Ch. 1: Introduction to Computers & Programming

Tuesday, December 18, 2018 1:00 AM

Overview:
This chapter begins by giving a very concrete and easy-to-understand explanation of how computers work, how data is stored and manipulated, and why we write programs in high-level languages. An introduction to Python, interactive mode, script mode, and the IDLE environment are also given.

1.1: Introduction

- Computers can perform a wide variety of tasks because they can be programmed. This means that computers are not designed to do just one job, but to do any job that their programs tell them to do.
- **Program: A set of instructions that a computer follows to perform a task.**
  - Commonly referred to as *Software.*
- **Programmer: A person who can design, create, and test computer programs**
  - Also known as a software developer.

1.2: Hardware & Software

- **Hardware: The physical devices that make up a computer.**
  - Computer is a system composed of several components that all work together.
- **Typical major computer components:**
  - Central Processing Unit (CPU)
  - Main memory (RAM)
  - Secondary storage devices
  - Input and output devices

The CPU:

- **Central Processing Unit (CPU): The part of the computer that actually runs programs.**
  - Most important component of a computer.
  - Without it, the computer cannot run software.
  - When a computer is performing the tasks that a program tells it to do, we say that the computer is *running* or *executing* the program.
- CPUs used to be huge devices consisting of vacuum tubes and switches!
- The ENCIAC, which is considered to be the world's first programmable electronic computer, was built in 1945 to calculate artillery ballistic tables for the U.S. Army. This machine, which was primarily one CPU, was 8 feet tall, 100 feet long, and weighed 30 tons.
- **Microprocessors: CPUs located on small chips.**
- Today, CPUs are small chips known as microprocessors. In addition to being many times smaller than the old electromechanical CPUs in early computers, microprocessors are also much more powerful.

Main Memory:

- **Main memory: Where the computer stores a program and the data used by the program while the program is running.**
  - For example, suppose you are using a word processor (like Microsoft Word) to write an essay. While you are writing this essay, both the word processing program and the essay are stored in the *main memory.*
- **Also known as *Random Access Memory,* or *RAM.***
  - We call it RAM because the CPU is able to quickly access data stored at any random location in RAM.
  - RAM is a volatile memory used for temporary storage while the program is running.

    - The contents of a computer's RAM are erased when the computer is turned off.
    - Inside your machine, RAM is stored in chips.

Secondary Storage Devices:

- **Secondary Storage: A device that can hold data for long periods of time.**
    - Programs are typically stored here and loaded to the main memory when needed.
- Types of secondary memory:
    - **Disc Drive: Magnetically encodes data onto a spinning circular disc.**
    - **Solid state drive (SSD): Faster than a disk drive, no moving parts, stores data in solid state memory.**
    - **Flash memory (flash drive, memory stick): Portable, no physical disc. Usually used by connecting it to a computer's universal serial bus (USB) communication port and appears to the computer as a disc drive, but does not contain a disc.**
    - **Optical devices (compact disc (CDs), digital versatile disc (DVDs)): Data is encoded optically as a series of pits on the disc surface. CD and DVD drives use a laser to detect the pits and thus read the encoded data. Optical discs hold large amounts of data, and for that reason, recordable CD and DVD drives are commonly used for creating backup copies of data.**

Input Devices:

- **Input: Data that the computer collects from people and other devices.**
- **Input device: Component that collects the data.**
    - Ex: keyboard, mouse, touchscreen, scanner, camera, etc.
    - Disc drives and optical drives can be considered input devices because programs and data are retrieved from them and loaded into the computer's main memory.

Output Devices:

- **Output: Data produced by the computer for other people or devices.**
    - Can be text, image, audio, or bit stream.
- **Output device: Formats and presents output.**
    - Ex: video display (monitor, screen), printer, etc.,
    - Disc drives and USB drives can be considered output devices because data is sent to them to be saved.

Software:

- Everything the computer does is controlled by software. From the time the computer is switched on until you shut the system down, the computer is entirely controlled by software.
    - General categories of software: System software and application software.
- **Application software: Programs that make the computer useful for everyday tasks.**
    - Ex: word processing, email, games, web browsers
- **System software: Programs that control and manage basic operations of a computer.**
    - **Operating system: The most fundamental set of programs on a computer.** The operating system controls the internal operations of the computer's hardware, manages all of the devices connected to a computer, allows data to be saved to and retrieved from storage devices, and allows other programs to run on the computer. Popular OS for computers include Windows, macOS, and Linux. Popular OS for mobile devices are Android and iOS.
    - **Utility program: Performs specific task to enhance computer operation or safeguard data.** These programs include virus scanners, file compression programs, and data backup programs.
    - **Software development tools: Used to create, modify, and test software programs.** Assemblers, compilers, and interpreters are examples of software development tools.

1.3: How Computers Store Data

- All data in a computer is stored in sequences of 0s and 1s.

- **Byte: Just enough memory to store a letter or a small number.**
    - Divided into into 8 *bits.*
    - ***So 1 byte = 8 bits.***
- In order to perform meaningful operations, a computer must have many bytes, usually millions (or even billions) of bytes.
- **Bit (binary digit): Electrical component that can hold positive or negative charge, like an on/off switch.**
    - The on/off pattern of bits in a byte represents data stored in the byte.
    - The positive charge corresponds to on, and the negative charge corresponds to off.

Storing Numbers:

- Bits represent two values: 0 and 1.
- Computers use the binary numbering system.
    - Position of digit j is assigned to the value of 2^j-1.
    - To determine the value of a binary number, sum the position value of the 1s.
- Byte size limits are 0 and 255.
    - 0 = all bits off; 255 = 255 all bits on.
    - To store larger number, use several bytes.

Storing Characters:

- Data stored in a computer must be stored as a binary number.
- Characters are converted into numeric code stored in memory.
    - The most important coding scheme is ASCII (American Standard Code for Information Interchange).
        - ASCII is limited; it only defines codes for 128 characters.
    - Unicode coding scheme is becoming the standard.
        - Compatible with ASCII.
        - Can represent characters for other languages.

Advanced Number Storage:

- The binary numbering system can only be used to represent integer only numbers, beginning with 0.
- To store negative numbers and real numbers, computers use binary numbering and encoding schemes.
    - Negative numbers are encoded using two's complement.
    - Real numbers are encoded using floating-point notation.
        - In this class, you don't need to know how these work, just that they are used to convert negative numbers and real numbers to binary format.

Other Types of Data:

- **Digital: Describes any device that stores data as binary numbers.**
- Digital images are composed of pixels.
    - To store images, each pixel is converted to a binary number representing the pixel's color.
- Digital music is composed of sections called samples.
    - To store music, each sample is converted to a binary number.

1.4: How a Program Works

- A computer's CPU can only understand instructions that are written in machine language. Because it is very difficult and inefficient to write entire programs in machine language, other programming languages have been invented.
- The CPU is designed to perform simple operations of pieces of data.
    - Ex: reading data, adding, subtracting, multiplying, and dividing numbers, other arithmetic and relational operator operations, etc.,

    - The CPU understands instructions written in machine language and included in its instruction set.
        - Each brand of CPU has its own instruction set.
    - To carry out meaningful calculation, the CPU must perform many operations.
- The CPU does nothing on its own. It has to be told what to do, and that's the purpose of a program. A program is nothing more than a list of instructions that cause the CPU to perform operations.
- Although a program can be stored in a secondary storage device, it has to be copied into the main memory, or RAM, each time the CPU executes it.
- The CPU executed program in a cycle of 3 steps:
    - **Fetch: Read the next instruction from memory into CPU.**
    - **Decode: CPU decodes the fetched instruction to determine which operation to perform.**
    - **Execute Perform the operation.**

From Machine Language to Assembly Language:

- It is impractical for people to write in machine language.
- **Assembly language: Uses short words (mnemonics) for instructions instead of binary numbers.**
    - Easier for programmers to work with.
- **Assembler: Translates assembly language into machine language for execution by CPU.**

Language Levels:

- **Low-level language: Close in nature to machine language.**
    - Ex: assembly language
- **High-level language: Close in nature to human language; allows simple creation of powerful and complex programs.**
    - No need to know how CPU works of write large number of instructions.
    - More intuitive to understand.

Key Words, Operators, & Syntax:

- **Key words: Predefined words used to write program in high-level language.**
    - Each keyword has a specific meaning.
- **Operators: Performs operations on data.**
    - Ex: Math operators are used to perform arithmetic.
- **Syntax: A set of rules to be followed when writing a program. (Similar to grammar.)**
- **Statement: Individual instruction used in high-level language.**

Compilers & Interpreters:

- Programs written in high-level languages must be translated into machine language to be executed.
- **Compiler: Translates high-level language program into separate machine language program.**
    - Machine language program can be executed at any time.
- **Interpreter: Translates and executes instructions in high-level language program.**
    - Used by the Python language.
    - Interprets one instruction at a time.
    - No separate machine language program.
- **Source code: Statements written by the programmer.**
- **Syntax error: Prevents code from being translated.**

1.5: Using Python

- Python must be installed and configured prior to use.
    - One of the items installed is the Python interpreter.
- Python interpreter can be used in 2 modes:
    - Interactive mode: enter statements on the keyboard.

        - *When the Python interpreter is running in interactive mode, it is commonly called the "Python Shell"*
    - Script mode: save statements in Python script.

Interactive Mode:

- When you start Python in interactive mode, you will see a prompt.
    - This indicates that the interpreter is waiting for a Python statement to be typed.
    - The prompt reappears after the previous statement is executed.
    - An error message is displayed if you incorrectly type a statement.
- Interactive mode is a great way to learn about Python and utilize your new skills!

Writing Python Programs & Running Them in Script Mode:

- Statements entered in interactive mode are not saved as a program.
- To create a program, use script mode.
    - Save a set of Python statements in a file.
    - The filename should have the .py extension as its suffix.
    - To run the file, or script, type

      Python filename

      At the operating system command line.

The IDLE Programming Environment:

- **IDLE (Integrated Development Program): A single program environment that provides tools to write, execute, and test a program.**
    - Automatically installed when the Python language is installed.
    - Runs in interactive mode.
    - Has built-in text editor with features designed to help write Python programs.

We Covered:

- Main hardware components of a computer
- Types of software
- How data is stored in a computer
- Basic CPU operations and machine language
- Fetch-decode-execute cycle
- Complex languages and their translation to machine code
- Installing Python and the Python interpreter modes