

# Sistemas Operacionais

## Gestão de arquivos - diretórios e atalhos

Prof. Carlos Maziero

DInf UFPR, Curitiba PR

Agosto de 2020

# Conteúdo

- 1 O conceito de diretório
- 2 Caminhos de acesso
- 3 Implementação de diretórios
- 4 Atalhos
- 5 Resolução de nomes de caminho

# Diretório

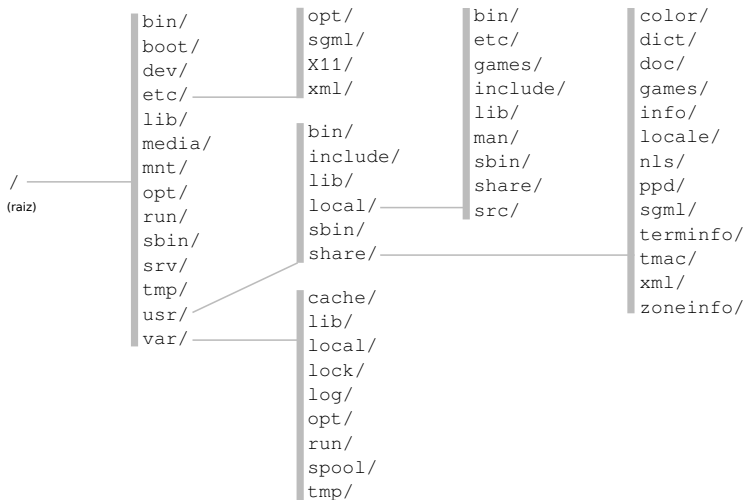
Conceito:

- Elemento estruturante do sistema de arquivos.
- “Contém” arquivos e outros diretórios.
- Também chamado *pasta* ou *folder*.

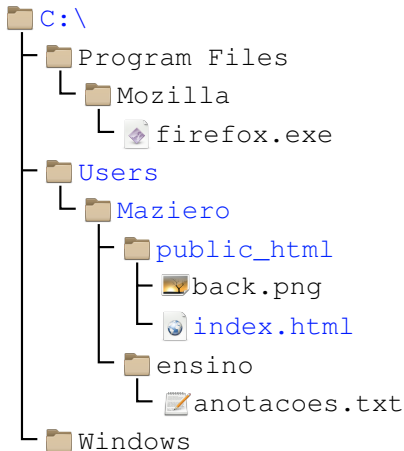
Diretórios organizam os volumes em árvores:

- UNIX: uma única árvore por sistema (/).
- Windows: uma árvore por dispositivo (C:, D:).

# Linux FHS - *Filesystem Hierarchy Standard*



# Caminhos de acesso



C:\Program Files\Mozilla\firefox.exe

C:\Users\Maziero\public\_html\index.html

C:\Users\Maziero\ensino\anotacoes.txt

C:\Windows

# Caminhos de acesso

Arquivos são referenciados por **nomes** e **caminhos**

- **Nome:** identifica o arquivo para o usuário.
- **Caminho:** trilha de diretórios que leva ao arquivo.
- **Separador:** / \ # : !
- **Referências especiais:** . ..

Formas de referenciar um arquivo:

- Referência direta
- Referência absoluta
- Referência relativa

# Caminhos de acesso

**Referência direta:** somente o nome do arquivo

```
1 materiais.pdf  
2 uma-bela-foto.jpg
```

**Referência absoluta:** caminho inicia no diretório raiz

```
1 \Windows\system32\drivers\etc\hosts.lm  
2 /home/maziero/bin/scripts/../../docs/proj1.pdf
```

**Referência relativa:** caminho inicia no diretório atual

```
1 imagens\satelite\brasil\geral.jpg  
2 ../../../../share/icons/128x128/calculator.svg
```

# Implementação de diretórios

Diretório = arquivo que contém uma **tabela de arquivos**.

Cada entrada dessa tabela contém:

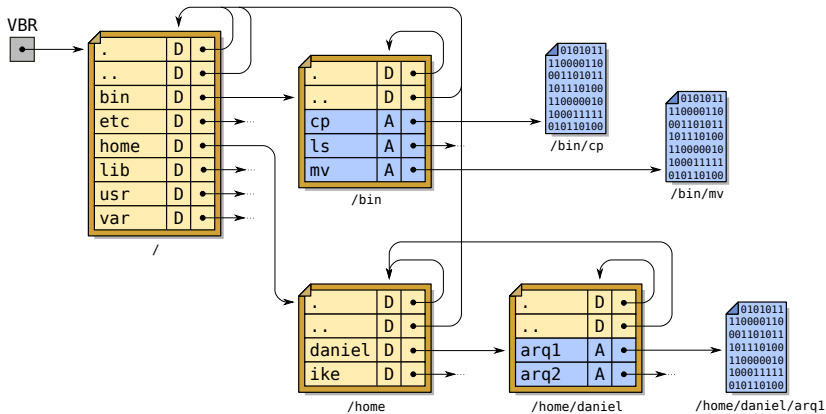
- Nome do arquivo
- Localização no disco
- Atributos (depende do *filesystem*)

Um diretório **não contém** arquivos, apenas os relaciona.

A localização do diretório raiz é indicada no *Volume Boot Record*.



# Implementação de diretórios



# Implementação de diretórios

Implementação da tabela de arquivos de um diretório:

- Lista linear (FAT, Ext2, Ext3)
- Árvore binária (NTFS, XFS, JFS, ReiserFS, BTRFS)
- Tabela de hash (ZFS)
- Árvore com hash (Ext3, Ext4)

# Atalhos

Atalhos: **referências indiretas** a arquivos ou diretórios.

Evita replicação desnecessária de arquivos.

Permite atribuir vários nomes/caminhos ao mesmo conteúdo.

```

1  $ ls -l /usr/lib/
2
3  ...
4  lrwxrwxrwx  1 root root libcrypt.so -> /lib/x86_64/libcrypt.so.1
5  lrwxrwxrwx  1 root root libcryptui.so -> libcryptui.so.0.0.0
6  lrwxrwxrwx  1 root root libcryptui.so.0 -> libcryptui.so.0.0.0
7  -rw-r--r--  1 root root libcryptui.so.0.0.0
8  ...
  
```

# Tipos de atalhos

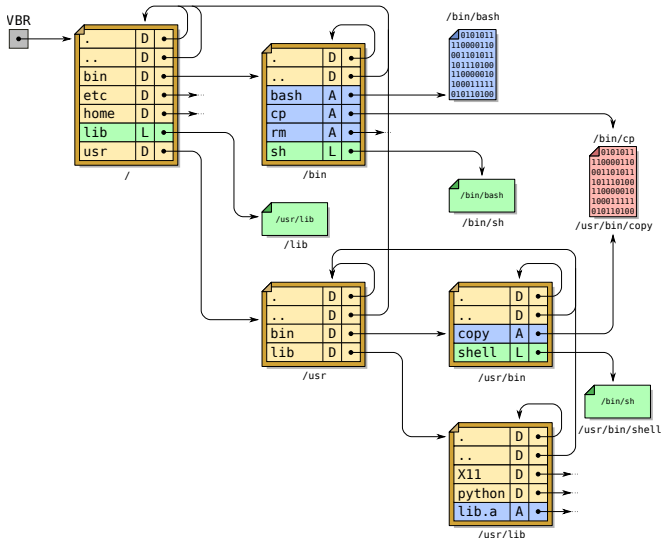
Atalho **simbólico** (*soft link*):

- Arquivo que contém uma string (caminho de acesso).
- Pode apontar para arquivo em outro sistema de arquivos.
- Pode apontar para arquivo inexistente (“broken link”).

Atalho **físico** (*hard link*):

- Aponta para uma área de disco.
- Só pode ser usado dentro do mesmo sistema de arquivos.
- Cada arquivo deve manter um contador de referências.

# Implementação de atalhos

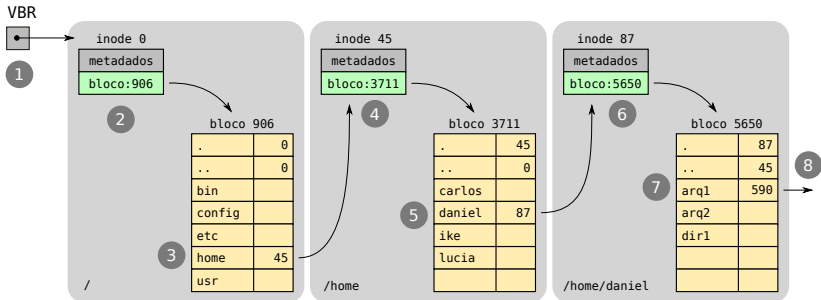


# Localização de um arquivo

Localizar arquivo `/home/daniel/arq1` (UNIX):

- 1 ler o VBR para obter localização do *inode* do diretório `/`
- 2 Ler o *inode* de `/` para:
  - Checar a permissão de acesso a `/`
  - Obter a localização do conteúdo de `/`
- 3 Ler o conteúdo de `/` e obter o *inode* de `/home`
- 4 Ler o *inode* de `/home` para:
  - Checar a permissão de acesso a `/home`
  - Obter a localização do conteúdo de `/home`
- 5 Idem para `/home` e `/home/daniel`

# Localização de um arquivo



# Caching de nomes

A resolução de um nome exige muitos acessos a disco:

```
1 /usr/share/texlive/texmf-dist/tex/generic/pgf/graphdrawing/  
2 lua/pgf/gd/force/jedi/algorithms/SimpleSpring.lua
```

Caminho com 15 níveis: 30 acessos a disco!

Muitas resoluções são repetidas (/usr/bin, /home)

**Cache de resolução de nomes:**

- Armazena pares [path : inode] recentes
- Acelera a localização de arquivos e diretórios
- Pode armazenar resultados negativos



# Caching de nomes

Cache de nomes: tabela *hash* com política LRU

Conteúdo do cache:

<b>caminho</b>	<b><i>i-node</i></b>
/home/daniel/arq1	590
/home/daniel	87
/home	45
/	0

Consultas:

```

/home/daniel/arq1 : inode 590
/home/daniel/arq2 : não
/home/daniel      : inode 87

/usr/bin/bash     : não
/usr/bin          : não
/usr              : não
/                 : inode 0
  
```