



河南省高等学校计算机教育研究会统编教材

C语言程序设计

董汉丽 主编



中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

河南省高等学校计算机教育研究会统编教材

C 语言程序设计

董汉丽 主编

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

内 容 简 介

本书在借鉴了同类教材的基础上,对 C 语言传统教材的章节进行了有益的探索和改进、合并和分解。本书编排合理、通俗易懂、语言简练、深入浅出。通过本书的学习,读者能较轻松顺利地掌握 C 语言的基本内容并进行基本的程序设计。全书的主要内容有数据类型、表达式及其运算,顺序结构程序设计,选择结构程序设计,循环结构程序设计,数组,函数,结构体与共用体,文件,位运算和图形程序设计。每章节后面都提供了大量的习题和上机实习,非常便于读者将理论和实践相结合。

本书既适合作为高等院校各专业的学生学习 C 语言的教材,也适合各类报考计算机等级考试 C 语言的读者学习,同时也可作为自学考试或各种计算机培训班的教材。

图书在版编目(CIP)数据

C 语言程序设计/董汉丽主编. —北京:中国铁道出版社, 2007.6

河南省高等学校计算机教育研究会统编教材

ISBN 978-7-113-07783-9

I. C… II. 董… III. C 语言—程序设计—高等学校—教材 IV. TP312

中国版本图书馆 CIP 数据核字(2007)第 085106 号

书 名: C 语言程序设计

作 者: 董汉丽 等

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 秦绪好

责任编辑: 杨 勇

特邀编辑: 李振华

封面制作: 白 雪

责任校对: 姚文娟

印 刷: 三河市华晨印务有限公司

开 本: 787×1092 1/16 印张: 20.25 字数: 462 千

版 本: 2007 年 7 月第 1 版 2007 年 7 月第 1 次印刷

书 号: ISBN 978-7-113-07783-9/TP·2148

定 价: 28.00 元

版权所有 侵权必究

本书封面贴有中国铁道出版社激光防伪标签,无标签者不得销售

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。

河南省高等学校计算机教育研究会

主任委员：段银田

副主任委员：甘 勇 普杰信 王贺明

秘 书：李学相

委 员：段银田 甘 勇 普杰信 王贺明 李学相

翁 梅 曲宏山 郭清溥 申石磊 周清雷

刘克成 陆桂明 程万里 马占欣 陈 涛

张东升 朱国华 李 敏 黄贻彬 商信华

连卫民 杨立峰

自上个世纪 80 年代初到本世纪初的 20 年间, 由于计算机奇迹般地展示出它惊人的运算速度、海量的存储能力和神奇的创造性, 使人类社会深深地感受到了计算机的存在和它的不可或缺性。在这种背景下, 全国各类高等学校已陆续开展了计算机基础教育, 普及了计算机文化基础知识和技术基础知识。相应地, 这两种类型的教材也大量涌现, 为计算机教育和应用的普及提供了丰富的智力资源。

然而, 进入 21 世纪以来, 高等学校的计算机基础教育面临着新的挑战。首先, 一个时期以来, 信息技术自身愈来愈向技术多元化的方向发展。网络、数据库、多媒体等技术已从科学的殿堂里走了出来, 并日益得到应用和普及, 各种信息技术在工程中的综合应用程度越来越高, 这一切促使全社会计算机的应用水平提升到了一个新高度。应用的普及也推动了需求的进一步多样化, 社会也因此更加迫切需要实用型信息技术人才。在这种背景下, 大学现行计算机基础教育教材已远远不能适应技术发展和应用的要求。其次, 由于近年来中小学信息技术教育的普遍开展, 使得原本在大学要完成的信息技术学习任务的一部分已经提前完成, 因此也需要调整当前高等学校计算机应用基础教学的内容, 可见更新现行教材已成为当前一项十分紧迫的任务。作为高等学校计算机基础教育教材改革创作的尝试, 河南省高等学校计算机教育研究会与中国铁道出版社共同策划了这套系列教材。

本套教材的创作是以社会对信息技术的应用需求为目标, 学习的方向应瞄准应用, 学习的目的是能够做事的观念。要知道, 仅能在操作层面上使用计算机并不是真正意义上的应用, 开发才是真正的应用, 也就是常说的开发应用, 这也就是大学生学习信息技术的方向和应采取的行动。这些观念应逐步成为教材创作的指导思想。

突出信息技术教育的目的性是本系列丛书内容的最大特色。信息技术何其多! 究竟学什么、写什么? 要改变那种无的放矢的、包罗万象的教材创作模式; 要有目的的去写过程, 摒弃那种遍历知识过程就是一切、没有目的、文字堆砌式的创作观念和方法。应当明白, 学习信息技术是为了做事情, 而不是为了其他。此外, 计算机基础教育的教材要提倡精简。要树立信息量观点, 能够释疑解惑的文字构成信息量, 可以写入教材, 不能起到释疑解惑作用的文字或冗余文字只能形成垃圾信息, 应当从教材中剔除出去。

例如, 对于操作技能类的教材来说, 完全可以按“展示一种目的, 精讲一个案例, 完成一个练习, 创造一个作品”这四句话的要求来进行教材创作。对于程序设计类教材来说, 教材应逐步体现并满足从程序设计向软件设计延伸的社会需求。

在教材创作中，应努力完成相关知识的整合，这不仅是本套教材所提倡的创作特色之一，也是信息技术教育改革的出路所在。对于计算机基础教育来说，知识整合主要体现在两个方面。其一，用公用事件整合适用的信息技术。把面向社会大众所发生的信息技术应用事件用其所必需的信息技术，而不是某个领域的全部信息技术加以整合应用。把从目的到技术的逆向思维作为新一代信息技术教材创作的思维方法和行为方法。整个创作过程应按照“目的决定过程，过程决定事件，事件决定对象，对象决定技术”的思路进行。其二，信息技术与其他特定学科的相互整合。这种整合开辟了信息技术与专业相结合进行教材创作的途径。更加有利于实现从目的到技术进行教材创作的思想，使特定学科的内容和信息技术实现“我中有你，你中有我”，达到更高层次的融合。这种融合有利于双方共同提高教学效率，拓宽知识领域，增加知识深度，激发创造思维。总之，本丛书的创作特色主要体现在用目的、事件、对象去整合适用的信息技术。帮助读者为了达到目的而学会利用信息技术做一些实实在在的事情。

最后，本人深知新一代计算机基础教育教材的创作远不是一蹴而就的事情，目标的实现尚需时日。序言的目的仅在于简要阐明本套教材在策划过程中提出的一些基本思想和对创作的原则要求，正确与否还需经过实践的检验。望作者和读者在创作与实践不断斧正。

河南省高等学校计算机教育研究会理事长

殷銀田

由于C语言特别适合于编写系统软件,所以,它不仅为计算机专业人员所广泛使用,更为广大计算机应用人员所喜爱。高等院校也都开设了C语言这门课程,学习C语言已成为广大青年学生的迫切要求。它既是一门实践性和应用性较强的公共必修课,又是计算机专业学生学习程序设计的启蒙语言课程,它还是掌握C++的基础课程。要使学生较好地掌握程序设计的基本思想和方法,具备运用计算机解决实际问题的能力,是离不开一本较适合的、有针对性的好教材的。出于这一目的,我们编写了这本C语言教材。

我们针对初学者的特点,从其思维方式、理解能力、学习C程序容易产生困惑的现实出发,对传统的C语言内容和结构进行了有益的探索和改进,合并和分解了一些内容,并融入了作者多年的C语言教学经验,合力编写了本书。本书语言上力求简明扼要、文字流畅、通俗易懂;结构上体现章节合理、脉络清晰、环环相扣;内容上做到删繁就简、深入浅出、别有新义。

本书的最大特点:

1. 将指针内容化解,分散到相关章节中,这样,不仅降低了学习指针的难度,使学生由浅入深,较容易地掌握指针的用法,还摆脱了以往教材中将指针独立成章,老师不好教、学生不好学的难题。

2. 本书每章文前有引言、文中配有大量实例,这样易于学生把握知识要点。为配合教学,加强理论与实践的结合,本书每一章节最后,都配有大量的习题及上机实习的内容。这便于学生学以致用、提高编程的实践能力。

3. 增加了图形、动画的内容。因为,图形是C语言中较为吸引人的内容。它的直观性可以激发、调动学生的学习及编程兴趣,也提高了教材的可读性。

4. 参加本教材编写工作的都是讲授“C语言程序设计”课程的一线教师。根据多年的教学经验和体会,从初学者的角度,在内容编排、概念描述、叙述方法、结构衔接等方面,本教材都进行了大胆的尝试和改进。

本书共分11章。董汉丽担任主编。第1章由李婷编写;第2章由侯枫编写;第3章由白艳玲编写;第4章、第5章由张玉华编写;第6章由王峰编写;第7章7.1节~7.9节由薛淑静编写;第7章7.10、7.11节由董洁编写;第8章由王文莉编写;第9章由董汉丽编写;第10章、第11章由韩彦峰编写;附录内容由炊昆编写。本书的编写得到了各级领导的关心和支持,在此一并表示深深的感谢。

由于编者水平有限,加之时间仓促,书中难免有疏漏和不足之处,敬请专家及广大读者批评指正。

编 者

2007年3月

第 1 章 C 语言概述	1
1.1 C 语言的发展与特点	1
1.1.1 C 语言的发展	1
1.1.2 C 语言的特点	2
1.2 C 程序结构与特点	3
1.2.1 C 程序结构	3
1.2.2 C 程序源文件的特点	6
1.3 Turbo C 集成环境与程序的上机操作	6
1.3.1 Turbo C 集成编程环境	6
1.3.2 上机操作与查错	9
习题	11
上机实习	11
第 2 章 数据类型、表达式及其运算	13
2.1 C 语言的数据类型	13
2.1.1 C 语言中的关键字、标识符	13
2.1.2 C 语言数据类型的取值范围	14
2.2 常量	15
2.2.1 整型常量	15
2.2.2 实型常量	16
2.2.3 字符型常量	16
2.2.4 字符串常量	17
2.2.5 符号常量	17
2.3 变量	18
2.3.1 变量的概念	18
2.3.2 变量的定义与初始化	18
2.4 C 语言的运算符及表达式	23
2.4.1 C 语言的运算符	23
2.4.2 算术运算符	25
2.4.3 算术表达式	27
2.4.4 算术表达式数据间的混合运算与类型转换	27
2.5 赋值运算符和赋值表达式	29
2.5.1 基本赋值运算符	29
2.5.2 复合赋值运算符	30
2.5.3 赋值表达式	30
2.6 指针运算	31
2.6.1 地址与指针的概念	31

2.6.2 指针变量的定义与引用	32
2.6.3 指针的运算	35
2.7 逗号运算符与逗号表达式	37
习题	38
上机实习	39
第3章 顺序结构程序设计	41
3.1 结构化程序设计的基本概念及C语句	41
3.1.1 结构化程序设计的基本概念	41
3.1.2 C语句概述	42
3.2 赋值语句	43
3.3 数据的输入/输出	44
3.3.1 格式输入与输出	44
3.3.2 字符数据的输入函数 getchar()与输出函数 putchar()	51
习题	52
上机实习	53
第4章 选择结构程序设计	56
4.1 关系表达式和关系运算符	56
4.1.1 关系表达式	56
4.1.2 关系运算符及其优先次序	56
4.2 逻辑运算符及逻辑表达式	57
4.2.1 逻辑运算符及其优先次序	57
4.2.2 逻辑表达式	58
4.3 if 语句	58
4.3.1 if 语句的3种形式	59
4.3.2 if 语句的嵌套	63
4.3.3 条件运算符	65
4.4 switch 语句	66
习题	68
上机实习	68
第5章 循环结构程序设计	72
5.1 while 语句	72
5.2 do...while 语句	73
5.3 for 语句	75
5.4 3种循环语句的比较	78
5.4.1 3种循环语句的不同用法	78
5.4.2 编写循环结构程序的要点	78
5.5 循环结构中用到的其他语句	78
5.5.1 break 语句	78
5.5.2 continue 语句	80
5.5.3 goto 语句	80

5.6 多重循环	81
习题	83
上机实习	84
第 6 章 数组	86
6.1 一维数组	86
6.1.1 一维数组的定义	86
6.1.2 一维数组元素的引用	87
6.1.3 一维数组的初始化	90
6.2 二维数组	90
6.2.1 二维数组的定义	90
6.2.2 二维数组的初始化	91
6.2.3 二维数组元素的引用	92
6.3 字符型数组与字符串	93
6.3.1 字符数组的定义	93
6.3.2 字符数组的初始化	94
6.3.3 字符数组的引用	94
6.3.4 字符串和字符串结束标志	95
6.3.5 字符数组的输入/输出	96
6.3.6 字符串处理函数	97
6.4 指针与一维数组	100
6.4.1 数组的指针和指向数组的指针变量	100
6.4.2 字符串的指针和指向字符串的指针变量	103
6.5 指针数组、指针与二维数组	105
6.5.1 指针数组的定义	105
6.5.2 指针数组的引用	105
6.5.3 指向二维数组元素的指针	107
6.5.4 指针引用二维数组元素	108
6.5.5 指向指针的指针变量	109
6.6 程序举例	111
习题	115
上机实习	116
第 7 章 函数	120
7.1 概述	120
7.1.1 模块化程序结构的概念	120
7.1.2 C 语言程序的构成	121
7.2 C 函数的分类与定义	122
7.2.1 C 语言函数的分类	122
7.2.2 C 语言函数的定义	122
7.3 函数的参数和函数的值	123
7.3.1 函数的形式参数和实际参数	123

7.3.2 函数的返回值	125
7.4 函数的调用	126
7.4.1 函数的原型与声明	126
7.4.2 函数的调用	128
7.5 函数的嵌套调用	129
7.6 函数的递归调用	130
7.6.1 递归的概念	130
7.6.2 递归三要素	131
7.6.3 递归函数程序设计举例	132
7.7 指针与函数	134
7.7.1 指针变量作为函数的参数	134
7.7.2 数组作为函数参数	135
7.7.3 返回指针值的函数	141
7.7.4 函数的指针和指向函数的指针变量	142
7.7.5 main 函数的形参	144
7.8 变量的作用域与存储类别	145
7.8.1 变量的作用域	145
7.8.2 变量的存储类别	149
7.9 内部函数和外部函数	154
7.9.1 内部函数	154
7.9.2 外部函数	154
7.9.3 多个源程序文件的编译与连接	155
7.10 程序编译预处理	156
7.10.1 宏定义	156
7.10.2 条件编译	159
7.11 指针运算小结和程序举例	161
7.11.1 有关指针的数据类型和指针运算小结	161
7.11.2 程序举例	162
习题	167
上机实习	168
第 8 章 结构体与共用体	171
8.1 结构体类型的定义	171
8.2 结构体变量的引用	173
8.3 结构体变量的初始化	175
8.4 结构体数组	175
8.4.1 定义结构体数组	175
8.4.2 结构体数组的初始化和引用	176
8.4.3 结构体数组应用举例	177
8.5 结构体类型的指针变量	179
8.5.1 指向结构体变量的指针	179

8.5.2	指向结构体数组的指针.....	181
8.5.3	用指向结构体的指针作函数参数.....	182
8.5.4	结构体应用举例.....	184
8.6	链表.....	185
8.6.1	链表概述.....	185
8.6.2	单向链表.....	186
8.6.3	动态链表所需要的函数.....	187
8.6.4	动态链表的建立与输出.....	188
8.6.5	链表的插入与删除.....	189
8.6.6	链表举例.....	191
8.7	共用体.....	193
8.7.1	共用体的概念.....	193
8.7.2	共用体变量的引用方式.....	195
8.7.3	共用体类型数据的特点.....	195
8.8	枚举类型.....	197
8.8.1	枚举类型和枚举变量的定义.....	197
8.8.2	枚举类型的使用.....	198
8.8.3	枚举类型程序举例.....	199
8.9	用 typedef 定义新的类型.....	199
	习题.....	200
	上机实习.....	202
第9章	文件.....	206
9.1	C 文件概述.....	206
9.1.1	文件的分类.....	206
9.1.2	文件的操作过程.....	207
9.2	文件类型指针.....	207
9.3	文件的打开与关闭.....	208
9.3.1	文件的打开函数 fopen().....	208
9.3.2	文件的关闭函数 fclose().....	209
9.4	文件的读写.....	209
9.4.1	字符读写函数 fputc()和 fgetc().....	210
9.4.2	字符串读写函数 fgets()和 fputs().....	212
9.4.3	数据块读写函数 fread()和 fwrite().....	213
9.4.4	格式化读写函数 fprintf()和 fscanf().....	215
9.5	文件定位与随机读写.....	217
9.5.1	函数 rewind().....	217
9.5.2	函数 fseek()与随机读写.....	218
9.5.3	函数 ftell().....	220
9.6	文件检测函数.....	221
9.6.1	函数 feof().....	221

9.6.2 函数 <code>ferror()</code>	221
9.6.3 函数 <code>clearerr()</code>	222
习题	222
上机实习	222
第 10 章 位运算	225
10.1 位运算符和位运算	225
10.1.1 位运算符	225
10.1.2 位运算的功能与应用	225
10.2 应用举例	230
10.3 位段	233
10.3.1 位段的定义和位段变量的说明	233
10.3.2 位段的引用方法	234
习题	237
上机实习	237
第 11 章 图形程序设计	239
11.1 图形模式	239
11.1.1 显示器基础	239
11.1.2 Turbo C 图形方式	240
11.2 基本绘图函数	243
11.2.1 绘图函数	243
11.2.2 填充函数	250
11.2.3 图形存取函数	256
11.3 图形方式下的文本输出	258
11.3.1 设置字体函数	258
11.3.2 输出字符串函数	259
11.4 程序举例	260
习题	265
上机实习	266
附录 A 常用字符与 ASCII 代码对照表	267
附录 B Turbo C 库函数	268
附录 C 常见错误分析和程序调试	279
附录 D 习题参考答案	287
参考文献	307

第 1 章 C 语言概述

C 语言是目前世界上最为流行的计算机高级程序设计语言之一。它设计精巧, 功能齐全, 既适合于编写应用软件, 又适合于编写系统软件。C 语言是一种很灵活的语言, 它把许多程序设计技巧交给了编程人员。掌握了 C 语言后, 再学 C++、Java、C# 语言就比较容易了。本章将简明扼要地介绍 C 语言的发展、特点、C 程序的结构与上机步骤。

1.1 C 语言的发展与特点

1.1.1 C 语言的发展

1. 概述

在诸多的计算机高级语言中, C 语言是目前国内外最为流行的计算机高级程序设计语言之一。它既可以用来编写系统软件, 也可以用来编写应用程序, 功能丰富, 表达力强, 应用面广。它既具有高级语言的特点, 又具有低级语言的许多功能。

以往, 操作系统等系统软件主要采用汇编语言编写。汇编语言的缺陷较多: 过分依赖于计算机硬件, 程序的可读性、可移植性都比较差。为克服这些缺点, 人们尝试采用高级语言来编写系统软件。可是, 高级语言又难以实现针对硬件的一些操作 (如内存地址的读写、二进制位的操作)。实践的需要, 迫使人们去开发一种高级语言和低级语言的特性兼而有之的语言。C 语言就在这种情况下应运而生。

C 语言是一种编译型程序语言, 是 UNIX 操作系统的主力语言, 据统计, PC 领域应用最多的系统软件, 就是 C 语言加上汇编语言子程序而编成的系统软件。世界上很多著名的软件都是用 C 语言编写的。C 语言有很多分支, 如 TC、BC、MSC 等, 其结构大同小异。本书主要讲解 Turbo C。

Turbo C 是美国 Borland 公司开发的产品。从 1987 年的 1.0 版本, 到 1989 年的 2.0 版本, 功能上增加了很多。Turbo C 可以在程序的任何地方嵌入汇编语言代码, 并提供了与 FoxBase、Pascal 等语言的接口。可以说, Turbo C 是最受欢迎的 C 编译程序之一。

PC 机 DOS、Windows 平台上常见的 C 语言版本有 Borland 公司的 Turbo C、Turbo C++、Borland C++、C++ Builder (Windows 版本), Microsoft 公司的 Microsoft C、Visual C++ (Windows 版本)。

2. C 语言的发展

C 语言是在早期的 B 语言基础上发展起来的。早期的计算机, 都是用机器语言和汇编语言来编写程序代码的。到了第二代计算机才有了高级语言。

1960 年开发的 ALGOL-60, 对其后的高级语言的发展, 起到了很好的推进作用。但是, 它是一种面向问题的语言, 过于抽象, 难以描述系统。因此, 没有得到真正的推广。

1963 年, 由英国剑桥大学推出了 CPL (Combind Programming Language) 语言。它比 ALGOL-60 更接近于硬件, 但其规模较大, 难以实现和学习。

1967 年,英国剑桥大学的 Martin Richards 对 CPL 语言作了简化,推出了 BCPL (Basic Combined Programming Language) 语言。

1970 年,由贝尔实验室的 Ken Thompson 所写的 B 语言,是对 BCPL 的进一步简化,且更接近于硬件,他还用 B 语言编写了 UNIX 操作系统。

1972 年~1973 年,贝尔实验室的 D. M. Ritchie,在 B 语言的基础上设计出了 C 语言。C 语言既保持了 BCPL 和 B 语言的优点,又克服了它的缺点。

1973 年,K. Thompson 和 D. M. Ritchie 两人合作,把原来由两人用汇编语言编写的 UNIX 操作系统中 90% 以上的代码用 C 语言来重写,即 UNIX 5。后来,C 语言虽作了多次改进,但主要还是用在贝尔实验室内部。直到 1975 年,用 C 语言写的 UNIX 6 公布后,才引起业内人士的广泛关注。

以 1978 年发表的 UNIX 7 中的 C 编译程序为基础,Brian W. Kernighan 和 Dennis M. Ritchie (合称 K&R) 合著了影响深远的著作《The C Programming Language》。该书中介绍的 C 语言成为后来被广泛使用的 C 语言的基础,称它为标准 C。

以后,相继出现了很多 C 语言版本,如 Microsoft C、Turbo C、Quick C、Borland C 等。它们在语法上基本是相同的,但在函数数量和功能上有较大的区别,使用时要注意区分。

1983 年,美国国际标准化协会 (ANSI),在参考 C 语言的各种版本基础上,制定了新的标准,称为 ANSI C。1988 年,K&R 按照 ANSI C 重写了《The C Programming Language》。1990 年,国际标准化组织 ISO (International Standard Organization),接受了以 87 ANSI C 为 ISO C 的标准 (ISO 9899—1990)。目前流行的 C 编译系统,都是以它为基础的。

高级语言发展至今,面向对象的程序设计方法越来越受到人们的青睐。例如,Visual FoxPro (VFP)、Visual Basic (VB)、Visual C++ (VC++)、C++、Turbo C++、Borland C++、Java、J++、Power Builder (PB) 等。其中,功能比较强大的还是 C++ 语言。它以 C 语言为基础,在很多方面二者兼容。因此,掌握了 C 语言,进一步学习 C++ 就能达到事半功倍的效果。

本书以目前流行的 Turbo C 2.0 为开发环境,全面介绍 C 语言及其程序设计。

1.1.2 C 语言的特点

C 语言之所以能够在众多的高级语言竞争中脱颖而出,成为高级语言中的佼佼者,主要是因为与普通高级语言相比,它具有以下独特的特点。

(1) 语言简洁、紧凑,使用方便、灵活。C 语言一共只有 32 个标准的关键字、45 个标准的运算符以及 9 种控制语句。程序书写形式自由,主要用小写字母表示。

(2) 数据类型丰富。C 语言数据类型包括整型、实型、字符型、枚举类型、结构体、共用体、数组、文件类型、指针类型、空类型。其中,整型、实型、字符型中还有多种小的类型。这些数据类型可以表示各种各样的数据结构(如链表、树、栈等)。指针类型是 C 语言中最具特点的一种数据类型。它使用起来非常灵活自如,并可以将 C 语言的功能特点发挥得淋漓尽致。由于指针在运用上非常灵活,所以,它也是初学者最难以驾驭的。

(3) 运算符多样,表达能力强。C 语言中的运算符包含的范围非常广泛,一共有 44 个运算符。除包括算术运算符、关系运算符、逻辑运算符等常规运算符之外,还含有指针运算符、地址运算符、位运算符、自增自减运算符、条件运算符、复合赋值运算符,甚至连圆括号、方括号、逗号、小数点等都是运算符。由于 C 语言的运算符类型极为丰富,表

达式类型多样化,所以,运用它能够实现各种各样的高级和低级运算。

(4) 函数是程序的主体。C语言中,每一项功能的完成都是由函数来实现的。既可以用C语言系统中提供的功能函数,也可以用用户自编的函数来实现。函数是程序的基本单位,是C语言的重要构件,没有函数,就没有C语言。用函数作构件,可以设计开发出结构清晰、功能齐全的大型程序。

(5) 语法检查不太严格,程序书写自由度大。例如,数组下标不作超界检查;整型、字符型、逻辑型可以通用;一个物理行可以写多个语句,一个语句也可以分写在连续的多个物理行上。对这些规则,初学者要认真掌握。

(6) C语言允许直接访问物理地址。C语言中含有位运算和指针运算,能够直接对内存地址进行访问操作,可以实现汇编语言的大部分功能,即直接对硬件进行操作。所以,它既具有高级语言的功能,又兼有低级语言的大部分功能。

(7) 生成的目标代码质量高。C语言简洁、紧凑,程序执行速度快。它比一般的高级语言生成的目标代码质量高约20%,但是,比汇编语言低10%~20%。这在高级语言中是出类拔萃的。

(8) 可移植性好。在C语言所提供的语句中,没有直接依赖于硬件的语句。与硬件有关的操作,如数据的输入、输出等,都是通过调用系统提供的库函数来实现的。库函数本身并不是C语言的组成部分。因此,用C语言编写的程序,能够很容易地从一种计算机环境移植到另一种计算机环境中。

综上所述,正是由于C语言具有其他语言不可比拟的这些优点,使得它越来越受到程序设计人员的重视,并在众多的领域里得到广泛的应用。

当然,任何事物都是一分为二的,C语言也具有弱点:其一,运算符的优先级和结合性比较复杂,不容易记忆;其二,C语言的语法限制不太严格,这在增强了程序设计的灵活性的同时,又在一定程度上降低了某些安全性,这对程序设计人员提出了更高的要求。

1.2 C程序结构与特点

1.2.1 C程序结构

下面通过几个例题,使用户了解C语言程序的结构及执行过程。程序,不要求完全能理解,先对C程序的全貌有一个感性认识,为以后的学习打下基础。

【例 1.1】一个简单的C程序。

```
#include<stdio.h>           /*文件包含*/
main()                      /*程序从函数main开始执行*/
{
    printf("Welcome to C! \n"); /*在屏幕上输出引号内的信息*/
}                             /*函数main结束*/
```

上述程序,是一个简单而完整的C语言程序。经过编辑、编译和连接,程序运行后显示字符串“Welcome to C!”。

说明:

① 本程序的功能是输出一行信息“Welcome to C!”。

② 语句#include<stdio.h>是文件包含。文件包含是把一个指定的文件包含到源程序中。

C语言中没有数据的输入、输出等功能,数据的输入、输出都是通过调用系统提供的库函数 `scanf()`和 `printf()`等来实现的。这些函数的说明都包括在 `stdio.h` 文件中。所以,一般C程序中在文件头都有`#include <stdio.h>`、`#include <math.h>`等语句。

【例 1.2】求两数之和。

```
#include<stdio.h>          /*包含文件*/
main()                     /*定义主函数*/
{
    int a,b,sum;           /*定义局部变量 a, b, sum*/
    scanf("%d%d",&a,&b);    /*读入两个整数,存入变量 a 和 b 中*/
    sum=a+b;               /*将 a, b 的和送入 sum*/
    printf("sum=%d\n",sum); /*打印结果*/
}
```

运行结果:

输入 10, 20✓

显示:

sum=30

上述程序经过编辑、编译和连接后,执行情况如下:

此程序从 `main` 函数开始执行,执行到 `scanf` 语句时,等待用户从键盘输入两个整型数据后,再继续执行。假如用户输入 10 和 20 (此时, `a` 的值为 10, `b` 的值为 20)。最后,由 `printf` 语句打印出结果: `sum=30`。

【例 1.3】求两个数的最大值。

```
int max(int x,int y)       /*定义函数 max, 求两个数的最大值*/
{
    int z;                 /*定义变量 z*/
    if(x>y)z=x;            /*进行条件判断,将最大值赋给变量 z*/
    z=y;
    return(z);             /*返回函数值*/
}
#include<stdio.h>         /*包含文件*/
main()                    /*主函数*/
{
    int a,b,c;             /*定义变量 a, b, c 为整型*/
    scanf("%d,%d",&a,&b);   /*输入 a, b 的值*/
    c=max(a,b);            /*调用 max 函数,将函数返回的最大值赋给变量 c*/
    printf("max=%d\n",c);  /*打印 c 的值*/
}
```

运行结果:

输入 10, 20✓

显示:

max=20

程序由主函数 `main()`和自定义函数 `max()`组成。此程序从 `main` 函数开始执行,执行到 `scanf` 语句时,等待用户从键盘输入两个整型数据后,再继续执行。假设输入 10 和 20,按

【Enter】键后,系统自动赋给 `a` 和 `b`,开始调用 `max` 函数。执行 `max` 函数时,把 `a` 和 `b` 的值传递给形参 `x`、`y`,然后求两个数的最大值,计算结束,退出 `max` 函数,返回到 `main` 函数,将返回的两个数的最大值送给变量 `c`,最后,由 `printf` 语句打印出结果: `max=20`。