SISTEMAS OPERACIONAIS

AULA 19 – GERENCIAMENTO DE ENTRADA/SAÍDA, PARTE 2

Prof.ª Sandra Cossul, Ma.

TÓPICOS DA AULA

- Estruturas de armazenamento de dados
  - HD e SSD
  - Estrutura física
  - Algoritmos de escalonamento de disco - HD
- RAID

DISCOS RÍGIDOS

- Dispositivos de armazenamento permanente destinados a grandes quantidades de programas e dados do usuário.
- Foram introduzidos em torno de 1981 pela IBM.
- Desde então, os HDs foram melhorando em aumento de **capacidade** e **confiabilidade**, além de **aumento da taxa de transferência** e **redução de custo.**

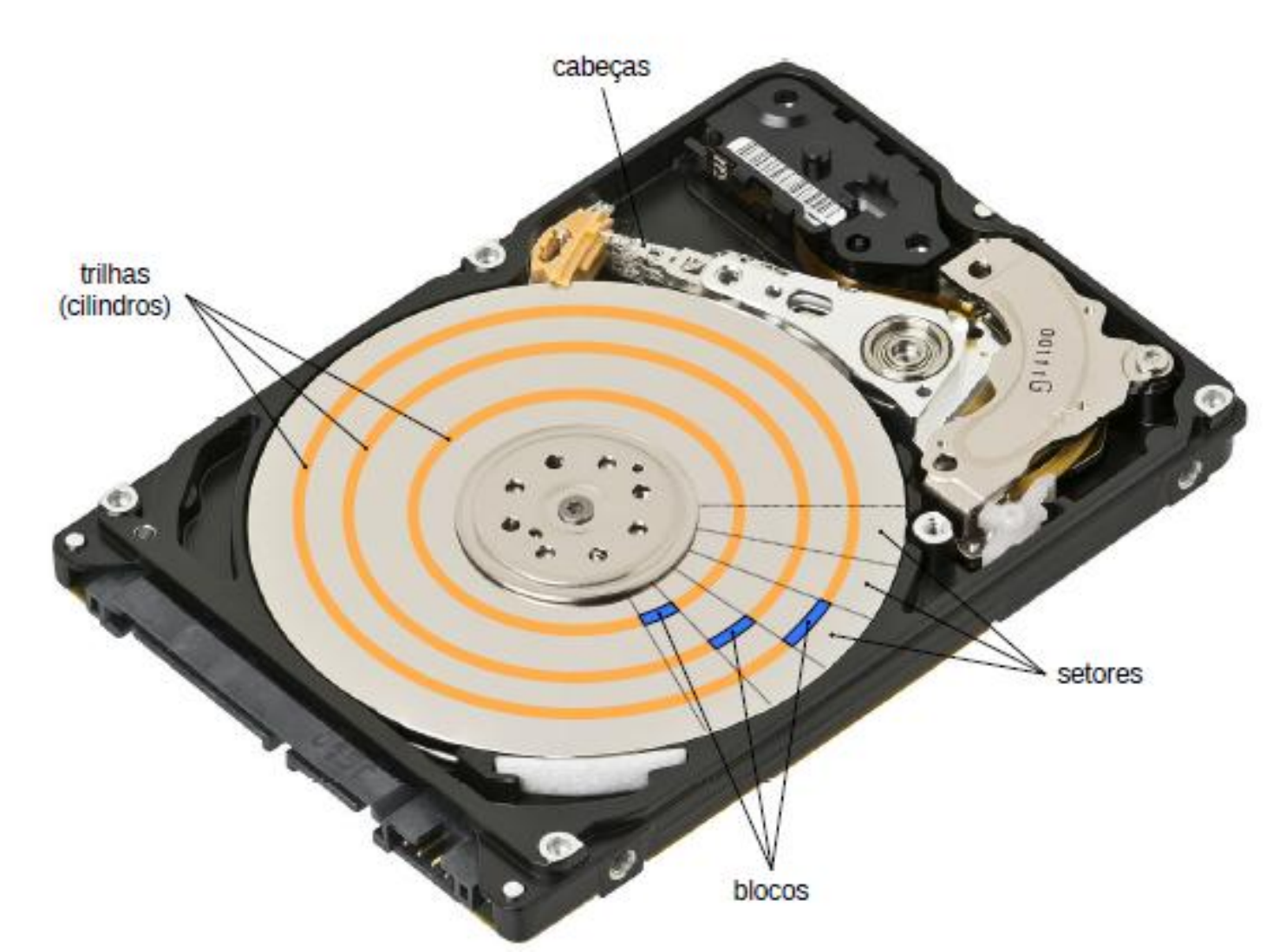
DISCO MAGNÉTICO (DISCO RÍGIDO) - HD

- Composto por **discos metálicos (pratos)** que giram juntos em alta velocidade (entre 4200 e 15000 rpm), acionados por um motor elétrico.

- **Cabeça de leitura móvel** para cada face de cada disco, responsável por ler e escrever dados através da **magnetização** de pequenas áreas da superfície metálica.

- Cada face é dividida logicamente em **trilhas e setores;** a interseção entre uma trilha e um setor define um **bloco físico,** que é a unidade básica de armazenamento e transferência de dados nos discos.
  - Padrão atual – blocos de 4096 bytes (4KB).

DISCO MAGNÉTICO (DISCO RÍGIDO) - HD



DISCO MAGNÉTICO (DISCO RÍGIDO) - HD

- **Cabeça leitura/gravação**
- É uma bobina condutora (eletro-imã) móvel utilizada para gravar e recuperar dados.

- Utiliza pulsos magnéticos para magnetizar uma área de gravação e interpretar a direção do campo magnético.

- Direção do campo magnético: bit 0 ou bit 1.

DISCO MAGNÉTICO (DISCO RÍGIDO) - HD

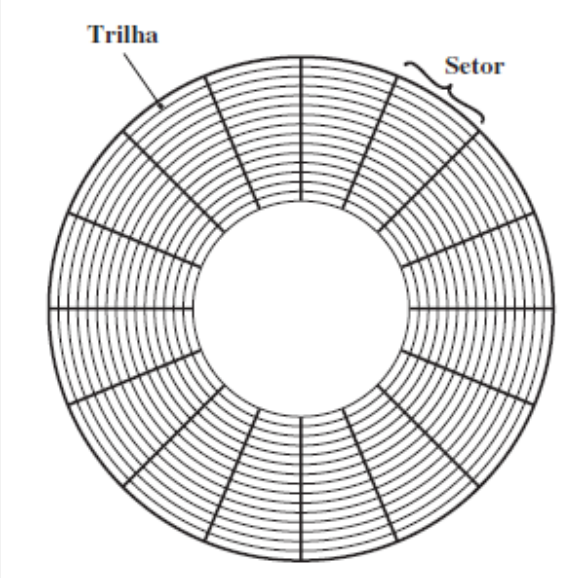
- **Placa lógica**
  - **Chip controlador,** gerencia uma série de ações, como a movimentação dos discos e da cabeça de leitura/gravação, o envio e recebimento de dados entre os discos e o processador/RAM, e até rotinas de segurança.
  - **Buffer,** chip de memória *cache*, armazena dados para acelerar a troca de dados (HD atuais: 256 MB+)

ORGANIZAÇÃO DOS DADOS - HD

- Um disco típico contém várias **faces** e milhares de **trilhas e setores por face**, resultando em **milhões de blocos de dados** disponíveis.

- Cada **bloco** pode ser individualmente acessado (lido ou escrito) através de seu **endereço.**

- A conversão de endereços é realizada pelo firmware do disco, de forma transparente para o sistema.



PARÂMETROS DE DESEMPENHO - HD

- Para realizar um acesso a um disco rígido, é necessário posicionar a cabeça de leitura e escrita sob um determinado setor e trilha onde o dado será lido ou escrito.

- **Tempo de busca:** tempo gasto para a cabeça de leitura se posicionar sobre uma determinada trilha

- **Atraso rotacional (latência rotacional):** tempo gasto para o disco girar até que o setor desejado esteja sob a cabeça de leitura
  - Depende da velocidade de rotação

- **Tempo/Taxa de transferência:** tempo para realizar a transferência de dados.

TIPOS DE DISCO RÍGIDO

- **Disco rígido Desktop**
  - SATA
  - Cache
  - 3,5 polegadas

- **Disco rígido Notebook**
  - SATA
  - Cache
  - 2,5 polegadas

TIPOS DE DISCO RÍGIDO

- **HD externo**
  - USB 3.0

- **Outros**
  - Vigilância
  - Massive Storage

MEMÓRIA FLASH – SSD, PENDRIVES

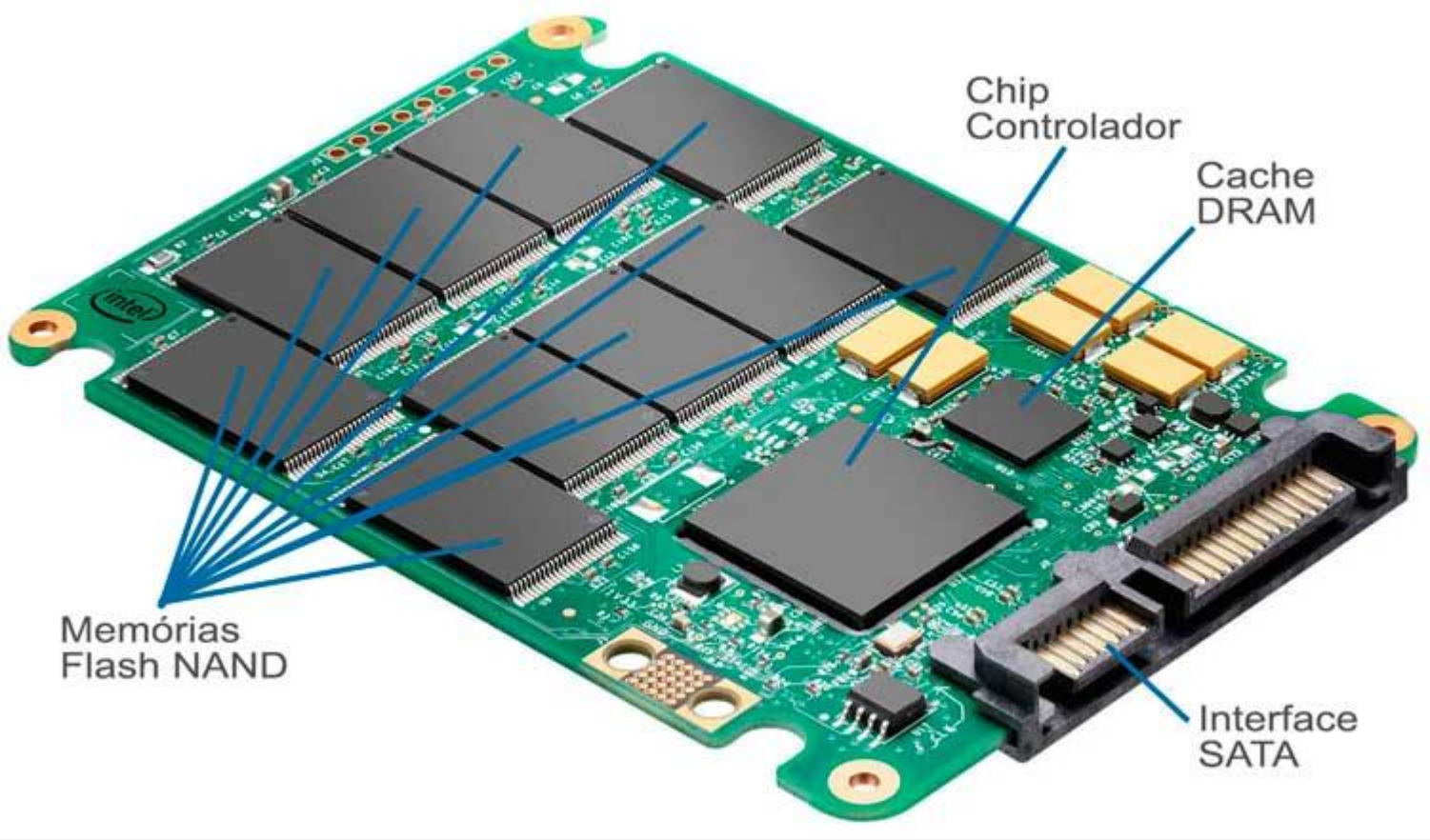
- Vem substituindo os discos rígidos
- São dispositivos eletrônicos e não mecânicos

- **Controlador +** chips semicondutores **flash NAND**

- Utilizados em discos de estado sólido (**SSD**) ou drivers USB (pendrives)

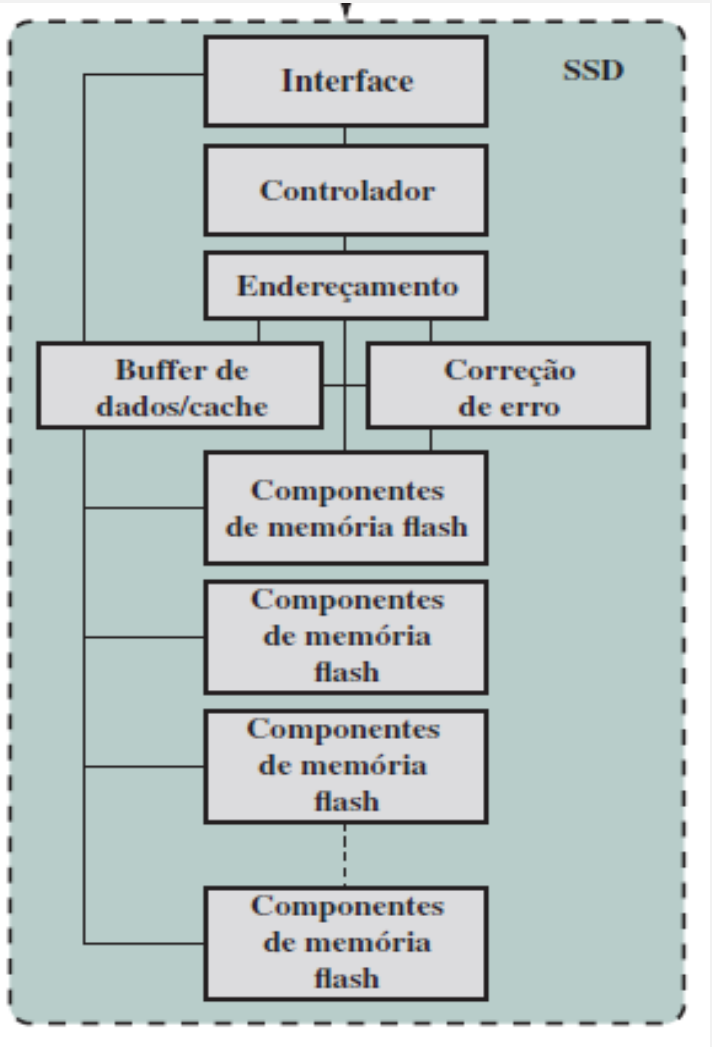
- São **mais confiáveis e mais rápidos** do que HDs pois não possuem partes móveis além de apresentar um **menor consumo de energia**.
- No entanto, tem um **custo maior e menor capacidade** (vem diminuindo a diferença de custo e capacidade em relação aos HDs)

MEMÓRIA FLASH – SSD, PENDRIVE



SSD - ARQUITETURA

- **Controlador**: proporciona o interfaceamento e a execução do firmware do dispositivo de SSD.
- **Endereçamento**: a lógica que apresenta a função de seleção nos componentes de memória flash.
- **Buffer de dados/cache**: componentes de memória RAM de alta velocidade usados para combinação da compatibilização da velocidade e para o aumento da taxa de transferência (throughput) de dados.
- **Correção de erros**: a lógica para a detecção e correção de erros.
- **Componentes de memória flash**: chips individuais de flash NAND.



SSD - FATOR DE FORMA

- 2,5”
- M.2

SSD - INTERFACES

- **SATA**
- **PCI Express**
  - NVMe

- **Obs.:** NVMe é um protocolo de conexão, não um formato. Por exemplo, existe SSD M.2 do tipo SATA e NVMe (PCIe), mas a conexão suporta apenas um tipo.

OPERAÇÕES E/S

- Um proceso realiza uma **operação de E/S** em disco através de uma **chamada de sistema**, fornecendo parâmetros como o **tipo de operação** (leitura ou escrita) e os **dados a serem escritos,** o que se reflete em posições diferentes em trilhas e setores do disco

- **Sistemas multiprogramados –** vários processos realizando operações de E/S simultaneamente (sendo bloqueados até que a operação solicitada seja realizada)

- **Como ordenar e atender os pedidos de E/S de forma a maximizar o atendimento e minimizar o tempo em que processos permanecerão bloqueados?**

OPERAÇÕES E/S - HD

- Tempo de uma operação de E/S → tempo de acesso ao disco

- Objetivo:
  - **Minimizar os movimentos da cabeça de leitura (tempo de acesso)**
  - **Maximizar o nº de bytes transferidos** (atender o maior nº possível de requisições no menor tempo possível)

- Solução:
  - **Algoritmos para realizar as movimentações do disco** (chamados de algoritmos de escalonamento de disco)

OPERAÇÕES E/S - HD

- A **ordem de atendimento das requisições** pendentes na fila de acesso ao disco é denominada **escalonamento de disco** e pode ter um grande **impacto no desempenho** do sistema operacional.

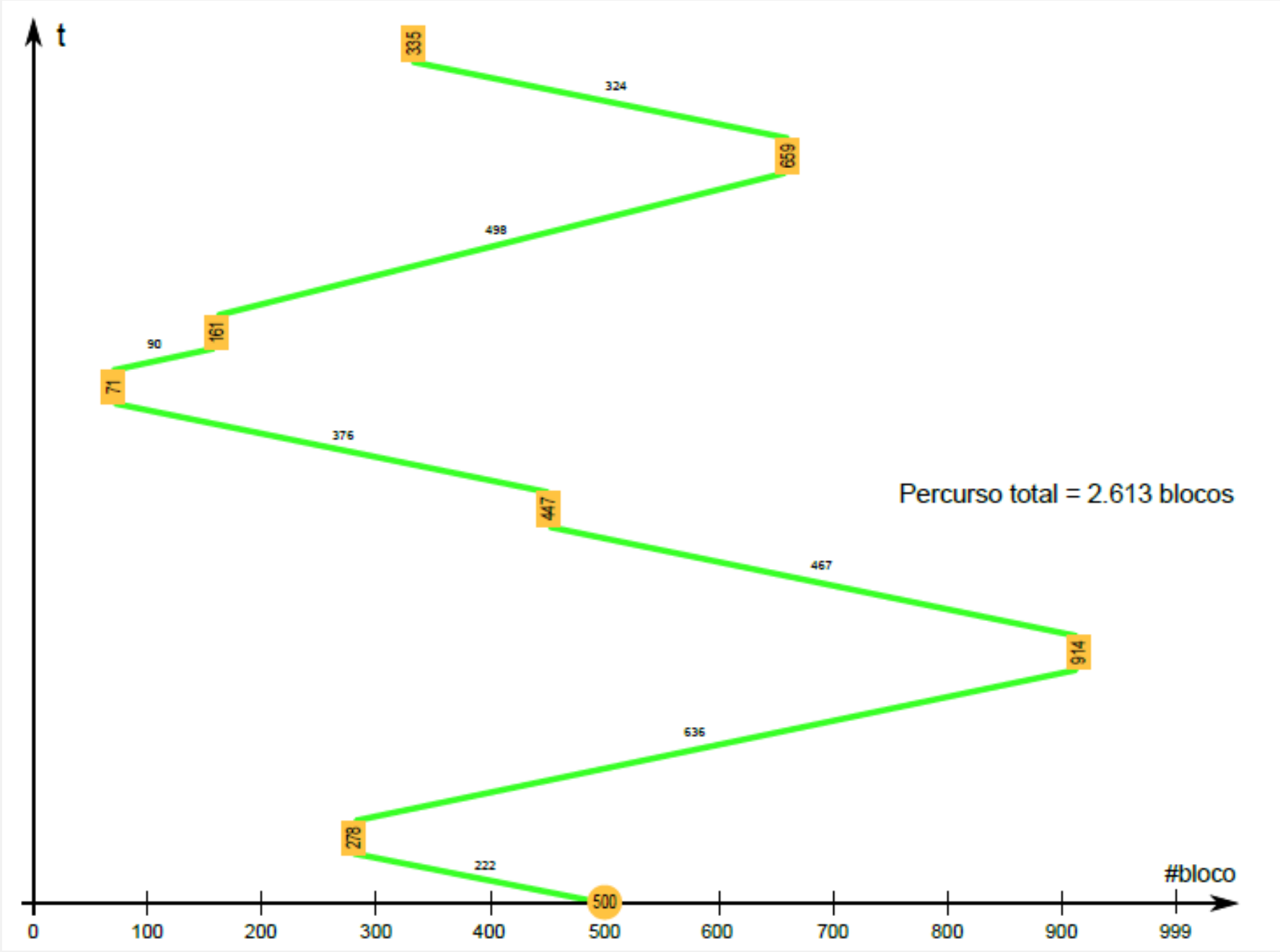
- **Exemplo funcionamento:**
  - Disco com 1000 blocos (0 a 999)
  - Cabeça de leitura se encontra inicialmente sobre o bloco 500
  - Fila de acessos com pedidos aos seguintes blocos do disco:
    - 278 – 914 – 447 – 71 – 161 – 659 – 335

ALGORITMOS DE ESCOLANAMENTO DE DISCO

- **FCFS – first come first served**
  - Mais simples: as solicitações de acesso ao disco são realizadas na ordem em que os pedidos são feitos
  - Nenhuma tentativa é feita para reorganizar a ordem dos pedidos visando a otimizar os movimentos da cabeça de leitura e escrita
  - Se os pedidos de acesso estiverem muito espalhados pelo disco, perde muito tempo com movimentações

$$500 \xrightarrow{222} 278 \xrightarrow{636} 914 \xrightarrow{467} 447 \xrightarrow{376} 71 \xrightarrow{90} 161 \xrightarrow{498} 659 \xrightarrow{324} 335 \text{ (2.613 blocos)}$$

- **FCFS**

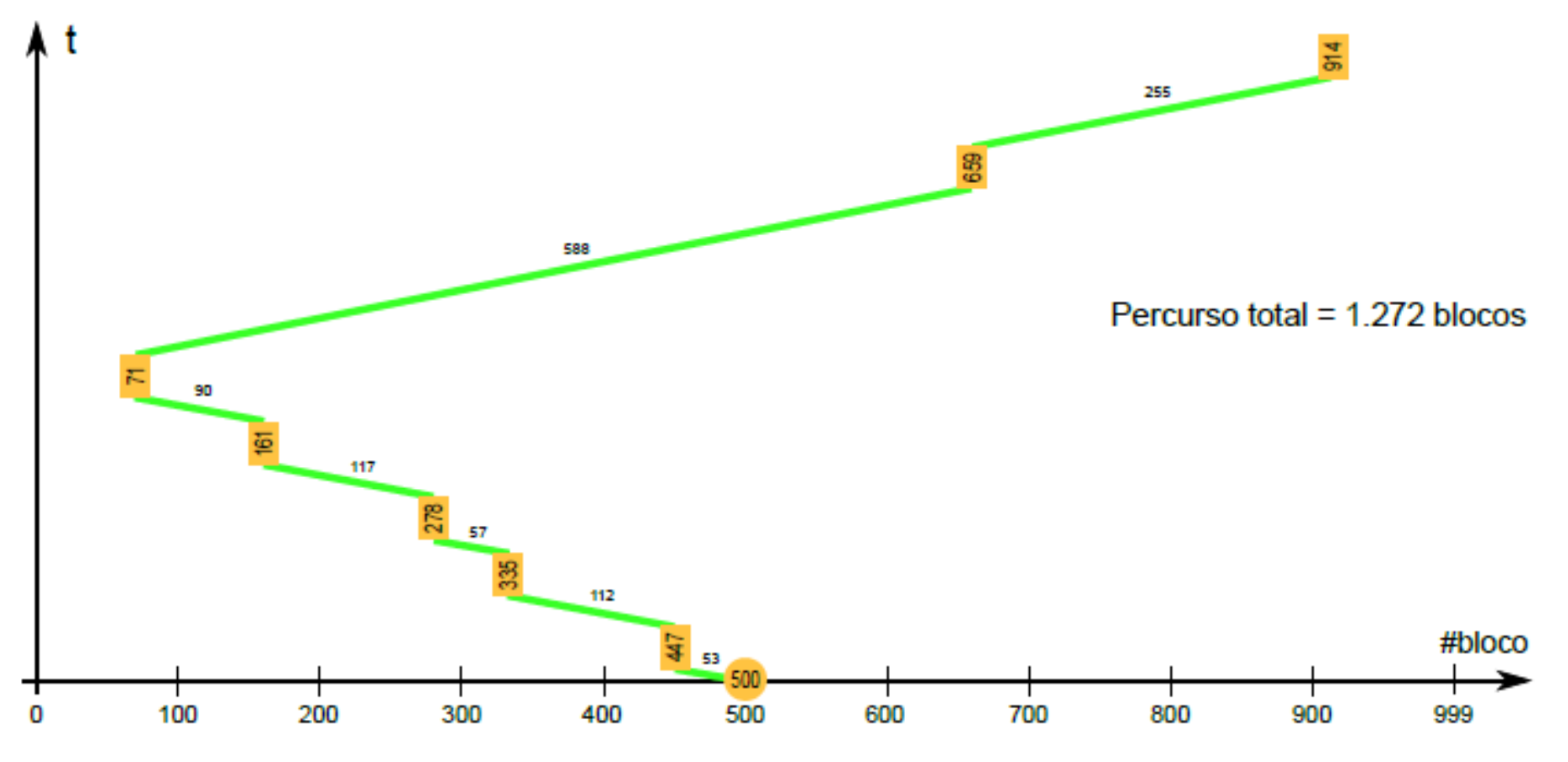

t
335
324
659
498
161
90
71
376
447
467
914
636
278
222
500
Percurso total = 2.613 blocos
#bloco
0
100
200
300
400
500
600
700
800
900
999

ALGORITMOS DE ESCOLANAMENTO DE DISCO

- **SSTF – shortest seek time first – menor tempo de busca primeiro**
  - A fila de pedidos é reordenada para atender as solicitações de forma a minimizar o movimento da cabeça de leitura e escrita
  - Novos pedidos são ordenados em relação à posição atual da cabeça de leitura e escrita, privilegiando assim o acesso aos blocos que estão mais próximos a posição do último pedido atendido
  - Pode levar a *starvation* de requisições de acessos – caso existam muitas requisições em uma determinada região, pedidos de acesso a blocos distantes podem ficar esperando indefinidamente
    - Utilizar estratégia de envelhecimento dos pedidos pendentes

- **SSTF**

$$500 \xrightarrow{53} 447 \xrightarrow{112} 335 \xrightarrow{57} 278 \xrightarrow{117} 161 \xrightarrow{90} 71 \xrightarrow{588} 659 \xrightarrow{255} 914 \text{ (1.272 blocos)}$$

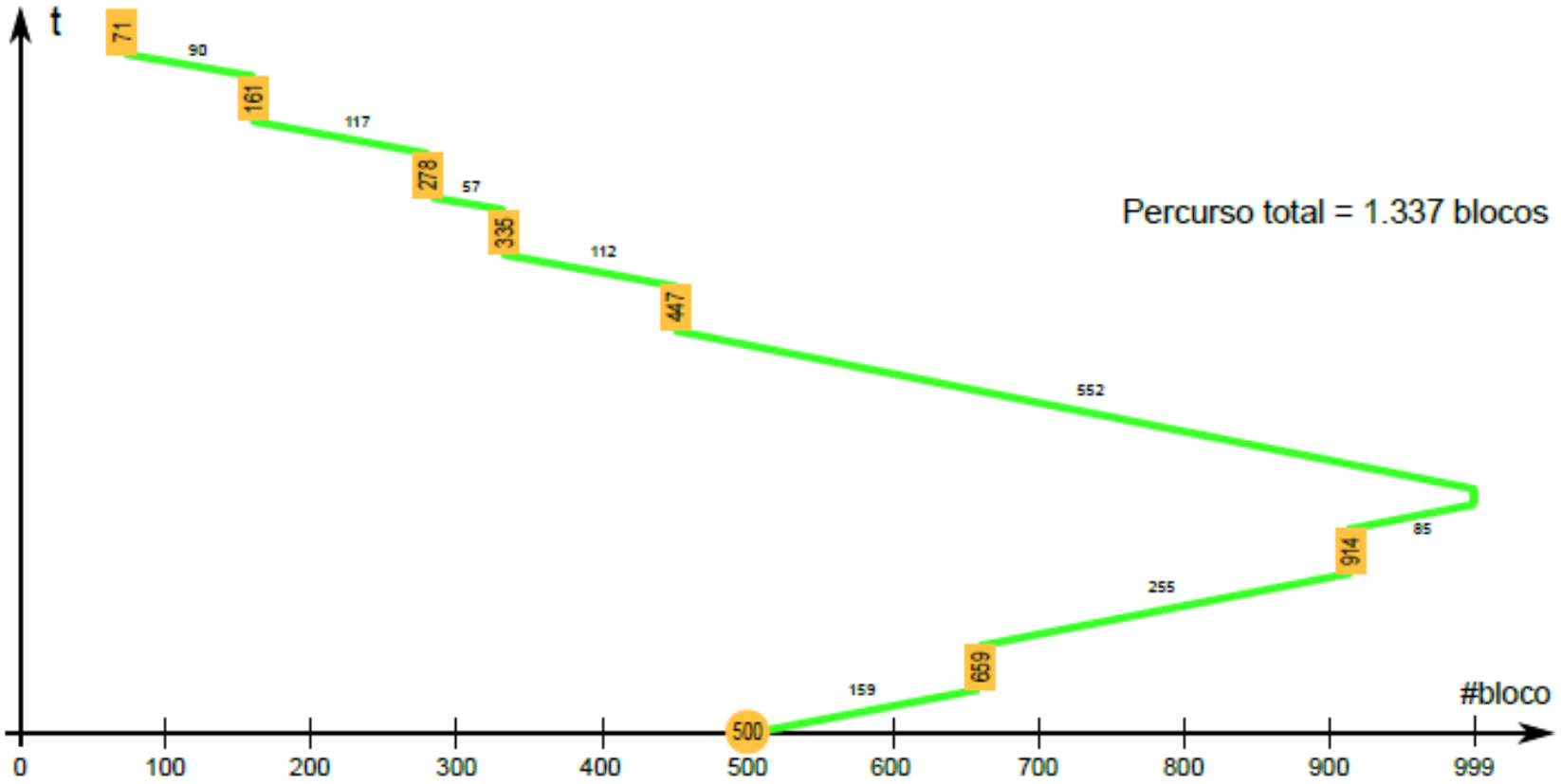


ALGORITMOS DE ESCOLANAMENTO DE DISCO

- **SCAN**
  - Neste algoritmo, a cabeça de leitura/escrita "varre" (*scan*) continuamente o disco, do início ao final, atendendo os pedidos que encontra pela frente; ao atingir o final do disco, ela inverte seu sentido de movimento e volta, atendendo os próximos pedidos.
    - "**Varre para um lado e depois inverte"**
  - Apesar de ser mais lento que o SSTF, **atende os pedidos de forma mais uniforme** ao longo do disco, eliminando o risco de *starvation*
    - Mantém um desempenho equilibrado para todos os processos
  - Adequado para sistemas com muitos pedidos simultâneos de acesso a disco, como servidores
  - Algoritmo do elevador – reproduz comportamento de um elevador

- **SCAN**

$$500 \xrightarrow{159} 659 \xrightarrow{255} 914 \xrightarrow{85} 999 \xrightarrow{552} 447 \xrightarrow{112} 335 \xrightarrow{57} 278 \xrightarrow{117} 161 \xrightarrow{90} 71 \text{ (1.337 blocos)}$$



ALGORITMOS DE ESCOLANAMENTO DE DISCO

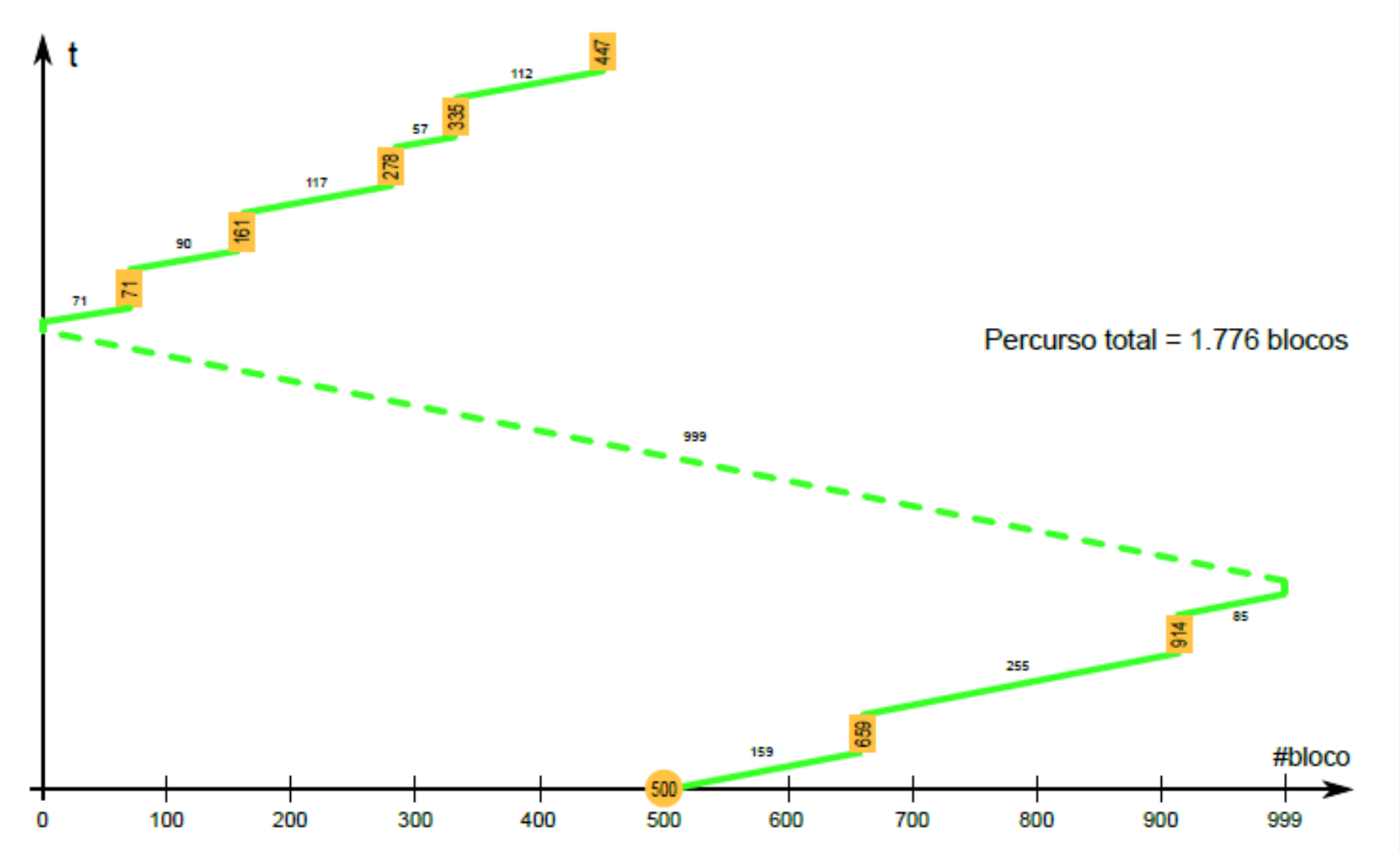
- **C-SCAN (circular SCAN)**
  - Variante "circular" do algoritmo SCAN

  - Varredura do disco ocorre somente em um sentido
  - Ao atingir o final do disco, retorna imediatamente ao início do disco, sem atender os pedidos intermediários, e recomeça a varredura
  - Circular – disco é visto como uma lista circular de blocos

  - Vantagem em relação ao SCAN: prover um tempo de espera mais homogêneo aos pedidos pendentes, o que é importante em servidores

- **C-SCAN**

$$500 \xrightarrow{159} 659 \xrightarrow{255} 914 \xrightarrow{85} 999 \xrightarrow{999} 0 \xrightarrow{71} 71 \xrightarrow{90} 161 \xrightarrow{117} 278 \xrightarrow{57} 335 \xrightarrow{112} 447 \text{ (1.776 blocos)}$$



ESCALONAMENTO - SSD

- Os algoritmos comentados anteriormente se aplicam a discos rígidos (HD)
- Foco na diminuição do movimento do disco

- Como essa não é uma preocupação de memórias flash por não conter partes móveis, **normalmente utilizam uma política de escalonamento FCFS**

- Acesso sequencial é ótimo para HD porque os dados estão próximos
- Já no acesso aleatório causa muitas movimentações de disco, e isso não é um problema para SSD

DETECÇÃO E CORREÇÃO DE ERROS

- Os dispositivos de armazenamento implementam métodos para detectar e corrigir erros

- **Detecção de erros –** identificar se um bloco de dados teve alguma alteração desde que foi escrito
- Após a detecção, o sistema pode reportar o erro, parar a operação ou avisar que um dispositivo está falhando.
- **Ex.:** bits de paridade

- **Correção de erros –** corrigir erros (ex.: código de hamming)

ESTRUTURA RAID

- **RAID – redundant array of independent disks**
- Aumento do volume de dados → uso de várias unidades físicas de discos → aumenta a probabilidade de que um desses discos apresente problemas físicos → **perda de dados**

- **O princípio básico de uma estrutura RAID é combinar vários discos rígidos físicos em uma estrutura lógica de discos de forma a aumentar a confiabilidade e o desempenho dos discos.**
  - O conjunto de discos armazena informações de forma redundante, permitindo a recuperação de dados em caso de falha física de um disco.
  - Desempenho obtido através da escrita em paralelo nos diferentes discos que compõem a estrutura RAID.

ESTRUTURA RAID

- A tecnologia RAID é utilizada para dois objetivos:
  - **Melhorar a confiabilidade por meio da redundância** (espelhamento de dados)
  - **Melhorar a performance por meio do paralelismo**

- Um sistema RAID é constituído de dois ou mais discos rígidos que são vistos pelo SO e pelas aplicações como um único disco lógico, ou seja, um grande espaço contíguo de armazenamento de dados

- Pode ser utilizado também em SSDs, mas são menos suscetíveis a falhas em comparação a HDs.

ESTRUTURA RAID

- Há várias formas de se organizar um conjunto de discos rígidos em RAID, cada uma com suas próprias características de desempenho e confiabilidade.

- Essas formas de organização são usualmente chamadas **Níveis RAID**

- A escolha do nível de RAID dependerá das necessidades específicas de armazenamento de cada usuário ou da aplicação.

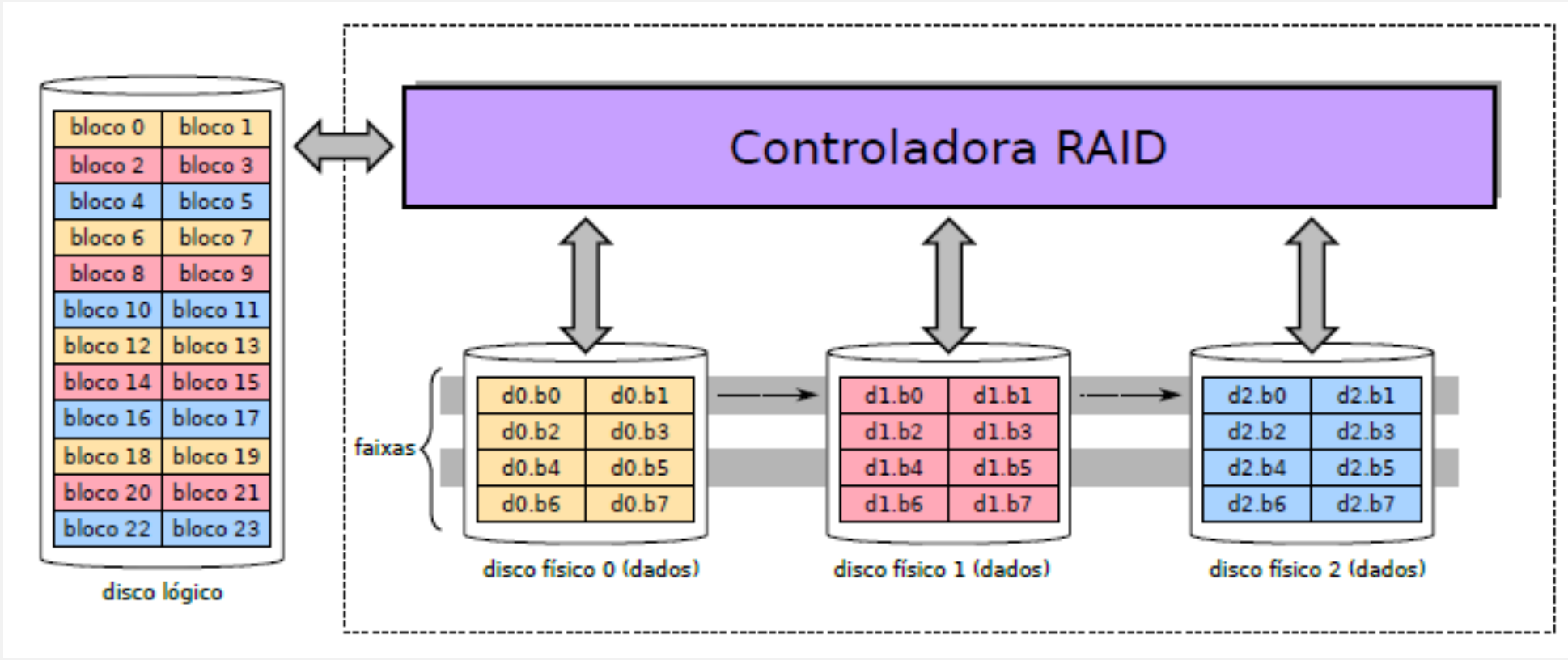
RAID 0

- Utilizado para melhorar o desempenho de leitura e gravação, dividindo os dados em blocos e armazenando-os em vários discos simultaneamente (acesso paralelo).

- Não oferece redundância de dados, o que significa que a perda de um disco resultará na perda de todos os dados.

- Conhecido como “striping”

RAID 0 - STRIPING



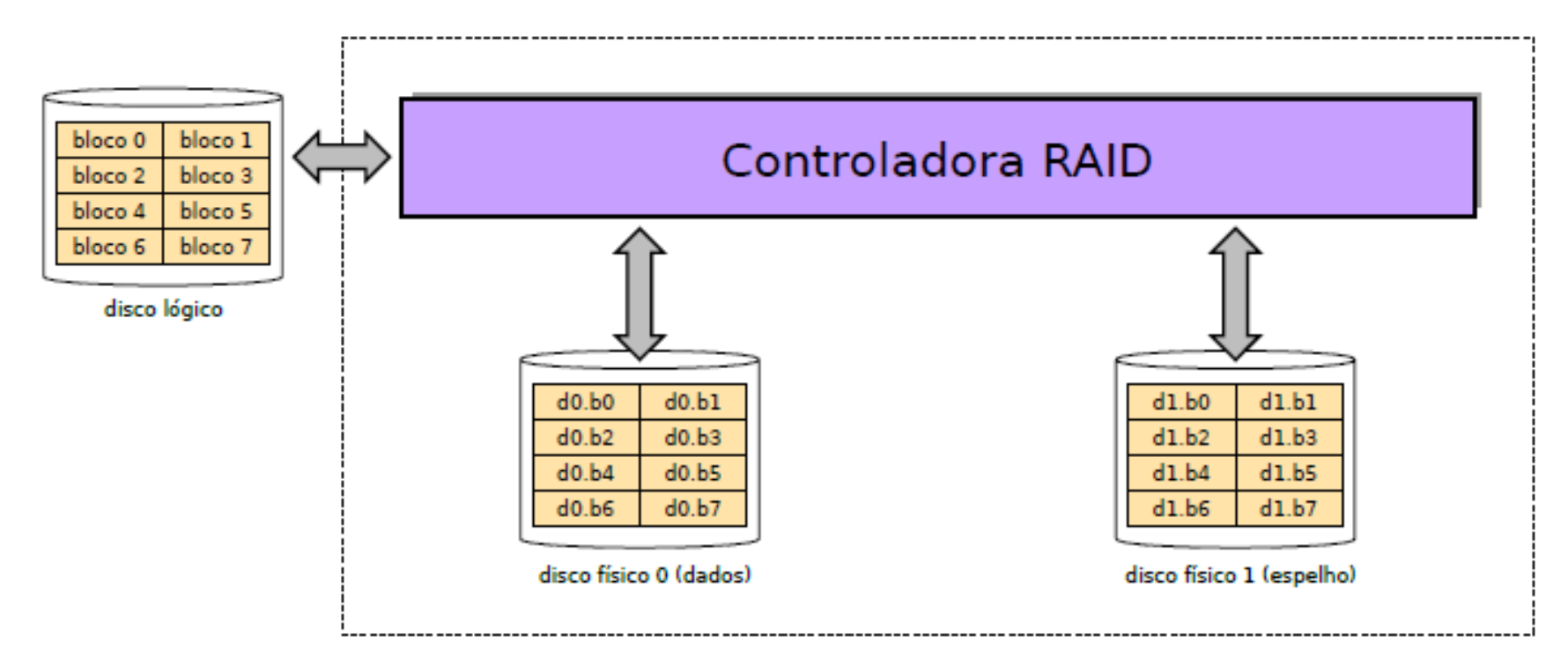
RAID 1

- Neste nível, **o conteúdo é replicado em dois ou mais discos**, sendo por isso comumente chamado de espelhamento de discos.

- Esta abordagem oferece uma excelente **confiabilidade**, pois cada bloco lógico está escrito em dois ou mais discos distintos; caso um deles falhe, os demais continuam acessíveis;

- O desempenho em leituras também é beneficiado, pois a controladora pode distribuir as leituras entre as cópias dos dados.
  - Não há ganho de desempenho em escrita, pois cada operação é replicada em todos os discos

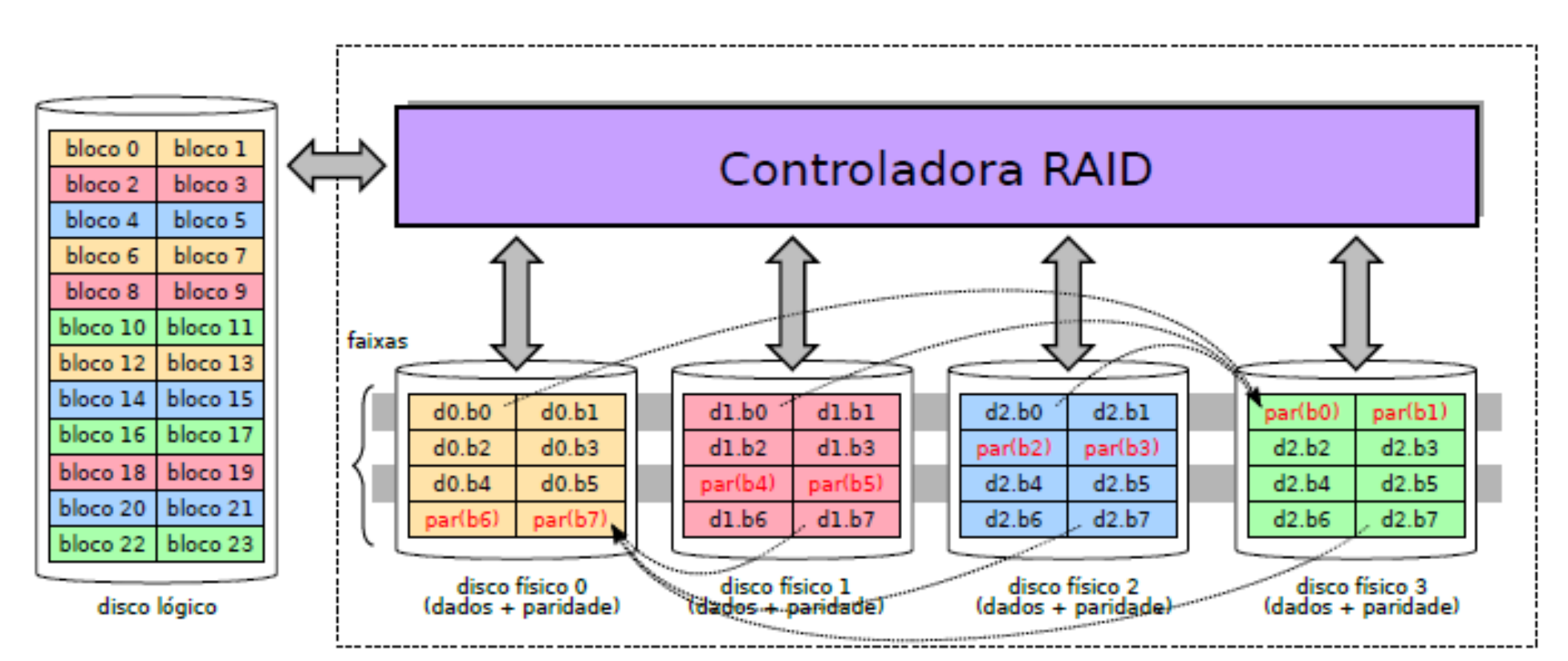
RAID 1


bloco 0
bloco 1
bloco 2
bloco 3
bloco 4
bloco 5
bloco 6
bloco 7
disco lógico
Controladora RAID
d0.b0
d0.b1
d0.b2
d0.b3
d0.b4
d0.b5
d0.b6
d0.b7
disco físico 0 (dados)
d1.b0
d1.b1
d1.b2
d1.b3
d1.b4
d1.b5
d1.b6
d1.b7
disco físico 1 (espelho)

RAID 5

- Usa distribuição de dados em blocos em vários discos, incluindo **paridade** para permitir a recuperação de dados em caso de falha de um dos discos.
  - Informações de paridade são distribuídas uniformemente entre os discos
- Requer um mínimo de três discos para ser configurado e pode tolerar a falha de um disco sem perda de dados.
- Abordagem popular, por oferecer um bom desempenho e redundância de dados, desperdiçando menos espaço que o espelhamento (RAID 1)

RAID 5


bloco 0
bloco 1
bloco 2
bloco 3
bloco 4
bloco 5
bloco 6
bloco 7
bloco 8
bloco 9
bloco 10
bloco 11
bloco 12
bloco 13
bloco 14
bloco 15
bloco 16
bloco 17
bloco 18
bloco 19
bloco 20
bloco 21
bloco 22
bloco 23
disco lógico
Controladora RAID
faixas
d0.b0
d0.b1
d0.b2
d0.b3
d0.b4
d0.b5
par(b6)
par(b7)
d1.b0
d1.b1
d1.b2
d1.b3
par(b4)
par(b5)
d1.b6
d1.b7
d2.b0
d2.b1
par(b2)
par(b3)
d2.b4
d2.b5
d2.b6
d2.b7
par(b0)
par(b1)
d2.b2
d2.b3
d2.b4
d2.b5
d2.b6
d2.b7
disco físico 0
(dados + paridade)
disco físico 1
(dados + paridade)
disco físico 2
(dados + paridade)
disco físico 3
(dados + paridade)

RAID 6

- Semelhante ao RAID 5, mas com a adição de um segundo esquema de paridade.

- O RAID 6 requer um mínimo de quatro discos para ser configurado e pode tolerar a falha de dois discos sem perda de dados.

RAID 10

- Striping + espelhamento
- Combina os benefícios do RAID 0 e RAID 2
- Os dados são divididos em blocos e armazenados em vários discos simultaneamente, oferecendo maior desempenho, enquanto as cópias idênticas dos dados são armazenadas em outros discos, oferecendo redundância de dados e tolerância a falhas.

BIBLIOGRAFIA

- Tanenbaum, A. S. **Sistemas Operacionais Modernos.** Pearson Prentice Hall. 3rd Ed., 2009.
- Silberschatz, A; Galvin, P. B.; Gagne G.; **Fundamentos de Sistemas Operacionais.** LTC. 9th Ed., 2015.
- Stallings, W.; **Operating Systems: Internals and Design Principles**. Prentice Hall. 5th Ed., 2005.
- Oliveira, Rômulo, S. et al. **Sistemas Operacionais** - V11 - UFRGS. Disponível em: Minha Biblioteca, Grupo A, 2010.