



21世纪计算机科学与技术专业系列教材

C

语言程序设计

主编 李 平

C language programming



电子科技大学出版社

• 21 世纪计算机科学与技术专业系列教材

C 语言程序设计

主 审	白宝兴
主 编	李 平
副主编	孙爽滋
	丁 岩
	李 奇

电子科技大学出版社

图书在版编目(CIP)数据

C 语言程序设计/李平主编. —成都:电子科技大学出版社, 2005. 4

(21 世纪计算机科学与技术专业系列教材)

ISBN 7-81094-733-8

I. C… II. 李… 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(2005)第 003074 号

21 世纪计算机科学与技术专业系列教材

C 语言程序设计

主 编 李 平

出 版: 电子科技大学出版社(成都市建设北路二段四号 邮编 610054)

责任编辑: 张致强

发 行: 新华书店

印 刷: 安徽省蚌埠市方达印刷厂

开 本: 787×960 1/16 **印张:** 23.25 **字数:** 357 千字

版 次: 2005 年 4 月第一版

印 次: 2005 年 4 月第一次印刷

书 号: ISBN 7-81094-733-8/TP·394

印 数: 1—5000 册

定 价: 29.40 元

编审说明

迈入被称为“E时代”的21世纪,计算机正成为普通而又重要的使用工具,对计算机的掌握程度更是成了评价新型人才的标准之一。作为21世纪的人才,将不再局限于只会简单地使用计算机,更要求他们具备基本编程能力。

选择一个适当的编程语言学习,既有利于编程思维的形成,又有利于加深对计算机的理解。在高级编程语言当中,C语言是当前得到迅速推广应用的—种计算机编程语言,无论是数据库应用系统开发,还是工程上计算机辅助设计,都普遍采用C语言。随着计算机科学技术的发展,C语言也不断得到发展和完善,其功能丰富,表达力强,使用灵活方便,深受广大用户欢迎。目前流行的各种C语言的扩充版本,如C++、C#等都要求用户具备基本的C语言知识,因此,学习C语言程序设计,对不同要求的用户(专业编程人员、非专业编程人员)都是非常重要、非常有意义的。

本书根据全国计算机等级考试委员会审议通过的《全国计算机等级考试大纲(2003级)》编写,既可作为高等学校计算机专业和非计算机专业课程教材,也可供广大工程技术人员学习参考。通过使用本教材和配套教材(《C语言程序设计上机指导及同步训练》),可以达到以下培养目标:

1. 程序设计入门,培养编程思维;
2. 掌握程序设计基本方法,达到国家计算机等级二级编程水平;
3. 为向C++、Visual C++过渡做准备。

编者根据多年的教学经验,感到C语言概念繁多,规则复杂,掌握起来十分困难,因此在本书编写时力求深入浅出、通俗易懂,以生活中的实例来增加读者的编程兴趣。同时,编者认为编程能力的形成,一定要着重加强实践操作,通过习题演练增强认识,加深理解,因此,建议读者充分利用配套教材,达到事半功倍的效果。

本书内容共分14章,分别由李平(第6、10章)、孙爽滋(第4、5、9、14章)、丁岩(第1、3、7、11章)、李奇(第2、8、12、13章)完成,其中第14章可作为参考,以加深对C语言的理解和使用。书中所有例程均经过上机调试(在Turbo C集成环境下)。

由于编写时间仓促,书中内容肯定存在不妥之处,望广大读者和有关专家、教授不吝赐教。

21世纪计算机科学与技术专业系列教材

2005年4月

目 录

第 1 章 C 语言程序设计初步	(1)
§ 1.1 程序设计语言	(1)
§ 1.2 C 语言符号集	(7)
§ 1.3 用库函数组装 C 语言	(9)
§ 1.4 自己设计 C 函数	(12)
第 2 章 程序设计基础知识	(19)
§ 2.1 利用计算机处理实际问题的具体步骤	(19)
§ 2.2 算法的概念	(20)
§ 2.3 结构化程序设计	(28)
第 3 章 C 数据及其表达式	(29)
§ 3.1 C 数据类型	(29)
§ 3.2 数据在 C 中体现方式	(30)
§ 3.3 C 运算符及其表达式	(42)
§ 3.4 C 语言中类型转换	(48)
第 4 章 顺序结构程序设计	(53)
§ 4.1 语句	(53)
§ 4.2 输出函数	(56)
§ 4.3 输入函数	(68)
§ 4.4 常用的数学函数	(75)
§ 4.5 顺序结构程序设计举例	(77)
第 5 章 选择结构程序设计	(80)
§ 5.1 逻辑量概述	(80)
§ 5.2 关系运算符和关系表达式	(81)
§ 5.3 逻辑运算符和逻辑表达式	(83)
§ 5.4 if 语句	(87)
§ 5.5 switch 语句	(100)
§ 5.6 选择结构程序举例	(103)
第 6 章 循环结构程序设计	(109)
§ 6.1 while 语句	(109)
§ 6.2 do-while 语句	(114)

§ 6.3	for 语句	(116)
§ 6.4	break 和 continue 语句	(120)
§ 6.5	goto 语句构成的循环	(123)
§ 6.6	循环的嵌套	(124)
第 7 章	数 组	(128)
§ 7.1	一维数组	(128)
§ 7.2	二维数组	(134)
§ 7.3	字符数组	(140)
第 8 章	函 数	(150)
§ 8.1	完整的 C 程序结构	(150)
§ 8.2	函数定义格式及返回值	(152)
§ 8.3	函数调用及其调用规范	(156)
§ 8.4	函数的嵌套调用	(173)
§ 8.5	函数的递归调用	(176)
§ 8.6	变量的存储类别及在函数调用中的作用	(182)
§ 8.7	多文件程序的函数调用及连接方法	(188)
第 9 章	指 针	(190)
§ 9.1	指针的基本概念	(190)
§ 9.2	变量与指针	(194)
§ 9.3	一维数组与指针	(206)
§ 9.4	多维数组和指针	(221)
§ 9.5	字符串和指针	(230)
§ 9.6	函数和指针	(238)
§ 9.7	指针数组	(248)
§ 9.8	指针和二级指针	(257)
§ 9.9	有关指针问题的小结	(262)
§ 9.10	指针应用程序举例	(268)
第 10 章	结构体和共用体	(273)
§ 10.1	结构体	(273)
§ 10.2	结构体数组	(280)
§ 10.3	结构体与指针	(283)
§ 10.4	链 表	(287)
§ 10.5	共用体的定义与引用	(300)
§ 10.6	枚举类型数据	(302)

第 11 章 文 件	(305)
§ 11.1 文件的概念	(305)
§ 11.2 文件的类型及其文件的指针	(305)
§ 11.3 常用的文件函数	(307)
§ 11.4 文件应用举例	(319)
第 12 章 位运算	(321)
§ 12.1 基本位运算符与位运算	(321)
§ 12.2 位移运算符与位移运算	(327)
§ 12.3 位运算的复合赋值运算符	(328)
§ 12.4 位 段	(328)
§ 12.5 位运算应用举例	(332)
第 13 章 编译预处理	(336)
§ 13.1 宏定义	(336)
§ 13.2 文件包含预处理	(341)
§ 13.3 条件编译预处理	(344)
第 14 章 屏幕管理与菜单设计	(347)
§ 14.1 字符显示原理	(347)
§ 14.2 字符屏幕函数	(349)
§ 14.3 保存屏幕与恢复屏幕	(355)
参考文献	(363)

第 1 章 C 语言程序设计初步

§ 1.1 程序设计语言

程序设计语言是人与计算机进行交流的工具。程序设计要在一定的程序设计语言环境下进行。只有熟练地掌握了一门程序设计语言才能更好地进行程序设计，为实际问题的解决提供方便。

1.1.1 程序设计语言的发展

从计算机诞生至今，程序设计语言也在伴着计算机技术的进步不断升级换代。

1. 机器语言

一种 CPU 的指令系统，也称该 CPU 的机器语言，它是该 CPU 可以识别的一组由 0 和 1 序列构成的指令码。下面是某 CPU 指令系统中的两条指令：

10000000 加

10010000 减

用机器语言编程序，就是从所使用的 CPU 的指令系统中挑选合适的指令，组成一个指令系列。这种程序虽然可以被机器直接理解和执行，却由于它们不直观，难记、难认、难理解、不易查错，只能被少数专业人员掌握，同时程序的生产效率很低，质量难以保证。这种繁重的手工方式与高速、自动工作的计算机极不相称。

2. 汇编语言

为减轻人们在编程中的劳动强度，20 世纪 50 年代中期人们开始用一些“助记符号”来代替 0、1 码编程，例如前面的两条机器指令可以写为：

$A+B \Rightarrow A$ 或 ADD A, B

$A-B \Rightarrow A$ 或 SUB A, B

这种用助记符号描述的指令系统，称为符号语言或汇编语言。

用汇编语言编程，程序的生产效率及质量都有所提高。但是汇编语言指令是机器不能直接识别、理解和执行的。用它编写的程序经检查无误后，要

先翻译成机器语言程序才能被机器理解、执行。这个翻译转换过程称为“代真”。代真后得到的机器语言程序称为目标程序 (Object program)，代真以前的程序，称为源程序 (Source program)。由于汇编语言指令与机器语言指令基本上具有一一对应的关系，所以汇编语言程序的代真可以由汇编系统以查表的方式进行。

汇编语言与机器语言，都是依 CPU 的不同而异，它们都称为面向机器的语言。用面向机器的语言编程，可以编出效率极高的程序。但是程序员用它们编程时，不仅要考虑解题思路，还要熟悉机器的内部结构，并且要“手工”地进行存储器分配。这种编程的劳动强度仍然很大，给计算机的普及和推广造成很大的障碍。

3. 面向过程的语言

汇编语言和机器语言是面向机器的，随机器而异。1954 年出现的 FORTRAN 语言以及随后相继出现的其他高级语言，使人们开始摆脱进行程序设计必须先熟悉机器的桎梏，把精力集中于解题思路和方法上，使程序设计语言开始与解题方法相结合。其中一种方法是把解题过程看成是数据被加工的过程。基于这种方法的程序设计语言称为面向过程的设计语言。C 语言就是一种面向过程的程序设计语言。下面是一个计算圆柱体体积的 C 语言程序片段：

```
main()                /*告诉编译器 C 语言程序由此开始执行*/
{                    /*这是一个程序段开始*/
    int r,h;          /*设半径 r 与高 h 为整数*/
    float v;          /*设体积 v 为实数*/
    v=3.14159*r*r*h;  /*计算体积 v*/
    printf("%f",v);   /*输出结果*/
}                    /*程序结束*/
```

我们看到这个程序是很好理解的，其中计算体积的语言与我们所习惯的数学式子没有什么根本的区别 (“/*” 与 “*/” 之间的内容称为注释，目的是让程序更容易被理解)。显然，使用高级语言编程可以较大地降低编程过程的劳动强度，提高编程的效率。高级语言的诞生是计算机技术发展史上的一个里程碑。它使人们能摆脱具体机器指令系统的束缚，用接近人们习惯的语言来构思解题过程，从而大大提高了编程效率，使人们能够编制出规模越来越大的程序，以满足日益广泛而深入的应用需求。

实际上，程序是对现实世界的运动状态的模拟。面向过程的设计认为，每个程序都需要完成一些规定的功能。每个功能实际上是通过数据进

行一系列加工的过程而实现的。因而程序设计包括组织数据——设计数据结构以及设计对数据结构进行加工的过程——设计算法两个部分。

从1954年第一种高级语言FORTRAN问世后不久,不同风格、不同用途、不同规模、不同版本的面向过程的高级语言便蜂涌而起。据统计,全世界已有成千种的高级语言,其中使用较多的有近百种。在这些语言中,C语言以其高效、灵活、功能丰富、表达力强、移植性好而受到青睐,被称为近十年来在计算机程序设计实践中做出了重大贡献的一种语言。

C语言目前已成为计算机程序语言中的主流语种。它既可用于编写系统软件,又可用于编写应用软件,它是学习和掌握更高层的开发工具C++和VC++的基础。1960年出现的ALGOL60语言是C语言的前身。ALGOL60离硬件较远,不宜用来编写系统程序。1963年英国剑桥大学推出了比ALGOL60更接近于硬件的CPL(Combined Programming Language)语言,但它的语言复杂,规模较大,不宜实现。1967年英国剑桥大学Martin Richards对CPL作了简化,推出了BCPL(Basic Combined Programming Language)语言。1970年美国贝尔实验室的Ken Thompson对BCPL作了简化,以第一个字母“B”为新语言的名称,它简单且接近硬件。并且用B语言编写了第一个Unix操作系统。C语言是1972年由美国的D.M. Ritchie设计发明的,并首次在UNIX操作系统的DEC PDP-11计算机上使用。它继承了早期BCPL和B语言的优点。1978年由美国电话电报公司(AT&T)贝尔实验室正式发表了C语言。同时由B.W.Kernighan和D.M.Ritchie合著了著名的“THE C PROGRAMMING LANGUAGE”一书。通常简称为《K&R》,也有人称之为《K&R》标准。但是,在《K&R》中并没有定义一个完整的标准C语言,由于没有统一的标准,使得这些C语言之间出现了一些不一致的地方。后来由美国国家标准学会在《K&R》基础上制定了一个C语言标准,于1983年发表。通常称之为ANSI C标准,它成为现行的C语言标准。

由于C语言的强大功能和各方面的优点逐渐为人们认识,到了20世纪80年代,C开始进入其他操作系统,并很快在各类大、中、小和微型计算机上得到了广泛的使用。随着微型计算机的日益普及,出现了许多C语言的版本,目前最流行的C语言有以下几种:

- 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。本书以Turbo C为编译环境介绍C语言的使用。

C 语言是一种结构化语言。它的语言简洁、紧凑、使用起来方便灵活。它层次清晰，便于按模块化方式组织程序，易于调试和维护。C 语言的表现能力和处理能力极强，它不仅具有丰富的运算符和数据类型，便于实现各类复杂的数据结构；它还可以直接访问内存的物理地址，进行位(bit)一级的操作；能实现汇编语言的大部分功能，可以对硬件直接进行操作。由于 C 语言实现了对硬件的编程操作，所以我们说 C 语言集高级语言和低级语言的功能于一体。此外，C 语言的语法限制不是很严格，编写程序自由度大。它还具有效率高、可移植性强等特点，因此被广泛地移植到了各种类型的计算机和不同的操作系统上。

4. 面向对象的程序设计语言

面向对象的程序设计是一种结构模拟方法，它把现实世界看成是由许多对象(Object)所组成，对象之间通过互相发送和接收消息进行联系；消息激发对象本身的运动，形成对象状态的变化。从程序结构的角度，每个对象都是一个数据和方法的封装体——抽象数据类型。

从分类学的观点看，客观世界中的对象都是可以分类的。也就是说，所有的对象都属于特定的“类”(class)，或者说每一个对象都是类的一个实例。因而，面向对象的程序设计的一个关键是定义“类”，并由“类”生成“对象”。

面向对象的程序比面向过程的程序更清晰、易懂，更适宜编写更大规模的程序，正在成为当代程序设计的主流。面向对象的程序设计语言有 Simmla-67、Smalltalk80、Java 等。此外，还要特别指出的是，由 C 语言派生出来 C++ 语言，它是一种多范型程序设计语言，不仅可以用它编写面向对象的程序，还可以用它编写面向过程的程序。

1.1.2 程序设计语言的支持环境

现在几乎每一种计算机都配备了操作系统。操作系统是各种软件中最重要的一种，或者说它是各种软件的核心与基础。所有其他程序的运行都要在操作系统的控制下进行。操作系统的作用是有效地组织和利用计算机的软、硬件资源，使各种程序能在操作系统的管理下协调工作。操作系统的类型有许多种，例如有分时操作系统、批处理操作系统、多道作业操作系统等。一个高级语言源程序必须经过编译系统处理(编译)，然后在操作系统控制下才能为计算机执行。

操作系统的功能主要包括：CPU 管理、存储管理、文件管理、设备管理、作业管理等。不同的操作系统的功能和工作方式是不同的。目前最广泛流行

的操作系统有 Unix、MS-DOS、OS/2、Windows 等。

使用高级语言编写程序时应当注意程序的运行环境，要充分运用所使用的计算机系统提供的软件环境支持（例如充分利用操作系统提供的功能），并了解源程序在所运行的操作系统管理下是如何工作的。例如在不同的操作系统下编译和运行 C 语言程序的方式和命令是不同的。

1.1.3 源程序的编辑、编译、连接和执行

C 语言采用编译方式将源程序转换为二进制的目标代码。从编写好一个 C 程序到完成运行一般经过以下几个步骤：

1. 编辑

所谓编辑，包括以下内容：①将源程序逐个字符输入到计算机内存中；②修改源程序；③将修改好的源程序保存起来。由用户选定文件名以“.c”为扩展名保存在某个目录下。所编辑的对象就是源程序，它是以 ASCII 代码的形式输入和存储的，不能被计算机执行。目前使用较多的编辑软件有：Unix 下的编辑程序 ed, vi 等；MS-DOS 下的 EDLIN、Wordstar；Windows 下的 Write、Word 等字处理软件。关于编辑软件的使用方法请参阅《C 语言程序设计之上机指导与同步训练》一书或其他有关手册。

2. 编译

编译就是将已编辑好的源程序（已存储在磁盘文件中）翻译成二进制的目标代码。在编译时，还要对源程序进行语法检查，如发现有错，则在屏幕上显示出错信息，此时应重新进入编译状态，对源程序进行修改后再重新编译，直到通过编译为止。编译后得到的二进制代码在 Unix 下是后缀为“.o”的文件，在 MS-DOS 下是后缀为“.obj”文件。这里需要注意的是：经编译后得到的二进制代码还不能直接执行，因为每一个模块往往是单独编译的，必须把经过编译的各个模块的目标代码与系统提供的标准模块（如 C 语言中的标准函数库）连接后才能运行。

3. 连接

将各模块的二进制目标代码与系统标准模块经连接处理后，得到具有绝对地址的可执行文件，它是计算机能直接执行的文件。在 Unix 下它以“.out”为后缀（例如 f.out），在 MS-DOS 以下“.exe”为后缀（例如 f.exe）。

4. 执行

现在你就可以把经过修改、连接好的 C 程序执行了，执行一个经过编译和

连接好的可执行的目标文件只有在操作系统的支持和管理下才能进行。这样你就可以得到程序的预期值了，如果结果不是想要的预期值，需要再进行调试、修改源程序。图 1.1 用以表示一个程序的编辑、编译、连接、运行的全过程。

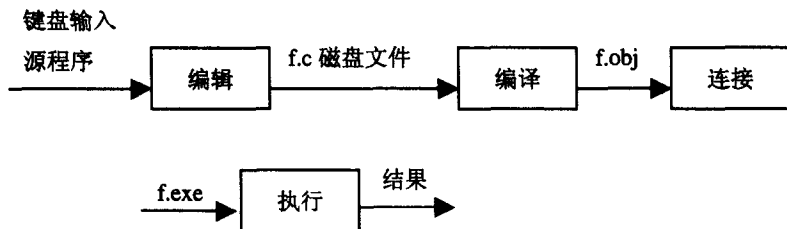


图 1.1

其中文件取名为 f，后缀按 MS-DOS 的规则表示。

近来“集成化”的工具环境已将编辑、编译、连接、调试工具集于一身（例如 Turbo C），用户可以方便地在窗口状态下连续进行编辑、编译、连接、调试、运行的全过程。

下面我们介绍一下 Turbo C 的窗口菜单。

(1) 进入 Turbo C 后屏幕顶部是一行“命令”菜单，显示如下，如图 1.2 所示：

File	Edit	Run	Compile	Project	Options	Debug	Break/Watch
Line 1	Col 1	Insert Indent Tab Fill Unindent				c:NONAME.c	
Message							

图 1.2

用 F10 可以激活主菜单，使用键盘上的“←”、“→”来移动光标，光标指到哪个命令名，按回车即执行该命令。在 File 处回车后它会弹出子菜单，用“↑”、“↓”键移到相应命令按回车执行。主菜单中 File 包括文件的建立、存盘、退出程序、进入 DOS 等命令。

(2) 在编辑窗口输入源程序。在编辑窗口的第一行是对当前编辑文件的状态说明：

Line 1 Col 1 Insert Indent Tab Fill Unindent c:NONAME.c

它说明光标所在位置第一行第一列正使用“Insert”方式。“Indent”自

动缩进开关, “Tab” 是制表开关。最右端的 “NONAME.c” 是正在编辑的文件名, 可自己根据需要定义不同的文件名。

(3) 编辑好一个 C 程序后, 使用 “File / Save” 命令把程序保存起来, 按 “F9” 键进行编译、连接, 此时在屏幕上会显示在编译、连接中出现的错误, 按任意键使光标回到编译状态, 光标会停留在出错位置。根据 Message 中提示的错误原因改正错误, 再按 “F9” 重新进行编译、连接直到没有错误。

(4) 在编译、连接没有错误后, 按 F10 键激活主菜单, 光标移到 “Run” 后回车, 便开始执行此程序, 屏幕会显示程序的输出结果。若程序需要用户输入数据, 则此时 (黑屏状态下) 输入数据, 然后程序会继续执行, 输出结果。若结果不对, 需要再修改源程序时则按 F10 键将光标移到 Edit 处回车进入编译状态修改程序。再重复(3)、(4)步。

以上的(3)、(4)步骤可以用 Ctrl+F9 代替, 它可以进行编译、连接与执行的全过程。使用 Alt+F5 可查看程序输出结果。使用 Alt+X 可退出 Turbo C 工作环境, 如果希望重新打开刚刚编译好的 C 程序, 按 F3 键或 “File / Load” 命令输入此文件名即可。

§ 1.2 C 语言符号集

1.2.1 基本符号集

C 语言的符号集是指在 C 程序中可以出现的字符, 包括大小写英文字母、数字、转义字符和键盘符号。这些字符多数是可见的, 其中在程序中体现的不可见字符, 例如回车换行符等控制字符, C 语言规定用转义字符来表示, 转义字符由 “反斜杠字符” 和其后的单个字符或若干字符组成, 转义字符是一种控制字符, 并不显示在屏幕上, 我们将在第三章重点介绍。键盘符号见表 1-1。

表 1-1

~	`	!	@	#	\$
=	+	(	)	*	&
{	}	[	]	:	;
%	,	.	?	'	“
-	_	>	<	/	\
	^	~	~		

1.2.2 标识符

和其他高级语言一样,标识符是用来标识变量名、函数名、数组名和文件名等对象的有效字符序列。其实标识符就是一个名字(由用户指定),C语言规定了标识符的定名规则:标识符由英文字母、数字和下划线组成;长度不大于32个字符,而PC机上通常取前8个字符为有效字符;必须以英文字母或下划线开头。例如下面是定义的合法的C语言标识符:

abc、 file2 、 _sun 、 Student_Name 、

下面是不合法的C语言标识符:

2y (数字不能作为标识符第一个字符)

book one (空格不能在标识符中出现)

Exe. (标识符中出现非法字符)

long (与关键字同名)

用户在自定义标识符时除了按照上述规则外,还应注意以下几点:

1.C语言为我们提供了32个关键字(又称保留字),关键字是具有特定含义的标识符,这些关键字是被C语言本身使用的,不能再用来作变量名、函数名、数组名和文件名。关键字见表1-2。

表 1-2

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

2.由于C语言严格区分大小写,标识符命名也区分大小写,即认为相同字母的大小写是不同的字符,例如Student、student、STUDENT是3个不同的标识符。标识符一般选用小写字母,特殊时可选用大写字母。关键字都用小写字母表示。

3.在选取标识符时应做到“见名知意”,以便识别标识的对象,提高程序的可读性。

§ 1.3 用库函数组装 C 语言

C 语言本身语句很少,许多功能都是通过函数来完成的。例如:输入输出的功能并不是 C 语言本身提供的,而是 C 的库函数所提供的,这是 C 的一个重要特色。这样做的目的是使 C 语言的核心部分规模较小,而外围(函数)可以做得很丰富,并且可以根据需要增加新的函数,使 C 语言有较大的灵活性和多方面的功能。在编写 C 程序时应尽量利用 C 的库函数所提供的函数功能,来实现自己的目的。

下面是利用库函数编程序的一些简单的例子。

【例 1.1】 一个最简单的 C 语言小程序。

```
main()  
{  
    printf("I am a student.\n");  
}
```

程序的运行结果:

I am a student.

通过这个最简单的小程序可以总结出 C 程序的基本结构:

1.C 程序是由函数组成的,每一个函数完成相对独立的功能。本程序是由一个称为 main 的函数构成的。main 是函数名,函数名后面一对圆括号内写函数的参数,本程序的 main 函数没有参数,但不能省略。一个完整的程序必须有一个 main 函数,它称为主函数,程序总是从 main 函数开始执行的,不论 main 函数在整个程序的什么位置。

main()后面有一对花括号,花括号内的部分称为函数体。本程序中的函数体只由一个语句组成。一般情况下函数体由“说明部分”和“执行部分”组成。本例中只有执行部分而无说明部分。某些特殊情况下可以没有说明部分,甚至没有执行语句部分,这种函数称为空函数。例如:

```
empt()  
{ }
```

2.C 规定每个语句以分号(;)结束。分号是语句不可缺少的组成部分,本程序中 main 函数的函数体内只有一个语句,也必须有一个分号。

3.printf 是 C 的库函数中的一个函数,它的作用是在显示屏上输出指定的内容,此例输出“I am a student.”字符串。字符串末尾的“\n”是 C 语言中

的转义字符,其作用是在输出“I am a student.”以后执行一个回车换行操作。以后还会讲述在反斜杠“\”的后面跟一个指定的字符,就会组成一个具有专门含义的“转义字符”。如果以后还有输出语句,将从下一行的最左端开始输出。

请看以下程序:

```
main()  
{  
    printf("I am \n");  
    printf("a student. \n");  
}
```

执行这个程序将得到如下输出:

I am a student.

没有“\n”时,后面的输出将不换行。请看下面的程序:

```
main()  
{  
    printf("I a");  
    printf("m a stu");  
    printf("dent.\n");  
}
```

程序的运行结果:

I am a student.

C 语言提供了丰富的库函数,每个函数实现一定的功能,例如可以用 cos 函数求余弦值,用 abs 函数求绝对值,用 exp 函数求一个数的指数等。C 编译系统将这些函数集中存放在一些库文件中,按函数名调用十分方便。现代程序设计方法认为:程序设计是一种艰巨的脑力劳动,人在进行程序设计的过程中,犯错误的机会很多,如果能找到已经验证的程序模块使用,是提高程序设计效率和程序可靠性的有效措施,按照这一思想,用 C 语言进行程序设计时,应优先采用 C 语言函数库中的函数,不要一拿到问题就立即动手从头到尾自己写程序。

4. 函数 printf 由主函数 main 调用,形成图 1.3 所示的层次结构。

【例 1.2】 计算一个数的余弦值的 C 语言程序。

```
#include "math.h"  
main()
```

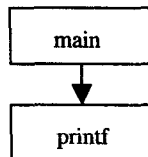


图 1.3