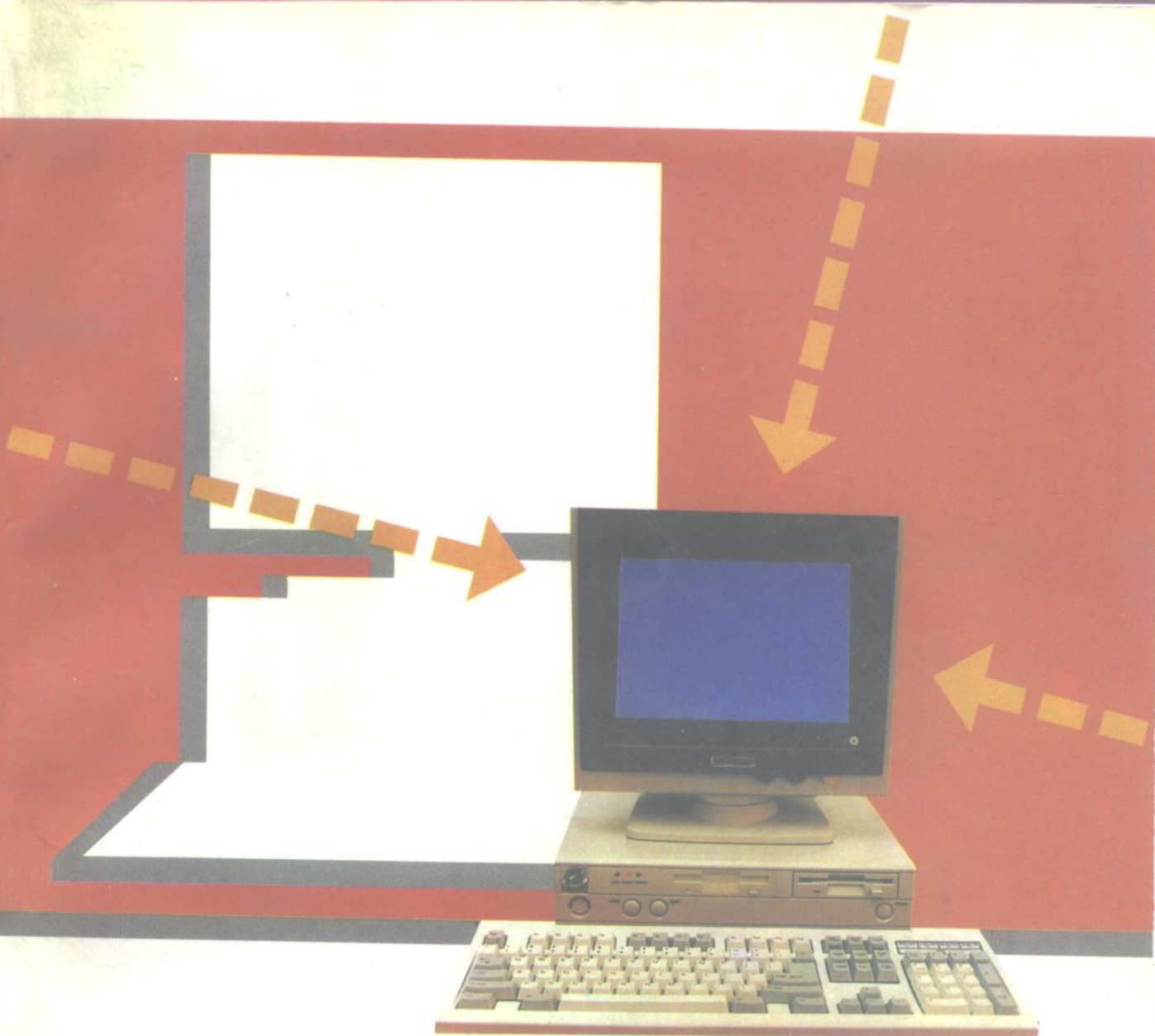


Microsoft Quick C 程序设计



杨振生 编著

# Microsoft Quick C 程序设计

中国科学技术大学出版社

中国科学

12

S11/2

版社

**图书在版编目(CIP)数据**

Microsoft Quick C 程序设计/杨振生 编著

—合肥:中国科学技术大学出版社,1995年6月

ISBN 7-312-00681-7

- I Microsoft Quick C
- II 杨振生
- III ①程序设计 ②C语言 ③Quick C
- IV TP

中国科学技术大学出版社出版发行

(安徽省合肥市金寨路96号,邮编:230026)

安徽省金寨县印刷厂印刷

全国新华书店经销

开本:787×1092/16 印张:20.5 字数:500千

1995年8月第1版 1995年8月第1次印刷

印数:1—10000册

ISBN 7-312-00681-7/TP·97 定价:18.00元

## 内 容 简 介

本书以 Microsoft Quick C 2.50 版本为主线,全面系统地介绍了 Quick C 语言的基本概念、语法规则和程序设计的方法与技巧。

本书内容丰富、逻辑性强、深入浅出、循序渐进,既注意知识结构的系统性,又强调其实用性。书中含有大量例题。这些例题取材新颖、富于启发,全部在 80386 微机上调试通过。各章最后均附有习题,供读者练习。

本书可作为高等院校 C 语言程序设计课程的教材,也可作为较高层次的计算机培训班的教材,还可供科技人员参考。

J5578/02

# 前 言

众所周知,C语言是一种结构化、模块化和可编译的通用程序设计语言。C语言兼有高级语言和低级语言的许多特性和优点,它既有高级程序设计语言的良好结构和优美的编程风格,又有类似于汇编语言那样贴近计算机硬件的能力。C语言的数据类型和运算符相当丰富,具有表达能力强、目标代码紧凑、程序运行效率高和可移植性好等特点。它广泛应用于系统软件开发、计算机辅助设计、过程自动控制、软件工程、科学计算、图象处理、数据分析、文字处理、企业管理和人工智能等诸多领域,目前已成为国际上应用非常广泛的新型的现代主流程序设计语言。

Quick C语言是C语言家族中的后起之秀,它集中了众多种C语言的优点,在目前种类林立的C语言版本中脱颖而出,独树一帜,以它特有的魅力和强大的生命力正风靡于世。

本书以最新的Microsoft Quick C 2.50版本为主线,较全面系统地介绍了Quick C语言的基本概念、语法规则和程序设计的基本方法与技巧。

本书系统性强,重点突出。全书共分11章,第一章概括介绍了Quick C语言的特点和编程风格;第二章介绍了Quick C系统的安装和使用方法;第三章介绍了Quick C语言的基本知识,其中包括基本的数据类型和变量的存储类型及运算符与结合原则;第四章介绍了Quick C的流程控制;第五章介绍了数组;第六章介绍了指针;第七章介绍了函数;第八章介绍了结构、联合和枚举等复杂的构造数据类型及处理方法;第九章介绍了文件概念及其标准函数;第十章介绍了内存动态分配与系统资源调用;第十一章介绍了图形开发基础。其中第四章至第九章是本书的重点。

本书可作为高等院校C语言程序设计课程的教材,也可作为较高层次的计算机培训班的教材,还可供广大科技人员阅读和参考。

周以海老师精心地绘制了本书的全部插图,在此表示感谢。

由于作者水平有限,书中错误在所难免,不妥之处,敬请读者指正。

作 者

1994年10月于合肥

# 目 次

前 言 .....	I
<b>第一章 Quick C 语言概述</b> .....	1
1.1 C 语言的发展和 Quick C 的产生 .....	1
1.2 Quick C 语言的新特点 .....	2
1.3 Quick C 程序的风格 .....	3
1.4 用计算机解决实际问题的步骤 .....	4
1.5 Quick C 程序的开发过程 .....	6
习题 .....	7
<b>第二章 Quick C 系统的安装与使用</b> .....	8
2.1 Quick C 的要求与环境 .....	8
2.2 Quick C 系统的安装 .....	9
2.3 Quick C 环境的进入与退出 .....	12
2.4 窗口和菜单的使用 .....	13
2.5 编辑器的使用 .....	18
2.6 编译、连接与执行 .....	20
习题 .....	25
<b>第三章 Quick C 语言的基本数据类型与运算符</b> .....	26
3.1 常量 .....	26
3.2 变量及其基本数据类型 .....	29
3.3 变量的存储类型及其作用域与生存期 .....	31
3.4 变量的初始化 .....	35
3.5 运算符及其优先级 .....	37
3.6 数据类型的转换 .....	45
习题 .....	48
<b>第四章 流程控制</b> .....	51
4.1 复合语句 .....	51
4.2 if 条件分支 .....	52
4.3 Switch~case 开关分支 .....	57
4.4 while 型循环 .....	60
4.5 for 型循环 .....	62
4.6 do~while 型循环 .....	67
4.7 continue 语句 .....	69
4.8 break 语句 .....	71
4.9 goto 语句与标号 .....	72
习题 .....	73

■

<b>第五章 数组</b> .....	75
5.1 数组及其说明 .....	75
5.2 字符数组 .....	77
5.3 多维数组 .....	79
5.4 数组的初始化 .....	83
习题 .....	86
<b>第六章 指针</b> .....	88
6.1 指针变量与地址 .....	88
6.2 指针变量的说明与初始化 .....	90
6.3 指针运算 .....	91
6.4 数组指针 .....	95
6.5 字符数组指针 .....	97
6.6 指针数组 .....	99
6.7 多级指针 .....	105
6.8 命令行参数 .....	110
习题 .....	111
<b>第七章 函数</b> .....	114
7.1 程序的结构化与模块化 .....	114
7.2 函数的定义、调用与返回 .....	117
7.3 函数间的参数传递 .....	121
7.4 数组在函数间的传递 .....	128
7.5 字符串在函数间的传递 .....	132
7.6 指针型函数 .....	134
7.7 函数的嵌套调用与递归调用 .....	136
7.8 函数指针 .....	141
7.9 编译预处理 .....	147
习题 .....	152
<b>第八章 结构、联合与枚举</b> .....	154
8.1 结构的定义与说明 .....	154
8.2 结构的成员引用 .....	158
8.3 结构的初始化 .....	160
8.4 结构数组 .....	161
8.5 结构指针 .....	164
8.6 向函数传递结构 .....	167
8.7 结构型函数和结构指针型函数 .....	172
8.8 结构嵌套 .....	176
8.9 位字段结构 .....	180
8.10 联合 .....	184
8.11 枚举 .....	187
8.12 类型定义 .....	190

习题	192
<b>第九章 文件及其标准函数</b>	196
9.1 文件概念和文件型指针	196
9.2 标准设备文件输入输出函数	198
9.3 标准设备文件格式化输入输出函数	200
9.4 文件的打开与关闭	204
9.5 文件的字符输入输出函数	205
9.6 文件的字符串输入输出函数	209
9.7 文件的整数输入输出函数	213
9.8 文件的格式化输入输出函数	215
9.9 文件的数据块输入输出函数	218
9.10 低级输入输出函数	221
习题	226
<b>第十章 内存动态分配与系统功能调用</b>	227
10.1 内存动态分配的必要性	227
10.2 内存动态分配与释放	228
10.3 链表	231
10.4 内存动态分配的扩张与收缩	245
10.5 系统资源调用	247
习题	255
<b>第十一章 图形开发基础</b>	257
11.1 图形方式的设置	257
11.2 文本坐标与图形坐标	265
11.3 生成图形	270
11.4 生成图形示例程序	274
习题	284
<b>附录</b>	285
附录 A ASCII 码表	285
附录 B Quick C 的常用标准库函数	286
附录 C Quick C 包含文件	293
附录 D 错误信息	294
<b>参考文献</b>	317

# 第一章 Quick C 语言概述

目前 C 语言以它精炼、灵活、高效、功能强大和易于移植等众多优点盛行于计算机界。C 语言已被广泛应用于系统程序设计、计算机辅助设计、控制工程、软件工程、科学计算、数据分析、图象处理、文字编辑和人工智能等诸多领域。Quick C 语言因集中了众多 C 语言版本的优点而深受计算机软件设计者和广大科技人员的欢迎。本章只对 Quick C 语言作一概括性介绍,使读者对 Quick C 语言有一个初步认识。

## 1.1 C 语言的发展和 Quick C 的产生

自 1946 年世界上第一台电子计算机诞生以来,无论硬件和软件都得到了迅猛发展,其功能不断增强和扩充,各种程序设计语言和系统软件层出不穷,并在各个领域得到了广泛应用。其中,操作系统的研制开发与不断充实和完善发挥了十分重要的作用。

早期的操作系统,主要是用汇编语言编写的,它直接依赖于计算机硬件,程序的可读性和可移植性都很差。用高级语言编写,又难以实现直接对硬件进行操作的某些功能。为此,促使人们开始寻求一种既具有一般高级语言特性,又具有低级语言特性的中级语言,这就是 C 语言产生的历史背景。

从 C 语言的发展历史来看,C 属于 ALGOL 语言族系。1960 年出现了一种高级语言 ALGOL60,它离硬件较远,不宜编写系统程序。1963 年英国的伦敦大学和剑桥大学在 ALGOL60 的基础上推出了 CPL 语言,它接近硬件一些,但规格较大,难以实现。1967 年英国剑桥大学的 Martin Richards 对 CPL 语言作了简化,推出了 BCPL 语言,但数据类型单一。1970 年美国贝尔实验室的 Ken Thompson 以 BCPL 语言为基础,设计出很接近硬件的 B 语言,并在 PDP-7 机上实现了用 B 语言开发的第一个 UNIX 操作系统,但 B 语言仍是单一的数据类型,过于简单,功能有限。1972 年美国贝尔实验室的 D. M. Ritchie 在 PDP-11 机上实现了 C 语言。它以 B 语言为基础,引入多种基本数据类型:字符、整数和浮点数,并导出了数组、指针、结构、联合、枚举和函数。既保持了 BCPL 和 B 语言精炼、接近硬件的优点,又丰富了数据类型。1973 年 K. Thompson 和 D. M. Ritchie 用 C 语言重写了 UNIX 操作系统,实现了 UNIX 第五版,开创了 UNIX 系统的新局面。

从 C 语言的发展过程中可以看出,C 语言与 UNIX 操作系统是相辅相成发展起来的,它们是一对孪生的兄弟。1978 年以后,C 语言逐渐独立于 UNIC 操作系统,B. W. Kernighan 和 D. M. Ritchie(称 K&R)合著了《The C Programming Language》,由此产生了 C 语言版本的基础,称为标准 C。1983 年,美国国家标准化协会将标准 C 作了扩充和发展,制定了新的标准,称为 ANSI C。

随着 C 语言的迅速发展,它已成为微、小、中、大和巨型等各类计算机共同使用的现代主流程序设计语言。多种不同版本的 C 语言编译系统不断出现。但目前最为流行的有三种 C 语



言编译器：

- Lattice 公司的 LC 语言；
- Microsoft 公司的 MSC 语言；
- Turbo 公司的 Turbo C 语言。

Quick C 语言综合了目前市场上最为流行的 LC、MSC 和 Turbo C 等诸种 C 语言编译器的优点，于 1987 年由美国 Microsoft 公司推出了 Quick C 1.0 版本，随后又相继推出了 Quick C 2.0 版本，最近，又推出了最新版本 Quick C 2.50。Quick C 语言具有库程序丰富、图形功能强、调试功能完整、与其它语言兼容性好、编译速度快和方法功能与环境密切结合等特点。因此，它在目前种类林立的 C 语言编译器中脱颖而出，受到广大用户的青睐。

## 1.2 Quick C 语言的新特点

在介绍 Quick C 语言的新特点之前，我们首先介绍一般 C 语言的特点。

一般 C 语言具有以下特点：

1. C 语言是介于高级语言和低级语言之间的一种中级语言。它既有像汇编语言那样可以直接访问硬件的功能，又具有高级语言面向用户、简单易学、便于阅读和理解、易于维护和适应性强等优点。

2. C 语言是一种结构化模块化程序设计语言。C 语言程序遵循结构化程序设计的原则，其逻辑结构是由顺序、分支和循环三种基本结构组成。非常适合采用自顶而下、逐步求精的结构化程序设计方法。

C 语言便于实现模块化程序设计，C 语言程序通常是函数的集合体。每个函数都具有各自独立的功能和鲜明的界面，从而使程序总体具有清晰的模块结构，并且提供了模块间的相互调用和数据传递的强有力的机制。由于这一特点，十分有利于复杂问题分散化、大型软件模块化的实现。

3. C 语言表达简洁灵活、代码紧凑质量高，其程序的目标代码执行速度快。

4. C 语言具有丰富的运算符和数据类型，表达能力强，能实现对各种复杂数据结构的运算和处理。C 语言除了有一般高级语言所具备的算术运算、关系运算和逻辑运算之外，它还能直接访问物理地址和进行位操作。它可以直接对硬件进行操作，能实现汇编语言的大部分功能。

5. C 语言具有很好的可移植性。C 语言的编译程序便于从一种机器上移植到另一种机器上，这就使得在一个环境中用 C 语言编写的诸多程序，可以不加修改或稍加修改就可以搬到另一个完全不同的环境中运行。

Quick C 语言除了具有一般 C 语言的上述特点之外，还具有以下优点：

1. Quick C 2.50 版本在语言级与 Microsoft C 6.0 版本兼容。

2. 增强了与 ANSI C 语言的兼容性。

3. Quick C 支持六种标准内存模式：

(1)微型模式；(2)小型模式；(3)中型模式；(4)紧缩型模式；(5)大型模式；(6)巨型模式。

4. 图形功能强，它有丰富的图形库函数，利用这些函数可以把数据显示成饼图、条形图、立图、线图和离散图。通过生成图形函数，可以把难以理解的栏目数字演变成单个图形。此外，

Quick C 还支持自定义窗口色彩。

5. Quick C 简化了与操作系统的接口,与低级系统调用打交道更为方便,形式更加直观。

6. Quick C 采用了集成式程序开发环境,集源程序编辑、编译、连接和调试于一体,构成一个完整的 C 语言开发环境。另外,Quick C 还增加了新型的快速监测调试特性。

7. Quick C 利用了内含程序库的技术,使编译效率达到每分钟超过一万行的高速度。

8. Quick C 在编译正确后能自动进行连接和执行,只要按一个键就能使程序自动地通过编译、连接和执行的全过程,真可谓一气呵成!

9. Quick C 库程序丰富,基本库程序常驻内存,支持快速连接。其它库程序和用户自定义库程序可以通过一个实用程序构成常驻内存库,以提高编译速度。

此外,Quick C 支持多模块编译,从而有利于大型程序的连接。

10. Quick C 具有内含的调试功能,这是最令人推崇的特性。当编译、连接完成后,Quick C 自动处于源程序调试状态,这时,你可以控制程序逐行执行,或执行到指定行,也可以检查变量的值,非常方便灵活。

### 1.3 Quick C 程序的风格

作为一个程序员,特别是初学者,可能想立即投身于 Quick C,但在此之前,你首先应该了解 Quick C 程序的基本模型,它的表现形式、构成规则和组成它的基本成份。为此,我们先剖析一个 Quick C 程序。

下面我们给出一个 Quick C 的简单程序,其文件名为 KS.C。

该程序的功能是,给定圆锥的高和底圆半径,计算圆锥的侧面积。

程序如下:

	<b>注 释</b>	<code>/* Calculating side area of cone */</code>
<b>预处理命令</b>	装入 stdio.h 文件	<code>#include &lt;stdio.h&gt;</code>
	装入 math.h 文件	<code>#include &lt;math.h&gt;</code>
	定义符号常量 PI	<code>#define PI 3.14159</code>
	函数原型声明	<code>double sarea(float r0,float h0);</code>
<b>主函数定义</b>	局部变量声明	<code>main()</code>
	输出提示信息	<code>{</code>
	输入初始数据	<code>float r,h;</code>
	函数调用	<code>double s;</code>
	输出计算结果	<code>printf("Enter r,h:\n");</code>
		<code>scanf("%f%f",&amp;r,&amp;h);</code>
		<code>s=sarea(r,h);</code>
		<code>printf("s=%lf\n",s);</code>
		<code>}</code>
<b>sarea 函数定义</b>		 
		<code>double sarea(float r0,float h0)</code>
		<code>{</code>

局部变量声明	double s0,t;
	t=sqrt(r0 * r0+h0 * h0);
	s0=PI * r0 * t;
返回值	return s0;
	}

对此程序进行以下剖析:

第一行以“/\*”开头,以“\*/”结尾,称为注释行,是非执行语句,只起注释说明作用。

第二至第四行为预处理命令,它通常包括程序运行时所需要的包含文件(如<stdio. h>和<math. h>)和符号常量(如PI)定义。

第五行是函数 sarea 的原型声明,以便 Quick C 编译器进行类型检查。

第六行 main()是主函数头,表示主函数从此开始。

第七行的花括号“{”和第十四行的“}”围住的部分是主函数的函数体。其中,第八和第九行是局部变量说明;第十行通过标准库函数 printf()输出提示信息;第十一行通过标准库函数 scanf()输入初始数据;第十二行调用函数 sarea();第十三行通过标准库函数输出计算结果。

第十五行是自定义函数 sarea()的函数头,其后的圆括号中给出了形式参数及其数据类型。

第十六行开头的花括号“{”到第二十一行的“}”围住的是自定义函数的函数体。其中,第十七行是局部变量说明;第十八和第十九行是进行具体计算;第二十行把计算结果返回到主函数的调用处。

由此可知,Quick C 程序具有以下风格:

1. Quick C 程序习惯上使用小写英文字母书写,而符号常量使用大写英文字母。
2. Quick C 程序通常是一些函数的集合体,其中有一个函数是主函数,其它函数供主函数调用,或它们之间的相互调用。程序的执行从主函数开始,主函数执行完,则整个程序执行结束。
3. Quick C 程序由一个个语句组成,每个语句都具有规定的语法格式和特定的功能。每个语句以分号结尾。
4. 一般情况下,每个语句占用一行,但也可以自由地使用任意的书写行,即一行可以写多个语句,一个语句也可以占用多行。
5. Quick C 程序中用花括号对{ },表示程序结构的层次范围,不同结构层次的语句,从不同的位置开始。这样可使结构层次清晰、便于阅读和理解。
6. 为了增加程序的可读性,可以适当加入注释行。

## 1.4 用计算机解决实际问题的步骤

用计算机实际问题大致经历以下几个步骤:

### 一、分析问题,明确要求

分析问题就是深入实际,对所要解决的问题进行全面了解和分析,在深入分析的基础上,

抓住主要矛盾,剔除次要矛盾,抽取问题的本质。

明确要求就是明确用户的要求,依据给出的条件和数据,需要进行哪些处理,输出什么样的结果,进行可行性论证和可靠性分析。

## 二、建立数学模型

在分析问题和明确要求的基础上,建立数学模型,将一个物理过程或工作状态用数学形式表达出来。

## 三、确定算法和处理方案

数学模型建立后,必须研究和确定算法。所谓算法是指解决某类问题的计算方法,不同类型的问题要用不同的计算方法。根据问题的特点,对计算方法进行优化。若没有现成方法可用,则要通过实践摸索,总结出算法思想和规律性。

## 四、画流程图

流程图是程序设计的有力工具,它以框图的方式把解决问题的先后次序和程序的逻辑结果直观地、形象地描述出来,使解题的思路清晰,有利于理解、阅读和编制调试程序,也有利于修改和排错等。

## 五、编制程序

在编制程序时,应当先分配好存储空间和工作单元及 CPU 内部的寄存器,采用结构化模块化的程序设计方法,根据流程图和确定的算法逐块逐条语句具体地编写程序。

## 六、上机调试

程序编好后,必须上机调试,特别是对于复杂的问题,往往要分解成若干个子问题,分别由几个人编写,而形成若干个程序模块,把它们组装在一起,才能形成总体程序。一般来说,总会有这样或那样的问题和错误,这些问题和错误在调试程序时通常都可以发现,然后进行修改,再调试,再修改,直到所有的问题解决为止。

## 七、试运行和分析结果

试运行和分析结果是检验设计是否达到了要求,是否满足用户提出的需求,所确定的设计是否可行。若没有达到设计要求,不满足用户的需求,就必须从分析问题开始检查修正原有的设计方案,直到符合设计要求和满足用户需求为止。

## 八、整理资料,投入运行

在试运行满足要求之后,应当系统地整理资料,其中包括现场勘察报告、可行性论证和分析报告、算法整理、源程序、流程图、调试记录或报告、使用说明书等。有关资料要及时提交给用户,以便正常投入运行。

综上所述,用计算机解决实际问题的基本步骤可用图 1.1 来描述。

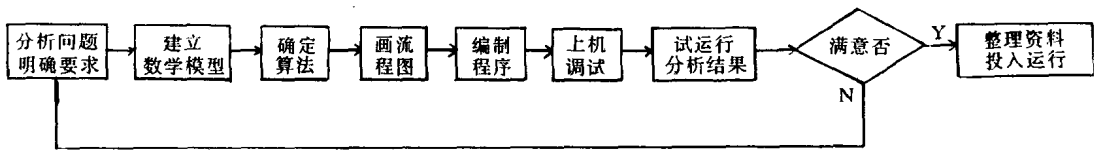


图 1.1 用计算机解决实际问题的步骤

## 1.5 Quick C 程序的开发过程

Quick C 程序的开发过程大致要经过编辑、编译、连接和执行四个步骤：

### 一、编辑

编辑的任务是利用系统提供的编辑程序来编辑 Quick C 语言源程序，并以磁盘文件的形式存入磁盘。源文件的名字由用户选定，但扩展名一定是“.C”。

用于建立 Quick C 语言源程序的编辑程序是由 Quick C 系统的编辑器提供的。当然也可以使用其它的编辑软件，如 PC-DOS 的行编辑程序 EDLIN 和字处理程序 Wordstar 等。

### 二、编译

编译的任务是调用 Quick C 的编译程序对源程序进行编译，生成目标程序文件。

Quick C 编译器在对 Quick C 程序编译的过程中，如发现源程序在语法、语义上有错误，将显示错误信息，并根据错误的级别、性质和数量，适时结束编译过程，这时必须返回编辑修改源程序。

### 三、连接

连接的任务是利用 Quick C 系统提供连接装配程序，对编译好的目标程序与系统库目标程序以及其它已编译好的目标程序，进行连接装配，生成可执行程序文件。

若连接过程中发现错误，也要返回编辑修改源程序。

### 四、执行

执行的任务是在 DOS 支持下，调用执行已经编译、连接而生成的可执行程序文件，以达到开发 Quick C 程序的预期目的。

对执行的结果进行分析，若未达到预期目的，应检查出错误，还要返回编辑修改源程序，并重复上述各步骤。

若执行结果正确，已达到预期目的，则程序开发工作结束。其整个开发过程如图 1-2 所示。

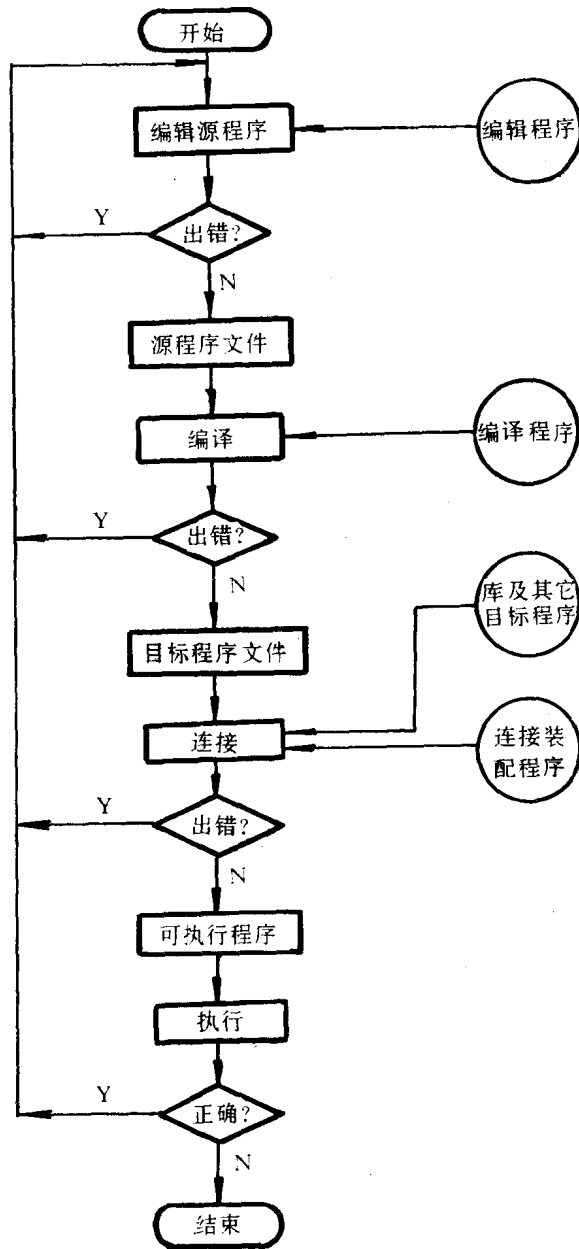


图 1.2 Quick C 程序开发过程流程图

## 习 题

1. C 语言的发展过程和 Quick C 语言产生的背景是什么？
2. Quick C 语言有哪些特点？特别是与一般 C 语言相比较有哪些新特点？
3. Quick C 语言程序的结构与风格特点是什么？
4. 用计算机解决实际问题的基本过程是什么？
5. Quick C 程序的开发过程是什么？并结合图 1.2 说明之。

## 第二章 Quick C 系统的安装与使用

在正式学习 Quick C 语言之前,首先应当了解 Quick C 系统的基本特点,其中包括 Quick C 系统的运行环境、Quick C 系统的安装、进入与退出、Quick C 系统的使用方法以及 Quick C 系统的编辑功能等。

Microsoft Quick C 2.50 编译器是一个用于开发 C 程序的复杂且功能强大然而又容易使用的集成环境。

Microsoft Quick C 将 C 语言的强大功能与集成环境结合起来,使 C 语言变得易学易用。你可以在不退出 Quick C 环境的情况下编写 C 程序并对其进行编译、连接、运行和调试。

### 2.1 Quick C 的要求与环境

#### 一、Quick C 的要求

在安装 Quick C 编译器之前,应当首先检查一下所使用的计算机系统是否满足 Quick C 编译器对它的最低要求。

Quick C 要求如下最低配置:

1. 一台可运行 DOS 2.1 或更高版本的 IBM PC 或其兼容机。
2. 一个硬盘驱动器和一个软盘驱动器。
3. 448K(建议 512K)字节的可用内存空间。

#### 二、Quick C 环境

Quick C 是一个基于窗口菜单系统的程序环境,它集成了一个编辑器,一个编译器,一个连接器,一个调试程序,一个 make 实用程序和一个联机帮助文件。

**窗口和菜单:**窗口和菜单系统能使用户快速找到所需要的命令和动作。后面我们将详细介绍如何打开和关闭窗口和如何选取菜单。

**编辑器:**当你编写 C 源程序代码时,将用较多时间使用 Quick C 编辑器,通过它将 C 语言源程序输入计算机。当然,你也可以用其它的编辑工具,如 EDLIN、WORDSTART 等,对 C 源程序进行编辑。

**编译器/连接器:**你可以在不退出编辑状态的情况下,对一个程序进行编译、连接和测试。Quick C 的集成环境缩短了你的开发时间。

**调试程序:**调试程序允许你设置断点,监视关键变量的状态,并可对程序一行一行地跟踪执行。

**联机帮助文件:**通过该文件可以向你提供关于 C 语言、运行库和由 Quick C 提供的实用程序等方面的信息,是你学习 Quick C 语言的得力助手。

## 2.2 Quick C 系统的安装

### 一、安装 Quick C 的步骤

安装 Quick C 需要以下四个步骤：

1. 对 Quick C 系统的所有原盘做一个备份。Quick C 系统共有 8 张 5.25 英寸软盘或 4 张 3.5 英寸软盘。

2. 阅读 README.DOC 文件中与安装 Quick C 有关内容。为阅读该文件，可使用联机帮助系统查阅此文件。

3. 运行 SETUP.EXE 文件。在 Quick C 系统的 setup 原盘中有一个 SETUP.EXE 程序。执行它将完成该系统的安装工作。SETUP.EXE 要做两件事。第一，它将原盘上的一部分程序（编译程序、连接程序、库管理程序、帮助系统及其它有关程序）拷贝到硬盘上。第二，它将产生一个或多个组合库。SETUP.EXE 是一个交互式程序，你所回答的问题将决定被安装的 Quick C 的环境，缺省回答都列在方括号内，每一屏的底部是对每个问题的简要说明。

4. 调整系统和环境变量。SETUP.EXE 将产生两个文件：NEW-VARS.BAT 和 NEW-CONF.SYS。你可以将 NEW-VARS.BAT 中的内容加入 AUTOEXEC.BAT（自动批处理文件）中。如果你不愿将这种改变永久化，也可以将 NEW-VARS.BAT 作为批处理文件运行。如果环境变量 files 和 Duffers 的当前值小于其在 NEW-CONF.SYS 中的相应的值，则应该修改 CONFIG.SYS 文件。文件修改之后应重新启动。

整个安装过程并不困难，并且每屏都提供帮助指导信息。

### 二、安装 Quick C 的过程——运行 SETUP

当安装 Quick C 时，先将 Setup 盘插入驱动器 A 并转换到该驱动器（键入 A:↵）。在 DOS 命令行键入 Setup 并回车，即

```
A>Setup↵
```

在安装过程中，每个问题都要求做出选择性的回答，并且每一屏的结尾都将问你是否改变任何选择项，这时你可以修改前面键入的任何回答。

安装 Quick C 必须运行 SETUP 程序，其过程共分三个阶段。

#### 1. SETUP 的第一阶段

SETUP 第一阶段显示一组问题，共分为三屏。

(1) 第一屏：库

第一屏显示以下问题：

Source of disk images [A:]

Math Options; Emulator [Y]; 8087[N]:

Memory Models; Small [Y]; Medium[N]; Compact[N]; Large[N]:

Delete the component libraries when finished [Y]:

include in combined libraries; GRPHICS.LIB [N]; PGCHART.LIB [N]:

Do you want to change any of the above options [Y]:



在回答问题时,先键入你的回答,然后按 ENTER 键。

每个问题的末尾带有一个括在方括号中的缺省回答。若直接按 ENTER 键可选择缺省值做为回答。若发现你对先前的某个问题应选择另一个回答,则只需重新运行 Setup 程序。

SETUP 的每一屏在结束时都将问你是否要改变你的选择,当对你的回答感到满意时,按 N 键(缺省回答是 Y,它将重新回到该屏的第一个问题)。

第一屏的第一个问题是问使用哪个驱动器(装有原盘的驱动器):

Source of disk images [A:]

若使用的是 A 驱动器,则直接按 ENTER 键。

要回答的第二个问题是决定哪个数学单元库包含于组合库之中:

Math Options;Emulator [Y];8087[N];

有些计算机带有 8087 或 80287 数学协处理器,而另一些计算机则不带协处理器。

当你的计算机带有 8087 或 80287 协处理器时,可将你的程序与专为使用 8087 或 80287 而设计的组合库连接,则将加速所有的浮点运算。

若将程序与仿真库连接,则它可在任何计算机上运行,无论其上是否装有协处理器。仿真库能检查协处理器是否存在,如果计算机上带有协处理器,则它完成所有的浮点运算;若没装协处理器,将仿真协处理器的动作完成浮点运算。

如果你不能决定应该包含哪个数学库,则应选择仿真库,因为它最灵活。

要回答的第三个问题是你选择哪种内存模式:

Memory Models;Small [Y];Medium [N];Compact [N];Large [N];

缺省值是小型内存模式。所谓内存模式是指内存使用的不同配置。

Quick C 支持 6 种标准内存模式:

①微型模式;②小型模式;③中型模式;④紧缩型模式;⑤大型模式;⑥巨型模式。

对于绝大多数的应用,小型模式都可胜任,微型模式除了将程序(包括源代码和数据)限制在 64K 以内,并且它生成 .COM 文件而不生成 .EXE 文件这两点之外,它和小型模式基本相同。如果你使用大型数据库管理程序,应当选择紧缩型、大型或巨型模式;若你开发的程序带有许多函数,则应当使用中型、大型或巨型模式。

巨型模式与大型模式使用相同的库,因此 SETUP 程序仅提供四种选择。大型模式与巨型模式的差别仅在于:巨型模式允许单个数组的大小超过 64K,而大型模式则将数组限制在 64K 以内。

你可以选择任何一种或所有的内存模式,SETUP. EXE 将分别为每种模式建立一个组合库。

给出的四种内存模式和两种数学程序包(仿真数学程序包和 8087 数学程序包),则可生成总共八个组合库,所建立的组合库越多,占用的硬盘空间就越大,安装的过程就越长。

要回答的第四个问题是问你是否包含图形库。

图形库 GRAPHICS. LIB 包含画线、四边形、圆和其它形状的大量函数,若你开发的程序中需要这些函数则在回答下列两个问题时按 Y 键:

Include in combined libraries;GRAPHICS. LIB [N];PGCHART. LIB [N];

若你的程序不需要这些函数,则按 N 键省去图形库。

PGCHART. LIB 库包含有用于产生高分辨率图象(线图、棒状图、直方图、离散图、饼图(百分比图))的图形函数。同样,如果你希望在程序中显示上述图形,则按 Y 键包含这个图形