Lecture Note//Dr. Ajoy Kumar Biswas, Associate Professor of Statistics
Bidhannagar College

Part 1: Introduction to Computer Systems

A computer system consists of hardware and software. The physical machines constituting a computer are referred to as *hardware* of the computer. A sequence of instructions for a computer to follow is a *program*, and the collection of programs used by a computer is referred to as the *software* of the computer. Hardware along is incapable of performing any task. Personal computer, workstation, and mainframe are the main types of today’s computers. The basic architectures and program-execution procedures for all the three types of computers are more or less same.

Hardware

To understand how the data and programs are stored and how the programs are executed to process the data within a computer, we consider a simple model of computer. The model consists of five hardware units as shown in Figure 1. The five units are

1. Input unit (Keyboard and Mouse are the main constituents)
2. Main memory unit
3. Secondary memory unit (Hard disk, Pen drive and CD are the main constituents)
4. Central processing unit
5. Output unit (Monitor and Printer are the main constituents).

The Central processing unit (CPU) contains all the computational power of a computer, while the main memory stores the program and the data for the program. The main memory makes the data and the program ready for the CPU. Following the program instructions, CPU does logical and arithmetic operations and stores the result in the main memory. Input unit, output unit

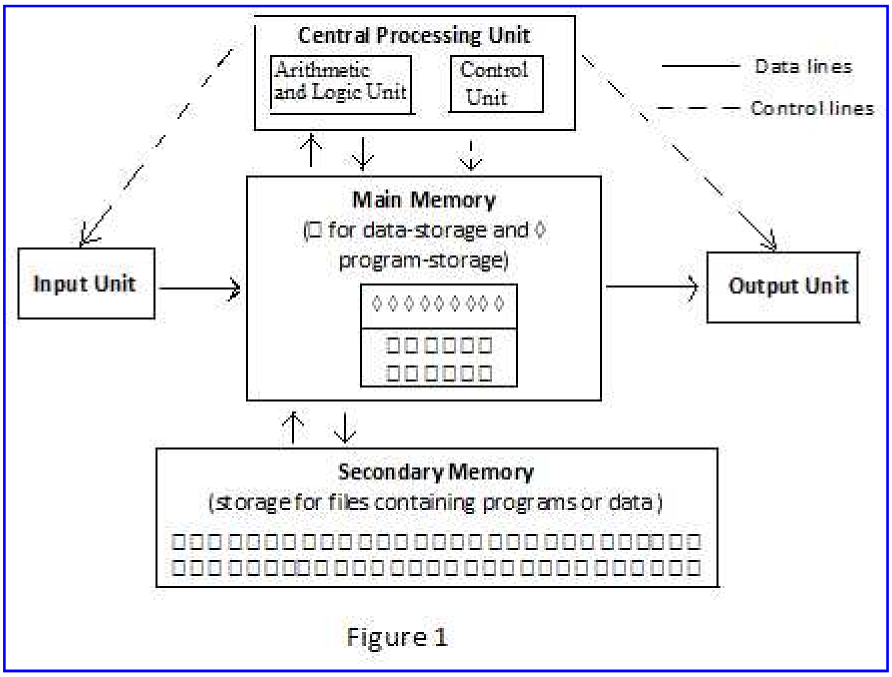

Central Processing Unit
Arithmetic and Logic Unit
Control Unit
Data lines
Control lines
Main Memory
(□ for data-storage and ◊ program-storage)
Input Unit
Output Unit
Secondary Memory
(storage for files containing programs or data)

Figure 1

and secondary memory unit operate under the control of CPU. In Figure 1, solid lines indicate the transfer of program and data and dash lines represent transfer of control signals.

A computer system stores alphanumeric inputs, such as a program and the data for the program written in terms of alphabet and decimal digits, as binary-digit forms of 0-1 strings; 0-1 strings are strings of the digits 0 and 1 of the binary number system. The computer internally encodes the alphanumeric inputs to 0-1 strings, performs instructions on the 0-1 strings that results outputs in the form of 0-1 strings, and finally decodes the 0-1 string outputs to understandable alphanumeric outputs. The computer system encodes alphanumeric inputs and decodes 0-1 string outputs according to certain coding schemes. Fortunately, a programmer seldom needs to be concerned with the codes.

Main and Secondary Memories

A computer has two forms of memory: main and secondary. The program and the data for the program entered into a computer, using the input unit, first go to the main memory. The program that is being executed is kept in the main memory. Main memory is also used as the storage of the intermediate and final results computed by the CPU. We may imagine the main memory to be partitioned into two parts; one part stores the program and the other part consists of many memory locations– one memory location per data item. The memory locations have distinct addresses.

A data item in a memory location is retrieved by referring to the address of the location. When a data item is read from a memory location a copy of the data item is used, the original data item is not destroyed. When a data item is written in main memory, the data item is stored in the specified location and the old content of the location is destroyed. Hence, we may think a memory location as a tiny blackboard on which the computer can write and erase.

An elementary memory location of a compute can be looked upon as a mechanical switch that can be set to be either OFF-state or ON-state. The OFF and ON-states can be map by the digits 0 and 1 of the binary number system, respectively. Consequently, each elementary memory location holds 1 bit of information, bit being the abbreviation of binary digit. Technically, a source of two pieces of information amounts 1 bit of information and the two pieces can be encoded by the digits 0 and 1 in a one-to-one way. A 1-bit memory location is usually referred to as a bit.

Usually, a computer system groups a number of adjacent bits and deals with the group rather than individual bits. The group of bits is called a *word* and the number of bits in a word is called *word length.* Word length may vary from one computer system to another. Earlier computer systems are 8-bit but currently there are 16-bit, 32-bit and 64-bit systems; 8, 16, 32 and 64 are the word-length of the systems. For a 16-bit system, for example, the CPU deals with a group of 16 bits in one-go. For giving the CPU access to a memory location of 16 bits in one-go, there are 16 wires running parallel to each other between the CPU and the main memory. And, consequently, the speed of a 16-bit system is higher than that of an 8-bit system. It may be noted that a word of length 8 bit is called a *byte*.

The main memory as a whole is organised as a large array of words. Every word has its own *address*. The address of a word is used to find the data item stored in it when required. To store a data item that is too large to fit within a word, computer uses several adjacent words to hold the data item. In this case the entire chunk of memory that holds the data item is a memory location whose address is the address of the first of the words. Thus, as a practical matter, we can think of the main memory of a computer as a long list of memory locations of varying sizes measured in terms of word. The sizes of the memory locations are not fixed, but can change when a new program is run on the computer. To note that most programming languages including C protect the programmer from the detail knowledge of words and their addresses. Instead, programming languages provide the abstractions of *variables* and their *types* for the programmer to manipulate. In the C programming language, for example, a variable (e.g., number of observations) is implemented as a memory location and the type of the variable (i.e., integer) determines the number of words required to hold a value of the variable.

The main memory discussed above is used only when the computer is actually following instructions of a program. The computer also has another form of memory called secondary memory or secondary storage. Secondary memory is used to store data and programs for a long period of time. Data and programs are stored in the secondary memory in unit called *file*. Files can be large and small. A program, for example, stored in a file in the secondary memory is copied into the main memory when the program is run.

Main memory is often referred to as random access memory (RAM) because the CPU can directly access a program and the program's data from this memory. Secondary memory often requires sequential access, which means that it requires going through all or very many files (memory locations) until the needed file is found. The main memory is volatile in nature in the sense that the content of the main memory are lost as soon as the computer is turned off. Unlike the main memory, secondary memory is non-volatile. The secondary memory is much-larger and inexpensive as compared to the main memory. For example, a personal computer's main memory is of size 256 MB here as secondary memory is of size 80 GB.

The various units for measuring computer memory are as follows: (=8) (=1024)

(i) Bit (in short b): the smallest unit	
(ii) Byte (in short B):	1 B = 2^3 b = 8 b
(iii) Kilobyte (in short KB):	1 KB = 2^{10} KB = 1024 KB
(iv) Megabyte (in short MB):	1 MB = 1024 KM
(v) Gigabyte (in Short GB):	1 GB = 1024 MB
(vi) Terabyte (in short TB):	1 TB = 1024 GB.

Central Processing Unit (CPU)

The CPU acts as the brain of the computer. It follows the instructions in a program and performs the computations specified by the program. It can retrieve or set the value of any location in the main memory. It consists of two parts: arithmetic-logic unit (A-L unit) and control unit. The arithmetic section of A-L unit performs arithmetic operations like addition, subtraction, multiplication and division. The logic section of A-L unit conducts logical operations like comparison of data for logical decisions. Complex arithmetic-operations are performed by the repetitive use of addition, subtraction, multiplication and division. The A-L unit contains a set of high-speed *registers* that temporarily hold data during arithmetic and logical operations.

The control unit control the operations of all parts of a computer; it does not carry out any actual data processing operation. Under the control of this unit, the data come from input unit to the main memory, are processed in A-L unit, and the result is stored back in the main memory. And processed results are converted to a form that can be understood easily by human being and is displayed with the help of an output device like monitor. The control unit works as follows:

(i) It retrieves instructions from the main memory.

(ii) It then retrieves the data required to be processed from the main memory.

(iii) It causes the A-L unit to actually carry out the required operations and decides whether those operations are carried out or not.

(iv) It places the processed results in the output area of the main memory.

(v) It fetches the next instruction from the main memory and repeats the whole cycle of operations outline above.

In addition to the above, the control unit also oversees admissibility of data and operations. When numerical data consisting of alphabet or dividing a number by zero, for example, appears, the control unit displays an error massage on the monitor screen to warn the programmer.

Software

The software of a computer helps in using the same hardware for many applications Software can be categorised as (i) system software and (ii) application software. A system-software is used for many applications; operating system, compiler, editor, linker are examples of system software. An application-software is used for a specific application; online admission software and railway reservation software are examples of application software.

Operating System

An operating system is system software, without which a computer (hardware and software collectively) cannot be used. We communicate with a computer through its operating system. The operating system manages all the hardware devices of the computer and allocates their respective task. It also helps in executing other system- and application-software. Windows, Linux, and MS-DOS are some commonly used operating systems.

We may think the operating system as the chief servant. It is in charge of all the other servants (programs and hardware devices). It delivers our request to the appropriate servant(s). If we want to edit a file, we tell the operating system the name of the file and its starts up the editor to work on that file.

Programming languages

Some of the commonly used languages for writing computer programs are C, Java, Basic, and Pascal. The instruction set of a program written by any of these languages is much compatible with human languages and human thought processes. So these are called *high-level languages*. High-level languages are easy for human being to write programs.

Languages that use only the binary digits 0 and 1 as the basic elements for writing programs are called *machine languages*. Computer can directly follow programs written in machine language. A program written in a high-level language must to be translated into machine language before it can be executed. A single instruction in a high-level language is commonly equivalent to several instructions in machine language, which greatly simplifies the task of writing complete, correct programmes.

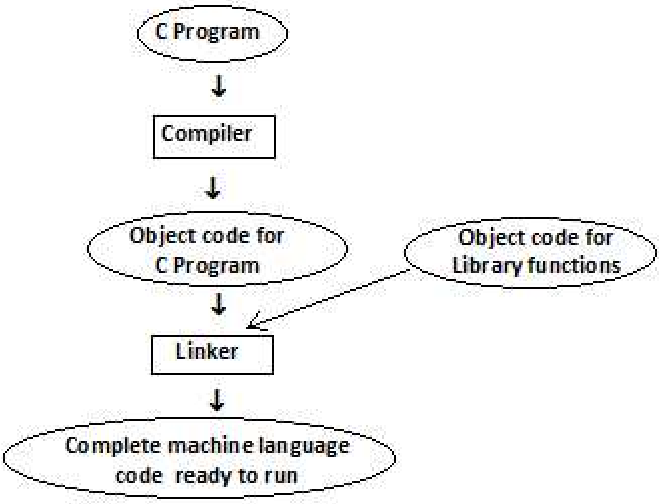
Complier

A *complier* is a program that translates a high-level language program, such as a C program, into a machine-language program that the computer can directly understand and execute. A compiler accepts a high-level language program as input, and generates a corresponding machine-language program as output. The original high-level language program is called the *source program* and the resulting machine-language program is called the *object program*. Commonly, a part or full of an object program is referred to as *object code*.

Linker

Any C program we write will contain some library functions (such as input and output functions) that have already been programmed and compiled for us. The object code of the library functions are to be combined with our program's object code to produce a complete machine-language program that can be run on the computer. The process of combining the object code is called linking and is done by a program called *linker*. In routine cases, linking is done automatically.

Figure 2: Preparing a C Program for Running


C Program
Compiler
Object code for C Program
Object code for Library functions
Linker
Complete machine language code ready to run