Universidade Federal do Rio Grande do Norte
Centro de Tecnologia
Departamento de Computação e Automação

Linguagens de Programação

Professor Responsável:
Luiz Affonso Henderson Guedes de Oliveira
Prof. Do Estágio Docente:
Kliger Kissinger F. Rocha
Valnaide Gomes Bittencourt

Turma:
Engenharia Química – 2004.1

Natal, RN, abril/2004

LINGUAGENS DE PROGRAMAÇÃO

- Usadas para descrever algoritmos; isto é, seqüências de passos que levam à solução de um problema.
- Permitir que os usuários especifiquem como estes passos devem ser seqüenciados para resolver um problema.
- Especificar algoritmos com precisão.

Tipos de Linguagem

- As *linguagens de baixo nível*
  - Restritas a linguagem de máquina
  - Forte relação entre as operações implementadas pela linguagem e as operações implementadas pelo hardware.
- As *linguagens de alto nível*
  - Aproximam-se das linguagens utilizadas por humanos para expressar problemas e algoritmos
  - Cada declaração numa linguagem de alto nível equivale a várias declarações numa linguagem de baixo nível.

Primeiras Linguagens

- Programadores usavam linguagem de máquina
  - Seqüências de dígitos binários (0s e 1s).
  - Por exemplo, a instrução “some 1 + 1” deveria ser representada como: 10100100
- Muitas desvantagens:
  - Grande probabilidade de erro em todos os estágios do processo de programação.
  - Mesmo sendo com algoritmos simples resulta em longos programas, o que dificulta o processo de validação e detecção de erros.
  - O cálculo de endereços de memória devem ser feitos manualmente, com um árduo trabalho e uma grande probabilidade de erros.

Assembler

- Algumas das desvantagens podem ser superadas fazendo com que o computador seja o responsável pelo estágio de tradução.
- O programa ainda é escrito em termos de operações básicas de máquina, mas a tradução em código binário é feita pelo computador.
- O programa que faz essa tradução é chamado de *assembler*
- Exemplo: ADD 1,1
- Trata do problema de cálculo de endereço, usando nomes em formato de texto para endereçar os dados.

Linguagens de Alto Nível *Vs* Linguagens de Baixo Nível

- Alto nível
  - Problemas podem ser solucionados muito mais rapidamente e com muito mais facilidade
  - A solução do problema não necessita ser obscurecida pelo nível de detalhes necessários em um programa em linguagem de baixo nível.
  - O programa em linguagem de alto nível é normalmente fácil de seguir e entender cada passo da execução.

Linguagens de Alto Nível *Vs* Linguagens de Baixo Nível

- Baixo nível
  - Indicada para funções que precisam implementar instruções de máquina específicas que não são suportadas por linguagens de alto nível
  - A grande eficiência e o reduzido tamanho dos programas
  - Impossibilidade de uso de linguagens de alto nível (hardware simples)

Histórico das Linguagens de Programação

- Existem centenas de linguagens de programação
- Agrupadas em 4 gerações:
  - Primeira geração: linguagem de máquina
  - Segunda geração: grande quantidade de bibliotecas de software, sistemas de execução em tempo real e desenvolvimento de gerenciadores de base de dados.
  - Terceira geração: capacidade procedural e estrutural
  - Quarta geração: sistemas especialistas, desenvolvimento de inteligência artificial execução dos programas em paralelo.

Processamento de Linguagens

- Embora seja teoricamente possível a construção de computadores especiais, capazes de executar programas escritos em uma linguagem de programação qualquer, os computadores existentes hoje em dia são capazes de executar somente programas em uma linguagem de nível baixo, a *linguagem de máquina*.
- Linguagens de máquina: rapidez de execução de programas, do custo de sua implementação

Processamento de Linguagens

- Linguagens de programação: facilidade na construção e da confiabilidade de programas
- Um problema básico:como uma linguagem de nível mais alto pode ser implementada em um computador cuja linguagem de máquina é bastante diferente, e de nível bem mais baixo.
- Existem basicamente duas alternativas para esta implementação: interpretação e tradução.

Interpretação

- Nesta solução, as ações indicadas pelos comandos da linguagem são diretamente executadas.
- Mais precisamente, um interpretador é um programa que executa repetidamente a seguinte seqüência:
  1. Obter o próximo comando do programa.
  2. Determinar que ações devem ser executadas.
  3. Executar estas ações.

Tradução

- Nesta solução, programas escritos em linguagem de alto nível são traduzidos para versões equivalentes em linguagem de máquina, antes de serem executados.
- Esta tradução é feita em vários passos. Por exemplo:
  - Inicialmente traduzidos para código Assembly (compilador)
  - Depois ser traduzido para código relocável (objeto), em linguagem de máquina
  - O programa inteiro é carregado na memória principal, como código executável de máquina.