

Unidade 3

Sistemas de Arquivos



Introdução

Sistemas computacionais armazenam um grande volume de dados. O armazenamento e a recuperação dos dados são atividades vitais para qualquer tipo de aplicação existente.

Os processos precisam ler e gravar um grande volume de dados em dispositivos como as memórias e HD's (*Hard disc*) de forma permanente, além de poder compartilhar com outros processos. É através da implementação de arquivos que os sistemas operacionais estrutura e organiza as informações. A parte do sistema operacional responsável pela essa organização é chamada de sistemas de arquivos.

Para quem está começando na área de programação é importante saber esses aspectos de arquivo, porque em desenvolvimento de softwares o mecanismo de trabalhar com arquivos é um dos itens mais importantes no desenvolvimento de um sistema.

Tanto na aula 1 e 2, será apresentado aspectos presentes no sistema de arquivos, como a organização, métodos de acesso utilizados, diretórios e alocação dos arquivos em disco. Então caro aluno, leia o texto e as referências bibliográficas pedidas, que com certeza você terá um grande aprendizado. Bons estudos.

Objetivos

- Entender os conceitos sobre Sistemas de arquivos;
- Aprender como se organiza arquivos em um sistema computacional.

Conteúdo programático

Aula 01 – Arquivos

Aula 02 – Implementação de arquivos



Você poderá também **assistir às videoaulas** em seu celular! Basta apontar a câmera para os **QR Codes** distribuídos neste conteúdo.

Pode ser necessário instalar um aplicativo de leitura QRcode no celular

Aula 01 – Arquivos

Não tem como utilizar um sistema computacional sem utilizar-se de um sistema de arquivos para trabalhar com informações. A todo momento sistemas fazem cópias e gravam informações. Um processo tem que ter a capacidade para ler e gravar informações em grande volume de dados em dispositivos de armazenamento. É com arquivos que o sistema operacional cria um ambiente de trabalho.

O sistema operacional (SO) tem funções que gerencia os arquivos e as suas operações – conhecida como gerência de arquivos. Essa gerência também é responsável pela segurança de acesso aos arquivos, que garante que um arquivo não seja usado por um usuário não autorizado. Outra função que podemos incluir para a gerência de arquivos é o compartilhamento de arquivos, através de uma rede, por vários processos e usuários do sistema operacional.

A função de gerência de arquivos é a parte mais “palpável” para um usuário de um sistema computacional, uma vez que sempre estará manipulando arquivos, ora para criar ora para editar seus documentos ou para executar programas.

Mas afinal, qual é definição de um arquivo? Para Machado e Maia (2000), um arquivo é constituído de informações logicamente relacionadas, podendo representar dados ou programas. Um programa contém instruções que são compreendidas pelo processador (arquivo executável), enquanto que um arquivo de dados tem uma estrutura mais livre, como em um texto ou de uma forma mais rígida, um banco de dados.

Para o usuário, existem diferentes tipos de arquivos: multimídia, imagem, textos etc. Para o sistema operacional um arquivo nada mais é que um conjunto de registros definido pelo sistema operacional.

Os arquivos dentro de um SO devem ser nomeados. Quando algum processo cria um arquivo ele fornece um nome para esse arquivo. Quando o processo termina, o arquivo

continua existindo e outros processos podem ter acessos a ele simplesmente buscando seu nome (TANENBAUM, 2015).

De modo geral, a identificação de um arquivo é composta por duas partes separadas com um ponto. A parte que fica após o ponto é denominada **extensão** do arquivo e tem como finalidade identificar o seu conteúdo. Dessa forma, é possível convencionar que a extensão .TXT identifica um arquivo-texto, enquanto a extensão .EXE indica um arquivo executável (geralmente para sistemas *WINDOWS*). Exemplos: fulano.txt, calculadora.exe. No quadro 1 são apresentadas algumas extensões conhecidas.

Quadro 1 - Extensão de arquivos

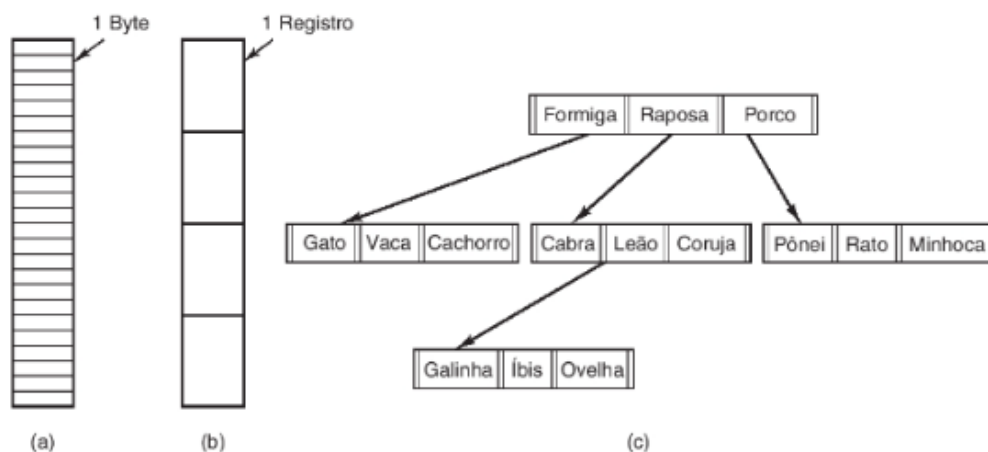
Arquivo .EXE	Arquivo executável
Arquivo .TXT	Arquivo de texto
Arquivo .PNG	Arquivo de imagem
Arquivo .BIN	Arquivo de formato binário

Fonte: o autor (2021)

Organização dos Arquivos

A organização de arquivos consiste em como os dados estão internamente armazenados. No momento da criação de um arquivo, é possível definir que organização será adotada. As três maneiras mais conhecidas de estruturação dos arquivos podem ser as seguintes, segundo Tanenbaum (2015): sequência de *bytes*, sequência de registros e árvore. Essas três possibilidades são exibidas na figura 1 abaixo:

Figura 1 - Três tipos de arquivos: a) Sequência de bytes b) Sequência de registros c) Árvore



Fonte: Tanenbaum (2015, p. 160)

A forma mais simples de organização de arquivos é a sequência de *bytes* não estruturada, veja a figura 1 (a). Organizar esses arquivos de maneira não estrutura oferece com certeza mais flexibilidade. Os programas de usuários podem inserir qualquer coisa que desejam em seus arquivos e chamá-los de qualquer nome. E ainda, por não ter uma estrutura definida de acesso, não é possível o acesso direto ao arquivo.

Vamos dar uma verificada agora na forma sequencial de registros, veja a figura 1 (b). Nesse modelo, o arquivo é composto de uma sequência de registros de tamanho fixo, no qual cada um tem uma determinada estrutura interna. A principal ideia é que a operação de leitura retorna um registro e a de escrita sobrepõe ou anexa um registro (TANENBAUM, 2015).

A figura 1 (c) mostra o último tipo. Nessa organização, um arquivo é composto de uma árvore de registros, não precisando ser do mesmo tamanho e cada um contendo um campo chave em uma posição fixa no registro. Dessa forma, a árvore é ordenada pelo campo chave com a função de se buscar mais rapidamente por uma chave específica. Então a operação básica nesse sistema não é obter o próximo registro, mas sim conseguir o registro com a chave especificada. No caso do exemplo da figura 1 (c), o usuário poderia necessitar de um registro cuja chave fosse "galinha", sem se importar com a posição exata do arquivo.

Videoaula 1

Agora assista ao vídeo no qual abordaremos os conceitos sobre organização de arquivos.



Videoaula 1

Utilize o QRcode para assistir!

Agora assista ao vídeo no qual abordaremos os conceitos sobre organização de arquivos.



Indicação de Vídeo

Assista a esse interessante vídeo do canal *aiedonline*, no qual explica sobre arquivos em sistemas computacionais. Disponível em: <https://www.youtube.com/watch?v=cw6kLHIGtdg>. Acesso em: 30 abr. 2021.

Método de Acesso aos dados do Arquivo

Nos primeiros sistemas operacionais o acesso aos arquivos era do tipo: **acesso sequencial**. Nesse tipo de acesso, os arquivos eram restritos à leitura dos registros na ordem em que eram gravados, dessa forma não era possível acessar diretamente um registro. A gravação de novos registros era possível apenas no final do arquivo (MACHADO; MAIA, 2000).

Com o tempo, os equipamentos de armazenamento foram evoluindo e o uso de novos métodos de acesso eficientes foram possíveis. O primeiro método que contempla essa evolução foi o **método direto**. Nesse método, é permitido a leitura/gravação de um registro diretamente na sua posição. O acesso é feito através do número de registro que é

a sua posição relativa ao início do arquivo. Aqui não existe restrição a ordem em que são lidos ou gravados os registros, porém sempre é necessário informar o número do registro.

O acesso direto é utilizado somente quando o arquivo é definido com registros de tamanho fixo, haja vista que é feito um cálculo da posição do registro dentro do arquivo com base no tamanho do registro e do número do registro (MACHADO; MAIA, 2000). Também, o acesso direto pode ser combinado com acesso sequencial. Dessa maneira, pode ser feito o acesso a um registro qualquer e posteriormente acessar sequencialmente os demais registros do arquivo.

Temos também um método mais moderno de acesso. Chamado de acesso ***indexado ou acesso por chave***. Para esse método em específico, o arquivo deve possuir uma área de índice onde existam ponteiros para os diversos registros. Então, ao querer acessar um registro, deverá ser especificada uma chave através da qual o sistema pesquisará na área índice o ponteiro correspondente.

Operações de Entrada/Saída

Os sistemas de arquivo oferecem um conjunto de *system calls*, que são mecanismos programáticos pelo qual um programa de computador solicita um serviço do núcleo do sistema operacional sobre o qual ele está sendo executado, o qual concede as aplicações realizar operações de entrada e saída, tais como: tradução de nomes em endereços, leitura e gravação de dados e criação ou eliminação de arquivos. Dessa maneira, o usuário se depara com uma interface simples e uniforme entre a aplicação e os diversos dispositivos.

Podemos visualizar no quadro 2 algumas rotinas encontradas na maioria das implementações de sistemas de arquivo.

Quadro 2 - Operações de entrada/saída

Comando	Descrição
CREATE	Criação de um arquivo
OPEN	Abertura de um arquivo
READ	Leitura de dados de um arquivo
WRITE	Gravação de dados em um arquivo
CLOSE	Fechamento de um arquivo
RENAME	Alteração do nome de um arquivo
DELETE	Eliminação de um arquivo

Fonte: Tanenbaum (2015, p. 164)

Os arquivos servem para armazenar informação e permitir que ela seja recuperada posteriormente. A seguir, com base no algoritmo da figura 2, abaixo, exemplificaremos algumas chamadas de sistema relacionadas aos arquivos utilizando da linguagem C.

Figura 2 - *Systems calls* para sistema de arquivos na linguagem C

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  int main(void)
4  {
5      // criando a variável ponteiro para o arquivo
6      FILE *pont_arq;
7
8      //abrindo o arquivo
9      pont_arq = fopen("arquivo.txt", "a");
10
11     // fechando arquivo
12     fclose(pont_arq);
13
14     //mensagem para o usuário
15     printf("O arquivo foi criado com sucesso!");
16
17     system("pause");
18     return(0);
19 }
```

Fonte: o autor (2021)

Observando a figura 2, acima, especificamente na linha 6 podemos notar o uso do recurso de ponteiros. Na linha 9, notamos a instrução *fopen* para a abertura do arquivo e a instrução *fclose* para o fechamento do mesmo. Como veem, essas chamadas ao sistema facilitam o uso de arquivos pelo usuário.

Indicação de Vídeo

Você que está estudando Análise de sistemas, com certeza irá se interessar em saber como é acessado um arquivo via programação. Assista a este vídeo e entenda. Disponível em: <https://www.youtube.com/watch?v=eriDnpkh5kA>. Acesso em: 30 abr. 2021.

Para suportar as operações de entrada e saída, o sistema deve manter informações de cada arquivo criado. Essas informações são conhecidas como atributo do arquivo. Esses atributos permitem ao sistema de arquivo ter conhecimento sobre várias configurações do arquivo, veja exemplos de atributo na tabela abaixo. Lembrando que um atributo de arquivo pode variar conforme o sistema operacional.

Quadro 3 - Atributos de arquivos

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

Fonte: o autor (2021)

Leitura Obrigatória

Vamos aprofundar o conhecimento a respeito dos arquivos. Por isso, para conhecer mais, leia o capítulo 4 do livro *Sistemas Operacionais Modernos* de Andrew Tanenbaum, o tópico 4.1 inteiro. Essa leitura com certeza irá ajudar você a complementar seus estudos. Acesse o *link*: Disponível em: <https://plataforma.bvirtual.com.br/Leitor/Publicacao/1233/pdf/186>. Acesso em: 30 abr. 2021.

Videoaula 2

Agora, assista ao vídeo no qual abordaremos os conceitos sobre arquivos.



Videoaula 2

Utilize o QRcode para assistir!

Agora, assista ao vídeo no qual abordaremos os conceitos sobre arquivos.



Diretórios

Para organizar os arquivos, os sistemas de arquivos usam de maneira geral diretórios ou pastas. A estrutura de diretório é a maneira como o sistema organiza logicamente os diversos arquivos contidos em um disco (MACHADO; MAIA, 2000). Essas estruturas contêm entradas que são associadas aos arquivos onde são armazenadas informações como localização física, nome, organização e demais atributos.

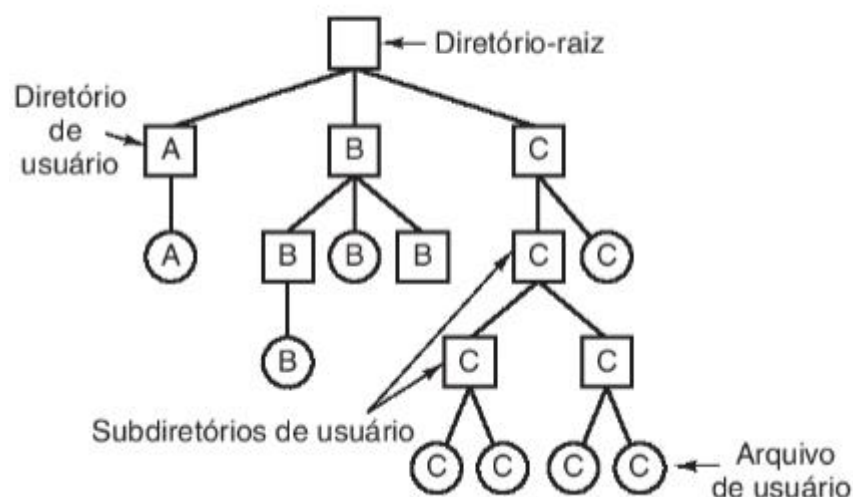
Quando um arquivo é aberto, o sistema operacional no momento procura a sua entrada na estrutura de diretórios em tabela mantida na memória principal, contendo todos

os arquivos. Por fim, toda vez que um arquivo for usado é necessário fechá-lo para que libere o espaço na tabela de arquivos aberta.

Uma das implementações mais simples de um sistema de diretórios é ter apenas um diretório contendo todos os arquivos. Essa implementação é conhecida como nível único (*single - level directory*). Nesse caso, todos os arquivos do disco são centralizados em apenas um diretório. Com certeza, que a maior vantagem desse sistema é a sua simplicidade e a capacidade de localizar os arquivos rapidamente, uma vez que existe apenas a um lugar para se pesquisar. Geralmente, é utilizado em sistemas embarcados simples. A desvantagem nesse sistema de nível único é a segurança dos arquivos armazenados, visto que, por exemplo, dois usuários podem criar dois arquivos com o mesmo nome (TANENBAUM, 2015).

Atualmente, usuários modernos tem milhares ou milhões de arquivos em seu computador. Dessa forma, encontrar algo em um diretório único seria algo bem trabalhoso. Por isso, houve a necessidade de criar algo mais prático onde poderia hierarquizar os arquivos. Essa ideia é conhecida como estrutura de diretórios em árvore e é utilizada na maioria dos sistemas operacionais, observe a figura 3, abaixo. Na figura, nota-se os diretórios A, B, C contidos no diretório raiz e cada diretório pode pertencer a um usuário diferente; dois desses usuários criaram subdiretórios para seus projetos.

Figura 3 - Sistema hierárquico de diretórios



Fonte: Tanenbaum (2015, p. 167)

Na estrutura em árvores existe a possibilidade de o usuário criar quantos diretórios desejar, podendo um diretório conter arquivos ou outros diretórios (chamados de subdiretórios). Esse sistema propicia uma ferramenta muito importante de estruturação para organização de trabalhos.

Cada arquivo nessa estrutura possui um *path* (caminho) único que descreve todos os diretórios desde a raiz até o diretório no qual o arquivo está ligado. Os caminhos a serem tomados podem ser absolutos e relativos. Os *paths* absolutos apontam sempre para a mesma direção em um sistema de arquivos, independentemente do diretório de trabalho atual. Para isso, tem que partir do diretório raiz. Como exemplo do sistema operacional UNIX, o caminho ***/usr/ast/caixapostal*** significa que o diretório-raiz contém um subdiretório *usr*, que por sua vez, contém o subdiretório *ast*, que contém o arquivo ***caixapostal*** (TANENBAUM, 2015).

Por outro lado, o caminho relativo parte da pasta que você está no momento, evitando assim a obrigação de fornecer o caminho absoluto completo. Para exemplificar mais a nossa leitura considere que você criou um projeto no caminho: **C:\Users\SeuNome\Projetos\MEUPROJETO\Home\Index.html**. Vamos acessar o arquivo **Index.html** das duas maneiras.

Caminho Relativo: **\Home\Index.html**

Caminho Absoluto: **C:\Users\SeuNome\Projetos\MEUPROJETO\Home\Index.html**

São dois conceitos que apesar de simples, são muito importantes para um sistema. Alguns programas precisam ter acesso a um arquivo específico sem ter a preocupação em saber qual é o diretório de trabalho. Para isso, eles devem sempre usar os nomes de caminhos absolutos.

No sistema operacional UNIX, os componentes do caminho são separados por /. No Windows o separador é o \. Desta forma, o mesmo caminho poderia ser escrito nesses sistemas como:

Windows: **\usr\ast\caixapostal**

Unix: **/usr/ast/caixapostal**

Indicação de Vídeo

Assista ao vídeo baixo no qual será explicado sobre as estruturas de diretórios. Disponível em: <https://www.youtube.com/watch?v=S008lgndlhv>. Acesso em: 30 abr. 2021.

Leitura Obrigatória

Vamos aprofundar o conhecimento a respeito dos arquivos. Por isso, para conhecer mais leia o capítulo 4 do livro *Sistemas Operacionais Modernos* de Andrew Tanenbaum, o tópico 4.2, página 166. Essa leitura com certeza vai ajudar a você complementar seus estudos. Disponível em: <https://plataforma.bvirtual.com.br/Leitor/Publicacao/1233/pdf/0>. Acesso em: 30 abr. 2021.

Implantação do sistema de arquivos

Anteriormente estudamos como os arquivos e diretórios podem ser vistos dentro do contexto do usuário. Porém, é importante também sabermos como toda essa estrutura é implementada em disco e como é gerenciada para funcionar de maneira eficiente e confiável.

Os sistemas de arquivos são armazenados em disco e, esses discos são divididos em uma ou mais partições que podem ter sistemas de arquivos independentes para cada partição. O setor 0 do disco é chamado de MBR (*master boot record – registro mestre de inicialização*) e atualmente temos o GPT (*GUID Partition Table*) que ambos servem para inicializar o computador. O fim do MBR contém a tabela de partição, a qual armazena os endereços iniciais e finais de cada partição.

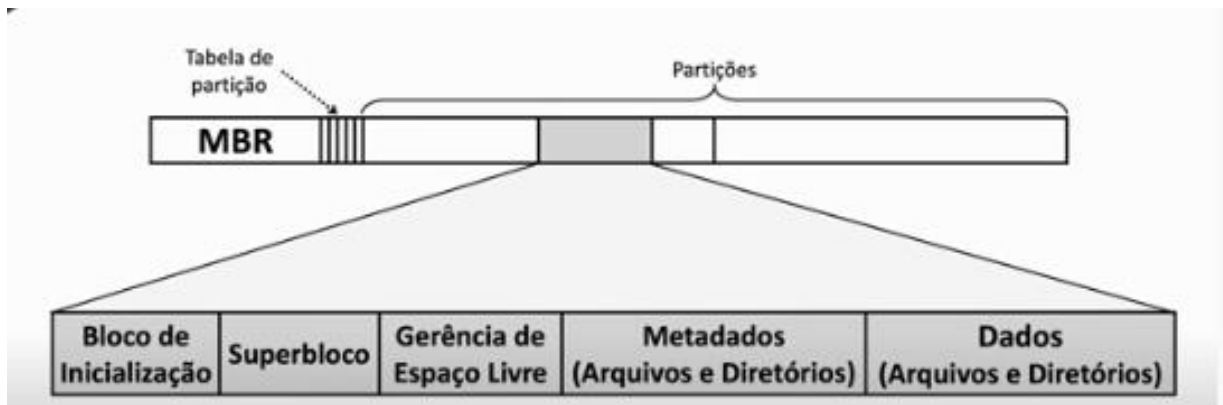
No momento do Boot (Inicialização dos sistemas básicos do computador) acontece os seguintes fatos:

1. A BIOS (*Basic Input/Output System*) lê e executa o MBR;
2. O programa localiza a partição ativa e lê seu primeiro bloco (Inicialização) e o executa;

3. Um programa que está contido no bloco de inicialização carrega o sistema operacional;

Geralmente os sistemas tem a seguinte configuração para os esquemas de arquivo que é mostrado na figura 4.

Figura 4 - Organização de um sistema de arquivo



Fonte: Adaptado de Tanenbaum (2015, p. 169)

Observando a figura 4, notamos um possível esquema de sistema de arquivos. Verifica-se a existência da MBR, da tabela de partição e as partições. Dentro da partição podemos ter os seguintes elementos:

- Blocos de inicialização: tem a função de armazenar o programa responsável pelo carregamento do sistema operacional;
- SuperBloco: armazena informações sobre o tipo de sistema utilizado de arquivos e o número de blocos;
- Gerência de espaço livre: estruturas de dados para gerenciar blocos livres ou usados de disco;
- Metadados: são estruturas e dados que irão armazenar metadados dos arquivos e diretórios;
- Dados: dados dos arquivos e diretórios pertencentes a partição.

Como visto então, em cima de partições que é instalado um sistema operacional e seus respectivos sistemas de arquivos e diretórios.

Videoaula 3

Agora assista ao vídeo no qual abordaremos os conceitos sobre diretórios.



Videoaula 3

Utilize o QRcode para assistir!

Agora assista ao vídeo no qual abordaremos os conceitos sobre diretórios.



Leitura Obrigatória

Vamos aprofundar o conhecimento a respeito dos arquivos. Por isso, para conhecer mais leia o capítulo 4 do livro *Sistemas Operacionais Modernos* de Andrew Tanenbaum, o tópico 4.3, página 169. Essa leitura com certeza vai ajudar a você complementar seus estudos. Disponível em: <https://plataforma.bvirtual.com.br/Leitor/Publicacao/1233/pdf/0>. Acesso em: 30 abr. 2021.

Aula 02 – Implementação de arquivos

Arquivos são alocados em blocos no disco. Assim, a criação de arquivos em disco exige que o sistema operacional tenha controle de quais áreas ou blocos no disco que estão livres. Esse controle pode ser realizado utilizando-se de uma estrutura de dados que armazena informações que podem ser utilizadas pelo sistema de arquivos para gerenciar o espaço livre do disco (MACHADO; MAIA, 2000). Nessas estruturas que podem ser uma lista ou tabela, é possível identificar blocos livres que poderão ser alocados para um novo arquivo.

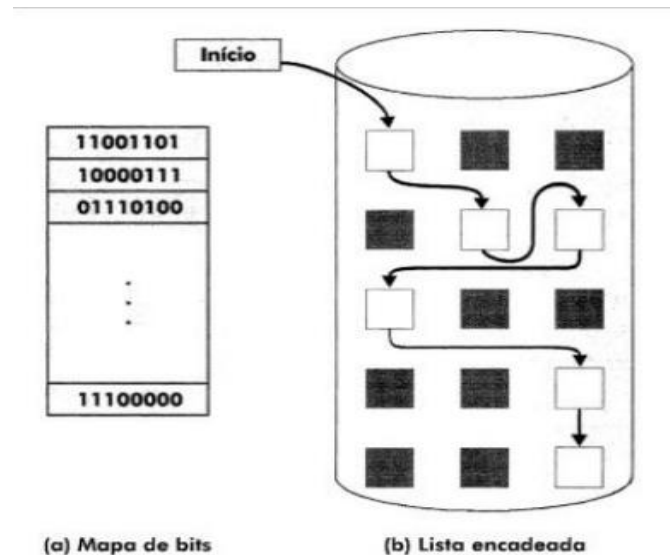
Quando um novo arquivo é alocado, o espaço do arquivo é removido para que não seja reutilizado. Também quando um arquivo é eliminado, todos os seus blocos são liberados para a estrutura de espaço livre.

O monitoramento de espaço livre em um disco é realizado principalmente através de dois diferentes métodos: mapa de bits e lista encadeada, vamos estudá-los agora.

Mapa de bits: Quando se utiliza esse método, um espaço fixo adicional de disco é usado especificamente para o mapeamento do espaço livre, de tal forma que para cada bloco de disco é utilizado um bit no mapa, ver figura 1 (a), abaixo. Então dessa maneira um disco com n blocos requer um mapa de bits com n bits.

Lista encadeada: Esse método é realizado por meio da ligação encadeada de todos os blocos livres do disco. Vale a pena salientar que os blocos livres são blocos que não contém arquivos de usuários ou programa. Mas esses blocos não ficam em brancos, porque enquanto não estão sendo utilizados, eles contêm informações que o sistema operacional armazena para mapear o espaço livre, ver figura 1 (b). Para acontecer o encadeamento, cada bloco possui uma área reservada de armazenamento do endereço do próximo bloco.

Figura 1 - (a) Mapa de bits;(b) Lista encadeada



Fonte: Machado; Maia (2000, p. 159)

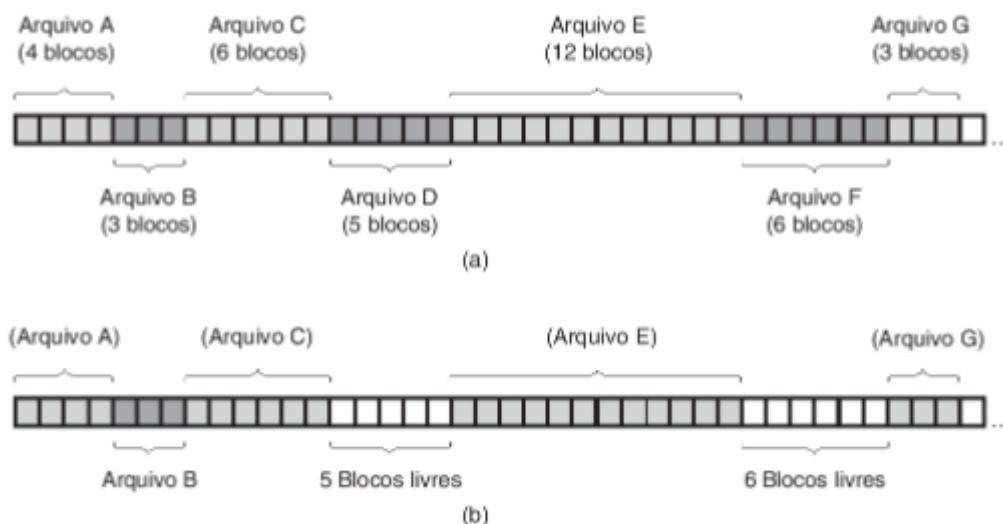
Muito bem amigo aluno! Agora que estudamos acima os principais métodos de gerenciamento de blocos livres, vamos nos concentrar agora em como os arquivos são implementados em disco.

Alocação contígua

A maneira mais simples de armazenar um arquivo é na forma alocação contígua. Esse método consiste em armazenar um arquivo em blocos sequencialmente dispostos no disco. Por exemplo, um disco com blocos de 1KB, um arquivo de 50KB seria alocado em 50 blocos consecutivos. Com blocos de 2 KB, o arquivo seria alocado em 25 blocos consecutivos (TANENBAUM, 2015).

Para entender o processo, observe a figura 2 abaixo:

Figura 2 - (a) Alocação contígua do espaço em disco. (b) O estado do disco depois de os arquivos D e F terem sido removidos



Fonte: Tanenbaum (2015, p. 170)

Observando a figura 2 (a) e (b) da esquerda para direita notamos alguns arquivos colocados de maneira sequencial. Notamos o arquivo A ocupando 4 blocos e após o arquivo B ocupando 3 blocos e assim sucessivamente com os demais arquivos. Podemos notar também, analisando a figura, que cada arquivo começa no início de um novo bloco; exemplificando, se o arquivo A ocupar três blocos e meio, um espaço no final do último bloco será desperdiçado, pois não se ocupa metade do bloco.

Esse tipo de alocação tem duas vantagens importantes. Primeiramente, é a simplicidade em controlar os blocos de um arquivo. Isso é feito apenas tendo dois números: o endereçamento em disco do primeiro bloco e os números de blocos no arquivo. Dessa forma, tendo o número do primeiro bloco, o número de qualquer outro bloco pode ser encontrado apenas adicionando os blocos.

Uma segunda vantagem é o desempenho da leitura, uma vez que todo arquivo pode ser lido do disco em uma única operação. Simplesmente posicionar a leitura para o primeiro bloco.

Nem tudo são flores, caro aluno! Então temos um ponto fraco importante também. Com o tempo o disco começa a ficar fragmentado. Para entender isso, observe a figura 2

(b). Na figura é mostrado que os arquivos D e F foram removidos. Quando um arquivo é removido, seus blocos são liberados deixando assim uma lacuna de blocos livres no disco. Esses blocos livres podem ser preenchidos com outro arquivo, desde que, o arquivo novo tenha o mesmo número de blocos livres que está no disco (TANENBAUM, 2015).

Obviamente precisa existir alguma estratégia de alocação para selecionar qual o segmento de bloco livre deve ser escolhido, dado que podem diferir do tamanho do novo arquivo que será gravado. Essas técnicas são as seguintes:

- **First-fit:** nessa técnica, o primeiro segmento livre com tamanho suficiente para alocar o arquivo é selecionado. Aqui se faz uma busca sequencial, sendo interrompida apenas quando é localizado um segmento de tamanho adequado.
- **Best-fit:** aqui é buscado o menor segmento livre disponível com tamanho suficiente para armazenar o arquivo. A busca em toda a lista se faz necessária para a seleção do segmento, salvo se a lista esteja ordenada por tamanho.
- **Worst-fit:** neste caso, o maior segmento é alocado. Mais uma vez, a busca em toda a lista se faz necessária, a menos que exista uma ordenação por tamanho.

Podemos concluir que a alocação contígua apresenta alguns inconvenientes: Fragmentação e a determinação do espaço em disco necessário a um arquivo.

Videoaula 1

Agora assista ao vídeo no qual abordaremos os conceitos sobre alocação de arquivos.



Videoaula 1

Utilize o QRcode para assistir!

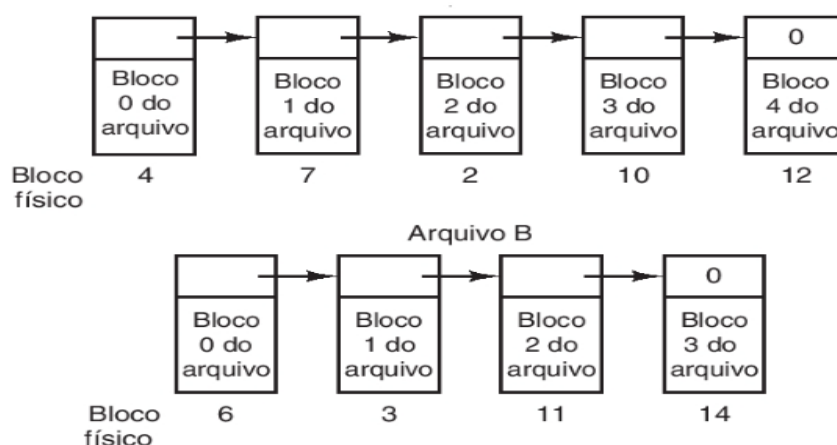
Agora assista ao vídeo no qual abordaremos os conceitos sobre alocação de arquivos.



Alocação Encadeada

Na alocação encadeada, um arquivo pode ser organizado como um conjunto de blocos ligados logicamente no disco, independentemente da sua localização física, conforme mostra a figura 3 (MACHADO; MAIA, 2000). A primeira palavra de cada bloco é usada como ponteiro para o próximo, assim os blocos restantes são usados para dados.

Figura 3 - Armazenamento de um arquivo como uma lista encadeada de blocos de disco



Fonte: Tanenbaum (2015, p. 171)

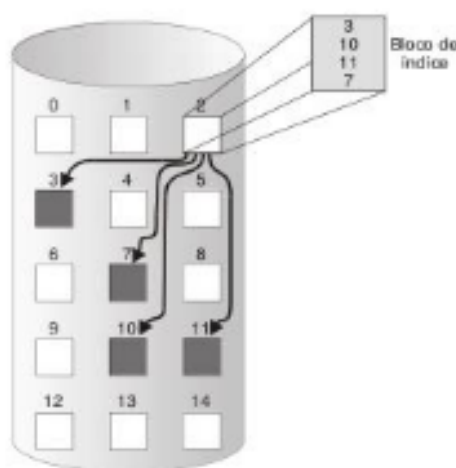
Repare na figura acima, o apontamento dos blocos: Bloco 4 aponta para o 7, o bloco 7 aponta para o bloco 2 e assim sucessivamente. De uma maneira diferente da alocação contígua, todo o bloco pode ser utilizado. Aqui nesse método nenhum espaço é perdido pela fragmentação (com exceção pela fragmentação interna do último bloco), pelo fato de que os blocos livres não precisam estar contíguos. O que ocorre então é a fragmentação de arquivos, que é a quebra do arquivo em diversos pedaços denominados *extFents*.

A alocação encadeada apenas permite que se faça acesso sequencial aos blocos dos arquivos. Nesse caso, isso é uma desvantagem dessa técnica, visto que não é possível o acesso direto aos blocos. Outro fato importante também, é que essa técnica desperdiça espaço nos blocos com o armazenamento com ponteiros.

Alocação indexada

A alocação indexada nos traz a solução para uma das principais limitações da alocação encadeada, que é a impossibilidade do acesso direto aos blocos dos arquivos. O princípio dessa técnica é manter os ponteiros de todos os blocos do arquivo em uma única estrutura denominada bloco de índice, veja a figura 4, abaixo. Além disso, não desperdiça espaço nos blocos de dados com controle.

Figura 4 - Alocação indexada



Fonte: Machado; Maia (2000, p. 163)

Videoaula 2

Agora assista ao vídeo no qual abordaremos os conceitos sobre alocações de arquivo.



Videoaula 2

Utilize o QRcode para assistir!

Agora assista ao vídeo no qual abordaremos os conceitos sobre alocações de arquivo.



Indicação de Vídeo

Assista ao vídeo abaixo, no qual você pode complementar os conceitos referentes a implementação de arquivos. Disponível em:

<https://www.youtube.com/watch?v=BC5bbCaYDzo>. Acesso em: 30 abr. 2021.

Leitura Obrigatória

Para aprofundar seu conhecimento sobre o assunto de implementação de arquivos, leia no livro *Sistemas Operacionais Modernos* de Andrew Tanenbaum, em especial o **tópico 4.3.2**, **p. 170**. Disponível em:

<https://plataforma.bvirtual.com.br/Leitor/Publicacao/1233/pdf/0?code=p/KENCsgBmm2XPJQdgcexDkGZmvjAVKHbpzj0tMmGYA8Ts2hX6evqP6yLlzxGT4F7zlRrk5QWbZg4rNHijyBA==>. Acesso em: 30 abr. 2021.

Proteção de Acesso à arquivos e diretórios

Até o momento discutimos o funcionamento dos arquivos e diretórios em um sistema operacional. Agora precisamos discutir ideias que são usadas pelos sistemas operacionais para manter a proteção individual de arquivos e diretórios, haja vista que esses são compartilhados. Qualquer sistema de arquivos deve possuir mecanismos próprios que ajude a proteger o acesso as informações gravadas no disco.

O tipo de acesso é mediante concessão ou não de acessos que podem ser realizados como a leitura (*read*), gravação (*write*), execução (*execute*) e eliminação (*delete*) (MACHADO; MAIA, 2000). Observe uma sumarização dos tipos de acesso que está no quadro abaixo.

Quadro 1 – Tipos de acesso

Acesso	Descrição
Leitura	Qualquer tipo de operação em que o arquivo possa ser visualizado, como a exibição de seu conteúdo, edição ou cópia de um novo arquivo.
Gravação	Alteração no conteúdo do arquivo, como inclusão ou alteração de registros.
Execução	Associado a arquivos executáveis ou arquivos de comandos, indicando o direito de execução do arquivo.
Eliminação	Permissão para se eliminar um arquivo.

Fonte: Machado; Maia (2000)

Podem existir diferentes mecanismos e níveis de proteção, no qual cada um tem suas vantagens e desvantagens, em que a adequação vai depender do sistema utilizado. Vamos apresentar três diferentes mecanismos de proteção que está presente na maioria dos sistemas de arquivos.

Um dos mais comuns mecanismos de proteção a arquivos e pastas é a associação de uma senha de acesso. O controle de acesso se resume ao usuário ter o conhecimento de uma senha e, como consequência, terá a liberação do acesso ao arquivo concedida pelo sistema. Cada arquivo possui apenas uma senha e o acesso pode ter diversos níveis. Como desvantagem temos o compartilhamento, porque todos os demais usuários precisam conhecer a senha de acesso.

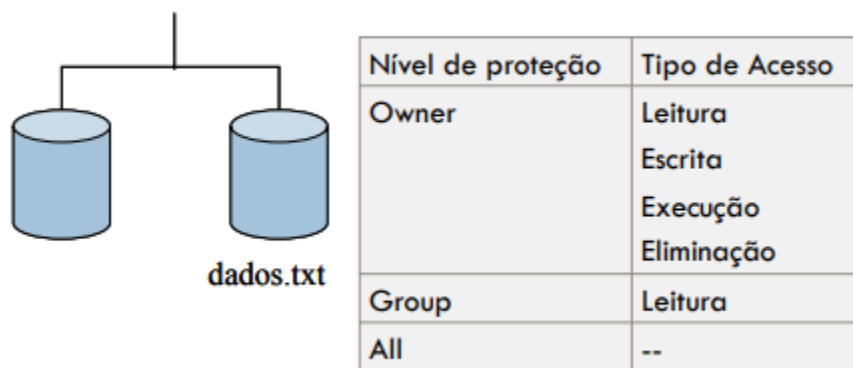
Outro mecanismo de proteção é o grupo de usuários. Essa técnica é bem conhecida de diversos sistemas operacionais. Consiste em associar cada usuário a um grupo de usuários que compartilham arquivos e diretórios. Esse mecanismo implementa três níveis de proteção ao arquivo a saber: *owner* (dono), *group* (grupo) e *all* (todos).

Na criação do arquivo, por exemplo, o usuário deve especificar se o arquivo deve ser acessado apenas pelo seu criador, pelos usuários de um grupo ou por todos os usuários do sistema. Nessas especificações é necessário associar o tipo de acesso (leitura, escrita, execução e eliminação) aos três níveis de proteção.

Como exemplo, observe a figura 5, na qual trata da proteção do arquivo **dados.txt**. Nessa proteção, é liberado completamente o acesso para o dono, permite apenas

compartilhamento de leitura dentro do grupo e não libera qualquer tipo de acesso para os demais usuários do sistema.

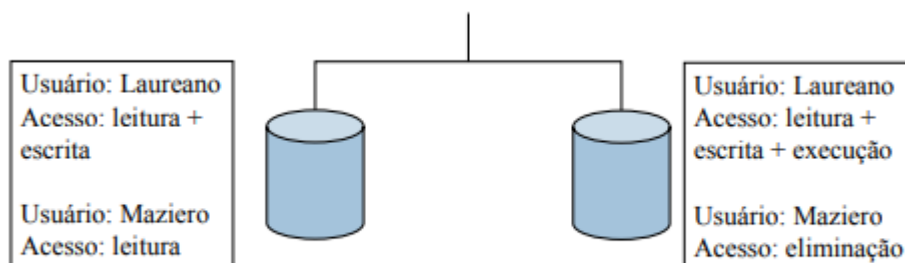
Figura 5 - Proteção por grupo de usuários



Fonte: Machado; Maia (2000, p. 165)

Veremos agora o terceiro mecanismo de proteção: Lista de Controle de Acesso (*Acess Control List* - ACL). Essa lista compõe-se de uma lista associada a cada arquivo, especificando usuários e tipos de acesso permitido, veja figura 6, abaixo. No caso, o Sistema Operacional verifica se a lista de controle autoriza a operação desejada pelo usuário. Por conta disso também, esse sistema poderá ter uma estrutura de tamanho bastante extenso se formos levar em conta que um determinado arquivo pode ter seu acesso compartilhado por diversos usuários.

Figura 6 - Lista de controle de Acesso

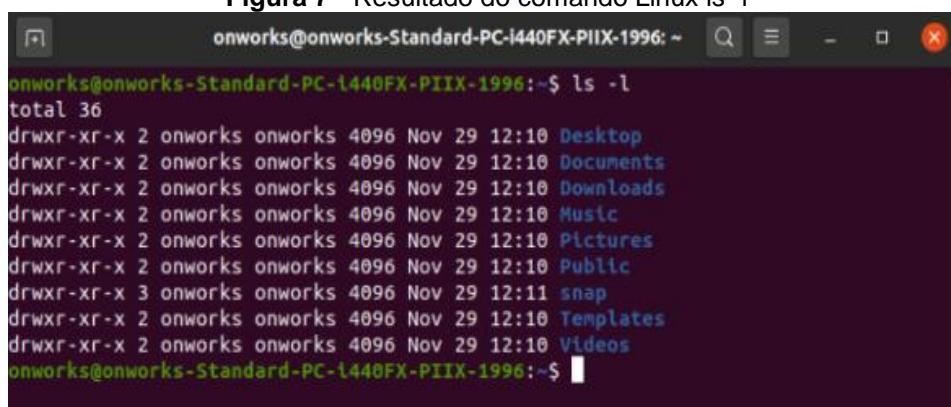


Fonte: Machado; Maia (2000, p. 165)

Como exemplo do uso na proteção de arquivos, veremos, como exemplo, o funcionamento do sistema operacional Linux tratando dessa questão. O teste exemplo foi realizado em um sistema operacional Linux *on-line*. Então, caro aluno! Você mesmo pode acessar e testar esse sistema operacional. Acesse: <https://www.onworks.net/os-distributions/ubuntu-based>.

Os arquivos no Linux podem ter permissões para leitura *r*, escrita *w* e execução *x*. Essas permissões podem ser concedidas ao dono do arquivo, ao grupo de usuários, e também, para outros usuários. Com o comando `ls -l` que exibe detalhes dos arquivos listados, podemos verificar a listagem de arquivos com todas as informações de permissões.

Figura 7 - Resultado do comando Linux `ls -l`



```
onworks@onworks-Standard-PC-l440FX-PIIX-1996: ~$ ls -l
total 36
drwxr-xr-x 2 onworks onworks 4096 Nov 29 12:10 Desktop
drwxr-xr-x 2 onworks onworks 4096 Nov 29 12:10 Documents
drwxr-xr-x 2 onworks onworks 4096 Nov 29 12:10 Downloads
drwxr-xr-x 2 onworks onworks 4096 Nov 29 12:10 Music
drwxr-xr-x 2 onworks onworks 4096 Nov 29 12:10 Pictures
drwxr-xr-x 2 onworks onworks 4096 Nov 29 12:10 Public
drwxr-xr-x 3 onworks onworks 4096 Nov 29 12:11 snap
drwxr-xr-x 2 onworks onworks 4096 Nov 29 12:10 Templates
drwxr-xr-x 2 onworks onworks 4096 Nov 29 12:10 Videos
onworks@onworks-Standard-PC-l440FX-PIIX-1996: ~$
```

Fonte: o autor (2021)

Analisando a figura 7, temos na primeira linha do resultado do comando, o diretório *Desktop*. Observe os caracteres mostrados que são: 10 **drwxr -xr -x**. A primeira letra “d” diz qual é o tipo do arquivo. Caso tiver um “d” é um diretório.

Os três caracteres seguintes “*rw*x” dizem quais são as permissões do usuário, neste caso *onworks*, ele tem a permissão de ler (*r* - *read*), gravar (*w* - *write*) e executar (*x* - *execute*) o diretório *desktop*. Os próximos 3 caracteres “*r-x*” indicam que todos os usuários que pertencem ao grupo *onworks* tem a permissão de ler (*r*), e também executar (*x*). Por fim as três últimas (*r-x*), indica que todos os usuários que não são donos do diretório têm a permissão somente para ler e executar.

Então, para acessar determinado arquivo/diretório é primeiramente verificado se o usuário que acessará o arquivo é o seu dono, caso seja, as permissões de dono do arquivo são aplicadas. Caso não seja o dono do arquivo/diretório, é verificado se ele pertence ao grupo correspondente, caso pertença, as permissões do grupo são aplicadas. Caso não

pertença ao grupo, são verificadas as permissões de acesso para os outros usuários que não são donos e não pertencem ao grupo correspondente ao arquivo/diretório (SILVA, 2010).

Como notou, o sistema operacional Linux utiliza o conceito de proteção por grupos de usuários. Agora, para se especializar mais nos sistemas de permissão de arquivos dos sistemas operacionais, convido a você aluno a pesquisar sobre como funciona as permissões no sistema operacional *Windows*.

Boa pesquisa e um grande abraço!

Videoaula 3

Agora assista ao vídeo no qual abordaremos sobre a proteção de acesso.



Videoaula 3

Utilize o QRcode para assistir!

Agora assista ao vídeo no qual abordaremos sobre a proteção de acesso.



Indicação de Vídeo

Assista ao vídeo que aborda sobre os sistemas de arquivos. Disponível em: <https://www.youtube.com/watch?v=BWtLgfWFZfl>. Acesso em: 30 abr. 2021.

Encerramento da Unidade

Opa! Chegamos ao final de mais uma unidade. Espero que você tenha aprendido muita coisa aluno. Nessa unidade continuamos com a sequência sobre os conceitos de sistema operacionais.

Estudamos aqui o assunto mais conhecido dos usuários a respeito de sistemas operacionais: Arquivos. Vimos na aula 01 que arquivos são informações logicamente relacionadas e que geralmente para funcionar precisam ter um nome e um atributo.

Devem ser organizados para um melhor gerenciamento e, por fim precisam ter métodos de acesso para que o sistema operacional leia suas informações. Podemos pensar que arquivos são um elemento essencial de qualquer sistema. Ainda dentro da ideia de arquivos, estudamos o diretório. Os diretórios são a maneira de como o sistema operacional organiza de maneira lógica os diversos arquivos contidos em disco.

Na aula 02, estudamos como são alocados arquivos em um disco e descobrimos que existem algumas técnicas para isso. Cada técnica tem sua vantagem e desvantagem. Com isso em mente, notamos que arquivos e diretórios pertencem ao que chamamos de sistema de arquivo de um sistema operacional. Por fim, aprendemos que os arquivos devem ter proteção para que os seus dados não sejam violados, por isso, estudamos proteção de acesso. Espero aluno que tenha gostado do conteúdo da unidade, mas sempre lembre, para ampliar seus conhecimentos sobre o tema abordado faça as leituras indicadas na unidade. Um grande abraço e até mais!

Referências

DEITEL, Harvey M. *et al.* **Sistemas Operacionais**. 3. ed. São Paulo: Pearson Education Brasil, 2005. 784 p.

MACHADO, Francis Berenger; MAIA, Luiz Paulo. **Arquitetura de Sistemas Operacionais**. 2. ed. Rio de Janeiro: Ltc, 2000. 232 p.

SILVA, Gleydson Mazioli da. **Guia Foca GNU/Linux**. 2010. Disponível em: <https://www.ppgia.pucpr.br/pt/arquivos/techdocs/linux/foca-intermediario/index.html#contents>. Acesso em: 19 maio 2021.

SIQUEIRA, Fernando de. **Gerência de Memória**. Disponível em: <https://sites.google.com/site/proffernandosiqueiraso/aulas/9-gerencia-de-memoria>. Acesso em: 27 fev. 2021.

TANENBAUM, Andrew S. **Sistemas Operacionais Modernos**. 4. ed. São Paulo: Pearson Education do Brasil, 2015. 864 p.

WIKIPÉDIA (comp.). **Master Boot Record**. 2021. Disponível em: https://pt.wikipedia.org/w/index.php?title=Master_Boot_Record&oldid=60571177. Acesso em: 07 maio 2021.



UNIFIL.BR