

Welcome Aboard

1 Overview

This is the first chapter of this book. In this chapter, we will introduce the ultimate goal of this lecture. You will learn some basic ideas of computer science. We will start at the foundation, and go on to add layer after layer, as we get closer and closer to what most people know as a full-blown computer.

2 Two major themes: abstraction & hardware vs. software

2.1 Abstraction

Abstraction is a technique for establish a simpler way for a person to interact with a system. Keep only the necessary part and omit the unnecessary details for people to interact.

Example

汽车实际上就是一种抽象，驾驶员开车的时候，只需要知道如何操作汽车，而不需要知道汽车是由什么构成的，什么驱动的汽车行驶等一系列细节上的问题。

Abstraction help us deal with all kinds of situations more efficiently. One can be effective without understanding what is below the abstraction as long as everything goes well.

However, there is an assumption for abstraction that *everything about detail is just fine*. If something went wrong in details, we need to go back to its real stuff.

2.2 Hardware vs. Software

Hardware generally means the physical computer and all the specifications associated with it. Software generally means the programs(like operating systems or application programs).

As we mentioned above, the charm of abstraction allows us to operate at a level without thinking about other layers. If you work at a higher level, it could be better if you understand the how does it work in lower level. By doing this, one can be able to a much better job. Thus, we urge you to take the approach that hardware and software are names for components of two parts of a computing system that work best when they are designed by people who take into account the capabilities and limitations of both.

Info

如果软件开发人员知道他们所用的机器的能力、局限性等方面的知识，他们能够因此设计出更适合于这台机器的软件。

3 Computer System

A computer system is a combination of software and hardware. The software directs and specifies the processing of information. The hardware performs the actual processing of information that the software asks the hardware to do. A more precise term of this hardware is a *Central Processing Unit*(CPU), or simply a *processor* or a *microprocessor*.

3.1 The Parts of a Computer System

Today, computer system is often a laptop with many additional devices rather than just a CPU. A computer system generally includes the processor, a key-board, a mouse, a monitor, a memory for temporarily storing information, disks and USB memory sticks. There are also some connections to other devices such as printer.

4 Two Very Important Ideas

- All computers (the biggest or the slowest) are capable of computing same things if they are given enough time.
- We need to transform the problem into program language and then transform into the voltage that can move the electrons.

5 How Do We Get the Electrons to Do the Work?

Problem(Natural language)
Algorithm (eliminate ambiguity)
Program (Python, C++, ...)
Instruction set architecture(ISA)(指令集架构)
Micro-architecture(微架构)
Logic circuit
Electronic circuit
Electrons

The table above shows the process we must go through to get the electrons (which actually do the work) to do our bidding.

5.1 Problem Statement

We describe our problems in "natural language"(such as Chinese, English...). But computer can not accept natural languages since natural languages are ambiguous(有歧义的). The electrons can only do as it is told, telling them something that are multiple interpretations would cause the computer to not know what to do.

5.2 The Algorithm

In this case, our first step is to translate the natural language into algorithm that do not have ambiguity. An algorithm is a step-by-step procedure that is guaranteed to terminate, such that each step is precisely stated and can be carried out by the computer.

There are some terms to describe the properties of algorithm:

- **definiteness**: each step is precisely stated. It should not contain some notions that are not precise
- **effective computability**: each step can be carried out by a computer. This means, a procedure should be computable for a computer.

- **finiteness**: the procedure can terminate.

5.3 The program

Our next step is to transform algorithm into programming language. Programming language is designed for use in specifying a sequence of instructions to a computer. Of course, programming languages are unambiguous.

5.4 The ISA

The next step is to translate the program into the instruction set of the particular computer that will be used to carry out the work of the program. The ISA of a computer specifies the interface between the computer program directing the computer hardware and the hardware carrying out those directions.

Example

汽车的 ISA 就是人需要知道他能让车做什么，以及车需要做到人指定的任务的规范。对于一辆车的踏板，人知道如果他踩下去，那么这辆车会刹车。车知道如果踏板受到了压力，车的硬件会让车停下。ISA 的作用就是将人踩刹车和车停下对应起来。

It is the definition that makes what **0s** and **1s** means.

The ISA of computers specify *opcodes*, *data types*, *addressing modes*, *the number of unique locations*, *the number of individual 0s and 1s*. The number of those can vary among different ISAs.

We need a compiler (编译器) to translate a high-level language to the low-level language that the computer supports. We need an assembler to translate the unique assembly language of a computer to binary code.

5.5 The Microarchitecture

The next step is the implementation of the ISA, referred to as its microarchitecture.

"如何理解 Microarchitecture (微架构)?"

你可以将其理解为如何根据指定的 ISA 设计架构，使得计算机能够运行该 ISA 下的指令。因此，一个 ISA 可以有多个微架构，但是一般一个微架构只能对应一个 ISA.

Example

所有的汽车都有相同的 ISA，例如所有的汽车中三个踏板的定义完全相同，即中间的是刹车、右边的是油门、左边的是离合器。而将 ISA 实现的具体组织（微结构）是指车盖板下的“内容”。所有的汽车，其制造和模型都不尽相同，这取决于设计者在制造之前所做的权衡决策，如有的制动系统采用刹车片，有的采用制动鼓；有的是八缸发动机，有的是六缸，还有的是四缸；有的有涡轮增压，有的没有。我们称这些差异

性的细节为一个特定汽车的“微结构”，它们反映了设计者在成本和性能之间所做的权衡决策。