

微型计算机IBM-PC丛书

IBM-FORTRAN 语言

朱耀庭 译
翁瑞琪 校



南开大学出版社

微型计算机 IBM-PC 丛书

IBM FORTRAN语言

朱耀庭 译
翁瑞琪 校

南开大学出版社

IBM FORTRAN 语言

朱耀庭 译 翁瑞琪 校

南 开 大 学 出 版 社 出 版

天津八里台南开大学校内

新 华 书 店 天 津 发 行 所 发 行

天 津 宝 坻 县 牛 家 牌 印 刷 厂 印 刷

1984年12月第1版 1984年12月第1次印刷

开本850×1168毫米1/32 印张6

字数: 151 千字 印数: 20,000

统一书号: 13301·7 定价: 1.44 元

译 者 前 言

本书是根据 IBM-PC 个人计算机的计算机语言丛书之一《FORTRAN Compiler》(Fortran 编译程序)一书(1982 年第一版)翻译的。本书是在 IBM-PC 个人计算机上使用 Fortran 语言算题的必读手册。

全书共分六章。第一章介绍表示法的几点约定、Fortran 程序结构、数据类型、表达式与 Fortran 名。第二章描述编译、连接、执行 Fortran 程序的机理。第三章介绍处理 Fortran 源文本的一些编译元命令。第四章介绍各种语句。第五章描述 Fortran 输入输出系统,并包括基本的 Fortran 输入输出概念和格式语句。第六章介绍 Fortran 程序中可用的内部函数。书末附有附录。附录 A 列出了机器给出的错误信息表。附录 B 描述 IBM Fortran 与标准子集语言的不同之处。附录 C 提供把程序进行连接方面的信息。附录 D 还介绍了如何用不同编译程序包来连接各目标模块。附录 E 给出了在 IBM-PC 上输入、执行、修正、再执行和运行一个 Fortran 程序的例子。

本书由朱耀庭译,翁瑞琪校。由于时间仓促且译校者水平有限,错误在所难免,殷切期望读者提出批评与指正。

184-24/59

目 录

第一章 前言	1
一、表示法的几点约定	1
二、FORTRAN 程序结构	2
三、数据类型	7
四、表达式	10
五、FORTRAN 名	17
第二章 FORTRAN程序的编译	21
一、准备	21
二、开始编译	23
三、设备标识	34
四、编译程序列表	35
第三章 编译元命令	45
一、概述	45
二、\$DEBUG 元命令	45
三、\$DO 66 元命令	46
四、\$INCLUDE 元命令	47
五、\$LINESIZE 元命令	49
六、\$LIST 元命令	49
七、\$NODEBUG 元命令	49
八、\$NOLIST 元命令	50
九、\$PAGE 元命令	50
十、\$PAGESIZE 元命令	50

十 一、\$STORAGE 元命令	51
十 二、\$SUBTITLE 元命令	51
十 三、\$TITLE 元命令	52
第四章 语句	53
一、控制语句	53
二、PROGRAM、FUNCTION 和SUBROUTINE 语句.....	56
三、I/O (输入/输出) 语句	59
四、说明语句	61
五、算术 IF 语句.....	62
六、赋值语句	63
七、标号赋值语句	64
八、赋值 GOTO 语句.....	65
九、BACKSPACE (回退) 语句	66
十、Block IF 语句	67
十 一、CALL (调用) 语句	68
十 二、CLOSE (关闭) 语句	69
十 三、COMMON (公用) 语句	70
十 四、计算 GOTO 语句	71
十 五、CONTINUE (继续) 语句	72
十 六、DATA (数据) 语句	73
十 七、DIMENSION (维数) 语句	74
十 八、DO 语句 (循环语句)	75
十 九、ELSE 语句	78
二 十、ELSEIF 语句	78
二十一、END 语句.....	79
二十二、ENDFILE 语句	80
二十三、ENDIF 语句	80

二十四、EQUIVALENCE (等价) 语句	81
二十五、EXTERNAL (外部) 语句	83
二十六、FUNCTION (函数) 语句	84
二十七、IMPLICIT 语句(隐含说明语句)	85
二十八、INTRINSIC 语句(内部语句)	87
二十九、逻辑 IF 语句	87
三十、OPEN (打开) 语句	88
三十一、PAUSE (暂停) 语句	92
三十二、PROGRAM 语句	92
三十三、READ (读)语句	93
三十四、RETURN (返回) 语句	94
三十五、REWIND (反绕) 语句	95
三十六、SAVE (保存) 语句	95
三十七、语句函数	96
三十八、STOP 语句	97
三十九、SUBROUTINE (子例程) 语句	98
四十、类型语句	99
四十一、无条件 GOTO 语句	100
四十二、WRITE (写) 语句	101
第五章 I/O 系统	104
一、概述	104
二、记录	104
三、文件	105
四、有格式 I/O 和 FORMAT 语句	112
五、输入/输出表的作用和格式说明	114
六、编辑描述符	115
七、托架控制	120
第六章 内部函数	121

附录A 关于错误信息	125
一、编译时错误信息	125
二、文件系统错误	134
三、其它运行时错误	138
附录B IBM FORTRAN与ANSI FORTRAN 77 的区别	141
附录C 连接程序	144
一、引言	144
二、文件	144
三、定义	145
四、命令提示	147
五、命令提示的详细说明	148
六、怎样启动连接程序	152
附录D 目标模块的连接	163
附录E 连接对话示例	172
专业术语 GLOSSARY	181

第一章 前 言

IBM 个人计算机上的 Fortran 与标准 ANSI X3.9—1978 (FORTRAN—77) 的一个子集相当, 在这本手册中称为 IBM Fortran, 它也包括完全 ANSI X3.9—1978 级的一些特征。

可以看出, IBM Fortran 已被增强, 以致易于对现有的 Fortran 66 程序进行转换。例如: Fortran 66 中的 DO 循环的语义是由编译程序选择的。

IBM-PC 的连接程序允许将 Fortran 目标模块与其它语言的目标模块结合在一起, 如与 IBM-PC 的宏汇编程序和 IBM-PC 的 Pascal 目标模块均可结合, 使需要用不同语言来书写程序不同部分的用法得以实现。

IBM Fortran 由 Fortran 编译程序和目标模块库构成, 该目标模块建立 Fortran 运行时的库。建立一个可执行 Fortran 程序的第一步是编译它的 (一个或多个) 源程序模块, 编译的结果形成目标模块, 它与 Fortran 运行库相连接后产生一个运行文件, 这一运行文件, 可以为 IBM-PC 的 DOS 调用。

和包括 IBM Fortran 编译程序的输出一样, 被连接的目标模块和库可以包括 IBM-PC 宏汇编程序和 IBM-PC Pascal 编译程序的输出。

一、表示法的几点约定

本手册上所用的表示法的约定如下:

- 出现在程序中的是大写字母和专用字符。(如: CONTINUE.)

- 小写的斜体字母和字是应在正文中描述的实际语句中提供的项。一旦一个小写项被定义, 它将在该程序的整个论述中保持其含义。(如: PROGRAM pname.)

例如, 描述整数的编辑的格式中的大写和小写字母表示为 Iw, 其中 w 是一个非零无符整常数。

于是, 在一个实际语句中, 一个程序可以包含 I 3 或 I 44。描述实数的编辑格式是 Fw.d, 这里 d 是一个无符整常数, 在实际语句中 F 7.4 或 F 22.0 是正确的。注意, 要照样采用作为专用字符的点号。

- 方括号中是可选择的项, 例如 A[w] 表明 A 或 A 12 (作为指定的一个字符格式) 都是允许的。

- 省略号 (...) 指出省略号前面可选择的项可以出现一次以上。

例如, 下述的计算 GOTO 语句表明, 用 s 表示的、用逗号隔开的语法项可以重复任意多次。

GOTO (s[, s]...)[,] i

- 空格在 FORTRAN 语句的描述中通常没有任何含义, 在本章“FORTRAN 程序结构”一节中描述的关于空格的一般规则给出空格的解释。

二、FORTRAN 程序结构

1. 字符集

一个 Fortran 源程序包含一系列字符, 这些字符是:

- 字母: A—Z 的 52 个大写和小写字母。

• 数字：0、1、2、3、4、5、6、7、8、9。

• 专用字符：ASCII 字符集中其余的可打印字符。

除了字符串常数和 Hollerith 字段，对一切上下文中的小写字母，Fortran 均作为大写字母来解释。因此，下面的用户定义名在 IBM FORTRAN 系统中不加区分。

ABCDE abcde AbCdE aBcDe

IBM Fortran 字符集的排序序列为 ASCII 序列。

2. 行

一个 Fortran 源程序也可以看作是行的序列，在一行中仅前面 72 个字符为编译程序认为是有效的。编译程序将忽略第 72 列以后的所有字符。

如果一行少于 72 个字符，编译程序把其长度看作是有效的（对这些的解释见本章“字符表达式”中字符常数的描述）。

3. 列

在一已知行中的字符被分为列。例如，第一个字符是在第 1 列，第二个字符在第 2 列，依此类推。

一标记字符可以放在一行的第 1—6 列，它导致下一字符被解释为在第 7 列，这个标记在列表文件中将被扩展到适当的空格数，所有其它标记被传递，如同传递到列表文件一样。

列	用 途
1—5	语句标号和注释指示符
6	继续指示符
7—72	源语句

4. 初始行

初始行是在第 6 列为一空格或一个 0 的任意行，并且它不是一个注释行或一个元命令行。这行的前五列是全部空格或包含一个标号。除去跟在逻辑 IF 语句之后的语句，Fortran 语句以一个初始行开始。

注：当在初始行之后使用续行时，在初始行继续列上使用一个 0 来改善可读性。

例：

```
0 IF ((A. LT. 0). AND. (B. LT. 0)
1 .AND. (C. LT. 0))
2 THEN
```

5. 空格

除了下面指出的空格以外，在 Fortran 源程序中空格是无效的，它可用来改善 Fortran 程序的可读性。

例外的情形是：

- 字符串常数中的空格。
- Hollerith 字段中的空格。
- 在第 6 列的空格，它区分初始行与续行。

6. 注释行

注释行对 Fortran 程序的执行无任何影响。

遇下面任一情况，一个行被当作一注释行。

- 第 1 列出现一个“C”（或“c”）。
- 第 1 列出现一个“*”。
- 一行全部为空格。

注释行后必须紧跟一个初始行或另一注释行，绝对不能跟一个续行。

注：在一个 Fortran 程序结尾的额外的空行导致编译时错误，因为 Fortran 把它们解释为注释行，尽管其后未跟一个初始行。

7. 标号

一个语句标号是一个 1—5 位数字的序列，在各程序单位中它是唯一的。至少一位数为非 0。一个标号可以放在一个初始行的第 1—5 列的任何位置。空格和先行 0 无效。

8. 续行

一个续行可以是非注释行或元命令行的任意行,它在第6列是除空格或0以外的任何字符,续行的前5列必须是空格。

续行提供了书写一个语句的更大空间,如果一个语句在单个初始行写不下时,最多可以扩展到九个续行。

9. 语句

一个Fortran语句包含一个初始行和最多九个续行。语句的字符是在这些行的第7列到第72列见到的字符,最多660个字符。END语句必须整个地写在一个初始行中,没有任何其它语句可以有出现一个END语句的初始行。

10. 程序单位

子程序与主程序都叫做程序单位。

一个子程序用SUBROUTINE语句或FUNCTION语句开始,用END语句结束。

一个主程序可任选地用一个PROGRAM语句开始。

Fortran语言规定了语句和行之间有一个确定的顺序以建立一个Fortran编纂。通常,一个编纂差不多包含一个主程序,零个或几个子程序(关于程序单位和子例程的编纂详见附录D)。语句的排序规则见下述。

11. 主程序和子程序

一个主程序以一个PROGRAM语句开始,或者以不是SUBROUTINE或FUNCTION语句的任何语句开始,用一个END语句作结束。

12. 语句次序

在一个程序单位内,无论是一个主程序还是子程序,语句必须以与下列规则一致的顺序出现:

(1) SUBROUTINE语句或FUNCTION语句或PROGRAM语句(如果有的话)必须作为程序单位的第一个语句。

(2) FORMAT 语句可以出现在 SUBROUTINE 语句或 FUNCTION 语句或 PROGRAM 语句（如果有的话）之后的任何位置。

(3) 全部说明语句必须先于所有 DATA 语句、语句函数语句和可执行语句。

(4) 在说明语句中，IMPLICIT 语句必须先于所有其它说明语句。

(5) 全部 DATA 语句必须出现在说明语句之后，先于所有语句函数语句和可执行语句。

(6) 所有语句函数语句必须先于全部可执行语句。

13. 程序单位中的语句次序

下列图表说明如下：

- 在其它语句之上或之下示出的语句必须按标出次序出现。
- 注释行或格式语句可以散布在对着它们出现的其它语句中间。

注 释 行	PROGRAM, FUNCTION 或 SUBROUTINE 语句	
	F O R M A T 语 句	IMPLICIT 语句
		其它说明语句
		DATA 语句
		语句函数语句
		可执行语句
END 语句		

三、数 据 类 型

在 IBM Fortran 中有四个基本数据类型。

- 1) 整型
- 2) 实型
- 3) 逻辑型
- 4) 字符型

本节描述每一类型常数的性质、值的范围和格式。

对于无格式文件与随机存取存储, IBM Fortran 各数据类型的存储要求是:

类 型	存储(字节)
LOGICAL	2 或 4
LOGICAL * 2	2
LOGICAL * 4	4
INTEGER	2 或 4
INTEGER * 2	2
INTEGER * 4	4
CHARACTER	1
CHARACTER * n	n
REAL	4
REAL * 4	4

1. 整型

数据类型 integer (整型) 是正数和负数的子集。一个整型值是相应整数的精确表示。一个整型变量占有 2 个或 4 个字节的存储。

IBM Fortran 中, 一个整数可以指定为 INTEGER * 2, INTEGER * 4 或 INTEGER。前两种分别指定 2 个和 4 个字节的整数, 第三种指定 2 个或 4 个字节的整数, 取决于 \$STORAGE 元命令的设置 (缺省元命令是 4 个字节)。

一个 2 字节的整数可以包括范围是 -32,767 到 32,767 的任意值, 一个 4 字节整数可包括范围是 -2,147,483,647 到 2,147,483,647 的任意值。

整型常数是由一位或多位十进制数前面加上一个任选的算术符“正”(+)或“负”(-)构成的, 在整型常数中不允许出现小数点, 下面是整型常数的例。

123	+123	-123	0
00000123	32767	-32767	

2. 实型

数据类型 real (实型) 是由实数的一个子集构成: 单精度实数。一个单精度实数值是一个要求的实数的近似, 一个单精度实数值占有 4 个字节存储。精度是六位十进制数。

一个实型常数是一个基本实常数, 一个基本实常数跟一个指数部分或一个整常数跟一个指数部分, 如:

+1.000E-2	1.E-2	1E-2
+0.01	100.0E-4	.0001E+2

都表示同样的实数值: 1/100

单精度实数值的范围是:

3.0E-39—1.7E+38	(正范围)
-1.7E+38—-3.0E-39	(负范围)

一个实型常数包括一个任选符号跟一个整数部分，一个小数点，一个小数部分和一个任选指数部分。整数和小数部分由一位或多位十进制数构成，小数点是一个点号（·），无论整数部分还是小数部分都可以省略，但不能两者均省略。

下面是一些基本实常数的例。

-123.456	+123.456	123.456
-123.	+123	123.
-.456	+.456	.456

指数部分由字母“E”或“e”后面跟一个任选地冠以符号的一位或二位整常数，一个指数表示其前面的值乘以 $10^{\text{整数次幂}}$ ，一些指数部分的例如下：

E12 E-12 E+12 E0

3. 逻辑型

数据类型 logical（逻辑型）由两个逻辑值 .TRUE.（真）和 .FALSE.（假）构成，一个逻辑型变量占有 2 个或 4 个存储字节。

一个逻辑数在 IBM Fortran 中指定为 LOGICAL * 2, LOGICAL * 4 或 LOGICAL。前二者分别指定 2 个或 4 个字节的逻辑数，第三个指定可以是 2 个或 4 个字节的逻辑数，这要取决于 \$STORAGE 元命令的设置（如果缺省该元命令，则为 4 个字节）。

只有两个逻辑常数，.TRUE. 和 .FALSE.；它们表示两个相应的逻辑值。.FALSE. 的内部表示是一个全“0”的字，.TRUE. 的内部表示是一个最低位为 1、其余全“0”的字，如果一个逻辑变量在任何其它位有值，则其逻辑值为未定义。

逻辑变量的含义不受元命令 \$STORAGE 的影响，其出现主要是为了可与 ANSI 要求兼容，即 LOGICAL, REAL 与 INTEGER 变量是同样大小。