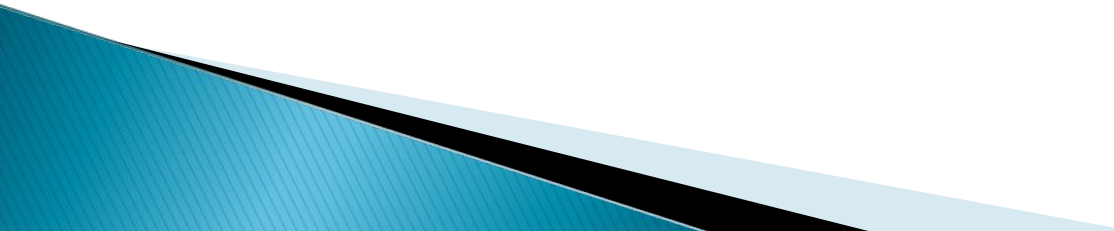


Hardware, sistemas operacionais, software básico, linguagens, compilação e representação interna dos dados

Delano M. Beder
ACH2001
EACH – USP

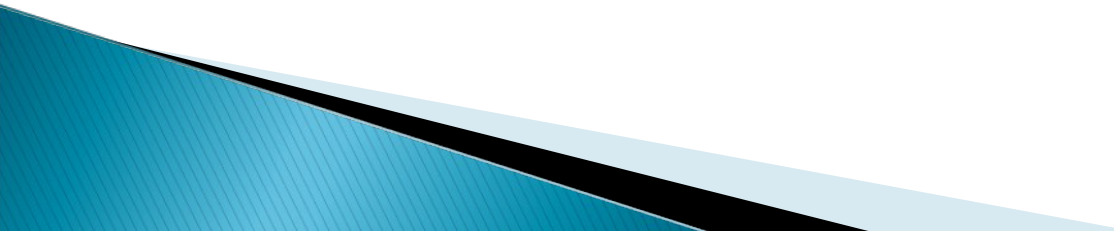
Quais as novidades desta aula?

- ▶ Mais sobre a arquitetura de Von Neumann.
 - ▶ Linguagens, compiladores e interpretadores.
 - ▶ Representação interna dos dados.
- 

Computadores

- ▶ Revolução Industrial → maior capacidade para a produção de bens materiais.
- ▶ Revolução da Informática → maior capacidade para armazenar e manipular grandes quantidades de dados e realizar cálculos de forma rápida. → maior capacidade mental.

Computadores

- ▶ Mas o computador é apenas uma máquina de calcular? O que o diferencia?
 - ▶ Sim, porém o computador é *programável* → permite realizar tarefas complexas.
 - ▶ Essas tarefas precisam ser descritas em linguagem apropriada de como deve ser realizada.
- 

Algoritmos e programas

- ▶ Algoritmos e programas → conceitos não restritos à computação.
- ▶ Tarefas algorítmicas:
 - Cozinhar;
 - Montar brinquedos;
 - Realizar cálculos matemáticos;
 - Tocar uma música em um instrumento musical.

Algoritmos e programas

- ▶ Algoritmos:
 - Seqüências finitas de regras ou instruções que especificam como determinadas operações básicas, executáveis mecanicamente, devem ser combinadas para a realização de tarefas desejadas.

Algoritmos e programas

- ▶ Algoritmos:
 - Receitas de cozinha, instruções de montagem, regras de cálculo e partituras musicais.
- ▶ Programa é uma notação para um algoritmo:
 - Receita de cozinha pode ser denotada por:
 - um texto;
 - um áudio;
 - um vídeo;

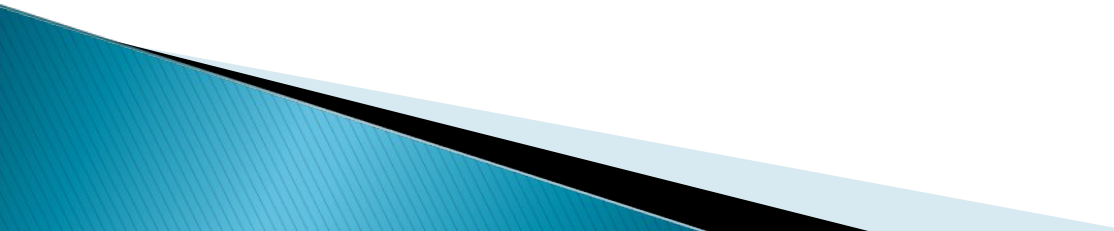
Algoritmos e programas

- ▶ Os programas, por sua vez, são descritos em diferentes linguagens:
 - Receita de cozinha:
 - ☐ um texto → língua escrita;
 - ☐ um audio → língua falada;
 - ☐ um vídeo → língua falada e imagens.
- ▶ As linguagens possuem diferentes níveis de expressividade.

Algoritmos e programas

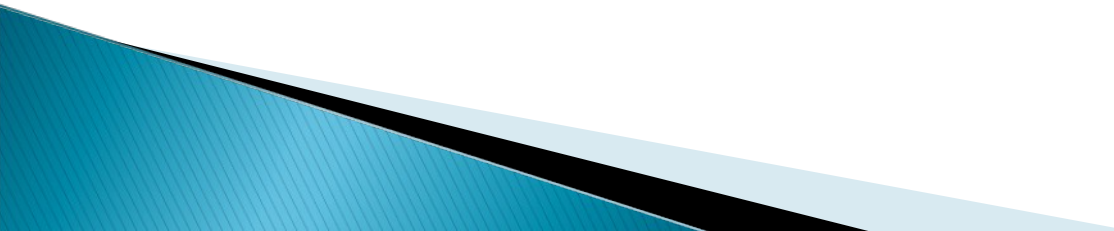
- ▶ O que diferencia os conceitos de *algoritmo* e *programas* em computação?
- ▶ Algoritmos:
 - tarefas automatizáveis por computadores.
- ▶ Programas:
 - Denotam tarefas automatizáveis de forma a serem realizadas por computadores.

Em resumo...

- ▶ A descrição das tarefas a serem realizadas pelos computadores é feita por meio de *programas*.
 - ▶ Os programas, por sua vez, são descritos utilizando *linguagens de programação*.
 - ▶ A idéia ou processo abstrato que esse programa representa é chamado de *algoritmo*.
- 

Organização de computadores

► Computadores:

- Capaz de executar apenas um pequeno número de operações básicas bastante simples – carregar um valor; somar, dividir, multiplicar valores;
 - Porém, em princípio, pode ser usado para resolver qualquer problema cuja solução pode ser obtida por meio de um algoritmo.
 - Conjunto reduzido de instruções para combinação dessas operações básicas é suficiente para expressar qualquer algoritmo computável.
- 

Organização de computadores

- ▶ Bohm e Jacopini (1966):
 - Mostraram que apenas três estruturas de controle são necessárias para representar um *algoritmo*:
 - ❑ Seqüência: ação 1; ação 2;
 - ❑ Seleção: seleção de um dentre vários cursos de ações de acordo com uma *condição*;
 - ❑ Repetição: repetição de várias ações ou seqüências de ações de acordo com uma *condição*.

Organização de computadores

- ▶ Dispositivos eletrônicos atuais são capazes de distinguir dois tipos de sinais elétricos:
 - Passagem de corrente em um sentido;
 - Passagem de corrente em sentido inverso.
- ▶ Ou seja, permitem distinguir um alfabeto com apenas dois símbolos:
 - 1 (um) e 0 (zero).

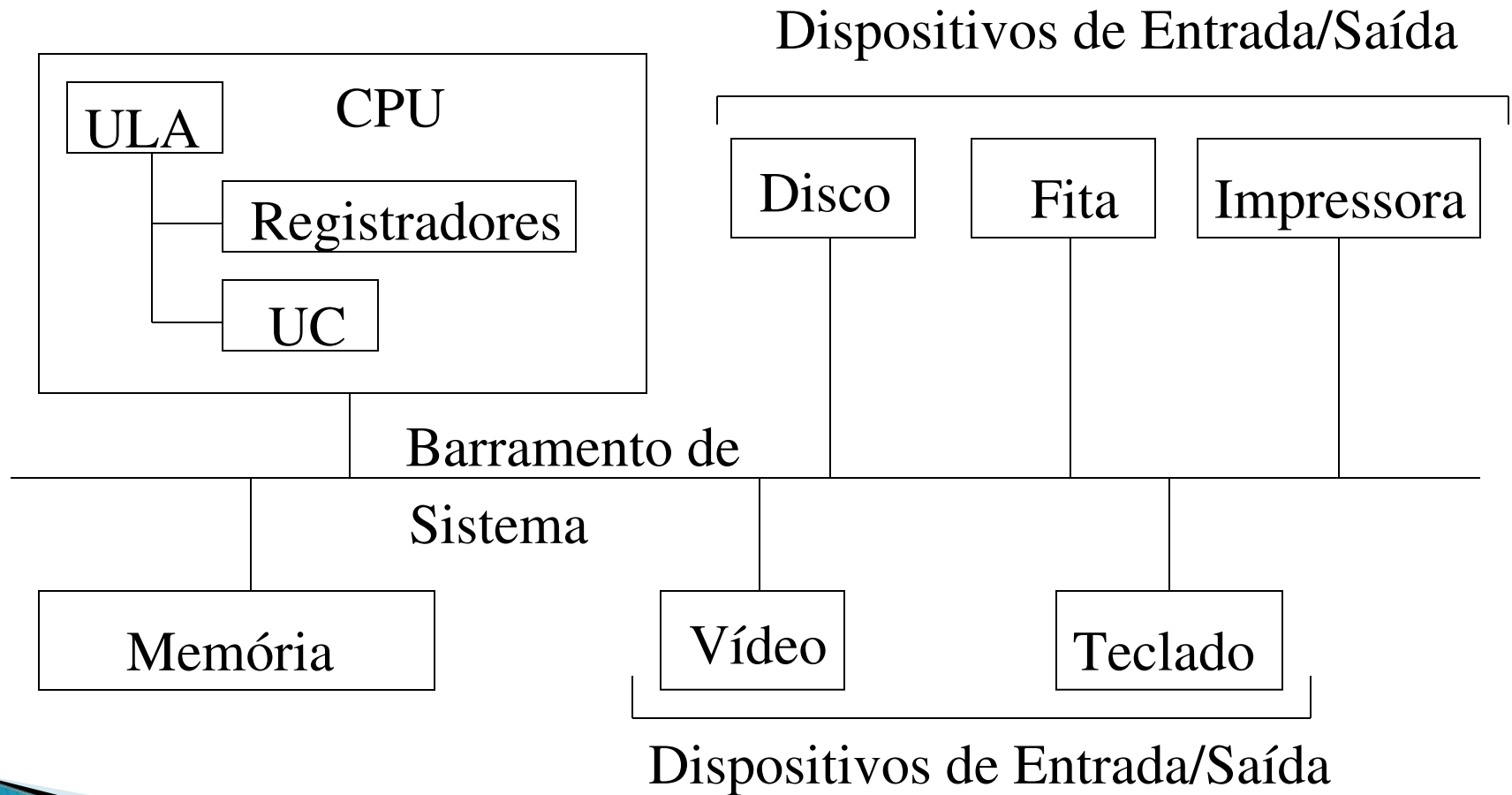
Organização de computadores

- ▶ Dados e operações são representados em um computador por meio da linguagem binária de *zeros e uns*.
 - 10101101010000 pode ser usado para representar a operação de soma de dois números.
- ▶ Números são representados, em um computador, usando notação arábica, no sistema de numeração de base 2 (falaremos mais disso à frente).

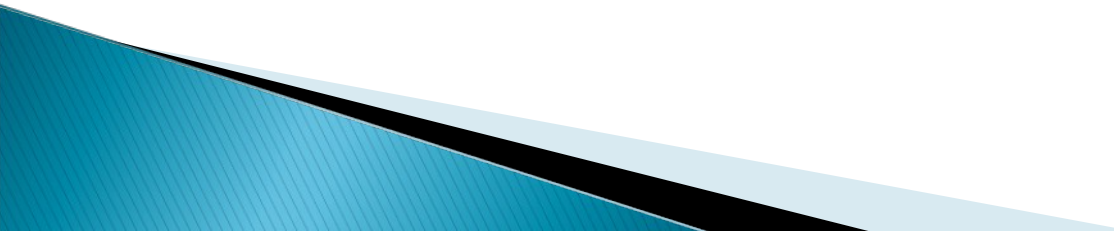
Organização de computadores

- ▶ A linguagem constituída pelas instruções executadas pelo computador, representadas na forma de seqüências de bits, é chamada de *linguagem de máquina*.
- ▶ Apesar do avanço da tecnologia de construção de computadores, todo computador moderno mantém a organização básica da arquitetura de Von Neumann.

Organização de computadores



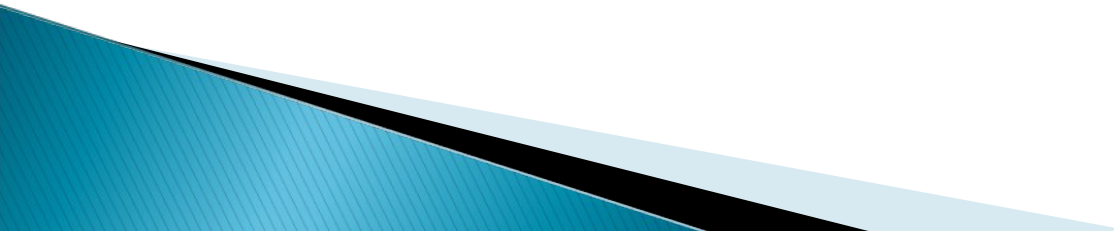
Processador ou unidade central de processamento

- ▶ Executa as instruções de um programa expresso em linguagem de máquina.
 - ▶ Interpreta a instrução corrente, executa a operação especificada nessa instrução;
 - ▶ Determina qual a próxima instrução que deve ser executada.
 - ▶ CPU – *Central Processing Unit*.
- 

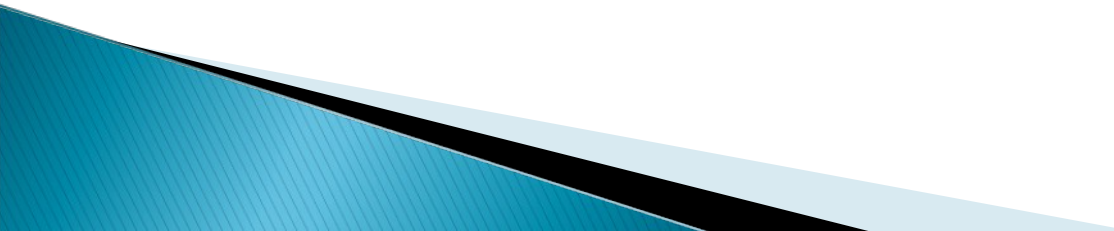
Unidade de controle

- ▶ Comanda a operação dos demais componentes do computador.
- ▶ Interpreta cada instrução do programa que está sendo executado e estabelece a ordem de execução dessas instruções.
- ▶ Usa pelo menos dois registradores:
 - Um para armazenar a instrução corrente;
 - Outro para armazenar o endereço da próxima instrução a ser executada.

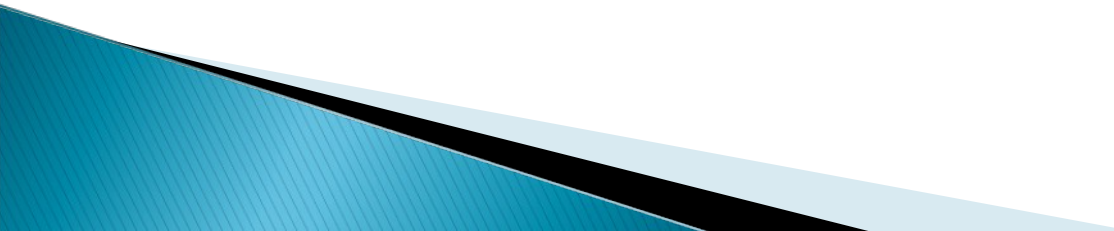
Unidade de lógica e aritmética

- ▶ Executa, a comando da UC, operações especificadas nas instruções de um programa tais como *adição* ou *comparação* de dois números.
 - ▶ Um conjunto de registradores é usado para armazenar temporariamente os dados e resultados das operações.
 - ▶ ULA + registradores = Unidade de operação
- 

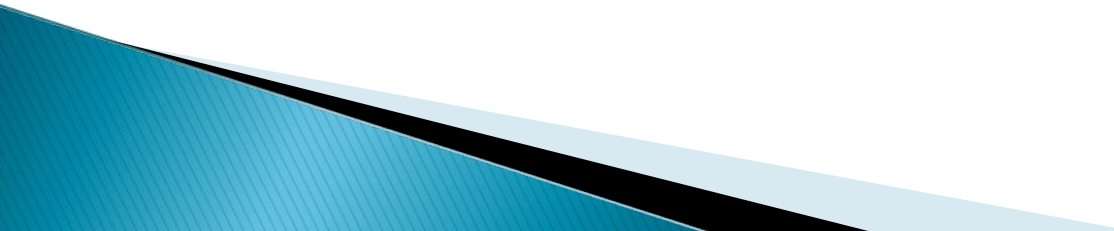
Memória

- ▶ Comumente chamada de memória principal ou RAM (*Random Access Memory*).
 - ▶ Usada para armazenar os programas a serem executados pelo computador e os dados manipulados por esses programas.
 - ▶ Arquitetura Von Neumann: programas e dados armazenados na memória principal.
- 

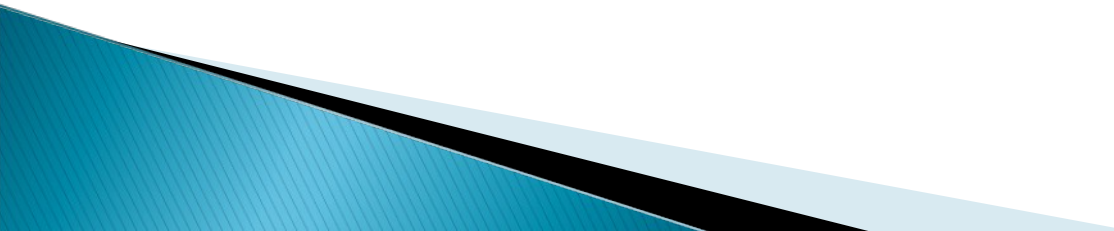
Memória

- ▶ Consiste em uma seqüência finita de unidades de armazenamento de dados, cada qual com o seu endereço.
 - ▶ Normalmente implementada como um conjunto de registradores implementados por flip-flops (circuitos que armazenam um bit).
 - ▶ Cada unidade de armazenamento de dados memória é chamada de *palavra*.
- 

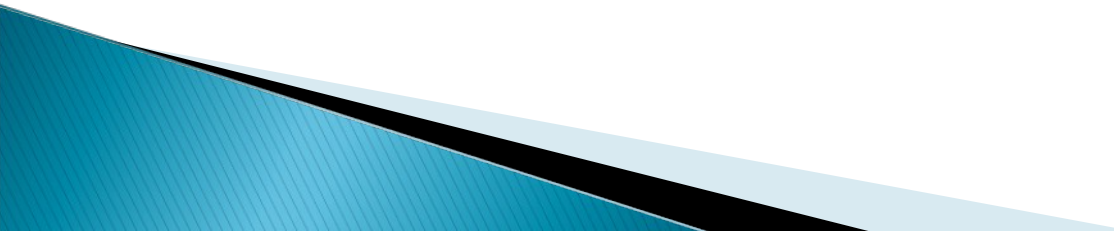
Memória

- ▶ Uma palavra de memória é composta de um número pequeno (em geral, 4 ou 8) de bytes – cada byte armazena uma seqüência finita de 8 bits.
 - ▶ Os dados ou instruções do programa são armazenados em uma ou mais palavras de memória do computador.
- 

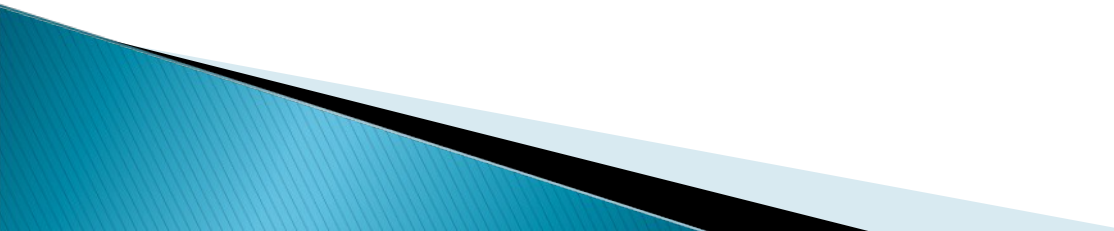
Memória

- ▶ A linguagem de máquina de um computador contém instruções para transferência de dados da *memória* para os *registradores* do processador, e vice-versa.
 - ▶ Nessas instruções, os dados ou programas são referenciados por meio dos endereços das palavras de memória.
- 

Dispositivos de entrada e saída

- ▶ Usados para comunicação de dados entre o computador e o mundo externo.
 - ▶ Teclado e *mouse* são exemplos de dispositivos de *entrada*.
 - ▶ A tela do computador e a impressora são exemplos de dispositivos de saída.
 - ▶ Discos e fitas são dispositivos de entrada e saída de dados.
- 

Linguagens, compiladores e interpretadores

- ▶ Linguagem de máquina:
 - Instruções para comandar a execução de operações primitivas, tais como: somar dois números ou comparar se dois números são iguais;
 - Instruções para transferência de dados entre memória e registradores do processador ou entre memória e dispositivos de E/S;
 - Instruções para controlar o fluxo de execução das instruções de um programa,
- 

Linguagens, compiladores e interpretadores

- ▶ Linguagem de máquina:
 - Exemplos:
 - ❑ Carregar no registrador R1 o valor armazenado no endereço 3006 da memória;
 - ❑ Somar os valores contidos nos registradores R1 e R2 e armazenar o resultado em R1;
 - ❑ “Desviar” para a instrução armazenada no endereço 520 da memória, se o resultado obtido pela execução da instrução anterior for diferente de zero.

Linguagens, compiladores e interpretadores

- ▶ As instruções anteriores, formadas por dezenas de bits, eram utilizadas para descrever algoritmos.
- ▶ Programar era trabalhoso, sujeito a erros difíceis de detectar e corrigir.
- ▶ Programadores começaram a utilizar *nomes* para as operações e dados, escrevendo as instruções anteriores como:

Linguagens, compiladores e interpretadores

- ▶ MOV R1, 3006
- ▶ ADD R1, R2
- ▶ JNZ 520
- ▶ As instruções mnemônicas acima eram traduzidas manualmente para seqüências de bits e depois carregadas na memória para execução.

Linguagens, compiladores e interpretadores

- ▶ A *montagem* do programa (escrito em linguagem mnemônica) para linguagem de máquina foi transferida para o computador.
- ▶ O programa que traduz para linguagem de máquina é chamado de *montador* (*assembler*).
- ▶ A notação mnemônica traduzida é chamada de *linguagem de montagem* (*assembly language*).

Linguagens, compiladores e interpretadores

- ▶ *Linguagem de montagem* também é difícil de programar → muito próxima da *linguagem de máquina* → pouca relação com as abstrações usualmente empregadas para construção de algoritmos e solução de programas.
- ▶ Nova linguagens foram desenvolvidas para facilitar a tarefa de programação.

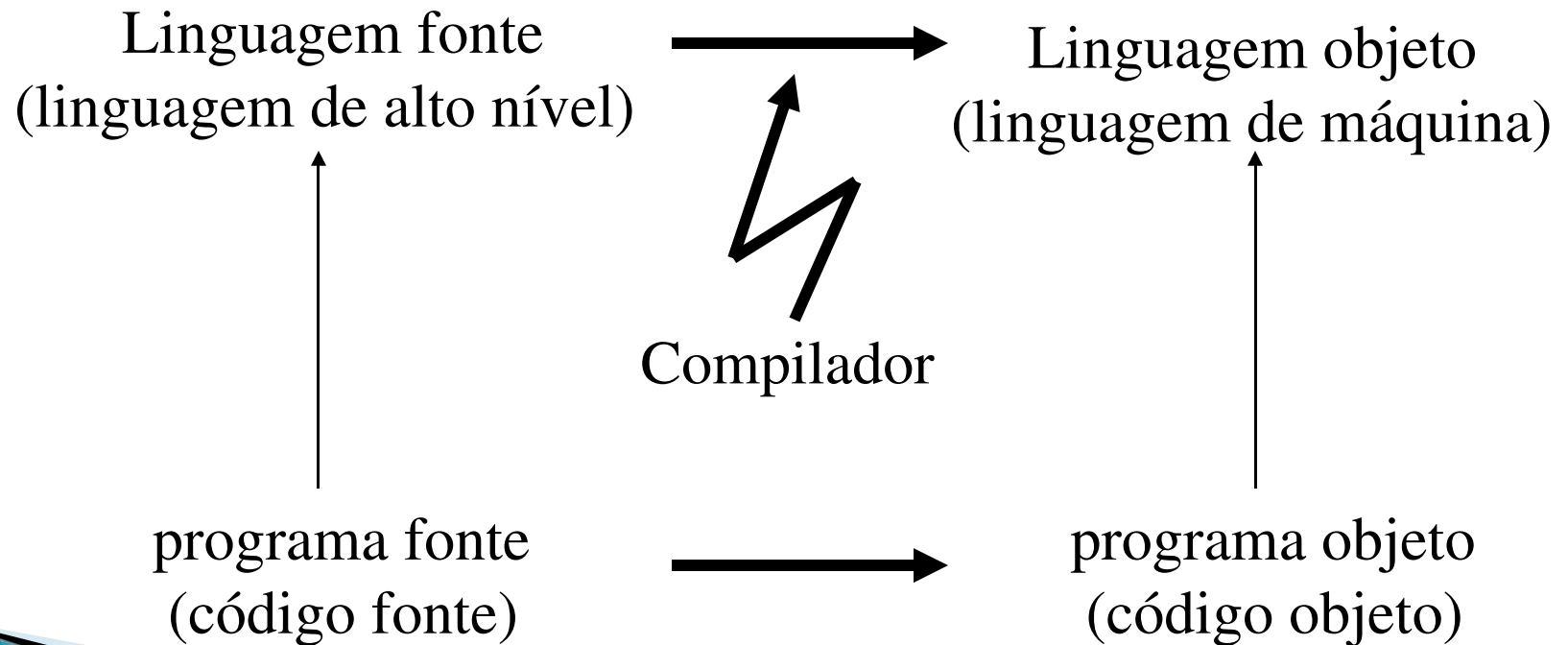
Linguagens, compiladores e interpretadores

- ▶ Chamadas de linguagem de alto nível, essas linguagens oferecem:
 - Um conjunto mais rico de operações e construções sintáticas adequadas para expressar de maneira natural algoritmos utilizados na solução de problemas.
- ▶ Linguagem de máquina e de montagem são chamadas *linguagem de baixo nível*, em contraposição.

Linguagens, compiladores e interpretadores

- ▶ Problema:
 - Programa escrito em uma linguagem de alto nível precisa ser *traduzido* para linguagem de máquina para ser utilizado em um computador.
- ▶ Esse processo de tradução é chamado de *compilação* e o programa que o realiza *compilador*.

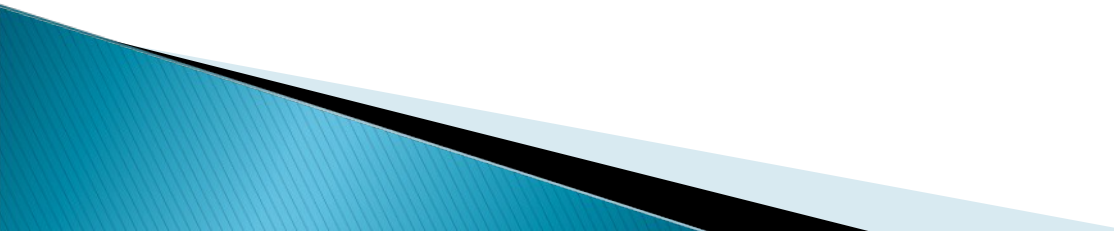
Linguagens, compiladores e interpretadores



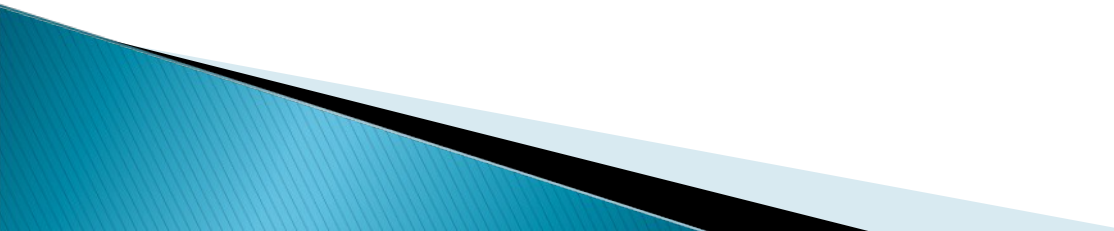
Compilador

- ▶ Analisa o texto do programa fonte para determinar se ele está sintaticamente e semanticamente correto, isto é, em conformidade com as regras da gramática da linguagem e, em caso afirmativo, gera o código objeto equivalente.

Interpretação

- ▶ Outra maneira de executar uma linguagem de alto nível.
 - ▶ A idéia é interpretar diretamente as frases do programa fonte.
 - ▶ Isto é, simular a execução do programa “fazendo de conta” que a linguagem de alto nível é a “linguagem de máquina do computador”.
- 

Interpretação

- ▶ Um programa, chamado *interpretador*, interpreta diretamente as frases do programa fonte fornecidos como entrada (juntamente com os dados).
 - ▶ Pode ser centenas de vezes mais lenta que a execução de código objeto gerado pelo compilador.
 - ▶ Por quê?
- 

Interpretação

- ▶ Envolve análise e simulação da execução de cada instrução do programa. → análise e simulação realizadas em *tempo de execução*.
- ▶ Na compilação, a análise é feita previamente (antes da execução). → análise realizada em *tempo de compilação*.

Interpretação

- ▶ Vantagens:
 - Mais fácil desenvolver um interpretador do que um compilador.
 - Execução independe da linguagem de máquina do computador, desde que exista interpretadores para diferentes máquinas.
- ▶ Java: mistura compilação e interpretação.

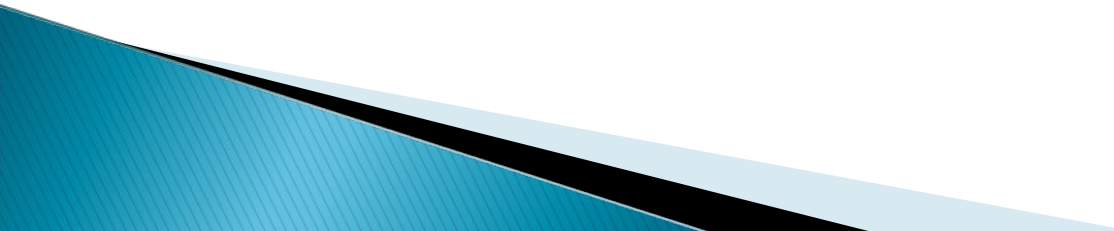
Interpretação

- ▶ Compilador Java traduz um programa Java para programas escritos em código de mais baixo nível → *bytecodes*.
- ▶ *Bytecodes* são interpretados; as instruções definidas pelos *bytecodes* constituem a “máquina virtual Java” (*JVM – Java Virtual Machine*).
- ▶ O interpretador Java para as diferentes plataformas (Win, Linux, Sun etc) é também chamado de “máquina virtual Java” (JVM).

Interpretação

- ▶ Vantagens da JVM:
 - *Bytecodes* são mais fáceis de analisar do que programas fontes Java → interpretação mais rápida.
 - Aplicações para a internet → programas em *bytecodes* podem ser transferidos através da rede e executado em qualquer outro computador que disponha de uma JVM.

Sistema Operacional (SO)

- ▶ Controla a operação em conjunto dos diversos componentes do computador – processador, memória e dispositivos de E/S – assim como a execução simultânea de diversos programas pelo computador.
 - ▶ Núcleo do SO é iniciado no momento em que é ligado o computador, quando é transferido do disco para a memória, ficando residente enquanto o computador estiver ligado.
- 

Sistema Operacional (SO)

- ▶ O núcleo do SO provê uma interface adequada entre a máquina e os demais programas do SO.
- ▶ Estes, por sua vez, oferecem uma interface adequada para os diversos componentes do computador e para os usuários e seus programas.

Bibliografia

- ▶ Camarão, C.; Figueiredo, L. “Programação de Computadores em Java”, Livros Técnicos e Científicos Editora, 2003. Capítulo 1.