21世纪高职高专计算机系列教材

C语言与算法分析

◎ 主 编 刘晓昱

华中科技大学出版社
http://press.hust.edu.cn

21世纪高职高专计算机系列教材

C语言与算法分析

主　编　刘晓昱

副主编　郭　倩　张　悦

主　审　范延滨

华中科技大学出版社

图书在版编目(CIP)数据

C语言与算法分析/刘晓昱　主编
武汉:华中科技大学出版社,2005年9月
ISBN 7-5609-3499-4

Ⅰ.C…
Ⅱ.①刘…　②郭…　③张…
Ⅲ.C语言-程序设计;算法分析
Ⅳ.TP31

C语言与算法分析　　刘晓昱　主编

责任编辑:曾　光　彭保林　　封面设计:刘　卉
责任校对:陈　骏　　责任监印:熊庆玉

出版发行:华中科技大学出版社
武昌喻家山　　邮编:430074　　电话:(027)87557437

录　　排:武汉万卷鸿图科技有限公司
印　　刷:湖北新华印务有限公司

开本:787×960　1/16　　印张:19.5　　字数:348 000
版次:2005年9月第1版　　印次:2005年9月第1次印刷　　定价:29.80元
ISBN 7-5609-3499-4/TP·585

内 容 简 介

本教材对 C 语言的精华部分作了较为细致的介绍，并针对目前高等院校和社会上举办的各种计算机等级考试而组织内容。

全书共分十二章。介绍了 C 语言的基础知识和算法的基本思想；C 语言的基本程序设计技术，C 语言函数的特点，函数的相互调用及变量的特性；C 语言的各种不同类型的数组及指针的使用；链表技术和文件的操作方法；数据结构中的树和图的基本概念和简单算法；C 语言的综合应用，图形的制作和音乐的演奏方法。

本书对传统课程的教学内容进行有机整合、精简、充实和提高，并辅以创新思维，教学内容更新幅度较大，具有较强的示范性和广泛的推广价值与辐射意义。

前　言

C 语言不仅为计算机专业人员所使用和喜爱，近年来，许多计算机应用人员也开始喜爱和使用 C 语言。实践证明，它是一种很好的程序设计语言。但 C 语言对一般初学者来说，规则较多，使用太灵活，不易掌握，学习会有一定的困难，而且 C 语言的应用范围越来越广，所涉及的知识也在不断地增加。基于此，我们编写了 C 语言与算法分析教程及配套辅导书，一方面以满足初学者的需要，另一方面，对 C 语言所涉及的深入技术作了一定介绍。此书既可作为 C 语言的教程，也可作为工具书以备编程时参考。

本教材本着理论够用、实践为主的原则将计算机应用专业的两个骨干课程——C 语言程序设计和数据结构的内容融为一体，以 C 语言为主线介绍了 C 语言的基本语法、C 程序设计方法以及 C 语言在数据结构上的应用，力求使读者在学习 C 语言的同时，理解数据结构的基本概念，掌握各种数据结构的存储方式、基本算法及其简单应用。

本书由刘晓昱编写第 5、6、8、11、12 章及附录一、二、三、四、五、六，郭倩编写第 4、7、9、10 章，张悦编写第 1、2、3 章，刘晓昱负责全书的统稿。青岛大学范延滨教授担任主审，并对本书初稿提出了许多宝贵意见，在此表示衷心感谢。

本教材内容基本覆盖了全国计算机等级考试的知识点，并收集和编写了大量的程序，由浅入深地培养读者的程序分析和设计能力。本书不仅可以作为高职高专院校计算机、自动化、电子、机电等专业的教材，也可以作为各种培训班的教材和计算机爱好者的自学参考书。

作为本书的姐妹篇，我们将同时出版本教程的学习指导，包括习题解答和实验指导，供读者学习时借鉴和参考。

在本书的编写和出版过程中得到青岛飞洋学院院长卢飞成院长和马仁富教授的帮助和支持，在此表示诚挚的感谢。

由于编者水平有限，错误在所难免，请广大读者批评指正。

编　者

2005 年 7 月

目　录

第1章

C 语言概述

1.1 C 语言简史

C 语言是在 B 语言的基础上发展起来的，它的原型是 ALGOL 60 语言。C 语言的发展如图 1.1 所示。

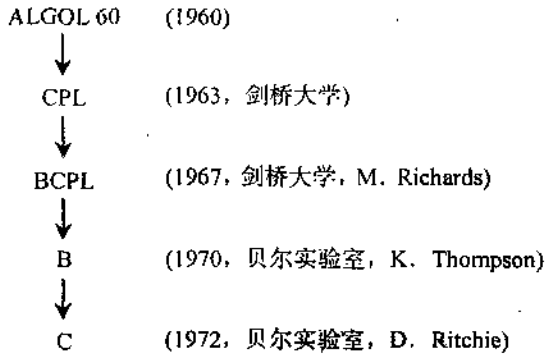

图1.1 C语言的由来

1963 年，英国剑桥大学将 ALGOL 60 语言发展成为 CPL(Combined Programming Language)语言。

1967 年，英国剑桥大学的马丁·理查德(Matin Richards)对 CPL 语言进行了简化，于是产生了 BCPL(Basic Combined Programming Language)语言。

1970 年，美国贝尔实验室的肯·托普生(Ken Thompson)将 BCPL 语言进行了修改，命名为“B 语言”，并且他用 B 语言写了第一个 UNIX 操作系统。

1973 年，美国贝尔实验室的 D. M. 里奇(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。

为了使 UNIX 操作系统推广，1977 年 D. M. Ritchie 发表了不依赖于具体机器系统的 C 语言编译文本《可移植的 C 语言编译程序》。

1978 年 Brian W. Kernighian 和 D. M. Ritchie 出版了名著《The C Programming

Language》，此书对 C 语言作了详细的描述，从而使 C 语言成为目前世界上广泛流行的高级程序设计语言。

随着微型计算机的日益普及，出现了许多 C 语言版本。由于没有统一的标准，使得这些 C 语言之间出现了一些不一致的地方。为了改变这种状况，美国国家标准研究所(ANSI)为 C 语言制定了一套 ANSI 标准，通常称为 ANSI C。

1987 年，ANSI C 又公布了新标准——87 ANSI C。目前流行的 C 编译系统都是以此标准为基础的。

1.2 C 语言的特点

(1) 简洁紧凑、灵活方便。

C 语言总共只有 32 个关键字，9 种控制语句，程序书写自由，主要用小写字母表示。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。

(2) 运算符丰富。

C 的运算符包含的范围很广泛，共有 34 个运算符。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 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) 数据结构丰富。

C 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类型的运算。C 语言引入了指针概念，使程序效率更高。另外，C 语言具有强大的图形功能，支持多种显示器和驱动器，且计算功能、逻辑判断功能强大。

(4) C 是结构式语言。

结构式语言的显著特点是代码及数据的分隔化，即程序的各个部分除了必要的信息交流外彼此独立。这种结构化方式可使程序层次清晰，便于使用、维护以及调试。C 语言是以函数形式提供给用户的，这些函数可方便地调用，并具有多种循环、条件语句控制程序流向，从而使程序完全结构化。

(5) C 语法限制不太严格、程序设计自由度大。

一般的高级语言语法检查比较严，能够检查出几乎所有的语法错误，而 C 语言允许程序编写者有较大的自由度。

(6) C 语言允许直接访问物理地址，可以直接对硬件进行操作，因此既具有高级语言的功能，又具有低级语言的许多功能，能够像汇编语言一样对位、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，可以用来编写系统软件。

(7) C 语言程序生成代码质量高，程序执行效率高。

一般只比汇编程序生成的目标代码效率低 10%～20%。

(8) C 语言适用范围大，可移植性好。

C 语言有一个突出的优点就是适合于多种操作系统，如 DOS、UNIX，也适用于多种机型。

(9) C++源于 C 语言。

1980 年贝尔实验室的 Bjarne Stroustrup 对 C 语言进行扩充，推出了“带类的 C”，多次修改后起名为 C++。C++改进了 C 语言的不足之处，支持面向对象的程序设计，在改进的同时保持了 C 语言的简洁性和高效性。

1.3　C 程序的基本结构

为了说明 C 语言源程序结构的特点，先看以下几个程序。这几个程序由简到难，表现了 C 语言源程序的组成结构。通过分析这几个简单的 C 程序，了解 C 程序的基本结构及书写规则。

例 1.1　C 语言程序举例。

```
main( )
{
 printf("HELLO  WORLD! ");
}
```

此程序的作用是输出一行信息，程序运行结果：

```
HELLO  WORLD!
```

分析此程序：

main 是主函数的函数名，表示这是一个主函数。每一个 C 源程序都必须有且只能有一个主函数即 main 函数。

花括号{}括起来的是函数体部分，此函数中只有一条语句“printf("HELLO WORLD! ");”。

函数调用语句，printf 函数的功能是把要输出的内容通过显示器显示出来。printf 函数是一个由系统定义的标准函数，可在程序中直接调用。

printf 函数中双引号内的字符串是按原样输出。printf 语句后的分号“;”表示一条语句的结束。

例 1.2　比较两个程序输出结果的不同。

程序 1：

```
main( )
{
 printf("HELLO  WORLD! ");
```

```
 printf("This is C program。");
}
```

程序运行结果：

```
HELLO  WORLD! This is C program。
```

程序 2：

```
main( )
{
 printf("HELLO  WORLD! \n This is C program。");
}
```

程序运行结果：

```
HELLO  WORLD!
This is C program。
```

分析此程序：

两个程序中只有 printf 中是否有“\n”。

printf 函数中“\n”是换行符，在换行符后的字符串换行后按原样输出。

例 1.3 求两个整数 a 和 b 之和 sum。

```
main( )
{
int a,b,sum;                          /*定义整型变量 a、b 和 sum*/
a=123; b=456;                         /*为整型变量 a、b 赋值*/
sum=a+b;
printf("sum is %d\n",sum) ;           /*输出整数 a 和 b 之和 sum*/
}
```

程序运行结果：

```
sum is 579
```

分析此程序：

“/*……*/”是注释部分，为了方便阅读程序而安排，对程序的编译和执行没有任何影响。注释可以加在程序中的任何位置。

程序中“int a,b,sum;”是变量的定义部分，这里我们用“int”定义“a,b,sum”为整型变量，C 语言规定任何变量使用之前都要先定义。

语句 printf("sum is %d\n",sum)；中“%d”表示“以十进制整数形式输出”，在执行时用 printf 语句中最右端的“sum”的值代替“%d”。

例 1.4 主函数调用子函数的例子。求 x,y 中的较大者。

```
main( )                    /* 主函数 */
{ int a,b,c;               /* 定义变量 */
scanf("%d, %d",&a,&b);     /* 输入变量 a,b 的值 */
c=max(a,b);                /* 输出 max 函数，将得到的值赋给 c */
printf("max=%d",c);        /* 输出 c 的值 */
```

```
}
int max(int x, int y)  /* 定义 max 函数，函数值为整型，x,y 为形式参数 */
{ int z;               /* max 函数中用到的变量 z，也要加以定义 */
if (x>y) z=x;
else z=y;
return(z);             /* 将 z 的值返回，通过 max 带回调用处 */
}
```

程序运行时：

输入：8，5

输出：max=8

分析此程序：

“scanf("%d,%d",&a,&b)；”语句是将两个数的值分别输入到变量 a 和 b 的地址所标志的单元中，就是给 a、b 变量赋值。

在此程序中，有两个函数，分别是 main( )和 int max(int x, int y)，在 main( )函数中调用了后者，main( )是主调函数，后者是被调函数。

“c=max(a,b)；”语句是将函数 max 的值赋给变量 c，其中 a、b 是参数，执行时将变量 a 和 b 的值传给被调函数 max。

max 的作用是将 x、y 中较大的数赋值给 z，并通过语句“return(z)；”将其返回给主调函数。

说明 1：通过以上几个例子可以看出 C 语言源程序的结构特点。

① 一个 C 语言源程序可以由一个或多个源文件组成。

② 每个源文件可由一个或多个函数组成。

③ 一个源程序不论由多少个文件组成，都有且只有一个 main 函数，即主函数。

④ 每一个说明，每一个语句都必须以分号结尾，但预处理命令，函数头和花括号“}”之后不能加分号。

⑤ 标志符，关键字之间必须至少加一个空格以示间隔。若已有明显的间隔符，也可不再加空格来间隔。

说明 2：C 语言源程序的一般结构。

```
函数部首
{
声明部分
                  } 函数体
执行部分
}
```