Lecture 2

Compilation

Machine Language

Implementation

Advantage

Disadvantage

Md. Fatin Sirat	1610159042
Ayman Muniat	1721321642

Faculty : Dr. Md. Shahriar Karim

Machine Language:

Machine language is the lowest level language, which the computer can respond to directly. It purely consists of binary digits (0,1). Since, this is the only language that the computer understands, programs written in other programming languages must be converted into machine readable form for execution.
01000110 10001011 00101100 11011101 10001100 10100010 01000110 10010101

Implementation of Programming Languages:

Generally, programming languages can be implemented mainly in 3 ways, which are :-

1) Compilation
2) Pure Interpretation
3) Hybrid Implementation

1) Compilation: In this process, programs are translated into machine language and are directly executable by the computer. This process is known as the compiler implementation. Most production implementations of languages, like C, C++, COBOL etc are by compilers.

The language that a compiler translates is called the source language. The process of compilation and program execution takes place in many phases, as shown in figure 1.

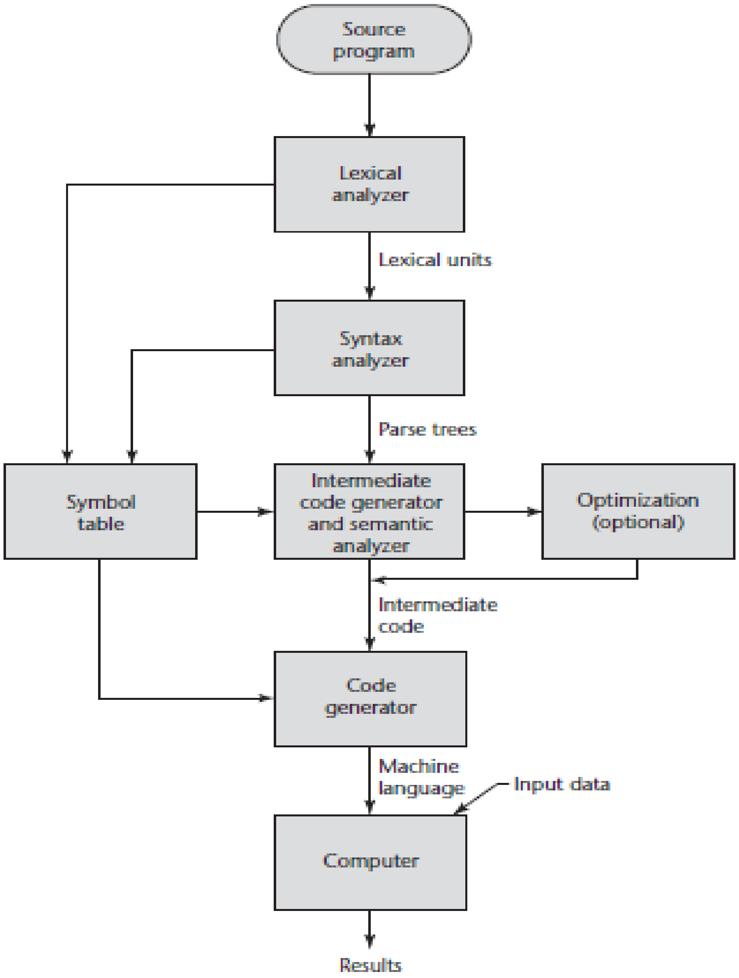

Figure 1 : The compilation process

Lexical Analyzer: The lexical analyzer gathers the characters of the source program into lexical units, which are, identifiers(constants, variables etc.), special words(int, string etc.), operators('+', '==' etc.), and punctuation symbols(for example, ';' for C).

Syntax Analyzer: The syntax analyzer takes the lexical units from the lexical analyzer and uses them to form hierarchical structures called parse trees, which represent the syntactic structure of the program.

Intermediate Code Generator: The intermediate code generator produces a program in a different language, at an intermediate level between the source program and the final result of the compiler, that is, the machine language program. Actually, the intermediate codes are quite similar to assembly languages, just probably at a level a little higher than the assembly language.

Advantage:

1. Compiler is advantageous over other methods in the sense that it is quite fast.
2. The machine language programs produced by a compiler are directly executable by computer, they must nearly always be run along with some other codes.
3. The compiler builds calls to required system programs when they are needed by the user program.

Disadvantage:

1. Debugging can be a tough task.
2.Compile time increases when code base grows.

2) Pure Interpretation: This is a one step process. With this approach, the program is interpreted by an interpreter, without any translation. Some simple early languages of the 1960s (APL, SNOBOL, and LISP) were purely interpreted, by the 1980s, the approach was rarely used on
high-level languages. However, in recent years, this process has made a comeback with JavaScript, PHP, Python etc.

An interpreter implements a language in the following way:

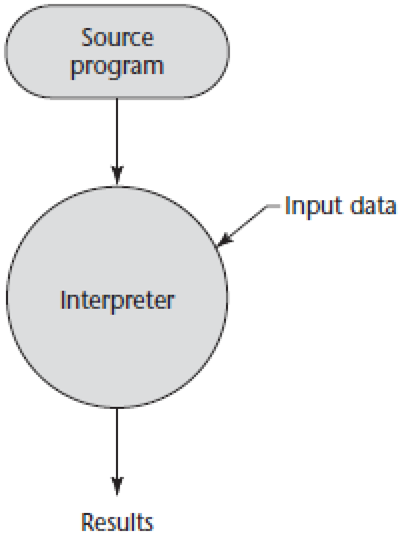

Figure 2 : Pure Interpretation

Advantage:

1. Pure interpretation has the advantage of allowing easy implementation of many source-level debugging operations, because all run-time error messages can refer to source-level units.
2. Interpreted implementations of languages tend to be more portable.
3. Interpreted implementations of languages are generally easier to create because writing compilers is difficult.
4. Debugging is easier.

Disadvantage:

1. It is about 10 to 100 times slower than compilation process, since, no matter how many times a statement is executed, it must be decoded every single time.
2. The symbol table has to be present during interpretation, so it requires more space.

3) Hybrid Implementation Systems: Hybrid implementation systems are a compromise between compilers and pure interpreters. This method translates high-level language programs into an intermediate language form. Instead of translating intermediate language code to machine readable code, it simply interprets the intermediate code. Perl, Java etc are languages that use this implementation method.

The process used in these implementation systems is as below:

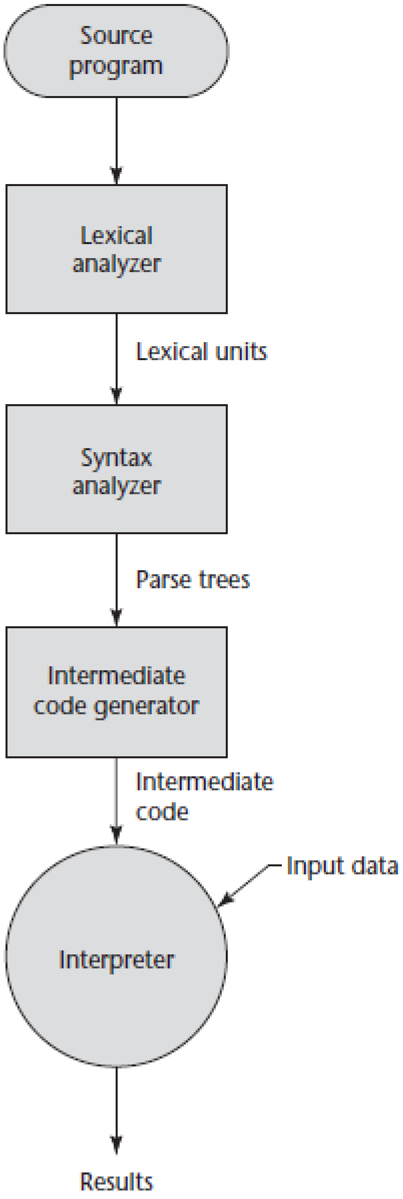

Figure 3 : Hybrid Implementation System

Here, the lexical analyzer, syntax analyzer and the intermediate code generator perform the same jobs as that of compilers.

Advantage:

1. This method is designed to allow easy interpretations.
2. Since the source codes are decoded only once, hybrid implementation systems are much faster than pure interpretation.

3. Sometimes system designer may provide both compiler and interpreter implementations for a language. In such cases, the interpreter is used to develop and debug programs. Then, after a bug-free state is reached, the programs are compiled to increase their execution speed.

Disadvantage:

There are not many disadvantages of a hybrid implementation system except that they're a little slower if compared to compilers. Still, not really much slow.

References:

1. Concepts of Programming Languages, 10th edition - Robert W. Sebesta
2. https://www.programmerinterview.com/general-miscellaneous/whats-the-difference-between-a-compiled-and-an-interpreted-language/
3. https://www.techwalla.com/articles/disadvantages-advantages-of-compilers
4. Qualitative Assessment of Compiled,Interpreted and Hybrid Programming Languages https://pdfs.semanticscholar.org/7cb2/e2fba33e3a47242ada469d20562314f3238c.pdf