

1A)

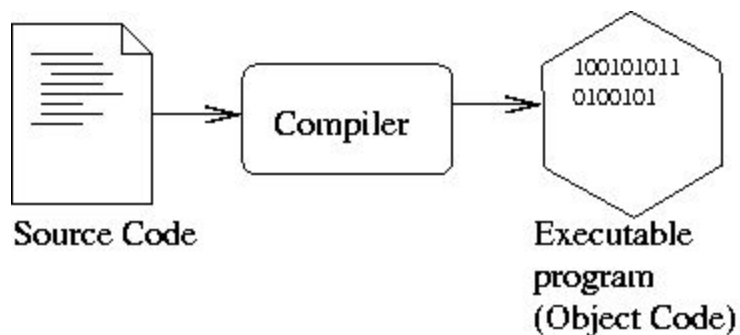
AIM: Study of various System Programs like Assembler, Linker, Loader, Interpreter, Compiler, Macro, etc.

System Programs

The system programs are used to program the operating system software. While application programs provide software that is used directly by the user, system programs provide software that are used by other systems such as SaaS applications, computational science applications etc.

1) Compilers:

- A compiler is a computer program (or a set of programs) that transforms source code written in a programming language (the source language) into another computer language (the target language).
- Typically, from high level source code to low level machine code or object code.



2) Interpreter:

- An interpreter is a common kind of language processor. Instead of producing target program as a translation, an interpreter appears to directly execute the operations specified in the source program on inputs supplied by the user.
- In contrast, an interpreter reads a statement from the input, converts it to an intermediate code, execute it, then takes the next statement in sequence.

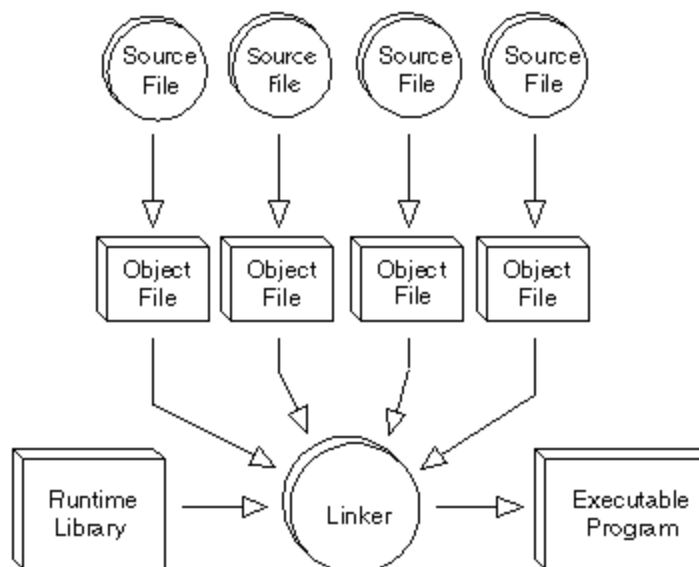
- If an error occurs, an interpreter stops execution and reports it.

3) Assembler:

- An assembler translates assembly language programs into machine code. The output of a assembler is called an object file, which contains a combination of machine instruction as well as the data required to place these instructions in memory.

4) Linker:

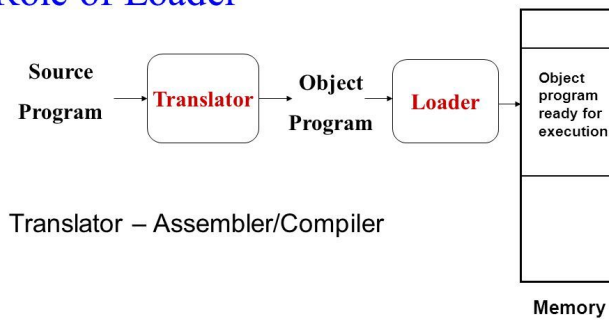
- Linker is a computer program that links and merges various object files together in order to make an executable file. All these files might have been compiled by separate assembler.
- The major task of a linker is to search and locate referenced module/routines in a program and to determine the memory location where these codes will be loaded making the program instruction to have absolute reference.



5) Loader:

- Loader is a part of operating system and is responsible for loading executable files into memory and execute them.
- It calculates the size of a program (instructions and data) and create memory space for it. It initializes various registers to initiate execution.

Role of Loader



-

6) Macro:

- Macro is a single line abbreviation for group of instructions.
- With the help of macro, programmer can define a single "instruction" to represent block of code.

1A)

AIM: To create your own library file in C/C++/Java

Library.java

```
1. package com.lib;
2.
3. public class Library {
4.     public static int add(int ...x){
5.         int a = 0;
6.         for(int s : x){
7.             a += s;
8.         }
9.         return a;
10.    }
11.
12.    public static float divide(int x, int y){
13.        try{
14.            return x / y;
15.        }
16.        catch(Exception e){
17.            System.out.println(e);
18.        }
19.        return 0;
20.    }
21.
22.    public static int sub(int x, int y){
23.        return x - y;
24.    }
25.
26.    public static int cube(int x){
27.        return x * x * x;
28.    }
29.
30.    public static double pow(int x, int y){
31.        return Math.pow(x, y);
32.    }
33.
34.    public static int mul(int x, int y){
35.        return x * y;
36.    }
37. }
```

App.java

```
1. package com.main;
2.
3. import com.lib.Library;
4.
5. public class App {
6.     public static void main(String[] args) {
7.         int a = 5;
8.         int b = 3;
9.         System.out.println("Addition: " + Library.add(5, 6, 7, 8));
10.        System.out.println("Subtraction: " + Library.sub(a, b));
11.        System.out.println("Division: " + Library.divide(a, b));
12.        System.out.println("Power: " + Library.pow(a, b));
13.        System.out.println("Cube: " + Library.cube(a));
14.        System.out.println("Multiplication: " + Library.mul(a, b));
15.    }
16. }
```

Output:

Addition: 26

Subtraction: 2

Division: 1.0

Power: 125.0

Cube: 125

Multiplication: 15