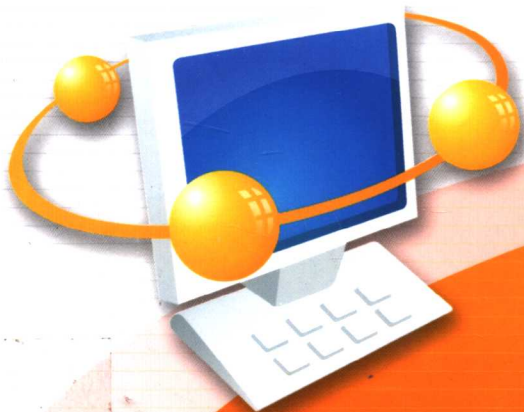




普通高等教育“十一五”国家级规划教材

C语言程序设计 与实训

周屹 万静 ○ 主编



 **机械工业出版社**
CHINA MACHINE PRESS



赠电子课件

TP312/2869

2008

普通高等教育“十一五”国家级规划教材

C 语言程序设计与实训

主 编 周 屹 万 静

副主编 黄成哲 李 朴

参 编 郑国华 王丁 运海红

机械工业出版社

本教材是为高等院校第一门程序设计课程而编写的教材。全教材分2个部分,第1部分是基础知识部分,详细地介绍了C语言概述、基本数据类型、运算符和表达式、顺序结构、选择结构、循环控制、数组、函数、编译预处理、指针、结构和其他类型、文件等程序设计内容。第2部分是实训部分,重点介绍了编译器的使用,一些常用基本算法和Windows程序设计。

本教材层次清晰、系统全面、例题丰富、实用性强、面向应用、注重培养应用技能和能力。本教材适合作为普通高等院校应用型本科非计算机专业学生的程序设计教材,也可以作为计算机专业本科、专科学生学习计算机语言的入门教材。

本教材配有电子教案及习题答案,凡使用本书作为教材的教师可登录机械工业出版社教材服务网 www.cmpedu.com 下载。咨询邮箱: cmpgaozhi@sina.com, 咨询电话: 010-88379375。

图书在版编目(CIP)数据

C语言程序设计与实训/周屹,万静主编. —北京:机械工业出版社, 2008.2

普通高等教育“十一五”国家级规划教材

ISBN 978-7-111-23296-4

I. C… II. ①周…②万… 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(2008)第005209号

机械工业出版社(北京市百万庄大街22号 邮政编码100037)

责任编辑:王玉鑫 版式设计:冉晓华 责任校对:陈延翔

封面设计:王伟光 责任印制:杨曦

北京机工印刷厂印刷(兴文装订厂装订)

2008年3月第1版第1次印刷

169mm×239mm·8.75印张·387千字

0 001—4 000册

标准书号:ISBN 978-7-111-23296-4

定价:24.00元

凡购本书,如有缺页、倒页、脱页,由本社发行部调换

销售服务热线电话:(010) 68326294

购书热线电话:(010) 88379639 88379641 88379643

编辑热线电话:(010) 88379543

封面无防伪标均为盗版

前 言

在众多的程序设计语言中，C 语言程序设计以其灵活性和实用性受到计算机应用人员的广泛喜爱，C 语言在面向应用教育类型的教学过程中，不但要注重知识的讲授，更要强调基本技能的训练，不仅要让学生学习程序设计的基本概念和方法、掌握编程技术，更重要的是培养学生分析和解决问题的能力、培养学生灵活应用编程思想和方法的能力。

本教材的培养目的是学生首先掌握程序设计入门，通过讲述程序结构、语言集成编程环境，通过对简单操作界面的设计，学习如何设计友好的人机界面接口，初步了解程序设计的概念和培养上机操作、调试程序的能力，使学生对程序设计初步建立起整体认识；其次培养学生的实训能力，全书给出大量的引例和实例，以趣味性的案例引导学生实训。从实例中导出 C 语言的基本知识点，淡化 C 语言的学科系统性；通过案例教学，将 C 语言作为解决实际问题的工具，强化学生的编程能力。

全书由 2 部分构成：第 1 部分是基础知识，第 2 部分是实训。课堂讲授时可根据学生及专业情况对内容酌情取舍，由于学时的限制，本书的第 2 部分可作为学生课外练习或课程设计内容，同时可对学生作不同层次的要求。在内容安排上，本教材也考虑了与普通教材的兼容性，并在实用性方面又作了补充。本教材适合作为普通高等院校应用型本科非计算机专业学生的程序设计教材，也可以作为大学计算机专业本科、专科学生学习计算机语言的入门教材。

本教材由长期承担程序设计基础课教学、具有丰富教学经验的一线教师编写，根据作者多年从事程序设计课程教学活动以及软件开发的经验，针对多数学生的认知规律，在详细介绍、解析 C 语言语法知识的同时，教材内容充分考虑到知识和技能的结合，由浅入深、循序渐进，把枯燥的概念、语法、算法融汇在生动、有趣的案例中，以调动学生的学习积极性。本教材还配有多媒体电子教案、PPT、习题答案等教学资源。

本教材由周屹、万静主编，黄成哲、李朴任副主编。第 1、8、10 章由周屹编写，第 2、4、5 章由万静编写，第 3 章及附录由黄成哲编写，第 6、12 章由郑国华编写，第 7、9 章由王丁编写，第 11 章由运海红编写，第 13~15 章由李朴编写，最后由周屹统稿，王培东教授主审。

本教材的编写得到作者所在的黑龙江工程学院计算机科学与技术系的大力支持，在此对所有人的工作与支持表示衷心的感谢。

由于编者的水平有限，书中难免存在一些缺点和错误，敬请广大读者批评指正。编者 E-mail: zhouyi_1@163.com。

编者

目 录

前言

第1部分 基础知识部分 1

第1章 C语言概述 1

1.1 C语言演变 1

1.2 C语言特点 1

1.3 C语言编程概述 3

1.4 C语言编程环境简介 7

1.4.1 MS DOS 编程环境 7

1.4.2 Windows 编程环境 8

1.4.3 UNIX 编程环境 9

本章小结 10

习题 11

第2章 基本数据类型 12

2.1 常量和变量 13

2.2 数据类型 14

2.2.1 整型数据类型 14

2.2.2 实型数据类型 17

2.2.3 字符型数据类型 18

2.2.4 字符串 21

2.3 数据类型转换的规则 21

本章小结 22

习题 23

第3章 运算符和表达式 25

3.1 算术运算符和算术表达式 25

3.2 赋值运算符和赋值表达式 27

3.3 自增和自减运算 28

3.4 关系运算和逻辑运算 29

3.4.1 关系运算符 29

3.4.2 逻辑运算符 30

3.5 条件运算和逗号运算 31

3.6 位操作运算 33

3.7 其他运算符 35

3.7.1 sizeof 运算符 35

3.7.2 "&"和"*"运算符 35

3.8 运算符优先级 36

本章小结 37

习题 37

第4章 顺序结构 40

4.1 三种基本结构 40

4.2 赋值语句 41

4.3 表达式语句和函数调用语句 42

4.4 复合语句和空语句 42

4.5 格式输入和输出函数 43

4.5.1 标准格式输入和输出函数 43

4.5.2 字符输入和输出函数 50

本章小结 51

习题 52

第5章 选择结构 54

5.1 if 语句 54

5.1.1 基本形式 54

5.1.2 if-else 形式 55

5.1.3 if-else-if 形式 56

5.2 if 语句的嵌套 58

5.3 switch 语句 59

本章小结 63

习题 63

第6章 循环控制 65

6.1 循环结构 65

6.1.1 while 循环 65

6.1.2 do-while 循环 67

6.1.3 for 循环 68

6.2 循环结构嵌套 71

6.3 转向语句 72

6.3.1 break 语句 72

6.3.2 continue 语句 74

6.3.3 goto 语句 75

6.3.4 return 语句	75	8.9 函数的嵌套与递归	129
6.4 循环应用	76	8.9.1 函数的嵌套调用	129
6.4.1 递推法	76	8.9.2 函数的递归调用	130
6.4.2 迭代法	79	本章小结	133
6.4.3 枚举法	80	习题	134
6.4.4 其他应用	81	第9章 编译预处理	135
本章小结	83	9.1 宏定义	135
习题	84	9.1.1 符号常量宏定义	135
第7章 数组	86	9.1.2 带参数宏定义	137
7.1 一维数组	86	9.2 文件包含	140
7.1.1 初始化	86	9.3 条件编译	142
7.1.2 下标引用	88	本章小结	144
7.2 多维数组	90	习题	144
7.2.1 二维数组	90	第10章 指针	146
7.2.2 多维数组	95	10.1 指针概念	146
7.2.3 多维数组应用	95	10.2 指针变量定义和引用	147
7.3 字符数组和字符串	96	10.2.1 指针变量定义	147
7.3.1 字符串和字符串结束标 志	96	10.2.2 指针变量赋值	148
7.3.2 字符串处理函数	98	10.2.3 指针变量运算	148
本章小结	101	10.3 指针与数组	151
习题	102	10.3.1 数组指针变量	151
第8章 函数	104	10.3.2 指针与一维数组	152
8.1 函数的定义	105	10.3.3 指针与二维数组	153
8.2 函数的形式参数和实际参数	108	10.3.4 指针数组	155
8.3 函数的返回值	112	10.4 指针与函数	157
8.4 函数的调用和传值	112	10.4.1 指针作为函数形参	157
8.4.1 函数的调用	113	10.4.2 指针作为函数返回值	160
8.4.2 被调函数的传值	114	10.4.3 函数指针	161
8.5 内部函数和外部函数	115	10.4.4 指针型函数	162
8.6 数组作为函数的参数	116	10.5 指针与字符串	164
8.7 局部变量和全局变量	118	10.5.1 字符串表示方法	164
8.7.1 局部变量	119	10.5.2 字符串处理函数的实现	166
8.7.2 全局变量	120	10.5.3 字符型指针数组	169
8.8 存储类型	122	10.6 多重指针	171
8.8.1 auto 存储类型	123	10.6.1 指向指针的指针	171
8.8.2 extern 存储类型	124	10.6.2 命令行参数	172
8.8.3 register 存储类型	125	本章小结	172
8.8.4 static 存储类型	126	习题	173
		第11章 结构和其他类型	175

11.1 结构概念.....	175	12.6 出错检测函数.....	214
11.1.1 结构定义.....	175	本章小结	214
11.1.2 结构类型变量说明.....	176	习题	215
11.2 结构操作.....	177	第 2 部分 实训部分	218
11.2.1 结构初始化.....	177	第 13 章 编译器	218
11.2.2 结构分量访问.....	178	13.1 Turbo C 编译器使用	218
11.2.3 结构数组.....	179	13.2 UNIX 编译器 cc 使用	228
11.2.4 结构指针变量.....	181	13.3 Visual C++ 编译器使用.....	229
11.3 在函数中使用结构.....	183	本章小结	232
11.4 动态结构类型.....	186	习题	233
11.5 联合.....	192	第 14 章 基本算法	234
11.5.1 联合定义.....	192	14.1 链表.....	234
11.5.2 联合变量说明.....	193	14.2 队列.....	238
11.5.3 联合变量赋值和使用.....	193	14.3 栈.....	241
11.6 枚举类型.....	195	14.4 表达式求值.....	243
11.6.1 枚举类型定义.....	195	14.4.1 波兰记法.....	243
11.6.2 枚举类型操作.....	195	14.4.2 算数表达式求值.....	244
11.7 使用 typedef	197	14.5 迷宫问题.....	246
本章小结	198	本章小结	252
习题	198	习题	252
第 12 章 文件	201	第 15 章 Windows 程序设计	253
12.1 C 语言文件系统概述.....	201	15.1 从 C 语言到 C++ 语言	253
12.2 文件类型指针.....	202	15.2 用 AppWizard 生成鼠标追踪程	
12.3 文件打开与关闭.....	203	序.....	256
12.3.1 文件打开函数.....	203	本章小结	259
12.3.2 文件关闭函数.....	205	习题	259
12.4 文件读写.....	205	附录	261
12.4.1 字符读写函数.....	205	附录 A 常用 C 语言标准库函数	261
12.4.2 字符串读写函数.....	208	附录 B ASCII 字符集	268
12.4.3 数据块读写函数.....	210	附录 C 运算符的优先级和结合性.....	269
12.4.4 格式化读写函数.....	211	参考文献	271
12.5 文件定位.....	212		

第 1 部分 基础知识部分

第 1 章 C 语言概述

1.1 C 语言演变

C 语言是由 UNIX 的研制者丹尼斯·里奇 (Dennis Ritchie) 和肯·汤普逊 (Ken Thompson) 设计的。C 语言第一次发展在 1969 ~ 1973 年之间。C 语言的很多特性是由一种早期的、被称为 B 语言的编程语言发展而来的。B 语言由更早的编程语言 BCPL (Basic Combind Programming Language) 发展演变而来。

1970 年, AT&T 贝尔实验室的 Ken Thompson 根据 BCPL 语言设计出较先进的并取名为 B 的语言, 最后导致了 C 语言的问世。随着微型计算机的日益普及, 出现了许多 C 语言版本。

1973 年, C 语言已经可以用来编写 UNIX 操作系统的内核, 这是第一次用 C 语言来编写操作系统的内核。Dennis Ritchie 和 Brian Kernighan 在 1978 年出版了《C 程序设计语言》(The C Programming Language, 经常简称为“白皮书”或“K&R”)。

1980 年以后, 贝尔实验室使得 C 语言变得更为广泛地流行, C 语言一度成为操作系统和应用程序编程的首选。甚至到今天, 它仍被用于编写操作系统以及作为广泛的计算机教育的语言。

1983 年, 美国国家标准委员会 (ANSI) 对 C 语言进行了标准化, 于 1983 年颁布了第一个 C 语言标准草案 (83 ANSI C), 后来于 1987 年又颁布了另一个 C 语言标准草案 (87 ANSI C)。最新的 C 语言标准是在 1999 年颁布并在 2000 年 3 月被 ANSI 采用的 C99。20 世纪 80 年代晚期, 布贾尼·斯特劳斯特卢普 (Bjarne Stroustrup) 和贝尔实验室为 C 语言添加了面向对象的特性, C 语言扩展出 C++ 语言。目前 C++ 语言广泛应用于 Microsoft Windows 下运行的商业应用程序的编制, 然而 C 语言仍然是 UNIX 世界的热门编程语言。

1.2 C 语言特点

C 语言广泛应用于不同的操作系统, 如 UNIX、MS-DOS、Microsoft Windows 及 Linux 等。C 语言是一种面向过程的语言, 同时具有高级语言和汇编语言的优点。

在 C 语言的基础上发展起来的有支持多种程序设计风格的 C++ 语言，网络上广泛使用的 Java、JavaScript、微软的 C# 等。

C 语言发展迅速，而且成为最受欢迎的语言之一，主要因为它具有强大的功能。许多著名的系统软件，如 DBASE III PLUS、DBASE IV 都是由 C 语言编写的。使用 C 语言加上一些汇编语言子程序，就更能显示 C 语言的优势了，像 PC-DOS、WORDSTAR 等就是用这种方法编写的。归纳起来，C 语言具有下列特点。

1. C 语言是中级语言

它把高级语言的基本结构和语句与低级语言的实用性结合起来。C 语言可以像汇编语言一样对位、字节和地址进行操作，而这三者是计算机最基本的工作单元。

2. C 语言是结构化语言

结构化语言的显著特点是代码及数据的分隔化，即程序的各个部分除了必要的信息交流外彼此独立。这种结构化方式可使程序层次清晰，便于使用、维护以及调试。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，并具有多种循环和条件语句实现控制程序流向，从而使程序完全结构化。

3. C 语言功能齐全

C 语言具有各种各样的数据类型，并引入了指针概念，可使程序效率更高。另外 C 语言也具有强大的图形功能，支持多种显示器和驱动器，而且计算功能、逻辑判断功能也比较强大。

4. C 语言适用范围大

C 语言还有一个突出的优点就是适合于多种操作系统，同时也适用于多种机型。由于 C 语言实现了对硬件的编程操作，因此 C 语言集高级语言和低级语言的功能于一体。既可用于系统软件的开发，也适合于应用软件的开发。此外，C 语言还具有效率高，可移植性强等特点。因此广泛地移植到了各种类型的计算机上，从而形成了多种版本的 C 语言。

5. C 语言的语法特点

1) C 语言简洁、紧凑，使用方便、灵活。ANSI C 一共有 32 个关键字（见表 1-1），9 种控制语句，程序书写自由，主要用小写字母表示，压缩了一切不必要的成分。

表 1-1 关 键 字

auto	break	case	char	const	continue	default
do	double	else	enum	extern	float	for
goto	if	int	long	register	return	short
signed	static	sizeof	struct	switch	typedef	union
unsigned	void	volatile	while			

Turbo C 扩充了 11 个关键字：

asm _cs _ds _es _ss cdecl far huge interrupt near pascal

注意：在 C 语言中，关键字都是小写的。

2) 运算符丰富，共有 34 种。C 语言把括号、赋值、逗号等都作为运算符处理，从而使 C 语言的运算类型极为丰富，可以实现其他高级语言难以实现的运算。

3) 数据结构类型丰富。

4) 具有结构化的控制语句。

5) 语法限制不太严格，程序设计自由度大。

6) C 语言允许直接访问物理地址，能进行位 (bit) 操作，能实现汇编语言的大部分功能，可以直接对硬件进行操作。因此有人把它称为中级语言。

6. C 语言生成目标代码质量高，程序执行效率高

与汇编语言相比，用 C 语言写的程序可移植性好。程序员用 C 语言写的程序会感到限制少、灵活性大、功能强，但 C 语言对程序员要求也高，学习起来较其他高级语言要困难一些。

目前较为流行的 C 语言有几个版本：Microsoft C 或称 MS C、Borland Turbo C 或称 Turbo C、AT&T C，这些 C 语言版本不仅实现了 ANSI C 标准，而且在此基础上各自作了一些扩充，使之更加方便、完美。

在 C 语言的基础上，1983 年又由贝尔实验室的 Bjarne Stroustrup 推出了 C++ 语言。C++ 语言进一步扩充和完善了 C 语言，成为一种面向对象的程序设计语言。C++ 语言目前流行的最新版本是 Borland C++ 4.5、Symantec C++ 6.1 和 Microsoft Visual C++ 6.0。C 语言是 C++ 语言的基础，C++ 语言和 C 语言在很多方面是兼容的，因此，掌握了 C 语言，再进一步学习 C++ 语言就能以一种熟悉的语法来学习面向对象的语言，从而达到事半功倍的目的。

1.3 C 语言编程概述

C 语言程序设计是结构化程序设计方法，利用 C 语言环境实现算法。结构化程序设计方法是从 20 世纪 70 年代以来逐步形成的一种新程序设计方法，它要求程序员按照一定的规范、采用成熟的设计方法进行程序设计，而不允许随心所欲地编写程序。它强调程序的风格，程序结构的规范化以及自顶向下、逐步细化和模块化的设计方法，追求的主要目标是提高程序的易读性和易维护性。

编写一个程序有两种不同的思考问题的方法：一种称“自顶向下”的方法，另一种称“自下向上”的方法。“自顶向下”的方法是先着眼于全局，然后逐步解决细节问题。“自下向上”的方法则相反，而是从具体细节着手。程序设计要坚持采用“自顶向下”的方法。

逐步细化方法是在“自顶向下”方法原则下，将问题求解分步（分层次）进行，逐层表示“做什么”与“怎么做”问题，直到整个问题全部描述清楚为止。

逐步细化又有两种方法。

(1) 着眼于整个问题的自然法 这种方法首先采用较为自然的抽象语言表示“做什么”，得到级别最高的抽象程序。然后，对抽象程序进一步分解，得到低一级的抽象程序。继续进行这样的细化过程，直到整个问题都能用程序设计语言的语句全部描述出来为止。它仅适用于简单问题的求解。在讨论三种基本程序结构时，所举例题的流程图和利用自然语言描述的算法就是这种逐步细化方法的简单实例。

(2) 基于函数的逐步细化法 这种方法是用函数调用（函数语句或函数命名符）表示“做什么”，而用函数定义表示“怎么做”。用这种方法求解问题，首先将问题或任务分解为一些子任务，用函数调用表示“做什么”，形成第1层。第0层是主函数，级别最高。每个子任务还可分解为一些子任务，也用函数调用表示“做什么”，并建立与上层的逻辑关系，形成第2层。继续进行这样的细化过程，形成结构清晰、层次分明的程序结构。它广泛用于复杂问题的求解。

结构化程序设计方法以算法为核心，数据结构和算法是两个独立的整体。

程序的核心是算法。算法是描述求解问题方法的操作步骤集合。早在计算机发明以前，算法就是数学家的工具，数学家用算法来描述特定问题的解决方法。开发出来的算法是以人能理解的语言描述的，为了让计算机能接受算法，计算机必须具有自己的语言系统。计算机能理解的语言被称作程序设计语言，程序设计语言规定了书写程序可使用的一组记号和一组语法规则。程序是处理特定问题的计算机可识别的步骤集合，是用程序设计语言表示出来的算法。程序 = (算法) + (数据结构)。

为了说明 C 语言源程序结构的特点，先看以下几个程序。这几个程序由简到难，表现了 C 语言源程序在组成结构上的特点。虽然有关内容还未介绍，但可从这些例子中了解到组成一个 C 语言源程序的基本组成和书写格式。

例 1-1 在屏幕上显示“Hello, C!”。

```
#include <stdio.h>

void main()                /* 主函数 */
{
    printf("Hello, C!\n");  /* 输出语句 */
}
```

main()是主函数的函数名，表示这是一个主函数。每一个 C 语言源程序都必须有且只能有一个主函数。一对花括号“{ }”是主函数的定界符，在主函数中只有一个语句；在该语句中调用了格式输出库函数 printf(), 用于向屏幕上输出一个字符串。函数 printf 的功能是把要输出的内容送到标准输出设备上，默认的标准输出设备是显示器。由“/*”和“*/”括起来的任何文字是注释，程序不执行注释部分。语句用分号结束，一行可以写多个语句，关键字用小写字母，书写采用自由格式。

主函数体分为两部分，一部分为说明部分，另一部分为执行部分。说明是指变

量的类型说明。例 1-1 中未使用任何变量，因此无说明部分。

C 语言规定，源程序中所有用到的变量都必须先说明，后使用，否则将会出错。这一点是编译型高级程序设计语言的一个特点，说明部分是 C 语言源程序结构中很重要的组成部分。

例 1-2 输出两个变量的和。

```
#include <stdio.h>

void main()
{
    int x,y;                /* 变量定义语句:定义 2 个整型变量 x、y */
    x = 3;                  /* 可执行的赋值语句:将 3 赋值给变量 x */
    y = 6;                  /* 可执行的赋值语句:将 6 赋值给变量 y */
    printf("%d\n",x+y);
}
```

例 1-2 中使用了两个变量 x、y，用类型说明符 int 来说明这两个变量。说明部分后的三行为执行部分或称为执行语句部分，用以完成程序的功能。执行部分的第一、二行是赋值语句，printf 函数在显示器上输出表达式 x + y 的值，至此程序结束。

例 1-3 利用函数实现输出最大值。

```
#include <stdio.h>

int max(int a,int b);      /* 函数说明 */
void main()                /* 主函数 */
{
    int x,y,z;              /* 变量说明 */
    int max(int a,int b);   /* 函数说明 */
    printf("input two numbers:\n");
    scanf("%d%d",&x,&y);    /* 输入 x、y 值 */
    z = max(x,y);           /* 调用 max 函数 */
    printf("max number = %d",z); /* 输出 */
}

int max(int a,int b)        /* 定义 max 函数 */
{
    if(a > b) return a; else return b; /* 把结果返回主调函数 */
}
```

例 1-3 程序功能是由用户输入两个整数，程序执行后输出其中较大的数。程序由两个函数组成，主函数为 main() 和 max() 函数。函数之间是并列关系，可从主函数中调用其他函数。max() 函数的功能是比较两个数，然后把较大的数返回给主

函数。max()函数是一个用户自定义函数，因此在主函数中要给出说明。可见，在程序的说明部分中，不仅可以有变量说明，还可以有函数说明。关于函数的详细内容将在以后章节中介绍。

程序的执行过程是：首先在屏幕上显示提示串，请用户输入两个数，回车后由scanf函数接收这两个数并送入变量x、y中，然后调用max()函数，并把x、y的值传送给max()函数的参数a、b。在max()函数中比较a、b的大小，把大者返回给主函数的变量z，最后在屏幕上输出z的值。

C语言词汇分为六类：标识符、关键字、运算符、分隔符、常量、注释符。

1. 标识符

在程序中使用的变量名、函数名、标号等统称为标识符。除库函数的函数名由系统定义外，其余都由用户自定义。C语言规定，标识符只能是字母(A~Z, a~z)、数字(0~9)、下划线(_)组成的字符串，并且其第一个字符必须是字母或下划线。

以下标识符是合法的：

a, x, _3x, BOOK1, sum5

以下标识符是非法的：

3s 以数字开头；s*T 出现非法字符*；-3x 以减号开头；bowy + 1 出现非法字符+（加号）。

在使用标识符时还必须注意以下几点：

1) 标准C语言不限制标识符的长度，但它受各种版本的C语言编译系统限制，同时也受到具体机器的限制。例如在某版本C语言中规定标识符前八位有效，当两个标识符前八位相同时，则被认为是同一个标识符。

2) 在标识符中，大小写是有区别的，例如：BOOK和book是两个不同的标识符。

3) 标识符虽然可由程序员随意定义，但标识符是用于标识某个量的符号。因此，命名应尽量有相应的意义，以便阅读理解，作到“见名知意”。所谓“见名知意”是指通过变量名就知道变量值的含义。通常应选择能表示数据含义的英文单词（或缩写）作变量名，或汉语拼音字头作变量名。

2. 关键字

关键字是由C语言规定的具有特定意义的字符串，通常也称为保留字。用户定义的标识符不应与关键字相同。C语言的关键字分为以下几类。

(1) 类型说明符 用于定义、说明变量、函数或其他数据结构的类型，如前面例题中用到的int等。

(2) 语句定义符 用于表示一个语句的功能，如例1-3中用到的if else就是条件语句的语句定义符。

(3) 预处理命令字 用于表示一个预处理命令，如前面各例中用到的以#开头

的 include。

3. 运算符

C 语言中含有相当丰富的运算符。运算符与变量、函数一起组成表达式，表示各种运算功能。运算符由一个或多个字符组成。

4. 分隔符

在 C 语言中采用的分隔符有逗号和空格两种。逗号主要用在类型说明和函数参数表中，分隔各个变量。空格多用于语句各单词之间，作间隔符。在关键字、标识符之间必须要有一个以上的空格符作间隔，否则将会出现语法错误，例如：把“int a;”写成“inta;”，C 语言编译器会把 inta 当成一个标识符处理，其结果必然出错。

5. 常量

C 语言中使用的常量可分为数字常量、字符常量、字符串常量、符号常量、转义字符等多种。在第 2 章中将专门给予介绍。

6. 注释符

C 语言的注释符是以“/*”开头并以“*/”结尾的串。在“/*”和“*/”之间的即为注释。程序编译时，不对注释作任何处理。注释可出现在程序中的任何位置。注释用来向用户提示或解释程序的意义。在调试程序中对暂不使用的语句也可用注释符括起来，使翻译程序跳过不作处理，待调试结束后再去掉注释符。

C 语言源程序的结构特点：一个 C 语言源程序可以由一个或多个源文件组成；每个源文件可由一个或多个函数组成；一个源程序无论由多少个文件组成，都有一个且只能有一个 main() 函数，即主函数；标识符、关键字之间必须至少加一个空格以示间隔。若已有明显的间隔符，也可不再加空格来间隔。

书写程序要清晰，以便于从阅读、理解、维护的角度出发，增加程序的可读性，养成良好的编程风格。

1.4 C 语言编程环境简介

1.4.1 MS DOS 编程环境

MS-DOS 是美国微软公司的产品。MS-DOS 是一个单用户、单作业的微型计算机操作系统，其主要功能是文件管理和设备管理。DOS 是 Disk Operating System 的缩写。MS-DOS 采用层次模块结构，由三个模块和一个引导程序 BOOT 组成。这三个模块是：输入、输出系统 IBMBIO、文件管理系统 IBMDOS 和命令处理程序 COM-MAND。

引导程序 BOOT 用来把 IBMBIO 和 IBMDOS 装入内存。输入、输出系统 IBMBIO 处在 MS-DOS 的最里层，负责驱动外围设备。文件管理系统 IBMDOS 是整个操作系统的核心部分，主要任务是管理所有的磁盘文件，负责建立、删除、打开、关闭、

读写和检索各类文件。IBMDOS 向外层模块提供一系列的系统功能调用,通过这些功能调用,使 MS-DOS 的外层程序或用户程序可以方便地使用计算机系统的资源。命令处理程序 COMMAND 是 MS-DOS 的最外层模块,负责接受、识别和执行用户从终端键入的各种命令。

由于 MS-DOS 是一个单用户操作系统,因此对系统没有提供相应的保护,是一个全开放的系统。在 MS-DOS 下编程时,应用软件可以使用全部的系统资源,甚至是修改操作系统。对系统的访问可以分成四个层次。

第一个层次是通过调用 C 语言标准函数库访问系统资源,这是一个最基本的方法,也是最简单和最安全的方法。这种方法由于使用的是标准函数库,通用性较强,可移植性较好,适合初学者使用。

第二个层次是通过 DOS 功能调用访问系统资源。这种方法使用起来难度较大,需要程序员对 DOS 操作系统有较深入的了解。使用 DOS 功能调用可以充分发挥 DOS 的潜力,开发出功能更强,效率更高的软件。

第三个层次是通过基本输入、输出系统 BIOS 访问系统资源。BIOS 是由主板生产厂商提供的底层驱动程序,是硬件和操作系统之间的接口。通过 BIOS 访问硬件资源可以越过 DOS 系统,能够实现很多 DOS 不支持的功能。

第四个层次是直接访问硬件。通过使用库函数中的端口操作函数或者是直接嵌入的汇编语言编程来操作硬件。这种方法要求程序员具备良好的硬件基础知识,对计算机的硬件结构有充分的了解。在一些驱动程序和工业控制软件的开发中经常使用这种直接访问硬件的方法。该方法效率高,控制能力强,但是软件的可移植性差。

在 DOS 环境下的 C 语言编译器常用的有 Turbo C 2.0、MS-C 7.0、Watcom C 和 Gnu 的 gcc 等。其中,比较流行的是 Turbo C 2.0。在 DOS 下编程一般以文本方式为主,也可以使用图形方式编程,由于操作系统没有对图形方式给予充分的支持,编写程序时难度较大。

DOS 应用程序的特点是可以访问系统的全部资源,不受任何的限制,同时具有系统稳定,能够长期连续运行的优点,在工业控制中得到了广泛的应用。

1.4.2 Windows 编程环境

微软 Windows 在市场中已取代 DOS,成为操作系统平台的主流软件。经过多年的不断改进,Windows 已经发展到了 Windows 2000/XP,乃至更高版本,它的设计融合了分层操作系统和客户/服务器(微内核)操作系统的特点。

Windows 2000/XP 通过硬件机制实现了核心态以及用户态两个特权级别。当操作系统状态为前者时,CPU 处于特权模式,可以执行任何指令,并且可以改变状态。而在后面一个状态下,CPU 处于非特权(较低特权级)模式,只能执行非特权指令。一般来说,操作系统中那些至关重要的代码都运行在核心态,而用户程序一

般都运行在用户态。当用户程序使用了特权指令，操作系统就能借助于硬件提供的保护机制剥夺用户程序的控制权并做出相应处理。

在 Windows 2000/XP 中，只有那些对性能影响很大的操作系统组件才在核心态下运行。因为核心态和用户态的区分，所以应用程序不能直接访问操作系统特权代码和数据，所有操作系统组件都受到了保护，以免被错误的应用程序侵扰。这种保护使得 Windows 2000/XP 可能成为坚固稳定的应用程序服务器，并且从操作系统服务的角度来看，Windows 2000/XP 作为工作平台仍是稳固的。

Windows 2000/XP 的核心态组件使用了面向对象设计原则。但是 Windows 2000/XP 并不是一个严格的面向对象系统，出于可移植性以及效率因素的考虑，Windows 2000/XP 的大部分代码不是用面向对象语言编写的，而是使用了 C 语言并采用了基于 C 语言的对象实现。

选择哪种编程工具决定了程序员在应用程序中可以做什么以及能做多快。Windows 上常用的 C 语言开发工具有 Visual C++ 和 Borland C++ Builder。这两种开发工具都可以开发出优秀的 Windows 应用程序，而在系统开发方面更常用的是 Visual C++。

Visual C++ 是在 Windows 上建立 32 位应用程序的强大、复杂的工具。Visual C++ 借助于生成代码向导，能在数秒内生成可运行的 Windows 应用程序的外壳。Visual C++ 不仅仅是程序设计语言，而且也还是一个非常全面的应用程序开发环境，使用它可以开发出具有专业水平的 Windows 应用程序。Visual C++ 附带的类库，即 Microsoft Foundation Classes (MFC)，已成为许多 C++ 语言编译器进行 Windows 软件开发的工业标准。

Visual C++ 中引入了微软定义的基本类库 (MFC) 后，使 Windows 程序设计彻底实现了模板化，从而降低了程序设计的复杂性。MFC 中包含了许多微软公司已经定义好的程序开发过程中最常用到的对象，具有很好的扩展性。程序员还可以利用面向对象技术中很重要的“继承”方法，从类库中的已有对象派生出自己所需要的对象。MFC 的应用使得程序员在编制应用程序时所需要编写的代码大为减少，并有力地保证了程序具有良好的可调试性。

1.4.3 UNIX 编程环境

UNIX 操作系统诞生于 1969 年，是由贝尔实验室的两位研究人员 Ken Thompson 和 Dennis Ritchie 开发的。当时，Ken Thompson 在一台 PDP-7 计算机上开发了一个新的操作系统，并称之为 UNIX，在该系统中 Thompson 组合了许多其他操作系统中最有价值的部分，很好地利用了这些操作系统的工作成果。

UNIX 最初是用汇编语言开发的。1973 年，Ken 和 Dennis 成功地用 C 语言重写了 UNIX 操作系统。UNIX 操作系统中 95% 的代码是用 C 语言编写的，其中很小的一部分还用汇编语言编写，这部分主要集中在系统内核，直接与硬件打交道。

UNIX 操作系统有两个主要版本, 即美国电信电话公司 (AT&T) UNIX 系统和伯克利 (Berkeley) UNIX 系统, 其他的 UNIX 变体都基于这两个版本。

UNIX 操作系统的 Linux 变体是芬兰赫尔辛基大学计算机科学专业的学生 Linux Torvalds 开发的。Linux 是为基于 Intel 处理器的个人计算机而设计的。Linux 是一个可自由发布的 UNIX 版本, 并且可以免费使用。由于版权方面的原因, 没有把 Linux 称为 UNIX, 但它实际上就是 UNIX。它遵从许多与 UNIX 一样的标准。

UNIX 操作系统是一个包括文本编辑器、编译器和其他系统工具程序的程序集, 是按分层软件模型实现的。

1) UNIX 内核, 也称为基本操作系统, 负责管理所有与硬件相关的功能。这些功能由 UNIX 内核中的各个模块实现。内核包括直接控制硬件的各个模块, 这样能够在极大程度上保护这些硬件, 以避免应用程序直接操作硬件而导致混乱。用户不能直接访问内核。

2) 常驻模块层提供执行用户请求服务的例程。这些服务包括输入/输出控制服务、文件/磁盘访问服务 (称为文件系统) 以及进程创建和中止服务。应用程序通过系统调用访问常驻模块层。

3) 工具层是 UNIX 的用户接口, 通常称为 shell。shell 和其他 UNIX 命令及工具都是独立的程序, 它们是 UNIX 系统软件的组成部分, 但不是内核的组成部分。

4) UNIX 操作系统向系统中的每个用户指定一个执行环境。这个环境称为虚拟计算机, 包括一个用户接口终端和共享的其他计算机资源。UNIX 是一个多用户操作系统, 可视为虚拟计算机的集合。对用户而言, 每个用户都有自己的专用虚拟计算机。由于与其他虚拟计算机共享 CPU 和其他硬件资源, 虚拟计算机比真实的计算机要慢。

5) UNIX 操作系统通过进程向用户和程序分配资源。每个进程都有一个进程标识号和一组与之相关的资源。进程在虚拟计算机环境下执行, 就好像在一个专用的 CPU 上执行一样。

UNIX 操作系统都自己带有 C 语言编译器 cc。既可以在文本界面下编程, 也可以用在图形界面下实现应用程序。

本章小结

1. C 语言的历史。

2. C 语言的特点: 语言简洁、紧凑、使用方便、灵活; 运算符丰富; 数据结构丰富; 结构化的控制语句; 语法限制不太严格, 程序设计自由度高; 允许直接访问物理地址, 能进行位运算; 生成目标代码质量高, 程序执行效率高; 程序可移植性好。

3. 字符是组成语言的最基本的元素。C 语言字符集由字母、数字、空格、标点