

<<编码>>

图书基本信息

书名：<<编码>>

13位ISBN编号：9787121181184

10位ISBN编号：7121181185

出版时间：2012-10

出版时间：电子工业出版社

作者：查尔斯·佩措尔德

页数：420

字数：442000

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

内容概要

这是一本讲述计算机工作原理的书。不过，你千万不要因为“工作原理”之类的字眼就武断地认为，它是晦涩而难懂的。作者用丰富的想象和清晰的笔墨将看似繁杂的理论阐述得通俗易懂，你丝毫不会感到枯燥和生硬。更重要的是，你会因此而获得对计算机工作原理较深刻的理解。这种理解不是抽象层面上的，而是具有一定深度的，这种深度甚至不逊于“电气工程师”和“程序员”的理解。

作者简介

Charles Petzold

是Windows编程界的一位大师，当今世界顶级技术作家。

1994年5月，Petzold作为仅有的七个人之一（并且是唯一的作家）被《视窗杂志》和Microsoft公司联合授予Windows

Pioneer奖，以表彰他对Microsoft Windows所做出的贡献。

Petzold从1984年开始编写个人计算机程序，从1985年开始编写Microsoft Windows程序。

1986年他在Microsoft Systems

Journal的12月号上发表了第一篇关于Windows程序设计的论文。

从1986年到1995年，Petzold为PC

Magazine撰写专栏文章，向读者介绍Windows和OS/2程序设计等方面的知识。

直到今天他依然保持着Windows

GDI程序设计首席技术作家的地位。

其大作Programming

Windows（Windows程序设计）是尽人皆知的Windows编程经典，曾深深地影响过一代程序员，该书目前已出至第5版。

<<编码>>

书籍目录

第1章 至亲密友

编码是什么？

在本书中，这个词的意思是指一种用来在机器和人之间传递信息的方式。

换句话说，编码就是交流。

对任何能听见我们的声音并理解我们所说的语言的人来说，我们发出的声音所形成的词语就是一种编码。

用手电筒能代替声音来与朋友交谈吗？

当然值得一试。

第2章 编码与组合

莫尔斯码也被称做二进制码，因为这种编码的组成元素只有两个——“点”和“划”。

不过，点、划的组合却可以表示你想要的任意数目的码字。

这其中的规律是什么？

本章我们就来探讨一番。

第3章 布莱叶盲文与二进制码

布莱叶盲文是为了便于盲人阅读而发明的一种编码。

在这一章中我们将解析布莱叶盲文，来看看它是如何工作的。

我们并不是要真的学习布莱叶盲文，而且也无须刻意记住关于它的什么内容。

我们仅仅希望从中归纳出编码的一些性质。

第4章 手电筒的剖析

为了理解电在计算机中的工作原理，我们先得仔细钻研一番电学，不过不要担心，只需要一部分基础知识就够了。

在本章，我们将以手电筒为教学道具，引导你走入神秘的电学世界。

第5章 绕过拐角的通信

在

第1章，我们曾经讲过用手电筒与朋友进行交谈的方法，但是这样的方式是有局限性的，你的交流对象必须住在街对过，而且你们卧室的窗口正好相对。

但是，现实不会总是如此。

当手电筒的光无法到达朋友的卧室时，怎样与他们进行无声的交流呢？

电路或许可以助你一臂之力。

第6章 电报机与继电器

全球性即时通信对于我们来说已经司空见惯，你要是生活在19世纪早期，可没这么方便。

你当然可以进行即时通信或者远距离通信，但是不能同时做到这两点。

即时通信受声音传播距离的限制，或者受视野的限制。

使用信件倒是可以进行更远距离的通信，但是寄信耗费的时间太多，并且需要借助于交通工具。

为了解决这个问题，电报应运，而生而伴随着电报诞生的继电器更是具有重要意义的伟大发明。

第7章 我们的十个数字

人们很容易理解，语言只不过是一种编码。

比如英文中的“cat”（猫）在其他语言中可以写做gato、chat、Katze、KOIIIK或k á tta。

然而，数字似乎并不是那么容易随文化的不同而改变。

<<编码>>

不论我们说什么语言，或使用什么样的发音，在这个星球上的所有人都用以下方式来书写数字：0, 1, 2, 3, 4, 5, 6, 7, 8, 9。

你了解这十个数字么？

第8章 十的替代品

对于人类而言，10是一个非常重要的数字。

它是我们大多数人拥有的手指或脚趾的数目。

我们人类已经适应了以10为基数的数字系统。

但是只能使用十进制来计数吗？

如果人类像卡通人物那样每只手只有4根手指会怎样？

第9章 二进制数

二进制是最简单的数字系统，其中只包含两个数字：0和1。

二进制中的1位（bit）称为1比特，我们可以用它来表达简单的信息：是或不是；亮或灭；打开或关闭，等等。

而事实上只要信息能转换成两种或多种可能性的选择，就都可以用比特来表示。

这种例子在日常

生活中随处可见，比如照相机胶卷的胶片速度，各种商品包装上的条形码。

第10章 逻辑与开关

对于古希腊人而言，逻辑是在追求真理的过程中所使用的一种分析方法，是一种哲学形式。

而英国的数学家乔治布尔却认为可以找到一种数学形式来描述逻辑，因此他发明了布尔代数。

更重要的是，布尔代数运算可以用开关、导线和灯泡组成的电路来实现，布尔代数中的AND和OR，与线路中开关的串联和并联，有着奇妙的对应关系。

第11章 门

继电器像开关一样，可以串联或并联在电路中执行简单的逻辑任务。

这种继电器的组合叫做逻辑门（logic gate），也简称门。

这里提到的逻辑门执行“简单”逻辑任务是指逻辑门只完成最基本的功能。

本章就介绍那些用以完成最基本逻辑任务的门。

第12章 二进制加法器

加法是算术计算中最基础的运算，如果想搭建一台计算机的话，首先就要搭建出计算两个数加和的器件。

本章我们将利用前面的章节中用过的开关、灯泡、导线、电池、逻辑门等这些简单的元件，搭建一个二进制加法器。

第13章 如何实现减法

当你确信继电器连接在一起真的可以实现二进制数加法的时候，你可能会问：“如何实现减法呢？”问得好！

这表明你是相当有觉察力的，加法和减法在某些方面互相补充，但是在机制上二者却存在本质区别。不过，没关系，我们可以想一些办法，把减法运算变成加法。

第14章 反馈与触发器

想象一下，如果你没有了记忆力，该如何去数数？

我们不记得刚刚数过的数，当然也就无法确定下一个数是什么！

<<编码>>

同理，一个能计数的电路必定需要触发器。
本章要介绍的就是各种触发器。

第15章 字节与十六进制

在前面的章节中，加法器、锁存器以及数据选择器的输入和输出形式都是8位的数据流，也即数据路径的位宽为8，为什么要定义为8位呢？
为什么不是6位、7位、9位或10位？
本章就要解释其中的缘由。

第16章 存储器组织

每天清晨，我们将自己从沉睡中唤醒，这时大脑的空白会很快被记忆填满。
我们立刻会意识到自己身在何方，最近做了些什么事情，有什么计划打算。
有的事情我们很快就能想起来，但有时并非如此。
我们可以借助许多工具来记录信息，比如笔和纸、磁带，当然现在还可以使用存储器。

第17章 自动操作

人类的本性中带有一些懒惰的特质。
我们总是抵触繁重的工作，对枯燥的、重复性的工作深恶痛绝。
所以，当你必须用前面搭建的加法器计算100个数，甚至更多个数的加法时，有一种念头就会不可遏制地从脑子里冒出来：怎样让加法器自动地完成数据输入和计算呢？
办法肯定是有的，那就是编写程序。

第18章 从算盘到芯片

算盘、滑尺、纳皮尔骨架、差分机、解析机、继电器、电子管、晶体管、芯片、计算机；甘特、帕斯卡、莱布尼兹、杰奎德、巴贝芝、图灵、冯诺依曼、香农；IBM、贝尔实验室……你觉得应接不暇了吗？
把这些你或者熟悉或者生疏的名词和名字串起来，就是人类的计算工具发展史。
让时光倒流，去看看那些精巧的工具，感受天才们的巧思吧！

第19章 两种典型的微处理器

将中央处理器的所有构成组件封装到一块硅芯片上，就得到了微处理器。

第一片微处理器芯片诞生于1971年，即Intel

4004系列，其中集成了2300个晶体管，你或许觉得可笑——如今家用计算机的微处理器上所安置的晶体管数量已经以亿为计量单位了。
但是，从本质上来说，微处理器实际所做的工作并没有变。
在本章，我们就来看看两种有着辉煌历史的典型微处理器：Intel 8080和Motorola 6800。

第20章 ASCII码和字符转换

计算机中的存储器唯一可以存储的形式是比特，因此如果想在计算机上处理信息，就必须把它们转换为比特的形式来存储。
我们已经掌握了如何用比特来表示数字和机器码。
如何用它来存储文本呢？
毕竟，人类所积累的大部分信息，都以各种文本形式保存的。
下面就轮到ASCII码出场了！

<<编码>>

第21章 总线

一台计算机包括很多部件：中央处理器、存储器、输入/输出设备等。

通常这些部件按照功能被分别安装在两个或更多的电路板上。

这些电路板之间通过总线（bus）通信。

如果对总线做一个简单的概括，可以认为总线就是数字信号的集合，而这些信号被提供给计算机上的每块电路板。

第22章 操作系统

你或许梦想过自己组装一台近乎完整的计算机，像老木偶匠盖比特雕刻木偶匹诺曹一样，全部亲自动手用小零件完成。

不过在你的机器能完成你想要的操作之前，还差一个重要的东西——操作系统！

第23章 定点数和浮点数

整数、分数以及百分数等各种类型的数字与我们形影不离，它几乎出现在我们生活的所有角落。

例如你加班2.75小时，而公司按正常工作时间的1.5倍支付你工资，你用这些钱买了半盒鸡蛋并交了8.25%的销售税。

在计算机的内存里，所有的数都表示为二进制形式。

通过前面的学习，我们知道2用二进制可以表示为102，可是2.75用二进制怎样表示呢？

这就是本章的主题。

第24章 高级语言与低级语言

第22章介绍了如何编写一段简单的程序，让我们可以利用键盘将十六进制机器码输入计算机，以及通过视频显示设备来检查这些代码。

但是使用机器码编写程序就如同用牙签吃东西，伸出手臂费半天劲刺向食物，但每次都只获取到小小的一块，用这种低级语言编写程序既费力又费时，有悖于我们发明计算机的初衷。

不过，人们想出了一种效率更高的编程方法——使用高级语言。

第25章 图形化革命

回顾历史，从

第一台继电器计算器出现到现在为止，六七十年过去了，计算机的处理速度飞速增长。

不过要充分利用计算机日益增长的运算和处理能力，就必须不断改进计算机系统用户接口（User Interface），因为它是人机交互的轴心。

图形化革命来了！

章节摘录

版权页：插图：只显示文本的视频显示适配器还必须支持光标（Cursor）功能。

光标是一个小小的下划线，用来表明从键盘上输入的下一字符会在屏幕的什么位置显示出来。

光标所在的行和列常被存储在两个8位的寄存器中，这两个寄存器也是视频板的一部分，而且微处理器可以对其进行写操作。

有的显示适配器不仅仅只显示文本，还可以显示其他数据，我们称这样的显示适配器为图形适配器（图形显卡）。

通过向图形显卡上的RAM写入数据，微处理器就可以画出图形了，当然能显示各种大小和样式的文本。

相比较而言，图形显卡要比只显示文本的显卡所需的存储空间更大。

320x200的图形显卡有64,000个像素，如果每个像素需要1位RAM，那么这样的图形显卡就需要64,000位的RAM，即8000字节。

然而，这只是最低的要求。

1位是和1个像素相对应的，只能用来表示两种颜色——例如黑白两色。

0可能对应于黑色像素，1可能对应于白色像素。

让我们仔细观察一下黑白电视机，很快会发现，它们不仅仅只显示黑色和白色，还能显示不同灰度的色彩。

为了让图形显卡拥有这种功能，通常每个像素对应于RAM中的一整个字节，其中00h表示的是黑色，FFh表示的是白色，介于两者之间的数值对应不同的灰度。

一个320 x 200的视频板若能显示256种灰度，就需要64,000字节的RAM。

这与一直在讨论的某个8位微处理器的整个地址空间非常接近。

如果想显示出丰富多彩的颜色，每个像素就需要至少3个字节。

如果现在你手头有放大镜的话，不妨用它观察一下彩色电视机或计算机视频显示器，你会发现，每种颜色都是由红、绿、蓝三原色的不同组合而形成的。

为了获取所有的颜色，三原色中每种颜色的强度都需要用一个字节来表示。

这么算来，就需要192,000字节的RAM（更多有关彩色图形的内容将在本书最后一章介绍）。

图形显卡到底能显示出多少种不同的颜色呢？

这与每个像素所赋予的比特数是有关的。

<<编码>>

编辑推荐

不管你是计算机高手，还是对这个神奇的机器充满敬畏之心的菜鸟，都不妨翻阅《编码:隐匿在计算机软硬件背后的语言》一下，读一读大师的经典作品，必然会有收获。

<<编码>>

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>