



C语言程序设计

尤晋元 江永清 编



上海科学技术文献出版社

C 语 言 程 序 设 计

尤晋元 江永清 编

上海科学技术文献出版社

JS151/17

C语言程序设计

尤晋元 江永清 编

*

上海科学技术文献出版社出版发行
(上海市武康路2号)

新华书店经销 商务印书馆上海印刷厂印刷

*

开本 787×1092 1/32 印张 11 字数 266,000

1987年5月第1版 1987年5月第1次印刷

印数: 1—7,500

书号: 15192·513 定价: 2.50 元

《科技新书目》139-255

前 言

C语言是目前国际上应用非常广泛的现代程序设计语言,不论在学术界还是在计算机软件产业界,它都受到了普遍欢迎。C语言是 D. M. Ritchie 1972 年左右在贝尔实验室中为编写 UNIX 操作系统而设计和实施的。由于 D. M. Ritchie 在 C 语言和 UNIX 系统设计和实施方面的突出贡献,美国计算机学会于 1983 年授予他计算机科学和工程方面的最高荣誉——图林奖。

C语言的主要特点是表达能力强,表达方式简练而灵活,生成的目标代码时、空效率高,可移植性好,提供了大型程序设计所需的各种基本手段等。在系统程序设计、计算机辅助设计、文字处理、企业管理、软件工程和程序设计环境等领域中,C语言占有非常重要的地位。

为使初学者易于入门、使已经熟悉 C 语言的读者又能有新的收获,本书介绍了 C 语言的各种特征和组成部分,并且也包含了它的最新进展。在内容安排上则遵循了从易到难、从简到繁、循序渐进的原则,突出了 C 语言的实用性和灵活性。

除了必要的语法、语义知识以外,学习程序设计语言的关键是实践,为此,本书使用了八十几个实例程序,它们都在 Altos 68000 机配置的 UNIX 系统上运行过,并得到了它们的运行结果。实例程序大部分比较简短,主要目的是帮助读者理解基本概念以及用 C 语言编程的方法。除第十一章外,其它各章都附有练习,以便读者检查自己的学习情况,如果具备条件,也可以

作为上机实习题。

本书可供自学 C 语言的读者使用,也可以作为讲授 C 语言的教材。

在编写初稿时,上海交通大学的孔祥松、陈奇明、张貽彤、徐良伟、盛吉人、周伟、徐飞跃等同志参加了部分工作,上海交通大学应用数学系计算实验室的何焕熹等同志对本书的编写工作也给予了大力支持,在此一并表示感谢。

尤晋元 江永清

1985 年 9 月

于上海交通大学

目 录

第一章 引论	1
一、C 语言的基本特点	1
二、开发 C 语言程序的基本过程	4
三、用 C 语言开始编写程序	7
练习	13
第二章 变量、常数、数据类型和算术表达式	15
一、变量	15
二、基本数据类型和常数	16
三、算术运算符、赋值算符和算术表达式	24
练习	32
第三章 控制流	34
一、语句和复合语句	34
二、关系和逻辑连接算符	34
三、if_else 和 else_if 语句	37
四、switch 语句	47
五、while 和 for 循环语句	50
六、do_while 循环语句	59
七、break 和 continue 语句	61
八、逗号算符和空操作语句	64
练习	65
第四章 数组	67
一、一维数组	67
二、数组元素的初始化	76
三、二维数组	82
练习	84

第五章 函数和变量	86
一、函数基础	86
二、函数嵌套调用	98
三、函数和数组	104
四、变量的存储类	113
五、函数递归调用	122
练习	126
第六章 结构	127
一、结构及结构变量	127
二、结构和函数	134
三、结构初始化	143
四、结构和数组	144
五、结构嵌套	148
练习	149
第七章 字符串	151
一、字符数组和字符串	151
二、变长字符串	153
三、字符串、结构和数组的综合应用	168
四、字符及有关操作	174
练习	180
第八章 指针	182
一、指针基础	182
二、指针和结构	188
三、指针和函数	198
四、指针和数组	203
五、其它	215
练习	218
第九章 二进制位运算	219
一、二进制数位运算符	219

二、位段.....	232
练习.....	237
第十章 预处理程序	238
一、宏定义.....	238
二、文件包括.....	248
三、条件编译.....	251
练习.....	256
第十一章 分块编译和模块通讯	257
一、分块编译.....	257
二、模块间的通讯.....	259
三、名字作用范围和变量初始化小结.....	264
第十二章 输入和输出	267
一、标准文件的字符输入和输出.....	268
二、标准文件的格式化输入和输出.....	269
三、标准输入输出文件转向和文件结尾符.....	280
四、一般文件的输入和输出.....	283
练习.....	289
第十三章 其它	290
一、联合.....	290
二、枚举类型.....	293
三、类型定义.....	296
四、数据类型转换.....	298
五、命令参数.....	301
六、动态存储分配.....	304
练习.....	309
附录 A C 语言摘要.....	310
附录 B UNIX C 库函数	339

第一章 引 论

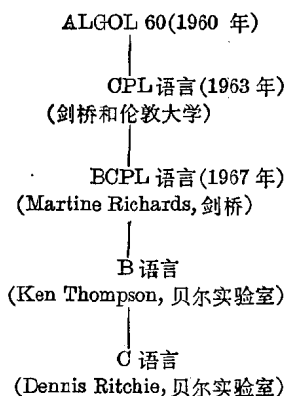
本章先简单说明 C 语言的特点以及 C 语言程序的开发过程, 然后介绍如何用 C 语言编制程序, 目的是使读者对 C 语言有初步的轮廓性的了解。

一、C 语言的基本特点

C 语言是一种受到广泛重视并已得到普遍应用的计算机程序设计语言, 也是国际上公认的最重要的少数几种通用程序设计语言之一。

C 语言是 D. M. Ritchie 1972 年左右在贝尔实验室中为编写 UNIX 系统而设计并加以实施的。UNIX 操作系统源代码的 90% 以上以及绝大多数 UNIX 系统实用程序都使用 C 语言编写。该系统的许多特点, 如简易便于理解、灵活易于修改、可移植性良好等在一定程度上受益于 C 语言。UNIX 系统获得的巨大成功与 C 语言是密切不可分割的。另一方面, C 语言也受到了 UNIX 系统的强有力支持。C 语言可以直接使用 UNIX 操作系统面向程序设计的界面——数十条系统调用, 可以直接调用 UNIX 系统的各种实用程序等。C 语言既是 UNIX 程序设计环境的一个最重要组成部分, 它又与环境中的其它部分有机地联系在一起, 这大大加强了其自身的能力。目前, C 语言已能在若干种非 UNIX 系统上运行, 如 DEC VMS 系统、IBM PC DOS 等, 但是相比较而言, 在 UNIX 系统支持下运行的 C 语言具有更强的能力。

图 1-1 示出了 C 语言的由来。ALGOL 语言是一种比较复杂



的语言, 它对很多种程序设计语言, 如 PASCAL、PL/1 等都有很重要的影响, 但这是一种面向问题的语言, 与计算机硬件距离很远, 不宜用于编写系统程序。CPL 语言则企图将 ALGOL 语言改造得更接近于计算机硬件一些, 但是它也是一种比较大的语言, 难于学习和实施。BCPL 语言对 CPL 语言作了精简并保持了它的基本优点。Ken

图 1-1 C 语言的发展概况

Thompson 在设计和实施 UNIX

系统时对 BCPL 语言作了进一步简化, 设计了非常简单又很接近硬件的 B 语言。但是 BCPL 和 B 语言在简化方面走得太远了, 因此语言的能力有限, 只适于解决一些特殊类型的问题。D. M. Ritchie 在设计 C 语言时, 既保持了 BCPL 和 B 语言的精练、能够接触计算机硬件的优点又恢复了这些语言失去的通用性。

70 年代中, UNIX 系统以及它所支持的最主要语言 C 在贝尔实验室内部和大学中得到了普遍应用。70 年代末 80 年代初, UNIX 系统已被多数人认为将成为第一个标准操作系统, 而 C 语言的用户也日益扩大, 进入了少数几种应用最广泛的语言的行列。C 语言的成功与它产生的环境及历史背景有关, 但起决定作用的是它自身的特点。简而言之, C 语言的特点是: 简洁、灵活, 表达能力强, 产生的目标代码质量高, 可移植性好。稍微具体一些, C 语言的特点可以分成如下几点:

(1) C 语言的基本组成部分紧凑、简洁, 使用一些简单、规

整的方法就可以构造出相当复杂的结构。

(2) 表达式简练、灵活、实用,既提高了编译效率和目标代码的质量,又提高了程序的可读性。

(3) 提供了某些接近于汇编语言的功能,如地址处理、二进制数位运算以及指定用寄存器存放变量等。所以,C语言适合于编写系统程序和各种软件工具。

(4) 具有编写结构良好的程序所需要的各种控制流结构,如两路、多路选择语句,for、while、do_while循环语句等。C语言是一种结构化程序设计语言。

(5) 具有各种现代程序设计语言普遍配置的数据结构,如结构、联合、位段等。

(6) 提供了与地址密切相关的指针以及有关运算符。指针可以指向各种类型的简单变量、函数、数组和结构等。

(7) 为字符和字符串处理提供了良好基础。C语言能够表示和识别各种能够直接显示的以及起控制作用的字符,能够区分单个字符和字符串等。

(8) 预处理程序和预处理语句提高了程序的可读性、可移植性,给程序调试提供了方便。C语言支持模块化程序设计方法,便于开发大型程序。

(9) 输入/输出(I/O)依靠函数调用实施,C语言的标准程序库具有多种使用方便、功能强的I/O函数。

(10) 生成的目标代码质量高。经验表明,针对同一问题,如果用C语言和汇编语言分别编写程序,那么前者生成的目标代码的运行效率仅比后者的低10%至20%。

(11) 用C语言编写的程序具有特别良好的可移植性。移植UNIX操作系统和C编译程序的广泛实践已经清楚地证明了这一点。

当然,和其它程序设计语言一样,C语言也有其弱点,如运算符的优先级较多,不容易记忆,某些语法部分不易用形式化方法进行描述,目前使用的C语言版本有若干种,相互之间略有差别,缺乏统一的标准等。另外,C语言注重程序的灵活性和运行效率,在一定程度上这是以牺牲某些安全性而得到的。因此,对C语言的程序设计员提出了较高的要求,特别是在使用C语言的某些高级手段时更是如此。但是,C语言的优点远超过它的弱点,这些优点使C语言具有强大的吸引力。经验表明,程序设计人员一旦接触了这种语言并且具有了一定应用经验后就会对它爱不释手。

二、开发C语言程序的基本过程

用C语言开发程序的基本过程如图1-2所示。

在系统帮助下,开发C语言程序的第一步是编辑,也就是使用系统提供的编辑程序(例如UNIX系统的文本行编辑程序ed、屏幕编辑程序vi)将以C语言编写的程序以文件形式存入文件系统,文件名可以任取,不过为了帮助记忆和使用,文件名最好与程序功能有一定联系。另外,在UNIX系统下,C程序文件名的最后两个字符应为.c,所以,下列C程序文件名都是合法的:

```
program1.c
project_2.c
test_prog.c
l_7.c
```

用C语言编写的程序称为源程序,它一旦以文件形式存放在文件系统中,就可以调用C语言编译程序对它进行编译,在UNIX系统中,这是通过使用cc命令实施的,例如:

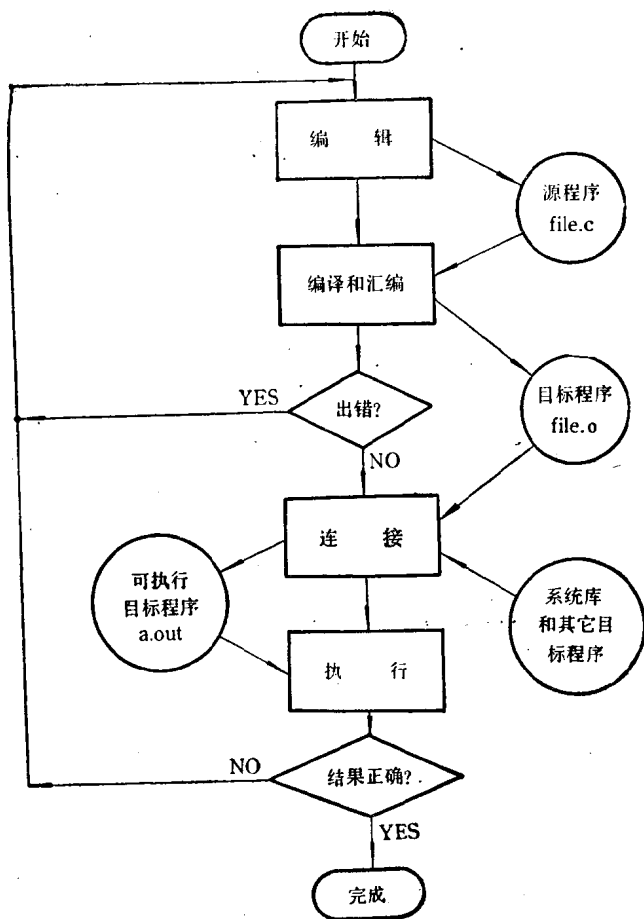


图 1-2 用 C 语言开发程序的基本过程

```
cc program1.c
```

即要求系统对文件 `program1.c` 中的 C 源程序进行编译处理。

在编译过程中如果发现源程序在词法、语法、语义等方面有错，那么编译程序向用户报告这些错误并适时结束编译过程。然后用户再使用编辑程序修改源程序中的错误，接着重新启动编译过程。如此不断反复，直至排除了源程序中所有语法错误。

编译程序对一个语法上正确的源程序进行处理的结果是构成一个功能相同的汇编语言形式的程序。编译过程的下一步是调用汇编处理程序将汇编程序翻译成机器指令形式的程序，这种程序称为目标程序。在 UNIX 系统中，汇编处理程序调用是由 `cc` 命令执行程序自动进行的，在另外一些系统中，用户可能需要使用系统提供的另一条命令。

汇编处理程序将汇编程序转换成目标程序，并将它存放到另一个文件中。在 UNIX 系统下，目标程序文件名与源程序文件名除最后一个字符外其余字符相同，其最后两个字符为 `.o`。例如，对 `program1.c` 进行编译和汇编处理后形成的目标码文件名为 `program1.o`。

汇编处理结束后，编译过程进入第三步——连接。在 UNIX 系统下，这同样是由 `cc` 命令执行程序自动进行的，在其它系统中用户可能需要使用另外一条命令。连接的目的是构成在计算机上可以立即执行的目标程序。连接的对象除上一步刚刚获得的目标程序外，还可以包括另外一些以前编译好的目标程序或者目标码形式的系统库程序。

连接程序构成的可执行目标程序存放在另一个文件中。在 UNIX 系统下，该文件名可以在调用 `cc` 命令时指定，例如：

```
cc -o program 1. out program 1. c
```

将可执行目标程序文件名指定为 `program1.out`，如果在使用 `cc` 命令时没有指定，那么将它取为缺省名 `a.out`。如果希望执行该程序，那么只需要使用可执行目标程序文件名作为命令名，例如：

`a.out`

在执行这条命令时，系统将 `a.out` 文件中的可执行目标码装入内存并分配一定的数据区和工作区，然后启动执行该程序。

程序在执行时如果没有产生预期的结果，就应当检查程序的逻辑，包括使用的算法、数据结构以及程序的组织等。这种检查过程称为排错。在进行排错处理时，通常需要修改源程序，于是编辑、编译、汇编、连接、装入和执行过程再次反复，直至程序提供了各种预期功能，整个 C 语言程序的开发过程才告一段落。

三、用 C 语言开始编写程序

学习一个新的程序设计语言时，最好先读几个用这种语言编写的程序，让我们从一个非常简单的程序开始。

程序 1-1

```
main( )
{
    printf( "Programming is fun.\n" );
}
```

如果将这段程序输入计算机、对其进行编译、最后装入执行，那么在终端上将显示：

Programming is fun.

通过这段程序读者可以了解到下列几点：

(1) 最简单的 C 语言程序只包括一个函数，函数名为 `main`。一般而言，一个 C 语言程序可以包括若干个函数，但是

其中应该有一个也只应有一个以 `main` 命名的函数。程序总是从 `main` 函数开始执行。函数名后的一对括号中可以包含若干个参数。本例中, `main` 函数不带任何参数, 但括号组仍须保留。

(2) 紧接着的一对花括号组相当于 PASCAL 语言中的 `BEGIN` 和 `END`, 分别表示程序的开始和结束, 其中包含的语句提供程序的预定功能。在程序 1-1 中只包含一条语句, 该语句调用名为 `printf` 的标准系统库函数。

(3) `printf` 函数的作用是进行格式化打印。但在这里其作用仅是将参数字符串

```
"Programming is fun. \n"
```

在终端上显示或打印出来。

(4) 参数字符串中的最末两个字符 `\` 和 `n` 合并起来表示终端控制字符‘新行’。在新行符后的任何字符将显示在终端的下一行上。

(5) 在 `printf` 的右括号后立即出现一个分号, 它表示语句终止。

(6) 考虑到人们读、写英语的习惯, C 语言主要使用小写英文字母, 这样就显著改善了程序的可读性。

(7) 在书写和显示格式上, C 语言并不计较每行的起始位置, 也就是说在每一行的任何位置上都可以输入语句。这一点有助于清楚地显示程序的结构特性, 因而大大地提高了程序的可读性。本书包含的大量示例程序都注意到了它们的书写和显示格式, 请读者在编写 C 语言程序时也同样对此引起足够重视, 并养成良好习惯。

在程序 1-1 的基础上略加扩充构成了程序 1-2。

程序 1-2

```
main( )
{
    printf( "Programming is fun.\n" );
    printf( "And programming in C is even
more fun.\n" );
    printf( "Right!\nRight!!\nRight!!!\n");
}
```

输出:

```
Programming is fun.
And programming in C is even more fun.
Right!
Right!!
Right!!!
```

程序 1-2 中包含了三条 printf 函数调用语句。前两条语句的作用是在终端上各显示一行(见输出中的第一、二行),第三条语句则在终端上显示出三行(见输出中的第三、四、五行)。可见,在 printf 参数字符串中只要包含若干个新行符(\n),一次 printf 函数调用就可以显示若干行。

printf 函数不仅可以显示字符串,也可以按照格式说明显示其它参数的值和计算结果,程序 1-3 使用了 printf 函数的这种功能。

程序 1-3

```
/* This program adds two integer values
and display the results.
*/
main( )
{
    /* Declare variables */
    int  value1, value2, sum;
```