

Advanced programming (Lecture1)

Dr. Seyed amir hossein tabatabaei

1403/11/30

Contents

- ▶ An introduction to computer programming
- ▶ An introduction to Python (data types, numerical data, etc,.)
- ▶ Control structures in python
- ▶ Strings and lists
- ▶ Functions
- ▶ Recursive functions and recursive programming
- ▶ Object oriented programming (class, inheritance, UML diagram, etc,.)
- ▶ Game design

Grading scheme

- ▶ Mid-term : 20%
 - ▶ Group 111:1404/02/08 (16:00-18:00)
 - ▶ Group 112:1404/02/09 (10:00-12:00)
- ▶ Final : 50%
- ▶ Project and exercises (TAs): 20% + 10%
 - ▶ Projects will be developed and located in Github (<https://github.com/>)
 - ▶ Exercises will be submitted through course portal (PLEASE DO NOT SEND THE EXERCISES AND/OR PROJECTS VIA EMAIL)

Group link

- ▶ Group link:

Software

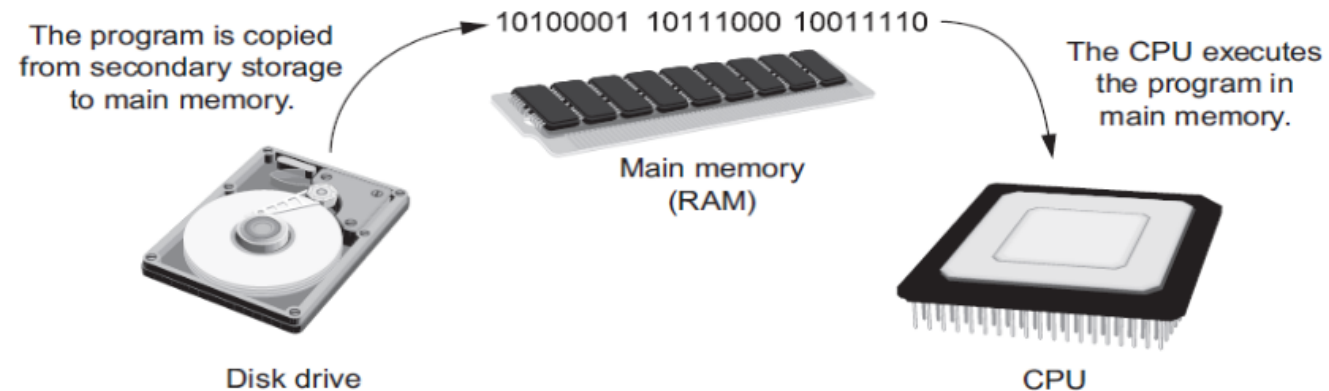
- ▶ **System software:** The programs that control and manage the basic operations of a computer are generally referred to as system software
 - ▶ Oss
 - ▶ Utility Programs
 - ▶ SDTs
- ▶ **Application software:** Programs that make a computer useful for everyday tasks are known as application software (Office, Adobe, etc.,).

Data storing in a computer

- ▶ Storing real numbers in a computer
 - Converting to a binary format
 - Floating point notation/other encoding system
- ▶ Other data types
 - Images (different data types, PNG, JPG,...)
 - Video (MP4)
 - Sound/acoustic (WAV)

Programming languages

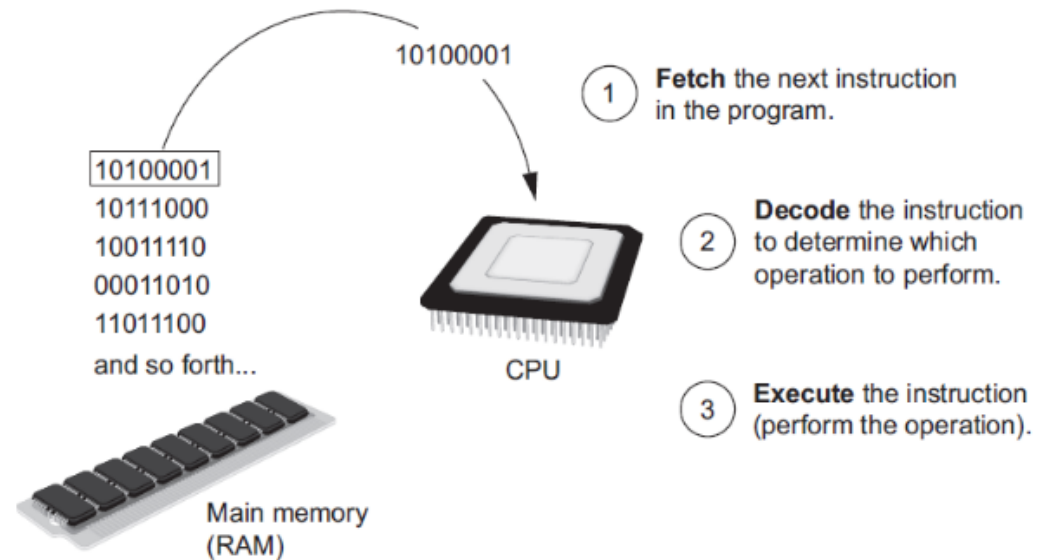
- ▶ CPU understands just machine language represented as binary strings
- ▶ CPU is able to perform just basic and simple operations
- ▶ Each CPU brand has its own instructions



Programming languages

► The main steps of a program execution:

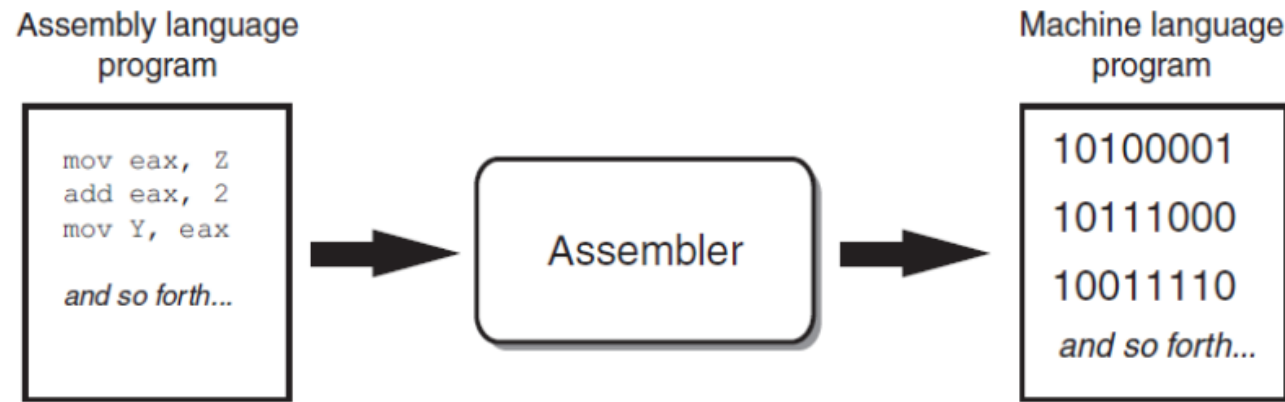
1. Reading the instruction in the code
2. Decode the instruction
3. Execute the instruction



Programming languages

- ▶ Assembly language: an interface between machine language and the user
- ▶ Low-level language: assembly is a low-level language as it requires many instructions for an operation, very close to machine language instructions
- ▶ Assembler: it translates the program written by assemble language to machine language to be executed by the CPU

Programming languages



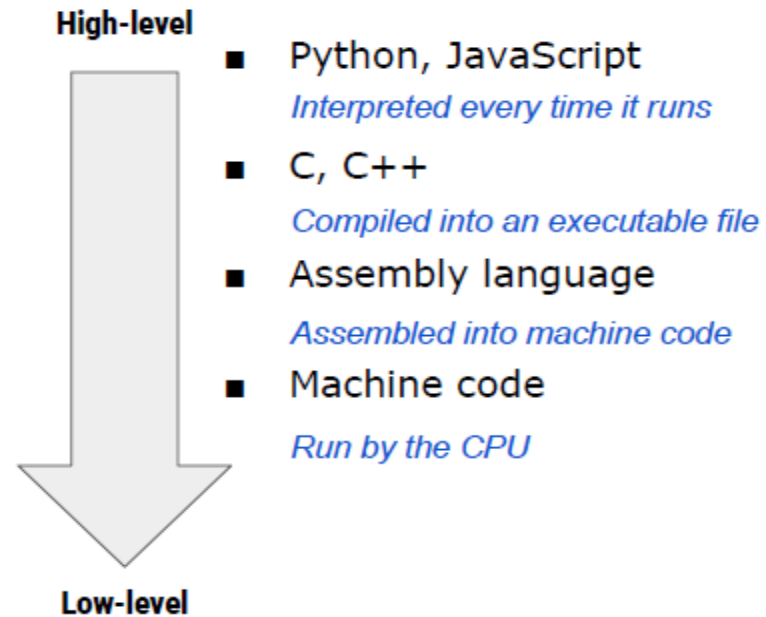
- Figure 6. from assembly to machine language

Programming languages

► High level languages

1. Developed in early 1950's
2. Was a response to the in-efficiency and difficulties of low-level languages
3. Close or very close to the human natural language
4. Consists of keywords, statements, operators....
5. Examples include C, C++, Java, Basic, Visual Basic...

Programming languages



(<https://www.mrdfinch.com/high-and-low-level-languages.html>)

Programming languages

```
ask_again = True

while ask_again == True:

    cont = input("Would you like me\
    to ask again?")

    if cont == "no":
        ask_again = False
```

High level language (Python)

- Easy to read and understand by humans

```
q00003,    AND fnc026    ;Address of q00003
fnc026,    CLE          ;Function q00003
          BSA pop       ;Getting return address
          STA temp1
          BSA pop       ;Processing arg q00028
          STA q00028
          LDA temp1
          BSA push      ;Putting return address back
          CLE          ;{WORD} q00027 = q00032
          LDA q00032
          BSA push
          BSA pop
          STA q00027

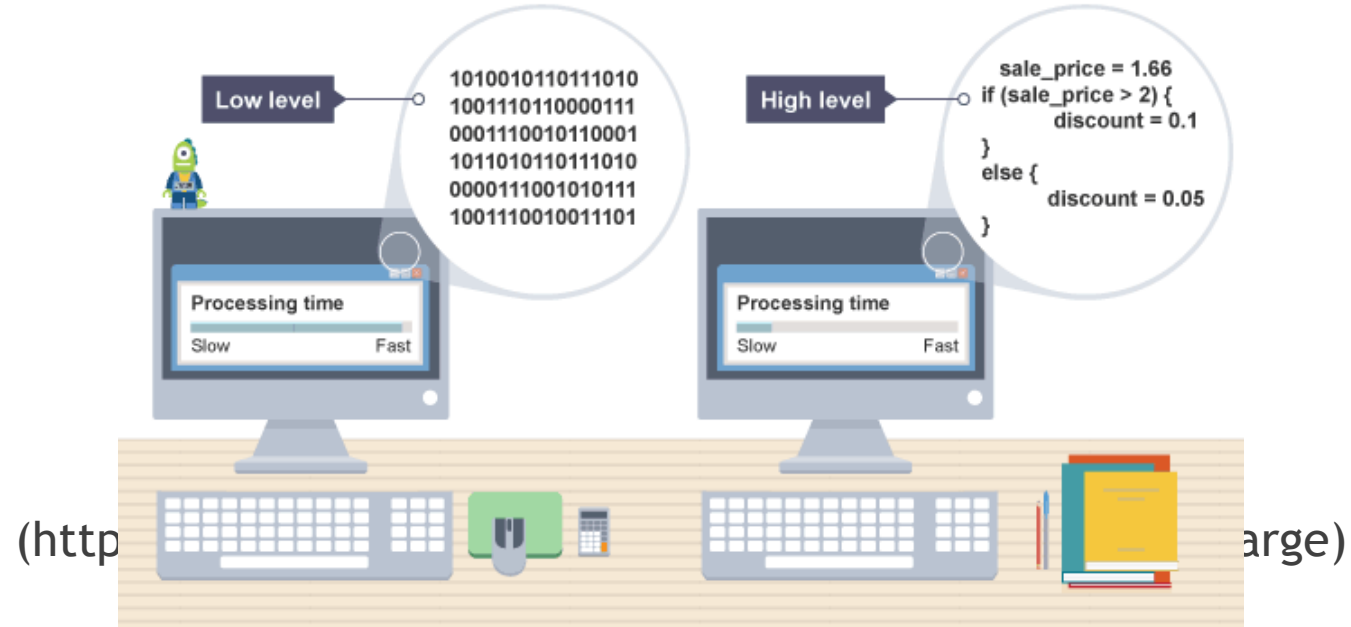
          CLE          ;{WORD} q00035 = q00030
          LDA q00030
          BSA push
          BSA pop
          STA q00035
```

Low level language (assembly code)

- Hard to read and understand by humans

(<https://www.mrdfinch.com/high-and-low-level-languages.html>)

Programming languages



Programming languages

- ▶ Compilers and interpreters
 - Compiler is a program that translates each code of a high-level language to a separate machine language code. The latter code can be executed at any time.
 - Interpreter is a program that both translates and execute the instructions in a high-level language.

Programming languages

- ▶ Debugging and error types
 - ▶ Bugs: programming errors, debugging is the process of taking them down and correction
 - ▶ Syntax error: incorrect writing a statement or keyword
 - ▶ Runtime error: an error occurred in executing the code
 - ▶ Semantic error: the error in generating the right output of the code (desired output)

Python

- ▶ Introduced by Guido Van Rossum in 1991
- ▶ Interpreter
- ▶ High-level
- ▶ General-purpose
- ▶ Can be executed in any platform (MacOS, Linux, Windows, iOS, etc,.)
- ▶ Versions from 2.7 till 3.12.xx

Python

- ▶ Applications:
 - ▶ Data Mining
 - ▶ Artificial Intelligence
 - ▶ Scientific computations
 - ▶ Web development
 - ▶ Apps development
 - ▶ Games