

# Sistemas Operacionais

---

## Sistema de Arquivos Introdução

Aula 23

## Introdução

---

- Armazenamento permanente de grande volume de informações
  - Memória primária (RAM) é volátil e limitada em tamanho
    - Solução é memória secundária (discos, fitas, etc)
    - Acesso randômico ou sequencial a essas informações
- Sistema de arquivos é a parte do sistema operacional responsável pelo gerenciamento de arquivos e seus recursos
  - Composto de duas partes básicas
    - Arquivos: armazenamento de dados e programas
    - Diretórios: organização e informações sobre arquivos

## Funções básicas do sistema de arquivos

---

- Fornecer uma visão lógica para arquivos e diretórios
  - Esconder detalhes específicos dos dispositivos de memória secundária
- Uso eficiente dos dispositivos de armazenamento
- Oferecer compartilhamento de arquivos entre usuários e/ou aplicações (inclui aspectos de proteção)
- Dois pontos de vista:
  - Sistema operacional
  - Usuário

## Requisitos mínimos: ponto de vista do sistema

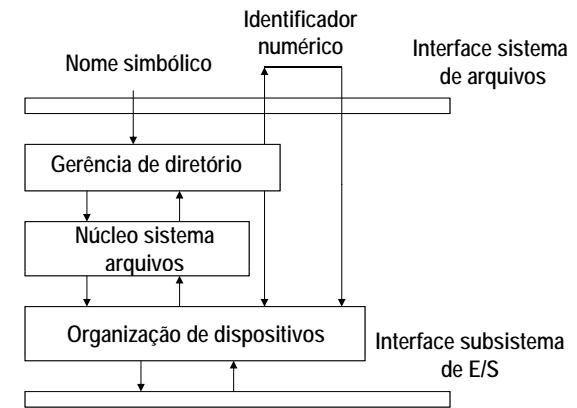
---

- O sistema operacional deve ser capaz:
  - Descrever a localização de todos os arquivos e de seus atributos
  - Gerenciar espaço físico do disco
    - Alocar blocos livres a arquivos em criação/expansão
    - Liberar blocos de arquivos removidos
    - Mecanismos para localizar eficientemente blocos (setores) que compõem arquivos

## Requisitos mínimos: ponto de vista do usuário

- Cada usuário deve ser capaz de:
  - Criar, apagar, ler e alterar arquivos
  - Controlar as permissões de acesso a seus arquivos
  - Nomear arquivos de forma simbólica
  - Organizar os arquivos de forma adequada a suas necessidades específicas
    - Criação de diretórios e subdiretórios
  - Realizar *back-ups* e recuperar arquivos em caso de problemas

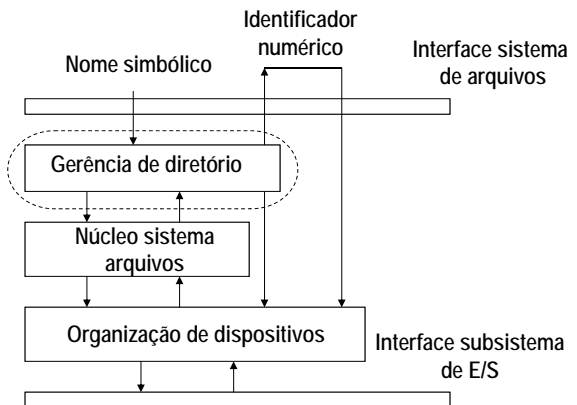
## Modelo hierárquico do sistema de arquivos



## Interface do sistema de arquivos

- Fornece ao usuário a abstração de arquivos e de diretórios
  - Diretórios são um tipo especial de arquivo
- Oferece primitivas para uso e manipulação de arquivos e diretórios
  - ex: `open()`, `read()`, `write()`, `close()`, `dir`, `rm`, `cd`, `mv`, etc
- Arquivos são identificados em dois níveis:
  - Usuários: através de nomes simbólicos (permanentes)
  - Sistema: através de identificadores numéricos (temporários)
- Criação de vínculos entre nomes simbólicos e identificadores numéricos

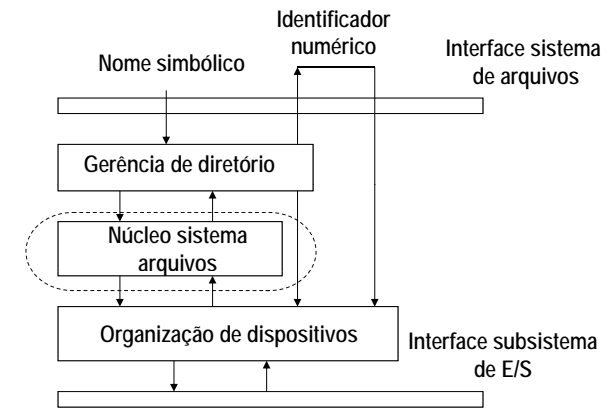
## Modelo hierárquico do sistema de arquivos



## Gerência de diretório

- Função básica é recuperar informações de um determinado arquivo a partir de seu nome simbólico
- Interage com o núcleo do sistema de arquivos para "abrir" o arquivo
  - Retorno é o identificador numérico do arquivo (*file pointer*)

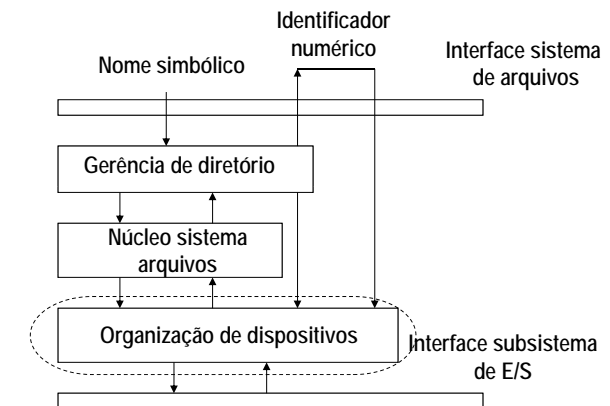
## Modelo hierárquico do sistema de arquivos



## Núcleo básico do sistema de arquivos

- Ativamento e desativamento de arquivos no sistema
  - Acionado através de rotinas `open()` e `close()`
  - Verificação dos direitos de acesso a arquivos
- Manutenção em memória de estruturas de dados que descrevem os arquivos
  - Descritores de arquivos
  - Tabela de arquivos abertos
- Associação entre o nome simbólico e o identificador numérico
  - Utilizado em subsequentes primitivas de manipulação de arquivos
    - ex: `fd = open("exemplo.txt", "rw")`

## Modelo hierárquico do sistema de arquivos



## Organização de dispositivos

---

- Mapeamento do arquivo em blocos lógicos
  - Organização do arquivo
- Converte operações em arquivos em conjuntos de requisições de blocos lógicos e repassa ao subsistema de E/S
- Alocação e liberação de blocos
  - Gerenciamento do espaço lógico em dispositivos

## Visão de usuário

---

- Arquivo é uma coleção de dados agrupados com objetivos de controle de acesso, leitura e modificação
- Logicamente representado por:
  - Nome simbólico
  - Tipo
  - Organização lógica
  - Atributos
  - Primitivas de operação

## Nome simbólico

---

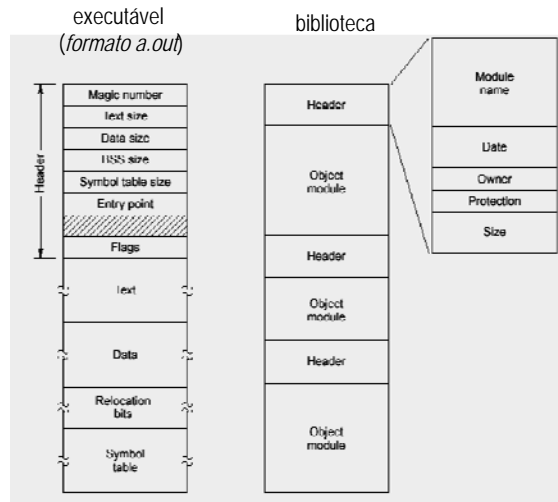
- Arquivo é identificado por um conjunto de caracteres
- Varia de sistema operacional a sistema operacional
  - Limitação de tamanho
  - Formato (nome + extensão + tamanho em caracteres )
  - Caracteres permitidos (case sensitive)
- Extensão: adendo ao nome do arquivo que pode indicar um tipo

## Tipos de arquivo

---

- Sistema operacional suporta vários tipos de arquivos
- Formato interno e semântica de arquivos
  - Determinam tipos de operações que podem ser feitas
  - Dependem do sistema operacional:
    - Mínimo: texto, executável, binário
    - Vinculado a aplicações (ex: windows - .ppt, .doc, .xls, etc)
- Tipos comuns:
  - Regular: arquivos de dados em ASCII e binário
  - Diretório: arquivos que mantêm a estrutura (organização) do sistema de arquivos
  - Arquivos especiais de caracter/bloco: vinculados a dispositivos de E/S

## Exemplos de arquivos UNIX



Sistemas Operacionais

17

## Organização lógica de arquivos

- O sistema de arquivos, no nível mais alto de abstração, enxerga:
  - Diretórios
  - Arquivos de dados
- Diretório
  - SO mantém uma estrutura própria (organização) visando facilitar a pesquisa e o gerenciamento de seu conteúdo
  - É um arquivo
- Arquivo de dados
  - SO oferece o mecanismo básico de manipulação de dados (*read* e *write*)
  - Sequência de registros lógicos que pode ser acessado por um conjunto de operações

Sistemas Operacionais

18

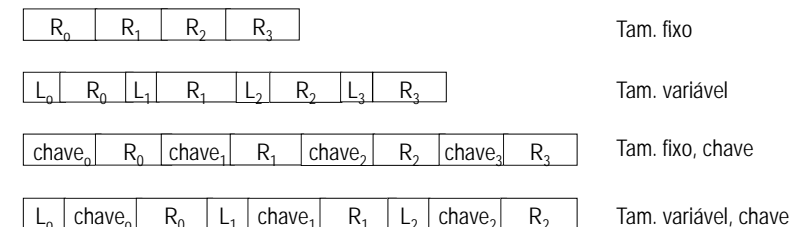
## Organização lógica de arquivos: registros

- Arquivo é um conjunto de registros acessados por operações específicas (métodos de acesso)
  - Registro é a menor unidade que pode ser lida ou escrita
- Registros podem ser:
  - Tamanho fixo (byte, palavra, estrutura própria)
  - Tamanho variável (estrutura que contém seu próprio tamanho)
- Registros são endereçados
  - Implícito: ponteiro interno mantido pelo SO
    - Acessado na ordem que aparecem no arquivo
    - Para acessar o registro  $i$  é necessário acessar os  $i-1$  anteriores
  - Explícito: ponteiro é especificado e visto "fora" do SO
    - Acesso randômico ou direto

Sistemas Operacionais

19

## Organização interna: tipos



- Acesso sequencial: fácil implementação em todos tipos
- Acesso randômico:
  - Fácil implementação nas organizações com tamanho fixo
  - Tamanho variável necessita ter uma estrutura auxiliar de índices (indexado)
  - Difícil encontrar suporte no próprio SO
    - Normalmente é dado por aplicativos (banco de dados, por ex.)
    - EXCEÇÃO: diretórios

Sistemas Operacionais

20

## Arquivos mapeados em memória

---

- Primitivas de leitura e escrita em arquivos difere das de memória
- Possível simplificar a programação usando as mesmas
  - Mapear arquivos na memória virtual, ou seja, incluir no espaço de endereçamento do processo
  - Acessar arquivo é equivalente a ler posições de memória
- Primitivas específicas para mapear arquivos em memória
  - Linux: *mmap()*
  - Windows: objetos seção

## Acesso a arquivos

---

- Sequencial
  - O acesso é sempre feito a partir do início do arquivo
- Randômico
  - O acesso pode ser feito a partir de qualquer posição do arquivo
    - Introduz a noção de uma operação de seek()

## Atributos

---

- Arquivo é composto por:
  - Nome
  - Dados
  - Atributos ou metadados (atribuídos pelo sistema operacional)
- Exemplos:
  - Proprietário, tamanho, histórico de uso, proteção, localização, arquivamento (backup)

## Primitivas de operações em arquivos

---

- Criação e remoção
- Abertura e fechamento
  - Arquivo deve ser “aberto” antes de sofrer qualquer operação
    - Alocação de estruturas de dados, buffers, identificadores, etc.
  - Fechamento libera todas as estruturas de dados, buffers, identificadores, etc
- Leitura e escrita
  - Transferência de dados arquivo/memória e vice-versa
  - Idéia de um ponteiro lógico
- Posicionamento e reposicionamento
  - *Seek*: posicionamento do ponteiro lógico para um registro / qualquer
  - *Rewind*: seek para o primeiro registro do arquivo
- Etc...

## Leituras complementares

---

- R. Oliveira, A. Carissimi, S. Toscani; Sistemas Operacionais. Editora Sagra-Luzzato, 2001.
  - Capítulo 8, seções 8.1, 8.2, e 8.3
- A. Silberchatz, P. Galvin; Operating System Concepts. Addison-Wesley, (4<sup>rd</sup> edition).
  - Capítulo 10, seções 10.1, 10.2, 10.4, e 10.5