Submitted by, Abiya Baby

# TRANSLATORS

Translators are software tools that convert code written in one programming language (usually high-level) into another form, often machine code or another programming language. This allows computers to understand and execute the instructions provided by humans. Translators ensure that programs written in high-level languages are executable on computer systems.

## Types of Translators

Translators are generally categorized into three main types:

### 1. Compiler

- **Definition**: Converts an entire high-level source code into machine code (binary) at once.
- **Working**:
    1. **Lexical Analysis**: Breaks down the code into tokens (keywords, variables, etc.).
    2. **Syntax Analysis**: Checks the code structure for errors (grammar of the language).
    3. **Semantic Analysis**: Ensures the code has meaningful instructions (e.g., data type consistency).
    4. **Intermediate Code Generation**: Creates an intermediary representation of the code.
    5. **Code Optimization**: Enhances code for better performance.
    6. **Code Generation**: Converts the optimized intermediate code into machine language.
- **Example**: GCC for C/C++, Java Compiler.

### 2. Interpreter

- **Definition**: Executes the high-level source code line-by-line, without producing a separate machine code file.
- **Working**:
    1. Reads the first line of code.
    2. Converts it to machine code.
    3. Executes it immediately.
    4. Repeats the process for the next line.
- **Example**: Python Interpreter, JavaScript Engine (V8).

### 3. Assembler

- **Definition**: Converts assembly language (low-level language) into machine code.
- **Working**:
    1. Reads the assembly instructions (mnemonics like MOV, ADD).
    2. Maps these instructions to corresponding machine code.

3. Outputs a binary file executable by the CPU.

- **Example**: NASM (Netwide Assembler), MASM (Microsoft Macro Assembler).