

21世纪高等学校公共课计算机规划教材丛书

gao deng xu e xiao gong gong ke ji suan ji gui hua jiao cai cong shu

C/C++

程序设计

■ 主编 乔保军



电子科技大学出版社

21 世纪高等学校公共课计算机规划教材丛书

C/C++ 程序设计

主 编 乔保军

副主编 马 飞 李海平 凌广明

电子科技大学出版社

图书在版编目(CIP)数据

C/C++ 程序设计 / 乔保军主编. — 成都: 电子科技大学出版社, 2011. 7

(21 世纪高等学校公共课计算机规划教材丛书)

ISBN 978-7-5647-0888-7

I. ①C… II. ①乔… 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 (2011) 第 133491 号

内 容 简 介

本书包括基础篇和实战篇两大部分。基础篇主要内容为: 初识 C、变量和常量、运算符与表达式、控制流、函数、指针、线性表、文件、位运算、预编译指令、C++、调试技术等。实战篇主要通过简易计算器、趣味火柴棍、英汉小字典、贪吃蛇游戏、迷宫求解、图书管理系统等实验对 C/C++ 程序设计充分掌握。

本书内容简练, 通俗易懂, 可以作为高等院校相应课程的参考教材, 也可供计算机专业工作者使用。

21 世纪高等学校公共课计算机规划教材丛书

C/C++ 程序设计

主 编 乔保军

出 版: 电子科技大学出版社 (成都市一环路东一段 159 号电子信息产业大厦 邮编: 610051)

策划编辑: 谢应成

责任编辑: 谢应成

主 页: www.uestcp.com.cn

电子邮箱: uestcp@uestcp.com.cn

发 行: 新华书店经销

印 刷: 郑州易佳印务有限公司

成品尺寸: 185mm × 260mm 印张 17.5 字数 440 千字

版 次: 2011 年 7 月第一版

印 次: 2011 年 7 月第一次印刷

书 号: ISBN 978-7-5647-0888-7

定 价: 36.00 元

■ 版权所有 翻印必究 ■

- ◆ 本社发行部电话: 028-83202463; 本社邮购电话: 028-83208003。
- ◆ 本书如有缺页、破损、装订错误, 请寄回印刷厂调换。

前言

C 语言自 1972 年问世以来,就得到了众多程序开发者的青睐和拥护,计算机技术日新月异的发展给 C 语言提供了极为广阔和自由的发展空间,取得了举世瞩目的辉煌成就。作为一门“历史悠久”的计算机语言,C 语言强大的生命力源自于语言设计者巨大的创新能力和 C 语言自身诸多显著的特性。无论从商业价值还是从教学价值来看,C 语言都具有极为重要的地位。C++ 语言作为 C 语言的扩展,不仅完全兼容 C 语言,而且提供了众多极为强大的语言特性,不仅丰富了语言的表现能力,更为重要的是提供了更为宽广的平台,使得开发者可以利用面向对象的思想去分析问题和解决问题,可以方便地进行大型项目的开发和维护。

C 和 C++ 语言不仅是我国乃至全球高等院校计算机专业的必修课程,而且是很多高等院校理工类专业的基础课程。可以说绝大多数程序开发人员都是从中汲取营养、叩开计算机世界的大门,进而领略并谱写着极为精彩的辉煌篇章。时至今日,C 和 C++ 语言给我们的不仅仅是思想和方法,虽然从某种意义上说这一点极为重要,但更为重要的是在发展异常迅猛的计算机行业里面,它依然发挥着巨大的作用,尤其是在嵌入式领域,再次向世人展示了极为强大的生机和无与伦比的语言魅力和应用价值。

本书的编写人员都是从事教育第一线的教师,具有丰富的项目开发经验。经过长期的教学实践和对新型计算机人才培养的探究,并结合企业一流的技术团队,逐渐形成了符合社会发展需求的教育理念和教学方法。从启迪思想入手,透过语法剖析本质,利用项目强化思想,从全新的视角引导学生畅游计算机的神奇世界。以实践教学环节的精心设计激发学生潜在的思维能力,力求改变单一的教学模式,使学生成为学习的主体进而充分融入程序设计的乐园之中,不仅要掌握 C 和 C++ 语言本身,更为重要的是为今后专业课程的学习打下坚实的基础。

本书内容全面,结构清晰。本书分为基础篇和实战篇两大部分。基础篇主要介绍了 C 及 C++ 语言的相关概念和基本语法,首先介绍了 C 程序的基本框架、数据类型、程序设计等基本概念;其次介绍了控制流,包括顺序结构、选择结构和循环结构三种基本结构;学习完这些内容之后进入了函数、指针等更具挑战性的章节;为了深入理解数据类型和指针的使用,我们提前引入《数据结构》中的线性表;然后是文件、位运算、预编译指令,并引入了 C++ 相关知识;最后一章列出了一些常见的错误及解决方案,介绍了一些常用的调试技术等。这些都是作者在教学过程和程序设计过程中积累的经验和感悟。实战篇主要侧重实际动手能力的培养,程序设计课程理应使学生具有较强的实践动手能力,这也是本书大力提倡和引导的宗旨之一。在实战篇中,通过简易计算器、趣味火柴棍、英汉小字典、贪吃蛇游戏、迷宫求解、图书管理系统等趣

味性和实践性较强、可操作性适中而且贴近生活的项目,让读者在巩固所学内容的同时提高了算法设计能力、编程能力及综合应用能力。这也是本书彰显实用价值之处。

本书内容简练,通俗易懂,可供计算机专业工作者使用,也可以作为高等院校相应课程的参考教材。

本书由乔保军担任主编,马飞、李海平、凌广明担任副主编。乔保军编写了基础篇的第3、9、10章和实战篇的图书管理系统,马飞编写了基础篇的第6、7、8章和实战篇的英汉小字典和贪吃蛇游戏,李海平编写了基础篇的第4、5、12章和实战篇的简易计算器和趣味火柴棍,凌广明编写了基础篇的第1、2、11章和实战篇的迷宫求解。全书由乔保军统稿。

由于编者水平有限,难免有错误和疏漏之处,敬请广大老师和同学们多提宝贵意见,批评指正。

编 者



目 录

第一部分 基础篇

第 1 章 初识 C 1

1.1 C 语言的诞生与发展 1

1.2 C 语言的特点 1

1.3 Visual C++ 6.0 开发环境 2

1.4 简单的 C 程序介绍 7

1.5 C 程序的构成 8

1.6 本章小结 9

1.7 练习题 9

第 2 章 变量和常量 10

2.1 计算机中的数据 10

2.2 数据的存储 11

2.3 数据角色 18

2.4 数据社会 27

2.5 本章小结 29

2.6 练习题 30

第 3 章 运算符与表达式 31

3.1 算术运算符 31

3.2 赋值运算符 34

3.3 关系运算符 35

3.4 逻辑运算符 37

3.5 条件运算符 38

3.6 逗号运算符 38

3.7 运算符的优先级 39

3.8 本章小结 40

3.9 练习题 40

第 4 章 控制流 41

4.1 顺序结构 41

4.2 选择结构 49

4.3 循环结构 62

4.4 本章小结 78

4.5 练习题 80

第 5 章 函数 86

5.1 函数概述 86

5.2 函数的类别 86

5.3 函数的使用 87

5.4 函数的参数传递 93

5.5 变量的作用域和存储类别 97

5.6 递归 105

5.7 本章小结 107

5.8 练习题 108

第 6 章 指针 112

6.1 指针 112

6.2 指针与函数 118

6.3 指针与数组 121

6.4 指针与字符串 128

6.5 结构体指针 132

6.6 void 指针 134

6.7 本章小结 135

6.8 练习题 135

第 7 章 线性表 137

7.1 顺序表 137

7.2 链式表 140



7.3	本章小结	149
7.4	练习题	149
第 8 章	文件	150
8.1	打开和关闭文件	150
8.2	文本文件读写	152
8.3	二进制文件的读写	157
8.4	文件的随机读写	159
8.5	本章小结	160
8.6	练习题	160
第 9 章	位运算	161
9.1	位运算符	161
9.2	位运算应用	165
9.3	位域	167
9.4	本章小结	169
9.5	练习题	169
第 10 章	预编译指令	170
10.1	宏定义	170
10.2	文件包含	174
10.3	条件编译	176
10.4	本章小结	179
10.5	练习题	179
第 11 章	C++	180

11.1	C++ 概述	180
11.2	C++ 程序设计	182
11.3	本章小结	195
11.4	练习题	196
第 12 章	调试技术	197
12.1	常见错误类型	197
12.2	调试技术	203
12.3	本章小结	204
12.4	练习题	205

第二部分 实 战 篇

简易计算器	206
趣味火柴棍	219
英汉小字典	223
贪吃蛇游戏	232
迷宫求解	239
图书管理系统	247

第三部分 附 录

附录 A	ASCII 字符码	265
附录 B	C 语言的关键字	266
附录 C	运算符的优先级	266
附录 D	库函数	268

第一部分 基础篇

第



章

初识C

C 语言(Combined Language)是一种既具有高级语言特点又具有汇编语言特点的计算机程序设计语言。因而 C 语言的应用范围非常广泛,而且具有巨大的发展潜力和应用前景。尤其是嵌入式系统开发的异军突起,为 C 语言提供了更为广阔的发展空间。

1.1 C 语言的诞生与发展

C 语言由贝尔实验室的 Dennis Ritchie 在 B 语言的基础上于 1972 年设计出来的。随着 C 语言不断地发展和日益广泛的应用,众多的编译器厂商根据不同的开发平台和应用领域提供了各具特色的编译环境,因而出现了许多不同的 C 语言版本。这给 C 语言的应用和学习带来了诸多不便,从某种程度上讲极大地阻碍了 C 语言的发展,为此 1989 年 ANSI(美国国家标准研究所)为 C 语言制定了一套 ANSI 标准,随后又经过不断地发展和完善,成为现行的 C 语言标准。其中 Microsoft C(MS C)、Borland Turbo C(Turbo C)、AT&T C 是比较流行的版本,得到了广泛的应用。

由于自身的优势,C 语言无论是在系统开发还是应用开发领域都有十分广泛的应用,我们可以从两个方面讨论一下 C 语言学习的必要性。

一方面,C 语言作为一门历史悠久而且极为成功的语言,很适合作为编程世界的入门教程,C 语言是许多著名学府的必修课程,本身具有极高的教学价值,更为重要的是,当前许多语言都可以说是发源于 C 语言,比如 C++、C#、Java、JavaScript 等,通过 C 语言的学习将为以后学习这些语言打下良好的基础。

另一方面,嵌入式技术和 Linux 程序开发技术为 C 语言提供了更为广阔的发展空间。目前嵌入式技术和 Linux 程序开发技术成为计算机界比较流行的两大技术。作为一种非常基础和常用的编程语言,C 语言在嵌入式系统开发中体现了其强大的软硬件操控能力。如果说 C 语言和 Unix 语言是一对孪生兄弟的话,那么 C 语言和 Linux 就是一对父子,在 Linux 系统中,C 语言得到了更为深入的发展。

1.2 C 语言的特点

C 语言备受世人关注,直到今天仍然是最受欢迎的语言之一,应用范围之广,使用人群之



多,发展历史之久,无疑已经成为计算机界的神话。这一切都要依赖于它强大的功能和极为突出的特点。归纳起来 C 语言具有下列特点:

(1) C 语言兼具高级语言和低级语言两家长

C 语言很好地体现了结构化编程的思想,利用控制语句、循环语句和条件语句将实现功能模块的函数组织起来,通过“自顶向下,逐步求精”的结构化思想解决问题,形成了严格的结构化程序设计方法,具有高级语言的特性。

同时 C 语言可以很方便很直接地对位、字节和地址进行操作,而且在 C 语言中可以直接嵌入汇编语言子程序,可以充分发挥汇编语言的优势,因而又具有低级语言的特性,可以说 C 语言是具有低级语言功能的高级语言。

(2) C 语言具有完备而强大的功能

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 语言提供了强大的图形处理、数值计算和逻辑判断的功能,为处理诸多复杂的现实问题提供了强有力的保障。

(3) C 语言适用范围广泛,具有良好的移植性

C 语言的初衷就是为了解决不同平台的兼容性问题,为跨平台的系统开发带来极大便利,提供了诸如宏等多种机制以便于移植。

(4) C 语言具有简洁灵活的书写形式

ANSI 标准 C 语言一共有 32 个关键字、9 种控制语句,非常简洁。同时 C 语言相对于其他高级语言来讲,几乎对书写不加“任何”限制,非常自由。

(5) C 语言提供了非常丰富的运算符和数据类型

在 ANSI 标准 C 语言中共有 34 个运算符。C 语言把括号、赋值、强制类型转换等都作为运算符处理,从而使 C 语言的运算类型极其丰富。与之相应的 C 语言提供了相当丰富的数据类型,比如整型、实型、字符型、数组类型、指针类型、结构体类型、共用体类型等。正因为如此,才使得 C 语言拥有异常强大的表达能力,可以通过很简洁的语句实现其他高级语言难以实现的功能。

(6) C 语言程序生成代码质量高,程序执行效率高

用 C 语言编译生成的目标代码一般可以达到汇编程序生成的目标代码效率的 80% ~ 90%。相对于其他高级语言来讲,这是难能可贵的。

总之,C 语言既有高级语言的特点,又具有汇编语言的特点;既是一个成功的系统设计语言,又是一个实用的程序设计语言;既能用来编写不依赖于计算机硬件的应用程序,又能用来编写各种系统程序,是一种颇受欢迎、应用广泛的程序设计语言。

1.3 Visual C++ 6.0 开发环境

下面学习如何在 Visual C++ 6.0 中开发 C 语言程序。

安装 Microsoft Visual Studio 6.0 或单独安装 Visual C++ 6.0 之后,选择【开始】→【程序】→【Microsoft Visual Studio 6.0】→【Microsoft Visual C++ 6.0】菜单命令,即可启动 Visual C++ 6.0。

启动 Visual C++ 6.0 并新建程序后的界面如图 1-1 所示。

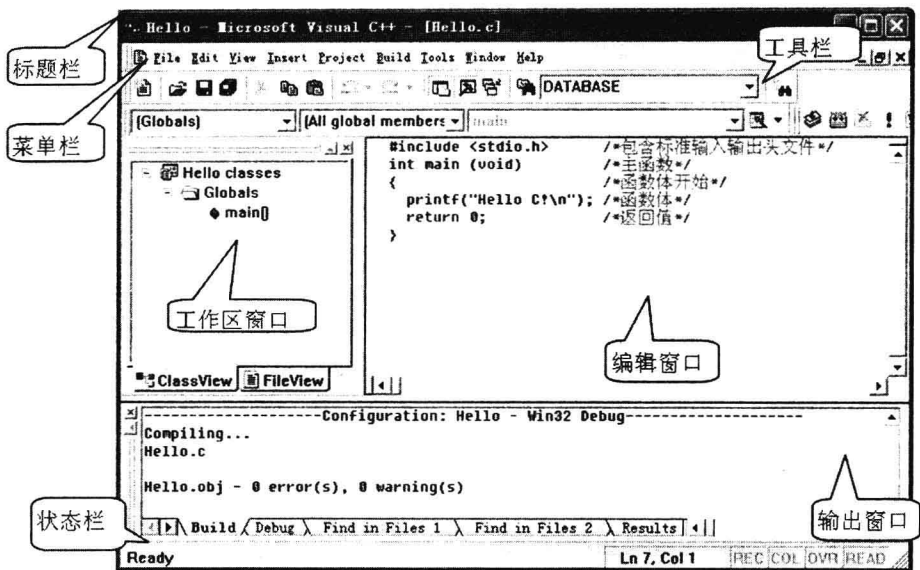


图 1-1

1.3.1 菜单栏

通过菜单栏中的菜单命令,可以轻松地完成编辑程序和资源、编译、链接及调试程序等各项工作。常用菜单如下:

(1)【File】文件菜单:File 菜单包含了各种对文件进行操作的选项,如加载、保存、打印和退出等。

(2)【Edit】编辑菜单:Edit 菜单中的命令是用来使用户便捷地编辑文件内容,如进行删除、复制等操作,其中的大多数命令功能与 Windows 中的标准字处理程序的编辑命令一致。

(3)【View】查看菜单:View 菜单中的命令主要用来改变窗口和工具栏的显示方式,激活调试时所用的各个窗口。

(4)【Insert】插入菜单:Insert 菜单中的命令主要用于项目及资源的创建和添加。

(5)【Project】工程菜单:Project 菜单中的命令主要用于项目的操作,如项目中添加源文件等。

(6)【Build】编译菜单:Build 菜单中的命令主要用于应用程序的编译、链接、调试、运行。

(7)【Tools】工具菜单:Tools 菜单中的命令主要用于选择或制定开发环境中的一些实用工具。

(8)【Window】窗口菜单:Window 菜单中的命令主要用于文档窗口的操作。如排列文档、打开或关闭一个文档窗口、重组或切分文档窗口等。

(9)【Help】帮助菜单:Visual C++ 6.0 同大多数其他的 Windows 应用软件一样,提供了大量详细的帮助信息,而 Help 菜单便是得到这些帮助信息的最有效和最主要的途径。

1.3.2 工具栏

工具栏中提供了大部分常用的操作命令,通过单击工具栏中相应的按钮可以快捷地进行

各种操作。在工具栏中的空白处右击,在弹出的快捷菜单中选择相应的菜单项可以定制(添加或删除)工具栏。

1.3.3 Workspace(工作区)窗口

工作区窗口共有以下 3 个标签,分别代表 3 种视图形式。

(1)【ClassView】类视图:用以显示项目中的所有的类信息。

(2)【ResourceView】资源视图:包含了项目中所有资源的层次列表。每一种资源都有自己的图标。在字符界面或者控制台界面中没有该项,只有 Windows 程序下才出现。

(3)【FileView】文件视图:可将项目中的所有文件分类显示,每一类文件在【FileView】页面中都有自己的目录项。可以在目录项中移动文件,还可以创建新的目录项,以及将一些特殊类型的文件放在该目录项中。

1.3.4 输出窗口

显示程序编译和链接的错误和警告。

1.3.5 编辑窗口

可进行输入、修改以及删除代码等操作。

C 程序的编写和运行流程如图 1-2 所示。

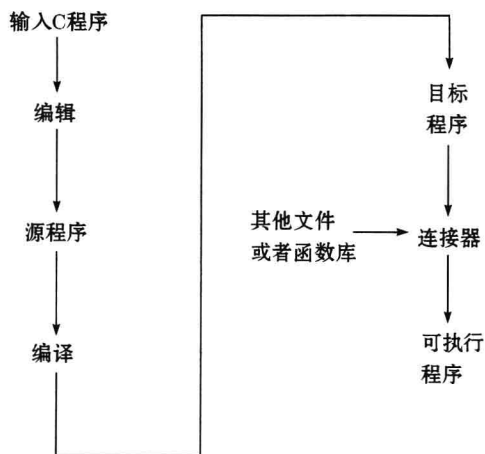


图 1-2

把编写的代码称为源文件或者源代码,输入修改源文件的过程称为编辑,这个过程中还要对源代码进行布局排版,使之美观有层次,并附加一些说明的文字帮助理解代码的含义。这些文字称之为注释,它们仅是起到说明作用,不是代码也不会被执行。经过编辑的源代码,保存后生成后缀名为 .c 的文件,这些源文件并不能够直接运行,它们需要经过编译,把源文件转换为以 .obj 为后缀名的目标文件。此时目标文件再经过一个链接的环节,最终生成以 .exe 为后缀名的可执行文件。能够运行的就是可执行文件。

每个 C 程序必须有且只有一个主函数,也就是 main 函数,它是程序的入口。使用 main 函数有时也作为一种驱动,按次序控制调用其他函数,C 程序是由函数构成的,这使得程序容易

实现模块化。main 函数后的“()”不可省略,表示函数的参数列表。“{”和“}”是函数体开始和结束的标志,不可省略。

图 1-3 是对主函数调用其他函数的说明。

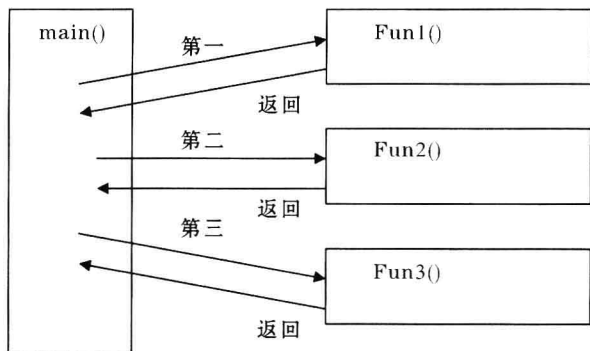


图 1-3 主函数调用其他函数

主函数 main() 在程序中可以放在任何位置,但是编译器都会首先找到它,并从它开始运行。它就像汽车的引擎,控制程序中各部分的执行次序。

图 1-4 是对主函数各部分名称的说明:

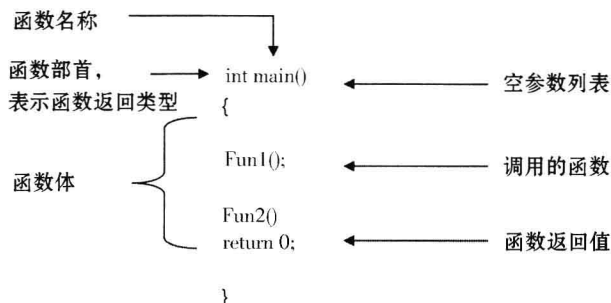


图 1-4

在上述范例中,主函数是 main 函数,main 前面的 int 表示此函数的类型是 int 类型,int 是整数 integer 单词的缩写,表示返回给系统的数据类型是整型数据。返回值是 0,在 return 语句中体现出来了。

从书写代码清晰,便于阅读、理解、维护的角度出发,应遵循以下规则:

(1) 一个说明或一个语句占一行。把空格符、制表符、换行符等统称为空白符。除了字符串、函数名和关键字,C 忽略所有的空白符,在其他地方出现时,只起间隔作用,编译程序对它们忽略不计。因此在程序中使用空白符与否,对程序的编译不产生影响,但在程序中适当的地方使用空白符将增加程序的清晰性和可读性。

比如下面的代码:

【例 1.1】 输出“Hello C!”。

```
int
main(
){
```

```
printf("Hello C! \n")
);
```

```
return 0;
```

```
/* 这样的写法能运行,但是太乱,很不妥 */
```

(2)用“{”和“}”括起来的部分,通常表示了程序的某一层结构。“{”和“}”一般与该结构语句的第一个字母对齐,并单独占一行。

比如下面的代码:

【例 1.2】 输出“Hello C!”。

```
int main()
```

```
{
```

```
printf("Hello C! \n");
```

```
return 0;
```

```
/* 这样的写法能运行,但是阅读起来比较费事 */
```

(3)低一层次的语句通常比高一层次的语句留有一个缩进,然后再书写。一般来说缩进指的是存在两个空格或者一个制表符的空白位置。

比如下面的代码:

【例 1.3】 输出“Hello C!”。

```
int main()
```

```
{
```

```
printf("Hello C! \n");
```

```
return 0;
```

```
}
```

(4)在程序中书写注释,注释可以明白程序做了什么,同样可以增加程序的清晰性和可读性。

比如下面的代码:

【例 1.4】 输出“Hello C!”。

```
int main()
```

```
{
```

```
/* 输出“Hello C!” */
```

```
printf("Hello C! \n");
```

//输出

“Hello C!”

```
return 0;
```

//返回

```
}
```

在例 1.4 中,在程序开头说明程序的意义,这样使程序功能更清晰。在函数体中的每一行后面添加注释,说明该语句的意义。

对于注释我们在后面还有更为详细的讨论,这里简单说明一下,在标准 C 语言中,注释都在“/*……*/”中,但是在 Visual C++ 6.0 中,可以以“//”开头的文字作为注释。

本节通过同一个功能的几个版本展示了编程风格的概念,大家在编程时应力求遵循这些规则,以养成良好的编程风格。



1.4 简单的 C 程序介绍

1.4.1 算法

程序 = 算法 + 数据结构

该公式中,算法是灵魂,数据结构是加工对象,语言是工具。算法是解决“做什么”和“怎么做”的问题。不了解算法就谈不上程序设计。一个好的算法应具有这样的特点:有穷性、确定性、有零个或多个输入、有一个或多个输出、有效性。

算法可用自然语言、流程图、伪代码、PAD 图等方法表示。算法被描述后,要得到运算结果(即算法的输出),必须实现算法。若用计算机实现算法,则需将算法转换成计算机语言程序。

1.4.2 C 程序处理步骤

计算机只能识别和执行二进制的指令,而不能识别和执行高级语言写的指令(源程序)。所以必须先用“编译程序”将源程序翻译成二进制形式的“目标程序”,然后将该目标程序与系统的函数库和其他目标程序连接起来,形成可执行的目标程序。

一个程序从编写到最后得到运行结果要经历以下一些步骤:

上机输入与编辑源程序→对源程序进行编译→与库函数连接→运行目标程序。

(1)用 C 语言编写程序: C 的源程序是以 .c 或 .cpp(C++) 作为后缀的。

(2)对源程序进行编译:编译是以源程序文件为单位分别编译的。目标程序一般以 .obj 作后缀(object 的缩写)。其作用是对源程序进行词法检查和语法检查。编译时对文件中的全部内容进行检查,编译结束后会显示出所有的编译出错信息。一般编译系统给出的出错信息分两种:错误(error)和警告(warning)。

(3)将目标文件连接:在通过编译后,得到一个或多个目标文件。此时要用系统提供的“连接程序(linker)”将一个程序的所有目标程序和系统的库文件以及系统提供的其他信息连接起来,最终形成一个可执行的二进制文件,它的后缀是 .exe,是可以直接执行的。

(4)运行程序:运行最终形成的可执行文件(.exe 文件),得到运行结果。

(5)分析运行结果:如果运行结果不正确,应检查程序或算法是否有问题。

【例 1.5】 一个简单的 C 程序,在屏幕上显示一个字符串信息,如图 1-5 所示。

(1)打开 Visual C++ 6.0 环境,建立一个控制台工程,其名称为 Example_1_5_DisplayInf。

(2)添加 C++ 源文件,并编写如下代码:

```
01  #include <stdio. h >
02  void main(void )
03  {
04      printf(" I am a student. \n" );          /* 输出字符串 */
05      return;
06  }
```

编译过程:



- (1) 启动 Visual C++, 选择“File”菜单中的“New”命令, 选择“Files”标签中的“C++ Source File”选项。
- (2) 选择源程序存放的目录和输入源程序名, 单击“OK”。
- (3) 在编辑器中编写源程序。
- (4) 单击“Build”中的“Rebuild All”编译源程序, 若编译通过, 单击执行按钮, 在 DOS 屏上看结果, 任按一键返回编辑器。

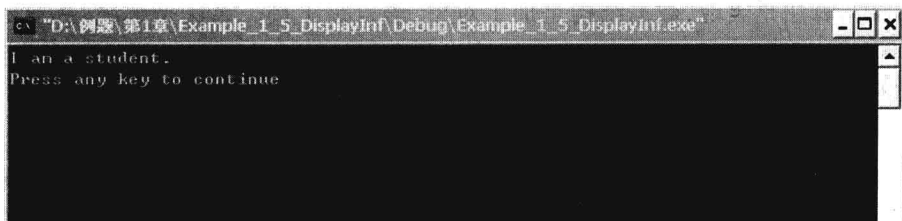


图 1-5 例 1.5 运行结果

1.5 C 程序的构成

(1) 一个 C 程序可以由一个程序单位或多个程序单位构成。每一个程序单位作为一个文件。在程序编译时, 编译系统分别对各个文件进行编译, 因此, 一个文件是一个编译单元。

(2) 在一个程序单位中, 可以包括以下几个部分:

① 预处理命令。比如例 1.5 中第一行的 `#include` 命令。

② 全局声明部分(在函数外的声明部分)。在这部分中包括对用户自己定义的数据类型的声明和程序中所用到的变量的定义。

③ 函数。函数是实现操作的部分, 因此函数是程序中必须有的和最基本的组成部分。每一个程序必须包括一个或多个函数, 其中必须有一个(而且只能有一个)主函数(main 函数)。

(3) 一个函数由两部分组成:

① 函数首部, 即函数的第一行。包括函数名、函数类型、函数参数(形参)名、参数类型。

② 函数体, 即函数首部下面的大括号内的部分。如果在一个函数中有多个大括号, 则最外层的一对 `{ }` 为函数体的范围。

函数体一般包括:

局部声明部分(在函数内的声明部分)。包括对本函数中所用到的类型、函数的声明和变量的定义。对数据的声明既可以放在函数之外(其作用范围是全局的), 也可以放在函数内(其作用范围是局部的, 只在本函数内有效)。

执行部分。由若干个执行语句组成, 用来进行有关的操作以实现函数功能。

(4) 语句包括两类。一类是声明语句, 另一类是执行语句。C 对每一种语句赋予一种特定的功能。语句是实现操作的基本成分, 显然, 没有语句的函数是没有意义的。C 语句必须以分号结束。

(5) 一个 C 程序总是从 main 函数开始执行的, 而不论 main 函数在整个程序中的位置如何。



(6) C 程序书写格式自由,一行内可以写几个语句,一个语句可以分写在多行上。

(7) 一个好的、有使用价值的源程序都应当加上必要的注释,以增加程序的可读性。C 语言的注释用“//”作注释时,有效范围只有一行,即本行有效,不能跨行。而用“/*……*/”作注释时有效范围为多行。只要在开始处有一个“/*”,在最后一行结束处有一个“*/”即可。因此,一般习惯是内容较少的简单注释常用“//”,内容较长的常用“/*……*/”。

1.6 本章小结

本章简单介绍了 C 语言的历史以及特点,重点介绍了 Visual C++ 6.0 开发环境和 C 语言程序的基本构成和书写风格。要务必注意 C 语言的书写风格,因为 C 语言本身的书写非常的自由,初学时稍有不慎将对今后的学习产生不良的影响。反之,从一开始就养成良好的书写风格,将为计算机语言的学习打下良好的基础。

1.7 练习题

简答题

- (1) 简述你对 C 语言和 C++ 语言的认识。
- (2) C 程序的基本构成有哪些?

数据是计算机处理的对象,在 C 语言中,通过变量和常量来存储和管理数据。本章将重点讨论变量和常量的概念和类型。

2.1 计算机中的数据

提到数据大家就会在脑海中浮现出诸如“1、2”这样的信息来,其实数据是一个很广义的概念,不是简单的数字呈现。在本节中我们要探讨在计算机中数据是如何表示的。

2.1.1 计算机的数据观

通过文字、声音、影像等方式,计算机可以描述形形色色的事物,给我们呈现出一个多姿多彩的世界。那么计算机是如何实现这些功能的呢?

其实,计算机只能感知电压、电流等物理量,这在《计算机组成原理》等书籍中已经有很深入的探讨,这里不再详细讨论。因而电压的高低、电流的通断等状态就很容易表示出来了。如果我们用“0”和“1”来表示这种状态的话,计算机中的数据就可以用“01”串来表示,实际上这就是二进制。

2.1.2 进制

在 2.1.1 节中,我们已经提到了二进制,那么如何理解二进制呢?它和十进制有什么联系呢?还有其他的进制么?这就是本节要讨论的问题。

我们认识数据总是从整数开始,这次也不例外,我们就以数字来讨论进制。

(1) 十进制

十进制使用十个数字(0、1、2、3、4、5、6、7、8、9)记数,基数为 10,逢十进一。由于我们是从十进制开始认识数字的,因为太过于熟悉,往往忽略了十进制的本质。但深入思考就能够发现 0~9 这十个数字实际上仅仅是符号而已。这十个符号表示了十种状态,又经过特定的组合形成了特定的记数方法。

(2) 二进制

二进制以 2 为基数,只用 0 和 1 两个数字表示,逢二进一。二进制与十进制数遵循一样的运算规则,但比十进制更简单。例如:

①加法: $0+0=0$; $0+1=1$; $1+0=1$; $1+1=0$ 。