# APPLE II 资料(七)

# APPLE FORTRAN 语言参考手册

# 目　录

# 前　言

本手册描述了 Apple Ⅱ 和 Apple Ⅱ —Plus 计算机上的 FORTRAN程序设计语言。Apple FORTRAN 采用的是美国国家标准 FORTRAN 的子集，即 ANSI FORTRAN 77 的子集。

Apple FORTRAN 对 ANSI 标准子集进行了一些扩充。如：它结合了全部语言中的一些特点，而这些特点是标准子集中所没有的。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 标准子集本身也包括了对全语言所进行的大部分重要修改，从而使它超过了以前的标准子集(即 ANSI FORTRAN 66)。

本手册的作用旨在：

* 帮助你熟悉 Apple FORTRAN 与标准 FORTRAN 77的差异和扩充。

* 帮助你熟悉 Apple FORTRAN 在 Apple Ⅱ 和 Apple Ⅱ—PLUS 上的操作环境。FORTRAN 使用 Apple Pascal 操作系统(OS)。

* 如果你不熟悉较新版本的 FORTRAN，本手册还向你介绍了 ANSI FORTRAN 77 与 ANSI FORTRAN 66 的主要差别。

* 为您提供完整的 Apple FORTTRAN 语言说明。

完整的 APPLE FORTRAN 文本还包括另一本手册：

«APPLE 语言系统的配置和操作手册»

为了使你熟悉 Pascal OS，这本 FORTRAN 手册会引导你去参阅有关 Pascal 文献，也就是 APPLE 语言系统中的 Pascal 手册。

特别要注意，为了运行 FORTRAN，你必须先配置你的 FORTRAN系统。该系统被安放在两个软盘 (FORT1：和 FORT2：)上。FORT2：上放 SYSTEM，COMPILER 和 SYSTEM，LIBRARY。FORT1：上放 FORTLIB，CODE。配置系统的指令在附录 A中，第一部分适用于单驱动器用户。第二部分适用于多驱动器用户。附录 A 中提出的配置假设 FORT2：就是你的系统盘。这本手册也就是按照这种观点来写的。对 FORTRAN 系统的配置，可以有不同的选择。在种情况下，你必须知道手册中是把 FQRT2：作为建立系统的软盘来使用。

还要注意的是：如果你以前没有使用过 APPle Pascalo 附录 A 不仅有该 OS 的指导还有 APPle FORTRAN 的启动过程，第一部分适用于单驱动器用户，第二部分适用于多驱动器用户。

APPle FORTRAN 所依据的国际标准是：

ANSI X3，9—1978，美国国家标准程序设计语言 FORTRAN 1 出自美国国家标准协会，Inc，1430 Broadwa New York，Nee York 10018

# 第一章 概 述

## § 1 引 言

Apple FORTRAN 是在 Apple Pascal OS 上运行的一种程序设计语言，通常这个功能很强的 OS 足以支撑多种语言而不只局限于 Pascal，把 FORTRAN 语言置于 Appel Pascal OS 上，对程序设计有以下几个明显的好处，如：

* 完整的 FORTRAN 程序开发设施(包括一个正文编辑程序，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。

* 可以很容易地搞一个 FORTRAN 系统的"开关键"，这个"开关键"可以使系统在开机时立即开始运行一个给定的程序。

* FORTRAN 或 Pascal 语言在同一个 Apple 上操作。

* 运行FORTRAN 或 Pascal 程序，只需学会一种操作系统。

* Pascal 例行子程序可以链接到 FORTRAN 程序上，反之亦然。

* 汇编语言子程序既可以链接到 FORTRAN 上，也可以链接到 Pascal 上，或两者都链接。

还有另外一些优点：Pascal 操作系提供了许多较理想的功能，这些功能已加进 Apple FORTRAN 中，这些扩充使得 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，也产生了对 FORTRAN 的两个小小的语言限制，这将在下面的""Apple 与 ANSI 77 Subset FORTRAN 的比较"中讨论：

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，还有几点不大的差别将在本手册的后面讨论。由 FORTRAN 编译程序和 Pascal 编译程序产生的输出代码是同样的。FORTRAN 和 Pascal 所建立的代码文件是由同一个操作系统交替进行管理的。

本程序开发系统由编辑程序、连接程序，文件管理程序和一些其它的实用程序及库程序组成。程序开发的顺序是：

* 用编辑程序写 FORTRAN 程序。

* 用文件管理程序来存取软盘上的文件。

* 用 FORTRAN 编译程序把正文文件翻译成代码文件。

* 用连接程序将各代码文件连接成一个可执行的代码文件。

* 执行该程序。

为了将 Pascal 和 FORTRAN 的程序连接在一起，或将 FORTRAN 程序与汇编语言程序链接在一起，还需要一些其它的附加措施。

## § 2 本手册的使用

下一章将讨论在 Pascal 操作系统环境中，FORTRAN 语言的使用。它将告诉你在有关

Pascal 操作系统的文献的什么地方去寻找查看最重要的特性。

本手册第一节剩下的部分和第三章至第五章讨论了怎样有效地编制 FORTRAN 程序，FORTRAN 编译程序输入的条件及连接程序的使用。本手册的下一个主要部分(第六章至第十四章)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。

剩下的章节(十五、十六章)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一些 Apple 的特殊功能。

在手册的其它地方给出了附加的摘要，提供了其它一些有用的东西，新的 Apple Pascal/FORTRAN 用户在想要使用系统之前，一定要先阅读附录 A。

Pascal 和 FORTRAN 所共用的操作系统的所有部分，已经写在 Pascal 操作系变的文献中，为了学习怎样建立编辑和运行 FORTRAN 程序，你必须有一本 Apple Pascal 参考手册的副本，或一本 Apple Pascal 操作系统手册的副本，注意，后一手册是前一手册的Pascal 操作系统的补充。

附录 A 还包括了一些指导性的内容，如怎样开始使用程序开发系统、编辑程序、连接程序等，还提供了有关运行单驱动器或多驱动器系统所需的全部信息。

Pascal 操作系统的文本描述了两点，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，在学习编辑程序，文件管理程序和连接程序等内容时不会产生什么问题，因为这些工具与所用的程序设计语言无关。

pascal 文献中的另一部分是专门面向 Pascal 语言用户的，本手册中抄录了这些章节的某些部分。在有些情况下，将要求你去读 Pascal 文献中的特殊章节中有关“FORTRAN 彩监色视屏幕”，软盘与软盘之间名的转换，FORTRAN 字与 Pascal 字之间的转换，以及诸如此类的东西。

## § 3 什么是APPLE FORTRAN?

FORTRAN 的历史几乎比所有的其它高级程序设计语言都要长。因此，它经历了几个发展阶段。1966 年美国国家标准化协会(ANSI)颁布了标准 FORTRAN，对人们理解该语言有很大的帮助。这就是通常说的 FORTRAN 66。

从那以后，该语言在继续发展，出现了大量的扩充部分，这些部分很有趣，并且具有普遍性。这样，1977 年 ANSI 制定了另一标准，称作 FORTRAN 77。这一标准包含了那些扩充部分，这个更新的标准现在开始被广泛地接受，Apple FORTRAN 是基于ANSI FORTRAN 77 全部语言上的一个正式的 ANSI 子集。FORTRAN 不断地发展不断地产生新的东西，所以，几乎每一种 FORTRAN 都具有一些特点，这些特点是所使用的处理机所特有的。Apple FORTRAN 当然也不例外。

因此，对于熟悉其他 FORTRAN 版本的用户来说，重要的是要搞清楚这样的概念：Apple FORTRAN 与其它版本的 FORTRAN 究竟有什么差别，这要分三个问题来回答：

* Apple FORTRAN 与 ANSI FORTRAN 标准子集之间的差别何在?
* ANSI 子集与全语言之间的差别何在?
* ANSI 66 与ANSI 77 的差别何在?

我们依次讨论这里的几个问题。

## § 3.1 Apple与ANSI FORTRAN77子集的比较

尽管 Apple FORTRAN 大部分是和 ANSI 标准子集一致的，但也有一些小小的差别，它不具备某些特性，但它从全部语言说明中取了另一特性。在某些情况下，它还超出了标准的范围。

在下面两种情况下：Apple FORTRAN 与 ANSI FORTRAN子集不同。

* 整型和实型数据使用存贮单元的数量不同。ANSI 规定它们必须是相同的。Apple 中实型数据占有存贮器的 4 个字节。而整型和逻辑型则占 2 个字节。这对整型数据就意味着可表示数的变化范围是-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。

* 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，但这些功能虽在全 ANSI FORTRAN 77 语言中，但却不在子集中。这些功能是：

* 下标表达式 —— Apple FORTRAN 和全 ANSI 77 语言允许在下标表示式中出现函数调用和数组引用。

* 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。

* I/O 设备号——Apple FORTRAN 可以用表达式来指定一个 I/O 设备。

* 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，但它们不能以左括号开头。注意对于这样的表达式：(A + B) * (C + D)可以在输出表中描述成：+ (A + B) * (C + D)，从而巧妙地解次了这个问题。附带说一句，在运行时，对前面的"+"号求值不产生任何代码。

* 在计算型 GOTO中的表达式——Apple FORTRAN 允许表达式作为计算型 GOTO 的计算值。

* 一般的 I/O——Apple FORTRAN 对顺序存取和直接存取文件可以是规格化的，也可以是非规格化的。子集语言则要求直接存取文件是非规格化的，顺序存取文件是规格化的。这里(Apple)还增加了全语言中的 OPEN(打开)语句来接收附加的参数，而在子集语言中是不包含这些附加参数的。这里还提供了 CLOSE (关闭)语句，也是子集语言中所没有的。I/O 的更详细的解释在第十一章中。

* CHAR 内部函数——Apple FORTRAN 含有 CHAR 内部函数。

在某些情况下，Apple FORTRAN 所具备的一些特性，是 ANSI 标准子集和全语言中都没有的。这些扩充是指已经增加的编辑程序控制命令，它允许你传送某些信息给FORTRAN 编译程序，增加了编译程序控制命令行 它由编译程序识别并接收。有关这些命令的叙述可参见第四章。另外，Apple FORTRAN 还含有内部函数 EOF。

## § 3.2 ANSI 77 与全语言的比较

为了帮助你弄清 ANSI 标准子集中的有用的特性，本手册有两个关于子集语言摘要的附录。附录 D 给出了完整的子集语言语法图，同时附有 Apple FORTRAN 的说明，附录 E 给出了子集语言中所有语句列表和它们的语法。

## § 3.3 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 HOLLERLTH 数据类型)将在附录 F 里讨论。ANSI 77 所增加的性能和在 ANSI 66 中未定义的部分已经阐明。

# 第二章　FORTRAN读者指南

## §1　预备知识

正如前面所说的，使用本手册，应同时参考任意一本 Pascal 手册，以获得使用 Apple FORTRAN 的一个完整的印象。这一章有两个目的：一是帮助你将手册中所讲的东西列一个表，二是帮助你了解手册中的 FORTRAN 术语。

对于编辑程序，文件管理程序、连接程序和其它大量的操作系统的概况，Pascal 文献中都作了完整的描述，文献中还给出了程序例子和软盘(供 Pascal 进行操作的)名；对FORTRRAN 多数情况下，只需解释一下 FORTRAN 与 Pascal 之间软盘名字的转换。在不容易讲清楚的地方附有例子以供参考。所有这些都将在本章予以讨论。

下面是一些 Pascal 操作系统上 Pascal 和 FORTRAN 语言之间关系的要点。

* 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SYSTEM LIBRARY，它包含了一些专用的单位，还有RTUNIT. CODE，都是只用于 FORTRAN 的。另外在 Pascal 语言系统中还有一个叫做 SYSTEM. SYNTAX的文件。它与 FORTRAN 丝毫无关。

* Pascal 文献用到了伪代码—P—代码。FORTRAN 和 Pascal 编译程序都产生 P——代码，而不产生 APPle 原有的 6502 机器代码。编译程序产生的 P—代码由 P—代码解释程序执行，P—代码解释程序把 P—代码指令翻译成 Apple 原来的机器代码。这样，FORTRAN 和 Pascal 便都可以在 Pascal 操作系统上运行了。

* 本操作系统使用 2 种基本软盘文件：TEXT 和 CODE，TEXT 文件采用人们可以阅读的格式，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你编辑具有后缀. TEXT 的文件，连接程序仅仅连接含有后缀.CODE 的文件或有后缀.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使用它们，因为该系统很注意文件的后缀。所以可为你改变任何已有文件的后缀。一般情况下，应确保正文文件有后缀.TEXT，2 进制代码文件有后缀.CODE。

## §2　Pascal 文献指南

接下来是一个 Pascal 操作系统的阅读指导。它提出了为解次 FORTRAN 的各种应用

所应阅读的内容，及所要进行的具体的替换和改写。不论读了那本手册的引导部分，你对Pascal的操作系统都必须有一个大概的了解

## §2.1 命令层

为能完全应用本系统，请阅读你所持有的Pascal手册的全部章节。命令层中的“开关键”系统部分在两本Pascal手册中都没有谈到，因为这些对于FORTRAN用户来说是很重要的，所以本手册包括了这部分内容。

Apple Pascal系统允许你建立一个开关键系统，当Apple开机时该系统能自动地开始运行一个特殊的程序。要为你的Apple建立一个开关键系统首先必须形成软盘FORT 2：上的所有文件副本，但SYSTEM。COMPILER除外，因为开关键系统并不需要它，使用文件管理程序中的C(chaege)命令来改变软盘名。例如，你也许想给副本取名为TURNKEY:，然后传送(Transfer)你的程序代码文件的副本给开关键软盘，传送命令是T。你的程序的新副本必须取名为SYSTEM。STARUP。

## §2.2 文件管理程序

为了充分应用本系统，请阅读文件管理程序一章，记住，文件管理程序是在软盘FORT 1:上。所以，在能够调用文件管理程序之前，该盘必须在你的某个驱动器中。

## §2.3 逻辑程序

为了充分应用本系统，请阅读编辑程序一章，须知编辑程序是放在软盘FORT 1:上的。

在“正文修改命令”一节中，在I(插入)之下，用A(anto—indeutu TRUE)，F(filling FALSE)插入。注意，这些通常就用来写FORTRAN程序及Pascacl程序。

## §2.4 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，在这章的末尾给出的许多程序例子却是用Pascal语言写的。汇编语言程序ASMDEMO用于FORTRAN时需要改变几行代码才能工作。因FORTRAN要求而改变过的ASMDEMO程序将在本手册的第十五里叙述。

## §2.5 连接程序

Pascal文献中有关连接程序的一节，本手册中以第五章代之。

连接程序将分段编译的代码文件连接成一个可执行的代码文件，Pascal和FORTRAN

系统的连接程序是同一个。因为 Pascal 和 FORTRAN 编译程序产生的目标代码文件具有相同结构，所以它可以正确地进行连接。不过 Pascal 和 FORTRAN 使用连接程序的方法有所不同。详细说明请看本手册第五章。

## § 2.6 实 用 程 序

为了充分应用本系统，要看一下新软盘规格化一节。Pascal 和 FORTRAN 所用软盘的格式及规格化程序都是相同的。要注意 FORTRAN 的代码文件被放在软盘 APPLE3: 上，这个盘在完成规格化之前，必须放在某一驱动器中，这个软盘是以 Pascal 程序包来提供的。(请参看附录 A)。

如果你需要建立、扩充或变更库文件，就应阅读系统库管理程序部分。有个叫做 SYSTEM.LIBRARY 的主库，它是建立在软盘 FORT 2: 上的，Pascal和 FORTRAN 用同一个管理程序来使用这个库。然而，库管理程序是建立在 FORT 1: 上的，FORTLIB.CODE，APPLE3 上的 LIBRARY.CODE 仅能用于 Pascal。库的结构是相同的，但 FORTRAN 中的 SYSTEM. LIBRARY 的内容稍有差别。举个例子说，FORTRAN 里有实用 RTUNIT 而 Pascal里没有。

使用库管理程序的读者还应看看库映象部分。这部分描述了一个允许你观察任何库的内容的程序。

如果你准备使用一个非标控准制台，那你将需要重构一部分操作系统，以便用这个控制台来代替 Apple 控制台。你可以象在系统重构和改变 GOTOXY 通讯系统两节中所讨论的那样来重构 Pascal 操作系统。

# 第三章　程序分段

## §1 引　言

FORTRAN 系统提供各种方法让你选择书写和构造程序的步骤。最重要的一点是对于大的程序编制课题，你可以从主程序里移出子程序集，并把它们放入一些独立的模块中。这些模块可以分别设计，甚至可以用于多个程序。把代码块作为模块来处理的办法比写一个大程序要有若干优点：

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，从而加快研制的速度。

2、能够用于许多程序中的子程序。只要编写一次，然后放在库中，便可供以后使用。

3、已经写好的许多常用的子程序，放在系统库里。它们可以直接用在你的程序中。记住，该库在软盘 FORT 2：上。

4、用 Pascal 和汇编语言编写的过程可以连接到 FORTRAN 语言主程序上去。

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，从而避免了磁盘或计算机内存的存贮容量因程序大而超容。这些方法在这一章的后面将有更充分的讨论。

程序分段使得处理作为模块的代码块成为可能实现的事。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 操作系统的性能的使用。

## §2 分段编辑

一个 FORTRAN 程序由一个主程序和它的相分的在程序组成。最简单的情况是主程序和所有子程序都出现在被称作源代码的正文文件中，一次编译的结果是叫做目标代码的文件它包括完整的程序，并被放子软盘上，当准备执行该程序时，把它一次全部装入内存，并开始执行。被装入内存的程序叫做代码映象。

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，下面将讨论如何利用各层的作用和特点。

### §2.1 源代码分段

可以把一个程序的任何部分分离出来，各自放入不同的正文文件中去。在被移去的程序部分的位置上，填入专门的语句，使得 FORTRAN 编译程序编译编译时那个分段的文件。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，并要保证编译程序重新将该程序集中起来时，这个文件集仍是一个完整的 FORTRAN 程序。

把程序分割成模块有两个作用，首先，你不必去编辑整个程序，其次可使一程序中的不

同工作分别出现在独立的正文文件中。这就有利于保持程序行组织结构的清晰。要了解更多的有关把程序分割为模块的内容，请参看第四章有关 $INCLUDE 语句的讨论。

## § 2.2 目标代码分段

如果在正文文件中的源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报儿编译。把这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被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是不必关心的，但所有的段都必须提供给编译程序，以保证子程序的完整性。

在最简单的情况下，分段编译按下面两个步骤进行。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不止一个完整的子程序。编译开始时，它发现不是主程序(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，当一个子程序成功地编译完时，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。连接程序包还包括在编译单位里的子程序的有关详情(如它们的类型，变量的数目第)。

分段编译处理的第二步是编译调用子程序的主程序，用 PROGRAM 语句来通知正在编译主程序的编译程序。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 语句中命名的正文文件的子程序。编译程序查出这个命名的文件并读入所有子程序的名和它们的说明。接着，编译程序编译主程序调用子程序的引用。$ USES 语句更详细的情况在第四章和第十四章里描述。

如果 FORTRAN 未遇到 $USES 语句或 $EXT 语句，那么编译程序将试图在当前正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，这组信息将告诉连接程序，在连接过程中怎样把子程序连接到主程序中去。

好，这是一个简单的情况，可利用这一设施做各种事情，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。

当主序程和被分段编译的子程序将被连接成一个可执行的代码文件时，你必须把构成一个可执行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 FORTARN $USES 语句相对应的设施。称之为 USES语句，它使得可以将 FORTRAN 和 Pascal 程序相连接，连接程序的实际使用在第五章讨论，有关 $USES 语句的更复杂的用法在第十四章讨论。

## § 3 单位段和程序库

当一个或多个子程序在一起编译时，形成一个编译单位；一个分段编译的子程序所生成的代码文件包含了一个编译单位，编译单位这个术语不可同 FORTRAN 的 I/O 单位的概念相混淆。I/O 单位是指一个特定的 I/O 设备。而一个编译单位却是代码文件的一部分。