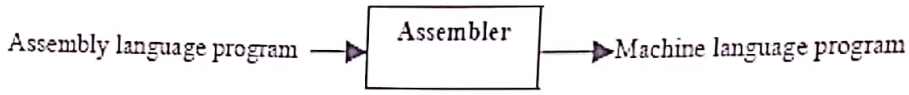
1.2 Different System Softwares

1. Assembler

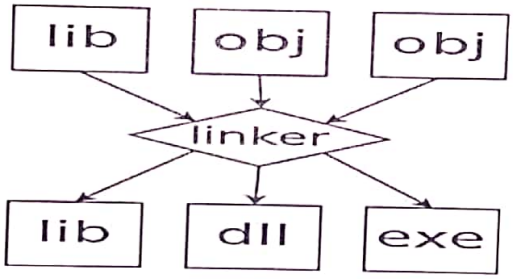
- Programmers found it difficult to write or read programs in machine language.
- In a quest for a convenient language, they began to use a mnemonic (symbol) for each machine instructions which would subsequently be translated into machine language. Such a mnemonic language is called Assembly language.
- Programs known as Assemblers are written to automate the translation of assembly language into machine language.



- Fundamental functions:
  1. Translating mnemonic operation codes to their machine language equivalents.
  2. Assigning machine addresses to symbolic tables used by the programmers.

2. Linker

Combines two or more separate object programs and supplies the information needed to allow references between them



3. Loader

Loader is a system program that loads the object program into memory for execution.

4. Macro processor

- A **macro** represents a commonly used group of statements in the source programming language.
- A macro instruction is simply a notational convenience for the programmer to write a shorthand version of a program.
- The macro instruction is replaced by the macro processor with the corresponding group of source language statements. This operation is called "expanding the macro"
- For example:

  – Suppose it is necessary to save the contents of all registers before calling a subroutine.

– This requires a sequence of instructions.
– We can define and use a macro, SAVEREGS, to represent this sequence of instructions.

- A **macro processor** functions essentially involve the substitution of one group of characters or lines for another. Normally, it performs no analysis of the text it handles.
- Macro processors are used in
  - assembly language
  - high-level programming languages, e.g., C or C++

5. Text editor

- An Interactive text editor has become an important part of almost any computing environment
- Text editor acts as a primary interface to the computer for all type of "knowledge workers" as they compose, organize, study, and manipulate computer-based information
- A text editor allows you to edit a text file (create, modify etc…)
- Text editors on Windows OS

  -Notepad, WordPad, Microsoft Word
- Text editors on UNIX OS

  - vi, emacs, jed, pico,

6. Debugger

- Tests & debugs errors
- Debugging is a methodical process of finding and reducing the number of bugs, or defects, in a computer program
- An interactive debugging system provides programmers with facilities that aid in testing and debugging of programs
- Conditional Expressions – Programmers can define some conditional expressions ,evaluated during the debugging session, program execution is suspended, when conditions are met, analysis is made, later execution is resumed
- Breakpoint- The programmer may define break points which cause execution to be suspended, when a specified point in the program is reached. After execution is suspended, the debugging command is used to analyze the progress of the program and to diagnose errors detected.

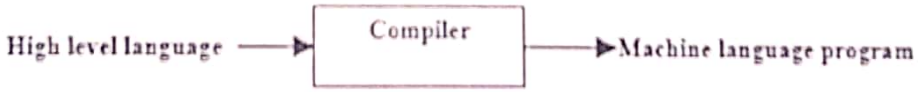
- A Debugging system should also provide functions such as tracing and trace back. Trace back can show the path by which the current statement in the program was reached. It can also show which statements have modified a given variable or parameter.

7. Device driver

- They are distinct "black boxes" that make a particular piece of hardware respond to a well-defined internal programming interface
- They hide completely the details of how the device works.
- User activities are performed by means of a set of standardized calls that are independent of the specific driver; mapping those calls to device-specific operations that act on real hardware is then the role of the device driver.
- This programming interface is such that drivers can be built separately from the rest of the kernel, and "plugged in" at runtime when needed.
- This modularity makes Linux drivers easy to write, to the point that there are now hundreds of them available.

8. Compiler

- A compiler is a program that translates programs written in any high level language into its equivalent machine language program.
- It bridges the semantic gap between a programming language domain and the execution domain.
- The program instructions are taken as a whole.



9. Interpreter

- It is a translator program that translates a statement of high-level language to machine language and executes it immediately. The program instructions are taken line by line.
- The interpreter reads the source program and stores it in memory
- During interpretation, it takes a source statement, determines its meaning and performs actions which implements it. This includes computational and I/O actions.
- Program counter (PC) indicates which statement of the source program is to be interpreted next. This statement would be subjected to the interpretation cycle.
- The interpretation cycle consists of the following steps:

- Fetch the statement.
- Analyze the statement and determine its meaning.
- Execute the meaning of the statement.

- The following are the characteristics of interpretation:
  - The source program is retained in the source form itself, no target program exists.
  - A statement is analyzed during the interpretation.

10. Operating system

- It is the most important system program that act as an interface between the users and the system.
- It makes the computer easier to use.
- It provides an interface that is more user-friendly than the underlying hardware.
- The functions of OS are:
  - Process management
  - Memory management
  - Resource management
  - I/O operations
  - Data management
  - Providing security to user's job.