

Sistemas Operacionais

SISTEMAS DE ARQUIVOS
MACHADO/MAIA: CAPÍTULO 11

Prof. Pedro Luís Antonelli
Anhanguera Educacional




SISTEMAS DE ARQUIVOS - INTRODUÇÃO

O armazenamento e a recuperação de informações é uma atividade essencial para qualquer tipo de aplicação.

Um processo deve ser capaz de ler e gravar de forma permanente grande volume de dados em dispositivos como fitas e discos, além de poder compartilhá-los com outros processos.

A maneira pela qual o sistema operacional estrutura e organiza estas informações é através da implementação de **arquivos**.




ARQUIVOS

Os arquivos são gerenciados pelo sistema operacional de maneira a facilitar o acesso dos usuários ao seu conteúdo.

A parte do sistema responsável por essa gerência é denominada **sistema de arquivos**.

O sistema de arquivos é a parte mais visível de um sistema operacional, pois a manipulação de arquivos é uma atividade frequentemente realizada pelos usuários, devendo sempre ocorrer de maneira uniforme, independente dos diferentes dispositivos de armazenamento.




ARQUIVOS

Um arquivo é constituído por **informações logicamente relacionadas**.

Estas informações podem representar **instruções** ou **dados**.

Um arquivo executável, por exemplo, contém instruções compreendidas pelo processador, enquanto um arquivo de dados pode ser estruturado livremente como um arquivo texto ou de forma mais rígida como um banco de dados relacional.



ARQUIVO - CONCEITO

Na realidade, um arquivo é um **conjunto de registros definidos pelo sistema de arquivos**, tornando seu conceito **abstrato e generalista**.


A partir dessa definição, o conteúdo do arquivo pode ser manipulado seguindo conceitos preestabelecidos.



ARQUIVOS - ARMAZENAMENTO

Os arquivos são armazenados pelo sistema operacional em diferentes dispositivos físicos, como fitas magnéticas, discos magnéticos e discos ópticos.

O **tipo de dispositivo** no qual o arquivo é armazenado deve ser **isolado pelo sistema operacional**, de forma que exista uma independência entre os arquivos a serem manipulados e o meio de armazenamento




ARQUIVOS - IDENTIFICAÇÃO

Um arquivo é identificado por um **nome**, composto por uma sequência de caracteres.

Em alguns sistemas de arquivos é feita distinção entre caracteres alfabéticos maiúsculos e minúsculos (“**sensível ao caso**”).

Regras como extensão máxima do nome e quais são os caracteres válidos também podem variar de um sistema de arquivos para outro.




ARQUIVOS - IDENTIFICAÇÃO

Em alguns sistemas operacionais, a identificação de um arquivo é composta por duas partes separadas com um ponto.

A parte após o ponto é denominada extensão do arquivo e tem como finalidade identificar o conteúdo do arquivo.

Assim é possível convencionar que uma extensão **TXT** identifica um **arquivo texto**, enquanto **EXE** indica um **arquivo executável**.



ARQUIVOS - IDENTIFICAÇÃO

Na tabela abaixo, são apresentadas algumas extensões de arquivos:


<i>Extensão</i>	<i>Descrição</i>
<i>ARQUIVO. BAS</i>	<i>Arquivo fonte em BASIC</i>
<i>ARQUIVO. COB</i>	<i>Arquivo fonte em COBOL</i>
<i>ARQUIVO. EXE</i>	<i>Arquivo executável</i>
<i>ARQUIVO. OBJ</i>	<i>Arquivo objeto</i>
<i>ARQUIVO. TXT</i>	<i>Arquivo texto</i>
<i>ARQUIVO. PAS</i>	<i>Arquivo fonte em PASCAL</i>

ARQUIVOS - ORGANIZAÇÃO

A organização de arquivos consiste em como os seus dados estão internamente armazenados.

A estrutura dos dados pode variar em função do tipo de informação contida no arquivo.

Arquivos texto possuem propósitos completamente distintos de **arquivos executáveis**, conseqüentemente, estruturas diferentes podem adequar-se melhor a um tipo do que a outro.



ARQUIVOS - ORGANIZAÇÃO


No momento da criação de um arquivo, seu criador pode definir qual a organização adotada.

Esta organização pode ser uma estrutura suportada pelo sistema operacional ou definida pela **própria aplicação**.



ARQUIVOS - ORGANIZAÇÃO

A grande vantagem deste modelo é a **flexibilidade** para criar **diferentes estruturas de dados**, porém todo o controle de acesso ao arquivo é de inteira responsabilidade da **aplicação**.




ARQUIVOS - ORGANIZAÇÃO

Alguns sistemas operacionais possuem diferentes organizações de arquivos.

Neste caso, cada arquivo criado deve seguir um modelo **suportado** pelo sistema de arquivos.

As organizações mais conhecidas e implementadas são a **sequencial**, **relativa** e **indexada**.



ARQUIVOS - ORGANIZAÇÃO

A forma mais simples de organização de arquivos é através de uma **sequência não-estruturada de bytes**.

Neste tipo de organização, o sistema de arquivos não impõe nenhuma estrutura **lógica para os dados**.

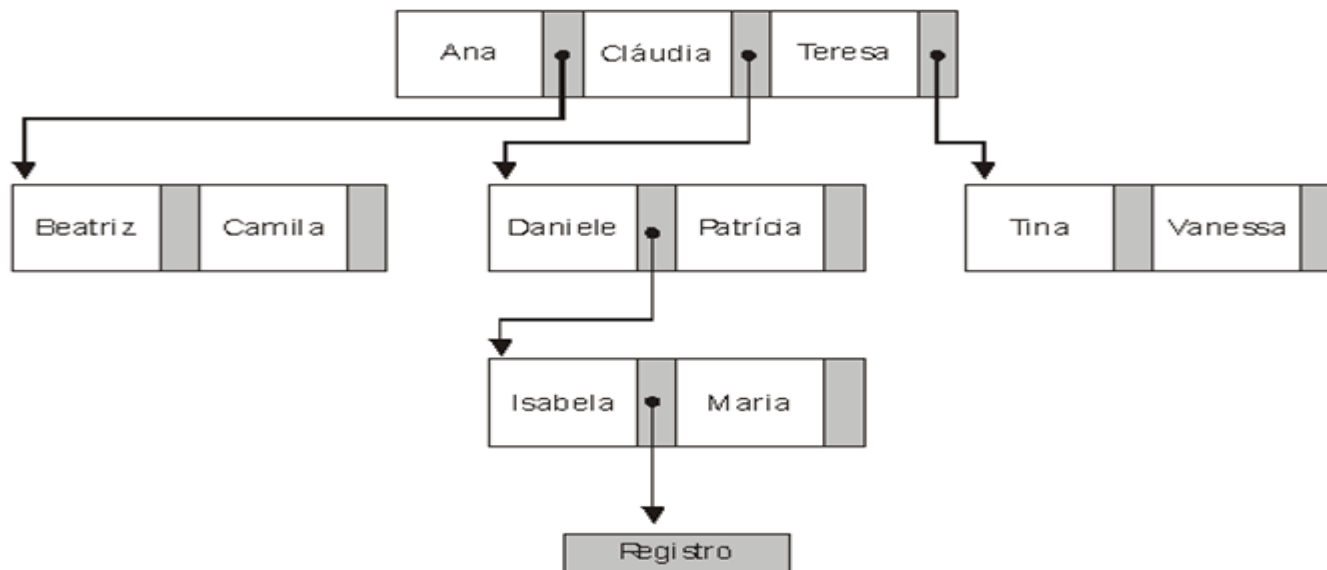
A aplicação deve definir toda a organização, estando livre para estabelecer seus próprios critérios.



ARQUIVOS - ORGANIZAÇÃO

Nestes tipos de organização, podemos visualizar um arquivo como um conjunto de registros.

Os registros podem ser classificados em registros de **tamanho** fixo, quando possuírem sempre o mesmo tamanho, ou registros de tamanho **variável**.




ARQUIVOS – MÉTODO DE ACESSO

Em função de como o arquivo está organizado, o sistema de arquivos pode recuperar registros de diferentes maneiras.

Inicialmente, os primeiros sistemas operacionais só armazenavam arquivos em fitas magnéticas.

Com isso, o acesso era restrito a leitura dos registros na **ordem em que eram gravados**, e a gravação de novos registros só era possível no **final do arquivo**.

Este tipo de acesso, chamado de **acesso sequencial**, era próprio da fita magnética, que, como meio de armazenamento, possuía esta limitação.




ARQUIVOS – MÉTODO DE ACESSO

Em função de como o arquivo está organizado, o sistema de arquivos pode recuperar registros de diferentes maneiras.

Inicialmente, os primeiros sistemas operacionais só armazenavam arquivos em fitas magnéticas.

Com isso, o acesso era restrito a leitura dos registros na **ordem em que eram gravados**, e a gravação de novos registros só era possível no **final do arquivo**.

Este tipo de acesso, chamado de **acesso sequencial**, era próprio da fita magnética, que, como meio de armazenamento, possuía esta limitação.

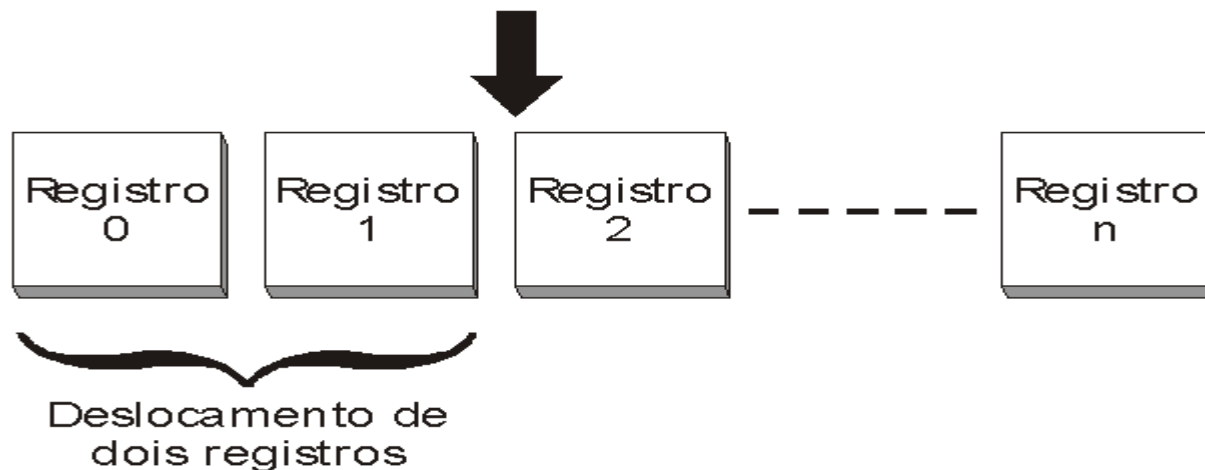


ARQUIVOS – MÉTODO DE ACESSO

Com o advento dos discos magnéticos, foi possível a introdução de métodos de acesso mais eficientes.

O primeiro a surgir foi o **acesso direto**, que permite a **leitura / gravação** de um registro diretamente na sua posição.

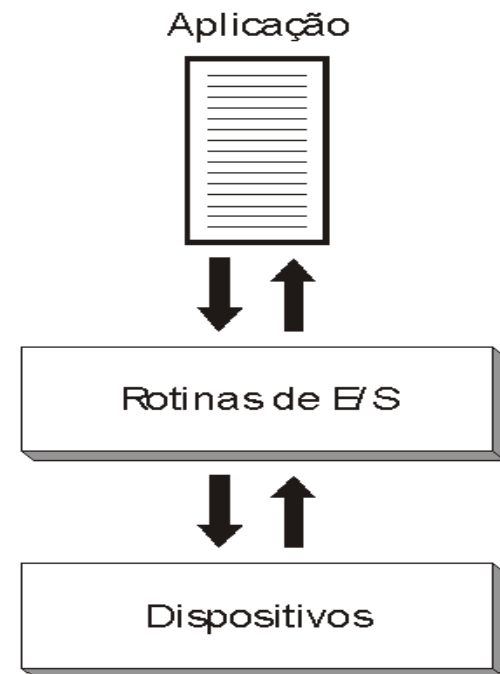
Este método é realizado através do **número do registro**, que é a sua posição relativa ao início do arquivo.



OPERAÇÕES DE ACESSO AO ARQUIVO

O sistema de arquivos disponibiliza um **conjunto de rotinas** que permite às aplicações realizarem operações de **E/S**, como:

- tradução de nomes em endereços;
- leitura e gravação de dados;
- criação/eliminação de arquivos.



Na realidade, as rotinas de E/S têm como função disponibilizar uma **interface simples** e uniforme entre a aplicação e os diversos dispositivos.

OPERAÇÕES DE ACESSO AO ARQUIVO

A tabela abaixo apresenta algumas destas rotinas encontradas na maioria das implementações de sistemas de arquivos:

<i>Rotina</i>	<i>Descrição</i>
<i>CREATE</i>	<i>Criação de arquivo</i>
<i>OPEN</i>	<i>Abertura de um arquivo</i>
<i>READ</i>	<i>Leitura de um arquivo</i>
<i>WRITE</i>	<i>Gravação em um arquivo</i>
<i>CLOSE</i>	<i>Fechamento de um arquivo</i>
<i>DELETE</i>	<i>Eliminação de um arquivo</i>


ARQUIVOS - ATRIBUTOS

Cada arquivo possui **informações de controle** denominadas **Atributos**.

Os atributos variam dependendo do sistema de arquivos, porém alguns, como **tamanho do arquivo**, **proteção**, **identificação do criador** e **data de criação** estão presentes em quase todos os sistemas.

Alguns atributos especificados na criação do arquivo **não podem** ser modificados em função de sua própria natureza, como **organização e data/hora de criação**.

Outros são alterados pelo próprio **sistema operacional**, como tamanho e data/hora do último backup realizado.



ARQUIVOS - ATRIBUTOS

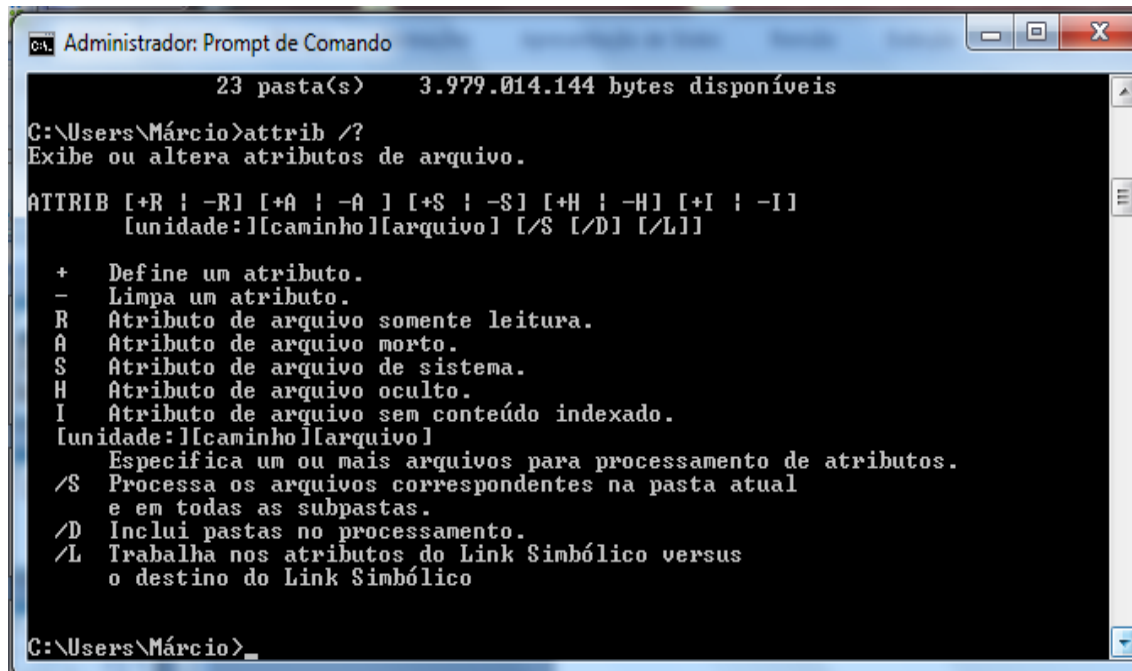
Existem ainda atributos que podem ser modificados pelo próprio **usuário**, como **proteção do arquivo**, tamanho máximo e senha de acesso.

Na tabela abaixo, são apresentados os principais atributos presentes nos sistemas de arquivos:

<i>Atributos</i>	<i>Descrição</i>
<i>Tamanho</i>	<i>Especifica o tamanho do arquivo</i>
<i>Proteção</i>	<i>Código de proteção de acesso</i>
<i>Proprietário</i>	<i>Identifica o criador do arquivo</i>
<i>Criação</i>	<i>Data e hora de criação do arquivo</i>
<i>Senha</i>	<i>Senha necessária para acessar o arquivo</i>
<i>Organização</i>	<i>Indica a organização lógica dos registros</i>
<i>Backup</i>	<i>Data e hora do último backup realizado</i>

ARQUIVOS - ATRIBUTOS

- No **Windows** – **Propriedades** ou **MSDOS** comando **ATTRIB**
- **ATTRIB +R TEXTO1.TXT**
 - R – leitura
 - H – oculto
 - A – leitura/gravação
 - S – arquivo de sistema



```
CA. Administrador: Prompt de Comando

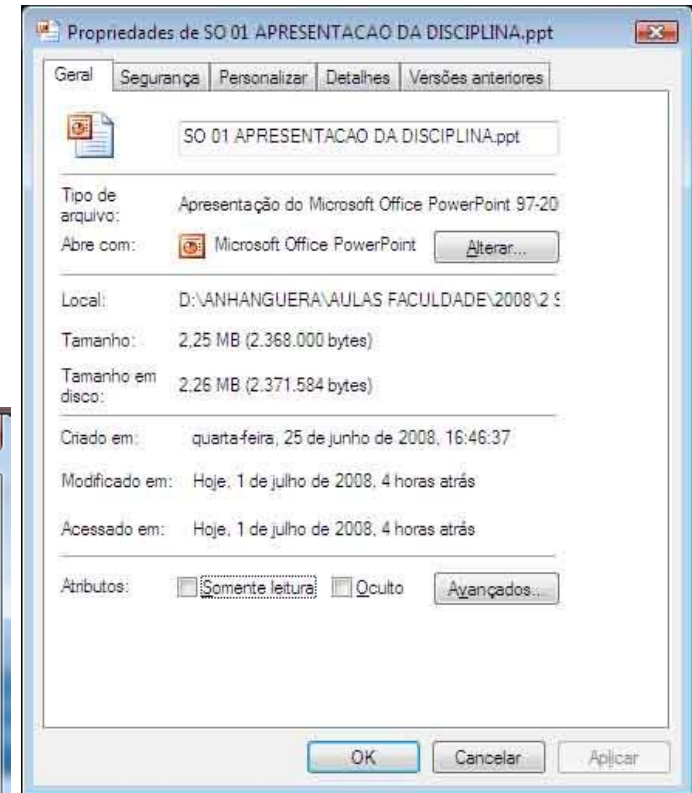
                23 pasta(s)      3.979.014.144 bytes disponíveis

C:\Users\Márcio>attrib /?
Exibe ou altera atributos de arquivo.

ATTRIB [+R | -R] [+A | -A ] [+S | -S] [+H | -H] [+I | -I]
        [unidade:] [caminho] [arquivo] [/S [/D] [/L]]

+   Define um atributo.
-   Limpa um atributo.
R   Atributo de arquivo somente leitura.
A   Atributo de arquivo morto.
S   Atributo de arquivo de sistema.
H   Atributo de arquivo oculto.
I   Atributo de arquivo sem conteúdo indexado.
unidade: [caminho] [arquivo]
    Especifica um ou mais arquivos para processamento de atributos.
/S   Processa os arquivos correspondentes na pasta atual
     e em todas as subpastas.
/D   Inclui pastas no processamento.
/L   Trabalha nos atributos do Link Simbólico versus
     o destino do Link Simbólico

C:\Users\Márcio>_
```



ARQUIVOS - ATRIBUTOS

- No Linux – **Propriedades** e comando **CHMODE**

- `chmod 750 texto1`

- 0 – nenhuma permissão
- 1 – execução
- 2 – escrita
- 3 – execução e escrita
- 4 – leitura
- 5 – execução e leitura
- 6 – leitura e escrita
- 7 – leitura, escrita e execução

- `chmod a+x texto1`

- r – leitura u – usuário
- w – escrita g – grupo
- x – execução a – todos (*all*)



DIRETÓRIOS

A **estrutura de diretórios** é como o sistema organiza **logicamente** os diversos arquivos contidos em um disco.

O diretório é uma **estrutura de dados** que contém entradas associadas aos arquivos onde cada entrada armazena informações como localização física, nome, organização e demais atributos.

Quando um arquivo é aberto, o sistema operacional procura a sua **entrada na estrutura de diretórios**, armazenando as informações sobre atributos e localização do arquivo em uma **tabela** mantida na memória principal.

DIRETÓRIOS

Esta tabela contém todos **os arquivos abertos**, sendo fundamental para **aumentar o desempenho** das operações com arquivos.

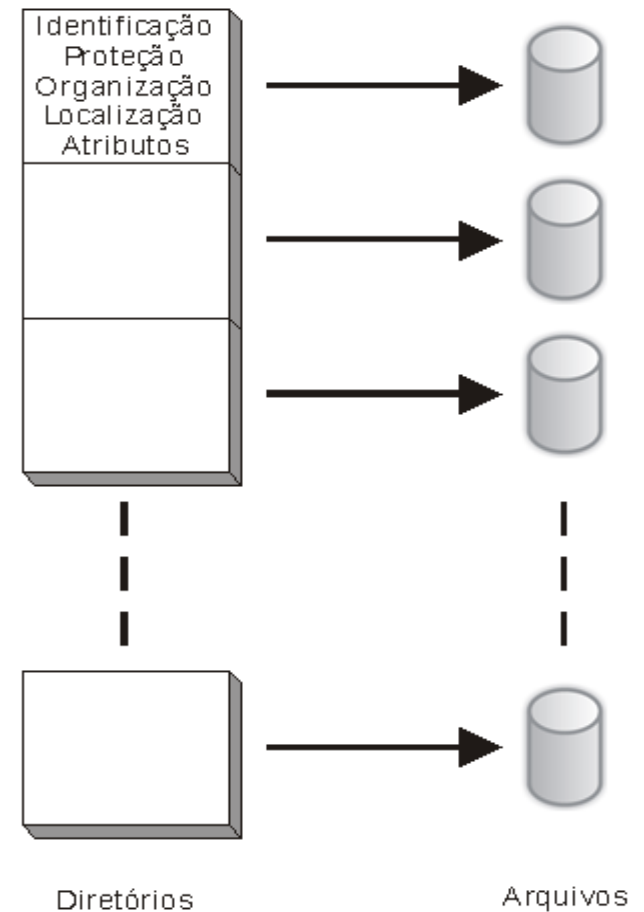
É importante que ao término do uso de arquivos, este seja fechado, ou seja, que se libere o espaço na tabela de arquivos abertos.

DIRETÓRIOS

A implementação mais simples de uma estrutura de diretórios é chamado de **nível único** (**single-level directory**).

Neste caso, somente existe um único diretório contendo todos os arquivos do disco.

Este modelo é bastante limitado, já que não permite que usuários criem arquivos com o mesmo nome, o que ocasionaria um conflito no acesso a os arquivos.



DIRETÓRIOS

Como o sistema de nível único é bastante limitado, uma evolução do modelo foi a implementação de uma estrutura onde para cada usuário existiria um diretório particular denominado **User File Directory (UFD)**.

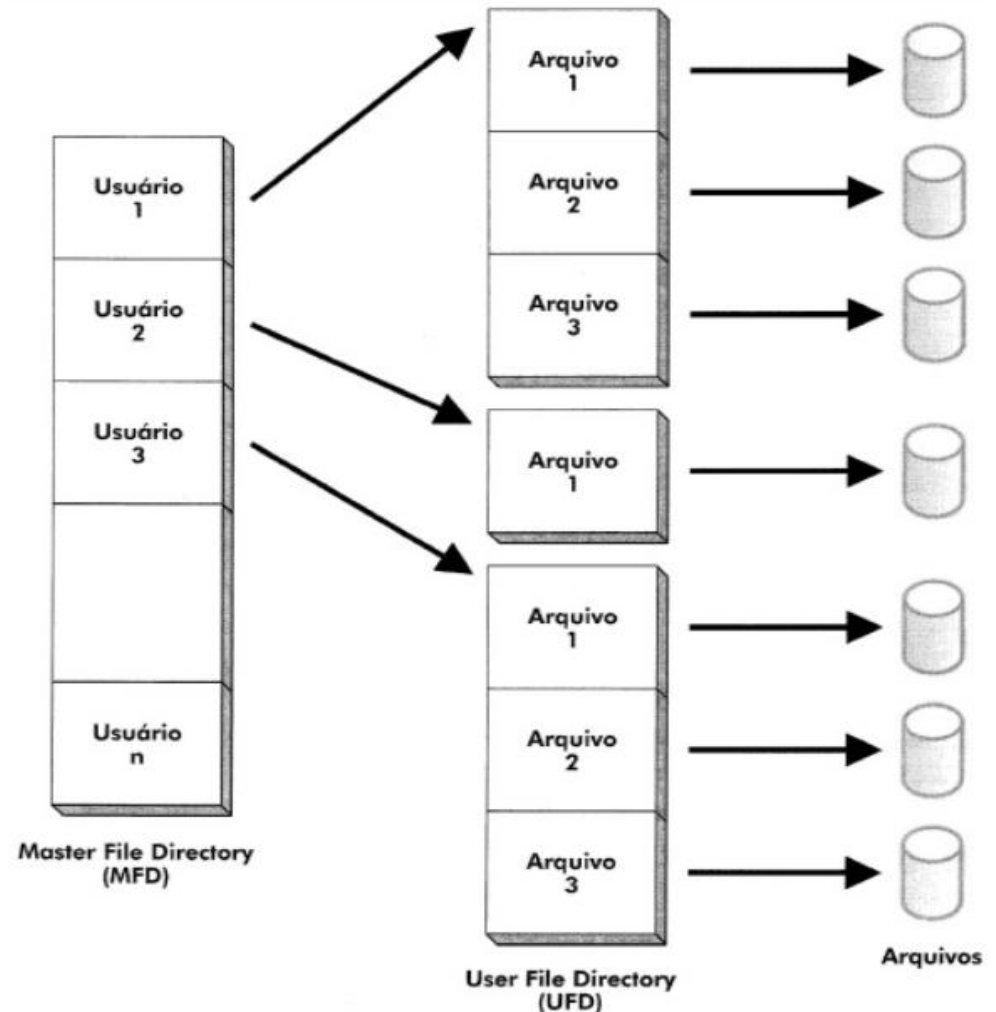
Com esta implementação, cada usuário passa a poder criar arquivos com qualquer nome, sem a preocupação de conhecer os demais arquivos do disco.

Para que o sistema possa localizar arquivos nessa estrutura, deve haver um nível de diretório adicional para controlar os diretórios individuais dos usuários.

DIRETÓRIOS

Este nível denominado **Master File Directory (MFD)**, é indexado pelo nome do usuário, onde cada entrada aponta para o diretório pessoal.

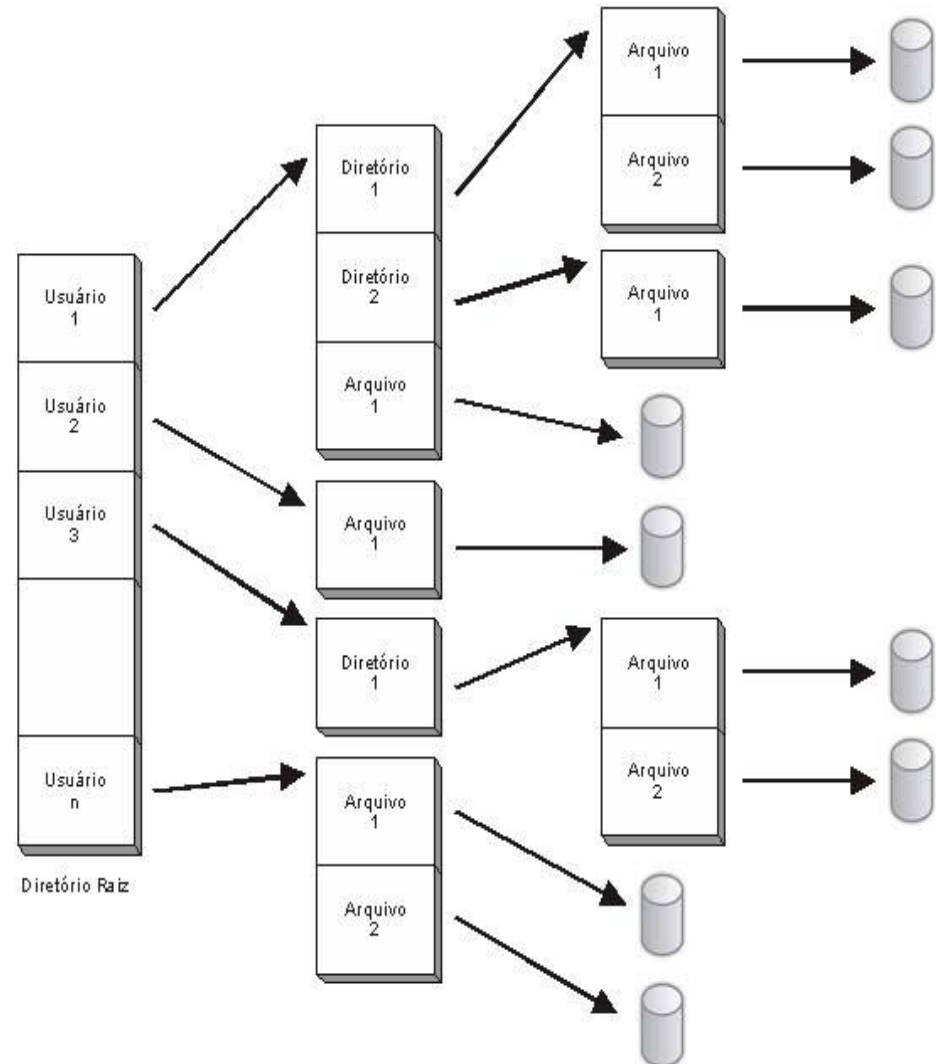
No exemplo um sistema com dois níveis.



DIRETÓRIOS

A extensão do modelo de dois níveis para um de múltiplos níveis permitiu que os arquivos fossem logicamente melhor organizados.

Este novo modelo, chamado estrutura de diretórios em árvore (**tree-structured directory**), é adotado pela maioria dos sistemas.



DIRETÓRIOS

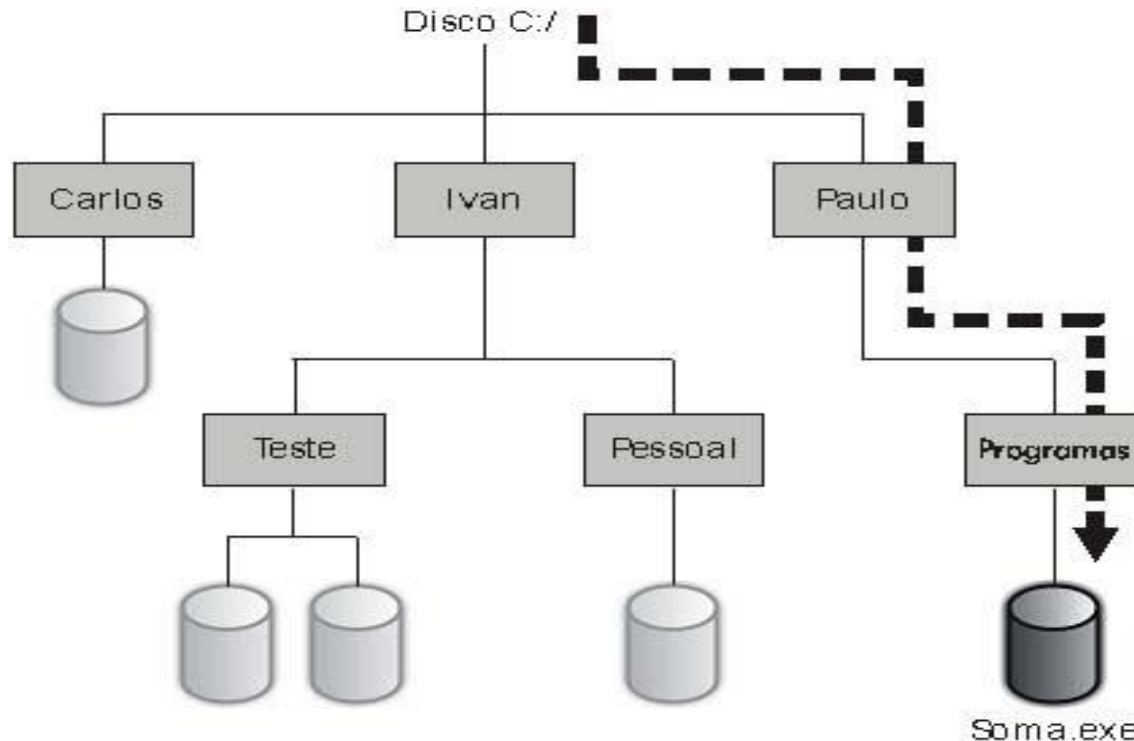
Na **estrutura em árvore**, cada usuário pode criar diversos **níveis de diretórios**, também chamados de **subdiretórios**.

Cada diretório pode conter arquivos ou outros diretórios.

O número de níveis de uma estrutura em árvore é dependente do sistema de arquivos de cada sistema operacional.

DIRETÓRIOS

Um arquivo, nesta estrutura em árvore, pode ser especificado unicamente através de um **path absoluto**, descrevendo todos os diretórios percorridos a partir da raiz (MFD) até o diretório no qual o arquivo está ligado.




GERÊNCIA DE ESPAÇOS LIVRES

A criação de arquivos em disco exige que o sistema operacional tenha o controle de quais **áreas ou blocos no disco estão livres**.

Este controle é realizado utilizando-se alguma estrutura de dados que armazena informações que possibilitam ao sistema de arquivos gerenciar o espaço livre do disco.

Nesta estrutura, geralmente uma **lista** ou **tabela**, é possível identificar blocos livres que poderão ser alocados a um novo arquivo.



GERÊNCIA DE ESPAÇOS LIVRES

A forma mais simples de implementar uma estrutura de espaços livres é através de uma **tabela** denominada **mapa de bits (bit map)**.

Cada entrada na tabela é associada a um bloco do disco representado por um bit, podendo assumir valor igual a **0** (indicando bloco livre) ou **1** (indicando bloco alocado).

Na figura, podemos observar um exemplo desta implementação que apresenta como principal problema um excessivo gasto de memória, já que para cada bloco do disco deve existir uma entrada na tabela.

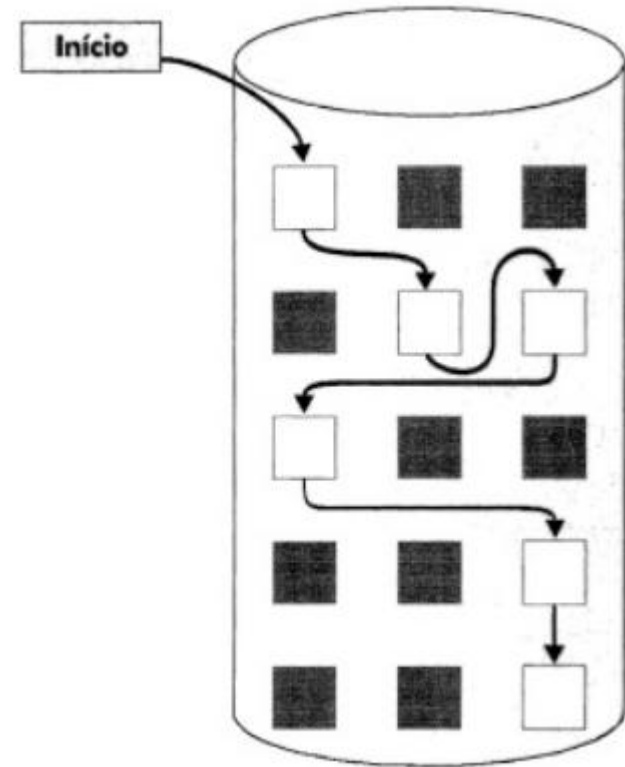
11001101
10000111
01110100
⋮
11100000

GERÊNCIA DE ESPAÇOS LIVRES

Uma segunda maneira de realizar este controle é com uma estrutura de **lista encadeada de todos os blocos** livres do disco.

Para que isto seja possível, cada bloco possui uma área reservada para armazenamento do endereço do próximo bloco.

A partir do primeiro bloco livre é, então, possível o acesso sequencial aos demais de forma encadeada.



GERÊNCIA DE ESPAÇOS LIVRES

Uma outra solução leva em consideração que blocos contíguos são geralmente alocados ou liberados simultaneamente.

Podemos, desta forma, enxergar o disco como um conjunto de segmentos de blocos livres.

Com base neste conceito, é possível manter uma tabela com o endereço do **primeiro bloco de cada segmento** e o **número de blocos livres contíguos** que se seguem.

Esta técnica de gerência de espaço livre é conhecida como **tabela de blocos livres**.

Bloco	Contador
4	2
10	1
13	7
25	20
50	5

Gerência de alocação em disco

Da mesma forma que o sistema operacional gerencia os **espaços livres** no disco, a gerência dos **espaços alocados** aos arquivos é de fundamental importância em um sistema de arquivos.

A seguir as principais **técnicas** são apresentadas:

- Alocação **Contígua**;
- Alocação **Encadeada**;
- Alocação **Indexada**.

Alocação Contígua

A alocação **contígua** consiste em armazenar um arquivo em blocos **sequencialmente dispostos** no disco.

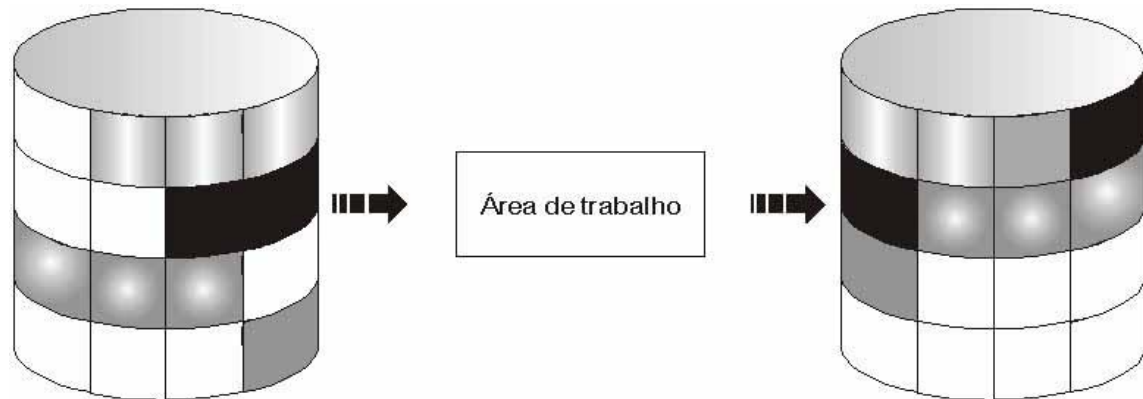
Algoritmos utilizados:

First-fit – primeiro segmento livre com tamanho suficiente para alocar o arquivo é selecionado;

Best-fit – menor segmento livre disponível com tamanho suficiente é selecionado

Worst-fit – maior segmento é selecionado

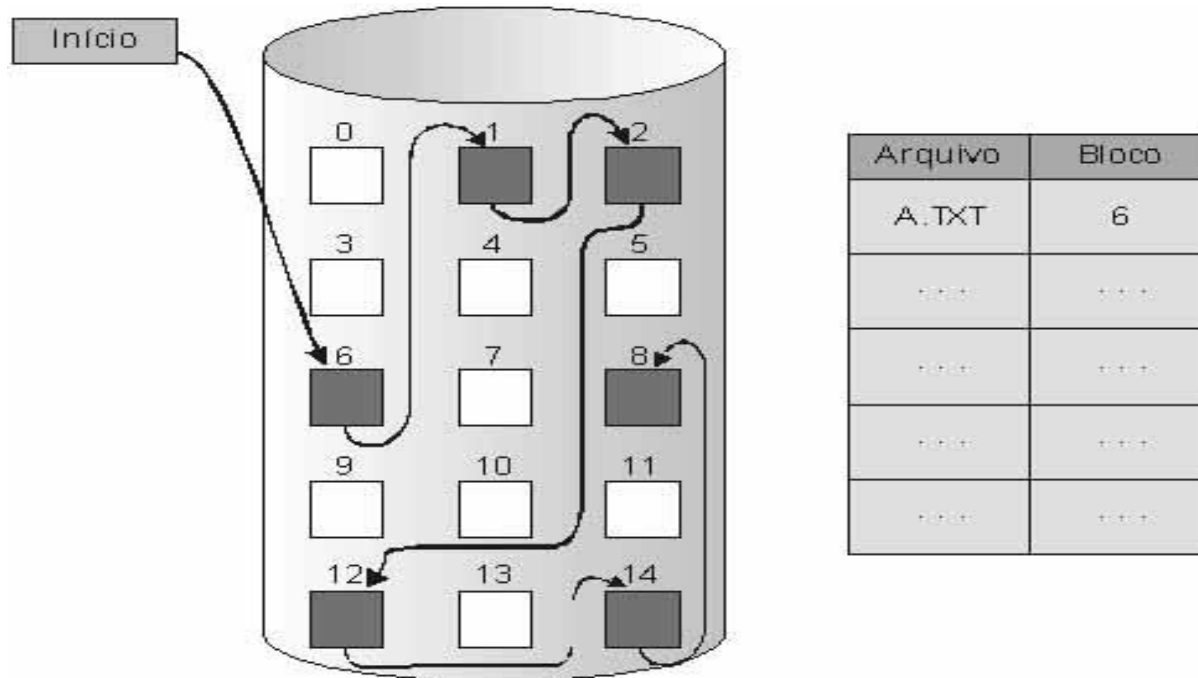
Independente do modelo, existe um problema que é a **fragmentação**.



Alocação Encadeada

Na alocação **encadeada**, um arquivo pode ser organizado como um conjunto de blocos **ligados logicamente** no disco, independente da sua localização física.

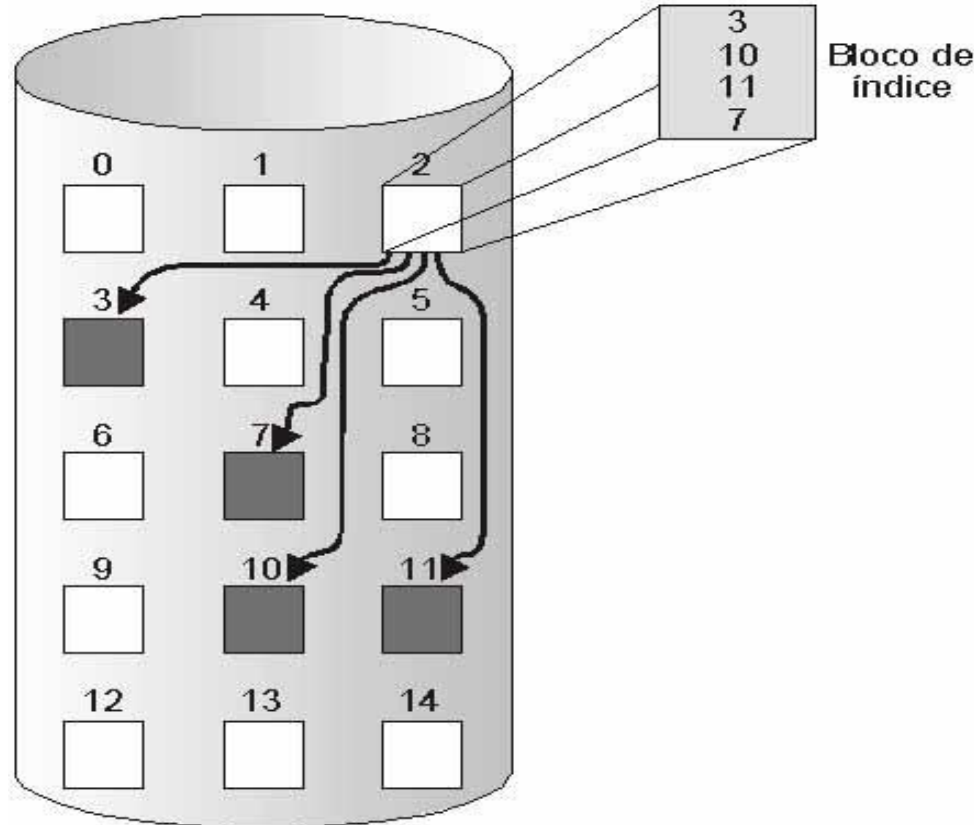
Cada bloco deve possuir um **ponteiro para o bloco seguinte** do arquivo e assim sucessivamente.



Alocação Indexada

A alocação **indexada** soluciona uma das principais limitações da alocação encadeada, que é a impossibilidade do acesso direto aos blocos dos arquivos.

O princípio desta técnica é manter os **ponteiros de todos os blocos** do arquivo em uma única estrutura denominada **bloco de índice**.




Proteção de Acesso

Considerando que os meios de armazenamento são **compartilhados entre diversos usuários**, é de fundamental importância que mecanismos de proteção sejam implementados para garantir a proteção individual de arquivos e diretórios.

Qualquer sistema de arquivos deve possuir mecanismos próprios para proteger o acesso às informações gravadas em discos e fitas, além de possibilitar o compartilhamento de arquivos entre usuários, quando desejado.

Em geral, o tipo de acesso a arquivos é implementado mediante a **concessão ou não** dos diferentes acessos que podem ser realizados, como **leitura** (read), **gravação** (write), **execução** (execute) e **eliminação** (delete).




Proteção de Acesso – Senha de Acesso

A associação de uma **senha de acesso** a um arquivo é um princípio bastante simples.

O controle de acesso se resume ao usuário ter o conhecimento da senha e, conseqüentemente, ter a **liberação do acesso** ao arquivo **concedida pelo sistema**.

Como cada arquivo possui apenas uma senha, o acesso é liberado ou não na sua totalidade, isto é, não é possível determinar quais tipos de operação podem ou não ser concedidas.

Outra desvantagem deste método é a **dificuldade de compartilhamento de arquivos**, pois, além do dono do arquivo, todos os demais usuários teriam que conhecer a senha de acesso.



Proteção de Acesso – Grupos de usuários

A proteção baseada em **grupos de usuários** é implementada por diversos sistemas operacionais.

Este tipo de proteção tem como princípio a **associação** de cada **usuário** do sistema a um **grupo**.

Os grupos de usuários são organizados logicamente com o objetivo de compartilhar arquivos e diretórios.

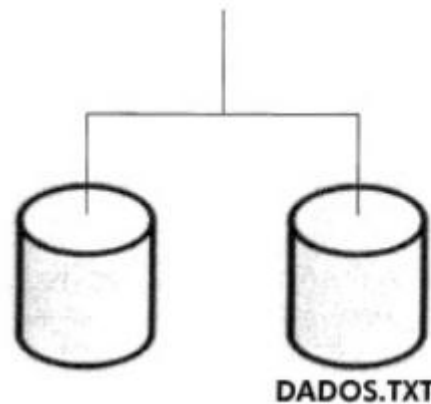
Esse mecanismo implementa três níveis de proteção ao arquivo: owner (**dono**), group(**grupo**) e all (**todos**).

Proteção de Acesso – Grupos de usuários

Na **criação do arquivo**, o usuário especifica se o arquivo deve ser acessado somente pelo seu criador, pelos usuários do grupo ao qual ele pertence ou por todos os usuários do sistema.

Nessa especificação é necessário **associar o tipo de acesso** (leitura, escrita, execução e eliminação) **aos três níveis de proteção**.

Em geral, somente o dono ou usuários privilegiados é que podem **modificar a proteção** dos arquivos.



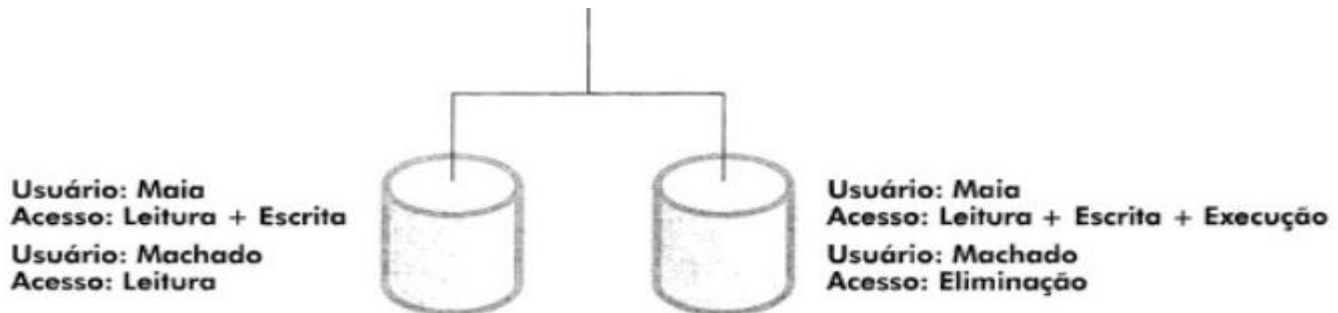
Nível de Proteção	Tipo de Acesso
Owner	Leitura Escrita Execução Eliminação
Group	Leitura
All	—

Proteção de Acesso – Lista de Acesso

A **Lista de Controle de Acesso** (Access Control List – ACL) consiste em uma lista associada a cada arquivo, onde são especificados quais os usuários e os tipos de acesso permitidos.

Nesse caso, quando um usuário tenta acessar um arquivo, o sistema operacional **verifica se a lista de controle** autoriza a operação desejada.

O tamanho dessa estrutura de dados pode ser bastante extenso se considerarmos que um arquivo pode ter seu acesso compartilhado por diversos usuários.



Implementação de Caches

O acesso a disco é bastante lento se comparado ao acesso à memória principal, devido à arquitetura dos discos magnéticos ou óticos).

Este é o principal motivo das operações de E/S com discos serem um **problema** para o **desempenho** do sistema.

Com o objetivo de minimizar este problema, a maioria dos sistemas de arquivos implementa uma técnica denominada **buffer cache**.


Implementação de Caches

Neste esquema, o sistema operacional reserva uma área da memória para que se tornem disponíveis “**cache**” utilizados em operações de acesso ao disco.

Quando uma operação é realizada, seja leitura ou gravação, o sistema verifica se a informação desejada se encontra no “**buffer cache**”. Em caso positivo, **não é necessário o acesso ao disco.**

Caso o bloco requisitado **não se encontre no “cache”**, a operação de E/S é realizada e o “**cache**” é atualizado.

Como existe uma limitação no tamanho do “**cache**”, cada sistema adota políticas para substituição de blocos.





BIBLIOGRAFIA

- **MACHADO, F. B. & MAIA, L. P., Arquitetura de Sistemas Operacionais, 4 Edição, São Paulo, LTC, 2007.**
 - **TANENBAUM, A. S. Sistemas Operacionais Modernos: 2ª edição, São Paulo, editora Prentice Hall, 2003.**
 - **SILBERSCHATZ, A. Sistemas Operacionais – Conceitos: São Paulo, editora LTC, 2004.**
 - **FERREIRA, R. E. Linux: Guia do Administrador do Sistema, Novatec Editora, São paulo, 2003.**
- 