

# Sistemas Operacionais

4º período

Professora: Michelle Hanne

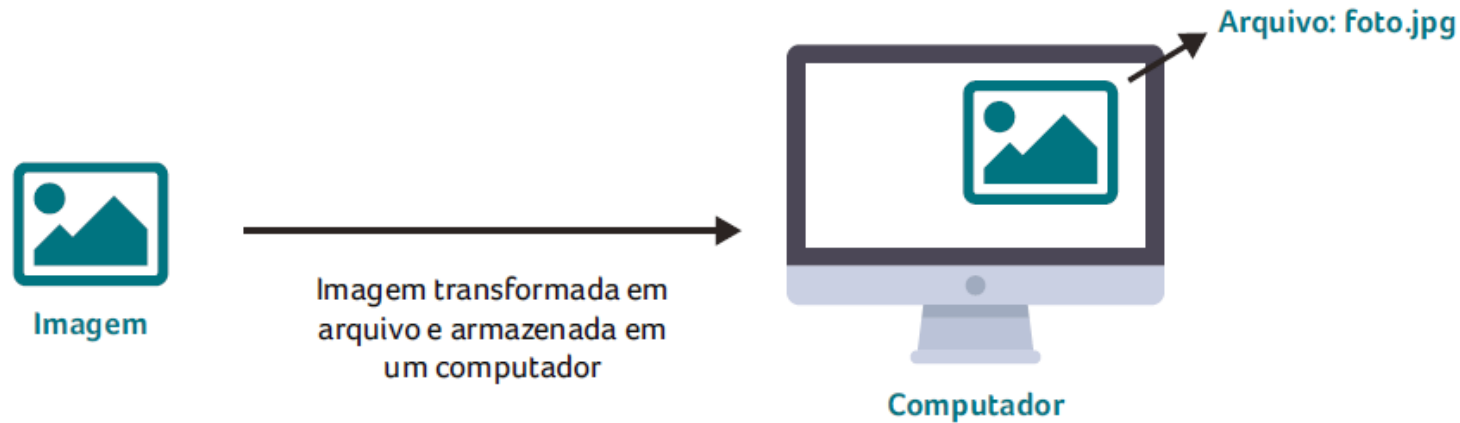
# Sistema de Arquivo

Uma das responsabilidades do Sistema Operacional é o Sistema de Arquivos. O Sistema de Arquivos gerencia operações de: gravação, alteração, acesso, exclusão entre outras operações como Segurança.

**O conhecimento de como o sistema operacional trata os arquivos internamente com relação à unidade de armazenamento aprimora o conhecimento de otimização de disco, evitando que espaços não utilizados sejam minimizados e melhorando o desempenho das unidades em operações, como leitura.**

# Arquivos

- O computador processa os arquivos existentes (TANENBAUM, 2016). O arquivo é armazenado em um dispositivo de armazenamento durável, como discos rígidos, CDs, etc



Um **arquivo** é um **mecanismo de abstração**, pois deve fornecer uma transparência ao usuário, de forma que operações, como criação, armazenamento, exclusão, cópia, entre outros, possam ser realizadas sem a necessidade de compreensão dos detalhes de como o sistema operacional trata esses arquivos.

# Nomeação de Arquivos

- As regras usadas para se nomear um arquivo podem mudar de sistema operacional para sistema operacional, mas hoje basicamente utilizam um conjunto de caracteres até oito letras (TANENBAUM, 2016). **Sistemas gráficos possuem recursos que permitem escrever nomes de arquivo até 255 caracteres**, caso do Linux e Windows

|            |       |       |                                   |  |
|------------|-------|-------|-----------------------------------|--|
| 22/09/2018 | 11:25 | <DIR> | .                                 |  |
| 22/09/2018 | 11:25 | <DIR> | ..                                |  |
| 24/04/2018 | 10:20 | <DIR> | .astah                            |  |
| 05/07/2018 | 12:35 | <DIR> | .eclipse                          |  |
| 18/09/2018 | 09:33 | <DIR> | .p2                               |  |
| 04/07/2018 | 17:15 | <DIR> | .tooling                          |  |
| 18/09/2018 | 21:53 | <DIR> | .VirtualBox                       |  |
| 15/08/2018 | 23:02 | <DIR> | 3D Objects                        |  |
| 22/09/2018 | 11:23 | <DIR> | arquivo                           |  |
| 22/09/2018 | 11:24 |       | 1.547 arquivo.txt                 |  |
| 22/09/2018 | 11:23 |       | 1.502 arquivo                     |  |
| 15/08/2018 | 23:02 | <DIR> | Contacts                          |  |
| 22/09/2018 | 11:25 |       | 1.612 contrato de seguradoras.txt |  |
| 20/09/2018 | 16:19 | <DIR> | Desktop                           |  |
| 22/09/2018 | 11:25 |       | 1.657 doc.txt                     |  |
| 11/09/2018 | 12:40 | <DIR> | Documents                         |  |
| 21/09/2018 | 10:14 | <DIR> | Downloads                         |  |
| 04/07/2018 | 17:07 | <DIR> | eclipse                           |  |

Nomes de arquivos grandes  
(23 caracteres)

Nomes de arquivos pequenos  
(3 caracteres)

# Arquivos

## Conceito:

Um arquivo é uma coleção de informações relacionadas, **identificada por um único nome e que é gravado em algum dispositivo de armazenamento secundário (não volátil).**

Nomeação de arquivos é o fato de um processo nomear um arquivo e este arquivo ser acessado por outros **processos por meio de seu nome**, mesmo depois de ser usado pelo processo criador (PAIXÃO, 2016).

O nome de um arquivo possui uma segunda parte que é chamada de extensão, e é separada do nome do arquivo por um ponto. A extensão identifica o tipo de arquivo que foi criado.

## Sistemas de Arquivos

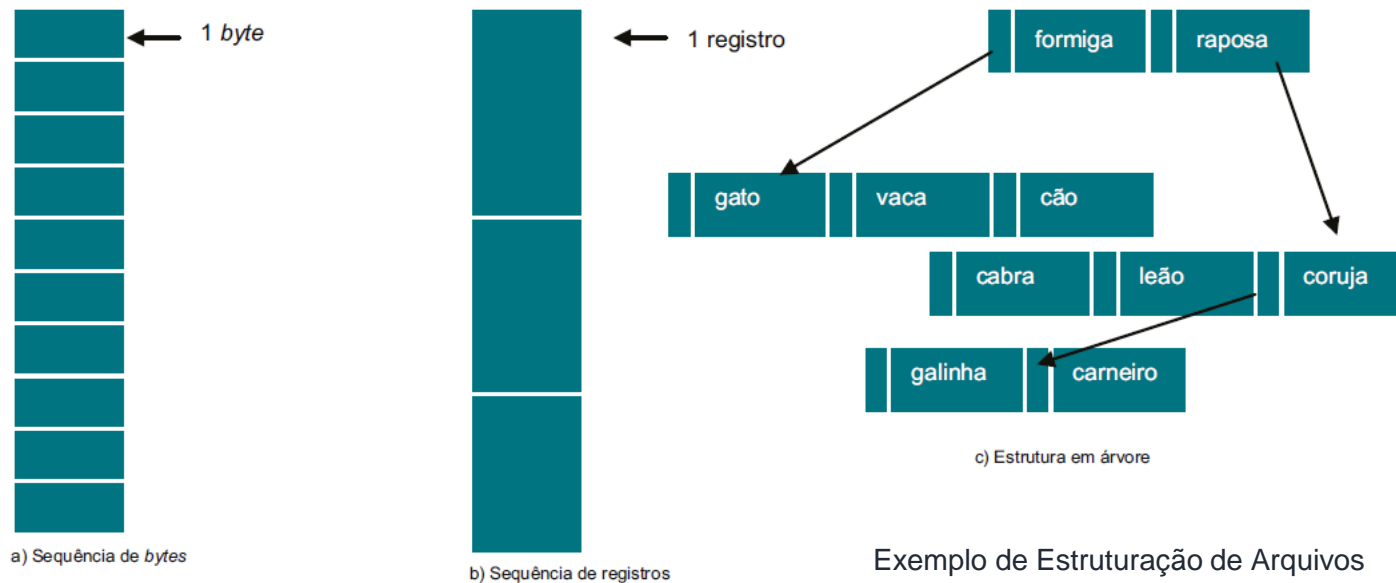
**Os arquivos são gerenciados pelo sistema operacional, visando facilitar o acesso dos usuários aos conteúdos dos arquivos.**

Aplicativos geralmente “carimbam” sua assinatura quando criam um determinado arquivo. Muitos programas atualmente permitem gravar outros tipos de extensão fora do padrão, fazendo com que os programas possam ser criados de arquivos lidos em aplicações diferentes (TANENBAUM, 2016).

| Nome do arquivo | Extensão |
|-----------------|----------|
| Planilha        | .xls     |
| Contrato        | .doc     |
| Código          | .txt     |
| planilha.xls    |          |
| contrato.doc    |          |
| código.txt      |          |

# Estrutura de um Arquivo

Um arquivo pode ser estruturado de várias formas. O sistema operacional não sabe do conteúdo do arquivo, simplesmente realiza operações de criação, remoção, alteração de nomes, cópias, fechamento, gravação, etc. (PAIXÃO, 2016).



- A informação com **significado** fica a cargo de **apresentação ao usuário**, como **nome e extensão**. O sistema **operacional somente enxerga um conjunto de bytes**. Ele trará a **sequência de bytes com transparência para o usuário**.



# Tipos de Arquivos

Vários tipos de arquivos são suportados pelos sistemas operacionais. Esses tipos podem ser divididos em:

- Regulares
- Diretórios
- Especiais de caracteres
- Especiais de bloco

## Tipos de Arquivos

# Arquivos Regulares

- em geral, são os arquivos que **armazenam as informações de usuário**. Tais arquivos são constituídos de tabela *American Standard Code for Information Interchange* (ASCII) ou **binários** (SILBERSCHATZ, 2015). São formados por linha de texto e podem ser impressos e editados em programas de edição de texto.
- Os arquivos binários são arquivos cujo conteúdo deve ser interpretado por um programa ou por um processador de hardware, ou seja, o arquivo não está em nenhum formato identificável (texto) externamente. Esses arquivos são geralmente considerados como uma sequência de bytes, agrupamento de oito bits.

## Especiais de Caracteres

- Estão relacionados à sistema de entrada e saída e com dispositivos de **E/S (console, portas seriais, etc.)**, leitura de fluxo de caractere. Um **arquivo de biblioteca ou driver** para dispositivos de caracteres implementa usualmente as chamadas de Sistema *open, close, read e write*.

# Tipos de Arquivos

## Arquivos especiais de blocos

- Estão relacionados a um dispositivo de bloco que permite acesso a informações em blocos de tamanho fixo (512, 1024 bytes, etc.), permitindo o acesso a um ou mais blocos em cada operação (SILBERSCHATZ, 2015).
- Fornecem acesso armazenado em memória temporária (buffer) para dispositivos de hardware e alguma abstração de suas especificidades. Dispositivos de armazenamento são exemplos típicos de dispositivos de blocos.

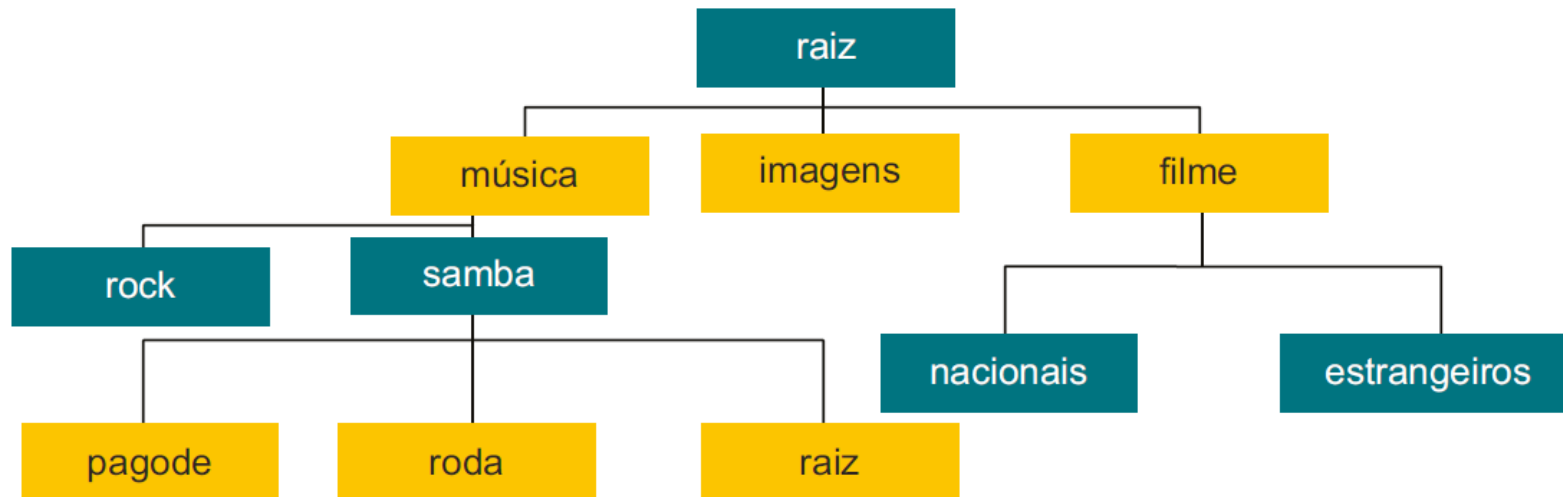
## Diretórios

- Qualquer arquivo armazenado em um computador está localizado dentro de uma diretório. Os diretórios são estruturas lógicas utilizadas para organização de uma estrutura de armazenamento, como discos rígidos, DVDs, pen drives, etc.

# Diretórios

Similar a nomeação de arquivos, porém, geralmente não se adiciona a extensão, ficando somente o nome, por exemplo: música, imagens, código, contratos, artigos, bibliotecas.

**Um diretório é uma estrutura de nível hierárquico, ou seja, possui um diretório principal chamada de raiz e os demais são criados a partir do diretório raiz**

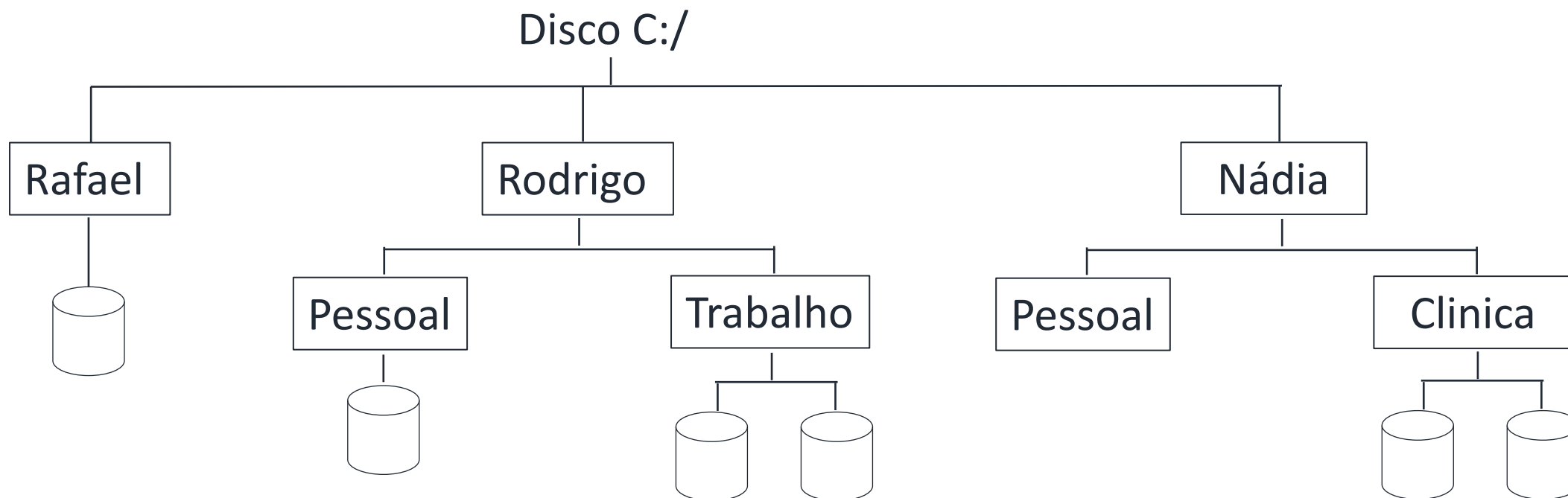


# Diretório

Na estrutura em árvore é possível criar quantos diretórios desejar, podendo um diretório conter arquivos ou outros diretórios que são denominados **subdiretórios**.

Cada arquivo nesta estrutura, possui um caminho “**patch**” único que descreve todos os diretórios desde a raiz até o diretório no qual o arquivo está localizado..

A estrutura de diretórios em árvore (**Tree-Structured Directory**), é atualmente a estrutura adotada pela maioria dos sistemas operacionais.



A estrutura de diretórios é o modo como o sistema operacional organiza logicamente os diversos arquivos contidos em um dispositivo de armazenamento (HD, SSD).

O processo mais simples é manter um único diretório. Que contenha todos os arquivos de todos os usuários.

A estrutura de diretórios em árvore (***Tree-Structured Directory***), é atualmente a estrutura adotada pela maioria dos sistemas operacionais.

# Nomes e Caminhos

- **Esse sistema hierárquico de diretórios determina o modo pelo qual o sistema operacional faz a localização de uma pasta dentro do sistema.** Para entender a localização exata do diretório, um sistema operacional precisa saber o seu nome e seu caminho (path).
- **Caminho é formado pelo diretório raiz e todos os subdiretório separados por barra (/) até a pasta onde se localiza o arquivo ou os arquivos.**
- O último diretório na representação do caminho é o diretório atual ou corrente.

raiz/aula\_puc/SO/[Aula13](#)



# Nomes e Caminhos

Um diretório pode conter dois tipos de caminho: **um caminho absoluto ou completo e um caminho relativo.**

- O caminho absoluto, representado indica a mesma localização em um sistema de arquivos, independentemente do diretório de trabalho atual, ou seja, inclui desde o diretório raiz.
- Já o caminho relativo tem seu início no local de pasta conhecida. Todo caminho relativo não contém uma “barra” no início. Um nome de arquivo localizado no diretório corrente pode ser considerado um caminho relativo. Por padrão, o sistema operacional busca no diretório corrente o arquivo a ser buscado.

**C:\Users\Nome\Documents\documentos.txt**



**RAIZ**      **SUBDIRETÓRIOS**      **ARQUIVO**

/Users/Nome/Documents/documentos.txt

# Nomes e Caminhos

**Um caminho relativo também faz referência ao local de um arquivo ou diretório.** Porém, a grande diferença é o **ponto de partida, que tem início no cwd, o current working directory** (diretório de trabalho atual, em tradução livre). Nesse caso, o nome do caminho é relativo ao diretório que você está no momento e não ao diretório raiz.

```
c:/users/user> dir arquivo.txt
```

# Tipos de Acesso aos Arquivos

- Sequencial: Fita Magnética, Fita DAT.
- Aleatório: HD, DVD/CD, Pen Drive dentre outros.

## Atributos de Arquivos

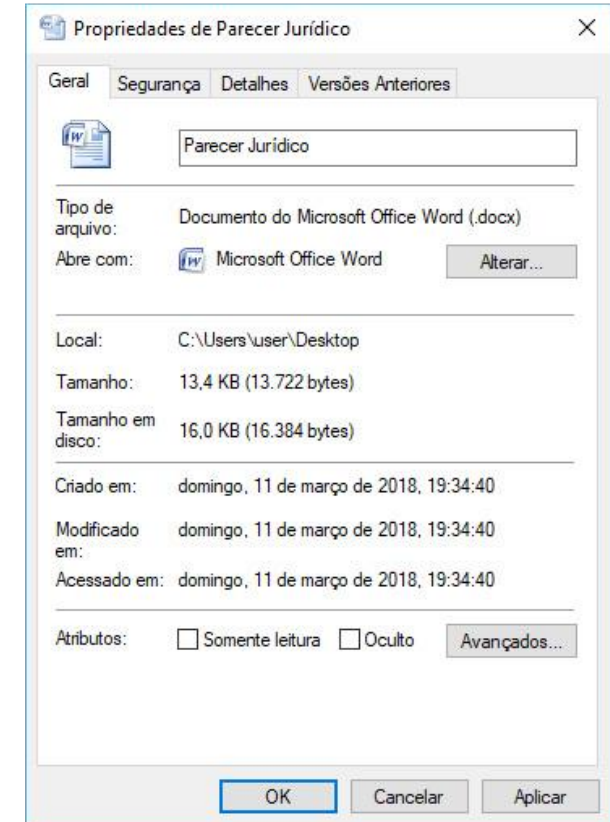
- Nome (até 255 caracteres).
- Tipo.
- Tamanho.
- Posição.
- Proteção.
- Hora, data e identificação do usuário.

# Atributos de um Arquivo

Todo arquivo possui uma série de **atributos** como:

- nome, extensão, tamanho, data e horário da criação, modificação, proteção(quem pode ter acesso), proprietário (quem o criou), a senha (senha para ter acesso ao arquivo), flags (indicadores: 0 - arquivos normais; 1 - arquivos do sistema), ocupação de alocação, etc.

Os flags são bits (0,1) que controlam e habilitam características mais específicas do arquivo. Determinam, por exemplo, se o arquivo é comum (0) ou de sistema (1), se o arquivo está para leitura/escrita (0) ou somente leitura (1), se o arquivo é ASCII (0) ou binário (1), etc. (SILBERSCHATZ, 2015).



# Acesso a um Arquivo

O acesso a um arquivo pode respeitar dois métodos: sequencial ou aleatório.

- No início dos sistemas, **o acesso do arquivo era sequencial**, pois todo o arquivo ou registro é lido a partir do início até o final. Esses arquivos eram geralmente gravados em fitas magnéticas.
- Hoje, **na maioria dos sistemas, encontra-se um acesso aleatório**, pois os arquivos podem ser acessados em qualquer ordem. Os registros podem ser acessados pela chave, e não pela posição.

# Operações com Arquivo

Os arquivos permitem que diversas operações possam ser realizadas. Essas operações podem ser (SILBERSCHATZ, 2015):

- **Create:** o arquivo é criado. Alguns atributos são especificados, como data e hora da criação.
- **Delete:** o arquivo é excluído do sistema. Liberação de espaço em disco.
- **Open:** antes de ler um arquivo, há necessidade de abrir. Permite que o sistema encontre e coloque na RAM os atributos e a lista de endereço de disco.
- **Close:** ao finalizar a tarefa e atributos, listas de endereçamento não são mais necessárias, esse método é chamado e libera tabela interna.
- **Read:** método responsável pela leitura dos dados.
- **Write:** método responsável pela escrita dos dados.
- **Seek:** esse método é responsável por determinar a posição dos dados no processo de acesso aleatório.
- **Get attributes:** método responsável pela leitura de atributos de um arquivo.
- **Set attributes:** método responsável pela alteração de atributos realizados pelo usuário após a criação.
- **Rename:** método responsável pela alteração do nome do arquivo

# Operações com Arquivos

- Criar (Create).
- Abrir (Open).
- Ler (Read).
- Escrever (Write).
- Fechar (Close).
- Renomear (Rename).
- Apagar (Delete).
- Acrescentar ou inserir um dado no final do arquivo (Append).
- Posicionar (Seek).
- Obter atributos ( Get Attributes).
- Definir atributos (Set Attributes).

## Abertura de arquivos

Solicitação  
do processo

|    |  |
|----|--|
| 1  |  |
| 2  |  |
| 3  |  |
| 4  |  |
| 5  |  |
| 6  |  |
| 7  |  |
| 8  |  |
| 9  |  |
| 10 |  |
| 11 |  |
| 12 |  |
| 13 |  |
| 14 |  |

Índice

Descritor do arquivo

| Nome/Identificador |
|--------------------|
| Proprietário       |
| Data de criação    |
| Tamanho            |
| Proteção           |
|                    |
|                    |
|                    |
|                    |

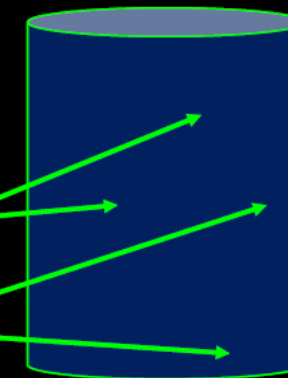


Tabela de descritores de arquivos



# Operações com Diretórios

- As chamadas de sistemas que gerenciam diretórios podem variar entre sistemas operacionais. Exemplos de chamadas de sistema para o Windows e Linux.

## **Windows**

*CreateProcess*

*WaitForSingleObject*

*ExitProcess*

*CreateFile*

*WriteFile*

*CreateDirectory*

## **Linux**

*fork*

*waitpid*

*exit*

*open*

*read*

*mkdir*

- ✓ Os sistemas de arquivos estão no núcleo do *kernel* do sistema operacional e seu conjunto de estruturas lógicas permite ao sistema operacional ter acesso e controlar os dados gravados no disco.
- ✓ Os sistemas de arquivos variam entre os sistemas operacionais, ou seja, cada SO possui seu próprio sistema de arquivos e cada um possui suas particularidades, como limitações, qualidade, velocidade de acesso, gerenciamento de espaço, entre outras características.
- ✓ O sistema de arquivos subdivide o disco de armazenamento em áreas menores para alocação de arquivos, chamadas setores. Um setor possui, no mínimo, 512 *bytes*. Imagine que um determinado usuário faça a gravação de um arquivo de aproximadamente 1 KB (1024 *bytes*), nesse caso, tal arquivo seria armazenado em dois setores.

Os computadores da Google, Chromebook, somente utiliza unidades de armazenamento para boot, a ideia principal é trabalhar com dados e software na nuvem.

O sistema Windows não armazena, por padrão, os dados diretamente em setores. Em vez disso, ele faz uso de um conjunto de setores de disco. Esse conjunto pode ter valores distintos, dependendo do tamanho do disco.

- Cluster é o nome dado a esse conjunto de setores que podem variar de tamanho: **4 kB, 8 kB, 16 kB**, etc.

## No Sistema Operacional Windows:

- **quanto menor o cluster, menos espaço em disco será desperdiçado.**
- Imagine um arquivo de **100 kB** de tamanho sendo armazenado em um cluster de **4 kB**. Esse arquivo seria armazenado em **25 clusters**. Imagine agora o mesmo arquivo sendo armazenado num cluster de 8 kB. Agora, esse mesmo arquivo sendo armazenado em **12,5 clusters**, mas como não há medição de 0,5 cluster, o arquivo seria armazenado em **13 clusters de 8 kB**, porém, no último cluster, somente 50% do cluster seria usado (MICROSOFT, 2018).

- **File Allocation Table (FAT):** Consiste em uma tabela que contém o mapeamento das unidades de alocação de um disco. O sistema operacional consulta esta tabela **para fazer a leitura e gravação em disco.**
  - ✓ Menores e simples (vantagem).
  - ✓ Dispersão de fragmentos (desvantagem).
  - ✓ Não possui segurança de acesso (desvantagem).

*Cluster* é o nome dado a um conjunto de setores que podem variar de tamanho: **4KB, 8KB, 16KB, e assim por diante.** Quanto **menor o cluster, menos espaço do disco será desperdiçado.**

- **New Technology File System (NTFS):** desenvolvido inicialmente para os sistemas operacionais de **rede da Microsoft, Windows NT.** Posteriormente foi adotado como sistema de arquivos padrão de todos os produtos Microsoft.
  - ✓ Possui inúmeras vantagens em relação ao sistema FAT.
  - ✓ Possibilita controles e implementação de segurança aos arquivos.

- No início, ainda na década de 1980, o sistema operacional para computadores pessoais foi o *Disk Operating System* (DOS). Esse sistema operacional fazia uso de um sistema de arquivos baseado em tabela de alocação de arquivos (FAT).
- FAT é uma tabela que contém o mapeamento das unidades de alocação em disco. Isso permite que o SO consulte essa tabela nas operações de leitura e gravação de dados nos discos.
- Uma FAT possui uma dispersão de fragmentos de dados em operação de exclusão/criação de arquivos muito séria. Dados podem ser fragmentados, tornando o processo de leitura e gravação mais lento.
- Não possui segurança de acesso, ou seja, qualquer usuário pode acessar qualquer arquivo/pasta armazenado nesse sistema de arquivos.
- Por ser mais simples e menos complexa, a velocidade de escrita e leitura da FAT é mais rápida, por isso esse sistema de arquivos é muito usado em dispositivos de armazenamento com cartões de memória, pen drives e outros semelhantes. Pode-se dividir a FAT em três tipos: **FAT, FAT 16 e FAT 32.**

# Sistema de Arquivo FAT, FAT12 e FAT16

- **O sistema de arquivos FAT é um sistema de 12 bits introduzido em 1980.** Não faz diferença entre letras maiúsculas e minúsculas. Era utilizado em disquetes e outros meios removíveis na época. Os primeiros sistemas operacionais MS-DOS faziam uso da FAT. A nomeação de arquivos era constituída por oito caracteres de nome, ponto, seguido de extensão com três caracteres (exemplo: arquivos.txt).
- **O sistema FAT12 suporta tamanhos de unidade e tamanhos de arquivo de até 16 MB usando clusters de 4 kB, ou 32 MB, usando 8 kB, com um número máximo de 4.084 arquivos em um único volume (ao usar clusters de 8 kB).** Diversos tipos de atributos de arquivo foram introduzidos pela primeira vez no FAT12, incluindo rótulo oculto, somente leitura, sistema e volume.
- **O FAT16 é uma versão mais sofisticada do que a anterior (FAT12).** Foi o sistema de arquivos primário do MS-DOS 4.0 até o Windows 95. Essa nova versão usava tamanhos variados de clusters **4 kB, 8 kB, 16 kB, 32 kB**. O tamanho máximo de clusters era de **(2<sup>16</sup> bits)**. Isso resulta em **65.536 clusters**. O tamanho máximo do volume de dados compreendido pela FAT16 é o resultado do **tamanho do cluster multiplicado pela quantidade de clusters**. Assim, um sistema **FAT16 com clusters de 4 kB** possui um volume de **(4 kB x 65.536)** aproximadamente **2 Mbytes**, entretanto, um sistema **FAT16 com 32 kB** de tamanho de clusters permitia um volume de **2 Gbytes (32 kB x 65536)**.

Um recurso adicionado foi o virtual FAT ou somente vFAT. Apresentado no Windows 95, tem como característica básica estender os nomes de arquivos ou **diretórios até 255 caracteres**, ou seja, com esse recurso, os nomes poderiam ser escritos além de oito caracteres de nome, como “trabalho de escolha – História da arte.doc”.



# Sistema de Arquivo FAT32

A última versão do sistema **FAT** foi lançada em **1996** com o sistema operacional **Windows 95 OSR/2** e o **MS-DOS 7.1**. Esse sistema lançado permaneceu como sistema operacional base até o Windows Millennium Edition (Windows ME).

O limite máximo de arquivo é de **268.173.300 (232 x 512)**. A multiplicação por setor, e não por *clusters*, foi determinada pela empresa Microsoft. Isso devido a problemas que poderiam ocorrer com a inicialização do sistema operacional, resultado da limitação na área de *boot* (setor de *boot*).

Como cada setor, geralmente, possui **512 bytes**, ou seja, **232** (4.294.967.296) multiplicado por **512**, resultando em **2.147.483.648 kB** ou **2 TB** de volume.

Caso esse tamanho fosse alterado para **64 kB (máximo)**, um volume poderia chegar a **16 Terabytes**.

**Assim como os demais sistemas FAT e FAT16, ele não possuía segurança de acesso aos arquivos/pastas.** Por trabalhar com *clusters* menores, o sistema FAT32 desperdiça menos unidades de alocação do que seus antecessores. Porém, tem um percentual de lentidão um pouco maior, devido aos *clusters* menores, o sistema leva mais tempo para localizar todos os *clusters* de armazenamento de um arquivo. Por padrão, o sistema FAT32 é vFAT.

- BHATTACHARJEE, A. Preserving Virtual Memory by Mitigating the Address Translation Wall. IEEE Micro, v. 37, n. 5, 2017. 6-10 p. DOI: 10.1109/MM.2017.3711640
- CRUZ , E M et al. Communication-aware thread mapping using the translation lookaside buffer. Concurrency and Computation: Practice and Experience, v. 27, n. 17, 2015. 4970-4992 p. DOI: 10.1002/cpe.3487.
- MACHADO, Francis Berenger; MAIA, Luiz Paulo. Arquitetura de sistemas operacionais. 5a ed. Rio de Janeiro: LTC, 2013. 263 p. ISBN:9788521622109.
- SILBERSCHATZ, Abraham. Fundamentos de Sistemas Operacionais. 9ª ed. LTC, 2015. 524 p. ISBN: 9788521629399. ISBN digital 9788521630012.
- STUART, Brian L.. Princípios de sistema operacionais: projetos e aplicações. São Paulo: Cenage Learning, 2011. 637 p. ISBN: 9788522107339.
- TANENBAUM, Andrew S. Sistemas Operacionais Modernos. 4a ed. São Paulo: Pearson, 2016. 864 p. ISBN: 9788543005676.