



APPLE-II 微型计算机

APPLE FORTRAN 语言参考手册

中国微型电脑应用协会武汉分会

序

这本手册是根据APPLE微型计算机软件资料APPLE FORTRAN Language Reference Manual编译而成的。

考虑到手册的适用性,在编译过程中,删去了某些对于用户来说可有可无或重复的内容,增加了程序设计实例和便于查阅的小结;对原文手册中介绍的操作和程序实例进行了复核,并根据实际操作结果作了订正或补充说明。与原文本相比编译本有如下明显变动:

(1)将原文第一章和第二章合并为一章,并作了少量删减;

(2)突出了双驱动器操作,删去了单驱动器操作中与双驱动器操作重复的部分;对原文中与实际操作结果不相符或不明确的部分,按实际操作结果进行了修改;

(3)为了使用户对APPLE FORTRAN程序设计和绘图功能有进一步了解,增加了绘图程序设计实例;为了便于用户查阅,把APPLEFORTRAN操作命令作了小结;还增加了主程序和子程序分别编译及其连接的实例。这些,分别作为附录B、附录C和附录J。

在编译过程中,借鉴了南京无线电厂的翻译本(DTS-033软件之六)。

由于我们的水平所限,书中肯定存在缺点和错误,恳请读者予以批评指正。

目 录

前 言	(1)
第一章 概述	(2)
§1 APPLE FORTRAN以及与其它FORTRAN 语言的比较.....	(2)
§1.1 APPLE FORTRAN与ANSI FORTRAN 77 子集的比较	(2)
§1.2 ANSI 77 与全语言的比较	(3)
§1.3 ANSI 77 与ANSI 66 的比较	(3)
§2. 在APPLE PASCAL 操作系统上运行APPLE FORTRAN	(4)
§3 本手册设及的内容及手册的使用	(5)
§4 PASCAL操作系统上PASCAL和FORTRAN语言之间的关系.....	(5)
§5 PASCAL文献指南	(6)
§5.1 命令层	(6)
§5.2 文件管理程序	(6)
§5.3 编辑程序	(7)
§5.4 6502汇编程序	(7)
§5.5 连接程序	(7)
§5.6 实用程序	(7)
第二章 程序分段	(8)
§1 分段编译	(8)
§1.1 源代码分段	(8)
§1.2 目标代码分段	(9)
§2 单位、段和库	(10)
第三章 编译程序	(11)
§1 运行FORTRAN编译程序需要的文件.....	(11)
§2 编译程序的使用	(12)
§3 输入程序的格式.....	(13)
§3.1 字母的大小写.....	(14)
§3.2 行的长度和定位.....	(14)
§4 编译程序控制指令.....	(15)
§5 编译程序列表.....	(16)
第四章 连接程序.....	(18)

§1	所需要的磁盘文件·····	(18)
§2	连接程序的使用·····	(19)
第五章	程序结构·····	(22)
§1	字符集·····	(22)
§2	行·····	(23)
§3	列·····	(23)
§4	空格·····	(23)
§5	注释行·····	(23)
§6	语句、标号和行·····	(24)
§7	语句顺序·····	(24)
§8	END语句 ·····	(25)
第六章	数据类型·····	(26)
§1	整型·····	(26)
§2	实型·····	(26)
§3	逻辑型·····	(27)
§4	字符型·····	(27)
第七章	FORTRAN语句 ·····	(28)
§1	FORTRAN名 ·····	(28)
§1.1	FORTRAN名的作用域 ·····	(28)
§1.2	未说明的名·····	(29)
§2	说明语句·····	(29)
§2.1	IMPLICIT语句 ·····	(29)
§2.2	DIMENSION语句 ·····	(30)
§2.3	Type语句 ·····	(31)
§2.4	COMMON语句·····	(32)
§2.5	EXTERNAL语句·····	(33)
§2.6	INTRINSIC语句·····	(33)
§2.7	SAVE语句·····	(33)
§2.8	EQUIVALENCE语句 ·····	(33)
§3	DATA语句·····	(34)
§4	ASSIGNMENT语句·····	(35)
§4.1	计算型赋值语句·····	(35)
§4.2	标号型赋值语句·····	(35)
第八章	表达式·····	(36)
§1	算述表达式·····	(36)
§1.1	整除·····	(37)

§1.2	类型转换和结果类型	(37)
§2	字符表达式	(37)
§3	关系表达式	(37)
§4	逻辑表达式	(38)
§5	运算符优先级	(39)
第九章	控制语句	(40)
§1	UNCONDITIONAL GOTO	(40)
§2	COMPUTED GOTO	(40)
§3	ASSIGNED GOTO	(40)
§4	ARITHMETIC IF	(41)
§5	LOGICAL IF	(41)
§6	BLOCK IF...THEN...ELSE	(41)
§7	BLOCK IF	(43)
§8	ELSEIF	(44)
§9	ELSE	(44)
§10	ENDIF	(44)
§11	DO	(44)
§12	CONTINUE	(45)
§13	STOP	(46)
§14	PAUSE	(46)
§15	END	(46)
第十章	输入/输出操作	(47)
§1	I/O概述	(47)
§1.1	记录	(47)
§1.2	文件	(47)
§1.3	格式化与非格式化文件	(48)
§1.4	顺序存取和直接存取	(48)
§1.5	内部文件	(48)
§1.6	设备	(49)
§2	选择一个文件的结构	(49)
§3	I/O限制	(50)
§4	I/O语句	(51)
§4.1	OPEN	(52)
§4.2	CLOSE	(53)
§4.3	READ	(54)
§4.4	WRITE	(55)
§4.5	BACKSPACE	(55)
§4.6	ENDFILE	(55)

§4.7	REWIND	(55)
§5	关于I/O操作的注意事项	(56)
第十一章 格式化I/O		(58)
§1	格式化I/O	(58)
§2	格式和I/O列表	(59)
§3	不可重复的编辑描述符	(60)
§3.1	撇号编辑	(60)
§3.2	H 何勒内斯编辑	(60)
§3.3	× 位置编辑	(60)
§3.4	/ 斜杠编辑	(60)
§3.5	\$ 美元符号编辑	(60)
§3.6	P 比例因子编辑	(61)
§3.7	BN/BZ 空格的翻译	(61)
§4	可重复的编辑描述符	(61)
§4.1	I 整型编辑	(62)
§4.2	F 实型编辑	(62)
§4.3	E 实型编辑	(62)
§4.4	L 逻辑型编辑	(63)
§4.5	A 字符型编辑	(63)
第十二章 程序单位		(64)
§1	主程序	(64)
§2	例行子程序	(64)
§2.1	SUBROUTINE语句	(64)
§2.2	CALL语句	(65)
§3	函数	(65)
§3.1	外部函数	(65)
§3.2	内部函数	(66)
§3.3	语句函数	(69)
§3.4	RETURN语句	(69)
§4	参数	(69)
第十三章 编译单位		(71)
§1	单位、段、部分编译	(71)
§2	连接	(73)
§3	\$USES编译程序控制指令	(73)
§4	分别编译	(74)
§5	FORTRAN复盖	(74)

第十四章 两种语言的程序设计	(75)
§1 在FORTRAN主程序中的PASCAL.....	(75)
§2 在PASCAL主程序中的FORTRAN.....	(77)
§3 两种语言的程序中的 I/O	(78)
§4 调用机器代码例行程序	(78)
第十五章 专用程序单位	(81)
§1 Turtle制图单位.....	(81)
§1.1 APPLE屏幕	(81)
§1.2 INITTU例行子程序	(81)
§1.3 GRAFMO例行子程序	(82)
§1.4 TEXTMO例行子程序	(82)
§1.5 VIEWPO例行子程序	(82)
§1.6 选择颜色的例行子程序.....	(82)
§1.7 笛卡尔制图例行子程序.....	(84)
§1.8 Turtle制图例行子程序	(84)
§1.9 Turtle制图函数.....	(85)
§1.10 把一个数组送到屏幕.....	(85)
§1.11 制图屏幕上的正文.....	(86)
§2 APPLESTUFF单位	(87)
§2.1 RANDOM函数/RANDOI例行子程序	(88)
§2.2 游戏控制的使用.....	(88)
§2.3 产生音乐;NOTE例行子程序	(89)
§2.4 KEYPRE函数	(89)
附 录	
附录A 第一部分:多驱动器操作	(90)
§1 用户所需要的设备.....	(90)
§2 配置用户系统.....	(91)
§3 系统的启动.....	(93)
§4 日期的修改.....	(93)
§5 制作后备软盘.....	(94)
§5.1 怎样制作后备软盘.....	(94)
§5.2 格式化软盘.....	(95)
§5.3 实际副本的制作.....	(96)
§6 系统的使用.....	(98)
§6.1 一些有趣的例子.....	(98)
§6.2 运行一个程序.....	(101)
§6.3 输入一个程序.....	(102)
§6.4 把系统盘留在驱动器中.....	(103)

附录A	第二部分:单驱动器操作.....	(103)
§1	配置用户系统.....	(103)
§2	制作后备软盘.....	(105)
§2.1	格式化软盘.....	(105)
§2.2	实际副本的制作.....	(106)
§3	系统的使用.....	(108)
§3.1	一些有趣的例子.....	(109)
§3.2	运行一个程序.....	(112)
§3.3	输入一个程序.....	(112)
附录B	一个绘图程序的例子.....	(114)
附录C	APPLE FORTRAN操作命令小结.....	(117)
附录D	FORTRAN出错信息表.....	(123)
附录E	表.....	(130)
附录F	FORTRAN语法图.....	(135)
附录G	FORTRAN语句小结.....	(157)
附录H	ANSI FORTRAN与ANSI FORTRAN 77的比较.....	(159)
附录I	APPLE FORTRAN与ANSI FORTRAN 77的比较.....	(162)
附录J	主程序和子程序分别编译及其连接的实例.....	(164)

前 言

这本手册介绍了APPLE II和APPLE II-Plus计算机的FORTRAN程序设计语言。APPLE FORTRAN采用的是美国国际标准FORTRAN的子集,也称为ANSI FORTRAN 77的子集。

APPLE FORTRAN对ANSI标准子集进行了扩充,例如:它结合了全语言的一些特点,而这些特点是标准子集所没有的。APPLE FORTRAN还具有一些特性,这些特性是APPLE独特操作环境的产物。ANSI标准子集也包括了对全语言所作的大部分重要修改,从而使它超过了先前的标准子集(即ANSI FORTRAN 66)。

本手册的作用在于:

(1)帮助用户熟悉APPLE FORTRAN与标准FORTRAN 77的差异和扩充。

(2)帮助用户熟悉APPLE FORTRAN在APPLE II和APPLE II-Plus计算机上的操作环境(FORTRAN使用PASCAL操作系统)。

(3)手册中介绍了ANSI FORTRAN与ANSI FORTRAN 66的主要差别,以帮助用户熟悉FORTRAN的较新版本。

(4)为用户提供完整的APPLE FORTRAN语言说明。

完整的APPLE FORTRAN文本还包括《APPLE语言系统的配置和操作手册》。

为了使用户熟悉PASCAL操作系统,手册中指出了应参阅的PASCAL文献,即APPLE语言系统中的PASCAL手册。

为了运行FORTRAN,用户必须事先配置FORTRAN系统,该系统安放在两个软盘(FORT1:和FORT2:)上。FORT2:上放有SYSTEM·COMPILER和SYSTEM·LIBRARY。FORT1:上放有FORTLIB·CODE。在附录A中介绍了如何配置用户系统,第一部分适用于多驱动器用户,第二部分适用于单驱动器用户。该附录中提出的配置方案,是假定FORT2:就是用户系统盘。这本手册也是按照这一假定来编写的。对于FORTRAN系统的配置,可以有不同的选择。但必须注意,本书中是把FORT2:作为系统的引导盘来使用。

此外,如果用户以前没有使用过APPLE PASCAL,可参考附录A中有关使用该操作系统的说明。该附录中还介绍了APPLE FORTRAN的启动过程。

第一章 概 述

本章简单介绍了APPLE FORTRAN的产生以及它与其它FORTRAN语言的比较,它使用的操作系统,本手册所包括的内容和如何使用这本手册。由于APPLE FORTRAN是运行于PASCAL操作系统上,因此,为了利于读者阅读本手册在这一章中对有关的PASCAL文献的内容也作了简单提示。

§1 APPLE FORTRAN以及与其它 FORTRAN语言的比较

1966年美国国际标准化协会(ANSI)颁布了标准FORTRAN,它对人们理解该语言有很大帮助。这就是通常说的FORTRAN 66。

从那以后,该语言仍在继续发展,出现了大量扩充部分,这些部分比较新颖,且具有普遍性。因此,1977年ANSI制定了另一标准,称为FORTRAN 77。这一标准包括了那些扩充部分。目前这个更新的标准已被广泛地接受。APPLE FORTRAN是建立在ANSI FORTRAN 77全语言基础上的一个正式ANSI子集。FORTRAN不断地发展,不断地产生新的内容。几乎每一种FORTRAN都具有一些特点是所使用的处理机所特有的。APPLE FORTRAN也不例外。

对于熟悉其它FORTRAN版本的用户来说,重要的是弄清楚APPLE FORTRAN与其它版本究竟有何差别。这需要分三个问题来回答:

(1) APPLE FORTRAN与ANSI FORTRAN 77标准子集之间的差别何在?

(2) ANSI子集与全语言之间的差别何在?

(3) ANSI 63与ANSI 77的差别何在?

下面依次讨论这三个问题。

§1.1 APPLE FORTRAN与ANSI FORTRAN 77子集的比较

尽管APPLE FORTRAN大部分是和ANSI标准子集一致的,但也有一些细微的差别。它不具备某些特性,但它从全语言说明中吸取了另一些特性,在某些情况下,它还超出了标准的范围。

在以下两种情况下,APPLE FORTRAN与ANSI FORTRAN子集不同:

(1) 整型和实型数据使用存贮单元的数量不同。ANSI规定它们必须相同。而在APPLE中, 实型数据占有存贮器的4个字节, 整型和逻辑型则占2个字节。这对整型数据意味着可表示数的范围是 -32768 到 $+32767$, 非零实型常数的数值最大范围约在 $5.8E-39$ 到 $1.7E+38$ 之间。

(2) 子程序名不能作为形式参数传送给另一子程序。

有些功能在APPLE FORTRAN中是允许的, 但这些虽然在全ANSI FORTRAN中, 却不为子集所具有。这些功能是:

(1) 下标表达式——APPLE FORTRAN和全ANSI 77语言允许在下标表达式中出现函数调用和数组引用。

(2) DO变量表达式——该子集对定义DO语句范围的表达式作了限制, 而全语言则没有。APPLE FORTRAN在DO语句的范围计算中也允许完全整型化的表达式。类似地, 也允许任意整型表达式出现在涉及读和写语句所隐含的DO循环中。

(3) I/O设备号——APPLE FORTRAN可以用表达式来指定一个I/O设备。

(4) I/O表中的表达式——APPLE FORTRAN允许表达式出现在写语句的I/O表中, 但表达式不能以左括号开头。但由于对以左括号开头的表达式, 如: $(A+B) - (C+D)$, 可以在输出表中写成: $+ (A+B) - (C+D)$ 从而巧妙地解决了这一问题。然而, 在运行时, 对前面的“+”号求值不产生任何代码。

(5) 在计算型GOTO中的表达式——APPLE FORTRAN允许以表达式作为计算型GOTO的计算值。

(6) 一般的I/O——APPLE FORTRAN对顺序存取和直接存取文件可以是格式化的, 也可以是非格式的。子集语言则要求直接存取文件是非格式的, 顺序存取文件是格式化的。APPLE中还增加了OPEN(打开)语句来接收附加的参数, 而在子集语言中不包括这些附加参数。此外, APPLE还提供了CLOSE(关闭)语言, 这也是子集语言所没有的。关于I/O更详细的解释见第十章。

(7) CHAR内部函数——APPLE FORTRAN含有CHAR内部函数。

在某些情况下, APPLE FORTRAN所具有的一些特性是ANSI标准子集和全语言中都没有的。这是指增加的编译程序控制指令。这类指令允许用户传送某些信息给FORTRAN编译程序。增加了编译程序控制指令行, 它由编译程序识别并接收这一信息。有关这些命令的介绍见第三章。另外APPLE FORTRAN还含有内部函数EOF。

§1.2 ANSI 77与全语言的比较

为了帮助读者弄清ANSI标准子集的有用特性, 这本手册有两个关于子集语言的摘要的附录。附录F给出了完整的子集语言语法图, 同时附有APPLE FORTRAN的说明, 附录G给出了子集语言中所有的语句列表和它们的语法。

§1.3 ANSI 77与ANSI 66的比较

ANSI FORTRAN 77与FORTRAN 66之间的差别(如ANSI 77删去了HOLLERITH数据类型)将在附录H里讨论。ANSI 77所增加的性能和在ANSI 66中未义的部分前面已作了介绍。

§2 在APPLE PASCAL操作系统上运行 APPLE FORTRAN

APPLE FORTRAN是在APPLE PASCAL操作系统上运行的一种程序设计语言，这个功能很强的操作系统通常是以支持多种语言，而不只局限于PASCAL。把FORTRAN语言置于APPLE PASCAL操作系统上，对程序设计有以下几个明显的优点：

(1) 完整的FORTRAN开发系统(包括正文编辑程序、文件管理程序、代码库和库管理程序、汇编语言编译程序,以及其它各种实用程序)与提供给PASCAL的相同。

(2) 可以很容易地形成一个FORTRAN承包系统(注),在开机时该系统立即开始运行一个给定的程序。

(3) FORTRAN和PASCAL语言在同一操作系统上操作。因此,运行FORTRAN或PASCAL程序,只需学会一种操作系统。

(4) PASCAL例行子程序可以链接到FORTRAN程序上,反之亦然。

(5) 汇编语言子程序既可以链接到FORTRAN上,也可以链接到PASCAL上,或与两者都链接。

还有其它一些优点:PASCAL操作系统提供了许多较理想的功能,这些功能也体现在APPLE FORTRAN中。这些扩充使得用FORTRAN编写程序及使用FORTRAN程序变得更加容易。把FORTRAN置于PASCAL操作系统上,也引起了对FORTRAN的两个较小的限制,这在上一节“APPLE FORTRAN与ANSI FORTRAN 77子集的比较”中已经谈到。

APPLE PASCAL程序包与APPLE FORTRAN程序包的本质差别就在于编译程序不同。另外几个不大的差别将在后面有关章节中讨论。FORTRAN编译程序和PASCAL编译程序产生的输出代码是相同的。FORTRAN和PASCAL所建立的代码文件是由同一操作系统交替管理的。

程序开发系统由编辑程序,连接程序、文件管理程序和其它实用程序及库程序组成。程序开发的顺序是:

(1) 用编辑程序写FORTRAN程序。

(2) 用文件管理程序来存取软盘上的文件。

(3) 用FORTRAN编译程序把正文文件翻译成代码文件。

(4) 用连接程序把各代码文件连接成一个可执行的代码文件。

(5) 执行该程序

为了把PASCAL和FORTRAN程序链接在一起,或把FORTRAN程序与汇编语言程序链接在一起,还需要其它的附加措施。

(注) FORTRAN承包系统是指:APPLE计算机的硬件、软件在安装前已经过全面而又可靠的调试,用户只要理解操作说明,就可以在APPLE上使用FORTRAN语言。

§3 本手册涉及的内容及手册的使用

这一章的后两节,将要讨论在PASCAL操作系统环境中FORTRAN语言的使用,指导用户在PASCAL操作系统文献的哪些部分查找最重要的特性。

从第二章到第十五章的内容是本手册的三个主要部分。第二章至第四章为第一部分,讨论怎样有效地编制FORTRAN程序, FORTRAN编译程序输入的条件及连接程序的使用。第二部分(第五章至第十三章), 包含了APPLE FORTRAN的语言说明和描述, 给出了数据类型、表达式、语句、I/O条件和FORTRAN编译程序所能接受的程序格式的有关细节。第三部分(第十四章和第十五章), 讨论FORTRAN和PASCAL之间的程序连接、从FORTRAN中调用汇编语言例行子程序、彩色图像技术和其它特殊的APPLE功能。

在手册的其它地方给出了附加的摘要, 提供了另一些有用的内容。APPLE PASCAL/FORTRAN新用户在使用系统之前, 必须先阅读附录A。该附录包括一些指导性内容, 如怎样开始使用程序、开发系统、编辑程序、连接程序等, 还提供了有关运行多驱动器或单驱动器系统所需的完整资料。

PASCAL和FORTRAN所共用的操作系统的部分, 已写在PASCAL操作系统文献中, 为了学习怎样建立、编辑和运行FORTRAN程序, 用户必须有APPLE PASCAL参考手册, 或APPLE PASCAL操作系统手册。后一手册是前一手册的PASCAL操作系统部分的补充。为能有效地应用本系统, 应阅读PASCAL手册的全部章节。

PASCAL操作系统的文本介绍了两方面的内容: 即程序开发系统和操作系统的特征。PASCAL文献中的例子是PASCAL程序。因为编辑程序、文件管理程序和连接程序等内容与所使用的程序设计语言无关, 因此在学习这些内容时不会遇到困难。

PASCAL文献中的另一部分是专门面PASCAL用户的, 这本手册中抄录了这些章节的某些部分。在有些情况下, 要求用户戴上“FORTRAN有色眼镜”去阅读PASCAL文献中的特殊章节, 如软盘与软盘之间盘名的转换、字FORTRAN与字PASCAL之间的转换, 等等。

§4 PASCAL操作系统上PASCAL和FORTRAN语言之间的关系

这一节和下一节有两个作用, 一是把APPLE PASCAL参考手册和操作系统手册中所讲的内容表列出来, 使用户对它们有一般了解, 二是帮助用户利用FORTRAN来解释那些手册。

对于编辑程序、文件管理程序、连接程序和操作系统的其它许多方面的概况, PASCAL文献中都作了完整的介绍。文献中还给出了程序例子和(供PASCAL进行操作的)软盘名。对于FORTRAN, 多数情况下, 只需对FORTRAN与PASCAL之AL间软盘名和字转换作些解释。在不容易讲清楚的地方附有例子以供参考。以上这些就是这一节和下一节所要讨论的。

PASCAL操作系统上PASCAL和FORTRAN语言之间的关系的要点是：

(1) PASCAL文献涉及到一些特殊文件，这些特殊文件是PASCAL操作系统的一部分，例如SYSTEM·APPLE，它对于APPLE FORTRAN和APPLE PASCAL的作用是相同的。对此有两个明显的例外：一是SYSTEM·COMPLIER，它是FORTRAN的编译程序另一个是包含了一些专用单位的SYSTEM·LIBRARY和RTUNIT·CODE，它们都只用于FORTRAN。另外在PASCAL语言系统中，还有一个叫做SYSTEM·SYNTAX的文件，它与FORTRAN毫无关系。

(2) PASCAL文献用到了伪代码——P-代码。FORTRAN和PASCAL编译程序产生P-代码（指令），而不产生APPLE固有的6502机器代码。编译程序产生的P-代码，由P-代码解释程序把P-代码指令翻译成APPLE（6502）机器代码。这样，就使FORTRAN和PASCAL都可以在PASCAL操作系统上运行。

(3) 本操作系统使用两种基本软盘文件：TEXT和CODE。TEXT文件采用人们可以阅读的格式，而CODE文件是机器可读的。TEXT文件是一个字符流，它们可以是英文的，FORTRAN的或其它形式的。CODE文件是FORTRAN和PASCAL编译程序通过读FORTRAN或PASCAL语言的TEXT文件而产生的。采用后缀.TEXT或.CODE为文件名结尾的文件由操作系统专门处理。例如编辑程序仅允许用户编辑具有后缀.TEXT的文件。连接程序仅仅连接具有后缀.CODE或后缀.LIBRARY的文件。带有后缀.CODE的文件具有特殊的内部结构且允许系统程序以适当的方式操作它们。虽然该系统很注意文件的后缀。但用户可以改变任何文件有任一后缀。一般情况下，应确保正文文件有后缀.TEXT，二进制文件有后缀.CODE。

§5 PASCAL文献指南

PASCAL操作系统的阅读指导，提出了为解决FORTRAN的各种应用所应阅读的内容，及所要进行的具体替换和改写。不论读那本手册的引导部分，用户对PASCAL操作系统都必须有一个大概的了解。

§5.1 命令层

命令层中承包系统这一部分，在两本PASCAL手册中都未谈到。因为这些资料对于FORTRAN用户来说很重要，所以在有必要在这里作些介绍。

APPLE PASCAL系统允许用户建立一个承包系统。当APPLE开机时，该系统能自动地开始运行一个特殊的程序。要在APPLE上建立一个承包系统，首先必须形成软盘FORT2：上的所有文件副本，但SYSTEM·COMPILER除外，因为承包系统不需要该文件。使用文件管理程序中的C（Change）命令来改变软盘名。例如，用户想给副本取名为TURNKEY：。然后用T（Transfer）命令传送程序代码文件的副本给承包软盘。程序的新副本必须取名为SYSTEM·STARUP。

§5.2 文件管理程序

文件管理程序是在FORT1：上，所以，在调用文件管理程序之前，该盘必须在某一

个驱动器中。

§5.3 编辑程序

编辑程序带是在FORT1:上。为了更好地运用它,应阅读PASCAL手册中文本编辑(Editor)一章中的修改文本命令(Text Changing Commands)一节,Insert, Auto-indent, TRUE, Filling FALSE等命令。这些通常是用来输入FORTRAN程序和PASCAL程序的命令。

§5.4 6502汇编程序

FORTRAN程序可以调用汇编语言子程序。在PASCAL手册中有关6502汇编程序的章节里,讨论了怎样书写和连接这样的子程序。然而,在PASCAL这一章的末尾给出的许多例子却是用PASCAL语言写的。汇编语言程序ASMDemo用于FORTRAN时,需要修改几行代码才能工作。根据FORTRAN要求而修改的ASMDemo程序,将在本手册的第十四章讨论。

§5.5 连接程序

本手册第四章将介绍连接程序。

连接程序将分别编译的代码文件连接成一个可执行的代码文件。提供给PASCAL和FORTRAN系统的是同一个连接程序。因为PASCAL和FORTRAN编译程序产生的目标代码文件具有相同的结构,所以能正确地进行连接。不过PASCAL和FORTRAN使用连接程序的方法有所不同(详见第四章)。

§5.6 实用程序

PASCAL和FORTRAN所用软盘的格式和格式化程序都相同,注意,FORMATTER代码文件是放在软盘APPLE3:上,在完成格式化之前,该盘必须放在某一驱动器中。这个软盘的内容是由PASCAL程序包来提供的(参见附录A)。

如果需要建立、扩充或变更库文件,则应阅第PASCAL手册中系统库管理程序部分。有一个叫做SYSTEM.LIBRARY的主库,是建立在软盘FORT2:上。PASCAL和FORTRAN用同一个库管理程序来管理这个库;但是,FORTRAN系统所用的库管理程序应是在FORT1:上的FORTLIB.CODE。APPLE3:上的LIBRARY.CODE仅能用于PASCAL。两者的库的结构相同。但FORTRAN中的SYSTEM.LIBRARY的内容稍有差别。举例来说,FORTRAN中有实用RTUNIT,而PASCAL却没有。

使用库管理程序的用户还应阅读库映象部分,这一部分介绍了允许用户观察任何库的内容的程序。

如果用户准备使用非标准控制台,则需要重构一部分操作系统,以使用此控制台来代替APPLE控制台。可以象在重构和改变GOTOXY通讯两节中所讨论的那样来重构PASCAL操作系统。

第二章 程序分段

FORTRAN系统提供各种方法让用户选择书写和构造程序的步骤。最重要的是对于大的程序编制课题,用户可以从主程序中移出子程序集,并把它们放入一些独立的模块中。这些模块可以分别设计,甚至可用于多个程序。把代码块作为模块来处理的方法比写一个大程序具有若干优点:

(1) 大程序可以概念性地分成一些可以独立设计的部分,以后再合并起来。因此,一次可以集中处理一个问题,从而可以加快研制速度。

(2) 能够用于许多程序中的子程序只需编写一次,然后存放在库中,便可供以后使用。

(3) 已经写好的许多常用子程序,放在系统库里。它们可以直接用在用户程序中。该库是在软盘F0RT2:上。

(4) 用PASCAL和汇编语言编写的过程,可以连接到FORTRAN语言主程序上。

(5) 操作系统可以使用几种非常有效的管理子程序的方法。这些方法可以帮助缩小被编译的程序,从而避免磁盘或计算机内存的存储量因程序大而超容。这些方法在本章后面将更充分地加以讨论。

程序分段使得有可能把代码块作为模块来处理。尽管这是一项非常有用的功能,但它也带来一些复杂的问题。

这一章的目的在于解释APPLE操作系统的使用。

§1 分段编译

一个FORTRAN程序由一个主程序和与其相应的子程序组成。最简单的情况是主程序和所有子程序都出现在同一个正文文件中,称为源代码,并且是一次编译而成。编译的结果叫做目标代码文件。它包括完整的程序并放在软盘上。当准备执行该程序时,把它一次全部装入内存,并开始执行。装入内存的程序叫做代码映象。

当程序正在运行时,程序可在三个层里分成模块:源代码层、目标代码层及该程序正在执行时的代码映象层。每层都有不同的作用,下面讨论如何利用各层的作用和特点。

§1.1 源代码分段

可以把一个程序的任何部分分离出来,放到不同的正文文件中。在移去部分程序的位置

上,填入专门的语句。这些语句使得FORTRAN编译程序编译时在正确的地点包含那些分段的文件。仅有的约束是:程序的分段必须保证行的完整。即不能将语句分段,并且要保证编译程序重新将程序集中起来时,这个文件集仍然是一个完整FORTRAN程序。

把程序分割成模块有两个作用。首先,不必一次编辑整个程序;其次,可使程序中的不同工作分别出现在分离的正文文件中。这就有利于保持程序组织结构的清晰。要更的了解更多有关把程序分割为模块的内容,可参看第三章有关 \$INCLUDE 语句的讨论。

§1.2 目标代码分段

如果在正文文件中的代码程序是一个完整的子程序,则它可以单独编译,而不必与它所属的整个程序一起编译,把主程序中的子程序分别编译的方法叫做分段编译。

如果需要的话,一个完整的子程序也可以分段送入不同的正文文件中。至于子程序是怎样被送到不同的正文文件中去则无需关心,但所有的段都必须提供给编译程序,至少保证构成一个完整的子程序。

在最简单的情况下,按下面两个步骤进行分段编译。第一,编译每一个子程序。如果需要,一个独立的正文文件实际上可以包括较多的完整子程序。编译开始时,它查到所含的不是主程序,因为第一个语句是SUBROUTINE或FUNCTION语句。然后由编译程序对子程序进行编译,当成功地编译完子程序时,编译程序将写出代码文件。该文件即包括编译过的程序代码,也包括一个描述代码文件所含内容的信息包。连接程序包也包含在编译单位中的子程序的有关详情,如它们的类型、变量的数目等等。

分段编译处理的第二步是编译调用子程序的主程序。用PROGRAM语句来告诉编译程序,现在正在编译的是一个主程序。在任何可执行语句之前的PROGRAM语句的前面必须有 \$USES语句。\$USES语句告诉编译程序:主程序需要在 \$USES语言中已命名的代码文件中的子程序。编译程序查出这个已命名的文件并读入所有子程序的名和它们的说明。这样编译程序就能够编译主程序所要用到的子程序引用。\$USES语句更详细的情况在第三章和第十三章中介绍。

如果FORTRAN编译程序未迁到 \$USES语句或 \$EXT语句,则编译程序将试图在当前在被编译的正文文件中寻找完整的程序。

注意:FORTRAN编译程序并不把分段编译的子程序的实际代码合并到当前正在生成的主程序正文文件中去,而是产生一组信息,这组信息告诉连接程序,在连接过程中怎样把子程序连接到主程序中去。

虽然这是一个简单的情况,但这可以用来做多种事情,包括在一分段编译的子程序中调用另一个分段编译的子程序。

当主程序和分段编译的子程序连接成一个可执行的代码文件时,必须把构成一个可执行程序的所有文件都告诉连接程序。

在PASCAL语言编译程序中有一个与FORTRAN \$USES语句相对应的USES语句。它可使FORTRAN与PASCAL程序相连接。连接程序的实际使用在第五章讨论,有关 \$USES语句更复杂的用法在第十三章讨论。