

SISTEMAS OPERATIVOS

UNIDAD 7: ADMINISTRACIÓN DE LOS PERIFÉRICOS

**Licenciatura en Ciencias de la Computación
Licenciatura en Sistemas de Información
Tecnatura Universitaria en Programación Web**

Departamento de Informática

Facultad de Ciencias Exactas, Físicas y Naturales

Universidad Nacional de San Juan

Esquema de contenidos

1. INTRODUCCIÓN	2
2. FUNCIONES	3
3. HARDWARE DE E/S	3
3.1. Tipos de periféricos	3
3.2. Canales de E/S	4
3.3. Procesador o Canal de E/S	4
4. ASIGNACIÓN DE PERIFÉRICOS	5
4.1. Políticas de asignación para periféricos dedicados	5
4.2. Políticas de asignación para periféricos compartidos	6
4.3. Políticas de asignación para periféricos virtuales	6
5. RUTINAS DEL ADMINISTRADOR DE PERIFÉRICOS	7
5.1. Controlador de tráfico de E/S	7
5.2. Planificador de E/S	7
5.3. Manipulador de periféricos (Driver)	7
6. EJECUCIÓN DE UNA OPERACIÓN DE E/S	8
7. PLANIFICACIÓN DE DISCO	8
7.1. FIFO, FCFS o cola	9
7.2. Más corto primero	9
7.3. Barrido Ascensor (SCAN)	10
7.4. Barrido Circular (C-SCAN, circular SCAN)	10
8. RAID	11
8.1. RAID 0	11
8.2. RAID 1	12
8.3. RAID 5	12
9. BIBLIOGRAFÍA	13

1. INTRODUCCIÓN

La administración de periféricos de E/S involucra a los dispositivos que vienen con el sistema (de fábrica) como los que se van agregando para hacer más potente o actualizar el sistema.

Probablemente, la gestión de los dispositivos de E/S sea el aspecto más complejo en el diseño de un SO, dado que existe una gran variedad de dispositivos y aplicaciones de los mismos difiriendo en aspectos tales como:

Velocidad de transferencia de datos (por ejemplo, un disco magnético está en torno a los 10 millones de c.p.s., mientras que en un teclado ronda los 50 c.p.s.)

Unidad de transferencia (puede ser carácter, palabra, byte, registro, bloque, ...)

Representación de los datos (distintas formas de codificación para diferentes soportes de E/S)

Operaciones permitidas (distintos periféricos, distintos tipos de operaciones, por ejemplo: una impresora puede imprimir un mouse no)

Condiciones de error (distintos periféricos tienen distintas causas de error, por ejemplo: falta de papel puede emitirlo un plotter, una memoria de estado sólido no)

Además, el SO debe ser capaz de administrar no sólo los dispositivos actuales, sino también los que aparezcan en el futuro.

2. FUNCIONES

Las funciones principales del Administrador de Periféricos son:

1. Determinar **políticas** para la asignación y desasignación de dispositivos. Esto es, qué proceso obtiene el dispositivo, cuándo y por cuánto tiempo
2. **Asignar y desasignar** los periféricos a los procesos
3. Mantener la **información de control** sobre el **estado, interrupciones y errores** de cada dispositivo del sistema
4. Ejecutar las **operaciones de E/S** (programación de E/S)

3. HARDWARE DE E/S

3.1. Tipos de periféricos

Existen distintas clasificaciones para periféricos, pero lo que interesa es determinar si ese periférico debe ser de uso **exclusivo** para un proceso dadas sus características.

Por ejemplo, una impresora no se puede asignar a más de un proceso. Ese periférico va a estar dedicado solamente a uno, por lo que se denomina *periférico dedicado*.

Aquellos que pueden ser utilizados por varios procesos a lo largo de un lapso de tiempo son los *periféricos compartidos*. El caso típico es el disco, que permite acceso aleatorio. Si bien el periférico va a estar dedicado a un proceso durante la transferencia, no es necesario que lo esté a lo largo de todo el proceso.

La otra clase que se considerará es la de los *periféricos virtuales*: son periféricos que se los simula de alguna manera. Y dado que, los periféricos dedicados obligan a ejecutar un sólo proceso por vez que use esos periféricos, entonces si se pudieran simular, se podría ejecutar más de un proceso de ese tipo simultáneamente.

Por lo tanto, existen distintos tipos de periféricos que se clasifican en función de su asignación en:

- **Dedicados**: un dispositivo está asignado a un solo proceso
- **Compartidos**: un dispositivo compartido por varios procesos
- **Virtuales**: periféricos dedicados son simulados sobre dispositivos de acceso compartido (disco)

Por lo tanto, la asignación adopta una de dos formas: de acceso dedicado y de acceso compartido.

3.2. Canales de E/S

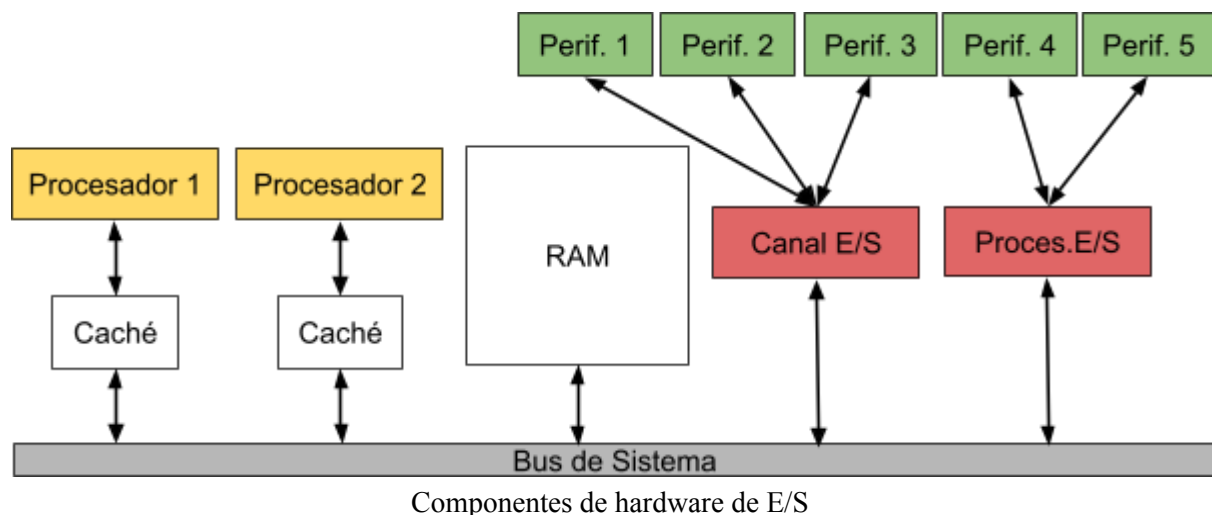
A medida que las computadoras han evolucionado, la complejidad y sofisticación de sus componentes se ha incrementado. En ningún lugar se hace más evidente que en el funcionamiento de la E/S.

Inicialmente, la CPU controlaba directamente al periférico lo que significaba un desperdicio del procesador, esto es debido a la enorme disparidad de velocidades entre el procesador y los periféricos de E/S. Entonces, el procesador debía sincronizarse con la velocidad del periférico (esta sincronización se hace a la velocidad del dispositivo más lento) e interactuar directamente con él.

La evolución tecnológica ha producido que cada vez más y más funciones de E/S se realicen sin la intervención de la CPU. La CPU es relevada de las tareas relacionadas con las tareas de E/S, mejorando las prestaciones.

Entonces, el procesador principal del sistema es liberado de interactuar con los dispositivos periféricos dejando esa tarea a un procesador de menor potencia (y, por lo tanto, de menor coste). Este procesador o procesadores se denominan **chipset** (conjunto de chips) y auxilian al procesador principal de tener que lidiar con los dispositivos de E/S.

Este procesador secundario es denominado **Módulo de E/S**, **Procesador de E/S** o **Canal de E/S** y, aunque estos términos no son iguales entre sí, serán utilizados indistintamente.



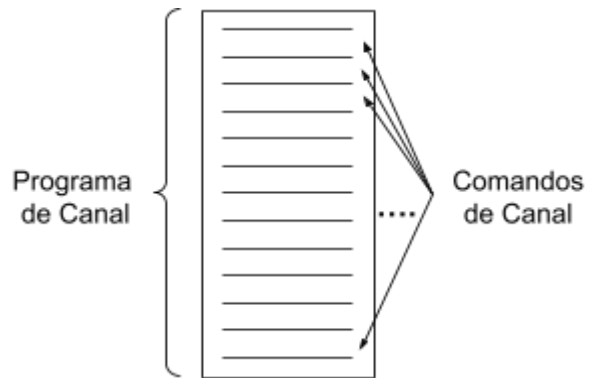
3.3. Procesador o Canal de E/S

Un canal es un *procesador auxiliar especializado* en operaciones de E/S, que como cualquier procesador, tiene la capacidad ejecutar programas, en este caso, son programas específicos denominados *programa de canal* que permiten manipular dispositivos de E/S.

Como todo programa está constituido por instrucciones, pero, en el caso de programas de E/S están formados por **instrucciones de E/S**, que se llaman **comandos de canal**.

Los comandos de canal otorgan un control total sobre el dispositivo de E/S ya que permiten dar órdenes al periférico y controlar la transferencia de información.

Resumiendo, un programa de canal está conformado por instrucciones de canal y puede estar almacenado en la memoria principal (o en la memoria del Canal de E/S, si la tuviere) y será ejecutado por el Canal de E/S.



4. ASIGNACIÓN DE PERIFÉRICOS

4.1. Políticas de asignación para periféricos dedicados

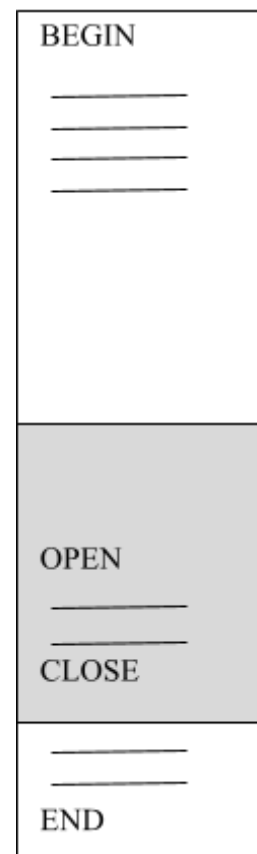
Se recuerda que, un periférico de tipo dedicado debe ser asignado de manera exclusiva a un proceso. Ejemplos de periféricos dedicados: impresora, cinta magnética, unidad de DVD, scanner, plotter.

Para los periféricos *dedicados* se puede realizar la asignación al proceso en tres momentos diferentes:

- Al comienzo del proceso
- Al comienzo de la etapa
- Al realizarse la instrucción Open

Asignando *al comienzo del proceso* se asegura que cada vez que un proceso necesite el periférico, lo va a encontrar disponible. Pero si el proceso tiene una larga secuencia de instrucciones y, éste, hace uso del periférico al final del programa, entonces se desperdicia el periférico durante la mayor parte de la ejecución del proceso. Con esta política un proceso pasa del estado de *Nuevo* a *Listo* cuando tiene todos los periféricos asignados.

Asignando *al comienzo de la etapa* (área sombreada) se intenta entonces evitar que los periféricos estén ociosos y asignarlos solo en el momento previo de las necesidades del proceso (por ejemplo, al invocar a la subrutina que utiliza al periférico). Puede ocurrir que en el momento de necesitar ejecutar esa etapa el periférico no se encuentre disponible y se deba tomar una decisión entre esperar a que el periférico se libere o cancelar el proceso. A pesar de todo, todavía pasa un tiempo hasta que se haga uso del dispositivo. Con esta política un proceso pasa del estado de *Nuevo* a *Listo* sin los periféricos asignados.



Asignarlo *en el momento de la instrucción Open* significa que solicita la asignación en el momento que realmente quiere usar el dispositivo. Puede ocurrir, en este caso también, que no se encuentre disponible y habrá que tomar la decisión de esperar o cancelar.

Asignación parcial y total de periféricos dedicados

La asignación de un dispositivo periférico puede hacerse de manera total o parcial.

En la *asignación total* el SO no entregará la totalidad de los periféricos dedicados que el proceso necesite hasta no tenerlos a todos disponibles. Si un proceso necesita 3 impresoras, hasta que no haya 3 impresoras libres no se le asignan, ni a nivel de etapa ni a nivel de proceso.

En la *asignación parcial* a medida que el SO encuentra periféricos libres, los va asignando a los procesos, aunque no tenga la totalidad de los dispositivos que ese proceso requiere, esta política de asignación es la que sigue cuando se asigna en el momento de la instrucción Open.

El caso de asignación parcial se debe controlar muy bien con el fin de evitar interbloqueos.

4.2. Políticas de asignación para periféricos compartidos

Los periféricos compartidos pueden ser utilizados por varios procesos, aunque, claramente, no al mismo tiempo.

Y, si bien, la mayoría de los periféricos de una sistema son de tipo compartido y hay muchos procesos intentando satisfacer sus necesidades de uso sobre ellos, el SO lleva adelante la gestión de los dispositivos compartidos sin mayores inconvenientes manteniendo un conjunto de tablas que indican la disponibilidad (estado) de periféricos y canales de E/S para gestionar las colas de espera de cada dispositivo.

Debido a que usualmente hay varios procesos intentando utilizar el periférico, se necesita definir alguna política para establecer qué pedido se satisface primero basándose, usualmente, en:

- **Lista de prioridades**
- **Conseguir una mejor utilización de los recursos del sistema**

En el caso de los periféricos compartidos, la asignación se hace en el momento de usarlos, es decir, en el instante de realizar la operación de E/S.

4.3. Políticas de asignación para periféricos virtuales

Algunos dispositivos que deben ser dedicados por su naturaleza (impresoras, scanners) pueden convertirse en compartidos a través de una técnica llamada *Spool* (Simultaneous Peripheral Operations On-Line).

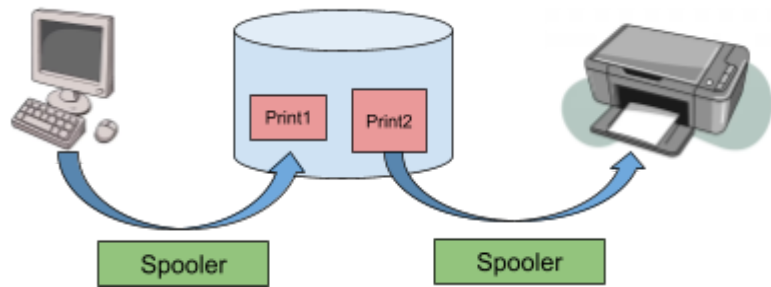
Esta técnica admite a un proceso la transferencia de datos hacia un buffer (área temporal de memoria secundaria), donde otro proceso (spooler) puede acceder para su procesamiento en otro momento.

El periférico sobre el cual se puede hacer la simulación debe tener 2 características:

- que el dispositivo sea de tipo compartido

- que permita llevar a cabo acceso directo (disco magnético) o, bien, aleatorio (disco de estado sólido).

Por ejemplo, cuando el spooler (rutina que aplica la técnica de Spooling) intercepta un pedido de impresión de un documento, procede a convertirlo en un archivo que se graba en disco. Estos documentos se



almacenan en disco hasta que el periférico dedicado esté disponible y entonces se inicia la impresión del documento. Note que el proceso envía el trabajo a imprimir independientemente de si la impresora le fue asignada o no, permitiendo continuar con su ejecución. Incluso si el proceso tuviera asignada la impresora, también sería mucho más rápido ya que envía a “imprimir a disco” continuando con su ejecución.

5. RUTINAS DEL ADMINISTRADOR DE PERIFÉRICOS

5.1. Controlador de tráfico de E/S

Esta rutina le permite almacenar la información del estado de los dispositivos y canales de E/S en tablas. Con esta información el Controlador de tráfico de E/S determina si es posible realizar una operación solicitada.

5.2. Planificador de E/S

La rutina de Planificador de E/S decide qué proceso toma el recurso aplicando una política (algoritmo) de selección. Determina el orden en que se asignan los dispositivos a los procesos.

Una vez que hay un orden, el Controlador de tráfico de E/S es el que determina si puede ser satisfecho el pedido.

Las políticas a considerar son las mismas que se usan para procesos.

5.3. Manipulador de periféricos (Driver)

En este caso, a diferencia de las rutinas anteriores, no existe una única rutina sino que cada periférico tiene su propia rutina de manipulación (device driver o driver o, en castellano, controlador) para que el SO pueda hacer uso del dispositivo. El manipulador de periféricos se encarga de:

- Armar el programa de E/S que ejecuta el Canal de E/S
- Emitir la instrucción de arranque de E/S
- Procesar las interrupciones del dispositivo
- Manejar errores del dispositivo
- Realizar una planificación de acuerdo al periférico (estrategia de acceso)

6. EJECUCIÓN DE UNA OPERACIÓN DE E/S

La secuencia de acciones para que se pueda realizar una operación de E/S es como se indica:

1. Cuando un proceso está en estado Activo significa que se están ejecutando sus instrucciones. Si una instrucción invoca una operación de E/S, entonces se produce una interrupción ya que es una instrucción que debe ejecutarse en modo privilegiado provocando un cambio de modo usuario a modo supervisor.
2. La primera rutina que interviene es el Manipulador de periféricos, el cual, con los datos que recibe del proceso que quiere ejecutar la operación de E/S, construye un *programa de canal* ya que conoce cuál es el periférico concreto, de qué tipo de periférico se trata y sabe si la operación es de lectura o de escritura.
3. A continuación, invoca al Controlador de Tráfico de E/S para que actualice las tablas de información con la lista de procesos en espera del periférico.
4. Una vez que el periférico se libera, el Controlador de Tráfico invoca al Planificador de Procesos de E/S para que aplique la política de selección y decida a cuál de los procesos en espera se le otorgará el uso del periférico.
5. Cuando al proceso le fue asignado el periférico, el Manipulador de periféricos (device driver) emite una instrucción de arranque provocando que el Canal de E/S acceda a un lugar de memoria (memoria principal del sistema o memoria propia del canal, si la tuviera) que contiene la dirección de la primera instrucción del *programa de canal* de E/S (construido en el punto 2).
6. Cuando se termina de ejecutar el programa de canal, el canal emite una interrupción de E/S para que el SO analice el resultado de la operación de E/S. Dicho resultado podría ser una operación exitosa o bien algún error en la operación del periférico.
7. Luego, el SO finaliza la interrupción y restablece la operación normal (conmuta de modo privilegiado a modo usuario) retornando el control y el resultado de la operación de E/S al proceso que la solicitó.

7. PLANIFICACIÓN DE DISCO

El disco magnético, disco duro, disco rígido o HDD (hard disk drive) son distintas maneras habituales de llamar, indistintamente, al almacenamiento masivo por excelencia, desde su invención allá por el año 1956 por parte de IBM. Desde entonces

han pasado casi 70 años y, si bien, su reinado estuvo amenazado en varias oportunidades ha logrado salir siempre victorioso.

En estos momentos, su retador es el disco de estado sólido y habrá que ver qué es lo que sucede en los próximos años.

Los aspectos en los cuales se impone el disco rígido son: menor costo, mayor vida útil, mayor capacidad de almacenamiento.

Mientras que el disco de estado sólido se impone en: velocidad, menor consumo de energía, silencioso.

Actualmente, una configuración típica sería una combinación de ambos dispositivos en el cual el SSD (aprovechando su mayor velocidad) se utiliza como disco de arranque (ahí está almacenado el SO y las aplicaciones) y el HDD (aprovechando su mayor capacidad) es utilizado para almacenamiento de datos.

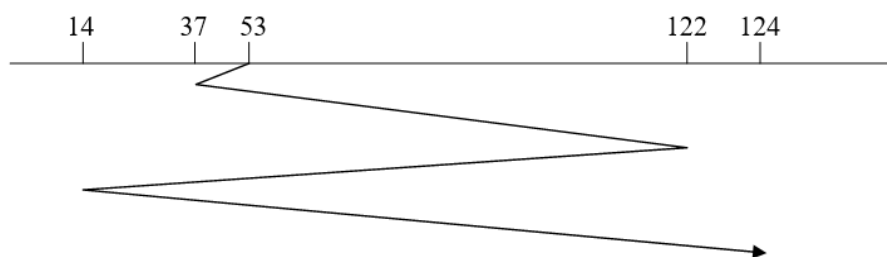
Mientras que en los grandes centros de datos sigue siendo el preferido el disco magnético y, mientras se siga imponiendo en los aspectos antes mencionados (costo, durabilidad y capacidad), no parece que al SSD le alcance con una mayor velocidad solamente para destronar a su competidor.

A partir de esta situación, es que se continúan mejorando los algoritmos que permiten la mejor utilización del disco rígido, ya que al tener desplazamientos producidos por piezas mecánicas, resulta fundamental para la performance del dispositivo. Se verán algunos de los algoritmos más utilizados.

7.1. FIFO, FCFS o cola

El primero que llega es el primero en ser servido (FCFS – First Come First Served). Es el más simple. Sin embargo, no da el mejor servicio promedio. Considere una cola que solicita accesos a disco a las siguientes pistas: 37, 122, 14 y 124 (el cabezal está inicialmente en la pista 53).

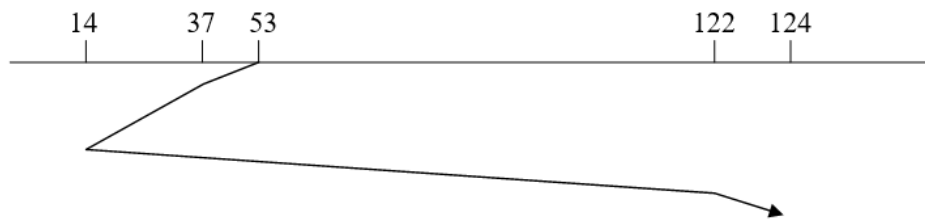
Con este algoritmo y los datos de prueba, el recorrido del cabezal de lectoescritura es de 319 pistas.



7.2. Más corto primero

Se selecciona el pedido con el menor recorrido de búsqueda desde la posición actual del cabezal, o sea, a la pista que está más cerca. Considerando los mismos datos del ejemplo anterior, el recorrido del cabezal es de 149 pistas.

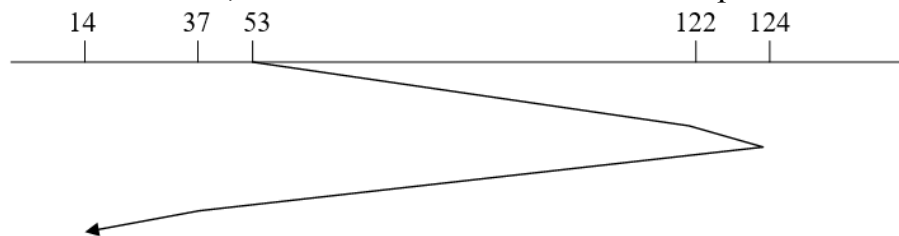
Este algoritmo puede ocasionar inanición (un pedido no es atendido nunca). Por lo tanto, no es óptimo. Aunque es mejor que el FIFO. Una solución a la inanición es encerrar a los pedidos en grupos.



7.3. Barrido Ascensor (SCAN)

Con esta técnica, primero, se ordenan las solicitudes de acceso a disco de manera ascendente o descendente. Luego, la cabeza lectora/grabadora comienza en un extremo del disco y se va moviendo hacia el otro extremo, atendiendo los pedidos a medida que llega a cada pista. Cuando llega al otro extremo, vuelve atendiendo los pedidos nuevos en orden inverso.

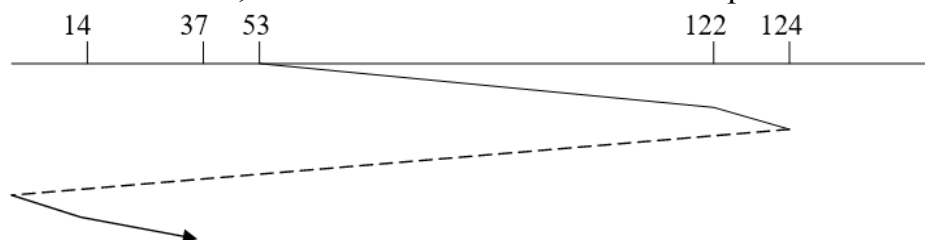
Para el mismo set de datos, el recorrido del cabezal es de 181 pistas.



7.4. Barrido Circular (C-SCAN, circular SCAN)

Este algoritmo está diseñado para atender los pedidos con un tiempo de espera uniforme. Mueve la cabeza lectora-grabadora de un extremo al otro del disco igual que en la técnica anterior. Cuando llega al extremo del disco, vuelve inmediatamente al comienzo del disco sin atender ningún pedido. Trata al disco como si fuese circular.

Para el mismo set de datos, el recorrido del cabezal es de 204 pistas.



En la práctica ni SCAN ni C-SCAN mueven el cabezal de un extremo al otro, en realidad, se desplaza hasta la posición del último pedido en esa dirección. Cuando no existen más pedidos en esa dirección se cambia el sentido del movimiento de la cabeza de lectoescritura.

8. RAID

RAID, del inglés redundant array of independent disks, (matriz o grupo redundante de discos independientes) se refiere a un sistema de almacenamiento de datos que utiliza múltiples (mínimo 2) unidades de discos magnéticos o SSD, entre las cuales se distribuyen o replican los datos.

La finalidad más importante de un sistema RAID de discos es la de proteger los datos en caso de que un disco duro falle, pero en algunos casos también tiene como función mejorar la velocidad de lectura de varios discos que conforman un único volumen. De esta manera, el SO ve un solo disco en lugar de ver varios discos diferentes.

En otras palabras, consiste en crear un único volumen (lógico) con varios discos físicos funcionando en conjunto. Las configuraciones más utilizadas son:

- **Disk mirroring (espejado):** configuración RAID que busca **redundancia** de datos ante un posible fallo en una de las unidades de almacenamiento
- **Disk striping (rebanado o rallado):** configuración RAID que busca aumentar la **velocidad** en la transferencia de datos
- **Distributed parity (paridad distribuida):** es una combinación de los anteriores permitiendo aumento de velocidad y redundancia de datos

Un sistema RAID puede ser implementado por software o por hardware. Actualmente, la implementación por software es la más utilizada en el ámbito hogareño y de pequeñas organizaciones, dejando la implementación por hardware (por su costo) para grandes organizaciones.

Si bien existen varias configuraciones, las más utilizadas son: RAID 0, RAID 1 y RAID 5.

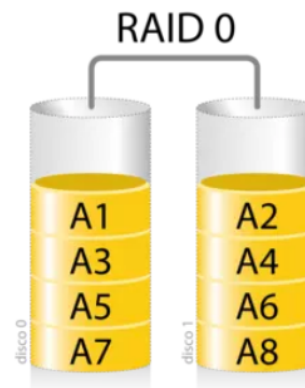
8.1. RAID 0

La configuración RAID 0 es de tipo configuración disk striping, por lo que no ofrece redundancia de datos. Para este tipo de configuración se requieren al menos dos unidades de almacenamiento y el número máximo viene determinado por la controladora. El SO verá esta configuración cómo si fuera un único disco duro, sin importar la cantidad de unidades instaladas.

En esta configuración los datos se escriben secuencialmente.

Si hay dos discos, uno denominado HDD1 y el otro HDD2, lo que hace es partir la información y escribirla primero en el HDD1 y luego en el HDD2, y así hasta almacenar toda la información.

Gracias a este sistema (acceso paralelo) se permite prácticamente duplicar la velocidad de escritura y también, de lectura. Pero, esta configuración tiene un grave problema. Si cualquiera de las unidades usadas en la RAID 0 sufre un daño, **toda la información** se perderá completamente.



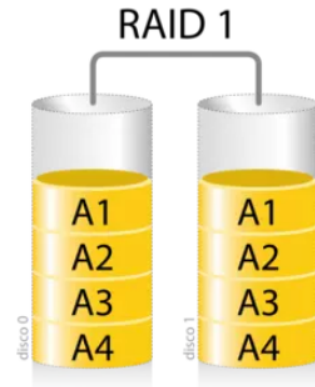
8.2. RAID 1

RAID 1 es la primera configuración de tipo mirroring o de espejo. Este tipo de configuración busca realizar una copia de seguridad de los datos almacenados. La información se escribe en los discos duros de manera simultánea. Se requiere de al menos dos unidades y el límite está determinado por la controladora.

En esta configuración la información se escribe en el disco duro primario y secundario al mismo tiempo.

La gran ventaja de este modelo es que, si se daña una de las unidades, se puede recuperar la información almacenada. No es habitual, pero ambas unidades se podrían dañar al mismo tiempo, en ese caso, la información se pierde.

Algo para tener en cuenta es que las dos unidades de almacenamiento deben tener la misma capacidad. Si se empareja un disco de 500 GB con otro de 1 TB, solo se utilizarán 500 GB de la unidad de 1 TB, independientemente de si es primaria o secundaria.

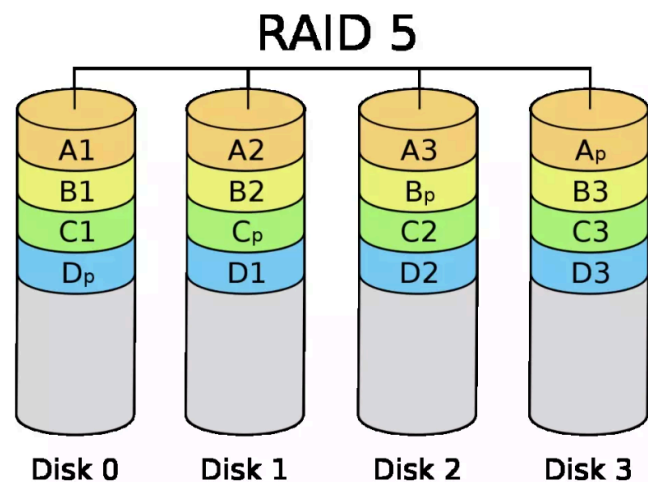


8.3. RAID 5

Esta configuración de discos duros con paridad distribuida es bastante utilizada en la actualidad. Las unidades de almacenamiento se dividen en bloques donde se almacena la información. Dentro del conjunto de bloques, uno se destina a paridad, para garantizar la redundancia de datos (los códigos de redundancia permiten detectar y, en algunos casos, corregir errores en los datos).

La paridad permite reconstruir los datos guardados en un disco duro, si éste termina fallando. El bloque de paridad, por otro lado, se almacena en un disco duro distinto para poder restaurar la información.

El acceso en paralelo a los datos permite que las operaciones de lectura y escritura se realicen a mucha más velocidad.



Se requiere para este tipo de configuración de, al menos, tres discos duros para así poder garantizar la redundancia de los datos.

El inconveniente de este sistema es que sólo permite el fallo de un disco a la vez. Si por cualquier motivo se dañan dos de las unidades de almacenamiento que conforman el RAID 5, la información se perderá.

Por último, la capacidad total que ofrece un sistema RAID 5 para datos de usuario se calcula como: $(\text{nro. de discos} - 1) * \text{capacidad de un disco}$

9. BIBLIOGRAFÍA

Fundamentos de Sistemas Operativos, A. Silberschatz

Sistemas Operativos. Aspectos internos y principios de funcionamiento, W. Stallings

Sistemas operativos modernos. A. Tanenbaum

Sistemas operativos. Un enfoque basado en conceptos. D. Dhamdhere