

C 程序设计语言及其应用



发贵 廖开际 编著

华南理工大学出版社

前 言

C语言自 70 年代问世以来,因其简洁、功能多、表达力强、代码紧凑、灵活、高效、实用和可移植性好,且兼具高级语言的优点及低级语言的许多特点,所以它既适于编写操作系统、编译程序、数据库管理系统等系统软件,又适于编写网络软件和应用软件,已成为当今流行的程序设计语言。

本书是应当前专业改造、课程建设及教学改革的需要,根据笔者近几年来采用多媒体教学的体会,在已使用多年的多媒体讲义的基础上编写的。书中的全部内容——板书部分和笔记部分、单独的板书部分、练习和上机实习部分、练习和上机实习部分的参考答案,都存储在向全国开放的华南理工大学的大型计算机 S/390 和小型机 RS/6000 上。本书有以下特点:

- 较全面、系统地介绍了 C 语言及其程序设计方法和应用,共分九章,包括 C 的数据类型、各种语句、程序结构及其程序设计的基本方法、技巧和实际应用,突出了 C 的特点。

- 内容的选择、组织和编排非常适合多媒体的教学:它以**教师板书(大字)**和学生上课将做的**笔记(小字)**形式组织。**板书(大字)**部分可制成投影胶片或直接用于实物投影仪,或直接通过计算机显示于大型屏幕或电视上,可改变传统的上课时教师忙于写黑板而学生忙于抄笔记的状况,这样可提高教和学的效率。

- 每章由**内容提要、目标、内容、笔记、内容、笔记……本章小结**组成。其中笔记部分的内容,教师可根据学生的情况来选择讲授。如第三章(语句和流程控制)中笔记部分的流程图,对初次接触程序设计的学生需要详细讲授,其他情况可少讲或不讲。

- 例子的选择从易到难,既有应用设计的,也有系统设计的,以充分体现“C 语言”的特点。在笔记部分和第九章也有一些实例,可供教师选择和学生理解掌握用。

- 练习内容的组织,使得练习方式既可采用传统的有纸练习方式,又可采用现代无纸练习部分的形式:在计算机上直接答题和提交练习。实验的内容,前后连贯、关联,使学生通过实验可具备进行大的实际的应用程序设计的能力。

- 本书的教学时数可依据学生的情况弹性处理:可用 48~72 学时,其中讲授 32~48 学时,上机实验 16~24 学时,实验环境可采用 UNIX C 或 TURBO C。

- 教材与电脑版本教材配合,以方便教师和学生使用。

华南理工大学电子与信息学院、计算机系许多领导和老师及华南理工大学出版社对本书的出版给予了热情的支持和鼓励,在此表示衷心的感谢。

书中不足之处恳请读者批评指正,以便修订补充。

编 著 者

1999 年 1 月

E-mail: fgliu@163.net, liaokj@163.net

本课程主要参考书

- 1 龚杰民, 金益群. C 语言程序设计及其应用. 西安: 西安电子科技大学出版社, 1986
- 2 孟庆昌, 孙玉芳. C 语言及其应用. 北京: 宇航出版社, 1988
- 3 谭浩强. C 程序设计. 北京: 清华大学出版社, 1991
- 4 李智渊. C 语言. 第三版. 成都: 电子科技大学出版社, 1991

目 录

1 C语言概述.....1	3.1 语句和分程序.....29
1.1 C语言的历史和特点.....1	3.1.1 表达式语句.....29
1.1.1 历史.....1	3.1.2 返回语句.....29
1.1.2 特点.....2	3.1.3 空语句.....29
1.2 C语言程序简述.....4	3.1.4 复合语句与分程序.....30
1.2.1 程序结构.....4	3.2 if 语句.....32
1.2.2 函数的结构.....6	3.2.1 if语句的一般形式.....32
1.2.3 C语言程序的编译与运行.....9	3.2.2 用条件表达式简化程序.....32
本章小结.....10	3.3 switch 语句.....34
2 基本数据类型、运算符和表达式.....12	3.4 循环.....38
2.1 词汇及词汇约定.....12	3.4.1 while语句.....38
2.1.1 词汇表.....12	3.4.2 do-while语句.....39
2.1.2 词法约定.....12	3.4.3 for 语句.....39
2.2 常量.....13	3.4.4 转向语句.....42
2.2.1 常量与符号常量.....13	本章小结.....44
2.2.2 整常量和长整常量.....13	4 函数和程序结构.....45
2.2.3 实常量.....13	4.1 标准I/O函数.....45
2.2.4 字符常量.....13	4.1.1 字符I/O函数.....45
2.2.5 字符串常量.....13	4.1.2 格式I/O函数.....46
2.3 变量及变量定义.....15	4.2 函数.....49
2.3.1 变量及变量名.....15	4.2.1 函数的定义与调用.....49
2.3.2 变量定义形式.....15	4.2.2 函数参数的传递.....54
2.4 基本数据类型.....15	4.2.3 函数的递归.....57
2.4.1 基本类型.....15	4.3 变量的作用域与存储类别.....59
2.4.2 类型转换.....16	4.3.1 自动存储类变量(auto)——局部变量.....61
2.5 运算符和表达式.....18	4.3.2 寄存器存储类变量.....62
2.5.1 概述.....18	4.3.3 静态变量.....62
2.5.2 算术运算符.....18	4.3.4 外部变量——全局变量.....63
2.5.3 增量和减量运算符.....19	4.4 C语言预处理程序.....70
2.5.4 关系运算符和逻辑运算符.....20	4.4.1 程序从编写到编译、执行的过程.....70
2.5.5 位操作.....21	4.4.2 宏替换.....71
2.5.6 条件运算符.....23	4.4.3 包含文件.....74
2.5.7 赋值运算符.....23	4.4.4 条件编译.....74
2.5.8 取地址.....25	4.4.5 行控制.....75
2.5.9 逗号运算符.....25	本章小结.....75
2.5.10 初等量运算符.....25	5 枚举类型和数组.....77
2.5.11 其他运算符.....25	5.1 枚举类型.....77
本章小结.....26	5.1.1 类型说明.....77
3 语句和流程控制.....29	5.1.2 枚举变量的定义形式.....78

5.2 数组	80	7.5.1 联合体的定义	116
5.2.1 数组定义和存储方式	80	7.5.2 联合体与结构体的异同	116
5.2.2 数组的引用	80	7.6 类型定义	120
5.2.3 数组的初始化	80	本章小结	121
本章小结	84	8 输入和输出	124
6 指针	85	8.1 标准输入输出	124
6.1 指针	85	8.1.1 字符串输入	124
6.1.1 指针的概念	85	8.1.2 字符串输出	124
6.1.2 指针的优点	85	8.2 C 文件系统概述	125
6.1.3 指针变量的定义	86	8.3 缓冲型文件系统	125
6.1.4 指针的使用和运算	86	8.3.1 文件类型指针	125
6.2 指针作为函数参数	89	8.3.2 打开文件(fopen 函数)	127
6.3 指针和数组	92	8.3.3 关闭文件(fclose 函数)	127
6.3.1 用指针描述一维数组及其元素的相对位置	92	8.3.4 字符读写函数	127
6.3.2 数组名和指针	92	8.3.5 数据块(记录)输入/输出函数(fread/ fwrite)	130
6.3.3 用指针描述二维数组及其元素的相对位置	94	8.4 文件定位与随机存储	132
6.3.4 指针数组和数组指针	95	8.5 非缓冲文件系统(低级 I/O)	135
6.4 主函数参数	97	8.5.1 open 函数	135
6.5 函数指针	99	8.5.2 close 函数	135
6.6 实例	102	8.5.3 creat 函数	136
本章小结	103	8.5.4 read 函数	136
7 结构体与联合体	104	8.5.5 write 函数	137
7.1 结构体	104	8.5.6 lseek 函数	137
7.1.1 结构体类型的定义	104	8.6 其他各类函数	139
7.1.2 结构体变量的定义	105	8.6.1 字符类别测试和转换函数	139
7.1.3 结构体变量的初始化	105	8.6.2 退回字符函数	139
7.1.4 结构体变量的使用	106	8.6.3 字符串处理函数	139
7.2 结构体数组和指向结构体的指针	106	8.6.4 系统调用函数	139
7.2.1 结构体数组	106	8.6.5 存储函数	139
7.2.2 指向结构体的指针	107	8.7 文件 I/O 应用实例	140
7.3 引用自身的结构体	111	本章小结	142
7.3.1 引用自身的结构体	111	9 C 语言程序综合应用实例	143
7.3.2 链表的建立	111	9.1 数据结构中的应用	143
7.3.3 链表的遍历	112	9.2 在数值计算中的应用	145
7.3.4 链表的插入	112	9.3 在系统设计中的应用	151
7.3.5 链表的删除	113	9.4 实用项目的开发	154
7.4 位段存取	115	本章小结	163
7.4.1 位段的定义	115	附录一 常用字符与 ASCII 代码对照表	164
7.4.2 位段的引用	116	附录二 C 系统常用库函数	165
7.5 联合体	116	附录三 关键字表	168
		附录四 练习	169
		附录五 上机实验	176

1 C语言概述

内容

本章介绍 C 语言的历史及其特点, 概述 C 语言的程序结构及其运行。

目标

- 了解 C 语言的产生及其家族;
- 描述 C 语言的特点;
- 描述 C 程序的结构;
- 写简单的 C 程序;
- 运行 C 程序。

1.1 C 语言的历史和特点

1.1.1 历史

C 语言产生于 70 年代, 由贝尔实验室的 Dennis Ritchie 设计并编写出来的。“家谱”如图 1.1 所示。

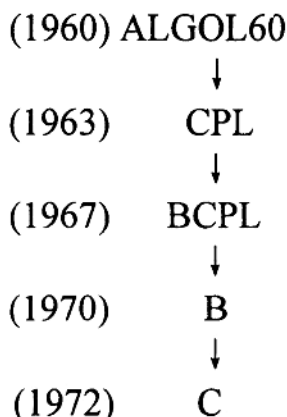


图 1.1 C 语言的发展历程

1.1.2 特点

(1)表达力强

(2)数据类型和控制流程丰富

- 数据类型

数据类型 { 基本类型: 整型(int, short, long, unsigned)
 实型(float, double)
 字符型(char)
 空类型(void)
枚举类型(enum)
指针类型
构造类型: 数组、结构体、联合体

- 控制流程:

顺序

选择(if-else、switch)

循环(while、do-while、for)

- 多种存储类:

自动存储类(auto)

寄存器存储类(register)

静态存储类(static)

外部存储类(extern)

(3) 具有预处理功能

(4) 表达简洁, 程序小型, 输入量少

(5) 大量使用指针

(6) 效率高

(7) 可移植性好

(8) 缺点

笔记:

(1)关于 C 语言的历史:

ALGOL60 语言是 1960 推出。

CPL 语言(Combined Programming Language)。1963 年剑桥大学在 ALGOL60 的基础上推出 CPL, 其特点是规模大, 实现难。

BCPL 语言(Basic Combined Programming Language)。1967 年 Martin Richards 对 CPL 作了简化, 推出了基本 CPL, 即 BCPL。

B 语言是 BCPL 的进一步简化。

C 语言(因 C 是 BCPL 的第二字母)。1970 年, 贝尔实验室的 Dennis Ritchie 开始在 B 语言的基础上设计 C 语言。Dennis Ritchie 后来又和 Ken Thompson 一起开发 UNIX 操作系统。几乎所有由 UNIX 提供的软件工具, 包括操作系统和 C 语言的编译程序本身, 都是用 C 语言编写的。

(2)关于 C 语言的特点:

有丰富的运算符。可表达数值运算、字运算、位运算和地址运算。

指针是 C 语言的灵魂。灵活多样的指针类型, 使构造链表、树、栈、图等复杂的数据结构变得非常方便。然而, 这也正是学习 C 语言的困难所在。

C 语言中无字符串类型和文件类型, 其字符串用字符数组表示, FILE 类型则是用 typedef 对结构体类型取的别名。

C 语言的结构化控制结构符合结构化程序设计的要求, 可编写结构非常好的程序。控制流程基本上是传统的形式, 有条件语句、循环结构(for 语句、while 语句、do-while 语句)、开关语句、从开关语句或循环体立即出口的 break 语句、使下次循环开始的 continue 语句等。C 语言还有标号和 GOTO 语句。

C 语言提供多种存储类别, 指明数据的存储方式。它的静态和外部存储机制有助于设计信息隐蔽、抽象的模块化结构程序。

C 程序运行的速度很快, 只比汇编程序慢 20% 左右。

事实上, C 语言的许多优点也是它的缺点。用 C 语言编程, 自由度大, 如对变量的类型约束不够严格, 整型、字符型及逻辑数据通用, 指针和数组通用等。过多的通用性限制了编译程序对 C 程序作充分的语法检查, 不能及时发现程序中的错误, 给程序的调试和排错造成一些困难; 语言的运算符、优先级太多, 不便于记忆; 类型检查弱, 数据类型转换比较随意, 影响了程序的安全性。

1.2 C 语言程序简述

1.2.1 程序结构

C 程序结构如图 1.2 所示。

预处理程序命令
外部数据说明或定义

函数

预处理程序命令
外部数据说明或定义

函数

⋮

预处理程序命令
外部数据说明或定义

函数

图 1.2 C 程序结构示意图

例 1.1 输出信息。

```
/*Output message */  
main()  
{  
    printf("Hello world!\n");  
    printf("Programming in C is fun.\n");  
}
```

输出结果：

```
Hello world!  
Programming in C is fun.
```

例 1.2 求和。

```
/*Sum of a, b, c */
```

```
main()
```

```
{ int a, b, c, sum;
```

```
  a=1; b=45;
```

```
  scanf("%d", &c);
```

/*一行可写多个语句*/

/*scanf()为格式输入函数，格式字符串"%d"表示以十进制格式显示参数值*/

```
  sum=a+b+c;
```

```
  printf("sum=%d\n", sum);
```

/*以十进制整数输出变量 sum 的值*/

```
}
```

输出结果：

54 (对 c 输入 54)

sum=100

笔记：

关于例 1.1：

例 1.1 是一个最简单的 C 程序，只包含一个 main 函数。

(1) C 程序由若干个函数并列组成，其中有且仅有一个名为 main 的函数，称为主函数。它是程序执行的入口点。

(2) “/*……*/” 表示注释。

(3) printf 为格式输出函数。其中引号内的字符串原样输出，“\n”表示输出字符串后，光标换行。

(4) C 程序中的每个语句必须以分号作为结束符。

(5) C 程序中函数的函数体用大括号“{ }”括起来。

1.2.2 函数的结构

一个函数定义由函数头和函数体组成。函数结构如下：

```
函数头
{
    函数体
}
```

• 函数头：包括函数属性、函数类型、函数名、形式参数(可以为空)、形式参数类型。如：

```
extern double function(int x, int y)
```

或

```
extern double function(x, y)
int x, y;
```

• 函数体：实现函数功能的代码。包括：
说明和定义部分——说明数据结构和定义函数专用的变量；

执行部分——描述函数功能的代码。

例1.3 将华氏温度转换为摄氏温度。

```
/*Temperature exchange */
main()
{int l,u,step;
 float f,c;
 float turn( );          /*函数说明*/
 l=0;
 u=300;
 step=20;
 f=l;
 while( f<=u )
 {c=turn(f);             /*调用函数 turn()*/
  printf("%4.0f%6.1f\n",f,c);
  f += step;
```

```

    }
}

/*以下定义函数 turn()*/
float turn(x)          /*函数返回浮点型值*/
float x;               /*说明形式参数 x*/
{ float p;
  p=(5.0/9.0)*(x-32.0);
  return (p);
}

```

笔记:

关于例 1.3:

(1) 例 1.3 中语句 “printf(“%4.0f%6.1fn”,f,c);” 的意义是: 变量 f 和 c 以浮点数格式输出, 变量 f 的输出结果至少占 4 个字符宽度, 不保留小数; 变量 c 的输出结果至少占 6 个字符宽度, 保留一位小数; 光标转到下一行。

(2) C 程序书写自由, 一行可写多个语句, 一个语句也可写成多行。

(3) 例 1.3 的程序中包含两个函数: 主函数 main() 和求数据转换函数 turn()。函数名后有一对圆括号, 圆括号内可以包含函数的参数, 即使无参数, 圆括号也不可省。

(4) 注意主函数中的函数说明语句 “float turn();”, 它用于说明名字 turn 是一个返回浮点型值的函数名。C 语言中遵守“先说明或定义, 后使用”的原则。若主函数中未对 turn() 作说明, 则主函数的第六行中的 turn() 函数被缺省设定为整型函数, 而后面的 turn() 函数定义却是浮点型, 这会出现类型不一致的错误。

(5) 在某些情况下, 函数可以没有说明和定义部分, 甚至可以没有执行部分。允许没有任何内容的函数的合法性, 便于程序功能的扩充。

(6) 在 C 程序中除了包含主函数和用户自定义的函数外, C 系统还提供了丰富的库函数。标准 C 系统提供一百多个库函数, 而 TURBO C、MS C4.0 提供了三百多个库函数。上述例子中的 scanf()、printf()、getchar() 都是 C 系统提供的库函数。在程序中, 如果使用了除 scanf() 和 printf() 以外的库函数, 必须在程序的预处理命令中包括 include 命令行。如例 1.4。

例 1.4 统计输入正文的字符的个数。

```
#include <stdio.h>
main()
{long nc=0;
  while(getchar() != EOF)      /*调用系统库函数*/
    ++nc;
  printf("The number of characters is %d\n",nc);
}
```

输入如下:

abc def ghi

jkl mnl

^z

The number of characters is 20

例 1.5 统计输入正文行中的词数。

```
#define YES 1      /*预处理命令, 定义符号常量*/
#define NO 0
#include <stdio.h>
main()
{ int c ,nword,inword;
  nword=0;
  inword=NO;
  while((c=getchar()) != '\n')
  { if (c==' ' || c=='\t') inword=NO;
    else if (inword==NO)
      { inword=YES;
        ++nword;
      }
  }
  printf("word: %d \n",nword);
}
```

例 1.6 统计输入正文的字符数、词数和行数。

```
#define YES 1
#define NO 0
#define EOF -1
#include <stdio.h>
main()
{ int c, nline, nword, nchar, inword;
  inword=NO;
  nline=nword=nchar=0;
  while((c=getchar())!=EOF)
  { ++nchar;
    if(c=='\n')
      ++nline;
    if(c==' '||c=='\n'||c=='\t')
      inword=NO;
    else if(inword==NO)
      { inword=YES;
        ++nword;
      }
  }
  printf("line: %d word: %d character:%d\n",
        nline,nword,nchar);
}
```

1.2.3 C 程序的编译与运行

Turbo C 集成环境:

(1)进入 TURBO C 程序开发环境: 执行命令

tc

或 tc 文件名

如: tc example.c

(2)编辑源程序文件。

(3)编译与连接：按“F9”功能键。

(4)执行：按“Ctrl+F9”键，编译并执行；或运行 RUN 菜单中的 RUN 命令。

(5)退出 TURBO C 环境：按“Alt+x”键。

UNIX 环境：

(1)编辑源程序文件：用编辑工具 vi, ed 等编辑源文件 (扩展名为.c)

如：vi example.c

(2)编译链接：用命令

cc -o [可执行文件名] 源文件名 [-l 库名]

或 cc 源程序文件名

如：cc -o example.exe example.c

或 cc example.c

(3)运行程序：用命令

可执行文件名

如：example.exe (编译时，用户已指定执行文件名)

或 a.out (编译时，用户未指定执行文件名)

本章小结

- C 语言属于 Algol 家族。
- C 语言具有简洁、灵活、高效、实用和可移植性好等特点。
- C 程序由若干函数并列而成，其中，有且只有一个主函数，它是程序执行的入口点。
- 可在多种环境下运行 C 程序。

笔记:

(1)关于 Turbo C 集成环境

打入命令 `tc` 进入 TURBO C 程序开发环境, 在屏幕顶部出现一行主菜单:

File Edit Run Compile Project Options Debug Break/watch

利用菜单可以方便地创建、打开、保存文件; 编译、调试、执行程序。记住下列快捷键可加快操作速度:

F2——保存文件

F3——装入文件

F9——编译并连接

F6——从显示错误状态返回到编辑状态

F10——选择主菜单中的命令

Ctrl+F9——编译并运行

Alt+F5——查看程序输出结果

Alt+x——退出 TUROB C, 返回操作系统

(2)关于 UNIX 环境

`cc` 命令的另外几种用法:

- `cc -O -o example.exe example.c -lm`

编译联结、产生优化的目标代码程序 `example.exe`。这里的 `lm` 选项为联结库文件 `/lib/libm.a` (数学库文件)。

- `cc -P example.c`

只运行宏预处理程序, 并将 `example.c` 文件处理后存入相应的 `example.i` 文件中。在 `example.i` 中不再包含以 “#” 引导的行。

- `cc -S example.c`

编译源程序 `example.c`, 保存得到的汇编语言程序于文件 `example.s` 中。

2 基本数据类型、运算符和表达式

内容

本章介绍C语言的基本数据类型、运算符和表达式。

目标

- 了解C语言的词汇及词法约定；
- 掌握常量、变量及其变量定义；
- 掌握基本数量类型及类型转换；
- 掌握运算符和表达式的定义等。

2.1 词汇及词法约定

2.1.1 词汇表

数字：0~9

大小写字母：a~z, A~Z

下划线：_

特殊符号：主要包括运算符和关键字

2.1.2 词法约定

标识符：只能由字母、数字和下划线三种字符组成，且第一个字符必须是字母或下划线。

关键字：C的保留字，不能作标识符。

注释：/*.....*/

特定字：define、include、ifdef等。