

第11章

XML 究竟是什么

215

- 11.1 XML 是标记语言 217
- 11.2 XML 是可扩展的语言 219
- 11.3 XML 是元语言 220
- 11.4 XML 可以为信息赋予意义 224
- 11.5 XML 是通用的数据交换格式 227
- 11.6 可以为 XML 标签设定命名空间 230
- 11.7 可以严格地定义 XML 的文档结构 232
- 11.8 用于解析 XML 的组件 233
- 11.9 XML 可用于各种各样的领域 235

第12章

SE 负责监管计算机系统的构建

239

- 12.1 SE 是自始至终参与系统开发过程的工程师 241
- 12.2 SE 未必担任过程序员 243
- 12.3 系统开发过程的规范 243
- 12.4 各个阶段的工作内容及文档 245
- 12.5 所谓设计，就是拆解 247
- 12.6 面向对象法简化了系统维护工作 249
- 12.7 技术能力和沟通能力 250
- 12.8 IT 不等于引进计算机 252
- 12.9 计算机系统的成功与失败 253
- 12.10 大幅提升设备利用率的多机备份 255

计算机是怎样跑起来的

——本书将要讲解的主要关键词

基础中的基础知识（开端）

第1章

计算机的三大原则

输入、运算、输出、指令、数据、计算机的处理方式、计算机不断进化的原因

知识的范围

编程

第4章

程序像河水一样流动着

流程的种类、流程图、结构化编程、中断、事件驱动

第5章

与算法成为好朋友的七个要点

辗转相除法、埃拉托斯特尼筛法、鸡兔同笼问题、线性搜索、哨兵

第6章

与数据结构成为好朋友的七个要点

变量、数组、栈、队列、结构体、自我引用的结构体、列表、二叉树

第7章

成为会使用面向对象编程的程序员吧

类、可维护性、建模、UML、消息传递、继承、封装、多态

目标

第12章

SE负责监管计算机系统的构建

瀑布模型、文档、审核、设计方法、信息化、设备利用率

读完本书，便可了解有关计算机的“基础中的基础知识”“知识范围”以及“目标”。

硬件和软件

第2章

试着制造一台计算机吧

CPU、内存、I/O、时钟信号、IC、数据总线、地址总线、控制信号线、DMA

第3章

体验一次手工汇编

机器语言、汇编语言、操作码、操作数、寄存器、内存地址和I/O地址

数据库

第8章

一用就会的数据库

关系型数据库、DBMS、规范化、索引、SQL、事务回滚

网络

第9章

通过七个简单的实验理解TCP/IP网络

NIC、MAC地址、以太网、IP地址、DHCP、路由器、DNS、TCP

第10章

试着加密数据吧

字符编码、密钥、XOR运算、对称密钥加密技术、公开密钥加密技术、数字签名

第11章

XML究竟是什么

标记语言、元语言、CSV、命名空间、DTD、XML Schema、DOM、SOAP

本书的结构

本书共分为 12 章，每章由热身问答、本章要点和正文三部分构成。全书还穿插了 2 个专栏。

● 热身问答

在各章的开头部分设有简单的问题作为热身活动，请诸位务必挑战一下。设置这一部分的目的，是为了让诸位能带着问题阅读正文的内容。

● 本章重点

各章的本章要点部分揭示了正文的主题。诸位可以读一读，以确认这一章中是否有想要了解的内容。

● 正文

正文部分会以讲座的方式，从各章要点中提到的角度出发，对计算机的运行机制予以解释说明。其中还会出现用 Visual Basic 或 C 语言等编程语言编写的示例程序，编写时已力求精简，即便是没有编程经验的读者也能看懂。

● 专栏“来自企业培训现场”

专栏部分将会与诸位分享笔者自担任讲师以来，从培训现场收集来的各种各样的轶事。诸位可以时而站在讲师的角度、时而站在听众的角度读一读这部分。专栏部分不仅有严肃认真的话题，更有有趣逗乐的笑话，想必会对诸位有所帮助。

第 1 章

计算机的三大原则

热身问答

在阅读本章内容前，让我们先回答下面的几个问题来热热身吧。



初级问题

硬件和软件的区别是什么？

中级问题

存储字符串“中国”需要几个字节？

高级问题

什么是编码 (Code) ？

怎么样？被这么一问，是不是发现有一些问题无法简单地解释清楚呢？下面，笔者就公布答案并解释。

答案

初级问题：硬件是看得见摸得着的设备，比如计算机主机、显示器、键盘等。而软件是计算机所执行的程序，即指令和数据。软件本身是看不见的。

中级问题：在 GBK 字符编码下，存储“中国”需要 4 个字节。

高级问题：通常将为了便于计算机处理而经过数字化处理的信息称作编码。

解释

初级问题：硬件（Hardware）代表“硬的东西”，而软件（Software）代表“软的东西”。是硬的还是软的取决于眼睛能否看得到，或者实际上能否用手摸到。

中级问题：存储汉字时，字符编码不同，汉字所占用的字节数也就不同。在 GBK 字符编码下，一个汉字占用 2 个字节。而在 UTF-8 字符编码下，一个汉字占用 3 个字节。

高级问题：计算机内部会把所有的信息都当成数字来处理，尽管有些信息本来不是数字。用于表示字符的数字是“字符编码”，用于表示颜色的数字是“颜色编码”。

本章重点

.....

现在的计算机看起来好像是种高度复杂的机器，可是其基本的构造却简单得令人惊讶。从大约 50 年前的第一代计算机到现在，计算机并没有发生什么改变。在认识计算机时，需要把握的最基础的要点只有三个，我们就将这三个要点称为“计算机的三大原则”吧。无论是多么高深、多么难懂的最新技术，都可以对照着这三大原则来解释说明。

只要了解了计算机的三大原则，就会感到眼前豁然开朗了，计算机也比以往更加贴近自己了，就连新技术接连不断诞生的原因也明白了。本书以本章介绍的计算机的三大原则为基础，内容延伸至硬件和软件、编程、数据库、网络以及计算机系统。在阅读之后的章节时，也请诸位时常将计算机的三大原则放在心上。

.....



1.1 计算机的三个根本性基础

下面就赶紧开始介绍计算机的三大原则吧。

1. 计算机是执行输入、运算、输出的机器
2. 程序是指令和数据的集合
3. 计算机的处理方式有时与人们的思维习惯不同

计算机是由硬件和软件组成的。诸位可以把硬件和软件的区别理解成游戏机（硬件）和收录在 CD-ROM 中的游戏（软件）的区别。这样就能理解硬件和软件各自的基础了（三大原则中的第一点和第二点）。

在此之上，计算机有计算机的处理方式也是一条重要的原则。而且请诸位注意，计算机的处理方式往往不符合人们的思维习惯（三大原则中的第三点）。

计算机三大原则中的每一条，都是从事计算机行业 20 余年的笔者深切领悟出来的。诸位可以把这本书拿给你周围了解计算机的朋友看，他们应该会对你说“确实是这样的啊”“当然是这样的了”这类话。过去的计算机发烧友们在不知不觉中就能逐渐领悟出计算机的三大原则。而对于那些打算从今日开始深入接触计算机的普通人来说，三大原则中的有些地方也许一时半会儿难以理解，但是不要担心，因为下面的解释会力求让诸位都能理解三大原则的具体含义。

1.2 输入、运算、输出是硬件的基础

首先从硬件的基础开始介绍。从硬件上来看，可以说计算机是执行输入、运算、输出三种操作的机器。计算机的硬件由大量的 IC (Integrated Circuit, 集成电路) 组成 (如图 1.1 所示)。每块 IC 上都带有许多引脚。这些引脚有的用于输入，有的用于输出。IC 会在其内部对外部输入的信息进行运算，并把运算结果输出到外部。运算这个词听起来也许有些难以理解，但实际上就是计算的意思。计算机所做的事就是“输入”数据 1 和 2，然后对它们执行加法“运算”，最后“输出”计算结果 3。



图 1.1 IC 的引脚中有些用于输入, 有些用于输出

小型的 IC 自不必说，就连在观察银行的在线系统这类巨型系统时，或是编写复杂的程序时，也要时常把输入、运算、输出这三者想成是一套流程，这一点很重要。其实计算机就是台简单的机器，因为它只能做这三件事。

“你说得不对，计算机能做的事远比这些多得多。”也许会有人这样反驳笔者。的确，计算机可以做各种各样的事，比如玩游戏、处理文字、核算报表、绘图、收发电子邮件、浏览网页，等等。但是无论是多么复杂的功能，都是通过组合一个又一个由输入、运算、输出构成的流程单位来实现的，这是毋庸置疑的事实。如果打算用计算机做点什么的话，就要考虑该如何进行输入、如何获取输出以及进行怎样的运算才能从输入得到输出。

输入、运算、输出三者必须成套出现，缺一不可。这样说的原因有几点。首先，现在的计算机还没有发展到能通过自发的思考创造出信息的地步。因此不输入信息，计算机就不能工作。所以，输入是必不可少的。其次，计算机不可能不执行任何运算。如果只是使输入的信息绕过运算环节直接输出，那么这就是电线而不是计算机了。可以说不进行运算，计算机也就没有什么存在的意义。最后，输入的信息如果经过了运算，那么运算结果就必然要输出。如果不输出结果，那么这也不是计算机而只是堆积信息的垃圾箱了。因此，输出也必不可少。

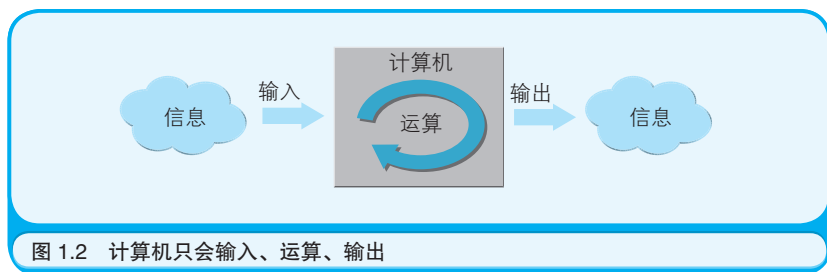


图 1.2 计算机只会输入、运算、输出

1.3 软件是指令和数据的集合

下面介绍软件，即程序的基础。所谓程序，其实非常简单，只不过是指令和数据的集合。无论程序多么高深、多么复杂，其内容也都是指令和数据。所谓指令，就是控制计算机进行输入、运算、输出的命令。把向计算机发出的指令一条条列出来，就得到了程序。这里成套出现的输入、运算、输出，就是之前在硬件的基础一节中说明过的流程。向计算机发出的指令与计算机硬件上的行为一一对应是理所当然的。

在程序设计中，会为一组指令赋予一个名字，可以称之为“函数”“语句”“方法”“子例程”“子程序”等。这里稍微说些题外话，在计算机行业，明明是同一个东西，却可以用各种各样的术语来指代它，这种现象请诸位注意。如果只想用一个名字的话，一般情况下笔者推荐称之为函数，因为这个名字通俗易懂。

程序中的数据分为两类，一类是作为指令执行对象的输入数据，一类是从指令的执行结果得到的输出数据。在编程时程序员会为数据赋予名字，称其为“变量”。看到变量和函数，诸位也许会联想到数学吧。正如数学中函数的表记方法那样，在很多编程语言中都使用着类似于下面的这种语法。

$$y = f(x)$$

这句话表示若把变量 x 输入到函数 f 中，经过函数内部的某种运算后，其结果就会输出到变量 y 中。因为计算机是先把所有的信息都表示成数字后才对其进行运算的，所以编程语言的语法类似数学算式也就不足为奇了。但是在程序中有一点与数学不同的是，变量和函数的名字都可以由一个以上的字符构成，比如下面这种情况。

```
output = operate(input)
```