



Sistemas Operacionais

Adriana Soares Pereira

Elisa Maria Vissotto

Roberto Franciscatto



Frederico Westphalen - RS
2015

Presidência da República Federativa do Brasil
Ministério da Educação
Secretaria de Educação Profissional e Tecnológica

© Colégio Agrícola de Frederico Westphalen
Este caderno foi elaborado em parceria entre o Colégio Agrícola de Frederico Westphalen – CAFW e a Universidade Federal de Santa Maria para a Rede e-Tec Brasil.

Equipe de Elaboração
Colégio Agrícola de Frederico Westphalen – CAFW

Reitor
Paulo Afonso Burmann/UFSM

Direção
Fernando de Cristo/CAFW

Coordenação Geral da Rede e-Tec – UFSM
Paulo Roberto Colusso/CTISM

Coordenação de Curso
Adriana Soares Pereira/CAFW

Professor-autor
Adriana Soares Pereira/CAFW
Elisa Maria Vissotto/CAFW
Roberto Franciscatto/CAFW

Equipe de Acompanhamento e Validação
Colégio Técnico Industrial de Santa Maria – CTISM

Coordenação Institucional
Paulo Roberto Colusso/CTISM

Coordenação de Design
Erika Goellner/CTISM

Revisão Pedagógica
Elisiane Bortoluzzi Scrimini/CTISM
Jaqueline Müller/CTISM

Revisão Textual
Carlos Frederico Ruviaro/CTISM

Revisão Técnica
Tatiani Elenusa de Oliveira Rodrigues/CTISM

Ilustração
Marcel Santos Jacques/CTISM
Ricardo Antunes Machado/CTISM

Diagramação
Leandro Felipe Aguiar Freitas/CTISM

Bibliotecária Nataly Soares Leite – CRB 10/1981

P436s **Pereira, Adriana Soares.**
Sistemas operacionais / Adriana Soares Pereira, Elisa Maria Vissotto, Roberto Franciscatto. – Frederico Westphalen : Universidade Federal de Santa Maria, Colégio Agrícola de Frederico Westphalen, 2015.
110 p. : il.
ISBN 978-85-63573-77-3

1. Sistemas operacionais. 2. Gerenciamento de processos. 3. Gerenciamento de memória. 4. Gerenciamento de arquivos. I. Vissotto, Elisa Maria. II. Franciscatto, Roberto. III. Universidade Federal de Santa Maria. Colégio Agrícola de Frederico Westphalen. III. Título.

CDU 004.451

Apresentação e-Tec Brasil

Prezado estudante,
Bem-vindo a Rede e-Tec Brasil!

Você faz parte de uma rede nacional de ensino, que por sua vez constitui uma das ações do Pronatec – Programa Nacional de Acesso ao Ensino Técnico e Emprego. O Pronatec, instituído pela Lei nº 12.513/2011, tem como objetivo principal expandir, interiorizar e democratizar a oferta de cursos de Educação Profissional e Tecnológica (EPT) para a população brasileira propiciando caminho de o acesso mais rápido ao emprego.

É neste âmbito que as ações da Rede e-Tec Brasil promovem a parceria entre a Secretaria de Educação Profissional e Tecnológica (SETEC) e as instâncias promotoras de ensino técnico como os Institutos Federais, as Secretarias de Educação dos Estados, as Universidades, as Escolas e Colégios Tecnológicos e o Sistema S.

A educação a distância no nosso país, de dimensões continentais e grande diversidade regional e cultural, longe de distanciar, aproxima as pessoas ao garantir acesso à educação de qualidade, e promover o fortalecimento da formação de jovens moradores de regiões distantes, geograficamente ou economicamente, dos grandes centros.

A Rede e-Tec Brasil leva diversos cursos técnicos a todas as regiões do país, incentivando os estudantes a concluir o ensino médio e realizar uma formação e atualização contínuas. Os cursos são ofertados pelas instituições de educação profissional e o atendimento ao estudante é realizado tanto nas sedes das instituições quanto em suas unidades remotas, os polos.

Os parceiros da Rede e-Tec Brasil acreditam em uma educação profissional qualificada – integradora do ensino médio e educação técnica, – é capaz de promover o cidadão com capacidades para produzir, mas também com autonomia diante das diferentes dimensões da realidade: cultural, social, familiar, esportiva, política e ética.

Nós acreditamos em você!
Desejamos sucesso na sua formação profissional!

Ministério da Educação
Janeiro de 2015

Nosso contato
etecbrasil@mec.gov.br



Indicação de ícones

Os ícones são elementos gráficos utilizados para ampliar as formas de linguagem e facilitar a organização e a leitura hipertextual.



Atenção: indica pontos de maior relevância no texto.



Saiba mais: oferece novas informações que enriquecem o assunto ou “curiosidades” e notícias recentes relacionadas ao tema estudado.



Glossário: indica a definição de um termo, palavra ou expressão utilizada no texto.



Mídias integradas: sempre que se desejar que os estudantes desenvolvam atividades empregando diferentes mídias: vídeos, filmes, jornais, ambiente AVEA e outras.



Atividades de aprendizagem: apresenta atividades em diferentes níveis de aprendizagem para que o estudante possa realizá-las e conferir o seu domínio do tema estudado.



Sumário

Palavra do professor-autor	9
Apresentação da disciplina	11
Projeto instrucional	13
Aula 1 – Introdução aos sistemas operacionais	15
1.1 Considerações iniciais.....	15
1.2 Funções básicas e caracterização.....	15
1.3 Tipos de sistemas operacionais.....	17
1.4 Sistemas operacionais usuais.....	20
Aula 2 – Conceitos de <i>hardware</i> e <i>software</i>	23
2.1 Considerações iniciais.....	23
2.2 <i>Hardware</i>	23
2.3 <i>Software</i>	30
Aula 3 – Estrutura do sistema operacional	35
3.1 Considerações iniciais.....	35
3.2 Funções do núcleo.....	35
3.3 Modo de acesso.....	36
3.4 Rotinas do sistema operacional e <i>system calls</i>	37
3.5 Operação do sistema de computação.....	38
3.6 Interrupções e exceções.....	39
3.7 Estrutura de acesso direto à memória (DMA).....	40
3.8 <i>Buffering</i>	41
3.9 <i>Spooling</i>	41
3.10 Reentrância.....	42
Aula 4 – Gerenciamento de processos	45
4.1 Considerações iniciais.....	45
4.2 O que é um processo?.....	45
4.3 Estrutura do processo.....	45

Aula 5 – Gerenciamento de memória	51
5.1 Considerações iniciais.....	51
5.2 Funções básicas da memória.....	51
5.3 Técnicas de gerenciamento de memória.....	53
Aula 6 – Gerenciamento de arquivos	63
6.1 Considerações iniciais.....	63
6.2 Arquivos.....	63
6.3 Diretórios.....	68
6.4 Alocação de espaço em disco.....	69
6.5 Proteção de acesso.....	72
Aula 7 – Estudo de caso: sistema operacional Windows Seven	75
7.1 Considerações iniciais.....	75
7.2 Requisitos.....	75
7.3 Instalação.....	76
7.4 Ferramentas do sistema.....	77
7.5 Principais comandos de administração do sistema.....	84
Aula 8 – Estudo de caso: sistema operacional Linux Ubuntu	91
8.1 Considerações iniciais.....	91
8.2 Requisitos.....	91
8.3 Instalação.....	92
8.4 Árvore de diretórios.....	96
8.5 Ferramentas de administração do sistema.....	99
8.6 Principais comandos de administração do sistema operacional.....	104
Referências	109
Currículo do professor-autor	110

Palavra do professor-autor

Prezado aluno, é com grande alegria e satisfação que chegamos até você por meio deste livro didático de Sistemas Operacionais. Para se tornar um bom técnico em informática é fundamental que tenhamos domínio dos sistemas operacionais, sua forma de funcionamento e organização, bem como, sua utilização prática no dia a dia. Para isso, além de estudá-lo é necessário usá-lo e querer conhecê-lo cada vez mais, sabendo sobre suas várias categorias, como os sistemas operacionais para servidores, *desktops* e dispositivos móveis.

Com este livro didático, pretendemos colaborar no aprendizado da disciplina de Sistemas Operacionais, para que você possa tirar proveito ao máximo desta disciplina e do curso como um todo. Neste material você conhecerá as funções básicas e a caracterização dos sistemas operacionais, os conceitos de *hardware* e *software* utilizados pelos mesmos, a estrutura por trás de um sistema operacional, como o sistema operacional gerencia os processos, memória, arquivos e diretórios, bem como, duas aulas especiais que abordam na prática o sistema operacional Windows Seven e o sistema operacional Linux Ubuntu.

Para um bom uso deste material, recomendamos a leitura completa do mesmo, bem como, a resolução dos exercícios propostos e o acesso ao conteúdo complementar indicado durante as aulas.

Desejamos que este livro possa ajudá-lo de forma efetiva em sua caminhada junto ao curso e em suas novas descobertas. Não se esqueça de ler constantemente este material, além de acompanhar a disciplina em seu ambiente virtual de aprendizagem, assim como, interagir com professores, tutores e colegas.

Lembre-se que o seu sucesso depende de seu esforço e sua dedicação.

Um grande abraço e bons estudos!

Adriana Soares Pereira

Elisa Maria Vissotto

Roberto Franciscatto



Apresentação da disciplina

Na disciplina de Sistemas Operacionais, você irá ter uma visão geral de como os sistemas operacionais se organizam, se estruturam e como funcionam. Para que este universo seja entendido, as aulas deste livro foram organizadas da seguinte forma: Na Aula 1, são apresentados os conceitos referentes aos sistemas operacionais, suas funções básicas, caracterização e os principais tipos de sistemas operacionais. Na Aula 2, são apresentados os conceitos de *hardware* e *software* que interagem diretamente com os sistemas operacionais. Na Aula 3, a estrutura do sistema operacional é apresentada com ênfase para as funções do núcleo do sistema, modos de acesso e rotinas do sistema operacional. Na Aula 4, são apresentados os processos e sua forma de gerenciamento. Na Aula 5, o gerenciamento de memória é descrito, bem como, as principais técnicas para alocação da mesma, utilizada pelos sistemas operacionais. Na Aula 6, o gerenciamento de arquivos é proposto, com ênfase para a forma de organização de arquivos e diretórios, bem como as técnicas de alocação de espaço em disco. Na Aula 7, é abordado de forma prática o sistema operacional Windows Seven, bem como, suas principais ferramentas de administração. Por fim, na Aula 8, é apresentado um estudo de caso envolvendo o sistema operacional Linux Ubuntu, sua forma de organização, suas ferramentas do sistema e os principais comandos para administração.



Projeto instrucional

Disciplina: Sistemas Operacionais (carga horária: 60h).

Ementa: Compreender os conceitos relacionados a projeto de sistemas operacionais e sua implementação. Utilizar procedimentos necessários para manter programas e administrar usuários em um sistema operacional.

AULA	OBJETIVOS DE APRENDIZAGEM	MATERIAIS	CARGA HORÁRIA (horas)
1. Introdução aos sistemas operacionais	Apresentar as funções básicas e caracterização dos sistemas operacionais. Mostrar os principais tipos de sistemas operacionais existentes. Conhecer os sistemas operacionais mais usuais.	Ambiente virtual: plataforma Moodle. Apostila didática. Recursos de apoio: <i>links</i> , exercícios.	07
2. Conceitos de <i>hardware</i> e <i>software</i>	Apresentar os principais dispositivos de <i>hardware</i> . Conceituar <i>software</i> voltado ao contexto dos sistemas operacionais. Mostrar a importância do <i>hardware</i> e <i>software</i> em um sistema computacional.	Ambiente virtual: plataforma Moodle. Apostila didática. Recursos de apoio: <i>links</i> , exercícios.	08
3. Estrutura do sistema operacional	Conhecer as principais funções que compõem um sistema operacional. Entender a estrutura de um sistema operacional. Ter um entendimento das principais técnicas utilizadas nos sistemas operacionais.	Ambiente virtual: plataforma Moodle. Apostila didática. Recursos de apoio: <i>links</i> , exercícios.	07
4. Gerenciamento de processos	Entender o conceito de processos e sua função junto ao sistema operacional. Conhecer seu ciclo de vida e atividades que desempenham. Mostrar os recursos de <i>hardware</i> e <i>software</i> associados aos processos.	Ambiente virtual: plataforma Moodle. Apostila didática. Recursos de apoio: <i>links</i> , exercícios.	08
5. Gerenciamento de memória	Entender como o sistema operacional gerencia a memória. Conhecer as principais técnicas para alocação de memória. Visualizar a importância da utilização da memória nos sistemas computacionais.	Ambiente virtual: plataforma Moodle. Apostila didática. Recursos de apoio: <i>links</i> , exercícios.	07

AULA	OBJETIVOS DE APRENDIZAGEM	MATERIAIS	CARGA HORÁRIA (horas)
6. Gerenciamento de arquivos	<p>Apresentar as principais técnicas de gerenciamento de arquivos e diretórios.</p> <p>Entender o processo de organização de arquivos e diretórios.</p> <p>Compreender as principais técnicas de alocação utilizadas no gerenciamento de arquivos e diretórios.</p>	<p>Ambiente virtual: plataforma Moodle.</p> <p>Apostila didática.</p> <p>Recursos de apoio: <i>links</i>, exercícios.</p>	08
7. Estudo de caso: sistema operacional Windows Seven	<p>Conhecer de forma prática o sistema operacional Windows Seven.</p> <p>Compreender seus requisitos e forma de instalação.</p> <p>Entender as principais ferramentas do sistema e como podemos utilizá-las na administração do sistema operacional.</p> <p>Conhecer os principais comandos funcionais de administração.</p>	<p>Ambiente virtual: plataforma Moodle.</p> <p>Apostila didática.</p> <p>Recursos de apoio: <i>links</i>, exercícios.</p>	07
8. Estudo de caso: sistema operacional Linux Ubuntu	<p>Conhecer de forma prática o sistema operacional Linux Ubuntu.</p> <p>Compreender seus requisitos e forma de instalação.</p> <p>Conhecer as principais ferramentas de administração do sistema.</p> <p>Compreender os principais comandos de administração do sistema operacional.</p>	<p>Ambiente virtual: plataforma Moodle.</p> <p>Apostila didática.</p> <p>Recursos de apoio: <i>links</i>, exercícios.</p>	08

Aula 1 – Introdução aos sistemas operacionais

Objetivos

Apresentar as funções básicas e caracterização dos sistemas operacionais.

Mostrar os principais tipos de sistemas operacionais existentes.

Conhecer os sistemas operacionais mais usuais.

1.1 Considerações iniciais

Nessa aula, começaremos a conhecer melhor o que é um sistema operacional, como se estrutura, sua organização e sua forma de funcionamento. Ao longo dos capítulos, será possível entender os sistemas operacionais em sua essência e como os mesmos evoluíram ao longo do tempo.

1.2 Funções básicas e caracterização

Um sistema operacional pode ser caracterizado como um conjunto de rotinas executadas pelo processador, de forma semelhante aos programas dos usuários. Seu principal objetivo é gerenciar os componentes de *hardware*, como processador, memória principal, discos, teclado, entre outros e fornecer aos programas do usuário uma interface com o *hardware* mais simples de ser utilizada.

Sem um sistema operacional, o usuário deveria ter um conhecimento aprofundado de diversos comandos e linguagens em geral para que pudesse manipular o computador, o que tornaria uma prática difícil e com grandes possibilidades de erro.

A principal diferença existente entre *softwares* aplicativos e um sistema operacional convencional está no fato de que os dois funcionam de formas diferentes. Enquanto um *software* aplicativo (Microsoft Word, Libre Office Writer, etc.) trabalha com início, meio e fim, um sistema operacional trabalha de forma assíncrona, ou seja, suas rotinas são executadas de forma concorrente através de eventos assíncronos (que podem ocorrer a qualquer momento).

Quanto às denominações para sistema operacional existem várias, que especificam as rotinas que ele executa, portanto, muitas vezes ele é chamado de modo supervisor, modo núcleo, privilegiado, entre outros.

Quanto às funções básicas que um sistema operacional é capaz de realizar, temos duas principais, que são: **facilitar o acesso aos recursos do sistema e compartilhamento de recursos de forma organizada e protegida.**

Para exemplificar melhor a estrutura de um sistema operacional e como este se relaciona com os demais componentes computacionais, vejamos a Figura 1.1.

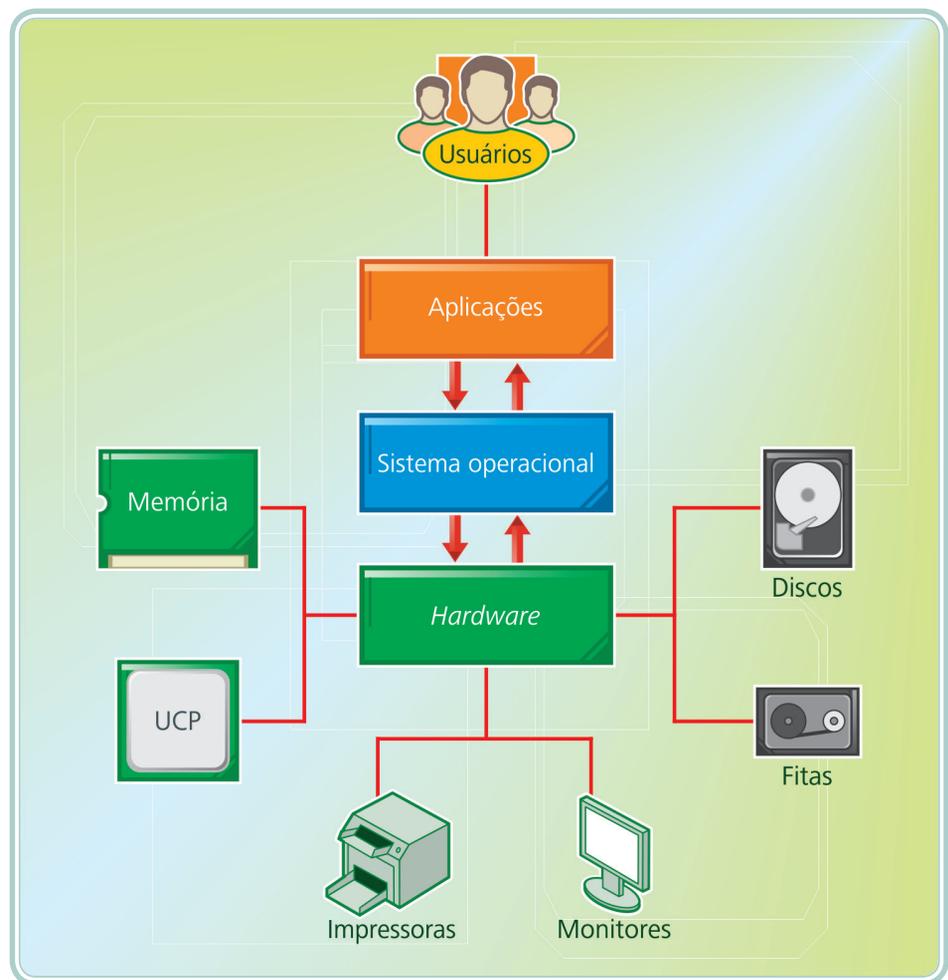


Figura 1.1: Visão de um sistema computacional

Fonte: CTISM, adaptado de Maia, 2007

Na Figura 1.1, é possível observar que no topo de um sistema computacional estão os usuários. Estes se comunicam com os computadores através das aplicações diversas, como editores de textos, planilhas eletrônicas, navegadores (Chrome, Internet Explorer, Firefox, etc.), entre outros. Estes *softwares*

aplicativos são gerenciados por um sistema operacional (tema central deste material) que, conforme as setas da Figura 1.1, fazem a ligação entre os *softwares* aplicativos instalados e os componentes de *hardware* existentes no computador. Esta estrutura é o ponto de partida para nossos estudos sobre os sistemas operacionais e sua forma de funcionamento.

1.3 Tipos de sistemas operacionais

Os sistemas operacionais evoluíram ao longo do tempo, muito pela contribuição da evolução do *hardware* e das aplicações suportadas por ele. Os sistemas operacionais são divididos basicamente em três tipos, conforme podemos visualizar na Figura 1.2.

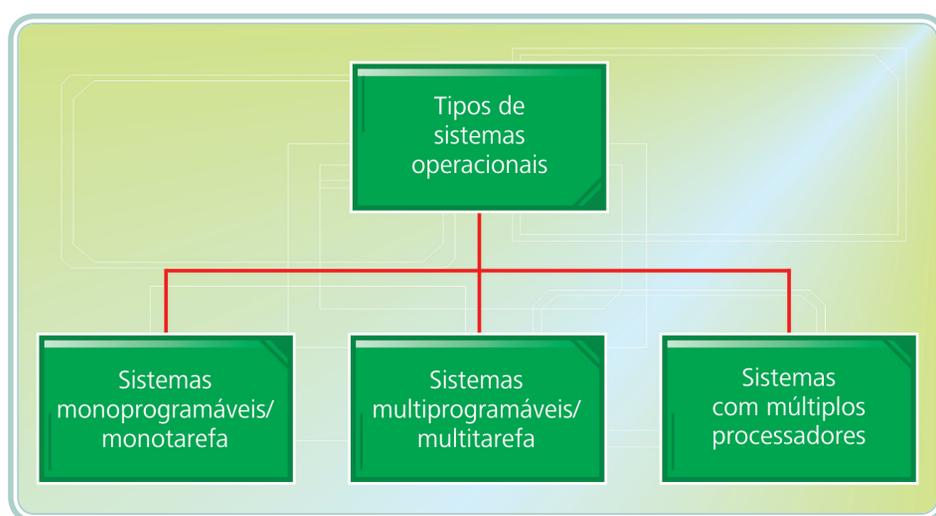


Figura 1.2: Tipos de sistemas operacionais

Fonte: CTISM, adaptado de Maia, 2007

A estrutura apresentada na Figura 1.2, mostra os tipos de sistemas operacionais quanto a sua forma de operação. Estes dividem-se em: sistemas monoprogramáveis/monotarefa, sistemas multiprogramáveis/multitarefa e sistemas com múltiplos processadores. Estudaremos cada um deles a partir de agora, confira.

1.3.1 Sistemas monoprogramáveis/monotarefa

Os sistemas monoprogramáveis ou monotarefa, como o próprio nome já diz (mono, noção de um, de unidade) são sistemas voltados tipicamente para a execução de um único programa. Qualquer outra aplicação para ser executada, deve aguardar o término no programa corrente. Neste tipo de sistema, o processador, a memória e os demais periféricos permanecem exclusivamente dedicados à execução de um único programa.



MS-DOS foi o primeiro sistema operacional da Microsoft (funcionava exclusivamente pela linha de comandos, ou seja, executando comandos digitados pelo usuário) que passou a integrar os computadores da IBM no ano de 1981, tornando-se o sistema base para a maioria dos computadores pessoais. O MS-DOS teve várias versões e aprimoramentos. Com o passar dos anos e o lançamento do Windows no ano de 1985, o MS-DOS passou a ser parte integrante das versões Windows e continua até hoje através do utilitário "Prompt de Comando".

Os sistemas monotarefa, estão tipicamente relacionados aos primeiros computadores da década de 1960. Os sistemas monoprogramáveis possuíam uma desvantagem muito clara, devido à limitação de tarefas (uma de cada vez) havia um grande desperdício de recursos de *hardware*. Como exemplo de sistema operacional monoprogramável/monotarefa temos o MS-DOS da Microsoft.

1.3.2 Sistemas multiprogramáveis/multitarefa

Diferentemente dos sistemas monoprogramáveis/monotarefa, os sistemas multiprogramáveis/multitarefa permitem que os recursos computacionais sejam compartilhados entre os diversos usuários e aplicações. Neste caso, enquanto um programa espera pela ocorrência de um evento, outros programas podem estar em execução neste mesmo intervalo de tempo, permitindo assim o compartilhamento de recursos como processador, memória principal e dispositivos de entrada e saída. O sistema operacional fica incumbido de gerenciar o acesso concorrente aos seus diversos recursos de forma ordenada e protegida.

As vantagens na utilização destes tipos de sistemas operacionais são a redução do tempo de respostas das aplicações, além dos custos computacionais reduzidos, devido ao compartilhamento dos recursos do sistema entre as diferentes aplicações.

Os sistemas operacionais multiprogramáveis/multitarefa podem ser classificados em três subdivisões segundo características de como suas aplicações são gerenciadas. Estas categorias são: sistemas **batch**, **de tempo compartilhado** e **de tempo real**.

1.3.2.1 Sistemas *batch*

Os sistemas *batch* foram implementados na década de 60. Os programas ou *jobs*, como eram conhecidos na época, eram submetidos para execução através da utilização de cartões perfurados, armazenados em discos ou fitas, para posteriormente serem executados (dependendo da disponibilidade da memória principal). Uma característica marcante dos sistemas *batch* era não exigir a interação do usuário com a aplicação. Exemplos de aplicações processadas em *batch* eram programas de cálculos numéricos, ordenações, compilações, *backups*, entre outros, onde não se fazia necessária a interação com o usuário.

1.3.2.2 Sistemas de tempo compartilhado

Estes sistemas, também conhecidos como *time-sharing* (tempo compartilhado), permitem que diferentes programas sejam executados a partir da divisão

do tempo do processador em pequenas fatias de tempo (conhecidas como *time-slice*). Caso a fatia de tempo seja pequena para as funções que o mesmo precisa realizar, ele aguarda uma nova fatia de tempo para que possa entrar em execução novamente. Nos sistemas de tempo compartilhado é criado, para cada usuário, um ambiente de trabalho próprio, simulando a ideia de que todo o sistema está dedicado exclusivamente a ele.

Os sistemas de tempo compartilhado permitem aos seus usuários interagir com o sistema através dos dispositivos de entrada de dados e comandos especiais. A grande maioria das aplicações comerciais existentes atualmente utilizam este tipo de sistema, uma vez que oferecem tempo de resposta razoáveis e custos baixos, em função do compartilhamento dos recursos do sistema, entre os programas.

1.3.2.3 Sistemas de tempo real

Também conhecidos como *real-time* (tempo real) possuem características semelhantes aos sistemas de tempo compartilhado, entretanto diferenciam-se pelo tempo exigido no processamento das aplicações. Nos sistemas de tempo real, quanto aos tempos de processamento, estes devem enquadrar-se em limites rígidos, para o êxito das operações realizadas, podendo comprometer a aplicação e seus resultados caso este limite de tempo não seja cumprido.

Diferentemente dos sistemas de tempo compartilhado, nos sistemas de tempo real o processador permanece ocupado durante o tempo que for necessário a execução de determinado programa, cedendo lugar a outro programa que tenha uma prioridade maior no sistema. Outra característica destes sistemas é que a prioridade de execução de um programa é definida pela própria aplicação e não pelo sistema operacional.

Exemplos destes sistemas, encontramos em aplicações de controle de processos, como controle de tráfego aéreo, usinas, refinarias, ou qualquer outra aplicação onde o tempo de processamento é fator fundamental para o sucesso.

1.3.3 Sistemas com múltiplos processadores

Os sistemas de múltiplos processadores recebem este nome por possuírem dois ou mais processadores interligados trabalhando em conjunto. Como vantagem desta arquitetura está o fato de permitir que vários programas possam ser executados ao mesmo tempo ou que um programa possa ser dividido em partes, entre os vários processadores, executando-os de forma simultânea.

Um dos fatores fundamentais para o desenvolvimento de sistemas operacionais com múltiplos processadores está na forma de comunicação entre os processadores (CPU), o compartilhamento de memória principal (RAM) e os dispositivos de entrada e saída (E/S).

Através dos sistemas com múltiplos processadores foi possível a criação de sistemas computacionais voltados para o desenvolvimento científico, aplicado em áreas como desenvolvimento aeroespacial, prospecção de petróleo, simulações, entre outros.

As práticas e estudos voltados ao desenvolvimento de sistemas com múltiplos processadores adicionaram vantagens em tais sistemas como a escalabilidade, disponibilidade e balanceamento de carga.

1.4 Sistemas operacionais usuais

A ideia desta seção é que possamos conhecer um pouco mais dos sistemas operacionais mais usuais e suas classificações, para que você possa testar, usar e começar a praticar em seu dia a dia, desde já.

1.4.1 Sistemas operacionais de computadores pessoais

Os sistemas operacionais para computadores pessoais são amplamente usados no dia a dia em *netbooks*, *notebooks*, computadores de mesa, etc. Seu objetivo é fornecer uma boa interface, permitindo que o usuário realize as tarefas que necessita de forma prática e intuitiva. Estes sistemas operacionais são amplamente utilizados para pacotes de escritório (editores de texto, planilhas eletrônicas), internet e aplicativos em geral.

Alguns exemplos mais comuns destes sistemas operacionais para computadores pessoais são as distribuições Windows e Linux.

Quanto às distribuições Windows para computadores pessoais, temos: Windows XP, Windows Vista, Windows Seven e Windows 8.1.

Já as distribuições Linux para computadores pessoais, temos como os mais usuais: Ubuntu, Red Hat, Debian, Fedora, Mint, Mageia, OpenSuse, entre outros.

As distribuições Windows são sistemas operacionais pagos, onde devemos escolher e adquirir as licenças para uso, conforme cada necessidade. Já os sistemas operacionais Linux, são caracterizados como *softwares* livres, ou

seja, podem ser baixados livremente na internet, modificados, adaptados e distribuídos livremente independente do número de computadores onde iremos realizar a sua instalação e posterior utilização.

1.4.2 Sistemas operacionais de servidores

Diferentemente dos sistemas operacionais para computadores pessoais, o objetivo dos sistemas operacionais para servidores é servir o maior número de usuários ao mesmo tempo, permitindo a eles compartilhar recursos de *hardware* e *software*.

Os sistemas operacionais de servidores podem fornecer diferentes tipos de serviços, como por exemplo: servidor de arquivos, servidor *web* (hospedagem de *site*, *e-mail*, *proxy*, entre outros) servidor de autenticação, *backup*, compartilhamento, entre outros.

Assim como acontece nos sistemas operacionais para computadores pessoais, os sistemas operacionais para servidores possuem suas distribuições específicas, uma vez que sua função é diferenciada e que necessita de um *hardware* específico para seu pleno funcionamento (na grande maioria dos casos).

Dessa forma, as distribuições Windows e Linux (mais usuais) dispõem de vários sistemas operacionais para servidores, sendo que no Windows, podemos citar como exemplo: Windows 2003 Server, Windows 2008 server e Windows 2012 Server. Nas distribuições Linux, temos como exemplo de sistemas operacionais para servidores: Ubuntu Server, Mandriva, Slackware, Suse e Debian.

1.4.3 Sistemas operacionais embarcados

Este tipo de sistema operacional tem crescido bastante nos últimos anos e podemos dizer que os mesmos são uma tendência cada vez maior. Quando nos referimos aos sistemas operacionais embarcados, estamos falando dos sistemas dos computadores de mão (*smartphones*, *tablets*, etc.). Os sistemas operacionais embarcados são executados diretamente nos dispositivos e já vêm instalados de fábrica nos mesmos, não permitindo a troca por outro sistema operacional, características de tais sistemas. Como diferença para os demais sistemas operacionais, apresentam restrições de tamanho, memória e consumo de energia, o que os fazem especiais.

Neste mercado, podemos citar como principais sistemas operacionais embarcados, os seguintes: Android, iOS, Windows Phone, entre outros.



Alguns dos repositórios mais usuais em cada distribuição, confira.

Android – Google Play
https://play.google.com/store?hl=pt_BR

iOS – iTunes
<https://www.apple.com/br/itunes/charts/free-apps/>

Windows Phone – Store
<http://www.windowsphone.com/pt-br/store>

Para utilização de aplicativos diversos nestes sistemas, faz-se necessário realizar o *download* dos aplicativos que o usuário deseja no repositório oficial de cada sistema operacional em particular.

Resumo

Nessa aula, vimos uma introdução sobre os sistemas operacionais, onde foi possível conhecer as funções básicas e caracterização dos mesmos, os principais tipos de sistemas operacionais, bem como os sistemas operacionais mais usuais. Em nossa próxima aula, abordaremos os principais conceitos relacionados a *hardware* e a *software*, com enfoque destes para o contexto dos sistemas operacionais.



Atividades de aprendizagem

1. Quais são as funções básicas de um sistema operacional?
2. Descreva a diferença entre sistemas monoprogramáveis/monotarefa, multiprogramáveis/multitarefa e de múltiplos processadores.
3. Faça uma pesquisa na internet sobre os três principais sistemas operacionais para computadores pessoais, servidores e dispositivos móveis (*smartphones* e *tablets*).
4. Construa uma tabela com os sistemas operacionais encontrados na atividade 3 e descreva as características e principais funcionalidades de cada sistema operacional.

Aula 2 – Conceitos de *hardware* e *software*

Objetivos

Apresentar os principais dispositivos de *hardware*.

Conceituar *software* voltado ao contexto dos sistemas operacionais.

Mostrar a importância do *hardware* e *software* em um sistema computacional.

2.1 Considerações iniciais

Nessa aula abordaremos os principais conceitos relacionados a *hardware* e *software* e como estes estão diretamente ligados ao projeto de sistemas operacionais. Ao longo das aulas será possível entender um pouco mais sobre os principais dispositivos de *hardware*, sua forma de funcionamento e os *softwares* utilitários usados no contexto dos sistemas operacionais.

2.2 Hardware

Podemos dizer que *hardware* e *software* são elementos vitais em um sistema de computação e que um não vive sem o outro. Em um sistema computacional temos um conjunto de *hardware* formado por processadores, memória, barramentos, registradores, monitores de vídeo, *mouse*, teclado, impressoras, discos magnéticos, entre outros dispositivos. Estes por sua vez manipulam dados de forma digital o que torna um sistema computacional confiável na representação e transmissão de dados.

Um sistema computacional geralmente é formado pela composição de três subsistemas básicos (também conhecidos como unidades funcionais): processadores (UCP), memória principal (RAM) e dispositivos de entrada e saída (E/S). Estes itens compõem qualquer tipo de computador digital existente independente de fabricante, marca ou modelo.

2.2.1 Processador

O processador ou Unidade Central de Processamento (UCP) tem como função controlar e executar instruções presentes na memória principal (RAM) através de

operações aritméticas (como somar, subtrair, multiplicar e dividir) e operações lógicas (comparações, movimentação de dados, etc.).

Um processador é composto por alguns elementos básicos como a Unidade de Controle (UC), a Unidade Lógica e Aritmética (ULA) e os registradores.

A UC gerencia as atividades de todos os dispositivos do computador, como por exemplo, a gravação de dados em discos, a busca de instrução em memória, entre outros. Já a ULA realiza as operações lógicas e aritméticas, conforme citado nos parágrafos anteriores. Os registradores por sua vez têm a função de guardar dados temporariamente, servindo como uma memória de alta velocidade interna do processador, porém com uma capacidade de armazenamento reduzida e um custo maior que a memória principal.

A-Z

clock

É um circuito oscilador que tem a função de sincronizar e ditar a medida de velocidade de transferência de dados entre duas partes essenciais de um processamento. Em outras palavras, refere-se a medida de frequência com que as operações são realizadas (ciclos por segundo).

A sincronização entre as funções do processador se dá através de um sinal de **clock**. Este sinal por sua vez, é gerado através de um pulso cíclico que altera variáveis de estado do processador.

2.2.2 Arquiteturas RISC e CISC

Uma CPU em relação ao número de instruções de processamento que pode reconhecer classifica-se em: CISC e RISC.

2.2.2.1 CISC

Um processador CISC reconhece mais de uma centena de instruções. Devido a esta característica, pode executar de modo direto a maioria das operações programadas pelos modernos *softwares*.

Os processadores com arquitetura CISC possuem instruções complexas que são interpretadas por microprogramas. O núcleo de registradores é pequeno e qualquer instrução pode referenciar a memória principal.

Os processadores são em sua grande maioria *chips* CISC. Quanto maior a quantidade de instruções que um microprocessador é capaz de identificar, mais lenta é a execução de cada uma delas.

2.2.2.2 RISC

Um processador RISC reconhece um número limitado de instruções, sendo que em contrapartida, estas instruções são otimizadas para que sejam executadas com mais rapidez.

A arquitetura RISC se caracteriza por possuir poucas instruções de máquina, que são executadas diretamente pelo *hardware*. Estas instruções, em sua maioria, não acessam a memória principal, trabalhando com registradores que neste tipo de processador se apresentam em grande número.

Portanto, a arquitetura RISC reduz o conjunto de instruções ao mínimo indispensável:

As instruções não contempladas (para execução) são executadas como combinações das instruções existentes. Com isso, consegue-se desempenho até 50 % a 75 % superior à de um processador CISC.



No Quadro 2.1, podemos observar as principais diferenças entre os processadores com arquiteturas RISC e CISC.

Quadro 2.1: Arquitetura RISC × arquitetura CISC

Arquitetura RISC	Arquitetura CISC
Poucas instruções	Muitas instruções
Instruções executadas pelo <i>hardware</i>	Instruções executadas por microcódigo
Instruções com formato fixo	Instruções com diversos formatos
Instruções utilizam poucos ciclos de máquina	Instruções utilizam múltiplos ciclos
Instruções com poucos modos de endereçamento	Instruções com diversos modos de endereçamento
Arquitetura com muitos registradores	Arquitetura com poucos registradores

Fonte: Adaptado de Maia, 2007

2.2.3 Memória principal

Um computador em seu interior possui vários tipos de memórias. A memória principal do computador é conhecida como memória RAM (*Random Access Memory*), ou memória de acesso aleatório. Ela é chamada de memória principal, pois é utilizada para manter os programas em execução, tanto dos usuários como do sistema operacional.

A memória é composta por unidades de acesso denominadas de células. Cada uma das células possui um determinado número de *bits*. O acesso ao conteúdo de uma célula se dá pela especificação de um número denominado de endereço. O endereço nada mais é do que uma referência única que fazemos a uma célula de memória. Quando um programa deseja fazer operações de gravação e leitura em uma célula, este deverá antes especificar o endereço de memória desejado, para então realizar a operação.

A quantidade de memória RAM disponível em um computador, exerce um grande efeito sobre o desempenho do mesmo, uma vez que sem memória RAM suficiente o sistema passa a utilizar a memória virtual, que é muito mais lenta que a memória RAM.

Uma questão importante a ser lembrada é que a memória RAM é volátil, ou seja, todo o conteúdo se apaga quando o computador é desligado ou reiniciado. Assim, caso tenhamos programas em memória e tivermos uma queda de energia elétrica, se não tivermos um dispositivo como o *no-break* (que garanta a estabilidade de energia), perderemos os dados que estão nesta memória.

Os *chips* de memória RAM são vendidos na forma de pentes de memória, com diferentes velocidades de funcionamento (atualmente na casa dos Gigahertz), tamanhos (01, 02, 04 GB, etc.) e tipos (DDR, DDR2, DDR3, etc.).

2.2.4 Memória cache

A memória *cache* pode ser caracterizada como uma memória de alta velocidade, volátil e na grande maioria das vezes com pequena capacidade de armazenamento. Sua principal função é diminuir a disparidade existente entre a velocidade na qual o processador executa instruções, bem como a velocidade com que os dados são lidos e gravados em memória principal.

O tempo de acesso a um dado presente na memória *cache* é muito menor que se o mesmo estivesse em memória principal.

A memória *cache* guarda os dados mais solicitados. Dessa forma, cada vez que o processador necessita de determinado dado ele verifica se esta informação encontra-se na memória *cache*, caso contrário, ele faz uma busca em memória principal (o que aumenta o tempo de acesso à informação).

A maioria dos processadores atuais apresentam um esquema de memória *cache* composto por múltiplos níveis. O funcionamento deste esquema tem por princípio que quanto menor a capacidade de armazenamento da memória *cache*, mais rápido é o acesso a um determinado dado.

Para aumentar o desempenho no funcionamento das memórias *caches* a hierarquização em múltiplos níveis torna-se necessária. Nesse esquema, o nível mais alto de *cache* é chamado de L1 (*Level 1*), com baixa capacidade de armazenamento e com alta velocidade de acesso. O nível L2 (*Level 2*) possui maior capacidade de armazenamento, porém velocidade de acesso inferior a L1.

2.2.5 Memória secundária

A memória secundária é utilizada para guardar dados de forma permanente no computador. Este tipo de armazenamento não necessita de alimentação, diferentemente da memória principal que necessita estar energizada para manter suas informações.

O acesso à leitura e gravação de dados na memória secundária é mais lento se comparado à memória principal, entretanto o custo é baixo e a capacidade de armazenamento é bastante superior.

Quanto aos tempos de acesso, a memória secundária funciona na ordem de milissegundos (10^{-3}), enquanto a memória principal trabalha na ordem de nanossegundos (10^{-9}). Podemos citar como exemplo de memória secundária os seguintes dispositivos:

- Discos magnéticos (discos rígidos – HD).
- Discos ópticos (CD, DVD, Blu-Ray).
- Memória *flash* (*pen drives*).

Na Figura 2.1, é possível visualizar um comparativo entre os diferentes tipos de dispositivos de armazenamento na relação custo, velocidade e capacidade de armazenamento.

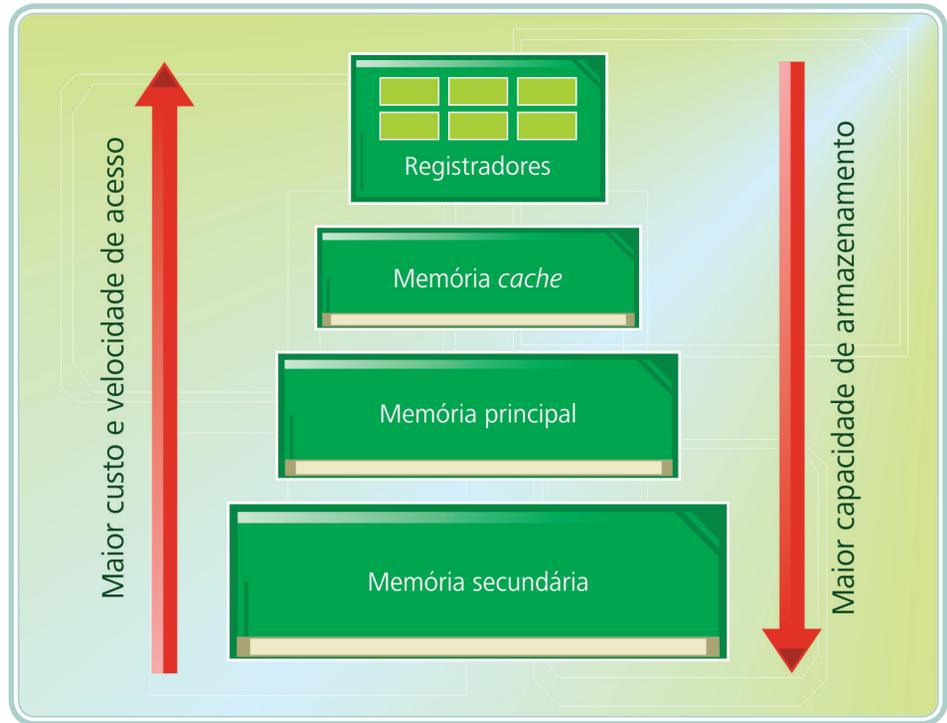


Figura 2.1: Relação entre dispositivos de armazenamento

Fonte: CTISM, adaptado de Maia, 2007

2.2.6 Dispositivos de entrada e saída

Os periféricos ou dispositivos de entrada e saída, são componentes de um computador que permitem expandir as funcionalidades do mesmo. Os mesmos podem ser divididos em duas categorias principais: os utilizados como memória secundária e os que servem como interface usuário-máquina. Veremos alguns exemplos de dispositivos de entrada e saída de dados.

2.2.6.1 Dispositivos de entrada

Os dispositivos de entrada ou periféricos de entrada têm a função de codificar os dados que entram no computador para que os mesmos possam ser processados pelo computador. São considerados dispositivos de entrada:

- Câmera digital.
- Câmera filmadora.
- Caneta ótica.
- *Drive* de Blu-Ray.
- *Drive* de CD/DVD-ROM.

- *Joystick*.
- Leitora de códigos de barra.
- Mesa gráfica.
- Microfone.
- *Mouse*.
- *Pen drive*.
- *Scanner*.
- Teclado.
- Tela sensível ao toque.

2.2.6.2 Dispositivos de saída

Os dispositivos de saída têm a função de decodificar os dados gerados pelos dispositivos de entrada, para que estes possam ser entendidos pelo usuário. Em outras palavras, os dispositivos de saída permitem extrair e visualizar informações que estão no computador.

Abaixo alguns exemplos de dispositivos de saída de dados:

- Caixas de som.
- *Drive* gravador de Blu-Ray.
- *Drive* gravador de CD-ROM/DVD-ROM.
- Impressora.
- Monitor de vídeo.
- *Pen drive*.
- *Plotter*.
- Projetores digitais.

Cabe salientar que existem dispositivos que funcionam tanto como periféricos de entrada como de saída, nestes casos sendo classificados como dispositivos de entrada/saída de dados.

Os dispositivos de entrada e saída são responsáveis pela interação da máquina com o homem. É por meio deles que os dados entram e saem do computador.

2.3 Software

O *software* é responsável por servir como interface entre as necessidades dos usuários e os dispositivos de *hardware*. Nos sistemas operacionais encontramos uma diversidade de *softwares* com funções diferenciadas, mas que exercem um papel importante. Veremos agora alguns dos principais utilitários aos sistemas operacionais.

2.3.1 Tradutor

Os primeiros programas de computador foram desenvolvidos através de uma linguagem denominada de “linguagem de máquina”. Além de exigir um conhecimento extremo por parte do programador era necessário também conhecer em detalhes o *hardware* do computador, como os endereços de memória onde seriam armazenados os programas.

Este cenário mudou quando começaram a surgir as primeiras linguagens de montagem ou assembly e as linguagens de alto nível. Estas linguagens facilitaram a construção de programas, documentações e manutenção dos mesmos.

Apesar das melhorias e facilidades proporcionadas por estas linguagens, o código fonte gerado por estes utilitários, não permitiam a execução direta pelo processador. Para isso, estes códigos precisavam da conversão de sua representação simbólica para a linguagem de máquina. Esta conversão era realizada por um utilitário chamado de tradutor.

O módulo gerado pelo tradutor é chamado de módulo-objeto. Dependendo do tipo de programa-fonte, existem dois tipos diferentes de tradutores que geram módulos-objeto: montador e compilador.

O montador é o utilitário capaz de traduzir um programa-fonte em linguagem de montagem para um programa-objeto não executável (módulo-objeto).

O compilador é o utilitário que tem a função de gerar (dado um programa escrito em linguagem de alto nível) um programa em linguagem de máquina, não executável. Como exemplos de linguagens de alto nível estão: C, Pascal, Cobol, Delphi, Java, entre outros. Os programas fonte, desenvolvidos nestas linguagens podem ser portados entre computadores de diferentes fabricantes, desde que existam padrões para a sintaxe da linguagem, permitindo assim o desenvolvimento de aplicações independentes do equipamento. Na Figura 2.2, é apresentado um resumo quando as funções de um tradutor, montador e compilador.

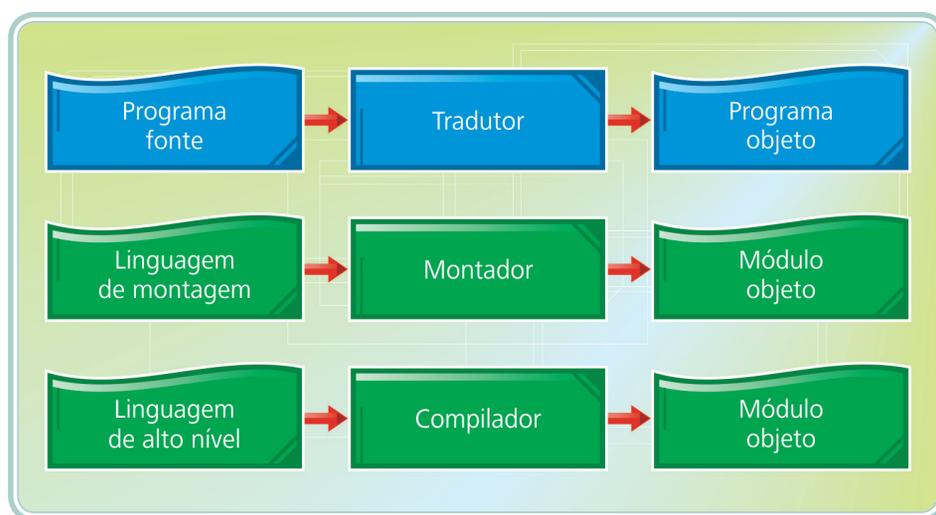


Figura 2.2: Tradutor, montador e compilador

Fonte: CTISM, adaptado de Maia, 2007

2.3.2 Interpretador

A função de um interpretador de código pode ser entendida como um tradutor que não gera módulo-objeto. Durante a execução de um programa, escrito em uma linguagem de alto nível, o interpretador traduz cada instrução e a executa imediatamente. São exemplos de linguagens de alto nível do tipo interpretadas: PHP, Basic, Perl, entre outras.

Nas linguagens interpretadas não existe a geração de um código-objeto, uma vez que o código fonte escrito pelo desenvolvedor já é o próprio código a ser interpretado. A desvantagem de tal método é o tempo gasto na tradução das instruções de um programa toda a vez que este for executado.

2.3.3 Linker

O *linker* ou editor de ligação tem por objetivo gerar, dado um ou mais módulos-objetos, um único programa executável. Entre suas funções básicas estão a resolução das referências simbólicas existentes entre os módulos e a reserva de memória para a execução do programa.

O *linker* também é responsável pela tarefa de relocação (determinar a posição de memória na qual o programa será carregado para execução).

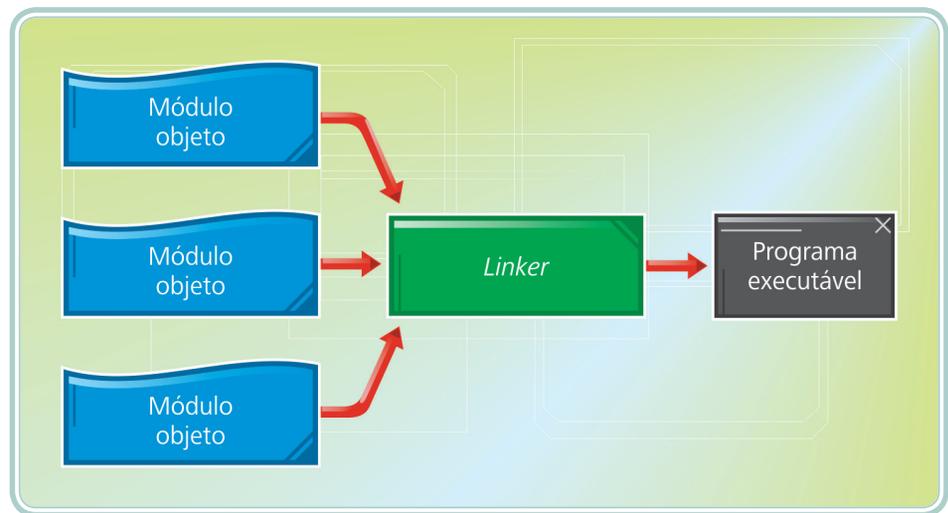


Figura 2.3: Função do linker

Fonte: CTISM, adaptado de Maia, 2007

2.3.4 Loader

Um *loader* tem a função, junto ao sistema operacional, de carregar na memória principal um programa a fim de que o mesmo possa ser executado. O *loader* pode ser classificado como do tipo absoluto ou relocável, dependendo do procedimento de carga utilizado.

Para código executável do tipo absoluto, o *loader* necessita saber o endereço de memória inicial e o tamanho do módulo para realizar o carregamento, dando início ao processo de transferência do programa da memória secundária para a memória principal.

Quanto ao código relocável, o programa pode ser carregado em qualquer posição de memória, sendo o *loader* responsável pela relocação no momento do carregamento.

2.3.5 Depurador

O desenvolvimento de programas é uma tarefa que exige além de raciocínio lógico apurado um bom conhecimento da linguagem de programação que se está utilizando na construção de um determinado programa. Porém, nenhum destes itens garante que não podemos ter problemas quanto a erros de sintaxe e lógica na escrita de nossos códigos.

Para auxiliar nesta tarefa de identificação e correção de erros, temos um utilitário depurador ou *debugger*. O depurador permite ao programador acompanhar

toda a execução de um programa (do início ao fim) permitindo detectar erros em sua lógica. Este utilitário oferece recursos, tais como:

- Acompanhar a execução de um programa, linha a linha de código.
- Permitir a alteração e visualização do conteúdo de variáveis.
- Definir pontos de parada dentro do programa (conhecidos como *break-points*), permitindo que durante a execução do programa o mesmo pare nestes pontos.

Resumo

Nessa aula, foi possível entender um pouco mais sobre os principais conceitos relacionados a *hardware* e *software* e como estes são importantes em um projeto de sistemas operacionais. É justamente a ligação entre o *hardware* e o *software* que permite aos sistemas operacionais oferecer uma grande quantidade de opções de recursos aos usuários. Em nossa próxima aula, falaremos sobre a estrutura do sistema operacional e seus principais recursos.

Atividades de aprendizagem

1. Um sistema computacional é formado em sua essência por três subsistemas básicos. Cite e explique a função deles junto ao SO.
2. Descreva a diferença e as características entre os tipos de memória do tipo registradores, *cache*, RAM e secundária.
3. Cite três dispositivos de entrada e três de saída que você conhece, descrevendo as características principais de cada um deles.
4. Diferencie: tradutor, interpretador, compilador, *linker*, *loader* e depurador.



Aula 3 – Estrutura do sistema operacional

Objetivos

Conhecer as principais funções que compõem um sistema operacional.

Entender a estrutura de um sistema operacional.

Ter um entendimento das principais técnicas utilizadas nos sistemas operacionais.

3.1 Considerações iniciais

Nessa aula, conheceremos a estrutura que compõe um sistema operacional em detalhes. Ao longo do texto você vai entender as funções do núcleo do sistema, os modos de acesso ao *kernel*, às rotinas do sistema operacional e as chamadas do sistema, além dos conceitos de interrupções, exceções, DMA, *buffering*, *spooling* e reentrância de dados.

3.2 Funções do núcleo

Compreender a estrutura de um sistema operacional, bem como, seu funcionamento não é uma tarefa trivial, uma vez que as rotinas do sistema operacional são executadas concorrentemente sem uma ordem pré-definida, com base em eventos assíncronos. Muitos desses eventos referem-se ao *hardware* e tarefas internas do próprio sistema operacional.

As principais funções do núcleo do sistema operacional são:

- Tratamento de interrupções e exceções.
- Criação e eliminação de processos.
- Sincronização e comunicação entre processos.
- Escalonamento e controle de processos.
- Gerência de memória.

- Gerência do sistema de arquivos.
- Gerência dos dispositivos de entrada e saída.
- Suporte a redes locais de computadores.
- Contabilização do uso do sistema.
- Auditoria e segurança do sistema.

3.3 Modo de acesso

Um problema recorrente nos projetos de sistemas operacionais, é a implementação de mecanismos de segurança que garantam o funcionamento e proteção ao núcleo do sistema e seus serviços. Caso uma aplicação qualquer possua acesso ao núcleo do sistema e realize uma alteração que afete sua integridade, todo o sistema operacional poderá ficar comprometido.

Para garantir a operação apropriada, é necessário proteger o sistema operacional e todos os outros programas e dados de qualquer programa que esteja com erro. A proteção é necessária para qualquer recurso compartilhado. Assim, são necessárias duas modalidades especiais de operação:

- A modalidade de usuário.
- A modalidade de monitor (também chamada de modalidade supervisor, do sistema ou privilegiada).

Um *bit*, chamado *bit* de modalidade, é adicionado ao *hardware* do computador para indicar a modalidade corrente:

- Monitor (0).
- Usuário (1).

Dessa forma, podemos identificar se uma tarefa é do sistema operacional ou do usuário em questão.

3.4 Rotinas do sistema operacional e *system calls*

As rotinas do sistema operacional são parte integrante do núcleo do sistema e oferecem serviços aos usuários e suas aplicações. Todas as funções pertencentes ao núcleo são desempenhadas através de rotinas do sistema, com códigos de instruções privilegiadas. Partindo desta condição, para que as rotinas possam ser executadas a CPU deve estar obrigatoriamente em modo *kernel*, o que necessita a implementação de mecanismos de proteção que garantam a confiabilidade do sistema.

O controle de execução de rotinas do SO realiza-se pelas *system calls*, ou chamadas do sistema. Toda a vez que uma aplicação deseja chamar uma rotina do sistema operacional o *system call* (mecanismo) é ativado. Caso a aplicação possua os privilégios necessários para chamar a rotina do sistema desejada, o sistema operacional iniciará uma série de passos a serem realizados, conforme podemos visualizar na Figura 3.1.

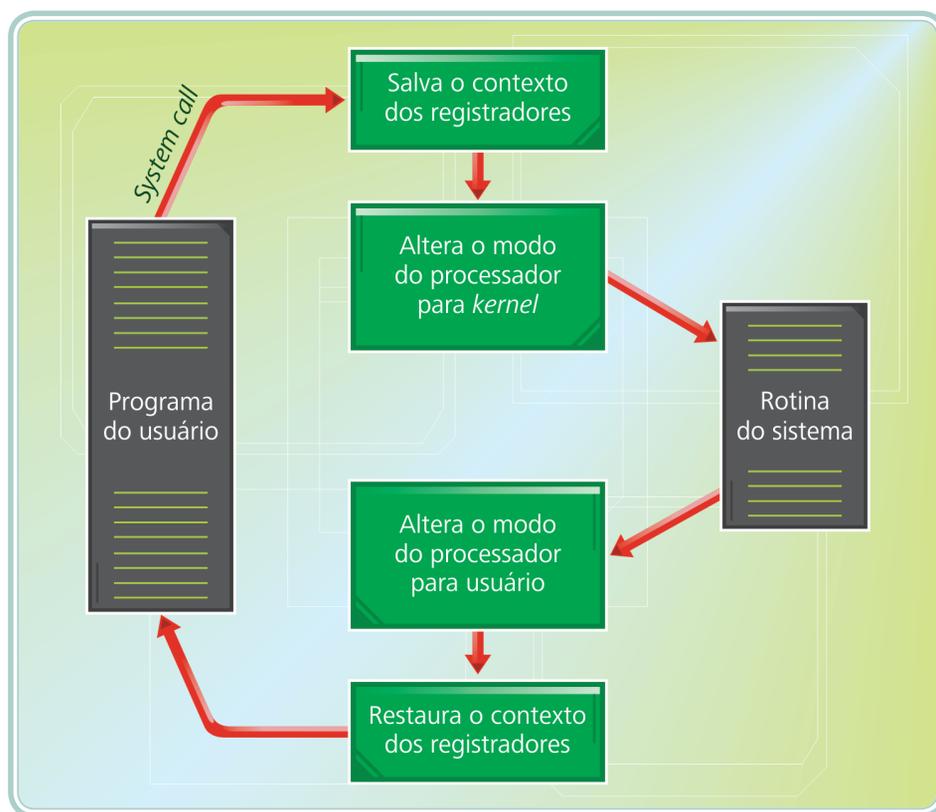


Figura 3.1: Chamada a uma rotina do sistema

Fonte: CTISM, adaptado de Maia, 2007

Ao chamar uma rotina do sistema, o sistema operacional salva o conteúdo atual dos registradores, realiza a troca do modo de acesso da CPU de usuário

para *kernel* e faz o desvio para a rotina modificando o registrador PC com o endereço da rotina chamada. Ao término da execução da rotina junto ao sistema, o modo de acesso é modificado de *kernel* para usuário e o contexto dos registradores é restaurado para que a aplicação possa continuar a execução a partir da instrução que chamou a rotina do sistema.

Os mecanismos de *system call* e de proteção por *hardware* garantem a segurança e a integridade do sistema. Dessa forma, as aplicações ficam impedidas de executarem instruções privilegiadas sem a devida autorização e a supervisão do sistema operacional.

3.5 Operação do sistema de computação

Conhecer a estrutura dos sistemas de computação é item primordial antes que possamos explorar os detalhes de seu funcionamento. Os itens básicos contemplam:

- Inicialização do sistema.
- Entrada e saída (*input/output – I/O*).
- Armazenamento.

Um moderno sistema de computação de uso geral compõe-se basicamente de: um processador (CPU), controladores de dispositivos e um *bus* comum. Processadores e controladores de dispositivos podem operar concorrentemente, competindo por ciclos de memória. Para que um computador possa entrar em funcionamento (ligar) ou reiniciar é preciso basicamente de:

- Um programa *bootstrap* (responsável por inicializar todos os aspectos do sistema, como registradores, controladores, entre outros. Este programa é armazenado na memória ROM do tipo *firmware*).
- Ocorrência de eventos (interrupção proveniente tanto do *hardware* como do *software*).

O sistema operacional espera que interrupções ocorram para a execução de processos, dispositivos de I/O para servir e a usuários a quem responder.

3.6 Interrupções e exceções

Enquanto um programa é executado, podem ocorrer alguns eventos inesperados, ocasionando um desvio forçado do seu fluxo de execução. A estes eventos, denomina-se de interrupção ou exceção. Tanto a interrupção como a exceção podem ocorrer em consequência da sinalização de algum dispositivo de *hardware* externo à CPU ou da execução de instruções do próprio programa.

Uma interrupção é sempre causada por um evento externo ao programa que está em execução, ou seja, independe da instrução que está sendo executada. Um exemplo de interrupção é quando um dispositivo de *hardware* avisa ao processador que alguma operação de E/S está completa. Neste caso, a CPU deve interromper suas atividades atuais no programa em execução para tratar o término da operação de E/S. A Figura 3.2, demonstra o mecanismo de interrupção e exceção de forma ilustrativa.

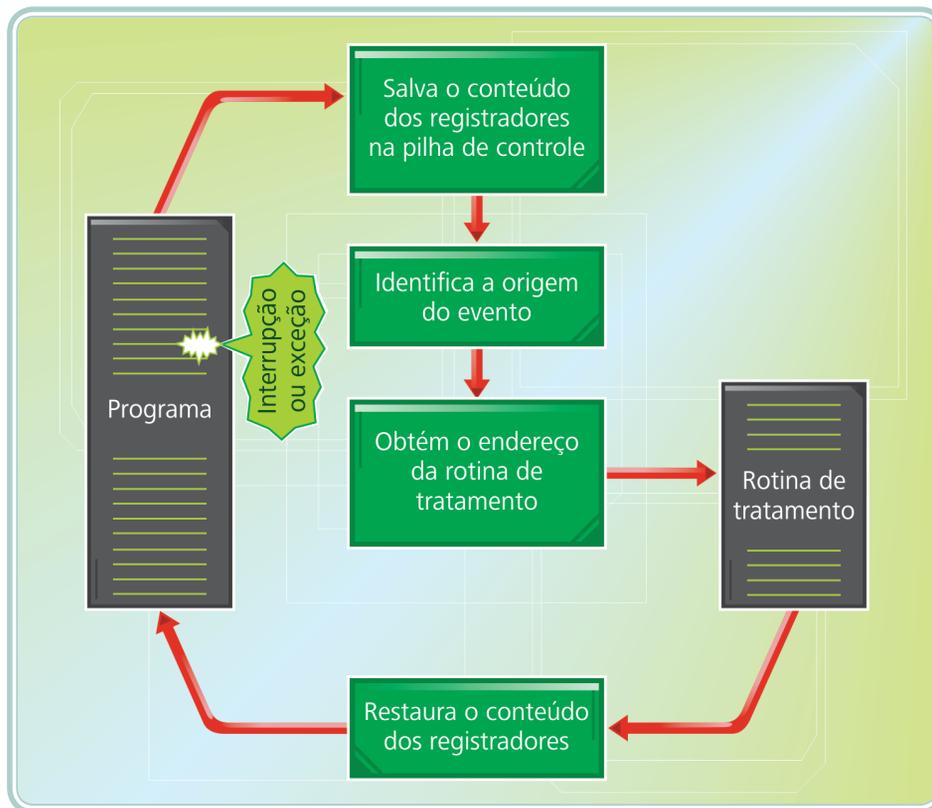


Figura 3.2: Mecanismo de interrupção e exceção

Fonte: CTISM, adaptado de Maia, 2007

As interrupções provêm de eventos assíncronos, ou seja, não estão relacionadas à instrução do programa corrente. Isto permite a ocorrência de múltiplas interrupções simultâneas. Uma interrupção é um evento tratável pelo sistema operacional.

Uma exceção é bastante semelhante a uma interrupção, sendo o que diferencia uma da outra é o motivo pelo qual o evento é gerado. Uma exceção pode ocorrer através de uma instrução do próprio *software*, como por exemplo, a divisão de um número por zero, entre outros. A exceção diferentemente da interrupção é gerada por um evento síncrono (resultado direto da execução do programa corrente).

Na grande maioria das vezes, uma exceção provoca um erro fatal no sistema, causando o término anormal do programa. Recomenda-se que a exceção deve ser tratada dentro do próprio programa, com instruções escritas pelo programador.

Da mesma forma que na interrupção, sempre que uma exceção é gerada o programa em execução é interrompido e o controle é desviado para uma rotina de tratamento de exceção. Para cada tipo de exceção existe uma rotina de tratamento adequada, para a qual o fluxo de execução deve ser desviado.

3.7 Estrutura de acesso direto à memória (DMA)

Para entendermos como funciona o Acesso Direto à Memória (*Direct Access Memory – DMA*), vamos imaginar o seguinte cenário, que ocorre nos sistemas operacionais atuais.

Os processos denominados “lentos” têm uma prioridade baixa de interrupção para entrada e saída assíncronos, como por exemplo, a leitura de caracteres de um terminal, permitindo assim que outras interrupções mais importantes sejam processadas em primeiro lugar. Entretanto, um dispositivo de alta velocidade como disco ou uma rede de comunicação é capaz de transmitir informações próximas às velocidades das memórias.

Nestes casos, o sistema operacional faz uso do acesso direto à memória para dispositivos de entrada e saída de alta velocidade. Nesta técnica, o controlador de dispositivos transfere um bloco inteiro de dados diretamente da memória para seu próprio *buffer* ou a partir dele para a memória, sem qualquer intervenção da CPU. Assim, somente uma interrupção é gerada por bloco, em vez de uma interrupção por *byte* (ou palavra) gerada por dispositivos de baixa velocidade.

3.8 Buffering

A técnica denominada de *buffering* consiste em utilizar uma área de memória principal, chamada *buffer*, criada e mantida pelo sistema operacional. Possui a finalidade de auxiliar a transferência de dados entre dispositivos de E/S e a memória. O *buffer* permite minimizar a disparidade de velocidade entre o processador e os dispositivos de E/S e, tem como objetivo principal, manter tanto os dispositivos de E/S como o processador, ocupados a maior parte do tempo.

O registro é a unidade de transferência do mecanismo de *buffering*. O *buffer* deve comportar o armazenamento de diversos registros, de forma que o processador tenha à sua disposição dados suficientes para processar sem ter que interromper o programa a cada leitura/gravação no dispositivo de E/S. Na Figura 3.3, é possível visualizar como as operações de entrada e saída utilizam o *buffer*.

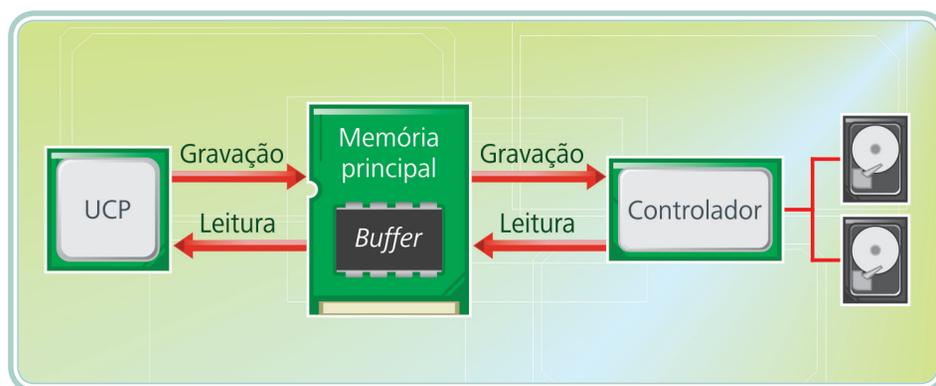


Figura 3.3: Operação de entrada e saída utilizando *buffer*

Fonte: CTISM, adaptado de Maia, 2007

3.9 Spooling

A técnica de *spooling* foi criada inicialmente para auxiliar a submissão de processos ao sistema, sendo os processos gravados em fita para posterior leitura e execução. Com o aparecimento dos terminais para acesso ao sistema, esta técnica teve sua função adaptada para armazenar o resultado da impressão dos programas em execução.

Isto é possível através da criação e manutenção, pelo sistema operacional de uma grande área em disco, com a finalidade de simular uma impressora. Assim, os usuários e seus programas imprimem, na verdade, para este arquivo em disco, liberando a associação dos dispositivos de impressão diretamente aos programas que estão executando.

A Figura 3.4 demonstra a técnica de *spooling* de modo simplificado.

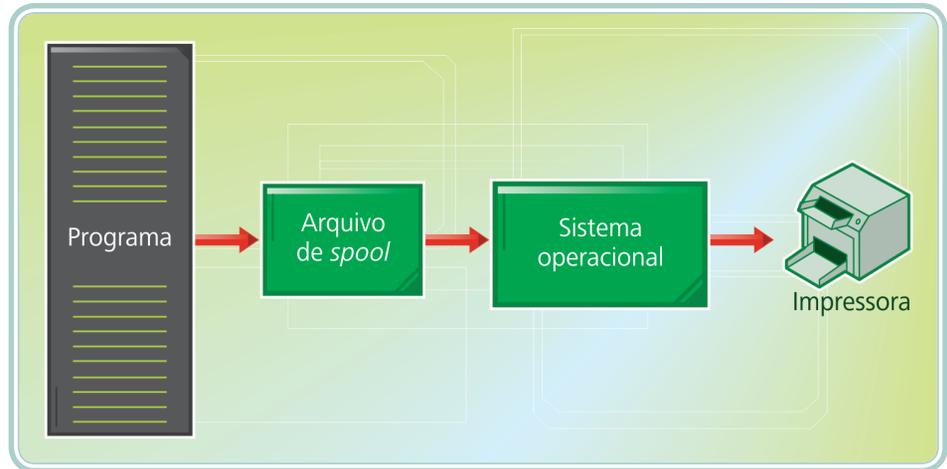


Figura 3.4: Técnica de *spooling*

Fonte: CTISM, adaptado de Maia, 2007

3.10 Reentrância

Em sistemas multiprogramáveis, é normal que vários usuários utilizem os mesmos aplicativos simultaneamente, como editores de texto, compiladores, entre outros utilitários.

Nestes casos, se cada usuário que fosse utilizar um destes aplicativos trouxesse o código executável para a memória haveria então diversas cópias de um mesmo programa ocupando espaço na memória, o que causaria um grande desperdício de espaço.

A reentrância é a capacidade de um código executável (código reentrante) ser compartilhado por vários usuários, exigindo apenas uma cópia do programa em memória. Esta técnica permite que cada usuário esteja executando um trecho diferente do código reentrante, manipulando dados próprios, exclusivos de cada usuário.

Assim, a reentrância tem como objetivo geral promover o uso mais eficiente da memória e um desempenho maior do sistema. A Figura 3.5, ilustra o cenário da reentrância.

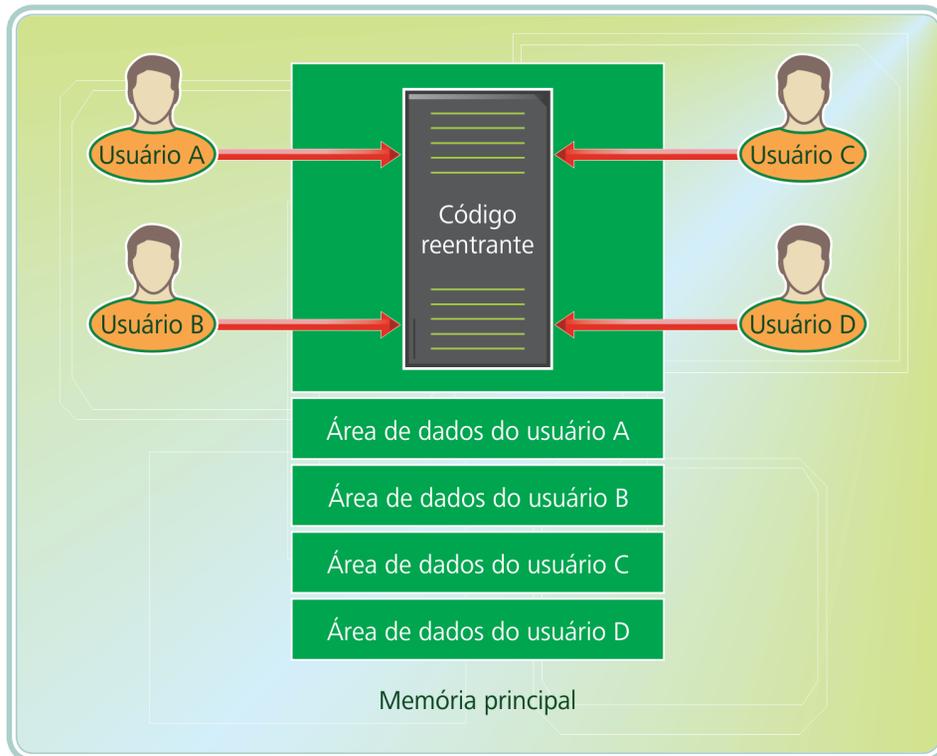


Figura 3.5: Reentrância

Fonte: CTISM, adaptado de Maia, 2007

Resumo

Nessa aula, vimos em detalhes os principais recursos e funções que dão sustentação aos sistemas operacionais. A estrutura do sistema operacional é um capítulo crucial para um bom entendimento de como as ações são realizadas internamente no sistema. Para fixar o conteúdo visto em cada aula, é importante que você realize os exercícios de aprendizagem. Em nossa próxima aula, falaremos sobre o gerenciamento de processos no sistema operacional.

Atividades de aprendizagem

1. Cite cinco funções do núcleo do sistema operacional.
2. Quais são os modos de acesso (interno) ao sistema operacional existente e por que eles são fundamentais ao bom funcionamento do sistema operacional?
3. Quais são os itens básicos para operação de um sistema computacional?
4. Cite e conceitue: DMA, *buffering*, *spooling* e reentrância.



Aula 4 – Gerenciamento de processos

Objetivos

Entender o conceito de processos e sua função junto ao sistema operacional.

Conhecer seu ciclo de vida e atividades que desempenham.

Mostrar os recursos de *hardware* e *software* associados aos processos.

4.1 Considerações iniciais

O gerenciamento de processos é parte fundamental para o entendimento dos sistemas operacionais em sua essência. Estudaremos nessa aula o que são os processos, como funcionam, como são gerenciados pelo sistema operacional junto aos recursos de *hardware*, juntamente com o seu ciclo de vida junto ao sistema.

4.2 O que é um processo?

Podemos definir um processo em sistemas operacionais como um programa em execução. Mas, não podemos confundir processo com um programa (*software*) instalado em nosso computador. Um programa é uma entidade passiva, assim como o conteúdo armazenado no disco rígido de nossos computadores. Enquanto que um processo é uma entidade ativa, ou seja, algo que entrou em execução e que possui um conjunto de recursos associados a ele. Um sistema operacional é constituído, portanto, por uma coleção de processos, que se subdividem em processos do sistema operacional e processos de usuário.

4.3 Estrutura do processo

Considerando os sistemas multiusuários, cada usuário do sistema tem seu programa associado a um processo. Nestes sistemas, o usuário ao executar seu programa tem a impressão que todos os recursos do computador estão disponíveis somente àquele programa. De fato, isto não é verdade, uma vez que todos os recursos estão sendo compartilhados inclusive o processador.

Para dar conta das tarefas associadas a um processo, o processador executa o programa de usuário durante um intervalo de tempo estabelecido, mudando de processo para atender aos demais.

Um processo é formado em sua essência por três partes, sendo eles: contexto de *hardware*, contexto de *software* e espaço de endereçamento, conforme pode ser visualizado na Figura 4.1. Estas três partes são responsáveis por manter todas as informações necessárias à execução de um programa.

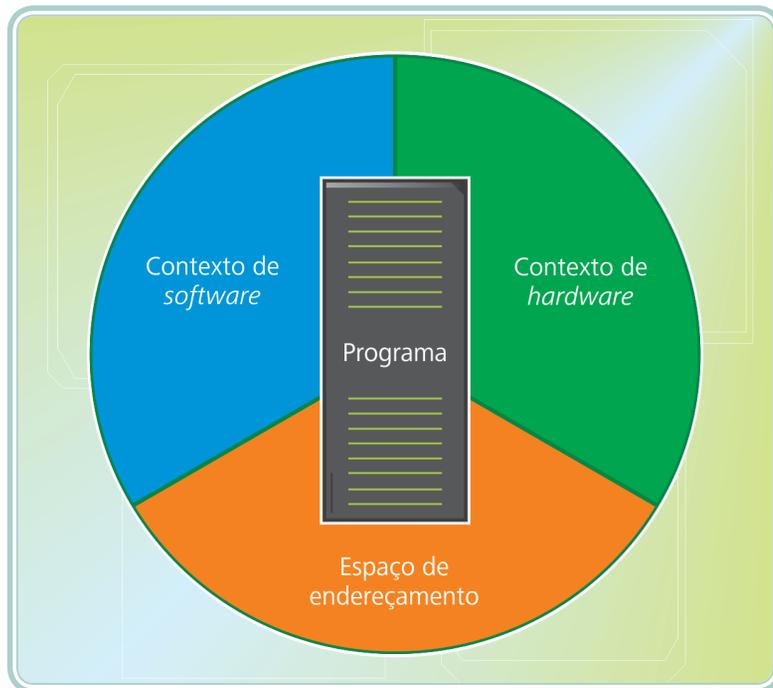


Figura 4.1: Estrutura de um processo
Fonte: CTISM, adaptado de Maia, 2007

4.3.1 Contexto de *hardware*

O contexto de *hardware*, relativo a um processo, armazena informações sobre: registradores gerais do processador, registradores de uso específico como o contador de programa (PC – *Program Counter*), *Stack Pointer* (SP) e registrador de *status*.

Estes são fundamentais para a implementação de sistemas multiprogramáveis, onde os processos precisam alternar na utilização do processador, podendo ter sua execução interrompida (mudança de contexto) e posteriormente restaurada.

4.3.2 Contexto de *software*

O contexto de *software* de um processo é composto por três grupos de informações, que são: identificação, quotas e privilégios.

4.3.2.1 Identificação

Para que um processo seja identificado junto ao sistema operacional utiliza-se um número, chamado de identificação do processo (PID – *Process Identification*). É através deste número que o sistema operacional e outros processos podem fazer referência a este processo. Além do PID, alguns sistemas fazem a identificação de um processo pelo seu nome.

4.3.2.2 Quotas

Representam os limites de cada recurso existente no sistema que um processo pode alocar. São exemplos de quotas existentes nos principais sistemas operacionais usuais: número máximo de arquivos abertos simultaneamente; tamanho máximo de memória que o processo pode alocar; número máximo de operações de entrada e saída; número máximo de processos, subprocessos e *threads* que podem ser criados.

4.3.2.3 Privilégios

Os privilégios definem basicamente o que um processo pode fazer frente ao sistema operacional, aos demais processos ou a ele mesmo. Estes privilégios variam baseados onde serão aplicados. Em geral, nos sistemas operacionais os privilégios estão associados a uma conta que fornece poderes totais para que um usuário específico possa fazer as alterações que necessitar. Este usuário com privilégios totais do sistema é chamado de “*root*” nos sistemas operacionais Linux e “administrador” nos sistemas operacionais Windows.

Na Figura 4.2, é apresentada uma ilustração que resume as características da estrutura de um processo, em cada uma das três divisões principais: contexto de *hardware*, contexto de *software* e espaço de endereçamento.



Figura 4.2: Características da estrutura de um processo

Fonte: CTISM, adaptado de Maia, 2007

4.3.3 Estados do processo

Um processo muda de estado durante a sua execução. Assim o estado de um processo é definido em parte pela sua atividade em curso. Cada processo pode estar em cada um dos seguintes estados:

- **Novo** – o processo está sendo criado.
- **Execução** – as instruções estão sendo executadas.
- **Espera** – o processo está esperando pela ocorrência de algum evento (como um término de I/O).
- **Pronto** – o processo está esperando para ser designado a um processador.
- **Terminado** – o processo terminou a sua execução.

4.3.4 Bloco de controle de processo

Cada processo presente no sistema operacional é representado por um bloco de controle de processo. Este contém muitas informações associadas ao processo específico, incluindo as seguintes:

- Estado do processo.
- Nome do processo.
- Prioridade do processo.
- Contador de programa.
- Registradores da CPU.
- Informação de gerenciamento da memória.
- Informação de contabilização.
- Informação de estado de I/O.

O PCB (Bloco de Controle de Processo) tem a função de repositório para qualquer informação que possa variar de um processo para outro.

Os PCBs de todos os processos ativos no sistema operacional residem na memória principal em uma área exclusiva para os mesmos. O tamanho desta área, na maioria das vezes é limitado por um parâmetro do sistema operacional, que define a quantidade máxima de processos que podem ser suportados de forma simultânea pelo sistema.

A gerência dos processos junto ao sistema operacional é realizada através de chamadas às rotinas do sistema, que realizam operações como: criação de processos, alteração, visualização, eliminação, sincronização, suspensão, entre outras.

Resumo

Nessa aula, vimos o que é um processo, sua estrutura, os contextos de *hardware* e *software* associados aos processos, os estados que o mesmo pode estar durante seu ciclo de vida, bem como, o bloco de controle de processos e suas



É possível visualizar os processos em execução no sistema operacional Windows, através do *Prompt* de comando, digitando o comando "tasklist". No sistema operacional Linux é possível visualizar os processos atuais através do Terminal de comandos, digitando o comando "ps-aux".

informações. Para fixar o conteúdo visto em cada aula, é importante que você realize os exercícios de aprendizagem. Em nossa próxima aula, falaremos sobre o gerenciamento de memória.



Atividades de aprendizagem

1. Cite com suas palavras o que é um processo e sua importância para o sistema operacional.
2. Considerando a estrutura de um processo, diferencie o contexto de *hardware* e o contexto de *software*.
3. Cite e explique cada um dos estados de um processo.
4. Qual a função do PCB e porquê ele é fundamental junto aos processos?
5. Faça uma pesquisa na internet, sobre como podemos visualizar os processos correntes nos sistemas operacionais Windows e Linux e quais comandos podem ser utilizados para parar um processo e eliminá-lo do sistema.

Aula 5 – Gerenciamento de memória

Objetivos

Entender como o sistema operacional gerencia a memória.

Conhecer as principais técnicas para alocação de memória.

Visualizar a importância da utilização da memória nos sistemas computacionais.

5.1 Considerações iniciais

A memória principal existente nos computadores, sempre foi vista como um recurso caro e escasso. Apesar deste conceito ter mudado nos últimos anos, utilizar a memória principal de maneira mais otimizada possível é fator fundamental para o sucesso dos sistemas operacionais. Abordaremos nessa aula as principais técnicas para gerenciamento de memória pelos sistemas operacionais atuais.

5.2 Funções básicas da memória

A memória principal do computador tem como função, geral e generalista, alocar os programas que estão sendo executados pelo sistema operacional em determinado momento. Na memória principal residem (por certo período de tempo) os processos que estão em execução, tanto do sistema operacional (necessários para manter o sistema operacional em funcionamento) quanto dos usuários (programas em execução pelo usuário).

Podemos caracterizar a memória principal, com as seguintes características:

- Recurso caro e escasso (apesar de ter seu valor bem mais acessível do que a tempos atrás).
- Programas só executam se estiverem na memória principal.
- Quanto mais processos estiverem na memória principal, melhor será o compartilhamento do processador.

- Necessidade de uso otimizado.
- O sistema operacional não deve ocupar muita memória.
- A gerência de memória caracteriza-se como um dos fatores mais importantes em um projeto de sistemas operacionais.

De uma forma geral, os programas são armazenados na memória secundária, para que possam ser acessados sempre que necessário. Entende-se como memória secundária os discos rígidos, *pen drives*, CDs, DVDs, entre outros. Estes dispositivos possuem geralmente um bom espaço de armazenamento e preços reduzidos se comparados a memória principal do computador, além de armazenar os dados de forma não-volátil, ou seja, estes dados não se apagam depois que o computador é desligado.

Entretanto, o processador somente executa programas (instruções) que estiverem na memória principal, sendo necessário transferir os programas da memória secundária para a memória principal, antes de serem executados.

Considerando que o tempo de acesso à memória secundária é muito maior que à memória principal, o sistema operacional buscará reduzir o número de operações de entrada e saída à memória secundária, mantendo na memória principal o maior número possível, evitando problemas de desempenho do sistema.

Mesmo, muitas vezes, sem ter espaço livre em memória principal, o sistema operacional deve prover que novos processos sejam aceitos e executados normalmente. Isto é possível através da transferência de programas em memória principal para a memória secundária, liberando, neste caso, espaço para que novos processos sejam executados. Para este procedimento denominamos de **swapping**.

Outro problema relacionado à memória que está no contexto da gerência de memória está em permitir a execução de programas que sejam maiores que a memória física disponível, utilizando para isso técnicas como o **overlay** e a **memória virtual**.

Veremos, a partir de agora, algumas técnicas utilizadas pelo sistema operacional para gerenciamento de memória, confira!

5.3 Técnicas de gerenciamento de memória

Por ser tratada como fundamental a um sistema operacional, a utilização de técnicas de gerenciamento de memória, torna-se necessária em ambientes multiprogramáveis, permitindo que as necessidades dos programas sejam atendidas pelo sistema operacional, em um tempo estimado, sem comprometer o desempenho, segurança e permitindo o compartilhamento de recursos do sistema. Conheceremos, agora, as principais técnicas de gerenciamento de memória e como funcionam junto ao SO.

5.3.1 Alocação contígua simples

Este tipo de alocação era implementada nos primeiros sistemas operacionais e ainda é encontrada em sistemas monoprogramáveis. Esta técnica consiste em dividir a memória principal em duas áreas:

- Área do sistema operacional.
- Área do usuário.

Nesta técnica, um usuário não pode usar uma área maior do que a disponível (para execução de seus programas).

Um dos grandes problemas da alocação contígua simples é que um programa de usuário pode acessar a área do sistema operacional, ocasionando diversos problemas (uma vez que o usuário tem controle total sobre a memória, podendo ter acesso a qualquer posição). Para evitar tal problema, alguns sistemas implementam proteção através de registradores que limitam as áreas do sistema operacional e do usuário. Assim, cada vez que um usuário precisa, através de seus programas, acessar uma área de memória, o sistema operacional faz a consulta para saber se os espaços disponíveis (endereços de memória) estão dentro dos limites permitidos. A Figura 5.1 mostra um exemplo de alocação contígua simples com o uso de registradores, confira.

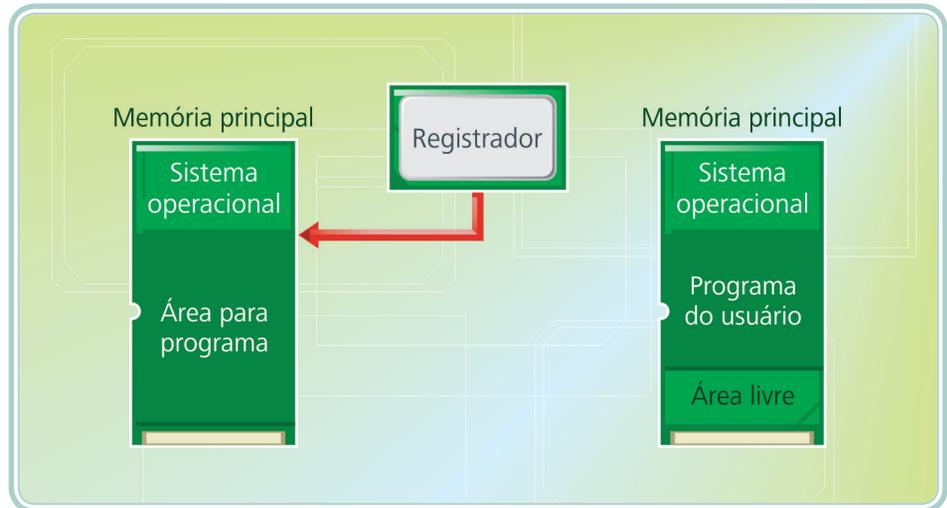


Figura 5.1: Alocação contígua simples

Fonte: CTISM, adaptado de Maia, 2007

Outro agravante desta técnica está no fato de que ela não permite uma utilização eficiente dos recursos computacionais, uma vez que somente um usuário pode dispor dos recursos. Se o programa do mesmo não preencher totalmente o espaço existente em memória, haverá um espaço livre sem utilização.

5.3.1.1 Técnica de *overlay*

A técnica de *overlay* visa resolver (em partes) o problema ocasionado pelo desperdício de espaço em memória da técnica anterior (alocação contígua de memória). Para resolver tal problema a técnica de *overlay*, propõe o compartilhamento de áreas de memória livres, através de programas independentes, fazendo com que permaneça na memória principal somente o módulo principal de um programa.

Tomamos como exemplo o seguinte cenário: imagine um programa composto por três módulos (um principal, um de cadastro de clientes e outro de impressão). Considere neste exemplo que os módulos de cadastro de clientes e impressão são independentes. Isso significa dizer que quando um dos módulos está em memória, o outro não precisa necessariamente estar lá. Já o módulo principal deve permanecer em memória o tempo todo, pois é comum aos dois módulos do sistema. A técnica de *overlay* neste caso utiliza uma única área de memória onde os módulos independentes podem compartilhar deste mesmo espaço. A Figura 5.2, mostra um exemplo de tal técnica.

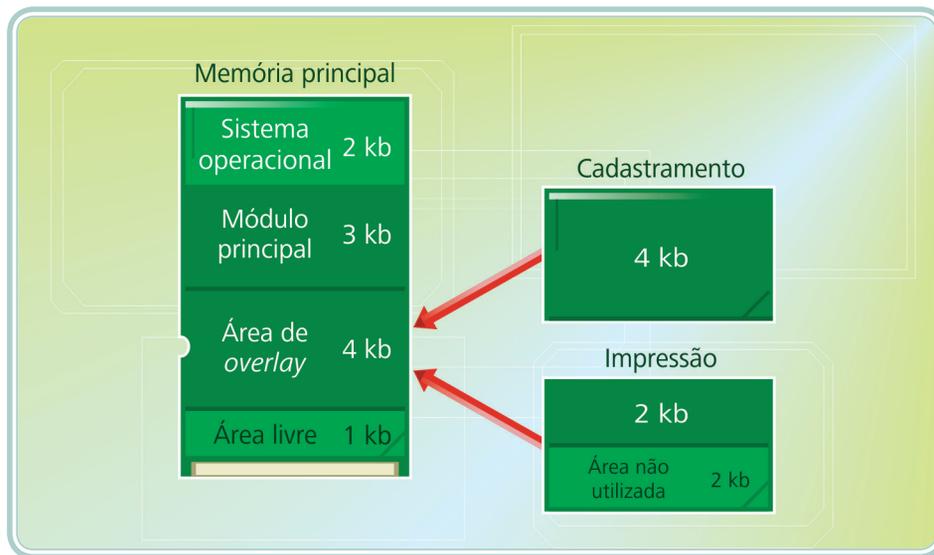


Figura 5.2: Técnica de *overlay*

Fonte: CTISM, adaptado de Maia, 2007

A vantagem da técnica de *overlay* está no fato de permitir ao programador aumentar os limites da memória principal.

5.3.2 Alocação particionada

Com a evolução dos sistemas operacionais (dos monoprogramáveis para os multiprogramáveis) tornou-se necessário um melhor aproveitamento dos recursos disponíveis. Tornar eficiente o uso do processador e garantir que o maior número possível de programas estivessem em memória, eram alguns dos desafios que novas técnicas de gerenciamento de memória deveriam prover.

5.3.2.1 Alocação particionada estática

A alocação particionada estática ou fixa, tinha como objetivo dividir a memória em pedaços de tamanho fixo, chamados de partições. O tamanho de cada partição era estabelecido na inicialização do sistema operacional e calculado segundo o tamanho dos programas que executariam no ambiente. Sempre que fosse necessária alguma alteração do particionamento previamente alocado, uma nova inicialização era realizada com uma nova configuração.

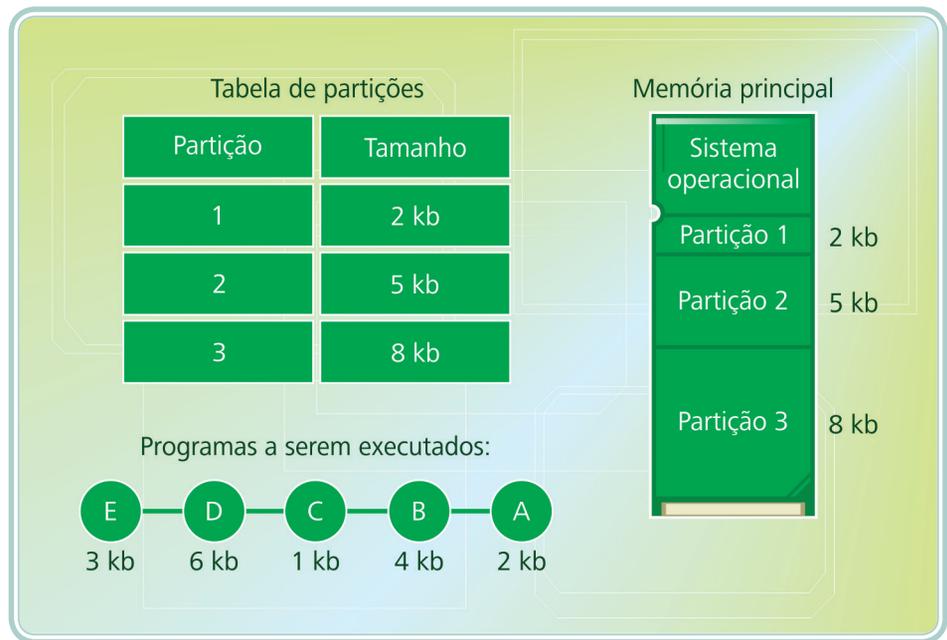


Figura 5.3: Alocação particionada estática

Fonte: CTISM, adaptado de Maia, 2007

Outra variante deste tipo de alocação chama-se alocação particionada estática absoluta. Esta técnica recebe este nome, pois os programas neste caso, somente podiam ser carregados e posteriormente executados em apenas uma partição específica, independente se outras partições estivessem disponíveis. No código absoluto (gerado pelos compiladores e montadores) as referências a endereços de programa apontam para posições físicas na memória principal, fazendo com que o programa só possa ser executado na posição em que foi destinado.

Uma evolução da alocação particionada estática absoluta chama-se **alocação particionada estática relocável**. Neste tipo de particionamento todas as referências a endereços no programa são referentes ao início do código e não a endereços físicos de memória, permitindo aos programas serem executados a partir de qualquer partição.

Um problema ocasionado pelo uso de partições fixas, como é o caso das alocações particionadas estática absoluta e relocável, é o fato de que ambas terão problema com a **fragmentação interna**, que nada mais é do que o espaço livre que sobra devido ao tamanho do programa ser menor que o tamanho disponível para sua execução, sendo este espaço livre desperdiçado.

5.3.2.2 Alocação particionada dinâmica

A alocação particionada estática mostrada anteriormente apresentava um problema fundamental que era a fragmentação interna, devido a sua forma de

alocação de espaço (de forma fixa). Era necessário então criar novas técnicas para esta alocação que permitisse um melhor aproveitamento destes espaços.

Pensando em sanar tal problema, a alocação particionada dinâmica surgiu com a ideia de eliminar o conceito de partições de tamanho fixo. Assim, cada programa utilizaria seu próprio espaço em memória, fazendo de sua dimensão o tamanho de sua partição. Neste esquema, como cada programa utilizaria sua própria dimensão, não teríamos mais o problema da fragmentação interna.

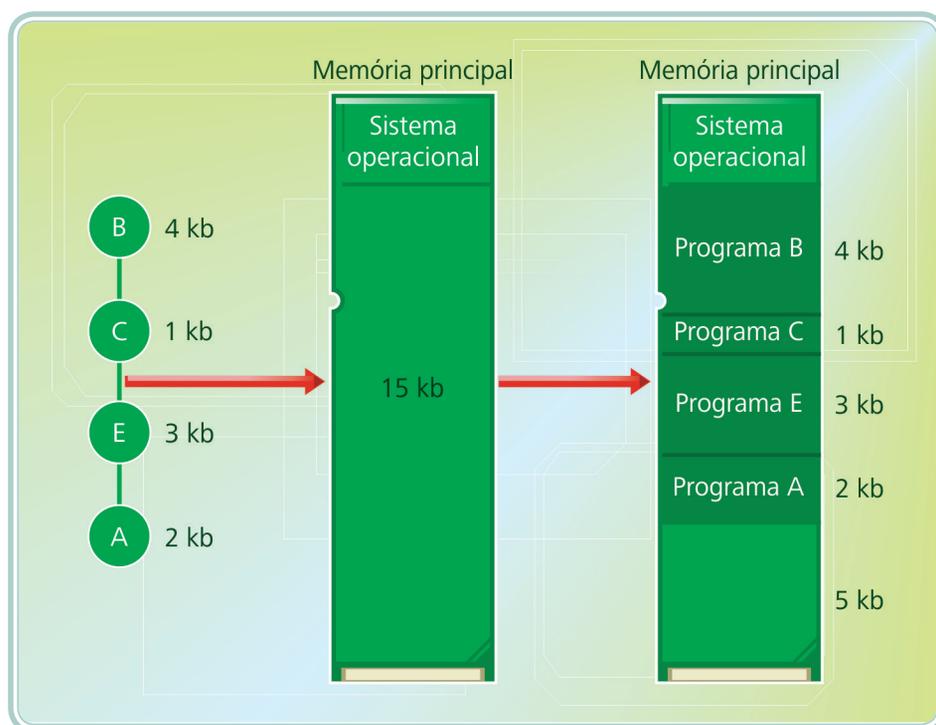


Figura 5.4: Alocação particionada dinâmica

Fonte: CTISM, adaptado de Maia, 2007

Entretanto, existe um problema neste tipo de alocação que talvez não seja tão visível, porém preocupante, estamos falando da fragmentação externa. Este problema ocorre quando os programas que foram alocados em memória principal, começam a terminar sua execução, deixando espaços de memória entre programas, de forma que estes espaços são menores que os tamanhos necessários aos programas que serão carregados para a memória. O problema disso é que mesmo tendo espaços de memória disponíveis, não é possível alocar um espaço para o próximo processo, conforme podemos visualizar na demonstração da Figura 5.5.

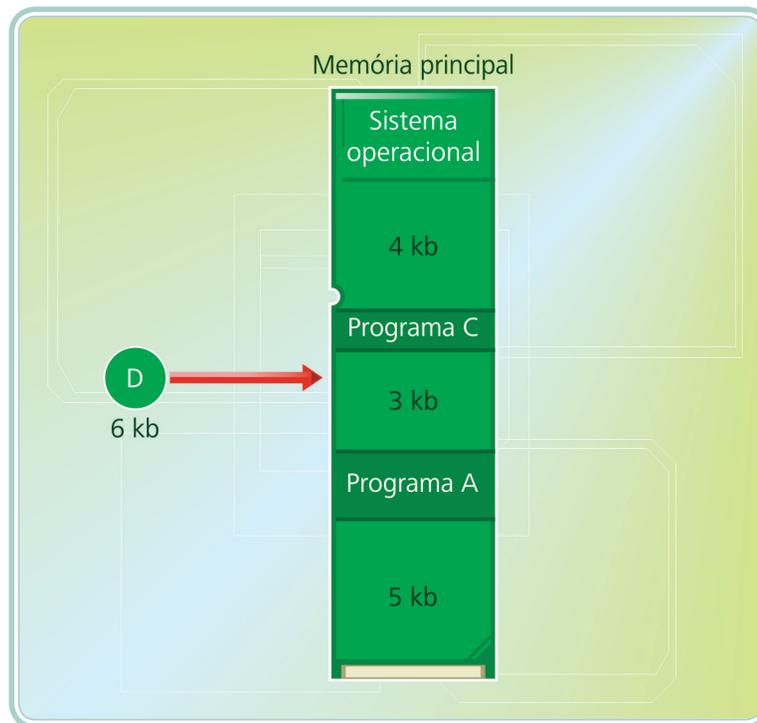


Figura 5.5: Fragmentação externa

Fonte: CTISM, adaptado de Maia, 2007

Para resolver o problema da fragmentação externa existem duas situações principais. A primeira é que conforme os programas vão encerrando sua execução, os espaços adjacentes podem ser unidos, como forma de ter maior espaço disponível para alocação de um programa. A outra alternativa seria compactar os programas em execução um ao lado do outro e juntar os espaços de memória livres, a este processo denominados de relocação dinâmica. Entretanto a relocação dinâmica tem seus custos computacionais, ou seja, esta técnica além de possuir um algoritmo complexo, demandaria consumo de diversos recursos do sistema, como por exemplo, processador e área em disco, o que podem torná-lo inviável em determinados casos.

5.3.3 Estratégias de alocação de partição

Para que o sistema operacional possa gerenciar a memória principal da melhor forma possível, são usadas basicamente três estratégias. Estas visam evitar ou diminuir o problema da fragmentação externa. Geralmente a estratégia leva em conta o tamanho do programa que será executado.

Vamos conhecer agora como funcionam (basicamente) as estratégias de alocação de partição.

5.3.3.1 Best-fit

Conhecida como “melhor alocação”, esta estratégia de alocação, procura solicitar o menor espaço em memória disponível suficiente para o tamanho do programa a ser alocado. Este algoritmo ordena as áreas de memória disponíveis pelo tamanho da mesma, diminuindo o tempo de resposta em buscar um espaço específico. A desvantagem deste algoritmo é a fragmentação.

5.3.3.2 Worst-fit

Na estratégia *worst-fit* é escolhido o maior espaço disponível em memória principal. Apesar de selecionar os maiores espaços em memória, esta estratégia deixa espaços livres maiores, permitindo que um maior número de programas possa utilizar a memória, fazendo com que a fragmentação diminua.

5.3.3.3 First-fit

Como o próprio nome sugere, o primeiro espaço de memória com espaço suficiente para alocar o programa é selecionado. Esta estratégia consome menos recursos do sistema, pois o seu algoritmo organiza as áreas de memória por tamanho, sendo bastante provável que encontre a área procurada com rapidez. Um esquema, mostrando de forma visual o funcionamento das estratégias, pode ser visualizado na Figura 5.6.

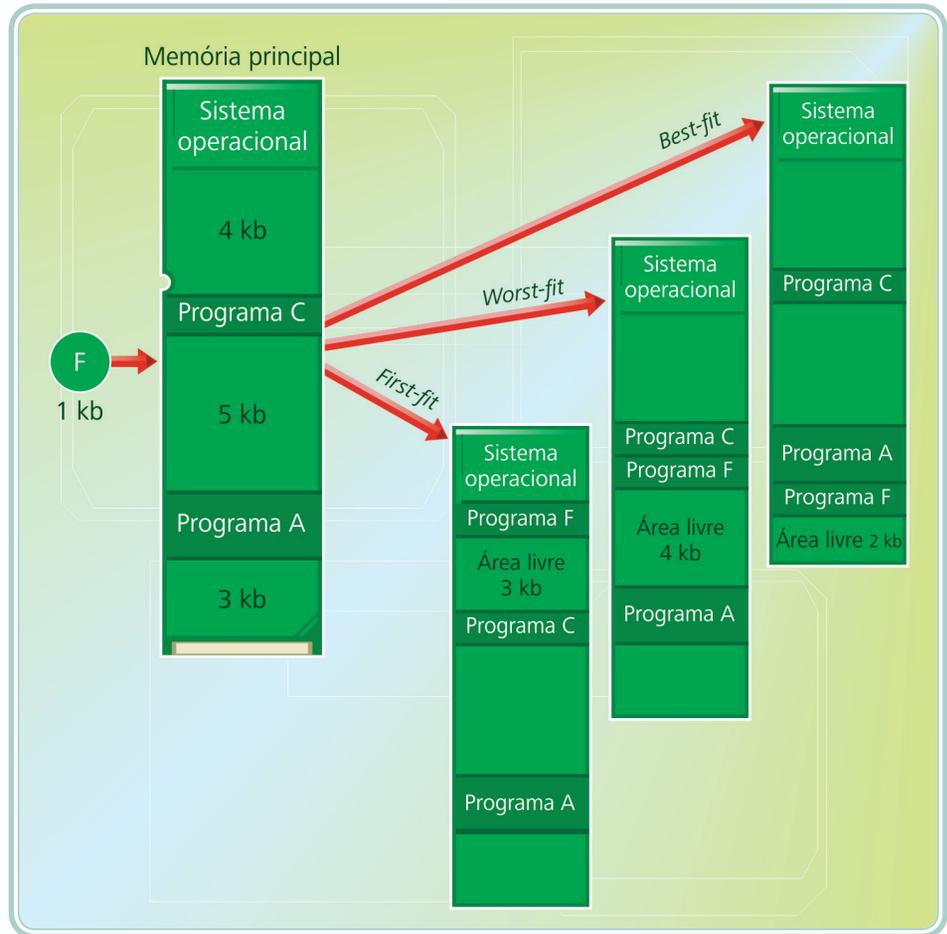


Figura 5.6: Estratégias para alocação de partição

Fonte: CTISM, adaptado de Maia, 2007

5.3.4 Swapping

O *swapping* surgiu com a intenção de resolver problemas que as diversas técnicas de alocação de memória não conseguiram. O principal deles é que mesmo com o gerenciamento de memória presente nos sistemas operacionais multiprogramáveis, muitas vezes um programa não era executado por falta de espaço livre. É justamente neste ponto que entra o *swapping*. A técnica de *swapping* consiste em utilizar um espaço em disco rígido como memória virtual. Para que este recurso seja possível, o sistema escolhe um processo residente (programa em memória principal) que é transferido da mesma para a memória secundária (memória virtual – espaço em disco). Para este processo de troca (memória principal/memória virtual) denominamos de *swap-out*. Posteriormente, o processo é carregado de volta (do disco para a memória principal) para que possa ser executado. Para este processo de troca, chamamos de *swap-in*. Para escolher o programa que deve deixar a memória principal e ir para a memória secundária (memória virtual) o sistema deve selecionar os programas com a menor probabilidade de serem executados num instante

próximo, para evitar assim o *swapping* desnecessário. Por outro lado, somente os programas em memória principal podem ser executados pelo processador, portanto, programas prestes a serem executados necessitam obrigatoriamente estarem na memória principal.

Assim, a técnica de *swapping* permite um maior compartilhamento da memória principal (RAM) e uma melhor utilização dos recursos do sistema computacional. O maior problema enfrentado pelo sistema ao ter que realizar o *swapping* é o elevado número de operações de entrada e saída. Estes trabalham com velocidades de acesso à leitura e escrita completamente diferentes (em comparação à memória principal e memória secundária), sendo que quanto mais se utiliza a memória virtual, maior o impacto causado ao pleno funcionamento do sistema operacional (podendo o sistema deixar de realizar tarefas ou impedir a execução de outros processos). A esta situação denominados de *trashing*, considerada um problema crítico quanto ao gerenciamento de memória nos sistemas operacionais modernos.

Resumo

Nessa aula, vimos como o gerenciamento de memória é fundamental para que o sistema operacional possa executar suas tarefas de maneira eficiente. Foi possível conhecer também as principais técnicas utilizadas para a alocação e gerenciamento de memória pelo sistema operacional. Para fixar o conteúdo visto em cada aula, é importante que você realize os exercícios de aprendizagem. Em nossa próxima aula, falaremos sobre o gerenciamento de arquivos.

Atividades de aprendizagem

1. Cite e explique três funções básicas da memória principal.
2. Diferencie alocação contígua simples de alocação particionada estática e dinâmica.
3. Diferencie as estratégias de alocação: *best-fit*, *worst-fit* e *first-fit*.
4. Descreva o que é *swapping*.
5. Diferencie fragmentação interna e fragmentação externa.



Aula 6 – Gerenciamento de arquivos

Objetivos

Apresentar as principais técnicas de gerenciamento de arquivos e diretórios.

Entender o processo de organização de arquivos e diretórios.

Compreender as principais técnicas de alocação utilizadas no gerenciamento de arquivos e diretórios.

6.1 Considerações iniciais

Nessa aula falaremos sobre os sistemas de arquivos existentes nos sistemas operacionais. Os arquivos são gerenciados pelo sistema operacional de forma a facilitar seu uso pelos usuários. O sistema de arquivos (parte do SO que faz a gerência dos arquivos) caracteriza-se como a parte mais visível de um sistema operacional, uma vez que os usuários manipulam arquivos a todo instante. Abordaremos, nessa aula, as principais formas para organização de arquivos, métodos de acesso, bem como, as principais técnicas de alocação.

6.2 Arquivos

Podemos caracterizar um arquivo como um conjunto de informações logicamente relacionadas que podem ser armazenados em diferentes dispositivos físicos e podem representar programas ou dados. Basicamente um arquivo pode conter informações entendíveis pelo processador (executável) ou conter dados manipulados por nós, seres humanos (arquivos de dados), além de outros tipos de informações diversas (diferentes tipos de arquivos e extensões).

Os arquivos são armazenados em diferentes dispositivos como: discos rígidos, discos externos, discos ópticos, *pen drives*, entre outros. Independente do dispositivo onde os arquivos são armazenados, estes devem ser isolados pelo sistema operacional, para que exista uma independência entre arquivos e o meio de armazenamento.

Em geral os arquivos são identificados por um nome e uma extensão. O nome identifica o arquivo, enquanto a extensão tem a finalidade de identificar o conteúdo presente no arquivo. No Quadro 6.1 é possível visualizar alguns exemplos dos principais tipos de arquivos separados por categorias, confira.

Quadro 6.1: Principais categorias e tipos de arquivos		
Categoria	Extensão	Descrição
Executável	.exe	Um arquivo executável é um programa em si. O sistema operacional dá um tratamento especial a este tipo de arquivo, alocando memória para ao mesmo, por exemplo. Um executável pode instalar um programa em seu computador ou até mesmo executar um vírus. Também são extensões executáveis: .bat e .com.
Áudio	.wav	WAVEForm Audio Format é um formato de representação de áudio digital sem compactação.
	.mp3	Extensão de áudio digital mais difundida. Consegue comprimir o conteúdo de um arquivo de áudio digital (".wav", por exemplo) em até 90 % eliminando frequências repetidas ou imperceptíveis ao ouvido humano.
	.wma	Formato equivalente ao MP3, criado pela Microsoft e que ganhou espaço pela compatibilidade com o Windows Media Player (<i>software</i> popular para reprodução de áudio e vídeo digitais e que é distribuído junto com o MS Windows).
	.aac	Criado pela Apple a fim de concorrer diretamente com o MP3 e o WMA, visando superá-los em qualidade sem aumentar demasiadamente o tamanho dos arquivos. Pode ser reproduzido em iPods e similares, além de <i>players</i> de mídia para computador. É um formato muito utilizado em lojas digitais que comercializam músicas pela internet.
Vídeo	.avi	Abreviação de <i>Audio Video Interleave</i> é um formato de vídeo criado pela Microsoft que combina trilhas de áudio e vídeo, podendo ser reproduzido na maioria dos <i>players</i> de mídia e aparelhos de DVD. Está atualmente associado ao <i>codec</i> de vídeo DivX que permite uma alta compressão.
	.mpg	Um dos padrões de compressão de áudio e vídeo de hoje, criado pelo Moving Picture Experts Group, origem do nome da extensão. É possível encontrar diversas taxas de qualidade neste formato.
	.mov	Formato de mídia utilizado pelo <i>player</i> QuickTime (Apple).
Imagens	.bmp	O mapa de <i>bits</i> (<i>bitmap</i>) é um dos formatos de imagem mais conhecidos. Apresenta uma imagem em sua forma mais crua, sem perdas e compressões, em função disso o tamanho das imagens é maior do que em outros formatos.
	.gif	<i>Graphics Interchange Format</i> é um formato de imagem semelhante ao BMP, mas que foi amplamente difundido com a internet. É encontrado em imagens de <i>sites</i> , programas de conversação e muitos outros. O maior diferencial do GIF é o fato de permitir pequenas animações com imagens sequenciais assim como o fundo transparente.
	.jpg ou .jpeg	<i>Joint Photographic Experts Group</i> é a origem da sigla, é um formato de imagem que utiliza compressão de dados. A compactação agrega blocos de 8×8 <i>bits</i> , tornando o arquivo final muito mais leve que em um <i>bitmap</i> .
	.png	Este formato surgiu em sua época pelo fato dos algoritmos utilizados pelo GIF serem patenteados, encarecendo a utilização daquele formato. Também permite animações e fundo transparente.

Compactadores (compressão de arquivos)	.zip	.zip é a extensão do famoso compactador Winzip.
	.rar	Este é o segundo formato mais utilizado de compactação, tido por muitos como superior ao zip.
Documentos	.txt	Refere-se aos arquivos de texto simples. São extremamente leves e podem ser visualizados por qualquer sistema operacional.
	.doc ou .docx	Denomina a extensão utilizada pelo editor de textos MS Word.
	.xls ou .xlsx	Extensão padrão das planilhas eletrônicas produzidas pelo MS Excel.
	.ppt ou .pptx	Extensões dos arquivos gerados pelo MS PowerPoint, aplicativo que permite a criação de apresentações de <i>slides</i> .
	.pdf	Formato criado pela Adobe e considerado um dos padrões para proteção e portabilidade de documentos. Há inúmeros <i>softwares</i> com capacidade para leitura de tais documentos.
	.rtf	<i>Rich Text Format</i> ou Formato Rico de Texto é um formato de arquivo de documento desenvolvido pela Microsoft para intercâmbio de documentos entre diversos aplicativos.
	.odf	<i>Open Document Format for Office Applications</i> (ODF) é um formato de arquivo usado para armazenamento e troca de documentos de escritório, como textos, planilhas, bases de dados, desenhos e apresentações.

Fonte: Boniati, Preuss, Franciscatto, 2014

6.2.1 Organização de arquivos

A organização de arquivos em um sistema operacional consiste na forma como os dados são armazenados internamente.

Quando um arquivo é criado pode-se definir que tipo de organização será adotada podendo ser uma estrutura suportada pelo próprio sistema operacional ou definida pela aplicação em uso.

Uma das formas mais simples de organização de arquivos é através da sequência não estruturada de *bytes*, não havendo nenhuma estrutura lógica para os dados, onde a aplicação é quem define toda a organização.

Os sistemas operacionais estabelecem diferentes organizações de arquivos e cada um segue um modelo suportado pelo sistema de arquivos. Neste contexto, as organizações mais conhecidas e implementadas são: sequencial, relativa e indexada.

6.2.2 Métodos de acesso

Os métodos de acesso levam em consideração de que forma os arquivos são organizados no sistema operacional. A partir desta organização o sistema de arquivos pode recuperar registros de diferentes formas, confira.

- **Acesso sequencial** – nos primeiros sistemas operacionais o armazenamento de dados se dava através da utilização de fitas magnéticas. Nestas, o acesso tornava-se restrito à leitura na ordem em que eram gravados, sendo a gravação de arquivos possível somente no final de cada arquivo.
- **Acesso direto** – em decorrência dos discos magnéticos, foi possível a inserção de métodos de acesso mais eficientes. Um deles é o acesso direto. Este permite a leitura/gravação de um registro especificamente na sua posição, através do número de registro. Neste método não existe restrição quanto à ordem em que os registros são lidos ou gravados, sendo necessário somente o número do registro. O único requisito neste tipo de acesso é que o mesmo é possível quando o arquivo é definido com registros de tamanho fixo.
- **Acesso indexado ou por chave** – caracteriza-se como um método de acesso mais sofisticado e tem como base o acesso direto. O arquivo em questão deve possuir uma área de índice onde existam ponteiros para os diversos registros. A partir desta informação realiza-se um acesso direto.

6.2.3 Operações de entrada e saída

Elas são vitais para a comunicação entre aplicação e dispositivos. Assim, poderíamos descrever simplificadaamente as operações de entrada e saída em um sistema operacional.

O sistema de arquivos (abordado anteriormente) oferece um conjunto de chamadas do sistema (*system calls*) que permite as aplicações realizarem operações de entrada e saída (E/S), como por exemplo: tradução de nomes em endereços, leitura e gravação de dados, criação e eliminação de arquivos, entre outras rotinas.

Elas (chamadas do sistema) têm como função principal oferecer uma interface simples e uniforme entre a aplicação que está sendo utilizada e os diversos dispositivos existentes em um computador.



Figura 6.1: Operações de entrada e saída

Fonte: CTISM, adaptado de Maia, 2007

6.2.4 Atributos

Os atributos são informações de controle que dependendo do sistema operacional e do sistema de arquivos podem variar, sendo alguns deles presentes de forma padrão, tais como: tamanho, criador, proteção, data, entre outros.

Alguns desses atributos (designados na criação do arquivo) não podem ser mudados, enquanto outros são modificados pelo próprio sistema operacional.

Ainda existem alguns atributos que podem ser alterados pelo usuário, como proteção, tamanho, senha, etc. No Quadro 6.2, são apresentados alguns dos principais atributos de arquivos existentes.

Quadro 6.2: Atributos de arquivos	
Atributos	Descrição
Tamanho	Especifica o tamanho do arquivo.
Proteção	Código de proteção de acesso.
Dono	Identifica o criador do arquivo.
Criação	Data e hora da criação do arquivo.
<i>Backup</i>	Data e hora do último <i>backup</i> realizado.
Organização	Indica a organização lógica dos registros.
Senha	Senha necessária para acessar o arquivo.

Fonte: Adaptado de Maia, 2007

6.3 Diretórios

Os diretórios presentes e criados nos sistemas operacionais são o modo como o sistema organiza os diferentes arquivos contidos em um disco.

O diretório pode ser descrito como uma estrutura de dados que contém entradas associadas aos arquivos, onde temos informações como: nome, localização física, organização, entre outros atributos.

Ao abrir um arquivo, o sistema operacional procura sua entrada na estrutura de diretórios, armazenando as informações do arquivo em uma tabela que é mantida na memória principal. Veremos agora, as principais implementações de estruturas de diretórios, vamos lá:

- **Nível único** (*single level directory*) – esta implementação é considerada a mais simples e nesta estrutura existe apenas um único diretório contendo todos os arquivos do disco. Devido ao seu formato e forma de funcionamento, acaba sendo bastante limitado, uma vez que não permite aos usuários criarem arquivos com o mesmo nome (causando neste caso um conflito no acesso aos arquivos).
- **Diretório de arquivos de usuário** (*User File Directory – UFD*) – esta estrutura provê para cada usuário um diretório particular, onde é possível criar arquivos com qualquer nome, diferentemente do modelo anterior. Para que seja possível localizar arquivos nessa estrutura, faz-se necessário um nível adicional para controle dos diretórios individuais dos usuários, chamado de *Master File Directory (MFD)* que é indexado pelo nome do usuário e nele cada entrada aponta para o diretório pessoal. Esta estrutura é análoga a uma estrutura de dados em árvore, sendo que o MFD é a raiz, o UFD são os galhos e os arquivos são as folhas. Assim, quando referenciamos um arquivo é necessário especificar também o seu nome e seu diretório. Este processo é chamado de *path* ou caminho.

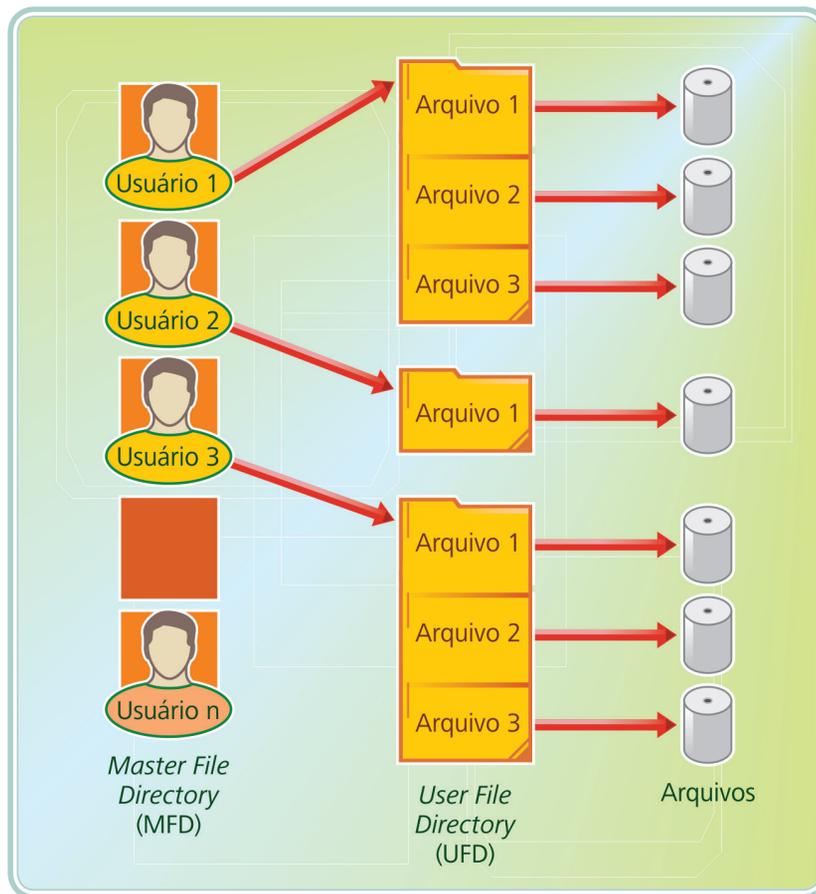


Figura 6.2: Estrutura de diretórios com dois níveis

Fonte: CTISM, adaptado de Maia, 2007

6.4 Alocação de espaço em disco

Para que seja possível a criação de arquivos junto ao sistema operacional, o mesmo deve ter um controle de quais blocos ou áreas do disco estão livres (disponíveis). Este controle só é possível graças a uma estrutura de dados mantida pelo SO (geralmente lista ou tabela) que mantém estas informações atualizadas e possibilita ao sistema de arquivos gerenciar o espaço livre, sempre que necessário. Veremos agora as principais técnicas de gerência de arquivos, confira.

6.4.1 Alocação contígua

Esta técnica tem por objetivo armazenar um arquivo em blocos dispostos de maneira sequencial no disco rígido. Nesta alocação o sistema localiza um arquivo através de seu endereço do primeiro bloco e sua extensão em blocos. O acesso é simplificado (independente da forma sequencial ou direta), sendo o principal problema a alocação de novos arquivos nos espaços livres, o que demanda uma quantidade de blocos livres em sequencia igual ou maior que o tamanho do espaço a ser alocado.

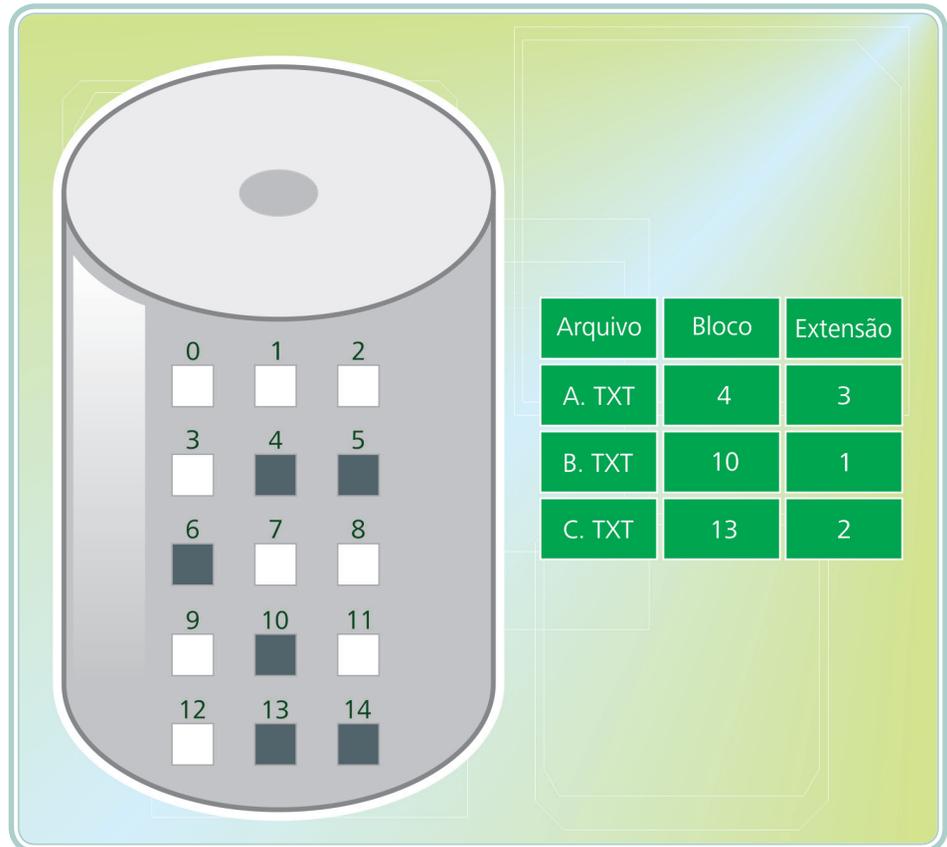


Figura 6.3: Alocação contígua

Fonte: CTISM, adaptado de Maia, 2007

Uma situação que pode ocorrer neste tipo de alocação é existirem diversos espaços livres para alocar um determinado arquivo. Neste caso, entram em cena as estratégias (algoritmos) de alocação, que permitem dinamizar esta escolha, através de três alternativas principais:

- **First-fit** – o primeiro segmento de espaços livres com tamanho suficiente para alocar o arquivo é o escolhido.
- **Best-fit** – seleciona o menor segmento livre disponível com tamanho suficiente para armazenar o arquivo. Neste caso é necessária uma busca por toda a lista para que seja possível saber qual entre todos os espaços tem o menor tamanho suficiente ou então agrupá-los previamente por tamanhos (do menor para o maior, por exemplo).
- **Worst-fit** – o maior segmento de espaço livre entre todos é selecionado. Porém, como no exemplo anterior (*best-fit*) se faz necessária uma busca por todos os espaços livres ou uma ordenação por tamanho.

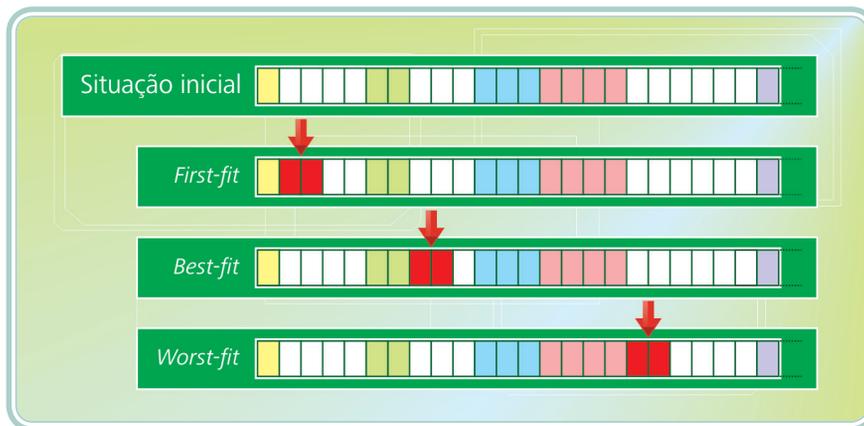


Figura 6.4: Estratégias de alocação contígua

Fonte: CTISM

Um dos problemas da alocação contígua (independente da estratégia a ser adotada) é a fragmentação de espaços livres. Este problema é ainda maior conforme existam blocos livres disponíveis, porém sem espaço suficiente contíguo onde um arquivo possa ser alocado. Para resolver tal problema a desfragmentação de disco se faz necessária, como forma de agrupar de um lado os arquivos e de outro os espaços disponíveis (reorganizando os espaços livres em um único segmento). Esta técnica demanda de um grande consumo de tempo e tem efeito temporário.

6.4.2 Alocação encadeada

Na alocação encadeada os arquivos são organizados na forma de conjunto de blocos ligados ao disco rígido (independente de sua localização física) e cada um destes blocos deve ter um ponteiro para o bloco seguinte, para que seja possível sempre buscar a próxima parte do arquivo que foi armazenado.

Neste método temos a fragmentação de arquivos (divisão do arquivo em pedaços menores, chamados de *extents*) o que aumenta o tempo de acesso a este arquivo haja vista, que se faz necessário procurá-lo em blocos que não estão em uma sequência contínua, mas sim, espalhados pelo disco.

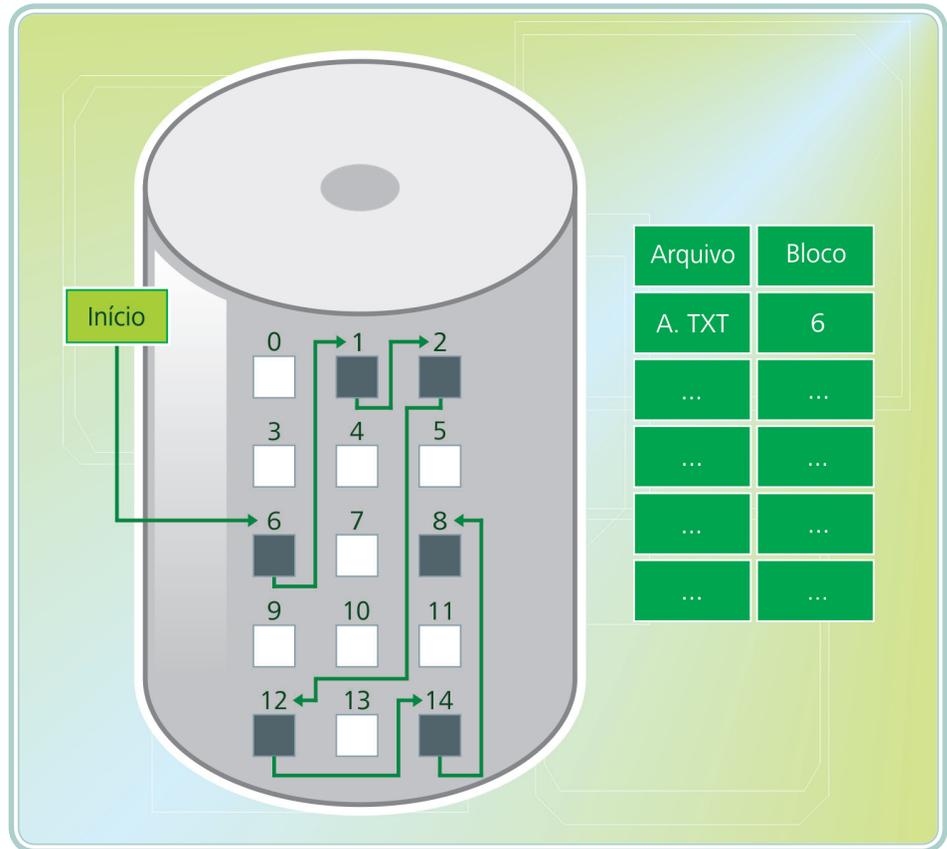


Figura 6.5: Alocação encadeada

Fonte: CTISM, adaptado de Maia, 2007

6.4.3 Alocação indexada

Este tipo de alocação vem a resolver um problema comum oriundo da alocação encadeada, que é não permitir o acesso direto aos blocos dos arquivos. Na alocação indexada é utilizada uma estrutura denominada de bloco de índice onde se armazena os ponteiros de todos os blocos de arquivos. Dessa forma, é possível ter acesso direto aos blocos do arquivo.

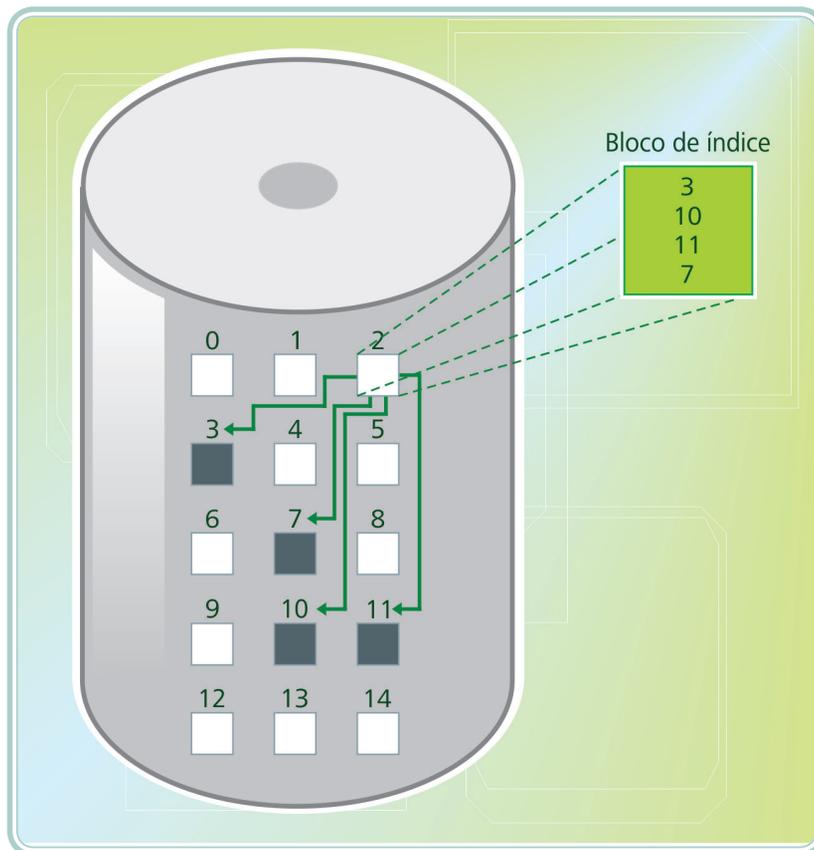


Figura 6.6: Alocação indexada
 Fonte: CTISM, adaptado de Maia, 2007

6.5 Proteção de acesso

A proteção de acesso é fator fundamental nos sistemas de arquivos, principalmente devido ao fato que os meios de armazenamento podem ser compartilhados de diferentes modos. Faz-se necessários mecanismos capazes de proteger a informação e garantir esta proteção a arquivos e diretórios.

Geralmente esta proteção, na grande maioria dos sistemas operacionais ocorre através da concessão do tipo de acesso a arquivos e diretórios, como por exemplo: leitura, gravação, execução ou eliminação dos mesmos.

Existem diferentes mecanismos que podem ser empregados de forma diferenciada a arquivos e diretórios, como forma de proteção dos mesmos. Vejamos agora, algumas delas.

6.5.1 Senha de acesso

Recurso pelo qual a proteção se dá, pela utilização de um código, previamente cadastrado, sendo que somente quem detém este código, pode ter acesso ao conteúdo protegido.

A utilização de senha de acesso é aconselhável para conteúdos sigilosos ou que necessitam de uma proteção prévia, em contrapartida acaba dificultando quando necessitamos compartilhar este arquivo protegido com mais pessoas, uma vez que os demais usuários também deveriam ter conhecimento da senha.

6.5.2 Grupos de usuário

Os grupos de usuário são uma boa alternativa para arquivos e diretórios compartilhados em uma rede de computadores, por exemplo. O princípio de um grupo de usuários é reunir em um grupo usuários que pertencem a uma mesma classificação, organização ou acesso afim.

Este mecanismo de acesso provê três níveis básicos de proteção: dono, grupo e todos. Ao aplicar uma regra de proteção a um arquivo ou diretório é necessário indicar se esta regra se aplica a cada um dos três níveis. Ex.: o arquivo “xyz” pode ser acessado somente pelo dono, grupo e não pelos demais. Ainda nesta especificação é necessário indicar o tipo de acesso que poderá ser feito: leitura, escrita, execução e eliminação, aos três níveis de proteção.

Resumo

Nessa aula, vimos como se dá a organização de arquivos e diretórios em um sistema operacional e como as técnicas de alocação de espaço em disco são importantes para que a leitura e gravação de dados possam acontecer de forma objetiva. Para fixar o conteúdo visto em cada aula, é importante que você realize os exercícios de aprendizagem. Em nossa próxima aula, abordaremos um estudo de caso sobre o sistema operacional Windows Seven, apresentando os principais recursos do mesmo.



Atividades de aprendizagem

1. Diferencie os modos de acesso sequencial, direto e indexado.
2. Cite três tipos de atributos que podem estar presentes em um arquivo, descrevendo a função dos mesmos.
3. Como podemos descrever a estrutura de diretórios com dois níveis?
4. Diferencie alocação contígua, encadeada e indexada.

Aula 7 – Estudo de caso: sistema operacional Windows Seven

Objetivos

Conhecer de forma prática o sistema operacional Windows Seven.

Compreender seus requisitos e forma de instalação.

Entender as principais ferramentas do sistema e como podemos utilizá-las na administração do sistema operacional.

Conhecer os principais comandos funcionais de administração.

7.1 Considerações iniciais

Os sistemas operacionais da Microsoft são bastante conhecidos e utilizados em todo o mundo. Eles variam (quanto aos tipos) dependendo do dispositivo onde são utilizados (*smartphones*, computadores pessoais, servidores, entre outros) e possuem inúmeros recursos e características peculiares. Abordaremos, nessa aula, um estudo de caso, caracterizando de forma prática, os principais recursos presentes no sistema operacional Windows Seven, confira.

7.2 Requisitos

Algo que sempre é aconselhado, antes de instalar um sistema operacional em um dispositivo (computador, *notebook*, servidor, etc.), é saber os requisitos mínimos de *hardware* necessários. Para isso, basta buscar esta informação no *site* oficial de cada fabricante. No caso do sistema operacional Windows Seven, os requisitos mínimos necessários são:

- Processador de 01 Gigahertz (GHz) ou superior de 32 ou 64 *bits*.
- 16 Gigabyte (GB) de espaço em disco rígido disponível (para sistemas 32 *bits*) ou 20 GB (para sistemas 64 *bits*).
- 01 GB de memória RAM (32 *bits*) ou 02 GB de memória RAM (64 *bits*).
- Dispositivo gráfico DirectX 9 ou superior.

7.3 Instalação

A instalação do Windows Seven geralmente se dá através de um CD ou DVD contendo os arquivos necessários para instalação do mesmo. O CD/DVD deve ser adquirido levando em consideração o sistema operacional ideal para cada usuário ou situação (se será utilizado em casa, ou para trabalho, tarefa específica, etc.).



Figura 7.1: Tela inicial de instalação

Fonte: Microsoft Windows Seven

A instalação do sistema operacional envolve alguns passos fundamentais que são:

- a) Processo de *boot* (carregamento dos arquivos necessários para iniciar o processo de instalação do sistema operacional pela mídia indicada).
- b) Seleção do idioma.
- c) Escolher entre instalação ou restauração do sistema (esta segunda opção é indicada se o sistema operacional estiver apresentando problemas).
- d) Aceitar os termos de licença.
- e) Selecionar o formato de instalação (atualização para uma nova versão ou avançada para instalação personalizada).
- f) Particionar e definir parâmetros (quantidade e tamanho das partições).

g) Aguardar o processo de instalação.

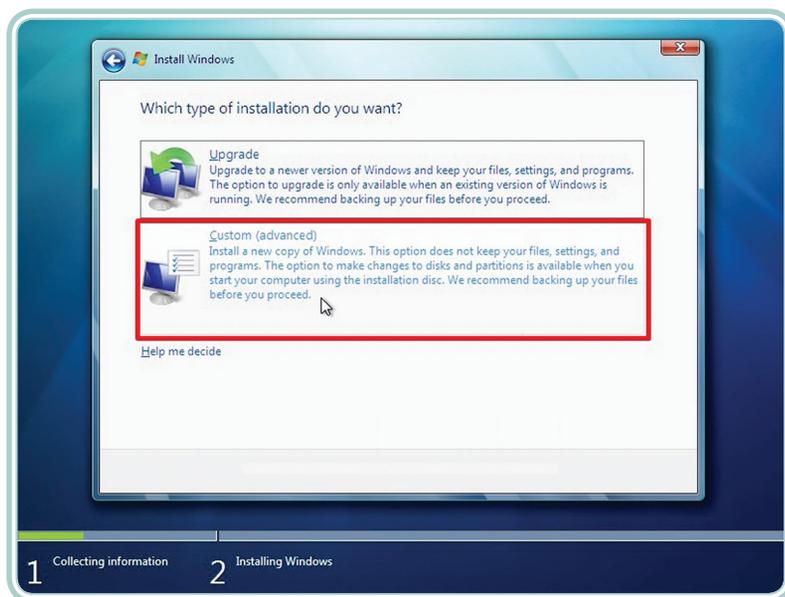


Figura 7.2: Tela de seleção do tipo de instalação

Fonte: Microsoft Windows Seven

7.4 Ferramentas do sistema

Como forma de destacar os principais recursos presentes no sistema operacional Windows Seven, descreveremos agora as principais ferramentas que o mesmo dispõe e uma rápida descrição sobre as mesmas.

7.4.1 Backup e restauração do sistema

Permite realizar uma cópia de segurança (*backup*) dos arquivos do sistema do computador, caso os mesmos sofram com a ação de vírus, exclusão acidental ou falhas de *software* ou *hardware*. A restauração do sistema permite restaurar (voltar) a um ponto anterior no tempo, sem que afete os arquivos pessoais (documentos, imagens, *e-mails*, entre outros). É importante salientar, que este recurso não recupera um arquivo pessoal que foi excluído ou danificado, somente arquivos do sistema operacional Windows. Para recuperar arquivos pessoais excluídos, existem diversos *softwares* que podem ser obtidos na internet que realizam tal procedimento (como é o caso do *software* Recuva). Para acessar as opções de *backup* e restauração do sistema devemos acessar o seguinte caminho:

- Disco local do nosso computador (geralmente C:\).
- Clicar com o botão direito e ir até o menu "Propriedades".

- Clicar na opção “Ferramentas”.
- Por fim, para iniciar as opções de Backup e Restauração, basta clicar no botão “Fazer backup agora...”

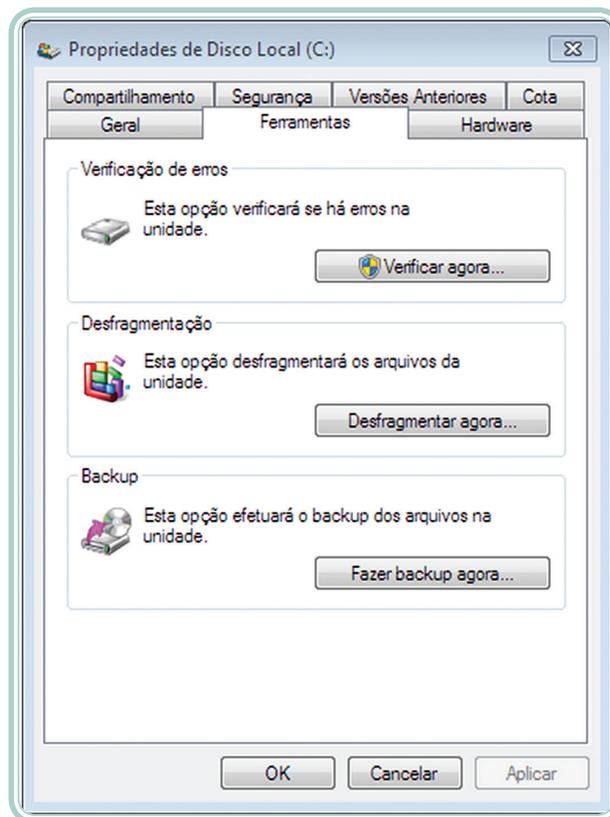


Figura 7.3: Opções de backup e restauração do sistema

Fonte: Microsoft Windows Seven

7.4.2 Ponto de restauração

O ponto de restauração é uma solução rápida que permite criar uma representação do sistema e guardá-lo para que seja possível voltar a uma configuração anterior. Os pontos de restauração são criados automaticamente e também quando o sistema detecta o início de uma alteração no computador, como ao instalar um programa, um *driver* de dispositivo, entre outros.

Para criar um ponto de restauração do sistema, basta seguir os seguintes passos:

- Clicar no botão Iniciar (Windows).
- Digitar no campo “Pesquisar programas e arquivos”: Criar ponto de restauração.

Você será direcionado diretamente a imagem presente na Figura 7.4:

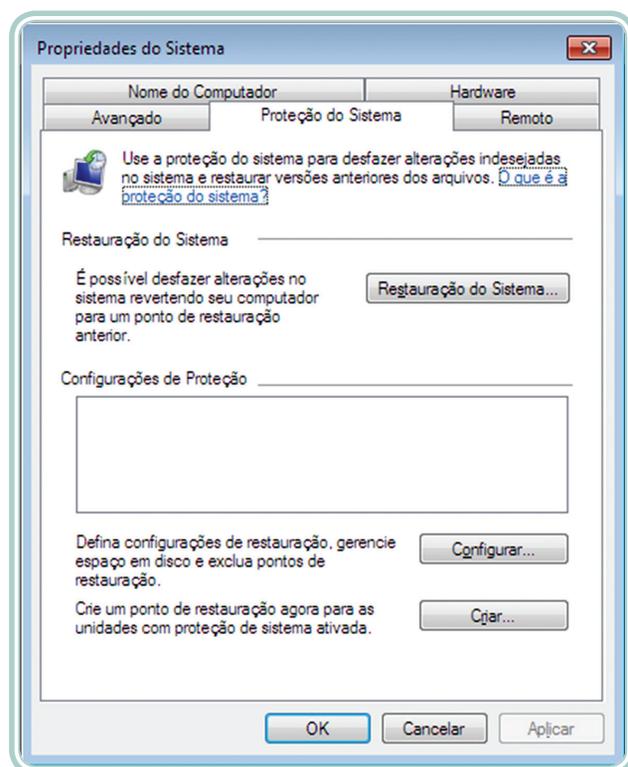


Figura 7.4: Criar ponto de restauração

Fonte: Microsoft Windows Seven

Para dar início a criação do ponto de restauração, basta clicar no botão “Criar” e preencher os dados que se pede.

7.4.3 Informações do sistema

O recurso “informações do sistema” permite saber detalhes importantes do sistema operacional, como informações de *software*, *hardware* e componentes do computador.

Para ter acesso a estas informações são necessários os seguintes passos:

- Clicar no botão Iniciar.
- Clicar na opção “Todos os programas” e em seguida em “Acessórios”, “Ferramentas do sistema” e por fim, “Informações do sistema”.

Uma tela semelhante a exposta na Figura 7.5, irá abrir.

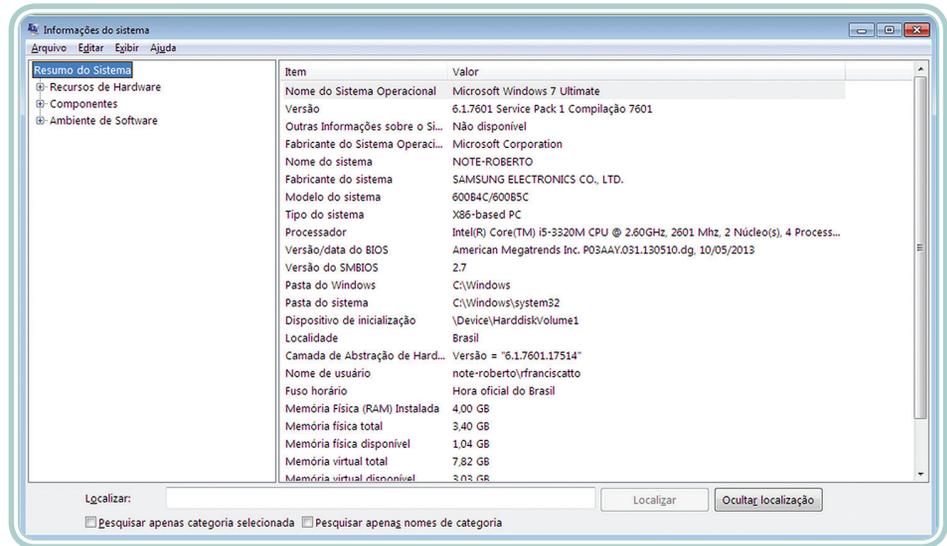


Figura 7.5: Informações do sistema

Fonte: Microsoft Windows Seven

Temos nas “Informações do sistema” quatro categorias principais, que são:

- **Resumo do sistema** – mostra informações gerais sobre o sistema operacional e o computador.
- **Recursos de hardware** – mostra detalhes sobre o *hardware* existente no computador (DMA, E/S, IRQs – indicado para usuários mais experientes).
- **Componentes** – exibe informações sobre os componentes instalados no computador (placas de rede, dispositivos USB, CD-ROM, etc.).
- **Ambiente de software** – mostra informações a nível de *software*, como por exemplo: *drivers* do sistema, serviços, relatórios de erros do Windows, entre outros.

7.4.4 Gerenciador de tarefas do Windows

O gerenciador de tarefas do Windows é um bom aliado quando precisamos saber informações sobre o sistema de uma forma prática e objetiva. Este recurso mostra informações sobre programas em execução, processos, serviços, entre outros.

Além destes, também podemos através deste recurso monitorar o desempenho do computador, obter informações de rede, bem como usuários conectados.

A maneira mais fácil de acessar o gerenciador de tarefas do Windows é pressionando simultaneamente as teclas: <Ctrl + Shift + Esc>. Esta combinação fará com que a tela exposta na Figura 7.6, seja exibida:

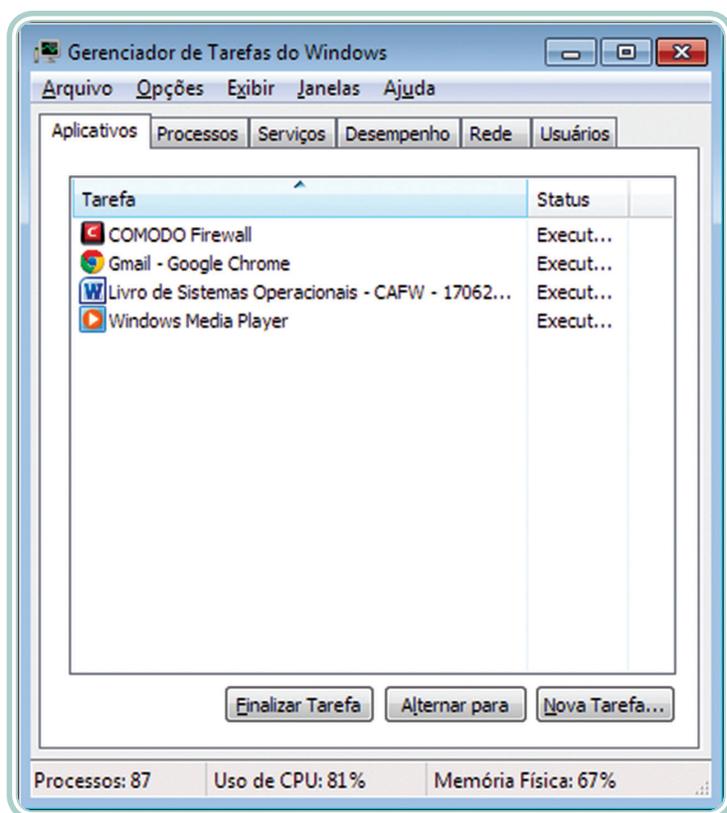


Figura 7.6: Gerenciador de tarefas do Windows

Fonte: Microsoft Windows Seven

Note que no topo da imagem temos seis guias, na ordem, descritas como: “Aplicativos”, “Processos”, “Serviços”, “Desempenho”, “Rede” e “Usuários”. Cada uma delas traz informações importantes acerca do sistema operacional, confira:

- **Aplicativos** – mostra informações sobre os programas (aplicativos) que estão sendo executados pelo sistema operacional neste momento, bem como, o *status* dos mesmos (executando). Os três botões que aparecem no rodapé da Figura 7.6 (“Finalizar Tarefa”, “Alternar para” e “Nova Tarefa...”) permitem ao usuário respectivamente, finalizar um programa que esteja funcionando de forma anormal, alternar para um programa presente na lista ou iniciar uma nova tarefa (executar um novo programa).
- **Processos** – a guia processos mostra informações sobre os processos que estão em execução. Estes processos podem ser do sistema, um serviço do

sistema operacional ou iniciados pelo próprio usuário. Como vimos em capítulos anteriores, os processos são vitais ao sistema operacional e saber interpretá-los faz-se necessário. É possível saber, por exemplo, quanto um processo está consumindo de recursos como: processador, memória, *bytes* de entrada e saída em disco, entre outros. Além disso, podemos eliminar um processo do sistema, através desta guia, bastando para isso clicarmos com o botão direito em cima do processo que desejamos e em seguida na opção “Finalizar Processo”.

- **Serviços** – os serviços do sistema operacional Windows são responsáveis por uma série de funções importantes para o sistema, como ativações de programas, controles e iniciação de rotinas específicas. Nesta opção podemos visualizar os serviços em execução e a qualquer momento podemos interromper um serviço, para configuração e reestabelecimento de uma nova função.
- **Desempenho** – esta guia permite obter informações acerca do desempenho do computador, mostrando dados gerais sobre a utilização atual dos principais recursos do computador. Entre outras informações podemos visualizar: uso do processador (em porcentagem), memória RAM, memória física (total, em *cache*, livre), informações do sistema, além da opção “Monitor de Recursos” que direciona o usuário a um aplicativo que monitora inúmeras funções do computador e da rede em tempo real.
- **Rede** – nesta opção, informações sobre os adaptadores de rede são fornecidos, bem como, um *status* sobre velocidade, utilização da rede, estado, entre outros.
- **Usuários** – exibe os usuários que estão utilizando o sistema operacional atualmente. Esta opção permite identificar um usuário, sua sessão, seus *status*, além de permitir desconectar ou fazer *logout* do mesmo.

7.4.5 Configuração do sistema

A configuração do sistema, como o próprio nome sugere permite personalizar algumas configurações do sistema, como por exemplo, programas inicializados ao carregar o sistema operacional, tipo de inicialização do sistema, serviços ativos, entre outros. Para acessar este interessante utilitário, devemos seguir os seguintes passos:

- Clicar no botão Iniciar.

- Digitar no campo “Pesquisar programas e arquivos”: msconfig.

A partir destes passos você terá acesso a uma janela semelhante à apresentada na Figura 7.7.

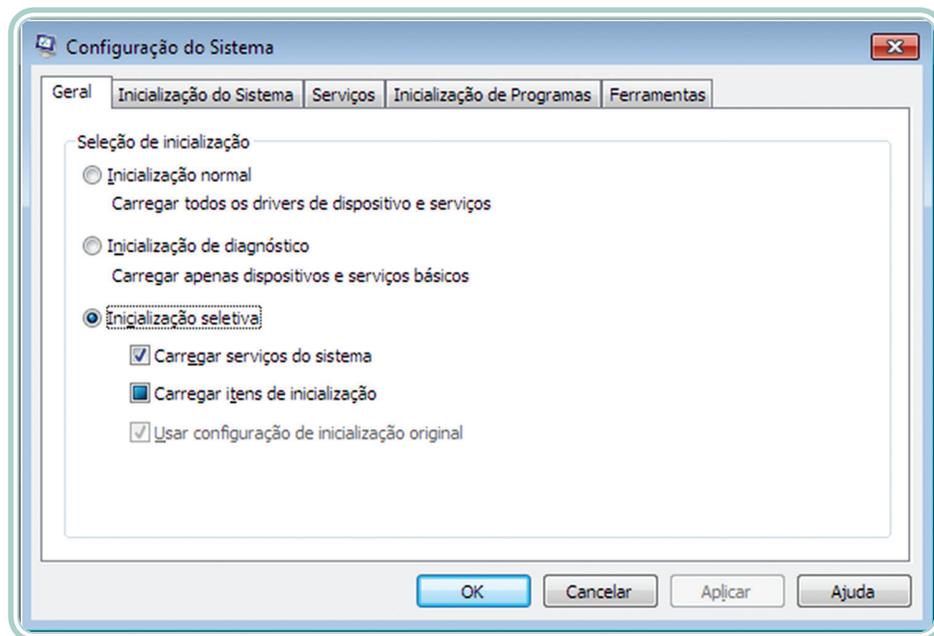


Figura 7.7: Configuração do sistema – msconfig

Fonte: Microsoft Windows Seven

Na aba “Geral” podemos definir que tipo de inicialização queremos para o sistema operacional Windows Seven. As opções “Inicialização normal”, “Inicialização de diagnóstico” e “Inicialização seletiva” podem ser selecionadas.

Na aba “Inicialização do Sistema” podemos definir algumas opções de inicialização, como por exemplo, a inicialização segura e o tempo limite da mesma.

Na aba “Serviços”, temos acesso aos principais serviços do sistema, bem como seu *status* atual.

A aba “Inicialização de Programas” permite selecionar os programas que serão carregados ao inicializar o sistema operacional. Dessa forma, podemos tornar o carregamento mais rápido sem a necessidade de carregar todos os programas na inicialização, mas somente aqueles realmente necessários.

A aba “Ferramentas” permite acessar os principais aplicativos de gerenciamento do sistema operacional, bastando localizá-lo na lista e clicar sobre o botão “Iniciar”.

7.5 Principais comandos de administração do sistema

O sistema operacional Windows Seven, possui uma grande quantidade de recursos administrativos e de configuração que podem ser acessados através de aplicativos gráficos e também através de comandos.

Os comandos são um meio rápido e eficaz de aplicar determinada operação ao sistema operacional. Conheceremos agora, alguns destes comandos de administração que podem ser úteis na utilização do sistema operacional Windows Seven, confira.

7.5.1 Acessando o *Prompt de comando* como administrador

Para que possamos digitar comandos no Windows, precisamos ter acesso ao “Prompt de Comando”. Este aplicativo possui uma interface simples a espera de comandos do usuário. Outro detalhe importante é que a maioria dos comandos necessita da autenticação por parte do usuário administrador do sistema. Para que possamos acessar o “Prompt de Comandos” como usuário administrador, fazemos os seguintes passos:

- Clicar no botão Iniciar – Todos os Programas – Acessórios.
- Clicar com o botão direito sobre o item “Prompt de Comando” e em seguida na opção “Executar como administrador”, conforme a Figura 7.8.

Com acesso ao Prompt de Comando do Windows Seven, podemos começar a interagir com o mesmo, através da digitação de comandos específicos, acompanhe.

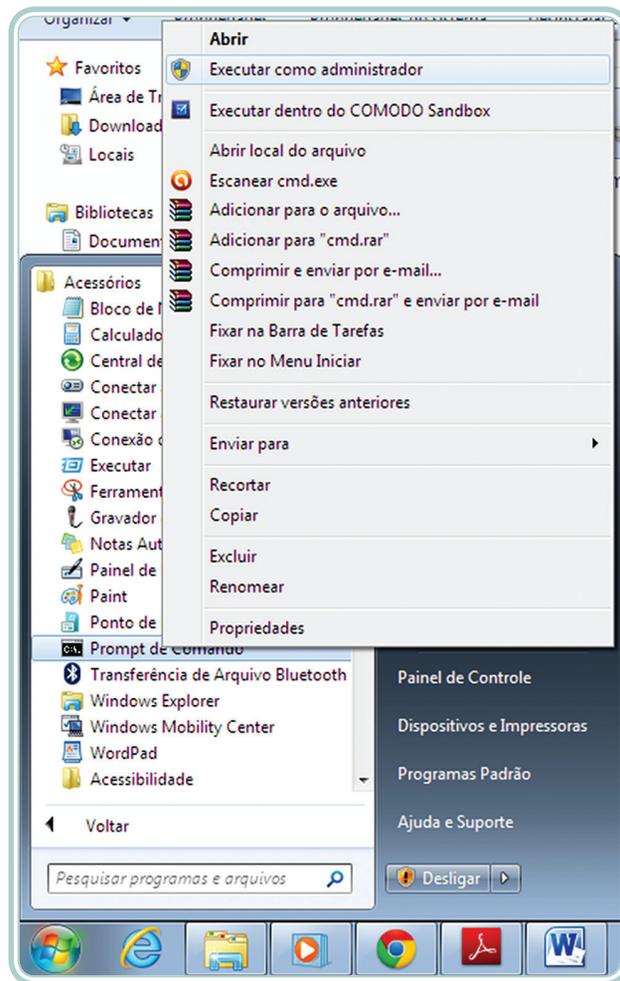


Figura 7.8: Acessar Prompt de Comando como administrador

Fonte: Microsoft Windows Seven

7.5.2 Checagem/correção de disco

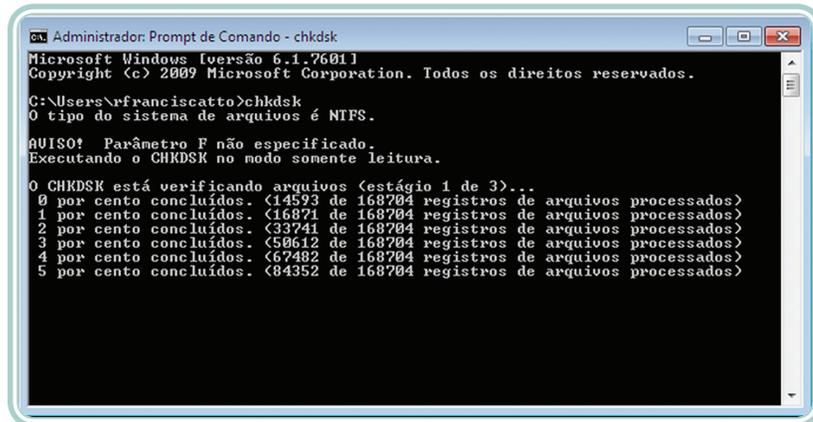
A correção de disco (seja ele o disco rígido presente no computador ou um disco externo conectado ao computador) é uma rotina comum e muitas vezes eficaz quanto a identificar e corrigir problemas nestes dispositivos. Para realizar tal procedimento via Prompt de Comando, devemos digitar o seguinte comando (considerando que estamos com o Prompt de Comandos aberto e executado como administrador, conforme exemplificado no item 7.5.1):

```
chkdsk
```

O comando "chkdsk" faz a checagem do disco atual (neste exemplo, Figura 7.9, Disco local C:\) e permite checar e corrigir também outras unidades de disco presentes no computador (bastando indicar a unidade a ser analisada). Para analisar a unidade de disco E:\ (por exemplo) deveríamos realizar os seguintes comandos:

cd E:\ (pressione a tecla "Enter" e depois digite o comando abaixo)
chkdsk

Estes comandos farão o mesmo processo da seção 7.5.2, porém, para outra unidade de disco (no exemplo a unidade E:\).



```
Administrador: Prompt de Comando - chkdsk
Microsoft Windows [versão 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. Todos os direitos reservados.

C:\Users\rfranciscatto>chkdsk
O tipo do sistema de arquivos é NTFS.
AVISO! Parâmetro F não especificado.
Executando o CHKDSK no modo somente leitura.

0 CHKDSK está verificando arquivos (estágio 1 de 3)...
0 por cento concluídos. <14593 de 168704 registros de arquivos processados>
1 por cento concluídos. <16871 de 168704 registros de arquivos processados>
2 por cento concluídos. <33741 de 168704 registros de arquivos processados>
3 por cento concluídos. <50612 de 168704 registros de arquivos processados>
4 por cento concluídos. <67482 de 168704 registros de arquivos processados>
5 por cento concluídos. <84352 de 168704 registros de arquivos processados>
```

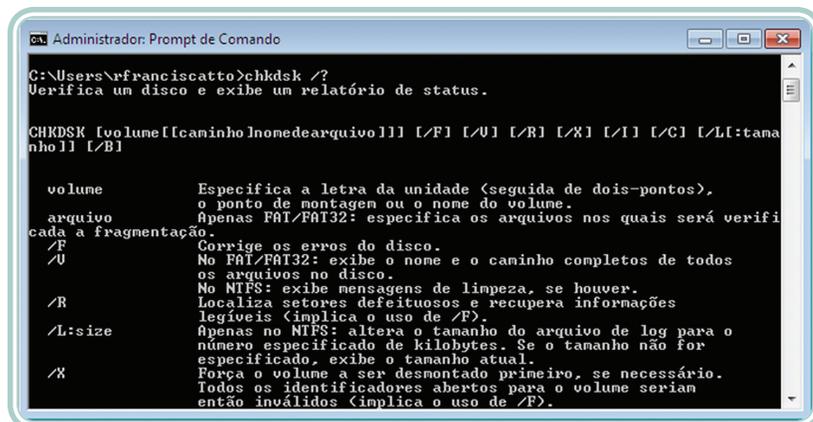
Figura 7.9: Correção de disco com o comando chkdsk

Fonte: Microsoft Windows Seven

Para ver as opções a serem utilizadas junto ao comando chkdsk, basta digitar o seguinte comando:

chkdsk /?

O resultado deste comando pode ser visualizado na Figura 7.10.



```
Administrador: Prompt de Comando

C:\Users\rfranciscatto>chkdsk /?
Verifica um disco e exibe um relatório de status.

CHKDSK [volume[[caninho]nomedearquivo]] [/F] [/U] [/R] [/X] [/I] [/C] [/LI:tam
nho] [/B]

volume      Especifica a letra da unidade <seguida de dois-pontos>,
             o ponto de montagem ou o nome do volume.
arquivo     Apenas FAT/FAT32: especifica os arquivos nos quais será verifi
cada a fragmentação.
/F          Corrige os erros do disco.
/U          No FAT/FAT32: exibe o nome e o caminho completos de todos
os arquivos no disco.
             No NTFS: exibe mensagens de limpeza, se houver.
/R          Localiza setores defeituosos e recupera informações
legíveis <implica o uso de /F>.
/L:size     Apenas no NTFS: altera o tamanho do arquivo de log para o
número especificado de kilobytes. Se o tamanho não for
especificado, exibe o tamanho atual.
/X          Força o volume a ser desmontado primeiro, se necessário.
             Todos os identificadores abertos para o volume seriam
então inválidos <implica o uso de /F>.
```

Figura 7.10: Opções do comando chkdsk

Fonte: Microsoft Windows Seven

7.5.3 Identificar endereço IP e configurações básicas da rede

Muitas vezes necessitamos saber o endereço IP de nosso computador ou o endereço MAC de nossa placa de rede. Esta informação pode ser obtida rapidamente através do seguinte comando:

```
ipconfig /all
```

O comando “ipconfig /all” mostra detalhes da conexão, conforme podemos visualizar na Figura 7.11.

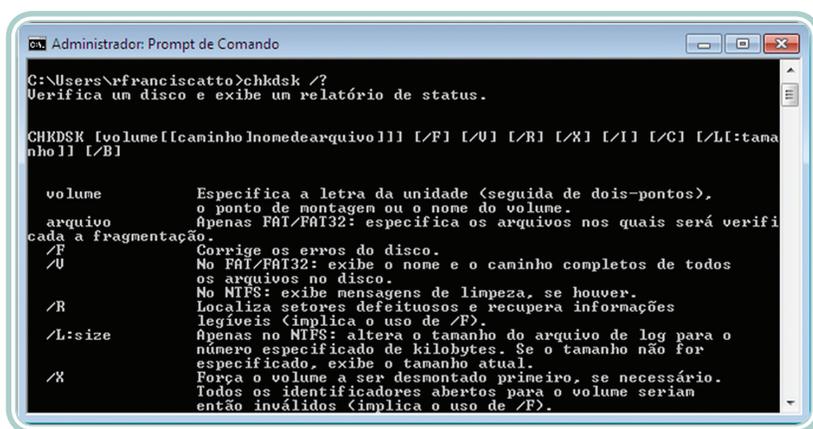


Figura 7.11: Comando ipconfig /all

Fonte: Microsoft Windows Seven

Veja que várias informações são mostradas após a execução do comando, como o nome do *host* (neste exemplo, **note-roberto**), o endereço físico (neste exemplo, **B4-B6-76-77-60-B4**, também conhecido como endereço MAC) além do endereço IPv4, utilizado na rede local (neste exemplo, **192.168.1.6**, também conhecido como endereço lógico).

7.5.4 Comandos básicos funcionais

Os comandos básicos funcionais são utilizados para as tarefas do dia a dia dos usuários que utilizam o Prompt de Comando. A combinação destes comandos permite a criação de *scripts* ou arquivos executáveis que podem fazer rotinas complexas e/ou automatizadas, dentro do sistema operacional Windows Seven. Conheça agora, alguns destes comandos e como podemos começar a utilizá-los.

- Acessar o painel de controle – `control`
- Configurações de teclado – `control keyboard`

- Configurações do *mouse* – `control mouse`
- Verificar versão do Windows – `winver`
- Saber o nome do computador – `hostname`
- Alterar a data do computador – `date`
- Alterar a hora do computador – `time`
- Listar arquivos ou diretórios – `dir/p`
- Limpar a tela – `cls`

Para criar diretórios, por exemplo, podemos utilizar o comando `mkdir` ou `md`. Sua sintaxe é a seguinte:

```
mkdir nome_do_diretorio_a_ser_criado
```

Para acessar diretórios ou mostrar caminho atual: `chdir` ou `cd`. Sua sintaxe é a seguinte:

```
chdir unidade/caminho_do_diretorio
```

Para renomear arquivos ou pastas utilizamos o comando `ren`. Sua sintaxe é a seguinte:

```
ren nome_atual nome_modificado
```

Para copiar arquivos ou pastas, utilizamos o comando: `copy`. Sua sintaxe é a seguinte:

```
copy nome_do_arquivo unidade/caminho_do_diretorio
```

Para remover arquivos ou diretórios utilizamos o comando `del` (cuidado ao utilizar este comando, ele pode eliminar diretórios essenciais para o funcionamento do sistema operacional). Sua sintaxe é a seguinte:

```
del nome_do_arquivo (para apagar arquivos)
```

```
del nome_do_diretorio (para apagar diretórios)
```

Resumo

Nessa aula, vimos de forma prática algumas funcionalidades do sistema operacional Windows Seven. Entre estes recursos, foi possível entender os requisitos necessários para a instalação, o processo de instalação, as principais ferramentas do sistema, bem como, os comandos básicos funcionais para administração do sistema, via Prompt de Comando. Para fixar o conteúdo visto em cada aula, é importante que você realize os exercícios de aprendizagem. Em nossa próxima aula, outro estudo de caso será apresentado, porém, agora com o sistema operacional Linux Ubuntu.

Atividades de aprendizagem



1. Quais são os requisitos para instalação do Windows Seven?
2. Quais são os passos necessários para criação de um ponto de restauração do sistema?
3. Quais são as principais opções que temos ao acessar o gerenciador de tarefas do Windows?
4. Como podemos definir os programas que serão carregados na inicialização do sistema operacional?
5. Informe os comandos necessários para que possamos:
 - a) Corrigir erros do disco automaticamente.
 - b) Saber o endereço IP e endereço MAC do computador.
 - c) Saber o nome do computador.
6. Faça uma pesquisa na internet, sobre a evolução dos sistemas operacionais Windows, descrevendo de forma simplificada as dez últimas versões do sistema operacional Microsoft.

Aula 8 – Estudo de caso: sistema operacional Linux Ubuntu

Objetivos

Conhecer de forma prática o sistema operacional Linux Ubuntu.

Compreender seus requisitos e formas de instalação.

Conhecer as principais ferramentas de administração do sistema.

Compreender os principais comandos de administração do sistema operacional.

8.1 Considerações iniciais

O sistema operacional Linux apresenta diferentes distribuições que podem ser utilizadas por inúmeros dispositivos (assim como acontece também nos sistemas operacionais Microsoft, vistos na aula anterior). A grande diferença neste caso é que os sistemas operacionais Linux podem ser modificados, distribuídos livremente, não possuem restrições quanto à quantidade de computadores que irão utilizar o sistema operacional, entre outras características. Veremos neste estudo de caso o sistema operacional Linux Ubuntu versão 12.04 Desktop, confira.

8.2 Requisitos

Sempre antes de instalar um sistema operacional é importante analisarmos se o dispositivo que temos em mãos (computador, *notebook*, etc.) possui os requisitos mínimos necessários a instalação deste sistema. Isto evita uma série de problemas posteriores ou durante a instalação. Devemos sempre buscar esta informação no *site* da própria distribuição do sistema operacional que iremos instalar. Para o Ubuntu 12.04, os requisitos mínimos para instalação são:

- Processador de 01 GHz.
- Espaço livre de no mínimo 4,4 GB em disco rígido.
- 512 MB de memória RAM.

- Estar conectado a internet (opcional).

8.3 Instalação

A instalação do Linux Ubuntu 12.04 versão para *desktop* é relativamente simples e envolve uma série de passos básicos. Estes passos são bastante semelhantes na instalação dos sistemas operacionais em um computador (independente de fabricante). Vamos então ao passo a passo da instalação, confira.

Semelhante a outros sistemas operacionais o primeiro passo aqui é obtermos a versão do sistema operacional Linux Ubuntu 12.04. Isto é possível através do acesso ao repositório que contém as versões de tal sistema. Para fazer o *download* desta versão acessamos o seguinte endereço: <http://releases.ubuntu.com/12.04/>

Realizado o *download* da versão que queremos e ligando o computador com o CD/DVD de instalação no *drive*, iniciamos a nossa instalação. A primeira tela que aparecerá será semelhante à apresentada na Figura 8.1.



Figura 8.1: Tela inicial – instalação Ubuntu 12.04

Fonte: Ubuntu 12.04

Na tela de “Bem-vindo” temos duas opções principais: “Experimentar o Ubuntu” e “Instalar o Ubuntu”. A primeira opção (Experimentar o Ubuntu) permite ao usuário experimentar o sistema operacional, ou seja, utilizar os aplicativos, executar comandos, interagir com o sistema, sem que qualquer alteração seja feita no sistema de arquivos atual (esta opção é executada

diretamente no CD/DVD de instalação). A segunda opção (Instalar o Ubuntu) permite iniciar o processo de instalação e as etapas subsequentes. Ainda na tela inicial é necessário selecionar o idioma a ser utilizado, no caso da Figura 8.1, está selecionada a opção: “Português do Brasil”.

No passo seguinte uma tela de verificação dos requisitos do sistema é mostrada como forma de verificar se o computador em questão dispõe dos recursos necessários para instalação do sistema operacional. Na sequência uma tela mostra os adaptadores de rede instalados no computador e se o usuário pretende conectar-se a uma rede nesse momento ou não. Neste caso o usuário pode optar por não se conectar a rede nenhuma e somente fazer a instalação do sistema operacional, ou conectar-se a uma rede e já buscar as últimas atualizações existentes durante a instalação do sistema operacional.

Realizado estes dois procedimentos, chegamos a um passo importante que se refere ao tipo de instalação que queremos realizar (Figura 8.2).



Figura 8.2: Tipo de instalação – Ubuntu 12.04

Fonte: Ubuntu 12.04

Partindo de nosso estudo de caso anterior, onde mostramos a instalação do Windows Seven, vamos imaginar que temos instalado em nosso computador o sistema operacional da Microsoft (Windows Seven) e que queremos instalar também o Ubuntu 12.04, mantendo os dois sistemas instalados no computador, para utilizarmos cada um quando quisermos.

Na Figura 8.2, podemos visualizar três opções principais que são: “Instalar o Ubuntu” ao lado do “Windows Seven”, “Substituir o Windows Seven pelo Ubuntu” e “Opção Avançada”. A primeira opção (Instalar o Ubuntu ao lado

do Windows Seven) permite instalar os dois sistemas operacionais lado a lado no disco rígido, permitindo que o usuário selecione qual deles utilizar cada vez que ligar o computador. A segunda opção (Substituir o Windows Seven pelo Ubuntu) faz com que o Windows Seven seja completamente apagado do computador e substituído pelo Ubuntu (documentos, imagens, vídeos, entre outros presentes no Windows Seven, são apagados). A terceira opção (Opção avançada) permite criar partições, redimensionar, definir o espaço para cada partição, bem como gerenciar o espaço em disco da maneira que o usuário julgar necessário (recomenda-se esta opção para usuários que conheçam tais procedimentos a serem realizados).

Neste estudo de caso, selecionaremos a opção “Opção Avançada”. Para realizar a instalação é necessário criar as seguintes partições:

- / (partição barra), com o sistema de arquivos do tipo **journaling EXT4** e pelo menos, 10 GB de tamanho nesta partição.
- swap (partição área de troca), com o sistema de arquivos do tipo **área de troca** (swap) e 02 GB de tamanho.
- /home (partição home), com o sistema de arquivos do tipo **journaling EXT4** com o espaço livre que estiver disponível.

Partindo da ideia que temos o Windows Seven instalado em nosso computador, teremos a próxima tela, semelhante à apresentada na Figura 8.3.

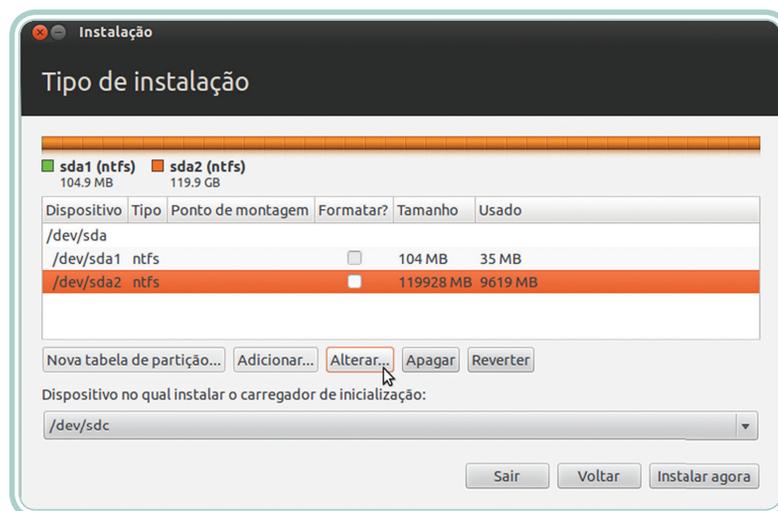


Figura 8.3: Tipo de instalação – partição Windows (NTFS) – Ubuntu 12.04
Fonte: Ubuntu 12.04

O próximo passo será selecionar a maior partição (neste caso, /dev/sda2) e clicar no botão “Alterar”. A partir deste procedimento, devemos definir um novo valor para esta partição, conforme Figura 8.4.

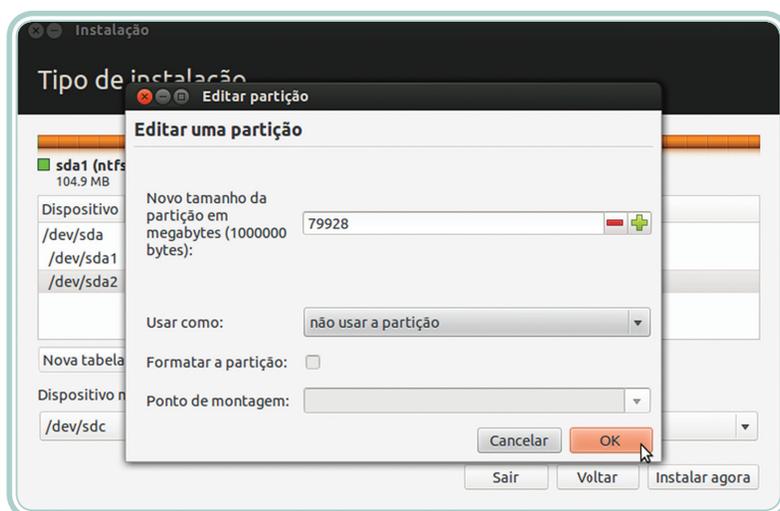


Figura 8.4: Redimensionando uma partição NTFS – Ubuntu 12.04

Fonte: Ubuntu 12.04

Repare que com esta mudança o valor da partição NTFS fica com aproximadamente 80 GB, com um espaço livre de 40 GB, conforme podemos visualizar na Figura 8.5.

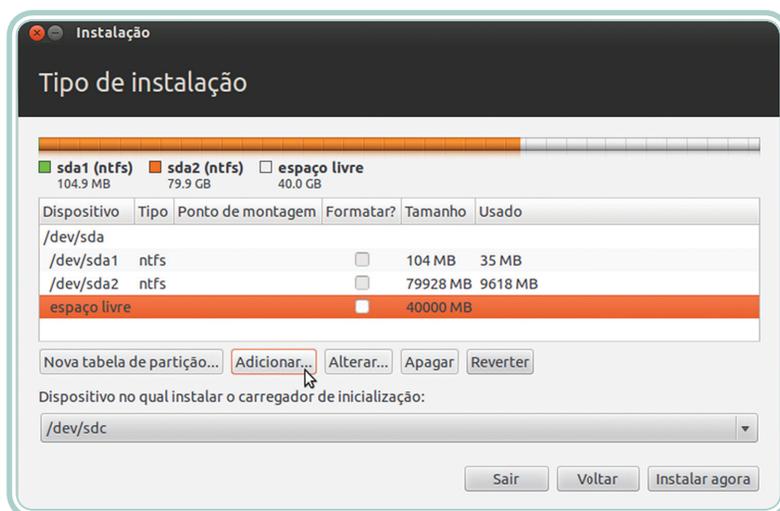


Figura 8.5: Redimensionando uma partição NTFS – Ubuntu 12.04

Fonte: Ubuntu 12.04

A partir deste passo, devemos selecionar sempre o espaço livre restante e criar as partições que necessitamos para instalação do sistema operacional. As partições a serem criadas são a “/swap” com tamanho de 02 GB (do tipo área de troca – swap), a partição “/” com tamanho de 10 GB (partição do tipo

journaling EXT4) e a partição “/home” com tamanho restante (também do tipo *journaling* EXT4). Criado estas partições, basta clicar no botão “Instalar agora”.

As próximas telas são de informações do usuário, como nome de *login* e senha, fuso-horário, *layout* de teclado, além da definição de uma imagem associada ao usuário. Feito isso, basta aguardar o termino da instalação e por fim clicar no botão “Reiniciar agora”, conforme Figura 8.6.

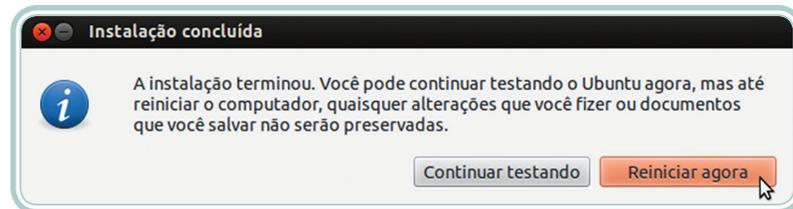


Figura 8.6: Instalação concluída – Ubuntu 12.04

Fonte: Ubuntu 12.04

8.4 Árvore de diretórios

No Linux os sistemas de arquivos como foi visto na seção de instalação, são diferentes dos utilizados no Windows. Enquanto nas versões Windows, temos os sistemas de arquivos FAT e NTFS, no Linux temos principalmente os sistemas de arquivos do tipo EXT2, EXT3, EXT4, *swap*, entre outros. A forma de organização dos arquivos e diretórios também ocorre de maneira diferente no Windows e Linux.

Nos sistemas operacionais Linux a estrutura de diretórios e arquivos pode ser visto como uma árvore, onde a raiz é o diretório barra (/), as ramificações são os subdiretórios (como os diretórios /home/etc /root, etc.) e as folhas são os arquivos. Na Figura 8.7 é possível ter uma visão geral desta estrutura, confira.



Figura 8.7: Estrutura de diretórios padrão do sistema operacional Linux

Fonte: CTISM, adaptado de <http://goo.gl/Lp7yfu>

8.4.1 Diretório raiz (/)

O diretório raiz é equivalente a unidade C:\ do Windows. Ele contém todos os demais subdiretórios que já vêm em uma instalação padrão do sistema operacional, assim como os demais que serão criados pelos usuários do sistema.

8.4.2 Diretório dos executáveis do sistema (/bin)

No diretório `/bin` temos os arquivos binários (executáveis) básicos do sistema operacional. Eles podem ser utilizados através do terminal de comandos

(geralmente o *bash*) para acionar diversas funcionalidades do sistema. Exemplos de comandos (arquivos executáveis) presentes neste diretório: *cd*, *ls*, *cat*, etc.

8.4.3 Diretório de carregamento do sistema (/boot)

O diretório */boot* armazena o núcleo do sistema Linux, isto é, o *kernel* e alguns arquivos utilizados para o *boot* do sistema. Entre os gerenciadores de *boot* mais conhecidos estão o Lilo e o Grub.

8.4.4 Diretório dos dispositivos do sistema (/dev)

No diretório */dev* estão os dispositivos presentes no computador, representados através de subdiretórios ou arquivos que apontam para os mesmos (como por exemplo */dev/sda* – representando o disco rígido, entre outros). Para poder utilizar estes dispositivos de forma efetiva faz-se necessário montar os mesmos, através de comandos específicos.

8.4.5 Diretório de configurações do sistema (/etc)

Neste diretório (*/etc*) são armazenados os arquivos de configuração do sistema. Estes arquivos em sua grande maioria recebem a extensão “.conf”. De modo geral os sistemas Linux e os programas que são instalados nele, costumam criar os arquivos de configuração com o mesmo nome dos serviços o que facilita a identificação dos mesmos.

8.4.6 Diretório dos usuários (/home)

O diretório */home* é destinado aos usuários e seus perfis. Cada vez que um novo usuário é adicionado no sistema operacional Linux uma pasta com seu nome de *login* é adicionada como subdiretório no diretório */home*. Assim seus arquivos pessoais são localizados nesta pasta.

8.4.7 Diretório de pontos de montagem (/media)

O diretório */media*, é o ponto de montagem para os diversos dispositivos que temos presentes em um computador, como discos rígidos, *drives* de CD e DVD, disco removíveis (*pen drives* e HDs externos), entre outros. Uma vez montados estes dispositivos, podemos acessar o seu conteúdo normalmente, através do diretório que contém os respectivos dados.

8.4.8 Diretório de dispositivos opcionais (/mnt)

O diretório */mnt* é uma alternativa opcional ao diretório */media*. Caso precisarmos montar um dispositivo que não tenha sido montado automaticamente em */media*, podemos utilizar uma combinação do comando “*mount*” e do diretório “*/mnt*” para montar determinado dispositivo e ter acesso a seu conteúdo.

8.4.9 Diretório de recursos do sistema (/usr)

Neste diretório /usr ficam armazenados grande parte dos aplicativos instalados no sistema, além dos principais executáveis do sistema operacional, biblioteca dos principais programas, entre eles os aplicativos gráficos.

8.4.10 Diretório dos binários dos aplicativos (/usr/bin)

O diretório /usr/bin é responsável por armazenar arquivos binários (executáveis) da maioria dos aplicativos instalados no sistema operacional. Podemos verificar onde um arquivo executável se encontra através do comando "which", via terminal de comandos do Linux.

8.4.11 Diretório de biblioteca dos aplicativos (/usr/lib)

Este diretório /usr/lib armazena as bibliotecas utilizadas pelos programas. Sua função é semelhante ao diretório C:\Windows\System32 do Windows Seven. As extensões ".a" são ditas estáticas e pertencem a um programa específico, enquanto as terminadas em ".so.versão" podem ser utilizadas por diversos programas e são classificadas como compartilhadas.

8.4.12 Diretório temporário (/tmp)

O diretório /tmp armazena conteúdo temporário de aplicativos, navegadores, instaladores, entre outros. Este conteúdo fica disponível a todo o sistema e seus usuários e é apagado cada vez que o computador é reiniciado ou desligado.

8.4.13 Diretório do sistema de arquivos e de log (/var)

O diretório de logs do sistema /var armazena informações gerais sobre determinado comportamento de um programa. Estas informações ficam armazenadas de modo que podemos saber em detalhes as operações realizadas por um programa, bem como, informações gerais sobre o mesmo. No caso de administrar e analisar determinado serviço, estes arquivos são de extrema importância.

8.5 Ferramentas de administração do sistema

O utilitário "System Settings" (configurações do sistema) do sistema operacional Linux Ubuntu 12.04 permite ter acesso aos principais recursos de configuração pessoal, de hardware e do sistema. Através dele podemos personalizar nosso computador aplicando as configurações que necessitamos em determinado momento. A Figura 8.8 permite visualizar o acesso a este recurso.

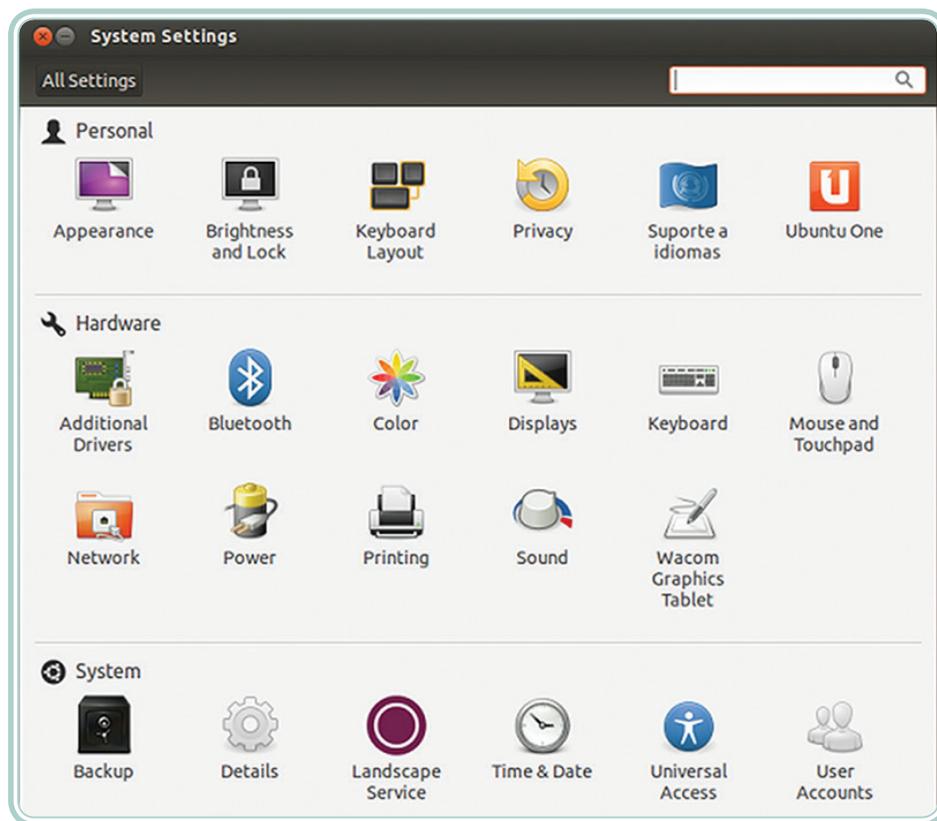


Figura 8.8: Acesso as configurações do sistema – Ubuntu 12.04

Fonte: Ubuntu 12.04

8.5.1 Configurações pessoais (*Personal*)

Nas configurações pessoais, temos algumas opções principais que são: aparência, brilho e bloqueio, configurações de teclado, privacidade, suporte a idiomas e o serviço Ubuntu One.

O recurso Aparência (*Appearance*) permite configurar opções de papel de parede, temas, planos de fundo, entre outros.

A opção Brilho e Bloqueio (*Brightness and Lock*) permitem definir as opções de brilho, bem como o tempo de bloqueio de tela que será feito após um período de inatividade do computador.

As configurações de Teclado (*Keyboard Layout*) permitem definir o tipo de teclado que será utilizado (layout), bem como suas preferências para cada usuário.

A opção Privacidade (*Privacy*) permite configurar o nível de privacidade desejado ao acessar e manipular diferentes tipos de arquivos, navegar na internet e manter informações sobre a atividade de diversos programas do usuário.

Na opção suporte a idiomas, é possível definir ou instalar um novo idioma para que o sistema operacional possa ser personalizado a linguagem padrão utilizada pelo usuário. Por padrão, o sistema operacional Linux Ubuntu em sua versão 12.04, vem com o idioma em inglês, sendo necessária a instalação de pacotes adicionais para mudança de idiomas.

8.5.2 Hardware

Nas opções de *hardware*, podemos configurar e definir padrões para os principais dispositivos de nosso computador, como por exemplo: monitor, teclado, *mouse*, rede, energia, impressora, entre outros. Vejamos a seguir as principais opções deste recurso:

- **Displays** – permite definir tamanho de tela do monitor, resolução, detectar entradas de diferentes dispositivos de visualização, entre outros.
- **Keyboard** – disponibiliza configurar o tipo de teclado, tempo de repetição, definição de teclas de atalho, entre outros.
- **Mouse and TouchPad** – possibilita a configuração de *mouse* e *touchpad*, quanto a velocidade, rastro de cursor, cliques, funções dos botões, *scroll*, etc.
- **Network** – as configurações de rede como os adaptadores presentes, endereços IPs, redes locais detectadas, bem como informações sobre o funcionamento das mesmas, podem ser obtidas através desta opção.
- **Power** – definições de energia, quanto ao uso do *notebook* ligado diretamente na corrente elétrica ou usado somente na bateria, podem ser personalizadas através desta opção, bem como, as definições de energia ao suspender, hibernar ou baixar a tampa do *notebook*.
- **Printing** – nesta opção é possível visualizar as impressoras instaladas no computador, bem como administrar as mesmas, fazendo a instalação de uma nova impressora ou excluindo uma impressora inexistente na rede.

8.5.3 Sistema (System)

As configurações de sistema permitem definir opções importantes quanto às funcionalidades e principais recursos que o sistema operacional disponibiliza e que podem ser personalizados a qualquer momento. Vejamos os principais:

- **Backup** – recurso que permite personalizar uma cópia de segurança, realizá-la e posteriormente restaurá-la, se assim for necessário. A vanta-

gem desta ferramenta é que trata-se de um recurso nativo do sistema operacional Linux Ubuntu 12.04 e que pode ser configurado sempre que necessário. A Figura 8.9, mostra as opções do recurso em questão.

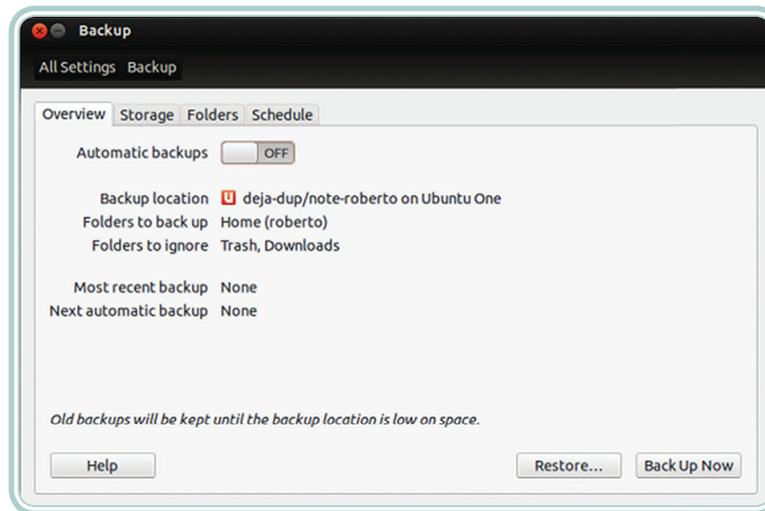


Figura 8.9: Sistema/backup – Ubuntu 12.04

Fonte: Ubuntu 12.04

- **Details** – esta opção mostra informações gerais sobre o sistema operacional, como o nome do computador, memória, processador, tamanho do disco, tipo do sistema operacional utilizado, personalização de aplicativos padrões, entre outros recursos, conforme pode ser visualizado na Figura 8.10.

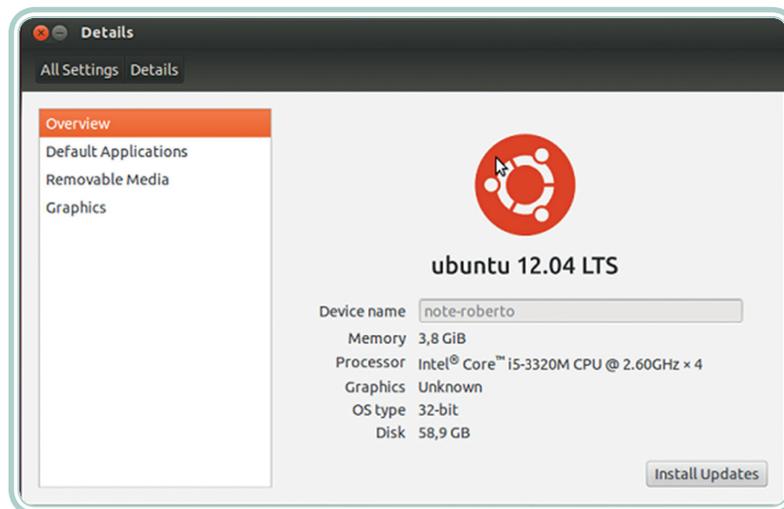


Figura 8.10: Sistema/details – Ubuntu 12.04

Fonte: Ubuntu 12.04

- **Time & Date** – permite configurar informações relativas à data e hora, bem como, definição de localização e fuso-horário. A Figura 8.11, demonstra esta opção.



Figura 8.11: Sistema/time & date – Ubuntu 12.04

Fonte: Ubuntu 12.04

- **Universal access** – a opção de acesso universal permite definir configurações que tornem o sistema operacional acessível a usuários de qualquer natureza, independente de sua limitação (visual, auditiva, entre outros). Os recursos de acessibilidade dispõem de diferentes opções personalizáveis, conforme é possível observar na Figura 8.12.

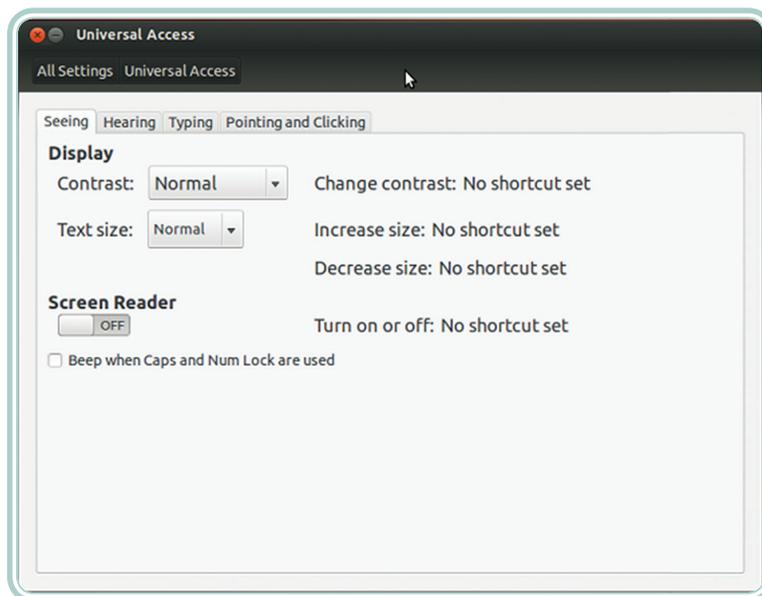


Figura 8.12: Sistema/universal access – Ubuntu 12.04

Fonte: Ubuntu 12.04

- **User accounts** – as contas de usuário permitem visualizar os usuários cadastrados no sistema operacional, bem como detalhes de sua conta como o tipo, o método de autenticação, a linguagem, o nome de *login*, entre outros. A Figura 8.13, demonstra este recurso.

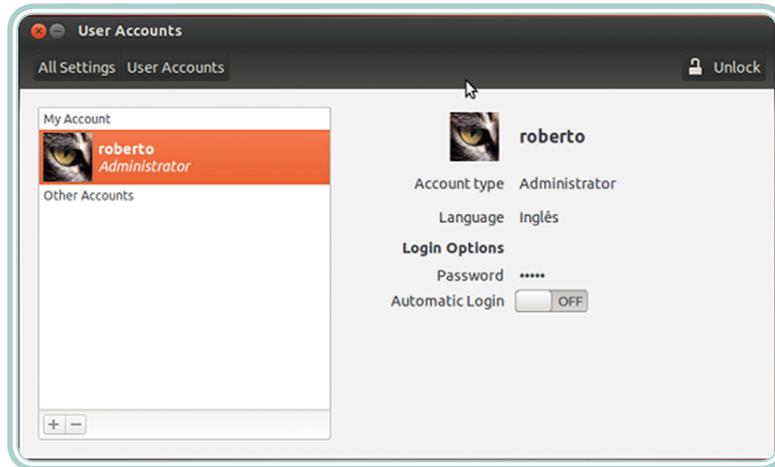


Figura 8.13: Sistema/user accounts – Ubuntu 12.04

Fonte: Ubuntu 12.04

8.6 Principais comandos de administração do sistema operacional

O objetivo desta seção final é demonstrar, de forma prática, os principais comandos Linux que podemos utilizar para administrar o sistema operacional e realizar as rotinas que necessitamos no dia a dia. Os comandos que serão apresentados abaixo são baseados e testados nas distribuições Linux Ubuntu 12.04.

8.6.1 Shell de usuário comum e root

A primeira noção de terminal de comandos que temos que aprender é quanto ao usuário que está logado (com a sessão ativa) atualmente no terminal de comandos. Obtemos esta informação ao analisarmos as seguintes linhas abaixo:

```
[aluno1@pc1 ~]$  
[root@pc1 ~]#
```

A linha 1 (“[aluno1@pc1 ~]\$”) indica que o usuário atual é um usuário comum, ou seja, com privilégios limitados no sistema, devido ao caractere “\$” estar presente em sua linha de comando.

A linha 2 (“[root@pc1 ~]#”) indica que o usuário em questão é do tipo *root* (semelhante a administrador), devido ao caractere “#” e que o mesmo tem totais controles do sistema (usuário com o maior nível de privilégios do sistema operacional).

8.6.2 Comandos de sessão

Os comandos de sessão permitem ao usuário iniciar/terminar uma sessão ou sair do *shell* (terminal de comandos). Observe os exemplos abaixo:

```
$ su
Password: (informe a senha do usuário root)
```

O comando “su” permite mudar o usuário logado, para o usuário administrador (*root*), caso necessitamos executar algum comando como usuário privilegiado (neste caso a senha do usuário *root* é necessária).

Para encerrar uma sessão de um determinado usuário conectado, digitamos:

```
$ logout
```

Para sairmos (encerrarmos) o *shell* de comandos devemos digitar comando:

```
$ exit
```

Veja abaixo uma lista de comandos úteis para reiniciar ou desligar o computador via terminal de comandos:

```
$ reboot (reinicia o computador)
```

```
$ shutdown -r now (reinicia o computador imediatamente)
```

```
$ shutdown -h 5 (desliga o computador após 5 minutos)
```

```
$ halt (desliga o computador)
```

```
$ poweroff (desliga o computador)
```

```
$ init 0 (desliga o computador)
```

Lembrando que os comandos acima devem ser digitados pelo usuário *root*.

8.6.3 Comandos de navegação

Os comandos de navegação permitem deslocar-se pelos diretórios, subdiretórios e arquivos existentes no sistema operacional, bem como obter informações. Veja os exemplos:

```
$ pwd (exibe o diretório corrente)
```

```
$ cd (muda de diretório corrente)
```

```
$ ls (lista arquivos e diretórios)
```

8.6.4 Localização de arquivos

Muitas vezes queremos buscar um arquivo que em algum momento criamos, utilizamos, mas não sabemos onde o mesmo se encontra armazenado no nosso disco rígido. Para resolver este problema podemos utilizar um comando específico (ou a combinação deste com outras opções) via terminal de comandos do Linux. Vejamos o exemplo:

```
$ find / -name teste1.txt
```

O comando em questão (find) procurará em todo o disco rígido (/) pelo nome do arquivo (opção -name) qualquer arquivo que tiver o nome "teste1.txt". Desta forma é possível localizar sua exata localização.

8.6.5 Manipulando arquivos e diretórios

Para que possamos manipular arquivos e diretórios de uma forma rápida e objetiva, necessitamos saber os comandos que mais são utilizados para este fim. Vejamos os exemplos abaixo:

```
$ mkdir pasta (cria um diretório chamado de "pasta")
```

```
$ rm -rf pasta/* (apaga somente o conteúdo do diretório "pasta"  
mantendo a mesma vazia)
```

```
$ rm -rf pasta (apaga todo o diretório "pasta", subdiretórios e arquivos  
existentes)
```

```
$ cp -rf pasta /tmp (copia o conteúdo do diretório "pasta" para dentro  
do diretório "/tmp" )
```

```
$ mv aula1.txt pasta (move o arquivo "aula1.txt" para dentro do
diretório "pasta")
```

```
$ mv aula3.txt teste.txt (renomeia o arquivo aula3.txt para teste.txt)
```

A prática dos comandos acima é fundamental para que tenhamos um conhecimento sobre a forma de manipular tais comandos, suas opções, etc.

8.6.6 Informações do sistema

Às vezes necessitamos saber informações dos sistemas, seja para instalar um aplicativo, fazer uma atualização, entre outros. Existem comandos simples que uma vez executados, podem trazer as informações que estamos buscando. Veja agora, alguns deles:

```
$ date (exibe data e hora atual)
```

```
$ cal (exibe o calendário atual)
```

```
$ hostname (exibe o nome do computador)
```

```
$ uptime (exibe a quanto tempo o sistema operacional está em
funcionamento, desde a última reinicialização)
```

```
$ uname (exibe informações sobre o kernel, processador, plataforma de
hardware e sistema operacional).
```

Resumo

Nessa aula, vimos de forma prática o sistema operacional Linux Ubuntu em sua versão 12.04. Foi possível conhecer os requisitos e processo de instalação, como o sistema operacional Linux se organiza (árvore de diretórios), as principais ferramentas para administração do sistema, bem como, os principais comandos de administração do sistema operacional. Para fixar o conteúdo visto em cada aula, é importante que você realize os exercícios de aprendizagem. Não se esqueça de ler cada aula sempre que julgar necessário e revisar seu conteúdo.



Atividades de aprendizagem

1. Quais são os requisitos mínimos para instalação do Linux Ubuntu 12.04?
2. Qual a diferença entre as opções “Experimentar” e “Instalar o Ubuntu”?
3. Quanto aos tipos de instalação, explique:
 - a) Instalar o Ubuntu ao lado do Windows Seven.
 - b) Substituir o Windows Seven pelo Ubuntu.
 - c) Opção avançada.
4. Cite e explique três diretórios do sistema operacional Linux que você achou mais interessante.
5. Quais comandos seriam necessários para:
 - a) Criar uma pasta.
 - b) Copiá-la para dentro do diretório /tmp.
 - c) Excluí-la.
6. Como faço para encontrar um arquivo com o nome teste.txt?
7. Faça uma pesquisa na internet sobre os principais comandos de administração do Linux, citando e explicando dez (10) comandos diferentes dos apresentados nesta aula.

Referências

ANDRADE, Denise de Fátima. **Windows 7 – Simples e rápido**. Santa Cruz do Rio Pardo, SP: Viena, 2010.

BONIATI, Bruno; PREUSS, Evandro; FRANCISCATTO, Roberto. **Introdução à informática**. Santa Maria, RS: Colégio Agrícola de Frederico Westphalen, 2014.

CARDOSO, Ramon. Como formatar o PC e instalar o Windows 7. **Techtudo**. 9 fev. 2001. Disponível em: <<http://www.techtudo.com.br/dicas-e-tutoriais/noticia/2011/02/como-formatar-o-seu-computador-e-instalar-o-windows-7.html>>. Acesso em: 8 ago. 2014.

MAIA, Luiz Paulo. **Introdução à arquitetura de sistemas operacionais**. São Paulo: LTC, 2007.

MICROSOFT. **Requisitos do sistema do Windows 7 – Ajuda do Windows**. 12 maio 2010. Disponível em: <<http://windows.microsoft.com/pt-br/windows7/products/system-requirements>>. Acesso em: 13 ago. 2014.

PEREIRA, Ana Paula. MS-DOS completou 30 anos de existência. **Tecmundo**. 28 jul. 2011. Disponível em: <<http://www.tecmundo.com.br/sistema-operacional/11968-ms-dos-completou-30-anos-de-existencia.htm>>. Acesso em: 13 jun. 2014.

SILBERSCHATZ, A.; GAGNE, G.; GALVIN, P. B. **Fundamentos de sistemas operacionais**. Rio de Janeiro: LTC, 2004.

MENDONÇA, Tales Araujo; MARTINI, Luciano Andress. **GNU Linux – Aprenda a operar o sistema na prática**. Santa Cruz do Rio Pardo, SP: Viena, 2009.

TANENBAUM, Andrew S. **Sistemas operacionais modernos**. Rio de Janeiro: Prentice-Hall, 2003.

MUNDO UBUNTU. **Instalação do Ubuntu 12.04**. 02 set. 2012. Disponível em: <<http://www.mundoubuntu.com.br/tutoriais/instalacao/77-instalacao-do-ubuntu-12-04-mostrando-manualmente-a-criacao-das-particoes-swap-e-home-em-dual-boot-com-o-windows-7>>. Acesso em: 17 jul. 2014.

VELLOSO, Fernando de Castro. **Informática conceitos básicos**. Rio de Janeiro: Elsevier, 2011.

Currículo do professor-autor



Adriana Soares Pereira é natural de Cruz Alta-RS. Graduada em Informática pela UNIJUI (Universidade Regional do Noroeste do Estado do Rio Grande do Sul), Mestre em Ciência da Computação pela UFRGS/RS (Universidade Federal do Rio Grande do Sul) e Doutora em Ciência da Computação pela UFRGS/RS (Universidade Federal do Rio Grande do Sul). Atualmente é professora em regime de dedicação exclusiva da Universidade Federal de Santa Maria, lotada no Colégio Agrícola de Frederico Westphalen. Atua nos cursos: Técnico em Informática, Graduação Tecnológica em Sistemas para Internet e Especialização em Gestão de Tecnologia da Informação. Atua principalmente nos seguintes temas: inteligência artificial, sistemas multiagentes, informática na educação e educação a distância.



Elisa Maria Vissotto é natural de Frederico Westphalen-RS, graduada em Ciência da Computação pela URI (Universidade Regional Integrada do Alto Uruguai e das Missões) e Especialista em Tecnologias para Aplicações WEB pela UNOPAR (Universidade Norte do Paraná). Atualmente é professora no curso Técnico em Informática Concomitante, no Colégio Agrícola de Frederico Westphalen e tutora de sala no curso Superior em Análise e Desenvolvimento de Sistemas na UNOPAR/FW.



Roberto Franciscatto é natural de Frederico Westphalen-RS. Graduado em Informática pela URI-FW (Universidade Regional Integrada do Alto Uruguai e das Missões campus de Frederico Westphalen) e Mestre em Computação Aplicada pela UNISINOS/RS (Universidade do Vale do Rio dos Sinos). Atualmente é professor em regime de dedicação exclusiva da Universidade Federal de Santa Maria, lotado no Colégio Agrícola de Frederico Westphalen. Atua nos cursos: Técnico em Informática, Graduação Tecnológica em Sistemas para Internet e Especialização em Gestão de Tecnologia da Informação. Atua principalmente nos seguintes temas: sistemas operacionais, segurança da informação, projeto e instalação de servidores, redes de computadores e desenvolvimento de aplicativos para dispositivos móveis.