

Sistemas Operacionais

Sistema de Arquivos – (parte I de II)

Sistema de Arquivos

Responsável por implementar em software um recurso que não existe no hardware:

- O hardware oferece simplesmente um grande conjunto de bytes contíguos, e a tarefa principal do sistema de arquivos é implementar a abstração de arquivo em cima do dispositivo de armazenamento.



Sistema de Arquivos

Após a formatação física, feita pelo próprio fabricante do disco rígido nas etapas finais da produção, temos um HD dividido em trilhas, setores e cilindros, toda a infraestrutura básica para permitir que a cabeça de leitura possa ler e gravar dados.

A formatação física permite que a placa lógica posicione corretamente as cabeças de leitura.



Sistema de Arquivos

Porém, para que este disco possa ser reconhecido e utilizado pelo sistema operacional, é necessária uma nova formatação, a chamada formatação lógica.

A formatação lógica consiste em escrever no disco a estrutura do sistema de arquivos utilizado pelo sistema operacional.



Sistema de Arquivos

Podemos visualizar a formatação lógica com a “linguagem” utilizada pelo sistema operacional.

Em um passado recente, os fabricantes de HD disponibilizavam formatadores físicos.

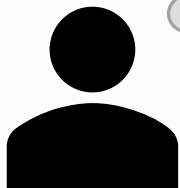
- Esta opção foi abandonada com o tempo.
- O que se faz hoje em dia é sobrescrever todas as posições do disco com 0's. (*zero fill*).
- O HD volta ao estado que saiu de fábrica.
 - Se não houver um dano físico.

Sistema de Arquivos

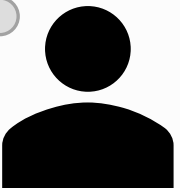
Um sistema de arquivos é um conjunto de estruturas lógicas e de rotinas, que permitem ao sistema operacional controlar o acesso ao disco rígido. Diferentes sistemas operacionais usam diferentes sistemas de arquivos:

- **MS-DOS, Win 3.x, Win95 e Win98:** FAT12, FAT16, FAT32.
- **Windows 2000, NT, XP, Vista, 7, 8, 8.1 e 10:** NTFS.
 - A Microsoft começou a trabalhar no desenvolvimento do WinFS (Windows Future Storage) para o Windows 7, entretanto, o projeto foi cancelado.
- **Linux:** Ext2, Ext3, Ext4, ReiserFS, XFS, VFAT, etc.

Sistema de Arquivos



**Por que existem
diferentes
sistemas de
arquivos?**



**Diferentes
hardwares
necessitam de
diferentes sistemas
de arquivos?**

Sistema de Arquivos



Disco rígido



Memória Flash na USB



IPOD usa memória flash



Cartão de memória que utiliza tecnologia Flash

Sistema de Arquivos

Um processo em execução pode armazenar uma quantidade limitada de informação na Memória Virtual.

Contudo, para algumas aplicações este tamanho não é adequado.

- Exemplo, sistemas que necessitam de um “banco de dados”.

Com o advento do “Big Data”, é comum SGBD's que trabalham com vários TB de informação.

Sistema de Arquivos

O dispositivo de armazenamento de um sistema de arquivos pode ser qualquer mídia (também chamada de memória secundária), a qual provê um meio de:

- Armazenamento em massa.
- Persistência de dados.
- Compartilhamento de informação entre processos.

Sistema de Arquivos

É então necessário que processos possam ler, criar e apagar arquivos (manipulação).

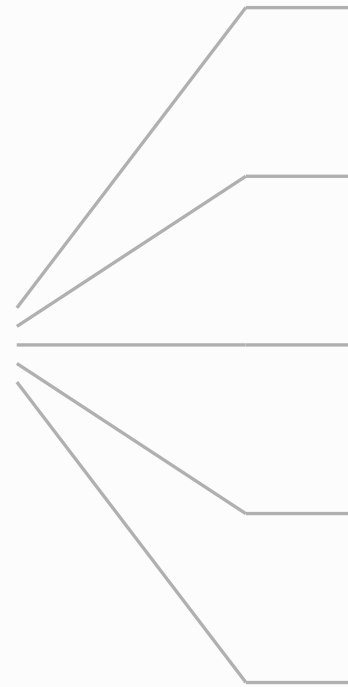
- Através de chamadas de sistema.

É então necessário que seja fornecida a persistência:

- A existência de um arquivo é independente ao processo, ou seja, quando o processo é finalizado, o arquivo pode continuar existindo.
- É então necessário que os arquivos devem ficar armazenados fora da memória principal (que é volátil).

Sistema de Arquivos

O projeto de um sistema de arquivos envolve principalmente:



Estrutura.

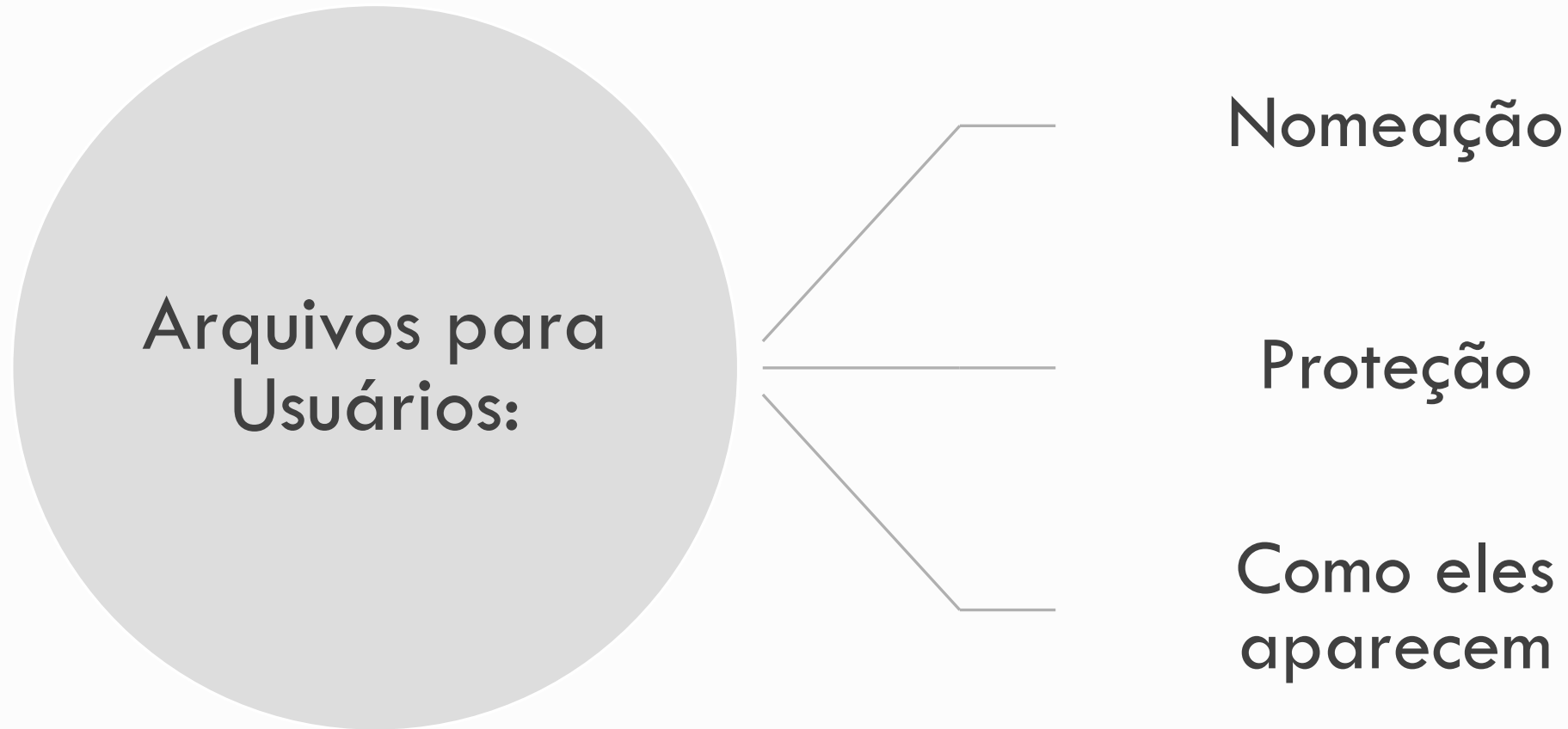
Nomeação.

Acesso.

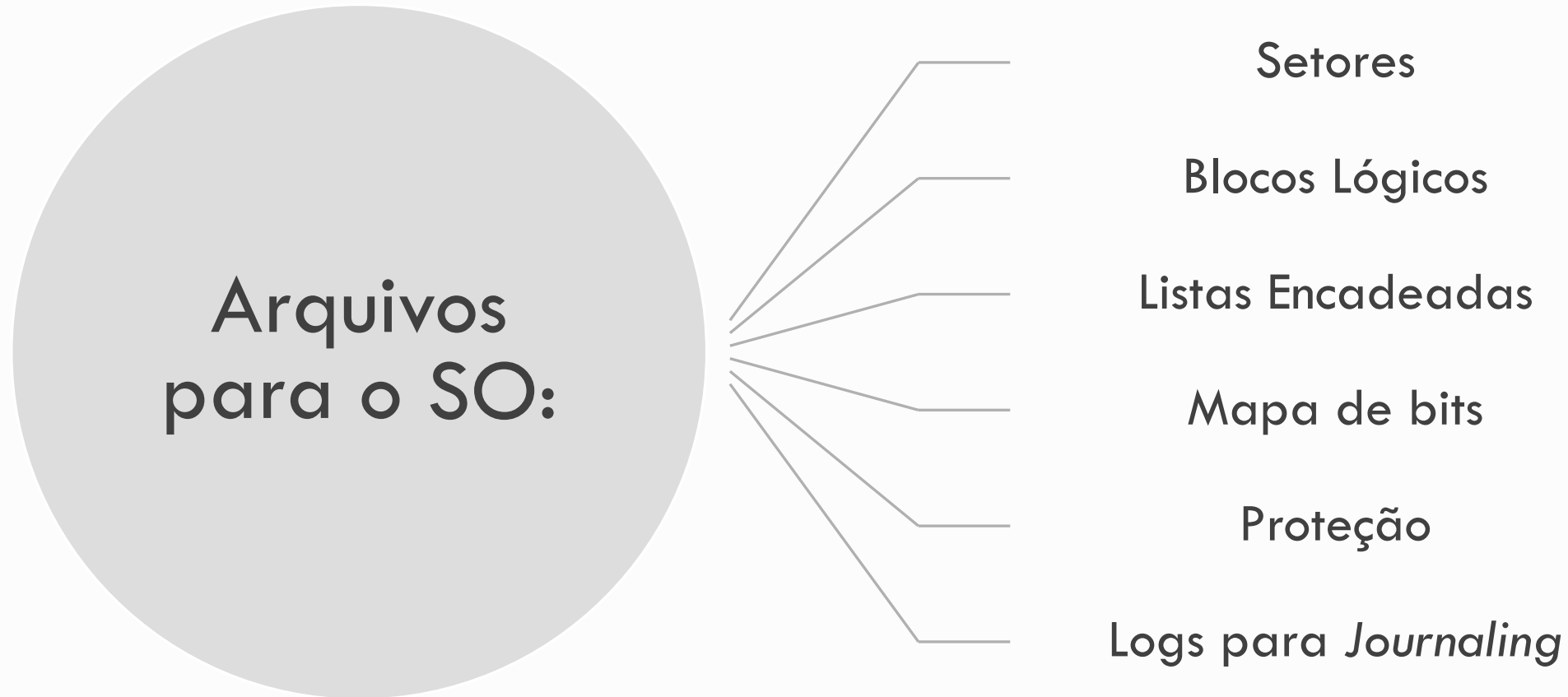
Uso.

Proteção.

Sistema de Arquivos



Sistema de Arquivos



Arquivos

Arquivo

Arquivo é um mecanismo de abstração.

O objetivo é isolar o usuário sobre detalhes de como e onde a informação está armazenada, e como os discos na verdade funcionam (máquina estendida).

Arquivo

Criação de arquivos:

- Quando um processo faz uma Chamada de Sistema para criação de um arquivo, ele dá um nome a este arquivo (primeiro passo).
- As regras de nomeação variam de sistema para sistema.
- Todos os sistemas permitiam strings de até 8 letras:
 - maria
 - roberto

Arquivo

Criação de arquivos:

- Alguns sistemas permitiam dígitos e caracteres especiais:
 - Fig.3-23
- Muitos S.O.s permitem nomes de até 255 caracteres.
 - Na verdade não é o S.O. que permite. É o seu o sistema de arquivos utilizado.

Arquivo

Alguns sistemas de arquivos diferenciam letras maiúsculas de minúsculas.

- Case sensitive

EXT2, EXT3, EXT4 do UNIX e o do Linux diferenciam:

- Podemos ter diferentes arquivos chamados:
 - Maria
 - maria
 - MARIA
 - MaRiA

Família FAT e NTFS não diferenciam:

- Maria e MARIA são considerados o mesmo arquivo.

Arquivo

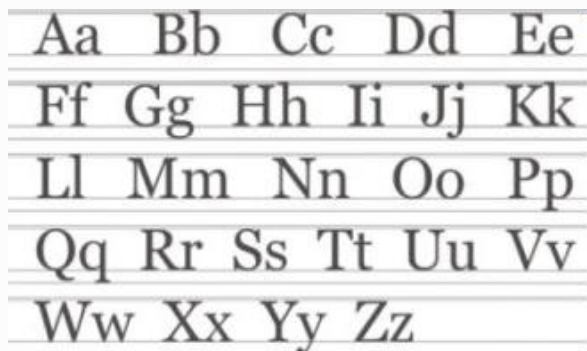
Win 95 e Win98 usam o mesmo sistema de arquivos do MS-DOS. Portanto, herdam todas as propriedades de nomeação.

O WinNT e Win2000 suportam o sistema de arquivos do MS-DOS. Portanto, também herdam suas propriedades de nomeação.

Arquivo

Contudo, o WinNT e o WIN 2000 suportam o sistema de arquivos chamado NTFS (com suporte a unicode).

- Unicode é um padrão que permite aos computadores representar e manipular, de forma consistente, texto de qualquer sistema de escrita existente.



Arquivo

Muitos S.O.s suportavam nomes de arquivos com duas partes:

- **Nome.extensão**, exemplo:
 - `codigo.cpp`; `MinhaClasse.java` ou; `Documento.doc`

No MS-DOS, por exemplo:

- Nome com até 8 caracteres e extensão de até 3 caracteres.

No UNIX, por exemplo:

- O tamanho da extensão fica a critério do usuário.
- O arquivo pode ter mais de uma extensão: `prog.c.z`

Arquivo

As extensões são apenas convenções, e geralmente não são impostas pelo S.O..

Servem apenas para lembrar o usuário, que determinado arquivo é de determinado tipo.

Mas a nível de aplicação, extensões podem ser exigidas, por exemplo:

- Compilador de C exige extensão .c
- MS-Word exige extensão .doc, .docx, .rtf, htm, html, etc.

Arquivo

No Windows, os usuários finais e as aplicações podem registrar extensões no sistema operacional.

O objetivo é especificar qual programa reconhece aquela extensão.

Tal requisito foi especificado apenas para facilitar o uso do usuário final.

Estrutura de Arquivos

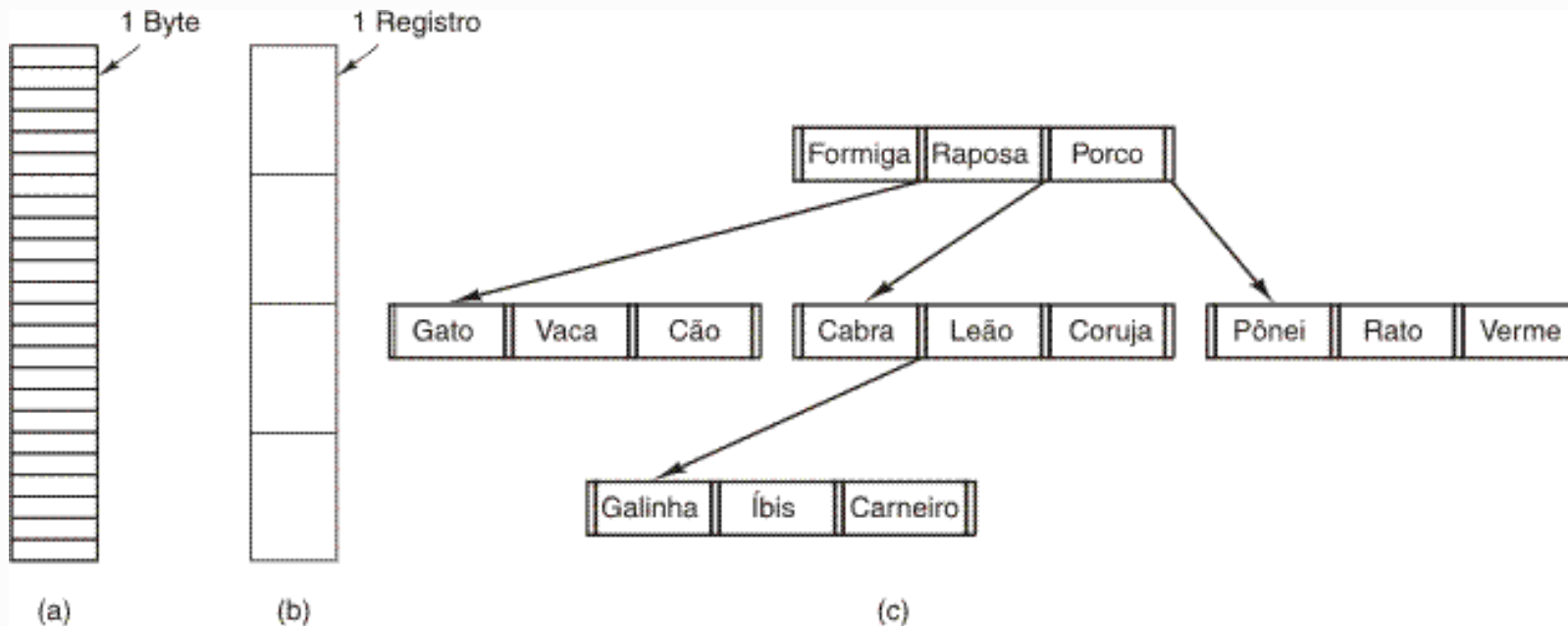
Estrutura de Arquivos

Arquivos pode ser estruturados de várias formas:

a)
Sequência de bytes

b)
Sequência de registros

c)
Em árvore



Estrutura de Arquivos



(a)

O sistema operacional não sabe o que o arquivo contém.

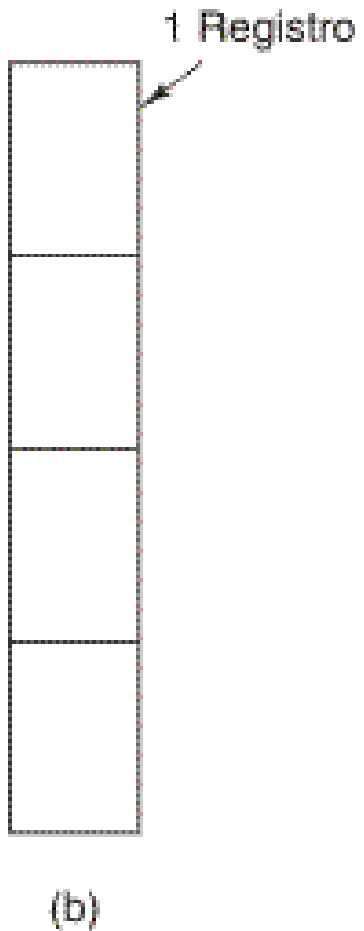
Qualquer significado é entendido a nível de programa de usuários.

Unix e Windows e a grande maioria dos S.O.s modernos utilizam esta estratégia.

Máxima flexibilidade.

É possível o programador desenvolver algo incomum, já que quem interpreta sua lógica é o próprio programa desenvolvido por ele.

Estrutura de Arquivos

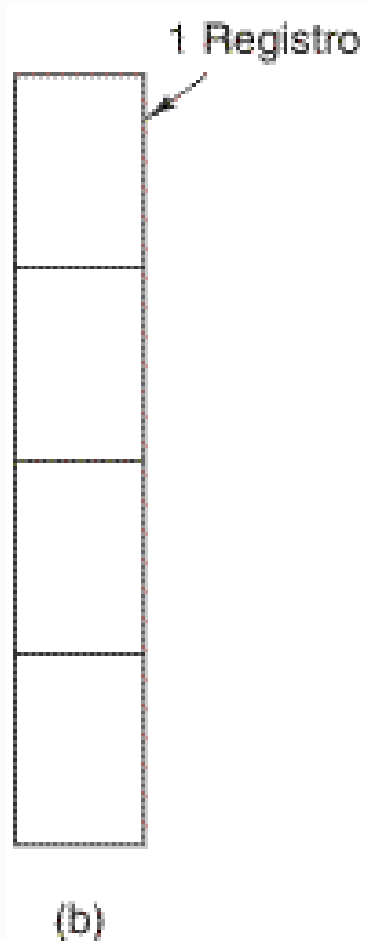


O arquivo é uma sequência de registros de tamanho fixo.

Cada registro com uma estrutura interna.

A operação de leitura e escrita passa a ser simplificada.

Estrutura de Arquivos



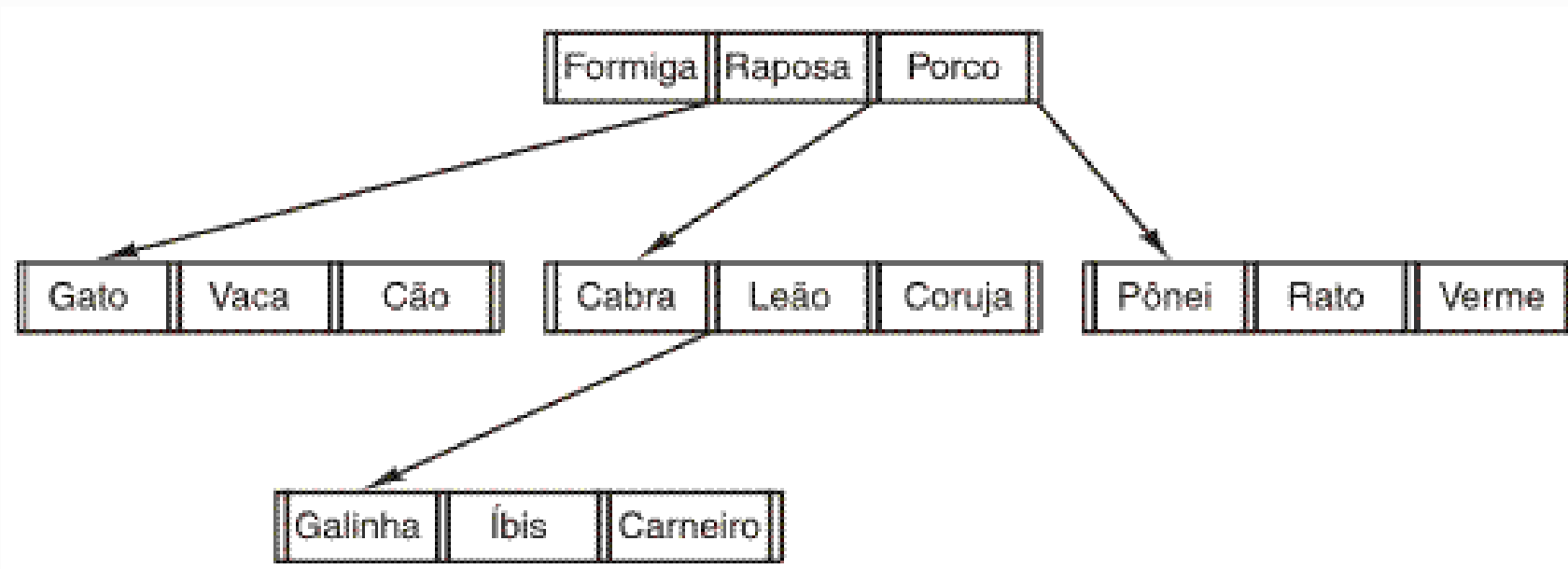
Como lembrete histórico:

- Cartões perfurados, por exemplo de 80 colunas.
- Na época, muitos sistemas operacionais tinham seus registros com 80 caracteres.
- Ou seja, os registros eram as imagens dos cartões.
- Nenhum S.O. atual funciona assim.

Estrutura de Arquivos

Os registros não tem necessariamente o mesmo tamanho.

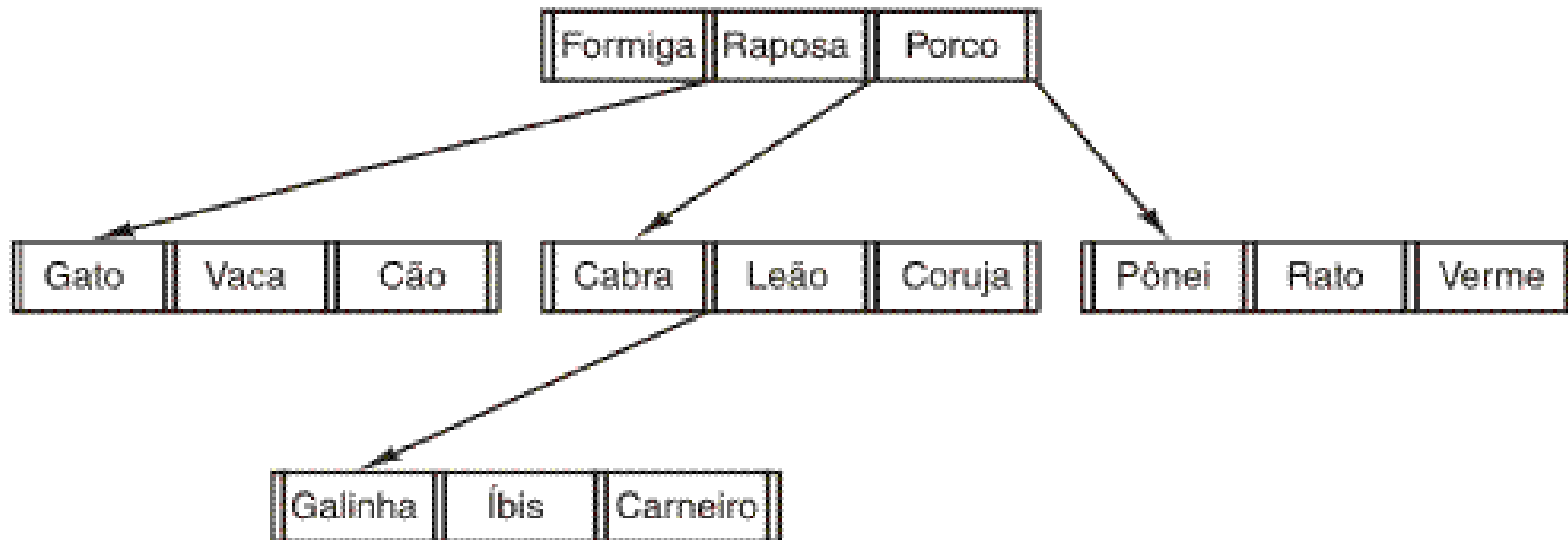
Cada registro possui um campo chave.



Estrutura de Arquivos

A árvore é ordenada pelo campo chave.

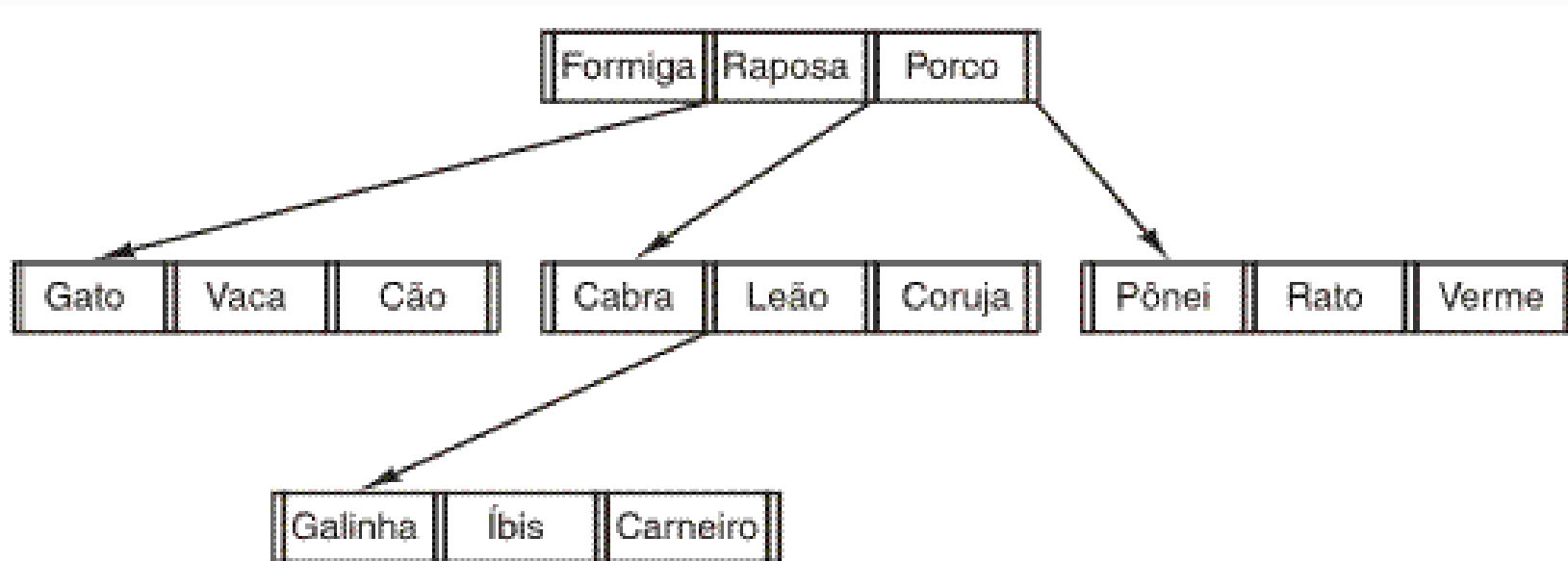
A ordenação tem o objetivo de acelerar a busca por chaves específicas.



Estrutura de Arquivos

Este esquema seria interessante para SGBD's.

Mas atualmente cada SGBD implementa a sua estrutura de dados com diversos tipos de índices para acelerar consultas.



Tipos de Arquivos

Tipos de Arquivos

Comum no UNIX e no Windows

- Arquivos regulares (ASCII ou binários).
- Arquivos diretório (cada SO implementa de uma forma).

Os arquivos regulares possuem informações do usuário.

Os arquivos diretório possuem informações da estrutura do sistema de arquivos.

- Utilizados para controle do S.O. e alterados apenas com chamadas de sistema.

Tipos de Arquivos

Os arquivos regulares são geralmente arquivos ASCII (Código Padrão Americano para o Intercâmbio de Informação).

Os arquivos ASCII são constituídos de linhas de texto. Elas não necessariamente possuem o mesmo tamanho.

Tipos de Arquivos

Um detalhe importante é que o caractere de quebra de linha varia entre os S.O.s.

- Windows: “\r\n” (caractere 13 e caractere 10).
- Unix: “\n” (caractere 10).

Muitos editores possuíam problemas na leitura de arquivos ASCII por este detalhe.

- Alguns conseguiam visualizar ambos de forma correta.

Tipos de Arquivos

A grande vantagem de arquivos ASCII é que podem ser mostrados e editados em qualquer editor de texto.

Se vários programas utilizam ASCII como entrada e saída, é fácil conectar a saída de um programa na entrada de outro (como o pipeline).

Tipos de Arquivos

A visualização um arquivo binário em um editor de textos nos apresenta algo incompreensível.

Sua estrutura interna é “traduzida” pelas aplicações que a utilizam.

Exemplos:

.zip

.class

.rar

.bin

.exe

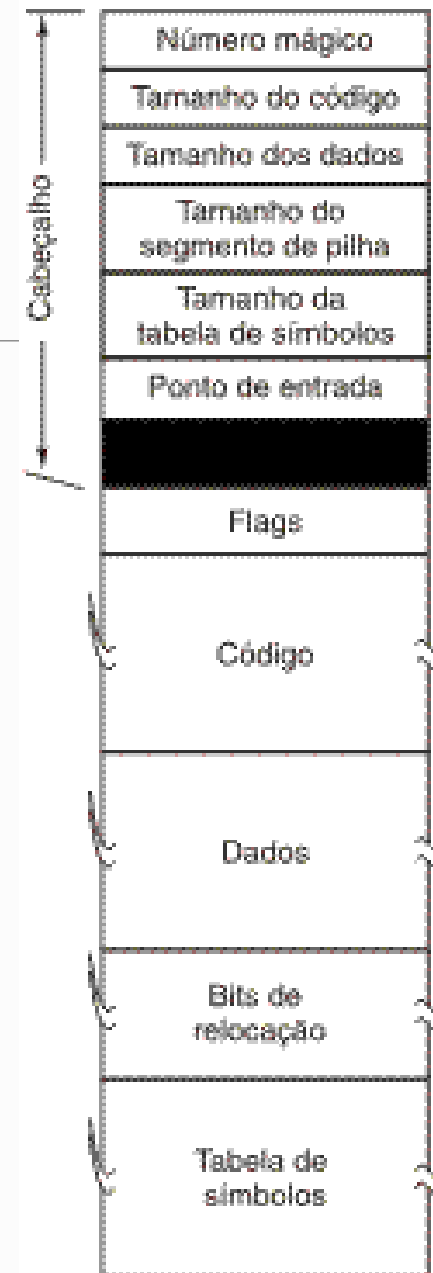
.dll

□□□%□\$»□□□□%\" data-bbox="431 510 922 852"/>

Tipos de Arquivos

Exemplo:

- Arquivo executável do UNIX.
 - Número mágico (evita que outros arquivos sejam executados por engano).
 - Outras informações de cabeçalho.
 - Texto do programa;
 - Etc.
- O S.O. é quem conhece a estrutura do arquivo.



Acesso dos Arquivos

Acesso aos Arquivos

Acesso sequencial

- O processo pode ler todos os bytes de um arquivo, sem efetuar saltos.
- Neste tipo de acesso, o processo pode voltar o cabeçote de leitura para o início do arquivo.
- Era conveniente quando o meio de armazenamento era a fita magnética.

Acesso aleatório

- O processo pode acessar as informações do arquivo fora de ordem (efetuando saltos).
- Mais adequado para discos rígidos.
- Geralmente os S.O.s disponibilizando a função seek.

Acesso aos Arquivos

Alguns sistemas operacionais antigos classificavam os arquivos como sendo de acesso sequencial ou de acesso aleatório.

Sistemas operacionais modernos consideram todos os arquivos como de acesso aleatório.

Isso influencia diretamente na programação.

Atributos de Arquivos

Atributo de Arquivos

Associado a cada arquivo, podemos ter:

- Data e Hora de criação.
- Data e hora do último acesso.
- Informações de proteção.
- Senha.
- Identificador do criador.
- Identificador do proprietário atual.
- Flag de apenas leitura.
- Flag de temporário.
- Etc.

Operações com Arquivos

Operações com Arquivos

Diferentes sistemas oferecem diferentes operações para armazenar e recuperar informações.

A seguir, as mais comuns:

- **Create:** Arquivo criado sem dados. Alguns de seus atributos são definidos.
- **Delete**
- **Open**
- **Close**
- **Read:** n bytes são lidos a partir da posição atual do cabeçote de leitura.

Operações com Arquivos

Diferentes sistemas oferecem diferentes operações para armazenar e recuperar informações.

A seguir, as mais comuns:

- **write:** Os bytes são escritos a partir da posição atual do cabeçalho de escrita. Write pode sobrescrever dados existentes; Ou escrever no fim do arquivo, aumentando seu tamanho.
- **Append:** apenas uma forma reescrita de write, mas com a limitação de escrever apenas no fim do arquivo.
- **Seek:** Efetua saltos nos cabeçotes de leitura e escrita.
- **Rename:** altera o nome do arquivo.

Próxima aula

Leitura:

Sistemas operacionais modernos

Sistemas de Arquivos

Referências

Sistemas Operacionais Modernos. Tanenbaum, A. S. 2ª edição. 2003.

Sistemas Operacionais. Conceitos e Aplicações. A. Silberschatz; P. Galvin; G. Gagne. 2000.

Sistemas Operacionais – Projeto e Implementação. Tanenbaum, A. S. 2ª edição. 2000.

Slides Prof. Humberto Brandão