GNG1106

Fundamentals of Engineering Computation

Instructor: **Hitham Jleed**

University of Ottawa

Fall 2023 ~

Outline

1. Computer Basics

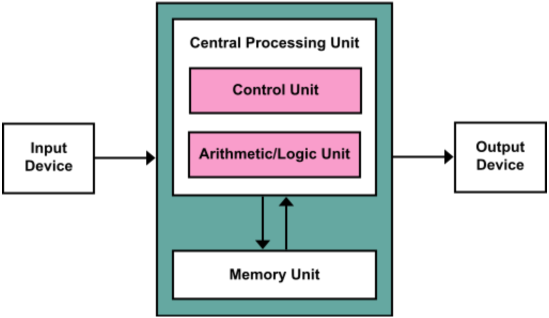
2. Programming Languages

3. First Program

What is a computer?

- A computer is a "general-purpose" device that can be programmed to carry out a set of arithmetic or logical operations automatically.
  - Desktop computers, laptops, smart phones, tablets, etc are all computers
- A computer operates on data according to directives or commands that are listed in a program.
- The program is written by a human (at least for now) using a programming language.
- For a computer to accomplish a task, both the computer (i.e., the hardware) and a program (software) are required.

Computer Architecture



Picture from Wikepedia, Von Neumann Architecture

`https://en.wikipedia.org/wiki/Von_Neumann_architecture`

Logical Organization of Computers

- Input Device: to obtain information from the outside world
    - keyboard, USB, hard drives and network cards, etc.
- Output Device: to make available to the outside the results of computation
    - screen, speakers, network card, USB, etc.
- Memory: to temporarily store data during program execution
    - Various kinds: RAM (random-access memory), flash memory, etc.
    - Low storage capacity but fast access
    - Stored data is lost on power off
- CPU ("Central Processing Unit"): to perform logic and arithmetic operations (done by Arithmetic Logic Unit, i.e., ALU), and as well as control and I/O operations
- Secondary Storage: for long-term storage of data and programs
    - Hard drive, USB, CD, etc.
    - High storage but slower access

An Analogy

- Teacher: *what is the volume of a sphere with radius 5.79 cm?*
- Student:
  1. writes $r = 5.79$cm on a piece of paper on his desk
  2. goes to his file cabinet to find the formula for sphere volume
  3. writes on the piece of paper: $V = (4/3)\pi r^3$
  4. goes to his file cabinet to find the value π
  5. writes on the piece of paper $\pi = 3.14$
  6. substitutes $r = 5.79cm$ and $\pi = 3.14$ into $V = (4/3)\pi r^3$
  7. calculates V on the piece of paper and gets 258.8 cm^3
  8. Says: *the volume is* 258.8 cm^3

- Teacher ∼ a computer user
- Student ∼ a computer

his ear	input device
his voice (mouth)	output device
his piece of paper	memory
his file cabinet	hard drive
his brain	CPU
the 8 steps he follows	program

Note

This analogy is not perfectly accurate, since the program actually also lives in the memory. A more accurate analogy would require that the 8-step instructions be written on the piece of paper and that the student follow the instructions to complete the 8 steps.

Computer Hardware

- memory
- CPU
- hard drive
- mouse
- keyboard
- sound card
- speaker
- network card
- motherboard
- ...

Note

- A hardware is a physical and tangible part of a computer.
- You can destroy or smash a hardware.

Computer Software

- Operating System (OS)
  - Windows
  - Linux
  - MacOs
  - Unix
  - iOS
  - Android
  - ...
- Application software ("Apps")
  - MS Office
  - Safari
  - Code Blocks
  - Acrobat Reader
  - ...

Note

- A software is a program or a suite of programs.
- You can not smash a software, but you can delete it.

Bits and Bytes

- Inside a computer data and programs are both represented as bits
- A bit is a binary digit (or binary information unit), taking values 0/1
- Eight consecutive bits form a byte.

In-Class Exercise:

- The size of memory (or of a secondary storage unit) is measured in terms of number of bytes it can hold.
- The units of memory size are bytes, kilobytes (KB), megabytes (MB), gigabytes (GB), terabytes (TB), ...
    - The convention is to write KB, MB etc all in capitals
    - 1KB has been taken as 1024 Bytes (and 1MB as 1024 KB and so on) by almost all manufacturers.
    - There are increasing debates of changing the definitions of these units.

Outline

1 Computer Basics

2 Programming Languages

3 First Program

Machine Language

- Machine language is the only language that the computer hardware can understand directly.
- All programs and data must be translated to machine language.
- Machine languages are machine-dependent.
- Machine language is generally difficult to read, to code and to manipulate

Example

Machine code for adding variables x and y and storing the result in z

```
FC  0900
F3  0902
7C  0904
```

Assembly Language

- Assembly language use English-like abbreviations or mnemonics to describing machine code.
- An assembler translates the program into machine code.
- Assembly language is still too cumbersome to manipulate for writing large and complex programs.

Example

Assembly code for adding two variables x and y and storing the result in z

```
LOAD   x
 ADD   y
STORE  z
```

High Level Languages

- A high level language consists a large number of powerful commands.
- Commands are descriptive, mathematical and/or English-like.
- A compiler translates the program written in a high level language to a program in machine code.

Example

C code for adding two variables x and y and storing the result in z

```
z = x + y;
```

Programming Lifecycles

The programming (or coding) with a high level language iterates over:

- Editing
- Preprocessing
- Compiling
- Linking
- Loading
- Execution

Editing

- Editing is done using a plain text editor.
- In the editing process, programmer writes or revises his code in an electronic form and save it in a computer file.
- In C programming, the file needs to have a name that ends with suffix "`.c`", e.g., `program1.c`
  - DO NOT include spaces in the name of the file, e.g., as "`my program.c`"
  - In some cases, your programming environment (say CodeBlocks) may give you trouble even when the name of the folder containing the file has spaces in it. For example, if the folder containing the file is called "`my programs`", it may cause you trouble (e.g., on the Mac version of CodeBlocks).

Preprocess + Compile +Link=“Build”

- Preprocessing executes the preprocessor directives in the program code before compilation. Common preprocessor directives include substitution and insertion of code.
- Compilation of the code is done by a compiler for that programming language. In the compilation process, the compiler checks if the code has the correct syntax, without checking its logic, and translates the code to an “object code”. Usually the object code filename has suffix “`.obj`” or “`.o`”. e.g., `program1.obj` or `program1.o`
- Linking is the process that link the object code with the necessary libraries (math, I/O, etc). The linker creates an executable file; the executable filename generally has suffix “`.exe`”, as in `program1.exe`, or without suffix as in `program1`

Load + Execute = “Run”

- Loading is the process that loads the executable code into the computer memory.
- Execution is the process in which the computer performs operations following the instructions in the code

IDE

- For the convenience of developing programs, software has been developed that integrates all required tools for these steps into a common environment. Such an environment is called an IDE (Integrated Development Environment).
- CodeBlocks is an IDE.

Popular High Level Languages

- Python
- Java
- BASIC
- Pascal
- Ada
- Visual Basic
- Delphi
- FORTRAN
- C++
- C#
- C (We use it in this course)

Pseudo-Code

- Pseudo-code instructions consist of English language text to describe the desired action to be taken by the computer.

  ```
  Assign 2 to var1
  ```

- A single line of pseudo-code corresponds to a single or multiple programming language instructions.
- Pseudo-code is not program; it only shows the logic and design of the program

Outline

1 Computer Basics

2 Programming Languages

3 First Program

First Program

```
#include <stdio.h>
/* This is our first program */
int main()
{
            printf("Hello World!\n");
            return 0;
}
```

Coding Demonstration