



Fundamentos e Gestão de Sistemas Operacionais

Marcelo Okano
Agnaldo da Costa
Ranieri Santos

Curitiba
2016

O41f Okano, Marcelo

Fundamentos e gestão de sistemas operacionais / Marcelo Okano,
Agnaldo da Costa, Ranieri Santos. - Curitiba: Fael, 2016.

176 p.: il.

ISBN 978-85-60531-54-7

1. Sistemas operacionais I. Costa, Agnaldo da II. Santos, Ranieri III.

Título

CDD 005.42

Direitos desta edição reservados à Fael.

É proibida a reprodução total ou parcial desta obra sem autorização expressa da Fael.

FAEL

Direção de Produção	Fernando Santos de Moraes Sarmento
Coordenação Editorial	Raquel Andrade Lorenz
Revisão	FabriCO
Projeto Gráfico	Sandro Niemicz
Capa	Vitor Bernardo Backes Lopes
Imagem Capa	Shutterstock.com/Rainbow-Pic
Diagramação	FabriCO
Arte-Final	Evelyn Caroline dos Santos Betim

Sumário

CARTA AO Aluno | 5

1. INTRODUÇÃO AOS Sistemas Operacionais | 7

2. INSTALAÇÃO | 19

3. PROCESSOS | 43

4. MEMÓRIA | 59

5. SISTEMAS DE Arquivos | 75

6. ENTRADA E saída (I/O) | 89

7. GERENCIAMENTO DE Usuários | 105

8. SEGURANÇA | 123

9. DESEMPENHO | 139

10. SISTEMAS OPERACIONAIS do Mercado,
Aplicabilidade e Aplicativos de Gestão | 155

CONCLUSÃO | 169

REFERÊNCIAS | 171



Carta ao Aluno

PREZADO ALUNO,

Os dispositivos móveis como celular, GPS, *tablets* etc., estão presentes no nosso dia a dia, e o *software* que controla e gerencia estes dispositivos é conhecido como sistema operacional, que conta com diversos tipos, fabricantes e aplicações.

Os sistemas operacionais equipam desde os dispositivos móveis até os maiores servidores e computadores que existem. Esta disciplina propiciará um entendimento sobre os conceitos principais, seu funcionamento, tipos e gerenciamento, permitindo que você compare os principais sistemas operacionais do mercado, selecionar os aplicativos de gestão de sistemas operacionais e instalar e gerenciar um sistema operacional.

Estes conhecimentos possibilitarão a você atuar com servidores e computadores de vários portes e sistemas.

- × Entender os principais conceitos para o funcionamento de um sistema operacional;
- × Comparar os principais sistemas operacionais do mercado;
- × Selecionar os aplicativos de gestão de sistemas operacionais;
- × Instalar e gerenciar um sistema operacional.

1

Introdução aos Sistemas Operacionais

NOS DIAS DE hoje, para um profissional de TI, não adianta conhecer os sistemas operacionais como meros usuários domésticos, é preciso estender estes conhecimentos para saber gerenciá-los e utilizá-los profissionalmente, fazer com que o equipamento tenha o máximo de desempenho.

A diversidade de equipamentos tecnológicos que utilizam os sistemas operacionais, desde celulares até servidores de grande porte, exigem que os profissionais de TI saibam como escolhê-los para atingirem uma maior eficiência.

Objetivos de aprendizagem

- × Entender os conceitos sobre sistemas operacionais, história, tipos e componentes;
- × Diferenciar os diversos tipos de sistemas operacionais;
- × Selecionar o melhor para cada tipo de uso.


1.1 Conceitos de sistema operacional

O sistema operacional é o conjunto de programas que controla, gerencia e opera um dispositivo computacional. Este dispositivo pode ser um computador, servidor, *tablet*, etc.



Importante

○ sistema operacional é o conjunto de programas que controla, gerencia e opera um dispositivo computacional.



Outras definições sobre sistema operacional:

- × É o único programa que fica o tempo todo em execução no computador (normalmente chamado de kernel) com todos os outros sendo programas do sistemas e aplicativos (SILBERSCHATZ& GALVIN, 2008).
- × Consiste em uma camada de *software* que oculta o hardware e fornece ao programador um conjunto de instruções mais adequado. Como o sistema operacional está localizado entre os aplicativos e o *hardware*, a única maneira dos aplicativos acessarem o *hardware* é por meio de chamadas às funções do sistema operacional (TANENBAUM, 2009; MOROZ, 2011).
- × É o programa que gerencia e coordena o acesso aos dispositivos de *hardware*. Caso mais de um processo queira acessar um mesmo dispositivo, cabe ao sistema operacional coordenar e permitir o acesso a apenas um processo de cada vez (TANENBAUM, 2009; MOROZ, 2011).

1.2 História

Os sistemas operacionais nasceram com a evolução do *hardware* e a necessidade de realizar cálculos cada vez mais rápidos e precisos. Foi essa necessidade que acelerou o processo de evolução dos sistemas operacionais. A funcionalidade dos sistemas operacionais antigos não é nada parecida com a dos sistemas operacionais modernos, com telas bonitas e de fácil utilização. Até chegarmos à variedade de sistemas operacionais que temos hoje em nosso mercado, o processo foi longo e acompanhado de muitas descobertas.

De acordo com TANENBAUM (2009), a história dos sistemas operacionais pode ser classificada em 4 gerações: válvulas e painéis, transistores e sistemas batch, circuitos integrados (CIs) e multiprogramação, e computadores pessoais.

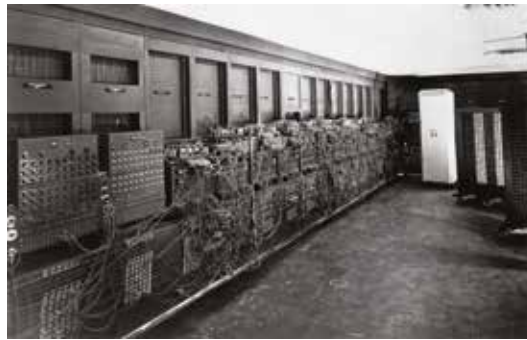
× **Primeira Geração (1945-1955):** válvulas e painéis com plugs.

Os primeiros computadores inventados não contavam com um conceito de sistemas operacionais, ou seja, o programa ou *software* não estava embutido na máquina para gerenciar seu *hardware*, pois todas as informações ainda eram configuradas no *hardware* da máquina. Eram quilômetros de fios e muitos programadores para a realização de pequenos cálculos controlados por *plugs*, que eram adaptados e configurados para gerar informações de acordo com a programação necessária. Era uma tarefa muito difícil de criação de rotinas para a realização das tarefas, pois nessa época nem a linguagem *Assembly* havia sido criada. Um outro problema eram as válvulas que queimavam e comprometiam os cálculos. Além disso, sua troca exigia um esforço diário para que o computador pudesse funcionar de forma precisa. O computador ENIAC, que em português significa computador integrador numérico eletrônico, possuía 17.468 válvulas.

Linguagem *Assembly*: *Assembly* ou linguagem de montagem é uma notação legível e entendido por humanos para gerar códigos de máquina, que uma arquitetura de computador específica usa para realização de cálculos.

Podemos observar na Figura 1 o computador ENIAC, um dos primeiros computadores eletrônicos, e sua quantidade de *plugs* e válvulas, necessários para sua operação. Sem um sistema operacional capaz de gerenciar o *hardware*, a tarefa era realizada pelos programadores.

Figura 1: Computador da primeira geração.



Fonte: Shutterstock (2016).

Saiba mais

Podemos aprender mais sobre o processo e a evolução dos Sistemas Operacionais. Para isso acesse o site <https://www.youtube.com/watch?v=Zl9w2HbUecU&list=PLWJVgHJ6b_aF6jWGcAVIURWqCvk_671aU>, onde poderá encontrar mais detalhes sobre essa evolução.

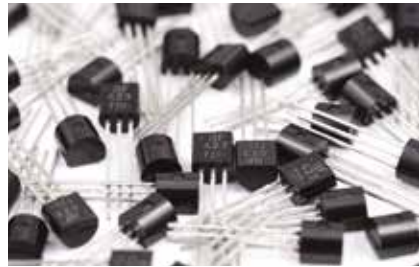
× Segunda Geração (1955 - 1965): transistores e sistemas batch

A descoberta dos transistores mudou radicalmente a forma como as operações computacionais eram realizadas, pois deram velocidade ao processamento dos dados. As válvulas eram mais lentas e queimavam com facilidade com o advento da tecnologia de transistores. O termo sistema operacional aparece por meio de um método de programação denominada de Batch, que consistia em vários comandos que poderiam ser executados em sequência por meio de cartões perfura-

dos, eliminando das parte do trabalho do operador de terminal a quantidade de pessoas necessárias para realizar rotinas no *hardware*.

O sistema de processamento em lote constituía-se no armazenamento prévio de diversos *Jobs* (trabalhos) para ser processado, sequencialmente, no computador, quando um *Jobs* realizava sua tarefa. Ou seja, quando terminava de utilizar o *hardware* da máquina, sequencialmente, outro *Jobs*, era executado. Por isso, o nome de Batch, arquivos em Lote. Essa tarefa demandava tempo e recurso de *hardware* e a ação humana sobre o processo de gerenciamento das informações ainda era latente.

Figura 2 - Transistores usados nos computadores da segunda geração.



Fonte: Shutterstock (2016).

Transistores: O termo provém do inglês *transfer resistor* (resistor/resistência de transferência). São utilizados principalmente como amplificadores e interruptores de sinais elétricos, além de retificadores elétricos em um circuito

Jobs: O termo provém do inglês, que significa trabalho

Saiba mais

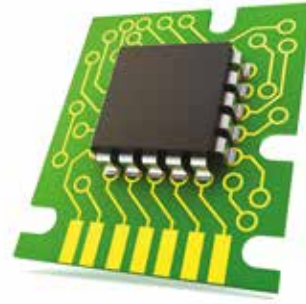
O transistor foi uma descoberta dos Laboratórios da Bell Telephone, por John Bardeen e Walter Houser Brattain em 1947, e revolucionou a área da computação. Nesse pequeno vídeo podemos entender como ele funciona: < <https://www.youtube.com/watch?v=Xsv03w9YJqI>>

- × **Terceira Geração (1965 - 1980):** circuitos integrados (CIs) e multiprogramação

A grande conquista desta geração foi a capacidade dos sistemas operacionais ler os cartões que continham as programações para o

disco e o aumento da capacidade de realização de grandes cálculos científicos e de processamento de dados. O conceito de processar as informações em forma de lote ainda estava vigente nessa geração. Mas surge nesse contexto o termo de multiprogramação, vários programas sendo carregados na memória da máquina e compartilhada com os usuários. O que começa a dar forma aos sistemas operacionais tradicionais, que podiam controlar quais programações deveriam ser executadas. Outro destaque desta geração que contribuía para a evolução dos sistemas operacionais foram a substituição dos transistores pelos (CIs), circuitos integrados, que forneciam uma vantagem maior de processamento em relação à segunda geração.

Figura 3 - Circuito integrado (CI) usado na terceira geração de computadores



Fonte: Shutterstock (2016).

Importante

O conceito de multiprogramação é um dos mais importantes nos sistemas operacionais modernos. Se existirem vários programas carregados na memória ao mesmo tempo, a CPU pode ser compartilhada entre eles, aumentando a eficiência da máquina e produzindo mais resultados em menos tempo.

Saiba mais

Para atender a necessidade de multiprogramação surgiram alguns sistemas operacionais. Ken Thompson reescreveu o *Multics* em *assembly* e batizou seu projeto de *Unics*. Em 1973, o próprio Ken Thompson em conjunto com Dennis Ritchie reescreve o Unix em linguagem C. Em 1976, Steve Jobs, o guru da Apple, teve uma ideia que revolucionou o mundo, um computador pequeno, portátil e barato o

suficiente para que qualquer pessoa pudesse ter acesso, o Apple 2.

Em 1980, a fabricante de computadores de grande porte, a IBM (International Business Machines) decide entrar para o mercado dos computadores pessoais, mas não possuía nenhum programa para rodar nele. A IBM fechou contrato com a Microsoft de Bill Gates para ela fornecer o Sistema Operacional. Surge o MS-dos, que inicialmente executava um programa por vez.



× **Quarta Geração (1980-1990): computadores pessoais**

Nesta geração houve a consolidação do mercado dos computadores pessoais e os Sistemas Operacionais tornaram-se mais amigáveis, com suas telas interativas, dando início a interação do homem e da máquina, a ergonomia de *software*, *layout* de telas, etc. Essa evolução veio acompanhada da capacidade computacional disponível, especialmente, a capacidade de computação com o desenvolvimento de *chips* contendo milhares de transistores em um centímetro quadrado de silício.

Paralelo a esse crescimento, surgiu a popularização da internet e as redes de computadores, que aumentaram as aplicações dos sistemas operacionais de rede e sistemas operacionais distribuídos. Na área de *hardware* houve um aumento da capacidade de gerenciar dispositivos de entrada e saída (teclado, mouses, monitores, impressoras, etc)

Figura 4 : Primeiros computadores pessoais



Fonte: Shutterstock (2016).

Em um sistema operacional de rede, podem ser conectados múltiplos computadores de forma física e remotas, que compartilham processos, arquivos, recursos etc.

A capacidade de gerenciar dispositivos de entrada de dados, memórias principal, memória de massa e processamento e a evolução das interfaces são características desta geração. Destacam-se nessa geração os Sistemas Operacionais como o computador pessoal: o MS-DOS, escrito pela Microsoft para o IBM PC e o UNIX, que é predominante em máquinas que usam a CPU da família Motorola.



Você sabia

Em 1984 Steve Jobs rouba da Xerox, como ele mesmo admite, a ideia de um sistema operacional baseado em objetos clicáveis com um mouse. Até esse período o gerenciamento dos sistemas operacionais era realizado por meio de comandos utilizando o teclado. No mesmo ano, Richard Stallman começa a desenvolver o projeto GNU, que deveria possuir as mesmas características do Unix, sem aproveitar sua plataforma, e surge a Filosofia de software livre.

Em 1986, a Microsoft lança o Windows 1, um aplicativo de janelas que rodava em cima do sistema operacional MS dos.

Cinco anos depois, em 1991, Linus Torvalds inicia o desenvolvimento do Linux, recebe apoio de milhares de programadores ao redor do mundo e grandes empresas como IBM, Sun Microsystems, Hewlett-Packard (HP), Red Hat, Novell, Oracle, Google, Mandriva e Canonical.



1.3 Tipos de sistemas operacionais

Com o avanço da tecnologia da informação e, conseqüentemente, com a popularização dos dispositivos computacionais, os sistemas operacionais se tornaram comuns entre os usuários, seja no seu computador pessoal, no celular, sistemas embarcados ou em grandes servidores temos algum tipo de sistema operacional.

Podemos tipificar os sistemas operacionais da seguinte forma:

- × **Sistemas operacionais de servidores:** servidores são todos e qualquer computador que ofereça um ou mais serviços, podendo ser

um gerenciador de banco de dados, aplicativos, servidor de páginas de internet, e-mail, autenticação, etc. Estes serviços precisam de um sistema operacional robusto para funcionarem, que tenha características próprias como suportar uma grande quantidade de recursos como discos, processadores, memórias e usuários, interconexão de redes e recursos de segurança. Podemos citar o VMS para os Mainframes, Solaris, Unix, Linux e Windows Server 2012.

Figura 5: Sistema Operacional Linux



Fonte: Shutterstock (2016).

- × **Sistemas operacionais de dispositivos móveis:** é crescente o uso dos dispositivos móveis e, cada vez mais rotineiros e usuais, como os celulares, *tablets*, GPS, etc. Os sistemas operacionais para estes dispositivos têm algumas características como manipulação de fotos e vídeos, redes sociais e telefonia. Os principais sistemas são o Android e o IOS.

Figura 6 : Sistema Operacional Android



Fonte: Shutterstock (2016).

GPS: é a sigla de “Global Positioning System” que significa sistema de posicionamento global. GPS é um sistema de navegação por satélite com um aparelho móvel que envia informações sobre a posição de algo em qualquer horário e em qualquer condição climática.

Android: é o nome do sistema operacional baseado em Linux que opera em celulares (*smartphones*), *netbooks* e *tablets*.

IOS: (antes chamado de iPhone OS) é um sistema operacional móvel da Apple Inc. desenvolvido originalmente para o iPhone, também é usado em iPod Touch, iPad e Apple TV.

- × **Sistemas operacionais de computadores pessoais:** são os sistemas operacionais que equipam os computadores pessoais como PC e MAC. São utilizados pelos usuários para executar aplicativos como planilhas e editores de texto e navegar na internet. Como exemplos temos o Linux, Windows 7,8 e 10 e o MAC-OS.
- × **Sistemas operacionais embarcados :** são executados em computadores que controlam dispositivos que geralmente não são considerados computadores, e que não aceitam *softwares* instalados por usuários. Exemplos típicos são fornos de micro-ondas, aparelhos de TV, carros, aparelhos de DVD, etc. (TANENBAUM, 2009, p.22).
- × **Sistemas Operacionais de tempo real:** Esses sistemas são caracterizados por terem o tempo como um parâmetro fundamental. Por exemplo, em sistemas de controle de processos industriais, computadores de tempo real devem coletar dados sobre o processo de produção e usá-los para controlar as máquinas na fábrica (TANENBAUM, 2009, p.22).
- × **Sistemas multiusuários:** são sistemas operacionais que permitem que vários usuários trabalhem simultaneamente. A maioria dos atuais são multiusuários como Linux e Windows.
- × **Sistemas monousuários:** são sistemas que executam somente as tarefas de um único usuário. Os primeiros computadores funcionavam desta forma.
- × **Sistemas multiprocessados:** são sistemas que são executados em computadores com vários processadores de forma simultânea. A maioria dos atuais são multiprocessados como Linux e Windows
- × **Sistemas monoprocessados:** são sistemas que são executados em computadores com um único processador.

Algumas funções dos sistemas operacionais:

- × Tratamento de interrupções e exceções;
- × Criação e eliminação de processos: gerenciar a criação ou eliminação de processos, ou seja, alocar na memória os recursos necessários para o funcionamento do processo;

- × Suporte a Redes: Permite interconectar o equipamento a uma rede de computadores e seus serviços;
- × Contabilização do uso do sistema: Registra o uso/consumo de recursos do sistema operacional como uso de cpu, discos, memória, etc.

1.4 Componentes

O sistema operacional é composto pelas seguintes partes:

- × Gerência de processos: tem a função de organizar e escalonar os processos para serem executados no processador;
- × Gerência de memória: gerencia a alocação de memória para os processos do sistema operacional;
- × Gerência de entrada e saída: gerencia os mecanismos de buffers e drivers dos dispositivos;
- × Gerência de sistema de arquivos: responsável pelo gerenciamento dos arquivos, implementando compartilhamento, controles de acesso, etc;
- × Proteção do sistema: mecanismos de proteção do sistema operacional como memória virtual e dois modos de execução.

1.5 Estudo de Caso

Uma microempresa está usando uma solução “caseira” para os seus cinco computadores, uma impressora, a internet compartilhada, em que o compartilhamento dos dados é feito simplesmente disponibilizando as pastas para os demais computadores e os sistemas operacionais são de computadores pessoais. O que você proporia para melhorar este cenário?

O uso de compartilhamento de pastas ou impressoras em um sistema operacional de computador pessoal foi planejado para poucas máquinas, um problema é que se o computador que está compartilhando estiver desligado, ninguém conseguirá usar o recurso.

Uma solução para melhorar este cenário é instalar um servidor para centralizar os dados e impressora, e usar um sistema operacional de servidor, para

uma microempresa, pode ser um Windows Server ou o Linux. As vantagens de ter um servidor é que os dados estarão em um único lugar, evitando a redundância, e não ficará dependente de um computador pessoal para controlar o acesso aos dados por meio de usuários e autenticação e implementar um sistema para *backup* dos dados.

Você Sabia

○ maior supercomputador do mundo, segundo a lista de novembro de 2015 do site top500 (www.top500.org), é o Tianhe-2 (Milky Way-2 em inglês) desenvolvido para *China's National University of Defense Technology* com 32000 processadores Intel Xeon e 1Peta-Byte de memória. ○ sistema operacional é o Linux.

Veja mais em <http://www.top500.org/system/177999>

Resumindo

Neste capítulo, vimos que o sistema operacional tem várias definições que podem variar de autor para autor, mas é importante lembrar que o sistema operacional é o conjunto de programas que controla, gerencia e opera um dispositivo computacional por meio de seus componentes como gerenciamento de memória, gerenciamento de entrada e saída, gerenciamento de processos, proteção do sistema e gerenciamento de arquivos. Os sistemas operacionais existem desde a década de 40 e passaram-se várias gerações.

Os sistemas operacionais estão presentes nos diversos dispositivos como celulares, servidores, desktops, etc. Podemos citar diversos como versões do Windows 7, 8 e 10, Unix, Linux, Mac OS, Android e IOS.

Android: Sistema operacional para celulares e *tablets* desenvolvido pela Google.

IOS: Sistema operacional para celulares e *tablets* da Apple.

Linux: Sistema Operacional livre.

Mainframe: Servidor de grande porte.

Windows: Sistema Operacional da Microsoft.

2

Instalação

ANTIGAMENTE, A INSTALAÇÃO de um sistema operacional demandava um conhecimento aprofundado, pois exigia diversas configurações (*hardware*), mas com o crescimento do mercado de *desktop* e de *hardware*, hoje os sistemas operacionais possibilitam uma instalação fácil, bastando alguns comandos e ações

A primeira etapa para trabalhar com os sistemas operacionais é aprender como eles são instalados nos computadores. Alguns passos são detalhes importantes para a configuração destes sistemas.

Neste capítulo, abordaremos as instalações de dois sistemas operacionais mais utilizados: o Windows 8.1 e o Linux Debian, prestando atenção na área de paginação, configuração de usuário administrador, como o *root* no Linux, as senhas de usuários e a formatação de partições.

Objetivos de aprendizagem:

- × Instalar um sistema operacional.

2.1 Instalando o Sistema Operacional Windows 8.1

Antes da Instalação de um Sistema Operacional algumas ações de segurança devem ser observadas, como a capacidade do disco rígido, pois caso a capacidade não seja suficiente a instalação não procederá. Outra dica é se perguntar se realmente há a necessidade da instalação, pois a maioria dos sistemas operacionais são reinstalados quando há registro por infecções de vírus e incompatibilidade de *hardware*. Verifique como será dividido seu disco rígido e, caso sejam instalados dois sistemas operacionais, observe como será feita essa divisão. Outro ponto importante em relação à segurança é reservar um espaço no disco para dados, separando a pasta de instalação dos sistemas operacionais dos arquivos de dados do usuário.

Figura 1 – Tela inicial do sistema operacional Windows 8.1.



Fonte: Shutterstock, 2016.

Saiba mais

O *root* é um usuário, ou seja, uma conta especial utilizada para administrar todo sistema operacional Linux, assim como o usuário gerado pelo Sistema Operacional Windows chamado de “administrador”, são denominados de superusuários, pois possuem privilégios na administração dos serviços dos SO’s.

Importante

A instalação de um sistema operacional da Microsoft pode ser encontrada em diversos sites na internet. Os passos executados para a instalação do Windows 8.1 neste módulo são baseados nas ins-

truções do site da Microsoft (<http://windows.microsoft.com/pt-BR/windows-8/clean-install>).



1. O primeiro passo é adquirir a versão original do sistema operacional Windows, neste caso, a versão 8.1. Você pode obtê-lo tanto em empresas especializadas ou, em alguns casos, pode vir com o computador. Segundo a Microsoft, há algumas configurações necessárias para a instalação do sistema operacional. Para executar o Windows 8.1 em seu computador, você precisa de:
 - × Processador: 1 GHz ou mais rápido;
 - × RAM: 1 GB (32 bits) ou 2 GB (64 bits);
 - × Espaço em disco: 16 GB (32 bits) ou 20 GB (64 bits).
2. Ligue o computador para que ele seja inicializado, insira a mídia (DVD ou CD) no leitor do computador e desligue o computador.
3. Ligue o computador novamente, caso não seja inicializado pelo DVD, então você deverá:

Se você reiniciar o PC e sua versão atual do Windows for iniciada, talvez seja necessário abrir um menu de inicialização ou alterar a ordem de inicialização nas configurações de BIOS ou UEFI do computador, para que ele seja inicializado a partir da mídia. Para isso, será necessário pressionar uma combinação de teclas (como F2, F12, Delete, Esc etc.) imediatamente após ligar o computador. Para obter instruções de como alterar a ordem de inicialização do computador, consulte a documentação fornecida com o equipamento ou vá para o site do fabricante (Microsoft, 2015).



Saiba mais

Basic Input OutPut System (BIOS) – Sistema básico de entrada e saída – gravada em programa assembler na memória *Complementary Metal Oxide Semiconductor* (CMOS). Seu trabalho é informar ao processador como trabalhar com os dispositivos essenciais para o boot do sistema, escolhendo as opções (Disco, PenDrive, DVD, etc)

de arranque do sistema operacional.

Unified Extensible Firmware Interface – (UEFI) - Versão mais eficiente de configuração de arranque do sistema operacional. Versão gráfica que permite a utilização de *mouse* para configuração.

4. Ao aparecer a tela de instalação do Windows 8, selecione o idioma e as outras preferências e clique em Avançar.

Figura 2 - Tela de opção para selecionar o idioma.



Fonte: elaborado pelo autor, 2016.

5. Na página Digite a chave do produto (*Product Key*) para ativar o Windows, digite a chave do produto.

Esta chave deve estar na mesma caixa do DVD do Windows 8.1 ou em um email de confirmação de compra. Ela tem esta aparência: PRODUCT KEY: XXXXX-XXXXX-XXXXX-XXXXX-XXXXX (Microsoft, 2015).

Saiba mais

Product Key – A chave do produto é um número composto de 25 dígitos (números), que são utilizados para ativar o sistema operacional. Tem como objetivo verificar se este não foi usado em mais computadores do que o permitido no Termo de Licença para Software.

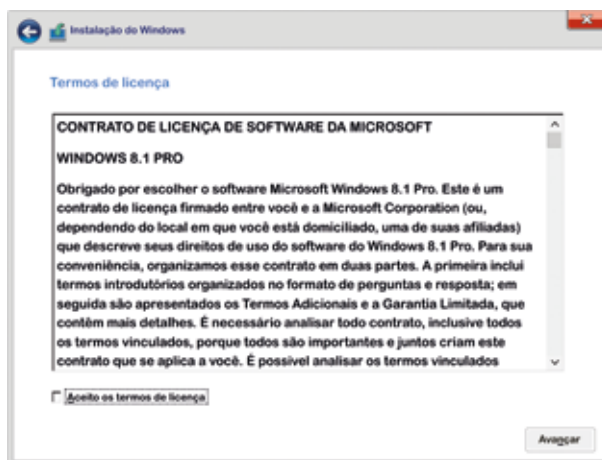
Figura 3 - Tela para inserção da chave do Windows.



Fonte: elaborado pelo autor, 2016.

6. Leia os termos de licença do Windows, se aceitar, marque o item “Aceito os termos de licença” e clique em avançar.

Figura 4 - Tela dos Termos de licença do Windows 8.1.



Fonte: elaborado pelo autor, 2016.

7. Na tela “Que tipo de instalação você deseja?”, selecione a opção Personalizada. A opção de atualização pode ser utilizada quando há problemas no funcionamento do sistema operacional como: arquivos corrompidos. Esta opção permite reinstalar o sistema e corrigir falhas.

Figura 5 -Tela que mostra as opções de instalação.



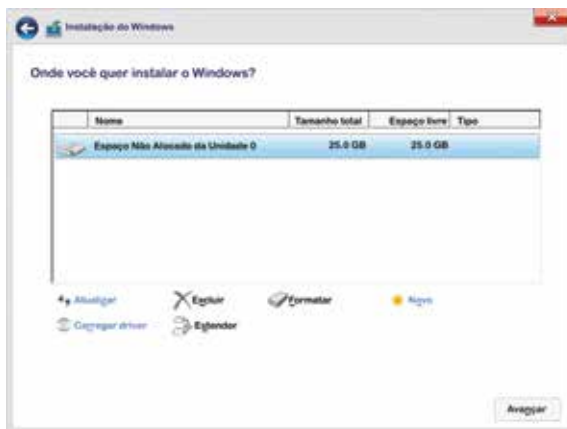
Fonte: elaborado pelo autor, 2016.

8. Na tela “Onde você quer instalar o Windows?”, se for instalar na partição existente, selecione a partição e clique em avançar.

Importante

Uma partição é uma divisão do espaço de um disco rígido. Cada partição pode conter um sistema de arquivos (forma de gerenciar as informações no disco) diferente. Consequentemente, em cada partição pode ser instalado um sistema operacional, sendo possível, portanto, a convivência de vários na mesma unidade de disco, ou você poderá realizar uma divisão dos discos em partes iguais ou diferentes para melhor gerenciar suas informações.

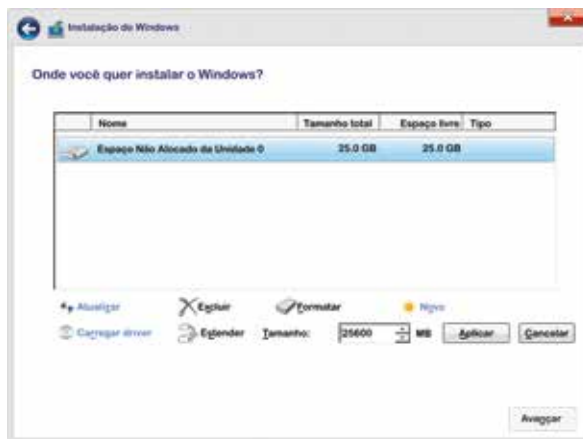
Figura 6 – Opções de particionamento do disco.



Fonte: elaborado pelo autor, 2016.

9. Caso tenha que criar uma nova partição com o tamanho diferente do espaço não alocado, clique em **novo**, selecione o tamanho desejado, clique em **aplicar** e em seguida, selecione **formatar**.

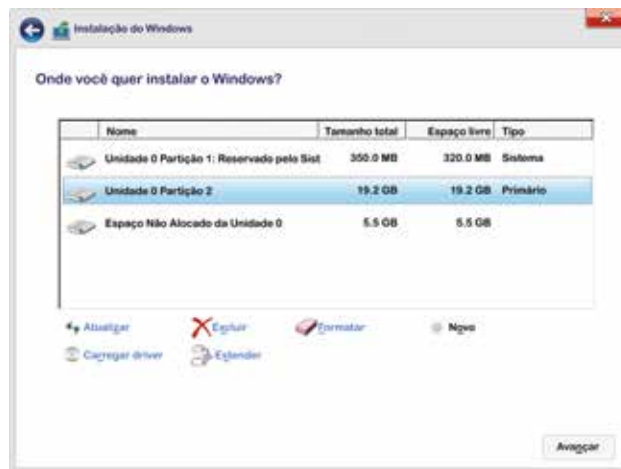
Figura 7 - Tela para formatação de partição nova.



Fonte: elaborado pelo autor, 2016.

10. Selecione a nova partição e clique em avançar.

Figura 8 - Tela para selecionar a nova partição.



Fonte: elaborado pelo autor, 2016.

11. Siga as instruções para personalizar e concluir a configuração do seu Windows. Na figura 9, podemos visualizar as opções de configuração.

Figura 9 – Tela de inicialização do Windows após as configurações.



Fonte: elaborado pelo autor, 2016.

Saiba mais

Você quer obter mais detalhes sobre a instalação do Windows 8.1? A Microsoft possui uma área exclusiva para isso. Acesse: <http://windows.microsoft.com/pt-br/windows-8/clean-install>.

2.2 Instalando o sistema operacional Linux

O Linux, diferente do Windows, é um *software* livre, ou seja seu núcleo pode ser adquirido e mudado, dentro das regras estabelecidas pela comunidade. Para nosso modelo de instalação, utilizaremos a Distribuição Debian.

Você sabia

O mascote do sistema operacional Linux é um pinguim e seu nome é Tux. Em 1996, muitos integrantes da lista de discussão "Linux-Kernel" estavam pensando sobre a criação de um logotipo ou de um mascote que representasse o Linux. Linus Torvalds acabou entrando nesse debate ao afirmar em uma mensagem que gostava muito de pinguins e surgiu o mascote TUX. Para conhecer mais sobre esta história acesse o site: <https://www.vivaolinux.com.br/artigo/Porque-a-mascote-do-Linux-e-um-pinguim>.

1. Você pode fazer *download* do DVD ou CD com a imagem de instalação do Debian no link: <http://cdimage.debian.org/debian-cd/8.2.0/amd64/iso-dvd/> e depois gravar o DVD com a imagem. As configurações necessárias para a instalação são:
 - × computador Pentium 4 1GHz (mínimo recomendado para um sistema desktop);
 - × 64 Bytes de memória;
 - × HD de 1Gbytes.

Figura 10 - TUX- Mascote do LINUX



Fonte: Shutterstock, 2016.

Importante

Nem todo *software* livre pode ser adquirido sem restrições. Existe uma fundação chamada de GNU Licença Pública Geral que regulamenta os *softwares* livre para a comunidade Linux. Leia mais sobre esse importante assunto no site: <http://www.gnu.org/licenses/licenses.pt-br.html>.

2. Ligue a máquina, e no menu, selecione a opção *Install*.

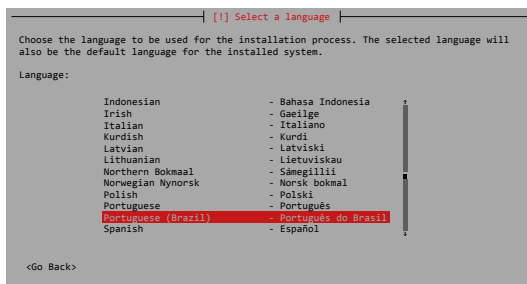
Figura 11- Tela de Inicialização da instalação do Linux.



Fonte: elaborado pelo autor, 2016.

3. Selecione o idioma da instalação “Português do Brasil”, e clique em Continue.

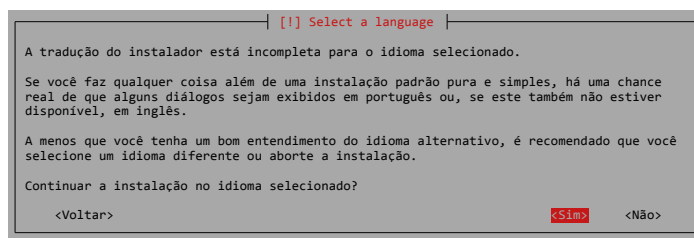
Figura 12 – Opção de escolha de idiomas.



Fonte: elaborado pelo autor, 2016.

4. Ao aparecer a mensagem que a tradução do instalador está incompleta, tecle sim.

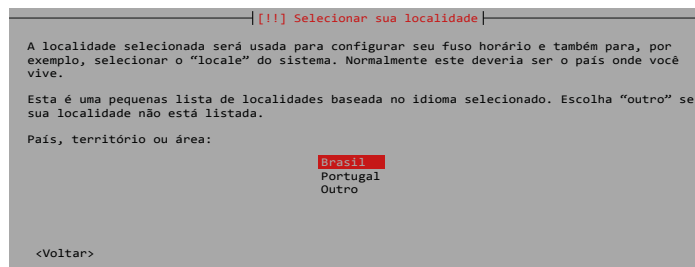
Figura 13 - Tela seleção de linguagem.



Fonte: elaborado pelo autor, 2016.

5. Selecione a localidade “Brasil”.

Figura 14 – Tela para selecionar a localidade.



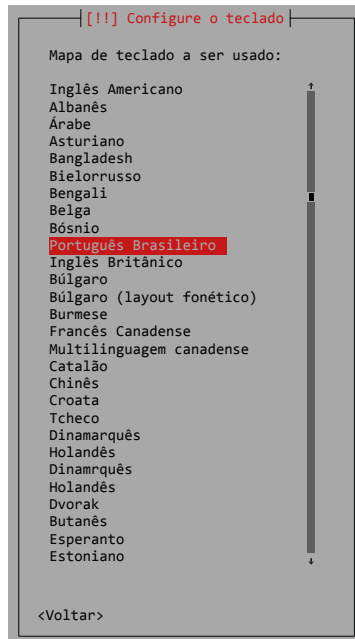
Fonte: elaborado pelo autor, 2016.

Importante

A Associação Brasileira de Normas Técnicas (ABNT) estabelece padrões de dois modelos de teclados utilizados no Brasil: ABNT1 e ABNT2. Ambos utilizam a tecla “Ç”, mas o ABNT2 apresenta uma tecla a mais: a Alt Gr.

6. Selecione o *layout* do teclado (Português Brasileiro) e tecle continuar.

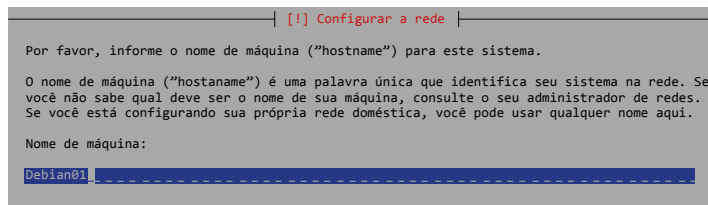
Figura 15 - Menu para seleção do *layout* do teclado.



Fonte: elaborado pelo autor, 2016.

7. Insira o nome da máquina "*hostname*" e tecle em continuar. O "*hostname*" é o nome dado ao computador em que está sendo instalado o sistema operacional. A identificação dessa máquina na rede, caso ela pertencer a uma , será o nome dado nessa configuração

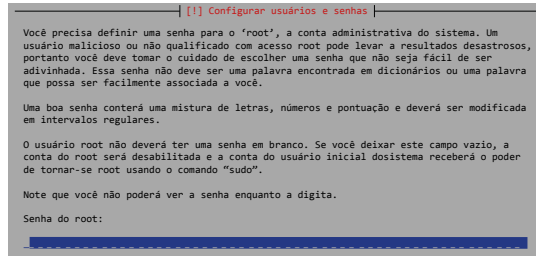
Figura 16 - Menu para inserir o *hostname*.



Fonte: elaborado pelo autor, 2016.

8. Neste item, iremos digitar a senha do *root*. *Root* é o usuário administrador do Linux, por isso mantenha a senha guardada em lugar seguro.

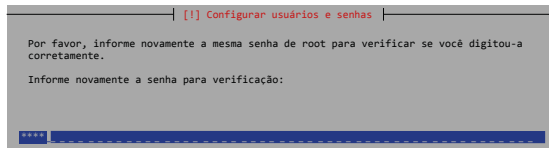
Figura 17 - Configurando a senha do *root*.



Fonte: elaborado pelo autor, 2016.

9. Digite a senha do *root* novamente.

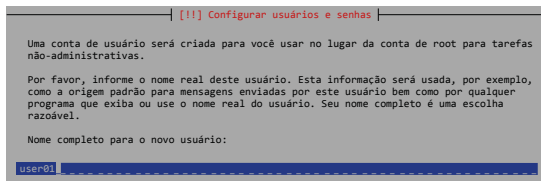
Figura 18- Verificando a senha do *root*.



Fonte: elaborado pelo autor, 2016.

10. Digite o nome completo do usuário, pode ser o seu mesmo. Nessa fase da instalação, estamos criando um usuário para o sistema operacional. A conta desse usuário poderá ser utilizada na inicialização do sistema operacional, após a conclusão de sua instalação.

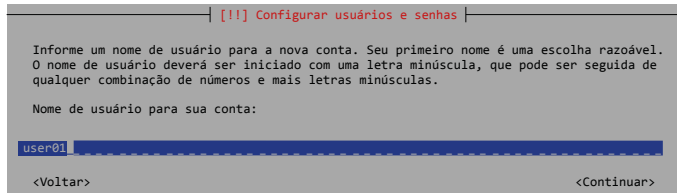
Figura 19 - Configurando um usuário comum.



Fonte: elaborado pelo autor, 2016.

11. Escolha o nome de usuário.

Figura 20 - Configurando o nome do usuário.



[[!]] Configurar usuários e senhas

Informe um nome de usuário para a nova conta. Seu primeiro nome é uma escolha razoável. O nome de usuário deverá ser iniciado com uma letra minúscula, que pode ser seguida de qualquer combinação de números e mais letras minúsculas.

Nome de usuário para sua conta:

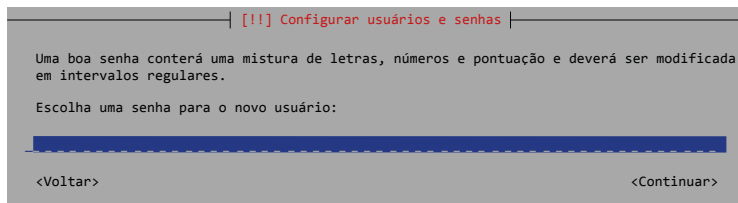
user01

<Voltar> <Continuar>

Fonte: elaborado pelo autor, 2016.

12. Digite a senha para o usuário.

Figura 21- Atribuindo uma senha.



[[!]] Configurar usuários e senhas

Uma boa senha conterá uma mistura de letras, números e pontuação e deverá ser modificada em intervalos regulares.

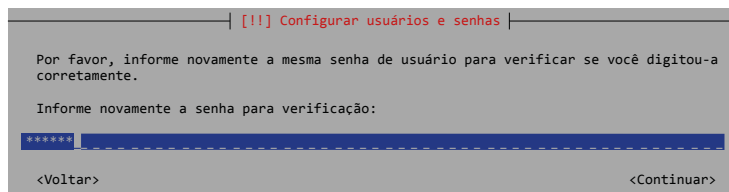
Escolha uma senha para o novo usuário:

<Voltar> <Continuar>

Fonte: elaborado pelo autor, 2016.

13. Digite a senha novamente.

Figura 22- Verificando a senha.



[[!]] Configurar usuários e senhas

Por favor, informe novamente a mesma senha de usuário para verificar se você digitou-a corretamente.

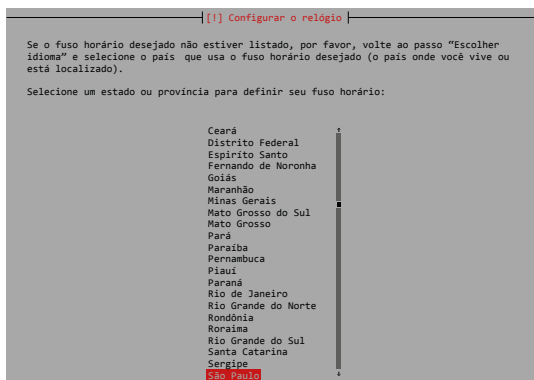
Informe novamente a senha para verificação:

<Voltar> <Continuar>

Fonte: elaborado pelo autor, 2016.

14. Selecione o Estado brasileiro onde o equipamento será utilizado para acertar o fuso horário.

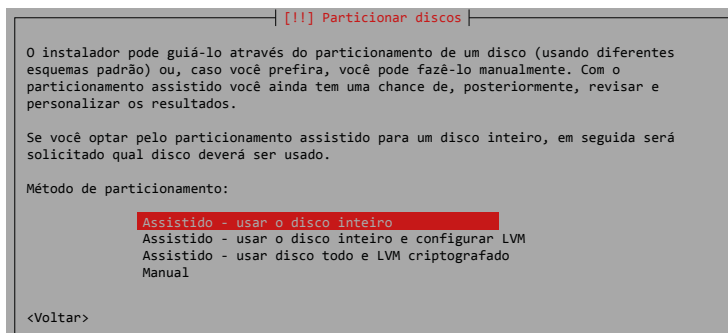
Figura 23- Configurando o relógio.



Fonte: elaborado pelo autor, 2016.

15. Será necessário dividir o disco rígido. Escolha o primeiro método “Assistido - usar o disco inteiro” para simplificar, e clique em Continuar:

Figura 24- Tela divisão do disco.

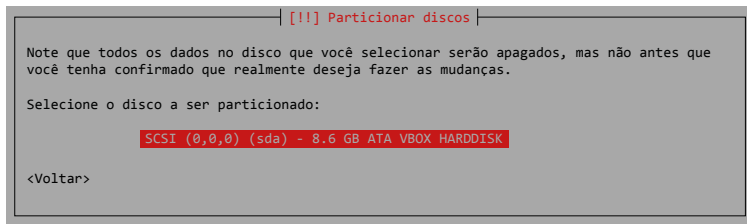


Fonte: elaborado pelo autor, 2016.

16. Na tela seguinte, selecione o disco que será particionado e clique em Continuar. O particionamento é uma técnica utilizada para dividir o discos em dois ou mais espaços para a instalação de mais

de um sistema operacional ou para instalar o sistema operacional em uma parte do disco e os arquivos pessoais em outra.

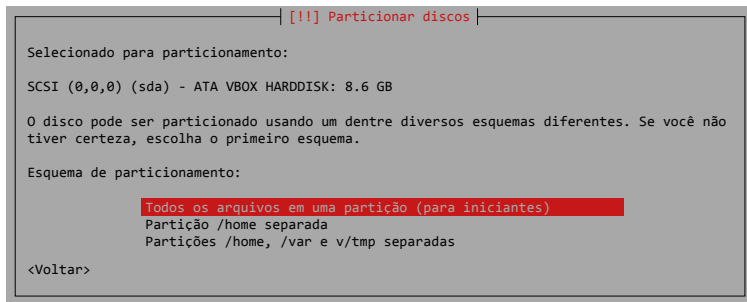
Figura 25- Escolhendo as partições.



Fonte: elaborado pelo autor, 2016.

17. Escolha a segunda opção, que usa a partição */home* separada. Clique em Continuar.

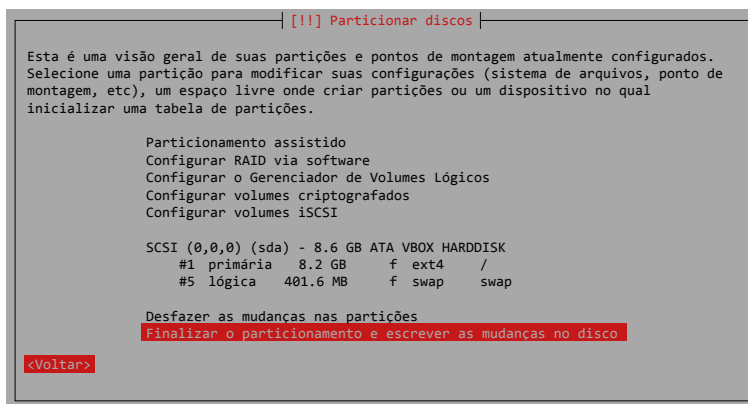
Figura 26- Escolhendo a opção *home*.



Fonte: elaborado pelo autor, 2016.

18. Na próxima tela, verifique se as opções de particionamento escolhidas estão corretas e, em caso positivo, selecione “Finalizar o particionamento e escrever as mudanças no disco”, e clique em Continuar.

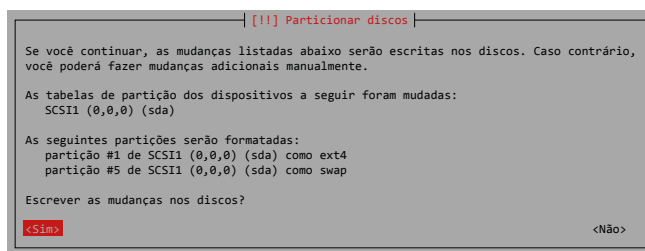
Figura 27- Verificando particionamento.



Fonte: elaborado pelo autor, 2016.

19. Na tela de confirmação do particionador, na opção “Escrever as mudanças nos discos?”, clique em “Sim” e, após, em Continuar.

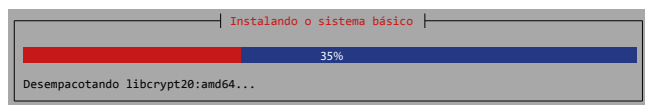
Figura 28- Confirmação de particionamento.



Fonte: elaborado pelo autor, 2016.

20. Em seguida, aguarde enquanto os discos são particionados e formatados, e o sistema operacional é instalado.

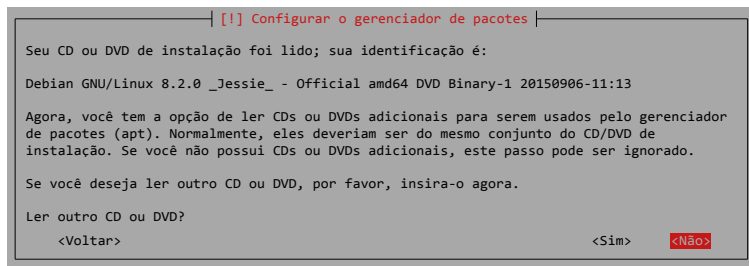
Figura 29- Particionando as configurações escolhidas.



Fonte: elaborado pelo autor, 2016.

21. Na tela seguinte, você pode optar por incluir outros DVDs com pacotes de *software* na instalação. Se tiver baixado DVDs adicionais, responda “Sim” à pergunta “Ler outro CD ou DVD?”. Caso contrário, clique em “Não”, e então em Continuar.

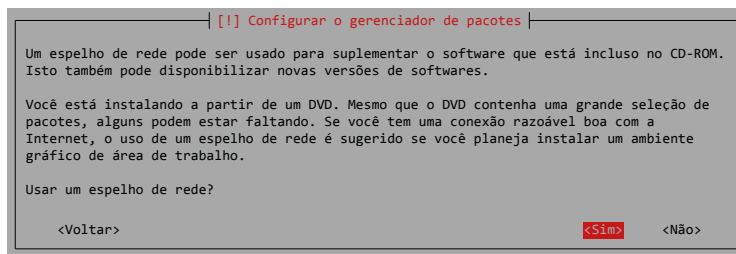
Figura 30- Tela opções de Dvd's para adicionar outros *softwares*.



Fonte: elaborado pelo autor, 2016.

22. Clique em “Sim” para escolher um espelho de rede (repositório de *software*), e clique em Continuar.

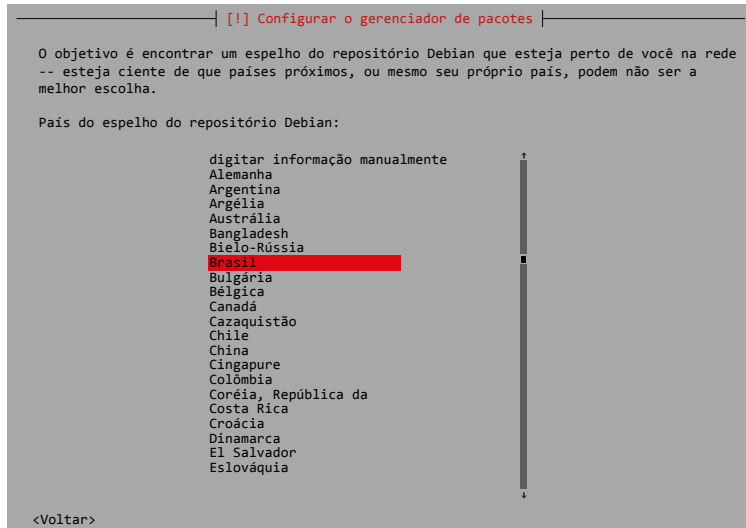
Figura 31- Tela de repositórios de *softwares*.



Fonte: elaborado pelo autor, 2016.

23. Selecione o país (Brasil) do espelho de rede e clique em Continuar.

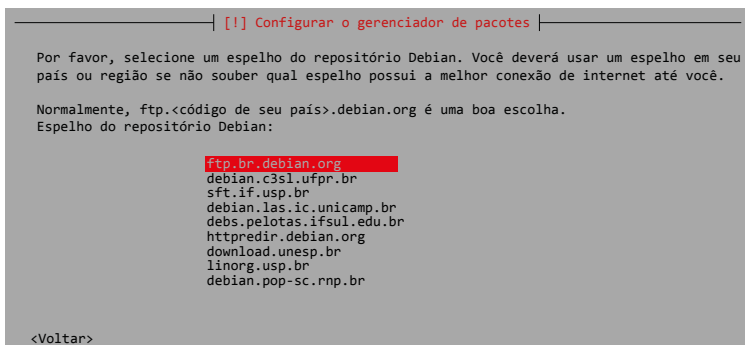
Figura 32- Tela opção de escolha por país.



Fonte: elaborado pelo autor, 2016.

24. Selecione o repositório desejado. É aconselhável usar o primeiro da lista, que é o **ftp.br.debian.org**. Clique em Continuar.

Figura 33- Tela para selecionar o repositório.



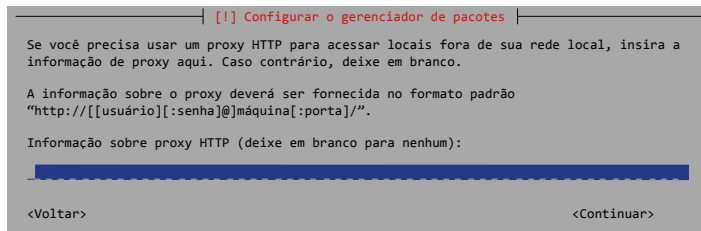
Fonte: elaborado pelo autor, 2016.

25. Na próxima tela, você pode informar o endereço do servidor *proxy* de sua rede, se houver. Caso não haja, deixe a caixa de texto em branco e simplesmente clique em Continuar.

Importante

Um servidor *proxy* é um computador que atua como intermediário entre uma rede local e a Internet. Neste computador em que foi instalado o servidor *proxy*, todos os outros podem acessar a Internet.

Figura 34- Tela opção servidor *proxy*.



[!] Configurar o gerenciador de pacotes

Se você precisa usar um proxy HTTP para acessar locais fora de sua rede local, insira a informação de proxy aqui. Caso contrário, deixe em branco.

A informação sobre o proxy deverá ser fornecida no formato padrão
"http://[[usuário][:senha]@]máquina[:porta]/".

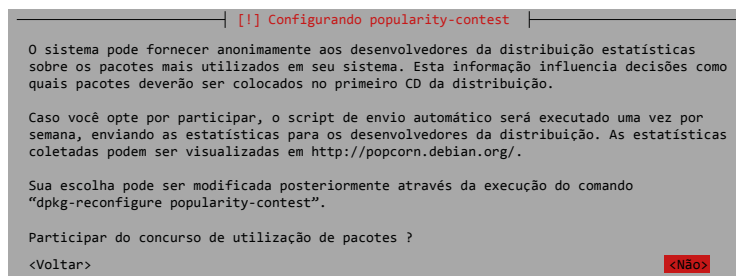
Informação sobre proxy HTTP (deixe em branco para nenhum):

<Voltar> <Continuar>

Fonte: elaborado pelo autor, 2016.

26. Na próxima tela, selecione se deseja participar do Concurso de Utilização de Pacotes. Sugerimos marcar a opção “Não” por se tratar apenas de uma máquina de testes. Clique em Continuar.

Figura 35- Tela opção participação do Concurso de Utilização de Pacotes.



[!] Configurando popularity-contest

O sistema pode fornecer anonimamente aos desenvolvedores da distribuição estatísticas sobre os pacotes mais utilizados em seu sistema. Esta informação influencia decisões como quais pacotes deverão ser colocados no primeiro CD da distribuição.

Caso você opte por participar, o script de envio automático será executado uma vez por semana, enviando as estatísticas para os desenvolvedores da distribuição. As estatísticas coletadas podem ser visualizadas em <http://popcorn.debian.org/>.

Sua escolha pode ser modificada posteriormente através da execução do comando "dpkg-reconfigure popularity-contest".

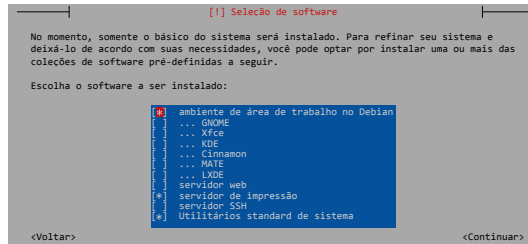
Participar do concurso de utilização de pacotes ?

<Voltar> <Não>

Fonte: elaborado pelo autor, 2016.

27. Na tela seguinte, selecione o *software* que será instalado em seu sistema operacional. Podemos escolher diversas interfaces gráficas, vamos optar pelo GNOME, deixe marcada também as opções servidor de impressão e utilitários *standard* de sistema. Clique então em Continuar.

Figura 36- Tela opção do GNOME.

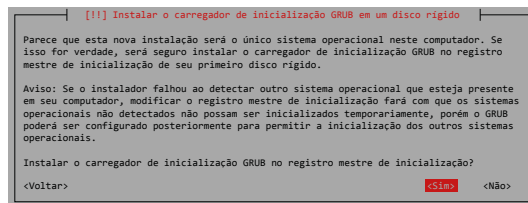


Fonte: elaborado pelo autor, 2016.

28. Após a instalação dos pacotes selecionados, configure a instalação do gerenciador de inicialização GRUB. Marque a opção “Sim” na tela seguinte e clique em Continuar.

Quando instalamos duas opções de inicialização de um sistema operacional, alteramos o registro mestre de inicialização chamado de MBR (*Master Boot Record*). Ele fica localizado no primeiro setor do disco rígido e é responsável por encontrar partições inicializáveis na tabela de partições e carregar o setor de *boot*. O Linux permite a instalação do GRUB e pode ser utilizado para escolher um dos sistemas operacionais instalados em seu computador.

Figura 37- Tela de configuração do GRUB.

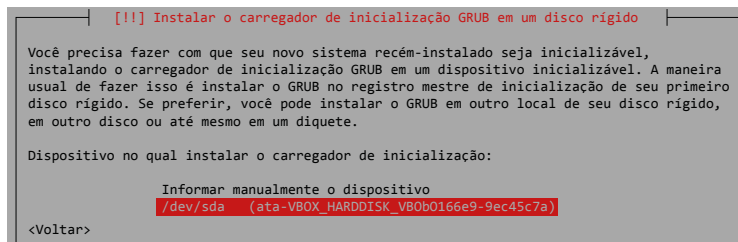


Fonte: elaborado pelo autor, 2016.

GRUB é a sigla para *Grand Unified Bootloader*. Trata-se de um gerenciador de *boot* (opção de inicialização do computador por um ou mais sistemas operacionais), desenvolvido inicialmente por Erich Stefan Boleyn.

29. E, na próxima tela, selecione o dispositivo onde o GRUB será instalado (normalmente, no dispositivo `/dev/sda`). Clique em Continuar.

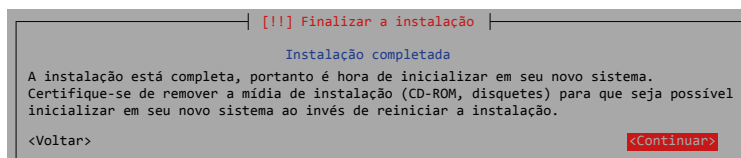
Figura 38- Tela configuração de inicialização do GRUB.



Fonte: elaborado pelo autor, 2016.

30. Instalação finalizada com sucesso!. Clique em Continuar para reiniciar o computador e começar a usar o novo sistema. Lembre-se de remover a mídia de instalação antes!

Figura 39 - Mensagem de finalização da instalação.



Fonte: elaborado pelo autor, 2016.

Resumindo

Neste capítulo, você aprendeu como instalar dois sistemas operacionais para *desktops*: o Windows 8.1 e o Linux. Importante observar cada passo a passo, pois apesar dos dois serem sistemas operacionais para usuários, cada um tem suas particularidades e peculiaridades, como onde obter as mídias de instalação, criar os usuários administradores, criar as senhas dos usuários, formatar a partição, configurar para entrar na rede e outros.

Lembre-se também que cada sistema operacional tem o seu procedimento de instalação, alguns mais fáceis como para os dispositivos moveis, e outros mais detalhados para os servidores. Experimente instalar outros tipos de sistemas operacionais para verificar as dificuldades e facilidades de cada um.

3

Processos

NO MERCADO DE computação existem diversos sistemas, de vários tipos, uso e fabricantes. O que existe em comum entre eles é que a base do funcionamento é o processo.

Conforme Tanenbaum (2009), o processo é uma abstração de um programa em execução. Para conhecermos os sistemas operacionais, temos que estudar os processos e o seu funcionamento.

Este capítulo permitirá que o aluno entenda os conceitos sobre sistemas operacionais e os processos, o thread, a comunicação entre processos e o escalonamento de processo.

Objetivos de aprendizagem:

- × Compreender como funcionam os processos de um sistema operacional.

3.1 Processo

Os processos são os programas em execução já adaptados, gerenciados e controlados pelos sistemas operacionais. O sistema operacional irá controlar o processo através do sistema de gerenciamento de processos, levando-se em conta a arquitetura do computador, características e projeto do processador. Por exemplo, um aplicativo pode ser executado em um determinado sistema operacional. O programador não precisa se preocupar se o computador tem um ou quatro processadores, quem cuida de executar como monoprocessado ou multiprocessado é o sistema operacional.

Importante

Tanenbaum (2009) considera que o processo é:

- Uma abstração de um programa em execução.
- O conceito mais central em qualquer sistema operacional é o processo.
- Processos são programas em execução, constituídos por: código executável, pilha de execução, estado do processo, prioridade do processo, valor do contador de programa (registrador PC), valor do apontador de pilha (registrador SP), valores de demais registradores.

Silberschatz, Galvin e Gagne (2008) consideram que processo é um programa em execução. Um processo necessita certos recursos como tempo de CPU, memória, arquivos e dispositivos de entrada e saída para executar sua tarefa. Estes recursos são associados ao processo no momento de criação ou em execução.

Monoprocessado: Computadores que possuem apenas um processador controlado por um sistema operacional

que realiza todas as operações do computador.

Multiprocessado: Computadores que possuem mais de um processador e estes podem compartilhar o mesmo sistema operacional, ou cada um possuir o seu próprio sistema operacional.

3.1.1 Criação de processo

Os sistemas operacionais precisam de mecanismos para criar e eliminar processos durante a operação. Há quatro eventos principais que fazem com que os processos sejam criados (TANENBAUM, 2009):

1. Início do sistema.
2. Execução de uma chamada de sistema (*system call*) de criação de processo por um processo em execução.
3. Uma requisição de usuário para criar um novo processo.
4. Início de uma tarefa em lote (*batch job*).

O sistema operacional Linux cria os processos da seguinte maneira:

- I. Ao iniciar o sistema, é criado um processo chamado *init* recebe a identificação de processo número 1.
- II. Este processo coloca os outros processos em execução usando uma chamada de sistema denominada *fork*.
- III. Os processos filhos são criados pelos processos pais.

O Sistema operacional Windows cria seus processos da seguinte forma:

- I. Cada processo do Windows possui um identificador próprio chamado de *handle* (manusear). Podemos comparar esses processos criados de forma hierárquica, pois um processo gera outro e a partir desse ponto o processo filho ganha uma ligação com o identificador do processo pai.
- II. O conceito de hierarquia é desmontado quando um processo pai passa seu *handle* (manusear) para outro processo, assim o

processo filho quebra sua antiga ligação com o processo pai e gera uma nova ligação com seu novo processo pai.

A diferença que existe entre o Linux e o Windows, quando se fala de processos, é que no Linux quando um processo pai é “morto”, seus filhos não morrem junto com o processo pai, isso é ruim quando se trata de vírus, pois o processo não é morto apenas se destruir o processo pai.

System call: é o mecanismo usado pelo programa para requisitar um serviço do sistema operacional, ou mais especificamente, do núcleo do sistema operacional.

Batch job: é um arquivo em lote utilizado para otimizar tarefas. Um arquivo em lote é basicamente um arquivo de *script* que é executado sequencialmente.

Init: Abreviação de “initialization”, onde o processo inicial, carrega todos os outros processos em sistemas Unix e Linux.

3.1.2 Término dos processos

Depois de criado, um processo começa a executar e fazer o seu trabalho mais cedo ou mais tarde esse processo chegara ao fim e um novo processo terminará, normalmente em razão de alguma das seguintes condições (TANENBAUM, 2009):

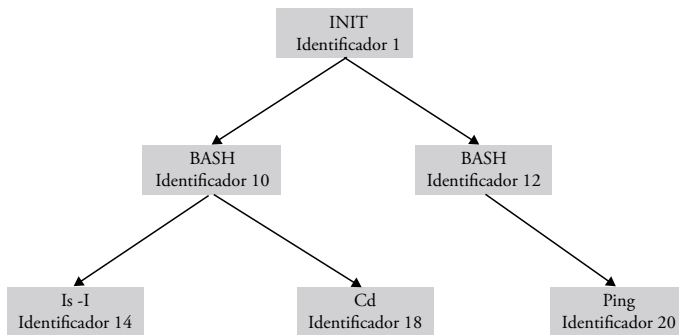
1. Saída normal (voluntária), o processo terminou de forma normal.
2. Saída por erro (voluntária), o processo não terminou de forma normal, saiu devido a um erro fatal.
3. Erro fatal (involuntário), erro causado pelo próprio processo.
4. Cancelamento por um outro processo (involuntário), através de um sinal.

3.1.3 Hierarquia de processos

Os processos, quando são criados, seguem uma hierarquia. Os processos filhos são criados por um processo pai e os processos filhos podem criar outros

processos. Segundo (SILBERSCHATZ, GALVIN, GAGNE, 2008), no sistema operacional Unix ou Linux, o processo principal chama-se init e tem o identificador do processo com o número um. Ele que cria os demais processos e cada processo receberá um identificador do processo seguindo a sequência numérica (hierarquia). No Windows não existe hierarquia rígida de processos ou grupos de processos. Todos os processos são tratados de forma igualitária pelo sistema operacional.

Figura 1 - Exemplo de hierarquia de processos no Linux.



Fonte: Elaborado pelo autor, 2016.

Saiba mais

Para saber mais na prática sobre Hierarquia dos processos, aprender algumas diferenças interessantes sobre hierarquia de processos no Windows e Unix e como isso afeta até mesmo em uma propagação de vírus, acesse o site:

<http://www.devmedia.com.br/hierarquia-de-processos-no-unix-e-windows/24739>

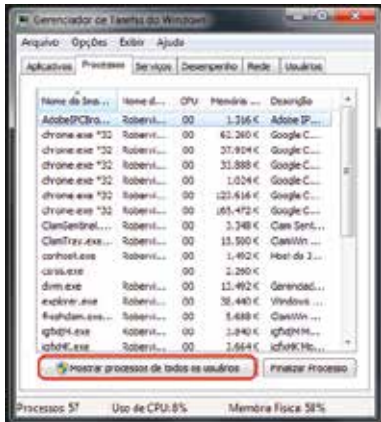
Na figura 1 podemos visualizar a criação dos processos a partir do primeiro processo, no caso a Init (identificador 1), que a partir de suas ações cria outros processos (identificador 10, 12, 14, 18, 20) e cada um com sua identificação. Essa identificação ajuda o processador e o sistema operacional identificar cada processo.

Exemplo de criação de Processos no Windows

Na figura 2 podemos visualizar os diversos processos no sistema Operacional Windows que estão acessando os recursos do Processador. Os processos não são criados como uma hierarquia. O acesso aos recursos são administrados pelo Sistema Operacional conforme a ordem de prioridades.

3.1.4 Estados de um processo

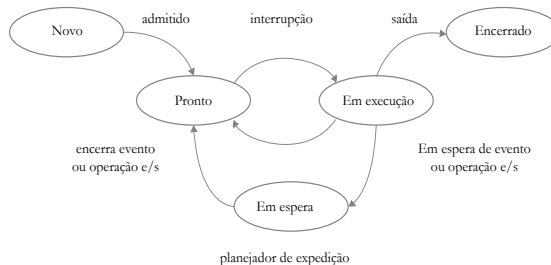
Um processo pode assumir os seguintes estados durante a sua existência (SILBERSCHATZ, GALVIN, GAGNE, 2008):



Fonte: Elaborado pelo autor, 2016.

- × Novo (*new*): o processo está em criação;
- × Em execução (*running*): O processo está sendo executado, ou seja, está utilizando o processador;
- × Em espera (*waiting*): o processo está aguardando a resposta de algum evento ou operação de entrada/saída;
- × Pronto (*ready*): o processo está aguardando para ser executado pela CPU;
- × Encerrado (*terminated*): o processo encerrou suas execuções.

Figura 3 - Diagrama de estados do processo.



Fonte: Elaborado pelo autor, com base em SILBERSCHATZ, GALVIN, GAGNE, 2008.

3.2 Threads

Os primeiros sistemas operacionais foram projetados para serem mono-processados, ou seja, eram executados em um único processador, dessa forma cada processo tinha um espaço de endereçamento e somente um *thread* de controle (TANENBAUM, 2009).

Cada processo fornece os recursos necessários para executar um programa. Um processo tem um espaço virtual (um endereço), código executável, identificadores abertos para objetos do sistema, um contexto de segurança, um identificador único processo, variáveis de ambiente, uma classe de prioridade, tamanhos conjunto de trabalho mínimo e máximo, e pelo menos um segmento de execução. Cada processo é iniciado com um único segmento, muitas vezes chamado o segmento principal, mas pode criar tópicos adicionais a partir de qualquer um de seus tópicos (MICROSOFT, 2015).

Segundo (TANENBAUM, 2009), um processo pode conter mais de uma instrução ou tarefa, que pode assumir várias ações, a qual damos o nome de *thread*, que é uma divisão do processo principal de um programa. Todavia, nem todos os processos são divididos em múltiplos *threads*, ou múltiplas ações, assim como nem todos os processadores são capazes de trabalhar “tranquilamente” com uma enormidade de *threads*.

Importante

A grande maioria das linguagens de programação como JAVA, #NET entre outras utilizam recursos de *threads*. Para entendermos como esse recurso é importante, quando usamos um *browse* (navegador internet), ele permite fazer o *download* de vários arquivos ao mesmo tempo, gerenciando as diferentes velocidades de cada servidor e, ainda assim, permitindo que o usuário continue interagindo, mudando de página enquanto os arquivos estão sendo carregados.

Segundo Tanenbaum (2009), as principais razões para existirem *threads* são:

1. O modelo de programação se torna mais simples se decomposmos uma aplicação em múltiplos *threads* sequenciais que executam em

quase paralelo. Considerando um espaço de tempo muito curto, esse paralelismo não é real, pelo que a execução de sistemas operativos multiprogramações num computador com um único processador recebe a designação de pseudoparalelismo.

2. Mais fáceis (isto é, mais rápidos) de criar e destruir que os processos, pois não tem quaisquer recursos associados a eles. Em muitos sistemas, criar um *thread* é cem vezes mais rápido do que criar um processo.
3. Os usos de *threads* não resultam em ganho de desempenho quando todos eles são *CPU-bound* (limitados pela CPU, isto é, muito processamento com pouca E/S). No entanto, quando há grande quantidade de computação e de E/S, os threads permitem que essas atividades se sobreponham e, desse modo, aceleram a aplicação.

Microsoft Windows oferece suporte a multitarefa preemptiva, que cria o efeito de execução simultânea de vários segmentos, de vários processos. Em um computador com múltiplos processadores, o sistema pode executar simultaneamente como muitos segmentos, como existem processadores no computador (MICROSOFT, 2015).

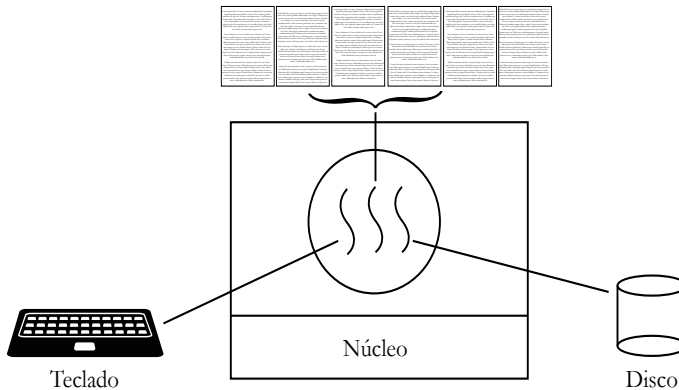
Saiba mais

Multitarefa Preemptiva: operação realizada pelo Sistema Operacional onde realiza a interrupção e a execução de um programa e passa o controle do sistema a outro programa que está em espera. A multitarefa preemptiva é um recurso do Sistema Operacional que impede que um programa monopolize o sistema. Outro termo utilizado para essa ação chama-se *time-slicemultitasking* (multitarefa por fatia de tempo, multitarefa por fração de tempo).

O Linux implementa a estrutura de processos desde as primeiras versões. Os threads podem ser criados pelos processos ou mesmo pelos programas que são executados.

Um exemplo dos threads funcionando é você utilizar um editor de texto, enquanto você digita. Um thread cuida disso, mas temos um outro thread cuidando de colocar o texto na tela e outro salvando em disco. Veja a figura 4:

Figura 4 - Editor de textos com três *threads*.



Fonte: Tanenbaum, 2009.

3.3 Comunicação entre processos

Segundo Tanenbaum (2009), os sistemas operacionais consideram os processos como independentes com as suas áreas de memórias e threads de controles. Mas, em certos momentos, os processos precisam se comunicarem para trocar dados e instruções.

Processos em um sistema podem ser Independentes ou Cooperantes. Processos Independentes não podem afetar ou ser afetados pela execução de outro processo. Processos Cooperantes podem afetar ou ser afetados pela execução de outro processo (SILBERSCHATZ, GALVIN, GAGNE, 2008). Os Processos interdependentes não sofrem nenhuma ação de outro processo, por exemplo: o processador precisa liberar o teclado para que o usuário possa digitar um texto, esse processo não pode ser afetado, por outro processo, senão o teclado ficaria indisponível. Já os processos cooperantes podem ser afetado sem um jogo que está utilizando uma porcentagem da memória, que poderá sofrer ações de outro processo que a CPU demandar.

Razões para cooperação entre processos (SILBERSCHATZ, GALVIN, GAGNE, 2008):

- × Compartilhamento de Informações

- × Aumento na velocidade da computação
- × Modularidade
- × Conveniência

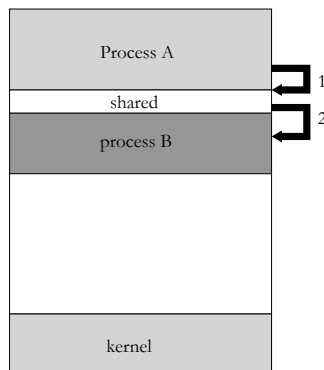
Processos cooperantes precisam de Comunicação entre Processos (IPC – interprocess communication) (SILBERSCHATZ, GALVIN, GAGNE, 2008).

Para Silberschatz e Galvin (2008), a comunicação entre processos é o mecanismo para que os processos se comuniquem e sincronizem suas ações. Os processos se comunicam entre si sem recorrer às variáveis compartilhadas.

Fundamentalmente, existem duas abordagens:

- × Suportar alguma forma de espaço de endereçamento compartilhado. Sharedmemory (memória compartilhada), os processos podem trocar informações através de leitura e escrita de uma área de memória compartilhada.

Figura 5 - Processos sendo compartilhado na Memória



Fonte: SILBERSCHATZ & GALVIN, 2008.

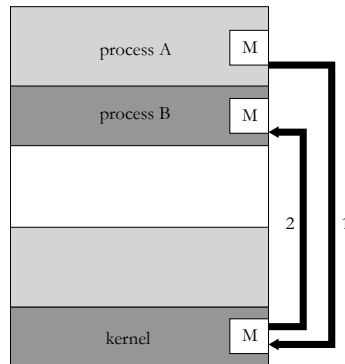
Process: processo que está sendo compartilhado na memória.

Shared: compartilhado, o processo que está sendo compartilhado na memória.

Kernel: núcleo do Sistema Operacional.

- × Utilizar comunicação via núcleo do S.O., que ficaria então responsável por transportar os dados de um processo a outro, a troca de mensagens. A comunicação acontece através da troca de mensagens dos processos cooperativos.

Figura 6 - Troca de Mensagens entre os processos na memória.



Fonte: SILBERSCHATZ & GALVIN, 2008.

O recurso utilizado para a troca de mensagens é chamado de *local procedure call* (LPC).

- a) Só funciona entre processos no mesmo sistema;
- b) Usa portas (como mailboxes) para estabelecer e manter canais de comunicação;
- c) Comunicação funciona da seguinte forma:
 - × O cliente abre um manipulador para o objeto porta de conexão do subsistema.
 - × O cliente envia uma solicitação de conexão.
 - × O servidor cria duas portas de comunicação privadas e retorna o manipulador de uma delas para o cliente.
 - × O cliente e o servidor usam o manipulador da porta correspondente para enviar mensagens ou retornos de chamadas e ouvir respostas.

3.4 Escalonamento de processos

O escalonamento de processos acontece quando mais de um processo ou thread (vários processos) querem utilizar o mesmo processador, um algoritmo de escalonamento será utilizado para verificar qual será executado primeiro e qual será a prioridade.

Conforme Silberschatz, Galvin e Gagne (2008), o objetivo da multiprogramação é ter processos em execução o tempo todo, para maximizar a utilização da CPU. O objetivo do tempo compartilhado é alternar a CPU entre processos de forma tão frequente que os usuários possam interagir com cada programa durante sua execução. Para atender a esses objetivos, o Escalonador de processo seleciona um processo disponível (possivelmente a partir de um conjunto de vários processos disponíveis) para a execução do programa na CPU.

Muitos dos problemas que se aplicam ao escalonamento de processos também são válidos para o escalonamento de threads, embora haja diferenças. Quando o núcleo gerencia *threads*, o escalonamento normalmente é feito por thread, dando pouca ou nenhuma atenção ao processo ao qual o thread pertence (TANENBAUM, 2009).

3.4.1 Filas de escalonamento de processo

Quando os processos entram no sistema, eles são colocados numa fila de tarefas (*jobqueue*). Os processos que são residentes na memória principal e estão prontos e esperando para executar são mantidos em uma lista chamada de fila de processos prontos (*readyqueue*). Essa fila é geralmente armazenada como uma lista ligada. Um cabeçalho pronto-fila contém indicações para o primeiro e o último PCB - *Process Control Block* (contém informações associadas a cada processo) na lista. Cada PCB inclui um campo apontador que aponta para o seguinte na fila de PCB pronto. O sistema também inclui outras filas. Quando um processo é atribuído a CPU, ele executa por um tempo e, eventualmente, sai, é interrompido, ou aguarda a ocorrência de um evento específico, como a conclusão de um pedido de E / S (SILBERSCHATZ, GALVIN e GAGNE, 2008). As filas de escalonamento são:

- × Fila de tarefas (*JobQueue*): conjunto de todos os processos no sistema.
- × Fila de Processos prontos (*Readyqueue*): conjunto de todos os processos residentes na memória principal, prontos e esperando para executar.

- × Fila de dispositivos: conjunto dos processos esperando por um dispositivo de E/S.

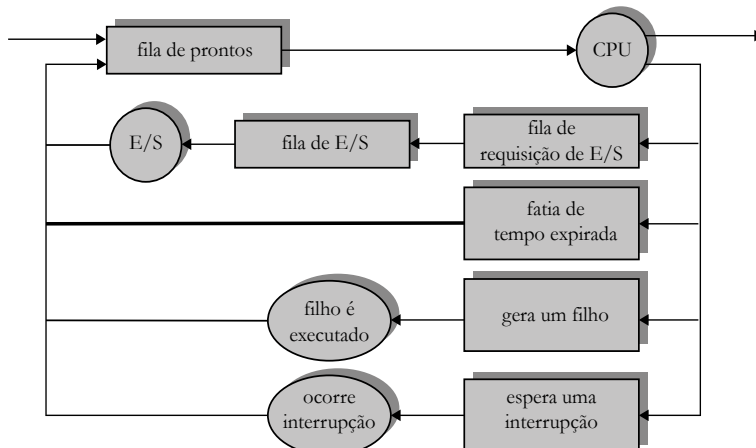
Lista ligada: termo utilizado para alocação, que consiste em cada arquivo composto por uma lista ligada de blocos do disco. Não precisa ser contínuo, armazena apenas blocos iniciais e sem acesso randômico

Processo: são os programas em execução já adaptados, gerenciados e controlados pelos sistemas operacionais.

Thread: Para um processo poder ser executado em mais de um processador simultaneamente, teríamos que ter mini processos dentro dos processos, estes minis processos podem ser chamados de *threads* (TANENBAUM, 2009)

Escalaonamento: O escalaonamento de processos acontece quando mais de um processo ou thread querem utilizar o mesmo processador

Figura 7 - Representação do Escalonamento de Processos



Fonte: SILBERSCHATZ, GALVIN E GAGNE, 2008.

3.4.2 Escalonadores

Segundo Tanenbaum (2009), escalonador de Longo Prazo (ou escalonador de Jobs) – seleciona quais processos devem ser trazidos para a fila de processos prontos. É invocado muito frequentemente (milissegundos) e deve ser rápido.


Escalonador de Curto Prazo (ou escalonador da CPU) – seleciona qual processo deve ser executado a seguir e aloca CPU para ele. É invocada com pouca frequência (segundos, minutos) e pode ser lento.

Estudo de caso:

A empresa OK&Sons roda um sistema de escrita fiscal em um servidor monoprocessado. O sistema não recebe mais atualizações de *software*, o servidor está apresentando problemas de desempenho e pela antiguidade não existe peças de reposição. O seu chefe pede uma solução para este problema e o fornecedor de *hardware* apresenta duas propostas: a primeira com dois processadores e o dobro de memória do servidor antigo e a segunda com oito processadores e o quádruplo de memória do servidor antigo. Qual seria a sua escolha? Justifique e aponte os problemas para realizar esta migração.

Você sabia

Segundo o Canaltech, a Intel lançou o seu primeiro modelo com a tecnologia *hyper-threading*, em 2002, no modelo Xeon MP Foster, processador voltado para a linha de servidores. Em seguida, incorporando ao Pentium 4 (arquitetura Northwood) e em todos os modelos que vieram na sequência. Embora o sistema operacional “enxergue” o dobro de núcleos de processamento presentes, na prática não é isso que acontece. Na verdade, é que cada núcleo físico possui duas unidades lógicas independentes, cada uma com um controlador de interrupção programável (APIC) e conjunto de registradores próprio. Veja mais em: <http://canaltech.com.br/dica/produtos/Como-funciona-o-Hyper-Threading/>. (CANALTECH, 2016).



Resumindo

Neste capítulo tivemos a oportunidade de entender como funcionam os processos nos sistemas operacionais, a sua evolução para os *threads* em computadores com mais de um processador, a comunicação entre os processos e o escalonamento de processos. A importância deste conhecimento é que possibilitará aos administradores de sistemas operacionais o entendimento se os aplicativos e programas estão preparados para tirar o máximo de proveito do servidor, verificando se os processos estão usando todos os processadores ou se os processos estão afunilando em um único processador.

4

Memória

NESTE CAPÍTULO IREMOS discutir o conceito de memória no âmbito de um sistema operacional. Vamos conhecer suas nuances, tipos, abordagens e formas de organização.

Assim como no corpo humano, os dispositivos computacionais, sejam eles um computador pessoal, um grande servidor corporativo, ou até pequenos dispositivos como celulares e tablets, também possuem sua memória. Da mesma forma, assim como algumas pessoas possuem mais ou menos capacidade de memorização, estes dispositivos também tem essa característica, fazendo com que cada vez mais o conceito de memória seja discutido e estudado em ambientes computacionais, visando atender às crescentes demandas da sociedade atual no quesito tecnologia. Por anos, o meio tecnológico vem se desenvolvendo buscando criar abordagens que sejam capazes de armazenar mais informações e em menos tempo.

Para tanto, neste capítulo serão apresentados os diferentes tipos de memória, como eles se organizam de forma hierárquica abrangendo os processos de tradução de instruções em memória, bem como os modos de gerenciamento de memória implementados pelos sistemas operacionais.

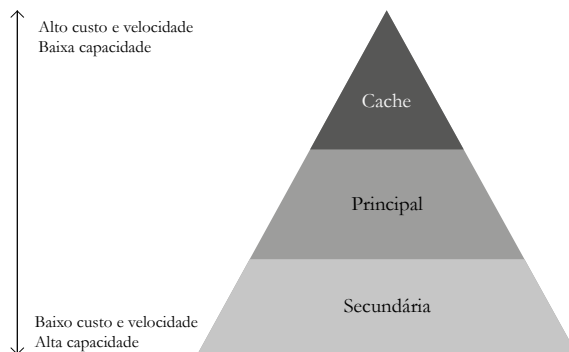
Objetivo de Aprendizagem:

- × Conhecer os tipos de memória e as suas relações hierárquicas entre si;
- × Identificar os objetivos da memória em um sistema operacional visando a eficiência da tecnologia;
- × Interpretar o ato de tradução de programas em processos e sua relação com as memórias em um sistema operacional;
- × Compreender as diferentes abordagens de gerenciamento de memória;
- × Conhecer os algoritmos utilizados pelo sistema operacional para alocar dados na memória.

4.1 Tipos de Memória

Em um ambiente computacional, as memórias são subdivididas em três tipos principais, organizados de forma hierarquizada. Cada uma destas divisões é responsável por determinadas operações e são alojadas em diferentes partes do computador. Em cada tipo de memória existem grandes diferenças entre custos, capacidades, velocidades e funções, ficando a cargo do sistema operacional o gerenciamento das operações entre as diferentes memórias disponíveis.

Figura 1 - Hierarquia de memórias.



Fonte: Elaborado pelo autor, com base em DEITEL; DEITEL; CHOFFNES, 2005.

Dentro da hierarquia de memória (figura 1), quanto maior a velocidade, menor é a sua capacidade de armazenamento e maior é o custo do mesmo. Por outro lado, quanto maior a capacidade, menor é o custo e a sua velocidade.

A primeira e mais rápida das memórias, a **memória cache**, devido ao seu alto custo, possui pouca capacidade de armazenamento. Esta é uma memória do tipo volátil e é extremamente rápida, fazendo com que a quantidade de tempo de acesso aos dados nela contidos sejam muito inferiores à duração de operações na memória principal ou secundária. Sua função é auxiliar a memória principal, alocando em si parte dos dados necessários para a operação da memória principal (TANENBAUM, 2009). Como não há viabilidade em executar um programa por completo apenas na memória cache dado o seu alto custo e baixa capacidade, os estágios seguintes na hierarquia garantem este funcionamento.

Na **memória principal** são armazenados os dados em execução no momento. É uma memória volátil, com mais capacidade de armazenamento que a memória cache, porém com menos velocidade. Os dados dos programas que são provenientes da memória secundária são armazenados temporariamente na memória principal para que estes sejam executados. A memória principal é dividida em memória lógica e memória física. Os programas tem acesso apenas à memória lógica, de forma que os dados que os mesmos precisam alocar na memória são armazenados única e exclusivamente na memória lógica, que serve como camada entre o programa e a memória física. Na memória física, os dados em que os programas tem acesso via memória lógica são devidamente alocados nos circuitos físicos da placa de memória.

Já na **memória secundária**, os dados são armazenados permanentemente de forma não-volátil. Este tipo de memória é de baixo custo, sendo assim sua capacidade é muito superior às outras, porém é de baixa velocidade, e é nela que os programas são armazenados (TANENBAUM, 2009). A memória secundária em um ambiente computacional se materializa por meio dos discos e demais unidades de armazenamento permanente não-voláteis, como *hard-disks*, *pendrives*, cartões de memória, entre outros.

Memória Cache: memória de altíssima velocidade residente na unidade de processamento.

Memória Primária: memória de alta velocidade residente na memória RAM.

Memória Secundária: memória de baixa velocidade, residente nos discos.

Há uma explicação clara sobre esta divisão dos dados em diferentes tipos de memória: a memória principal como o próprio nome sugere é a protagonista de toda arquitetura pois nela que os dados em execução são armazenados. Por questões óbvias, o ideal é que ela seja extremamente rápida e de alta capacidade. Deve ser rápida para auxiliar no dia a dia fazendo com que o computador nunca trave ou atrase processos, e ela também deve possuir alta capacidade para que caibam muito mais dados durante a execução. Porém nada disso é viável economicamente, pois uma memória principal com estas características seria muito cara.

Importante

Memórias do tipo **volátil** dependem de alimentação para manter o armazenamento, enquanto as memórias do tipo **não-volátil** possuem a capacidade de manter o armazenamento mesmo quando não há alimentação.

Para solucionar o problema da capacidade existe a memória secundária, onde sua construção física permite que com menor custo sejam armazenados mais dados. E visando solucionar o problema da velocidade, surge a memória cache como uma alternativa muito mais rápida para atender a demanda do processador em extrair dados da memória principal.

Com a memória cache, os dados trafegam em velocidades compatíveis com o processador, para que sempre que seja necessário um dado, o processador verifique antes na memória cache se ele já está ali alocado antes de buscar na memória principal, que é mais lenta. Por ser muito mais performática, ela é mais cara, por isso sua capacidade é limitada. Desta forma, em algumas arquiteturas, a memória cache se divide em duas para garantir mais performance, ficando uma parte no processador e outra na placa-mãe.

Quadro 1 - Esquema geral da hierarquia de memórias.

Memória	Volátil	Local	Velocidade	Capacidade	Custo
Cache	Sim	Processador ou Placa-mãe	Muito Alta	Muito Baixa	Muito Alto
Principal	Sim	Pentes de memória	Alta	Baixa	Alto
Secundária	Não	Discos de armazenamento, <i>pendrives</i> , <i>flash drives</i> , etc.	Baixa	Muito Alta	Baixo

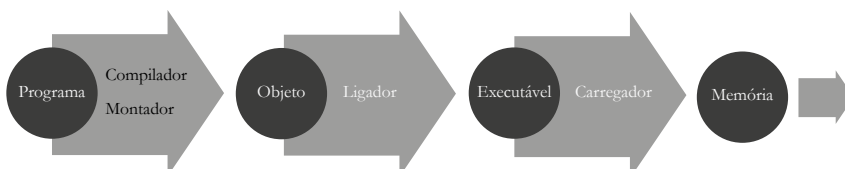
Fonte: Elaborado pelo autor, com base em TANENBAUM 2009.

É papel do sistema operacional orquestrar esta troca de dados, carregando os programas oriundos da memória secundária na memória principal, para assim garantir a sua execução. Todo o processo de gestão da transferência de dados entre as hierarquias de memória é realizado pelo gerenciador de memória, uma funcionalidade do sistema operacional que controla os espaços livres e as alocações de dados entre as memórias disponíveis no ambiente (SILBERSCHATZ, 2008).

4.2 Tradução de Programas

Para que os programas armazenados sejam executados, é necessário que os mesmos sejam traduzidos e transformados em processos. Este processo se inicia quando a Unidade de Gerência de Memória (MMU – *Memory Management Unit*), uma funcionalidade de hardware que organiza o roteamento dos espaços de memória, mapeia os endereços da memória lógica com seus respectivos endereços de memória física (TANENBAUM, 2009). Mas para que o programa seja um processo, ele passa por alguns passos ilustrados na figura 2.

Figura 2 - Hierarquia de memórias.



Fonte: Elaborado pelo autor, com base em TANENBAUM, 2009.

O processo de tradução de programas para a execução na memória se inicia com o **compilador**. Ele serve para transformar o código-fonte do programa, um arquivo de texto produzido na linguagem de programação em linguagem de máquina. O compilador transforma a codificação feita na linguagem de programação também em um arquivo texto, mas em uma linguagem em que o carregador consegue interpretar. A linguagem de máquina possui muito menos instruções e para que haja a execução do programa, estas poucas instruções devem ser combinadas para criar rotinas, diferentemente das linguagens de programação, que possuem mais instruções fazendo com que o processo de codificação do programa seja mais produtiva.

Após a compilação, em geral, o **montador** converte o programa já “compilado” em um arquivo de objeto, mas já em linguagem binária. Este processo serve para transformar o programa que foi unicamente transformado em um arquivo em linguagem de máquina em um arquivo com as corretas instruções para a execução e a alocação dos dados na memória. Sendo assim, cada instrução criada em linguagem de máquina é traduzida de forma binária.

Com as instruções já “montadas” no arquivo de objetos, o **ligador** une as rotinas em um arquivo executável. Ele cria um arquivo já pronto para execução de acordo com as bibliotecas do sistema operacional. Sua função é criar o arquivo dividindo as instruções compiladas e montadas em procedimentos separados, visando reaproveitar esta execução quando há alterações no programa. Desta forma não é necessária uma nova compilação, montagem e ligação a cada alteração de dados no programa.

Em termos de execução, o último estágio é quando o **carregador**, uma funcionalidade do sistema operacional, porta o arquivo executável e inicia sua execução, alocando os seus dados na memória principal. O carregador lê o arquivo executável e determina os espaços, dimensões e endereços de memória para a execução do programa. Finalizando então o processo de tradução de programas em processos na memória principal (SILBERSCHATZ, 2008).

4.3 Gerenciamento de Memória

O gerenciador de memória, conforme vimos na seção anterior, deve ser capaz de organizar o tráfego de dados. Isto é, ele deve fazer com que os dados provenientes da memória secundária sejam alocados em espaços de memória

vagos da memória primária. Sendo que após isso eles devem ser desalojados quando seus processos terminam, garantindo que haverá espaços de memória principal vagos para novos processos.

Esta operação se faz necessária, pois dada a sua capacidade, na memória principal não comporta todos os dados para a execução dos processos, sendo assim se faz necessária a alocação e retirada continua de dados. Em sistemas operacionais mono-tarefa, onde apenas um processo é executado por vez, esta operação é mais simples, pois basta compartilhar a memória entre o sistema operacional e o programa, tendo apenas um programa sendo executado por vez.

Porém, nos ambientes computacionais de hoje em dia, a alocação de memória deve ser capaz de operar em multitarefa, onde diversas operações podem ser realizadas ao mesmo tempo, aumentando assim consideravelmente o grau de complexidade dos processos de alocação de memória. Para resolver estes problemas, diversas abordagens e algoritmos distintos foram criados para atender às situações mono-tarefa e multitarefa. Nas próximas seções veremos as diversas abordagens e algoritmos de alocação de memória (TANENBAUM, 2009).

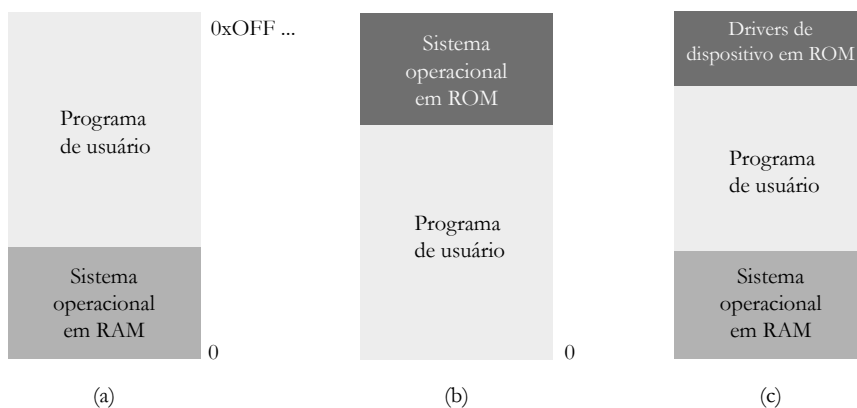
4.3.1 Gerenciamento sem Troca de Processos ou Páginas

Os sistemas operacionais mono-tarefa, ou mono-programados, são abordagens computacionais mais antigas, pois apenas um programa é executado por vez. Porém, em dispositivos computacionais limitados como placas programáveis, dispositivos móveis simples e em computadores para operações específicas é comum ainda a abordagem da mono-programação (DEITEL; DEITEL; CHOFFNES, 2005).

Estas operações mono-tarefa, para gerenciar a memória, costumam não utilizar troca de processos ou páginas, já que o compartilhamento de memória é simples, pois a memória faz apenas o gerenciamento entre o programa e o sistema operacional. Sendo assim, o programa pode utilizar toda a memória disponível, porém ele está limitado apenas a este limite. O sistema operacional carrega o programa na memória e aloca os dados de acordo com a demanda, porém os próximos dados irão sobrepor os espaços de memória já alocados em dados anteriores. Como há apenas uma tarefa sendo executada naquele momento, não há grandes problemas com isso.

A arquitetura mais antiga para a alocação mono-tarefa, sem trocas ou paginações (figura 3-a), utilizada em grandes computadores, é onde o sistema operacional estava na parte inferior da memória RAM e o programa na parte superior. Outra abordagem, utilizada em pequenos dispositivos (figura 3-b), utiliza a alocação do sistema operacional apenas para leitura na parte superior da memória ROM e o programa na parte inferior. Uma terceira abordagem, utilizada em computadores pessoais antigos (figura 3-c), os dados do sistema operacional relacionados com drivers dos dispositivos ficam na memória ROM e o restante fica na parte inferior (RAM) (TANENBAUM, 2009).

Figura 3 - Exemplo para uso da ferramenta 5W2H.



Fonte: Elaborado pelo autor, com base em Tanenbaum, 2009.

Memória ROM (*Read only memory*): São memórias apenas para leitura, estes tipos de memórias não permitem que novos dados sejam gravados de forma dinâmica. Em geral, apenas no momento de fabricação ou em procedimentos específicos, os dados podem ser gravados. Exemplos são CD-ROM, memórias EPROM de placas programáveis, BIOS de placas-mãe, entre outros.

Memória RAM (*Random Access Memory*): São memórias de acesso aleatório, ou seja, os processos de gravação e alocação de

memória são dinâmicos, podendo ocorrer a qualquer momento durante a execução dos processos. Elas são representadas pela memória principal do ambiente, que visa dar maior velocidade aos dados contidos nos discos no momento de sua execução.


4.3.2 Gerenciamento com Memória Virtual

O conceito de memória virtual é viabilizar a execução de programas que são maiores do que a memória RAM e sejam executados de forma satisfatória. Para tanto, esta abordagem de gerenciamento de memória utiliza partes da memória secundária, utilizando na memória principal apenas os dados ativos e necessários para a execução.



Você sabia

Do ponto de vista comercial, sempre que se cita a quantidade de memória de um computador, está se referenciando a memória principal, contida na Memória RAM. Sendo assim, sempre que a “quantidade de memória” de um computador é citada, de forma básica está sendo tratado apenas da capacidade dos pentes de memória RAM, ignorando as outras memórias da hierarquia.



Este tipo de abordagem elimina a rigidez do endereçamento físico da memória principal, fazendo com que toda a alocação seja feita apenas com base na memória lógica. Assim, se desvincula a origem da memória fazendo com que exista um endereçamento real dividido entre a memória principal e a secundária, e o armazenamento virtual, que é apenas utilizado como interface visando suprir as necessidades de execução. Assim, os programas fazem referência à memória virtual sem a necessidade de se preocupar com os endereços físicos.

Quando estamos em um ambiente de memória virtual, diversos casos distintos podem acontecer. Vamos conhecer alguns deles, verificando quando e porquê que eles ocorrem. O primeiro deles é a **segmentação**, onde os pro-

gramas são organizados em blocos de informações, subdividindo-se por funcionalidades chamadas de segmentos. Cada segmento possui em seus blocos de memória o endereçamento para uso dos programas, utilizando apenas a memória virtual para acessá-los.

Em alguns casos, onde não há memória suficiente sendo referenciada na memória real, o sistema operacional deve eleger processos que serão eliminados, visando abrir espaço para a alocação destes novos dados. O *swapping* faz este trabalho elegendo por prioridade e estado os processos que estão em espera, para que estes sejam eliminados garantindo que os próximos dados sejam alocados com efetividade.

Quando há muito trâmite de dados entre a memória real e a virtual, ocorre o *thrashing*. Um estado do sistema operacional oriundo do tempo empenhado em efetuar as transferências entre as instância de memória, diminuindo assim a efetividade das alocações. Este estado força o ambiente a deter mais memória física, necessitando assim de um upgrade no hardware para atender a esta necessidade.

Visando diminuir a quantidade de dados na memória principal, uma das soluções é implementar o **compartilhamento de memória**, uma abordagem que processos distintos que utilizam os mesmos dados compartilhem as mesmas referências de memória. Desta forma, mais processos conseguem operar utilizando cada vez menos dados, liberando mais espaço na memória, diminuindo o custo desta aplicação, por exemplo (TANENBAUM, 2009).

4.3.3 Gerenciamento com Troca de Processos

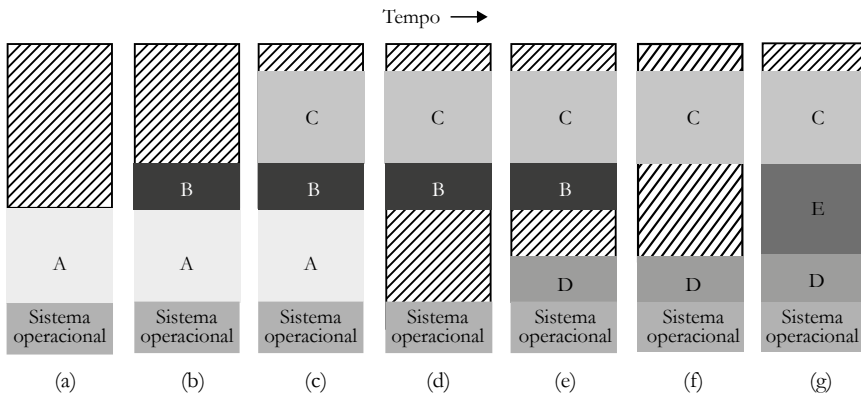
Com um gerenciamento de memória permitindo a troca de processos, sempre que um processo é terminado, ele deixa seu espaço de memória livre. Porém, a realocação de memória não utiliza uma fila, os novos processos são alocados dinamicamente nos espaços de memória vagos de acordo com o espaço disponível para o novo processo, que deve ir para a memória (TANENBAUM, 2009).

Desta forma, a alocação é mais otimizada, pois em uma alocação baseada em filas, os espaços de memória liberados podem ser grandes ou peque-

nos demais para o próximo processo. Nesta troca de processos, a dinâmica, quando vários processos são desalojados, eles podem criar um grande espaço para um processo maior, ou menor, permitindo assim alocações de diversos tamanhos, dificultando a possibilidade de não haver espaço para um novo processo. Porém isso não quer dizer que nunca irá acontecer, para isso existem outras abordagens que veremos a seguir.

Mas nesta abordagem de troca de processos, tendo a figura 4 como exemplo, o processo é iniciado tendo apenas um alocado (processo A), além do sistema operacional (figura 4-a). Na sequência, como existem espaços vagos, o processo B (figura 4-b) e o processo C são alocados (figura 4-c). Na figura 4-d, o processo A foi finalizado, deixando um espaço maior vago, sendo assim, ali o processo D é alocado (figura 4-e). Após isso, o processo B termina (figura 4-f) deixando mais um espaço vago, permitindo que o processo E seja alocado (figura 4-g).

Figura 4 - Arquitetura de troca de processos.



Fonte: Tanenbaum, 2009.

4.3.4 Gerenciamento com Paginação

Os endereços de memória lógica são divididos em unidades virtuais denominadas como páginas, sendo que seus respectivos espaços na memória física são chamados de molduras, devidamente mapeadas para gerência do MMU (Unidade de Gerência de Memória). Esta abordagem de gerencia-

mento pode causar falhas na alocação das páginas nas molduras, para tanto os algoritmos devem prever problemas de alocação visando alocar corretamente as memórias. Vamos conhecer abaixo as várias formas de gerenciamento de memória utilizando a abordagem de Tanenbaum (2009):


O algoritmo com a abordagem mais óbvia para substituição de páginas, considerado o **algoritmo ótimo**, é o que seria capaz de desalojar os dados da página que não está sendo utilizada, fazendo com que os dados da execução atual sejam mantidos, removendo apenas dados não utilizados. Porém, este algoritmo deverá prever também quando estes dados poderão ser utilizados, pois para uma alocação de grande volume, ele precisará de diversas páginas. (TANENBAUM, 2009)

Sendo assim, este tipo de algoritmo precisa remover as páginas que serão utilizadas o mais tarde possível e esta é uma tarefa extremamente complexa para um sistema operacional. Desta forma, o algoritmo ótimo nada mais é do que um método de avaliação de outros algoritmos de alocação de páginas, pois ele serve apenas como alvo à atingir. Ele é uma forma de medir o quanto é são os outros algoritmos, pois hoje é computacionalmente inviável desenvolver uma abordagem capaz de executar com o devido primor esta tarefa. (TANENBAUM, 2009)



Saiba mais

Para conhecer mais sobre o histórico dos sistemas operacionais e como funcionavam os ambientes computacionais de forma limitada, ao menos uma vez você deve assistir ao filme *Piratas do Vale do Silício*, que conta a história de Steve Jobs (Apple) e de Bill Gates (Microsoft) na criação dos seus respectivos sistemas operacionais, o Macintosh e o Windows. Disponível em: <http://www.imdb.com/title/tt0168122/>



Uma das abordagens mais simples para a troca de páginas é o uso do **algoritmo FIFO** (*first-in-first-out*), que utiliza o conceito de “fila”, onde a primeira página a “entrar na fila” será a primeira a ser definida para a alocação. O sistema operacional armazena esta fila em uma lista de páginas de memória. As páginas mais antigas ficam no início da lista e as alocadas depois na sequ-

ência, fazendo com que a última página seja a mais “nova”, se comparada com a primeira, a mais antiga. Se houver uma falha, quando uma nova instrução deve ser armazenada, a página mais antiga da fila é liberada para a inserção da nova informação (TANENBAUM, 2009).

A partir da remoção desta página mais antiga, as demais páginas da fila “andam mais um quadro” dentro da lista. Porém, nesta abordagem, a página antiga removida pode conter dados necessários para a próxima execução fazendo com que seja necessária mais uma alocação, fazendo com que caia em um ciclo de problemas. (TANENBAUM, 2009)

Existe então, **algoritmo segunda chance**, um algoritmo que serve para suprir a necessidade deixada pelo algoritmo FIFO, visando evitar que páginas importantes sejam substituídas na sequência de filas do FIFO. O que esta abordagem faz é dar uma nova chance à última página da fila. Se o sistema operacional verificar que esta página, que segundo o FIFO deveria ser substituída, é consideravelmente utilizada, ela será mantida e a próxima página da fila que será realocada (TANENBAUM, 2009).

O algoritmo verifica a última posição da fila. Se for constatado que esta página já foi referenciada em algum momento, ela recebe esta nova chance e é limpa para uma próxima verificação. Após isso, ele passa para a próxima posição. Caso o algoritmo verifique todas as posições da fila e todas estiverem referenciadas, ele inicia novamente sua busca, novamente pelo final da fila, que como este foi limpo nas iterações anteriores. Ele não estará referenciado, fazendo com que ele seja alocado. (TANENBAUM, 2009)

Por fim temos o **algoritmo relógio**, uma abordagem de gerenciamento de memória que atua de forma cíclica, visando suprir a necessidade que o de segunda chance deixa, pois é possível ainda que várias páginas sejam adicionadas como sendo últimas, fazendo com que estas páginas, que podem ser importantes ainda sejam removidas. Para resolver isso, o algoritmo relógio faz com que a fila seja um círculo, fazendo com que o início da mesma, ou seja, a página mais antiga da fila, seja definida pelo “ponteiro do relógio”. Desta forma, sempre que for dada uma “segunda chance” para a última página, o ponteiro ainda estará definindo quem é a última posição para a próxima iteração, repetindo os procedimentos ciclicamente até que uma página zerada pela verificação de “segunda chance” seja encontrada (TANENBAUM, 2009).

Outra solução é o uso do **algoritmo LRU** (*least-recently-used*), que faz alocações na página menos utilizada recentemente. Este algoritmo, visando atingir o algoritmo ótimo, lista as páginas que estão a mais tempo sem alocação, deixando reservados estes espaços para novas alocações. Esta abordagem mede o quanto cada página é utilizada, partindo assim do princípio de que as páginas mais utilizadas até o momento serão justamente mais utilizadas no futuro, fazendo com que o contrário denote futuras faltas de uso, tornando estas disponíveis nas futuras alocações (TANENBAUM, 2009).

No LRU, as páginas com falhas pouco utilizadas, dinamicamente medidas em sua intensidade, são classificadas como disponíveis. Este algoritmo não é muito utilizado, pois a cada alocação, as páginas precisam ser atualizadas na última alocação, causando lentidão na busca das páginas com menos uso, tornando uma implementação correta e eficaz deste algoritmo, algo complexo e caro. (TANENBAUN, 2009)

Existe então uma abordagem do **algoritmo NRU**, uma versão simplificada do LRU, onde fica disponível para alocações a página não utilizada recentemente. Para definir se elas são utilizadas ou não, o sistema operacional classifica as páginas em quatro classes, definindo sequencialmente por pesos se elas foram referenciadas ou modificadas. Este é um algoritmo mais simples de entendimento e de implementação, tornando mais viável o seu uso. Assim como os outros, este algoritmo busca alcançar o ótimo, atingindo um nível satisfatório nesta tarefa. (TANENBAUN, 2009)

Outra abordagem é a do **algoritmo LFU** (*last-frequently-used*), que faz a alocação pelas páginas utilizadas menos frequentemente. Este algoritmo utiliza um contador de usos para as páginas, sendo este incrementado a cada uso ou zerado quando os dados são desalojados. Com estas informações, o algoritmo elege os menos utilizados para as próximas alocações. Este tipo de paginação pode não se mostrar tão eficiente pois as páginas mais recentemente alocadas serão justamente as menos valores no contador de uso, fazendo com que estas não sejam tão utilizadas. (TANENBAUN, 2009)

Por fim, temos ainda a opção de utilizar o **algoritmo MRU** (*most-recently-used*) em um sistema operacional para realizar o gerenciamento de memória para a substituição de páginas, que realiza a alocação nas páginas mais

utilizadas recentemente. Esta abordagem, próxima do LRU, mas distinta do algoritmo ótimo, busca apenas pelas páginas recentemente utilizadas, permitindo assim que as páginas não utilizadas continuem não sendo utilizadas. Sua implementação não é simples, tornando seu uso pouco viável, visto que em termos ela pode se contrapor ao algoritmo ótimo. (TANENBAUM, 2009)

Tabela 1 - Comparativo entre os algoritmos

ALGORÍTMO	FORMA	FUNCIONAMENTO
Ótimo	Não viável	Remove dados de páginas não utilizadas, mantendo os dados da execução.
FIFO	Fila	As primeiras tabela alocada será a primeira a ter dados desalojados, e assim sucessivamente.
Segunda Chance	Fila	Utiliza o mesmo conceito da Fila, mas é dada uma segunda chance para tabelas já utilizadas.
Relógio	Fila Circular	Transforma a fila em um círculo, onde as tabelas onde foram dadas a “segunda chance” os “ponteiros” armazenam onde a próxima rotação irá iniciar.
LRU	Último utilizado recentemente	Elege as posições utilizadas recentemente, buscando alocar nestes espaços.
NRU	Não utilizado recentemente	Evolução do LRU, aloca nas tabelas não utilizadas recentemente.
LFU	Menos frequentemente utilizado	Realiza as alocações nas posições menos utilizadas frequentemente.
MRU	Mais utilizado recentemente	Realiza as alocações nas posições mais utilizadas recentemente.

Fonte: Elaborado pelo autor, 2016.

Resumindo

Do ponto de vista do sistema operacional, o gerenciamento de memória é uma área crítica, visto que boa parte do desempenho de uma plataforma tecnológica se baseia na memória. A eficiência na gestão de memória torna-se

um ponto chave para garantir a velocidade e a capacidade de uma determinada aplicação.

O presente capítulo abrangeu cada um dos tipos de memória dentro da sua hierarquia. Você pode conhecer a função de cada um deles no ambiente computacional e como eles são capazes de transformar programas codificados em uma linguagem de programação em instruções capazes de serem executadas. Foi abordado ainda como é o gerenciamento de memória no sistema operacional, bem como os diferentes tipos de algoritmos que fazem o tráfego de dados.

O estudo profundo da área de memórias é crucial para o profissional envolvido com o meio tecnológico, sendo que é uma área em constante aprimoramento pois seu grande objetivo é atingir uma capacidade que cada um de nós possui: a memória. Exercite a sua se dedicando nos estudos!

5

Sistemas de Arquivos

CADA PLATAFORMA COMPUTACIONAL e por sua vez, cada sistema operacional possui ferramentas e abordagens distintas para operar cada uma de suas funcionalidades. Isso implica em formas diferentes na leitura e armazenamento de arquivos, dadas as diferenças tanto na parte física quanto lógica do ambiente.

Todas essas diferenças se aplicam também na forma de organização destes arquivos, fazendo com que as operações sejam diferentes. Com base nesta premissa surgem os sistemas de arquivos, padronizações que garantem a organização de como são feitas as operações em seus devidos sistemas operacionais e ambientes computacionais (TANENBAUM, 2009).

Neste capítulo será apresentado o funcionamento dos arquivos e diretórios dentro de um sistema operacional, bem como quais as funcionalidades destes artefatos. O capítulo abordará também como funciona o gerenciamento dos arquivos e suas operações. Por fim, você conhecerá diversos exemplos de sistemas de arquivos para os mais distintos sistemas operacionais.

Objetivos de aprendizagem:

- × Identificar o conceito de sistemas de arquivos em um sistema operacional;
- × Compreender o papel dos arquivos e diretórios para o funcionamento de um ambiente computacional;
- × Identificar como são gerenciados os arquivos no sistema operacional;
- × Conhecer os exemplos de diferentes padrões de sistemas de arquivos;

5.1 Arquivos

Os arquivos são informações organizadas de forma sistematizada que podem representar programas e dados gerados por estes programas. Eles são referências abstratas para organizar os dados armazenados com o objetivo de deixá-los disponíveis para futuros acessos. Sendo assim, os arquivos servem para que o usuário do sistema operacional consiga acessar suas informações, sejam elas dados, textos, aplicativos, vídeos, entre outros, de forma transparente, sem existir a preocupação com suas localizações físicas, tamanho, ou outras características (TANENBAUM, 2009).

5.1.1 Atributos

Os sistemas operacionais gerenciam e tratam seus arquivos de formas distintas. Porém, em geral, eles mantêm mesmo que de formas diferentes, organizações com nomes de arquivos, extensões e formatos, que são os atributos dos arquivos (TANENBAUM, 2009).

O principal atributo, do ponto de vista do usuário do sistema operacional, é o **nome**. Cada sistema operacional possui sua própria regra para o

tamanho de nomes de arquivos, uso de caracteres específicos, entre outros. Quando um determinado processo, dentro sistema cria um arquivo, seja ele um processo automático ou algo executado pelo usuário, é atribuído um nome para este arquivo.

Fazendo ainda parte do nome do arquivo, existe a sua **extensão**, um segundo atributo, definido pelos caracteres expostos após o ponto do nome do arquivo. Por exemplo, em sistemas operacionais Windows, no arquivo “volume.txt”, “volume” é o nome do arquivo propriamente dito e “txt” é sua extensão, que nos sistemas operacionais do mercado são relacionados aos arquivos do tipo texto (SILBERSCHATZ, 2008).

Saiba mais

Conheça neste artigo os principais formatos e extensões de arquivos:
<http://br.ccm.net/faq/3549-formatos-e-extensoes-de-arquivos>

Outro atributo comum nos sistemas operacionais é o **tamanho** do arquivo, dependendo do sistema é possível obter informações sobre o tamanho atual do arquivo, o tamanho máximo que o arquivo pode ter. É comum ainda existir os atributos sobre a **criação** do arquivo, quem criou e quando, bem como a **alteração** contendo quem alterou e quando foi feita a operação. Existem ainda vários outros atributos que um arquivo pode ter, conforme exposto na tabela 1, dependendo do sistema operacional.

Quadro 1 - Possíveis atributos de arquivo.

Campo	Significado
Proteção	Quem pode acessar o arquivo e de maneira
Senha	Senha necessária para acessar o arquivo
Criador	Id da pessoa que criou o arquivo
Proprietário	Proprietário atual
Sinalizador de somente-leitura	0 para leitura/gravação; 1 para somente leitura
Sinalizador de oculto	0 para normal; 1 para não exibir em listagens

Campo	Significado
Sinalizador de sistema	0 para arquivos normais; 1 para arquivo de sistema
Sinalizador de arquivo	0 para salvo em backup; 1 para ser salvo em backup
Sinalizador de ASCII/binário	0 para arquivo ASCII; 1 para arquivo binário
Sinalizador de acesso aleatório	0 para acesso seqüencial somente; 1 para acesso aleatório
Sinalizador de temporário	0 para normal; 1 para excluir o arquivo na saída do processo
Sinalizador de bloqueio	0 para destravado; não-zero para bloqueado
Comprimento do registro	Número de bytes em um registro
Posição da chave	Deslocamento da chave dentro de cada registro
Comprimento da chave	Número de bytes no campo-chave
Tempo de criação	Data e hora em que o arquivo foi criado
Tempo do último acesso	Data e hora em que o arquivo foi acessado pela última vez
Tempo da última alteração	Data e hora em que o arquivo foi alte- rado pela última vez
Tamanho atual	Número de bytes no arquivo
Tamanho máximo	Número de bytes até o qual o arquivo pode crescer

Fonte: Tanenbaum, 2009.

Em alguns sistemas operacionais, determinados tipos de arquivos permitem a criação de atributos próprios para serem lidos por outros programas. Estes atributos são chamados de **metadados**, ou seja, são “dados sobre dados”, informações adicionais inseridas sobre o arquivo. Estes metadados podem estar disponíveis para o usuário e os programas visando permitir a leitura e alteração destes dados auxiliares, ou podem ser metadados disponíveis no sistema de arquivos, sendo utilizados em processos internos do sistema para atribuir dados relacionados com os arquivos e diretórios do sistema de arquivos.

Os sistemas de arquivos costumam utilizar metadados também para armazenar dados auxiliares sobre os logs de operações visando criar estruturas de segu-

rança para recuperação. Estes logs armazenam os metadados com as informações sobre onde estão os blocos dos dados de arquivos se for necessário restaurar operações em que houveram falhas (DEITEL; DEITEL; CHOFFNES, 2005).




Você Sabia

Os arquivos de foto possuem um formato de metadados específico, chamado de EXIF (*Exchangeable Image file format*). Nele, as câmeras armazenam os detalhes da foto (comuns entre fotógrafos), data e hora, local, etc.

Faça o teste fazendo o upload de uma foto do seu celular ou câmera no site: <http://regex.info/exif.cgi>.

Ele irá extrair do EXIF os metadados do arquivo da foto, contendo os detalhes da configuração da câmera, e se for uma foto capturada por um smartphone com a opção de geolocalização habilitada, será exibido no mapa o local de origem da foto.



5.1.2 Operações

Dentro do sistema operacional, os arquivos passam por diversas operações que exemplificam quais são os processos em que estão envolvidos, ou que podem ser executados por meio deles. Novamente, cada sistema operacional possui suas próprias operações, e quando são as operações que coincidem, sua forma de operar são ainda mais distintas. Dentre estas operações padrão conheceremos as principais (TANENBAUM, 2009):

Criação: Momento inicial do processo, onde o arquivo é criado de forma vazia, contendo apenas seus atributos, sem seus dados.

Escrita: Processo que armazena os dados dentro do arquivo “vazio” já criado, ou sobre arquivos já escritos, acrescentando novos dados.

Exclusão: Operação que destrói o arquivo, visando liberar o espaço em disco para novos armazenamentos.

Abertura: Processo invocado antes da execução do arquivo, visa armazenar temporariamente os atributos e os endereços de memória relacionados com o arquivo para ter o acesso futuro otimizado.

Fechamento: Serve para remover todas as referências de endereços da memória quando os processos são finalizados e não mais necessários.

Leitura: Processo onde o conteúdo de dados são lidos.

Renomeação: Momento onde é possível alterar os nomes e extensões de arquivos.

Ver Atributos: Processo onde é possível selecionar atributos para serem lidos, ou pelo usuário ou pelo próprio sistema.

Escrever Atributos: Operação automática ou selecionada pelo próprio usuário onde é possível escrever dados nos atributos do arquivo.

Importante

Sempre que um dispositivo de armazenamento, adquirido com uma determinada capacidade, é aberto é possível verificar que nem toda a sua capacidade está sendo atingida. Por exemplo, um *pendrive* de 8 gigabytes possui na verdade 7,8 gigabytes para armazenamento. Isso ocorre pois o espaço “inexistente” é dedicado aos **Inodes**, que são estruturas de metadados dedicadas ao armazenamento dos dados, arquivos, grupos e permissões (SILBERSCHATZ, 2001)

5.2 Diretórios

Os arquivos são organizados dentro do sistema operacional utilizando o conceito de diretórios, ou pastas, que são artefatos lógicos responsáveis por estruturar os arquivos. Em alguns sistemas operacionais baseados em Unix, por exemplo, os diretórios também são arquivos, porém arquivos do tipo diretório. Nos diretórios ficam armazenadas as informações relacionadas com a localização física dos arquivos ali armazenados. Eles são as estruturas que obtêm os dados necessários para o acesso aos arquivos. Porém, os diretórios

não são apenas as “pastas” do disco em um sistema operacional, toda a gerência de arquivos é baseada em diretórios (TANENBAUM, 2009).

De forma parecida com a dos arquivos, nos diretórios também existem atributos, tratados como entradas. Estas entradas delimitam o tamanho, a criação, o nome e as datas de criação e edição, por exemplo. Da mesma forma que com os atributos, assim como nos arquivos comuns, os sistemas operacionais também possuem operações específicas para os diretórios. Existem as operações de **criação**, **exclusão** e **renomeação**, que possuem os mesmos objetivos destas operações em arquivos. Mas os diretórios em geral, dependendo do sistema operacional possuem ainda outras operações. Vamos conhecer agora as operações e as suas respectivas chamadas em ambiente Linux:

Opendir: Operação realizada para abrir um diretório, este processo é necessário antes de listar seus arquivos. O comando para abrir o diretório, disponível para ambientes de programação é: `opendir` (diretório), passando por parâmetro o diretório e retornando o ponteiro de memória relacionado ao diretório aberto.

Closedir: Operação para fechar o diretório. Com o diretório aberto, após o seu uso é necessário fechá-lo para liberar o espaço de memória. O comando para fechar o diretório, disponível para ambientes de programação é: `closedir` (diretório), passando por parâmetro o diretório.

Readdir: Operação de leitura de arquivos do diretório. Com o diretório já aberto, esta operação é realizada para ler os arquivos do diretório. O comando para ler o diretório, disponível para ambientes de programação é: `closedir` (diretório), passando por parâmetro o diretório e retornando o ponteiro de memória relacionado com o diretório lido.

Link: Processo de vinculação de arquivos em vários diretórios. Esta operação cria uma espécie de atalho dos arquivos em outros diretórios. Comando para criar o link, disponível nos terminais é: `link` (nome, diretório), onde é passado por parâmetro o nome do link e o diretório que será referenciado.

Unlink: Processo que desvincula os arquivos colocados como atalho em outros diretórios. Comando para desvincular o link, disponível nos terminais é: `link` (nome), onde é passado por parâmetro o nome do link e será desvinculado.

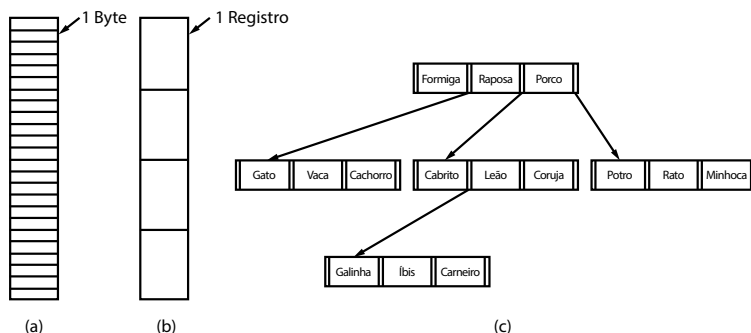
Já no sistema operacional Windows não possuímos comandos específicos para as operações citadas. Enquanto no Linux basta utilizar o prompt de comando para executá-los, no Windows, mesmo que de forma transparente, são executadas as mesmas funções, seja utilizando a interface gráfica ou em operações de “baixo nível”, quando desenvolvidas em programas que realizam estas operações.

5.3 Gerenciamento de Sistemas de Arquivos

Existem três formas principais para a organização de arquivos no sistema operacional (TANENBAUM, 2009). A primeira delas é a sequência de bytes (figura 1-a), deste modo, não-estruturada, não há uma forma de arquivos, é apenas uma coleção de bytes.

A segunda alternativa é organizar como uma sequência de registros (figura 1-b), neste modo há capacidades físicas para os registros definidas, sendo assim, o arquivo passa a ser uma coleção de registros com tamanho fixo. A terceira forma é organizar de forma contextual, como uma árvore de registros (figura 1-c). Neste modo cada bloco possui um campo-chave que indica o índice daquela posição para mostrar quais dados ocuparão aquele espaço.

Figura 1 – Forma de organização de arquivos.



Fonte: Tanenbaum, 2009.

Cada sistema de arquivos possui a sua forma de organização no momento de gerenciar as operações de arquivos dentro do disco. Estas formas de orga-

nização, além do sistema de arquivos, pode variar também de acordo com o tipo de arquivo.

A forma mais simples de organizar o armazenamento de dados em arquivos é a sequencial, onde os dados são armazenados ou lidos, um após o outro, onde não é possível acessar dados, sejam para gravação ou leitura, em pontos anteriores. Esta abordagem não é tão eficiente nem performática, com a evolução das unidades de armazenamento foi necessário evoluir este modelo, criando a possibilidade de realizar acesso direto à posição utilizando um registro com o endereço.

Utilizando o acesso direto para o acesso de dados, não é necessário ler ou armazenar os dados em ordem, como no modo sequencial, pois existindo o número do endereço da posição, estes dados podem ser acessados aleatoriamente. Porém, há ainda uma forma híbrida onde a leitura e armazenamento é feito por acesso direto, pelos endereços, e a partir de então os dados seguintes são acessados sequencialmente.

Porém, para que os dados sejam acessados, sejam eles para leitura ou gravação, o arquivo deve ser aberto e neste processo é lido o diretório, visto que neles residem as informações sobre os blocos onde os dados do arquivo estão alocados. Nos diretórios ficam definidos como o arquivo será acessado, seja ele de forma sequencial ou direta.

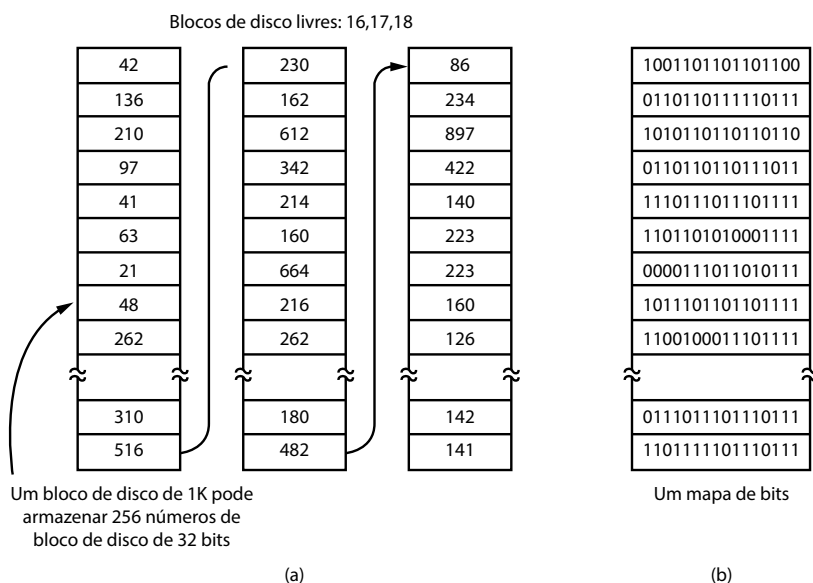
Para que os dados sejam gravados no disco, é necessário que exista espaço disponível para tal. Desta forma, os arquivos podem ser organizados em uma sequência de blocos linear ou dividido em blocos, sejam eles sequenciais ou não.

Estes arquivos podem ser armazenados em forma de blocos, tendo ainda o tamanho destes blocos capacidades fixas. Desta forma é necessário definir qual será o tamanho que estes blocos terão. Existe ainda uma forma de organização baseada em cilindros ou em páginas, ficando a cargo do sistema de arquivos dividir o disco desta forma.

Definido o tamanho dos blocos e páginas, o próximo passo da gerência de arquivos é definir quais posições estão livres para alocação. A primeira forma de efetuar esta verificação é armazenar uma lista encadeada com os blocos do disco (figura 2-a), onde a lista representa os blocos disponíveis livres tendo em cada um a mesma divisão proposta para todo o disco.

A segunda alternativa é criar uma mapa de bits (figura 2-b), que cria um bit representando cada um dos blocos. Em cada bit dos blocos é armazenado um zero para espaços livres e o número um para espaços em uso. Esta alternativa ocupa menos espaço pois cada bloco representado possui apenas um bit de tamanho.

Figura 2 - Formas de organização de blocos livres.



Fonte: Tanenbaum, 2009.

5.4 Sistemas de Arquivos nos Sistemas Operacionais

Cada sistema operacional implementa o seu próprio sistema de arquivos, sendo assim, arquivos e diretórios criados em determinado sistema operacional podem não ser compatíveis para sua execução em um outro sistema operacional que esteja atuando com um outro sistema de arquivos. O sistema de arquivos utilizado no sistema operacional é definido no momento onde o disco é formatado pela primeira vez.

Embora existam sistemas de arquivos intercambiáveis entre os sistemas operacionais, é importante saber qual o sistema de arquivos que o seu dispositivo está formatado para operar se for necessário abrir os seus arquivos em vários sistemas operacionais. Vamos conhecer abaixo os principais sistemas operacionais disponíveis no mercado (TANENBAUM, 2009):

Gigabyte: 1024 kilobytes

Terabyte: 1024 megabytes

Petabyte: 1024 terabytes

Exabyte: 1024 petabytes

5.4.1 FAT – *File Allocation Table*

O FAT é um sistema de arquivos do Windows que teve a sua origem no seu sistema operacional precursor, o MS-DOS. A estrutura do FAT cria tabelas que servem para definir o endereçamento das informações de cada arquivo. Cada tabela é dividida em blocos onde serão endereçados cada um dos arquivos. Um disco formatado em FAT tem sua estrutura dividida em blocos, agrupados em clusters.

Porém, com o decorrer do tempo, surgiram novas necessidades. As unidades de armazenamento passaram a ter mais capacidade e arquivos cada vez maiores precisaram ser armazenados. No FAT era possível armazenar no máximo 2 gigabytes, por isso foi lançado o FAT12, com capacidade de armazenar arquivos maiores, e após isso o FAT16 e por fim o FAT32, que, ainda embora suporte apenas arquivos de 2 gigabytes, pode armazenar ao todo 2 terabytes em seu disco.

Existiram ainda outras variações do FAT, como o VFAT, onde era possível criar arquivos com mais de 8 caracteres no nome, característica esta que por fim também foi incorporada no FAT32. Existe ainda o exFAT (*Extended File Allocation Table*), um sistema de arquivos ideal para pequenos discos como pendrives e HD's externos, onde a velocidade de transporte é maior e a capacidade de armazenamento também foi multiplicada, também denominado como FAT64.

5.4.2 NTFS – *New Technology File System*

O NTFS é um sistema de arquivos, também da Microsoft, voltado ao sistema operacional Windows, que em seu início era voltado ao público corporativo, onde os servidores eram formatados neste formato visando maior segurança, confiabilidade e flexibilidade. Neste sistema de arquivos foram implementadas novas funcionalidades de tolerância a falhas, onde a partir delas é possível recuperar dados caso ocorram quedas na alimentação do disco.

Neste sistema de arquivos foi implementado também o controle de acesso aos arquivos, a gerência de usuários e o controle de permissões. Este controle de acessos e permissões não se limita apenas a quais arquivos determinados usuários podem ter acesso, mas também é referente a cotas de espaço que cada usuário terá no disco, se este for formatado utilizando o sistema de arquivos NTFS.

Além disso, em comparação com o FAT, o NTFS permite o armazenamento de maior quantidade de dados, atingindo 16 exabytes. Isso deve-se ao fato de que, ao contrário do FAT, o NTFS não forma clusters para dividir o disco, ele cria uma base para cada um dos setores físicos do disco. Porém, por utilizar os setores físicos do dispositivo, discos removíveis como pendrives, se formatados em NTFS podem corromper seus arquivos com maior facilidade.

5.4.3 EXT – *Extended File System*

O EXT é um sistema de arquivos voltado aos sistemas operacionais baseados em Linux, que utiliza os padrões de gerenciamento de arquivos da família Unix. Este sistema de arquivos iniciou sua popularidade com a versão EXT2, quando evoluiu sua versão anterior (EXT) para atender aos padrões da família Unix para sistemas de arquivos. A partir do EXT2, discos formatados desta forma passavam a ser divididos por blocos, que são grupos de setores do disco com o objetivo de formar as dimensões alocáveis para armazenamento.

A terceira versão, o EXT3 adicionou novas funcionalidades ao EXT2, como a possibilidade de registrar cada uma das operações do disco fazendo com que seja possível que em caso de falhas exista a recuperação destes dados. Existe ainda uma nova versão, o EXT4, que possui as funcionalidades do seu antecessor, o EXT3, como a possibilidade de gerenciar até 1024 petabytes no disco e 1 exabyte por arquivo, armazenar hierarquicamente mais diretórios e o undelete, uma funcionalidade que delimita alguns arquivos como não-apagáveis.

5.4.4 UFS – *Unix File System*

O UFS, que dada a sua principal característica também é chamado de FFS – *Fast File System* (Sistema de Arquivos Rápido) é utilizado em sistemas operacionais baseados em Unix também dividido em blocos, nós e cilindros. Além de versões específicas do Linux, diversos outros sistemas operacionais da família Unix também implementam o sistema UFS, como os sistemas BSD e Solaris.

Os blocos são segmentados em blocos na parte inicial do disco, que servem para o boot, um superbloco e cilindros. Estes cilindros são agrupados em nós que gravam informações sobre os dados ali gravados. Este foi um dos primeiros sistemas de arquivos para Unix e serviu como base para o desenvolvimento dos demais, influenciando direta e principalmente na criação do sistema de arquivos EXT.

5.4.5 HFS – *Hierarchical File System*

O HFS é o principal sistema de arquivos para sistemas operacionais Apple, como o Machintosh e o Mac OSX. De forma parecida com os outros sistemas de arquivo Unix, o HFS também possui uma ferramenta de armazenamento de operações, porém em um catálogo contendo uma numeração para as operações.

Existe ainda uma nova versão, o HFS+, que além de operar no Mac OSX, também é utilizado sistema iOS, para dispositivos móveis da Apple. Esta nova versão, assim como nas versões mais atuais dos outros sistemas de arquivos, implementa maior número de arquivos, diretórios, caracteres nos nomes de arquivos, entre outras funcionalidades.

5.4.6 Outros Sistemas de Arquivos


Além dos sistemas de arquivos aqui citados, existem ainda outros sistemas voltados para operações específicas. Um deles é o VFS – *Virtual File System*, que serve como um sistema abstrato sobre outros sistemas de arquivos, fornecendo aos usuários uma opção única para acessar arquivos de diversos sistemas operacionais. Há também o NFS – *Network File System*, que compartilha os dados do sistema de arquivos de forma remota, via rede. Existe o

ReiserFS, que possui boot rápido do disco, sendo assim mais performático. E temos também o XFS – *X File System*, extremamente veloz, comumente utilizado para sistemas de banco de dados.



Importante

No meio corporativo, com a infinidade de computadores pessoais, notebooks, servidores e diversos outros dispositivos conectados em uma mesma rede, é muito comum que estes utilizem sistemas operacionais distintos. Isso fatalmente acarreta em sistemas de arquivos distintos. Porém é necessário ainda o compartilhamento de arquivos entre estes diversos computadores. Para isso existem servidores capazes de fornecer arquivos de forma independente do sistema de arquivos, como o **Samba**, uma ferramenta que fornece em sistemas Linux, arquivos simulando sistemas de arquivo Windows. Permitindo assim o acesso aos arquivos independente do sistema de arquivos em que o disco foi formatado.



Resumindo

Do simples ato de tentar copiar um arquivo em um *pendrive* do Linux para o Windows até efetuar a transferência de dados entre servidores, tudo isso interfere no uso de diferentes sistemas de arquivos. Isso pode impedir ou dificultar este tipo de processo.

Por isso, neste capítulo foi abordado o que são os sistemas de arquivos e como os arquivos e diretórios interagem com o disco no âmbito de um sistema operacional. Foi tratado também como funciona a gerência de arquivos no disco no sistema operacional utilizando os sistemas de arquivos, e por final foram expostos os principais sistemas de arquivos de mercado e os sistemas operacionais que os implementam.

Embora existam ferramentas para conversão e sistemas de arquivos interoperáveis, é importante o conhecimento sobre os diferentes sistemas para compreender a importância e o uso de cada um deles. Em ambientes complexos e em grandes aplicações, o conhecimento sobre o gerenciamento de arquivos é imprescindível para a compreensão das diferentes variáveis que o ambiente pode impor.

6

Entrada e saída (I/O)

CARO ALUNO, o gerenciamento de informações de entrada e saída é um dos assuntos mais importantes quando estudamos sistemas operacionais. A eficiência desse processo executado pelo SO vai fazer uma grande diferença nas interações executadas com o *hardware* da máquina. Sabemos que o computador é dividido em três processos simples de execução: entrada de dados, processamento de dados e saídas de dados.

Nessa aula, você irá aprender como se dá a interação do sistema operacional com os sistemas de entrada e saída de um computador e seus componentes como: barramentos, *slots*, placa de vídeo, placa de som, a comunicação com processadores e memórias, etc. Vamos entender a forma que cada dispositivo conversa com o processador e as memórias para realizarem suas operações. Elas são executadas por meio de técnicas computacionais, para que não haja ingerência nos processos que serão demandados pelo usuário e sistema operacional com o *hardware*.

Objetivos de aprendizagem:

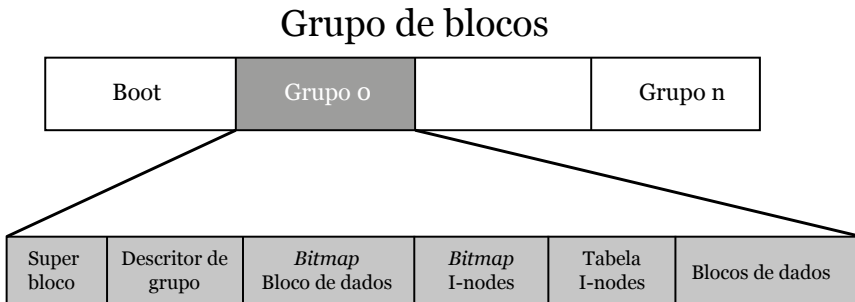
- × Compreender sobre gerenciamento de dispositivos de E/S.

6.1 Sistema de entrada e saída

Uma das principais funções de um SO é controlar as informações de entradas e saídas (E/S), conhecido também como I/O (*input/output*). Para realizar essas operações, o SO envia comandos para os dispositivos tendo como fim a captura de interrupções e tratamentos de erros para serem enviadas aos processadores - Unidade Central de Processamento (UCP). Os dispositivos precisam de autorização do sistema operacional para que as operações realizadas tenham uma sequência de acesso, que pode ser por prioridade, por processo, tamanho dos dados, etc. Para compreendermos como essa comunicação é realizada nos dispositivos de entrada e saída, vamos estudar os dispositivos em duas categorias genéricas: **dispositivos de blocos**, **dispositivos de caracteres**.

Interrupções: sinal de um dispositivo que tipicamente resulta em uma troca de contextos, o processador para de fazer o que está fazendo para atender o dispositivo que pediu a interrupção.

Figura 1 - Exemplos de blocos usados em discos rígidos, cada bloco com seu endereço.



Fonte: elaborado pelo autor, 2015.

- a) **Dispositivos de blocos:** técnica de entrada e saída que armazena as informações recebidas do processador em blocos de tamanho fixo, cada bloco com seu endereço. Essa técnica permite que as leituras possam ser realizadas de maneira independente. Dispositivos que utilizam essa técnica são os discos rígidos, e dispositivos de armazenamentos em geral, pois o volume de informações é grande e interdependente um dos outros, por isso, são utilizados *buffer* (memórias), para gravar as informações entre processadores e controladoras.

Buffer: é uma região de memória física utilizada para armazenar dados temporariamente enquanto são movidos de um lugar para outro.

- b) **Dispositivos de caracteres:** técnica de entrada e saída que não utiliza uma estrutura de blocos, as informações entre as controladoras dos dispositivos e o processador é realizada por meio de um fluxo de caracteres. Dispositivos como teclado, mouse utilizam essa técnica, pois priorizam a eficiência da comunicação e não o volume como os dispositivos que utilizam blocos. Os dispositivos de caracte-

teres não usam *buffer* (memórias), pois os fluxos enviados são em menores quantidades.

Para gerenciar todos os fluxos de informação, seja ela por blocos ou caracteres, o sistema operacional utiliza algumas técnicas. Vamos estudar algumas dessas técnicas: entrada e saída programada, comunicação via interrupção e acesso direto à memória (DMA).

- c) **Entrada e saída programada** – nessa técnica, os controladores fornecem os comandos para leitura e escrita de dados, cabe ao processador testar se a informação possui erros, se a comunicação com o *hardware* ou dispositivo foi realizada, e isso ocupa o tempo do processador que poderia estar realizando outras operações. A essa técnica dá-se o nome de verificação de *Polling*.

Polling: significa ler seu registrador de estado tantas vezes for necessário, até que satisfaça a condição.

Registrador: tem a função de armazenar dados temporariamente.

- d) **Comunicação via interrupção** – técnica de comunicação via interrupção é realizada via *software*. Todos os comandos são enviados para a controladora, que acessa o *hardware* via *drive* e, quando a execução do comando termina, uma mensagem via *software* é enviada para o processador avisando que o processo chegou ao fim. A grande vantagem dessa técnica é que o processador fica liberado para executar outras tarefas, otimizando assim o sistema operacional para gerenciar outras tarefas.
- e) **Acesso direto à memória** – esta é uma das técnicas mais eficientes de comunicação entre as controladoras e o processador,

pois toda a comunicação é realizada por meio de um canal de dados, onde a controladora acessa o processador sem precisar de permissão, o nome dado a essa técnica é: DMA- *Dinamic Access Memory*, acesso direto à memória. A eficiência dessa técnica está na forma de ligação das controladoras, que são ligadas fisicamente ao barramento de dados e endereços do computador. Segundo Tanebaum (2009), um sistema operacional pode utilizar um DMA somente se o *hardware* tem instalado um controlador de DMA, caso contrário não conseguirá realizar a comunicação com eficiência.



Importante

O uso da tecnologia de DMA (acesso direto à memória), proporcionou uma velocidade maior aos computadores e a gestão dos sistemas operacionais aos dispositivos de Entrada e Saída, pois houve menos desgaste do processador.

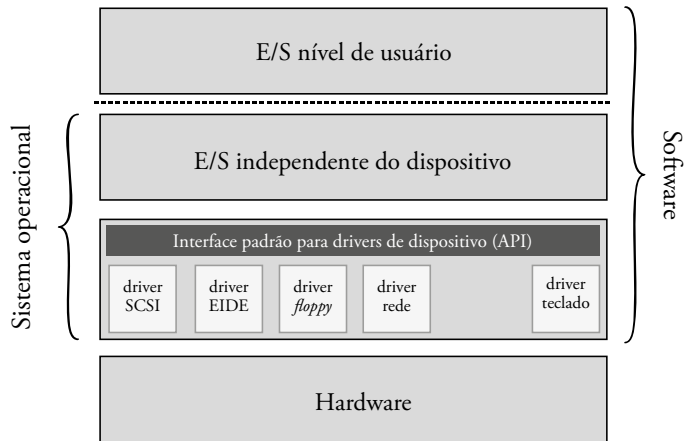


Os dispositivos estão ligados ao processador por meio de um barramento (filamentos metálicos que realizam a comunicação entre os dispositivos de *hardware*), e esses barramentos à CPU – Unidade Central de Processamento, por estarem no mesmo canal de comunicação seu acesso a memória se dá por meio da técnica de DMA.

Depois de estudarmos as técnicas de comunicação relacionadas ao envio de informações, acesso ao processador e memória, iremos ter uma visão mais abrangente de como tudo isso funciona na percepção de um especialista, e não mais de um usuário.

Na figura 2 podemos perceber uma divisão de todo o controle que um sistema operacional realiza para ter acesso ao *hardware*, esta divisão é realizada em 4 camadas, que são utilizadas para padronizar o acesso e controle dos dispositivos, permitindo que o usuário possa adicionar novos dispositivos sem a necessidade de outros *softwares* auxiliares.

Figura 2 – Gerenciamento de entrada e saída (I/O)



Fonte: TOSCANI, S.; OLIVEIRA, R. S.; CARISSIMI, 2010.

Cada camada descrita na figura tem uma função específica. Vamos compreendê-las estudando-as na sequência proposta pela figura:

- I. **E/S nível de Usuário** – quando instalamos um programa em um computador, as configurações de acesso de E/S ao *hardware* da máquina, por meio da programação, já realiza a configuração por meio do sistema operacional. Podemos visualizar na Figura 2 que esse é o último nível da camada, onde temos uma abstração transparente de acesso aos recursos, ou seja, o usuário irá enxergar o *hardware* sem a sua complexidade computacional. Essa camada é controlada por *softwares*, divididos em baixo nível, em que sua principal função é esconder do usuário as especificidades do *hardware*, e apresentar uma boa interface de comunicação, que seja simples, fácil de usar.
- II. **E/S Independente de dispositivo** - esse tipo de técnica é utilizada para qualquer dispositivo. Mas quais são estes dispositivos? São os drives na camada abaixo, podendo ser: um disco rígido, placa de som, placa de rede, mouse, teclado, controladoras, etc. Como essa camada consegue manipular qualquer

dispositivo? A resposta é simples, cada dispositivo possui uma interface de acesso (realiza a comunicação entre as camadas).

Nessa camada, os níveis mais altos de *software* só devem tomar conhecimento de um erro, caso não consiga tratar com as camadas mais baixas. Podemos destacar os seguintes serviços executados nessa camada: os tratamentos de erros, bufferização de dados, escalonamento de E/S, acesso as controladoras, entre outros. Todas as operações de acesso aos dispositivos como: teclados, mouses, discos, etc. são realizados por esse *software* por meio dessa interface.

Escalonamento: tem a tarefa de realizar um agendamento das tarefas utilizado pela CPU, dando prioridade para alguns processos, quando achar necessário.

Importante

Buffer é uma memória que armazena dados temporários de escrita e leitura. São utilizados quando existe uma diferença entre as taxas de envio e a capacidade de recebimentos do dispositivo. A bufferização é o processo de guardar essas informações até que o dispositivo tenha capacidade de recebê-las (www.hardware.com.br/termos/buffer)

III. Interface padrão drive de dispositivos: a interface de um drive de dispositivo tem por objetivo implementar as rotinas de acesso aos dispositivos instalados na placa mãe do computador. A ligação entre eles e a placa controladora é realizada por meio da interface controladora-dispositivo, que é de baixo nível, no caso o *hardware*.

Nos drives, os *softwares* de E/S realizam a programação dos registradores internos instalados nas controladoras e implementam as formas de comunicação com os dispositivos. Na imagem abaixo, podemos visualizar um disco rígido em que a interface com a placa controladora é realizada por meio de um cabo.

Figura 3 - Interface de comunicação de disco rígido



Fonte: Shutterstock, 2016.

IV. Hardware: essa é a última camada de acesso e refere-se ao *hardware* instalado na máquina. A instalação desse dispositivo requer um *drive* que é fornecido pelos fabricantes. Os *drives* de cada fabricante devem ser configurados de acordo com o sistema operacional instalado e as suas especificidades. Podemos encontrar vários dispositivos que integram essa camada: mouse, teclados, discos rígidos, placas, monitores. Todos esses dispositivos possuem *drives*, que são acessados pelas controladoras, e controladas via *software*.

Saiba mais

Para conhecer mais sobre os processos de comunicação entre o *hardware* e os processadores, acesse a página onde poderá encontrar mais informações sobre assunto o: http://fubica.lsd.ufcg.edu.br/hp/cursos/so/LabSO/ent_saida.html.

6.2 Dispositivos, tipos e exemplos

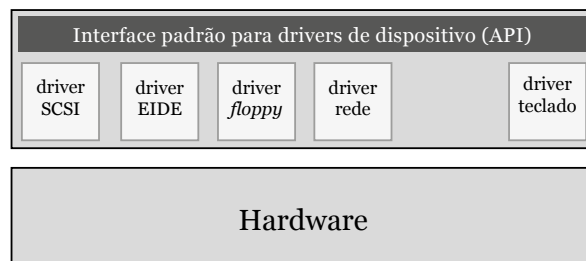
Para entendermos como as informações transitam entre os dispositivos dos computadores, temos que saber quais realizaram esse processo. Os dispositivos que fazem parte dos sistemas de entrada e saída são: *device drives* (*driver*), controladores (ou interface), dispositivos de entrada e saída, discos magnéticos.

6.2.1 Device Drivers (*driver*).

Os drives de dispositivos ou (*Device Drives*) têm a função de realizar a comunicação com o dispositivo físico por meio de códigos, e estes códigos são escritos ou desenvolvidos na linguagem *Assembly* para cada sistema operacional. Analise a Figura 4 em que temos a interface padrão, e cada uma é ligada a um *drive* (discos rígidos, disquetes, placa de rede, etc.). Quando instalamos um novo dispositivo, uma placa de rede por exemplo, o sistema operacional precisa realizar a comunicação com esse novo dispositivo, para isso, solicita que o usuário instale o *drive* respectivo. Um outro exemplo para entendermos: um *drive* de um mouse, quando instalado, envia informações referente a sua movimentação no computador, de acordo com o lado pressionado do periférico.

Um *drive* de disco rígido enviará informações de endereços de arquivos, setores, trilhas, cilindros, cabeça, etc. Por esse motivo, cada dispositivo deverá possuir o seu *drive*, mas há casos que um *drive* consegue realizar a comunicação com vários dispositivos.

Figura 4 – Comunicação dos *drives* com o *hardware*



Fonte: TOSCANI, S.; OLIVEIRA, R. S.; CARISSIMI, 2010.

Segundo Tanebaum (2009), para realizar a comunicação com o *hardware*, os *drives* dependem das controladoras, que por sua vez recebem comandos dos sistemas operacionais denominados de *System Calls*, e traduzem aos comandos específicos desenvolvidos pelos fabricantes para serem executados pelas controladoras, e as controla com os dispositivos de *hardware*. A principal função dos *drives* dos dispositivos é a realização da comunicação com o *hardware* por meio das controladoras. Vamos conhecer sobre as controladoras no próximo tópico.

System Calls é uma chamada executada pelo sistema operacional para acesso a um hardware, essa chamada é autorizada para um usuário especial, denominado de modo privilegiado.

6.2.2 Controladores (ou interfaces)

As controladoras ou interfaces são dispositivos eletrônicos (*hardware*), responsáveis pela manipulação direta nos dispositivos de entrada e saída. Realiza a comunicação dos sistemas operacionais com eles. Devido ao grande número de acesso a determinados dispositivos de *hardware*, as controladoras possuem memórias e registradores.

Figura 5 - Controladoras para dispositivos de entrada e saída



Fonte: Shutterstock, 2015.

Quando realiza a operação de leitura no dispositivo, o controlador armazena uma quantidade de *bits* em sua memória interna, e verifica a ocorrência de erros, não havendo erros na leitura de dados, o bloco de comandos é transferido para a memória principal do computador (RAM).

A grande maioria dos dispositivos utiliza-se de blocos de comandos ou vetores de blocos para controle de leitura e escrita. Para facilitar a quantidade de dados, alguns dispositivos como discos rígidos utilizam-se de uma técnica chamada de DMA para a transferência de dados entre a controladora e a memória principal. Para avisar que

os processos enviados para as controladas estão executando, finalizando ou esperando retorno, as controladoras realizam essas tarefas enquanto a UCP (unidade central de processamento) poderá realizar outras tarefas.

Na figura 5, podemos observar vários *chips* conectados a placa mãe, alguns deles são controladoras designadas para comandar o *hardware* da máquina. Temos controladoras no mouse, teclados, placa de rede, placa de som, etc.

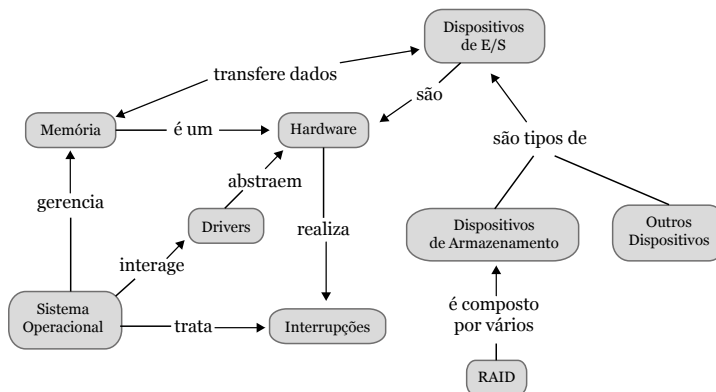
Importante

Algumas controladoras podem ser acopladas no próprio dispositivos de I/O das placas mães mais antigas. Em placas mais atuais esses controladores podem controlar diversos dispositivos e somente algumas destas são capazes de realizar o controle de I/O de vários dispositivos.

6.2.3 Dispositivos de entrada/saída

Os principais dispositivos que encontramos no computador, que realizam a comunicação entre usuário e o computador, ou seja, o mundo externo, são os dispositivos para entrada de dados (teclado, mouse), dispositivos de saída de dados (impressoras) e dispositivos para entrada e saída de dados (*modems*, discos, fitas).

Figura 6 – Gerenciamento de entrada e saída realizada pelo sistema operacional



Fonte: TANENBAUM, 2009.

Podemos entender de forma gráfica os diversos dispositivos que são controlados pelo SO. Para acessar um dispositivo de entrada e saída, o SO interage com os *drives* fornecidos pelos fabricantes ou já instalados pelo SO. As informações de acesso aos dispositivos (I/O) podem ser armazenados em uma memória temporária que transfere para os dispositivos (placa, mouses, teclados etc) as informações necessárias pedidas pelo processador, ou SO. Quando há a necessidade de interromper as execuções, o SO utiliza-se de técnicas de interrupções para que os processos mais importantes sejam executados. Fazem parte dos dispositivos de entrada e saída(I/O) os de armazenamento como: discos rígidos e outras tecnologias.

Segundo Stallings (2002), os periféricos como teclados, mouses, impressoras, etc., são acoplados ao computador por meio de um componente de *hardware* chamado interface, elas são fixadas aos barramentos da placa mãe, que são os canais ou linhas que realizam intercomunicação entre todos os dispositivos de um computador.

Para controlar todas as informações que trafegam nessas vias, temos um controlador, que é um tipo de processador, projetado para realizar essa função de controle dos dados. A comunicação ocorre por meio de blocos de informações ou palavra a palavra, realizada via controladoras e a unidade central de processamento. Podemos encontrar dois tipos de dispositivos: estruturados, e os não estruturados.

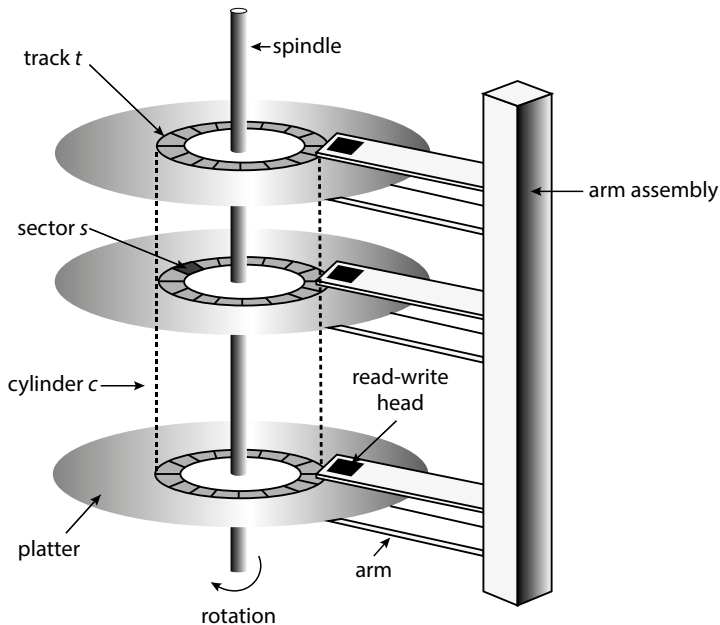
Dispositivos estruturados: os dispositivos estruturados utilizam a técnica de armazenar informações em blocos de tamanho fixo, que pode variar entre 128 e 1.024 bytes. Os blocos podem ser lidos ou gravados de forma independente e utiliza o acesso direto e o acesso sequencial. O acesso direto é realizado por meio de um endereço, um exemplo que utiliza essa técnica são os discos rígidos. No acesso sequencial, o dispositivo deve percorrer toda a sequência para encontrar a informação, um exemplo seria as fitas magnética.

Dispositivos não estruturados: podem enviar e receber informações em sequência, e esta sequência não precisa estar estruturada no formato de blocos. A sequência enviada não pode ser consultada após seu envio. Podemos exemplificar os dispositivos não estruturados como: terminais e impressoras.

6.2.4 Discos magnéticos

Discos rígidos são dispositivos magnéticos eletromecânicos que são utilizados para armazenar uma quantidade maior de informações. A memória é um dispositivo muito caro, por isso o disco rígido é utilizado para gravar a maior parte das informações nos computadores. Por ser um dispositivo mecânico e eletrônico, os discos são lentos comparados às memórias, que são totalmente eletrônicas e propiciam um desempenho maior para os computadores.

Figura 7 - Organização física do disco magnético



Fonte: TOSCANI, S.; OLIVEIRA, R. S.; CARISSIMI (2010).

Um disco é dividido em vários pratos ou discos sobrepostos, e podem ser encontrados mais de 8 discos em um único compartimento. Eles possuem, em sua superfície, uma película magnética em que os dados são gravados. Um dispositivo mecânico chamado cabeçote de leitura realiza as gravações e leitura de dados no disco, conforme ilustrado na Figura 7, com movimentos retilíneos, e toda a estrutura dos discos são cobertas.

Um disco é dividido em trilhas e setores por um sistema operacional.

Esta divisão serve para organização dos dados que são gravados logicamente

Figura 8 - Novas tecnologias de discos SSD - *solid-state drive*



Fonte: Shutterstock, 2015.

nestas unidades, outra forma de divisão é a criação de outras unidades realizando partições nos discos.

No mercado, podemos encontrar opção de discos totalmente eletrônicos chamados de: SSD- *solid-state-drive*, ou unidade de estado sólido, que utiliza a mesma tecnologia encontrada nos *pendrives*, possuindo

menor peso e os tempos de acessos são bastante reduzidos em relação aos discos eletromecânicos. Outro fator importante desta tecnologia é a diminuição da quantidade de erros de leitura e escrita, e durabilidade.

Saiba mais

A tecnologia de discos SSD está evoluindo rapidamente, e possui características singulares relacionados à antiga tecnologia (discos eletromecânicos). Vamos conhecer mais um pouco sobre essa tecnologia. Acesse: <http://www.infowester.com/ssd.php>

Uma forma de armazenar os dados de forma segura é proposta por uma técnica chamada de Raid (*Redundant Array of Independent Disks* ou Conjunto Redundante de Discos Independentes), consiste em instalar vários discos em um servidor para satisfazer a necessidade de desempenho e confiabilidade. Podemos encontrar Raid implementado em estruturas lógicas (Raids 0, 1, 2, 3, 4, 5, 6, 10), cada configuração entre 0 e 10 pode apresentar configurações controladas por *hardware* ou *software*.

Resumindo

Quando falamos em dispositivos de entrada e saída, logo vem a nossa mente o conceito de mouse, teclado, como dispositivos de entrada e monitores, impressora como dispositivos de saída, não é errado pensar dessa forma, mas agora sabemos como se dá esse processo de forma mais técnica.

O sistema operacional realiza a comunicação do processador com os dispositivos de *hardware* por meio de blocos de comandos ou caracteres, essa comunicação depende do tipo de *hardware* acessado. A comunicação é realizada com as controladoras dos dispositivos via técnicas de acesso aos processadores, podemos nomeá-las de: DMA, acesso direto à memória, a mais utilizada; comunicação via interrupção; e entrada e saída programada. Podemos verificar que toda a comunicação é realizada de forma transparente aos usuários, pois cada dispositivo instalado no computador possui recursos para serem acessados em nível de usuário e em baixo nível, por meio das interfaces de dispositivos. Para realizar essa comunicação as controladoras instaladas exercem papel importantíssimo, pois controlam todas as informações trocadas com os processadores, enviando sinais de status das controladoras para os processadores para avisar como estão sendo processadas as informações requisitadas.

O principal desafio dos sistemas operacionais é a realização das comunicações de forma eficiente e sem erros, para que os usuários possam ter um sistema estável e sem surpresas quando acessar o *hardware* da máquina.

7

Gerenciamento de Usuários

CARO ALUNO, o assunto gerenciamento de usuários por meio de um sistema operacional é um trabalho extremamente importante no contexto dos Sistemas Operacionais, pois abordará assuntos relacionados à segurança e todas as permissões que poderão ser efetuadas pelo administrador do SO como: acessos dos recursos gerenciados pelo sistema operacional, pastas, arquivos, espaços em discos, autorização de acessos, etc. Para gerenciar esses recursos, os sistemas operacionais permitem gestar usuários por meio de comandos e permissões, criando grupos, usuários, gerenciamento de usuários, autenticação e tipos de autenticação. Vamos aprender como esses processos funcionam nos sistemas operacionais Windows e Linux.

Objetivo de aprendizagem:

- × Criar e gerenciar usuários, grupos e permissões.

7.1 Usuários

A criação de usuários em um sistema operacional requer planejamento, pois está relacionado ao organograma da empresa, quem é o funcionário, qual setor trabalha, quais horários ele atua, quais softwares e pastas deve acessar. Todos esses itens devem ser levados em consideração na hora de criar usuários em um sistema operacional, a criação de uma tabela com todos os detalhes das atividades profissionais de cada usuário é de extrema relevância para que haja segurança. Por serem multiusuários, os SOs podem ser acessados por inúmeros usuários simultaneamente, sem que nenhum atrapalhe a atividades de outro. Imagine uma empresa que possui 1000 funcionários, todos acessando recursos disponibilizados pelo sistema operacional, a gerencia de tudo isso deve ser projetada, caso contrário a rede de acesso ficara um caos.

Multiusuários: define a característica de sistemas operacionais que permite acesso simultâneo de múltiplos usuários aos recursos do Sistema Operacional.

Recursos do sistema: podem estar relacionados a parte lógica do Sistema como: pastas, *softwares*, arquivos, permissões de leitura e gravação e a parte de *hardware*: memórias, processadores, discos, etc.

Todo os sistemas operacionais possuem recursos para serem protegidos contra: invasões e acessos não autorizados de pessoas. Para gerenciá-los, os SOs possuem usuários especiais, que são criados para administrarem, podemos dividir esses usuários em três tipos distintos: super usuário, usuários do sistema, usuários comuns.

Superusuário:

Nos sistemas operacionais Windows, temos o usuário administrador que possui autorização completa sobre todos os procedimentos de um sistema opera-


cional, seu perfil é criado quando o sistema operacional é instalado, mas é possível dar acesso a outros usuários com o mesmo perfil por meio deste. No sistema operacional Linux, temos o usuário *root*, que possui privilégios parecidos com os administradores dos Sistemas Operacionais do Windows. Todos os superusuários são considerados administradores dos SOs, pois permitem o gerenciamento de todas as contas dos usuários e acesso aos recursos do Sistema Operacional como: criação de usuário, senha, compartilhamentos, permissão de leitura e escrita, etc.

Cada processo (programa em execução) do sistema tem um dono, um proprietário, que determina quem pode e quem não pode utilizar esse recurso, quem determina essa gerencia são os superusuários.



Importante

Tornar-se dono de um arquivo pode ser necessário quando se deseja acessar ou alterar um arquivo pertencente a outro usuário. Embora superusuários sejam capazes de alterar arquivos de outros usuários, alguns arquivos só podem ser modificados pelo próprio dono e mais ninguém, nem mesmo administradores.



Usuário do sistema

É um usuário especial, criado pelo próprio Sistema Operacional para gerenciar tarefas específicas, e não necessita estar logado no sistema para controlar os serviços que lhe são designados. Portanto esse tipo de usuário não precisa de senha.

Usuários comuns

São os usuários que terão acesso aos recursos do sistema, e são criados pelo superusuários, que podem conceder permissões de ler, gravar, modificar arquivos, acessar programas, acessar espaço físico no disco, impressoras, entre outros recursos. A este usuário não é concedida a possibilidade de administrar o SO, somente se o administrador achar conveniente dar um perfil de um superusuário, mas são casos que devem ser bem planejados.

Comandos para criação de usuários no Linux

Para podermos entender como são criados usuários, vamos utilizar o Sistema Operacional Linux Debian 8.2 já instalado e realizarmos os comandos

para essa operação. Todo usuário que é criado no Linux possui uma identificação que é única, chamada de UID (Identificação Usuário), que será o número que identificará quem é o usuário no SO. Outra identificação importante é chamada de GID (Identificação Grupo), que é a identificação primária do usuário.

UID: número utilizado para identificar cada usuário criado.

GID: número utilizado para identificar um grupo de usuário criado.

A criação e configuração de usuários é realizada por um arquivo de configuração que está no diretório `/etc/passwd`, e possui sete campos distribuídos da seguinte forma: `[login]:[senha]:[UID]:[GID]:[Nome Completo]:[home]:[shell]`

Todas essas configurações são gravadas quando são criados os usuários, e servem para identificar quem é esse usuário no sistema, permitindo ao administrador mapeá-los, caso haja algum tipo de invasão. Na Figura 1 podemos observar os detalhes de como os arquivos documentam um usuário no Linux, todas essas informações são adicionadas no arquivo `/etc/passwd`

Figura 1 - Arquivo `passwd` do SO Linux.

```
cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
proxy:x:13:13:proxy:/bin:/bin/sh
www-data:x:33:33:www-data:/var/www:/bin/sh
backup:x:34:34:backup:/var/backups:/bin/sh
list:x:38:38:Mail List Manager:/var/list:/bin/sh
irc:x:39:39:ircd:/var/run/ircd:/bin/sh
gnats:x:41:41:Gnats Bug-Reporting System (admin)/var/lib/gnats:/bin/sh
nobody:x:65534:65534:nobody:/nonexistent:/bin/sh
```

Fonte: Elaborado pelo autor, 2016.

Observe na figura 1 que no lugar da senha existe um “X”, isso significa que a senha está criptografada e é guardada em um local chamado de `/etc/shadow`, no qual um usuário comum não tem permissão nem mesmo de leitura.

Para adicionar um usuário no sistema, é necessário ter privilégios de administrador, no caso ser usuário `root`.

Criando um usuário

Os dois comandos básicos utilizados para a criação de usuários são “`adduser`” e o “`passwd`”, que permitem adicionar novos usuários e suas senhas.

Vamos ao comando:

```
# adduser Agnaldo
```

Criando uma senha para o usuário Agnaldo

```
# passwd Agnaldo
```

O sistema operacional Linux é *case-sensitive*, isso quer dizer que ele diferencia letras maiúscula de minúsculas, no caso o usuário “Agnaldo”, é diferente do usuário agnaldo.

Alterando senha

Para alterar a senha criada, o próprio usuário poderá realizar essa operação. Basta utilizar o comando “`passwd`”, desde de que saiba a senha antiga.

Apagando um usuário

Para apagar um usuário, utiliza-se o comando “`userdel`” e o nome do usuário que deseja remover. No nosso exemplo:

```
# userdel Agnaldo
```

Para aprimorar esse comando, temos a opção de remover os diretórios ligados ao usuário criado. Esse diretório é feito no diretório `home`, o comando “`userdel`” apenas apaga o usuário, mas não remove as pastas relacionadas à ele, que são criadas juntamente com a inserção de cada novo usuário. Por questão de segurança, devemos remover o usuário apagando também o seus subdire-

tórios, para isso, adicione o parâmetro “-r” e o comando “sudo”, que permite privilégios de um superusuário, como em:

```
# sudo userdel -r Agnaldo
```

Importante

Para conhecer mais comandos de criação de usuários e mais opções de parâmetros para ampliar seu conhecimento, acesse o site (<http://www.hardware.com.br/tutoriais/usuarios-grupos-permissoes/>), e leia esse artigo sobre criação de usuários.

Comandos para criação de usuários no Windows

Para criar um usuário no Windows 8 de forma gráfica, basta irmos em painel de controle e acessar o ícone “Usuários”, e clicar “Adicionar usuários”. Nesse nosso exemplo vamos criar usuários por meio de linhas de comando ou prompt de comando. Para isso, acesse o menu iniciar, na opção executar digite “CMD” e pressione Enter. A seguinte tela irá aparecer.

Figura 2 - Arquivo do Windows



Fonte: Elaborado pelo autor, 2016.

Criação de Usuários

Para criar usuários no Windows basta digitar no prompt de comando, conforme a tela da figura 2. Digite o seguinte comando e tecla Enter: **net user**

username password /add. No comando acima, substitua “username” com um nome personalizado e “password” com uma senha forte para proteger sua conta.

Note que você deve executar o comando acima no prompt de elevação (como administrador). Executar o comando sem direitos de administrador irá mostrar “Acesso negado”.

Alterando Senha

Para alterar a senha de um usuário escreva o comando **net user e pressione Enter.** Este comando irá retornar uma lista das contas dos usuários que estão cadastrados. Procure pelo nome da pessoa que deseja alterar a senha. Escreva no Prompt de comando **net user** “nome do usuário”.

Uma dica é não esquecer nenhum dos espaços em branco, pois, caso isso aconteça, o comando não irá funcionar.

Apagando um Usuário

Abra o Prompt de comando como administrador. Digite o seguinte comando e tecla Enter: **net user username /delete.**

Para este comando substitua “username” com o nome de conta de usuário que você gostaria de excluir e confirme a exclusão. Assim que enviado, o comando irá lhe dar a opção de escrever uma nova senha para o usuário.

7.2 Grupos

Criando grupos no Linux

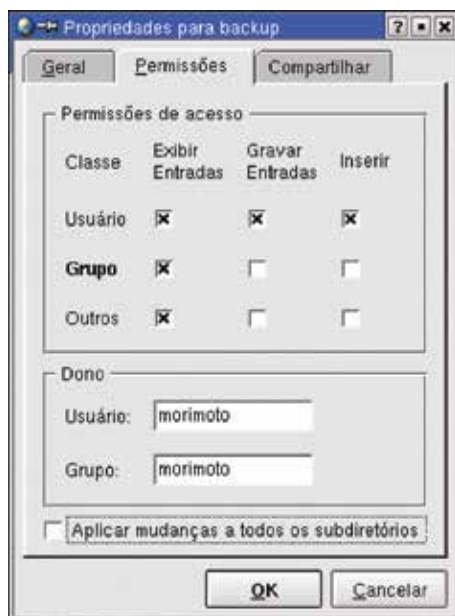
Quando criamos um usuário com uma senha, devemos especificar quais recursos esse usuário terá em nosso sistema operacional, para isso, devemos dar permissões de acesso aos recursos dos sistemas, caso não façamos essa configuração, o usuário terá apenas o acesso a uma pasta do diretório /home do sistema. Antes de tratarmos do assunto permissão, temos que entender o que são grupos em um sistema operacional.

O arquivo /etc/group controla o acesso dos usuários a seus grupos cadastrados e permite que vários usuários tenham permissão de acesso ao mesmo arquivo do sistema, visto que cada usuário pode pertencer a um ou mais grupos e acessar arquivos pertencentes ao seu grupo ou a outros grupos.

Para pertencer a um grupo, devemos configurar três opções que a grande maioria dos sistemas operacionais oferecem: o dono do grupo, o grupo ao qual pertence, e outros. Vamos verificar cada uma destas opções:

Donos ou Proprietários: o dono da pasta ou arquivo refere-se a quem criou, essa configuração é definida pelo administrador (root), para um determinado arquivo ou diretório. O nome do dono do arquivo/diretório é o mesmo do usuário utilizado para entrar no sistema. As permissões de acesso ao arquivo/diretório somente podem ser modificadas por ele, pois ele tem essa prerrogativa. O “id” do usuário e o nome do grupo são gravados /etc/passwd e /etc/group. Na Figura 3, abaixo, podemos visualizar como é configurada essa permissão em que possibilita aos donos e proprietários as permissões de ler, gravar e executar uma pasta ou diretório.

Figura 3 - Tipos de permissão.



Fonte: Elaborado pelo autor, com base em Morimoto, 2008.

Grupos: é o grupo ao qual o usuário dono pertence e seu arquivo de configuração está localizado em /etc/group. Esse recurso possibilita que vários

usuários tenham acesso ao mesmo arquivo do sistema. Cada usuário pode pertencer a um ou mais grupos, podendo acessar arquivos pertencentes ao mesmo grupo que o seu, mesmo eles sendo de outro dono.

Outros: segundo Tanenbaum (2009), outros são grupos externo de usuários que não pertencem ao grupo padrão do arquivo ou diretório. Quando é criado um novo usuário, o grupo que ele pertencerá corresponde ao mesmo de seu primeiro grupo, sendo o ID do grupo chamado de GID (Group ID). O usuário, no entanto, poderá ser inserido em mais grupos se desejar.



Você sabia

Todas as configurações que podem ser adicionadas para um arquivo/diretório são realizadas pela administração do sistema operacional. Em suma temos três opções que podemos configurar um usuário:

usuário dono: é o proprietário do arquivo;

grupo dono: é um grupo, que pode conter vários usuários;

outros: se encaixam os outros usuários em geral.



7.3 Permissões

As permissões são de fundamental importância na administração de usuários. Elas servem para proteger o sistema operacional, arquivos e pastas de usuários mal intencionados e, principalmente, dar estabilidade no uso dos recursos disponibilizados pelo Sistema Operacional

Manipular permissões é uma atividade que requer bastante atenção, pois possui uma certa complexidade, e lida com uma variedade de configurações.

- × permissão de ser dono do arquivo ou diretório;
- × permissão de pertencer a um grupo de usuários;
- × permissão de outros.

Cada privilégio concedido pode ser dividido em três níveis de permissão: leitura (r), escrita (w) e execução (x), são essas configurações que podem ser distribuídas para donos de arquivos, grupos e outros.

- × permissão de leitura (r)
- × permissão de escrita (w)
- × permissão de execução (x)

Os arquivos são criados com uma permissão padrão, e podem ser modificados via comando. Veremos, mais adiante, a especificação nos arquivos /etc/profile, que define para todos os usuários do sistema, e no arquivo do usuário /home/usuario/.bash_profile no Debian. As permissões padrões que temos para arquivos e diretórios são as seguintes:

Permissões padrões para diretórios:

- × dono - ler (r), escrever (w) e executar (x)
- × grupo - ler (r) e executar (x)
- × outros - ler (r) e executar (x)

Permissões padrões para arquivos:

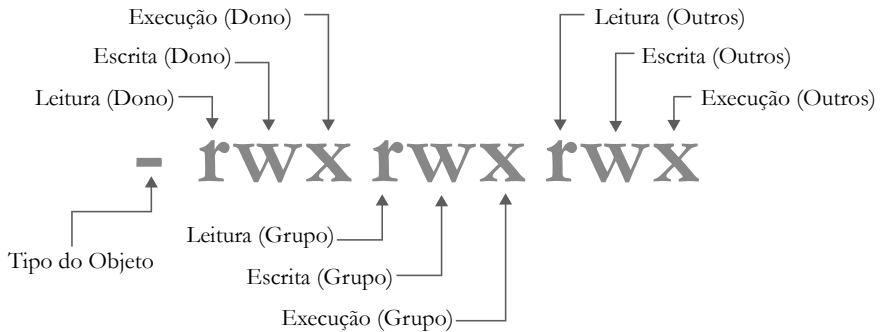
- × dono - ler (r) e escrever (w)
- × grupo - ler (r)
- × outros - ler (r)

No Linux, temos essa permissão associadas a um número que pode variar de 0 a 7, para cada tipo de permissão. Analise a Figura 4 abaixo: temos três letras r w x (dono), r w x (grupo), r w x (outros), com essas permissões podemos variar as configurações para cada usuário.

Configuração de Permissão:

- 4 → Leitura
- 2 → Gravação
- 1 → Execução

Figura 4 - Tipos de Permissão no Linux.



Fonte: Elaborado pelo autor, 2016.

No livro *Sambando com Linux*, Ferrari (2009) enumera três números que indicam, respectivamente, as permissões de acesso para o dono, grupo e para os outros. Cada número representa a soma das permissões desejadas, sendo que:

0 : Sem permissão alguma. Acesso negado.

1 : Permissão apenas para executar (não é possível ler o arquivo ou alterá-lo, apenas executar um programa) ou, no caso das pastas, permissão apenas para ver a lista dos arquivos dentro da pasta, sem poder abri-los.

4 : Apenas leitura. Se usado em uma pasta, o usuário não conseguirá listar o conteúdo, ou seja, conseguirá abrir os arquivos apenas se indicar o caminho completo.

5 (4+1): Ler e executar (no caso de um arquivo) ou ver os arquivos e abrí-los, no caso de uma pasta.

6 (4+2): Leitura e gravação. Assim como no caso do "4", se usado em uma pasta faz com que o usuário não consiga listar o conteúdo, apenas acessar os arquivos diretamente.

7 (4+2+1): Controle total.

Figura 5 - Combinação de permissões no Linux

4 2 1 4 2 1 4 2 1
rwx rwx rwx
7 7 7

Fonte: Elaborado pelo autor, com base em Tumblr.

Para um usuário que tivesse controle total sobre os arquivos /diretório, poderíamos configurar com as permissões 777, (7) dono, (7) grupo, (7) outros.

Para servir de exemplo, vamos dar permissão para o usuário “Agnaldo”, já criado, para que possa ter acesso total a todos os diretórios e subdiretórios de uma pasta chamada “teste”. O comando ficaria assim:

```
# chmod 775 teste
```

Para aplicar permissão recursivamente (aplicar a permissão no diretório e todos os seus arquivos e sub-diretórios), use a opção -R:

```
chmod -R 777 teste
```

No sistema operacional Windows podemos realizar essas permissões

Figura 6 - Opções para configurar permissões no Windows



Fonte: Elaborado pelo autor, 2016.

clikando nas opções disponíveis para permissões dos arquivos. Na figura 6 podemos observar as opções que temos para configurar as permissões nos arquivos dos sistemas operacionais Windows. As modificações e tipos de permissões que são realizadas são definidas para cada arquivo que o administrador achar necessário.

Cada arquivo ou pasta possui permissões associadas e que podem restringir o acesso de usuários do Sistema. Como administrador você pode modificar essas permissões de acesso especiais. Para modificar é preciso ser

o proprietário do objeto ou ter permissão como administrador. Para definir, exibir ou alterar ou remover especiais no Windows siga os seguintes passos:

1. Clique com o botão direito do mouse no objeto que deseja conceder permissões avançadas e em Propriedades e clique na guia Segurança.
2. Na caixa escolha a opção: Avançado e em Alterar Permissões.
3. Na guia Permissões, execute os seguintes procedimentos:
 - × Defina permissões de acesso especiais para um grupo ou usuário adicional. Clique em Adicionar. Em Digite o nome do objeto a ser selecionado, digite o nome do usuário ou do grupo e clique em OK.
 - × Abra ou altere permissões de acesso especiais para um grupo ou usuário existente. Clique no nome do grupo ou do usuário e clique em Editar, para alterar as permissões.
 - × Para remover um grupo ou usuário existente e suas permissões especiais, realize o seguinte procedimento: clique no nome do grupo ou usuário e clique em Remover. Caso o botão Remover não estiver disponível desmarque a caixa de seleção Incluir permissões herdáveis provenientes do pai do objeto e clique em na opção Remover.

Saiba mais

Para conhecer mais sobre as permissões de usuários Linux, acesse o site jsbusinnes (<http://www.jsbusiness.com.br/foca/iniciante/ch-perm.htm>)

7.4 Gerenciamentos de usuários

Neste tópico, vamos aprender como gerenciar esses usuários criando suas contas. O Sistema Operacional não se restringe apenas para criar as permissões aos usuários, mas serve para configurar o espaço que cada um deverá obter com sua conta. Uma pessoa poderá ter seus próprios diretórios, personalizar seus espaços e realizar suas próprias configurações. O administrador do sistema tem como grande desafio o gerenciamento dessas configurações.

Para criar, gerenciar ou eliminar contas de usuários no Sistema Operacional Linux, é necessário estar autenticado com o super usuário root (ou outro usuário que tenha privilégios de administrador). Como vimos anteriormente, apenas como superusuário é possível ter privilégios para manipular outras contas, do contrário, a segurança do sistema seria seriamente comprometida, pois qualquer usuário poderia criar, alterar ou apagar contas.

Importante

○ site <http://www.infowester.com/usuarioslinux.php> apresenta uma variedade de comandos do Sistema Operacional Debian para serem praticados. Confira!

Criando Usuários

MORIMOTO (2008), no artigo intitulado “Gerenciamento de usuários, grupos e permissões”, descreve as principais permissões que podemos encontrar no Linux.

adduser -group grupo: comando utilizado para criar um grupo de usuários, poderá ser utilizado o comando **addgroup**;

adduser -home diretório usuário: comando utilizado para definir o diretório “home” do usuário. Se esse parâmetro não for usado, o sistema criará o “home” no diretório padrão em */home/nome_do_usuario*).

adduser -uid número usuário: comando utilizado para criar ID sequenciais. Em algumas situações, é necessário ter o controle do número ID dos usuários criados para manter uma organização melhor sobre os usuários. Quando usuários são criados, o sistema geralmente adiciona a eles UIDs sequenciais, mas você pode especificar o UID que quiser, usando o parâmetro **uid** seguido de um número.

adduser -gid número usuário: semelhante ao comando acima, mas especifica manualmente um grupo para o usuário ao invés de criar um parâmetro.

adduser -ingroup grupo usuário: comando que adiciona o usuário criado a um grupo já existente, ao invés de criar um novo grupo para ele.

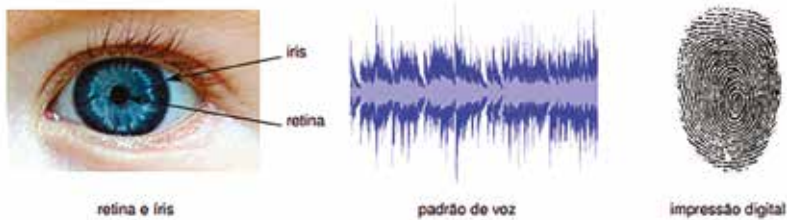
7.5 Autenticação e seus tipos

Segundo Tanenbaum (2011), o objetivo da autenticação consiste em identificar os diversos usuários que usufruem do sistema computacional e verificar, por meio da autenticação, se realmente o usuário é quem afirma ser. Para isso, várias técnicas são utilizadas. Inicialmente, a autenticação tinha por objetivo identificar apenas os usuários para garantir que somente eles, devidamente credenciados, teriam acesso ao sistema. Hoje esse cenário mudou e também é necessário identificar o sistema para o usuário.

Quando um usuário acessa o *site* de um banco via *internet*, ele gostaria de ter a certeza que aquele sistema é realmente do *site* do banco preterido e não um sistema falso. Outro exemplo seria a instalação de um *drive* em uma máquina, o sistema operacional deve assegurar que realmente provem de uma fonte confiável e que não possui um *software* malicioso sendo instalado. Vamos tratar das principais técnicas de autenticação utilizadas para confirmar se os atores realmente são o que alegam ser. Segundo Maziero (2013), as técnicas usadas para a autenticação de um usuário podem ser classificadas em três grandes grupos:

1. **SYK – Something You Know** (“algo que você sabe”): esta técnica consiste em analisar o nome e a senha de um usuário. É considerada uma autenticação falha, pois facilmente alguém poderá repassar a senha e usuário ou mesma ser roubada.
2. **SYH – Something You Have** (“algo que você tem”): esse tipo de autenticação é geralmente uma chave criptográfica, ou algum dispositivo material, como um *smartcard*, um cartão magnético, um código de barras, etc. São técnica mais eficazes que SYK, mas também podem ser roubados ou copiados.
3. **SYA – Something You Are** (“algo que você é”): se baseiam em características associadas a um usuário, como seus dados biométricos, impressão digital, diferente do SYH que usam dispositivos, nessa técnica de autenticação são usados elementos intrinsecamente associados a um usuário.

Figura 7 - Exemplo de características biométricas.



Fonte: Elaborado pelo autor, com base em Mazieiro, 2013.

A grande maioria dos sistemas utilizam autenticação de por *login/senha* (SYK). Sistemas mais modernos SYH utilizam cartões, *smartcards* ou técnicas SYA, com a biometria, como os sensores de impressão digital. Sistemas computacionais modernos usufruem de uma mescla das técnicas de implementação, chamada de autenticação *multi-fator*. Como o sistema bancário, que utiliza uma senha e um cartão para apurar a autenticidade do usuário, alguns sistemas empregam cartões, biometria e senha, geralmente são locais de segurança máxima, onde não pode haver falhas de segurança.

Importante

Multi-fator: esse sistema de senha misturando vários tipos de autenticação garante que sua senha não será violada, um invasor pode ter sua senha de acesso, mas não uma identificação biométrica.

Resumindo

Tivemos a oportunidade de aprender várias técnicas nessa aula, e como melhor gerenciar um Sistema Operacional Linux. Vimos o que são grupos de usuários e seus diferentes modelos existentes, aprendemos sobre os tipos de grupos que podemos criar. Estudamos também que o gerenciamento desses usuários é fundamental para que o sistema operacional possa trabalhar sem interrupções. Sobre permissões e criação de usuário, vimos a

importância de contextualizarmos o cenário empresarial para podermos criar os grupos, quem é dono, quem pertence a outros.

A autenticação de usuários por meio de senhas, ou cartões e biometria, são recursos utilizados pelos gerenciadores de Sistemas Operacionais quando exige uma atenção maior no quesito segurança das informações. O papel de gerenciamento de usuários requer prática e conhecimentos técnicos avançados para administrar diversas situações, por isso deverá praticar exaustivamente todos os comandos e analisar seus efeitos na administração do Sistema Operacional.

8

Segurança

CARO ALUNO,

Neste capítulo vamos tratar sobre segurança, um ponto cada vez mais importante no setor de tecnologia da informação. Vamos conhecer as principais vulnerabilidades que um sistema operacional pode apresentar, bem como as formas de prevenção. Vamos conhecer também o que gera estas vulnerabilidades e quais são as formas de garantir o menor esforço e prejuízo na recuperação destas informações, conhecendo assim o conceito de segurança da informação.

Da mesma forma que a sua casa, a sua rua, ou a sua empresa está sujeita a falhas, problemas e invasões, o ambiente computacional também sofre este tipo de vulnerabilidades. Seja por questões climáticas, por má intensão de indivíduos ou por falhas em geral, todo ambiente complexo deve ter formas de se prevenir.

Para isso, este capítulo apresentará os conceitos de técnicas de segurança da informação, visando introduzir o conceito de gerência de riscos e vulnerabilidades. Após isso, o capítulo apresentará os principais aspectos de segurança em sistemas operacionais, abrangendo os tipos de falhas, o que pode gerar estas falhas e as principais formas de prevenção. E por fim, conheceremos ainda as falhas de segurança e suas formas de prevenção quando estamos em um ambiente distribuído, onde dois ou mais computadores estão ligados em rede, aumentando consideravelmente a probabilidade de falhas de segurança.

Objetivo de aprendizagem:

- × Conhecer os conceitos de segurança da informação;
- × Identificar as principais causas de falhas em ambientes tecnológicos;
- × Compreender como ocorrem as falhas de segurança;
- × Conhecer os mecanismos de prevenção de falhas.

8.1 Técnicas de Segurança da Informação

A gestão da segurança da informação é constituída pelos processos para garantir a proteção de informações, sistemas, recursos e dispositivos contra erros, ataques, desastres e acesso à informações não autorizadas. A segurança da informação busca ainda reduzir o impacto e a probabilidade de incidência destas falhas de segurança (COELHO; ARAÚJO; BEZERRA, 2014).

Estes conceitos são baseados em normas ISO/NBR, propondo a implantação de um catálogo de serviços de segurança. Estes serviços são voltados à confiabilidade, autenticidade, integridade, não repúdio, conformidade, controle de acesso e disponibilidade.

A **confidencialidade** tratada na segurança da informação é voltada à proteção de dados contra acessos não autorizados, utilizando criptografia e gerenciamento de permissões de usuários. A **autenticidade** é ligada com a garantia de que o trâmite de dados seja autêntico, sendo assim um dado enviado deve ser integralmente recebido, garantido que seja exatamente a mesma informação.

A garantia de **integridade** é focada em proteger os dados de alterações e exclusões realizadas por usuários não autorizados. O **não repúdio** procura prevenir que seja possível garantir que um dado enviado possa ser provado o seu envio, o mesmo servindo para a recepção.

A **conformidade** procura garantir que as transações entre os dados estão de acordo com as políticas institucionais da organização. A conformidade faz com que estas transações sigam legislações do país, as normas da empresa e os regulamentos em vigor, por exemplo.


O **controle de acesso** busca restringir o acesso aos dispositivos do meio corporativo utilizando identificação e autorização para proteger o uso de usuários não autorizados. Já o **controle de disponibilidade** garante que estes dispositivos estejam disponíveis ao usuário que possui a sua permissão no local e momento em que este deveria acessá-lo.

A gestão da segurança da informação recomenda ainda a criação de um papel dentro da organização, o **CISO** (*Chief Information Security Officer*), o gerente de segurança da informação. Um profissional focado na garantia dos serviços de segurança dentro de um setor de tecnologia da informação.



Importante

O papel do CISO segue a tendência das profissões de alta gerência, como o CEO (*Chief Executive Officer*), que equivale a um diretor executivo; CFO (*Chief Executive Officer*), que equivale ao diretor financeiro e CIO (*Chief Information Officer*). Embora possa ser compartilhado com o CIO, o papel do CISO é específico, direcionado à garantia da segurança da informação.



O CISO deve desempenhar seu papel criando o plano diretor de segurança, um documento que norteia o gerenciamento dos ativos para manter a política de segurança da informação. Ele deve ainda realizar a manutenção contínua nos processos de análise de riscos e vulnerabilidades, garantindo a segurança dos dados dentro da organização (SÊMOLA, 2003).

8.2 Segurança em Sistemas Operacionais

Problemas gerados por falhas de dispositivos físicos de um computador são facilmente resolvidos, pois em sua grande maioria basta substituir este equipamento. O grande problema é quando a falha é gerada no sistema de arquivos, quando há alguma falha de arquivos, ou seja, quando há perda de informações.

Ambientes computacionais no meio corporativo possuem dados e informações de alto valor e sigilosas. É importante proteger estas informações dos possíveis problemas, bem como do uso das mesmas por usuários não autorizados.

Equipamentos podem chegar ao fim da sua vida útil, ou podem conter falhas de fábrica, quedas de energia, falhas em equipamentos, fenômenos da natureza como chuvas e raios, que podem gerar falhas no sistema fazendo com que arquivos sejam perdidos. Arquivos quando utilizados no âmbito profissional possuem custo intangível, sendo assim, arquivos perdidos podem causar grandes prejuízos à uma corporação.

Estes são problemas de segurança ligados às vulnerabilidades físicas que um sistema pode sofrer. Existem ainda as vulnerabilidades de segurança ligadas com usuários maliciosos que buscam destruir ou roubar informações de sistemas computacionais (DEITEL; DEITEL; CHOFFNES, 2005).

Tanenbaum (2009) divide os problemas de segurança em três grupos. O primeiro é o grupo das **ações divinas**, e são relacionadas às ações do tempo como enchentes, terremotos, incêndios, maremotos, temporais e ações de animais sobre o equipamento. Ou seja, são ações que hoje podemos caracterizar como sendo ações do tipo “naturais”. O segundo grupo, o grupo de **erros de hardware** estão ligados ao mal funcionamento dos equipamentos, falhas nos discos e erros de telecomunicações. O terceiro e último grupo está ligado aos **erros humanos**, quando estes são originados pelo usuário na montagem incorreta dos equipamentos, mal uso, perda de equipamentos e enganos em geral.

Para isso, ao longo dos anos, diversas abordagens foram desenvolvidas para garantir a integridade dos dados. Vamos conhecer agora as principais vulnerabilidades conhecidas, bem como as abordagens de segurança voltadas à prevenção destes problemas.

8.2.1 Tipos de vulnerabilidades

Qualquer ambiente computacional no meio empresarial está sujeito a interferências de usuários intrusos, que são classificados por Tanenbaum (2009) em dois grupos: o grupo dos intrusos passivos e ativos, dividindo ainda em quatro categorias estas intrusões. Os intrusos **passivos**, apesar de não estarem autorizados, querem apenas ler arquivos sem a devida permissão. Já os intrusos **ativos** querem fazer alterações, cópias ou exclusões em arquivos onde eles não possuem esta permissão.

Dentre as categorias das intrusões, uma das classificações indica que os intrusos podem fazer alterações do tipo “**bisbilhotice casual**”, onde os usuários não envolvidos com o setor tecnológico, ao compartilharem o mesmo ambiente ou computadores, leem e acessam as informações umas das outras, sem ter a devida permissão. Há ainda a categoria de **espionagem por pessoas de dentro**, neste caso, usuários envolvidos com o setor tecnológico, munidos de permissões mais privilegiadas buscam, por motivações pessoais, quebrar a segurança dos ambientes computacionais visando assim acessar informações alheias puramente pelo desafio pessoal de atingir esta meta.

Há ainda uma terceira categoria, a da **tentativa determinada de fazer dinheiro**, onde usuários técnicos atuam como intrusos dentro de ambientes computacionais do setor bancário e buscam burlar os mecanismos de segurança para alterar valores de contas bancárias, alterar taxas de juros e furtar valores. Por último há ainda a categoria de **espionagem comercial ou militar**, que atua de forma organizada, por motivação de concorrência, oposições e coalizões entre países e grandes companhias, visando espionar projetos, valores e planos.

Todas estas categorias utilizam as mais diversas abordagens para se infiltrar nos sistemas. Uma das mais conhecidas é a de **cavalo de Troia**, que de atua de forma análoga à estratégia grega para invadir a cidade de Troia utilizando um grande cavalo de presente. Ela envia um arquivo aparentemente inofensivo (Cavalo de Troia) ao sistema terceiro, sendo que ele possui instruções maliciosas para executar procedimentos indesejados e inesperados. Estes comandos procuram roubar dados do sistema terceiro de forma transparente, para que estes sejam utilizados posteriormente pelo criador do “Cavalo de Troia”.

Contudo, a abordagem mais conhecida de intrusão ainda é o uso de **vírus**. Estas abordagens utilizam instruções maliciosas dentro de terminados arquivos, sendo que estas “infectam” outros arquivos do sistema de destino, fazendo com que o vírus se prolifere pelo sistema. Estes vírus são enviados junto a anexo de e-mails, junto com arquivos de áudio, vídeo, jogos, para que estes infectem outros.

Um vírus pode ser de **setor de boot**, onde teste pode infectar o setor de inicialização do sistema operacional no disco do ambiente, fazendo assim com que este vírus controle o sistema. Um vírus pode ser também do tipo **transiente**, onde o arquivo infectado faz com que o vírus entre em ação apenas quando é executado. Sendo assim, quando finalizada a execução do arquivo, a ação do vírus propriamente dita também é interrompida. Diferente do vírus transiente, há também o vírus **residente**, que quando alojado no sistema este entra em ação sempre que a máquina está ligada, finalizando apenas quando esta é desligada (DEITEL; DEITEL; CHOFFNES, 2005).

Existem ainda os vírus do tipo **bomba lógica**, onde os usuários técnicos, com conhecimento avançado sobre o ambiente computacional da corporação, até então devidamente empregados na empresa onde existe o sistema, criam programas maliciosos. Estes problemas a princípio não causam problema algum, mas caso estes usuários sejam demitidos, ou por qualquer outro motivo, quando acharem conveniente, eles “ativam” esta bomba lógica, fazendo com que arquivos sejam corrompidos ou excluídos, por exemplo (TANEMBAUM, 2009).

Os vírus, em geral, costumam ter o seu desenvolvimento complexo, exigindo alto conhecimento do sistema operacional e de programação, porém existe uma outra categoria de vírus mais simples, os vírus de **macro**. Esta categoria é composta por vírus simples, criados utilizando linguagens de programação de fácil aprendizado, baseado em arquivos comuns, como Excel e Visual Basic. A ação destes vírus é iniciada quando um arquivo simples, como uma planilha do Excel, utilizando macros em Visual Basic, acessam, alteram ou excluem alguns arquivos.

Há também os ditos vírus de **código-fonte**, que faz uma varredura no disco buscando programas e altera o seu código-fonte procurando incluir seus códigos e fazer com que estes vírus sejam espalhados. Outra forma de vírus,

mais complexa de detecção, é o **polimórfico**, que altera sua forma cada vez que é instalado. Estes vírus polimórficos, apesar de ter a mesma funcionalidade, cada vez que infecta arquivos muda a sua forma, fazendo com que não seja identificado facilmente por softwares antivírus.

Existem os vírus **criptografados**, eles infectam as máquinas estando criptografados. Quando são executados, eles são descriptografados, assim eles também são mais difíceis de identificação por meio de antivírus. Há ainda o vírus **furtivo**, este vírus também busca não ser identificado. Este busca alterar os locais do sistemas operacionais que teriam condições de identificá-lo. Existe também o vírus de **multipartite**, que se divide em vários, infectando assim diversas partes do sistema operacional e dificultando que ele seja detectado.

Há ainda o vírus **blindado**, que é desenvolvido para que sua identificação por parte do antivírus seja complexa. Um vírus blindado costuma ser compactado e sua infecção dentro do sistema operacional é distribuída desta forma (SILBERSCHATZ, 2008).

Por fim, há a abordagem de **porta dos fundos**, uma espécie de vírus que faz acesso a todos os itens do sistema infectado. Em um acesso via porta dos fundos, os intrusos podem acessar qualquer informação, ver os registros realizados no sistema, editar, excluir e recuperar arquivos, entre outros (DEITEL; DEITEL; CHOFFNES, 2005).

Boot: Processo do sistema operacional realizado quando o disco é inicializado e o sistema operacional é carregado na memória.

8.2.2 Abordagens de Segurança

A abordagem primária para prover segurança em sistemas operacionais é o controle de acessos por meio de autenticação. Desta forma, cada usuário do sistema operacional possui as suas devidas permissões, sejam ela de escrita, lei-

tura, alteração ou exclusão. Este controle de usuários geralmente é baseado em **senhas**, onde cada usuário possui sua chave para autenticação de sua escolha.

Porém, existem formas de burlar este tipo de autenticação por senha. A abordagem em si já é consideravelmente fraca do ponto de vista da segurança, pois os usuários costumam escolher chaves um tanto quando simples demais para uso como forma de segurança. Visando suprir este problema, vários sistemas criam regras com quantidade de caracteres para enriquecer a chave de acesso.

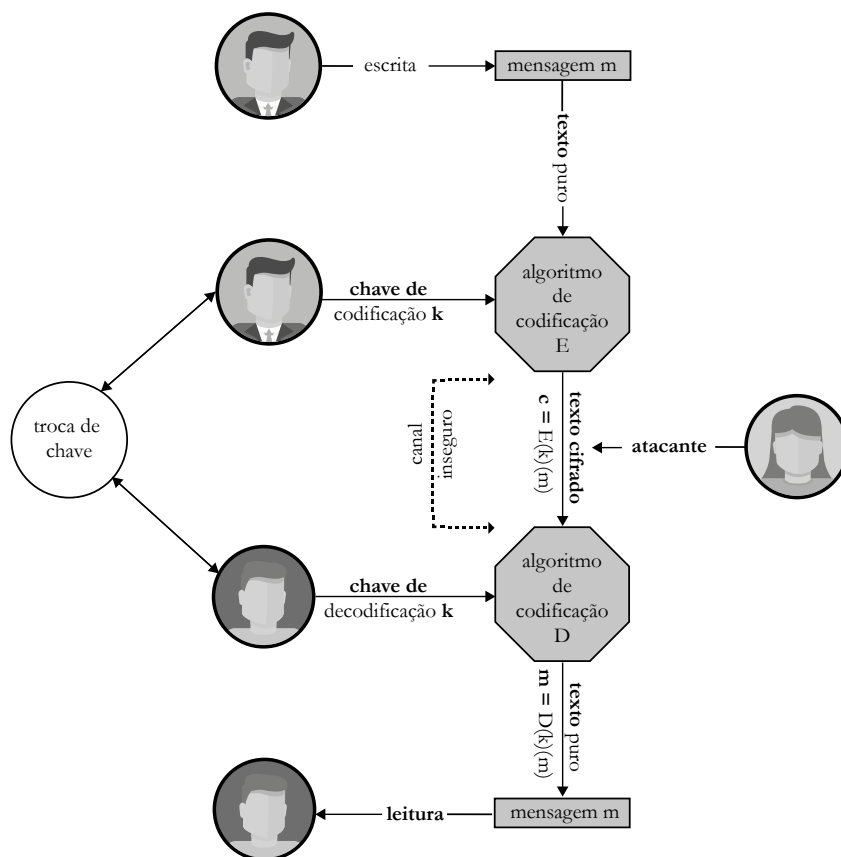
Mas da mesma forma ainda é possível burlar senhas, por mais fortes e complexas que elas sejam. Um dos exemplos é a abordagem de “força bruta”, onde programas específicos fazem diversas tentativas de submissão de senhas utilizando diversas sequências de caracteres. Por isso, existem ainda outros tipos de autenticação para serem utilizados em conjunto com senhas ou de forma exclusiva. Uma destas formas é a **identificação física**, onde são utilizados cartões e leitoras, bem como dispositivos de biometria para o uso de autenticação por impressão digital, controle por voz (DEITEL; DEITEL; CHOFFNES, 2005).

Porém, mesmo controlando o acesso, atribuindo permissões e gerenciando usuários e senhas, ainda assim é necessário prover segurança em ambientes mesmo após a autenticação. Por isso, os dados, senhas e arquivos armazenados no disco precisam de uma outra camada segura. Em muitos sistemas, em algum nível, é utilizado o conceito de **criptografia**.

A criptografia soluciona diversos problemas de segurança em um ambiente computacional. Ela é basicamente uma forma de codificar bits, alterando sua forma para que quando necessário eles sejam decodificados. Este conceito não é nada novo, desde a antiguidade o ser humano costuma utilizar códigos e chaves para garantir confiabilidade e segurança.

A figura 1 exemplifica um processo padrão de cifragem e criptografia, onde uma mensagem é escrita em texto puro, comum e precisa novamente ser lida, também em texto puro, ou seja, em linguagem comum. Esta mensagem passa por um algoritmo de codificação que criptografa a mensagem em texto puro. Antes da leitura, um novo algoritmo, munido da chave de decodificação, transforma o texto, então criptografado, novamente em um texto puro, em linguagem comum.

Figura 1 - Exemplo de comunicação segura por meio inseguro.



Fonte: Elaborado pelo autor, com base em Silberschatz, 2008.

Quanto aos algoritmos existem diversos disponíveis para criar criptografias. Um deles é o algoritmo de **codificação simétrica**, que utiliza o mesmo algoritmo que codifica, também para decodificar. Existe também o seu oposto, o algoritmo de **codificação assimétrica**, onde os métodos de codificação e de decodificação são diferentes. Há ainda uma nova camada, chamada de codificação com **autenticação**, onde além de receber a mensagem codificada, o emissor deve também provar sua origem, fornecendo uma autenticação junto com a mensagem (SILBERSCHATZ, 2008).

Com relação aos vírus, de forma análoga às ciências naturais, os sistemas operacionais possuem antídotos chamados de **antivírus**. Estes são softwares que servem para a proteção do sistema operacional contra os softwares maliciosos. Os antivírus mantêm listas com vírus conhecidos, bem como os seus códigos. Em geral, cada vírus possui um atributo com uma espécie de assinatura, este é o atributo armazenado nas listas dos antivírus.

Um antivírus costuma trabalhar com o conceito de **verificação de assinatura**, onde ele detecta no sistema operacional a incidência de arquivos que contenham códigos contidos na lista de vírus e assinaturas conhecidas. Um dos grandes problemas do uso das listas nos antivírus é que estas tornam-se grandes em demasiado, visto que a quantidade de vírus se prolifera exponencialmente. Além disso, obviamente, o antivírus é eficaz apenas com os vírus já conhecidos. Sendo assim, os vírus criados recentemente ou ainda não identificados pela assinatura no antivírus não serão detectados.

Com a quantidade de vírus, assinaturas, códigos e comportamentos armazenados no antivírus, há ainda a possibilidade de que ele detecte de forma falsa positiva comportamentos próximos do que sendo um vírus, mesmo este arquivo não estando infectado e nem possuindo funções maliciosas. Uma solução para esta forma de atuação dos softwares de antivírus é utilizar a **verificação heurística**. Esta abordagem, partindo do princípio de que os vírus costumam atuar replicando informações destrutivas pelo sistema, busca no disco arquivos que costumam replicar informações.

Utilizando a verificação heurística, um software de antivírus é capaz de identificar vírus sem a assinatura em sua lista, sendo assim, esta abordagem consegue ser eficaz na proteção contra vírus novos. Porém, assim como na abordagem tradicional, a forma heurística também pode detectar erroneamente, sendo assim, é mais conveniente que os antivírus combinem as duas formas de verificação, a heurística com a de assinatura. Há ainda uma abordagem mais moderna, que é a **verificação em tempo real**, onde a cada execução de processos, estes são verificados para constatar a existência de funcionalidades maliciosas nos arquivos em execução (DEITEL; DEITEL; CHOFFNES, 2005).

Dentro do conceito de abordagens de segurança, existem abordagens preventivas, uma delas é a de **backup**. Com esta técnica todos os dados gerados

pelo sistema e armazenados no disco são compactados e armazenados em uma nova instância. Os mecanismos de backup devem ser periódicos, garantindo que em uma determinada janela de tempo haverá uma cópia fiel dos arquivos daquela data armazenada em um outro local seguro. Em geral, os backups são armazenados em outros computadores. Assim, caso haja algum problema no computador em questão, os dados estão armazenados em outro local.

É muito comum hoje que os dados de backup sejam armazenados não apenas em um local. Um backup pode ser armazenado na própria máquina, em um segundo computador local, ligado via rede, ou ainda em um servidor na nuvem.


O objetivo do backup é garantir que havendo uma determinada necessidade, os dados, tão valiosos para a corporação, estarão seguros para que seja feito o **restore**. Este é o processo de reverter os dados, restaurando os mesmos do backup para que sejam devidamente utilizados no ambiente computacional padrão.

Ainda no âmbito dos ambientes, uma abordagem muito eficaz para prover a segurança em sistemas operacionais é o uso da **redundância**, onde a plataforma provê mais de um dispositivo para a mesma operação. Desta forma, há uma garantia de que caso haja alguma falha, a operação não será impactada, visto que existirá ao menos mais um dispositivo idêntico, pronto para operar.



Você sabia

Uma das principais abordagens de redundância é o uso do RAID (*Redundant Array of Independent Disks*), responsável por clonar discos rígidos em tempo real, fazendo com que os discos sempre estejam redundantes.



Há também as abordagens preventivas de auditoria ligadas aos processos de criação de **logs**. Esta abordagem consiste em criar logs para cada uma das operações realizadas no sistema operacional e são utilizadas para analisar o comportamento destes processos. Todas as operações são armazenadas em log, de forma categorizada, onde falhas, intrusões e demais eventos são armazenados em categorias distintas. Com a auditoria de logs é possível, por exemplo, verificar a incidência de ataques, falhas, entre outros. (SILBERSCHATZ, 2008).

8.2.3 Níveis de Segurança

O Departamento de Defesa dos Estados Unidos criou uma classificação de níveis de segurança em sistemas operacionais. Estes níveis são divididos em quatro grupos, de A à D, onde A é o nível mais alto de segurança e por sua vez o nível D é o menos seguro (DEITEL; DEITEL; CHOFFNES, 2005):

- × **Nível D:** Todos os sistemas que não atendem às características de A à C são classificados como de nível D. Este é o nível dos sistemas operacionais que não oferecem os princípios mínimos de segurança, taxados então como sistemas inseguros.
- × **Nível C:** Um sistema com esta classificação deve dividir os usuários e os dados. Sendo assim, para acessar os dados, deve ser necessário possuir autenticação. As informações dos usuários e grupos devem estar disponíveis apenas aos mesmos.
- × **Nível B:** Os sistemas de nível B, além de possuírem todas as premissas do nível C, devem também possuir permissões centrais padrão, criando classificações para objetos. Estas classificações adicionam etiquetas aos arquivos, informando o seu nível de permissão.
- × **Nível A:** Os sistemas de nível A devem atender a todas as características do nível B e, por sua vez do nível C também, além de exigir que toda a segurança do sistema seja verificada de forma completa.

8.3 Segurança em Ambientes Distribuídos

Além dos problemas com segurança que um sistema operacional deve prever normalmente, em ambientes distribuídos, onde há transações entre diferentes máquinas em rede, há novos problemas que devem ser previstos. Com o uso da internet, os mesmos problemas são multiplicados, surgindo ainda novos, como invasões, download de vírus, DoS, etc. Vamos conhecer agora alguns dos mais comuns tipos de ataques pela rede, bem como as abordagens de segurança para limitá-los.

8.3.1 Tipos de Ataques

Em se tratando de internet, boa parte dos ataques estão voltados à **invasão**. No cinema diversos filmes abordam este tema, onde hackers, os “piratas da internet”, invadem sistemas e roubam senhas, valores e informações. Apesar de parecer um tema de ficção, a invasão de sistemas é muito mais constante do que pode parecer.

Os usuários responsáveis pelas invasões costumam realizar diversas tentativas nos mais variados ambientes e sistemas operacionais procurando violar a segurança do mesmo. Quando esta tentativa é bem sucedida e realizada por um usuário externo e que não possui tal alteração, isso se denomina invasão. Estas invasões procuram roubar, alterar e excluir dados, ou até mesmo derrubar sistemas e serviços.

Existem ainda os **ataques de recusa de serviço** (*Denial of Service - DoS*), que consistem basicamente em realizar um volume extremamente alto de acessos ao sistema, fazendo com que o seu desempenho seja seriamente afetado, de tal forma a deixar com que ele fique indisponível (DEITEL; DEITEL; CHOFFNES, 2005).



Saiba mais

Para conhecer mais sobre boas práticas para garantir a sua segurança na Internet, o Comitê Gestor da Internet (CGI.br) desenvolveu a Cartilha de Segurança para Internet.

Acesse em: <http://cartilha.cert.br/livro>

Disponível no formato PDF e no formato ePub (para leitores de e-Books).



8.3.2 Abordagens de Segurança

A abordagem mais conhecida para oferecer segurança em redes é o uso de **firewalls**, que são servidores colocados como intermediários entre dois pontos que servem para limitar o acesso entre determinadas portas. Ele limita as conexões entre os computadores, determinando quais endereços terão acesso à cada uma das portas de rede disponíveis para os serviços (SILBERSCHATZ, 2008).

Há ainda o uso das **redes virtuais privadas** (VPN's), redes que utilizam a internet para realizar conexões privadas utilizando protocolos mais seguros. As VPN's utilizam túneis seguros para efetuar as comunicações, exigindo autenticação para que a requisição seja autorizada. Elas são alternativas econômicas às redes ponto a ponto de longa distância, pois utilizam a mesma infraestrutura da internet.

Ainda no âmbito das redes, com o crescimento das redes sem fio, é necessário o uso de mecanismos de segurança também neste tipo de rede, visto que as mais variadas transações já trafegam por este meio. Como sendo uma rede padrão, as redes sem fio também são susceptíveis ao uso de firewalls, porém é necessário utilizar ainda outras abordagens.

Para isso existem os protocolos de segurança e criptografia em redes sem fio, como o **WEP** (*Wired Equivalent Privacy*), o protocolo de privacidade equivalente à das redes sem fio, que criptografa as informações para garantir que não haverá acessos à rede sem fio quando não há a devida permissão. Há também o protocolo **WPA** (*Wi-fi Protected Access*), que também faz a criptografia do WEP, porém com maior quantidade de bits, garantindo maior segurança (DEITEL; DEITEL; CHOFFNES, 2005).

Outra abordagem, cada vez mais crescente é o uso de **assinaturas digitais**. Esta abordagem, utiliza a mesma ideia das assinaturas físicas, escritas a mão, onde cada assinatura digital corresponde à autenticação da identidade do usuário. Elas são tecnologias que utilizam criptografias para representar uma codificação que corresponde à assinatura do usuário devidamente cadastrada em uma entidade certificadora competente.

As assinaturas digitais podem ser simples arquivos baixados a partir da entidade certificadora, onde após o seu cadastro na mesma poderão ser utilizados como forma de autenticação. O sistema que necessita da assinatura digital no momento da autenticação, verifica no disco a existência destes arquivos de assinatura digital. Há a opção de assinaturas digitais por meio de *pendrives*, onde o sistema só autenticará se encontrar na porta USB um dispositivo contendo a determinada assinatura. Há ainda a opção de utilizar cartões inteligentes de assinatura digital, onde através de uma leitura com um dispositivo específico o sistema valida a autenticação no cartão (DEITEL; DEITEL; CHOFFNES, 2005).

Uma outra abordagem de segurança ainda é o uso da **esteganografia**, que armazena de forma oculta uma informação dentro de outra, como um metadado. Na internet esta técnica é utilizada para garantir a autenticidade dos arquivos, criando “marcas d’água” online em imagens, músicas, entre outros documentos.

Resumindo

Se qualquer ambiente complexo é sujeito a falhas é papel de todo indivíduo saber evita-las, sejam estas falhas naturais, propositais ou involuntárias, é necessário o constante aprendizado para que sua incidência seja reduzida. O ideal é que falhas nunca ocorram e se ocorrerem, estas devem ser estudadas para que não se repitam.

O conceito de segurança da informação trabalha neste sentido, analisando erros e falhas clássicas para que não ocorram. Por isso neste capítulo você conheceu o papel do CISO, um profissional focado na redução de falhas no ambiente tecnológico. Você conheceu também quais são as falhas e vulnerabilidades que um sistema operacional pode estar susceptível, conhecendo também as formas de evitar que ocorram.

Sendo assim, continue seus estudos, pois a área tecnológica está em constante evolução, fazendo com que os sistemas sejam cada vez mais abrangente, fazendo com que existam cada vez mais vulnerabilidades, usuários maliciosos, vírus e falhas. Por isso, seja um CISO, estude este papel e esteja preparado para os grandes desafios do meio tecnológico.

9

Desempenho

INDEPENDENTE DO OBJETIVO de uso, sempre que pretendemos utilizar um sistema computacional para algo, esperamos que este tenha uma performance aceitável.

Não há nada mais irritante do que utilizar um computador e ele não funcionar. Em ambientes corporativos esta preocupação é ainda maior, pois a falta de agilidade nos processos implica drasticamente na produtividade da corporação.

Sendo assim, com a crescente popularização da informatização de processos é cada vez maior a necessidade de criar abordagens que possibilitem o monitoramento e o provisionamento de ambientes que forneçam melhor performance para os usuários.

Para isso, o presente capítulo apresenta uma abordagem geral sobre os itens que influenciam diretamente no desempenho de sis-

temas operacionais. Aqui serão apresentados os fatores que influenciam no desempenho do ponto de vista do processamento, do uso de memória e nos processos de entrada e saída para o armazenamento.

Objetivo de aprendizagem:

- × Identificar os principais gargalos que influenciam no desempenho de sistemas.
- × Conhecer as ferramentas e itens relacionados com a performance de processamento.
- × Compreender as formas de monitoramento e de uso da memória para identificar como obter mais desempenho.
- × Explicar a importância do monitoramento dos processos de armazenamento em disco como indicadores de performance.

9.1 Desempenho de Processamento

Boa parte das questões relacionadas com o desempenho em um sistema estão ligadas ao processamento da CPU. Sendo assim, para garantir desempenho e escalabilidade em um sistema computacional é necessário adquirir processadores para o ambiente que sejam compatíveis com a necessidade, analisando a quantidade de usuários que utilizarão o ambiente, a quantidade de usuários simultâneos que acessarão e o sistema que este ambiente utilizará (DEITEL; DEITEL; CHOFFNES, 2005).



Saiba mais

A empresa Adrenaline fez um comparativo entre os principais processadores do mercado, criando uma “batalha” entre os equipamentos Intel x AMD, comparando aspectos como temperatura, consumo de energia, processamento de vídeo, etc.

<http://adrenaline.uol.com.br/2015/09/28/37338/comparativo-de-processadores-a-batalha-amd-vs-intel-no-cpu-chart-adrenaline->



Para monitorar o desempenho do processamento é necessário analisar alguns atributos relacionados com a execução dos processamentos no sistema operacional (NEMETH; HEIN; SNYDER, 2004). O primeiro deles é a **utilização global de CPU**, que expressa o valor de operação total do processador. Este atributo apresenta o uso global da velocidade do CPU.

O segundo é a **média de cargas**, que apresentam os valores em média de processos que estão em execução no momento, ou de processos que estão em espera para serem executados. O terceiro atributo é o **consumo de CPU por processos**, onde são listados os processos em execução atual no sistema. Desta forma é possível identificar quais são os processos que estão consumindo mais recursos.

Existe ainda o **tempo de usuário**, que é a porcentagem de tempo empregada pelo processador para executar aplicações invocadas pelo usuário. Da mesma forma, existe o **tempo de sistema**, que mede a porcentagem de uso do processamento de processos invocados pelo próprio núcleo do sistema.


Outro atributo é referente às **alterações de contexto por intervalo**, que apresenta a quantidade de vezes que o núcleo do sistema operacional alterou o processo em execução no momento. Há também as **interrupções por intervalo**, que ilustra a quantidade de interrupções que houveram no processo, geradas pelo núcleo do sistema operacional ou por dispositivos de hardware. Por fim, há ainda o **tempo de inatividade**, relacionado com a quantidade de tempo de inatividade dos processos e o **tempo de espera**, que exhibe a quantidade de tempo de espera para os processos de E/S do sistema operacional.



Importante

Existe uma diferença importante entre o Desempenho e a Escalabilidade. O desempenho está ligado à capacidade do sistema de operar de modo performático. A escalabilidade diz respeito à capacidade do sistema de operar com diversos usuários simultâneos.

Um item costuma implicar no outro, sendo assim, a configuração de um sistema visando o desempenho pode não influenciar positivamente em sua escalabilidade, sendo assim o desafio da arquitetura de sistemas é garantir ambientes escaláveis e performáticos.



Para verificar estes atributos, a figura 1 apresenta o uso da ferramenta vmstat, um comando do Linux, que fornece em tempo real de forma simplificada o uso de processamento de CPU do sistema. A primeira linha do comando apresenta os valores de processamento médios entre a inicialização do sistema operacional e o momento atual. As linhas a seguir são os valores atualizados geralmente a cada cinco segundos, após a linha anterior.

Figura 1 - Monitoramento de processos do servidor.

```
stat 5 5
s----- memory----- --swap-- ----io--  --system--  ----cpu----
swpd    free buff      cache    si so      bi bo      in  cs    us sy id wa
820    2606356 428776 487092    0 0      4741 65    1063 4857 25 1 73 0
820    2570324 428812 510196    0 0      4613 11    1054 4732 25 1 74 0
820    2539028 428852 535636    0 0      5099 13    1057 5219 90 1 9 0
820    2472340 428920 581588    0 0      4536 10    1056 4686 87 3 10 0
820    2440276 428960 605728    0 0      4818 21    1060 4943 20 3 77 0
```

Fonte: NEMETH; HEIN; SNYDER, 2004.

Nas colunas “system”, no campo “in” fica o valor referente às interrupções por intervalo e no campo “cs” apresenta as alterações de contexto por intervalo. Nas colunas “cpu“, no campo “us” ficam os valores de tempo de usuário, no campo “sy” os valores de tempo de sistema, em “id” ficam os valores de tempo de inatividade e em “wa” os valores de tempo de espera para processos de E/S.

Com estes itens, é possível identificar os principais gargalos de desempenho do sistema operacional quando relacionados com os processos de processamento. A ferramenta vmstat apresenta de forma simplificada em tempo real o uso atual da CPU, servindo como forma de monitoramento e diagnóstico da performance de processamento do sistema operacional.

Existe ainda o comando “top” no Linux (figura 2), que monitora o desempenho de processamento sob outros aspectos. A primeira linha do comando informa o tempo atual de uptime, o número de usuários conectados ao sistema operacional no momento e a média de carga do processamento em 5, 10 e 15 minutos, em sequência (NEMETH; HEIN; SNYDER, 2004).

Figura 2 - Comando top no Linux.

```

top - 16:29:46 up 1:55, 1 user, load average: 0,00, 0,06, 0,06
Task: 89 total, 1 running, 88 sleeping, 0 stopped, 0 zombie
%Cpu(s): 0,3 us, 0,0 sy, 0,0 ni, 99,7 id, 0,0 wa, 0,0 hi, 0,0 si, 0,0 st
Kib Mem: 8218132 total, 3408268 used, 4809864 free, 23544 buffers
Kib Swap: 16584700 total, 0 used, 16584700 free, 472924 cached

```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
2224	tomcat7	20	0	2980m	2,3G	11m	S	1,3	29,8	6:08.86	java
1	root	20	0	2284	748	644	S	0,0	0,0	0:01.10	init
2	root	20	0	0	0	0	S	0,0	0,0	0:00.00	kthreadd
3	root	20	0	0	0	0	S	0,0	0,0	0:00.04	ksoftirqd/0
6	root	rt	0	0	0	0	S	0,0	0,0	0:00.00	migration/0
7	root	rt	0	0	0	0	S	0,0	0,0	0:00.01	watchdog/0
8	root	rt	0	0	0	0	S	0,0	0,0	0:00.00	migration/1
9	root	20	0	0	0	0	S	0,0	0,0	0:00.00	kworker/1:0
10	root	20	0	0	0	0	S	0,0	0,0	0:00.02	ksoftirqd/1
11	root	20	0	0	0	0	S	0,0	0,0	0:00.67	kworker/0:1
12	root	rt	0	0	0	0	S	0,0	0,0	0:00.01	watchdog/1
13	root	rt	0	0	0	0	S	0,0	0,0	0:00.00	migration/2
14	root	20	0	0	0	0	S	0,0	0,0	0:00.00	kworker/2:0
15	root	20	0	0	0	0	S	0,0	0,0	0:00.00	ksoftirqd/2
16	root	rt	0	0	0	0	S	0,0	0,0	0:00.01	watchdog/2
18	root	20	0	0	0	0	S	0,0	0,0	0:00.00	kworker/3:0

Fonte: Elaborado pelo autor, 2016.

A segunda linha apresenta um resumo dos processos atuais. Exibindo o total geral de processos, a quantidade de processos em execução, a quantidade de processos em espera, a quantidade de processos parados e a quantidade de processos em modo zumbi. A terceira linha representa os estados do processador, contendo a porcentagem de uso para processos de usuário (us), a porcentagem de processos do kernel (sy), o tempo de inatividade (id), o tempo destinado para processos de entrada e saída (wa), o tempo dedicado às interrupções de hardware (hi) e o tempo dedicado às interrupções de software (si) (NEMETH; HEIN; SNYDER, 2004).

Na quarta e na quinta linhas são exibidos os valores de uso de memória e na linha “mem” aparece o uso da memória física e na linha “swap” aparecem os valores de uso de memória virtual. Por fim aparecem as linhas com os processos, na primeira coluna aparece o “PID” com o número de identificação do processo, na segunda coluna (user) aparece o usuário responsável por aquele processo, a terceira e quarta colunas (PR e NI) apresentam a prioridade de agendamento do processo, a coluna “VIRT” exibe a quantidade de uso da memória virtual no processo em questão.

Na coluna “RES” é exibido o tamanho de memória física utilizada pelo processo, na coluna “SHR” é exibido o total de memória compartilhada utilizada pelo processo, a coluna “S” apresenta o estado do processo, podendo ser ele ininterrupto (D), executando (R), inativo (S), parado (T) e zumbi (Z). Ao final, na coluna “%CPU” aparece a porcentagem de tempo de CPU destinada ao uso do processo em questão e na coluna “%MEM” é exibida a porcentagem de memória física destinada ao processo em questão (NEMETH; HEIN; SNYDER, 2004).

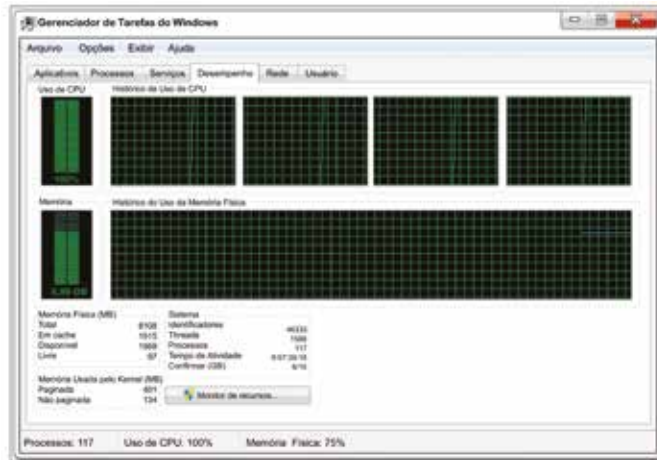
A ferramenta top é um pouco mais completa do que a ferramenta vmstat. Com este comando é possível verificar não apenas os parâmetros de uso atual de processamento, mas também permite verificar parâmetros a cada processo.

Visando garantir maior desempenho, sobretudo maior escalabilidade em ambientes computacionais, uma das alternativas é o uso do multiprocessamento. Esta técnica consiste em utilizar diversos processadores em um mesmo ambiente operacional, fazendo com que toda a capacidade de processamento seja balanceada entre as diversas unidades do ambiente.

Existem basicamente três formas principais para se obter o multiprocessamento. Uma delas é o uso de vários processadores em um mesmo computador, onde a arquitetura da máquina permite a troca e a alocação de mais de um processador por vez. A segunda alternativa é quando os processadores já possuem mais de um núcleo, permitindo o multiprocessamento com apenas um processador. Há ainda a terceira alternativa, onde é construído um cluster de vários computadores, ou seja, mais de um computador processa as mesmas informações visando balancear o processamento entre as diversas máquinas (NEMETH; HEIN; SNYDER, 2004).

O Windows também possui a sua própria ferramenta para monitoramento de processos (Figura 3). Ele apresenta os processos que estão consumindo mais CPU e a quantidade de memória utilizada, a porcentagem de processamento total do CPU, sendo que na aba processos, de forma semelhante ao comando top do Linux, o usuário pode verificar o desempenho de cada processo do sistema (NEMETH; HEIN; SNYDER, 2004).

Figura 3 - Gerenciador de Tarefas do Windows.



Fonte: Elaborado pelo autor, 2016.

CPU: Unidade Central de Processamento (*Central Processing Unit*), parte do ambiente computacional responsável pelo processamento aritmético e o controle de entrada e saída de dados, representado também como “processador”.

Uptime: Tempo de atividade desde o início da execução. Por exemplo, desde o início da execução de um processo, desde o início da execução do sistema operacional, desde que a máquina foi ligada.

Cluster: Grupo de computadores ligados em sequência visando o uso para o mesmo fim.

Desta forma, um dos principais gargalos de desempenho em um sistema operacional torna-se o processamento. Sendo assim, é importante conhecer as ferramentas de monitoramento de processos do sistema operacional em questão visando identificar quais processos estão consumindo desempenho da CPU.

9.2 Desempenho de Memória

Para garantir um bom desempenho de um sistema operacional, boa parte do sucesso desta operação deve-se ao desempenho de memória principal. Em computadores pessoais, por exemplo, para boa parte das operações depende do uso de memória, sendo assim, a velocidade do computador não está ligada diretamente à velocidade do processador, mas sim à capacidade de memória.

Em estruturas computacionais do meio corporativo, como em computadores que operam como sendo servidores, o desempenho de um sistema deve-se em grande parte também à capacidade de memória. De nada adianta o servidor possuir equipamentos de boa procedência, ou diversos processadores com alta velocidade, se não houver boa capacidade de memória.

Isso ocorre pois como é na memória que as informações são trafegadas entre o processador e o disco. Uma máquina com boa capacidade de memória acaba auxiliando em muito no desempenho de um sistema, pois como na memória são armazenados os dados para futuros acessos, quanto mais dados a memória é capaz de armazenar temporariamente, mais rápido é o sistema, visto que mais dados foram armazenados temporariamente para futuros acessos (NEMETH; HEIN; SNYDER, 2004).

Ao implantar um ambiente computacional para uma determinada operação, é necessário que, ao criar a arquitetura para o ambiente, seja verificado se o mesmo possui a quantidade de memória suficiente. Dados os atuais preços para os pentes de memória, cada vez mais baixos, é comum investir em ter a capacidade máxima de memória do ambiente.

Para medir o desempenho de memória no sistema operacional, os atributos que são monitorados são o total de memória virtual ativa, a quantidade de swap e a quantidade de paginação. Para garantir o desempenho de memória, o objetivo é reduzir o uso de memória até que os valores de paginação sejam reduzidos (NEMETH; HEIN; SNYDER, 2004).

No Windows, é possível verificar também o uso de memória no Gerenciador de Tarefas (figura 3), enquanto no Linux, por exemplo, utilizando o comando *free* (figura 4), é possível verificar a quantidade uso de memória, na primeira linha (memória total, utilizada e livre), na terceira linha, o uso de swap, e na última linha, a memória virtual total:

Figura 4 - Comando *free* para verificar a memória.

```

$ free -t
              total        used        free     shared    buffers     cached
Mem:          127884        96888        30996        46840         57860         10352
-/+ buffers/cache:        28676         99208
Swap:          265032         3576        261456
Total:          392916       100464       292452

```

Fonte: Nemeth; Hein; Snyder, 2004.

Este comando *free* tem como principal objetivo exibir de forma simples o uso atual de memória física e virtual, tornando a operação mais simples. Sendo assim, em casos extremos e em emergências, a ferramenta apresenta informações simples e rápidas de interpretação, podendo a partir disso o administrador de sistemas tomar suas devidas providências.

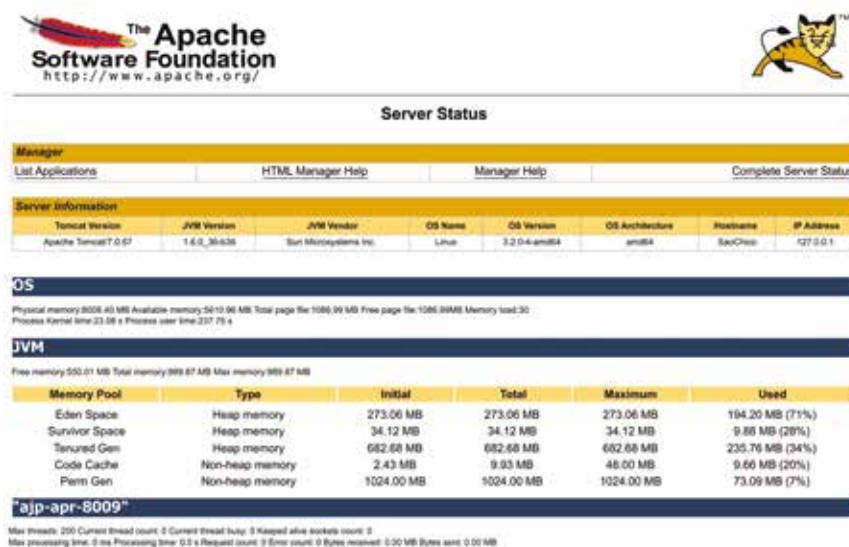
No comando *vmstat* (figura 1), apresentado no item anterior (5.1), também é possível monitorar os processos de memória no sistema operacional. Dentro da coluna “*memory*”, o setor “*swpd*” apresenta a quantidade de memória virtual em uso no momento, o setor “*free*” apresenta a quantidade de memória livre e “*buff*” apresenta a quantidade de memória usada em buffers, “*cache*” indica a quantidade de memória utilizada como cache. Na coluna “*swap*”, o setor “*si*” indica a quantidade de memória vinda do swap em disco (*swapped in*) e o setor “*so*” indica a quantidade de memória armazenada no swap (*swapped out*) (NEMETH; HEIN; SNYDER, 2004).

Esta ferramenta é um pouco mais completa, pois apresenta divisões diferentes para os atributos de memória, fornecendo um monitoramento temporal para o diagnóstico em tempo real dos gargalos de memória no sistema operacional.

Swap: processo de troca de dados entre a memória virtual e a memória física, criada pelo sistema operacional para armazenamento no disco.

Serviços específicos instalados sobre o sistema operacional também possuem suas próprias ferramentas para o uso de memória. O servidor de aplicativos web Java, Apache Tomcat (APACHE.ORG, 2016), possui a sua própria ferramenta de monitoramento (figura 5). Nesta ferramenta as instâncias de memória da máquina virtual Java são monitoradas exibindo a quantidade de memória inicial, total, máxima e a porcentagem de uso.

Figura 5 - Ferramenta de Monitoramento do Apache Tomcat



Fonte: Elaborado pelo autor, 2016.

O “Eden Space” é relacionado com o espaço de memória alocado para os objetos do programa, “Survivor Space” é destinado ao armazenamento de processos que sobreviveram à “coleta de lixo” de objetos Java. O “Tenured Gen” armazena os objetos que já ficaram muito tempo no “Survivor Space”, a “Code Cache” é destinada ao armazenamento de códigos fonte para a compilação e armazenamento rápido dos programas (NEMETH; HEIN; SNYDER, 2004).

Como foi visto, é possível verificar que existem inúmeras formas de mensurar a capacidade de memória de um sistema operacional. Dada a importância dos dispositivos de memória para a garantia da performance e

escalabilidade de um ambiente computacional, o correto provisionamento de memória, bem como o constante monitoramento do seu desempenho, é um fator decisivo no sucesso de uma informatização.

9.3 Desempenho de Armazenamento

Outro ponto importante para o bom desempenho de um ambiente operacional são as variáveis ligadas ao desempenho de armazenamento. É necessário que o ambiente possua capacidades de armazenamento voltadas ao desempenho. Não basta possuir alta velocidade de processamento, alta capacidade de memória, se não houver velocidade nos processos de entrada e saída de discos de armazenamento.

O armazenamento em disco pode ter o seu desempenho mensurado por diversos atributos. Um deles é o RPM, um atributo de fábrica do disco, que indica a quantidade de rotações por minuto que o disco é capaz de realizar. Este atributo influencia diretamente na velocidade com que o disco opera para buscar e acessar os dados dentro da estrutura do disco. Sendo assim, quanto maior o RPM, maior a velocidade.

Outro item que influencia na velocidade do disco é capacidade de memória buffer para a transferência e a taxa de transferência de dados. A memória buffer é um espaço de memória utilizado para acessar de forma mais rápidas os dados, fazendo com que o cálculo de armazenamento seja antecipado mesmo antes do uso do processador.

Boa parte do desempenho de armazenamento deve-se as taxas de entrada e saída de disco. Estas operações são medidas em “transferências de E/S”, “blocos lidos por segundo”, “blocos escritos por segundo”, “total de blocos lidos” e “total de blocos escritos”.

A figura 6 apresenta o comando Linux “iostat” para medir as capacidades de E/S para cada disco. Cada linha representa um disco, onde a coluna “tps” representa a taxa de transferências de E/S por segundo, a coluna “Blk_read/s” representa os blocos lidos por segundo, a coluna “Blk_wrtn/s” representa os blocos escritos por segundo, a coluna “Blk_read” representa o total de blocos lidos e por fim, a coluna “Blk_wrtn” representa o total de blocos escritos (NEMETH; HEIN; SNYDER, 2004).

Figura 6 - Comando iostat para medir a E/S com o disco.

```
...
Device:      tps      Blk_read/s    Blk_wrtn/s    Blk_read    Blk_wrtn
hdisk0       0.54         0.59         2.39       304483     1228123
hdisk1       0.34         0.27         0.42       140912     216218
hdisk2       0.01         0.02         0.05        5794      15320
hdisk3       0.01         0.00         0.00         0         0
```

Fonte: NEMETH; HEIN; SNYDER, 2004.

Você sabia

Em geral, dada a sua arquitetura e as distintas formas de instalação destinadas aos mais diversos dispositivos, os sistemas operacionais considerados com o melhor desempenho são os baseados na família Unix.

Para mensurar o desempenho de discos dentro do sistema operacional, utilizando sistemas Linux é possível mensurar o desempenho dos discos utilizando a ferramenta “hdparm” (figura 7).

Figura 7 - Comando hdparm para medir o desempenho de um disco.

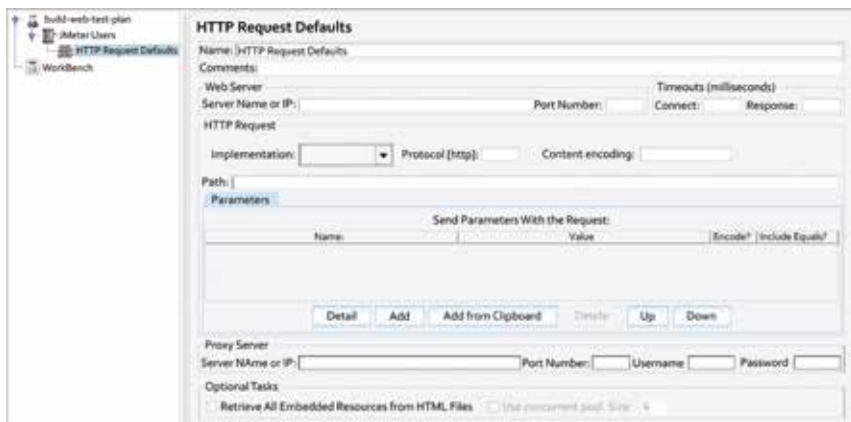
```
/dev/sda:
Timing cached reads:   2438 MB in  2.00 seconds = 1218.93 MB/sec
Timing buffered disk reads: 304 MB in  3.00 seconds = 101.17 MB/sec
```

Fonte: Elaborado pelo autor, 2016.

Com esta ferramenta, para cada disco é possível constatar a velocidade de trâmite de dados no disco. Para isso, o comando retorna os dados de velocidade divididos entre o tempo de leitura de dados em cache e em buffer. A primeira coluna exibe o volume de dados utilizados para o cálculo do teste, seguido do espaço de tempo, ao final aparece o resultado da velocidade utilizando como unidade de medida os “megabytes por segundo”. Quando maior este número, mais rápido é este disco.

De nada adianta implantar um sistema rápido capaz de trafegar dados com extrema velocidade entre os discos sem que se tenha certeza disso. Por isso, antes de enviar para a produção um novo servidor ou um novo serviço, é papel do responsável testar esta sua eficiência. Para isso existem ferramentas capazes de realizar “Testes de Stress”, que são testes que simulam diversos acessos simultâneos, pré-configurados. Assim, durante o teste, é possível monitorar todos os parâmetros e verificar quantos usuários o ambiente suporta sem perda de desempenho. A ferramenta livre JMeter, da Apache, é capaz de realizar estas operações em sistemas web, onde de qualquer computador é possível disparar testes em aplicações instaladas no servidor que se deseja “estressar”.

Figura 8 - Ferramenta JMeter para testes de Stress



Fonte: Elaborado pelo autor, 2016.

Quadro 1 - Comparativo entre os comandos.

Ferramenta	Sistema Operacional	Tipo de Verificação	Objetivo
vmstat	Linux	Memória, Processador, E/S	Analisa de forma consolidada o desempenho do sistema operacional
top	Linux	Memória, Processador, E/S	Analisa de forma analítica o desempenho do sistema operacional

Ferramenta	Sistema Operacional	Tipo de Verificação	Objetivo
Gerenciador de tarefas	Windows	Memória, Processador, E/S	Analisa de forma analítica e consolidada o desempenho do sistema operacional
JMeter	Windows e Linux	Teste de Stress	Simular diversos acessos simultâneos visando medir o desempenho de um serviço.
free	Linux	Memória	Verificar o uso de memória
iostat	Linux	E/S de Disco	Verificar o desempenho de E/S no disco
hdparm	Linux	Disco	Verificar o desempenho de armazenamento

Fonte: Elaborado pelo autor, 2016.

O quadro 1 apresenta um comparativo entre as diferentes ferramentas apresentadas no capítulo, bem como seus objetivos de uso.

Mais do que unicamente armazenar, o administrador de sistemas deve ter ciência de que os dispositivos de armazenamento devem possuir capacidades de fornecer o melhor desempenho possível, tanto nas operações de entrada e saída, quando nos procedimentos internos de armazenamento. Sendo assim, é necessário, antes de iniciar a implantação de um ambiente operacional, uma intensa pesquisa de mercado para adquirir discos com capacidades de desempenho compatíveis com o uso que será dado a este sistema. Garantindo assim a sua performance e escalabilidade.

Resumindo

Da forma com que as organizações estão cada vez mais informatizadas, é imprescindível estudar e compreender sobre os fatores relacionados com o desempenho de sistemas operacionais.

Por isso, o constante estudo na área de desempenho torna possível o desenvolvimento de arquiteturas de ambientes computacionais capazes de ope-

rar com performance e escalabilidade, permitindo que estes ambientes sejam cada vez mais rápidos e suportem cada vez mais usuários simultaneamente.

Neste capítulo você conheceu os principais itens que influenciam na performance de um sistema e identificou quais são os conceitos relacionados com o desempenho, performance e escalabilidade de um sistema. Você conheceu também as principais formas, conceitos e ferramentas envolvidas no monitoramento e no diagnóstico de problemas de desempenho nos atributos relacionados com o processamento, memória e armazenamento.

Continue aprimorando seus conhecimentos na área de desempenho, pois mais do que simples computadores, as organizações necessitam de um setor de tecnologia da informação organizado e ativo, capaz de garantir a maior eficiência possível de seu parque tecnológico. Sendo assim, a sua constante atualização em técnicas de performance, ferramentas de monitoramento e diagnóstico de desempenho e testes de escalabilidade são imprescindíveis para este setor da cadeia produtiva.

10

Sistemas Operacionais do Mercado, Aplicabilidade e Aplicativos de Gestão

CARO ALUNO,

Os sistemas operacionais possuem objetivos e funcionalidades disponíveis para as mais variadas ocasiões de segmentos do mercado. Sendo assim, cada sistema possui suas próprias peculiaridades, fazendo com que a escolha do sistema operacional para cada aplicabilidade seja essencial. Para o uso em ambiente executivo, por exemplo, há sistemas operacionais mais indicados, já para o meio corporativo, de servidores, há um perfil de sistema operacional determinado.

Além da grande gama e das variações que existem entre os próprios sistemas operacionais, existem ainda abordagens avançadas que fazem com que a manutenção e a administração de ambientes corporativos sejam realizadas de forma mais efetiva e profissional, como o acesso remoto e a virtualização.

Para isso, este capítulo apresentará os principais sistemas operacionais disponíveis no mercado de tecnologia da informação, suas aplicabilidades e apresentará também ferramentas para controle e redução de infraestrutura utilizada em sistemas operacionais.

Aproveite este capítulo para aprofundar os seus estudos, criando assim mais intimidade com este tema tão importante para o segmento de tecnologia da informação, pois cada ferramenta de um sistema operacional tem seus objetivos, e conhecer cada uma delas garante maior base para implantações futuras em ambiente computacionais.

Objetivo de aprendizagem:

- × Identificar os principais sistemas operacionais do mercado e suas aplicabilidades;
- × Conhecer os objetivos de cada sistema operacional para as diversas operações que cada um dispõe;
- × Compreender os principais protocolos e ferramentas para realizar acesso remoto à outros sistemas operacionais;
- × Conhecer os conceitos e ferramentas utilizadas para criar arquiteturas de virtualização.

10.1 Sistemas Operacionais do Mercado

Dependendo dos objetivos do uso, cada usuário, seja ele pessoal ou corporativo, possui algumas opções de sistemas operacionais disponíveis no mercado para adotar. Para garantir a sua escolha, é necessário observar sua experiência com informática, orçamento, equipamentos e objetivos. Cada sistema operacional possui objetivos bem definidos, podendo se dizer que para cada objetivo, cada usuário possui o seu “sistema operacional ideal”.

Existem sistemas operacionais voltados para o meio corporativo. Para uso como servidores, existem ainda sistemas operacionais destinados à dispositivos específicos, como relógios inteligentes e telefones, para pequenos dispositivos e para grandes computadores. Os sistemas operacionais podem

ser capazes de rodar a partir do disco ou em dispositivos USB como *pendrives* ou até mesmo pela rede.

Você sabia

Existe um sistema operacional para desktop que era capaz de rodar a partir de um simples disquete. Trata-se do MenuetOS, um sistema operacional levíssimo, que possui diversas funcionalidades de um sistema operacional de mercado, como servidor http, suporte a USB, placas de vídeo, TV digital, streaming e games.

Fonte: <http://www.menuetos.net/>

Nesta seção você irá conhecer os principais sistemas operacionais utilizados no mercado bem como as suas principais ferramentas e aplicabilidades.

10.1.1 Windows

O Windows, sistema operacional da Microsoft, é o sistema operacional mais utilizado e difundido do mercado, sendo o padrão para a esmagadora maioria dos usuários (NETMARKETSHARE, 2016). Ele é o principal produto do conhecido Bill Gates, fundador da companhia e criador do sistema operacional, lançado em 1985.

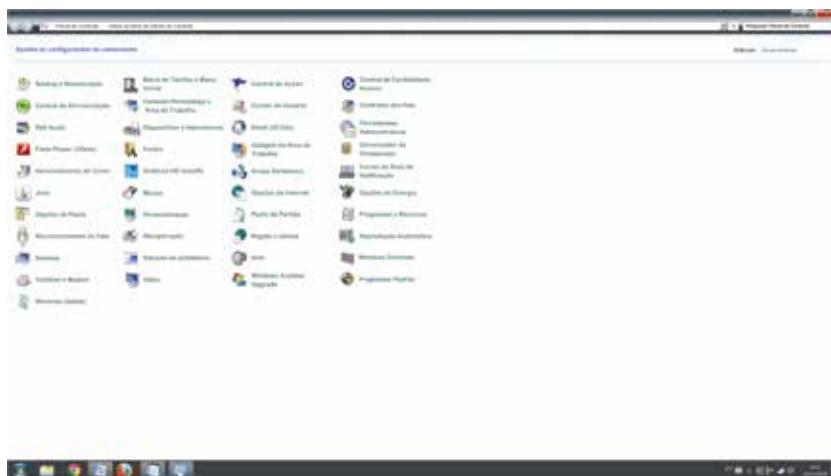
É um sistema operacional relativamente simples de aprendizado e é muito indicado para usuários básicos, que necessitam utilizar apenas ferramentas simples e que não necessitam realizar operações avançadas. Ele já é integrado com diversos aplicativos clássicos de escritório e sua instalação e manutenção torna-se simples, visto que é um sistema operacional muito difundido e utilizado por todo o mercado de tecnologia da informação.

O Windows ainda é o sistema operacional mais difundido no mundo dos games, visto que boa parte dos grandes jogos de computador disponíveis no mercado estejam disponíveis apenas para a plataforma Windows. Graças a este foco, fabricantes de placas de vídeo investem em recursos e em compatibilidades com este sistema operacional, tornando o Windows o padrão para games no computador (OGLETREE, 2002).

Porém, embora seja muito utilizado por usuários domésticos e em escritórios, o Windows também é amplamente utilizado em ambientes corporativos com em servidores. Este sistema operacional também possui versões específicas para servidores, como o Windows NT Server e o Windows Server 2012.

Estas versões para servidor funcionam como base para rodar ambientes voltados à execução de serviços baseados em Windows, como Microsoft SQL Server, o servidor de banco de dados da Microsoft, o IIS (Internet Information Services), servidor de páginas web da Microsoft e as aplicações baseadas em .Net, a plataforma de desenvolvimento de software da Microsoft. Existem ainda diversas outras aplicabilidades corporativas, pois dados os anos de intensa popularidade entre seus usuários, é muito comum que outros fabricantes invistam na compatibilidade apenas com ambientes baseados em Windows.

Figura 1 - Captura de tela do sistema operacional Windows.



Fonte: Elaborado pelo autor, 2016.

Exemplos de investimentos em compatibilidade entre fabricantes é o uso do Windows como único sistema operacional capaz de se conectar a impressoras de corte específicas, equipamentos de diagnóstico eletrônico de exames, sensores específicos, entre outros. Desta forma, embora a grande popularidade do Windows diminua com o decorrer dos anos, ele ainda é

muito utilizado, fazendo com que ele seja sempre compatível com diversos fabricantes de dispositivos (OGLETREE, 2002).

10.1.2 Linux

O Linux, na verdade, não é um sistema operacional, ele trata-se na verdade de uma versão do UNIX, um sistema de alto desempenho desenvolvido nos anos 60 e distribuído de forma livre. Com esta premissa, nos anos 90, Linus Torvalds, com base no UNIX, desenvolveu o Linux como sendo uma implementação UNIX, também livre para distribuição.

Desta forma, não há um sistema operacional denominado Linux, existem várias distribuições que implementam o kernel do Linux, como o Linux Ubuntu, o Linux Fedora, SuSE, Slakware, Mint, RedHat, Debian, entre muitos outros. Sendo assim, cada distribuição Linux também possui suas devidas características e objetivos.

De forma geral, o sistema operacional Linux não é totalmente voltado ao uso doméstico e em escritórios, embora seja capaz de atender satisfatoriamente também esta necessidade. As distribuições Linux costumam ser voltadas a ambientes computacionais focados em servidores, pois possuem grande flexibilidade e desempenho.

Como o Linux é um sistema operacional livre, em geral, não há custos para instalar uma distribuição, tornando o orçamento de todo um parque tecnológico um tanto quanto mais em conta, visto que para cada máquina não é necessário em investir nas licenças do sistema operacional. Apesar de boa parte das distribuições Linux serem gratuitas, os sistemas baseados em Linux são muito versáteis. Existem diversos pacotes de serviços pré-instalados e configurados que com o conhecimento específico das ferramentas e comandos Linux tornam uma operação em servidores algo robusto e funcional (NEMETH; HEIN; SNYDER, 2004).

Mesmo sendo os mais conhecidos, os sistemas baseados em Linux não são os únicos sistemas operacionais da “Família UNIX” disponíveis no mercado. Existem ainda os sistemas baseados em BSD, que também foram feitos com base em UNIX e são amplamente utilizados em ambientes de servidores, sendo sempre

destinados à aplicações de alto desempenho. Exemplos de sistemas operacionais baseados em BSD são o FreeBSD, OpenBSD, Solaris, OpenBSD e o NetBSD.

Figura 2 - Captura de tela de uma distribuição Linux Mint.



Fonte: Elaborado pelo autor, 2016.

10.1.3 Mac OSX

Antigamente conhecido como Macintosh, OSX é o sistema operacional elaborado por Steve Jobs, em 1984, para os computadores da empresa Apple. Diferentemente dos outros sistemas operacionais do mercado, os sistemas da Apple, desde o Macintosh até o OSX, são voltados para operar apenas em computadores de arquitetura específica, ou seja, voltados apenas à especificações da própria empresa.

O OSX é um sistema operacional de fácil operação, que atende as necessidades de usuários domésticos e de escritórios comuns, porém seu grande foco está no mercado de criação artística. Boa parte dos softwares de edição de mídia começaram sua operação justamente em ambiente Macintosh, sendo assim tornou-se referência no segmento.

Os computadores Apple com OSX são amplamente utilizados em ambientes de criação gráfica, edição de vídeo e de produção musical. Existem

diversos softwares disponíveis apenas para o sistema operacional OSX justamente com este fim, fazendo dos computadores Apple o foco de operação neste ramo da cadeia produtiva.

As primeiras versões do Macintosh eram baseadas no kernel do sistema BSD, sendo que até hoje, com o OSX, o sistema é baseado na família UNIX. É um sistema estável e de simples operação, permitindo diversas operações domésticas e corporativas, suportando também servidores e diversas funcionalidades ligadas às diferentes mídias (PROGUE, 2010).

Figura 3 - Captura do OSX utilizando software para produção musical.



Fonte: Elaborado pelo autor, 2016.

Importante

O OSX é amplamente utilizado no meio musical, sendo praticamente um padrão nesta indústria. Existem diversos equipamentos de gravação compatíveis somente com OSX.

10.1.4 Sistemas Operacionais para Dispositivos Móveis

No mercado de dispositivos móveis como de celulares inteligentes, *tablets*, *netbooks* e relógios inteligentes existem também seus sistemas operacionais específicos para estas operações. Os mais conhecidos são os sistemas operacionais Android, iOS e Windows Phone. Ele é o sistema operacional móvel mais utilizado no mercado, compatível ainda para integração com diversos outros dispositivos e ferramentas.

O sistema operacional Android, hoje pertencente ao Google, é baseado em Linux, sendo assim, pode ser considerado uma distribuição Linux específica para dispositivos móveis. É utilizado em 57% dos dispositivos móveis, estando disponível para telefones celulares, *tablets*, relógios inteligentes, *netbooks*, entre outros (NETMARKETSHARE, 2016).

É um sistema altamente flexível, com uma gama muito grande de aplicativos disponíveis para download em seus dispositivos desenvolvidos em um ambiente baseado em Java, fazendo com que o seu desenvolvimento de aplicativos seja simplificado.

O iOS é o sistema operacional da Apple destinado aos dispositivos móveis da companhia. O mesmo roda sobre os celulares iPhone, os *tablets* iPad, players de mídia iPod e relógios inteligentes Apple Watch. É amplamente compatível com o OSX, permitindo a interoperabilidade entre dispositivos Apple extremamente simples e funcional.

Possui também uma loja de aplicativos com diversos programas para download no aparelho, desenvolvidos sobre uma plataforma própria da Apple destinada apenas à aplicativos do seu sistema operacional iOS. Ele opera em 35% dos dispositivos móveis do mercado e foi o precursor do conceito de Multi-touch, suportando diversos “toques” simultâneos para realizar operações no aparelho.

Por fim, o Windows Phone é o sistema operacional da Microsoft para dispositivos móveis. Ele opera em aparelhos celulares e *tablets* compatíveis, suportando também o download de aplicativos específicos para a plataforma, desenvolvidos utilizando tecnologias padrão Microsoft.

10.2 Acesso Remoto

Com o uso distribuído dos recursos computacionais com base na internet, o acesso a computadores e servidores à distancia é uma necessidade comum nos ambientes corporativos da atualidade. Sendo assim, os sistemas operacionais implementam protocolos e ferramentas de acesso remoto à seus terminais.

Uma das ferramentas mais simples para realizar o acesso remoto entre os mais diversos sistemas operacionais é o *TeamViewer*, capaz de permitir o controle e o compartilhamento de telas em tempo real bastando os dois terminais possuírem a ferramenta instalada.

Existe também o protocolo SSH (*Secure Shell*). Ele é um protocolo de acesso remoto via prompt de comando para controle via rede de ambientes computacionais baseados em UNIX. Embora ele sirva para acessar máquinas UNIX, existem diversos clientes que implementam o protocolo SSH que permitem o seu acesso, sendo assim, qualquer sistema operacional com um cliente SSH pode fazer acesso à uma máquina via SSH.

Para que seja feito o seu uso basta que o computador de destino implemente este protocolo e que o computador que deseja acessá-lo tenha o cliente de SSH. A figura 4 apresenta uma máquina Windows com o cliente Putty, acessando via SSH um servidor Linux via prompt de comando.

Figura 4 - Máquina Windows com acesso remoto a um Linux via SSH.

```

top - 09:52:31 up 1 day 1:38, 1 user, load average: 0,00, 0,01, 0,05
task: 98 total, 1 running, 97 sleeping, 0 stopped, 0 zombie
%cpu(s):  0,0 us,  0,1 sy,  0,0 ni, 99,9 id,  0,0 wa,  0,0 hi,  0,0 si,  0,0 st
Mem: 8218112 total, 2986200 used, 5389932 free, 67216 buffers
Mem Swap: 16584768 total, 0 used, 16584768 free, 804232 cached

  PID USER      PR  NI  VIRT  RES  SHR  S  %CPU  %MEM     time+  COMMAND
11116 root        20   0 45128 1356 1016  R   0,1   0,0   0:00.01 top
1 root        20   0 2284   744 644   S   0,0   0,0   0:01.51 init
2 root        20   0 0       0 0     S   0,0   0,0   0:00.00 kthreadd
3 root        20   0 0       0 0     S   0,0   0,0   0:00.19 ksftirqd/0
6 root        20   0 0       0 0     S   0,0   0,0   0:00.03 migration/0
7 root        20   0 0       0 0     S   0,0   0,0   0:00.20 watchdog/0
8 root        20   0 0       0 0     S   0,0   0,0   0:00.00 migration/1
9 root        20   0 0       0 0     S   0,0   0,0   0:00.00 kworker/1:0
10 root       20   0 0       0 0     S   0,0   0,0   0:00.09 ksftirqd/1
11 root       20   0 0       0 0     S   0,0   0,0   0:04.86 kworker/1:1
12 root        20   0 0       0 0     S   0,0   0,0   0:00.17 watchdog/1
13 root        20   0 0       0 0     S   0,0   0,0   0:00.00 migration/2
15 root       20   0 0       0 0     S   0,0   0,0   0:00.04 ksftirqd/2
16 root        20   0 0       0 0     S   0,0   0,0   0:00.15 watchdog/2
17 root        20   0 0       0 0     S   0,0   0,0   0:00.00 migration/3
18 root       20   0 0       0 0     S   0,0   0,0   0:00.00 kworker/3:0
19 root       20   0 0       0 0     S   0,0   0,0   0:00.04 ksftirqd/3

```

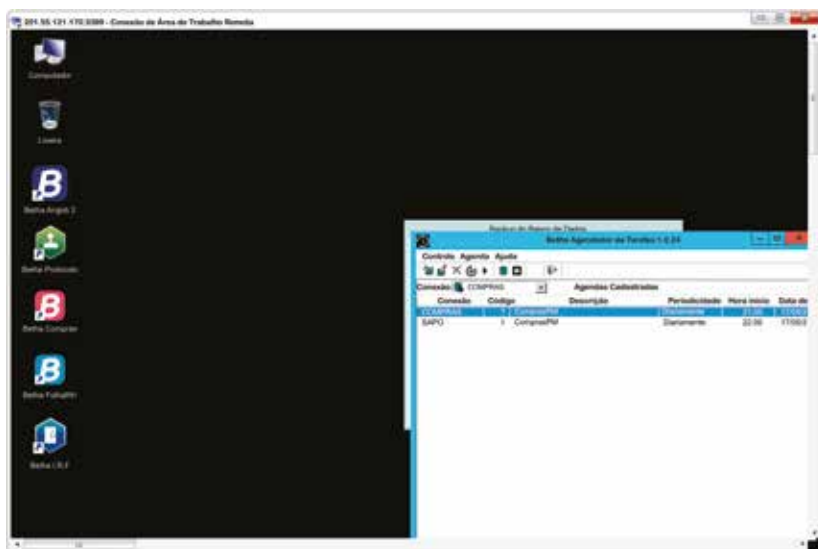
Fonte: Elaborado pelo autor, 2016.

Cliente: Em ambientes distribuídos, o cliente é o computador que consome recursos realizando operações no servidor.

Servidor: Computador do ambiente distribuído responsável por fornecer os recursos onde são realizadas as operações do cliente.

Há ainda o RDP (*Remote Desktop Protocol*), um protocolo utilizado para acessar remotamente ambientes nativos Windows. Para haver o controle via RDP é necessário que o protocolo esteja liberado na máquina de destino e que a máquina que deseja acessá-la possua um cliente RDP. Existem clientes disponíveis em diversos sistemas operacionais. A figura 5 apresenta uma máquina Windows acessando um servidor Windows utilizando o protocolo RDP.

Figura 5 - Máquina Windows com acesso remoto via RDP.



Fonte: Elaborado pelo autor, 2016.

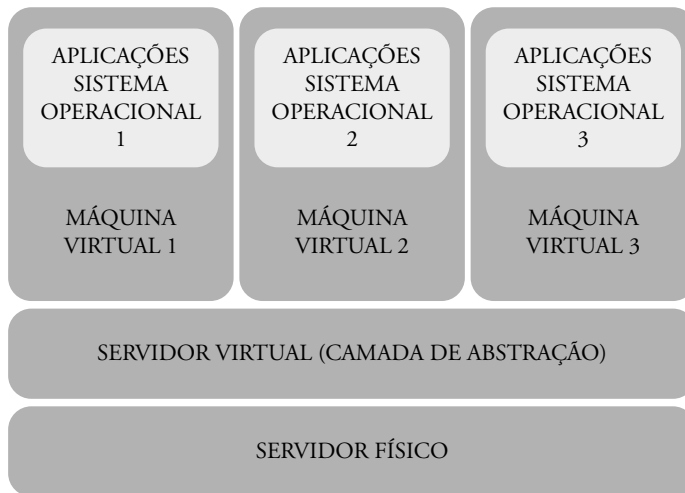
10.3 Virtualização

Para permitir a alocação de sistemas operacionais distintos permitindo a operação distribuída e simultânea em uma mesma infraestrutura computacional, foi criado o conceito de virtualização. Que trata-se da divisão de uma única unidade de processamento para que esta tenha a capacidade de simular a existência de mais de uma CPU por máquina (TANENBAUM; VAN STEEN, 2007).

Na virtualização, o ambiente operacional transforma uma única máquina física em várias outras máquinas iniciadas de forma virtual. Esta abordagem visa reduzir os custos e os espaços físicos da infraestrutura de tecnologia da informação, diminuindo toda a estrutura, multiplicando a existente em diversas instâncias (VERAS, 2011).

A figura 6 ilustra a arquitetura de um ambiente virtualizado quando um único computador, destinado a ser o servidor físico possui instalada a camada de abstração, o servidor virtual. Este servidor de virtualização instanciou mais três máquinas virtuais, cada uma com o seu próprio sistema operacional, possuindo em cada uma delas as aplicações específicas, operando de forma autônoma.

Figura 6 - Arquitetura de Virtualização.



Fonte: Elaborado pelo autor, com base em VERAS, 2011.

Os servidores de virtualização utilizando os dispositivos físico do servidor como placa de rede, discos e memória dedicam partes destes para serem divididos entre as máquinas virtuais que serão criadas. Sendo assim, um computador que possui 8 gigabytes de memória RAM e 1 terabyte de capacidade no disco, pode alocar 2 gigabytes de memória e 100 gigabytes de espaço para cada máquina virtual operar.

O virtualizador cria equipamentos virtuais que simulam o comportamento da placa de rede, placa de som, portas USB e demais dispositivos, fazendo com que as máquinas virtuais possuam as mesmas funcionalidades de um computador comum. Desta forma, é possível obter diversos serviços em máquinas diferentes, rodando sob a mesma infraestrutura (TANENBAUM; VAN STEEN, 2007).

Figura 7 - Virtualizador VirtualBox configurando os dispositivos virtuais.



Fonte: Elaborado pelo autor, 2016.



Saiba mais

A virtualização é a base da Computação em Nuvem, a abordagem que virtualiza a infraestrutura para aplicações via internet.

Para conhecer melhor sobre o conceito de Computação em Nuvem, conheça o Ebook da Endeavor "Cloud Computing na Prática".

<http://info.endeavor.org.br/cloud-computing-na-pratica>



Resumindo

Técnicas como de virtualização e acesso remoto são essenciais dada a elevada situação atual do meio tecnológico. Cada vez mais corporações estão buscando a informatização dos seus processos, sendo assim o investimento em infraestrutura é crescente. Quando existem opções sustentáveis para o controle e a implantação de dispositivos tecnológicos de forma remota e virtualizada, boa parte dos custos com operação e infraestrutura são reduzidos graças à divisão de recursos que estas tecnologias propiciam.

Por estes motivos, você conheceu também os principais sistemas operacionais disponíveis no mercado atual para cada aplicação distinta. Conheceu também as diferenças entre os sistemas para dispositivos móveis com mais uso na atualidade e as operações suportadas em cada sistema operacional.

Aproveite para definir seus próprios objetivos quanto ao uso de implantação de sistemas operacionais e escolha um para aprofundar seus conhecimentos. Afinal, o meio tecnológico precisa cada vez mais de especialidades nos profissionais. Defina a sua!

Conclusão

No decorrer desta disciplina você pôde constatar o quanto os sistemas operacionais estão envolvidos no dia a dia do mundo globalizado. Seja em computadores pessoais ou em grandes servidores, bem como em pequenos dispositivos, sempre há um sistema operacional para gerenciá-los. Pudemos observar que os dispositivos estão cada vez mais inteligentes e conhecer sistemas operacionais para relógios, aparelhos celulares e *tablets*.

Estudamos ainda os sistemas operacionais específicos, desenvolvidos e destinados para aplicações de automatização de processos, com a utilização, inclusive, da antiga forma de processamento monotarefa; além de aprendermos sobre os sistemas de mercado, conhecidos como multitarefa, que possuem diversas versões.

Desta forma, aprofundamos o conhecimento sobre o papel do sistema operacional, bem como o seu funcionamento, e pudemos compreender também desde o momento em que é preciso decidir sobre o sistema operacional a ser utilizado, bem como quais são os sistemas de arquivo disponíveis para cada sistema operacional, passando ainda pela sua instalação, gerenciamento e operação. Na disciplina você aprendeu mais sobre o funcionamento e a gestão de processos dentro do sistema operacional e a sua relação com a memória e os procedimentos de entrada e saída.

Além disso, foi possível ainda conhecer e exemplificar as formas de controle de acesso e gerenciamento de usuários, permissões e grupos dentro de um sistema operacional. Apresentamos ainda as principais causas e soluções para as falhas relacionadas com a segurança e o desempenho de sistemas operacionais. Por fim, estudamos sobre as aplicabilidades de cada um dos principais sistemas operacionais do mercado, bem como as ferramentas de gerenciamento e as técnicas avançadas de operação, como a virtualização e o acesso remoto.

Conforme vimos, a área de sistemas operacionais, embora extensa, continua em constante aprimoramento. Sendo assim, o aperfeiçoamento em técnicas, ferramentas e em sistemas operacionais são essenciais para uma carreira promissora.

Referências

ADRENALINE. **Comparativo de processadores:** a batalha AMD vs Intel no CPU Chart Adrenaline!. Disponível em: <<http://adrenaline.uol.com.br/2015/09/28/37338/comparativo-de-processadores-a-batalha-amd-vs-intel-no-cpu-chart-adrenaline->>. Acesso em 24. jun 2016.

Alterar permissão. **Blog Hospedando Sites.** Disponível em: <<http://blog.hospedandosites.com.br/wp-content/uploads/2012/04/logo36.gif>>. Acesso em: 29 dez. 2015.

APACHE.ORG. Apache JMeter. Disponível em: <<http://jmeter.apache.org>>. Acesso em 17 jan. 2016.

APACHE.ORG. Apache Tomcat. Disponível em: <<http://tomcat.apache.org>>. Acesso em 17 jan. 2016.

CANALTECH. **Como funciona o Hyper-Threading?** <http://canaltech.com.br/dica/produtos/Como-funciona-o-Hyper-Threading/>. Acesso em 12 jan. 2016.

CERT.br. **Cartilha de Segurança para Internet.** Disponível em: <<http://cartilha.cert.br/livro>>. Acesso em: 24 jan. 2016.

COELHO, Flavia Estélia Silva; ARAÚJO, Luiz Geraldo Segadas de Araújo; BEZERRA, Edson Kowask. **Gestão da segurança da informação:** NBR 27001 e NBR 27002. Rio de Janeiro: RNP/ESR, 2014.

COMO executar uma instalação limpa do Windows. **MICROSOFT.** Disponível em: <<http://windows.microsoft.com/pt-BR/windows-8/clean-install>>. Acesso em: 27 jan. 2016.

DEITEL, Harvey M.; DEITEL, Paul J.; CHOFFNES, David R. **Sistemas operacionais.** Pearson Prentice Hall, 2005.

FERRARI, S. R. **Sambando com Linux.** Alta books. 2009. MAZIERO, Carlos Alberto. **Sistemas Operacionais:** Conceitos e Mecanismos, UTFPR, 2013.

GUIA de Instalação de Debian GNU/Linux. **DEBIAN.** Disponível em: <<https://www.debian.org/releases/wheezy/i386/install.pdf.pt>>. Acesso em: 28 nov. 2015.

Interface de comunicação do disco rígido. **Archmemory.** Disponível em: <<http://www.archmemory.com/cm/images/Image/inputid.jpg>>. Acesso em: 29 dez. 2015.

- KUROSE, James F. **Rede de Computadores e a Internet**: uma nova abordagem. Sao Paulo: Addison Wesley, 2003.
- LICENÇAS. **GNU/Linux**. Disponível em: <<http://www.gnu.org/licenses/licenses.pt-br.html>>. Acesso em: 27 jan. 2016.
- MENUETOS. Disponível em: <<http://www.menuetos.net>>. Acesso em 25 jan. 2016.
- MICROSOFT. **About processes and Threads**. Disponível em <https://msdn.microsoft.com/pt-br/library/windows/desktop/ms681917%28v=vs.85%29.aspx>. Acesso em 09/12/2015
- MONTEIRO, Claudio de Castro. **Sistemas Operacionais**. Curitiba: Fael, 2013.
- MORIMOTO, Carlos **Linux**: Gerenciamento de usuários, grupos e permissões. 2008. Disponível em: <http://www.hardware.com.br/tutoriais/usuarios-grupos-permissoes/>. Acesso: 28 dez. 2015.
- MOROZ, Maiko Rossano. **Critérios para adoção e seleção de sistemas operacionais embarcados**. 2011. 75 f. Dissertação Programa de Pós Graduação em Eng. Elétrica e Informática Industrial, Universidade Tecnológica Federal do Paraná. Curitiba, 2011.
- NEMETH, Evi; HEIN, Trent R.; SNYDER, Garth. **Manual completo do Linux**: guia do administrador. 2004.
- NETMARKETSHARE. Disponível em: <<http://netmarketshare.com>>. Acesso em: 18 jan. 2016.
- OGLETREE, Terry W. **Dominando Microsoft Windows XP**. 1 ed. Pearson, 2002.
- PORQUÊ o mascote do Linux é um pinguim. **Viva o Linux**. Disponível em: <<https://www.vivaolinux.com.br/artigo/Porque-a-mascote-do-Linux-e-um-pinguim>>. Acesso em: 28 jan. 2016.
- PROGUE, David. **Mac OS X Snow Leopard**: O Manual que Faltava. Digerati Books, 2010.

RUDOLPH, S., *et al.* **Instalando Debian GNU/Linux 3.0 para Intel x86, versão 3.0.24.** Dezembro, 2002. Disponível em: <<http://www.br.debian.org/releases/stable/i386/install.pt.txt>>. Acesso em: 28 dez. 2015.

SÊMOLA, Marcos et al. **Gestão da segurança da informação.** Elsevier Brasil, 2003.

SILBERSCHATZ, Abraham; GALVIN, Peter Baer; GAGNE, Greg. **Fundamentos de Sistemas Operacionais.** 7ª Ed. Editora McGraw. 2008

SILBERSCHATZ, Abraham. **Sistemas operacionais com Java.** Elsevier Brasil, 2008.

SILBERSCHATZ, Abraham; GALVIN, Peter Baer. **Sistemas Operacionais com Java.** 7ª Ed. Editora Campus. 2008

SILVA, Alexandre Tadeu Rossini da, et.al. **Fundamentos de informática.** Curitiba: Fael, 2013.

STALLINGS, William. **Arquitetura e Organização de Computadores: projeto para o desempenho.** 5 ed. São Paulo: Prentice Hall, 2002.

TANENBAUM, Andrew S. **Redes de computadores.** São Paulo: Pearson Prentice Hall, 2011.

TANENBAUM, Andrew S; VAN STEEN, Maarten. **Sistemas Distribuídos: princípios e paradigmas.** 2 ed. Pearson Prentice Hall, 2007.

TANENBAUM, Andrew S. **Sistemas Operacionais Modernos.** 3 ed. Pearson Prentice Hall, 2009.

Técnica de DMA. **Imgur.** Disponível em: <<http://i.imgur.com/8JLWR2p.png>>. Acesso em: 29 dez. 2015.

TOSCANI, S.; OLIVEIRA, R. S.; CARISSIMI, A. **Sistemas operacionais.** Bookman Companhia: 2010

VERAS, Manoel. **Virtualização: Componente Central do Datacenter.** Brasport, 2011.

Viva o Linux. Disponível em: <www.vivaolinux.com.br>. Acesso em: 29 dez. 2015.



