# What is Language Processing Systems in Compiler Design?

In a language processing system, the source code is first preprocessed. The modified source program is processed by the compiler to form the target assembly program which is then translated by the assembler to create relocatable object codes that are processed by linker and loader to create the target program. It is based on the input the translator takes and the output it produces, and a language translator can be defined as any of the following.
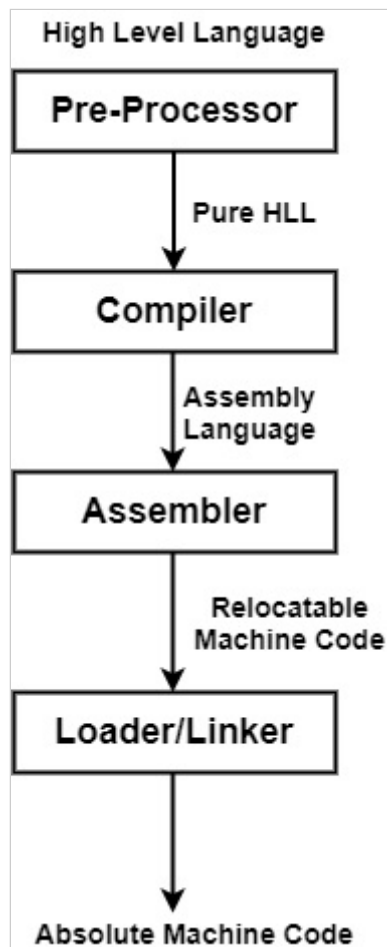
High-Level Language – If a program includes #define or #include directives, including #include or #define, it is known as HLL.

Pre-Processor – The pre-processor terminates all the #include directives by containing the files named file inclusion and all the #define directives using macro expansion. A pre-processor can implement the following functions –

- **Macro processing** – A preprocessor can enable a user to define macros that are shorthands for higher constructs.

- **File inclusion** – A preprocessor can include header files into the program text.

- **Rational preprocessor** – These preprocessors augment earlier languages with additional current flow-of-control and data structuring facilities.

- **Language Extensions** – These preprocessors try to insert capabilities to the language by specific amounts to construct in macro.

Pure HLL – It means that the program will not contain any # tags. These # tags are also known as preprocessor directives.

Assembler – Assembler is a program that takes as input an assembly language program and changes it into its similar machine language code.

Assembly Language – It is an intermediate state that is a sequence of machine instructions and some other beneficial record needed for implementation. It neither in the form of 0's and 1's.

Advantages of Assembly Language

- Reading is easier.

- Addresses are symbolic & programmers need not worry about addresses.

- It is mnemonics. An example we use ST instead of 01010000 for store instruction in assembly language.

- It is easy to find and correct errors.

Relocatable Machine Code – It means that you can load that machine code at any point in the computer and it can run. The address inside the code will be so that it will maintain the code movement.

Loader / Linker – This is a code that takes as input a relocatable program and compiles the library functions, relocatable object records, and creates its similar absolute machine program.

The loading includes taking the relocatable machine program, changing the relocatable addresses, and locating the modified instructions and information in memory at the suitable area.

Linking enables us to create a single program from several documents of a relocatable machine program. These documents can have been resulting in different compilations, one or more can be library routines supported by the system available to any code that requires them.