

# **Sistemas Operacionais**

## **Gerência de Arquivos**

Prof. José Paulo G. de Oliveira  
Eng. da Computação, UPE  
[jpgo@ecomp.poli.br](mailto:jpgo@ecomp.poli.br)

# Conteúdo

---

- Organização de volumes
  - Blocos de Bytes
- Implementação de diretórios
- Referência a arquivos
- Atalhos
- Montagem de volumes

# Conteúdo

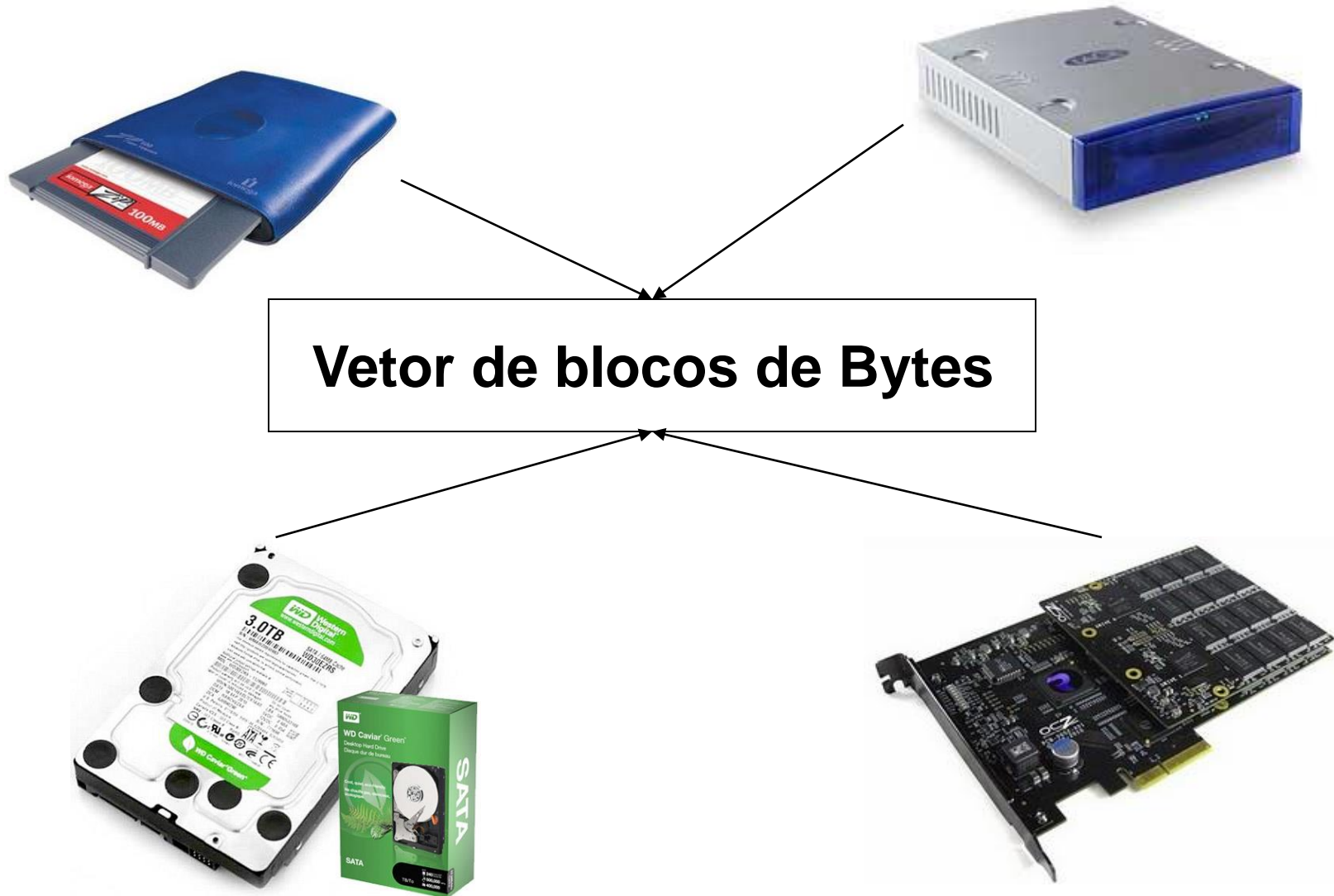
---

- **Organização de volumes**
  - Blocos de Bytes
- Implementação de diretórios
- Referência a arquivos
- Atalhos
- Montagem de volumes

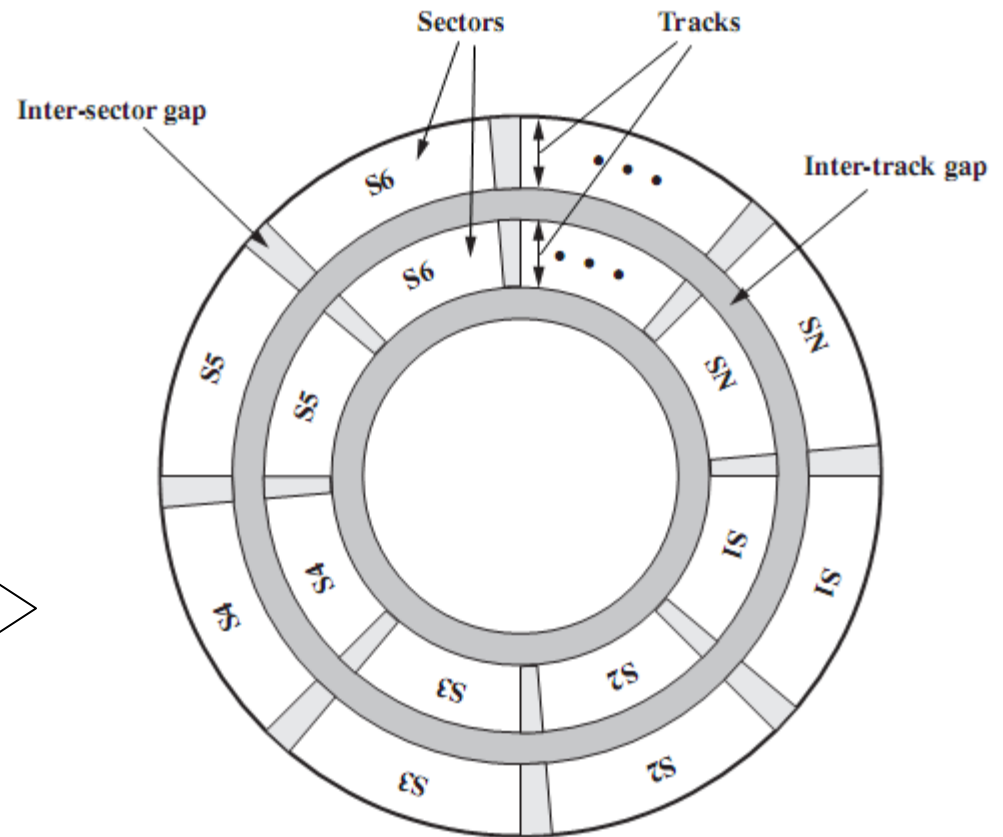
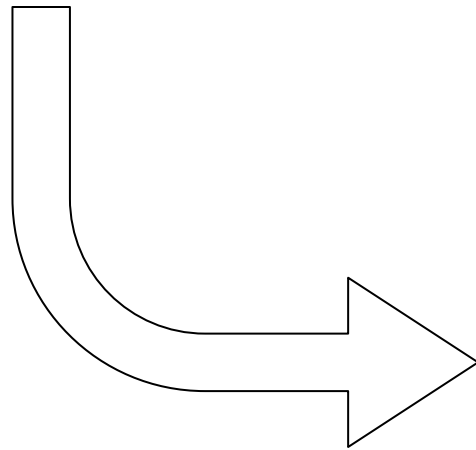
A decorative graphic on the left side of the slide. It consists of a vertical light blue bar, a vertical orange bar, and a grey rectangle. A horizontal dark blue bar extends from the grey rectangle across the top of the slide.

# **Organização de Volumes**

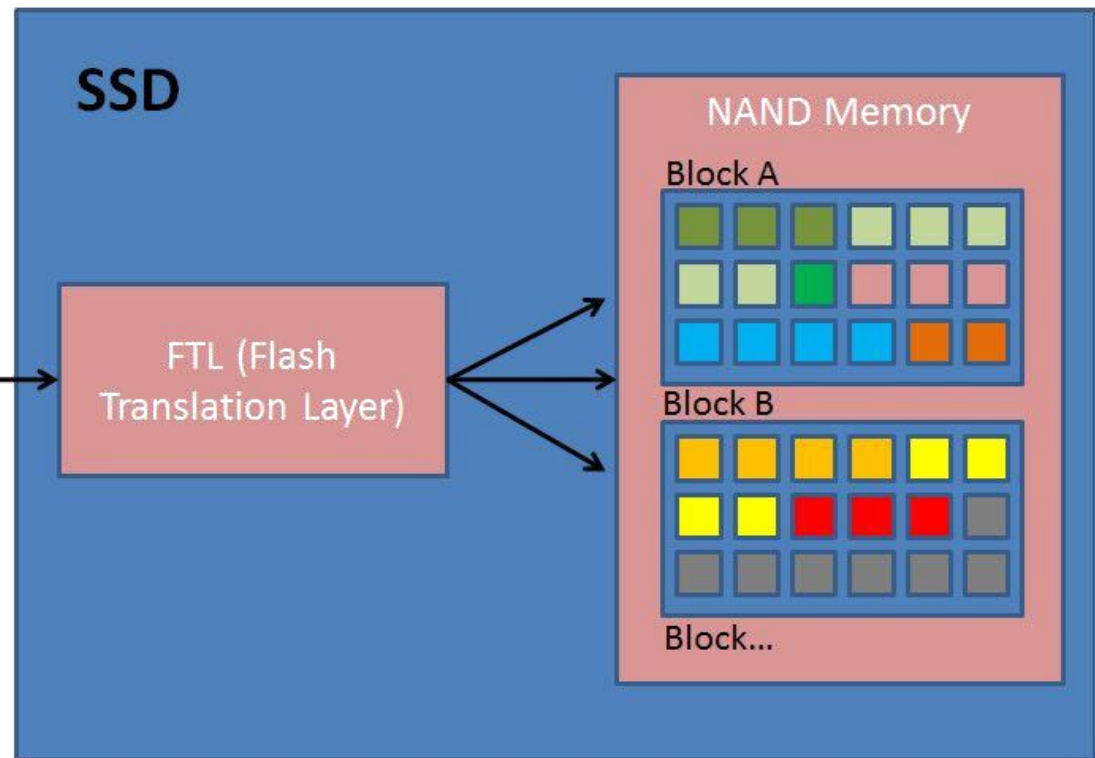
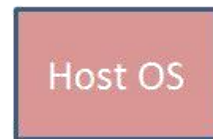
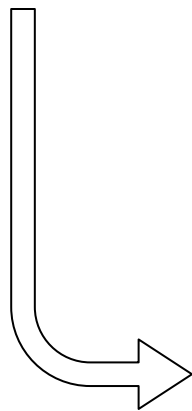
# Organização de Volumes



# Organização de Volumes



# Organização de Volumes



# Blocos de Bytes

---

- Também denominados **setores**
  - Tamanho fixo → entre 512 e 4.096 bytes
- Numerados sequencialmente
- **Leitura e Escrita** de dados feitas bloco a bloco
  - SSD → página
- Tamanho do bloco é importante
  - Gerência, desempenho e eficiência



A decorative graphic on the left side of the slide, consisting of a vertical light blue bar, a grey rectangle, and an orange vertical bar.

# **Partições e Volumes**

# Particionamento

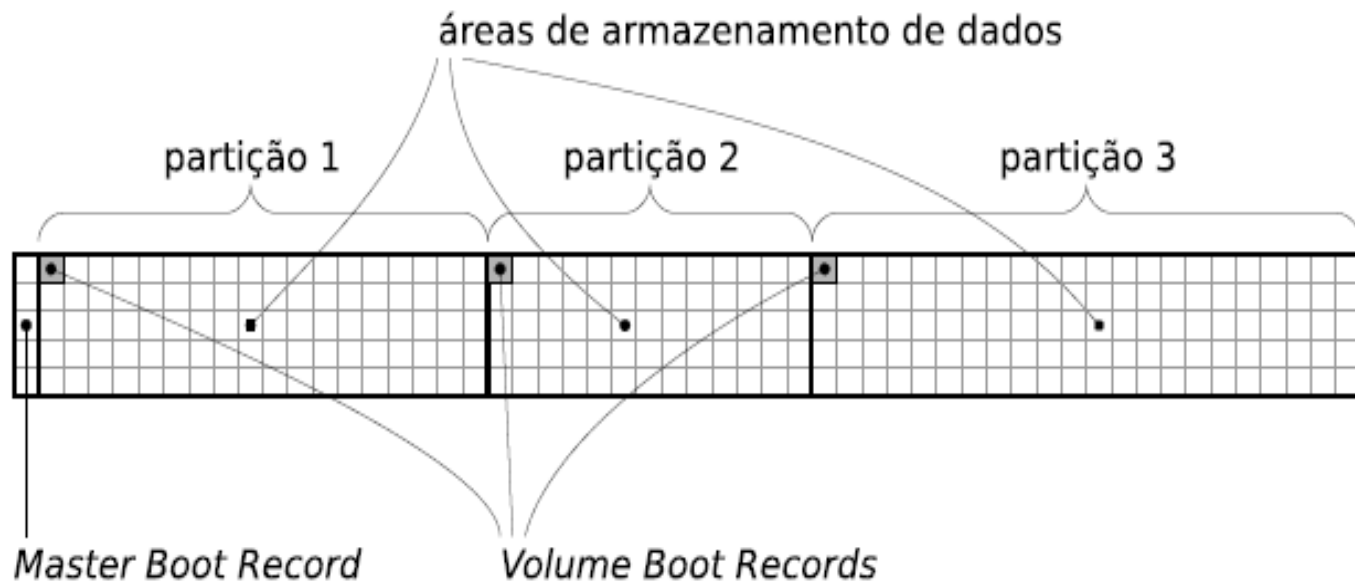
---

- Divisão do dispositivo físico em espaços lógicos independentes
- SO trata cada partição como um dispositivo lógico independente (volume)
- Particionamento depende do SO e da plataforma
  - Windows/intel
  - Mac
  - Unix
- Fisicamente: conjunto de setores contíguos

# Partições e Volumes

O espaço de armazenamento de cada dispositivo é dividido:

- Área inicial de configuração
- Uma ou mais *partições*



# HDs

Master Boot Record (MBR)

Boot Loader Code

Partition 1 Entry

Partition 2 Entry

Partition 3 Entry

Partition 4 Entry

Partition Table Entry

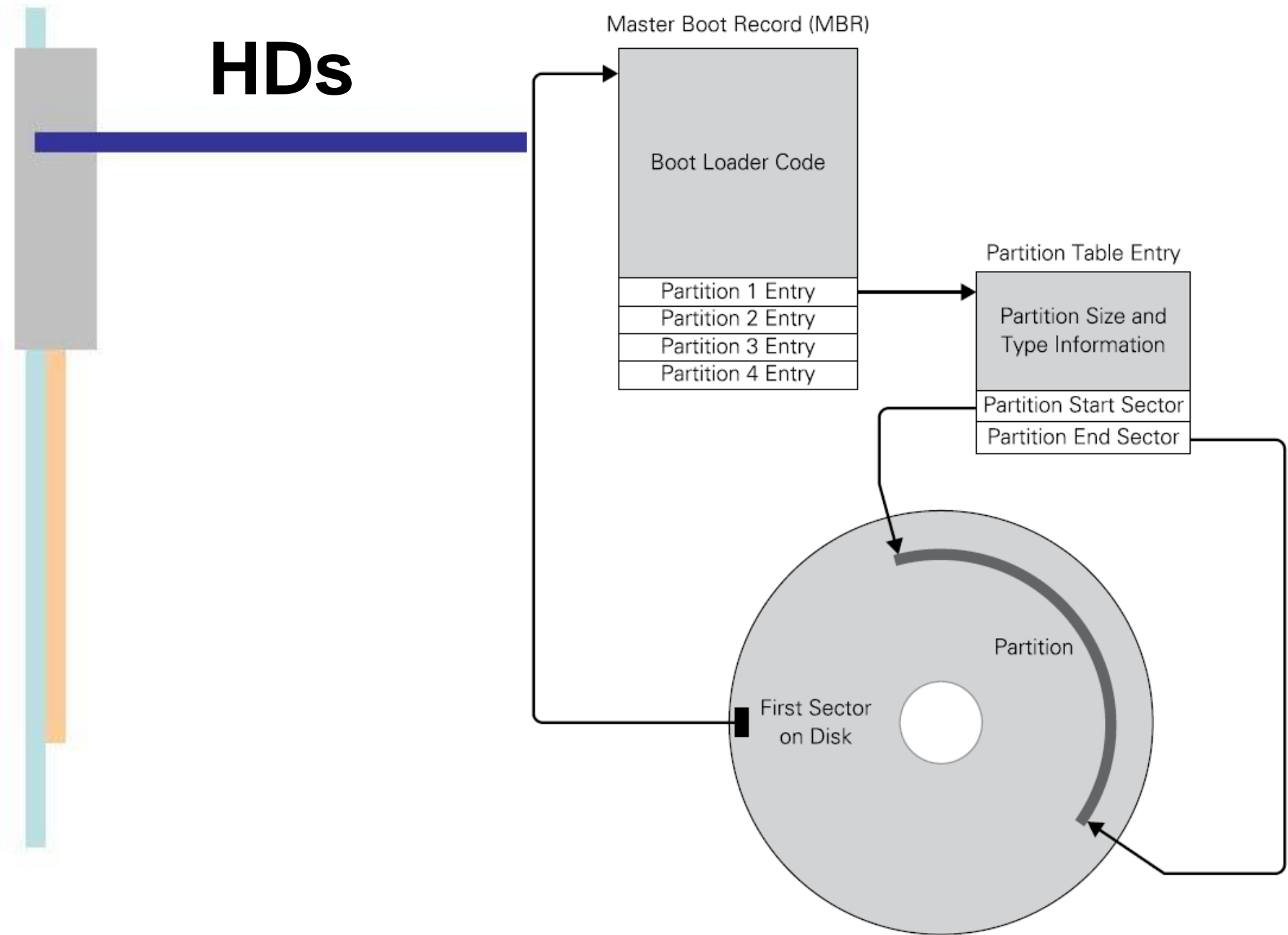
Partition Size and  
Type Information

Partition Start Sector

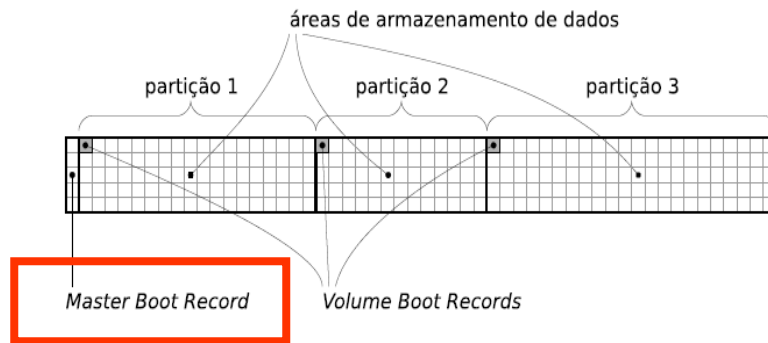
Partition End Sector

Partition

First Sector  
on Disk



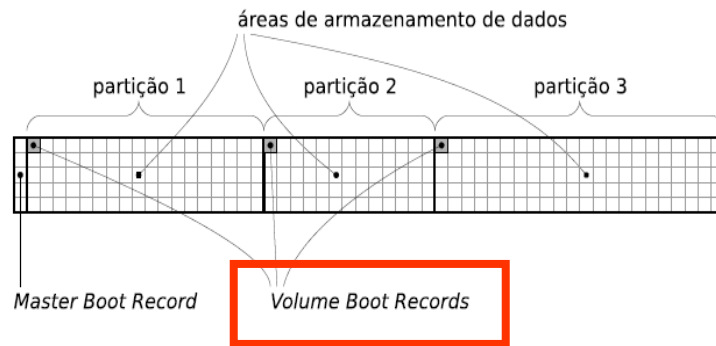
# Partições e Volumes



## MBR - Contém:

- *Tabela de partições* com informações sobre o particionamento do dispositivo
- Pequeno código executável, usado no processo de inicialização do sistema operacional

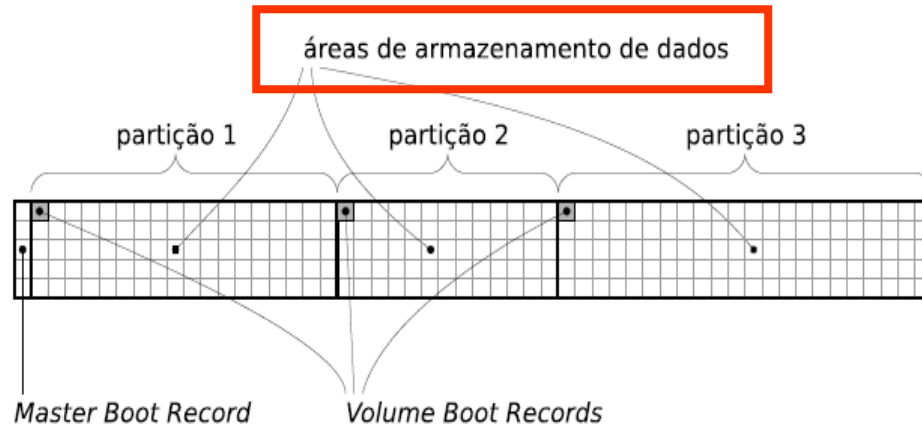
# Partições e Volumes



VBR - Utilizado para:

- Descrição do conteúdo da partição
- Armazenar código de lançamento do sistema operacional (partição *inicializável*)

# Partições e Volumes



O restante dos blocos da partição está disponível para o **armazenamento de arquivos**.

# Limitações do MBR

---

- Baseado em tecnologia antiga
  - Desde o MS-DOS
- MBR está em uma única posição no disco
  - Se danificar, perde acesso à informação
- Limite de 2 TB
- Limite de 4 partições primárias
- Solução: GUID partition tables (GPTs)
  - GUID: Globally Unique Identifier



# Conteúdo

---

- Organização de volumes
  - Blocos de Bytes
- **Implementação de diretórios**
- Referência a arquivos
- Atalhos
- Montagem de volumes

# Diretórios

---

- Quantidade de **arquivos** em um sistema atual pode chegar facilmente a **milhões** em um computador pessoal
- Para permitir a organização de arquivos dentro de uma partição são usados **diretórios**
- Cada espaço de armazenamento possui ao menos um diretório principal, denominado **diretório raiz**

# Diretórios

---

## Definição:

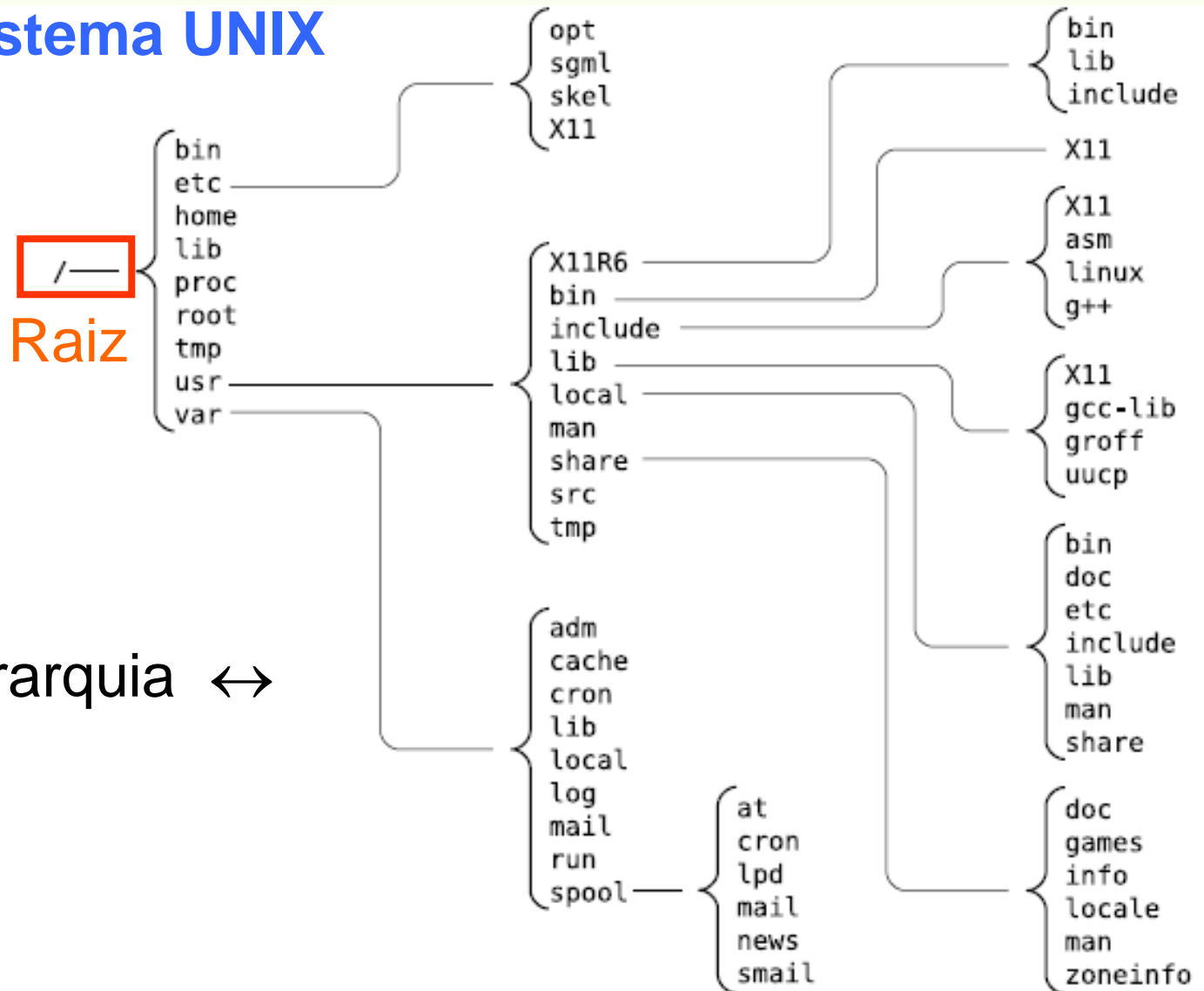
Um **diretório** (ou *pasta*, ou *folder*) representa um **contêiner de informações**. Pode conter:

- Arquivos
- Outros diretórios

Têm **nome e atributos** que são usados na **localização e acesso** aos arquivos neles contidos

# Estrutura de Diretórios

## Ex.: Sistema UNIX



Hierarquia ↔

# Implementação de Diretórios

---

Um diretório é implementado como um **arquivo estruturado**, cujo conteúdo é uma relação de **entradas**. Os tipos de **entradas** são:

- Arquivos normais, diretórios, *atalhos*
- Arquivos especiais

# Implementação de Diretórios

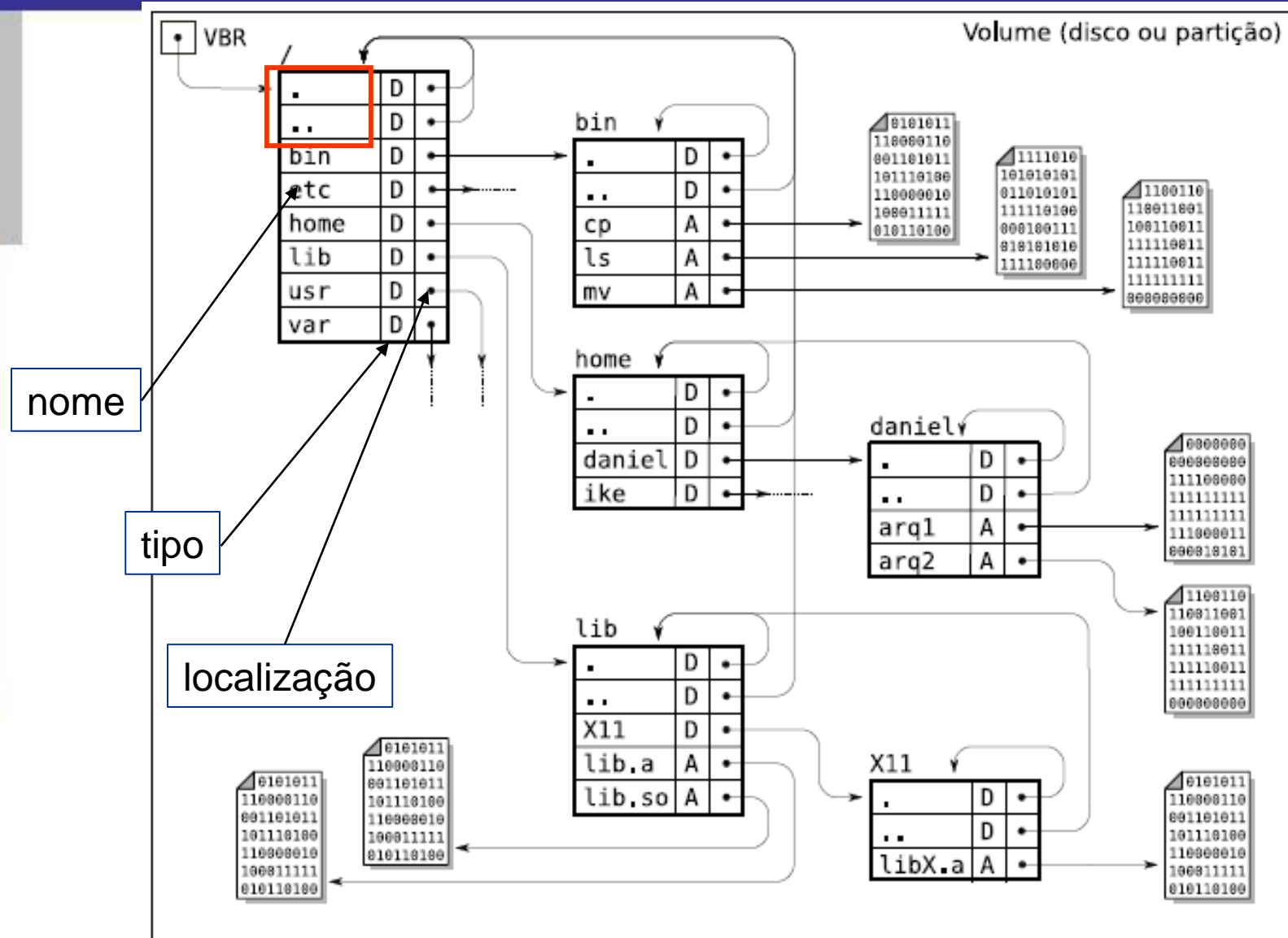
Um diretório é implementado como um **arquivo estruturado**, cujo conteúdo é uma relação de **entradas**. Os tipos de **entradas** são:

- Arquivos normais, diretórios, *atalhos*
- Arquivos especiais

Entradas contêm o **nome** do arquivo ou diretório, **tipo** e **localização** física no volume

→ Um diretório **não** contém fisicamente os arquivos e subdiretórios, apenas os **relaciona**

# Implementação de Diretórios



# Conteúdo

---

- Organização de volumes
  - Blocos de Bytes
- Implementação de diretórios
- **Referência a arquivos**
- Atalhos
- Montagem de volumes



# Caminhos de Acesso

---

- A posição de um arquivo dentro do sistema de arquivos é chamada de ***caminho de acesso*** ao arquivo

# Caminhos de Acesso

---

- A posição de um arquivo dentro do sistema de arquivos é chamada de ***caminho de acesso*** ao arquivo
- Composto por uma **sequência** de nomes de diretórios que levam até ele separados por um **caractere específico**:
  - \Windows\system32\ole32.dll
  - /usr/bin/bash

# Diretórios de Trabalho

## *(de um processo)*

---

### Definição:

A maioria dos sistemas implementa o conceito de **diretório de trabalho** de um processo

Cada **novo** processo recebe um **diretório de trabalho** para criar novos arquivos ou abrir arquivos existentes

# Referência a Arquivos



---

- Referência direta
- Referência absoluta
- Referência relativa

# Referência a Arquivos

---

## Referência direta:

- Somente o **nome** do arquivo é informado
- Considera-se que o arquivo está (ou será criado) no **diretório de trabalho** do processo

`prova1.doc  
materiais.pdf  
uma-bela-foto.jpg`

# Referência a Arquivos

## Referência absoluta:

- O caminho de acesso ao arquivo é indicado a partir do **diretório raiz** do sistema de arquivos e **não depende do diretório de trabalho** do processo

`\Windows\system32\drivers\etc\hosts.lm`

`/usr/local/share/fortunes/brasil.dat`

`\Documents and Settings\Carlos Maziero\..\All Users\notas.xls`

`/home/maziero/bin/scripts/../../docs/proj1.pdf`

# Referência a Arquivos

## Referência relativa:

- O caminho de acesso ao arquivo tem como **início o diretório de trabalho** do processo e indica subdiretórios ou diretórios anteriores através de referências “..”

```
imagens\satelite\brasil\geral.jpg
```

```
..\users\maziero\documentos\prova-2.doc
```

```
public_html/static/fotografias/rennes.jpg
```

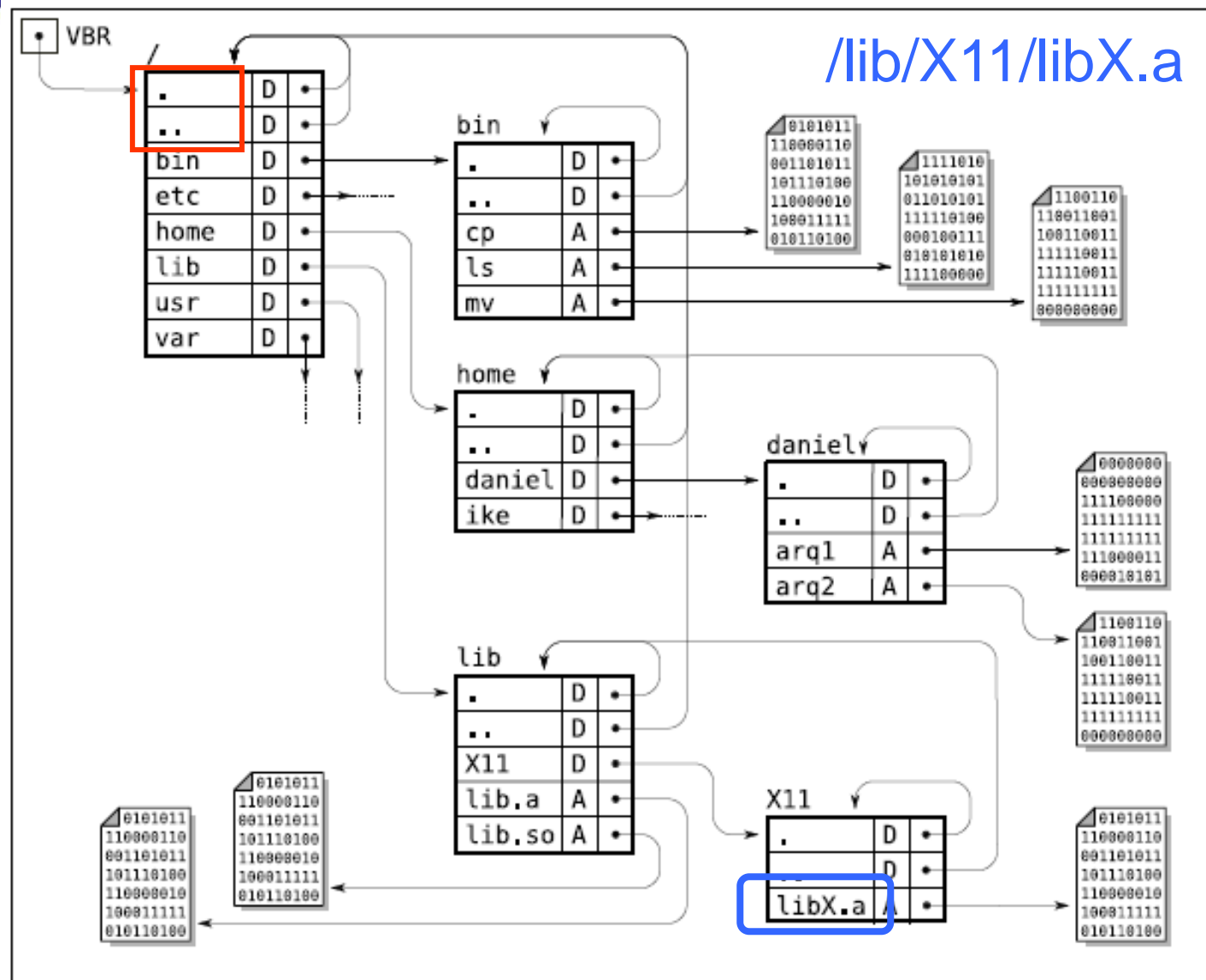
```
../.../share/icons/128x128/calculator.svg
```



# **Busca de Arquivo no Disco**



# File Lookup



1. Acessar o disco para ler o VBR (*Volume Boot Record*) do volume
2. Nos dados lidos, descobrir onde se encontra o diretório raiz (/) daquele sistema de arquivos
3. Acessar o disco para ler o diretório raiz
4. Nos dados lidos, descobrir onde se encontra o diretório `lib`
5. Acessar o disco para ler o diretório `lib`
6. Nos dados lidos, descobrir onde se encontra o diretório `X11`
7. Acessar o disco para ler o diretório `X11`
8. Nos dados lidos, descobrir onde se encontra o arquivo `libX11.a`
9. Acessar o disco para ler o bloco de controle do arquivo `libX11.a`, que contém seus atributos
10. Criar as estruturas em memória que representam o arquivo aberto
11. Retornar uma referência ao arquivo para o processo solicitante

# Conteúdo

---

- Organização de volumes
  - Blocos de Bytes
- Implementação de diretórios
- Referência a arquivos
- **Atalhos**
- Montagem de volumes

# Atalhos

---

**Definição:** Referências indiretas a arquivos ou diretórios

Pode ser necessário ter um mesmo arquivo ou diretório replicado

# Atalhos

**Definição:** Referências indiretas a arquivos ou diretórios

Pode ser necessário ter um mesmo arquivo ou diretório replicado. Mais **econômico**:

- Armazenar apenas **uma instância** dos dados do arquivo no sistema de arquivos e
- Criar **referências indiretas** (ponteiros) para essa instância

# Atalhos

**Definição:** Referências indiretas a arquivos ou diretórios

```
~> ls -l /usr/lib/

...
lrwxrwxrwx 1 root root libcrypt.so -> /lib/x86_64-linux-gnu/libcrypt.so.1
lrwxrwxrwx 1 root root libcryptui.so -> libcryptui.so.0.0.0
lrwxrwxrwx 1 root root libcryptui.so.0 -> libcryptui.so.0.0.0
-rw-r--r-- 1 root root libcryptui.so.0.0.0
...
```

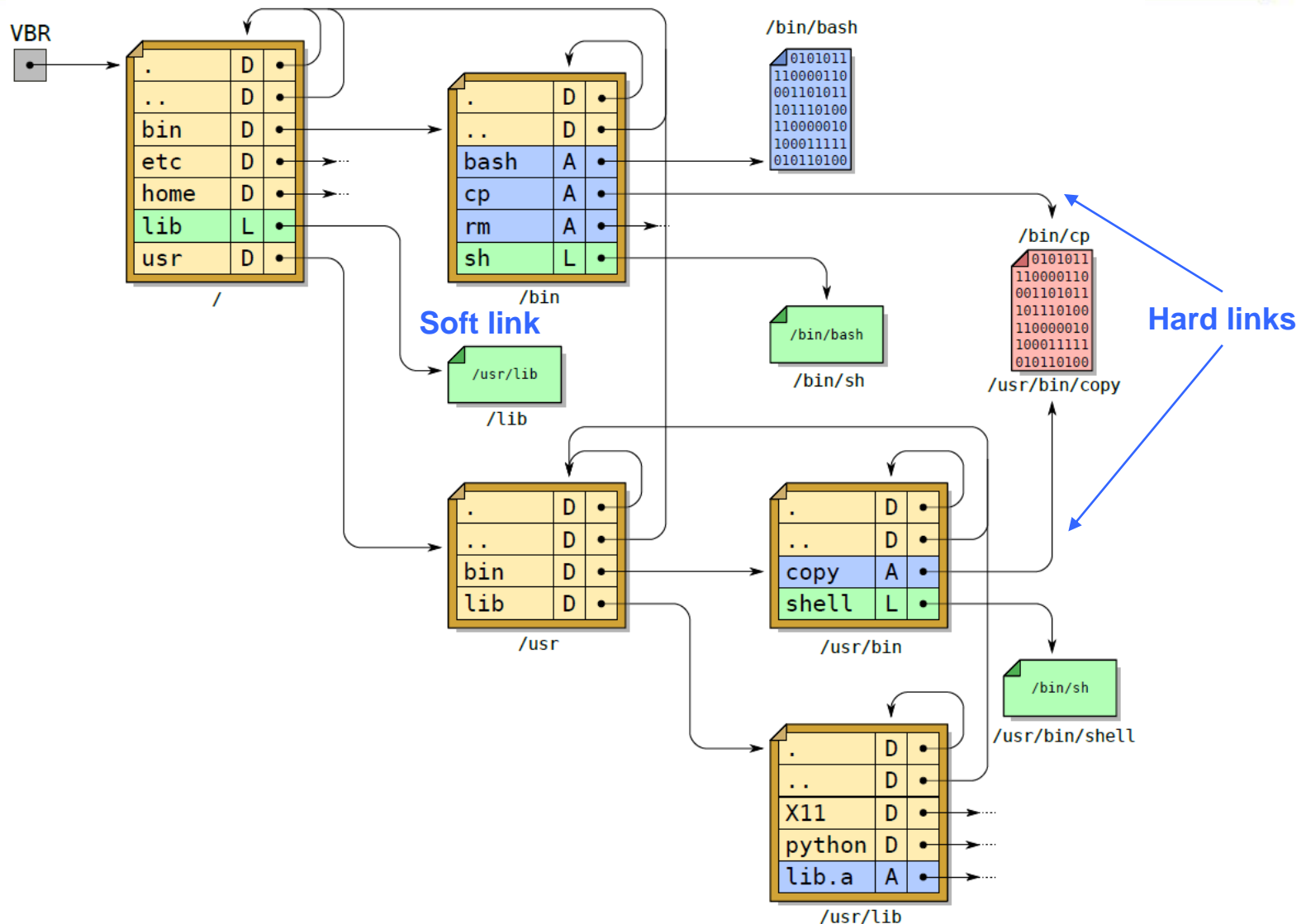
# Atalhos - Tipos

---

**Atalhos simbólicos** (*soft links*): cada “cópia” do arquivo original é na verdade um **pequeno arquivo de texto** contendo uma *string* com o caminho até o arquivo original

**Atalhos físicos** (*hard links*): várias referências do arquivo no sistema de arquivos apontam para a mesma **localização do dispositivo** físico onde o conteúdo do arquivo **está de fato** armazenado

# Atalhos - Tipos





# Atalhos - Tipos

## Atalhos simbólicos

- Possuem *i-nodes*\* e permissões diferentes do arquivo original
- Permissões não são alteradas, caso haja alteração no arquivo original
- Contêm apenas link para arquivo original, não o conteúdo

\*Abordado mais adiante

# Atalhos - Tipos

---

## Atalhos físicos

- Possuem *i-nodes* e permissões idênticos ao do arquivo original
- Permissões não são alteradas, caso haja alteração no arquivo original
- Possuem o conteúdo do arquivo original
  - Original removido conteúdo ainda disponível no atalho

# Conteúdo

---

- Organização de volumes
  - Blocos de Bytes
- Implementação de diretórios
- Referência a arquivos
- Atalhos
- **Montagem de volumes**

# Montagem de Volumes

---

Para o sistema operacional acessar o **sistema de arquivos** ele deve:

1. Ler os dados presentes em seu **bloco de inicialização**
2. Criar as **estruturas em memória** que representam esse volume dentro do núcleo
3. Definir um **identificador** para o volume

Esse procedimento é denominado **montagem do volume**

# Desmontagem de Volumes

---

## Procedimento oposto

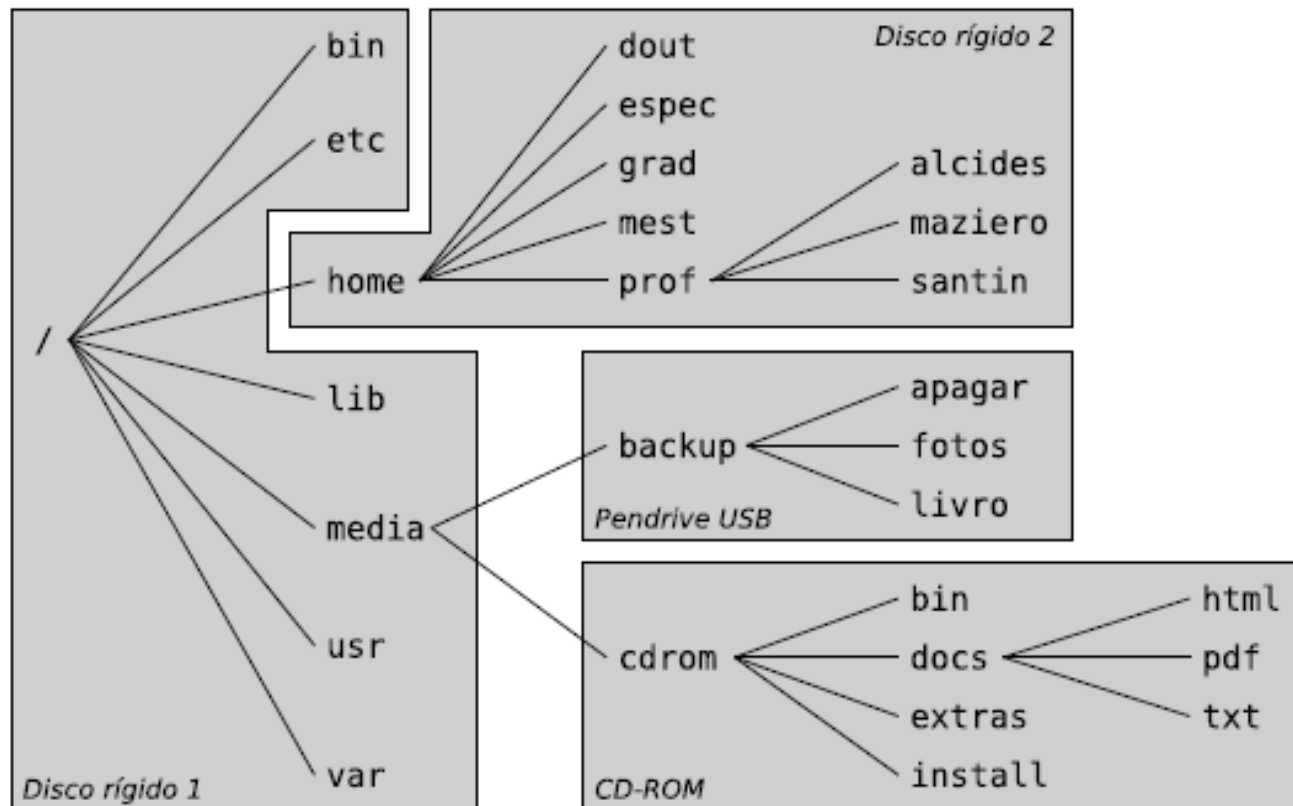
Consiste em:

1. Fechar todos os arquivos abertos no volume
2. Remover as estruturas de memória usadas para gerenciá-lo

# Montagem de Volumes

## Ex.: Montagem no UNIX

Montagem de volumes: posições dentro da **árvore** principal do sistema de arquivos



# Montagem de Volumes

---

Nos sistemas **DOS** e **Windows** é comum definir cada **volume montado** como um disco lógico distinto chamado simplesmente de **disco** ou **drive** e identificado por uma letra (**“A:”, “C:”, “D:”, etc.**)