



Linguagens de Programação

Introdução a Linguagens de Programação

Aula Teórica 1

Introdução

O que é computador?

- ✓ Dispositivo electrónico;
- ✓ Conjunto de hardware e software;
- ✓ Recebe instruções;
- ✓ Processa dados;
- ✓ Entende uma linguagem específica.

Sistema Computacional

Associação entre dois conceitos utilizados na terminologia de informática:

1. **Hardware:** todos os componentes físicos que fazem parte de um sistema informático, (nomeadamente os processadores, *motherbord*, disco, placa de som, placa de vídeo, memórias, etc
2. **Software:** conjunto de programas responsáveis pela execução das tarefas.

Sistema Operativo

- ✓ Programa ou conjunto de programas que tem como função servir de interface entre o computador e o usuário;
- ✓ Responsavel pelo funcionamento do computador.



Tipos de Software

Os softwares podem ser classificados em três tipos:

1. Software de Sistema: Tem como função servir de base ou suporte a outros softwares.

- ✓ Responsável pela gestão de todos os recursos disponíveis no sistema.
- ✓ Permite que vários programas sejam executados ao mesmo tempo.

Software de Sistema

Há dois tipos de Interfaces de Sistemas Operativos:

- ✓ CLI - Command Line Interface

comandos digitados pelo usuário através do teclado

Ex: MS DOS,

- ✓ GUI - Graphics User Interface

facilidade na operação e comunicação

Ex: Microsoft Windows,.

Tipos de Software

2. Software de Programação: conjunto de ferramentas que permitem ao programador desenvolver sistemas informáticos, usando linguagens de programação por meio de ambiente de desenvolvimento integrado.

3. Software de Aplicação: são programas de computadores que permitem ao usuário executar tarefas específicas em diversas áreas de atividade como arquitectura, contabilidade, educação, medicina, comercial, etc.



Conceito de LP

Uma **linguagem de programação** é um método padronizado para comunicar instruções a um computador através de um conjunto de regras sintáticas e semânticas;

Permite que o programador especifique precisamente sobre quais os dados que o computador deve operar, como estes dados serão armazenados e que acções devem ser tomadas sob diferentes circunstâncias.



Conceito de LP

Conjunto de instruções, escrito através de uma linguagem própria, que orienta o computador a realizar tarefas.

A única linguagem que o computador "entende" é a chamada Linguagem de Máquina, formada por um conjunto de códigos numéricos ou bits. Porém, estes códigos são extremamente complicados para o entendimento humano, por isso surgiram as linguagens de programação.



Porquê Tantas Linguagens?

Perl, C, Assembly, PHP, Pascal, C++,
Python, Ruby, Swift, Visual Basic, Cobol,
Lisp, Haskell, Java, Oberon, Prolog, Ada,
Smalltalk, Símula, Algol, Fortran,
Scheme, CLOS, Maude,



Porquê Tantas Linguagens?

- ✓ Propósitos Diferentes;
- ✓ Avanços Tecnológicos;
- ✓ Interesses comerciais;
- ✓ Cultura.



Porquê Tantas Linguagens?

Existem varias linguagens de programação e novas linguagens surgem frequentemente, trazendo novos paradigmas e estabelecendo novos padrões para programadores.

É importante conhecer as principais diferenças entre as linguagens de programação, e quando o uso de cada uma delas é apropriado.



Classificação das LP

- ✓ Quanto ao Nivel;
- ✓ Quanto à Geração;
- ✓ Quanto ao Paradigma;
- ✓ Quanto ao Domínio de Aplicação;



Níveis de LP

1. Linguagens de Baixo Nível

Linguagens voltadas para a máquina, escritas utilizando as instruções do microprocessador, de difícil compressão humana.

Ex: Linguagem de Máquina

Na linguagem de máquina, a representação dos dados e das operações (instruções), baseia-se no sistema binário (0 e 1), que é a forma compreendida e executada pelo hardware.

Sequências de dígitos binários (0s e 1s).

A instrução “some 1 + 1” pode ser representada como:
10100100

Linguagens de Baixo Nível

Linguagem Hexadecimal

Para simplificar a compreensão, foi adotado (num primeiro momento) a notação hexadecimal para representar programas em linguagens de máquina. Entretanto, a programação e leitura usando essa linguagem continuou impraticável.

Linguagem Assembly

A linguagem Assembly é na realidade uma versão legível da linguagem de máquina, utilizando palavras abreviadas, chamadas mnemônicos, indicando a operação a ser realizada pelo processador.

Exemplo: `MOV AX, 21`

`MOV DS, 3333`

Linguagens de Baixo Nível

Vantagens

- ✓ Executados com maior velocidade de processamento;
- ✓ Ocupam menos espaço na memória;
- ✓ Altamente recomendavel quando o “tempo” e factor critico.



Linguagens de Baixo Nível

Desvantagens

- ✓ Dificil entendimento humano;
- ✓ Grande probabilidade de erros em todos os estágios de programação;
- ✓ Dificil detecção de erros;
- ✓ Não é portátil, apenas dentro de uma família de processadores;
- ✓ Endereços de memória são feitos manualmente;
- ✓ Como o programador utiliza diretamente os recursos do processador e memória, deve conhecer muito bem a máquina;
- ✓ Custos elevados.

Linguagens de Alto Nível

2. Linguagens de Alto Nível

Mais próximas à linguagem do homem, e independentes da arquitetura do computador, podendo ser migradas de uma máquina a outra.

Permitem ao programador abstrair-se completamente do funcionamento interno da máquina.



Linguagens de Alto Nível

Vantagens

- ✓ Facilidade de compreensão e aprendizado humano;
- ✓ Utiliza sintaxe estruturada, tornando código mais fácil de entender;
- ✓ Independente da arquitetura do computador;
- ✓ Portabilidade, podem ser executadas em diferentes plataformas;
- ✓ Facilidade na detecção e correção de erros;
- ✓ Baixo custo de manutenção.

Linguagens de Alto Nível

Desvantagens

- ✓ Ocupam muita memória, tornando-as lentas;



Geração de Linguagens de Programação

As linguagens são também classificadas em gerações em função das características que apresentam, considerando-se linguagens com características semelhantes pertencendo à mesma geração.

1ª Geração - Linguagem máquina- Conjunto de dígitos binários.

Ex: 0010 0001 0110 1100



Geração de Linguagens de Programação

2ª Geração: compreende as linguagens simbólicas ou de montagem (Assembly), projetada para minimizar as dificuldades da programação em notação binária.

Códigos de operação e endereços binários foram substituídos por mnemônicos (abreviações).

Ex: mov mul add label goto

Geração de Linguagens de Programação

3ª Geração: AS primeiras consideradas de alto nível, e de fácil entendimento para o ser humano, usando compiladores e interpretadores.

Ex: Fortran, ALGOL e COBOL- primeiras linguagens deste tipo.
As mais utilizadas recentemente BASIC, C , C++.

4ª Geração: linguagens desta geração, usam a abstração, eliminando a especificação detalhada, agregando diferentes características.

Ex: SQL, JAVA.



Geração de Linguagens de Programação

5ª Geração: Métodos específicos de resolução dos problemas, com foco na inteligência artificial.

Ex: PROLOG, LISP.



FIM

