Compiler Design - Introduction

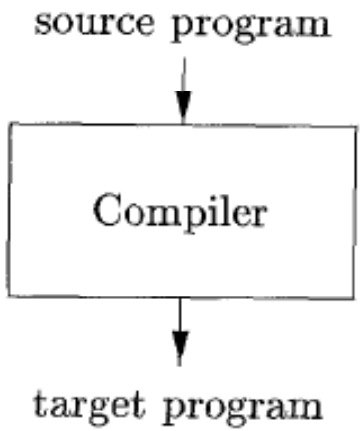
Lecture – I

References: Compilers: Principle, techniques & tools book and other Internet resources

Compiler

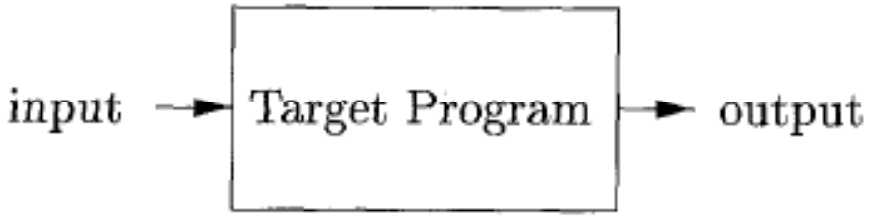
- It is a translator to convert a source language in to the target language.

- It is a program which translates a high level language program into a machine language program.



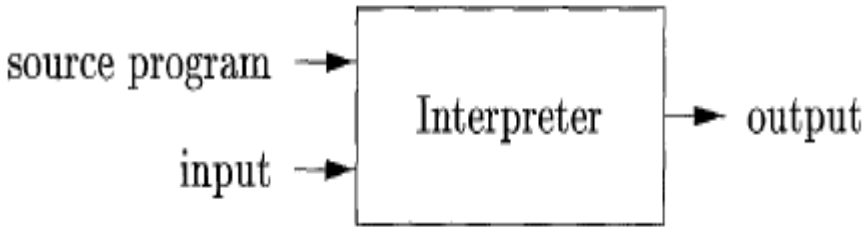
Target Program

- If the target program is the executable file then user will call the executable file to get the output by passing the input.



Interpreter

- An interpreter is another common kind of language processor.



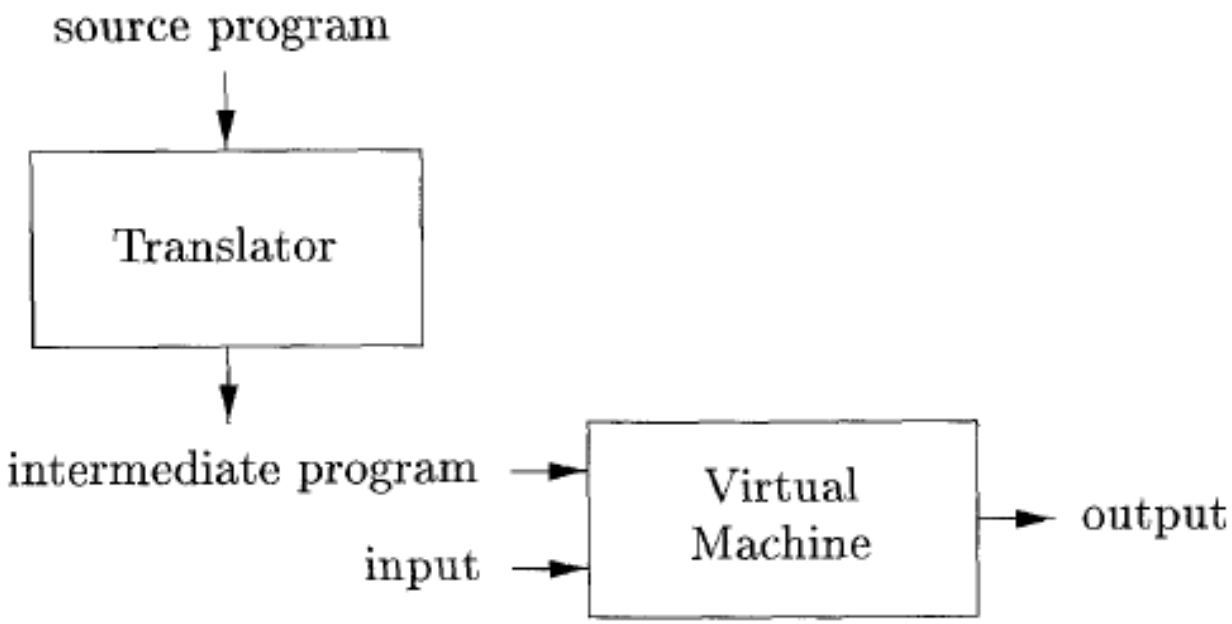
- Instead of producing a target program as a translation, an interpreter appears to directly execute the operations specified in the source program on inputs supplied by the user.

C vs. I

- The machine-language target program produced by a compiler is usually much **faster** than an interpreter at mapping inputs to outputs .

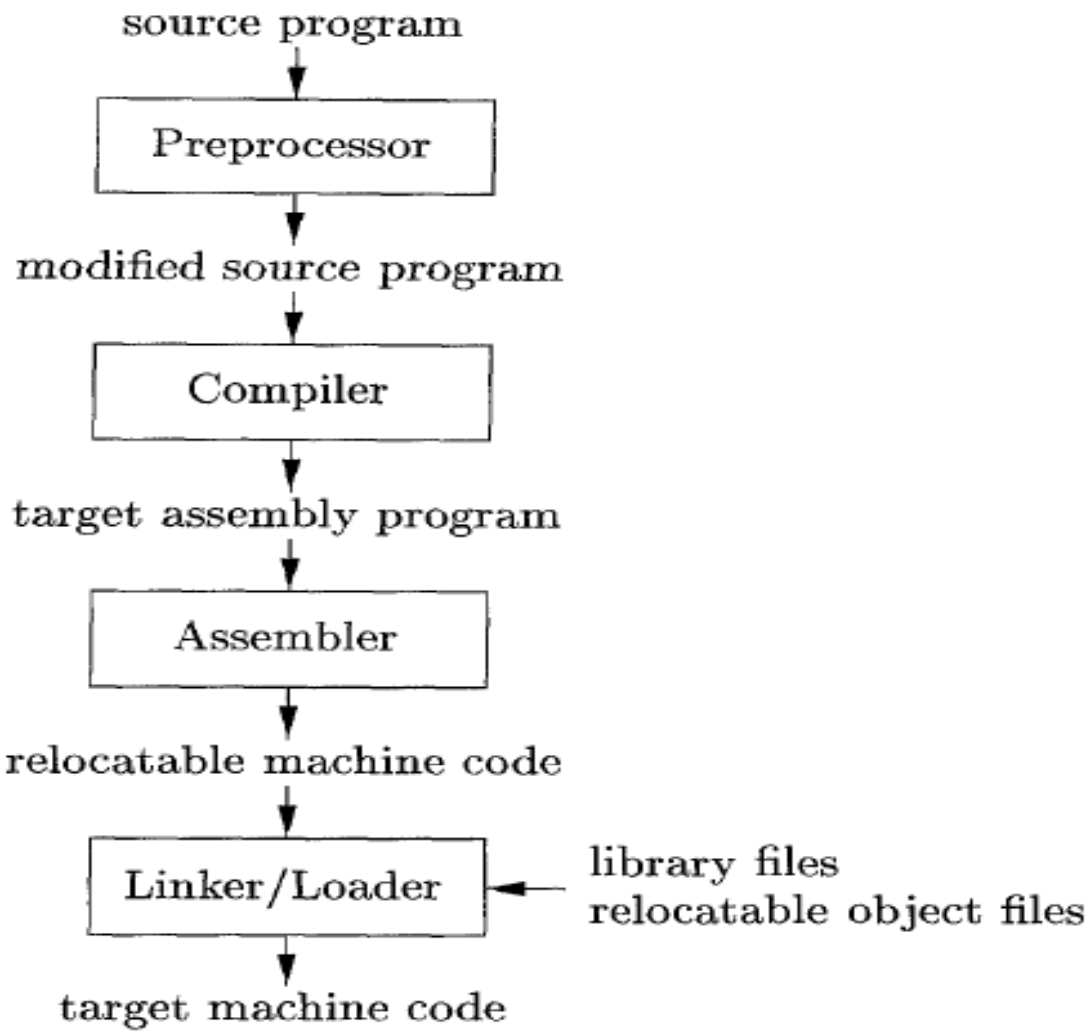
- An interpreter, however, can usually give better **error diagnostics** than a compiler, because it executes the source program statement by statement.

Hybrid Compiler



In order to achieve faster processing of inputs to outputs, some Java compilers, called *just-in-time compilers, translate the bytecodes into machine language* immediately before they run the intermediate program to process the input.

Language Processing System



Preprocessor

- A source program may be divided into modules stored in separate files.
- The task of collecting the source program is sometimes entrusted to a separate program, called a *preprocessor.*
- The preprocessor may also expand macros, into source language statements.

Compiler

- The modified source program is then fed to a compiler.

- The compiler may produce an assembly-language program as its output, because **assembly language is easier to produce as output and is easier to debug.**

Assembler

- The assembly language is then processed by a program called an *assembler that produces* **relocatable machine code** as its output.

Linker

- Large programs are often compiled in pieces, so the relocatable machine code may have to be linked together with other relocatable object files and library files into the code that actually runs on the machine.

- The *linker resolves* external memory addresses, where the code in one file may refer to a location in another file.

Loader

- The *loader then puts together all of the executable object files* into memory for execution.

- This will start at the time of execution.

1: Example Input/Output to a preprocessor

Input To Preprocessor

```
// This is a test Example with Preprocessing
#define Print printf(...)
#define Ret  return 0
#define BEGIN   {
#define END     }

Print;
int main(void)
        BEGIN
        printf("Hello World");  Ret;
        END
```

Output Of Preprocessor:

```
// This is a test Example with Preprocessing
printf(...);
int main(void)
        {
        printf("Hello World");  return 0;
        }
```

Linking

- Dynamic linking is performed at run time by the operating system.
- Static linking is performed by programs called linkers as the last step in compiling a program.
- Linkers are also called link editors.

Object Files

- **Relocatable object file**, which contains binary code and data in a form that can be combined with other relocatable object files at compile time to create an executable object file.

- **Executable object file**, which contains binary code and data in a form that can be directly loaded into memory and executed.

- **Shared object file**, which is a special type of relocatable object file that can be loaded into memory and linked dynamically, either at load time or at run time.

Thank You