

“C”

〔日〕 椋田 美 著

# 语言入门



机械工业出版社

# “C”语言入门

〔日〕 棕田 実 著  
徐东安 译



机械工业出版社

C语言是70年代发展起来的一种通用程序设计语言，与其它计算机语言比较，它有经济实用的表达式、先进的控制流和数据结构、丰富的运算符，很适于描述操作系统、编译程序及软件工具。C语言的程序与PASCAL语言一样易于阅读，但在空间、时间和效率上却可与汇编语言相媲美。用它编写的程序结构紧凑、清晰，其编译程序简单、好写，编译出的目标程序质量好。它可和汇编语言一样有效地书写系统软件，用它可描述操作系统中诸如设备驱动程序这样的低级程序。C语言又是独立于任何一台机器的语言，有高度的机动性。它的代码80%是公共的，很便于移植。因此C语言目前已成为微型机的一种专用语言，对于小型机、超小型机、大型机及超大型机均可使用。

本书是入门读物，以初学C语言的读者为对象，目的是使读者习惯于C语言的符号和描述方法，通过书中105段由计算机执行过并打印出来的程序，具体掌握理解C语言的语法问题。

75756/102  
はじめての“C”

昭和60年1月20日 初版第1刷発行

著者 椋田 実

発行者 片岡 巖

発行所 株式会社 技術評論社

\* \* \*  
“C” 语言入门

〔日〕 椋田 実 著

徐东安 译

\*

机械工业出版社出版（北京阜成门外百万庄南里一号）

（北京市书刊出版业营业登记证出字第117号）

机械工业出版社印刷厂印刷

新华书店北京发行所发行·新华书店经售

\*

开本 850×1168<sup>1</sup>/<sub>32</sub>·印张 7<sup>0</sup>/<sub>4</sub>·字数 258 千字  
1988年8月北京第一版·1988年8月北京第一次印刷

印数 0,001—5,700·定价：3.25 元

\*

ISBN 7-111-00429-9/H·7

## 译者的话

1985年在北京机床研究所建成了我国第一个柔性制造系统(FMS)——电机制造管理系统。笔者有幸参与了整个过程的技术工作,历时两年,深感资料不足。经日本朋友的推荐,翻译了《C语言入门》一书,为学习C语言,掌握、消化、吸收电机制造管理系统的软件起了一定的作用。

1986年以该译文为基础,重新作了修正、整理,遂成此书。如果本书对读者学习C语言有所帮助,将是莫大的安慰。笔者学识肤浅,错误之处难免,恳请读者批评指正。

译者 1986年冬于北京

# 前 言

C语言是D. Ritchie为小型计算机操作系统UNIX而开发的程序设计语言，但目前从通用计算机直到个人计算机不论规模大小都在使用它。特别是在八位、十六位个人计算机方面，以程序设计语言C(The C programming Language——Kernighan, Ritchie 合著)为标准的C编译程序正由若干公司在市场上出售。可以认为，从C语言的简便性、实用性看，今后C语言将会得到广泛的应用。

对于准备学习C语言的读者，是否会觉得，与其它程序设计语言相比，它是“太符号化的古怪的语言”呢，再有，指针(C语言特点之一)一旦出现，是否又会想：“固然简便，但却是难读的程序”。不过，如果仔细阅读C程序就会明白，同已往的BASIC、FORTRAN、PASCAL语言一样，它是过程语言。其算符种类多，光凭符号就能阅读。如果习惯了，用C语言可以编制非常容易阅读的程序。

本书是入门读物，是以初学C语言的读者为对象的。关于C语言的详细情况，请参阅《语言手册》和《程序设计语言C(K&R著)》。本书的目标首先是让读者习惯C语言的符号和描述方法。为此，在编排方面举了许多例题，以便读者能够通过程序去理解语法。

学习任何一种语言，可以说最重要的是习惯它。所谓习惯C语言，就是逐渐熟悉C语言的形式。预先了解这一点，就已经打下学习C语言的基础。光靠思维理解还不能编制程序，为了做到能够使用语言，需要直观的理解。对于编排程序的算法必须深思熟虑，这就是所谓“语言由感官感觉，程序则用思维编制”。如果手头有台个人计算机的话，一边阅读本书一边熟悉例题的各项功能，这样就最为理想了。

本书第一章安排了概论，以掌握C语言的全貌。请初学C语言的读者无论如何读完第一章。通过这一步，理应达到具备阅读以后各章的能力。从第二章开始分别介绍数据类型、运算、控制指令、数组、指针、标准程序库函数等等。

作者期望，本书将对初学C语言的读者有所帮助。

最后，对于出版本书的技术评论社第二编辑部各位人士表示感谢。

椋田 実 1984年12月

## 〔书中用到的说明符号和注意事项〕

1. \

(5C<sub>(16)</sub>)

这是ASCII代码的符号，叫做反斜线，在JIS中相当于¥。日本的个人计算机几乎都采用JIS代码，所以\则为¥。

2. □

这是说明上的符号，表示一个字符的空白。只在希望强调空白时使用。

3. A2<sub>(16)</sub>

在英文字母数字的右下方，写上带括号的数字，它表示用什么进制来标记。例如，A2<sub>(16)</sub>表示16进制标记。

4. K & R

表示B. W. Kernighan和D. M. Ritchie著的〔The C Programming Language〕。

5. ^A, CTRL + A

01<sub>(16)</sub>

在按下控制·位移键的条件下，按A键。例如，^A→01<sub>(16)</sub>，  
^B→02<sub>(16)</sub>，……^Z→1A<sub>(16)</sub>。

# 目 录

第1章 C语言的内容、特征 .....	1
1.1 什么叫C语言 .....	2
用C语言编程 .....	2
1.2 书写明白易懂的程序 .....	8
C语言的编程式样 .....	8
1.3 如何对数据类型进行运算 .....	14
C语言数据类型的表达和运算 .....	14
1.4 改变命令的执行顺序 .....	26
C程序的控制流程 .....	26
1.5 扩展C功能的程序 .....	31
预处理程序 .....	31
1.6 承担C语言输入输出功能的程序 .....	36
标准程序库函数的作用 .....	36
第2章 数据类型 .....	41
2.1 数据的四种类型 .....	42
C语言中使用的数据基本类型 .....	42
2.2 在变量上说明存储分类 .....	47
变量和存储分类 .....	47
2.3 变量的有效范围有多大 .....	50
变量的存储分类和有效范围 .....	50
2.4 直接表达实际的值 .....	54
常量的表示方法 .....	54
2.5 给变量、数组赋值 .....	58
变量和数组的初始化 .....	58
第3章 算符 .....	61
3.1 用C语言进行加减乘除 .....	62
C语言的算术算符 .....	62
3.2 用C语言进行比较、逻辑运算 .....	64

关系运算符和逻辑运算符	64
3.3 加1减1运算	67
增量减量算符	67
3.4 直接对位进行操作	70
位运算符和移位算符	70
3.5 指示地址或者指示其内容	75
地址算符和间接算符	75
3.6 C语言的赋值算符、条件算符及其它	78
赋值算符、cast算符, 顺次算符, 条件算符、sizeof算符	78
3.7 运算之前确立的规则	82
算符的优先次序和结合规则	82
第4章 控制流	85
4.1 让控制流转移	86
由if-else语句实现的条件转移	86
4.2 重复执行相同处理·1	91
利用while语句执行重复处理	91
4.3 重复执行相同处理·2	94
利用for语句执行重复处理	94
4.4 根据条件选择分支入口	98
由switch-case语句选择分支入口	98
4.5 重复执行相同处理·3	102
利用do-while语句执行重复处理	102
4.6 跳出循环控制	104
break语句和continue语句	104
4.7 C语言的无条件转移指令	107
由goto语句实现无条件转移	107
第5章 函数	111
5.1 C语言中所使用的函数的意义	112
C程序的构成单位——函数	112
5.2 说明函数的类型	117
函数的类型和存储分类	117
5.3 函数间数据的来往	119



数据的交接方法 .....	119
5.4 将函数本身作为函数而加以调用 .....	121
C语言的递归调用 .....	121
第6章 数组与指针 .....	125
6.1 将数据存放在数组上 .....	126
C语言中所使用的数组 .....	126
6.2 将字符集存放在数组上 .....	131
处理字符型数据的数组 .....	131
6.3 自由地操作数组的数据 .....	134
指针和数组 .....	134
6.4 函数间指针的来往 .....	138
指针和函数 .....	138
6.5 存放指针的数组 .....	141
指针数组 .....	141
第7章 结构和联合 .....	147
7.1 数据的结构化便于操作 .....	148
数据的集合——结构 .....	148
7.2 为了引用结构的成员 .....	153
结构成员的引用和指针 .....	153
7.3 预先将值存入结构 .....	157
结构的初始值设定 .....	157
7.4 使用指针将数据传送给函数 .....	160
结构和函数 .....	160
7.5 往同一地址分配不同的数据 .....	162
联合的说明 .....	162
第8章 输入输出函数 .....	169
8.1 从C程序调用的输入输出函数 .....	170
标准程序库的输入输出函数 .....	170
8.2 对数值、字符串进行变换输出 .....	172
Printf函数 .....	172
8.3 对数字、字符拼字进行变换输入 .....	181
Scanf函数 .....	181
8.4 从标准输入读入一个字符 .....	187

## VII

Getchar 函数 .....	187
8.5 向标准输出上输出一个字符 .....	191
Putchar 函数 .....	191
8.6 不通过标准输入输出的输入输出函数 .....	194
Fopen函数和Fclose 函数 .....	194
8.7 逐个字符输入输出文件的函数 .....	197
Getc, Putc函数 .....	197
8.8 对字符串进行操作的函数 .....	202
strcat, strcpy, strlen, strcmp 函数 .....	202
8.9 检查字符分类的函数 .....	207
Isalpha, Isdigit, Isupper, Isspace函数 .....	207
8.10 对英文大写字母和小写字母进行变换的函数 .....	211
tolower函数和toupper函数 .....	211
第9章 低级输入输出函数 .....	213
9.1 依存于操作系统的低级输入输出函数 .....	214
CP/M, MS-DOS用的输入输出函数 .....	214
9.2 做到能够读写文件 .....	215
Open函数, Close 函数, Create函数 .....	215
9.3 在文件上写入数据 .....	220
Write 函数 .....	220
9.4 读出文件上的数据 .....	223
Read 函数 .....	223
9.5 随机地读写文件 .....	226
Lseek 函数 .....	226
附录 .....	231
附录 A .....	231
附录 B .....	233
程序索引 .....	234

## 第 1 章

# C 语言的内容、特征

# 1.1 什么叫C语言

## 用C语言编程

### C语言是通用的编程语言

编程语言可以说是一种特别的语言，它是让计算机完成某项工作的作业指示。用这种语言书写的作业步骤清单就叫做程序，而编制作业步骤清单就叫做编程。

针对不同的作业目的，目前已开发了若干种编程语言，诸如用于事务处理方面的编程语言，用于科学技术计算方面的编程语言，用于描述系统方面的编程语言，用于符号处理方面的编程语言等等。

用于事务处理方面的编程语言，其特点在于文件处理的记述性，数值的表达（数值的1、2、3与字符串的“1、2、3”是相同的），典型的事务处理语言有COBOL语言。

用于科学技术计算方面的编程语言，其特点在于数学上的记述性和运算能力，数值的内部表达在整数和实数方面是不同的。这类语言有FORTRAN语言、BASIC语言。

用于符号处理方面的编程语言，有LISP语言、PROLOG语言。

最近，用于教育的Pascal语言取代了FORTRAN语言，从Pascal语言的数据结构、程序结构来看，它也可以用于系统描述。换句话说，Pascal语言是冲破了已往的编程语言框框的语言。

C语言当初是作为系统描述用的语言而研制的<sup>⊖</sup>，它不单能够编制通常用汇编语言书写的系统软件，这种语言也适用于事务处理、科学技术计算应用软件开发方面。也就是说，C语言也是超出了已往的编程语言框框的语言。

C语言应用范围非常广泛的理由如下：

- (1) C语言的指令接近硬件，具有较低级的功能。
- (2) 在数据结构方面，可以采取结构<sup>⊖</sup>、联合<sup>⊕</sup>。
- (3) 程序结构是以函数为基本的集合体。

⊖ 它是为了描述UNIX这种操作系统而研制的。

⊕ 关于特定内容的信息集合体。

⊖ 在同一地址上，可以分配不同类型变量的变量说明（参照第7章7.5节）。

(4) 就象黑盒一样, 可以把文件的输入输出、字符串操作、算术运算等功能作为一个函数来对待。

(5) 由于有指针变量 $\ominus$  (指针变量是间接地指示存储分类——自动变量以及数据的), 因此可以对变量区进行柔性的动态管理。

所谓自动变量, 就是在函数中所说明的变量, 仅在该函数执行期间有效, 一旦执行完毕它就失效。因此, 在没有必要时, 自动变量完全不占存储空间, 而数据区则得到充分有效的利用。

C语言的数据类型有整数型(int)、实数型(float)和字符型(char), 其中整数型和实数型是存放数值的, 而字符型则是存放字符串的, 通过它们的组合, 可以构成更加复杂的数据类型。在程序中, 对若干个数据类型处理数据的地址的, 便是指针变量。对稍微了解汇编语言的读者来说, 所谓指针就是既具有汇编语言中的间接寻址功能, 又具有考虑数据类型的变址寻址功能的东西, 这样解释大概就会明白了。

下面, 用C语言写一段简单的程序加以说明。

输出“C programming!!”

让我们编写一段程序, 输出“C programming!!”这些字符。

```
main()
{
    printf("C programming !!\n");
}
```

↑  
—在ASCII代码中是\n, 而在  
JIS代码中则为¥n.

main ( ) 这是主程序的函数名。C语言的程序是由各种功能的函数构成的。此外, 由于从一个函数可以调用别的函数, 因此, 作为函数的集合体可以构成更大的程序。

关于函数名, 虽然可以自由地起自己喜爱的名字, 但是, main ( ) 这个名字是特别的, 是从操作系统转向控制的函数。main 的后面 ( ) 表明, 参数是空的。例如, 即使没有参数, ( ) 也是必要的。大括号所括住的部分 ({...}) 则记述处理步骤。

printf是输出字符的函数, 在这个例子中输出“C programming!!”。所有的函数都采取下列的形式:

函数名 (参数表) 参数的类型说明 {......}

$\ominus$  它决定有效范围, 即变量什么时候使用, 存储在什么地方 (参照第2章2.2节)。

如果对上述那段输出字符的程序进行编译并加以执行，那么，将打印出下列字符，

```
C programming!
```

\n (在JIS代码中相应为¥n) 在同一个括号内，但是却没有打印输出，这是因为使用\表示控制字符的缘故。在输出了“C programming !!”之后，\n进行换行。

在下列程序的场合，则输出如下，

```
main()
{
    printf("C programming !!\n");
    printf("C prog");
    printf("ramming !!\n");
}
```

← 这里没有\n，所以不换行

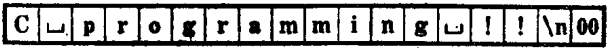
(执行结果)

```
C programming !!
C programming !!
```

← 这个字符串由第二、第三行 printf 语句输出

printf (……) 叫做输出字符的函数，但是，这个函数作为标准库函数从一开始就同C编译程序一起提供的。在C语言中由于没有输入输出功能，所以它是这样准备函数的。

printf函数的参数，在上例的情况下是“C programming !! \n”，而由双引号 (“ ”) 括起来的字符串叫做字符串常数。一旦由双引号括起来，便自动地在末尾附加上一个字的空代码 (null code)，这个空代码是16进代码的00 (16)，所以，C语言中的字符串是指字符串的最后有空代码的。



使用刚才出现过的\ (在JIS代码中为¥) 也可以表达空代码。

```
\0
```

这是在反斜线上外加0。这个反斜线叫做换码符。但是，它与通信控制代码的ESC (换码代码，1B<sub>(16)</sub>) 是不同的。换码符是对C编译程序的指示码。

### C程序编程上的规定

所谓编程就是记述程序。同其它语言 (例如FORTRAN) 一样，在程序

的书写方法上C语言也有某些规定，现分述于后。

### (1) 自由格式

语句分隔符有空白，列表，换行，注解。每个语句都使用分隔符并没有什么特殊意义，但是在单词的拼写之间不许用分隔符。只有在单词同单词之间用作分隔时才有意义。

例

`printf("C programming !! \n");` 这个语句虽然可以分解为 `[printf]`, `[ (]`, `[ "C programming !! \n" ]`, `[ ) ]` 和 `;`, 但是分隔符还未用在相应的各部分上，下面在相应部分加入分隔符。

```

main()
{
printf
" C programming !! \n"
;
}

```

·  是回车键  
· 由列表键 **TAB** 右移四格

加入了分隔符并改写了程序，这在语法上也没有问题。不过，如果将 `printf` 分隔成 `prin` 和 `tf` 两部分，那么将被解释成两个单词，因而语法上便错了。不过，由双引号括起来的部分，空白也是有意义的，它是作为字符串的一部分。

### (2) 注解由 `/*` 和 `*/` 括起来

注解可以写若干行，注解语句由 `/*` 开始，而以 `*/` 结束。

```

开始 → /*
          *
          *      xxxxxxxxxxx program
          *      file name : zzzzzz.c } 若干行注解
结束 → */

main()          /* function : yyyyyyy */ 输入的
{
printf("ABCDEFG\n");
}

```

### (3) 函数名，变量名是由英文字母或下划线开始的字母数字串

函数名、变量名必须是由英文字母（“A”～“Z”，“a”～“z”）或下划线（“\_”）开始的字母数字串。长度是自由的，但是C编译程序上的有效长度为8个字符。下划线的操作同英文字母一样。

灵活运用大写英文字母和小写英文字母，可以使程序一目了然。大写字母可以用于置换宏名（由`#define`指定的字缀），而小写字母可以用于通常的变量，函数名。

例

```
#define CTRLA 0x01
#define CTRLP 0x10 } 置换宏的指示

main()
{
    char c1 = CTRLA, c2 = CTRLP;
    while(c1 != CTRLP) {
        putchar('^');
        moji20(c1); c1++;

        putchar(' '); putchar(' '); putchar('^');
        moji20(c2); c2++;
    }
    printf("\n");
}
```

由于预处理程序，被置换成由`#define`所定义的字符。在本例中是“0x10”。

▷ `putchar, moji, printf` 是函数名

(4) 程序语句由分号(;)分隔

一个语句可以用分号来分隔，但在别的意义上也可以使用分号。

例

```
bitdsp(cc) /* bit display */
char cc;
{
    int i;
    char aschex();
    for(i=7; i>=0; i--)
        putchar(aschex((cc>>i)&0x01));
}
```

函数名      参数表  
参数类型说明  
内部变量的类型说明  
函数的类型说明  
for 语句

- 分号; 并不是for语句的结束。
- ; 表示语句的结束，同时也表示跨越两行的for语句的结束。

例如，在下列场合的|; |并不是for语句的结束。

```
for (式1; 式2; 式3)
```

(5) 保留字不能用于函数名、变量名

保留字（编译时的关键字）开列如下，它们不能用作函数名、变量名、数组名。



int	short	goto	for
char	unsigned	return	do
float	auto	sizeof	while
double	extern	break	switch
struct	register	continue	case
union	typedef	if	default
long	static	else	entry

(根据Kernighan & Ritchie)