

SSC0800 - ICC1 – Teórica

Introdução à Ciência da Computação I

**Conceitos básicos
sobre computadores**

Prof. Claudio Fabiano Motta Toledo

Apresentação utiliza transparências elaboradas
pelo prof. Vanderlei Bonato.

Sumário

- ✓ O que é um computador e onde podemos encontrá-los?
 - ✓ Divisão: Hardware (HW) e Software (SW)
 - ✓ Sistema Operacional (SO)
 - ✓ Representação Interna de dados
 - ✓ Linguagem de programação
 - ✓ Compiladores
-

Definição

O que é um computador?

Segundo dicionário de Cambridge: “an electronic machine which is used for storing, organizing and finding words, numbers and pictures, for doing calculations and for controlling other machines”

Onde podemos encontrá-los?

Hardware

- ✓ É a parte física do computador, que inclui os componentes eletrônicos, placas de circuito impresso, gabinete, teclado, mouse, monitor e etc.
 - ✓ É a parte que podemos “tocar!”
 - ✓ Veja a seguir os componentes de hardware normalmente encontrados num PC (*Personal Computer*)
-



Case



CD-ROM
DVD-ROM
CDRW
DVD +RW



CPU or processor



Case Fan



CPU
Fan



Hard
Drive



Keyboard
Mouse



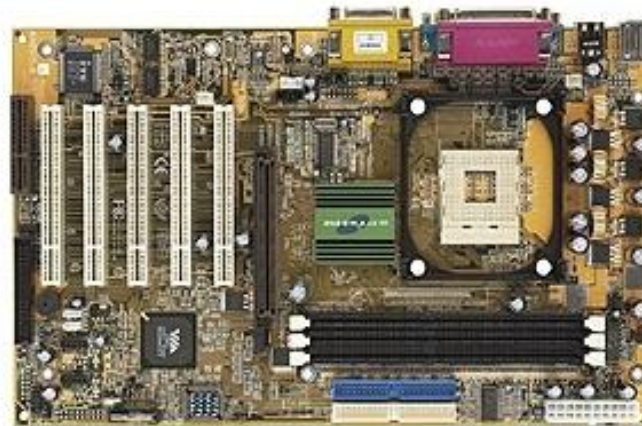
Memory



Modem



Monitor



Motherboard



Network card
NIC



Power
Supply



Sound card



Video Card

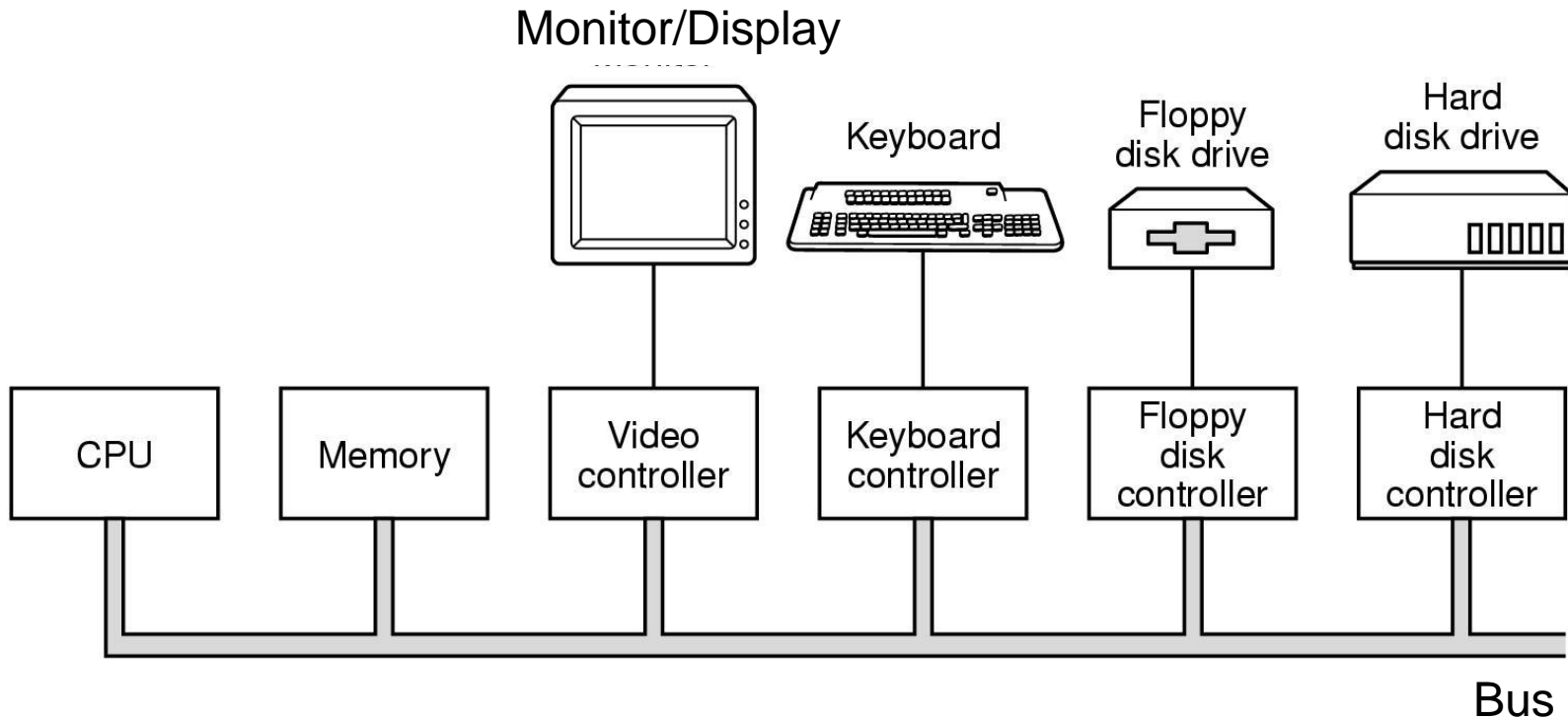


Speakers



Zip Drive

Estrutura genérica de um PC



Software

- ✓ São instruções que controlam o hardware de modo a realizar tarefas determinadas por um “algoritmo”. O conjunto dessas instruções implementado numa linguagem computacional é denominado “programa”
 - ✓ As instruções utilizam o hardware para realizar operações sobre dados armazenados na memória do computador
-

Software

- ✓ Há instruções de entrada e saída, de operações lógicas e aritméticas, de teste condicional, de escrita e leitura de dados na memória do computador, de atribuição de valores, de configuração do sistema e etc.
-

Tipos de Software

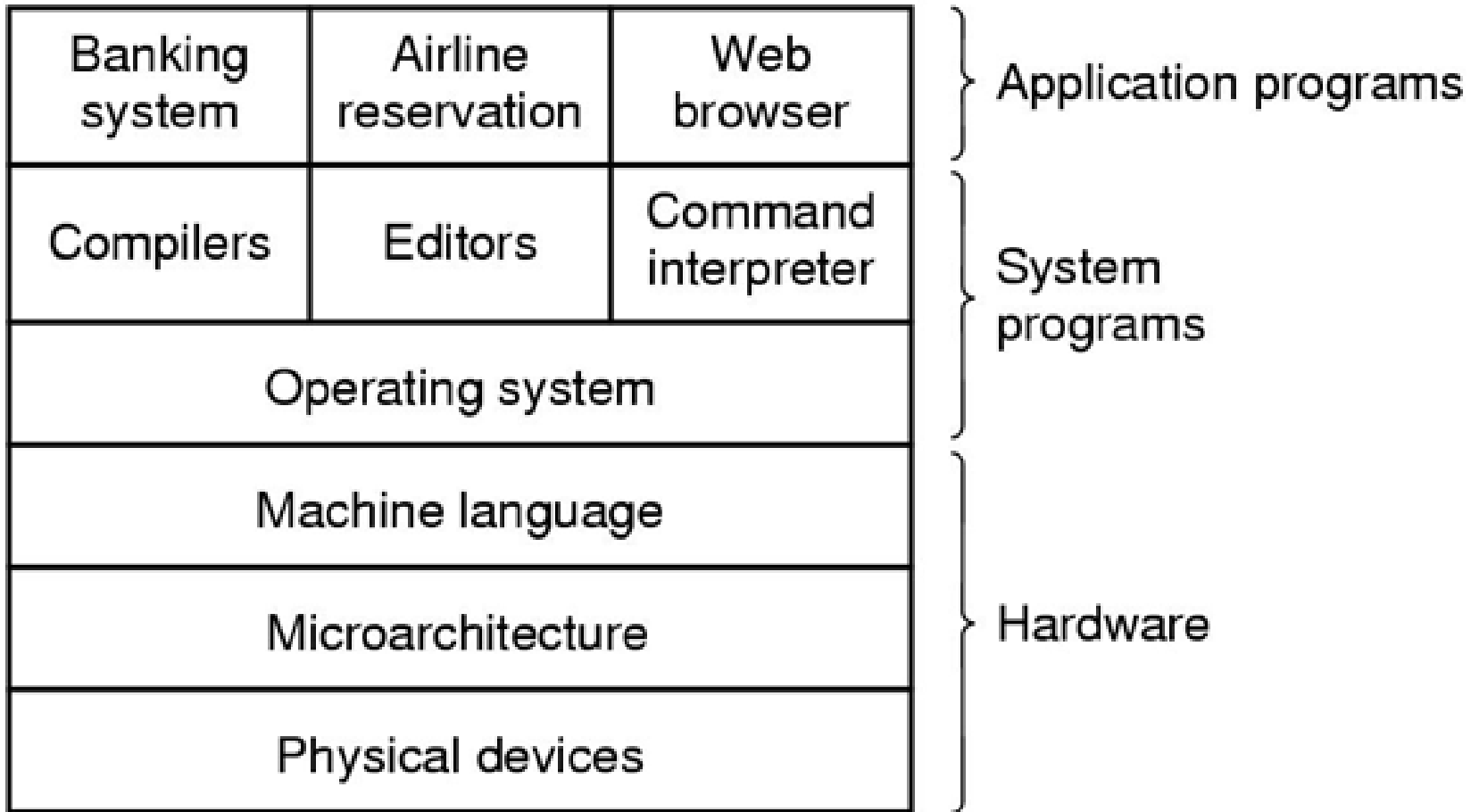
✓ **Software/programa aplicativo**

- ✓ São programas utilizados diretamente pelo usuário para executar uma ou mais tarefas específicas
 - ✓ Também conhecido como programa de usuário, APP (Application Software) ou simplesmente aplicação
 - ✓ Exemplos:
 - ✓ Editor de texto e imagem
 - ✓ Tocador de vídeo e áudio
 - ✓ Jogos
 - ✓ ..
-

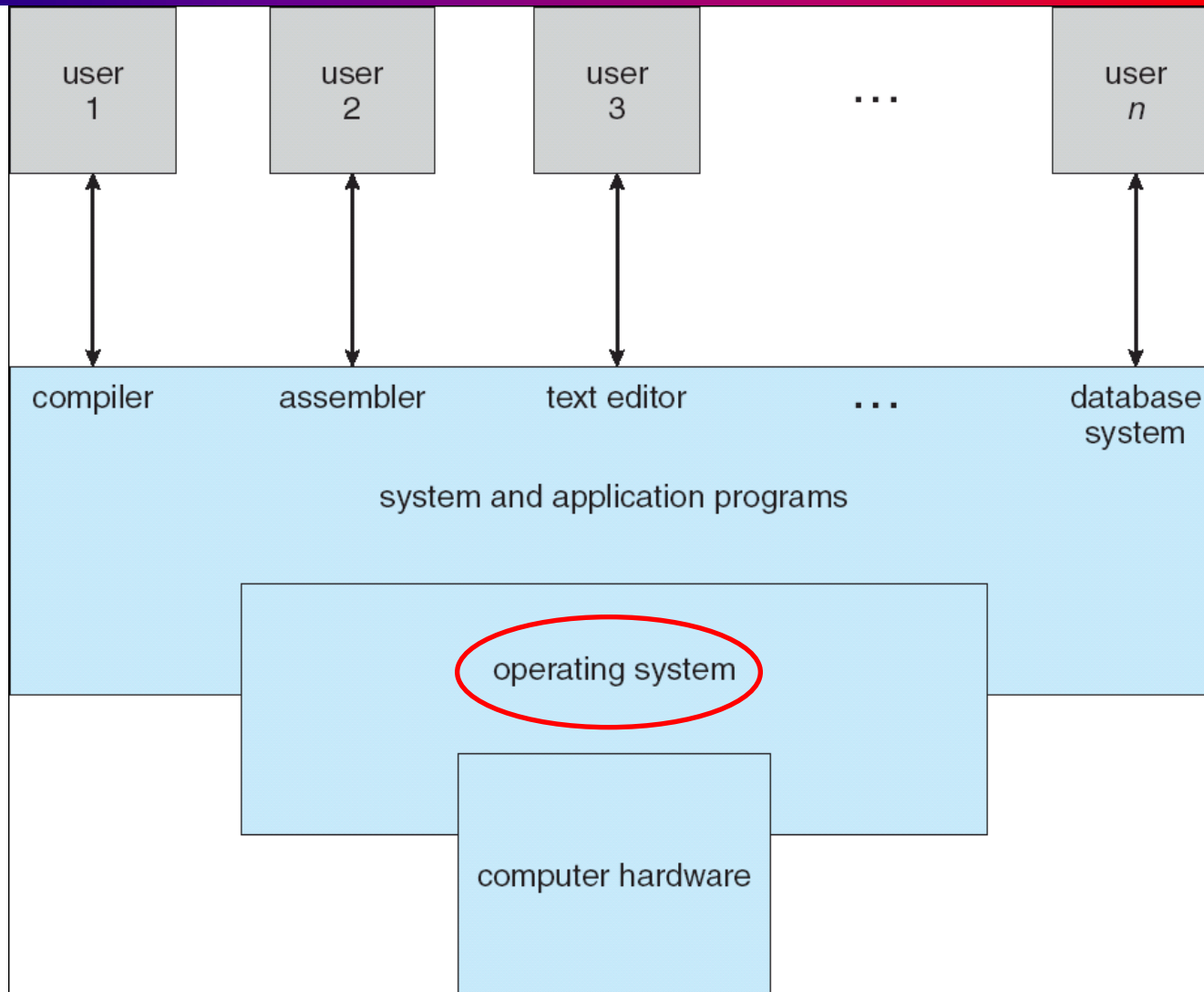
Tipos de Software

- ✓ **Software/programa de sistema**
 - ✓ Abrange todos os programas voltados ao gerenciamento e controle dos recursos de hardware do computador
 - ✓ Serviços providos pelo SO
 - ✓ Por exemplo:
 - ✓ Ferramentas de suporte ao desenvolvimento de programas (ex. compilador e montador, ligador)
 - ✓ Gerenciador de arquivos
 - ✓ Serviços de comunicação
 - ✓ ...
-

Camadas de software e hardware



Sistema Operacional



Sistema Operacional

- Controla e coordena o uso do hardware entre várias aplicações e usuários
 - SO é um alocador de recursos
 - ✓ Gerencia todos os recursos
 - ✓ Resolve conflitos de requisições para o uso eficiente e justo dos recursos de hardware
 - SO é um programa de controle
 - ✓ Controla a execução de programas para prevenir a ocorrência de erros e uso impróprio do computador
-

Sistema Operacional: MS/DOS

- ✓ Desenvolvido pela Microsoft no início dos anos 80 para atender a linha IBM-PC
 - ✓ Sistema monousuário com arquitetura pouco estruturada
 - ✓ Não dispõe de rotinas para gerenciar compartilhamento de recursos de processamento
-

Sistema Operacional: UNIX

- Criado em 1969 no laboratório da Bell
 - Considerado robusto e estável (ex. IBM AIX, Sun Solaris e HP-UX)
 - Trata-se de um sistema multiusuário que dispõe de rotinas de gerenciamento de processos e memória
 - Como a maioria dos SOs, é constituído de duas partes: programas de sistema e *Kernel* (núcleo)
-

Sistema Operacional: Linux

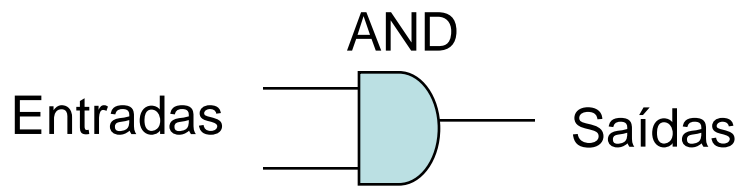
- Iniciado por Linus Torvalds como uma alternativa mais barata e prática ao sistema UNIX
 - Código fonte sob licença GPL (software livre)
 - Atualmente é o SO mais utilizado em servidores
-

Outros Sistemas Operacionais

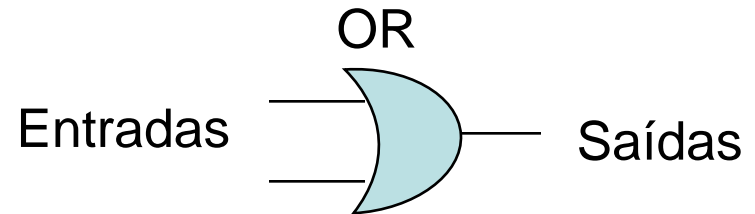
- Microsoft - Windows
 - Windows Mobile
 - Google Chrome OS (Linux-based)
 - Android (voltado para sistemas embarcados)
 - Apple - Mac OS X (UNIX-based) – Apple
 - iOS (iPhone OS)
 - IBM i, z/OS, z/VM, z/VSE (SOs da IBM)
-

Representação interna

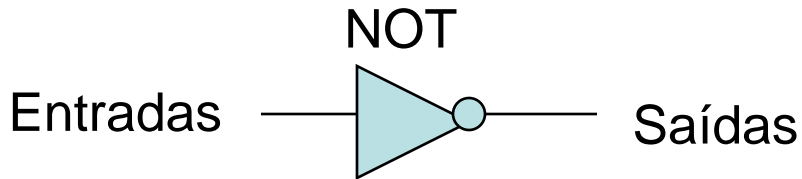
- A informação é representada através de padrões de bits.
- Operações booleanas manipulam valores verdadeiros ou falsos.
- Porta lógica é um dispositivo que fornece a saída de uma operação booleana a partir de suas entradas.



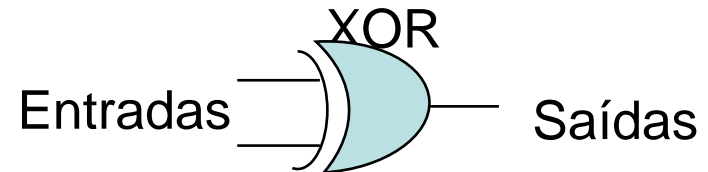
Entradas	Saídas
0 0	0
0 1	0
1 0	0
1 1	1



Entradas	Saídas
0 0	0
0 1	1
1 0	1
1 1	1



Entradas	Saídas
0	1
1	0



Entradas	Saídas
0 0	0
0 1	1
1 0	1
1 1	0

- **Flip-flop** é um circuito cuja saída apresenta um dos dois valores binários, assim permanecendo até que ocorra alguma alteração em suas entradas.
 - Ideal para o armazenamento de bits já que podem ser facilmente observado e alterado por outros circuitos.
 - **Flip-flops** podem ser miniaturizados a tal ponto que milhões deles caibam em uma única lâmina de silício de tamanho não superior a uma moeda.
-

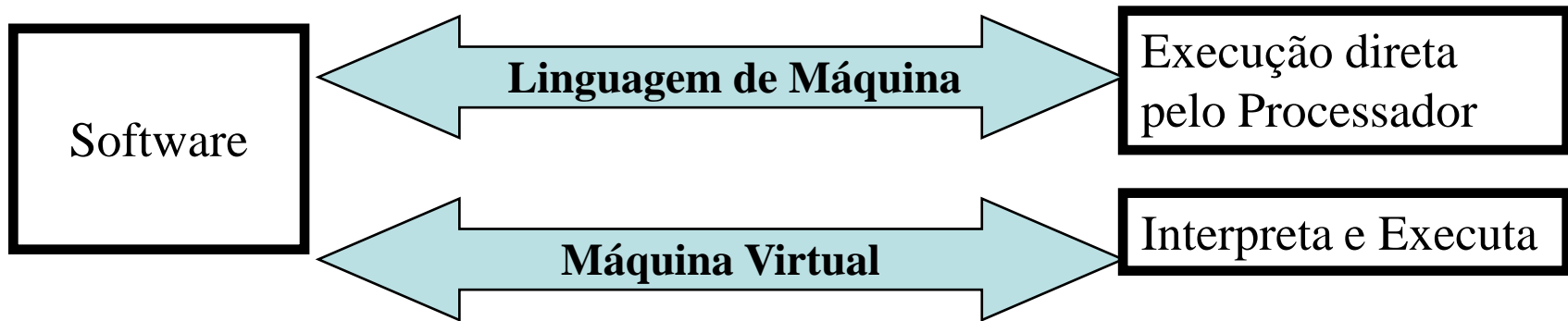
Representação interna

- Células ou posições de memórias são as unidades manipuláveis em que estão organizados os circuitos de armazenamento da memória principal.
- Cada célula tem um byte ou oito bits .
- Mega byte: $2^{20} = 1.048.576$ bytes.

Célula 0	Célula 1	Célula 2	Célula 3	...	Célula n
10111010	00111101	11110001	01010010	...	00000111

- Se a memória estiver dividida em células de tamanho de 1 byte, uma cadeia de 16 bits é armazenada em duas células consecutivas.
 - Cada célula é individualmente referenciada, acessada e modificada.
 - Processamento em qualquer ordem - Memória de acesso aleatório (Random Access Memory – RAM).
-

Linguagem de programação



- ✓ Um programa pode ser executado por qualquer dispositivo capaz de interpretar e executar as instruções de que é formado.
 - ✓ Máquinas programáveis: computador, telefone celular, máquinas de automação industrial, calculadora, etc.
 - ✓ Linguagens de programação foram projetadas para se aproximar das linguagens usadas por seres humanos.
 - ✓ Raramente linguagem de máquina é usada para desenvolver um programa, porém, algumas vezes, para aumentar o desempenho, parte de um programa pode ser desenvolvido dessa forma.
-

Linguagem de programação

- ✓ **Linguagens de primeira geração:** representadas pelas linguagens de máquina.
 - Programadores escreviam em papel os programas e depois traduziam para utilização em máquinas. Posteriormente, o processo de tradução foi identificado como um procedimento a ser executado pela máquina.
 - ✓ **Linguagens de segunda geração:** representadas pelas linguagens de montagem.
 - O uso de mnemônicos foi formalizado como uma **linguagem de montagem** (*assembly language*).
 - Um **programa montador** (*assembler*) foi desenvolvido para **traduzir** os programas escritos em linguagem de montagem.
 - O programa recebeu o nome de montador porque sua tarefa era a de montar instruções de máquinas a partir de códigos de operação obtidos da conversão de mnemônicos e identificadores.
-

Linguagem de programação

- ✓ **Linguagens de terceira geração:** suas primitivas eram de nível mais alto, e independentes da máquina.
 - Cada primitiva foi projetada para ser implementada através de uma seqüência de primitivas de baixo nível, disponíveis nas linguagens de máquina.
 - Por exemplo, **assign TOTAL the value VALOR + IMPOSTO** expressa uma atividade de alto nível sem descrever de que forma uma máquina executará tal tarefa.
 - Uma vez identificado este conjunto de primitivas de alto nível, um **programa tradutor** (*translator*) converte os programas, escritos em linguagem de alto nível, em programas de linguagem de máquina.
 - O **tradutor** se diferencia dos **montadores** de segunda geração ao agregar (**compilar**) várias instruções de máquina, na forma de pequenas seqüências, para simular a atividade representada por uma única primitiva de alto nível. Por isso, os programas tradutores são conhecidos pelo nome de **compiladores**.
-

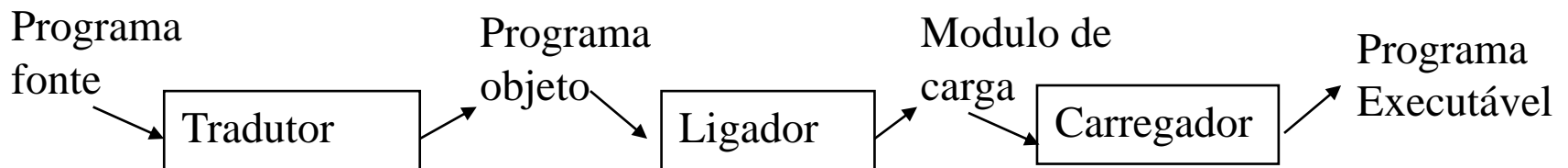
Linguagem de programação

- ✓ **Linguagens de quarta geração:** pacotes de software que permitem a seus usuários personalizar o *software* para suas aplicações sem o auxílio de especialistas.
 - Programar em tais linguagens, em geral , significa selecionar opções, apresentadas na tela do monitor em forma de frases ou ícones.
 - Planilhas eletrônicas, sistemas de banco de dados, softwares gráficos, processadores de textos, etc.

 - ✓ **Linguagens de quinta geração:** usada em referência ao conceito de programação declarativa com ênfase em programação lógica.
 - A idéia é permitir ao usuário do computador resolver um problema concentrando-se apenas na análise de suas características, em vez de determinar a forma como ele deve ser resolvido.
-

Compilador

- ✓ O compilador faz a tradução de um programa em linguagem de alto nível para um programa em linguagem de máquina.
- ✓ Transforma um programa em determinada linguagem (programa-fonte) em outro programa semanticamente equivalente em uma linguagem diferente (programa-objeto).



- ✓ O **programa-objeto** é, de fato, um programa em **linguagem de máquina** que contém algumas “pontas soltas”, as quais devem ser devidamente conectadas a outros programas-objetos, antes de se conseguir um programa executável.
-

Compilador

- ✓ O **programa ligador** (*linker*) executa a tarefa de construir tais conexões. Ele une **programas-objetos**, **rotinas do sistema operacional** e outros **softwares utilitários**, para produzir um **programa completo (módulo de carga)**. Esse programa é armazenado na forma de arquivo, no sistema de armazenamento de massa do computador.
 - ✓ O **programa carregador** (*loader*) transfere para a memória o módulo de carga para que ele possa ser executado.
 - ✓ Uma vez realizados os passos de **tradução e ligação**, o programa pode ser repetidamente **carregado e executado**, sem a necessidade de retornar à versão original.
 - ✓ No entanto, se for preciso executar alguma alteração no programa, esta deve ser feita no **programa-fonte**, o qual, por sua vez, é novamente traduzido e ligado, resultando um novo módulo de carga, que incorpora as alterações efetuadas.
-

Referências

- Fedeli, R.D.; Polloni, E.G.F.; Peres, F.E. Introdução à Ciência da Computação. 2ª Edição. São Paulo: Cengage Learning, 2010. 250p.
 - Brookshear, J.G. *Ciência da Computação: uma visão abrangente*. 5ª Edição. Porto Alegre: Bookman, 2000. 500p.
 - Silberschatz, A.; Galvin, P. B.; Gagne, G.; Operating System Concepts. 7ª Edição. USA, 2004, 921p.
-

FIM Aula 1
