



计算思维之 快乐编程

(初级·C++描述)

沈军 薛志坚 张婧颖 管赋胜 谢志峰 · 编著



东南大学出版社
SOUTHEAST UNIVERSITY PRESS

计算思维之快乐编程

(初级·C++描述)

沈 军 薛志坚 张婧颖 管赋胜 谢志峰 编著



东南大学出版社
SOUTHEAST UNIVERSITY PRESS

• 南京 •

内 容 提 要

本书围绕程序设计基本方法的构建及应用,以思维能力培养为目标而展开。全书分为8章,第1章主要解析程序设计的相关概念和环境;第2、3章分别解析程序两个DNA(数据组织和数据处理)的基本构建方法及其C++语言支持机制;第4章主要解析面向功能方法的构建及其C++语言支持机制;第5章主要解析面向对象方法的构建及其C++语言支持机制;第6、7章主要解析两种程序设计基本方法的基本应用(第6章主要解析排序,第7章主要解析搜索);第8章主要解析程序设计的应用要素。

本书主要面向青少年程序设计科普活动的教学与培训,也可以作为新课标程序设计相关课程的教学参考和辅导教材,同时也适用于爱好程序设计的广大读者的科普。

图书在版编目(CIP)数据

计算思维之快乐编程. 初级. C++描述 / 沈军等编著.

—南京:东南大学出版社,2019.6

ISBN 978-7-5641-8336-3

I. ①计… II. ①沈… III. ①C++语言—程序设计

IV. ①TP312.8

中国版本图书馆CIP数据核字(2019)第046830号

计算思维之快乐编程(初级·C++描述)

JISUAN SIWEI ZHI KUAILE BIANCHENG(CHUJI·C++ MIAOSHU)

编 著 沈 军 薛志坚 张婧颖 管赋胜 谢志峰

出版发行 东南大学出版社

出 版 人 江建中

责任编辑 张 煦

社 址 南京市四牌楼2号 (邮编:210096)

经 销 全国各地新华书店

印 刷 兴化印刷有限责任公司

开 本 787 mm × 1092 mm 1/16

印 张 13.75

字 数 326千

版 次 2019年6月第1版

印 次 2019年6月第1次印刷

书 号 ISBN 978-7-5641-8336-3

定 价 46.00元

本社图书若有印装质量问题,请直接与营销部联系。电话(传真):025-83791830

前 言

随着泛计算社会的到来,程序设计正逐步成为每个人的一种生活、工作和娱乐习惯。程序设计涉及方法、语言、环境和应用四个基本要素,其中,方法是核心,语言、环境和应用都围绕方法而展开,语言和环境支持方法的实现,应用体现方法的具体运用。更进一步,四个要素由计算思维实现统一,它们都是计算思维的具体运用。因此,程序设计的本质在于培养对计算机学科核心——计算思维的认知与灵活运用。

目前,伴随着 AI 的发展,程序设计的教学与培训如火如荼,尤其是针对青少年的程序设计教学和培训,受商业利益的驱动及受制于对计算机学科特征的肤浅认知,其教学目标、理念和方法等存在严重问题,最突出的一点就是机械式训练,缺乏针对思维能力的培养。更为重要的是,这些教学模式带来了几乎不可逆转的反作用。因为思维的培养有其时间属性,并且,思维模式及习惯的形成很难转变。因此,针对青少年的程序设计教学和培训而言,其启蒙教学十分重要。如何有效地辅助他们构建良好的思维模式,培养其计算思维及系统化思维应用能力,成为程序设计教学和培训应有的目标。

作者长期从事江苏省青少年信息学奥赛及科普活动,对青少年程序设计教学与培训的相关动态有比较深入的了解。同时,作为普通高校教师,作者对计算思维有着深入的研究,开设相关课程和讲座,并将研究成果应用于江苏省青少年信息学奥赛训练教练组的教学实践。基于作者对计算思维及其应用的长期研究及多年的实践,构建了本书的体系结构。本书的主要特色是:(1)从元认知层面深入解析了程序设计相关的知识及其关系;(2)强调计算思维的具体应用,注重培养计算思维;(3)结合计算机学科固有特征,采用多维度共进方式展开;(4)考虑青少年的认知特征,采用游戏方式引入相关知识和内容,并隐式地诠释计算思维原理及其具体运用;(5)兼顾奥赛实战应用。

本书由东南大学计算机学院沈军教授(江苏省青少年信息学奥赛委科学委员会主任委员)策划并设计,沈军和管赋胜(苏州工业园区星海实验中学)编写第 1、2 章,沈军和谢志峰(泰州市第二中学附属初级中学)编写第 3、4 章,沈军和张婧颖(江苏省青少年科技中心)编写第 5、6 章,沈军和薛志坚(江苏省淮阴中学)编写第 7、8 章,沈军的学生周晓、崔效伟、刘少希、王晓风等对全书样例进行了调试并参与部分审阅工作。

特别感谢东南大学计算机学院、江苏省青少年信息学奥赛委员会、东南大学出版社对本书出版工作的大力支持,衷心感谢东南大学出版社张煦编辑为本书出版所做的工作。

本书中的观点都是基于作者的认识、理解和感悟,难免存在错误和不妥之处,希望读者来信批评与指正。作者恳切盼望各位同仁来信切磋,作者的 E-mail 地址是 kutushen@126.com,junshen@seu.edu.cn

(扫描书后二维码获取各章习题参考答案)

作 者

2018 年 12 月 30 日于古都金陵

目 录

第1章 Hello World!	1
1.1 神奇宝贝小C	1
1.2 神奇宝贝小C的基因——二进制	2
1.2.1 进位计数制	2
1.2.2 二进制	3
1.3 如何与神奇宝贝小C进行交流	5
1.3.1 基本交流方式——人机交互式接口	5
1.3.2 程序交流方式——程序式接口	6
1.4 baby程序的诞生	12
1.5 进一步认识Dev-Cpp的能力	14
1.5.1 语法错误调试	14
1.5.2 语义错误调试	15
本章小结	17
习题	18
第2章 “2+3”的游戏	20
2.1 程序如何看世界:数据与数据类型	20
2.2 程序如何存放一个数据:常量和变量	22
2.3 程序如何存放一组数据:数据之间的关系	24
2.3.1 堆叠	24
2.3.2 关联	26
2.3.3 绑定	27
2.4 构建数据组织的基本方法	28
2.5 程序设计中常用的数据组织形态	28
2.5.1 数据组织形态及其描述	28
2.5.2 实战应用	34
2.5.3 STL中预定义的常用数据组织结构	39
本章小结	43

习题	43
第3章 “5+2”的游戏	46
3.1 程序中如何表达运算:表达式	46
3.1.1 概述	46
3.1.2 表达式的神奇魔力——蕴含的计算思维	47
3.2 程序如何描述基本处理:基本语句	47
3.2.1 计算赋值语句	47
3.2.2 输入输出语句	47
3.2.3 注释语句与空语句	52
3.2.4 逻辑控制语句	52
3.3 程序如何描述复杂处理:基本语句之间的堆叠与嵌套	54
3.4 程序如何建立数据处理的基本方法	61
3.5 程序中常用的数据处理方法及其描述	61
3.6 实战应用	72
本章小结	75
习题	75
第4章 “积木块”的构造与搭建	81
4.1 编写程序的基本方法	81
4.2 如何构造程序“积木块”	81
4.3 如何搭建程序“积木块”	83
4.4 程序“积木块”的一种特殊搭建方法——递归	88
本章小结	92
习题	92
第5章 “积木块”的进化及搭建	97
5.1 如何让程序“积木块”更加完美	97
5.2 如何构造新的程序“积木块”	97
5.3 如何搭建新的程序“积木块”	99
5.3.1 继承	99
5.3.2 多态	100
5.3.3 嵌入	101
5.4 程序构造的宝库——STL	102
5.5 C++ STL 中的常用新程序“积木块”	102

5.6 新程序“积木块”对程序设计的影响	106
5.6.1 类机制的元驱动力	106
5.6.2 第二代程序设计方法的诞生	107
本章小结	107
习题	108
 第6章 让“数据世界”变得有序	112
6.1 如何让数据有序化	112
6.2 有序化方法的世界	112
6.2.1 常用有序化方法的基本图谱	112
6.2.2 基于求最值并放到指定位置基本小方法的排序方法	113
6.2.3 基于数据分组基本小方法的排序方法	119
6.2.4 排序方法的维度拓展	123
6.2.5 常用有序化方法的高级问题	127
6.3 实战应用	131
本章小结	134
习题	134
 第7章 寻找心仪的“她”	138
7.1 认识“她”	138
7.2 如何寻找“她”	139
7.3 寻找“她”的常用方法	140
7.3.1 面向线性数据组织结构的基本查找方法	140
7.3.2 面向层次型/网状型数据组织结构的基本搜索方法	141
7.3.3 基于映射的查找方法	147
7.3.4 穷举所有可能的查找方法	150
7.3.5 搜索方法的基本优化	154
7.4 实战应用	160
本章小结	166
习题	166
 第8章 “m + n”的游戏	178
8.1 什么是“m + n”	178
8.2 构建自己的“m”	178
8.3 学会“n”	179

8.4 如何运用“m+n”	183
8.4.1 学习的思维桥梁	183
8.4.2 难题的奥秘	183
8.4.3 从问题中映射“m”和“n”	187
8.5 实战应用	191
本章小结	197
习题	197
附录 A C++ 语言定义的运算符	199
附录 B ASCII 字符集	201
附录 C 标准库常用字符串处理函数(cstring 库)	202
附录 D 标准库字符串类型 string(C++ string 类定义)	204
附录 E C++ STL 常用算法简介	207
参考文献	212

第 1 章 Hello World!

1.1 神奇宝贝小 C

工具是人类文明的载体,烙下了人类文明的印迹。我们生活在信息时代,信息文明的表征工具就是神奇宝贝小 C,也就是电脑或计算机(Computer)。神奇宝贝小 C 无所不能,可以看电影、听音乐、玩游戏,也可以办公和教学,还可以控制飞机和汽车,几乎可以实现你想要做的任何事情。小 C 为什么神奇呢?因为它给予我们一个创造的环境,可以让我们充分发挥自己的思维潜能。可是它是怎么做到的呢?

首先,工程师们利用晶体管和电阻器、电容器等物理器件搭建出几种基本的数字电路单元,这些电路单元可以使输出信号与输入信号之间产生一定的逻辑关系。例如:与门、或门、非门和异或门,等等;然后,再由这些基本电路单元搭建出各种各样具有特定功能的电路单元,例如:存储单元、计算单元、译码单元、控制单元,等等;最后,再依据大科学家冯·诺伊曼提出的通用计算机器的结构(如图 1-1 所示),利用这些电路单元搭建出神奇宝贝小 C 的躯体(人们通常称之为硬件或硬件系统)。

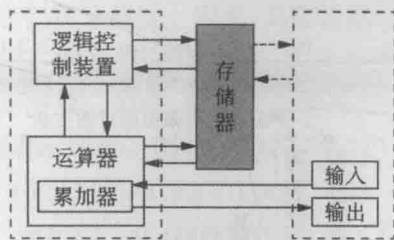


图 1-1 计算机硬件的逻辑结构

躯体的诞生只是具备了神奇的基础,真正神奇的是应该赋予小 C 一个大脑,让其具有记忆和判断能力,能够模拟人类的思维和表达人类的意愿。因此,工程师们首先在躯体中通过电路单元的综合搭建,让小 C 能够认识和执行少量固定的动作,每个动作称为指令,所有动作称为指令系统。然后,允许人们按照自己的想法,通过利用这些指令(相当于各种句子),构造一个程序(相当于书写一篇作文),让小 C 去阅读和理解。从而,使人类各种各样的奇思妙想得以实现。相对于躯体,各种各样的程序,以及相关的文档资料构成了软件。可见,正是采用了硬件(不变部分)+软件(可变部分)的结构,神奇宝贝小 C 具备了创造的条件和环境。

1.2 神奇宝贝小C的基因——二进制

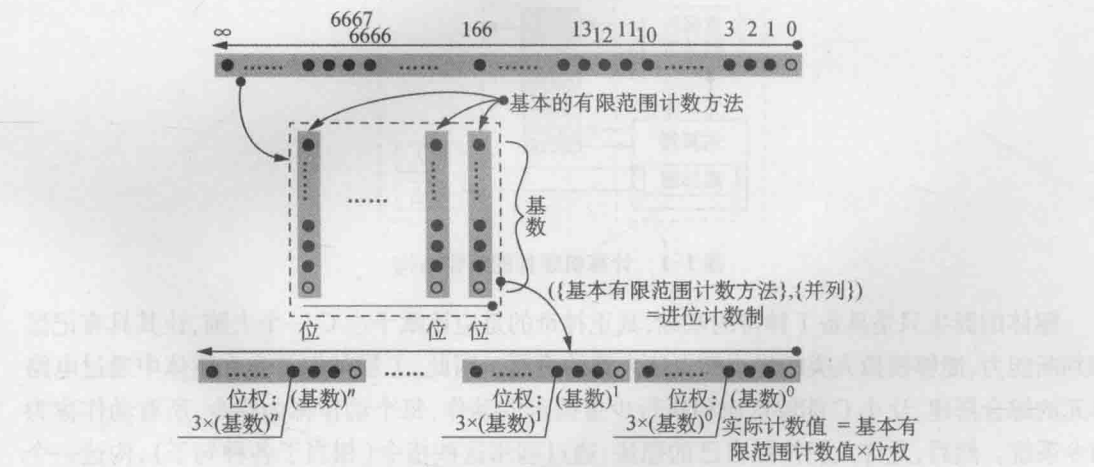
1.2.1 进位计数制

计数是数学的最基本问题,我们从小就开始学数数。人类发展历史中,出现过各种各样的计数方法,例如:打绳结、丢石子或小木棍、刻印记,等等。然而,最终确定和发展了基于进位的计数制,称为进位计数制。

所谓进位计数制,是指采用进位方式的一种计数方法。进位计数制的原理是,首先规定一个基本的有限范围计数方法并定义其表意的直观符号,其计数范围称为基数。例如:对于八进制、十进制和十六进制,其基数和表意直观符号如表1-1所示;其次,通过堆叠(或并列)多个基本的有限范围计数方法,实现无限的计数需求。为了区分每一个堆叠的基本有限范围计数方法,引入位概念(即堆叠的每一个基本有限范围计数方法都占一个位)。并且,每个位都规定一个权值(相当于该位的计数单位量纲。量纲的改变产生进位),称为位权。位权等于 $(\text{基数})^i$ (i 表示位,从0开始,从右向左扩展分别取1,2,⋯,从左向右扩展分别取-1,-2,⋯);最后,通过基数和位权,构建了进位计数制(如图1-2所示)。

表 1-1 各种常用进制的基本计数法表意符号定义

计数法	基数	表意符号
八进制	8	0, 1, 2, 3, 4, 5, 6, 7
十进制	10	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
十六进制	16	0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F



由图 1-2 可知,同样一个基本的有限范围计数值,依据其出现的不同数位,其所表示的

实际计数值是不同的,具体的计数值由该基本的有限范围计数值和其所在位的位权共同决定,即(该基本的有限范围计数值)×该位的位权。例如:十进制数 6666.66 的真实计数值为 $6 \times 10^3 + 6 \times 10^2 + 6 \times 10^1 + 6 \times 10^0 + 6 \times 10^{-1} + 6 \times 10^{-2}$ 。

【例 1-1】 如果 $6C + 5G = C9$, 那么 $9B - 6F = ?$

首先,基于对十六进制计数表意符号设计思想的基本常识,显然该计算式的基数大于 9;其次, $C + G$ 即为 $12 + 16$, 结果为 28。而计算式的个位是 9, 因此,依据进位计数制原理,28 中应该剩余 9, 其他向高位进位。于是, $28 - 9 = 19$, 即满 19 向高位进位;再次, $6 + 5 + 1$ (低位的进位) = 12, 即 C。因此,该计算式是 19 进制计算式。最后, $9B - 6F = 2F$ 。

本质上,进位计数制是一种二维平面型的计数方法,相对于打绳结、丢石子或小木棍、刻印记等一维线性型的计数方法,它通过增加维度提高了灵活性和简洁性,便于快速识别、理解、记忆和使用,从而获得了强大的生命力。

1.2.2 二进制

二进制是进位计数制中基数最小的一种计数方法,正是其简单性,使得它成为神奇宝贝小 C 的基因。也就是说,神奇宝贝小 C 的内部是二进制的世界。首先,各种电路单元都称为“门”电路或数字电路,一个“门”的开(导通)与关(不导通)正好通过 1 和 0 表示,多个“门”的开与关通过 1 和 0 的位串表示,反映相应电路的一种工作状态。指令就是按时间序列(可以先后,也可以同时)完成多个电路状态改变,以实现一种具体的功能。指令本身也是用 1 和 0 的位串表示;其次,人们编写的程序最终都翻译成一条条的指令,用到的数据、文档、图片、声音、动画等,都是通过 1 和 0 的位串表示。因此,二进制成为与神奇宝贝小 C 打交道的基础和关键。

除了具有进位计数制的共同特征和特性外,二进制还有其自身的特点。例如:二进制只有 1 和 0 两个表意直观符号,它们正好相反。因此,只要是能够表示决然不同两种状态的东西,都可以看成是二进制的应用。例如:灯的开与关、木棍的横放与竖放等,因此,在没有电脑的场所,我们也可以做二进制游戏。另外,对于二进制真实计数值,可以将通用的进位计数制表示方法简化为只要权值表示即可。例如: $(1011010.11)_2 = (1 \times 2^6 + 0 \times 2^5 + 1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 + 1 \times 2^{-1} + 1 \times 2^{-2}) = (2^6 + 2^4 + 2^3 + 2^1 + 2^{-1} + 2^{-2})$ 。再者,二进制的一些运算可以简化。例如 $(1 - 0.0000001)_2$, 可以直接用减数的逐位取反即可,即 $(0.1111111)_2$ 。由此可见,二进制是不是很神奇呢?

尽管小 C 内部是二进制世界,然而考虑到有时二进制位串比较长,书写不方便且容易写错(例如少写一个 0 或 1 等),因此,书面表达中经常用八进制(因为 8 是 2^3 , 一个八进制位可以用三个二进制位表示,反之,三个二进制位可以用一个八进制位表示)或十六进制(因为 16 是 2^4 , 一个十六进制位可以用四个二进制位表示,反之,四个二进制位可以用一个十六进制位表示)来表示一个二进制位串。为了区分各种进制的表示,通常以后缀 H 或前缀 0x 或 $()_{16}$ 表示十六进制,以前缀 0 或 $()_8$ 表示八进制。

【例 1-2】 $(5639)_{10} = ()_2$

依据二进制计数原理, $(5639)_{10} = (4096)_{10} + (1024)_{10} + (512)_{10} + (4)_{10} + (2)_{10} + (1)_{10} = 2^{12} + 2^{10} + 2^9 + 2^2 + 2^1 + 2^0 = (1011000000111)_2$

【例 1-3】 十进制小数 13.375 对应的二进制数是()。(NOIP2017)

- A) 1101.011 B) 1011.011 C) 1101.101 D) 1010.01

依据二进制计数原理, $(13.375)_{10} = ((8 + 4 + 1) \cdot (25 + 125))_{10} = ((2^3 + 2^2 + 2^0) \cdot (2^{-2} + 2^{-3}))_{10} = (1101.011)_2$ 。因此, 答案为 A。

【例 1-4】 与十进制数 28.5625 相等的四进制数是()。(NOIP2008)

- A) 123.21 B) 131.22 C) 130.22 D) 130.21
E) 130.20

依据二进制计数原理, $(28.5625)_{10} = ((16 + 8 + 4) \cdot (5 + 0625))_{10} = ((2^4 + 2^3 + 2^2) \cdot (2^{-1} + 2^{-4}))_{10} = (11100.1001)_2 = (01\ 11\ 00.\ 10\ 01)_2 = (130.21)_4$ 。因此, 答案为 D。

【例 1-5】 地址总线的位数决定了 CPU 可直接寻址的内存空间大小, 例如: 地址总线为 16 位, 其最大的可寻址空间为 64 KB。如果地址总线是 32 位, 则理论上最大可寻址的内存空间为()。(NOIP2012)

- A) 128 KB B) 1 MB C) 1 GB D) 4 GB

由于计算机内部是二进制的世界, 因此, 在此的位数是指二进制数的位数。从逻辑上看, 内存空间大小是以字节(Byte, 8 个二进制位)为单位的线性排列, 每个字节有一个地址, 地址从 0 开始编排。内存空间大小的计算量纲如表 1-2 所示(注: 存储设备厂商通常使用 1000 作为进率, 不采用 1024 作为进率)。

16 位地址总线表示用 16 位二进制数来编排地址, 即 0000H ~ FFFFH, 其存储空间大小为: $2^{16}\text{B} = 2^6 * 2^{10}\text{B} = 2^6\text{KB} = 64\text{KB}$ 。因此, 32 位地址总线, 其最大可寻址的内存空间为: $2^{32}\text{B} = 2^2 * 2^{30}\text{B} = 2^2 * 2^{20}\text{KB} = 2^2 * 2^{10}\text{MB} = 2^2\text{GB} = 4\text{GB}$ 。本题答案为 D。

表 1-2 存储空间大小的计算量纲

量纲	单位	换算
字节	B(Byte)	—
千字节	KB(KiloByte)	1 KB = 1024 B = 2^{10} B
兆字节	MB(MegaByte)	1 MB = 2^{10} KB = 1024 KB = 2^{20} B
吉字节	GB(GigaByte)	1 GB = 2^{10} MB = 1024 MB = 2^{30} B
太字节	TB(TeraByte)	1 TB = 2^{10} GB = 1024 GB = 2^{40} B
拍字节	PB(PetaByte)	1 PB = 2^{10} TB = 1024 TB = 2^{50} B
艾字节	EB(ExaByte)	1 EB = 2^{10} PB = 1024 PB = 2^{60} B
泽字节	ZB(ZetaByte)	1 ZB = 2^{10} EB = 1024 EB = 2^{70} B
尧字节	YB(YottaByte)	1 YB = 2^{10} ZB = 1024 ZB = 2^{80} B
B 字节	BB(BrontoByte)	1 BB = 2^{10} YB = 1024 YB = 2^{90} B
N 字节	NB(NonaByte)	1 NB = 2^{10} BB = 1024 BB = 2^{100} B
D 字节	DB(DoggaByte)	1 DB = 2^{10} NB = 1024 NB = 2^{110} B

【例1-6】 将一棵完全二叉树结构(参见图2-19b)按照自上而下的层次顺序映射到一个数组(参见第2.3.1小节)中,则完全二叉树第 k 层第2个结点的左子女和右子女分别位居数组的哪个位置?

二叉树结构具有一些特殊性质,其每层的最大结点数与二进制有着内在的关系,可以看成是二进制的一种典型应用。假设树根结点所在的层次为0,则第 n 层的最大结点数为 2^n 。

完全二叉树是二叉树的一种形态,它只允许最后一层右边可以缺省部分结点。因此,对于完全二叉树第 k 层第2个结点的左子女而言,其前面 $k+1$ 层共有 $(2^0 + 2^1 + 2^2 + \dots + 2^{k-1} + 2^k) = (2^{k+1} - 1)$ 个结点,再加上第 k 层第一个结点的两个子女结点。因此,完全二叉树第 k 层第2个结点的左子女和右子女分别对应数组的第 $2^{k+1} + 1$ 和 $2^{k+1} + 2$ 个元素(C++语言中,数组元素的位置从0开始)。

1.3 如何与神奇宝贝小C进行交流

1.3.1 基本交流方式——人机交互式接口

神奇宝贝小C的躯体——硬件,包含各种功能单元和设备(参见图1-1),其大脑部分——软件,也千姿百态。为了方便与人类交流,或者说方便人类使用,小C雇用了一个大管家——操作系统(一个预先编写好的特殊软件。例如:Windows、Linux、Mac OS、IOS、安卓等),小C与人类的一切交流都通过大管家进行。操作系统一方面帮助小C管理所有的硬件和软件,并向外部提供完成各种功能的接口,称为系统功能调用(简称系统调用);另一方面,面向人类提供两种人机交互的接口:交互式接口(包括命令行界面和图形用户界面)和程序式接口,并且合理安排、组织和管理各种程序的执行。交互式接口用于实时与小C交流(相当于面对面直接交流),程序式接口用于间接地与小C交流(相当于写一个纸条交给小C)。例如:平时人们操作电脑,就属于交互式接口。它可以输入命令,也可以点击鼠标。无论是哪种接口,最终都是通过大管家,将人类的意愿转变为内部相应的一系列系统调用。图1-3所示给出了人机交互的基本原理。

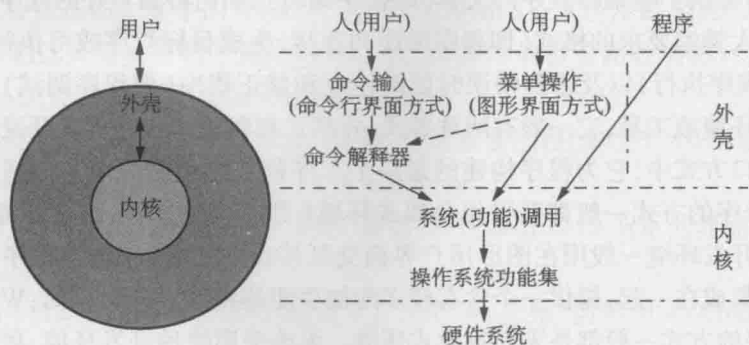


图 1-3 人机交互的基本原理

1.3.2 程序交流方式——程序式接口

作为人机接口的一种高级方式,程序交流方式相对比较复杂,涉及计算机语言、程序开发环境或工具以及程序设计方法。首先,尽管二进制语言是神奇宝贝小C的母语,在其诞生的初期,人类就是通过二进制语言(用一组开关表示0和1的位串)与小C直接交流。然而,人类毕竟习惯于自然语言,希望能用自然语言与小C交流。为此,人类做了各种努力来解决这个问题。尽管小C还不能完全地理解自然语言,但人类发明了多种类似或接近自然语言的计算机语言——程序设计高级语言(简称高级语言。相对而言,将小C的母语称为机器语言或低级语言、二进制语言),例如:PASCAL语言、C语言、C++语言、Python语言等。从语言学的体系结构来看,高级语言与自然语言具有同样的语义层次,如图1-4所示。

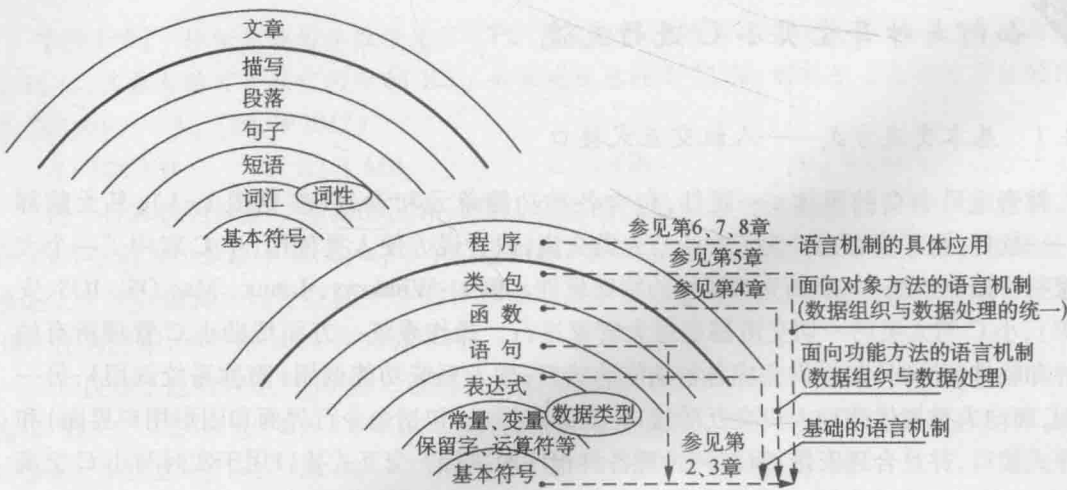


图 1-4 高级语言的语义层次

有了高级语言,人们就可以用它来编写程序或进行程序设计(相当于为小C写一篇文章)。显然,程序的编写要有方法,这由本书后面的章节来逐步详细介绍。在此,首先解决另一个问题,即如何提供一些编写程序的文具(即程序编辑)、如何将编写好的程序翻译成小C的母语及组织成大管家要求的格式(即程序编译和连接,生成目标程序或可执行程序)并交给小C执行(即程序执行)以及遇到错误时如何检查和修正程序(即程序调试)。这个工作就交给程序开发环境或工具,它一般有两种形式:分离式和集成式。分离式开发环境一般用在命令行交互接口方式中,它为程序构建的每一个工序都提供相应的命令。例如:Linux操作系统中,构建程序的方式一般都是采用分离式环境(目前,也提供集成式环境,例如:NOI Linux)。集成式开发环境一般用在图形用户界面交互接口方式中,它将与程序构建每个工序对应的命令都集成在一起,提供一个含有较多附加功能的特殊程序。例如:Windows操作系统中,构建程序的方式一般都是采用集成式环境。无论采用哪种开发环境,程序开发和构建的过程都是一样的,如图1-5所示。

本书主要介绍C++程序设计语言,相应的程序构造工具主要介绍开源的Dev-Cpp。用

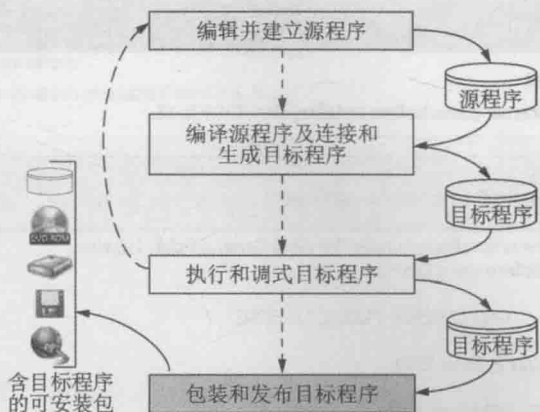


图 1-5 程序开发或和构建的过程

户可以从网上搜索并下载 Dev-Cpp 工具安装程序(例如:Dev-Cpp 5.11 TDM-GCC 4.9.2 Setup. exe),然后,按照下列步骤安装该工具。

(1) 双击下载的 Dev-Cpp 5.11 TDM-GCC 4.9.2 Setup. exe,出现如图 1-6 所示的界面。

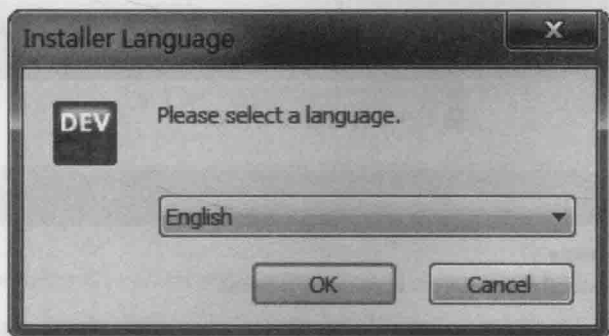


图 1-6 选择语言

(2) 用鼠标点击图 1-6 中的“OK”按钮,出现如图 1-7 所示的界面。

(3) 用鼠标点击图 1-7 中的“I Agree”按钮,出现如图 1-8 所示的界面。

(4) 用鼠标点击图 1-8 中的“Next >”按钮,出现如图 1-9 所示的界面。在此,可以通过“Browse...”按钮选择需要安装工具的具体位置(即具体的目录路径名称。当然,也可以不选择,采用默认的位置 C:\Program Files\Dev-Cpp)。

(5) 用鼠标点击图 1-9 中的“Install”按钮,开始安装过程(如图 1-10 所示。大约需要一分钟左右)。安装完成后,出现如图 1-11 所示的界面。

(6) 在图 1-11 中,用鼠标点击“Finish”按钮,出现如图 1-12 所示的界面,进行相应的配置。

(7) 在图 1-12 中的语言选择区,通过上拉滚动条,选择“简体中文/Chinese”。然后,用鼠标点击“Next”按钮,出现如图 1-13 所示的界面。

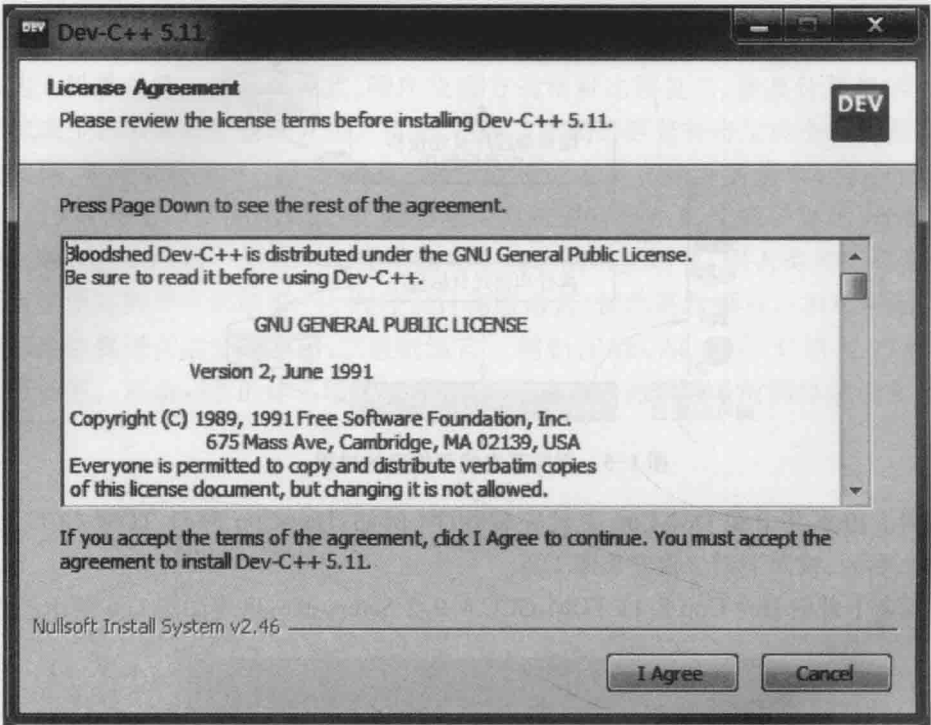


图 1-7 确认版权与协议

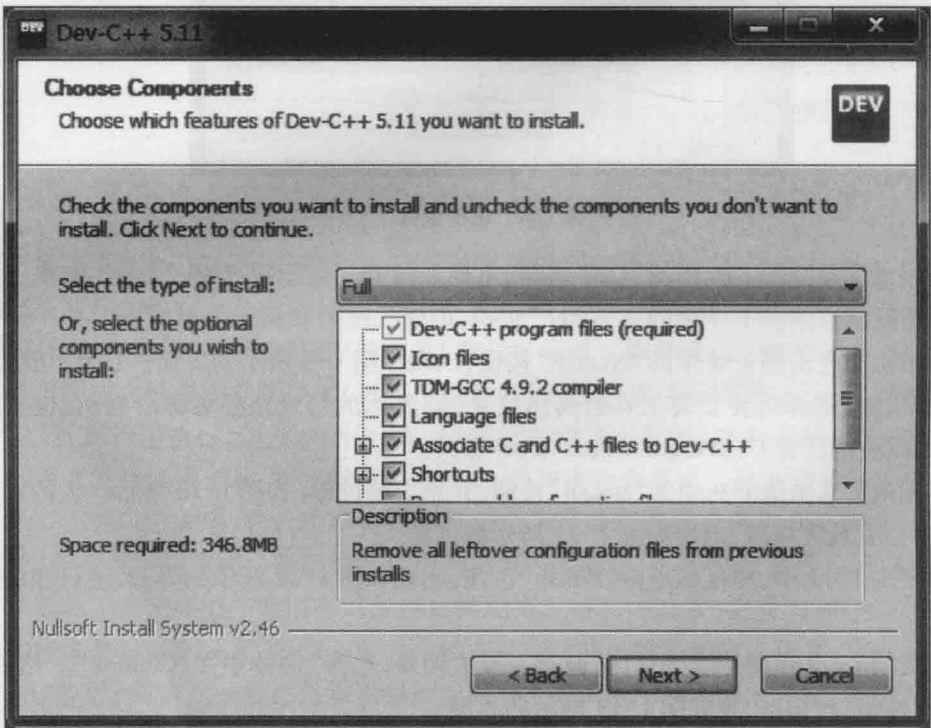


图 1-8 选择需要安装哪些组件