C语言程序设计
实用指南

王继贤　编

清华大学出版社

目　录

第一章 概 论

本章主要介绍C语言的发展历史、特点、符号以及程序结构和程序格式等问题。

§1.1 C语言的发展史

计算机技术从40年代中期问世以来，在短短的几十年时间里，其发展速度是十分惊人的。软硬件更新的周期都很短，每一个创新，都刻划着时代的特征，推动着计算机技术的进步。C语言的出现以及相继问世的UNIX操作系统，专家们一致认为是具有划时代意义的。

C语言的发展有一个不断演变、扩充及改进的过程，如图1-1所示。C语言的前身是复合程序设计语言CPL[1,2,3](Combined Programming Language)，它是仿照ALGOL 60而设计出来的语言，由于CPL庞大而又复杂，不便于使用，推广受到了很大的限制。于是Martin Richard改进设计了一个BCPL语言(BASIC CPL)，但BCPL在实践中仍不十分方便，1970年Ken Thompson再一次对BCPL加以改进，提出了一种B语言。接着D. M. Ritchie在B语言的基础上，再作了若干充实、改进，于1972至1973年间，提出了一种C语言。

在C语言发展的每一个阶段，都有它的优点和不足，科学家们总是在吸收前人研究成果的基础上，补充新的功能，改进它的不足。例如B语言就缺乏丰富的数据类型，而D. M. Ritchie不仅改进了这一点，而且还增加了语言的控制能力，对整个B语言进行了

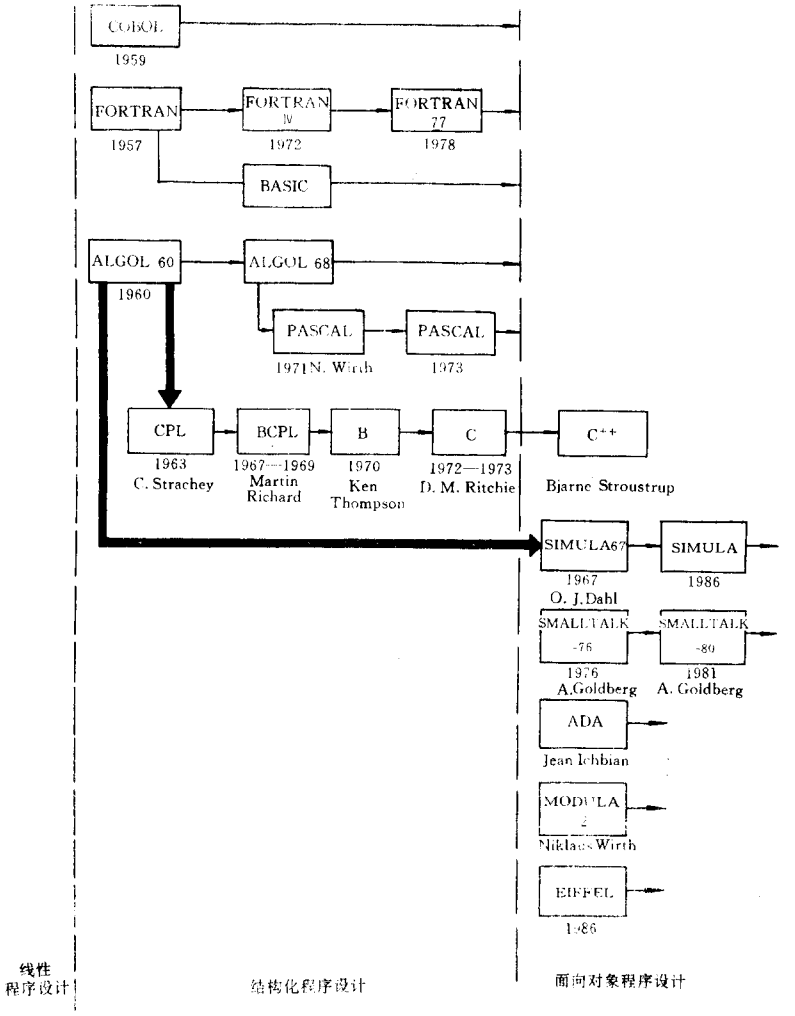

图 1-1　程序设计语言及程序设计的发展

总体设计和清理，从而形成了一个较为理想的编程工具——C 语言。

Ken Thompson 和 D. M. Ritchie 于 1973 年利用新的版本 C 语言，重新设计了 UNIX，进而扩大了 UNIX 的功能。所以 C 语言的发展和它在编写UNIX操作系统中的应用，是紧密相关的。UNIX 的发展之所以有今天这样的扩展面，正是因为有了 C 语言这种好的编程语言。

§1.2 C 语言的特点

C 语言在它短短的十多年发展中，能够成为当今计算机界公认的一种通用的、好的程序设计语言，究竟有什么特点呢？

1. C 语言是一种结构化的程序设计语言。如图 1-1 所示，程序设计大致可以划分为三个阶段：线性程序设计、结构化程序设计和面向对象的程序设计[3,5]。

所谓线性程序设计，是直接用机器语言和符号（机器代码）来编写程序，这些程序仅能完成简单的任务，而且是线性的、顺序的系列。

结构化程序设计就完全不同了，它能针对各种复杂的大型应用问题进行设计。按照系统论的观点，一个大型的复杂问题（系统），可以分成若干小系统，每个小系统又可分成若干进程，或者需要解决的问题（C 语言将这些进程称为一个函数）。当程序运行时，把这些不同的函数（或者叫做模块）组装起来就可以了。所以函数是 C 语言中结构化程序设计的基本单元。

结构化程序设计技术，逻辑思维清晰、容易编制，也易于理解和维护。今天几乎所有的计算机语言都支持这种技术，C 语言以其结构简洁、逻辑清晰而深受广大程序设计人员的喜爱。

2. C 语言独立于计算机，对机器的依赖性很小。今天无论微

型机(PC)机，工作站(WS)，大、中、小型机，乃至巨型机，都配置有C语言编译系统，这是C语言能够获得广泛应用的又一个重要原因。

C语言既有面向结构化程序设计的许多高级语言所具有的功能，也有面向硬件，具有汇编语言的某些功能。例如：它可以直接处理字符、位加工，地址及指针运算等，这些通常需要由汇编语言来完成。这一重要特征可以使C语言大大提高运算效率[3,5]。C语言在硬件、软件中的地位示于图1-2。图中椭圆的大小，标志着该语言的地位及作用范围的相对大小。

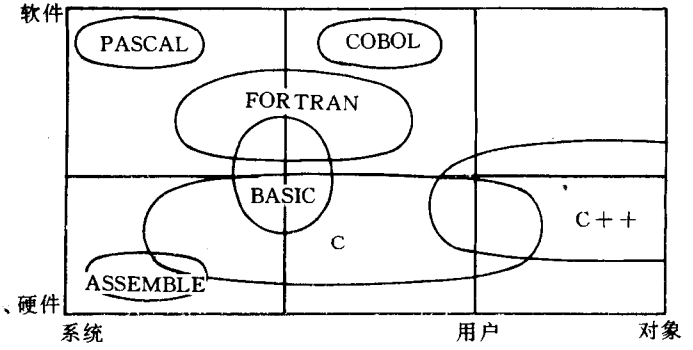

图1-2 C语言在硬件、软件中的地位

3. C语言具有丰富的运算功能。C语言除了具有一般高级语言所具有的四则运算：加(+)、减(-)、乘(×)、除(/)；逻辑运算：逻辑与(&&)、逻辑或(||)、逻辑非(!)，位运算(&，|，≫，≪)以及关系运算而外，还有其它高级语言所没有的单项运算x++和Y--以及各种复合运算(+=，-=，*=，/=)，指针运算，条件运算等等。

4. C语言具有丰富的数据类型。一种语言一方面要对数据有较强的处理能力，即具有丰富的运算功能，另一方面也要对处理对

象——数据，有较强的描述(定义)能力。C 语言除了同其它高级语言一样，具有整型、实型、字符型、数组型等而外，更有其它高级语言所没有的类型，如结构型、联合型、指针型等。

5. C 语言预处理程序的功能，独树一帜，别具一格，是其它高级语言所没有的。C 语言利用它独有的 #define，#include，及 #if，#else 等语句，就可以进行字符串，参数的各种宏定义，文件包含，条件编译等各种预处理，从而大大提高了软件开发的效率，提高了软件开发的质量。

6. C 语言是面向对象程序设计的基础。C 语言中几乎所有的功能和特点，都已被 C++ 吸引，而 C++ 经过改进和扩充，已发展成为面向对象的程序设计语言[5]。面向对象的程序设计是软件设计技术今后发展的方向(参看图 1-1)。所以我们学习掌握了 C 语言，也就为跨入程序设计的第三阶段，为今后的发展，提前做好了准备。正是由于 C 语言具有这种独有的特点，才使它具有广泛的应用领域和强大的生命力。

§1.3 C 语言的符号

任何一种语言都有它自己的一些规定的语言符号，利用这些符号就可以定义语法，并组成不同的语句。

C 语言的基本符号有 4 类 103 种：

1. 字母：英文大小写字母各 26 个，共 52 个。

2. 阿拉伯数字：0—9，共 10 个。

3. 特殊字符：

(1) 算术运算符：加(+)，减(−)，乘(*)，除(/)，取模(%)；

(2) 自增自减运算符：自增加 1(++)，自减减 1(−−)；

(3) 关系运算符号：小于(<)，小于等于(<=)，大于(>)，大于等于(>=)，不等于(!=)，全等比较(==)；

(4) 逻辑运算符：逻辑与(&&)，逻辑或(||)，逻辑非(!)；

(5) 位逻辑运算符：与(&)，或(|)，加(∧)，求反(～)；

(6) 位移运算符：左移(≪)，右移(≫)；

(7) 赋值运算符：=；

(8) 组合运算符：加等(+=)，减等(-=)，乘等(*=)，除等(/=)，取模等(%=)，左移等(≪=)，右移等(≫=)；与等(&=)，或等(|=)，按位加等(∧=)；

(9) 逗号运算符：，；

(10) 条件运算符：？：；

(11) 地址运算符：取地址(&)，取内容(*)；

(12) 其它运算符：不同C语言版本，各有不同。如数组下标定界[ ]，结构、联合体取成员：函数定义及引用( )，结构指针取成员→，等等。

4. 下线符：_。

§1.4 C语言的程序结构

下面通过几个简单的例子来说明C语言的程序结构及其最基本的组成部分。

例1-1 要在一个终端上显示一句话：

C语言是非常重要的。即：C language is very important!

要用C语言来实现这个任务，应该这样编写：

```
main()
  {
    printf("C language is very important! \n");
  }
```

这是一个非常简单的C语言程序，如果将它键入计算机，并经过编译、执行，就可以在终端上看到该程序的输出结果是：

C language is very important!

现在我们根据C语言的一些规定，来对例1-1作些解释。

程序的第一行：main()表示该程序的名字为main。main是C语言编译系统用的专有名字，它表示该程序是从这里开始。按照惯例：C语言将程序称为函数，所以本书也采用函数一词。在FORTRAN，BASIC等语言中有主程序、子程序的说法，而在C语言中，为便于理解以及叙述的方便，我们将main()称为主函数，其他函数，则称为子函数。如果有几个函数时，C语言编译系统就从带有main这个名字的主函数开始执行。当然例1-1只有一个函数，它就是main函数。main后边括号中，可以有参数，也可以不写参数(用隐含值)，但这对圆括号(　)必须要有，不能省略。

{　}是一对花括号，它相当于ALGOL，PASCAL语言中的begin和end。在{　}中可以包含许多语句，组成函数体。本例中，只有一个语句。

printf(　)是一个函数，是用C语言编写的标准的库函数。目前，在UNIX系统中，大多数计算机都装有这种标准的库函数，以下简称C库。printf的功能是：在输出设备或终端荧光屏上打印或显示一些信息以及需要输出的内容(将在以后各章节中介绍具体实例)。在这里是调用这一标准库函数输出一句话："C语言是非常重要的!"

双引号"　"中所包含的字符串"C language is very important!"，就是printf这一函数的参数，也就是我们要输出的信息。在"　"中的最后两个符号\n，表示换行。它表示在\n后面的信息均要输出到下一行去。在这里\n后面已无信息可传输，读者将在以后有关的实例中看到，\n后面就有换行的输出。

在printf(…)后面是一个分号；这是C语言中语句的结束符。它是不可缺少的。

如果我们还要显示或者打印输出一句话：

Programming in c is very useful!

应该如何编写呢？请看下面的例题。

例 1-2 要求输出更多信息的程序。

```
main()
  {
    printf("C language is very important! \n");
    printf("Programming in c is very useful! \n");
  }
```

这一程序的输出结果是：

C language is very important!

Programming in C is very useful!

例 1-3 求三个整型数 a,b,c 之和。

```
main()
  {
    int a, b, c, sum;
    a=2;
    b=4;
    c=6;
    sum=a+b+c;
    printf("The sum of a+b+c is % d\n", sum);
  }
```

在这个例题中，函数体由 6 个语句组成。其中：int a, b, c, sum 是说明语句，它说明 a, b, c, sum 四个变量为整型数。a=2,b=4,c=6,是赋值语句，其中“=”号为赋值运算符，表示将 2 赋给变量 a,4 赋给 b,6 赋给 c,sum=a+b+c 也是一个赋值语句，表示将 a+b+c 之和赋给 sum。最后一行是输出语句，它将要输出：The sum of a+b+c is，然后按% d 指示符的规定(%d 规定输出十进制数)输出后面紧跟着的变量 sum 的值。这个例子的输出结果是：

The sum of a+b+c is 12

在C语言中,常常可以用注释语句来提高程序的可读性。例1-3可以如例1-4那样来写。

例1-4 C语言注释语句的使用。

```
/* calculate adds of three integer values
 * and displays the results: sum
 */
main()
  {
    /* Declare variables */
    int a, b, c, sum;
      a=2;
      b=4;
      c=6;
      sum=a+b+c;   /* compute the result */

  /* Display the results */
      printf("The sum of a+b+c is %d\n", sum);

  }
/*  *  *  *  *
 * output: The sum of a+b+c is 12
 * * * * *
 */
```

C语言的注释语句由/*两个字符开始,由*/两个字符结束。在/*…*/之间的任何字符,都将被C编译系统忽略。注释语句可以写在C语言程序的任何地方。程序的头部,尾部,或中间,甚至在某一行中,例如在sum=a+b+c;之后也写了一个注释行:/* Compute the Result */。

为了进一步理解C语言的程序结构，我们先设计出计算三个变量之和的一个函数。

例 1-5 计算三个变量之和。

```
/* calculate adds of three integer values */
caladds()
  {
    /* Declare variable */
      int a, b, c, sum;
    /* Assign values and compute the sum result */
      a=2;
      b=4;
      c=6;
      sum=a+b+c;/* compute the result */
    /* Display the results */
      printf ("the sum of 2+4+6 is %d\n", sum);
  }
```

这个函数中，有变化的是：在 printf 中，“The sum of 2+4+6 is %d\n”，将 a+b+c 换成 2+4+6 了，输出结果将是什么呢？是否与例 1-4 相同呢？请读者考虑。

下面设计一个 main()函数。并且要调用 Caladds ()。

例 1-6 多于一个函数的结构。

```
/* This is main() */
main()
  {
    /* call function of calculate adds of
     * three integer values
     */
      caladds();
```

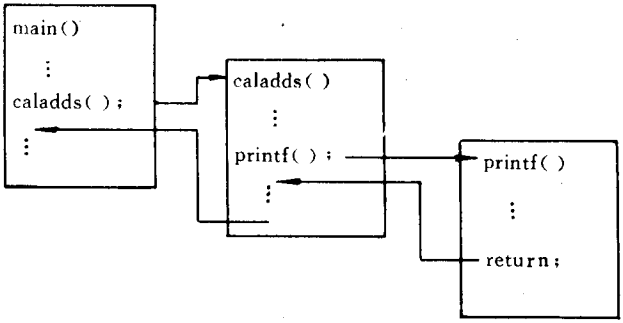

图 1-3 C 语言函数的调用关系

```
main(  )
{
函数体
}
function 1 (  )
{
函数体
}
function 2 (  )
{
函数体
}
⋮
function n (  )
{
函数体
}
```

图 1-4 C 语言的程序结构

```
}
/*  *  *  *  *
 * output:The sum of 2+4+6 is 12
  *  *  *  *  *
  */
```

现在我们有三个函数了：一个是 main()，一个是 caladds()，还有一个 printf()。前两个是自己编写的，后一个是库函数。其间关系如图 1-3 所示。当主函数启动之后，就调用子函数 caladds()，而子函数 caladds()又调用一个子函数 printf()，每次执行完一个子函数，其程序的控制就返回到调用函数的下一个语句去继续执行。

归纳以上所述，C 语言的程序结构可概括为图 1-4 所示。从中可以看出：

(1) main()函数，只有一个。

(2) 子函数 subfunction 可以有 n 个，也可

以没有。

(3) 无论 main(),subfunction(),都有自己的函数体。

(4) main()与 subfunction 排列的先后次序无关。

(5) 函数可以自己定义。

§1.5 C 语言的格式

从上述例题中已经看出,C 语言的书写格式是非常灵活的,自由的。可在一行的任意位置开始书写,一行可以写一个语句,也可以写一个以上的语句,注释行可在任意位置插入。总之,它不像 FORTRAN 语言那样,必须从第几列开始,对继续行、注释行都有特殊的要求。但为了便于阅读,加深理解,可用注释行,花托号{},以及空行来改善程序的层次结构。

在例 1-5 中用 4 个注释语句划分了程序的层次,提高了程序的可读性。在赋值及计算这一步中,将

```
a=2;
b=4;
c=6;
sum=a+b+c;
```

对齐,并向右错移几位,使之醒目。同时在注释行前后适当留有空行,既便于阅读,又使结构更加清晰。对于一个小型的程序,适当注意格式就可以了,而对于一个大型的程序,这些措施和作法是相当有效的。

思考题及练习

1. C 语言的特点有哪些?
2. C 语言的程序结构是什么?