filesystem** que tiene para pasárselo como argumento a mount. Como ejemplo se muestra la siguiente sintaxis:

su

```
cd /mnt
mkdir discoDuro1
ls
mount –t <filesystem>  <device> <dir>
umount <dir>
```

Donde:

- **filesystem,** puede ser por ejemplo: vfat, ext3, ntfs, hfs;
- **device,** puede ser alguno de tipo: /dev/sdb1, /dev/hda2, /deb/hdb3;
- **dir**, es el directorio donde se quiere montar.

Ver usuarios conectados

Los sistemas GNU/Linux **son multi-usuarios en tiempo real**; esto significa que pueden por defecto soportar múltiples usuarios (humanos) accediendo de forma simultánea a los recursos del sistema desde el mismo equipo o de forma remota. Por lo mismo existen ciertas herramientas para facilitar este ambiente.

En la práctica sobre shell programming tuviste contacto con el comando **who**, que permite ver qué usuarios están conectados a una computadora. Este comando arroja información sobre el nombre de usuario, la terminal a la que está conectado el usuario, la última fecha de acceso, y la dirección desde donde se realizó la conexión. Por ejemplo:

```
[user@localhost home]$ who
operator    tty0         Jul 15 17:44
al702782    pts/1        Aug 05 19:26          (a estas maquinas)
al194834    pts/2        Aug 05 19:29          (131.178.74.107)
al159884    pts/4        Aug 05 19:13          (a estas maquinas)
al333793    pts/7        Aug 05 19:19          (131.178.74.149)
al172871    pts/10       Aug 05 18:35          (131.178.82.22)
amartine    pts/12       Aug 05 19:45          (tserv.mty.itesm.)
jnolazco    pts/13       Aug 05 19:55          (ciencias.mty.ite)
```

Existe otro, el comando **w**, que muestra información más detallada sobre los usuarios conectados a la computadora, puedes ver que la descripción de los campos con el comando man w.

```
[user@localhost home]$ w
11:22:57 up 11 days, 19:17,  3 users,  load average: 0.00, 0.01, 0.00
USER     TTY      FROM                LOGIN@   IDLE   JCPU   PCPU WHAT
robertoa pts/0    10-11-12-13 11:09    0.00s  0.30s  0.00s w
sgeadmin tty8     :0          Mon16   23:45  48.21s  0.08s x-session-manager
```

TIP: *El usuario Root puede terminar lo que estos usuarios están haciendo mediante kill (aunque primero deberá obtener el process-ID del proceso del usuario).*

Enviando mensajes a usuarios.

En algunas ocasiones es necesario mandar mensajes a la terminal de otros usuarios conectados a la computadora, por ejemplo para avisarles que se reiniciará el sistema. El comando que permite esta tarea es write, y su sintaxis es la siguiente:

```
write <nombre-de-usuario>  <ENTER>
```

```
cualquier mensaje …
<CTRL+D>
```

El comando **write** permite la comunicación en un solo sentido, por lo que si un usuario desea responder los mensajes que otro le envía debe también responder mediante el comando **write**. Es una especie de chat. Para terminar la comunicación se deben presionar las teclas <CTRL+D>

A continuación se muestra un ejemplo de una conversación entre dos usuarios, puedes seguir la secuencia mediante los números entre paréntesis:

```
(01) [robertoaceves@sge-handler32 ~]$ write paola
(02) hola
(05) como ça va?

(09) Message from paola@sge-handler32 on pts/1 at 11:51 ...
(10) ça va bien
(11) ok
(13) au revoir
(15) bye
(17) <CTRL+D> EOF

[robertoaceves@sge-handler32 ~]$


paola@sge-handler32:~$
(03) Message from robertoaceves@sge-handler32 on pts/0 at 11:50 ...
(04) hola
(06) como ?ça va??

(07) paola@sge-handler32:~$ write robertoaceves
(08) ça va bien
(12) ok
(14) au revoir
(16) bye
(18) EOF
paola@sge-handler32:~$
```

Existe, también un comando que permite mandarle mensajes a las terminales de todos los usuarios: **wall**, su sintaxis es la siguiente:

```
[user@localhost home]$ wall
mensaje
<CTRL+D>
```

Este comando manda un mensaje en ***broadcast*** a todos los usuarios, y al igual que el comando anterior, permite la comunicación en solo un sentido.

TIP: Los usuarios que no tienen abierta alguna terminal no recibirán un aviso, lo cual les puede *ser contraproducente si se ejecuta el comando para apagar la máquina y estos no pudieron guardar sus trabajos.*

Programando la ejecución de tareas

Como administrador, o como usuario, alguna vez tendrás la necesidad de programar la ejecución de algún comando a determinada hora y día. Para ello existen algunos comandos en los sistemas Unix-like, como: at, cron, nice.

El comando **at** permite ejecutar un comando a un tiempo específico. Puedes consultar la lista completa de opciones con el comando **man at**. La sintaxis del comando es la siguiente:

```
[user@localhost home]$ at <hora> <fecha>
at> comando-a-ejecutar
at> uno-por-línea
at> <CTRL+D>
```

Donde las opciones son:

- **<hora>** : Hora a la que se ejecutará el comando, con formato HH:MM
- **<fecha>** : Día que se ejecutará el comando, con formato DD.MM.YY , si se omite se toma como referencia el día de hoy.

“at>” simboliza el prompt del comando at, aquí es donde se teclean los comandos, uno por línea, que se desean ejecutar. Para terminar de capturar comandos hay que presionar las teclas <CTRL+D> .

Este comando no produce salida, cualquier mensaje que se imprima a la salida estándar o cualquier error se registrará en el archivo: /var/mail/nombre-de-usuario

Se muestra un ejemplo del uso del comando at a continuación:

```
[user@localhost home]$ ls
[user@localhost home]$
[user@localhost home]$ date
Mon Sep  7 18:20:44 CDT 2009

[user@localhost home]$ at 18:29 07.09.09
warning: commands will be executed using /bin/sh
at> echo "Esta es una prueba del comando at, que ejecuta un comando en el tiempo programado"
at> mkdir cosas
at> cd cosas
at> echo "1 2 3 4 5" > numeros.txt
at> <EOT>
job 5 at Mon Sep  7 18:29:00 2009

[user@localhost home]$ date
Mon Sep  7 18:25:31 CDT 2009
[user@localhost home]$ date
Mon Sep  7 18:27:12 CDT 2009
[user@localhost home]$ You have mail in /var/mail/nolazco
[user@localhost home]$ date
Mon Sep  7 18:29:32 CDT 2009

[user@localhost home]$ tail /var/mail/nolazco
             id 1MlA7Q-0000G3-6d
             for nolazco@sge-handler32.mty.itesm.mx; Mon, 07 Sep 2009 18:29:00 -0500
Subject: Output from your job        5
To: nolazco@sge-handler32.mty.itesm.mx
Message-Id: <E1MlA7Q-0000G3-6d@sge-handler32.mty.itesm.mx>
From: nolazco@sge-handler32.mty.itesm.mx
Date: Mon, 07 Sep 2009 18:29:00 -0500

Esta es una prueba del comando at, que ejecuta un comando en el tiempo programado
```

```
[user@localhost home]$ ls
cosas
[user@localhost home]$ ls cosas
numeros.txt
[user@localhost home]$ cat cosas/numeros.txt
1 2 3 4 5
[user@localhost home]$
```

Actividad 10:

Respaldos.

Dado que lo más importante que existe en una computadora o servidor es la información que contiene, es vital hacer respaldos de modo periódico. Para una computadora personal, es recomendable tener al menos un respaldo para prevenir pérdida de datos en caso de una falla del disco duro.Una herramienta que facilita esta labor (sobre todo si se utiliza junto con **cron**) es **rsync**, ya que es una forma de respaldo rápida y versátil de copiar archivos ya sea de modo local o de modo remoto.

Una ventaja que tiene **rsync** sobre otras herramientas (**scp** o **ftp**) es su capacidad de manejar información de modo diferencial, esto significa que puede encontrar qué necesita ser transmitido haciendo una búsqueda de los archivos que han cambiado en el origen de la sincronización y sólo transfiere esos archivos.

Su uso más común está definido del siguiente modo: `rsync -avz /src/foo/ /dest/foo`

De este modo se sincroniza /dest/foo con los cambios hechos en /src/foo. Esto se puede hacer a través de la red incluyendo la máquina en la dirección de origen o destino e.g:

```
rsync -avz /src/foo/ maquina-remota:/dest/foo
```

```
mkdir orig dest
cd orig
wget http://launchpad.net/exaile/0.3.0/0.3.0.1/+download/exaile-0.3.0.1.tar.gz
tar xvfz exaile-0.3.0.1.tar.gz
rsync -avz ./ ../dest
echo "prueba" >> exaile-0.3.0.1/README
rsync -avz ./ ../dest
```

Nota como solo se transfiere el archivo README ya que es lo único que se ha modificado.

Actividad 11,12:

Conexiones remotas.

El modo en el que comúnmente se manejan los servidores es teniendo una computadora corriendo servicios sin periféricos conectados a ella, esto es, sin teclado o monitor. La administración de estos servicios se hace de modo remoto gracias a herramientas como **ssh**.

ssh es un programa que permite hacer conexiones de ingreso remotas a los equipos que estén corriendo dicho servicio, de un modo seguro. Una vez conectado a dicha máquina y autenticado como usuario, uno es capaz de ejecutar comandos como si estuviera físicamente junto a la computadora. La sintaxis de este comando es: `ssh usuario@maquina-remota`

Donde ***usuario*** es tu nombre de usuario, ***máquina-remota*** es una dirección ip o un nombre de dominio. Por ejemplo, para conectarse a la máquina *cluster.itesm.mx* con el usuario *pedro:*

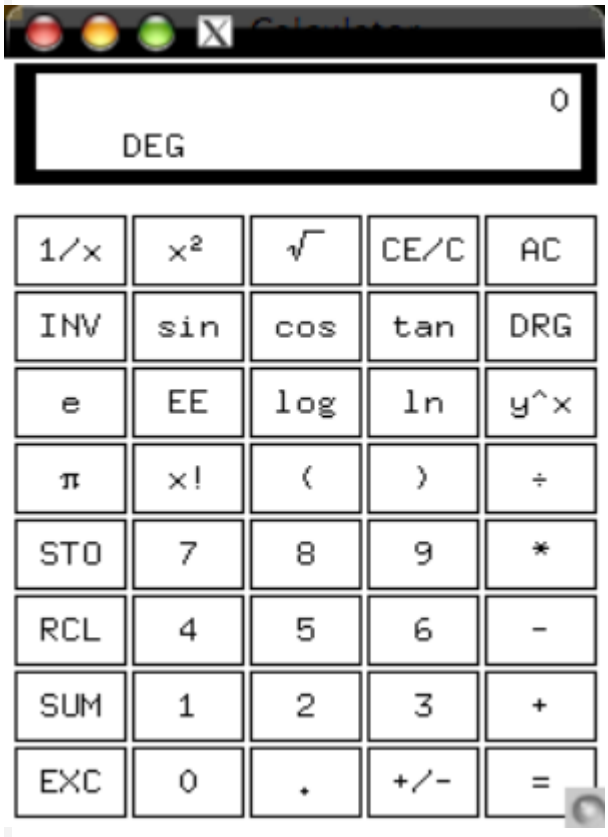
```
ssh  pedro@cluster.itesm.mx
```

Si el servidor tiene instalado Xorg (servidor del sistema de ventanas X) y el ancho de banda de la red es suficiente, uno puede, a través de **ssh**, ejecutar programas con interfaz gráfica en la computadora remota y verla desplegada en la sesión de X local. Esto se logra con la opción "**-XYg"** de **ssh**.

```
ssh -XYg usuario@maquina-remota
```

Por ejemplo, si deseas ejecutar la calculadora (xcalc) de la máquina *cluster.itesm.mx* y tu usuario es *pedro* lo harías de la siguiente manera:

```
[user@localhost home]$ ssh  -XYg  pedro@cluster.itesm.mx
[pedro@cluster.itesm.mx home]$ xcalc &
```



scp es una herramienta que permite transferir archivos de un modo seguro a una o desde una computadora remota utilizando el protocolo de **ssh**. Este es uno de los métodos más comunes

para subir archivos individuales a los servidores. La sintaxis de este comando es:

```
scp archivo-origen  usuario@maquina-remota:/ruta/al/destino
```

Información adicional

En este blog se puede hallar información de la configuración de visudo, que con un manejo adecuado permite crear un sistema de control del comando **SUDO** de forma eficiente.

Comandos de administración y más allá...

Lo mostrado hasta ahora es solo una pequeña introducción. Algunos comandos útiles para administración se pueden hallar en la siguiente liga: app_guide/unix_commands.html