高等院校计算机应用系列教材

C 语言程序设计

(微课版)

赵　彩　杨宏霞　主　编
丁　凰　许大炜　田文文　薛　薇　副主编

北　京

内 容 简 介

本教材针对应用型人才培养目标，从学生思维方式、理解能力及后续课程中的应用等方面出发编写而成。全书分为 9 章，主要内容包括 C 语言概述，数据类型、运算符及表达式，常用输入输出函数，程序控制结构，数组，函数，指针，结构体与共用体，文件操作等。本书还在每一章的“编程经验”模块中融入各种编程小技巧，可使读者在学习过程中少走弯路，在掌握 C 语言技术精髓的同时快速提升 C 语言程序开发技能。

作为一本微课教材，本书配备了 121 集与实例同步的微课视频，学生可以跟着视频学 C 语言，高效、快捷。另外，本书配套了丰富的教学资源，如实例源代码、PPT 教学课件和课后习题答案，从而方便教师教学和读者自学。与本书同步的实验教材《C 语言程序设计实践教程》则能够方便读者深入学习 C 语言并进行上机操作。

本教材既可以作为高等学校本科及专科学生的 C 语言程序设计教材，也可以作为自学者的参考用书，同时可供各类计算机等级考试人员复习参考。

本书对应的电子课件、习题答案和实例源代码可以到 http://www.tupwk.com.cn/downpage 网站下载，也可通过扫描前言中的二维码下载。扫描封底二维码可以直接观看微课视频。

图书在版编目(CIP)数据

C 语言程序设计：微课版 / 赵彩，杨宏霞主编. —北京：清华大学出版社，2021.7
高等院校计算机应用系列教材
ISBN 978-7-302-58439-1

Ⅰ. ①C… Ⅱ. ①赵… ②杨… Ⅲ. ①C 语言—程序设计 Ⅳ. ①TP312.8

中国版本图书馆 CIP 数据核字(2021)第 102326 号

责任编辑：胡辰浩
封面设计：高娟妮
版式设计：妙思品位
责任校对：成凤进
责任印制：丛怀宇

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

印 装 者：三河市科茂嘉荣印务有限公司
经　　销：全国新华书店
开　　本：185mm×260mm　　**印　　张：**19.25　　**字　　数：**493 千字
版　　次：2021 年 7 月第 1 版　　**印　　次：**2021 年 7 月第 1 次印刷
定　　价：79.00 元

产品编号：092652-01

前　言

C 语言是世界上最重要、影响最深远的程序设计语言。在 C 语言的基础上，诞生了 C++、Java 等各种极具生产力的语言。根据世界编程语言排行榜，C 语言近二三十年来一直排在编程语言的第一位或第二位。只有真正掌握了 C 语言，才能深入理解当今计算机系统的工作原理。

程序设计既是一种技术，也是一项工程。作为一本程序设计教材，不仅要介绍关于 C 语言的基本语法知识，还要强调程序设计思想方法的掌握，并且着眼于应用现代软件工程的思想进行程序开发能力的训练。如何解决好以上三个方面的衔接，将它们有机地结合起来，是当前程序设计教材需要解决的一个重要问题，也是一个难点。本书在以下几个方面做了补充：

1) 讲解 C 语言最基本、最常用的内容，重点主要放在语言本身的难点和程序设计的思想、技巧方面；各章节之间密切结合并且以阶梯式前进，使读者在学习过程中能够循序渐进。

2) 书中每一章节的实例都配备了教学讲解微视频，使用手机扫描二维码即可观看、学习，能够快速引导初学者入门，感受编程的快乐和成就感，进一步增强学习的信心。

3) 通过第 9 章的综合案例提供完整的软件系统开发过程，将每一章的知识点融入其中，使读者学以致用，最终能够开发出学生信息管理系统，并且得到设计软件的实践经验。

4) 每一章的“本章小结”和“编程经验”部分总结了一些易错的概念和知识点。通过介绍一些软件开发过程中的编程经验，本书使读者在学习过程中就能领悟到高质量的 C 语言编程知识。

5) 本书加入了计算机等级考试中的部分经典试题，使学生学完后，不仅能够掌握 C 语言的理论知识，具备一定的实践能力，而且能够为参加计算机等级考试打下基础。

本书由西安交通大学城市学院的赵彩、杨宏霞、丁凰、许大炜、田文文、薛薇老师编写。感谢西安交通大学陆丽娜老师对本书提出了一些很好的建议。感谢西安交通大学城市学院领导对我们的关心、支持和帮助。

欢迎使用本书进入美妙的 C 语言世界。C 语言博大精深，本书只是从实用易懂的角度来描述 C 语言，希望能够抛砖引玉，使读者尽可能达到一种专业的编程境界。

由于笔者的水平和编程经验有限，加之时间比较紧促，本书尚有很多不足之处，希望能够得到专家和读者的指正。我们的信箱是 992116@qq.com，电话是 010-62796045。

本书配套的电子课件、习题答案和实例源代码可以到 http://www.tupwk.com.cn/downpage 网站下载，也可通过扫描下方的二维码下载。扫描下方二维码可以直接观看微课视频。

电子课件、习题答案和实例源代码

多媒体视频教程

作　者

2021 年 2 月

目　录

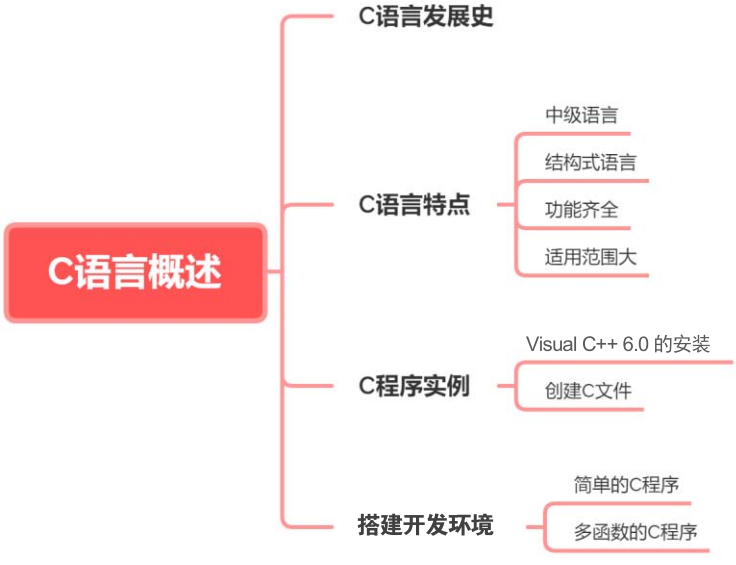
第 1 章

C语言概述

本章概览

本章首先简单介绍 C 语言的发展、特点和强大应用，然后详细介绍开发工具 Visual C++ 6.0 的安装与配置，最后通过三个简单程序带领大家体验 C 语言的编程过程并了解程序的基本组成。“千里之行，始于足下！”下面就开始我们的 C 语言编程之旅吧！

知识框架



1.1 C 语言发展史

1.1.1 程序语言简述

1. 机器语言

计算机诞生之初，人们只能直接用二进制形式的机器语言写程序。对于人类而言，二进制的机器语言很不方便，用它写程序非常困难，不但工作效率极低，程序的正确性难以保证，而且即便有错误也很难辨认和改正。下面是一台假想的计算机上的指令序列：

```
0000000100000001000 --将存储单元 1000 中的数据装入寄存器 0
0000000100010001010 --将存储单元 1010 中的数据装入寄存器 1
0000010100000000001 --将寄存器 1 中的数据乘到寄存器 0 中的原有数据上
0000000100010001100 --将存储单元 1100 中的数据装入寄存器 1
0000010000000000001 --将寄存器 1 中的数据加到寄存器 0 中的原有数据上
0000001000000001110 --将寄存器 0 中的数据存入存储单元 1110
```

以上指令序列描述的是计算算术表达式 a × b + c(这里的符号 a、b、c 分别代表地址为 1000 、1010 和 1100 的存储单元)，而后将结果保存到存储单元 1110 的计算过程(程序)。复杂的程序如果用二进制的机器语言来写的话，将十分困难且难以理解。

2. 汇编语言

为缓解使用机器语言的问题，人们研究出了以符号形式表示且使用起来相对容易的汇编语言。使用汇编语言编写的程序需要经专用软件(汇编系统)加工并翻译成二进制的机器指令后，才能在计算机上使用。

下面是使用某种假想的汇编语言写出的程序，它能完成与上述指令系列相同的工作：

```
load 0 a --将存储单元 a 中的数据装入寄存器 0
load 1 b --将存储单元 b 中的数据装入寄存器 1
mult 0 1 --将寄存器 1 中的数据乘到寄存器 0 中的原有数据上
load 1 c --将存储单元 c 中的数据装入寄存器 1
add 0 1 --将寄存器 1 中的数据加到寄存器 0 中的原有数据上
save 0 d --将寄存器 0 中的数据存入存储单元 d
```

汇编语言的每条指令对应一条机器语言指令，但采用了助记的符号名，存储单元也用符号形式的名称来表示，这样每条指令的意义就更容易理解和把握了。但是，使用汇编语言编写的程序仍然完全没有结构，而仅仅是使用许多这样的指令堆积形成的长长的指令序列，可谓一盘散沙。因此，复杂程序作为整体仍然难以理解。

3. 高级程序语言

为克服低级语言(机器语言与汇编语言)的缺点，1954 年诞生了第一个高级程序语言 FORTRAN，它采用接近于人们习惯使用的自然语言，用类似数学表达式的形式描述数据计算。

FORTRAN 不仅提供了有类型的变量，而且提供了一些流程控制机制，如循环和子程序等。这些高级机制使编程者可以摆脱许多具体细节，方便了复杂程序的书写，写出的程序更容易阅读，错误也更容易辨认和改正了。FORTRAN 语言在诞生后受到广泛欢迎。

高级程序语言更接近人们习惯的描述形式，更容易被接受，从而促使更多的人加入程序设计活动中。使用高级语言书编写程序的效率更高，使人们开发出更多的应用系统，这反过来又大大推动了计算机应用的发展。计算机应用的发展又推动了计算机工业的大发展。可以说，高级程序语言的诞生和发展，对于计算机发展到今天起到极其重要的作用。从 FORTRAN 语言诞生至今，人们提出的编程语言超过千种，其中大部分只是实验性的，只有少数编程语言得到广泛使用，如 C、C++、Pascal、Ada、Java、LISP、Smalltalk、PROLOG 等，这些编程语言都曾在计算机程序语言或计算机的发展历史上起过(有些仍在起着)极其重要的作用。随着时代的发展，如今绝大部分程序都是使用高级程序语言编写的。人们也已习惯于用程序设计语言特指各种高级程序语言。使用高级程序语言(例如 C 语言)描述前面同样的程序片段时，只需要一行代码：

```
d = a * b + c;
```

这表示要求计算机求出等于符号右边的表达式，然后将计算结果存入由 d 代表的存储单元中。这种表示方式与人们熟悉的数学形式直接对应，因此更容易阅读和理解。高级程序语言完全采用符号形式，使人们可以摆脱难用的二进制形式和具体计算机的细节。此外，高级程序语言还提供了许多高级的程序结构，从而方便人们组织复杂的程序。

计算机能否理解使用这些高级程序语言编写的指令呢？答案是不能，那么如何使计算机执行用高级程序语言编写的程序指令呢？我们需要将这些程序指令翻译成机器指令。编译器就是这样一种用来完成上述翻译任务的特殊程序，每一种编程语言都有不同的编译器。编译器有如下两种工作方式：

- 编译方式。人们首先针对具体的编程语言(例如 C 语言)开发出翻译软件(程序)，其功能是将采用这种高级程序语言编写的程序翻译为所使用计算机的机器语言的等价程序。这样，在使用这种高级编程语言写出程序后，只要将它们发送给翻译程序，就能得到与之对应的机器语言程序。此后，只要命令计算机执行机器语言程序，计算机就能执行我们所要完成的工作了。
- 解释方式。人们首先针对具体的高级程序语言开发解释软件，其功能是逐条读入使用高级程序语言编写的程序，并一步步地按照程序的要求开展工作，完成程序描述的计算任务。有了解释软件，直接将写好的程序发送给运行着解释软件的计算机，就可以完成程序描述的任务了。

1.1.2　C 语言的发展过程

C 语言是由 UNIX 的研制者 Dennis Ritchie(丹尼斯•里奇)和 Ken Thompson(肯•汤普逊)于 1970 年在 BCPL 语言(简称 B 语言)的基础上发展并完善起来的一种编程语言。20 世纪 70 年代初期，AT&T Bell 实验室的程序员丹尼斯•里奇第一次把 B 语言改为 C 语言。

最初，C 语言运行于 AT&T 的多用户、多任务的 UNIX 操作系统上。后来，丹尼斯•里奇用 C 语言改写了 UNIX C 的编译程序，UNIX 操作系统的另一名研制者肯•汤普逊又用 C 语言成

功改写了 UNIX，从此开创了编程史上的新篇章。UNIX 成为第一个不是使用汇编语言编写的主流操作系统。

随着计算机应用的发展，人们强烈地希望 C 语言能成为一种更安全可靠、不依赖于具体计算机和操作系统(如 UNIX)的标准程序设计语言。美国国家标准局(ANSI)于 20 世纪 80 年代专门成立了小组来研究 C 语言的标准化问题，并于 1988 年颁布了 ANSI C 标准。这个标准已被国际标准化组织和各国标准化机构接受，也被采纳为中国国家标准。此后，经过人们持续的努力和推动，于 1999 年通过了 ISO/IEC 9899:1999 标准(一般称为 C99)，这一新标准对 ANSI C 做了一些小的修订和扩充。

1.2 C 语言特点

C 语言之所以能被计算机界广泛接受，缘于 C 语言自身的如下特点。

(1) C 语言是中级语言。C 语言把高级语言的基本结构和语句与低级语言的实用性结合了起来。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 语言是结构式语言。结构式语言的显著特点是代码与数据实现了分隔，换言之，程序的各个部分除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、维护及调试。C 语言是以函数形式提供给用户的，这些函数除方便调用外，还具有多种循环和条件语句，用于控制程序流向，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，还引入了指针的概念，可使程序效率更高。另外，C 语言的计算功能、逻辑判断功能也比较强大。

(4) C 语言适用范围大。C 语言适用于多种操作系统，如 Windows、DOS、UNIX 等，并且适用于多种机型。

对于编写需要结合硬件进行操作的场合，C 语言明显优于其他编程语言，一些大型应用软件就是使用 C 语言编写的。

C 语言得到业界的广泛赞许。一方面，C 语言在程序设计语言研究领域具有一定价值，由它引出了不少后继语言，另有许多语言吸收了 C 语言的不少优点。另一方面，C 语言对计算机工业和应用的发展也起到十分重要的推动作用。正是由于这些原因，C 语言的设计者获得了世界计算机科学技术界的最高奖——图灵奖。

1.3 简单的 C 程序实例

1.3.1 C 语言程序的构成与格式

我们先通过几个简单的程序来了解 C 语言程序的编写特点。

【例 1-1】输出“This is a C program.”

```
#include <stdio.h>                    // 编译预处理命令
```

```
void main( )                          // 定义 main 函数
{                                     // 函数开始的标志
    printf("This is a C program.\n");        // 输出一行信息
}                                     // 函数结束标志
```

程序运行结果如图 1-1 所示。

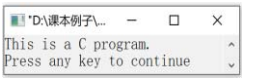


图 1-1　程序运行结果

(1) 上面这个简单的 C 语言程序可分为两个基本部分：第一行是特殊行，include <stdio.h>说明程序需要用到 C 语言系统提供的标准功能(参考标准库文件 stdio.h)。其他几行是程序的基本部分。

(2) main 是主函数名，void 是函数类型。每个 C 语言程序都必须有一个 main 函数，main 函数是每个 C 语言程序的执行起点(入口点)。

main 函数的函数体是用一对花括号{ }括起来的。{和}是函数的开始和结束标志，不可省略。main 函数中的所有操作(或语句)都在这对花括号之间。也就是说，main 函数的所有操作都在 main 函数体中。

(3) printf 是 C 语言提供的库函数，功能是进行输出(显示在计算机屏幕上)，此处用于输出字符串"This is a C program.\n"，也就是在计算机屏幕上显示“This is a C program.”。其中的\n 表示输出后换行。

(4) 注意，函数体中的每条语句必须以分号结束。

(5) 在使用库函数中的输入输出函数时，必须提供有关这些函数的信息：#include <stdio.h>。

【例 1-2】用 C 语言解决求和问题。

```
#include <stdio.h>
void main( )
{
    int a,b,sum;                      // 定义变量 a、b、sum
    a=123;b=456;                      /*以下 3 行为 C 语句*/
    sum=a+b;
    printf("sum=%d\n",sum);          // %d 为格式控制符
}
```

程序运行结果如图 1-2 所示。

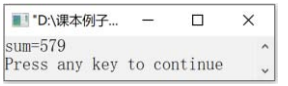


图 1-2　程序运行结果

(1) 同样，这个程序也必须包含 main 函数作为程序的执行起点。{和}之间为 main 函数的函数体，main 函数的所有操作都在 main 函数体中。

(2) 注释可以使用//以及/*和*/来标识。从//开始到换行符结束的内容为单行注释，使用/*和*/括起来的内容为段落注释。注释只是为了改善程序的可读性，在编译、运行时不起作用。因此，注释可以使用汉字或英文字符，既可以出现在一行中的最右侧，也可以单独成为一行。注释允许占用多行，只是需要注意/*与*/必须配对使用，一般不要嵌套使用。

(3) 语句 t a,b,sum;定义了三个整数类型的变量 a、b、sum。C 语言中的变量必须先声明再使用。

(4) a=123;b=456;是两条赋值语句，作用是将整数 123 赋给整型变量 a，并将整数 456 赋给整型变量 b。注意这是两条语句，每条语句均以;结束。也可以将这两条语句写成两行：

```
a=123;
b=456;
```

由此可见，C 语言程序的书写可以相当随意，但是为了保证代码容易阅读，建议遵循一定的规范。

(5) sum=a+b;的作用是将 a、b 两个变量的值相加，然后将结果赋给整型变量 sum。此时，sum 变量的值为 579。

(6) printf("sum=%d\n",sum);的作用是调用库函数 printf 以输出 sum 变量的值。%d 为格式控制符，表示 sum 变量的值将以十进制整数形式输出。

【例 1-3】输入两个数，输出其中较大的那个数。

```
#include <stdio.h>
void main( )
{                                   // main 函数体开始
    int a,b,c;                      // 定义变量 a、b、c
    scanf("%d,%d",&a,&b);           // 输入变量 a 和 b 的值
    c=max(a,b);                     // 调用 max 函数，将调用结果赋给变量 c
    printf("max=%d",c);             // 输出变量 c 的值
}                                   // main 函数体结束
int max(int x,int y)                // max 函数用于判断并返回两个数中较大的那个数
{                                   // max 函数体开始
    int z;                          // 定义函数体中的变量 z
    if(x>y) z=x;                    // 若 x>y，将 x 的值赋给变量 z
    else z=y;                       // 否则，将 y 的值赋给变量 z
    return z;                       // 将变量 z 的值返回到调用 max 函数的地方
}                                   // max 函数体结束
```

程序运行结果如图 1-3 所示。

图 1-3　程序运行结果

(1) 这个程序包括两个函数。其中，main 函数仍然是整个程序的执行起点，max 函数的功能是计算两个数中较大的那个数。

(2) main 函数首先调用 scanf 函数以获得两个整数，将它们存入变量 a 和 b，然后调用 max 函数以获得两个数中较大的那个数并赋给变量 c，最后输出变量 c 的值(结果)。

(3) int max(int x,int y)是 max 函数的函数头，这表明 max 函数将获得两个整型参数并返回一个整数。

(4) max 函数同样使用花括号{}将函数体括起来。max 函数的函数体是函数功能的具体实现：从参数中获得数据，处理后得到结果，然后将结果返回给调用函数 main。

例 1-3 还表明除了调用库函数之外，还可以调用用户自己定义并编写的函数。

1.3.2　C 语言程序的结构

结合上面的三个例子，大家对 C 语言程序的基本组成和形式(程序结构)应该有了初步的了解：

(1) C 语言程序由多个函数构成。

① C 语言程序至少需要包含一个 main 函数，也可以包含一个 main 函数和若干其他函数。函数是 C 语言程序的基本单位。

② 被调用的函数可以是系统提供的库函数，也可以是用户根据需要自行设计编写的函数。C 语言是函数式语言，程序的全部工作都是由各个函数完成的。编写 C 语言程序就是编写一个个函数。

③ 函数库非常丰富，ANSI C 提供了 100 多个库函数。

(2) main 函数(主函数)是每个程序的执行起点。

main 函数既可以放在整个程序的最前面，也可以放在整个程序的最后，还可以放在其他函数之间。但 C 程序总是从 main 函数开始执行，而不论 main 函数处在程序中的什么位置。

(3) 函数由函数头和函数体两部分组成。

① 函数头。函数的第一行，格式如下：

返回值类型 函数名([函数参数类型 1 函数参数名 1],…,[函数参数类型 *n*，函数参数名 *n*])

例如 int max(int x, int y);。

注意：函数可以没有参数，但是后面的一对圆括号不能省略，这是规定。

② 函数体。在函数头的下方，使用一对花括号括起来的部分为函数体。如果函数体内有多对花括号，那么最外层才是函数体的范围。函数体一般都包括声明部分和执行部分。

```
{
  [声明部分]：定义函数需要使用的变量。
  [执行部分]：由若干语句组成的命令序列(可在其中调用其他函数)。
}
```

(4) C 语言程序书写自由。

① 在书写 C 语言程序时，一行可以写一条或多条语句，一条语句也可以分多行书写。

② C 语言程序没有行号，并且没有像 FORTRAN、COBOL 那样严格规定书写格式(语句必须从某一列开始)。

③ 每条语句在最后必须以分号结束。

(5) 可以使用//以及/*和*/对 C 语言程序中的任何部分进行注释。

注释可以提高程序的可读性，使用注释是编程人员必须养成的良好习惯。使用注释的原因如下：

① 编写好的程序往往需要修改、完善，事实上，并不存在任何不需要修改和完善的应用系统。很多人会发现，自己编写的程序在经历了一些时间以后，由于缺乏必要的文档和注释，最后连自己都很难再读懂，因而需要花费大量时间重新思考和理解原来的程序。如果一开始就对程序进行注释，刚开始虽然麻烦一些，但日后可以节省大量的时间。

② 实际的应用系统往往由多人合作开发，程序文档、注释是其中重要的交流工具。

(6) C 语言本身不提供输入输出语句，输入输出操作是通过调用库函数(如 scanf、printf 函数)完成的。

输入输出操作涉及具体的计算机硬件，把输入输出操作放在函数中进行处理，可以简化 C 语言的编译系统，便于 C 语言在各种计算机上实现。不同的计算机系统需要对函数库中的函数进行不同的处理，以便实现相同或类似的功能。

不同的计算机系统除了提供函数库中的标准函数之外，还会按照硬件情况提供一些专用函数。因此，不同计算机系统提供的函数在数量和功能上也存在一定差异。

1.3.3 良好的编程风格

C 语言是一种“格式自由”的编程语言，除若干简单限制外，编写程序的人完全可以根据自己的想法和需要选择程序格式，确定在哪里换行，在哪里增加空格等。这些格式变化并不影响程序的意义，但是，不规定程序格式并不说明格式不重要。对于阅读而言，程序格式非常重要。在多年的程序设计实践中，人们在这方面的认识得到了统一。由于程序可能很长，结构可能很复杂，因此程序必须采用良好的格式来编写，所用格式应很好体现程序的层次结构，反映各个部分之间的关系。

关于程序格式，人们普遍采用的方式如下：

- 在程序里适当加入空行，分隔程序中处于同一层次的不同部分。
- 将同一层次的不同部分对齐排列，下一层次的内容通过适当地添加空格(在一行的开头添加空格)，使程序结构更清晰。
- 在程序里添加一些说明性信息。

大家在刚开始学习程序设计时就应养成注意程序格式的习惯。对于小的程序，虽然采用良好格式的优势并不明显，但对于稍大一点的程序，情况就不一样了。有人为了方便，根本不关心程序格式，想的只是少输入几个空格或换行，这样做的结果就是导致自己在随后的程序调试检查中遇到更多麻烦。正因为如此，这里特别提醒读者：注意程序格式，从一开始编写最简单的程序时就注意养成好习惯。目前多数程序设计语言(包括 C 语言)都是格式自由语言，这使人们能够十分方便地根据自己的需要和习惯写出具有良好格式的程序。总之，优秀的程序员往往：

- 使用 Tab 键缩进代码。
- 对齐各个层次的花括号。
- 添加足够的注释并使用适当的空行。

1.4 搭建 Visual C++ 6.0 开发环境

“工欲善其事，必先利其器”，本节将详细介绍学习 C 语言程序开发的常用工具：Visual C++ 6.0。Visual C++ 6.0 是功能强大的可视化软件开发工具，已将程序的代码编辑、编译、连接和调试等功能集于一身。

1.4.1 Visual C++ 6.0 的安装

Visual C++ 6.0(中文版)的具体安装步骤如下：

(1) 双击 Visual C++ 6.0 安装文件夹中的 SETUP.EXE 安装文件，如图 1-4 所示。弹出如图 1-5 所示的“程序兼容性助手”界面，单击“运行程序”按钮，进入安装向导界面。

(2) 在如图 1-6 所示的界面中单击“下一步”按钮，进入如图 1-7 所示的“最终用户许可协议”界面，选中“接受协议”单选按钮，然后单击“下一步”按钮。

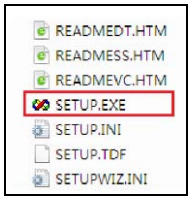


图 1-4　双击安装文件

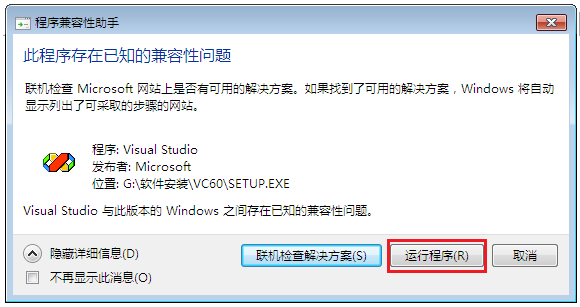


图 1-5　单击“运行程序”按钮

图 1-6　进入安装向导

图 1-7　“最终用户许可协议”界面

(3) 进入如图 1-8 所示的“产品号和用户 ID”界面，根据自身情况填写姓名和公司名称，也可采用默认设置，单击“下一步”按钮。

(4) 进入“Visual C++ 6.0 中文企业版”界面，如图 1-9 所示。选中“安装 Visual C++ 6.0 中文企业版”单选按钮，然后单击“下一步”按钮。

图 1-8　“产品号和用户 ID”界面

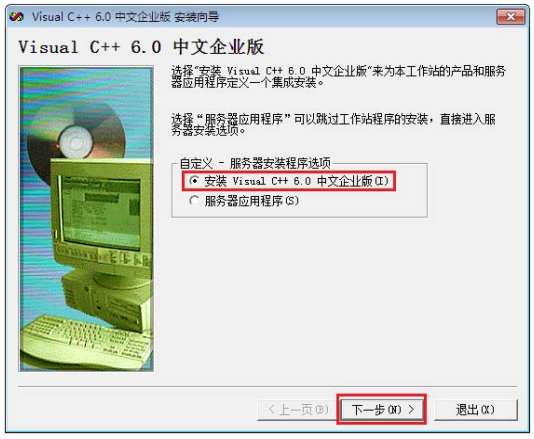


图 1-9　“Visual C++ 6.0 中文企业版”界面

(5) 进入“选择公用安装文件夹”界面，如图 1-10 所示。公用文件默认存储在 C 盘中，单

击“浏览”按钮，可选择其他存放位置。单击“下一步”按钮，进入如图 1-11 所示的安装程序的欢迎界面，单击“继续”按钮。

(6) 进入如图 1-12 所示的产品 ID 确认界面。此处显示了我们将要安装的 Visual C++ 6.0 软件的产品 ID，在向微软请求技术支持时，需要提供软件的产品 ID，单击“确定”按钮。

(7) 如果读者的计算机中安装过 Visual C++ 6.0，尽管已经卸载，在重新安装时也仍会提示如图 1-13 所示的信息。安装软件检测到系统之前安装过 Visual C++ 6.0，如果想要覆盖安装的话，单击“是”按钮；如果要将 Visual C++ 6.0 安装到其他位置，单击“否”按钮。这里单击“是”按钮，继续安装。

图 1-10 “选择公用安装文件夹”界面

图 1-11 安装程序的欢迎界面

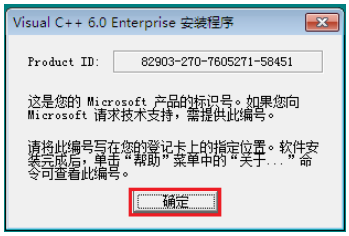


图 1-12 产品 ID 确认界面

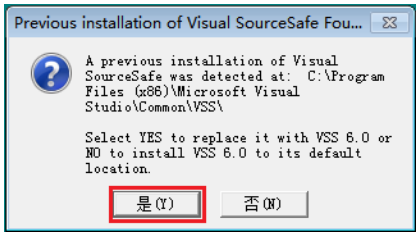


图 1-13 覆盖以前的安装

(8) 进入如图 1-14 所示的安装类型选择界面。Typical 为典型安装，Custom 为自定义安装，这里选择 Typical 安装类型。

(9) 进入如图 1-15 所示的环境变量注册界面。选中 Register Environment Variables 复选框以注册环境变量，单击 OK 按钮。

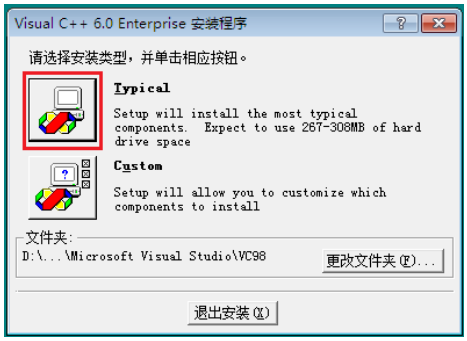


图 1-14 安装类型选择界面

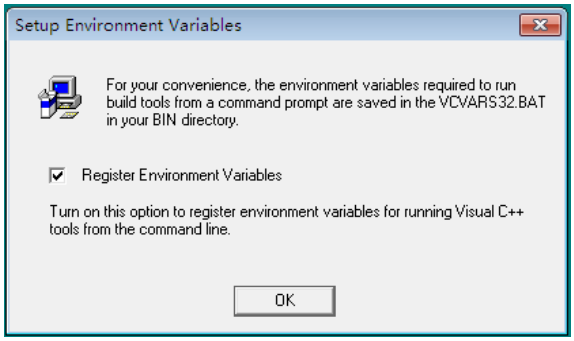


图 1-15 环境变量注册界面

(10) 进入如图 1-16 所示的 Visual C++ 6.0 安装进度界面，当进度条达到 100%时，表示安装成功，如图 1-17 所示。

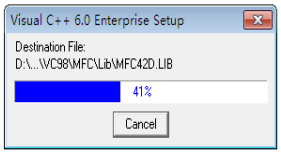


图 1-16　安装进度界面

图 1-17　安装成功

1.4.2　使用 Visual C++ 6.0 创建 C 文件

安装完之后，就可以使用 Visual C++ 6.0 创建 C 文件了，步骤如下：

(1) 选择“开始”菜单中的 Microsoft Visual C++ 6.0 命令，如图 1-18 所示。

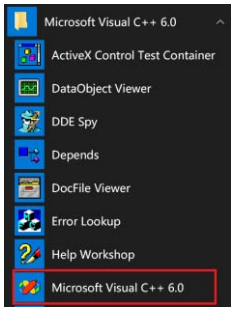


图 1-18　启动 Visual C++ 6.0 开发环境

(2) 进入 Visual C++ 6.0 开发界面，如图 1-19 所示。

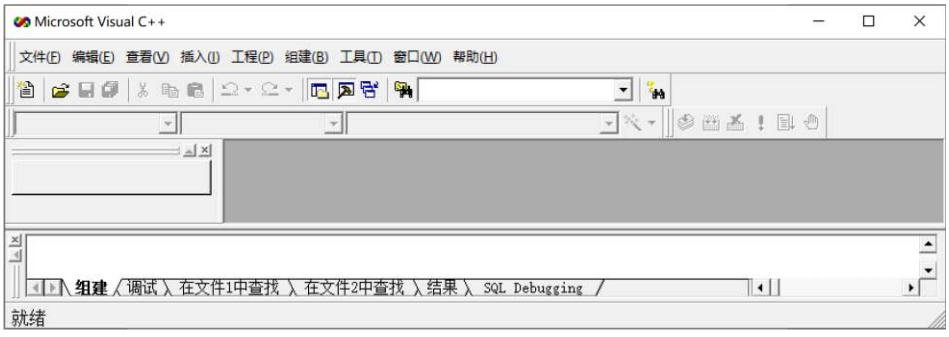


图 1-19　Visual C++ 6.0 开发界面

(3) 在编写程序之前，我们需要新建程序文件。在 Visual C++ 6.0 开发界面中选择“文件”菜单中的“新建”选项，如图 1-20 所示，或者按 Ctrl+N 快捷键，打开“新建”对话框。

图 1-20　选择“新建”选项

(4) 在“新建”对话框中选择想要创建的文件类型。首先选择“文件”选项卡，下方的列表框中显示了可以创建的文件类型。因为要创建 C 源文件，所以选择 C++ Source File 选项。在右侧的“文件名”文本框中输入所要创建的文件的名称，例如 hello.c。在“位置”文本框设置源文件的保存地址，可以通过单击右边的...按钮来修改源文件的存储位置。具体操作如图 1-21 所示。

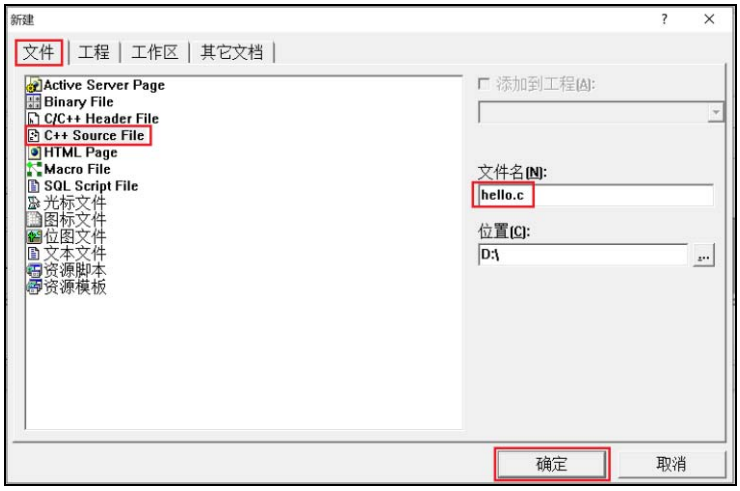


图 1-21　创建 C 源文件

(5) 输入文件名称并指定源文件的保存地址后，单击“确定”按钮，即可成功创建一个新的 C 源文件。此时，我们可以在开发环境中看到创建好的 C 源文件，如图 1-22 所示。

(6) 现在开始编写程序代码。程序代码的输入界面如图 1-23 所示。

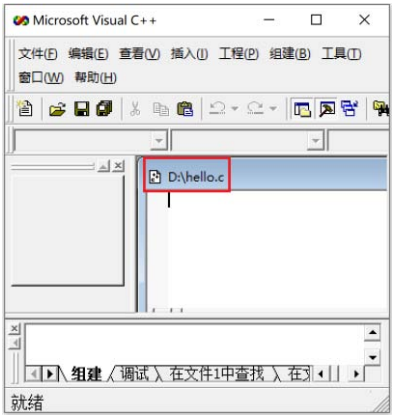


图 1-22　新创建的 C 源文件

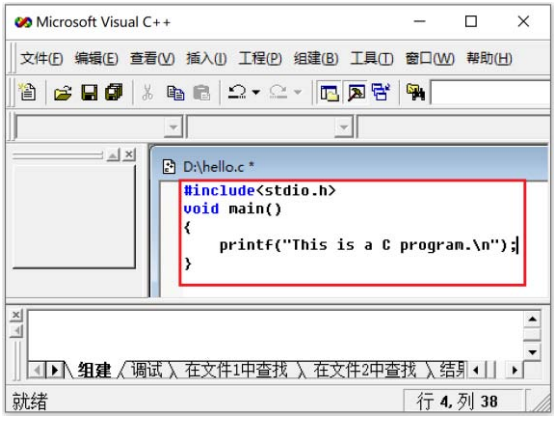


图 1-23　程序代码的输入界面

(7) 编写完程序后，选择“组建”菜单中的“编译”选项，对程序进行编译，如图 1-24 所示。

(8) 系统弹出如图 1-25 所示的提示框，询问是否创建默认的项目工作环境。

(9) 单击“是”按钮，此时会询问是否改动源文件的保存地址，如图 1-26 所示。

(10) 单击“是”按钮，开始编译程序。程序如果没有错误，即可被成功编译。单击工具栏中的按钮，使用连接程序创建.exe 文件。最后，单击工具栏中的!按钮，即可显示程序的执行结果。

图 1-24　选择“编译”选项

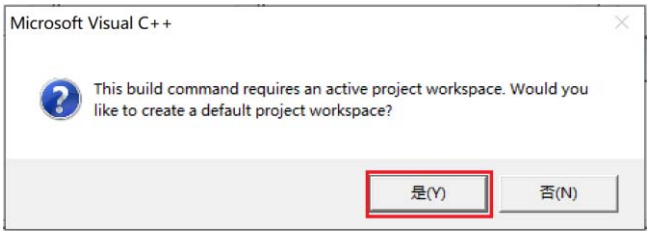


图 1-25　询问是否创建默认的项目工作环境

图 1-26　询问是否改动源文件的保存地址

1.4.3　Visual C++ 6.0 中 C 文件的编辑、编译与运行

C 语言是高级程序设计语言，使用 C 语言编写的程序通常称作 C 语言源程序(扩展名为.c)，这种程序虽然容易使用、书写和阅读，但计算机不能直接执行，因为计算机只能识别和执行二进制形式的机器语言程序。为使计算机完成某个 C 语言源程序描述的工作，就必须首先把这个 C 语言源程序转换成二进制形式的机器语言程序，这种转换被称为“C 语言源程序的加工”。“C 语言源程序的加工”包括“编译”和“连接”两个步骤，如图 1-27 所示。

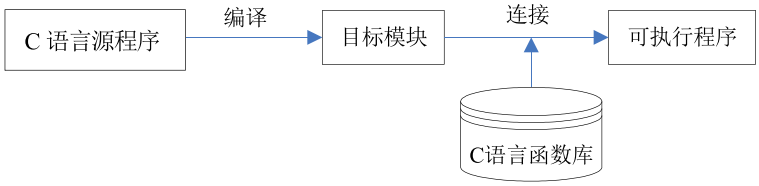


图 1-27　C 语言源程序的加工

第一步，由编译程序对源文件进行分析和处理，生成相应的机器语言目标模块，由目标模块构成的代码文件称为目标文件(扩展名为.obj)。目标文件还不能执行，因为缺少运行 C 语言程序所需的运行系统。此外，C 语言程序一般都要使用函数库提供的某些功能，例如标准函数库中的输出函数 printf。

第二步，由连接程序将编译后得到的目标模块与其他必要部分(运行系统、函数库提供的功能模块等)拼装起来，组成可执行程序(扩展名为.exe)。

1.4.4　编程中的注意事项

程序设计是一种智力劳动，编程时面对的是需要解决的问题，最终完成的是符合解题要求的程序。有了编程语言之后，该如何着手编写程序呢？在程序设计领域里，解决小问题与解决大问题，为完成练习而写程序与为解决实际应用需求而写程序之间并没有本质的区别。

使用编程语言编写程序时需要注意以下几个重要方面：

(1) 培养分析问题的能力，特别是从计算和程序的角度分析问题的能力。应学会从问题出发，通过逐步分析和分解，把原始问题转换为能用计算机通过程序方式解决的问题。在此过程中，构造出解决方案。这方面的探索没有止境，许多专业性问题都需要用计算机来解决。为此，参与者既需要熟悉计算机，又需要熟悉专业领域；未来的世界将特别需要这种复合型人才。虽然课程和教科书里的问题很简单，但它们是通向复杂问题的桥梁。

(2) 掌握所使用的编程语言，熟悉编程语言中的各种结构，包括它们的形式和意义。编程语言是解决程序问题的工具，要想写好程序，就必须熟悉编程语言。应注意，熟悉编程语言绝不是背诵定义，这种熟悉过程只有在程序设计实践中才能完成。就像上课再多也不能学会开车一样，仅靠看书、读程序、抄程序不可能真正学会写程序，必须反复亲手实践。

(3) 学会写程序。虽然写过程序的人很多，但是会写程序、能写出好程序的人不多。什么是“好程序”？例如，为解决同样的问题，使写出的程序比较简单就是衡量标准之一。这里可能有算法的选择问题，有语言的使用问题，还需要确定适用的程序结构等。除了程序本身是否正确之外，人们还特别关注写出的程序是否具有良好的结构，是否清晰，是否易于阅读和理解，以及当问题中的有些条件或要求发生改变时，它们是否容易修改以满足新的要求，等等。

(4) 检查程序错误的能力。初步编写的程序通常会包含一些错误。虽然编译系统能帮我们查出其中一些错误，并通告发现错误的位置，但确认实际错误和实际位置，以及确定应如何改正等，这些永远是编程人员自己的事。对于系统提出的各种警告，以及系统无法检查出来的错误的认定等，更需要编程人员具备一定的纠错能力，这种能力也需要在学习过程中有意识地加以锻炼。

(5) 熟悉所使用的工具和环境。程序设计需要用到一些编程工具，当在具体的计算机环境中进行编程时，熟悉工具和环境是很重要的。大部分读者可能要用某种集成开发环境进行编程，熟悉这种集成开发环境能够大大提高我们的工作效率。

1.5 本章小结

本章主要介绍了以下内容：

(1) C 语言的发展过程。

(2) C 语言的特点与标准化。

(3) 通过三个简单的 C 语言实例了解 C 语言程序的结构。

(4) Visual C++ 6.0 开发环境的搭建，C 文件的创建、编辑、编译与运行。

(5) 了解 C 语言程序的执行过程。

(6) 养成良好的编程风格，了解编程中的注意事项。

1.6 编程经验

(1) 在实际的编程过程中，如果遇到 C 语言的某些特征不清楚，可以编写小的程序运行一

下，看看得到的结果和自己了解到的是否一致，从而加深理解。

(2) 所有的不以花括号开始和结束的语句都必须以分号结束，以#开始的语句除外。

(3) 一般不要在一行中编写多行语句。

(4) 在编写程序时，花括号应该成对出现；否则，当花括号中的程序较长时，可能会忘记输入匹配的}符号。

(5) C 语言对大小写很敏感，所以在编写 C 语言程序时，大小写一定要分清楚。

(6) 在自定义函数中，一定要添加注释。

1.7 本章习题

1. 简述 C 语言的基本特点。

2. 举例说明 C 语言程序由哪几部分组成。

3. C 语言程序从开发到执行一般需要经过几个阶段？各个阶段的作用是什么？

4. 熟悉自己学习 C 语言程序设计时准备使用的编译系统或集成开发环境，了解编译系统的基本使用方法、基本操作(命令方式或包含窗口和菜单的图形界面方式)，弄清楚如何取得联机帮助信息。设法找到并翻阅编译系统的手册，了解手册的结构和各部分的基本内容。了解在编译系统中编写简单程序的基本步骤。安装并熟悉 Visual C++。

5. 输入本章正文中给出的 C 程序实例(注意程序格式)，在自己选用的系统中创建 C 源程序；对 C 源程序进行加工，得到对应的可执行程序；运行可执行程序，看看效果如何(输出了什么信息等)。

6. 下列 C 语言程序的写法是否正确？若有错误，请改正。

```
1) void main()
   {
        printf("C program1")
   }
```

```
2) void main
   {
      printf("C program1");
      printf("C program2");
   }
```

7. 在 C 语言中，main 函数的用途是什么？

8. 描述 C 语言程序从编辑到运行都经过了哪些过程？

9. 说明源代码和可执行程序之间的关系。

10. 编写一个 C 语言程序，生成以下图形。

```
    *
   ***
  *****
 *******
  *****
   ***
    *
```

11. 选择题

(1) 关于程序设计的步骤和顺序，以下说法中正确的是____。

(A) 确定算法后，整理并写出文档，最后进行编码和上机调试。

(B) 首先确定数据结构，然后确定算法，编码并上机调试，最后整理文档。

(C) 首先编码和上机调试，然后在编码过程中确定算法和数据结构，最后整理文档。

(D) 首先写好文档，然后根据文档进行编码和上机调试，最后确定算法和数据结构。

(2) 计算机能直接执行的是____。

(A) 源程序　(B) 目标程序　(C) 汇编程序　(D) 可执行程序

(3) 以下叙述中错误的是____。

(A) C 语言的可执行程序是由一系列机器指令构成的。

(B) 用 C 语言编写的源程序不能直接在计算机上运行。

(C) 通过编译得到的二进制目标程序需要连接才可以运行。

(D) 在没有安装 C 语言集成开发环境的计算机上，不能运行通过 C 源文件生成的.exe 文件。