

“十三五”高校计算机应用技术系列规划教材

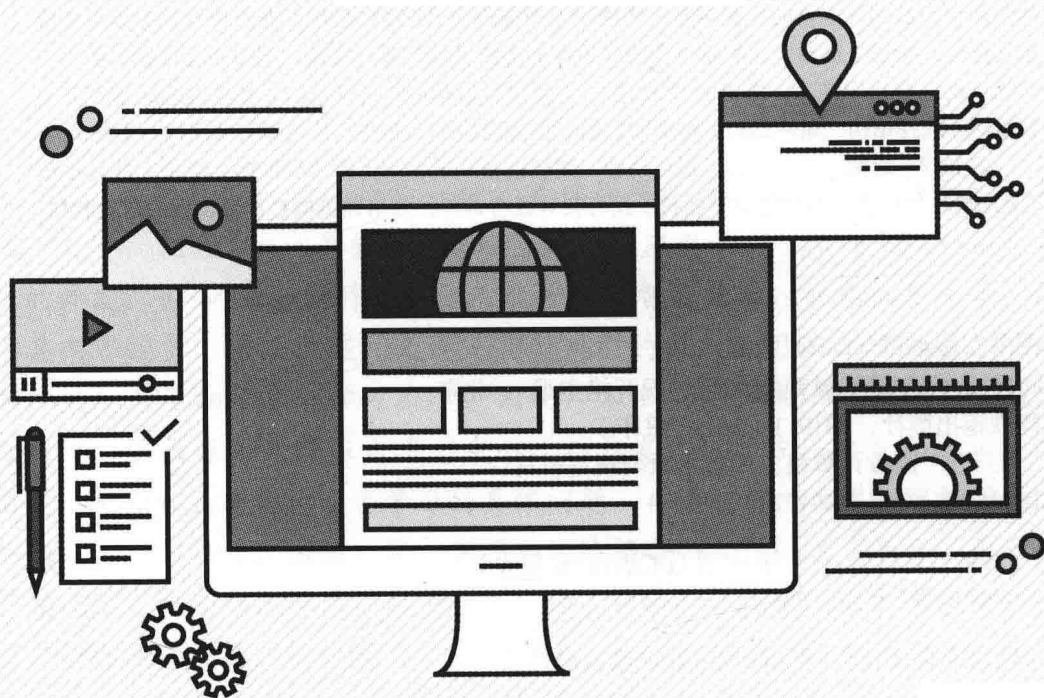


# 问答式C程序设计教程

赵旭宝 祝开艳 主编

中国铁道出版社有限公司  
CHINA RAILWAY PUBLISHING HOUSE CO., LTD.

“十三五”高校计算机应用技术系列规划教材



# 问答式C程序设计教程

赵旭宝 祝开艳 主编

中国铁道出版社有限公司  
CHINA RAILWAY PUBLISHING HOUSE CO., LTD.

## 内 容 简 介

C语言是目前使用比较广泛的一种结构化高级程序设计语言,由于其具有高效、灵活、运算能力强等特点,特别适合嵌入式系统和底层操作系统程序的开发。

本书从计算机“存储程序”工作原理出发,以启发式教学方法,采用一问一答的形式,由简入繁、循序渐进地将每个知识点逐步展开,引导学生在深入理解知识点的同时,进行程序开发。每个知识点都配有丰富的实践例题。在语法上严格遵守ANSI C标准。在程序设计方面强调模块化思想,特别注重知识点的分析和程序设计能力的训练,通过实践训练提高程序设计能力和知识的综合运用能力。本书共分11章,主要内容包括:C语言入门、数据类型、运算符与表达式、选择结构、循环结构、函数、数组、预处理命令、指针、结构体与共用体、文件。

本书教法新颖、深入浅出、通俗易懂、逻辑性强,适合作为高等学校理工类学生学习C语言的教材,也可作为初学者自学教材以及各类等级考试和社会培训机构C语言课程的培训教材。

### 图书在版编目(CIP)数据

问答式C程序设计教程/赵旭宝,祝开艳主编. —北京:  
中国铁道出版社, 2019.1 (2019.7重印)  
“十三五”高校计算机应用技术系列规划教材  
ISBN 978-7-113-25432-2

I. ①问… II. ①赵… ②祝… III. ①C语言-程序设计-  
高等学校-教材 IV. ①TP312.8

中国版本图书馆CIP数据核字(2019)第016272号

书 名: 问答式C程序设计教程  
作 者: 赵旭宝 祝开艳 主编

策 划: 汪 敏 读者热线: (010) 63550836  
责任编辑: 汪 敏 包 宁  
封面设计: 刘 颖  
责任校对: 张玉华  
责任印制: 郭向伟

出版发行: 中国铁道出版社有限公司(100054,北京市西城区右安门西街8号)

网 址: <http://www.tdpress.com/51eds/>

印 刷: 三河市航远印刷有限公司

版 次: 2019年1月第1版 2019年7月第2次印刷

开 本: 787 mm × 1 092 mm 1/16 印张: 19.75 字数: 479千

书 号: ISBN 978-7-113-25432-2

定 价: 49.80元

版权所有 侵权必究

凡购买铁道版图书,如有印制质量问题,请与本社教材图书营销部联系调换。电话:(010) 63550836

打击盗版举报电话:(010) 51873659



# 前言

C语言是目前比较流行的一种计算机程序设计基础语言，由于其具有高效、灵活、表达力与运算能力强等特点，普遍应用于底层操作系统和嵌入式系统的开发，例如Linux操作系统和智能手机、智能电器、智能机器人等。但由于其内容较多，语法严谨，尤其是指针类型的引入，用法比较抽象，难于理解，更加深了初学者的学习难度。因此，本书从计算机“存储程序”工作原理出发，强调程序开发以内存为中心，以启发式教学方法为主导，从初学者角度，把知识点转化为读者学习的具体问题，再通过进一步详细解答问题讲解知识点，一问一答，由简入繁，循序渐进地将每个知识点逐步展开，最后形成一套比较完整的知识体系。本书在引导读者理解知识点的同时，更注重培养读者分析问题、解决问题的能力，激发读者自主编程的热情，提高学习兴趣。在程序设计方面强调模块化设计思想，引导读者对程序的设计按功能进行模块分解，分而治之，分工协作，并以“服务外包”思想解读模块之间的调用过程。教材中每个知识点都配有生动、丰富的实践例题，通过实践例题的训练提高读者程序设计开发能力和知识综合运用能力。

本书共11章，每章的开始部分都介绍一些基本概念和原理，让初学者知道本章内容适用于解决什么问题及在什么情况下使用。然后在后续各节中，采用一问一答的形式教会读者如何使用本章的知识点。第1章主要以问答的方式介绍了C语言的特点、C语言程序的开发步骤，通过讲解读者可了解C语言程序的结构和执行原理，开发出属于自己的第一个C语言程序。第2章介绍了基本数据类型和各种输入和输出函数。第3章介绍了各种运算符和表达式。第4章介绍了选择结构，讲解时注重与实际问题结合，选择贴近生活的实例，让读者理解各种选择结构如何实现对程序流程的控制。第5章介绍了循环结构，重点介绍了循环结构的运算特点及循环结构在程序开发中的运用，提高读者使用循环结构解决实际问题的能力。第6章是本书的重点之一，介绍了函数的定义、调用、返回和程序中变量的作用域和生命周期。强调程序开发模块化的设计原

则，并通过“服务外包”思想，结合生动的实例，一步一步讲解函数的调用过程。使读者对函数的使用有更深入的理解，提高读者模块化的程序设计能力。第7章是本书的另一个重点，介绍了数组元素的存储和地址的分布特点，详细介绍了数组与循环结合对数据进行批处理的使用方法，同时结合实践例题讲解了一些实用的算法。第8章介绍了预处理命令。第9章是本书的难点——指针，通过启发式提问，介绍了指针的工作原理和指针操作变量、指针操作数组、指针操作函数、指针操作字符串的使用过程。第10章介绍了结构体、共用体与枚举的定义及它们在处理复杂问题时的使用方法。第11章介绍了文件操作的基本原理和步骤，详细讲解了文件读/写函数和文件读/写控制函数的使用。

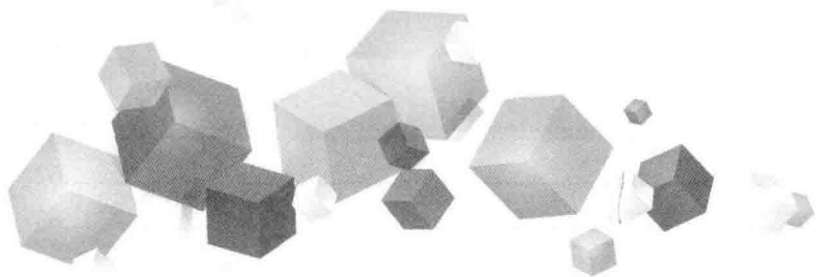
学习C语言编程，学习语法是基础，学习解决问题的算法是关键。因此本书在讲授语法的同时还详细地介绍了一些实用的算法（如打擂台算法、穷举算法、排序算法和开关控制算法等），并结合一些典型例题引导学生进行算法设计。同时也在算法设计过程中，深入理解C语言的语法知识。

本书由大连交通大学软件学院赵旭宝、大连海洋大学信息工程学院祝开艳主编。其中，第1~3章由祝开艳编写，第4~11章由赵旭宝编写。本书是作者多年教学和培训成果的结晶，采用启发式教授方法，特别适合作为C语言初学者入门级教材。本书在编写过程中参考了大量的著作和教材，在此对其作者深表感谢。

由于作者水平有限，书中难免存在疏漏和不足之处，敬请广大读者不吝指正，不胜感谢。

编者

2018年10月



# 目 录

<b>第1章 C语言入门</b> .....	1
1.1 C语言历史背景.....	1
1.2 一个简单的C语言程序.....	3
1.3 C语言程序的开发与运行步骤.....	6
1.4 如何学习C语言.....	14
1.5 结构化程序设计方法简介.....	15
小 结.....	16
习 题.....	16
<b>第2章 数据类型</b> .....	18
2.1 数据类型介绍.....	18
2.2 基本数据类型.....	19
2.3 数据类型的转换.....	31
2.4 数据的输入/输出.....	33
2.5 单个字符的输入/输出.....	34
2.6 有格式数据的输入/输出.....	35
2.7 字符串的输入/输出.....	42
小 结.....	42
习 题.....	43
<b>第3章 运算符与表达式</b> .....	47
3.1 运算符介绍.....	47
3.2 算术运算符和算术表达式.....	48
3.3 关系运算符和关系表达式.....	51
3.4 逻辑运算符和逻辑表达式.....	52
3.5 赋值运算符和赋值表达式.....	56
3.6 位运算符.....	58

3.7 其他运算符.....	60
小 结.....	62
习 题.....	63
<b>第4章 选择结构.....</b>	<b>68</b>
4.1 选择结构概述.....	68
4.2 简单if选择结构.....	69
4.3 if else选择结构.....	70
4.4 if...else if...else选择结构.....	72
4.5 选择结构的嵌套.....	74
4.6 switch case多分支选择结构.....	76
4.7 switch case嵌套结构.....	79
4.8 选择结构在程序开发中的应用.....	81
小 结.....	83
习 题.....	83
<b>第5章 循环结构.....</b>	<b>89</b>
5.1 循环结构概述.....	89
5.2 while循环结构.....	90
5.3 do while循环结构.....	94
5.4 for循环结构.....	96
5.5 break和continue关键字.....	99
5.6 各种循环结构嵌套.....	101
5.7 循环结构在程序开发中的应用.....	104
小 结.....	108
习 题.....	108
<b>第6章 函数.....</b>	<b>116</b>
6.1 函数的结构.....	116
6.2 函数声明与函数原型.....	127
6.3 C程序的基本结构.....	128
6.4 函数的递归.....	130
6.5 全局变量和局部变量.....	133
6.6 存储类别.....	136
小 结.....	140
习 题.....	140

<b>第7章 数组</b> .....	147
7.1 数组的引入.....	147
7.2 一维数组的定义与使用 .....	148
7.3 二维数组的定义与使用 .....	158
7.4 字符数组的定义与使用 .....	165
7.5 数组在程序开发中的应用 .....	174
小 结.....	176
习 题.....	177
<b>第8章 预处理命令</b> .....	185
8.1 预处理命令的引入 .....	185
8.2 宏定义.....	186
8.3 文件包含.....	188
8.4 条件编译.....	192
小 结.....	195
习 题.....	195
<b>第9章 指针</b> .....	198
9.1 内存地址的介绍 .....	198
9.2 指针的引入.....	199
9.3 指向单个变量的指针变量 .....	200
9.4 指针与数组元素的关系 .....	205
9.5 指向一维数组的指针变量 .....	213
9.6 指向字符串的指针变量 .....	216
9.7 指针数组.....	221
9.8 指针与函数.....	226
9.9 指针在程序开发中的应用 .....	228
小 结.....	230
习 题.....	231
<b>第10章 结构体与共用体</b> .....	240
10.1 结构体类型的引入 .....	240
10.2 结构体类型的定义 .....	241
10.3 指向结构体的指针变量 .....	249
10.4 结构体在程序开发中的应用 .....	252

10.5	共用体类型.....	259
10.6	枚举类型.....	262
10.7	typedef类型重命名.....	265
小 结	.....	266
习 题	.....	266
<b>第11章</b>	<b>文件.....</b>	<b>274</b>
11.1	文件介绍.....	274
11.2	文件顺序读/写操作.....	276
11.3	文件读/写控制.....	285
11.4	文件在程序开发中的应用.....	287
小 结	.....	289
习 题	.....	290
<b>附录A</b>	<b>标准ASCII表.....</b>	<b>295</b>
<b>附录B</b>	<b>运算符表.....</b>	<b>296</b>
<b>附录C</b>	<b>习题参考答案.....</b>	<b>297</b>
<b>参考文献</b>	.....	<b>307</b>



# 第1章

## C语言入门



### 主要内容

- ◎ C语言介绍、一个简单的C语言程序
- ◎ C语言程序的开发与运行过程
- ◎ 如何学习C语言



### 重点与难点

- ◎ 重点：C语言程序的开发与运行过程
- ◎ 难点：开发环境的使用及编写一个可执行的C语言程序

## 1.1 C语言历史背景



### 实问1：你知道C语言是谁发明的吗

C语言是1973年，由美国贝尔实验室的丹尼斯·里奇（Dennis M. Ritchie）在B语言的基础上设计出的一种新语言。丹尼斯·里奇被称为C语言之父。1978年Brian W. Kernighan和Dennis M. Ritchie一起出版了名著*The C Programming Language*，使C语言逐渐成为世界上广泛使用的高级程序设计语言之一。

目前，C语言广泛用于嵌入式系统微处理器程序以及底层操作系统程序的开发。例如：Linux操作系统和智能机器人、智能汽车、智能家用电器、某些航空航天设备等底层的处理程序。



### 实问2：为什么C语言能成为世界上使用最广泛的编程语言

C语言之所以能成为世界上使用最广泛的编程语言，原因在于它是一种高级程序设计语言，并具有极强的表现能力和硬件处理能力。主要体现在如下几方面：

#### 1. C语言是结构化的高级程序设计语言

高级程序设计语言的特征是它比较接近人的自然语言（相对于汇编语言而言），用其书

写程序代码更符合人的思维逻辑，容易理解，可读性好。比如：要计算a与b的和，可以写成 $a+b$ 。但早期的低级程序设计语言则不能这样写（如汇编语言）。同时，高级程序设计语言易于灵活使用各种选择结构和循环结构控制程序的流向，实现根据不同条件，执行不同的“代码块”。这使得程序完全结构化，就像堆积木一样，可以根据不同的需求，选择不同形状的积木模块，搭出不同的结构。如图1-1所示，可以在C块的后面选择H、I、N、A块，组成“CHINA”单词，也可以根据其他的条件选择不同的方块，组成其他单词。这种结构化的程序，更便于调试和维护代码。

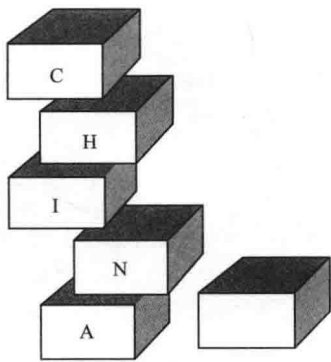


图1-1 结构化模块图

## 2. C语言类型丰富、表现能力强

C语言具有丰富的数据类型和运算符。C语言提供了整型（int和long）、浮点型（float和double）、字符型（char）、结构体类型（struct）等多种数据类型，使得程序能有效地表示和处理各种复杂类型的数据。同时，C语言还提供了算术、关系、赋值、逻辑等多种运算符，从而使C语言表达式类型更加灵活、多样化。

## 3. C语言能够直接操作硬件

在C语言诞生之前，系统软件主要是用汇编语言编写。汇编语言的优点是能直接访问硬件的存储器，程序运行速度快，但缺点是汇编语言依赖于硬件平台，其可读性差，程序开发效率低。在C语言中，由于引入了一种称为指针的数据类型，利用指针变量能够直接与硬件的内存地址打交道，实现了汇编语言的大部分功能。所以说，C语言是既具有汇编语言，又具有高级语言特性的一种高级程序设计语言。指针也被称为是C语言的精髓。

### 实训3：为什么指针是C语言的精髓

在C语言中，由于引入了指针类型，利用指针变量可以实现对硬件系统内存地址（内存地址就是内存单元的门牌号）的直接访问，提高了对硬件系统的访问效率。从而编写出更高效的程序。因此，把指针称为C语言的精髓。

指针就是地址，按地址方式操作内存数据将大大提高程序执行的效率。对内存中数据的存取操作就好比邮递员送信。如果邮递员已知信件投递的具体地址（即省市区街道门牌号），他将会按指定的地址投递信件，投递效率最高。如果邮递员仅知道收件人的姓名，则投递的速度相对要慢很多。

C语言是高级的结构化程序设计语言，既适合编写系统软件，又适合编写应用软件。因此，学好指针对今后的软件开发尤为重要。但要注意，凡是有利必有弊，由于C语言的指针操作没有保护措施，因此也会给开发的系统软件和应用软件带来很多安全隐患。（指针将在第9章讲述）。

### 实践A：哪些系统软件是用C语言编写的

答：Linux系统；微软系列产品；手机和智能设备中的某些程序。

## 1.2 一个简单的C语言程序

### 实问1：一个简单的C语言程序是什么样

一个简单的C语言程序是仅由一个被称为“main()”的函数构成。请看实践B。

### 实践B：给出下面程序的输出结果

Practice\_B程序代码如下：

```
#include <stdio.h>    // 这行代码称为“头文件包含”
int main()           // 这行代码称为“函数头部”
{
    // 这段“{”和“}”中的代码称为函数体
    printf("C语言的发明人是丹尼斯·里奇."); // 调用printf()函数，将字符串输出到显示器上
    return 0;
}
```

运行结果：

C语言的发明人是丹尼斯·里奇。

### 实问2：能详细解读一下实践B中每行代码的含义吗

实践B中的程序代码由“头文件”和“main()函数”两部分组成。

(1) #include <stdio.h>称为头文件包含。使用头文件的原因在于程序中用到了printf()函数，这个函数不是定义在本程序中，它是被定义在头文件stdio.h中。因此当使用printf()函数时，就需要把它所在的头文件包含进来（printf()函数的详细用法在第2章讲述）。“include”是包含的意思。将printf()函数所在的头文件包含在本程序中，告诉我们，程序中用到的任何函数和变量，必须做到先定义，再使用。

(2) 程序中“int main()”和“{”和“}”括起来的这段代码称为“函数”。

(3) main()函数是一个特殊的函数。任何可运行的C程序都必须包含main()函数，否则程序无法运行。在main()函数体内调用了printf()函数。调用printf()函数的作用是让它帮忙在计算机屏幕上输出一句话。在main()函数体内使用了另一个函数printf()，这种做法称作“函数调用”。函数调用的过程相当于一种功能上的“服务外包”，即main()函数把自己不能或不想完成的任务外包给其他函数来做。可以想象一下，今后读者在开发程序时是否也可以这样做，把自己不能或不想做的任务外包给其他函数来做。

(4) return 0;这条语句的作用是将“0”值返回给使用main()函数的人。main()是个非常特殊的函数，当程序运行时，操作系统负责调用main()函数。因此该值将直接返回给操作系统。一般main()函数中的返回值为0或1，返回0表示函数正常退出，如果返回1，则表示函数异常退出。

(5) 实践B的执行过程是：操作系统调用了main()函数，然后main()函数调用了printf()函数，实现了将指定的字符串输出到显示器上。

 提示:

在编写C语言程序时, 如果想让程序能被计算机运行, 就必须要编写一个main()函数(有且只有一个), main()函数是一个非常特殊的函数。如果编写了多个函数, 那么函数之间是调用与被调用的关系。

 实训3: C语言程序为什么一定要包含main()函数

C语言是一种面向过程的高级程序设计语言。这类语言的特点是以函数作为程序的组织结构。当C程序被运行时, 编译系统总要选择—个函数作为程序执行的起点。main()函数是被各编译系统公认的程序运行起点, 也就是程序的入口。只是不同程序设计语言中main()函数的定义形式有所不同。每次运行程序时, 编译系统都会寻找这个起点是否存在, 如果起点存在, 程序能正常运行, 否则程序将无法运行。

因此, 每个可运行的C语言程序必须包含一个main()函数。main()函数又称“主函数”。最新C99标准中, 主函数有两种定义结构:

## 1. 无参数的主函数结构

```
int main()      // 无参数形式
{ ... ..
  return 0;
}
```

## 2. 带参数的主函数定义结构

```
int main(int argc, char *argv[]) // main() 函数中包含两个形式参数
{ ... ..
  return 0;
}
```

其中, 所带的两个参数是指程序运行时的命令行参数。argc变量中存放了包括程序名在内的所有命令行参数的个数; argv[]数组中存放了多个字符指针, 分别用于指向包括程序名在内的所有命令行参数。关于这部分内容将在第9章中讲述, 此处仅了解一下即可。

 实践C: 编写一个简单的可运行的C语言程序, 输出“我的梦想是学好C语言。”

设计思路: 本着编写“一个可运行程序”的想法, 程序中必须编写一个main()函数。然后在main()函数中, 编写想要实现的功能。由于要输出一个指定的字符串, 但main()函数本身不具备输出功能, 所以需要调用printf()函数实现字符串的输出(这个调用过程可理解为main()函数将自己不能做的输出任务外包给了printf()函数来完成)。printf()函数是系统已经定义好的函数, 放在了stdio.h头文件中, 因此程序的开始部分要包含stdio.h头文件。

Practice\_C程序代码如下:

```
#include <stdio.h> // 由于用到了printf()函数, 所以包含stdio.h头文件
int main()
{
  printf("我的梦想是学好C语言."); // 调用printf()函数, 完成了字符串的输出
}
```

```
return 0;
}
```

#### 实问4：函数是如何构成的

事实上，函数是一个独立的能实现某种特定功能的程序模块（如main()和printf()函数），它的组成结构与自然界的生物类似，都是由头部和身体构成。函数头部又由函数类型（int）、函数名字（main）和函数参数三部分构成。函数体是由“{”和“}”括起来的多条语句构成。

函数的构成：任何一个函数都是由函数首部和函数体组成。

- (1) 函数首部指定了函数类型（又称函数返回值类型）、函数名、函数参数三部分。
- (2) 函数体从“{”开始，到“}”结束，括号不可省略。
- (3) 函数体内有各种语句。

函数只有存在了，才能被别人使用。即函数也要先定义再使用。

例如：main()函数的结构如下。

```

函数返回值类型  函数名  函数形式参数
      |           |           |
      v           v           v
int main (int argc, char *argv[])
{
    int k;
    k=3+2;
    printf("%d\n", k);
    return 0;
}
      } 函数体
  
```

#### 实问5：能否对C语言程序做个总结

无论C语言程序多么庞大复杂，都是由一个或多个函数构成。每个函数都对完成一个特定的功能。各个函数之间是调用与被调用的关系。显然，要学会编程，就要学会编写函数。

程序构成规则：

- (1) 函数是C语言程序的基本单位，程序中可以包含一个或多个函数。
- (2) 程序想要运行，必须有main()函数，且仅有一个main()函数，main()函数是程序的入口。
- (3) main()函数不一定是代码中的第一个函数。程序运行选择入口时与main()函数在代码中的位置无关。程序运行时，总是从main()函数开始。当main()函数执行完毕时，程序运行结束。
- (4) 函数由多条语句构成；每条语句用“;”结束。
- (5) 函数分为用户自定义函数和系统库函数。

## 1.3 C语言程序的开发与运行步骤

### 实训1: 编写C语言程序的步骤是什么

开发C语言程序需要完成如下步骤:

#### 第一步: 编写程序代码 (edit)。

为了能让计算机运行出预想的结果。编程人员需要按照人类解决实际问题的过程, 一步步编写相应的程序代码 (即程序代码的书写一定要有先后顺序), 并最终将程序代码保存成扩展名为.c的文件。书写程序代码时要注意每行代码都要符合C语言语法规则。

#### 第二步: 编译程序 (compile)。

C语言是面向过程的一种高级计算机编程语言, 高级语言所编写的代码比较符合人的思维逻辑, 但是它不能被计算机的CPU识别和运行。计算机只识别二进制的“0”和“1”。如果想让计算机能运行C语言代码, 就必须通过编译器进行编译, 编译器像翻译官一样, 能将C语言写的程序代码解析成被计算机CPU识别的二进制指令代码。这种二进制代码又称“目标程序 (扩展名为.obj或.o, 视编译环境而定)”。

#### 第三步: 连接程序 (link)。

将编译后的目标程序代码 (.obj或.o文件) 和系统中其他目标程序代码 (.obj或.o文件) 进行连接, 并加入一些硬件启动信息, 最终形成可执行的程序 (扩展名为.exe)。

#### 第四步: 运行程序 (run)。

开发步骤如图1-2所示。

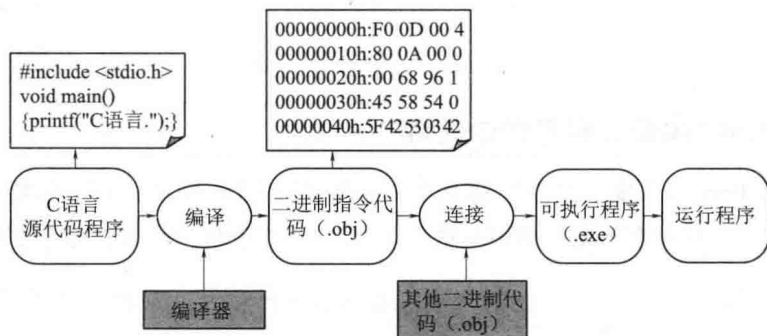


图1-2 C程序开发过程

### 实训D: 编写程序计算并输出两个整数相加的和

设计思路: 为了保证程序能够运行, 首先编写main()函数, 在main()函数体内完成两个整数相加求和, 最后再使用printf()函数将和值输出。使用printf()函数时需要包含stdio.h头文件。

另外, 编写main()主函数时还需要定义两个整型变量 (int a, b), 目的是用来存储给定的两个整数值。

Practice\_D程序代码如下:

```

#include <stdio.h> // 由于用到了 printf() 函数, 所以要包含 stdio.h 头文件
int main()
{

```

```

int a=1;           // 定义 int 整型变量 a, 用于保存整数 1
int b=2;           // 定义 int 整型变量 b, 用于保存整数 2
printf("%d",a+b); // 调用 printf() 函数, 将 a+b 的和值输出到显示器
return 0;
}

```

说明: int是一种数据类型,使用该类型的目的是决定程序中用到的数值1和2在内存中占据多大的存储空间(占4字节);a、b称为变量标识符(数据类型和标识符将在第2章中讲述,这里只需了解即可)。

### 实践E: 请在实践D的基础上, 用函数调用的方式实现两个整数相加

设计思路: 实践D是main()函数负责完成两个整数相加,然后调用printf()函数输出和值。此时,假设main()函数不会计算两个整数的加法,那该怎么办?因此我们可仿照实践C,将主函数不能或不想做的任务外包给其他函数(如add()函数)来做,然后让add()函数把计算后的结果告诉main()函数,最后main()函数使用printf()函数将和值输出,这个过程就是函数的调用过程。但add()函数并不存在,需要自己编写。(关于函数定义的详细内容,详见本书第6章)。

#### add()函数的定义:

(1) 函数的返回类型: int(表示函数需要将计算后的结果返回给调用自己的函数,两数相加的结果为int型,故返回类型设定为int)。

(2) 函数名: add。

(3) 函数参数: int a,int b(计算两个数相加,必须事先已知这两个数,因此形参有两个)。

(4) 函数体: 对a和b两个数进行求和计算,并使用return关键字将计算结果返回给main()函数。

Practice\_E程序代码如下:

```

#include <stdio.h>
int add(int a, int b) // 函数的定义过程
{
    int c;           // 定义 int 整型变量 c
    c=a+b;          // 将两个数相加, 并将结果赋值给 c
    return c;       // return 关键字的作用是将结果 c 返回给 main() 函数
}
int main()
{
    int a=3,b=4,k;
    k=add(a,b);    // main() 函数调用 add() 函数, 完成 a、b 两个整数相加的外包任务,
                  // add() 函数将返回的结果交给 main() 函数中的变量 k
    printf("%d\n", k); // 输出变量 k 的值
    return 0;
}

```

说明: 计算机中的等号“=”为赋值运算符,它与数学中的等号不同。计算机中的等号是将右侧的值赋给左侧的变量。如语句c=a+b;是将二者相加的结果赋给变量c。初学者一定要理解计算机中“等号”的用法(从右往左赋值)。另外,读者可以对比一下,实践D和实践E的

两种编程方式，哪种更好一些？关于两种编程方式的对比详见本书6.1节。

## 实问2：编译器的作用是什么

编译器是介于用户和计算机之间的一个软件系统，它就像一名“翻译官”，将某种语言所编写的源代码（如C语言）转换成能够被计算机CPU识别并运行的指令代码（0或1二进制码），如图1-3所示。在转换过程中，编译器要进行源代码的语法规则检查和最终目标代码连接。因此，编写的C语言程序代码必须符合C语言语法规则。所谓指令代码就是一组能被CPU执行的二进制代码，这种代码又称机器码。

但要注意，编译器并不是完全统一的。如果C语言编译系统的版本不同，则对应的语法规则会略有不同，程序的执行结果也会有差异。所以实际工作中一定要格外注意，最好先确定开发环境，然后再进行程序开发工作。

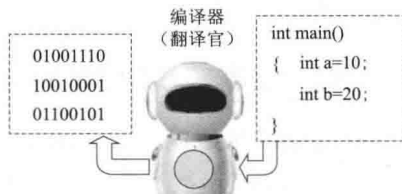


图1-3 编译器工作原理示意图

## 实问3：目前比较流行的C语言开发环境有哪几种

目前，常用的C语言开发环境有如下几种：

- (1) Borland Turbo C 或称 TC。
- (2) Microsoft VC++ 6.0和Visual Studio 或称VC和VS。
- (3) Dev-C++。
- (4) Code::Blocks。

总的来说，C语言开发环境可以分为C和C++两大类，其中C++是C的超集，均向下支持C。TC系列是Borland公司的产品。TC 2.0是一个体积小、简单易学的C语言集成开发环境，是早期最经典的C语言开发环境，但它不支持鼠标操作。VC和VS系列是Microsoft公司的产品。它们都是Windows版集成开发环境，具有可视化的编程界面，是目前主流的C/C++语言开发平台，但VC和VS安装包比较大，安装流程复杂。Dev-C++是Bloodshed开发的基于Windows环境下的适合初学者使用的C/C++集成开发环境，但目前原公司已经停止软件的更新。Code::Blocks是一个开源的全功能的跨平台C/C++集成开发环境。Code::Blocks由纯粹的C++语言开发而成，并使用了著名的图形界面库wxWidgets，捆绑了MinGW编译器。

对于C语言初学者，开发环境的选择并不重要，它们都可以完成基本的C语言编译，不过在面向等级和升学考试中，还是要根据考试的要求，选择合适的开发环境。

## 实问4：本书采用哪种开发环境且如何使用

本书中所有代码均采用Code::Blocks集成开发环境进行编译调试。下面对Code::Blocks开发环境的使用进行详细介绍。

### 1. 下载与安装

读者可自行在官方网站：<http://www.codeblocks.org/downloads/26>上下载适合自己的Code::Blocks版本。其中codeblocks-17.12mingw-setup.exe版本是适用于Windows操作系统的自带GCC编译器的版本，也是大家常用的版本。下载后单击codeblocks-17.12mingw-setup.exe可执