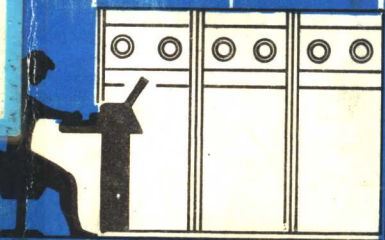


SVS

FORTRAN 77

程序设计语言



大连工学院出版社

SVS FORTRAN-77 程序设计语言

周 承 倜 译

大连工学院出版社

SVS FORTRAN-77 程序设计语言

SVS FORTRAN-77 CHENGXU SHEJI YUYAN

周 承 倜 译

责任编辑：尉迟喆斐

大连工学院出版社出版（大连市甘井子区凌水桥）

大连工学院印刷厂印刷 辽宁省新华书店发行

开本：787×1092 1/32 印张：7.5 字数：170,000

1986年8月第1版

1986年8月第1次印刷

印 数 1—5,000册

统一书号：15400·1

定价：1.60元

译 者 序

FORTRAN 是“公式翻译”(FORMULA TRANSLATION)的缩写,是广泛应用于各类工程和科学计算的一种程序设计语言。

本书所介绍的 SVS FORTRAN-77 是出自美国硅谷软件公司(American Silicon Valley Software, Inc.) 1983 年的版本(Version 2.0, 1 st Sept. 1983)。这是目前国际标准 FORTRAN-77 中的最新版本。它可以兼容各类较低级版本。

SVS FORTRAN-77 程序设计语言可用于微型计算机(例如, IBM PC-XT 机),也可用于 MC 68000 计算机系统的超级微型机(例如,使用 UNIX 操作系统的 DUAL 83/20 和 83/80 多用户超级微型机)。此外,在一些新型的国产电子计算机上凡是配备 FORTRAN-77 语言的都可以使用本版本。近年来从国外引进的许多中小型的微型电子计算机上,几乎毫无例外地配备有 FORTRAN-77 语言。由于 FORTRAN-77 的最新版本比 FORTRAN-IV 具有更为广泛的功能,所以用 FORTRAN-77 语言所编制的程序具有更高的效率和功能。可以预料,在近几年内,国产电子计算机上也会普遍配备 FORTRAN-77 语言。

FORTRAN-77 的功能很强,但是语言比较抽象和繁琐,并且语法规则十分严格。本文所介绍的 SVS FORTRAN-77 版本编写精炼,条理清晰,用语简洁,定义明确;同时书中附有相当数量的程序实例,对于学习和掌握语法规则很有帮助。

FORTRAN-77 程序设计语言已经形成了一个完整的算法语言系统。本书对于已经具有 FORTRAN-IV 语言知识的读者不会有什么困难。对于初学者来说，只要循序渐进，并配合一定的上机实习，相信也能顺利掌握要领，对于使用 IBM PC—XT 机或者使用 DUAL 83/20 或 DUAL 83/80 机的读者，本书可作为程序设计语言手册。

因为本版本具有规范文本的性质，所以对本书的翻译力求严格和准确。但是由于译者的水平所限，书中难免有不足和错误之处，望广大读者予以批评指正。

译 者

一九八六年二月

前 言

SVS FORTRAN-77是全集型 ANSI FORTRAN-77 计算机程序设计语言的一种实现语言，此语言适用于莫托罗拉公司 (MOTOROLA) MC 68000 计算机系统。本书是 SVS FORTRAN-77 的一本参考手册。

FORTRAN 是目前最为广泛应用的程序设计语言之一，它是最初应用于科学计算的语言。

FORTRAN 的发展最早是从 FORTRAN-II 开始进展到 FORTRAN-IV，而 FORTRAN-IV 在 1966 年被采用为 ANSI 标准。到七十年代中期，FORTRAN-IV 的扩充版本已广泛应用，形成了一种新的 FORTRAN 语言标准的基础，终于在 1977 年宣布了一种最新的语言（一致同意称之为 FORTRAN-77）。这种语言的标准形式在 1978 年由美国国家标准协会 (ANSI) 颁布。

FORTRAN-77 的 ANSI 标准实际上有两种语言版本：一种是“全集”语言；另一种是“子集”语言。子集略去了某些内容，例如复数数据类型和表式输入-输出。SVS FORTRAN-77 是全集语言的实现语言。

目 录

前 言

第一章 引 论 (1)

1.1 FORTRAN 语言的概述 (1)

1.2 本书所用的记号和专用名词 (6)

1.3 FORTRAN 的基本单元 (7)

第二章 行、语句以及控制流程 (10)

2.1 行 (10)

2.2 语句 (13)

2.3 执行顺序和控制转移 (17)

第三章 数据类型和常数 (19)

3.1 数据类型规则 (19)

3.2 常数 (20)

3.3 整数数据类型 (整数型) (20)

3.4 实数型 (21)

3.5 双精度数据型 (23)

3.6 复数型 (23)

3.7 字符数据型 (24)

3.8 逻辑数据型 (25)

第四章 FORTRAN 名字、数组和子串 (26)

4.1 FORTRAN 名字 (26)

4.2 数组说明 (29)

4.3 字符子串 (33)

第五章 表 达 式 (35)

5.1 算术表达式 (35)

5.2 字符表达式 (40)

5.3 关系表达式 (41)

5.4 逻辑表达式 (43)

5.5	运算符的优先规定	(44)
5.6	表达式的计算规则 and 限制	(45)
第六章 说明语句		(46)
6.1	类型语句——说明数据类型	(46)
6.2	维数语句 (DIMENSION) ——说明数据维数	(50)
6.3	公用语句 (COMMON) ——说明一个公用块	(50)
6.4	参数语句 (PARAMETER) ——作为一个符号关联的语句	(51)
6.5	隐式语句 (IMPLICIT)——建立缺席数据类型	(52)
6.6	外部语句 (EXTERNAL) ——说明外部过程或哑过程	(54)
6.7	内部语句 (INTRINSIC) ——说明内部函数	(54)
6.8	保存语句 (SAVE) ——保存定义状态	(56)
6.9	等价语句 (EQUIVALENCE) ——在单元之间共享存储	(57)
第七章 数据初始化		(59)
7.1	初始化字符变量	(61)
7.2	初始化非字符变量到字符值	(61)
7.3	在数据语句 (DATA语句) 中的隐式循环 (隐循环)	(63)
第八章 赋值语句		(65)
8.1	算术赋值	(65)
8.2	逻辑赋值	(67)
8.3	语句标号赋值	(67)
8.4	字符赋值	(68)
第九章 控制语句		(70)
9.1	块式 IF-THEN-ELSE 语句 (如果-则-否则)	(71)
9.2	逻辑 IF 语句	(75)
9.3	算术 IF 语句	(76)
9.4	DO 语句——循环控制	(77)
9.5	CONTINUE 语句 (继续语句)——空语句	(80)
9.6	STOP 语句 (停止语句) ——停止程序执行	(81)
9.7	PAUSE 语句 (暂停语句)——暂停程序执行	(81)
9.8	无条件 GO TO 语句	(81)
9.9	计算 GO TO 语句	(82)

9.10 赋值 GO TO 语句	(83)
第十章 输入和输出	(85)
10.1 输入-输出系统概述	(85)
10.2 输入输出系统的一般讨论	(91)
10.3 输入和输出语句单元	(97)
10.4 专用输入和输出语句	(101)
10.5 表式输入和输出	(113)
第十一章 格式说明	(118)
11.1 格式说明和格式语句	(118)
11.2 格式说明和输入/输出表 (I/O List) 之间的交互作用	(121)
11.3 编辑解说符	(123)
第十二章 程序和子程序结构	(135)
12.1 主程序	(135)
12.2 存取命令行变元	(136)
12.3 形式变元和实际变元	(137)
12.4 子例行程序	(140)
12.5 函数	(143)
12.6 ENTRY (入口) 语句	(147)
12.7 RETURN (返回) 语句	(149)
12.8 定义状态	(150)
12.9 BLOCK DATA (块式数据) 子程序	(151)
12.10 FORTRAN 内部函数	(153)
第十三章 FORTRAN 编译时间选择	(154)
13.1 \$INCLUDE——包括源文件	(154)
13.2 \$XREF——生成交叉引用	(154)
13.3 \$SEGMENT——指定分段名	(155)
13.4 \$COL72——限制源行到 72 列	(155)
13.5 FORTRAN-66 兼容性选择	(155)
附录 A——从 FORTRAN 系统得到的信息	(158)
A.1 编译时间错误信息	(158)
A.2 运行时间错误信息	(169)

附录 B——内部函数	(175)
B.1 内部函数的注解	(178)
B.2 变元范围的限制条件	(179)
B.3 非标准内部函数和子例行程序	(180)
附录 C——数据表达式	(183)
C.1 存储分配	(183)
C.2 数据表达式	(185)
C.3 变元传送机构	(191)
C.4 函数结果	(192)
C.5 寄存器约定	(192)
附录 D——ASCII 字符集表	(193)
附录 E——SVS FORTRAN 系统的操作	(194)
E.1 系统组成部份	(194)
E.2 命令行指令和编译程序选择	(197)
E.3 应用 Pascal 和 C 语言时的程序连接	(199)
附录 F——UNIX 操作系统专用信息	(205)
F.1 编译一个简单程序	(205)
F.2 错误信息文件	(206)
F.3 U 连接器	(207)
F.4 连接到 UNIX 汇编代码	(214)
F.5 存取到命令行变元	(215)
F.6 从 FORTRAN 程序的返回值	(215)
名词对照	(216)

第一章 引 论

FORTRAN 是一种计算机程序设计语言，其功能是进行数字计算。SVS FORTRAN-77 是由 ANSI 标准化协会提供的，迄今为止仍然是最新的语言版本。这一本 FORTRAN 参考手册所叙述的语言称为 FORTRAN-77，它是由美国硅谷软件公司 (SVS) 提供的实现语言，此后，名词 FORTRAN 就代表这一 FORTRAN-77 的实现语言。

1.1 FORTRAN 语言的概述

一个 FORTRAN 程序最终是由字符组成的。因为由字符组成行，行组成程序单元，程序单元再组成程序。

“行”可以是“注解行”，也可以是语句的“初始行”，或者语句的“继续行”。“行”所含的字符可以从第 1 列到第 120 列。与较老的 FORTRAN 实现语言相比较，SVS FORTRAN-77 编译程序在用户选择 \$COL72 编译指令时将略去各行的 72 列以后的字符（参阅第十三章——“FORTRAN 编译程序时间选择”）。

注解行是一些“空行”，这些空行是用字母 C（大写或小写字母）或者星号“*”写在第 1 列上。注解行可以放置在程序中的任何地方，包括放在语句的初始行和继续行之间。

初始行必须从第 7 列开始，要保持第 6 列是空格或者是一个零字符。一个语句的继续行需要在第 6 列放置一个除了零字符以外的任何的其他字符，或在第一列放置一个“和号

字符” (&)。

一个语句最多可以有 19 个继续行。在语句的初始行的第 1 列到第 5 列之间可以放置一个“语句标号”，每一个语句标号由 1 到 5 位数字组成，其中必须有一个数字是非零数字。语句标号的作用是作为某一语句的一个标志，以便其他语句引用。

语句大体上可以分成两类，即可执行语句和非执行语句。可执行语句在程序中所完成的作用是：给变量赋值、计算表达式、改变程序的执行流程和完成数据传送。非执行语句一般是说明程序内容的类型和属性。以下各段将要对语句进行更详细的讨论。

程序的量包括常量和变量。常量是一个数字串或者是一些其他符号，它的数值是不变的。变量占有存储单元，它的数值在程序执行过程中是可变的。变量和常量都可以被赋予一个“名字”和一个“数据类型”。名字的作用是为了标志程序中的一个目标。数据目标类型定义了一些其他内容，例如数据占有的存储量、数据的精度范围以及在某些情况下，这些数据所能完成的运算。在 FORTRAN 中，名字可以是“缺席数据类型”的，这种缺席型的名字要按照一种命名规则去确定，或者用一种显式说明方式来确定，而不考虑缺席命名规则。

一个变量可以是单一的，也可以是一个“聚集”。聚集数据变量的形式有两种，即“数组变量”和“字符变量”。一个数组变量是一个数据集合体，它占有连续存储单元。数组的维数最多只能是 7。一个字符变量代表一个字符串数据，并且是一个字符序列，它可以被单独赋值，也可以在一个“子串”的形式下被集体地赋值。

一个完整的 FORTRAN 程序由一个“主程序”和任

何数目的“子程序”组成。子程序的类别有：SUBROUTINE 子程序，它可以通过调用（CALL）语句的作用去执行外部的语句群；FUNCTION（函数）子程序，它在表达式的上下文中可以计算和返回一个数值；以及 BLOCK DATA（块式数据）子程序，它的作用是使数据初始化，这在 COMMON blocks（公用块）中做了说明。主程序和子程序一起便是所谓的“程序单元”。在一般情况下，名词“子程序”和“程序单元”可以被交换使用。由用户定义的“子例程序”和“函数”均可被称之为“过程”。

对于一个变量可以用“相联处理”的办法给予多个名字。相联数据可以有几种途径。用“公用语句”（COMMON）可以共享分隔的程序单元之间的数据。用“等价语句”（EQUIVALENCE）可以在同一程序单元中相联变量。当引用子程序或者函数时，变量也可以通过变元扫视机构而被相联。

变量的名字有一个“作用域”，它取决于定义变量的方法。除了那些程序单元名字，公共区域名字，以及某些其他的名字之外，在一般情况下，大多数变量名字的作用域是局部的，它们仅局限于在所定义的程序单元之内。如果一个名字被定义与一个被调用的外部函数一样，则在缺省时，它就具有全局作用域。在公用区域内所定义的名字仅局限于在定义它们的程序单元之内。公用区域本身的名字是全局的。形式参数对于语句函数具有一个作用域，它只局限于这个语句函数的语句本身。

“说明语句”是语句的两类主要群之一。说明语句的作用是解释变量和符号常数。说明语句包括：“类型”语句，它的作用是定义一个变量的数据类型；“维数”语句（DIMENSION），它的作用是定义数组变量的长度；“公

用”语句 (COMMON) 和“等价”语句 (EQUIVALENCE) 可提供变量的相联; “参数”语句 (PARAMETER) 是给予某一常数一个符号名字; “外部”语句 (EXTERNAL) 和“内部”语句 (INTRINSIC) 则具有定义其他程序单元的属性。

“数据”语句 (DATA) 是提供数据初始化的一个机构。数据语句中可以有一个“隐循环”语句, 它能以紧凑的形式促成数组变量的初始化。

“表达式”将数据和运算符组合在一起, 从而产生新的数值。FORTRAN 表达式包括算术表达式、字符表达式、逻辑表达式和关系表达式, 只要操作数和运算结果之间有明确定义, 那么采用混合型表达式是允许的。

赋值语句 (ASSIGN) 是将一个表达式的数值赋给一个变量。赋值的形式有三种: 即算术的、字符的、和逻辑的。赋值语句 (ASSIGN) 的作用是将一个语句标号的值, 赋值给一个整数变量。

控制语句是在程序中控制执行流程的语句。IF 语句的作用是为了控制程序的执行, 为此, 它根据所计算的是逻辑表达式的或是算术表达式的结果而选择各种不同类型的语句。在 DO 语句中, 将数值序列赋值给一个控制变量, 就可使语句块重复执行。所提供的 CALL 语句以及 RETURN 语句是为了执行子例程序和函数的。所提供的 GO TO 语句的各种变化形式可以在一个程序单元中执行控制的转移。

“语句-函数”语句的特征是在程序单元中定义一个用哑变元操作的单一语句“模板”。语句函数在程序单元中被引用时就好象一个具有实际变元的函数。变元要依据语句-函数的定义而组合以便产生一个可被应用于表达式的结果。

FORTRAN 提供了强有力的输入和输出功能。一个文

件可能是“外部的”(与外部设备相联结),也可能是“内部的”(对应于一个字符变量);文件可以是“格式化的”,也可是“非格式化的”;文件可以被“顺序地”存取,也可被“随机地”存取。格式化文件可以是数据转换操作的主题,其转换是从内部存储表示到外部字符串表示,或者反之。

格式化转换是通过“读”(READ),“写”(WRITE),或者“打印”(PRINT)语句来完成的。“格式说明”有一组功能强的语句去控制转换数据的形式和格式。还有一种表式输入-输出功能,在那里缺席格式化规则可被用于转换进程。

“子例行程序”(SUBROUTINE)和“函数”(FUNCTION)程序单元可以具有变元,在处理进程中用调用程序传送它们。在说明子例行程序或者函数时,它的形式变元也同时被说明了;当子例行程序和函数被引用时,形式变元便被实际变元所代替。一个子例行程序或者函数可以有多个入口点(ENTRY)。除了第一句可执行语句之外,一个入口(ENTRY)语句,可以在一个语句的开始处执行一个子例行程序或函数。在遇到END语句,或者执行一个RETURN语句时,控制可从一个子例行程序或者函数程序单元返回。对于子例行程序,FORTRAN还提供了另外一种返回说明,它使子例行程序可以返回到与调用(CALL)语句所说明的不同的地方。

FORTRAN有一组内容众多的内部函数,它们可以完成数据类型的转换,可以提供一组各种类型的算术函数和超越函数的集合体。

1.2 本书所用的记号和专用名词

本节将要叙述应用于 FORTRAN 语言结构的符号。

大写字母和特殊字母将在程序中示出。小写字母和文字所表示的将是在程序的上下文中所描述的实际语句中的内容。一个小写的目标一旦已被定义，它就能够在已被定义的语言结构的其余部分中保持它的意义。

大写和小写的应用实例

Iw 表示描述整数编辑的格式说明，在这里 w 是一个非零的，无正负号的整常数。在一个实际的“格式”(FORMAT)语句中，编辑说明必须写成 I5 或者 I21。对于实数的编辑说明是 Fw.d，此处 d 是一个无正负号的整常数。一个实际的 FORMAT 语句应该包含一个象 F8.4 或 F14.0 这样的编辑说明。注意，根据以上定义，句号字符是一个特殊字符，它被看作是一个文字字符。

方括号“[”和“]”起闭合任选项的作用。例如，A[w] 表示或者是 A 或者是 A12 这两种形式都是正确的（这里作为一个字符格式的说明方法）。

略字记号“...”表示写在略字记号前的任选项可以出现一次或者重复多次。例如，计算的 GOTO 语句可以具有这样形式：

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

此处表示语法项，s' 可以重复出现任何次数。

空格（空白），在 FORTRAN 语句中通常是无效的（无意义的）。由于在所有的书籍中都提供了空格的解释，所以在本章的最后将介绍空格的一般规则。在本书中，“空格”和“空白”将被看作是同义词。一般情况下使用“空

格”一词。

1.3 FORTRAN 的基本单元

本节要叙述构成一个 FORTRAN 程序所需要的基本的文字单元和语法单元。

1.3.1 FORTRAN 字符组

FORTRAN 字符组包括大写字母、小写字母、数字以及特殊字符。

字符表:

字 符	字 符 名 字
	空白或空格
=	等号
+	加号
-	减号 (负号)
*	星号
/	斜杠
\	反斜杠
(	左括号
)	右括号
,	逗号
.	小数点
\$	货币符号
'	上撇符号 (上撇)
:	冒号
&	和符号 (和号)