

深入浅出系列丛书

深入浅出

C & C++

程序设计

松桥工作室 编著

本书特色：

- ❶ 本书全面介绍C&C++程序设计语言的基本概念与编程方法，图文并茂，理论与实践完美结合。
- ❷ 从C&C++入门操作到实用项目的开发，内容循序渐进，范例详实，步骤清楚。
- ❸ 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&C++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 01-2006-1088 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&C++程序设计/松桥工作室编著. —北京:
中国铁道出版社, 2006
(深入浅出系列)
ISBN 7-113-07172-4

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 061406 号

书 名: 深入浅出 C&C++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 王慧亮

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: 30.5 字数: 735 千

版 本: 2006 年 7 月第 1 版 2006 年 7 月第 1 次印刷

印 数: 1~5 000 册

书 号: ISBN 7-113-07172-4/TP·1896

定 价: 48.00 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。

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	10
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原型声明	13
2-1-4 main()主函数	14
2-1-5 定义自定义函数主体区	15
2-2 使用 Dev C++ 5.0.....	15
2-2-1 添加项目文件	16
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 Visual C++ 6.0.....	19
2-3-1 添加项目文件	19
2-3-2 程序的编写、保存与运行	21
2-3-3 打开与关闭项目文件	22
2-4 使用 Borland C++ Builder 6.0.....	23
2-4-1 添加项目文件	23
2-4-2 程序的编写、保存与运行	25
2-4-3 打开与关闭项目文件	26
2-5 使用 Borland C++ 5.0.....	28
2-5-1 程序的编写、保存与运行	28
2-5-2 打开与关闭 C 语言程序文件.....	29
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	31

2-6-3 打开与关闭 C 语言程序文件.....	33
2-7 习题.....	33
第 3 章 变量与基本数据类型.....	35
3-1 标识符与保留字.....	36
3-1-1 标识符.....	36
3-1-2 保留字.....	36
3-2 数据类型.....	37
3-2-1 基本数据类型.....	37
3-2-2 扩展数据类型.....	38
3-3 常量.....	39
3-3-1 字符常量.....	39
3-3-2 字符串常量.....	41
3-3-3 整型常量.....	41
3-3-4 浮点型常量.....	41
3-3-5 符号常量.....	41
3-4 变量.....	41
3-5 运算符.....	42
3-5-1 赋值运算符.....	43
3-5-2 算术运算符.....	43
3-5-3 复合赋值运算符.....	43
3-5-4 递增和递减运算符.....	44
3-5-5 关系运算符.....	44
3-5-6 逻辑运算符.....	45
3-6 数据类型转换.....	46
3-7 习题.....	47
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	62
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	64
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	76
5-4-3	嵌套选择结构	78
5-4-4	条件运算符	82
5-5	多重条件选择 switch 语句	83
5-6	实例——综合所得税算法	86
5-7	习题	88
第 6 章	循环结构	93
6-1	概述	94
6-2	for 循环	94
6-2-1	for 的循环语句	94
6-2-2	for 语句的用法范例	96
6-2-3	嵌套 for 循环	100
6-3	while 语句	103
6-4	do...while 语句	105
6-5	使用 break 与 continue 语句退出循环	109
6-6	无限循环	111
6-7	实例	113
6-7-1	猜数字游戏	113
6-7-2	统计输入字符种类的个数	116
6-8	习题	118
第 7 章	数 组	121
7-1	概述	122
7-2	数组的声明及使用	122

7-2-1	数组的声明	122
7-2-2	如何初始化数组	123
7-2-3	如何访问数组的数据	123
7-2-4	数组的大小	127
7-3	多维数组	128
7-3-1	二维数组的声明与初始化	128
7-3-2	如何访问二维数组的数据	129
7-4	数组的排序与查找	132
7-4-1	冒泡排序法	132
7-4-2	顺序查找法与二分查找法	134
7-5	实例	139
7-5-1	矩阵相乘	139
7-5-2	大乐透开奖	141
7-5-3	保龄球分数计算	144
7-6	习题	148
第8章	函 数	151
8-1	函数概述	152
8-2	内置函数	152
8-3	用户自定义函数的定义和调用	154
8-3-1	如何声明与定义自定义函数	154
8-3-2	如何调用自定义函数	155
8-4	全局变量与局部变量	157
8-5	在函数间传递数组	159
8-6	递归函数	161
8-7	实例	162
8-7-1	数学组合公式求法	162
8-7-2	存款利息算法	164
8-7-3	贷款利息与本金算法	165
8-8	习题	168
第9章	C的预处理器	173
9-1	C语言的编译过程	174
9-2	预处理器	175
9-3	宏替换命令	175
9-3-1	如何定义宏	175
9-3-2	宏定义符号常数	175
9-3-3	宏使用参数	177
9-3-4	宏的副作用	177
9-3-5	宏和函数的差异	178
9-4	自定义头文件	180

9-4-1	包含头文件	180
9-4-2	头文件的建立	181
9-5	条件编译命令	184
9-5-1	如何控制程序的编译流程	184
9-5-2	如何判断宏是否定义	186
9-6	实例——书籍介绍	187
9-7	习题	189
第 10 章	变量的存储类型	191
10-1	变量存储类型	192
10-2	自动变量	192
10-3	静态变量	192
10-4	外部变量	194
10-5	寄存器变量	197
10-6	习题	197
第 11 章	指针的活用	201
11-1	什么是指针	202
11-2	指针的声明与使用	202
11-2-1	如何声明指针	202
11-2-2	如何访问指针变量	203
11-2-3	多重指针的使用	205
11-2-4	指针的移动	206
11-2-5	指针类型的转换	207
11-3	传值调用与传址调用	208
11-3-1	传值调用	208
11-3-2	传址调用	209
11-4	指针与数组	211
11-5	动态内存分配	213
11-6	实例——堆栈	216
11-7	习题	221
第 12 章	字符串处理	223
12-1	字符串	224
12-1-1	什么是字符串	224
12-1-2	字符串的声明与初始化	224
12-1-3	如何输入字符串	225
12-2	如何声明字符串数组	226
12-3	常用字符串函数介绍	230
12-3-1	字符串转换函数	230
12-3-2	字符串处理函数	230
12-4	函数指针	236

12-5 实例	238
12-5-1 扑克牌随机数发牌	238
12-5-2 验证会员证号码	240
12-6 习题	243
第 13 章 结构自定义数据类型	245
13-1 概述	246
13-2 结构数据类型	246
13-2-1 结构的定义与声明	246
13-2-2 结构字段的初始化	247
13-2-3 结构变量的内存空间大小	250
13-3 嵌套的结构	251
13-4 结构指针	253
13-5 结构数组	255
13-6 结构内字段指针	259
13-7 函数间结构的传递	263
13-7-1 结构变量传值调用	263
13-7-2 结构变量传址调用	265
13-7-3 传递结构数组给函数	267
13-8 实例——多项式相加	271
13-9 习题	277
第 14 章 派生数据类型	279
14-1 枚举数据类型	280
14-2 联合数据类型	284
14-2-1 union 的定义与声明	285
14-2-2 union 和 struct 分配内存比较	286
14-3 typedef 类型定义	287
14-4 习题	289
第 15 章 文件处理	293
15-1 文件函数概述	294
15-2 数据流的概念	294
15-3 顺序文件与随机文件	295
15-4 C 语言文件处理函数	295
15-5 标准输入/输出函数	296
15-5-1 标准输入/输出函数简介	296
15-5-2 如何打开文件	298
15-5-3 如何关闭文件	299
15-5-4 如何获取文件大小	300
15-5-5 如何以字符读写文件	301
15-5-6 如何以字符串读写文件	307

15-5-7	如何复制数据文件	309
15-5-8	如何访问格式化文本文件	311
15-6	随机文件	316
15-6-1	随机文件的访问	316
15-6-2	随机文件的修改与删除	319
15-7	系统输入/输出函数	323
15-7-1	系统输入/输出函数简介	323
15-7-2	打开与关闭文件	324
15-7-3	文件的访问	325
15-8	习题	329
第 16 章	C++集成开发环境介绍	333
16-1	C++的程序架构	334
16-2	使用 Dev C++ 5.0	337
16-2-1	添加项目文件	337
16-2-2	程序的编写、保存与运行	339
16-2-3	打开与关闭项目文件	340
16-3	使用 Visual C++ 6.0	341
16-3-1	添加项目文件	341
16-3-2	程序的编写、保存与运行	343
16-3-3	打开与关闭项目文件	344
16-4	使用 Borland C++ Builder 6.0	345
16-4-1	添加项目文件	346
16-4-2	程序的编写、保存与运行	347
16-4-3	打开与关闭项目文件	347
16-5	习题	350
第 17 章	C++格式化输入/输出	353
17-1	C++输入/输出概述	354
17-2	cout 输出对象	354
17-2-1	cout 对象的使用	354
17-2-2	cout 对象的输出方式	355
17-3	cin 输入对象	356
17-3-1	cin 对象的使用	356
17-3-2	cin.getline()成员函数	357
17-4	格式化输出	359
17-4-1	setf()、unsetf()成员函数及格式化标记的使用	359
17-4-2	width()、precision()、fill()函数的使用	362
17-4-3	I/O 控制器的使用	364
17-5	习题	367

第 18 章 C 与 C++ 的区别	371
18-1 C 与 C++ 的区别	372
18-2 布尔数据类型	373
18-3 重载函数	373
18-4 inline 函数	375
18-5 引用变量	377
18-5-1 引用变量的使用	377
18-5-2 引用调用	378
18-6 string 类	380
18-6-1 使用 string 类建立字符串对象	380
18-6-2 string 类的运算符	381
18-6-3 string 类常用的成员函数（方法）	382
18-7 习题	385
第 19 章 对象与类	389
19-1 概述	390
19-2 面向对象程序设计的概述	390
19-2-1 什么是面向对象程序设计	390
19-2-2 什么是对象（Object）	390
19-2-3 什么是类（Class）	391
19-2-4 使用面向对象程序设计的好处	391
19-2-5 抽象化	392
19-3 如何定义类与建立对象实例	393
19-3-1 如何定义类	393
19-3-2 如何产生对象实例	394
19-4 数据封装	395
19-4-1 数据成员和成员函数	395
19-4-2 :: 全局范围解析操作符	400
19-4-3 重载成员函数	402
19-4-4 对象指针	403
19-5 构造函数与析构函数	403
19-6 只读与只写属性	407
19-7 静态成员	408
19-8 使用 this 指向目前的类成员	410
19-9 对象数组	411
19-10 函数间对象的传递	414
19-11 C++ 动态分配内存	416
19-12 习题	419
第 20 章 继承与多态	423
20-1 继承	424

20-1-1 程序设计中的继承	424
20-1-2 继承的实现	425
20-1-3 多重继承	427
20-1-4 覆盖成员函数	431
20-2 虚函数	433
20-3 多态	436
20-4 重载运算符	437
20-5 命名空间	441
20-6 习题	443
第 21 章 模 板	445
21-1 模板概述	446
21-2 模板函数	446
21-3 模板类	450
21-4 习题	453
附录 A ASCII 表	455
附录 B C 语言常用函数	457
B-1 转换函数	458
B-2 字符串处理函数	458
B-3 字符判断函数	461
B-4 数学函数	462
B-5 其他常用函数	463
附录 C Dev C++ 下载与安装	465
C-1 下载 Dev C++	466
C-2 安装 Dev C++	468

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

1-1 程序语言介绍

计算机是 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、存储设备、输入/输出设备等所构成的有形设备，目前已向轻、薄、短、小等方面发展。“软件”（Software）用来指挥硬件运作，是解决问题的命令集合，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。所谓“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，能更有效地达到其要求，例如：操作系统（Operating System）、编译器（Compiler）、解释器（Interpreter）、编辑器（Editor）、连接器（Linker）、加载器（Loader）、调试程序（Debugger）和通用程序（Utility）等。“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，是由用户根据需求而自行设计的程序，以解决特定问题，例如：Office、DreamWeaver、Flash 等软件包或项目开发的程序。

1-1-1 程序语言的分类

计算机是一台机器，只要给予正确的命令，就会按照命令的指示来执行工作，最后再将结果输出，这些命令的组合称为“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。人与人之间的沟通和信息的传达是通过“语言”，同样，与计算机沟通就需要使用计算机了解的语言，称为“程序语言”（Programming Language），所以程序语言是计算机用户和计算机沟通的桥梁。至于“程序”就是由一连串的“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）。若使用计算机无法识别的程序语言来编写程序，必须将程序转换为计算机所能识别的执行命令，才能在计算机上运行。将程序语言转换的过程称为“编译”。所以，程序就像是一出戏的剧本，那么计算机就像是演员，程序（剧本）来指引计算机（演员）的每一个动作，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存储器内存储，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执行。一般程序语言按照难易的程度可分为如下 5 种。

1. 第一代语言：机器语言（Machine Language）

（1）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，不需要经过编译就能直接运行，因此运行速度最快。

（2）是属于面向机器的语言。与厂商所设计 CPU 的架构不同，因此使用的机器语言也不同。所以，此种语言与机器相结合的紧密度高，可移植性（不同平台之间程序移植）极低。

（3）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。

（4）语法：

内存地址	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）

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

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

(2) 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，与计算机硬件相结合的紧密度高，不同 CPU，语法是不同的，因此可移植性低。

(3) 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。

(4) 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）

(1) 语法更接近人类语言与数学表示式，只要对程序稍加修改即可在不同的计算机系统中执行，可移植性高。

(2) 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，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）。

(3) 其中 C 语言与 BASIC 语言是目前的主流。其中 BASIC 是最容易为初学者所接受也是最普及的一种高级语言。至于 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）

(1) 是属于非程序语言，它只描述问题但不叙述解决问题的步骤。

(2) 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。

(3) 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

(1) 面向对象（Object Oriented Programming: OOP）语言，是一种比面向程序更高级的语言，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，由于对象具有程序重用（Reused）、继承（Inheritance）、封装（Encapsulation）、多态（Polymorphism）等特性，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。

(2) Java 是由 C++改进而来，是因特网兴起后流行的语言，它提供相当多的网络功能。它是随着因特网蓬勃发展的万维网（WWW），提供超媒体的信息服务，为了让超媒体信息能更活泼、更具特色，许多厂商纷纷提供能结合首页设计的程序语言，如：Java、

VB.NET、C#.NET。

(3) 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), 多用于专家系统。

1-1-2 编译器的分类

随着计算机硬件 CPU 速度的增快、内存容量的增大和操作系统 (Operating System:OS) 的快速发展, 程序的开发环境也有很大改善。高级语言的编译器, 由早期的解释式开发环境、演变成编译式的开发环境, 如今已发展为可视化开发环境。由于开发环境更加人性化, 使得程序设计人员能更有效地开发程序。

高级的语言翻译器 (Language Translator) 分为编译器和解释器两种。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转换成目标文件, 下次执行时程序若不需要修改即可马上执行, 可以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需要修改必须重新编译, 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存储空间、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, 根据其语句的逻辑顺序, 将命令逐一转换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。缺点是每次执行都需要重新编译, 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拖曳, 不用写程序便可轻易的制作出输入/输出接口, 至于中间处理的过程就必须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。因此在可视化开发环境下编写的 Windows 窗口应用程序有些是自己编写、有些是系统自动产生。所以, 一个程序由多个相关文件组成。属于可视化开发环境这类的程序语言包括: 微软的 Visual Basic .NET、Visual C++.NET、Visual C#.NET、Borland 公司的 Borland C++ Builder、Sun 的 Java 等。

1-2 C 语言的发展

早期系统软件都是使用汇编语言来编写的, 由于汇编语言编写不易和难维护, 而且一般的高级语言又难于控制硬件, 因此需要开发出一种能控制硬件的高级语言来取代汇编语言, C 语言就是在这种情况下应运而生。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, 当时得到很好的评价, 可惜不适合编写系统软件。在 1963 年英国剑桥和伦敦大学以 ALGOL60 为基础, 共同推出与硬件有关 CPL (Combined Programming Language) 语言, 由于当时考虑层面过宽, 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) 在英格兰剑桥简化 CPL 语言而发展出 BCPL (Basic Combined Programming Language)。1970 年 Ken Thompson 在美国 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 在美国 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, 才引起普遍的关注。同时 Ritchie 和 Brian Kernighan 在 1978 年出版《The C Programming Language》一书, 因此奠定了 C 语言的完整架构, 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, 产生出多种版本的 C 语言, 如: Lattice C、MS-C、Quick C 等, 为了使 C 语言能标准化, 在 1983 年夏天, 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, 最后在 1989 年 12 月 ANSI 标准终于完成, 1991 年初 ANSI C 第 1 版终于出现。

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, 该公司又在 1989 年陆续推出 Turbo C++、Borland C++, 1999 年 3 月推出了 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, C 语言的初学者建议使用 C++ Builder 较易学习。当然更高级的应用, 由 INPRISE (原 Borland) 公司的报告可得知, C++ Builder 5 的功能是不逊于 Visual C++ 6.0 的。因此, 目前很多企业里的 C 程序设计人员仍偏爱 Borland C++ Builder。INPRISE 公司在 2002 年推出的 Borland C++ Builder 6.0, 除保有 Builder 5.0 已有的功能外, 着重在 Internet 和整合企业数据库上, 以及应用到电子商务方面。2002 Microsoft 发布 Visual Studio .NET, 使得 Visual Basic .NET、Visual C# .NET 和 Visual C++.NET 等程序语言又得到进一步的改进, 在 2003 年将 Visual C++.NET 的使用变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所编写的工作, 并让 C 语言所编写出的程序能在不同的环境上开发系统与执行。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 语言所编写, 因此具有自编译的特性。

(2) 为 UNIX 操作系统所采用的程序语言。它是一种介于低级语言和高级语言的中级程序语言, 表示 C 语言一方面能像低级语言一样处理位 (Bit)、字节 (Byte) 运算, 这是一般高级语言所不具有的。所以, 它可调用处理硬件的函数库或自行设计需要函数库来直接控制硬件, 以提高硬件的运行速度。另一方面 C 语言可用来发展高级软件接口。例如 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) 为一种可移植性的系统程序发展语言。由于 C 语言将和机器相结合紧密度高的输入/输出部分独立出来制成一个函数库, 所以, 在编写 C 语言的程序时, 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 的标准程序, 再把 I/O 部分的程序代码稍加改写, 即可轻易将 C 语言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的应用程序或操作系统。例如: 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。