LFS201

Fundamentos de Administración de Sistemas Linux

Versión 1.0

LFS201: Version 1.0

Publicado por:

Linux Foundation
http://www.linuxfoundation.org

Contenidos

Capitulo 1

Prólogo

Lab 1.1: Configurar el sistema con sudo

Es muy peligroso ejecutar una **terminal de root** a menos de que sea absolutamente necesario: un solo error de tipeo o de otro tipo puede causar daños graves (incluso no recuperables).

Por lo tanto, el procedimiento recomendado es configurar el sistema de tal forma que comandos únicos puedan ser ejecutados con privilegios de superusuario, a través del mecanismo de **sudo**. Con **sudo** el usuario necesita conocer su propia clave solamente y nunca la del usuario root.

Si usted está usando una distribución como **Ubuntu**, es posible que no necesite realizar este laboratorio para tener **sudo** configurado de forma apropiada para el curso. Sin embargo, todavía necesita asegurarse de comprender el procedimiento.

Para comprobar si su sistema ya está configurado para permitir que la cuenta de usuario que está usando ejecute **sudo**, ejecute un comando simple como el siguiente:

```
$ sudo ls
```

Se le debería pedir la clave de usuario y luego el You should be prompted for your user password and then el comando sería ejecutado. Si en cambio obtiene un mensaje de error, entonces necesitará realizar el siguiente procedimiento.

Inicie una terminal de root a través del comando **su** y luego provea la clave de **root**, no su clave de usuario.

En todas las distribuciones recientes de **Linux** usted debería ir al subdirectorio /etc/sudoers.d y crear un archivo, generalmente con el nombre del usuario al cual root desea concederle acceso a **sudo**. Sin embargo, esta convención no es realmente necesaria, ya que **sudo** escaneará todos los archivos en este directorio. El archivo puede contener algo tan simple como lo siguiente:

```
estudiante ALL=(ALL)  ALL
```

si el usuario es `estudiante`.

Una práctica antigua (la que aún funciona) es agregar la línea al final del archivo /etc/sudoers. Lo más recomendable es hacerlo con el programa **visudo**, ya que se ocupa de que usted esté usando la sintaxis adecuada mientras edita.

Es posible que necesite configurar los permisos adecuados en el archivo, de la siguiente forma:

```
$ chmod 440 /etc/sudoers.d/estudiante
```

Tenga en cuenta que algunas distribuciones **Linux** pueden requerir permisos `400` en vez de `440` .

Luego de haber realizado esos pasos, salga de la consola root con `exit` e intente de nuevo `sudo ls` .

Hay muchas otras cosas que un administrador puede configurar en **sudo**, incluyendo permisos para usuarios específicos, limitar las búsquedas a ciertos directorios, etc. El archivo /etc/sudoers está muy bien autodocumentado.

Sin embargo, hay un ajuste adicional que recomendamos altamente que realice, aún si su sistema ya tiene configurado **sudo**. La mayoría de las distribuciones establecen directorios diferentes para los directorios en donde se encuentran los ejecutables de los usuarios normales y los de root. En particular los directorios /sbin y /usr/sbin no son encontrados en las búsquedas, ya que **sudo** hereda el `PATH` del usuario, no del superusuario.

Por lo tanto, en este curso estaremos constantemente recordándole la ruta completa de varias herramientas de administración; cualquier otra mejora en cuanto a la seguridad de esta implementación probablemente no valdrá la pena (como intentar esconder los binarios del superusuario, por ejemplo).

En consecuencia, sugerimos agregar la siguiente línea al archivo .bashrc en su directorio de usuario:

```
PATH=$PATH:/usr/sbin:/sbin
```

No es necesario que reinicie, en vez de eso, puede salir de la sesión y entrar nuevamente, lo cual será completamente efectivo.

Capitulo 2

Inicio y Apagado del Sistema

Lab 2.1: shutdown vs. halt vs. reboot

Nota: este ejercicio requiere ser ejecutado desde la consola, es decir, no a través de la red usando SSH.

1. Lleve el sistema a modo monousuario usando el comando **shutdown**.
2. Desde el modo monousuario, reinicie el sistema con el comando **shutdown**.
3. Una vez que el sistema haya reiniciado, apáguelo completamente usando el comando **shutdown**.
4. Vuelva a iniciar el sistema.

Solution 2.1

1. `$ sudo shutdown now`
2. `$ sudo shutdown -r now`
3. `$ sudo shutdown -h now`
4. Presione el botón de apagado o reinicie su máquina virtual.

Capitulo 3

GRUB

Lab 3.1: Inicio en modo no gráfico usando GRUB

Nota: Este ejercicio requiere ser ejecutado desde la consola, es decir, no a través de **SSH**.

1. Reinicie su máquina y vaya a la consola interactiva de **GRUB** presionando **e**, o cualquier otra tecla que sea requerida para tal efecto, según se indique en la pantalla.

2. Haga que su sistema inicie en modo no gráfico. La forma de hacerlo dependerá de su sistema específico.

   En sistemas tradicionales que respetan los **runlevels** (los cuales trataremos en la próxima sección) es posible agregar un `3` a la línea de comandos del kernel, en la entrada específica que seleccionó desde las opciones en el menú de **GRUB**.

   En otros sistemas (incluyendo **Ubuntu**) es necesario agregar `text` en cambio.

3. Teclee la tecla apropiada para que el sistema continue iniciando.

4. Luego de que el sistema está completamente operacional en modo no gráfico, llévelo a modo gráfico. Dependiendo de su sistema, uno de los siguientes comandos debería hacerlo:

   ```
   $ sudo telinit 5
   $ sudo service gdm restart
   $ sudo service lightdm restart
   ```


Capitulo 4

init: SystemV, Upstart, systemd

Lab 4.1: Agregar un servicio de arranque nuevo con SysVinit

En este y sl siguiente ejercicio crearemos un servicio simple de inicio. Primero lo haremos para un sistema **SysVinit**. Tenga en cuenta que si está usando un sistema basado en **systemd**, todo debería funcionar debido a la capa de compatibilidad hacia atrás que todas las distribuciones tienen. Sin embargo, lo haremos de forma nativa **systemd** en el próximo ejercicio.

Si está en un sistema basado en **Debian** como **Ubuntu**, asegúrese de tener instalados los paquetes **sysvinit-utils** y **chkconfig**. Sin embargo, las versiones recientes de **Ubuntu** ya no proveen el paquete **chkconfig**; en cambio, tendrá que usar la herramienta **update-rc.d**.

Primero es necesario crear el script específico del servicio; usted lo puede crear por sí solo en caso que lo desee, o utilizar el procedimiento que se describe a continuación (como root). En este último caso cree un archivo llamado `/etc/init.d/fake_service` con el siguiente contenido:

```
#!/bin/bash
# fake_service
# Starts up, writes to a dummy file, and exits
#
# chkconfig: 35 69 31
# description: This service doesn't do anything.
# Source function library

. /etc/sysconfig/fake_service

case "$1" in
 start) echo "Running fake_service in start mode..."
   touch /var/lock/subsys/fake_service
   echo "$0 start at $(date)" >> /var/log/fake_service.log
   if [ ${VAR1} = "true" ]
   then
     echo "VAR1 set to true" >> /var/log/fake_service.log
   fi
```

```
   echo
   ;;
 stop)
  echo "Running the fake_service script in stop mode..."
  echo "$0 stop at $(date)" >> /var/log/fake_service.log
  if [ ${VAR2} = "true" ]
  then
      echo "VAR2 = true" >> /var/log/fake_service.log
  fi
  rm -f /var/lock/subsys/fake_service
  echo
  ;;
*)
  echo "Usage: fake_service {start | stop}"
  exit 1
esac
exit 0
```

Si está tomando la versión autodidacta en línea del curso, el script está disponible para descargarlo desde la pantalla **Lab**.

Asígnele permisos de ejecución (y todos los otros permisos necesarios) al archivo anterior:

```
$ sudo chmod 755 /etc/init.d/fake_service
```

Se dará cuenta que el script incluye el archivo `/etc/sysconfig/fake_service`. En sistemas diferentes a **RHEL** deberá cambiar esto a `/etc/default/fake_service`. Créelo e inserte el siguiente contenido:

```
VAR1="true"
VAR2="true"
```

Compruebe si el script funciona correctamente a través de los siguientes comandos:

```
$ sudo service fake_service
$ sudo service fake_service start
$ sudo service fake_service stop
```

Revise el archivo /var/log/fake_service.log. ¿Qué contiene?

Por diversión podría agregar modos adicionales al archivo de script como `restart` ; eche un vistazo a otros scripts en el directorio para obtener ejemplos de lo que puede hacer.

Lo siguiente que necesitamos es que el servicio **fake_service** se inicie cada vez que el sistema arranque y se detenga cuando el sistema se apaga. Si usted hace:

```
$ sudo chkconfig --list fake_service
```

obtendrá un error debido a que aún el servicio no se ha configurado. Puede hacerlo de forma fácil como sigue:

```
$ sudo chkconfig --add fake_service
```

ahora puede habilitarlo y deshabilitarlo en el arranque (respectivamente) de la siguiente forma:

```
$ sudo chkconfig fake_service on
$ sudo chkconfig fake_service off
```

Para probarlo completamente tendrá que reiniciar el sistema con el fin de confirmar si el servicio inicia automáticamente. También puede modificar los runlevels en los cuales el servicio debería ejecutarse.

Lab 4.2: Agregar un servicio de arranque nuevo con systemd

Como se mencionó en el ejercicio anterior, todavía es posible utilizar el script de inicio **SysVinit** con **systemd**, pero esto está en desuso.

El procedimiento análogo consiste en crear un archivo (como root) directamente bajo /etc/systemd/system o en otro lugar en ese árbol de directorio; somewhere else in that directory tree; las distribuciones varían un poco en esto. Este es un ejemplo de un archivo con contenido mínimo, llamado /etc/systemd/system/fake2.service:

```
[Unit]
Description=fake2
After=network.target

[Service]
ExecStart=/bin/echo Estoy iniciando el servicio fake2
ExecStop=/bin/echo Estoy deteniendo el servicio fake2

[Install]
WantedBy=multi-user.target
```

Existen muchas configuraciones que se pueden realizar en el archivo **unit**. El parámetro `After=network.target` significa que el servicio debería ser iniciado sólo después de que la red lo hizo, mientras que `WantedBy=multi-user.target` significa que debería iniciarse cuando se alcanzó el modo multiusuario. Esto es equivalente a runlevels 2 y 3 en **SysVinit**. Tenga en cuenta que `graphical.target` se correlaciona con runlevel 5.

Cambie los permisos del archivo para hacerlo ejecutable:

```
$ chmod 755 /etc/systemd/system/fake2.service
```

Ahora lo único que tenemos que hacer es iniciar el servicio, comprobar su estado y detenerlo:

```
$ sudo systemctl start fake2.service
$ sudo systemctl status fake2.service
$ sudo systemctl stop fake2.service
```

Si usted hizo cambios a la sección unit, debe hacer lo siguiente para recargar el servicio con la información nueva:

```
$ sudo systemctl daemon-reload
```

y el sistema le mostrará una advertencia.

Para habilitar/deshabilitar que el servicio inicie durante el arranque, puede usar los siguientes comandos:

```
$ sudo systemctl enable fake2.service
$ sudo systemctl disable fake2.service
```

Una vez más, es necesario reiniciar el sistema para asegurarse que los cambios realizados están siendo efectivos.

Capitulo 5

Diseño del árbol del sistema de archivos de Linux

Lab 5.1: Tamaños de los directorios de Linux por defecto

Use la herramienta **du** para calcular el tamaño total de cada uno de los directorios de primer nivel de su sistema.

Ejecute el comando:

```
$ du --help
```

para tener una idea de cómo obtener y mostrar la información de forma eficiente.

Solution 5.1

Para obtener una lista completa de los directorios principales bajo / y sus tamaños:

```
$ sudo du --max-depth=1 -hx /
```

```
4.3M    /home
16K     /lost+found
39M     /etc
4.0K    /srv
3.6M    /root
178M    /opt
138M    /boot
6.1G    /usr
1.1G    /var
16K     /mnt
```

```
4.0K    /media
869M    /tmp
8.4G    /
```

Donde hemos usado las siguientes opciones:

- `--maxdepth=1`: Baja un nivel solamente desde / y suma recursivamente todo lo que hay bajo ese árbol.
- `-h`: Provee números legibles para humanos (KB, MB, GB).
- `-x` Permanezca en un sistema de archivos; no busque directorios que no están en la partición /. En este caso significa ignorar:

  ```
  /dev /proc /run /sys
  ```

  debido a que son seudosistemas de archivos que existen en memoria solamente; de hecho cuando el sistema no está corriendo son puntos de montaje vacíos. Debido a que este es un sistema **RHEL 7**, los siguientes puntos de montaje tampoco son tomados en cuenta:

  ```
  /bin /sbin /lib /lib64
  ```

  ya que son sólo enlaces simbólicos a sus contrapartes bajo /usr.

Lab 5.2: Un recorrido del sistema de archivos /proc

Lo que usted verá exactamente en este ejercicio dependerá de la versión del kernel que esté usando, por lo cual la salida que obtenga en los comandos podría diferir un poco.

1. Como root, haga **cd** en /proc y liste lis archivos. Esto debería desplegar un número de archivos y directorios:

   ```
   $ cd /proc
   $ ls -F
   1/     17/    2180/   2541/  34/    508/   636/   773/          locks
   10/    1706/  22/     259/   3469/  510/   644/   794/          meminfo
   1009/  1707/  2203/   26/    35/    512/   645/   8/            misc
   1014/  1775/  2231/   2626/  36/    513/   66/    825/          modules
   1015/  1779/  2233/   263/   37/    515/   67/    826/          mounts@
   1019/  18/    2234/   2635/  374/   517/   676/   879/          mtrr
   1023/  1846/  2241/   264/   3792/  519/   68/    9/            net@
   11/    1898/  23/     266/   3857/  521/   681/   acpi/         pagetypeinfo
   1144/  19/    2319/   27/    3858/  5217/  6824/  asound/       partitions
   12/    1901/  2323/   271/   3865/  537/   6909/  buddyinfo     sched_debug
   1242/  1905/  2337/   278/   3866/  538/   6979/  bus/          schedstat
   1265/  1908/  2338/   279/   395/   555/   7/     cgroups       scsi/
   1295/  1923/  2363/   28/    397/   556/   7053/  cmdline       self@
   1296/  1931/  238/    2897/  3990/  5564/  7091/  config.gz     slabinfo
   1297/  1935/  239/    29/    409/   5571/  7123/  consoles      softirqs
   1298/  1941/  23957/  2928/  42/    5768/  7188/  cpuinfo       stat
   1299/  2/     24/     2945/  43/    583/   7222/  crypto        swaps
   13/    2015/  240/    2946/  4529/  584/   723/   devices       sys/
   1306/  2018/  241/    2947/  453/   5858/  7236/  diskstats     sysrq-trigger
   14/    2041/  242/    2950/  472/   5872/  725/   dma           sysvipc/
   1405/  2046/  243/    2951/  473/   5878/  726/   driver/       thread-self@
   1449/  2049/  244/    2952/  476/   593/   728/   execdomains   timer_list
   1457/  2055/  245/    2953/  477/   594/   7312/  fb            timer_stats
   1470/  2059/  246/    2954/  479/   596/   7313/  filesystems   tty/
   1490/  2062/  24697/  2955/  480/   597/   7321/  fs/           uptime
   1495/  2070/  247/    2956/  481/   6130/  738/   interrupts    version
   1508/  2082/  248/    2957/  482/   6131/  740/   iomem         vmallocinfo
   1550/  2091/  249/    2965/  485/   616/   745/   ioports       vmnet/
   1560/  2096/  24962/  2966/  486/   617/   746/   irq/          vmstat
   ```



```
1561/  2099/  2503/  3/     491/  6181/  748/  kallsyms       zoneinfo
1587/  21/    2506/  30/    497/  624/   749/  kcore
16/    2111/  2513/  3072/  498/  625/   752/  keys
1626/  2117/  2514/  3079/  499/  627/   758/  key-users
1664/  2120/  2516/  3090/  5/    628/   759/  kmsg
1669/  2125/  2517/  31/    501/  631/   762/  kpagecount
1675/  2137/  2520/  32/    502/  632/   763/  kpageflags
1685/  2173/  2521/  3256/  504/  634/   765/  latency_stats
1698/  2175/  2523/  33/    507/  635/   767/  loadavg
```

Tenga en cuenta que muchos de los nombres de los directorios son números; cada uno corresponde a un proceso en ejecución y sus nombres son el **process ID**. Un subdirectorio importante que veremos más adelante es /proc/sys, bajo el cual es posible ver o modificar muchos parámetros del sistema.

2. Vea el contenido de los siguientes archivos:

   - /proc/cpuinfo:
   - /proc/meminfo:
   - /proc/mounts:
   - /proc/swaps:
   - /proc/version:
   - /proc/partitions:
   - /proc/interrupts:

   Los nombres de cada uno dan una buena idea acerca de la información que contienen.

   Tenga en cuenta que esta información no se actualiza de forma constante, sino que es obtenida sólo cuando uno quiere visualizarla.

3. Eche un vistazo a cualquier directorio de proceso. Si no es un proceso del cual usted es dueño, el acceso a la información podría ser limitada, a menos que use **sudo**):

```
$ ls -F 5564
auxv             cwd@     latency     net/           projid_map  statm
cgroup           environ  limits      ns/            root@       status
clear_refs       exe@     maps        oom_adj        sched       syscall
cmdline          fd/      mem         oom_score      schedstat   task/
comm             fdinfo/  mountinfo   oom_score_adj  smaps       uid_map
coredump_filter  gid_map  mounts      pagemap        stack       wchan
cpuset           io       mountstats  personality    stat
```

   Eche un vistazo a algunos de los campos aquí, tales como `cmdline`, `cwd`, `environ`, `mem`, y `status`

Capitulo 6

Servicios del Kernel y Configuración

Lab 6.1: Ajustes del sistema con sysctl

1. Verifique si puede hacer **ping** a su sistema. Tenga en cuenta que en **RHEL 7** es necesario ser root para hacer **ping** en la mayoría de redes externas.
2. Verifique el valor actual de `net.ipv4.icmp_echo_ignore_all`, el cual se usa para habilitar y deshabilitar que su sistema responda a **ping**. El valor `0` permite a su sistema responder a pings.
3. Configure el valor en `1` usando la herramienta de línea de comandos **sysctl** y luego verifique si el sistema dejó de responder a pings.
4. Configure el valor de vuelta a `0` y confirme si se restauró el comportamiento original.
5. Ahora modifique el valor a través de la edición de /etc/sysctl.conf y fuerce al sistema a activar esa configuración en el archivo sin reiniciar el sistema.
6. Verifique que la modificación esté funcionando correctamente.

Una vez que haya terminado, puede reiniciar su sistema para asegurarse que todo está de vuelta en el punto original.

Solution 6.1

Usted puede usar ya sea `localhost` o `127.0.0.1` (loopback) o la dirección IP actual del sistema como objeto de prueba para el **ping** de a continuación.

1. `$ ping localhost`
2. `$ sysctl net.ipv4.icmp_echo_ignore_all`

3.
```
$ sudo sysctl net.ipv4.icmp_echo_ignore_all=1
$ ping localhost
```

4.
```
$ sudo sysctl net.ipv4.icmp_echo_ignore_all=0
$ ping localhost
```

5. Agregue la línea siguiente a /etc/sysctl.conf:

   ```
   net.ipv4.icmp_echo_ignore_all=1
   ```

   y luego haga:

   ```
   $ sysctl -p
   ```

6.
```
$ sysctl net.ipv4.icmp_echo_ignore_all
$ ping localhost
```

Ya que los cambios en el archivo /etc/sysctl.conf son persistentes, sería buena idea restaurar los valores originales.

Lab 6.2: Modificar el ID de proceso máximo - maximum process ID

El comportamiento normal de un sistema **Linux** es que los IDs de proceso comiencen en `PID=1` para el proceso **init**, el primer proceso en el sistema, y luego en forma secuencial a medida en que procesos nuevos van siendo creados y terminados (de forma constante).

However, when the PID reaches the value shorted in /proc/sys/kernel/pid_max, which is conventionally 32768 (32K), they will wrap around to lower numbers. If nothing else, this means you can't have more than 32K processes on the system since there are only that many slots for PIDs.

El comportamiento normal de un sistema **Linux** es que los IDs de proceso comiencen en `PID=1` para el proceso **init**, el primer proceso en el sistema, y luego en forma secuencial a medida en que procesos nuevos van siendo creados y terminados (de forma constante).

Sin embargo, cuando el PID alcanza el valor especificado en /proc/sys/kernel/pid_max, el cual es generalmente 32768 (32K), se intentará utilizar números bajos. Si no hay más números disponibles no hay nada que hacer, ya que no es posible tener más de 32K procesos en el sistema (ese es el número máximo par los PIDs).

1. Obtenga el valor actual del PID máximo.
2. Averigue cuáles son los PIDs que está siendo creados actualmente.
3. Configure `pid_max` a un valor menor.
4. Inicie un proceso y vea qué valor de PID se le asigna.

Solution 6.2

A continuación vamos a usar dos métodos; uno usando **sysctl**, el otro escribiendo valores directamente con echo a /proc/sys/kernel/pid_max. Note que el método **echo** require ser root, **sudo** no funcionará. Dejaremos que usted descubra porqué, en caso que no lo sepa todavía.

1.
```
$ sysctl kernel.pid_max
$ cat /proc/sys/kernel/pid_max
```

2. Type:

   ```
   $ cat &
   ```



```
[1] 29222
$ kill -9 29222
```

3. ```
   $ sudo sysctl kernel.pid_max=24000
   $ echo 24000 >  /proc/sys/kernel/pid_max # Esto debe ser realizado como root
   $ cat /proc/sys/kernel/pid_max
   ```

4. ```
   $ cat &
   [2] 311
   $ kill -9 311
   ```

Tenga en cuenta que cuando se comienza de nuevo, el kernel comienza en PID=300, no en un número menor. Es posible que se de cuenta que asignar PIDs a los procesos nuevos no es algo trivial; debido a que el sistema puede haber comenzado de nuevo a asignar los números correspondientes, el kernel siempre tiene que verificar que al crear esos nuevos PIDs no están en uso. . El kernel **Linux** tiene una manera muy eficiente de hacer esto, la cual no depende del número de procesos corriendo en el sistema.

Capitulo 7

Módulos del Kernel

Lab 7.1: Módulos del kernel

1. Liste todos los módulos del kernel que están cargados actualmente en su sistema.
2. Cargue un módulo que no esté en uso en su sistema. Si está ejecutando un kernel que viene con la distribución es fácil encontrar los módulos; simplemente tiene que buscar en el directorio `/lib/modules/<kernel-version>/kernel/drivers/net` y elegir uno (los kernels que vienen con las distribuciones incluyen controladores para cada dispositivo, sistema de archivos, protocolo de red, etc., todo lo que un sistema puede necesitar). Sin embargo, si está corriendo un kernel personalizado es posible que no tenga muchos módulos disponibles para cargar.
3. Liste los módulos del kernel nuevamente y vea si el módulo fue realmente cargado.
4. Remueva el módulo que cargó anteriormente.
5. Liste los módulos del kernel nuevamente y compruebe si el módulo fue descargado correctamente.

Solution 7.1

1. `$ lsmod`

   En la sección que viene a continuación, substituya cualquier nombre de módulo que usó en vez de `3c59x`. Cualquiera de estos dos métodos va a funcionar, pero el segundo es más fácil.
2. ```
   $ sudo insmod /lib/modules/$(uname -r)/kernel/drivers/net/3c59.x
   $ sudo /sbin/modprobe 3c59x
   ```
3. `$ lsmod | grep 3c59x`
4. De nuevo, cualquiera de los siguientes métodos va a funcionar.
   ```

```
$ sudo rmmod 3c59x
$ sudo modprobe -r 3c59x
```

5. `$ lsmod | grep 3c59x`

Capitulo 8

Dispositivos y udev

Lab 8.1: `udev`

1. Cree e implemente una regla en su sistema, la cual debe crear un enlace simbólico llamado `myusb` cuando un dispositivo **USB** es conectado.
2. Conecte un dispositivo **USB** a su sistema. Puede ser un pendrive, mouse, webcam, etc.

   Nota: si está ejecutando una máquina virtual bajo un hipervisor, tendrá que asegurarse que el dispositivo **USB** es visto por el guest, lo cual generalmente significa un click en la aplicación correspondiente (al mismo tiempo el dispositivo se desconecta desde el host).
3. Obtenga un listado del directorio /dev y compruebe si su enlace simbólico fue creado.
4. Remueva el dispositivo **USB**. En caso que sea un disco, debería ejecutar **umount** primero, por seguridad.
5. Verifique si el enlace simbólico todavía existe en /dev.

Solution 8.1

1. Cree un archivo llamado /etc/udev/rules.d/75-myusb.rules e incluya una línea de contenido:

   ```
   $ cat /etc/udev/rules.d/75-myusb.rules

   SUBSYSTEM=="usb", SYMLINK+="myusb"
   ```

   No use el valor obsoleto `BUS` en vez de `SUBSYSTEM`, ya que versiones recientes de **udev** lo han removido.

   Note que el nombre de este archivo no importa. Si hubiera un componente `ACTION` en la regla, el sistema debería ejecutarlo; eche un vistazo a otras reglas para que le sirvan de ejemplo.
2. Conecte un dispositivo.

3. `$ ls -lF /dev | grep myusb`

4. Si el dispositivo ha sido montado:

   `$ umount /media/whatever`

   donde /media/whatever es el punto de montaje. Remueva el dispositivo de forma segura.

5. `$ ls -lF /dev | grep myusb`

Capitulo 9

Particionamiento y formateo de discos

Lab 9.1: Usar un archivo como imagen de partición de disco

Para los propósitos de los ejercicios en este curso usted necesitará espacio en disco sin particionar. No es necesario que sea de gran tamaño, uno o dos GB es suficiente.

Si está usando su máquina nativa y no tiene espacio disponible, va a tener que encoger una partición y el sistema de archivos en ella (esto último se debe realizar primero), y luego disponer de ella usando **gparted** y/o los pasos que tratamos en la sección de manejo de particiones.

También puede usar el mecanismo **loop device** con o sin el programa **parted** program, como lo haremos en los primeros dos ejercicios en esta sección.

Si tiene espacio físico sin particionar usted no **necesita** realizar el procedimiento que se describe a continuación, sin embargo, aún es un ejercicio de aprendizaje muy útil.

Vamos a crear un archivo que será usado como contenedor de una imagen de partición del disco duro, y para todos los propósitos puede utilizarse como una partición real. En el siguiente ejercicio mostraremos cómo poner más de una partición en ella y cómo usarlo como si fuera un disco completo.

1. Crear un archivo lleno de ceros de 1 GB de tamaño:

   ```
   $ dd if=/dev/zero of=imagefile bs=1M count=1024
   ```

   Puede crear un archivo más pequeño si lo desea o si no tiene suficiente espacio en la partición en la cual está creándolo.

2. Cree un sistema de archivos en ella:

   ```
   $ mkfs.ext4 imagefile
   mke2fs 1.42.9 (28-Dec-2013)
   imagefile is not a block special device.
   Proceed anyway? (y,n) y
   Discarding device blocks: done
   .....
   ```

Obviamente puede darle formato con un sistema de archivos diferente, haciendo **mkfs.ext3**, **mkfs.vfat**, **mkfs.xfs**, etc.

3. Móntela en algún lugar:

```
$ mkdir mntpoint
$ sudo mount -o loop imagefile mntpoint
```

Ahora puede usar la partición a su antojo, poniendo archivos, etc.

4. Una vez que haya terminado, desmóntela con:

```
$ sudo umount mntpoint
```

Un método alternativo es usar la opción `loop` para montarla:

```
$ sudo losetup /dev/loop2 imagefile
$ sudo mount /dev/loop2 mntpoint
 ....
$ sudo umount mntpoint
$ sudo losetup -d /dev/loop2
```

Revisaremos **losetup** en un ejercicio más adelante. Puede usar /dev/loop[0-7] pero tenga cuidado de que no estén en uso actualmente.

Si bien es cierto que usar un dispositivo de tipo loop en vez de una partición real puede ser útil, debe saber que es es bastante inútil para propósitos de realizar cualquier tipo de benchmarking. Esto se debe a que está poniendo una capa de sistema de archivos sobre otra, lo cual tendrá un efecto negativo en el rendimiento. Por lo anterior, cualquier juicio al respecto va a estar basado en cómo se comporta un sistema de archivos arriba de otro.

Lab 9.2: Particionar un archivo de imagen de disco

El siguiente paso consiste en dividir el archivo de contenedor en múltiples particiones, cada uno de los cuales puede ser usado para almacenar un sistema de archivos o un área de intercambio.

Usted puede reutilizar el archivo de imagen creado en el ejercicio previo o puede crear uno nuevo.

1. Ejecute **fdisk** en su archivo de imagen:

```
$ sudo fdisk -C 130 imagefile
Device does not contain a recognized partition table
Building a new DOS disklabel with disk identifier 0x6280ced3.
Welcome to fdisk (util-linux 2.23.2).

Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Command (m for help):
```

2. Type `m` to get a list of commands:

```
Command (m for help): m

Command action
   a   toggle a bootable flag
   b   edit bsd disklabel
   c   toggle the dos compatibility flag
   d   delete a partition
   g   create a new empty GPT partition table
   G   create an IRIX (SGI) partition table
```



```
   l   list known partition types
   m   print this menu
   n   add a new partition
   o   create a new empty DOS partition table
   p   print the partition table
   q   quit without saving changes
   s   create a new empty Sun disklabel
   t   change a partition's system id
   u   change display/entry units
   v   verify the partition table
   w   write table to disk and exit
   x   extra functionality (experts only)

Command (m for help):
```

3. The `-C 130` which sets the number of phony cylinders in the drive is only necessary in old versions of **fdisk**, which unfortunately you will find on **RHEL 6**. However, it will do no harm on other distributions.

   Create a new primary partition and make it 256 MB (or whatever size you would like:

```
Command (m for help): n
Partition type:
   p   primary (0 primary, 0 extended, 4 free)
   e   extended
Select (default p): p
Partition number (1-4, default 1): 1
First sector (2048-2097151, default 2048):
Using default value 2048
Last sector, +sectors or +size{K,M,G} (2048-2097151, default 2097151): +256M
Partition 1 of type Linux and of size 256 MiB is set
```

4. Add a second primary partition also of 256 MB in size:

```
Command (m for help): n
Partition type:
   p   primary (1 primary, 0 extended, 3 free)
   e   extended
Select (default p): p
Partition number (2-4, default 2): 2
First sector (526336-2097151, default 526336):
Using default value 526336
Last sector, +sectors or +size{K,M,G} (526336-2097151, default 2097151): +256M
Partition 2 of type Linux and of size 256 MiB is set

Command (m for help): p

Disk imagefile: 1073 MB, 1073741824 bytes, 2097152 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk label type: dos
Disk identifier: 0x6280ced3

         Device Boot      Start         End      Blocks   Id  System
imagefile1            2048      526335      262144   83  Linux
imagefile2          526336     1050623      262144   83  Linux
```

5. Write the partition table to disk and exit:

```
Command (m for help): w
The partition table has been altered!

Syncing disks.
```


Si bien esta ha sido una buena práctica, aún no hemos visto una forma de utilizar las particiones que creamos recién. En el próximo ejercicio vamos a ver una forma que nos permitirá hacerlo.

Lab 9.3: Uso de losetup y parted

Vamos a experimentar con:

- Dispositivos Loop y **losetup**
- **parted** para trabajar con particiones desde la línea de comandos de forma no interactiva.

Es de nuestro interés que usted lea las **páginas man** de **losetup** y **parted** antes de realizar los procedimientos que vienen a continuación.

Una vez más, usted puede reutilizar el archivo de imagen, o mejor aún, recrearlo para comenzar con un archivo nuevo.

1. Asocie el archivo de imagen con un dispositivo **loop**:

   ```
   $ losetup -f
   /dev/loop1
   $ sudo losetup /dev/loop1 imagefile
   ```

   Donde el primer comando encuentra el primer dispositivo loop **libre**. La razón para hacer esto es que su sistema podría estar usando uno o más dispositivos loop. Lo siguiente es ejecutado como ejemplo en un sistema de pruebas, antes de crear el dispositivo loop:

   ```
   $ losetup -a
   /dev/loop0: []: (/usr/src/KERNELS.sqfs)
   ```

   un sistema de archivos comprimido de sólo lectura **squashfs** está montado y usando /dev/loop0. Nota: la salida del comando anterior puede variar en función de la distribución. Si ignoráramos lo anterior y ejecutamos **losetup** en /dev/loop0 probablemente corromperíamos el archivo.

2. Crear una etiqueta de partición de disco en el dispositivo loop (archivo de imagen):

   ```
   $ sudo parted -s /dev/loop1 mklabel msdos
   ```

3. Crear tres particiones primarias en el dispositivo loop:

   ```
   $ sudo parted -s /dev/loop1 unit MB mkpart primary ext4 0 256
   $ sudo parted -s /dev/loop1 unit MB mkpart primary ext4 256 512
   $ sudo parted -s /dev/loop1 unit MB mkpart primary ext4 512 1024
   ```

4. Verificar la tabla de particiones:

   ```
   $ fdisk -l /dev/loop1
   Disk /dev/loop1: 1073 MB, 1073741824 bytes, 2097152 sectors
   Units = sectors of 1 * 512 = 512 bytes
   Sector size (logical/physical): 512 bytes / 512 bytes
   I/O size (minimum/optimal): 512 bytes / 512 bytes
   Disk label type: dos
   Disk identifier: 0x00050c11

         Device Boot      Start         End      Blocks   Id  System
   /dev/loop1p1               1      500000      250000   83  Linux
   /dev/loop1p2          500001     1000000      250000   83  Linux
   /dev/loop1p3         1000001     2000000      500000   83  Linux
   ```

5. Lo que sucede a continuación dependerá en la distribución en la cual está trabajando. Por ejemplo, en **RHEL 7** y **Ubuntu 14.04** notará que los nodos de dispositivo nuevos han sido creados:


```
$ ls -l /dev/loop1*
brw-rw---- 1 root disk   7, 1 Oct  7 14:54 /dev/loop1
brw-rw---- 1 root disk 259, 0 Oct  7 14:54 /dev/loop1p1
brw-rw---- 1 root disk 259, 3 Oct  7 14:54 /dev/loop1p2
brw-rw---- 1 root disk 259, 4 Oct  7 14:54 /dev/loop1p3
```

los que utilizaremos a continuación. Sin embargo, en **RHEL 6** los nodos no aparecerán. En cambio, es necesario hacer lo siguiente:

```
$ sudo kpartx -lv /dev/loop1
$ sudo kpartx -av /dev/loop1
$ ls -l /dev/mapper/loop1*
lrwxrwxrwx 1 root root 7 Oct  9 07:12 /dev/mapper/loop1p1 -> ../dm-8
lrwxrwxrwx 1 root root 7 Oct  9 07:12 /dev/mapper/loop1p2 -> ../dm-9
lrwxrwxrwx 1 root root 8 Oct  9 07:12 /dev/mapper/loop1p3 -> ../dm-10
```

para asociar los nodos de dispositivo con las particiones. En lo que sigue puede reemplazar /dev/loop1p[1-3] con los nombres reales bajo /dev/mapper, o incluso más fácil, puede hacer esto:

```
$ sudo ln -s /dev/mapper/loop1p1 /dev/loop1p1
$ sudo ln -s /dev/mapper/loop1p2 /dev/loop1p2
$ sudo ln -s /dev/mapper/loop1p3 /dev/loop1p3
```

6. Darle formato a las particiones:

```
$ sudo mkfs.ext3 /dev/loop1p1
$ sudo mkfs.ext4 /dev/loop1p2
$ sudo mkfs.vfat /dev/loop1p3
```

7. Monte los tres sistemas de archivos y muestre que están disponibles:

```
$ mkdir mnt1 mnt2 mnt3

$ sudo mount /dev/loop1p1 mnt1
$ sudo mount /dev/loop1p2 mnt2
$ sudo mount /dev/loop1p3 mnt3

$ df -Th
Filesystem               Type      Size  Used Avail Use% Mounted on
/dev/sda1                ext4       29G  8.5G   19G  32% /
....
/dev/loop1p1             ext3      233M  2.1M  219M   1% mnt1
/dev/loop1p2             ext4      233M  2.1M  215M   1% mnt2
/dev/loop1p3             vfat      489M     0  489M   0% mnt3
```

8. Una vez que ha terminado de utilizar los sistemas de archivos puede deshacer lo realizado con:

```
$ sudo umount mnt1 mnt2 mnt3
$ rmdir mnt1 mnt2 mnt3
$ sudo losetup -d /dev/loop0
```

Lab 9.4: Particionado de un disco duro real

Si usted tiene un disco duro real con espacio disponible y que no está particionado, experimente con **fdisk** para crear particiones nuevas, ya sean primarias o lógicas con una partición extendida. Escriba la tabla de particiones nueva al disco y luego formatee y monte las particiones recién creadas.

Capitulo 10

Cifrado de discos

Lab 10.1: Cifrado de discos

En este ejercicio usted cifrará una partición en el disco para proveer de seguridad en caso de que su disco duro o portátil es robado. Revisar la documentación de **cryptsetup** antes de comenzar es una buena idea (`man cryptsetup` y `cryptsetup --help`).

1. Cree una partición nueva para el dispositivo de bloque cifrado con **fdisk**. Asegúrese que el kernel está al tanto de la nueva tabla de partición. Un reinicio lo logrará, pero hay otros métodos también.
2. Formatee la partición con **cryptsetup** usando **LUKS** para la capa de cifrado.
3. Cree la clave para abrir el dispositivo de bloque cifrado.
4. Agregue una entrada a /etc/crypttab para que el sistema pregunte la clave en el reinicio.
5. Formatee el sistema de archivos con **ext4**.
6. Cree un punto de montaje para el sistema de archivos nuevo, por ejemplo /secret.
7. Agregue una entrada a /etc/fstab para que el sistema sea montado en el arranque.
8. Intente montar el sistema cifrado.
9. Reinicie y valide la configuración completa.

Solution 10.1

1. `$ sudo fdisk /dev/sda`

   Cree una partición nueva (en el ejemplo trabajaremos con /dev/sda4) y luego ejecute:

```
$ sudo partprobe -s
```

para que el sistema relea la tabla de partición modificada, o reinicie (lo cual es lejos lo más seguro).

Nota: Si no puede usar una partición real, use el método descrito en el capítulo anterior para trabajar con un dispositivo loop o un archivo de imagen.

2. `$ sudo cryptsetup luksFormat /dev/sda4`

3. `$ sudo cryptsetup luksOpen /dev/sda4 secret-disk`

4. Agregue lo siguiente a /etc/crypttab:

```
secret-disk    /dev/sda4
```

5. `$ sudo mkfs -t ext4 /dev/mapper/secret-disk`

6. `$ sudo mkdir -p /secret`

7. Agregue lo siguiente a /etc/fstab:

```
/dev/mapper/secret-disk    /secret    ext4    defaults    1 2
```

8. Monte el sistema de archivos:

```
$ sudo mount /secret
```

o monte todos los sistemas de archivos mencionados en /etc/fstab:

```
$ sudo mount -a
```

9. Reinicie.

Lab 10.2: Área de intercambio cifrada

En este ejercicio vamos a cifrar la **partición de intercambio**. La información escrita al dispositivo de intercambio puede contener información sensible. Debido a que el área de intercambio está sobre una partición, es importante considerar las implicancias de seguridad que tiene una partición de intercambio sin cifrar.

El proceso de cifrado es similar al del ejercicio previo, con la excepción de que no crearemos un sistema de archivos en este dispositivo de bloques cifrado.

En este caso vamos a usar el área de intercambio existente; primero la desactivaremos y luego formateada para el uso como área de intercambio cifrada. Podría ser un poco más seguro usar una partición nueva, o también usar la partición que creó en el ejercicio previo. Al final explicaremos qué hacer en caso de que tenga problemas para restablecer el sistema al punto original.

Vamos a discutir la administración del área de intercambio en un capítulo posterior, pero de todas formas mostraremos algunos comandos para trabajar con esta componente.

Una vez que haya terminado, puede volver a la partición original sin cifrar ejecutando el comando **mkswap** en el dispositivo.

1. Determine cuál es la partición que está usando actualmente para el área de intercambio y desactívela:

```
$ cat /proc/swaps
Filename                                Type            Size    Used    Priority
/dev/sda11                              partition       4193776 0       -1
$ sudo swapoff /dev/sda11
```

2. Realice los mismos pasos del ejercicio anterior para configurar el cifrado:


```
$ sudo cryptsetup luksFormat /dev/sda11  # may use --ciper aes option
$ sudo cryptsetup luksOpen   /dev/sda11  swapcrypt
```

3. Formatee el dispositivo cifrado para usarlo como área de intercambio:

```
$ sudo mkswap /dev/mapper/swapcrypt
```

4. Ahora active la partición y verifique si está funcionando:

```
$ sudo swapon /dev/mapper/swapcrypt
$ cat /proc/swaps
```

5. Para asegurarse que la partición de intercambio cifrada se active en el arranque, es necesario hacer dos cosas:

   (a) Agregue una línea a /etc/crypttab para que el sistema pregunte por la clave en el reinicio:

   ```
   swapcrypt  /dev/sda11   /dev/urandom  swap,cipher=aes-cbc-essiv:sha256,size=256
   ```

   (Note que /dev/urandom es preferido sobre /dev/random ya que podría estar relacionado a **entropy shortages, o problemas de rendimiento** como se lee en la página **man** de `crypttab`.)

   No es necesario que siga las opciones detalladas que siguen a continuación, pero las damos como ejemplo de lo que usted puede hacer.

   (b) Agregue una entrada al archivo `/etc/fstab` para que el dispositivo de aŕea de intercambio sea activado en el inicio.

   ```
   /dev/mapper/swapcrypt  none    swap    defaults    0 0
   ```

6. Reinicie y valide la configuración completa.

Para restaurar el sistema al punto original:

```
$ sudo swapoff /dev/mapper/swapcrypt
$ sudo cyyptsetup luksClose swapcrypt
$ sudo mkswap /dev/sda11
$ sudo swapon -a
```

Si el comando **swapon** falla, probablemente se debe a que el archivo /etc/fstab no describe adecuadamente la partición de intercambio. No hay ningún problema si la partición está descrita aquí por el dispositivo actual (/dev/sda11). Puede resolverlo al cambiar la línea a:

```
/dev/sda11  swap   swap  defaults 0 0
```

otra alternativa es asignarle una etiqueta al instante del formateo, como se muestra aquí:

```
$ sudo mkswap -L SWAP /dev/sda11
```

y luego agréguelo al archivo:

```
LABEL=SWAP  swap   swap  defaults 0 0
```


Capitulo 11

Sistemas de archivos Linux y el VFS

Lab 11.1: El sistema de archivos especial tmpfs

tmpfs es uno de muchos sistemas de archivos especiales usados en **Linux**. Algunos de estos no son usados realmente como sistemas de archivos, pero toman ventaja de la capa de abstracción que poseen. Sin embargo, **tmpfs** es un sistema de archivos real sobre el cual las aplicaciones pueden realizar operaciones de E/S.

Esencialmente, **tmpfs** opera como **ramdisk**; reside completamente en memoria. Pero tiene algunas características interesantes que las implementaciones convencionales y antiguas de ramdisk no tenían:

1. El sistema de archivos ajusta su tamaño (The filesystem adjusts its size (y por lo tanto la memoria que se usa) dinámicamente; parte en cero y se expande tanto como sea necesario hasta el tamaño máximo de la partición en la que está montada.
2. Si la RAM se agota, **tmpfs** puede utilizar espacio del área de intercambio. De todas formas no es posible poner más en el sistema de archivos de la capacidad máxima que soporta.
3. **tmpfs** no requiere tener un sistema de archivos normal, tales como **ext3** o **vfat**; posee sus métodos propios para lidiar con archivos y operaciones de E/S, los cuales están conscientes de que es sólo espacio en memoria (y que en realidad no es un dispositivo de bloque), y como tal está optimizado para velocidad.

   Por lo tanto no hay necesidad de preformatear el sistema de arcivos con el comando `mkfs` ; simplemente hay que montarlo y usarlo.

Monte una instancia nueva de **tmpfs** en cualquier lugar en su estructura de directorios, con un comando como el siguiente:

```
$ sudo mkdir /mnt/tmpfs
$ sudo mount -t tmpfs none /mnt/tmpfs
```

Vea cuánto espacio se le ha asignado al sistema de archivos y cuánto está usando:

```
$ df -h /mnt/tmpfs
```

Debería notar que se le ha asignado un valor por defecto de la mitad de la RAM del sistema; sin embargo, el uso es cero, y sólo comenzará a crecer en espacio utilizado en la medida de que se ponen archivos en /mnt/tmpfs.

Es posible cambiar el tamaño asignado como una opción a la hora de montarlo:

```
$ sudo mount -t tmpfs -o size=1G none /mnt/tmpfs
```

Podría intentar llenarlo hasta que se alcance la capacidad máxima y luego vea qué sucede. No olvide desmontarlo una vez que haya terminado las pruebas, con el comando:

```
$ sudo umount /mnt/tmpfs
```

Prácticamente todas las distributiones modernas de **Linux** montan una instancia de **tmpfs** en /dev/shm:

```
$ df -h /dev/shm

Filesystem     Type   Size  Used Avail Use% Mounted on
tmpfs          tmpfs  3.9G   24M  3.9G   1% /dev/shm
```

Muchas aplicaciones hacen esto en casos como cuando se usa memoria compartida **POSIX** como un mecanismo de comunicación interprocesos. Cualquier usuario puede crear, leer y escribir archivos en /dev/shm, por lo que es un buen lugar para crear archivos temporales en memoria.

Cree algunos archivos en /dev/shm y observe con **df** cómo el sistema de archivos se va llenando.

Adicionalmente, muchas distribuciones montan instancias múltiples de **tmpfs**; por ejemplo, se observa lo siguiente en un sistema **RHEL 7**:

```
$ df -h | grep tmpfs

devtmpfs                                        devtmpfs  3.9G     0  3.9G   0% /dev
tmpfs                                           tmpfs     3.9G   24M  3.9G   1% /dev/shm
tmpfs                                           tmpfs     3.9G  9.2M  3.9G   1% /run
tmpfs                                           tmpfs     3.9G     0  3.9G   0% /sys/fs/cgroup
/tmp/vmware-coop/564d9ea7-8e8e-29c0-2682-e5d3de3a51d8 tmpfs     3.3G     0  3.3G   0% /tmp/vmware-coop/
                                                                                  564d9ea7-8e8e-29c0-2682-e5d3de3a51d8
/tmp/vmware-coop/564d7668-ec55-ee45-f33e-c8e97e956190 tmpfs     2.3G  2.0G  256M  89% /tmp/vmware-coop/
                                                                                  564d7668-ec55-ee45-f33e-c8e97e956190
none                                            tmpfs     1.0G  1.0G     0 100% /tmp/ohno
```

Note que el comando anterior fue ejecutado en un sistema con 8GB de RAM, por lo cual usted probablemente no tendrá todos esos sistemas de archivos **tmpfs** usando los 4 GB con los cuales han sido asignados.

Algunas distribuciones como **Fedora** podrían montar por defecto /tmp como un sistema **tmpfs**; en estos casos uno podría evitar poner archivos grandes en /tmp para que el sistema no se quede sin memoria. Otra posibilidad es deshabilitar ese comportamiento, como se mencionó anteriormente al describir /tmp.

Capitulo 12

Características de los sistemas de archivos

Lab 12.1: Trabajo con atributos de archivo

1. Con su cuenta de usuario normal use **touch** para crear un archivo vacío llamado /tmp/appendit.
2. Use **cat** para agregar el contenido de /etc/hosts a /tmp/appendit.
3. Compare los contenidos de /tmp/appendit con /etc/hosts; no debería existir ninguna diferencia.
4. Intente agregar el atributo agregar solamente a /tmp/appendit usando **chattr**. Debería obtener un error. ¿A qué se debe esto?
5. Como root, intente agregar el atributo agregar solamente; esta vez debería funcionar. Eche un vistazo a los atributos extendidos del archivo usando **lsattr**.
6. Como usuario normal intente copiar con **cat** el contenido de /etc/passwd a /tmp/appendit. Debería obtener un error, ¿a qué se debe?
7. Intente lo mismo esta vez como root. También debería obtener un error. ¿Por qué?
8. Como usuario normal, use el operador de redirección para agregar (>>) e intente agregar el contenido del archivo /etc/passwd a /tmp/appendit. Esta vez debería funcionar. Examine el archivo para confirmar.
9. Como root, configure el atributo de inmutabilidad sobre /tmp/appendit, y liste los atributos extendidos nuevamente.
10. Intente agregar contenido a /tmp/appendit. Intente renombrar el archivo, crear un enlace duro al mismo e incluso borrarlo como usuario normal y como root.
11. Es posible eliminar el archivo si se remueven los atributos extendidos. Hágalo por favor.

Solution 12.1

1.
```
$ cd /tmp
$ touch appendit
$ ls -l appendit
-rw-rw-r-- 1 coop coop 0 Oct 23 19:04 appendit
```

2.
```
$ cat /etc/hosts > appendit
```

3.
```
$ diff /etc/hosts appendit
```

4.
```
$ chattr +a appendit
chattr: Operation not permitted while setting flags on appendit
```

5.
```
$ sudo chattr +a appendit
$ lsattr appendit
-----a-------e-- appendit
```

6.
```
$ cat /etc/passwd > appendit
bash: appendit: Operation not permitted
```

7.
```
$ sudo su
$ cat /etc/passwd > appendit
bash: appendit: Operation not permitted
$ exit
```

8.
```
$ cat /etc/passwd >> /tmp/appendit
$ cat appendit
```

9.
```
$ sudo chattr +i appendit
$ lsattr appendit
----ia-------e- appendit
```

10.
```
$ echo hello >> appendit
-bash: appendit: Permission denied
$ mv appendit appendit.rename
mv: cannot move ‘appendit’ to ‘appendit.rename’: Operation not permitted
$ ln appendit appendit.hardlink
ln: creating hard link ‘appendit.hardlink’ => ‘appendit’: Operation not permitted
$ rm -f appendit
rm: cannot remove ‘appendit’: Operation not permitted

$ sudo su
$ echo hello >> appendit
-bash: appendit: Permission denied
$ mv appendit appendit.rename
mv: cannot move ‘appendit’ to ‘appendit.rename’: Operation not permitted
$ ln appendit appendit.hardlink
ln: creating hard link ‘appendit.hardlink’ => ‘appendit’: Operation not permitted
$ rm -f appendit
rm: cannot remove ‘appendit’: Operation not permitted
$ exit
```

11.
```
$ sudo su
$ lsattr appendit
----ia-------e- appendit
$ chattr -ia /appendit
$ rm appendit
rm: remove regular file ‘appendit’? y
$ ls appendit
ls: cannot access appendit: No such file or directory
```


Lab 12.2: Opciones de montaje

En este ejercicio tendrá que crear una partición nueva o usar un archivo loopback. La solución va a variar un poco y proveeremos los detalles de ambos métodos.

1. Use **fdisk** para crear una partición nueva de 250 MB en el sistema, probablemente en /dev/sda. O cree un archivo lleno de ceros para usar como archivo loopback para simular una partición nueva.
2. Use **mkfs** para formatear un sistema de archivos nuevo en la partición o archivo loopback que creó recién. Hágalo tres veces, cambiando el tamaño de bloque cada vez. Fíjese en las ubicaciones de los súperbloques, el número de los grupos de bloques y cualquier otra información pertinente, para cada caso.
3. Cree un subdirectorio nuevo (digamos /mnt/tempdir) y monte el sistema de archivos nuevo en esa ubicación. Confirme que haya sido montado.
4. Desmonte el sistema de archivos nuevo y luego móntelo nuevamente con permisos de sólo lectura.
5. Intente crear un archivo en el directorio recién montado. Debería obtener un error, ¿por qué?
6. Desmonte el sistema de archivos nuevamente.
7. Agregue una línea a su archivo /etc/fstab para que ese sistema de archivos sea montado en el arranque.
8. Mounte el sistema de archivos.
9. Modifique la configuración para el sistema de archivos nuevo, de tal manera que los archivos binarios no puedan ser ejecutados (cambie la configuración por defecto a `noexec` en la entrada /mnt/tempdir). Luego monte nuevamente el sistema de archivos y copie un archivo ejecutable (algo /bin/ls) a `/mnt/tempdir` e intente ejecutarlo. Debería obtener un error, ¿a qué se debe?

Una vez que esté listo puede volver a la situación original removiendo la entrada desde /etc/fstab.

Solution 12.2

Solución con una partición física

1. No mostraremos los pasos detalles de **fdisk**, ya que lo hemos visto anteriormente. Para efectos deñ ejercicio asumiremos que la partición creada es /dev/sda11.

   ```
   $ sudo fdisk /dev/sda
   .....
   w
   $ partprobe -s
   ```

   A veces **partprobe** no funciona y para asegurarse de que el sistema esté al tanto de la partición nueva es necesario reiniciar.

2. ```
   $ sudo mkfs -t ext4         -v /dev/sda11
   $ sudo mkfs -t ext4 -b 2048 -v /dev/sda11
   $ sudo mkfs -t ext4 -b 4096 -v /dev/sda11
   ```

   Note que el parámetro `-v` (reporte detallado) le proveerá la información requerida; verá que para una partición pequeña como esta el valor por defecto es de 1024 bloques de byte.

3. ```
   $ sudo mkdir /mnt/tempdir
   $ sudo mount /dev/sda11 /mnt/tempdir
   $ mount | grep tempdir
   ```


4.
```
$ sudo umount /mnt/tempdir
$ sudo mount -o ro /dev/sda11 /mnt/tempdir
```

   Si obtiene un error mientras desmonta el dispositivo, asegúrese que no está actualmente en ese directorio.

5.
```
$ sudo touch /mnt/tempdir/afile
```

6.
```
$ sudo umount /mnt/tempdir
```

7. Agregue esta línea a /etc/fstab:

```
/dev/sda11 /mnt/tempdir ext4 defaults 1 2
```

8.
```
$ sudo mount /mnt/tempdir
$ sudo mount | grep tempdir
```

9. Cambie la línea en /etc/fstab a:

```
/dev/sda11 /mnt/tempdir ext4 noexec 1 2
```

   Luego haga:

```
$ sudo mount -o remount /mnt/tempdir
$ sudo cp /bin/ls /mnt/tempdir
$ /mnt/tempdir/ls
```

   Debería obtener un error, ¿a qué se debe?

Solución de archivo loopback

1.
```
$ sudo dd if=/dev/zero of=/tmp/imagefile bs=1M count=250
```

2.
```
$ sudo mkfs -t ext4         -v
$ sudo mkfs -t ext4 -b 2048 -v /imagefile
$ sudo mkfs -t ext4 -b 4096 -v /imagefile
```

   Será advertido en relación a que este es un archivo y no una partición, sólo siga adelante.

   Note que el parámetro `-v` (reporte detallado) le proveerá la información requerida; verá que para una partición pequeña como esta el valor por defecto es de 1024 bloques de byte.

3.
```
$ sudo mkdir /mnt/tempdir
$ sudo mount -o loop /imagefile /mnt/tempdir
$ mount | grep tempdir
```

4.
```
$ sudo umount /mnt/tempdir
$ sudo mount -o ro,loop /imagefile /mnt/tempdir
```

   Si obtiene un error mientras desmonta el dispositivo, asegúrese que no está actualmente en ese directorio.

5.
```
$ sudo touch /mnt/tempdir/afile
```

6.
```
$ sudo umount /mnt/tempdir
```

7. Agregue esta línea a /etc/fstab:

```
/imagefile /mnt/tempdir ext4 loop 1 3
```

8.
```
$ sudo mount /mnt/tempdir
$ sudo mount | grep tempdir
```

9. Cambie la línea en /etc/fstab a:


```
/tmp/imagefile /mnt/tempdir ext4 loop,noexec 1 3
```

Luego haga:

```
$ sudo mount -o remount /mnt/tempdir
$ sudo cp /bin/ls /mnt/tempdir
$ /mnt/tempdir/ls
```

Debería obtener un error, ¿a qué se debe?

Capitulo 13

Características del sistema de archivos: intercambio, cuotas, uso

Lab 13.1: Gestión del área de intercambio

Examine el área de intercambio actual haciendo:

```
$ cat /proc/swaps
```

```
Filename                                Type            Size    Used    Priority
/dev/sda11                              partition       4193776 0       -1
```

Agregaremos más espacio de área de intercambio ya sea usando una partición o un archivo. Para usar un archivo realizaremos lo siguiente:

```
$ dd if=/dev/zero of=swpfile bs=1M count=1024
```

```
1024+0 records in
1024+0 records out
1073741824 bytes (1.1 GB) copied, 1.30576 s, 822 MB/s
```

```
$ mkswap swpfile
```

```
Setting up swapspace version 1, size = 1048572 KiB
no label, UUID=85bb62e5-84b0-4fdd-848b-4f8a289f0c4c
```

En el caso de una partición real sólo ejecute **mkswap** con el dispositivo de la partición, pero tenga en cuenta que toda la información sobre ésta será eliminada.

Active el espacio de intercambio nuevo:

```
$ sudo swapon swpfile
```

```
swapon: /tmp/swpfile: insecure permissions 0664, 0600 suggested.
swapon: /tmp/swpfile: insecure file owner 500, 0 (root) suggested.
```

RHEL 7 advierte que hay un problema de seguridad en los permisos, lo cual solucionaremos de la siguiente:

```
$ sudo chown root:root swpfile
$ sudo chmod 600 swpfile
```

y asegúrese que está siendo usada:

```
$ cat /proc/swaps
```

```
Filename                                Type            Size    Used    Priority
/dev/sda11                              partition       4193776 0       -1
/tmp/swpfile                            file            1048572 0       -2
```

Fíjese en el campo `Priority`; las particiones o archivos de intercambio de menor prioridad no serán utilizadas hasta que las de mayor prioridad están llenas.

Desactive el archivo de área de intercambio y bórrelo para ahorrar ese espacio:

```
$ sudo swapoff swpfile
$ sudo rm swpfile
```

Lab 13.2: Cuotas del sistema de archivos

1. Modifique la entrada en /etc/fstab para que su sistema de archivos nuevo use cuotas. Cambie `noexec` a `usrquota` en la entrada para /mnt/tempdir. Luego desmonte y monte nuevamente el sistema de archivos.
2. Inicialice las cuotas en el sistema de archivos nuevo y luego habilite el sistema de cuotas.
3. Configure algunos límites de cuotas para el usuario normal: un límite soft de 500 bloques y un límite hard de 1000 bloques.
4. Como usuario normal, use **dd** para crear algunos archivos e intentar superar los límites de cuota. Cree bigfile1 (200 bloques) y bigfile2 (400 bloques).

   Debería recibir una advertencia. ¿A qué se debe?
5. Cree bigfile3, de 600 bloques.

   Ahora debería recibir un mensaje de error. ¿Por qué? Revise meticulosamente los tamaños de los archivos.
6. Elimine la línea de montado persistente que había insertado en /etc/fstab.

Solution 13.2

1. Modifique /etc/fstab para tener una de las dos líneas que se muestran, de acuerdo a si tiene una partición real o un archivo loopback:

   ```
   /dev/sda11     /mnt/tempdir ext4 usrquota      1 2
   /tmp/imagefile /mnt/tempdir ext4 loop,usrquota 1 2
   ```

   Luego móntelo nuevamente:

   ```
   $ sudo mount -o remount /mnt/tempdir
   ```


2.
```
$ sudo quotacheck -u /mnt/tempdir
$ sudo quotaon -u /mnt/tempdir
$ sudo chown student.student /mnt/tmpdir
```

   Usualmente no es necesario realizar lo que se muestra en la línea, pero lo estamos haciendo para que la próxima parte sea más fácil.

3. Reemplace la cuenta `student` por su nombre de usuario.

4.
```
$ sudo edquota -u student
```

5.
```
$ cd /mnt/tempdir
$ dd if=/dev/zero of=bigfile1 bs=1024 count=200
200+0 records in
200+0 records out
204800 bytes (205 kB) copied, 0.000349604 s, 586 MB/s
$ quota
Disk quotas for user student (uid 500):
Filesystem blocks quota lim grace files qu lim gr
/dev/sda11    200   500  1000   1   0   0
$ dd if=/dev/zero of=bigfile2 bs=1024 count=400
sda11: warning, user block quota exceeded.
400+0 records in
400+0 records out
4096600 bytes (410 kB) copied, 0.000654847 s, 625 MB/s
```

   Create bigfile3 (600 blocks).

6.
```
$ quota
Disk quotas for user student (uid 500):
Filesystem blocks quota limit grace files qu lim gr
/dev/sda11    600*   500  1000 6days   2  0  0
$ dd if=/dev/zero of=bigfile3 bs=1024 count=600
sda11: write failed, user block limit reached.
dd: writing ‘bigfile3’: Disk quota exceeded
401+0 records in
400+0 records out
409600 bytes (410 kB) copied, 0.00177744 s, 230 MB/s
$ quota
Disk quotas for user student (uid 500):
Filesystem blocks  quota limit grace files quota limit grace
/dev/sda11   1000*   500  1000 6days     3     0     0
$ ls -l
total 1068
-rw------- 1 root    root      7168 Dec 10 18:56 aquota.user
-rw-rw-r-- 1 student student 204800 Dec 10 18:58 bigfile1
-rw-rw-r-- 1 student student 409600 Dec 10 18:58 bigfile2
-rw-rw-r-- 1 student student 409600 Dec 10 19:01 bigfile3
drwx------ 2 root    root     16384 Dec 10 18:47 lost+found
-rwxr-xr-x 1 root    root     41216 Dec 10 18:52 more
```

   Examine de cerca los tamaños de los archivos.

7. Restablezca /etc/fstab a su contenido original.

Capitulo 14

Sistemas de archivos Ext2/Ext3/Ext4

Lab 14.1: Desfragmentación

Quienes recién conocen **Linux** suelen sorprenderse de que no se hable de herramientas de **desfragmentación** del sistema de archivos, debido a que tales programas son muy utilzados en el mundo de **Windows**.

Sin embargo, los sistemas de archivos nativos en sistemas operativos tipo **UNIX**, incluyendo **Linux**, tienden a no sufrir problemas serios de fragmentación.

Esto se debe principalmente a que no tratan de poner archivos en las regiones más internas del disco, en donde el acceso es más rápido. En vez de eso, dejan espacio libre a través del disco, de tal forma de que cuando un archivo va a ser creado, hay mejores posibilidades de que haya una región de bloques libres lo suficientemente grande como para contener el archivo completo, ya sea en una o pocas partes.

En cuanto al hardware moderno, el concepto de regiones internas del disco no es tan claro debido a los cambios que está experimentando la tecnología; en cuanto a los dispositivos **SSDs**, la defragmentación podría acortar la vida útil del almacenamiento, debido a que posee ciclos finitos de lectura/borrado/escritura.

Además, los sistemas de archivos con **journaling** más nuevos (incluyendo **ext4**), trabajan con **extents** (regiones contiguas grandes) por diseño.

Pese a lo anterior, existe una herramienta para desfragmentar sistemas de archivos **ext4**:

```
$ sudo e4defrag

Usage   : e4defrag [-v] file...| directory...| device...
        : e4defrag  -c  file...| directory...| device...
```

e4defrag es parte del paquete **e2fsprogs** y debería estar en todas las distribuciones modernas de **Linux**, aunque no viene en **RHEL 6**, la cual se está quedando atrás.

Las únicas dos opciones son las siguientes:

- `-v`: Muestra los detalles de la operación.

- `-c`: No haga nada realmente, sólo analizar y reportar.

El argumento puede ser:

- Un archivo
- Un directorio
- Un dispositivo completo

Ejemplos:

```
$ sudo e4defrag -c /var/log

<Fragmented files>                            now/best       size/ext
1. /var/log/lastlog                             5/1              9 KB
2. /var/log/sa/sa24                             3/1             80 KB
3. /var/log/rhsm/rhsm.log                       2/1            142 KB
4. /var/log/messages                            2/1           4590 KB
5. /var/log/Xorg.1.log.old                      1/1             36 KB

 Total/best extents                           120/112
 Average size per extent                      220 KB
 Fragmentation score                          1
 [0-30 no problem: 31-55 a little bit fragmented: 56- needs defrag]
 This directory (/var/log) does not need defragmentation.
 Done.

$ sudo e4defrag  /var/log

ext4 defragmentation for directory(/var/log)
[2/152]/var/log/Xorg.2.log:     100%    [ OK ]
[3/152]/var/log/Xorg.0.log.old: 100%    [ OK ]
[4/152]/var/log/messages-20141019.gz:   100%    [ OK ]
[5/152]/var/log/boot.log:       100%    [ OK ]
[7/152]/var/log/cups/page_log-20140924.gz:      100%    [ OK ]
[8/152]/var/log/cups/access_log-20141019.gz:    100%    [ OK ]
[9/152]/var/log/cups/access_log:        100%    [ OK ]
[10/152]/var/log/cups/error_log-20141018.gz:    100%    [ OK ]
[11/152]/var/log/cups/error_log-20141019.gz:    100%    [ OK ]
[12/152]/var/log/cups/access_log-20141018.gz:   100%    [ OK ]
[14/152]/var/log/cups/page_log-20141018.gz:     100%    [ OK ]
...
[152/152]/var/log/Xorg.1.log.old:       100%    [ OK ]

        Success:                        [ 112/152 ]
        Failure:                        [ 40/152 ]
```

Ejecute **e4defrag** en varios archivos, directorios y dispositivos completos, siempre intentando con `-c` primero.

Encontrará que generalmente los sistemas de archivos **Linux** tienden a necesitar defragmentación cuando están cercanos a llenarse, sobre el 90 por ciento, o cuando son pequeños y tienen archivos relativamente grandes, como sucede con las particiones usadas para **boot**.

Lab 14.2: Modificación de parámetros del sistema de archivos con tune2fs

Vamos a modificar algunas propiedades de un sistema de archivos formateado con **ext4**. Esto no requiere desmontar el sistema de archivos.

En el ejercicio de a continuación puede trabajar con un archivo de imagen que puede crear de la siguiente forma:


```
$ dd if=/dev/zero of=imagefile bs=1M count=1024
```

o puede reemplazar `imagefile` por /dev/sdaX , usando cualquier partición que contenga el sistema de archivos que desea modificar.

1. Usando **dumpe2fs** obtenga información acerca del sistema de archivos del cual desea ajustar las propiedades.
2. Determinar la configuración para el conteo máximo de montaje luego del cual el sistema de archivos será forzado a verificación y modifíquelo a `30`.
3. Configure el parámetro `Check interval` a tres semanas (la cantidad de tiempo luego del cual un sistema de archivos es forzado a una verificación).
4. Calcule el porcentaje de bloques reservados y luego reconfigúrelo a 10%.

Solution 14.2

1.
```
$ dumpe2fs imagefile > dump_results
```

2.
```
$ grep -i "Mount count" dump_results
Mount count:              0
Maximum mount count:      -1
$ sudo tune2fs  -c  30 imagefile
$ grep -i "Mount count" dump_results
Mount count:              0
Maximum mount count:      30
```

3.
```
$ grep -i "Check interval" dump_results
Check interval:           0 (<none>)
$ sudo tune2fs -i 3w imagefile
$ grep -i "Check interval" dump_results
Check interval:           1814400 (3 weeks)
```

4.
```
$ grep -i "Block Count" dump_results
Block count:              131072
Reserved block count:     6553

$ echo "scale=4; 6553/131072" | bc
.0499
$ sudo tune2fs -m 10 imagefile
$ tune2fs 1.42.9 (28-Dec-2013)
Setting reserved blocks percentage to 10% (13107 blocks)
$ grep -i "Block Count" dump_results
Block count:              131072
Reserved block count:     13107
```


Capitulo 15

Los sistemas de archivos XFS y btrfs

Lab 15.1: Más información acerca de xfs

No tenemos un ejercicio de laboratorio detallado para **xfs**; muchos sistemas todavía no tienen instalados los módulos del kernel y herramientas de usuario importantes. Sin embargo, si su kernel **Linux** y su distribución lo soportan, puede crear un sistema de archivos fácilmente con `mkfs -t xfs`.

Puede encontrar información relacionada a las herramientas **xfs** con:

```
$ man -k xfs

attr (1)             - extended attributes on XFS filesystem objects
filesystems (5)      - Linux file-system types: minix, ext, ext2, ext3, ext4,...
fs (5)               - Linux file-system types: minix, ext, ext2, ext3, ext4,...
fsck.xfs (8)         - do nothing, successfully
fsfreeze (8)         - suspend access to a filesystem (Linux Ext3/4, ReiserFS...
mkfs.xfs (8)         - construct an XFS filesystem
pmdaxfs (1)          - XFS filesystem performance metrics domain agent (PMDA)
xfs (5)              - layout of the XFS filesystem
xfs_admin (8)        - change parameters of an XFS filesystem
xfs_bmap (8)         - print block mapping for an XFS file
xfs_copy (8)         - copy the contents of an XFS filesystem
xfs_db (8)           - debug an XFS filesystem
xfs_estimate (8)     - estimate the space that an XFS filesystem will take
xfs_freeze (8)       - suspend access to an XFS filesystem
xfs_fsr (8)          - filesystem reorganizer for XFS
xfs_growfs (8)       - expand an XFS filesystem
xfs_info (8)         - expand an XFS filesystem
xfs_io (8)           - debug the I/O path of an XFS filesystem
xfs_logprint (8)     - print the log of an XFS filesystem
xfs_mdrestore (8)    - restores an XFS metadump image to a filesystem image
xfs_metadump (8)     - copy XFS filesystem metadata to a file
xfs_mkfile (8)       - create an XFS file
xfs_ncheck (8)       - generate pathnames from i-numbers for XFS
```

```
xfs_quota (8)        - manage use of quota on XFS filesystems
xfs_repair (8)       - repair an XFS filesystem
xfs_rtcp (8)         - XFS realtime copy command
xfsdump (8)          - XFS filesystem incremental dump utility
xfsinvutil (8)       - xfsdump inventory database checking and pruning utility
xfsrestore (8)       - XFS filesystem incremental restore utility
xqmstats (8)         - Display XFS quota manager statistics from /proc
```

Lea acerca de estos programas y vea si puede realizar algunas operaciones con el sistema de archivos que creó.

Lab 15.2: Más información acerca de btrfs

No tenemos un ejercicio de laboratorio detallado para **btrfs**; muchos sistemas todavía no tienen instalados los módulos del kernel y herramientas de usuario importantes. Sin embargo, si su kernel **Linux** y su distribución lo soportan, puede crear un sistema de archivos fácilmente con `mkfs -t btrfs`.

Puede encontrar información relacionada a las herramientas **btrfs** con:

```
$ man -k btrfs

btrfs-image (8)      - create/restore an image of the filesystem
btrfs-show (8)       - scan the /dev directory for btrfs partitions and print...
btrfsck (8)          - check a btrfs filesystem
btrfsctl (8)         - control a btrfs filesystem
mkfs.btrfs (8)       - create an btrfs filesystem
btrfs (8)            - control a btrfs filesystem
btrfs-convert (8)    - convert ext2/3/4 to btrfs.
btrfs-debug-tree (8) - dump Btrfs filesystem metadata into stdout.
btrfs-find-root (8)  - filter to find btrfs root.
btrfs-map-logical (8) - map btrfs logical extent to physical extent
btrfs-show-super (8) - show btrfs superblock information stored in devices
btrfs-zero-log (8)   - clear out log tree.
btrfstune (8)        - tune various filesystem parameters.
```

Lea acerca de estos programas y vea si puede realizar algunas operaciones con el sistema de archivos que creó.

Capitulo 16

Logical Volume Manager (LVM)

Lab 16.1: Volúmenes lógicos

Vamos a crear un volumen lógico usando dos particiones de 250 MB. Supondremos que tiene espacio físico disponible en disco para particionar.

1. Crear dos particiones de 250 MB de tipo volumen lógico (`8e`).
2. Convertir las particiones a volúmenes físicos.
3. Crear un grupo de volúmenes llamado `myvg` y agregue los dos volúmenes físicos al mismo. Use el tamaño por defecto para el extent.
4. Asignar un volumen lógico de 300 MB, llamado `mylvm` desde el grupo de volúmenes `myvg`.
5. Formatear y montar el volumen lógico `mylvm` en /mylvm
6. Use **lvdisplay** para ver información del volumen lógico.
7. Extienda el volumen lógico y el sistema de archivos correspondiente a 350 MB.

Solution 16.1

1. Ejecute:

   ```
   $ sudo fdisk /dev/sda
   ```

   use el disco duro que sea apropiado y cree las dos particiones. Estando en **fdisk** escriba `t` para configurar el tipo de partición a `8e`. Si bien es cierto no es requerido definir el tipo, es buena idea hacerlo para evitar confusiones. Use `w` para reescribir la tabla de partición y salga. Luego haga

   ```
   $ sudo partprobe -s
   ```

o reinicie para asegurarse que las particiones nuevas tomen efecto.

2. Asumiendo que las particiones nuevas son /dev/sdaX y /dev/sdaY:

```
$ sudo pvcreate /dev/sdaX
$ sudo pvcreate /dev/sdaY
$ sudo pvdisplay
```

3.
```
$ sudo vgcreate myvg /dev/sdaX /dev/sdaY
$ sudo vgdisplay
```

4.
```
$ sudo lvcreate -L 300M -n mylvm myvg
$ sudo lvdisplay
```

5.
```
$ sudo mkfs.ext4 /dev/myvg/mylvm
$ mkdir /mylvm
$ sudo mount /dev/myvg/mylvm /mylvm
```

   Si desea que el montaje sea persistente, edite /etc/fstab e incluya la línea:

```
/dev/myvg/mylvm /mylvm ext4 defaults 0 0
```

6.
```
$ sudo lvdisplay
```

7.
```
$ df -h
$ sudo lvextend -L 350M /dev/myvg/mylvm
$ sudo resize2fs /dev/myvg/mylvm
$ df -h
```

   or

```
$ sudo lvextend -r -L +50M /dev/myvg/mylvm
```


Capitulo 17

RAID

Lab 17.1: Creación de un dispositivo RAID

Normalmente cuando se crea un dispositivo **RAID**, deberíamos usar particiones en discos separados. Sin embargo, en este ejercicio probablemente no vamos a disponer de tal hardware.

Por lo tanto vamos a necesitar dos particiones en el mismo disco, o podemos usar particiones **LVM** solo para propósitos de demostración. Note que no podemos usar archivos de imágenes y loopback para este ejercicio.

El proceso será el mismo si las particiones están en un disco o en varios. Aunque obviamente hay pocas razones para crear un **RAID** en un disco solamente.

1. Cree dos particiones de 200 MB de tipo raid (`fd`), ya sea en su disco duro, usando **fdisk**, o con **LVM**.
2. Cree un **RAID 1** y use /dev/md0 para el dispositivo. Use las dos particiones para tal efecto.
3. Formatee el dispositivo **RAID** con sistema de archivos **ext4**. Luego móntelo en /myraid y configure el punto de montaje de forma persistente.
4. Ponga la información acerca de /dev/md0 en el archivo /etc/mdadm.conf usando **mdadm**. Dependiendo de su distribución, este archivo podría no existir previamente.
5. Examine /proc/mdstat para ver el estado de su dispositivo **RAID**.

Solution 17.1

1. Si usted está usando particiones en un disco real, haga lo siguiente:

   ```
   $ sudo fdisk /dev/sda
   ```

y cree las particiones como lo hemos hecho anteriormente. Para propósitos del ejercicio los llamaremos /dev/sdaX y /dev/sdaY. Necesitará correr **partprobe**, **kpartx** o reiniciar luego que ha terminado para asegurarse de que el sistema reconozca las particiones nuevas.

Las particiones **LVM** estarán perfectamente bien para este ejercicio. Pueden crearse de la siguiente forma:

```
$ sudo lvcreate -L 200M -n MD1 VG
$ sudo lvcreate -L 200M -n MD2 VG
```

donde hemos nombrado el grupo de volúmenes como `VG`. No es necesario hacer nada más luego de la creación de las particiones **LVM** nuevas para que el sistema esté al tanto de ellas.

2. ```
   $ sudo mdadm -C /dev/md0 --level=1 --raid-disks=2 /dev/sdaX /dev/sdaY
   ```

   o

   ```
   $ sudo mdadm -C /dev/md0 --level=1 --raid-disks=2 /dev/VG/MD1 /dev/VG/MD2
   ```

3. ```
   $ sudo mkfs.ext4 /dev/md0
   $ sudo mkdir /myraid
   $ sudo mount /dev/md0 /myraid
   ```

   y agregar a /etc/fstab

   ```
   /dev/md0  /myraid    ext4    defaults  0 0
   ```

4. ```
   $ mdadm --detail --scan >> /etc/mdadm.conf
   ```

5. ```
   $ cat /proc/mdstat
   Personalities : [raid1]
   md0 : active raid1 dm-14[1] dm-13[0]
         204736 blocks [2/2] [UU]

   unused devices: <none>
   ```

Probablemente deberá verificar que el volumen **RAID** se monta automáticamente luego de reiniciar el sistema. Cuando esté listo, remueva la línea de /etc/fstab para eliminar las referencias de las particiones utilizadas en este ejercicio.

Capitulo 18

Seguridad del sistema local

Lab 18.1: Seguridad y opciones de mount

Vamos a montar una partición o dispositivo loop con la opción `noexec` para evitar la ejecución de programas que en el sistema de archivos. Si bien es cierto que es posible hacer esto con una partición preexistente, puede ser difícil cambiar el comportamiento mientras la partición está montada. Por lo tanto, para la demostración usaremos un dispositivo loop, lo cual es un procedimiento inofensivo.

1. Cree un archivo vacío, ponga un sistema de archivos en él y móntelo.
2. Copie un archivo ejecutable desde alguna parte a este sistema de archivos y pruebe que funciona en el lugar nuevo.
3. Desmóntelo y móntelo nuevamente con la opción `noexec`.
4. Pruebe si el ejecutable aún funciona. Debería dar un error debido a la opción `noexec` de mount.
5. Limpie lo anterior.

Solution 18.1

1. ```
   $ dd if=/dev/zero of=image bs=1M count=100
   $ sudo mkfs.ext3 image
   $ mkdir mountpoint
   $ sudo mount -o loop image mountpoint
   ```

2. ```
   $ sudo cp /bin/ls mountpoint
   $ mountpoint/ls
   ```

3. ```
   $ sudo umount mountpoint
   $ sudo mount -o noexec,loop image mountpoint
   ```
   ```

or

```
$ sudo mount -o noexec,remount image mountpoint
```

4. `$ mountpoint/ls`

5. ```
   $ sudo umount mountpoint
   $ rm image
   $ rmdir mountpoint
   ```

Tenga en cuenta que esto no es persistente. Para hacerlo persistente tendría que agregar la opción a /etc/fstab, con una línea como la siguiente:

```
/home/student/image  /home/student/mountpoint    ext3    loop,rw,noexec 0 0
```

Lab 18.2: Más de setuid y scripts

Supongamos que tenemos el siguiente programa en **C** (./writeit.c), el cual intenta sobreescribir un archivo llamado `afile` en el directorio actual:

```
#include <stdio.h>
#include <unistd.h>
#include <fcntl.h>
#include <stdlib.h>
#include <string.h>
#include <stdlib.h>
#include <sys/stat.h>

int main(int argc, char *argv[])
{
        int fd, rc;
        char *buffer = "TESTING A WRITE";
        fd = open("./afile", O_RDWR | O_CREAT | O_TRUNC, S_IRUSR | S_IWUSR);
        rc = write(fd, buffer, strlen(buffer));
        printf("wrote %d bytes\n", rc);
        close(fd);
        exit(EXIT_SUCCESS);
}
```

Si está tomando la versión de autoaprendizaje de este curso, el código fuente está disponible para su descarga desde la pantalla **Laboratorio**.

Si el programa se llama writeit.c, puede ser compilado haciendo:

```
$ make writeit
```

o de forma equivalente:

```
$ gcc -o writeit writeit.c
```

Si intenta ejecutar este programa como un usuario normal sobre un archivo del cual root es el dueño, obtendrá lo siguiente:

```
$ sudo touch afile
$ ./writeit

wrote -1 bytes
```


```

pero si lo ejecuta como root:

```
$ sudo ./writeit

wrote 15 bytes
```

Por lo tanto, el usuario root fue capaz de sobreescribir el archivo del cual es dueño, pero un usuario normal no podría.

Tenga en cuenta que no ayudará cambiar el dueño de **writeit** a root:

```
$ sudo chown root.root writeit
$ ./writeit

wrote -1 bytes
```

porque todavía no le permitirá sobreescribir afile.

Al configurar el bit **setuid** usted puede habilitar a cualquier usuario normal para que lo haga:

```
$ sudo chmod +s writeit
$ ./writeit

wrote 15 bytes
```

Usted se podría preguntar por qué simplemente no escribimos un script que realice la operación, en vez de escribir y compilar un programa ejecutable.

Bajo **Linux**, si se cambia el **setuid** en un script ejecutable, no hará nada a menos que usted cambie el bit setuid en la shell (tal como bash), lo cual sería un gran error; cualquier cosa que se corra desde ahí podría escalar privilegios.

Capitulo 19

Módulos de seguridad de Linux

Lab 19.1: SELinux

Antes de comenzar este ejercicio verifique que **SELinux** esté instalado y en modo **enforcing**. Edite /etc/selinux/config y reinicie si es necesario.

Obviamente solo puede hacer esto en un sistema que tiene **SELinux** instalado. En este ejemplo estamos usando **RHEL**.

1. Instale los paquetes **vsftpd** y **ftp**.
2. Cree una cuenta **user1** con la contraseña **password**.
3. Cámbiese a la cuenta **user1** y escriba algún texto en un archivo llamado /home/user1/user1file.
4. Salga de la cuenta **user1** y asegúrese que el servicio **ftp** (el nombre del servicio es **vsftpd**) esté en ejecución.
5. Haga **ftp** al `localhost`, conéctese como `user1` e intente obtener user1file. Esto debería fallar.

   Tenga en cuenta que esto podría fallar ya sea al conectarse a la cuenta o al transferir el archivo. La solución a ambos problemas es el mismo, por lo cual el ejercicio no se verá afectado. Las diferencias en el comportamiento son una consecuencia de las diferencias en las políticas de **SELinux**.
6. Verifique /var/log/messages para determinar porqué. Usted debería ver un error relativo a **setroubleshoot**. Ejecute el comando **sealert** mostrado anteriormente.
7. Resuelva el problema e intente de nuevo hacer **ftp**, conectarse como `user1` y obtener el archivo user1file. Esta vez debería funcionar.

Solution 19.1

1. `$ sudo yum install vsftpd ftp`
2. `$ sudo useradd user1`

   `$ sudo passwd user1`

```
Changing password for user user1.
New password: password
BAD PASSWORD: The password fails the dictionary check - it is based on a dictionary word
Retype new password: password
passwd: all authentication tokens updated successfully.
```

3. `$ sudo su - user1`
   `[user1@rhel7 ~]$ echo 'file created at /home/user1' > user1file`
   `[user1@rhel7 ~]$ ls`
   `user1file`

4. `[user1@rhel7 ~]$ exit`

   `$ sudo systemctl status vsftpd.service`

   ```
   vsftpd.service - Vsftpd ftp daemon
      Loaded: loaded (/usr/lib/systemd/system/vsftpd.service; disabled)
      Active: active (running) since Fri 2014-11-21 14:08:14 CET; 32min ago
   ...
   ```

5. `$ ftp localhost`

   ```
   Trying ::1...
   Connected to localhost (::1).
   220 (vsFTPd 3.0.2)
   Name (localhost:peter): user1
   331 Please specify the password.
   Password: password
   230 Login successful.
   Remote system type is UNIX.
   Using binary mode to transfer files.
   ftp> get user1file
   local: user1file remote: user1file
   229 Entering Extended Passive Mode (|||35032|).
   550 Failed to open file.
   ftp> quit
   221 Goodbye.
   ```

6. `$ tail /var/log/messages`

   ```
   Nov 21 14:23:26 rhel7 setroubleshoot: SELinux is preventing /usr/sbin/vsftpd from read access on the file .
   For complete SELinux messages. run sealert -l 7f8e5e6f-bcee-4c59-9cd1-72b90fb1f462
   *****  Plugin catchall_boolean (47.5 confidence) suggests   ******************

   If you want to allow ftp to home dir
   Then you must tell SELinux about this by enabling the 'ftp_home_dir' boolean.

   Do
   setsebool -P ftp_home_dir 1
   ```

   Notice that the suggestion to fix the issue can be found at the log file, and it is not even necessary to run **sealert**.

7. `$ sudo setsebool -P ftp_home_dir 1`

   `$ ftp localhost`

   ```
   Trying ::1...
   Connected to localhost (::1).
   220 (vsFTPd 3.0.2)
   Name (localhost:peter): user1
   331 Please specify the password.
   Password:
   230 Login successful.
   Remote system type is UNIX.
   Using binary mode to transfer files.
   ```



```
ftp> get user1file
local: user1file remote: user1file
229 Entering Extended Passive Mode (|||18769|).
150 Opening BINARY mode data connection for user1file (28 bytes).
226 Transfer complete.
28 bytes received in 4.2e-05 secs (666.67 Kbytes/sec)
ftp> quit
221 Goodbye.


$ cat user1file
file created at /home/user1
```


Capitulo 20

Procesos

Lab 20.1: Control de procesos con ulimit

Por favor haga:

```
$ help ulimit
```

y lea /etc/security/limits.conf antes de ejecutar los pasos que vienen a continuación.

1. Inicie una terminal shell ejecutando **bash** (o abra una terminal nueva) de tal forma que los cambios sean efectivos en esa shell solamente. Vea el límite actual del número de archivos abiertos y explícitamente el número de los límites hard y soft.
2. Configure el valor del límite hard y verifique que haya funcionado.
3. Configure el valor del límite hard a 2048 y verifique que haya funcionado.
4. Intente configurar el límite al valor previo. ¿Funcionó?

Solution 20.1

1.
```
$ bash
$ ulimit -n
1024
$ ulimit -S -n
1024
$ ulimit -H -n
4096
```

2.
```
$ ulimit -n hard
$ ulimit -n
4096
```

3.
```
$ ulimit -n 2048
$ ulimit -n
2048
```

4.
```
$ ulimit -n 4096
bash: ulimit: open files: cannot modify limit: Operation not permitted
$ ulimit -n
2048
```

   ¡No es posible hacerlo!

   Tenga en cuenta que si hubiéramos escogido un límite diferente, como el tamaño del stack (`-s`), podríamos aumentar el valor nuevamente debido a que el límite hard es `unlimited`.

Lab 20.2: Examinar la actividad de System V IPC

System V IPC es un método bastante antiguo de Comunicación Entre Procesos (**IPC**) que se remonta a los primeros días de **UNIX**. Este involucra tres mecanismos:

1. **Segmentos de memoria compartida**
2. **Semáforos**
3. **Colas de mensajes**

Algunos programas modernos tienen a usar los métodos de **POSIX IPC** para esos tres mecanismos, pero todavía se puede encontrar un montón de aplicaciones **System V IPC**.

Ejecute el siguiente comando para obtener un resumen general de la actividad **System V IPC** en su sistema:

```
$ ipcs

------ Message Queues --------
key        msqid      owner      perms      used-bytes   messages

------ Shared Memory Segments --------
key        shmid      owner      perms      bytes      nattch     status
0x01114703 0          root       600        1000       6
0x00000000 98305      coop       600        4194304    2          dest
0x00000000 196610     coop       600        4194304    2          dest
0x00000000 23068675   coop       700        1138176    2          dest
0x00000000 23101444   coop       600        393216     2          dest
0x00000000 23134213   coop       600        524288     2          dest
0x00000000 24051718   coop       600        393216     2          dest
0x00000000 23756807   coop       600        524288     2          dest
0x00000000 24018952   coop       600        67108864   2          dest
0x00000000 23363593   coop       700        95408      2          dest
0x00000000 1441811    coop       600        2097152    2          dest

------ Semaphore Arrays --------
key        semid      owner      perms      nsems
0x00000000 98304      apache     600        1
0x00000000 131073     apache     600        1
0x00000000 163842     apache     600        1
0x00000000 196611     apache     600        1
0x00000000 229380     apache     600        1
```


Tenga en cuenta que casi todos los segmentos de memoria compartida actualmente en ejecución tienen `0` en el campo key (también conocido como `IPC_PRIVATE`), lo cual significa que solo se comparten entre procesos en una relación padre/hijo. Además, todos con la excepción de uno, están marcados para destrucción, ya que no tienen procesos asociados.

Es posible obtener más información acerca de los procesos que han creado los segmentos y que se han asociado a ellos con:

```
$ ipcs -p

------ Message Queues PIDs --------
msqid      owner      lspid      lrpid

------ Shared Memory Creator/Last-op PIDs --------
shmid      owner      cpid       lpid
0          root       1023       1023
98305      coop       2265       18780
196610     coop       2138       18775
23068675   coop       989        1663
23101444   coop       989        1663
23134213   coop       989        1663
24051718   coop       20573      1663
23756807   coop       10735      1663
24018952   coop       17875      1663
23363593   coop       989        1663
1441811    coop       2048       20573
```

Por lo tanto, haciendo:

```
$ ps aux |grep -e 20573 -e 2048

coop      2048  5.3  3.7 1922996 305660 ?      Rl   Oct27  77:07 /usr/bin/gnome-shell
coop     20573  1.9  1.7 807944 141688 ?       Sl   09:56   0:01 /usr/lib64/thunderbird/thunderbird
coop     20710  0.0  0.0 112652  2312 pts/0    S+   09:57   0:00 grep --color=auto -e 20573 -e 2048
```

vemos que **thunderbird** está usando un segmento de memoria compartida creado por **gnome-shell**.

Realice estos pasos en su sistema e identifique los diversos recursos que están siendo usados y por quién. ¿Hay alguna **fuga potencial** en el sistema (recursos compartidos que no están en uso por ningún proceso)? Por ejemplo:

```
$ ipcs

....
------ Shared Memory Segments --------
key        shmid      owner      perms      bytes      nattch     status
....
0x00000000 622601     coop       600        2097152    2          dest
0x0000001a 13303818   coop       666        8196       0
....
```

muestra un segmento de memoria compartida que no tiene procesos asociados y que no está marcado para ser destruido. Por lo anterior, si el proceso continúa así para siempre y no se asocia a ningún proceso, podría significar una fuga de memoria.

Capitulo 21

Señales

Lab 21.1: Examinar prioridades de las señalas y su ejecución

Le hemos proporcionado un programa en **C** que incluye un manejador de señales el que puede manejar cualquier señal. El manejador en cuestión evita hacer llamadas al sistema (tales como las que podrían ocurrir mientras se realizan operaciones de E/S).

```
/*
 * Examining Signal Priorities and Execution.
 *
 * The code herein is: Copyright the Linux Foundation, 2014
 * Author: J. Cooperstein
 *
 * This Copyright is retained for the purpose of protecting free
 * redistribution of source.
 *
 * This code is distributed under Version 2 of the GNU General Public
 * License, which you should have received with the source.
 *
 @*/

#include <stdio.h>
#include <unistd.h>
#include <signal.h>
#include <stdlib.h>
#include <string.h>
#include <pthread.h>

#define NUMSIGS 64

/* prototypes of locally-defined signal handlers */

void (sig_handler) (int);
```

```
int sig_count[NUMSIGS + 1];        /* counter for signals received */
volatile static int line = 0;
volatile int signumbuf[6400], sigcountbuf[6400];

int main(int argc, char *argv[])
{
        sigset_t sigmask_new, sigmask_old;
        struct sigaction sigact, oldact;
        int signum, rc, i;
        pid_t pid;

        pid = getpid();

        /* block all possible signals */
        rc = sigfillset(&sigmask_new);
        rc = sigprocmask(SIG_SETMASK, &sigmask_new, &sigmask_old);

        /* Assign values to members of sigaction structures */
        memset(&sigact, 0, sizeof(struct sigaction));
        sigact.sa_handler = sig_handler;        /* we use a pointer to a handler */
        sigact.sa_flags = 0;        /* no flags */
        /* VERY IMPORTANT */
        sigact.sa_mask = sigmask_new;        /* block signals in the handler itself  */

        /*
         * Now, use sigaction to create references to local signal
         * handlers * and raise the signal to myself
         */

        printf
            ("\nInstalling signal handler and Raising signal for signal number:\n\n");
        for (signum = 1; signum <= NUMSIGS; signum++) {
                if (signum == SIGKILL || signum == SIGSTOP || signum == 32
                    || signum == 33) {
                        printf("  --");
                        continue;
                }
                sigaction(signum, &sigact, &oldact);
                /* send the signal 3 times! */
                rc = raise(signum);
                rc = raise(signum);
                rc = raise(signum);
                if (rc) {
                        printf("Failed on Signal %d\n", signum);
                } else {
                        printf("%4d", signum);
                        if (signum % 16 == 0)
                                printf("\n");
                }
        }
        fflush(stdout);

        /* restore original mask */
        rc = sigprocmask(SIG_SETMASK, &sigmask_old, NULL);

        printf("\nSignal  Number(Times Processed)\n");
        printf("----------------------------------------\n");
        for (i = 1; i <= NUMSIGS; i++) {
                printf("%4d:%3d  ", i, sig_count[i]);
                if (i % 8 == 0)
                        printf("\n");
        }
        printf("\n");
```



```
        printf("\nHistory: Signal  Number(Count Processed)\n");
        printf("-----------------------------------------\n");
        for (i = 0; i < line; i++) {
                if (i % 8 == 0)
                        printf("\n");
                printf("%4d(%1d)", signumbuf[i], sigcountbuf[i]);
        }
        printf("\n");
        exit(EXIT_SUCCESS);
}

void sig_handler(int sig)
{
        sig_count[sig]++;
        signumbuf[line] = sig;
        sigcountbuf[line] = sig_count[sig];
        line++;
}
```

Si está tomando la versión autodidacta en línea de este curso, encontrará el código fuente disponible para su descarga en la pantalla **Lab**.

Necesitará compilarlo y ejecutarlo como se muestra a continuación:

```
$ gcc -o signals signals.c
$ ./signals
```

Al ser ejecutado, el programa realiza lo siguiente:

- No envía las señales `SIGKILL` o `SIGSTOP`, las cuales no pueden ser manejadas y siempre finalizan un programa.
- Almacena la secuencia de señales a medida en que llegan y actualiza un arreglo de contadores para cada señal que indica cuántas veces la señal ha sido manejada.
- Comienza por suspender el proceso de todas las señales y luego instala un conjunto nuevo de manejadores de señal para todas ellas.
- Envía cada señal posible múltiples veces, luego desbloquea el manejo de señales e invoca a los manipuladores de señales que estaban en espera.
- Imprime las estadísticas, incluyendo:
  - El número total de veces que cada señal fue recibida.
  - El orden en el cual se recibieron las señales, señalando cada vez el número total de veces que la señal se había recibido hasta ese momento.

Tenga en cuenta lo siguiente:

- Si una señal determinada **se emite** en varias oportunidades mientras el proceso las había bloqueado, ¿el proceso las **recibe** múltiples veces? ¿El comportamiento de señales en **tiempo real** es diferente de las señales normales?
- ¿El proceso recibe todas las señales, o algunas de ellas son manejadas antes que lleguen a él?
- ¿En qué orden se reciben las señales?

La señal `SIGCONT` (18 en **x86** puede que no logre llegar a destino, ¿se le ocurre por qué?

Nota:

En algunas distribuciones **Linux** las señales 32 y 33 no pueden ser bloqueadas y causarán que el programa falle. A pesar de que los archivos de cabecera del sistema indican `SIGRTMIN=32`, el comando `kill -l` indica `SIGRTMIN=34`.

Tenga en cuenta que **POSIX** dice que se deberían usar nombres en vez de números, los cuales están habilitados para ser completamente dependientes de la implementación.

En general se debería evitar el envío de estas señales.

Capitulo 22

Monitoreo del sistema

Lab 22.1: Uso de stress

stress es un programa escrito en lenguaje **C** por Amos Waterland de la Universidad de Oklahoma, licenciado bajo la **GPL v2**. Está diseñado para imponer una cantidad configurable de estrés a través de la generación de varios tipos de carga de trabajo en el sistema.

Si usted tiene suerte podrá instalar **stress** directamente desde el sistema de empaquetamiento de su distribución. De lo contrario, puede obtener el código fuente desde `http://people.seas.harvard.edu/~apw/stress` y luego compilarlo de la siguiente forma:

```
$ tar zxvf stress-1.0.4.tar.gz
$ cd stress-1.0.4
$ ./configure
$ make
$ sudo make install
```

Pueden existir paquetes binarios descargables en los formatos **.deb** y **.rpm**; revise el sitio web para detalles y ubicaciones de los mismos.

Una vez que está instalado, puede hacer:

```
$ stress --help
```

para obtener una lista rápida de opciones, o

```
$ info stress
```

para acceder a una documentación más detallada.

Por ejemplo, el siguiente comando:

```
$ stress -c 8 -i 4 -m 6 -t 20s
```

will:

- Creará 8 procesos intensivos en la CPU, cada uno realizando un cálculo `sqrt()`.
- Creará 4 procesos intensivos de E/S, cada uno realizando una operación `sync()`.
- Creará 6 procesos intensivos de memoria, cada uno realizando una operación `malloc()`, asignando 256 MB por defecto. El tamaño puede cambiarse con la opción `--vm-bytes 128M`.
- Ejecutar la prueba de estrés por 20 segundos.

Luego de haber instalado **stress**, usted puede iniciar el monitor de sistema gráfico, el cual seguramente encontrará en el menú de aplicaciones. También lo puede ejecutar desde la línea de comandos, ya sea **gnome-system-monitor** o **ksysguard**.

Ahora comience a poner estrés en el sistema. Los números exactos que usarán estarán en función de los recursos de su sistema, tales como el número de CPU y el tamaño de la RAM.

Por ejemplo, si hace

```
$ stress -m 4 -t 20s
```

estresa solamente la memoria del sistema.

Juegue con las combinaciones de los parámetros y vea el impacto que generan entre ellos. Probablemente se de cuenta que el programa **stress** es útil para simular diversas condiciones de carga alta.

Capitulo 23

Monitoreo de procesos

Lab 23.1: Procesos

1. Ejecute ps con las opciones `-ef`. Luego ejecútelo de nuevo pero con las opciones `aux`. Note las diferencias en la salida de cada comando.
2. Ejecute **ps** de tal forma que solo se despliegue el ID de proceso, la prioridad, el valor de nice y la línea de comandos del proceso.
3. Ejecute **bash** en la línea de comandos para iniciar una sesión nueva de esa shell. Inicie otra sesión **bash** usando el comando **nice**, pero esta vez dándole a nice un valor de `10`.
4. Ejecute **ps** como en el paso 2 y note las diferencias entre los valores de prioridad y nice. También tome nota de los ID de proceso de las dos sesiones de bash.
5. Usando **renice**, cambie el valor de nice de una de las sesiones de **bash** a 15. Observe nuevamente el cambio en los valores de prioridad y nice.
6. Ejecute **top** y vea la salida a medida en que cambia. Presione `q` para detener el programa.

Solution 23.1

1. ```
   $ ps -ef
   $ ps aux
   ```

2. ```
   $ ps -o pid,pri,ni,cmd
     PID PRI  NI CMD
    2389  19   0 bash
   22079  19   0 ps -o pid,pri,ni,cmd
   ```

   (Nota: no debería haber espacios entre los parámetros.)

3.
```
$ bash
$ nice -n 10 bash
$ ps -o pid,pri,ni,cmd
    2389  19   0 bash
22115  19   0 bash
22171   9  10 bash
22227   9  10 ps -o pid,pri,ni,cmd
```

4.
```
$ renice 15 -p 22227
$ ps -o pid,pri,ni,cmd
  PID PRI  NI CMD
 2389  19   0 bash
22115  19   0 bash
22171   4  15 bash
22246   4  15 ps -o pid,pri,ni,cmd
```

5.
```
$ top
```

Lab 23.2: Monitoreo del estado de los procesos

1. Use **dd** para iniciar un proceso que corra en segundo plano, que lea desde /dev/urandom y escriba a /dev/null.
2. Verifique el estado del proceso. ¿Cuál debería ser?
3. Traiga el proceso a primer caso usando el comando **fg**. Luego teclee `Ctrl-Z`. ¿Qué hace esto? Mire el estado del proceso nuevamente, ¿cuál es ahora?
4. Ejecute el programa **jobs**. ¿Qué información le entrega?
5. Lleve la tarea a segundo plano, luego termínela usando **kill** desde otra ventana.

Solution 23.2

1.
```
$ dd if=/dev/urandom of=/dev/null &
```

2.
```
$ ps -C dd -o pid,cmd,stat
25899 dd if=/dev/urandom of=/dev/ R
```

   Should be `S` or `R`.

3.
```
$ fg
$ ^Z
$ ps -C dd -o pid,cmd,stat
  PID CMD                         STAT
25899 dd if=/dev/urandom of=/dev/ T
```

   ¿El estado debería ser `T`.

4. ejecute el comando **jobs**. ¿Qué información le entrega?

```
$ jobs
[1]+  Stopped                 dd if=/dev/urandom of=/dev/null
```

5. Lleve la tarea de vuelta al segundo plano, luego termínela usando el comando **kill** desde otra ventana.

```
$ fg
$ kill  25899
```


Capitulo 24

Monitoreo y ajuste de E/S

Lab 24.1: bonnie++

bonnie++ es un programa de benchmarking ampliamente disponible que pone a prueba y mide el rendimiento de discos y sistemas de archivos. Proviene de **bonnie**, una implementación anterior.

Los resultados pueden leerse desde la terminal o ser direccionados a un archivo, el que puede ser exportado a formato **csv** (**c**omma **s**eparated **v**alue) en caso de ser necesario. Los programas auxiliares textbfbon_csv2html y **bon_csv2txt** pueden usarse para convertir a formatos de salida html y texto plano.

Le recomendamos leer la página **man** de **bonnie++** antes de usarlo, ya que tiene bastantes opciones acerca de qué tests realizar y cuan exhaustivos y estresantes pueden ser. Una sinopsis rápida se obtiene con:

```
$ bonnie++ -help

 bonnie++: invalid option -- 'h'
usage:
bonnie++ [-d scratch-dir] [-c concurrency] [-s size(MiB)[:chunk-size(b)]]
      [-n number-to-stat[:max-size[:min-size][:num-directories[:chunk-size]]]]
      [-m machine-name] [-r ram-size-in-MiB]
      [-x number-of-tests] [-u uid-to-use:gid-to-use] [-g gid-to-use]
      [-q] [-f] [-b] [-p processes | -y] [-z seed | -Z random-file]
      [-D]

Version: 1.96
```

Un test rápido puede obtenerse con un comando como el siguiente:

```
$  time sudo bonnie++ -n 0 -u 0 -r 100 -f -b -d /mnt
```

donde cada parámetro y valor significa:

- `-n 0` no realizar las pruebas de creación de archivos.

- `-u 0` ejecútelo como root.
- `-r 100` finja que tiene 100 MB de RAM.
- `-f` saltar las pruebas de caracter de E/S.
- `-b` hacer **fsync** después de cada escritura, lo cual fuerza a escribir a disco en vez de al caché.
- `-d /mnt` especifica el directorio que contendrá temporalmente los archivos creados; asegúrese que tiene suficiente espacio disponible, en este caso 300 MB.

Si no suministra una cifra para el tamaño de memoria, el programa averiguará cuánta memoria tiene el sistema y creará un archivo de prueba de 2 a 3 veces ese tamaño. No haremos eso aquí debido a que tomaría mucho tiempo para probar el programa.

En un sistema **RHEL 7**:

```
$ time sudo bonnie++ -n 0 -u 0 -r 100 -f -b -d /mnt

Using uid:0, gid:0.
Writing intelligently...done
Rewriting...done
Reading intelligently...done
start 'em...done...done...done...done...done...
Version  1.96       ------Sequential Output------ --Sequential Input- --Random-
Concurrency   1     -Per Chr- --Block-- -Rewrite- -Per Chr- --Block-- --Seeks--
Machine        Size K/sec %CP K/sec %CP K/sec %CP K/sec %CP K/sec %CP  /sec %CP
q7             300M           99769  14 106000  12           +++++ +++ 257.3   1
Latency                         226us     237us               418us     624ms

1.96,1.96,q7,1,1415992158,300M,,,,99769,14,106000,12,,,+++++,+++,257.3,1,,,,,,,,,,,,,,,,,,,226us,237us,,418us,624ms,,,,,,
```

En un sistema **Ubuntu 14.04**, corriendo una máquina virtual bajo un hipervisor en la misma máquina física:

```
$ time sudo bonnie++ -n 0 -u 0 -r 100 -f -b -d /mnt

Using uid:0, gid:0.
Writing intelligently...done
Rewriting...done
Reading intelligently...done
start 'em...done...done...done...done...done...
Version  1.97       ------Sequential Output------ --Sequential Input- --Random-
Concurrency   1     -Per Chr- --Block-- -Rewrite- -Per Chr- --Block-- --Seeks--
Machine        Size K/sec %CP K/sec %CP K/sec %CP K/sec %CP K/sec %CP  /sec %CP
ubuntu         300M           70000  61 43274  31           470061  96  2554  91
Latency                         306ms     201ms               9276us     770ms

1.97,1.97,ubuntu,1,1415983257,300M,,,,70000,61,43274,31,,,470061,96,2554,91,,,,,,,,,,,,,,,,,,,306ms,201ms,,9276us,770ms,,,
```

Puede apreciar claramente la disminución del rendimiento.

Asumiendo que ha guardado las salidas previas en un archivo llamado bonnie++.out, puede convertir la salida a html de la siguiente forma:

```
$ bon_csv2html < bonnie++.out > bonnie++.html
```

o a texto plano con:

```
$ bon_csv2txt < bonnie++.out > bonnie++.txt
```

Luego de leer la documentación, ejecute pruebas más largas y ambiciosas. Intente alguna de las pruebas que no realizamos. Si su sistema se comporta bien, guarde los resultados para realizar comparaciones de benchmarking en el futuro, en caso de que su sistema manifieste algún problema de rendimiento.

Lab 24.2: fs_mark

El programa **fs_mark** sirve para realizar benchmarking de bajo nivel en sistemas de archivos, utilizando operaciones asincrónicas de E/S sobre múltiples directorios y unidades, las cuales generan una carga. Es programa escrito por Ric Wheeler es bastante antiguo y ha resistido el paso del tiempo.

Puede descargarse desde http://sourceforge.net/projects/fsmark/. Una vez que hay obtenido el archivo, puede desempaquetarlo y compilarlo de la siguiente forma:

```
$ tar zxvf fs_mark-3.3.tgz
$ cd fs_mark
$ make
```

Lea el archivo `README` para obtener más información, ya que vamos a realizar lo básico.

Si la compilación falla con un error como este:

```
$ make

....
/usr/bin/ld: cannot find -lc
```

es porque no está instalada la versión **estática** de **glibc**. En sistemas basados en **Red Hat** lo puede hacer como sigue:

```
$ sudo yum install glibc-static
```

y en sistemas basados en **SUSE** con:

```
$ sudo zypper install glibc-devel-static
```

En sistemas basados en **Debian** la biblioteca estática está instalada junto a la compartida, por lo cual no es necesario instalar algún paquete adicional.

A modo de prueba, vamos a crear 1000 archivos, cada uno de 10 KB, luego de lo cual ejecutaremos **fsync** para escribir los cambios al disco. Esto puede hacerse en el directorio /tmp con el comando:

```
$ fs_mark -d /tmp -n 1000 -s 10240
```

Mientras esto está ejecutándose, obtenga estadísticas extendidas en otra terminal con **iostat**:

```
$ iostat -x -d /dev/sda 2 20
```

Las estadísticas en las que debería fijarse son los números de archivos por segundo reportados por **fs_mark** y el porcentaje de tiempo de CPU utilizado por **iostat**. Si este se aproxima a 100 por ciento, el sistema de E/S está estresado.

Dependiendo del tipo de sistema de archivos que está usando, podría mejorar los resultados cambiando las opciones de **mount**. Por ejemplo, para **ext3** o **ext4** puede probar lo siguiente:

```
$ mount -o remount,barrier=1 /tmp
```

para **ext4** puede intentar:

```
$ mount -o remount,journal_async_commit /tmp
```

Vea cómo cambian los resultados.

Tenga en cuenta que estas opciones podrían causar problemas si usted tiene un corte de energía eléctrica o cualquier otro apagado inesperado; es decir, es probable que exista una relación inversa entre estabilidad y velocidad.

La documentación acerca de algunas opciones de **mount** se puede encontrar junto a las fuentes del kernel en Documentation/filesystems y en la página **man** de **mount**.

Capitulo 25

Planificación de E/S

Lab 25.1: Comparación de planificadores de E/S

A continuación proveemos un script que se usa para comparar planificadores de E/S:

```
#!/bin/bash

#/*
# * The code herein is: Copyright the Linux Foundation, 2014
# * Author J. Cooperstein
# *
# * This Copyright is retained for the purpose of protecting free
# * redistribution of source.
# *
# * This code is distributed under Version 2 of the GNU General Public
# * License, which you should have received with the source.
# *
# */

NMAX=8
NMEGS=100
[[ -n $1 ]] && NMAX=$1
[[ -n $2 ]] && NMEGS=$2

echo Doing: $NMAX parallel read/writes on: $NMEGS MB size files

TIMEFORMAT="%R   %U   %S"

################################################################
# simple test of parallel reads
do_read_test(){
    for n in $(seq 1 $NMAX) ; do
        cat file$n > /dev/null &
    done
```

```
# wait for previous jobs to finish
    wait
}

# simple test of parallel writes
do_write_test(){
    for n in $(seq 1 $NMAX) ; do
        [[ -f fileout$n ]] && rm -f fileout$n
        (cp file1 fileout$n && sync) &
    done
# wait for previous jobs to finish
    wait
}

# create some files for reading, ok if they are the same
create_input_files(){
    [[ -f file1 ]] || dd if=/dev/urandom of=file1 bs=1M count=$NMEGS
    for n in $(seq 1 $NMAX) ; do
        [[ -f file$n ]] || cp file1 file$n
    done
}

echo -e "\ncreating as needed random input files"
create_input_files

################################################################
# begin the actual work

# do parallel read test
echo -e "\ndoing timings of parallel reads\n"
echo -e " REAL    USER    SYS\n"
for iosched in noop deadline cfq ; do
    echo testing IOSCHED = $iosched
    echo $iosched > /sys/block/sda/queue/scheduler
    cat /sys/block/sda/queue/scheduler
#    echo -e "\nclearing the memory caches\n"
    echo 3 > /proc/sys/vm/drop_caches
    time do_read_test
done
################################################################
# do parallel write test
echo -e "\ndoing timings of parallel writes\n"
echo -e " REAL    USER    SYS\n"
for iosched in noop deadline cfq ; do
    echo testing IOSCHED = $iosched
    echo $iosched > /sys/block/sda/queue/scheduler
    cat /sys/block/sda/queue/scheduler
    time  do_write_test
done
################################################################
```

Si está tomando la versión autodidacta en línea de este curso, encontrará el código fuente disponible para su descarga en la pantalla **Lab**.

Recuerde darle permisos de ejecución con:

```
$ chmod +x ioscript.sh
```

Lo que viene a continuación es una explicación del funcionamiento del script y de cómo usarlo:

El script realiza lo siguiente:

- Cambiar entre los planificadores de E/S disponibles en un disco duro, mientras se hace un número configurable de

lecturas y escrituras en paralelo de archivos de tamaño ajustable.

- Pruebas de lectura y escritura en pasos por separado.
- Al llevar a cabo las pruebas de lectura, asegúrese de que está leyendo desde el disco y no desde las páginas en memoria caché. Puede forzar que el caché se escriba en disco ejecutando el comando:

  ```
  $ echo 3 > /proc/sys/vm/drop_caches
  ```

  antes de realizar el test de lectura. Puede hacer **cat** a /dev/null para evitar escribir en el disco.
- Asegúrese de que todas las lecturas están completas antes de obtener información del tiempo tomado; esto puede llevarse a cabo ejecutando el comando **wait** en una shell.
- Las pruebas de escritura se hacen simplemente copiando un archivo (el cual estará en el caché luego de la primera lectura) múltiples veces de forma simultánea. Para asegurarse que todas las operaciones de escritura se han completado antes de obtener información del tiempo tomado, puede ejecutar el comando **sync**.

El script proporcionado toma dos argumentos. El primero es el número simuntáneo de lecturas y escrituras a realizar. El segundo es el tamaño en MB de cada archivo.

Este script debe ejecutarse como root, dado a que obtiene valores desde los árboles de directorios /proc y /sys.

Compare los resultados obtenidos usando diferentes planificadores de E/S.

Para realizar una exploración adicional, podría intentar cambiar algunos de los parámetros ajustables y ver cómo varían los resultados.

Capitulo 26

Memoria: monitoreo y ajustes

Lab 26.1: Invocando el OOM Killer

Vea qué particiones y archivos de intercambio (swap) están presentes en su sistema en el archivo /proc/swaps.

Deshabilite todas las áreas de intercambio con el comando

```
$ sudo /sbin/swapoff -a
```

Asegúrese de habilitar las áreas de intercambio una vez que haya terminado, con

```
$ sudo/sbin/swapon -a
```

Ahora vamos a poner el sistema bajo una presión de memoria creciente. Una forma de hacerlo es con el programa **stress** que instalamos anteriormente, al ejecutarlo con los argumentos que se muestran a continuación:

```
$ stress -m 8 -t 10s
```

lo cual mantendría ocupados 2 GB por 10 segundos.

Usted debería ver al **OOM** (Out of Memory) lanzarse en picada a terminar procesos para mantener vivo al sistema. Puede ver que está sucediendo con **dmesg** o monitoreando /var/log/messages, o /var/log/syslog, o a través de alguna interfaz gráfica que exponga lo que está sucediendo en los registros del sistema.

¿Qué proceso se finaliza primero?

Capitulo 27

Sistemas de gestión de paquetes

Lab 27.1: Sistema de control de versiones con git

Es posible que su sistema ya tenga instalado **git**. El comando `which git` le mostrará si está presente en el sistema. Si no lo está, usted puede obtener las fuentes, compilarlas e instalarlas, pero suele ser mucho más fácil instalar los paquetes binarios precompilados; su instructor puede ayudarlo a identificar los paquetes necesarios en caso que no estén instalados, o que no puedan ser instalados con alguno de los comandos siguientes:

```
$ sudo yum install git*
$ sudo zypper install git*
$ sudo apt-get install git*
```

de acuerdo a su distribución en particilar.

Comencemos a ver cómo **funciona git** y cuán fácil es usarlo. Por el momento nos limitaremos a crear nuestro proyecto local propio.

1. Primero crearemos un directorio de trabajo y luego inicializaremos **git** para que trabaje con él:

   ```
   $ mkdir git-test
   $ cd git-test
   $ git init
   ```

2. La inicialización del proyecto crear un directorio `.git`, el cual contenerá toda la información del control de versiones; los directorios principales que se incluyen en el proyecto permanecerán intactos. El contenido inicial del directorio luce así:

   ```
   $ ls -l .git
   total 40
   drwxrwxr-x 7 coop coop 4096 Dec 30 13:59 ./
   drwxrwxr-x 3 coop coop 4096 Dec 30 13:59 ../
   drwxrwxr-x 2 coop coop 4096 Dec 30 13:59 branches/
   -rw-rw-r-- 1 coop coop   92 Dec 30 13:59 config
   ```

```
-rw-rw-r-- 1 coop coop   58 Dec 30 13:59 description
-rw-rw-r-- 1 coop coop   23 Dec 30 13:59 HEAD
drwxrwxr-x 2 coop coop 4096 Dec 30 13:59 hooks/
drwxrwxr-x 2 coop coop 4096 Dec 30 13:59 info/
drwxrwxr-x 4 coop coop 4096 Dec 30 13:59 objects/
drwxrwxr-x 4 coop coop 4096 Dec 30 13:59 refs/
```

Más adelante describiremos el contenido de este directorio y sus subdirectorios, los que en su mayor parte comienzan vacíos.

3. A continuación crearemos un archivo y lo agregaremos al proyecto:

```
$ echo some junk > somejunkfile
$ git add somejunkfile
```

4. Podemos ver el estado actual de nuestro proyecto con:

```
$ git status

# On branch master
#
# Initial commit
#
# Changes to be committed:
#   (use "git rm --cached <file>..." to unstage)
#
#       new file: somejunkfile
#
```

Note que la salida anterior está mostrando que el archivo está **preparado** pero todavía no ha sido **registrado**.

5. Ahora modifiquemos el archivo y luego veamos la historia de las diferencias:

```
$ echo another line >> somejunkfile
$ git diff
diff --git a/somejunkfile b/somejunkfile
index 9638122..6023331 100644
--- a/somejunkfile
+++ b/somejunkfile
@@ -1 +1,2 @@
 some junk
+another line
```

6. Para registrar los cambios en el repositorio hacemos lo siguiente:

```
$  git commit -m "My initial commit" --author="A Genius <a_genius@linux.com>"
Created initial commit eafad66: My initial commit
 1 files changed, 1 insertions(+), 0 deletions(-)
 create mode 100644 somejunkfile
```

La opción `--author` es opcional. Si usted no especifica un mensaje de identificación con la opción `-m` al realizar el commit, usted será llevado a un editor para que ingrese el contenido correspondiente. Usted **debe** hacer esto o el commit será rechazado. Se elejirá el editor que esté configurado en la variable de ambiente `EDITOR` , la cual puede ser reemplazada con la configuración de `GIT_EDITOR`.

7. Puede ser tedioso agregar la información del autor cada vez que haga un commit. Es posible hacerlo de forma automática con:

```
$ git config user.name "Another Genius"
$ git config user.email "b_genius@linux.com"
```

lo cual se utilizará en el próximo commit.

8. Es posible ver la historia con:


```
$ git log

commit eafad66304ebbcd6acfe69843d246de3d8f6b9cc
Author: A Genius <a_genius@linux.com>
Date:   Wed Dec 30 11:07:19 2009 -0600

    My initial commit
```

y puede ver la información que hay ahí. Notará que el string largo haxadecimal es un identificador único de 160 bit y 40 dígitos, el cual corresponde al **número de commit**. **Git** trabaja con estos identificadores y no con los nombres de archivos.

9. Ahora usted es libre de modificar el archivo existente y agregar archivos nuevos con `git add`. Pero ellos estarán en estado staged (modificados y preparados para ser enviados al repositorio) hasta que usted haga otro `git commit`.

10. Hasta aquí vamos bien, aunque hemos visto solo la superficie de git.

Capitulo 28

RPM

Lab 28.1: Uso de RPM

Vamos a hacer algunas operaciones simples de consulta y verificación de paquetes **rpm**.

Este laboratorio funcionará igualmente bien tanto en sistemas basados en **Red Hat** como **SUSE**.

1. Encuentre a qué paquete pertenece el archivo /etc/logrotate.conf.
2. Liste la información acerca del paquete, incluyendo todos los archivos que contiene.
3. Verifique la instalación del paquete.
4. Intente desinstalar el paquete.

Solution 28.1

1.
```
$ rpm -qf /etc/logrotate.conf
logrotate-3.8.6-4.el7.x86_64
```

2.
```
$ rpm -qil logrotate
...
```

   Lo mismo podría hacerse de una forma más elegante, que combina estos dos pasos:

```
$ rpm -qil $(rpm -qf /etc/logrotate.conf)
```

3.
```
$ rpm -V logrotate
..?......    /etc/cron.daily/logrotate
S.5....T.  c /etc/logrotate.conf
```

4. En **RHEL 7**:

```
$ sudo rpm -e logrotrate
error: Failed dependencies:
        logrotate is needed by (installed) vsftpd-3.0.2-9.el7.x86_64
        logrotate >= 3.5.2 is needed by (installed) rsyslog-7.4.7-7.el7_0.x86_64
```

En **openSUSE 13.1**:

```
$ sudo  rpm -e logrotate
error: Failed dependencies:
        logrotate is needed by (installed) xdm-1.1.10-24.2.1.x86_64
        logrotate is needed by (installed) syslog-service-2.0-772.1.2.noarch
        logrotate is needed by (installed) wpa_supplicant-2.0-3.4.1.x86_64
        logrotate is needed by (installed) mcelog-1.0pre3.6e4e2a000124-19.4.1.x86_64
        logrotate is needed by (installed) apache2-2.4.6-6.27.1.x86_64
        logrotate is needed by (installed) net-snmp-5.7.2-9.8.1.x86_64
        logrotate is needed by (installed) kdm-4.11.12-119.1.x86_64
```

Tenga en cuenta que el árbol exacto de dependencias de paquetes está en función tanto de la distribución como del software instalado.

Lab 28.2: Reconstrucción de la base de datos RPM

Existen condiciones bajo las cuales la base de datos RPM almacenada en /var/lib/rpm puede corromperse. En ese ejercicio construiremos una nueva y verificaremos su integridad.

Este laboratorio funcionará igualmente bien tanto en sistemas basados en **Red Hat** como **SUSE**.

1. Realice una copia de seguridad de /var/lib/rpm ya que el proceso de reconstrucción sobreescribirá el contenido. Si usted no lo hace y algo sale mal, podría estar en serios problemas.
2. Reconstruya la base de datos.
3. Compare el contenido nuevo del directorio con la copia de seguridad; no examine el contenido de los archivos, ya que son datos binarios, sino más bien el número de archivos y los nombres.
4. Obtenga una lista de todos los **rpms** en el sistema. Sería interesante que tome una lista antes del proceso de reconstrucción y la compare con la obtenida después del mismo. Si el comando de consulta funciona, la base de datos nueva debería estar bien.
5. Compare de nuevo los contenidos de los dos directorios. ¿Tienen los mismos archivos ahora?
6. Usted podría borrar la copia de seguridad (de unos 100 MB de tamaño probablemente), pero podría ser buena idea mantenerla por un tiempo mientras se asegura de que el sistema se está comportando adecuadamente.

Eche un vistazo a http://www.rpm.org/wiki/Docs/RpmRecovery para examinar con más detalle los pasos para verificar y/o recuperar la integridad de la base de datos.

Solution 28.2

1.
```
$ cd /var/lib
$ sudo cp -a rpm rpm_BACKUP
```

2.
```
$ sudo rpm --rebuilddb
```

3.
```
$ ls -l rpm rpm_BACKUP
```


4. `$ rpm -qa | tee /tmp/rpm-qa.output`

5. `$ ls -l rpm rpm_BACKUP`

6. ¡Realice este paso una vez que esté seguro de que el sistema está funcionando correctamente!

   `$ sudo rm -rf rpm_BACKUP`

Capitulo 29

DPKG

Lab 29.1: Uso de dpkg

Haremos algunas operaciones simples de consulta y verificación de paquetes de **Debian**.

1. Encuentre a qué paquete pertenece el archivo /etc/logrotate.conf.
2. Liste la información acerca del paquete, incluyendo todos los archivos que contiene.
3. Verifique la instalación del paquete.
4. Intente desinstalar el paquete.

Solution 29.1

1.
```
$ dpkg -S /etc/logrotate.conf
logrotate: /etc/logrotate.conf
```

2.
```
$ dpkg -L logrotate
...
```

3.
```
$ dpkg -V logrotate
```

4.
```
$ sudo dpkg -r logrotate
dpkg: dependency problems prevent removal of logrotate:
 libvirt-bin depends on logrotate.
 ubuntu-standard depends on logrotate.

dpkg: error processing package logrotate (--remove):
```

```
 dependency problems - not removing
Errors were encountered while processing:
 logrotate
```


Capitulo 30

yum

Lab 30.1: Comandos básicos de YUM

1. Verifique si hay actualizaciones disponibles para su sistema.
2. Actualice un paquete en particular.
3. Liste todos los paquetes instalados que tienen relación con el kernel. También liste todos los instalados o disponibles, relacionados al kernel.
4. Instale el paquete **httpd-devel** o cualquier otro que no esté instalado. Ejecute el siguiente comando simple:

   ```
   $ sudo yum list
   ```

   el cual le mostrará una lista completa; puede proveer un comodín como argumento para reducir la lista.

Solution 30.1

1. ```
   $ sudo yum update
   $ sudo yum check-update
   $ sudo yum list updates
   ```

   Solo la primera forma intentará realizar las instalaciones.

2. ```
   $ sudo yum update bash
   ```

3. ```
   $ sudo yum list installed "kernel*"
   $ sudo yum list "kernel*"
   ```

4. ```
   $ sudo yum install httpd-devel
   ```

Lab 30.2: Uso de yum para encontrar información acerca de un paquete

Usando **yum** (y no **rpm** directamente), encuentre:

1. Todos los paquetes que contienen una referencia a **bash** en su nombre o descripción.
2. Los paquetes **bash** instalados y disponibles.
3. La información del paquete **bash**.
4. Las dependencias del paquete **bash**.

Ejecute los comandos del ejercicio tanto como `root` y usuario normal. ¿Nota alguna diferencia?

Solution 30.2

Nota: en **RHEL 7** podría recibir algunos errores de permisos si no usa **sudo** con los siguientes comandos, a pesar de que estamos obteniendo información solamente.

1. `$ sudo yum search bash`
2. `$ sudo yum list bash`
3. `$ sudo yum info bash`
4. `$ sudo yum deplist bash`

Todos los comandos de arriba deberían funcionar tanto para usuarios normales como para `root`.

Lab 30.3: Manejo de grupos de paquetes con yum

Nota: En **RHEL 7** podría recibir errores de permisos si no usa **sudo** con algunos de los siguientes comandos, aun cuando estemos obteniendo información solamente.

yum provee la habilidad de manejar grupos de paquetes.

1. Use el siguiente comando para listar todos los grupos de paquetes disponibles en el sistema:

   ```
   $ yum grouplist
   ```

2. Identifique el grupo `Backup Client` y genere información acerca de él usando el comando:

   ```
   $ yum groupinfo "Backup Client"
   ```

3. Instálelo usando:

   ```
   $ sudo yum groupinstall "Backup Client"
   ```

4. Identifique un grupo de paquetes que esté instalado actualmente en el sistema y que usted no lo necesite. Desinstálelo usando `yum groupremove`, como se muestra a continuación:

   ```
   $ sudo yum groupremove "Backup Client"
   ```

   Tenga en cuenta que se le pedirá confirmar la desinstalación, por lo que puede ejecutar el comando de forma segura y ver cómo funciona.

   Puede notar que `groupremove` **no** desinstala todo lo que estaba instalado; si se trata de un bug o una característica queda para una discusión.

Lab 30.4: Agregar un repositorio yum

De acuerdo a sus autores (en http://www.webmin.com/index.htm):

"**Webmin** es una interfaz web para administración de sistemas Unix. Usando cualquier navegador web moderno, es posible configurar cuentas de usuario, Apache, DNS, compartir archivos y mucho más. Webmin elimina la necesidad de editar manualmente archivos de configuración como /etc/passwd, y le permite gestionar un sistema desde la consola o remotamente."

Crearemos un repositorio para la instalación y actualización. Si bien es cierto que podríamos ir a la página de descargas y obtener el **rpm** actual, esto no nos proporcionaría ninguna actualización automática.

1. Cree un archivo de repositorio llamado webmin.repo en el directorio /etc/yum.repos.d. Debería contener lo siguiente:

   ```
   [Webmin]
   name=Webmin Distribution Neutral
   baseurl=http://download.webmin.com/download/yum
   mirrorlist=http://download.webmin.com/download/yum/mirrorlist
   enabled=1
   gpgcheck=0
   ```

   (Note que también puede copiar y pegar el contenido desde http://www.webmin.com/download.html.)

2. Instale el paquete webmin.

   ```
   $ sudo yum install webmin
   ```


Capitulo 31

zypper

Lab 31.1: Comandos básicos de zypper

1. Verifique si hay actualizaciones disponibles para su sistema.
2. Actualice un paquete en particular.
3. Liste todos los repositorios conocidos por el sistema, tanto si están habilitados o no.
4. Liste todos los paquetes relacionados al kernel, también liste todos los instalados o disponibles.
5. Instale el paquete **apache2-devel** o cualquier otro que no lo haya instalado aún (tenga en cuenta que **httpd** es **apache2** en sistemas **SUSE**). Haciendo un simple:

   ```
   $ sudo zypper search
   ```

   le mostrará una una lista completa; puede pasarle un comodín como argumento para reducir la lista.

Solution 31.1

1. `$ zypper list-updates`
2. `$ sudo zypper update bash`
3. `$ zypper repos`
4. `$ zypper search -i kernel`
   `$ zypper search kernel`
5. `$ sudo zypper install apache2-devel`

Lab 31.2: Usando zypper para encontrar información acerca de un paquete

Usando **zypper** (y no **rpm** directamente), encuentre:

1. Todos los paquetes que contienen una referencia a **bash** en su nombre o descripción.
2. Todos los paquetes **bash** instalados y disponibles.
3. La información del paquete **bash**.
4. Las dependencias del paquete **bash**.

Ejecute los comandos de arriba tanto como `root` y como usuario normal. ¿Nota alguna diferencia?

Solution 31.2

1. `$ zypper search -d bash`

   Sin la opción `-d` se muestran solamente los paquetes con `bash` en el nombre del archivo. Tendría que hacer `zypper info` en el paquete para ver donde se menciona **bash**.

2. `$ zypper search bash`

3. `$ zypper info  bash`

4. `$ zypper info--requires  bash`

   le dará una lista de archivos requeridos por **bash**. Tal vez la forma más fácil es ver las dependencias de **bash** cuando está instalado, haciendo

   ```
   $ sudo zypper remove --dry-run bash
   ```

   Para este ejercicio **bash** es una mala elección ya que es una parte esencial del sistema. De todas formas no podrá desinstalarlo.

Capitulo 32

APT

Lab 32.1: Comandos básicos APT

1. Verifique si hay actualizaciones disponibles para su sistema.
2. Actualice un paquete en particular.
3. Liste todos los paquetes relacionados al kernel, también liste todos los instalados o disponibles.
4. Instale el paquete **apache2-devel** o cualquier otro que no lo haya instalado aún. Haciendo un simple:

   ```
   $ apt-cache pkgnames
   ```

   le mostrará una una lista completa; puede pasarle un comodín como argumento para reducir la lista.

Solution 32.1

1. Primero sincronice los archivos de índice de paquetes con los repositorios remotos:

   ```
   $ sudo apt-get update
   ```

   Para actualizar realmente:

   ```
   $ sudo apt-get upgrade
   $ sudo apt-get -u upgrade
   ```

   También puede usar `dist-upgrade`, como se discutió anteriormente. Solo la primera forma tratará de realizar las instalaciones.

2. `$ sudo apt-get upgrade bash`

3. `$ apt-cache search "kernel"`
   `$ apt-cache search -n "kernel"`
   `$ apt-cache pkgnames "kernel"`

   La segunda y tercera formas solo buscarán paquetes que tengan `kernel` en el nombre.

   `$ dpkg --get-selections "*kernel*"`

   para obtener los paquetes instalados solamente. Tenga en cuenta que en los sistemas basados en **Debian** tendrá que usar `linux` en vez `kernel` para los paquetes relacionados al kernel, ya que usualmente no tienen `kernel` en el nombre.

4. `$ sudo apt-get install apache2-dev`

Lab 32.2: Usando APT para encontrar información acerca de un paquete

Usando **apt-cache** y **apt-get** (y no **dpkg**), encuentre:

1. Todos los paquetes que contienen una referencia a **bash** en su nombre o descripción.
2. Todos los paquetes **bash** instalados y disponibles.
3. La información del paquete **bash**.
4. Las dependencias del paquete **bash**.

Ejecute los comandos de arriba tanto como `root` y como usuario normal. ¿Nota alguna diferencia?

Solution 32.2

1. `$ apt-cache search bash`
2. `$ apt-cache search -n bash`
3. `$ apt-cache show bash`
4. `$ apt-cache depends bash`
   `$ apt-cache rdepends bash`

Lab 32.3: Gestionando grupos de paquetes con APT

APT provee la habilidad de gestionar grupos de paquetes, de forma similar a como lo hace **yum** a través del uso de los **metapaquetes**. Estos pueden ser vistos como **paquetes virtuales**, los que reúnen los paquetes relacionados entre sí que deben ser instalados y desinstalados como grupo.

Para obtener una lista de los **metapaquetes** disponibles:

```
$ apt-cache search metapackage

bacula - network backup service - metapackage
bacula-client - network backup service - client metapackage
bacula-server - network backup service - server metapackage
cloud-utils - metapackage for installation of upstream cloud-utils source
compiz - OpenGL window and compositing manager
emacs - GNU Emacs editor (metapackage)
....
```


Usted puede instalarlos fácilmente como cualquier paquete individual, tal como se muestra aquí:

```
$ sudo apt-get install bacula-client

Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  bacula-common bacula-console bacula-fd bacula-traymonitor
Suggested packages:
  bacula-doc kde gnome-desktop-environment
The following NEW packages will be installed:
  bacula-client bacula-common bacula-console bacula-fd bacula-traymonitor
0 upgraded, 5 newly installed, 0 to remove and 0 not upgraded.
Need to get 742 kB of archives.
After this operation, 1,965 kB of additional disk space will be used.
Do you want to continue? [Y/n]
```

Seleccione un metapaquete desinstalado y remuévalo.

Capitulo 33

Gestión de cuentas de usuario

Lab 33.1: Trabajando con cuentas de usuario

1. Examine /etc/passwd y /etc/shadow, comparando los campos en cada archivo, especialmente para las cuentas de usuario normales. ¿Cuáles son iguales y cuáles diferentes?
2. Usando **useradd** cree una cuenta llamada `user1`.
3. Usando `ssh` conéctese como `user1`. Puede hacerlo así:

   ```
   $ ssh user1@localhost
   ```

   Debería fallar porque se necesita una contraseña para `user1`; no ha sido configurada todavía.
4. Configure la contraseña de `user1` como `user1pw` e intente conectarse de nuevo como `user1`.
5. Revise los registros nuevos que fueron creados en los archivos /etc/passwd, /etc/group y /etc/shadow.
6. Revise el archivo /etc/default/useradd y vea cuáles son los valores por defecto actuales. También eche un vistazo al archivo /etc/login.defs.
7. Cree una cuenta de usuario llamada `user2` que use la shell **Korn** (**ksh**) por defecto (si no tiene **/bin/ksh** en el sistema, instálela o use la shell **C** ubicada en **/bin/csh**). Configure la contraseña en `user2pw`.
8. Eche un vistazo a /etc/shadow. ¿Cuál es la fecha de expiración para la cuenta `user1`?
9. Use **chage** para configurar la fecha de expiración del usuario `user1` a Diciembre 1, 2013.

   Eche un vistazo a /etc/shadow para ver cuál es la nueva fecha de expiración.
10. Use **usermod** para bloquear la cuenta `user1`.

    Eche un vistazo a /etc/shadow para ver qué ha cambiado en relación a la contraseña del usuario `user1`. Restablezca la contraseña de la cuenta a `userp1` para completar este ejercicio.

Solution 33.1

1. `$ sudo grep student /etc/passwd /etc/shadow`

   ```
   /etc/passwd:student:x:1000:100:LF Student:/home/student:/bin/bash
   /etc/shadow:student:$6$jtoFVPICHhba$iGFFUU8ctrtOGoistJ4/30DrNLi1FS66qnn0VbS6Mvm
     luKI08SgbzT5.IcOHo5j/SOdCagZmF2RgzTvzLb11H0:16028:0:99999:7:::
   ```

   Puede usar cualquier nombre de usuario normal en vez de `student`. Lo único que coincide es el campo de nombre de usuario.

2. `$ sudo useradd user1`

3. `$ ssh user1@localhost`

   ```
   user1@localhost's password:
   ```

   Tenga en cuenta que quizás tenga que iniciar primero el servicio **sshd** de la siguiente forma:

   ```
   $ sudo service sshd restart
   ```

   o

   ```
   $ sudo systemctl restart sshd.service
   ```

4. `$ sudo passwd user1`

   ```
   Changing password for user user1.
   New password:
   ```

5. `$ sudo grep user1 /etc/passwd /etc/shadow`

   ```
   /etc/passwd:user1:x:1001:100::/home/user1:/bin/bash
   /etc/shadow:user1:$6$OBE1mPMw$CIc7urbQ9ZSnyiniVOeJxKqLFu8fz4whfEexVem2
      TFpucuwRN1CCHZ19XGhj4qVujslRIS.P4aCXd/y1U4utv.:16372:0:99999:7:::
   ```

6. Ya sea en sistemas **RHEL 7** o **openSUSE 13.1**, por ejemplo:

   ```
   $ cat /etc/default/useradd
   # useradd defaults file
   GROUP=100
   HOME=/home
   INACTIVE=-1
   EXPIRE=
   SHELL=/bin/bash
   SKEL=/etc/skel
   CREATE_MAIL_SPOOL=yes
   $ cat /etc/login.defs
   ....
   ```

   No vamos a pegar el contenido del segundo archivo porque es largo, pero examínelo en su sistema.

7. ```
   $ sudo useradd -s /bin/ksh user2
   $ sudo passwd user2
   Changing password for user user2.
   New password:
   ```

8. `$ sudo grep user1 /etc/shadow`

   ```
   user1:$6$OBE1mPMw$CIc7urbQ9ZSnyiniVOeJxKqLFu8fz4whfEexVem2TFpucuwRN1CCHZ
      19XGhj4qVujslRIS.P4aCXd/y1U4utv.:16372:0:99999:7:::
   ```

   No debería haber ninguna fecha de expiración.


   ```

9. 
```
$ sudo chage -E 2013-12-1 user1
$ sudo sudo grep user1 /etc/shadow
```
```
user1:6OBE1mPMw$CIc7urbQ9ZSnyiniVOeJxKqLFu8fz4whfEexVem2TFpucuwRN1CCHZ
19XGhj4qVujslRIS.P4aCXd/y1U4utv.:16372:0:99999:7::16040:
```

10.
```
$ sudo usermod -L user1
```

```
$ sudo passwd user1
```


Capitulo 34

Gestión de grupos

Lab 34.1: Trabajando con grupos

1. Cree dos cuentas de usuario (`rocky` y `bullwinkle`) y asegúrese de que tengan directorios home.
2. Cree dos grupos, `friends` y `bosses` (con GID 490). Eche un vistazo al archivo /etc/group. Vea qué GID se le asignó a cada grupo.
3. Agregue `rocky` a ambos grupos.

   Agregue `bullwinkle` al grupo `friends`.

   Eche un vistazo al archivo /etc/group para ver cómo cambió.
4. Conéctese como `rocky`. Cree un directorio llamado `somedir` y configure el propietario del grupo en `bosses` (use **chgroup**, el cual se discutirá en la próxima sesión).

   Nota: probablemente necesitará agregar permisos de ejecución para todos en el directorio home de `rocky`.
5. Conéctese como `bullwinkle` e intente crear un archivo en `/home/rocky/somedir` llamado `somefile`, usando el comando **touch**.

   ¿Puede hacer esto? No, debido al propietario del grupo y los permisos `chmod a+x` en el directorio.
6. Agregue `bullwinkle` al grupo bosses e intente de nuevo. Note que tendrá que desconectarse y conectarse de nuevo para que la participación en el grupo sea efectiva. Haga lo siguiente:

Solution 34.1

1. ```
   $ sudo useradd -m rocky
   $ sudo useradd -m bullwinkle
   $ sudo passwd rocky
   ```
   ```

```
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
```

```
$ sudo passwd bullwinkle
```

```
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
```

```
$ ls -l /home
```

```
total 12
drwxr-xr-x  2 bullwinkle bullwinkle 4096 Oct 30 09:39 bullwinkle
drwxr-xr-x  2 rocky      rocky      4096 Oct 30 09:39 rocky
drwxr-xr-x 20 student    student    4096 Oct 30 09:18 student
```

2.
```
$ sudo groupadd friends
$ sudo groupadd -g 490 bosses
$ grep -e friends -e bosses /etc/group
```

```
friends:x:1003:
bosses:x:490:
```

3.
```
$ sudo usermod -G friends,bosses rocky
$ sudo usermod -G friends bullwinkle

$ grep -e rocky -e bullwinkle /etc/group
```

```
rocky:x:1001:
bullwinkle:x:1002:
friends:x:1003:rocky,bullwinkle
bosses:x:490:rocky
```

```
$ groups rocky bullwinkle
```

```
rocky : rocky friends bosses
bullwinkle : bullwinkle friends
```

4.
```
$ ssh rocky@localhost
$ cd ~
$ mkdir somedir
$ chgrp bosses somedir
$ ls -l
```

```
total 16
-rw-r--r-- 1 rocky rocky  8980 Oct  4  2013 examples.desktop
drwxrwxr-x 2 rocky bosses 4096 Oct 30 09:53 somedir
```

```
$ chmod a+x .
```

5.
```
$ ssh bullwinkle@localhost
$ touch /home/rocky/somedir/somefile
```

```
touch: cannot touch /home/rocky/somedir/somefile: Permission denied
```

```
$ exit
```

6.
```
$ sudo usermod -a -G bosses bullwinkle
$ ssh bullwinkle@localhost
$ touch /home/rocky/somedir/somefile
$ ls -al /home/rocky/somedir
```

(fíjese en quién es el propietario de los archivos)

Capitulo 35

Permisos de archivos y propietarios

Lab 35.1: Usando chmod

Es posible usar ya sea el método de dígitos octales o simbólico para especificar los permisos al usar **chmod**. Elaboremos un poco más en el método simbólico.

Se puede otorgar permisos directamente, agregar o quitar permisos. La sintaxis es bastante obvia. Intente los siguientes ejemplos:

```
$ chmod u=r,g=w,o=x afile
$ chmod u=+w,g=-w,o=+rw afile
$ chmod ug=rwx,o=-rw afile
```

Después de cada paso haga:

```
$ ls -l afile
```

para ver cómo cambiaron los permisos. Intente algunas variaciones.

Lab 35.2: umask

Cree un archivo vacío con:

```
$ touch afile
$ ls -l afile
```

```
-rw-rw-r-- 1 coop coop 0 Jul 26 12:43 afile
```

lo que demuestra que por defecto se crea tanto con permisos de lectura y escritura para el propietario y el grupo, pero solo de lectura para el mundo.

En realidad, a nivel de sistema operativo los permisos por defecto que se dan al crear un archivo o directorio son lectura/escritura para el propietario, grupo **y** mundo (0666); los valores predeterminados han sido modificados por el actual **umask**.

Si ejecuta **umask** obtendrá su valor actual:

```
$ umask

0002
```

lo cual es el valor más convencional que los administradores de sistemas configuran para los usuarios. Este valor se combina con los permisos de creación de archivos para obtener el resultado actual. Es decir:

```
0666 & ~002 = 0664; i.e., rw-rw-r--
```

Intente modificar el **umask**, cree archivos y vea los permisos resultantes, como en:

```
$ umask 0022
$ touch afile2
$ umask 0666
$ touch afile3
$ ls -l afile*
```


Capitulo 36

Pluggable Authentication Modules (PAM)

Lab 36.1: Configuración de PAM

Una de las configuraciones más comunes de **PAM** es denegar el acceso luego de un cierto número de intentos fallidos. Esto se hace con el módulo `pam_tally2`. En este ejercicio vamos a denegar el acceso a través de **ssh** después de tres intentos fallidos de acceso.

1. Edite /etc/pam.d/sshd y configúrelo para denegar el acceso luego de tres intentos fallidos. Pista: agregue las siguientes dos líneas al archivo

   ```
   auth required pam_tally2.so deny=3 onerr=fail
   account required pam_tally2.so
   ```

2. Intente acceder tres veces con un usuario en particular (uno que tenga cuenta) y escriba mal la contraseña.
3. Intente acceder con el mismo usuario, pero esta vez con la contraseña correcta.
4. Verifique cuántos inicios de sesión fallidos registra el usuario.
5. Reinicie el contador de inicios fallidos.
6. Verifique nuevamente cuántos inicios de sesión fallidos registra el usuario.
7. Intente acceder de nuevo con la contraseña correcta.

Solution 36.1

1. Agregue las siguientes dos líneas a /etc/pam.d/sshd:

   ```
   auth required pam_tally2.so deny=3 onerr=fail
   account required pam_tally2.so
   ```

2. `$ ssh student@localhost`

```
Password:
Password:
Password:
Permission denied (publickey,keyboard-interactive).
```

3. `$ ssh student@localhost`

```
Password:
Account locked due to 3 failed logins
```

4. `$ sudo  pam_tally2`

```
Login           Failures Latest failure     From
student              3    11/01/14 20:41:12  localhost
```

5. `$ sudo pam_tally2 -u student -r`

```
Login           Failures Latest failure     From
student              3    11/01/14 20:41:12  localhost
```

6. `$ sudo pam_tally2 -u student -r`

```
Login           Failures Latest failure     From
student              0
```

7. `$ ssh student@localhost`

```
Password:
Last failed login: Sat Nov  1 20:41:14 CDT 2014 from localhost on ssh:notty
There were 6 failed login attempts since the last successful login.
Last login: Sat Nov  1 20:28:38 2014 from localhost
Have a lot of fun...
```


Capitulo 37

Métodos de respaldos y recuperación de la información

Lab 37.1: Usando tar en copias de seguridad

1. Cree un directorio llamado `backup` y ponga aquí un archivo **tar** comprimido de todos los archivos bajo /usr/include, en donde el nivel de directorio más alto sea `include`. Puede usar cualquier método de compresión, ya sea **gzip**, **bzip2** o **xzip**.
2. Liste los archivos del archivo comprimido.
3. Cree un directorio llamado `restore`, desempaque y descomprima el archivo aquí.
4. Compare el contenido con el directorio original desde el cual el archivo tar fue creado.

Solution 37.1

1. ```
   $ cd backup
   $ cd /usr ; tar zcvf include.tar.gz include
   ```

   o

   ```
   $ tar -C /usr -zcf include.tar.gz include
   $ tar -C /usr -jcf include.tar.bz2 include
   $ tar -C /usr -Jcf include.tar.xz include
   ```

   Note la eficacia de la compresión de cada uno de los tres métodos:

   ```
   $ du -sh /usr/include
   55M        /usr/include
   ```
   ```

2. `$ ls -lh include.tar.*`
```
-rw-rw-r-- 1 coop coop 5.3M Nov  3 14:44 include.tar.bz2
-rw-rw-r-- 1 coop coop 6.8M Nov  3 14:44 include.tar.gz
-rw-rw-r-- 1 coop coop 4.7M Nov  3 14:46 include.tar.xz
```

3. `$ tar tvf include.tar.xz`
```
qdrwxr-xr-x root/root         0 2014-10-29 07:04 include/
-rw-r--r-- root/root     42780 2014-08-26 12:24 include/unistd.h
-rw-r--r-- root/root       957 2014-08-26 12:24 include/re_comp.h
-rw-r--r-- root/root     22096 2014-08-26 12:24 include/regex.h
-rw-r--r-- root/root      7154 2014-08-26 12:25 include/link.h
.....
```

   Note que no es necesario proveer la opción `j`, `J` o `z` al descomprimir, ya que **tar** es lo suficientemente inteligente para determinar lo que se necesita hacer.

4. `$ cd .. ; mkdir restore ; cd restore`
   `$ tar xvf ../backup/include.tar.bz2`
```
include/
include/unistd.h
include/re_comp.h
include/regex.h
include/link
.....
$ diff -qr include /usr/include
```

Lab 37.2: Usando cpio en copias de seguridad

Vamos a realizar el mismo ejercicio ahora, pero usando **cpio** en vez de **tar**. Vamos a repetir las instrucciones con leves modificaciones para facilitar el uso.

1. Cree un directorio llamado `backup` y ponga aquí un archivo **cpio** comprimido de todos los archivos bajo /usr/include, en donde el nivel de directorio más alto sea `include`. Puede usar cualquier método de compresión, ya sea **gzip**, **bzip2** o **xzip**.
2. Liste los archivos del archivo comprimido.
3. Cree un directorio llamado `restore`, desempaque y descomprima el archivo aquí.
4. Compare el contenido con el directorio original desde el cual el archivo tar fue creado.

Solution 37.2

1. `$ (cd /usr ; find include | cpio -c -o > /home/student/backup/include.cpio)`
```
82318 blocks
```

   o para ponerlo de forma comprimida:

```
$ (cd /usr ; find include | cpio -c -o | gzip -c > /home/student/backup/include.cpio.gz)
82318 blocks
$ ls -lh include*
total 64M
-rw-rw-r-- 1 coop coop  41M Nov  3 15:26 include.cpio
-rw-rw-r-- 1 coop coop 6.7M Nov  3 15:28 include.cpio.gz
-rw-rw-r-- 1 coop coop 5.3M Nov  3 14:44 include.tar.bz2
-rw-rw-r-- 1 coop coop 6.8M Nov  3 14:44 include.tar.gz
-rw-rw-r-- 1 coop coop 4.7M Nov  3 14:46 include.tar.xz
```


2. `$ cpio -ivt < include.cpio`

```
drwxr-xr-x  86 root     root            0 Oct 29 07:04 include
-rw-r--r--   1 root     root        42780 Aug 26 12:24 include/unistd.h
-rw-r--r--   1 root     root          957 Aug 26 12:24 include/re_comp.h
-rw-r--r--   1 root     root        22096 Aug 26 12:24 include/regex.h
.....
```

   Note la redirección de la entrada; el archivo no es un argumento. También se podría hacer:

```
$ cd ../restore
$ cat ../backup/include.cpio | cpio -ivt
$ gunzip -c include.cpio.gz | cpio -ivt
```

3.
```
$ rm -rf include
$ cpio -id < ../backup/include.cpio
$ ls -lR include
```

   or

```
$ cpio -idv < ../backup/include.cpio

$ diff -qr include /usr/include
```

Lab 37.3: Usando rsync en copias de seguridad

1. Usando **rsync** vamos a crear de nuevo una copia completa de /usr/include en su directorio de respaldo:

```
$ rm -rf include
$ rsync -av /usr/include .
sending incremental file list
include/
include/FlexLexer.h
include/_G_config.h
include/a.out.h
include/aio.h
.....
```

2. Ejecutemos el comando una segunda vez y veamos si hace algo:

```
$ rsync -av /usr/include .
sending incremental file list

sent 127398 bytes  received 188 bytes  255172.00 bytes/sec
total size is 41239979  speedup is 323.23
```

3. Un asunto confuso de **rsync** es que usted podría haber esperado que el comando correcto fuera:

```
$ rsync -av /usr/include include
sending incremental file list
...
```

   Sin embargo, si hace esto se dará cuenta que en realidad se crea un nuevo directorio, include/include.

4. Para evitar los archivos adicionales puede usar la opción `--delete`:

```
$ rsync -av --delete /usr/include .
```



```
sending incremental file list
include/
deleting include/include/xen/privcmd.h
deleting include/include/xen/evtchn.h
....
deleting include/include/FlexLexer.h
deleting include/include/

sent 127401 bytes  received 191 bytes  85061.33 bytes/sec
total size is 41239979  speedup is 323.22
```

5. Para hacer otro ejercicio simple, remueva un árbol de subdirectorio en su copia de seguridad y luego ejecute **rsync** de nuevo con y sin la opción `--dry-run`:

```
$ rm -rf include/xen
$ rsync -av --delete --dry-run /usr/include .
sending incremental file list
include/
include/xen/
include/xen/evtchn.h
include/xen/privcmd.h

sent 127412 bytes  received 202 bytes  255228.00 bytes/sec
total size is 41239979  speedup is 323.16 (DRY RUN)
$ rsync -av --delete  /usr/include .
```

6. Un script simple con un buen conjunto de parámetros para usar **rsync**:

```
#!/bin/sh
set -x

rsync --progress -avrxH -e "ssh -c blowfish" --delete $*
```

   el cual funcionará en una máquina local como también en red. Tenga en cuenta la importancia de la opción `-x`, la cual impide que **rsync** cruce los límites del sistema de archivos.

Para mayor diversión, si tiene acceso a más de un computador, intente realizar estos pasos con una fuente y destino en distintas máquinas.

Capitulo 38

Direcciones de red

No hay ejercicios de laboratorio en este capítulo. Solamente se prepara el escenario para el capítulo siguiente sobre la configuración de red, el cual tiene varios laboratorios.

Capitulo 39

Configuración de dispositivos de red

Lab 39.1: Configuración estática de una interfaz de red

Nota: puede que tenga que usar una interfaz de red diferente a `eth0`. Usted puede hacer este ejercicio desde una interfaz gráfica, pero nosotros presentaremos una solución en la línea de comandos.

1. Muestre la dirección IP actual, ruta por defecto y la configuración del **DNS** para `eth0`. Tome una copia de los valores para reconfigurarlos más tarde.
2. Deshabilite `eth0` y reconfigúrela para usar una dirección estática en vez de **DCHP**, usando la información que registró en el punto anterior.
3. Habilite la interfaz y configure el cliente del servidor de nombres con la información que anotó previamente.

   Verifique el hostname del sistema y luego hágale **ping**.
4. Asegúrese que la configuración que realizó funciona después de reiniciar el sistema.

Restaure la configuración original una vez que haya terminado el ejercicio.

Solution 39.1

1. ```
   $ ifconfig eth0
   $ route -n
   $ cp /etc/resolv.conf resolv.conf.keep
   ```

2. ```
   $ sudo ifconfig eth0 down
   ```

   Asegúrese que realiza lo siguiente en /etc/sysconfig/network-scripts/ifcfg-eth0, en sistemas basados en **Red Hat**:

```
DEVICE=eth0
BOOTPROTO=static
ONBOOT=yes
IPADDR=noted from step 1
NETMASK=noted from step 1
GATEWAY=noted from step 1
```

En sistemas basados en **SUSE** edite el archivo /etc/sysconfig/network de la misma forma. En sistemas basados en **Debian** edite /etc/networking/interfaces e incluya:

```
iface eth0 inet static
     address noted from step 1
     netmask noted from step 1
     gateway noted from step 1
```

3. `$ sudo ifconfig eth0 up`

   ```
   $ sudo cp resolv.conf.keep /etc/resolv.conf
   $ cat /etc/sysconfig/network
   $ cat /etc/hosts
   $ ping yourhostname
   ```

4. ```
   $ sudo reboot
   $ ping hostname
   ```

Lab 39.2: Agregando un hostname estático

En este ejercicio agregaremos entradas a la base de datos de hosts locales.

1. Edite /etc/hosts y agregue una entrada para `mysystem.mydomain` que apunte a la dirección IP asociada a su tarjeta de red.
2. Agregue una segunda entrada que haga que todas las referencias a `ad.doubleclick.net` apunten a `127.0.0.1`.
3. Como ejercicio opcional, descargue el archivo de host desde: http://winhelp2002.mvps.org/hosts2.htm o más directamente desde http://winhelp2002.mvps.org/hosts.txt e instálelo en su sistema. ¿Nota alguna diferencia en su navegador al usar/remover el archivo nuevo de host?

Solution 39.2

1. As root do:

   ```
   $ echo "192.168.1.180    mysystem.mydomain"  >> /etc/hosts
   $ ping mysystem.mydomain
   ```

2. As root do:

   ```
   $ echo "127.0.0.1       ad.doubleclick.net" >> /etc/hosts
   $ ping ad.doubleclick.net
   ```

3. ```
   $ wget http://winhelp2002.mvps.org/hosts.txt
   --2014-11-01 08:57:12--  http://winhelp2002.mvps.org/hosts.txt
   Resolving winhelp2002.mvps.org (winhelp2002.mvps.org)... 216.155.126.40
   Connecting to winhelp2002.mvps.org (winhelp2002.mvps.org)|216.155.126.40|:80... connected.
   HTTP request sent, awaiting response... 200 OK
   Length: 514744 (503K) [text/plain]
   Saving to: hosts.txt
   ```



```
100%[======================================>] 514,744      977KB/s   in 0.5s

2014-11-01 08:57:13 (977 KB/s) - hosts.txt saved [514744/514744]
```

As root do:

```
$ cat hosts.txt >> /etc/hosts
```

Lab 39.3: Agregando un alias de interfaz de red

1. Configure su sistema con un nuevo alias de dispositivo de red llamado `eth0:0`, que use una nueva dirección IP que usted le asignará. Esta dirección debe ser persistente.

   Habilite el dispositivo y prúebelo.

Solution 39.3

1. ```
   $ cd /etc/sysconfig/network-scripts
   $ cp ifcfg-eth0 ifcft-eth0:0
   ```

   Edite este archivo (como root) y asegúrese que tiene las siguientes líneas:

   ```
   DEVICE=eth0:0
   BOOTPROTO=static
   ONBOOT=yes
   IPADDR=192.168.1.110
   NETMASK=255.255.255.0
   ```

   usando cualquier dirección que desee, En **RHEL 7** usted debería usar `NAME` en vez de `DEVICE`.

   Para habilitar el dispositivo puede usar **ifconfig**, **ifup** o **ip**, pero haciendo simplemente:

   ```
   $ sudo service network restart
   ```

   también demostrará que el alias nuevo es persistente. Puede probarlo con

   ```
   $ sudo ping 192.168.1.110
   ```

   usando la dirección que configuró.


   ```

 

# Capitulo 40

# Cortafuegos


### Lab 40.1: Instalación de firewalld

Si bien es cierto que el paquete **firewalld** (el cual incluye la herramienta multipropósito **firewall-cmd**) está disponible en la mayoría de las distribuciones **Linux** recientes, es posible que no esté instalado en su sistema.

Primero es necesario verificar si ya está instalado, de la siguiente forma:

```
$ which firewalld firewall-cmd
/usr/sbin/firewalld
/usr/bin/firewall-cmd
```

Si no encuentra el programa, entonces instálelo de la forma usual, usando una de las treas maneras que se muestran aquí, dependiendo de su distribución:

```
$ sudo yum install firewalld
$ sudo zypper install firewalld
$ sudo apt-get install firewalld
```

Si esto falla, entonces el paquete **firewalld** no está disponible en su distribución. Por ejemplo, este sería el caso de las distribuciones **RHEL6/CentOS6**. En este caso tendrá que instalarlo desde las fuentes.

Para hacer esto vaya a `https://fedorahosted.org/firewalld/`, desde donde puede obtener el repositorio fuente de **git**, pero es mucho más fácil descargar el archivo comprimido (firewalld-0.3.13.tar.bz2, versión disponible a la fecha en que se escribió este artículo.

Tendrá que seguir el procedimiento general para instalar desde las fuentes:

```
$ tar xvf firewalld-0.3.13.tar.bz2
$ cd firewalld-0.3.13
$ ./configure
$ make
$ sudo make install
```

Tenga en cuenta que la fuente también tiene el parámetro **uninstall**:

```
$ sudo make uninstall
```

en el caso que se arrepienta y quiera volver atrás.

Tendrá que lidiar con cualquier problema que se presente en el paso `./configure`, tal como una biblioteca faltante, etc. Cuando realiza una instalación desde un sistema de paquetes, la distribución se hace cargo de esto, pero en el caso de instalar desde las fuentes esto puede ser un complejo. Si ha ejecutado en su sistema el script **ready-for.sh** de la **Linux Foundation**, es poco probable que tenga problemas.

Nota: A pesar de que en **openSUSE 13.2** la compilación e instalación funcionará, la ejecución de **firewall-cmd** fallará con un mensaje informando de la ausencia de `python-slip`. Desafortunadamente este paquete no existe en los repositorios de **zypper**, por lo cual tendrá que descargarlo desde el mismo sitio web, https://fedorahosted.org/firewalld/, y luego hacer:

```
$ tar xvf /tmp/python-slip-0.6.1.tar.bz2
$ cd  python-slip-0.6.1
$ make
$ sudo make install
```

substituyendo por el nombre adecuado de la versión que descargó. Esperemos que la próxima versión de **openSUSE** elimine la necesidad de compilar desde las fuentes, ya que han habido solicitudes de agregar **firewalld** a las opciones disponibles.

Lab 40.2: Examinando firewall-cmd

Solo hemos revisado la superficie de cómo se puede usar el paquete **firewalld**. Casi todo se hace a través de la herramienta **filewall-cmd**, la cual está facultada para realizar una gran variedad de tareas, usando opciones con nombres muy claros.

Para tener una idea de esto solo hay que hacer:

```
$ firewall-cmd --help

Usage: firewall-cmd [OPTIONS...]
....
Service Options
  --new-service=<service>
                       Add a new service [P only]
  --delete-service=<service>
                       Delete and existing service [P only]
....
```

lo cual no reproduciremos aquí, ya que tiene 208 líneas en un sistema **RHEL 7**.

Para una explicación más detallada de todo lo que despierte su interés, haga **man firewall-cmd**, lo cual provee de un resumen, como también una lista de otras páginas **man** que describen los diversos archivos de configuración en /etc. También se aclaran conceptos como **zonas** y **servicios**.

Lab 40.3: Agregando servicios a una zona

Agregue los servicios **http** y **https** a la **zona pública** y verifique que se muestran correctamente.

Solution 40.3


```
$ sudo firewall-cmd  --zone=public --add-service=http
success
$ sudo firewall-cmd  --zone=public --add-service=https
success
$ sudo firewall-cmd --list-services --zone=public
dhcpv6-client http https ssh
```

Tenga en cuenta que si ejecutó

```
$ sudo firewall-cmd --reload
$ sudo firewall-cmd --list-services --zone=public
dhcpv6-client ssh
```

después de agregar los servicios nuevos, van a desaparecer de la lista. Este comportamiento curioso se debe a que no incluimos el parámetro `--permanent`, y la opción `--reload` recarga los servicios persistentes solamente.

Lab 40.4: Uso de la GUI del cortafuegos

Cada distribución tiene su propia herramienta gráfica para administrar el cortafuegos. En sistemas basados en **Red Hat** es posible ejecutar **firewall-config**. En **Ubuntu** se llama **gufw** y en **openSUSE** la puede encontrar como parte de **yast** en el menú gráfico de sistema.

Nos hemos concentrado en la aproximación de la línea de comandos simplemente porque queremos ser flexibles con la distribución. Sin embargo, para la mayoría de las tareas de configuración relativamente simples, probablemente encontrará que es más eficiente realizarlas desde la GUI, ya que requiere de una menor memorización.

Una vez que ha iniciado la GUI de configuración del cortafuegos, haga el ejercicio previo de agregar **http** y **https** a la zona **public**, y asegúrese de que tomó efecto.

Asegúrese de tomarse el tiempo para entender la interfaz gráfica.

Capitulo 41

Resolución básica de problemas

No hay ejercicios de laboratorio en este capítulo. Solamente se resumen los puntos discutidos anteriormente al considerar la configuración y monitoreo del sistema. Adicionalmente, prepara el escenario para el capítulo siguiente sobre el rescate del sistema, el cual tiene varios laboratorios.

Capitulo 42

Rescate del sistema

Lab 42.1: Preparando el escenario para usar medios de rescate/recuperación

En los siguientes ejercicios vamos a dañar deliberadamente el sistema y luego recuperarlo a través del uso de medios de rescate. Por lo tanto, resulta obviamiente prudente asegurarse de que puede arrancar desde el medio de rescate antes de intentar cualquier cosa más ambiciosa.

Entonces asegúrese de tener el medio de rescate, ya sea una imagen de rescate/recuperación dedicada o una imagen de instalación o Live ya sea en un disco óptico o usb.

Reinicie y asegúrese de que sabe cómo forzar el sistema para que arranque desde el medio de rescate (es probable que tenga que jugar con la configuración de la **BIOS**). Elija el modo de rescate en el inicio del sistema.

Si está usando una máquina virtual el procedimiento es lógicamente el mismo, con dos diferencias:

- Acceder a la **BIOS** puede ser difícil dependiendo del hipervisor que utilice. Algunos de ellos requieren combinaciones de teclas muy rápidas, así es que lea la documentación y asegúrese que sabe cómo hacerlo.
- Puede usar un disco óptico o físico, asegurándose que esté montado en la máquina virtual, y si es **USB** podría tener algunos obstáculos para asegurarse de que la máquina virtual puede reclamar el dispositivo físico. Generalmente lo más fácil es conectar un archivo de imagen `.iso` directamente a la máquina virtual.

Si está trabajando con una máquina virtual, obviamente las cosas son menos peligrosas, y si tiene miedo de corromper el sistema de forma irreparable, simplemente haga una copia de seguridad de la imagen de la máquina virtual antes de realizar estos ejercicios. Siempre puede reemplazar la imagen con la del respaldo más tarde.

¡No realice los siguientes ejercicios a menos que esté seguro de que puede arrancar su medio de rescate/recuperació

Lab 42.2: Recuperación de una configuración de GRUB dañada

1. Edite su archivo de configuración de **GRUB** (/boot/grub/grub.cfg, /boot/grub2/grub.cfg o /boot/grub/grub.conf), y modifique la línea del `kernel` quitando el primer caracter del valor del campo llamado `UUID`. Tome nota del caracter

que eliminó, ya que lo necesitará saber en el modo de rescate (si su sistema de archivos raíz se identifica ya sea por etiqueta o nodo del dispositivo del disco duro, haga un cambio análogo simple). Mantenga una copia de seguridad del original.

2. Reinicie la máquina. El sistema fallará al iniciar, diciendo algo como no se encontró ningún `dispositivo raíz`. También verá que se produjo un kernel `panic`.

3. Inserte en su máquina el **DVD**, **CD** o unidad **USB** (o medio de arranque por red si tiene acceso a un servidor de instalación) de **instalación** o **Live**. Reinicie de nuevo. Cuando aparezca el menú de arranque, elija entrar en modo de rescate.

4. Como alternativa puede intentar seleccionar una imagen de rescate desde el menú de **GRUB**, lo cual es ofrecido por la mayoría de las distribuciones. Tendrá la misma experiencia que usando un medio de rescate, pero no siempre funciona. Por ejemplo, si el sistema de archivos raíz está dañado será imposible hacer algo.

5. En modo rescate, asienta cuando se le pregunte si desea buscar sistemas de archivos. Si se le solicita, abra una shell y explore el sistema de rescate, ejecutando herramientas tales como **mount** y **ps**.

6. Repare su sistema dañado modificando el archivo de configuración de **GRUB**, ya sea editándolo o restaurándolo desde una copia de seguridad.

7. Escriba `exit` para volver al instalador, remueva el medio de booteo y siga las instrucciones acerca de cómo reiniciar. Reinicie la máquina. Esta vez debería iniciar normalmente.

Lab 42.3: Recuperación de la contraseña de root

1. Como root (no con **sudo**), cambie la contraseña de root. Vamos a pretender que no conocemos la contraseña nueva.

2. Desconéctese e intente conectarse de nuevo como root, usando la contraseña antigua. Obviamente no podrá.

3. Arranque usando el medio de rescate y seleccione `Rescue` cuando se le presente la opción. Permita que se monten los sistemas de archivos y luego vaya a una línea de comandos.

4. Vaya al ambiente **chroot**, de tal forma que tenga acceso a su sistema:

```
$ chroot /mnt/sysimage
```

   y reconfigure la contraseña de root a su valor original.

5. Salga, remueva el medio de rescate y reinicie. Debería poder conectarse de forma normal ahora.

Lab 42.4: Recuperación de la corrupción de la tabla de particiones

1. Conéctese como root y haga una copia de seguridad del **MBR**:

```
$ dd if=/dev/sda of=/root/mbrsave bs=446 count=1
 1+0 records in
 1+0 records out
 446 bytes (446 B) copied, 0.00976759 s, 45.7 kB/s
```

   Sea cuidadoso: asegúrese de ejecutar el comando correcto y que el archivo tiene el largo adecuado:

```
 $ sudo ls -l /root/mbrsave
-rw-r--r-- 1 root root 446 Nov 12 07:54 mbrsave
```

2. Ahora vamos a borrar el **MBR** con:

```
$ dd if=/dev/zero of=/dev/sda bs=446 count=1
1+0 records in
1+0 records out
446 bytes (446 B) copied, 0.000124091 s, 3.6 MB/s
```


3. Reinicie el sistema; este debería fallar.
4. Reinicie en el ambiente de rescate y restaure el **MBR**:

```
$ dd if=/mnt/sysimage/root/mbrsave of=/dev/sda bs=446 count=1
```

5. Salga del ambiente de rescate y reinicie. El sistema debería arrancar correctamente esta vez.

Lab 42.5: Recuperación usando la imagen de instalación

1. Este ejercicio ha sido creado específicamente para sistemas basados en **Red Hat**. Usted debería ser capaz de construir fácilmente las substituciones adecuadas para otras familias de distribuciones.

   Desinstale el paquete **zsh** (si es que está instalado):

```
$ yum remove zsh
```

   o

```
$ rpm -e zsh
```

   Tenga en cuenta que hemos elegido un paquete que generalmente no tiene dependencias con el fin de simplificar las cosas. Si usted elige algo que las tiene, tenga el cuidado de reinstalar cualquier cosa que desinstale y que sea necesario.

2. Arranque en el ambiente de rescate.
3. Reinstale (o instale) **zsh** desde el ambiente de rescate. Primero monte el medio de instalación en /mnt/source:

```
$ mount /dev/cdrom /mnt/source
```

   Luego reinstale el paquete:

```
$ rpm -ivh --force --root /mnt/sysimage /mnt/source/Packages/zsh*.rpm
```

   La opción `--force` le indica a **rpm** que use el directorio fuente para determinar la información de las dependencias y otras cosas. Tenga en cuenta que si la imagen de instalación es mucho más antigua que su sistema, el cual ha tenido probablemente muchas actualizaciones, el procedimiento completo podría colapsar.

4. Salga y reinicie.
5. Verifique que **zsh** ha sido instalado:

```
$ rpm -q zsh
zsh-5.0.2-7.el7.x86_64
```

6.
```
$ zsh
....
[coop@q7]/tmp/LFS201%
```

