FORTH SYSTEM

FORTH系统

杨健铎　李福顺　编著

学报出版社

FORTH系统

杨建钎
李福顺
编著

宇航出版社

内 容 简 介

FORTH系统为第四代软件，它是一种高级语言，又是一种汇编语言；既是操作系统又是一种开发软件的研究工具，同时也是一种新型软件设计方法。本书所涉及的FORTH字汇，以美国FIG公布的标准模块FORTH79为兰本，参考了有关FORTH系统的其它资料，并在APPLE—Ⅱ微型机上调试运行后，整理编译成书。全书共分八章，包括FORTH字汇、编辑字汇、汇编字汇、FORTH系统中各种类型的数及应用中的问题，还特别在第八章中介绍了浮点操作。

本书特别适合于从事工业控制、现代化企业管理及其它现代化事务处理的工程技术人员、管理人员、大专院校师生及从事微电脑开发的人员阅读。

FORTH系统

杨建铎　李福顺　编著

责任编辑　廖寿琪

•

宇航出版社出版

新华书店北京发行所发行

各地新华书店经售

天津静一胶印厂印装

•

开本：787×1092 1/32　印张：12 1/4　字数：273千字

1986 年 1月第一版第一次印刷　印数：1—8,000册

统一书号：15244·0022　定价：2.80元

前　言

微电脑是一种引人入胜的新型逻辑设备，它的问世使人们的生活方式发生了巨大的变化。在国际上，微电脑已进入社会化、普及化阶段。在中国，目前正以飞速发展的姿态进入工业、交通运输、能源、教育、科学技术、国防等各个部门，尤其是在教育领域中的发展趋势，简直使人难以置信。微电脑日益增多，新的应用成果和用微电脑做成的新产品每时每刻都在不断地涌现出来。中国要实现四个现代化，而现代化这个概念就意味着生产活动的各个领域跨入自动化、信息化、电脑化的科学时代。评论家指出：以电脑为基础的现代化信息技术革命是解放人类脑力劳动的划时代科技革命，是继解放人类体力劳动的蒸汽革命、电气革命之后的又一划时代科技革命。微电脑的蓬勃发展，又迅速地推进科学、技术、生产、经济的发展。

为了适应我国的微电脑发展形势，本书介绍一种近几年才发展起来的高级专用语言—FORTH系统。FORTH是Charles Moore在1960年为射电望远镜编写控制程序时设计的。1978年有经验的程序专家组成一个团体叫做FORTH兴趣小组（FORTH Interest Group）开始设计一种较大的FORTH专用语言，另外的一个欧洲FORTH用户团体（European FORTH User's Group）也在积极研究和设计这种系统。本书目的在于使读者能更好地应用微电脑设备解决各自需要解决的问题，帮助读者能尽快成为一名不同专业的专用软件设计人员。

本书在序言部分讲述FORTH系统的概貌，主要篇幅

用来介绍FORTH系统的基本单字的使用。在第七章介绍FORTH系统的应用和其系统内部结构情况，帮助读者在熟悉应用的基础上进一步剖析系统。在第八章介绍浮点操作并公布FORTH浮点程序清单，目的是给出一个完整的例子以便了解该系统的扩充能力。作为系统开发和研究工具，在第八章也给予了充分的证明。

高级语言FORTRAN、COBOL、PASCAL、Ada和FORTH都在迅速发展。一些专家预测．高级语言正向两个方向发展，一方面是发展通用性语言，如Ada；另一方面是发展专用性语言，如FORTH。FORTH将语言编译、设备管理、处理机调度等多种功能汇集于一体，它能够脱离传统的操作系统，独立对计算机进行管理，能同时使用汇编语言、FORTH高级语言和机器语言进程，用户可通过内核字汇直接控制硬件功能。该系统具有短小、快速、系统结构紧凑、时空开销低、程序开发时间短、移植方便等优点，因而该语言特别适合作为微电脑和小型计算机进行过程控制、事务处理、数据处理、设备管理等，也是软件的一种独特的开

FORTH系统的应用和开发已在国际计算机界产生巨大影响。在美国的加州的.Martin&Stern公司利用FORTH语言发展了轻型的便携式战场处理机，这个名叫Parts的系统重量不到23公斤，采用两片Intel8162微处理器负责处理多台传感器送来的数据，并瞬时转换成战场作战信息。密执安州Allen测试设备公司采用8080计算机和FORTH语言生产了一种汽车发动机测试仪Smartscope，15分钟能测出汽车故障。另外，在电影与电视片拍摄系统、商业系统、艺术界、航天技术界等领域均有成功应用

FORTH系统的范例。

目前，我国各个行业应用FORTH系统的呼声很高，许多单位在应用FORTH解决实际问题的尝试中已经看到该系统所独有的特点。上海的工业大学，北京的清华大学等学校都在应用FORTH系统方面作出了很好的成绩。

本书以FORTH（79）标准文本结构资料为依据，参照近几年来国际上FORTH系统发展的有关资料，编写时在“APPLE II PLUS”微电脑上做了大量的运行和扩充试验，并成功地编译了FORTH系统浮点处理方式。现将该系统的通用资料公开发表，提供给各行业、各部门使用和参考。

在本书编写之前，北京空气动力研究所的FORTH兴趣小组汪艺云、居绍一、王荣、冯克仁等同志做了许多工作，对本书的编写起到了推动作用。在编写本书过程中，所列举的程序实例都已经调试完毕，可以运行，这是乔占清、万美华、孙粤梅等同志的工作。另外北京市技术交流站、北京市测控中心、北京大学等单位的有关同志、老师对本书的编写分别都给予了不同形式的大力支持、帮助和鼓励。在此对上述这些同志一一表示衷心的感谢。

由于FORTH系统还在不断发展，编者水平有限，编写时间又十分仓促，书中难免有不少谬误和不妥当之处，恳请读者批评指正。

目　录

第一章 序 言

FORTH语言是以一种独特风格建立计算环境的计算机高级语言，它与诸如BASIC、PASCAL、FORTRAN等高级语言极不相同。它融合计算机语言、操作系统、正文编辑、监控程序、汇编语言等五种系统为一体，形成了一个完整的计算机系统。因而该语言既是一种计算机高级语言，同时也是一种软件设计方法。该语言内部结构严谨，用户使用、操作方便，能满足不同专业的各种需要。

FORTH语言的许多特点不能在此一一列出，仅举几个读者所关心的事例。

FORTH是可“扩充的”。这个特点意味着用户可以在这个系统上增加自己所需要的专业软件，允许用户定义自己的小型数据库，甚至定义自己的计算机语言，以适应各自的专业需要。简而言之，该系统能很快地帮助用户成为一名软件设计人员。

FORTH语言允许用户构造自己的操作和数据类型的字，叫做“定义字”。FORTH程序是由系统原有的字定义的一系列新字组成。这两种字（系统原有的字和新字）构成FORTH正文字汇表。在编写语言程序过程中，字汇表不断增加，这正是不同专业人员扩充本专业应用软件的前提。一旦这种扩充工作经试验、校正，即可输出文本作为本专业的专用软件。在今后的工作中，还可以根据需求的增加不断扩充这个新软件版本，从而可大大提高工作效率，完全回避了一些不必要的重复工作。当然，FORTH也设计了

这样的功能，可随时将所扩充的一部分或全部字定义从原有系统中删除。

FORTH语言代码在存贮上是极其压缩的，它编译的目标代码可以与机器语言相比。全部FORTH只占8KRAM。尽管如此，在大多数请求运行的时间里，仍然不一定全部都需要，因而又存在着一种专门的编辑技术来压缩这8K代码。

FORTH语言是结构式的语言，它不存在“GOTO”语句，备有“IF”和“ELSE”控制结构及“DO……LOOP”、“BEGIN……UNTIL”、“BEGIN……WHILE……REPEAT”循环，所有这些结构都可以嵌套任意的层次。模块之间信息的传递完全依靠一种“栈”来完成。

FORTH语言使用一种“字典”结构方式。字典保存了该系统的所有定义(单字)，既包含了系统原有的定义，也包括用户的扩充定义，这些定义都以“字典条目”的形式安排在字典里。执行FORTH程序的过程，就是“解释程序”的检索“字典”过程。这种“检索”过程也就是程序所涉及单字的“地址链”的链接过程，因而当在终端键盘上把编制的程序输入完毕时，整个程序的“链接”也就完成了，也就是说，此时已完成整个语言程序的编译工作。实践证明，FORTH与BASIC相比，在程序设计条件完全相同情况下，FORTH的运算处理速度比BASIC约快十多倍。

FORTH使用一种“存贮栈”结构及与之相配合的后缀表示法（逆波兰表示法）。这就使得子程序（单字）之间的链接只付出很少的内部操作。当一些操作符号（单字）对存在栈里的数值进行操作时，尽管程序给出的表达式比较复杂，但是由于许多操作符号（单字）之间存在着相互作用，致使最后的操作仍然十分简单。

“存贮栈”结构的特点更有利于“模块式”程序设计，以致使调试程序时具有很高的可靠性。每一软件模块（单字）都能独立地进行调试和检查，所有的模块（单字）与外界的通讯都依赖于一种内部栈来完成；每一模块（单字）又依赖于比较早期已经调试过的模块，按照先后次序，新字的检查和调试又反过来帮助捕捉任何早期工作中还存在的错误。几乎全部的FORTH单字都如同键盘命令一样直接执行。任何参量都可以在使用之前存在“栈”上去，或者通过另外的命令（单字）产生，其结果都立即打印出来。任何新的定义的一部分都可在键盘上交互实现，这就非常有利于程序设计的调试工作。

FORTH语言对数据的存取相当方便，所有的存贮器和输入、输出缓冲区都可以直接进行通讯。

如果需要整个机器的高速度，FORTH语言又可包括一种FORTH汇编语言，通过汇编语言建立起来的程序也是一些FORTH单字名字，其特性精确得如同规则的FORTH字定义一样。但是，在一般情况下，一个应用程序应当尽可能完全使用FORTH单字来写。如果还想提高速度的话，程序的一部分应该换成汇编语言。

FORTH语言在一些机器之间具有灵便的“可传递”性。FIG（美国FORTH兴趣小组）公布的标准模块在一般的微型计算机上实现起来比较方便，而对小型或大型机器也能实现。目前，已公布的标准模块全部使用FORTH写成，不带有对机器代码或者另外条件的依赖性。

由于FORTH语言采用后缀表示，这与通常书写表达式的习惯不同，因而会由此引起一点麻烦。但是，这种后缀表示正好适应了实现工业控制的需要。况且，人的习惯并不

是不可以改变的，只要稍费一点精力，就能掌握变一般表达式为后缀表示的书写法则。

本书针对装配在APPLE I 微型机的FORTH系统而写，书中所介绍的所有单字，在操作系统DOS.3.3支持下均能运行或者由用户自行扩充后使用。

第二章　概况、约定、后缀表示法则、栈、字典。

§1　概　　况

何谓FORTH程序？用文字概括起来有两条：第一，定义要开展的工作（过程），对每一工作（过程）起名字，作定义。第二，把这些带名字的工作（过程）进行归纳，使之成为较大的工作（过程），再给这比较大的工作（过程）起名字，作定义，如此等等，依次下去，直到最后整个程序还是一个单字。

作为例子，编一个FORTH程序，使它在屏幕上显示一个大写字母“F”，字母“F”由一些“*”号组成。

```
0 ( LARGE LETTER—F )
1 : STAR 42 EMIT ;
2 : STARS 0 DO STAR LOOP ;
3 : MARGIN CR 30 SPACES ;
4 : BLIP MARGIN STAR ;
5 : BAR MARGIN 5 STARS ;
6 : F BAR BLIP BAR BLIP BLIP CR ;
7 F
8                    *****
                     *
9                    *****
                     *
                     *
```

程序解释和分步调试：0行是注解行，说明本程序要打印一个大写字母“F”。注解行可以放到程序的任何地方，但是一般不能跨行，注解行用“（”和“）”括起来。1行是新定义“STAR”，它负责转换出一个“*”符号。在键盘上打印“STAR”，然后按“RUTERN”键（以下简称®），在屏幕上显示“*”，即表示该单字已调试完毕。在这里使用了ASCII符号“*”对应的十进制数“42”。操作过程和结果显示可用下面的FORTH符号串表示：

STAR ® * OK

2行是新定义“STARS”，当在栈里给定循环终值时，对循环体“STAR”进行由“0”到指定循环终值的循环。可在键盘上这样调试这个单字：

5 STARS ® *****OK

3 STARS ® ***OK

这就表示单字“STARS”的设计功能已经实现。3行是新定义“MARGIN”，它的功能应当是“回车换行”并印出30个空格，可以这样调试它：

MARGIN ®

OK

单字“MARGIN”由基本定义“CR”和“SPACES”以及参数“30”构成。4行是新定义的单字“BLIP”，它由已经定义好的字“MARGIN”与“STAR”组成，功能是回车，换行，印出一个“*”号。5行是新定义的单字“BAR”，它由刚才已经定义的字“MARGIN”和“STARS”以及参数5组成，它的功能是回车换行并在30个空格后印出5个“*”符号，可以这样进行调试：

BAR ®

```
*****OK
```

6行是一系列已经定义完毕的字定义的归纳，给这样归纳起个名字叫“F”，当执行“F”时，应当分别按这些已经定义的字定义出现的顺序，实现各自的功能。“F”就是整个程序的名字，这个新名字便能在屏幕上显示大写字母“F”的整个程序的总名字。

```
F ®

*****
*
*****
*
*
OK
```

在以上的程序解释中，从1行到6行每行中的开始出现的号冒“：”和结尾分号“；”都是FORTH系统的基本字，在后面的正文中还要详细解释，在这里谈到程序的字定义的组成时，没有把这两个单字计算在内，确切的说法，如单字“STAR”应当由“：”、“42”、“EMIT”、“；”组成，在这里由于是简单的概况介绍，为避免重复，因而在谈新单字组成时，没有计算这两个基本的单字在内。

总之，简单的FORTH单字，可以构成较复杂的单字，如此下去，当对这一系列单字列表时，就构成了一系列功能越来越强的单字定义，更确切地说，是一系列按顺序完成的模块结构，其整体仍然是一个单字（模块）。

由以上的程序可以看出下面的几个问题：

① 每个字都是相对独立的，每个字可以单独地进行调试，如：

STAR ® ＊ OK

5 STARS ® ＊＊＊＊＊OK

② 整个程序在设计思想上采取“自顶向下”设计，而在编程序时，却采取“倒置编程”，也就是说，从系统基本字开始作新的字定义，一级一级地朝上推的。

③ 用户定义的新字和在程序编写中用到的系统基本字交互组织运用，对系统来说，新定义的字同样进入正文字典，与系统基本字享有同样的工作条件。

④ 程序的最终名称仍是一个字“F”，因而对整个系统而言，编写的程序可归纳为：字→字→字。

⑤ 整个程序的结构如同树的根系一般。（见图2—1）

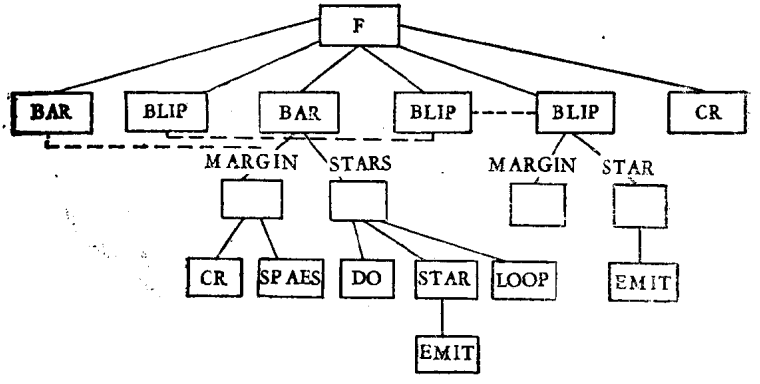

图2—1 “F” FORTH程序根系图

由以上的程序分析可看出这种设计思想的优越性，它提高了程序的可靠性，并增强了程序的修改和调试的灵活性。

程序员可以选择一个名字，代表其预计完成的整个任务的最后结果，其次向下分解任务，使其变成几个部分的小任