MEMORIAS

Las computadoras tienden a acceder al almacenamiento en formas particulares

- El acceso tiende a ser secuencial: accede a la dirección N, es muy probable que la dirección N+1 sea la próxima a acceder.
- El acceso tiende a ser localizado: Si se accede a la dirección X, es muy probable que otras direcciones alrededor de X también serán accedidas en el futuro.

Distintos tipos de memorias

- Registros de CPU
- Memorias caché
- RAM
- Discos duros
- Almacenamiento fuera de línea para respaldos

Registros de la CPU

Todos los diseños de CPU de hoy día incluyen registros. Los registros de CPU se ejecutan a la misma velocidad que el resto del CPU. La razón para esto es que casi todas las operaciones realizadas por el CPU envuelven registros de una forma u otra.

Memoria caché

El propósito de la memoria caché es actuar como una memoria temporal. Cuando se leen datos desde la RAM, el sistema de hardware verifica si los datos están en caché, si es así estos son recuperados rápidamente y utilizados por el CPU. De lo contrario, estos se leen desde la RAM y, mientras se transfieren al CPU, también se colocan en caché. La única diferencia entre el acceso de los datos en caché y desde la RAM es la cantidad de tiempo que toma para que los datos sean recuperados.

Niveles de caché

- La caché L1 a menudo está ubicada en el chip del CPU mismo y se ejecuta a la misma velocidad que el CPU.
- La caché L2 usualmente es parte del módulo de CPU, se ejecuta a las mismas velocidades que el CPU (o casi) y normalmente es un poco más grande y lenta que la caché L1.

Algunos sistemas (normalmente servidores de alto rendimiento) también tienen caché L3, que usualmente forma parte del sistema de la tarjeta madre. Como puede imaginarse, la caché L3 es más grande (y casi con seguridad más lenta) que la caché L2.

Memoria principal – RAM

La RAM es utilizada tanto para almacenar datos como para almacenar los programas en uso. La velocidad de la RAM en la mayoría de los sistemas actuales está entre la velocidad de la memoria caché y la de los discos duros.

La operación básica de la RAM en el nivel más bajo, están los chips de RAM — circuitos integrados que "recuerdan". Estos chips tienen cuatro tipos de conexiones con el mundo externo: Conexiones de energía, de datos, de lectura/escritura y de direcciones.

Pasos para almacenar datos en RAM:

- Los datos a almacenar se presentan a las conexiones de datos.
- La dirección en la que los datos se almacenan se presentan a las conexiones de dirección.
- La conexión de lectura/escritura se coloca en modo de escritura.

La recuperación de datos:

- La dirección de los datos deseados se presenta a las conexiones de direcciones.
- La conexión en la que los datos se almacenan se presenta a las conexiones de dirección.
- Los datos deseados son leídos desde las conexiones de datos.

Discos duros

Los discos duros son no-volátiles — lo que significa que los datos se mantienen allí, aún después que se ha desconectado la energía. Su naturaleza no-volátil los hace ideal para el almacenamiento de programas y datos para su uso a largo plazo, no es posible ejecutar los programas directamente cuando son almacenados en discos duros. Los discos duros son de al menos un orden de magnitud más lento que todas las tecnologías electrónicas utilizadas para caché y RAM.

Memoria virtual en términos sencillos

Un programa necesita determinada cantidad de memoria disponible para poder ejecutarse, un byte menos y la aplicación no será capaz de ejecutarse. Con la memoria virtual en vez de concentrarse en cuanta memoria necesita una aplicación para ejecutarse, un sistema operativo con memoria virtual continuamente trata de encontrar una respuesta a la pregunta “¿qué tan poca memoria necesita la aplicación para ejecutarse?".
El número real de bytes necesarios para cada acceso de memoria varían de acuerdo a la arquitectura del CPU, la instrucción misma y el tipo de dato. Pero si solamente una parte de la aplicación está en memoria en un momento dado, ¿dónde está el resto?
El resto se mantiene en disco, el disco actúa como un almacenamiento de respaldo para la RAM; un medio más lento y también más grande que actúa como un "respaldo" para un almacenamiento más rápido y más pequeño. Las partes de la aplicación que actualmente se necesitan se mantienen en RAM solamente por el tiempo en que son realmente requeridas.

Memoria virtual: Los detalles

El espacio de direcciones virtuales es el espacio de direcciones máximo disponible para una aplicación.
La palabra "virtual" en el espacio de direcciones virtuales, significa que este es el número total de ubicaciones de memoria direccionales disponibles para una aplicación.
Para implementar la memoria virtual, es necesario tener un hardware especial de administración de memoria. Este hardware a menudo se conoce como un MMU (Memory Management Unit). Sin un MMU, cuando el CPU accede a la RAM, las ubicaciones reales de RAM nunca cambian — la dirección de memoria 123 siempre será la misma dirección física dentro de la RAM.

Sin embargo, con un MMU, las direcciones de memoria pasan a través de un paso de traducción antes de cada acceso de memoria. Como resultado de esto, la sobrecarga relacionada con el seguimiento de las traducciones de memoria virtual a física sería demasiado. En vez de esto, la MMU divide la RAM en páginas — secciones contiguas de memoria de un tamaño fijo que son manejadas por el MMU como unidades sencillas.

Memoria secundaria. Conceptos fundamentales y administración

La memoria secundaria es aquella memoria no volátil y que sirve de soporte al sistema, almacenando todos los programas de manera permanente, y desde la cual se toman los programas que deben ser ejecutados, para luego moverlos a la memoria principal.

¿Qué significa “particionar” un disco?

Particionar un disco significa dividir un disco físico en varios discos lógicos. Una partición es un conjunto de bloques contiguos dentro de un disco rígido físico, y cada partición es interpretada.

- Tener múltiples particiones permite “encapsular” los datos.

- Múltiples particiones pueden tener formatos distintos. Esto hace que podamos tener un mismo disco, particionado para distintos usos, lo que mejora el rendimiento del sistema.
- Realizando particiones podemos limitar el tamaño que puede utilizar un proceso o un usuario.

Existen distintos tipos de particiones:

- Las particiones primarias son particiones que toman hasta cuatro de las ranuras de particiones en la tabla de particiones del disco duro.
- Las particiones extendidas fueron desarrolladas en respuesta a la necesidad de más de cuatro particiones por unidad de disco.

Las particiones lógicas son aquellas que están contenidas dentro de una partición extendida, son iguales a una partición primaria no extendida.

Sistema de archivos

Un sistema de archivos es un método para representar datos en un dispositivo de almacenamiento masivo. Los sistemas de archivos usualmente incluyen las características siguientes:

- Almacenamiento de datos basados en archivos.
- Estructura de directorio jerárquico.
- Seguimiento de la creación de archivos, tiempos de acceso y de modificación.
- Algún nivel de control sobre el tipo de acceso permitido para un archivo especifico.
- Un concepto de propiedad de archivos.
- Contabilidad del espacio utilizado.

Almacenamiento de datos basado en archivos

Mientras que los sistemas de archivos que utilizan esta metáfora para el almacenamiento de datos son prácticamente universales que casi se consideran como la norma, todavía existen varios aspectos que se deben considerar.

Estructura de directorios jerárquico

Los directorios son usualmente implementados como archivos, lo que significa que no se requiere de utilidades especiales para mantenerlos.
Más aún, puesto que los directorios son en sí mismos archivos, y los directorios contienen archivos, los directorios pueden a su vez contener otros directorios, conformando una estructura jerárquica de múltiples niveles.

Seguimiento de la creación de archivos, tiempo de acceso y modificación

La mayoría de los sistemas de archivos mantienen un seguimiento del tiempo en el que se creó un archivo; otros mantienen un seguimiento de los tiempos de acceso y modificación.

Control de acceso

La mayoría de los sistemas de archivos modernos combinan dos componentes en una metodología cohesiva de control de acceso:

- Identificación del usuario.
- Lista de acciones permitidas.

La identificación de usuarios significa que el sistema de archivos primeramente debe ser capaz de identificar unívocamente a usuarios individuales. Luego, el sistema de archivos debe ser capaz de mantener listas de las acciones que son permitidas (o prohibidas) para cada archivo. Las acciones a las que se les hace seguimiento más a menudo son: leer, escribir y ejecutar un archivo.

Contabilidad del espacio utilizado

Un administrador de sistemas debería al menos ser capaz de determinar fácilmente el nivel de espacio libre disponible para cada sistema de archivos. Además, los sistemas de archivos con capacidades de identificación de usuarios bien definidas, a menudo incluyen la característica de mostrar la cantidad de espacio que un usuario particular ha consumido.
Tomando este paso un poco más allá, algunos sistemas de archivos incluyen la habilidad de establecer los límites de uso del usuario (conocidos comúnmente como cuotas de disco) en la cantidad de espacio en disco que pueden consumir. Algunos sistemas de archivos permiten que el usuario se exceda de su límite solamente una vez, mientras que otros implementan un "período de gracia" durante el que aplica un segundo límite más alto.

Tecnologías avanzadas de almacenamiento

Almacenamiento accesible a través de la red

Combinando redes con las tecnologías de almacenamiento masivo puede resultar en una flexibilidad excelente para los administradores de sistemas. Con este tipo de configuración se tienen dos beneficios posibles:

- Consolidación del almacenamiento.
- Administración simplificada.

Con la configuración apropiada, es posible suministrar acceso al almacenamiento a velocidades comparables al almacenamiento conectado directamente, y hace posible reducir los costos. El espacio libre está consolidado, en vez de esparcido (pero no utilizable globalmente) entre los clientes.
Los servidores de almacenamiento centralizado también pueden hacer muchas tareas administrativas más fáciles. Los respaldos también se pueden simplificar en gran medida usando un servidor de almacenamiento centralizado.

Almacenamiento basado en RAID

RAID es el acrónimo para Redundant Array of Independent Disks, Formación de Discos Independientes Redundantes2. Como su nombre lo implica, RAID es una forma para que discos múltiples actúen como si se tratasen de una sola unidad. RAID se ve como el método para tener varios discos menos costosos sustituyendo una unidad más costosa.

Niveles de RAID

Existen tres niveles de RAID que terminaron siendo ampliamente utilizados:

- Nivel 0.
- Nivel 1 (RAID I).
- Nivel 5 (RAID V).

RAID 0

Se utiliza para doblar el rendimiento y para fusionar todos los discos duros en un sólo disco para aumentar la capacidad de almacenamiento. Es una partición lógica cuyo tamaño es igual a la suma de los discos integrados en el sistema RAID.

RAID 1

Es utilizado para garantizar la integridad de los datos, en caso de un fallo de uno de los discos duros, es posible continuar las operaciones en el otro disco duro sin ningún problema. El tipo de RAID 1 se llama comúnmente “mirroring” debido a que éste hace una simple copia del primer disco.

RAID 5

RAID 5 trata de combinar los beneficios de RAID 0 y RAID 1, a la vez que trata de minimizar sus desventajas. Igual que RAID 0, un RAID 5 consiste de múltiples unidades de disco, cada una dividida en porciones. Esto permite a una formación RAID 5 ser

más grande que una unidad individual. Como en RAID 1, una formación RAID 5 utiliza algo de espacio en disco para alguna forma de redundancia, mejorando así la confiabilidad.

Niveles RAID anidados

Este es el tipo de cosas que se pueden hacer. He aquí los niveles de RAID más comunes:

- RAID 1+0
- RAID 5+0
- RAID 5+1

RAID anidados:

- El orden en el que los niveles RAID son anidados pueden tener un gran impacto en la confiabilidad.
- Los costos pueden ser altos .

Implementaciones RAID

Se deben llevar a cabo las tareas siguientes:

- Dividir las peticiones de E/S entrantes a los discos individuales de la formación
- Para RAID 5, calcular la paridad y escribirla al disco apropiado en la formación
- Supervisar los discos individuales en la formación y tomar las acciones apropiadas si alguno falla
- Controlar la reconstrucción de un disco individual en la formación, cuando ese disco haya sido reemplazado o reparado
- Proporcionar los medios para permitir a los administradores que mantengan la formación

Hardware RAID

Una implementación de hardware RAID usualmente toma la forma de una tarjeta controladora de disco. La tarjeta ejecuta todas las funciones relacionadas a RAID y controla directamente las unidades individuales en las formaciones conectadas a ella.

- Programas de utilerías especializados que funcionan como aplicaciones bajo el sistema operativo anfitrión, presentando una interfaz de software a la tarjeta controladora.
- Una interfaz en la tarjeta usando un puerto serial que es accedido usando un emulador de terminal.
- Una interfaz tipo BIOS que solamente es accesible durante la prueba de encendido del sistema.

Algunas controladoras RAID tienen más de un tipo de interfaz administrativa disponible. Por razones obvias, una interfaz de software suministra la mayor flexibilidad, ya que permite funciones administrativas mientras el sistema operativo se está ejecutando.

Software RAID

Software RAID es RAID implementado como. Como tal, proporciona más flexibilidad en términos de soporte de hardware. Esto puede reducir dramáticamente el costo de implementar RAID al eliminar la necesidad de adquirir hardware costoso especializado.
El software RAID tiene ciertas limitaciones que no están presentes en hardware RAID. La más importante a considerar es el soporte para el arranque desde una formación de software RAID.

Administración de volúmenes lógicos

La administración de volúmenes lógicos o logical volume management (LVM). LVM hace posible tratar a los dispositivos físicos de almacenamiento masivo como bloques de construcción a bajo nivel en los que se construyen diferentes configuraciones de almacenamiento.

Agrupamiento de almacenamiento físico

Los dispositivos lógicos de almacenamiento masivo (o volúmenes lógicos) pueden ser más grandes en capacidad que cualquiera de los dispositivos físicos de almacenamiento subyacentes.

Esto hace posible que un administrador de sistemas trate a todo el almacenamiento como un sólo parque de recursos de almacenamiento, disponible para ser utilizado en cualquier cantidad. Además, posteriormente se pueden añadir unidades a ese parque, haciendo un proceso directo el mantenerse al día con las demandas de almacenamiento de sus usuarios.

Redimensionamiento de volúmenes lógicos

LVM hace posible incrementar fácilmente el tamaño de un volumen lógico, permite incrementar su tamaño sin ninguna reconfiguración física. Dependiendo del entorno de su sistema operativo, se puede hacer esto dinámicamente o quizás requiera una pequeña cantidad de tiempo fuera de servicio para llevar a cabo el redimensionamiento.

PROCESOS

Proceso de arranque en GNU/Linux

El proceso de arranque consta de cuatro etapas:

1. BIOS.
2. Bootloader.
3. Kernel.
4. Init

Primera etapa: BIOS

Esta etapa inicia en el momento en que se enciende la PC. Es allí cuando el BIOS (Basic Input Output System) toma el control del sistema para poder realizar operaciones básicas de hardware (reconocimiento, prueba de la memoria, etc.). Una vez que el BIOS completa las operaciones, se encargará de cargar en memoria el bootloader que inicia la segunda etapa.

Segunda etapa: Bootloader

El bootloader o cargador de arranque, es un programa encargado de iniciar el sistema operativo instalado. Este programa se guarda en una porción del disco llamada MBR (Master Boot Record). El MBR ocupa solo 512 bytes en el disco, de los cuales 2 bytes corresponden al "magic number", 64 bytes a la tabla de particiones y 446 bytes al bootloader. El bootloader puede contener la información de distintos sistemas.
En GNU/Linux, existen dos bootloaders principales: LILO y GRUB. LILO sólo soporta hasta 16 sistemas operativos instalados y no tiene la posibilidad de iniciar alguno desde la red. No contiene una consola interactiva que permita modificar los parámetros de inicio de un determinado sistema operativo y, por último, si luego de instalado nuestro sistema, realizamos un cambio que requiera modificaciones del bootloader, será necesario modificar los archivos de configuración de LILO y reescribir el booloader en el MBR.
GRUB es uno de los bootloaders más utilizados para sistemas operativos GNU/Linux. Las características más importantes son la posibilidad de manejar una cantidad ilimitada de sistemas operativos y de bootear por red, contiene una interfaz de línea comandos interactiva que permite establecer parámetros al iniciar un sistema operativo y, por último, si se realiza un cambio que requiera modificaciones en la configuración de GRUB, solamente es necesario modificar cambiar los archivos de configuración sin reescribir el MBR.

Tercera etapa: Kernel

El proceso de carga del kernel se realiza en dos etapas: etapa de carga y etapa de ejecución. La etapa de carga del kernel se encargará de descomprimir el kernel y copiarlo entero en la memoria principal (RAM, se cargarán los drivers necesarios

mediante un proceso llamado initrd. Este proceso creará un sistema de archivos temporal que sólo es utilizado durante la fase de carga.

Cuarta etapa: init

Un proceso es un programa que se ejecuta en un determinado momento en el sistema. Cada proceso contiene un conjunto de estructuras de datos y una dirección en la memoria principal. Existen dos tipos de procesos, los procesos de usuario, que son aquellos procesos ejecutados directamente por el usuario, y los procesos demonio. Los procesos demonio son aquellos que no requieren la intervención del usuario y se ejecutan en un segundo plano.
Una vez que la etapa de carga y ejecución del kernel se completa, se iniciará el proceso init. Este proceso es ejecutado por todos los sistemas basados en Unix y es el responsable de la inicialización de todos los nuevos procesos excepto el proceso swapper. Init se conoce como un proceso *dispatcher* o planificador, encargado de decir qué proceso se ejecutará y cuáles serán copiados/borrados de la memoria principal. A partir del momento de ejecución de init, éste es conocido como el proceso 1.

Manejo de procesos

Las estructuras de datos referidas a los procesos contienen información que permite el manejo de éstos. Algunos de los datos que contienen son: mapa de espacio del proceso, estado actual, prioridad de ejecución, máscara actual de la señal del proceso y propietario. Salvo el proceso *init* que tiene el PID 1, todos los demás procesos son creados por otros procesos.

Atributos de un proceso

Algunos de los parámetros asociados a los procesos afectan de forma directa a su ejecución.
PID (Process IDentificacion). Es un número que identifica al proceso en el sistema.
PPID (Parent Process IDentification). Es un número que se corresponde con el PID del proceso <padre>. El proceso <padre> es aquel que creó al proceso actual, llamado proceso <hijo> del anterior.
UID, GID (User Identification, Group IDentification). Estos números ya aparecieron en la unidad anterior. Son el número de identificación del usuario que creó el proceso UID, y el número de identificación del grupo de usuario, GID. Sólo el superusuario y el usuario que creó el proceso, llamado propietario del proceso, pueden modificar el estado de operación de los procesos.
EUID, EGID (Effective User IDentification, Effective Group IDentification). Estos números identifican al usuario que ejecuta el proceso; es atráves de estos números y no del UID y GID cómo el sistema determina para qué ficheros el proceso tiene acceso. El propietario y el proceso mismo pueden modificar la prioridad, pero siempre en sentido decrecimiento sólo el superusuario no tiene restricciones para cambiar la prioridad de un proceso.
Control de terminal. Son enlaces que determinan de dónde toman la entrada y la salida de datos y el canal de error los procesos durante su ejecución.

Creación y ejecución de un proceso

Cuando un proceso quiere crear un nuevo proceso, el primer paso consiste en realizar una copia de sí mismo mediante la llamada del sistema fork. La operación fork crea una copia idéntica del proceso padre, salvo en los siguientes casos:

- El nuevo proceso tiene un PID distinto y único.
- El PPID del nuevo proceso es el PID del proceso padre.
- Se asume que el nuevo proceso no ha usado recursos.
- El nuevo proceso tiene su propia copia de los ficheros descriptores del proceso padre.

Estado de los procesos

Existen cinco estados posibles que puede adoptar un proceso:

- *Ejecutándose (running R).* El proceso se está ejecutando.
- *Durmiendo (sleeping S):* Esperan a que ocurra un determinado evento.

- *Intercambiado (swapped).* El proceso no está en memoria.
- *Zombi (Zombie Z).* El proceso trata de finalizar su ejecución.
- *Parado (stopped).* El proceso no puede ser ejecutado, se puede para desde un Shell csh se pulsa Ctrl-Z, a petición de un usuario o proceso y cuando un proceso que se ejecuta en segundo plano trata de accedes a un terminal.

Señales

Las señales se utilizan para que un proceso suspenda la tarea que está realizando y se ocupe de otra. Cuando se envía una señal a un proceso, éste puede actuar de dos maneras: si el proceso dispone de una rutina específica para esa señal, llamada "manejador" o *handler*, la utiliza, en caso contrario, el Kernel utiliza el manejador por defecto para esa señal. Utilizar un manejador específico para una señal en un proceso se denomina capturar la señal.
Hay dos señales que no pueden ser ni capturadas ni ignoradas, *KILL* y *STOP*. La señal de KILL hace que el proceso que la recibe sea destruido. Un roceso que recibe la señal de STOP suspende su ejecución hasta reciba una señal CONT.

.Número	Nombre	Descripción	Por defecto	¿Capturada?	¿Bloqueada?
1	SIGHUP	Hangup.	Terminar	SI	SI
2	SIGINT	Interrumpir.	Terminar	SI	SI
3	SIGQUIT	Salir.	Terminar	SI	SI
4	SIGILL	Instrucción ilegal.	Terminar	SI	SI
5	SIGTRAP	Trazar.	Terminar	SI	SI
6	SIGIOT	IOT.	Terminar	SI	SI
7	SIGBUS	Error de de Bus.	Terminar	SI	SI
8	SIGFPE	Excepción aritmética.	Terminar	SI	SI
9	SIGKILL	Destruir	Terminar	NO	NO
10	SIGUSR1	Primera señal definida por el usuario.	Terminar	SI	SI
11	SIGSEGV	Violación de segmentación.	Terminar	SI	SI
12	SIGUSR2	Segunda señal definida por el usuario.	Terminar	SI	SI
13	SIGPIPE	Escribir en un pipe.	Terminar	SI	SI
14	SIGALRM	Alarma del reloj.	Terminar	SI	SI
15	SIGTERM	Terminación del programa.	Terminar	SI	SI
16	SIGSTKFLT		SI		SI
17	SIGCHLD	El estado del hijo ha cambiado.	Ignorar	SI	SI
18	SIGCONT	Continuar después de parar.	Ignorar	SI	NO
19	SIGSTOP	Parar.	Parar	NO	NO
20	SIGSTP	Parada desde el teclado.	Parar	SI	SI
21	SIGTTIN	Lectura en segundo plano.	Parar	SI	SI
22	SIGTTOU	Escritura en segundo plano.	Parar	SI	SI

23	SIGURG	Condición urgente de socket.	Ignorar	SI	SI
24	SIGXCPU	Excedido el tiempo de CPU.	Terminar	SI	SI

Enviar señales a un proceso

Sólo el propietario del proceso y el superusuario (root) pueden mandar señal KILL a un proceso.

USUARIOS, GRUPOS Y PERMISOS

Linux fue diseñado para permitir múltiples usuarios al mismo tiempo. Para que estos usuarios no afecten las sesiones del resto de los usuarios, se tuvo que desarrollar un complejo modelo de permisos.

Permisos de lectura, escritura y ejecución

Los permisos son los "derechos" que tiene un usuario o grupo sobre un archivo o directorio. Los permisos básicos son *lectura, escritura y ejecución.*

• *Permiso de lectura (read).* Este permiso hace que el contenido de un archivo sea visible, o, en el caso de los directorios, el que contenido del mismo sea visible.
• *Permiso de escritura (write)*. Este permiso hace que el contenido de un archivo pueda ser modificado, mientras que para directorios permite realizar acciones sobre los archivos que contiene (borrarlos, agregar nuevos archivos, etc.)
• *Permiso de ejecución (execute)*. En archivos, este permiso hará posible la ejecución del mismo. Para esto, el archivo deberá ser un *script* o un programa.

Listar los permisos de un archivo o directorio

Para listar los permisos de un archivo o directorio, basta con ejecutar el comando ***ls***, junto con la opción ***-l***. El resultado de la ejecución de dicho comando mostrará el contenido de un directorio de la siguiente manera:

-rw-r--r-- 1 root root 1031 Nov 18 09:22 /etc/passwd

Los primeros diez caracteres del ejemplo representan los permisos. El primer carácter (que en este caso es un -) representa el tipo de archivo que está listando. Los tipos de archivos son:

• d representa directorios

• s representa un archivo especial

• - representa un archivo regular

Administrando cuentas de usuario

El comando para crear un nuevo usuario estándar es ***useradd***. La sintaxis es la siguiente:

useradd <nombre>

Las opciones que podemos utilizer son las siguientes:

Opción	Descripción
-d <home_dir>	Nos permite especificar cuál será el directorio principal del usuario que estamos creando.
-e <fecha>	Permite especificar una fecha en la que expirará la cuenta del nuevo usuario.
-s	Especifica el intérprete de línea de comandos que usará por defecto el nuevo usuario. Por ejemplo: *useradd nuevousuario -s /bin/bash*

Una vez creado el nuevo usuario, es necesario configurar un password para el mismo. Esto se hace utilizando el comando ***passwd***. La sintaxis es la siguiente:

passwd <usuario>

Para remover una cuenta de usuario, el comando a utilizar es ***userdel***. La sintaxis del mismo es:

userdel <usuario>

Para poder borrar un usuario completamente, el comando deberá ser ejecutado utilizando la opción ***-r***.

userdel -r <usuario>

Entendiendo *sudo*

Para que un usuario determinado sea pueda utilizar ***sudo***, será necesario que el mismo esté presente en el archivo ***sudoers***. Este archivo se encuentra ubicado dentro del directorio ***/etc.***
Es importante tener en cuenta que, para agregar un usuario a la lista de ***sudoers*** (agregarlo al archivo), es necesario tener permisos *root*.

Trabajando con grupos

Los grupos en Linux son una manera de organizar los usuarios, principalmente como una medida de seguridad, a través del archivo ***/etc/group***, que contiene la lista de grupos y sus miembros. Cada usuario tiene un grupo primario por defecto, cada vez que el usuario cree un archivo, o ejecute un programa, esto quedará asociado al grupo primario. Para cambiar el grupo con el que un usuario está asociado en un momento determinado, se utiliza el comando ***chgrp***.

Cambiando permisos de archivos y directorios

Comando *chmod*

El comando ***chmod*** nos permitirá cambiar permisos en archivos o directorios. Existen dos maneras de de especificar nuevos permisos al utilizar el comando ***chmod***, con letras o números (en base octal).
Para utilizar el comando ***chmod*** con letras, hay que tener en cuenta lo siguiente: El signo ***+*** agregará permisos y el signo ***–*** quitará permisos.

- ***r*** representa el permiso de lectura.
- ***w*** representa el permiso de escritura.
- ***x*** representa el permiso de ejecución.
- ***X*** representa el permiso de ejecución (sólo aplica a directorios).

Valor octal	Lectura (r)	Escritura (w)	Ejecución (x)
7	r	w	x
6	r	w	-
5	r	-	x
4	r	-	-
3	-	w	x
2	-	w	-
1	-	-	x
0	-	-	-

Permisos adicionales en archivos

Estos permisos adicionales son el *sticky bit (t)* y *setuid bit (s)*. Estos dos permisos describen el comportamiento de los archivos y ejecutables en situaciones "multiusuario".
Cuando se define en un archivo o directorio, el modo ***sticky bit***, significa que solo el propietario (o *root*) pueden eliminar el archivo.
chmod +t <archivo>

El permiso ***setuid*** se puede explicar este comportamiento es similar a la opción "ejecutar como", que permite ejecutar un programa como si fuera un usuario distinto.

Cambiando el propietario de un archivo

Por defecto, el propietario de un archivo es aquel que crea el mismo, y por defecto, el grupo asociado es grupo en que esté registrado el usuario al momento de crear el archivo. Para cambiar el propietario de un archivo, se utiliza el comando ***chown***.