

Introduction To Computers and Programming

Outline

- Computer and information
- History of computer technology

What is a computer?

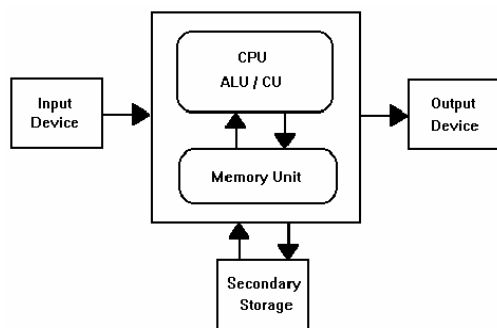
- A *computer* is an information-processing machine that performs simple tasks according to specific instructions. This means it can store, retrieve, output and process *data*.
- John Von Neumann proposed this stored computer concept in 1946.
- *Data* is a collection of unorganized facts, which includes words, numbers, images, and sound.

Computer system

- Hardware -- physical components of computer that you see or touch
- Software -- computer programs that instruct hardware to perform specific tasks
- A *computer program* is a set of instructions written in a *programming language*.

Basic Computer Architecture

- central processing unit (CPU)
- memory unit
- input devices
- output devices



Central Processing Unit - CPU

- *arithmetic logic unit (ALU)*: performs arithmetic and logic calculations.
- *control unit (CU)*: coordinates the actions of the other components so that instructions (the program) are executed in sequence.

Memory

- consists of memory locations (or words)
- each memory location contains a string zeros or ones.

Input/Output Device

- **Input device** An input device is a device that allows a user to communicate information to the computer.
- **Output device** An output device is a device that the computer communicates with the user.

Secondary/Auxiliary Storage

- magnetic tape drives
- disk drives
- CD ROM drives

Terminology

- **Peripheral Device:**
 - Input device
 - Output device
 - Secondary storage device
- **Hardware**
 - The physical components of a computer
- **Software**
 - Computer programs

Categories of Computers

- Personal computers (PCs)
- Minicomputers
- Mainframe computers
- Supercomputers

Personal Computers

- Desktop computers
- Network of computers and Web appliances -- WebTV
- Laptop or notebook computers
- Handheld computers -- small personal computers, PDAs, cellular phones

Minicomputers

A *minicomputer* is designed for a small group of organizations with a more powerful computing capabilities. The computing process of a minicomputer can be accessed by several users via terminal that connected to it.

Mainframe Computers

A *mainframe computer* is a large, expensive, and powerful computing process that allows hundred and thousand users access its computing capabilities.

Supercomputers

A *supercomputer* is the fastest, most powerful, and most expensive. It is designed specifically for applications requiring complex, sophisticated mathematical calculations -- weather forecasting, medical image processing, petroleum exploration,...

Servers

A *server* is a computer, commonly a desktop or a more powerful desktop-like computer, connected to a computer network. It provides resources such as programs and information to be accessed by the desktop computers called clients in the network.

History of Computer Technology

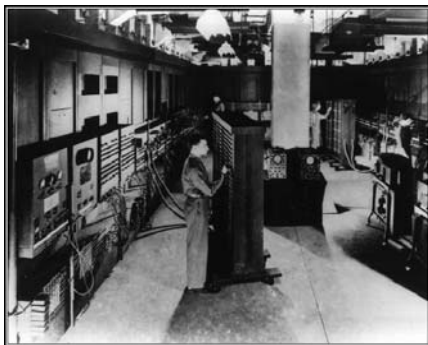
- The first computer
- The first generation (1950s)
- The second generation (1960s)
- The third generation (1965 to 1975)
- The fourth generation (1975 to 1991)
- The fifth generation (1991)

The First Computer

The first machine ABC (Atanasoff Berry Computer) to employ electronics (vacuum tubes) was developed in 1939 by a physicist John V. Atanasoff and Clifford Berry at Iowa State University. This was used to solve simultaneous linear equations.

ENIAC

In 1946, J. Presper Eckert and John Mauchly at Moore School of Engineering, University of Pennsylvania developed the first large-scale computer called ENIAC (Electronic Numerical Integrator and Computer). It used 17,480 vacuum tubes. This Machine uses the program to control calculations.



Stored-Program Computer

In 1946, John Von Neumann proposed the concept of stored program computer.

- encode both program and data as binary number,
- store the program along with the data electronically in a set of switches (computer memory),
- provide a central processing unit that not only perform calculations but also fetch, decode and execute the instructions contained in the program.

The First Generation

In 1951, J. Presper Eckert and John Mauchly built the first general-purpose commercial computer, the UNIVAC. This is the first generation of commercial computers. The instructions were written machine language. UNIVAC used less number of vacuum tubes than ENIAC.



The Second Generation

- Vacuum tubes were replaced by transistors.
- Computers are faster and smaller.
- It has punched cards for input, tape storage, and disk storage.
- Development of high-level programming languages: COBOL (Common Business-Oriented Language) and FORTRAN (Formula Translator).

The Second Generation

- ASCII (American Standard Code for Information Interchange) was developed in 1963.
- In 1964 IBM announced a new line of computers called System/360.

The Third Generation

- Timesharing
- Batch processing
- Development of integrated circuits (ICs)
- Small-scale integration and medium-scale integration
- Wide area networks
- Internet

The Fourth Generation

- Very-large-scale integration
- Microprocessor
- MS-DOS
- Command line interface
- Graphical interface

The Fifth Generation

- Artificial intelligence (AI)
- World Wide Web
- Local area networks
- Wireless technology
- E-commerce

What is Programming?

A *programming* is a process of planning a sequence of instructions for a task or an event.

Computer Programming

- A *computer programming* is a process of planning a sequence of instructions for a task or an event to be performed by a computer.
- A *computer program* is a sequence of instructions for a task to be performed by a computer.

How do we write a Program?

There are two phases to write a program:

- Problem solving
- Implementation

Problem -Solving Phase

- *Analysis and Specification:* Determine precise objective of the solution to the problem
- *Design a solution (Algorithm):* Develop a logical sequence of steps to solve the problem.
- *Verification:* Check whether the solution does solve the problem

Implementation Phase

- *Coding (Program):* Translate the design or algorithm into a programming language
- *Testing:* Have the computer follow the instruction in the program (*Run the program*) and check the results

Maintenance Phase

- *Utilization:* Use the program
- *Maintain:* Revise or modify the program according to changing requirements

What is an algorithm?

- An *algorithm* is a set of steps for solving a problem in a finite amount of time.

Example 1

The Euclidean for finding the greatest common divisor of two positive integers:

- Assign the larger number to M and smaller to N
- Divide M by N, and call the remainder R
- If R is not 0, then assign M the value of N, assign N the value of R, and return to step 2; otherwise, the greatest common divisor is the value currently assigned to N

Example 2

A new student wants to attend University of Houston-Downtown

- Apply for admission or readmission and financial aid.
- Attend New Student Orientation.
- Take any required placement tests.
- Plan your schedule and get course approvals.
- Register for classes.
- Pay tuition and fees.
- Get UHD ID, parking permit, and locker

What is a Programming Language?

A Programming language is a set of rules, symbols, and special words used to construct a program.

Classification:

- High level Language
- Assembly Language
- Machine Language

High-Level Language

A High-level language is a language that uses code resembling the English language. For examples, C++, Java, FORTRAN, COBOL, and Ada.

Assembly Language

An Assembly language is a low-level language that uses a mnemonic to represent each of the machine language instructions for a particular computer.

Machine Language

A Machine language is the lowest level of programming languages that contains binary-coded instructions which directly instructs a computer to perform the task.

Example of Assembly and Machine Language Instructions

- ADD 100101
- SUB 010011

Compiler

- **An assembler** is a program that translates an assembly language into machine code.
- **A compiler** is a program that translates a high-level language into machine code.
- **A source program** is a program written in a high-level programming language.
- **An object program** is a source program in machine code.

Computer Software Organization

Types of Software

- System software - programs that manage computer hardware and software
 - operating systems
 - System utilities - anti-virus software, uninstall software
- Application software - programs that provide users to perform specific tasks such as word processing, video editing software, Web browsers

Operating System

An operating system is a primarily resource manager that manages computer hardware in the form of processor, storage, I/O devices, communication devices, and data.

Example

It controls:

- file system
- execution of user commands
- access to the system
- scheduling user's time
- sharing data

Types of Operating Systems

- Command-line interface - DOS
- Graphical User interface (GUI) -- Windows

Application Software

- Off-the-shelf software or Canned software
 - Microsoft Office
 - Netscape
 - Internet explorer
- Custom software

Programming Environment

An *editor* is an application program that allows the user to create a program file (edit and store a program in a file) called source program or create a data file.

A *compiler* is a program that translates a source program into a sequence of instructions and data (in machine code called object program) that can be executed by a computer.

Representation and Storage of Information

In a computer data is represented by 1s and 0s (binary). A *bit* (binary digit) can be either 1 or 0. A *byte* is a group of 8 bits.

Classification of Information

- visual information (e.g. pictures, images)
- numeric information (e.g. numbers, fractions, etc.)
- character or symbol information (e.g. a, b,...z, A,..., Z, 1,..., 9, 0, &, #, @, <, /, ?, etc.)
- audio information (e.g. music, speech, bird songs, etc.)
- command or instructional information (e.g. recipes, computer programs, etc.)

Information Representation

Information can be represented by binary numbers that can be associated with parts of an electronic machine and their state or condition at a given moment

Binary Numbers

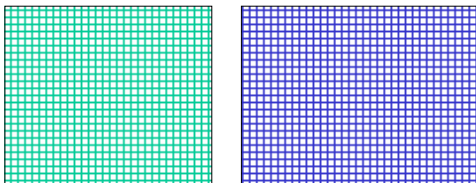
- similar to the familiar decimal numbers
- binary numbers use only the two symbols 1 and 0. Also, the position values are different from those of the decimal system

Representing Symbol and text

- ASCII (American Standard Code for Information Interchange) pp. A17
- EBCDIC (Binary Coded Decimal Interchange Code) pp. A18

Representing Picture

- pixel (PICTure ELement)



Representing Sound

- numbers

Representing Instructions

ADD Instruction

011 1101100
code for addition address of the number
 to be added