

王兴波 杨利 聂勇军 编著

C/C++

程序设计

基 础 教 程

国防科技大学出版社

C/C++ 程序设计基础教程

王兴波 杨利 聂勇军 编著

国防科技大学出版社

湖南长沙

图书在版编目(CIP)数据

C/C++ 程序设计基础教程/王兴波,杨波,聂勇军编著.—长沙:国防科技大学出版社,
2004.12

ISBN 7-81099-126-4

I.C… II.①王…②杨… III. 计算机程序—C/C++ 语言—本科生教材 IV.TP303

中国版本图书馆 CIP 数据核字(2004)第 029876 号

国防科技大学出版社出版发行

电话:(0731)4572640 邮政编码:410073

E-mail:gfkdcbs@public.cs.hn.cn

责任编辑:文 慧 责任校对:何 晋

新华书店总店北京发行所经销

国防科技大学印刷厂印装

*

787×1092 1/16 印张:20.75 字数:454 千

2004 年 12 月第 1 版第 1 次印刷 印数:1-3500 册

ISBN 7-81099-126-4/TP·8

定价:29.00 元

序

刚上研究生时,我学习计算机高级语言 Turbo Pascal 4.0。因为在此以前没有接触过计算机,我学习得如痴如醉,非常认真。这时,一个同学对我讲:“你学 C 吧,C 好!”。当时我对他的话不以为然,因为我觉得计算机语言是帮助人们解决问题的工具之一,只要掌握一种就可以了。及至做硕士论文课题时,我发现自己几乎无法集中时间研究课题中的问题,而不得不将大量的时间投入到程序设计中。我认识到:只有采用一种灵活的程序设计语言才能将我解脱出来。我想起那个同学的话,开始学习 C 语言程序设计。后来我用 C 语言很漂亮地编写了硕士课题中的软件。

研究生毕业后,单位安排我给本科生讲授 C 语言程序设计,给了我一个系统地了解 C 语言的机会。开始时,我选用了谭浩强先生编著的《C 程序设计》作为教材。教学过程中,我还编写了一个计算机辅助 C 语言教学的软件。随着自己对 C 语言系统有了更加深入的了解,我萌发了自己写一本 C 语言教材的念头。我的第一本教材《C 语言电脑投影教学实践》由国防科技大学出版社出版,主要是针对当时方兴未艾的计算机辅助教学实践活动编写的。后来的科研工作使我从 C 又走向 C++。

与杨教授和聂博士结识后我们有了一个共同的想法,那就是写一本 C/C++ 教材。经过合议,大家决定让我主笔,对《C 语言电脑投影教学实践》进行整合,修订原书第一篇的内容,将原书第二篇有关 DOS 环境下开发应用的部分改写为 C++ 入门的内容,后面编写这本

《C/C++ 基础教程》。承蒙国防科技大学出版社的再次支持,本书得以顺利出版。

之所以称本书为基础教程,是因为我们希望它能够成为一本经典教材。书中特别强调一些基本概念和技巧,注重章节内容简繁的安排,适合课堂讲授。在编写过程中,杨利教授对每个章节进行了认真把关,参与了书稿的讨论;聂博士对于 C++ 部分也给出了许多建设性的意见。

与我的学生时代不同的是,C 语言已经成为计算机程序设计的普及型语言。现在几乎所有工科大学都开设了 C/C++ 语言课程,这是一件十分令人高兴的事情。曾有人言“英语、数学、计算机是未来科学技术的根本!”。笔者深信此言极其正确!作为信息领域工作的科技工作人员,我们向读者奉献此书,希望能够对培养未来科学技术工作者有所贡献。

王兴波

2004 年 10 月于长沙

目 录

第一篇 C 语言基础

1 C语言的特点与程序结构	(1)
1.1 C语言的特点	(1)
1.1.1 C语言的发展历史	(1)
1.1.2 C语言的编程特点	(1)
1.2 C语言程序的基本结构	(4)
1.2.1 程序头部	(5)
1.2.2 main函数	(6)
1.2.3 其他部分	(6)
1.2.4 示例	(7)
1.3 程序设计中的几个名词术语	(8)
1.3.1 源程序	(8)
1.3.2 编译/解释、编译/解释型语言系统	(8)
1.3.3 编译程序、目标程序	(8)
1.3.4 链接程序与链接	(8)
1.3.5 解释程序与解释	(9)
1.3.6 编译型程序系统与解释型程序系统	(9)
2 C语言的基本数据类型与简单 I/O	(10)
2.1 C语言的基本数据类型	(10)
2.2 C语言的常量与变量	(12)
2.2.1 常量	(12)
2.2.2 变量	(14)
2.2.3 标识符	(15)
2.3 C语言的基本 I/O 操作	(16)

2.3.1	printf	(17)
2.3.2	scanf	(18)
2.3.3	getch, getche, gets	(20)
2.3.4	putch, puts	(20)
2.3.5	/* ... */ C 语言的注释	(20)
2.4	复习练习题	(21)
3	C 语言的语句特征	(22)
3.1	语句结构概述	(22)
3.1.1	语句	(22)
3.1.2	流程及流程图	(24)
3.1.3	表达式	(25)
3.2	C 程序的三种基本结构及其语句	(29)
3.2.1	顺序结构	(29)
3.2.2	选择结构	(29)
3.2.3	循环结构	(33)
3.3	复习练习题	(39)
4	函 数	(41)
4.1	库函数	(41)
4.2	自定义函数	(42)
4.2.1	定义函数	(43)
4.2.2	声明原型	(43)
4.2.3	函数结构的要素	(43)
4.3	函数的参数调用	(47)
4.3.1	调用条件与方式	(47)
4.3.2	形参与实参	(48)
4.4	复习练习题	(52)
5	变量的存储属性	(54)
5.1	变量性质概述	(54)
5.1.1	变量的分类	(54)
5.1.2	数据在内存的存储	(54)
5.2	各类变量的存储属性分析	(55)

5.2.1	局部变量	(55)
5.2.2	全局变量	(59)
5.2.3	动态、静态、全局、局部的关系	(60)
5.3	复习练习题	(63)
6	宏与工程	(65)
6.1	宏及其特点	(65)
6.1.1	宏定义的类型	(66)
6.1.2	宏定义的注意事项	(67)
6.1.3	宏与函数的比较	(67)
6.2	工程与工程文件	(69)
6.2.1	工程的概念	(69)
6.2.2	建立工程文件的方法	(69)
6.2.3	开发工程时的注意事项	(70)
6.3	复习练习题	(70)
7	C 语言的数组及其应用	(72)
7.1	数组的概念与基本属性	(72)
7.1.1	数组的概念	(72)
7.1.2	一维数组的定义	(72)
7.1.3	一维数组的初始化	(73)
7.1.4	一维数组的操作	(73)
7.1.5	一维数组的下标运算	(74)
7.1.6	数组的存储属性	(75)
7.2	数组的引用	(76)
7.2.1	一维数组作为函数的参数的特点	(76)
7.2.2	使用数组的注意事项	(78)
7.3	二维数组简介	(80)
7.3.1	二维数组的定义	(80)
7.3.2	二维数组的初始化	(80)
7.3.3	二维数组的引用	(81)
7.4	字符数组与字符串	(83)
7.4.1	字符数组的定义	(84)
7.4.2	字符数组的初始化	(84)

7.4.3 字符'\0'的特殊意义	(85)
7.5 复习练习题	(86)
8 指针及其属性	(89)
8.1 C语言的指针及其性质	(89)
8.1.1 指针的概念	(89)
8.1.2 首地址	(90)
8.1.3 指针变量	(90)
8.1.4 指针变量容易产生的误区	(90)
8.2 指针变量的引用规律	(92)
8.2.1 一般引用	(92)
8.2.2 指针变量作为函数的参数	(94)
8.2.3 用指针作函数的参数的优点	(94)
8.3 指针与数组	(97)
8.3.1 指针访问数组元素	(97)
8.3.2 指针与字符串	(98)
8.3.3 指针、数组、字符串的关系	(101)
8.4 指针的其他问题	(105)
8.4.1 返回指针的函数	(105)
8.4.2 指针数组	(105)
8.4.3 指向函数指针的应用	(109)
8.5 复习练习题	(111)
9 结构体、共用体数据及其应用	(115)
9.1 结构数据类型	(115)
9.1.1 结构体的构造与特点	(115)
9.1.2 结构体类型变量的定义	(116)
9.1.3 类型与变量	(117)
9.1.4 不同定义方法的比较	(117)
9.1.5 结构体变量的存储属性	(118)
9.2 结构体变量的引用规律	(118)
9.2.1 一般引用	(119)
9.2.2 作为函数的参数的引用	(120)
9.3 结构数组、指针及函数	(122)

9.3.1	结构数组的定义与引用	(122)
9.3.2	结构指针的定义与引用	(122)
9.3.3	结构体类型的函数	(127)
9.4	链表及其应用	(128)
9.5	共用体	(132)
9.5.1	共用体的概念与特点	(132)
9.5.2	共用体变量的引用规律	(134)
9.6	复习练习题	(137)
10	C语言的文件 I/O 及其应用	(143)
10.1	C语言文件及其属性	(143)
10.2	缓冲文件的操作	(144)
10.2.1	缓冲文件的打开与关闭	(145)
10.2.2	文件的顺序读写	(147)
10.2.3	文件的随机读写	(150)
10.2.4	文件结束符	(153)
10.2.5	缓冲文件的其他操作	(154)
10.3	非缓冲文件及其操作	(155)
10.3.1	非缓冲文件的基本操作	(155)
10.3.2	非缓冲文件的应用	(156)
10.4	复习练习题	(156)
11	C语言的位运算与低级操作	(158)
11.1	位运算	(158)
11.1.1	位运算的概念	(158)
11.1.2	C语言的位操作运算	(159)
11.1.3	按位操作的一些简单应用技巧	(159)
11.2	位运算的应用	(160)
11.3	C语言的低级操作	(164)
11.3.1	int86	(165)
11.3.2	bdos	(168)
11.3.3	intdos	(170)
11.4	复习练习题	(173)

第二篇 C++ 语言入门

12 类与对象	175
12.1 C++ 语言及其特点	175
12.1.1 基于过程的思考与基于对象的思考	175
12.1.2 C++ 语言	177
12.2 C++ 的核心数据——类与对象	178
12.2.1 类	179
12.2.2 成员函数	180
12.2.3 作用域分辨符与外联成员函数的定义	181
12.2.4 构造函数	184
12.2.5 析构函数	185
12.2.6 对象	186
12.2.7 对象的初始化	190
12.2.8 对象的访问	190
12.2.9 有关类与对象的其他几个概念	195
13 继承	199
13.1 继承的概念	199
13.2 继承的定义与特征	200
13.2.1 类的派生与定义	200
13.2.2 派生类对象的创建与撤销	201
13.2.3 有关派生的其他几个常见问题	205
13.2.4 虚函数与多态性	208
14 C++ 的基本 I/O	212
14.1 C++ 的流	212
14.2 基本的流类	212
14.3 格式化的 I/O	213
14.3.1 输出运算符《与输入运算符》	213
14.3.2 标准输出 cout 与标准输入 cin	213
14.3.3 I/O 流的格式控制	214

14.4 C++ 的文件 I/O	(216)
14.4.1 文件的打开与关闭	(217)
14.4.2 文件的读写	(220)
附录 A C 语言综合测试题	(224)
第一部分 单项选择题	(224)
第二部分 阅读理解题	(240)
第三部分 测试题参考答案	(270)
附录 B Turbo C 常用库函数简介	(274)
1. Turbo C 头文件简介	(274)
2. 常用库函数简介	(274)

第一篇 C 语言基础

1 C 语言的特点与程序结构

计算机语言是学习和了解计算机知识的工具,这已经是毋庸置疑的公理。C 语言作为高级语言中的低级语言,在软件设计和开发中起着重要的作用,其学习和掌握有其独特的规律。本章介绍一些学习和掌握 C 语言的入门知识。

1.1 C 语言的特点

学习 C 语言,首先应该了解 C 语言的特点。这可以从以下两个方面入手。

1.1.1 C 语言的发展历史

C 语言不是历史最悠久的高级语言,却是生命力最旺盛的高级语言之一。

早期 C 语言是围绕能否与硬件打交道的问题而发展的。人们希望能找到一种取代汇编语言、脱离具体的计算机、能够编写操作系统的语言。C 语言是一些在当时已经使用的高级语言 ALGOL 等的基础上发展起来的。其历史可简单描述如下:

1960:ALGOL-60→面向问题的语言,远离硬件。

1963:CPL→在 ALGOL-60 的基础上向硬件接近,但规模庞大,难以实现。

1967:BCPL→简化了 CPL。

1970:B 语言→简化了 BCPL,基本能够与硬件打交道,产生了简单的 UNIX 操作系统。

1973:C 语言→完善了 B 语言,但移植性较差。

1977:可移植的 C 语言问世。

从此,C 语言开始风靡全球。

1.1.2 C 语言的编程特点

从程序设计来讲,C 语言具有如下特点:

(1)符合结构化、模块化程序设计规范

结构化模块化是 20 世纪 60 年代中期提出的一种程序设计规则,它要求程序按照模块进行结构设计,每个模块具有相对的独立性。按照这种要求设计的程序可读性好、移植性好。由于 C 语言是在 20 世纪 60 年代中后期发展起来的。因此,它一问世就必然满足这种规则。

(2)基本数据简单,数据处理内容丰富,范围广泛

一种语言的数据处理能力决定了这种语言的应用范围。C 语言具有很强的数据处理能力。尽管 C 语言直接处理的基本数据很简单(只有整数型、实数型、字符型、指针型和 void 类型),但是 C 语言通过一些规则将这些数据扩充成许多丰富的类型。这与其他一些语言不一样。有些高级语言如 PASCAL、BASIC 等的基本数据类型多,而扩充处理的灵活性较差。

(3)运算丰富,支持简单明了的运算符

与其他语言相比,C 语言的运算符和表达式要简单得多。表 1-1 是 C 语言与 PASCAL 语言的简单比较。

表 1-1

C 语言的表达式	PASCAL 语言的表达式
$x = y + z$	$x := y + z$
$x++$	$x := x + 1$
$x \& \& y$	$(x > 0) \text{ and } (y > 0)$

C 语言支持的运算有 34 种之多。

(4)功能强大,能进行多种低级操作

C 语言是功能最强大的语言之一。它能实现许多低级操作,如对磁盘、打印机、I/O 端口、鼠标等直接进行操作。利用 C 语言能编写控制硬件的程序,对一些硬件进行操作。例如,用 C 语言编制一些对磁盘操作的程序,如创建磁盘目录、删除磁盘文件、修改文件属性、读写磁盘物理扇区等程序是非常方便的。

(5)目标码质量高,可移植性好

C 语言的编译程序是高效编译系统,生成的目标代码质量高。经过实验统计,C 语言编译的目标码只比汇编语言编译的目标码效率略低,而高于其他所有高级语言的目标码。C 语言生成的目标码可以与其他语言链接生成可执行文件,其移植性能极好。

(6)丰富多彩的系统工具、库函数

所有的 C 语言系统都有一个共通的特点——提供大量的库函数。现在的 C 系统都

拥有几百个库函数。利用这些库函数,程序设计者能够开发丰富多彩的软件。

(7)能培养具有良好素质的程序员

其他高级语言的编译程序对源程序进行严格的语法检查,在编译时就把一些可能潜在的错误全部排除。而C语言编译程序则不同,C语言的编译程序对一些语法不作严格的检查。例如,如果我们定义了一个长度为10的数组,用这个数组作一个函数的参数,则下面的语句:

```
void myfunc(int array[]);  
void myfunc(int array[10]);  
void myfunc(int array[100]);
```

在C语言里全部能够通过编译,而在其他高级语言如PASCAL里,只有第二个句子能够通过。因此像这样的问题需要编程的人员在设计时做好总体规划工作,在实时编程时要心中有数。长期下去,自然会养成良好的习惯,提高素质。

本节要点

- C语言不是历史最悠久的高级语言,却是生命力最旺盛的高级语言之一。
- 符合结构化、模块化程序设计规范。
- 基本数据简单,数据处理范围广泛丰富。
- 运算丰富,支持简单明了的运算符。
- 功能强大,能进行多种低级操作。
- 目标码质量高,可移植性好。
- 丰富多彩的系统工具、库函数。
- 能培养具有良好素质的程序员。

小知识——C语言的家族

C语言的发展历史虽然短,C语言的家族却是计算机高级语言中最庞大的。针对不同的任务和对象,现在有数种C系统正在被广泛地使用。常见的有以下几种:

Turbo C 系列(简称 TC):美国 Borland 公司的产品,它有一个集成环境 IDE,使用方便,现在几乎所有的大学都采用它作为教学工具。但是它只能够编译出 DOS 或 WINDOWS 实模式的 .EXE 文件,因此适用于一般教学和科研。

Borland C/C++ 系列(简称 BC):美国 Borland 公司的产品,它有一个集成环境 IDE,使用方便,功能比 TC 更强大。2.0 以上的版本支持面向对象的程序设计思想;4.0 以下版本只能编译出 DOS 或 WINDOWS 实模式的 .EXE 文件,4.0 以上版本需要 32 位操作系统的

支持。适用于一般教学和科研。

Microsoft C/C++ 系列(简称 MSC):美国 Microsoft 公司的产品。6.0 以下的版本 C 和 TC 提供更多的库函数。7.0 以上版本能开发 WINDOWS 程序。但是由于它的早期版本没有集成环境,另外,由于 MSC 通常用于专业性较强的开发领域,现在国内使用它的人不多。

WATCOM C/C++ 系列(简称 WC):加拿大 WATCOM 公司的产品。9.0 以前版本没有集成环境;10.0 有一个 WINDOWS 集成环境 IDE。WC 的功能强大,许多专业开发需要用到它,如 NOVELL NETWARE 系统开发、AutoCAD - ADS 开发等。它不仅能够编译出 DOS 或 WINDOWS 实模式的 .EXE 文件,也能编译出保护模式下的程序。在有些专业开发方面,几乎是非它莫属。但是由于它没有 DOS 下的集成环境,国内用户极少。

High C/C++ 系列(简称 HC):MetaWare 的产品。基于 intel80386 及 intel80486 处理器以及 MS-DOS Extender 的 C 语言系统。HC 能够编写 DOS 实模式及保护模式的程序以及 WINDOWS 程序。到现在为止 HC 没有集成开发环境,也不能单独使用,它需要 Phar Lap 公司的链接器 386LINK 才能生成可执行文件。HC 的功能较强大,但是由于它的使用条件较苛刻,因此国内用户较少。HC 常用于开发一些专业软件,如 AutoCAD 的二次开发等。

Visual C/C++ 系列(简称 VC):美国 Microsoft 公司的产品。基于 WINDOWS 环境的 C 语言系统,具有集成开发环境 IDE。VC 是新型的 C 语言系统,由于它基于 WINDOWS 环境,因此特别适用于开发 WINDOWS 应用程序。2.0 以下版本可以开发 DOS 及 16 位 WINDOWS 程序。2.0 以上版本需要 32 位操作系统的支持,常用于开发 WIN95 及 WIN-NT 应用程序。

1.2 C 语言程序的基本结构

在一个源文件里,C 语言程序的结构基本上是有规律的。一般地,我们可以把一个源文件分成三个大的部分:

```
/* 程序头部 */
#include <xyz.h> /* 当 .H 文件在系统路径时 */
#include "xyz.h" /* 当 .H 文件在当前目录时 */
#define A B /* 程序中定义的各种宏 */
...
#define A B /* 程序中定义的各种宏 */
...
/* 其他包含的头文件 */
void MyFunc(); /* 自定义函数的原型 */
```



```
...  
int a,b;          /* 全局变量的定义 */  
...  
  
/* main()函数 */  
void main()  
{  
    int x,y;      /* main 的局部变量定义区 */  
    ...  
    x = 2;y = 0;  /* main 的运算处理区域 */  
    y + = x;  
    ...  
    ...  
    return;      /* main 的返回值 */  
}  
  
/* 自定义函数区域 */  
MyFunc1()  
{...}  
MyFunc2()  
{...}  
...  
...
```

1.2.1 程序头部

程序头部是由一些预处理包含命令和宏定义组成的。

在一个程序里,我们不可避免地要调用一些系统的库函数,或者使用一些系统定义的常数、变量等。这些库函数的调用规范和常数、变量的定义都是描述在一些后缀为 .H 的文本文件里。这些文件称为“头文件”。

C语言程序的第一部分要通过包含头文件来表明该程序调用了哪些库函数,定义了哪些常数、变量。其格式为:

include < C 的头文件名 >

或 # include "C 语言的头文件名"

尖括号 < > 表示引用的头文件在系统路径里;双引号""表示引用的头文件在当前目录里。

例如: # include "stdio.h"表示 stdio.h 这个文件在当前工作目录下,而 stdio 是由 stan-