Se pretende aprender sobre la ubicación de la información del kernel, información del proceso, las librerías, los archivos de registro, y los paquetes de software.

En este laboratorio llevarás a cabo las siguientes tareas:

* Investiga cómo el kernel utiliza al sistema de archivos /proc
* Utiliza el comando ps para ver la información del proceso
* Aprende a gestionar los procesos al iniciarlos, detenerlos y continuarlos
* Visualización de archivos de registro
* Gestionar la capacidad de cargar las librerías compartidas

El kernel y /proc

En esta tarea explorarás el directorio /proc y los comandos que se comunican con el kernel de Linux. El directorio /proc parece ser un directorio ordinario, como el /usr o /etc, pero no lo es. A diferencia de los directorios /usr o /etc, que por lo general se escriben en una unidad de disco, el directorio /proc es un seudo sistema de archivos ubicado en la memoria de la computadora.

El directorio /proc contiene un subdirectorio para cada proceso presente en el sistema. Los programas tales como pd y top leen la información sobre los procesos en ejecución de estos directorios. El directorio /proc también contiene la información acerca del sistema operativo y su hardware en los archivos como /proc/cpuinfo, /proc/meminfo y /proc/devices.

El subdirectorio /proc/sys contiene los seudo archivos que se pueden utilizar para alterar la configuración del kernel en ejecución. Ya que estos archivos no son archivos «reales», no se debe utilizar el editor para modificarlos; en su lugar debes utilizar el comando echo o sysctlpara sobrescribir el contenido de estos archivos. Por la misma razón, no intentes ver estos archivos en un editor, sino utiliza el comando cat o sysctl en lugar.

Para los cambios de configuración permanentes, el kernel utiliza el archivo /etc/sysctl.conf. Normalmente, el kernel utiliza este archivo para realizar cambios en los archivos /proc al iniciar el sistema.

En esta tarea explorarás algunos de los archivos ubicados en el directorio /proc:

ls /proc

Recuerda que los directorios que tienen los números para los nombres representan procesos que se ejecutan en el sistema. El primer proceso es siempre /sbin/init, po lo que el directorio /proc/1 contendrá los archivos con la información sobre el proceso init en ejecución.

El archivo cmdline dentro del directorio del proceso (/proc/1/cmdline, por ejemplo) mostrará el comando que se ejecuta. El orden en que se inician otros procesos varía mucho de un sistema al otro. Dado que el contenido de este archivo no contiene un carácter de nueva línea, un comando echo se ejecutará para hacer que el prompt vaya a una nueva línea.

Utiliza cat y después ps para ver la información sobre el proceso /sbin/init(identificador de proceso (PID) de 1):

cat /proc/1/cmdline; echo ps -p 1

Visualiza el archivo /proc/cmdline para ver qué argumentos pasan al kernel durante el arranque:

cat /proc/cmdline

Desde la terminal, escribe el siguiente comando: (La salida del ping está siendo redirigida al archivo /dev/null (que se conoce comúnmente como bit bucket).

ping localhost > /dev/null

Terminado el proceso en el primer plano presionando Ctrl-C.

A continuación, para iniciar el mismo proceso en el segundo plano, introduce:

ping localhost > /dev/null &

Al añadir el signo & al final del comando, el proceso se inicia en el segundo plano y permite al usuario mantener el control de la terminal.

La salida será algó como : [1] 158

Esto significa que este proceso tiene un número de trabajo 1 (como lo muestra la salida [1]) y un identificador de proceso (PID) de 158. Cada terminal/shell tendrá sus números de trabajo únicos. El PID vale para todo el sistema; cada proceso tiene un número de identificación único.

Para ver qué comandos se ejecutan en la terminal actual, escribe el siguiente comando:

jobs

A continuación, inicia otro comando ping en segundo plano escribiendo lo siguiente:

ping localhost > /dev/null &

Ahora, debe haber dos comandos ping ejecutándose en segundo plano. Para verificarlo, emite el comando jobs de nuevo:

jobs

Una vez que hayas comprobado que dos comandos ping se ejecutan, lleva al primer comando al primer plano escribiendo lo siguiente:

fg %1

El resultado debe ser similar al siguiente:

Observa que, una vez más, el comando ping ha tomado el control de la terminal. Para suspender (pausar) el proceso y recuperar el control de la terminal, escribe **Ctrl-Z**:

Para que este proceso continúe ejecutándose en segundo plano, ejecuta el siguiente comando:

bg %1

Emite el comando jobs de nuevo para verificar dos procesos en ejecución:

jobs

A continuación, inicia el comando ping escribiendo lo siguiente:

ping localhost > /dev/null &

Emite el comando jobs de nuevo para verificar tres procesos en ejecución:

jobs

Utilizando el número de trabajo, detenga el último comando ping con el comando kill y verifica que se haya detenido la ejecución del comando jobs:

kill %3 jobs

Por último, puedes detener todos los comandos ping con el comando killall. Después de ejecutar el comando killall, espera unos instantes, y luego ejecuta el comando jobs para comprobar que todos los procesos se hayan detenido:

killall ping jobs

En esta tarea vas a utilizar el comando top para trabajar con los procesos. Por defecto, el programa top ordena los procesos en orden descendente del porcentaje de uso de la CPU, por lo que los programas de mayor actividad estarán en la parte superior de tu lista.

Desde la ventana de la terminal, escribe los siguientes comandos:

ping localhost > /dev/null & ping localhost > /dev/null &

**Toma nota de la salida PID según los comandos anteriores! Serán diferentes a los ejemplos que estamos proporcionando. Va a utilizar los PID en los pasos posteriores**

A continuación, inicia el comando top escribiendo lo siguiente en la terminal:

top

El comando top es un programa interactivo, lo que significa que puedes emitir comandos dentro del programa. Vas a utilizar el comando top para terminar los procesos ping. Primero introduce de la letra k. Observa que apareció un prompt debajo de Swap:

En el prompt PID to kill: escribe el PID del primer proceso ping en ejecución, y a continuación, presiona **Entrar**. Observa que el prompt cambia como a continuación:

En el prompt Kill PID with signal [15]: introduce la señal para enviar a este proceso. En este caso, solamente presiona la tecla **Entrar** para utilizar la señal por defecto. Ten en cuenta que el primer comando ping se elimina y sólo un comando ping permanece en el listado (es posible que tengas que esperar unos segundos mientras el comando topactualiza):

**Para considerar**

Existen diferentes tipos de valores numéricos que se pueden enviar a un proceso. Estos son los valores predefinidos, cada uno con un significado diferente. Si quieres obtener más información sobre estos valores, teclea man kill en la ventana de la terminal.

El símbolo indica que la señal por defecto es la señal indicada por la terminar SIGTERM o el número 15.

A continuación, termina los procesos restantes ping como antes, excepto que esta vez, en el prompt de la señal Kill PID with signal [15]: utiliza el valor de 9 en lugar de aceptar el valor predeterminado 15. Presiona **Entrar** para aceptar y entrar.

**Para considerar**

La señal de matar 9 o SIGKILL es una señal «contundente» que no puede ser ignorada, a diferencia del valor por defecto de 15. Observa que todas las referencias al comando ping se han eliminado del top.

Introduce q para salir del comando top. La siguiente pantalla refleja que ambos comandos ping fueron terminados:

En esta tarea, vamos a seguir trabajando con los procesos. Vas a utilizar pkill y kill para terminar los procesos.

Para empezar, escribe los siguientes comandos en la terminal:

sleep 888888 & sleep 888888 &

El comando sleep se utiliza normalmente para hacer una pausa en un programa (script shell) por un período de tiempo específico. En este caso se utiliza sólo para proporcionar un comando que se tarda mucho tiempo en ejecutarse.

Asegúrate de tener en cuenta los PID en el sistema de los procesos sleep para los siguientes pasos! Tus PID serán diferentes a los demostrados en el laboratorio.

A continuación, determina qué trabajos se están ejecutando actualmente escribiendo:

jobs

Ahora, utiliza el comando kill para detener la primera instancia del comando sleepescribiendo lo siguiente (sustituye *PID* con el ID de proceso de tu primer comando sleep). También, ejecuta el comando jobs para verificar que el proceso se haya detenido:

kill PID jobs

**Consejo útil**: Si recuerdas el PID del primer proceso, sólo tienes que introducir el comando ps (proceso) tal como se muestra arriba.

A continuación, utiliza el comando pkill para terminar el comando sleep restante, utilizando el nombre del programa en lugar del PID:

pkill -15 sleep

Puedes utilizar el comando ps para ver los procesos. De forma predeterminada, el comando ps sólo mostrará los procesos ejecutados en el shell actual.

Inicia un proceso en segundo plano utilizando ping y visualiza los procesos actuales utilizando el comando ps:

ping localhost > /dev/null & ps

**Anota el PID del** ping**, ya va a utilizar el PID en un paso posterior.**

Ejecuta el comando ps usando la opción -e, para que se muestren todos los procesos.

ps -e

Debido a que este entorno es de un sistema operativo virtualizado, hay muchos menos procesos que lo que normalmente se muestra con Linux corriendo directamente en el hardware.

Utiliza el comando ps con la opción -o para especificar las columnas de salida.

ps -o pid,tty,time,%cpu,cmd

Utilizar la opción --sort para especificar el orden de la(s) columna(s). De manera predeterminada, una columna especificada para la clasificación se ordenará de manera ascendente, esto puede ser forzado colocando el símbolo plus + que aparece delante del nombre de la columna. Para especificar un orden descendente, utiliza el signo menos -delante del nombre de columna.

Ordena la salida del ps por %mem:

ps -o pid,tty,time,%mem,cmd --sort %mem

Mientras que el comando ps puede mostrar el porcentaje de memoria utilizado por el proceso, el comando free mostrará el uso total de la memoria del sistema:

free

Detén el comando ping con el siguiente comando kill y verifica con el comando jobs:

kill *PID* jobs