UniRuy & Área 1 | Wyden

PROGRAMA DE CIÊNCIA DA COMPUTAÇÃO

TEORIA DE COMPILADORES

JOÃO MARCELO TAVARES SOUZA MATOS

Teoria de Compiladores: Noções e Terminologias Matemáticas

Salvador - Bahia - Brasil

2022

JOÃO MARCELO TAVARES SOUZA MATOS

Teoria de Compiladores: Noções e Terminologias Matemáticas

Trabalho Acadêmico elaborado junto ao programa de Engenharia UniRuy & Área 1 | Wyden, como requisito para obtenção de nota parcial da AV1 na disciplina Teoria de Compiladores no curso de Graduação em Ciência da Computação, que tem como objetivo consolidar os tópicos do plano de ensino da disciplina.

Orientador: Prof. MSc. Heleno Cardoso

Salvador - Bahia - Brasil
2022

da Tal, Aluno Fulano

Teoria de Compiladores: Resenha / Mapa Mental / Perguntas

– Aluno Fulano de Tal. Salvador, 2022.
18 f. : il.

Trabalho Acadêmico apresentado ao Curso de Ciência da Computação, UniRuy & Área 1 | Wyden, como requisito para obtenção de aprovação na disciplina Teoria de Compiladores.

Prof. MSc. Heleno Cardoso da S. Filho.

1. Resenha
2. Mapa Mental
3. Perguntas/Respostas (Mínimo de 03 – Máximo de 05)
4. Conclusão

I. da Silva Filho, Heleno Cardoso II. UniRuy & Área 1 | Wyden. III. Trabalho Acadêmico

CDD:XXX

TERMO DE APROVAÇÃO

JOÃO MARCELO TAVARES SOUZA MATOS

TEORIA DE COMPILADORES: NOÇÕES E TERMINOLOGIAS MATEMÁTICAS

Trabalho Acadêmico aprovado como requisito para obtenção de nota parcial da AV1 na disciplina Teoria de Compiladores, UniRuy & Área 1 | Wyden, pela seguinte banca examinadora:

BANCA EXAMINADORA

Profº. MScº. Heleno Cardoso
Wyden

Salvador, 09 de Outubro de 2022

Dedico este trabalho acadêmico a todos que contribuíram direta ou indiretamente com minha formação acadêmica.

Agradecimentos

Primeiramente agradeço a Deus. Ele, sabe de todas as coisas, e através da sua infinita misericórdia, se fez presente em todos os momentos dessa trajetória, concedendo-me forças e saúde para continuar perseverante na minha caminhada.

E a todos aqueles que contribuíram direta ou indiretamente para a minha formação acadêmica.

"A educação tem raízes amargas, mas os seus frutos são doces".

Aristóteles.

Resumo

Diante de todos os avanços tecnologicos, a linguagem de programação teve importancia na maioria delas, sendo também um avanço, desde 1884 até os dias atuais, avançamos em uma direção em que as linguagens propociam soluções para todos os tipos de problemas computacionais.

Com diversas linguagens existentes, podemos solucionar muitos problemas sendo esses problemas empresariais, pessoais, IoT, Machine Learning, Web e muitas outras.

Palavras-chaves: Lógica, Computação, Linguagem de programação, IoT, Machine Learning, algoritmos.

Abstract

languages of all technological advances, programming programming had importance in most languages, being also an advance, since the present day, we have advanced in a direction for the proposed solutions as all kinds of computational problems.

Keywords: Logic, Computing, Programming language, IoT, Machine Learning, algorithms.

Lista de abreviaturas e siglas

COBOL	- COMMON BUSINESS ORIENTED LANGUAGE
IBM	- International Business Machines Corporation
IEC	- International Electrotechnical Commission
Assembly	- Código de máquina.
Scanner	- Analisador Léxico.
Parser	- Analisador Sintático / Semântico.

Sumário

[1] De acordo com a Associação Brasileira de Normas Técnicas. NBR 6023.

1 Linguagens de programação

1.1 Introdução

Diante do surgimento das tecnologias, não foi o criador do primeiro computador que criou a primeira linguagem de programação, por volta de 1883, Ada Lovelace criou o que seria a primeira linguagem de programação, o codigo escrito por Ada servia para computar o número de Bernolli.

Atualmente as linguagens de programação servem para resolver problemas do dia a dia, construir novas soluções, resolver problemas matematicos e muitas outras finalidades. Dito isso, sem as linguagens de programação muitos avanços tecnologicos não seriam possiveis

1.1.1 Perguntas e Respostas - Mínimo de 2 e Máximo de 5

1° Pergunta: Entre a compilação e interpretação, qual tem melhor otimização de codigo?

2° Pergunta: Na Linguagem de Programação SQL, ele resolve qual paradigma da programação?

1.2 Conclusão

Conforme os tópicos citados nesse artigo, desde a evolução das linguagens até os dias de hoje, a programação e suas linguagens passaram por diversas mudanças e melhorias. Hoje possuímos facilmente linguagens que nos trazem soluções mais rápidas e fáceis para problemas simples ou robustos.

É necessário pensar que, o desenvolvimento é continuo dessas linguagens e a cada dia que passa se criam novos paradigmas e novas resoluções desses mesmos paradigmas, isso torna a programação uma ciência continua e que avança junto as evoluções tecnológicas como um todo.

2 Evolução das linguagens de programação

2.1 Evolução das linguagens de programação

De 1884 até 2022 são quase 138 anos de evolução desde a primeira linguagem, hoje temos linguagens de sobra e até linguagens criando inteligência artificial que cria linguagens. Essa evolução só foi possível pela persistência dos pesquisadores que em toda história buscaram evoluir para termos as linguagens de alto nível que temos hoje.

As linguagens de programação são dividias em dois subtópicos, as linguagens de baixo nível que se aproxima mais da linguagem de máquina e a linguagem de alto nível que se aproxima mais da linguagem humana, é obvio que hoje, a maioria dos desenvolvedores de software usam linguagem de alto nível para escrever seus algoritmos, tanto por ser mais intuitivo, tanto por terem mais atualizações e comunidades enormes.

2.1.1 Primeira era das linguagens

Na primeira era das linguagens, podemos citar várias, desde a sua criação com Ada Lovelace, até Plankalul desenvolvida por Konrad Zuse, que tinham a maior finalidade de realizar repetidamente operações de rotina.

2.1.2 Assembly

Em todo o mundo, não existe uma linguagem que seja mais importante que o Assembly, é considerada uma das linguagens de baixo nível mais importantes de todas, ela consiste em instruções que somente máquinas entendem.

2.1.3 Linguagens de baixo nivel

Falando em linguagem de baixo nível, não podemos entrar nesse tópico sem falar do FORTRAN, a primeira linguagem de programação a ser comercializada, em 1957 a IBM foi a lançadora dessa linguagem de programação no mercado, ela foi projetada para ser usada na computação numérica e na ciência, ela foi usada em diversas ocasiões, mas uma das mais honrosas, foi ter sido usada na sonda espacial NASA Voyager 1 e 2.

Outra bem importante é o COBOL, que foi lançada em 1959, que tinha como paradigma a orientação a para o processamento de banco de dados comerciais, isso fazia com que dados computacionais armazenados pudessem ser usados nas empresas, em 1997 cerca de 80 porcento de todos os sistemas usavam COBOL.

E por fim a linguagem de programação mais popular até hoje, o C é uma linguagem de programação procedural e é a mãe de todas as linguagens de alta no nível como CSharp, Java, Python e PHP.

2.1.4 Linguagens de alto nivel

As linguagens de alto nível são linguagens que se aproximam da comunicação humana, hoje são responsáveis pela maioria das soluções de software e web e tornam nossa vida melhor, redes sociais como Facebook e todos os aplicativos Google, usam algumas dessas linguagens de alto nível, podendo citar o JavaScript com seus frameworks React e Angular, criados respectivamente pelo Facebook e Google.

Atualmente a linguagem de programação mais popular é o Python que nasceu em 1991 e que possui soluções para muitos paradigmas, desde orientação a objetos, a IoT e Machine Learning. (LIMA, 2020)

3 Tipos de tradutores

3.1 Tradutores de linguagem de programação

No mundo de programação, existem a comunicação entre o programador e programa tradutor, este tem como objetivo entender as instruções do programador e fazer com que essas instruções sejam lidas pela máquina, independentemente de como são escritas.(GOIAS, 2017)

3.1.1 Tipo interpretador

No tipo interpretador, as instruções são definidas na linguagem de alto nível e traduzida em comandos de máquina, um de cada vez. (GOIAS, 2017)

Exemplificando, o dado entra, o programa traduz e devolve um dado de saída.(GOIAS, 2017)

3.1.2 Tipo Compilador

Já o tipo compilador é um pouco diferente, ele produz a partir do programa de entrada outro programa que é equivalente ao original, porém em uma linguagem que é executável pelo programador.(GOIAS, 2017)

4 Estrutura de um compilador: Fase de análise e fase de síntese

4.1 Fases de um compilador

Em um grosso modo, um compilador é um programa que traduz e cria um outro programa executável a partir do código fonte traduzido, e para que esse programa seja criado, há quatro etapas em um compilador. (MARANGON, 2015)

4.1.1 Analise léxica

A analise léxica é a primeira fase de um compilador, ele lê o código fonte caractere por caractere buscando a identificação desse código a partir de símbolos léxicos ou tokens.(MARANGON, 2015)

4.1.2 Analise Sintatica

A análise sintática é nada mais que a análise gramatical do código fonte, ele determina se os símbolos léxicos ou tokens para saber se realmente formam um programa valido ou não.

4.1.3 Analise Semantica

Diferente das duas analises anteriores, a analisa semântica tem um peso maior, pois ela tem a responsabilidade de analisar o que o analisador sintático envia e verifica se eles podem ser executadas.(MARANGON, 2015)

4.1.4 Fase de Sintese

É nessa fase que é construído o programa executável, nele existem módulos criacionais que irão otimizar o código recebido através das análises e gerar um executável. (MARANGON, 2015)

5 Tabela de símbolos e atendimento de erros

5.1 Tradução x Interpretação

A tradução ou compilação, faz com que cada instrução do programa de alto nível seja substituída por uma sequência equivalente de instruções em linguagem de máquina Já a interpretação converte as instruções de um programa fonte para a forma binaria e as executa imediatamente.(GOIAS, 2017)

Esses tradutores podem ser classificados como: Montadores, MacroAssemblers, Compiladores, Pré Compiladores e Pré Processadores.(GOIAS, 2017)

As maiores vantagens em relação entre compiladores e interpretadores são que os compiladores têm a execução mais rápida do código e não necessitam de tradução durante a execução, os compiladores também tem maior economia de memória na execução (GOIAS, 2017)

6 Ferramentas para construção de compiladores

Os criadores de compiladores podem usar algumas ferramentas para auxiliar a implementação de várias fases do compilador, existem algumas como:

Gerador de Parser, que produz analisadores de sintaxe(parsers) a partir da entrada que se baseia em uma descrição gramatical da linguagem de programação.

Scanner Generator, gera analisadores lexicais a partir da entrada que consiste na descrição de expressões regulares com base em tokens de uma linguagem. (LIMA, 2022)

7 Padrões Internacionais

7.1 Norma IEC 61131

Para atender às demandas da comunidade industrial internacional, foi formado um grupo de trabalho dentro da International Electrotechnical Commission (IEC) para avaliar o projeto completo de controladores lógicos programáveis, incluindo hardware, instalação, testes, documentação, programação e comunicação. (IEC, 2013)

Na IEC 61131 Parte 3, há a padronização internacional de linguagens, estrutura de software e execução de programas em CLPS's.

Existem cinco tipos básicos de linguagem que normalmente são encontradas em controladores programáveis e são padronizadas pela norma IEC 61131-3: Linguagens Textuais Texto Estruturado (Strutured Text – ST) Lista de Instruções (Instruction List – IL) Linguagens Gráficas Diagrama Ladder (LD) Diagrama Blocos Funcionais (Function Block Diagram – FBD) Dentro dos elementos comuns definidos pela norma existe o Seqüenciamento Gráfico de Funções ou Sequential Function Chart (SFC). O SFC descreve graficamente o comportamento seqüencial de um programa de controle e é derivado das técnicas de modelagem por Redes de Petri e da norma IEC 848 que define o padrão Grafcet.(IEC, 2013)

7.1.1 Linguagens Textuais

Texto Estruturado (Strutured Text – ST) É uma linguagem de alto nível muito poderosa, com raízes em Pascal e "C". Contém todos os elementos essências de uma linguagem de programação moderna, incluindo condicionais (IF-THEN-ELSE e CASE OF) e iterações (FOR, WHILE e REPEAT). (IEC, 2013)

7.1.2 Linguagens Gráficas

Diagrama Ladder (LD) A linguagem Ladder é, conforme mencionado anteriormente, a linguagem de programação de PLCs mais comum e a mais difundida, é também conhecida como lógica de diagrama de contatos, pois se assemelha à tradicional notação de diagramas elétricos e de painéis de controle a relés(IEC, 2013)

7.1.3 Estrutura de Software e Execução de Programas na IEC 61131

Existem elementos comuns da norma que são usados por todas as linguagens de programação IEC 61131 citadas. Um dos aspectos mais importantes da programação de qualquer sistema é a capacidade de decompor o software em partes menores. A norma prevê que sejam desenvolvidos ambientes de programação capazes de decompor programas complexos em diferentes elementos de software, os quais possuem uma interface padronizada e bem definida entre os mesmos. O modelo de software (software model) consiste em um conjunto de conceitos que definem uma infra-estrutura para decomposição em partes do projeto de automação. A programação baseada na norma IEC é orientada para o desenvolvimento de programas tanto a partir da abordagem de cima para baixo (top-down) como de baixo para cima (botton-up). (IEC, 2013)

Referências[1]

DEVSKILLER. *História das linguagens de programação*. 2020. Citado na página 21.

GOIAS, P. *Compiladores*. 2017. Citado 3 vezes nas páginas 15, 17 e 21.

IEC. *A Norma IEC 61131*. 2013. Citado 3 vezes nas páginas 19, 20 e 21.

LIMA, A. *A EVOLUÇÃO DAS LINGUAGENS DE PROGRAMAÇÃO*. 2020. Citado 2 vezes nas páginas 14 e 21.

LIMA, A. *FERRAMENTAS DE CONSTRUÇÃO DE COMPILADORES*. 2022. Citado 2 vezes nas páginas 18 e 21.

MARANGON, J. D. *Compiladores Para Humanos*. 2015. Citado na página 16.

SCUDERO, E. *Linguagens de alto Nível vs Baixo Nível: qual é Melhor?* 2018. Citado na página 21.

(GOIAS, 2017) (LIMA, 2020) (LIMA, 2022) (DEVSKILLER, 2020) (SCUDERO, 2018) (IEC, 2013)

[1] De acordo com a Associação Brasileira de Normas Técnicas. NBR 6023.