

Course Name : O Level(B3-1st sem.)
Topic : Language Processor

Subject : ITT&NB
Date : 06-04-20

Language Processors: Assembler, Compiler and Interpreter

Language Processors –

A special translator system software is used to translate the program written in high-level language into machine code is called **Language Processor** .

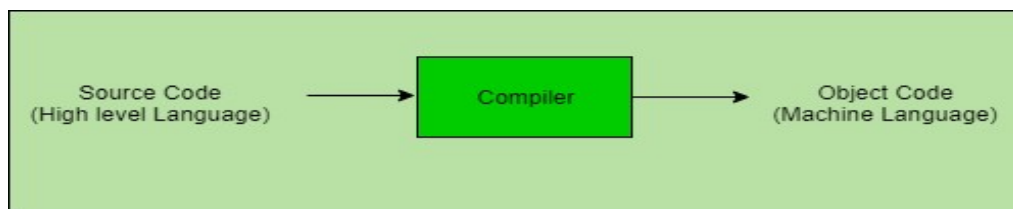
The language processors can be any of the following three types:

1. **Compiler –**

The language processor that reads the complete source program written in high level language as a whole in one go and translates it into an equivalent program in machine language is called as a Compiler.

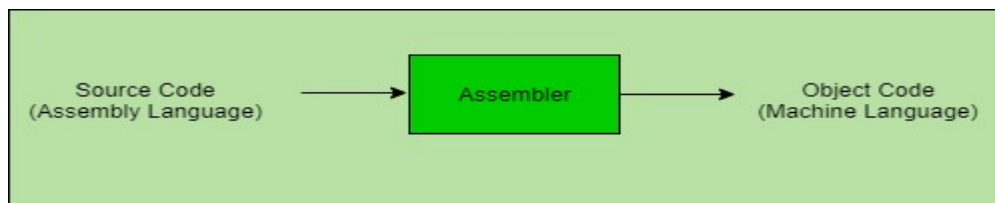
Example: C, C++, C#, Java

In a compiler, the source code is translated to object code successfully if it is free of errors. The compiler specifies the errors at the end of compilation with line numbers when there are any errors in the source code. The errors must be removed before the compiler can successfully recompile the source code again.>



2. **Assembler –**

The Assembler is used to translate the program written in Assembly language into machine code. The source program is a input of assembler that contains assembly language instructions. The output generated by assembler is the object code or machine code understandable by the computer.



3. **Interpreter –**

The translation of single statement of source program into machine code is done by language processor and executes it immediately before moving on to the next line is called an interpreter. If there is an error in the statement, the interpreter terminates its translating process at that statement and displays an error message. The interpreter moves on to the next line for execution only after removal of the

error. An Interpreter directly executes instructions written in a programming or scripting language without previously converting them to an object code or machine code.

Example: Perl, Python and Matlab.

Difference between Compiler and Interpreter –

COMPILER	INTERPRETER
A compiler is a program which converts the entire source code of a programming language into executable machine code for a CPU.	Interpreter takes a source program and runs it line by line, translating each line as it comes to it.
Compiler takes large amount of time to analyze the entire source code but the overall execution time of the program is comparatively faster.	Interpreter takes less amount of time to analyze the source code but the overall execution time of the program is slower.
Compiler generates the error message only after scanning the whole program, so debugging is comparatively hard as the error can be present anywhere in the program.	Its Debugging is easier as it continues translating the program until the error is met
Generates intermediate object code.	No intermediate object code is generated.
Examples: C, C++, Java etc.	Examples: Python, Perl

Exercise:

Write short notes on assembler, compiler and interpreter