



UniRuy & Área 1 | Wyden
PROGRAMA DE CIÊNCIA DA COMPUTAÇÃO
TEORIA DE COMPILADORES

HEBER MAGNO DA SILVA REIS

Teoria de Compiladores: Linguagens de
Programação

Salvador - Bahia - Brasil

2022

HEBER MAGNO DA SILVA REIS

Teoria de Compiladores: Linguagens de Programação

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

HEBER MAGNO DA SILVA REIS

TEORIA DE COMPILADORES: LINGUAGENS DE PROGRAMAÇÃO

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

Prof^o. MSc^o. Heleno Cardoso
Wyden

Salvador, 08 de Setembro 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 educaão tem raízes amargas, mas os seus frutos são doces".

Aristóteles.

Resumo

O seguinte trabalho, tem como objetivo trazer informações referentes as linguagens de programação, sua evolução, as tecnologias por trás da transformação das linhas de código em um programa que de fato possa ser executados e todo o processo por trás das linguagens. Para isso foi realizados pesquisas sobre o tema em dois repositórios distintos e realizado uma resenha sobre os trabalhos selecionados, obtendo-se um resumo das informações citadas.

Palavras-chaves: Compiladores, Linguagens de Programação, Computação, Tradutores.

Abstract

The following work aims to bring information regarding programming languages, their evolution, as technologies behind the transformation of lines of code into a program that can actually be executed and the entire process behind the languages. For this, research was carried out on the subject in two different works and a review of the selected works was carried out, obtaining a selected summary of the cited information.

Keywords: Compilers, Programming Languages, Computing, Translators.

Sumário

1	Linguagens de Programação	10
1.1	Introdução	10
1.2	Execução/Método	10
1.2.1	Repositório de Pesquisa	10
1.2.2	String de Busca por Repositório	10
1.2.3	Artigos Seleccionados - Mínimo de 2 e Máximo de 5	10
1.2.4	Resenha dos Artigos Seleccionados	11
1.2.4.1	Linguagens de Programação	11
1.2.4.2	Teoria de Construção de Compiladores	12
1.2.4.3	Tradutores	13
1.2.4.4	Tecnologias Para a Construção de Compiladores	13
1.3	Análise de Resultados	14
1.3.1	Perguntas e Respostas	14
1.4	Conclusão	14
	Referências¹	15

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

1 Linguagens de Programação

1.1 Introdução

Atualmente vivemos em um mundo informatizado, onde percebemos que a computação teve uma grande evolução nos últimos anos, trazendo soluções para inúmeros dos problemas enfrentados pela sociedade, muitos softwares são desenvolvidos a todo para realizar tarefas que nos auxiliam no dia a dia.

Mas isso só se tornou possível graças ao surgimento das mais diversas linguagens de programação, para os mais diversos objetivos permitindo que os programadores possam solucionar diversos problemas através de sequencias de instruções que serão executadas pelo computador.

Diante disso este trabalho análise das linguagens de programação, seus paradigmas, as ferramentas para compilar os programas desenvolvidos e demais tópicos pertinentes ao que se refere a linguagens de programação.

1.2 Execução/Método

1.2.1 Repositório de Pesquisa

Para obtenção dos resultados de pesquisa para desenvolvimento deste trabalho, foram usados o Repositório Institucional da UTFPR e o Google Acadêmico.

1.2.2 String de Busca por Repositório

As seguintes strings de buscas foram aplicadas para obter os resultados da pesquisa: Repositório Institucional da UTFPR - linguagens de programação AND evolução AND compiladores AND simbolos. Google Acadêmico - linguagens de programação OR compiladores AND tipos de tradutores.

1.2.3 Artigos Selecionados - Mínimo de 2 e Máximo de 5

Para o seguinte trabalho foram selecionados das fontes de pesquisa os seguintes trabalhos:

1. MÉTODO PARA A CRIAÇÃO DE LINGUAGENS DE PROGRAMAÇÃO E COMPILADORES PARA O PARADIGMA ORIENTADO A NOTIFICAÇÕES EM PLATAFORMAS DISTINTAS de ([RONSZCKA, 2019](#))
2. Algoritmos Lógica para desenvolvimento de programação de computadores de ([MANZANO; OLIVEIRA,](#))

1.2.4 Resenha dos Artigos Selecionados

1.2.4.1 Linguagens de Programação

Antes de falar sobre as linguagens de programação é interessante entender o conceito de paradigmas de programação, este conceito, na ciência da computação, é utilizado como um modo de abstração de pensamento relacionado a um certa estrutura computacional apta a definir fluxos de execução de programas. De acordo com Peter Van Roy (2009), citado por ([RONSZCKA, 2019](#)), um paradigma de programação é um sistema formal que define como a programação é realizada. Ademais, cada paradigma tem o seu conjunto de técnicas e conceitos de programação que, conjuntamente, definem sua forma de estruturar o pensamento na construção de programas.

Na tese MÉTODO PARA A CRIAÇÃO DE LINGUAGENS DE PROGRAMAÇÃO E COMPILADORES PARA O PARADIGMA ORIENTADO A NOTIFICAÇÕES EM PLATAFORMAS DISTINTAS, o autor mostra que um modelo de paradigma de programação, para o programador, está disponível por meio de uma linguagem de programação. As primeiras linguagens de programação trouxeram em um primeiro momento os paradigmas imperativos, em seguida os paradigmas imperativo-procedimental. Após esse período novos paradigmas foram surgindo como o Paradigma Funcional, Paradigma Orientado a Objetos entre outros ([RONSZCKA, 2019](#)).

Nesse contexto, surgiram diversas foram surgindo diversas linguagens de programação para suportar esses paradigmas, a exemplo as linguagens Java para paradigma orientado a objetos, comumente chamada de POO, as linguagens Haskell e Standard ML para paradigmas funcionais, e ainda há as linguagens multiparadigmas, que são aquelas no qual os programadores podem trabalhar com diferentes paradigmas fazendo relações com as diferentes estruturas.

Evolução

A programação de computadores teve início por volta da década de 30 com os primeiros computadores elétricos, mas as primeiras linguagens de programação modernas começaram a surgir por volta da década de 50 como exemplo o FORTRAN (1955), em seguida surgiram os paradigmas de programação e assim vieram linguagens como o C, que ainda tem uma certa influência no mundo atual e o Pascal. Nos anos 80 surgiu a linguagem C++, que teve como objetivo ser compatível com a linguagem C e teve grande importância por ser simples e dinâmica, já nos anos 90 teve o surgimento da internet, que deu novas caras ao rumo da programação, e com isso trouxe algumas das linguagens mais famosas atualmente, como o Javascript e o PHP que tiveram como objetivo inicial o desenvolvimento de web sites, e também a linguagem Java, que como dito anteriormente foi criada para o paradigma orientado a objetos.

1.2.4.2 Teoria de Construção de Compiladores

O surgimento da teoria de construção dos compiladores, teve como objetivo trazer uma definição para os parâmetros que permitem o desenvolvimento de linguagens de programação e suas execuções, por meio de tradutores ou compiladores. Um compilador, de modo geral, recebe um programa escrito em determinada linguagem e traduz para um programa equivalente em outra linguagem. (RONSZCKA, 2019).

O processo de compilação é dividido em duas etapas:

1. Fase de Análise: Compreende as etapas de análise léxica, sintática e semântica. A etapa léxica, basicamente faz uma análise se os símbolos presentes no código são válidos, extraíndo os elementos para a composição de uma certa estrutura de dados denominada de tabela de símbolos, que tem como função basicamente mapear as funções, constantes, variáveis e tipos de dados de um programa. A etapa sintática, fez uma análise se os símbolos estão com o encadeamento de acordo com as especificações sintáticas da linguagem, criando as chamadas árvores abstratas sintáticas, que mapeiam a estrutura algorítmica de um programa. Por fim a etapa semântica, que com base nas duas últimas etapas avalia, de modo geral, se as atribuições de variáveis, passagens de parâmetros e retornos de funções estão tipados corretamente.

2. Fase de Síntese: Tem como objetivo gerar o código alvo, que geralmente é para uma linguagem de mais baixo nível, utilizando as informações da etapa de análise. Vale ressaltar que essa etapa pode sofrer algumas variações de um compilador para outro.

1.2.4.3 Tradutores

Após um projeto de um programa de computador estiver concluído, precisamos fazer com que ele de fato se transforme em um software, para isso é necessário realizar a tradução do código-fonte do projeto para uma linguagem que é entendida pelo computador, para tal é utilizado os chamados tradutores, que são ferramentas que possibilitam a tradução de um linguagem de programação para outra. Existem basicamente dois tipos de tradutores: os interpretadores que executam as instruções diretamente, traduzindo um comando de cada vez e chamando uma rotina para completar a execução. O segundo tipo são os compiladores, que produz um programa equivalente ao programa de entrada, podendo ser em uma outra linguagem que seja diretamente executável como as linguagens de máquina, ou indiretamente executável como outras linguagens que já contem um tradutor (MANZANO; OLIVEIRA,).

1.2.4.4 Tecnologias Para a Construção de Compiladores

De um modo geral, como qualquer outro problema computacional, os compiladores podem ser desenvolvidos utilizando linguagens de programação. no início, só existia a linguagem de máquina, que foi utilizada para desenvolver o montador Assembly, que posteriormente foi responsável por dar vida a primeira linguagem de auto nível, o Fortran, essa por sua vez foi a responsável por desenvolver outros compiladores. A exemplo temos a linguagem C Sharp, na qual seu compilador foi desenvolvido em C++ e posteriormente reescrito com o próprio C Sharp.

1.3 Análise de Resultados

1.3.1 Perguntas e Respostas

1. **Qual os tipos de tradutores e suas principais diferenças?** R: São os interpretadores, que executam o programa diretamente e os Compiladores que produz um programa equivalente em outra linguagem diferente da escrita inicialmente.
2. **Qual o principal objetivo da fase de compilação síntese?** R: Tem como objetivo gerar o código alvo, que geralmente é para uma linguagem de mais baixo nível, utilizando as informações da etapa de análise.

1.4 Conclusão

Durante o desenvolvimento do trabalho foi possível compreender um pouco mais sobre as diversas ferramentas que integram as linguagens de programação, entender os conceitos e como as linhas de código escritas por um programador podem se entendidas pelo hardware presente nos computadores. Foi possível perceber o quanto as linguagens de programação tem contribuído para a evolução da computação.

Referências¹

MANZANO, J.; OLIVEIRA, J. D. *Algoritmos Lógica para desenvolvimento de programação de computadores*. Saraiva Educação S.A. ISBN 9788536528403. Disponível em: <<https://books.google.com.br/books?id=c4uwDwAAQBAJ>>. Citado 2 vezes nas páginas 11 e 13.

RONSZCKA, A. F. *MÉTODO PARA A CRIAÇÃO DE LINGUAGENS DE PROGRAMAÇÃO E COMPILADORES PARA O PARADIGMA ORIENTADO A NOTIFICAÇÕES EM PLATAFORMAS DISTINTAS*. 2019. Citado 2 vezes nas páginas 11 e 12.

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