



高等学校计算机类“十二五”规划教材

C语言与程序设计教程

◎主编 胡元义

CYUYAN YU CHENGXI
SHEJI JIAOCHENG



西安电子科技大学出版社
<http://www.xduph.com>

高等学校计算机类“十二五”规划教材

C 语言与程序设计教程

主 编 胡元义

副主编 宁耀斌 张治元 石 磊

李玉军 刘典型 陈 曦

参 编 佘 勇 张 悦

西安电子科技大学出版社

内 容 简 介

本书作为程序设计课程的教材,在结构上注重知识的系统性、完整性和连贯性,将理论与实践有机结合。作者在总结多年教学与实践的基础上,精选了约 200 道设计独到的例题来作为典型概念示例和程序精讲,并且兼顾 C 语言等级考试,且所有程序例题与习题都在 VC++ 6.0 环境下上机通过。对重点章节如函数和指针内容,作者采用了独创的动态图分析方法来分析程序执行中函数或指针变化的情况,使函数和指针内容中难以掌握的难题迎刃而解。本书在写法上循序渐进、深入浅出且图文并茂,力求使读者达到深入掌握 C 语言程序设计的目的。

本书除了可以作为程序设计语言教材外,还可以作为全国计算机等级考试的教材或参考书。对于从事计算机专业的工作者,本书也是难得的一本资料书。

图书在版编目(CIP)数据

C 语言与程序设计教程/胡元义主编.

—西安:西安电子科技大学出版社,2013.8

高等学校计算机类“十二五”规划教材

ISBN 978-7-5606-3137-0

I. ① C… II. ① 胡… III. ① C 语言—程序设计—高等学校—教材
IV. ① TP312

中国版本图书馆 CIP 数据核字(2013)第 166865 号

策 划 胡华霖

责任编辑 胡华霖 李惠萍

出版发行 西安电子科技大学出版社(西安市太白南路 2 号)

电 话 (029)88242885 88201467 邮 编 710071

网 址 www.xduph.com 电子邮箱 xdupfxb001@163.com

经 销 新华书店

印刷单位 陕西华沐印刷科技有限责任公司

版 次 2013 年 8 月第 1 版 2013 年 8 月第 1 次印刷

开 本 787 毫米×1092 毫米 1/16 印张 23.5

字 数 558 千字

印 数 1~3000 册

定 价 40.00 元

ISBN 978-7-5606-3137-0 / TP

XDUP 3429001-1

*** 如有印装问题可调换 ***

本社图书封面为激光防伪覆膜,谨防盗版。

前 言

作为程序设计课程的教材，本书在结构上注重知识的系统性、完整性和连贯性，在写法上循序渐进、深入浅出，将理论与实践有机结合，融知识传授与能力培养于一体。

本书主要介绍 C 语言及程序设计的相关知识。除了传统的知识外，作者在总结多年教学与实践的基础上，精选了大量内容生动、设计独到的例题来作为典型概念示例和程序精讲，并且兼顾 C 语言等级考试，许多例题就选自历年二级 C 语言等级考试试题。全书给出了 400 多道例题和习题，且所有程序例题与习题都在 VC++ 6.0 环境下上机通过。在例题分析中大量采用了图示说明，使例题更加一目了然。对重点章节如函数和指针内容，作者采用了独创的动态图分析方法来分析程序执行中函数或指针变化的情况，使函数和指针内容中难以掌握的难题迎刃而解。此外，对采用指针来指向数组元素的相关内容，作者采用了新颖的表述方法来解决同一个数组元素有多种表示法的问题。

本书第 1 章介绍了计算机的基本知识和程序设计的基本概念，并在此基础上介绍了 C 语言的发展历程和特点，同时还介绍了 C 语言程序的基本组成以及在 Visual C++ 6.0 环境下的上机操作。第 2 章介绍有关 C 语言程序设计的基础知识，如：C 语言的基本符号与基本数据类型，C 语言常量、变量的概念和使用规则，C 语言的运算符与表达式以及对 C 语言数据的输入和输出方法。第 3 章介绍了如何使用顺序、选择和循环三种基本结构来进行程序设计的方法，这是程序设计最基本的内容，也是真正掌握编程的一个必由之路。第 4 章的数组实际上是一个“量”的扩展，即由对少量的个别数据的处理编程扩展到对大量的成批数据的处理编程，因此引入了存放成批数据的数据结构——数组。第 5 章函数实际上是对“程序结构”的扩展，即程序由一个单一的主函数扩展到多个函数时如何定义和调用这些函数？参数如何在函数之间传递？计算结果又如何由被调函数返回？这些都将在第 5 章里得到解答。第 6 章的指针实际上是对变量访问的扩展，通过指针可以有效地表示各种复杂的数据结构，从而编写出精练且高效的程序来。第 7 章的结构体是在第 4 章数组简单“量”的扩展基础上的又一个更高层次的扩展，即将不同的简单“量”组合在一起形成一个复杂

的“量”——结构体，进而也可以形成一批结构体的“量”。第8章介绍了C语言程序如何处理来自外存的数据，即如何与外存文件中的数据打交道。此外，对于那些与各章内容没有紧密联系或无关紧要又较少使用的内容，则统统归于第9章“C语言与程序设计补遗”，这样使各章的知识更为紧凑、清晰和精练。本书在第3章至第8章的最后增加了一节“典型例题精讲”，以利于开拓读者解题思路和提高编程能力，达到举一反三的目的。本书所讲授的内容都是基于VC++环境的，对VC和TC有歧义的地方，书中也尽量给出说明。

本书与配套的教材《C语言与程序设计教程习题解析及上机指导》配合使用将会达到更好的效果。书中带“*”的内容为选讲内容，可根据讲授的时数进行取舍。

本书除了可以作为程序设计语言教材外，还可以作为全国计算机等级考试的教材或参考书。对于从事计算机专业的工作者，本书也是难得的一本资料书。

欢迎读者对本书的内容及本书中作者某些见解和表述方法提出批评指正。

编 者

2013年于西安

目 录

第 1 章 C 语言与程序设计简介	1	2.5.3 格式输入函数	53
1.1 计算机和程序设计基本概念	1	习题 2	57
1.1.1 计算机系统组成	1	第 3 章 三种基本结构的程序设计	62
1.1.2 程序与程序设计语言	3	3.1 程序基本结构及 C 程序语句分类	62
1.2 C 语言的发展历程和特点	4	3.1.1 程序的基本结构	62
1.2.1 C 语言发展历程	4	3.1.2 C 程序中的语句分类	63
1.2.2 C 语言主要特点	5	3.2 顺序结构程序设计	65
1.3 C 语言程序的基本组成	6	3.2.1 赋值语句	65
1.4 Visual C++ 上机操作	9	3.2.2 顺序结构程序	66
1.4.1 Visual C++ 的安装和启动	9	3.3 选择结构程序设计	68
1.4.2 Visual C++ 环境的使用	10	3.3.1 if 语句	68
习题 1	17	3.3.2 if 语句的嵌套	73
第 2 章 C 语言与程序设计基础	20	3.3.3 条件运算符与条件表达式	75
2.1 C 语言基本符号与基本数据类型	20	3.3.4 switch 语句	76
2.1.1 C 语言基本符号	20	3.4 循环结构程序设计	81
2.1.2 C 语言的数据类型	22	3.4.1 while 语句	81
2.2 常量	23	3.4.2 do...while 语句	84
2.2.1 整型常量、实型常量及符号常量	24	3.4.3 for 语句	86
2.2.2 字符常量与字符串常量	26	3.4.4 逗号运算符与逗号表达式	90
2.3 变量	28	3.4.5 break 语句、continue 语句和 goto 语句	91
2.3.1 变量的概念、定义与初始化	28	3.4.6 循环嵌套	94
2.3.2 整型变量、实型变量与字符型变量	30	3.5 典型例题精讲	97
2.4 运算符与表达式	34	习题 3	115
2.4.1 C 语言运算符简介	34	第 4 章 数组	125
2.4.2 算术运算符与算术表达式	37	4.1 一维数组	125
2.4.3 关系运算符与关系表达式	40	4.1.1 一维数组的定义	125
2.4.4 逻辑运算符与逻辑表达式	41	4.1.2 一维数组的引用和初始化	126
2.4.5 赋值运算符与复合赋值运算符	43	4.2 二维数组	130
2.4.6 表达式中数据类型自动和强制转换	45	4.2.1 二维数组的定义	130
2.5 数据的输入/输出	47	4.2.2 二维数组的引用和初始化	131
2.5.1 字符输入/输出函数	47		
2.5.2 格式输出函数	49		

4.3 字符数组和字符串	134	6.4 指针变量与函数	221
4.3.1 字符数组的定义、引用及初始化	134	6.4.1 指针变量作为函数参数	221
4.3.2 字符串	137	6.4.2 用数组名作函数参数	224
4.3.3 常用字符串处理函数	139	6.4.3 返回指针值的函数	228
4.4 典型例题精讲	143	*6.5 动态数组	230
习题 4	152	6.6 典型例题精讲	233
第 5 章 函数	160	习题 6	241
5.1 函数的概念及特点	160	第 7 章 结构体	250
5.1.1 函数的概念	160	7.1 结构体类型定义与结构体变量	250
5.1.2 函数的分类	160	7.1.1 结构体类型的定义	250
5.2 函数的定义和调用	161	7.1.2 结构体变量	253
5.2.1 函数的定义	161	7.1.3 用 typedef 定义类型标识符	258
5.2.2 函数的调用和返回值	163	7.2 结构体数组及指向结构体的指针变量	262
5.2.3 函数执行的分析方法	165	7.2.1 结构体数组	262
5.2.4 函数的声明	167	7.2.2 指向结构体的指针变量	265
5.3 变量的作用域	169	7.3 链表	268
5.3.1 全局变量与局部变量	169	7.3.1 链表的概念	268
5.3.2 函数的副作用	170	7.3.2 动态存储分配	270
5.4 函数的嵌套调用与递归调用	171	7.3.3 动态链表的建立与查找	271
5.4.1 函数的嵌套调用	171	7.3.4 链表结点的插入与删除	275
5.4.2 函数的递归调用	172	7.4 共用体	280
5.5 典型例题精讲	176	7.4.1 共用体的概念与定义	280
*5.6 递归转化为非递归研究	185	7.4.2 共用体变量的引用和赋值	281
5.6.1 汉诺塔问题递归解法	185	7.5 典型例题精讲	285
5.6.2 汉诺塔问题非递归解决	188	习题 7	293
习题 5	190	第 8 章 文件	303
第 6 章 指针	199	8.1 文件的概念	303
6.1 指针和指针变量	199	8.1.1 文件的分类	303
6.1.1 地址和指针的概念	199	8.1.2 文件指针变量及文件操作过程	304
6.1.2 指针变量的定义和初始化	200	8.2 文件的打开与关闭	305
6.1.3 指针变量的引用和运算	201	8.2.1 文件的打开	306
6.2 指针变量与数组	205	8.2.2 文件的关闭	307
6.2.1 指针变量与一维数组	205	8.3 文件的读写	308
6.2.2 指针变量与二维数组	209	8.3.1 字符读/写函数	309
6.2.3 指针数组	214	8.3.2 字符串读/写函数	311
6.3 指针变量与字符串及多级指针变量	216	8.3.3 数据块读/写函数	313
6.3.1 指针变量与字符串	216	8.3.4 格式化读/写函数	314
6.3.2 多级指针变量	218	8.4 文件的定位与随机读/写	316

8.5 典型例题精讲.....	321	9.5 枚举类型.....	348
习题 8.....	326	9.6 位运算.....	352
*第 9 章 C 语言与程序设计补遗.....	331	习题 9.....	357
9.1 变量的存储类别与生命期.....	331	附录	363
9.2 指向函数的指针变量.....	336	附录 1 ASCII 表	363
9.3 带参数的主函数 main.....	340	附录 2 C 运算符和优先级	364
9.4 编译预处理命令	342	附录 3 常用 C 库函数	365
9.4.1 宏定义命令.....	342	参考文献.....	367
9.4.2 文件包含命令	345		



第1章 C语言与程序设计简介

1.1 计算机和程序设计基本概念

1.1.1 计算机系统组成

计算机是一种能够自动、高速处理数据的工具。一个完整的计算机系统包括硬件和软件两大部分，其组成如图 1-1 所示。

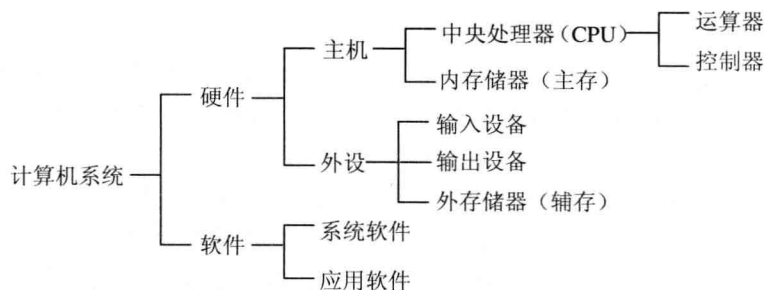


图 1-1 计算机系统组成

1. 硬件

硬件是指计算机的机器部分，即我们所见到的物理设备和器件的总称。计算机硬件结构如图 1-2 所示。

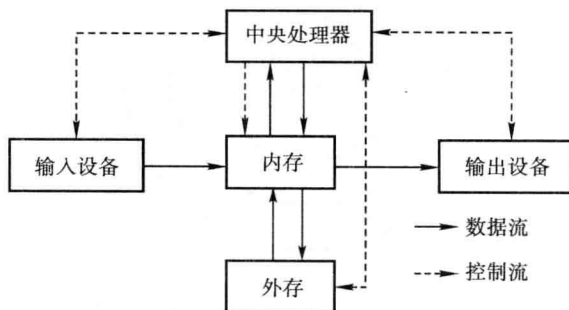


图 1-2 计算机硬件结构图

中央处理器(CPU)是计算机的核心，它由控制器和运算器两部分组成。控制器是计算机的神经中枢，它统一指挥和控制计算机各部分的工作。运算器对数据进行运算和处理。



计算机存储器分为内存储器和外存储器两种。内存储器简称内存，也叫主存，是计算机用于直接存取程序和数据的地方。内存可直接与 CPU 交换信息。其存取信息的速度快，但容量有限。外存储器简称外存，也叫辅存，常用的外存有磁盘、U 盘等。由于内存容量的限制，常用外存来存放大量暂时不用的信息，这些信息一般以文件形式存放在外存上。外存的特点是存储量大，信息可以长期保存，但存取信息的速度较慢。CPU 不能直接处理外存中的信息，处理前必须先将这些信息由外存调入内存后再进行处理，程序也只有装入内存后才可运行。

输入、输出设备是计算机与外界传递信息的通道。输入设备用于把数据、图像、命令、程序等信息输入给计算机，给计算机直接输入信息的最常用设备是键盘。输出设备用于将计算机执行的结果输出反馈给使用者，主要的输出设备有显示器和打印机。磁盘和 U 盘既可作为输入设备，又可作为输出设备。

2. 软件

软件通常指计算机系统程序和数据库，并按功能分为系统软件和应用软件两类。系统软件是指为进行计算机系统的管理和使用而必须配置的那部分软件，如：操作系统、汇编程序、编译程序等。应用软件是指针对某类专门应用需要而配置的软件，如：计算机辅助教学 CAI、财务管理软件以及火车和飞机订票系统等。

仅有硬件的计算机系统(称之为裸机)是难以进行工作的。为了对计算机所有软、硬件资源进行有效的控制和管理，在裸机基础上形成了第一层软件，这就是操作系统。

操作系统是最基本的系统软件，是对硬件机器的首次扩充。其他软件都是建立在操作系统之上的，通过操作系统对硬件功能进一步扩充，并在操作系统的统一管理和支持下运行(见图 1-3)。因此，操作系统在整个计算机系统中具有特殊的地位，它不仅是硬件与其他软件的接口，而且是整个计算机系统的控制和管理中心，它为人们提供了与计算机进行交互的良好界面。

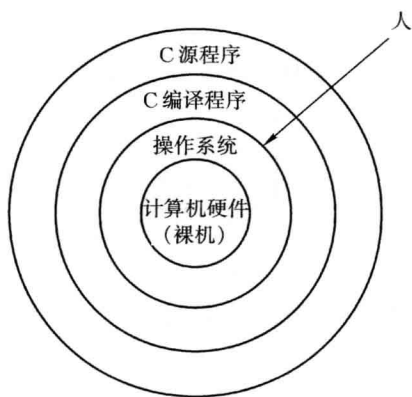


图 1-3 计算机硬件功能的扩展和人机交互的界面

综上所述，硬件是计算机的物质基础，软件是建立在硬件基础之上，对硬件功能的扩充与完善。两者缺一不可，没有软件，计算机的硬件难以工作；没有硬件，软件的功能无法实现。



1.1.2 程序与程序设计语言

要使计算机能够完成人们指定的工作，就必须把工作的具体实现步骤编写成能够由计算机识别并执行的一条条指令。计算机执行这个指令序列后，就能完成指定的任务，这样的指令序列就是程序。因此，程序就是由人编写的指挥和控制计算机完成特定功能的指令序列。书写程序所使用的语言称为程序设计语言，它是人与计算机进行信息通信的工具，而设计、编写和调试程序的过程则称为程序设计。

计算机发展到今天，程序设计语言经历了由机器语言到汇编语言，最后发展到高级语言这三个阶段。

我们知道，计算机所能执行的每一项操作称为一条指令，计算机能够执行的全部指令集合就是该计算机的指令系统。由于计算机硬件的器件特性，决定了计算机本身只能直接接受由0和1编码的二进制指令和数据，这种二进制形式的指令集合称为该计算机的机器语言，它是计算机唯一能够直接识别并接受的语言。

用机器语言编写程序很不方便且容易出错，编写出来的程序也难以调试、阅读和交流。为此，出现了用助记符代替机器语言二进制编码的另外一种语言，这就是汇编语言。汇编语言是建立在机器语言之上的，因为它是机器语言的符号化形式，所以较机器语言直观。但是计算机并不能直接识别这种符号化的语言，因此用汇编语言编写的程序必须翻译成机器语言之后才能够执行，这种“翻译”是通过专门的软件——汇编程序来实现的。

尽管汇编语言与机器语言相比在阅读和理解上有了长足的进步，但其依赖具体机器的特性是无法改变的。能够编写好汇编程序除了掌握汇编语言之外，还必须了解计算机的内部结构和硬件特性，再加上不同的计算机上的汇编语言又各不相同，这无疑给程序设计增加了难度。

随着计算机应用需求的不断增长，出现了更加接近人类自然语言的功能更强、抽象级别更高的面向各种应用的高级语言。高级语言由于接近人们的自然语言，因而使用方便，编写的程序也符合人们的习惯，能够较自然地描述各种问题，极大地提高了编程的效率，编写的程序也便于查错、阅读和修改。更为重要的是，高级语言已经从具体机器中抽象出来，摆脱了依赖具体机器的问题，使高级语言编写的程序几乎在不改动的情况下就能在任何计算机上运行。并且编程人员在编程时也无需去了解计算机内部的硬件结构。这些都是机器语言和汇编语言难以做到的。

与汇编语言一样，计算机也不能够直接识别用高级语言编写的程序，即必须经过编译程序的分析、加工，将其翻译成机器语言程序，然后才能执行。编译程序是把高级语言程序翻译成等价的机器语言程序的程序。在编译方式下，高级语言程序的执行分为两个阶段：编译阶段和运行阶段。编译阶段把高级语言程序翻译成机器语言程序，运行阶段才真正执行这个机器语言程序(见图1-4)。

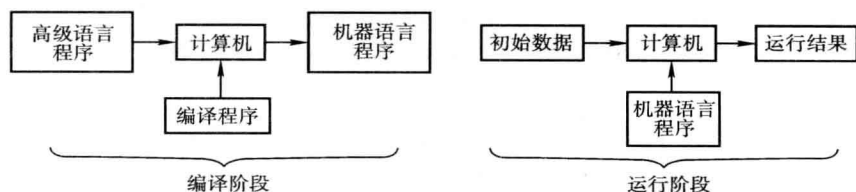


图 1-4 高级语言程序执行过程示意



例如,要给内存 16 进制地址为 1000 单元中的数据加上十进制数 10,则用机器语言、汇编语言 and 高级语言表示如下:

(1) 用 8086/8088 机器语言表示

```
10100001 11010000 00000111    /*将 16 进制 1000 地址中数据⇒AX 寄存器*/
10000011 00001010              /*给 AX 寄存器中的数据加 10*/
10100011 11010000 00000111    /*将 AX 寄存器中的数据⇒16 进制 1000 地址*/
```

(2) 用 8086/8088 汇编语言表示

```
MOV AX, [1000]                  /*将 1000 地址中数据⇒AX 寄存器*/
ADD AX, 10                      /*给 AX 寄存器中的数据加 10*/
MOV [1000], AX                  /*将 AX 寄存器中的数据⇒1000 地址*/
```

(3) 用 C 高级语言表示

```
X=X+10;                          /* X 为 1000 地址的变量名*/
```

自从 20 世纪 50 年代中期第一个高级语言——FORTRAN 语言问世以来,全球已经出现了几千种高级语言,但广泛使用的高级语言也不过数十种。例如:适用于科学计算的 FORTRAN 语言,适用于商业事务处理的 COBOL 语言,第一个体现结构化程序设计思想的 PASCAL 语言,用于人工智能程序设计的 PROLOG 语言,功能丰富的 C 语言,面向对象程序设计的 C++、Java 语言等。

1.2 C 语言的发展历程和特点

1.2.1 C 语言发展历程

由于操作系统等系统程序依赖于计算机硬件,所以以前这类系统程序主要是用汇编语言编写的。但是汇编语言程序的可读性和可移植性都很差,严重影响了系统程序的编写效率。在这种情况下,人们希望有一种语言既有高级语言可读性高、便于移植的优点,同时又具有汇编语言能够直接访问计算机硬件的特点。因此,C 语言就在这种情况下产生了。

C 语言的起源可以追溯到 ALGOL 60 语言。1963 年,英国剑桥大学在 ALGOL 60 的基础上推出了 CPL(Combined Programming Language)语言,但该语言规模较大而难以实现。1967 年,英国剑桥大学的 Martin Richards 对 CPL 语言做了简化和改进,推出了 BCPL(Basic Combined Programming Language)语言。1970 年,美国贝尔实验室的 Ken Thompson 以 BCPL 语言为基础,又做了进一步的简化,设计出简单且接近硬件的 B 语言(取 BCPL 的第一个字母),而且用 B 语言写出了第一个 UNIX 操作系统并在 DEC PDP-7 型计算机上实现。1971 年,在 DEC PDP-11 计算机上实现了 B 语言。1972 年,由美国的 D. M. Ritchie 在 B 语言的基础上设计出了 C 语言(取 BCPL 的第二个字母),并首次在 UNIX 操作系统的 DEC PDP-11 计算机上使用。

后来 C 语言又进行了多次改进,但主要还是在贝尔实验室内部使用。1977 年,D. M. Ritchie 发表了不依赖于具体机器的 C 语言编译文本《可移植 C 语言编译程序》,使 C 语言移植到其他机器时所做的工作大为简化,这也推动了 UNIX 操作系统迅速地在各种机



器上实现。随着 UNIX 操作系统的广泛使用, C 语言也迅速得到推广, 成为世界上应用最为广泛的程序设计语言之一。

1978 年, B. W. Kernighan 和 D. M. Ritchie 两人合作出版了 C 语言白皮书《The C Programming Language》, 给出了 C 语言的详细定义。1983 年, 美国国家标准化协会(ANSI)对 C 语言的各种版本作了扩充和完善, 制定了 C 语言的标准(称为 ANSI C), 这就给 C 语言程序的移植创造了更有利的环境。1990 年, ANSI C 为国际标准化组织(ISO)所接受。

微机上使用的 C 语言编译系统多为 Microsoft C、Turbo C、Borland C、Quick C 等, 它们都是按标准 C 语言编写的, 相互之间略有差异, 每一种编译系统又有不同的版本, 各版本之间也存在着差异, 版本越高则编译系统所提供的函数越多、编译能力越强, 使用也就越方便。

1.2.2 C 语言主要特点

C 语言与其他语言相比之所以发展如此迅速, 成为最受欢迎的语言之一, 主要原因是它具有强大的功能。许多著名的系统软件, 如 UNIX 操作系统就是用 C 语言编写的。归纳起来, C 语言具有下列特点:

(1) 简洁、紧凑、方便、灵活。C 语言共有 32 个关键字, 9 种控制语句, 其程序书写自由, 主要用小写字母表示, 压缩了一些不必要的成份。因此 C 语言程序比其他许多高级语言编写的程序要简练得多。

(2) 运算符丰富。C 语言的运算符包含的范围很广泛, 共 34 个运算符。C 语言把括号、下标、赋值、强制类型转换等都作为运算符处理, 从而使 C 语言的运算类型丰富、表达式类型多样化。灵活使用各种运算符可以实现在其他高级语言中难以实现的运算。

(3) 数据结构丰富。C 语言的数据类型有整型、实型、字符型、数组类型、结构体类型、共用体类型等, 具有现代高级语言所具有的各种数据类型, 能够实现复杂数据结构的各种运算。尤其是 C 语言的指针类型, 使程序运行的效率更高。此外, C 语言还具有强大的图形功能, 支持多种显示器和驱动器。

(4) 结构化语言。结构化语言的显著特点是代码及数据的分离, 即程序的各种部分除了必要的信息交流之外, 彼此独立。这种结构化方式可使程序层次清晰, 便于使用、维护和调试。C 语言是以函数形式提供给用户的, 这些函数可以方便地调用, 并有多种循环语句、条件语句来控制程序的流向, 从而使程序实现结构化。

(5) 语法检查不太严格, 程序设计自由度大。一般的高级语言语法检查比较严格, 能够检查出几乎所有的语法错误, 而 C 语言放宽了语法检查, 允许程序编写者有较大的自由度, 这是 C 语言的优点, 同时也是 C 语言的缺点。限制严格就失去了灵活性, 而强调灵活性必然放松了限制。也即, 在 C 语言程序设计中, 不要过分依赖编译器的语法检查。因此对于初学者来说, 编写一个正确的 C 语言程序比编写一个其他高级语言程序要困难一些。

(6) 允许直接访问物理地址。C 语言中含有位运算和指针运算, 能够实现对内存地址的直接访问和操作, 即 C 语言可以实现汇编语言的大部分功能, 可以直接对硬件操作。所以, C 语言既有高级语言的功能, 又兼有汇编语言(低级语言)的大部分功能, 这就是有时也称 C 语言为“中级语言”的原因。



(7) 生成目标代码效率高。C 语言仅比汇编程序生成的目标代码(即机器语言程序)执行效率低 10%~20%，这远高于其他高级语言的执行效率。

(8) 适用范围大，可移植性好。C 语言的一个突出优点就是适用于多种操作系统，如 DOS、Windows、UNIX，同时也适用于多种机型。这样，C 语言程序就可以很容易地移植到其他类型的计算机上。

1.3 C 语言程序的基本组成

我们通过下面几个简单的 C 语言程序，来大致了解一下 C 语言程序的基本组成，以便有一个初步的概念。

例 1.1 在显示器输出 “Hello,China!”。

```
#include<stdio.h>           /*使用 C 语言提供的标准输入输出函数*/
void main()                 /*主函数 main */
{
    printf("Hello,China!\n"); /*用输出函数 printf 实现输出显示字符串*/
}
```

运行结果：

Hello, China!

程序说明如下：

(1) 一个 C 程序有一个名为 main 的主函数，main 是主函数名。程序执行时就是从主函数 main 开始，具体来讲就是从 main 下面的 “{” 开始到 “}” 结束。花括号 “{ }” 中间的内容称为函数体，该函数体就是实现某种功能的一段程序。例 1.1 的函数体只有一个 printf 函数调用语句，该语句实现将双引号 “ ” 括起的字符串照原样输出到显示器上，而字符串中的 ‘\n’ 字符表示换行的意思，即在输出以下字符串：

Hello, China!

后光标换到下一行的开始处。

(2) 在 main 前加上 “void”，表示主函数 main 没有返回值。main 后面的圆括号 “()” 表明主函数 main 没有参数。

(3) printf 函数调用语句后面有一个分号 “;” 表示该语句的结束。C 语言规定：语句以分号 “;” 表示结束。

(4) 程序中的 “/* … */” 表示其中的文字是注释内容。程序中的注释是给阅读程序的人看的，是为了提高程序的可读性而加入的说明性信息。注释内容的有无都不影响程序的功能和运行结果。

(5) 程序第一行的 “#include<stdio.h>” 是编译预处理命令行，它通知编译系统，将包含输入和输出标准库函数的 stdio.h 文件作为当前源程序的一部分。像输出函数 printf 以及输入函数 scanf 都要使用 stdio.h 才能实现输出和输入。

例 1.2 求两个数 a 与 b 之和。

```
#include<stdio.h>
void main()
```



```

{
    int x,y,sum;                /*定义 x、y、sum 三个整型变量*/
    printf("Input x and y:\n"); /*在显示器上显示提示输入的信息*/
    scanf("%d%d",&x,&y);        /*由键盘输入 x 和 y 的值*/
    sum=x+y;                    /*完成 x+y 的求和并将结果送给 sum */
    printf("x+y=%d\n",sum);     /*输出求和结果*/
}

```

运行结果:

Input x and y:

12 15✓

x+y=27

程序说明如下:

为了表示变化的数据,程序中引入了变量,即通过变量来保存数据的值。在例 1.2 中定义了三个变量分别命名为 x、y 和 sum。第一个函数调用语句 printf 输出 “Input x and y:” 用来提示输入 x 和 y 值的提示信息,这时就可以由键盘输入给变量 x 和变量 y 两个数值了。scanf 是 C 语言的输入函数,语句中的 “%d” 是整型数输入格式,用来指定由键盘输入数据的格式和类型。两个 “%d%d” 表示读入两个数,输入时要用空格符分隔。scanf 语句中的 “&x,&y” 则表示将输入的两个数分别送给变量 x 和变量 y,这里的 “&” 表示地址,即数据是送到 x 和 y 对应的内存地址中。语句 “sum=x+y;” 实现 x+y 的求和并将结果送给变量 sum。第二个 printf 语句先输出 “x+y=”, 然后按格式 “%d” 将 sum 的值输出,即为:

x+y=27

例 1.3 从键盘上输入 2 个整数,在屏幕上输出它们的最大值。

```

#include<stdio.h>
int max(int x,int y);        /*函数声明*/
void main()                  /*主函数*/
{
    int a,b,c;                /*定义变量 a、b、c 为整型*/
    printf("Input a,b=");    /*输出提示字符串 "Input a,b=" */
    scanf("%d,%d",&a,&b);    /*由键盘输入 a、b 值*/
    c=max(a,b);              /*调用函数 max(), 并将 max()的返回值送给变量 c */
    printf("Max is:%d\n",c); /*输出结果*/
}

int max(int x,int y) /*定义函数 max(), 形参 x、y 为整型。打头的 int 表示返回值为整型*/
{
    int z;                    /*定义变量 z 为整型*/
    if(x>y)                   /*条件判断语句,判断 x 是否大于 y*/
        z=x;                  /* x > y 为真时将 x 值赋给 z */
    else
        z=y;                  /* x > y 为假时将 y 值赋给 z */
}

```



```
    return (z);                /*将 z 值返回给调用函数 main() */
}
```

运行结果:

Input a,b=8,12✓

Max is:12

程序说明如下:

(1) 程序中定义了两个函数: 主函数 `main` 和被调函数 `max`。注意函数 `scanf` 和 `printf` 是 C 语言提供的标准函数, 无需用户定义, 只要通过预处理命令“`#include`”将“`stdio.h`”包含在程序中就可直接使用了。而函数 `max` 是用户自行定义的函数, 即在程序中必须写出函数 `max` 的定义。

(2) 一个自定义函数由两部分组成:

① 函数首部: 包括函数名、函数类型、参数类型和参数名。例如:

int	max	(	int	x	,	int	y	)
↓	↓		↓	↓		↓	↓	
函数类型	函数名		参数类型	参数名		参数类型	参数名	

② 函数体: 即函数首部下面的花括号“`{...}`”内的部分, 如果一个函数内有多个花括号“`{...}`”, 则最外层的一对“`{...}`”为函数的范围。函数体一般包括说明部分和执行部分, 它们都是 C 语言的语句。说明部分用来定义函数内部所用到的变量及其数据类型。执行部分则是实现函数要完成的功能, 如函数 `max` 就是将 `x`、`y` 两个数中大者送给 `z`, 然后返回 `z` 值给主函数 `main`。

③ 由于本例中自定义函数放置于主函数 `main` 之后, 因此要先在主函数 `main` 之前予以声明, 即将自定义函数 `max` 的首部作为声明语句在主函数 `main` 之前重写一次(并添加“`;`”)进行声明。

(3) 主函数 `main` 中函数 `printf` 和函数 `scanf` 功能同例 1.2, 语句“`c=max(a,b);`”的作用是: 用 `a` 和 `b` 作为实参调用函数 `max`, 即将 `a` 和 `b` 的值分别传给函数 `max` 的形参 `x` 和 `y`, 且函数 `max` 最终通过 `return` 语句将 `z` 值返回给主函数 `main` 中的 `c` 变量。

通过以上几个例子, 可以概括出 C 语言程序的结构特点:

(1) C 语言程序主要由函数构成。C 语言程序中有主函数 `main`、系统提供的库函数如 `printf` 和 `scanf`, 以及由编程者自己设计的自定义函数如 `max` 等 3 种类型函数。

(2) 一个函数由说明部分和执行部分这两部分组成。说明部分在前, 执行部分在后, 这两部分顺序不能颠倒。

(3) 无论主函数 `main` 写在程序中的什么位置, 一个程序总是从主函数 `main` 开始执行的。

(4) C 语言书写格式自由。一个语句可以占多行, 一行也可写多个语句。

(5) C 语言的语句都是以分号“`;`”结尾。

(6) 为了增加程序的可读性, C 程序中可用“`/* 字符串 */`”对程序进行注释, 注释部分可以放置到程序的任何位置。

(7) 程序中可以有由“`#`”开头的预处理命令行(include 仅为其中的一种), 预处理命令



行通常应放在程序的最前面。

1.4 Visual C++ 上机操作

Visual C++ 是一个功能强大的可视化软件开发工具。自 1993 年 Microsoft 公司推出 Visual C++ 1.0 后,其新版本不断问世,现在常用的是 Visual C++ 6.0 版本。Visual C++ 6.0 不仅是一个 C++ 编译器,而且是一个基于 Windows 操作系统的可视化集成开发环境,它由许多组件组成,包括编辑器、编译器、链接器、生成实用程序、调试器以及各种各样为开发 Windows 下的 C/C++ 程序而设计的工具。本书以 Visual C++ 6.0 英文版为背景来介绍 Visual C++ 的上机操作。

1.4.1 Visual C++ 的安装和启动

如果你的计算机未安装 Visual C++ 6.0,则应先安装。Visual C++ 是 Visual Studio 的一部分,因此需要找到 Visual Studio 的光盘,执行其中的 setup.exe,并按屏幕上的提示进行安装即可。安装结束后,在 Windows 的“开始”菜单的“程序”子菜单中就会出现 Microsoft Visual Studio 6.0 子菜单。

在使用 Visual C++ 时,只需从桌面上顺序选择“开始”→“程序”→“Microsoft Visual Studio 6.0”→“Visual C++ 6.0”即可,此时屏幕上出现 Visual C++ 6.0 的主窗口,如图 1-5 所示。

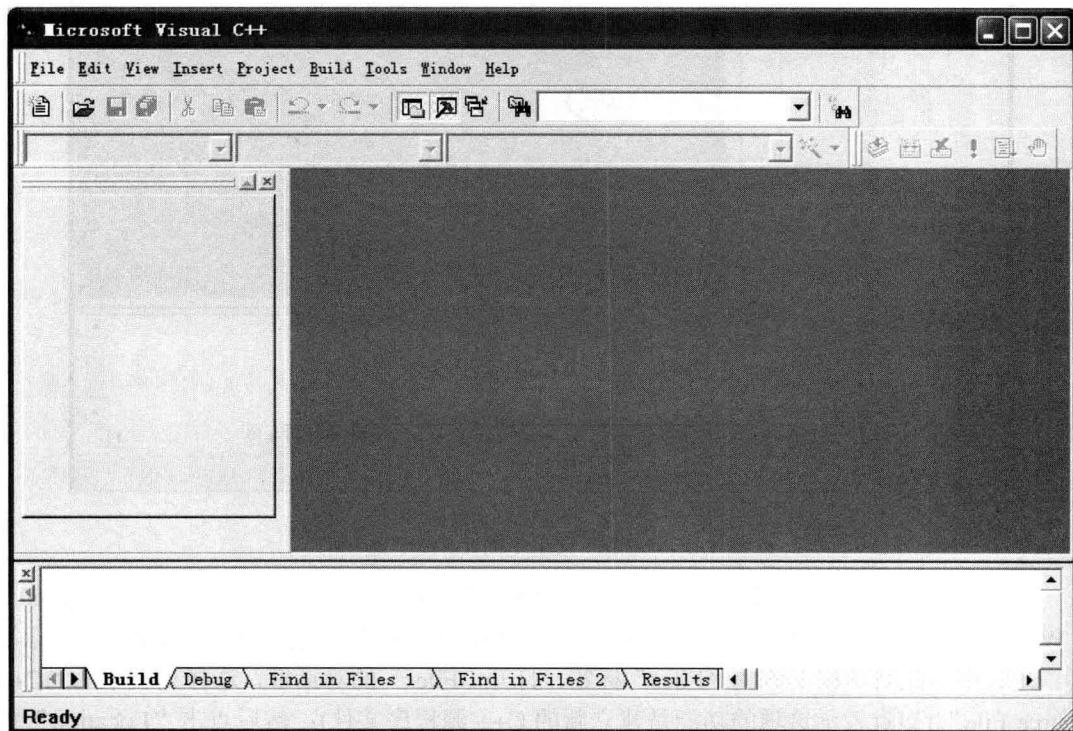


图 1-5 Visual C++ 6.0 主窗口