

Danton Cavalcanti Franco Junior  
falecom@dantonjr.com.br

# *Sistema de Arquivos*

- **Requisitos para armazenamento a longo prazo:**
  - Deve ser possível o armazenamento de grande quantidade de informação.
  - A informação deve sobreviver ao término do processo que estiver usando.
  - Vários processos devem ser capazes de manipular concorrentemente a informação.

# *Sistema de Arquivos*

## □ **Arquivos**

- São constituído de informações logicamente relacionadas (programas: instruções compreendidas pelo processador; ou dados: estruturado livremente).
- Podem ser armazenados pelo Sistema Operacional.
- Possuem identificação através de um nome.

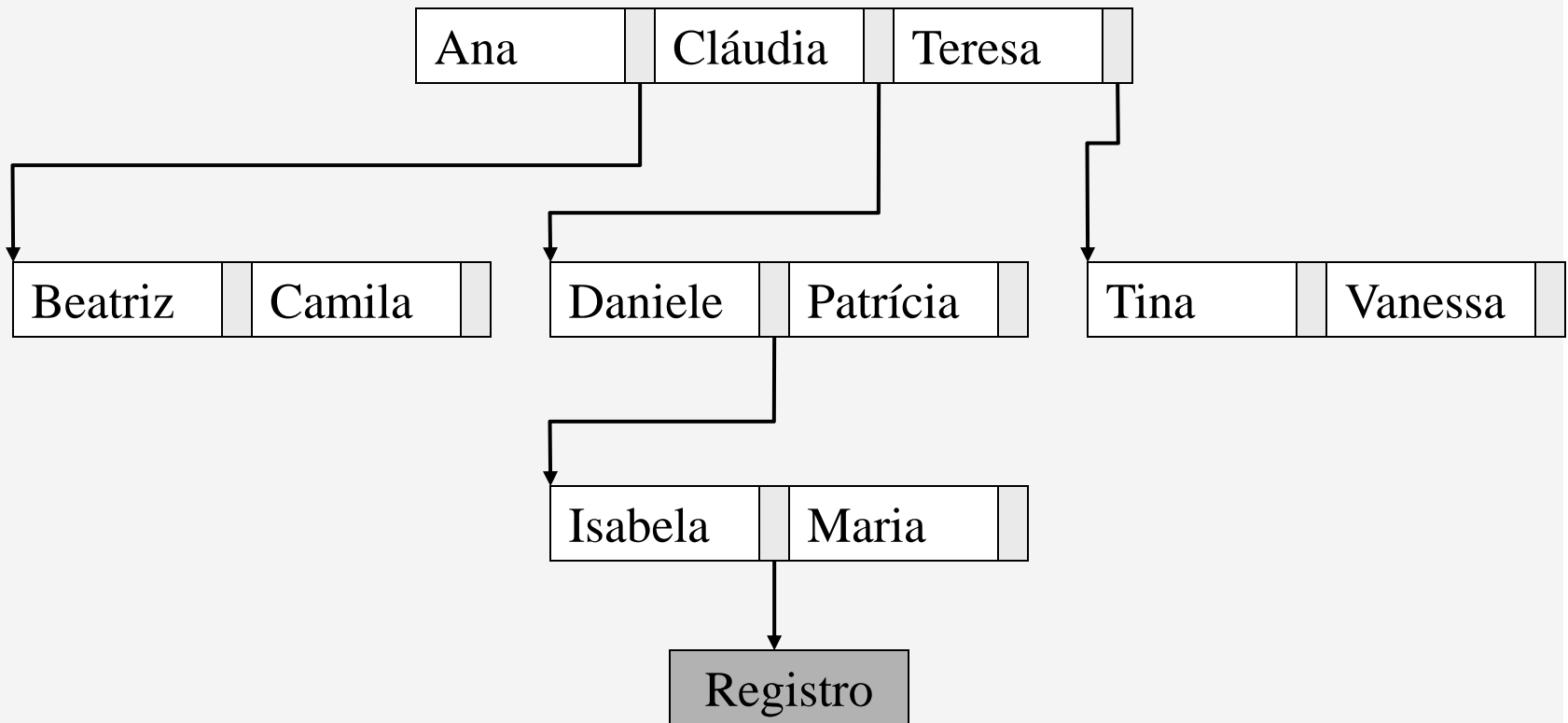
# *Sistema de Arquivos*

## □ **Organização de Arquivos**

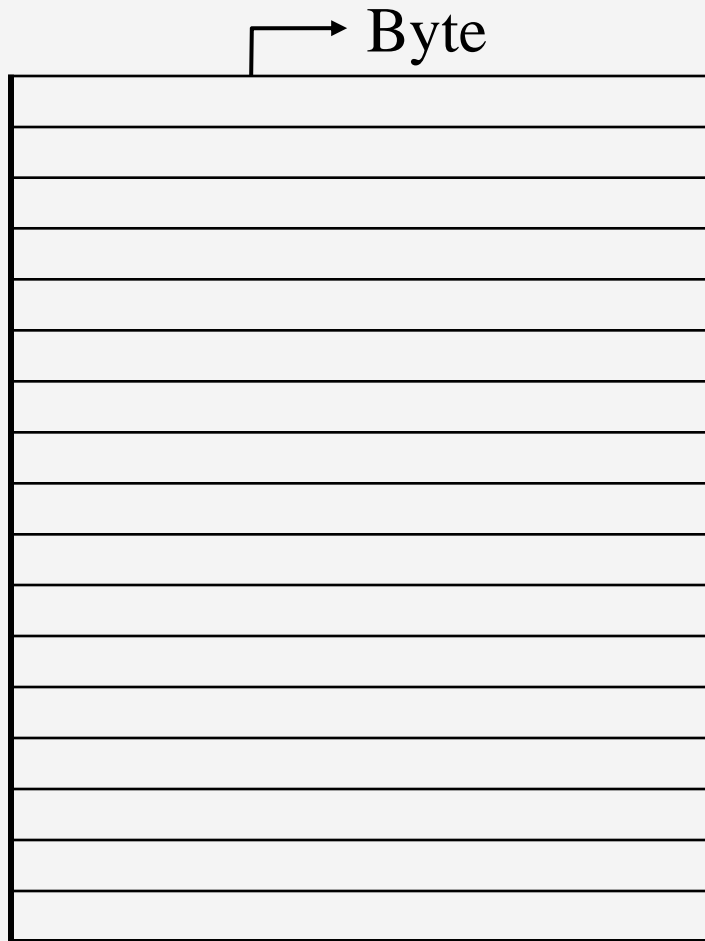
- Consiste no modo como os seus dados estão internamente armazenados
- A organização do arquivo é definida no momento da criação do mesmo.
- A forma mais simples de organização é através de uma sequência não estruturada de bytes (não impõe estrutura lógica para os dados – a aplicação define toda a organização) – Flexível.

# *Sistema de Arquivos*

Organização Indexada



# *Sistema de Arquivos*



Não-estruturada

# ***Métodos de Acesso***

- ❑ Arquivos armazenam informações.
- ❑ Quando utilizadas, estas informações devem ser acessadas e lidas na memória do computador.
- ❑ Há muitas formas de acesso às informações. Alguns sistemas fornecem apenas um tipo, já outros disponibilizam diferentes métodos, conforme a necessidade da aplicação.

# *Métodos de Acesso*

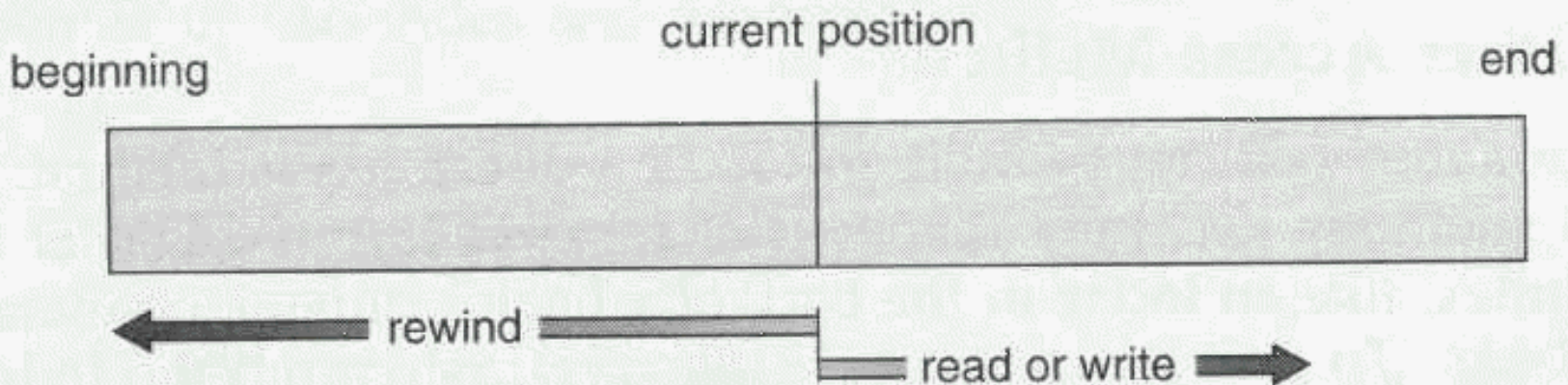
## □ *Acesso Seqüencial*

- É o método mais simples.
- Informação no arquivo é processada em seqüência, um registro após o outro (fitas).
- Leitura na ordem de gravação dos registros.



# *Métodos de Acesso*

## Acesso Seqüencial



Sequential-access file.

# ***Métodos de Acesso***

## **□ Acesso Direto ou Acesso Relativo**

- Um arquivo é criado com registros de tamanho fixos que permite aos programas ler e gravar registros rapidamente e sem uma sequência particular.
- O acesso direto assemelha-se ao acesso aos arquivos no disco.

# ***Métodos de Acesso***

## □ ***Acesso Direto ou Acesso Relativo***

- O arquivo é visto como uma seqüência numerada de blocos de registros.
- Permite ler e gravar blocos arbitrariamente.
- Pode-se ler o bloco 14, em seguida o 53 e então gravar o bloco 7.
- Não há restrição na ordem de leitura e gravação para este tipo de acesso.

# *Métodos de Acesso*

## Acesso Direto



O arquivo precisa ter registros de mesmo tamanho

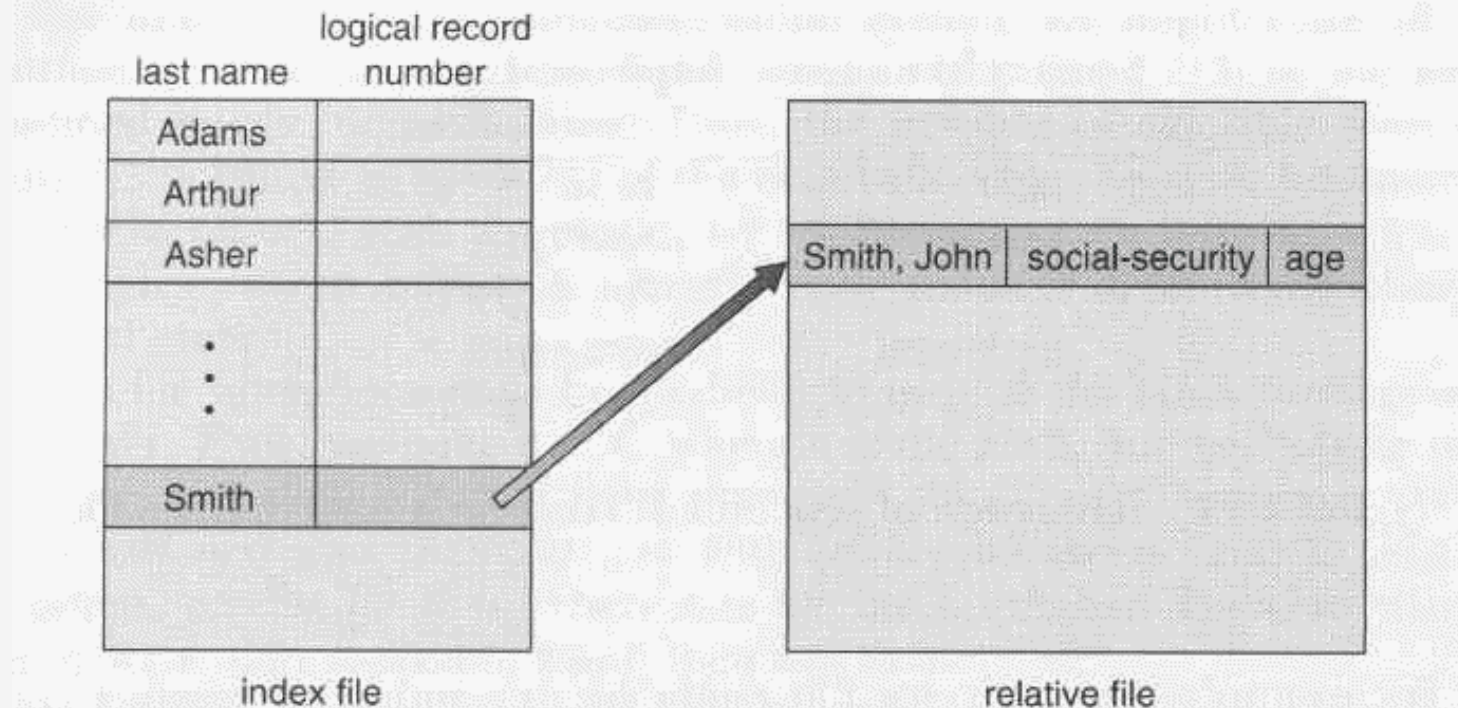
# ***Métodos de Acesso***

## □ ***Acesso por Chave ou Indexado***

- Estas outras formas geralmente envolvem a criação de um arquivo de índice para o arquivo de dados.
- O arquivo de índice possui apontadores para vários blocos.
- Para encontrar um arquivo, primeiro pesquisa-se o arquivo de índice e então usa o apontador para obter acesso direto ao arquivo.

# *Métodos de Acesso*

## Acesso Indexado

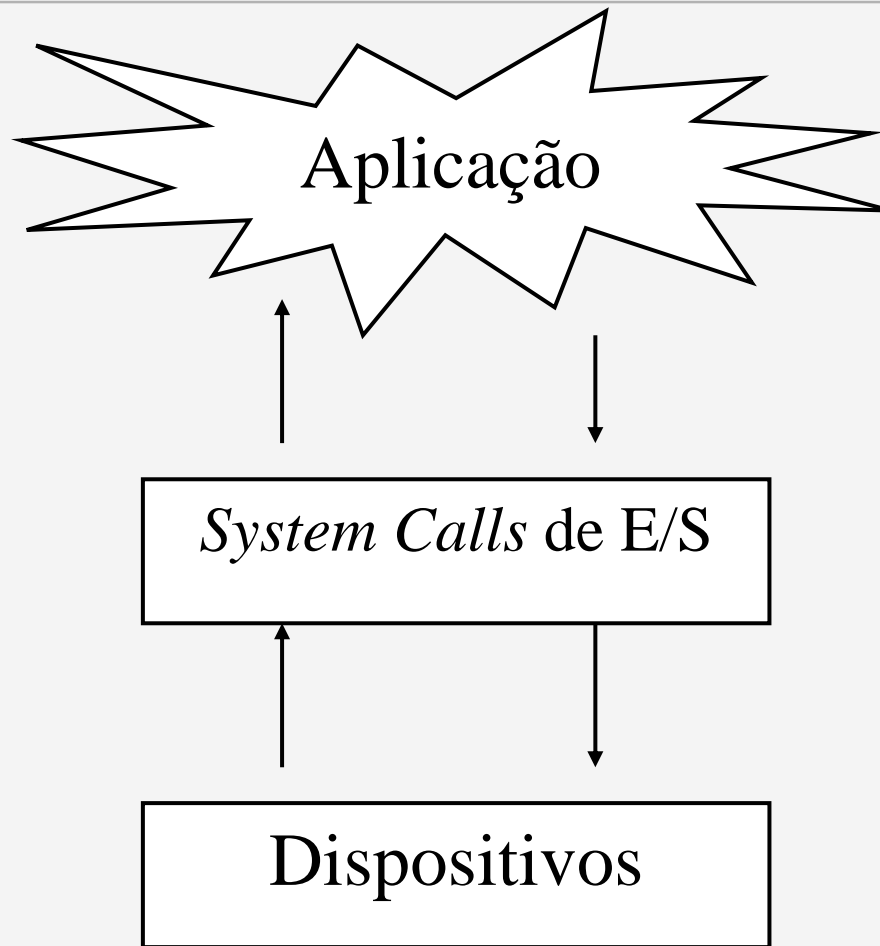


Example of index and relative files.

# *Operação de E/S*

- O sistema de arquivos oferece um conjunto de system calls que permite às aplicações realizar operações de I/O:
  - Tradução de nomes de endereços
  - Leitura e gravação de dados
  - Criação e eliminação de arquivos
- System calls oferece uma interface simples e uniforme entre aplicação e diversos dispositivos.

# *Operação de E/S*





# *Operação de E/S*

| <b>Comando</b> | <b>Descrição</b>                |
|----------------|---------------------------------|
| CREATE         | Criação de um arquivo           |
| OPEN           | Abertura de um arquivo          |
| READ           | Leitura de um arquivo           |
| WRITE          | Gravação de dados em um arquivo |
| CLOSE          | Fechamento de um arquivo        |
| RENAME         | Alteração do nome do arquivo    |
| DELETE         | Eliminação de um arquivo        |

# *Atributos de Arquivos*

- Cada arquivo possui informações de controle chamadas de atributos.
- Dependendo do sistema, os atributos variam, porém alguns são comuns a maioria dos SOs:
  - Tamanho
  - Proteção
  - UID do criador
  - Data de criação.

# *Atributos de Arquivos*

- Alguns atributos especificados na criação do arquivo não podem ser modificados em função de sua própria natureza:
  - Organização
  - Data/hora de criação.
- Outros alterados pelo próprio SO:
  - Tamanho
  - Data/hora do último acesso
  - Backup realizado.

# *Atributos de Arquivos*

- Existem também atributos do usuário:
  - Proteção do arquivo
  - Tamanho máximo
  - Senha de acesso.

# *Atributos de Arquivos*

| <b>Atributo</b> | <b>Descrição</b>                        |
|-----------------|---|
| TAMANHO         | Especifica o tamanho do arquivo         |
| PROTEÇÃO        | Código de proteção de acesso            |
| DONO            | Identifica o criador do arquivo         |
| CRIAÇÃO         | Data e hora da criação do arquivo       |
| BACKUP          | Data e hora do último backup realizado  |
| ORGANIZAÇÃO     | Indica a organização lógica do arquivo  |
| SENHA           | Senha necessária para acessar o arquivo |

# *Divisão do Disco*

- ❑ O sistema de arquivos dos computadores são extensos.
- ❑ Para gerenciar todos os arquivos há necessidade de organizá-los de alguma forma.
- ❑ Esta organização é feita, geralmente, de duas partes:
  - Partições
  - Diretórios

# *Partições*

## □ *Partições – minidisks*

- O sistema de arquivos é dividido em partições, também conhecidas como minidisks, volumes ou discos virtuais.
- Cada disco possui no mínimo uma partição, que é uma estrutura de baixo nível, em que arquivos e diretórios residem.

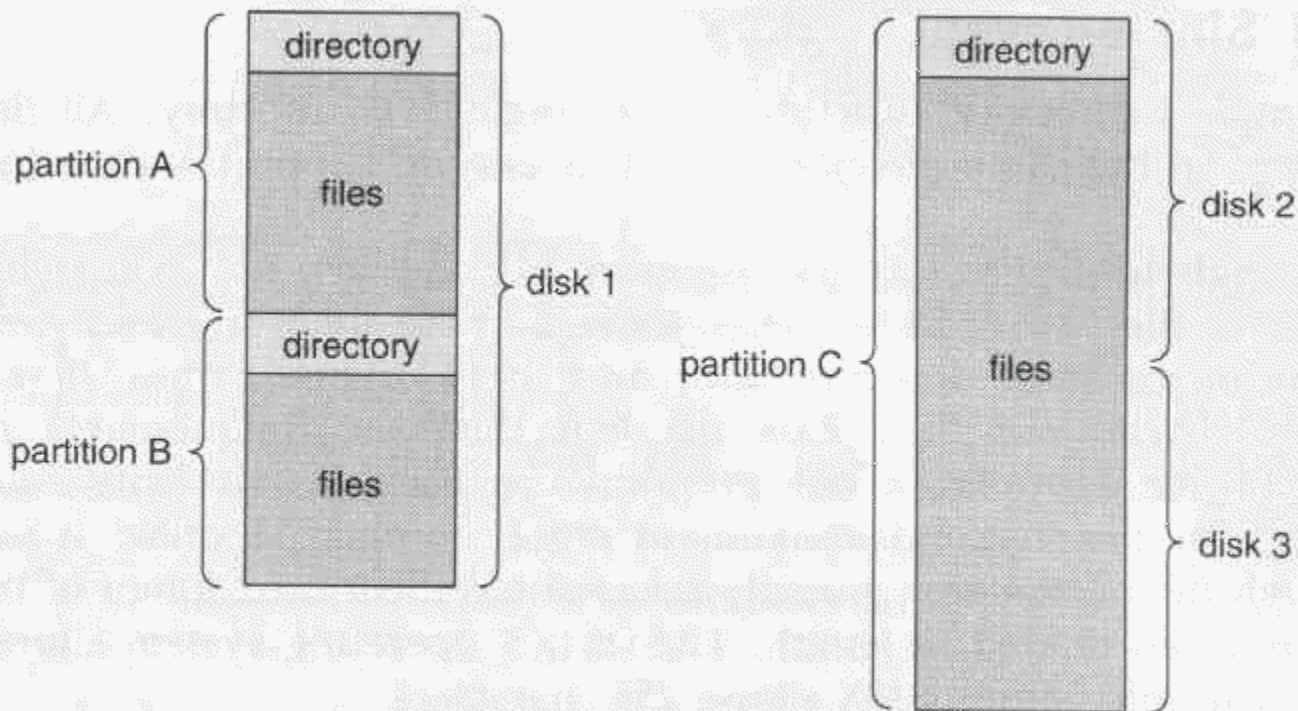
# *Partições*

## □ *Partições – minidisks*

- Partições podem ser usadas para separar áreas de um mesmo disco, sendo que estas áreas podem possuir sistemas de arquivos diferentes.
- Cada partição é tratada como um disco separado.



# Partições



A typical file-system organization.

# *Estrutura de Diretórios*

- *Tabela de conteúdo do volume ou diretório*
  - Cada partição possui informações sobre os arquivos armazenados nela.
  - O diretório mantém informações sobre os arquivos da partição específica - como o nome, localização, tamanho e tipo.
  - O diretório pode ser visto como uma tabela de símbolos que traduz o nome do arquivo em posição para o mesmo ser manipulado.

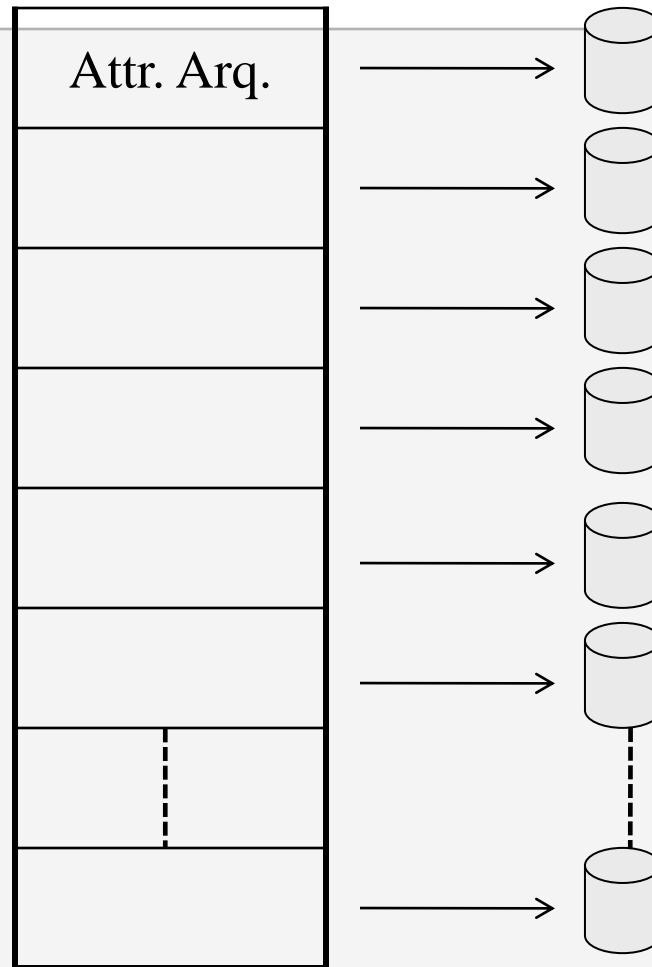
# *Estrutura de Diretórios*

## □ *Nível único*

- É a forma mais simples de implementação de diretórios.
- Há apenas um nível contendo todos os arquivos.
- Impossível de criar arquivos com mesmo nome.

# *Estrutura de Diretórios*

Nível único



Diretórios

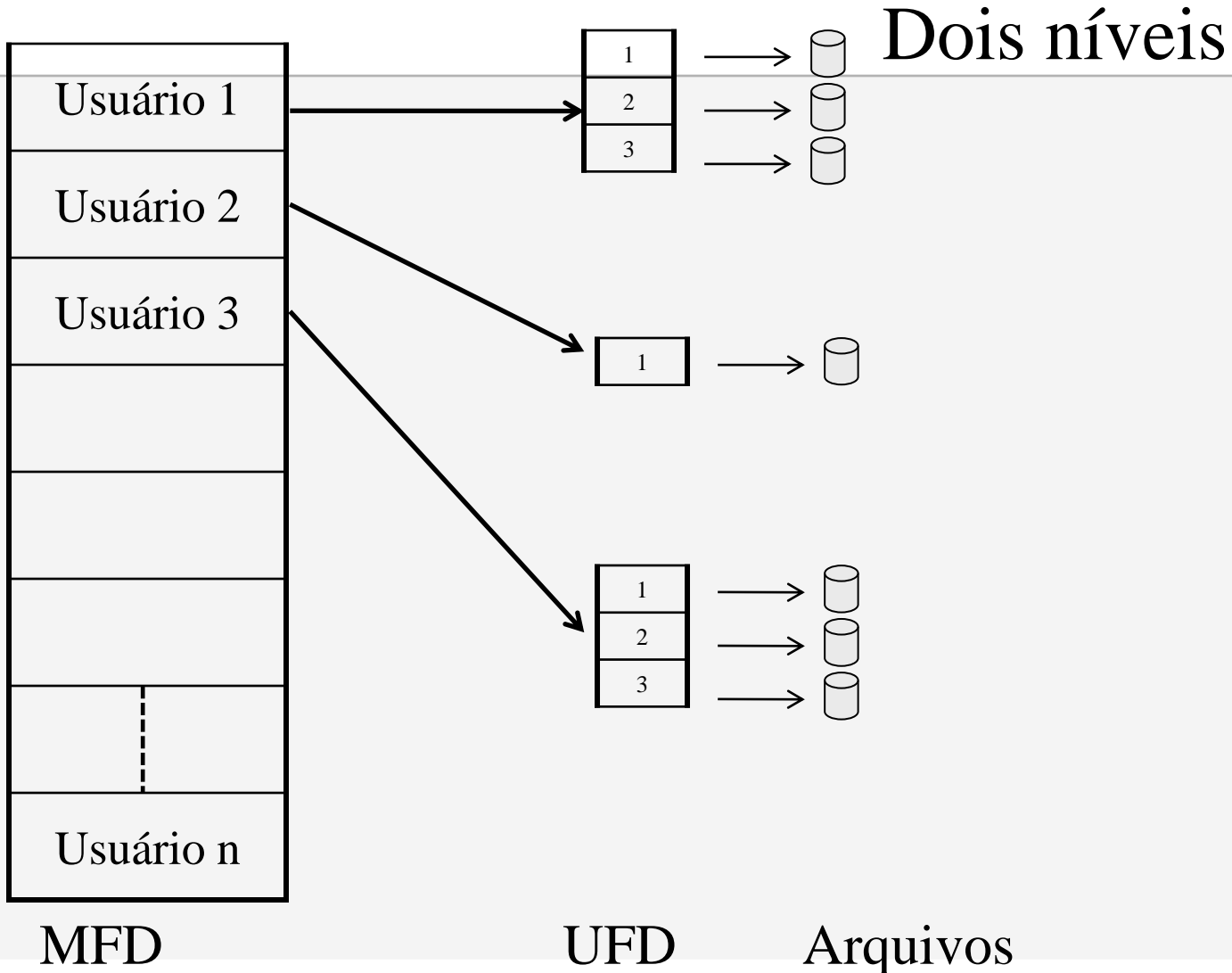
Arquivos

*Sistemas Operacionais*

# *Estrutura de Diretórios*

- ***Diretório com dois níveis (two-level directory)***
  - Posteriormente foi implementada a estrutura onde cada usuário tinha seu diretório (UFD – User File Directory) – cada usuário poderia criar os arquivos que desejasse.
  - Desta forma há uma Master File Directory (MFD), que na realidade é um nível adicional para controlar os diretórios individuais dos usuários.
  - Contem entradas que apontam para o diretório pessoal, sendo indexadas pelo nome do usuário.

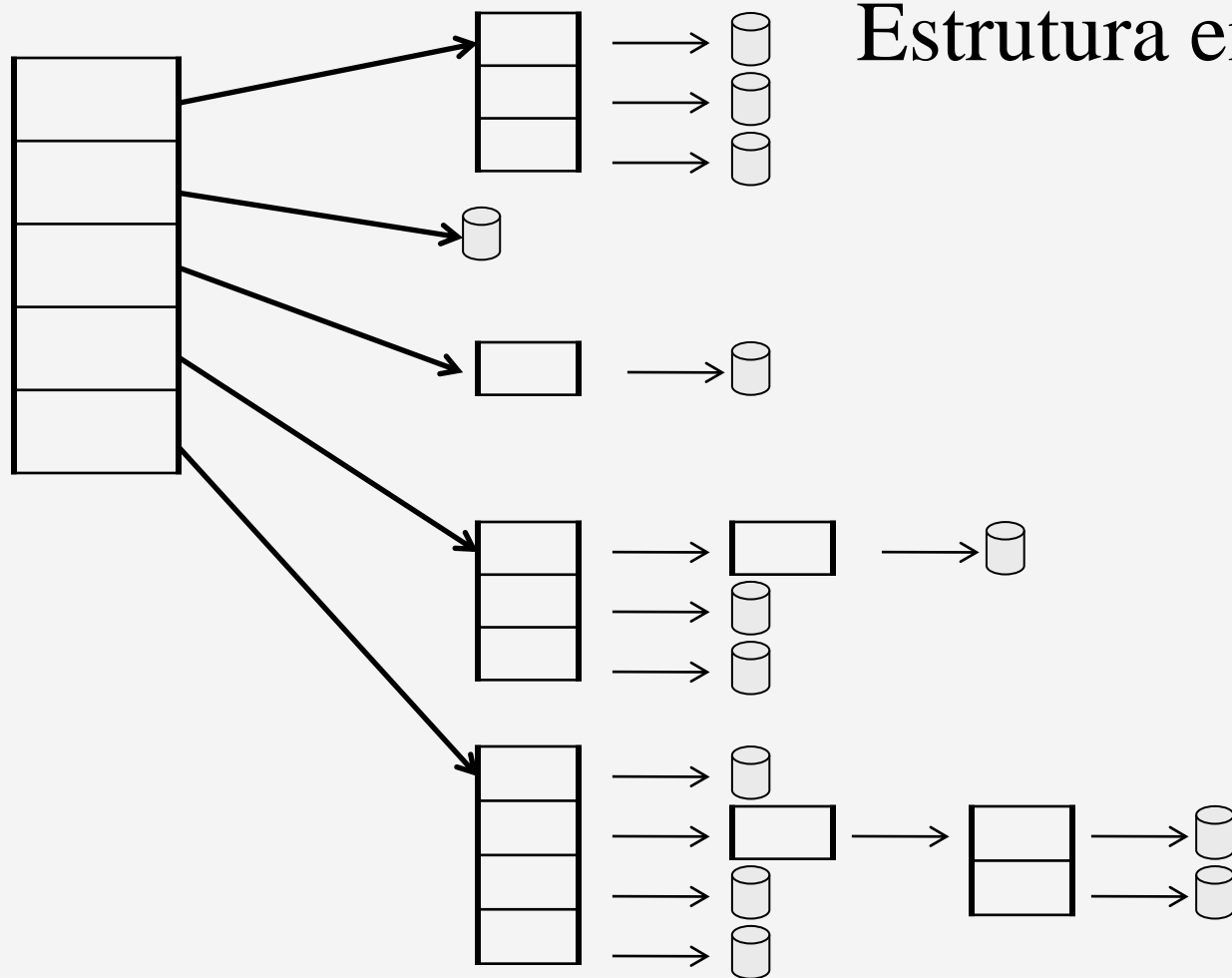
# *Estrutura de Diretórios*



# *Estrutura de Diretórios*

- *Diretórios em árvore (tree-structured directory)*
  - O nó raiz é a MFD, os galhos são os UFD e as folhas os arquivos.
  - Quando se referencia um arquivo é necessário especificar seu nome bem como o diretório onde ele se encontra (path).
  - Cria-se quantos diretórios desejar, e subdiretórios e arquivos.
  - Diretórios também podem ser tratados como “arquivos”, possuindo alguns atributos.

# *Estrutura de Diretórios*



Estrutura em árvore



# *Sistema de Arquivos - Métodos de acesso*

- Operações que podem ser realizadas em um diretório:
  - Pesquisa por um arquivo
  - Criação de um arquivo
  - Eliminação de um arquivo
  - Listar as entradas do diretório
  - Alterar o nome de um arquivo