



普通高等教育“十二五”重点规划教材 计算机基础教育系列

C语言程序设计

付明柏◎主 编



科学出版社

普通高等教育“十二五”重点规划教材 计算机基础教育系列

C 语言程序设计

付明柏 主编

科学出版社

北 京

内 容 简 介

本书共 12 章, 内容包括: C 语言的基本概念和基本语法, 基本数据类型、运算符及表达式, 算法和基本程序设计, 选择分支结构程序设计, 循环结构程序设计, 数组, 函数, 结构体与共用体, 指针, 编译预处理和位运算, 文件, 并给出了一个小型超市管理系统的综合实例, 还介绍了 C 语言图形程序设计的基本方法。

本书注重应用性和实践性, 内容全面、专业, 讲解细致入微, 通过一些经典的实例程序给读者一些解题示范和启发。

本书既适合于教学, 又适合于自学, 可作为高等院校各专业 C 语言程序设计课程的教材, 也可供准备参加计算机等级考试和考研的读者阅读参考, 同时也可作为工程技术人员和计算机爱好者的参考用书。

图书在版编目 (CIP) 数据

C 语言程序设计/付明柏主编. —北京: 科学出版社, 2013
(普通高等教育“十二五”重点规划教材·计算机基础教育系列)
ISBN 978-7-03-037280-2

I. ①C… II. ①付… III. ① C 语言-程序设计-高等学校-教材
IV. ①TP312

中国版本图书馆 CIP 数据核字 (2013) 第 071619 号

责任编辑: 陈晓萍 / 责任校对: 柏连海

责任印制: 吕春珉 / 封面设计: 北大彩印

科 学 出 版 社 出 版

北京东黄城根北街16号

邮政编码: 100717

<http://www.sciencep.com>

北京鑫丰华彩印有限公司印刷

科学出版社发行 各地新华书店经销

*

2013 年 9 月 第 一 版 开本: 787×1092 1/16

2013 年 9 月第一次印刷 印张: 23 1/2

字数: 550 000

定价: 42.00 元

(如有印装质量问题, 我社负责调换〈鑫丰华〉)

销售部电话 010-62142126 编辑部电话 010-62138978-2009

版权所有, 侵权必究

举报电话: 010-64030229; 010-64034315; 13501151303

本书编写人员

主 编 付明柏

副主编 付在琦 宋昭寿 侯 波

参 编 (按姓氏拼音排序)

胡忠旭 姜 娜 冉 婕 申云成

杨雪松 张朝鑫 赵 伦

前 言

C 语言是目前国际上广泛流行的一种结构化程序设计语言,不仅适合于开发系统软件,而且也是开发应用软件和进行大规模科学计算的常用程序设计语言,因此成为程序设计语言的常青树。

本书第 1、2 章介绍了 C 语言的特点和 C 语言的基本概念;第 3~5 章介绍了算法的描述和结构化程序设计的基本方法及三种基本结构(顺序结构、选择结构和循环结构);第 7 章介绍了函数的定义和使用;第 6、8、9 章介绍了数据的构造类型(包括数组、字符串、结构体和共同体等)和指针类型;第 10、11 章介绍了编译预处理和文件的概念,并给出了一个小型超市管理系统的综合实例;第 12 章介绍了 C 语言图形程序设计的基本方法。本书的目标是让读者深刻理解和掌握 C 语言的语法规则及编程特点,克服“懂 C 语言,但不会做题、不会编程序”的通病。

本书选编了较多的例子及习题,教师可在教学中根据需要进行选择,也可供读者自学提高。C 语言是一门实践性很强的课程,读者通过多读例子和动手上机编程,可开阔思路,训练编程和调试程序的能力。为此,在附录中还分别介绍了 Visual C++、Win-TC 等目前常用的 C 语言集成开发环境。本书中的例题均经过 Turbo C 2.0 运行环境的调试。

本书力求内容的完整性,部分内容教师可根据实际情况安排学生自学,其实际讲授学时数应不低于 72 学时,其中实验课占 24 学时。学习完本书后,建议安排两周的“课程设计”,以完成一个小型应用系统的设计与实现。有需求的读者可向 fumingbai@163.com 或 cxp666@yeah.net 索取与本书配套的电子课件和习题答案。

本书由付明柏副教授任主编。作者根据课程要求,结合 C 语言程序设计的教学经验,并参考了大量相关同类教材,对多年来的讲稿进行了精心的总结、修改和整理,编成本书。在编写过程中,昭通学院计算机科学系的全体教师提出了许多宝贵意见和建议,本书的出版得到了编者所在学校的大力支持,在此一并向他们表示衷心的感谢。

本书可作为高等院校 C 语言程序设计课程的教材,也可供准备参加计算机等级考试和考研的读者阅读参考,同时也可作为工程技术人员和计算机爱好者的参考用书。

由于作者水平有限,书中难免存在缺点和错误,恳请广大读者批评指正。

付明柏

2013 年 3 月

于昭通学院

目 录

第 1 章 概述	1
1.1 程序设计语言	1
1.1.1 计算机语言	1
1.1.2 程序设计语言的发展	1
1.2 C 程序设计语言	2
1.2.1 C 语言的发展历史	2
1.2.2 C 语言的标准与版本	3
1.2.3 C 语言的特点	3
1.3 C 语言的基本语法成分	4
1.3.1 字符集	4
1.3.2 标识符	4
1.3.3 运算符	6
1.4 C 语言程序的组成	6
1.4.1 简单的 C 语言程序介绍	6
1.4.2 C 语言程序的结构	8
1.4.3 C 语言程序的书写	9
1.5 C 语言程序开发过程	9
1.5.1 C 语言程序的上机步骤	9
1.5.2 Turbo C 2.0 简介	10
1.5.3 C 语言程序的基本调试技术	14
1.6 总结与提高	15
1.6.1 小结	15
1.6.2 典型题例	16
习题 1	16
第 2 章 基本数据类型、运算符及表达式	18
2.1 C 语言的基本数据类型	18
2.1.1 数据类型	18
2.1.2 C 语言数据类型简介	18
2.1.3 C 语言的基本数据类型	19
2.2 常量	19
2.2.1 整型常量	20
2.2.2 实型常量	21
2.2.3 符号常量	21
2.2.4 字符型常量	22

2.3	变量	23
2.3.1	变量名	23
2.3.2	变量类型	24
2.3.3	变量值	24
2.4	运算符及表达式	24
2.4.1	C 运算符简介	24
2.4.2	算术运算符与算术表达式	26
2.4.3	赋值运算符与赋值表达式	28
2.4.4	逗号运算符和逗号表达式	30
2.4.5	不同类型数据间的混合运算与类型转换	30
2.5	总结与提高	31
2.5.1	小结	31
2.5.2	典型题例	32
	习题 2	33
第 3 章	算法和基本程序设计	36
3.1	算法	36
3.1.1	算法的概念	36
3.1.2	算法的评价标准	37
3.1.3	算法的表示	37
3.2	结构化程序设计的方法	39
3.3	程序的基本结构	40
3.3.1	顺序结构	40
3.3.2	选择结构	40
3.3.3	循环结构	41
3.4	顺序结构程序设计	42
3.5	数据的输入与输出	43
3.5.1	C 语言中数据的输入与输出	43
3.5.2	字符数据的输入与输出	44
3.5.3	格式的输入与输出	46
3.6	总结与提高	51
3.6.1	小结	51
3.6.2	典型题例	51
	习题 3	52
第 4 章	选择分支结构程序设计	56
4.1	关系运算	56
4.1.1	关系运算符	56
4.1.2	关系表达式	56
4.2	逻辑运算	57
4.2.1	逻辑运算符	57

4.2.2 逻辑表达式	58
4.3 二路分支——if 语句	59
4.3.1 if 语句的一般形式	59
4.3.2 if 语句的嵌套	66
4.4 条件运算符与表达式	68
4.5 多路分支——switch 语句	69
4.5.1 switch 语句的一般形式	70
4.5.2 switch 语句的嵌套	71
4.6 选择分支结构程序举例	72
4.7 总结与提高	75
4.7.1 小结	75
4.7.2 典型题例	75
习题 4	77
第 5 章 循环结构程序设计	83
5.1 while 语句	83
5.2 do-while 语句	85
5.3 for 语句	86
5.4 goto 语句	89
5.5 循环的嵌套	90
5.6 break 和 continue 语句	92
5.6.1 continue 语句	92
5.6.2 break 语句	93
5.7 总结与提高	93
5.7.1 小结	93
5.7.2 典型题例	94
习题 5	98
第 6 章 数组	103
6.1 一维数组	103
6.1.1 一维数组的定义和初始化	103
6.1.2 一维数组元素的引用	104
6.1.3 一维数组应用举例	105
6.2 二维数组	109
6.2.1 二维数组的定义和初始化	109
6.2.2 二维数组元素的引用	111
6.3 字符数组	115
6.3.1 字符数组的定义和引用	115
6.3.2 字符数组的输入与输出	116
6.3.3 字符串处理函数	117

6.4 总结与提高	122
6.4.1 小结	122
6.4.2 典型题例	123
习题 6	128
第 7 章 函数	135
7.1 概述	135
7.1.1 C 语言程序的基本结构	135
7.1.2 函数分类	137
7.2 函数的定义	137
7.3 函数的参数和函数的返回值	139
7.3.1 形式参数和实际参数	139
7.3.2 函数与数组	140
7.3.3 函数的返回值	143
7.4 函数的调用	144
7.4.1 函数调用的一般形式	144
7.4.2 函数调用的方式	145
7.5 函数声明和函数原型	146
7.6 函数的嵌套调用	147
7.7 函数的递归调用	148
7.8 变量的作用域	152
7.8.1 局部变量	152
7.8.2 全局变量	152
7.9 变量的存储类型	154
7.9.1 静态存储方式和动态存储方式	154
7.9.2 静态局部变量	155
7.9.3 自动变量	156
7.9.4 寄存器变量	156
7.9.5 静态全局变量和非静态全局变量	157
7.9.6 存储类型小结	158
7.10 内部函数和外部函数	159
7.10.1 内部函数	159
7.10.2 外部函数	159
7.11 多文件程序的运行	160
7.12 总结与提高	161
7.12.1 小结	161
7.12.2 典型题例	162
习题 7	165
第 8 章 结构体与共用体	176
8.1 结构体类型定义	176

8.2 结构体变量	177
8.2.1 结构体变量的定义与初始化	177
8.2.2 结构体变量的引用	179
8.3 结构体数组	180
8.3.1 结构体数组的定义与初始化	181
8.3.2 结构体数组元素的引用	182
8.4 结构体和函数	183
8.4.1 结构体作函数参数	183
8.4.2 返回值为结构体类型的函数	184
8.5 共用体	185
8.5.1 共用体类型定义	185
8.5.2 共用体变量定义与引用	186
8.5.3 共用体应用举例	187
8.6 枚举类型	188
8.7 typedef 语句	190
8.8 总结与提高	191
8.8.1 小结	191
8.8.2 典型题例	192
习题 8	194
第 9 章 指针	199
9.1 地址和指针的概念	199
9.1.1 变量的内容和变量的地址	199
9.1.2 直接访问和间接访问	200
9.1.3 指针的概念	200
9.2 指针变量	201
9.2.1 指针变量的定义	201
9.2.2 指针变量的引用	202
9.2.3 实现引用传递	203
9.3 指针与数组	206
9.3.1 指向数组元素的指针	206
9.3.2 字符指针、字符数组和字符串	211
9.3.3 地址越界问题	213
9.3.4 指针数组	214
9.3.5 多维数组和指向分数组的指针	216
9.4 结构体与指针	221
9.4.1 指向结构体的指针变量	221
9.4.2 用指向结构体的指针作函数参数	221
9.5 返回值为指针类型的函数	222
9.6 指针与链表	224

9.6.1	存储空间的分配和释放	224
9.6.2	动态数据结构——链表	225
9.6.3	指向函数的指针和函数参数	230
9.7	总结与提高	233
9.7.1	小结	233
9.7.2	典型题例	234
习题 9		235
第 10 章	编译预处理和位运算	246
10.1	文件包含处理——#include	246
10.2	宏定义——#define	248
10.2.1	不带参数的宏定义	248
10.2.2	带参数的宏定义	249
10.3	条件编译	250
10.4	位运算符和位运算	252
10.5	位段	254
10.6	位运算举例	256
10.7	总结与提高	257
10.7.1	小结	257
10.7.2	典型题例	257
习题 10		259
第 11 章	文件	264
11.1	文件的概述	264
11.1.1	数据流	264
11.1.2	文件	264
11.1.3	文件的操作流程	266
11.1.4	文件和内存的交互处理	267
11.2	文件类型的指针	267
11.3	标准输入/输出函数	268
11.3.1	打开文件	268
11.3.2	关闭文件	270
11.3.3	获取文件的属性	271
11.3.4	文件的顺序读写	272
11.3.5	文件的随机读写	282
11.3.6	出错检查	285
11.4	系统输入/输出函数	286
11.5	总结与提高	286
11.5.1	小结	286
11.5.2	典型题例	287
习题 11		296

第 12 章 C 语言图形程序设计	302
12.1 图形显示的坐标和像素	302
12.1.1 图形显示的坐标	302
12.1.2 像素	302
12.2 Turbo C 支持的图形适配器和图形模式	303
12.3 图形模式的初始化	304
12.3.1 图形系统的初始化函数	304
12.3.2 图形系统的自动检测函数	305
12.3.3 关闭图形模式函数	306
12.3.4 屏幕颜色的设置	306
12.3.5 清屏函数	307
12.4 基本绘画	307
12.4.1 画点	307
12.4.2 画直线函数	309
12.4.3 画矩形和多边形函数	310
12.4.4 画圆和椭圆函数	311
12.4.5 设定线形函数	312
12.5 图形填充	313
12.5.1 画填充图函数	313
12.5.2 设定填充方式函数	314
12.5.3 可对任意封闭图形填充的函数	316
12.6 图形模式下的文本输出	317
12.6.1 文本输出函数	317
12.6.2 字体格式和输出方式的设置	318
12.7 视口和视口函数	320
12.8 屏幕操作	322
12.8.1 设置显示页函数	322
12.8.2 屏幕图像处理存储和显示函数	323
12.8.3 键盘对屏幕的控制操作	325
12.9 总结与提高	327
12.9.1 小结	327
12.9.2 典型题例	327
习题 12	330
附录	333
附录 I 常用字符与 ASCII 码对照表	333
附录 II C 语言中的关键字表	336
附录 III C 语言中运算符的优先级与结合性一览表	336
附录 IV C 库函数	337
附录 V Visual C++ 集成开发环境	347

附录VI Win-TC 环境的使用	357
附录VII 实验报告单	359
参考文献	362

第1章 概 述

人们使用计算机解决问题时，必须用某种“语言”来和计算机进行交流。具体地说，就是利用某种计算机语言提供的命令来编制程序，并把程序存储在计算机的存储器中，然后利用这个程序来控制计算机的运行，以达到解决问题的目的。这种用于编写计算机可执行程序的语言称为程序设计语言。目前已发明的计算机程序设计语言有上千种，无论什么样的计算机语言，其程序设计的基本方法都是相同的。本书将以国际上广泛流行的 C 程序设计语言为例，介绍程序设计的基本概念和基本方法。

1.1 程序设计语言

1.1.1 计算机语言

计算机语言是人们描述计算过程（即程序）的规范书写语言。程序是对计算机处理对象和计算规则的描述。语言的基础是一组记号和语法规则。根据语法规则由记号构成记号串的全体就是语言。

人类是使用像英语、汉语这样的自然语言相互交流和表达思想的。人与计算机如何“交流”呢？人和计算机交流也要用人和计算机都容易接受和理解的语言，这就是计算机语言。计算机语言是根据计算机的特点而编制的，是计算机能够“理解”的语言，它是有限规则的集合。计算机语言不像自然语言那样包含复杂的语义和语境，而是用语法来表达程序员的思想，所以编写程序时必须严格遵守语法规则。

1.1.2 程序设计语言的发展

计算机是一种具有内部存储能力、由程序自动控制的电子设备。人们将需要计算机做的工作写成一定形式的指令，并把它们存储在计算机的内部存储器中。当人们需要结果时就向计算机发出一条简单的命令，计算机就按指令顺序自动进行操作。人们把这种可以连续执行的一条条指令的集合称为程序。也就是说，程序是计算机指令的序列，编制程序的工作就是为计算机安排指令序列。

程序设计语言伴随着计算机技术的发展层出不穷，从机器语言到高级语言，从面向过程的语言到面向对象的语言。至目前为止，计算机语言的发展大致经历了五代。

第一代也称机器语言，它将计算机指令中的操作码和操作数均以二进制代码形式表示，是计算机能直接识别和执行的语言。它在形式上是由“0”和“1”构成的一串二进制代码，每种计算机都有自己的一套机器指令。机器语言的优点是无需翻译、占用内存少、执行速度快；缺点是随机而异，通用性差，并且因指令和数据都是二进制代码形式，难以阅读和记忆，编码工作量大，难以维护。

第二代语言也叫汇编语言，是用助记符号来表示机器指令的符号语言。如用 ADD

表示加法,用 SUB 表示减法,用变量取代各类地址,这样构成的计算机符号语言称为汇编语言。用汇编语言编写的程序称为汇编语言源程序。这种程序必须经过翻译(称为汇编),变成机器语言程序才能被计算机识别和执行。汇编语言在一定程度上克服了机器语言难以辨认和记忆的缺点,但对于大多数用户来说,仍然不便于理解和使用。

第三代语言即高级语言,也称为面向过程的语言。高级语言是具有国际标准的,描述形式接近自然语言的计算机语言。它采用了完全符号化的描述形式,用类似自然语言的形式描述对问题的处理过程,用数学表达式的形式描述对数据的计算过程。常用的计算机高级语言有 Basic、Fortran、Cobol、Pascal 和 C 语言等。由于高级语言只是要求人们关心计算机描述问题的求解过程,而不关心计算机的内部结构,所以把高级语言称为面向过程的语言。使用面向过程的语言编程时,编程者的主要精力放在算法过程的设计和描述上。

第四代语言又叫非过程化语言,是一种功能更强的高级语言。主要特点是:非过程性、采用图形窗口和人机对话形式、基于数据库和“面向对象”技术。易编程、易理解、易使用、易维护。但是,程序的运行效率和语言的灵活性不如过程化语言高。常用的非过程化语言有 Visual Basic、Java、C++和 Delphi 等。

如果说第三代语言要求人们告诉计算机“怎么做”,那么第四代语言只要求人们告诉计算机“做什么”。

第五代语言也称智能化语言,主要使用在人工智能领域,帮助人们编写推理、演绎程序。

目前,国内外大多数计算机上运行的程序,大多是用第三代或第四代计算机语言编写的。因此,应当熟练地掌握用高级语言编写程序的方法和技巧。

由于面向过程的语言是程序设计的基础,所以,本书将以面向过程的 C 语言为背景,介绍程序设计的基本概念和基本方法。

1.2 C 程序设计语言

1.2.1 C 语言的发展历史

C 语言是国际上流行的,使用最广泛的高级程序设计语言。它既用来写系统软件,也可用来写应用软件。C 语言具有语言简洁紧凑,使用方便灵活及运算符丰富等特点,它具有现代化语言的各种数据结构,有结构化的控制语句,并且语法限制不太严格,程序设计自由度大,能实现汇编语言的大部分功能。另外,C 语言生成目标代码质量高,不仅程序执行效率高,而且程序可移植性好。

C 语言的产生基于两个方面的需要。一是为满足 UNIX 操作系统开发的需要。UNIX 操作系统是一个通用的、复杂的计算机管理系统。二是为接近硬件的需要,即直接访问物理地址、直接对硬件进行操作的需要。C 语言集高级语言与汇编语言优点于一身。C 语言面对实际应用的需要而产生,直至今日仍不改初衷。C 语言是从 BCPL (Basic Combined Programming Language) 语言和 B 语言演化而来的。1960 年出现的 ALGOL 语言是一种面向问题的高级语言,远离硬件,但不适于开发系统软件。1963 年,英国剑桥大学推出 CPL 语言,CPL 语言比 ALGOL 语言接近硬件一些,但规模较大,难以实现。1969 年,

剑桥大学的 M.Richards 对 CPL 语言进行简化, 推出 BCPL 语言。1970 年, 贝尔实验室的 K.Thompson 为 DEC 公司 PDP-7 计算机上运行的一种早期 UNIX 操作系统设计了一种类 BCPL 语言, 称为 B 语言。B 语言规模小, 接近硬件, 1971 年在 PDP-11 计算机上实现, 并用 B 语言编写了 UNIX 操作系统和绝大多数实用程序。由于 B 语言面向字存取、功能过于简单、数据无类型和描述问题的能力有限, 而且编译程序产生的是解释执行的代码, 运行速度慢, 因此没有流行起来。1972~1973 年, 贝尔实验室的 D.M.Ritchie 在保留 B 语言优点的基础上, 设计出一种新的语言。这种新语言克服了 B 语言功能过于简单、数据没有类型和描述问题的能力有限的缺点, 扩充了很多适合于系统设计和应用开发的功能。因为这种新语言是在 B 语言的基础上开发出来的, 不管是在英文字母序列中也好, 还是在 BCPL 这个名字中也好, 排在 B 后面的均为 C, 因此将这种语言命名为 C 语言。1973 年, UNIX 操作系统被用 C 语言改写, 称为 UNIX 第五版。最初的 C 语言只是一种 UNIX 操作系统的工作语言, 依附于 UNIX 系统, 主要在贝尔实验室内部使用。

UNIX 以后的第六版、第七版、SYSTEM III 和 SYSTEM V 等都是在第五版的基础上发展起来的, C 语言也做了多次改进。到 1975 年, 随着 UNIX 第六版的公布, C 语言越来越受到人们的普遍注意。

UNIX 操作系统的广泛使用, 促进了 C 语言的迅速发展及普及。同时, C 语言的发展与普及也促进了 UNIX 操作系统的推广。到 1978 年出现了独立于 UNIX 和 PDP 计算机的 C 语言, 从而, C 语言被迅速移植到大、中、小与微型机上。当年, B.W.Kernighan 和 D.M.Ritchie 以 UNIX 第七版的 C 编译程序为基础, 出版了影响深远的名著《C 程序设计语言》。如今, C 语言已经风靡全球, 成为世界上应用非常广泛的程序设计语言之一。

1.2.2 C 语言的标准与版本

随着 C 语言的普及, 各机构分别推出了自己的 C 语言版本。某些执行过程的微小差别不时引起 C 语言程序之间的不兼容, 给程序的移植带来很大的困难。美国国家标准协会 (ANSI) 从 1983 年开始, 经过长达五年的努力, 制定了 C 语言的新标准——ANSI C。现在提及 C 语言的标准就是指该标准。ANSI C 比原标准 C 有很大的发展, 解决了经典定义中的二义性, 给出了 C 语言的新特点。任何 C 语言程序都必须遵循 ANSI C 标准, 本教材也以 ANSI C 作为基础。尽管这样, 各种版本的 C 编译系统还是略有差异, 因此, 读者在使用具体的 C 语言编译系统时, 还应参考相关的手册以了解具体的规定。

C 语言环境有多种版本: TC 2.0、BC 3.1、BC 5.0 和 VC 6.0 等, 最新的是为 Windows 和 Web 应用程序提供动态开发环境的 Visual C++、.NET (C#)。

1.2.3 C 语言的特点

C 语言的主要特点有如下几方面。

(1) C 语言简洁、紧凑, 编写的程序短小精悍

C 编译程序的代码量较小, 便于在微型机上应用。

(2) 运算符丰富, 数据结构丰富

C 的数据类型有整型、实型、字符型、数组类型、指针类型、结构体类型和共用体

类型等, 能实现各种复杂数据类型的数据进行运算, 并引入了指针概念, 使程序效率更高。另外, C 语言具有强大的图形功能。

(3) C 语言是一种结构化程序设计语言, 具有结构化语言所要求的三种基本结构

这种结构化方式可使程序层次清晰, 便于使用、维护和调试。C 语言是以函数形式提供给用户的, 这些函数可方便地调用, 并具有多种循环、条件语句控制程序流向, 从而使程序完全结构化。

(4) C 语言允许直接访问物理地址

C 语言能按位进行运算, 能实现汇编语言的大部分功能, 能直接对硬件进行操作。C 语言既具有高级语言的功能, 又具有低级语言的许多特点, 可以用来编写系统程序。

(5) C 语言预处理机制

C 语言提供预处理机制, 有利于大程序的编写和调试。

(6) C 语言可移植性好

编写的程序不需要做很多改动就可从一种机型上移到另一种机型上运行。C 语言有一个突出的优点就是适合于多种操作系统, 如 DOS、UNIX, 也适用于多种机型。

(7) C 语言语法限制不太严格, 程序设计自由度大, 对程序员要求不高

一般的高级语言语法检查比较严格, 能够检查出几乎所有的语法错误。而 C 语言允许程序编写者有较大的自由度。

(8) C 语言程序生成代码质量高, 程序执行效率高

一般只比汇编程序生成的目标代码效率低 10%~20%。

C 语言也存在某些缺点, 例如, 运算符较多, 某些运算符优先顺序与习惯不完全一致, 类型转换比较随便等。

1.3 C 语言的基本语法成分

1.3.1 字符集

字符是可以区分的最小符号, 是构成程序的基础。C 语言字符集是 ASCII 字符集的一个子集, 包括英文字母、数字及特殊字符。

1) 英文字母: a~z 和 A~Z。

2) 数字: 0~9。

3) 特殊字符: 空格 ! # % ^ & * - _ + = ~ < > /\ | . , ; ; ? ' " () [] { } 。

字符集中的字符可以构成 C 语言的语法成分, 如标识符、关键字和特殊的运算符等。

1.3.2 标识符

标识符在程序中是用来标识各种程序成分, 命名程序中的一些实体, 如变量、常量、函数、类型和标号等不同对象的名字。

C 语言规定, 合法的标识符必须由英文字母或下划线开头, 是字母、数字和下划线的序列, 不能跨行书写, 自定义的标识符不能与关键字同名。