COMPILADOR: ANÁLISE LÉXICA, SINTÁTICA E GERAÇÃO DE CÓDIGO

JOÃO VICTOR BATISTA DE OLIVEIRA

UNIC: Universidade de Cuiabá.

Introdução

A construção de um compilador, mesmo em sua forma mais elementar, representa uma das experiências mais enriquecedoras no estudo da ciência da computação. Ao compreender os mecanismos que transformam código fonte em instruções executáveis, o estudante é conduzido ao cerne da linguagem de programação e da arquitetura de software. Este trabalho, desenvolvido como avaliação oficial do bimestre na disciplina de Compiladores, sob orientação do professor Felipe Douglas na Universidade de Cuiabá (UNIC), teve como proposta a implementação de um compilador simples utilizando a linguagem Python. O projeto foi concebido com o objetivo de ilustrar, de forma prática e didática, os principais estágios de um compilador: análise léxica, análise sintática, geração de código e execução.

Objetivo

O objetivo central deste projeto foi desenvolver um compilador funcional capaz de interpretar expressões aritméticas básicas encapsuladas em instruções de impressão, como por exemplo print(2+3). A proposta visou não apenas a execução correta da expressão, mas também a separação clara entre os módulos responsáveis por cada etapa do processo de compilação. Dessa forma, buscou-se proporcionar uma compreensão aprofundada dos conceitos fundamentais envolvidos na construção de compiladores, além de estimular a organização modular e a reutilização de código.

Metodologia

Para a realização do projeto, adotou-se uma abordagem modular, estruturando o compilador em quatro arquivos principais: scanner.py, syntax.py, generator.py e main.py. O arquivo scanner.py foi responsável pela análise léxica, utilizando expressões regulares para identificar os tokens presentes no código fonte, como números, operadores, parênteses e palavras-chave. Em seguida, o módulo syntax.py realizou a análise sintática, validando a estrutura da expressão e construindo a árvore sintática abstrata (AST). O módulo generator.py, por sua vez, converteu a AST em código Python equivalente, pronto para ser executado. Por fim, o arquivo main.py integrou todos os módulos, realizando a leitura da expressão, a tokenização, a construção da AST, a geração do código e sua execução.

REFERÊNCIAS:

https://repositorio.ucs.br/xmlui/bitstream/handle/11338/6308/TCC%20Felipe%20Miotto.pdf?sequence=1

https://www.inf.ufrgs.br/site/publicacoes/livros-didaticos/livros09/

COMPILADOR: ANÁLISE LÉXICA, SINTÁTICA E GERAÇÃO DE CÓDIGO

JOÃO VICTOR BATISTA DE OLIVEIRA

UNIC: Universidade de Cuiabá.

A estrutura do projeto foi organizada em uma pasta denominada meu_compilador, localizada em C:\Projetos, contendo um arquivo __init__.py para que o Python reconhecesse o diretório como um pacote. A execução do projeto foi realizada por meio do terminal, utilizando o comando python -m meu_compilador.main, o que garantiu o correto funcionamento dos imports e a integridade do pacote.

Todo o projeto foi idealizado e criado com suporte fundamental de inteligência artificial, como chatGPT e Copilot, cuja capacidade foi prioritária para o desenvolvimento do projeto.

Resultado

O compilador desenvolvido demonstrou pleno funcionamento ao interpretar corretamente expressões como print(2+3), gerando o código Python correspondente e exibindo o resultado esperado no terminal. A saída apresentada foi:

1. print(2 + 3)
2. 3

Além da execução correta, o projeto evidenciou a importância da organização estrutural em projetos Python, especialmente no que diz respeito à definição de pacotes e à forma de execução. A modularidade adotada permitiu a identificação e correção rápida de erros, bem como a possibilidade de expansão futura do compilador com novas funcionalidades.

Conclusão

A realização deste projeto proporcionou uma experiência prática valiosa na compreensão dos princípios que regem a construção de compiladores. Ao implementar cada etapa de forma independente e integrada, foi possível visualizar com clareza o fluxo de transformação do código fonte em instruções executáveis. O projeto cumpriu seu propósito acadêmico, servindo como instrumento de aprendizado e consolidação dos conceitos abordados na disciplina de Compiladores. Com sua estrutura modular e funcional, o compilador está apto a ser expandido com suporte a variáveis, múltiplas expressões, novos operadores e até mesmo entrada por arquivos externos. Mais do que um exercício técnico, este trabalho representa um marco na formação acadêmica, evidenciando a capacidade de aplicar teoria na prática e de construir soluções computacionais a partir de fundamentos sólidos.

REFERÊNCIAS:

https://repositorio.ucs.br/xmlui/bitstream/handle/11338/6308/TCC%20Felipe%20Miotto.pdf?sequence=1

https://www.inf.ufrgs.br/site/publicacoes/livros-didaticos/livros09/