

CROMEMCO 微型计算机

软件资料汇编

清华大学计算中心译编

(三)

清华大学出版社

1980

内 容 简 介

以Z-80A为中央处理机的CROMEMCO微型计算机(即M-5或M-8),有较丰富的软件配置。

“CROMEMCO微型计算机软件资料汇编”主要内容是该微型机各种软件的使用说明和用户手册。本书暂分三册,第一册是关于操作系统,文本编辑,宏汇编的使用。这册中还包括由清华大学计算中心编写的“系统简介”,“磁盘操作系统分析报告”及“配接光电输入机的报告”。第二册是有关FORTRAN语言及COBOL语言,还有“屏幕编辑”的使用和用户手册。第三册是有关BASIC和BASIC多用户语言以及数据库管理程序的使用。第四册是有关结构BASIC、跟踪系统模拟程序、文本格式化指令手册和DBR数据库报表生成语言使用手册。

本书的主要对象是该系统的用户和计算机程序员,从事计算机工作人员也可作为参考资料。

CROMEMCO 微型计算机 软 件 资 料 汇 编 (三)

清华大学计算中心译编



清华大学出版社出版

北京 清华园

清华大学印刷厂胶印

新华书店北京发行所发行·各地新华书店经售



开本: 787×1092 1/16 印张13 3/4 字数 299 (千)

1980年6月第一版 1983年10月第四次印刷

印数: 22001~47000

统一书号: 15235·3 定价: 2.00 元

目 录

一、CROMEMCO 16K 扩展 BASIC 说明书及使用手册

第一章 引言	2
1.0 设计原则	2
1.1 计算机和计算能力	5
1.2 高级语言和低级语言	5
1.3 BASIC 和它的应用	6
1.4 BASIC 结构简介	7
1.4.1 语法相关表	7
1.4.2 程序列表	8
1.4.3 用户区中的表	8
1.4.4 例子	9
1.4.4.1 句法检查	9
1.4.4.2 运行	10
第二章 指令的句法	11
2.1 空格和空白字符	11
2.2 大写字符	12
2.3 BASIC 的提示符	12
2.4 命令	12
2.5 语句	13
2.5.1 复合指令行	13
第三章 数和字符串的机器内部表示形式	14
3.1 整数	14
3.2 短浮点数	14
3.3 长浮点数	15
3.4 十六进制	15
3.5 字符串	15
第四章 常数和字符串文字的格式	16
4.1 整数和浮点数	16
4.1.1 浮点数的存贮	16
4.2 十六进制常数	17
4.3 字符串文字	17

第五章 变量的表示方法	18
5.1 数值变量	18
5.1.1 数值变量的格式	18
5.1.2 整型变量	18
5.1.3 短浮点变量	18
5.1.4 长浮点变量	18
5.1.5 矩阵和向量	18
5.1.5.1 格式	19
5.2 字符串变量	19
5.2.1 格式	19
5.2.2 定义字符串变量	19
5.2.3 字符串变量的引用	20
5.2.3.1 格式 1	20
5.2.3.2 格式 2	20
5.2.3.3 格式 3	21
5.2.3.4 格式 4	21
第六章 运算符	23
6.1 算术运算符	23
6.2 赋值运算符	24
6.3 关系运算符	25
6.4 布尔运算符	26
第七章 程序举例	28
7.1 启动	29
7.1.1 命令和直接方式	29
7.1.2 运行方式或程序执行方式	31
7.1.3 启动系统打印机	32
7.1.4 程序编辑	32
7.2 程序举例一	33
7.2.1 用 LIST 命令把程序记入磁盘	34
7.2.2 用 ENTER 命令把磁盘文件送入内存	35
7.3 程序举例二	35
7.3.1 使用 SAVE 和 LOAD 命令	39
7.4 程序举例三	40
7.5 程序举例四	43
7.6 程序举例五	45
第八章 程序调试指令	47
AUTOL	47

BYE	48
DELETE	48
DIR	49
ENTER	50
LIST	50
LOAD.....	51
RENUMBER	51
RUN	54
SAVE	55
SCR	55
TRACE	56
NTRACE	57
第九章 注释指令	57
REM	57
第十章 赋值指令	57
LET	57
MAT	58
第十一章 初始化指令	59
DEG	59
DIM	59
IMODE	60
INTEGER.....	61
LFMODE	61
LONG.....	62
RAD	62
SFMODE	63
SHORT	63
第十二章 控制指令	64
CON	64
END	64
FOR...NEXT	64
GOSUB ... RETURN	67
GOTO.....	68
IF...THEN	69
ON...GOTO	70
ON ... GOSUB	70
STOP	71

第十三章 控制台和数据输入、输出指令	71
INPUT	71
PRINT	73
READ	75
RESTORE	76
DATA	77
第十四章 输出格式指令	77
PRINT USING	77
SPC	83
TAB	84
第十五章 数据文件的输入和输出	85
15.1 数据文件	85
15.1.1 记录	85
15.1.2 字段	85
15.1.3 输入、输出和读、写	85
15.2 软磁盘	86
15.3 建立数据文件	86
15.3.1 打开输入（输出）通道	86
15.3.2 关闭文件指令	86
15.4 机内代码表示法和 ASCII 码表示法的比较	87
15.5 PRINT 和 INPUT	87
15.6 PUT 和 GET	88
15.7 有益的提示	88
12.8 文件指针	88
15.9 顺序文件	89
15.10 随机文件	89
15.11 CDOS 中的 DUMP 公用程序	90
15.12 数据文件输入（输出）指令	90
CREATE	90
OPEN	90
CLOSE	91
PRINT	91
INPUT	93
PUT	93
GET	94
第十六章 编程序用的函数	95
16.1 算术函数	96

ABS	96
BINAND	96
BINOR	96
BINXOR	97
EXP	97
FRA	97
INT	98
IRN	98
LOG	99
MAX	100
MIN	100
RANDOMIZE	101
RND	101
SGN	102
SQR	103
16.2 三角函数	103
ATN	103
COS	104
SIN	104
TAN	105
16.3 自定义函数 (DEF FNS)	106
16.4 字符串函数	107
ASC	107
CHR\$	108
LEN	108
POS	109
STR\$	110
VAL	111
第十七章 系统, 文件状态的指令和函数	112
DSK	112
ECHO	112
NO ECHO	112
ERASE	113
ESC	113
NO ESC	114
FRE	114
IOSTAT	114

ON ERROR	115
ON ESC	116
RENAME	117
SET	117
SYS	118
第十八章 机器级指令	119
INP	119
OUT	120
PEEK	120
POKE	120
USR	121
第十九章 名词解释	123
第二十章 出错信息	130
20.1 致命性错误	130
20.2 用户可捕捉的非致命性错误	132
第二十一章 附录	135
附录一 ASCII 字符代码	135
附录二 程序复盖	137
附录三 在 CDOS 时直接执行 BASIC 程序	138
附录四 不同数值类型的混合运算	139
附录五 BASIC 的零碎空间	141
附录六 Cromemco 16K 扩展 BASIC 中用户有关的区域	141
附录七 确定字符串地址	144
附录八 附加设备驱动程序	144
附录九 设备驱动程序表	148
附录十 附属的输入(输出)驱动程序	151
附录十一 改变输入输出通道的数目	153
附录十二 从 PROM 装入 16K BASIC	153

二、CROMEMCO 多用户 BASIC

第一章 引言	156
综述	156
启动	157
备份	157
第二章 用户的保护	158
目录保护	158
文件完整性的保护	158

预防死锁	159
第三章 操作系统的描述	161
第四章 操作系统的特征	161
提前打入	161
CDOS 调用显示作业信息	162
第五章 多用户 CDOS 命令	162
BACKG	162
BAS	163
CHA	163
DETACH	164
ATTACH	164
KILL	165
LOGOFF	165
MORE	165
PRT:	166
RESCUE	167
SET	167
SYS	168
TIME	168
第六章 多用户 BASIC 变量名和行名	169
变量名	169
行名	169
第七章 多用户 BASIC 指令	170
CLEAR	170
COMMON	171
DELREM	172
EXPAND	172
IF...THEN DO...ELSE...ENDDO	173
LVAR	175
NOLIST	176
OPEN	176
PROTECT	177
第八章 Cromemco 联机的 BASIC 编辑指令	178
EDIT	178
FIND	179
CHANGE	179
第九章 多用户 BASIC 出错信息	180

第十章 多用户 BASIC 的修改	180
设置用户启动命令行	180
限制存取 CDOS 程序	182
限制使用 CHA 命令	183
限制使用 MORE 命令	183
限制使用 KILL 命令	183
限制使用机器级 BASIC 指令	183
磁盘驱动器结构	184
第十一章 硬件结构	184
系统 RAM 板	185
用户 RAM 板	185
磁盘控制器板	186
数据接口板	186
中断优先电缆	187
特殊结构	187

三、CROMEMCO 数据库管理系统使用说明

引言	190
第一章 建立新数据库系统	192
第二章 检查系统格局	194
第三章 向数据库输入新数据	197
第四章 建立分类文件	198
第五章 显示、改变和删除记录	199
第六章 进行数据库查询	201
第七章 对主文件进行修改	203
第八章 改变主文件和驱动器的分配	204
第九章 打印寄信标签	205

一、扩展 BASIC 说明书 及使用手册

1111522

CROMEMCO 16K 扩展 BASIC

说明书及使用手册

第一章 引 言

1.0 Cromemco 16K 扩展 BASIC 的设计原则

由不同的计算机厂商研制出了许多不同的 BASIC 版本，这些 BASIC 的版本已经被人们广泛使用。不同的 BASIC 版本在计算精度、程序的功能、程序设计的难易，执行的速度和 BASIC 系统所占的内存数量等方面具有很大的差异。目前，许多的 BASIC 在其功能方面和当年由 Dartmouth 学院研制的早期的 BASIC 相似。这些 BASIC 一般要求 8K 的内存。也有少数 BASIC 只要求 2K 内存，但是，它们的功能与 8K 的 BASIC 相比，是不同的，或者说是较差的。今天的用户要求有比当初 Dartmouth 的 BASIC 功能更强的 BASIC 版本。功能较强的 BASIC 通常称为扩展 BASIC，但是不同的计算机厂家搞的扩展 BASIC 其特点也有所不同。

Cromemco 16K 扩展 BASIC 是由 Shepardson 微处理器公司专门为 Cromemco 设计的，由于充分利用了 Z-80 微处理器扩展的 158 条指令，因而它具有最高的计算精度，较好的程序设计效能和较快的计算速度。

Cromemco 扩展 BASIC 特别能满足商业方面的要求（如报表和统计），而对工业和家庭生活方面的实时控制来说，也能提供必要的适用性和计算速度。

下面简单地介绍 Cromemco 16K 扩展 BASIC 的一些特点。

Cromemco 16K 扩展 BASIC 的一个主要特点是利用 Z-80 微处理器特有的 BCD（二~十进制）算术指令进行高速的 14 位的算术运算。它的全部算术运算是利用中央处理器的十六个高速寄存器作为中间存贮来实现的。用 BCD 算术运算所得到的速度和精度，在 8080 那样的只具有七个寄存器的微处理器上是不可能得到的。许多 BASIC 利用面向计算机的二进制运算可得到六位或七位精度。使用二进制算术运算有可能获得和使用 BCD 算术运算相同的精度。但是，许多用二进制算术运算的 BASIC，在商业和金融的程序中计算“元”和“分”的相互转换时，会产生明显的误差。而使用 BCD 算术运算的 BASIC 就没有这些转换误差。在 Cromemco 16K 扩展 BASIC 中，除算术运算外，所有的函数运算也能有十四位的精度。而许多其它的 BASIC 版本，虽然也能提供十四位的精度，但却必须大大地降低其运行速度，而且在程序执行中如果经过许多子程序时，常常会达不到足够的准确度。

实时控制的应用中往往要求整数型的、十六位（bit）的（即十进制中四位半）的算

术运算。Cromemco 16K 扩展 BASIC 通过 BASIC 指令使用直接存取、输入/输出和机器代码子程序的方法，提供了这种功能。在实时控制系统的应用中，常常需要存贮很大的数据数组。为了节省执行时间，Cromemco 16K 扩展 BASIC 允许用户灵活地使用和存取六位有效位的短浮点数。这种压缩形式的运算要比 14 位数字格式的运算速度快一些。

此外，和大多数微处理机的 BASIC 系统不同，在 Cromemco 16K BASIC 中，在程序中引用（访问、调用）常数的速度与引用变量的速度相比，二者差不多或者前者稍快些。另外，整型常数所占用的单元不会比变量多。

在 16K 扩展 BASIC 中，由于利用了半编译程序设计，使执行速度达到最大限度。半编译程序设计集中了编译程序和解释程序两者的许多重要的特点。换句话说，把编译程序具有的较高的执行速度这一特点和解释程序具有的会话功能以及程序设计比较简单的特点结合起来。例如，程序可以在执行过程中暂停，以便检查和修改。修改完后还可以命令计算机继续执行这个程序，而不会影响程序的正确执行。暂停、检查和修改程序这一功能使程序设计所化的时间和成本大大减小。16K 扩展 BASIC 的半编译程序的编译工作是在程序每输入完一行后进行的。句法上的错误是在一行输入后即时检查出来的，而不是在程序执行时才检查出来的。因此，引入和调试一个程序所化的时间显著地减少了。

有些 BASIC 版本的功能不完善，没有处理错误信息的功能。使用这些版本时，如果在执行过程中出现致命的错误而使程序破坏或意外地停止运行，那么，用户就需要去找出程序中的故障。另外一些 BASIC 在程序出错时，会给出一个数字，这个数字代表的意思可以从错误信息表中查出。这些功能更强的 BASIC 程序在出错时会回答出英语的字句，说明发生的是什么错误。Cromemco 16K 扩展 BASIC 不仅给出英语的错误信息，而且它重新把该错误语句显示出来，指出错误所在。用这种错误信息来发现错误，效果是十分显著的。在对实时控制系统进行校正时（在这里修改偶然性的错误往往要花很多的时间和代价），这种功能就显得特别重要。在一般的程序设计中，用这种错误信息的会话方式也是十分方便的，可以大大减少程序设计的时间和错误。

在 Cromemco 16K 扩展 BASIC 中，对于特定程序流程，根据其出错的条件，提供了动态错误陷阱法。运用这种错误陷阱的特点，使得上溢和下溢的算术运算错误可以不需操作员的干预而得到处理。对于不熟练的操作员来说，这种方法的作用是不能低估的。例如，在出现一个磁盘读/写的错误时，16K 扩展 BASIC 就编出一个程序打印出下面的信息：CALL MR. HIGGINS (415 - 030 - 1234) IN THE ACCOUNTING DEPARTMENT，这样，细心的熟练的人就能改正这个错误。

在 Cromemco 16K BASIC 中，提供了一种跟踪程序的方法，以帮助确定错误的来源，这种错误在程序中往往是难以发现的。在用这种方法时，已执行过的语句标号被记录下来。在跟踪状态中，FOR 循环会自动地按照嵌套的情况记录已执行过的行标号。

Cromemco 16K 扩展 BASIC 还有较强的软磁盘输入/输出的能力。例如，无论程序和数据都可以存贮在磁盘内。程序可以按机器的内部形式（二进制）或 ASCII 码

存贮。数值数据和 ASCII 码的字符串数据可以用顺序存取或随机存取的方法存到磁盘上。可以同时打开的文件个数，只受用户分配给它的内存大小的限制。而在一个程序运行过程中先后能够打开和关闭的文件个数是不受限制的。如果程序段太长，不能一次存放到内存中，可以通过磁盘文件来实现链接。文件的名称可以是字符串变量，而且可以在运行过程中由 BASIC 程序随时改变之。用 BASIC 程序还可以写出、修改或运行另一些 BASIC 的程序。程序中行的标号可以改变。可以从磁盘中调出两个程序并链接成一个程序。由 CDOS（即 Cromemco 磁盘操作系统）和磁盘输入/输出功能的相互作用所提供的这些能力是 Cromemco 16K 扩展 BASIC 所特有的。

许多 BASIC 具有字符串处理功能。但是，其中有不少是限制了允许存贮在字符串中的字符个数的，例如限制它不能超过 255 个。在 Cromemco 16K 扩展 BASIC 中，可以允许达到 32767 个字符（或存贮单元），它的限制是较小的。还可以利用子字符串来快速地处理字符串。此外，语言的设计特别注意了速度、多功能和精确度。为了增加字符串处理和分类运算处理能力（流量），例如在邮寄名单的处理程序中就常有这种情况，BASIC 提供了专门的、高速度的字符串插索函数。

为了便于阅读和得到清晰的程序文本，在 16K BASIC 中，可以打印出合适的程序清单，它是按统一的格式排列和留出间隔的。在列表输出时，FOR 循环自动嵌套并将循环内的语句向里收缩成锯齿形。在半编译程序工作过程中，程序输入时，唯一能直接存入内存的语句是 REM（注释）语句（此时，存贮空间是压缩了的）。其它所有的语句都是半编译的，是以机器可读的格式存贮的。在列表时，这种机器可读格式转换成 ASCII 代码格式，然后打印出来。在转换过程中，所有的空格都消失了。这种转换过程能够使 Cromemco 16K 扩展 BASIC 以半编译的形式一次存入大型的程序。对比看一下，许多其它的基本 BASIC 却经常保存着一个未经编译的、与打入的字符一一对应的机器内部表示形式。在这种 BASIC 中，如果不小心打入一个无用（多余）的空格，则要求有额外的存贮空间。而 Cromemco 16K 扩展 BASIC 对无效的空格既不存贮也不产生任何影响。因此，在列表输出时，这种无用的空格是不输出的。

在 Cromemco 16K 扩展 BASIC 中，包括了许多扩展 BASIC 版本的特点。这些特点包括：一行可写多个语句；数组可以是一维、二维和三维的；允许自定义 26 个函数；使用 PRINT USING, TAB, SPACE 和 SYS/SET 函数能够获得多样化的打印格式；用 INP, OUT, PEEK, POKE, 和 USR 指令的直接机器语言的会话功能；最后，Cromemco 16 K 扩展 BASIC 还有能适用于各种终端和行式打印机的 INPUT 和 PRINT 的内部程序。

这本手册可以被没有 BASIC 使用经验的用户使用，也可以被有经验的程序员使用。经验少或没有经验的用户应该阅读 1.0 到 1.3 节，然后阅读第七章（程序例题），这些例题是为初学者熟悉 Cromemco 16K 扩展 BASIC 而设计的。第十九章是词汇解释，以备查阅。

初学者和有经验的程序员都可以通过阅读第十五章（数据文件的输入/输出），进一步了解 BASIC 的文件指令和存取方法。

有使用经验而需要进一步深入的用户需要参考第十七、十八和二十一章，以及第 1.4 节。

1.1 计算机和计算能力

计算机是一种按照事先确定的程序实现高速度的算术运算或逻辑运算，或处理代码信息的设备。这种预先给定的程序是一种按逻辑顺序排列的指令集，它指挥计算机去实现某种所要求的操作。

所有的计算机是建立在二进制基础上的。二进制的数字系统只用 0 和 1。二进制数的概念之所以如此有用，是因为它可以用具有两个稳定状态的任何元件来实现（如“开”和“关”）。一台计算机，即使是最复杂形式的计算机，也是由简单地反映“有电压”或“无电压”的逻辑门组成的。这种电压的有或无，称之为“逻辑电平”，在逻辑电路中，用它表示二进制的数字状态。一般来说，这种逻辑电平把“关”状态定义为 0，把“开”状态定义为 1。

二进制数字（通常称为二进制的位 bit），组合在一起，能够用来表示任何大小的数。在一个特定的计算机中，组合在一起的二进制数字的位数称为字长。绝大多数微处理器用 8 位字长。这种由 8 位数字组成的单元称为字节（byte）。如果我们排出计算机的所有位二进制数的各种组合，那么就有 2 的 8 次方或 256 种可能的组合。因此，我们可以在一台计算机中，使不同的字节分别表示相应的不同指令。例如，一条全为 0 的指令意味着停止，而 10101010 这条指令意味着给一个数值加 1。一个特定的字节在这一台计算机中代表一种操作，而在另一台计算机系统中它可以代表完全不同的操作。应当着重指出，所有的计算机都是通过这种由 0 和 1 表示的开关状态组成的二进制语言来执行指令的。

1.2 高级语言和低级语言

正如在引言中所述，为了实现计算或处理信息，必须事先编写程序。程序是用具有严格的文法规则的语言编写的。目前所用的计算机语言可归为一种或两种主要的类型：高级的语言或者低级的语言。在计算机内部，计算机是根据前面已描述的二进制语言实现不同的操作的。这种二进制的语言通常称为机器语言。机器语言是计算机语言中最低级的语言，它也是一系列开关状态的集合。一个机器语言的字是由八个 0 或 1 的任一组合而成的（如：01100100）。正如你所看到的那样，一个给定的机器语言字几乎不可能从它的二进制的字中一目了然地告诉你它的含义是什么。所以，用这种方法编制计算机程序是很费时间而且又很麻烦的。

为了简化编制程序的过程，要求有比较容易送入指令和识别指令的方法。在低级语言中，下一步（进一步）是“汇编语言”，它使用短的，帮助记忆的符号，这种符号是与指令相对应的（由指令定义的）。汇编语言把简单的指令（如 HALT 或 INCREMENT）翻译成相应的机器指令。但是，对于应用来说，这种程序设计的形式还是太不方

便。在汇编语言中，在一个终端上打印出“HELLO”这个字需要多到 50 个步骤。而且，没有充分的计算机经验的人难以把各种具体问题转换成汇编语言或机器语言的计算机程序。

结果，发展了高级语言，它的指令和文法规则易学好用。面向问题的高级语言包括 FORTRAN, COBOL, PASCAL, PL1, 和各种其它语言。这种高级语言通过称为编译的程序翻译成机器指令。任何一种语言，在它化为机器代码之前需要被编译一次或几次的，它就称之为高级语言。

1.3 BASIC 和它的应用

在上节中说到的一些高级语言大多数是为比较熟练的专门的计算机用户而设计的。通常，这些语言中某些部分是很难学的。无论如何，当计算机从设计实验室转向大学的实验室，然后再转向商业、家庭和中学使用时，就需要有一种易于学习、语法简单、建立在英语基础上的程序设计语言。为了满足这种需要，Dartmouth 学院的 John G. Kemeny 和 Thomas E. Kurtz 教授在 1965 年提出了第一个 BASIC 版本。BASIC 是一种高级语言，它容易学，可以应用于绝大多数的计算问题中。

除了容易学习之外，BASIC 还具有许多特点，这些特点使它在两个非常重要的和通用的方面显得很有用，即：分时系统和会话式的程序设计。在 BASIC 中，在程序执行时，通过所谓的解释程序把简单的、英语写的指令一行一行地翻译成机器代码。这种解释系统使它容易中断其转换过程，去处理另一个任务，然后再返回原处继续工作。因此，一个计算机可以用“分时”系统使多个使用 BASIC 程序的用户同时工作。在分时方式中，计算机按特定的顺序处理每一个用户。由于计算机速度高，因此，对所有用户来说，好象它们在同时进行工作一样（即如同自己独占一样）。

BASIC 也特别适合于在会话程序中使用，这种会话程序要求用户与计算机不断地进行会话。会话程序允许用户不断地根据当前的结果去修改校正一个程序。对于一些复杂的问题，在它得到解答之前，需要改变许多参数，会话程序功能对它们是一种很有效的工具。

由于具有这些重要的特性，用 BASIC 写的程序文本（它们之中有许多已公开出版了）已迅速地推广了。许多复杂的商业程序，包括存货控制，薪金支付，货物进出，和会计记账等，都可采用 BASIC。用 BASIC 可以编写出计算机辅助数学（CAI, Computer-assisted instruction）程序和范围广泛的会话游戏，在这种游戏中，用模拟实际中的各种情况，用户可以与另一用户或与计算机进行竞赛。

此外，同一个计算机系统可以用于把多个会计统计程序联成一个整体、生产过程的自动控制、调整能源的使用、博弈游戏等。只需要改变 BASIC 程序即可。随着廉价的由微处理机组成的，分时系统的引进（如 Cromemco 系统计算机的多用户版本），真正的多用户的计算机操作才变成可能，而它的成本只是五年前的一个零头。

1.4 Cromemco 16K 扩展 BASIC 结构简介

本节的内容大部分是为那些在系统的程序设计方面还不大熟练的初学者而写的。同时,也能帮助希望进一步深入的程序工作者了解有关 16K BASIC 内部结构的一些情况。

16K 扩展 BASIC 语言是加州的 Cupertino 的 Shepardson 微计算机系统公司专门为 Cromemco 公司研制的,它是用 Z-80 代码写的可增编译程序。

可增编译程序是解释程序的一种,它不仅能在源程序进入时全面地检查句法的正确性,而且能够对于所引用的未经定义的项目立即作出判断。可增编译程序的第一个好处是它的执行速度。它先把变量和标号分解出来,这样在执行时就不需要花费执行时间去处理诸如变量名的配对查表等工作。

1.4.1 语法相关表

从键盘(或从某些其它输入/输出设备)输入计算机的每一个语句行(或直接的命令指令),都要经过语言的句法正确性检查,并把它们转换成机器内部形式。实际上,进行语法检查的例行程序也在同时把源程序中的各项转换成为相应的内部标志。

BASIC 语法相关表实际上是由一系列的语法表所构成,它们的作用将在下面介绍。

保留名字表

保留名字表是包括 BASIC 所能识别的指令、函数和操作符名字的表。如果用户以 ASCII 码送入的词和该表中某一名字相同,则 BASIC 将它转换成一个特定的内部标志。除了保留名字以外,每一个登记项(entry)也包括所用到的内部标志。

主句法表 如果已找到一指令中的第一个保留名字(关键字),则根据该保留名字对此指令归类。根据指令的类别使 BASIC 转到主句法表的相应入口。在句法表中的登记项中,应该补充入这类指令所要求的在语言中应出现的内容(实际上是它们的内部标志)。如果在送入的行内指定的位置上找不到它所要求的项(内容),则会产生一个句法错误。这个句法表是非常复杂的,譬如,对于一个算术表达式的各种各样的变化情况,它都能辨别和处理。又如,一个子字符串的名字要求顺序地用一个字符串变量、一个左括弧、下标值以及一个右括弧来表示,这样它就能调用一个子程序来处理子字符串。(请注意,这里讲的不是找寻字符串,而是对字符串进行分割和重新组织。)

句法表包括关于指令类型的其它信息,例如该指令是命令方式,还是语句方式(即在程序中的语句),还是二种形式都可以用。它还可以表示此指令是一行中的最后一个指令呢,还是在它后面还可以跟其它指令。

连接的强度表(优先度表)

把它说成是运行的次序表更合适。因为它来决定各个运算符(十、一、AND、NOT 等)的优先度。通过它将不同的运算符和它们相应的优先度的内部标志联系起来。