DOI:10.13657/j.cnki.gxkxyxb.1982.01.010

使用中的Commodore CBM 4040

广西计算中心　　刘连芳

摘　　要

文章通过实例说明使用CBM 4040 机遇到的问题，提供了解决文件结构问题的办法。

• • • • • • •

微型机一般由于速度、字长等因素的限制不宜作大型数值计算，但仍可作一般数值计算。另外，因为它具有较强的文件处理功能，所以在数据处理中非数值计算方面的应用颇受欢迎。加之操作简单、方便，因此，近几年，微型机得到了广泛的应用。

但是，目前有些微型机还存在一定问题。我们仅介绍使用 Commodore公司的CBM 4040中遇到的主要问题（有些问题属于硬件故障），并提出个别问题的解决办法。

一、文件处理功能

CBM 4040机的文件处理功能较差，且相关的一些语句的功能不稳定。

1、文件结构

由于在写入每个数据时，丢失了数据间的分隔符，所以无论是相关文件还是顺序文件，一般简单处理成一个记录只能写入一个数据，方能保证取出时不产生混乱。

例1：有时一个记录可以写入一个以上的数据，读出时不混乱。

```
10 A$="113,456" : B$="654,321"
20 DOPEN#1, "TASK7", W
30 PRINT#1, A$, B$
40 DCLOSE#1
50 DOPEN#1, "TASK7"
60 INPUT#1, A$, B$
70 DCLOE#1
80 PRINT VAL(B$)
```

*注：我们所用CBM 4040只配有BASIC语言

本文1982年7月21日收到

```
RUN
123.456                    654.321
READY.
```

例2，　有时会发生混乱。

```
A=1234                     B=0
5 A$="12": B$="34"

20 DOPEN#1,"D5",W
30 PRINT#1,A$,B$
40 DCLOSE#1
50 DOPEN#1,"D5"
60 INPUT#1,A$,B$
70 DCLOSE#1
80 PRINT"A=";VAL(A$),"B=";VAL(B$)

READY.
```

例3、

```
A=123456                   B=0                    C=0

20 DOPEN#1,"D1",W
30 PRINT#1,12;34;56
40 DCLOSE#1
50 DOPEN#1,"D1"
60 INPUT#1,A,B,C
70 DCLOSE#1
80 PRINT"A=";A,"B=";B,"C=";C

READY.
```

例4、

```
A=1234                     B=0                    C=0

20 DOPEN#1,"D2",L9
30 PRINT#1,12;34;56
40 DCLOSE#1
50 DOPEN#1,"D2"
60 INPUT#1,A,B,C
70 DCLOSE#1
```

```
80 PRINT "A=" ; A, "B=" ; B, "C=" ; C

READY.
```

一个记录只是一个数据，这势必增加存、取数据所需的语句数量，过于繁琐。特别对相关文件，还会浪费存储空间。例如，要建一个人事档案文件，每人的资料包括编号、姓名、性别、年零、职务……,其中有字符串数据,也有数字数据，各项长度差别较大。例如性别代码只占一位，而姓名要占十几位。但不同人同一项数据长度相近。若每个人的所有数据项放在一个记录里，各个记录长度就近似相等，据此定义的记录长度不会造成存储空间的大量浪费。但若一个记录只能放一项数据，所定义的记录长度（选各记录长的最大值）与有些数据的实际长度（例如性别代码）差距很大，就会造成较大的浪费。另外，在相关文件中，一个人的有关信息分为若干个记录，在检索时，需对记录号进行计算，又会大大增加工作量。所以一个记录只有一个数据会造成众多不便，大大降低了文件处理的功能。

对系统设计的这种问题，可以想办法予以解决。

参照“EYC BASIC 编程指南”（广州远华电气公司）中对磁带文件的处理方法——强制加分隔符。（为了便于叙述，以下，我们称之为强制分隔符）。

1)、采用字符常量的形式。

例如：PRINT#1,F$; ","; M$; ","; L$

2)、采用字符串函数CHR$(A)的形式。其中A为分隔符的ASCII码。

例如：PRINT#1, F$; CHR$(44); M$; CHR$(44); L$

其中CHR$(44)即为逗号(,)。

强制分隔符在里写入时是字符串常量或字符函数，但在读出时是分隔符。

与“EYC BASIC 编程指南”所介绍方法不同的是CBM 4040不能用空格作强制分隔符，无论是对顺序文件还是对随机文件；无论是读数字，还是读字符串，均是如此。

例5、

```
A=13      24      35          B=          C=


20 DOPEN#1,"B3",W
30 PRINT#1,13; " "; 24; " ", 35
40 DCLOSE#1: DOPEN#1, "B3"
50 INPUT#1,A$,B$, C$
60 PRINT"A="; A$,"B="; B$,"C="; C$
70 DCLOSE#1
```

例6、

```
A=13      24      35          B=          C=
```

```
20 DOPEN#1,"B9",L13
30 PRINT#1,13; " "; 24; " "; 35
40 DCLOSE#1: DOPEN#1, "B9"
50 INPUT#1, A$,B$,C$
60 PRINT"A=": A$, "B=": B$, "C=": C$
70 DCLOSE#1
```

例 7 、

```
A=  132435          B= 0          C= 0
```

```
20 DOPEN#1, 13, "A7", L53
30 PRINT#1,13," ",24," ",35
40 DCLOSE#1: DOPEN#1,"A7
50 INPUT#1,A,B,C
60 PRINT"A=": A,"B=": B,"C=; C
70 DCLOSE#1
```

例 8 、

```
A=  132435          B= 0          C= 0
```

```
20 DOPEN#1,"A8",L13
30 PRINT#1,13; " "; 24; " "; 35
40 DCLOSE#1: DOPEN#1,"A8"
50 INPUT#1,A,B,C
60 PRINT"A=": A,"B=": B,"C=": C
70 DCLOSE#1
```

分号有时可以作强制分隔符，有时又不可以，所以最好不用、

例 9 、

```
20 DOPEN#1,"A9",L13
30 PRINT#1,13; "; "; 24; "; "35
40 DCLOSE#1: DOPEN#1,"A9"
50 INPUT#1,A,B,C
60 PRINT"A=": A,"B=": B,"C=": C
```

```
70 DCLOSE#1

READY
RUN
? FILE DATA ERROR IN 50
READY。
```

例10、

```
A=13; 24; 35          B=          C=

20 DOPEN#1,"A9",L13
30 PRINT#1,13; "; "; 24; "; "; 35
40 DCLOSE#1; DOPEN#1,"A9"
50 INPUT#1,A$,B$,C$
60 PRINT"A="; A$,"B="; B$,"C="; C$
70 DCLOSE#1

READY。
```

例11、

```
A=13; 24; 35          B=          C=

20 DOPEN#1,"B6",W
30 PRINT#1,13; "; "; 24; "; "; 35
40 DCLOSE#1; DOPEN#1,"B6"
50 INPUT#1,A$,B$,C$
60 PRINT"A="; A$,"B="; B$,"C="; C$
70 DCLOSE#1

READY。
```

例12、

```
5 A$="123"; B$="456"; C$="789"
10 DOPEN#5,"TASK10",W
20 PRINT#5,A$,B$,C$
30 DCLOSE#5
40 DOPEN#5,"TASK10"
50 INPUT#5,A$,B$,C$
```

```
60 DCLOSE# 1
70 PRINTA$,B$,C$
RUN
123                456                789
```

唯一可靠的强制分隔符是逗号。

例13、

```
A=13              B$=24              C=35

20 DOPEN#1,"A2",L13
30 PRINT#1,13; ", "; 24; ", "; 35
40 DCLOSE#1: DOPEN#1,"A2"
50 INPUT#1,A,B$,C,D$,E
60 PRINT"A="; A,"B$="; B$,"C="; C
70 DCLOSE#1

READY.
```

例14、

```
A=13              B$=24              C=35

20 DOPEN#1, "A1",W
30 PRINT#1, 13; ", "; 24; ", "; 35
40 DCLOSE#1: DOPEN#1, "A1"
50 INPUT#1,A,B$,C,D$, E
60 PRINT"A="; A, "B$=; B$, "C="; C
70 DCLOSE#1

READY.
```

在加强制分隔符字串时，对PRINT#语句中各数据间的分隔符也需注意。该分隔符是逗号还是分号影响到每个数据所占据的存储空间。

顺序文件中，数据占据的空间可从错误地使用空格或分号作强制分隔符时，读出数据的状况来判断。

例15、分隔符为分号，每个数字多占一个字节

```
A=13; 24; 35              B=              C=

20 DOPEN#1, "B2",W
30 PRINT#1,13; "; "; 24; "; "; 35
```

```
40 DCLOSE#1: DOPEN#1, "B2"
50 INPUT#1,A$,B$,C$
60 PRINT"A=": A$,"B=" : B$,"C=": C$
70 DCLOSE#1
80 CMD5: LIST
READY.
```

例16、分隔符为逗号，数字多占11个字节

```
A=13          :          24          :          35

20 DOPEN#1, "B5",W
30 PRINT#1,13,": ",24, ": ",35
40 DCLOSE#1: DOPEN#1,"B5"
50 INPUT#1,A$,B$,C$
60 PRINT"A=": A$
70 DCLOSE#1
80 CMD5: LIST
READY.
```

例17、

```
A=13          24          35

20 DOPEN#1,"B4",W
30 PRINT#1,13, " ", 24, " ",35
40 DCLOSE#1: DOPEN#1,"B4"
50 INPUT#1,A$,B$,C$
60 PRINT"A=": A$
70 DCLOSE#1

READY.
```

例18、分隔符为逗号，每个字符串多占10个字节

```
A$=13          24          35

20 DOPEN#1,"C2",W
30 PRINT#1,"13"," ","24"," ","35"
40 DCLOSE#1: DOPEN#1,"C2"
50 INPUT#1,A$,B$,C$
60 PRINT"A$=": A$
```

```
60 PRINT"A$="; A$
70 DCLOSE#1

READY。
```

对相关文件，可从其记录长度判断每个数据所占空间。

例19、

A=12　　B=34

```
20 DOPEN#1,"C7", L28
30 PRINT#1,12,", ",34
40 DCLOSE#1: DOPEN#1, "C7"
50 INPUT#1,A,B
60 PRINT"A="; A,"B="; B
70 DCLOSE#1

READY。
```

例20、

A=12　　B= 3

```
20 DOPEN#1,"C6",L27
30 PRINT#1,12,", ",34
40 DCLOSE#1: DOPEN#1, "C6"
50 INPUT#1,A,B
60 PRINT"A="; A,"B="; B
70 DCLOSE#1

READY。
```

从例19、20可以看出存两个2位数，记录长度至少为28。

例21、

A=123　　B=34

```
20 DOPEN#1, "C9",L29
30 PRINT#1,123,", ",345
40 DCLOSE#1: DOPEN#1, "C9"
50 INPUT#1,A,B
```

```
60 PRINT"A=";A,"B=";B
70 DCLOSE#1
```

READY。

例22、

A=123　　B=345

```
20 DOPEN#1,"C8",L30
30 PRINT#1,123,",",345
40 DCLOSE#1:DOPEN#1,"C8"
50 INPUT#1,A,B
60 PRINT"A=";A,"B=";B
70 DCLOSE#1
```

READY。

例21、22说明，存两个3位数，记录长度至少为30。

例23、存3个2位数。

A=13　　B=24　　C=35

```
20 DOPEN#1,"A6",L53
30 PRINT#1,13,",",24,",",35
40 DCLOSE#1:DOPEN#1,"A6"
50 INPUT#1,A,B,C
60 PRINT"A=";A,"B=";B,"C=",C
70 DCLOSE#1
```

READY。

例24、

A=13　　B=24　　C=3

```
20 DOPEN#1,"A5",L52
30  PRINT#1,13,",",24,",",35
40 DCLOSE#1:DOPEN#1,"A5"
50 INPUT#1,A,B,C
60 PRINT"A=";A,"B=";B,"C=";C
```

```
70 DCLOSE#1

READY
```

从例19—24可以看出，分隔符为逗号，每个数字多占11个字节。

例25、分隔符为分号，每个数字多占一个字节

A=13　　B$=24　　C=3

```
20 DOPEN#1,"A3",L12
30 PRINT#1,13;",";24;",";35
40 DCLOSE#1: DOPEN#1,"A3"
50 INPUT#1,A,B$,C,D$,E
60 PRINT"A=";A,"B$=";B$,"C=";C
70 DCLOSE#1

READY。
```

例26、分隔符为分号，字符串不多占用存储单元

A$=13　　B$=24　　C$=35

```
20 DOPEN#1,"B1",L8
30 PRINT#1,"13";",";"24";",";"35"
40 DCLOSE#1: DOPEN#1,"B1"
50 INPUT#1,A$,B$,C$
60 PRINT"A$=";A$,"B$=";B$,"C$=";C$
70 DCLOSE#1

READY。
```

综上所述，分隔符为分号时，若写入的是数字，就多占一个字节。字符串不多占空间。分隔符为逗号，则所有数据后面都要比分隔符为分号时多占10个字节。所以，使用逗号比使用分号多浪费很多空间。在这里，逗号和分号的作用与打印机输出时PRINT# 语句中逗、分号作用相似。所以对数据文件使用强制分隔符时，PRINT#语句中各数据间的分隔符拟采用分号为宜。（强制分隔符字串在此处也是数据）。

若需进一步压缩存储，还可将数字数据转换为字符串数据。但这样做，就要增加程序设计和存取数据、特别是存取数据的工作量。

2、APPEND#语句

该语句的功能是向顺序文件末尾添加数据。但在使用中，有时正常，有时又会失去它应

有的功能。

例27、用APPEND#可以向文件末尾添加数据

```
A=111                    B=222

20 DOPEN#1, "D6",W
30 PRINT#1,111
40 DCLOSE#1
50 APPEND#1, "D6"
55 PRINT#1,222
60 DCLOSE#1
70 DOPEN#1,"D6"
80 INPUT#1,A,B
90 DCLOSE#1
100 PRINT"A="; A, "B="; B

READY。
```

例28、有时用APPEND#却将数据"添加"到文件开头

```
5 A$="123" : B$="456" : C$="789"
10 DOPEN#5, "TASK10",W
20 PRINT#5,A$,B$,C$
30 DCLOSE#5 : A$="ABC" : B$="QWE"
40 APPEND#5, "TASK10"
50 PRINT#5,A$,B$
60 DCLOSE#5
70 DOPEN#5, "TASK10"
80 INPUT#5,A$,B$,C$
90 DCLOSE#5
100 PRINTA$,B$,C$
RUN
ABC          QWE          789
```

3、INPUT#语句

有时使用INPUT#语句读出多个变量竟会出现错误，再重复运行，错误又消失。

例29、

```
10 DOPEN#1, "DATA",W
20 PRINT#1,10,20,30,40,50,60
30 DCLOSE#1
40 DOPEN#1, "DATA"
50 INPUT#1,A,B,C,D,E,F
```

```
60 DCLOSE#1
70 PRINT"A="; A,"B="; B,"C="; C
80 PRINT"D="; D, "E="; E,"F="; F
RUN
? FILE DATA ERROR IN 50
```

4、PRINT#语句

该语句有时会无缘无故出现句法错误。

例30、

```
        ⋮
110 PRINT#1,A$
111 PRINT#1,B$
112 PRINT#1,C$
        ⋮
RUN
? SYNTAX ERROR IN 110
```

5、超过顺序文件末尾与超过相关记录末尾读取数据。

i)使用顺序文件时，超过文件末尾读取数据不出现错误，只相当于该语句未执行。

例31、

```
A=13; 24; 35
A=
A=


 20 DOPEN#1,"B2",W
 30 PRINT#1,13; "; "; 24; "; "; 35
 40 DCLOSE#1: DOPEN#1,"B2"
 50 INPUT#1,A$,B$,C$
 60 PRINT"A="; A$
 65 M=M+1
 66 IF M<3 THEN A$=" ": GOTO 50
 70 DCLOSE#1

READY.
```

ii)使用相关文件时，可以超过记录末尾读取数据。

例32、

```
A=12                                C=0

 20 DOPEN#1, "C4",L3
```

```
30 PRINT#1,12
50 RECORD#1,1
60 INPUT#1,A,C
70 PRINT"A=";A,"C=";C

READY
```

（但试验证明，不可超过相关文件末尾读取数据。）

以上两种情况易引起读取数据的混乱。

二、数值计算中的误差及混乱

一般微型机虽不宜作大规模数值计算，但总可胜任一般计算，且精度完全达到原设计要求。

但我们的CBM4040在计算中常发生不应有的误差，甚至发生严重混乱，所以计算量稍大些的题目在CBM4040上运算，不太可靠。

1、不应有的误差（该机至多九位有效数字）。

a、循环变量为实型时，其当前值的计算有误差。

例33.

```
10 FOR I=1.95 TO 19.9 STEP 0.001
20 PRINT I
30 NEXT I
RUN
19.5
⋮
19.77
19.7799999
19.79
19.7999999
19.8099988
19.8199999
⋮
```

b. 有些简单运算中的误差会引起完全错误的结果。

例如，在某课题中，记录号A2、A3通过计算得到，但由于意外的计算误差，致使原来的奇数变为偶数，则导致了记录定位错误，从而取错了数据。（注：记录号为实数则截尾）

例、34

```
H2=21.34                    A2=368.999999
H3=21.67                    A3=435 000001
```

```
H2=21.35          A2=  371
H3=21.43          A3=  387
H2=21.3           A2=  360.999999
H3=21.38          A3=  377.000001
H2=21.3           A2=  369.999999
```

```
10 FOR I=1 TO 5
20 INPUT H2
30 INPUT H3
40 A2=(H2-19.5)*200+1
50 A3=(H3-19.5)*200+1
60 PRINT"H2=";H2,"A2=";A2
70 PRINT"H3=";H3,"A3=";A3
80 NEXT I

READY.
```

所以，在使用中间运算结果（实型）时，要采取必要的措施，避免错误。例如用它作记录号时，先加0.5，截尾时就不会产生上述问题。

2、计算中频繁地产生混乱

在今年4月我们进行了一个运算量较大的水能计算，在不长的时间内发生过四次大错误。

i）4月4日进行方案1的30年数据计算，所计算的第一年第一个月（即1951年6月）第三旬发生错误，而其余1079旬完全正确。程序运行中无间断。

	Q3	H	N
误	3602	1.38	41357
正	59062	2.57	93828

Q3为已知，用其它方案的Q3与之对照发现3602只是该旬第二天的数据，也即，此处循环只进行了一次，就转入下一旬的计算，该旬根本没累加后八天的数据（这旬第一天未参加计算）。这个错误不仅影响了这一旬的小计，而且影响该年及三十年总计。

ii）4月4日运行第一方案，4月7日运行第二方案，其中Q3、Hm数据一样，公式（NRL=9.81*Q3*HM）一样，但结果中NRL不同。

iii）4月10日运行第4方案。第一次运行了一个月就出现"FILE DATA ERROR N 190"，此时，打印输出中Q3、Z3全为0；事实上Q3、Z3是原始数据，根本不是0。而Hm、H又不正常地大。查190行程序为

```
190 INPUT 1#, Z3(L)
```

又查"Z3(L)"文件，数据正确。没对程序进行任何修改又运行，结果正常。

两次运行结果如下：

第一次运行结果：（错误信息在屏幕显示）

⋮

	Q3 (M↑3/s)	Q (M↑3/s)	HM (M)	H (M)	N (KW)	E E04(KWH)	NRL (KW)	H·N (M KW)
9	0	0	257.4	637.09	0	0	0	0
10	0	0	286	707.88	0	0	0	0
10	0	0	257.4	737.09	0	0	0	0
M0.6	0	0	800.8	1982.1	0	0	0	0

? FILE DATA ERROR IN 190

第二次运行结果：

⋮

	Q3 (M↑3/s)	Q (M↑3/s)	HM (M)	H (M)	N (KW)	E EO4(KWH)	NRL (KW)	H·N (MKW)
	27871	0	7.47	0	0	0	179963	0
	32436	0	2.21	0	0	0	66886	0
	59062	0	5.68	0	0	0	338427	0
M0.6	119369	0	12.07	0	0	0	585277	0

⋮

iV）4月15日计算方案7，前三年Q3（原始数据）与原有值不符，并由此引起了结果的错误。后二十七年正确，运行中间无停顿。查“Q3”数据文件也没有错误。

3、程序复盖功能

文本介绍可以用DLOAD、RUN语句实现程序的连接，但在实际应用中除极小的程序（几个语句）外，一般不是运行不下去就是产生错误。

例35、

某课题程序分为三个——“SHI0”、“SHI1”和“SHI2”，程序间用DLOAD、RUN连接，“SHI0”部分程序如下：

⋮

155 DLOAD “SHI1”, DO

160 RUN

程序执行155、160行以后，不是出象“满天星”（散布在萤光屏上的，、〈，〉，0等等），就是只执行“SHI1”的第一条语句：10 ? “M, BC, MC”; MI, BC, MC 此时，屏幕上出现

M, BC, MC　　0　0　0

然后机器死锁，关机后才能重新使用。

例36、

用DLOAD、RUN连接两个排序程序“SORT1”、“SORT2”，第二个程序“SORT2”号虽然运行了，但没运行完就结束了。

CBM 4040的内存只有32K，又不能成功地使用复盖技术，那么在该机上只能搞较小的程序，机器的功能又进一步减弱。

三、直接方式命令的功能极不稳定

该机有几个常用的直接方式命令功能极不稳定，常使工作前功尽弃，这就使微型机的使用简单、灵活、方便的特点在CBM 4040上大为失色。

i) DLOAD有时失去清除内存的功能。

a、若内存现有文件与调入的文件体积总和不超过32K。则两个程序就混在一起

b、若超过32K，出现“OUT OF MEMORY ERROR”。

ii) DLOAD有时装入文件后产生死锁。

例37、

```
DLOAD "SANTA"
SEACHING FOR 0：SANTA
LOADING
READY.
```

然后机器发生死锁（无闪烁光标，也不能打入任何字符），只好关机重来。

另一次调“BOP”程序也是如此。

2、DSAVE命令

i) 有时会无缘无故出现错误，改换一下命令格式，又正常。

例38、

```
DSAVE "VARIFY1"
? DEVICE NOT PRESENT ERROR
DSAVE "VARIFY1", D0,ON 08
（正常执行）
READY.
  （光标闪烁）
```

ii) 有时，在有足够磁盘空间的情况下该命令失效，需反复使用几次才能将文件存入磁盘。

iii) 有时，不仅文件存不进磁盘，机器还发生死锁。只能关掉驱动器才出现“READY.”及闪烁的光标。有时还必须关掉主机，重新启动。

3、COPY命令

一次为使用上的方便将一个262块的数据文件复制到另一个盘上。使用复制后的数据文件时，发现2406和4946两个记录是空的，数据丢失。

有时，规定的文件还未复制完，机器就停止工作。

另外，打入COPY命令后立即出现“READY.”及闪烁的光标，但此时文件的复制工作刚刚开始。用户只能根根驱动器的指示灯判断复制工作是否结束。

4、DI—命令

i)在磁盘目录没混乱的情况下，有时会显示错误的目录。

例39、

```
DI—D0
1 "73    19.6    19.54    1" 55    19
11     "HI"                    REL
20     "MLT—7"                 PRG
21     "NEW  MLT—HO."          PRG
6     "HM—H"                   PG
5     "H—"                     REL
⋮
1534    BLOCKS    FREE
```

ii)如果想在打印机上打印两个驱动器的目录，常常发生错误。

a)若用“OPEN1，4：CMD1：DI—”命令，则只打印0号驱动器的目录。

b)若先打印一个驱动器的目录，打印完毕，再发命令打印第二个驱动器的目录，则第二个驱动器的目录发生混乱。

例40、

```
1     ".52    5.75    4.97" . 28
26     "MLT10  PLAN—1"      PRG
262    "HM(L)               REL
24     "MLT10 PLAN—2"       PRG
19     "MLT10 A YEAR"       PRG
25     "PLAM NO—28.6"       PRG
263    "Z1"                 REL
20     "NEW MLT 1982.5.4"   PRG
1430  BLOCKS FREE
```

四、不明原因的混乱

在4月份运行上述水能计算程序中还发生过几次数据分档的混乱。

i)4月6日，对NRL进行分档排序，数据共10957个，分四档。第一档7000～8000内多出82个第7档1000～2000的数据，其余均未混乱。原始数据文件也未混乱。

ii)4月9日对Q进行分档排序，总量10957个，分四档，在325—350一挡内出现许多0，还有许多不属于该档的数据，其余各档及原始数据文件未混乱。

iii)4月21日，对N分档排序，总量10957个，分四档，其中35000—40000及小于35000两档全部混乱。N最大值为50000以上数据，其余两档及原始数据正常。

类似情况在5月份还发生一次。上述混乱发生后，立即重新处理，又全部正常。

由于该机程序执行过程不可跟踪，又无法打印现场清单，错误常常又是随机的，不可重复，所以有些问题无法弄清原因所在。

五、小结

该机的编辑功能还有许多不便之处。如：按回车键有时不能换行、修改中使用了引号则不能再继续修改该行程序……。此外，还存在有时打印会丢失数据、TAB函数实际为第二个SPC函数等等许多问题，这里不再一一赘述。

由于该机采用了软件固化技术，因此无法针对所存在问题对系统软件进行修改或扩充，所以所存在的问题（除属于硬件的问题外）大多数都无法解决．必须在使用中格外小心。

考机工作及试验总结由刘连芳、张正铀共同完成。