

# BCM—Ⅱ微型机BASIC80的使用

广西计算中心 兰红星 张正铀 刘连芳 陈振坤

## 提 要

本文介绍了在实际应用BASIC—80中所遇到的内存、磁盘、打印机的使用,文件合并以及其它方面的问题,提出了一些处理办法,并就如何提高使用效率谈了点体会。

## 前 言

北京计算机技术研究所生产的BCM—Ⅰ微型电子计算机,使用Z80微处理器作为芯片,配备了CP/M1.4磁盘操作系统,由此支持的语言有FORTRAN、COBOL、BASIC80 CBASIC、ALGOL等等,而BASIC80对于8080和Z80处理器来说是最为广泛应用的一种语言。它具有磁盘文件管理、字符串运算、逻辑运算等功能。另外,北京计算机技术研究所还把BCM—2汉字打印输出系统链接到BASIC—80上去,开扩了它的使用范围。

尽管BCM—Ⅰ是微型机,BASIC—80是解释执行的语言,但经过几个月的使用,我们认为,只要摸清了它的使用规律,尽量扬长避短,那么使用起来不仅比较简单、容易,而且可以提高使用效率。

## 一、内存的使用

因为BASIC—80解释程序占二十多K内存,供用户使用的只有三十多K的内存空间。

1. 目前大部分计算机应用均是进行数据处理,不可避免地要同磁盘打交道,所以就存在内存允许的情况下尽量减少磁盘操作以提高速度的问题。对于数据文件较大,不能一次全读入内存,或者不能确定较适合的读入记录数时,可以定维一个足够大的字符串数组,执行时,一次从磁盘读入一批记录赋给该数组。因为系统是动态管理字符串数组所占据的内存空间(后面将具体介绍),所以,当这个数组的内容把剩余内存都占满后,再给这个数组的下一个元素赋值,就会引起“内存溢出”错误,这时应在程序中安排捕捉错误语句,使程序转到对读入的数据进行处理的程序段。处理完毕,再紧接着读下一批记录,复盖上次所读内容,如此反复,直至全部处理结束。这样,读盘时,磁头一直处于加载状态,减少了磁头动作和磁头的来回移动,也实现了数据的批处理,使得程序的执行速度明显提高。

2. 数组定维后,每个整型数组元素占两个字节,单精度数组元素占四个字节,双精度数组元素占八个字节。所以在使用中,在运算精度允许的情况下,应尽量使用整型数组或单

精度数组，以节省内存。在内存紧张而数组又较大时，这样处理效果比较显著。

对于字符串数组，在定维时，系统给每个元素预分配三个字节。在对字符串数组元素赋值时，再根据字符串的实际长度扩充使用的字节数。所以，如果内存紧张，在使用完字符串元素之后，并且在以后的程序中不再使用它了，则应把这些元素都置空，以减少占用的字节数。

3. 正常的ERASE语句有益于提高内存的使用效率，允许重新支配已经不用的数组所占的内存空间。但该机BASIC—80的ERASE往往会引起混乱，使系统陷入瘫痪状态。

(1) 一个ERASE语句只可以释放一个数组，否则就会出错；

(2) 若定维了几个数组，而后用ERASE释放了一个数组，则其它数组内容要发生混乱，至使程序无法正常运行。

所以ERASE语句实际是不可使用的。

## 二、文件的合并

1. 使用MERGE命令链接两段同类型程序（或均是一般程序，或均是只含DATA语句的程序）

(1) 若两段体积相差不大，则应先将行号小的程序装入内存，再用MERGE命令链接行号大的程序段；

(2) 若两段体积相差悬殊应先装入体积大的程序段，再链连另一个程序。

2. 链接一个只含DATA语句的程序段和一个一般控制程序：

因为后用MERGE命令装入内存的程序必须以ASCII码存储，对于含大量DATA语句的程序段，用ASCII存储易发生混乱（我们在实际应用中碰到多次），所以，为避免混乱，应将只含DATA语句的程序段行号变小，而控制程序用大行号，先装入只含DATA的程序段，再用MERGE链接控制程序段。

在以上两种情况下分别采取的措施不仅可减少不必要的错误，而且可以大大提高链接程序的速度。

## 三、磁盘的使用

1. BCM—Ⅰ的磁盘控制器支持双面单密度磁盘，一个盘面仅有240K左右的可用存贮空间。在磁盘紧张的情况下，节省磁盘空间也比较重要。在内存的程序，可用紧缩二进制码、ASCII码和编码二进制三种格式存入磁盘。但ASCII码格式要占用较多的存贮空间，而且，在使用中我们还发现，用ASCII码格式存贮程序，还可能把原来磁盘上的其它信息破坏。所以，如果不是十分必要，则应避免使用ASCII格式，即使使用，也要保证有足够的磁盘空间。

2. 当在一个程序中要同时处理多个数据文件时，这些文件一定不要分别放在一个盘的不同盘面上，只能放在同一盘面。或者放在另一驱动器的磁盘的任一面上。否则在访问某盘面的文件后，立即访问同一个磁盘的另一盘面上的文件会导致“\* \* 10H \* \*”错误。

3. 在输入或修改程序时，如果程序较大，可能把内存空间占满。这时，不要忙于向磁盘上存这个程序，而应删除某些语句行，减小程序的体积，再发SAVE命令，否则，下次从

磁盘将该程序装入内存，会因为把内存占满后仍未读到文件结束符而产生“内存溢出”错误。此时内存并不接收这个程序。

对这种情况的唯一挽救办法是减小BASIC—80解释程序的体积，例如，在引导BASIC—80时只选择使用一个通道等等。

4. 在打开磁盘文件后，一定要先关闭它才能从驱动器取出文件所在的盘，换入另一个磁盘。否则，会破坏后插入的磁盘上的信息。在调试程序时要特别注意这一点，如果因为程序出错而暂停执行，应该先执行直接命令CLOSE，再进行下一步工作，以免破坏磁盘信息。

#### 四、打印机的使用及屏幕显示格式

1. 打印机要在220V电压下工作。电压低会引起打印缺字现象。

2. 打印机工作时，要避免电磁干扰，否则会出现反向打印。

3. 一般情况下，打印输出为80列/行。可以用CHR\$函数控制每行输出的列数，可控制的打印字体有四种：80列、64列、120列、40列。在原始状态是80列/行的情况下，执行：

```
LPRINT CHR$(28)
```

输出变为132列/行；执行：

```
LPRINT CHR$(14)
```

输出变为40列/行；执行：

```
LPRINT CHR$(75)
```

输出变为64列/行。

4. 同样可以用PRINT CHR\$(X)控制屏幕显示格式：

X = 18 反底显示

X = 19 反底并闪烁

X = 20 闪烁显示

X = 21 字符下加横线

X = 22 字符下加横线并闪烁

X = 23 恢复正常显示

X = 26 清屏幕并恢复正常显示

X = 139 光标下移一行，列位置不变

X = 141 光标返回当前行首

X = 143 光标移到下一行行首

X = 0 反卷走向显示

#### 五、其 它

##### 1. 系统恢复

因为CP/M操作系统从100H开始存放BASIC—80解释程序，所以，如果由于外界干扰或操作失误使系统瘫痪，可以先按[RES]键硬件复位，然后直接用“G100”命令从监控状态进入BASIC—80环境。如果这样做不成功，则重新进行硬件复位，复位后引导CP/M

操作系统，然后再次硬件复位，再使用“G100”命令进入BASIC—80环境。大多数系统瘫痪状态均可用上述方法恢复。

## 2. BASIC—80程序和CBASIC程序的相互转换

如果程序中没有涉及文件处理的语句，BASIC—80程序和该机的CBASIC程序可以互相转换。在CBASIC程序作为BASIC—80程序执行时，只需在每个语句前添加行号，而在将BASIC—80程序作为CBASIC程序处理时，只需将BASIC—80程序以ASCII码形式存入磁盘。

## 3. WIDTH语句

一旦使用了WIDTH语句，就不能有任何读、写磁盘的动作，否则，打印机会随着磁盘操作而发生误动作，严重了，还会导致打印机烧保险。

## 4. 数据混乱时的处理

在数据语句多的情况下常常会由于各种原因发生混乱，常见的有以下两种情况：

### (1) 数据语句重复出现

例、

```

100 DATA 1, 2, 3, ...
200 DATA...
   :
1000 DATA...
1100 DATA...
100 DATA 1, 2, 3, ...
200 DATA...
   :
```

在这种情况下，一般重复的数据行并不真正是独立的DATA语句，是紧挨着的前一个DATA语句的继续。如上例，若显示1100行，则会有：

```

1100 DATA...
100 DATA 1, 2, 3, ...
200 DATA...
```

此时只需要删除1100行，重复的数据就会消失。

### (2) 数据语句中间出现多余数据

例、

```

100 DATA...
200 DATA...
   :
1100 DATA...
38 DATA...
38
38
   :
38
   :
```

在这种情况下，需要紧接混乱数据之后加上一程序行，比如2000DATA2000，然后将1100—2000全部删除，一般可将其中的混乱数据删除，问题得以解决。

## 结 束 语

以上仅是我们在实际使用中的一些体会，还缺乏从系统角度对它们的分析。我们只是希望起到抛砖引玉的作用，使我们今后能够更有效、更巧妙地使用该系统，充分发挥现有机器在电算服务中的作用。