

Software: the Information Machine

Nitin Verma
mathsanew.com

April 8, 2020

We will try to look at the modern computer hardware and software with some abstraction. And with their abstracted form, understand their most basic structure and working.

We can look at the computer hardware (hereafter referred just as “hardware”) consisting of these two only:

1. a memory area: series of memory locations, each location able to contain 1 byte of 0s and 1s. This is where all information can be stored, in its 0/1 form. Locations are numbered 1, 2, 3, . . . to identify them.
2. a processor: a device which can perform very basic operations like: read or write a byte at some memory location, very basic arithmetic operations like add, subtract, increment, decrement, compare with given two memory-locations. Each such operation performed can be called a command, an *Instruction*.

Note, the modern hardware can store its information in multiple places like registers, cache, main-memory, hard-disk etc. But we have generalized all such types of available memory to a single memory area: a series of memory-locations.

Carrying out only a single instruction need not be of some value. But we can perform one instruction after another, to reach at some goal. For example, consider 100 sequential memory-locations each containing an english alphabet in its byte-form. We may want to count the occurrences of ‘a’. So, a series of instructions can be formed, which will go over each location

Copyright © 2020 Nitin Verma. All rights reserved.

one-by-one, read and compare it with ‘a’ and if match then increment a counter placed at some other memory-location.

What we call a software or a program is mainly this: a series of instructions.

We can be able to solve many such problems by writing out a series of instructions carefully. Such software, the series of instructions, itself needs to be stored in a separate area within the memory. The processor would start at the 1st instruction and after performing it would move to the 2nd, then 3rd and so on.

Such software setup can be much more useful if the processor can be asked to jump to an instruction somewhere else after performing Nth instruction. Thus, a processor can also accept a special kind of instruction called *jump*. Jumps can be very useful. They allow to do repetitive tasks by performing a jump to the beginning of an instruction-portion again and again (*for-loops*). They also allow to execute alternative portions of instructions based on some conditions; say, whether a memory-location reads ‘a’ or ‘b’ (*if/else*). Using jumps, we can also reuse a portion of instructions under multiple situations, without having to duplicate it wherever it is needed (*procedures*).

Coming again to what is software. It is a series of instructions which are performed in sequence by the processor, until asked to jump to some other instruction and continue from there.

The hardware provides a place to store information via the memory area, and a few basic operations over that memory area via the processor. So, it is a basic information processing platform. The software provides an accurate sequence of operations to perform, to reach a specific goal. The very significance of the software is its fine specification of what instructions to perform in what order. Its correctness is completely driven by logic, and should be provable — thus it is a mathematical work. A software is an intangible outcome of analysis, design and creativity, and is as precious on paper as on the hardware.

Even the input and output to a software are intangible; they are just plain information encoded in byte-form. To make software easy to use, we have devices like keyboard and monitor to ease our writing/reading of information to/from the hardware’s memory area.

We can have better alternatives for the word “software” as this itself hardly conveys any meaning (except that in contrast with “hardware”). Probably, “information-processor”, which can be abbreviated for ease. After all, this special type of machine starts with information (input), processes it as per the series of instructions (the software itself), and produces some useful information (output).

The basic and generic nature of the hardware allows the execution of any number of software to solve different different problems. It gives us freedom to find multiple ways to solve any given problem. The single hardware platform thus allows the creation of an uncountable number of information-processing-machines called software. This makes the hardware-software combination a unique kind of machine we have ever made.