

# Sistemas Operacionais: Sistema de Arquivos



## Sistema de Arquivos

- Arquivos
  - Espaço contíguo de armazenamento
  - Armazenado em dispositivo secundário
- Estrutura
  - Nenhuma: seqüência de bytes
  - Registros, documentos, programa executável
  - Semântica: SO + aplicação



## Atributos

- Nome
- Tipo (em alguns sistemas)
- Tamanho
- Localização
- Dono do arquivo
- Proteção
- Último acesso
- Última alteração



## Operações

- Criar
- Escrever
- Ler
- Reposicionar o ponteiro do arquivo
- Apagar o arquivo
- Truncar o arquivo
- Mapeamento de arquivo na memória



## Nomes e extensões

- Tamanho máximo de caracteres
- Extensões
  - Em alguns sistemas controla as operações que podemos fazer nos arquivos
    - Exe, doc, bat
  - Opcional



## Tipos de acesso

- Seqüencial
  - Implementação mais simples
  - O arquivo é processado de forma seqüencial
    - Compiladores, editores de texto
  - O ponteiro do arquivo é automaticamente atualizado quando é realizada uma leitura
  - Reposicionamento do ponteiro no início do arquivo
- Acesso direto
  - O arquivo é composto por “registros” de tamanho fixo
  - Uma operação de leitura e escrita é realizada diretamente em um endereço n
  - Acesso randômico: discos



## Diretórios

- Contém informações dos arquivos armazenados no disco
- Cada disco ou partição contém uma estrutura de diretórios
- Operações
  - Buscar um arquivo
  - Criar um arquivo
  - Apagar um arquivo
  - Listar os arquivos
  - Renomear um arquivo
  - Verificar o conteúdo do sistema de arquivo



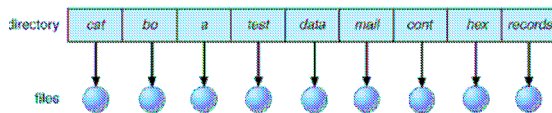
## Diretório

- Eficiência: localizar um arquivo rapidamente
- Nomes: apropriado para usuários
- Agrupamento
  - Arquivos pertencentes a uma mesma aplicação são organizados através dos diretórios



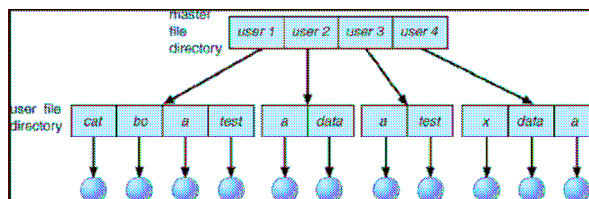
## Diretório – único nível

- Único nível para todos os usuários
- Fácil implementação
- Problemas: conflitos de nome



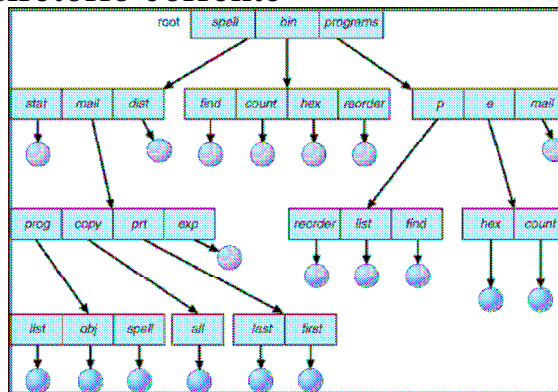
## Diretório – dois níveis

- Cada usuário tem o seu diretório
- Usuários podem ter arquivos com o mesmo nome
- Nomes de arquivos compostos por caminhos (path)
- Busca eficiente



## Diretório – estrutura em arvore

- Agrupamento
- Busca eficiente
- Conceito de diretório corrente

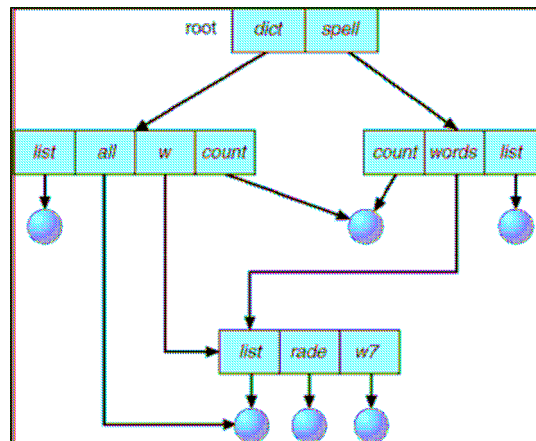


## Diretório – estrutura em arvore

- Caminho relativo e absoluto
- Apagar
  - Arquivo
  - Diretório: apaga todos os subdiretórios e arquivos



## Diretórios- Grafos acíclicos



## Diretórios- Grafos acíclicos

- Diretórios e arquivos compartilhados
- Dois nomes diferentes
- Se *dict* apaga *list*. Se a referência é armazenada na forma de endereço, ocorrerá uma inconsistência
- Se um link é apagado, não haverá problemas.
- Se o arquivo é apagado
  - Deixar os ponteiros perdidos
  - Backpointers: apagar todas as referências aos arquivos do sistema
  - Contador



## Mount/Umount

- Mount
  - um diretório é montado em qualquer parte do sistema de arquivo
  - Endereço relativo e absoluto
- Umount



## Proteção

- Dono do arquivo controla
  - As operações sobre os arquivos
  - Quais os usuários podem fazer operações
- Tipos de acesso
  - Leitura
  - Escrita
  - Execução
  - Apagar
  - Listar



## Lista de acesso

- Leitura, escrita e execução
- Três classes de usuários
  - Dono
  - Grupo
  - Público



## Implementação do sistema de arquivos

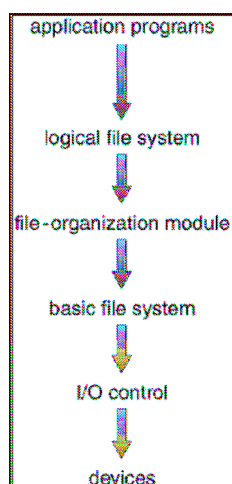


## Sistema de arquivos

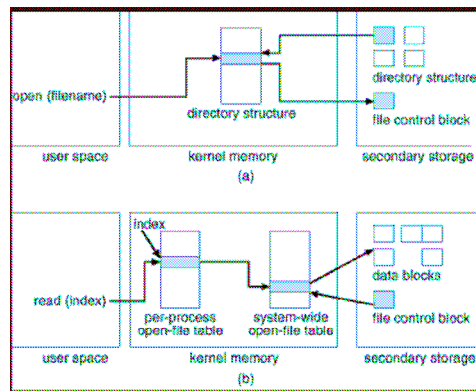
- Organizado em camadas
- Bloco de controle do arquivo (file control block): estrutura de dados que armazena as informações do arquivo
  - Permissões
  - Acessos (data/hora)
  - Dono/ Grupo
  - Tamanho
  - Blocos do arquivo



## Estrutura em camadas



## Estrutura de dados no acesso a arquivos

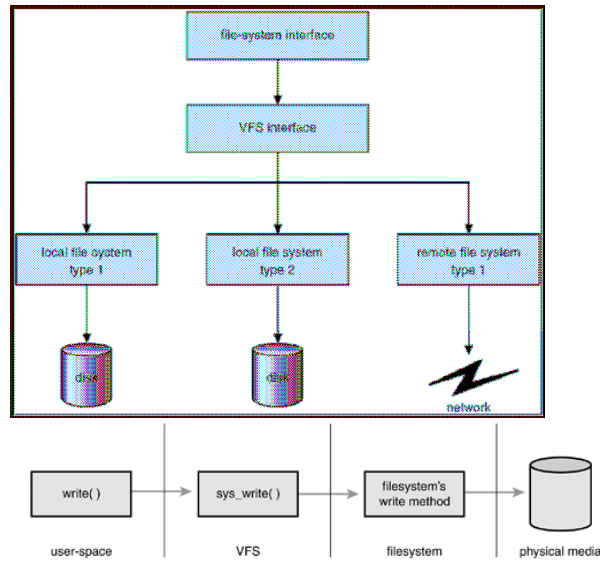


## VFS – Virtual file system

- Modelo orientado a objetos
- API única para acesso a diferentes tipos de sistema de arquivos



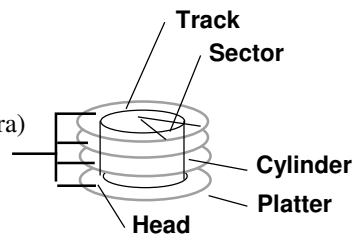
# VFS- virtual file system



# Métodos de alocação

## Arquitetura de um disco

- Latência do disco= tempo médio de busca + tempo médio de atraso rotacional + tempo de transferência + controlador
- Seagate Barracuda 320Gb (2 discos /4 cabeças)
  - Tempo médio de busca = 8.5 ms
  - Atraso rotacional =  $0.5 * (1/7200\text{rpm})$
  - Tempo de transferência = 78 Mbytes/s
  - Tempo controlador= 0.1 ms



- Track-to-track seek time: 1.0 ms (leitura)

## Latência do disco

- Leitura de 64 Kb
- Latência do disco=  $8.5 \text{ ms} + 0.5 * (7200\text{rpm}) + 64\text{Kb} / (78\text{MBytes}) + 0.1 \text{ ms}$
- Latência do disco=  $8.5 \text{ ms} + 0.5 * (7200/60000 \text{ ms}) + 64\text{Kb} / (78\text{Kbytes/ms}) + 0.1 \text{ ms}$
- Latência do disco=  $8.5 + 4.2 + 0.8 + 0.1$
- Latência do disco= 13.6 ms



## Métodos de alocação

- Contígua
- Lista ligada
- Indexada

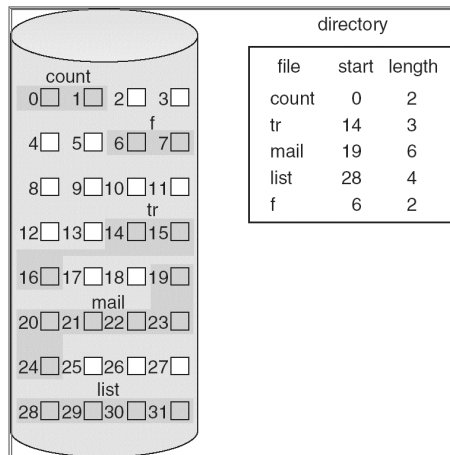


## Alocação contígua

- Cada arquivo contém um conjunto de blocos alocados de forma contígua no disco
- Armazenar apenas o bloco inicial e o número de blocos do arquivo
- Acesso randômico
- Problema da alocação dinâmica
- Aumento de tamanho do arquivo?
  - Não pode crescer
  - Alocar um novo espaço



## Alocação contígua



## Endereçamento – alocação contígua

- Mapeamento Endereço lógico p/ físico
  - End/ 512 (considerando blocos de 512 bytes)
    - Q bloco
    - R deslocamento

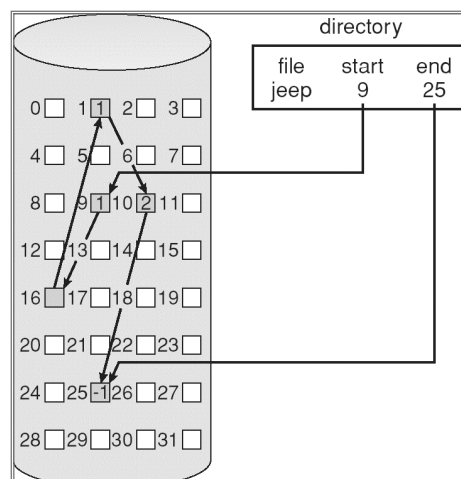


## Alocação – lista ligada

- Cada arquivo composto por uma lista ligada de blocos do disco
- Não precisa ser contígua
- Armazena apenas o bloco inicial
- Sem acesso randômico



## Alocação – lista ligada

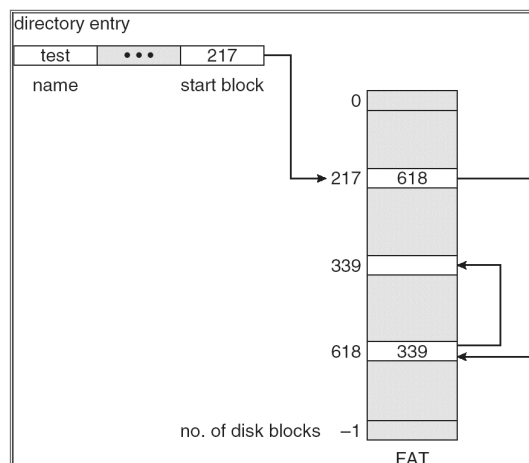


## Endereçamento – lista ligada

- Endereço / 511
  - Q= bloco
  - R+1= endereço do byte

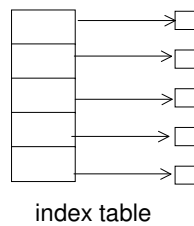


## FAT – file allocation table

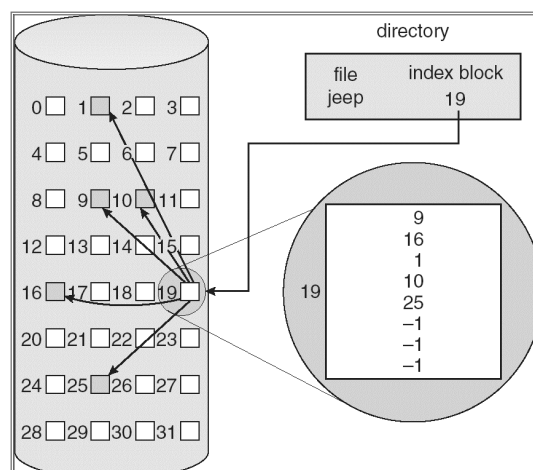


## Alocação indexada

- Todos os ponteiros para o arquivo são armazenados em uma tabela
  - Tabela de índices
- Acesso randômico



## Alocação indexada

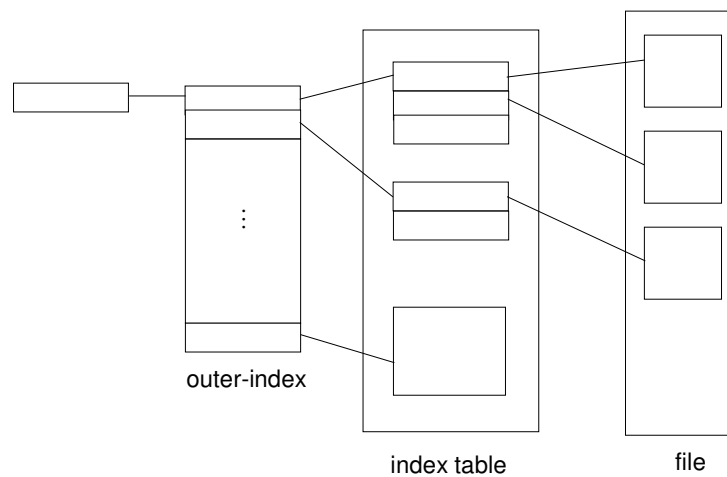


## Endereçamento- alocação indexada

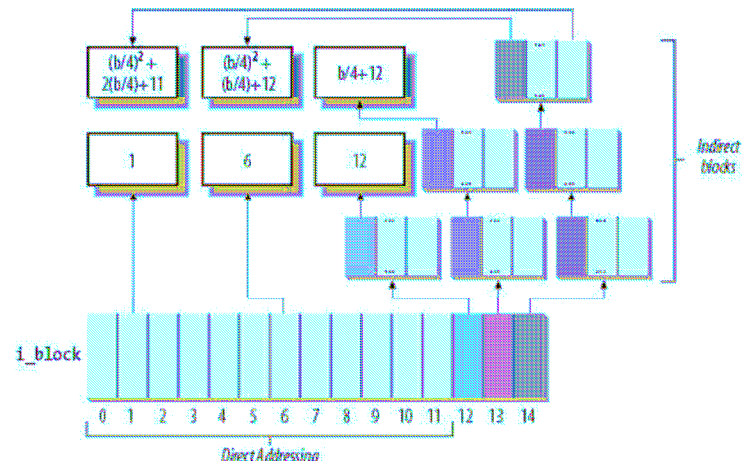
- Endereço /512
  - Q = numero bloco
  - R = deslocamento



## Tabela de indices- 2 níveis



# UNIX



## Gerenciamento de espaços livres



## Mapa de bit

- Cada bit representa um bloco do disco
- Calculo do bloco livre
  - (número de bits por palavra) \* (número de 0) + deslocamento até o primeiro bit 1
- É necessário armazenar os mapas de bits no disco
  - Qual o overhead?
- Fácil obtenção de um espaço contíguo

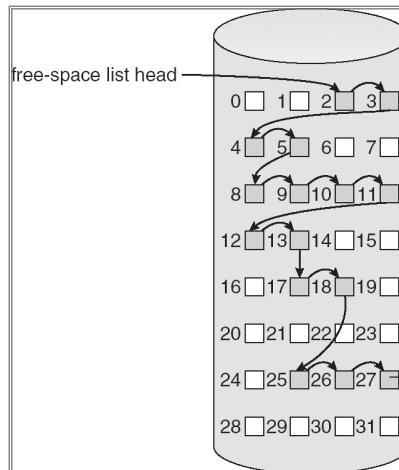


## Lista ligada

- Espaço livre: Armazenar apenas o ponteiro do início da lista
  - Sem perda de espaço
- Difícil obtenção de um conjunto de blocos contíguos



## Lista ligada



## Sistema baseado em Log

- Log (journaling): cada atualização no sistema de arquivos é armazenada em um arquivo de log (transação)
  - Uma transação é considerada aceita quando, quando escrita no arquivo de log
  - O sistema de arquivo não necessariamente está atualizado
- As transações no arquivo de log são processadas de forma assíncrona
  - Quando a transação é processada, a mesma é armazenada no arquivo de log
- Se um sistema é reinicializado, as transações no arquivo são processadas antes do início da utilização do sistema
- Duas escritas são necessárias: log + sistema de arquivos



## Ext3

- Journaling pode ser configurado em 3 modos
  - Journal: armazena no arquivo de log os dados e metadados (diretórios e informações sobre o arquivo) no arquivo de log.
    - Duas escritas
  - Ordered: armazena no arquivo de log apenas as atualizações nos metadados.
    - As atualizações no conteúdo do arquivo são armazenadas diretamente no sistema de arquivo. Os metadados são escritos em definitivo depois da atualização
  - WriteBack: também armazena no arquivo de log apenas as atualizações nos metadados.
    - As atualizações nos metadados e nos arquivos são realizados de forma assíncrona.



## Escalonamento do disco



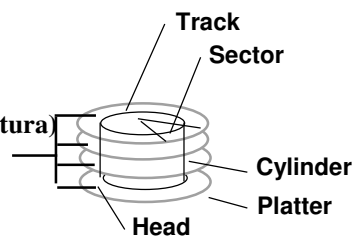
## Escalonamento do disco

- O sistema operacional é responsável por utilizar o HW de forma eficiente – disco: acesso rápido e largura de banda
- Acesso
  - Tempo de busca
  - Latência rotacional
- Minimizar o tempo de busca
- Tempo de busca  $\approx$  Distância da busca



## Arquitetura de um disco

- Latência do disco = tempo médio de busca + tempo médio de atraso rotacional + tempo de transferência + controlador
- Seagate Barracuda 320Gb (2 discos / 4 cabeças)
  - Tempo médio de busca = 8.5 ms
  - Atraso rotacional =  $0.5 * (1/7200\text{rpm})$
  - Tempo de transferência = 78 Mbytes/s
  - Tempo controlador = 0.1 ms
- Track-to-track seek time: 1.0 ms (leitura)



## Algoritmos para escalonamento do disco

- Fila de requisições(0-199)

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

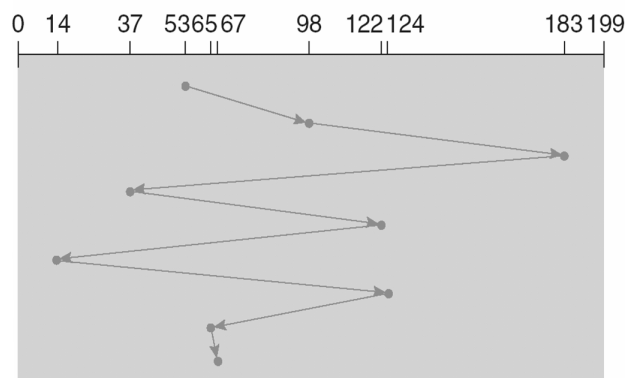
Cabeça do disco: 53



## Primeiro a chegar, primeiro a ser servido

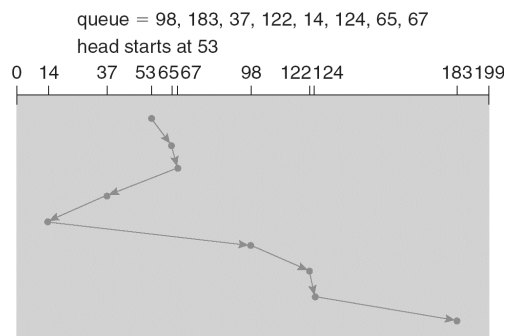
- Total de movimentos: 640 cilindros

queue = 98, 183, 37, 122, 14, 124, 65, 67  
head starts at 53



## Menor tempo de busca primeiro

- Seleciona o menor tempo de busca considerando a posição atual da cabeça do disco
- Pode ocorrer starvation

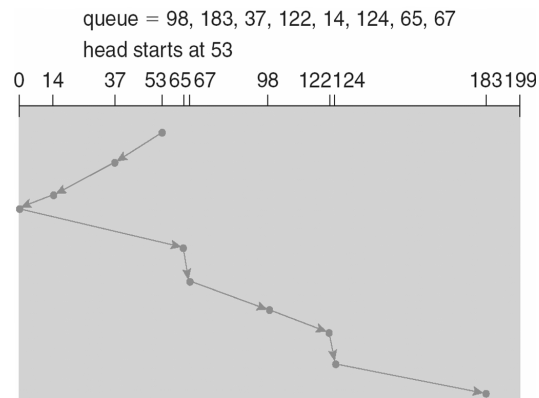


236 movimentos



## SCAN

- Algoritmo do elevador: atende todas as requisições em uma direção e depois retorna atendendo as demais

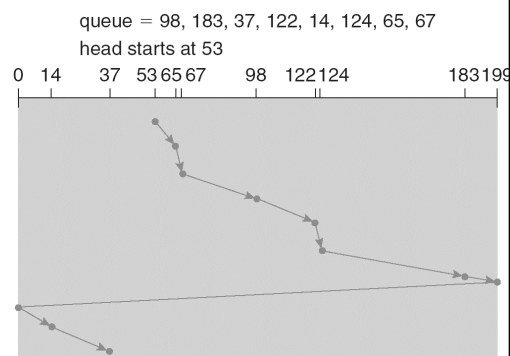


208 movimentos



## C-SCAN

- Atendimento mais uniforme
- Move a cabeça em uma direção
  - Quando ele chega ao final, retorna imediatamente ao início do disco



## RAID

- RAID – redundant array of inexpensive disks ou redundant array of independent disks. Redundância -> aumenta a disponibilidade
- O sistema continua respondendo na presença de falhas
- RAID é definido em 6 níveis diferentes



# RAID

- Vários discos trabalhando cooperativamente
- Disk striping utiliza faixas de um conjunto de discos como uma única unidade de armazenamento
- RAID aumenta o desempenho e a confiabilidade através da redundância dos dados
  - Mirroring or shadowing (RAID 1) duplica cada disco
  - Faixas espelhadas (RAID 1+0) or espelhamento de faixas (RAID 0+1) alto desempenho e alta confiabilidade
  - Block interleaved parity (RAID 4, 5, 6) menor redundância
- Normalmente discos reservas vazios estão disponíveis para a troca e reconstrução dos dados



## RAID: níveis

(a) RAID 0: non-redundant striping.

(b) RAID 1: mirrored disks.

(c) RAID 2: memory-style error-correcting codes.

(d) RAID 3: bit-interleaved parity.

(e) RAID 4: block-interleaved parity.

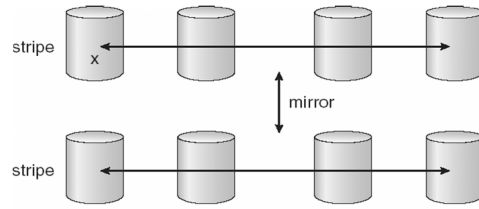
(f) RAID 5: block-interleaved distributed parity.

(g) RAID 6: P + Q redundancy.

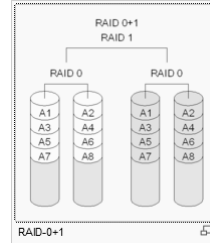
Nível	Falhas	Discos de Dados	Disco de check
0	0	4	0
1	1	4	4
2	1	4	3
3	1	4	1
4	1	4	1
5	1	4	1
6	2	4	2



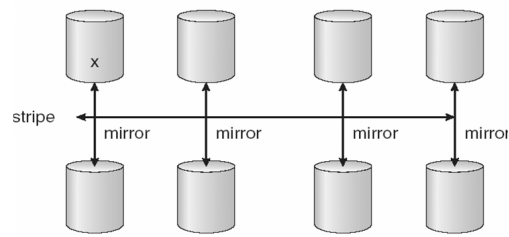
# RAID (0+1) e (1+0)



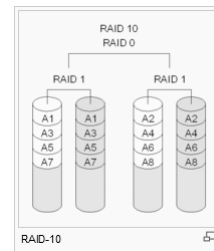
a) RAID 0 + 1 with a single disk failure.



RAID-0+1



b) RAID 1 + 0 with a single disk failure.



RAID-1+0

