Sistemas de Armazenamento em Massa

MCTA026-13 - Sistemas Operacionais

Emilio Francesquini
e.francesquini@ufabc.edu.br

2019.Q1

Centro de Matemática, Computação e Cognição
Universidade Federal do ABC

UFABC

Disclaimer

UFABC

- Estes slides foram preparados para o curso de **Sistemas Operacionais na UFABC**.
- Este material pode ser usado livremente desde que sejam mantidos, além deste aviso, os créditos aos autores e instituições.
- Este material foi baseado nas ilustrações e texto preparados por Silberschatz, Galvin e Gagne [SGG] e Tanenbaum e Bos [TB] detentores do copyright.
- Algumas figuras foram obtidas em: `http://pngimg.com`

Introdução

Roteiro

UFABC

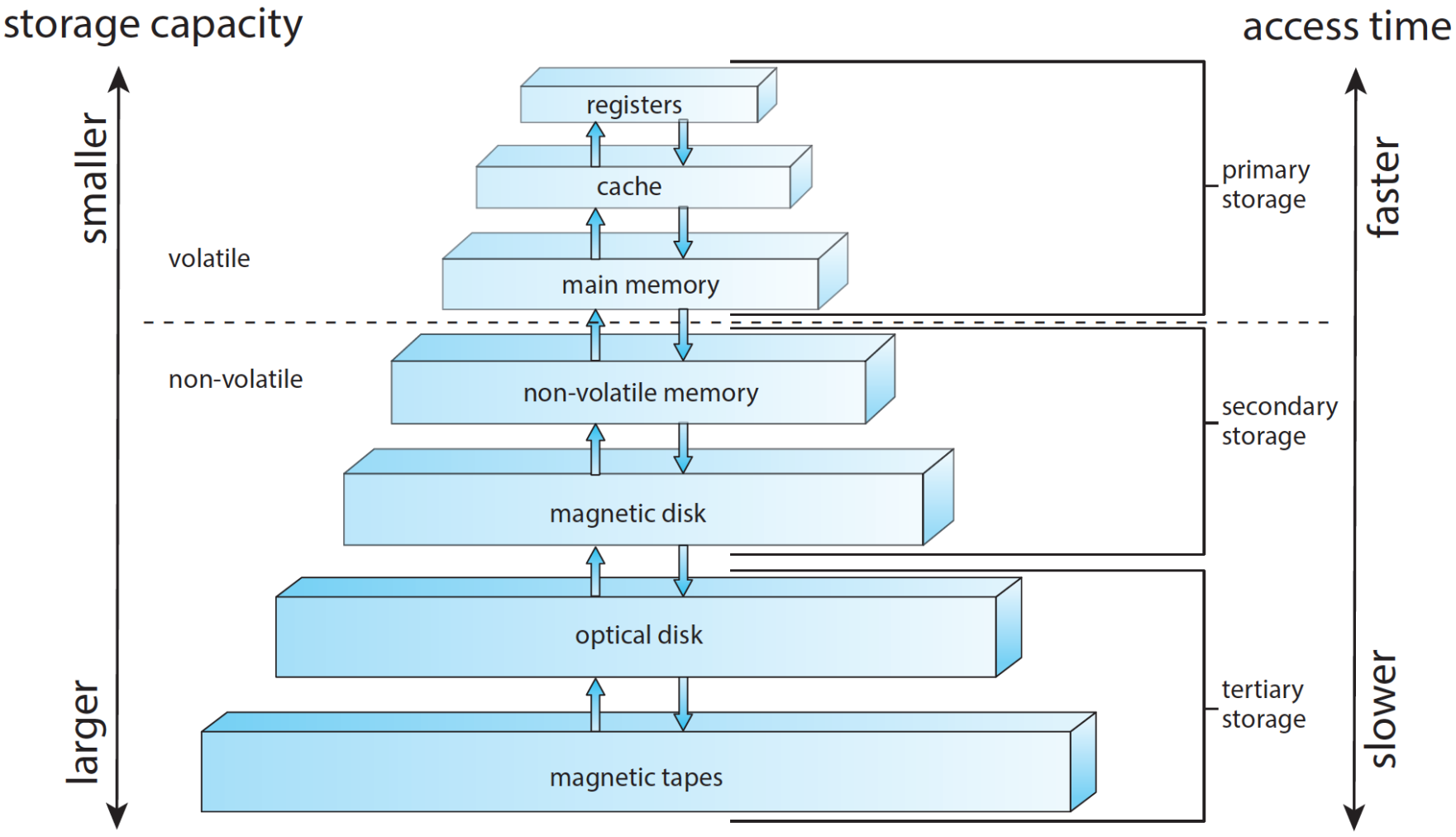
- Visão geral de sistemas de armazenamento em massa
- Conexão de dispositivos de armazenamento
- Escalonamento de E/S para dispositivos de armazenamento secundário
- Gerenciamento de dispositivos de armazenamento
- Gerenciamento de espaço de troca em disco
- Estrutura de RAID

Objetivos

UFABC

- Descrever a estrutura física dos dispositivos de armazenamento secundários e seus efeitos no seu uso
- Explicar as características de desempenho de dispositivos de armazenamento em massa
- Avaliar algoritmos de escalonamento de E/S
- Discutir serviços oferecidos pelo SO relativos ao armazenamento em massa como, por exemplo, RAID
- Descrever a estrutura de NVMs e seus efeitos no uso dos dispositivos

Hierarquia dos dispositivos de armazenamento



Visão geral da estrutura de armazenamento em massa

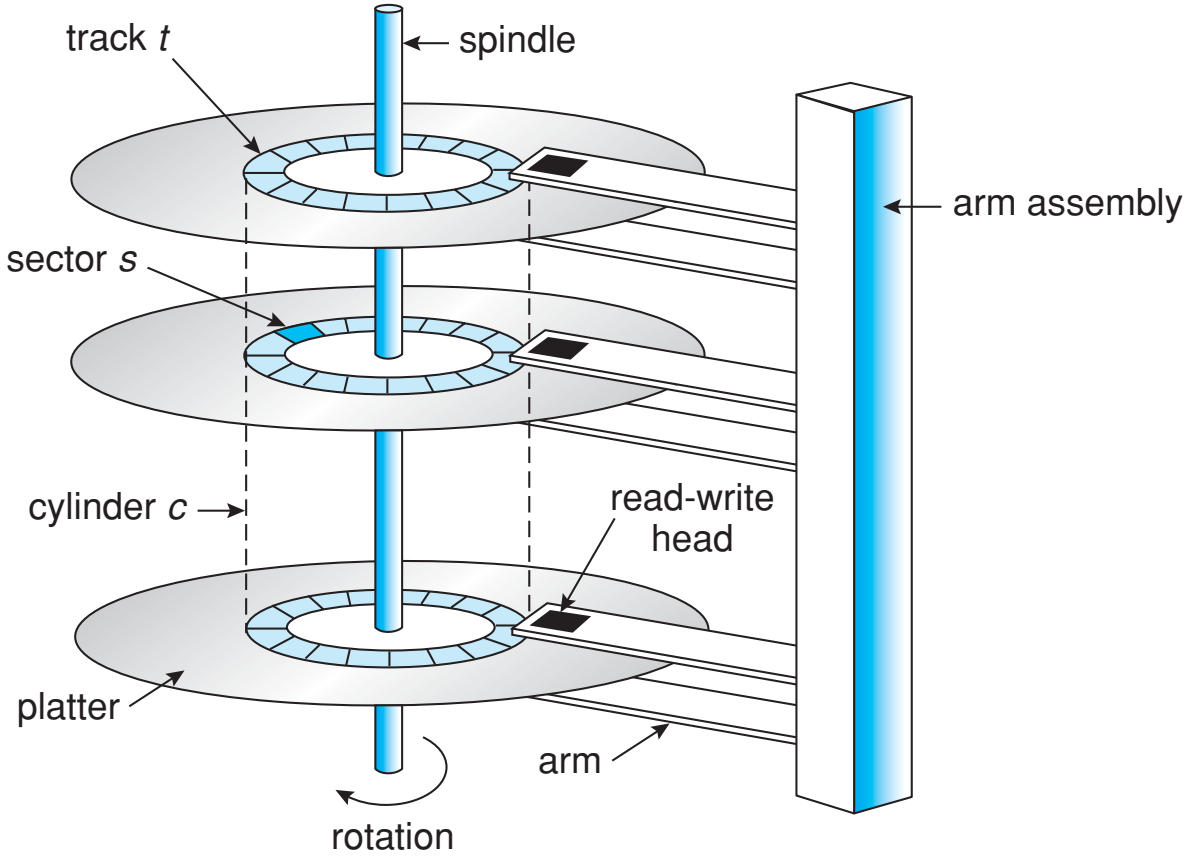
UFABC

- Em sua maioria, o armazenamento em massa de computadores modernos é baseado em
  - Discos rígidos (*hard disk drives* - HDD)
  - Memórias não voláteis (*Non-Volatile* Memories - NVM)
- O tipo mais comum de armazenamento terciário hoje em dia é fita magnética
- Neste aula descreveremos
  - O funcionamento básico destes dispositivos
  - Como os sistemas operacionais traduzem as idiossincrasias físicas destes dispositivos de armazenamento através do mapeamento de endereços

Discos rígidos - Mecanismo de cabeça móvel

UFABC

- Historicamente os discos variam de .85”a 14”
  - Mais comumente têm o tamanho de 3.5”, 2.5”e 1.8”
- Capacidade atual típica de centenas de gigabytes a alguns terabytes por dispositivo



Discos rígidos - HDDs

UFABC

- HDDs funcionam de 60 a 250 rotações por segundo
  - 3600 RPM - 15000 RPM
- A Taxa de transferência (*transfer rate*) é a taxa pela qual os dados fluem entre o HDD e o computador
- O tempo de posicionamento (*positioning time*) ou tempo de acesso aleatório (*random access time*) é o tempo necessário para
  - Levar a cabeça de leitura até o cilindro desejado (tempo de busca - *seek time*) adicionado do tempo necessário para
  - Esperar o setor desejado chegar em uma posição apropriada para leitura, ou seja, abaixo da cabeça de leitura (latência rotacional - *rotational latency*)

Discos rígidos - HDDs

UFABC

- Uma colisão da cabeça de leitura (*head crash*) ocorre quando a cabeça de leitura encosta na superfície do disco
  - Não é muito legal...
- Discos podem ser removíveis
- Outros tipos de mídias com características de funcionamento semelhantes são: CDs, DVDs, Blu-rays, fitas magnéticas

Um HDD de 3.5 polegadas

UFABC

O primeiro disco rígido comercial

UFABC

1956

- O computador IBM RAMDAC vinha com uma unidade de disco IBM Model 350
- Era capaz de armazenar 5 milhões de caracteres (7 bits), ou aproximadamente **4 megabytes**
- Tinha 50 discos de 24”cada
- Tempo de acesso $\leq$ 1 segundo

Desempenho de discos magnéticos

UFABC

- Taxa de transferência teórica - 6 Gbps
- Taxa efetiva de transferência - 1 Gbps
- Tempo de busca - de 3 ms a 12 ms
  - 9 ms é o valor típico para HDs de desktop
  - Tempo médio de busca calculado baseando-se em 1/3 das trilhas
  - Latência baseada na rotação do disco - $1/(\text{RPM}/60) = 60/\text{RPM}$
  - Latência média = $1/2$ latência
- Dados da Wikipédia:

RPM	Latência Média (ms)
4200	7.14
5400	5.56
7200	4.17
10000	3.00
15000	2.00

Desempenho de discos magnéticos

UFABC

- Latência de acesso (*access latency*) = tempo médio de acesso (*average access time*) = tempo de busca + latência média
  - Para discos rápidos: 3 ms + 2 ms = 5 ms
  - Para discos lentos: 9 ms + 5.56 ms = 14.56 ms
- Tempo médio de E/S = tempo médio de acesso + (quantidade de dados a serem transferidos / taxa de transferência) + sobrecarga (*overhead*) do controlador

Analizando o desempenho - Exemplo

UFABC

- Considere
  - Uma transferência de um bloco de 4 KB
  - Disco a 7200 RPM
  - Tempo de busca médio de 5 ms
  - Taxa de transferência de 1 Gbps
  - Overhead de 0.1 ms do controlador

- 5 ms + 4.17 ms + 0.1 ms + tempo de transferência
- **Tempo de transferência** = 4 KB / 1 Gbps * 8 Gb / GB * 1 GB / 1024^2 KB = 32 / 1024^2 = 0.031 ms
- **Tempo médio de E/S para um bloco de 4 KB** = 9.27 ms + 0.31 ms = 9.301 ms

Estrutura do disco

UFABC

- HDDs são endereçados como um grande vetor unidimensional de blocos lógicos
  - Um bloco lógico é a menor unidade de dados que pode ser transferida
- O vetor unidimensional de blocos lógicos é mapeado sequencialmente em setores do disco
  - O setor 0 é o primeiro setor da primeira trilha do cilindro mais externo
  - O mapeamento continua, em ordem, por esta trilha, seguido pelos demais cilindros dos mais externos para os mais internos

Estrutura do disco

UFABC

- A conversão de endereços lógicos para físicos é (em sua essência) fácil
  - O difícil é levar em conta os setores danificados (*bad sectors*)
  - Há também um número não constante de setores por trilha devido à constante velocidade angular
- Tipicamente os setores são de 512B a 4KB e são a menor unidade de E/S que o HDD é capaz de efetuar

UFABC

https://www.youtube.com/watch?v=kSlkgjNb7YM

Memórias não voláteis (NVM)

UFABC

- NVM (*non-volatile memory*) não possuem componentes móveis
- São tipicamente fabricadas usando tecnologia FLASH, mas também vem em outros formatos como DRAM com bateria, PCM, RERAM, ...
- O tipo mais comum é chamado de SSD (*solid-state disk*) que além dos chips de memória inclui um circuito controlador

- Também pode ser em formatos mais portáteis como um pen drive[1]

[1] Em inglês tipicamente se fala em *thumb drive* ou *USB drive*

Memórias não voláteis (NVM)

UFABC

- Apesar das diferentes tecnologias, em geral todas são tratada da mesma maneira
- Mais confiável que HDs (por não conterem partes móveis), podem ser mais rápidas (já que não tem tempo de busca ou uma alta latência de acesso) e consomem menos energia
- Mais caras por MB, menor capacidade e podem ter um tempo de vida inferior (escritas podem desgastar cada uma das células de memória)
- As velocidades mais rápidas exigiram novos métodos de acesso
  - Por exemplo: NVMe que nada mais é que uma conexão direta ao barramento PCIe
- São usadas como substitutas aos discos ou até mesmo para caching

NVM - NAND FLASH - Detalhes

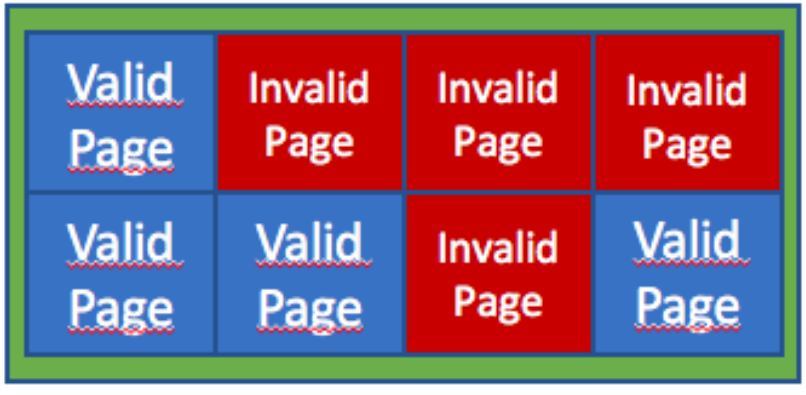
UFABC

- São lidas em páginas
  - O tamanho da página depende do dispositivo
- Não sobrescrevem os dados, é preciso apagá-los primeiro
- O processo de apagamento é feito em blocos que são compostos de várias páginas
- Apagar leva muito mais tempo do que escrever ou ler
- NAND se desgasta a cada apagamento
  - Vida útil de aproximadamente 100.000 ciclos

NVM - NAND FLASH - Controlador do dispositivo

UFABC

- Diversos algoritmos são utilizados
  - São tipicamente implementados no próprio dispositivo
  - Logo, os SOs simplesmente ignoram os detalhes e apenas leem e escrevem blocos deixando a tarefa na mão do próprio dispositivo
  - Contudo, essas características podem ter um impacto no desempenho então é importante conhecê-las
- Um bloco NAND com páginas válidas e inválidas



UFABC

- Como NAND não pode ser sobrescrito
  - Há normalmente páginas contendo dados inválidos
  - Para manter o controle de quais blocos lógicos contêm os dados válidos o controlador mantém a *Flash Translation Layer* (FTL)
  - A FTL mantém um mapeamento de qual página física contem o bloco lógico válido
  - A FTL também é usada para guardar o estado físico do bloco (por exemplo, se o bloco contém apenas páginas inválidas e portanto pode ser apagado)
  - Cada bloco lógico pode ter muitas versões (cada uma armazenada em uma página física diferente, porém com apenas uma delas válida)

NVM - NAND FLASH - Algoritmos completos

UFABC

- Todas as páginas podem estar escritas e várias contém valores inválidos
- Fisicamente todas as páginas podem estar ocupadas, mas logicamente há espaço livre
  - Coleta de lixo (*garbage collection*) copia dados de páginas válidas de diversos blocos para outras localizações para liberar blocos a serem apagados
    - Se o dispositivo estiver cheio, pode ser impossível fazê-lo
  - Over-provisioning 20% do dispositivo reservado para as escritas do GC
    - Conforme os blocos que contém apenas páginas inválidas são liberados pelo GC eles são colocados no pool de overprovision ou devolvidas ao pool de blocos livres

UFABC

- Overprovision ajuda no controle de desgaste (*wear leveling*)
  - É desejado que as células do dispositivo se desgastem por igual para aumentar a vida útil do dispositivo
  - Há um controle do número de escritas por bloco, GC e overprovision levam isto em conta para escolher os blocos ”menos escritos”
- Dados protegidos na NVM por códigos de correção de erros (ECC)
  - Se um bloco começar a apresentar muitos erros ele é marcado para evitar futuros usos
  - Alguns erros só podem ser recuperados via RAID

Fita magnética

UFABC

- Nos primórdios da computação era usada como dispositivo de armazenamento secundário
  - Evoluiu de bobinas abertas para a versão moderna contida em cartuchos
- Apresenta boa durabilidade e é capaz de armazenar uma grande quantidade de dados
- Tempo de acesso, por outro lado, é alto
- Tempo de acesso aleatório chega a ser ~1000× mais lento do que discos
- Usada normalmente para backups, arquivos mortos ou como um meio de transferência entre sistemas

Fita magnética

UFABC

- Mantida em bobinas e é bobinada ou rebobinada para alcançar a posição desejada e poder ser lida pela cabeça de leitura
- Quando a posição desejada está sob a cabeça de leitura as taxas de transferência são comparáveis às de um disco
  - > 140 MBps
- Capacidade de armazenamento típica: de centenas de GB a vários TB
- Tecnologias mais comuns: LTO-{5,6}, SDLT, 4, 8, e 19mm, com larguras de 1/4 e 1/2 polegada

Métodos de conexão de dispositivos de armazenamento secundário

UFABC

- Conectados ao host utilizando portas de E/S e barramento
- Diversas tecnologias de barramento: advanced technology attachment (ATA), serial ATA (SATA), eSATA, universal serial bus (USB), fibre channel (FC), serial attached SCSI (SAS)
- A transferência de dados é feita por processadores eletrônicos dedicados: controladores
  - O controlador do host (*host controller*) é do computador, o controlador do dispositivo (*device controller*) fica no dispositivo
  - Conversam entre si através de portas de E/S mapeadas em memória
- Controladores de dispositivo têm caches embutidas
  - A transferência se dá da mídia para a cache e da cache para o host através do barramento (e vice-versa)

Mapeamento de endereços

UFABC

- Dispositivos de armazenamento são endereçados como um grande vetor unidimensional de blocos lógicos
  - Um bloco lógico é a menor unidade de dados que pode ser transferida
  - Cada bloco lógico é mapeado a um setor físico ou a uma página no dispositivo
    - Por exemplo, o bloco lógico 0 pode ser mapeado no setor 0 do cilindro 0 do disco 0 do HDD
  - É mais conveniente utilizar um endereço lógico $(0 \ldots n)$ do que endereços do tipo <setor,cilindro,cabeça> ou <chip,bloco,página>
  - Tem também a vantagem de que setores ou blocos defeituosos podem ser retirados de uso apenas por uma modificação no mapeamento
  - O mapeamento também simplifica o acesso aos dispositivos onde o número de setores por trilha não é fixo

Mapeamento de endereços

UFABC

- Alguns dispositivos como CDs e DVDs apresentam velocidade linear constante (*constant linear velocity* - CLV)
  - A densidade de bits por trilha é uniforme
  - Quanto mais distante uma trilha está do centro do disco $\rightarrow$ maior o comprimento $\rightarrow$ mais setores
  - O drive aumenta a velocidade angular conforme a cabeça se desloca para manter a mesma taxa de dados para a cabeça de leitura/escrita
- Outros dispositivos, como HDDs, apresentam uma velocidade angular constante (*constant angular velocity* - CAV)
  - A densidade dos bits diminui dependendo da distância do centro de rotação
  - Blu-rays são híbridos: dependendo da mídia podem assumir um comportamento CLV ou CAV

Escalonamento de operações no disco

UFABC

- O sistema operacional é o responsável por utilizar o hardware de maneira eficiente
  - No caso dos dispositivos de disco isso significa acesso rápido com alta banda
- Visam, portanto minimizar o tempo de busca
- Tempo de busca $\approx$ distância de busca (*seek distance*)
- A largura de banda do disco (*disk bandwidth*) é o número total de bytes transferidos dividido pelo tempo total entre a primeira requisição pela operação de E/S e a finalização da última transferência

Escalonamento de operações no disco

UFABC

- Há muitas fontes de requisições às operações de E/S
  - SO
  - Processos do sistema
  - Processos do usuário
- Requisições de E/S incluem modo de entrada ou saída, endereço do disco, endereço de memória, número de setores a serem transferidos
- O SO mantém uma fila de requisições por disco ou dispositivo
- Se o disco está ocioso ele pode trabalhar na requisição imediatamente, caso contrário ela precisa ser enfileirada
  - Os algoritmos de otimização que discutimos apenas fazem sentido na presença de uma fila

Escalonamento de operações no disco

UFABC

- Note que controladores de disco possuem pequenos buffers e que eles próprios podem gerenciar a sua própria fila de requisições de E/S
  - Tamanho da fila varia de dispositivo para dispositivo
- Há diversos algoritmos para escalonar as operações no disco
- A análise que apresentamos é válida para um ou vários discos por HDD
- Ilustraremos o funcionamento do algoritmo para uma fila de requisição (0-199)

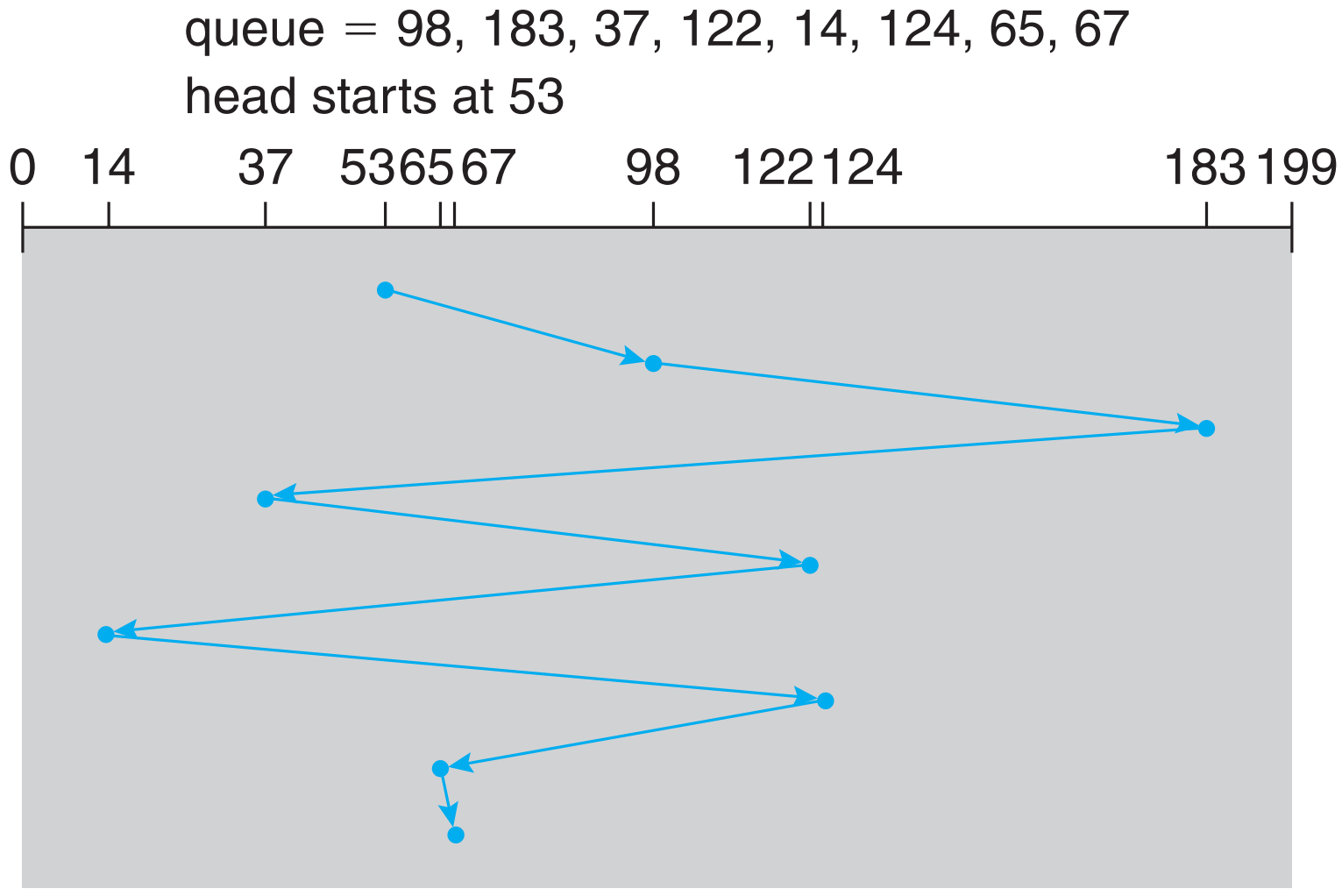
```
98, 183, 37, 122, 14, 124, 65 67
```

Ponteiro da cabeça: `53`

FCFS - First-Come, First-Served

UFABC

A ilustração mostra uma movimentação total de 640 cilindros

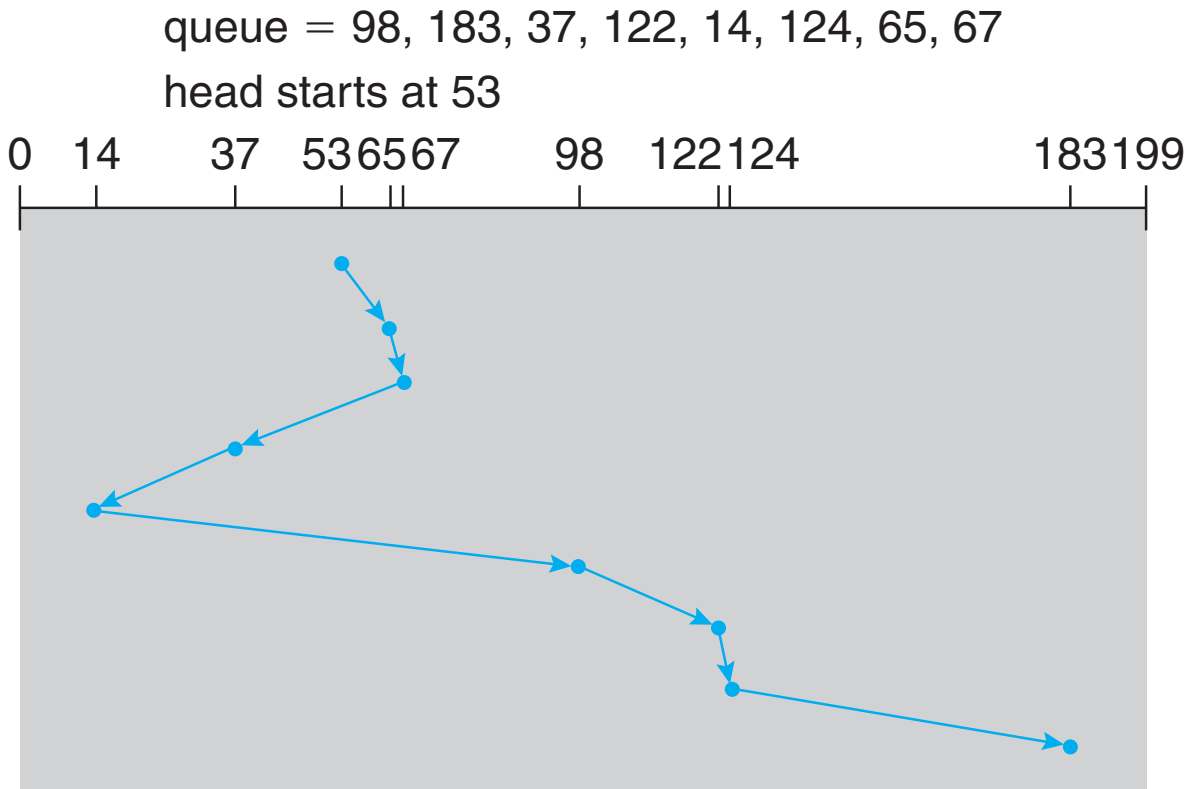


SSTF - Shortest Seek Time First

UFABC

- Escolhe a requisição com o menor tempo de busca a partir da posição atual da cabeça
- SSTF é uma forma de escalonamento SJF: pode causar inanição (*starvation*) em algumas requisições

A ilustração mostra uma movimentação total de 236 cilindros



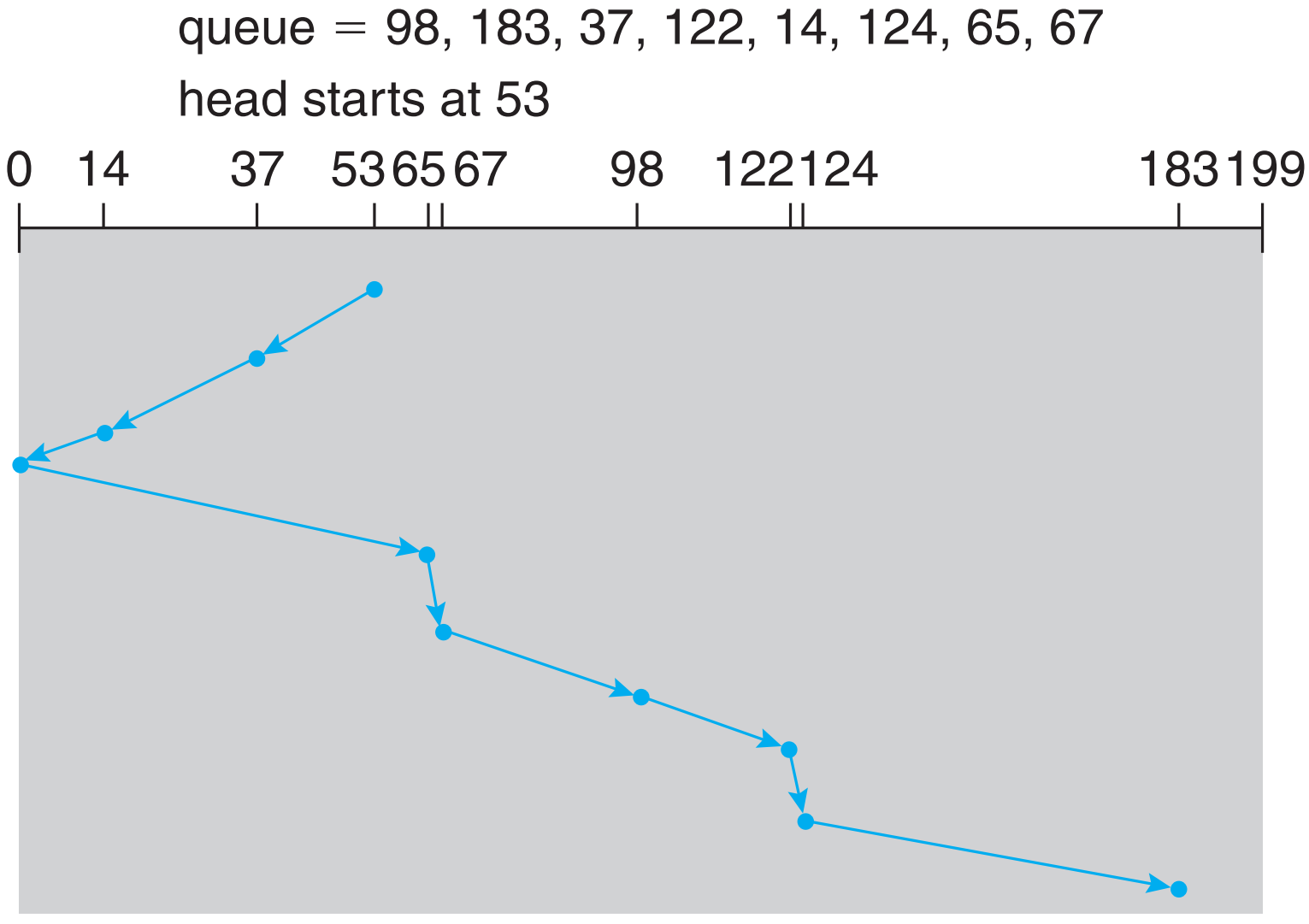
O algoritmo do elevador - SCAN

UFABC

- Também conhecido como algoritmo SCAN
- O braço de leitura inicia em um extremo do disco e se move em direção ao outro extremo. As requisições são atendidas até que a cabeça alcance a extremidade do disco, neste ponto a direção do braço é invertida e o atendimento às requisições prossegue
- Note que este algoritmo **evita inanição**
- Contudo, se as requisições forem distribuídas de maneira uniforme a maior parte das requisições pendentes está no outro extremo do disco (Por que?)
  - Logo o tempo na fila dessas requisições pode ser maior que as demais

O algoritmo do elevador - SCAN

A ilustração mostra uma movimentação total de 208 cilindros

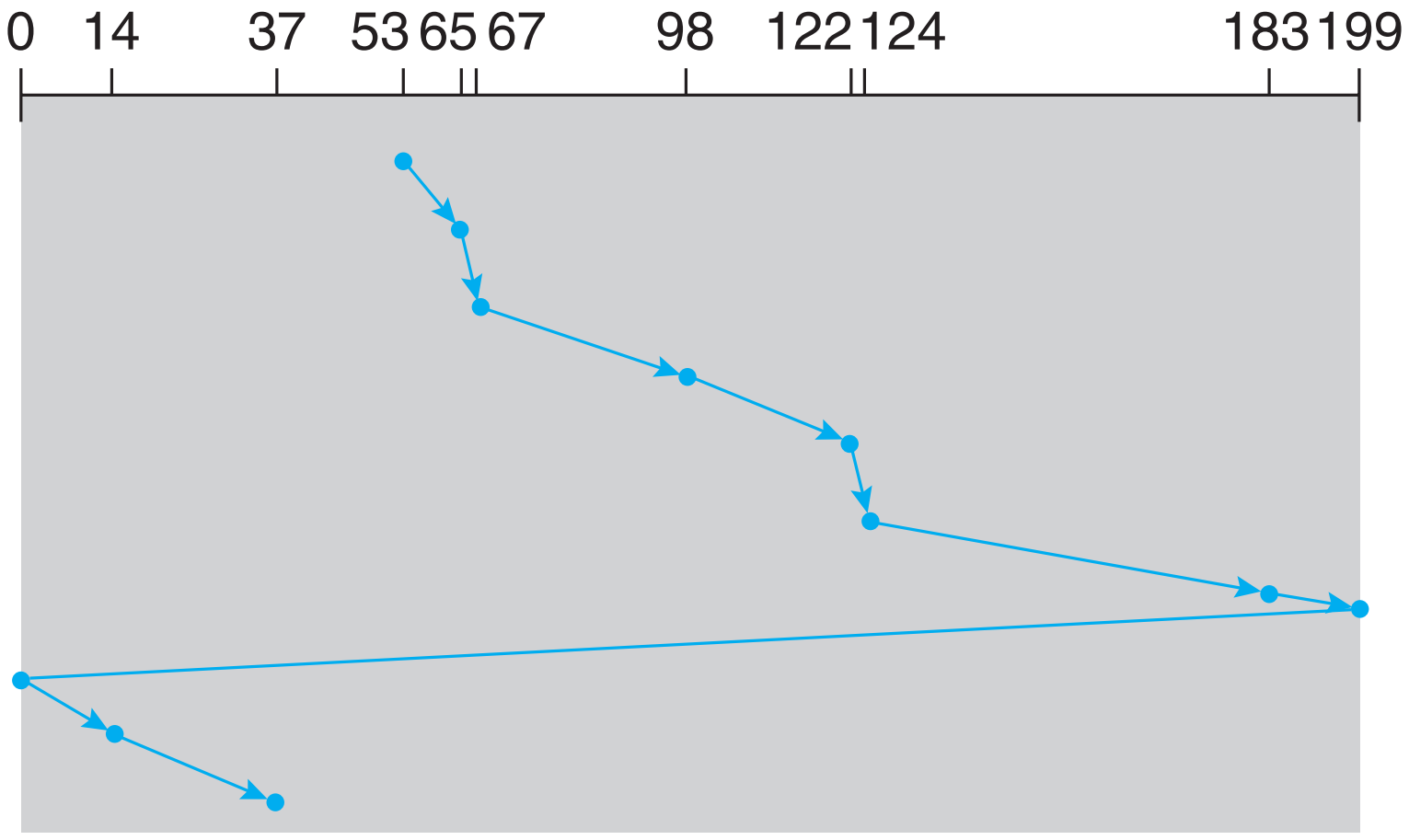


UFABC

- Criado para prover um tempo de espera mais uniforme entre as requisições
- A cabeça se move até uma extremidade até a outra atendendo às requisições conforme se desloca
  - Quando atinge uma extremidade, ela retorna imediatamente a extremidade inicial sem atender a nenhuma requisição no caminho de volta
- Trata os cilindros como uma lista circular que se conecta do cilindro final ao inicial

queue = 98, 183, 37, 122, 14, 124, 65, 67

head starts at 53



LOOK e C-LOOK

UFABC

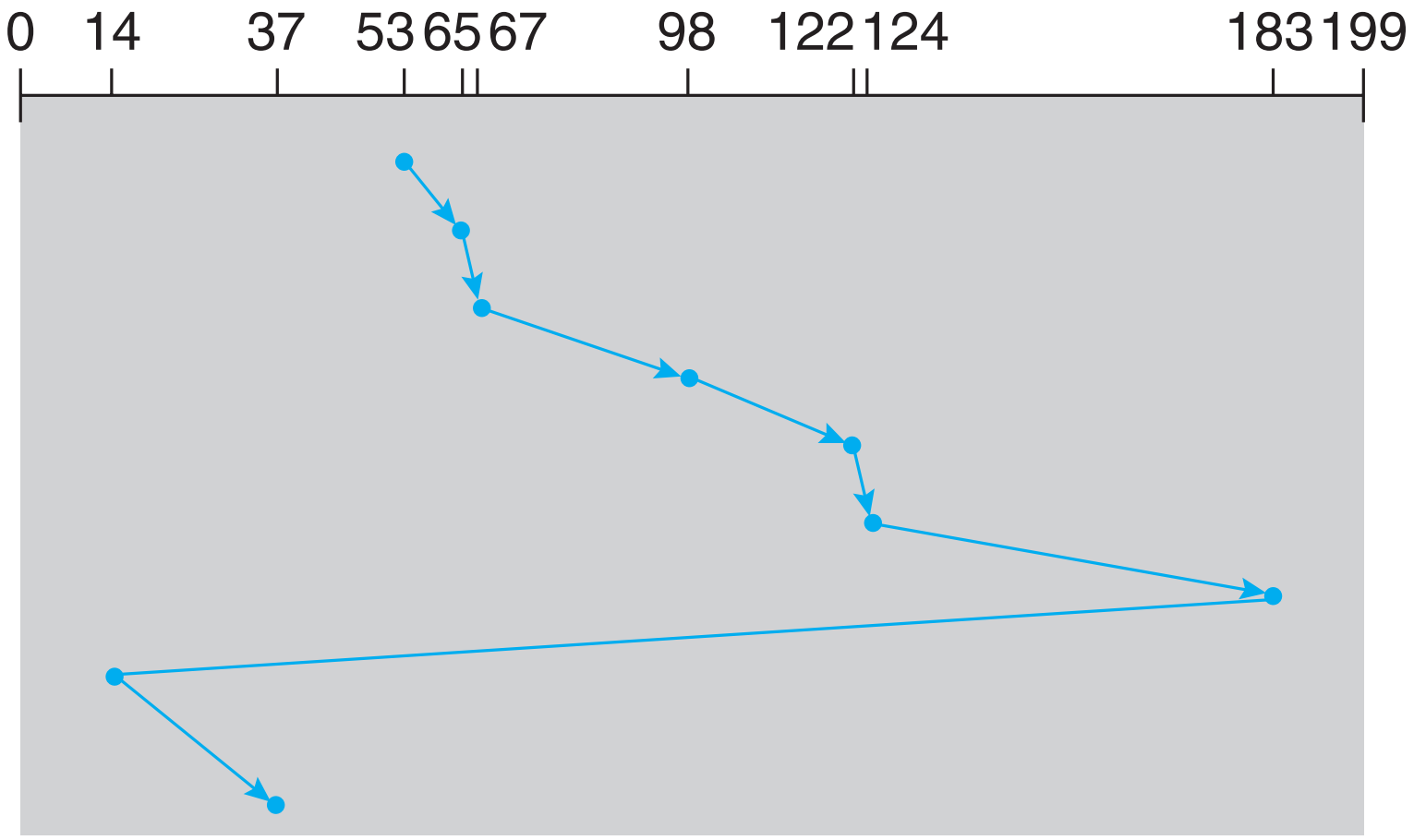
- LOOK é uma versão do SCAN onde o braço só vai até a última requisição **em cada direção** e então inverte a direção (não é necessário chegar à extremidade do disco)
- C-LOOK é uma versão do C-SCAN onde o braço só vai até a última requisição **em uma direção** e então inverte a direção (não é necessário chegar à extremidade do disco)

C-LOOK

UFABC

queue = 98, 183, 37, 122, 14, 124, 65, 67

head starts at 53



Escolhendo um algoritmo de escalonamento

UFABC

- SSTF é comum e tem a vantagem de ser intuitivo
- SCAN e C-SCAN tem um desempenho melhor em sistemas que tem acessos pesados ao disco
  - Têm menos inanição
- O desempenho vai depender do número e dos tipos de requisições
- As requisições para o disco podem sofrer influência do modelo de alocação de arquivos
  - E de seus metadados

Escolhendo um algoritmo de escalonamento

UFABC

- O algoritmo de escalonamento deve ser escrito como um módulo do sistema operacional de modo a possibilitar a sua troca caso necessário
- Tanto SSTF ou LOOK são algoritmos que funcionam razoavelmente bem na prática
- E a latência rotacional?
  - É muito difícil para um SO levar isso em conta
- Que tipo de influência o escalonador do próprio dispositivo tem nos resultados da política de enfileiramento feita pelo SO?

Network-attached storage (NAS)

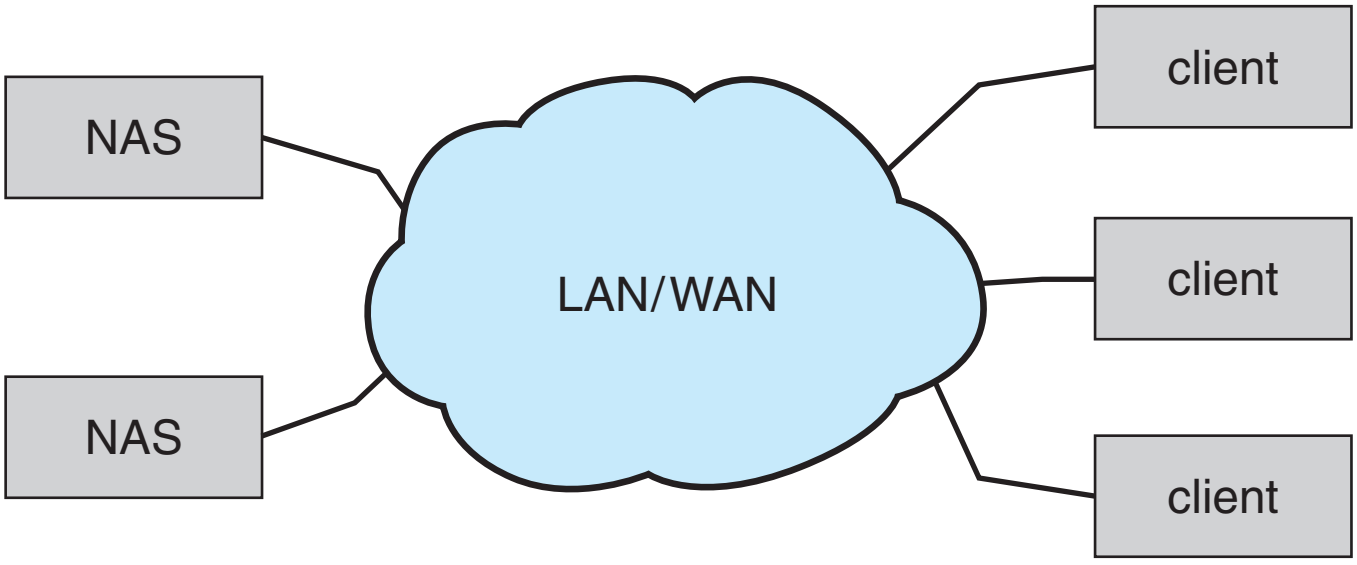
UFABC

- Um NAS é um dispositivo de armazenamento disponibilizado pela rede e não pelo barramento
  - SOs em geral permitem ”montar”sistemas de arquivos remotos (NFS, CIFS, SMB/Samba, ...)
  - Tem sistemas de controle de dados, interfaces de rede, ...
- São implementados via chamadas de procedimentos remotos (*remote procedure calls* - RPC) entre o NAS e o cliente, tipicamente utilizando TCP/UDP numa rede IP

Network-attached storage (NAS)

UFABC

- O protocolo iSCSI é uma adaptação do protocolo SCSI (para barramentos convencionais) para ambientes de rede IP
- Permite a montagem remota de dispositivos (blocos)



Armazenamento em nuvem

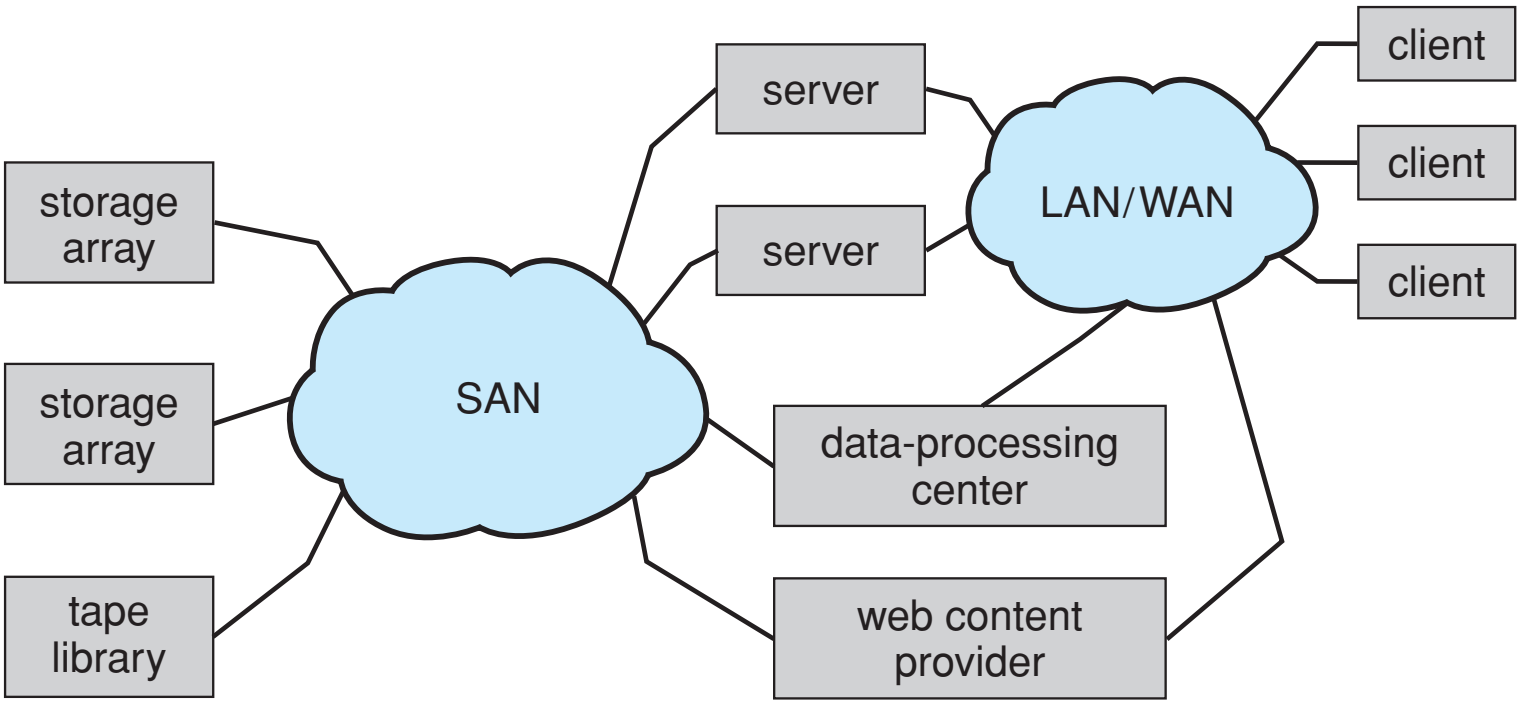
UFABC

- Parecido com NAS
- Provê acesso através da rede, contudo em uma WAN em lugar de uma LAN
- Normalmente não é possuída pelo usuário (ou empresa) mas sim fornecido por um terceiro por uma taxa que varia por tempo, capacidade, E/S feitas, ...
- É tipicamente bem mais lento que NAS e muito mais suscetível a falhas de conexão
  - Ainda assim é possível usar CIFS, NFS, iSCSI porém não é muito comum
- Frequentemente tem suas próprias APIs e aplicações
  - Dropbox, Microsoft OneDrive, Apple iCloud, etc

Storage Area Network

UFABC

- Comum em ambientes com ampla necessidade de armazenamento
- Múltiplos hosts conectados a múltiplos *storage arrays*



Confiabilidade e Redundância

UFABC

- Reliability e Redundancy
- Tempo médio para falha (*Mean time to failure* - MTTF) - O tempo que um disco leva em média para apresentar uma falha
- Tempo médio para reparação (*Mean time to repair* - MTTR) - O tempo que se leva em média para substituir um disco e restaurar os dados
- Espelhamento (*Mirroring*) - Manter uma cópia completa de um disco em outro
  - Considere um disco com MTTF de 100.000 horas e MTTR de 10 horas
  - Tempo médio para a perda de dados é = 100.000^2 / (2 * 10) = 500 * 10^6 horas, ou 57.000 anos supondo-se que as falhas dos discos sejam eventos aleatórios independentes

A BlackBlaze divulga anualmente os MTTFs de discos de diversas marcas e modelos. Confira: https://www.backblaze.com/blog/2018-hard-drive-failure-rates/

Melhorando a confiabilidade

UFABC

- Várias melhorias nas técnicas de uso de discos envolvem o uso de vários discos de maneira conjunta
- O espelhamento completo pode ser matar uma mosca com um tiro de canhão

RAID - Redundant Array of Inexpensive Disks

UFABC

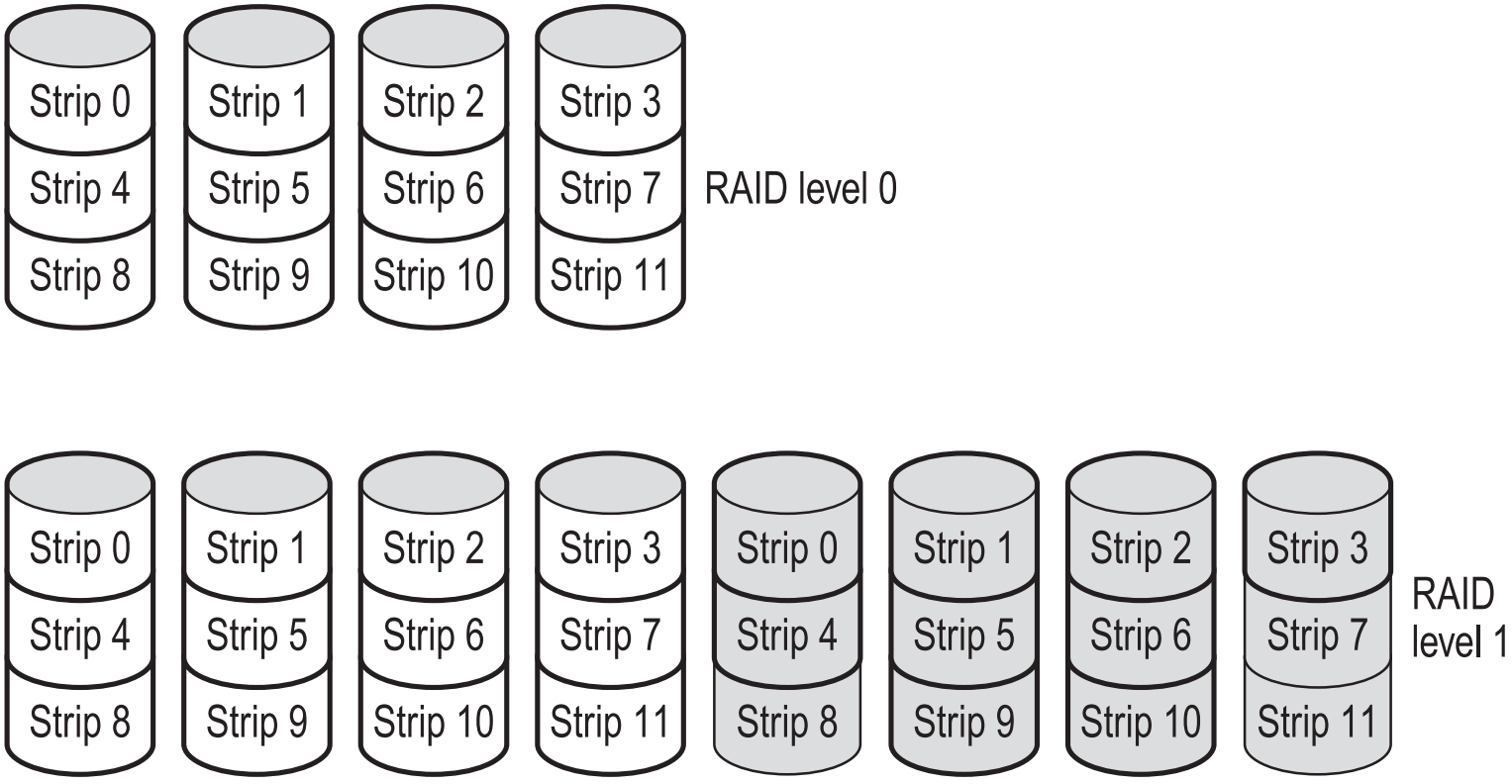
- O uso vários discos **pioram** o MTTF
  - 100 discos com MTTF de 100.000 horas
  - 100.000/100 = 1.000 horas ou 41.66 dias
- A solução envolve utilizar redundância nos 100 discos
- Disk striping - Dividir os bits (ou blocos) entre vários discos
  - bit-level striping - Os bits de um byte são espalhados entre os vários discos
  - block-level striping - Os blocos de um arquivo são espalhados entre vários discos
- Por exemplo, se desejamos fazer um RAID de 8 discos podemos escrever o *i*-ésimo bit de um byte no *i*-ésimo disco. O conjunto dos 8 discos pode então ser tratado como um único disco, com setores com 8x o tamanho padrão e, mais importante, com 8x o desempenho

RAID - Redundant Array of Inexpensive Disks

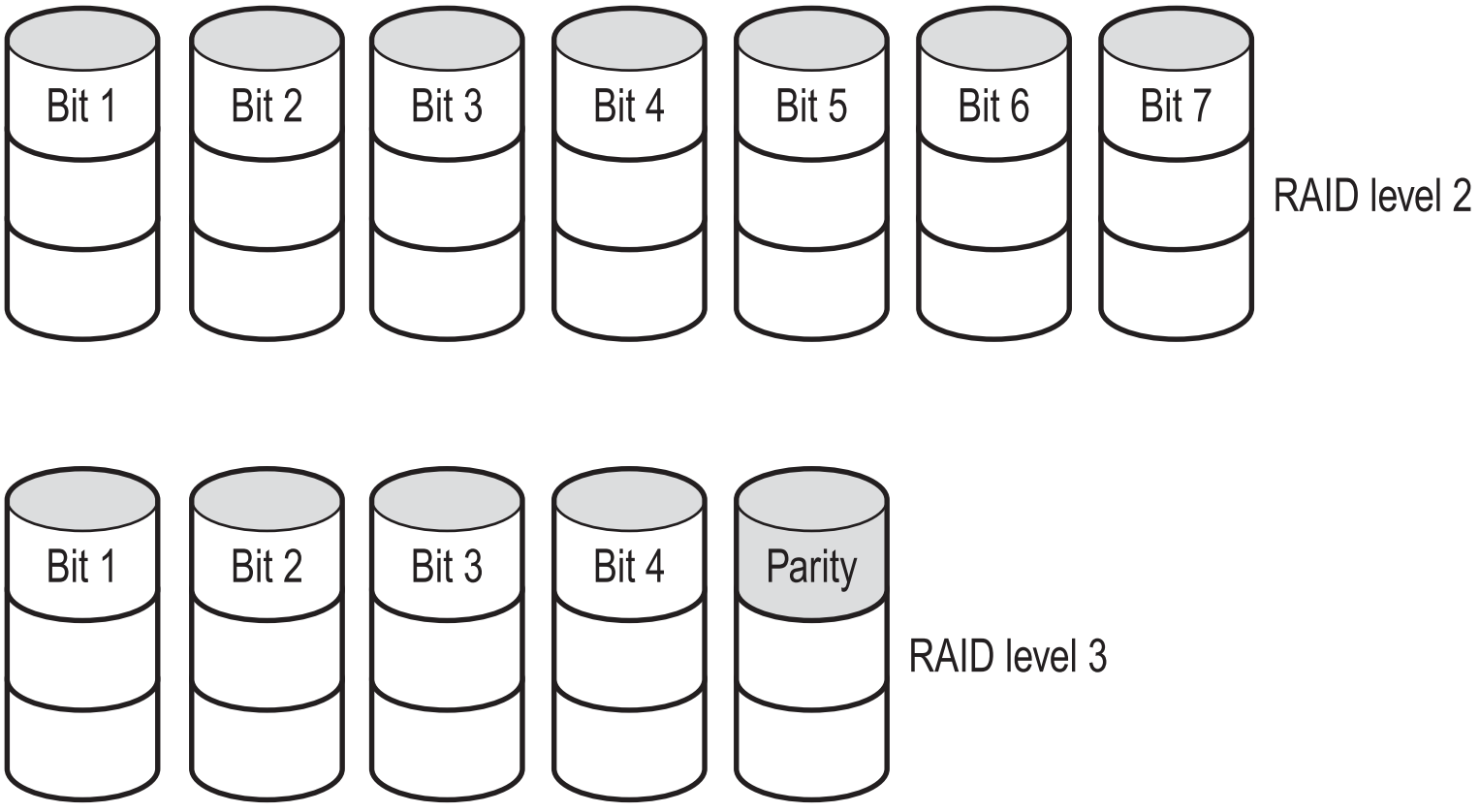
UFABC

- Estruturas de RAID aumentam o desemepenho e a confiabilidade utilizando redundância de dados
- O RAID como um todo ainda pode falhar
  - Replicar os dados entre RAIDs é comum
    - Escritas duplicadas entre sites separados geograficamente
    - A redundância provê a possibilidade de recuperação de desastres
    - Pode ser feita de maneira síncrona ou assíncrona
  - Frequentemente uma pequena quantidade de discos reservas quentes (*hot spare disks*) são mantidas conectados, porém não alocados, para substituir automaticamente um disco que falhou (e receber a sua cópia dos dados)

RAID-0 e RAID-1

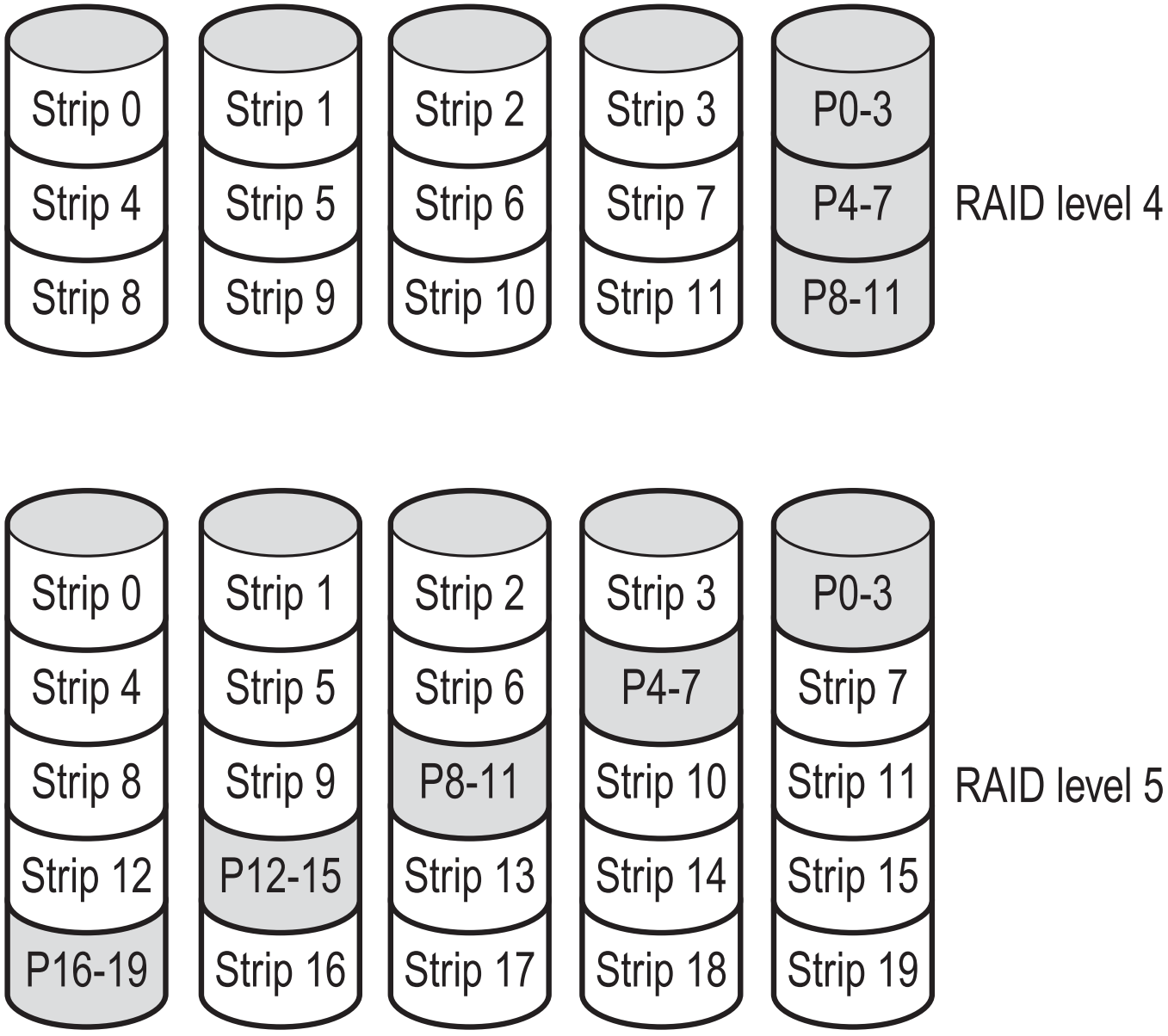


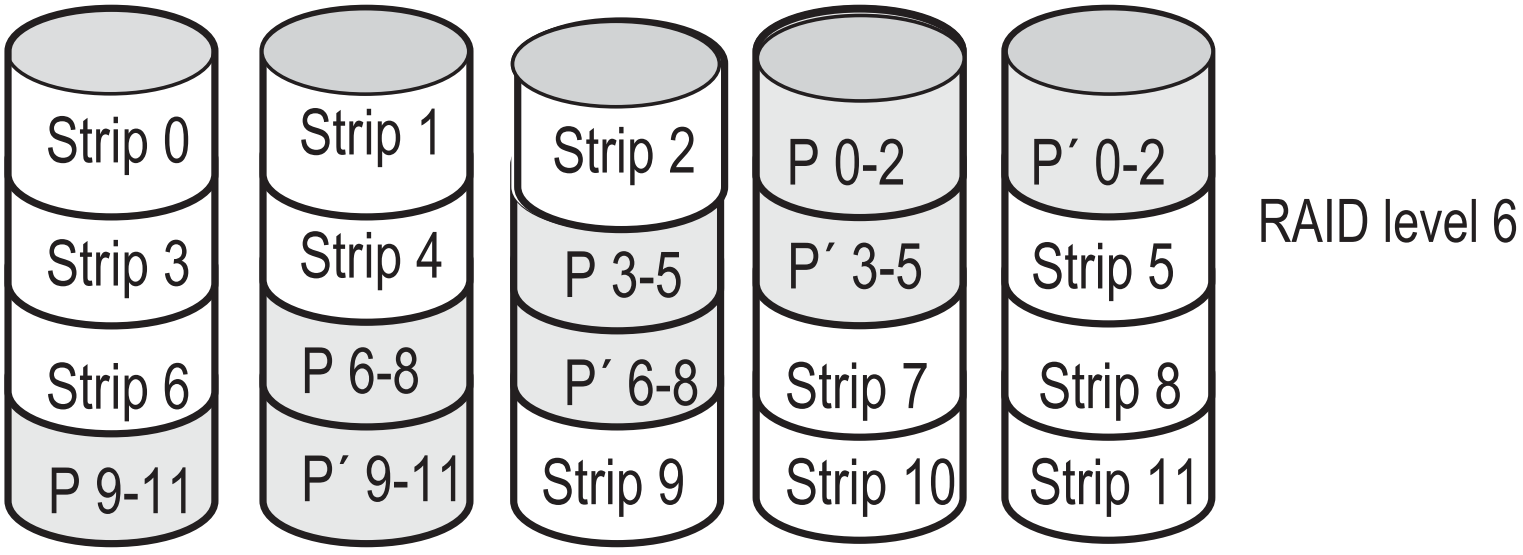
RAID-2 e RAID-3



RAID-4 e RAID-5

UFABC




Strip 0
Strip 3
Strip 6
P 9-11
Strip 1
Strip 4
P 6-8
P´ 9-11
Strip 2
P 3-5
P´ 6-8
Strip 9
P 0-2
P´ 3-5
Strip 7
Strip 10
P´ 0-2
Strip 5
Strip 8
Strip 11

RAID level 6