

Sistemas operacionais

Aula 7 - Armazenamento de massa

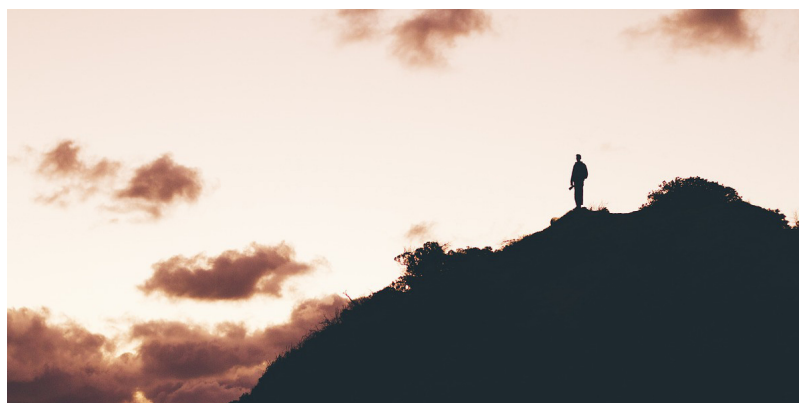
INTRODUÇÃO



Quando desejamos armazenar dados, no computador, para uso posterior utilizamos algum tipo de dispositivo com grande capacidade de armazenamento como o HD.

Este é um exemplo de dispositivo de armazenamento de massa que será objeto de estudo desta aula.

OBJETIVOS



Analisar o funcionamento do HD.

Identificar os tipos de RAID.

ESTRUTURA DE ARMAZENAMENTO DE MASSA

Você pode estar se perguntando: “Afinal, o que é o armazenamento de massa?”

Armazenamento de massa

Um dispositivo de armazenamento é capaz de registrar informações (dados) para posterior consulta ou uso.

O termo “em massa” significa que o dispositivo ou aparelho tem capacidade para arquivar grandes volumes de dados.

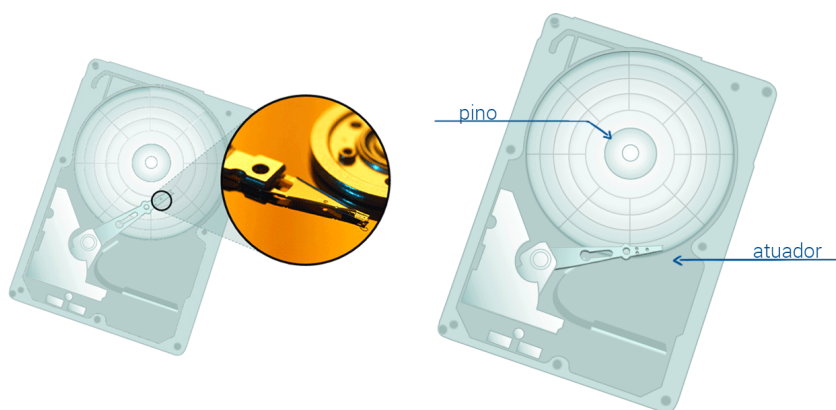
Estes dispositivos podem ser de vários tipos e incluem tanto o armazenamento local, utilizando discos rígidos magnéticos, como [discos SSD \(glossário\)](#).

DISCOS MAGNÉTICOS

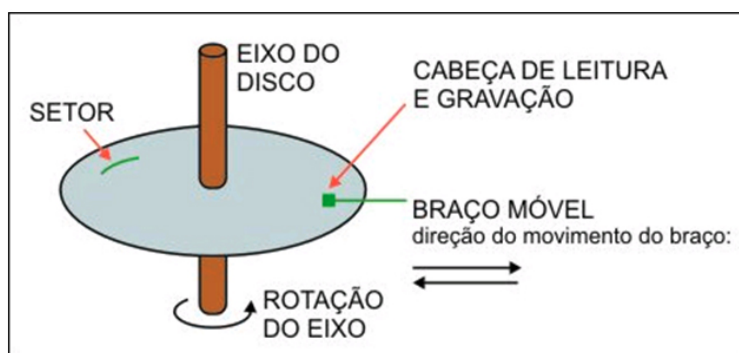
Discos magnéticos são dispositivos para [armazenamento não volátil \(glossário\)](#) de dados.

Os atuais discos magnéticos ou discos rígidos incorporam eletrônica de controle, motor para girar o disco, cabeças de leitura / gravação e o mecanismo para o posicionamento das cabeças.

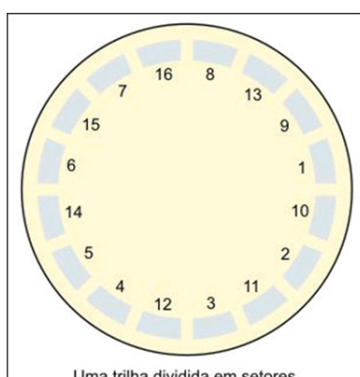
Observe, a seguir, como acontece a leitura do disco:



Os discos possuem um formato circular. Suas duas superfícies são cobertas com um material magnético. Observe a ilustração de um disco a seguir:



Durante a [formatação \(glossário\)](#), cada superfície é dividida em trilhas e cada trilha é dividida em setores (também chamados de bloco do disco), onde são armazenadas as informações conforme a ilustração a seguir:



As informações são lidas ou escritas através de uma cabeça de leitura/gravação.

TEMPO DE ACESSO AO DISCO

O tempo de acesso aos dados de um disco é definido como o período decorrido entre a ordem de acesso e o final da transferência dos dados. Ele não é constante, e varia em função da localização dos dados no disco.

O tempo de acesso pode ser definido com a seguinte fórmula:

$$\text{Tempo de Acesso} = \text{Tempo de (Seek + Latência + Transferência)}$$

Tempo Seek ou Tempo de Busca

É o tempo decorrido entre a cabeça de gravação se deslocar de sua posição atual até a trilha onde está o setor do dado a ser lido.

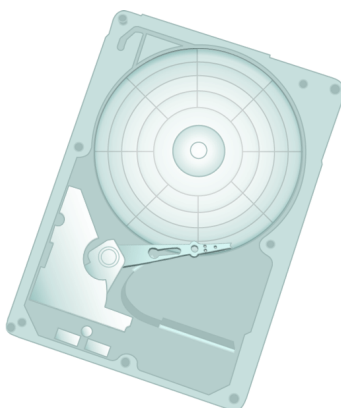
Tempo de Latência

Também chamada latência rotacional, é o tempo gasto entre a chegada da cabeça de leitura/gravação sobre a trilha e a passagem do setor desejado na posição da cabeça.

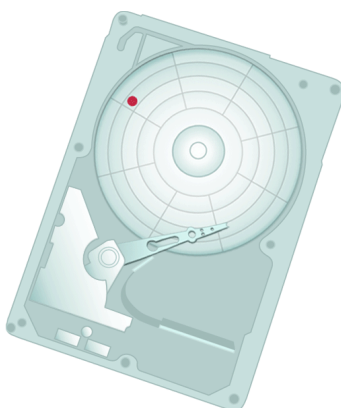
Tempo de Transferência:

É o tempo consumido na transmissão dos bits entre computador e disco e vice-versa. Esse tempo depende da interface e do disco, que definem o throughput (taxa de transferência) do disco.

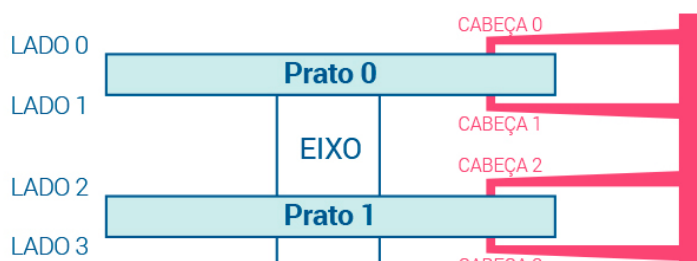
Veja, a seguir, uma animação que representa o tempo de busca:



Agora, veja como funciona a latência rotacional:



Os discos rígidos atuais são construídos com muitas superfícies de gravação, montadas em torno de um eixo comum. Os braços atuadores responsáveis pelo posicionamento das cabeças de leitura/gravação são montados em uma única estrutura, de forma que eles se movam solidariamente.



Observe que quando uma determinada cabeça é posicionada sobre uma trilha, as demais cabeças estarão também posicionadas sobre as trilhas das outras superfícies localizadas à mesma distância do eixo central e, portanto, todas as trilhas (localizadas a uma mesma distância do eixo central do disco) poderão ser acessadas simultaneamente por todas as cabeças.

Surge, neste caso, o conceito de [cilindro \(glossário\)](#).

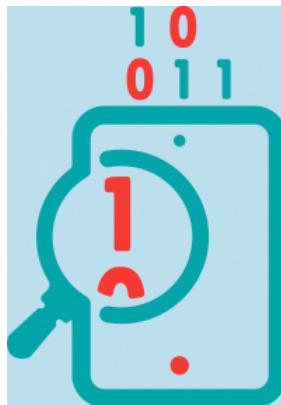


GERENCIAMENTO DE ESPAÇO LIVRE EM DISCO

O tempo de acesso aos dados de um disco é definido como o período decorrido entre a ordem de acesso e o final da transferência dos dados. Ele não é constante, e varia em função da localização dos dados no disco.

O tempo de acesso pode ser definido com a seguinte fórmula:

MAPA DE BITS

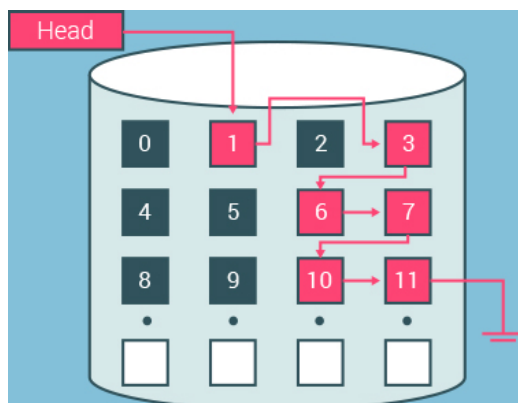


Fonte da Imagem:

Cada bloco é representado por 1 bit; se o bloco está livre, o bit é "0", senão é "1".

Por ex.: considere um disco onde os blocos 2,3,4,5,8,9,10 estão livres, e o resto dos blocos estão ocupados.
11000011000....

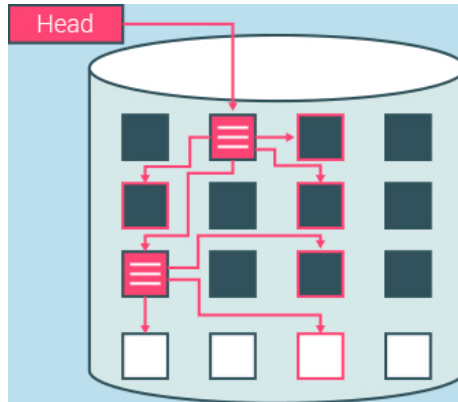
LISTA DE BLOCOS LIVRES



Fonte da Imagem:

Mantém todos os blocos livres ligados por uma lista, e guardar a cabeça (endereço inicial da lista) da lista. Aproveita os próprios blocos livres para armazenar os endereços da lista. Não é possível ter uma visão geral sobre a situação de um conjunto de setores; deve-se ler uma a um, para saber se o setor está disponível ou não.

BLOCO DE ENDEREÇOS

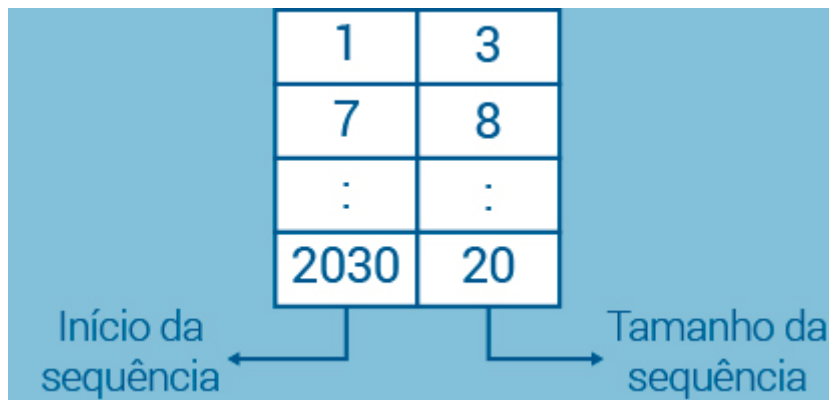


Fonte da Imagem:

Utiliza-se o primeiro bloco livre, como bloco de endereços, para armazenar o endereço dos próximos "n" blocos livres, sendo que o bloco número "n", é outro bloco de endereços.

Guarda-se o endereço do primeiro bloco de endereços. Este método também permite, em uma única leitura (do bloco de endereços), saber sobre a situação de vários blocos, incluindo os endereços.

LISTA DE BLOCOS LIVRES CONTÍGUOS



Fonte da Imagem:

Aproveita a ideia de que sempre existem vários blocos livres contíguos. Mantém uma tabela de blocos livres, onde cada entrada, na tabela, contém o endereço do 1º bloco de um grupo de blocos contíguos e o tamanho do grupo.

ALOCÇÃO DE ESPAÇO EM DISCO

Uma das preocupações do sistema de arquivos é como alocar os espaços livres do disco para os arquivos, de forma que o disco seja mais bem utilizado e os arquivos possam ser acessados mais rapidamente.

Os três principais métodos de alocação de espaço do disco são:

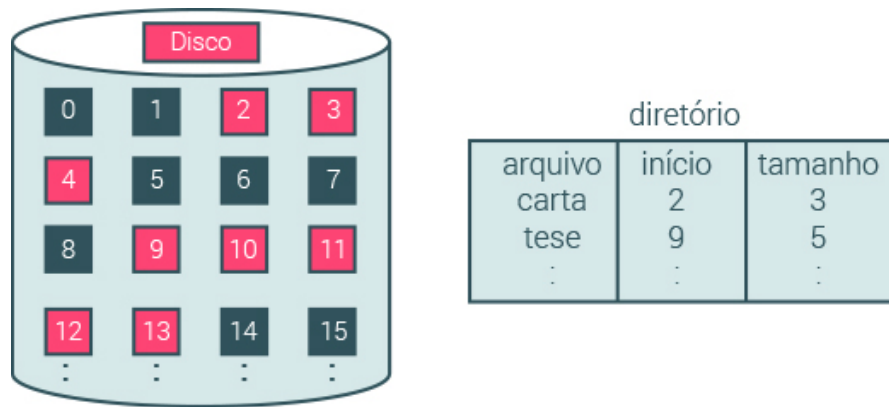
1º Alocação Contígua

Este método procura identificar uma sequência contínua de blocos que apresente um tamanho suficiente para armazenar todas as informações relativas ao arquivo, ou seja, caso um arquivo precise de N blocos para o seu armazenamento, o sistema operacional procurará identificar uma sequência contínua de N blocos livres do disco.

Nesta técnica, o espaço só será alocado se for contíguo, caso contrário, o arquivo não poderá ser armazenado ou estendido. Se, por acaso, existir mais de uma possibilidade, então poderá ser utilizada uma técnica para selecionar uma opção.

As técnicas que podem ser utilizadas são: Worst-Fit, Best-Fit e First-Fit.

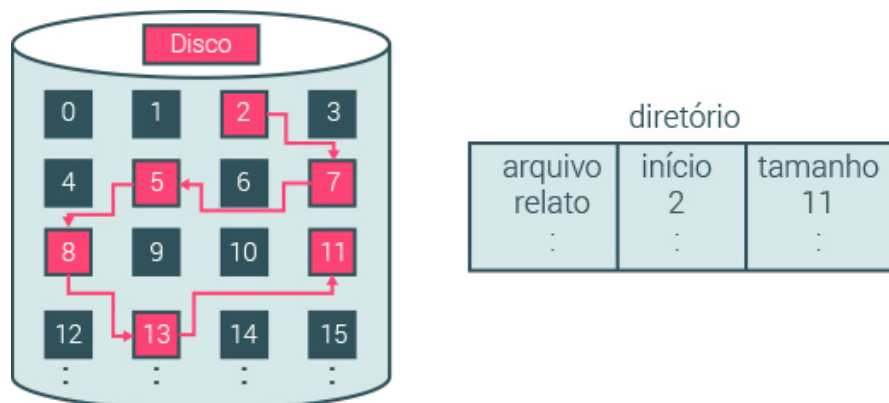
A alocação contígua de um arquivo é definida pelo endereço do primeiro bloco e seu comprimento, e, dessa forma, é colocada no diretório. Veja, abaixo:



2ª Alocação Ligada

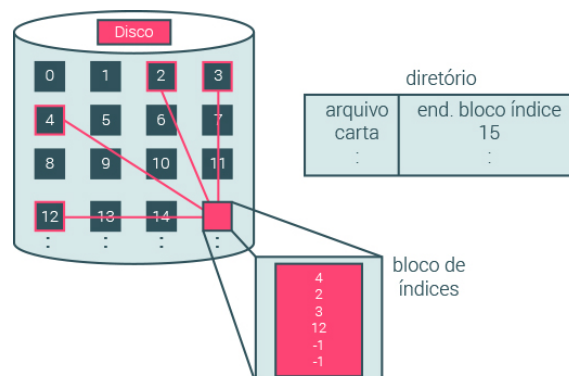
Nesta técnica, os blocos alocados para um arquivo são encadeados em uma lista, não importando a contiguidade. Assim, seria armazenado no diretório o nome do arquivo e duas variáveis, isto é, os identificadores do primeiro e do último bloco do arquivo.

Em cada um dos blocos utilizados pelo arquivo, seria reservado um espaço para conter o endereço do próximo bloco da lista.



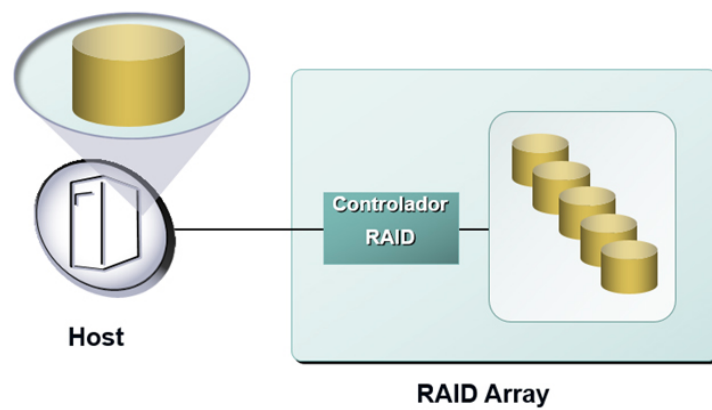
3ª Alocação Indexada

Neste método, cada arquivo possui seu próprio bloco de índice, que é um vetor de endereços. A *i*-ésima entrada no bloco de índice aponta para o *i*-ésimo bloco do arquivo, como mostra a imagem:



RAID

É a sigla para Redundant Array of Independent Disks ou, em tradução livre, algo como "Matriz Redundante de Discos Independentes". Trata-se, basicamente, de uma solução computacional combinando vários discos rígidos (HDs) para formar uma única [unidade lógica \(glossário\)](#) de



NÍVEIS DE RAID

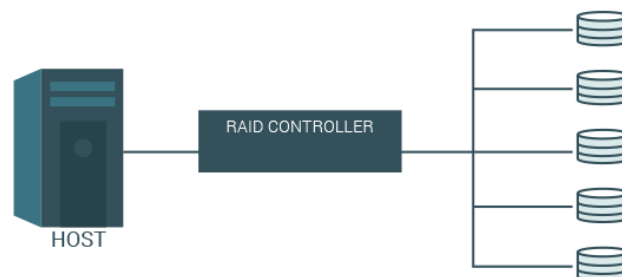
Para que um sistema RAID seja criado, é necessário utilizar pelo menos dois HDs (ou SSDs). Mas não é só isso: é necessário também definir o nível de RAID do sistema. Cada nível possui características distintas justamente para atender às mais variadas necessidades. A seguir, veremos os níveis mais comuns:

RAID 0 (ZERO)

Também conhecido como *striping* (fracionamento), é aquele onde os dados são divididos em pequenos segmentos e distribuídos entre os discos.

Trata-se de um nível que não oferece proteção contra falhas, já que nele não existe redundância. Isto significa que uma falha, em qualquer um dos discos, pode ocasionar perda de informações para o sistema todo, especialmente porque "pedaços" do mesmo arquivo podem ficar armazenados em discos diferentes.

O foco do RAID 0 acaba sendo o desempenho, uma vez que o sistema praticamente soma a velocidade de transmissão de dados de cada unidade. Assim, pelo menos teoricamente, quanto mais discos houver no sistema, [maior é a sua taxa de transferência \(glossário\)](#). Veja o exemplo a seguir:



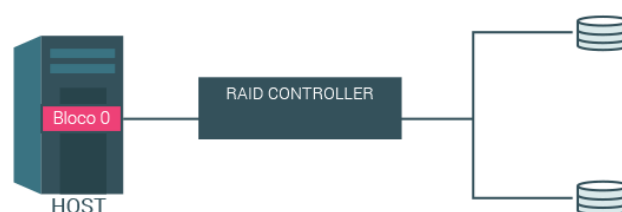
RAID 1

O RAID 1 é, provavelmente, o modelo mais conhecido. Nele, uma unidade "duplica" a outra, isto é, faz uma "cópia" da primeira, razão pela qual o nível também é conhecido como *mirroring* (espelhamento).

Com isso, se o disco principal falhar, os dados podem ser recuperados imediatamente porque existem cópias no outro.

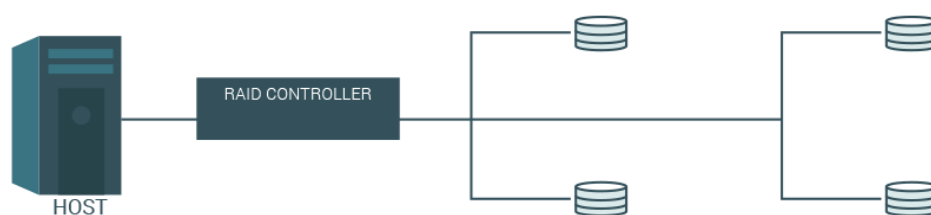
Perceba que, por conta desta característica, sistemas RAID 1 devem funcionar em pares, de maneira que uma unidade sempre tenha um "clone".

O nível RAID 1 é claramente objetivado na proteção dos dados, ou seja, não torna o acesso mais rápido. Na verdade, pode até ocorrer uma ligeira perda de desempenho, uma vez que o processo de gravação acaba tendo que acontecer duas vezes, uma em cada unidade. Observe, como o RAID 1 funciona:

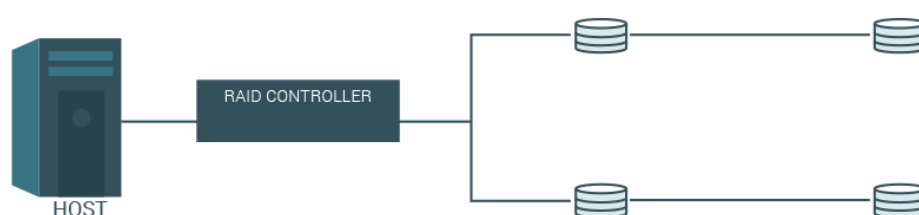


RAID 0+1 E RAID 10

Tal como você já deve ter imaginado, o nível RAID 0+1 é um sistema "híbrido" (hybrid RAID), ou seja, que combina RAID 0 com RAID 1. Para isso, o sistema precisa ter pelo menos quatro unidades de armazenamento, duas para cada nível. Assim, tem-se uma solução RAID que considera tanto o aspecto do desempenho quanto o da redundância. Observe, abaixo:

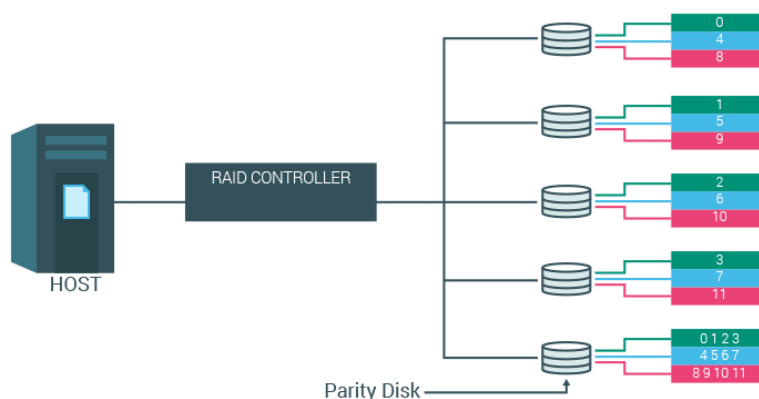


Há uma variação chamada RAID 10 (ou RAID 1+0) de funcionamento semelhante. A diferença essencial é que, no RAID 0+1, o sistema se transforma em RAID 0 em caso de falha; no RAID 1+0, o sistema assume o nível RAID 1.



RAID 5

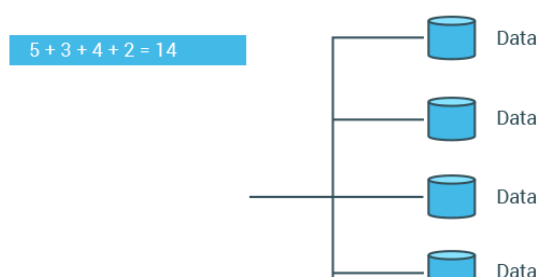
É outro nível bastante conhecido. Nele, o aspecto da redundância também é considerado, mas de maneira diferente: em vez de existir uma unidade de armazenamento inteira como réplica, os próprios discos servem de proteção. Deste modo, pode-se inclusive montar o sistema com quantidade ímpar de unidades. Mas, como isso é possível? Com o uso de um esquema de [paridade \(glossário\)](#).



As informações de paridade — assim como os próprios dados — são distribuídas entre todos os discos do sistema. Via de regra, o espaço destinado à paridade é equivalente ao tamanho de um dos discos.

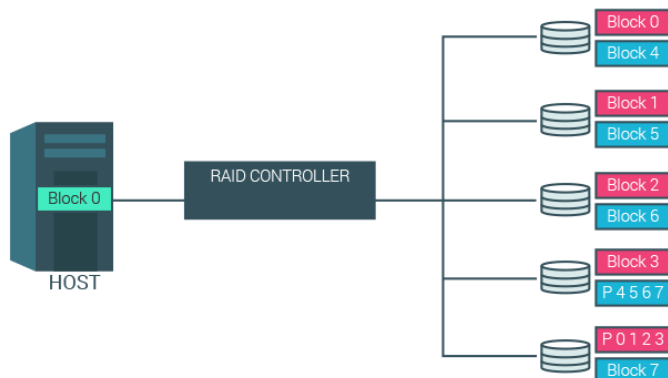
A partir daí, se em uma tarefa de verificação o sistema constatar, por exemplo, que o bit de paridade de um bloco é '1', mas ali há uma quantidade par de bits, percebe que há um erro.

Se houver apenas um bit com problema e se o sistema conseguir identificá-lo, conseguirá substituí-lo imediatamente. A restauração dos dados poderá ser feita inclusive depois de o HD ter sido trocado. Veja o exemplo:

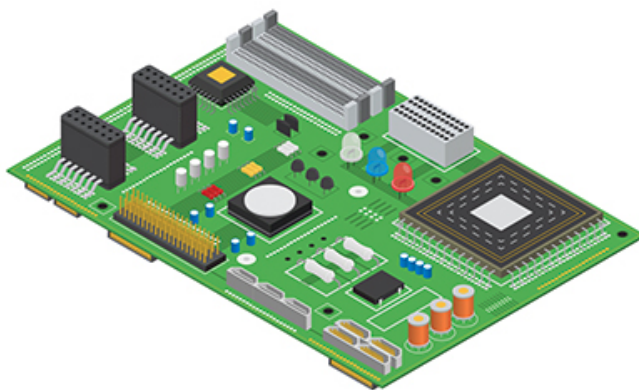




Como exemplo, imagine um bloco de dados com os bits '110X' e paridade '1'. O X indica um bit perdido, mas será que ele é '0' ou '1'? Como a paridade é '1', significa que o bloco é composto por quantidade ímpar de bits '1'. Logo, se X fosse '0', a paridade também deveria ser '0', pois ali existiria quantidade par de bits '1'. Isso significa que o bit X só pode ser '1'.



IMPLEMENTAÇÃO DE RAID



Fonte da Imagem:

Antigamente, montar sistemas RAID não era uma tarefa das mais simples e seu uso normalmente se limitava a servidores. Hoje, no entanto, é possível implementá-los até mesmo em computadores pessoais, mesmo porque praticamente qualquer sistema operacional moderno (Windows, Linux, Mac OS X, entre outros) suporta este recurso.

A maneira mais fácil de fazer isso é adquirindo uma placa-mãe contando com uma controladora RAID. Em poucas palavras, este dispositivo, que pode funcionar com interfaces PATA, SATA ou SCSI, identifica as unidades de armazenamento conectadas e as fazem trabalhar como um sistema RAID. Sua configuração geralmente é feita a partir do setup do BIOS, embora algum software de controle possa ser fornecido para funcionar no sistema operacional.

Se a placa-mãe não possuir controladora RAID, é possível adicionar placas que acrescentam esta função. Estes dispositivos, normalmente, podem ser encontrados utilizando interface PCI ou PCI Express.

Veja um exemplo baseado em Windows.

galeria/aula7/img/slide17_1.jpg

Passo 1

Abra o gerenciador de discos: Iniciar > executar > diskmgmt.msc

Você verá uma tela similar à imagem a seguir, onde estão listados todos os seus discos (internos ou externos).

galeria/aula7/img/slide17_2.jpg

Passo 2

Em um dos discos não alocados, clique com o botão direito do mouse e escolha o tipo de volume que você deseja criar.

É nesse ponto em que você irá escolher o tipo de arranjo:

- Volume Distribuído (RAID-0);

- Volume Espelhado (RAID-1);

- Volume RAID-5.

Neste exemplo, essa opção está esmaecida porque não existem três discos disponíveis para a criação de um RAID-5.



Passo 3

Apenas a tela de boas-vindas do assistente de criação de um novo volume.




Passo 4

Esta é a primeira tela de configuração do arranjo, onde indicamos quais discos farão parte do volume RAID escolhido no passo 2.



Passo 5

Atribuição de uma letra ao volume, essa tela é utilizada para você determinar qual será a letra que irá identificar o novo disco (novo volume).



Passo 6

Tela de formatação do volume, nesse momento, poderemos escolher os seguintes pontos:

Sistema de arquivo:

O recomendável é utilizar o NTFS e não o ExFAT (ou FAT64).

Tamanho da unidade de alocação:

Caso você não saiba antecipadamente o tamanho médio dos arquivos que serão armazenados nessa unidade é melhor deixar como padrão.

Rótulo do volume:

Um nome mais amigável a unidade, pode ser deixado em branco.

Formatação rápida:

Formata o disco sem realizar verificações nos setores do disco.



Passo 7

Tela com resumo das configurações realizadas.



Passo 8

Tela informativa sobre volumes dinâmicos, onde está sendo avisado que o volume que está prestes a ser criado não será capaz de inicializar um sistema operacional.



Passo 9

Nessa última tela, é mostrado o gerenciador de discos com o RAID em funcionamento. Cada tipo de volume possui uma coloração diferente, a verde representa um volume em RAID-0.

Glossário

DISCO SSD

O SSD (solid-state drive) é uma nova tecnologia de armazenamento considerada a evolução do disco rígido (HD). Ele não possui partes móveis e é construído em torno de um circuito integrado semicondutor, uma memória flash, que é responsável pelo armazenamento, diferentemente dos sistemas magnéticos (como os HDs). Esta tecnologia possui vantagens em relação aos HDs, pois, como elimina as partes mecânicas, reduz as vibrações e torna os SSDs completamente silenciosos. Outra vantagem é o tempo de acesso reduzido à memória flash presente nos SSDs em relação aos meios magnéticos e óticos. O SSD também é mais resistente que os HDs comuns devido à ausência de partes mecânicas.

ARMAZENAMENTO NÃO VOLÁTIL

Tipo de armazenamento que independe de alimentação de energia para manter seu conteúdo e, portanto permanece gravado mesmo após ser desligado o computador, mas que pode, a critério do usuário, ser apagado ou alterado.

FORMATÇÃO

O processo de marcação magnética das trilhas e dos setores, em um disco, faz parte da "formatação" do disco. Esta formatação é dependente do sistema operacional que usará o disco.

CILINDRO

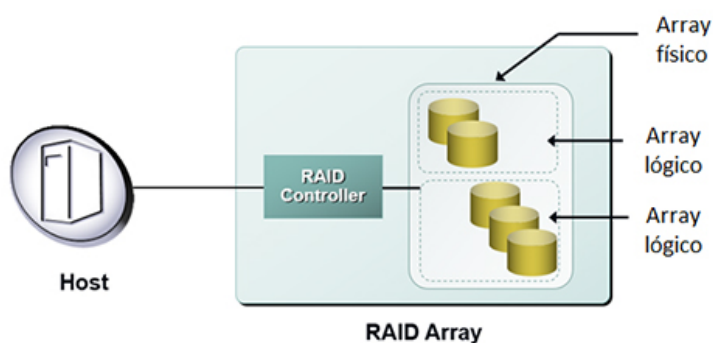
O conjunto de trilhas localizado a uma mesma distância do eixo central. Desta forma, em um disco de quatro cabeças (quatro faces) o cilindro 10 seria composto por todos os setores localizados na trilha 10 da face 1, na trilha 10 da face 2, na trilha 10 da face 3 e na trilha 10 da face 4. Os dados gravados no mesmo cilindro (na mesma trilha, porém em superfícies diferentes) podem ser acessados (lidos ou gravados) sem que o braço atuador das cabeças de leitura/gravação tenha que ser movido.

UNIDADE LÓGICA

É forma como o SO enxerga o armazenamento, normalmente aparece como uma letra Drive C, Drive D etc.

No caso do RAID, a ideia é fazer com que o sistema operacional enxergue o conjunto de HDs como uma única unidade de armazenamento, independente da quantidade de dispositivos que estiver em uso.

Podem ser utilizados também disco de Estado Sólido (SSD).



MAIS DISCOS = MAIOR TAXA DE TRANSFERÊNCIA

Não é difícil entender o porquê: como os dados são divididos, cada parte de um arquivo é gravada em unidades diferentes ao mesmo tempo. Se este processo acontecesse apenas em um único HD, a gravação seria um pouco mais lenta, já que teria que ser feita sequencialmente.

PARIDADE

Neste método de proteção, os dados são divididos em pequenos blocos. Cada um deles recebe um bit adicional — o bit de paridade — de acordo com a seguinte regra: se a quantidade de bits '1' do bloco for par, seu bit de paridade é '0'; se a quantidade de bits '1' for ímpar, o bit de paridade é '1'.