

Sistemas Operacionais

- Discos
- Arquivos
- Diretórios



DISCOS

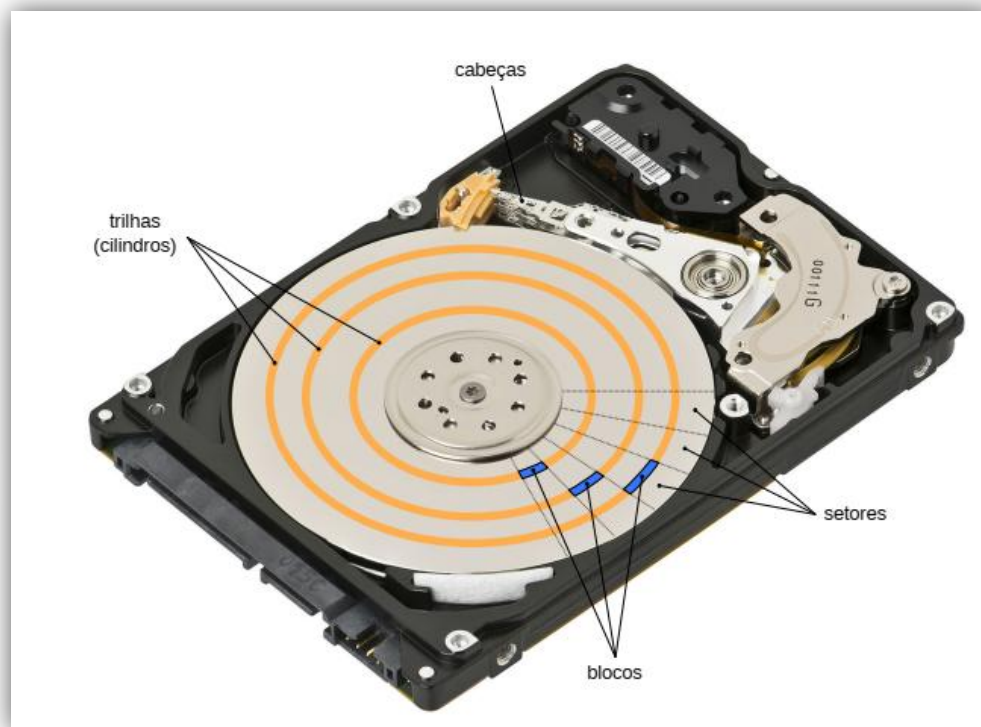


Discos

- Um disco permite o armazenamento persistente (não-volátil) de grandes volumes de dados com baixo custo e tempos de acesso razoáveis.
- A leitura e escrita de dados em um disco é mais simples e flexível que em outros meios.
- Os discos também são frequentemente usados como área de armazenamento de páginas em sistemas de paginação em disco.

Discos

- Um disco típico contém várias faces e milhares de trilhas e de setores por face resultando em milhões de blocos de dados disponíveis.
- Cada bloco pode ser individualmente acessado (lido ou escrito) através de seu *endereço*.



Discos

O controlador de disco rígido normalmente é conectado a um barramento do computador através de uma interface de conexão que pode usar diversas tecnologias:

*IDE (Integrated Drive Electronics) ***

- *Padrão também conhecido como PATA (Parallel ATA - Advanced Technology Attachment); Anos 1980. Suporta velocidades de até 1 Gbit/s.*

*SCSI (Small Computer System Interface)***

- *Anos 1980, foi muito usada em servidores e estações de trabalho de alto desempenho. Atinge taxas de transferência de até 2,5 Gbit/s.*

SAS (Serial Attached SCSI)

- É uma evolução do padrão SCSI, permitindo atingir taxas de transferência de até 12 Gbit/s em cada dispositivo conectado ao controlador. É usado em equipamentos de alto desempenho, como servidores.

SATA (Serial ATA)

- É o padrão de interface de discos em *desktops* e *notebooks* atuais. A transmissão dos dados entre o disco e a controladora é serial, atingindo taxas de transferência de 6 Gbit/s através de cabos com 7 vias.



Discos- Escalonamento de acessos

Em um sistema operacional multitarefas, várias aplicações e processos podem solicitar acessos ao disco simultaneamente, para escrita e leitura de dados.

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.

Considere um disco hipotético com 1.000 blocos (enumerados de 0 ao 999), cuja cabeça de leitura se encontra inicialmente sobre o bloco 500. A fila de pedidos de acesso pendentes contém pedidos de acesso aos seguintes blocos do disco, em sequência:

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



Discos- Escalonamento de acessos

FCFS (*First Come, First Served*): este algoritmo consiste em atender as requisições na ordem da fila, ou seja, na ordem em foram pedidas pelos processos.

Se os pedidos de acesso estiverem muito espalhados pelo disco, este irá perder muito tempo movendo a cabeça de leitura de um lado para o outro.

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

Percebe-se que, para atender os pedidos de leitura na ordem indicada pelo algoritmo FCFS, a cabeça de leitura teve de deslocar-se por 2.613 blocos do disco ($222 + 636 + 467 + \dots$).



Discos- Escalonamento de acessos

SSTF (*Shortest Seek Time First – Menor Tempo de Busca Primeiro*): esta estratégia de escalonamento de disco consiste em sempre atender o pedido que está mais próximo da posição atual da cabeça de leitura (que é geralmente a posição do último pedido atendido).

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

Apesar de oferecer um ótimo desempenho, a estratégia SSTF pode levar à inanição (starvation). Para resolver esse problema, torna-se necessário implementar uma estratégia de envelhecimento dos pedidos pendentes.



Discos- Escalonamento de acessos

SCAN: Neste algoritmo, a cabeça “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.

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 (1.337 blocos)

Ele é adequado para sistemas com muitos pedidos simultâneos de acesso a disco, como servidores de arquivos.



Discos- Escalonamento de acessos

C-SCAN: esta é uma variante “circular” do algoritmo SCAN, na qual a cabeça de leitura varre o disco somente em um sentido. Ao atingir o final do disco, ela retorna diretamente ao início do disco, sem atender os pedidos intermediários, e recomeça a varredura.

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 (1.776 blocos)

O nome “circular” é devido ao disco ser visto pelo algoritmo como uma lista circular de blocos. Sua vantagem em relação ao algoritmo SCAN é prover um tempo de espera mais homogêneo aos pedidos pendentes, o que é importante em servidores.



Discos- Escalonamento de acessos

LOOK: é uma otimização do algoritmo SCAN, na qual a cabeça do disco não avança até o final do disco, mas inverte seu movimento assim que tiver tratado o último pedido em cada sentido do movimento.

Deadline: este escalonador é baseado no algoritmo do C-SCAN, mas associa um prazo (*deadline*) a cada requisição, para evitar problemas de inanição.



Discos- Escalonamento de acessos

Noop (*No-Operation*): Este escalonador é voltado para discos de estado sólido (SSD, baseados em memória *flash*) ou sistemas de armazenamento que façam seu próprio escalonamento interno, como sistemas RAID.

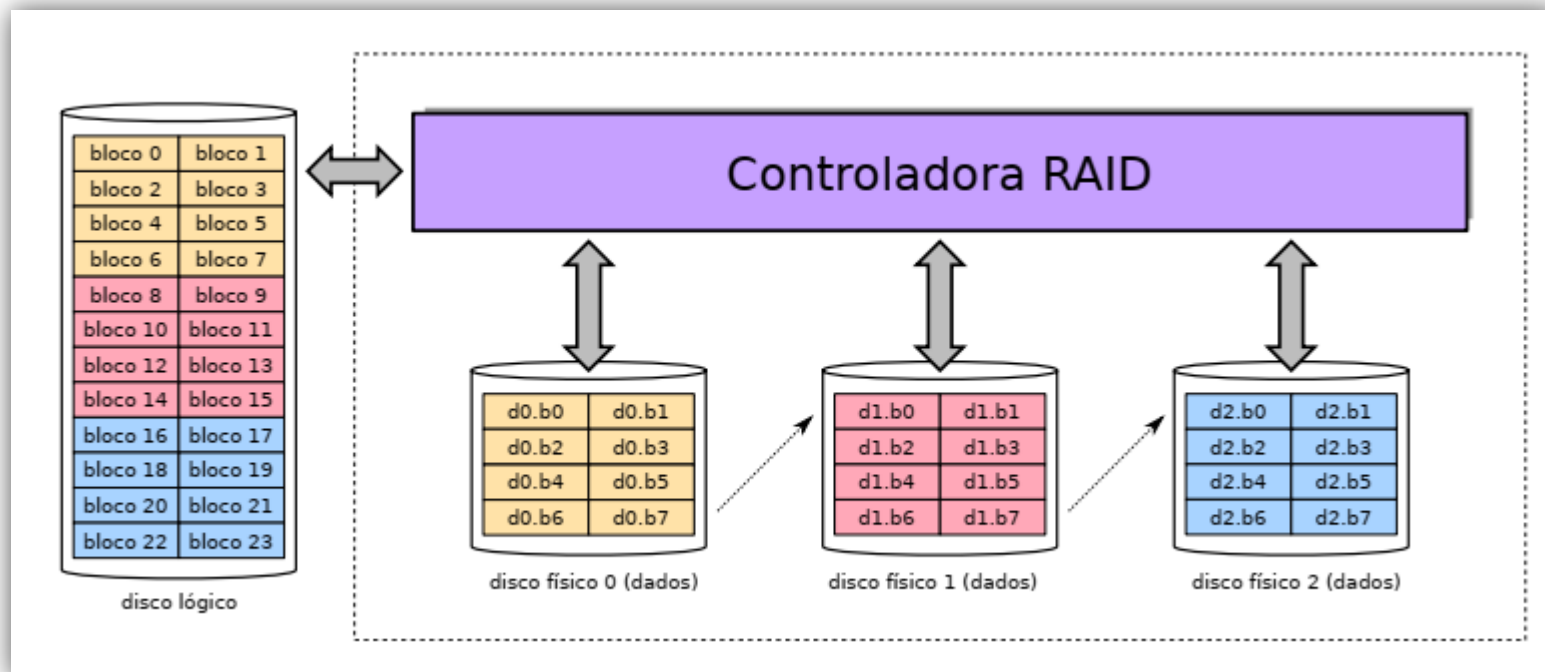
CFQ (*Completely Fair Queuing*): os pedidos dos processos são divididos em várias filas (64 filas por default); cada fila recebe uma fatia de tempo para acesso ao disco, que varia de acordo com a prioridade de entrada/saída dos processos contidos na mesma. **Este é o escalonador default do Linux na maioria das distribuições mais recentes**

Discos- RAID

- Um sistema RAID (*Redundant Array of Independent Disks*) é constituído de dois ou mais discos rígidos que são vistos pelo sistema operacional e pelas aplicações como um único disco lógico.
- O objetivo central de um sistema RAID é proporcionar mais desempenho (através do paralelismo) e mais confiabilidade (redundância).
- Um sistema RAID pode ser construído “por hardware” (placas) ou “por software”, na qual são usados *drivers* apropriados dentro do sistema operacional para combinar os discos rígidos conectados ao computador em um único disco lógico.

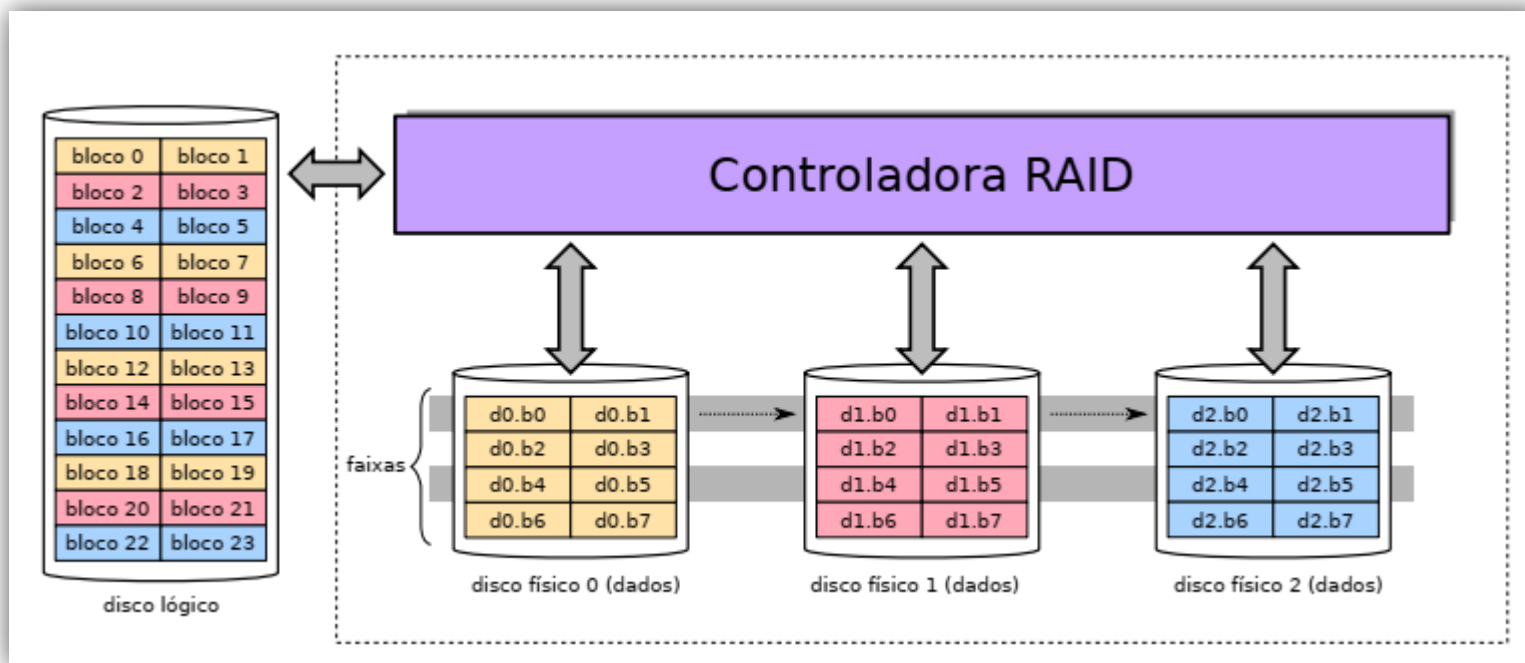
Discos- RAID

RAID 0 (linear): Neste nível os discos físicos (ou partições) são simplesmente concatenados em sequência para construir um disco lógico.



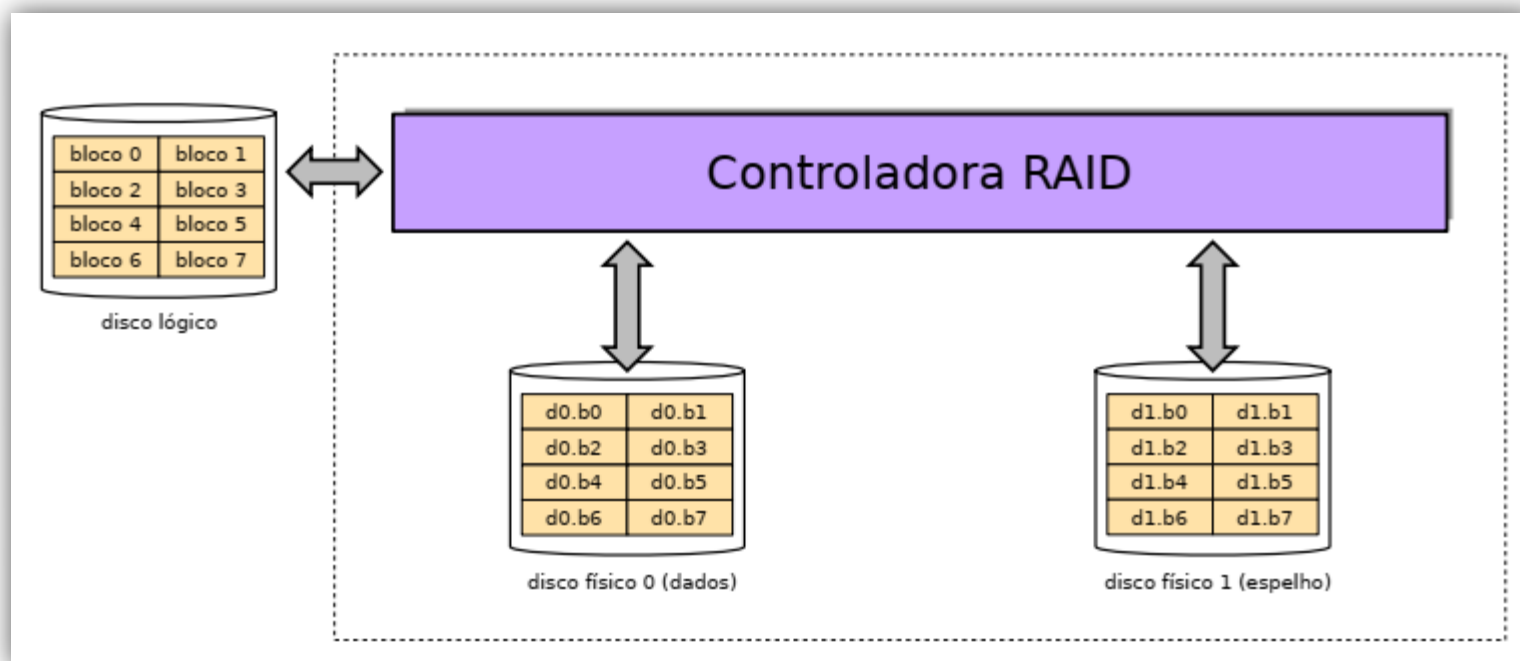
Discos- RAID

RAID 0 (striping): neste nível os discos físicos são divididos em áreas de tamanhos fixo chamadas *fatias* ou *faixas* (*stripes*). O maior espalhamento dos blocos sobre os discos físicos contribui para distribuir melhor a carga de acessos entre eles e assim ter um melhor desempenho.



Discos- RAID

RAID 1 (Espelhamento) neste nível, o conteúdo é replicado em dois ou mais discos. O espelhamento também pode ser associado a conjuntos de discos em RAID 0, levando a configurações híbridas como RAID 0+1.



Discos- RAID

RAID 2: neste nível os dados são “fatiados” **em bits individuais** que são escritos nos discos físicos em sequência; discos adicionais são usados para armazenar códigos corretores de erros (*Hamming Codes*).

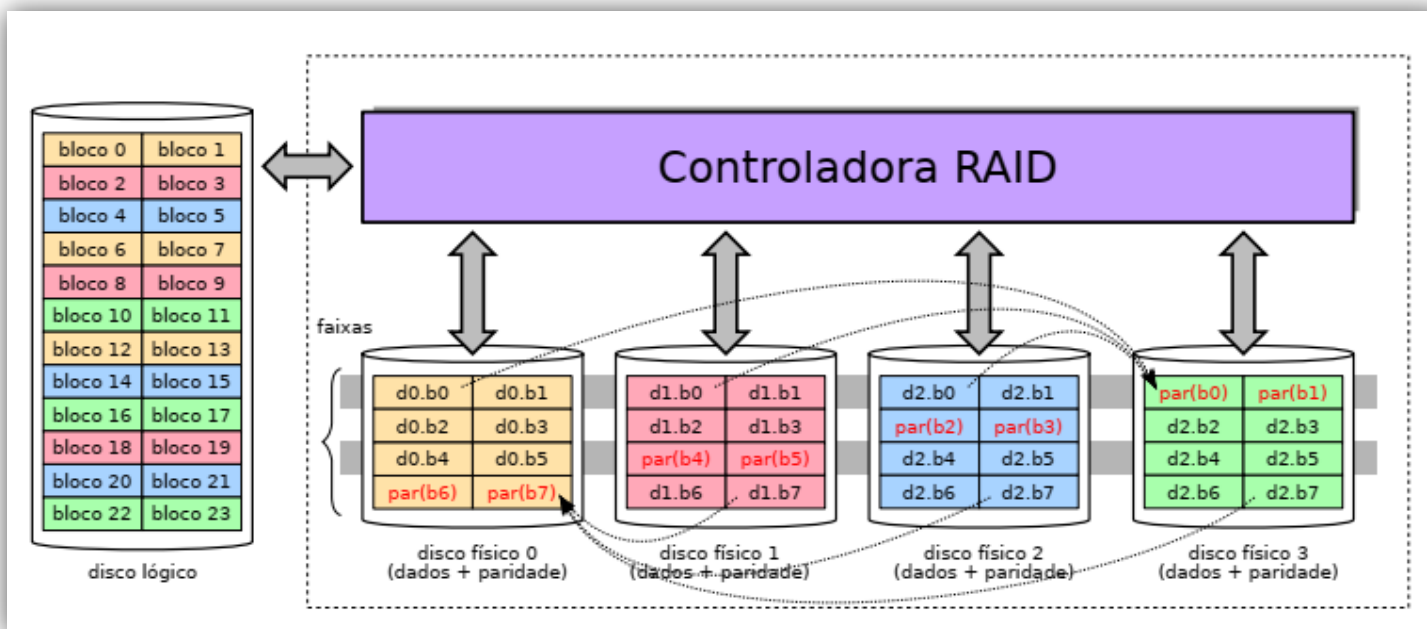
A cada leitura ou escrita, os dados de paridade devem ser atualizados, o que transforma o disco de paridade em um gargalo de desempenho.

RAID 3: de forma similar ao RAID 2, este nível fatia os **dados em bytes** escritos nos discos em sequência.

RAID 4: Esta abordagem é similar ao RAID 3, com a diferença de que o fatiamento é **feito em blocos** ao invés de bits.

Discos- RAID

RAID 5: assim como o RAID 4, esta abordagem também armazena informações de paridade para tolerar falhas em blocos ou discos. Todavia, essas informações **não ficam concentradas em um único disco físico, sendo distribuídas uniformemente entre eles.**



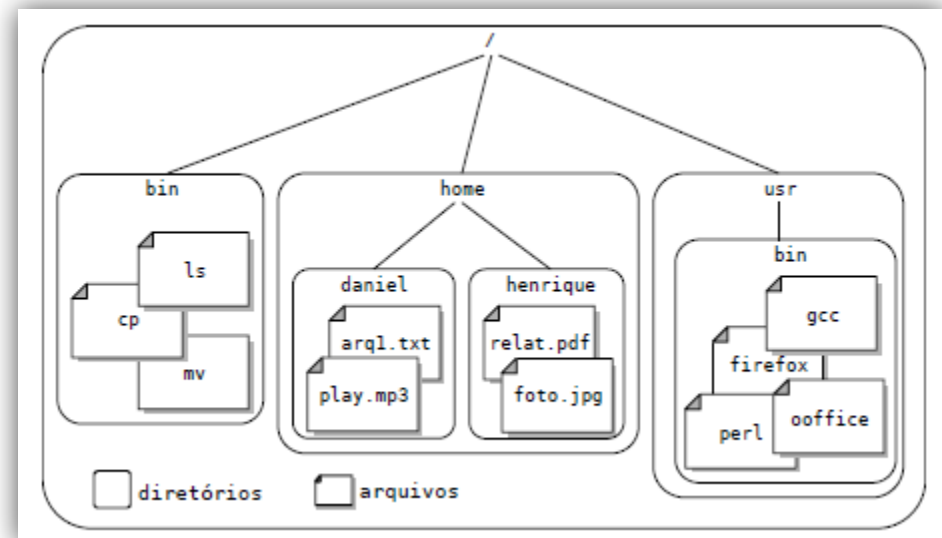
ARQUIVOS



Definição Arquivo

Conjunto de dados armazenados em um dispositivo físico não-volátil, com **um nome ou outra referência que permita sua localização posterior**.

São organizados numa estrutura hierárquica chamada de diretórios.



Atributos de Arquivos

Os arquivos possuem um conjunto de características conforme o sistema de arquivos:

Nome

- *Texto* que identifica o arquivo para o usuário, como “foto1.jpg”, “documento.pdf”, “hello.c”, etc.;

Tipo

- Indicação do formato dos dados contidos no arquivo, como áudio, vídeo, imagem, texto, etc. Muitos sistemas operacionais usam parte do nome do arquivo para identificar o tipo de seu conteúdo, na forma de uma extensão: “.doc”, “.jpg”, “.mp3”, etc.;

Tamanho

- Indicação do tamanho do conteúdo do arquivo, geralmente em bytes;

Datas

- Para fins de gerência, é importante manter as datas mais importantes relacionadas ao arquivo, como suas datas de criação, de último acesso e de última modificação do conteúdo;

Proprietário

- Em sistemas multiusuários, cada arquivo tem um proprietário, que deve estar corretamente identificado;

Permissões de acesso

- Indicam que usuários têm acesso àquele arquivo e que formas de acesso são permitidas (leitura, escrita, remoção, etc.);

Localização

- Indicação do dispositivo físico onde o arquivo se encontra e da posição do arquivo dentro do mesmo;

Operações com Arquivos

Geralmente implementadas por meio de chamadas de sistemas:

Criar

- A criação de um novo arquivo implica em alocar espaço para ele no dispositivo de armazenamento e definir valores para seus atributos (nome, localização, proprietário, permissões de acesso, datas, etc.);

Abrir

- Antes que uma aplicação possa ler ou escrever dados em um arquivo, ela deve solicitar ao sistema operacional a “abertura” desse arquivo. O sistema irá então verificar se o arquivo desejado existe, verificar se as permissões associadas ao arquivo permitem aquele acesso, localizar seu conteúdo no dispositivo de armazenamento e criar uma referência para ele na memória da aplicação;

Ler

- Permite transferir dados presentes no arquivo para uma área de memória da aplicação;

Escrever

- Permite transferir dados na memória da aplicação para o arquivo no dispositivo físico; os novos dados podem ser adicionados ao final do arquivo ou sobrescrever dados já existentes;

Fechar

- Ao concluir o uso do arquivo, a aplicação deve informar ao sistema operacional que o mesmo não é mais necessário, a fim de liberar as estruturas de gerência do arquivo mantidas na memória do núcleo;

Remover

- Para eliminar o arquivo do dispositivo, descartando seus dados e liberando o espaço ocupado por ele.

Alterar atributos

- Para modificar os valores dos atributos do arquivo, como nome, proprietário, permissões, datas, etc.

Formato dos Arquivos

Arquivos permitem armazenar dados para uso posterior. A forma como esses dados são estocados dentro do arquivo é denominada *formato do arquivo*.

Núcleo de S.O. reconhece:

- Binários executáveis.
- Bibliotecas.

Outros formatos são interpretados pelas aplicações.

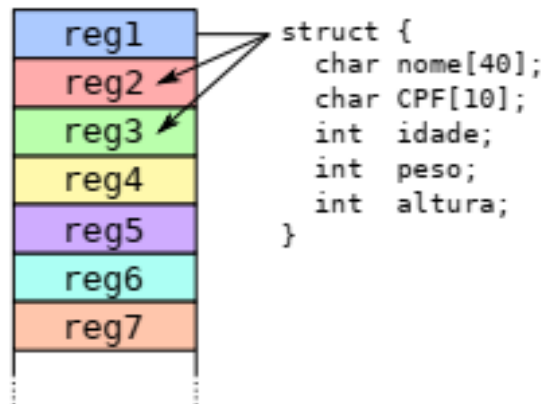
- Formato definido pela aplicação.
- Formato público padronizado: JPEG, GIF, PNG, etc

(Para o SO, os outros formato são apenas sequência de bytes).

Formato: arquivo de registro

Alguns núcleos do sistema operacional fornecem arquivos com estrutura interna.

- Sequência linear de registros (tamanho fixo ou variável)
- Arquivos indexados (pares de chave/valor)
 - EX. Berkeley Bd e SQLite.



nome (chave)	telefone (valor)
daniel	9977-1173
marina	9876-5432
henrique	8781-9750
gabriel	8858-8286
renata	9663-9293
andressa	8779-5538
guilherme	9979-4166

Formato: Arquivo de Texto

Formado por linhas de caracteres ASCII

Separados por caracteres de controle

- New line no Unix: ASCII 10 ou “\n”
- No windows: Carriage return (ASCII 13 ou “\r”) seguido de New line (ASCII 10 ou “\n”)

Formato: Arquivo de Texto

```
int _main()
{
    _printf("Hello, _world\n");
    _exit(0);
}
```

Representação no UNIX

1	0000	69 6e 74 20 6d 61 69 6e 28 29 0a 7b 0a 20 20 70
2		i n t _ m a i n () \n { \n _ _ p
3	0010	72 69 6e 74 66 28 22 48 65 6c 6c 6f 2c 20 77 6f
4		r i n t f (" H e l l o , _ w o
5	0020	72 6c 64 5c 6e 22 29 3b 0a 20 20 65 78 69 74 28
6		r l d \ n ") ; \n _ _ e x i t (
7	0030	30 29 3b 0a 7d 0a
8		0) ; \n } \n

Representação WINDOWS

1	0000	69 6e 74 20 6d 61 69 6e 28 29 0d 0a 7b 0d 0a 20
2		i n t _ m a i n () \r \n { \r \n _
3	0010	20 70 72 69 6e 74 66 28 22 48 65 6c 6c 6f 2c 20
4		_ p r i n t f (" H e l l o , _
5	0020	77 6f 72 6c 64 5c 6e 22 29 3b 0d 0a 20 20 65 78
6		w o r l d \ n ") ; \r \n _ _ e x
7	0030	69 74 28 30 29 3b 0d 0a 7d 0d 0a
8		i t (0) ; \r \n } \r \n

Arquivos Executáveis

Depende do Sistema Operacional:

ELF (*Executable and Linking Format*): formato de arquivo usado para programas executáveis e bibliotecas na maior parte das plataformas UNIX modernas.

É composto por um cabeçalho e várias seções de dados, contendo código executável, tabelas de símbolos e informações sobre relocação de código, usadas quando o código executável é carregado na memória.

- **PE** (*Portable Executable*): é o formato usado para executáveis e bibliotecas na plataforma Windows.

Consiste basicamente em uma extensão do formato COFF (*Common Object File Format*), usado em plataformas UNIX mais antigas.

Identificação de Conteúdo

PROBLEMA: Como saber qual o tipo de informação os bytes de um arquivo de dados representa?

1ª solução: extensão do arquivo

Inserida pelo DOS

Exemplo: sol.jpg

2ª solução: bytes no início do arquivo

Sistemas Unix

3ª solução: atributo adicional

MacOS

Atributo *file type*

Atributo *creator application*

4ª solução: tipos MIME (*Multipurpose Internet Mail Extensions*)

Emails

tipo/subtipo

Identificação de Conteúdo

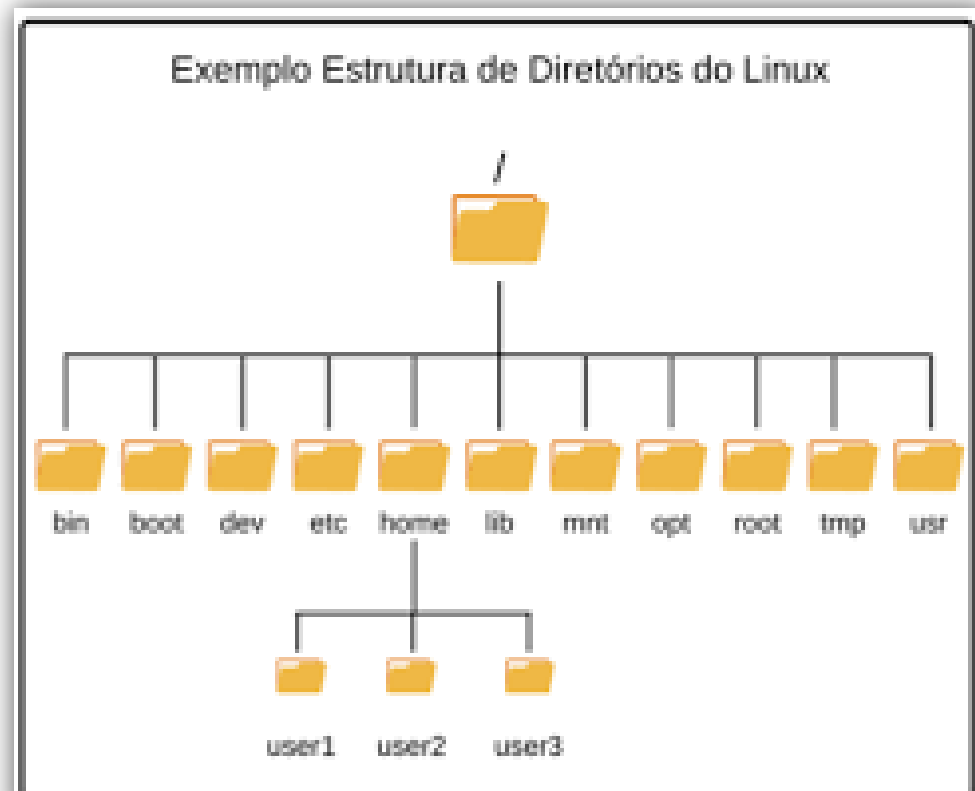
O comando file no Linux é utilizado para se determinar qual é o tipo de arquivo informado como parâmetro com base no Magic Number (dois primeiros bytes)

Tipo de arquivo	bytes iniciais	Tipo de arquivo	bytes iniciais
Documento PostScript	%!	Documento PDF	%PDF
Imagem GIF	GIF89a	Imagem JPEG	0xFF D8 FF
Música MIDI	MThd	Classes Java	0xCA FE BA BE
Arquivo ZIP	0x50 4B 03 04	Documento RTF	{\rtf1

ARQUIVOS ESPECIAIS

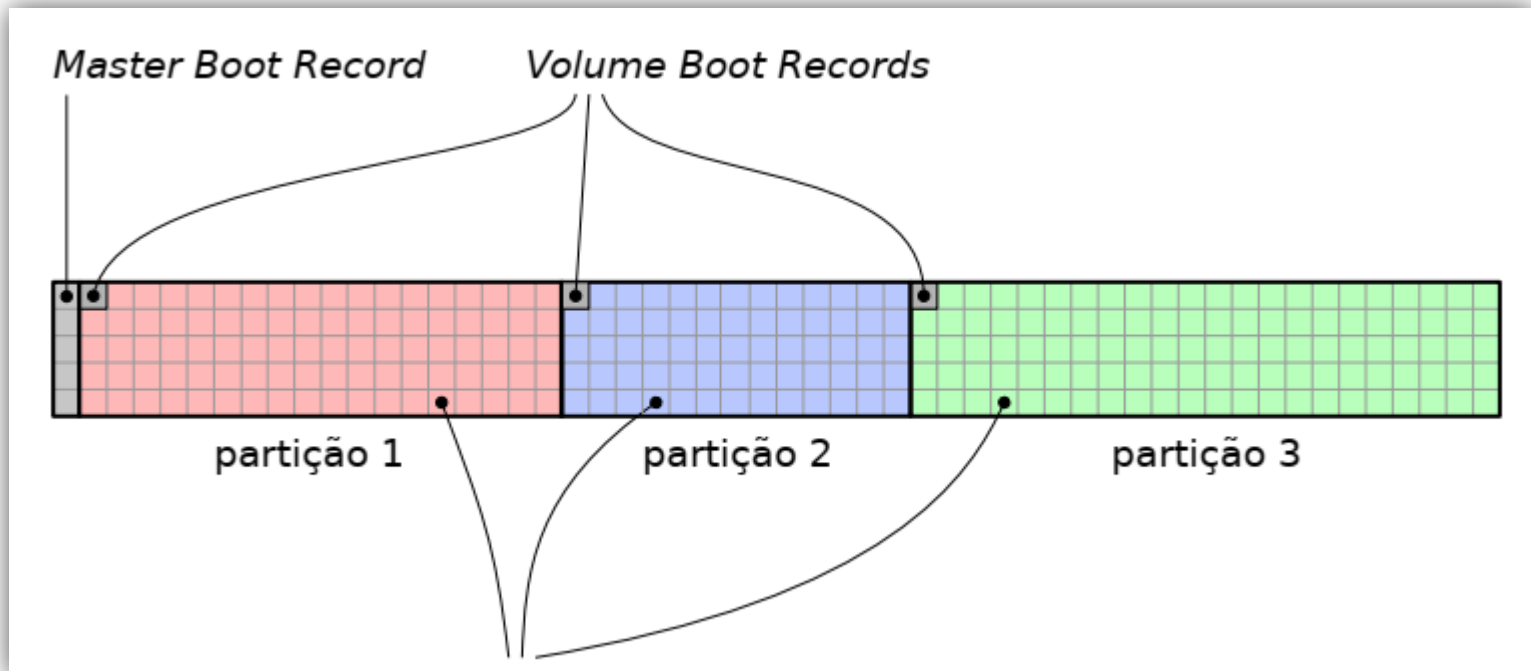
- 1. Que tipo de abstração é feito no Linux para os arquivos existentes na pasta /dev? O que eles mapeiam?**
- 2. Que tipo de mapeamento é feito em cada um dos arquivos abaixo:**
 - a) /dev/ttyS0
 - b) /dev/audio
 - c) /dev/sda1
- 3. Que tipo de abstração é feito no Linux para os arquivos existentes na pasta /proc e na pasta /sys? O que eles mapeiam?**
- 4. Que tipo de mapeamento é feito em cada um dos arquivos abaixo:**
 - a) /proc/cpuinfo
 - b) /proc/<PID>/maps
 - c) /sys/block/sda/queue/scheduler

DIRETÓRIOS



Organização de Volumes

Cada dispositivo ou partição devidamente preparado e formatado para receber um sistema de arquivos é designado como um volume.



Organização de Volumes

Existem diversos formatos possíveis para os blocos de inicialização de disco e de partição, além da estrutura da própria tabela de partição. Esses formatos devem ser reconhecidos pelo código de inicialização do computador (BIOS) e pelo sistema operacional instalado, para que os dados do disco possam ser acessados.

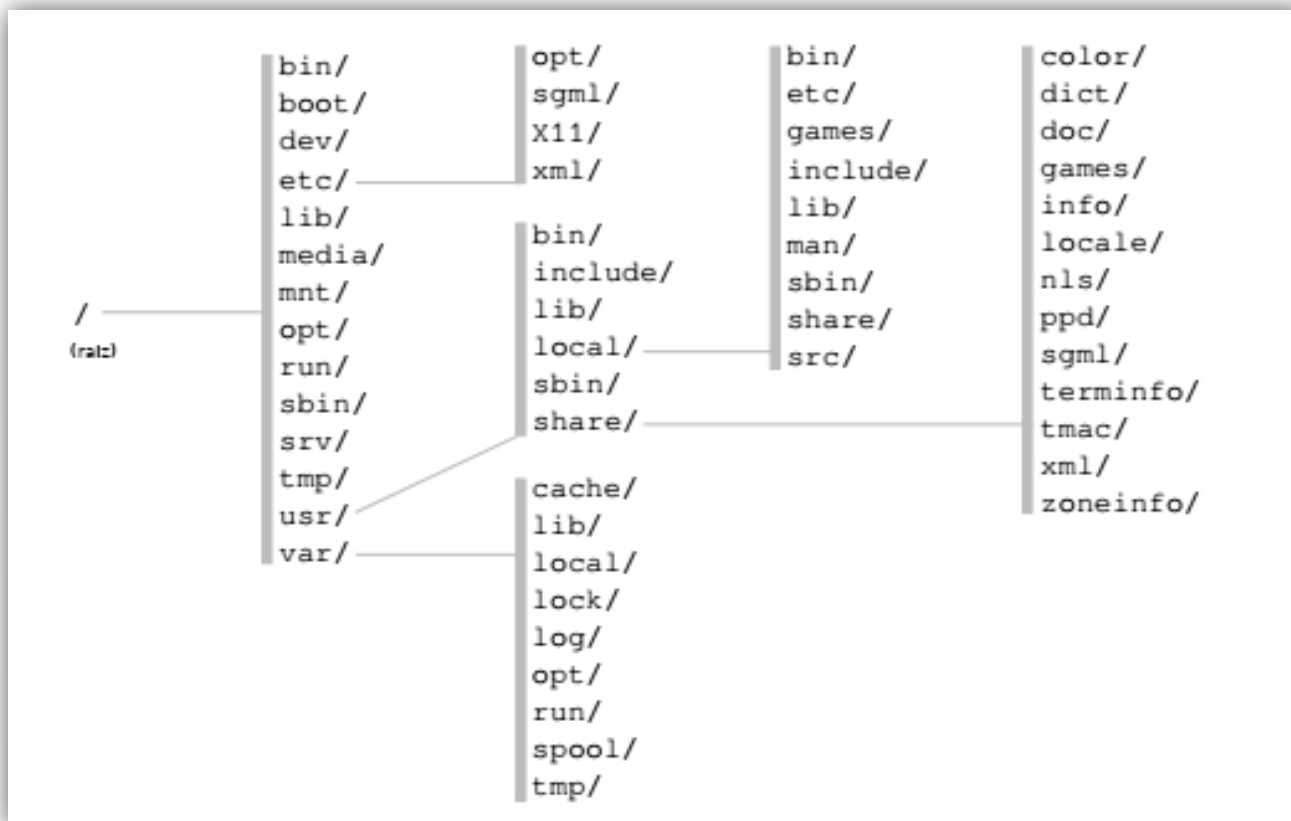
Um termo frequentemente utilizado em sistemas de arquivos é o **volume**, que significa um espaço de armazenamento de dados, do ponto de vista do sistema operacional.

Em sua forma mais simples, cada volume corresponde a uma partição, mas configurações mais complexas são possíveis.

Antes de ser usado, cada volume ou partição deve ser *formatado*, ou seja, preenchido com as estruturas de dados necessárias para armazenar arquivos, diretórios, atalhos e outras entradas.

Diretórios

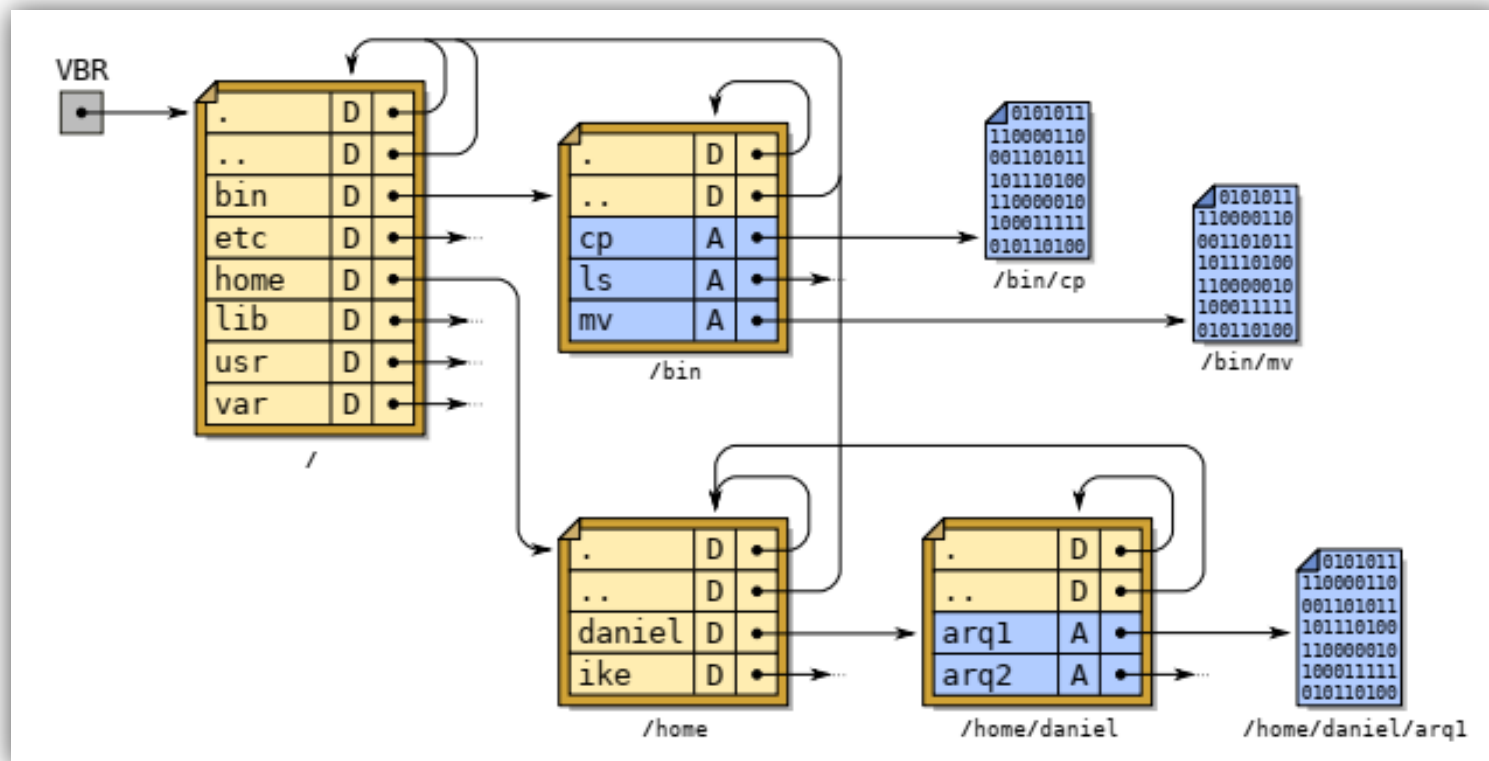
Container para organização e separação de arquivos organizado em formato de árvore Contendo um diretório raiz



DIRETÓRIOS

Diretórios são implementados como um arquivo.

- Conteúdo é uma relação de entradas (arquivos ou outros diretórios)
- Diretório não contém fisicamente os arquivos e subdiretórios, o diretório apenas aponta.



Atalhos

ATALHO SIMBÓLICO (soft link)

- Arquivo com a string do caminho do alvo.
- Ao remover o alvo, o atalho permanece, deixará de funcionar.
- Pode apontar para qualquer local, dispositivo.

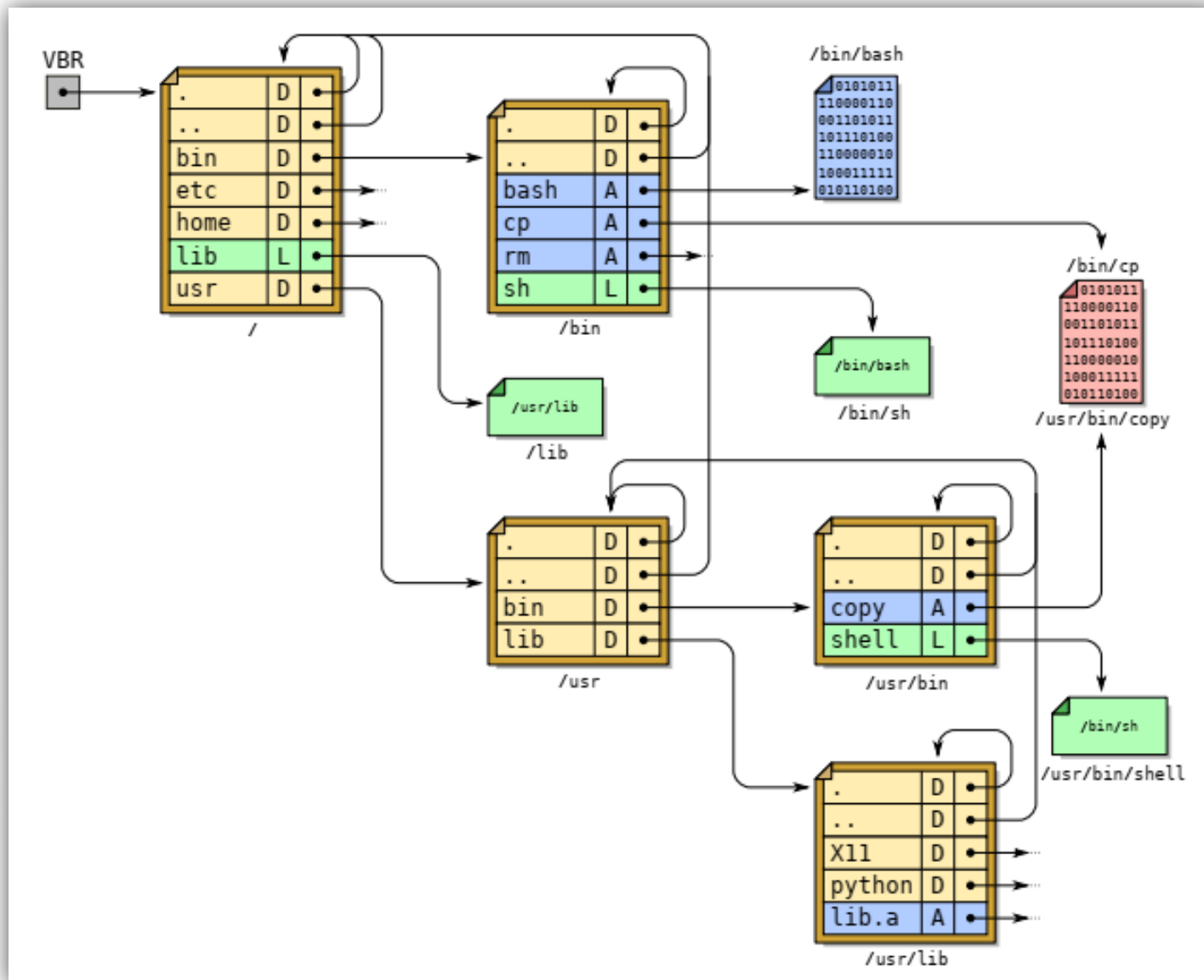
```
ln -s arquivo link_arquivo
```

ATALHO FÍSICO (hard link)

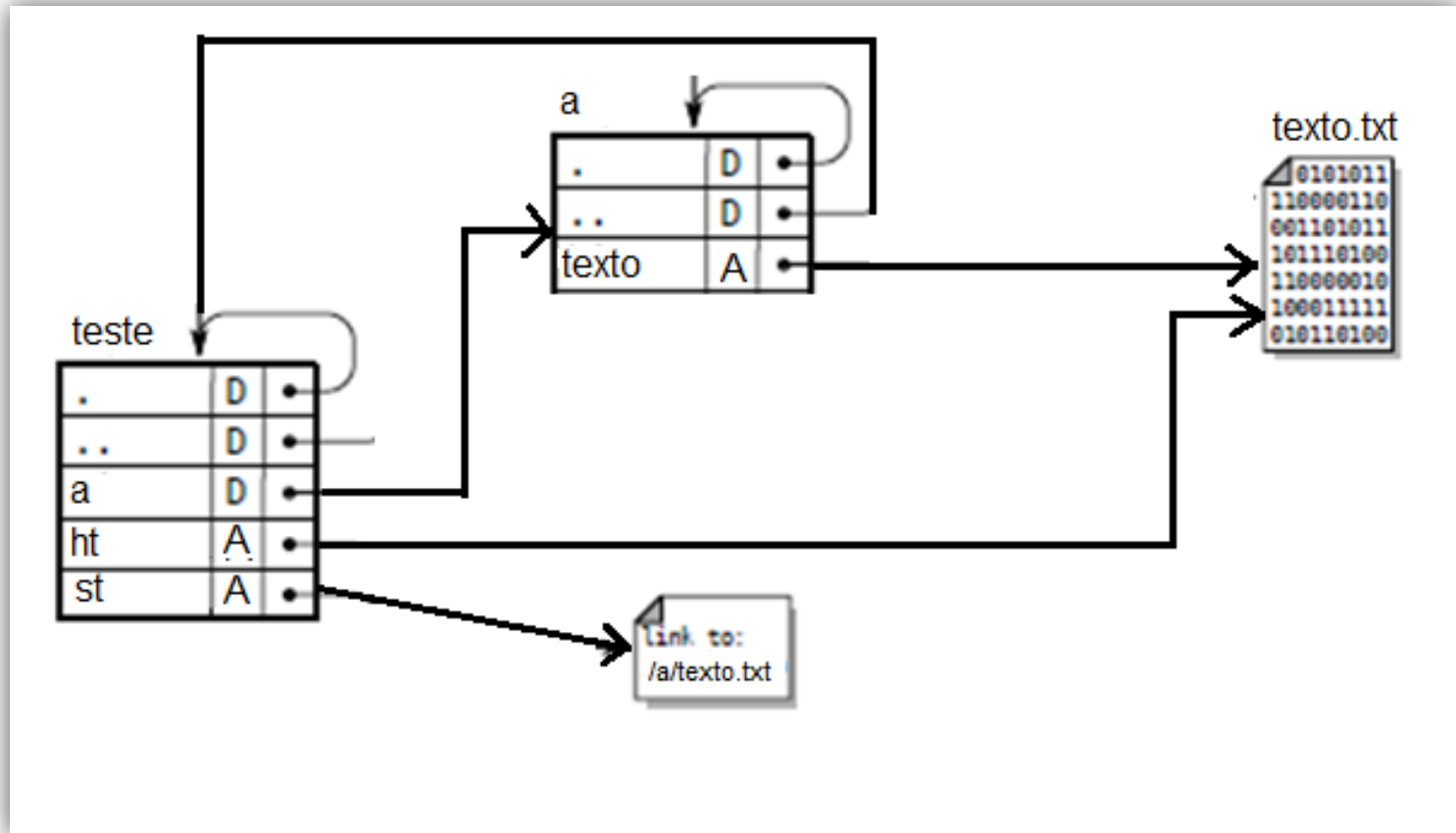
- Referência interna ao sistema de arquivo.
- Alvo tem um contador de atalhos físicos.
- Alvo só é apagado quando contador for zero.
- Alvo tem que estar no mesmo volume.

```
ln -n arquivo link_arquivo
```

Atalhos Simbólicos x Físicos



Link no WINDOWS



Montagem de volumes

Processo composto por:

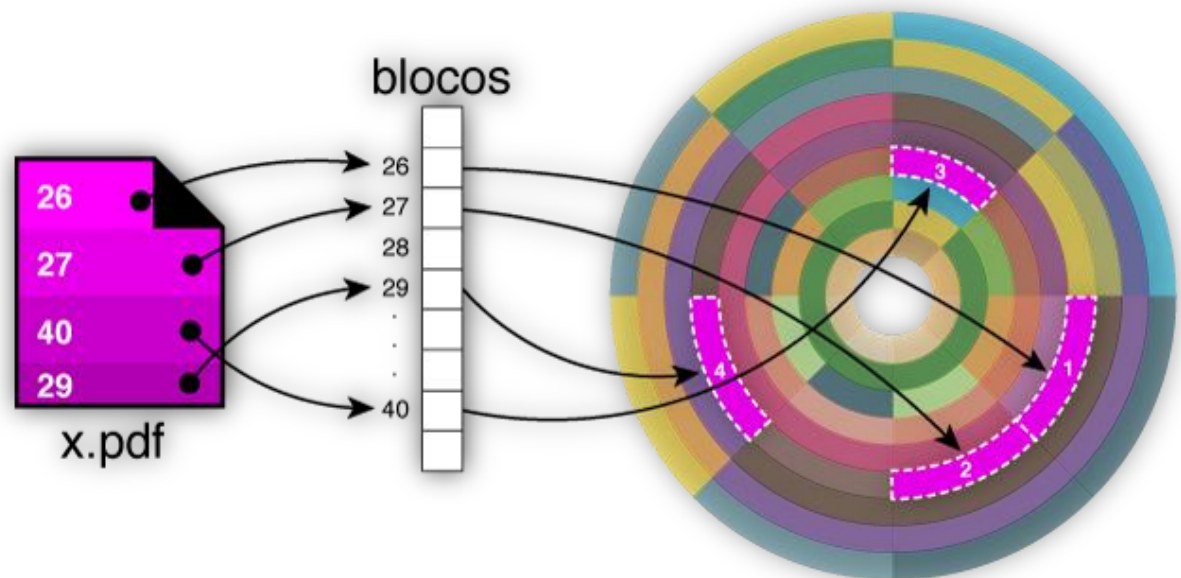
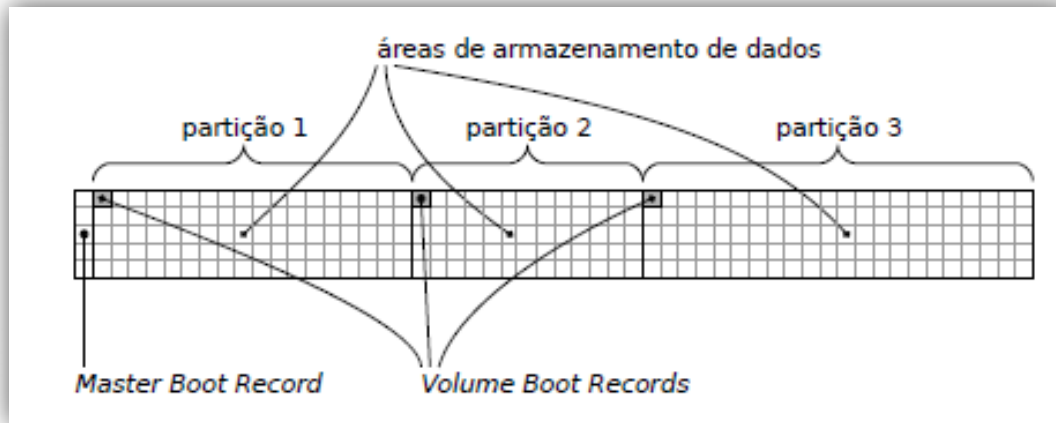
- Leitura do bloco de inicialização
- Criar estruturas em memória que represente esse volume no núcleo do S.O.

Ponto de montagem:

- Local para ser acessado pelos processos e usuários

BLOCOS

São as unidades de armazenamento no disco (Também conhecidos como setores)



Atividades

Linux (Dê exemplos):

- 1) Qual é o Comando para Listar as Partições do Disco?
- 2) Qual é o comando utilizado para verificar os pontos de montagens ativos?
- 3) Qual é o comando utilizado para listar os Diretórios?
- 4) Como é possível verificar os sockets ativos?
- 5) Explique as funções dos Diretórios do Linux.
(<https://debian-handbook.info/browse/pt-BR/stable/sect.filesystem-hierarchy.html>)
- 6) Qual é o comando utilizado para visualizar o Magic Number?
- 7) Qual é o comando utilizado para visualizar os inodes de um arquivo?
- 8) Qual é o comando utilizado para verificar o espaço livre das partições?
- 9) Qual é o comando utilizado para verificar o espaço utilizado pelos arquivos?
- 9) Qual é o comando utilizado para localizar um arquivo?
- 10) Qual é o comando utilizado para localizar um executável?