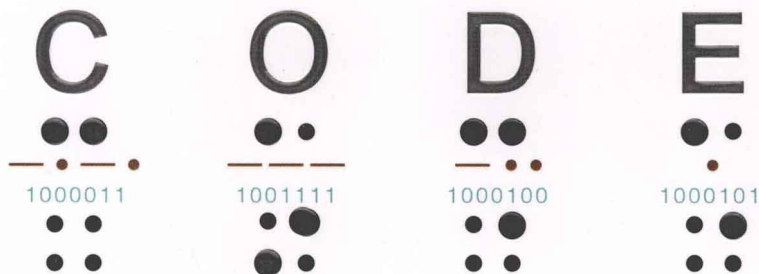


100001110011111000100100010110000111001111100010010001011000011100111110001001000101

永不褪色的计算机科学经典著作

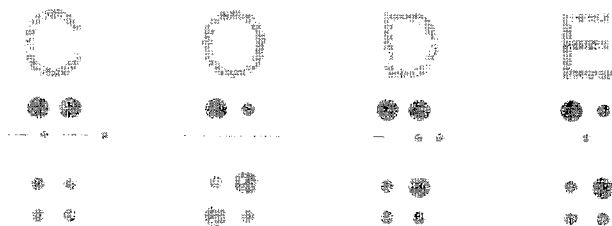


编 码

隐匿在计算机软硬件背后的语言

Code: The Hidden Language of
Computer Hardware and Software

[美] Charles Petzold 著
左飞 薛佟佟 译



编 码

隐匿在计算机软硬件背后的语言

Code: The Hidden Language of
Computer Hardware and Software

[美] Charles Petzold 著

左飞 薛佟佟 译

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

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。

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，都不妨翻阅一下本书，读一读大师的经典作品，必然会有收获。

Original English language Edition ©2000 by Charles Petzold. All rights reserved. Chinese edition published by arrangement with the original publisher, Microsoft Corporation, Redmond, Washington, U.S.A.

本书中文简体版专有出版权由 Microsoft Corporation 授予电子工业出版社，专有出版权受法律保护。

版权贸易合同登记号 图字：01-2009-3031

图书在版编目（CIP）数据

编码：隐匿在计算机软硬件背后的语言 / (美) 佩措尔德 (Petzold, C.) 著；左飞，薛佟佟译.

北京：电子工业出版社，2012.10

书名原文：Code: The Hidden Language of Computer Hardware and Software

ISBN 978-7-121-18118-4

I. ①编… II. ①佩… ②左… ③薛… III. ①计算机科学—理论 IV. ①TP3-0

中国版本图书馆 CIP 数据核字(2012)第 204356 号

策划编辑：刘 皎

责任编辑：许 艳

印 刷：三河市双峰印刷装订有限公司

装 订：三河市双峰印刷装订有限公司

出版发行：电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本：720×1000 1/16 印张：27.25 字数：442 千字

印 次：2012 年 10 月第 1 次印刷

定 价：59.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：(010) 88254888。

质量投诉请发邮件至 zits@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线：(010) 88258888。

编码 (biān mǎ)

3. a. 一种在信息传输过程中用来表述字母或数字的信号系统。
b. 由被赋予了一定主观意义的符号、字母以及单词所组成的系统，该系统可用于传输需要保密的或简短的信息。

4. 一种由若干符号和规则组成的系统，用来向计算机表述指令。

——摘自《美国传统英语词典》

code (kɒd) ...

3. a. A system of signals used to represent letters or numbers in transmitting messages.

- b. A system of symbols, letters, or words given certain arbitrary meanings, used for transmitting messages requiring secrecy or brevity.

4. A system of symbols and rules used to represent instructions to a computer...

— The American Heritage Dictionary of the English Language

推荐序

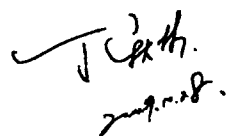
《道德经》有云：大方无隅，大象无形，也就是老子所说“道”的至高境界。世界上最恢宏、壮丽的气派和境界，往往并不拘泥于一定的事物和格局，而是表现出“气象万千”的面貌和场景，科学知识的论述也是如此。科学不一定等同于晦涩和深奥，最复杂的理论也可用最简单的方式描述。这就是普遍存在于现实世界的辩证的、朴素的唯物主义精神。

我想 Charles Petzold 就是这样一位智者。作为 Windows 编程界开创者之一和世界顶级的技术专家，他在 Windows 编程方面著述丰富，在基础计算机理论书籍方面也取得了卓尔不凡的成就。他善用丰富的想象和清晰的笔墨将看似繁杂的理论描述得生动、简单，使读者不会感到枯燥和生硬。尤其这本畅销多年、经久不衰的《编码——隐匿在计算机软硬件背后的语言》(*Code: The Hidden Language of Computer Hardware and Software*) 更是化腐朽为神奇，改晦涩为通俗之杰作。即使读者不是计算机专业的人士，在 Petzold 笔墨的引导之下，也能够轻松阅读该书。书中使用像电线、电灯泡、触发器等这些非常常见的零件，拼拼接接，最后令人惊异地建造了一台电子计算机。读者在阅读过程中，循序渐进地学习有关计算机系统基础和构造的知识，作者的奇思妙想和妙趣横生的讲述方式不得不令人赞叹。

透过现象进而探索本质可以发现新事物；将复杂的事物简单化，可以发现解决问题的新方法。作者将那些看似复杂的理论转换成我们熟悉的诸多景象，使读者不由自主地跟随他的思路打造这台只有电线、电灯泡和触发器组构成的庞大的计算机系统。

我阅读了本书的译稿，并非常欣喜地将其推荐给每一位读者，我相信读者都能获益良多。本书的译者做了一件非常有意义的事情，将一部优秀的作品介绍给读者。他们严谨的工作将本书诠释得非常出色，使之在不丧失原作风采的基础上更贴合中国读者的习惯。

最后，我还是想将《道德经》上的两句话来送给各位读者——“为学日益，为道日损，损之又损，以至于无为，无为而无不为”。衷心希望各位读者有朝一日都能够在学问上到达无为而治的境界。



丁秋林
2009.11.28

丁秋林教授，博士生导师，南京航空航天大学信息科学与技术学院和机电工程学院资深教授。英国皇家航空学会高级会员、英国剑桥世界名人录传记人物（IBC Cambridge the 28th Edition）、美国世界名人录传记人物（Who's Who in the World the 16th Edition）、国际高级制造技术杂志编辑部成员（Springer）。

译者序

在想到要给《编码——隐匿在计算机软硬件背后的语言》(*Code: The Hidden Language of Computer Hardware and Software*)写一篇序言的时候,内心就不由得激动起来。能够翻译这部经典之作实属吾辈之荣幸。早在一年多之前,电子工业出版社的编辑跟我提到这本书的翻译之事时,我便有颇大的兴趣。一是由于该书作者 Charles Petzold 的鼎鼎大名,二是由于该部作品本身内容之不凡。

作为 Windows 编程界的一位传奇人物, Charles Petzold 早在 1994 年就作为仅有的七人之一(并且是唯一的作家)被《视窗杂志》和 Microsoft 公司授予 Windows Pioneer 奖。而作为一名世界顶级的技术作家,他的大作 *Programming Windows* 则是尽人皆知的 Win32 API 编程经典,是学习 Windows 编程的必读之作。令我们欣喜的是,除了 Windows 程序设计方面的著作, Petzold 还撰写了这部讲述计算机工作原理的书籍,而且同样堪称经典。

本书的翻译过程漫长而又愉快,书中的精彩内容不断给我们以惊喜。Petzold 采用的叙述方式奇特却又如此自然,他首先从历史的角度审视了计算机技术的发展脉络,继而自底向上地将计算机体系结构方面的知识娓娓道来。在 Petzold 的笔触下,导线、灯泡以及触发器等简单部件,通过精妙的组装,竟然构建出了一台电子计算机,令我们在翻译的过程中不禁有动手一试的冲动,但总是又被他精彩的笔触所吸引而欲罢不能。Petzold 在这本技术书籍里旁征博引,从美国当红的影评人到百年之前的布莱叶盲文,从史密斯·索尼娅的滑尺到冯·诺依曼的构想,从石破天惊的论文《思维之际》到图形化革命,看似信手拈来,然则处处别出心裁。

这些被有机结合起来的零星知识，仿佛又把我带回了大学的课堂。所涉及的问题包括计算机体系结构、汇编语言及数字电子技术等等。那些已经久远而又模糊的知识骤然间变得异常清晰而且格外生动。这些知识原本只会出现在高校的计算机专业课堂里，令人惊奇的是，尽管遍布于世界各地的本书读者往往有着各不相同的专业背景，而这却丝毫不影响本书内容所带给他们的震撼！

所谓经典，就是历经岁月的洗涤，依然毫不褪色，然其光彩却历久弥新。我想本书被奉为经典之作当之无愧。

为了能够将该书的精彩原原本本地呈现给广大中国读者，并在保证质量的前提下力争让该书早日与广大读者见面，本书的译者始终秉持着谨慎严肃的态度，对许多词句都推敲再三，力求精益求精。在此期间，我们得到了许多师长和友人的支持与帮助，感激之情溢于言表。特别是南京航空航天大学丁秋林教授，他对我们的翻译工作给予了极大的支持，并欣然为本书提笔作序，在此我们对丁教授表示诚挚的谢意。

参与本书翻译工作的人员还有初甲林、刘悦、高新、高阳、邓明强和赵德志。此外赵学健、欧阳键两位博士也参与了部分章节的审校工作，在此也向他们一并表示感谢。

在本书即将付梓之际，除了激动喜悦之外，译者又多了一份惶恐，尽管我们几经校对，并在翻译过程中始终以谨慎求实的态度对一些技术细节和翻译方式反复讨论，力求表意准确。然而限于时间仓促，纰漏和欠缺之处在所难免，言语之中有失偏颇之处，还望读者不吝赐教和批评。联系信箱：beckham@vip.163.com。

薛佟佟

2009 年秋于南京



作者序

在真正开始撰写这本书之前，纷繁的思绪在我的脑海中已经萦绕了近十年之久。但从着手写作，一直到出版，我不断反复地问着自己一个问题：这本书是讲什么的？

关于这个问题我总是找不到合适的答案。我想说：“这本书将带领大家通过一段信息技术革命的旅程重新感触现代计算机技术。”语毕，我内心纠结，这个回答真的将这本书的内容完整诠释出来了吗？

最后我不得不承认：“这本书其实是讲述计算机如何工作的。”

我之所以有些底气不足，那是因为我已经猜到了某些读者的反应：“嗨，这种书我早就读过了。”但我会立刻告诉他：“不，这种书你并没有读过。”对于这点，我坚信不疑。因为这本书并不是讲“计算机——如何——工作”。书中并没有用一张又一张描述磁盘的插图，用各种箭头解释数据是如何被输送到计算机的，书里面也没有装载着一节又一节“0, 1 数据”的“火车”图片。明喻与暗喻是文学描述中精妙的辅助手段，但它们常常掩盖了科学技术的真正光芒。

我还听过另一个说法：“人们不愿意了解计算机如何工作。”对于这句话我毫无保留地赞同，因为就我个人来说，也是在不经意之间才体会到理解事物如何运作这一过程是多么有趣。但是这并不意味着我对世上一切事物的运作机制都感兴趣，都有所了解，因此我绝对不会在任何场合向大家解释我房间里冰箱的工作原理。

然而与此同时，经常有人向我问起一些涉及计算机内部运行机理的问题。一个常见

的例子就是：“‘存储器（storage）’与‘内存（memory）’有何区别？”

对于计算机用户而言，这个问题要是搞不清楚，的确“寸步难行”，可以看到，市场上考察个人计算机的存储性能，最主要的就是这两个概念。即便对于最初级的计算机用户来说，他们也一定需要了解到底多少“兆字节”或多少“吉字节”的存储器才能应对运行在其上的程序。如果进一步去思考，这些初级用户或许更加想了解计算机中的“文件”是什么概念，甚至连带这些文件如何从存储器加载进内存，又如何从内存存储到存储器，他们也非常期望学习这些知识。

像这类“存储器-内存”问题的解答大都使用类比法：“内存就好比你的办公桌，而存储器就好比你的文件柜。”就问题本身而言，它的确给出了满意的回答。但我对此答案并不满意，主要原因在于这个答案将计算机的体系结构与办公室的结构等同起来，但是内存与存储器的区别其实是在逻辑层面上的，它体现着计算机体系结构的实际需求与存储器客观性能之间的矛盾，简单地说就是我们找不到一种同时具备这两种存储器所有优点的存储媒介，这些优点就包括存储速度快、存储容量大、非易失性等等。今天的计算机都采用“冯·诺依曼体系结构”——五十年来它一直是计算机体系结构的主导，而内存与存储器的区分也正是由于这种体系结构的不足所导致的。

还有一些计算机用户问我这样一个问题：“为什么 Macintosh 环境下的程序不能在 Windows 下运行？”我想立刻回答这个问题，但在我刚刚张开嘴的时候就立刻意识到，这个问题涉及太多的技术细节，要想彻底搞清楚，那提问的这位朋友也一定非准备和我来一次“茶话会”不可。

我希望这本书能够成为大家理解这些问题的“助手”，这种理解我希望不是抽象层面上的，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。我同时也非常希望大家能够理解：计算机是二十世纪技术领域的“登峰造极之作”，它是一种值得欣赏、具有“美”学文化底蕴的人类伟大成果，这种“美”不需要明喻与暗喻的额外修饰。

计算机拥有与生俱来的层次化体系结构，这种结构的底层是晶体管，其顶层则是计算机显示器上所呈现的信息。自底向上分析该结构的每一层——这也是本书的编写结构——其实这一切并没有人们想象中那么难。当然，现代计算机的内部结构不断推陈出新，但其本质上仍然是一些常见且简捷的操作集合。

尽管今天的计算机比起 25 年前，以及 50 年前的都复杂许多，但它们在本质上是完全一致的。学习技术发展史的重要意义正在于此：追溯的历史越久远，技术的脉络就变得越来越清晰。因此，我们需要做的就是确定某些关键的历史阶段，在这些阶段，技术最天然、最本质的一面将清晰可见。

在这本书中，我回溯了自己所能找得到资料的计算机发展史。令我自己也感到惊讶的是，有时竟然一直追溯到 19 世纪，甚至使用了早期的电报设备来演示计算机是如何构建的。至少从理论角度来看，本书的前 17 章中提到过的所有设备，都可以利用已经存在了一个世纪的简单电子器件来构造。

这些古董级技术的使用令本书蒙上了一层怀旧的面纱。我要强调的是 *Code* 是这样一本书，它永远不会被命名为诸如 *The Faster New Faster Thing* 或 *Business @ the Speed of a Digital Nervous System*。在这本书中，“bit”定义在第 70 页，“byte”的定义则出现于第 186 页。晶体管直到第 146 页才被介绍，而且只是顺便提及而已。

在对计算机工作原理介绍方面，本书将会一直深入到本质（例如，只有少数书会去介绍计算机处理器的实际工作机制），但整本书的节奏是相对缓和的。在保证内容深度的基础上，我尽量使读者在学习的旅程中保持轻松愉悦的心情。

我最后还是要说，书里面没有那种装载着一节又一节“0, 1 数据”的“火车”图片。

Charles Petzold

2000 年 8 月 16 日



目 录

1

至亲密友 / 1

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。

2

编码与组合 / 7

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。

3

布莱叶盲文与二进制码 / 13

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。

4

手电筒的剖析 / 21

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。

5

绕过拐角的通信 / 32

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。

6

电报机与继电器 / 40

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。

7

我们的十个数字 / 47

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、KOIHK或kátta。然而，数字似乎并不是那么容易随文化的不同而改变。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？

8

十的替代品 / 55

对人类而言，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？

9

二进制数 / 71

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。

10

逻辑与开关 / 90

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。

11

门 / 108

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。

12

二进制加法器 / 135

加法是算术计算中最基础的运算，如果想搭建一台计算机，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。

13

如何实现减法 / 147

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。

14

反馈与触发器 / 160

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!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。

15

字节与十六进制 / 186

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。

16

存储器组织 / 197

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。

17

自动操作 / 215

人类的本性中带着一些懒惰的特质。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。

18

从算盘到芯片 / 252

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?把这些你或熟悉或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!

19

两种典型的微处理器 / 276

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。

20

ASCII码和字符转换 / 307

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！

21

总线 / 325

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。

22

操作系统 / 346

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！

23

定点数和浮点数 / 365

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 10_2 ，可是2.75用二进制怎样表示呢？这就是本章的主题。

24

高级语言与低级语言 / 381

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。

25

图形化革命 / 398

回顾历史，从第一台继电器计算机到现在为止，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中的用户接口（User Interface），因为它是人机交互的轴心。图形化革命来了！