

第 1 章 C 语言程序设计概述

1.1 程序与程序设计语言

1.1.1 程序

计算机是可以按照人们事先编写的程序高速、精确地进行数据加工、处理的电子装置。如果我们需要计算机完成什么工作，将要完成工作的步骤用诸条指令描述出来，并把这些指令存放在计算机的存储器中，需要结果时就向计算机发出一条简单的命令，计算机就会自动逐条顺序地执行指令，全部指令执行完就得到了预期的结果。这种可以被连续执行的一条条指令的集合称为计算机的程序。也就是说，程序是计算机的指令序列，是用计算机语言对所要解决的问题中的数据以及处理问题的方法和步骤所做的完整而准确的描述。

1.1.2 计算机语言

人们用自然语言讲述和书写，目的是给别人传播信息。同样，我们使用计算机语言把我们的意图表达给计算机，目的是使用计算机，让计算机给我们完成一定信息的处理。

为了使计算机进行各种工作，就需要有一套用以编写计算机程序的数字、字符和语法规则，由这些数字、字符和语法规则组成计算机的各种指令（或各种语句），就是计算机能接受的语言。

计算机每做一次动作，完成一个步骤，都是按照已经用计算机语言编好的程序来执行的。计算机语言没有自然语言那么丰富多样，而只是有限规则的集合，所以它简单易学。但是，也正因为它根据机器的特点编制的，所以交流中无法意会和言传，更多地表现了说一不二，表现了规则的严谨。

1. 程序语言的发展

计算机语言不断从低级向高级发展，其发展过程可分为三代：机器语言、汇编语言和高级语言。

机器语言是用二进制代码表示的计算机能直接识别和执行的机器指令的集合。它是计算机的设计者通过计算机的硬件结构赋予计算机的操作功能，它与计算机同时诞生，是第一代的计算机语言。

例如，计算 $A=12+10$ 的机器语言程序如下：

10110000 00001100	: 把 12 放入累加器 A 中
00101100 00001010	: 10 与累加器 A 的值相加，结果仍放入 A 中
11110100	: 结束，停机

使用机器语言的缺点是编程工作量大，难学、难记、难修改，它只适合专业人员使用；而且由于不同的计算机，其指令系统不同，机器语言随机而异，通用性差，是面向机器的语言。

机器语言的优点是程序代码不需要翻译，所占空间少，执行速度快。

汇编语言将机器指令的二进制代码用英文助记符来表示，代替机器语言中的指令和数据。例如用 ADD 表示加、SUB 表示减、JMP 表示程序跳转等，这种指令助记符的集合就是汇编语言。

例如，计算 $A=12+10$ 的汇编语言程序：

MOV A, 12	: 把 12 放入累加器 A 中
ADD A, 10	: 10 与累加器 A 相加, 结果存入 A 中
HLT	: 结束, 停机

汇编语言的优点是克服了机器语言难读等缺点, 保持了其编程质量高, 占存储空间少、执行速度快的优点。

汇编语言的缺点是依赖于机器, 通用性差。

汇编语言源程序必须通过汇编程序翻译成机器语言, 计算机才能执行。汇编语言常用于过程控制等编程。

高级语言是一种接近于自然语言和数学公式的程序设计语言。它采用了完全符号化的描述形式, 用类似自然语言的形式描述对问题的处理过程, 用数学表达式的形式描述对数据的计算过程。高级语言主要是相对于汇编语言而言, 它并不是特指某一种具体的语言, 而是包括了很多编程语言。

例如, 计算 $A=12+10$ 的 BASIC 语言程序如下:

A=12+10	: 12 与 10 相加的结果放入 A 中
PRINT A	: 输出 A
END	: 程序结束

高级语言的优点是通用性强, 编程效率高。它使程序员可以不用与计算机的硬件打交道, 可以不必了解机器的指令系统, 集中精力解决问题本身而不受机器制约, 极大地提高了编程的效率。

高级语言源程序要通过翻译程序翻译成机器语言才能运行, 程序的效率不如优化的汇编程序高。

机器语言和汇编语言都是面向机器的语言, 属于低级语言; 而高级语言又分为面向过程和面向对象两种。前一种程序设计是数据被加工的过程, 后一种程序设计的关键是定义类, 并由类派生对象。客观世界可以分类, 对象是类的实例, 对象是数据和方法的封装, 对象间通过发送和接收消息发生联系。

面向过程的高级语言只是要求人们向计算机描述问题的求解过程, 而不关心计算机的内部结构, 它易于被人们理解和接受。典型的面向过程语言有 BASIC、FORTRAN、COBOL、C、Pascal 等。程序设计语言的发展如图 1-1 所示。

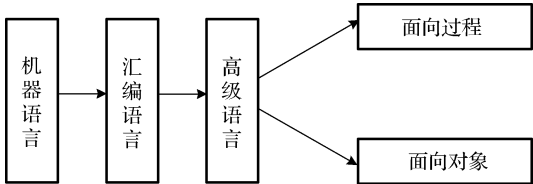


图 1-1 程序设计语言的发展

面向对象的高级语言是“面向过程”的一次革命, 如果说面向过程的语言要求人们告诉计算机怎么做, 那么面向对象的语言只要求人们告诉计算机做什么。面向对象是通过类和对象把程序所涉及的数据结构和对它施行的操作有机地组织成模块, 对数据和数据的处理细节进行最大限度的封装, 从而使开发出来的软件易重用、易修改、易调试、易扩充。

2. 语言处理程序

在所有的程序设计语言中, 除了用机器语言编制的程序能够被计算机直接理解和执行外, 其他的程序设计语言编写的源程序都必须经过一个翻译过程才能转换为计算机所能识别的机器语言程序。实现这个翻译过程的工具是语言处理程序, 即翻译程序。翻译程序也称为编译器。用非机

器语言写的程序称为源程序，通过翻译程序翻译后的程序称为目标程序。针对不同的程序设计语言编写出的程序，有各自的翻译程序，互相不通用。

翻译程序翻译源程序通常有两种方式：解释方式和编译方式。

(1) 解释方式：解释方式的翻译工作由解释程序来完成，这种方式如同“口译”。解释程序对源程序进行逐句分析，如果没有错误，将该语句翻译成一个或多个机器语言指令；然后立即执行这些指令；若解释时发现错误，会立即停止，报错并提醒用户更正代码，具体过程如图 1-2 所示，解释方式不生成目标程序。

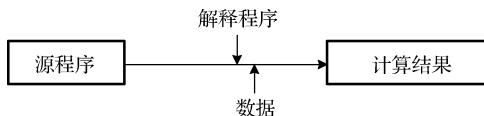


图 1-2 解释方式的执行过程

采用解释方式的优点是查找错误的语句行和修改方便。缺点是执行速度慢，采用解释方式运行的源程序，每次运行都必须重新解释，若程序较大，错误发生在程序的后面，则前面运行的结果是无效的，解释程序无法对整个程序进行优化。

(2) 编译方式：翻译工作由编译程序完成。如同“笔译”，在纸上记录翻译后的结果。编译程序过程：对源程序编译产生目标程序，连接程序将目标程序和有关的程序库组合成可执行程序，编译方式的执行过程如图 1-3 所示。

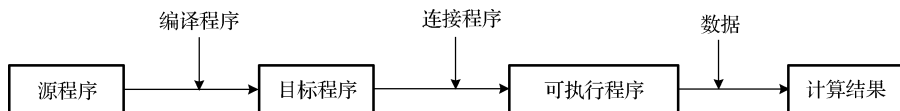


图 1-3 编译方式的执行过程

采用编译方式的优点是程序执行速度快，产生的可执行程序可以脱离编译程序和源程序独立存在并反复运行，但修改源程序后都必须重新编译生成目标程序。

一般高级语言（C/C++、Pascal、FORTRAN、COBOL 等）都是采用编译方式。

1.2 程序设计（解决什么问题、如何解决、实现方法）

计算机之所以能够产生如此大的影响，其原因不仅在于人们发明了机器本身，更重要的是人们为计算机开发出了不计其数的能够指挥计算机完成各种各样工作的程序。正是这些功能丰富的程序给了计算机无尽的生命力，是程序设计工作智慧的结晶。

所谓程序，是用计算机语言对所要解决的问题中的数据以及处理问题的方法和步骤所做的完整而准确的描述，而程序设计就是用某种程序语言编写这些解决问题的步骤的过程。对数据的描述就是指明确所要处理问题中的数据结构（即数据和数据之间的关系）；对问题处理方法和步骤的描述就是算法。因此，数据结构与算法是程序设计过程中密切相关的两个方面。著名计算机科学家 Niklaus Wirth 教授关于程序提出了著名公式：程序=数据结构+算法。这个公式说明了程序设计的主要任务。

如何进行程序设计呢？一个简单的程序设计一般包含以下步骤。

1. 分析问题

使用计算机解决具体问题时，首先要对问题进行充分的分析，确定问题是什么，解决问题

的步骤又是什么。针对所要解决的问题，找出已知的数据和条件，确定所需的输入、处理及输出对象。

2. 确定数据结构和算法

在分析求解问题的基础上，将所研究问题的数据和数据间关系抽象出来，确定程序中数据的类型和数据组织存储形式，即确定存放数据的数据结构。针对问题的分析和确定的数据结构，选择合适的算法加以实现。注意，这里所说的“算法”泛指解决某一问题的方法和步骤。

3. 编制程序

根据确定的数据结构和算法，用一种程序设计语言把这个解决方案严格地描述出来，也就是编写出程序代码。

4. 调试程序

在计算机上用实际的输入数据对编好的程序进行调试，分析所得到的运行结果，进行程序的测试和调整，直至获得预期的结果。

5. 分析结果

对程序执行结果进行验证和分析，发现程序中存在的问题并修改完善。

6. 写出程序的文档

程序是提供给用户使用的，如同正式的产品应当提供产品说明书一样，正式提供给用户使用的程序，必须向用户提供程序说明书。内容应包括程序名称、程序功能、运行环境、程序的装入和启动、需要输入的数据，以及使用注意事项等，为程序的使用、修改做好基础工作。

1.3 C 语言的发展

C 语言是世界上广泛流行的计算机高级程序设计语言，它于 1973 年由美国贝尔实验室设计发布。由于 C 语言同时具备高级语言的优点和低级语言的效率，而且拥有很好的可移植性，使它成为程序员最喜欢的语言。

C 语言起源于对系统程序设计的深入研究和发展的。C 语言是贝尔实验室的 Ken Thompson 和 Dennis Ritchie 等人开发的 UNIX 操作系统的“副产品”。Thompson 独自编写了 UNIX 操作系统的最初版本，与同时代的其他操作系统一样，UNIX 最初也是用汇编语言编写的，用汇编语言编写的程序往往难以调试和改进，UNIX 也不例外。Thompson 意识到需要用一种更加高级的编程语言来完成 UNIX 系统未来的开发，于是他设计了一种小型的 B 语言。B 语言是在 BCPL (Basic Combined Programming Language) 的基础上开发的，而 BCPL 是在 CPL (Combined Programming) 语言的基础上开发的，CPL 可以追溯到最早的语言之一 Algol60 语言。

1970 年，K. Thompson 用 B 语言在 PDP-7 机上实现了第一个实验性的 UNIX 操作系统。1972 年，贝尔实验室的 Dennis M. Ritchie 为克服 B 语言的诸多不足，在 B 语言的基础上重新设计了一种语言，由于是 B 语言的后继，故称为 C 语言。1973 年，贝尔实验室的 K. Thompson 和 Dennis M. Ritchie 合作，首先用 C 语言重新改写了 UNIX 操作系统，在当时的 PDP-11 计算机上运行。此后，C 语言作为 UNIX 操作系统上标准的系统开发语言，伴随着 UNIX 操作系统的发展，C 语言越来越广泛地被人们接受和应用。

1977 年，出现了独立于具体机器的 C 语言编译版本。由于 C 语言的独立和推广，也推动了 UNIX 操作系统在各种机器上的迅速实现。

1978 年，美国电话电报公司（AT&T）贝尔实验室正式发表了 C 语言。同时，由 B.W.Kernighan 和 D.M.Ritchie 合著了著名的《The C Programming Language》一书，简称为《K&R》，也有人称之为《K&R》标准。此书中介绍的 C 语言成为后来广泛使用的 C 语言版本基础，它被称为标准 C 语言。

1983 年，美国国家标准化协会（ANSI）根据各种 C 语言版本对 C 的扩充和发展，颁布了 C 语言的新标准 ANSI C。ANSI C 比标准 C 有了很大的扩充和发展。

1987 年，美国国家标准化协会在综合各种 C 语言版本的基础上，又颁布新标准，为了与标准 ANSI C 区别，所以称为 87 ANSI C。1990 年，国际标准化组织 ISO 接受了 87 ANSI C 作为 ISO C 的标准。这是目前功能最完善、性能最优良的 C 新版本。目前流行的 C 编译系统都是以它为基础的。

1989 年，X3J11 提出了一个报告[ANSI 89]，后来这个标准被 ISO 接受为 ISO/IEC 9899—1990。

1990 年，国际标准化组织 ISO（International Organization for Standards）接受 89 ANSI C 作为 ISO C 的标准（ISO 9899—1990）。1994 年，ISO 修订了 C 语言的标准。

1995 年，ISO 对 C90 做了一些修订，即 1995 基准增补 1（ISO/IEC/9899/AMD1:1995）。1999 年，ISO 再次对 C 语言标准进行了修订，在保留原来 C 语言特征的基础上，针对实际编写应用程序的需要，增加了一些功能，尤其是完善了 C++ 语言中的一些功能，并将其命名为 ISO/IEC9899:1999。

2001 年和 2004 年先后进行了两次技术修正。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，但不同版本的 C 编译系统所实现的语言功能和语法规则又略有差别。

2011 年 12 月，ISO 正式公布 C 语言新的国际标准草案：ISO/IEC 9899:2011。

新的标准修订了 C11 版本，提高了对 C++ 的兼容性，并将新的特性增加到 C 语言中。新功能包括支持多线程，基于 ISO/IEC TR 19769:2004 规范下支持 Unicode，提供更多用于查询浮点数类型特性的宏定义和静态声明功能。

1.4 C 语言的特点

C 语言发展如此迅速，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。许多著名的系统软件，如 DBASE III PLUS、DBASE IV 都是由 C 语言编写的。用 C 语言加上一些汇编语言子程序，就更能显示 C 语言的优势了，像 PC-DOS、WORDSTAR 等就是用这种方法编写的。从语言体系和结构上讲，它是结构化程序设计语言；但从用户应用、实现难易程度、程序设计风格等角度来看，C 语言的特点又是多方面的。

1. C 是中级语言

C 语言通常称为中级计算机语言，它把高级语言的基本结构和语句与低级语言的实用性结合起来。中级语言并没有贬义，不意味着它功能差、难以使用，或者比 BASIC、Pascal 那样的高级语言原始，也不意味着它与汇编语言相似，会给使用者带来类似的麻烦。作为中级语言，C 允许对位、字节和地址这些计算机功能中的基本成分进行操作。C 语言可以实现汇编语言的大部分功能，可以直接操作计算机硬件如寄存器，各种外设 I/O 端口等。C 语言的指针可以直接访问内存物理地址，类似汇编语言的位操作可以方便地检查系统硬件的状态等。

所有的高级语言都支持数据类型的概念。一个数据类型定义了一个变量的取值范围和在其上

操作的一组运算。常见的数据类型是整型、字符型和实数型。虽然 C 语言有五种基本数据类型，但与 Pascal 或 Ada 相比，它却不是强类型语言。C 程序允许几乎所有的类型转换。例如，字符型和整型数据能够自由地混合在大多数表达式中进行运算。这在强类型高级语言中是不允许的。

C 语言的另一个重要特点是，它仅有 32 个关键字，这些关键字就是构成 C 语言的命令。和 IBM PC 的 BASIC 相比，后者包含的关键字达 159 个之多。

2. C 是结构式语言

虽然从严格的学术观点上看，C 语言是块结构（block-structured）语言，但是它还是常被称为结构化语言。这是因为它在结构上类似于 ALGOL、Pascal 和 Modula-2（从技术上讲，块结构语言允许在过程和函数中定义过程或函数。用这种方法，全局和局部的概念可以通过“作用域”规则加以扩展，“作用域”管理变量和过程的“可见性”。因为 C 语言不允许在函数中定义函数，所以不能称之为通常意义上的块结构语言）。

结构化语言的显著特征是代码和数据的分离。这种语言能够把执行某个特殊任务的指令和数据从程序的其余部分分离出去、隐藏起来。获得隔离的一个方法是调用使用局部（临时）变量的子程序。通过使用局部变量，我们能够写出对程序其他部分没有副作用的子程序。这使得编写共享代码段的程序变得十分简单。如果开发了一些分离很好的函数，在引用时，我们仅需要知道函数做什么，不必知道它如何做。切记：过度使用全局变量（可以被全部程序访问的变量）会由于意外的副作用而在程序中引入错误。

结构化语言比非结构化语言更易于程序设计，用结构化语言编写的程序的清晰性使得它们更易于维护。这已是人们普遍接受的观点了。C 语言的主要结构成分是函数 C 的独立子程序。

在 C 语言中，函数是一种构件（程序块），是完成程序功能的基本构件。函数允许一个程序的诸多任务被分别定义和编码，使程序模块化。可以确信，一个好的函数不仅能正确工作且不会对程序的其他部分产生副作用。

3. C 语言功能齐全

C 语言具有各种各样的数据类型，包括整型、实型、字符型、数组类型、指针类型、结构体类型、联合体类型等，因此，便于实现各种复杂的数据结构，如线性表、栈、队列、树和图等等的运算。

并引入了指针概念，可使程序效率更高。另外，C 语言也具有强大的图形功能，支持多种显示器和驱动器。而且，计算功能、逻辑判断功能也比较强大，可以实现决策目的。

指针是 C 语言的一大特色，可以说是 C 语言优于其他高级语言的一个重要原因。就是因为它有指针，所以可以直接进行靠近硬件的操作，但是对 C 语言的指针操作不做保护，也给它带来了许多不安全的因素。C++在这方面做了改进，在保留了指针操作的同时又增强了安全性，受到了一些用户的支持。但是，由于这些改进增加语言的复杂度，也为另一部分人诟病。Java 则吸取了 C++的教训，取消了指针操作，也取消了 C++改进中一些备受争议的地方，在安全性和适合性方面均取得良好的效果，但其本身解释在虚拟机中运行，运行效率低于 C++/C。一般而言，C、C++、Java 被视为同一系的语言，它们长期占据着程序使用榜的前三名。

4. C 语言适用范围大

C 语言还有一个突出的优点是适合于多种操作系统，如 DOS、UNIX，也适用于多种机型。C 语言拥有一个庞大的数据类型和运算符集合，这个集合使得 C 语言具有强大的表达能力，可以用来编写各种系统软件和应用软件。

5. 程序生成的目标代码质量高、程序运行效率高

试验表明，C语言源程序生成的运行程序的效率仅比汇编程序的效率低10%~20%，但C语言的编程速度快，程序可读性好，易于调试、修改和移植，这些优点是汇编语言所无法比拟的。

6. 可移植性好

C语言程序非常容易移植。可移植性表示为某种计算机写的软件可以用到另一种机器上去。举例来说，如果为苹果机写的一个程序能够方便地改为可以在IBM PC上运行的程序，则称为是可移植的。C语言在某种计算机上编制的程序，只需少量改动，甚至不做修改就能移植到各种不同型号的计算机上。

7. C语言存在的不足之处

编程自由度大，编译程序查错纠错能力有限，给不熟练的程序员带来一定困难；C语言的理论研究及标准化工作也有待推进和完善。

高级语言接近人类语言和人们习惯使用的数学用语，不依赖于具体的机器，有严格的语法规则。相对于机器语言和汇编语言，高级语言易学易用，所编写的程序易读易改，通用性更强。C语言把高级语言的基本结构与低级语言的高效实用性很好地结合起来，不失为一个出色而有效的现代通用程序设计语言。C语言在计算机程序语言研究方面具有一定价值，由它引出了许多后继语言。C语言已经成为世界上广泛流行的计算机高级程序设计语言，由于C语言同时具备高级语言的优点和低级语言的效率，而且拥有很好的可移植性，因此它成为程序员最喜欢的编程语言之一。C语言以功能强大、数据结构丰富、目标代码质量高、程序运行效率高、可移植性好等特点成为众多程序员学习程序设计的首选语言，对整个计算机工业和应用的发展都起了很重要的推动作用。

1.5 C程序的基本组成

1.简单C程序介绍

用C语言的语句编写的程序称为C语言程序（简称C程序）或C语言源程序。本节通过几个简单的C程序实例，介绍C程序的基本组成和结构，使读者对C语言和C程序的特性有初步的了解。

【例 1.1】在屏幕上输出：Hello,World!。

C语言源程序如下：

```
//example1.1 The first C Program
#include <stdio.h>      //头文件
void main()
{
    printf("Hello,World!\n");
}
```

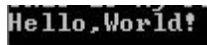


图 1-4 例 1.1 运行结果

运行结果如图 1-4 所示。

分析：

①void main 表示“主函数”。每个C程序都必须有且只能有一个main函数，它是每一个C程序执行的起始点（入口点）。void 表示该函数没有返回值。

②用{}括起来的是主函数main的函数体。main函数中的所有操作（或：语句）都在这一对{}之间，即main函数的所有操作都在main函数体中。

③本程序的“主函数”main 中只有一条函数调用语句，printf()是 C 语言的库函数，用于程序中数据的输出（显示在屏幕上），本例是将一个字符串“Hello,World!\n”的内容输出，即在屏幕上显示：Hello,World!。

④每条语句用“;”号结束语句。

⑤头文件是每一个 C 程序必不可少的组成部分，因为一个 C 程序至少要包含输入或输出函数，而这些函数将被存放在以“.h”为扩展名的头文件中，比如本例的 printf 函数就包含在头文件“stdio.h”中。

⑥/*example1.1 The first C Program*/为注释，注释只是为了改善程序的可读性（提示、解释作用），在编译、运行时不起作用（编译时会跳过注释，目标代码中不会包含注释）。注释可以放在程序任何位置，并允许占用多行，只是需要注意“/*”、“*/”匹配，不要嵌套注释。

注释与软件的文档同等重要，要养成书写注释的良好习惯，这对软件的维护相当重要。因为程序是要给别人看的，自己也许还会看自己几年前编制的程序，清晰的注释有助于读者理解程序、算法的思路。

在软件开发过程中，还可以用注释帮助调试程序，即暂时屏蔽一些不需要运行的语句，以后可以方便地恢复。

【例 1.2】计算两数之和，并输出结果。

```
main()                /*计算两数之和*/
{
    int a,b,sum;        /*定义变量*/
    a=11;
    b=22;                /*以下 3 行为 C 语句*/
    sum=a+b;
    printf("sum=%d\n",sum);
}
```



图 1-5 例 1.2 运行结果

运行结果如图 1-5 所示。

分析：

①同样，此程序也必须包含一个 main 函数作为程序执行的起点。{}之间为 main 函数的函数体，main 函数所有操作均在 main 函数体中。

②int a,b,sum;是变量声明。声明了三个具有整数类型的变量 a,b,sum。C 语言的变量必须先声明再使用。

③a=11;b=22;是两条赋值语句。将整数 11 赋给整型变量 a，将整数 22 赋给整型变量 b。a,b 两个变量的值分别为 11, 22。注意这是两条赋值语句，每条语句均用“;”结束。

也可以将两条语句写成两行：

```
a=11;
b=22;
```

由此可见，C 程序的书写可以相当随意，但是为了保证容易阅读要遵循一定的规范。

④sum=a+b;是将 a,b 两变量内容相加，然后将结果赋值给整型变量 sum。此时，sum 的内容为 33。

⑤printf("sum=%d\n", sum);是调用库函数输出 sum 的结果。%d 为格式控制，表示 sum 的值以十进制整数形式输出。程序运行后，输出（显示）：sum=33。

【例 1.3】求两个数的较大值。

```
int max(int a,int b);           /*函数说明*/
main()                          /*主函数*/
{
    int x,y,z;                  /*变量说明*/
    int max(int a,int b);       /*函数说明*/
    printf("input two numbers:\n");
    scanf("%d%d",&x,&y);        /*输入 x,y 值*/
    z=max(x,y);                 /*调用 max 函数*/
    printf("maxmum=%d",z);      /*输出*/
}
int max(int a,int b)            /*定义 max 函数*/
{
    if(a>b)return a;else return b; /*把结果返回主调函数*/
}
```

运行结果如图 1-6 所示。

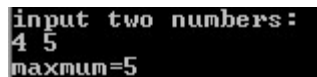


图 1-6 例 1.3 运行结果

分析：程序由两个函数组成，即主函数和 max 函数。函数之间是并列关系，可从主函数中调用其他函数。max 函数的功能是比较两个数，然后把较大的数返回给主函数。max 函数是一个用户自定义函数。因此，在主函数中要给出说明（程序第三行）。可见，在程序的说明部分中，不仅可以有变量说明，还可以有函数说明。关于函数的详细内容将在以后章节中详细介绍。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。

程序的执行过程是，首先在屏幕上显示提示串，请用户输入两个数，回车后由 scanf 函数语句接收这两个数送入变量 x,y 中，然后调用 max 函数，并把 x,y 的值传送给 max 函数的参数 a,b。在 max 函数中比较 a,b 的大小，把大者返回给主函数的变量 z，最后在屏幕上输出 z 的值。

2. C 程序的基本组成

综合上述三个例子，我们对 C 程序的基本组成和程序结构有了一个初步了解。

①C 程序由函数构成（函数是 C 程序的基本单位），所有的 C 程序都由一个或多个函数构成。其中，main 函数必须有且只能有一个。

②被调用的函数可以是系统提供的库函数，也可以是用户根据需要自己设计编写的函数。程序的全部工作由各个函数完成。编写 C 程序就是编写一个个函数。

③main 函数（主函数）是每个程序执行的起始点。无论 main 函数在程序中的哪个位置，一个 C 程序总是从 main 函数开始执行，也是从主函数结束。

④一个函数由函数首部和函数体两部分组成：

函数类型 函数名(参数类型及参数名表)

函数体：函数首部下方用一对{}括起来的部分。函数体一般包括声明和执行两部分。

例如函数 max：

函数类型	函数名称	参数类型	参数名	参数类型	参数名	
↓	↓	↓	↓	↓	↓	
int	max	(int	x,	int	y)	/*函数首部*/
{	/*函数开始*/					
	int z; /*声明部分，定义变量*/					
	if(x>y)					

```

z=x;
    else
z=y;    /*函数体, 执行部分*/
    return z;
}    /*函数结束*/

```

⑤C 程序书写格式自由, 一行可以写几个语句, 一个语句也可以写在多行上。每条语句的最后必须有一个分号“;”表示语句的结束。

⑥可以使用/* */对 C 程序中的任何部分做注释。注释可以提高程序的可读性, 使用注释是编程人员的良好习惯。实践中, 编写好的程序往往需要修改、完善, 事实上没有一个应用程序是不需要修改、完善的。很多人会发现自己编写的程序在经历了一些时间以后, 由于缺乏必要的文档、必要的注释, 最后连自己都很难再读懂, 需要花费大量时间重新思考、理解原来的程序, 这浪费了大量的时间。如果一开始编程时就对程序做必要的注释, 虽然刚开始麻烦一些, 但日后可以节省大量的时间。一个实际的系统往往是多人合作开发的, 程序文档、注释是其中重要的交流工具。

⑦C 语言本身不提供输入/输出语句, 输入/输出操作是通过调用库函数 (scanf, printf 等) 完成。

输入/输出操作涉及具体计算机硬件, 把输入/输出操作放在函数中处理, 可以简化 C 语言和 C 的编译系统, 便于 C 语言在各种计算机上实现。不同的计算机系统需要对函数库中的函数做不同的处理, 以便实现同样或类似的功能。

3. C 语言基本语法成分

①C 程序由函数构成 (C 是函数式的语言, 函数是 C 程序的基本单位)。

- 一个 C 源程序至少包含一个 main 函数, 也可以包含一个 main 函数和若干个其他函数。函数是 C 程序的基本单位。
- 被调用的函数可以是系统提供的库函数, 也可以是用户根据需要自己编写设计的函数。C 是函数式的语言, 程序的全部工作都由各个函数完成。编写 C 程序就是编写一个个函数。
- C 函数库非常丰富, ANSI C 提供 100 多个库函数, Turbo C 提供 300 多个库函数。

②main 函数 (主函数) 是每个程序执行的起始点。

一个 C 程序总是从 main 函数开始执行, 而不论 main 函数在程序中的哪个位置。可以将 main 函数放在整个程序的最前面, 也可以放在整个程序的最后, 或者放在其他函数之间。

③源程序中可以有预处理命令 (例 1.1 中的 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。

④C 语言字符集: C 语言字符集由字母、数字、各种符号组成 (在字符串常量和注释中还可以使用汉字等其他图形符号), 由字符集中的字符可以构成 C 语言的语法成分 (如标识符、关键字、运算符等)。

- 字母: A~Z, a~z。
- 数字: 0~9。
- 各种符号: 空白符、标点符号、特殊字符等。

空白符: 空格, 制表符 (跳格), 换行符 (空行) 的总称。空白符除了在字符、字符串中有意义外, 编译系统忽略其他位置的空白。空白符在程序中只是起到间隔作用。在程序的恰当位置使用空白将使程序更加清晰, 增强程序的可读性。

标点符号、特殊字符:

```

!   #   %   ^   &   +   -   *   /   =   ~   <   >   \   |   .   ,   ;   :   ?
'   "   (   )   [   ]   {   }

```

以上字符集构成了 C 语言基本的语法元素。

⑤标识符：用来标识变量名、符号常量名、函数名、数组名、类型名等的有效字符序列。简单地说，标识符就是一个名字。

C 语言标识符命名规则：

标识符只能由字母、数字和下画线三种字符组成，且第一个字符必须为字母或下画线。

例如：

合法的标识符：sum, f1, average, _total, Class, day, stu_name, lotus_1_2_3。

不合法的标识符：M.D.John, \$123, #45, 7days, x-y, a+b。

注意：

- C 语言中大小写字母是两个不同的字符。例如：sum 不同 Sum, BOOK 不同 book。
- 标识符不能与“关键字”同名，也不与系统预先定义的“标准标识符”（如 main、printf 等）同名。
- 长度在一定范围内。各编译系统都有自己的规定和限制，一般环境允许取 32 个字符。
- 建议：标识符命名应当有一定的意义，做到见名知义。例如：sum 代表和，aver 代表平均值。

⑥关键字：C 语言规定的具有特定意义的字符串，用户只能按其预先规定的意义来使用它，不能改变其意义。C 语言可以使用以下 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	sizeof	static
struct	switch	typedef	union	unsigned	void	volatile	while

⑦运算符：运算符将常量、变量、函数连接起来组成表达式，表示各种运算。运算符可以由一个或多个字符组成，各运算符的运算规则将在后续的学习中陆续介绍。

⑧分隔符：逗号、空格等。起分隔、间隔作用。

⑨数据：程序要加工处理的对象。一个程序的主要任务就是对数据进行处理，一般程序中包括说明数据、输入数据、加工数据、输出数据等。

⑩C 语言本身不提供输入/输出语句，输入/输出的操作是通过调用库函数（scanf, printf）完成。

输入/输出操作涉及具体计算机硬件，把输入/输出操作放在函数中处理，可以简化 C 语言和 C 语言的编译系统，便于 C 语言在各种计算机上实现。不同的计算机系统需要对函数库中的函数做不同的处理，以便实现同样或类似的功能。

不同的计算机系统除了提供函数库中的标准函数外，还按照硬件的情况提供一些专门的函数。因此，不同计算机系统提供的函数数量、功能会有一定差异。

4. C 程序书写注意事项

①C 程序书写格式自由。

- 一行可以写几个语句，一个语句也可以写在多行上。
- C 程序没有行号，也没有 FORTRAN、COBOL 那样严格规定书写格式（语句必须从某一行开始）。
- 每条语句的最后必须有一个分号“;”表示语句的结束。

②可以使用 /*...*/（可换行写）或者 //...（不可换行写）对 C 程序中的任何部分做注释，注释可以提高程序可读性，使用注释是编程人员的良好习惯。

- 代码注释的作用一直都被程序员们广泛讨论，很多人认为注释不是必要的，写注释是因为

代码可读性太差。实际上，编写程序比阅读程序要容易，代码的可读性显得尤为重要，所以程序员必须通过一些技术来提高程序的可读性，其中一点就包括代码注释。同一行的注释不要写得很长，更不要为了注释而注释，不要重复读者已经知道的内容，应该把长注释放在代码上面，短注释放在代码后面。

- 一个实际的系统往往是多人合作开发的，程序文档、注释是其中重要的交流工具。
- ③从书写清晰，便于阅读、理解和维护的角度出发，在书写程序时一般应遵循以下规则：
- 一个说明或一个语句占一行。
- 用{ }括起来的部分，通常表示程序的某一层结构。{ }一般与该结构语句的第一个字母对齐，并单独占一行。
- 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。

1.6 C 语言的上机执行过程

编写出 C 程序仅仅是程序设计工作中的一个环节，写出来的程序需要在计算机上进行调试运行，直到得到正确的运行结果为止。

1.6.1 C 程序的开发过程

开发一个 C 程序一般要经过四个步骤，即编辑、编译、连接和运行，才能得到程序的运行结果。下面分别说明上机执行过程。

1. 编辑

编辑是 C 语言源程序的输入和修改，并以文本文件的形式存放在磁盘上。其标识为：“文件名.C”。其中，文件名是由用户指定的符合操作系统文件命名规则的任意字符组合，扩展名要求为“.C”，表示是 C 语言源程序。例如 file1.c、t.c 等。用于编辑源程序的软件是编辑程序。编辑程序是提供给用户书写程序的软件环境，可用来输入和修改源程序。

2. 编译

编译是把 C 语言源程序翻译成用二进制指令表示的目标文件。编译过程由编译系统提供的编译程序完成。编译程序自动对源程序进行句法和语法检查，当发现错误时，将错误的类型和所在的位置显示出来，提供给用户，以帮助用户修改源程序中的错误。如果未发现句法和语法错误，就自动形成目标代码并对目标代码进行优化后生成目标文件。目标程序的文件标识是：“文件名.obj”，扩展名“.obj”是目标程序的文件类型标识。

3. 连接

连接过程是连接程序（也称链接程序或装配程序）将目标程序、库函数或其他目标程序连接装配成可执行程序。可执行程序的文件名为：“文件名.exe”，扩展名“.exe”是可执行程序的文件类型标识。

4. 运行

运行程序是指将可执行程序投入运行，得到程序处理的结果。如果程序运行结果不正确，可重新回到第一步，重新对程序进行编辑修改、编译和运行。

与编译、连接不同，运行可以脱离语言处理环境。因为它是对一个可执行程序进行操作，与 C 语言源程序本身已经没有联系，所以可以在语言开发环境下运行，也可直接在操作系统下运行。

以上四个步骤的执行过程如图 1-7 所示。其中，图中虚线框内是 C 集成开发系统提供的语言处理程序和 C 标准库函数。

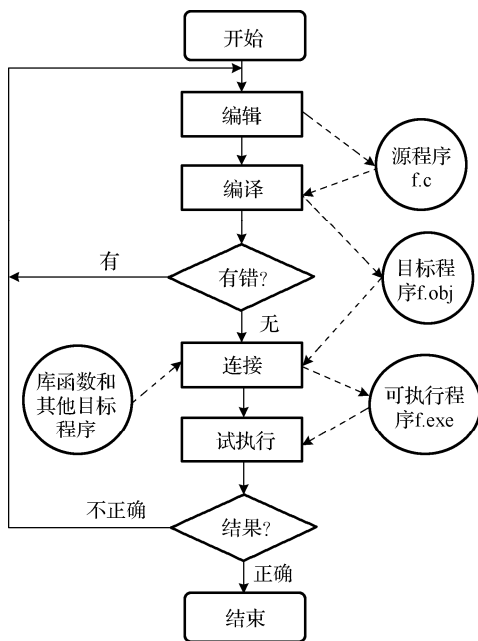


图 1-7 开发 C 程序的步骤

1.6.2 Visual C++6.0 开发环境及程序测试与调试

1. Visual C++6.0 开发环境

Visual C++6.0 是 Microsoft 公司推出的运行在 Windows 操作系统中的交互式、可视化集成开发软件，集程序的编辑、编译、连接、调试等功能于一体，为编程人员提供了一个既完整又方便的开发平台，它不仅支持 C++ 语言，也支持 C 语言，成为常用的 C++ 语言和 C 语言的开发环境。

Visual C++ 是一个集成的语言开发环境，首先必须在计算机系统中安装，安装完毕后，选择 Windows 的“开始”菜单中的“Microsoft Visual Studio 6.0 Microsoft”|“Visual C++6.0”命令，启动 Visual C++6.0；也可在桌面建立 Visual C++6.0 的快捷方式图标，双击即可启动；或者直接双击一个 C 语言或 C++ 语言的源程序，启动 Visual C++6.0。下面简单介绍在 VC 环境下，开发 C 语言源程序的过程。

2. C 语言源程序开发过程

1) 编辑 C 源程序代码

启动 Visual C++6.0 (简称 VC) 集成开发环境，在 VC 主窗口 (如图 1-8 所示) 中选择“File|New”菜单命令，弹出“New (新建)”对话框。选择“New”对话框上的“File”选项卡，并选择“C++ Source File”选项，建立 C 语言源程序，如图 1-9 所示。

在对话框右侧的“Location (位置)”文本框中输入或选择新文件的存储位置。

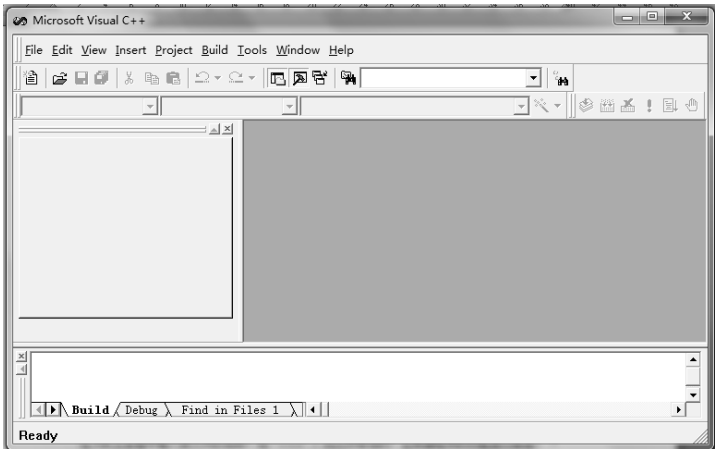


图 1-8 VC 主窗口

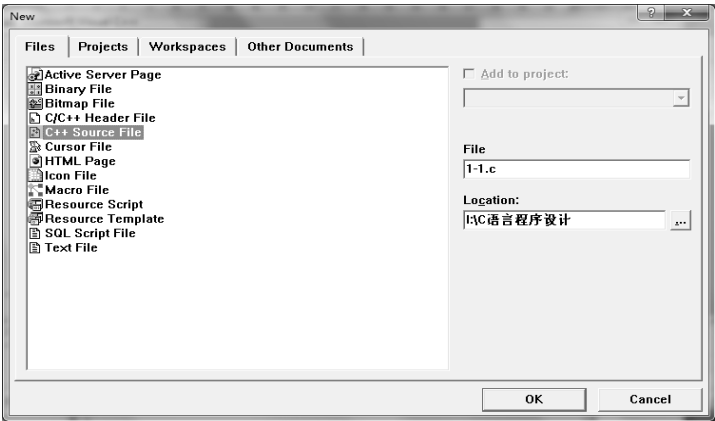


图 1-9 “New（新建）”对话框

在“File”文本框中输入新文件的名称，如“1-1.c”。对文件命名时，应根据程序的功能进行命名。

单击“OK”按钮，回到 VC 主窗口，在编辑窗口看到光标闪烁，此时可输入和修改源程序，如图 1-10 所示。

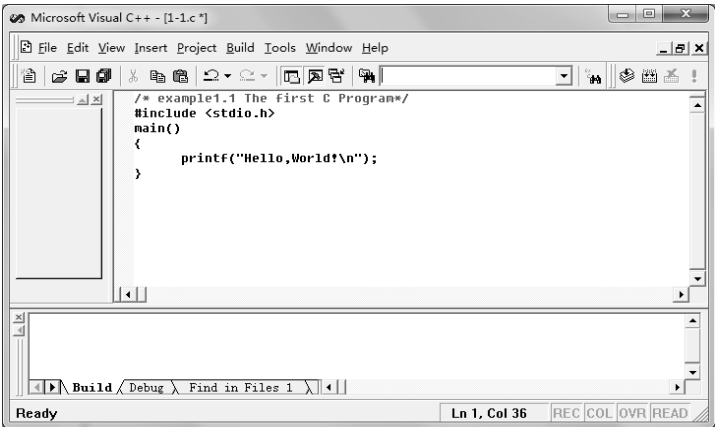


图 1-10 C 源程序编辑窗口

选择“File|Save”菜单命令，或单击工具栏中的“保存”按钮或按 Ctrl+S 组合键保存源程序文件。

2) 编译、连接和运行程序

在 VC++6.0 中，文件是在项目的管理之下，而项目则在工作区的管理之下，在编辑 C 源代码时没有进行“新建工作区，在工作区下建立项目，在项目下建立 C 源程序代码”的过程，在进行 C 源程序代码编译时，可为源代码建立默认的工作区和项目。

(1) 编译。单击工具栏上的“Compile”按钮（)或选择“Build”菜单中的“Compile 1-1.c (编译 1-1.c)”命令，编译系统对文件 1-1.c 进行编译，弹出如图 1-11 所示的对话框，单击“是”按钮，表示同意由 VC 系统建立一个默认的项目工作区和一个工程，并对源程序进行编译；若单击“否”按钮，将不会对源程序进行编译。

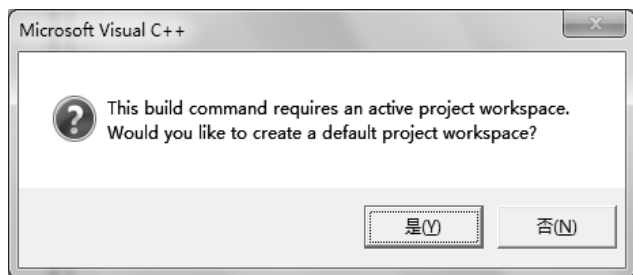


图 1-11 创建工作区对话框

若编译成功，则在 VC 输出窗口显示“0 error(s),0 warning(s)”，表示没有致命错误（error），也没有警告（warning），编译系统生成一个目标文件 1-1.obj，之后可以进行程序的连接与执行，VC 运行编译后的 VC 窗口如图 1-12 所示。

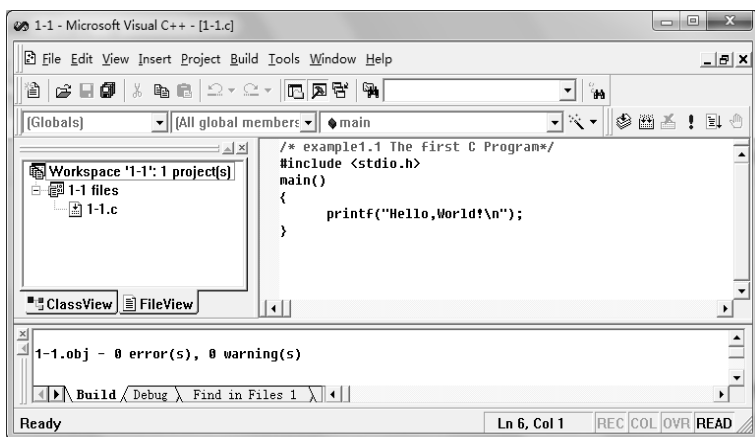


图 1-12 编译后的 VC 窗口

若编译有错，则在输出窗口显示错误信息。按 F4 键或双击错误提示行，在源程序出错行左侧出现标记，并且光标定位到此行。此时，应检查标记所在行（可能会是标记前一行或多行）的程序代码，找出错误的原因并改正，然后再编译，若出现错误，则再修改，直到编译通过为止。

(2) 链接。单击工具栏上的“Build”按钮（)，或选择“Build”菜单下的“Build 1-1.exe”命令，或按 F7 键对目标文件 1-1.obj 进行连接。若连接成功，则生成一个可执行文件 1-1.exe。

(3) 运行。单击工具栏上的“Execute”按钮, 或选择“Build”菜单下的“!Execute 1-1.exe”命令, 或按 Ctrl+F5 组合键。运行后产生结果输出窗口。

程序执行完毕, 单击“File”菜单下的“Save (保存)”命令, 对源程序进行保存, 也可以按 Ctrl+S 组合键来保存文件。

程序执行、保存完毕, 单击“File”菜单下的“Close Workspace (关闭工作区)”命令, 如图 1-13 所示, 按以上步骤即可进行下一个程序的开发工作。

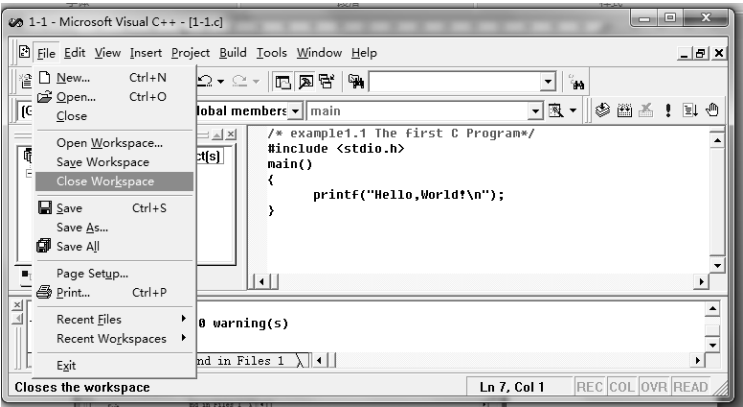


图 1-13 关闭工作区界面

工程工作区就像一个“容器”, 由它来“盛放”相关工程的所有有关信息, 创建默认工程工作区之后, 系统将创建出一个相应的工作区文件 (.dsw), 用来存放与该工作区相关的信息; 另外还将创建出的其他几个相关文件, 即工程文件 (.dsp) 以及选择信息文件 (.opt) 等。

开发一个 C 语言源程序, 也可以按照“创建工程工作区”|“创建工程”|“在工程下创建 C 源程序”的步骤进行, 下面简略说明过程。

在如图 1-8 所示的 VC 主窗口下选择“file”|“new”菜单命令, 出现如图 1-9 所示的对话框, 选择“Projects”选项卡下的“Win32 Console Application”, 在右侧的“Location”文本框和“Project name”文本框中填入工程存放在磁盘的位置 (目录或文件夹位置) 以及工程的名字, 单击“OK”按钮进入如图 1-14 所示的界面。这个界面主要询问用户想要构成一个什么类型的工程。

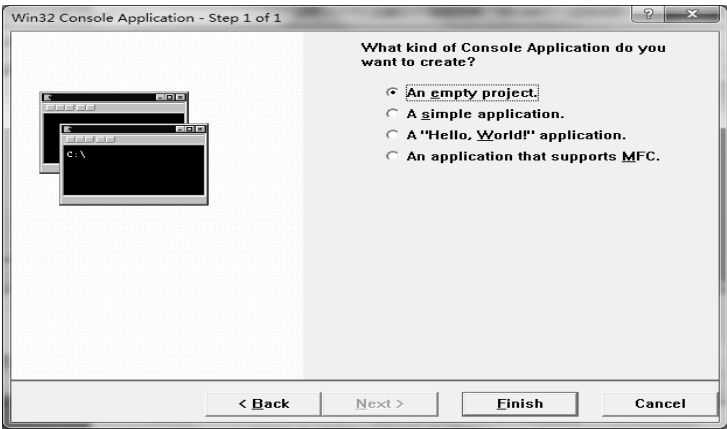


图 1-14 创建工程界面

选择“An empty project.”项将生成一个空的工程, 工程内不包括任何东西。

选择“A simple application.”项将生成包含一个空的 main 函数和一个空的头文件的工程。选择“A"Hello, World!"application.”项与选择“A simple application.”项没有什么本质的区别,只是需要包含显示出“Hello, World!”字符串的输出语句。若选择“An application that supports MFC.”项,则可以利用 VC6 所提供的类库来进行编程。

选择“Project”菜单的“Add To Project”子菜单下的“new”项,在出现的对话框的“Files”选项卡中,选择“C++ Source File”项,在右中处的“File”文本框中为将要生成的文件取一个名字,取名为 1-1.c (见名知义),然后单击“OK”按钮,进入输入源程序的编辑窗口,输入 C 源程序代码,编译、连接、执行即可。

1.6.3 Turbo C 2.0 开发环境及程序测试与调试

1. Turbo C 简介

1) Turbo C 的产生与发展

Turbo C 是美国 Borland 公司的产品。Borland 公司是一家专门从事软件开发、研制的大公司。该公司相继推出了一套 Turbo 系列软件,如 Turbo BASIC、Turbo Pascal、Turbo Prolog,这些软件很受用户欢迎。该公司在 1987 年首次推出 Turbo C 1.0 产品,其中使用了全然一新的集成开发环境,即使用了一系列下拉式菜单,将文本编辑、程序编译、连接以及程序运行一体化,大大方便了程序的开发。1988 年,Borland 公司又推出 Turbo C 1.5 版本,增加了图形库和文本窗口函数库等,而 Turbo C 2.0 则是该公司在 1989 年出版的。Turbo C 2.0 在原来集成开发环境的基础上增加了查错功能,并可以在 Tiny 模式下直接生成.COM (数据、代码、堆栈处在同一 64KB 内存中)文件。还可对数学协处理器(支持 8087/80287/80387 等)进行仿真。

Borland 公司后来又推出了面向对象的程序软件包 Turbo C++,它继承、发展了 Turbo C 2.0 的集成开发环境,并包含了面向对象的基本思想和设计方法。

1991 年,为了适用 Microsoft 公司的 Windows 3.0 版本,Borland 公司又对 Turbo C++ 做了更新,即 Turbo C 的新一代产品 Borland C++也问世了。

2) Turbo C 2.0 基本配置要求

Turbo C 2.0 可运行于 IBM-PC 系列微机,包括 XT、AT 及 IBM 兼容机。此时要求 DOS 2.0 或更高版本支持,并至少需要 448KB 的 RAM,可在任何彩、单色 80 列监视器上运行。支持数学协处理器芯片,也可进行浮点仿真,这将加快程序的执行。

3) Turbo C 2.0 内容简介

Turbo C 2.0 包括以下主要文件:

INSTALL.EXE	安装程序文件
TC.EXE	集成编译
TCINST.EXE	集成开发环境的配置设置程序
TCHELP.TCH	帮助文件
THELP.COM	读取 TCHHELP.TCH 的驻留程序
README	关于 Turbo C 的信息文件
TCCONFIG.EXE	配置文件转换程序
MAKE.EXE	项目管理工具
TCC.EXE	命令行编译

TLINK.EXE	Turbo C 系列连接器
TLIB.EXE	Turbo C 系列库管理工具
C0?.OBJ	不同模式启动代码
C?.LIB	不同模式运行库
GRAPHICS.LIB	图形库
EMU.LIB	8087 仿真库
FP87.LIB	8087 库
*.H	Turbo C 头文件
*.BGI	不同显示器图形驱动程序
*.C	Turbo C 例行程序（源文件）

其中，上面的?分别为：

T	Tiny（微型模式）
S	Small（小模式）
C	Compact（紧凑模式）
M	Medium（中型模式）
L	Large（大模式）
H	Huge（巨大模式）

2. Turbo C 环境下 C 程序的运行

为了使计算机能按照人们的意志工作，就要根据问题的要求，编写相应的程序。程序是一组计算机可以识别和执行的指令，每一条指令使计算机执行特定的操作。程序可以用高级语言或汇编语言编写，用高级语言或汇编语言编写的程序称为源程序。C 程序源程序的扩展名为“.c”。事实上，我们编写的程序，不管采用什么计算机语言，都是源程序，很少有人会用机器语言去编程！源程序不能直接在计算机上执行，需要用“编译程序”将源程序翻译为二进制形式的代码。

源程序经过“编译程序”翻译所得到的二进制代码称为目标程序。目标程序的扩展名为“.obj”。目标代码尽管已经是机器指令，但是还不能运行，因为目标程序还没有解决函数调用问题，需要将各个目标程序与库函数连接，才能形成完整的可执行的程序。

目标程序与库函数连接，形成的完整的可在操作系统下独立执行的程序称为可执行程序。可执行程序的扩展名为“.exe”（在 DOS/Windows 环境下）。

1) 启动 TC（执行 TC .exe 文件）

进入 Turbo C 2.0 集成开发环境中后，屏幕上显示如图 1-15 所示。



图 1-15 Turbo C 2.0 集成开发环境