Práctica 2, explicación

Configuración de discos IDE (Integrated Device Electronics)

Referido a la forma en que se conectan los discos duros o unidades de almacenamiento a una placa base. IDE es una tecnología antigua reemplazada por SATA.

Los discos IDE se denominan ,/dev/hbb, /dev/hdc y /dev/hdd según la ubicación de su bus.

- /dev/hda: es el primer disco duro, configurado como Master
- /dev/hdb es el segundo disco duro, configurado como Slave
- /dev/hdc es el tercer disco duro
- /dev/hdd es el cuarto disco

Las particiones primarias se enumeran del 1 al 4 y se usan para dividir un disco duro en secciones separadas. Las particiones lógicas se numeran a partir de 5 y se usan para crear particiones adicionales en una partición primaria ya existente.

Configuración de discos SCI (Small Computer System Interface)

En resumen, la principal diferencia entre la configuración de discos IDE y la configuración de discos SCSI es la forma en que se nombran los discos y la identificación de los dispositivos. Además, en la configuración de discos SCSI, la partición primaria se puede marcar como activa, mientras que en la configuración de discos IDE, la partición primaria se utiliza para arrancar el sistema operativo.

GNU/ Linux evolucionó y ahora se usa UDEV para gestionar dispositivos. Su función es controlar dinámicamente los archivos del sistema de archivos /dev solo en base al hardware detectado. Permite que los dispositivos se nombres de forma consistente en distintos arranques del SO.

Desde Debian/Squeeze se adoptó una nueva nomenclatura para los dispositivos en GNU/Linux, en la que los dispositivos que antes se llamaban hdX ahora se llaman sdX. Se han utilizado nuevos mecanismos de nomenclatura, como la asignación de nombres persistentes por UUID y la asignación de etiquetas a los dispositivos, para mejorar la consistencia y la legibilidad del sistema de archivos /dev. Los UUID son códigos únicos asignados a cada dispositivo, mientras que las etiquetas son nombres descriptivos asignados por el usuario. Estos mecanismos permiten identificar los dispositivos de manera consistente y fácil de entender.

Características GNU/Linux

- No hay extensión en el nombre de un archivo.
- Los subdirectorios no se separan con \.
- Case sensitive.
- Entre un comando y sus parámetros debemos dejar obligatoriamente un espacio en blanco.
- Separación entre entorno gráfico y texto.

VIM

El editor de textos vim está presente en cualquier distribución de GNU/Linux y ofrece tres modos de ejecución: modo Insert (Ins o i), modo Visual (v) y modo de Órdenes o Normal (Esc). Además, se pueden enviar una serie de comandos útiles, como w para escribir cambios, q o q! para salir del editor, dd para cortar, y para copiar al portapapeles, p para pegar desde el portapapeles, u para deshacer y /frase para buscar la palabra "frase" dentro del archivo.

Usuarios

El sistema de usuarios es una forma de controlar y administrar el acceso a recursos y servicios.

Cada usuario del sistema debe tener credenciales para acceder al mismo. Root es el usuario con más privilegios y tiene el control total sobre el sistema.

- /etc/passwd es el archivo que almacena la información de todos los usuarios del sistema. El archivo incluye el nombre del usuario, su identificador de usuario (UID), identificador de grupo (GID), descripción y ubicación de su directorio principal.
- /etc/group almacena información de grupos de usuarios y sus miembros.
- /etc/shadow almacena contraseñas de los usuarios.

Algunos comandos son

- useradd permite agregar un nuevo usuario al sistema y modificar los archivos /etc/passwd y /etc/group. También se puede usar la alternativa "adduser".
- passwd se utiliza para asignar o cambiar la contraseña de un usuario y modifica el archivo /etc/shadow.
- usermod se utiliza para modificar un usuario existente. Con el parámetro -g se puede cambiar el grupo de login del usuario, con -G se pueden agregar o eliminar grupos adicionales y con -d se puede modificar el directorio de inicio (home directory) del usuario. Cada vez que se utiliza el comando "usermod", los archivos /etc/passwd y /etc/group se modifican para reflejar los cambios.

- userdel se utiliza para eliminar un usuario del sistema. Si se utiliza este comando, el usuario y su directorio de inicio se eliminan del sistema.
- groupdel se utiliza para eliminar un grupo. Si un usuario pertenece a un grupo que se elimina, se le retirará automáticamente la pertenencia a ese grupo.

Permisos

Forma de restringir o permitir el acceso a archivos y directorios. Se aplican a usuarios y se basan en tipos de permisos: lectura (R), escritura (W) y ejecución (X). Cada permiso se asocia con un valor octal, donde L= 4, W=2 y X=1.

Se aplican sobre 3 tipos de usuarios, dueño del archivo, grupo del archivo, otros usuarios.

Un ejemplo podría ser "chmod 755 /tmp/script". En este caso, el dueño tiene permisos de lectura, escritura y ejecución (valor octal de 7), mientras que el grupo y otros usuarios tienen permisos de lectura y ejecución (valor octal de 5). Esto significa que el dueño del archivo puede leer, escribir y ejecutarlo, mientras que el grupo y otros usuarios solo pueden leer y ejecutarlo.

Bootloader

Programa que arranca el SO. Puede cargar un entorno previo a la carga del sistema. El MBR es la parte del bootloader que se encuentra en los primeros 512 bytes de un disco duro y tiene un MBC y la tabla de particiones. El MBR existe en todos los discos y es el responsable de cargar el primer sector del disco y dar inicio al proceso de arranque del sistema operativo.

System V

Conjunto de pasos que se siguen para iniciar el SO.

1. Ejecuta el código del BIOS.
2. BIOS ejecuta el POST (Prueba de auto-encendido), el cual es una prueba de diagnóstico hecha cada vez que encendemos una computadora para asegurar el funcionamiento en general.
3. BIOS lee el sector de arranque (MBR).
4. Se carga el gestor de arranque (MBC).
5. El bootloader carga el kernel y el initrd. El initrd es un sistema de archivos temporal para darle al kernel un entorno mínimo de arranque.
6. Se monta el initrd como sistema de archivos raíz y se inicializan componentes esenciales.
7. El kernel ejecuta el proceso init y se desmonta el initrd.
8. Se lee /etc/inittab.
9. Se ejecutan los scripts apuntados por el runlevel 1.

10. El final del runlevel 1 le indica que vaya al runlevel por defecto.
11. Se ejecutan los scripts apuntados por el runlevel por defecto.
12. El sistema está listo para usarse.

Init

Se encarga de cargar los subprocesos para el funcionamiento del SO. Tiene el PID 1 y es el padre de todos los procesos, monta los filesystems y hace disponible los demás dispositivos.

Runlevels

Referido a los diferentes modos de arranque del SO. Cada runlevel representa un estado diferente del sistema y cada uno es responsable de iniciar o detener servicios.

Los runlevels se enumeran del 0 al 6. 0 representa sistema apagado, 1 es el modo de usuario único (en el que se inician solo los servicios básicos necesarios para el mantenimiento del sistema), 2 a 5 son niveles de usuario múltiple con diferentes configuraciones de servicios, y 6 es el nivel de reinicio del sistema.

En la mayoría de las distribuciones de Linux, el runlevel predeterminado es el nivel 3 o el nivel 5. En el nivel 3, el sistema arranca en modo de consola de texto, mientras que en el nivel 5, el sistema arranca en modo gráfico.

Cada runlevel tiene su propio conjunto de scripts de inicio y parada de servicios que se ejecutan automáticamente cuando el sistema se inicia o se apaga en ese runlevel en particular. Estos scripts se encuentran generalmente en el directorio `/etc/rc.d` o `/etc/init.d`, dependiendo de la distribución.

- Nivel 0 detiene todos los procesos del sistema y lo apaga seguramente.
- Nivel 1 (single user mode) se usa para hacer tareas de mantenimiento en modo usuario único, sin iniciar servicios de red o gráficos.
- Nivel 2 (multiuser) inicia servicios básicos de red.
- Nivel 3 (full multiuser mode console) inicia todos los servicios básicos y permite el acceso al sistema por consola, pero no inicia el entorno gráfico por defecto.
- Nivel 4 no es usado, se reserva para uso personalizado.
- Nivel 5 (modo multiusuario completo con login gráfico) inicia todos los servicios básicos y permite el acceso al sistema con un entorno gráfico.
- Nivel 6 (reboot) reinicia el sistema después de detener los procesos seguramente.

Al iniciar o apagar el SO, se ejecutan scripts ubicatos en /etc/init.d los cuales se encargan de iniciar o detener servicios y procesos.

Además para controlar los scripts para cada nivel, se usan enlaces simbólicos.

Insserv

Herramienta para administrar enlaces simbólicos y resuelve dependencias entre scripts de inicio y detención de servicios y establece el orden en que deben ejecutarse en el inicio o apagado del sistema.

Una dependencia es la relación entre dos servicios donde uno necesita que otro se ejecute para que se inicie. Una facility es un recurso del sistema que un servicio necesita para funcionar.

La palabra *provides* en los scripts de inicio y detención de servicios indica qué facilities son proporcionadas por el servicio en cuestión.

Proceso de arranque, upstart

Upstrart es un sistema de inicio de servicios que reemplaza SystemV. La diferencia principal es que permite ejecutar trabajos de forma asincrónica.

Los trabajos en Upstart se denominan jobs y definen servicios y tareas a ejecutarse por el sistema de inicio. Cada job está detallado en /etc/init.

Hay jobs de tipo task y service. Los jobs task realizan tareas y luego terminan en cambio los jobs service tienen una ejecución indeterminada y pueden ser reiniciados en caso de fallo.

Los jobs se ejecutan respondiendo a eventos como arrancar el equipo, inserción de dispositivos USB, etc. Cada job tienen un objetivo y un estado que determina qué proceso ejecuta.

Systemd

Sistema que centraliza la administración de demonios y librerías del SO. Mejora el proceso de arranque del sistema permitiendo el paralelismo en la ejecución de procesos.

El proceso systemd reemplaza al proceso init y se convierte en el proceso principal del sistema.

Un *demonio* es un proceso que se ejecuta en segundo plano sin la intervención directa del usuario. Se encargan de tareas como gestión de archivos, comunicación en red, etc.

Systemd usa “units” para representar los diferentes componentes del sistema que deben ser controlados. Las cuales se dividen en diferentes tipos, como service, socket, target, etc.

- Las unidades service controlan un servicio específico como un servidor web o una BBDD
- Las unidades socket encapsulan la comunicación entre procesos y se pueden activar mediante una conexión de socket.
- Target agrupan varias unidades o establecen puntos de sincronización durante el proceso de arranque del sistema.
- Snapshot almacenan el estado de un conjunto de unidades que se pueden restaurar más tarde.

Cada unidad puede estar en estado activo o inactivo, lo que indica si está en ejecución.

Systemd tiene una función llamada "Activación por Socket" que permite iniciar servicios cuando se requieren, en lugar de iniciarlos todos durante el arranque del sistema. Cuando un socket recibe una conexión, systemd activa el servicio correspondiente y le pasa el socket para su uso. Esto evita la necesidad de definir dependencias entre servicios y mejora el rendimiento del sistema al evitar la carga innecesaria de servicios. En resumen, la Activación por Socket en systemd permite una carga de servicios más eficiente y efectiva en el sistema.

Systemd utiliza "cgroups" para organizar un grupo de procesos en una jerarquía y agruparlos según su relación, como un servidor web con sus dependencias.

Cgroups permite realizar el seguimiento de los procesos mediante un subsistema de cgroups, lo que significa que no se utiliza el PID para el seguimiento. Además, systemd permite limitar el uso de recursos para los procesos de forma individual o en grupo.

En resumen, systemd utiliza cgroups para organizar y agrupar procesos, lo que permite un seguimiento más eficiente y limitar el uso de recursos en el sistema.

Redirecciones

- Destructiva >: si el archivo de destino no existe, se creará uno nuevo. Pero si el archivo ya existe, se eliminará todo el contenido anterior y se escribirá el nuevo contenido en su lugar.
- No destructiva >>: si el archivo de destino no existe, se creará uno nuevo. Si el archivo ya existe, el nuevo contenido se agregará al final del archivo, sin eliminar la información anterior.

Pipes |

El símbolo `|` nos permite comunicar dos procesos mediante una tubería desde la shell. Esto significa que el pipe conecta la salida estándar (stdout) del primer comando con la entrada estándar (stdin) del segundo comando.

Por ejemplo, al ejecutar el comando `ls | more`, la salida del comando `ls` se envía como entrada al comando `more`. Se pueden anidar varios pipes para conectar varios comandos.