



Sistemas Operacionais

Sistemas Operacionais

Cynthia da Silva Barbosa

© 2018 por Editora e Distribuidora Educacional S.A.

Todos os direitos reservados. Nenhuma parte desta publicação poderá ser reproduzida ou transmitida de qualquer modo ou por qualquer outro meio, eletrônico ou mecânico, incluindo fotocópia, gravação ou qualquer outro tipo de sistema de armazenamento e transmissão de informação, sem prévia autorização, por escrito, da Editora e Distribuidora Educacional S.A.

Presidente

Rodrigo Galindo

Vice-Presidente Acadêmico de Graduação e de Educação Básica

Mário Ghio Júnior

Conselho Acadêmico

Ana Lucia Jankovic Barduchi

Camila Cardoso Rotella

Danielly Nunes Andrade Noé

Grasiele Aparecida Lourenço

Isabel Cristina Chagas Barbin

Lidiâne Cristina Vivaldini Olo

Thatiane Cristina dos Santos de Carvalho Ribeiro

Revisão Técnica

Francisco Ferreira Martins Neto

Ruy Flávio de Oliveira

Editorial

Camila Cardoso Rotella (Diretora)

Lidiâne Cristina Vivaldini Olo (Gerente)

Elmir Carvalho da Silva (Coordenador)

Leticia Bento Pieroni (Coordenadora)

Renata Jéssica Galdino (Coordenadora)

Dados Internacionais de Catalogação na Publicação (CIP)

Barbosa, Cynthia da Silva

B238s Sistemas operacionais / Cynthia da Silva Barbosa.

– Londrina : Editora e Distribuidora Educacional S.A., 2018.

200 p.

ISBN 978-85-522-1177-8

1. Sistema de arquivos. 2. Shell. 3. Kernel. I. Barbosa, Cynthia da Silva. II. Título.

CDD 005

Thamiris Mantovani CRB-8/9491

2018

*Editora e Distribuidora Educacional S.A.
Avenida Paris, 675 – Parque Residencial João Piza
CEP: 86041-100 – Londrina – PR
e-mail: editora.educacional@kroton.com.br
Homepage: <http://www.kroton.com.br/>*

Sumário

Unidade 1 Introdução aos sistemas operacionais 2	7
Seção 1.1 - Introdução aos sistemas operacionais	9
Seção 1.2 - Evolução dos sistemas operacionais	22
Seção 1.3 - Características e composição dos sistemas operacionais	37
Unidade 2 Processos e Threads 2	55
Seção 2.1 - Processos	57
Seção 2.2 - Comunicação entre processos	71
Seção 2.3 - Escalonamento de processos	88
Unidade 3 Sistema de arquivos 2	103
Seção 3.1 - Arquivos e sistemas de arquivos	105
Seção 3.2 - Sistemas de diretórios	119
Seção 3.3 - Implementação e segurança de sistemas de arquivos	134
Unidade 4 Gerenciamento de dispositivos 2	153
Seção 4.1 - Gerenciamento de memória	155
Seção 4.2 - <i>Swapping</i> : troca de processos	169
Seção 4.3 - Memória virtual	183

Palavras do autor

Caro aluno, seja bem-vindo à disciplina de Sistemas Operacionais! Aqui iremos apresentar um pouco do universo dos sistemas operacionais, que têm por objetivo gerenciar os recursos de hardware e fazer a interação entre o hardware e o software, além de ser o responsável por conectar o hardware e o usuário do computador. Inicialmente, iremos abordar os conceitos e o histórico dos sistemas operacionais. Neste tema, trataremos da importância dos sistemas operacionais, as principais funcionalidades e serviços. Partiremos para a estrutura e os tipos dos sistemas operacionais monoprogramáveis, multiprogramáveis e multiprocessamento. Seguiremos com a abordagem sobre as características, a composição e os exemplos de sistemas operacionais do mercado, além de estudarmos sobre Linux e Windows, mostrando suas características e diferenças. Saber sobre os sistemas operacionais e as abordagens em diferentes plataformas, como o Windows e o Linux, é um conhecimento imprescindível na vida profissional, pois é através deles que haverá um melhor entendimento sobre a importância e a complexidade deste software.

Este livro didático será dividido em quatro unidades:

Na primeira, você conhecerá desde a definição, conceito, histórico e evolução até os tipos de sistemas operacionais.

Na segunda unidade, você aprenderá sobre os processos, como é feita a comunicação e o escalonamento de processos.

Já na terceira unidade, você conhecerá sobre os arquivos, os diretórios e a implementação e segurança nos sistemas de arquivos.

E, por fim, na quarta unidade, você aprenderá sobre gerenciamento de memória, swapping e memória virtual.

Convido você a se dedicar ao estudo para que, no final da disciplina, esteja apto a conhecer e aplicar os conhecimentos em sistemas operacionais. Que a leitura deste material e a prática do conteúdo possam auxiliá-lo a melhorar o entendimento sobre os sistemas operacionais e seu funcionamento.

Bons estudos!

Introdução aos sistemas operacionais 2

Convite ao estudo

Caro, aluno, seja bem-vindo!

Nesta unidade veremos as diferentes aplicações dos sistemas operacionais e como seus diferentes tipos são utilizados em diversas situações, adequando-se às necessidades dos usuários. Os conceitos de sistemas operacionais como segurança, permissão de acesso, armazenamento de dados, recuperação das informações, dentre outros estão presentes em todas as ramificações da área da tecnologia da informação, e o conhecimento em sistemas operacionais aperfeiçoa as habilidades do profissional de TI em sua área de atuação. Por exemplo:

- Na área de banco de dados – armazenamento e acesso às informações.
- Em programação de sistemas – quando um programa é desenvolvido, é necessário definir a permissão de acesso dos usuários para garantir a segurança do sistema.
- Na área de redes – performance e segurança da rede.

Vamos começar a estudar o universo dos sistemas operacionais, sua história e suas características. Após o término desta unidade, você terá condições de entender os recursos essenciais dos principais sistemas operacionais. Para isso, você estudará a definição, os conceitos, a história e a evolução deles, desde o primeiro computador até os atuais, além de aprender sobre as funções e os serviços oferecidos pelo sistema operacional. Aprender sobre os sistemas operacionais proporcionará um conhecimento que envolve várias áreas da computação, fazendo com que você, aluno, se interesse e queira cada vez mais conhecer e ser capaz de utilizar os recursos essenciais dos principais sistemas operacionais.

Neste contexto, imagine que um professor de informática da escola municipal de sua cidade te convidou para dar um curso durante um simpósio “Informática para todos” entre as escolas municipais da região. O objetivo do evento é ampliar os conhecimentos dos alunos na área de tecnologia. O tema do curso escolhido para você foi Sistemas Operacionais, no qual será abordado desde a definição e os conceitos, avançando até as características do Windows e do Linux. Ao realizar o planejamento juntamente com o professor de informática da escola, ficou definido que seriam abordados os seguintes conteúdos: definição, conceitos e um breve histórico dos sistemas operacionais e sua evolução até os dias atuais, além das principais funções e serviços; a estrutura e os tipos de sistemas operacionais; as características e a composição dos sistemas operacionais mostrando as propriedades do Linux e do Windows, exemplificando cada um. O professor de informática relata a você que às vezes escuta dos alunos os seguintes questionamentos: qual a importância dos sistemas operacionais para o avanço tecnológico? E qual a sua aplicabilidade para o ensino? Por que devemos incluir a informática em sala de aula? Ao final do curso, os alunos farão um relatório dos temas abordados utilizando um editor de texto e seguindo os padrões de formatação da ABNT, apresentando as principais características dos sistemas operacionais Linux e Windows.

Vamos juntos conhecer mais sobre os sistemas operacionais e suas aplicações!

Seção 1.1

Introdução aos sistemas operacionais

Diálogo aberto

Os Sistemas Operacionais evoluíram com o progresso dos computadores. No início, não existiam sistemas operacionais e os computadores eram manipulados manualmente. Atualmente, executam diversas funções e oferecem serviços que auxiliam o usuário e os programadores na execução do trabalho, tomando para si as funções de controlar o hardware disponível. Nessa seção, você conhecerá os conceitos e a história dos sistemas operacionais e suas particularidades, além de iniciar também o aprendizado sobre as funções e os serviços que eles oferecem.

Relembrando nosso contexto sobre o curso de sistemas operacionais ministrado por você durante o simpósio “Informática para todos” entre as escolas municipais da região, você organizou os alunos em grupos de discussão, disponibilizou alguns artigos abordando questões sobre definição, conceito, história e principais funções e serviços dos sistemas operacionais. Posteriormente à leitura do artigo, durante a realização de um debate, você é questionado quanto à importância dos sistemas operacionais. Ainda, surgem as seguintes perguntas: qual o papel do sistema operacional como um gerenciador de recursos do computador? Qual a principal diferença quanto ao acesso do código fonte do Windows e do Linux? Quais são os serviços oferecidos pelos sistemas operacionais Linux e do Windows? Compile as informações obtidas por meio dos questionamentos, para lhe auxiliar na produção do relatório final. Para que você consiga responder essas e outras dúvidas sobre os sistemas operacionais, nesta seção vamos conhecer mais sobre os sistemas operacionais e os conteúdos pertinentes a este tema.

Bons estudos!

Não pode faltar

Definição e conceitos dos sistemas operacionais

Segundo Tanenbaum (2003), o sistema operacional é parte essencial de qualquer sistema computacional. Se eles não existissem, os sistemas computacionais funcionariam, porém, o usuário teria que saber os detalhes de hardware para utilizar o computador, o que seria bem complexo.

Um sistema computacional é composto por hardware e software. Os hardwares são os componentes físicos do computador, como a CPU (unidade central de processamento), o processador, memória, mouse, teclado, monitor, entre outros. Já o software são programas (conjunto de instruções) instalados no hardware para executar uma determinada tarefa. O sistema operacional é um software responsável por controlar o computador e tem por objetivo gerenciar os recursos de hardware (processador, memória, periféricos do computador como teclado, mouse e impressora, dados, dentre outros) e fazer a interação entre o hardware e o software, além de ser o responsável por conectar o hardware e o usuário do computador, conforme apresentado na Figura 1.1.

Figura 1.1 | Sistema operacional – elo entre hardware e software



Fonte: elaborado pelo autor.

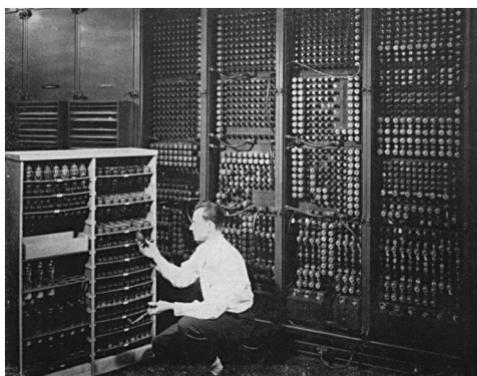
Segundo Machado (1997), quando o computador é ligado, o sistema operacional é o primeiro programa a ser executado e

permanece gerenciando os recursos de hardware e software até o computador ser desligado. O sistema operacional tem o objetivo de gerenciar o computador de forma eficiente e produtiva facilitando o seu uso, além de garantir a integridade e a segurança dos dados durante o processamento e na memória.

Breve histórico dos sistemas operacionais

Segundo Tanenbaum (2003), a evolução dos sistemas operacionais caminhou em conjunto com a evolução da arquitetura dos computadores. A primeira geração de computadores se deu entre 1945 a 1955 (válvulas e painéis de programação). As máquinas eram enormes, lentas e compostas por válvulas, ocupando salas inteiras, e as atividades eram realizadas por uma pessoa por meio de painéis de programação. Naquela época, não existiam sistemas operacionais e nem linguagens de programação, e as máquinas basicamente realizavam cálculos matemáticos como logaritmos, sendo usadas para fins militares. Se durante o processamento de um cálculo um erro acontecesse, o processamento recomeçava do início e, com isso, perdia-se muito tempo. Ainda, se uma válvula queimasse, todo o processamento estava perdido. A Figura 1.2 apresenta as máquinas da primeira geração de computadores.

Figura 1.2 | Primeira geração de computadores

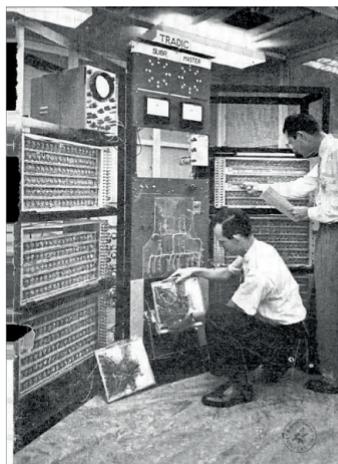


Fonte: <https://commons.wikimedia.org/wiki/File:ENIAC-changing_a_tube.jpg>. Acesso em: 19 abr. 2018.

Segundo Tanenbaum (2003), a segunda geração de computadores foi entre 1955 a 1965 (Transistores e sistema em Lote,

ou *Batch*, em inglês). Nessa época, surgiram os computadores de grande porte, os mainframes, porém somente grandes instituições como bancos e universidades tinham acesso, devido ao alto custo. Os computadores eram utilizados para cálculos científicos como equações diferenciais presentes na física e na engenharia. A Figura 1.3 apresenta as máquinas da segunda geração de computadores. Outro fato é o surgimento das primeiras linguagens de programação Fortran e Assembly.

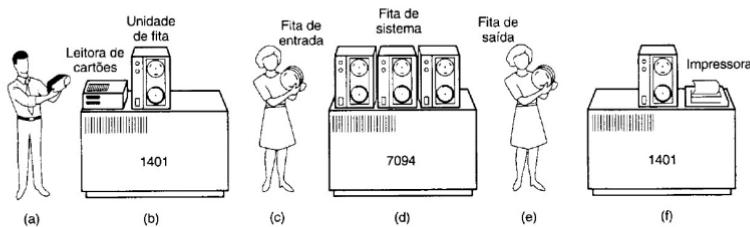
Figura 1.3 | Computador de segunda geração



Fonte: <https://commons.wikimedia.org/wiki/File:TRADIC_computer.jpg>. Acesso em: 19 abr. 2018.

Os mainframes possuíam sistemas operacionais e operavam através de *jobs* (programa ou conjunto de programas). Os programas eram escritos em papel e depois passados para os cartões perfurados. O processamento dos cartões até se obter uma saída era muito demorado e, com isso, surgiu outra solução: o sistema de lote (**batch**), cujo objetivo era gravar vários *jobs* em fita magnética usando um computador relativamente mais barato, bom para ler cartões, copiar fitas e imprimir saídas (TANENBAUM, 2003). A Figura 1.4 apresenta um sistema batch antigo.

Figura 1.4 | Processamento em lote



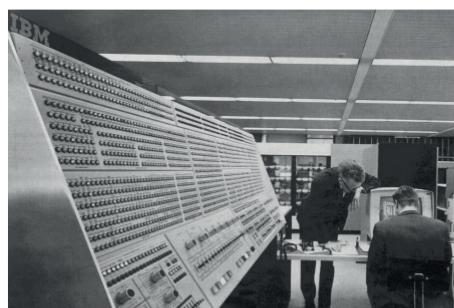
- (a) Os programadores trazem os cartões para o 1401.
- (b) O 1401 lê os Jobs em lote na fita.
- (c) O operador leva a fita de entrada para o 7094.
- (d) O 7094 realiza a computação.
- (e) O operador leva a fita de saída para o 1401.
- (f) O 1401 imprime a saída.

Fonte: Tanenbaum (2003, p. 5)

Segundo Tanenbaum (2003), a terceira geração de computadores compreendeu o período entre 1965 a 1980 (Circuitos Integrados e Multiprogramação). Nesta geração, os fabricantes de computadores ofereciam duas linhas de produtos: os computadores científicos de grande escala orientados a palavras, usados para cálculos numéricos na ciência e na engenharia e os computadores comerciais orientados a caracteres, usados por bancos e companhia de seguros.

Porém, tinha um custo elevado para desenvolver e manter estes produtos. A IBM, para resolver este empasse, desenvolveu o OS/360. As máquinas possuíam a mesma arquitetura e conjunto de instruções compatíveis, permitindo o uso tanto científico quanto comercial, além de um melhor custo benefício. A Figura 1.5 apresenta as máquinas da terceira geração de computadores.

Figura 1.5 | Terceira geração de computadores



Fonte: <<https://commons.wikimedia.org/wiki/File:360-91-panel.jpg>>. Acesso em: 19 abr. 2018.

Segundo Tanenbaum (2003), uma das técnicas desenvolvidas nesta geração foi a multiprogramação, cujo objetivo era permitir que vários programas executassem ao mesmo tempo, compartilhando os recursos de memória. Da necessidade de respostas mais rápidas de processamento surgiu o conceito de compartilhamento de tempo ou timesharing, que consiste em dividir o tempo de CPU por intervalo de tempo a cada programa a ser executado. Nesta época, foi desenvolvido um sistema operacional que suportasse vários usuários conectados ao mesmo tempo, conhecido com Multics. O projeto do Multics inseriu conceitos inovadores, porém somente nos anos seguintes foi criado o Unix, que é multitarefa e multiusuário.

A quarta geração de computadores compreende desde de 1980 até a presente data (computadores pessoais). Com o desenvolvimento acelerado dos circuitos integrados ou microchip (circuito eletrônico), foi iniciada a era dos computadores de uso pessoal. Desde então, eles tiveram uma grande evolução em agilidade e praticidade, tornaram-se menores, mais rápidos e mais baratos. Os sistemas operacionais usados nesta geração foram o MS-DOS e o Unix, quando se iniciou a interface gráfica, sendo o MS-DOS a base da evolução para o conhecido Windows. Com o desenvolvimento das redes de computadores pessoais, surgiram os sistemas operacionais de rede, que permitem conectar-se a máquinas remotas e copiar arquivos de uma máquina para outra, e os sistemas operacionais distribuídos, que possuem múltiplos processadores. Hoje, os sistemas operacionais gerenciam redes, aplicações para redes, sistemas operacionais de servidores, sistemas operacionais em tempo real, além dos sistemas operacionais para celulares.



Pesquise mais

Para saber mais sobre outros sistemas operacionais, acesse o artigo Conheça outros sistemas operacionais. Disponível em: <<https://www.tecmundo.com.br/macos/726-conheca-outros-sistemas-operacionais.htm>>. Acesso em: 19 abr. 2018.

Principais funções dos sistemas operacionais

As funções dos sistemas operacionais são: estender a máquina e gerenciar os recursos (TANENBAUM, 2003).

- Estender a máquina (ou máquina virtual):

A função do sistema operacional como uma máquina estendida é esconder a complexidade do hardware do programador, conhecida também como abstração. Por exemplo, o sistema operacional trata cada dispositivo físico como um arquivo e esses arquivos, ao serem manipulados emitindo os comandos de leitura/escrita ou de abrir/fechar, muito complexos devido à quantidade de parâmetros que recebem, ficam a cargo do sistema operacional controlar esse dispositivo diretamente com o hardware.



Assimile

Vimos nesta seção que uma das funções do sistema operacional é estender a máquina. O sistema operacional como uma máquina estendida ou máquina virtual esconde do programador a complexidade do hardware, apresentando uma interface amigável e gerenciável do sistema.

- Gerenciar os recursos:
1. Controlar de forma ordenada e compartilhada os recursos do computador, como a memória, o processador e os dispositivos de E/S para os processos (programas) que estão aguardando por eles.
 2. Controlar quem está usando qual recurso, garantindo suas requisições de recursos e mediando os conflitos entre programas e usuário.
 3. O gerenciamento de recursos controla o compartilhamento de recursos de duas formas: no tempo e no espaço. Quando um recurso é compartilhado no tempo, vários programas aguardam a sua vez de usá-lo. O sistema operacional é quem decide qual programa irá utilizar o recurso e por quanto tempo. Outro tipo de compartilhamento é o de espaço, no qual vários programas utilizam uma parte do recurso. Por exemplo, a memória principal é dividida entre vários programas em execução, permitindo assim que cada programa execute ao mesmo tempo, sendo mais eficiente.



Exemplificando

O sistema operacional controla de forma ordenada e compartilhada os recursos do computador. Por exemplo, se você estiver editando um texto num editor de textos e gravando uma mídia de dados, o sistema operacional não permite que estes programas acessem ao mesmo tempo a memória principal, pois isso causaria a perda dos dados salvos durante a edição do arquivo e ao acessar os dados da mídia, os arquivos teriam sido apagados.

Principais serviços dos sistemas operacionais

O sistema operacional oferece os seguintes serviços para os aplicativos dos usuários e também ao próprio sistema (MACHADO, 1997):

1. Meios para que um programa seja carregado em memória e executado.
2. Um sistema de arquivos, permitindo criar, ler, escrever e excluir arquivos.
3. Uma interface de acesso aos periféricos (impressoras, scanner, câmera, pen-drive, etc.).
4. Mecanismos de monitoração de recursos, capazes de identificar possíveis gargalos no sistema.
5. Meios para armazenar/manter o estado do sistema.



Refita

Diante do que aprendemos: seria possível operar um computador sem sistema operacional?

Sem medo de errar

Agora que você já conheceu a história dos sistemas operacionais e aprendeu sobre as principais funções e os principais serviços oferecidos por eles, vamos voltar ao nosso contexto sobre o curso de sistemas operacionais ministrado por você durante o simpósio “Informática para todos” entre as escolas municipais da região. Na

realização do debate, surgem os seguintes questionamentos: qual a importância dos sistemas operacionais e qual o papel deles como gerenciador de recursos do computador? Qual a principal diferença quanto ao acesso do código fonte do Windows e do Linux? Quais são os serviços oferecidos pelos sistemas operacionais Linux e Windows? O sistema operacional é o software mais importante do computador, pois é ele quem irá gerenciar todos os recursos e todas as ações realizadas pelos usuários no computador. Por exemplo, quando você deseja imprimir um documento, o sistema operacional verifica se o arquivo está no disco do computador, confere se as dimensões são as ideais para impressão, verifica se o drive da impressora está instalado e se comunica com a impressora para que o documento seja impresso de acordo com a configuração definida por você.

Os sistemas operacionais Windows e Linux possuem características diferentes quanto aos serviços e às funções oferecidas, por exemplo, o Windows não permite o acesso ao código fonte, sendo restrito aos desenvolvedores da Microsoft. Já o código fonte do Linux é aberto e todos os usuários podem acessar ou até mesmo modificar o código. Quanto à licença de uso, o Linux está licenciado sob a GPL (GNU Public License) e, por ser livre, qualquer usuário pode fazer os ajustes que quiser e lançar no mercado novas versões, além de não existir restrições para o número de computadores que possuem o Linux instalado. Com o Windows você adquire uma licença que pode ser instalada em apenas um computador. O Linux possui um instalador de aplicativos, permitindo que você baixe os programas que desejar. Para instalar aplicativos no Windows, é necessário saber onde adquirir o programa, baixar ou comprar o CD de instalação. Em relação ao suporte aos usuários, existem na internet grupos e fóruns de suporte do Linux, enquanto o do Windows é realizado através de uma central de atendimento e o usuário precisa pagar por este suporte.

Avançando na prática

Funções dos sistemas operacionais - compartilhamento de recursos

Descrição da situação-problema

Pedro é um dos alunos que participa do curso de sistemas operacionais ministrado por você durante o simpósio “Informática para todos” e é um técnico de suporte em uma pequena empresa de marketing. Os funcionários da empresa dele compartilham a impressora e arquivos de dados que contêm informações para o trabalho de cada setor. A empresa possui os setores de análise de mercado, desenvolvimento do produto, comercial e vendas, com duas ou no máximo três pessoas envolvidas nas atividades em cada departamento. O setor comercial relatou a Pedro que algumas análises de mercado realizadas havia dois meses não estavam aparecendo nas pastas de sua área. Pedro identificou que os arquivos citados pelo comercial foram deletados pelos funcionários de outros departamentos. Desta forma, Pedro questiona a você sobre qual seria a solução a ser adotada para solucionar o compartilhamento destes recursos? Outra dúvida levantada pelos alunos foi que, mesmo que o sistema operacional gerencie de forma eficaz o compartilhamento de recursos das máquinas, que é uma das suas funções, por que alguns usuários conseguem apagar arquivos importantes da empresa?

Resolução da situação-problema

Os funcionários da empresa tinham acesso às pastas e diretórios como administrador e foi por isso que aconteceu de arquivos de outros setores serem deletados. Neste caso, muitos sistemas operacionais permitem o controle de acesso para as pastas e arquivos de dados dos diretórios. Como exemplo de controle de acesso para a empresa de marketing, somente os funcionários da área comercial poderão alterar ou modificar os arquivos e os demais funcionários terão acesso somente para leitura. Pedro poderia configurar as pastas de uso comum somente para leitura, o que garantiria a permanência dos arquivos na pasta. Além disso, ele poderia recorrer à cópia (backup) dos arquivos para voltar os arquivos que foram apagados das pastas.

Faça valer a pena

1. O sistema operacional (SO) é um conjunto de programas cuja função básica é servir de interface entre o hardware de um computador e os seus aplicativos, administrando e gerenciando recursos como processadores, memórias e discos. É algo tão essencial para o computador quanto é o coração para o corpo humano. (IBM 2018 - 2).

Analise as principais funções dos sistemas operacionais:

I – Gerenciar os recursos de hardware de forma protegida.

II – Garantir a utilização segura do computador.

III – Facilitar o acesso aos recursos do sistema.

IV – Esconder a complexidade do hardware dos usuários.

Assinale a alternativa correta que apresenta as funções do sistema operacional:

a) I, II e III estão corretas.

b) I, III e IV estão corretas.

c) II, III e IV estão corretas.

d) I e IV estão corretas.

e) I, II, III e IV estão corretas.

2.

Há não muito tempo, a indústria estava em polvorosa para saber quem ganharia a "guerra" entre os sistemas operacionais e qual deles sobreviveria para tornar-se padrão. Na verdade, não é possível que alguma companhia ou sistema operacional isoladamente prevaleça para suplantar todos os outros concorrentes - como também não é provável que uma empresa automobilística produza um carro tão bom que coloque todos os outros fora do mercado. Igualmente, as pessoas às vezes questionam os méritos de uma interface com o usuário em relação a outra e esperam que a sua preferida torne-se padrão - mas isso simplesmente não acontecerá. (NÓBREGA FILHO, R. G., [s.d.]. Disponível em: <<http://www.di.ufpb.br/raimundo/SistOper/IntSO/IntrodSO.html>>. Acesso em: 20 abr. 2018).



A seguir, analise as asserções:

As pessoas são diferentes e gostam de ambientes computacionais diferentes para diferentes aplicações e tarefas.

PORQUE

A concorrência entre as empresas de sistemas operacionais faz com que cada vez mais os sistemas se tornem ágeis e práticos e, assim, os usuários possuam um maior número de opções que se adequem aos seus objetivos.

Com relação às duas asserções, assinale a alternativa correta:

- a) As asserções I e II são proposições verdadeiras, e a II é uma justificativa da I.
- b) As asserções I e II são proposições verdadeiras, mas a II não é uma justificativa da I.
- c) A asserção I é uma proposição verdadeira, e a II é uma proposição falsa.
- d) A asserção I é uma proposição falsa, e a II é uma proposição verdadeira.
- e) As asserções I e II são proposições falsas.

3. Segundo o site da Globo, em 03/04/2017,



o Android, do Google, passou o Windows e se tornou o sistema operacional mais usado do mundo em março de 2017, informou a StatCounter em relatório divulgado nesta segunda-feira (3). É a primeira vez desde que foi lançado na década de 1980 que o topo não é ocupado pelo software da Microsoft. O levantamento da firma de análise considera computadores, notebooks, tablets e smartphones. O Android estava em 37,93% dos aparelhos, enquanto o Windows equipava 37,91% deles. (G1, 2017. Disponível em: <<https://g1.globo.com/tecnologia/noticia/android-passa-windows-e-se-torna-o-sistema-operacional-mais-usado-do-mundo.ghtml>>. Acesso em: 20 abr. 2018).

Considerando o marco na história da tecnologia e a atual liderança do sistema operacional Android, escolha a opção correta:

- a) Este marco da tecnologia deve-se ao fato do avanço e do aumento do uso de smartphones fabricados no mercado, e o acesso a esses aparelhos é feito através do sistema Android.
- b) Durante anos, o Windows foi o sistema operacional dos computadores e notebooks, fazendo com que os usuários quisessem novas tecnologias e, por isso, com o surgimento dos smartphones, o Windows perdeu a sua liderança.

- c) Mesmo a diferença sendo pequena no percentual entre Android e Windows apresentado pela pesquisa, a Microsoft não tem com o que se preocupar, uma vez que a qualidade dos seus serviços é muito boa e os usuários estão acostumados a usar somente o Windows, facilitando a consolidação da marca nos smartphones.
- d) Este marco da tecnologia deve-se ao fato de que os smartphones desenvolvidos somente nos EUA elevaram os índices de uso do sistema do Google, e, em breve, o sistema iOS da Apple passará o uso do Android.
- e) Os dados apresentados pela pesquisa são pontuais, o que não representa significativamente a perda do Windows da liderança competitiva do mercado.

Seção 1.2

Evolução dos sistemas operacionais

Diálogo aberto

Olá, aluno! Seja bem-vindo a mais uma seção de estudos deste livro didático!

Você já pensou sobre como os sistemas operacionais conseguem gerenciar diversos programas e aplicativos do usuário e do sistema sem comprometer a performance do computador? Quando um computador está ligado, além dos seus aplicativos que estão sendo executados (como editor de textos, internet, aplicativos de música, entre outros), outros programas estão rodando em paralelo, como o antivírus (rastreando as possíveis ameaças que poderiam danificar o sistema operacional), atualização de versões de aplicativos instalados, etc. Esse gerenciamento eficaz somente é possível pelo fato de os sistemas operacionais atuais serem multitarefa.

Nesta seção vamos aprofundar o conhecimento sobre a estrutura interna dos sistemas operacionais. Conheceremos os tipos e aprenderemos sobre os sistemas operacionais monoprogramáveis, multiprogramáveis e com múltiplos processadores.

Vamos relembrar da nossa situação-problema sobre o curso de sistemas operacionais ministrado por você durante o simpósio “Informática para todos” entre as escolas municipais da região. Durante o curso, outro artigo foi disponibilizado aos alunos, que trata da classificação dos sistemas operacionais quanto à sua capacidade de controlar mais de uma tarefa ao mesmo tempo (monotarefa, multitarefa e multiprocessamento) e sua performance durante este controle. Desta forma, você é questionado: quais as principais estruturas dos sistemas operacionais? Como os sistemas operacionais da atualidade, como Windows e Linux, conseguem realizar diversas tarefas ao mesmo tempo sem prejudicar o desempenho do computador? Quais as principais características e aplicações dos sistemas operacionais multiprogramáveis/multitarefa? Qual a principal diferença entre os sistemas multiprogramáveis. Para os sistemas operacionais multiusuários, o Linux e Windows oferecem suporte?

Não pode faltar

O sistema operacional é composto por um conjunto de rotinas chamado kernel ou núcleo do sistema operacional. O kernel é o principal componente dos sistemas operacionais, pois ele é quem faz o gerenciamento dos recursos do computador (MACHADO; MAIA, 2007). Quando o computador é ligado, o kernel executa programas que inicializam o computador e o conjunto de rotinas é executado ao mesmo tempo sem uma ordem pré-definida.

As principais funções do núcleo são (SIQUEIRA, 2018):

- Tratamento de interrupções e exceções.
- Gerenciamento, sincronização, comunicação e escalonamento de processos e *threads*.
- Gerenciamento da memória.
- Gerenciamento dos sistemas de arquivos.
- Gerenciamento dos dispositivos de entrada/saída.
- Auditoria e segurança do sistema.

Os sistemas operacionais limitam as ações executadas pelos programas em função da segurança e da estabilidade, como o acesso à memória do computador. Assim, os modos de acesso aos serviços do núcleo do computador referem-se aos privilégios de execução de um programa, garantindo que a memória não seja invadida por outro programa, por exemplo.

Os modos de acesso são realizados de duas formas: usuário e kernel (núcleo). No modo usuário, os programas podem executar ações sem privilégios, como leitura de um arquivo. No modo kernel, o acesso pode ser realizado em modo usuário ou em modo com privilégio total no computador, como acesso ao disco.

Quando um programa é executado, o kernel é consultado para saber se o acesso será realizado em modo usuário ou kernel. Os acessos aos serviços do núcleo são realizados através de chamadas ao sistema ou *system call*. As chamadas ao sistema fazem a interface entre o sistema operacional e os programas dos usuários, além de poderem solicitar serviços de acesso ao disco rígido ou para a criação e execução de processos, por exemplo.

Durante uma chamada ao sistema, o sistema operacional recebe um comando (por exemplo, abrir um arquivo) e seus parâmetros e, como resposta à chamada, recebe um outro código sinalizando sucesso (se a abertura do arquivo aconteceu com sucesso), falha (se houve erro na abertura do arquivo) ou o resultado do próprio comando (o arquivo é aberto para o usuário).



Refletir

Os usuários de computador conseguem identificar claramente quais são as funções do núcleo, diferenciando-as das funções do software aplicativo (editores de texto, Internet Explorer, entre outros)? Como pode ser feita essa diferenciação?

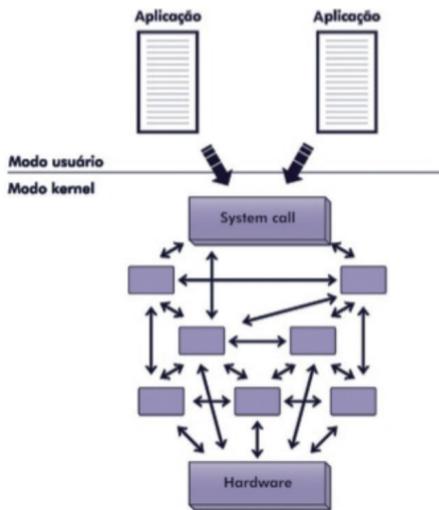
A estrutura do sistema operacional, ou seja, a maneira como o código do sistema é organizado e o inter-relacionamento entre seus diversos componentes podem variar conforme a concepção do projeto. Veremos nesta seção os modelos das principais arquiteturas dos sistemas operacionais, os quais, segundo Tanenbaum (2003), são:

- Sistemas Monolíticos.
- Sistemas em Camadas.
- Máquinas Virtuais.
- Modelo Cliente-Servidor.

Sistemas Monolíticos

No sistema monolítico, o sistema operacional é escrito como um conjunto de módulos compilados separadamente e depois agrupados em um arquivo executável, em que cada procedimento pode ser chamado por outro quando precisar, conforme apresentado na Figura 1.6. A vantagem do sistema monolítico refere-se à simplicidade e à eficiência em função da integração interna dos componentes, permitindo que os detalhes de hardware sejam explorados. Um exemplo de sistema monolítico é o sistema operacional MS-DOS.

Figura 1.6 | Modelo de um sistema monolítico



Fonte: Machado; Maia (2018, p. 54).

Sistemas em camadas

O sistema em camadas organiza o sistema operacional como uma hierarquia de camadas construídas umas sobre as outras. Cada camada oferece um conjunto de funções que podem ser utilizadas pelas camadas superiores (TANENBAUM, 2003). A Figura 1.7 apresenta um modelo de sistema de camadas.

Figura 1.7 | Modelo do sistema em camadas

Camada	Função
5	O operador
4	Programas do usuário
3	Gerenciamento de entrada/saída
2	Comunicação operador-processo
1	Gerenciamento da memória e do tambor magnético
0	Alocação de processador e multiprogramação

Fonte: Tanenbaum (2003, p. 43).

A camada 0 faz a alocação do processador alternando entre os processos e realizando a multiprogramação. A camada 1 gerencia a memória, alocando espaço na memória principal e

em um tambor (meio magnético de armazenamento de dados). A camada 2 gerencia a comunicação entre o processo e o console (componentes de entrada (teclado) e saída (monitor) de dados) do operador. A camada 3 gerencia os dispositivos de entrada/saída. Na camada 4 estão os programas de usuários e na camada 5 está o processo do operador do sistema.

Máquinas Virtuais

Uma máquina virtual (VM) é similar a uma máquina real. Assim, um nível intermediário é criado entre o sistema operacional e o hardware. Esse nível intermediário cria várias máquinas virtuais independentes, cada qual possuindo uma cópia virtual do hardware. Em um mesmo hardware e em plataformas diferentes podem rodar mais de uma máquina virtual (MACHADO; MAIA, 2007). As principais vantagens das máquinas virtuais são o aproveitamento dos recursos das máquinas, a garantia da segurança e confiabilidade, pois se acontecer um problema com uma máquina virtual, as demais não serão impactadas, além de facilitar a recuperação de dados e backups.



Pesquise mais

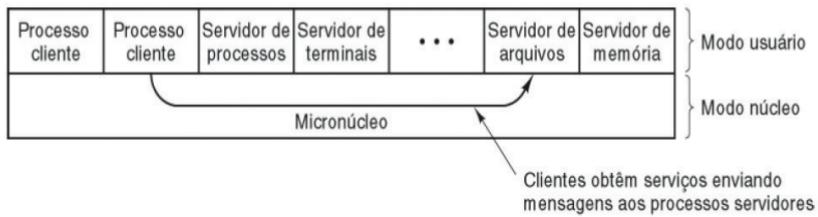
Você sabia que um sistema operacional pode "rodar" dentro de outro?
Para saber mais sobre máquina virtual acesse o artigo:

HAMMERSCHMIDT, R. O que são máquinas virtuais? **tecmundo**, 5 jul. 2012. Disponível em: <<https://www.tecmundo.com.br/maquina-virtual/232-o-que-sao-maquinas-virtuais-.htm>>. Acesso em: 22 jun. 2018.

Modelo cliente-servidor

No modelo cliente-servidor a ideia é implementar a maior parte das funções em modo usuário. Tudo o que o kernel faz é cuidar da comunicação entre cliente e servidor, conforme apresentado na Figura 1.8, que também apresenta a divisão do sistema operacional em várias partes, como servidor de processos, de terminais, de arquivos e de memória, tornando cada parte gerenciável. Todos os servidores executam em modo usuário e, com isso, caso ocorra um erro no servidor de arquivos, não impactará nos demais servidores (TANENBAUM, 2003).

Figura 1.8 - Modelo do sistema em camadas



Fonte: Tanenbaum (2003, p. 46).

Classificação dos sistemas operacionais

Segundo Machado e Maia (2007), os sistemas operacionais podem ser classificados em:

1. Monoprogramáveis/monotarefa.
2. Multiprogramáveis/multitarefa.
3. Sistemas com múltiplos processadores.

1. Sistemas Monoprogramáveis/monotarefa

Segundo Machado e Maia (2007), os sistemas operacionais monoprogramáveis/monotarefa referem-se aos primeiros computadores pessoais criados em 1960. Os sistemas monoprogramáveis/monotarefa executam um único programa por vez e todos os recursos da máquina são alocados exclusivamente para o programa em execução, mesmo que não esteja utilizando o recurso. Por exemplo, você deseja atualizar uma planilha eletrônica e editar um texto, ou você atualiza a planilha eletrônica e fecha o editor de textos ou edita o texto e fecha a planilha eletrônica. A implementação de sistemas monoprogramáveis/monotarefa é simples porque não é necessário se preocupar com a concorrência de recurso. Um exemplo de monoprogramáveis/monotarefa foi o MS-DOS.

2. Sistemas Multiprogramáveis/multitarefa

Os sistemas operacionais multiprogramáveis/multitarefa dividem os recursos do computador com os demais programas e, com isso, a CPU não fica ociosa como no caso de sistemas operacionais monoprogramáveis/monotarefa (MACHADO; MAIA, 2007). Por exemplo, você pode atualizar uma planilha eletrônica e utilizar

um editor texto ao mesmo tempo. Os sistemas multiprogramáveis podem ser classificados pela maneira com que as aplicações são gerenciadas e pelo número de usuários que utilizam o sistema. Os sistemas multiprogramáveis em relação ao número de usuários que os utilizam são:

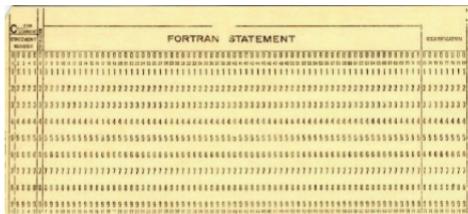
- Monousuários: somente um usuário utilizando os recursos da máquina.
 - Multiusuários: vários usuários logados utilizando os recursos da máquina.

Os sistemas multiprogramáveis, em relação à maneira com que as aplicações são gerenciadas, podem ser classificados como batch, de tempo compartilhado ou de tempo real.

Sistemas batch

Os sistemas batch foram os primeiros sistemas operacionais multiprogramáveis e o processamento era realizado por meio de grupos de registros (lote). Os programas, conhecidos como *jobs*, eram executados através de cartões perfurados, conforme a Figura 1.9, e armazenados em disco ou fitas, onde aguardavam para serem processados. Nos sistemas batch não havia interação do usuário com a aplicação. Backups, cálculos numéricos e compilações são exemplos de processamento batch.

Figura 1.9 | Exemplo de cartão perfurado



Fonte: <https://commons.wikimedia.org/wiki/File:Hollerith_card.jpg>. Acesso em: 23 abr. 2018.

Sistemas de tempo compartilhado (time-sharing)

Os sistemas de tempo compartilhado (*time-sharing*) permitem que vários programas sejam executados a partir da divisão do tempo do processador em pequenos intervalos, conhecidos como fatia de tempo (*time-slice*) (MACHADO; MAIA, 2007). No sistema

de tempo compartilhado, cada usuário possui um ambiente de trabalho próprio, dando a impressão de que o sistema está totalmente dedicado a ele. Se um tempo destinado à execução de um programa não for suficiente, o programa é interrompido pelo sistema operacional, que passa a vez de execução para outro programa, e o que foi interrompido fica aguardando uma nova fatia de tempo para executar. Nos sistemas de tempo compartilhado, os comandos dados pelos usuários aos sistemas operacionais são respondidos em segundos e, por isso, ficaram conhecidos como sistemas on-line. Muitas aplicações comerciais utilizam este sistema por ter um tempo de resposta razoável.

Sistemas de tempo real (*real-time*)

Nos sistemas de tempo real, o tempo é o principal parâmetro. Um programa utiliza os recursos do computador até aparecer outro com maior prioridade. Os prazos são rigorosos quanto à execução de uma tarefa (MACHADO; MAIA, 2007) e aplicados em processos de controles industriais, como monitoramento de refinarias de petróleo, controle de tráfego aéreo, de usinas termoelétricas e nucleares ou em qualquer aplicação onde o tempo de respostas é fator fundamental.

3. Sistemas com múltiplos processadores

Segundo Machado e Maia (2007), os sistemas com múltiplos processadores possuem duas ou mais CPUs interligadas e permitem que vários programas sejam executados ao mesmo tempo. Os sistemas com múltiplos processadores permitem que um programa seja subdividido e executado ao mesmo tempo em mais de um processador. Com isso, além de esses sistemas serem rápidos, possuem uma maior capacidade de processamento e controle da CPU. Os sistemas com múltiplos processadores são complexos, gastam uma parte do tempo gerenciando a CPU e também estão presentes nas aplicações de simulações e prospecção de petróleo. Os sistemas com múltiplos processadores podem ser classificados em:

- Sistemas fortemente acoplados**

Possuem dois ou mais processadores, compartilham a mesma memória e os dispositivos de entrada/saída são controlados por um único sistema operacional. São utilizados em sistemas

que usam intensivamente a CPU, voltando-se à solução de apenas um problema.

Os sistemas fortemente acoplados são classificados em:

- **Simétricos**

Nos sistemas simétricos os processadores compartilham uma única memória e utilizam o mesmo sistema operacional. Usam a técnica de paralelismo em que os programas podem ser divididos em partes, podendo rodar concorrentemente em processadores que estejam disponíveis. Com o processamento paralelo, há um aumento do processamento e caso haja uma falha em um dos processadores, não impacta na performance do sistema. O Windows e o Linux suportam os sistemas simétricos.

- **Assimétricos**

Nos sistemas assimétricos existe um processador principal (mestre) que controla o sistema e delega atividades para os demais processadores. Se acontecer algum problema com o processador principal, os demais não continuam o processamento.

- **Sistemas fracamente acoplados**

Os sistemas fracamente acoplados funcionam de forma independente, possuindo seu sistema operacional e gerenciando seus próprios recursos como memória, CPU e dispositivos de entrada e saída. O sistema é fracamente acoplado por depender de uma rede para distribuir as atividades do processamento (MACHADO; MAIA, 2007).



Assimile

Uma das diferenças entre os sistemas operacionais fortemente acoplados e os fracamente acoplados é que os sistemas fortemente acoplados possuem apenas uma memória compartilhada pelos processadores, já os fracamente acoplados possuem a sua memória individualmente.

Os sistemas fortemente acoplados são classificados em:

- **Operacionais de rede**

Os sistemas operacionais de rede são independentes e estão conectados por meio de uma rede. Na rede, cada estação de trabalho ou nó possui um sistema operacional próprio e tem a capacidade de processamento das suas aplicações (MACHADO; MAIA, 2007). Eles também permitem o compartilhamento de recursos como impressora, diretórios, cópia de arquivos, entre outros.

- **Operacionais distribuídos**

Os sistemas operacionais distribuídos permitem que um programa seja dividido em partes e que cada parte seja executada em nós diferentes da rede. Os usuários do sistema veem como um sistema centralizado e não como um sistema em rede (MACHADO; MAIA, 2007).



Exemplificando

Um exemplo de sistemas distribuídos são os clusters. Um cluster é um conjunto de computadores conectados como se fossem um grande sistema. Esse conjunto consegue processar aplicações como se fossem supercomputadores. Cada computador que faz parte do cluster recebe um nome de um nó, e o usuário não conhece os nomes dos membros do cluster e nem quantos são. Clusters são muito utilizados em servidores de bancos de dados e Web. ALECRIM, E. **Cluster: conceito e características**. Disponível em: <<https://www.infowester.com/cluster.php>>. Acesso em: 23 abr. 2018. Por exemplo, o Google contém vários datacenters (compostos por várias máquinas em cluster) espalhados pelo mundo e quando uma solicitação de pesquisa é feita na web, a informação é procurada no datacenter mais próximo, para que tenha um tempo de resposta mais rápido.

Sem medo de errar

Agora que você já conheceu a estrutura dos sistemas operacionais e estudou sobre seus tipos, monoprogramáveis e multiprogramáveis e com múltiplos processadores, vamos voltar ao contexto sobre o curso de sistemas operacionais ministrado por você durante o

simpósio “Informática para todos” entre as escolas municipais da região. Um dos artigos disponibilizados por você aos alunos refere-se à capacidade dos sistemas operacionais de controlar mais de uma tarefa ao mesmo tempo e sua performance durante este controle. Os questionamentos dos alunos direcionados a você foram: quais as principais estruturas dos sistemas operacionais? Como os sistemas operacionais da atualidade, como o Windows e Linux, conseguem realizar diversas tarefas ao mesmo tempo sem prejudicar o desempenho do computador? Quais as principais características e aplicações dos sistemas operacionais multiprogramáveis/multitarefa? Qual a principal diferença entre os sistemas multiprogramáveis? Para os sistemas operacionais multiusuários, o Linux e Windows oferecem suporte?

Os sistemas operacionais possuem a estrutura de acordo com o projeto e as principais arquiteturas são: os sistemas monolíticos, caracterizados pela simplicidade de implementação, os sistemas em camadas, em que o sistema operacional é organizado em hierarquia de camadas, as máquinas virtuais, que simulam as máquinas reais e são instaladas dentro de outro sistema operacional, e o modelo cliente-servidor, que executa as funções em modo usuário. Os sistemas operacionais da atualidade, como Windows e Linux, permitem a execução de vários programas ao mesmo tempo e gerenciam, de forma eficaz, todos os recursos da máquina por serem multitarefa e multiusuário. Os sistemas operacionais multiprogramáveis/multitarefa gerenciam o tempo de processamento de um programa de acordo com as prioridades e o tempo que este programa já executou. Assim, todos os programas possuem uma fatia de tempo para uso dos recursos da máquina (processador, memória e dispositivos de entrada/saída). Uma das principais características dos sistemas multiprogramáveis/multitarefa é a redução do tempo de resposta dos programas, além da redução de custos no compartilhamento de recursos do computador entre os demais programas. Os sistemas multiprogramáveis podem ser do tipo batch, tempo compartilhado e sistemas em tempo real. Os sistemas batch executam por meio de lotes de tarefas e o sistema operacional transfere a execução para a próxima tarefa após a finalização completa da que está em execução. Um exemplo de uso do sistema batch é o processamento de uma conta telefônica. Os sistemas de tempo compartilhado dividem o processamento

com vários usuários do sistema e muitas aplicações comerciais utilizam este sistema. Já nos sistemas em tempo real, o tempo é o fator principal, pois o tempo de resposta precisa ser rápido, como, por exemplo, a esteira de produção de um carro. O suporte aos usuários Linux é realizado por meio de grupos e fóruns na internet ou assistência especializada (serviço pago). O suporte do Windows é realizado através de uma central de atendimento e o usuário precisa pagar por este suporte.

Avançando na prática

Estrutura dos sistemas operacionais - máquina virtual

Descrição da situação-problema

Cláudio, um dos alunos que participam do curso de sistemas operacionais, sugeriu ao seu tio, dono de uma empresa prestadora de serviços de impressão, a instalação de máquinas virtuais. Nessa empresa, os clientes solicitam a impressão de arquivos de imagem, documentos de texto e apostilas utilizando um pen-drive ou pelo celular. Isso gera a infecção das máquinas por vírus, mesmo usando um antivírus eficiente, sendo constantemente necessário formatar as máquinas que fazem a impressão. Nas máquinas da empresa estão instalados o sistema operacional Windows e através da máquina virtual foi instalado o sistema operacional Linux. O tio de Cláudio utilizou o MVM (*Microsoft Virtual Machine*) para realizar a instalação. Ao final, o proprietário identificou que o mouse e o teclado não estavam funcionando. Desta forma, Cláudio questiona a você sobre qual seria a solução a ser adotada para solucionar o problema. Além disso, outro aluno pergunta se com a instalação da máquina virtual não haverá mais vírus.

Resolução da situação-problema

Para um bom funcionamento dos dispositivos de entrada e saída conectados ao computador, é necessário que os drivers (softwares que permitem a comunicação do sistema operacional e do computador) estejam instalados corretamente. As máquinas virtuais suportam todos os drivers de entrada e saída. Em alguns casos, reiniciando as máquinas virtuais resolve-se o problema do

não funcionamento dos drivers. O problema relatado por Cláudio acontece quando não existem os drives para fazer interface entre a máquina virtual e a máquina real. Neste caso, é necessário instalar manualmente os drives ou reinstalar a máquina virtual. A máquina virtual não é imune a vírus, mas não compromete o hardware. Além disso, não é necessário a formatação constante da máquina, sendo necessário apenas a desinstalação da máquina virtual.

Faça valer a pena

1. Com relação à classificação dos sistemas operacionais multiprogramáveis (vários programas utilizando o mesmo recurso) e à maneira com que as aplicações são gerenciadas, analise as questões a seguir e associe as colunas de acordo com a característica de cada sistema:

- | | |
|---|---|
| 1. Sistema batch. | (<input type="checkbox"/>) Vários programas são executados a partir da divisão de tempo do processador. |
| 2. Sistema de tempo real. | (<input type="checkbox"/>) Não interagem com o usuário com a aplicação. |
| 3. Sistemas de tempo compartilhado (<i>time-sharing</i>). | (<input type="checkbox"/>) O tempo é o principal parâmetro de funcionamento.
(<input type="checkbox"/>) O usuário tem a impressão de que o sistema está totalmente disponível para ele.
(<input type="checkbox"/>) Os prazos são rígidos na execução das tarefas. |

Assinale a alternativa que possui a ordem correta da associação das duas colunas:

- a) 3, 1, 2, 3, 2.
- b) 2, 1, 3, 2, 3.
- c) 1, 3, 2, 3, 2.
- d) 2, 3, 1, 2, 3.
- e) 3, 1, 2, 2, 3.

2.

O controle remoto das TVs atuais possui microfone e aceita comandos por voz; comunica-se com a tela por Bluetooth, servindo como mouse ou cursor, com sensor de movimentos ou *touchpad*. As TVs Smart possuem processadores de alto desempenho e sistema operacional como os computadores, com plataforma multitarefas que podem processar mais de uma tela ao mesmo tempo, de forma mais intuitiva e integrada.



Fonte: SANTOS, A. TVs: comparando os sistemas operacionais. Home Theater e Casa Digital, 2015. Disponível em: <<http://revistahometheater.uol.com.br/portal/2015/11/04/tvs-comparando-os-sistemas-operacionais/>>. Acesso em: 23 abr. 2018.

Analisando o texto acima e conhecendo as características dos sistemas operacionais multitarefa, marque a questão correta:

- a) No caso do texto apresentado, as televisões atuais conseguem realizar várias tarefas ao mesmo tempo, pois o sistema operacional instalado nelas gerencia de forma eficaz as atividades solicitadas pelos usuários, como acessar a internet e ouvir música.
- b) Nos sistemas operacionais multitarefa, os recursos do computador não são tão eficientes, uma vez que é necessário uma maior atuação do sistema operacional, consumindo mais memória e recursos de CPU.
- c) Os sistemas operacionais multitarefa têm a capacidade de rodar mais de um programa ao mesmo tempo, porém executam apenas três programas por vez.
- d) O sistema operacional Unix introduziu os conceitos de multitarefa, porém não se tornou conhecido e não serviu de base para a grande maioria dos sistemas operacionais atuais.
- e) Nos sistemas operacionais multitarefa, os usuários têm a impressão de que os programas rodam ao mesmo tempo e demoram muito para executar, pois permitem que vários usuários estejam logados.

3. Uma máquina virtual é um software de ambiente computacional em que um sistema operacional ou programa pode ser instalado e executado. De maneira mais simplificada, podemos dizer que a máquina virtual funciona como um “computador dentro do computador”.

Fonte: HAMMERSCHMIDT, R. O que são máquinas virtuais? Disponível em: <<https://www.tecmundo.com.br/mquina-virtual/232-o-que-sao-maquinas-virtuais-.htm>>. Acesso em: 23 abr. 2018.

Considerando os conceitos de máquina virtual, escolha a opção correta:

- a) Uma das desvantagens de usar máquinas virtuais é poder testar vários sistemas operacionais sem precisar particionar o HD.
- b) Nem todos os usuários de computadores precisam de uma máquina virtual, ela é indicada quando é necessário rodar outro sistema operacional no computador.
- c) Utilizando uma máquina virtual não há economia de espaço em disco, pois são executados mais de um sistema operacional.
- d) A recuperação de dados em uma máquina virtual é muito lenta, pois o controle é transferido para o sistema operacional instalado na máquina e somente ele tem acesso aos dados para dar um retorno, o que demanda um tempo maior.
- e) As máquinas virtuais não podem ser instaladas em qualquer plataforma de sistema operacional, pois os recursos são restritos dependendo da arquitetura do sistema operacional.

Seção 1.3

Características e composição dos sistemas operacionais

Diálogo aberto

Olá, aluno. Seja bem-vindo a mais uma seção de estudos deste livro didático.

Uma das maiores discussões na área de tecnologia está relacionada a: qual o melhor sistema operacional, Linux, Windows, Mac OS, Android, iOS, Solaris ou outros sistemas operacionais existentes no mercado? Por exemplo, tanto o Windows quanto o Linux possuem características próprias e, em determinados momentos, é aconselhável usar o Linux ao invés do Windows e vice-versa. Mas uma das questões indiscutíveis é que o Linux é gratuito e possui o código aberto, enquanto o Windows é pago e gerenciável pela Microsoft.

Nesta seção vamos aprender as características e a composição dos sistemas operacionais. Serão apresentados exemplos de sistemas operacionais como Mobile e na Nuvem, além das características e propriedades do Unix, do Linux e do Windows. Vamos relembrar nossa situação hipotética sobre o curso de sistemas operacionais ministrado por você durante o simpósio “Informática para todos” entre as escolas municipais da região. Uma das discussões mais aguardadas durante o simpósio era sobre Windows e Linux. Durante uma discussão, você percebe que os alunos não chegam a um consenso sobre qual sistema operacional usar e que há grupos apoiando o Windows e outros apoiando o Linux. Porém, há um consenso entre os alunos sobre uma vantagem do Linux: o software é livre e pode receber contribuições de melhoria das pessoas em todo o mundo. Diante disso, o professor que lhe convidou pede que você responda para os alunos: quais são as principais características do Linux e Windows? Quais são as principais diferenças entre eles quanto a preço, popularidade e suporte? Quais são as configurações de hardware necessárias para a instalação do Linux e do Windows? Para que você consiga responder esses e outros questionamentos

sobre os sistemas operacionais Windows e Linux, vamos conhecer mais sobre eles e, então, você deverá elaborar um relatório apresentando as principais características e aplicações destes sistemas operacionais.

Bons estudos!

Não pode faltar

Com a evolução dos computadores inserindo em sua arquitetura mais eficiência e praticidade, os sistemas operacionais estão vindo mais poderosos e com funções cada vez mais avançadas. Existem vários tipos de sistemas operacionais para gostos e necessidades diferentes, descritos a seguir.

- Sistemas Operacionais Embarcados (TANENBAUM, 2003)

São usados para computadores de mão, podendo ser utilizados em celulares, aparelhos de TV e forno micro-ondas. Estes sistemas possuem características dos sistemas operacionais de tempo real, mas possuem limitações de memória e consumo de energia. São exemplos de sistemas embarcados tvOS (Apple), WebOS (LG) e Tizen (Samsung).

- Sistemas Operacionais Mobile

Os sistemas operacionais mobile são encontrados em celulares, tablets e MP3 players. São mais simples e permitem a comunicação de dados sem fio por bluetooth e wi-fi. Ainda permitem a utilização de rádio, câmera, gravador de voz, entre outros. São exemplos de sistemas operacionais mobile: Android, Windows Phone, iOS, entre outros (GCFAPRENDELIVRE, 2018).



Exemplificando

Um dos sistemas operacionais mobile mais utilizado e vendido é o Android, que é um sistema operacional do Google e é baseado no Linux, podendo ser facilmente adaptado, atendendo às necessidades do usuário.

- Sistemas Operacionais na Nuvem

Os sistemas operacionais na nuvem utilizam os conceitos (todos os serviços oferecidos como banco de dados, redes, etc. são feitos

pela internet) com base na computação na nuvem. Todos os dados do usuário e aplicativos ficam na nuvem (armazenamento de dados através da web) e o acesso é via internet. A Google lançou em 2009 o sistema operacional Chrome OS, que utiliza recursos armazenados on-line (TERRA, 2018).

- Sistemas Operacionais de Cartões Inteligentes (*smart cards*) (TANENBAUM, 2003)

Os cartões inteligentes são os menores sistemas operacionais, são dispositivos do tamanho de cartões de crédito e contêm um chip de CPU. Estes sistemas operacionais têm restrições severas de memória e de energia e são limitados a pagamentos eletrônicos e a saques, por exemplo.



Pesquise mais

O YouOs é um sistema operacional online que roda pela internet, cuja característica é ser leve e personalizável. Para saber mais sobre ele, acesse: Ferreira, J. L. **YouOS- Seu sistema operacional na internet**. Viva o Linux, 2007. Disponível em: <<https://www.vivaolinux.com.br/artigo/YouOS-Seu-sistema-operacional-na-internet/>>. Acesso em: 22 maio 2018.

Os sistemas operacionais Unix, Linux e Windows serão descritos a seguir.

Unix

Segundo Tanenbaum (2003), a história do Unix começou como um projeto de um pesquisador e se tornou uma indústria milionária, envolvendo diversas instituições como universidades, governos e outras corporações.

Em 1960, o foco era desenvolver um sistema operacional de tempo compartilhado, então foi projetado o sistema operacional MULTICS (*Multiplexed Information and Computing Service*). Posteriormente, um dos projetistas abandonou o projeto e os desenvolvedores que permaneceram criaram sua própria versão do sistema operacional, chamada UNICS e, logo em seguida, chamada de Unix, (MACHADO; MAIA, 2007). O Unix foi desenvolvido em *assembly*, mas, para ser adaptado a outras plataformas, ele foi reescrito em linguagem C.

De acordo com Machado e Maia (2007), o Unix é um sistema multiprogramável e multiusuário, que suporta múltiplos processadores e implementa memória virtual. A seguir serão apresentadas algumas características do Unix:

- O Unix foi escrito em uma linguagem de alto nível, o que facilita a compreensão e a portabilidade para outras plataformas.
- Flexibilidade, sendo usado em diversas aplicações.
- Suporte a protocolos de rede.
- Possui um sistema de arquivos com uma estrutura simples.
- Possui uma interface simples e uniforme com os dispositivos de entrada/saída.

O Unix se baseia em uma estrutura monolítica, ou seja, as funções são executadas em modo núcleo, e é composto por:

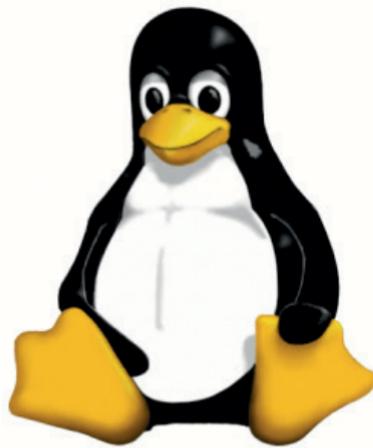
- 1) O kernel: núcleo do sistema operacional. O kernel pode ser dividido em duas partes (MACHADO; MAIA, 2007):
 - **Dependente do hardware**: é composta por rotinas de tratamento de interrupções e exceções e deve ser reescrita quando se está instalando o sistema Unix em outra plataforma.
 - **Independente do hardware**: não existe vínculo com outra plataforma onde está sendo executada, sendo responsável pelo tratamento de system call, gerência de processos, gerência de memórias, entre outras.
- 2) O shell: é o responsável pela conexão dos usuários com os sistemas operacionais através da linha de comando. Sua função é ler e interpretar os comandos e criar processos à medida que são requisitados. Os interpretadores de shell mais populares são Bourne Shell (sh), C Shell (csh) e Korn Shell (ksh).
- 3) Sistemas de arquivos: responsáveis pela organização dos dados armazenados no Unix através de arquivos e diretórios.
- 4) Aplicações: são as aplicações do usuário, como editores de texto, browser de navegação web e compiladores.

Linux

Linus Torvalds, um estudante finlandês, desenvolveu um sistema operacional com base nas características do Minix (um sistema

desenvolvido por Andrew Tanenbaum com objetivos educacionais chamado Linux, em 1991. O termo Linux refere-se ao kernel do sistema operacional (MACHADO; MAIA, 2007). A Figura 1.10 apresenta a mascote do Linux, o Pinguim Tux, escolhido por Linus Torvalds por seu gosto por este animal.

Figura 1.10 | O Pinguim Tux, mascote do Linux



Fonte: <<https://commons.wikimedia.org/wiki/File:Tux2.png>>. Acesso em: 23 maio 2018.

Os programas que interagem com o kernel foram desenvolvidos pela fundação GNU. O Linux em si é somente o kernel e, para que ele funcione, são necessárias ferramentas como o compilador do código-fonte. Por isso, o correto é chamar GNU/Linux (VIVAOLINUX, 2018).

O Linux cresceu rapidamente com a ajuda de vários colaboradores espalhados pelo mundo, contribuindo no desenvolvimento do kernel, utilitários e aplicativos. Hoje, o sistema é usado tanto para fins acadêmicos, quanto para fins comerciais, podendo ser obtido sem custos, juntamente com seu código-fonte.

Por ser um sistema operacional de código-fonte aberto, é flexível e adaptável às necessidades do usuário, além da sua compatibilidade com outros hardwares, possuindo uma alta performance e estabilidade. O fato de o código-fonte do Linux ser aberto permite que as pessoas vejam seu funcionamento e que sejam realizadas alterações e sugestões de melhoria (SILVA, 2018).

A estrutura do Linux é baseada no modelo monolítico, possuindo as mesmas características e composição do Unix: kernel, shell, sistemas de arquivos e aplicações, citados acima.

O Linux possui várias versões conhecidas como distribuições (Kernel mais programas). Cada distribuição tem a sua particularidade como interface gráfica, instalação de programas e suporte gráfico. A seguir serão apresentadas algumas das distribuições do Linux (VIVAOLINUX, 2018):

- DEBIAN: é uma das distribuições mais antigas, estáveis e populares, servindo de base para a criação das distribuições do Ubuntu e Kurumin.
- UBUNTU: é uma das distribuições mais populares por ser amigável e fácil de usar, instalar e obter ajuda na resolução de problemas.
- MINT: é uma das distribuições preferidas dos iniciantes em Linux, por ser fácil de usar.
- FEDORA: é uma distribuição para quem deseja ter ferramentas de configurações gráficas, um desktop robusto e um servidor estável.
- KURUMIN: esta distribuição caracteriza-se pela facilidade de uso, suporte a programas do Windows, interface familiar, entre outras.

Existem outras distribuições que podem ser encontradas nas comunidades Linux.

É comum que os usuários que estão iniciando no Linux fiquem na dúvida sobre qual distribuição escolher. Porém, esta escolha dependerá dos gostos e das necessidades de cada um.

Pontos importantes em relação ao Linux:

- O Linux é licenciado pela GNU Public License (GPL) e o usuário pode baixar e usar em quantas máquinas quiser.
- O código-fonte do Linux é aberto e todos os usuários têm acesso e podem modificá-lo.
- Um dos pontos fortes do Linux é a linha de comando que permite uma administração efetiva do sistema operacional.
- O Linux permite que o usuário adeque o sistema operacional do jeito que ele desejar em relação ao ambiente gráfico.

- Em relação a preço, o Linux é gratuito e possui suporte em diversos fóruns e sites de ajuda, além de o usuário ter a opção de contratar o suporte de grandes empresas.
- A cada dia cresce a popularidade do Linux e cada vez mais as pessoas conhecem e aprendem sobre as vantagens de seu uso.
- Para instalar e executar o Linux, é necessária uma configuração mínima de hardware para que o sistema operacional rode "confortavelmente", além de ser fundamental ter uma máquina com no mínimo 2 Gb de memória RAM, um processador Dual Core e um HD com 250 Gb.



Vimos nesta seção que existem diversas distribuições do GNU/Linux. Na verdade, não existe uma distribuição melhor do que a outra, mas sim uma que atenda às diferentes necessidades de cada usuário.

Windows

Segundo Machado e Maia (2007), o primeiro sistema operacional lançado pela Microsoft, em 1981, foi o MS-DOS. O MS-DOS foi criado com a interface em linha de comando e tinha as características de ser monoprogramável e monousuário.

A Microsoft decidiu dar ao MS-DOS uma interface gráfica com o usuário chamada Windows. As versões de 3.0 até 3.11 do Windows não eram sistemas operacionais, e sim interfaces gráficas com o usuário executando sobre o MS-DOS, (TANENBAUM, 2003). O Windows 95 foi lançado e quase todas as características da parte MS-DOS foram transferidas para a parte Windows, porém o MS-DOS ainda não havia sido totalmente eliminado. Logo após, foi lançada a versão do Windows 98 que tinha poucas diferenças em relação ao 95. A palavra Windows traduzida significa janela, e o sistema foi desenvolvido para competir com a interface gráfica da Macintosh.

Em função das limitações e deficiências do MS-DOS, a Microsoft idealizou o Windows NT (New Technology). Este projeto foi realizado pela DEC (Digital Equipment Corporation), responsável pelo desenvolvimento de vários sistemas operacionais como o PDP/RSX e o VAX/VMS.

Em 1993, a Microsoft lançou o Windows NT nas versões para desktops e servidores, cujo objetivo era desenvolver um sistema operacional multitarefa para executar em ambientes monousuário e multiusuário (TANENBAUM, 2003).

O Windows 2000 é uma evolução do Windows NT e a principal diferença entre eles está na oferta de serviços orientados a ambientes distribuídos e de rede. O Windows 2000 foi escrito em linguagem C.

Os principais objetivos do desenvolvimento do Windows 2000 foram:

- robustez e confiabilidade: proteção do mau funcionamento do sistema;
- extensibilidade e facilidade de manutenção do sistema: adaptação às novas necessidades de hardware e software.
- portabilidade e desempenho: permitir que o sistema fosse executado em diferentes plataformas de hardware e melhorar sua performance.

A arquitetura do Windows 2000 foi baseada em micronúcleo, ou seja, cada funcionalidade é gerenciada por um único componente do sistema operacional. Cada módulo pode ser removido, atualizado ou substituído sem precisar de alterar todo o sistema.

A estrutura do Windows 2000 pode ser dividida em duas partes (TANENBAUM, 2003):

- **modo núcleo**: gerencia a memória, processos, sistemas de arquivos, entre outros.
- **modo usuário**: onde ficam os subsistemas do ambiente e interage através de mensagem.

Depois do Windows 2000, vieram as seguintes versões do Windows, (HARADA, 2018):

- **Windows XP**: tornou-se uma das melhores versões de sistema operacional da Microsoft, devido à mudança no visual e à estabilidade do sistema.
- **Windows Vista**: esta versão teve muitas críticas dos usuários e apresentou muitos problemas, como a transparência e os recursos visuais chamativos, deixando o hardware mais lento.
- **Windows 7**: tornou-se o sistema operacional mais utilizado no mercado, sendo rápido, estável e fácil de usar.

- **Windows 8**: esta versão foi um fracasso em função da mudança no visual e da chegada dos dispositivos sensíveis ao toque.
- **Windows 10**: é a versão mais recente e tem algumas vantagens como leveza, uma adaptação de tela sensível ao toque, uma plataforma unificada aproximando os aplicativos para as plataformas existentes, entre outras.

Pontos importantes em relação ao Windows:

- Para usar o Windows, é necessário adquirir a licença da Microsoft, que é paga por computador que vai utilizá-la.
- O código-fonte do Windows é restrito apenas aos desenvolvedores do sistema operacional.
- O Windows também possui linha de comando, mas não é tão efetiva quanto a do Linux, porque quase todas as configurações são realizadas pela interface gráfica.
- O ambiente gráfico é padronizado pela Microsoft e o usuário não possui autonomia de alteração.
- Quanto ao suporte, o usuário do Windows pode contratar o pago da Microsoft ou utilizar os fóruns de ajuda pela internet.
- O que tornou o Windows um padrão de uso foi o acordo feito entre a Microsoft e os fabricantes de computadores. Eles já saíam de fábrica juntamente com o sistema operacional da Microsoft e, devido à popularidade dos computadores, a Microsoft conseguiu atingir um grande número de usuários.
- Para instalar e executar o Windows, é necessária uma configuração mínima de hardware para que o sistema operacional rode "confortavelmente", sendo necessário ter uma máquina com no mínimo 4 Gb de memória RAM, um processador Core 2 Duo e um HD com 250 Gb.



Reflita

A Microsoft conseguirá lançar uma versão de Windows que agrade a maioria dos seus usuários, independentemente do tipo de uso do sistema operacional?

Componentes do sistema operacional

Segundo Tanenbaum (2003), o sistema operacional tem quatro componentes:

- Gerenciamento de dispositivos de entrada e saída: é responsável por gerenciar os comandos e interrupções geradas pelos dispositivos, por tratar erros e por fornecer uma interface simples e fácil de usar entre os dispositivos e o restante do sistema. Os dispositivos de entrada de dados são os teclados, mouses, microfones, câmeras, entre outros, e os dispositivos de saída de dados são a impressora, o monitor, alto-falantes, pen drives, entre outros.
- Gerenciamento de processos: é responsável por criar, finalizar, escalonar e sincronizar processos e threads. Um processo é um programa em execução. Veremos com mais detalhes o gerenciamento de processos e threads na Unidade 2.
- Gerenciamento de arquivos: é responsável por gerenciar os arquivos (criar e excluir, por exemplo) e os diretórios (criar e excluir, por exemplo). Veremos com mais detalhes o gerenciamento de arquivos na Unidade 3.
- Gerenciamento de memória: gerencia quais partes da memória estão em uso e quais estão disponíveis, além de administrar a troca de processos entre memória principal e memória secundária, o que veremos com mais detalhes na Unidade 4.

Sem medo de errar

Agora que você já aprendeu sobre as características e a composição dos sistemas operacionais e um pouco mais sobre o Unix, Linux e Windows, vamos relembrar o nosso contexto sobre o curso de sistemas operacionais ministrado por você durante o simpósio “Informática para todos” entre as escolas municipais da região.

Osalunosduranteodebatenãoconseguemchegaraumconsenso sobre qual sistema operacional utilizar. Alguns apoiam o Linux e outros o Windows. Assim, surgem os seguintes questionamentos direcionados a você: quais as principais características do Linux e do Windows? Quais as principais diferenças entre eles em relação

a preço, popularidade e suporte? Quais configurações de hardware são necessárias para a instalação do Linux e do Windows?

O Windows e o Linux possuem características distintas em relação: (GUIAPC, 2018)

- À licença: para usar o Windows, é necessário adquirir a licença da Microsoft, paga por computador que vai utilizá-la. Já o Linux é licenciado pela GNU Public License (GPL) e o usuário pode baixar e usar em quantas máquinas quiser.
- Ao acesso ao código-fonte: o código-fonte do Windows é restrito apenas a seus desenvolvedores. Já o código-fonte do Linux é aberto e todos os usuários têm acesso e podem modificá-lo.
- À linha de comando: um dos pontos fortes do Linux é a linha de comando que permite uma administração efetiva do sistema operacional. O Windows também possui linha de comando, mas não é tão efetiva quanto a do Linux, porque quase todas as configurações são realizadas pela interface gráfica.
- À flexibilidade e à rigidez: o Linux permite que o usuário adeque o sistema operacional do jeito que ele desejar em relação ao ambiente gráfico. No Windows, as regras são definidas pela Microsoft.

Além destas características, existem outras que fazem distinção entre Windows e Linux.

Em relação a preço, o Linux é gratuito, já o Windows você paga por licença adquirida. Por exemplo, se precisar ter o Windows instalado em 5 máquinas, é necessário adquirir 5 licenças. Quanto ao suporte, o usuário do Windows pode contratar o suporte pago da Microsoft ou utilizar os fóruns de ajuda pela internet. O Linux possui suporte em diversos fóruns e sites de ajuda, além de o usuário ter a opção de contratar o suporte de grandes empresas.

Quanto à popularidade, o que tornou o Windows um padrão de uso foi o acordo feito entre a Microsoft e os fabricantes de computadores. Eles já saíam da fábrica juntamente com o sistema operacional da Microsoft e, devido à popularidade dos computadores, a Microsoft conseguiu atingir um grande número de usuários. Por outro lado, a popularidade do Linux cresce a cada dia e cada vez mais as pessoas conhecem e aprendem sobre as vantagens de seu uso.

Para instalar e executar tanto o Windows quanto o Linux, é necessária uma configuração mínima de hardware para que o sistema operacional rode “confortavelmente”. Para instalar e executar o Linux, é essencial ter uma máquina com no mínimo 2 Gb de memória RAM, um processador Dual Core e um HD com 250 Gb. Já para instalar e rodar o Windows, é preciso ter máquina com no mínimo 4 Gb de memória RAM, um processador Core 2 Duo e um HD com 250 Gb.

Compile as informações obtidas por meio dos questionamentos e elabore um relatório explicativo, apresentando as principais características e aplicações destes sistemas operacionais, que deverá ser entregue para o professor.

Avançando na prática

Software Livre

Descrição da situação-problema

Daniel, um dos alunos do curso de sistemas operacionais ministrado por você durante o simpósio “Informática para todos” participa como desenvolvedor contribuinte de uma comunidade de software livre. Nela participam desenvolvedores de softwares, empresários do ramo de tecnologia, estudantes e pessoas que queiram contribuir com novas ideias. Anualmente, eles se reúnem no Fórum Internacional de Software Livre para discutir inovações e novas técnicas do mundo do software livre. Durante um dos fóruns on-line da comunidade de software livre, um dos assuntos apresentados foi o motivo de ainda existirem aplicativos que ainda não estão disponíveis para o software livre. Assim, um dos alunos te questiona por quê ainda existem aplicativos não disponíveis para o software livre. Outro aluno o questiona sobre qual a diferença de software livre e software de código aberto (*open source*).

Resolução da situação-problema

O Linux possui diversos aplicativos compatíveis com ele, porém ainda há outros que não possuem versão para o sistema. Por exemplo, quando falamos em editor de textos, o LibreOffice do

Linux substitui o Microsoft Word. Agora, quando se trata de jogos, o Linux não possui uma interface gráfica que atenda aos padrões de qualidade. Isso porque a placa gráfica é dependente das APIs (Interface de Programação de Aplicações) do Windows, e o Linux ainda não conseguiu chegar a uma qualidade que atenda aos seus usuários. Porém, a cada dia essa incompatibilidade se reduz e aplicativos que antes não rodavam em software livre agora estão disponíveis. Em alguns casos, é possível emular o Windows no Linux, ou seja, traduzir as bibliotecas do Windows quando um programa é executado para poder rodar os aplicativos no software livre. A diferença entre o software livre e software de código aberto é que mesmo que o software possua o código aberto, ele não é livre. O software livre pode ser usado, modificado e redistribuído, desde que seja livre para outras pessoas poderem modificar. O usuário pode modificar o código do software de código aberto, mas quem o desenvolveu poderá determinar como esse software será usado e redistribuído. Um dos exemplos é o navegador Firefox.

Faça valer a pena

1.

"É mais fácil encontrar aplicativos, é mais fácil encontrar drivers nos dispositivos que você vai usar nele, além de ser mais fácil de achar pessoas que saibam usar, por ser o mais utilizado no mundo. Então, não há dificuldade para treinar pessoas em uma empresa para usá-lo", conta Wellington Watanabe, CEO da Bug Busters, empresa de assistência técnica para computadores. "Por ser o mais utilizado, é mais fácil de ser atacado por vírus, então, ele acaba sendo muito mais assediado por este tipo de problema de vírus e malware, porque pode-se atingir uma quantidade muito maior de usuários", conclui.



OLHAR DIGITAL. Windows, Mac ou Linux? Veja as vantagens e as falhas de cada um deles. 2011. Disponível em: <https://olhardigital.com.br/video/analisamos_os_principais_sistemas_operacionais_do_mercado/18880>. Acesso em: 24 maio 2018.

Com relação ao sistema operacional Windows, analise as asserções a seguir:
O Windows é o sistema operacional mais atacado por vírus e malwares.

PORQUE

Além da quantidade de usuários que utilizam o sistema operacional, a segurança do Windows é falha, uma vez que ele não foi desenvolvido para ser um sistema com uma excelente segurança. No início, o Windows não tinha controle de acesso a usuários e restrições de segurança e, quando a Microsoft se preocupou com a última questão, o sistema operacional já havia se difundido entre os usuários que muitas vezes utilizam o Windows de forma pirateada.

A respeito dessas asserções, assinale a opção CORRETA.

- a) As asserções I e II são proposições verdadeiras, e a II é uma justificativa da I.
- b) As asserções I e II são proposições verdadeiras, mas a II não é uma justificativa da I.
- c) A asserção I é uma proposição verdadeira, e a II é uma proposição falsa.
- d) A asserção I é uma proposição falsa, e a II é uma proposição verdadeira.
- e) As asserções I e II são proposições falsas.

2.



Até mesmo a Microsoft, inimiga de longa data do software livre, já se rendeu aos benefícios do Linux, e passou a permiti-lo em seu serviço de computação em nuvem, o Azure. De fato, pelo menos 33% dos clientes da Microsoft utilizam o Linux em vez do Windows no Azure. A própria companhia já chegou a abraçar o Linux para sustentar algumas partes do próprio Azure, e fechou uma parceria com a Canonical (que desenvolve o Ubuntu) para fazer com que fosse mais fácil para os programadores construírem aplicações Linux no Windows, com a implementação do shell Bash.

SANTINO, R. Linux completa 25 anos de existência maior e mais profissional do que nunca. Olhar Digital, 2016. Disponível em: <<https://olhardigital.com.br/noticia/linux-completa-25-anos-de-existencia-maior-e-mais-profissional-do-que-nunca/61548>>. Acesso em: 24 maio 2018.

Considerando os benefícios do Linux, escolha a opção correta (IBM 2018):

- a) O Linux é a melhor opção no mercado de servidores, porém em função da grande dificuldade de suporte, as empresas acabam optando por outros servidores.
- b) O Linux é altamente customizável, porém tem um alto consumo de energia.
- c) Devido à sua capacidade de ser escalável e usar qualquer um dos processadores integrados disponíveis no mercado,

o Linux se torna uma plataforma pouco usada em televisões, entretenimento em automóveis, sistemas de navegação e muitos outros tipos de dispositivos.

- d) Qualquer pessoa pode obter um kernel Linux e agrupar um conjunto de aplicativos de usuário para um modelo de uso específico, aproveitando os diversos benefícios do Linux.
- e) Um dos motivos para não usar o Linux em plataformas de smartphone de rápido crescimento é sua baixa versatilidade, sendo que muitos modelos não são contemplados pelo Linux.

3.

[O Linux Educacional é] uma das distribuições brasileiras de maior sucesso e talvez uma das mais injustiçadas. Linux Educacional está presente em uma grande parte dos computadores em escolas públicas, graças a sua customização voltada para a educação. Ela é desenvolvida pelo Centro de Computação Científica e Software Livre (C3SL) da Universidade Federal do Paraná (UFPR) e com apoio dos técnicos dos Núcleos de Tecnologia Educacional, do Ministério da Educação (MEC). Ela já foi baseada no Debian e hoje é baseada no Kubuntu. Só pelo gabarito de quem desenvolve a distribuição e da matriz da família Ubuntu já dava para ver que o sistema teria uma qualidade excelente. Teria e tem, mas infelizmente o preconceito e a falta de planejamento de muitos estados e municípios sujaram a imagem da distribuição. Muitos colocam a culpa pela falta de planejamento, suporte e treinamento dos profissionais de educação no sistema. Apesar de ser estável, ter um número magníficos de programas próprio para a educação e ter repositórios com conteúdo invejável, fazendo a distribuição ser a melhor opção frente ao Microsoft Windows, o número de pessoas jogando pedras nesse sistema GNU/Linux é incrível, fazendo até pensar que o “Complexo de Vira-Latas”, explicado por Nelson Rodrigues como um comportamento adotado por brasileiros que se colocam em um patamar de inferioridade em relação ao resto do mundo, é real e é um responsável maior ainda do que a ingerência estatal já falada.



Sobre o texto acima referente ao Linux Educacional e aos problemas enfrentados diante da resistência do uso do sistema operacional, analise as seguintes afirmações:

I – Conforme o texto apresenta, o não sucesso do uso do Linux Educacional está relacionado somente à falta de planejamento, suporte e treinamento dos profissionais de educação no sistema.

II – O problema relatado no texto é porque os brasileiros possuem um complexo de inferioridade em relação ao restante do mundo, por isso, o sistema não foi um sucesso.

III – A questão do não sucesso do uso do Linux Educacional está diretamente ligada ao Governo que não apoia o uso de software livre.

IV – Na verdade, a resistência das pessoas em usar o Linux Educacional é um dos motivos do não sucesso, uma vez que elas estão tão presas no Windows, que qualquer dificuldade de uso é um motivo para não usar o sistema operacional. Também pode ser que tenha ocorrido uma falta de planejamento, suporte e treinamento dos profissionais de educação no sistema, o que não invalida seu uso.

Assinale a alternativa que apresenta a resposta correta:

- a) I e III estão corretas.
- b) I e IV estão corretas.
- c) II e III estão corretas.
- d) II e IV estão corretas.
- e) IV está correta.

Referências

ALECRIM, E. **Cluster: conceito e características**. Disponível em: <<https://www.infowester.com/cluster.php>>. Acesso em: 18 mar. 2018.

FERREIRA, J. L. **YouOS- Seu sistema operacional na internet**. Viva o Linux, 2007. Disponível em: <<https://www.vivaolinux.com.br/artigo/YouOS-Seu-sistema-operacional-na-internet/>>. Acesso em: 24 maio 2018.

G1. **Android passa Windows e se torna o sistema operacional mais usado do mundo**. Disponível em: <<https://g1.globo.com/tecnologia/noticia/android-passa-windows-e-se-torna-o-sistema-operacional-mais-usado-do-mundo.ghtml>>. Acesso em: 20 abr. 2018.

GCFAPRENDELIVRE. **Os sistemas operacionais para dispositivos móveis**. Disponível em: <https://www.gcfaprendelivre.org/tecnologia/cursinho/informatica_basica/sistemas_operacionais/5.do>. Acesso em: 24 maio 2018.

HAMMERSCHMIDT, R. **O que são máquinas virtuais?** Disponível em: <<https://www.tecmundo.com.br/maquina-virtual/232-o-que-sao-maquinas-virtuais-.htm>>. Acesso em: 15 mar. 2018.

HIGA, P. **As 10 principais diferenças entre o Windows e o Linux**. Guia do PC, 2011. Disponível em: <<https://www.guiadopc.com.br/artigos/3394/as-10-principais-diferencias-entre-o-windows-e-o-linux.html>>. Acesso 24 maio 2018.

IBM. **Técnicas para proteger o Sistema Operacional**. Disponível em: <https://www.ibm.com/support/knowledgecenter/pt-br/SSEP7J_10.2.1/com.ibm.swg.ba.cognos.crn_arch.10.2.1.doc/c_securing_the_operating_system.html>. Acesso em: 20 abr. 2018.

_____. **Os Sistemas Operacionais estão em toda parte**. Disponível em: <https://www.ibm.com/developerworks/community/blogs/tlcbr/entry/os_sistemas_operacionais_estao_em?lang=en>. Acesso em: 20 abr. 2018.

MACEDO, G. **Dicas para Linux**: 10 distribuições Linux para você. Guia do PC, 2013. Disponível em: <<https://www.guiadopc.com.br/dicas/33798/10-distribuicoes-linux-voce.html>>. Acesso em: 24 maio 2018.

HARADA, E. **Do Windows 1 ao Windows 10**: os 29 anos de evolução do SO da Microsoft. TecMundo, 2014. Disponível em: <<https://www.tecmundo.com.br/windows-10/64136-windows-1-windows-10-29-anos-evolucao-do-so-microsoft.htm>>. Acesso em: 24 maio 2018.

LOPES, S. **O que é um sistema operacional?** Oficina da Net, 2017. Disponível em: <https://www.oficinadanet.com.br/artigo/851/o_que_e_um_sistema_operacional>. Acesso em: 20 abr. 2018.

MACHADO, F. B. **Arquitetura de sistemas operacionais**. 2. ed. Rio de Janeiro: LTC, 1997.

_____. **Fundamentos de sistemas operacionais**. 1. ed. Rio de Janeiro: LTC, 2011.

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

NÓBREGA FILHO, R. G. **Introdução aos Sistemas Operacionais**. Disponível em: <<http://www.di.ufpb.br/raimundo/SistOper/IntSO/IntrodSO.html>>. Acesso em: 20 abr. 2018.

OLHAR DIGITAL. **Windows, Mac ou Linux? Veja as vantagens e as falhas de cada um deles**. 2011. Disponível em: <https://olhardigital.com.br/video/analisamos_os_principais_sistemas_operacionais_no_mercado/18880>. Acesso em: 24 maio 2018.

_____. **Windows perde o posto de sistema operacional mais usado do mundo**. 2017. Disponível em: <<https://olhardigital.com.br/noticia/windows-perde-o-posto-de-sistema-operacional-mais-usado-do-mundo/67223>>. Acesso em: 20 abr. 2018.

OLIVEIRA, R. S.; CARISSIMI, A. S.; TOSCANI, S. S. **Sistemas Operacionais**. 4. ed. Porto Alegre: Bookman, 2010.

SANTINO, R. **Linux completa 25 anos de existência maior e mais profissional do que nunca**. Olhar Digital, 2016. Disponível em: <<https://olhardigital.com.br/noticia/linux-completa-25-anos-de-existencia-maior-e-mais-profissional-do-que-nunca/61548>>. Acesso em: 24 maio 2018.

SANTOS, A. TVs: comparando os sistemas operacionais. **Home Theater e Casa Digital**, 2015. Disponível em: <<http://revistahometheater.uol.com.br/portal/2015/11/04/tvs-comparando-os-sistemas-operacionais/>>. Acesso em: 23 abr. 2018.

SILVA, J. E. **Entendendo a estrutura do Linux**. Viva o Linux, 2003. Disponível em: <<https://www.vivaolinux.com.br/artigo/Entendendo-a-estrutura-do-Linux>>. Acesso em: 24 maio 2018.

SIQUEIRA F. **Estrutura do Sistema Operacional**. Disponível em: <<https://sites.google.com/site/profernandosiqueiraso/aulas/4-estrutura-do-sistema-operacional>>. Acesso em: 14 mar. 2018.

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

TECMUNDO. **Conheça outros sistemas operacionais**. Disponível em: <<https://www.tecmundo.com.br/macos/726-conheca-outros-sistemas-operacionais.htm>>. Acesso em: 20 abr. 2018.

TECHSOUP Brasil. **Onze dicas para solucionar problemas de software**. Disponível em: <<https://www.techsoupbrasil.org.br/node/1912>>. Acesso em: 20 abr. 2018.

TERRA. **Sistemas operacionais na nuvem são opção barata e segura**. 2012. Disponível em: <<https://www.terra.com.br/noticias/tecnologia/sistemas-operacionais-na-nuvem-sao-opcao-barata-e-segura,b1e8201fd70ea310VgnCLD200000bbcceb0aRCRD.html>>. Acesso em: 24 maio 2018.

UEYAMA, J. **Sistemas Operacionais – Aula 02 – Tipos e Estruturas de SO**. UNIVESP, 2017. Disponível em: <https://youtu.be/_J4CVHgXQ1c>. Acesso em: 23 abr. 2018.

VIVA O LINUX. O que é GNU/Linux. [s.d.]. Disponível em: <<https://www.vivaolinux.com.br/linux/>>. Acesso em: 24 maio 2018.

Processos e Threads 2

Convite ao estudo

Caro aluno, seja bem-vindo!

Nesta unidade veremos: o conceito, as características e o funcionamento de processo e threads e como eles são implementados, como os processos são criados e finalizados pelo sistema operacional e a hierarquia e os estados de processos. Um processo pode ser definido como um programa em execução. Porém, o sistema operacional não executa apenas os programas que solicitamos que sejam rodados.

Por exemplo, quando abrimos uma página em um navegador, ela pode acionar outros processos para que seja devidamente carregada, sem impactar as demais abas abertas. Além disso, existem outros processos executados no computador que não são propriedade do usuário, mas que são acionados para garantir a performance do sistema operacional, como o antivírus e o gerenciamento da rede.

Após o término desta unidade, você terá condições de entender o funcionamento e a implementação dos processos e threads. Aprendendo sobre os processos, você terá um conhecimento sobre o Gerenciador de Tarefas de seu sistema operacional, o que o ajudará a manter o bom funcionamento do seu computador.

Neste contexto, Lucas acabou de ser contratado como gerente da área de Tecnologia da Informação de uma empresa prestadora de serviços hospitalares. Além de fornecer suporte aos usuários dos sistemas hospitalares, outra função realizada na empresa é fazer pesquisas na internet sobre a qualidade de serviços na área da saúde, baixar livros e arquivos da área que auxiliam o desenvolvimento de aplicativos em TI e editar textos sobre o processo de avaliação e certificação da qualidade dos serviços hospitalares. Sempre que uma dessas atividades realizadas por Lucas é executada, um processo é criado e

ele pode chamar outros processos para auxiliá-lo por meio da transferência de informações entre eles. Para escolher qual processo será executado, o escalonador é acionado pelo sistema operacional e define os critérios de acesso à CPU que cada processo terá para que não afete a execução das aplicações. Ao receber um chamado de atendimento de suporte aos usuários dos sistemas hospitalares, um estagiário da equipe liderada por Lucas o aciona para ir até a sala do usuário que realizou a abertura do chamado. Durante o percurso, Lucas é questionado sobre como os processos são criados e finalizados, sobre como o sistema operacional trata a hierarquia de processos e seus estados, sobre o funcionamento da comunicação entre processos e seus problemas clássicos da comunicação e sobre como trabalham o escalonador de processos e os algoritmos de escalonamento. Para entender melhor o funcionamento dos processos e os algoritmos de escalonamento, Lucas sugere ao estagiário que seja implementado um algoritmo de escalonamento de processos por meio de semáforos.

Vamos juntos conhecer mais sobre essas questões através de algoritmos!

Seção 2.1

Processos

Diálogo aberto

Caro aluno, os processos são programas ou tarefas em execução e o sistema operacional é o responsável por administrá-los, por meio do gerenciador de processos. Existem os processos iniciados pelo usuário, como executar um editor de textos, abrir uma página na internet, abrir o aplicativo de músicas, entre outros. Há também os processos iniciados por outros processos, por exemplo, uma página da internet solicitando a ajuda de outro processo para fazer o carregamento dos seus elementos. Nessa seção você conhecerá o conceito, as características, a hierarquia e os estados dos processos e threads e como se dá a criação e o término de processos. Relembrando nosso contexto, Lucas acabou de ser contratado como gerente da área de Tecnologia da Informação de uma empresa prestadora de serviços hospitalares. Durante o atendimento do chamado (supervisionado por Lucas), o usuário questiona sobre o software de edição de arquivos usado na empresa, por travar muito ao formatar e salvar o arquivo. Para ambientar e auxiliar o estagiário, Lucas assumiu o atendimento, fechou o programa e o executou novamente, porém o software continuou travando. Ele então acionou o Gerenciador de Tarefas e identificou que o software de edição de arquivo ainda estava em execução. O estagiário fez os seguintes questionamentos: o que aconteceu para que o programa não tenha sido finalizado corretamente? Como fazer para conseguir executar o software corretamente? No caso de o software travar durante o salvamento e formatação, a implementação de threads auxiliaria o processo? Não se esqueça de compilar todas as informações obtidas para lhe auxiliar no desenvolvimento de um algoritmo de escalonamento de processos por meio de semáforos.

Para que você consiga responder esse e outros questionamentos, nesta seção vamos conhecer mais sobre os processos e os conteúdos pertinentes a este tema.

Bons estudos!

Não pode faltar

Um dos conceitos principais em sistemas operacionais gira em torno de processos.

Um processo pode ser definido como um programa em execução, porém o seu conceito vai além desta definição (MACHADO; MAIA, 2007).

Nos computadores atuais, o processador funciona como uma linha de produção executando vários programas ao mesmo tempo de forma sequencial, como ler um livro on-line, baixar um arquivo e navegar na internet. A CPU é responsável por alternar os programas, executando-os por dezenas ou centenas de milissegundos, para que cada um tenha acesso ao processamento, dando a ilusão ao usuário de paralelismo ou pseudoparalelismo (TANENBAUM, 2003). O pseudoparalelismo é a falsa impressão de que todos os programas estão executando ao mesmo tempo, mas na verdade o que acontece é que um processo em execução é suspenso temporariamente para dar lugar ao processamento de outro processo e assim sucessivamente.

Segundo Tanenbaum (2003), para tratar o paralelismo de forma mais fácil, foi desenvolvido um modelo responsável por organizar os programas executáveis em processos sequenciais. Um processo pode ser definido como um programa em execução incluindo os valores do contador de programa atual, registradores e variáveis. A CPU alterna de um processo para outro a cada momento, essa alternância é conhecida como multiprogramação.

A diferença entre processos e programa é importante para que o modelo seja entendido. A seguinte analogia é utilizada para exemplificar a diferença entre processo e programa: fazer um bolo. Para fazer um bolo, é necessário todos os ingredientes e a receita. A receita pode ser considerada como o programa, os ingredientes são os dados de entrada e a pessoa que prepara o bolo é o processador. Os processos são as atividades que a pessoa faz durante a preparação do bolo: ler a receita, buscar os ingredientes, misturar a massa e colocar o bolo para assar, que é o processo final desse programa "receita de bolo". Ainda neste exemplo, imagine que o filho da pessoa que está fazendo o bolo tenha se machucado. A pessoa guarda em que parte do processamento parou e vai socorrer o filho. Ela pega o kit de primeiros socorros e lê o procedimento

para tratar do machucado do filho. Neste momento, vemos o processador (a pessoa) alternando de um processo (fazer o bolo) para outro processo com prioridade maior (socorrer o filho), cada um com seu programa – receita versus procedimento de tratamento do machucado. Assim que o filho estiver medicado, então a pessoa retornará a fazer o bolo do ponto em que parou.

Podemos considerar então que um processo é uma atividade que contém um programa, uma entrada, uma saída e um estado. Veremos a seguir a criação de processo e os estados dos processos.



Exemplificando

Os serviços que os sistemas operacionais podem implementar através de processos são (MACHADO; MAIA, 2007):

- Auditoria e segurança do sistema.
- A contabilização do uso de recursos.
- A contabilização de erros.
- Gerência de impressão.
- Comunicação de eventos.
- Serviços de redes.
- Interface de comandos (Shell), entre outros.

Criação de processos

Os sistemas operacionais devem oferecer formas para que processos sejam criados. Segundo Tanenbaum (2003), existem quatro eventos que fazem com que um processo seja criado:

1) **Início do sistema:** quando o sistema operacional é inicializado, são criados vários processos. Existem os de primeiro plano, que interagem com os usuários e suas aplicações, e os de segundo plano, que possuem uma função específica, como um processo para atualizar e-mails quando alguma mensagem é recebida na caixa de entrada.

Para visualizar os processos em execução no Windows, pressione as teclas CTRL+ALT+DEL e no Linux utilize o comando ps.

2) **Execução de uma chamada ao sistema de criação por um processo em execução:** por exemplo, quando um processo está fazendo

download, ele aciona um outro processo para ajudá-lo. Enquanto um faz o download, o outro está armazenando os dados em disco.

3) Uma requisição do usuário para criar um novo processo: quando o usuário digita um comando ou solicita a abertura de um ícone para a abertura de um aplicativo.

4) Início de um job em lote: esses processos são criados em computadores de grande porte, os mainframes.



Assimile

Vimos nesta seção que existem os processos de primeiro plano, que são processos dos usuários e interagem com suas aplicações, e os de segundo plano, que possuem função específica. Os processos de segundo plano, ao executarem uma função específica, são chamados de daemons.

Término de processos

Após a criação, os processos podem ser finalizados nas seguintes condições, (TANENBAUM, 2003):

1) Saída normal (voluntária): acontece quando o processo acaba de executar por ter concluído seu trabalho.

2) Saída por erro (voluntária): acontece quando o processo tenta acessar um arquivo que não existe e é emitida uma chamada de saída do sistema. Em alguns casos, uma caixa de diálogo é aberta perguntando ao usuário se ele quer tentar novamente.

3) Erro fatal (involuntário): acontece quando ocorre um erro de programa, por exemplo, a execução ilegal de uma instrução ou a divisão de um número por zero. Neste caso, existe um processo com prioridade máxima que supervisiona os demais processos e impede a continuação do processo em situação ilegal.

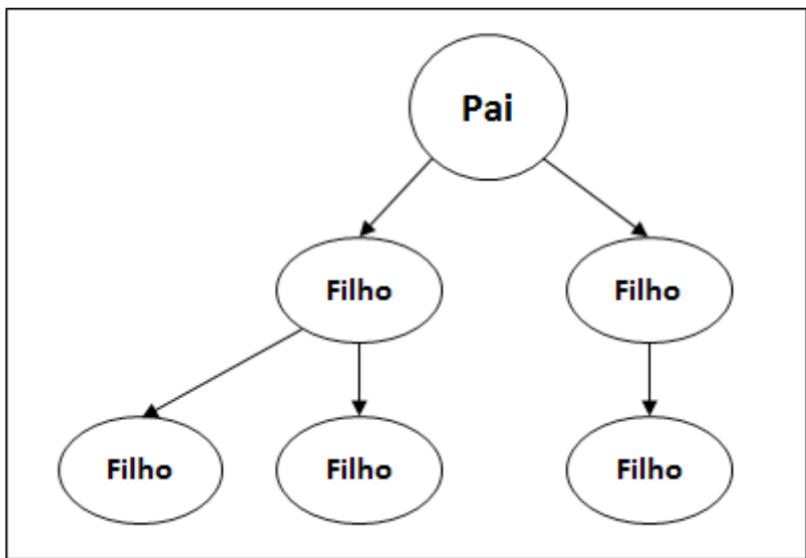
4) Cancelamento por um outro processo: acontece quando um processo que possui permissão emite uma chamada ao sistema para cancelar outro processo.

Hierarquia de Processos

Segundo Tanenbaum (2003), em alguns sistemas, quando um processo cria outro, o processo-pai e o processo-filho ficam

associados. O filho pode gerar outros processos, criando, assim, uma hierarquia de processos, conforme apresentado na Figura 2.1.

Figura 2.1 | Hierarquia de processos



Fonte: elaborada pelo autor.

No Unix, um processo-pai, seus filhos e descendentes formam um grupo de processos. Por exemplo, quando um usuário envia um sinal do teclado (como *CTRL + ALT + DEL*), este sinal é entregue para todos os processos que compõem o grupo de processos do teclado. Quando um processo-pai é “morto”, todos os filhos vinculados a ele são “mortos” também.

O Windows não possui uma hierarquia de processos. Cada um possui um identificador próprio e quando um processo cria outro, existe uma ligação entre eles, mas ela é quebrada quando o processo-pai passa seu identificador para outro processo. Quando um processo-pai é “morto”, os processos vinculados a ele não são mortos.

Estados do Processo

Os processos podem passar por diferentes estados ao longo do processamento. Um processo ativo pode estar em três estados (MACHADO; MAIA, 2007):

- **Em execução:** um processo está em execução quando está sendo processado pela CPU. Os processos são alternados para a utilização do processador.
- **Pronto:** um processo está no estado de pronto quando possui todas as condições necessárias para executar e está aguardando. O sistema operacional é quem define a ordem e os critérios para execução dos processos.
- **Espera ou Bloqueado:** um processo está no estado de espera quando aguarda por um evento externo (um comando do usuário, por exemplo) ou por um recurso (uma informação de um dispositivo de entrada/saída, por exemplo) para executar.

Quatro mudanças podem acontecer entre os estados, representadas na Figura 2.2.

Figura 2.2 | Transições dos processos



Fonte: Tanenbaum (2003, p. 58).

Conforme apresentado na Figura 2.2, a mudança 1 ("Em execução" para "Bloqueado") acontece quando um processo aguarda um evento externo ou uma operação de entrada/saída e não consegue continuar o processamento.

As mudanças 2 ("Em execução" para "Pronto") e 3 ("Pronto" para "Em Execução") são realizadas pelo escalonador sem que o processo saiba. O escalonador de processos é o responsável por decidir em qual momento cada processo será executado (veremos as atividades do escalonador com detalhes na seção 2.3). A mudança 2 acontece quando o escalonador decide que o processo já teve tempo suficiente em execução e escolhe outro processo para executar. A mudança 3 ocorre quando os demais processos já

utilizaram o seu tempo de CPU e o escalonador permite a execução do processo que estava aguardando.

A mudança 4 (“Bloqueado” para “Pronto”) ocorre quando a operação de entrada/saída ou o evento externo que o processo estava esperando ocorre. Assim, o processo retorna para a fila de processamento e aguarda novamente a sua vez de executar.

Threads

Segundo Machado e Maia (2007), o conceito de thread foi introduzido para reduzir o tempo gasto na criação, eliminação e troca de contexto de processos nas aplicações concorrentes, assim economizando recursos do sistema como um todo.

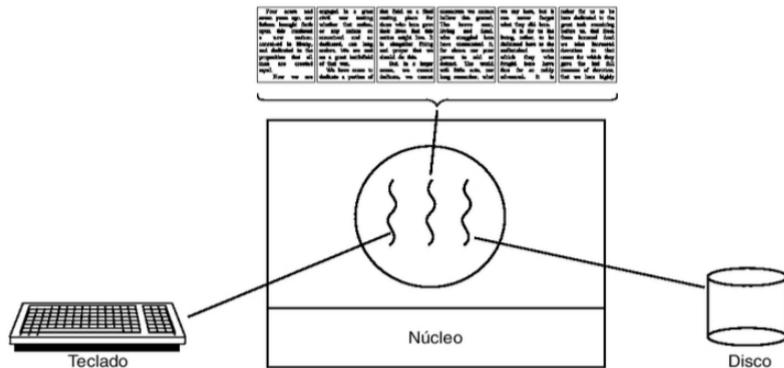
Thread é um fluxo de controle (execução) dentro do processo, chamado também de processos leves. Um processo pode conter um ou vários threads que compartilham os recursos do processo.

A principal razão para o uso de thread é que as aplicações da atualidade rodam muitas atividades ao mesmo tempo e quando são compostas por threads, podem ser executadas em paralelo. Outro motivo para a criação de thread é que são mais fáceis de criar e destruir, por não terem recursos vinculados a eles. Em relação ao desempenho, quando uma aplicação processa muitas informações de entrada/saída, o uso de threads acelera a execução da aplicação.

Para entender melhor o conceito de thread, vamos apresentá-lo utilizando como exemplo um processador de texto (TANENBAUM, 2003).

Imagine que você esteja escrevendo um livro e que seja necessário trocar um termo errado escrito em diversos capítulos. Se o livro estiver em um único arquivo, fica fácil fazer a alteração, porém se estiver em arquivos separados, será necessário fazer a correção em cada um. Outra questão é: se uma ou duas páginas forem removidas de um arquivo de 500 páginas, o processador de textos terá que formatar todo o documento em todas as páginas até chegar à página retirada, o que causará uma grande demora. Neste caso, se forem utilizados threads, um thread auxiliaria a formatação do texto, outro atenderia aos requisitos do usuário via teclado e um terceiro thread faria o backup dos dados em disco sem interferir nas ações dos demais threads, conforme apresentado na Figura 2.3.

Figura 2.3 | Processador de texto com três threads



Fonte: Tanenbaum (2003, p. 64).

Outro exemplo do uso de threads é em um navegador web, enquanto um thread carrega imagens ou textos de uma página e outro thread recupera dados de uma rede.



[Pesquise mais](#)

Algumas linguagens de programação da atualidade, como Java, possuem recursos para a implantação de threads, chamados programação concorrente. Para saber mais sobre esse assunto, leia o artigo: LANHELLAS, R. **Trabalhando com Threads em Java**. DEVMEDIA, [s.l.], 2013. Disponível em: <<https://www.devmedia.com.br/trabalhando-com-threads-em-java/28780>>. Acesso em: 11 jul. 2018.

Implementação de Processos

Para implementar o modelo de processos, o sistema operacional mantém um quadro de processos contendo informações sobre o estado do processo, seu contador de programa, o ponteiro da pilha, a alocação de memória, o status dos arquivos abertos, entre outros, que permitem que o processo reinicie do ponto em que parou (TANENBAUM, 2003). O Quadro 2.1 apresenta os campos mais importantes. No primeiro campo estão os dados necessários para o armazenamento, referentes ao gerenciamento de processos. O segundo e o terceiro campos referem-se aos dados do gerenciamento de memória e ao gerenciamento de arquivos.

Quadro 2.1 | Processos

Gerenciamento de processos	Gerenciamento de memória	Gerenciamento de arquivos
Registradores	Ponteiro para o segmento de código	Diretório-raiz
Contador do programa	Ponteiro para o segmento de dados	Diretório de trabalho
Palavra de estado do programa	Ponteiro para o segmento de pilha	Descritores de arquivos
Ponteiro de pilha		Identificador (ID) do usuário
Estado do processo		Identificador (ID) do grupo
Prioridade		
Parâmetros de escalonamento		
Registro identificador (ID) de processadores		
Processo-pai		
Grupo do processo		
Sinais		
Momento em que o processo iniciou		
Tempo usado da CPU		
Tempo de CPU do filho		
Momento do próximo alarme		

Fonte: Tanenbaum (2003, p. 59).

Implementação de Threads

A implementação de threads pode ocorrer no espaço do usuário, no núcleo do sistema operacional e em uma implementação híbrida (no espaço do usuário e do núcleo).

- **Thread de usuário:** são implementados pela aplicação do usuário e o sistema operacional não sabe de sua existência. A vantagem é que não é necessária nenhuma mudança entre os modos de usuário e núcleo, tornando-se rápido e eficiente.

- **Thread do núcleo:** são implementados e gerenciados pelo núcleo do sistema operacional. A desvantagem desta implementação é que todo o gerenciamento dos threads é feito por chamadas ao sistema, o que compromete a performance do sistema.
- **Threads híbridos:** são implementados tanto no espaço do usuário quanto no núcleo do sistema operacional. O sistema operacional sabe dos threads do usuário e faz o seu gerenciamento. A vantagem desta implementação é a flexibilidade em função das duas implementações.



Refletia

Diante do que aprendemos sobre processos e threads, quais são os desafios de implementar processos e threads em sistemas operacionais multiprogramados?

Sem medo de errar

Agora que você já aprendeu sobre os conceitos, as características e o funcionamento de processo e threads, sobre como os processos são criados e finalizados pelo sistema operacional, sobre a hierarquia e os estados de processos e sobre como os processos e threads são implementados, vamos voltar ao nosso contexto? Lucas estava auxiliando o atendimento realizado por um estagiário de sua equipe e surgiram os seguintes questionamentos: o que aconteceu para que o programa não tenha finalizado corretamente? Como fazer para conseguir executar o software corretamente? No caso de o software travar durante o salvamento e a formatação, a implementação de threads auxiliaria o processo? O problema de travamento de um software pode acontecer tanto em função do próprio software, quanto por causa de um hardware. Como Lucas identificou que o processo estava em execução ao acionar o Gerenciador de Tarefas, porém estava travado, constatou-se o problema no processo que estava rodando. Neste caso, o processo travou por algum erro encontrado, por exemplo, na memória durante a gravação do arquivo feita pelo editor de texto. O correto é “matar” o processo

e reiniciar o software de edição. Ao fazer isso, verifique se outro processo em execução impactará no salvamento do arquivo e, em caso positivo, feche-o antes de reiniciar o software de edição. A implementação de threads durante a execução do processo agiliza o processamento e melhora o desempenho das aplicações. O próprio conceito de threads diz “são as entidades programadas para a execução sobre a CPU” (TANENBAUM, WOODHULL, 2008, p.78). Logo, ao adicionar threads em uma aplicação, esta conseguirá processar duas ou mais coisas ao mesmo tempo. Por exemplo, no caso do software de edição de arquivos, um thread seria responsável por receber os comandos do usuário via teclado, outro thread seria responsável por formatar o arquivo e outro seria responsável por salvar o arquivo, todos os threads trabalhando em sincronia e executando as atividades em paralelo.

Avançando na prática

Erro fatal ao abrir um software

Descrição da situação-problema

Um outro chamado foi aberto para tratar o erro do Adobe Reader, pois o programa travou e fechou inesperadamente. Lucas, durante o atendimento, reiniciou o computador para ver se resolveria o problema, porém não resolveu. Assim, o usuário questionou: por que esse erro aconteceu e por que a reinicialização do computador não resolveu o problema, mesmo tendo “matado” o processo?

Resolução da situação-problema

O travamento e o fechamento inesperado de softwares podem acontecer por vários motivos, como conflito de hardware e software e dados corrompidos em arquivos. Nem sempre reiniciar o computador resolverá o problema. No caso relatado pelo usuário em relação ao travamento e ao fechamento inesperado do Adobe Reader, alguns processos de segundo plano, quando executados juntamente com ele, podem causar erros. É necessário fechar esses processos de segundo plano e reiniciar o computador para que o Adobe Reader seja executado. Para isso, vá ao Gerenciador de

Tarefas, clique na aba Processos, selecione o processo e clique em Finalizar processo. Após esta ação, reinicie o computador e abra novamente o Adobe Reader.

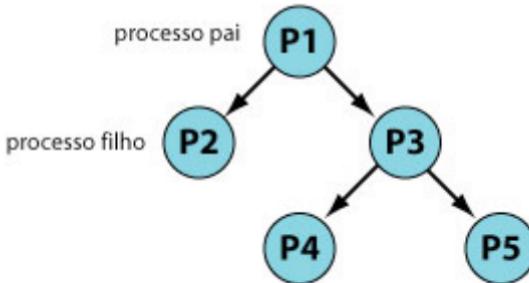
Faça valer a pena

1.



No Unix, processos podem gerar outros processos, dando a nomenclatura de processo "pai" e processo "filho" gerando assim uma hierarquia, alertando que processos filhos possuem apenas um processo pai, mas um processo pai pode ter vários processos filhos. O fato de ser um processo filho não impede que o mesmo também tenha processos filhos, veja na figura um exemplo de hierarquia onde o processo P1 está no topo: (SANCHES, 2012)

Figura 2.4 | Hierarquia de processos no UNIX



Fonte: Sanches (2012).

SANCHES, R. O. Hierarquia de Processos no Unix e Windows. DEVMEDIA, [s.l.], 2012. Disponível em: <<https://www.devmedia.com.br/hierarquia-de-processos-no-unix-e-windows/24739>>. Acesso em: 11 jul. 2018.

Analise as asserções a seguir:

A hierarquia de processos do Unix dificulta a propagação de vírus nos sistemas operacionais.

Porque

Quando um processo-pai é "morto" seja pelo sistema operacional ou pelo próprio usuário, todos os processos que estiverem abaixo dele na hierarquia serão "mortos" também.

Com relação às duas asserções assinale a alternativa correta:

- a) As asserções I e II são proposições verdadeiras, e a II é uma justificativa da I.
- b) As asserções I e II são proposições verdadeiras, mas a II não é uma justificativa da I.
- c) A asserção I é uma proposição verdadeira, e a II é uma proposição falsa.
- d) A asserção I é uma proposição falsa, e a II é uma proposição verdadeira.
- e) As asserções I e II são proposições falsas.

2.

Os sistemas operacionais executam de maneiras diferentes os processos e threads. No caso do Windows, ele trabalha com maior facilidade para gerenciar programas com apenas um processo e diversos threads do que quando gerencia vários processos e poucos threads. Isso acontece porque no sistema da Microsoft a demora para criar um processo e alterná-lo é muito grande. Enquanto isso, o Linux e os demais sistemas baseados no Unix podem criar novos processos de maneira muito mais rápida. No entanto, ao serem alterados, os programas podem apresentar o mesmo desempenho tanto no Linux quanto no Windows.



CANALTECH. O que é Thread? [S.I.], [s.d.]. Disponível em: <<https://canaltech.com.br/produtos/o-que-e-thread/>>. Acesso em: 11 jul. 2018.

Considerando as características entre processos e threads, escolha a opção correta:

- a) Durante a execução de um processo, caso ocorra algum erro e o processo seja finalizado, a thread continua executando, independentemente do processo.
- b) Threads podem ser escalonadas pelo sistema operacional e rodar como entidade independente dentro do processo.
- c) A facilidade que o Linux tem em criar processos é devido ao processo ser criado fora do kernel, o que melhora o desempenho do sistema operacional.
- d) A dificuldade do Windows de criar processos e alterná-los é em função do grande volume de vírus que impedem que o desempenho do sistema operacional seja satisfatório.
- e) Um processo pode conter vários threads e cada thread tem autonomia dentro do processo para realizar suas atividades, sendo que os demais threads não veem as mudanças realizadas por ele.

3.



Um processo passa por diferentes estados desde sua criação até seu término. Enquanto ele é criado, seu estado é considerado "Novo"; em ação, muda para "Executando"; quando depende da ocorrência de algum evento, vira "Esperando"; quando não mais necessário, o processo é "Terminado".

AMOROSO, D. O que são processos de um sistema operacional e por que é importante saber. TecMundo, [s.l.], 4 dez. 2009. Disponível em: <<https://www.tecmundo.com.br/memoria/3197-o-que-sao-processos-de-um-sistema-operacional-e-por-que-e-importante-saber.htm>>. Acesso em: 11 jul. 2018

Sobre os estados dos processos, marque a alternativa correta.

- a) Um processo no estado pronto significa que está executando (utilizando recursos da CPU).
- b) Um processo ativo pode estar em somente dois estados diferentes: pronto ou em execução.
- c) Um processo é considerado bloqueado (esperando) quando ele não pode prosseguir em função da alocação de outro processo ocupando a CPU pelo sistema operacional.
- d) Um processo não pode passar do estado pronto para o estado em execução, pois esta transição não é permitida.
- e) Um processo em execução não significa que ele está rodando e utilizando a CPU. O processo pode estar com o estado em execução e aguardando uma entrada externa, por exemplo um comando do usuário.

Seção 2.2

Comunicação entre processos

Diálogo aberto

Caro aluno, em um sistema operacional, os processos e threads trocam informações entre si ou solicitam a utilização de recursos simultaneamente, como arquivos, dispositivos de entrada/saída e memória. Um exemplo de comunicação interprocessos é a transferência de dados entre processos. Se um processo deseja imprimir um arquivo, ele o insere em um diretório de impressão com um nome para identificá-lo e outro processo é responsável por verificar periodicamente se existem arquivos a serem impressos. Nessa seção você saberá como é feita a comunicação entre processos e threads. Veremos alguns pontos sobre essa comunicação, como condições de disputa, regiões críticas e exclusão mútua com espera ociosa. Além disso, estudaremos os mecanismos de sincronização que resolvem a exclusão mútua: dormir e acordar, semáforos, monitores e troca de mensagens. Relembrando nosso contexto, Lucas acabou de ser contratado como gerente da área de Tecnologia da Informação de uma empresa prestadora de serviços hospitalares e, durante o acompanhamento do atendimento realizado por um estagiário, em relação aos constantes travamentos do software de edição de arquivos ao salvar ou realizar a formatação, eles observam que outro usuário está fazendo o download de um arquivo. Durante o download, um processo (A) é responsável por baixar o arquivo e outro processo (B) é responsável por gravar estas informações em disco. Caso tenha outro processo acessando o disco, o processo (B) aguarda a sua vez para poder acessar o disco e gravar as informações. Caso contrário, o processo (B) grava os dados em disco e libera o recurso compartilhado para outro processo. Dessa forma, o estagiário faz os seguintes questionamentos a Lucas: quais são os desafios da comunicação entre os processos? Quais os problemas clássicos da comunicação entre processos? E se fosse utilizado um algoritmo de escalonamento de processos por meio de semáforos? Qual a importância e os benefícios de sua utilização? Para que

você consiga responder esse e outros questionamentos sobre a comunicação entre processos e os mecanismos de sincronização que resolvem a exclusão mútua, nesta seção vamos conhecer mais sobre os conteúdos pertinentes a este tema.

Bons estudos!

Não pode faltar

Em uma aplicação concorrente (execução cooperativa de processos e threads), os processos precisam se comunicar entre eles, então solicitam o uso de recursos como memória, arquivos, dispositivos de entrada/saída e registros. Por exemplo, uma região de memória é compartilhada entre vários processos. O sistema operacional deve garantir que esta comunicação seja sincronizada para manter o bom funcionamento do sistema e a execução correta das aplicações.

Segundo Tanenbaum (2003), é necessário levar em consideração três tópicos:

1. Como um processo passa a informação para outro processo.
2. Garantir que dois ou mais processos não invadem uns aos outros quando estão em regiões críticas (será detalhada no decorrer da seção). Quando um processo estiver usando uma região de memória, o outro processo deve aguardar a sua vez.
3. É necessário existir uma hierarquia quando houver dependências. Se o processo A produz dados e o processo B os imprime, B deve esperar até que A produza dados para serem impressos.

Estas questões serão discutidas ao longo desta seção e também se aplicam a threads.

Condições de disputa ou condições de corrida

Condições de disputa ou condições de corrida acontecem quando dois ou mais processos estão compartilhando alguma região da memória (lendo ou escrevendo dados) e o resultado final depende das informações de quem executa e quando.

Como exemplo, podemos citar o problema da Conta_Corrente relatado por Machado e Maia (2007). Nesta situação, o saldo

bancário de um cliente é atualizado por meio de um programa após o lançamento de um débito ou crédito no arquivo de contas correntes (nesta arquivo são armazenadas informações sobre o saldo dos correntistas do banco). O registro do cliente e o valor depositado ou sacado são lidos por um programa e o saldo do cliente é atualizado. Suponha que dois funcionários do banco atualizem o saldo do mesmo cliente simultaneamente. O processo do primeiro funcionário lê o registro do cliente e soma ao saldo o valor sacado pelo cliente. Porém, antes de gravar o novo saldo no arquivo, o segundo funcionário lê o registro do mesmo cliente que está sendo atualizado e lança um crédito a ser somado ao saldo. Independentemente do processo que atualizar primeiro, o dado gravado no arquivo referente ao saldo está inconsistente.

Regiões críticas

Para impedir as condições de disputa, é necessário definir maneiras que impeçam que mais de um processo leia e escreva ao mesmo tempo na memória compartilhada. Esses métodos são chamados de exclusão mútua, ou seja, quando um processo estiver lendo ou gravando dados, sua região crítica ou processo deve esperar.

A parte do programa em que o processo acessa a memória compartilhada é chamada de região crítica ou seção crítica.

Segundo Tanenbaum (2003), para termos uma boa solução, é necessário satisfazer quatro itens:

1. Dois ou mais processos jamais estarão ao mesmo tempo em suas regiões críticas.
2. Não se pode afirmar nada sobre o número e a velocidade de CPUs.
3. Nenhum processo que esteja executando fora de sua região crítica pode bloquear outros processos.
4. Nenhum processo deve esperar sem ter uma previsão para entrar em sua região crítica.

A seguir veremos as soluções propostas para realizar a exclusão mútua: exclusão mútua com espera ociosa, dormir e acordar, semáforos, monitores e troca de mensagens.

Exclusão mútua com espera ociosa

Segundo Tanenbaum (2003), existem alguns métodos que impedem que um processo invada outro quando um deles está em sua região crítica. São eles:

Desabilitando interrupções

Nesta solução, as interrupções são desabilitadas por cada processo (qualquer parada que pode ocorrer por um evento) assim que entra em sua região crítica e ativadas novamente antes de sair dela. Desta forma, a CPU não será disponibilizada para outro processo.

Esta solução não é prudente, uma vez que, ao dar autonomia para processos, a multiprogramação fica comprometida. Se um processo, ao entrar em sua região crítica, desabilitasse as interrupções e se esquecesse de habilitá-las novamente ao sair, o sistema estaria comprometido. Em sistemas com múltiplos processadores, a interrupção acontece em apenas um processador e os outros acessariam normalmente a memória compartilhada, comprometendo essa solução.



Assimile

A solução de exclusão mútua com espera ociosa, como desabilitando interrupções e instrução TSL são mecanismos implementados no hardware do computador. As demais soluções são implementadas via software.



Exemplificando

Suponha que um processo de atualização da base de dados de uma empresa de vendas de seguros de vida desabilite as interrupções durante a noite para fazer uma nova carga da base de dados. Esta interrupção é necessária, uma vez que a carga demorará aproximadamente dez horas. Se o processo finalizar e não reabilitar as interrupções, a base de dados no dia seguinte não poderá ser usada e a consequência pode ser a perda financeira e de fidelização de clientes.

Variáveis de impedimento

Esta solução contém uma variável chamada *lock*, inicializada com o valor 0. Segundo Tanenbaum (2003), o processo testa e verifica o valor dessa variável antes de entrar na região crítica e, caso o valor seja 0, o processo o altera para 1 e entra na região crítica. Caso o valor da variável seja 1, o processo deve aguardar até que seja alterado para 0.

Variáveis de impedimento não resolvem o problema de exclusão mútua e ainda mantêm a condição de disputa. Quando o processo 1 vê o valor da variável 0 e vai para alterar o valor para entrar na região crítica, chega o processo 2 e altera o valor da variável para 1, antes de o processo 1 ter alterado. Logo, os dois processos entram, ao mesmo tempo, na região crítica.

Alternância obrigatória

Segundo Tanenbaum (2003), essa solução utiliza uma variável *turn* compartilhada que informa qual processo poderá entrar na região crítica (ordem). Esta variável deve ser alterada para o processo seguinte, antes de deixar a região crítica. Suponha que dois processos desejam entrar em sua região crítica. O processo A verifica a variável *turn* que contém o valor 0 e entra em sua região crítica. O processo B também encontra a variável *turn* com o valor 0 e fica testando continuamente para verificar quando ela terá o valor 1.

O teste continuo é chamado de espera ociosa, ou seja, quando um processo deseja entrar em sua região crítica, ele examina se sua entrada é permitida e, caso não seja, o processo fica esperando até que consiga entrar. Isso ocasiona um grande consumo de CPU, podendo impactar na performance do sistema.

Ainda segundo Tanenbaum (2003), assim que o processo A deixa sua região crítica, a variável *turn* é atualizada para 1 e permite que o processo B entre em sua região crítica. Vamos supor que o Processo B é mais ágil e deixa a região crítica. Os processos A e B estão fora da região crítica e *turn* possui o valor 0. O processo A finaliza antes de ser executado em sua região não crítica. Como o valor de *turn* é 0, o processo A entra de novo na região crítica, e o processo B ainda permanece na região não crítica. Ao deixar a

região crítica, o processo A atualiza a variável *turn* com o valor 1 e entra em sua região não crítica.

Os processos A e B estão executando na região não crítica e o valor da variável *turn* é 1. Se o processo A tentar entrar de novo na região crítica, não conseguirá, pois o valor de *turn* é 1. Desta forma, o processo A fica impedido pelo processo B, que **NÃO** está na sua região crítica. Esta situação viola a seguinte condição: nenhum processo que esteja executando fora de sua região crítica pode bloquear outros processos.

Solução de Peterson

Segundo Tanenbaum (2003), essa solução foi implementada por meio de um algoritmo que consiste em dois procedimentos escritos em C, baseado na definição de duas primitivas (*enter_region* e *leave_region*) utilizadas pelos processos que desejam utilizar sua região crítica. Antes de entrar na região crítica, todo processo chama *enter_region* com os valores 0 ou 1. Este apontamento faz com que o processo aguarde até que seja seguro entrar. Depois de finalizar a utilização da região crítica, o processo chama *leave_region* e permite que outro entre. Como a solução de Alternância Obrigatória, a Solução de Peterson precisa da espera ociosa.

Instrução TSL

Segundo Tanenbaum (2003), a instrução TSL (*test and set lock*, ou seja, teste e atualize a variável de impedimento) conta com a ajuda do hardware.

A instrução TSL RX, LOCK faz uma cópia do valor do registrador RX para LOCK. Um processo somente pode entrar em sua região crítica se o valor de LOCK for 0. A verificação do valor de LOCK e sua alteração para 0 são realizadas por instruções ordinárias.

A solução de Alternância Obrigatória, a Solução de Peterson e a instrução TSL utilizam a espera ociosa.

Dormir e acordar

As soluções apresentadas até aqui utilizam a espera ociosa (os processos ficam em um laço ocioso até que possam entrar na

região crítica). Para resolver este problema, são realizadas chamadas *sleep* (dormir) e *wakeup* (acordar) ao sistema, que bloqueiam/desbloqueiam o processo, ao invés de gastar tempo de CPU com a espera ociosa.

A chamada *sleep* faz com que o processo que a chamou durma até que outro processo o desperte, e a chamada *wakeup* acorda um processo.

Para exemplificar o uso dessas chamadas, vamos apresentar o problema do produtor/consumidor.

Problema do produtor/consumidor

De acordo com Tanenbaum (2003), o problema do produtor/consumidor normalmente acontece em programas concorrentes em que um processo gera informações (produtor) para uso de outro processo (consumidor).

Imagine que dois processos compartilham um *buffer* (memória) de tamanho fixo. O problema acontece quando o produtor quer inserir um novo item, porém o *buffer* está cheio ou o consumidor deseja remover um item e o *buffer* está vazio.

A solução é colocar o processo, impedido pela capacidade do *buffer*, para dormir (através da chamada *sleep*) até que o outro modifique o *buffer* e acorde o anterior (por meio da chamada *wakeup*). Para controlar a quantidade de itens no *buffer*, é utilizada uma variável *count*.

O problema da condição de disputa acontece quando o *buffer* está vazio e o consumidor verifica o valor da variável *count*. O escalonador decide parar a execução do consumidor e executa o produtor, que inclui itens no *buffer* enviando um sinal de acordar para o consumidor. Como o consumidor ainda não está logicamente adormecido, o sinal para acordar é perdido e tanto o consumidor, quanto o produtor ficarão eternamente adormecidos.

Semáforos

Segundo Machado e Maia (2007), a utilização de semáforos é um dos mecanismos utilizados em projetos de sistemas operacionais e em aplicações concorrentes. Hoje, grande parte das linguagens

de programação disponibiliza procedimentos para que semáforos sejam utilizados.

Um semáforo é uma variável inteira que realiza duas operações: DOWN (decrementa uma unidade ao valor do semáforo) e UP (incrementa uma unidade ao valor do semáforo). As rotinas DOWN e UP são indivisíveis e executadas no processador. Um semáforo com o valor 0 indica que nenhum sinal de acordar foi salvo e um valor maior que 0 indica que um ou mais sinais de acordar estão pendentes (TANENBAUM, 2003).

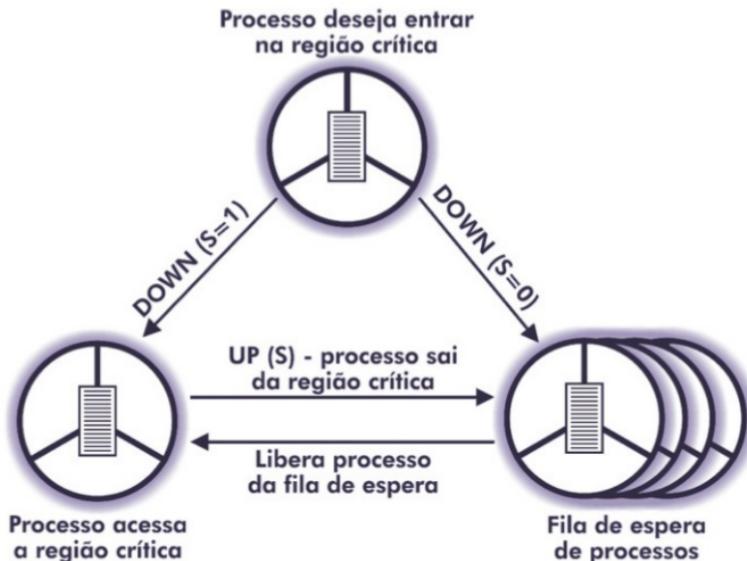
De acordo com Machado e Maia (2007), os semáforos são classificados como:

- Binários, também conhecidos como mutexes (*mutual exclusion semaphores*), que recebem os valores 0 ou 1.

- Contadores, que recebem qualquer valor inteiro positivo, além do 0.

Para compreendermos melhor o uso de semáforos, utilizaremos um exemplo de semáforos binários. Segundo Machado e Maia (2007), o semáforo com o valor igual a 1 significa que nenhum recurso está utilizando o processo e valor igual a 0 significa que o recurso está em uso. A Figura 2.5 apresenta o uso do semáforo binário na exclusão mútua. Quando um processo deseja entrar em sua região crítica, é executada a instrução DOWN. Caso o valor do semáforo seja igual a 1, o valor é decrementado e o processo pode entrar em sua região crítica. Caso o valor seja 0 e a operação DOWN seja executada, o processo é impedido de entrar em sua região crítica, permanecendo em fila no estado de espera. O processo que utiliza o recurso executa a instrução UP ao deixar a região crítica incrementa o valor do semáforo e libera o acesso ao recurso. Caso existam processos aguardando na fila para serem executados, o sistema selecionará um e alterará o seu estado para pronto.

Figura 2.5 | Semáforo binário



Fonte: Machado e Maia (2007, p. 108).

O problema do produtor/consumidor (perda de sinal de acordar) apresentado acima pode ser resolvido através de semáforos, conforme apresentado na Figura 2.6, em que três semáforos são usados: um chamado *mutex*, que controla o acesso à região crítica e é inicializado com o valor 1, outro chamado *full*, que conta os valores preenchidos no *buffer*, sendo o valor inicial 0 e o terceiro chamado de *empty*, que conta os lugares vazios no *buffer* (representa o número de lugares do *buffer*). Segundo Tanenbaum (2003), um processo, ao executar a operação DOWN, decrementa o seu valor inteiro. Caso o valor do semáforo seja negativo, o processo é impedido e inserido ao fim da fila. Quando um processo executa a operação UP, o seu valor inteiro é incrementado. Se existir algum processo impedido na fila, o primeiro processo é liberado.

Figura 2.6 | Problema do produtor/consumidor usando semáforos

```
#define N 100
typedef int semaphore;
semaphore mutex = 1;
semaphore empty = N;
semaphore full = 0;

void producer(void)
{
    int item;

    while (TRUE) {
        item = produce_item();
        down(&empty);
        down(&mutex);
        insert_item(item);
        up(&mutex);
        up(&full);
    }
}

void consumer(void)
{
    int item;

    while (TRUE) {
        down(&full);
        down(&mutex);
        item = remove_item();
        up(&mutex);
        up(&empty);
        consume_item(item);
    }
}
```

Fonte: Tanenbaum (2003, p. 82).



Pesquise mais

Para saber mais sobre o funcionamento de semáforos, acesse:

ROJAS, A. **Semáforos**. 8 out. 2014. Disponível em: <<https://youtu.be/8YTV7cMyOSU>>. Acesso em: 12 jul. 2018.

Monitores

Segundo Tanenbaum (2003), um monitor é uma coleção de rotinas, estrutura de dados e variáveis que ficam juntos em um módulo ou pacote. Também pode ser definido como uma unidade de sincronização de processos de alto nível.

Um processo, ao chamar uma rotina de um monitor, verifica se existe outro processo ativo. Caso esteja, o processo que chamou é bloqueado até que o outro deixe o monitor, senão, o processo que o chamou poderá entrar.

A utilização de monitores garante a exclusão mútua, uma vez que só um processo pode estar ativo no monitor em um determinado momento e os demais processos ficam suspensos até poderem estar ativos. O compilador é o responsável por definir a exclusão mútua nas entradas do monitor.

É preciso definir métodos para suspenderem os processos caso não possam prosseguir, mesmo que a implementação da exclusão mútua em monitores seja fácil.

Assim, é necessário introduzir variáveis condicionais, com duas operações: *wait* e *signal*. Se um método do monitor verifica que não pode prosseguir, um sinal *wait* é emitido (bloqueando o processo), permitindo que outro processo que estava bloqueado acesse o monitor. A linguagem de programação Java suporta monitores.



Refletia

Os monitores e semáforos são soluções para CPUs que utilizam memória compartilhada. E como seria esse processo em sistemas distribuídos, em que existem troca de informações entre processos que estão em máquinas diferentes?

Troca de Mensagens

Segundo Tanenbaum (2003), esse método utiliza duas chamadas ao sistema:

- *send (destination, &message)* - envia uma mensagem para um determinado destino.
- *receive (source, &message)* - recebe uma mensagem de uma determinada origem.

Caso nenhuma mensagem esteja disponível, o receptor poderá ficar suspenso até chegar alguma.

A troca de mensagens possui problemas como sua perda pela rede. Para evitar este problema, assim que uma mensagem é recebida, o receptor enviará uma mensagem de confirmação de recebimento. Caso o receptor receba e não confirme o recebimento, não será problema, uma vez que as mensagens originais são numeradas de forma sequencial.

Uma questão importante refere-se à autenticação, pois é necessário saber se a fonte é real. Além disso, as mensagens enviadas e recebidas não podem ser ambíguas.

Quanto ao desempenho, copiar mensagens é um procedimento mais lento do que realizar operações sobre semáforos ou monitores. Uma solução seria realizar a troca de mensagens através de registradores.



Assimile

Vimos nesta seção que, para resolver o problema de exclusão mútua, existem as seguintes soluções: exclusão mútua com espera ociosa, dormir e acordar, semáforos, monitores e troca de mensagens. Estes métodos são equivalentes quando implementados em um único processador.



Pesquise mais

Existem diversos problemas clássicos da comunicação interprocessos, como o problema do jantar dos filósofos e o problema do barbeiro sonolento. Para saber mais sobre eles, leia: TANENBAUM, A. S. **Sistemas Operacionais Modernos**. 2. ed. São Paulo: Pearson, 2003.

Sem medo de errar

Agora que você já aprendeu como os processos e threads fazem a comunicação, conheceu as questões sobre a comunicação entre processos como condições de disputa, regiões críticas e exclusão mútua com espera ociosa e estudou sobre os mecanismos de sincronização que resolvem a exclusão mútua dormir e acordar, semáforos, monitores e troca de mensagens, vamos voltar ao nosso contexto. Durante o atendimento do chamado referente

aos constantes travamentos do software de edição de arquivos ao salvar ou realizar a formatação, eles observam que outro usuário está fazendo o download de um arquivo. Como os sistemas operacionais precisam manter a comunicação entre processos para um bom funcionamento do sistema e para não comprometer a execução das aplicações, os desafios são grandes. Os processos que concorrem por recursos do computador devem ter suas execuções sincronizadas, garantindo o correto processamento das aplicações. Para isso, o sistema operacional deve implementar mecanismos de controle e sincronismo, como a exclusão mútua que consiste em não permitir que dois processos acessem o mesmo recurso ao mesmo tempo, garantindo assim que nenhum processo invada a região crítica de outro. Neste caso, foram desenvolvidas soluções para evitar a exclusão mútua, como: **dormir e acordar**, que faz com que um processo durma enquanto outro esteja acessando o recurso, até que seja acordado pelo outro processo; **semáforo**, uma variável responsável por contar a quantidade de sinais de dormir e acordar, sendo vantajoso por não existir a espera ocupada que desperdiça o processamento de máquina; entre outras soluções. Analisando esta situação, um dos mecanismos de sincronização de processos chama-se Semáforos. Os Semáforos garantem que processos não acessem um mesmo recurso simultaneamente através de duas variáveis *Down* e *Up*, que são verificadas e alteradas sem a possibilidade de interrupções. A variável *Down* é decrementada quando o processo grava os dados e libera o disco para outro processo. A variável *Up* é incrementada quando outro processo já está usando a CPU.

Existem os problemas clássicos da comunicação entre processos e existem vários métodos para resolver estes problemas. Podemos citar o problema do jantar dos filósofos e o problema do barbeiro adormecido, conforme apresentado em Tanenbaum (2003). O problema do jantar dos filósofos foi proposto e resolvido assim: cinco filósofos estão sentados em uma mesa circular e cada um tem um prato de espaguete e garfos. Porém, para comer são necessários dois garfos, pois o espaguete está escorregadio. Cada filósofo alterna entre comer e pensar e, quando algum deles fica com fome, tenta pegar os garfos da sua esquerda e direita, um de cada vez, para poder comer o espaguete. Caso consiga pegar os dois garfos, ele come por um tempo, liberando os garfos ao final

e voltando à sua atividade de pensar. O problema é não gerar o bloqueio dos filósofos implementando soluções para resolver a questão do acesso exclusivo a um número restrito de recursos. O problema do barbeiro adormecido acontece em uma barbearia. A barbearia tem um barbeiro, uma cadeira de barbeiro e várias cadeiras para que os clientes esperem a sua vez. Se não tiver clientes, o barbeiro adormece e quando um cliente chega, o barbeiro acorda e corta seu cabelo. Se chegarem outros clientes, eles verificam se há cadeiras de espera disponíveis e, caso não tenha, os clientes vão embora. O problema principal é programar o barbeiro e os clientes para não cairem em condição de disputa.

Avançando na prática

Condições de Disputa – venda de ingressos para a Copa do Mundo 2018

Descrição da situação-problema

O estagiário que trabalha na área de Tecnologia da Informação da empresa prestadora de serviços hospitalares comentou com Lucas que um de seus colegas é o responsável pelo monitoramento do sistema de vendas de ingressos para a Copa do Mundo de 2018. O estagiário comentou sobre a falha no sistema que ocasionou a venda duplicada de ingresso, problema que ocorreu em um ponto de venda de ingressos no Rio de Janeiro e em outro em Nova York (EUA). Tanto o vendedor do Rio de Janeiro quanto o de Nova York viram que o último ingresso para a partida entre Brasil e Estados Unidos estava disponível para a venda, então, ambos venderam o ingresso aos torcedores. Diante da situação relatada pelo colega do estagiário, o SAC do sistema de vendas de ingressos foi acionado e a falha foi reportada à empresa desenvolvedora do software. O estagiário fez as seguintes perguntas para Lucas: por que ocorreu a condição de disputa, uma vez que os vendedores são de países diferentes? Como garantir que essa condição não aconteça? Apresente um relatório explicando a falha ocorrida no sistema de vendas de ingressos para a Copa do mundo 2018.

Resolução da situação-problema

Em um sistema unificado de vendas, normalmente a aplicação desenvolvida deve garantir que não ocorram erros. Mesmo estando em países diferentes, a aplicação deve ser íntegra, não permitindo que problemas como a condição de disputa aconteça. É possível que tenha ocorrido uma falta de sincronismo, ocasionando essa condição, uma vez que os pontos de venda dependem da internet. Para garantir que isso não aconteça, é necessário verificar se a solução de exclusão mútua (que garante que um processo não terá acesso a uma região crítica enquanto outro estiver utilizando essa região) foi corretamente implementada através do método escolhido pelo desenvolvedor da aplicação. Logo, é preciso ver com a empresa que desenvolveu o sistema para reportar o erro.

Faça valer a pena

- 1.** O sistema operacional deve implementar formas de controle e sincronismo de processos que concorrem por recursos do computador, não permitindo que dois processos acessem o mesmo recurso ao mesmo tempo, garantindo assim a integridade do sistema e das aplicações.

Considerando as questões acerca da comunicação entre processos, marque a alternativa correta:

- a) A condição de disputa ocorre quando dois ou mais processos não podem acessar os recursos compartilhados e, com isso, há uma disputa entre os processos para utilizar o recurso.
- b) Seção crítica ou região crítica é a parte do programa que o processo não pode acessar por não ser compartilhada.
- c) Exclusão mútua é um método para evitar a condição de disputa, impedindo que mais de um processo acesse uma determinada área de memória compartilhada ao mesmo tempo.
- d) Em alguns casos, o sistema operacional permite que dois processos acessem simultaneamente sua região crítica.
- e) Se um processo estiver executando fora de sua região crítica, ele pode bloquear outro processo até finalizar a execução.

- 2.** Existem diversas soluções que resolvem o problema da exclusão mútua durante a comunicação entre processos. Entre elas podemos citar as

soluções de hardware, em que é implementada a solução diretamente no hardware (por exemplo, desativando interrupções) e de software, que são implementadas através de algoritmos.

Em relação às soluções de exclusão mútua, marque a alternativa correta.

- a) A solução de desativar as interrupções é uma das mais eficazes, pois nenhuma interrupção de relógio pode ocorrer e o processo pode finalizar a sua execução.
- b) A solução de Peterson é implementada através de uma variável e utilizada para permitir a entrada de um processo na região crítica quando um outro já está na região.
- c) A solução de dormir e acordar faz com que um processo "durma" até ser "acordado" por outro processo. Um dos problemas que pode ser resolvido com esta solução é a do produtor/consumidor.
- d) Quando um processo deseja entrar na região crítica, é atribuído um valor para a variável *lock*, em que o 0 significa que existem processos na região crítica e 1 significa que não existem processos. Essa solução é conhecida como semáforo.
- e) Na solução de monitores, vários processos podem estar ativos dentro do monitor em um mesmo instante.

3.



Há ocasiões onde é extremamente necessário que dois ou mais processos/threads acessem um único recurso comum. Caso esse tipo de paralelismo não ocorra de forma controlada, podemos fazer com que um processo "sequestre/atropelé" a operação de outro. É aí que entram nossos sinalizadores, vulgo semáforos. Com eles, é possível acesso controlado a processos, de forma que só haverá disponibilidade quando a operação em andamento for finalizada.

PEREIRA, J. W. Semáforos, Kernel, Memória Compartilhada e Cia. Dicas-L, [s.l.], 21 set. 2005. Disponível em: <http://www.dicas-l.com.br/arquivo/semaforos_kernel_memoria_compartilhada_e_cia.php#.WtTCJH9ryUk>. Acesso em: 12 jul. 2018.

Considerando as questões acerca dos semáforos, marque a alternativa correta.

- a) O semáforo é um mecanismo que resolve problemas apenas de sincronismo e não de exclusão mútua em sistemas operacionais.
- b) Um semáforo não pode ser usado em linguagens orientadas a objetos por ser de difícil implementação.

- c) Quando um processo muda o valor de um semáforo, um outro processo pode alterar este valor simultaneamente.
- d) Uma vez que uma operação sobre semáforos começa, ela não pode ser interrompida no meio e nenhuma outra operação sobre o semáforo deve ser iniciada.
- e) A variável Down decremente o valor do semáforo e não é necessário testar se é menor que 0, visto que a implantação de semáforos é muito segura.

Seção 2.3

Escalonamento de processos

Diálogo aberto

Prezado aluno, em um sistema operacional vários processos compartilham recursos ao mesmo tempo, e quem faz a escolha de qual processo deve ser executado é o escalonador, feita por meio de um algoritmo (algoritmo de escalonamento), sendo necessário seguir as seguintes premissas: dar a cada processo o tempo necessário de uso da CPU, verificar se a política estabelecida é cumprida e manter ocupadas todas as partes do sistema. Por exemplo, quando um processo termina sua execução, outro processo deve ser escolhido entre os que estão no estado “pronto para executar”. Nessa seção você verá como é realizado o escalonamento entre processos e threads e estudará os tipos de escalonamento e seus principais algoritmos. Relembrando nosso contexto, Lucas acabou de ser contratado como gerente da área de Tecnologia da Informação de uma empresa prestadora de serviços hospitalares. Um usuário abriu um chamado para atualização do Adobe Reader em seu computador. Durante o atendimento, Lucas acompanha a resposta do chamado, delegada a um estagiário, que inicia a atualização do software e observa que no computador do usuário existem dois processos ativos: A e B. O processo A vai atualizar o Adobe Reader, enquanto o processo B vai gravar em disco os dados da alteração de um arquivo iniciada pelo usuário. Neste caso, o escalonador, além de fazer todas as verificações necessárias, deve conferir a prioridade de execução, pois a gravação em disco é prioritária em relação à atualização de um aplicativo. O escalonador de processos deve escolher um algoritmo de escalonamento para realizar esta ação, e Lucas foi designado para implementar um algoritmo de escalonamento de processos por meio de semáforos, que deverá ser entregue ao diretor da área de TI do hospital. Dessa forma, o estagiário fez os seguintes questionamentos: quais critérios o escalonador leva em consideração na escolha do uso da CPU? Quais são os objetivos do algoritmo de escalonamento? Por que

em alguns algoritmos de escalonamento é necessário saber o tempo de execução de cada processo para executar? Como os sistemas operacionais atuais fazem a escolha do algoritmo de escalonamento a ser utilizado? A fim de conseguir responder essas e outras questões sobre o escalonamento de processos e sobre os algoritmos de escalonamento, vamos conhecer mais sobre eles, para que você consiga implementar um algoritmo de escalonamento de processos através de semáforos.

Bons estudos!

Não pode faltar

Segundo Tanenbaum (2003), nos computadores existem vários processos que competem pela CPU e é necessário que o sistema operacional escolha, de forma eficiente, os que estejam aptos a executar. O responsável por isso é o escalonador de processos, por meio da aplicação de algoritmos ou políticas de escalonamento para otimizar a utilização do processador, definindo o processo que ocupará a CPU.

Segundo Machado e Maia (2007), além de escolher o processo a ser executado, o escalonador deve prezar pelos critérios e pelos objetivos (Quadro 2.2).

Quadro 2.2 | Critérios e objetivos do escalonador de processos

Critérios	<ul style="list-style-type: none">- Utilização do processador: eficiência do uso da CPU mantendo o processador ocupado na maior parte do tempo.- Throughput: maximizar a produtividade (<i>throughput</i>), executando o maior número de processos em função do tempo.- Tempo de processador: tempo de execução do processo.- Tempo de espera: reduzir o tempo total que um processo aguarda na fila para ser executado.- Tempo de turnaround: minimizar o tempo que um processo leva desde sua criação até seu término, considerando a alocação de memória, tempo de espera e tempo do processador e aguardando as operações de entrada/saída.- Tempo de resposta: reduzir o tempo de resposta para as aplicações interativas dos usuários.
-----------	---

Objetivos

Dar privilégios para aplicações críticas.

- Balancear o uso da CPU entre processos.
- Ser justo com todos os processos, pois todos devem poder usar o processador.
- Maximizar a produtividade (*throughput*).
- Proporcionar menores tempos de resposta para usuários interativos.

Fonte: adaptado de Machado e Maia (2007).

Diferentes sistemas operacionais apresentam características de escalonamento distintas. Podemos citar como exemplos o sistema operacional em tempo real e o de tempo compartilhado. O primeiro prioriza as aplicações críticas, enquanto o segundo aloca todos os processos com tempo igual para acesso à CPU, a fim de que os processos não esperem muito tempo para ter acesso ao processamento.

Segundo Tanenbaum (2003), alternar processos é oneroso, uma vez que é necessário alternar do modo usuário para o modo núcleo para iniciar a execução. Nessa execução, o estado do processo e o mapa de memória devem ser salvos, armazenando os dados dos registradores na tabela de processos e, a cada troca de processos, a memória cache (memória de acesso rápido) é invalidada.

As principais situações que levam ao escalonamento, segundo Tanenbaum (2003), são:

- **A criação de um novo processo:** é necessário escolher entre executar o processo pai ou o filho.
- **O término de um processo:** quando um processo é finalizado, é necessário escolher outro para ser executado.
- **Bloqueio do processo:** quando um processo é bloqueado e está aguardando uma entrada/saída, é necessário escolher outro processo.
- **Interrupção de entrada/saída:** se a interrupção for gerada por um dispositivo que finalizou a execução, o processo passará de “bloqueado” para “pronto” e o escalonador deve escolher entre continuar executando o processo atual ou o que acabou de ficar pronto.
- **Interrupções de relógio:** a cada interrupção do hardware de relógio pode haver um escalonamento de processos.

Em relação ao tratamento das interrupções de relógio, os algoritmos ou políticas de escalonamento são classificados em não-preemptivo e preemptivo (MACHADO; MAIA, 2007). No não-preemptivo um processo executa até finalizar, independentemente do tempo de uso da CPU, ou até que seja bloqueado aguardando entrada/saída de outro processo. Este escalonamento foi implementado no processamento batch. Já no escalonamento preemptivo um processo é executado por um tempo pré-determinado e quando o tempo de execução dado ao processo finaliza, a CPU é alocada para outro processo. No escalonamento preemptivo é possível priorizar aplicações em tempo real em função dos tempos dados aos processos. Os algoritmos de escalonamento preemptivo são complexos, porém permitem a implantação de vários critérios de escalonamento.



Refletia

Já pensou na complexidade e na criticidade de se definir a quantidade certa de tempo de execução de cada processo (*quantum*)? Por que esse é um processo crítico? Quais seriam as implicações se este processo fosse malfeito?

Segundo Tanenbaum (2003), existem três ambientes diferentes de escalonamento: lote, interativo e tempo real. A seguir descreveremos cada um deles.

- Lote: como não existem usuários aguardando uma resposta, tanto algoritmos preemptivos como não-preemptivos são aceitáveis para sistemas em lote. Os algoritmos de escalonamento aplicados aos sistemas em lote são:

- **FIFO** (*First in first out*): primeiro a chegar, primeiro a sair. Neste algoritmo os processos são inseridos em uma fila à medida que são criados, e o primeiro a chegar é o primeiro a ser executado. Quando um processo bloqueia e volta ao estado de pronto, ele é colocado no final da fila e o próximo processo da fila é executado. O FIFO é um algoritmo não-preemptivo, simples e de fácil implementação.
- **Job mais curto primeiro** (*SJF – shortest job first*): é um algoritmo de escalonamento não-preemptivo, em que

são conhecidos todos os tempos de execução dos *jobs*. O algoritmo seleciona primeiro os *jobs* mais curtos para serem executados. Este algoritmo é recomendado quando todos os *jobs* estão disponíveis ao mesmo tempo na fila de execução. A Figura 2.7 (a) apresenta quatro *jobs* (A, B, C e D) aguardando numa fila, com os respectivos tempos de execução em minutos (8, 4, 4 e 4). Se os *jobs* forem executados nesta ordem, teremos uma média de espera de execução de 14 minutos (o retorno do *job* A é de 8 minutos, o retorno do *job* B é de 12 minutos ($8 + 4$), o retorno do *job* C é de 16 minutos ($12 + 4$) e o retorno do *job* D é de 20 minutos ($16 + 4$). Logo, $(8+12+16+20) / 4 = 14$ minutos. Se os *jobs* forem executados selecionando primeiramente o mais curto, conforme apresentado na Figura 2.7 (b), teremos uma média de espera de execução de 11 minutos (o retorno do *job* B é de 4 minutos, o do *job* C é de 8 minutos ($4 + 4$), o do *job* D é de 12 minutos ($8 + 4$) e o do *job* A é de 20 minutos ($12 + 8$)). Logo, $(4+8+12+20) / 4 = 11$ minutos.

Figura 2.7 | Escalonamento do *job* mais curto primeiro



Fonte: Tanenbaum (2003, p. 102).

Uma versão preemptiva para o algoritmo *job* mais curto é, primeiro, o algoritmo próximo de menor tempo restante. O escalonador conhece os tempos de execução e escolhe sempre o *job* cujo tempo restante ao seu término seja o menor. Quando um novo *job* chega na fila para execução, seu tempo total é comparado ao tempo restante do processo que está utilizando a CPU.

- **Interativo:** nos sistemas interativos, a preempção se faz necessária para que outros processos tenham acesso à CPU. Os algoritmos de escalonamento aplicados aos sistemas interativos e que podem também ser aplicados a sistema em lote são:

- **Escalonamento Round Robin:** é um algoritmo antigo, simples, justo e muito usado. Também é conhecido como algoritmo de escalonamento circular. Nele os processos são organizados em uma fila e cada um recebe um intervalo de tempo máximo (*quantum*) que pode executar. Se ao final de seu *quantum* o processo ainda estiver executando, a CPU é liberada para outro processo. A Figura 2.8 (a) mostra a lista de processos que são executáveis mantida pelo escalonador. Quando um processo finaliza o seu *quantum*, é colocado no final da fila, conforme apresentado na Figura 2.8 (b).

Figura 2.8 | Escalonamento Round Robin



Fonte: Tanenbaum (2003, p. 104).

Segundo Machado e Maia (2007), o *quantum* varia de acordo com a arquitetura do sistema operacional e a escolha deste valor é fundamental, uma vez que afeta a política do escalonamento circular. Os valores variam entre 10 e 100 milissegundos. O escalonamento circular é vantajoso porque não permite que um processo monopolize a CPU.

- **Escalonamento por prioridades:** o algoritmo por prioridades considera todos os processos importantes, sendo associados a ele uma prioridade e um tempo máximo de execução. Por exemplo, um processo que carrega os dados em uma página web deve ter uma prioridade maior do que um processo que atualiza em segundo plano as mensagens de correio eletrônico. Quando os processos estiverem disponíveis para execução, o que tiver a maior prioridade é selecionado para executar. Para que os processos com prioridades altas não sejam executados infinitamente, a cada interrupção de relógio o escalonador pode reduzir a prioridade do processo.

Segundo Machado e Maia (2007), as prioridades de execução podem ser classificadas em estática ou dinâmicas.

A prioridade estática não altera o valor enquanto o processo existir. Já a dinâmica ajusta-se de acordo com os critérios do sistema operacional.

- **Escalonamento garantido:** se existirem vários usuários (n) logados em uma máquina, cada um deles receberá $1/n$ do tempo total da CPU. O sistema gerencia a quantidade de tempo de CPU de cada processo desde sua criação.



Exemplificando

Para exemplificar o escalonamento garantido, imagine que em um computador multiusuário existam 4 usuários logados. O tempo de CPU estabelecido para cada usuário executar os seus processos são de 7 segundos. Assim, o processo do usuário 1 executa por 7 segundos e para, dando lugar ao 2, que também executa por 7 segundos. Inicia então o processo do usuário 3 que executa pelo mesmo tempo e para, iniciando o processo do usuário 4. Este ciclo se repete até que todos os processos dos usuários finalizem a sua execução.

- **Escalonamento por loteria:** o escalonamento por loteria é baseado em distribuir bilhetes aos processos e os prêmios recebidos por eles são recursos de sistema, incluindo tempo de CPU. Cada bilhete pode representar o direito a um *quantum* de CPU e cada processo pode receber diferentes números de bilhetes, com opções de escolha distintas. Também existem as ações como compra, venda, empréstimo e troca de bilhetes.
- **Escalonamento fração justa (fair-share):** nesse caso, cada usuário recebe uma fração da CPU. Por exemplo, se existem dois usuários conectados em uma máquina e um deles tiver nove processos e o outro tiver apenas um, não é justo que o usuário com o maior número de processos ganhe 90% do tempo da CPU. Logo, o escalonador é o responsável por escolher os processos que garantam a fração justa.



Pesquise mais

O LinSched é um simulador de *scheduler* do Linux que reside no espaço do usuário. Ele isola o subsistema do *scheduler* do Linux (que fica dentro do kernel) e desenvolve ao redor dele uma quantidade suficiente do ambiente do kernel para que possa ser executado dentro do espaço do usuário. Para saber mais sobre este *scheduler*, acesse: JONES, M. **Simulação do Linux Scheduler**. IBM, 12 abr. 2011. Disponível em: <<https://www.ibm.com/developerworks/br/library/linux-scheduler-simulator/index.html>>. Acesso em: 13 jul. 2018.

- **Tempo real:** nesses sistemas, o tempo é um fator importantíssimo e os processos, ao utilizarem a CPU, fazem seu trabalho rapidamente e são bloqueados, dando oportunidade para outros processos executarem. Segundo Machado e Maia (2007), o escalonamento por prioridades seria o mais adequado em sistemas de tempo real, uma vez que uma prioridade é vinculada ao processo e, assim, a importância das tarefas na aplicação são consideradas. No escalonamento de tempo real, a prioridade deve ser estática, além de não existir fatia de tempo para cada processo executar.



Assimile

Vimos nesta seção que no escalonamento de tempo real, o tempo é um fator crucial. Esse sistema pode ser classificado em crítico e não crítico. O crítico atende a todos os prazos de um processo. No não crítico, os prazos de tempo do processo não podem ser garantidos.

Escalonamento de Threads

Da mesma forma que processos são escalonados, threads também são. O escalonamento de threads depende se estas estão no espaço do usuário ou do núcleo. Se forem threads de usuário, o núcleo não sabe de sua existência e o sistema operacional escolhe um processo A para executar, dando a ele o controle de seu *quantum*. O escalonador do thread A escolhe qual thread deve executar, através dos algoritmos de escalonamento descritos anteriormente. Se forem threads do núcleo, o sistema

operacional escolhe um thread para executar até um *quantum* máximo e, caso o *quantum* seja excedido, o thread será suspenso (TANENBAUM, 2003).

Uma das diferenças entre threads do usuário e do núcleo é o desempenho, uma vez que a alternância entre threads do usuário e do núcleo consome poucas instruções do computador. Além disso, os threads do usuário podem utilizar um escalonador específico para uma aplicação (TANENBAUM, 2003).

Segundo Deitel, Deitel e Choffnes (2005), na implementação de threads em Java, cada thread recebe uma prioridade. O escalonador em Java garante que o thread com prioridade maior execute o tempo todo. Caso exista mais de um thread com prioridade alta, eles serão executados através de alternância circular.

Sem medo de errar

Agora que você já aprendeu sobre como é realizado o escalonamento de processos e threads e como os algoritmos de escalonamento funcionam, vamos relembrar o nosso contexto.

Durante a atualização do Adobe Reader no computador do usuário, enquanto o processo A atualiza o software, o processo B grava em disco os dados da alteração de um arquivo iniciada pelo usuário. O escalonador de processos deve escolher um algoritmo para realizar esta ação, e Lucas foi designado para implementar um algoritmo de escalonamento de processos por meio de semáforos, que deverá ser entregue ao diretor da área de TI da empresa prestadora de serviços hospitalares. Desta forma, Lucas foi questionado pelo estagiário: quais critérios o escalonador leva em consideração na escolha do uso da CPU? Quais são os objetivos do algoritmo de escalonamento? Por que em alguns algoritmos de escalonamento é necessário saber o tempo de execução de cada processo para executar? Como os sistemas operacionais atuais fazem a escolha do algoritmo de escalonamento a ser utilizado?

Em um computador multiprogramado, ou seja, que executa vários programas ao mesmo tempo, os processos disputam a CPU, sendo necessário que o escalonador defina qual processo executará. Escolher corretamente o processo é importante, uma vez que alternar processos é caro, sendo preciso se preocupar com o uso eficiente da CPU. Para isso, é necessário que o escalonador siga os seguintes critérios:

- Manter o processador ocupado a maior parte do tempo, prezando pela eficiência da CPU.
- Executar o maior número de processos em função do tempo.
- Reduzir o tempo total que um processo aguarda na fila para ser executado.
- Minimizar o tempo que um processo leva desde a sua criação até o seu término.
- Considerar a alocação de memória, o tempo de espera e do processador e aguardar as operações de entrada/saída.
- Reduzir o tempo de resposta para as aplicações interativas dos usuários.

Os objetivos do escalonador de processos são: dar privilégios para aplicações críticas, balancear o uso da CPU entre processos, ser justo com todos os processos, pois todos devem ter poder usar o processador, maximizar a produtividade (*throughput*) e proporcionar menores tempos de resposta para usuários interativos. Em alguns algoritmos de escalonamento, por exemplo, no **Job mais curto primeiro**, o tempo de execução de cada processo é conhecido, porque estes algoritmos eram utilizados em sistemas em lote e, nestes sistemas, era possível prever o tempo de execução de cada programa, por serem rotinas que executavam constantemente. Assim, o processo que possuía o menor tempo de execução era selecionado para ocupar a CPU.

Os sistemas operacionais atuais precisam levar em consideração o tempo de resposta rápido das aplicações dos usuários, manter em conformidade os processos com prioridades altas e baixas, entre outros. Em tempo de execução, o escalonador pode definir qual processo executará de acordo com a sua política de escalonamento. Além disso, ele deve manter a CPU ocupada na maior parte do tempo, executando o maior número de processos possíveis. Os escalonadores dos sistemas operacionais atuais são preemptivos, isto é, dividem o tempo do processador em partes e cada parte é alocada aos processos. Assim, todos os processos que chegam para executar possuem seu tempo de CPU garantido.

Após analisar o cenário da empresa prestadora de serviços hospitalares, Lucas implementou o algoritmo de escalonamento de processos por meio de semáforos Binários. Foi definido que quando nenhum recurso estiver sendo utilizado pelo processo, o valor do

semáforo será igual a 1, caso o valor seja igual a 0 significa que o recurso está em uso. O Quadro 2.3 apresenta o pseudocódigo utilizado por Lucas para implementar as funções *down* e *up*. Quando um processo deseja entrar em sua região crítica, é executada a instrução *down*. Se o valor do semáforo for igual a 1, o valor é decrementado e o processo pode entrar em sua região crítica. Caso o valor seja 0 e a operação *down* seja executada, o processo é impedido de entrar em sua região crítica, permanecendo em fila no estado de espera. O processo que está utilizando o recurso executa a instrução *up* ao sair da região crítica, incrementando o valor do semáforo e liberando o acesso ao recurso. Caso existam processos aguardando na fila para serem executados, o sistema selecionará um e alterará o seu estado para pronto.

Quadro 2.3 | Pseudocódigo Semáforo

Função Down	Função UP
<p>Se (semáforo = 1) então semáforo = semáforo – 1 entrar na região crítica Senão Processo bloqueado fimSe</p>	<p>Se (semáforo = 0) então semáforo = semáforo + 1 sair da região crítica Senão processo pronto fimSe</p>

Fonte: elaborado pelo autor.

A implantação de semáforos na empresa prestadora de serviços hospitalares foi bem executada e eliminou os problemas de concorrência entre processos.

Avançando na prática

Escalonamento por prioridades

Descrição da situação-problema

A empresa C&C, prestadora de serviços de TI como suporte via Service Desk e desenvolvimento de websites para pequenas e médias empresas de todos os ramos, possui filiais em Brasília, São

Paulo, Rio de Janeiro e Belo Horizonte. Diariamente, ocorrem videoconferências para alinhamento de procedimentos técnicos e para tratar de assuntos específicos dos projetos de TI das empresas para as quais a C&C presta serviços. As videoconferências são realizadas por um notebook que fica na sala de reuniões. João é o técnico de TI responsável pela videoconferência da sede da C&C e observou que o tempo de resposta da videoconferência da sua filial estava muito lento. João entrou no gerenciador de tarefas do computador e identificou que o processo responsável pela videoconferência não estava executando com a prioridade correta. Observou também que o processo de atualização de e-mails tinha uma prioridade superior ao da videoconferência. Diante do exposto, surgem os seguintes questionamentos: como um processo responsável pela videoconferência pode ter prioridade inferior ao processo de atualização de e-mails? A alteração de prioridade foi realizada pelo sistema operacional ou por outra pessoa? Como fazer para alterar a prioridade dos processos em execução?

Resolução da situação-problema

Um sistema de videoconferência trabalha em tempo real, sendo o tempo um fator importantíssimo. Em um computador, as prioridades são definidas pelo sistema operacional ou podem ser mudadas pelos usuários. Como relatado por João, as videoconferências são realizadas por um notebook que fica na sala de reuniões. Logo, alguém por interesse próprio alterou a prioridade da videoconferência e priorizou o processo de atualização de e-mails. Mesmo que a videoconferência tenha prioridade sobre o processo de atualização de e-mails, se o usuário fizer a alteração de prioridade, o sistema operacional vai obedecer a ordem de prioridades definida. Para alterar a prioridade dos processos, basta seguir os seguintes passos:

- No Windows: abra o Gerenciador de Tarefas e selecione o processo do qual deseja alterar a prioridade. Clique com o botão direito do mouse, selecione a opção “Definir prioridade” e clique na prioridade desejada.

- No Linux: as prioridades do Linux variam de -19 a 20. Para alterar a prioridade, basta digitar o seguinte comando: `renice -n prioridade -p $(pidof) ->` -n representa o parâmetro do comando, prioridade representa um número de -19 a 20 e o pidof representa o PID do processo.

Faça valer a pena

1. Os algoritmos de escalonamentos podem ser classificados em preemptíveis e não-preemptíveis. De acordo com esta classificação, analise as afirmações a seguir.

I – o algoritmo não-preemptivo escolhe um processo e o deixa executar até que seja bloqueado ou deixe a CPU.

II - o algoritmo preemptivo escolhe um processo e o deixa executar até um tempo pré-determinado.

III – caso não exista relógio, o algoritmo preemptivo é a única alternativa a ser escolhida.

IV – o algoritmo não-preemptivo é de fácil implementação, enquanto o preemptivo possui uma implementação mais complexa.

Assinale a alternativa correta:

- a) I, II e III estão corretas.
- b) I, III e IV estão corretas.
- c) II, III e IV estão corretas.
- d) I e II estão corretas.
- e) III e IV estão corretas.

2.



Um Escalonador de Processos é um subsistema do Sistema Operacional responsável por decidir o momento em que cada processo obterá a CPU. É utilizado algoritmos de escalonamento que estabelecem a lógica de tal decisão. Nesse momento de decidir qual escalonador será utilizado no sistema operacional, cabe avaliar o cenário que o sistema será utilizado.

NOVATO, D. *Sistemas Operacionais - O que é escalonamento de processos?* Oficina da Net, 22 maio 2014. Disponível em: <<https://www.oficinadanet.com.br/post/12781-sistemas-operacionais-o-que-e-escalonamento-de-processos>>. Acesso em: 16 jul. 2018.

Analise as asserções:

Durante a escolha do escalonador sobre qual processo executar, é necessário ter cuidado com algumas variáveis (como tempo de uso da CPU de cada processo, redução do tempo de resposta, todos os processos devem utilizar a CPU, entre outros).

PORQUE

Os processos são únicos e imprevisíveis, sendo necessário equilibrar todas estas variáveis.

Com relação às asserções acima, assinale a alternativa correta.

- a) As asserções I e II são proposições verdadeiras, e a II é uma justificativa da I.
- b) As asserções I e II são proposições verdadeiras, mas a II não é uma justificativa da I.
- c) A asserção I é uma proposição verdadeira, e a II é uma proposição falsa.
- d) A asserção I é uma proposição falsa, e a II é uma proposição verdadeira.
- e) As asserções I e II são proposições falsas.

3. Com relação aos algoritmos de escalonamento de processos, analise as questões a seguir e associe as colunas de acordo com a característica de cada algoritmo:

1 - FIFO	() a cada processo é atribuído um intervalo de tempo no qual ele é permitido executar.
2 - Job mais curto primeiro	() cada processo possui uma prioridade e o processo com maior prioridade é executado primeiro.
3 - Escalonamento por alternância circular	() quando um processo termina o seu intervalo de tempo, ele é colocado no final da fila de execução.
4 - Escalonamento por prioridades	() o tempo de execução de todos os processos são previamente conhecidos.
	() o primeiro processo que solicita a CPU é o primeiro a ser alocado.

Assinale a alternativa que possui a ordem correta da associação das duas colunas:

- a) 3, 4, 3, 1, 2.
- b) 2, 1, 3, 4, 3.
- c) 4, 3, 2, 3, 1.
- d) 1, 3, 4, 2, 4.
- e) 3, 4, 3, 2, 1.

Referências

- AMOROSO, D. **O que são processos de um sistema operacional e por que é importante saber.** TecMundo, [s.l.], 4 dez. 2009. Disponível em: <<https://www.tecmundo.com.br/memoria/3197-o-que-sao-processos-de-um-sistema-operacional-e-por-que-e-importante-saber.htm>>. Acesso em: 11 jul. 2018.
- CANALTECH. **O que é Thread?** Disponível em: <<https://canaltech.com.br/produtos/o-que-e-thread/>>. Acesso em: 10 abr. 2018.
- DEITEL, H. M.; DEITEL, P. J.; CHOFFNES, D. R. **Sistemas Operacionais.** 3. ed. São Paulo: Prentice Hall, 2005.
- JONES, M. **Simulação do Linux Scheduler.** IBM, 12 abr. 2011. Disponível em: <<https://www.ibm.com/developerworks/br/library/l-linux-scheduler-simulator/index.html>>. Acesso em: 13 jul. 2018.
- LANHELLAS, R. **Trabalhando com Threads em Java.** DEV MEDIA, [s.l.], 2013. Disponível em: <<https://www.devmedia.com.br/trabalhando-com-threads-em-java/28780>>. Acesso em: 11 jul. 2018.
- MACHADO, F. B.; MAIA, L. P. **Arquitetura de Sistemas Operacionais.** 4. ed. Rio de Janeiro: LTC, 2007.
- NOVATO, D. **Sistemas Operacionais - O que é escalonamento de processos?** Oficina da Net, 22 maio 2014. Disponível em: <<https://www.oficinadanet.com.br/post/12781-sistemas-operacionais-o-que-e-escalonamento-de-processos>>. Acesso em: 16 jul. 2018.
- PATIL, S. S. **Limitations and capabilities of Dijkstra's semaphore primitives for coordination among processes.** USA: MIT, 1971.
- PEREIRA, J. W. **Semáforos, Kernel, Memória Compartilhada e Cia.** Dicas-L, [s.l.], 21 set. 2005. Disponível em: <http://www.dicas-l.com.br/arquivo/semaforos_kernel_memoria_compartilhada_e_cia.php#.WtTCJH9ryUk>. Acesso em: 12 jul. 2018.
- ROCHA, R. B. Desenvolvendo aplicações concorrentes estáveis e escaláveis. **Revista Modelo Java Magazine,** [s.l.], 2009. Disponível em: <<http://summa.com.br/wp-content/uploads/2009/06/programacaoconcorrente.pdf>>. Acesso em: 12 jul. 2018.
- ROJAS, A. **Semáforos.** 8 out. 2014. Disponível em: <<https://youtu.be/8YTV7cMyOSU>>. Acesso em: 12 jul. 2018.
- SANCHES, R. O. **Hierarquia de Processos no Unix e Windows.** DEV MEDIA, [s.l.], 2012. Disponível em: <<https://www.devmedia.com.br/hierarquia-de-processos-no-unix-e-windows/24739>>. Acesso em: 11 jul. 2018.
- TANENBAUM, A. S. **Sistemas Operacionais Modernos.** 2. ed. São Paulo: Pearson, 2003.
- TANENBAUM, A. S.; WOODHULL, A. S. **Sistemas operacionais: projeto e implementação.** 3. ed. Porto Alegre, Bookman, 2008.

Sistema de arquivos 2

Convite ao estudo

Caro aluno, seja bem-vindo!

Nesta unidade, veremos como os arquivos são usados pelo usuário, suas características, nomeação e estrutura, além de apresentarmos os tipos suportados pelos sistemas operacionais. Estudaremos também como é realizado o acesso, os atributos dos arquivos e as operações realizadas nos mesmo, como se dá a hierarquia de diretórios e como são realizados o acesso e as operações. Por fim, abordaremos a implementação de arquivos e diretórios, o gerenciamento de espaço em disco e como se dá a segurança e a confiabilidade do sistema de arquivos. Os arquivos armazenam informações em disco e, posteriormente, permitem que essas informações sejam recuperadas. Um sistema de arquivos é um conjunto organizado de arquivos e diretórios. No Windows e no Linux, existe um diretório raiz em que os diretórios e arquivos ficam vinculados. Por exemplo, dentro do diretório "Computador" do Windows existe uma hierarquia de diretórios vinculados a ele, tais como os diretórios C:, D:, E:, dentre outros que possam existir. No Linux, o diretório raiz (/) contém os seguintes diretórios: /etc (configurações do sistema), /home (arquivos pessoais), /boot (arquivos de inicialização do sistema), entre outros. Nesse contexto, você é um engenheiro da computação que foi contratado pela empresa XYZ Ltda., que atua no ramo de seguros automotivos e residenciais, com filiais em todo o país, para organizar os dados e implementar o seu sistema de arquivos. Para isso, é necessário que você tenha domínio de como os arquivos são organizados logicamente, como o sistema operacional facilita a interação do usuário com os arquivos, os tipos de arquivos permitidos, as operações que podem ser realizadas com arquivos e as estruturas de diretórios. A empresa XYZ Ltda. possui os sistemas operacionais Linux e

Windows e movimenta um grande volume de dados. Você deve fazer a escolha certa, de acordo com o uso dos sistemas da empresa, para que futuramente não ocorram lentidão ou outros problemas. Para iniciar a operação, será necessário realizar as seguintes etapas:

- I. Nomear os arquivos, definir a sua estrutura, apresentar os tipos de arquivos suportados, o acesso aos arquivos deve ser de forma aleatória, definir os atributos e operações a serem realizadas nos arquivos;
- II. Definir os diretórios para controle dos arquivos, especificar os nomes de caminhos e as operações com os diretórios; e
- III. Implementar e garantir a segurança do sistema de arquivos.

Como poderão ser definidos os nomes e a estrutura de arquivos para a empresa XYZ Ltda.? Como poderão ser definidas a hierarquia de diretórios e as operações a que cada usuário terá acesso? Como será implementada a segurança de arquivos? Ao final, você deverá entregar um anteprojeto para a avaliação dos gestores da empresa, apresentando todos os dados obtidos no trabalho realizado. Após o término desta unidade, você terá condições de entender a organização de arquivos e diretórios nos sistemas operacionais, as operações que podem ser realizadas e como podem ser implementados os sistemas de arquivos e diretórios.

Vamos juntos conhecer mais sobre os arquivos e suas propriedades?

Seção 3.1

Arquivos e sistemas de arquivos

Diálogo aberto

O sistema de arquivos é a maneira como as informações são armazenadas nos dispositivos (disco rígido, pen drives, dentre outros) e os arquivos são a maneira que o sistema operacional permite a leitura e gravação dos dados. Um arquivo é composto por um nome e por atributos (tamanho do arquivo, código de proteção de acesso, quem é o criador do arquivo, data de criação do arquivo) que são gerenciados pelo sistema operacional. Por exemplo, o arquivo TESTE.TXT, é um arquivo do tipo texto, com o nome TESTE e possui a extensão de .TXT (arquivo de texto). Nesta seção você conhecerá como os arquivos são usados pelo usuário, suas características, como se dá a nomeação e a estrutura de arquivos, os tipos de arquivos suportados pelos sistemas operacionais, como é realizado o acesso aos arquivos, os atributos dos arquivos e as operações realizadas em arquivos. Relembrando nosso contexto, você é um engenheiro da computação que foi contratado pela empresa XYZ Ltda. do ramo de seguros automotivos.

A empresa relata que tem perdido arquivos importantes, dado que a forma de armazenamento dos mesmos é arcaica, sendo dispostos em caixas em um depósito. Além do risco de degradação por umidade ou outras intempéries, tem-se a dificuldade da busca de documentos específicos, fatores estes que prejudicam as atividades da mesma. Os arquivos de cadastros de clientes de todas as filiais da empresa XYZ Ltda. estão armazenados em caixas, num depósito na matriz, localizada em São Paulo. As caixas estão separadas por cidades em que a empresa possui filiais e por nomes, como pré-vendas, vendas e boletos. Por exemplo: Belo Horizonte – Vendas, Rio de Janeiro – Boletos, Salvador – Pré-vendas. Uma parte desse cadastro foi armazenada em CDs-ROM e outra parte em HD.

Desta forma, você foi encarregado de organizar e implementar um sistema de arquivos. Portanto, surgem os seguintes

questionamentos: quais levantamentos devem ser feitos antes de iniciar os trabalhos com os arquivos da empresa XYZ Ltda.? Qual sistema de arquivos será implementado nos sistemas operacionais Linux e Windows? Como serão definidos os nomes, atributos, o acesso e as operações dos arquivos da empresa XYZ Ltda.? Quais são operações a serem realizadas com os arquivos? Para exemplificar os conceitos e justificar a implementação do sistema aos executivos da empresa, os gestores solicitaram que utilize os documentos referentes ao “Levantamento de Vendas Mensais”, que contêm informações de todas as vendas realizadas no mês, por setor. Lembre-se que estas informações auxiliarão na elaboração do anteprojeto de um sistema de proteção de arquivos.

Para que você consiga responder a esses e outros questionamentos sobre os arquivos, nesta seção vamos conhecer mais sobre eles e os conteúdos pertinentes a este tema.

Bons estudos!

Não pode faltar

As aplicações armazenam e recuperam informações durante seu processamento, e um processo deve ser capaz de ler e gravar grandes volumes de informações em disco rígido, pen drives, entre outros, além de dividir as informações com outros processos. Segundo Tanenbaum (2003), existem três requisitos essenciais para o armazenamento da informação a longo prazo:

- Deve ser possível armazenar um grande volume de informações (a exemplo dos sistemas bancários, companhias aéreas, entre outros).
- A informação deve sobreviver ao término do processo que a usa (armazenamento em banco de dados).
- Múltiplos processos têm que ser capazes de acessar a informação concorrentemente (a informação deve ser independente de qualquer processo).

O sistema operacional organiza e estrutura essas informações por meio de arquivos.

De acordo com Tanenbaum (2003), um arquivo pode ser definido como um mecanismo de abstração, oferecendo meios de

armazenamento de dados e permitindo que esses dados sejam lidos posteriormente. Segundo Machado e Maia (2007), o responsável por gerenciar os arquivos no sistema operacional é o sistema de arquivos, que é a parte visível de um sistema operacional, uma vez que os usuários manipulam constantemente os arquivos. Assim, é importante que o sistema de arquivos possua uma interface amigável e de fácil implementação. Cada sistema operacional possui seu próprio sistema de arquivos.

Nomeação de arquivos

Quando um processo cria um arquivo, atribui-se um nome a tal arquivo, e quando o processo é encerrado, o arquivo continua existindo e outros processos tem acesso a ele, buscando-o pelo seu nome (TANENBAUM, 2003). As regras de nomeação de arquivos variam de acordo com o sistema operacional. Os sistemas operacionais atuais aceitam cadeias de caracteres com entre uma e oito letras como nomes válidos de arquivos. Caracteres especiais e nomes com tamanhos de até 255 caracteres são aceitos em alguns sistemas de arquivos, por exemplo o Unix. O sistema de arquivos do Unix faz distinção entre letras maiúsculas e minúsculas (*case sensitive*), já o sistema de arquivos do Windows não faz esta distinção. A identificação do arquivo é composta por duas partes, separadas por um ponto, sendo que a parte após o ponto é denominada extensão do arquivo e identifica o conteúdo do arquivo (MACHADO; MAIA, 2007). O Quadro 3.1 apresenta algumas extensões de arquivos.

Quadro 3.1 | Algumas extensões de arquivos

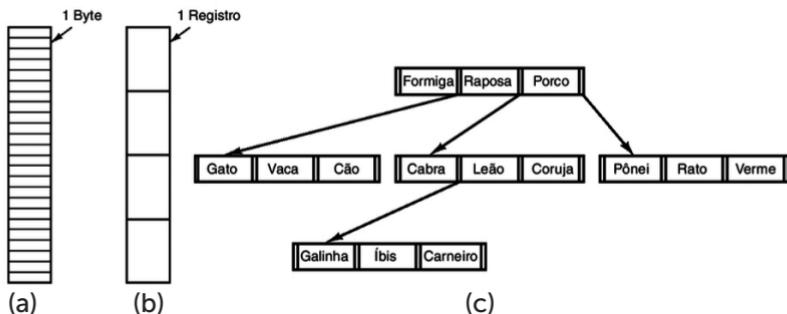
Extensão	Significado
Arquivo.txt	Arquivo de textos
Arquivo.zip	Arquivo Comprimido
Arquivo.jpg	Imagem codificada com o padrão JPEG
Arquivo.c	Programa fonte em C
Arquivo.bak	Arquivo de cópia de segurança
Arquivo.pdf	Arquivo no formato portátil de documentos
Arquivo.mp3	Música codificada de áudio MPEG

Fonte: adaptado de Tanenbaum (2003, p. 59).

Estrutura dos arquivos

Segundo Tanenbaum (2003), os arquivos podem ser estruturados de várias formas, sendo que as três mais comuns são: sequência estruturada de bytes, sequência de registro de comprimento fixo e árvore de registros. A Figura 3.1 apresenta a estrutura dos arquivos.

Figura 3.1 | Estruturas de arquivos



Fonte: Tanenbaum (2003, p. 288).

- **Sequência estruturada de bytes:** na estrutura da Figura 3.1 (a), o sistema operacional não sabe qual é o conteúdo do arquivo, e tudo o que ele vê são bytes. A sequência estruturada de bytes oferece flexibilidade, uma vez que os programas de usuário podem dar o nome que quiserem aos seus arquivos e inserir o conteúdo que desejarem.
- **Sequência de registro de comprimento fixo:** na estrutura da Figura 3.1 (b), um arquivo é uma sequência de registros de tamanho fixo, cada um com uma estrutura interna. O objetivo é que a operação de leitura retorne um registro e a operação de escrita sobreponha ou anexe um registro.
- **Árvore de registros:** na estrutura da Figura 3.1 (c), um arquivo é formado por uma árvore de registros, não necessariamente do mesmo tamanho, cada um contendo um campo-chave em uma posição fixa do registro. A árvore é ordenada pelo campo-chave, para que se busque mais rapidamente por uma chave específica. Novos registros podem ser inseridos no arquivo, e é o sistema operacional que decide onde colocá-los. Esse tipo de arquivo é amplamente aplicado em computadores de grande porte, usados ainda para alguns processamentos de dados comerciais.

Tipos de arquivos

Diferentes tipos de arquivos são suportados pelos sistemas operacionais, tais como arquivos regulares, diretórios, arquivos especiais de caracteres e especiais de blocos.

Tanembaum (2003) afirma que os arquivos regulares contêm informações do usuário, e são classificados em:

- **Arquivos ASCII:** são formados por linhas de texto. A grande vantagem dos arquivos ASCII é que eles podem ser mostrados e impressos como são e podem ser editados com qualquer editor de textos, além de facilitar a conexão entre a saída de um programa e a entrada de um outro.
- **Arquivos binários:** possuem alguma estrutura interna, conhecida pelos programas que os usam. Os sistemas operacionais devem reconhecer pelo menos um tipo de arquivo: seu próprio arquivo executável.

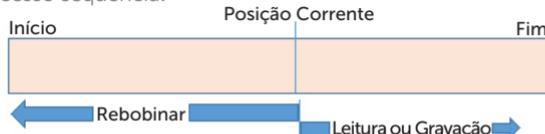
Os arquivos de diretórios mantêm a estrutura dos sistemas de arquivos. Já os arquivos especiais de caracteres são relacionados à entrada/saída e usados para modelar dispositivos de E/S, como terminais, impressoras e rede. Os arquivos especiais de blocos são usados para modelar discos.

Acesso aos arquivos

Os arquivos podem recuperar informações de diferentes formas, de acordo com sua organização. Segundo Machado e Maia (2007), os primeiros sistemas operacionais armazenavam os arquivos em fitas magnéticas e seu acesso era realizado de forma sequencial. No acesso sequencial, os arquivos são lidos sequencialmente, a partir do início do arquivo, sempre na ordem em que os registros foram gravados.

A Figura 3.2 apresenta um modelo de acesso sequencial. A leitura do arquivo era realizada avançando o ponteiro do arquivo para o próximo registro, e a gravação era realizada sempre no final do arquivo.

Figura 3.2 | Acesso sequencial



Fonte: Galvin et al (2013, p. 431).

Com a criação dos discos magnéticos, surgiu o acesso direto, cuja leitura/gravação é feita na sua posição por meio do número do registro. Não existem restrições em relação à ordem em que os registros são lidos ou gravados, sendo sempre necessário especificar o número do registro. O acesso direto somente é possível quando os registros do arquivo são de tamanho fixo, sendo que o mesmo pode ser combinado com o acesso sequencial, sendo possível acessar qualquer registro de um arquivo e, assim, acessar os demais.

Um método mais sofisticado é o acesso indexado ou por chave. O arquivo deve possuir uma área de índice em que existam ponteiros para os diversos registros. A aplicação que desejar acessar o registro deverá especificar uma chave para que o sistema pesquise, na área de índice, o ponteiro correspondente, acessando o arquivo diretamente.

Atributos dos arquivos

Segundo Machado e Maia (2007), atributos são informações de controle de cada arquivo, que variam de acordo com o sistema de arquivos, mas o tamanho do arquivo, a proteção, a identificação do criador e a data de criação estão presentes em quase todos os sistemas. Alguns atributos especificados na criação de arquivos não podem ser modificados, como a organização e data/hora da criação. Outros atributos podem ser alterados pelo sistema operacional, como o tamanho e a data/hora do último backup realizado. Existem, ainda, atributos que podem ser modificados pelo usuário, como proteção do arquivo, tamanho máximo e senha de acesso.

Observe no Quadro 3.2 alguns atributos dos sistemas de arquivos:

Quadro 3.2 | Atributos e características de arquivos

Atributo	Característica
Proteção	Quem pode ter acesso ao arquivo e de que forma.
Senha	Para ter acesso ao arquivo.
Criador	ID do criador do arquivo.
Proprietário	Atual proprietário.
Tamanho do registro	Número de bytes em um registro.
Tamanho Atual	Número de bytes no arquivo.
Momento da Criação	Data/horário da criação do arquivo.

Fonte: adaptado de Tanenbaum (2003, p. 291).



Vimos nesta seção que os atributos são informações de controle de cada arquivo e variam de acordo com o sistema de arquivos. Além dos atributos apresentados nesta seção, temos os seguintes atributos: "somente leitura", que indica que o arquivo não deve ser modificado ou deletado, e o atributo "oculto", usado quando um arquivo não deve ser visto por todos.



Pesquise mais

Para garantir a segurança dos seus arquivos na nuvem, é necessário seguir alguns passos importantes: criar uma senha forte, não utilizar redes WiFi públicas para acesso aos arquivos, reforçar a autenticação das contas, manter seus programas e aplicativos atualizados e **não permitir o backup automático dos seus dados**. Para saber mais sobre esse assunto, consulte o artigo:

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: 8 maio 2018.

Operações com arquivos

Segundo Tanenbaum (2003), sistemas de arquivos diferentes oferecem operações distintas para **armazenar e recuperar informações**. As principais operações com arquivos são:

- Create: cria um arquivo sem dados e define alguns atributos.
- Delete: deleta um arquivo quando ele não é mais necessário, liberando espaço de memória.
- Open: um processo deve abrir um arquivo antes de usá-lo, permitindo que o sistema busque e coloque na memória principal os atributos e a lista de endereços do disco, agilizando o acesso as informações.
- Close: fecha um arquivo e libera espaço na memória.
- Read: leitura dos dados do arquivo.
- Write: grava os dados no arquivo.

Sistemas de Arquivos – Linux

Segundo Machado e Maia (2007), o sistema de arquivos no Linux é baseado numa estrutura de diretórios hierárquica, sendo o diretório raiz (root) representado por uma (/). Veremos essa estrutura hierárquica com maior profundidade na Seção 3.2. Os sistemas de arquivos suportados pelas distribuições do Linux são (SILVA, 2018):

- EXT2: Second Extended File System ou EXT2, estava presente nas primeiras versões do Linux, mas era pouco eficiente.
- EXT3: o sistema EXT3 é uma versão do EXT2, e se tornou popular, estável e robusto por possuir um suporte eficiente, conhecido como *journaling*. *Journaling* é um recurso que gerencia a manipulação de arquivos. *Journals* (ou logs) registram nos discos as alterações realizadas nos arquivos antes de serem gravadas, e, por meio desse log, caso aconteça algum erro, como queda de energia ou o travamento inesperado, as operações podem ser recuperadas.
- EXT4: é a versão mais recente do EXT, apresentando um bom desempenho com a manipulação de arquivos grandes, porém ainda instável.
- ReiserFS: foi criado recentemente, possui suporte *journaling* e uma boa performance, além de ser suportado pelas versões do Linux.

Os sistemas EXT3 e ReiserFS são os mais utilizados pelas distribuições do Linux. Além destes sistemas existem outros suportados pelo Linux, como SWAP, XFS, entre outros.



Exemplificando

O Linux oferece suporte a vários sistemas de arquivos antigos, como MINIX, MS-DOS e EXT2. Também suporta os novos sistemas de arquivos com registro de mudança, como EXT3, JFS e ReiserFS, além do suporte a sistemas de arquivos criptográficos, como CFS, e a sistemas de arquivos virtuais, como /proc.

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.

Sistemas de Arquivos – Windows

Os sistemas de arquivos do Windows são o FAT16, FAT32 e NTFS. O nome FAT significa *File Allocation Table*.

Segundo Tanenbaum (2003), o FAT16, usado no MS-DOS, utiliza endereços de memória de 16 bits e não suporta discos maiores que 2GB, gerando desperdício de espaço. Já o FAT32 utiliza endereços de memória de 32 bits, mas não reconhece arquivos maiores que 4GB.

O NTFS (*New Technology File System*) é o atual sistema de arquivos do Windows. O NTFS é seguro, possui suporte a discos maiores, além de implementar permissões de acessos aos arquivos. A partir do Windows 8, foi criado o sistema de arquivos ReFS (*Resilient File System*), com o objetivo de verificar e corrigir automaticamente os dados e manipular um grande volume de dados.



Refletia

Qual sistema operacional é melhor ao trabalhar com arquivos? O Windows, com o padrão NTFS, ou o Linux, com o padrão Ext4?

Sem medo de errar

Agora que você já conheceu como os arquivos são usados pelo usuário, suas características, como se dá a nomeação e a estrutura de arquivos, os tipos de arquivos suportados pelos sistemas operacionais, como é realizado o acesso aos arquivos, os atributos dos arquivos e as operações realizadas em arquivos, vamos voltar ao nosso contexto. Na empresa XYZ Ltda., os arquivos de cadastros de clientes de todas as filiais estão armazenados em caixas, num depósito na matriz, localizada em São Paulo. Uma parte deste cadastro foi armazenada em CDs-ROM e outra parte em HD. Quais levantamentos devem ser feitos antes de iniciar os trabalhos com os arquivos da empresa XYZ Ltda.? Qual sistema de arquivos será implementado nos sistemas operacionais Linux e Windows? Como serão definidos os nomes, os atributos, o acesso e as operações dos arquivos da empresa XYZ Ltda.? Antes de definir o sistema de arquivos a ser usado, é necessário fazer um levantamento dos documentos que estão armazenados no depósito, verificando a estrutura

destes arquivos. Como os arquivos ficam guardados em caixas, é necessário antes migrar estes dados para um banco de dados que irão gerenciar melhor estas informações. Após isso, é necessário definir o sistema de arquivo a ser utilizado. Na empresa XYZ Ltda. são utilizados os sistemas operacionais Linux e Windows. Como sistema de arquivos do Windows, podemos usar o padrão adotado atualmente, o NTFS, que é seguro e manipula um grande volume de dados. Para o Linux, muitas distribuições, ao serem instaladas, já trazem o sistema de arquivos instalado. Na maioria das distribuições, o sistema que vem instalado é o EXT4, que não é a versão atual, é seguro, porém instável, e atenderá bem às necessidades da empresa XYZ Ltda. Como a empresa trabalha por filiais, podemos sugerir que os nomes dos arquivos comecem identificando-as, conforme apresentado no Quadro 3.3. Por exemplo, a empresa possui o documento “Levantamento de Vendas Mensais”, que contém informações de todas as vendas realizadas no mês por setor. Esse arquivo pode ser nomeado iniciando pela filial, o nome do setor, o tipo de vendas (diário, semanal, mensal, anual) e o mês/ano do arquivo. É necessário salientar que o Linux faz diferenciação de letras maiúsculas e minúsculas, e, durante a definição dos nomes, essa questão deve ser levada em consideração.

Quadro 3.3 | Sugestão de nomes de arquivos

Documento	Sugestão de nome
Levantamento de Vendas Mensais	BH_Vendas_Mensais_05/2018 RJ_Vendas_Mensais_01/2018
Clientes de Pré-vendas anuais	SP_Pre_Vendas_2018

Fonte: elaborado pela autora.

Quanto aos atributos, será necessário definir:

- Quem terá acesso a sistema - por exemplo, o setor de Pré-vendas tem acesso somente aos arquivos do seu setor. Já o setor de vendas terá acesso aos seus arquivos e aos arquivos do setor de Pré-vendas.
- Inserção de senha para acesso a todos os arquivos.
- Identificação de quem criou o arquivo.
- Tamanho do arquivo.

- Data da criação do arquivo.
- Data da modificação do arquivo.

O acesso aos arquivos pode ser do tipo indexado, para acessar de forma rápida o registro desejado no arquivo. É necessário definir alguns parâmetros para que as operações sejam realizadas.

Quadro 3.4 | Sugestão de operações a serem realizadas.

Operação	Permissão
Criar	Todos os usuários podem criar seus próprios arquivos.
Deletar	Somente quem criou terá a permissão para deletar o arquivo.
Abrir	Somente usuários autorizados pelo setor poderão abrir o arquivo.
Ler	Todos os usuários poderão abrir o arquivo.
Gravar	Somente quem criou terá a permissão para gravar dados no arquivo.

Fonte: elaborado pela autora.

Avançando na prática

Métodos de acesso aos arquivos

Descrição da situação-problema

Durante a organização dos arquivos da empresa XYZ Ltda., o gerente de vendas questiona de qual maneira os dados dos arquivos serão acessados, uma vez que existem várias filiais e o volume de dados movimentados é grande. Existem os métodos de acesso sequencial, direto e indexado (ou por chave). O que levar em conta na escolha do método de acesso ao arquivo? Qual é o melhor método de acesso para resolver essa questão?

Resolução da situação-problema

A empresa XYZ Ltda. possui um volume elevado de dados sendo necessário escolher um método robusto de acesso aos arquivos. O método sequencial não permite o acesso direto ao arquivo, o que dificulta o acesso a informação. O método direto permite a leitura da posição do registro, porém, existe uma limitação: o arquivo deve ser definido com registros de tamanho fixo. O acesso indexado, ou por chave, é o indicado

para a empresa XYZ Ltda., por ser o método mais sofisticado. Quando uma aplicação deseja ter acesso ao registro, é necessário informar a chave do registro. É feita uma pesquisa no sistema de arquivos por meio da chave para se ter acesso direto ao registro do arquivo.

Faça valer a pena

1.



Na prática, um sistema de arquivo (*file system*, do inglês) é um conjunto de estruturas lógicas, ou seja, feitas diretamente via software, que permite ao sistema operacional ter acesso e controlar os dados gravados no disco. No universo Windows, o número de sistemas de arquivos é mais limitado. Na época do Windows 95, a Microsoft usava o sistema de arquivos FAT16. Devido às suas limitações, foi substituído pelo FAT32 que, anos depois, foi substituído pelo NTFS. Este é usado até hoje e se estabeleceu devido à flexibilidade. Já no amplo universo Linux, onde é possível encontrar uma enorme variedade de distribuições, o leque de sistemas de arquivos é bem maior. Os mais usados são o EXT3 e o EXT4, bem como o ReiserFS.

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: 7 mai. 2018.

Sobre os sistemas de arquivo Linux e Windows, marque a alternativa correta.

- a) O Windows consegue reconhecer outros sistemas de arquivos além do NTFS.
- b) No Linux, o sistema de arquivos é estruturado em formato de árvore.
- c) O sistema de arquivo do Windows é case sensitive, ou seja, diferencia letras minúsculas e maiúsculas.
- d) O Linux consegue suportar uma quantidade limitada de sistemas de arquivos.
- e) O sistema FAT32 suporta diversos tipos de usuários do Windows.

2.

Nos sistemas *Linux*, tudo o que pode ser manipulado pelo sistema operacional é representado sob a forma de arquivo, incluindo diretórios, dispositivos e processos. Se tudo é arquivo, como o sistema operacional faz a distinção entre arquivo regular, diretório, dispositivo, processo?

PAULO, D. R. **Fundamentos do sistema Linux - arquivos e diretórios**. 28 out. 2005. Disponível em: <<https://www.viaolinux.com.br/artigo/Fundamentos-do-sistema-Linux-arquivos-e-diretorios/>>. Acesso em: 09 mai. 2018.

A seguir, analise as asserções:

Todo arquivo tem cabeçalho, que contém informações, como tipo, tamanho, data de acesso e de modificação.

Porque

Por meio do cabeçalho, o *GNU/Linux* sabe quando se trata de arquivo regular, diretório, dispositivo, e processo.

Com relação às duas asserções assinale a alternativa correta:

- a) As asserções I e II são proposições verdadeiras, e a II é uma justificativa da I.
- b) As asserções I e II são proposições verdadeiras, mas a II não é uma justificativa da I.
- c) A asserção I é uma proposição verdadeira, e a II é uma proposição falsa.
- d) A asserção I é uma proposição falsa, e a II é uma proposição verdadeira.
- e) As asserções I e II são proposições falsas.

3.

Segundo a Microsoft, o ReFS foi criado para atender as novas necessidades de armazenamento do Windows, sendo elas a capacidade de lidar com grandes volumes, a resistência ao corrompimento de arquivos e o compartilhamento de conjuntos de armazenamento entre máquinas diferentes acessando diretamente o arquivo.

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 mai. 2018.

Considerando as características do sistema de arquivos ReFs da Microsoft, escolha a opção correta:

- a) O sistema de arquivos ReFS foi desenvolvido do zero, não se baseando nos sistemas de arquivos existentes.
- b) A resiliência do ReFS está diretamente ligada a verificar se o arquivo está corrompido e recuperá-lo.
- c) O sistema de arquivos ReFS não é compatível com o NTFS, e caso a versão do Windows seja diferente, é necessário instalar o Windows 8 para que ele funcione.
- d) O sistema de arquivos ReFS é mais moderno, porém não suporta volumes grandes de dados nos arquivos.
- e) O ReFS, por suportar a otimização em tempo real de arquivos, prejudica a performance do sistema tornando-o lento.

Seção 3.2

Sistemas de diretórios

Diálogo aberto

Caro aluno, o sistema de arquivos contém diretórios e pastas que gerenciam os arquivos. Normalmente, os atuais sistemas de arquivos permitem que os usuários criem vários diretórios e subdiretórios, apresentando uma estrutura organizada dos arquivos. Por exemplo, o diretório "/" do Linux representa a raiz do sistema de arquivos e os demais diretórios estão abaixo dele, como arquivos de sistemas, arquivos de usuário, arquivos de aplicações, entre outros. No Windows, o diretório C:/ representa a raiz do sistema. Nesta seção você conhecerá como os diretórios são usados pelo usuário e suas propriedades, como funcionam os sistemas de diretório simples e hierárquico, como se dá a nomeação dos caminhos de diretórios e as operações realizadas em diretórios. Relembrando nosso contexto, a empresa XYZ Ltda. do ramo de seguros automotivos e residenciais, com filiais em todo o país, deseja organizar os dados e implementar o seu sistema de arquivos e, para isso, lhe contratou, já que você é um engenheiro da computação. Na empresa XYZ Ltda., os arquivos dos setores de pré-vendas e vendas são acessados durante todo o dia pelos funcionários dessas áreas para traçarem planos de ação das equipes de vendas. Porém, não existem diretórios específicos definidos para cada setor armazenar seus arquivos e diretórios do próprio sistema operacional. Desta forma, todos os usuários da empresa podem acessá-los, o que compromete a segurança dos dados, uma vez que, caso algum usuário acesse um diretório indevidamente, ele poderá alterá-lo e/ou exclui-lo. Na empresa XYZ Ltda., já está definido o sistema de arquivos a ser usado tanto no Linux (EXT4) como no Windows (NTFS), além dos nomes, atributos, o acesso e as operações a serem realizadas nos arquivos. Como você fará a organização de diretórios? Como serão definidas as permissões de diretórios para os setores de vendas e pré-vendas? Todas essas questões darão base para você elaborar o anteprojeto de um sistema de proteção de arquivos. Para que você consiga

responder a esses e outros questionamentos sobre os diretórios, nesta seção vamos conhecer mais sobre eles e os conteúdos pertinentes a este tema. Vamos lá?

Bons estudos!

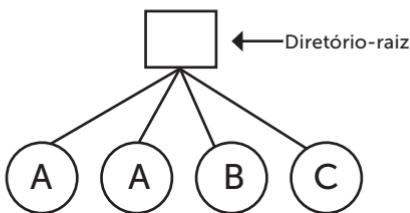
Não pode faltar

Segundo Machado e Maia (2007), a estrutura de diretórios é a organização lógica dos arquivos, em disco. O diretório é uma estrutura de dados que possuem entradas associadas aos arquivos, sendo que cada entrada possui os atributos de localização do arquivo, nome, dono, organização, dentre outros. Em alguns sistemas operacionais, como o Linux, todos os diretórios e pastas são considerados arquivos.

Diretórios Simples

Uma forma simples de sistema de diretório é manter um diretório contendo todos os arquivos, chamado de diretório raiz (TANENBAUM, 2003). Esse formato era utilizado nos primeiros computadores pessoais por haver apenas um usuário e, como vantagem, apresenta a simplicidade e a capacidade de encontrar rapidamente os arquivos. A Figura 3.3 apresenta um exemplo de sistema de diretórios com quatro arquivos de três usuários diferentes, A, B e C.

Figura 3.3 | Sistema de Diretórios Simples.



Fonte: Tanenbaum (2003, p. 296).

Sistemas de Diretório Hierárquico

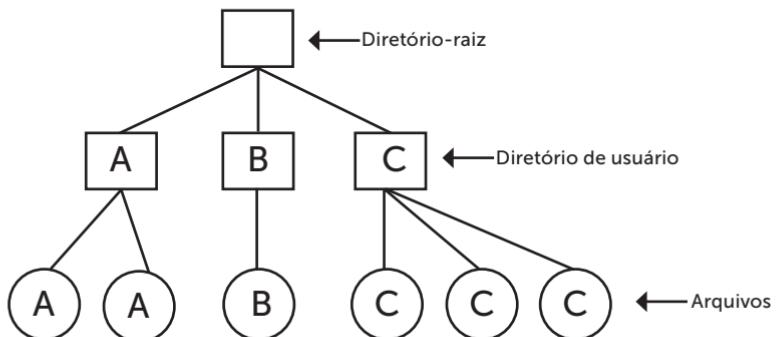
Segundo Tanenbaum (2003), o problema existente em um sistema com vários usuários é que pode ocorrer de diferentes

usuários nomearem seus arquivos com os mesmos nomes. Por isso, esse esquema não é mais utilizado em sistemas multiusuário.

Para evitar conflitos causados por diferentes usuários escolhendo o mesmo nome, é necessário dar um diretório privado para cada um. Assim, os nomes escolhidos por um usuário não interfeririam nos nomes escolhidos por outro, podendo existir arquivos com o mesmo nome em dois ou mais diretórios. Esse esquema é chamado de sistema de diretórios em dois níveis.

A Figura 3.4 apresenta um sistema de diretórios em dois níveis, em que cada usuário possui seu diretório. Assim, o sistema, conhecendo o usuário, sabe em qual diretório buscar o arquivo.

Figura 3.4 | Sistema de Diretórios em dois níveis



Fonte: Tanenbaum (2003, p. 296).

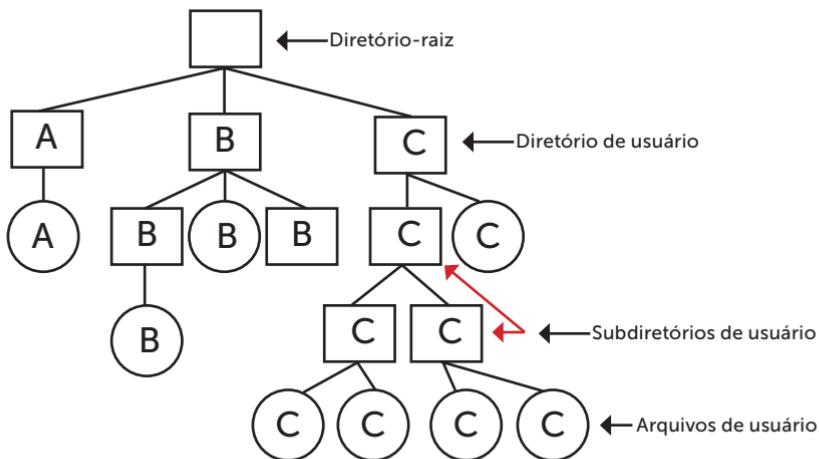
O esquema de diretórios em dois níveis elimina o conflito de nomes entre usuários, porém, não é adequado para usuários com um grande número de arquivos. Por exemplo, um programador de sistemas pode ter:

- Um conjunto de arquivos de códigos, que juntos formam um sistema.
- Outro conjunto de arquivos referentes a projetos já entregues.
- Outro conjunto de arquivos tratando de livros específicos da área, e assim por diante.

Dessa forma, faz-se necessário agrupar esses arquivos segundo as necessidades do usuário. Assim, é necessária uma hierarquia geral ou árvore de diretórios (*folder tree*) para permitir que os usuários tenham tantos diretórios quanto precisarem para agrupar

os seus arquivos. A Figura 3.5 apresenta a estrutura de diretórios hierárquicos. Nela, os diretórios A, B e C pertencem a usuários diferentes, e dois desses usuários criaram subdiretórios para organizar os seus arquivos. Quase todos os sistemas de arquivos modernos são organizados com essa estrutura, por proporcionar uma melhor organização dos trabalhos.

Figura 3.5 | Sistema de Diretórios Hierárquicos



Fonte: Tanenbaum (2003, p. 297).



Pesquise mais

Para saber mais sobre o gerenciamento de arquivos e diretórios do Linux acesse:

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.

Nomes de Caminhos dos Diretórios

Segundo Tanenbaum (2003), quando o sistema de arquivos é organizado por meio de uma árvore de diretórios, é necessário definir uma forma de especificar o nome dos arquivos. Para isso, são usados dois métodos: nome de caminho absoluto e nome de caminho relativo.

O método nome de caminho absoluto é formado pelo caminho entre o diretório-raiz e o arquivo. Os nomes de caminhos absolutos sempre iniciam no diretório-raiz e são únicos. Como exemplo, o caminho /usuário/meus_documentos/atividades.txt significa que dentro do diretório-raiz há um diretório chamado "usuário", dentro do diretório "usuário" existe um subdiretório chamado "meus_documentos" e dentro do subdiretório "meus_documentos" existe um arquivo chamado "atividades.txt". No Windows, os componentes do caminho são separados por \. No Unix, são separados por /.

O método nome de caminho relativo é usado em conjunto com o conceito de diretório atual ou diretório de trabalho. Desta forma, é possível que o usuário designe um diretório como o diretório atual de trabalho, em que todos os nomes de caminhos não começem no diretório-raiz.

Como exemplo, se o diretório atual for /usuário/meus_documentos/, o arquivo cujo caminho absoluto for /usuário/meus_documentos/atividades pode ser referenciado apenas como atividades.



Exemplificando

Analise o endereço do seguinte projeto de um sistema hospitalar aloocado num servidor Linux: /home/user/projetos/sistema_hospitalar. O caminho absoluto do endereço é /home/user/projetos/sistema_hospitalar e o caminho relativo do endereço é sistema_hospitalar, se estivermos no diretório /home/user/projetos.

Ainda, segundo Tanenbaum (2003), cada processo possui seu próprio diretório de trabalho, e caso ocorra alguma alteração, nenhum outro processo é afetado e nenhum vestígio da mudança é deixado no sistema de arquivos. Procedimentos de biblioteca (conjunto de subprogramas que auxiliam no desenvolvimento do software) raramente alteram o diretório de trabalho, pois se alterarem o diretório de trabalho e não retornarem aonde estavam quando finalizarem, o resto do programa poderá não funcionar.

Os sistemas operacionais que suportam um sistema de diretório hierárquico possuem duas entradas especiais em cada diretório: “.” (ponto), que se refere ao diretório atual, e “..” (ponto-ponto), referindo-se a seu pai.

Por exemplo, se um usuário estiver trabalhando no diretório /usuário/meus_documentos/ e utilizar o ponto-ponto, ele estará subindo na árvore de diretórios, ou seja, referenciando o diretório /usuário. Caso ele utilize o ponto, estará referenciando o próprio diretório (/usuário/meus_documentos/).

Operações com Diretórios

As chamadas aos sistemas que gerenciam as operações com diretórios variam de sistema para sistema. Segundo Tanenbaum (2003), as principais operações de manipulação de diretórios são:

- Create: cria um diretório. Um diretório é criado sem conteúdo, ou seja, vazio.
- Delete: apaga um diretório. Somente um diretório vazio pode ser apagado.
- OpenDir: lê um diretório. Antes de ser lido, um diretório precisa ser aberto.
- Closedir: após a leitura do diretório, o mesmo precisa ser fechado, liberando, assim, espaço em disco.
- Rename: permite a troca do nome de um diretório.



Exemplificando

Seguem as principais operações realizadas nos diretórios dos sistemas operacionais Linux e Windows.

Linux:

- Criar: mkdir EXEMPLO (cria o diretório EXEMPLO no diretório atual do usuário).
- Deletar: rmdir EXEMPLO (apaga o diretório EXEMPLO se ele estiver vazio).
- Abrir: cd EXEMPLO (abre o diretório EXEMPLO a partir do diretório atual do usuário).

- Fechar: closedir EXEMPLO (fecha o diretório EXEMPLO).
- Renomear: mv EXEMPLO TESTE (renomeia o diretório EXEMPLO para TESTE).

Windows

- Criar: clique com o botão direito do mouse no local que deseja criar um novo diretório e, em seguida, na opção "Novo -> Pasta".
- Deletar: selecione o diretório que deseja excluir e clique com o botão direito do mouse na opção "Excluir".
- Abrir: basta dar um duplo clique no diretório.
- Fechar: clique no X do diretório para fechá-lo.
- Renomear: selecione o diretório que deseja renomear e clique com o botão direito do mouse na opção "Renomear".



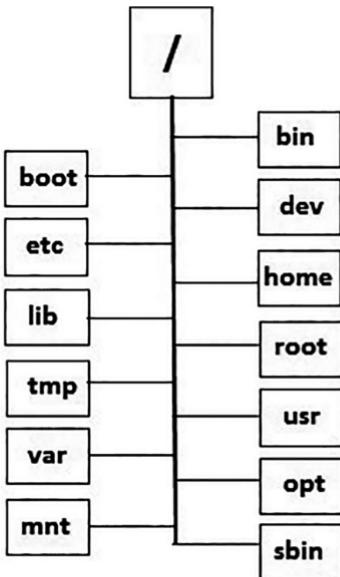
Assimile

Vimos nesta seção as principais operações realizadas com diretórios nos sistemas operacionais. Além dessas operações, podemos citar como exemplo a operação Link, que permite a um arquivo aparecer em mais de um diretório. Essa chamada ao sistema especifica um arquivo existente e um nome de arquivo, criando uma ligação do arquivo existente com o nome especificado pelo caminho (TANENBAUM, 2003).

Sistemas de Diretórios do Linux

No Linux, o sistema de diretórios é baseado numa estrutura hierárquica e os diretórios são implementados por meio de arquivos. As partições e discos fazem parte do diretório-raiz ou "/". Dentro do diretório-raiz existem diretórios que possuem funções distintas. A Figura 3.6 apresenta a estrutura de diretórios do Linux.

Figura 3.6 | Sistema de Diretórios do Linux.



Fonte: elaborada pela autora.

A seguir, apresentamos as definições de cada diretório que estão abaixo do diretório-raiz (JUNIOR, 2018):

- **/bin:** contém aplicativos e utilitários do sistema.
- **/boot:** contém todos os arquivos necessários para a inicialização do sistema e o gerenciador de boot.
- **/etc:** contém todos os arquivos de configuração do sistema.
- **/dev:** contém todos os dispositivos do sistema, tais como: modem, HD, floppy disk, CD-ROM, entre outros.
- **/home:** contém diretórios e arquivos de trabalho dos usuários comuns do sistema.
- **/lib:** contém as bibliotecas compartilhadas e módulos do kernel.
- **/root:** contém diretórios e arquivos de trabalho do superusuário (root).
- **/usr:** contém os programas de usuários, sistemas de janelas, jogos, entre outros.
- **/tmp:** contém os arquivos temporários utilizados por algumas aplicações.

- **/var:** contém os arquivos de dados variáveis, como log do sistema e diretórios de *spool*.
- **/mnt:** diretório reservado para montagem de dispositivos e sistemas de arquivos em partições e discos locais e de rede.
- **/opt:** diretório reservado para instalação de aplicações de terceiros, como OpenOffice, softwares pagos, entre outros.
- **/sbin:** contém ferramentas de configuração e manutenção do sistema.

Segundo Machado e Maia (2007), a localização de um arquivo dentro da estrutura de diretórios é indicada por um *pathname*, que pode ser dos tipos absoluto ou relativo. Um *pathname* absoluto indica a localização do arquivo por meio do caminho completo do diretório. Já um relativo é representado pelo nome parcial do diretório. Ainda segundo os mesmos autores, um sistema de arquivos proporciona um mecanismo de compartilhamento chamado link (ligação). Um link é a entrada de um diretório que referencia um arquivo em outro diretório. Um arquivo pode ter vários links. Com a utilização de links existe uma redução de espaço em disco, a possibilidade de compartilhamento entre diversos usuários, entre outros.

Todo arquivo ou diretório do Linux tem um dono (*user*) e pertence a um grupo (*group*). Para cada categoria de usuário existem três tipos de acesso: leitura (r), gravação (w) e execução (x). O Quadro 3.5 apresenta as permissões (privilegios) que os arquivos e diretórios possuem, de acordo com o tipo de acesso.

Quadro 3.5 | Permissões de arquivos e diretórios

Permissão	Arquivos	Diretórios
r	Leitura e cópia	Lista conteúdo
w	Gravação	Cria, elimina e renomeia
x	Execução	Acessa os arquivos

Fonte: elaborado pela autora.

Existe ainda a definição de quem pode acessar os diretórios: somente o dono, todos os usuários, somente usuários do mesmo grupo ou qualquer conjunto de usuários.



O comando `-rwxr--r--` significa:

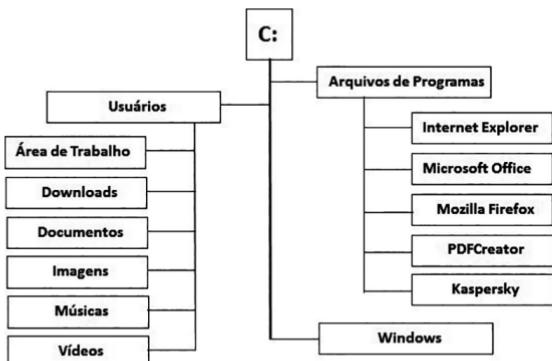
- ↓ Os três caracteres (r--) representam a permissão para outros usuários do sistema: somente leitura.
- ↓ Os três caracteres (r--) representam a permissão do grupo do usuário: somente leitura.
- ↓ Os três caracteres (rwx) representam a permissão do dono do arquivo: leitura, gravação e execução.
- ↓ O primeiro caractere informa se é um arquivo ou diretório.

Sistemas de Diretórios do Windows

Segundo Tanenbaum (2003), o sistema de arquivos NTFS é baseando numa estrutura hierárquica. No Windows, o sistema de arquivos está concentrado em pastas ou diretórios C:, D:, entre outras.

A Figura 3.7 apresenta a estrutura de diretórios do Windows. O diretório C: contém as pastas "Usuários", "Arquivos de Programas" e "Windows". A pasta "Usuários" contém as principais pastas utilizadas pelos usuários. A pasta "Arquivos de Programas" é onde estão concentrados todos os programas instalados no computador. A pasta "Windows" é a mais importante, por nela constar toda a estrutura do sistema operacional.

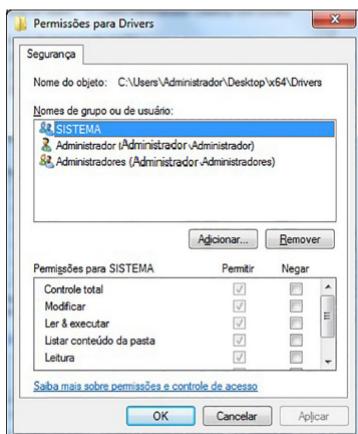
Figura 3.7 | Sistema de Diretórios do Windows.



Fonte: elaborada pela autora.

No Windows, o separador de nomes é representado por “\” (contra barra) e também existe o conceito de caminho absoluto e relativo, como no Linux. Além disso, é possível inserir permissões de acesso para cada usuário (administrador do sistema e usuário comum). Para isso, basta configurar as permissões na pasta desejada. A Figura 3.8 apresenta a pasta Drivers e as permissões que o usuário SISTEMA possui. Observe que o usuário SISTEMA possui as permissões de controle total, modificar, ler e executar, listar conteúdo de pastas, leitura, gravar e permissões especiais.

Figura 3.8 | Sistema de Diretórios do Windows



Fonte: captura de tela do Windows, elaborada pela autora.



Reflita

Na sua opinião, a hierarquia de diretórios do Linux é mais complexa do que a do Windows? Por quê?

Sem medo de errar

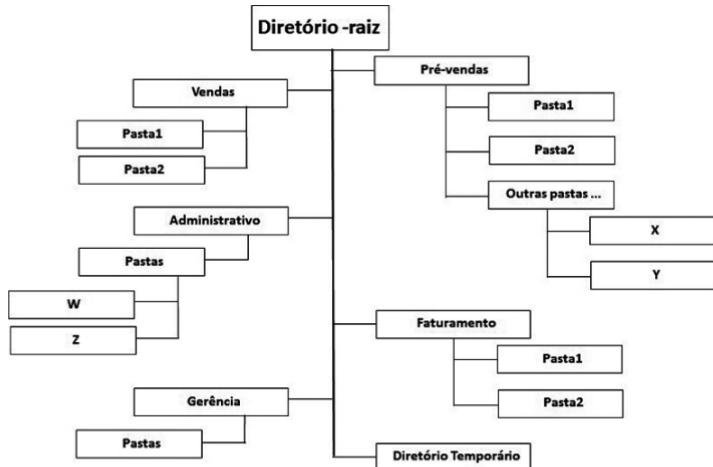
Agora que você já conheceu como os diretórios são usados pelo usuário, suas propriedades e as operações que podem ser realizadas, vamos voltar ao nosso contexto. Na empresa XYZ Ltda., os arquivos dos setores de pré-vendas e vendas são acessados durante todo o dia, porém, não existem diretórios específicos definidos para

cada setor armazenar seus arquivos e diretórios do próprio sistema operacional. Isso pode comprometer a segurança, pois caso um usuário acesse um diretório indevidamente, poderá alterá-lo e/ou excluí-lo. O sistema de arquivos já foi definido tanto no Linux (EXT4) como no Windows (NTFS), além dos nomes, os atributos, o acesso e as operações a serem realizadas nos arquivos. Como você fará a organização de diretórios? Como seriam definidas as permissões de diretórios para os setores de vendas e pré-vendas?

Um diretório fornece acesso aos arquivos e a outros diretórios vinculados a ele. A questão da falta de estrutura dos diretórios da empresa XYZ Ltda. pode ser em função da má implementação de diretórios realizada anteriormente. Assim, é necessário analisar a estrutura atual e reorganizar os diretórios e arquivos, implementando e relacionando cada entrada do diretório aos arquivos que estão armazenados nele. Um fator importantíssimo é que os diretórios do próprio sistema operacional estão acessíveis a todos os usuários. Isso não pode acontecer. Logo, os diretórios devem ser reestruturados para que somente os administradores do sistema tenham acesso a eles.

A empresa XYZ Ltda. possui os sistemas operacionais Linux e Windows, e a organização de diretórios para ambos pode ser representada pelo exemplo da Figura 3.9. A organização de diretórios pode ser definida pelo administrador do sistema ou pelo próprio usuário.

Figura 3.9 | Sistema de Diretórios do Windows



Fonte: elaborada pela autora.

Na Figura 3.9 é apresentada uma organização estruturada de diretórios. Dentro do diretório-raiz são encontrados os diretórios Vendas, Pré-vendas, Administrativo, Faturamento, Gerência e Diretório Temporário. Em cada diretório há subdiretórios criados pelos usuários.

Será necessário definir as permissões de acesso dos setores de vendas e pré-vendas para que cada setor consiga operar seus diretórios de forma eficiente. O setor de pré-venda terá acesso para gravar, ler e executar arquivos, ou seja, terá permissão total ao seu diretório de arquivos, porém, será permitido apenas ler os arquivos do diretório de vendas. Já o setor de vendas, terá acesso total ao seu diretório, podendo ler e executar dados do setor de pré-vendas. Uma vez definidas as permissões de acesso de cada setor, as operações permitidas para cada setor são:

Quadro 3.6 | Definição das permissões de diretórios para o setor de vendas e pré-vendas

Permissão	Ler	Gravar	Executar
Pré-vendas	X	X	X
Vendas	X	X	X
Acesso aos arquivos de outro setor	X	-	-

Fonte: elaborado pela autora.

Avançando na prática

Limite de arquivos em um diretório

Descrição da situação-problema

A empresa XYZ Ltda. possui um grande volume de dados, que são armazenados em diretórios e pastas. O diretor da área de pré-vendas identificou que o diretório de armazenamento das propostas de vendas mensais não suporta um armazenamento superior a 50000 arquivos. Além disso, o diretor relatou que ao gravar o arquivo “Conexão de Vendas” com o nome “con”, o sistema não permitiu a sua criação. Diante do exposto, os seguintes questionamentos são feitos a você: a quantidade de arquivos é limitada num diretório? Caso a quantidade de arquivos em um diretório seja limitada, quem gerencia é o sistema operacional ou o sistema de arquivos? Por que a criação de diretórios com determinados nomes não é possível?

Resolução da situação-problema

Existe um limite para a quantidade de arquivos em um diretório. Este limite é controlado pelo sistema de arquivos. Para verificar a quantidade de arquivos suportada num diretório, verifique qual é o sistema de arquivo utilizado. O sistema de arquivos NTFS e o EXT4 suportam mais de 4 bilhões de arquivos numa pasta ou diretório. Alguns nomes são utilizados nos sistemas operacionais, como palavras reservadas, ou seja, exclusivas para uso do sistema. Logo, não é possível criar arquivos com esses nomes. Por exemplo, a criação de um arquivo com o nome "con" é barrada pelo sistema, pois "con" vem de console, e este é um comando que prepara a entrada de dados via teclado.

Faça valer a pena

1.



Árvore de diretórios ou *folder tree* é a forma como as pastas dos sistemas Linux estão organizadas. Elas têm uma hierarquia, para facilitar a organização do sistema, seus arquivos, bibliotecas e inclusive para melhorar a segurança do sistema.

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

Sobre o sistema de diretório hierárquico do Linux, marque a alternativa correta.

- a) No primeiro nível, estão os "diretórios de Usuário", usados pelo HOME e Kernel do sistema.
- b) A partir do diretório /usr começa a terceira hierarquia dos sistemas de arquivos Linux.
- c) Na segunda hierarquia, ficam funcionalidades privadas /usr/local.
- d) /tmp – diretório onde estão armazenados os arquivos temporários, que não são deletados ao iniciar o sistema.
- e) /boot – diretório onde estão os arquivos usados na inicialização do sistema.

2.



Quando usuários acostumados com o Windows se deparam pela primeira vez com o sistema Linux, uma das confusões que aparecem é com relação a diferença entre os sistemas de arquivos implementado pela Microsoft e pelo Linux.

A seguir, analise as asserções:

No Linux, os discos e partições não aparecem necessariamente como unidades diferentes, como o C:, D: e E: do Windows.

POR QUE

Tudo faz parte de um único diretório, chamado diretório-raiz. Assim, de acordo com tipo de arquivo, são direcionados para os diretórios específicos.

Com relação às duas asserções assinale a alternativa correta:

- a) As asserções I e II são proposições verdadeiras, e a II é uma justificativa da I.
- b) As asserções I e II são proposições verdadeiras, mas a II não é uma justificativa da I.
- c) A asserção I é uma proposição verdadeira, e a II é uma proposição falsa.
- d) A asserção I é uma proposição falsa, e a II é uma proposição verdadeira.
- e) As asserções I e II são proposições falsas.

3. O caminho absoluto é o nome completo do caminho e o caminho relativo é o nome parcial do caminho. Imagine que você tenha criado um projeto de um sistema web no caminho C:\usuarios\carlos.pereira\Projetos\SistemaWeb\Paginas\Index.html.

Qual é o nome do caminho absoluto e qual o nome do caminho relativo para o projeto do sistema web?

Assinale a alternativa correta.

- a) Caminho absoluto: C:\usuarios\carlos.pereira\Projetos\SistemaWeb\Paginas\Index.html; e caminho relativo: \Paginas\Index.html.
- b) Caminho absoluto: \SistemaWeb\Paginas\Index.html; e caminho relativo: C:\usuarios\carlos.pereira\Projetos\SistemaWeb\Paginas.
- c) Caminho absoluto: C:\usuarios\carlos.pereira\; e caminho relativo: \Projetos\SistemaWeb\Paginas\Index.html.
- d) Caminho absoluto: C:\usuarios\carlos.pereira\Projetos; e caminho relativo: \Paginas\Index.html.
- e) Caminho absoluto: C:\usuarios\carlos.pereira\Projetos\SistemaWeb\Paginas\Index.html; e caminho relativo: \SistemaWeb\Paginas\Index.html.

Seção 3.3

Implementação e segurança de sistemas de arquivos

Diálogo aberto

Caro aluno, existem vários métodos de implementação do sistema de arquivos e diretórios, e nesta seção 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. Nesta seção, você conhecerá 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. 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!

Não pode faltar

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.

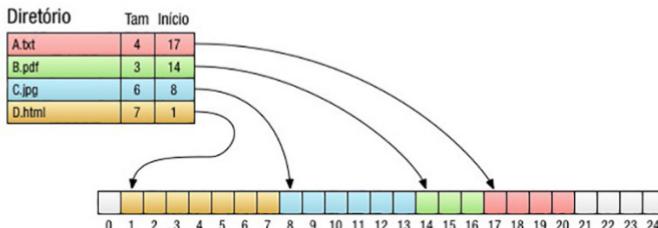
Implementação do Sistema de Arquivos

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 3.10) 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 3.10 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.

Figura 3.10 | Alocação Contígua

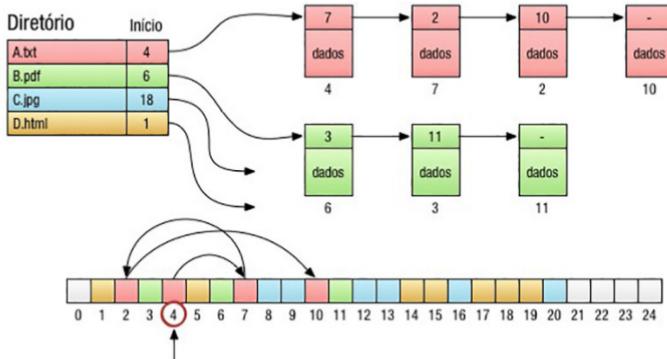


Fonte: Junior ([s.d.]).

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 3.11 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.

Figura 3.11 | Alocação por Lista Encadeada

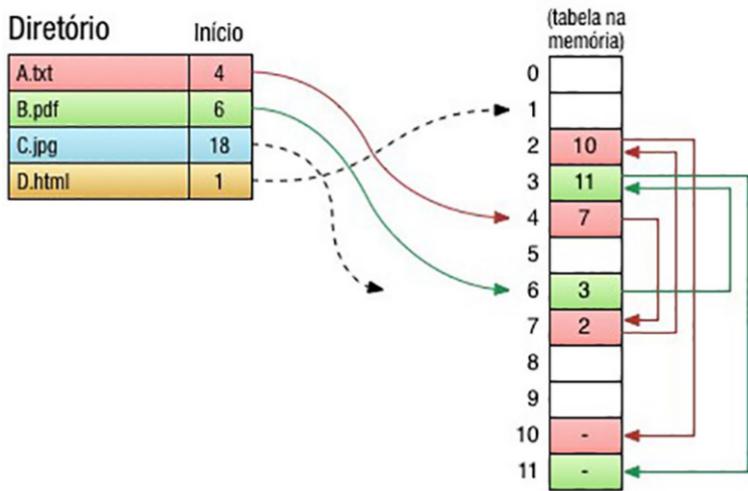


Fonte: Junior ([s.d.]).

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 3.12 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.

Figura 3.12 | 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.

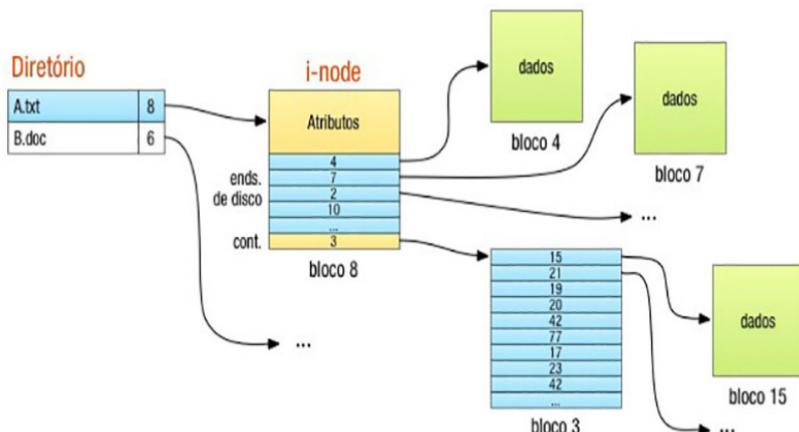


Exemplificando

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 3.13), com os seus atributos e endereços de disco, que fazem referência aos blocos de arquivos.

Figura 3.13 | 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.



Refletia

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:

LINUXANDO. **Compreender o Inode em Linux.** Disponível em: <https://www.linuxando.com/tutorial.php?t=Compreender%20o%20Inode%20em%20Linux_7>. Acesso em: 22 maio 2018.

Implementação de Arquivos e Diretórios

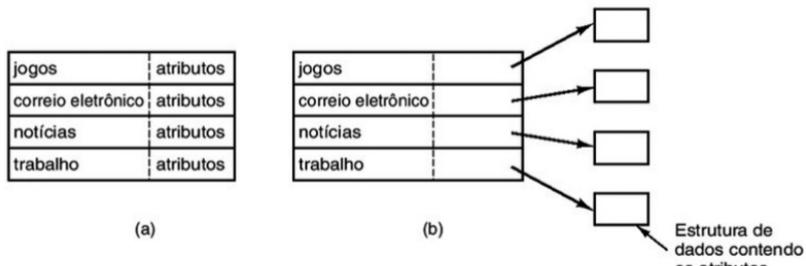
Segundo Tanenbaum (2003), para localizar a entrada de um diretório, o sistema operacional usa o nome do caminho do arquivo dado pelo usuário. A entrada de um diretório contém a informação

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 3.14 (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 3.14 (b).

Figura 3.14 | Atributos na entrada do diretório (a) e atributos em I-nodes (b)



Fonte: Tanenbaum (2003, p. 305).

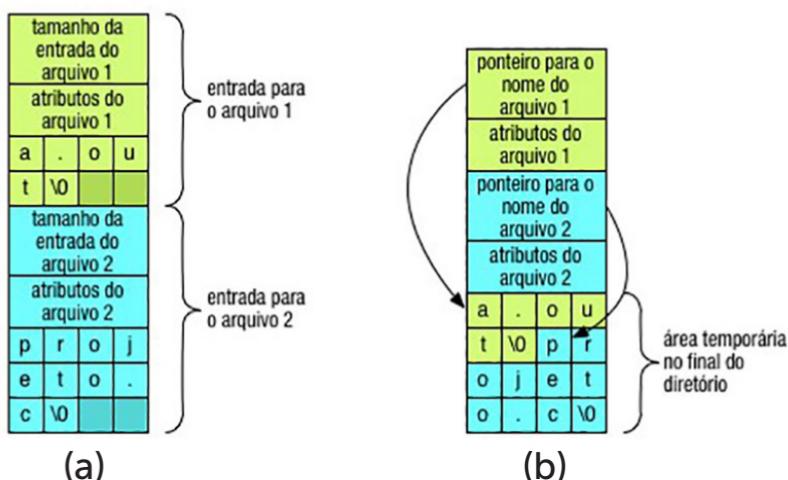
Ainda segundo Tanenbaum (2003), os sistemas operacionais modernos suportam nomes de arquivos mais longos e com tamanhos variáveis. Para fazer essa implementação, é necessário definir um limite de 255 caracteres para o tamanho do nome do arquivo. Essa implementação é simples, mas consome uma quantidade grande e desnecessária de espaço no diretório, uma vez que nem todos os nomes de arquivos são grandes.

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 3.15 (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 3.15 (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.

Figura 3.15 | Atributos na entrada do diretório (a) e atributos em I-nodes (b)

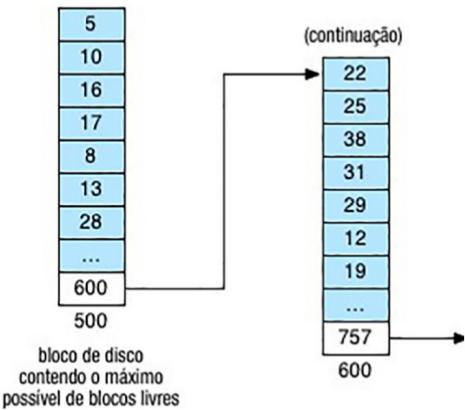


Fonte: Junior ([s.d.]).

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 contém a quantidade de espaços livres que puderem ter e possuem a localização dos blocos livres, conforme apresentado na Figura 3.16. Por exemplo, um disco de 16GB precisa de uma lista livre de 16.794 blocos para conter 2^{24} número de blocos de disco.

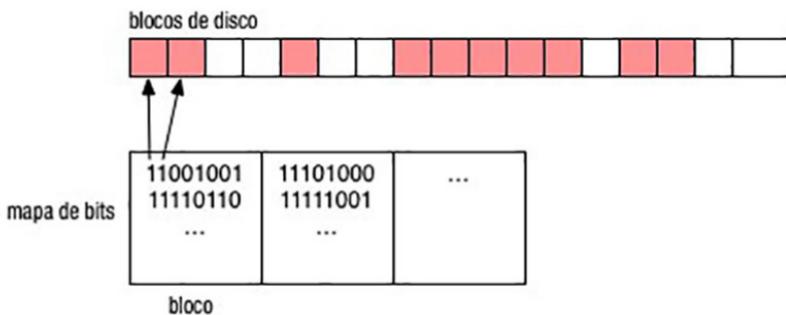
Figura 3.16 | Lista encadeada de blocos



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 3.17. 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.

Figura 3.17 | 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 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.



Assimile

Vimos nesta seção que existem diferentes mecanismos de proteção de um sistema de arquivos. A escolha do mecanismo dependerá do sistema operacional utilizado. Por exemplo, no Linux, os usuários e grupos de usuários são agrupados por atividades em comum e possuem acesso para realizarem essas tarefas.

Sem medo de errar

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.

Segurança do sistema de arquivos

Descrição da situação-problema

Uma empresa de leilão on-line contratou diversos funcionários para auxiliarem na força tarefa de vendas de produtos, batizada de Fidelização de Clientes de Produtos on-line. No site da empresa são vendidos produtos de vários tipos e marcas, de lojas virtuais, com descontos de até 70%. A empresa trabalha com máquinas Windows. Você foi contratado como técnico de TI e sua primeira atividade é criar as contas de usuário para cada funcionário admitido para trabalhar na força tarefa. Para evitar qualquer problema no sistema de arquivos, antes da utilização por parte dos novos funcionários, você fez o backup dos arquivos e rodou o antivírus no servidor Windows. No dia seguinte à criação de contas e backup, um coordenador da área de vendas identificou que vários arquivos de fidelização de vendas salvos não estavam mais disponíveis. Ele entrou em contato com a TI, relatando o acontecido. Você foi questionado sobre o desaparecimento dos arquivos, uma vez que o sistema possui segurança e controle de usuários. Como fazer para reverter esta situação?

Resolução da situação-problema

Em algumas situações, pode acontecer de sumirem alguns arquivos do computador, mesmo com a proteção dos arquivos por parte do sistema operacional. Por exemplo, no Windows, em função da criação de novas contas de usuários, por vírus deletados ou até mesmo por alguma falha no disco rígido, pode acontecer a perda de dados pessoais. Como você criou as contas dos usuários e rodou o antivírus do computador, isso pode ter ocasionado o desaparecimento dos arquivos que provavelmente estavam infectados com algum vírus, ou, durante a criação destas contas, o sistema operacional ter criado uma nova conta para o coordenador de vendas. Neste caso, basta acessar o gerenciador de contas de usuários que a conta antiga do coordenador de vendas estará lá com todos os arquivos. Também pode ser feita a recuperação dos arquivos de backup para recompor os arquivos que sumiram.

Faça valer a pena

1. Com relação aos métodos de implementação do sistema de arquivos, analise as questões a seguir e associe as colunas de acordo com a característica de cada método:

1 - Alocação Contígua

() nesse método, os atributos e os endereços em disco dos blocos de arquivos são associados a uma estrutura de dados.

2 - Alocação Por Lista Encadeada

() nesse método, o acesso aleatório aos arquivos é mais fácil, porém a tabela deve estar na memória o tempo todo.

3 - Alocação Por Lista Encadeada usando uma tabela na memória

() nesse método, a estrutura de dados só precisa estar na memória enquanto o arquivo estiver aberto.

4 - I-nodes

() nesse método, o acesso aleatório aos arquivos é muito lento.

() esse método apresenta a vantagem de possuir um bom desempenho e a desvantagem de fragmentar o disco.

Assinale a alternativa que possui a ordem correta da associação das duas colunas:

- a) 4, 2, 4, 1, 2.
- b) 2, 1, 3, 4, 3.
- c) 4, 3, 4, 2, 1.
- d) 1, 3, 4, 2, 4.
- e) 3, 4, 3, 2, 1.

2. Ao criar um arquivo, é necessário gerenciar o espaço livre, identificando quais blocos e setores estão livres para criação dos arquivos. De acordo com os métodos de Gerenciamento de Espaço em Disco, analise as afirmações a seguir.

I – No método mapa de bits, para cada bloco do sistema de arquivos é definido um bit.

II - No método da lista encadeada de blocos, cada bloco contém a quantidade de blocos livres que puder ter e possui a localização dos blocos livres.

III – No mapa de bits, o bit igual a 1 corresponde ao bloco está em uso e o bit 0 ao que não estiver.

IV – a desvantagem do mapa de bits em relação à lista encadeada de blocos é que o sistema de arquivos consegue saber rapidamente onde há blocos livres.

Assinale a alternativa correta:

- a) I, II e III estão corretas.
- b) I, III e IV estão corretas.
- c) II, III e IV estão corretas.
- d) I e II estão corretas.
- e) III e IV estão corretas.

3.



De acordo com uma pesquisa recente da ESET, 87% dos usuários fazem backup de suas informações, principalmente em HDs externos e depois na nuvem, escolhendo prioritariamente documentos de trabalho ou estudo, fotos e senhas. Entre as principais razões, 32% dizem que é por medo de defeito ou erro de hardware, 21% por roubo de informação e 20% por informação corrompida. Por outro lado, entre os usuários que não realizam backup, 72% mencionaram que isso se deve ao esquecimento e, por não saber fazê-lo, 78% perderam informações e 31% perderam dinheiro.

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: 5 jun. 2018.

Com relação aos mecanismos de proteção dos sistemas operacionais, analise as asserções a seguir:

Os sistemas operacionais devem fornecer técnicas para a recuperação dos dados afetados em caso de falhas no sistema de arquivos. A maioria dos sistemas operacionais implementam técnicas de cópias de segurança (backups), de forma que o sistema recupere os dados perdidos em caso de um desastre.

POR QUE

Os dados corrompidos ou perdidos, especialmente dados críticos de negócios coletados em um backup, podem representar a sobrevivência de uma empresa.

A respeito dessas asserções, assinale a opção CORRETA.

- a) As asserções I e II são proposições verdadeiras, e a II é uma justificativa da I.

- b) As asserções I e II são proposições verdadeiras, mas a II não é uma justificativa da I.
- c) A asserção I é uma proposição verdadeira, e a II é uma proposição falsa.
- d) A asserção I é uma proposição falsa, e a II é uma proposição verdadeira.
- e) As asserções I e II são proposições falsas.

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: 7 maio 2018.
- COTA, A. **Diferenças entre o sistema de arquivos do Windows e Linux.** 17 mar. 2005. Disponível em: <<https://www.vivaolinux.com.br/artigo/Diferencias-entre-o-sistema-de-arquivos-do-Windows-e-Linux>>. Acesso em: 11 maio 2018.
- 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: 22 maio 2018.
- DEITELL, H. M.; DEITEL, P. J.; CHOHNES, 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: 24 maio 2018.
- 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: 5 jun. 2018.
- 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: 15 maio 2018.
- 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: 22 maio 2018.
- 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: 9 maio 2018.

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: 10 maio 2018.

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: 8 maio 2018.

Gerenciamento de dispositivos 2

Convite ao estudo

Caro aluno, seja bem-vindo! Nesta unidade, veremos como é realizado o gerenciamento de memória nos sistemas operacionais e suas características, como se dá a monoprogramação sem troca de processos ou paginação, e a multiprogramação com partições fixas, além de apresentar os problemas da multiprogramação: relocação e proteção. Estudaremos a troca de processos, conhecida como *swapping*, e a multiprogramação com partições variáveis. Ainda, aprenderemos como são realizados o gerenciamento de memória por meio de mapa de bits e os algoritmos de troca de processos, o funcionamento da memória virtual, da paginação e da tabela de páginas, assim também como são realizadas a segmentação da memória e os algoritmos de substituição de páginas. A memória é um dos recursos computacionais mais importantes, pois é o local onde são armazenados dados e informações no computador. Logo, é necessário que seja feito um bom gerenciamento da memória nos sistemas operacionais, uma vez que a tendência dos programas é consumir cada vez mais este recurso durante sua execução. Neste âmbito, vamos adentrar no seguinte contexto: você trabalha em uma empresa de infraestrutura como gerente de TI e foi convidado para representa-la em uma importante feira de tecnologia que acontecerá em sua cidade dentro de alguns dias. A empresa fornece serviços e soluções para problemas de recursos computacionais de acordo com a necessidade dos contratantes. Neste evento estarão presentes as melhores empresas na área de tecnologia do país e todas as inovações de gerenciamento de memória, *swapping* e memória virtual serão apresentadas. Você será

o responsável pela apresentação técnica das inovações em gerenciamento de memória, *swapping* (que é a paginação de memória em disco) e memória virtual da empresa para os clientes, apresentando os conteúdos teóricos e práticos sobre eles. Para isso será necessário dominar os seguintes conceitos: I) Saber como funcionam o controle e a alocação de memória principal para os processos, o gerenciamento de memória e mostrar como são gerenciados os problemas da multiprogramação (relocação e proteção); II) Saber como se dá o tratamento da insuficiência de memória dos processos que estão sendo executados, utilizando a técnica *swapping* (troca de processos) e como são definidas as estratégias de organização da memória através da multiprogramação com partições variáveis e do gerenciamento da memória com mapa de bits, além de apresentar os principais algoritmos de troca de processos; III) Mostrar como é tratada a insuficiência de memória dos processos que estão sendo executados, utilizando a técnica de memória virtual, e também mostrar as técnicas de paginação e segmentação e apresentar os algoritmos de substituição de páginas. Algumas observações importantes que podem surgir durante o evento: Qual a função do gerenciador de memória? Quais são os esquemas de gerenciamento de memória? Quando ocorrem os erros de páginas? E quais as ações que o sistema operacional torma quando ocorre este erro? Você deverá elaborar um anteprojeto de gerenciamento de memória e de dispositivos de entrada e saída para cada cliente que procurar o seu stand. Após o término desta unidade você terá condições de entender como é realizado o gerenciamento de memória, o funcionamento do *swapping* e da memória virtual no computador.

Vamos juntos conhecer mais sobre o gerenciamento de memória e suas propriedades! Vamos lá!

Seção 4.1

Gerenciamento de memória

Diálogo aberto

Prezado aluno, em um sistema operacional, o gerenciamento de memória é fundamental para garantir a eficiência das aplicações que rodam no computador, uma vez que a tendência é consumir cada vez mais memória durante sua execução. Iremos apresentar a evolução do gerenciamento de memória, desde o mais simples até os sistemas de gerenciamento de memória utilizados em computadores multiprogramados. Existem diferentes tipos de memória que auxiliam na execução das aplicações. Por exemplo, a memória cache é rápida e é utilizada juntamente com o processador, elevando assim o desempenho do computador. Quando uma página é acessada constantemente num navegador, ao invés do processador fazer uma busca das informações na memória RAM, ela busca as informações na memória cache. Neste caso, as informações acessadas são copiadas para essa memória, agilizando a recuperação da página desejada. O controle de todas é de responsabilidade do gerenciador de memória, que realiza a gestão de quais partes estão sendo usadas e quais não estão. Relembrando nosso contexto, você trabalha em uma empresa de infraestrutura como gerente de TI e irá representa-la em uma importante feira de tecnologia. Durante a feira, um funcionário de uma empresa do ramo de produtos de beleza, responsável pelo setor de TI, procura seu stand interessado em substituir os computadores nela utilizados. Ele ressalta que precisará de máquinas potentes, com memória que suporte o processamento de um grande volume de dados, uma vez que a empresa atua em nível nacional. Ele informou que no início da empresa, não obteve êxito na gestão das informações e teve que começar do zero, devido ao fato da memória dos computadores não suportar a quantidade de dados processados como pedidos de produtos, vendas realizadas, orçamento dos revendedores, entre outros. Em algumas situações os dados foram perdidos e/ou sobreescritos.

Surgem os seguintes questionamentos: qual a solução para este cliente referente ao gerenciamento de memória dos computadores de sua empresa? Qual estratégia poderá ser adotada para obter um bom desempenho de memória considerando a situação atual da empresa? Para que você consiga responder a esses e outros questionamentos sobre o gerenciamento de memória e os problemas da multiprogramação, vamos conhecer mais sobre eles. Não se esqueça que, ao fim desta unidade, você deverá entregar um relatório para o seu gestor, com os anteprojetos elaborados durante a feira.

Bons estudos!

Não pode faltar

Segundo Machado e Maia (2007), desde os primeiros sistemas operacionais a memória do computador era considerada como um recurso caro e escasso. Mesmo com a redução de custo e aumento da capacidade de armazenamento das memórias, gerenciar o seu uso é fundamental e complexo. Memória é o local de armazenamento de informações no computador e o seu gerenciamento em um sistema operacional é importante para garantir a eficiência das aplicações que rodam no computador, uma vez que a tendência das aplicações dos usuários é consumir cada vez mais esse recurso durante sua execução. Além disso, o gerenciamento de memória permite que vários processos sejam executados ao mesmo tempo, mantendo um bom desempenho do sistema (TANENBAUM, 2003).

Machado e Maia (2007) afirmam que o sistema operacional deve proteger as áreas de memória utilizadas pelos processos de forma que, se um programa tente acessá-la indevidamente, o sistema seja impedido.



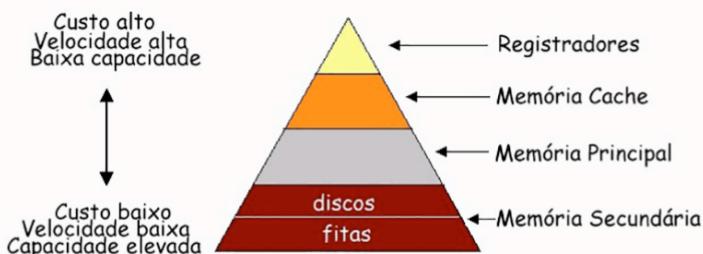
Refletia

A memória é um recurso importantíssimo para um bom funcionamento de um computador. Quais são os desafios do gerenciamento de memória? Quais seriam os efeitos se o controle da memória fosse mal gerenciado?

Segundo Tanenbaum (2003), na maioria dos computadores existe o conceito de hierarquia de memória (Figura 4.1), combinando uma pequena quantidade de memória cache (rápida, alto custo e volátil – guardam a informação temporariamente), uma grande quantidade de memória principal (RAM – volátil, possui um médio custo) e uma memória secundária (baixo custo, alto armazenamento das informações, não volátil, com as informações armazenadas em disco).

Quanto mais alto as memórias estiverem na pirâmide, mais caras serão, possuirão alta velocidade de processamento e baixa capacidade de armazenamento. E quanto mais baixo e mais baratas são, possuirão uma grande capacidade de armazenamento e baixa velocidade de processamento.

Figura 4.1 | Hierarquia de memórias



Fonte: <<http://www.di.ufpb.br/raimundo/Hierarquia/Hierarquia.html>>. Acesso em: 11 jun. 2018.

Segundo Tanenbaum (2003), a hierarquia de memória é controlada pelo gerenciador de memória, responsável por gerenciar quais partes estão em uso e quais não estão, alocando-a quando os processos precisarem, liberando-a após o término dos processos e controlando a troca de processos entre a memória e o disco quando a memória principal não é suficiente para manter todos os processos em execução.



Pesquise mais

A hierarquia de memória fornece para as aplicações dos usuários e do computador uma grande quantidade de memória barata, rápida e não volátil. É necessário que estas memórias estejam em sintonia enquanto o computador esteja ligado, ou seja, que elas se interliguem

para garantir o bom funcionamento do sistema operacional. Para saber mais sobre este assunto acesse:

KLEINA, N. **Como funcionam as diferentes memórias quando o computador está em uso.** Disponível em: <<https://www.tecmundo.com.br/armazenamento/9415-como-funcionam-as-diferentes-memorias-quando-o-computador-esta-em-uso.htm>>. Acesso em: 12 jun. 2018.

Segundo Tanenbaum (2003), o gerenciamento de memória pode ser dividido em duas classes: **1)** Sistemas que durante o processamento levam e trazem a informação da memória para o disco (troca de processos e paginação) e **2)** Sistemas que não o fazem. A troca de processos (*swapping*) carrega todo o programa para a memória principal, o executa por um determinado tempo e depois o mesmo retorna para o disco. A paginação divide a memória em partições para a execução das aplicações de forma eficiente. Veremos a troca de processos na Seção 4.2 e a paginação na Seção 4.3.

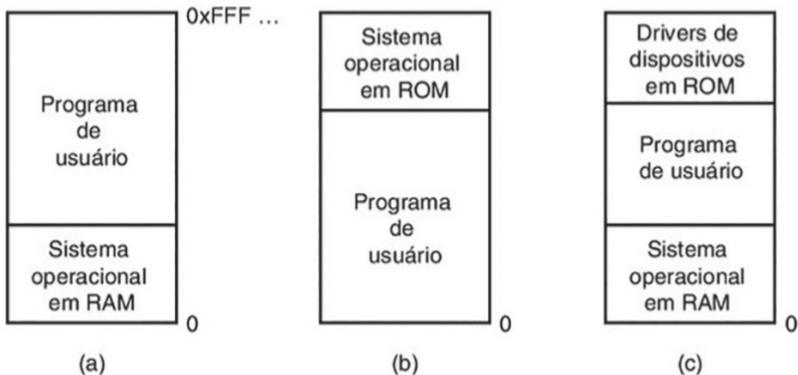
A seguir serão apresentados os mecanismos de gerenciamento de memória que não fazem troca de processos ou paginação.

Monoprogramação sem troca de processos ou paginação

Segundo Tanenbaum (2003), este método de gerenciamento de memória é o mais simples, pois somente um programa é executado por vez, e a memória é compartilhada entre o sistema operacional e o programa. A Figura 4.2 apresenta três formas que podem ocorrer neste esquema:

- O sistema operacional está utilizando o espaço de endereçamento em RAM (Figura 4.2 (a)), modelo aplicado aos mainframes e minicomputadores.
- O sistema operacional está utilizando o espaço de endereçamento em ROM somente para a leitura (Figura 4.2 (b)), usado em alguns computadores de mão e em sistemas embarcados.
- Os drives de dispositivos estão em ROM e os programas do usuário e o sistema operacional está em RAM (Figura 4.2 (c)) – modelo utilizado nos primeiros computadores pessoais (MS-DOS).

Figura 4.2 | Monoprogramação sem troca de processos ou paginação



Fonte: Tanenbaum (2003, p. 140)

Segundo Machado e Maia (2007), a alocação contigua simples (outro termo usado para Monoprogramação sem Troca de Processos ou Paginação) foi utilizada nos primeiros sistemas operacionais e ainda está presente em alguns sistemas monoprogramáveis. O usuário tem o controle sobre toda a memória, podendo acessar qualquer posição dela e, até mesmo, a área do sistema operacional. Neste método, a utilização de recursos computacionais não é eficiente, pois somente um usuário tem acesso aos recursos.

Ainda segundo Machado e Maia (2007), todos os programas são limitados à área de memória principal disponível para o usuário. Para resolver essa limitação, utiliza-se uma técnica chamada *overlay* (sobreposição), que consiste em dividir o programa em módulos, sendo cada um executado independentemente, utilizando a mesma área de memória. Cada módulo de código ou dados só é sobreposto quando não for mais necessário utilizar o que estava naquele local na memória anteriormente. O *overlay* era definido pelo programador e foi utilizado no MS-DOS.



Exemplificando

Por exemplo, imagine que um programa tenha três módulos: um principal, um de cadastramento e outro de impressão. Os módulos de cadastramento e impressão são independentes. Assim, quando o módulo de cadastramento estiver em execução na memória,

o módulo de impressão não precisa estar presente. O modulo principal é comum aos módulos de cadastramento e impressão, e deve estar na memória durante o tempo de execução do programa (MACHADO; MAIA, 2007).

Multiprogramação com Partições Fixas

Segundo Tanenbaum (2003), a maioria dos sistemas operacionais modernos permite que vários processos executem ao mesmo tempo (multiprogramação) e quando um processo é bloqueado aguardando uma informação de entrada/ saída, outro processo poderá utilizar a CPU, aumentando a sua utilização.

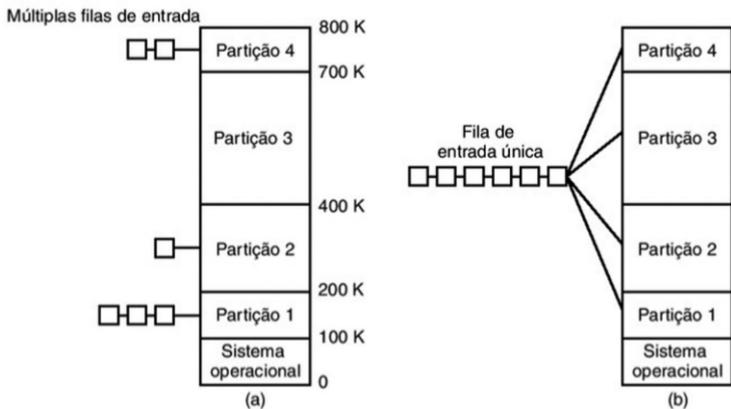
Nesse método, a memória é dividida em n partições de tamanhos diferentes, podendo ser definida quando o sistema for iniciado. Segundo Machado e Maia (2007), sempre que o tamanho de uma partição for alterado, o sistema deverá ser reiniciado com a nova configuração.

Segundo Tanenbaum (2003), quando um processo chega para ser executado, ele é inserido em uma fila associada à menor partição suficiente para armazená-lo. O espaço não utilizado pelo processo é perdido, uma vez que o tamanho das partições é fixo.

A Figura 4.3(a) apresenta filas de entrada separadas para cada partição. Por exemplo, na partição 4 existem dois processos na fila, sendo que estes podem ter um tamanho de até 100 k para utilizar. A desvantagem deste método é que podem existir muitos processos aguardando para serem executados em algumas partições (como nas partições 1 e 4) e em outras filas não existe nenhum processo (como na partição 3).

Uma solução é implementar uma fila única, assim, gerando um processo mais próximo do início da fila e que caiba na partição para ser carregado e executado (Figura 4.3 (b)). Para evitar desperdício de espaço não utilizado em uma partição, poder-se-ia percorrer a fila a procura do maior processo que comporte na partição. Porém, processos pequenos não seriam selecionados para executar. Uma solução seria implementar uma partição pequena e definir que um processo não poderia aguardar para receber uma partição mais do que n vezes.

Figura 4.3 | Partições fixas com (a) múltiplas filas de entrada e (b) com uma única fila



Fonte: Tanenbaum (2003, p. 141).



Vimos nesta seção como é importante gerenciar o uso da memória do computador. Nas linguagens de programação, o gerenciamento da memória é fundamental, pois a tendência das aplicações dos usuários é consumir cada vez mais este recurso. Em muitas linguagens de programação não é necessário se preocupar com o gerenciamento, porém é importante que sejam conhecidas as restrições e capacidades do gerenciador de memória para uma programação eficaz. Por exemplo, nas linguagens de programação C/C++ é preciso realizar o gerenciamento de memória.

BARTLETT, D. Por dentro do gerenciamento de memória. Disponível em: <<https://www.ibm.com/developerworks/br/library/l-memory/index.html>>. Acesso em: 14 jun. 2018.

Problemas da Multiprogramação: relocação e proteção

Segundo Tanenbaum (2003), a multiprogramação apresenta dois problemas essenciais: **relocação** e **proteção**. A **relocação** é a transferência de um código de um local para outro de forma que um programa não escreva na área de outro programa. A relocação é necessária uma vez que processos executam em diferentes endereços na memória física.

Como exemplo, podemos relatar o problema de relocação entre dois programas: quando o programa 1 está executando no endereço de memória entre 0 e 20. Após a finalização do programa 1, o programa 2 também foi alocado no endereço de memória entre 0 e 20, ambos com as suas instruções (ADD, MOV, JMP, dentre outras). Os programas 1 e 2, ao serem relocados, foram alterados os seus endereços iniciais de memória. Ao iniciarem a execução de suas instruções com novos endereços, o programa 2, que possui uma instrução de saltar para o endereço 20, invade a área de memória do programa 1. Esta é uma operação ilegal, uma vez que os processos não podem ler ou escrever endereços que pertençam a outros processos/usuários.

Segundo Tanenbaum (2003), quando um programa é ligado (combinação do programa principal e procedimentos executados pelo usuário e procedimentos de biblioteca), o *linker* – responsável por gerar o código executável – tem que saber o endereço inicial do programa para relocar todas as instruções que fazem uso de um endereço relativo de memória quando o programa for executado.

Uma solução para o problema de relocação é alterar as instruções do programa segundo a partição de memória em que ele está carregado. No caso da relocação dos programas 1 e 2, a instrução contida no endereço 20, seria direcionada para um endereço de memória dentro da área dos endereços do programa para não comprometer sua execução. Assim, o *linker* deve inserir no código binário uma lista ou um mapa informando quais são os códigos de operação precisam ser relocados e quais não precisam.

Segundo Tanenbaum (2003), a relocação de um programa durante a carga não resolve a proteção. Diferentes aplicações não podem acessar dados que não estejam dentro de sua partição. Se não houvesse esta proteção, caso um programa mal-intencionado insira uma nova instrução, ele poderá desviar o controle para ela, ocasionando uma falha de proteção.

Tanto os problemas de relocação quanto de proteção podem ser resolvidos fornecendo o processador com dois registradores (memória localizada no processador): base e limite. O registrador é uma memória localizada no processador do computador e

acelera a execução das aplicações. Ele está no topo da hierarquia de memória, possuem baixa capacidade de armazenamento, alta velocidade de processamento, porém são caros. Quando um processo é escalonado, o registrador-base é carregado com o endereço de início da partição e o registrador-limite com o tamanho da partição. Os endereços gerados são verificados no registrador-limite garantindo que não seja possível endereçar memória fora da região alocada ao processo durante a execução.



O registrador é uma memória rápida e cara localizada no processador do computador. Sua função é armazenar e processar operações através de instruções. O registrador-base recebe o endereço inicial da memória, enquanto que o registrador-limite recebe o tamanho da região de memória que o processo poderá executar.

Sem medo de errar

Agora que você já aprendeu sobre como é realizado o gerenciamento de memória nos sistemas operacionais e suas características e sobre os problemas da multiprogramação, vamos relembrar o nosso contexto. Durante a feira um funcionário de uma empresa do ramo de produtos de beleza, responsável pelo setor de T.I., procura seu stand interessado em substituir os computadores utilizados na empresa. Assim, surgem os seguintes questionamentos: qual a solução para este cliente referente ao gerenciamento de memória dos computadores de sua empresa? Qual estratégia poderá ser adotada para obter um bom desempenho de memória considerando a situação atual da empresa?

No caso deste cliente é necessário fazer um estudo em sua empresa para definir qual será a melhor estratégia para o seu negócio. O cliente relatou que processa um grande volume de informações, e a memória dos computadores não consegue processar a quantidade de dados geridos na empresa. Neste caso, é preciso ver quais programas e aplicações são realmente necessários nas máquinas da empresa para não sobrecarregar o processamento

dos dados principais do negócio do cliente. Estas aplicações extras instaladas nas máquinas podem fazer com que o computador fique lento ou pare de executar.

Uma solução inicial, é a aquisição de mais memória RAM, pois a maior quantidade permitirá que mais programas e aplicações sejam executados. A memória RAM auxilia o processador na transferência dos dados entre o disco e a memória. Porém o aumento da memória nem sempre acelerará o desempenho da máquina. Pode ser que o processador dos computadores não seja adequado para a quantidade de memória existente, não conseguindo endereçar um volume tão grande de memória RAM. Os processadores mais antigos não suportam uma memória maior que 3 GB. Talvez seja necessário realizar a troca do sistema operacional que contenha um processador mais eficiente que consiga atender a uma maior quantidade de memória.

Ainda segundo o relato do cliente, em algumas situações os dados foram sobreescritos. Pode ter acontecido um atraso na atualização dos dados pela rede e, assim, o último arquivo salvo sobre escreveu os arquivos existentes. Outra análise a ser feita é verificar como é realizada a relocação dos arquivos do computador.

O problema de relocação altera a partição (endereços) de memória em que ele está carregado. Mesmo os processos não podendo ler ou escrever endereços que pertençam a outros processos/usuários se tiver ocorrido algum problema na rede, por exemplo, o último processo que acessou a memória, invadiu o endereço de memória de outro processo apagando os dados. Verifique se os registradores do computador estão inibindo que outros processos aloquem memória em regiões que estejam sendo utilizadas.

As soluções apresentadas auxiliam no desempenho de memória das máquinas e, consequentemente, o cliente não terá problemas. Porém, mesmo após todas as soluções apresentadas os computadores continuarem não suportarem a quantidade de dados processados, será necessário a troca das máquinas para outras mais potentes, para que não ocorra prejuízos nos negócios da empresa do cliente.

Avançando na prática

Gerenciando os problemas da multiprogramação: relocação e proteção

Descrição da situação-problema

A empresa KLS atua no ramo de venda de suprimentos de hardware (placa de vídeo, processador e memória do computador), e possui em seu portfólio serviços de automação residencial, telecomunicações, suporte a servidores e usuários. Você faz parte do time de vendas e suporte desta empresa. Um dono de uma *lan house* está com problemas de gerenciamento de memória em sua loja e procurou a sua empresa por ter assistência especializada nesta área. Ele relatou que possui máquinas antigas para realizar somente a impressão de documentos solicitados pelos clientes. Estes utilizam essas máquinas para selecionar o arquivo via pen-drive ou por e-mail e, após a seleção, os arquivos são enviados para uma máquina central para que o funcionário da Lan House imprima o documento. Porém, quando existem mais de dois clientes imprimindo seus arquivos, ele observou que durante a impressão, quando o último cliente envia os arquivos a serem impressos, o arquivo do cliente anterior é sobreescrito. Assim, surgem os seguintes questionamentos: por que acontece este problema? Como resolver o problema da relocação e proteção dos arquivos no computador central?

Resolução da situação-problema

Neste caso, deve ser verificado primeiramente se o problema está na rede na hora que o cliente envia o arquivo para ser impresso. Pode ter acontecido um atraso, e quando o último cliente enviou o arquivo foi sobreescrito pelo cliente anterior. Assim, desligue o computador e a impressora e faça o teste novamente. Se o problema persistir, o erro pode estar na relocação e proteção da máquina central de onde os arquivos são impressos. A relocação é a transferência de um código de um local para outro, de forma que um programa não escreva na área de outro programa. A relocação é necessária, uma vez que processos executam em diferentes endereços na memória física. Como os computadores são

antigos, a memória não consegue ser mapeada rapidamente, e o último cliente que gravou o arquivo invadiu a área de memória de outro cliente, fazendo com que o arquivo fosse perdido. Como a implementação dos registradores base e limite não é tão simples, recomenda-se que a máquina central seja trocada por uma máquina que atenda aos requisitos da *lan house* evitando assim futuros problemas.

Faça valer a pena

1. A memória pode ser vista como um grande gargalo que limita o desempenho do computador e, ao mesmo tempo, uma vilã, responsável pelo aumento dos custos para a implementação da solução computacional. Entretanto, os usuários gostariam de ter memórias com capacidade de armazenamento infinita, baratas, com baixíssimos tempos de resposta (análogos ao tempo de processamento da CPU) e que não fossem voláteis. Ah, sim, e que nunca estragassem! (SOUZA; BRUSCHI, [s.d.].)

Analise as asserções:

Em um computador é importante ter diferentes tipos de memórias para que todas as aplicações funcionem de forma eficiente e tenham baixos tempos de resposta.

PORQUE

Devido a diferentes memórias existentes em um computador, a hierarquia de memória contribui para uma efetiva e harmoniosa relação entre elas classificando-as em função do seu desempenho, contribuindo para o bom funcionamento do sistema.

Com relação às asserções acima assinale a alternativa correta.

- a) As asserções I e II são proposições verdadeiras e a II é uma justificativa da I.
- b) As asserções I e II são proposições verdadeiras, mas a II não é uma justificativa da I.
- c) A asserção I é uma proposição verdadeira e a II é uma proposição falsa.
- d) A asserção I é uma proposição falsa e a II é uma proposição verdadeira.
- e) As asserções I e II são proposições falsas.

2. Em relação aos mecanismos de gerenciamento de memória que não fazem troca de processos ou paginação, analise as afirmações a seguir.

- I) Na Monoprogramação sem Troca de Processos ou Paginação o usuário tem o controle sobre toda a memória, porém o acesso só pode ser realizado nos endereços especificados.
- II) A técnica *overlay* foi utilizada no MS-DOS e consiste em dividir o programa em módulos de forma que eles utilizem a mesma região de memória.
- III) A Multiprogramação com partições fixas com múltiplas filas tem a vantagem do processo saber em qual endereço de memória irá executar e, com isso, não acontece o desperdício de espaço na partição.
- IV) A implementação de uma partição pequena para atender processos menores evitando que eles não sejam selecionados para executar refere-se à multiprogramação com partições fixas com uma única fila.

Assinale a alternativa correta.

- a) I, II, III estão corretas.
- b) I, III, IV estão corretas.
- c) II, III, IV estão corretas.
- d) I, II estão corretas.
- e) II e IV estão corretas.

3. A multiprogramação permite que vários programas sejam executados em memória ao mesmo tempo, dando a ilusão ao usuário que as atividades são executadas em paralelo, como fazer o download de um arquivo e ouvir música. Os problemas encontrados na multiprogramação são a relocação e proteção.

Considerando as questões acerca da relocação e proteção, marque a alternativa correta.

- a) No registrador-base é armazenado o tamanho da partição, e antes de executar o programa os endereços de memória são conferidos, garantindo que um processo não invada a área de outro processo.
- b) A solução para o problema de relocação é manter as instruções do programa segundo a partição de memória em que ele está carregado para que, durante a carga, não existam problemas no endereçamento.
- c) Na relocação os processos podem ler ou escrever endereços que pertençam a outros processos/usuários, desde que seja sinalizado durante a execução.

- d) Programas maliciosos podem acessar o controle para suas instruções sem prejudicar a proteção das partições de memória.
- e) O *linker* tem que saber o endereço inicial do programa para relocar todas as instruções que fazem uso de um endereço relativo de memória quando o programa for executado.

Seção 4.2

Swapping: troca de processos

Diálogo aberto

Olá, aluno! Seja bem-vindo a mais uma seção de estudos deste livro didático! Nessa seção você aprenderá os conceitos do *swapping*, os tipos e suas características, como se dá a multiprogramação com partições variáveis e o gerenciamento de memória com mapa de bits, além de entender o funcionamento dos algoritmos de troca de processos.

O *swapping* (troca de processos) é realizado quando não existe memória principal suficiente para executar todos os programas do computador ao mesmo tempo. No swapping um programa é totalmente carregado em memória e executado por um tempo definido, enquanto os demais programas aguardam, em disco, sua vez de executar. Para exemplificarmos o conceito de *swapping*, imagine que estamos usando o navegador para pesquisar sobre um determinado assunto e, logo que encontramos a informação que buscamos, abrimos o editor de texto para inserir essa informação. Em alguns casos, quando temos vários programas abertos e realizamos essa troca – no caso do navegador e do editor de texto –, percebemos um pouco de lentidão no computador. Na verdade, o que ocorre é a troca de processos (*swapping*), pois o navegador – que estava sendo executado até então – é tirado da memória e colocado em disco, enquanto o editor de texto, que estava no disco, por falta de utilização, é trazido para a memória e reativado.

Relembrando nosso contexto, você trabalha em uma empresa de infraestrutura como gerente de T.I. e irá representá-la em uma importante feira de tecnologia. Durante a feira, um representante de T.I. de uma empresa de digitalização de documentos visita o seu stand. A empresa digitaliza e gerencia diversos documentos de pessoas físicas e de outras empresas, preservando e prevenindo a perda de documentos importantes. O representante informou que processa um grande volume de documentos diariamente e liberou a memória física dos computadores, usando a partição do HD como

uma área de troca (*swap*) em que o sistema pode armazenar áreas de memória que não estão sendo usadas. Porém quanto menos memória física os computadores usavam e o swap era utilizado, ficavam cada vez mais lentos. Assim você é questionado: porque ocorre este problema? Quais as sugestões você dará para resolvê-lo? Aumentar a memória é sempre a melhor solução para evitar o *swapping* e a lentidão do computador? Qual o melhor algoritmo para definir em qual área livre o processo deverá ser executado na memória? Para que você consiga responder esses e outros questionamentos sobre *swapping* e os algoritmos de troca de processos, vamos conhecer mais sobre eles. Lembre-se que ao fim dessa unidade, você deverá entregar um relatório para o seu gestor, com os anteprojetos elaborados durante a feira.

Bons estudos!

Não pode faltar

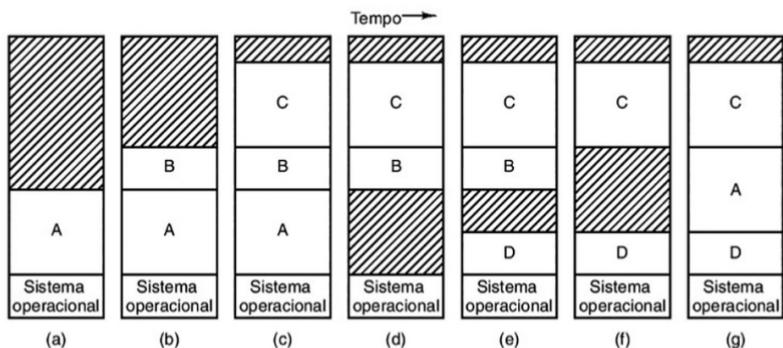
Segundo Tanenbaum (2003), em sistemas de compartilhamento de tempo ou computadores pessoais com memória RAM insuficiente para conter todos os processos ativos do sistema, é necessário que eles permaneçam em disco e sejam trazidos dinamicamente para a memória para serem executados somente quando necessário. Neste caso, podem ser utilizados dois métodos para o gerenciamento de memória: a troca de processos (*swapping*) que veremos a seguir e a memória principal que estudaremos na Seção 4.3.

Swapping: troca de processos

Segundo Tanenbaum (2003), a troca de processos (*swapping*) traz totalmente cada processo para a memória, o executa por algum tempo e o retorna para o disco.

A Figura 4.4 apresenta o funcionamento da troca de processos, sendo que, inicialmente, somente o processo **A** está na memória (Figura 4.4 (a)). O processo **B** é trazido do disco para a memória (Figura 4.4 (b)) e em seguida o processo **C** (Figura 4.4 (c)). O processo **A** é retirado da memória e retorna ao disco (Figura 4.4 (d)) e em seguida o processo **D** é trazido para a memória (Figura 4.4 (e)) e o processo **B** retorna para o disco (Figura 4.4 (f)). O processo **A** é trazido novamente na memória (Figura 4.4 (g)).

Figura 4.4 | Troca de processos na memória principal



Fonte: Tanenbaum (2003, p. 145).

Segundo Machado e Maia (2007), o *swapping* permite maior compartilhamento da memória principal e utilização dos recursos do sistema computacional. Uma desvantagem é que aumenta o custo operacional de entrada/saída. Quando existe pouca memória RAM disponível, o sistema pode ficar dedicado à execução do *swapping*, deixando de realizar tarefas mais críticas, se tornando ineficiente.

O *swapping* não pode ser utilizado como solução para a questão da falta de memória RAM pelo fato de ser um método lento. Outra questão é que nem sempre adicionando mais memória no computador será resolvido o problema do *swapping* e performance, pois se o HD e a CPU estiverem ultrapassados, influenciarão o desempenho da máquina. Nos sistemas operacionais atuais, as memórias tendem a ser maiores e raramente acontece a troca entre memória e disco.



Refletá

Qual a vantagem de utilizar o *swapping* uma vez que os computadores atuais possuem memórias com elevada capacidade de processar várias informações ao mesmo tempo? E caso não seja suficiente, o usuário pode adquirir mais memória RAM para ser inserida no computador? Um computador pode ser usado sem *swapping*?

Além disso, é necessário manter o antivírus do computador sempre atualizado, uma vez que os vírus e programas maliciosos tendem a consumir muita memória, dificultando que programas com prioridade alta executem, forçando a utilização do *swapping*.

Um computador sem a devida utilização do antivírus, também pode ocasionar na demora para carregar os aplicativos e até mesmo travá-lo.



Exemplificando

Para exemplificarmos o conceito de *swapping*, imagine que computador tenha uma memória de 512 MB e você tem quatro processos para serem executados com os tamanhos 481 MB, 508 MB, 380 MB e 369MB, respectivamente. O gerenciador de memória seleciona um processo inteiro para ser executado em memória e os demais processos aguardam em disco a sua vez de executar. Se o primeiro processo (com o tamanho de 481 MB) for selecionado para ser executado, ele é carregado para memória e executado por um tempo determinado. Assim que o tempo finalizar, o processo retorna para o disco e outro é selecionado para executar. Para a seleção de qual processo será executado são utilizados algoritmos que utilizam de critérios para realizar a escolha.

Segundo Simioni (2018), no Windows o *swap* é um arquivo que está dentro do diretório raiz (C:/) chamado PAGEFILE.SYS. Este arquivo é criado quando o Windows é instalado. O tamanho do arquivo de *swap* varia de acordo com a quantidade de memória RAM do computador. Por exemplo, se o seu computador possui 2 GB de memória RAM, o tamanho do arquivo será de 2048MB. Além disso, o tamanho do arquivo pode ser alterado e/ou desativado. No Linux, o *swap* é uma partição no diretório raiz (/) e esta partição pode ser utilizada por vários sistemas operacionais.



Pesquise mais

O grande problema com o uso de *swap* é que o HD é muito mais lento que a memória RAM. Enquanto uma sequência de quatro leituras em um módulo de memória DDR2-800 demora cerca de 35 bilionésimos de segundo, um acesso a um setor qualquer do HD demora pelo menos 10 milésimos. Para saber mais sobre esse assunto, leia:

MORIMOTO, C. E. **Hardware para iniciantes, parte 3.** Disponível em: <<https://www.hardware.com.br/tutoriais/hardware-iniciantes-3/pagina3.html>>. Acesso em: 20 jun. 2018.

Multiprogramação com Partições Variáveis

Segundo Machado e Maia (2007), o esquema conhecido como alocação particionada dinâmica ou variável, consiste em ajustar dinamicamente o tamanho das partições de memória quando os processos chegam para serem executados. Além disso, cada processo utiliza um espaço necessário para executar, não acontecendo a fragmentação interna. Porém, em alguns casos, pode acontecer a fragmentação externa quando os processos, ao retornarem para o disco, deixarem pequenos espaços na memória, não permitindo que novos processos sejam executados, como por exemplo, o espaço entre os processos B e D (Figura 4.4 (e)).

A vantagem das partições variáveis é a flexibilidade por não estar preso a um número fixo de partições, melhorando a utilização da memória, porém impactando o gerenciamento das trocas de processos e na alocação e liberação da memória.

Segundo Tanenbaum (2003), quando as trocas deixam muitos espaços vazios é necessário agrupar esses espaços e combiná-los nos endereços mais baixos. Esta técnica é conhecida como compactação de memória, porém não é utilizada devido ao alto tempo de processamento.



Exemplificando

A compactação de memória é muito cara, pois gasta muito tempo de processamento. Por exemplo, se um computador possui uma memória de 256 MB e que possa copiar 4 bytes em 40 nanosegundos (ns), gastaria em torno de 2,7 segundos para compactar toda a memória (TANENBAUM, 2003, p. 145).

Quando os processos precisam consumir mais memória durante o processamento, é necessário alocar memória dinamicamente. Existem dois métodos de gerenciamento de memória com alocação dinâmica: com **Mapa de Bits** e com **Listas Encadeadas**.

Gerenciamento de Memória com Mapa de Bits

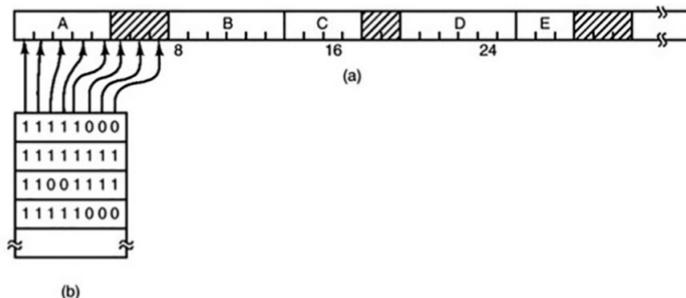
Segundo Tanenbaum (2003), neste método a memória é dividida em unidades de alocação, a qual é associado um bit no mapa de

bits. Se o valor do bit for 1, indica que a unidade está ocupada e o se o bit for 0 ela está livre.

A Figura 4.5 apresenta o mapa de bits e a memória dividida em unidades de alocação A Figura 4.5 (a) mostra uma parte da memória com cinco segmentos alocados a processos (**A**, **B**, **C**, **D** e **E**) e três segmentos de memória livre (espaços vazios). A Figura 4.5 (b) apresenta o mapa de bits correspondente a memória. Segmento é uma área de memória alocada a um processo ou uma área livre de memória entre dois processos.

Por exemplo, na primeira partição da memória representada por 8 bits, foi alocado o processo **A** com cinco unidades de alocação (bits igual a 1) e três unidades de memória livre (bits igual a 0). Na partição seguinte, o processo **B** alocou seis unidades e o processo **C** somente 2, uma vez que as partições foram divididas em 8 bits cada e não comportava todo o processo. O processo **C** utilizou duas unidades de alocação da próxima partição, duas unidades de memória livre e o processo **D** utilizou 4 unidades. Na última partição do exemplo, o processo **E** utilizou duas unidades de alocação, o processo **E** três unidades e três unidades de memória livre. Quanto menor for a unidade de alocação, maior será o mapa bits.

Figura 4.5 | Mapa de bits e uma parte da memória



Fonte: Tanenbaum (2003, p. 146).

A vantagem do mapa de bits é a simplicidade do gerenciamento de alocação de memória, uma vez que o tamanho do mapa depende somente do tamanho da memória e da unidade de alocação. A desvantagem surge quando é necessário carregar um processo com um determinado número de unidades consecutivas, necessitando que o gerenciador de memória procure essa sequência no mapa de bits, cujo processo é lento.

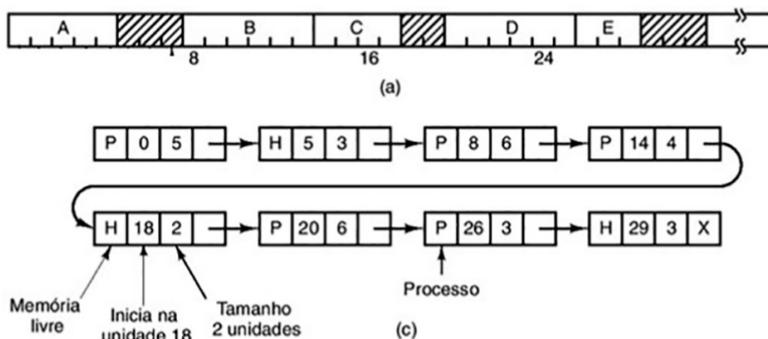
Gerenciamento de Memória com Listas Encadeadas

Segundo Tanenbaum (2003), este método consiste em manter uma lista encadeada de segmentos alocados e livres na memória. A Figura 4.6(a) mostra uma parte da memória com cinco segmentos alocados a processos (**A**, **B**, **C**, **D** e **E**) e três segmentos de memória livre (espaços vazios). A Figura 4.6(c) apresenta a lista encadeada e a memória dividida em unidades de alocação. Cada elemento dessa lista especifica um segmento de memória disponível (**H**) ou de memória alocada ao processo (**P**), o endereço onde se inicia o segmento e um ponteiro para o próximo elemento da lista.

Por exemplo, o processo **A** possui memória alocada (**P**), inicia na posição 0 (zero), tem o tamanho de cinco unidades de alocação e possui um ponteiro para o próximo elemento. O próximo elemento é um segmento de memória disponível (**H**), inicia na posição 5, possui o tamanho de três unidades e aponta para o próximo segmento.

A vantagem deste método é que a lista está ordenada por endereço facilitando a rápida atualização após o término do processo.

Figura 4.6 | Lista encadeada e uma parte da memória



Fonte: Tanenbaum (2003, p. 146).

Algoritmos de Troca de Processos

Segundo Machado e Maia (2007), existem estratégias ou algoritmos para definir em qual área livre os processos serão executados por meio da lista encadeada. Estes algoritmos reduzem ou evitam a fragmentação externa.

Os algoritmos de alocação de memória são (TANENBAUM, 2003):

- *First Fit* (primeiro que couber): é o algoritmo mais simples e que consome menos recurso do sistema. O gerenciador de memória procura ao longo da lista por um segmento livre que seja suficientemente grande para esse processo.
- *Next Fit* (próximo que couber): este algoritmo é uma variação do *First Fit*. A posição em que encontra o segmento de memória disponível é memorizada, não precisando percorrer toda lista quando se quer alocar.
- *Best Fit* (melhor que couber): percorre toda lista e escolhe o menor segmento de memória livre suficiente ao processo. Este algoritmo é mais lento uma vez que procura em toda a lista.
- *Worst Fit* (pior que couber): sempre é escolhido o maior segmento de memória disponível de maneira que, quando dividido, o segmento disponível restante fosse suficientemente grande para ser útil depois. Simulações mostram que este algoritmo não é uma boa ideia a ser usado.
- *Quick Fit* (mais rápido que couber): É um algoritmo rápido e mantém listas separadas por tamanhos de segmentos de memória mais solicitados disponível.



Assimile

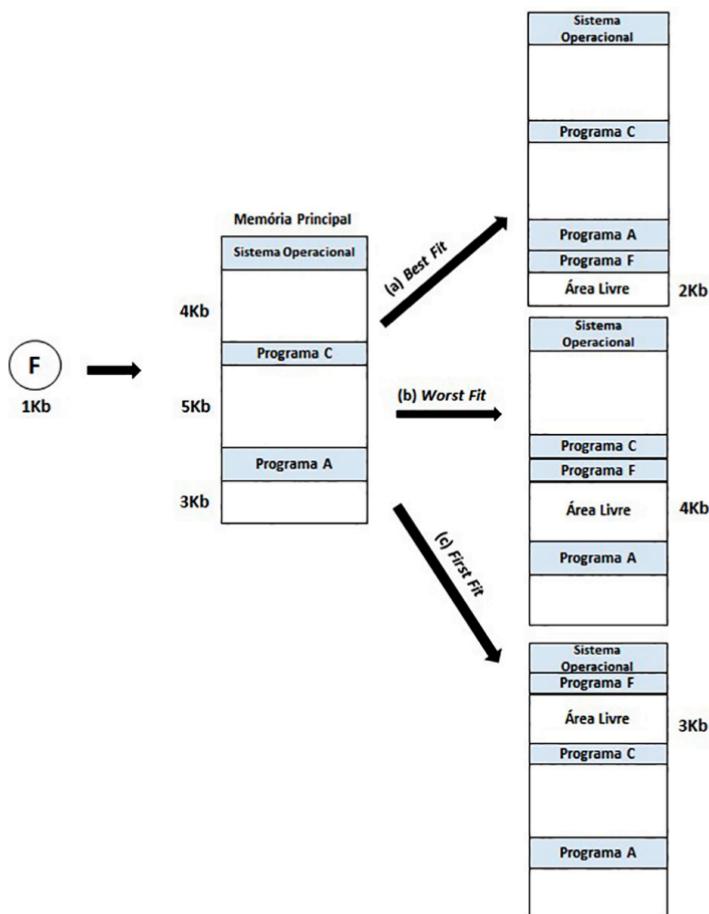
Vimos nesta seção como funcionam os algoritmos de troca de processos. Eles são usados para escolher em qual segmento de memória livre será alocado o processo que será executado. A estratégia a ser usada pelo gerenciador de memória para a escolha de qual algoritmo utilizar irá depender de fatores, como o tamanho dos processos executados no ambiente e das áreas livres disponíveis.

A Figura 4.7 exemplifica o funcionamento dos algoritmos *Best Fit*, *Worst Fit* e *First Fit*. O processo **F** (com tamanho de 1 kb) está aguardando para ser executado na memória principal. Na memória principal está em execução os processos **C** e **A** e contém três segmentos de memória livres com os tamanhos de 4 kb, 5 kb e 3 kb, respectivamente. Antes do processo **F** iniciar a execução na memória principal, o algoritmo *Best Fit* seleciona na lista o menor

segmento de memória livre. Assim, a partição com o tamanho 3 kb foi selecionada para executar o processo, deixando 2 kb de espaço vazio. Utilizando o algoritmo *Worst Fit* que escolhe o maior segmento de memória disponível, a partição com o tamanho 5 kb foi selecionada para executar o processo, deixando 4 kb de espaço vazio. Se for utilizado o algoritmo *First Fit* que procura na lista pelo primeiro segmento livre que seja suficientemente grande para esse processo, será selecionada a partição com o tamanho 4 kb para executar o mesmo, deixando 3 kb de espaço vazio.

Nos algoritmos *First Fit* e *Best Fit* ocorre a fragmentação externa.

Figura 4.7 | Algoritmos de alocação de memória



Fonte: Machado e Maia (2007, p. 155).

Sem medo de errar

Agora que você já conheceu o funcionamento do *swapping* e dos algoritmos de troca de processos, vamos voltar ao nosso contexto. Você é um gerente de T.I. de uma empresa de infraestrutura e está representando-a numa feira de tecnologia. Um representante de T.I. de uma empresa de digitalização de documentos visita o seu stand e relata que digitaliza e gerencia diversos documentos de pessoas físicas e de outras empresas, preservando e prevenindo a perda de documentos importantes. Ele ainda relatou que os computadores ficavam cada vez mais lentos, apesar de diminuírem o uso de memória física. Porque ocorre este problema? Aumentar a memória RAM é sempre a melhor solução para evitar o *swapping* e a lentidão do computador? Quais as sugestões você dará para resolvê-lo? Qual o melhor algoritmo para definir em qual área livre o processo deverá ser executado na memória?

Os sistemas operacionais quando possuem pouca memória RAM utilizam o disco ou um arquivo para processar os programas, carregando-os na memória principal, executando-os e, logo após a execução, o retorna para o disco. Esse método evita que os programas travem, porém impacta diretamente no desempenho do sistema operacional. O *swapping* não pode ser utilizado como solução para a questão da falta de memória RAM. No caso do estudante, ele reservou uma partição do HD para ser utilizada como uma área de *swapping*. A utilização do *swapping* deixa a máquina mais lenta, porque o processador precisa procurar no disco as informações necessárias. A memória RAM auxilia o processador, gravando as informações que são usadas frequentemente pelos processos e auxiliando na transferência dos dados entre memória e disco (DAQUINO, 2018). Nem sempre adicionando mais memória será resolvido o problema do *swapping* e performance, pois outros fatores como, HD e a CPU antigos, influenciaram o desempenho da máquina. No caso do representante de T.I. da empresa de digitalização de documentos, é necessário que ele adquira mais memória RAM para o seu computador de forma que um maior número de processos consiga executar diretamente nela, reduzindo o uso do *swapping*. Além disso, ele deve reduzir o tamanho da área do *swapping* para permitir que a maioria dos processos executem diretamente na memória principal, e somente quando necessário.

seja realizada a troca de processos. A estratégia utilizada pelo gerenciador de memória para a escolha de qual algoritmo utilizar irá depender de fatores, por exemplo, o tamanho dos processos executados. O algoritmo *First Fit* (primeiro que couber) é o que consome menos recurso do sistema por escolher a primeira partição livre que comporte a execução do processo, porém existe um grande desperdício de espaço livre. O algoritmo *Best Fit* (melhor que couber) verifica em toda a memória a melhor partição que comporte a execução do processo, o que o torna mais lento. Nos algoritmos *First Fit* e *Best Fit* ocorre a fragmentação externa. Já o algoritmo *Worst Fit* (pior que couber) escolhe a maior partição livre para a execução do processo, porém não é recomendado o seu uso. Em casos que seja necessário criar mais espaço na memória é recomendado o uso do algoritmo *Worst Fit*. Já para agilidade do processo é recomendado que seja utilizado o algoritmo *Best Fit*.

Avançando na prática

Swapping – alto consumo de memória RAM

Descrição da situação-problema

Durante a feira de tecnologia, um cliente te procurou relatando que seu computador, utilizado apenas para reuniões da empresa, foi emprestado para outro funcionário em home office, uma vez que sua máquina está na manutenção. Esse cliente é consultor de marketing de uma multinacional e constantemente é acionado, via chat, para atendimento emergencial das unidades da empresa. Porém, durante um dos plantões, ao ligar o computador, seu carregamento demorou muito, além do aplicativo de chat e nenhum navegador da Internet abrir. Pelos seus conhecimentos em informática, ele identificou que a memória RAM estava 90% utilizada. Verificou ainda que um programa que anteriormente não estava instalado em seu computador, estava com um consumo elevado de memória. Assim ele te pergunta: porque aconteceu este problema? O consumo elevado de memória pelo programa pode ser vírus? Além disso, porque o *swapping* do computador não funcionou? Qual a solução você dará para este cliente?

Resolução da situação-problema

No caso deste cliente, a primeira questão a ser levada em consideração é o uso do computador por outro funcionário em home office, que pode ter acessado sites e links que continham vírus, o que explicaria o consumo elevado de memória pelo programa que não estava instalado no computador. Os vírus e programas maliciosos tendem a consumir muita memória, dificultando que programas com prioridade alta executem. Por isso, existe a necessidade constante de manter o antivírus do computador sempre atualizado. O *swapping* é utilizado sempre em sistemas operacionais quando a memória RAM do computador não consegue executar a quantidade de processos evitando que travamentos acontecem. Neste caso, como o consumo da memória estava alto, o *swapping* não suportou o processamento de todos os programas do computador ocasionando a demora para carregar, além do aplicativo de chat e nenhum navegador da internet abrir. Primeiramente deve ser feita a limpeza de todos os vírus da máquina e acompanhar o consumo de memória. Mesmo após a limpeza, se os processos continuarem a consumir alto volume de memória impedindo que o *swapping* seja efetivamente realizado, é necessário adquirir mais memória para melhorar a performance da máquina.

Faça valer a pena

1.



Considere um sistema de troca de processos entre memória e disco em que a memória é constituída dos seguintes tamanhos de espaços vazios em ordem na memória: 10 KB, 4 KB, 20 KB, 18 KB, 7 KB, 9 KB, 12 KB e 15 KB. Utilizando os algoritmos *First Fit*, *Worst Fit* e o *Best Fit*, qual seriam as partições utilizadas pelos processos A e B com os tamanhos de 12 KB e 9 KB, respectivamente? (TANENBAUM, 2003, p. 197)

Assinale a alternativa correta:

- a) Processo A: 20 KB, 12 KB e 18 KB - Processo B: 18 KB, 9 KB e 20 KB.
- b) Processo A: 18 KB, 9 KB e 15 KB - Processo B: 15 KB, 12 KB e 18 KB.
- c) Processo A: 20 KB, 12 KB e 20 KB - Processo B: 20 KB, 9 KB e 20 KB.
- d) Processo A: 18 KB, 15 KB e 12 KB - Processo B: 12 KB, 15 KB e 20 KB.
- e) Processo A: 12 KB, 20 KB e 20 KB - Processo B: 9 KB, 20 KB e 20 KB.

2. Em relação aos métodos de gerenciamento de memória com alocação dinâmica com Mapa de Bits e com Listas Encadeadas, analise as afirmações a seguir.

- I) No gerenciamento de memória com mapa de bits a memória é dividida em unidades de alocação.
- II) A vantagem do método de lista encadeada é que a lista está ordenada por endereço facilitando a rápida atualização após o término do processo.
- III) No mapa de bits, quanto maior for a unidade de alocação, menor será o mapa bits.
- IV) A desvantagem do gerenciamento de memória com mapa de bits é a complexidade da alocação de memória, o tornando inoperante.

Assinale a alternativa correta:

- a) I, II, III estão corretas.
- b) I, III, IV estão corretas.
- c) II, III, IV estão corretas.
- d) I, II estão corretas.
- e) III e IV estão corretas.

3. "Afinal, o que é memória SWAP? Podemos dizer que seja uma técnica computacional usada pelos sistemas operacionais para aumentar quantidade de memória real do computador a fim de rodar os programas e o próprio sistema sem travamentos". (SIMIONI, 2014).

Analise as asserções:

O objetivo do *swapping* é auxiliar a memória RAM armazenada no seu HD, porém não é uma solução definitiva.

PORQUE

Se o seu sistema operacional está utilizando a área de swapping constantemente é necessário adquirir mais memória RAM para melhorar a performance do sistema.

Com relação asserções acima, assinale a alternativa correta.

- a) As asserções I e II são proposições verdadeiras, e a II é uma justificativa da I.

- b) As asserções I e II são proposições verdadeiras, mas a II não é uma justificativa da I.
- c) A asserção I é uma proposição verdadeira, e a II é uma proposição falsa.
- d) A asserção I é uma proposição falsa, e a II é uma proposição verdadeira.
- e) As asserções I e II são proposições falsas.

Seção 4.3

Memória virtual

Diálogo aberto

Prezado aluno, a memória virtual é um espaço reservado no disco rígido do computador para ser utilizado quando a memória RAM não é suficiente para executar os processos. Neste caso, uma parte dos processos ficam em disco aguardando a liberação da memória para executar enquanto outros processos estão em execução.

Para entendermos melhor o conceito de memória virtual, observe o seguinte exemplo: suponha que você quer executar um aplicativo de restauração de imagens que possui um tamanho de 100 MB e um programa de edição de textos com um tamanho de 60 MB, porém a memória RAM do seu computador tem espaço para comportar programas de até 145 MB. Neste caso, seria carregado na memória RAM o aplicativo de restauração de imagens (com 100 MB) e uma parte do programa de edição de textos (45 MB) e o restante ficaria em disco para serem executados posteriormente a medida que a memória fosse liberada.

Nesta seção você aprenderá os conceitos de memória virtual e suas características, como se dá a paginação, a tabela de páginas e a segmentação, além de entender o funcionamento dos algoritmos de substituição de páginas.

Relembrando nosso contexto, você trabalha em uma empresa de infraestrutura como gerente de T.I. e irá representá-la em uma importante feira de tecnologia. Durante a feira, um proprietário de uma loja de jogos digitais está à procura de ajuda para resolver o problema de baixa memória virtual dos computadores de sua loja e procura o seu stand. Ele alega que quando os computadores executam vários programas de uma vez e o consumo de memória é mais elevado, os mesmos chegam no limite do consumo e travam, além de fechar os programas que estão abertos.

Assim surgem os seguintes questionamentos: o que aconteceu no computador neste caso? Qual a solução que você dará para este problema? Quais são os algoritmos de substituição de páginas

mais eficientes em relação ao desempenho do sistema? Para que você consiga responder a esses e a outros questionamentos sobre memória virtual, vamos conhecer mais sobre eles, e, assim, você conseguirá entregar um relatório para o seu gestor, com os anteprojetos elaborados durante a feira.

Bons estudos!

Não pode faltar

Segundo Tanenbaum (2003), um dos grandes desafios dos programadores era lidar com programas maiores que a memória física disponível. Neste caso, a técnica de *overlay* (sobreposição) era utilizada para dividir o programa em módulos em que cada um destes era carregado em memória ou permanecia em disco à medida que eram solicitados. Porém, a divisão do programa em *overlay* era feita pelo programador, o que ocasionava demora em seu trabalho. Com isso, foi desenvolvida uma técnica cuja divisão do programa era feita pelo sistema operacional, chamada **memória virtual**.

Memória Virtual

Segundo Tanenbaum (2003), um computador que utiliza memória virtual permite que o volume de informações de um programa como código dado e pilha ultrapasse a quantidade total de memória física disponível para ele, mantendo as partes ativas na memória e as demais no disco rígido. A memória virtual é um arquivo dinâmico e de tamanho variável na maioria dos sistemas operacionais.

Segundo Machado e Maia (2007), a memória virtual permite que vários processos compartilhem a memória principal, uma vez que, somente algumas partes dos processos estarão ativas na mesma, possibilitando uma utilização eficiente do processador e reduzindo a fragmentação da memória principal. Em sistemas com memória virtual é utilizada uma técnica chamada paginação que será descrita a seguir.



Pesquise mais

A utilização da memória virtual permite o compartilhamento da memória física entre vários aplicativos ao mesmo tempo. Além disso,

cada programa enxerga uma versão virtual própria da memória. Para saber mais sobre esse assunto você encontra neste artigo:

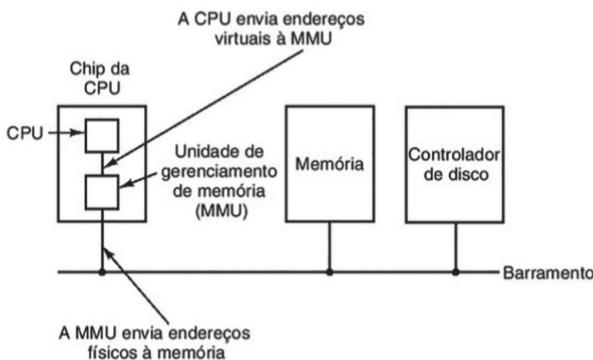
GUGIK, G. **Como funciona a memória virtual?** Disponível em: <<https://www.tecmundo.com.br/internet/2190-como-funciona-a-memoria-virtual-.htm>>. Acesso em: 28 jun. 2018.

Paginação e tabela de páginas

Segundo Machado e Maia (2007), a paginação é a técnica de gerência de memória em que o endereçamento virtual e o espaço de endereçamento real são divididos em blocos do mesmo tamanho, chamado páginas. Foi criada para fornecer um espaço de endereçamento linear sem a necessidade de adquirir mais memória física.

Segundo Tanenbaum (2003), os programas geram endereços virtuais e constituem o espaço de endereçamento virtual. Nos sistemas operacionais que não possuem memória virtual, o endereço virtual e o endereço físico são os mesmos. Já nos sistemas operacionais que trabalham com a memória virtual, o endereço virtual é enviado para a MMU (*memory management unit*, em que um chip está localizado na CPU (*Central Processing Unit* ou unidade central de processamento) que mapeia (estrutura) os endereços virtuais em endereços físicos, conforme apresentado na Figura 4.8. A CPU gera os endereços virtuais e os envia a MMU, que por sua vez envia os endereços físicos para a memória.

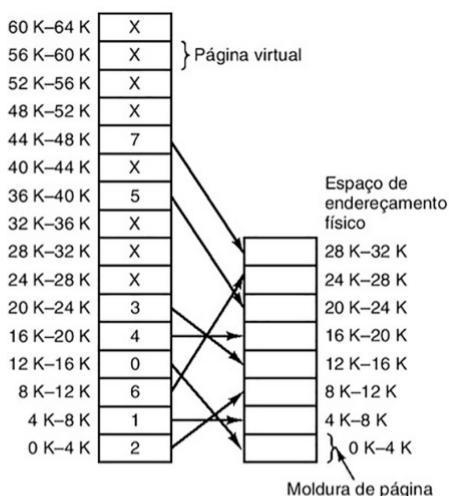
Figura 4.8 | Localização da MMU e sua função



Fonte: Tanenbaum (2003, p. 150).

Ainda segundo Tanenbaum (2003), a Figura 4.9 apresenta o funcionamento do mapeamento de endereços virtuais para endereços físicos realizado pela MMU. O computador possui uma memória virtual de 64 KB (gerando endereços virtuais de 16 bits de 0 a 64 K) e uma memória física de 32 KB. Programas podem ser maiores que o tamanho da memória física, mas não podem ser totalmente carregados nela, devendo permanecer em disco. O endereço virtual divide-se em unidades conhecidas como páginas e sua referência na memória física são as molduras de página. As páginas e as molduras de páginas têm o mesmo tamanho e a movimentação entre disco e memória são sempre realizadas em unidades de página.

Figura 4.9 | MMU – mapeamento de endereço virtual em físico



Fonte: Tanenbaum (2003, p. 151).

Na Figura 4.9 existem 16 páginas virtuais e oito molduras de página. Quando um programa tenta acessar o endereço 0, o endereço virtual 0 é enviado a MMU, que o localiza na página virtual (0 a 4 K) e que corresponde a moldura de página 2 (8 K a 12 K). A MMU então transforma o endereço virtual 0 no endereço físico 8 K e o envia à memória por meio do barramento. Como existem apenas oito molduras de página física, somente oito páginas virtuais podem ser mapeadas.

Se um programa acessar uma página que não esteja mapeada (como a página entre 32 k e 36 k), a MMU faz esta validação e desvia para o sistema operacional. Esta interrupção (*trap*) é chamada de falta de página.

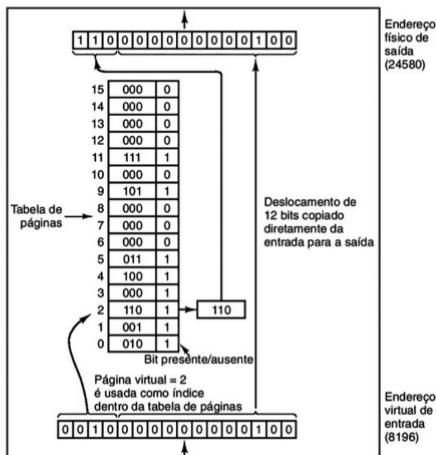
Segundo Tanenbaum (2003), na falta de página o sistema operacional seleciona uma moldura de página para salvar no disco (por exemplo, a moldura 1), marca na tabela de páginas virtuais a entrada escolhida como não mapeada (marcando com um X), carrega a página virtual referenciada pela instrução na moldura de página recém liberada (substituindo o X pela moldura 1), e marca na tabela de páginas virtuais a entrada da nova página virtual (página 8) como mapeada (moldura 1). Ou seja, a MMU transformará o endereço virtual 32 K no endereço físico 4 K. O mapeamento realizado pela MMU dá-se por meio da tabela de páginas. O objetivo da tabela é mapear páginas virtuais em molduras de página física.

Segundo Silberschatz, Galvin e Gagne (2015), a tabela de páginas contém o endereço virtual de cada moldura de página na memória física, e o número da página é utilizado como um índice na tabela. Cada processo possui sua tabela própria e cada página possui uma entrada nela.

Segundo Tanenbaum (2003), cada página na tabela possui um bit presente/ausente. Se o bit for 0 (zero), indica uma interrupção por falta de página e caso o bit tenha o valor 1, a página está mapeada na memória. O número da moldura deve ser concatenado com os bits de deslocamento formando o endereço físico.

A Figura 4.10 apresenta um exemplo de como a MMU trabalha junto com a tabela de páginas. O endereço virtual, 8196 (0010000000000100), é mapeado para o endereço físico 24580 (endereço físico de saída). O endereço virtual de 16 bits (0010000000000100) que chega a MMU é dividido num número de página de 4 bits (0010) mais significativos e um deslocamento de 12 bits (000000000100) menos significativos. O número da página (0010 = 2) é usado como índice para a tabela de páginas. Se o bit presente/ausente for 1, o número da moldura de página encontrado na tabela será copiado para os três bits (110) mais significativos concatenados ao deslocamento de 12 bits do endereço virtual que não sofreu alteração (**1100000000000100**). O registrador envia para a memória o endereço físico via barramento, conforme apresentado na Figura 4.10.

Figura 4.10 | Operação interna da MMU com 16 páginas



Fonte: Tanenbaum (2003, p. 152).

Algoritmos de substituição de páginas

Segundo Machado e Maia (2007), um dos maiores desafios na gerência de memória virtual por paginação é decidir quais as páginas devem ser liberadas. Quando uma falta de página ocorre, o sistema operacional precisa escolher uma a ser removida da memória, a fim de liberar espaço para uma nova ser trazida. Quanto menor for o tempo gasto com as recargas de páginas, mais eficiente será o algoritmo.

Segundo Tanenbaum (2003), se na memória a página teve alteração, ela deverá retornar ao disco rígido e atualizar a cópia da página virtual que está lá. Senão, não é preciso retornar com a página para o disco. Nos casos em que ocorre falta de página, pode ser escolhido de forma aleatória uma página a ser removida, porém é melhor escolher uma página que não foi muito usada.

Segundo Machado e Maia (2007), os algoritmos de substituição de páginas têm o objetivo de selecionar as páginas com as menores chances de serem referenciadas (utilizadas) no futuro. A seguir veremos os algoritmos de substituição de páginas.

- **Algoritmo de substituição de página ótimo**

Segundo Machado e Maia (2007), o algoritmo de substituição de página ótimo seleciona uma página que não será

referenciada no futuro ou aquela que demorará a ser utilizada novamente. Este algoritmo garante uma menor paginação, porém é impossível de ser implementado, uma vez que o sistema operacional não consegue prever o futuro das aplicações e saber quando cada página será referenciada novamente.

- **Algoritmo de substituição de página não recentemente utilizada (NUR)**

Segundo Tanenbaum (2003), muitos computadores que utilizam a memória virtual possuem dois bits R (referenciada) e M (modificada) associados a cada página virtual para identificar quais páginas físicas estão sendo usadas e quais não estão. No algoritmo NUR, quando um processo é iniciado os bits R e M são colocados em 0 para todas as páginas. Periodicamente, o bit R é limpo de modo que diferencie as páginas que não foram referenciadas recentemente daquelas que foram. Quando acontece uma falta de página, o sistema operacional verifica todas as páginas e as separa em quatro categorias, com base nos bits R e M:

- Classe 0: não referenciada, não modificada.
- Classe 1: não referenciada, modificada.
- Classe 2: referenciada, não modificada.
- Classe 3: referenciada, modificada.

O algoritmo remove aleatoriamente uma página de classe de ordem mais baixa que não esteja vazia, ou seja, remove uma página modificada, mas não referenciada. O algoritmo NUR é simples e de fácil implementação.

Por exemplo, suponha que as páginas virtuais 0, 1, 2 e 3, iniciaram todas com os valores zero. Num determinado momento foram associados aos bits referenciada e modificada os seguintes valores:

Página	Tempo	R	M
0	10	0	0
1	30	1	0
2	40	0	1
3	80	1	1

Como o algoritmo retira a página de ordem mais baixa, que esteja vazia e que tenha sido modificada, mas não referenciada, as páginas 0 e 2 poderiam ser removidas segundo o algoritmo NUR.

- **Algoritmo de substituição de página FIFO**

Segundo Machado e Maia (2007), no algoritmo de substituição de páginas FIFO (*first in first out*), a primeira página utilizada será a primeira a ser escolhida para ser removida. As páginas são inseridas em uma fila sendo que as mais antigas estão no início e as mais recentes no final. É um algoritmo de baixo custo, porém quase não é implementado em função de escolher páginas antigas que sempre são referenciadas.



Exemplificando

Para exemplificar o funcionamento do algoritmo FIFO, suponha que as páginas virtuais 0, 1, 2 e 3, iniciaram todas com os valores zero e estão em fila. A página 0 foi a primeira página a ser utilizada no tempo 10, logo em seguida a página 1 no tempo 30, depois a página 2 no tempo 40 e por último a página 3 no tempo 80.

Página	Tempo	R	M
0	10	1	1
1	30	0	1
2	40	1	1
3	80	1	0

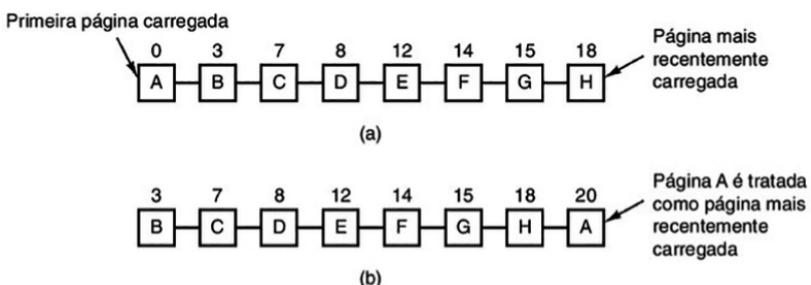
Como o algoritmo FIFO desconsidera as páginas que foram referenciadas na última utilização, independentemente de sua importância, a página 0 seria removida.

- **Algoritmo de substituição de página de segunda chance (SC)**

O algoritmo FIFO não leva em consideração a retirada de uma página usada constantemente. Segundo Tanenbaum (2003), uma alteração simples evitaria a eliminação de uma página muito usada. Basta verificar o bit de referência da página mais antiga. Se o bit R for 0, a página não está sendo usada e pode ser substituída. Se o bit R for 1 será colocado

em 0, e essa página será inserida no final da lista como se ela tivesse acabado de ser carregada na memória. A Figura 4.11(a) apresenta uma lista com as páginas de A até H ordenadas por ordem de chegada na memória (a página A chegou no instante 0, a página B no instante 3, a página C no instante 7, e assim por diante). Caso ocorra uma falta de página no instante 20, o bit R da página A (que é a mais antiga) é verificado. Se o valor do bit R for 0 (zero) a página é retirada da memória, caso contrário A será inserida no final da fila no instante 20 (Figura 4.11(a)) e R receberá o valor 0. O algoritmo continua a percorrer a lista a partir da página B para encontrar uma página que possa ser substituída.

Figura 4.11 | Algoritmo segunda chance

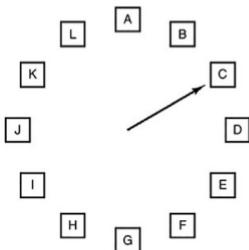


Fonte: Tanenbaum (2003, p. 162).

- **Algoritmo de substituição de página relógio**

Este algoritmo mantém todas as páginas em uma lista circular em forma de relógio e um ponteiro apontando para a 'cabeça' da lista, ou seja, para a página mais antiga. A Figura 4.12 apresenta o funcionamento do algoritmo, (TANENBAUM, 2003). Quando ocorre uma falta de página, o ponteiro aponta para a página mais antiga e o bit R é verificado (página C). Se o valor do bit R for 0 (zero) a página é retirada da memória e o ponteiro avança uma posição. Caso contrário, R receberá o valor 0 e o ponteiro avançará para a página mais antiga. A diferença entre o algoritmo SC para o algoritmo relógio está na implementação. O algoritmo SC é implementado através de fila e o relógio através de uma lista circular.

Figura 4.12 | Algoritmo de substituição de página relógio



Fonte: Tanenbaum (2003, p. 162).

- **Algoritmo de substituição de página menos recentemente utilizada (MRU)**

Segundo Tanenbaum (2003), este algoritmo é baseado na observação de que páginas referenciadas intensamente nas últimas instruções provavelmente serão novamente utilizadas, e páginas que não foram referenciadas não serão utilizadas na próxima instrução. Este algoritmo aproxima-se do desempenho do algoritmo ótimo e possui uma implementação onerosa, pois mantém uma lista encadeada na memória com as páginas mais utilizadas no início da lista e as menos utilizadas no final, sendo necessário a sua atualização a cada referência de memória.



Assimile

Para melhor assimilação dos algoritmos de substituição de páginas, segue um resumo apresentando as características de cada um:

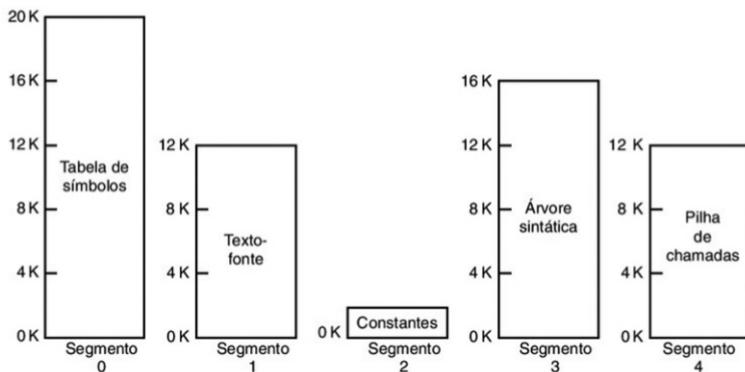
- O algoritmo ótimo possui um bom desempenho, porém não é implementável.
- O algoritmo NUR é simples e de fácil implementação.
- Já o algoritmo FIFO não leva em consideração páginas que são constantemente utilizadas.
- O algoritmo SC é o algoritmo FIFO melhorado.
- O algoritmo relógio é realista e o algoritmo MRU é um bom algoritmo porém difícil de ser implementado.

Segmentação

Segundo Tanenbaum (2003), a memória virtual é unidimensional, uma vez que inicia da posição 0 e vai até um endereço máximo. Em alguns casos isso pode ser um problema. Por exemplo, durante a execução de um compilador são gerados o texto fonte pré-processado, uma tabela de símbolos, uma tabela de constantes, uma árvore sintática e uma pilha. O texto fonte, a tabela de símbolos, a tabela de constantes e a árvore sintática crescem continuamente durante a compilação, enquanto a pilha varia de tamanho. Se um programa possui um número grande de variáveis, o espaço reservado para elas na tabela de símbolos pode se esgotar à medida que o compilador é executado e sobrará espaço nas outras tabelas.

Para resolver este problema, basta fornecer ao computador vários espaços de endereçamento independentes chamados de segmentos. O segmento é uma unidade lógica que pode ser um vetor ou uma pilha, por exemplo, sendo de conhecimento do programador. Cada segmento tem um tamanho dinâmico e independente dos outros (que varia de 0 a um valor máximo), conforme apresentado na Figura 4.13, permitindo que o segmento aumente ou diminua durante a execução. Os endereços são compostos pelo número do segmento e um deslocamento dentro do segmento.

Figura 4.13 | Memória segmentada



Fonte: Tanenbaum (2003, p. 187).

A segmentação facilita o compartilhamento de procedimentos ou dados entre vários processos, por exemplo, o uso de uma biblioteca compartilhada, evitando que estas bibliotecas sejam

replicadas em cada endereçamento de cada processo. Segmentos podem ter diferentes proteções, como um segmento que armazena código pode ser definido apenas para execução, um segmento que armazena constantes, somente leitura e um segmento de pilha como leitura/escrita. As tentativas de acesso indevido podem ser capturadas e tratadas pelo hardware ou pelo sistema operacional.



Refletia

Levando em consideração os computadores atuais em relação a quantidade de memória principal disponível, qual a melhor técnica a ser utilizada: paginação ou segmentação? Em relação à proteção qual a complexidade de implementação da segmentação? E da paginação?

Para finalizarmos, seguem as principais diferenças entre paginação e segmentação:

Quadro 4.1 | Critérios de segmentação e paginação

	Segmentação	Paginação
O programador sabe do seu uso?	Sim	Não
Endereços lineares existentes?	Vários	Um
O tamanho dos dados podem exceder a quantidade de memória física?	Sim	Sim
Facilidade de uso entre usuários?	Sim	Não

Fonte: elaborado pela autora (2018).

Pronto! Agora você já conhece os principais aspectos acerca da memória virtual! Encerramos aqui nossa disciplina e esperamos que a tenha aproveitado ao máximo! Em caso de dúvidas, não deixe de consultar os conteúdos já estudados!

Sem medo de errar

Agora que você já aprendeu sobre os conceitos de memória virtual, paginação e segmentação, vamos relembrar o nosso contexto. Durante a feira, um proprietário de uma loja de jogos digitais está à procura de ajuda para resolver o problema de baixa memória virtual dos computadores de sua loja e procura o seu

stand. Diante dos problemas relatados por ele, surgem os seguintes questionamentos: o que aconteceu no computador neste caso? Qual a solução que você dará para este problema? Quais são os algoritmos de substituição de páginas mais eficientes em relação ao desempenho do sistema?

Neste caso, quando a mensagem de memória virtual baixa aparece é porque vários programas estão executando e consumindo memória, e o computador está no limite do seu uso, ou seja, pouca memória física disponível para processar o número de processos. Em alguns casos, ao iniciar o computador, nenhum processo consegue ser executado, e todos os programas abertos pelo usuário são fechados, uma vez que, dependendo da quantidade de memória e se existirem programas pesados sendo executados, isso irá acontecer. A memória virtual, por ser um arquivo e funcionar como uma extensão da memória RAM, deve ser configurado de forma a atender a falta de memória física. O recomendado é que o arquivo de memória virtual tenha 1,5 o tamanho da memória física do computador. O tamanho do arquivo pode ser alterado manualmente para corrigir os problemas de travamento nos sistemas operacionais Linux e Windows.

Os algoritmos de substituição de páginas têm o objetivo de selecionar as páginas com as menores chances de serem referenciadas (utilizadas) no futuro, e escolher uma página a ser removida da memória a fim de liberar espaço para uma nova a ser trazida. Quanto menor for o tempo gasto com as recargas de páginas, mais eficiente será o algoritmo. A escolha do melhor algoritmo a ser utilizado dependerá dos pré-requisitos definidos pelo *kernel* do sistema operacional. O algoritmo ótimo possui um bom desempenho, porém não é implementável. O algoritmo NUR é simples e de fácil implementação. Já o algoritmo FIFO não leva em consideração páginas que são constantemente utilizadas. O algoritmo SC é o algoritmo FIFO melhorado. O algoritmo relógio é realista e o algoritmo MRU é um bom algoritmo, porém difícil de ser implementado.

O relatório com os anteprojetos elaborados durante a feira e entregue para o gestor levou em consideração a necessidade de cada cliente e a solução proposta para eles. Para o cliente que possui pouca memória física em seu computador e não conseguem

processar os dados da empresa foi sugerido ao cliente a aquisição de mais memória RAM para suportar o volume de dados. Mesmo aumentando a quantidade de memória da empresa se as máquinas não conseguirem processar os dados corretamente, será necessário trocar o computador para um mais potente. Em relação ao estudante de tecnologia que questionou sobre o uso do swap que deixava a máquina lenta, foi sugerido que o swapping não pode ser utilizado como solução para a questão da falta de memória RAM. Assim ele deve adquirir mais memória e reduzir a área de swap para melhor o desempenho do computador. Em relação ao cliente com a mensagem de memória virtual baixa o mesmo foi orientado a aumentar o tamanho do arquivo de memória virtual para auxiliar a execução dos programas.

Avançando na prática

Paginação – falta de página

Descrição da situação-problema

A empresa CCAD LTDA do ramo de serviços financeiros resolveu modernizar os seus computadores e adquiriu máquinas potentes para terem um maior poder de processamento de suas transações financeiras. A empresa oferece aos seus clientes um considerável ganho financeiro e de tempo, uma vez que os clientes não precisam investir em funcionários para gerenciar a área financeira de suas empresas (como a parte contábil). Você foi contratado como supervisor de infraestrutura para realizar a troca das máquinas e dar suporte aos usuários da empresa. Antes de efetuar a troca dos computadores, você observa que durante uma transação financeira de um dos clientes, um programa tenta acessar uma página que não está mapeada na memória RAM, ocasionando em uma falta de página. Porque a falta de páginas acontece? Como fazer para resolver esta situação?

Resolução da situação-problema

A falta de página acontece quando um programa tenta acessar uma página que não está mapeada na memória RAM. O sistema operacional verifica se tem alguma moldura de página disponível e

substitui pela página que não está mapeada. Se na memória física não tiver espaço disponível para realizar a troca é necessário escolher um algoritmo de substituição de página para definir os critérios para remover da memória uma página, liberando espaço para que uma nova possa ser trazida. O gerenciador de memória atualiza milhares de vezes por segundo a tabela de páginas, atualizando as posições de memória dos endereços. Como a empresa está em processo de troca de máquinas, este problema não irá acontecer novamente, uma vez que as máquinas adquiridas possuem memória suficiente para processar as transações.

Faça valer a pena

1.

A memória virtual é uma técnica pela qual é possível utilizar uma parte do disco rígido como se fosse memória RAM, assim sendo, o sistema operacional é capaz de executar aplicações mesmo que a soma de todos os programas em execução simultânea supere a da memória RAM instalada no computador". (SALES, 2008)



Assinale a alternativa correta sobre memória virtual:

- a) Os programas geram endereços virtuais constituindo o espaço do endereçamento virtual e nos computadores que não possuem memória virtual, o endereço virtual e o endereço físico são diferentes.
- b) A utilização da memória virtual permite o compartilhamento da memória física entre vários aplicativos ao mesmo tempo e todos os programas enxergam uma versão virtual geral da memória.
- c) A paginação é uma técnica de memória virtual criada para fornecer um espaço de endereçamento linear sem a precisar comprar mais memória física.
- d) Nos computadores seria mais simples se fosse utilizado apenas a memória virtual, pois além da agilidade de acesso aos arquivos economizaria com a compra de memória RAM.
- e) A MMU é a responsável por mapear os endereços físicos em endereços virtuais e enviá-los para a memória.

2. Suponha que um computador tenha quatro molduras de página. O tempo em que a página é carregada em memória, o último acesso a memória (última referência) e os valores presentes nos bits R e M para as páginas de 0 a 3 são apresentados a seguir (TANEMBAUM, 2003). Se utilizarmos os algoritmos segunda chance (SC) e página não recentemente utilizada (NUR) aplicada a este problema, quais as páginas que serão substituídas?

Página	Carregado	Última referencia	R	M
0	126	280	1	0
1	230	265	0	1
2	140	270	0	0
3	110	285	1	1

Assinale a alternativa que contém a resposta correta sobre as páginas que serão substituídas pelos algoritmos SC e NUR:

- a) NUR substitui a página 2 e SC substitui a página 1.
- b) NUR substitui a página 3 e SC substitui a página 0.
- c) NUR substitui a página 0 e SC substitui a página 3.
- d) NUR substitui a página 2 e SC substitui a página 2.
- e) NUR substitui a página 1 e SC substitui a página 2.

3. Em relação aos conceitos de segmentação, analise as afirmações a seguir.

- I) O segmento é uma unidade lógica que pode ser um vetor ou uma pilha, por exemplo, sendo de conhecimento do programador.
- II) A segmentação facilita o compartilhamento de procedimentos ou dados entre vários processos, como o uso de uma biblioteca individual para cada processo, para agilizar o acesso aos dados.
- III) O programador sabe da existência de cada segmento e pode utilizá-lo como uma unidade lógica.
- IV) A principal diferença entre a segmentação e a paginação é que as páginas possuem um tamanho fixo e os segmentos não.

Assinale a alternativa correta que define os conceitos sobre segmentação:

- a) I, II, III estão corretas.
- b) I, III, IV estão corretas.
- c) II, III, IV estão corretas.
- d) I, II estão corretas.
- e) III e IV estão corretas.

Referências

- BARTLETT, D. **Por dentro do gerenciamento de memória.** Disponível em: <<https://www.ibm.com/developerworks/br/library/l-memory/l-memory-pdf.pdf>>. Acesso em: 14 jun. 2018.
- DAQUINO, F. **Adicionar memória RAM sempre vai deixar o meu PC mais rápido?** Disponível em: <<https://www.tecmundo.com.br/memoria-ram/48168-adicionar-memoria-ram-sempre-vai-deixar-o-meu-pc-mais-rapido-.htm>>. Acesso em: 22 jun. 2018.
- FILHO, R. G. N. **Hierarquia de memórias.** Disponível em: <<http://www.di.ufpb.br/raimundo/Hierarquia/Hierarquia.html>>. Acesso em: 11 jun. 2018.
- GUGIK, G. **Como funciona a memória virtual?** Disponível em: <<https://www.tecmundo.com.br/internet/2190-como-funciona-a-memoria-virtual-.htm>>. Acesso em: 28 jun. 2018.
- JUNIOR, L. A. P. L. **Gerência de memória.** Disponível em: <http://www.ppgia.pucpr.br/~laplima/ensino/so/materia/03_memoria.html>. Acesso em: 12 jun. 2018.
- KLEINA, N. **Como funcionam as diferentes memórias quando o computador está em uso.** Disponível em: <<https://www.tecmundo.com.br/armazenamento/9415-como-funcionam-as-diferentes-memorias-quando-o-computador-esta-em-uso-.htm>>. Acesso em: 12 jun. 2018.
- MACHADO, F. B.; MAIA, L. P. **Arquitetura de sistemas operacionais.** 4. ed. Rio de Janeiro: LTC, 2007.
- MORIMOTO, C. E. **Hardware para iniciantes, parte 3.** Disponível em: <<https://www.hardware.com.br/tutoriais/hardware-iniciantes-3/pagina3.html>>. Acesso em: 20 jun. 2018.
- SALES E. **Memória virtual.** Disponível em: <https://www.oficinadanet.com.br/artigo/730/memoria_virtual>. Acesso em: 9 jul. 2018.
- SILBERSCHATZ, A.; GALVIN, P. B.; GAGNE, G. **Fundamentos de sistemas operacionais.** 9. ed. Rio de Janeiro: LTC, 2015.
- SIMIONI, D. **O que é a memória SWAP?** Disponível em: <<https://www.diolinux.com.br/2014/09/o-que-e-memoria-swap.html>>. Acesso em: 21 jun. 2018.
- SOUZA, P. S. L.; BRUSCHI, S. M. **Por que estudar a hierarquia de memória?** Disponível em: <<http://amnesia.lasdpc.icmc.usp.br/hierarquia-de-memoria/>>. Acesso em: 12 jun. 2018.
- TANENBAUM, A. S. **Sistemas operacionais modernos.** 2. ed. São Paulo: Pearson, 2003.

ISBN 978-85-522-1177-8



9 788552 211778 >