

Memorias

***Arquitectura y Sistemas Operativos.
Tecnicatura Superior en Programación.
UTN-FRA***

Autores: *Prof. Martín Isusi Seff*

Revisores: *Prof. Marcos Pablo Russo*

Versión: 1



Esta obra está bajo una [Licencia Creative Commons Atribución-CompartirIgual 4.0 Internacional](https://creativecommons.org/licenses/by-sa/4.0/).

John Von Neumann, matemático y físico austríaco definió muchos de los principios que hoy rigen en el diseño de las computadoras. Uno de esos principios expresa que un programa debe estar almacenado en una memoria para poder ser ejecutado por la computadora.

Las computadoras tienden a acceder al almacenamiento en formas particulares. De hecho, la mayoría del acceso a almacenamiento tiende a exhibir uno (o ambos) de los siguientes atributos:

- El acceso tiende a ser secuencial
- El acceso tiende a ser localizado

El acceso secuencial significa que, si el CPU accede a la dirección N, es muy probable que la dirección N+1 sea la próxima a acceder. Esto tiene sentido, ya que muchos programas consisten de grandes secciones de instrucciones que ejecutan — en orden — una instrucción tras la otra.

El acceso localizado significa que, si se accede a la dirección X, es muy probable que otras direcciones alrededor de X también serán accedidas en el futuro.

Estos atributos son cruciales, debido a que permite que unidades de almacenamiento pequeña y más rápida, coloque efectivamente en memoria temporal almacenamiento más grande y lento. Esto es lo básico para implementar la memoria virtual. Pero antes de que discutamos la memoria virtual, debemos examinar las diferentes tecnologías de almacenamiento usadas actualmente.

Distintos tipos de memorias

Las computadoras de hoy utilizan una variedad de tecnologías de almacenamiento. Cada tecnología está orientada hacia una función específica, con velocidades y capacidades en combinación. Estas tecnologías son:

- Registros de CPU
- Memoria caché
- RAM
- Discos duros
- Almacenamiento fuera de línea para respaldos (cintas, discos ópticos, etc.)

En términos de capacidades y costos, estas tecnologías forman un espectro. Por ejemplo, los registros de CPU son:

- Muy rápidos (tiempos de acceso de unos pocos nanosegundos)
- Baja capacidad (usualmente menos de 200 bytes)
- Capacidades de expansión muy limitadas (se requiere un cambio en la arquitectura del CPU)
- Costosas

Sin embargo, en el otro lado del espectro, el almacenamiento fuera de línea es:

- Muy lento (tiempos de acceso se miden en días, si la media de respaldo debe ser entregada sobre largas distancias)
- Capacidad muy alta (10s - 100s de gigabytes)
- Capacidades de expansión prácticamente ilimitadas (solamente limitadas por el espacio físico requerido para hospedar la media de respaldo)
- Bajo costo

Usando diferentes tecnologías con diferentes capacidades, es posible afinar el diseño del sistema para un máximo rendimiento al costo más bajo posible. Las secciones siguientes exploran cada tecnología en el espectro del almacenamiento.

Registros de la CPU

Todos los diseños de CPU de hoy día incluyen registros para una variedad de propósitos, desde el almacenamiento de direcciones de la instrucción recientemente ejecutada, hasta propósitos más generales de almacenamiento y manipulación de datos. Los registros de CPU se ejecutan a la misma velocidad que el resto del CPU; de lo contrario habría un cuello de botella grave sobre el rendimiento completo del sistema. La razón para esto es que casi todas las operaciones realizadas por el CPU envuelven registros de una forma u otra.

El número de registros de CPU (y sus usos) dependen estrictamente en el diseño arquitectónico del CPU mismo. No hay forma de cambiar el número de registros de CPU, solamente puede migrar a un CPU con una arquitectura diferente. Por estas razones, el número de registros de CPU se puede considerar como una constante, ya que sólo pueden cambiarse con mucho dolor y grandes costos.

Memoria caché

El propósito de la memoria caché es actuar como una memoria temporal entre los registros de CPU, limitados y de gran velocidad y el sistema de memoria principal, mucho más grande y lento — usualmente conocido como RAM¹. La memoria caché tiene una velocidad de operación similar a la del CPU mismo, por eso cuando el CPU accede a datos en la caché, no tiene que quedarse esperando por los datos.

La memoria caché es configurada de forma tal que, cuando se leen datos desde la RAM, el sistema de hardware verifica primero para determinar si los datos deseados están en caché. Si los datos están en caché, estos son recuperados rápidamente y utilizados por el CPU. Sin embargo, si los datos no están en caché, estos se leen desde la RAM y, mientras se transfieren

¹ Mientras que "RAM" es un acrónimo para "Random Access Memory," y un término que podría ser fácilmente aplicable a cualquier tecnología de almacenamiento que permita el acceso no secuencial de los datos almacenados, cuando los administradores de sistemas hablan sobre RAM invariablemente se refieren al sistema de memoria principal.

al CPU, también se colocan en caché (en caso de que se necesiten más tarde). Desde la perspectiva del CPU, todo esto se hace de forma transparente, por lo que 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.

En términos de la capacidad de almacenamiento, la caché es mucho más pequeña que la RAM. Por lo tanto, no todos los bytes en la RAM tienen su ubicación única en caché. Como tal, es necesario dividir la caché en secciones que se puedan utilizar para alojar diferentes áreas de RAM y tener un mecanismo que permita que cada área de la caché haga un "caché" de la RAM en diferentes momentos. Aunque existe una diferencia en tamaño entre la caché y la RAM, dada la naturaleza secuencial y localizada del acceso a almacenamiento, una pequeña cantidad de caché puede efectivamente acelerar el acceso a grandes cantidades de RAM.

Cuando se escriben datos desde el CPU, las cosas se complican un poco. Existen dos enfoques que se pueden utilizar. En ambos casos, los datos son escritos primero a la caché. Sin embargo, puesto que el propósito de la caché es funcionar como una copia muy rápida de los contenidos de porciones seleccionadas de RAM, cada vez que una porción de datos cambia su valor, ese nuevo valor debe ser escrito tanto a la caché como a la RAM. De lo contrario, los datos en caché y los datos en la RAM ya no coincidirían.

Los dos enfoques se diferencian en cómo se logra hacer esto. En un enfoque, conocido como write-through caching, los datos modificados se escriben inmediatamente a la RAM. Sin embargo, en el write-back caching, se retrasa la escritura de los datos modificados a la RAM. La razón para hacer esto es la de reducir el número de veces que una porción de datos modificada frecuentemente debe ser escrita nuevamente a la RAM.

La caché "write-through" o inmediata es un poco más simple de implementar; por esta razón es la más común. La caché "write-back" es un poco más complicada; además de almacenar los datos, es necesario mantener cierto tipo de mecanismo que sea capaz de notificar que los datos en caché están al día o "limpios" (los datos en caché son los mismos que los datos en RAM), o que están "sucios" (los datos en caché han sido modificados, lo que significa que los datos en RAM ya no están actualizados). También es necesario implementar una forma de vaciar periódicamente entradas "sucias" en caché de vuelta en RAM.

Niveles de caché

Los subsistemas de caché en los diseños de computadoras de hoy día pueden ser de niveles múltiples; esto es, puede haber más de un conjunto de caché entre el CPU y la memoria principal. Los niveles de caché a menudo están enumerados, con los números menores más cercanos a la CPU. Muchos sistemas tienen dos 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.

En cualquier caso, el objetivo de todos los subsistemas de caché — bien sean simples o de múltiples niveles — es el de reducir el tiempo de acceso promedio a la RAM.

Memoria principal – RAM

La RAM resuelve la mayoría del almacenamiento electrónico en las computadoras de hoy en día. 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 y está mucho más cercana a la velocidad de la primera que a la segunda.

La operación básica de la RAM es en realidad bien sencilla. 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 (para operar la circuitería dentro del chip)
- Conexiones de datos (para permitir la transferencia de datos hacia adentro y fuera del chip)
- Conexiones de lectura/escritura (para controlar si los datos se almacenaran o se recuperaran desde el chip)
- Conexiones de direcciones (para determinar si los datos en el chip serán leídos/escritos)

He aquí los pasos requeridos para almacenar datos en RAM:

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

La recuperación de datos es también muy directa:

- La dirección de los datos deseados se presenta a las conexiones de direcciones.
- La conexión de lectura/escritura es colocada a modo de lectura.
- Los datos deseados son leídos desde las conexiones de datos.

Mientras que estos pasos parecen bastante simples, estos toman lugar a grandes velocidades, con el tiempo consumido en cada paso medido en nanosegundos.

Casi todos los chips de memoria creados hoy en día se venden como módulos. Cada módulo consiste de un número individual de chips de RAM conectados a una pequeña tarjeta de circuitos. La distribución mecánica y eléctrica del módulo sigue los estándares de la industria,

haciendo posible la compra de memorias de
sde diferentes fabricantes.

Discos duros

Todas las tecnologías discutidas hasta ahora son volátiles por naturaleza. En otras palabras, los datos contenidos en almacenamiento volátil se pierden cuando se desconecta el poder.

Por otro lado, los discos duros son no-volátiles — lo que significa que los datos se mantienen allí, aún después que se ha desconectado la energía. Debido a esto, los discos duros ocupan un lugar muy especial en el espectro del almacenamiento. Su naturaleza no-volátil los hace ideal para el almacenamiento de programas y datos para su uso a largo plazo. Otro aspecto único de los discos duros es que, a diferencia de la RAM y la memoria caché, no es posible ejecutar los programas directamente cuando son almacenados en discos duros; en vez de esto, ellos deben primero ser leídos a la RAM.

Otra diferencia de la caché y de la RAM es la velocidad del almacenamiento y recuperación de los datos; los discos duros son de al menos un orden de magnitud más lento que todas las tecnologías electrónicas utilizadas para caché y RAM. La diferencia en velocidad es debida principalmente a su naturaleza electromecánica. Hay cuatro fases distintas que toman lugar durante cada transferencia de datos desde o hacia un disco duro. La lista siguiente ilustra estas fases, junto con el tiempo que en promedio tomaría una unidad de alto rendimiento, para completar cada fase:

- Movimiento del brazo de acceso (5.5 milisegundos)
- Rotación del disco (.1 milisegundos)
- Lectura/escritura de datos de cabezales (.00014 milisegundos)
- Transferencia de datos hacia/desde la electrónica del disco (.003 Milisegundos)

Fundamentos de la memoria virtual

Aun cuando la tecnología detrás de la construcción de las implementaciones modernas de almacenamiento es realmente impresionante, el administrador de sistemas promedio no necesita estar al tanto de los detalles. De hecho, solamente existe un factor que los administradores de sistemas deberían tener en consideración: Nunca hay suficiente RAM.

Mientras que esta frase puede sonar al principio un poco cómica, muchos diseñadores de sistemas operativos han empleado una gran cantidad de tiempo tratando de reducir el impacto de esta limitación. Esto lo han logrado mediante la implementación de la memoria virtual — una forma de combinar RAM con un almacenamiento más lento para darle al sistema la apariencia de tener más RAM de la que tiene instalada realmente.

Memoria virtual en términos sencillos

Vamos a comenzar con una aplicación hipotética. El código de máquina que conforma esta

aplicación tiene un tamaño de 10000 bytes. También requiere otros 5000 bytes para el almacenamiento de datos y para la memoria intermedia de E/S. Esto significa que, para ejecutar la aplicación, deben haber más de 15000 bytes de RAM disponible; un byte menos y la aplicación no será capaz de ejecutarse.

Este requerimiento de 15000 bytes se conoce como el espacio de direcciones de la aplicación. Es el número de direcciones únicas necesarias para almacenar la aplicación y sus datos. En las primeras computadoras, la cantidad de RAM disponible tenía que ser mayor que el espacio de direcciones de la aplicación más grande a ejecutar; de lo contrario, la aplicación fallaría con un error de "memoria insuficiente".

Un enfoque posterior conocido como solapamiento intentó aliviar el problema permitiendo a los programadores dictar cuales partes de sus aplicaciones necesitaban estar residentes en memoria en cualquier momento dado. De esta forma, el código requerido solamente para propósitos de inicialización podía ser sobrescrito con código a utilizar posteriormente. Mientras que el solapamiento facilitó las limitaciones de memoria, era un proceso muy complejo y susceptible a errores. El solapamiento también fallaba en solucionar el problema de las limitaciones de memoria globales al sistema en tiempo de ejecución. En otras palabras, un programa con solapamiento requería menos memoria para ejecutarse que un programa sin solapamiento, pero si el sistema no tiene suficiente memoria para el programa solapado, el resultado final es el mismo — un error de falla de memoria.

Con la memoria virtual el concepto del espacio de direcciones de las aplicaciones toma un significado diferente. 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?".

Aunque inicialmente pareciera que nuestra aplicación hipotética requiere de un total de 15000 bytes para ejecutarse — el acceso a memoria tiende a ser secuencial y localizado. Debido a esto, la cantidad de memoria requerida para ejecutar la aplicación en un momento dado es menos que 15000 bytes — usualmente mucho menos. Considere los tipos de accesos de memoria requeridos para ejecutar una instrucción de máquina sencilla:

- La instrucción es leída desde la memoria.
- Se leen desde memoria los datos requeridos por la instrucción.
- Después de completar la instrucción, los resultados de la instrucción son escritos nuevamente en memoria.

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. Sin embargo, aún si una instrucción requiere de 100 bytes de memoria por cada tipo de acceso de memoria, los 300 bytes requeridos son mucho menos que el espacio de direcciones total de la aplicación de

15000 bytes. Si hubiese una forma de hacer un seguimiento de los requerimientos de memoria de la aplicación a medida que esta se ejecuta, sería posible mantener la aplicación ejecutándose usando menos memoria que lo que indicaría su espacio de direcciones.

Pero esto genera una pregunta: Si solamente una parte de la aplicación está en memoria en un momento dado, ¿dónde está el resto?

La respuesta corta a esta pregunta es que el resto de la aplicación se mantiene en disco. En otras palabras, 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. Esto puede parecer al principio como un gran problema de rendimiento en su creación — después de todo, las unidades de disco son mucho más lentas que la RAM.

Aunque esto es cierto, es posible tomar ventaja del comportamiento de acceso secuencial y localizado de las aplicaciones y eliminar la mayoría de las implicaciones de rendimiento en el uso de unidades de disco como unidades de respaldo para la RAM. Esto se logra estructurando el subsistema de memoria virtual para que este trate de asegurarse de que esas partes de la aplicación que actualmente se necesitan — o que probablemente se necesitaran en un futuro cercano — se mantengan en RAM solamente por el tiempo en que son realmente requeridas.

En muchos aspectos, esto es similar a la situación entre la caché y la RAM: haciendo parecer una poca cantidad de almacenamiento rápido con grandes cantidades de un almacenamiento lento actuar como que si se tratase de grandes cantidades de almacenamiento rápido.

Con esto en mente, exploremos el proceso en más detalle.

Memoria virtual: Los detalles

Primero, debemos introducir un nuevo concepto: espacio de direcciones virtuales. El espacio de direcciones virtuales es el espacio de direcciones máximo disponible para una aplicación. El espacio de direcciones virtuales varía de acuerdo a la arquitectura del sistema y del sistema operativo. El espacio de direcciones virtuales depende de la arquitectura puesto que es la arquitectura la que define cuántos bits están disponibles para propósitos de direccionamiento. El espacio de direcciones virtuales también depende del sistema operativo puesto que la forma en que el sistema operativo fue implementado puede introducir límites adicionales sobre aquellos impuestos por la arquitectura.

La palabra "virtual" en el espacio de direcciones virtuales, significa que este es el número total de ubicaciones de memoria direccionables disponibles para una aplicación, pero no la cantidad de memoria física instalada en el sistema, o dedicada a la aplicación en un momento dado.

En el caso de nuestra aplicación de ejemplo, su espacio de direcciones virtuales es de 15000 bytes.

Para implementar la memoria virtual, para el sistema 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. Esto significa que la dirección de memoria 123 puede ser redirigida a la dirección física 82043 en un momento dado y a la dirección 20468 en otro. 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.

Mantener un seguimiento de estas páginas y sus direcciones traducidas puede sonar como un paso adicional confuso e innecesario, pero de hecho es crucial para la implementación de la memoria virtual. Por tal razón, considere el punto siguiente:

Tomando nuestra aplicación hipotética con un espacio de direcciones virtuales de 15000 bytes, asuma que la primera instrucción de la aplicación accede a los datos almacenados en la dirección 12374. Sin embargo, también asuma que nuestra computadora solamente tiene 12288 bytes de RAM física. ¿Qué pasa cuando el CPU intenta acceder a la dirección 12374?

Lo que ocurre se conoce como un fallo de página.

Memoria secundaria. Conceptos fundamentales y administración

Anteriormente, en este capítulo se habló del concepto de memoria principal. Aquí se almacenan aquellos programas que se deben ser ejecutados por la computadora. Si bien la memoria virtual basa su funcionamiento en el disco rígido que es no volátil, la memoria principal sí es volátil. Esto no sucede en la memoria secundaria: 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.

Los discos rígidos son el dispositivo más utilizado como memoria secundaria, aunque hoy en día existen alternativas. La alternativa principal es el almacenamiento de estado sólido (conocido como “disco de estado sólido” o SSD, por sus siglas en inglés Solid State Disk). Si bien este no es un disco rígido, ya que no posee platos metálicos ni ningún tipo de parte mecánica, puede ser manejado como un disco rígido regular.

Una vez que el disco principal está instalado, poco se puede hacer con él sin antes establecer particiones y especificar un sistema de archivos.

¿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

por el sistema operativo como discos independientes. El disco rígido, necesitará lo que se conoce como “tabla de particiones”, para poder especificar qué sectores físicos del disco corresponden a cada partición.

Muchas veces, conociendo el concepto de partición surge la pregunta ¿Por qué tener múltiples particiones?

- Tener múltiples particiones permite “encapsular” los datos. Esto quiere decir que cada partición tendrá su propio sistema de archivos, que será independiente del resto de las particiones. En caso de corrupción, la pérdida de datos se limita a esa partición.
- Múltiples particiones pueden tener formatos distintos, variando, por ejemplo, el tipo de sistema de archivos o la cantidad de bloques en la que se divide lógicamente el disco. 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. Por ejemplo, podríamos realizar una partición en la que se almacene el sistema operativo y una partición para los datos del usuario. Si los datos del usuario completan el tamaño disponible en su partición, esto no hará que el sistema operativo se quede sin espacio para trabajar.

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. Una partición extendida puede contener dentro de sí múltiples particiones, extendiendo el número de particiones posibles en una sola unidad de disco. La introducción de las particiones extendidas se generó por el desarrollo constante de las capacidades de los discos duros.

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

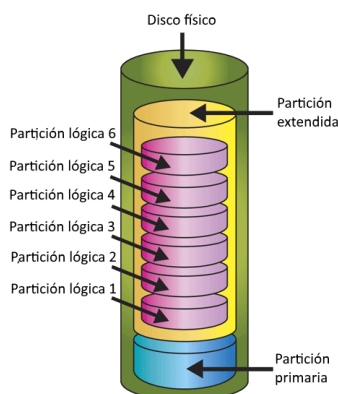


Figura 1 Diagrama representativo de los distintos tipos de particiones.

Sistema de archivos

Aun teniendo el dispositivo de almacenamiento configurado y particionado correctamente, sería difícil almacenar y recuperar información — todavía nos falta una forma de estructurar y organizar esa información. Lo que necesitamos es un sistema de archivos.

El concepto de un sistema de archivos es tan fundamental para el uso de los dispositivos de almacenamiento masivo que el usuario de computadoras promedio ni siquiera hace una distinción entre los dos. Sin embargo, los administradores de sistemas no se pueden permitir ignorar los sistemas de archivos y su impacto en el trabajo diario.

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 basado en archivos
- Estructura de directorio jerárquico (algunas veces llamado "carpeta")
- 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 específico
- 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.

Primero debe estar consciente de cualquier restricción de nombres. Por ejemplo, ¿cuáles son los caracteres permitidos en un nombre de archivo? ¿Cuál es el largo máximo para un nombre de archivo? Estas preguntas son importantes, pues dictan cuales nombres de archivos se

pueden utilizar y cuáles no. Los sistemas operativos más antiguos con sistemas de archivos más primitivos permitían solamente caracteres alfanuméricos (y solamente mayúsculas) y únicamente nombres de archivos 8.3 (lo que significa un nombre de archivo de ocho caracteres, seguido de una extensión de tres caracteres).

Estructura de directorios jerárquico

Mientras que los sistemas de archivos en ciertos sistemas operativos antiguos no incluían el concepto de directorios, todos los sistemas de archivos de hoy día incluyen esta característica. 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. Este es un concepto poderoso con el cual muchos administradores de sistemas deberían de estar familiarizados. Usando las jerarquías de múltiples niveles puede hacer la administración de archivos mucho más fácil para usted y sus usuarios.

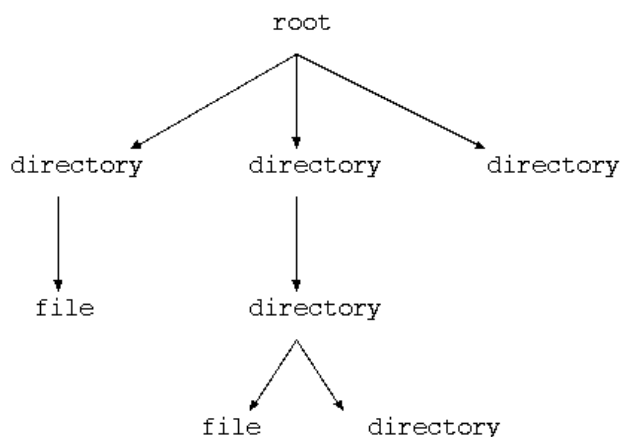


Figura 2 En una estructura de directorios jerárquicos, tendremos un directorio principal dentro del cual se ubicarán otros directorios o incluso archivos. Más adelante se verá la estructura de directorios utilizada en sistemas operativos GNU/Linux

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. Más allá de la conveniencia de poder determinar cuándo un archivo dado fue creado, accedido o modificado, estas fechas son vitales para la operación adecuada de los respaldos incrementales.

Control de acceso

El control de acceso es un área en la que los sistemas de archivos difieren dramáticamente. Algunos sistemas de archivos no tienen un modelo claro para el control de acceso, mientras que otros son mucho más sofisticados. En términos generales, 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 (y el sistema operativo subyacente) primeramente debe ser capaz de identificar unívocamente a usuarios individuales. Esto hace posible tener una responsabilidad completa con respecto a cualquier operación a nivel de sistema de archivos. Otra funcionalidad de ayuda es la de los grupos de usuarios. Se utilizan grupos más a menudo en organizaciones donde los usuarios pueden ser miembros de uno o más proyectos. Otra funcionalidad que algunos sistemas de archivos soportan es la creación de identificadores genéricos que se pueden asignar a uno o más usuarios.

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 el archivo
- Escribir al archivo
- Ejecutar el archivo

Varios sistemas de archivos pueden extender la lista para incluir otras acciones tales como eliminar, o hasta la habilidad de hacer cambios al control de acceso del archivo.

Contabilidad del espacio utilizado

Una constante en la vida de un administrador de sistemas es la de que nunca hay suficiente espacio libre, y aún si lo hubiese, no estará disponible por mucho tiempo. Por lo tanto, 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.

Esta característica es vital en grandes entornos de usuarios, pues es un hecho que la regla de 80/20 se aplica a menudo al espacio en disco — 20 por ciento de sus usuarios serán responsables por el consumo de 80 por ciento de su espacio disponible en disco. Al facilitar la identificación de estos usuarios en el 20 por ciento, puede más efectivamente manejar sus activos relacionados al almacenamiento.

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. Los detalles específicos varían de un sistema de archivos al otro, pero en general a cada usuario se le puede asignar una cantidad

específica de almacenamiento que un usuario puede utilizar. Más allá de allí, los sistemas de archivos varían. 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

Aunque todo lo que se ha presentado hasta ahora en este capítulo solamente trata con discos duros únicos conectados directamente a un sistema, existen otras opciones más avanzadas.

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

El almacenamiento se puede consolidar implementando servidores de alto rendimiento con conexiones de red de alta velocidad y configurados con grandes cantidades de almacenamiento rápido. Con la configuración apropiada, es posible suministrar acceso al almacenamiento a velocidades comparables al almacenamiento conectado directamente. Más aún, la naturaleza compartida de tal configuración a menudo hace posible reducir los costos, ya que los gastos asociados con suministrar almacenamiento centralizado y compartido pueden ser menores que suministrar el almacenamiento equivalente para cada uno de los clientes. Además, 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. Por ejemplo, monitorizar el espacio libre es mucho más fácil cuando el almacenamiento a supervisar existe en un sólo servidor centralizado. Los respaldos también se pueden simplificar en gran medida usando un servidor de almacenamiento centralizado. Es posible hacer respaldos basados en la red para múltiples clientes, pero se requiere más trabajo para configurar y mantener.

Existen varias tecnologías disponibles de almacenamiento en red; seleccionar una puede ser complicado. Casi todos los sistemas operativos en el mercado hoy día incluyen alguna forma de acceder a almacenamiento en red, pero las diferentes tecnologías son incompatibles entre ellas.

Almacenamiento basado en RAID

Una habilidad que un administrador de sistemas debería cultivar es la de hacer las complejas configuraciones de sistemas y observar las diferentes limitaciones inherentes a cada

configuración. Mientras que esto, a primera vista, puede parecer un punto de vista deprimente a tomar, puede ser una forma excelente de ver más allá de las brillantes cajas nuevas y visualizar una futura noche del sábado con toda la producción detenida debido a una falla que se podría haber evitado fácilmente con pensarlo un poco.

Con esto en mente, utilicemos lo que conocemos sobre el almacenamiento basado en discos y veamos si podemos determinar las formas en que los discos pueden causar problemas. Primero, considere una falla absoluta del hardware:

Un disco duro con cuatro particiones se muere completamente: ¿qué pasa con los datos en esas particiones?

Está indisponible inmediatamente (al menos hasta que la unidad dañada pueda ser reemplazada y los datos restaurados desde el respaldo más actual).

Un disco duro con una sola partición está operando en los límites de su diseño debido a cargas de E/S masivas: ¿qué les pasa a las aplicaciones que requieren acceso a los datos en esa partición?

Las aplicaciones se vuelven más lentas debido a que el disco duro no puede procesar lecturas y escrituras más rápido.

Tiene un archivo de datos que poco a poco sigue creciendo en tamaño; pronto será más grande que el disco más grande disponible en su sistema. ¿Qué pasa entonces?

La unidad de disco se llena, el archivo de datos deja de crecer y su aplicación asociada deja de funcionar.

Cualquiera de estos problemas puede dejar su centro de datos inútil, sin embargo, los administradores de sistemas deben enfrentarse a este tipo de problemas todos los días. ¿Qué se puede hacer?

Afortunadamente, existe una tecnología que puede resolver cada uno de estos problemas. El nombre de esta tecnología es 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 trataran de una sola unidad.

Las técnicas RAID primero fueron desarrolladas por investigadores en la Universidad de California, Berkeley a mitad de los 80. En ese tiempo, había una gran separación entre las unidades de disco de alto rendimiento utilizadas por las grandes instalaciones computacionales del momento, y las unidades de disco más pequeñas utilizadas por la joven industria de las computadoras personales. RAID se veía como el método para tener varios discos menos costosos sustituyendo una unidad más costosa.

² Cuando comenzaron las primeras investigaciones de RAID, el acrónimo venía de Redundant Array of *Inexpensive* Disks, pero con el paso del tiempo los discos "independientes" que RAID pretendía sustituir se volvieron más y más económicos, dejando la cuestión del costo un poco sin sentido.

Más aún, las formaciones RAID se pueden construir de varias formas, resultando en características diferentes dependiendo de la configuración final. Vamos a ver las diferentes configuraciones (conocidas como niveles RAID) en más detalles.

Niveles de RAID

Al principio, fueron definidos cinco niveles RAID y los nombraron del "1" al "5." Luego, otros investigadores y miembros de la industria del almacenamiento definieron niveles RAID adicionales. No todos los niveles RAID eran igualmente útiles; algunos eran de interés solamente para propósitos de investigación y otros no se podían implementar de una forma económica.

Al final, había 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 necesario tener 2 discos duros como mínimo. Por ejemplo, si tenemos dos discos que funciona a una velocidad alrededor de 20 Mb/s, al poner dos discos se duplicaría la velocidad es decir 40 Mb/s (2x20 Mb/s). Es una partición lógica cuyo tamaño es igual a la suma de los discos integrados en el sistema RAID.



Figura 3

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. No se mejora el rendimiento y no se suman el espacio de los discos como en RAID 0. El tipo de RAID 1 se llama comúnmente "mirroring" debido a que éste hace una simple copia del primer disco.



Figura 4

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.

Una formación RAID 5 debe consistir de al menos tres discos idénticos en tamaño (aunque se pueden utilizar más discos). Cada unidad está dividida en porciones y los datos se escriben a las porciones siguiendo un orden. Sin embargo, no cada porción está dedicada al almacenamiento de datos como en RAID 0. En cambio, en una formación con n unidades en ella, la enésima porción está dedicada a la paridad.

Las porciones que contienen paridad hacen posible recuperar los datos si falla una de las unidades en la formación. La paridad en la porción x se calcula matemáticamente combinando los datos desde cada porción x almacenado en todas las otras unidades en la formación. Si los datos en una porción son actualizados, la correspondiente porción de paridad debe ser

recalculada y actualizada también.

Esto también significa que cada vez que se escriben datos en la formación, al menos dos unidades son escritas a: la unidad almacenando los datos y la unidad que contiene la porción con la paridad.

Un punto clave a tener en mente es que las porciones de paridad no están concentradas en una sola unidad de la formación. En cambio, están distribuidas uniformemente a lo largo de todas las unidades. Aun cuando es posible dedicar una unidad específica para que contenga únicamente paridad (de hecho, esta configuración se conoce como RAID nivel 4), la actualización constante de la paridad a medida que se escriben datos a la formación significa que la unidad de paridad se podría convertir en un cuello de botella. Distribuyendo la información de paridad uniformemente a través de la formación, se reduce este impacto.

Sin embargo, es importante recordar el impacto de la paridad en la capacidad general de almacenamiento de la formación. Aun cuando la información de paridad se distribuye uniformemente a lo largo de todos los discos, la cantidad de almacenamiento disponible se reduce por el tamaño de un disco.

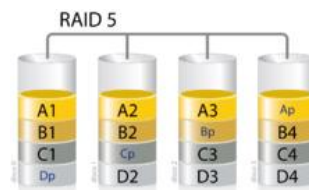


Figura 5

Niveles RAID anidados

Como debería ser obvio a partir de la discusión sobre los diferentes niveles de RAID, cada nivel tiene sus fortalezas y debilidades específicas. No fue mucho tiempo después de que se comenzó a implementar el almacenamiento basado en RAID que la gente comenzó a preguntarse si los diferentes niveles RAID se podrían combinar de alguna forma, produciendo formaciones con todas las fortalezas y ninguna de las debilidades de los niveles originales.

Por ejemplo, ¿qué pasa si los discos en una formación RAID 0 fuesen en verdad formaciones RAID 1? Esto proporcionaría las ventajas de la velocidad de RAID 0, con la confiabilidad de un RAID 1.

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

Debido a que los RAID anidados se utilizan en ambientes más especializados, no nos vamos a ir en más detalles aquí. Sin embargo, hay dos puntos a tener en mente cuando se piense sobre

RAID anidados:

- Otros aspectos - El orden en el que los niveles RAID son anidados pueden tener un gran impacto en la confiabilidad. En otras palabras, RAID 1+0 and RAID 0+1 no son lo mismo.
- Los costos pueden ser altos - Si hay alguna desventaja común a todas las implementaciones RAID, es el costo; por ejemplo, la formación RAID 5+1 más pequeña posible, consiste de seis discos (y se requieren hasta más discos para formaciones más grandes).

Ahora que ya hemos explorado los conceptos detrás de RAID, vamos a ver cómo se puede implementar RAID.

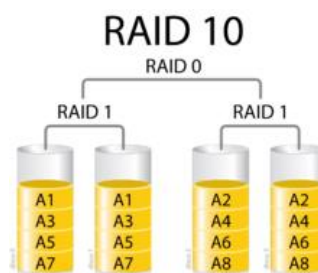


Figura 6 Imagen representativa de un RAID 1+0. Aquí vemos como se combinan distintos tipos de RAID, para generar una formación de discos más compleja.

Implementaciones RAID

Es obvio a partir de las secciones anteriores que RAID requiere "inteligencia" adicional sobre y por encima del procesamiento usual de discos de E/S para unidades individuales. Como mínimo, 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 (eliminando y añadiendo discos, iniciando y deteniendo reconstrucciones, etc.)

Hay dos métodos principales que se pueden utilizar para lograr estas tareas. Las próximas dos secciones las describen en más detalles.

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. Con el controlador adecuado,

las formaciones manejadas por una tarjeta de hardware RAID aparecen ante el sistema operativo anfitrión como si se tratasen de unidades de disco normales.

La mayoría de las tarjetas controladoras RAID trabajan con SCSI, aunque hay algunos controladores RAID basados en ATA también. En cualquier caso, la interfaz administrativa se implementa usualmente en una de tres formas:

- 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. Sin embargo, si está arrancando un sistema operativo desde una controladora RAID, se necesita una interfaz que no requiera un sistema operativo en ejecución.

Puesto que existen muchas tarjetas controladoras RAID en el mercado, es imposible ir en más detalles aquí. Lo mejor a hacer en estos casos es leer la documentación del fabricante para más información.



Figura 7 Controladora RAID. Allí podemos ver en la parte superior, los 4 conectores SATA disponibles para conectar hasta cuatro discos rígidos en simultáneo.

Software RAID

Software RAID es RAID implementado como kernel - o software a nivel de controladores para un sistema operativo particular. Como tal, proporciona más flexibilidad en términos de soporte de hardware - siempre y cuando el sistema operativo soporte ese hardware, se pueden configurar e implementar las formaciones RAID. Esto puede reducir dramáticamente el costo de implementar RAID al eliminar la necesidad de adquirir hardware costoso especializado.

A menudo el exceso de poder de CPU disponible para los cálculos de paridad RAID exceden en gran medida el poder de procesamiento presente en una tarjeta controladora RAID. Por lo

tanto, algunas implementaciones de software RAID en realidad tienen mejores capacidades de rendimiento que las implementaciones de hardware RAID.

Sin embargo, 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. En la mayoría de los casos, solamente se puede utilizar formaciones RAID 1 para arrancar, ya que el BIOS del computador no está al tanto de RAID. Puesto que no se puede distinguir una unidad única de una formación RAID 1 de un dispositivo de arranque no RAID, el BIOS puede iniciar exitosamente el proceso de arranque; luego el sistema operativo puede cambiarse a la operación desde el software RAID una vez que haya obtenido el control sobre el sistema.

Administración de volúmenes lógicos

Otra tecnología de almacenamiento avanzada es 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. Las capacidades exactas varían de acuerdo a la implementación específica, pero pueden incluir la agrupación del almacenamiento físico, redimensionamiento de volúmenes lógicos y la migración de datos.

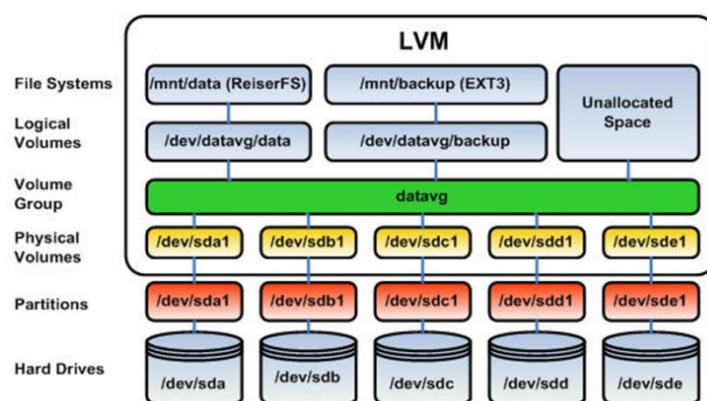


Figura 8 Esquema representativo de la distribución y el agrupamiento con LVM.

Agrupamiento de almacenamiento físico

Aunque los nombres dados a esta capacidad pueden variar, la agrupación del almacenamiento físico es la base para todas las implementaciones de LVM. Como su nombre lo implica, los dispositivos físicos de almacenamiento masivo se pueden agrupar de forma tal para crear uno o más dispositivos lógicos de almacenamiento. Los dispositivos lógicos de almacenamiento masivo (o volúmenes lógicos) pueden ser más grandes en capacidad que cualquiera de los dispositivos físicos de almacenamiento subyacentes.

Por ejemplo, dadas dos unidades de 100GB, se puede crear un volumen lógico de 200GB. Sin embargo, también se pueden crear un volumen lógico de 150GB y otro de 50GB. Es posible

cualquier combinación de volúmenes lógicos igual o menor que la capacidad total (en nuestro ejemplo 200GB). Las posibilidades están limitadas solamente a las necesidades de su organización.

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

En una configuración de sistemas no LVM, el quedarse sin espacio significa - en el mejor de los casos - mover archivos desde un dispositivo lleno a uno con espacio disponible. A menudo esto significa la reconfiguración de sus dispositivos de almacenamiento masivo. Sin embargo, LVM hace posible incrementar fácilmente el tamaño de un volumen lógico. Asuma por un momento que nuestro parque de almacenamiento de 200GB fue utilizado para crear un volumen lógico de 150GB, dejando el resto de 50GB en reserva. Si el volumen lógico de 150GB se llena, LVM permite incrementar su tamaño (digamos por 10GB) 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.