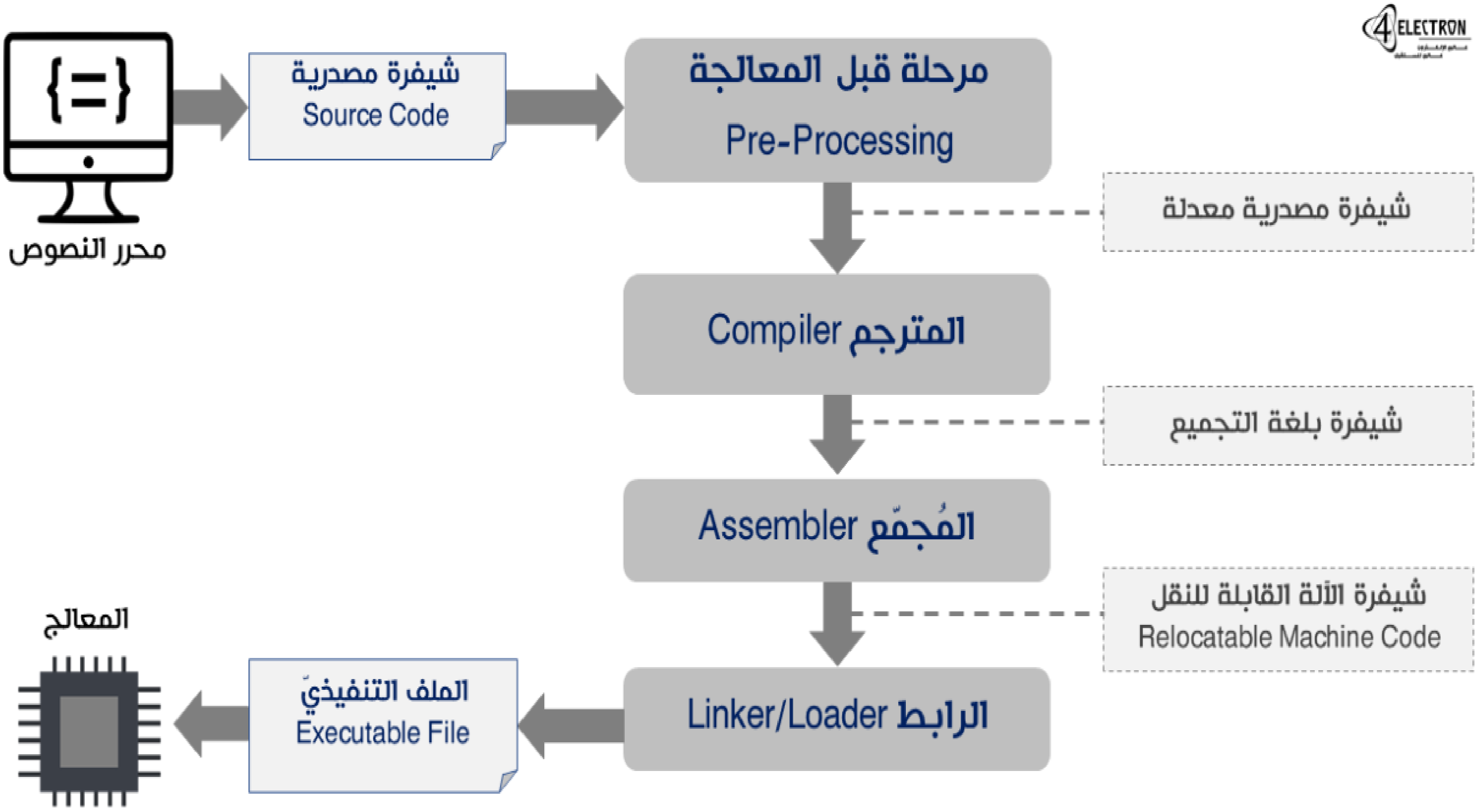
Basic Language Translator

FIRST LECTURE

Language Processing System نظام مُعالجة لغات البرمجة



محرر النصوص

شيفرة مصدرية
Source Code

مرحلة قبل المعالجة
Pre-Processing

شيفرة مصدرية معدلة

Compiler المترجم

شيفرة بلغة التجميع

Assembler المُجمّع

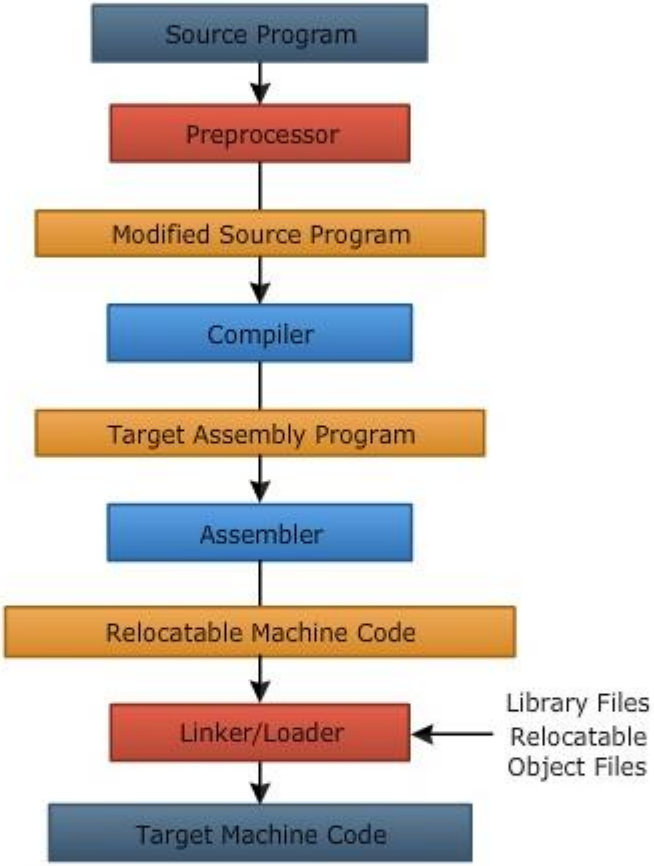
شيفرة الآلة القابلة للنقل
Relocatable Machine Code

Linker/Loader الرابط

الملف التنفيذيّ
Executable File

المعالج

Language Processing System



Language Processing System

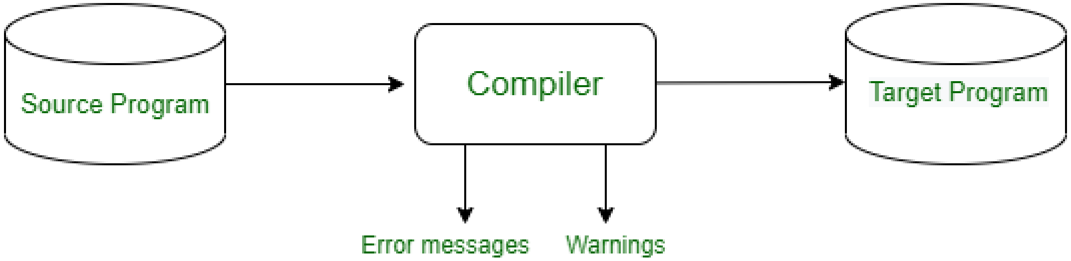
Language Processing System

The term "**Language Processing System**" is used to refer to the different stages through which a high-level script is converted into machine language, and this system consists of the following components and stages:

Source Code: It represents the program written in one of the high-level programming languages and required to be executed by the processor.

Pre-processing: This stage is part of the translation process itself, where a new representation of the high-level source code is generated that is used by the compiler, and the main goal of this stage is to identify pre-programming guidelines, definitions, and embedded offices.

Compiler: The compiler takes the modified code as input and produces the target code as output.



Language Processing System

Assembler: The assembler takes the target code as input and produces real locatable machine code as output.

Linker: Linker or link editor is a program that takes a collection of objects (created by assemblers and compilers) and combines them into an executable program.

Loader: The loader is one of the parts of the operating system and its function is to download the executable file resulting from the work of the linker, put it in memory, and then run it.

Executable code:It is low-level and machine-specific code that the machine can easily understand. Once the job of the linker and loader is done the object code is finally converted it into executable code.

Programming Languages

In computer programming ,programming language serves as means of communication between the person with a problem and the computer used to solve it.

Programming language is a set of symbols ,words, and rules used to instruct the computer.

Programming Languages

A hierarchy of programming languages based on increasing machine independence include the following:

1- Machine Language

2- Assembly Language

3- High-Level Language

4- Problem-Oriented Language

Programming Languages

1- Machine Language: is the actual language in which the computer carries out the instructions of program. otherwise ,it is the lowest form of computer language, each instruction in program is represented by numeric code, and numeric of addresses are used throughout the program to refer to memory location in the computer memory.

Programming Languages

2- Assembly Languages: is a symbolic version of a machine language, each operation code is given a symbolic code such **ADD,SUB,…**

3- High-Level Language(HLL): is a programming language where programming not require knowledge of the actual computing machine to write a program in the language. HLL offer a more enriched set of language features such as control structures, nested statements, block,…ect.

Programming Languages

4-**Problem-Oriented Language: it** provides for the expression of problems in a specific application. Examples of such language are SQL for database application.

Programming Language

Advantage of HLL over LLL

1- HLL are easier to learn then LLL.

2- A programmer is not required to know how to convert data from external form to internal within memory.

3- Most HLL offer a programmer a variety of control structures which are not available in LLL.

4- Programs written in HLL are usually more easily debugged than LLL equivalents.

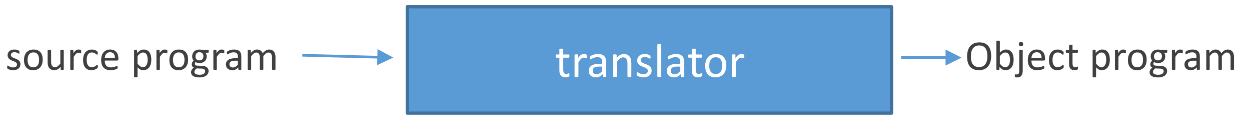
5- Most HLL offer more powerful data structure than LLL.

6- HLL are relatively machine-independent .

Translator

High-level language programs must be translated automatically to equivalent machine language program. A translator converts a source program into an object or target program.

The source program is written in a source language and the object program belong to an object language.



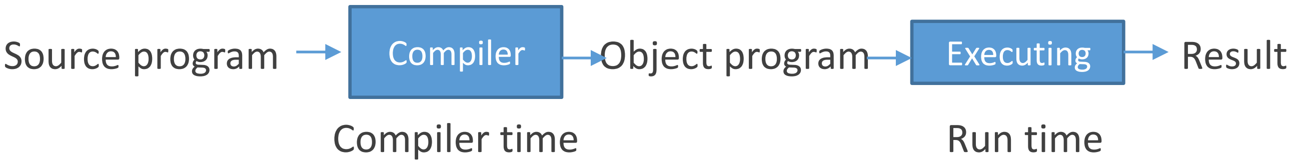
Translator

1- If the source program is written in assembly language and the target program in machine language the translator is called **Assembler.**

2- If the source program is written in HLL language and the object language is LLL then the translator is called **Compiler**.

3-If the source program is written in LLL language and the object language is HLL then the translator is called **Decompiler.**

Compilation Process



Source program → Compiler → Object program → Executing → Result

Compiler time

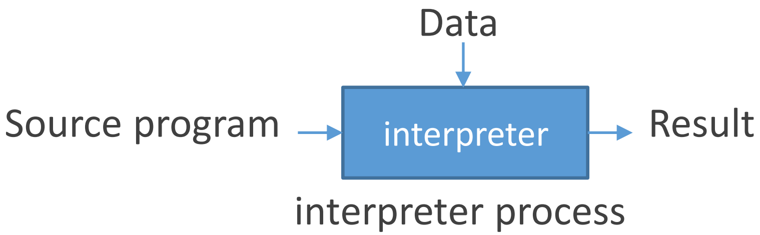
Run time

The time at which conversion of a source program to an object program occurs is called Compile time ,the object program is executed at Run time .

Note that the source program and data are process at different time.

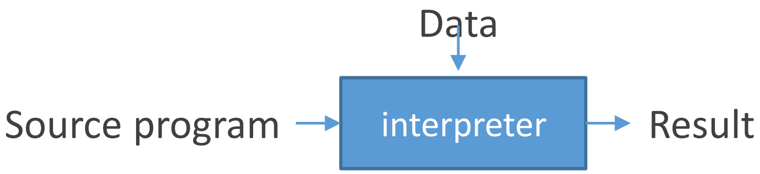
Interpreter

Another kind of translator called an **interpreter** in which processes an internal form of source program and data at the same time. That is interpretation of the internal source form occurs at run time and no object program is generated.



Interpreter

Another kind of translator called an interpreter in which processes an internal form of source program and data at the same time. That is interpretation of the internal source form occurs at run time and no object program is generated.



interpreter process

Comparison between compiler and interpreter

Compiler program usually run faster than interpreter ones because the overhead of understanding and translating has already been done.

However ,Interpreters are frequent easier to write than Compilers, and can more easily support interactive debugging of program.

Thanks for listening