

深入浅出系列丛书

深入浅出 C 语言程序设计

松桥工作室 编著

- ❁ 全面介绍了C语言程序设计的基本概念及多种开发环境下的编程方法，图文并茂，理论与实践结合，使读者轻松掌握。
- ❁ 从C语言的入门到实用项目的开发，内容循序渐进，范例详实，步骤清晰。
- ❁ 随书配有光盘，内附书中范例程序，方便读者学习时使用。



内附范例光盘

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

深入浅出

C 语言程序设计

松桥工作室 编著

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

北京市版权局著作权合同登记 图字：01-2006-0502 号

版 权 声 明

本书中文繁体字版由台湾知城数位科技股份有限公司出版（2005）。本书中文简体字版经台湾知城数位科技股份有限公司授权由中国铁道出版社出版（2006）。任何单位或个人未经出版者书面允许不得以任何手段复制或抄袭本书内容。

图书在版编目（CIP）数据

深入浅出C语言程序设计 / 松桥工作室编著. —北京：
中国铁道出版社，2006.5
（深入浅出）
ISBN 7-113-07076-0

I. 深... II. 松... III. C语言—程序设计
IV. TP312

中国版本图书馆CIP数据核字（2006）第051340号

书 名：深入浅出C语言程序设计

作 者：松桥工作室

出版发行：中国铁道出版社（100054，北京市宣武区右安门西街8号）

策划编辑：严晓舟 郭毅鹏

责任编辑：苏茜 郭毅鹏 王丹

封面设计：薛为

封面制作：白雪

责任校对：张国成

印 刷：北京鑫正大印刷有限公司

开 本：787×1092 1/16 印张：24 字数：563千

版 本：2006年7月第1版 2006年7月第1次印刷

印 数：1~5000册

书 号：ISBN 7-113-07076-0/TP·1823

定 价：39.00元（含盘）

版权所有 侵权必究

凡购买铁道版的图书，如有缺页、倒页、脱页者，请与本社计算机图书批销部调换。

前言

随着计算机硬件技术的不断进步,软件亦随之日新月异。而指挥计算机运行的程序语言,从早期的机器语言、汇编语言到 Fortran、COBOL、Basic、C、PHP、Java、C++和 C#等,一直在不断发展。新程序语言不断地被推出,一些早期的程序语言也不不断地被淘汰,经过近 20 年的发展, Basic 与 C 语言成为目前程序设计入门的两大主流。

C 语言是一个高效率的且介于高级和低级之间的语言,适合开发各种系统或应用程序,可移植性及运行效率高,很受程序员的喜爱。由 C 语言扩展出来的 C++ 是属于面向对象程序语言,很适合用来开发大型应用程序,已成为当今程序设计的主流。UNIX 操作系统就是利用 C 语言所开发出来的,像 Java、PHP、C++、C#以及 Flash MX 2004 的 ActionScript 2.0 等也都是由 C 语言发展而来。由此可知 C 语言的重要性。虽然市面上 C 语言的书籍很多,但不少都已出版多年。本书于近期编写而成,教授初学者基本的程序设计并融入设计程序的思考模式,是一本适合初学者自学或作为教科书的好书。书中的内容以循序渐进的方式,引导初学者由最基本的 C 语言程序设计进而掌握更高级的程序设计技巧。

C 语言程序的开发环境种类很多,为了提高初学者学习效果与减少教师教学上的困扰,本书所有范例均可在 Dev C++ 5.0、Visual C++ 6.0、Borland C++ Builder 6 环境下运行。教师授课或读者自学时只要在书附光盘中切换到 3 种 C 语言项目文件夹之一读取指定的程序文件便可轻松地学习。若初学者没有 C 语言的集成开发环境,本书在书附光盘内提供 Dev C++ 5.0 (Beta 9) 的安装程序“devcpp-4.9.9.2_setup.exe”,可参阅附录 C 学习如何安装。下面是附带光盘的内容说明。

- devsample 文件夹: Dev C++ 5.0 环境开发的 C 程序文件。
- vc6sample 文件夹: Visual C++ 6.0 环境开发的 C 程序文件。
- bcb6sample 文件夹: Borland C++ Builder 6 环境开发的 C 程序文件。
- devcpp-4.9.9.2_setup.exe: Dev C++ 5.0 (Beta 9) 安装程序。

由于本书主要是针对 C 程序设计初学者而编写的,较偏重程序设计能力。限于篇幅,书中难免有疏漏之处,希望读者不吝指正,以求再版时能更趋完美。

本书由知城数位有限公司提供版权,由彭木根、周芸、余艳、刘健、何佩攀,殷勇等同志完成本书的整稿工作,在此表示感谢。

编 者

2006 年 4 月

目 录

第 1 章 C 语言概述	1
1-1 程序语言介绍	2
1-1-1 程序语言的分类	2
1-1-2 编译器的分类	4
1-2 C 语言的发展	4
1-3 C 语言的特点	5
1-4 程序设计的步骤	6
1-5 流程图	9
1-6 习题	9
第 2 章 C 语言集成开发环境介绍	11
2-1 C 语言的程序架构	12
2-1-1 一个简单的 C 程序	12
2-1-2 预处理命令	13
2-1-3 全局变量与自定义函数声明	13
2-1-4 main()主函数	13
2-1-5 声明自定义函数主体区	15
2-2 使用 Visual C++ 6.0	15
2-2-1 添加项目文件	15
2-2-2 程序的编写、保存与运行	17
2-2-3 打开与关闭项目文件	19
2-3 使用 Borland C++ Builder 6	20
2-3-1 添加项目文件	20
2-3-2 程序的编写、保存与运行	23
2-3-3 打开与关闭项目文件	23
2-4 使用 Dev C++ 5.0	25
2-4-1 添加项目文件	26
2-4-2 程序的编写、保存与运行	27
2-4-3 打开与关闭项目文件	29
2-5 使用 Borland C++ 5.0	29
2-5-1 程序的编写、保存与运行	29
2-5-2 打开与关闭 C 语言程序文件	30
2-6 使用 Turbo C 2.01	30
2-6-1 下载安装 Turbo C 2.01	30
2-6-2 程序的编写、保存与运行	32

2-6-3 打开与关闭 C 语言程序文件.....	34
2-7 习题	35
第 3 章 变量与基本数据类型.....	37
3-1 标识符与保留字.....	38
3-1-1 标识符.....	38
3-1-2 保留字.....	38
3-2 数据类型	39
3-2-1 基本数据类型.....	39
3-2-2 扩展数据类型.....	39
3-3 常量	41
3-3-1 字符常量.....	41
3-3-2 字符串常量.....	42
3-3-3 整型常量.....	42
3-3-4 浮点型常量.....	43
3-3-5 符号常量.....	43
3-4 变量	43
3-5 运算符	44
3-5-1 赋值运算符.....	44
3-5-2 算术运算符.....	45
3-5-3 复合赋值运算符.....	45
3-5-4 递增和递减运算符.....	45
3-5-5 关系运算符.....	46
3-5-6 逻辑运算符.....	46
3-6 数据类型转换	48
3-7 习题	49
第 4 章 输入输出介绍	51
4-1 输入/输出函数介绍	52
4-2 printf()输出函数.....	52
4-2-1 printf()函数.....	52
4-2-2 转义字符的输出.....	54
4-2-3 字符变量的输出.....	55
4-2-4 整型变量的输出.....	55
4-2-5 浮点型变量的输出.....	57
4-2-6 字符数组(字符串)的输出.....	57
4-3 scanf()输入函数.....	58
4-3-1 scanf()函数.....	58
4-3-2 字符变量的输入.....	59
4-3-3 整型变量的输入.....	59
4-3-4 浮点型变量的输入.....	60

4-3-5	字符数组（字符串）的输入	61
4-4	字符输入/输出函数	62
4-4-1	getchar()输入字符函数	62
4-4-2	getche()输入字符函数	62
4-4-3	getch()输入字符函数	62
4-4-4	putchar()输出字符函数	63
4-5	字符串输入/输出函数	63
4-5-1	gets()字符串输入函数	63
4-5-2	puts()字符串输出函数	63
4-6	习题	65
第 5 章	选择结构	67
5-1	概述	68
5-2	关系运算符	68
5-3	逻辑运算符	69
5-4	选择语句	70
5-4-1	if-else 语句	70
5-4-2	if-else if 语句	77
5-4-3	嵌套选择结构	79
5-4-4	条件运算符	82
5-5	多重条件选择——switch 语句	84
5-6	实例——综合所得税算法	86
5-7	习题	89
第 6 章	循环结构	95
6-1	概述	96
6-2	for 语句	96
6-2-1	for 循环语句	96
6-2-2	for 语句的用法实例	97
6-2-3	嵌套 for 循环	102
6-3	while 语句	105
6-4	do-while 语句	107
6-5	使用 break 与 continue 语句中止循环	111
6-6	无限循环	113
6-7	实例	116
6-7-1	猜数字游戏	116
6-7-2	统计输入字符种类的个数	119
6-8	习题	121
第 7 章	数组	125
7-1	概述	126
7-2	数组的声明和使用	126

7-2-1	数组的声明	126
7-2-2	如何给数组初始化	127
7-2-3	如何访问数组的数据	128
7-2-4	数组的大小	131
7-3	多维数组	132
7-3-1	二维数组的声明与初始化	132
7-3-2	如何访问二维数组的数据	133
7-4	数组的排序与查找	136
7-4-1	冒泡排序法	136
7-4-2	顺序查找法与二分查找法	139
7-5	实例	144
7-5-1	矩阵相乘	144
7-5-2	大乐透开奖	146
7-5-2	保龄球分数算法	149
7-6	习题	154
第8章	函数	157
8-1	函数简介	158
8-2	内置函数	158
8-3	用户自定义函数的定义和调用	160
8-3-1	如何声明与定义自定义函数	160
8-3-2	如何调用自定义函数	161
8-4	全局变量与局部变量	163
8-5	在函数间传递数组	165
8-6	递归函数	167
8-7	实例	168
8-7-1	数学组合数的求法	168
8-7-2	存款利息算法	170
8-7-3	贷款利息与本金算法	171
8-8	习题	174
第9章	C的预处理器	179
9-1	C语言的编译过程	180
9-2	预处理器	181
9-3	宏替换命令	181
9-3-1	如何定义宏	181
9-3-2	宏定义符号常数	181
9-3-3	宏使用参数	183
9-3-4	宏的副作用	183
9-3-5	宏和函数的差异	184
9-4	自定义头文件	186

9-4-1	包含头文件.....	186
9-4-2	头文件的建立.....	187
9-5	条件编译命令.....	190
9-5-1	如何控制程序的编译流程.....	190
9-5-2	如何判断宏是否定义.....	192
9-6	实例——书籍介绍.....	193
9-7	习题.....	195
第 10 章	变量的存储类型.....	197
10-1	变量存储类型.....	198
10-2	自动变量.....	198
10-3	静态变量.....	198
10-4	外部变量.....	200
10-5	寄存器变量.....	203
10-6	习题.....	203
第 11 章	指针的活用.....	207
11-1	什么是指针.....	208
11-2	指针的声明与使用.....	208
11-2-1	如何声明指针.....	208
11-2-2	如何访问指针变量.....	209
11-2-3	多重指针的使用.....	211
11-2-4	指针的移动.....	212
11-2-5	指针类型的转换.....	213
11-3	传值调用与传址调用.....	214
11-3-1	传值调用.....	214
11-3-2	传址调用.....	215
11-4	指针与数组.....	217
11-5	动态内存分配.....	220
11-6	实例——堆栈.....	223
11-7	习题.....	229
第 12 章	字符串处理.....	233
12-1	字符串.....	234
12-1-1	什么是字符串.....	234
12-1-2	字符串的声明与初始赋值.....	234
12-1-3	如何输入字符串.....	235
12-2	如何声明字符串数组.....	236
12-3	常用字符串函数.....	240
12-3-1	字符串转换函数.....	240
12-3-2	字符串处理函数.....	241
12-4	函数指针.....	247

12-5 实例	249
12-5-1 扑克牌随机发牌	249
12-5-2 验证会员证号码	251
12-6 习题	254
第 13 章 结构自定义数据类型	257
13-1 概述	258
13-2 结构数据类型	258
13-2-1 结构的定义与声明	258
13-2-2 结构字段的初始化	259
13-2-3 结构变量的内存空间大小	262
13-3 嵌套的结构	263
13-4 结构指针	265
13-5 结构数组	267
13-6 结构内字段指针	272
13-7 函数间结构的传递	276
13-7-1 结构变量传值调用	276
13-7-2 结构变量传址调用	278
13-7-3 传递结构数组给函数	281
13-8 实例——多项式相加	285
13-9 习题	291
第 14 章 派生数据类型	293
14-1 枚举数据类型	294
14-2 联合数据类型	298
14-2-1 union 的定义与声明	298
14-2-2 union 和 struct 分配内存的比较	300
14-3 typedef——类型定义	301
14-4 习题	303
第 15 章 文件处理	307
15-1 文件函数简介	308
15-2 数据流的概念	308
15-3 顺序文件与随机文件	309
15-4 C 语言文件处理函数	309
15-5 标准输入/输出函数	310
15-5-1 标准输入/输出函数简介	310
15-5-2 打开文件	312
15-5-3 关闭文件	313
15-5-4 获取文件大小	313
15-5-5 以字符读写文件	315
15-5-6 以字符串读写文件	322

15-5-7 复制数据文件	324
15-5-8 访问格式化文本文件	326
15-6 随机文件	330
15-6-1 随机文件的访问	330
15-6-2 随机文件的修改与删除	334
15-7 系统输入/输出函数	338
15-7-1 系统输入/输出函数简介	338
15-7-2 打开与关闭文件	339
15-7-3 文件的访问	341
15-8 习题	344
附录 A 本书 C 语言主要用字索引	349
附录 B ASCII 码表	353
附录 C C 语言常用函数	355
C-1 转换函数	356
C-2 字符串处理函数	356
C-3 字符判断函数	359
C-4 数学函数	360
C-5 其他常用函数	361
附录 D Dev C++ 的下载与安装	363
D-1 下载 Dev C++	364
D-2 安装 Dev C++	366

1

C 语言概述

- 1-1 程序语言介绍
- 1-2 C 语言的发展
- 1-3 C 语言的特点
- 1-4 程序设计的步骤
- 1-5 流程图
- 1-6 习题

深入浅出

C 语言程序设计

1-1 程序语言介绍

计算机是人类 20 世纪最重要的发明之一。一部计算机是由硬件与软件构成。“硬件”（Hardware）主要负责执行解决问题所必需的基本运算和处理，是由中央处理单元、存储装置、输入输出装置等所构成的有形设备，目前都朝着轻、薄、短、小以及功能强大的趋势发展。“软件”（Software）是用来指挥硬件运作，使其为我们解决问题的命令集合，我们将这些命令的集合称为“程序”（Program）。一般将软件分成“系统软件”和“应用软件”两种。“系统软件”（System Software）是启动计算机各单元运作必备的程序，主要担任硬件与软件间沟通的桥梁（接口），负责系统资源的分配与管理，此类软件大都由计算机制造商研发，提供给用户使用，以方便用户在使用计算机时能更有效率。譬如，操作系统（Operating System）、编译器（Compiler）、解释器（Interpreter）、编辑器（Editor）、连接器（Linker）、载入器（Loader）、调试程序（Debugger）和通用程序（Utility）等均属于系统软件。至于“应用软件”（Application Software），是指必须在系统软件支持下才能执行的程序，是由用户根据应需求而自行设计的程序，以解决特定问题，譬如 Office、DreamWeaver、Flash 等软件包或项目开发的程序均属之。

1-1-1 程序语言的分类

计算机是一部机器，只要给予正确的命令，就会按照命令的指示进行工作，最后再将结果输出，这些命令的组合就是所谓的“程序”，程序必须使用计算机所能识别（看得懂）的程序语言（即机器语言）来编写方可执行。人与人之间的沟通和信息的传达是通过“语言”，同样地，和计算机沟通就需使用计算机了解的称为“程序语言”（Programming Language）的语言，所以程序语言是计算机用户和计算机沟通的桥梁。“程序”是由一连串的“命令”（Instruction）组合而成的，命令有一定的语法和符号，在高级语言中也称为“语句”（Statement）。若使用计算机无法辨识的程序语言来编写程序，必须经过一定程序转换为计算机所能辨识的执行命令，方可在计算机上执行。我们将程序语言转换的过程称为“编译”。所以，程序好比是一出戏的剧本，那么计算机就好比是演员，程序（剧本）来指引计算机（演员）的每一个动作，如此，计算机才会正确地执行指定的工作。一般计算机执行程序的过程，是先通过输入设备将已经编写好的程序加载到主存储器内储存，然后程序中一行行的命令会依次被控制单元提取，经译码或编译成机器看得懂的机器语言才能在计算机中执行。一般程序语言按照难易的程度可分为下列 5 个层次类别。

1. 第一代语言：机器语言（Machine Language）

- 以连续的 0、1 二进制代码来编写程序，由于 0、1 是计算机内部电路唯一能直接识别的语言，不需经过编译就能直接运行，因此运行速度最快。
- 是属于面向机器的语言。不同厂商所设计 CPU 的架构不同，使用的机器语言当然不同。所以，此种语言与机器相依度高，可移植性（不同平台之间程序移植）极低。
- 编写此种程序都是由连续的 0、1 组合而成，费时费力，实用性差且难维护。
- 语法如下。

存储地址	内容 (2 进制)	内容 (16 进制)
1000	1010 0011 0000 0001	A301
1002	0000 0001 1011 0010	01B2
1004	0001 0011 1101 0101	13D5

2. 第二代语言：汇编语言 (Assembly Language)

- 也称低级语言，是使用助记码 (Mnemonic Code)，由字母和数字组合而成，是属于一种符号式语言。譬如， $sum = 10 + 20$ 使用 Intel CPU 所提供汇编语言的写法如下。

```
mov ax,10 ;
add ax,20;
mov sum,ax;
```

- 是属于面向机器的语言。编写此种程序时必须先了解计算机 CPU 内部缓存器和内存的运作才能编写，和计算机硬件的相依性高，不同 CPU，语法是不同的，因此可移植性低。
- 适用于计算机专业人员来编写有关计算机系统或输入输出接口的驱动程序。
- 必须使用汇编程序 (Assembler) 将编写的程序逐行编译成机器语言才能执行。

3. 第三代语言：高级语言 (High-Level Language)

- 语法更接近人类语言与数学表示式，只要对程序稍加修改便可以在不同的计算机系统上执行，可移植性高。
- 属于面向程序的语言，如 Basic (交互式操作环境)、ForTran (工程)、COBOL (商业应用)、Pascal、C 等程序语言均属之，由于这些都是属于传统的高级语言，它们共同的特点就是按照命令的逻辑顺序执行，是属于面向程序的语言 (Procedure-Oriented Language)。
- 其中 C 语言与 Basic 语言是目前的主流。Basic 是最容易为初学者所接受也是最普及的一种高级语言。C 语言则是理工科或程序设计人员用来学习程序设计的利器。

4. 第四代语言：查询语言 (Query Language)

- 是属于非程序语言，以问题为导向，它只描述问题不必叙述解决问题的步骤。
- 此种语言必须事先通过预处理器转换成第三代的程序语言才能编译成可执行码。
- 包括结构化查询语言 SQL (Structural Query Language, 适用于数据库查询)，或 AutoCAD (适用于工程绘图)。

5. 第五代语言：面向对象与自然语言

- 面向对象 (Object Oriented Programming, OOP) 语言，是一种比面向程序更高级的语言，例如，C++ 语言是在 C 中加入面向对象的语法的程序语言。此种语言每个对象拥有自己的属性和方法，由于对象都具有程序重用 (Reused)、继承 (Inheritance)、封装 (Encapsulation)、多态 (Polymorphism) 等特性，使得对象都具有某些小功能，对象与对象之间利用调用可以互传信息或组成一个大程序。
- Java 是由 C++ 改进而来，是 Internet 兴起后流行的语言，它提供相当多的网络功能。它协助 Internet 蓬勃发展的全球信息网 (WWW) 提供多媒体信息服务。为了让多媒体信息能更活泼、更具特色，许多厂商纷纷提供能结合首页设

计的程序语言，这类的程序语言有 Java、VB.NET、C#.NET。

- 自然语言 (Natural Language) 属于人工智能语言，近似人类的语言，是程序语言的终极目标。如 LISP (LIST Processing)、PROLOG (LOGic PROgramming) 等，多用于专家系统。

1-1-2 编译器的分类

随着计算机 CPU 速度的增快、内存容量的增大再配合操作系统 (Operating System, OS) 的快速发展，程序的开发环境也跟着日新月异。高级语言的编译器，由从早期的解释式开发环境、演变成编译式的开发环境，发展至今日的可视化开发环境。由于开发环境朝人性化改进，使得程序设计人员能更有效率的开发程序。

高级的语言翻译器 (Language Translator) 有编译器和解释器两种类型。编译器 (Compiler) 是计算机厂商提供的系统程序之一，其功能是将高级语言所写的程序代码转换成能被机器接受的等效目标程序 (Object Program)。其优点是程序经编译存成目标文件，下次执行时程序若未修改则可马上执行，较节省编译和执行的时间；缺点是编译和连接时间较长而且程序若修改过必须重新编译，程序执行时必须将整个可执行文件一次加载，需要较大的内存空间，程序存盘时也需要较大的辅助储存空间，运行阶段发生错误时出错较难处理。

解释器 (Interpreter) 也是计算机厂商提供的系统程序之一，主要功能是，将高级语言所编写的程序代码依其语句的逻辑顺序逐一转为机器语言命令后执行。其优点是执行时所需要的内存空间和存盘时所需要的磁盘空间较小，而且程序较易调试，适合初学者。缺点是每次执行均须重新编译，执行所需的时间较长，程序若供多人使用时效率较差。

由于视窗操作系统的普及，使得可视化开发环境成为主流。在这种环境下，用户只需要使用鼠标在屏幕上拉一拉、点一点，不用写程序便可轻松地制作出输入输出接口，至于中间处理的过程，依靠程序语言以及更改对象的属性便可设计出需要的应用程序。因此窗口环境下编写的 Windows 窗口应用程序有些是自己编写、有些是系统自动产生，一个程序就有多个相关文件组成。属于可视化开发环境的程序语言包括微软公司的 Visual Basic .NET、Visual C++.NET、Visual C#.NET，Borland 公司的 Borland C++ Builder，Sun 公司的 Java 等。

1-2 C 语言的发展

系统软件在早期都是使用汇编语言来编写，由于汇编语言编写不易且难维护，而且一般的高级语言又难于控制硬件，因此希望能发展出一种能控制硬件的高级语言来取代汇编语言，C 语言便是在这种情形下应运而生的。C 语言的前身可以追溯到 1960 年以解决问题为导向的高级语言 ALGOL 60，该语言当时博得很好的评价，可惜不适合来编写系统软件。1963 年英国剑桥和伦敦大学以 ALGOL 60 为基础，共同推出与硬件有关的 CPL (Combined Programming Language) 语言，由于当时考虑层面过于宽广，造成不方便编写系统软件。1968 年 Martin Richards (世界公认 C 语言的鼻祖) 在英国剑桥简化 CPL 语言而发展出 BCPL (Basic Combined Programming Language)。1970 年 Ken Thompson 于美国 Bell 实验室再度精简 BCPL 语言设计出十分接近硬件的 B 语言，是一种没有类型的语言，使用在当时 DEC 公司 PDP-7 计算机上的一些系统软件，这些软件就是 UNIX 操作系统的前身。

1972~1973 年间 Dennis Ritchie 于美国 Bell 实验室, 为新型 PDP-11 计算机重新改写 UNIX 操作系统, 结合 B 语言和 BCPL 语言重要观念加上数据类型以及一些其他概念发展成 C 语言, 此时 C 语言已能提供整数、浮点数、数组、结构、文件等数据类型。1973 年 K. Thompson 和 D. M. Ritchie 两人合作把 UNIX 的 90% 以上代码用 C 改写 (即 UNIX 第 5 版)。直到 1975 年 UNIX 第 6 版公布后, 才引起普遍的注意。Ritchie 和 Brian Kernighan 于 1978 年出版“The C Programming Language”一书, 因而奠定了 C 语言的完整架构, 我们将此种版本的 C 语言称为 K&R C 语言。随后, C 语言百家争鸣, 产生多种版本的 C 语言, 如 Lattice C、MS-C、Quick C 等, 为了使 C 语言能标准化, 1983 年, 美国国家标准协会 (ANSI) 制定了一套 ANSI 的 C 语言标准。标准化过程达六年之久, 最后于 1989 年 12 月 ANSI 标准终于完成, 1991 年初 ANSI C 第一版出现。

1987 年美国 Borland 公司结合了 K&R C 和 ANSI C 推出 Turbo C, 深受当时程序设计师的喜爱, 随着信息科技的进步, 导致面向对象程序设计的流行, 该公司又于 1989 年陆续推出 Turbo C++、Borland C++, 1999 年 3 月推出 Borland C++ Builder 4, 同年, Microsoft 推出 Visual C++ 6.0。由于 C++ Builder 的整合开发环境比 Visual C++ 更容易学习, 建议 C 语言的初学者使用 C++ Builder 学习。当然, 更高级的应用由 INPRISE (原 Borland) 公司的报告可知, C++ Builder 5 的功能是不逊于 Visual C++ 6.0 的。因此, 目前很多企业的 C 程序设计人员仍偏爱 Borland C++ Builder。INPRISE 公司于 2002 年推出的 Borland C++ Builder 6, 除保留 Builder 5 旧有的功能外, 着重于 Internet 和整合企业数据库, 以及电子商务应用方面。2002 Microsoft 发布 Visual Studio .NET, 使得 Visual Basic .NET、Visual C# .NET 和 Visual C++ .NET 等程序语言又迈进崭新的境界, 更于 2003 年将 Visual C++ .NET 完善得更加容易及更具亲和力。

1-3 C 语言的特点

C 语言诞生的主要目的是为了开发 UNIX 操作系统以替代部份由汇编语言编写的工作, 并让 C 语言所编写出的程序能在不同的环境上开发与执行。C 语言的特色如下。

1. 程序具有块结构 (Block Structure) 以及不严谨的数据类型检查 (Type-Checking)。由于 C 语言的编译器大多由 C 语言编写, 因此具有自编译的特性。
2. 是 UNIX 操作系统所采用的程序语言。C 语言是一种介于低级和高级语言的中级程序语言, 它能像低级语言一样处理位 (bit)、字节 (Byte) 运算, 这是一般高级语言未具有的。所以, 它可调用处理硬件的函数库或自行设计函数库来直接控制硬件, 以提升硬件的运行速度。另一方面 C 语言可用来发展高级软件接口。譬如, C 语言像一般的高级语言一样提供功能强大的函数库, 使得编写程序变得很容易, 不但可读性高而且容易维护和调试。所以 C 语言具有高级架构和低级功能。
3. 是一种可移植性的系统程序发展语言。由于 C 语言将和机器依赖度高的输入/输出部分独立出来制成一个函数库, 所以, 在编写 C 语言程序时, 只要语法符合美国国家标准局 ANSI 标准, 并且所使用的函数都是 ANSI 标准程序, 再把 I/O 部分的程序代码稍加改写, 就可轻易将 C 的程序转移到其他不同操作系统的计算机上运行。因此, C 语言具有可移植性, 很容易适应各种计算机编写应用程序或操作系统。譬如, 可以很容易将 DOS 下编写的程序转换 (移植) 在 Windows 下运行。

所以 C 语言具有可移植性及高跨平台功能。

4. 结构化程序设计流程的特色就是将程序代码和数据划分, 使用局部变量达到程序区块化, 并提供顺序、选择以及循环 3 种结构。通过这 3 种结构的组合便可编写出具有模块结构化的程序。由许多小程序(模块)组合成大程序。
5. C 语言所写的程序都是由函数构成, 和一般高级语言不同的是, C 语言不将这些函数当成语句, 而是将函数按照性质存于不同函数库, 程序需要时才连接到程序中。由于函数库不附带在编译器中, 使得 C 语言显得特别精简。
6. C 语言提供指针及地址运算能力。通过指针可访问内存地址, 若 CPU 采用内存和 I/O 共享地址方式, 可以使用 C 语言编写硬件控制程序来控制外围的 I/O 装置。参数传递也可使用指针方式。
7. C 语言允许使用动态数据结构, 程序需要使用到此种结构时才配置内存给它使用, 通过指针可指到所配置的内存地址。

1-4 程序设计的步骤

计算机是协助人类解决问题的工具, 因此, 程序设计者必须先要充分了解问题才能编写出合适的应用程序。一般程序设计的步骤可分为下面五大阶段。

一、问题定义

对问题一定要有充分的研究与分析, 以了解该问题是否适合计算机来处理, 如此才能明确地定义出要解决问题的方法。描述问题时避免使用含义不清的语句, 以方便程序设计人员对问题的语句有深入的了解。

二、问题分析

先认清问题的症结, 对症下药才能解决问题, 对现有的信息加以整理, 再根据输出格式的需求找出需要输入哪些数据, 并明确各种输入输出的限制。

三、设计算法

根据问题的输入输出需求, 详细规划解决问题的步骤。在这个阶段, 不需要考虑使用何种程序语言来编写。若是简单的问题, 绘制流程图即可, 至于较复杂的问题则采用伪码方式来描述。若能有其他算法, 试着比较两者, 选出一个较佳的算法。一个好的算法应具备下列五大要点。

- ① 有限性: 算法必须在有限个步骤内解决问题。
- ② 确定性: 算法中的每个步骤都必须很清楚地表达出来。
- ③ 可行性: 必须在有限的时间内完成。
- ④ 输入: 包含零个或一个以上的输入数据。
- ⑤ 输出: 至少产生一个输出结果。

四、编写程序

根据选定的算法, 选择合适的程序语言, 再依照算法步骤来编写程序, 同时尽量以模