Implementing a Programming Language

Nachiket Namjoshi

February 18, 2017

Introduction

What is A Programming Language?

- A programming language is basically a set of instructions to be followed by a computer so as to accomplish a particular task.

How Does it work?

- Computers *do not* understand a programming language.
- All they do understand is *machine code*.
- A machine code is a stream of bits (0s and 1s).

So, HOW does it work?

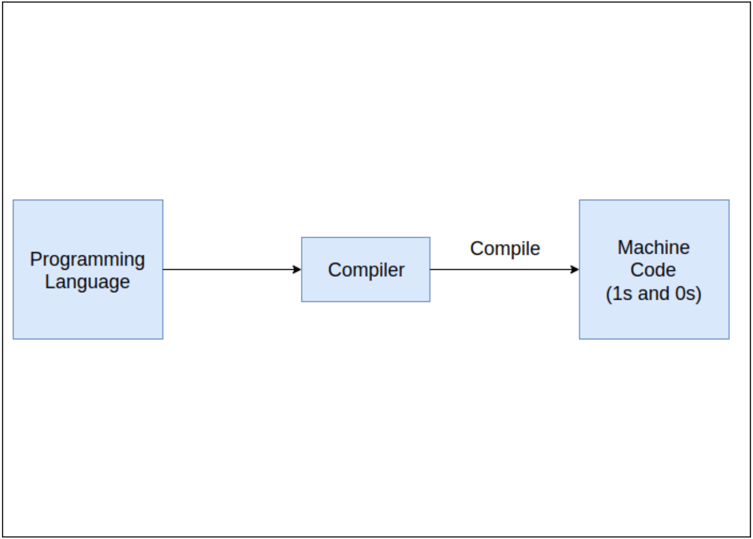
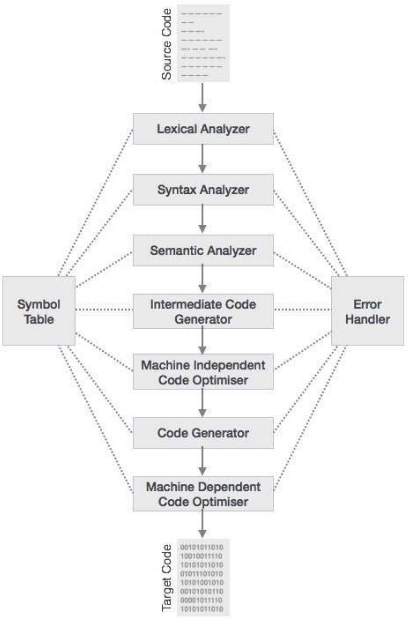


Figure 1: General Compilation

What is a Compiler?

- Compiler is basically a software that converts a programming language into another programming language.
- This Process is known as compilation.

Compilation Process



Problem Definition

Problem Definition

To design a compiler that compiles a C program and gives the following output:

- Pre-processed Code.
- Equivalent assembly language code
- Final output of the C program

Proposed System

Software/Hardware Requirements

Hardware:

- At least a dual core processor
- Minimum 2GB RAM

Software:

- Any GNU/Linux Based OS.
- A C Compiler *(GCC Preferred)*

How To Start?

Steps I followed *(and am following)* while implementation:

- Plan out the basic approach from start to end
    - Separate out all the lexemes
    - Add them all to an AST
    - Check the syntax
    - Emit proper Assembly code for each statement in the program.
- Work on each module one by one.
- Integrate all the modules.

Usage and Applications

Applications

- Implementing a compiler opens up the scope towards creating/establishing a *NEW* programming language.
- Concepts of Compilers are not limited to Compilers, whereas they are used in *several* other fields.
- Implementing such a significant project gives an *in-depth* idea about real life software development.

Limitations

- The implementation of a compiler is done by several other commercial companies, so it can only be used for educational purposes.
- A crudely formed compiler WILL produced *un-optimised* results.
- Failing to implement.

References

- Abstract syntax in theory and practice, Eugene J. Rollins
- AST Implementation in Python
- Atsushi Yoshida; Yoshinari Hachisu: A Pattern Search Method for Unpreprocessed C Programs Based on Tokenized Syntax Trees, 2014, IEEE 14th Conference.
- Yoshitaka Kato; Masaya Ozaki; Jun'ya Kani; Nobuhiro Ito; Yoshinobu Kawabe: Developing Compiler for Nihongo Programming Language PEN, 2015, IEEE 15th Conference

THANK YOU