UNIT--2

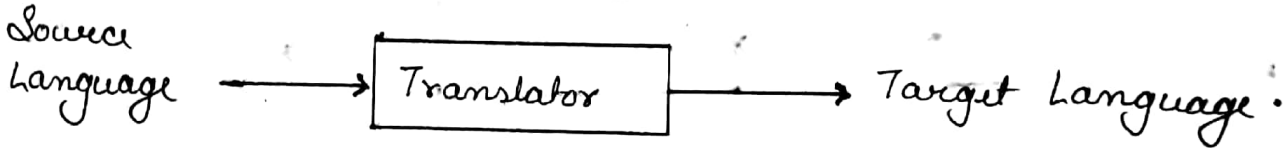
INTRODUCTION TO ASSEMBLER

Translator :-

Translator is a system tool which converts source language into target language. Occasionely target Language is Machine Language.

There are two works of translator are :-

1) Convert source language into target language.
2) Error diagonostic.



Source Language may be high-level or Assemly language.

There are three (3) types of Translator :-

* 1) Assembler
* 2) Compiler.
* 3) Interpretor.

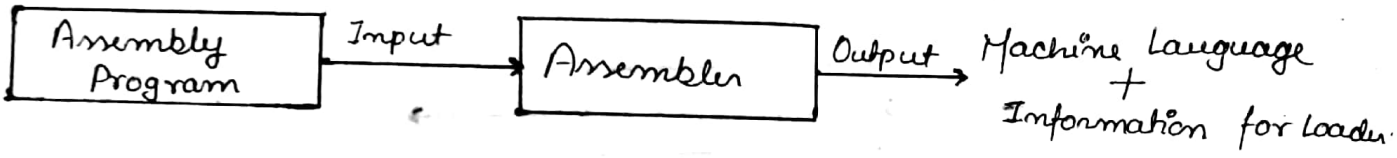
There are some s/w tools which supports translator in conversion, these are

1) Macroprocessor.

2) ~~Compiler~~ Linker.

3) Loader.

Assembler : Assembler is a System S/w which takes input as an assembly program & converts into Machine Language & information for Loader.

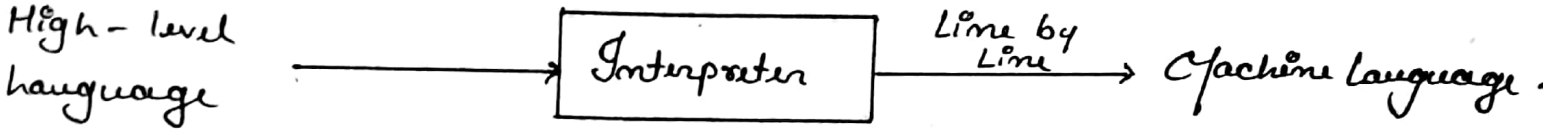


Information for Loader includes additional Library functions which added in Assembly program. It stores the information as that Library function & whenever required it transfer at least as the Library function.

Compiler :) It is a System Software in which a high-level language must be translated into Machine Language.

* Converts the entire program to machine code, when all the syntax errors are removed execution takes place.

Interpreter :- It is a system software which translates the program statement by statement.



- Each time the program is executed, every line is checked for syntax error & then converted to equivalent Machine Language.

S/W tools which support translator in conversion these are:-

i) Macroprocessor.
ii) Linker.
iii) Loader.

i) Macroprocessor or)

*) Many programs contain sequence of instructions which are repeated in identical form.

*) The repetitious writing of such sequence is controlled by the Macroprocessor.

*) Macroprocessor allows a sequence of Source language code to be defined once & then referred by name each time it is to be referred.

*) Each assembler support Macro.

Example:-

```
A    1,X   ADD content of X to register 1.
A    1,X   "   "   "  "  "  "  "  "  2.
A    1,X   "   "   "  "  "  "  "  "  1
A    2,X   "   "   "  "  "  "  "  "  1.
X    D.C   F   '6'  Actual Value of Y,X.
```

In the above program the sequence

A 1,X
A 2,X occurs twice. It can occur many time also.

A Macro facility permit us to attach a name to this sequence & to use this name in it's place.

Such as:-

```
Macro  CNTR  (For Ex-)
                any name
A   1, X .
A   2,X .
MEND          (End of Macro defn).
```

ii) Linker or

* Usually a longer program is divided into a no of smaller subprograms called Modules.
* It is easier to develop, test & debug smaller programs.
* A linker is a program, that links (combines) smaller programs to form a single program.

While developing a program, sub-routines are frequently used. The sub-routines are stored in Library file. The linker also links sub-routines with the main program.

iii) Loader or

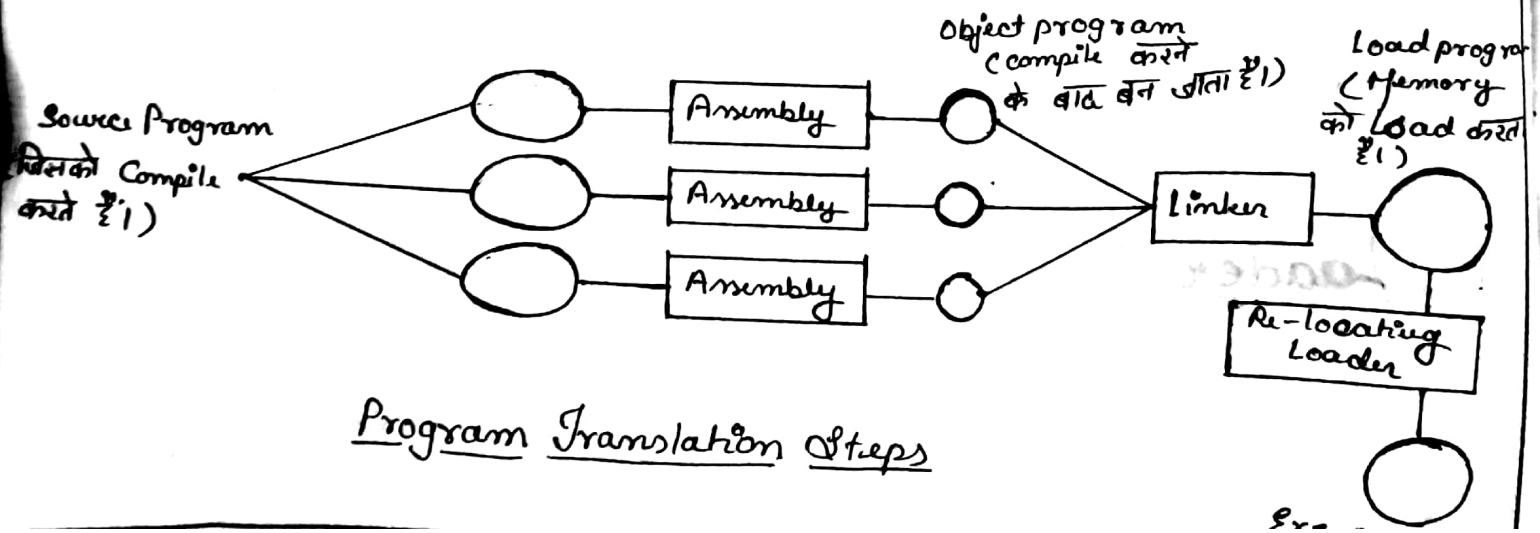
A loader is a program that places programs into Main Memory & prepare them for execution.

The Loader's target Language is Machine Language. It's Source Language is nearly Machine Language.

Loading is ultimately bound with the storage management function of O.S & is usually performed later than Assembly or compilation.

- The period of executions of user's programs is called Execution Time.
- The period of translating a user's source program is called Assembly or Compile Time.
- The period of Loading & preparing an object program for execution called Wait time.

Assembler Implementation :-



Program Translation Steps

Assembler is a program which takes input as an assembly program converts it into machine language, there are implemented as follow.

* There are three entity (existence) of assembler as follows →

① Absolute Entity →

It includes the operation code (op code), operands & location of members as the Assembly program. The addresses are independent from the binary code (Exe). They are fixed for the time of conversion.

② Relative Entity →

It has the information of symbol table & Memory Storage.

The addresses are fixed conversion are not started. When the conversion begin the address are varies. It make take the address of exe program also.

③ Object program :-)

It identify all internal & external references of the assembly program. It also includes the Internal reference of the External function.

Relocating Loader :-)

It's main work is to give relative address of program into absolute address in Memory location. It places all program at one location for the translation into binary code.

Linker :-)

It collect all object program addresses at one place so that they are easy to Load into relocation loader & this process is Load program.

There are three types of Assembler :-)

① Load & go Assembler :-)

This is a simplest assembler. It collects all external references & main program at fixed location & then load it into Main Memory for the

execution.

In this assembler the address are fixed for the machine code before translation & cannot be change in any circumstances.

It has two main Drawbacks :-

* No program modular development
* No generation of Symbol table.

② One pass Module Assembler :-

In one-pass assembler mnemonics, symbol, operands & Memory location are translated into binary code in single pass.

There are no symbol table generated in one-pass module & assembler. It is very slow to work. It needs slow secondary device or minimal use of Secondary devices.

③ Two-pass Module Assembler :-

In two-pass module two pass are :-

One-pass :-

→ Translate mnemonics symbol, operands.
→ General Symbol Tables.

2) Define Memory location

Pass-two :-

2) Generate object program

Example of two-pass assembler :-

Line	Address	Label	Operation	Operand1	Operand2
1	30		COPY	ZERO	OLDER
2	32	FRONT	Load	OLDER	
3	33	FINAL	WRITE	NEW	
4	36	OLDER	SPACE		
5	38	ZERO			

Symbol Table :-

ZERO 30

OLDER 32 etc.