

Sistemas Operacionais

Prof. Dr. Helder Oliveira

Plano de Aula

- Arquivos
- Organização de arquivos
- Diretórios
- Métodos de acesso

Introdução

- Condições essenciais para armazenamento de informações por um longo prazo:
 - Deve ser possível armazenar uma quantidade muito grande de informações.
 - As informações devem sobreviver ao término do processo que as está utilizando.
 - Múltiplos processos têm de ser capazes de acessá-las ao mesmo tempo.

Introdução

- Pense em um disco como uma sequência linear de blocos de tamanho fixo e que apoiam a leitura e registro dos blocos. As questões a seguir surgem rapidamente:
 - Como encontrar a informação?
 - Como manter um usuário de uma outra leitura de dados?
 - Como saber se os blocos são livres?
- Abstração: **Arquivos**

Introdução

- Os arquivos são gerenciados pelo sistema operacional e é mediante a implementação de arquivos que o sistema operacional estrutura e organiza as informações.
- A parte do sistema responsável pela gerência é denominada sistema de arquivo que é a parte mais visível do sistema operacional pois é uma atividade frequentemente realizada pelos usuários.
- Deve ocorrer de maneira uniforme independente dos diferentes dispositivos de armazenamento.

Arquivos

- São um conjunto de registros definidos pelo sistema de arquivos e podem ser armazenados em diferentes dispositivos físicos.
- É constituído de informações logicamente relacionadas, podendo representar programas ou dados.
- São identificado por meio de um nome, formado por uma sequência de caracteres. Em alguns sistemas operacionais, a identificação de um arquivo é composta por duas partes separadas por um ponto, a parte após o ponto é chamada extensão do arquivo e serve para identificar o conteúdo.

Arquivos

- Processos podem ler arquivos existentes e criar novos se necessário.
- Informações armazenadas em arquivos devem ser persistentes, isto é, não devem ser afetadas pela criação e término de um processo.
- Um arquivo deve desaparecer apenas quando o seu proprietário o remove explicitamente.
- Operações mais comum: para leitura e escrita de arquivos

Arquivos

- Arquivos são gerenciados pelo sistema operacional.
- A parte do sistema operacional que lida com arquivos é conhecida como **sistema de arquivos**.
- Arquivos são estruturados, nomeados, acessados, usados, protegidos, implementados e gerenciados.
- Do ponto de vista do usuário o aspecto mais importante de um sistema de arquivos é como ele aparece.

Nomeação de arquivos

- Um arquivo é um mecanismo de abstração.
- Ele fornece uma maneira para armazenar informações sobre o disco e lê-las depois.
- Nomeação de arquivos variam de certa maneira de sistema para sistema.
 - Podem distinguir entre letras maiúsculas e minúsculas.
- Muitos sistemas operacionais aceitam nomes de arquivos de duas partes, com as partes separadas por um ponto, como em **prog.c**.
- A parte que vem em seguida ao ponto é chamada de **extensão do arquivo** e costuma indicar algo seu a respeito.

Nomeação de arquivos

FIGURA 4.1 Algumas extensões comuns de arquivos.

| Extensão | Significado |
|----------|--|
| .bak | Cópia de segurança |
| .c | Código-fonte de programa em C |
| .gif | Imagem no formato Graphical Interchange Format |
| .hlp | Arquivo de ajuda |
| .html | Documento em HTML |
| .jpg | Imagem codificada segundo padrões JPEG |
| .mp3 | Música codificada no formato MPEG (camada 3) |
| .mpg | Filme codificado no padrão MPEG |
| .o | Arquivo objeto (gerado por compilador, ainda não ligado) |
| .pdf | Arquivo no formato PDF (Portable Document File) |
| .ps | Arquivo PostScript |
| .tex | Entrada para o programa de formatação TEX |
| .txt | Arquivo de texto |
| .zip | Arquivo compactado |

- Em alguns sistemas, as extensões de arquivos são apenas convenções.
- Mas um compilador C pode realmente insistir em que os arquivos que ele tem de compilar terminem em **.c**.

Organização de Arquivos

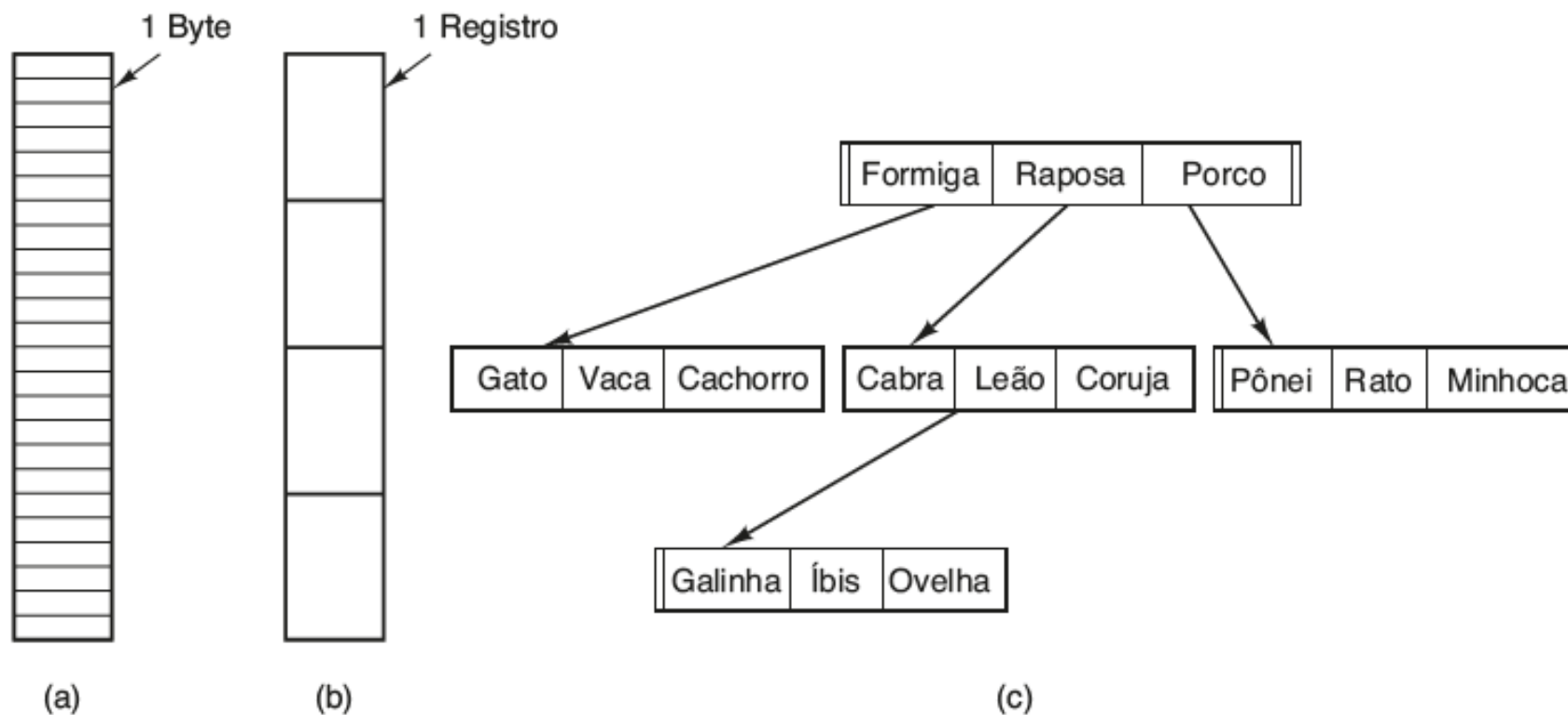
- Consiste no modo como seus dados estão internamente armazenados.
- Quando o arquivo é criado pode-se definir que organização será adotada, que pode ser uma estrutura suportada pelo sistema operacional ou definida pela própria aplicação.
- A forma mais simples de organização de arquivos é através de uma sequência não-estruturada de bytes, onde o sistema de arquivos não impõe nenhuma estrutura lógica para os dados, a aplicação deve definir toda a organização.

Organização de Arquivos

- A grande vantagem deste modelo é a flexibilidade para criar estruturas de dados, porém todo o controle de dados é de responsabilidade da aplicação.
- Alguns sistemas operacionais estabelecem diferentes organizações de arquivos e cada um deve seguir um modelo suportado pelo sistema de arquivos.

Estrutura de Arquivos

FIGURA 4.2 Três tipos de arquivos. (a) Sequência de bytes. (b) Sequência de registros. (c) Árvore.



Estrutura de Arquivos

- Na **sequencia desestruturada de bytes** o SO não sabe o que o arquivo contém. Tudo o que ele vê são bytes. Qualquer significado deve ser imposto pelos programas em nível de usuário.
- UNIX, MS-DOS e Windows utilizam essa estratégia.
- Oferece máxima flexibilidade.
- Os programas de usuários podem colocar qualquer coisa em seus arquivos e nomeá-los como lhes convier.

Estrutura de Arquivos

- No tipo **sequencia de registros**, um arquivo é uma sequencia registros de tamanho fixo, cada um com alguma estrutura interna.
 - A ideia central é que a operação de leitura retorna um registro e a de escrita sobrepõe ou anexa um registro.
 - Nenhum sistema atual usa esse modelo como sistema primário de arquivo, mas ele era comum nos computadores de grande porte na época dos cartões perfurados.

Estrutura de Arquivos

- No último tipo, um arquivo é constituído de uma **árvore de registros**, não necessariamente todos de mesmo tamanho, cada um contendo um campo-chave em uma posição fixa no registro.
 - A árvore é ordenada pelo campo chave pra que se busque mais rapidamente por uma chave específica. A operação básica não é obter o próximo registro mas obter o registro com a chave específica.
 - Esse tipo é bem diferente do fluxo de bytes desestruturado usados no UNIX e Windows mas é aplicado em computadores de grande porte ainda usados para alguns processamentos de dados comerciais.

Tipos de Arquivos

- Muitos sistemas operacionais aceitam vários tipos de arquivos.
- Arquivos regulares são aqueles que contêm informações do usuário.
- Diretórios são arquivos do sistema para manter a estrutura do sistema de arquivos.
- Arquivos especiais de caracteres são relacionados com entrada/saída e usados para modelar dispositivos de E/S seriais como terminais, impressoras e redes. Arquivos especiais de blocos são usados para modelar discos.

Tipos de Arquivos

- Arquivos regulares geralmente são arquivos ASCII ou arquivos binários.

Tipos de Arquivos

- A representação da informação dentro de um computador digital é feita na forma de números.
- No Linux ou no Windows, arquivos são concebidos e tratados como STREAMS de BYTES.
- O que é um BYTE ? Um BYTE é um número entre 0 e 255.
 - Um BYTE é um conjunto de 8 bits (dígitos binários)

Decifrando os números: o código

- Tabela ASCII
 - A tabela ASCII dá os caracteres associados aos números 0 a 127.

| | | |
|---------------|----------------|---------------|
| A : 65 | a : 97 | 0 : 48 |
| B : 66 | b : 98 | 1 : 49 |
| ... | ... | ... |
| Z : 90 | z : 122 | 9 : 57 |

Caracteres especiais :

| | |
|----|---------------------|
| 0 | : NULO |
| 8 | : BACKSPACE |
| 9 | : TAB |
| 10 | : mudança de linha |
| 12 | : mudança de página |
| 13 | : ENTER |

Representação de um arquivo

Representação interna do conteúdo de um arquivo

| | | | | | | | | | | | | | |
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 65 | 66 | 67 | 10 | 49 | 50 | 51 | 10 | 68 | 69 | 10 | 97 | 98 | 10 |
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|

Convertendo a representação interna de um arquivo
que eu *considero* como um arquivo texto:

| | | | | | | | | | | | | | |
|---|---|---|----|---|---|---|----|---|---|----|---|---|----|
| A | B | C | 10 | 1 | 2 | 3 | 10 | D | E | 10 | a | b | 10 |
|---|---|---|----|---|---|---|----|---|---|----|---|---|----|

*mudança
de linha*

*mudança
de linha*

*mudança
de linha*

*mudança
de linha*

Representação de um arquivo

Visualizando o arquivo no meu Terminal de Vídeo:

| | | | | | | | | | | | | | |
|---|---|---|----|---|---|---|----|---|---|----|---|---|----|
| A | B | C | 10 | 1 | 2 | 3 | 10 | D | E | 10 | a | b | 10 |
|---|---|---|----|---|---|---|----|---|---|----|---|---|----|

```
$ cat arquivo
```

```
ABC
```

```
123
```

```
DE
```

```
ab
```

```
$
```

Métodos de Acesso aos arquivos

- Em função de como o arquivo esta organizado o sistema de arquivos pode recuperar registros de diferentes maneiras:
 - **Acesso sequencial.**
 - Um processo podia ler todos os bytes ou registros em um arquivo em ordem.
 - Inicia do princípio, mas não pode pular nenhum ou lê-los fora de ordem.
 - Podiam voltar ao ponto de partida.
 - Fita magnética
 - **Acesso aleatório**
 - Processo podia ler os bytes ou registros de um arquivo fora de ordem, ou acessar os registros pela chave em vez de pela posição.
 - Discos

Atributos de arquivos

- São informações de controle que variam de acordo com sistema de arquivos.
- Todo arquivo possui um nome e sua data. Além disso, os sistemas operacionais associam outras informações com os arquivos, como data e horário em que foi modificado pela última vez, e tamanho do arquivo. Esses itens extras são atributos do arquivo ou metadados.
- Alguns atributos especificados na criação do arquivo não podem ser mudados e outros são modificados pelo próprio sistema operacional. E ainda existe alguns que podem ser alterados pelo usuário tais como proteção, tamanho e senha.

Atributos de arquivos

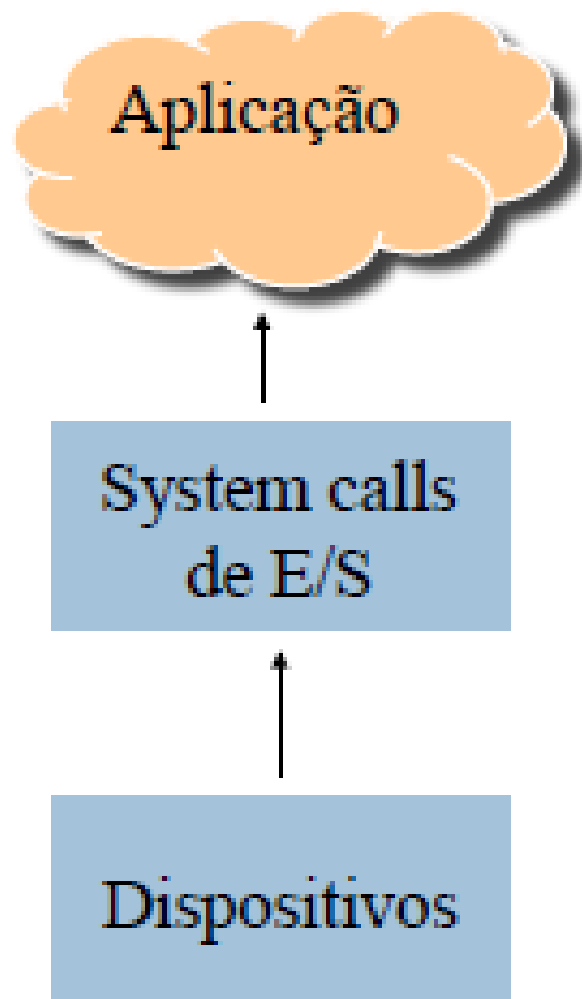
FIGURA 4.4 Alguns possíveis atributos de arquivos.

| Atributo | Significado |
|-----------------------------|--|
| Proteção | Quem tem acesso ao arquivo e de que modo |
| Senha | Necessidade de senha para acesso ao arquivo |
| Criador | ID do criador do arquivo |
| Proprietário | Proprietário atual |
| Flag de somente leitura | 0 para leitura/escrita; 1 para somente leitura |
| Flag de oculto | 0 para normal; 1 para não exibir o arquivo |
| Flag de sistema | 0 para arquivos normais; 1 para arquivos de sistema |
| Flag de arquivamento | 0 para arquivos com backup; 1 para arquivos sem backup |
| Flag de ASCII/binário | 0 para arquivos ASCII; 1 para arquivos binários |
| Flag de acesso aleatório | 0 para acesso somente sequencial; 1 para acesso aleatório |
| Flag de temporário | 0 para normal; 1 para apagar o arquivo ao sair do processo |
| Flag de travamento | 0 para destravados; diferente de 0 para travados |
| Tamanho do registro | Número de bytes em um registro |
| Posição da chave | Posição da chave em cada registro |
| Tamanho da chave | Número de bytes na chave |
| Momento de criação | Data e hora de criação do arquivo |
| Momento do último acesso | Data e hora do último acesso do arquivo |
| Momento da última alteração | Data e hora da última modificação do arquivo |
| Tamanho atual | Número de bytes no arquivo |
| Tamanho máximo | Número máximo de bytes no arquivo |

Operações com arquivos

- Arquivos existem para armazenar informações e permitir que elas sejam recuperadas depois.
- Sistemas diferentes proporcionam operações diferentes para permitir armazenamento e recuperação.
- O sistema de arquivos oferece um conjunto de system calls que permite às aplicações realizar operações de E/S, como tradução de nomes em endereços, leitura e gravação de dados e criação/eliminação de arquivos.

Operações com arquivos



As system calls de E/S tem como função oferecer uma interface simples e uniforme entre a aplicação e os diversos dispositivos.

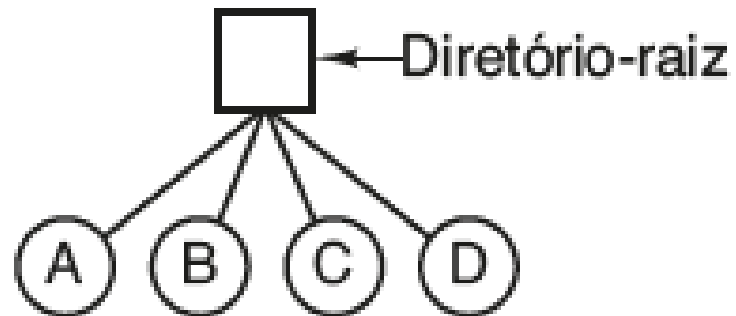
Operações com arquivos

| Comando | Descrição |
|---------------|---------------------------------|
| <i>create</i> | Criação de um arquivo |
| <i>open</i> | Abertura de um arquivo |
| <i>read</i> | Leitura de dados de um arquivo |
| <i>write</i> | Gravação de dados de um arquivo |
| <i>close</i> | Fechamento de um arquivo |
| <i>rename</i> | Alteração de nome de um arquivo |
| <i>erase</i> | Eliminação de um arquivo |

Diretórios

- Para controlar os arquivos, sistemas de arquivos normalmente têm **diretórios** ou **pastas**, que são em si arquivos. A forma mais simples de um sistema de diretório é ter um diretório contendo todos os arquivos, um **diretório-raiz**.

FIGURA 4.6 Um sistema de diretório em nível único contendo quatro arquivos.

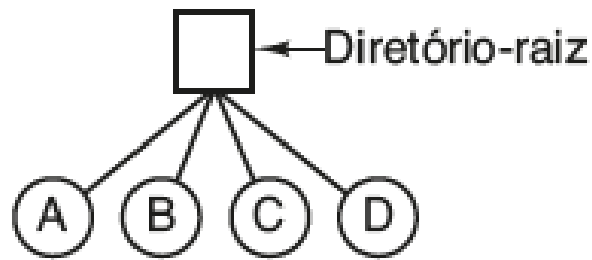


Diretórios

Nível Único (single-level directory):

- É a implementação mais simples, existe apenas um único diretório contendo todos os arquivos do disco. É bastante limitado já que não permite que usuários criem arquivos com o mesmo nome o que ocasionaria um conflito no acesso aos arquivos.

FIGURA 4.6 Um sistema de diretório em nível único contendo quatro arquivos.



Diretórios

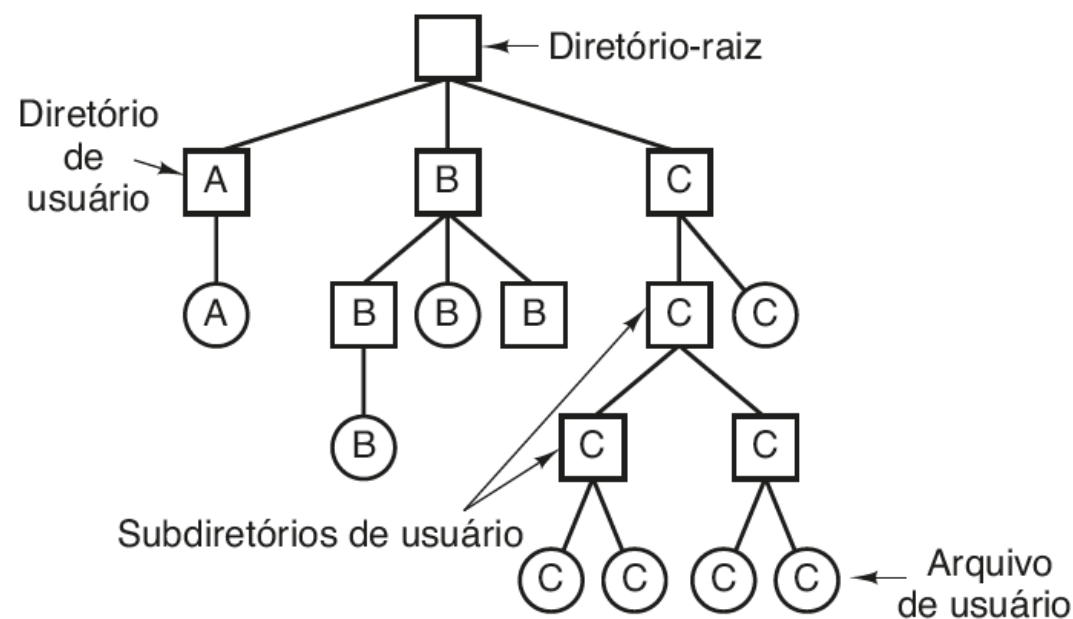
- User File Directory (UFD): para cada usuário existe um diretório particular e assim poderia criar arquivos com qualquer nome.
- Deve haver um nível de diretório adicional para controle chamado de Master File Directory (MFD) que é indexado pelo nome do usuário e cada entrada aponta para o diretório pessoal.
- É análoga a uma estrutura de dados em árvore onde o MFD é a raiz, os galhos são a UFD e os arquivos são as folhas.
- Quando se referencia a um arquivo é necessário especificar seu nome e seu diretório isto é chamado de path (caminho).

Diretórios

Sistemas de diretórios hierárquicos

- Maneira para agrupar arquivos relacionados em um mesmo local.
- Com hierarquia, o usuário pode ter tantos diretórios quantos forem necessários.
- Além disso, se múltiplos usuários compartilham um servidor de arquivos comum, cada usuário pode ter um diretório-raiz privado para sua própria hierarquia.

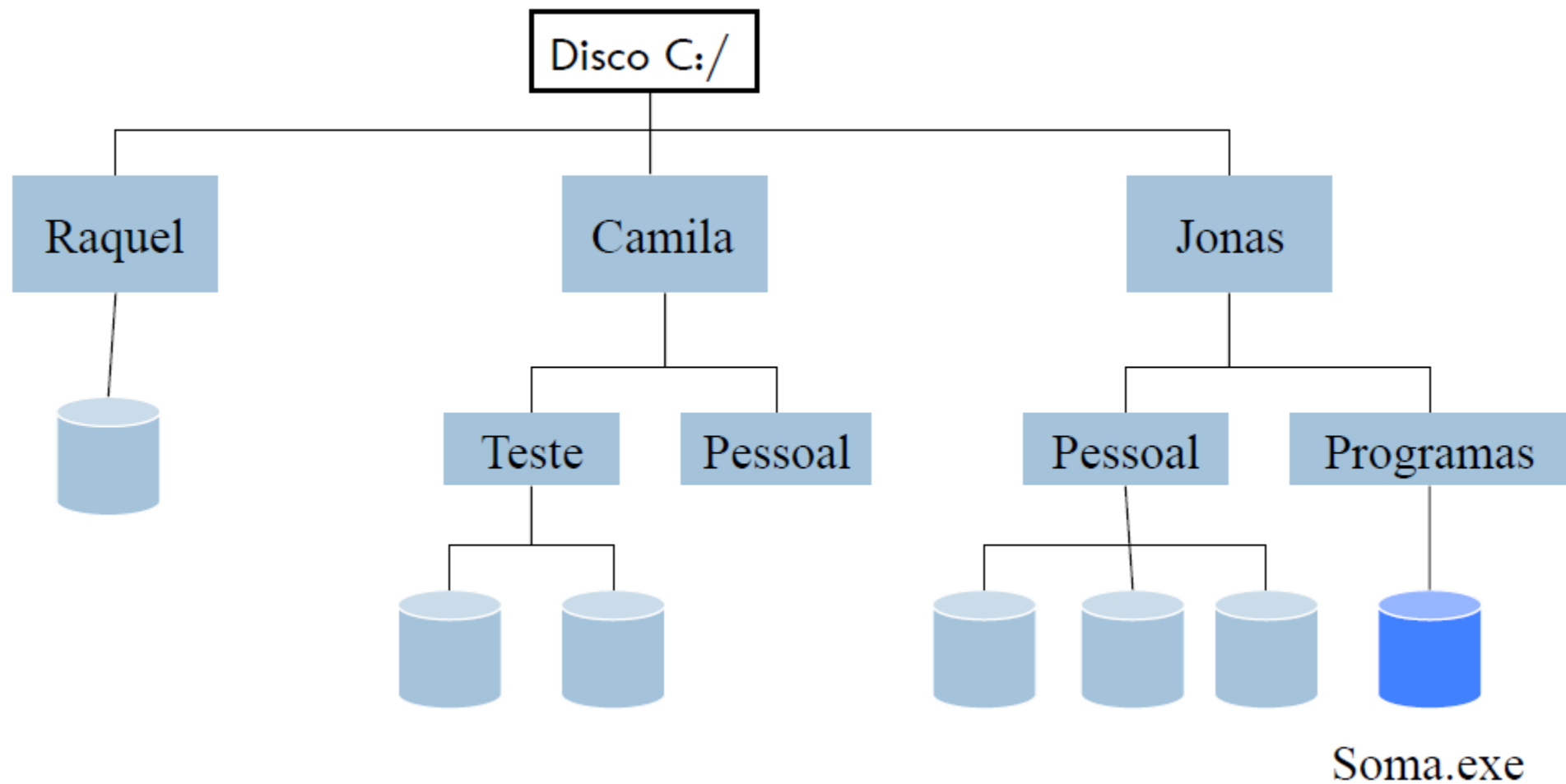
FIGURA 4.7 Um sistema hierárquico de diretórios.



Nomes de caminhos

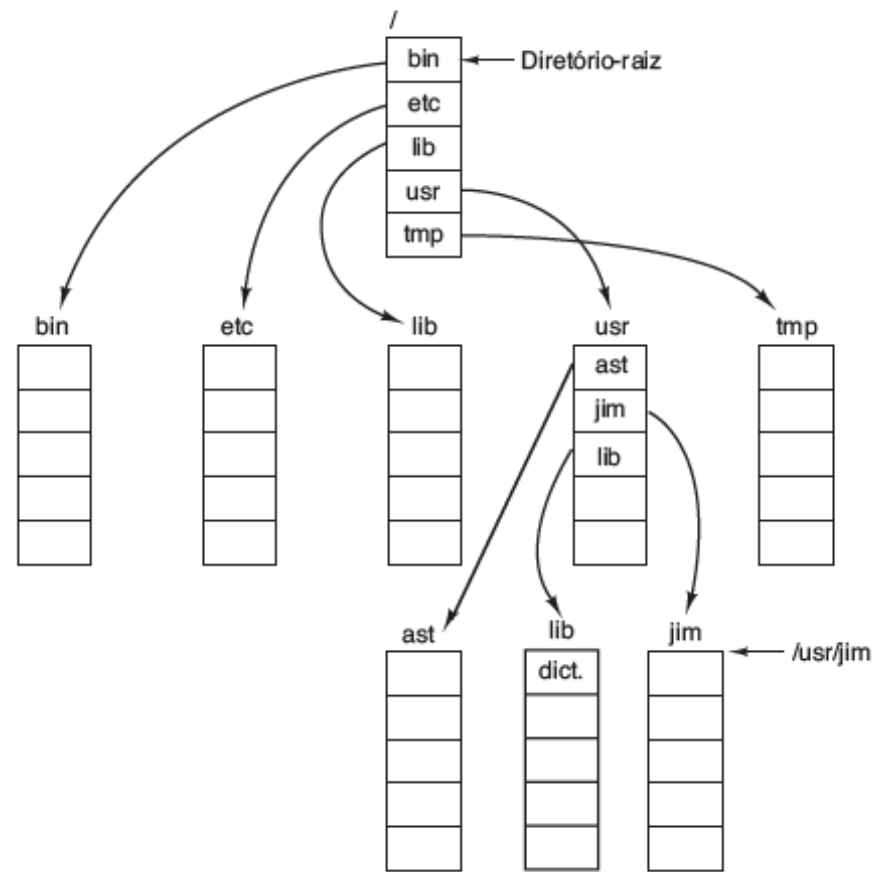
- Maneira necessária para especificar os nomes dos arquivos.
- Dois métodos diferentes são os mais usados.
 - **Nome de caminho absoluto**
 - Cada arquivo recebe o nome do diretório-raiz.
 - Caminhos absolutos sempre começam no diretório-raiz e são únicos.
 - Ex: */usr/ast/caixapostal*
 - **Nome de caminho relativo**
 - Usado em conjunção com o conceito do diretório de trabalho.
 - Usuário pode designar um diretório como o de trabalho atual, caso em que todos os nomes de caminho não começando no diretório-raiz são presumidos como relativos ao diretório de trabalho.
 - Ex: */usr/ast -> /usr/ast/caixapostal*

Nomes de caminhos



Nomes de caminhos

FIGURA 4.8 Uma árvore de diretórios UNIX.



Implementação do sistema de arquivos

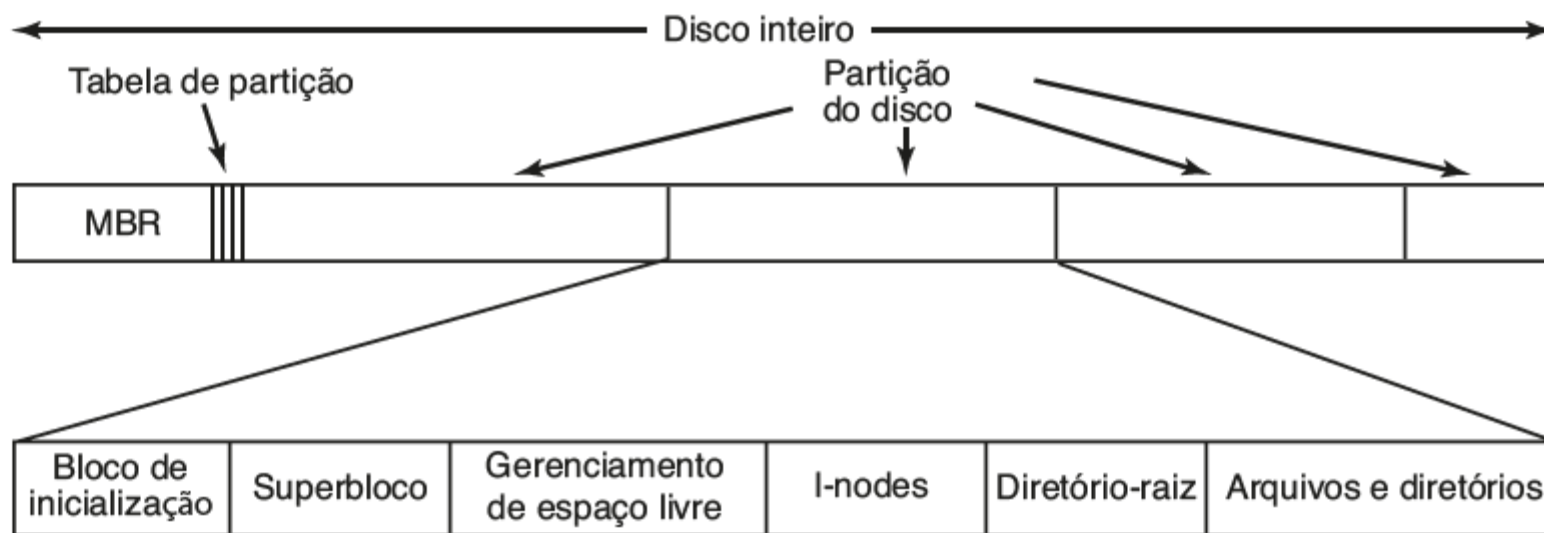
- Implementadores estão interessados em como os arquivos e os diretórios estão armazenados, como o espaço de disco é gerenciado e como fazer tudo funcionar de maneira eficiente e confiável.
- Sistemas de arquivos são armazenados em discos. A maioria dos discos pode ser dividida em uma ou mais partições, com sistemas de arquivos independentes em cada partição. O Setor 0 do disco é chamado de MBR (Master Boot Record — registro mestre de inicialização) e é usado para inicializar o computador.

Esquema do sistema de arquivos

- Sistemas de arquivos são armazenados em discos.
- A maioria dos discos pode ser dividida em uma ou mais partições, com sistemas de arquivos independentes em cada partição.
- O Setor 0 do disco é chamado de MBR e é usado para inicializar o computador.
- O fim do MBR contém a tabela de partição.
 - Dá os endereços de início e fim de cada partição.
- BIOS lê e executa o MBR.

Esquema do sistema de arquivos

FIGURA 4.9 Um esquema possível para um sistema de arquivos.



Próxima aula

- Implementação de arquivos
- Implementação de diretórios
- Arquivos compartilhados
- Sistemas de arquivos
- Gerenciamento de arquivos

Leitura

- SISTEMAS OPERACIONAIS MODERNO 4^a edição
 - 4 – Sistemas de arquivos

Dúvidas?