

可下载教学资料

<http://www.tup.tsinghua.edu.cn>

21世纪普通高校计算机公共课程规划教材

C/C++语言程序设计

常东超 郭来德 刘培胜 等编著



清华大学出版社

21 世纪普通高校计算机公共课

C/C++语言程序设计

常东超 郭来德 刘培胜 等编著

清华大学出版社
北京

内 容 简 介

本书参照最新的计算机等级考试(二级 C)教学大纲及 C99 的新特性,并根据高校最新 C 语言程序设计教学大纲要求编写而成。全书共分 11 章,主要内容有 C 程序的基本组成以及程序开发过程;C 语言的基本数据类型、运算符、表达式、数据类型转换及标准的输入输出函数;C 语言的基本语句和流程控制语句;数组、函数、指针的概念及用法;C 语言的编译预处理功能;C 语言结构体与共用体、C 语言中文件的相关概念以及文件的各种操作方法;最后一章介绍了 C++ 程序设计基础和面向对象程序设计的概念。

本书既可作为高等学校本科计算机 C 语言程序设计教材,也可作为培养读者计算机编程能力和参加全国计算机等级考试(C 语言)的自学参考书。

本书配有电子教案(PPT 格式)与课后习题解答(Word 格式),联系邮箱:changdc885@126.com。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

C/C++ 语言程序设计/常东超等编著. —北京:清华大学出版社,2013 (2013.10 重印)

21 世纪普通高校计算机公共课程规划教材

ISBN 978-7-302-32084-5

I. ①C… II. ①常… III. ①C 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 082744 号

责任编辑:付弘宇 李 晔

封面设计:傅瑞学

责任校对:白 蕾

责任印制:何 芊

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座 邮 编:100084

社总机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

课 件 下 载: <http://www.tup.com.cn>, 010-62795954

印 刷 者:北京世知印务有限公司

装 订 者:三河市漂源装订厂

经 销:全国新华书店

开 本:185mm×260mm 印 张:20.5 字 数:501 千字

版 次:2013 年 10 月第 1 版 印 次:2013 年 10 月第 2 次印刷

印 数:2001~4000

定 价:34.50 元

产品编号:052693-01

出版说明

随着我国改革开放的进一步深化,高等教育也得到了快速发展,各地高校紧密结合地方经济建设发展需要,科学运用市场调节机制,加大了使用信息科学等现代科学技术提升、改造传统学科专业的投入力度,通过教育改革合理调整和配置了教育资源,优化了传统学科专业,积极为地方经济建设输送人才,为我国经济社会的快速、健康和可持续发展以及高等教育自身的改革发展做出了巨大贡献。但是,高等教育质量还需要进一步提高以适应经济社会发展的需要,不少高校的专业设置和结构不尽合理,教师队伍整体素质亟待提高,人才培养模式、教学内容和教学方法需要进一步转变,学生的实践能力和创新精神亟待加强。

教育部一直十分重视高等教育质量工作。2007年1月,教育部下发了《关于实施高等学校本科教学质量与教学改革工程的意见》,计划实施“高等学校本科教学质量与教学改革工程(简称‘质量工程’)”,通过专业结构调整、课程教材建设、实践教学改革、教学团队建设等多项内容,进一步深化高等学校教学改革,提高人才培养的能力和水平,更好地满足经济社会发展对高素质人才的需要。在贯彻和落实教育部“质量工程”的过程中,各地高校发挥师资力量强、办学经验丰富、教学资源充裕等优势,对其特色专业及特色课程(群)加以规划、整理和总结,更新教学内容、改革课程体系,建设了一大批内容新、体系新、方法新、手段新的特色课程。在此基础上,经教育部相关教学指导委员会专家的指导和建议,清华大学出版社在多个领域精选各高校的特色课程,分别规划出版系列教材,以配合“质量工程”的实施,满足各高校教学质量和教学改革的需要。

本系列教材立足于计算机公共课程领域,以公共基础课为主、专业基础课为辅,横向满足高校多层次教学的需要。在规划过程中体现了如下一些基本原则和特点。

(1) 面向多层次、多学科专业,强调计算机在各专业中的应用。教材内容坚持基本理论适度,反映各层次对基本理论和原理的需求,同时加强实践和应用环节。

(2) 反映教学需要,促进教学发展。教材要适应多样化的教学需要,正确把握教学内容和课程体系的改革方向,在选择教材内容和编写体系时注意体现素质教育、创新能力与实践能力的培养,为学生知识、能力、素质协调发展创造条件。

(3) 实施精品战略,突出重点,保证质量。规划教材把重点放在公共基础课和专业基础课的教材建设上;特别注意选择并安排一部分原来基础比较好的优秀教材或讲义修订再版,逐步形成精品教材;提倡并鼓励编写体现教学质量和教学改革成果的教材。

(4) 主张一纲多本,合理配套。基础课和专业基础课教材配套,同一门课程针对不同层次、面向不同专业的多本具有各自内容特点的教材。处理好教材统一性与多样化,基本教材与辅助教材、教学参考书,文字教材与软件教材的关系,实现教材系列资源配套。

(5) 依靠专家,择优选用。在制订教材规划时要依靠各课程专家在调查研究本课程教

材建设现状的基础上提出规划选题。在落实主编人选时,要引入竞争机制,通过申报、评审确定主题。书稿完成后要认真实行审稿程序,确保出书质量。

繁荣教材出版事业,提高教材质量的关键是教师。建立一支高水平教材编写梯队才能保证教材的编写质量和建设力度,希望有志于教材建设的教师能够加入到我们的编写队伍中来。

21 世纪普通高校计算机公共课程规划教材编委会

联系人:魏江江 weijj@tup.tsinghua.edu.cn

前言

本书是在普通高等教育“十一五”国家级规划教材《C 语言程序设计》的基础上改编的。根据计算机技术的发展和C 语言标准和编译技术、集成开发环境的变更,本书在延续《C 语言程序设计》编写风格的基础上,结合数位一线教师多年教学实践与研发经验,并考虑读者的反馈信息,对各个章节的内容、结构等进行了修订、调整、完善和补充,特别是在复杂结构的叙述方法上,作者根据多年教学实践的心得体会对教学内容进行了重新组织和编排,力求深入浅出、通俗易懂,使广大读者尽早、尽快地掌握 C 语言程序设计方法,具备一定的程序设计的能力。全书共分为 11 章,主要内容有 C 程序的基本组成以及程序开发过程; C 语言的基本数据类型、运算符、表达式、数据类型转换及标准的输入输出函数; C 语言的基本语句和流程控制语句; 数组、函数、指针的概念及用法; C 语言的编译预处理功能; C 语言结构体与共用体、C 语言中文件的相关概念以及文件的各种操作方法; 最后一章介绍了 C++ 程序设计基础和面向对象程序设计的概念。

尤其值得一提的是,编者在 C 语言语法结构等基础知识的介绍方面做了进一步的总结和归纳,从而使读者阅读此书和学习 C 语言时有章可循。本书紧紧围绕全国计算机等级考试大纲(二级与三级),采用“案例驱动”的编写方式,以基础语法、语义训练为中心,语法介绍精练,内容叙述深入浅出、循序渐进,程序案例与实际相结合、生动易懂,具有很好的启发性。每章均配备教学课件和精心设计的习题。本书配套的《C/C++ 语言程序设计实验指导与习题精选》与教材结合紧密,在精品课网站(<http://202.118.127.58/website/jpkc/dxjsj/index.html>)上可以下载电子书籍和 C 语言题库,同时在校内配备了无纸化考试软件与练习系统,以方便本校学生复习考试和上机操作,其中大容量题库及练习软件系统经过长期的测试和验证,对教学具有很高的参考价值,同时对广大读者而言也具有鲜明的指导意义。

本书除保持原有特点外,还在下面 3 个方面进行了修改和强化:

- (1) 对部分文字做了仔细修改,以使读者更容易理解。
- (2) 为了扩大读者视野和更深入掌握 C 语言程序设计的方法,本书增加了部分新内容并删改了陈旧内容,同时删改了部分习题。
- (3) 采用 C99 的标准和全面的 32 位编译环境对全书的例题进行了调试并通过。

全书常东超、郭来德、刘培胜、杨妮妮等编著,高文来、张国玉、王杨、卢紫薇、王福威、冯瑶、刘海军等参加了本书部分章节的编写和校对工作;参与实验指导的何愉高级实验师、宁获高级实验师、杨逸放实验师、刘玉山实验师等诸多人员参与校对和实例验证,同时辽宁

石油化工大学顺华能源学院王敏老师和抚顺师范高等专科学校韩云萍老师就其学院学生的特点提出了宝贵意见并参与了部分程序调试,全书由辽宁石油化工大学常东超统稿。辽宁石油化工大学魏海平教授和孙铁教授对全书进行了审阅,在此深表谢意!

限于作者水平,书中如有疏漏或不足之处,敬请读者批评指正,以利作者改进。

编 者

2013 年 5 月

目 录

第 1 章 C 语言程序设计概述	1
1.1 程序和程序设计的基本概念	1
1.2 C 语言简介	2
1.2.1 C 语言的发展历史	2
1.2.2 C 语言的特点	3
1.2.3 C 语言程序的基本结构及书写规则	3
1.2.4 C 语言的基本标识符	5
1.3 算法与程序设计	6
1.3.1 算法的基本特征	6
1.3.2 算法的基本要素	7
1.3.3 算法描述的方法	8
1.3.4 程序设计	10
习题 1	12
第 2 章 数据类型、运算符与表达式	13
2.1 C 语言的数据类型	13
2.2 整型常量与变量	13
2.2.1 常量与变量的概念	13
2.2.2 整型常量	14
2.2.3 整型变量	14
2.3 实型常量与变量	16
2.3.1 实型常量	16
2.3.2 实型变量	16
2.4 字符型常量与变量	18
2.4.1 字符常量	18
2.4.2 字符串常量	19
2.4.3 符号常量	20
2.4.4 字符型变量	20
2.5 赋值运算符和赋值表达式	22
2.6 算术运算符和算术表达式	23

2.6.1	C语言运算符简介	23
2.6.2	算术运算符和算术表达式	23
2.6.3	复合赋值运算符及表达式	25
2.6.4	各类数值型数据之间的混合运算	25
2.6.5	自增与自减运算符	28
2.7	逗号运算符和逗号表达式	29
2.8	位运算符	30
2.8.1	位运算符和位运算	30
2.8.2	位运算赋值运算符	33
2.9	变量的地址和指针型变量	34
2.9.1	变量的地址和指针型变量的概念	34
2.9.2	指针型变量的定义和指针变量的基类型	35
2.9.3	给指针变量赋值	36
2.9.4	对指针变量的操作	37
习题 2		39
第 3 章	顺序结构程序设计	44
3.1	C 语句概述	44
3.2	数据的输入/输出	47
3.2.1	字符输入/输出函数	47
3.2.2	格式输入/输出函数	48
3.3	程序举例	58
习题 3		61
第 4 章	分支结构程序设计	64
4.1	关系运算符和关系表达式	64
4.1.1	关系运算符	64
4.1.2	关系表达式	65
4.2	逻辑运算符和逻辑表达式	66
4.2.1	逻辑运算符	66
4.2.2	逻辑表达式	67
4.3	if 语句以及用 if 语句构成的分支结构	67
4.3.1	if 语句的两种基本形式	68
4.3.2	嵌套的 if 语句	70
4.3.3	条件表达式构成的分支结构	73
4.4	switch 语句	75
4.4.1	switch 语句及用 switch 语句构成的分支结构	75
4.4.2	在 switch 语句体中使用 break 语句	76
4.5	程序举例	77

习题 4	80
第 5 章 循环结构程序设计	84
5.1 while 语句以及用 while 语句构成的循环结构	84
5.1.1 while 循环的一般形式	84
5.1.2 while 循环的执行过程	84
5.2 do...while 语句以及用 do...while 语句构成的循环结构	87
5.2.1 do...while 语句构成的循环结构	87
5.2.2 do...while 循环的执行过程	87
5.3 for 语句以及用 for 语句构成的循环结构	90
5.3.1 for 语句构成的循环结构	90
5.3.2 for 循环的执行过程	91
5.3.3 有关 for 语句的说明	91
5.4 break 语句和 continue 语句在循环结构中的应用	94
5.4.1 break 语句	94
5.4.2 continue 语句	94
5.5 循环的嵌套	95
5.6 3 种循环的比较	98
5.7 程序举例	99
习题 5	102
第 6 章 数组与指针	108
6.1 一维数组	108
6.1.1 一维数组的定义	108
6.1.2 一维数组元素的引用	109
6.1.3 一维数组的初始化	112
6.1.4 一维数组程序举例	112
6.2 二维数组	114
6.2.1 二维数组的定义	114
6.2.2 二维数组元素的引用	115
6.2.3 二维数组的初始化	116
6.2.4 二维数组程序举例	117
6.3 字符数组和字符串	119
6.3.1 字符数组	119
6.3.2 字符串	120
6.3.3 字符串的输入输出	122
6.3.4 字符串处理函数	123
6.3.5 程序举例	126
6.4 数组和指针	129

6.4.1	一维数组和指针	129
6.4.2	二维数组的地址	132
6.4.3	指向二维数组的指针变量	134
6.4.4	指针数组的定义和应用	136
6.5	字符串和指针	138
6.5.1	单个字符串的处理方法	138
6.5.2	多个字符串的处理方法	140
6.6	指向指针的指针	142
	习题 6	144
第 7 章	函数与指针	146
7.1	概述	146
7.2	函数的定义	146
7.3	函数的参数和函数的值	148
7.3.1	形式参数和实际参数	148
7.3.2	函数的返回值	150
7.4	函数的调用	151
7.4.1	函数的简单调用	151
7.4.2	函数的嵌套调用	153
7.4.3	函数的递归调用	155
7.5	函数与指针	159
7.5.1	指针变量作为函数参数	159
7.5.2	数组作为函数参数	159
7.5.3	返回指针值的函数	162
7.5.4	指向函数的指针	163
7.6	有关指针的数据类型和指针运算的小结	165
7.6.1	有关指针的数据类型的小结	165
7.6.2	指针运算的小结	166
7.6.3	void 指针类型	166
7.7	变量的作用域	166
7.7.1	局部变量	167
7.7.2	全局变量	168
7.8	变量的存储类别	169
7.8.1	动态存储方式与静态存储方式	169
7.8.2	auto 变量	170
7.8.3	用 static 声明局部变量	170
7.8.4	register 变量	171
7.8.5	用 extern 声明外部变量	172
7.9	内部函数和外部函数	173

*7.10	main 函数的参数	173
	习题 7	175
第 8 章	编译预处理	184
8.1	宏定义	184
8.1.1	无参宏定义	184
8.1.2	带参宏定义	186
8.2	文件包含	189
8.3	条件编译	190
	习题 8	192
第 9 章	结构体与共用体	194
9.1	结构体类型的定义	194
9.2	结构体类型变量	195
9.2.1	结构体变量的定义	195
9.2.2	结构体变量的引用	197
9.2.3	结构体变量的初始化	199
9.2.4	结构体变量的输入与输出	199
9.3	结构体类型数组	200
9.3.1	结构体数组的定义	200
9.3.2	结构体数组的初始化	201
9.3.3	结构体数组的引用	202
9.4	结构体类型指针	203
9.4.1	指向结构体变量的指针	203
9.4.2	指向结构体数组的指针	205
9.5	结构体与函数	206
9.5.1	结构体变量作为函数参数	206
9.5.2	指向结构体变量的指针作为函数参数	207
9.5.3	函数的返回值为结构体类型	209
9.6	链表	210
9.6.1	链表概述	211
9.6.2	处理动态链表所需的函数	213
9.6.3	链表的基本操作	214
9.7	共用体	222
9.7.1	共用体类型与共用体变量	222
9.7.2	共用体变量的引用	224
9.7.3	共用体变量的应用	225
9.8	枚举类型	226
9.9	用 typedef 定义类型	229

习题 9	231
第 10 章 文件	233
10.1 文件概述	233
10.1.1 数据文件	233
10.1.2 文件的存取方式	234
10.1.3 文件指针类型	234
10.1.4 文件操作的步骤	234
10.2 文件的打开与关闭	235
10.2.1 文件的打开(fopen 函数)	235
10.2.2 文件的关闭(fclose 函数)	236
10.3 文件的读写	237
10.3.1 字符读写函数 fgetc 和 fputc	237
10.3.2 字符串读写函数 fgets 和 fputs	239
10.3.3 数据块读写函数 fread 和 fwrite	240
10.3.4 格式化读写函数 fscanf 和 fprintf	242
10.4 文件的随机读写	243
10.4.1 文件定位	243
10.4.2 文件的随机读写	244
10.5 文件检测函数	245
习题 10	245
第 11 章 C++ 基础	247
11.1 C++ 概述	247
11.2 C++ 对 C 的扩充	249
11.2.1 C++ 的输入输出	249
11.2.2 C++ 的行注释	252
11.2.3 const 常量定义与使用	253
11.2.4 局部变量的定义与全局变量作用域运算符	255
11.2.5 变量的引用	256
11.2.6 函数重载	259
11.2.7 带默认参数的函数	260
11.2.8 内联函数	261
11.2.9 new 和 delete 算符	263
11.3 C++ 面向对象程序设计	264
11.3.1 类与对象	265
11.3.2 构造函数与析构函数	269
11.3.3 静态成员	275
11.3.4 友元	277

11.3.5 运算符重载	279
11.3.6 继承与派生	282
11.3.7 多态性与虚函数	286
11.4 应用举例	292
习题 11	296
附录 A 常用字符与 ASCII 码对照表	297
附录 B C99 标准的新特性	298
附录 C Turbo C 常用标准库函数	300
参考文献	314

1.1 程序和程序设计的基本概念

目前计算机的应用已经深入到社会的各个领域,成为人们生活、学习和工作的必备工具,顾名思义,早期的计算机是用来计算的机器,但是目前的计算机尤其是微型计算机已不再是简单的计算机器了,而是具有强大的存储和计算能力、由程序自动控制的智能化电子设备。人与人之间交流需要语言,那么人与机器要进行交流也需要“语言”,这种“语言”就是“程序”。程序来自生活,通常指完成某一事务的既定方式和过程。比如,我们常说做什么事情需要什么样的程序,在日常生活中可以看成是对一系列执行过程的描述。以学生去食堂打饭为例,为了完成这件事情需要几个步骤呢?首先,带上饭卡去食堂;其次,到相应的窗口去排队;然后,挑选饭菜并刷卡;接着,食堂职工负责盛取饭菜相应事宜;最后,离开食堂窗口,这样就完成了去食堂打饭的程序。

那么计算机中的程序是怎样定义的呢?是指为计算机执行某些操作,或解决某个问题而编写的一系列有序指令的集合。这里涉及“指令”的概念,以日常生活中的指令来解释,比如,办公室里的老板对秘书发出指令,需要秘书做口述笔记,输入信函内容、发传真等,那秘书就需要按照指示逐步完成老板发出的每一条指令,最终完成老板交代的事情。在计算机中,就是由程序员对计算机发出指令,想让计算机解决某个问题,就可将解决问题的过程用计算机能够接受的方式或者选择某一种计算机语言将它一步一步地描述出来,计算机就会按照预先存储在它里面的代码逐步去执行,这就是计算机中的指令,而指令的集合就构成了计算机中的程序,编写程序的过程就称为“程序设计”。

计算机程序就是人与计算机交流的“语言”,也就是程序设计语言。正如人与人交流有不同的语言一样,程序设计语言也有很多种,基本上分为高级语言和低级语言两大类。目前常见的高级语言有 Visual Basic、C++、Java、C 等,这些语言都是用接近人们习惯的自然语言和数学语言作为表达形式的,使人们学习和操作起来感到十分方便。但是,对于计算机本身来说,它并不能直接识别由高级语言编写的程序,它只能接受和处理由 0 和 1 构成的二进制指令或数据。由于这种形式的指令是面向机器的,因此也称为“机器语言”。

我们把由高级语言编写的程序称为“源程序”,把由二进制代码表示的程序称为“目标程序”。为了把源程序转换成机器能接受的目标程序,软件工作者编制了一系列软件,通过这些软件可以把用户按规定语法写出的语句一一翻译成二进制机器指令。这种具有翻译功能的软件称为“编译程序”,每种高级语言都有与它对应的编译程序。例如,C 语言编译程序就是这样的一种软件,其功能如图 1.1 所示。

无论采用哪种语言编写程序,经过编译(Compile)最终都将转换成二进制的机器指令。

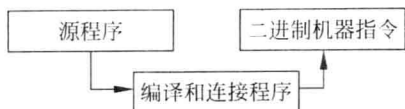


图 1.1 C 语言编译程序功能示意图

C 源程序经过 C 编译程序编译之后生成一个后缀为 .OBJ 的二进制文件(称为目标文件),然后由称为“连接程序”(link)的软件,把此 .OBJ 文件与 C 语言提供的各种库函数连接起来生成一个后缀为 .EXE 的可执行文件。在操作系统环境下,只需单击或输入此文件的

的名字,该可执行文件就可运行。

1.2 C 语言简介

1.2.1 C 语言的发展历史

C 语言是当今社会应用广泛,并受到众多用户欢迎的一种计算机高级语言。它既可用于编写系统软件,也可用来编写应用软件。

早期的系统软件设计均采用汇编语言,例如,大家熟知的 UNIX 操作系统。尽管汇编语言在可移植性、可维护性和描述问题的效率等方面远远不及高级程序设计语言,但是一般的高级语言有时难以实现汇编语言的某些功能。那么,能否设计出一种集汇编语言与高级语言的优点于一身的语言呢?这种思路促成了 UNIX 系统的开发者——美国贝尔实验室的 Ken Thompson,他在 1970 年设计出了既简单又便于硬件操作的 B 语言,并用 B 语言写了第一个在 PDP-7 上实现的 UNIX 操作系统。

以历史发展的角度看,C 语言起源于 1968 年发表的 CPL 语言(Combined Programming Language),它的许多重要思想来自于 Martin Richards 在 1969 年研制的 BCPL 语言,以及以 BCPL 语言为基础的 B 语言。Dennis M. Ritchie 在 B 语言的基础上,于 1972 年研制了 C 语言,并用 C 语言写成了第一个在 PDP-11 计算机上实现的 UNIX 操作系统(主要在贝尔实验室内部使用)。以后,C 语言又经过多次改进,直到 1975 年用 C 语言编写的 UNIX 操作系统第 6 版公诸于世后,C 语言才举世瞩目。1977 年出现了独立于机器的 C 语言编译文本《可移植 C 语言编译程序》,从而大大简化了把 C 语言编译程序移植到新环境所需做的工作,这本身也就使 UNIX 操作系统迅速地在众多的机器上实现。例如 VAX、AT&T 等计算机系统都相继开发了 UNIX。随着 UNIX 的日益广泛使用,C 语言也迅速得到推广。1978 年以后,C 语言先后移植到大、中、小、微型计算机上,它的应用领域已不再限于系统软件的开发,而成为当今最流行的程序设计语言之一。

以 1978 年发布的 UNIX 第 7 版的 C 语言编译程序为基础,Brian W. Kernighan 和 Dennis M. Ritchie 合著了影响深远的名著 *The C Programming Language*,这本书中介绍的 C 语言成为后来广泛使用的 C 语言版本的基础,它被称为标准 C。

1983 年美国国家标准化协会(ANSI)根据 C 语言问世以来的各种版本,对 C 语言的发展和扩充制定了新的标准,称为 ANSI C。1987 年 ANSI 又公布了新标准,称为 87 ANSI C。

目前流行的 C 语言编译系统大多是以 ANSI C 为基础进行开发的,不同版本的 C 编译系统所实现的语言功能和语法规则基本部分是相同的,但在有关规定上又略有差异。本书的叙述基本上以 ANSI C 为基础。

1.2.2 C 语言的特点

C 语言是一种通用的程序设计语言。C 语言的通用性和无限制性使得它对于程序设计者来说显得更加通俗、更加有效。目前,C 语言程序设计已应用于各个领域,无论设计系统软件(操作系统、编译系统)或应用软件(图形处理),还是在数据处理(如企业管理)或数值计算等方面都可以很方便地使用 C 语言。

C 语言与其他高级语言相比具有以下特点:

(1) C 语言具有结构语言的特点,程序之间很容易实现段的共享。它具有结构化的流程控制语句(如 if...else 语句、switch 语句、while 语句、do-while 语句、for 语句等),支持若干种循环结构,允许编程者采用缩进书写形式编程。因此,用 C 语言设计出的程序层次结构清晰。

(2) C 语言的主要结构成分为函数,函数可以在程序中被定义完成独立的任务,独立地编译成代码,以实现程序的模块化。

(3) C 语言运算符丰富,运算符包含的范围很广泛。C 语言把赋值、括号、强制类型转换都当作运算符处理。灵活地使用各种运算符可以实现在其他的高级语言中难以实现的运算。

(4) C 语言数据类型丰富。数据类型有整型、实型、字符型、数组型、指针型、结构体型、共用体型等。能用来实现各种复杂的数据结构(如链表、树、栈等)的运算。尤其是 C 语言的指针型数据的运算,更是灵活、多样。

(5) C 语言允许直接访问物理地址,即可直接对硬件进行操作,实现汇编语言的大部分功能。C 语言的这一特点,使得它成为编制系统软件的基本语言(UNIX 的绝大部分就是由 C 语言写成的)。

(6) C 语言语法限制不太严格,程序设计自由度大。这样使 C 语言能够减少对程序员的束缚。“限制”与“灵活”是一对矛盾。限制严格,就易失去灵活性;而强调灵活,就必然放松限制。从这个角度来看,使用 C 语言编程,要求编程者对程序设计技巧要更加熟练一些。

(7) 用 C 语言编程,生成的目标代码质量高,程序执行效率高。同时用 C 语言编写的程序可移植性好。

(8) C 语言很容易被其他领域接受。单片机领域首选 C 语言作为开发工具,嵌入式系统和数字信号处理(Digital Signal Processing, DSP)领域也将 C 语言作为自己的开发工具。

C 语言优点很多,但是它也存在一些缺点,如运算优先级太多,不便记忆;数值运算能力方面不像其他高级语言那样强,语法限制不严格等。尽管 C 语言目前还存在一些不足之处,但由于它目标代码质量高、使用灵活、数据类型丰富、可移植性好而得到广泛的普及和迅速的发展,成为一种深受广大用户欢迎的程序设计语言,同时也是一种在系统软件开发、科学计算、自动控制等各个领域被广泛应用的程序设计语言。

1.2.3 C 语言程序的基本结构及书写规则

1. C 语言程序的基本结构

在学习 C 语言的具体语法之前,我们先通过一个简单的 C 语言程序示例,初步了解 C 语言程序的基本结构。