TRANSLATOR

TRANSLATOR

1. A translator is a programming language processor that modifies a computer program from one language to another.
2. It takes a program written in the source program and modifies it into a machine program. It can find and detect the error during translation.

Purpose of Translator

1. It translates a high-level language program into a machine language program that the central processing unit (CPU) can understand. It also detects errors in the program.

DIFFERENT TYPES OF TRANSLATORS

There Are 3 Different Types Of Translators As Follows:

1. Compiler
2. Interpreter
3. Assembler

COMPILER

1. A compiler is a translator used to convert high-level programming language to low-level programming language.
2. It converts the whole program in one session and reports errors detected after the conversion.
3. The compiler takes time to do its work as it translates high-level code to lower-level code all at once and then saves it to memory.
4. A compiler is processor-dependent and platform-dependent.
5. But it has been addressed by a special compiler, a cross-compiler and a source-to-source compiler.
6. Before choosing a compiler, the user has to identify first the Instruction Set Architecture (ISA), the operating system (OS), and the programming language that will be used to ensure that it will be compatible.

Advantages of the compiler:

1. The whole program is validated so there are no system errors.

2. The executable file is enhanced by the compiler, so it runs faster.
3. User do not have to run the program on the same machine it was created.

Disadvantages of the compiler:

1. It is slow to execute as you have to finish the whole program.
2. It is not easy to debug as errors are shown at the end of the execution.
3. Hardware specific, it works on specific machine language and architecture.

INTERPRETER

1. Just like a compiler, is a translator used to convert high-level programming language to low-level programming language.
2. It converts the program one at a time and reports errors detected at once while doing the conversion.
3. With this, it is easier to detect errors than in a compiler. An interpreter is faster than a compiler as it immediately executes the code upon reading the code.
4. It is often used as a debugging tool for software development as it can execute a single line of code at a time.
5. An interpreter is also more portable than a compiler as it is not processor-dependent, you can work between hardware architectures.

Advantages of the interpreter:

1. You discover errors before you complete the program, so you learn from your mistakes.
2. Program can be run before it is completed so you get partial results immediately.
3. You can work on small parts of the program and link them later into a whole program.

Disadvantages of the interpreter:

1. There's a possibility of syntax errors on unverified scripts.
2. Program is not enhanced and may encounter data errors.
3. It may be slow because of the interpretation in every execution.

ASSEMBLER

1. An assembler is is a translator used to translate assembly language to machine language.

2. It is like a compiler for the assembly language but interactive like an interpreter. Assembly language is difficult to understand as it is a low-level programming language.
3. An assembler translates a low-level language, an assembly language to an even lower-level language, which is the machine code.
4. The machine code can be directly understood by the CPU.

Advantages of the assembler:

1. The symbolic programming is easier to understand thus time-saving for the programmer.
2. It is easier to fix errors and alter program instructions.
3. Efficiency in execution just like machine level language.

Disadvantages of the assembler:

1. It is machine dependent, cannot be used in other architecture.
2. A small change in design can invalidate the whole program.
3. It is difficult to maintain.