



程序员书库

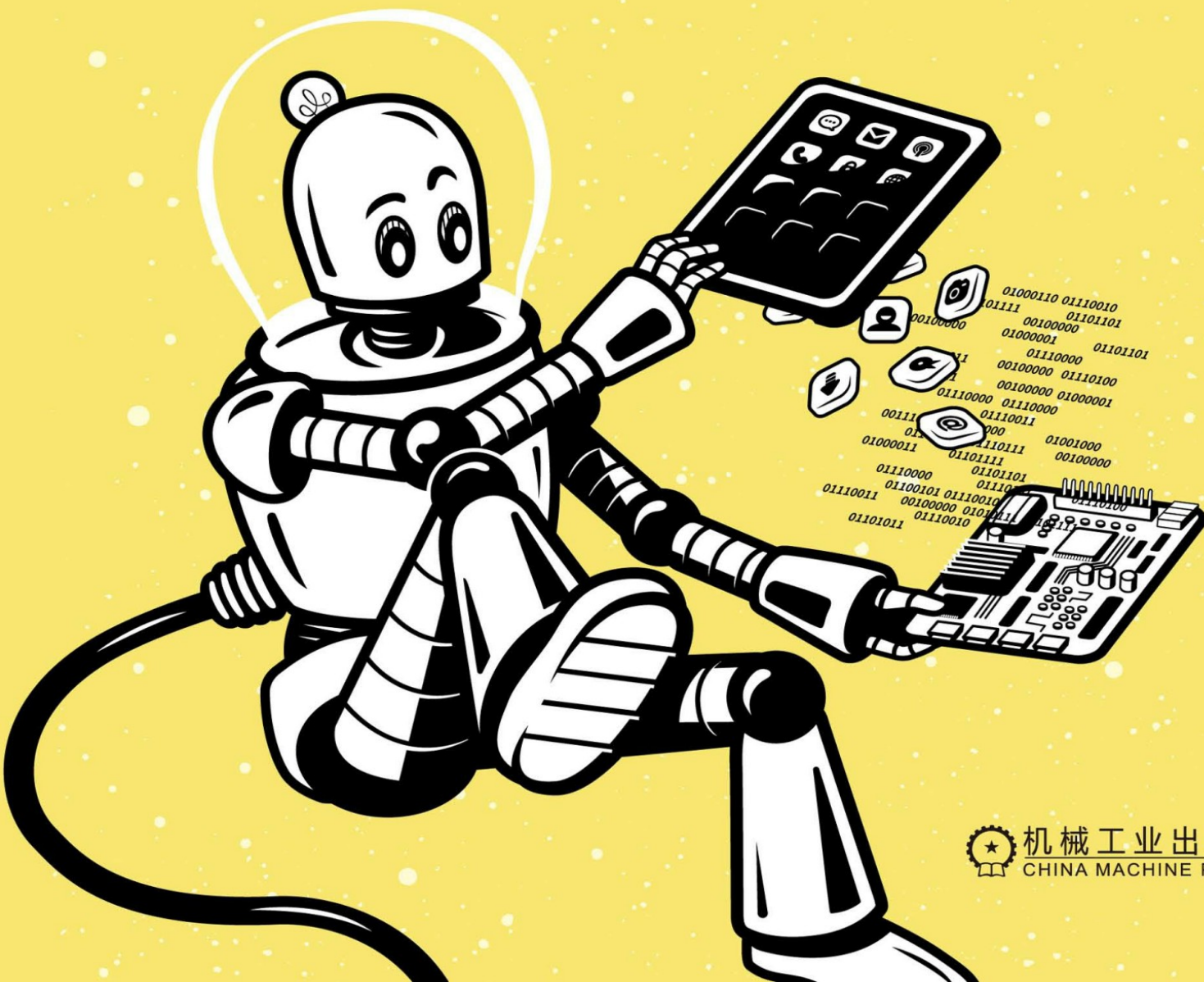
# 计算机是如何工作的

## 人人都能懂的计算机软硬件工作原理

### HOW COMPUTERS REALLY WORK

A HANDS-ON GUIDE TO THE INNER WORKINGS OF THE MACHINE

[美] 马修·贾斯蒂斯 (Matthew Justice) 著 贺莲译



机械工业出版社  
CHINA MACHINE PRESS

## 前言

你是否好奇计算机是如何工作的？获得对计算机的广泛理解通常是一个漫长而曲折的过程。问题不在于缺乏相关文档，在网络上快速地搜索一下就能找到很多致力于解释计算机工作原理的书籍和网站。编程、计算机科学、电子学、操作系统.....那里有大量的信息。这是件好事，却会让人望而生畏。那么，你应该从哪里开始呢？一个主题是如何联系到另一个主题的？本书的目的是为大家提供一个切入点，方便大家学习计算机的关键概念，并了解这些概念是如何结合在一起的。

在我担任工程经理时，我面试了很多软件开发方面的求职者。通过与他们的交谈，我了解到他们知道如何编写代码，但是相当一部分人似乎并不了解计算机实际上是如何工作的。他们知道如何让计算机执行命令，但并不理解幕后的情况。对这些采访结果的反思，以及我自己努力学习计算机的经历，促使我撰写了这本书。

我的目标是用一种易于理解、可动手操作的方式呈现计算机的基础知识，让抽象概念更加真实。本书并没有深入介绍每个主题，而是主要介绍计算机的基础概念，并将这些概念联系起来。我希望大家能在脑海中勾勒出计算机是如何工作的，这样就能深入挖掘感兴趣的课题了。

计算机无处不在，随着我们的社会越来越依赖于技术，我们需要广泛理解计算机的人才。我希望本书能帮助大家获得广阔的视野。

### 本书读者

本书适合想要了解计算机工作原理的任何人。读者不必具备与所述主题相关的预备知识，因为本书是从零开始介绍的。如果你已经有编程或电子学方面的背景，那么本书可以帮助你扩展其他领域的知

识。本书是为以下这些自学者编写的，他们熟悉基础数学和科学，且熟练使用计算机和智能手机，但仍然对它们的工作原理存有疑问。本书的内容对教师也有用，我相信书中的内容设计非常适合课堂讲解。

## **本书内容**

本书把计算机看作一个技术栈。现代计算设备（如智能手机）就是由技术层组成的。这个栈的底层是硬件，顶层是应用程序，顶层与底层之间是多个技术层。层次模型的优点在于，每一层都受益于较低层的全部功能，但任意给定层都只需要建立在其下面一层上就可以了。在介绍了一些基础概念后，我们将自下而上地逐层讲解该技术栈，从电子电路开始，一直推进到驱动网络 and 应用程序运行的技术。以下是各章所包含的内容。

第1章涵盖计算机的基础概念，比如模拟和数字、二进制数字系统和SI（国际单位制）前缀。

第2章探索如何用二进制表示数据和逻辑状态，介绍逻辑运算符。

第3章解释电学和电路的基本概念，包括电压、电流和电阻。

第4章介绍晶体管和逻辑门，并总结第2章和第3章的概念。

第5章展示如何用数字电路执行加法运算，进一步揭示数字是如何在计算机中表示的。

第6章介绍存储器设备和时序电路，演示如何用时钟信号进行同步。

第7章介绍计算机的主要组成部分：处理器、存储器和输入/输出。

第8章展示处理器执行的低级机器码，介绍汇编语言——一种人类可读形式的机器码。

第9章介绍不依赖于特定处理器的编程语言，包含C语言和Python语言的代码示例。

第10章介绍操作系统系列以及操作系统的核心功能。

第11章讲解互联网的工作原理及常用网络协议套件。

第12章解释网络的工作原理及核心技术：HTTP、HTML、CSS和JavaScript。

第13章概述一些现代计算机主题，如app、虚拟化和云计算。

阅读本书时，你将会看到用于解释概念的电路图和源代码。这些都是教学工具，主要是为了直白地讲解内容，而不是为了说明工程师在设计硬件和软件时需要考虑的性能、安全性等因素。换句话说，本书中的电路和代码虽然能帮助你领会计算机是如何工作的，但它们不见得是最好的例子。同样，书中的技术示例也偏向于简单，不够完整。有时，我会省去某些细节，以免介绍得太复杂。

## **关于练习和设计任务**

本书几乎每章都配有练习和实践设计任务。这些练习是让你动脑或用纸笔来解决的问题。实践设计任务不仅是脑力练习，还经常涉及电路搭建或计算机编程。

做这些设计任务需要购买一些硬件（附录B给出了所需组件的列表）。之所以要加这些设计任务，是因为我相信最好的学习方法就是自己去尝试。如果想从本书获得最大收益，那么一定要去完成这些实践设计任务。尽管如此，即使你一个电路都不搭建，一行代码都不输入，也可以继续跟进书中所呈现的内容。

附录A中给出了练习的答案，相应章末给出了每个设计任务的详细资料。附录B包含了启动这些设计的信息，在需要时，设计文本可以为你指明方向。

各设计任务所用源代码可以从  
<https://www.howcomputersreallywork.com/code/>获得。你还可以在  
<https://nostarch.com/how-computers-really-work/>上访问本书，了解相关更新内容。

## 我的计算机之旅

我对计算机的痴迷可能是从小时候玩电子游戏开始的。每当我去拜访祖父母时，我都会连着几小时玩我阿姨的Atari 2600游戏机上的*Frogger*、*Pac-Man*和*Donkey Kong*。后来，当我上五年级的时候，我的父母在圣诞节把任天堂娱乐系统当作礼物送给我，我好激动！虽然我喜欢玩《超级马里欧兄弟》和《双截龙》游戏，但随着时间的推移，我开始好奇电子游戏和计算机是如何工作的。可惜的是，我的任天堂游戏机并没有为我提供许多关于其内部情况的线索。

大约是在同一时期，我家购买了第一台“真正的”计算机——苹果II GS，它为我打开了一扇新的大门，让我去探索这些机器究竟是如何工作的。幸运的是，我的初中开设了一门关于苹果II计算机的BASIC编程课程，我很快就发现我对探索编程的需求是无止境的！我会在学校写好代码，把它复制到软盘带回家，然后在家里继续编程。在整个初中和高中阶段，我学到了更多关于编程的知识，我确认自己非常喜欢它。我还开始意识到，尽管BASIC和其他类似的编程语言能相对容易地告诉计算机做什么，但它们还是隐藏了很多关于计算机工作原理的细节，我希望在这方面能更加深入。

我在大学学习的是电气工程，我开始了解电子电路和数字电路。我学习了C语言和汇编语言课程，终于了解了计算机是如何执行指令的。计算机工作原理的底层细节也开始变得有意义起来。在大学，我

还开始学习被称为“万维网”（World Wide Web）的新事物，我甚至还制作了自己的网页（这在当时似乎很了不起）！我开始编写Windows应用程序，并开始接触UNIX和Linux。这些内容有时似乎与数字电路和汇编语言的硬件具体细节相去甚远，我很想知道它们是如何结合在一起的。

大学毕业后，我很幸运地在Microsoft公司找到一份工作。在那里的17年中，从调试Windows内核到开发Web应用程序，我从事过各种软件工程师工作。这些经历使我对计算机有了更广、更深的理解。我和许多非常聪明且知识渊博的人一起共事，我认识到，关于计算机总是有学不完的知识。理解计算机是如何工作的，已经成为我一生的追求，我希望能通过本书把我学到的一些东西传递给大家。

## 致谢

非常感谢我的妻子Suzy，她是我的非正式编辑，给我提供了宝贵的反馈意见。她通读了本书的多版草稿，仔细地检查了每个字和每个概念，帮助我完善了自己的思路并清晰地把它们表达出来。在从本书设想到完成这一过程中，她都鼓励并支持着我。

感谢我的女儿Ava和Ivy，她们阅读了本书早期的草稿，帮助我从年轻学习者的视角来审视我的作品。她们帮助我避免了混淆性语言表述，并告诉我哪里需要花更多的时间进行解释。

我还想感谢我的父母Russell和Debby Justice，他们总是信任我并给我提供许多学习的机会。我对文字的热爱来自我的母亲，而我的工程思维则来自我的父亲。

感谢No Starch出版社的整个团队，尤其是Alex Freed、Katrina Taylor和我的文字编辑Rebecca Rider。这是我第一次写书，No Starch出版社的编辑在整个过程中给予我耐心的指导。他们发现了我没有考虑到的改进机会，并帮助我清楚地表达自己的想法。我对出版团队能带来的价值有了新的认识。

我要感谢本书的技术审稿人John Hewes、Bryan Wilhelm和Bill Young，他们仔细检查了本书的细节。他们的贡献使本书内容更加准确、完整，每个人都给出了独特的见解，并分享了宝贵的专业知识。

感谢Microsoft公司多年来指导我以及和我合作的所有人员。很幸运能与他们这些才华横溢、聪明博学的人一起工作，由于篇幅所限，不能一一列出。我之所以能写出这样一本书，是因为一些杰出的微软人愿意花时间与我分享他们的知识。

## 技术审稿人简介

Bill Young博士是得克萨斯大学奥斯汀分校计算机科学系的教学副教授。在2001年入职得克萨斯大学之前，他在该行业已经拥有20年的经验。他虽然专门研究形式化方法和计算机安全，但经常教授计算机体系结构等课程。

Bryan Wilhelm是软件工程师。他拥有数学和计算机科学学位，并且已经在Microsoft公司工作了20年，从Windows内核调试到商业应用程序开发都属于其工作范围。他喜欢阅读、科幻电影和古典音乐。

John Hewes从小就开始搭建电路，十几岁的时候就从事电子项目设计。之后，他获得了物理学学位，并继续发展他对电子学的兴趣，作为科学技术人员，他还帮助在校生完成他们的项目。John曾在英国教授电子学和物理学的高级知识，并为11~18岁的孩子开设了校园电子俱乐部，并建立了支持该俱乐部的网站<http://www.electronicclub.info/>。他相信，不论年龄和能力，每个人都可以享受构建电子项目的乐趣。



## 第1章

# 计算机的概念

现在，计算机无处不在：家里、学校、办公室——你可能在自己的口袋里、手腕上，甚至冰箱上发现计算机。当前，找到和使用计算机比以往任何时候都更容易，但却很少有人真正了解计算机是如何工作的。这并不奇怪，因为学习计算机的困难过程会让人难以应对。本书的目标是：以一种任何有好奇心并有一点技术的人都能跟得上的方式来展示计算机的基本原理。在深入研究计算机工作原理之前，我们先花点时间来熟悉计算机的一些主要概念。

本章我们首先讨论计算机的定义，然后介绍模拟数据和数字数据之间的差异，之后探索数字系统和用于描述数字数据的术语。

## 1.1 计算机的定义

让我们从一个基本问题开始：什么是计算机？当人们听到“计算机”这个词的时候，大多数人会想到笔记本计算机或台式计算机——有时也被称为个人计算机或PC。这是本书会涉及的一类设备，但是让我们的思考角度更广一些。想一想智能手机。智能手机当然也是计算机，它们执行与PC同类型的操作。事实上，对现在的很多人来说，智能手机是他们主要的计算机设备。如今的大多数计算机用户还依赖互联网，互联网由服务器支持，服务器是另一类计算机。每当浏览网站或使用连接到互联网的app时，其实都在与一个或多个连接到全球网络的服务器进行交互。电子游戏机、健身追踪器、智能手表、智能电视.....这些都是计算机！

计算机是可以被编程并执行一组逻辑指令的电子设备。有了这个定义，就会发现许多现代设备实际上都是计算机！

### 练习1-1：发现你家里的计算机

花点时间，看看在你家里能识别出多少台计算机。当我和家人一起做这个任务的时候，我们快速识别出了大概30台设备！

## 1.2 模拟和数字

你可能听说过计算机被描述为数字设备。这与诸如机械时钟的模拟设备相反。但是，这两个术语究竟是什么意思？理解模拟与数字之间的差异是理解计算机的基础，所以，让我们仔细来看看这两个概念。

### 1.2.1 模拟方法

看看你的周围并挑选出一个对象。问问自己：它是什么颜色的？有多大？有多重？回答这些问题其实就是在描述这个对象的属性，或者说数据。现在，再挑一个不同的对象并回答同样的问题。如果对更多的对象重复这个过程，你就会发现其中的每个问题都有无数潜在的答案。你可以选择一个红色的对象、一个黄色的对象，或一个蓝色的对象，甚至是三原色混合颜色的对象。这种类型的变化不仅适用于颜色。对于给定的属性，在我们这个世界的各个对象之间所发现的属性变化可能是无限的。

口头描述对象是一回事，但是，想更精确地测量它的一个属性则是另一回事。例如，如果想测量一个对象的重量，需要把它放到秤上。这个秤会根据其上放置的对象的重量，顺着数字刻度轴移动指针，直到达到与重量对应的位置才停止。从秤上读取该数字，就得到了该对象的重量。

这种测量很常见，但是让我们再思考一下我们是如何测量这个数据的。秤上指针的位置不是实际的重量，它是重量的一种表示形式。指针指向的数字刻度轴为我们提供了一种简单的方式将代表重量的指针位置和重量的数值进行转换。换句话说，虽然重量是对象的一个属性，但在这里我们可以通过别的东西（指针在数字刻度轴上的位置）来理解这个属性。指针的位置会按照秤上对象的重量成比例地变化。因此，秤可以作为一种模拟方法，我们可以通过秤的指针在数字刻度轴上的位置来理解对象的重量。这就是我们把这种测量方法称为模拟方法的原因。

模拟测量工具的另一个例子是水银温度计。水银的体积会随着温度的上升而增大。温度计制造商利用这个特性把水银放进玻璃管中，玻璃管上不同的刻度对应不同温度下水银的预期体积。因此，水银在玻璃管中的位置就可以用来表示温度。注意，对于这两个例子（秤和温度计），当我们进行测量时，我们可以把仪器上的刻度转换成特定的数值。但是，我们从仪器上读到的值只是一个近似值。指针和水银的真实位置可以是仪器最小刻度之间的任何地方，我们向上或向下舍入到最近的刻度值。所以，虽然看上去这些工具似乎只能产生一组有限的测量值，但这种限制来源于数字转换，而不是模拟方法本身。

在人类历史的大多数时间里，人类都是用模拟方法进行测量的。但是，人们并不仅仅使用模拟测量方法，他们还设计了用模拟方式存储数据的巧妙方法。留声机唱片使用调制沟槽作为已录制音频的模拟表示。沟槽的形状随其路径而变化，变化方式与音频波随时间的变化相对应。沟槽不是音频，但它是原始声音波形的模拟。胶片相机与之类似，它把胶片短暂地曝光于来自相机镜头的光线，从而导致胶片发生化学变化。胶片的化学性质不是图像，但它表示了所捕获的图像，是图像的模拟。

### **1.2.2 数字化**

所有这些与计算机有什么关系呢？事实证明，数据的模拟表示对于计算机来说是难以处理的。模拟系统类型是如此丰富多样，以至于创造一个能理解全部这些系统的通用计算设备几乎是不可能的。例如，创造一台能测量水银体积的机器与创造一台能读唱片沟槽的机器是完全不同的任务。此外，计算机还需要能高度可靠和准确地表示某些类型的数据，比如数字数据集和软件程序。数据的模拟表示难以精确测量，而且精度会随着时间衰减，在被复制的时候可能失去保真度。计算机需要一种方式来表示所有类型的数据，且格式能被准确地处理、存储和复制。

如果我们不想把数据表示为可能无限变化的模拟值，那么该怎么办呢？我们可以采用数字方法。数字系统把数据表示成一系列符号，其中每个符号都代表一组有限值中的一个。现在，这个描述听起来有点格式化和令人困惑，这里没有深入讲解数字系统的理论，而是解释这在实践中意味

着什么。在几乎所有的现代计算机中，数据都被表示为两个符号的组合，这两个符号就是0和1。虽然数字系统可以使用两个以上的符号，但是增加符号会提高系统的复杂度和成本。只采用两个符号能简化硬件并提高可靠性。大多数现代计算机中的所有数据都是用0和1的序列来表示的。从此刻开始，当谈及数字计算机时，你可以认为我说的系统只处理0和1，而不是其他一些符号集。

有一点需要强调：计算机上所有的数据都是以0和1的形式存储的。你最近用智能手机拍的一张照片也是如此吗？是的，设备把这张照片存储为0和1的序列。从网上下载的歌曲呢？0和1的序列。在计算机上编写的文档呢？0和1的序列。安装的app呢？也是0和1的序列。访问的网站呢？还是0和1的序列。

我们只用0和1便可表示自然界中的无限值，这听起来似乎是有局限性的。怎么把音乐录音或精细的照片精简为0和1呢？很多人发现，用有限的“词汇”表示复杂的思想是违反直觉的。这里的关键是数字系统使用的是0和1的序列。例如，一张数字照片通常是由数百万个0和1组成的。

那么，这些0和1到底是什么呢？你可能会看到其他用来描述这些0和1的术语，如假和真、关和开，以及低和高等。这是因为计算机不能按字面意思来存储数字0或1。它存储的是一系列的项（entry），每一项都只有两种可能的状态。每一项都像电灯的开关一样，要么开，要么关。实际上，这些1和0的序列是以各种方式存储的。在CD或DVD上，0和1是以盘片上凸起（0）或平坦（1）的形式表示的。在闪存驱动器中，1和0以电荷的形式存储。硬盘驱动器通过磁化与否来存储0和1。正如你将在第4章看到的，数字电路用电平来存储0和1。

在我们继续介绍后面的内容之前，关于“模拟”最后需要注意的一点是：它通常只用来表示“非数字”。例如，工程师可能会说“模拟信号”，其含义是信号是连续变化的，不符合数字值特性。换句话说，它是非数字信号，但并不一定代表其他东西的模拟。所以，当你看到“模拟”这个词的时候，要考虑到它并不一定总是你想的意思。

## 1.3 数字系统

到目前为止，我们已经确定计算机是处理0和1的数字机器。对许多人来说，这个概念看起来有些奇怪，他们习惯用0到9来表示数字。如果我们限制自己只使用2个符号，而不是10个符号，那我们怎么表示大的数字呢？要回答这个问题，我们需要回顾一下小学的数学知识：数字系统。

### 1.3.1 十进制数

我们通常用所谓的十进制位值记数法（decimal place-value notation）来书写数字。让我们来分析一下。位值记数法（或按位记数法）是指被书写数字的每个位置都代表一个不同的数量级，十进制（或以10为基数）是指数量级的因数是10，每个位置可以是0~9这十个不同符号中的一个，参见图1-1的位值记数法示例。

在图1-1中，该数字用十进制位值记数法写作275。5在个位上，代表它的值是 $5 \times 1 = 5$ 。7在十位上，代表它的值是 $7 \times 10 = 70$ 。2在百位上，代表它的值是 $2 \times 100 = 200$ 。所有位置代表的值的总和为 $5 + 70 + 200 = 275$ 。



图1-1 十进制位值记数法表示的275

很容易吧？你可能在一年级就明白了。但是，让我们仔细研究一下为什么最右边的是个位？为什么下一个位置是十位？这是因为我们采用的是十进制，或者以10为基数，所以每个位置的权重就是10的幂，如图1-2所

示，最右边的位置是10的0次幂，也就是1，下一个位置是10的1次幂，也就是10，再下一个位置是10的2次幂，也就是100。

$$\begin{array}{ccc} 2 & 7 & 5 \\ \hline 10^2 & 10^1 & 10^0 \\ 10 \times 10 & 10 & 1 \end{array}$$

图1-2 在十进制位值记数法中，每个位置的权重都是10的幂

如果需要用十进制表示大于999的数，就在左边再增加一位，即千位，它的权重等于10的3次幂（ $10 \times 10 \times 10$ ），也就是1000。继续按这个模式扩展，我们就可以得到任意大的数字。

我们已经知道了为什么不同的位置有不同的权重，让我们继续深挖一下为什么每个位置都使用符号0~9？当使用十进制时，我们只有10个符号，因为根据定义，每个位置只能表示10个不同的值。0~9是目前所使用的符号，但其实可以使用任何一组10个具有唯一性的符号，这其中的每个符号对应一个特定的数值。

大多数人喜欢把以10为基数的十进制系统作为数字系统。据说这是因为我们有10个手指和10个脚趾，但不管理理由是什么，现代世界的大多数人阅读、书写和思考数字时都使用十进制。当然，这只是我们集体选择用来表示数字的一种约定。正如我们前面提到的，这种约定不适合应用在计算机上，计算机只使用了两个符号。让我们看看在限定两个符号的同时，如何运用位值系统。

### 1.3.2 二进制数

只包含两个符号的数字系统是以2为基数，或是二进制的。二进制系统仍然是位值系统，所以其基本机制与十进制系统相同，但是它也有一些变化。首先，每个位置的权重是2的幂，而不是10的幂。其次，每个位置只能是2个（而不是10个）符号中的一个，这两个符号就是0和1。图1-3举例说明了如何用二进制表示一个数。

图1-3中给出了一个二进制数：101。对你来说，这个数看起来可能挺像一百零一，但在二进制中，它实际表示的是5！如果你想口头表达出来，那么可以读作“二进制二零一”。

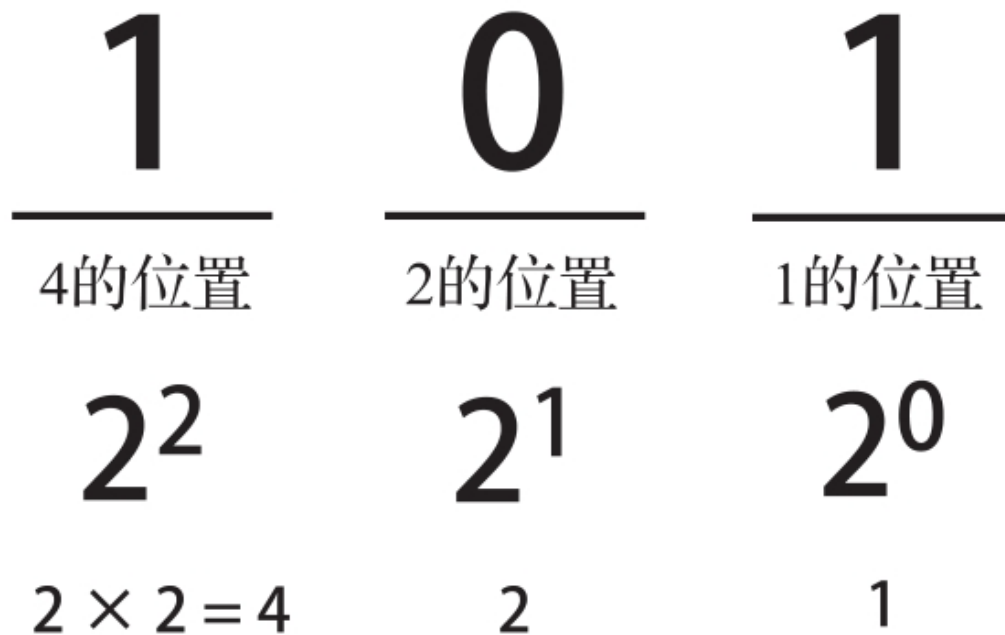


图1-3 用二进制位值记数法表示的十进制数5

就像十进制一样，每个位置都有一个等于基数各次幂的权重。当基数等于2时，最右边的位置是2的0次幂，值为1；下一个位置是2的1次幂，值为2；再下一个位置是2的2次幂，值为4。另外，和十进制相同的是，要得