**ACTIVIDAD CLASE 12**

● Investigar y contestar las siguientes preguntas:

**¿Que es un usuario root en Linux?**

El usuario root en GNU/Linux es el usuario que tiene acceso administrativo al sistema. Los usuarios normales no tienen este acceso por razones de seguridad. Sin embargo, Kubuntu no incluye el usuario root. En su lugar, se da acceso administrativo a usuarios individuales, que pueden utilizar la aplicación "sudo" para realizar tareas administrativas. La primera cuenta de usuario que creó en su sistema durante la instalación tendrá, de forma predeterminada, acceso a sudo.

**¿Por qué ubuntu no me deja establecer la contraseña durante la instalación?**

Después de instalar Ubuntu, en realidad cualquier Linux después de instalarlo deberemos cambiar el password nada más instalarlo.Muchos sistemas, sobre todos los basados en Ubuntu, vienen sin password en el usuario root pero con nuestro usuario podemos cambiar el password desde la linea de comandos.

**¿Cuáles son los procesos típicos de Linux?**

**¿Cómo identificarlos?**

Un proceso se refiere a un programa en ejecución; es una instancia en ejecución de un programa. Se compone de la instrucción del programa, los datos leídos de archivos, otros programas o la entrada de un usuario del sistema.

Existen fundamentalmente dos tipos de procesos en Linux:

* Procesos en primer plano (también denominados procesos interactivos): estos se inicializan y controlan a través de una sesión de terminal. En otras palabras, tiene que haber un usuario conectado al sistema para iniciar dichos procesos; no se han iniciado automáticamente como parte de las funciones/servicios del sistema.
* Procesos en segundo plano (también denominados procesos automáticos/no interactivos): son procesos que no están conectados a una terminal; no esperan ninguna entrada del usuario.

Estos son tipos especiales de procesos en segundo plano que comienzan al inicio del sistema y continúan ejecutándose para siempre como servicio; no mueren. Se inician como tareas del sistema (se ejecutan como servicios), de forma espontánea. Sin embargo, un usuario puede controlarlos a través del proceso de inicio.

Diagrama

Descripción generada automáticamente

Creación de un proceso en Linux

Normalmente, se crea un nuevo proceso cuando un proceso existente hace una copia exacta de sí mismo en la memoria. El proceso hijo tendrá el mismo entorno que su padre, pero solo el número de identificación del proceso es diferente.

Hay dos formas convencionales que se utilizan para crear un nuevo proceso en Linux:

* Uso de la función System(): este método es relativamente simple, sin embargo, es ineficiente y tiene riesgos de seguridad significativamente determinados.
* Uso de la función fork() y exec(): esta técnica es un poco avanzada pero ofrece mayor flexibilidad, velocidad y seguridad.

¿Cómo identifica Linux los procesos?

Debido a que Linux es un sistema multiusuario, lo que significa que diferentes usuarios pueden ejecutar varios programas en el sistema, cada instancia en ejecución de un programa debe ser identificada de forma única por el kernel.

Y un programa se identifica por su ID de proceso (PID), así como su ID de proceso principal (PPID), por lo tanto, los procesos se pueden clasificar en:

* Procesos principales: estos son procesos que crean otros procesos durante el tiempo de ejecución.
* Procesos secundarios: estos procesos son creados por otros procesos durante el tiempo de ejecución.

El proceso de inicialización es la madre (padre) de todos los procesos del sistema, es el primer programa que se ejecuta cuando se inicia el sistema Linux; gestiona todos los demás procesos del sistema. Lo inicia el propio kernel, por lo que, en principio, no tiene un proceso padre.

El proceso init siempre tiene el ID de proceso 1. Funciona como padre adoptivo para todos los procesos huérfanos.

Puede usar el comando pidof para encontrar el ID de un proceso:

# pidofsystemd

# pidof top

# pidof httpd

Imagen que contiene Texto

Descripción generada automáticamente

Para encontrar el ID del proceso y el ID del proceso principal del shell actual, ejecute:

$ echo $$

$ echo $PPID

Texto

Descripción generada automáticamente con confianza baja

Una vez que ejecute un comando o programa (por ejemplo, cloudcmd - CloudCommander), iniciará un proceso en el sistema. Puede iniciar un proceso en primer plano (interactivo) de la siguiente manera, se conectará al terminal y un usuario puede enviarlo de entrada:

# cloudcmd

Texto

Descripción generada automáticamente

Para iniciar un proceso en segundo plano (no interactivo), use el símbolo & , aquí, el proceso no lee la entrada de un usuario hasta que se mueve al primer plano.

# cloudcmd&

# jobs

Texto

Descripción generada automáticamente con confianza baja

También puede enviar un proceso a un segundo plano suspendiéndolo usando [Ctrl + Z] , esto enviará la señal SIGSTOP al proceso, deteniendo así sus operaciones; se vuelve inactivo:

# tar -cf backup.tar /backups/\* **#press Ctrl+Z**

# jobs

Para continuar ejecutando el comando suspendido anteriormente en segundo plano, use el comando bg:

# bg

Para enviar un proceso en segundo plano al primer plano, use el comando fg junto con el ID del trabajo así:

# jobs

# fg %1

Texto

Descripción generada automáticamente

También te puede interesar: Cómo iniciar el comando de Linux en segundo plano y desconectar el proceso en la terminal

Durante la ejecución, un proceso cambia de un estado a otro dependiendo de su entorno/circunstancias. En Linux, un proceso tiene los siguientes estados posibles:

* En ejecución: aquí está ejecutándose (es el proceso actual en el sistema) o está listo para ejecutarse (está esperando ser asignado a una de las CPU).
* En espera: en este estado, un proceso está esperando que ocurra un evento o un recurso del sistema. Además, el kernel también diferencia entre dos tipos de procesos de espera; Los procesos de espera interrumpibles - pueden ser interrumpidos por señales y los procesos de espera ininterrumpida - están esperando directamente en las condiciones del hardware y no pueden ser interrumpidos por ningún evento/señal.
* Detenido: en este estado, un proceso se detuvo, generalmente al recibir una señal. Por ejemplo, un proceso que se está depurando.
* Zombi: aquí, un proceso está muerto, se ha detenido pero todavía tiene una entrada en la tabla de procesos.

Hay varias herramientas de Linux para ver/enumerar los procesos en ejecución en el sistema, los dos tradicionales y más conocidos son los comandos principales:

Muestra información sobre una selección de los procesos activos en el sistema como se muestra a continuación:

# ps

# ps -e | head

Texto

Descripción generada automáticamente

vista dinámica en tiempo real de un sistema en ejecución como se muestra en la captura de pantalla a continuación:

# top

Imagen que contiene texto, verde, grande, firmar

Descripción generada automáticamente

Lea esto para ver más ejemplos de uso principales: 12 ejemplos de comandos TOP en Linux

Glances es una herramienta de supervisión del sistema relativamente nueva con funciones avanzadas:

# glances

Imagen que contiene Texto

Descripción generada automáticamente

Para obtener una guía de uso completa, lea: Glances: una herramienta avanzada de monitoreo del sistema en tiempo real para Linux

Hay varias otras herramientas útiles de monitoreo del sistema Linux que puede usar para enumerar los procesos activos, abra el enlace a continuación para leer más sobre ellos:

1. 20 herramientas de línea de comandos para supervisar el rendimiento de Linux
2. 13 herramientas de supervisión de Linux más útiles

Cómo controlar procesos en Linux

Linux también tiene algunos comandos para controlar procesos como kill, pkill, pgrep y killall, a continuación se muestran algunos ejemplos básicos de cómo usarlos:

$ pgrep -u tecmint top

$ kill 2308

$ pgrep -u tecmint top

$ pgrep -u tecmint glances

$ pkill glances

$ pgrep -u tecmint glances

Texto

Descripción generada automáticamente

Para aprender a usar estos comandos en profundidad, para matar/terminar procesos activos en Linux, abra los enlaces a continuación:

1. Una guía para los comandos Kill, Pkill y Killall para finalizar los procesos de Linux
2. Cómo encontrar y eliminar procesos en ejecución en Linux

Tenga en cuenta que puede usarlos para eliminar aplicaciones que no responden en Linux cuando su sistema se congela.

La forma fundamental de controlar los procesos en Linux es enviándoles señales. Hay varias señales que puede enviar a un proceso, para ver todas las señales que se ejecutan:

$ kill -l

Texto

Descripción generada automáticamente

Para enviar una señal a un proceso, use los comandos kill, pkill o pgrep que mencionamos anteriormente. Pero los programas solo pueden responder a las señales si están programados para reconocer esas señales.

Y la mayoría de las señales son para uso interno del sistema o para programadores cuando escriben código. Las siguientes son señales que son útiles para un usuario del sistema:

* SIGHUP 1: enviado a un proceso cuando su terminal de control está cerrada.
* SIGINT 2: enviado a un proceso por su terminal de control cuando un usuario interrumpe el proceso presionando [Ctrl + C] .
* SIGQUIT 3: se envía a un proceso si el usuario envía una señal de salida [Ctrl + D] .
* SIGKILL 9: esta señal termina (mata) inmediatamente un proceso y el proceso no realizará ninguna operación de limpieza.
* SIGTERM 15: esta es una señal de finalización del programa (kill enviará esto por defecto).
* SIGTSTP 20 - enviado a un proceso por su terminal de control para solicitar que se detenga (parada del terminal); iniciado por el usuario presionando [Ctrl + Z] .

Los siguientes son ejemplos de comandos de eliminación para eliminar la aplicación Firefox usando su PID una vez que se congela:

$ pidoffirefox

$ kill 9 2687

OR

$ kill -KILL 2687

OR

$ kill -SIGKILL 2687

Para matar una aplicación usando su nombre, use pkill o killall así:

$ pkillfirefox

$ killallfirefox

En el sistema Linux, todos los procesos activos tienen una prioridad y cierto valor agradable. Los procesos con mayor prioridad normalmente obtendrán más tiempo de CPU que los procesos de menor prioridad.

Sin embargo, un usuario del sistema con privilegios de root puede influir en esto con los comandos nice y renice.

Desde la salida del comando top, el NI muestra el valor agradable del proceso:

$ top

Imagen que contiene texto, verde, grande, firmar

Descripción generada automáticamente

Utilice el comando nice para establecer un valor agradable para un proceso. Tenga en cuenta que los usuarios normales pueden atribuir un buen valor de cero a 20 a los procesos que poseen.  
Solo el usuario root puede usar valores agradables negativos.

Para cambiar la prioridad de un proceso, use el comando renice de la siguiente manera:

$ renice +8 2687

$ renice +8 2103

Consulte nuestros artículos útiles sobre cómo administrar y controlar los procesos de Linux.

1. Gestión de procesos de Linux: arranque, apagado y todo lo demás
2. Encuentre los 15 procesos principales por uso de memoria con "top" en modo por lotes
3. Encuentre los principales procesos en ejecución por mayor uso de memoria y CPU en Linux
4. Cómo encontrar un nombre de proceso usando un número PID en Linux

● Investigar y establecer una contraseña para el usuario root:1 Abre la Terminal (Control + Alt + T)2 Teclea (sin comillas) "sudo su"3 Introduce tu clave actual4 Teclea "passwd root" y escribe tu nueva clave5 Pulsa enter y cierra la terminal

Actividad opcional terminal

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza baja

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente