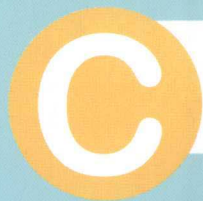


贵州大学规划教材——工科基础化学课程教学改革系列
贵州大学教材建设委员会审核批准



语言程序设计教程

The Programming Language Design

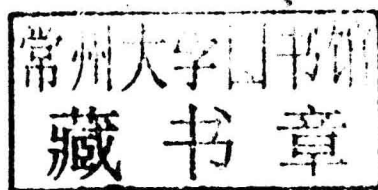
主编 刘平

C 语言程序设计教程

主 编：刘 平

编 委：李立凯 陈 琨 林 序

代昆玉 邓瑞新



贵州大学出版社

Guizhou University Press

图书在版编目(CIP)数据

C语言程序设计教程 / 刘平编写. -- 贵阳:
贵州大学出版社, 2011.8
ISBN 978-7-81126-368-8

I. ①C… II. ①刘… III. ①C语言 - 程序设计 - 教材
IV. ①TP312

中国版本图书馆CIP数据核字(2011)第167267号

C语言程序设计教程

著 者: 刘 平
责任编辑: 申 云 熊曙佳
出版发行: 贵州大学出版社
印 刷: 贵阳佳迅印务有限公司
开 本: 787mm × 1092mm 1/16
印 张: 23.5
字 数: 496千字
版 次: 2011年8月第1版 第1次印刷
书 号: ISBN 978-7-81126-368-8
定 价: 42.00元

版权所有 侵权必究
本书若出现印装质量问题, 请与出版社联系调换
电话: (0851) 5981027

目 录

第一章 C 语言概述

1.1 C 程序	(2)
1.1.1 如何学习 C 语言	(2)
1.1.2 第一个 C 语言程序	(3)
1.1.3 C 程序的构成及书写格式	(3)
1.2 Visual C + + 6.0 集成开发环境简介	(4)
1.2.1 VC 6.0 的安装	(4)
1.2.2 VC 6.0 的启动、退出	(7)
1.2.3 VC 6.0 集成开发环境介绍	(8)
1.2.4 编译、执行源程序	(9)

第二章 基本数据类型、运算符及表达式

2.1 C 语言数据类型	(12)
2.2 常量、变量和标识符	(12)
2.2.1 标识符	(13)
2.2.2 关键字	(13)
2.2.3 常量和符号常量	(15)
2.2.4 变量	(15)
2.3 整型数据	(16)
2.3.1 整型常量就是整常数	(16)
2.3.2 整型变量	(17)
2.4 实型数据	(18)
2.5 字符型数据	(19)
2.6 不同数据类型之间的转换	(19)
2.7 算术运算符和算术表达式	(21)
2.7.1 算术运算符	(21)
2.7.2 算术表达式	(22)
2.8 赋值运算符和赋值表达式	(22)

2.9 自增、自减运算符式	(24)
2.10 逗号运算符和逗号表达式	(25)

第三章 C 语言编程语句

3.1 结构化程序的三种语句	(28)
3.2 C 语句概述	(29)
3.3 顺序结构程序设计	(30)
3.3.1 赋值语句	(30)
3.3.2 变量赋初值	(30)
3.3.3 输入/输出的实现	(31)
3.3.4 顺序结构程序设计举例	(36)
3.4 选择结构程序设计	(37)
3.4.1 if 语句	(37)
3.4.2 switch 语句	(43)
3.5 循环结构程序设计	(46)
3.5.1 当型循环 while 语句	(46)
3.5.2 直到型循环 do...while 语句	(48)
3.5.3 for 型循环与 for 语句	(50)
3.5.4 几种循环结构的比较	(53)
3.5.5 循环嵌套	(54)
3.5.6 循环过程控制语句	(55)
3.6 程序设计基本结构小结	(57)

第四章 数 组

4.1 数组的基本概念	(63)
4.2 一维数组	(63)
4.2.1 一维数组的定义	(63)
4.2.2 一维数组的引用	(65)
4.2.3 一维数组的初始化	(66)
4.2.4 一维数组应用举例	(67)
4.3 二维数组	(70)
4.3.1 二维数组的定义	(70)
4.3.2 二维数组的引用	(72)
4.3.3 二维数组的初始化	(73)

4.3.4 二维数组应用举例	(75)
4.4 字符数组	(76)
4.4.1 字符数组的定义	(76)
4.4.2 字符数组的引用	(77)
4.4.3 字符数组的初始化	(78)
4.4.4 字符数组的输入输出	(79)
4.4.5 字符串处理函数	(81)
4.4.6 字符数组应用举例	(84)

第五章 函 数

5.1 函数概述	(89)
5.2 函数的定义	(91)
5.3 函数参数与函数的值	(92)
5.3.1 形式参数与实际参数	(92)
5.3.2 函数的返回值	(93)
5.4 函数的调用	(95)
5.4.1 函数的调用方式	(95)
5.4.2 函数调用的参数传递	(97)
5.5 函数的声明与函数原型	(105)
5.6 函数的递归调用	(108)

第六章 指 针

6.1 指针概述	(113)
6.1.1 指针与指针变量	(113)
6.1.2 指针变量的定义与初始化	(114)
6.1.3 指针变量的赋值和引用	(115)
6.1.4 指针变量的运算	(120)
6.1.5 指向指针的指针变量	(123)
6.2 指针与数组	(124)
6.2.1 指针与一维数组	(124)
6.2.2 指针与二维数组	(128)
6.2.3 指针数组	(132)
6.3 指针与字符串	(132)
6.3.1 字符指针变量	(132)

6.3.2 关于字符指针变量与字符数组的讨论	(136)
6.3.3 用指针数组处理多个字符串	(138)
6.4 指针与函数	(141)
6.4.1 指针变量作为函数的参数	(141)
6.4.2 返回指针值的函数	(154)
6.4.3 指向函数的指针变量	(155)
6.4.4 带参数的主函数	(160)

第七章 变量作用域及存储类别

7.1 变量作用域	(163)
7.1.1 局部变量	(163)
7.1.2 全局变量	(164)
7.2 变量的存储类型	(166)
7.2.1 局部变量的存储类型	(167)
7.2.2 全局变量的存储类型	(172)
7.2.3 关于变量的声明和定义	(175)
7.3 内部函数和外部函数	(176)
7.3.1 内部函数	(176)
7.3.2 外部函数	(176)

第八章 结构体与共用体

8.1 结构体类型	(181)
8.1.1 结构体类型的定义	(181)
8.1.2 结构类型变量的定义	(182)
8.1.3 结构体变量的引用	(185)
8.1.4 结构体变量的初始化	(185)
8.2 结构体数组及指向结构体的指针	(186)
8.2.1 结构体数组	(186)
8.2.2 指向结构体变量的指针	(189)
8.2.3 指向结构体数组的指针	(192)
8.2.4 结构体变量作函数参数	(194)
8.3 链表	(198)
8.3.1 指向自身结构的指针	(198)
8.3.2 动态存储分配函数	(198)

8.3.3 链表的概念	(200)
8.3.4 链表的基本操作	(201)
8.4 共用体类型	(209)
8.4.1 共用体类型与共用体类型变量	(209)
8.4.2 共用体变量的引用	(211)
8.4.3 共用体变量的赋值及使用	(211)
8.5 类型定义	(214)
8.5.1 枚举类型	(214)
8.5.2 类型定义符 typedef	(216)

第九章 位运算

9.1 位运算的概念	(218)
9.1.1 位的概念	(218)
9.1.2 数的补码表示	(218)
9.2 位运算符	(219)
9.2.1 取反(~)运算符	(219)
9.2.2 左移位(<<)运算符	(220)
9.2.3 右移位(>>)运算符	(220)
9.2.4 按位与(&)运算符	(220)
9.2.5 按位异或(^)运算符	(221)
9.2.6 按位或()运算符	(222)
9.2.7 位赋值运算符	(223)

第十章 文 件

10.1 C 语言文件的概念	(225)
10.1.1 文件的概念	(225)
10.1.2 文件的分类	(225)
10.1.3 缓冲文件系统	(226)
10.1.4 文件指针	(226)
10.1.5 文件打开	(227)
10.1.6 文件关闭	(229)
10.2 文件的顺序读写	(230)
10.2.1 字符方式的文件读写	(230)
10.2.2 字符串方式的文件读写	(234)

10.2.3 格式化方式的文件读写	(236)
10.2.4 数据块的读写	(237)
10.3 文件定位与随机读写	(240)
10.3.1 fseek 函数	(240)
10.3.2 ftell 函数	(241)
10.3.3 rewind 函数	(241)
10.3.4 相关函数	(242)

第十一章 编译预处理

11.1 文件包含	(246)
11.2 宏定义	(249)
11.2.1 不带参数的宏	(249)
11.2.2 带参数的宏	(252)
11.3 条件编译	(257)

二级 C 语言程序设计模拟试题

二级 C 语言程序设计模拟试题(一)	(261)
二级 C 语言程序设计模拟试题(二)	(272)
二级 C 语言程序设计模拟试题(三)	(283)
二级 C 语言程序设计模拟试题(四)	(296)
二级 C 语言程序设计模拟试题(五)	(307)
二级 C 语言程序设计模拟试题(六)	(316)
二级 C 语言程序设计模拟试题(七)	(327)
二级 C 语言程序设计模拟试题(八)	(337)

全国计算机等级考试二级公共基础知识

第一章 数据结构与算法	(347)
第二章 程序设计基础	(355)
第三章 软件工程基础	(357)
第四章 数据库设计基础	(362)

第一章 C 语言概述

C 语言是国际上广泛流行的高级语言,而且成为最受欢迎的语言之一,主要是因为它具有强大的功能。任何事物的产生都有一定的历史背景,C 语言也是这样。20 世纪 60 年代,随着计算机技术的迅速发展,高级程序设计语言得到了广泛的应用,但是仍没有一种可以用于开发操作系统等系统软件的高级语言,人们不得不用汇编语言(甚至机器语言)来开发,然而汇编语言存在着不可移植、可读性差、研制软件效率不如高级语言等缺点,给编程带来了极大的不便。为此,人们希望能够找到一种既具有一般高级语言的特点(方便易用),又具有低级语言特性(可以直接在硬件上进行操作)的语言,集它们的优点于一身。于是,C 语言就在这种历史背景下产生了。早期的 C 语言主要是用于 UNIX 系统。由于 C 语言的强大功能和各方面的优点逐渐为人们认识,到了八十年代,C 语言开始进入其它操作系统,并很快在各类大、中、小和微型计算机上得到了广泛的使用,成为当代最优秀的程序设计语言之一。

在 C 语言的基础上,1983 年又由贝尔实验室的 Bjarne Stroustrup 推出了 C++。C++ 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++ 目前流行的最新版本是 Microsoft Visual C++ 6.0。C++ 提出了一些更为深入的概念,它所支持的这些面向对象的概念容易将问题空间直接地映射到程序空间,为程序员提供了一种与传统结构程序设计不同的思维方式和编程方法,因而也增加了整个语言的复杂性,掌握起来有一定难度。但是,C 语言是 C++ 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++ 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。

C 语言出现的历史背景

1983 年,美国国家标准化协会(ANSI)根据 C 语言各种版本对 C 的发展和扩充,制定了新的标准 ANSI C,比标准 C 有了很大的发展。

1988 年,K & R 按照 ANSI C 修改了他们的《The C Programming Language》。

1987 年,ANSI 公布了新标准——87 ANSI C。

1990 年,国际标准化组织接受了 87 ANSI C 为 ISO C 的标准(ISO9899—1990)。

1994 年,ISO 又修订了 C 语言标准。

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。

在 C 语言的基础上,1983 年贝尔实验室又推出了 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 语言是 C++ 的基础,C++ 语言与 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++ 语言时,会感到更加容易和便利。

1.1 C 程序

程序是指存储在计算机内部存储器中,可以连续执行的一条条指令的集合。它是人与机器进行“对话”的语言,也就是人们常说的“程序设计语言”。

C 语言是一种“高级语言”。高级语言都是以接近人们习惯的自然语言和数学语言作为语言的表达形式,人们学习和操作起来十分方便。但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,只能接受与处理由 0 和 1 的代码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”。

人们把由高级语言编写的程序称为“源程序”,由二进制代码表示的程序称为“目标程序”,而“编译程序”是指可以把源程序转换成目标程序的软件,即可以把人们按规定语法写出的语句逐一翻译成二进制的机器指令。每一种高级语言都有它对应的编译程序。

每条 C 语句经过编译(Compile)最终都将转换成二进制的机器指令。由 C 语言构成的指令序列称为 C 语言源程序,按 C 语言的语法编写 C 程序的过程,称为 C 语言的代码编写。

C 语言源程序经过编译后生成一个后缀为 .obj 的二进制文件(目标文件),最后还要由“链接程序”(Link)将此 .obj 文件与 C 语言中提供的各种库函数链接起来生成一个后缀为 .exe 的可执行文件。只需输入此文件的名称(不必输入后缀 .exe),就可以运行。

这里有一个问题,既然有了面向对象的 C++ 语言,为什么还要学习 C 语言?

解释 1: C++ 是由于开发大型应用软件的需要而产生的,并不是所有的人都要去编写大型软件。

解释 2: 面向对象的基础是面向过程。C++ 是面向对象的语言,C 是面向过程的,学起来比 C 语言困难得多,所以不太适合程序设计的初学者。

1.1.1 如何学习 C 语言

要学习好 C 语言,首先就要了解 C 语言的一些根本特点和基本规则。

1. C 语言简洁、紧凑,使用方便、灵活。

ANSI C 一共只有 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份。另外,Turbo C 扩充了 11 个关键字:asm _cs _ds _es _ss _cdecl for hnge _interrupt _near Pascal。

表 1.1 ANSI C 的 32 个关键字

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			

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 把括号、赋值、逗号等都作为运算符处理。从而使 C 的运算类型极为丰富,可以实现其它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。

7. 生成目标代码质量高,程序执行效率高。

8. 与汇编语言相比,用 C 语言写的程序可移植性好。

但是,C 语言对程序员要求也高,程序员用 C 写程序会感到限制少、灵活性大、功能强,但较其它高级语言在学习上要困难一些。

1.1.2 第一个 C 语言程序

```
#include <stdio.h>          /* 文件包含 */
void main()                 /* 主函数 */
/* 函数体开始 */
{
    int x,y,sum;            /* 定义三个整型变量 */
    x = 5;
    y = 6;                  /* 给 x,y 变量赋值 */
    sum = x ++ y;           /* 计算结果 */
    printf("两数之和 sum = %d\n",sum);
    printf("This is a C program. \n"); /* 输出语句 */
} /* 函数体结束 */
```

说明:

①上面例中程序的功能是在屏幕上输出一个字符串“This is a C program.”。

②本程序由一个主函数组成。

③在程序的每行后用“/*”和“*/”括起来的内容为注释部分,程序不执行注释部分。

④main 为主函数名, void 表示无返回函数类型。

⑤每个 C 程序必须有一个主函数 main。

⑥{、} 是函数开始和结束的标志,不可省略。

⑦每个 C 语句以分号结束。

1.1.3 C 程序的构成及书写格式

1. C 程序构成:

(1) 一个 C 语言源程序可以由一个或多个源文件组成。

(2) 每个源文件可由一个或多个函数组成,这使得程序容易实现模块化。

(3) 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。

(4) 源程序中可以有预处理命令(include 命令仅为其中的一种),预处理命令通常应放在源文件或源程序的最前面。

(5) 每一个说明、每一个语句都必须以分号结尾。但预处理命令,函数头和花括号“{”之后不能加分号。

(6) 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。

2. 良好的 C 程序书写格式:

(1) 一个说明或一个语句占一行。

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

(3) 低一层次的语句或说明可比高一层次的语句或说明缩进若干格后书写,以便看起来更加清晰,增加程序的可读性。

(4) C 程序总是从 main 函数开始执行的,与 main 函数的位置无关。

(5) 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,C 程序没有行号。

(6) 每个语句和数据声明的最后必须有一个分号。

(7) C 语言本身没有输入输出语句。输入和输出的操作是由库函数 scanf 和 printf 等函数来完成的。C 对输入输出实行“函数化”。

在编程时应力求遵循这些规则,以养成良好的编程风格。

1.2 Visual C++ 6.0 集成开发环境简介

本书采用 Visual C++6.0 集成开发环境来调试 C 语言程序。Visual C++ 开发环境目前广泛采用的是 Microsoft Visual Studio 系列集成开发环境,Visual C++6.0 是 Microsoft 公司出品的支持 C++ 语言的集成开发环境(IDE),它集成了编辑器、编译器等开发工组,使得创建、调试 Windows 应用程序十分简单。

1.2.1 VC 6.0 的安装

目前市面上流行的 Visual C++6.0 安装程序一般包含在 Microsoft Visual Studio 6.0 中。Microsoft Visual Studio 包含了 Visual C++、Microsoft Visual J++、Visual Basic 和其它几个产品的安装文件。

Visual C++6.0 对操作系统要求很低,现有的绝大部分计算机均能满足安装要求。

这里以 Microsoft Visual Studio 6.0 Enterprise 版本为例,简要说明 Visual C++6.0 集成开

发环境的安装过程。

在安装光碟上获取安装文件后,执行“Setup.exe”,出现如图 1.1 所示的安装向导界面,根据提示,一步一步执行即可。在安装过程中,向导会提示用户输入产品的序列号、安装位置以及安装类型,一般采用系统默认设置即可。

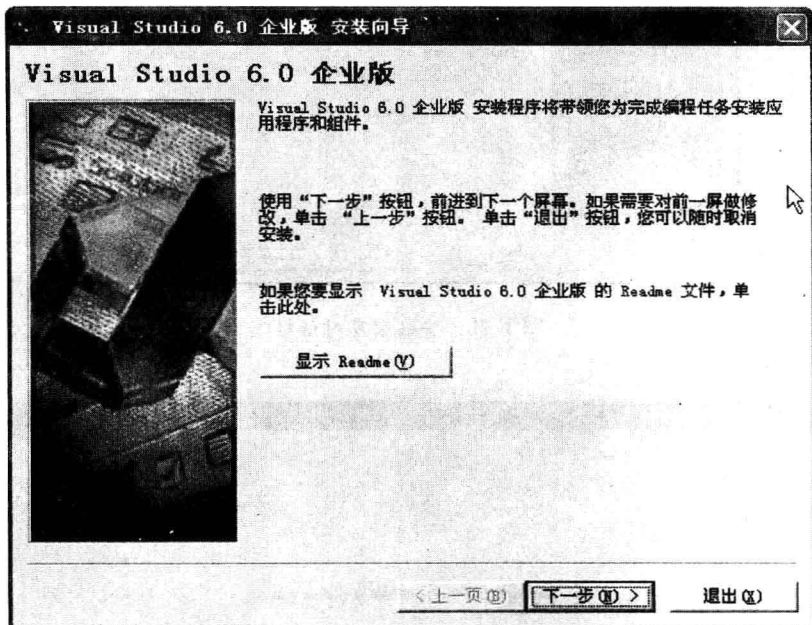


图 1.1 安装向导界面

安装过程中出现如图 1.2 所示界面,提示用户选择要安装的程序。Visual C++ 6.0 必须选中,其它可根据需要自由选择。

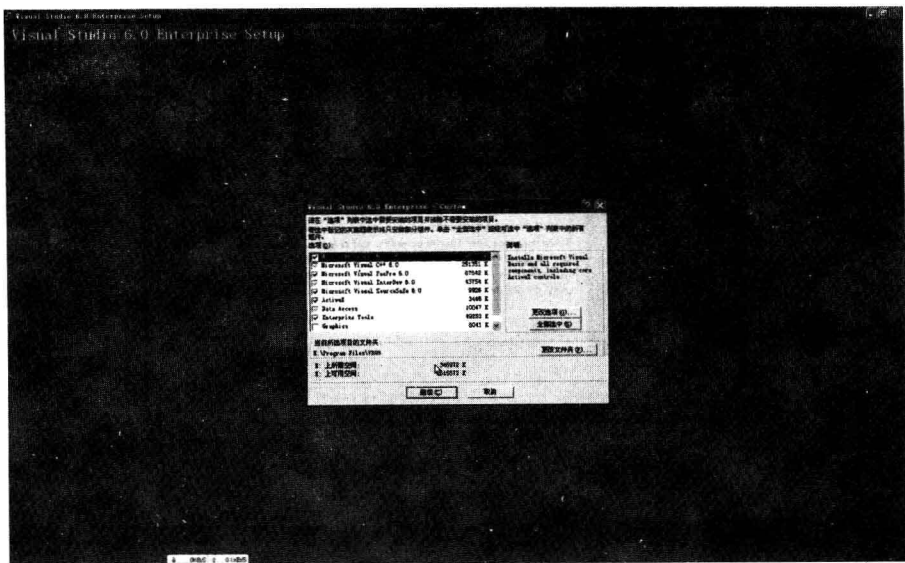


图 1.2a 安装程序对话框

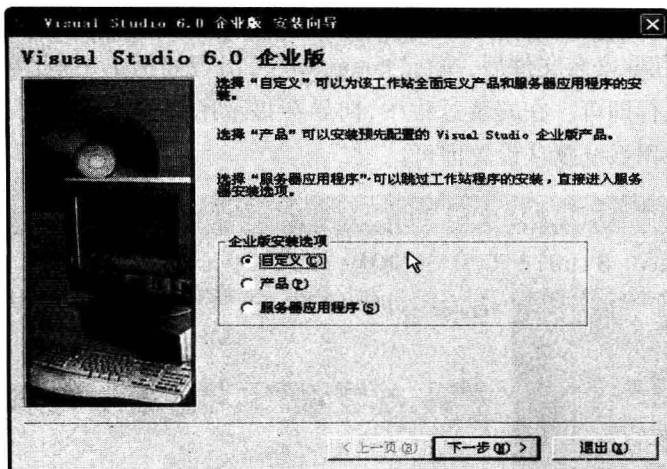


图 1.2b 安装程序对话框

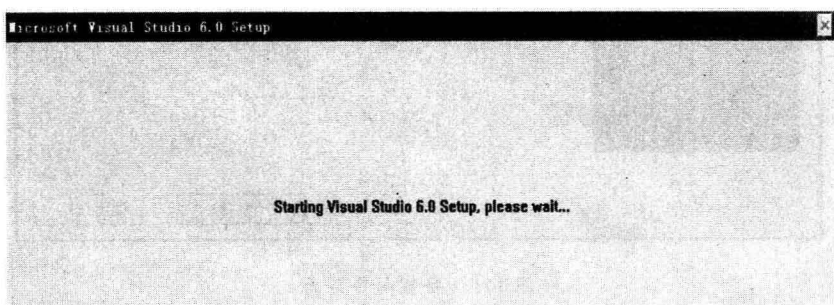


图 1.2c 安装程序对话框

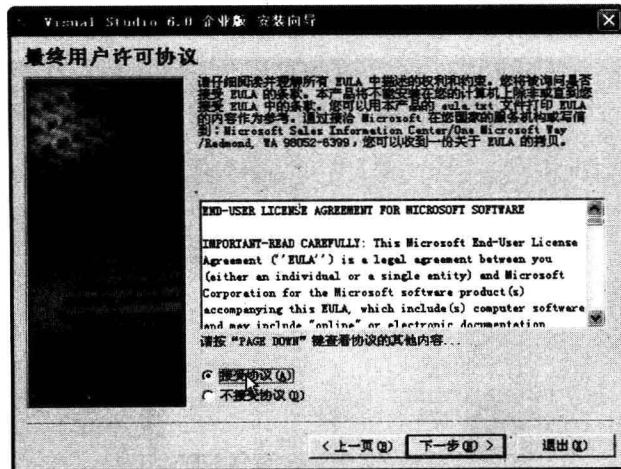


图 1.2d 安装程序对话框

安装完毕后,系统重新启动计算机,会提示是否安装 MSDN。MSDN 是对 VC 应用程序开发十分有帮助的文件,建议用户尽量安装。

1.2.2 VC 6.0 的启动、退出

在开始菜单中启动 Microsoft Visual C++ 6.0, 选中 C++ Source File 进入编辑区。如图 1.3 ~ 1.5。

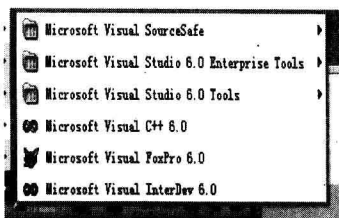


图 1.3 启动 VC

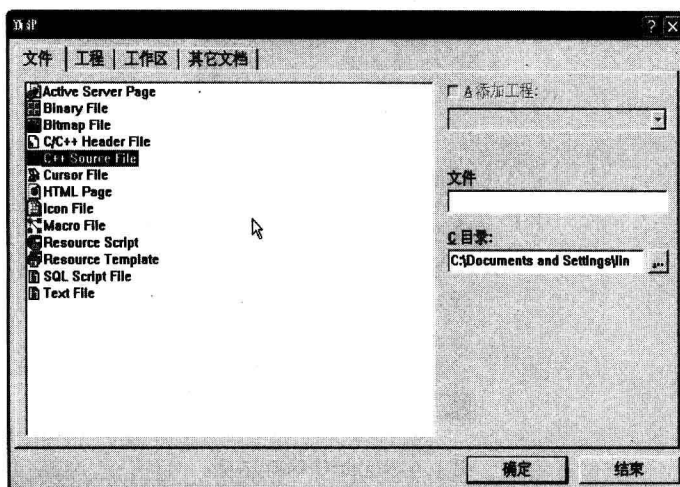


图 1.4 选中建立 C 源文件

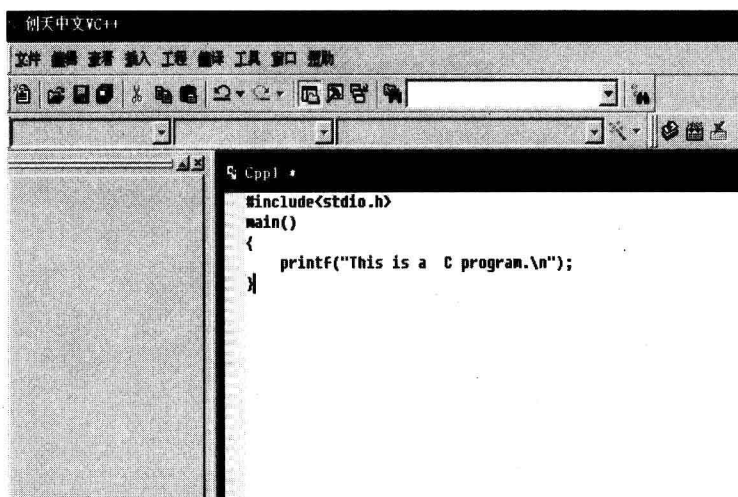


图 1.5 C 源程序编辑器

1.2.3 VC 6.0 集成开发环境介绍

Visual C++ 6.0 菜单栏由多个下拉菜单组成,每个下拉菜单又包含若个菜单项,下面着重介绍一下一些重要菜单项的使用。

1. “文件”菜单

用于打开或保存文件操作。

2. “编辑”菜单

包含一些文本编辑的基本功能及一些变量的查询功能。

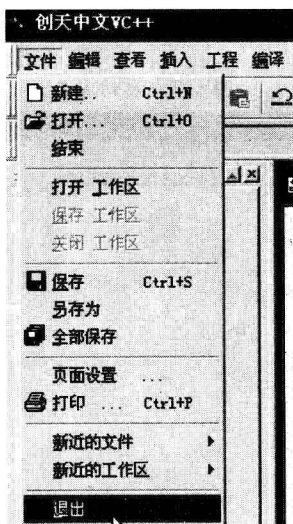


图 1.6 退出 C 编辑环境

3. “编译”菜单

用于编译、建立和执行应用程序。

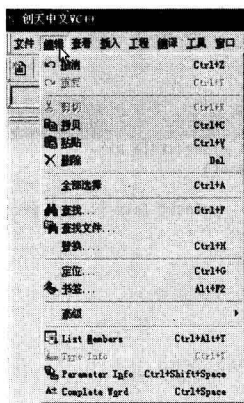


图 1.7 编辑菜单