

## High-level-Language(HLL):-

High-Level-programming language is one type of programming language available. The other type of programming language is known as low-level-language or assembly language.

- High level-language is easier to learn and understood than the assembly language, because high level languages uses names and commands the resemble English, while the assembly language uses mnemonic codes.

- Some of the common high-level-language are:

Fortran(Formula Translation) for engineers.

COBOL(Common Business Oriented Language) for business programmer's.

Basic (Beginner's All-purpose symbolic Instruction) Code) for engineer's and scientists.

Pascal.

Unlike assembly programs, high-level-languages programs may be used with different makes of computers, while the assembly languages are machine oriented.

Other advantages of high-level-languages are

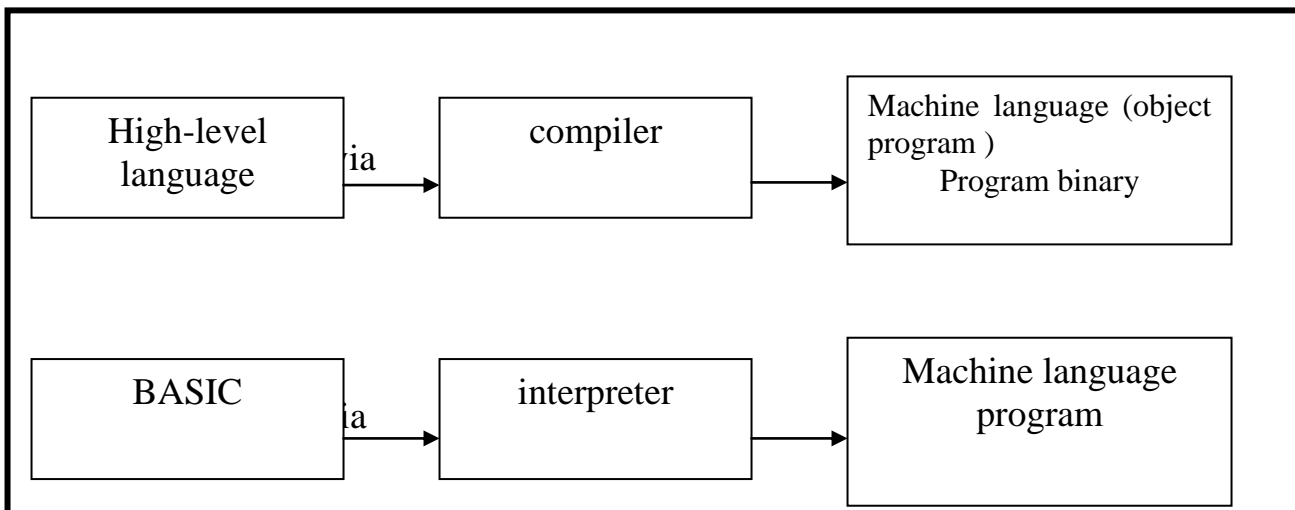
- 1.They are easier to learn than assembly languages.
- 2.They are easier to use for problem solving, than assembly.
- 3.They require less time to write, than assembly.
- 4.They provide better documentation.
- 5.They are easier to maintion.

## Compiler Translations:-

Naturally, a source program written in a high-level language, must also be translated into a machine usable code, that can be executed by the computer. A translating program that can perform this operation is called " compiler".

The compiler:- converts the entire source high level languages program into a machine language object program, before the program is executed as in FORTRAN.

Another type of translating program called " Interpreter" with the interpreter each statement line is translated and executed immediately(one statement at a time) as with Basic.

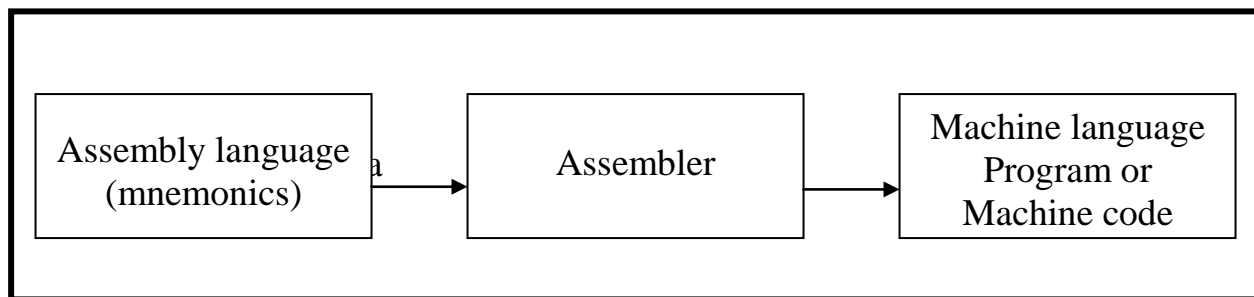


Therefore the interpreter is a slower than compiler.

## Assembly language

Assembly language program using instruction abbreviation called mnemonics, such as LD(load), ST(store) and ADD(Add to Accumulator). This is converted to machine language program with a translator called " assembler" .

Assembler:- is a special program that enables assembly language program(some times called source program) into own machine language program(some times called an object code).



Note:- Compilers are similar to assemblers in the fact. That they take the source program and convert(translate) it in to an executable machine language program (object program).

Type of software:- software can be divided into two main types:

**1)System software.**

In general includes all programs designed to help programmers or to control the computer system. System software includes:

- a. Operating system (O.S).
- b. Programming languages(H.L.L&L.L.L).
- c. The language processor translator(interpreter, compiler and Assembler).
- d. Utility (Service) programs.

**2)Application system(software):-**

This type of softwares , comprises programs written to performs a specific tasks for the user, and can be divided as:-

- a. Application systems: this type of systems are written by a special software houses.  
e.g. bank systems, salary system, airlines system, information banks.
- b. Application programs: this programs direct the computer to accomplish specific activities and it in general small programs and called user programs. e.g. writing a letter using word processor aided design packages.

**System software:-**

System software has man objectives:-

- Making it easier for the user to get a job into a computer.
- Making it easier for the user to run the job and test it when it is in the computer.

**Operating system(O.S):-** is asset of inter-related by provided by a computer infect that control the basic operations of the computer(such as loading) and manage the execution of the programs. It also store and retrieves files on disk.

The main parts of an operating system are:

- a. Supervisor.
- b. Input / output manager.
- c. File manager.
- d. Command processor.

**Supervisor:** At the heart of all operating systems is the supervisor program, it schedules and coordinates is running. Its supervisor program is loaded in the internal memory, directing and controlling.

**I/O manager:** In general, all data transformed to and form peripheral devices are filtered through the I/O manager. It is insulates the rest of the programs in the computer from the special of the peripheral devices. For example the I/O manager might translate the keyboard character codes into the coding system used by the rest of the computer.

With a good O.S, it is possible to add a hard disk or a faster printer to the computer system just by modifying the I/O manager without making any changes to other software).

This is called device independence. An even stronger from of hardware independence, called machine independence (or software portability). Allows application software to be moved from one type of computer system to another without programming changes.

**File manager:** anything on disk is stored in file. Each file has its. Own name and stores one type of information, either program or data. A data file might be a digitized picture, while a program file might be a Basic program or word processor.

Whatever the contents of a file. The file manager takes care of saving, deleting, coping, loading.

As we know data are stored on disks as individual bytes, grouped into sectors, with each sector forming part of a track people think of a data file in logical terms as a letter are stored physically on the disk in tracks and sectors.

The file manager translates between the logical and the physical arrangement of data maintaining a file allocation table (FAT) for each disk.

The file allocation table (FAT) is an index telling where each file is stories.

Another function of the file manager is to format(also called initialize) disk. Formatting a disk involves erasing the disk and giving it an empty file allocation table.

Disk come from the manufacture in blank (or un formatted) condition, files can not be stored on an unformatted disk.

Disk cannot be used until it has been formatted, but formatting a disk by mistake erases its content completely.

### **Command processor:-**

The command processor communicates between the user and the rest of the operating system. It accepts commands and the user makes sure they are valid, and than takes the appropriate action.

Ex:- If the ask to copy(file name) and call the new file, the command processor will translate the command and can the request to the file manager. If the disk does not have enough room to store, then the file manager sends a coded error massage, which the command processor might translate to, Insufficient free spare on disk command a aborted.