

# Sistemas Operacionais

## Gestão de entrada/saída - discos rígidos

Prof. Carlos Maziero

DInf UFPR, Curitiba PR

Agosto de 2020

# Conteúdo

- 1 Discos rígidos
- 2 Escalonamento de disco
- 3 Escalonadores no Linux
- 4 Sistemas RAID

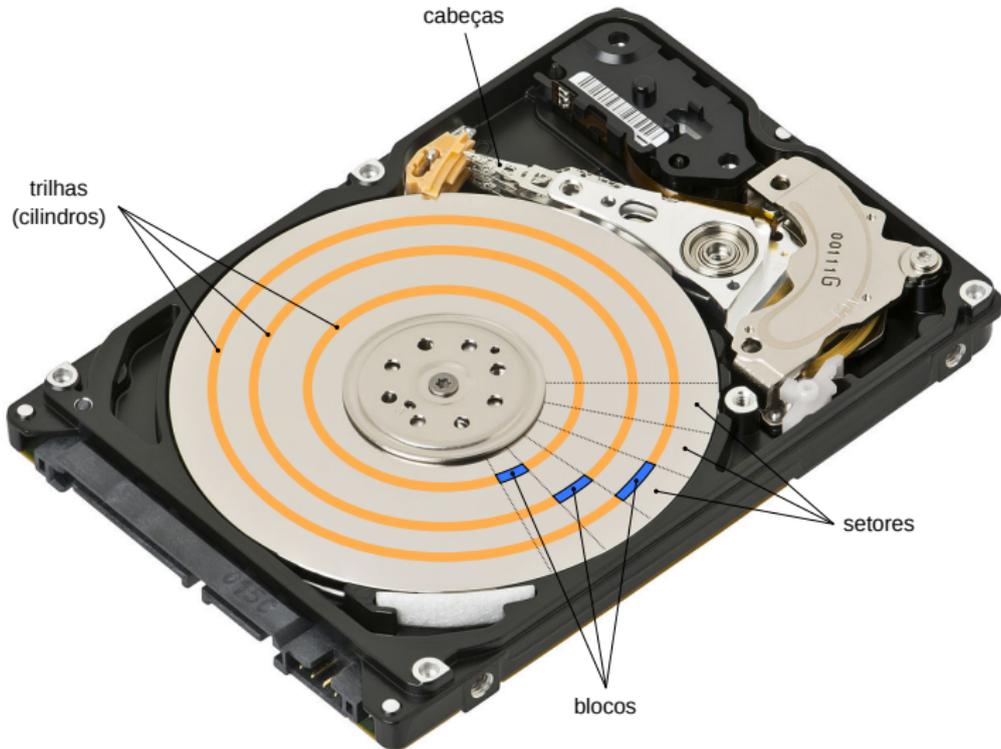
# Discos rígidos

Dispositivo de armazenamento **magnético**

Características:

- Criado em 1954
- Um ou mais discos metálicos
- Velocidade de rotação entre 4.200 e 15.000 RPM
- Capacidade entre 100s GB e 10 TB
- Taxa de transferência entre 0.5 e 2 Gbps
- Latência entre 2 e 10 ms

# Estrutura física



# Estrutura lógica do disco rígido

Estrutura:

- Faces (ou cabeças): duas por disco metálico
- Trilhas (ou cilindros): faixas concêntricas
- Setores: “fatias” angulares

Blocos físicos:

- Interseção entre cabeça, trilha e setor
- Tamanho fixo de 512 ou 4.096 bytes

Endereçamento dos blocos:

- Esquema CHS: *Cylinder, Head, Sector* (interno)
- Esquema LBA: *Large Block Array* (*firmware* ou BIOS)

# Interface de acesso

Padrões de interface do controlador:

Padrão	velocidade	protocolo	aplicação	status
IDE	1 Gbit/s	paralelo	desktops	obsoleto
SCSI	2,5 Gbit/s	paralelo	servidores	obsoleto
SATA	6 Gbit/s	serial	desktops	atual
SAS	12 Gbit/s	serial	servidores	atual

Estrutura do driver:

- Interação por eventos e DMA
- Transfere grupos de blocos físicos (*clusters*)
- Blocos lógicos ou *clusters*: 512 a 64 K Bytes

# Escalonamento de acessos

O disco é um dispositivo **lento!**

- Latência rotacional  $t_r \approx 5ms$
- Tempo de busca  $t_s \approx 10ms$  (*seek time*)

O disco é um dispositivo **sequencial**: trata um pedido por vez!

Tratamento dos pedidos de acesso ao disco:

- Pedidos dos processos são mantidos em uma fila
- A fila é organizada de acordo com um algoritmo
- Busca-se **desempenho** e **justiça**

# Algoritmos de escalonamento clássicos

- FCFS - *First Come, First Served*
- SSTF - *Shortest Seek-Time First*
- SCAN, C-SCAN, LOOK e C-LOOK (“elevador”)

Exemplo: fila de pedidos de acesso aos blocos:

278, 914, 447, 71, 161, 659, 335

Cabeça do disco se encontra no bloco 500

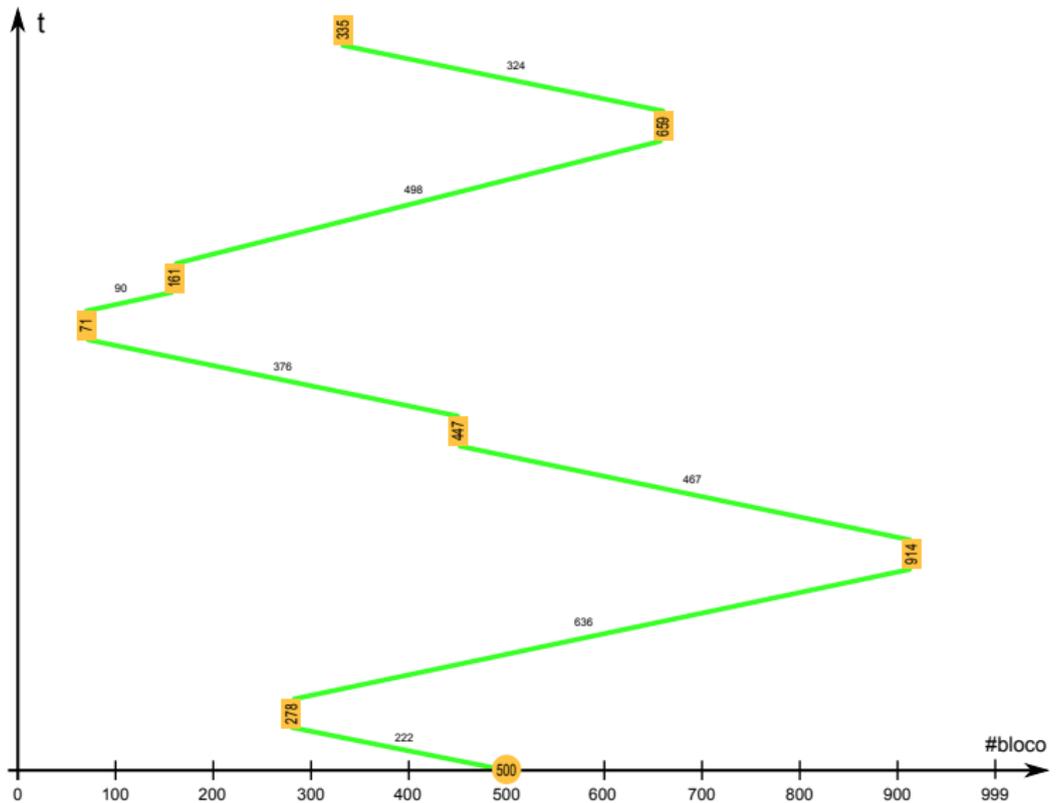
# Escalonamento FCFS

Atender as requisições **na ordem** em que foram emitidas.

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

Deslocamento da cabeça: 2.613 blocos

# Escalonamento FCFS



# Escalonamento SSTF

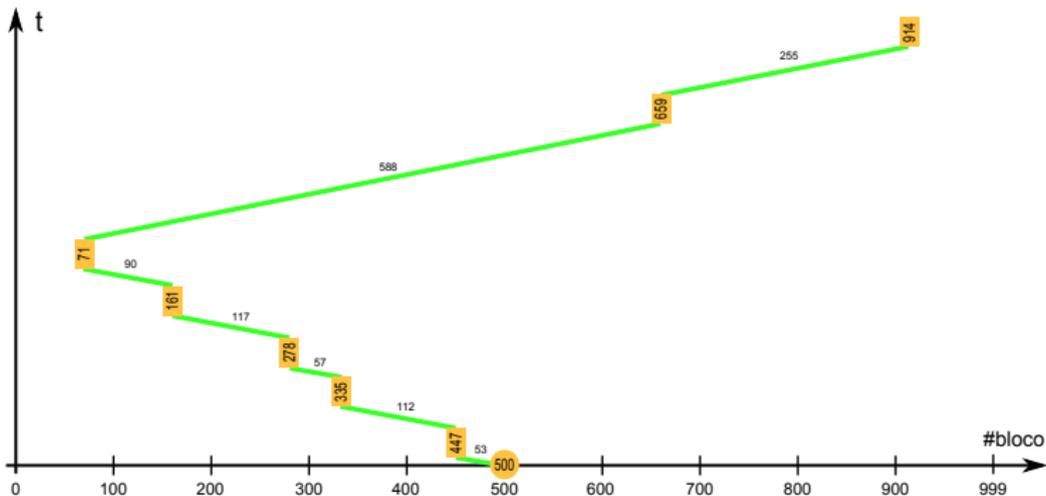
*Shortest Seek Time First*: menor tempo de busca primeiro.

Atender o pedido que está **mais próximo** da cabeça.

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

Deslocamento da cabeça: 1.272 blocos

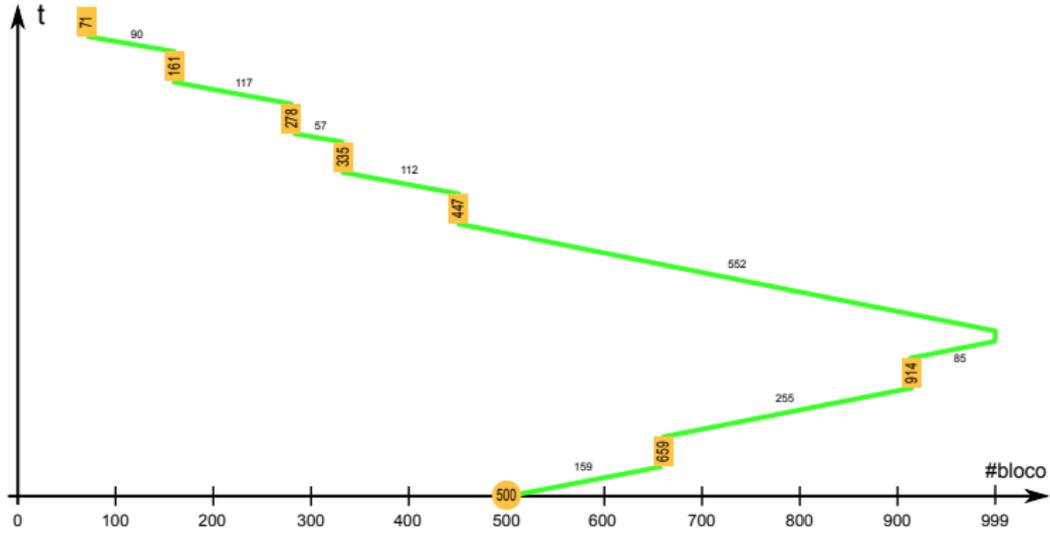
# Escalonamento SSTF



**Risco de inanição** (*starvation*) para blocos distantes



# Escalonamento SCAN

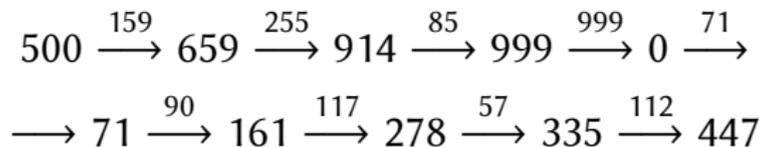


# Escalonamento C-SCAN

Variante “circular” do algoritmo SCAN.

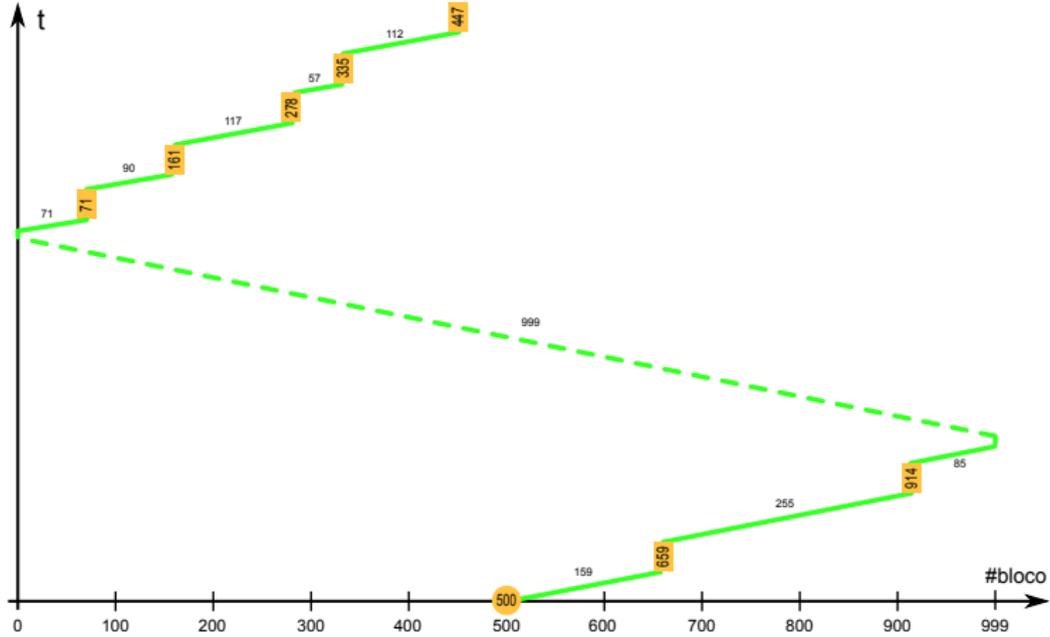
Varre o disco somente em um sentido.

Tempo de espera mais homogêneo aos pedidos pendentes.



Deslocamento da cabeça: 1.945 blocos.

# Escalonamento C-SCAN



# Escalonador LOOK

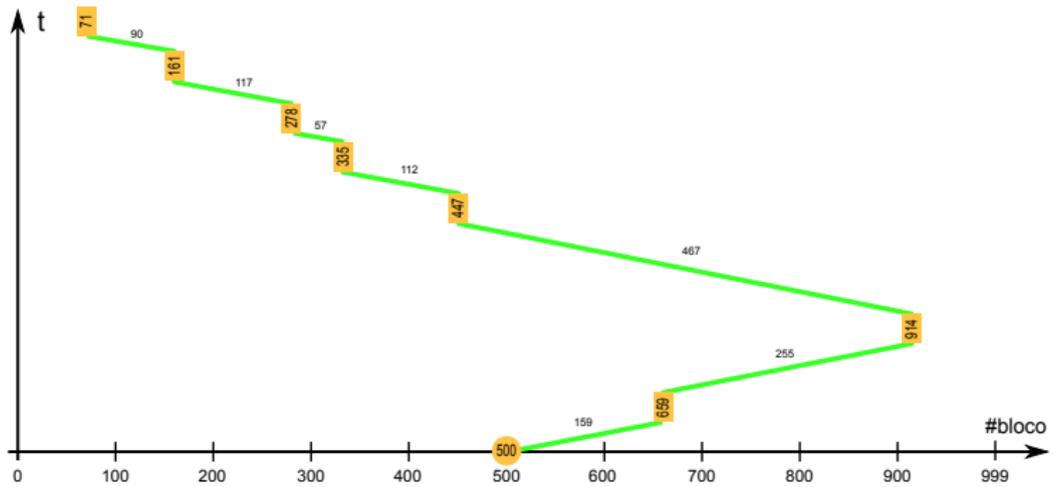
Otimização do algoritmo SCAN

A cabeça não avança até o final do disco

$500 \xrightarrow{159} 659 \xrightarrow{255} 914 \xrightarrow{467} 447 \xrightarrow{112} 335 \xrightarrow{57} 278 \xrightarrow{117} 161 \xrightarrow{90} 71$

Deslocamento da cabeça: 1.257 blocos

# Escalonamento LOOK



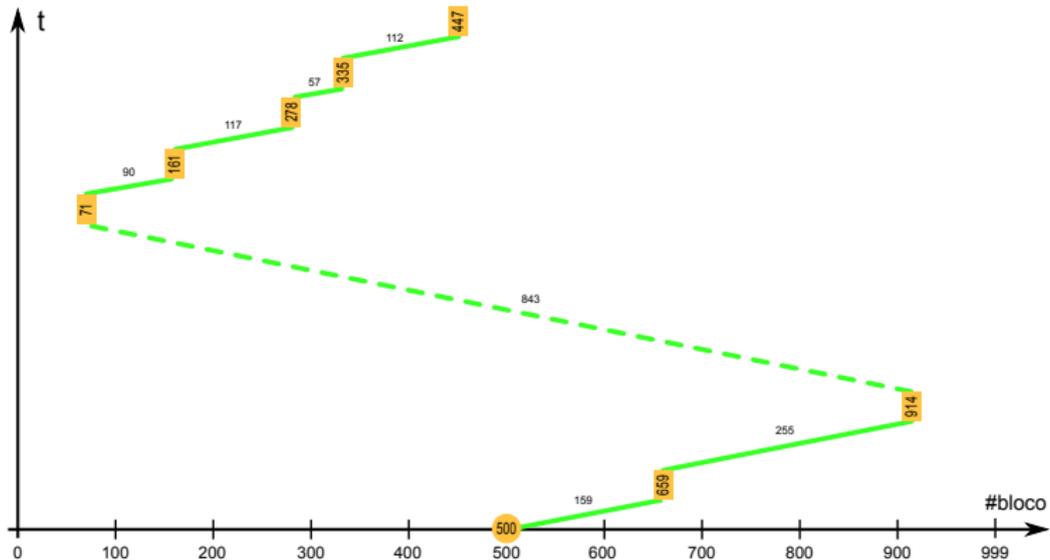
# Escalonador C-LOOK

Otimização do algoritmo C-SCAN

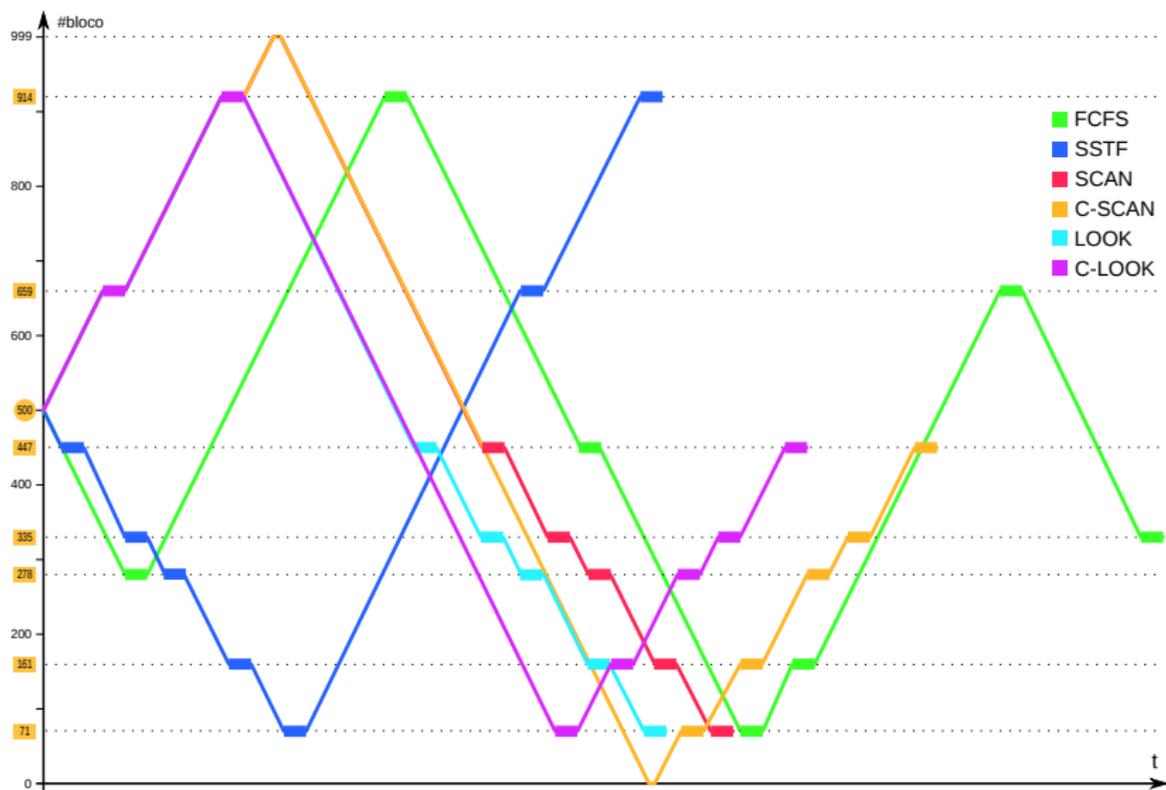
500  $\xrightarrow{159}$  659  $\xrightarrow{255}$  914  $\xrightarrow{843}$  71  $\xrightarrow{90}$  161  $\xrightarrow{117}$  278  $\xrightarrow{57}$  335  $\xrightarrow{112}$  447

Deslocamento da cabeça: 1.644 blocos

# Escalonamento C-LOOK



# Comparativo de escalonadores



# Escalonadores de disco no Linux

## ■ Noop

- Baseado em FCFS
- Agrupa pedidos ao mesmo bloco ou blocos adjacentes
- Usado em SSD e sistemas RAID

## ■ Deadline e Anticipatory

- Associa prazos aos pedidos (500 ms leitura, 5s escrita)
- Baseado no algoritmo C-SCAN, priorizando os prazos
- *Anticipatory*: agrupa leituras do mesmo processo

## ■ CFQ - Completely fair queueing

- Pedidos são distribuídos em várias filas (64 por default)
- Cada fila tem uma fatia de tempo para acessar o disco

Consultar `/sys/block/[device]/queue/scheduler`

# Sistemas RAID

Problemas dos discos rígidos:

- Discos são lentos
- Discos podem falhar, levando à perda de dados

Estratégia RAID: *Redundant Array of Independent Disks*

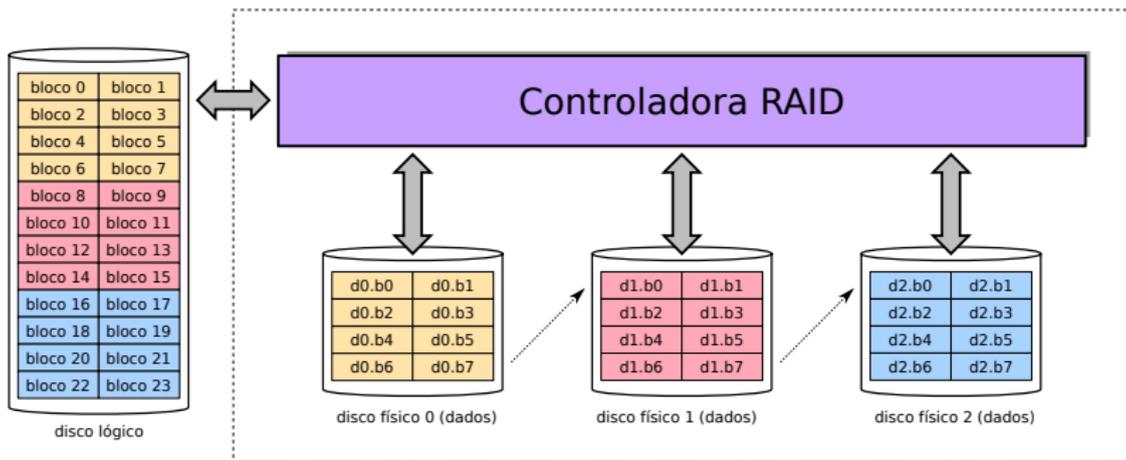
- Criar um **disco lógico** a partir de discos físicos
- **Operações em paralelo** permitem maior desempenho
- **Redundância** (cópias) permitem tolerar falhas
- Implementado em hardware dedicado ou software
- Opera com blocos (abaixo dos arquivos)

# Níveis RAID

- **RAID 0** - soma de discos (linear ou *stripping*)
- **RAID 1** - espelhamento de discos
- RAID 2 - redundância de bits (não usado)
- RAID 3 - redundância de bytes (não usado)
- RAID 4 - redundância de blocos, disco de paridade
- **RAID 5** - redundância de blocos, blocos de paridade distribuídos
- RAID 6 - dois blocos de paridade, para tolerar mais erros
- **RAID 1+0 ou 0+1** - combinações de RAID 0 e 1
- ...

# RAID 0 linear

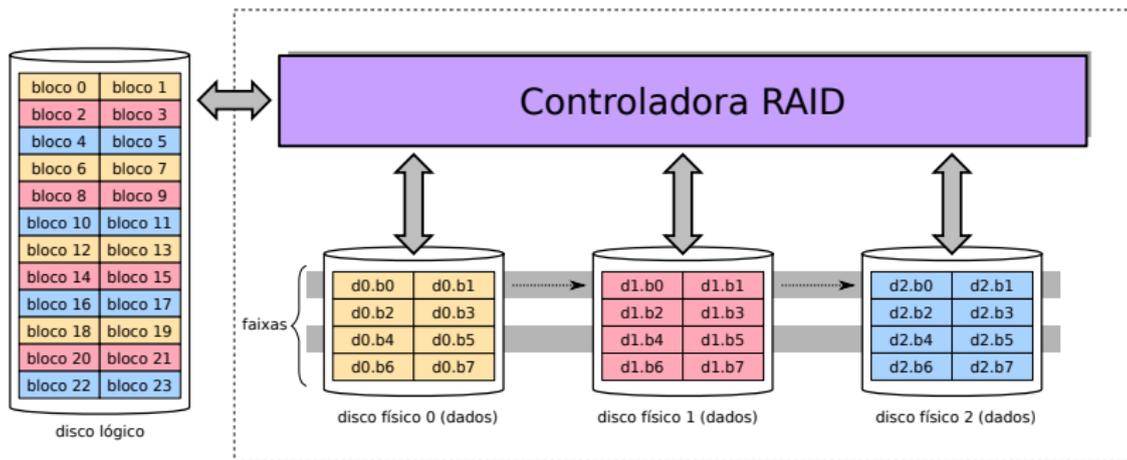
**Estratégia:** concatena discos em sequência



Mais espaço e velocidade, sem redundância.

# RAID 0 *striping*

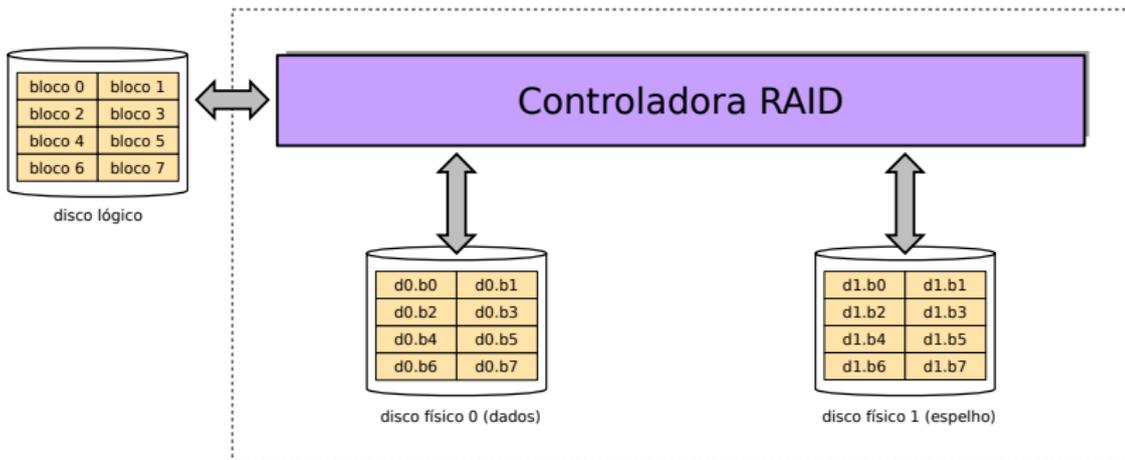
**Estratégia:** concatena discos em faixas de blocos



Desempenho mais equilibrado entre discos.

# RAID 1

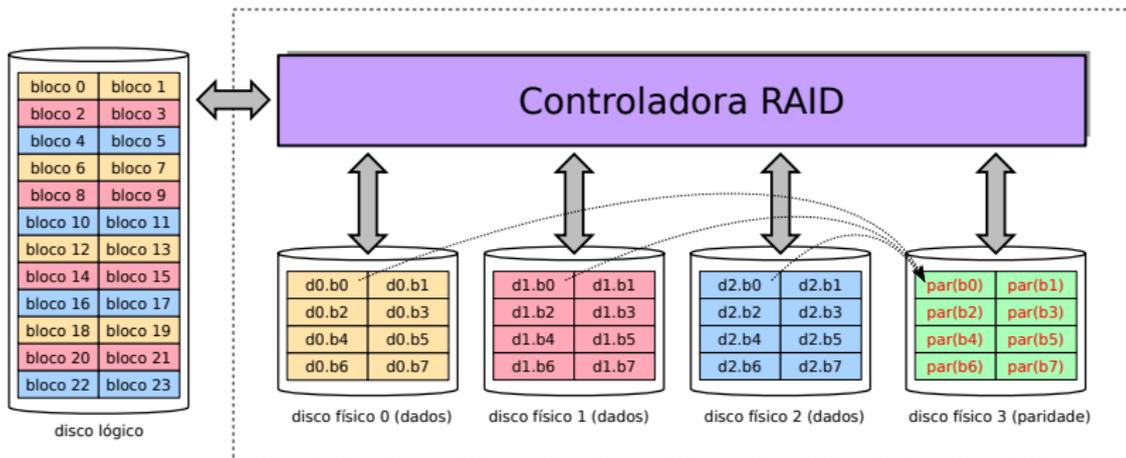
**Estratégia:** espelhamento (cópias do disco)



Boa velocidade, tolera falhas de disco, mas tem alto custo.

# RAID 4

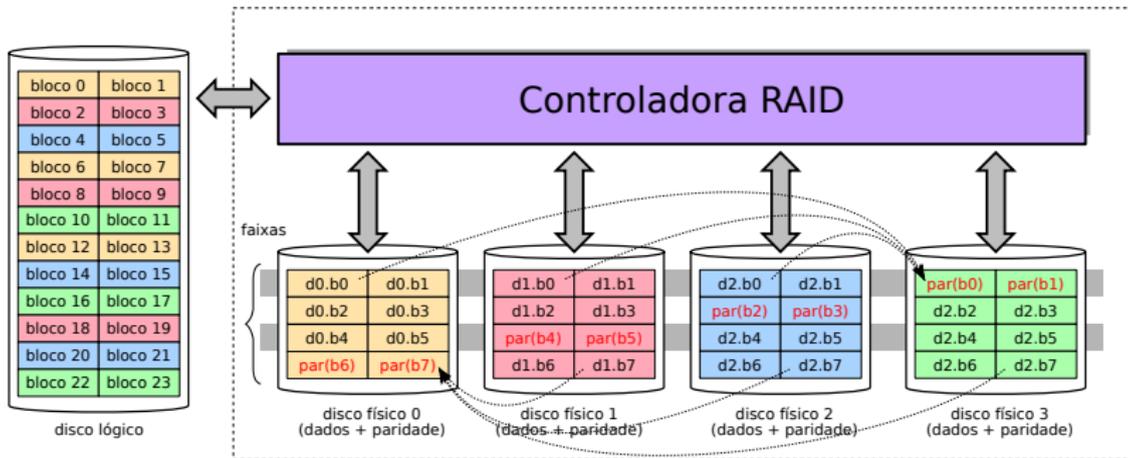
**Estratégia:** disco com blocos de paridade dos demais discos



Não é usado, base conceitual do RAID 5.

# RAID 5

**Estratégia:** blocos de paridade espalhados nos discos



Mais velocidade com tolerância a falhas e baixo custo.

# Comparação de níveis RAID

Considerando arranjos com  $N$  discos de tamanho  $T$ :

Estratégia	Leitura	Escrita	Espaço	Falhas	Discos
<b>RAID 0 linear</b>	até $N$	até $N$	$N \times T$	0	$\geq 2$
<b>RAID 0 strip</b>	até $N$	até $N$	$N \times T$	0	$\geq 2$
<b>RAID 1</b>	até $N$	1	$T$	$N - 1$	$\geq 2$
<b>RAID 4</b>	até $N - 1$	1	$(N - 1) \times T$	1	$\geq 3$
<b>RAID 5</b>	até $N$	1	$(N - 1) \times T$	1	$\geq 3$
<b>RAID 6</b>	até $N$	1	$(N - 2) \times T$	2	$\geq 4$