

可下载教学资料

<http://www.tup.tsinghua.edu.cn>

21 世纪普通高校计算机公共课程规划教材

C 语言程序设计

王晓彬 主编

罗福强 侯孟书 梁良 副主编



清华大学出版社

21 世纪普通高校计算机公共课程规划教材

C 语言程序设计

王晓彬 主编

罗福强 侯孟书 梁良 副主编

清华大学出版社
北京

内 容 简 介

C 语言功能丰富,使用灵活,可移植性好,既具有高级语言特点,又具有低级语言的特点,既可以用来编写系统软件,又可用于编写应用软件,是国内外广泛使用的计算机语言。C 程序设计是计算机相关专业人员应掌握的基本功。针对 C 语言比较难学的情况,本书在内容上做了精心安排,降低难度,突出重点,用简洁的语言介绍 C 语言的基本语法,同时使用大量的实例展现 C 语言在程序设计中的应用。本书共 10 章、上机实验 6 个。

本书是普通高校各专业学生学习 C 语言的理想教材,也是想学 C 语言又苦于无从下手的爱好者的理想入门教材。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

C 语言程序设计/王晓彬主编. —北京:清华大学出版社,2009.2

(21 世纪普通高校计算机公共课程规划教材)

ISBN 978-7-302-18079-1

I. C… II. 王… 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 099512 号

责任编辑:梁 颖 李玮琪

责任校对:时翠兰

责任印制:王秀菊

出版发行:清华大学出版社

地 址:北京清华大学学研大厦 A 座

<http://www.tup.com.cn>

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:北京鑫海金澳胶印有限公司

经 销:全国新华书店

开 本:185×260 印 张:13 字 数:322 千字

版 次:2009 年 2 月第 1 版 印 次:2009 年 2 月第 1 次印刷

印 数:1~4000

定 价:20.00 元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系调换。联系电话:(010)62770177 转 3103 产品编号:029373-01

出版说明

随着我国改革开放的进一步深化,高等教育也得到了快速发展,各地高校紧密结合地方经济建设发展需要,科学运用市场调节机制,加大了使用信息科学等现代科学技术提升、改造传统学科专业的投入力度,通过教育改革合理调整和配置了教育资源,优化了传统学科专业,积极为地方经济建设输送人才,为我国经济社会的快速、健康和可持续发展以及高等教育自身的改革发展做出了巨大贡献。但是,高等教育质量还需要进一步提高以适应经济社会发展的需要,不少高校的专业设置和结构不尽合理,教师队伍整体素质亟待提高,人才培养模式、教学内容和教学方法需要进一步转变,学生的实践能力和创新精神亟待加强。

教育部一直十分重视高等教育质量工作。2007年1月,教育部下发了《关于实施高等学校本科教学质量与教学改革工程的意见》,计划实施“高等学校本科教学质量与教学改革工程(简称‘质量工程’)”,通过专业结构调整、课程教材建设、实践教学改革、教学团队建设等多项内容,进一步深化高等学校教学改革,提高人才培养的能力和水平,更好地满足经济社会发展对高素质人才的需要。在贯彻和落实教育部“质量工程”的过程中,各地高校发挥师资力量强、办学经验丰富、教学资源充裕等优势,对其特色专业及特色课程(群)加以规划、整理和总结,更新教学内容、改革课程体系,建设了一大批内容新、体系新、方法新、手段新的特色课程。在此基础上,经教育部相关教学指导委员会专家的指导和建设,清华大学出版社在多个领域精选各高校的特色课程,分别规划出版系列教材,以配合“质量工程”的实施,满足各高校教学质量和教学改革的需要。

本系列教材立足于计算机公共课程领域,以公共基础课为主、专业基础课为辅,横向满足高校多层次教学的需要。在规划过程中体现了如下一些基本原则和特点。

(1) 面向多层次、多学科专业,强调计算机在各专业中的应用。教材内容坚持基本理论适度,反映各层次对基本理论和原理的需求,同时加强实践和应用环节。

(2) 反映教学需要,促进教学发展。教材要适应多样化的教学需要,正确把握教学内容和课程体系的改革方向,在选择教材内容和编写体系时注意体现素质教育、创新能力与实践能力的培养,为学生的知识、能力、素质协调发展创造条件。

(3) 实施精品战略,突出重点,保证质量。本规划教材把重点放在公共基础课和专业基础课的教材建设上;特别注意选择并安排一部分原来基础比较好的优秀教材或讲义修订再版,逐步形成精品教材;提倡并鼓励编写体现教学质量和教学改革成果的教材。

(4) 主张一纲多本,合理配套。基础课和专业基础课教材配套,同一门课程有针对不同层次、面向不同专业的多本具有各自内容特点的教材。处理好教材统一性与多样化,基本教材与辅助教材、教学参考书,文字教材与软件教材的关系,实现教材系列资源配套。

(5) 依靠专家,择优选。在制定教材规划时要依靠各课程专家在调查研究本课程教

材建设现状的基础上提出规划选题。在落实主编人选时，要引入竞争机制，通过申报、评审确定主题。书稿完成后要认真实行审稿程序，确保出书质量。

繁荣教材出版事业，提高教材质量的关键是教师。建立一支高水平教材编写梯队才能保证教材的编写质量和建设力度，希望有志于教材建设的教师能够加入到我们的编写队伍中来。

21 世纪普通高校计算机公共课程规划教材编委会

联系人：梁颖 liangying@tup.tsinghua.edu.cn

前言

程序设计是软件工作人员的基本功。C 语言由于其功能强、使用灵活、可移植性好、目标程序质量好而受到广泛的欢迎。目前,在高等学校,各计算机类相关专业都把 C 语言程序设计开设为基础课程,而且越来越多的学校把 C 语言作为大一学生的入门必修课程。

如何做到让学生以更短的时间、以更容易的方式迈过程序设计这道“门槛”,进入程序设计殿堂,这是我们作为教师、作为编者首先要思考的问题。为了使《C 语言程序设计》一书更具有针对性(尤其是针对我国各类高职高专院校的学生),在编写该书时打破传统的以 C 语言语法为中心的编写模式,改为以应用为目标,在内容架构上面向应用、注意实用、精讲多练,引导学生通过“读一改一写”这一循序渐进的过程,最终学会程序设计。

传统的 C 语言程序设计教材以语法为中心,精讲细讲 C 语言语法,让人感觉内容枯燥、教条死板,更为严重的是,学生学完《C 语言程序设计》课后普遍不会设计程序。为此,本教材设计以学生为中心,强调换位思考,从一个刚刚勉强学完高中数理化的大专一年级学生的角度思考问题、组织内容、设计案例和实训,目的是要让学生不仅掌握 C 语言基本的语法,还要让学生养成程序思维方式,并逐步学会分析问题、解决问题,并使用 C 语言设计程序。

本书作者是长期担任 C 语言教学的教师,为了使本书更贴近教学实际,本书在设计时避免面面俱到,对内容进行了具有针对性的优化组合,突出 C 语言的基本的、核心的内容,包括 C 语言的数据类型、运算符与表达式、数据的输入输出、程序的三种结构(顺序结构、选择结构、循环结构)、数组、函数、指针、结构型与联合型、文件等模块。

为了方便学生自学和教师教学,每一章开始均列出本章的总体要求、相关知识点、学习重点和难点等内容,每一章在结尾时进行了简明扼要的小结,在每一章的后面附有习题。同时,本书还对核心内容设计了实训任务,以方便教学。

全书分为十章,第 1 章、第 6、第 7 章由罗福强编写,第 2 章由邹倩颖编写,第 3 章、第 4 章由胡德坤编写,第 5 章由杨剑编写,第 8 章、第 9 章由梁亮编写,第 10 章由何晓羲编写。全书由罗福强统稿。

本书如有不足之处,请读者指正。

编者

2008 年 12 月 1 日

目 录

第 1 章 C 语言与程序的算法	1
1.1 简单的 C 程序及特点	1
1.1.1 几个简单的 C 程序	1
1.1.2 C 程序的特点	3
1.2 C 程序的上机操作	5
1.2.1 C 程序的上机操作过程	5
1.2.2 C 程序的编译器	5
1.2.3 Visual C++ 6.0 环境下 C 程序的操作过程	6
1.3 程序的算法	9
1.3.1 算法的概念与特性	9
1.3.2 算法描述方法	10
本章小结	12
习题一	13
第 2 章 数据类型、运算符与表达式	14
2.1 标识符	14
2.2 C 语言的数据类型	15
2.3 常量	16
2.3.1 整型、实型及符号常量	16
2.3.2 字符型常量	18
2.4 变量	19
2.4.1 变量的概念	19
2.4.2 变量的定义	20
2.5 各类数值型数据之间的混合运算	20
2.6 运算符与表达式	22
2.6.1 算术运算符与算术表达式	23
2.6.2 赋值运算符与赋值表达式	26
2.6.3 逗号运算符与逗号表达式	28
本章小结	29
习题二	29

第 3 章 数据的输入输出与顺序程序设计	30
3.1 头文件与库函数	30
3.1.1 头文件	30
3.1.2 标准库函数	31
3.2 printf 与 scanf	31
3.2.1 printf 函数	31
3.2.2 scanf 函数	38
3.3 putchar 与 getchar	41
3.3.1 字符输出函数 putchar()	41
3.3.2 字符输入函数 getchar()	41
本章小结	42
习题三	42
第 4 章 选择结构程序设计	45
4.1 程序结构	46
4.1.1 程序的三种基本结构	46
4.1.2 复合语句	47
4.2 关系运算与关系表达式	47
4.3 逻辑运算与逻辑表达式	48
4.4 if 条件分支语句	49
4.4.1 if 流程	49
4.4.2 if else 流程	51
4.4.3 else if 流程	53
4.4.4 if 语句嵌套	55
4.5 switch 多路开关语句	56
本章小结	62
习题四	62
第 5 章 循环控制	64
5.1 循环概述	64
5.2 while 语句	65
5.3 do-while 语句	67
5.4 for 语句	70
5.5 循环结构的嵌套	73
5.6 break 语句和 continue 语句	76
5.6.1 break 语句	76
5.6.2 continue 语句	77
本章小结	78

习题五	78
第 6 章 数组	81
6.1 一维数组	81
6.1.1 一维数组的定义	81
6.1.2 一维数组的引用	82
6.1.3 一维数组的初始化	82
6.1.4 一维数组的应用实例	83
6.2 二维数组	86
6.2.1 二维数组的定义	86
6.2.2 二维数组的引用	88
6.2.3 二维数组的初始化	89
6.3 字符数组与字符串	90
6.3.1 字符数组	90
6.3.2 字符串	92
6.3.3 字符数组的应用实例	98
本章小结	99
习题六	99
第 7 章 函数	102
7.1 函数的定义与调用	102
7.1.1 函数的定义	102
7.1.2 函数的参数	104
7.1.3 函数的返回值	105
7.1.4 函数的调用	106
7.2 函数的嵌套调用与递归调用	109
7.2.1 函数的嵌套调用	109
7.2.2 函数的递归调用	110
7.3 数组作为函数参数	113
7.3.1 数组元素作函数的实参	113
7.3.2 数组名作函数参数	114
7.4 变量的分类	116
7.4.1 变量的作用范围	116
7.4.2 变量的存储方式	119
7.5 内部函数和外部函数	121
7.5.1 内部函数	122
7.5.2 外部函数	122
7.5.3 程序、文件、函数与变量的关系	123
本章小结	124

习题七	124
第 8 章 指针、结构型与联合型	127
8.1 指针	127
8.1.1 指针的概念	127
8.1.2 指针变量的定义	128
8.1.3 指针变量的赋值	129
8.1.4 指针变量的运算	129
8.1.5 指针和数组	133
8.2 枚举型、结构型	137
8.2.1 枚举型	137
8.2.2 结构型	140
8.3 联合型	147
8.3.1 联合型的概念	147
8.3.2 联合型的定义	148
8.3.3 联合型变量的说明	148
8.3.4 联合型变量的赋值和使用	149
8.4 定义类型——typedef	150
本章小结	152
习题八	152
第 9 章 预处理命令	155
9.1 概述	155
9.2 宏定义	155
9.2.1 无参宏定义	155
9.2.2 带参宏定义	158
9.3 条件编译	161
本章小结	163
习题九	163
第 10 章 文件	165
10.1 C 语言文件概述	165
10.1.1 普通文件与设备文件	165
10.1.2 ASCII 码文件与二进制文件	166
10.2 文件指针	166
10.3 文件的打开与关闭	167
10.3.1 文件的打开 (fopen 函数)	167
10.3.2 文件关闭函数 (fclose 函数)	169
10.4 文件的读写	169

10.4.1	字符读写函数 fgetc 和 fputc	169
10.4.2	字符串读写函数 fgets 和 fputs	174
10.4.3	数据块读写函数 fread 和 fwrite	176
10.4.4	格式化读写函数 fscanf 和 fprintf	177
10.5	文件的随机读写	178
10.5.1	文件定位 (rewind 函数和 fseek 函数)	179
10.5.2	文件的随机读写	179
10.6	文件检测函数	180
10.6.1	文件结束检测函数	181
10.6.2	读写文件出错检测函数	181
10.6.3	文件出错标志和文件结束标志置 0 函数	181
	本章小结	181
	习题十	181
	上机实验	183
	上机实验一	183
	上机实验二	184
	上机实验三	184
	上机实验四	185
	上机实验五	186
	上机实验六	187
	附录 C 语言函数库	188
	参考文献	194

第 1 章

C 语言与程序的算法

总体要求:

- 了解 C 程序的一般结构及特点。
- 了解 C 程序的编辑、调试与运行等上机操作过程。
- 理解算法的概念及其特性, 了解算法的描述方法。

相关知识点:

- 了解计算机软件、计算机语言及分类的知识。
- 熟悉 Windows 系统基础知识及操作。

学习重点:

- C 程序的结构、特点以及算法的概念。

C 语言是 20 世纪 70 年代初出现的、国际上广泛流行的计算机高级语言, 它既可以用来编写底层系统软件, 也可以用来开发应用软件。目前, C 语言在我国已经成为学习程序设计的入门语言。本章将首先向大家展示几个简单的 C 程序, 并介绍 C 程序的特点, 然后在 Visual C++ 6.0 中演示 C 程序的编辑、编译、调试和运行的操作过程, 最后介绍程序算法的概念及其描述方法。

1.1 简单的 C 程序及特点

1.1.1 几个简单的 C 程序

【实例 1-1】 显示 Hello, World!

```
#include "stdio.h"
main()
{
    printf("Hello,World!\n");
}
```

本程序的作用是输出以下一行信息:

Hello,World!

在该程序中, #include 用来引用 C 语言的库函数文件。stdio.h 是标准输入输出库函数文件。main 表示主函数的名字。每一个 C 程序都必须有一个 main() 函数。函数体由若干行语句组成, 由大括号 {} 括起来。函数体中的每一条语句以分号结尾。本例的函数体中只包含一条语句。printf() 是 C 语言中的输出函数, 将圆括号 () 内的字符串显示输出。字符串可以是英文, 也可以是中文, 其中的 “\n” 是换行符, 表示在输出 “Hello,World!” 之后自动

换行。

【实例 1-2】 计算并显示购物的应付款。

```
#include "stdio.h"
main()
{
    int a,b,money; /*定义了三个变量*/
    a=1219; b=23; /*设单价 a=1219 元/台, 数量 b=23 台*/
    money=a*b;
    printf("应付款是%d 元\n",money);
}
```

本程序的作用是计算两个整数 a 和 b 的乘积, 并显示以下一行结果信息:

应付款是28037元

在该程序的主函数 main 中, 第一行的 int 表示定义整数型变量。“/*.....*/”表示注释, 增加注释的目的是为了方便人阅读或修改程序, 程序被编译时它将被忽略, 在运行时不起作用。注释可以添加在程序中的任何位置。第三行“money=a*b”表示先将变量 a 和变量 b 的值相乘, 再将乘积保存到变量 money 中。第四行中“%d”是输入输出的“格式字符串”, 表示以十进制整数形式输出 money 的值。

【实例 1-3】 显示使用字符拼成的图案。

```
#include "stdio.h"
main()
{
    void stars(void); /*声明函数 stars*/
    stars(); /*调用函数 stars*/
    printf("* 你好, 我是小罗 *\n");
    stars(); /*调用函数 stars*/
}
void stars() /*定义函数 stars*/
{
    printf("*****\n");
}
```

本程序的作用是输出以下 3 行字符组成的图案:

```
*****
* 你好, 我是小罗 *
*****
```

该程序由两个函数组成, 一个是主函数 main(), 另一个是显示星号的函数 stars()。在该程序中, 主函数体的第一行是声明函数 stars, stars 前的 void 表示不需要 stars() 返回任何值给 main(), 圆括号中的 void 表示不需要 main() 传送任何值给 stars()。第二行和第四行表示重复调用函数 stars()。stars() 函数的作用是显示一行由 18 个星号组成的图案, 因此其函数体仅由一条 printf 语句组成。

【实例 1-4】 求两个数的较大数。

```

#include "stdio.h"
main()
{
    int a,b,c;                /*定义三个整型变量 a、b、c*/
    int max(int,int);         /*声明函数 max*/
    printf("a="); scanf("%d",&a);    /*输入变量 a 的值*/
    printf("b="); scanf("%d",&b);    /*输入变量 b 的值*/
    c=max(a,b);               /*调用函数 max 并将结果返回给变量 c*/
    printf("max=%d\n",c);     /*显示变量 c 的值*/
}

int max(int x,int y)         /*定义函数 max, x 和 y 是整数型的形式参数*/
{
    int z;
    if(x>y) z=x;              /*if_else 表示汉语逻辑: 如果……, 否则……*/
    else z=y;
    return(z);                /*返回比较结果 z*/
}

```

本程序的作用是首先输入两个任意整数，然后显示较大的那个数。本程序由两个函数组成：一个是主函数 `main()`，另一个是求较大数的函数 `max()`。在本程序中，`max()` 函数的功能是首先求出 `x` 和 `y` 中的较大者，并存放到变量 `z` 中，然后通过 `return` 语句将 `z` 的值返回给调用它的函数 `main()`。

`main()` 函数体的第一行是定义 3 个整数型变量 `a`、`b`、`c`，第二行是声明函数 `max()`，第三行和第四行提示用户运行程序时输入变量 `a` 和 `b` 的值，第五行调用函数 `max()` 并返回较大数给变量 `c`，第六行显示变量 `c` 的值。其中的 `scanf()` 是 C 语言中的输入函数，其后面的圆括号包括了两个东西：一个是 `"%d"`，表示输入给变量 `a` 或 `b` 的值是十进制整数；另一个是 `&a` 或 `&b`，其中 `"&"` 表示“取地址”，不可缺少。`scanf` 语句执行的结果是将十进制整数的数值输入到变量 `a` 或 `b` 的地址所标志的内存单元中。`main()` 函数在调用函数 `max()` 时，首先将实际参数 `a` 和 `b` 的值分别传送给 `max()` 函数中的形式参数 `x` 和 `y`，执行函数 `max()` 后，即可获得 `a` 和 `b` 的较大数。

本程序的运行效果如下：

```

a=5
b=8
max=8

```

1.1.2 C 程序的特点

通过上述例子，我们可以看到，C 程序的结构和书写形式具有以下特点：

1. 函数是 C 程序的基本构成单位

一个 C 程序由一个主函数 `main()` 和若干个其他函数组成。C 程序中的主函数有且只能有一个，其名字必须是 `main`，其他函数可以有多个，也可以没有。函数之间可以相互调用，但主函数不能被其他函数调用。被调用的函数可以是系统提供的库函数（例如 `printf()` 函数

和 `scanf()` 函数), 也可以是用户自己定义的函数。用户自己定义的函数由用户自己命名 (须符合标识符的有关规定, 详见第二章标识符的定义)。每一个函数名后紧跟一对圆括号, 不能省略, 圆括号中可以带若干个参数, 也可以没有参数, 例如在实例 1-3 中的函数 `stars()`。

2. C 程序中的函数的组成

C 程序中一个函数由两部分组成: 函数的头部和函数体。

(1) 函数的头部

函数的头部即函数的第一行, 包括函数返回值类型、函数名、函数形参名及函数形参类型的说明。一个函数的形参可以没有, 也可以有多个。当一个函数带多个形参时, 形参之间用逗号隔开。例如在实例 1-4 的 “`int max(int x,int y)`” 中 `max` 是函数名, `x` 和 `y` 是函数的形参名, `int` 表示函数返回值和形参的类型都是整型。

(2) 函数体

函数体使用一对大括号 `{}` 括起来, 通常由声明部分和执行部分组成。函数的声明部分用来定义即将使用的变量名或声明即将调用的函数, 例如在实例 1-4 中的语句 “`int a,b,c;`” 和 “`int max(int,int);`”, 前者表示定义 3 个整型变量 `a`、`b`、`c`, 后者表示声明函数 `max`。函数体的执行部分由若干个语句组成。一个函数体可以没有声明部分, 也可以没有执行部分, 甚至不包括任何语句, 例如:

```
empty()  
{  
}
```

它是一个空函数。

3. C 程序的执行

一个 C 程序总是从主函数 `main()` 开始执行, 而不论它处在整个程序的什么位置。因此, `main` 函数可以放在所有其他函数之前, 也可以放在所有其他函数之后。

4. C 程序的语句

C 程序中的每个语句必须以分号结尾。在书写时, 源程序的一行可以书写几条语句, 一条语句也可以分写在多行上。

5. C 程序的输入与输出操作

C 语言本身没有输入输出语句。输入输出操作是通过调用头文件 `stdio.h` 中的库函数 `scanf()` 和 `printf()` 来完成的, 因此在程序的最开始需要使用 `include` 来引用头文件 `stdio.h`。函数 `scanf()` 表示将数据输入到变量名所标识的内存单元, 例如在实例 1-4 中的 `scanf("%d",&a)`。函数 `printf()` 用来输出字符串数据, 如果需要输出变量的值, 则只需在字符串中用诸如 `%d` 的格式字符代替变量名, 并在右括号之前列出变量名即可, 例如在实例 1-2 中的 `printf("应付款是%d 元\n",money)`。

6. C 程序的注释

在 C 程序中, 用户可以使用 `/*.....*/` 添加注释信息, 增加注释的目的是为了方便人阅读或修改程序, 程序被编译时它将被忽略, 在运行时不起作用。注释可以添加在程序中的任何位置。经验表明适当地添加注释, 对程序的重要部分进行说明, 可大大增强程序的可读性。

1.2 C 程序的上机操作

从上一节所展示的例子可以看出，C 程序实际上是根据问题的要求并遵照 C 语言的某些规定书写出来的东西。作为程序设计的初学者，要知道良好的程序设计习惯是先在纸上写好程序再上机运行，切忌盲目上机。

1.2.1 C 程序的上机操作过程

在编好一个 C 程序后，该如何上机运行呢？在纸上写好一个程序后，要经过“上机输入并编辑源程序→对源程序进行编译→与库函数进行连接→运行可执行的目标程序”这样几个步骤，操作过程如图 1-1 所示。其中，实线表示操作流程，虚线表示文件的输入和输出。`.cpp`、`.obj`、`.exe` 分别表示完成编辑、编译、连接后所产生的文件的扩展名。

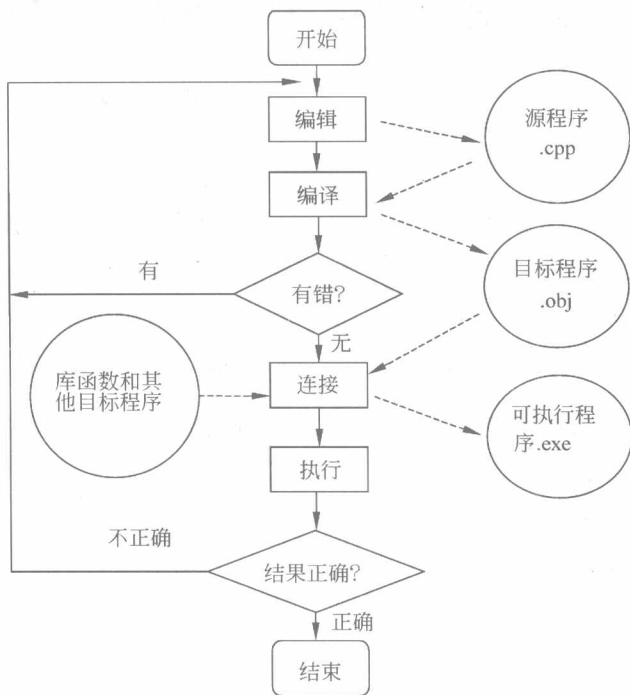


图 1-1 C 程序的上机操作过程

1.2.2 C 程序的编译器

C 语言源程序经过编辑保存后，将在磁盘上生成一个源程序文件，不同的编辑软件所生成的文件的扩展名往往是不相同的，但无论其扩展名是什么，源程序文件仍然是一个文本文件。正因为如此，我们可以使用任何一种文字处理软件完成源程序的编辑，但要注意保存为文本文件同时修改其扩展名，以符合你所使用的 C 语言编译器的要求。

目前，C 语言的编译器是比较多的，较常用的有在 DOS 平台上使用的 Turbo C、在

Windows 平台上使用的 Borland C++ 和 Microsoft Visual C++ 以及在 Linux 平台上使用的 GCC 等。本书以 Microsoft Visual C++ 6.0 为蓝本阐述 C 语言在程序设计上的应用。

1.2.3 Visual C++ 6.0 环境下 C 程序的操作过程

1. 编辑 C 语言源程序

Microsoft Visual C++ 6.0 是一个功能非常强大的集编辑、编译、连接、调试和运行程序于一体的软件开发平台。若要编辑 C 语言源程序，则按如下操作步骤进行：

(1) 启动计算机，进入 Windows XP。

(2) 选择“开始→所有程序→Microsoft Visual Studio 6.0→Microsoft Visual C++ 6.0”菜单命令，以启动 Microsoft Visual C++ 6.0。Microsoft Visual C++ 启动后的操作界面如图 1-2 所示。

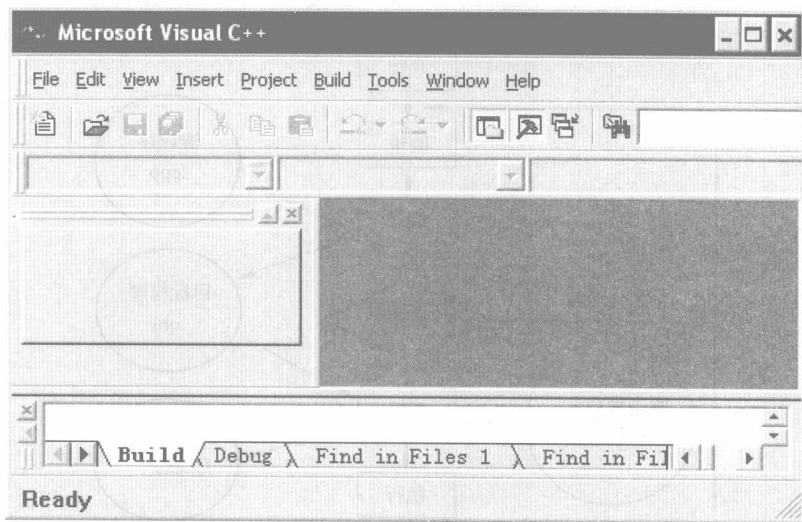


图 1-2 Microsoft Visual C++ 的主窗口

(3) 在 Microsoft Visual C++ 主窗口中选择“File→New”菜单命令，以弹出“New”对话框。

(4) 在“New”对话框选择“Files”选项卡，在列表框选择“C++ Source File”，然后在“File”文本框中输入想要创建的源程序文件名（如 myfile），并在“Location”文本框中输入存盘路径（如 d:\test），也可以通过单击右边的按钮 ，弹出“Choose Directory”对话框后选择磁盘路径，如图 1-3 所示。之后，单击“OK”按钮，返回 Microsoft Visual C++ 主窗口。

(5) 在新出现的源程序文档窗口（如“d:\test\myfile.cpp”）中输入源程序代码。源程序文档的编辑操作与 Word 操作相同，如图 1-4 所示。

(6) 源程序文档编辑操作结束后，如果需要保存，选择“File→Save”菜单命令即可。注意，在 Microsoft Visual C++ 中 C 语言源程序文件扩展名为.cpp。