

C 语言概述

计算机语言是为了完成人与计算机之间交流而诞生的一种语言,也称为程序设计语言,是指一个能完整、准确和规则地表达人们的意图,并用以指挥或控制计算机工作的“符号系统”,它经历了从机器语言、汇编语言到高级语言的发展历程。本书中介绍的 C 语言就是计算机语言发展到高级语言的一种产物,是一种应用范围非常广泛的程序设计语言,它有着丰富的运算符、表达式和数据结构,具有表达能力强、目标程序效率高、语言简单灵活、容易移植等优点。本章主要介绍 C 语言的发展历史、特点以及 C 程序的基本结构和 C 程序的开发过程。

学习目标

- (1) 了解计算机语言和 C 语言的发展历史和特点。
- (2) 理解 C 程序的结构特点。
- (3) 掌握 C 程序的开发过程。
- (4) 掌握 Visual C++ 6.0 调试 C 程序的方法。

1.1 C 语言简介

C 语言是一种面向过程的计算机程序设计语言,是举世公认的优秀结构化程序设计语言之一。本节详细介绍了计算机语言及 C 语言的发展历程,并阐述了 C 语言的主要特点。

本节学习目标:

- 了解计算机语言的相关概念。
- 了解 C 语言的发展历程。
- 理解 C 语言的主要特点。

【任务提出】

任务 1.1: 分别写出用机器语言、汇编语言和高级语言实现对两数求和(例如求 $2+3$)的核心代码。

【任务分析】

本任务提出的目的在于比较三种语言编程特点,学习者可通过网络查找资料等方法了解机器语言、汇编语言的加法指令和指令格式以及高级语言中加法运算符的用法,再写出相应的核心代码即可。

【任务实现】

1. 机器语言

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一种机器指令的集合。它的

一条指令包括操作码和地址码,操作码用来表示该指令所要完成的操作(如加、减、乘、除、数据传送等),其长度取决于指令系统中的指令条数。地址码用来描述该指令的操作对象,它可以是操作数,也可以是操作数的存储器地址或寄存器地址(即寄存器名)。不同型号的计算机具有不同的指令系统,以 8086/8088 兼容机为例,用机器语言完成计算 $2+3$ 的核心代码如下:

```
10110000 00000010 ;将 2 放进累加器 acc 中
00101100 00000011 ;累加器的值与 3 相加,结果仍然在累加器中
```

2. 汇编语言

汇编语言指令是机器指令的一种符号表示,它最典型的一种指令格式如下:

标号: 功能助记符 目的操作数,源操作数 ;注释

以 MCS-51 为例,用汇编语言完成计算 $2+3$ 的核心代码如下:

```
MOV A, #02H ;将 2 放进累加器 A 中
ADD A, #03H ;累加器的值与 3 相加,结果仍然在累加器中
```

3. 高级语言

高级语言是以人类的日常语言为基础的一种编程语言,允许人们用熟悉的自然语言和数学语言编写程序代码。高级语言并不是特指某一种具体的语言,而是包括很多编程语言。以其中的 C 语言为例,完成计算 $2+3$ 的核心代码如下:

```
int a; //定义整型变量 a
a = 2 + 3; //将 2 与 3 相加,结果存放在变量 a 中
```

【知识讲解】

1. 计算机语言的发展

计算机语言也称为程序设计语言,即编写计算机程序所用的语言。计算机语言种类较多,伴随着计算机技术的发展而不断变化,总的来说,可分为机器语言、汇编语言和高级语言三大类。

(1) 机器语言

机器语言是计算机诞生和发展初期使用的语言。它是由 0 和 1 组成的机器指令的集合,是一种最低级的计算机语言。计算机发明之初,人们只能用计算机能直接“听懂”的语言去命令计算机干这干那,也就是需要写出一串串由 0 和 1 组成的指令序列交由计算机执行,这种计算机能够直接识别的语言就是机器语言。机器语言直接面向硬件,所以执行速度快,但不同机型间不能通用,因此可移植性差,编程效率低;另外,对使用计算机的人来说,这也是十分难懂的语言,它难读、难写、难记,容易出错。

(2) 汇编语言

汇编语言是人们为了降低使用机器语言编程的困难,进行了有益改进的语言。汇编语言用一些简洁的英文字母、符号串来替代一个特定的二进制串指令,比如,用 ADD 代表加法,

MOV 代表数据传递等,这样一来,人们很容易读懂并理解程序在干什么,纠错及维护都变得方便了。汇编语言使程序设计工作前进了一大步,它拥有机器语言的全部优点,且比机器语言更容易理解,易于调试和修改。但是它对计算机硬件的依赖性还是很强,可移植性较差,大部分语句还是和机器指令一一对应,语句功能不强,程序一般比较冗长、复杂,容易出错,因此编写较大的程序时仍然很烦琐。在今天的实际应用中,它通常被应用在底层,往往用于硬件操作和高要求的程序优化的场合。驱动程序、嵌入式操作系统和实时运行程序往往需要用到汇编语言。

(3) 高级语言

高级语言是一种比较接近自然语言和数学语言的程序设计语言。由表达不同意义的“关键字”和“表达式”按照一定的语法规则组成,完全不依赖机器的指令系统,因此,编程者不需要具备太多的专业知识,也不必了解机器的内部结构和工作原理。和汇编语言相比,高级语言通用性好、易学易用、可移植性好且易于交流和推广。目前使用较多的高级语言有 FORTRAN、VB、VC、FoxPro、C、C++、PASCAL 等,这些语言的语法、命令格式各不相同。

2. C 语言的发展历史

在 C 语言诞生以前,系统软件主要是用汇编语言编写的,应用汇编语言进行编程可以实现对计算机硬件的直接操作,但是由于它依赖于计算机硬件,其代码可读性和可移植性都很差。而一般的高级语言又难以实现对计算机硬件的直接操作,所以人们希望有一种计算机语言既能有高级语言的优点,同时又能实现低级语言的功能,C 语言就是在这种背景下产生的。

C 语言的原型为 ALGOL 60 语言(简称为 A 语言)。1963 年,剑桥大学将 ALGOL 60 语言发展成为 CPL 语言。1967 年,剑桥大学的 Martin Richards 对 CPL 语言进行了简化,产生了 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,设计出很简单且很接近硬件的 B 语言(取 BCPL 的首字母),并且他用 B 语言写了第一个 UNIX 操作系统。1972 年,美国贝尔实验室的 D. M. Ritchie 在 B 语言的基础上最终设计出了一种新的语言,他取了 BCPL 的第二个字母作为这种语言的名字,这就是 C 语言。

1973 年年初,C 语言的主体完成。1983 年,美国国家标准化协会(ANSI)根据 C 语言问世以来的各种版本对 C 语言的发展和扩充制定了 ANSI C 标准。1990 年,C 语言成为国际标准化协会(ISO)通过的标准语言。

3. C 语言的主要特点

C 语言的发展如此迅速且经久不衰,成了最受程序设计者欢迎的语言之一,主要是由于它的功能十分强大,许多操作系统和应用软件都是用 C 语言编写的。归纳起来 C 语言具有下列特点。

(1) 简洁紧凑,灵活方便

C 语言一共只有 32 个关键字和 9 种控制语句,程序书写形式自由,其代码区分大小写。

(2) 运算符丰富

C 语言的运算符包含的范围很广泛,共有 34 种运算符。C 语言把括号、赋值、强制类型转换等都作为运算符处理。从而使 C 语言的运算类型极其丰富,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) 数据类型丰富

C语言的数据类型有整型、实型、字符型、数组类型、指针类型、结构体类型、共用体类型等,能用来实现各种复杂的数据结构的运算。并引入了指针概念,从而使程序效率更高。

(4) 表达方式灵活实用,程序设计自由度大

C语言提供多种运算符和表达式值的方法,对问题的表达可通过多种途径实现,其程序设计更主动、灵活。它的语法限制不太严格,程序设计自由度大,如整型数据与字符型数据及逻辑型数据有时可以通用等。

(5) 允许直接访问物理地址,对硬件进行操作

由于用C语言可以直接访问物理地址,进而实现直接对硬件进行操作,因此它既具有高级语言的优点,又具有低级语言的许多功能,能够像汇编语言一样对位(bit)、字节和地址进行操作,而这三者是计算机最基本的工作单元,可用来开发系统软件。

(6) 生成目标代码质量高,程序执行效率高

C语言描述问题比汇编语言迅速,程序员工作量小且代码可读性好,易于调试、修改和移植,而代码质量与汇编语言相当。

(7) 可移植性好

在一个环境中用C语言编写的程序,不改动或稍加改动就可移植到另一个完全不同的环境中运行。

(8) 具有结构化的控制语句

用函数作为程序单元便于实现程序的模块化。C语言是理想的结构化程序设计语言,符合现代编程风格的要求。

【知识小结】

(1) 计算机语言经历了从机器语言、汇编语言到高级语言的发展历程。其发展过程是功能不断完善、描述问题的方法越来越贴近人类思维方式的过程。

(2) C语言是一种结构化的程序设计语言,兼有高级语言和低级语言的许多优点。它既用来编写系统软件,又用来开发应用软件,已成为一种通用程序设计语言。

1.2 C程序简介

由C语言编写的程序称为C程序,所有的C程序都是由一个或多个文件组成的,一个文件又可以由一个或多个函数组成。

本节学习目标:

- 了解C程序的基本结构。
- 掌握C程序的结构特点和书写规则。

【任务提出】

任务 1.2: 制作一张自己的名片,内容包括姓名、性别、职务。

【任务分析】

本任务要求在屏幕上输出个人信息,其解题思路是在主函数 main() 中调用输出函数 printf()

输出要显示的内容。

【任务实现】

参考代码如下：

```
1 /*This is the first C program */
2 #include <stdio.h>
3 int main()
4 {
5     printf("*****\n");
6     printf("姓名: 张三\n");
7     printf("性别: 男\n");
8     printf("职务: 班长\n");
9     printf("*****\n");
10    return 0;
11 }
```

程序运行结果如图 1.1 所示。



```
*****
姓名: 张三
性别: 男
职务: 班长
*****
Press any key to continue
```

图 1.1 任务 1.2 程序运行结果

程序分析：

(1) 程序第 1 行是注释信息,用于说明程序的功能和目的,编译系统会跳过注释行不对其进行处理。在 C 程序中,注释由“/*”开始并由“*/”结束,可以实现多行注释;由“//”可以实现单行注释。

(2) 程序第 2 行是预处理命令,C 语言的预处理命令都是以符号“#”开头。其中 `stdio.h` 是一个头文件,也可写成 `#include "stdio.h"`。

(3) 程序第 3 行的 `main` 是主函数名,`int` 是函数类型,表示返回值为一个整型数据。一个 C 语言程序有且仅有一个 `main()` 函数。用 `{}` 括起来的部分是一个程序模块,在 C 语言中也称为分程序。每个函数中都至少有一个分程序。程序执行时就是从 `main()` 函数开始,具体讲就是从“{”开始到“}”结束。

(4) 程序第 5~10 行是函数体部分,其中 `printf` 是 C 语言的内部函数名,其功能是将指定的字符或数据显示在计算机的屏幕上(双引号不显示,“\n”表示回车换行)。`return` 是 C 语言中返回语句的关键字,用于函数内部,此处的含义是返回一个整型数据并退出函数。

(5) 分号“;”是 C 语言的执行语句和声明语句的结束符。

【知识讲解】

1. C 程序的基本结构

一个完整的 C 程序是由多个模块组成的,每个模块可由一至多个源程序文件组成。C 程序的结构示意图如图 1.2 所示。

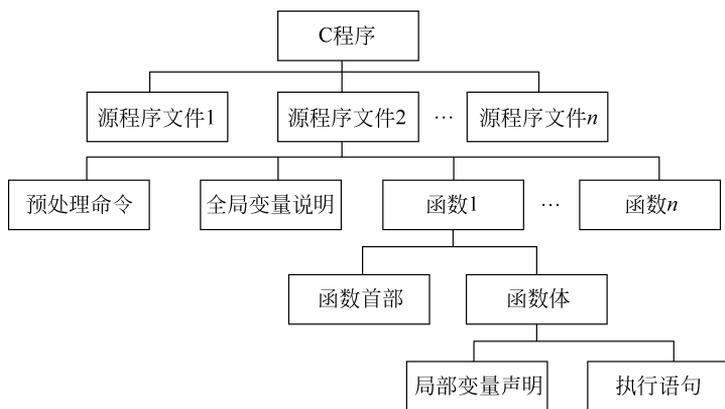


图 1.2 C 程序的结构示意图

2. C 语言程序的结构特点

(1) 组成 C 语言程序的基本单位是函数。每一个 C 程序都是由一个或多个函数构成的，其中有且仅有一个 `main()` 函数。

(2) C 程序没有行号，每条语句都是以“;”作为语句结束符。当一条语句没有结束时，一定不要加分号。

(3) C 语言源程序的执行总是由 `main()` 函数的第一个可执行语句开始，到 `main()` 函数的最后一个可执行的语句结束；而其他函数都是在 `main()` 函数开始执行以后，通过函数的调用才得以运行。

(4) C 语言本身没有输入/输出语句，执行输入/输出操作需要调用标准库函数来完成。引用这些标准库函数时，必须要用预处理命令将其头文件包含进来。预处理命令通常应放在源程序的最前面。

(5) 一个完整的函数由函数首部和函数体两部分组成。函数体即为函数首部下面的大括号 `{}` 内的部分，它一般包含两个部分：声明部分和执行部分。基本结构如下：

```

{
    声明部分：在这部分声明本函数所使用的变量
    执行部分：由若干条语句组成
}
  
```

在 C 程序中，声明部分在前，执行部分在后，这两部分的顺序不能颠倒，也不能有交叉。

(6) C 程序书写格式自由。一行内可以写几个语句，一个语句也可分写在多行上。

3. C 程序的书写规则

在学习 C 语言编程的入门阶段，养成良好的书写习惯相当重要，需要按照人们的约定和习惯来书写 C 语言程序，这样有助于提高程序的可读性。一个 C 语言程序如果书写不规范，虽然可以通过编译并能输出正确结果，但是阅读程序很困难，有时会因为书写不规范而引起误解，使程序可读性变差，甚至造成分析上的错误。所以，我们在书写程序时应遵循以下规则以养成良好的编程风格。

(1) C 程序格式常用锯齿形书写格式。用 `{}` 括起来的部分，通常表示了程序的某一层结构。`{}` 一般与该结构语句的第一个字母对齐，并单独占一行。每一组 `{}` 对齐。建议初学者

书写代码时,一行只写一条语句。

(2) 低一层次的语句或注释可比高一层次的语句或注释缩进若干格后书写。以便看起来更加清晰,增加程序的可读性。

(3) C 程序中可适当添加注释和空行,注释用来增强程序的可读性。注释部分的格式如下:

```
/* 注释内容 */
```

或

```
//注释内容
```

(4) 输入引号、括号等符号时应成对输入,然后再在中间添加内容。

(5) 变量名一般用小写字母表示,符号常量一般用大写字母表示。

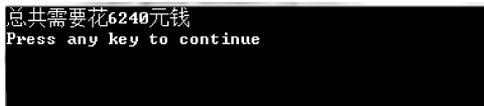
【知识拓展】

拓展任务 1.1: 庐山门票 120 元一张,某班级有 52 人,编写程序计算该班需要花多少元钱才能给每个人买一张门票,并在屏幕上显示计算结果。

参考代码如下:

```
1  #include "stdio.h"
2  int main ( )
3  {
4      /*声明部分*/
5      int count,sum;                //定义两个整型变量
6      /*执行部分*/
7      count = 52;                    //将 52 赋值给 count
8      sum = count * 120;             //将计算结果赋给 sum
9      printf("总共需要花 %d 元钱\n",sum); //显示计算结果
10     return 0;
11 }
```

程序运行结果如图 1.3 所示。



```
总共需要花6240元钱
Press any key to continue
```

图 1.3 拓展任务 1.1 程序运行结果

本任务着重说明一般函数体由两部分组成,一部分为用来声明变量,如代码的第 5 行;另一部分为进行相关操作的执行部分,即函数体中除声明之外的语句。

拓展任务 1.2: 现有两个班需要购买门票,软件班 48 人,计应班 58 人,通过调用自定义函数计算并在屏幕上输出两个班级各需要多少元钱?

参考代码如下:

```
1  #include "stdio.h"
2  int fun_sum(int x);           //声明一个整型函数,且有一个参数
3  int main ( )
4  {
5      printf("软件班总共需要花 %d 元钱\n", fun_sum(48));
6      printf("计应班总共需要花 %d 元钱\n", fun_sum(58));
7      return 0;
8  }
```

```
9   int fun_sum(int x)           //自定义函数 fun_sum
10  {
11     int sum;
12     sum = x * 120;
13     return sum;
14 }
```

程序运行结果如图 1.4 所示。



```
软件班总共需要花5760元钱
计应班总共需要花6960元钱
Press any key to continue
```

图 1.4 拓展任务 1.2 程序运行结果

本任务强调的是 C 语言中除主函数以外,程序员还可以自定义其他函数,这些函数可像 printf() 函数一样在程序中被调用。

【知识小结】

- (1) 函数是组成 C 语言结构化程序的最小模块。
- (2) C 程序是由一个或多个函数构成的,其中有且仅有一个 main() 函数。无论 main() 函数出现在哪个位置,程序都是从 main() 函数开始执行。
- (3) 每个 C 程序中的函数体一般包含两个部分,即声明部分和执行部分。在 C 程序中,声明部分在前,执行部分在后,这两部分的顺序不能颠倒,也不能有交叉。

1.3 C 程序的开发过程

C 语言是高级程序设计语言,不是计算机能识别和执行的由 0 和 1 组成的二进制指令,因此 C 语言程序是不能被计算机直接执行的,需要一种称为“编译程序”的软件将它翻译成二进制形式的“目标程序”,然后再将该目标程序与系统的函数库以及其他目标程序连接起来,形成“可执行程序”。本节主要介绍 C 程序的实现过程和开发环境。

本节学习目标:

- 了解 C 程序的开发过程。
- 掌握 Visual C++ 6.0 环境下调试 C 程序的方法和步骤。

【任务提出】

任务 1.3: 在 Visual C++ 6.0 环境下调试运行实现任务 1.2 的源代码。

【任务分析】

掌握 C 程序的实现过程和 Visual C++ 6.0 调试 C 程序的步骤方法是解决本任务的关键。Visual C++ 6.0 环境下调试运行程序主要分为创建工程、新建文件、源代码编辑和编译并调试运行程序等步骤。

【任务实现】

1. 创建工程

- (1) 启动 Visual C++ 6.0,进入如图 1.5 所示的主界面。

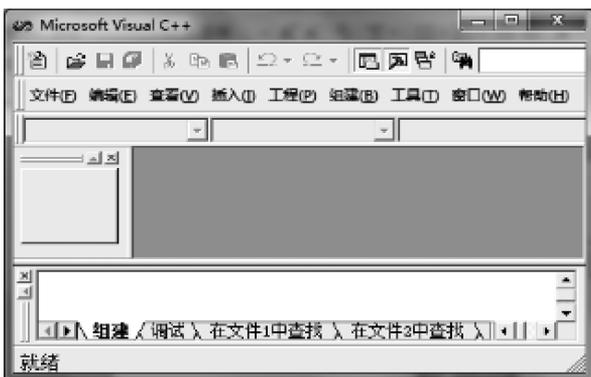


图 1.5 Visual C++ 6.0 的启动界面

(2) 选择“文件”菜单中的“新建”命令,新建一个 Win32 Console Application 控制台应用工程,并给新建工程命名,选择工程保存路径,单击“确定”按钮。如图 1.6 所示,本任务的工程名为 exam,保存路径为 E:\DEBUG。



图 1.6 “新建”对话框中的“工程”选项卡

(3) 单击“确定”按钮后,会弹出如图 1.7 所示的对话框,选择建立“一个空工程”,单击“完成”按钮,返回到 Visual C++ 6.0 的主窗口。



图 1.7 选择控制台程序类型

2. 创建源程序文件

在主窗口界面中选择“文件”菜单中的“新建”命令,打开如图 1.8 所示的“新建”对话框,选择 C++ Source File,在文件名中输入扩展名为.c 的源文件名。

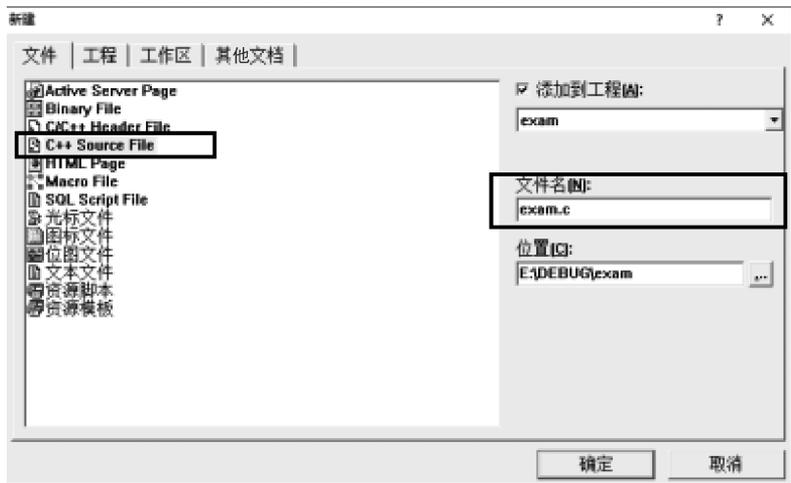


图 1.8 “新建”对话框

3. 编辑源程序

单击“确定”按钮后,即可在编辑区输入源程序,如图 1.9 所示,在编辑窗口中输入实现任务 1.2 的源代码并保存。

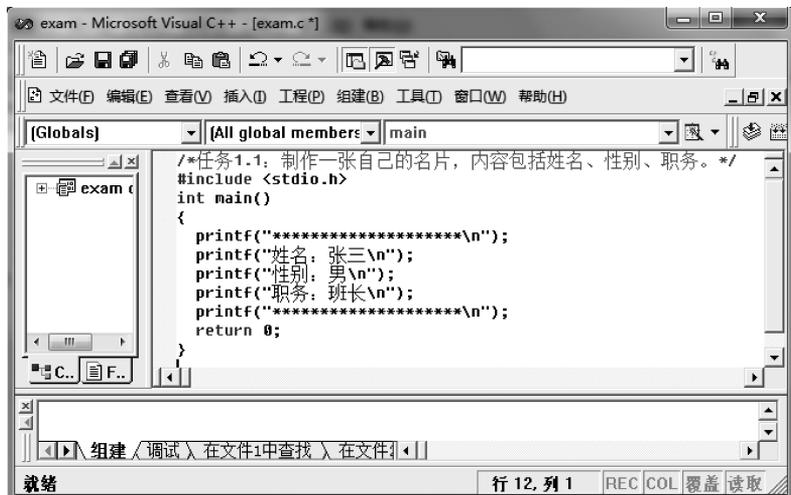


图 1.9 源程序编辑窗口

4. 编译

源代码编辑完成后,单击工具栏中的 Compile 按钮进行编译。如果没有错误,则会生成目标文件 exam.obj;如果出现错误,会在输出区提示错误内容,如图 1.10 所示。根据提示的错误信