**Introdução da aula**



**Qual é o foco da aula?**

Nesta aula, vamos aprender sobre como é realizada a implementação do Sistema de Arquivos e os métodos de implementação de Arquivos e Diretórios, como é feito o gerenciamento de espaço em disco e a importância da segurança e confiabilidade do Sistema de Arquivos.

**Objetivos gerais de aprendizagem**

Ao longo desta aula, você irá:

* descrever os principais critérios para implementação de arquivos utilizados nos sistemas operacionais;
* explicar os métodos para controle do armazenamento dos arquivos em disco;
* analisar os processos para segurança e confiabilidade do sistema operacional, para manter a integridade dos arquivos.

**Situação-problema**

Existem vários métodos de implementação do sistema de arquivos e diretórios, e nesta aula conheceremos alguns deles. Por exemplo, o Netflix possui um sistema de arquivos onde são armazenados os filmes e séries que são exibidos por streaming ao cliente. Além disso, possui um algoritmo de recomendação que determina as sugestões de filmes e séries para cada usuário.

Relembrando nosso contexto, você foi contratado como engenheiro da computação pela empresa XYZ Ltda., que atua no ramo de seguros automotivos e residenciais, com filiais em todo o país e que deseja organizar os dados e implementar um sistema de arquivos.

A empresa XYZ Ltda. é segmentada nas seguintes áreas: pré- vendas, responsável por captar os clientes que possuem um desejo de obter um seguro; vendas, área que efetiva o seguro do cliente; faturamento, que é responsável pelo faturamento de todas as filiais; e administrativo, responsável pela emissão e envio dos boletos aos clientes e pela comissão de cada funcionário de vendas.

Ela precisa que os dados contidos em seus arquivos sejam protegidos para que somente as pessoas ou setores autorizados tenham acesso a eles. Dessa forma, surgem os seguintes questionamentos:

* qual método você utilizá na implementação de arquivos na empresa XYZ Ltda.?
* como você fará o gerenciamento de espaço em disco destes arquivos?
* quais são os mecanismos de proteção que podem ser implementados na empresa XYZ Ltda., considerando os sistemas operacionais Linux e Windows?

Para que consiga responder a esses e outros questionamentos sobre a implantação e segurança do sistema de arquivos, vamos conhecer mais sobre eles, e você conseguirá elaborar um anteprojeto que deverá ser entregue para a avaliação dos gestores da empresa, apresentando todos os dados obtidos no trabalho realizado.

Bons estudos!

**Métodos de implementação de arquivos e diretórios**



Para cada sistema operacional existem diferentes métodos de implementação de arquivos e diretórios, sendo importante levar em consideração como será realizado o controle do armazenamento dos arquivos em disco.

**Implementação do Sistema de Arquivos**

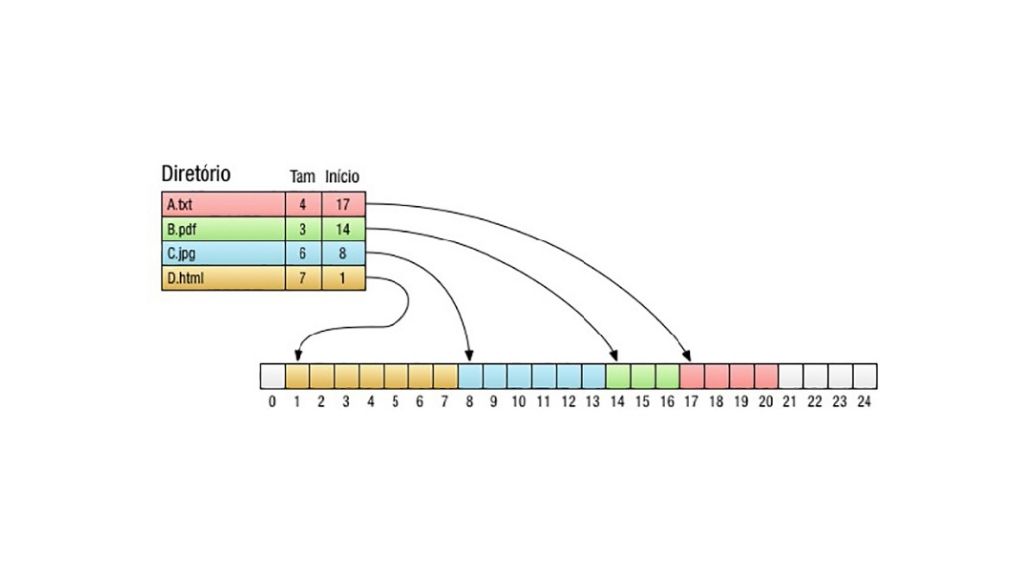
Segundo Machado e Maia (2007), o sistema operacional precisa controlar quais áreas ou blocos no disco estão livres quando um arquivo é criado. A seguir serão apresentados os principais métodos de implementação de arquivos utilizados nos sistemas operacionais.

**Alocação Contígua**: segundo Tanenbaum (2003), a implementação de arquivos por alocação contígua (figura “Alocação Contígua”) é o método mais simples, em que os arquivos são armazenados de forma sequencial no disco. Assim, se você tem um disco rígido com blocos de tamanho 1MB, por exemplo, e um arquivo cujo tamanho seja de 40MB, você utilizará 40 blocos sequenciais de disco para alocar o arquivo, e assim por diante.

A implementação desse método é simples e possui um bom desempenho, porém, como desvantagem, gera a fragmentação do disco, ou seja, os arquivos ficam espalhados por todo disco rígido do computador, o que compromete o desempenho, e ainda é necessário especificar o tamanho do arquivo durante a criação.

Na figura “Alocação Contígua” são apresentados quatro arquivos, com seus nomes, o tamanho dos blocos de disco e o respectivo bloco onde começará a alocação do arquivo.

Por exemplo, o arquivo D.html inicia a alocação a partir do bloco de disco 1, com tamanho de 7 blocos de disco.

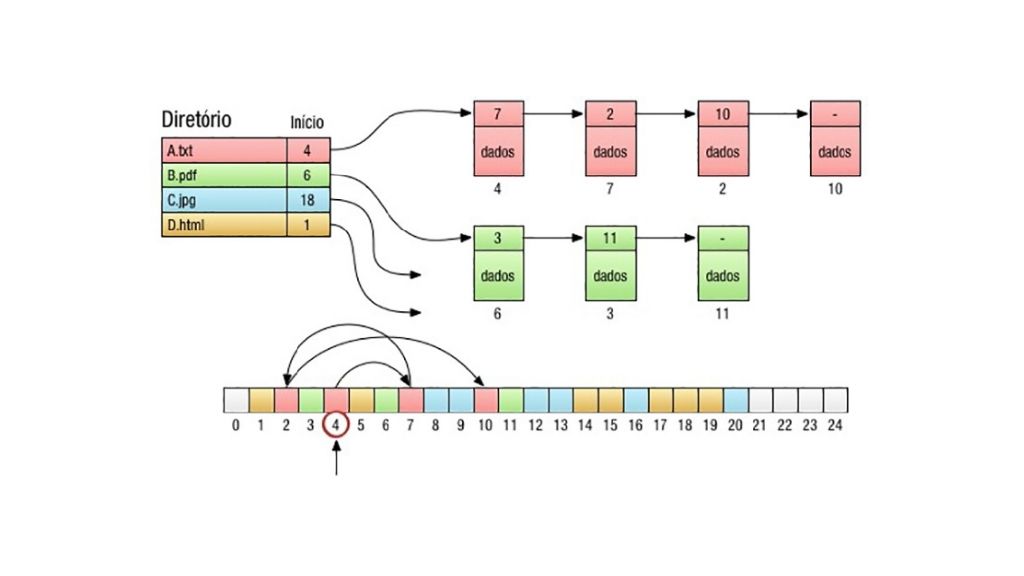
Alocação Contígua. Fonte: Junior ([s.d.]).

**Alocação por Lista Encadeada**: segundo Machado e Maia (2007), outro método de armazenamento de arquivos consiste em organizar os mesmos como um conjunto de blocos ligados logicamente no disco, independentemente de sua localização física. Cada bloco contém um ponteiro para o bloco seguinte do arquivo.

A figura “Alocação por Lista Encadeada” apresenta a implementação de alocação por lista encadeada. Nela são apresentados quatro arquivos, com seus respectivos nomes e o bloco onde começará a alocação do arquivo.

Segundo Junior ([s.d.]), cada bloco do arquivo é composto por dados e pelo endereço do próximo bloco onde está a continuação dos dados.

Por exemplo, o arquivo A.txt se inicia na posição 4 para o primeiro bloco e contém o endereço para o próximo bloco do arquivo, que é o bloco 7. O próximo bloco de dados do arquivo será armazenado no bloco 2. O bloco 2 faz referência para o próximo endereço de armazenamento, o bloco 10, e assim por diante.

Alocação por Lista Encadeada. Fonte: Junior ([s.d.]).

Segundo Tanenbaum (2003), neste método não há desperdício de espaço com a fragmentação, além de precisar somente do primeiro endereço do disco para acessar todo o arquivo, pois os demais são encontrados a partir dele.

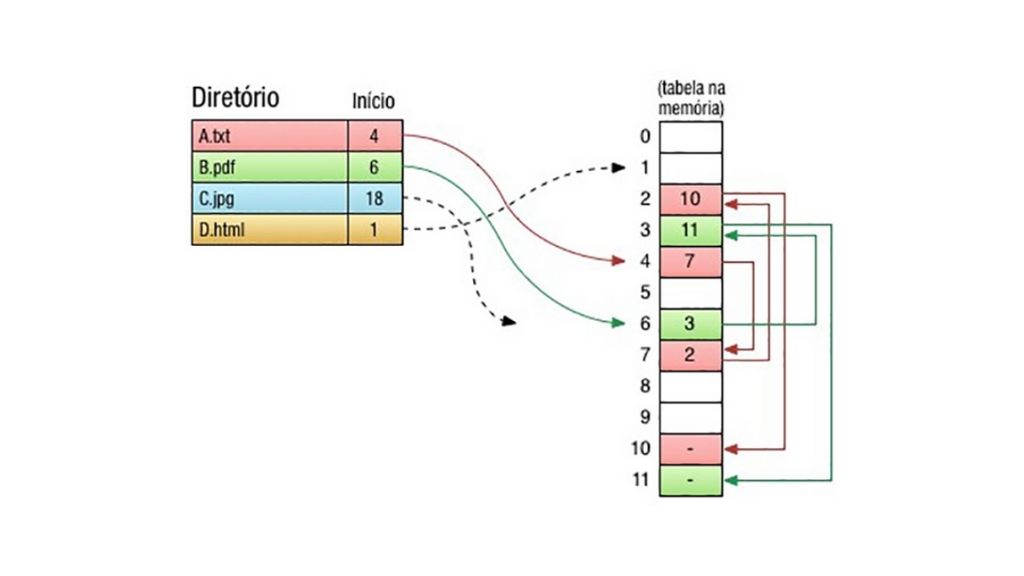
A desvantagem é que o acesso aleatório é lento, além de gastar espaço de memória para armazenar os ponteiros.

**Alocação por Lista Encadeada usando uma tabela na memória**: segundo Tanenbaum (2003), nesse método cada palavra de ponteiro de cada bloco de disco é inserida em uma tabela na memória principal, chamada de FAT (*File Allocation Table*).

A figura “Alocação por Lista Encadeada usando uma tabela na memória” apresenta a implementação de alocação por lista encadeada usando uma tabela na memória, onde são apresentados quatro arquivos, com seus respectivos nomes e o bloco onde começará a alocação do arquivo.

Cada arquivo aponta para uma entrada na tabela de memória, que, por sua vez, contém o endereço do próximo bloco onde está a continuação dos dados.

Por exemplo, o arquivo A.txt aponta para a entrada 4 da tabela de memória, e esse endereço contém o endereço para o próximo bloco do arquivo, que é o bloco 7. A entrada 7 da tabela de memória faz referência para o próximo endereço de armazenamento, o bloco 2, e assim por diante.

Alocação por Lista Encadeada usando uma tabela na memória. Fonte: Junior ([s.d.]).

A vantagem desse método é que o acesso aleatório se torna mais fácil pela tabela estar carregada na memória. Porém, como desvantagem, a tabela deve estar na memória o tempo todo para funcionar.

\_\_\_\_\_\_

**📝 Exemplificando**

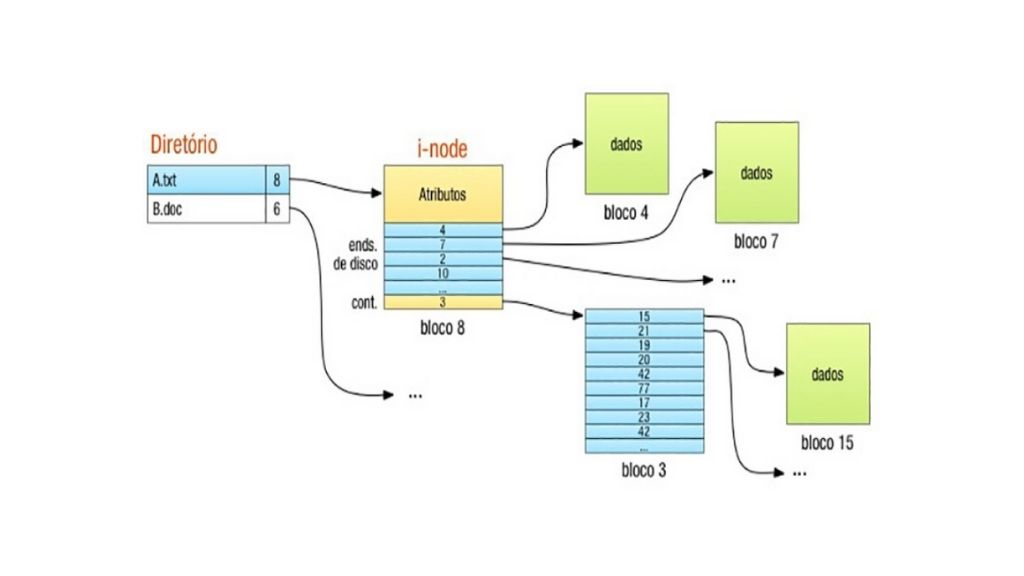
A desvantagem do método de alocação por lista encadeada usando uma tabela na memória é que a tabela deve estar na memória o tempo todo para funcionar.

Por exemplo, para um disco de 20GB e blocos de 1KB, a tabela precisará de 20 milhões de entradas, uma para cada um dos 20 milhões dos blocos de disco, ou seja, a tabela pode ser muito grande para ser mantida em memória.

\_\_\_\_\_\_

**I-nodes**: segundo Tanenbaum (2003), nesse método cada arquivo é associado a uma estrutura chamada I-node (index-node), relacionando os atributos e os endereços em disco dos blocos de arquivos. Com o I-node é possível encontrar todos os blocos de arquivos.

Por exemplo, o arquivo A.txt foi associado a uma tabela I-node (figura “ I-nodes”), com os seus atributos e endereços de disco, que fazem referência aos blocos de arquivos.

I-nodes. Fonte: Junior ([s.d.]).

Nesse método, a tabela precisa ser carregada somente quando o arquivo correspondente estiver aberto. Uma desvantagem é que se existirem arquivos que precisam crescer além do esperado, seria necessário ter dois ou mais endereços de disco apontando para outros blocos de disco cheios de endereços.

\_\_\_\_\_\_

**💭 Reflita**

Você sabia que num determinado dispositivo de armazenamento o I-node pode ocupar até 1% do espaço total (LINUXANDO, 2018)? A quantidade de arquivos e diretórios influenciam sobre a quantidade de I-nodes?

\_\_\_\_\_\_

Esse método é utilizado no sistema operacional Unix. No Linux, o método I-node armazena num disco rígido, por exemplo, as informações de permissões de acesso, identificação dos donos dos arquivos, o tamanho do arquivo e os ponteiros.

Segundo Couto (2018), o núcleo do sistema operacional, ao ler um arquivo ou diretório, verifica as permissões contidas no I-node do arquivo e, caso sejam negadas, o usuário não terá acesso às suas informações. Somente após a verificação das permissões do I-node é que são validadas as permissões dos arquivos ou diretórios.

\_\_\_\_\_\_

**➕ Pesquise mais**

Cada entrada de registo de um I-node tem um tamanho de 128 bytes, e as informações armazenadas podem incluir: o número do I-node, o controle de acesso, as propriedades ou atributos, o número dos blocos do dispositivo (localização no disco), entre outros.

Para saber mais sobre esse assunto, consulte este artigo [**Compreender o Inode em Linux**](https://www.linuxando.com/tutorial.php?t=Compreender%20o%20Inode%20em%20Linux_7).

**Implementação de Arquivos e Diretórios**

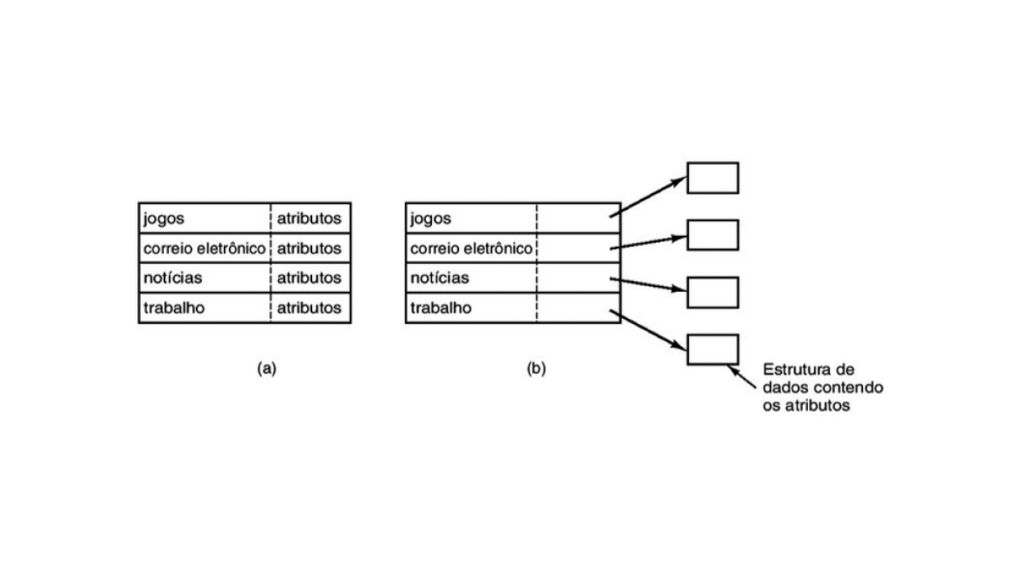


Segundo Tanenbaum (2003), para localizar a entrada de um diretório, o sistema operacional usa o nome do caminho do arquivo dado pelo usuário. A entrada de um diretório contém a informação necessária para encontrar os blocos de disco, que pode ser: **o endereço do disco do arquivo, o número do primeiro bloco, o número do I-node.**

A função principal do sistema de diretório é mapear o nome do arquivo na informação necessária para localizar os dados. Os atributos do arquivo (proprietário do arquivo e data da criação, por exemplo) são armazenados na entrada do diretório.

A figura “Atributos na entrada do diretório (a)” apresenta um diretório com entradas de tamanho fixo, contendo um nome, os atributos e os endereços de discos.

Os sistemas que utilizam I-nodes podem armazenar os atributos em I-nodes, assim, as entradas de diretórios conterão um nome de arquivo e um número I-node, conforme apresentado na figura “Atributos em I-nodes (b)”

Atributos na entrada do diretório (a) e atributos em I-nodes (b). Fonte: Tanenbaum (2003, p. 305).

Ainda segundo Tanenbaum (2003), os sistemas operacionais modernos suportam nomes de arquivos mais longos e com tamanhos variáveis.

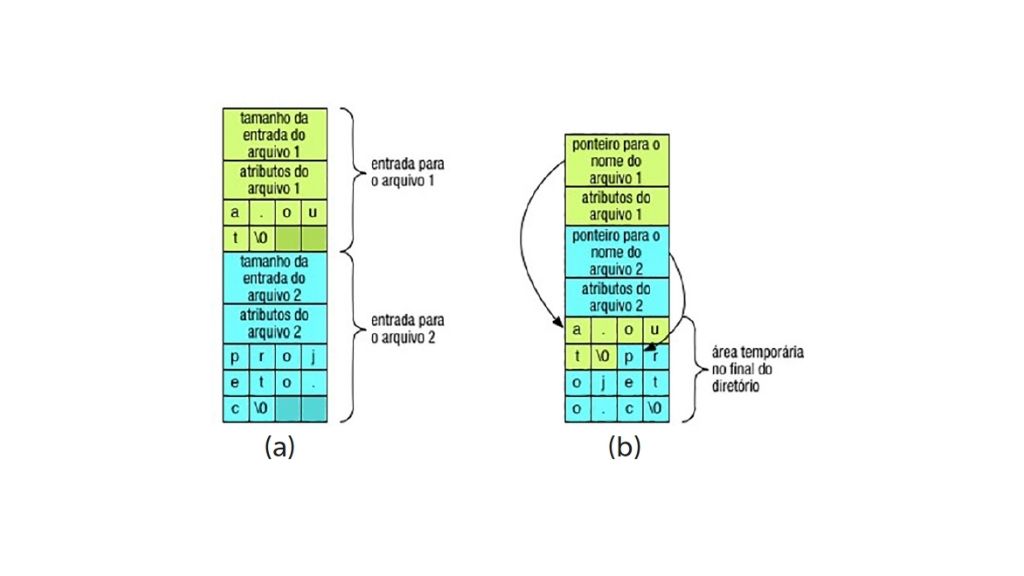
Para fazer essa implementação, é necessário definir um limite de 255 caracteres para o tamanho do nome do arquivo. Essa implementação é simples, mas consome uma quantidade grande e desnecessária de espaço no diretório, uma vez que nem todos os nomes de arquivos são grandes. diretório com dois arquivos (“a.out" e “projeto.c”), com o tamanho da entrada variável e os atributos do arquivo.

Outra forma de tratar nomes de arquivos variáveis é apresentada na figura “Atributos na entrada do diretório em I-nodes” (b). As entradas de diretório são de tamanho fixo e os nomes de arquivos ficam agrupados no final do diretório, em uma área chamada heap. Desta forma, os ponteiros dos arquivos apontam para início dos nomes dos arquivos no final do diretório.

Uma alternativa para esse método é definir a entrada do diretório como uma parte fixa a partir do tamanho da entrada dos arquivos (variável), seguida por dados de formato fixo (proprietário do arquivo, horário de criação, entre outros).

A figura “Atributos na entrada do diretório (a)” apresenta um diretório com dois arquivos (“a.out" e “projeto.c”), com o tamanho da entrada variável e os atributos do arquivo.

Outra forma de tratar nomes de arquivos variáveis é apresentada na figura “Atributos em I-nodes (b)”. As entradas de diretório são de tamanho fixo e os nomes de arquivos ficam agrupados no final do diretório, em uma área chamada heap. Desta forma, os ponteiros dos arquivos apontam para início dos nomes dos arquivos no final do diretório.

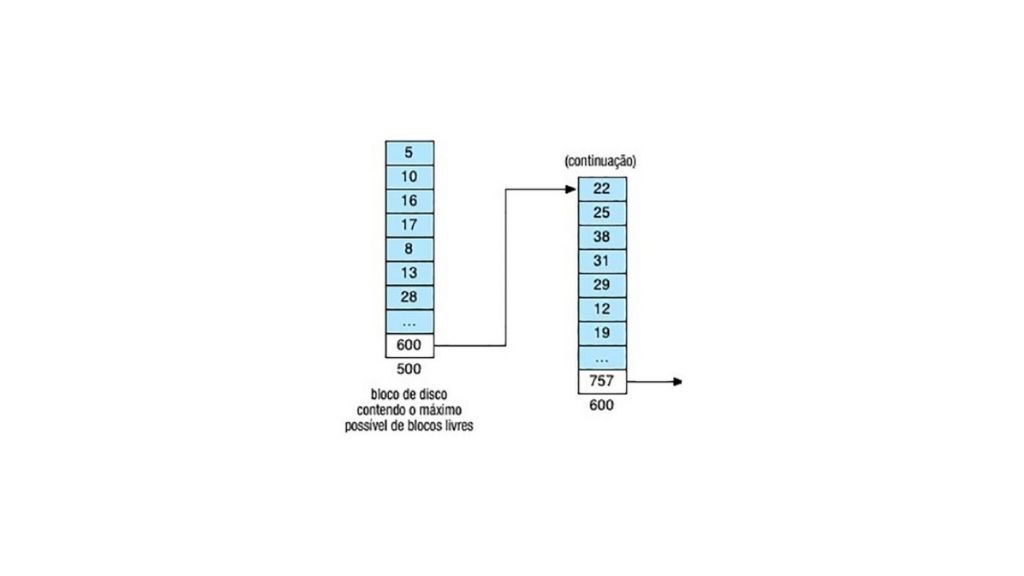
Atributos na entrada do diretório (a) e atributos em I-nodes (b). Fonte: Junior ([s.d.]).

**Gerenciamento de Espaço em Disco**

O sistema operacional precisa gerenciar o espaço em disco dos blocos livres, garantindo um bom desempenho ao sistema. Segundo Tanenbaum (2003), são utilizados dois métodos para monitorar os blocos livres: lista encadeada de blocos e mapa de bits.

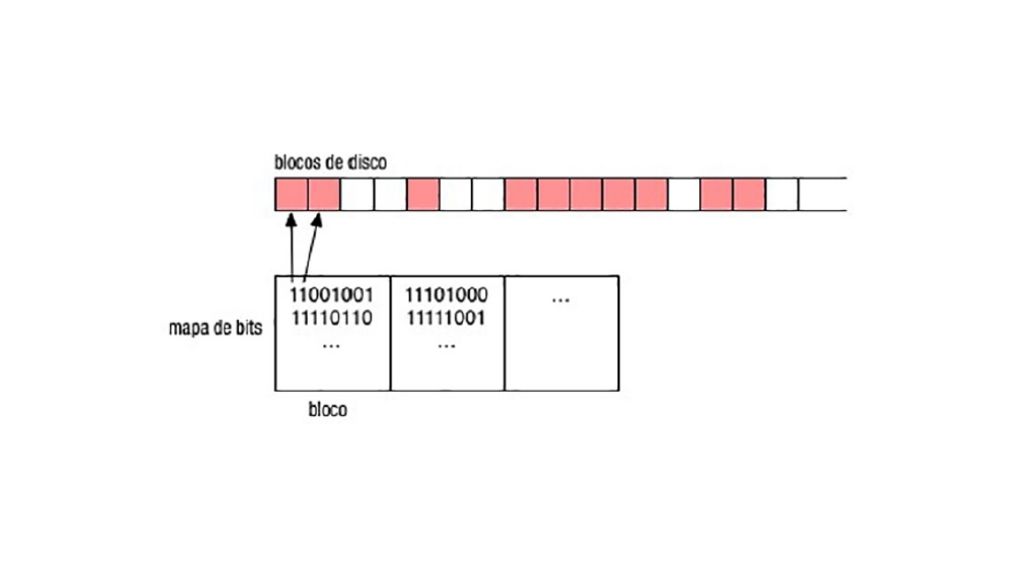
No método da lista encadeada de blocos, cada bloco contem a quantidade de espaços livres que puderem ter e possuem a localização dos blocos livres, conforme apresentado na figura a seguir.

Por exemplo, um disco de 16GB precisa de uma lista livre de 16.794 blocos para conter 224 número de blocos de disco.

Lista encadeada de blocos. Fonte: Junior ([s.d.]).

Segundo Deitel, Deitel e Choffnes (2005), um mapa de bits possui um bit para cada bloco físico do sistema de arquivos. O bit igual a 1 indica que o bloco está em uso, e o bit 0 que não está, conforme apresentado na figura “Mapa de Bits”.

Caso seja necessário alocar mais um bloco físico para o disco, é só percorrer o mapa de bits para encontrar o bit igual a 0.

Mapa de Bits. Fonte: Junior ([s.d.]).

Ainda segundo Deitel, Deitel e Choffnes (2005), a vantagem do mapa de bits em relação à lista encadeada de blocos é que o sistema de arquivos consegue saber rapidamente onde há blocos livres, além de ser simples a sua implementação.

**Segurança e Confiabilidade do Sistema de Arquivos**



Segundo Machado e Maia (2007), os arquivos são compartilhados com usuários e utilizados para diversas finalidades. Assim, é necessário que o sistema operacional ofereça proteção desses arquivos para que usuários e processos que não tenham permissão consigam acessá- los.

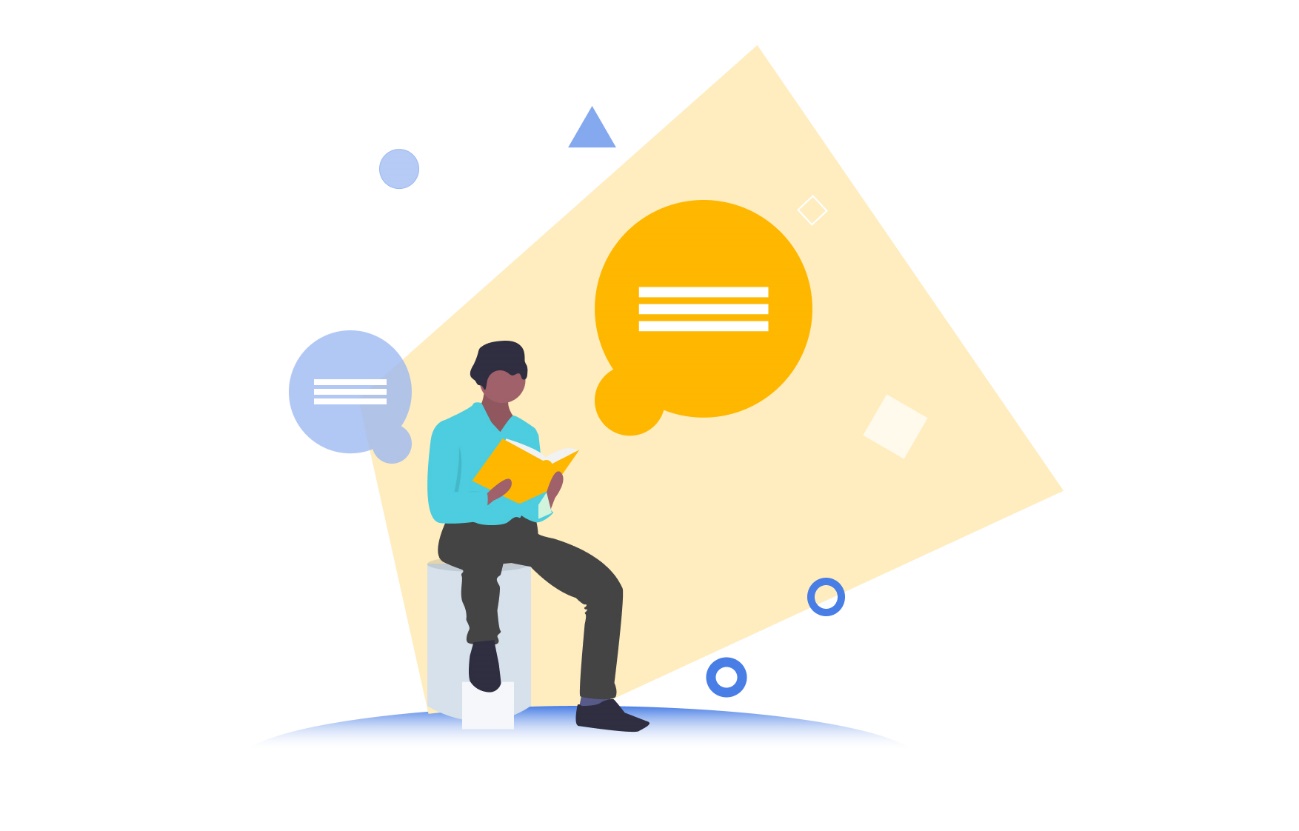
Ainda segundo Machado e Maia (2007), um sistema de arquivos possui diferentes mecanismos de proteção, como:

* **senha de acesso**: para ter acesso ao arquivo, é necessário que o usuário a conheça e que o sistema conceda o acesso ao arquivo. Nesse caso, como um arquivo possui apenas uma senha, não é possível definir quais tipos de operações serão realizadas.
* **grupo de usuários**: essa proteção consiste em associar cada usuário a um grupo para compartilhar arquivos. Durante a criação do arquivo, o usuário define quais usuários terão acesso a ele.
* **lista de controle de acesso (*Access Control List - ACL*)**: é uma lista associada a cada arquivo, com as permissões de cada usuário. Quando um usuário tenta acessar um arquivo, o sistema operacional verifica sua permissão para autorizar ou não a operação realizada.

Segundo Tanenbaum (2003), existem questões de confiabilidade que devem manter a integridade dos arquivos, como cópias de segurança e consistência do sistema de arquivos. As cópias de segurança (*backups*) garantem que os dados sejam copiados e armazenados em diferentes lugares e dispositivos.

A consistência do sistema de arquivos permite que os arquivos sejam salvos antes de finalizar sua tarefa. Assim, caso um usuário esteja modificando um arquivo e a luz acabe, por exemplo, a parte trabalhada até o momento é salva em disco.

**Conclusão**



Agora que você já conheceu como é realizada a implementação do Sistema de Arquivos, os métodos de implementação de Arquivos e Diretórios, como é realizado o gerenciamento de espaço em disco e a importância da segurança e confiabilidade do Sistema de Arquivos, vamos voltar ao nosso contexto.

A empresa XYZ Ltda. precisa que os dados contidos em seus arquivos sejam protegidos, para que somente as pessoas ou setores autorizados tenham acesso a eles. Assim, surgem os seguintes questionamentos:

* qual método será utilizado para a implementação de arquivos na empresa XYZ Ltda.?
* como será feito o gerenciamento de espaço em disco desses arquivos?
* quais são os mecanismos de proteção que podem ser implementados, considerando os sistemas operacionais Linux e Windows?

Existem vários métodos de implementação de arquivos. Neste caso, deve-se analisar qual método será melhor aplicado ao sistema operacional utilizado pelo usuário. A empresa XYZ Ltda. utiliza os sistemas operacionais Linux e Windows.

O Linux, por exemplo, utiliza o método I-node, onde são armazenados num disco rígido ou arquivo as informações de permissões de acesso, identificação dos donos dos arquivos, o tamanho do arquivo e os ponteiros. O núcleo do sistema operacional verifica as permissões contidas no I-node do arquivo antes de verificar as permissões dos arquivos ou diretórios. O Sistema de Arquivos do Linux e do Windows conseguem trabalhar com uma quantidade grande de arquivos, além de gerenciar arquivos que precisam aumentar o consumo de memória durante a execução.

No caso da empresa XYZ Ltda., o mais adequado é gerenciar o espaço em disco por meio do mapa de bits, uma vez que o sistema de arquivos consegue saber rapidamente onde há blocos livres, além de ser simples a sua implementação. O bit igual a 1 corresponde ao bloco que está em uso e o bit 0 ao que não está.

Como a empresa XYZ Ltda. é segmentada nas áreas de pré-vendas, vendas, faturamento e administrativo, é necessário garantir a segurança dos diretórios e arquivos de cada setor, para que somente usuários autorizados consigam acessá-los.

Para implementar os mecanismos de proteção, será necessário utilizar os métodos de senha de acesso, grupo de usuários e lista de controle de acesso. No Linux, cada arquivo possui permissão de acesso para ler, gravar e executar, e os tipos de usuários são dono, grupo e outros usuários do sistema. No Windows, a permissão de acesso aos arquivos e pastas pode ser configurada pelo administrador do sistema para cada usuário.

O administrador dos sistemas Linux e Windows, na hora da criação dos usuários do sistema, deve atribuir a cada um a permissão (leitura, gravação e execução) ao sistema e diretórios. Durante a criação deles, poderão ser inseridos em um grupo, de acordo com as áreas da empresa (pré-vendas, vendas, faturamento e administrativo), de forma que somente os funcionários de uma determinada área poderão incluir, deletar ou editar os arquivos de seu diretório.

Além da senha de acesso e do grupo de usuários, pode ser implementada a lista de controle de acesso (*Access Control List – ACL*), que é associada a cada arquivo com as permissões de cada usuário. Desta forma, a segurança é reforçada, pois quando um usuário tenta acessar um arquivo, o sistema operacional verifica sua permissão para autorizar ou não a operação realizada.

Para finalizar suas atividades, agora você deverá elaborar o anteprojeto que será entregue para a avaliação dos gestores, apresentando as informações levantadas durante a sua atividade na empresa. Ele deverá conter os seguintes dados:

* o levantamento dos documentos que estavam armazenados dentro de caixas no depósito, analisando a estrutura desses arquivos.
* a definição do sistema de arquivos a ser utilizado nos sistemas operacionais Linux e Windows.
* a definição dos nomes, atributos e operações realizadas, com os arquivos e diretórios que atenderão a cada setor da empresa.
* os métodos de implantação de arquivos no Linux e Windows.
* a implantação do mecanismo de proteção, utilizando os métodos de senha de acesso, grupo de usuários e lista de controle de acesso.

**Referências**



ALENCAR, F. Entenda o que é sistema de arquivos e sua utilidade no PC e no celular. **TechTudo**. 26 fev. 2016. Disponível em: <http://www.techtudo.com.br/dicas-e-tutoriais/noticia/2016/02/entenda-o-que-e-sistema-de-arquivos-e-sua-utilidade-no-pc-e-no-celular.html>. Acesso em: 04 jul. 2021.

COTA, A. **Diferenças entre o sistema de arquivos do Windows e Linux**. 17 mar. 2005. Disponível em: <https://www.vivaolinux.com.br/artigo/Diferencas-entre-o-sistema-de-arquivos-do-Windows-e-Linux>. Acesso em: 04 jul. 2021.

COUTO, E. **O que é um inode?** 23 set. 2009. Disponível em: <https://gnulinuxbr. wordpress.com/category/comandos-intermediarios/o-que-e-um-inode/>. Acesso em: 04 jul. 2021.

DEITELL, H. M.; DEITEL, P. J.; CHOFFNES, D. R. **Sistemas Operacionais**. 3. ed. São Paulo: Prentice Hall, 2005.

EASEUS. **Meus Arquivos de Computador Sumiram. Preciso de Ajuda para Encontrá-los**. 8 jan. 2018. Disponível em: <https://br.easeus.com/file-recovery/recuperar-arquivos-e-pastas-sumiram-do-computador-windows-10.html>. Acesso em: 04 jul. 2021.

GALVIN, P. B., GAGNE, G.; SILBERSCHATZ, A. **Operating system concepts**. 8. ed. United States of America: John Wiley & Sons, Inc., 2013.

GUTIERREZ, C. **Backup**: a importância das cópias de segurança. 8 abr. 2018. Disponível em: <http://www.administradores.com.br/noticias/tecnologia/backup-a-importancia-das-copias-de-seguranca/124318/>. Acesso em: 04 jul. 2021.

HAMMERSCHMIDT, R. Microsoft revela mais detalhes sobre o ReFS, seu novo sistema de arquivos. **Tecmundo**. 17 jan. 2012. Disponível em: <https://www.tecmundo.com.br/windows-8/17975-microsoft-revela-mais-detalhes-sobre-o-refs-seu-novo-sistema-de-arquivos.htm>. Acesso em: 10 maio 2018.

JONES, M. **Anatomia do Sistema de Arquivos do Linux**. 30 out. 2007. Disponível em: <https://www.ibm.com/developerworks/br/library/l-linux-filesystem/index.html>. Acesso em: 9 maio 2018.

JUNIOR, J. S. V. **Diretórios do Linux**. 10 set. 2009. Disponível em: <https://www. vivaolinux.com.br/dica/Diretorios-do-Linux>. Acesso em: Acesso em: 04 jul. 2021.

JUNIOR, L. A. P. L. **Sistemas de Arquivos**. [S.d.]. Disponível em: <http://www.ppgia. pucpr.br/~laplima/ensino/so/materia/04\_arquivos.html>. Acesso em: 21 maio 2018.

LINUXANDO. **Compreender o Inode em Linux**. [S.d.]. Disponível em: <https://www. linuxando.com/tutorial.php?t=Compreender%20o%20Inode%20em%20Linux\_7>. Acesso em: 04 jul. 2021

MACHADO, F. B.; MAIA, L. P. **Arquitetura de Sistemas Operacionais**. 4. ed. Rio de Janeiro: LTC, 2007.

PAULO, D. **Fundamentos do sistema Linux - arquivos e diretórios**. 28 out. 2005. Disponível em: <https://www.vivaolinux.com.br/artigo/Fundamentos-do-sistema-Linux-arquivos-e-diretorios/>. Acesso em: 04 jul. 2021

SANTOS, T. **A Arvore de diretórios do Linux**. 13 out. 2016. Disponível em: <https:// fsocietybrasil.org/a-arvore-de-diretorios-do-linux/>. Acesso em: 11 maio 2018.

SHIELDS, I. **Gerenciamento de arquivo e diretório**. 10 nov. 2009. Disponível em:<https://www.ibm.com/developerworks/br/linux/library/l-lpic1-v3-103-3/index. html>. Acesso em: 14 maio 2018.

SILVA, R. R. **Sistemas de arquivos suportados pelo Linux**. 13 nov. 2006. Disponível em: <https://www.vivaolinux.com.br/artigo/Linux-Sistema-de-arquivos>. Acesso em: 04 jul. 2021.

TANENBAUM, A. S. **Sistemas Operacionais Modernos**. 2. ed. São Paulo: Pearson, 2003.

WALTRICK, R. Saiba como proteger seus arquivos salvos na nuvem. **Gazeta do Povo**. 3 set. 2014. Disponível em: <http://www.gazetadopovo.com.br/tecnologia/ saiba-como-proteger-seus-arquivos-salvos-na-nuvem-ecz71ltnny743iyd4u3yeas7i>. Acesso em: 04 jul. 2021.