

PC-DOS 系统汉化和 CC-DOS 分析与使用

李玉林 钱培德

测绘出版社

前 言

我国目前拥有微机30多万台，销售量还在逐年增加，而PC系列微机（及其兼容机）作为我国流行机种，占领着绝大部分市场，它所配的操作系统PC-DOS由于适应软件较多，而成为单用户操作系统典范。汉化后的CC-DOS，在PC-DOS原有的基础上增加了汉字功能，更适合我国国情，因此颇受各界用户欢迎。

本书首先对PC-DOS作了简单分析，从而找出了汉化的关键所在，接着对汉化后的CC-DOS进行了剖析。书中全部文本均用DEBUG工具打出，并附有详细注解，书后还附有DOS 2.0系统的软中断功能与INT 21H.系统调用，有较完整的功能概要，便于读者作进一步的分析。该书对微机的系统开发、软件编程及深入了解和应用，特别有参考价值。另外，本书还为初学者准备了二章内容，即第二章和第九章，它讲了一些最基础的上机知识、最基本的操作方法和最常用的命令，是初学者最好的助手。

本书原为培训教材，经几年试用，得到学员们热情支持，为了满足更多读者的需要，经全面修订和补充后，交给测绘出版社出版，并向全国发行。本书原教材第三至八章文字选用钱培德编资料，略有修改，第二、九章承张志豪同志编写。在本书编写过程中还得到华彬文总编的大力协助，并提宝贵意见，在此表示衷心的感谢。

编者

1988. 5

目 录

第一章 PC-DOS 2.0系统分析	(1)
1.1 概论.....	(1)
1.2 PC-DOS 的组成与功能.....	(1)
1.3 PC-DOS 的命令处理实例分析.....	(2)
第二章 PC-DOS操作说明	(29)
2.1 系统启动.....	(29)
2.2 软盘的初始化和复制.....	(30)
2.3 键盘命令.....	(31)
2.4 文件及文件操作命令.....	(32)
第三章 CC-DOS的概述和初步分析	(35)
3.1 系统概述.....	(35)
3.2 CC-DOS的自举原理.....	(35)
3.3 CC-DOS中汉字显示输出的实现.....	(36)
3.4 CC-DOS中汉字输入的实现.....	(38)
3.5 CC-DOS中汉字打印输出的实现.....	(40)
第四章 CC-BIOS的引导过程	(44)
4.1 概 述.....	(44)
4.2 BIOS的结构与修改.....	(44)
4.3 CC-DOS的自举过程.....	(44)
4.4 CC-BIOS的引导过程.....	(45)
4.5 对FILE1.EXE的分析.....	(46)
4.6 对CCCC.EXE的分析.....	(47)
4.7 CC-BIOS打印模块的形成.....	(48)
4.8 引导源程序.....	(49)
第五章 CC-BIOS的键盘管理模块	(57)
5.1 引 言.....	(57)
5.2 键盘中断的处理.....	(57)
5.3 键盘输入请求程序的结构.....	(57)
5.4 代码转换和扫描表.....	(59)
5.5 汉字输入功能的实现.....	(60)
5.6 读入字符处理模块.....	(61)
5.7 输入码处理模块.....	(62)
5.8 键盘管理模块的源程序.....	(65)

第六章	CC - BIOS的显示器控制模块	(112)
6.1	总 述.....	(112)
6.2	显示器控制模块的结构.....	(112)
6.3	几个重要的工作区.....	(113)
6.4	CRT的初始化.....	(114)
6.5	光标的产生和定位.....	(115)
6.6	屏幕滚动的实现.....	(118)
6.7	读字符的实现.....	(120)
6.8	字符显示的实现.....	(120)
6.9	TTY显示方式的实现.....	(120)
6.10	提示行管理模块.....	(122)
6.11	显示器控制模块的源程序.....	(124)
第七章	CC - BIOS的打印驱动模块	(168)
7.1	总 述.....	(168)
7.2	打印屏幕驱动程序.....	(168)
7.3	打印机驱动程序的组成和输出字型.....	(169)
7.4	打印机驱动程序的工作区.....	(169)
7.5	打印机驱动程序的总体.....	(170)
7.6	17类中断程序的 0 号功能块.....	(172)
7.7	图形态字符处理模块.....	(175)
7.8	打印驱动模块的源程序.....	(177)
第八章	CC - BIOS的高级打印驱动模块	(216)
8.1	概 述.....	(216)
8.2	高级打印驱动模块的生成.....	(216)
8.3	21类中断程序.....	(217)
8.4	17类中断程序.....	(217)
8.5	17类中断程序的 0 号功能块.....	(219)
8.6	非ESC态处理模块.....	(221)
8.7	机内码处理模块.....	(222)
8.8	高级打印驱动模块的源程序.....	(225)
第九章	CC - DOS使用说明	(263)
9.1	系统启动.....	(263)
9.2	汉字输入.....	(263)
9.3	汉字打印输出.....	(265)
9.4	在CC - DOS支持下的软件使用.....	(267)
附录 1	PC - DOS软中断程序简介.....	(272)
附录 2	PC - DOS INT 21H系统调用简介.....	(283)
附录 3	CC - DOS其它版本使用简介.....	(293)

第一章 PC-DOS 系统分析

1.1 概论

PC-DOS已成为16位机单用户操作系统的范例。由于它分散模块结构、配置灵活、资料公开,深受用户欢迎。

在1980年后的短短几年里,先由Seattle Computer Products公司开发出来,microsoft公司取得其专利后改名为MS-DOS,IBM公司决定采用MS-DOS作PC机操作系统,又命名为PC-DOS。它有较强的文件处理功能,拥有众多的语言软件,而且它本身具有丰富的资源,(如外部和内部命令及许多系统功能调用)。

1.2 PC-DOS的组成与功能

PC-DOS的组成共有4个主要部分:

1. 引导块——由FORMAT命令装在盘上0道1扇区,它能自动装入DOS系统。
2. IBMBIO.COM——是IBM PC的基本输入输出处理程序(隐文件)。
3. IBMDOS.COM——主要提供用户文件管理功能和INT 2/H系统调用功能(隐文件)。
4. COMMAND.COM——命令处理程序,接受键盘命令,执行相应的命令子程序。

各部分功能如下所述。

1. 引导块功能——系统加电或复位时,执行POM起始地址FFFF:00的启动程序,首先对系统进行初始化和自检,然后进入ROMBIOS的Boot-strap(INT 19H),检查驱动器有否插盘若有则读入DOS引导块,否则进入ROMBIOS。

读入引导块放在起始地址0:7C00处,并开始执行它。引导过程如下:

- (1) 查看目录块(0道,5扇区),有否BIO.COM和DOS.COM二个文件;
- (2) 若有,则将此二文件都读到60:0000处;
- (3) 转到60:0000执行BIO.COM;
- (4) 若没有此二文件,则非系统盘,即显示提示信息等待换盘。

2. IBMBIO.COM功能——此文件有以下三个主要部分:

- (1) 初始化程序
 - a. 建立磁盘参数表(INT 1EH)入口;
 - b. 初始化RS232口和打印机口;
 - c. 建立1,3,4和1B中断入口;
 - d. 建立50段中标志单元(DOS通讯区);
 - e. 将IBMDOS从当前位置前移到BF段,复盖部分BIOS程序。
 - f. 确定软盘驱动器数和RAM大小;
 - g. 调IBMDOS初始化程序,建立用户区段;

- h. 填25, 26级中断入口;
- i. 将 COMMAND.COM读到用户区100以后处;
- j. 建立磁盘缓冲 (DTA) 地址 (偏移为80H), 执行 COMMAND.COM

(2) BIOS 接口功能共有11个子程序

- a. 打屏幕处理程序 (INT 5H)
- b. 日时钟中断程序 (INT 8H)
- c. 读, 改日时钟程序 (INT 1AH)
- d. 显示器接口程序 (INT 10H)
- e. 系统设备检查程序 (INT 11H)
- f. 内存容量检查程序 (INT 12H)
- g. 磁盘读写程序 (INT 13H)
- h. RS232 接口程序 (INT 14H)
- i. 键盘读判程序 (INT 16H)
- j. 打印机接口程序 (INT 17H)
- k. 启动自举 BOOT STRAP (INT 19H)

(3) 其他服务子程序——各软件中断子程序用法及子功能说明请参看附录1。

3. IBMDOS.COM功能——此文件有以下二个主要部分:

(1) 初始化

- a. 依照 IBM B I O . C O M 固有 D P T , 再建磁盘参数表, 每驱动器备两张表, 每表14H 字节 (双面盘一表, 单面盘一表);
- b. 各驱动器建一个文件分配表 (F A T);
- c. 在最后一张 F A T 末地址后建立用户区段址;
- d. 检查 R A M 大小指标的正确性;
- e. 填中断0, 20—26及30的中断入口地址;
- f. 在程序段建立段前缀控制块 (P S P)。

(2) I N T 21 系统功能调用——它是 D O S 系统的核心, 许多 D O S 命令都用到这些功能, 资源可供用户共享。其用法及子功能参数说明请查附录2。

4. COMMAND.COM功能将在下一节中详细介绍。

1.3 COMMAND.COM命令处理程序

在 IBMDOS.COM 所建立段前缀 (P S P) 之后, 由 IBM B I O 初始化程序将 COMMAND.COM 的常驻部分调入 CS: 100 开始位置, 由该地址开始执行 COMMAND.COM 的初始化程序, 再调 COMMAND.COM 的复盖部分。复盖部分是命令处理程序本身, 它包括内部命令处理程序、批命令处理程序及装入和执行外部命令的例行程序。

常驻部分主要完成三方面工作:

- 1. 建立并管理程序中止 (INT 22H)
 - 中断退出 (INT 23H)
 - 差错处理 (INT 24H)
 - 结束, 保留程序 (INT 27H)

(前三个建立时要冲去IBMDOS初始化所建入口)

2. 与IBMDOS用户区程序、COMMAND.COM的暂存部分通讯。

3. 当用户程序终止后还要检查暂存部分是否被用户程序复盖，若是则重新调入。

DOS 2.0在内存0:0处都有中断向量表，此表是INT软中断程序转子程序的地址表，如西文盘的(因机种与DOS版本的不同而不同)KH机数据如下:

INT 10H 转子地址为F000:E6FD

INT 13H 转子地址为C800:0328

INT 16H 转子地址为F000:EFEC

INT 17H 转子地址为F000:EE8A

INT 19H 转子地址为C800:01AF

西文盘上这些软中断子程序都固化在系统ROM中，要汉化西文盘为CC-DOS，必须将有关子程序修改为另一套汉字处理子程序。所以对比西文DOS与CC-DOS，其中某些中断向量必须有所改动。第四章将讨论此问题。

西文DOS 2.0中断向量INT (KH兼容机)

```

A>debug
-r
AX=0000 BX=0000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=0A86 ES=0A86 SS=0A86 CS=0A86 IP=0100 NV UP DI PL NZ NA PD NC
0A86:0100 0000          ADD     [BX+SI],AL          DS:0000=CD

-d0:0
0000:0000 72 30 EB 00 47 01 70 00-97 E1 00 F0 47 01 70 00   r0k.G.p..a.p6.p.
0000:0010 47 01 70 00 33 F7 00 F0-00 00 00 00 00 00 00 00   G.p.3w.p.....
0000:0020 6E FE 00 F0 38 F0 00 F0-00 00 00 00 00 00 00 00   n*.p8p.p.....
0000:0030 00 00 00 00 4D 08 00 CB-BA FF 00 F0 47 01 70 00   ....M..H:..p6.p.
0000:0040 FD E6 00 F0 7E FF 00 F0-73 FF 00 F0 2B 03 00 CB   }f.p*..ps..p(..H
0000:0050 19 EF 00 F0 1C FF 00 F0-EC EF 00 F0 8A EE 00 F0   .o.p:..plo.p.n.p
0000:0060 1C FF 00 F0 AF 01 00 CB-89 FF 00 F0 F2 00 18 05   ...p/..H...pr...
0000:0070 1C FF 00 F0 37 E7 00 F0-22 05 00 00 00 00 00 00   ...p7q.p*.....

-d
0000:0060 07 08 EB 00 80 01 C9 06-42 02 95 07 70 02 95 07   ..k...I.B...p...
0000:0090 E2 04 C9 06 E0 13 EB 00-2E 14 EB 00 13 27 EB 00   b.I.'k...k..'k.
0000:00A0 13 08 EB 00 E5 02 18 05-00 00 00 00 00 00 00 00   ..k.e.....
0000:00B0 00 00 00 00 00 00 00 00-6D 03 C9 06 00 00 00 00   .....a.I.....
0000:00C0 EA 14 08 EB 00 00 00 00-00 00 00 00 00 00 00 00   j..k.....
0000:00D0 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   .....
0000:00E0 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   .....
0000:00F0 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00   .....

```

带硬盘的KH机还有: INT 40H 入口F 0 0 0: F401

INT 41H 入口 C800 : 02D1

```

0000:0100 01 F4 00 F0 1D 02 00 CB-00 00 00 00 00 00 00 00 .t.p...H.....
0000:0110 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
0000:0120 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
0000:0130 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
0000:0140 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
0000:0150 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
0000:0160 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
0000:0170 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....

```

CC-DOS向量表变动部分有:

INT 10H 404 A : 1848 INT 16H 404 A : 98B 3

INT 17TH 4 B 03: 1A63 INT 1DH 404 A : 1888

INT 1FH 404A : 2798 INT 05H 4 B 03: 201C

AX=0000 BX=0000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000

DS=4DE4 ES=4DE4 SS=4DE4 CS=4DE4 IP=0100 NV UP DI PL NZ NA PD NC

4DE4:0100 0000 ADD [BX+SI],AL DS:0000=CD

-d9:0

CC-DOS 2.0 / 2.1 向量表

```

0000:0000 43 31 E3 00 3F 01 70 00-97 E1 00 F0 3F 01 70 00 C1c.?.p..a.p?.p.
0000:0010 3F 01 70 00 33 F7 00 F0-00 00 00 00 00 00 00 00 ?.p.3w.p.....
0000:0020 6E FE 00 F0 3B F0 00 F0-00 00 00 00 00 00 00 00 n?.pBp.p.....
0000:0030 00 00 00 00 4D 08 00 CB-BA FF 00 F0 3F 01 70 00 ...H..H:..p?.p.
0000:0040 4B 1B 4A 40 7E FF 00 F0-73 FF 00 F0 2B 03 00 CB H.Je?.ps..p(..H
0000:0050 19 EF 00 F0 1C FF 00 F0-B3 9B 4A 40 8A EE 00 F0 .a.p...p3.Je.n.p
0000:0060 1C FF 00 F0 AF 01 00 CB-89 FF 00 F0 3B 01 70 00 ...p/..H...pB.p.
0000:0070 1C FF 00 F0 8B 1B 4A 40-22 05 00 00 9B 27 4A 40 ...p..Je?....'Je

```

-d

```

0000:0080 FB 0B E3 00 80 01 42 05-42 02 F3 4A 70 02 F3 4A (.c...B.B.s)p.sJ
0000:0090 E2 04 42 05 D4 14 E3 00-21 15 E3 00 E7 27 E3 00 b.B.T.c!.c.g'c.
0000:00A0 07 0C E3 00 26 01 70 00-00 00 00 00 00 00 00 00 ..c.&p.....
0000:00B0 00 00 00 00 00 00 00 00-6D 03 42 05 00 00 00 00 .....n.B.....
0000:00C0 EA 0B 0C E3 00 00 00 00-00 00 00 00 00 00 00 00 j..c.....
0000:00D0 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
0000:00E0 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
0000:00F0 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....

```

-d

```

0000:0100 01 F4 00 F0 1D 02 00 CB-00 00 00 00 00 00 00 00 .t.p...H.....
0000:0110 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
0000:0120 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
0000:0130 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
0000:0140 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
0000:0150 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
0000:0160 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
0000:0170 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....

```

-q

B:\debug

CC-DOS2.4 向量表

-r

AX=0000 BX=0000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=4EBE ES=4EBE SS=4EBE CS=4EBE IP=0100 NV UP DI PL NZ NA PO NC
4EBE:0100 0000 ADD [BX+SI],AL DS:0000=CD

-d0:0

0000:0000 72 30 EB 00 47 01 70 00-97 E1 00 F0 47 01 70 00 rOk.G.p..a.pG.p.
0000:0010 47 01 70 00 33 F7 00 F0-00 00 00 00 00 00 00 00 G.p.3w.p.....
0000:0020 6E FE 00 F0 3B F0 00 F0-00 00 00 00 00 00 00 00 n*.p8p.p.....
0000:0030 00 00 00 00 4D 0B 00 CB-BA FF 00 F0 47 01 70 00M..H:..pG.p.
0000:0040 4B 1B 4A 40 7E FF 00 F0-73 FF 00 F0 28 03 00 CB H.Je*..ps..p(..H
0000:0050 19 EF 00 F0 1C FF 00 F0-83 99 4A 40 8A EE 00 F0 .o.p...p..Je.n.p
0000:0060 1C FF 00 F0 AF 01 00 CB-89 FF 00 F0 40 01 70 00p/..H...pE.p.
0000:0070 1C FF 00 F0 8B 1B 4A 40-22 05 00 00 98 27 4A 40p..Je*....'Je

-d

0000:0080 07 0B EB 00 80 01 42 05-42 02 9D 4B 70 02 9D 4B .k...B.B..Kp..K
0000:0090 E2 04 42 05 E0 13 EB 00-2E 14 EB 00 13 27 EB 00 b,B..'k...k..'k.
0000:00A0 13 0B EB 00 2E 01 70 00-C6 AE 4A 40 00 00 00 00 .k...p.F.Je....
0000:00B0 00 00 00 00 00 00 00 00-6D 03 42 05 00 00 00 00m.B.....
0000:00C0 EA 14 0B EB 00 00 00 00 00-00 00 00 00 00 00 00 00 jJ.k.....
0000:00D0 00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
0000:00E0 00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
0000:00F0 00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00

-d

0000:0100 01 F4 00 F0 1D 02 00 CB-00 00 00 00 00 00 00 00 .t.p...H.....
0000:0110 00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
0000:0120 00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
0000:0130 00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
0000:0140 00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
0000:0150 00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
0000:0160 00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
0000:0170 00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00

-d40:00

0040:0000 FB 03 00 00 00 00 00 00-7B 03 00 00 00 00 00 00 x.....x.....
0040:0010 6D 42 01 00 02 C0 01 00-00 00 36 00 36 00 0D 1C mB...e....6.6...
0040:0020 64 20 0D 1C 64 20 0D 1C-64 20 34 05 30 0B 3A 27 d .d .d 4.0.: '
0040:0030 30 0B 30 0B 0D 1C 64 20-30 0B 3A 27 30 0B 03 00 0.0...d 0.: '0...
0040:0040 97 00 05 00 00 11 01 0B-02 06 50 00 00 10 00 00P.....
0040:0050 0B 17 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 =.....
0040:0060 67 00 00 D4 03 1E 20 00-00 00 00 00 1A 0C 00 00 g..T.
0040:0070 00 00 AA 55 00 01 B0 00-14 14 14 14 01 01 01 01 ..*U.....

-q

A>b: debug

CC-DOS3.0 向量表

-r

AX=0000 BX=0000 CX=0000 DX=0000 SP=CFDE BP=0000 SI=0000 DI=0000
DS=60C9 ES=60C9 SS=60C9 CS=60C9 IP=0100 NV UP DI PL NZ NA PO NC
60C9:0100-0000 ADD [BX+SI],AL DS:0000=CD

-d0:0

```

0000:0000 5A 50 27 01 ED 01 70 00-97 E1 00 F0 ED 01 70 00 ZP'.m.p..a.p.p.p.
0000:0010 ED 01 70 00 33 F7 00 F0-00 00 00 00 00 00 00 m.p.3w.p.....
0000:0020 6E FE 00 F0 38 F0 00 F0-00 00 00 00 00 00 00 n^p8p.p.....
0000:0030 00 00 00 00 4D 08 00 CB-BA FF 00 F0 ED 01 70 00 ....M..H:..p.p.
0000:0040 47 18 2A 53 7E FF 00 F0-73 FF 00 F0 54 09 70 00 B.*S^..ps..pT.p.
0000:0050 19 EF 00 F0 1C FF 00 F0-15 99 2A 53 BA EE 00 F0 .o.p...p..*S.n.p
0000:0060 1C FF 00 F0 75 0A 70 00-89 FF 00 F0 E7 01 70 00 ...pu.p...pg.p.
0000:0070 1C FF 00 F0 87 18 2A 53-22 05 00 00 96 27 2A 53 ...p..*S^....'S
-d
0000:0080 5B 14 27 01 7A 14 27 01-42 02 D8 5D 70 02 D8 5D [.'z.'B.X]p.X]
0000:0090 81 04 40 08 88 15 27 01-FB 15 27 01 E4 5C 27 01 ..e.B.'(.d\'.
0000:00A0 61 14 27 01 4E 01 70 00-61 14 27 01 61 14 27 01 a.'N.p.a.'a.'
0000:00B0 61 14 27 01 61 14 27 01-F6 01 40 08 EC 16 27 01 a.'a.'v.e.l.'.
0000:00C0 EA 62 14 27 01 14 27 01-61 14 27 01 61 14 27 01 jb...'a.'a.'.
0000:00D0 61 14 27 01 61 14 27 01-61 14 27 01 61 14 27 01 a.'a.'a.'a.'.
0000:00E0 61 14 27 01 61 14 27 01-61 14 27 01 61 14 27 01 a.'a.'a.'a.'.
0000:00F0 61 14 27 01 61 14 27 01-61 14 27 01 61 14 27 01 a.'a.'a.'a.'.
-d
0000:0100 01 F4 00 F0 1D 02 00 CB-00 00 00 00 00 00 00 00 .t.p...H.....
0000:0110 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
0000:0120 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
0000:0130 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
0000:0140 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
0000:0150 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
0000:0160 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00- .....
0000:0170 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
-d40:0
0040:0000 FB 03 00 00 00 00 00 00-78 03 00 00 00 00 00 00 x.....x.....
0040:0010 6D 42 01 00 02 C0 01 00-00 00 3A 00 3A 00 30 08 #B...e...:..0.
0040:0020 3A 27 30 0B 0D 1C 64 20-0D 1C 64 20 0D 1C 64 20 :^0...d .d .d
0040:0030 34 05 30 0B 3A 27 30 0B-0D 1C 0D 1C 64 20 03 00 4.0.:^0....d ..
0040:0040 5C 00 05 00 00 11 01 0B-02 06 50 00 00 10 00 00 K.....P.....
0040:0050 0B 17 00 00 00 00 00 00-00 00 00 00 00 00 00 =.....
0040:0060 67 00 00 D4 03 1E 20 00-00 00 00 00 9A 0B 00 00 g..T.. .....
0040:0070 00 00 AA 55 00 01 80 00-14 14 14 01 01 01 01 ..*U.....

```

内存中断向量表由于汉字盘版本不同会有差异，对照向量表的不同就可以知道PC-DOS汉化主要工作是在修改向量表、设计相应汉字处理程序。

调汉字库后，INT中断向量表随机种与汉字盘版本的不同而变动。

```

-d0:0
16行屏幕显示的ECDOS2.1向量表
0000:0000 43 31 E3 00 D2 EF 00 F0-97 E1 00 F0 00 F0 70 00 C1c.Ro.p.a.p.pp.
0000:0010 3F 01 70 00 33 F7 00 F0-00 00 00 00 00 00 00 ?p.3w.p.....
0000:0020 6E FE 00 F0 38 F0 00 F0-00 00 00 00 00 00 00 n^p8p.p.....
0000:0030 00 00 00 00 4D 08 00 CB-BA FF 00 F0 3F 01 70 00 ....M..H:..p?p.p.
0000:0040 48 18 22 06 7E FF 00 F0-73 FF 00 F0 2B 03 00 CB H."^..ps..p(..H
0000:0050 19 EF 00 F0 1C FF 00 F0-83 98 22 06 BA EE 00 F0 .o.p...p3..n.p
0000:0060 1C FF 00 F0 AF 01 00 CB-89 FF 00 F0 38 01 70 00 ...p/..H...p8.p.
0000:0070 1C FF 00 F0 88 18 22 06-22 05 00 00 98 27 22 06 ...p..". "...".

```

```

-3
0000:0080 FB 0B E3 00 80 01 42 05-42 02 D4 59 70 02 D4 59 (.c...B.B.Typ.TY
0000:0090 E2 04 42 05 04 14 E3 00-21 15 E3 00 E7 27 E3 00 b.B.T.c.!c.g'c.
0000:00A0 07 0C E3 00 26 01 70 00-00 00 00 00 00 00 00 ..c.&p.....
0000:00B0 00 00 00 00 00 00 00-6D 03 42 05 00 00 00 00 .....n.B.....
0000:00C0 EA 08 0C E3 00 00 00 00-00 00 00 00 00 00 00 j..c.....
0000:00D0 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
0000:00E0 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
0000:00F0 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
-d
0000:0100 01 F4 00 F0 1D 02 00 CB-00 00 00 00 00 00 00 00 .t.p...H.....
0000:0110 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
0000:0120 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
0000:0130 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
0000:0140 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
0000:0150 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
0000:0160 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
0000:0170 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
-d40:0
0040:0000 FB 03 00 00 00 00 00 00-78 03 00 00 00 00 00 00 x.....x.....
0040:0010 6D 42 01 00 02 C0 01 00-00 00 38 00 38 00 3A 27 mB...e...B.B.:
0040:0020 30 08 0D 1C 64 20 0D 1C-64 20 0D 1C 64 20 34 05 0...d .d .d 4.
0040:0030 30 08 3A 27 30 0B 0D 1C-0D 1C 64 20 30 0B 03 00 0.: '0.....d 0...
0040:0040 FC 00 05 00 00 11 01 0B-02 06 50 00 00 10 00 00 l.....P.....
0040:0050 0B 17 00 00 00 00 00 00-00 00 00 00 00 00 00 00 =.....

```

DOS内部命令表在系统启动并进入A >标志符后，放在地址0：CAF0附近，处理某些命令之后，该表会移动到内存高端（640K RAM在9000段），命令表中每个命令后有该命令子程序入口地址。如：

COPY入口地址为CS：1 ACC，1 ACC→DX，然后调用DX地址的COPY命令子程序来执行。其它命令亦照此办理。内部命令表后接着是文件处理区（FCB）。

下面例举COPY，DIR二个命令，说明Command.COM中命令处理过程。

DOS 2.0 命令处理分析

```
A>debug command.com
```

```
-r
```

```
AX=0000 BX=0000 CX=4500 DX=0000 SP=FFFE BP=0000 SI=0000 DI=0000
DS=0913 ES=0913 SS=0913 CS=0913 IP=0100 NV UP DI PL NZ NA PO NC
0913:0100 E97D0B JMP 0C80
```

```
-t
```

```
AX=0000 BX=0000 CX=4500 BX=0000 SP=FFFE BP=0000 SI=0000 DI=0000
DS=0913 ES=0913 SS=0913 CS=0913 IP=0C80 NV UP DI PL NZ NA PO NC
0913:0C80 BC001 MOV SP,0180
```

```

-u
0913:0C80 8C8001      MOV     SP,0180
0913:0C83 8CC8      MOV     AX,CS
0913:0C85 05C800     ADD     AX,00C8
0913:0C88 A3BD0B      MOV     [08BD],AX
0913:0C8B A10200     MOV     AX,[0002]
0913:0C8E 8C1EC30B   MOV     [08C3],DS
0913:0C92 8C1EC70B   MOV     [08C7],DS
0913:0C96 8C1EBB0B   MOV     [08BB],DS
0913:0C9A A36709     MOV     [0967],AX
0913:0C9D 2D5003     SUB     AX,0350

```

```

-u
0913:0CA0 A36309     MOV     [0963],AX
0913:0CA3 A12C00     MOV     AX,[002C]
0913:0CA6 0BC0      OR      AX,AX
0913:0CAB 7406      JZ      0C80
0913:0CAA FE068A10   INC     BYTE PTR [108A]
0913:0CAE EB0B      JMP     0CBB
0913:0CB0 BBE00B      MOV     AX,0BE0
0913:0CB3 B104      MOV     CL,04
0913:0CB5 D3E8      SHR     AX,CL
0913:0CB7 BCDA      MOV     DX,DS
0913:0CB9 03C2     ADD     AX,DX
0913:0CBB A31C0B     MOV     [081C],AX
0913:0CBE B80037     MOV     AX,3700

```

```

-u
0913:0CC1 CD21      INT     21
0913:0CC3 8816BF0B   MOV     [08BF],DL
0913:0CC7 80FA2F   CMP     DL,2F
0913:0CCA 750B      JNZ     0CD7
0913:0CCC C60667105C   MOV     BYTE PTR [1067],5C
0913:0CD1 90          NOP
0913:0CD2 C606EE0B5C   MOV     BYTE PTR [0BEE],5C
0913:0CD7 06          PUSH   ES
0913:0CD8 B82135     MOV     AX,3521
0913:0CDB CD21      INT     21
0913:0CDD 891EAB0B   MOV     [0BAB],BX

```

-g=cs:c80,460

执行此段后，CRT显示DOS2.0 标牌

The IBM Personal Computer DOS
Version 2.00 (C)Copyright IBM Corp 1981, 1982, 1983

```

AX=8000 BX=0014 CX=0000 DX=8B2F SP=0180 BP=0000 SI=08C9 DI=2CAE
DS=0913 ES=7CB0 SS=0913 CS=0913 IP=0460 NV UP DI PL ZR NA PE NC
0913:0460 FF2E6109     JMP     FAR [0961]                DS:0961=012C

```

-t

```

AX=8000 BX=0014 CX=0000 DX=8B2F SP=0180 BP=0000 SI=0BC9 DI=2CAE
DS=0913 ES=7CB0 SS=0913 CS=7CB0 IP=012C NV UP DI PL ZR NA PE NC
7CB0:012C FC          CLD
-u
7CB0:012C FC          CLD
7CB0:012D BCCB       MOV     AX,CS
7CB0:012F 8ED0       MOV     SS,AX
7CB0:0131 BC632F     MOV     SP,2F63
7CB0:0134 BEC0       MOV     ES,AX
7CB0:0136 36         SS:
7CB0:0137 BE1EA02C   MOV     DS,[2CA0]
7CB0:0138 FB         STI
7CB0:013C 36         SS:
7CB0:013D C606062B80 MOV     BYTE PTR [2B06],80
7CB0:0142 36         SS:
7CB0:0143 C606892B80 MOV     BYTE PTR [2B89],80
7CB0:0148 0BED       OR     BP,BP
7CB0:014A 7409       JZ     0155
-u
7CB0:014C 36         SS:
7CB0:014D C706072B010D MOV     WORD PTR [2B07],0D01
7CB0:0153 EB12       JMP     0167
7CB0:0155 B03E710900   CMP     BYTE PTR [0971],00
7CB0:015A 740B       JZ     0167
7CB0:015C 1E         PUSH   DS
7CB0:015D 0E         PUSH   CS
7CB0:015E 1F         POP    DS
7CB0:015F BA0C2C     MOV     DX,2C0C
7CB0:0162 B43B       MOV     AH,3B
7CB0:0164 CD21       INT     21
7CB0:0166 1F         POP    DS
7CB0:0167 B03E780A00   CMP     BYTE PTR [0A7B],00
-u
7CB0:016C 740A       JZ     0178
7CB0:016E B03E770A00   CMP     BYTE PTR [0A77],00
7CB0:0173 7503       JNZ    0178
7CB0:0175 EBFA05     CALL   0772
7CB0:0178 C6066B0900   MOV     BYTE PTR [096B],00
7CB0:017D C606710900   MOV     BYTE PTR [0971],00
7CB0:0182 BCC8       MOV     AX,CS
7CB0:0184 8ED8       MOV     DS,AX
7CB0:0186 2B06A22C   SUB     AX,[2CA2]
7CB0:018A BA1000     MOV     DX,0010
-g=cs:12c,456

```

此时跳出A >提示符，可打入DOS命令如下：

```
A>copy dr.dac dd.txt
```

AX=09FA BX=290D CX=0003 DX=2B00 SP=2F63 BP=0000 SI=2A10 DI=2CDA
DS=7CB0 ES=7CB0 SS=7CB0 CS=7CB0 IP=0456 NV UP EI NG NZ AC PE CY
7CB0:0456 75EC JNZ 0444
-t

AX=09FA BX=290D CX=0003 DX=2B00 SP=2F63 BP=0000 SI=2A10 DI=2CDA
DS=7CB0 ES=7CB0 SS=7CB0 CS=7CB0 IP=0444 NV UP EI NG NZ AC PE CY
7CB0:0444 BFD92C MOV DI,2CD9

-u
7CB0:0444 BFD92C MOV DI,2CD9
7CB0:0447 8A0C MOV CL,(SI)
7CB0:0449 E332 JCXZ 047D
7CB0:044B F3 REPZ
7CB0:044C A6 CMPSB
7CB0:044D 9F LAHF
7CB0:044E 03F1 ADD SI,CX
7CB0:0450 9E SAHF
7CB0:0451 AC LODSB
7CB0:0452 A2AE2C MOV [2CAE],AL
7CB0:0455 AD LODSW
7CB0:0456 75EC JNZ 0444
7CB0:0458 8BD0 MOV DX,AX
7CB0:045A 803EAE2C00 CMP BYTE PTR [2CAE],00
7CB0:045F 740B JZ 046C
7CB0:0461 A0B12C MOV AL,[2CB1]

-u
7CB0:0464 0A06B22C OR AL,[2CB2]
7CB0:0468 3CFF CMP AL,FF
7CB0:046A 74CD JZ 0439
7CB0:046C EB7315 CALL 19E2
7CB0:046F FFD2 CALL DX ; DX指向COPY子程序入口
7CB0:0471 E990FC JMP 0104
7CB0:0474 E989FC JMP 0100
7CB0:0477 FECA DEC DL
7CB0:0479 FECB DEC AL
7CB0:047B 74F7 JZ 0474
7CB0:047D C606AF2C00 MOV BYTE PTR [2CAF],00
7CB0:0482 8A16BC2C MOV DL,[2CBC]

-g=cs:444,46f

AX=1A00 BX=290D CX=0000 DX=1ACC SP=2F63 BP=0000 SI=2A48 DI=2CDE
DS=7CB0 ES=7CB0 SS=7CB0 CS=7CB0 IP=046F NV UP EI PL ZR NA PE NC
7CB0:046F FFD2 CALL DX

-t

AX=1A00 BX=290D CX=0000 DX=1ACC SP=2F61 BP=0000 SI=2A48 DI=2CDE
DS=7CB0 ES=7CB0 SS=7CB0 CS=7CB0 IP=1ACC NV UP EI PL ZR NA PE NC
7CB0:1ACC 33C0 XOR AX,AX

```

-u
7CB0:1ACC 33C0      XOR     AX,AX
7CB0:1ACE A3B92C    MOV     [2CB9],AX
7CB0:1AD1 A2B22C    MOV     [2CB2],AL
7CB0:1AD4 A2CC2E    MOV     [2ECC],AL
7CB0:1AD7 A2CB2E    MOV     [2ECB],AL
7CB0:1ADA A2CB2E    MOV     [2ECB],AL
7CB0:1ADD A3C42C    MOV     [2CC4],AX
7CB0:1AE0 A2C92E    MOV     [2EC9],AL
7CB0:1AE3 A2B12C    MOV     [2CB1],AL
7CB0:1AE6 A2102E    MOV     [2E10],AL
7CB0:1AE9 A36F2E    MOV     [2E6F],AX

-u
7CB0:1AEC A3B82D    MOV     [2DB8],AX
7CB0:1AEF A3182E    MOV     [2E18],AX
7CB0:1AF2 A3982D    MOV     [2D98],AX
7CB0:1AF5 A3552D    MOV     [2D55],AX
7CB0:1AF8 A3152D    MOV     [2D15],AX
7CB0:1AFB 4B      DEC     AX
7CB0:1AFC A2C12C    MOV     [2CC1],AL
7CB0:1AFF A20F2E    MOV     [2E0F],AL
7CB0:1B02 A2B32D    MOV     [2DB3],AL
7CB0:1B05 BEB100    MOV     SI,00B1
7CB0:1B08 B32B      MOV     BL,2B
7CB0:1B0A 33ED      XOR     BP,BP

-u
7CB0:1B0C BF6F2E    MOV     DI,2E6F
7CB0:1B0F E80708    CALL    2319
7CB0:1B12 7247      JB      1B5B
7CB0:1B14 F6C780    TEST    BH,80
7CB0:1B17 7405      JZ      1B1E
7CB0:1B19 C606CC2E01  MOV     BYTE PTR [2ECC],01
7CB0:1B1E F6C7C1    TEST    BH,01
7CB0:1B21 740A      JZ      1B2D
7CB0:1B23 092EB72C  OR      [2CB7],BP
7CB0:1B27 092EB92C  OR      [2CB9],BP
7CB0:1B2B EBDD      JMF     1B0A

-u
7CB0:1B2D F6C780    TEST    BH,80
7CB0:1B30 7504      JNZ     1B36
7CB0:1B32 FE06B22C  INC     BYTE PTR [2CB2]
7CB0:1B36 56      PUSH   SI
7CB0:1B37 A1D32E    MOV     AX,[2ED3]
7CB0:1B3A BE6F2E    MOV     SI,2E6F
7CB0:1B3D 2BC6      SUB     AX,SI
7CB0:1B3F BFB82D    MOV     DI,2DB8
7CB0:1B42 03C7      ADD     AX,DI
7CB0:1B44 A3B52D    MOV     [2DB5],AX

```

: COPY子程序入口为CS:1A CC

: 清工作区

: COPY源文件名单元CS: 2 E 18

CS: 2 D 98

: DOS命令文件名单元CS: 2 D 55

: 目标文件名单元CS: 2 D 15

7CB0:1B47	8B0EB42D	MOV	[2DB4],CL
7CB0:1B48	41	INC	CX
7CB0:1B4C	F3	REPZ	
7CB0:1B4D	A4	MOVSB	
-u			
7CB0:1B4E	8B3EB72D	MOV	[2DB7],BH
7CB0:1B52	C706B72C0000	MOV	WORD PTR [2CB7],0000
7CB0:1B58	5E	POP	SI
7CB0:1B59	EBAF	JMP	1B0A
7CB0:1B5B	ACCC2E	MOV	AL,[2ECC]
7CB0:1B5E	A2B12C	MOV	[2CB1],AL
7CB0:1B61	D0E0	SHL	AL,1
7CB0:1B63	D0E0	SHL	AL,1
7CB0:1B65	A2C52E	MOV	[2EC5],AL
7CB0:1B68	BA8329	MOV	DX,2983
7CB0:1B6B	A0B22C	MOV	AL,[2CB2]
-u			
7CB0:1B6E	0AC0	OR	AL,AL
7CB0:1B70	7404	JZ	1B76
7CB0:1B72	3C02	CMP	AL,02
7CB0:1B74	7603	JBE	1B79
7CB0:1B76	E993EB	JMP	070C
7CB0:1B79	BDB32D	MOV	BP,2DB3
7CB0:1B7C	3C01	CMP	AL,01
7CB0:1B7E	7520	JNZ	1BA0
7CB0:1B80	A0B02C	MOV	AL,[2CB0]
7CB0:1B83	0441	ADD	AL,41
7CB0:1B85	B43A	MOV	AH,3A
7CB0:1B87	C6460102	MOV	BYTE PTR [BP+01],02
7CB0:1B8B	BFB82D	MOV	DI,2DB8
-u			
7CB0:1B8E	AB	STOSW	
7CB0:1B8F	C706B72C0000	MOV	WORD PTR [2CB7],0000
7CB0:1B95	C6460402	MOV	BYTE PTR [BP+04],02
7CB0:1B99	C6460000	MOV	BYTE PTR [BP+00],00
7CB0:1B9D	EB9006	CALL	2230
7CB0:1BA0	807E0102	CMP	BYTE PTR [BP+01],02
7CB0:1BA4	7515	JNZ	1BBB
7CB0:1BA6	803EB92D3A	CMP	BYTE PTR [2DB9],3A
7CB0:1BAB	750E	JNZ	1BBB
7CB0:1BAD	804E0402	OR	BYTE PTR [BP+04],02
-u			
7CB0:1BB1	BFB82D	MOV	DI,2DB8
7CB0:1BB4	C6460000	MOV	BYTE PTR [BP+00],00
7CB0:1BB8	EB7506	CALL	2230
7CB0:1BBB	8B7E02	MOV	DI,[BP+02]
7CB0:1BBE	803D00	CMP	BYTE PTR [DI],00
7CB0:1BC1	7514	JNZ	1BD7

7CB0:1BC3 BAE227	MOV	DX,27E2
7CB0:1BC6 807DFE3A	CMP	BYTE PTR [DI-02],3A
7CB0:1BCA 75AA	JNZ	1B76
7CB0:1BCC C6460002	MOV	BYTE PTR [BP+001],02
7CB0:1BD0 804E0406	OR	BYTE PTR [BP+04],06
-u		
7CB0:1BD4 EB5906	CALL	2230
7CB0:1BD7 BAEC26	MOV	DX,26EC
7CB0:1BDA A1B92C	MOV	AX,[2CB9]
7CB0:1BDD 3D0080	CMP	AX,B000
7CB0:1BE0 7494	JZ	1B76
7CB0:1BE2 A91000	TEST	AX,0010
7CB0:1BE5 7414	JZ	1BFB
7CB0:1BE7 B454	MOV	AH,54
7CB0:1BE9 CD21	INT	21
7CB0:1BEB 1E	PUSH	DS
7CB0:1BEC BE1EA02C	MOV	DS,[2CA0]
7CB0:1BF0 32E4	XOR	AH,AH
7CB0:1BF2 A37509	MOV	[0975],AX
-u		
7CB0:1BF5 1F	POP	DS
7CB0:1BF6 B8012E	MOV	AX,2E01
7CB0:1BF9 CD21	INT	21
7CB0:1BFB 33ED	XOR	BP,BP
7CB0:1BFD BEB100	MOV	SI,00B1
7CB0:1C00 B32B	MOV	BL,2B
7CB0:1C02 BF6F2E	MOV	DI,2E6F
7CB0:1C05 EB1107	CALL	2319
7CB0:1C08 F6C701	TEST	BH,01
7CB0:1C0B 75F5	JNZ	1C02
7CB0:1C0D 092EB72C	OR	[2CB7],BP
7CB0:1C11 F7C50800	TEST	BP,000B
-u		
7CB0:1C15 750C	JNZ	1C23
7CB0:1C17 803EB12C00	CMP	BYTE PTR [2CB1],00
7CB0:1C1C 7405	JZ	1C23
7CB0:1C1E C606CB2E04	MOV	BYTE PTR [2ECB],04
7CB0:1C23 56	PUSH	SI
7CB0:1C24 A1D32E	MOV	AX,[2ED3]
7CB0:1C27 BE6F2E	MOV	SI,2E6F
7CB0:1C2A 2BC6	SUB	AX,SI
7CB0:1C2C BF182E	MOV	DI,2E18
7CB0:1C2F 03C7	ADD	AX,DI
7CB0:1C31 A3152E	MOV	[2E15],AX
7CB0:1C34 880E142E	MOV	[2E14],CL
-u		
7CB0:1C38 41	INC	CX
7CB0:1C39 F3	REPZ	

7CB0:1C3A A4	MOVSB	
7CB0:1C3B 883E172E	MOV	[2E17],BH
7CB0:1C3F 5E	POP	SI
7CB0:1C40 8BC5	MOV	AX,BP
7CB0:1C42 E82004	CALL	2065
7CB0:1C45 E8BFE7	CALL	0407
7CB0:1C48 EB1A04	CALL	2065
7CB0:1C4B E85D00	CALL	1CAB
7CB0:1C4E E99B00	JMP	1CEC
7CB0:1C51 E81202	CALL	1E66
7CB0:1C54 8B36C42C	MOV	SI,[2CC4]
-u		
7CB0:1C58 33FF	XOR	DI,DI
7CB0:1C5A E875EF	CALL	0BD2
7CB0:1C5D B8526	MOV	DX,2685
7CB0:1C60 E86D06	CALL	22D0
7CB0:1C63 E99EE4	JMP	0104
7CB0:1C66 803EB12C00	CMP	BYTE PTR [2CB1],00
7CB0:1C6B 7558	JNZ	1CC8
7CB0:1C6D BA182E	MOV	DX,2E18
7CB0:1C70 E86206	CALL	22D5
7CB0:1C73 E842EF	CALL	0BB8
7CB0:1C76 BADB25	MOV	DX,25DB
-u		
7CB0:1C79 E9AF03	JMP	2028
7CB0:1C7C 56	PUSH	SI
7CB0:1C7D A1D32E	MOV	AX,[2ED3]
7CB0:1C80 BE6F2E	MOV	SI,2E6F
7CB0:1C83 2BC6	SUB	AX,SI
7CB0:1C85 BF182E	MOV	DI,2E18
7CB0:1C88 03C7	ADD	AX,DI
7CB0:1C8A A3152E	MOV	[2E15],AX
7CB0:1C8D 880E142E	MOV	[2E14],CL
7CB0:1C91 41	INC	CX
7CB0:1C92 F3	REPZ	
7CB0:1C93 A4	MOVSB	
7CB0:1C94 883E172E	MOV	[2E17],BH
7CB0:1C98 5E	POP	SI
-u		
7CB0:1C99 8BC5	MOV	AX,BP
7CB0:1C9B E8C703	CALL	2065
7CB0:1C9E E866E7	CALL	0407
7CB0:1CA1 E8C103	CALL	2065
7CB0:1CA4 803EB12C00	CMP	BYTE PTR [2CB1],00
7CB0:1CA9 750B	JNZ	1CB6
7CB0:1CAB 33C0	XOR	AX,AX
7CB0:1CAD A2BB2C	MOV	[2CBB],AL
7CB0:1CB0 A3BF2C	MOV	[2CBF],AX

```

7CB0:1CB3 A2BC2C      MOV     [2CBC],AL
7CB0:1CB6 8936C32E     MOV     [2EC3],SI
-u
7CB0:1CBA BF0C2C      MOV     DI,2C0C
7CB0:1CBD BD132E      MOV     BP,2E13
7CB0:1CC0 E89E04      CALL    2161
7CB0:1CC3 8B36152E     MOV     SI,[2E15]
7CB0:1CC7 C3          RET
7CB0:1CC8 803ECC2E00    CMP     BYTE PTR [2ECC],00
7CB0:1CCD 7502      JNZ     1CD1
7CB0:1CCF EB80      JMP     1C51
7CB0:1CD1 33ED      XOR     BP,BP
7CB0:1CD3 8B36C32E     MOV     SI,[2EC3]
7CB0:1CD7 B32B      MOV     BL,2B
7CB0:1CD9 BF6F2E     MOV     DI,2E6F
-g=cs:1acc,471

```

COPY程序执行后跳回A>，再打命令看文件控制区形成。

```

A>debug command.com
-d7cb0:29a0
7CB0:29A0 0A 24 45 72 72 6F 72 20-77 72 69 74 69 6E 67 20      .Error writing
7CB0:29B0 74 6F 20 64 65 76 69 63-65 0D 0A 24 08 20 08 00      to device..$. ..
7CB0:29C0 04 18 5B 32 4A 44 4B 18-54 E1 18 50 93 11 4E 83      ..[2JDK.Ta.P..N.
7CB0:29D0 11 56 60 0E 47 07 0F 4C-08 0F 42 0F 0F 5F BD 0B      .V'.G..L..B..=.
7CB0:29E0 24 DA 19 45 03 0F 4B F9-0E 51 FF 0E 00 03 4E 4F      $Z.E..Hy.Q....NO
7CB0:29F0 54 9A 15 0A 45 52 52 4F-52 4C 45 56 45 4C 28 16      T...ERRORLEVEL(.
7CB0:2A00 05 45 58 49 53 54 EA 15-00 04 44 49 52 01 FA 09      .EXISTj...DIR.z.
7CB0:2A10 07 52 45 4E 41 4D 45 01-B2 0C 04 52 45 4E 01 B2      .RENAME.2..REN.2
-d
7CB0:2A20 0C 06 45 52 41 53 45 01-3D 0C 04 44 45 4C 01 3D      ..ERASE.=..DEL.=
7CB0:2A30 0C 05 54 59 50 45 01 EF-0C 04 52 45 4D 00 04 01      ..TYPE.o..REM...
7CB0:2A40 05 43 4F 50 59 01 CC 1A-06 50 41 55 53 45 00 1F      .COPY.L..PAUSE..
7CB0:2A50 13 05 44 41 54 45 00 3B-18 05 54 49 4D 45 00 CE      ..DATE.B..TIME.N
7CB0:2A60 18 04 56 45 52 00 57 0E-04 56 4F 4C 01 C8 0D 03      ..VER.W..VOL.H..
7CB0:2A70 43 44 01 A6 12 06 43 48-44 49 52 01 A6 12 03 4D      CD.&..CHDIR.&..M
7CB0:2A80 44 01 D9 12 06 4D 4B 44-49 52 01 D9 12 03 52 44      D.Y..MKDIR.Y..RD
7CB0:2A90 01 0E 13 06 52 4D 44 49-52 01 0E 13 06 42 52 45      ....RMDIR....BRE
-d
7CB0:2AA0 41 48 00 92 17 07 56 45-52 49 46 59 00 C7 17 04      AK....VERIFY.G..
7CB0:2AB0 53 45 54 00 0F 10 07 50-52 4F 4D 50 54 00 FA 0F      SET....PROMPT.z.
7CB0:2AC0 05 50 41 54 48 00 A0 0F-05 45 58 49 54 00 C9 11      .PATH. ..EXIT.I. : 在命令表下面有
7CB0:2AD0 05 43 54 54 59 01 F7 11-05 45 43 48 4F 00 59 17      .CTTY.w..ECHO.Y. 文件处理工作区
7CB0:2AE0 05 47 4F 54 4F 00 96 16-06 53 48 49 46 54 00 56      .GOTO....SHIFT.V

```

```

7CB0:2AF0 16 03 49 46 00 50 15 04-46 4F 52 00 68 14 04 43 ..IF.P..FOR.h..C
7CB0:2B00 4C 53 00 53 12 00 80 11-64 65 62 75 67 20 63 6F LS.S....debug co
7CB0:2B10 6D 6D 61 6E 64 2E 63 6F-6D 0D 0D 00 00 00 00 00 mand.com.....
-d
7CB0:2B20 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
7CB0:2B30 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
7CB0:2B40 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
7CB0:2B50 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
7CB0:2B60 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
7CB0:2B70 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
7CB0:2B80 00 00 00 00 00 00 00 00-00 80 11 64 65 62 75 67 .....debug
7CB0:2B90 20 63 6F 6D 6D 61 6E 64-2E 63 6F 6D 0D 0D 00 00 command.com....
-d
7CB0:2BA0 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
7CB0:2BB0 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
7CB0:2BC0 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
7CB0:2BD0 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
7CB0:2BE0 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
7CB0:2BF0 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
7CB0:2C00 00 00 00 00 00 00 00 00-00 00 00 00 41 3A 5C 00 .....A:\.
7CB0:2C10 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
-d
7CB0:2C20 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
7CB0:2C30 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
7CB0:2C40 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 41 .....A
7CB0:2C50 3A 44 45 42 55 47 2E 43-4F 4D 00 00 00 00 00 00 :DEBUG.COM.....
7CB0:2C60 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
7CB0:2C70 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
7CB0:2C80 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
7CB0:2C90 00 00 00 00 00 00 00 00-00 00 00 00 00 00 91 04 .....
-d
7CB0:2CA0 13 09 01 09 2F 5C 83 02-13 09 64 04 13 09 00 04 ..D./\....d.....
7CB0:2CB0 00 00 00 00 00 00 00 00-00 00 00 00 00 FF FF 00 .....
7CB0:2CC0 00 00 00 00 01 00 3E 20-20 20 20 20 20 20 31 00 .....> 1.
7CB0:2CD0 00 00 00 00 00 00 00 00-00 00 44 45 42 55 47 20 .....DEBUG
7CB0:2CE0 20 20 3F 3F 3F 00 00 00-00 16 00 1F 42 E3 00 00 ???.....Bc..
7CB0:2CF0 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
7CB0:2D00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
7CB0:2D10 00 00 00 00 00 00 44 44-20 20 20 20 20 20 54 58 .....DD TX : 目标文件名
-d
7CB0:2D20 54 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 T.....
7CB0:2D30 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
7CB0:2D40 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
7CB0:2D50 00 00 00 00 00 01 44 45-42 55 47 20 20 20 43 4F .....DEBUG CD
7CB0:2D60 4D 20 00 00 00 00 00 00-00 00 00 00 00 60 68 06 M .....h.
7CB0:2D70 91 00 80 2E 00 00 00 00-00 00 00 00 00 00 00 00 .....
7CB0:2D80 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
7CB0:2D90 00 00 00 00 00 00 00 00-01 44 52 20 20 20 20 20 .....DR

```



```

7CB0:013C 36          SS:
7CB0:013D C606062B80  MOV   BYTE PTR [2B06],80
7CB0:0142 36          SS:
7CB0:0143 C606892B80  MOV   BYTE PTR [2B89],80
7CB0:0148 0BED        OR    BP,BP
7CB0:014A 7409        JZ    0155
-q=cs:12c,456

```

执行此段后，DOS初始设定完毕，推出DOS提示符A >，然后打入显示目录命令
A:\dir/w

```

AX=09FA BX=290D CX=0000 DX=2B00 SP=2F63 BP=0000 SI=2A10 DI=2CDD
DS=7CB0 ES=7CB0 SS=7CB0 CS=7CB0 IP=0456 NV UP EI PL ZR NA PE NC
7CB0:0456 75EC          JNZ   0444
-t

```

```

AX=09FA BX=290D CX=0000 DX=2B00 SP=2F63 BP=0000 SI=2A10 DI=2CDD
DS=7CB0 ES=7CB0 SS=7CB0 CS=7CB0 IP=0458 NV UP EI PL ZR NA PE NC
7CB0:0458 8B00          MOV   DX,AX
-u

```

```

7CB0:0458 8B00          MOV   DX,AX
7CB0:045A 803EAE2C00    CMP   BYTE PTR [2CAE],00
7CB0:045F 740B          JZ    046C
7CB0:0461 A0B12C        MOV   AL,[2CB1]
7CB0:0464 0A06B22C     OR    AL,[2CB2]
7CB0:0468 3CFF        CMP   AL,FF
7CB0:046A 74CD        JZ    0439
7CB0:046C E87315     CALL 19E2
7CB0:046F FFD2        CALL DX
7CB0:0471 E990FC     JMP 0104
7CB0:0474 E989FC     JMP 0100
7CB0:0477 FECA        DEC   DL
-q=cs:458,46a

```

运行此段后尚未执行DIR/W命令

```

AX=0900 BX=290D CX=0000 DX=09FA SP=2F63 BP=0000 SI=2A10 DI=2CDD
DS=7CB0 ES=7CB0 SS=7CB0 CS=7CB0 IP=046A NV UP EI PL NZ AC PD CY
7CB0:046A 74CD        JZ    0439
-t

```

```

AX=0900 BX=290D CX=0000 DX=09FA SP=2F63 BP=0000 SI=2A10 DI=2CDD
DS=7CB0 ES=7CB0 SS=7CB0 CS=7CB0 IP=046C NV UP EI PL NZ AC PD CY
7CB0:046C E87315     CALL 19E2
-u
7CB0:046C E87315     CALL 19E2
7CB0:046F FFD2        CALL DX
7CB0:0471 E990FC     JMP 0104
7CB0:0474 E989FC     JMP 0100
7CB0:0477 FECA        DEC   DL
7CB0:0479 FEC8        DEC   AL

```

```

7CB0:047B 74F7      JZ      0474
7CB0:047D C606AF2C00    MOV     BYTE PTR [2CAF],00
7CB0:0482 8A16BC2C    MOV     DL,[2CBC]
7CB0:0486 8B16D92C    MOV     [2CD9],DL
7CB0:048A EBA10E      CALL    132E
-g=c5:46c,46f

```

```

AX=0900 BX=290D CX=0000 DX=09FA SP=2F63 BP=0000 SI=2A10 DI=2CDD
DS=7CB0 ES=7CB0 SS=7CB0 CS=7CB0 IP=046F NV UP EI PL ZR NA PE NC
7CB0:046F FFD2      CALL    DX ; DX为DIR/W命令子程序入口 09FAH
-t

```

```

AX=0900 BX=290D CX=0000 DX=09FA SP=2F61 BP=0000 SI=2A10 DI=2CDD
DS=7CB0 ES=7CB0 SS=7CB0 CS=7CB0 IP=09FA NV UP EI PL ZR NA PE NC
7CB0:09FA EBED03      CALL    ODEA

```

```

-u
7CB0:09FA EBED03      CALL    ODEA
7CB0:09FD B03F      MOV     AL,3F
7CB0:09FF BF5D00    MOV     DI,005D
7CB0:0A02 B90B00    MOV     CX,000B
7CB0:0A05 F3       REPZ
7CB0:0A06 AA       STOSB
7CB0:0A07 BEB100    MOV     SI,00B1
7CB0:0A0A EBF9F9    CALL    0407
7CB0:0A0D BF5C00    MOV     DI,005C ; 命令分析
7CB0:0A10 B80D29    MOV     AX,290D
7CB0:0A13 CD21      INT     21
7CB0:0A15 A1B32C    MOV     AX,[2CB3]
7CB0:0A18 0B06B92C    OR      AX,[2CB9]

```

```

-u
7CB0:0A1C A3A42D    MOV     [2DA4],AX
7CB0:0A1F C606AC2D17  MOV     BYTE PTR [2DAC],17
7CB0:0A24 A801      TEST    AL,01
7CB0:0A26 B001      MOV     AL,01
7CB0:0A2B 7402      JZ      0A2C
7CB0:0A2A B005      MOV     AL,05
7CB0:0A2C A2C32C    MOV     [2CC3],AL
7CB0:0A2F A2C22C    MOV     [2CC2],AL
7CB0:0A32 C706C42C0000  MOV     WORD PTR [2CC4],0000
7CB0:0A38 BA552D    MOV     DX,2D55
7CB0:0A3B B41A      MOV     AH,1A ; 设DTA地址在DS:DX

```

```

-u
7CB0:0A3D CD21      INT     21
7CB0:0A3F EB31FF    CALL    0973
7CB0:0A42 9C       PUSHF
7CB0:0A43 730A     JNB     0A4F
7CB0:0A45 B03EB32D00  CMP     BYTE PTR [2DB3],00
7CB0:0A4A 7403     JZ      0A4F

```

7CB0:0A4C E90C02	JMP	0C5B	
7CB0:0A4F BE5C00	MOV	SI,005C	
7CB0:0A52 BF552D	MOV	DI,2D55	
7CB0:0A55 89D7	MOV	DX,DI	
7CB0:0A57 B90C00	MOV	CX,000C	
7CB0:0A5A F3	REPZ		
7CB0:0A5B A4	MOVSB		
7CB0:0A5E B40F	MOV	AH,0F	: 打开文件FCB在DS:DX
-u			
7CB0:0A5E CD21	INT	21	
7CB0:0A60 BA6328	MOV	DX,2863	
7CB0:0A63 50	PUSH	AX	
7CB0:0A64 E86918	CALL	22D0	
7CB0:0A67 EB2207	CALL	118C	
7CB0:0A6A E85001	CALL	0BBD	
7CB0:0A6D 58	POP	AX	
7CB0:0A6E 0AC0	OR	AL,AL	
7CB0:0A70 750A	JNZ	0A7C	
7CB0:0A72 F6066D2D80	TEST	BYTE PTR [2D6D],80	
7CB0:0A77 7403	JZ	0A7C	
7CB0:0A79 E9B8FC	JMP	0734	
7CB0:0A7C B411	MOV	AH,11	: 找文件名首地址
-u			
7CB0:0A7E C6065500FF	MOV	BYTE PTR [0055],FF	
7CB0:0A83 C6065B0010	MOV	BYTE PTR [005B],10	
7CB0:0A88 9D	POPF		
7CB0:0A89 7226	JB	0AB1	
7CB0:0A8B 7418	JZ	0AA5	
7CB0:0A8D 803F	MOV	AL,3F	
7CB0:0A8F BF5D00	MOV	DI,005D	
7CB0:0A92 B90B00	MOV	CX,000B	
7CB0:0A95 F3	REPZ		
7CB0:0A96 AA	STOSB		
7CB0:0A97 8B36B52D	MOV	SI,[2DB5]	
7CB0:0A9B BF5C00	MOV	DI,005C	
-u			
7CB0:0A9E B80E29	MOV	AX,290E	
7CB0:0AA1 CD21	INT	21	
7CB0:0AA3 B411	MOV	AH,11	
7CB0:0AA5 BA5500	MOV	DX,0055	
7CB0:0AAB CD21	INT	21	
7CB0:0AAA 50	PUSH	AX	
7CB0:0AAB E8A102	CALL	0D4F	
7CB0:0AAE 58	POP	AX	
7CB0:0AAF EB05	JMP	0AB6	
7CB0:0AB1 BA5500	MOV	DX,0055	
7CB0:0AB4 CD21	INT	21	
7CB0:0AB6 FEC0	INC	AL	

7CB0:0AB8 7506	JNZ	0AC0
7CB0:0ABA E96FFC	JMP	072C
7CB0:0ABD E9A100	JMP	0B61
-u		
7CB0:0AC0 FF06C42C	INC	WORD PTR [2CC4]
7CB0:0AC4 BE5D2D	MOV	SI,2D5D
7CB0:0AC7 EBD200	CALL	0B9C
7CB0:0ACA F606A42D01	TEST	BYTE PTR [2DA4],01
7CB0:0ACF 75EC	JNZ	0ABD
7CB0:0AD1 BE682D	MOV	SI,2D68
7CB0:0AD4 F60410	TEST	BYTE PTR [SI],10
7CB0:0AD7 740B	JZ	0AE1
7CB0:0AD9 BA9027	MOV	DX,2790
7CB0:0ADC E8F117	CALL	22D0
7CB0:0ADF EB03	JMP	0AE4
-u		
7CB0:0AE1 EBE600	CALL	0BCA
7CB0:0AE4 A1752D	MOV	AX,[2D75]
7CB0:0AE7 0BC0	OR	AX,AX
7CB0:0AE9 7476	JZ	0B61
7CB0:0AEB BFC62C	MOV	DI,2CC6
7CB0:0AEE 50	PUSH	AX
7CB0:0AEF BB2020	MOV	AX,2020
7CB0:0AF2 AB	STOSW	
7CB0:0AF3 5B	POF	AX
7CB0:0AF4 8BD0	MOV	DX,AX
7CB0:0AF6 B105	MOV	CL,05
7CB0:0AFB D3EB	SHR	AX,CL
7CB0:0AFA 240F	AND	AL,0F
7CB0:0AFC B710	MOV	BH,10
7CB0:0AFE EBCB0E	CALL	19C9
-u		
7CB0:0B01 B02D	MOV	AL,2D
7CB0:0B03 AA	STOSB	
7CB0:0B04 8AC2	MOV	AL,DL
7CB0:0B06 241F	AND	AL,1F
7CB0:0B08 EBBE0E	CALL	19C9
7CB0:0B0B B02D	MOV	AL,2D
7CB0:0B0D AA	STOSB	
7CB0:0B0E 8AC6	MOV	AL,DH
7CB0:0B10 D0E8	SHR	AL,1
7CB0:0B12 055000	ADD	AX,0050
7CB0:0B15 3C64	CMP	AL,64
7CB0:0B17 7202	JB	0B1B
7CB0:0B19 2C64	SUB	AL,64
7CB0:0B1B EBAB0E	CALL	19C9
7CB0:0B1E 8B1E732D	MOV	BX,[2D73]

-u

7CB0:0B22 0BDB	OR	BX, BX
7CB0:0B24 7432	JZ	0B58
7CB0:0B26 B82020	MOV	AX, 2020
7CB0:0B29 AB	STOSW	
7CB0:0B2A D1EB	SHR	BX, 1
7CB0:0B2C D1EB	SHR	BX, 1
7CB0:0B2E D1EB	SHR	BX, 1
7CB0:0B30 D0EB	SHR	BL, 1
7CB0:0B32 D0EB	SHR	BL, 1
7CB0:0B34 8AC7	MOV	AL, BH
7CB0:0B36 B661	MOV	DH, 61
7CB0:0B38 3C0C	CMPS	AL, 0C
7CB0:0B3A 7206	JB	0B42
7CB0:0B3C B670	MOV	DH, 70
7CB0:0B3E 7402	JZ	0B42
7CB0:0B40 2C0C	SUB	AL, 0C
-u		
7CB0:0B42 0AC0	OR	AL, AL
7CB0:0B44 7502	JNZ	0B48
7CB0:0B46 B00C	MOV	AL, 0C
7CB0:0B48 B710	MOV	BH, 10
7CB0:0B4A EB7C0E	CALL	19C9
7CB0:0B4D B03A	MOV	AL, 3A
7CB0:0B4F AA	STOSB	
7CB0:0B50 8AC3	MOV	AL, BL
7CB0:0B52 EB740E	CALL	19C9
7CB0:0B55 8AC6	MOV	AL, DH
7CB0:0B57 AA	STOSB	
7CB0:0B58 33C0	XOR	AX, AX
7CB0:0B5A AA	STOSB	
7CB0:0B5B BAC62C	MOV	DX, 2CC6
7CB0:0B5E EB7417	CALL	22D5
7CB0:0B61 FE0EC22C	DEC	BYTE PTR [2CC2]
-u		
7CB0:0B65 752E	JNZ	0B95
7CB0:0B67 A0C32C	MOV	AL, [2CC3]
7CB0:0B6A A2C22C	MOV	[2CC2], AL
7CB0:0B6D E84D00	CALL	0BBD
7CB0:0B70 F606A42D02	TEST	BYTE PTR [2DA4], 02
7CB0:0B75 7419	JZ	0B90
7CB0:0B77 FE0EAC2D	DEC	BYTE PTR [2DAC]
7CB0:0B7B 7513	JNZ	0B90
7CB0:0B7D C606AC2D17	MOV	BYTE PTR [2DAC], 17
7CB0:0B82 BACD26	MOV	DX, 26CD
-u		
7CB0:0B85 EB4817	CALL	22D0
7CB0:0B88 B9080C	MOV	AX, 0C08
7CB0:0B8B CD21	INT	21

7CB0:0B8D E82D00	CALL	0BB0
7CB0:0B90 B412	MOV	AH,12
7CB0:0B92 E91CFF	JMP	0AB1
7CB0:0B95 B009	MOV	AL,09
7CB0:0B97 E8400E	CALL	19DA
7CB0:0B9A EBF4	JMP	0B90
7CB0:0B9C BFC62C	MOV	DI,2CC6
7CB0:0B9F B90B00	MOV	CX,000B
7CB0:0BA2 F3	REPZ	
7CB0:0BA3 A4	MOVSB	
7CB0:0BA4 8020	MOV	AL,20
-u		
7CB0:0BA6 AA	STOSB	
7CB0:0BA7 B90300	MOV	CX,0003
7CB0:0BAA F3	REPZ	
7CB0:0BAB A4	MOVSB	
7CB0:0BAC 33C0	XOR	AX,AX
7CB0:0BAE AA	STOSB	
7CB0:0BAF 52	PUSH	DX
7CB0:0BB0 BAC62C	MOV	DX,2CC6
7CB0:0BB3 E81F17	CALL	22D5
7CB0:0BB6 5A	POP	DX
7CB0:0BB7 C3	RET	
7CB0:0BB8 8020	MOV	AL,20
7CB0:0BBA E91D0E	JMP	19DA
7CB0:0BBD 52	PUSH	DX
7CB0:0BBE B6B929	MOV	DX,29B9
7CB0:0BC1 1E	PUSH	DS
7CB0:0BC2 0E	PUSH	CS
7CB0:0BC3 1F	POP	DS
7CB0:0BC4 E80917	CALL	22D0
-u		
7CB0:0BC7 1F	POP	DS
7CB0:0BC8 5A	POP	DX
7CB0:0BC9 C3	RET	
7CB0:0BCA 8B36792D	MOV	SI,[2D79]
7CB0:0BCE 8B3E7B2D	MOV	DI,[2D7B]
7CB0:0BD2 33C0	XOR	AX,AX
7CB0:0BD4 8BD8	MOV	BX,AX
7CB0:0BD6 8BE8	MOV	BP,AX
7CB0:0BD8 B92000	MOV	CX,0020
7CB0:0BDB 01E6	SHL	SI,1
7CB0:0BDD 01D7	RCL	DI,1
7CB0:0BDF 95	XCHG	BP,AX
7CB0:0BE0 E84F00	CALL	0C32
7CB0:0BE3 95	XCHG	BP,AX
7CB0:0BE4 93	XCHG	BX,AX
7CB0:0BE5 E84A00	CALL	0C32

-g=cs:46f,471 : 执行DIR/W命令后返回DEBUG

Volume in drive A has no label
Directory of A:\

COMMAND	COM	FORMAT	COM	CHKDSK	COM	SYS	COM	DISKCOPY	COM
DISKCOMP	COM	COMP	COM	EDLIN	COM	MODE	COM	FDISK	COM
BACKUP	COM	RESTORE	COM	PRINT	COM	RECOVER	COM	ASSIGN	COM
TREE	COM	GRAPHICS	COM	MORE	COM	FONT	BAS	BASICA	COM
DEBUG	COM	RS	COM	TIMER	COM	DR	COM	SD	COM
DISKTEST	COM	2024P16	EXE	ALL9PH	EXE	NEW9PH	EXE	2024PH	EXE
MUSIC	BAS	DONKEY	BAS	CIRCLE	BAS	PIECHART	BAS	SPACE	BAS
BALL	BAS	COMM	BAS	6WBASIC	EXE	RS	COM	TIMER	COM
DR	COM	SD	COM	DISKTEST	COM	DR	DOC	CONFIG	SYS
DD	TXT								

46 File(s) 106496 bytes free

AX=0000 BX=0020 CX=0F00 DX=26A1 SP=2F61 BP=0030 SI=0000 DI=2CD0
DS=7CB0 ES=7CB0 SS=7CB0 CS=7CB0 IP=0471 NV UP EI PL ZR NA PE NC
7CB0:0471 E990FC JMP 0104

-a

AX=0000 BX=0020 CX=0F00 DX=26A1 SP=2F61 BP=0030 SI=0000 DI=2CD0
DS=7CB0 ES=7CB0 SS=7CB0 CS=7CB0 IP=0104 NV UP EI PL ZR NA PE NC
7CB0:0104 2E CS:
7CB0:0105 BE1EA02C MOV DS,[2CA0] CS:2CA0=0CD9

-u

7CB0:0104 2E CS:
7CB0:0105 BE1EA02C MOV DS,[2CA0]
7CB0:0109 B8FFFF MOV AX,FFFF
7CB0:010C B7067509 XCHG [0975],AX
7CB0:0110 3DFFFF CMP AX,FFFF
7CB0:0113 7404 JZ 0119
7CB0:0115 B42E MOV AH,2E
7CB0:0117 CD21 INT 21
7CB0:0119 2E CS:
7CB0:011A FF1E9E2C CALL FAR [2C9E]
7CB0:011E 33ED XOR BP,BP
7CB0:0120 833E7309FF CMP WORD PTR [0973],-01

-u

7CB0:0125 7505 JNZ 012C
7CB0:0127 0E PUSH CS
7CB0:0128 1F POP DS
7CB0:0129 E99D10 JMP 11C9
7CB0:012C FC CLD
7CB0:012D 8CCB MOV AX,CS
7CB0:012F BED0 MOV SS,AX
7CB0:0131 BC632F MOV SP,2F63

```

7CB0:0134 BEC0      MOV     ES,AX
7CB0:0136 36        SS:
7CB0:0137 8E1EA02C     MOV     DS,[2CA0]
7CB0:013B FB        STI
7CB0:013C 36        SS:
7CB0:013D C606062B80     MOV     BYTE PTR [2B06],80
7CB0:0142 36        SS:
7CB0:0143 C606892B80     MOV     BYTE PTR [2B89],80

```

-t

```

AX=FFFF BX=0020 CX=0F00 DX=26A1 SP=2F61 BP=0030 SI=0000 DI=2CD0
DS=0CD9 ES=7CB0 SS=7CB0 CS=7CB0 IP=010C NV UP EI PL ZR NA PE NC
7CB0:010C 87067509      XCHG   [0975],AX          DS:0975=FFFF
-g:cs:10c,110

```

```

AX=FFFF BX=0020 CX=0F00 DX=26A1 SP=2F61 BP=0030 SI=0000 DI=2CD0
DS=0CD9 ES=7CB0 SS=7CB0 CS=7CB0 IP=0110 NV UP EI PL ZR NA PE NC
7CB0:0110 3DFFFF      CMP    AX,FFFF
-g:cs:104,119

```

```

AX=FFFF BX=0020 CX=0F00 DX=26A1 SP=2F61 BP=0030 SI=0000 DI=2CD0
DS=0CD9 ES=7CB0 SS=7CB0 CS=7CB0 IP=0119 NV UP EI PL ZR NA PE NC
7CB0:0119 2E        CS:
7CB0:011A FF1E9E2C     CALL   FAR [2C9E]        CS:2C9E=499

```

-t

```

AX=FFFF BX=0020 CX=0F00 DX=26A1 SP=2F5D BP=0030 SI=0000 DI=2CD0
DS=0CD9 ES=7CB0 SS=7CB0 CS=0CD9 IP=0491 NV UP EI PL ZR NA PE NC
0CD9:0491 E80100      CALL   0495

```

-u

```

0CD9:0491 E80100      CALL   0495
0CD9:0494 CB        RETF
0CD9:0495 33DB      XOR    BX,BX
0CD9:0497 8B0E6F09     MOV     CX,[096F]
0CD9:049B 8B161800     MOV     DX,[0018]
0CD9:049F 3ACA      CMP    CL,DL
0CD9:04A1 740B      JZ     04AB
0CD9:04A3 B43E      MOV    AH,3E
0CD9:04A5 CD21      INT    21
0CD9:04A7 8B0E1800     MOV     [0018],CL
0CD9:04AB 43        INC    BX
0CD9:04AC 3AEE      CMP    CH,DH
0CD9:04AE 740B      JZ     04BB
0CD9:04B0 B43E      MOV    AH,3E

```

-g:cs:491,494

```

AX=0006 BX=0014 CX=0000 DX=0101 SP=2F5D BP=0030 SI=0000 DI=2CD0
DS=0CD9 ES=7CB0 SS=7CB0 CS=0CD9 IP=0494 NV UP EI PL NZ NA PE CY

```

0CD9:0494 CB RETF

-t

AX=0006 BX=0014 CX=0000 DX=0101 SP=2F61 BP=0030 SI=0000 DI=2C00
DS=0CD9 ES=7CB0 SS=7CB0 CS=7CB0 IP=011E NV UP EI PL NZ NA PE CY

7CB0:011E 33ED	XOR	BP,BF
-u		
7CB0:011E 33ED	XOR	BP,BF
7CB0:0120 833E7309FF	CMP	WORD PTR [0973],-01
7CB0:0125 7505	JNZ	012C
7CB0:0127 0E	PUSH	CS
7CB0:0128 1F	POP	DS
7CB0:0129 E99D10	JMP	11C9
7CB0:012C FC	CLD	
7CB0:012D 8CCB	MOV	AX,CS
7CB0:012F BED0	MOV	SS,AX
7CB0:0131 BC632F	MOV	SP,2F63
7CB0:0134 BEC0	MOV	ES,AX
7CB0:0136 36	SS:	
7CB0:0137 BE1EA02C	MOV	DS,[2CA0]
7CB0:013B FB	STI	
7CB0:013C 36	SS:	
7CB0:013D C606062B80	MOV	BYTE PTR [2D06],80
-u		
7CB0:0142 36	SS:	
7CB0:0143 C6068928B0	MOV	BYTE PTR [2B89],80
7CB0:0148 0BED	OR	BP,BP
7CB0:014A 7409	JZ	0155
7CB0:014C 36	SS:	
7CB0:014D C706072B010D	MOV	WORD PTR [2B07],0D01
7CB0:0153 EB12	JMP	0167
7CB0:0155 803E710900	CMP	BYTE PTR [0971],00
7CB0:015A 740B	JZ	0167
7CB0:015C 1E	PUSH	DS
7CB0:015D 0E	PUSH	CS
7CB0:015E 1F	POP	DS
7CB0:015F BA0C2C	MOV	DX,2C0C
-u		
7CB0:0162 B43B	MOV	AH,3B
7CB0:0164 CD21	INT	21
7CB0:0166 1F	POP	DS
7CB0:0167 803E7B0A00	CMP	BYTE PTR [0A7B],00
7CB0:016C 740A	JZ	017B
7CB0:016E 803E770A00	CMP	BYTE PTR [0A77],00
7CB0:0173 7503	JNZ	017B
7CB0:0175 EBFA05	CALL	0772
7CB0:017B C6066B0900	MOV	BYTE PTR [096B],00
7CB0:017D C606710900	MOV	BYTE PTR [0971],00

```

-u
7CB0:0182 8CC8      MOV     AX,CS
7CB0:0184 8ED8      MOV     DS,AX
7CB0:0186 2B06A22C     SUB     AX,[2CA2]
7CB0:018A BA1000     MOV     DX,0010
7CB0:018D F7E2      MUL     DX
7CB0:018F 0BD2      OR      DX,DX
7CB0:0191 7403      JZ      0196
7CB0:0193 8BFFFF     MOV     AX,FFFF
7CB0:0196 A38D2C     MOV     [2CBD],AX
7CB0:0199 8E1EA02C     MOV     DS,[2CA0]
7CB0:019D F6066E09FF     TEST    BYTE PTR [096E],FF

```

```

-u
7CB0:01A2 7408      JZ      01AC
7CB0:01A4 EB6B05     CALL    0712
7CB0:01A7 7203      JB      01AC
7CB0:01A9 EB110A     CALL    0BBD
7CB0:01AC B419      MOV     AH,19
7CB0:01AE CD21      INT     21
7CB0:01B0 36        SS:
7CB0:01B1 A2B02C     MOV     [2CB0],AL
7CB0:01B4 F6066E09FF     TEST    BYTE PTR [096E],FF
7CB0:01B9 7408      JZ      01C3
7CB0:01BB EB5405     CALL    0712
7CB0:01BE 7203      JB      01C3
7CB0:01C0 EBE20C     CALL    0EA5

```

-q=cs:104,119

```

AX=FFFF BX=0014 CX=0000 DX=0101 SP=2F61 BP=0030 SI=0000 DI=2CD0
DS=0CD9 ES=7CB0 SS=7CB0 CS=7CB0 IP=0119  NV UP EI PL ZR NA PE NC
7CB0:0119 2E        CS:
7CB0:011A FF1E9E2C     CALL    FAR [2C9E]                CS:2C9E=0491

```

-t

```

AX=FFFF BX=0014 CX=0000 DX=0101 SP=2F5D BP=0030 SI=0000 DI=2CD0
DS=0CD9 ES=7CB0 SS=7CB0 CS=0CD9 IP=0491  NV UP EI PL ZR NA PE NC
0CD9:0491 EB0100     CALL    0495

```

```

-u
0CD9:0491 EB0100     CALL    0495
0CD9:0494 CB        RETF
0CD9:0495 33DB     XOR     BX,BX
0CD9:0497 8B0E6F09     MOV     CX,[096F]
0CD9:049B 8B161800     MOV     DX,[0018]
0CD9:049F 3ACA     CMP     CL,DL
0CD9:04A1 7408      JZ      04AB
0CD9:04A3 B43E     MOV     AH,3E
0CD9:04A5 CD21      INT     21
0CD9:04A7 8B0E1800     MOV     [0018],CL

```

: 关闭所打开的文件

```

0CD9:04AB 43      INC    BX
0CD9:04AC 3AEE    CMP    CH, DH
0CD9:04AE 7408    JZ     04B8
0CD9:04B0 843E    MOV    AH, 3E
-u
0CD9:04B2 CD21    INT    21
0CD9:04B4 8B2E1900  MOV    [0019], CH
0CD9:04B8 83C304    ADD    BX, +04
0CD9:04BB B90F00    MOV    CX, 000F
0CD9:04BE 843E    MOV    AH, 3E
0CD9:04C0 CD21    INT    21
0CD9:04C2 43      INC    BX
0CD9:04C3 E2F9    LOOP  04BE
0CD9:04C5 C3      RET
0CD9:04C6 1E    PUSH  DS
0CD9:04C7 53    PUSH  BX
0CD9:04C8 50    PUSH  AX
0CD9:04C9 B451    MOV    AH, 51
0CD9:04CB CD21    INT    21
0CD9:04CD 8EDB    MOV    DS, BX
0CD9:04CF A11800  MOV    AX, [0018]
-g=c5:491,4a7  执行此段后, 显示DOS提示符A>

```

: 关闭文件

B3

第二章 PC-DOS操作说明

本章和第九章是为初学者而设的，它介绍了一些最基本的操作方法和常用命令，是初学者上机操作的说明书。

2.1 系统启动

如果整个计算机系统尚未接通电源，可以通过冷启动的方式（即给系统加电）来启动系统。具体步骤是：将载有DOS 2.0（其他版本亦一样）的系统盘插入A驱动器，关上小门。然后依次接通打印机电源、显示器电源，最后打开主机电源。这时系统开始自检，若发现错误，则屏幕显示：

```
no system Disk (or Disk error)
```

系统盘错

或 Disk error reading (or writing) F A T

盘读写文件分配表错

若自检无误，则DOS进入内存，屏幕显示提示信息：

```
Current data is Tue 1-01-1980
```

```
Enter new data :
```

按mm dd yy或mm dd/yy格式打入当天日期，（其中mm表示月份，可以取1~12中任一数，dd表示日子，可以取1~31中任一数，yy表示年份，可以取80~99中任一数）或直接按回车键后，屏幕显示提示信息：

```
Current time is 0:01:26.94
```

```
Enter new time:
```

按hh:mm:ss.xx格式输入当前时间，（其中，hh表示小时，可以取0~23中任一数，mm表示分，可以取0~59中任一数，ss表示秒，可以取0~59中任一数，xx表示1/100秒，可以取0~99中任一数）或直接按回车键后，屏幕显示：

```
The IBM Personal Computer DOS
```

```
Version 2.0 (C) Copyright IBM Corp 1981, 1982, 1983
```

```
A
```

A 是PC-DOS的提示符：表示系统启动成功，可以接收输入的键盘命令。其中A为当前盘盘符，它可用选盘命令来改变，打入下列命令：

```
A B: <CR>
```

则提示符变为B，表示把当前盘改为B盘，其中下横线标出部分表示用户输入信息。

如果机器配有硬盘，也可以从硬盘启动系统。此时，不需要从A驱动器插入盘片，机器将会直接从硬盘上引导PC-DOS。而且，从硬盘上启动系统比从软盘上启动系统速度快，也比较方便。

2.2 软盘的初始化和复制

1. 软盘初始化

新盘在使用前必须初始化, 初始化的命令格式为:

```
FORMAT B: /S/V
```

其中 B: 表示要将 B 驱动器中的新盘初始化, /S 表示需要在初始化盘片后, 写上操作系统, 如果初始化的盘为数据盘, 则可省去此项。V 表示要写“卷名”。下面是一个初始化软盘的全过程。

```
A >FORMAT B: /S/V < CR >
```

```
Insert new diskette for drive B:
```

```
and strike any key when ready < CR >
```

```
Formatting ..... Format complete
```

```
System transferred
```

```
Volume label (11 characters, ENTER for none ) ? DataDisk1 < CR >
```

```
362496 bytes total disk space
```

```
40960 bytes used by system
```

```
321536 bytes available on disk
```

```
Format another (Y/N) ? N
```

随后, 用 CHKDSK 命令检查磁盘的状态以及内存情况。

```
A >CHKDSK B: < CR >
```

```
Volume MYDATADISK created May 22, 1984 1: 05p
```

```
362496 bytes total disk space
```

```
22528 bytes in 3 hidden files
```

```
18432 bytes in 1 user files
```

```
321536 bytes available on disk
```

```
131072 bytes total memory
```

```
106496 bytes free
```

在显示出的信息中, 第 2 行表示盘空间总共 360K (362496 bytes), 第 3 和第 4 行表示盘上已存在的三个隐含文件和一个普通文件的大小; 第 5 行表示盘上剩余空间为 320K, 最后两行给出内存总空间和可用空间的大小。

2. 软盘复制

为了防止工作盘被破坏, 用户可以用复制命令 DISKCOPY 复制一张备用盘, 操作过程如下:

在 PC DOS 状态下, 打入命令

```
A >DISKCOPY A: B: < CR >
```

屏幕显示

```
Insert source diskette in drive A:
```

```
Insert target diskette in drive B:
```

```
Strike any key when ready
```

按提示要求, 把源盘插入 A 驱动器, 把一张空盘 (可以是未初始化的新盘) 插入 B 驱动器, 然后任意按下一键, 如回车键, 则屏幕显示

```

Copying 9 sectors per track 2 side(s)
Formatting while coeyiny
Copy complete
Copy another ( Y/N ) ?

```

打入 Y, 则重复上过程, 继续复制, 打入 N, 则回到 A > 状态。

对于复制好的盘, 还可以使用全盘比较命令 DISKCOMP 与源盘进行比较, 检查一下复制得是否正确, 其过程如下:

```

A > DISKCOMP A: B: < CR >
Insert first diskette in drive A:
Insert second diskette in drive B:
Strike any key when ready < CR >
Comparing 9 sectors per track, 2 side (s)
Diskettes compare ok
Compare more diskettes ( Y/N ) ? N

```

应注意的是: DISKCOPY 命令是把一张软盘上的内容完完全全地复制到另一张软盘上; 当需要对个别文件进行复制时, 需用 COPY 命令。

2.3 键盘命令

键盘命令是人机对话的工具, 操作人员可以通过输入命令, 使 PC-DOS 完成某些特定的功能。PC-DOS 操作系统为用户提供了 5 种操作命令: 选盘命令、控制键、内部命令、外部命令、批处理命令。其具体内容如下:

(1) 选盘命令

```
< 盘符 > < CR >
```

其中盘符可以为 A:, B:, C:, D: 之一; 分别代表 A 盘、B 盘、C 盘、D 盘。

(2) 控制键

CTRL + ALT + DEL	系统复位 (热启动)
CTRL + C 或 CTRL + BREAK	结束当前操作
CTRL + P	联机打印
CTRL + S	暂停输出
SHIFT + PRTSC	屏幕 硬拷贝
< CR >	回车换行

(3) 内部命令

```

COPY CON: < 文件名 > 建立一个 ASCII 码文件
PROMPT [ < 字符串 > ] 修改系统提示符
DATE 置日期
TIME 置时间

```

(4) 外部命令

```

DISKCOPY [ < 盘符 > ] [ < 盘符 > ] 全盘复制
DISKCOMP [ < 盘符 > ] [ < 盘符 > ] 全盘比较

```

FORMAT [<盘符>] [/V] [/S] 初始化磁盘
CHKDSK [<盘符>] 检查磁盘状态
FDISK 硬盘分区

(5) 批处理

AUTOEXEC、BAT 自动启动批处理文件

另外，在PC DOS支持下，PC机还有如下功能键：

功 能	键盘键名	说 明
< COPY1 >	F 1	打一次COPY一个字符
< COPYUP >	F 2	打 F 2 后要指定 COPY 终止字符
< COPYALL >	F 3	打一次全部复制
< SKIP1 >	DEL	删去一个字符
< SKIPUP >	F 4	打 F 4 后要指定 COPY 起始字符
< VOID >	esc	取消当前输入
< INsert >	I NS	进入 / 退出插入方式
< RePlace >	F 6	退出插入或字符终止。(见 r 命令用例)
< Newline >	F 5	使当前行成为样板行。 或脱出命令态仍可接受编辑命令。

2.4 文件及文件操作命令

1. 文件和目录

文件是一组相关信息的集合，如程序、数据等都可以看成文件。文件根据其内容和形式的不同可以分为若干类，下面是常用的文件类型及其含义：

- . COM 可执行的二进制代码文件
- . EXE 可执行的浮动代码文件
- . BAT 可执行的批处理文件
- . SYS 系统文件
- . BAK 编辑程序的后备文件
- . OBJ 汇编语言或高级语言的目标码文件
- . LIB 库文件
- . MAP 目标程序模块全局量列表文件
- . ASC ASCII码文件
- . LST 源程序列表文件
- . PRG dBASE II程序文件

PC DOS的文件系统采用树型目录结构，如图2-1所示。即根目录A中有目录文件D，而D中又有O、P、Q三个文件。这样，对文件进行操作时，必须指明找到该文件的路径。通常的表示方式是：

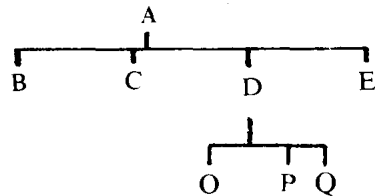


图 2-1

D [<盘符>] \ [<路径>] \ <文件名>

2. [<盘符>] \ <路径>

其中①表示指定某个文件，②表示指定某个目录。为方便起见，分别称为“文件路径名”(或“路径名”)和“目录路径名”。

2. 文件操作命令

①显示目录命令 DIR

它的功能是列出指定盘、指定目录或指定文件的目录，其三种格式分别如下

```
DIR [ <盘符> ] [ /P ] [ /W ]
```

```
DIR [ <目录路径名> ] [ /P ] [ /W ]
```

```
DIR [ <文件路径名> ] [ /P ] [ /W ]
```

其中 / P表示逐屏显示； / W表示多列显示。

②文件换名命令 REN或RENAME

该命令用来更改文件名，格式为：

```
REN <旧路径名文件> <新文件名>
```

或 RENAME <旧路径名文件> <新文件名>

其中，新文件名应由文件名加类型名组成。

③文件复制命令 COPY

该命令可以将一个文件复制到另一个盘上或同一个盘上。其格式为：

```
COPY <路径名> <文件名> [ <路径名> ] [ /A ] [ /B ] [ /V ]
```

其中， / A表示ASCII码文件； / B表示二进制文件； / V表示复制过程要进行校验。如果上式中第二个参数缺省，则表示复制到当前盘上。例如：

```
COPY B: ALL9P.EXE A:
```

表示把B盘中的AU9P·EXE复制到A盘上去。

```
COPY B: ALL9P·EXE A: TP2.EXE
```

表示把B盘上的ALL9P·EXE复制到A盘上，并改名为TP2·EXE。

```
COPY A: BASIC.COM
```

表示把A盘中的BASIC·COM文件复制到当前盘上。

COPY命令还可以将几个文件联结起来成为一个文件，其格式为：

```
COPY <路径名文件> [+ <路径名文件>...] [ <路径名文件> ] [ /A ] [ /B ] [ /V ]
```

④文件比较命令 COMP

该命令用来将一个文件的内容同另一个文件的内容进行比较，检查文件复制的正确性。被比较的两个文件可以在同一盘上，也可以不在同一盘上。命令格式为：

```
COMP <路径名> <路径名>
```

⑤系统复制命令 SYS

PC DOS的两个隐含文件IBMBIO.COM和IBMDOS.COM可在磁盘初始化时，由选择 S开关来复制，如当时未加 / S开关，事后可用该命令来复制，其命令格式为：

```
SYS <盘符>
```

盘符缺省表示当前盘。

⑥文件输出命令 TYPE

该命令可以将指定文件的内容输出到屏幕或打印机上。要输出到打印机上须事先按一个

CTRL-P键或CTRL-PRTSC键,接通打印机。其命令格式为:

TYPE <路径名> <文件名>

⑦文件删除命令DEL或ERASE

该命令用来删除某一个或某一组指定的文件,但不能删除子目录。其命令格式为:

DEL <路径名> \<文件名>

或 ERASE <路径名> \<文件名>

当删除一组文件时须加综合符*,如

DEL B: *·BAK

表示将B盘中所有的,BAK文件都删除。

⑧硬软盘间的复制命令BACKUP和RESTORE

硬、软盘间的数据交换,除COPY命令外,还可以使用BACKUP,命令把硬盘上的文件复制到软盘上,也可以用RESTORE命令把软盘上的文件复制到硬盘上去。其命令格式分别为:BACKUP <路径名> <文件名> <盘符> [/S] [/M] [/A]

RESTORE <盘符> <路径名> <文件名> [/S] [/P]

其中,/S开关表示复制包括子目录在内的所有文件;/M开关表示仅复制修改过的文件;/A开关表示把文件添加到备份盘中去,/P表示复制时给出提示。

第三章 CC-DOS的概述和初步分析

3.1 系统概述

CC-DOS汉字操作系统已被广泛地使用在国内的IBM PC微型计算机及其兼容机上。CC-DOS是在PC-DOS (IBM PC微型机的主操作系统)的基础上设计的,它具有PC-DOS的全部功能,且增加了汉字功能。在CC-DOS下,可继续使用PC-DOS支持的各种软件。CC-DOS的字库存放在外存上,系统启动时一次调入内存,加快了汉字处理的速度。它具有多种汉字输入方式。在输出方面也提供了多种选择。CC-DOS的使用不需要对IBM PC的原硬件系统进行改造,也不要增加新的硬设备,它是用软方案来实现PC-DOS的汉字化的。

在汉字处理方面,CC-DOS把汉字作为和西文一样的字符处理,汉字的容许度可以达到文件各级。在各种高级语言和应用程序中,汉字作为字符串,可以和西文混杂处理。CC-DOS的字库采用不压缩字模,每个字模32个字节,汉字点阵为 16×16 。其中CC-BIOS 2.0提供的字形为仿宋体,CC-DOS 2.0/2.1提供的字形为印刷体,它的字库中共有6763个汉字和616个图形符号。

CC-DOS向用户提供多种输入方式,它们都有与用户进行对话的功能。用户可以方便地实现各种输入方式间的转换(通过ALT键和相应的功能键进行)。CC-DOS的汉字显示和打印输出采用图形方式。显示的汉字字符为 16×16 点阵。打印时可选择16种字型,同时可选择打印的行宽。配上高级打印模块后,还可打印出 24×24 点阵的高质量汉字。

CC-DOS是对PC-DOS的扩充,扩充了汉字输入输出功能。一般来说,涉及到汉字输入输出的只是操作系统中直接控制外部设备的那一部分。在PC-DOS中,直接控制外部设备的那一部分被称为BIOS。由于它被固化在系统板上的ROM中,故亦被称为ROM-BIOS。要为PC-DOS扩补汉字功能,只要对ROM-BIOS进行一些修改和扩充即可。经过这样修改和扩充后,就形成了CC-BIOS,它是CC-DOS中直接控制外部设备的部分。因此,我们要对CC-DOS进行分析,只要对其中的CC-BIOS进行分析即可。由此可知,CC-BIOS是CC-DOS中的精华。

3.2 CC-DOS的自举原理

CC-DOS自举的关键是由ROM-BIOS形成CC-BIOS。

ROM-BIOS是PC-DOS中的最底层软件。高层软件对外部设备的调用,是通过调用ROM-BIOS中的相应模块来进行的。而调用ROM-BIOS的相应模块是通过调用相应的软中断(10H-1AH类)来完成的,软中断的响应,则是根据被调用的软中断的中断号,取中断向量表中的相应的中断指针,然后转去执行此中断指针指向处的程序。

CC-BIOS是CC-DOS的基本输入输出部分,由ROM-BIOS扩充而成的。它主要是扩充了汉字处理部分,这部分存在RAM中,(在CC-DOS自举时被引入)。CC-BIOS与ROM-BIOS一样,也是由若干个外部设备控制模块组成的。它们完成对系统部件中主要I/O设备的控制,使得高层软件(包括系统程序和用户程序)均不要直接与外部设备打交道,并使得系统硬件

的改进或扩充,对用户而言是“透明”的。对CC BIOS中的模块的调用,也是通过Intel 8088提供的软中断来实现的,其参数由指定的8088寄存器传递。所以,这些模块又被称为软中断处理程序(简称中断程序)。

由以上可知,要把ROM BIOS改为CC BIOS,只要把CC BIOS中的模块引入内存,然后修改相应的中断指针值,使其指向相应的CC BIOS的模块。以后通过软中断调用的就不再是ROM BIOS中的相应模块,而是CC BIOS中的相应模块。

CC DOS的自举过程与PC DOS是一样的,先是由ROM中的引导程序把系统盘上的系统引导程序引入内存。然后此程序运行,把系统盘上的三个文件:IBMBIO.COM,IBMDOS.COM和COMMAND.COM引到内存的指定区域。完成这些工作后就要检查系统盘上是否有AUTOEXEC.BAT文件存在,如果存在则执行这个批命令文件。CC DOS的系统盘上就存在这个批命令文件,CC DOS自举时,就是由这个文件来完成把ROM BIOS改为CC BIOS的。另外,它还完成引入汉字库的工作。这个文件中有两条主要的命令,即FILE 1和CCCC。前者所完成的工作是检查汉字库(CCLIB文件)在系统盘上的完好性,再为汉字库进入内存预先申请好内存空间。后者所完成的工作是把汉字装入到申请好的内存空间中(由FILE 1申请的),再修改有关的中断指针,使它们指向CC BIOS的相应模块。通过AUTOEXEC.BAT批命令文件的执行,就完成了CC DOS自举的关键,形成了CC BIOS。

经过上面的分析和介绍,我们可以知道,要弄清CC DOS是怎样完成汉字的输入输出功能的,只要对被修改了中断指针值的中断程序进行分析即可。这些中断是5H类、10H类、16H类和17H类。它们分别为CC BIOS的打印屏幕驱动模块、显示控制模块、键盘管理模块和打印驱动模块。

下面对CC DOS作一个初步的分析,主要分析其中实现汉字输入输出的部分。

3.3 CC DOS中汉字显示输出的实现

汉字的显示输出是由CC BIOS中的CRT控制程序即10类中断处理程序实现的。实现汉字显示输出的过程,实际上是把汉字的机内码转换成汉字字模,再把字模送屏幕输出的过程。

在CC DOS(V2.1)中,汉字的机内码用国标码最高位置1表示(称为异形国标码),为区别于一般的ASCII码,一个汉字的机内码用两字节表示。汉字字模存放在汉字库中,其存放位置也是按国标次序存放的。每个汉字字模用32个字节表示,即其点阵为 16×16 。这32个字节可看成16个字,每个字顺次表示汉字一行上16个点的情况。另外,CC DOS在显示汉字时必须工作在图形方式下,故CRT的字符发生器不发生作用,即使显示一般字符时,其字模亦要自己产生(其点阵为 8×8)。这些字符字模存放在字符字模库中,该库已由CCCC.EXE带入内存。

1. 有关的工作区

(1) 字模缓冲区——把获得的字模信息放到这里进行一定的排列,然后再送去显示。它分为左半区和右半区,汉字的显示需用到两个半区,字符的显示只用到一个半区。字模缓冲区的地址为CS:0078—0099。

(2) CRT刷新区——本区是存放屏幕显示信息的,它的每一个位(bit)与屏幕上的象元是一一对应的。改变了刷新区中某一位的内容,就改变了屏幕上相应象元的状态。其地址为B000:8000—9F3F(偶数扫描线区)和B000:A000—BF3F(奇数扫描线区)。

(3) 虚拟刷新区——其“虚”是针对CRT刷新区而言的。它是CRT屏幕所显示的字符

内容的副本。虚拟刷新区分为三个部分（分区），这三部分中的单元是一一对应的。这三个部分分别用以存放显示的字符、属性字和状态字。它们的地址分别为 CS: 00B0—087F（字符分区），CS: 0880—104F（属性字分区），CS: 1050—181F（状态字分区）。

2. CRT 控制程序

CRT 控制程序（即 10 类中断程序）由二十个功能块组成。在 AH 寄存器中置上功能块号（0—19），再使用 INT 10H 指令就可以调用相应的功能块。以下对其中最常用的 10 号功能块进行分析和介绍。它的具体功能是把字符送 CRT 屏幕显示。在显示的字符中不包括下列特殊字符：回车（0DH）、换行（0AH）、退格（08H）和报警（07H）。对这些特殊字符的处理不包含在本功能块中（由 14 号功能块实现对这些特殊字符的处理）。本功能块的执行流程如图 3-1 所示。

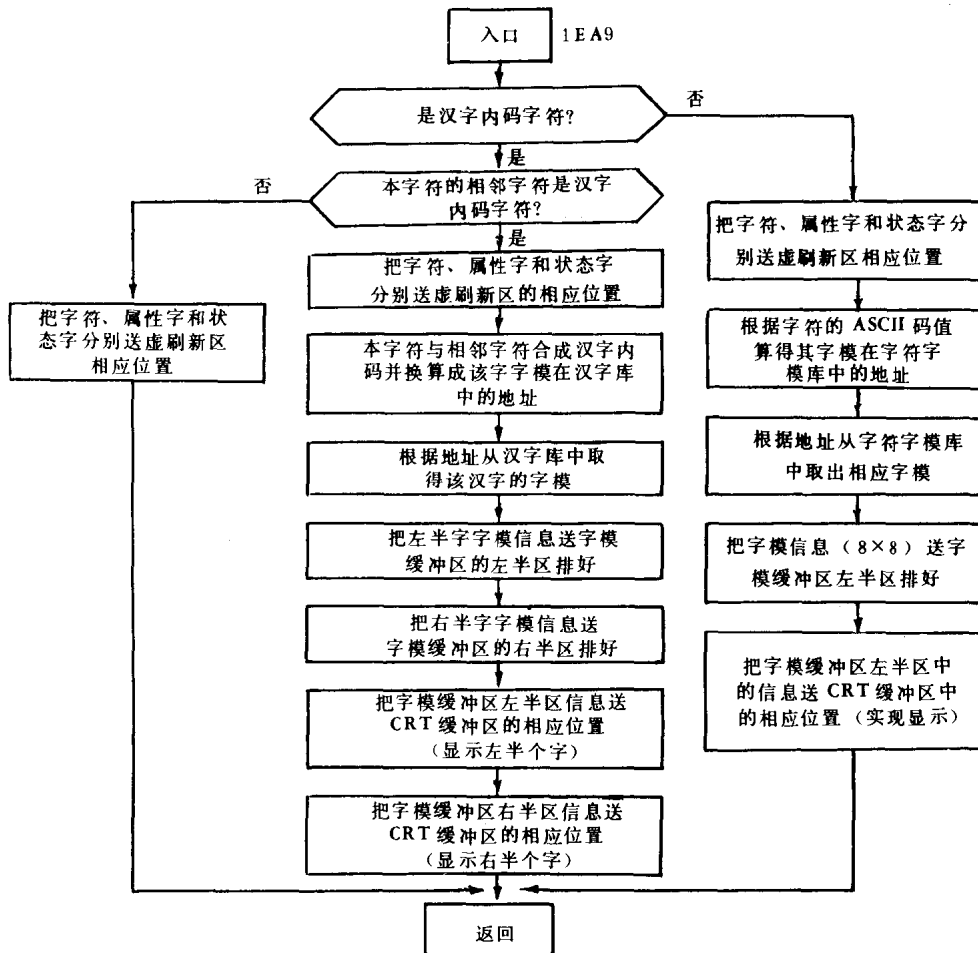


图 3-1 10 类中断 10 号功能块流程

下面对图 3-1 中的某些部分作几点说明。

- 1) 判断一个字符是否是汉字内码字符，即判其是否大于 A0H，大于则为汉字内码字符。
- 2) 字符字模库的首地址就是 1F 类中断指针的内容。这在 CCCC·EXE 执行时就置好了。
- 3) 判别一个字符的相邻字符是否是汉字内码字符，是通过判其相邻字符在虚拟刷新区中的状态字实现的，状态字的内容可以反映出这一点。
- 4) 汉字库的首地址被保存在指定单元中（OS:0075—0076）这是在 CCCC·EXE 执行

时保存的。

5) 把汉字字模信息送入字模缓冲区时, 为了保持屏幕每行间的间隔, 在最末要加送两行“空白”信息。把字符字模信息存入字模缓冲区时, 为了使字符与汉字保持一样的高度(字符高为8, 汉字高为16), 在其前与后要送几行“空白”信息。

6) 为了避免一个汉字的一半显示在一行末, 另一半显示在下一行首, 程序中设有行尾处理部分。当汉字显示的位置正好在一行末时, 则自动把其调整到下行首显示。

3.4 CC-DOS中汉字输入的实现

汉字的输入是由CC-BIOS中的键盘控制程序(即16类中断程序)来实现的。实现汉字输入的过程, 实际上是把由键盘输入的汉字输入码(如区位码、首尾码、拼音码等)转换成汉字机内码(如异形国际码)的过程。

与键盘有关的中断处理程序有两个。一个是9类中断程序, 当键盘上有字符输入时, 形成该字符的扫描码(关于扫描码请参阅IBM PC机的随机资料)和字符的ASCII码, 并把它们存入键盘缓冲区中。CC-BIOS中仍用原来ROM-BIOS中的9类中断程序。另一个是16类中断程序, 当要取键盘输入的字符时, 就调用它。它的主要功能是把键盘缓冲区中的字符送给调用者。CC-DOS对每个中断程序进行了较大的扩充, 使其能对输入的汉字输入码进行处理。

在CC-DOS中实现汉字输入码转换成机内码, 采用了两种方法。对有规律的输入码(如区位码)采用算法, 对另外一些输入码(如首尾码、拼音码等)采用查表法, 也就是借助于一张输入码—机内码对照表(简称扫描表)。这儿扫描表的每一个项对应于一个汉字, 表项的内容为其对应汉字的输入码。表项的排列是按国标字库顺序排的。当得到一个输入码后, 只要在扫描表中查找到内容符合输入码的项, 根据得到的项的序号, 经过简单的计算即可得到其对应的机内码。

1. 有关工作区

下面介绍几个与本程序有关的工作区

(1) 键盘缓冲区——存放由键盘上输入的字符(包括它的扫描码和ASCII码), 其中的内容由9类中断程序存放。地址为0040:001E—003D。

(2) 汉字输入码缓冲区——存放已输入的汉字输入码字符。地址为CS:2BB1—2BBA。

(3) 汉字内码缓冲区——存放由输入码转换成的汉字内码。地址为CS:959D—959E。

(4) 汉字重码缓冲区——当输入码所对应的汉字不止一个时, 则把这些汉字内码均存入此区(最多存10个内码)。地址为CS:2BBC—2BCF。

(5) 标志字节——它的不同值对应于当前不同的工作方式。要知当前工作方式是什么(如拼音码方式、首尾方式等), 只要判此字节的内容即可。地址为CS:963A。

2. 0号功能块流程

CC-BIOS中的16类中断程序分为七个功能块, 其调用法与10类中断相同。0号功能块是其中最主要的部分, 它的功能是从键盘缓冲区读出一字符, 并对其中的特殊字符(包括功能字符和汉字内码字符)进行处理, 并返回相应的信息; 对于一般字符, 则直接返回其扫描码和ASCII码。也就是这一功能块完成了汉字输入码到汉字机内码的转换工作。其流程如图3-2所示。

下面对图3-2作几点说明。

1) 所谓功能字符, 是指CTRL-F5至F10以及ALT-F1至F6。其中ALT-F1至F6又称为转换字符。

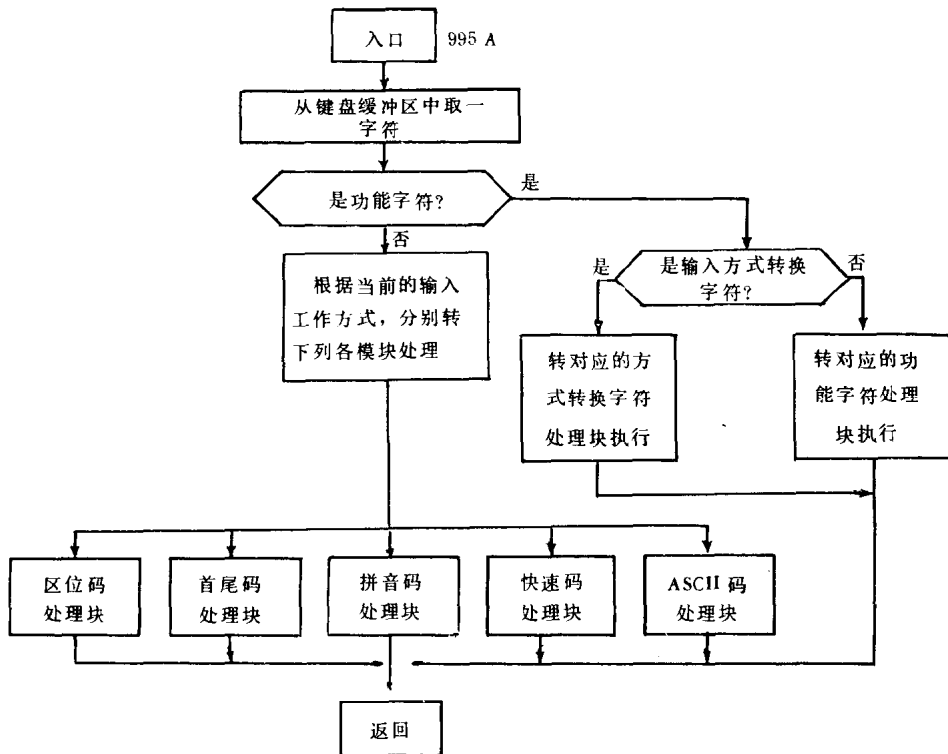


图 3 2 16类中断 0号功能块流程

2) CTRL - F 5至 F 10这六种字符所表示的功能, 请参阅第九章, 这里不再重复。它们的相应处理模块就是完成它们所表示的功能。

3) ALT F1至 F6是进行输入码方式转换, 它们的相应模块所做的工作, 就是在提示行(屏幕最底下一行)中显示当前工作方式的名字和把标志字节置成与之相应的值。

4)当前工作方式(即当前输入码方式)的判别, 是根据标志字节的值确定的。

5)各种输入码的处理模块就是完成把相应的输入码转换成机内码的工作。

3. 输入码的处理模块流程

下面介绍两种输入码的处理模块流程。一是区位码, 二是首尾码, 它们分别用计算法和查表法实现转换, 有一定的代表性。

(1) 区位码处理模块的工作流程(入口为 9 D C B)

1)对回车符(0 D H)和退格符(08 H)进行处理。

2)如是字符“0”—“9”, 则转下步执行, 否则直接返回。

3)把字符送入输入码缓冲区, 当此缓冲区中的字符满四个后, 即进行下一步, 否则返回。

4)把第一、二个字符转换成十进制 B C D码形式作为区号; 把第三、四个字符转换成十进制 B C D码形式作为位号。

5)把区号、位号转换成十六进制数(即形成国标码), 再转换成异形国标码(机内码), 然后返回。

(2) 首尾码的处理模块工作流程

本工作流程如图 3 - 3 所示。另外, 快速码、拼音码等的处理与此相似。

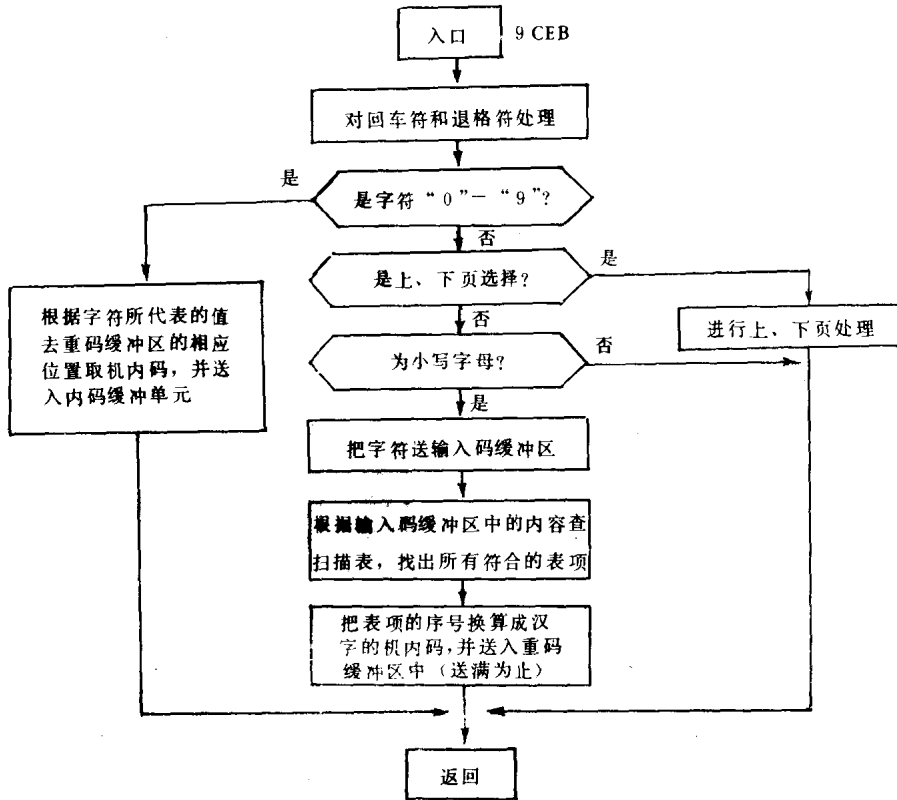


图 3 3 首尾码处理流程

下面对图 3-3 作一些说明。

1) 首尾码是三字符码, 它是有重码的。也就是说, 针对一个输入码有多个汉字与它对应。CC-DOS 对重码的处理是这样的, 把重码依次在提示行中显示出来, 让用户用数字键来选择其中的某个 (这是 CC-DOS 的特色之一)。所以字符“0”-“9”表示选择键, 根据其值, 可在重码缓冲区的相应位置上取到用户所选中汉字的机内码。

2) 上、下页选择是这样的: 提示行一次最多只能显示出十个重码汉字, 当重码汉字不止十个时, 则可用上、下页选择键来进行前后页的转换。对这种选择的处理, 就是把前十个或后十个重码汉字找出来送提示行显示, 并把它们的机内码存入重码缓冲区中。

3) CC-DOS 规定, 首尾码、快速码、拼音码等输入码的字符须用小写字母表示, 如是大写字母, 则表示非输入码字符, 作一般 ASCII 码处理, 这样便于西文、汉字输入转换。

4) 输入码字符顺次存放在输入码缓冲区中, 其个数可能为一个、两个或三个。在查表处理时, 要根据输入码缓冲区中字符的个数来查找表项中最前面的相应个字符是否与输入码缓冲区中的字符相同。

5) 把表项序号转换成机内码是很容易的, 因为表项的次序是按国标字库序列排的。

3.5 CC-DOS 中汉字打印输出的实现

用 DIR 命令, 可以看到在 CC-DOS 的系统盘上还有两个文件, 它们是 ALL9P·EXE 和 ALL24P·EXE。这两个文件是用来生成 CC-DOS 的 17 类中断处理程序 (打印机控制程

序)和5类中断处理程序(打印屏幕程序)的,这两个中断程序组成了CC BIOS的打印输出部分。CC DOS就是靠这部分来实现汉字的打印输出的。因为这部分程序与系统配置的打印机有直接关系,故CC DOS给出了两个文件,其中ALL9P·EXE用于九针打印机,ALL24P·EXE用于二十四针打印机。它们之所以没有被编入AUTOEXEC·BAT文件中,是为了让用户有选择打印机的余地。一旦选定了打印机,则可把其对应的文件编入AUTOEXEC·BAT文件中,以便在一开始就生成CC BIOS的打印输出部分。下面以ALL9P·EXE为例来介绍和分析CC DOS在九针打印机上输出汉字的实现方法。

1. 5类中断处理程序

用户调用打印机有两种目的,一种是调打印机打印文件或运算结果,另一种是调打印机打印当前屏幕上的显示内容。前者涉及到17类中断程序,后者涉及到5类中断程序。现先对5类中断程序进行分析。

CC BIOS的5类中断处理程序较简单。它的执行过程如下:(入口为CS:1FD3)

首先判CRT的工作方式,如是字符方式则直接去执行ROM-BIOS中原来的5类中断处理程序(入口为F00:F54),这段程序所做的工作就是把CRT刷新区中当前页的内容,逐字符送打印机输出;如果CRT的工作方式是图形方式,则向17类中断程序发送ESC“W”控制代码,由17类中断程序中的相应模块去完成把屏幕上的内容送打印机输出的功能。然后进行中断返回

2. 17类中断处理程序

CC BIOS的17类中断处理程序共有四个功能块组成,其中最主要的是0号功能块。它的功能是把字符送打印机输出。实现汉字打印输出的部分,就包含在这个功能块中。

(1) 有关的工作区

1) 打印数据缓冲区(简称缓冲区):用来存放要送打印机输出的打印数据,并使用它对字模点阵重新排列处理。在图形方式下(打印汉字就是在此方式下进行的)要用到这缓冲区。它共分成四个部分,即缓冲区1、缓冲区2、缓冲区3、缓冲区4。每区均为816个字节。下面介绍这四个区的作用。

缓冲区1(地址为CS:00E2—0741)和缓冲区2(地址为CS:0742—0DA1)用以存放要进行纵向扩展的汉字上半部的打印数据。

缓冲3(地址为CS:0DA2—1401)和缓冲区4(地址为CS:1402—1A61)用以存放要进行纵向扩展的汉字下半部的打印数据。对于不进行纵向扩展的汉字,缓冲区3存放汉字上半部的打印数据,缓冲区4存放汉字下半部的打印数据。对于一般字符,缓冲区4存放字符的打印数据

2) 属性工作区:存放要打印的字符的属性字(它表明该字符是一般图形打印,还是紧密图形打印);它共102个字节,每一个字节对应于缓冲区中的八个字节(因一个字符有八个打印数据),即指明这八个字节的打印属性。地址为CS:0016—00E1

3) 状态字:用于表示打印的几种状态,比如:是否紧密打印,是否横向扩展,是否纵向扩展,是否为横向字,前一字符是否是ESC,前两字符是否是ESC“1”等等。根据此状态字可以判断出,当前字符是否是控制代码,以及要打印的汉字为何种字型(CC DOS V2.1提供十六种字型)。它的地址为CC:000F--0010

(2) 0号功能块的流程

0号功能块的工作流程如图3·4所示。

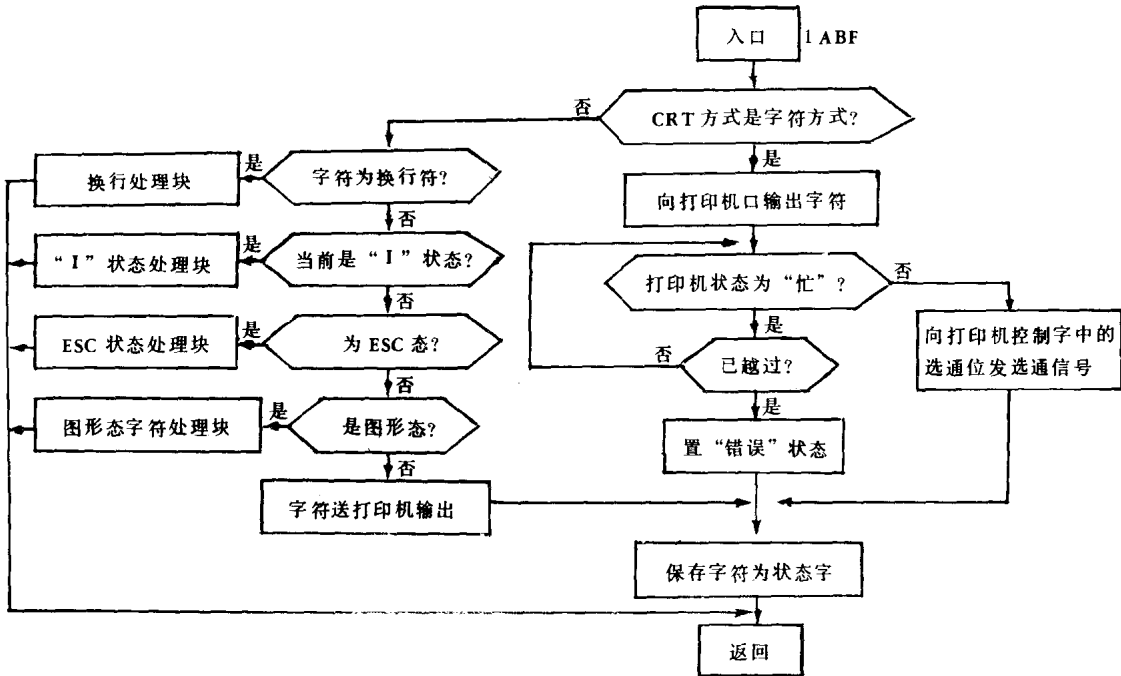


图 3 4 17类中断 0号功能块流程

下面对图3-4中的有关部分作些说明。

1) 向打印机输出字符时，没有采用中断方式，而是采用查询方式。故要先从打印机状态口取状态字，如打印机“忙”，则等待，否则向其控制口发选通信号，从而完成输出一个字符。

2) 换行处理块完成以下工作：判别缓冲区是否为空，若为空，则进行行间距为8/72英寸的换行（8/72英寸正好为半个汉字的高）；否则把缓冲区中的内容全部送打印机输出。

3) “I”状态处理块的功能是改变汉字的字型。在16类中断程序中，向用户提供了一个改变打印的汉字字型的功能，实际上就是由16类中断程序向17类中断程序发出一个ESC“ I”控制代码。后跟一个字母（A—P）代表字型的选择（共十六种）。I状态处理块就是对ESC“ I”控制代码进行解释，它根据选择的字型，分别对状态字中的有关位置赋以相应的值。

4) ESC状态处理块的功能是对一些控制代码进行处理。这里主要处理由5类中断程序发来的ESC“W”控制代码（详见前面5类中断程序部分），即实现把屏幕上的内容在打印机上打印出来。本处理块在判到有ESC“W”控制码时，就把CRT刷新区中的内容转换成打印数据送打印机输出。

5) 图形态字符处理块的功能是把字符或汉字的字模信息转换成打印数据。前面已说过，字库中的字模信息是以行表示的，即每个字节代表字符横向的点的情况，这适合于CRT显示。而打印机正好相反，它要求字模信息以列表示，即每个字节代表字符纵向的点的情况（这样的字节中的数称为打印数据）。所以，必须把字模的横向信息转换成纵向信息，才能向打印机输出。本处理块的工作流程如图3-5所示。

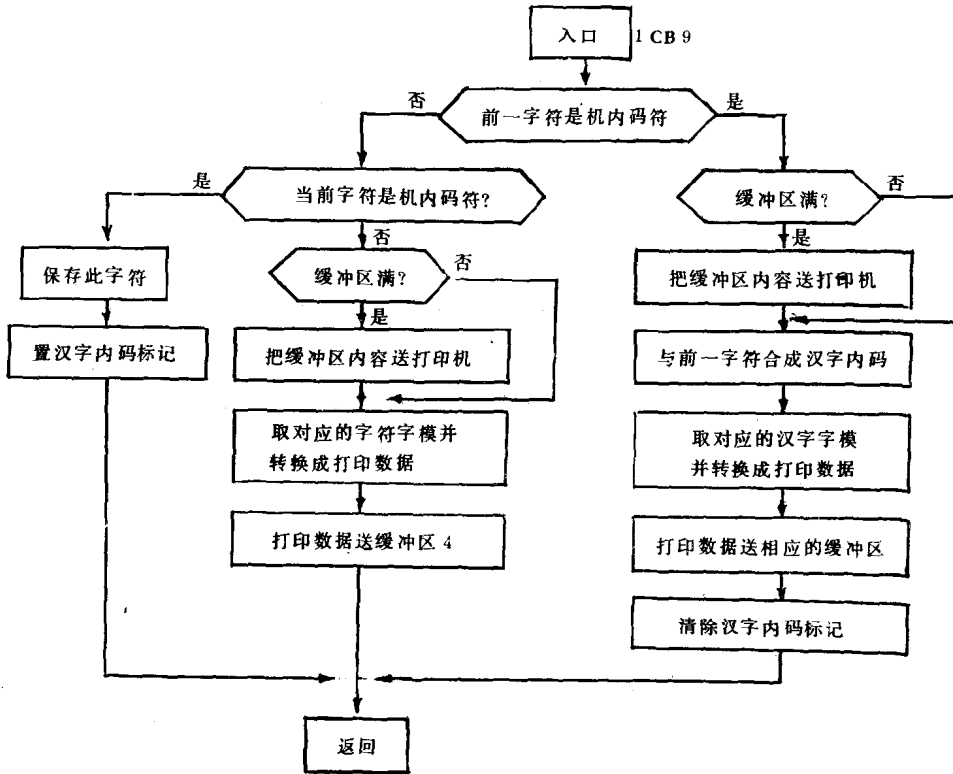


图 3-5 图形态的字符处理流程

现对图 3-5 中的内容作一些说明。

(i) 判断前一字符是否是汉字内码字符，只要判断汉字内码标记即可。

(ii) 判断缓冲区是否满，并不是判断缓冲区是否被全部装满，而是判断缓冲区是否能再装下当前要放入的全部打印数据（如字符、一般汉字和横向扩展汉字的打印数据个数是不一样的），如果不能全部装入，即使缓冲区有空余字节，也算缓冲区满。只有这样才能确保打印输出的内容的正确和完整。

(iii) 把汉字的打印数据送相应缓冲区要做以下工作：根据指定的字型（由状态字内容可知），对打印数据进行横扩、纵扩、转向等处理。然后，当是纵向扩展时，则把汉字的上半部打印数据送缓冲区 1 和 2，把汉字的下半部打印数据送缓冲区 3 和 4。当是非纵向扩展时，则把汉字上、下半部的打印数据分别送缓冲区 3 和 4。

以上只是对 CC-DOS 操作系统进行了一个初步的分析，其目的旨在使大家对它有一定的认识，为下面对 CC-BIOS 的深入分析作准备。

以下各章中所附源程序的段值，为随机数全用 DEBUG 打出。

第四章 CC-BIOS 的引导过程

4.1 概述

CC-DOS 汉字功能的实现,一般只与直接驱动和控制外部设备的这部分程序有关。IBM PC 的系统板上有一个 8 K 的 ROM, 里面装的是系统外部设备的控制程序, 被称为 ROM-BIOS。它是 IBM PC 系统软件的最底层, 其他软件中如要调用外部设备, 要通过调用 ROM-BIOS 中的相应模块来实现。若使外部设备能输入和输出汉字, 只要对 ROM-BIOS 进行修改, 把它改为 CC-BIOS (CC-DOS 中的 BIOS)。那么, 怎样把 ROM-BIOS 改成 CC BIOS 呢? 这个修改的过程, 就是 CC-BIOS 的引导过程。

4.2 BIOS 的结构与修改

为了讨论怎样修改 BIOS, 首先简单地介绍一下 BIOS 的结构。

BIOS 是由若干个独立的设备驱动模块组成的, 每个模块对应于一种设备, 即为该种设备的控制 (驱动) 程序。每个模块的入口地址被存放在系统的中断向量表中 (地址为 0000:0000—03FF)。每个地址均由段值和偏移量组成, 占四个字节。这四个字节的内容被称为中断向量。对 BIOS 中的模块的调用, 是通过软中断 (10 类—1 A 型) 来实现的, 故这些模块亦被称为软中断处理程序 (简称中断程序)。

要改变 BIOS 中某模块的内容, 我们采用“贴补代替”的办法。即把该模块在中断向量表中的相应中断向量改为新模块的所在地址即可。经这样修改后, 当用相应的软中断来调用该模块时, 就不再执行原来模块中的程序, 而转到新的模块之入口执行。但是, 新的模块应放在什么地方呢? 在 CC-DOS 中, 新模块是放在 RAM 中的。也就是说, CC-BIOS 的程序分布在 ROM 和 RAM 中。以上就是改 ROM-BIOS 为 CC-BIOS 的基本方法。IBM PC 系统向我们提供了采用这种方法来修改 BIOS 的可能性, 因 IBM PC 的内存容量较大, 故新的 BIOS 模块占用一部分内存, 关系不大; 另外 IBM PC 系统提供了使程序常驻内存的系统功能调用, 使得新的 BIOS 模块在被引导进内存后, 可以常驻内存而取代相应的 ROM 中的部分。

另外, 在把 ROM-BIOS 改为 CC-BIOS 的过程中, 并不要修改原来所有的模块, 而只要修改与汉字输入输出有关的模块。这些模块如表 4-1 所示。

与输入输出有关的 BIOS 模块

表 4-1

中断类号	实 现 功 能	中断类号	实 现 功 能
5	打印屏幕内容驱动程序	16	键盘管理程序
10	CRT 控制程序	17	打印机驱动程序

4.3 CC-DOS 的自举过程

CC-BIOS 的引导过程是包含在 CC-DOS 的自举过程中的, 让我们先来看一下 CC-DOS 的自举过程。

当系统启动或复位时，则自动执行ROM中的引导程序，该程序把CC-DOS系统盘上0道1扇区中的引导记录读入内存，并使它开始执行。引导记录对系统盘进行一些正确性的检查，然后把系统盘上的IBM BIO，COM文件和IBM DOS·COM文件读到内存的指定区域。

首先进入内存的是IBM BIO·SYS，它先进行初始化处理，然后向下重新定位IBM DOS·SYS，并把控制权交给它。IBM DOS·SYS对有关的工作表、缓冲区和向量等进行初始化处理，并为COMMAND·COM建立程序段前缀，然后返回IBM BIO·SYS。IBM BIO·SYS把COMMAND·COM装入确定的内存位置，然后把控制权交给COMMAND·COM。

COMMAND·COM查找系统盘上是否有CONFIG·SYS文件存在，如存在则执行之。然后执行AUTOEXEC·BAT文件（CC-DOS系统盘上一定存在此文件），最后显示操作系统的提示符，等待用户输入命令。CC-DOS的整个自举过程到此结束。图4-1给出了整个自举过程的流程。

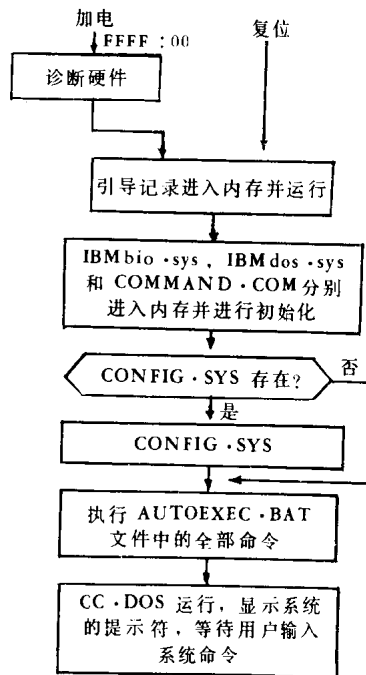


图 4-1 CC-DOS 自举流程

4.4 CC-BIOS的引导过程

让我们先了解一下CC-DOS系统盘的目录内容，其目录清单如下：

```

A>
A> dir/w
Volume in drive A has no label
Directory of A:\
DOMMAND COM FILE1 EXE CCCC EXE CCLIB BROTHE16 EXE
9 P EXE AUTOEXEC BAT CDEBUG CON FONT BAS
          9 File(s)      2048 bytes free
A>
  
```

CC-DOS 汉字库占237K，5 $\frac{1}{4}$ 盘一张所剩空间不多，此盘附有二种打印机驱动程序，调试工具和打印字型测试程序文本如下：

```

10 REM CHINESE FONT TEST
20 LPRINT CHR$(27)+CHR$(73)+CHR$(67); "汉字打印16字型"
30 FOR I = 0 TO 15
40 A = 65 + I
50 LPRINT CHR$(27)+CHR$(73)+CHR$(A); CHR$(A); "PM";
  
```

```

60 PRINT "汉字系统"
70 LPRINT "汉字系统:"THE INSTITUTE OF COMPUTING TECHNOLOGY"
80 LPRINT: LPRINT
90 NEXT I
100 LPRINT CHR$(27)+CHR$(73)+CHR$(67)
110 END

```

从上面的清单中我们可以看到，系统盘上存在一个AUTOEXEC·BAT文件，CC-DOS自举时要执行这个文件。AUTOEXEC·BAT文件为批命令文件，它的内容为若干条命令或可执行文件名，当该文件被执行时，就把这些命令或可执行文件，逐个递交给系统执行。CC-BIOS的引导，就是利用这个文件来进行的。这个文件中存放了引导CC-BIOS的命令，其具体内容如下所示：

```

echo off
cls
echo PLEASE WAIT
file1
cccc
verify on

```

在这批命令中，前三条命令所完成的工作就是在屏幕的左上角显示“PLEASE WAIT”字样。最后一条命令是开放校验。值得注意的是file1和cccc两条命令。从上面给出的系统盘目录清单中我们可以知道，这两条命令的对应文件分别为FILE1·EXE和CCCC·EXE。下面我们对这两个文件进行分析，分析他们的结构和它们在CC-BIOS的引导过程中所完成的功能。

4.5 对FILE1·EXE的分析

FILE1·EXE的功能是做好装入汉字库之前的准备工作。具体完成了下列工作：检查汉字库（即CCLIB文件）在系统盘上的完好性，为汉字库申请好内存。其执行流程如图4-2所示。

从图4-2中可以看到，当汉字库（CCLIB）不完好时，则要准备再次引导系统。实际上是这样来完成的：先显示再次引导的提示信息，然后等待用户从键盘输入信息，一旦用户输入一字符（任意字符），则通过软中断19来调用ROM-BIOS中的系统引导程序，执行图4-1的流程，重新引导系统。

从图4-2中还可以看到，当本命令结束时，它并没有正常退出（即交还占用的资源），而是驻留内存退出运行。也就是说，它没有交还占用的内存资源（包括结束前申请的那部分），这部分内存就不会被系统分配出去，留着以后装入汉字库。同时，此内存块的起始地址已被保存在1F类中断指针（0000:007C—007F）中了，当调入汉字

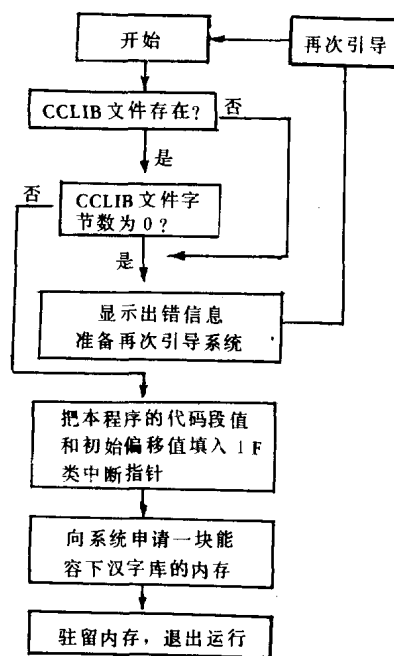


图 4-2 FILE1·EXE的流程

库时要用到此地址。另外，本命令的程序虽然驻留内存（用31H系统功能调用实现），但以后不再用到它，在汉字库装入时，它将被覆盖掉。

本文第八节中给出了FILE 1.EXE的源程序，读者可参照图4-2阅读之。

4.6 对CCCC·EXE的分析

CCCC·EXE所完成的工作主要是，把汉字库装入内存，把ROM-BIOS改为CC-BIOS。这一步是CC-BIOS引导过程中最重要的一步。那么，CC-BIOS的程序是从那里装入RAM的呢？这些CC-BIOS的程序模块就编写在CCCC·EXE文件中。在CCCC·EXE被系统调入内存执行时，它们亦被一起带入了内存。不过，在CCCC·EXE执行时，它们并没有被执行。CCCC·EXE本身的程序代码执行完后就驻留内存，退出运行。这样，这些CC-BIOS的程序模块也就驻留内存了。表4-2给出了CCCC·EXE文件的具体结构。

CCCC·EXE文件的结构

表4-2

段内相对地址	内 容	段内相对地址	内 容
0000—1847	工作区和数据区	9596—98B2	工作区和数据区
1848—2797	CC-BIOS的10类中断程序	98B3—AA76	CC-BIOS的16类中断程序
2798—2B97	字符字模库	AA77—AABF	工作区和数据区
2B98—2BD4	工作区和数据区	AAC0—AB43	CCCC·EXE本身的执行代码
2BD5—9995	汉字输入码-机内码对照表		

由表4-2可知，CCCC·EXE的执行代码入口地址为AAC0，执行到AB43为止。CCCC·EXE的执行流程如图4-3所示。

在图3-3中，先把汉字库装入内存（用3FH系统功能调用实现）和保存1F类中断指针，这实际上是把汉字库的起始地址保存好，供以后访问汉字库时使用。接着对有关的中断指针进行了修改，这就是用CCCC·EXE中的CC-BIOS程序模块去替换ROM-BIOS中的相应模块（用25H系统功能调用实现），从而基本上形成了CC-BIOS。这里要指出的是，其中的1D类和1F类指针指向的并不是中断程序。1D类指针指向的是CRT初始化参数，在对CRT进行初始化时要用到它。1F类指针指向的是字符字模库，在图形方式下输出字符时要用到它。另外，10类和16类指针分别指向CC-BIOS的CRT控制模块和键盘管理模块。图4-3中所谓显示CC-BIOS的标题，就是在CC-BIOS引导完成时，在屏幕上显示CC-BIOS的版本号、研制单位名称和日期等信息。最后让程序驻留内存（用27H中断来实现），使CC-BIOS基本形成。

本文第八节中给出了CCCC·EXE本身执行代码的

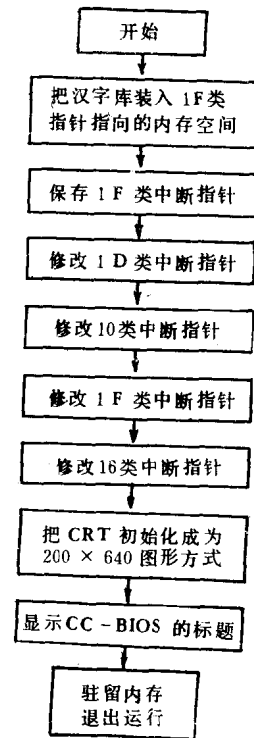


图4-3 CCCC·EXE的流程

源程序，读者可参阅。

4.7 CC-BIOS 打印模块的形成

CCCC·EXE 执行完后，CC-BIOS 就已基本形成。但是，对照表 4-1，大家可以发现，与汉字输入输出有关的 5 类中断指针和 17 类中断指针还没有被修改，即 CC-BIOS 中的打印模块还没有形成。下面就来分析这部分形成过程。

在第四节里列出的系统盘目录清单中，我们可以找到 ALL 9P·EXE 文件和 ALL 24P·EXE 文件。这两个文件是用于形成 CC-BIOS 的打印模块（5 类中断程序和 17 类中断程序）的。按理讲，应把它们也编入 AUTOEXEC·BAT 文件中。但是，因用户配用的打印机有不同的类型，一般有 9 针和 24 针两种。不同类型的打印机要配不同的打印模块，故系统给出了 ALL 9P·EXE 文件和 ALL 24P·EXE 文件。它们分别对应于 9 针和 24 针打印机。用户选定打印机后，应把相应的文件名插入 AUTOEXEC·BAT 文件中去。

ALL 9P·EXE 文条的结构 表 4-3

段内相对地址	内 容
0000—0015	17类中断程序的工作区
0016—1A61	17类中断程序的缓冲区
1A62—1FD2	CC-BIOS的17类中断程序
1FD3—200F	CC-BIOS的5类中断程序
2010—2037	17类中断程序的一个子程序
2038—2074	ALL 9P·EXE本身的执行代码

下面以 ALL 9P·EXE 为例，来分析 CC-BIOS 中 9 针打印机的打印模块的形成过程（24 针打印机的打印模块的形成与此类似）。先介绍 ALL 9P·EXE 文件的结构。它与 CCCC·EXE 一样，前面部分是 CC-BIOS 的有关模块，后面部分为其本身的执行代码。表 4-3 给出了它的具体结构。

由表 4-3 可知，ALL 9P·EXE 本身的执行代码入口地址为 2038，它执行到 2074 为止。它的前面部分仅是被带入内存，而并没有被执行。图 4-4 给出了 ALL 9P·EXE 本身执行代码的执行流程。

图 4-4 中一开始就保存了 1F 类中断指针，这实际上是把字符字模库的起始地址保存起来。它与后面保存的汉字库起始地址一样，在图形方式下打印字符和汉字时都要用到它们。接着修改了 5 类和 17 类中断指针，这就形成了 CC-BIOS 的打印模块。

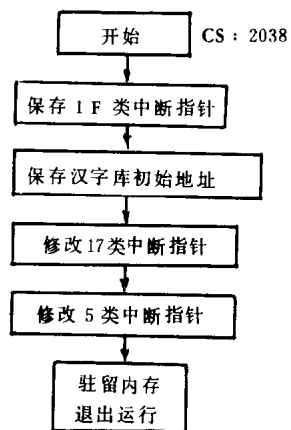


图 4-4 ALL 9P·EXE 的流程

ALL9P·EXE 的源程序(9 针打印, A~P 16 种字型)

A>debug all9p.exe

-r

AX=0000 BX=0000 CX=2080 DX=0000 SP=0000 BP=0000 SI=0000 DI=0000

DS=4DF8 ES=4DF8 SS=4E08 CS=4E08 IP=2038 NV UP DI PL NZ NA PO NC

4E08:2038 0E PUSH CS

-u

4E08:2038 0E PUSH CS

4E08:2039 1F POP DS

4E08:203A 8B1F35 MOV AX,351F ; 取 1F 中断向量放 ES:BX

4E08:203D CD21 INT 21

4E08:203F 891E0300 MOV [0003],BX ; 存 1FH 向量单元

4E08:2043 8CC3 MOV BX,ES

4E08:2045 891E0100 MOV [0001],BX

4E08:2049 8B1035 MOV AX,3510 ; 改 10H 向量

4E08:204C CD21 INT 21

4E08:204E 8B7500 MOV BX,0075 ; CCLIB 字模库段址

4E08:2051 26 ES:

4E08:2052 8B07 MOV AX,[BX]

4E08:2054 2E CS: ; 16×16 字模起始段值→CS:2010

4E08:2055 A31020 MOV [2010],AX

-u

4E08:2058 8D16621A LEA DX,[1A62] ; 改 17H 向量到 1A62

4E08:205C 8B1725 MOV AX,2517

4E08:205F CD21 INT 21

4E08:2061 8D16D31F LEA DX,[1FD3] ; 改 05H 向量到 1FD3

4E08:2065 8B0525 MOV AX,2505

4E08:2068 CD21 INT 21

4E08:206A 8D163B20 LEA DX,[203B] ; 执行指令起点

4E08:206E 81C20301 ADD DX,0103

4E08:2072 CD27 INT 27 ; 加余量, 驻留, 退出

4E08:2074 0000 ADD [BX+SI],AL

4E08:2076 0000 ADD [BX+SI],AL

-q

NEW9P·EXE 的源程序(9 针打印: A, B, I, J 4 种小字型)

C>debug new9p.exe

-r

AX=0000 BX=0000 CX=2100 DX=0000 SP=0000 BP=0000 SI=0000 DI=0000

DS=4DF8 ES=4DF8 SS=4E08 CS=4E08 IP=2081 NV UP DI PL NZ NA PO NC

4E08:2081 0E PUSH CS

-u

4E08:2081 0E PUSH CS

4E08:2082 1F POP DS

4E08:2083 8B1F35 MOV AX,351F ; 取 1F 中断向量放 ES:BX

4E08:2086 CD21 INT 21

4E08:2088 891E0300 MOV [0003],BX

4E08:208C 8CC3 MOV BX,ES

4E08:208E 891E0100 MOV [0001],BX

```

4E08:2092 B81035      MOV     AX,3510      ; 取10 H 中断向量放 ES :BX
4E08:2095 CD21        INT     21
4E08:2097 BB7500      MOV     BX,0075
4E08:209A 26          ES:
4E08:209B BB07        MOV     AX,[BX]
4E08:209D 2E          CS:
4E08:209E A35920      MOV     [2059],AX    ; 转存 CCLIB 字模库段址
-t
4E08:20A1 8E1A631A    LEA     DX,[1A63]
4E08:20A5 B81725      MOV     AX,2517      ; 改17 H 中断到 CS :1A 63
4E08:20A8 CD21        INT     21
4E08:20AA 8D161C20    LEA     DX,[201C]
4E08:20AE B80525      MOV     AX,2505      ; 改05 中断到 CS :201 C
4E08:20B1 CD21        INT     21
4E08:20B3 BFC070      MOV     SI,20CC
4E08:20B6 BF7F20      MOV     DI,207F
4E08:20B9 FD          CLD
4E08:20BA B90900      MOV     CX,0009
4E08:20BD 0E          PUSH   CS
4E08:20BE 0E          PUSH   CS
4E08:20BF 07          POP    ES
4E08:20C0 1F          POP    DS
-t
4E08:20C1 F3          REPZ
4E08:20C2 A4          MOVSB
4E08:20C3 B81200      MOV     DX,2081      ; 执行指令起点
4E08:20C6 81C20301    ADD     DX,0103      ; 加点余量. 驻留. 退出
4E08:20CA CD27        INT     27
4E08:20CC 03065920    ADD     AX,[2059]
4E08:20D0 89C2      MOV     DX,AX
4E08:20D2 58          POP    AX
4E08:20D3 C3          RET
4E08:20D4 0000      ADD     [BX+SI],AL
4E08:20D6 0000      ADD     [BX+SI],AL
4E08:20D8 0000      ADD     [BX+SI],AL
4E08:20DA 0000      ADD     [BX+SI],AL
4E08:20DC 0000      ADD     [BX+SI],AL
4E08:20DE 0000      ADD     [BX+SI],AL
4E08:20E0 0000      ADD     [BX+SI],AL

```

-q

C:\debug a:9p.exe

-r

```

AX=0000 BX=0000 CX=2100 DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=4DF8 ES=4DF8 SS=4E08 CS=4E08 IP=2081  NV UP DI PL NZ NA PO NC
4E08:2081 0E          PUSH   CS

```

-u

```

4E08:2081 0E          PUSH   CS

```

```

4E08:2082 1F      POP      DS
4E08:2083 B81F35   MOV      AX,351F      ; 取 1 FH 向量放 ES :BX
4E08:2086 CD21   INT      21
4E08:2088 B91E0300  MOV      [0003],BX    ; 存 1 FH 向量
4E08:208C BCC3     MOV      BX,ES
4E08:208E B91E0100  MOV      [0001],BX
4E08:2092 B81035   MOV      AX,3510      ; 改 10 H 向量
4E08:2095 CD21   INT      21
4E08:2097 BB7500   MOV      BX,0075
4E08:209A 26     ES:
4E08:209B 8B07   MOV      AX,[BX]
4E08:209D 2E     CS:
4E08:209E A35920   MOV      [2059],AX    ; 存 CCLIB 字模库段址
-u
4E08:20A1 BD16631A  LEA      DX,[1A63]
4E08:20A5 B81725   MOV      AX,2517      ; 改 17 H 向量到 CS :1 A 63
4E08:20A8 CD21   INT      21
4E08:20AA BD161C20  LEA      DX,[201C]    ; 改 05 H 向量到 CS :201C
4E08:20AE B80525   MOV      AX,2505
4E08:20B1 CD21   INT      21
4E08:20B3 BD16B120  LEA      DX,[20B1]    ; 执行指令起址 加点余量, 驻留, 退出
4E08:20B7 B1C20301  ADD      DX,0103
4E08:20BB CD27   INT      27
4E08:20BD 0000   ADD      [BX+SI],AL
4E08:20BF 0000   ADD      [BX+SI],AL
-g 209e

```

```

AX=0622 BX=0075 CX=2100 DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=4E08 ES=404A SS=4E08 CS=4E08 IP=209E NV UP DI PL NZ NA PD NC
4E08:209E A35920      MOV      [2059],AX      DS:2059=0000

```

```

-d es:0075
404A:0075 22 06 36-00 00 00 00 00 00 00 00 00 00 00 00 00 00 16x16字模库段址在 404 A :0075
404A:0080 00 00 00 00 00 00 00 00 00-00 00 00 00 00 C0 60 30 30 ...0!.....e'00
404A:0090 30 30 30 30 30 30 60 C0-00 00 00 00 00 00 00 00 00 000000'e.....
404A:00A0 00 00 00 00 00 00 0E 17 1F-17 00 00 00 00 00 FF FF 00 .....C.....
404A:00B0 34 45 30 3B 3A 32 30 39-45 20 41 33 35 39 32 30 4E08:209E A35920
404A:00C0 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
404A:00D0 44 5B 2C 5B 31 41 36 33-5D 20 20 20 20 20 20 20 DX,[1A63]
404A:00E0 20 20 20 20 20 20 20 20-20 20 20 20 20 20 20 20 25行字符区地址在 404 A :00 B0
404A:00F0 20 20 20 00 00 ..

```

```

-d cs:00
4E08:0000 00 4A 40 9B 27 00 00 00-00 00 00 80 02 00 00 02 此处是 CCCC .exe 工作区
4E08:0010 00 00 00 00 00 00 00 FF-FF FF FF FF FF FF FF FF .....
4E08:0020 FF FF FF FF FF FF FF FF-FF FF FF FF FF FF FF FF .....
4E08:0030 FF FF FF FF FF FF FF FF-FF FF FF FF FF FF FF FF .....
4E08:0040 FF FF FF FF FF FF FF FF-FF FF FF FF FF FF FF FF .....
4E08:0050 FF FF FF FF FF FF FF FF-FF FF FF FF FF FF FF FF .....
4E08:0060 FF FF FF FF FF FF FF FF-FF FF FF FF FF FF FF FF .....
4E08:0070 FF FF FF FF FF FF FF FF-FF FF FF FF FF FF FF FF .....

```

FILE1·EXE 的源程序

C>debug a:file1.exe

-r

AX=0000 BX=0000 CX=0100 DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
 DS=4DF8 ES=4DF8 SS=4E08 CS=4E08 IP=0000 NV UP DI PL NZ NA PD NC
 4E08:0000 EB07 JNP 0009

-d

4E08:0000 EB 07 90 63 63 6C 69 62-00 1E 0E 1F 8D 16 03 00 k..ccliib.....
 4E08:0010 BB 00 3D CD 21 72 73 8B-D8 88 02 42 CD 21 72 6A B.=M!rs.X8.BM!rj
 4E08:0020 F6 C2 0F 75 05 3D 00 00-74 60 52 50 8D 16 00 00 vB.u.=.t RP....
 4E08:0030 BB 1F 25 CD 21 B4 3E CD-21 58 5A B9 0C 00 D3 E2 8.XM!4>M!XZ9..Sb
 4E08:0040 B9 04 00 D3 E8 03 D0 80-C2 20 80 C6 10 1F 88 03 9..Sh.P.B .F..B.
 4E08:0050 31 CD 21 07 07 46 49 4C-45 20 22 43 4C 49 42 22 1M!..FILE "CLIB"
 4E08:0060 20 45 52 52 4F 52 0A 0D-4B 49 54 20 41 20 41 4E ERROR..HIT A AN
 4E08:0070 59 20 4B 45 59 20 54 4F-20 42 4F 4F 54 20 44 4F Y KEY TO BOOT DO

-d

4E08:0080 53 20 41 47 41 49 4E 20-21 24 BD 16 53 00 B4 09 S AGAIN !\$.S.4.
 4E08:0090 CD 21 B4 07 CD 21 CD 19-00 00 00 00 00 00 00 00 M!4.M!M.....
 4E08:00A0 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00
 4E08:00B0 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00
 4E08:00C0 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00
 4E08:00D0 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00
 4E08:00E0 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00
 4E08:00F0 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00

-dds:00 file.exe 命令文件段前缀 (PSP)

4DF8:0000 CD 20 00 80 00 9A F0 FF-0D F0 42 02 F3 4A 70 02 Mp..pB.sJp.
 4DF8:0010 F3 4A E2 04 42 05 F3 4A-01 01 01 00 02 FF FF FF sJb.B.sJ.....
 4DF8:0020 FF FF FF FF FF FF FF FF-FF FF FF FF F5 4D CB 2AuMH*
 4DF8:0030 F3 4A 00 00 00 00 00 00-00 00 00 00 00 00 00 sJ.....
 4DF8:0040 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00
 4DF8:0050 CD 21 CB 00 00 00 00 00-00 00 00 00 20 20 20 M!K.....
 4DF8:0060 20 20 20 20 20 20 20 20-00 00 00 00 20 20 20
 4DF8:0070 20 20 20 20 20 20 20 20-00 00 00 00 00 00 00

-d

4DF8:0080 00 0D 61 3A 66 69 6C 65-31 2E 65 7B 65 0D 0D 41 ..a:file1.exe..A
 4DF8:0090 49 54 49 4E 47 0D 00 00-00 00 00 00 00 00 00 ITING.....
 4DF8:00A0 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00
 4DF8:00B0 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00
 4DF8:00C0 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00
 4DF8:00D0 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00
 4DF8:00E0 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00
 4DF8:00F0 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00

-t

AX=0000 BX=0000 CX=0100 DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
 DS=4DF8 ES=4DF8 SS=4E08 CS=4E08 IP=0009 NV UP DI PL NZ NA PD NC
 4E08:0009 1E PUSH DS


```

-u
4E08:0009 1E      PUSH    DS          ; 改 DS= CS
4E08:000A 0E      PUSH    CS          ; 取文件名 CCLIB
4E08:000B 1F      FOP     DS          ; 打开汉字库文件供读入内存
4E08:000C 8D160300     LEA    DX,[0003]
4E08:0010 B8003D      MOV    AX,3D00
4E08:0013 CD21      INT    21
4E08:0015 7273      JB     00BA
4E08:0017 8BD8      MOV    BX,AX        ; FCB → BX 移文件读写指针
4E08:0019 B80242      MOV    AX,4202
4E08:001C CD21      INT    21
4E08:001E 726A      JB     00BA
4E08:0020 F6C20F      TEST   DL,0F        ; 测字模库文件尺寸
4E08:0023 7505      JNZ    002A
4E08:0025 3D0000     CMP    AX,0000      ; 若长度= 0 转出错处理
4E08:002B 7460      JZ     00BA

-u
4E08:002A 52      PUSH    DX
4E08:002B 50      PUSH    AX
4E08:002C 8D160000     LEA    DX,[0000]
4E08:0030 B81F25      MOV    AX,251F      ; 修改 1 FH 向量到 DS :DX
4E08:0033 CD21      INT    21
4E08:0035 B43E      MOV    AH,3E        ; 关文件
4E08:0037 CD21      INT    21
4E08:0039 5B      POP     AX          ; 恢复 CCLIB 库文件尺寸
4E08:003A 5A      POP     DX
4E08:003B B90C00     MOV    CX,000C      ; 计算 CCLIB 字库所需空间
4E08:003E D3E2      SHL    DX,CL
4E08:0040 B90400     MOV    CX,0004
4E08:0043 D3EB      SHR    AX,CL
4E08:0045 03D0     ADD    DX,AX
4E08:0047 80C220     ADD    DL,20

-u
4E08:004A B0C610     ADD    DH,10
4E08:004D 1F      POP     DS          ; 恢复数据段值
4E08:004E B80331     MOV    AX,3103      ; 为 CCLIB 申请内存 起段址在 DX
4E08:0051 CD21      INT    21          ; AL= 03 为退出码
4E08:0053 07      POP     ES          ; 出错信息字符码
4E08:0054 07      POP     ES
4E08:0055 46      INC    SI          F
4E08:0056 49      DEC    CX          I
4E08:0057 4C      DEC    SP          L
4E08:0058 45      INC    BP          E
4E08:0059 2022     AND    [BP+SI],AH  "
4E08:005B 43      INC    BX          C
4E08:005C 4C      DEC    SP          L
4E08:005D 49      DEC    CX          I
4E08:005E 42      INC    DX          B
4E08:005F 2220     AND    AH,[BX+SI]  "

```

```

4E08:0061 45      INC     BP
4E08:0062 52      PUSH   DX      R
4E08:0063 52      PUSH   DX      R
4E08:0064 4F      DEC    DI      O
4E08:0065 52      PUSH   DX      R
4E08:0066 0A0D   OR     CL,[DI]
4E08:0068 4B      DEC    AX      H
4E08:0069 49      DEC    CX      I

```

-dcs:68

```

4E08:0068 4B 49 54 20 41 20 41 4E      HIT A AN
4E08:0070 59 20 4B 45 59 20 54 4F-20 42 4F 4F 54 20 44 4F  Y KEY TO BOOT DO
4E08:0080 53 20 41 47 41 49 4E 20-21 24 8D 16 53 00 B4 09  S AGAIN !$.S.4.
4E08:0090 CD 21 B4 07 CD 21 CD 19-00 00 00 00 00 00 00  M!4.M!M.....
4E08:00A0 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00  .....
4E08:00B0 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00  .....
4E08:00C0 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00  .....
4E08:00D0 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00  .....
4E08:00E0 00 00 00 00 00 00 00 00  .....

```

-u cs:8a

```

4E08:008A 8D165300  LEA    DX,[0053]      ; 出错信息 FILE " CLIB " error 显示
4E08:008E B409      MOV    AH,09
4E08:0090 CD21      INT    21
4E08:0092 B407      MOV    AH,07          ; 等键
4E08:0094 CD21      INT    21
4E08:0096 CD19      INT    19             ; 重新启动
4E08:0098 0000      ADD    [BX+SI],AL
4E08:009A 0000      ADD    [BX+SI],AL
4E08:009C 0000      ADD    [BX+SI],AL
4E08:009E 0000      ADD    [BX+SI],AL
4E08:00A0 0000      ADD    [BX+SI],AL
4E08:00A2 0000      ADD    [BX+SI],AL
4E08:00A4 0000      ADD    [BX+SI],AL
4E08:00A6 0000      ADD    [BX+SI],AL
4E08:00A8 0000      ADD    [BX+SI],AL

```

-q

CCCC · EXE 的源程序

C>debug a:cccc.exe

-r

```

AX=0000 BX=0000 CX=AB44 DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=4DFB ES=4DFB SS=4E08 CS=4E08 IP=AAC0 NV UP DI PL NZ NA PD NC
4E08:AAC0 1E      PUSH   DS

```

-dcs:00

显示器管理数据区，工作区

```

4E08:0000 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00  .....
4E08:0010 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00  .....
4E08:0020 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00  .....
4E08:0030 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00  0 页, 1 页, 2 页, 3, 4, 5, 6, 7 页光标
4E08:0040 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00  .....
4E08:0050 01 00 01 00 01 00 01 00-01 00 01 00 01 00 01  .....

```

4E0B:0060	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00		75 存放 16 × 16 字模库段地址
4E0B:0070	00 00 00 00 00 00 00 36-00 00 00 00 00 00 00		78 汉字模加工区
-d			
4E0B:00B0	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00	
4E0B:0090	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00	
4E0B:00A0	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00	
4E0B:00B0	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00	
4E0B:00C0	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00	
4E0B:00D0	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00	
4E0B:00E0	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00	
4E0B:00F0	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00	
-u			
4E0B:AAC0	1E	PUSH	DS
4E0B:AAC1	33C0	XOR	AX,AX
4E0B:AAC3	50	PUSH	AX
4E0B:AAC4	0E	PUSH	CS
4E0B:AAC5	1F	POP	DS
4E0B:AAC6	8D1679AA	LEA	DX,[AA79]
4E0B:AACA	BB003D	MOV	AX,3D00
4E0B:AACD	CD21	INT	21
4E0B:AACF	7303	JNB	AA04
4E0B:AAD1	EB70	JMP	AB43
4E0B:AAD3	90	NOP	
4E0B:AAD4	50	PUSH	AX
4E0B:AAD5	BB1F35	MOV	AX,351F
4E0B:AADB	CD21	INT	21
4E0B:AADA	8CC2	MOV	DX,E5
4E0B:AADC	89167500	MOV	[0075],DX
-u			
4E0B:AAE0	BEDA	MOV	DS,DX
4E0B:AAE2	8B03	MOV	DX,BX
4E0B:AAE4	88	POP	BX
4E0B:AAE5	8900B0	MOV	CX,8000
4E0B:AAE8	B43F	MOV	AH,3F
4E0B:AAEA	CD21	INT	21
4E0B:AAEC	7255	JB	AB43
4E0B:AAEE	3D0000	CMP	AX,0000
4E0B:AAF1	740C	JZ	AAFF
4E0B:AAF3	8CDA	MOV	DX,DS
4E0B:AAF5	81C20008	ADD	DX,0800
4E0B:AAF9	BEDA	MOV	DS,DX
4E0B:AAFB	33D2	XOR	DX,DX
4E0B:AAFD	EBE6	JMP	AAE5
4E0B:AAFF	0E	PUSH	CS
-u			
4E0B:AB00	1F	POP	DS
4E0B:AB01	B43E	MOV	AH,3E
4E0B:AB03	CD21	INT	21

: 保存 CCLIB 文件控制块

: 汉字字模库首址段码送 CS :0075 单元
(汉字字模库起始段码单元)保存

: CCLIB 文件从 FCB 弹出到 BX

: 若读完, 转 AAFF

```

4E08:AB05 8D168818 LEA DX,[1888] ; 修改 1DH 类中断指针, 指向 CS:1888
4E08:AB09 B81D25 MOV AX,251D ; 初始化数据表
4E08:AB0C CD21 INT 21
4E08:AB0E 8D164818 LEA DX,[1848] ; 修改 10H 类中断指针, 指向 CS:1848
4E08:AB12 B81025 MOV AX,2510 ; CCBIOS 显示器管理模块
4E08:AB15 CD21 INT 21
4E08:AB17 8D169827 LEA DX,[2798]
4E08:AB1B B81F25 MOV AX,251F
4E08:AB1E CD21 INT 21

-u
4E08:AB20 8D16B398 LEA DX,[98B3] ; 修改 16H 类中断指针, 指向 CS:98B3
4E08:AB24 B81625 MOV AX,2516
4E08:AB27 CD21 INT 21
4E08:AB29 B80600 MOV AX,0006 ; 将显示器设置成 640x200 图形方式
4E08:AB2C CD10 INT 10
4E08:AB2E 8D167FAA LEA DX,[AA7F] ; 取 CCBIOS 标题首址送 BX
4E08:AB32 B409 MOV AH,09
4E08:AB34 CD21 INT 21
4E08:AB36 8D1679AA LEA DX,[AA79]
4E08:AB3A 81C20301 ADD DX,0103
4E08:AB3E 90 NOP
4E08:AB3F 58 POP AX ; 恢复 AX

-u
4E08:AB40 1F POP DS
4E08:AB41 CD27 INT 27 ; 驻留内存
4E08:AB43 CB RETF
4E08:AB44 0000 ADD [BX+SI],AL
4E08:AB46 0000 ADD [BX+SI],AL
4E08:AB48 0000 ADD [BX+SI],AL
4E08:AB4A 0000 ADD [BX+SI],AL
4E08:AB4C 0000 ADD [BX+SI],AL
4E08:AB4E 0000 ADD [BX+SI],AL
4E08:AB50 0000 ADD [BX+SI],AL
4E08:AB52 0000 ADD [BX+SI],AL
4E08:AB54 0000 ADD [BX+SI],AL
4E08:AB56 0000 ADD [BX+SI],AL
4E08:AB58 0000 ADD [BX+SI],AL
4E08:AB5A 0000 ADD [BX+SI],AL
4E08:AB5C 0000 ADD [BX+SI],AL
4E08:AB5E 0000 ADD [BX+SI],AL

-d 622:00 CCLIB 段址值因机型变
0622:0000 00 00 00 00 00 00 00 00-00 00 00 00 07 F0 0C C0 .....p.e
0622:0010 0C C0 07 80 0C 00 0F 80-1C E0 18 60 0C C0 07 80 .e.....e..
0622:0020 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
0622:0030 00 00 00 00 30 00 1E 00-0F 00 07 00 00 00 00 ....0.....
0622:0040 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
0622:0050 00 00 00 00 3C 00 66 00-66 00 66 00 3C 00 00 ....<.f.f.f.<...
0622:0060 00 00 00 00 00 00 00 00-00 00 00 00 01 80 03 C0 .....e
0622:0070 03 C0 01 80 00 00 00 00-00 00 00 00 00 00 00 .....e

```

第五章 CC-BIOS的键盘管理模块

5.1 引言

CC-BIOS的键盘管理模块就是键盘驱动和控制程序,亦称键盘输入模块。键盘管理模块由两部分组成,即键盘中断处理程序(9类中断程序)和键盘输入请求程序(16类中断程序)。其中9类中断程序仍用原来ROM-BIOS中的9类中断程序,而16类中断程序则比原来ROM-BIOS中的16类中断程序有较大的改动和扩充,主要是增加了对汉字代码的处理。也就是说,在这部分中实现了汉字输入功能。因此,我们将重点对CC-BIOS的16类中断程序进行分析。

5.2 键盘中断的处理

为了要对键盘输入请求程序(16类中断程序)进行分析,首先要介绍一下与之有关的键盘中断处理程序(9类中断程序)。它的主要功能是对键盘发来的中断请求进行处理。

IBM PC的键盘是一个独立体,它带有一个Intel 8048单片机。用8048实现下列功能:在系统需要时进行自检;检测重键;进行键盘扫描;实现扫描码的缓冲;维持与主机的双向串行通讯。8048在扫描键盘时所产生的编码是扫描码,这是IBM PC系统的特点之一。对IBM PC键盘上的每个键,根据其所处的位置,系统给它规定了一个序号,这个序号就是该键所对应的扫描码。

9类中断程序实质上就是一个键盘硬中断处理程序,当键盘上有字符输入时,就产生一个硬中断(即可屏蔽中断)。IBM PC系统中的可屏蔽中断共有八级(0—7级),键盘中断是其中的第1级。当发生键盘中断时(即键盘有信息输入),就转9类中断程序执行。9类中断程序所完成的工作是,从可编程I/O口8255A的Port A接受8048产生的扫描码信息,然后把它转换成相应的ASCII码;再把输入字符的扫描码和ASCII码都存入键盘数据缓冲区(该缓冲区由本章第三节介绍)中。每个字符在键盘数据缓冲区中占一个字(16位),高8位为其扫描码,低8位的内容是这样的:当该字符为ASCII码时,则为其ASCII码值,否则就为全0。键盘数据缓冲区中的内容是供16类中断程序读取使用的。实际上,9类中断程序与16类中断程序是通过键盘数据缓冲区和有关状态单元来进行信息传递的。

5.3 键盘输入请求程序的结构

键盘输入请求程序(即16类中断程序)的功能是对系统程序和用户程序要求从键盘输入字符的请求进行处理。它是CC-BIOS键盘管理模块的主体,它具有对汉字代码进行处理的功能。我们先来分析一下它的结构情况。

16类中断程序由功能块和工作区组成,其中共有七个功能块(块号为0—6),每块均是独立的,用以完成某个规定的功能。具体情况如表5-1所示。

调用16类中断程序中的某个功能块,只要把功能块号置入AH寄存器内,然后使用指令INT 16H,即可实现调用。这些功能块中最重要的是0号功能块,它的功能是从键盘数据缓冲区中读出字符,把字符返回给调用者。CC-DOS中的汉字输入处理部分就包含在这一功能块中,由此可知此功能块的重要性。

功能块号	完成的工作内容
0	从键盘数据缓冲区中取一个字到 AX 中
1	判键盘数据缓冲区中是否有有效信息
2	从 0040:0017 单元取当前换码状态到 AL 中
3	把 BP:DX 指向处的 27072 个单元之内容送到从 CS:2BD6 起始的单元中 (装扫描表)
4	把 BP, CX 和 DX 的内容分别送至 CS:2B98~2B99 和 2BA0~2BA3 中
5	0 号子块: 把 BP, DX 之内容送至 CS:9461~9644 中
	1 号子块: 把 CS:9641~9644 之内容送至 BP, DX 中
	2 号子块: 把 CS:9645~9948 之内容送至 DI 指向处的单元中
	3 号子块: 把 BP, DX 之内容送至 CS:9645~9648 中
6	把 CS:2BD5 之内容送至 CS:963A 中

注: 本章中的 CS 均为 CC-BIOS 中 16 类中断程序的代码段值。

16 类中断程序使用了不少工作区和工作单元, 下面介绍其中主要的几个。

1. 键盘数据缓冲区

本缓冲区存放从键盘上输入的字符 (包括它的扫描码和 ASCII 码), 其中的内容由 9 类中断程序放入。地址为 0040:001E~002D。当该缓冲区存满后, 如还要存入信息时, 则会产生报警。

2. 汉字输入码缓冲区

存放已输入的汉字输入码字符。地址为 CS:2BB1~2BBA。

3. 汉字输入码缓冲区计数器

记录当前汉字输入码 (外码) 缓冲区存放的输入码字符个数 (外码计数)。地址为 CS:959C。

4. 汉字机内码缓冲区

存放由汉字输入码转换成的汉字机内码字符。当有重码时, 其中存放的是选中的那个汉字机内码之字符。地址为 CS:959D~959E。

5. 汉字机内码缓冲区计数器

记录汉字机内码缓冲区中当前已存放的机内码字符个数。本计数器的其他用处在本章第五节中介绍。地址为 CS:95D5。

6. 汉字重码缓冲区

当汉字输入码所对应的汉字不止一个时 (有重码), 则把这些汉字的机内码存入此缓冲区中 (最多存十个), 以供选择。地址为 CS:2BBC~2BCF。

7. 汉字重码缓冲区计数器

记录汉字重码缓冲区中存入的机内码的个数。地址为 CS:2BD2。

8. 工作方式标志字节

记录当前的工作方式, 它的每一位对应于一种工作方式, 其对应情况如表 4-2 所示。地址为 CS:963A。

9. 键盘数据缓冲区信息的首指针和尾指针

它们分别指向键盘数据缓冲区内有效信息的首和尾。当两指针值相同时, 表示键盘数据缓冲区为空。它们的地址分别为 0040:001A~001B 和 0040:001C~001D。

10. 输入码完成状态字节

记录输入码的完成与否。完成态表示输入码的字符已全部输入，否则为非完成态。地址为CS:959B。

11. 扫描表

它的情况由本章第四节介绍。地址为CS:2BD6~9595。

12. 扫描表指针

指向扫描表的当前表项之首址（仅包括段内偏移量），在查找扫描表时要用。地址为CS:2BAD。

工作方式标志字节的内容 表 5-2

位号	位置代表的方式
0	区位码输入方式
1	首尾码输入方式
2	拼音码输入方式
3	快速码输入方式
4	ASCII码方式

5.4 代码转换和扫描表

实现汉字输入的过程，实际上是把汉字输入码转换成汉字机内码的过程。实现这种转换，一般有两种方法——计算法和查表法。在CC-DOS(V2.1)中，对有计算规则可循的输入码（如区位码）采用计算法来转换成机内码。对其他一些输入码（如拼音码首尾码和快速码）则采用查表法来转换成机内码。查表法的具体实现，与所采用的“输入码-机内码对照表”（简称扫描表）有直接关系。下面介绍一下CC-DOS(V2.1)中的扫描表。

CC-DOS(V2.1)中用查表法来进行转换的输入码有拼音码、首尾码和快速码，其中快速码是首尾码的快速实现，其本质也是首尾码，所以我们只要讨论拼音码和首尾码即可。一般说来，有一种输入码，就要有一张与其对应的扫描表。但是，在CC-DOS(V2.1)中，拼音码和首尾码合用一张扫描表。这是因为对于同一个汉字，其首尾码的最后一个字符（即此汉字读音的首字符）与其拼音码的第一个字符相同，所以可把这两种输入码的扫描表合为一张，以节省内存开销。扫描表由表项组成，每个表项对应于一个汉字，每个表项的内容即为其对应汉字的输入码。表项的序号是按其对应的汉字在字库中的次序定的，又因为CC-DOS(V2.1)中的机内码采用的是异形国标码（国标码之最高位置“1”），所以由表项的序号很容易算得它所相应汉字的国标码和机内码。

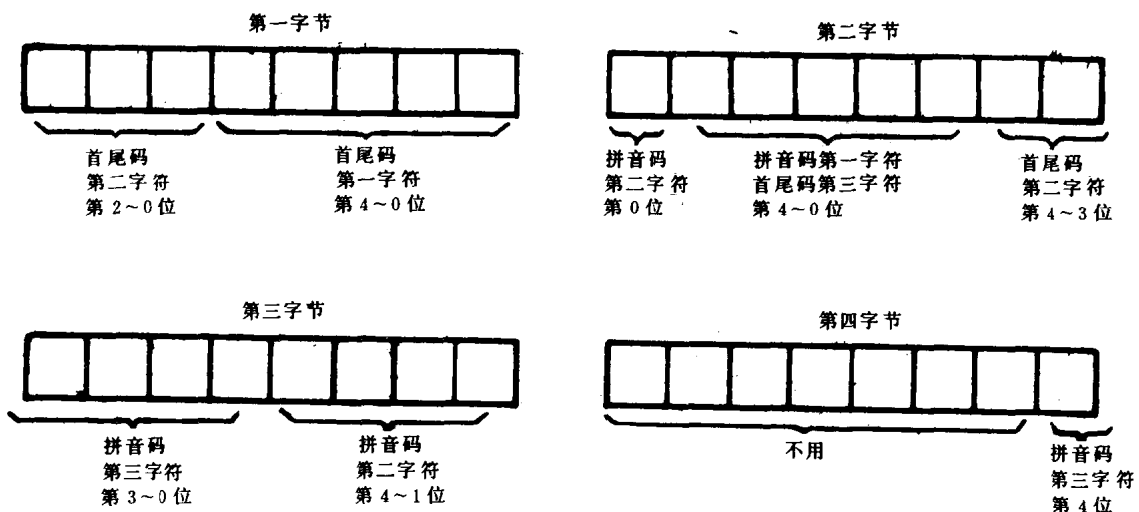


图 5-1 扫描表的表项结构

CC-DOS (V2.1) 的扫描表项要包含它所对应之汉字的首尾码和拼音码的内容。因这两种码均为三字符码, 并且它们中有一个字符是公用的, 所以每个表项应含五个字符的内容。在 CC-DOS (V2.1) 中, 规定这两种输入码的组成字符均为小写英文字母 (共二十六种), 故只要用 5 个位 (bit) 就可表示一个输入码字符 (5 个位可以表示三十二种, 而现在只有二十六种, 故足够了), 用 00001~11010 依次表示小写字母 a-z。因此每个表项要 25 个位来表示五个输入码字符。也就是说, 每个表项要用四个字节 (其中有一个字节仅用 1 个位)。其结构如图 5-1 所示。

5.5 汉字输入功能的实现

CC-DOS 的汉字输入功能是在 CC-BIOS 中 16 类中断程序的 0 号功能块内实现的。0 号功能块所做的主要工作是: 从键盘数据缓冲区中读输入字符所对应的内容, 对其中的功能字符进行处理, 对其中的汉字输入码进行转换, 并返回相应信息, 对其他字符则不处理, 而直接返回其扫描码和 ASCII 码, 从而完成了从键盘输入字符的功能。图 5-2 给出了这个功能块的全部流程。

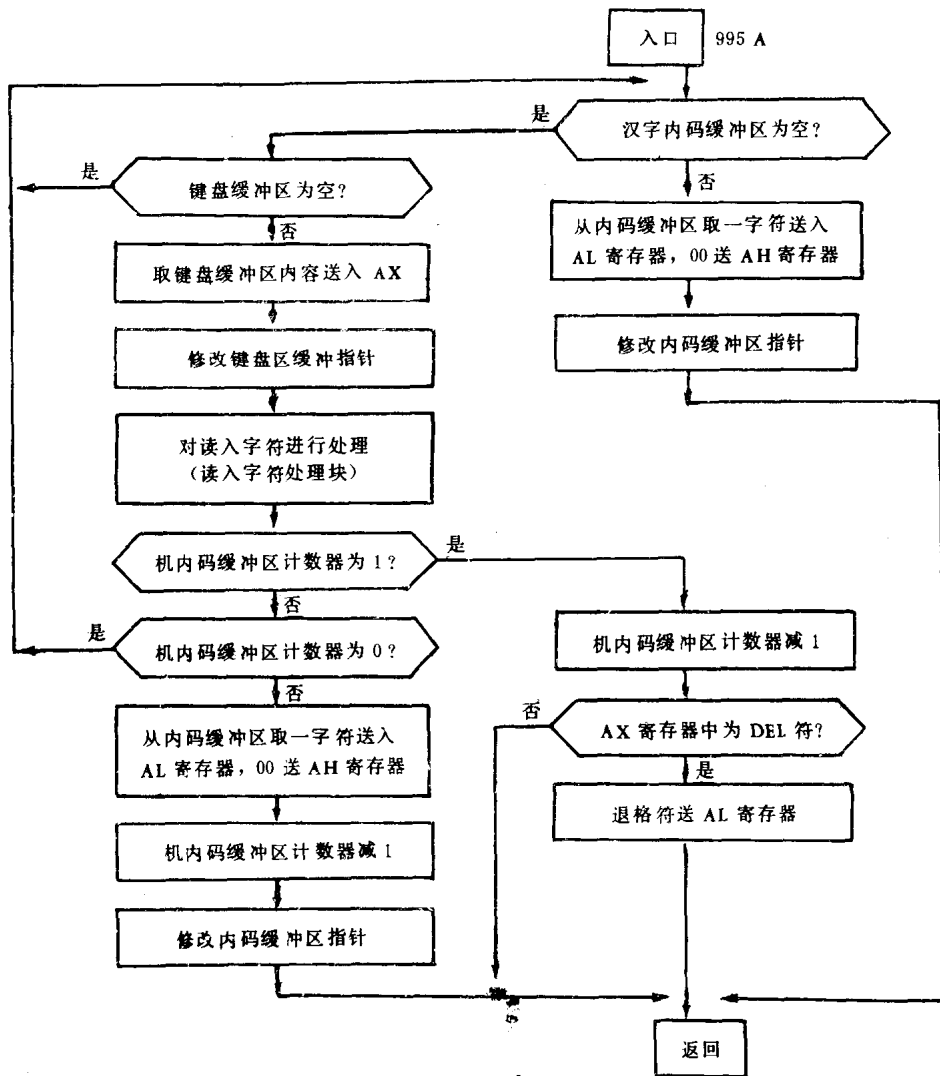


图 5-2 16 类中断程序的 0 号功能块流程

下面对图 5 - 2 进行几点说明。

1) 判断汉字机内码缓冲区内是否有信息,就是判断汉字机内码缓冲区计数器之值是否为零。如果该缓冲区内有信息,说明前一次输入字符时已形成了汉字机内码(两个字节),其中前一字节已由前一次返回了,后一字节尚在该缓冲区内,故要立即把后一字节返回,而不是去键盘数据缓冲区中取信息。

2) 判断键盘数据缓冲区是否为空,实际上是判断键盘数据缓冲区信息的首指针与尾指针的值是否相同,相同则表示该缓冲区内为空。

3) 读入字符处理模块的具体情况,由下节介绍和分析。

4) 读入字符处理模块与 0 号功能块的关系是子程序与主程序的关系,后者调用前者。前者执行完后,将在汉字机内码缓冲区计数器中反馈信息给调用者(0 号功能块),向它报告其执行的情况(详见下节)。然后,0 号功能块根据这个反馈信息,分别进行相应的处理。这也就是汉字机内码缓冲区计数器的又一作用,即为读入字符处理模块和 0 号功能块的通讯单元。

5.6 读入字符处理模块

本模块是 0 号功能块的一个子程序,它所完成的主要工作是,对从键盘数据缓冲区读入的字符进行判别,分别予以处理。如是功能符,则完成此功能符定义的功能。如是方式转换符,则完成工作方式的转换。如是汉字输入码字符,则进行对输入码的处理。它的流程如图 5 - 3 所示。

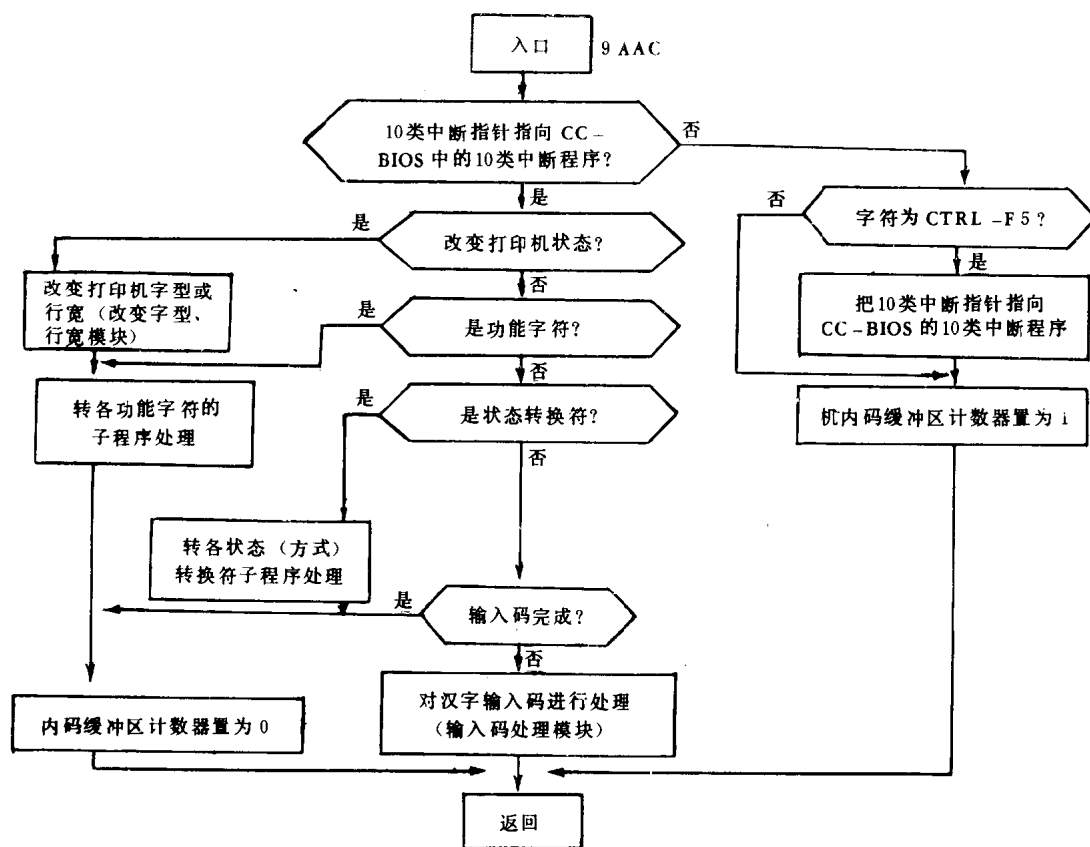


图 5 - 3 读入字符处理模块流程

下面对图 5 - 3 作一些解释。

1) 所谓 10 类中断程序, 即 CRT 驱动程序。10 类中断指针如指向 CC - BIOS 中的 10 类中断程序, CRT 驱动程序则为 CC - DOS 中的模式 (有汉字输出功能)。10 类中断指针如指向 ROM - BIOS 中的 10 类中断程序, CRT 驱动程序则为 PC - DOS 中的模式。CC - DOS 提供了可用 CTRL - F 5 功能键任意选择这两种模式的功能。

2) 所谓改变打印机状态, 是指 CC - DOS 提供了用 CTRL - F 10 功能键来重新定义打印机输出的字型和行宽的功能。若前一字符是 CTRL - F 10 时, 则当前的字符就用以确定打印机的字型或行宽。

3) 改变字型、行宽模块所做的工作较简单。改变字型时, 该模块向 17 类中断程序 (打印机驱动程序) 发送 ESC + "I" + 字型符 (字母 A ~ P 中之一), 由 17 类中断程序中的相应模块去完成改变字型的工作。改变行宽时, 则由本模块调用 17 类中断程序中的 3 号功能块来完成把行宽改为指定值的工作。

4) 功能符是指 CTRL - F 5 至 F 10。这些功能符在 CC - DOS 中被定义为相应的功能 (如表 5 - 3 所示), 它们对应的子程序就是分别完成这些功能的。

5) 方式转换符是指 ALT - F 1 至 F 4, F 6。在 CC - DOS 中, 它们被定义为各种工作方式 (如表 5 - 3 所示)。它们相应的子程序所做的工作是把工作方式状态字节中的相应位置位, 并在提示行 (屏蔽的最末一行) 中显示提示信息 (当前工作方式名)。

CC - DOS (V 2.1) 中功能键定义 表 5 - 3

功 能 键	定 义
ALT + F 1	区位码输入方式
ALT + F 2	首尾码输入方式
ALT + F 3	拼音码输入方式
ALT + F 4	快速码输入方式
ALT + F 6	ASCII 码输入方式
CTRL + F 5	改变 BIOS 中 10 类中断程序的模式
CTRL + F 6	改变显示字符的颜色
CTRL + F 7	纯西文和纯中文方式转换
CTRL + F 8	自动光标的建立和取消
CTRL + F 9	纯中文方式的建立和取消
CTRL + F 10	定义打印机的字型和行宽

●改色功能退出要再打 CTRL + F 6 和回车键。●纯中文方式是将 ASCII 字符放大。

6) 所谓输入码完成, 是指输入码完成状态字节为“完成”态 (详见 5.3 节)。

7) 输入码处理模块的情况, 在下节中进行介绍和分析。

5.7 输入码处理模块

输入码处理模块的主要功能是, 根据当前的工作方式, 调用相应的输入码处理子程序, 把相

应的输入码转换成机内码。它的主要部分是几个输入码处理子程序（子模块）。本模块的工作流程较简单，如图 5 - 4 所示。

在具体的程序实现中。首尾码处理模块与拼音码处理模块是合在一起的，称为首尾码 - 拼音码处理模块。下面我们对区位码处理模块和首尾码 - 拼音码处理模块进行剖析。

区位码处理模块的主要功能是，把输入的区位码经过计算，转换成机内码，并把它存入汉字机内码缓冲区中。它的工作流程如图 5 - 5 所示。

图 5 - 5 对区位码转换成机内码的整个过程，描述得很详细了，故不再另加说明了。

13 w

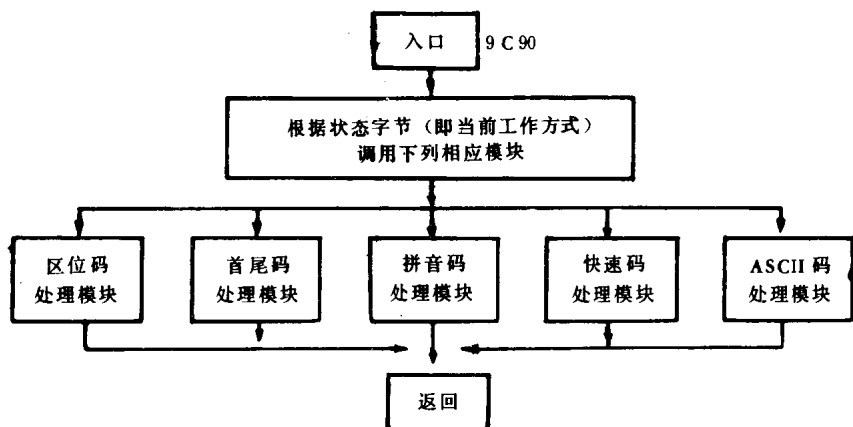


图 5 - 4 输入码处理模块流程

首尾码 - 拼音码处理模块的功能是：把输入的首尾码或拼音码所对应的所有汉字的机内码均找出来，并在提示行中把它们显示出来；再根据用户的选择，把选中的汉字之内机码存入汉字机内码缓冲区中。该模块的工作流程如图 5 - 6 所示。对图 5 - 6 要作下面几点说明。

1) 所谓上页和下页键，是指上、下页转换。提示行中每十个汉字为一页。提示行中每次最多只能显示十个汉字（一页），当重码汉字数超过十个时，则提示行中一次显示不下。用上、下页转换，可以把前十个或后十个重码汉字在提示行中显示出来，以供用户选择和确定汉字。

2) 上、下页处理模块的工作是：把前十个或后十个重码汉字显示到指示行中，把它们对应的机内码存入汉字重码缓冲区中。

3) 查扫描表模块主要完成以下工作：查找扫描表，把与当前汉字输入码缓冲区中内容相同的表项找出来。然后把找出的表项之序号转换成与其对应的机内码，再把机内码送重码缓冲区，并在提示行中显示其对应的汉字。要指出的是，我们的扫描表是首尾码和拼音码合用的，它的表项中前面为首尾码内容，后面为拼音码内容（见图 5 - 1）。所以，对于首尾码和拼音码在查表处理中有所不同。对于首尾码是查表项中从第一个字节开始的部分（到哪儿止，要根据当前输入码缓冲区中的字符数字），而对于拼音码，则是查表项中从第二个字节开始的部分（到哪儿止，要根据当前输入码缓冲区中的字符数字）。另外，在查表前要先对输入码缓冲区中的内容进行处理，把它们转换成表项内容的相应形式，然后再与表项内容（该内容亦要进行一些转换）进行对照。由此可见，查找的方法随输入码缓冲区中的字符数而定。当输入码缓冲区中的字符数分别为一字符、二字符和三字符时，其对应的查找方法分别为一字符查找重码法、二字符查找重码法和三字符查找

重码法。图 5 - 7 给出了一字符查找重码程序模块的工作流程。

图 5 - 7 对一字符查找重码的整个过程已描述得很详细了，故不再加说明。二字符查找重码和三字符查找重码的方法与图 5 - 7 类似，只是在字符内容转换上有些不同。

X

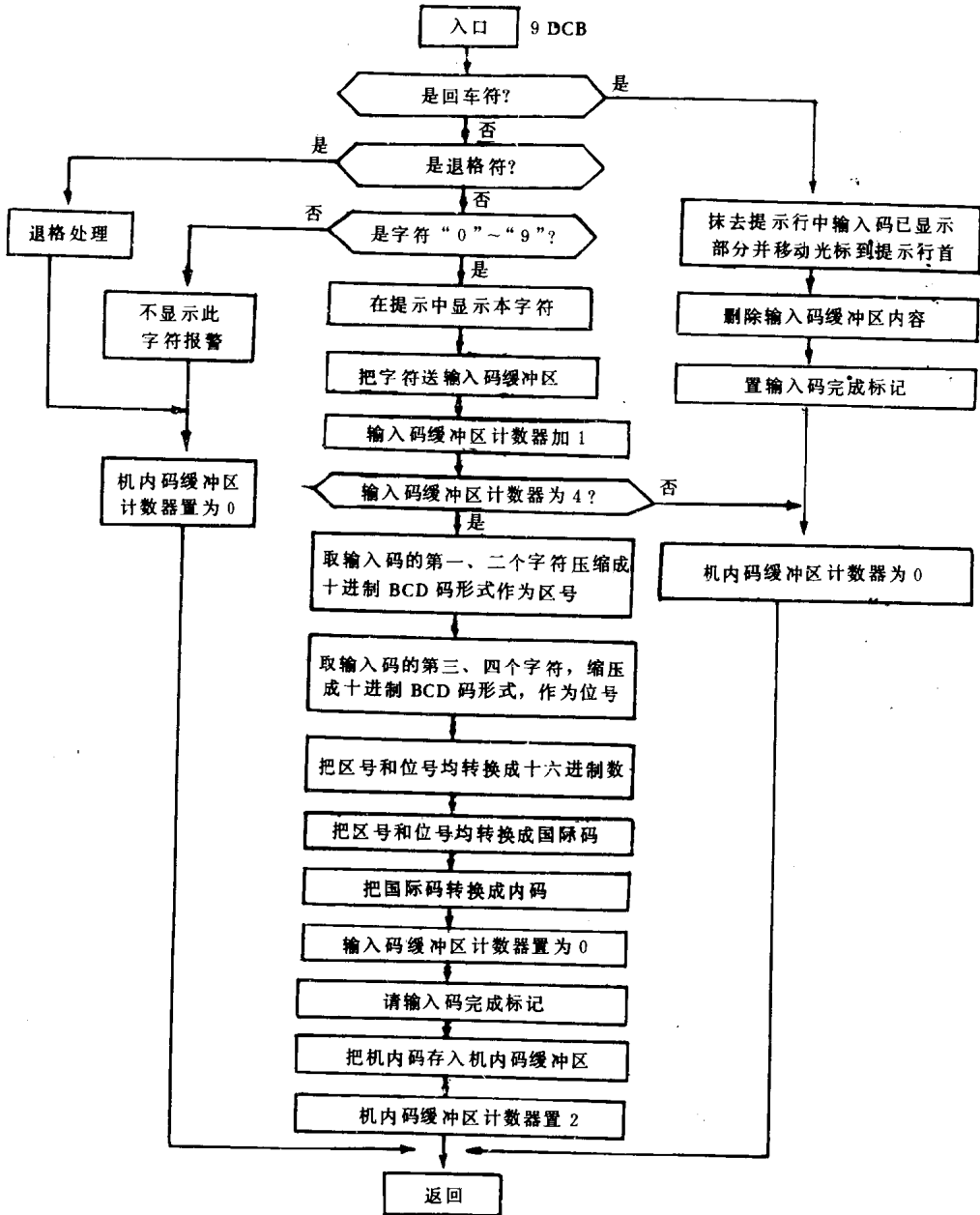


图 5 - 5 区位码处理模块流程

5.8 键盘管理模块 INT 16H

汉字库调入内存，CCBIOS形成后，内存主要模块有：显示器控制模块（INT 10H）— 内存地址 CS: 1848

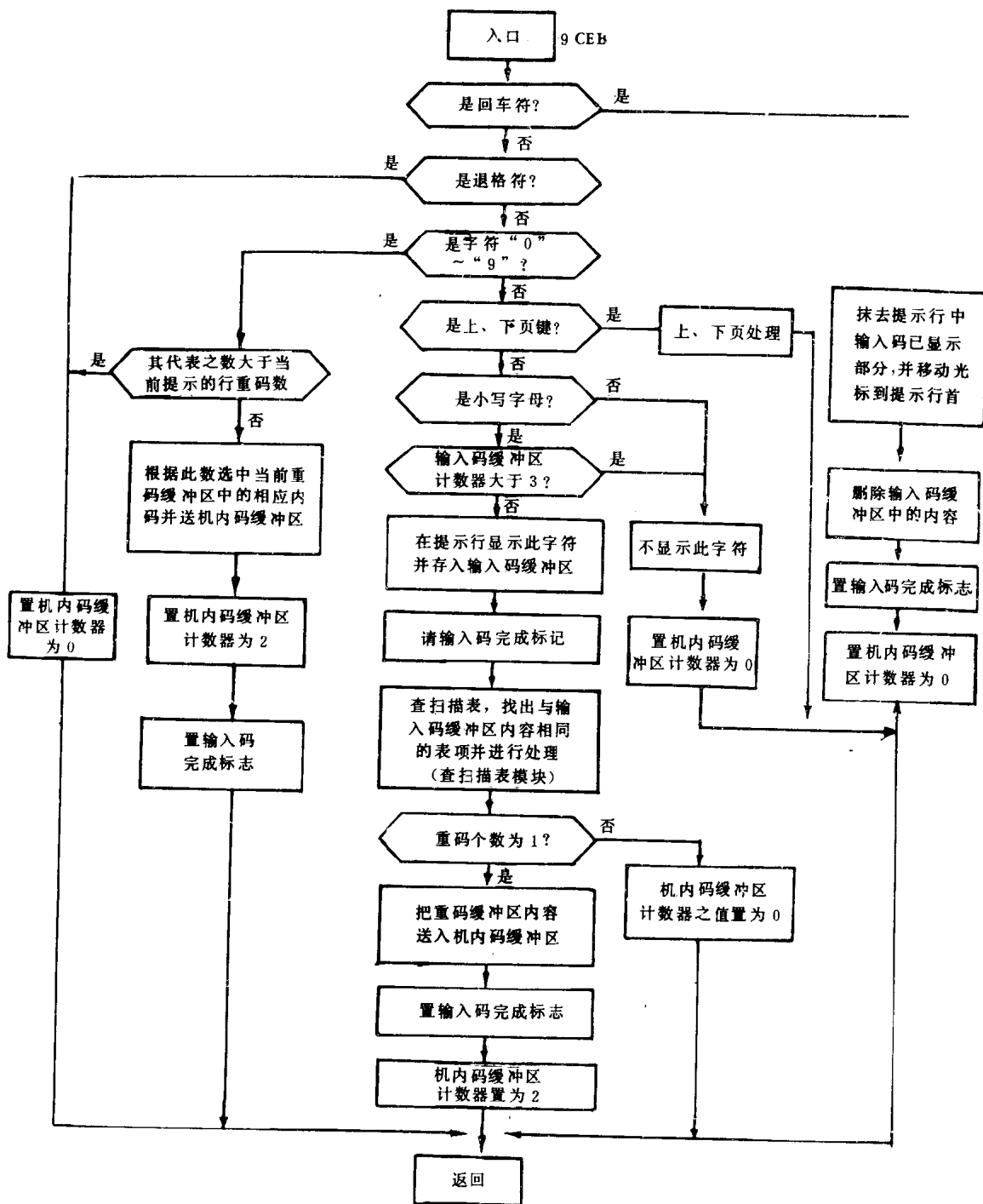


图 5-6 首尾码-拼音码处理模块流程

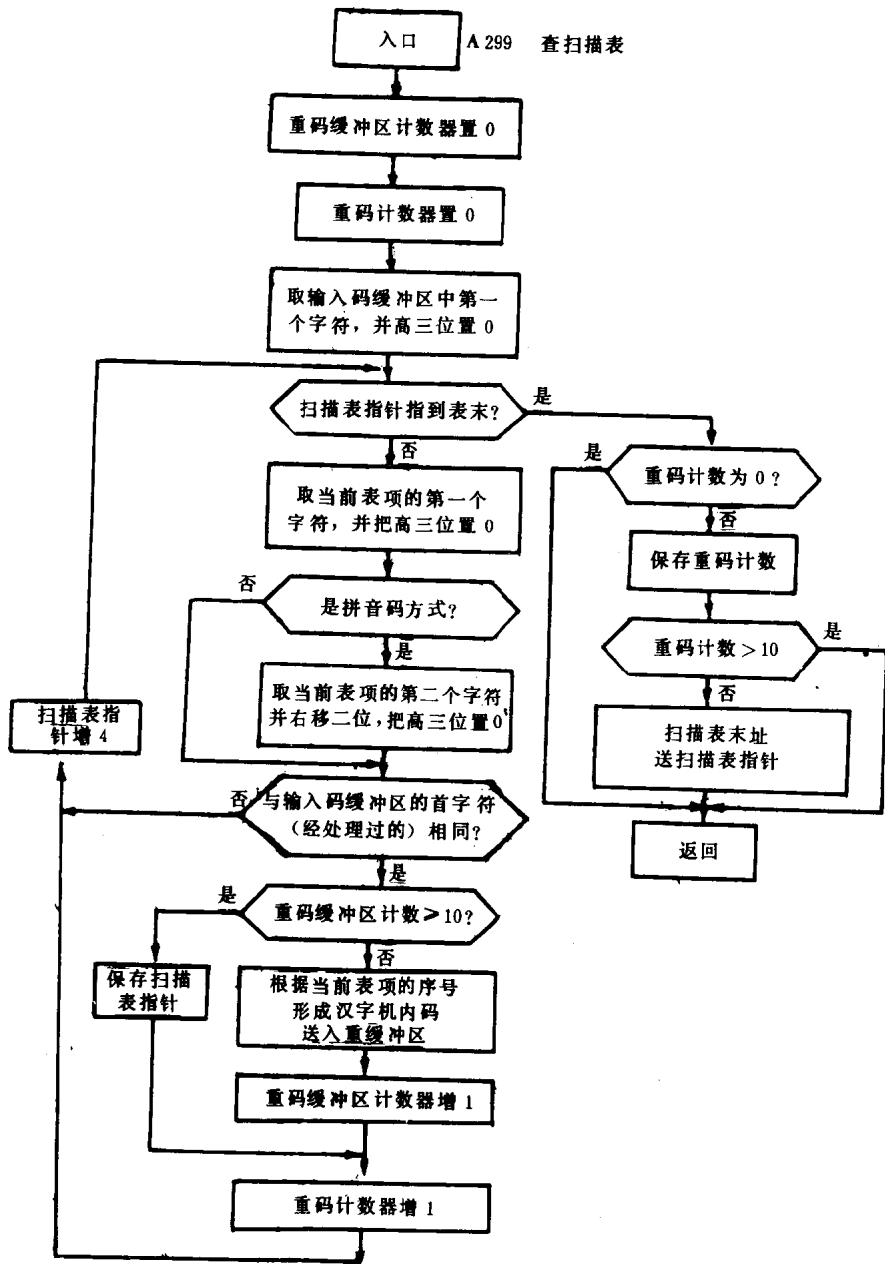


图 5-7 一字符查找重码处理流程

键盘管理数据区与工作区

-dcs:2b98 2B 98 词组表段地址 2B 9A 查词内码区指针

```
4E08:2B98 00 00 00 00 00 00 00 00
4E08:2BA0 00 00 00 00 00 00 00-00 00 00 00 00 D6 2B D6
4E08:2BB0 2B 00 00 00 00 00 00-00 00 00 20 00 00 00 00
4E08:2BC0 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
4E08:2BD0 0A 00 00 00 00 10 00 00-00 00 45 85 BD 01 23 B1
4E08:2BE0 BD 01 27 B1 BD 01 E6 B1-BD 01 26 85 B4 01 E4 B0
4E08:2BF0 BD 01 94 B2 BD 01 57 B0-BD 01 AA B2 BD 01 7A B2
4E08:2C00 BD 01 EA B1 BD 01 51 B1-BD 01 34 B0 BD 01 25 B3
4E08:2C10 BD 01 6A B2 BD 01 3A AB
```

-d

```
4E08:2C18 BD 01 62 AA BD 01 8B AA
4E08:2C20 BD 01 67 AA BD 01 B5 AA-BD 01 A2 A9 BD 01 6F AA
4E08:2C30 BD 01 82 A9 BD 01 8F A2-BD 01 B5 A0 BD 01 32 A3
4E08:2C40 BD 01 85 AE BD 01 E3 AD-BD 01 B3 AD BD 01 94 AC
4E08:2C50 BD 01 2D AD BD 01 EB AD-BD 01 74 AE BD 01 21 AD
4E08:2C60 BD 01 28 AD BD 01 BA 8A-B0 01 47 89 B0 01 67 8A
4E08:2C70 B0 01 66 8A B0 01 B6 8A-B0 01 98 8A B0 01 73 8A
4E08:2C80 B0 01 97 8A B0 01 95 8A-B0 01 27 B8 B0 01 26 88
4E08:2C90 B0 01 8A 8A B0 01 87 8A
```

9599 外码区首址 (2B B 1)

-dcs:8596

对照表结束地址

959D 内码区放提示行重码

```
4E08:8596 00 00-00 B1 2B FF 00 00 00 00
4E08:85A0 00 00 00 00 00 00 00-00 00 00 00 00 00 00
4E08:85B0 00 00 00 00 00 00 00-00 00 00 00 00 00 00
4E08:85C0 00 00 00 00 00 00 00-00 00 00 00 00 00 00
4E08:85D0 00 00 00 00 00 00 00 BB-C4 B1 E4 B5 B1 C7 B0 D7
4E08:85E0 D6 B7 FB D1 D5 C9 AB 3A-CA D7 CE B2 3A C7 F8 CE
4E08:85F0 BB 3A C6 B4 D2 F4 3A BF-EC CB D9 3A B4 CA D7 E9
4E08:8600 3A BD AB C1 A2 B4 BF D6-D0 CE C4 B7 BD CA BD CB
4E08:8610 A1 CF FB B4 BF D6
```

-d

```
4E08:8616 D0 CE-C4 B7 BD CA BD 00 00 00
4E08:8620 BD AB C1 A2 D7 D4 B6 AF-B9 E2 B1 EA FF CB A1 CF
4E08:8630 FB D7 D4 B6 AF B9 E2 B1-EA 00 10 41 53 43 49 49
4E08:8640 3A 00 00 00 00 00 00 00-00 D6 2B 00 00 00 00 00
4E08:8650 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00
4E08:8660 00 00 00 00 00 00 00 00-00 00 00 00 00 77 6D 58
4E08:8670 04 7A 67 5B 04 7A 64 5B-0A 64 5B 5B 14 6C 62 69
4E08:8680 44 7A 64 68 14 7A 73 5B-04 7A 6C 5B 04 7A 64 58
4E08:8690 18 6C 62 69 04 6C
```

-d

96A 1 词组表内码区起址

```
4E08:8696 62 69-44 68 75 69 AE 6C 75 78
4E08:86A0 60 CE D2 C3 C7 D6 D0 B9-FA D6 D0 B9 FA B9 B2 B2
4E08:86B0 FA B5 B3 B5 E7 D7 D3 B9-AA D2 B5 B2 BF B5 DA C1
4E08:86C0 F9 D1 D0 BE BF CB F9 B3-AF B4 C7 B0 D7 B5 DB B2
4E08:86D0 CA D4 C6 BC E4 2C 20 20-C7 A7 C0 EF BD AD C1 EA
```

.....
.....V+V -
+.....

2BD 6 输入码与机内码对照表

.....
.....
.....
.....
.....
.....

.....
.....
.....
.....
.....
.....
.....
.....
.....
.....

...1c....

.....
.....
.....
.....
.....
.....
.....
.....
.....
.....

形姆纹疆,

建立自动光标,取
自动光标,ASCII
.....
.....

966D 词组表外码

,zg[,zd[,d[,lbi
Dzdh,zs[,zlc[,zdc
lbi l

biDhui,lux

我们中国中国共
党电子工业部第
y研究所朝辞白帝
试津渡 千里江陵

4E08:96E0 D2 BB C8 D5 B8 B9 2E 20-20 C1 BD B0 B6 D4 B3 C9
4E08:96F0 F9 CC E4 B2 BB D7 A1 2C-20 20 C7 E0 D6 DB BC BA
4E08:9700 B9 FD CD F2 D6 D8 C9 BD-2E 20 20 D6 D0 B9 FA B9
4E08:9710 B2 B2 FA B5 B3 D6

一日还。两岸猿声不住，青舟已
过万重山。中国
共产党

-d

4E08:9716 D0 D1-EB CE AF D4 B1 BB E1 D5
4E08:9720 E2 CA C7 D5 E2 C0 EF D6-D0 B9 FA B5 E7 D7 D3 B9
4E08:9730 A4 D2 B5 B2 BF B5 DA C1-F9 D1 D0 BE BF CB F9 C0
4E08:9740 EE B0 D7 B9 CA C8 CB CE-F7 B4 C7 BB C6 BA D7 C2
4E08:9750 A5 2C 20 20 D1 CC BB AB-C8 FD D4 C2 CF C2 D1 EF
4E08:9760 D6 DD 2E 20 20 B9 C2 B7-AB D4 B6 D3 B0 B1 CC BF
4E08:9770 D5 BE A1 2C 20 20 CE AB-BC FB B3 A4 BD AD CC EC
4E08:9780 BC CA C1 F7 2E 20 20 BB-D8 B9 CB CD F9 CA C2 A3
4E08:9790 AC D3 CC CB E7 BF

雁群盼被
催饮铁船池 电子
挡康康研究所
吹坠啼来魏腔坪苗
力 烟花三月下扬
州。 孤帆远影碧
站 唯见长江天
际流。 西顾往事
从倒叙

-d

4E08:9796 B4 C3-A3 C3 A3 D4 C6 BA A3 A3
4E08:97A0 AC C2 D2 D4 C6 E7 CD B7-D7 A3 AC D2 B8 C6 AC C6
4E08:97B0 AC D4 DA CC EC BF D5 D6-D0 B7 C9 B6 C9 B6 FB B9
4E08:97C0 FD A3 AC D2 B8 C8 A5 B2-B8 B8 B4 B7 B5 A3 AC CE
4E08:97D0 AB D3 D0 C4 C7 D7 EE B2-D3 C0 F6 B5 C4 D4 C6 B6
4E08:97E0 E4 A3 AC C8 B4 D4 DA B9-D8 D4 C6 D5 DF B5 C4 C4
4E08:97F0 D4 BA A3 CC EF B3 A4 BE-C3 C6 AE B8 A1 A3 AE CD
4E08:9800 AF C4 EA C6 AC B6 CE CB-C6 B3 AF D4 C6 A3 AC CF
4E08:9810 BC D3 B0 C0 C3 C2

瘦C丁坪#
缚以歧头祝勿黄曲
从诺伟碧碧啥啥
}, 一去不复返,
坝心亲顺永能随贫
洞悉从控整替叩哪
院@钱天问院。叙
统和炊咽管吟囊
加扶痛

-d

4E08:9816 FE A3-AC D2 F2 B6 F8 D4 DA C8
4E08:9820 CB C3 C7 95 C4 BC C7 D2-E4 D6 D0 D2 C0 D2 C0 C1
4E08:9830 F4 C1 B5 A3 AE C2 B3 D1-B8 D4 F8 CB B5 A3 BA A3
4E08:9840 A2 CE D2 C3 C7 D0 E8 D2-AA B5 C4 A3 AC CA C7 C8
4E08:9850 AB B2 BF D7 F7 C6 B7 D6-D0 B5 C4 D5 E6 CA B5 B5
4E08:9860 C4 C9 FA BB EE A3 AC C9-FA C1 FA BB EE BB A2 B5
4E08:9870 C4 D5 BD CA BF A3 AC CC-FB B6 AF B5 C4 C2 F6 B2
4E08:9880 AB A3 AC CB BC CF EB BA-CD CB C8 C7 E9 B5 CB B5
4E08:9890 C8 A3 AE A3 A2 00

。 因而在
嗣型舞且道幸路
袁担埔逞网x说:
响随切身羽模河桥
巢孔黛分城悟燥档
纳:活, 生龙活虎
悟续没动的脉
常此御舞慧梢韵
酥叮

-dcs:9895

4E08:9895 00 B4 F2-D3 A1 D7 D6 BA C5 28 41
4E08:98A0 2D 50 29 3A 20 D6 BD BF-ED 28 38 30 2D 31 33 34
4E08:98B0 29 3A 20 FC FB 06 1E 53-56 57 51 55 52 BB 40 00
4E08:98C0 BE C3 8C CB BE DB 0A E4-75 03 E9 8D 00 FE CC 75
4E08:98D0 03 EB 4A 90 FE CC 74 7C-FE CC 74 2D FE CC 74 26
4E08:98E0 FE CC 74 36 FE CC 75 15-3C 00 75 0B A0 3A 96 A2
4E08:98F0 D5 2B B0 10 EB 04 90 A0-D5 2B A2 3A 96 5A 5D 59
4E08:9900 5F 5E 5B 1F 07 CF E9 21-01 BE DD 8B F2 8D 3E D6
4E08:9910 2B 0E 07 B9 C0

打印字号
-P) 纸宽(89-134
) (C..SUHQIR)@
.C.K.[edu.i.. "Lu
.kJ. "L@i" "Lt-" "Lt&
" "Lt6" "Lh.<.u. :.
" "k., U+" :ZIV
_[" "神"] .x.)U
+.. 估

-u cs:98b3

4E08:98B3 FC CLD ; INT 16H 键盘管理入口
4E08:98B4 FB STI
4E08:98B5 06 PUSH ES

4E08:98B6 1E	PUSH	DS	
4E08:98B7 53	PUSH	BX	
4E08:98B8 56	PUSH	SI	
4E08:98B9 57	PUSH	DI	
4E08:98BA 51	PUSH	CX	
4E08:98BB 55	PUSH	BP	
4E08:98BC 52	PUSH	DX	
4E08:98BD B84000	MOV	BX,0040	
4E08:98C0 8EC3	MOV	ES,BX	
4E08:98C2 8CCB	MOV	BX,CS	: 改数据段 DS = CS
4E08:98C4 8EDB	MOV	DS,BX	
4E08:98C6 0AE4	OR	AH,AH	: 若AH = 0, 转 995A, 0号功能模块
4E08:98C8 7503	JNZ	98CD	汉字功能键, 控制键处理
4E08:98CA E98D00	JMP	995A	
4E08:98CD FECC	DEC	AH	: AH = 1, 转991D, 1号功能块
4E08:98CF 7503	JNZ	98D4	读键盘或内码缓冲区
4E08:98D1 EB4A	JMP	991D	
-d			
4E08:98D3 90	NOP		
4E08:98D4 FECC	DEC	AH	: AH = 2, 转9954, 2号功能模块
4E08:98D6 747C	JZ	9954	
4E08:98D8 FECC	DEC	AH	: AH = 3, 转9909, 3号功能块
4E08:98DA 742D	JZ	9909	传送外码—内码对照表
4E08:98DC FECC	DEC	AH	: AH = 4, 转9906, 4号功能块
4E08:98DE 7426	JZ	9906	建词组表寻址指针
4E08:98E0 FECC	DEC	AH	: 若AH = 5, 转991A, 5号功能块
4E08:98E2 7436	JZ	991A	保存CRT管理块地址
4E08:98E4 FECC	DEC	AH	: 若AH > 6, 转99FD, 中断返回
4E08:98E6 7515	JNZ	98FD	
4E08:98E8 3C00	CMP	AL,00	
4E08:98EA 750B	JNZ	98F7	
4E08:98EC A03A96	MOV	AL,[963A]	
4E08:98EF A2D52B	MOV	[2B05],AL	
4E08:98F2 B010	MOV	AL,10	
-f			
4E08:98F4 EB04	JMP	98FA	
4E08:98F6 90	NOP		
4E08:98F7 A0D52B	MOV	AL,[2B05]	
4E08:98FA A23A96	MOV	[963A],AL	: 恢复现场
4E08:98FD 5A	POP	DX	
4E08:98FE 5D	POP	BP	
4E08:98FF 59	POP	CX	
4E08:9900 5F	POP	DI	
4E08:9901 5E	POP	SI	
4E08:9902 5B	POP	BX	
4E08:9903 1F	POP	DS	
4E08:9904 07	POP	ES	
4E08:9905 CF	IRET		

20 × 15 = 300

4E08:9906 E92101	JMP	9A2A	; 转 9 A 2 A , 4 号功能模块
4E08:9909 8EDD	MOV	DS,BP	; 取用户外码—内码对照表地址送 DS:SI
4E08:990B 8BF2	MOV	SI,DI	
4E08:990D 8D3ED62B	LEA	DI,[2BD6]	
4E08:9911 0E	PUSH	CS	; ES值改为CS值
4E08:9912 07	POP	ES	
4E08:9913 B9C069	MOV	CX,69C0	; 69C 0 H → CX (外码—内码表长度)
-u			
4E08:9916 F3	REPZ		
4E08:9917 A4	MOVSB		; 外码—内码对照表传送到ES: 2 B D 6 开始
4E08:9918 EBE3	JMP	98FD	
4E08:991A E92101	JMP	9A3E	
4E08:991D FA	CLI		
4E08:991E 26	ES:		; ES = 0400H
4E08:991F 8B1E1A00	MOV	BX,[001A]	
4E08:9923 26	ES:		
4E08:9924 8B07	MOV	AX,[BX]	; 键盘缓冲区当前首指针下的键盘码 → A X
4E08:9926 26	ES:		
4E08:9927 3B1E1C00	CMP	BX,[001C]	; 键盘缓冲区有数据吗?
4E08:992B 751B	JNZ	9948	
4E08:992D 8036A92BFF	XOR	BYTE PTR [2BA9],FF	
4E08:9932 F606A92B80	TEST	BYTE PTR [2BA9],80	
-u			
4E08:9937 740F	JZ	9948	
4E08:9939 803ED59500	CMP	BYTE PTR [95D5],00	
4E08:993E 7408	JZ	9948	
4E08:9940 8B3EAA2B	MOV	DI,[2BAA]	
4E08:9944 B400	MOV	AH,00	; ≠ 0 , 根据汉字内码区指针读内码 → AL
4E08:9946 8A05	MOV	AL,[DI]	
4E08:9948 FB	STI		; 开中断
4E08:9949 5A	POP	DX	; 恢复现场
4E08:994A 5D	POP	BP	
4E08:994B 59	POP	CX	
4E08:994C 5F	POP	DI	
4E08:994D 5E	POP	SI	
4E08:994E 5B	POP	BX	
4E08:994F 1F	POP	DS	
4E08:9950 07	POP	ES	
4E08:9951 CA0200	RETF	0002	
4E08:9954 26	ES:		
4E08:9955 A01700	MOV	AL,[0017]	; 当前键盘状态 → AL
-u			
4E08:9958 EBA3	JMP	98FD	
4E08:995A FB	STI		
4E08:995B 90	NOP		
4E08:995C FA	CLI		
4E08:995D 803ED59500	CMP	BYTE PTR [95D5],00	; 内码区有数据吗?
4E08:9962 7574	JNZ	99D8	

4E08:9964 26	ES:		
4E08:9965 8B1E1A00	MOV	BX,[001A]	
4E08:9969 26	ES:		
4E08:996A 3B1E1C00	CMP	BX,[001C]	
4E08:996E 74EA	JZ	995A	
4E08:9970 26	ES:		
4E08:9971 8B07	MOV	AX,[BX]	: 从键盘缓冲区读字符→AX
4E08:9973 E8A100	CALL	9A17	: 修改缓冲区首指针
4E08:9976 26	ES:		
4E08:9977 891E1A00	MOV	[001A],BX	
-u			
4E08:997B 50	PUSH	AX	: 保存当前读入的字符
4E08:997C FB	STI		
4E08:997D E82E01	CALL	9AAC	: 键盘码处理子程序对读入的字符进行处理
4E08:9980 FA	CLI		
4E08:9981 58	POP	AX	: 恢复读入的字符
4E08:9982 803ED59501	CMP	BYTE PTR [95D5],01	
4E08:9987 7520	JNZ	99A9	
4E08:9989 FE0ED595	DEC	BYTE PTR [95D5]	: 若 = 1, 内码计数器 - 1
4E08:998D 3D7F0E	CMP	AX,0E7F	: Ctr "1" + "←" ?
4E08:9990 7514	JNZ	99A6	
4E08:9992 B00B	MOV	AL,0B	: 是, 08H → AL
4E08:9994 C606D59501	MOV	BYTE PTR [95D5],01	
4E08:9999 C606A42BFF	MOV	BYTE PTR [2BA4],FF	: FFH → 退格标志
-u			
4E08:999E C606A92BFF	MOV	BYTE PTR [2BA9],FF	: FFH → 读内码区标志
4E08:99A3 A39D95	MOV	[959D],AX	: AX → 汉字内码区
4E08:99A6 EB27	JMP	99CF	
4E08:99A8 90	NOP		
4E08:99A9 803ED59500	CMP	BYTE PTR [95D5],00	: 内码区有数据吗?
4E08:99AE 74AA	JZ	995A	
4E08:99B0 C606A92BFF	MOV	BYTE PTR [2BA9],FF	: 否则, 设置读内码区标志
4E08:99B5 BF9D95	MOV	DI,959D	: 取内码缓冲区首址
4E08:99B8 893EAA2B	MOV	[2BAA],DI	: 内码缓冲区指针指向内码缓冲区首址
4E08:99BC 8A16D595	MOV	DL,[95D5]	
-u			
4E08:99C0 FECA	DEC	DL	: 内码计数器 - 1
4E08:99C2 8B16D595	MOV	[95D5],DL	: 送回内码计数器
4E08:99C6 8A05	MOV	AL,[D1]	
4E08:99C8 47	INC	DI	: 内码缓冲区指针 + 1
4E08:99C9 893EAA2B	MOV	[2BAA],DI	
4E08:99CD B400	MOV	AH,00	
4E08:99CF 5A	POP	DX	: 恢复现场
4E08:99D0 5D	POP	BP	
4E08:99D1 59	POP	CX	
4E08:99D2 5F	POP	DI	
4E08:99D3 5E	POP	SI	
4E08:99D4 5B	POP	BX	

4E08:99D5 1F	POP	DS	
4E08:99D6 07	POP	ES	
4E08:99D7 CF	IRET		
4E08:99D8 803EA42BFF	CMP	BYTE PTR [2BA4],FF	: 退格标志=FFH (Crt "1" + "←")
4E08:99DD 750F	JNZ	99EE	
4E08:99DF C606A42B00	MOV	BYTE PTR [2BA4],00	: 若是, 00→退格标志
-u			
4E08:99E4 A19D95	MOV	AX,[959D]	: 内码缓冲区头两个字节→AX
4E08:99E7 C606D59500	MOV	BYTE PTR [95D5],00	
4E08:99EC EBE1	JMP	99CF	
4E08:99EE 8B3EAA2B	MOV	DI,[2BAA]	
4E08:99F2 8A05	MOV	AL,[DI]	: 内码→AL
4E08:99F4 B400	MOV	AH,00	
4E08:99F6 47	INC	DI	
4E08:99F7 893EAA2B	MOV	[2BAA],DI	
4E08:99FB 8A16D595	MOV	DL,[95D5]	: 内码计数器-1
4E08:99FF FECA	DEC	DL	
4E08:9A01 8B16D595	MOV	[95D5],DL	
-u			
4E08:9A05 75C8	JNZ	99CF	
4E08:9A07 BF9D95	MOV	DI,959D	
4E08:9A0A 893EAA2B	MOV	[2BAA],DI	
4E08:9A0E 803E6B9600	CMP	BYTE PTR [966B],00	: 词组计数器=0?
4E08:9A13 750E	JNZ	9A23	: 若≠0, 转9A23, 选中词组处理
4E08:9A15 EBB8	JMP	99CF	
4E08:9A17 83C302	ADD	BX,+02	: 首指针+2, 指向下一个字
4E08:9A1A 83FB3E	CMP	BX,+3E	: 首指针指向键盘缓冲区尾部吗?
4E08:9A1D 7503	JNZ	9A22	: 若没有指向尾部, 转9A22子程返回
4E08:9A1F BB1E00	MOV	BX,001E	: 若指向尾部, 修改指针指向键盘缓冲区首址
4E08:9A22 C3	RET		
4E08:9A23 50	PUSH	AX	
4E08:9A24 EBDA0C	CALL	A701	: 选中词组内码处理
-u			
4E08:9A27 5B	POP	AX	
4E08:9A28 EBA5	JMP	99CF	: 转99CF, 中断返回
4E08:9A2A 892E982B	MOV	[2B98],BP	: BP→词组表段码单元
4E08:9A2E C6066C96FF	MOV	BYTE PTR [966C],FF	: FF→用户词组表标志
4E08:9A33 8916A22B	MOV	[2BA2],DX	: DX→词组表外码区首址单元
4E08:9A37 890EA02B	MOV	[2BA0],CX	: CX→词组表内码区首址单元
4E08:9A3B E9BFFE	JMP	98FD	
4E08:9A3E 57	PUSH	DI	
4E08:9A3F BF4000	MOV	DI,0040	
4E08:9A42 0AC0	OR	AL,AL	: 若AL=0
4E08:9A44 7410	JZ	9A56	: 转9A56, 保存汉字显示入口地址
4E08:9A46 FE08	DEC	AL	: 若AL=1
-u			
4E08:9A48 7418	JZ	9A62	: 转9A62, 切换10H中断为汉字显示
4E08:9A4A FE08	DEC	AL	: 若AL=2

4E08:9A4C 7433	JZ	9A81	: 转 9 A 81, 切换 10 中断为西文显示
4E08:9A4E FECB	DEC	AL	: 若 AL = 3
4E08:9A50 744E	JZ	9AA0	: 保存西文显示入口地址
4E08:9A52 5F	POP	DI	: 转 98 FD, 中断返回
4E08:9A53 E9A7FE	JMP	9BFD	
4E08:9A56 0E	PUSH	CS	
4E08:9A57 1F	POP	DS	: 改 DS = CS
4E08:9A5B 892E4196	MOV	[9641],BP	: 将 BP:DX 给出的汉字显示口地址汉字显示口地址单元
4E08:9A5C 89164396	MOV	[9643],DX	
4E08:9A60 EBF1	JMP	9A53	: 转 9 A 53, 中断返回
4E08:9A62 E80200	CALL	9A67	: 修改 10H 中断指针指向汉字显示口
4E08:9A65 EBEC	JMP	9A53	
4E08:9A67 50	PUSH	AX	
-u			
4E08:9A68 1E	PUSH	DS	
4E08:9A69 33C0	XOR	AX,AX	
4E08:9A6B 8ED8	MOV	DS,AX	: 数据段指向 DOS 中断向量段地 (0 0 0 0:)
4E08:9A6D 2E	CS:		
4E08:9A6E 8B1E4396	MOV	BX,[9643]	
4E08:9A72 891D	MOV	[DI],BX	
4E08:9A74 83C702	ADD	DI,+02	
4E08:9A77 2E	CS:		
4E08:9A78 8B1E4196	MOV	BX,[9641]	: 汉字显示口地址 → BX
4E08:9A7C 891D	MOV	[DI],BX	
4E08:9A7E 1F	POP	DS	
4E08:9A7F 5B	POP	AX	
4E08:9A80 C3	RET		
4E08:9A81 E80200	CALL	9A86	
4E08:9A84 EBEC	JMP	9A53	
4E08:9A86 50	PUSH	AX	
4E08:9A87 1E	PUSH	DS	
-u			
4E08:9A88 33C0	XOR	AX,AX	
4E08:9A8A 8ED8	MOV	DS,AX	: 数据段指向 DOS 中断向量地址 (0 0 0 0:)
4E08:9A8C 2E	CS:		
4E08:9A8D 8B1E4796	MOV	BX,[9647]	: 切换 DOS 10H 中断向量
4E08:9A91 891D	MOV	[DI],BX	: (由 DS: DI 给出) 指向 ROM - BIOS
4E08:9A93 83C702	ADD	DI,+02	: 10H 中断入口 (即西文显示口) 其地址保存在
4E08:9A96 2E	CS:		西文显示口地址单元
4E08:9A97 8B1E4596	MOV	BX,[9645]	
4E08:9A9B 891D	MOV	[DI],BX	
4E08:9A9D 1F	POP	DS	
4E08:9A9E 5B	POP	AX	
4E08:9A9F C3	RET		
4E08:9AA0 0E	PUSH	CS	: 改 DS = CS
4E08:9AA1 1F	POP	DS	
4E08:9AA2 892E4596	MOV	[9645],BP	: 将 BP:DX 给出的西文显示口地址保存在西文
4E08:9AA6 89164796	MOV	[9647],DX	显示口地址单元

```

-u
4E08:9AAA EBA7      JMP      9A53
4E08:9AAC F606399610 TEST     BYTE PTR [9639],10
4E08:9AB1 7427      JZ       9ADA
4E08:9AB3 EB01      JMP      9AB6
4E08:9AB5 90        NOP
4E08:9AB6 3D0062     CMP     AX,6200          ; Ctrl+F5 (改INT 10H中断)
4E08:9AB9 751C      JNZ     9AD7
4E08:9ABB C606399600 MOV     BYTE PTR [9639],00
4E08:9AC0 BCCD      MOV     BP,CS
4E08:9AC2 BD164818 LEA     DX,[1848]        ; 取汉字口10H中断地址→DX
4E08:9AC6 892E4196 MOV     [9641],BP      ; 汉字口地址(CC-BIOS 10H中断程序入口)
                          ; 送汉字显示口地址单元
-u
4E08:9ACA 89164396 MOV     [9643],DX
4E08:9ACE BF4000     MOV     DI,0040
4E08:9AD1 EB93FF     CALL    9A67          ; 切换10H中断向量指向汉字口
4E08:9AD4 E9E200     JMP     9BB9
4E08:9AD7 E9DF00     JMP     9BB9
4E08:9ADA 3D0064     CMP     AX,6400        ; Ctrl+F1, 汉字与西文转换
4E08:9ADD 751E      JNZ     9AFD
4E08:9ADF 26        ES:
4E08:9AE0 A04900     MOV     AL,0049
4E08:9AE3 3C03      CMP     AL,03
4E08:9AE5 7605      JBE     9AEE
4E08:9AE7 2C03      SUB     AL,03
4E08:9AE9 EB03      JMP     9AEE          ; 转9 A EE显示器模式转换
-u
4E08:9AEB 90        NOP
4E08:9AEC 0403      ADD     AL,03
4E08:9AEE 32E4      XOR     AH,AH
4E08:9AF0 CD10      INT     10
4E08:9AF2 90        NOP
4E08:9AF3 8BEC      MOV     BP,SP
4E08:9AF5 C746020D1C MOV     WORD PTR [BP+02],1C0D
4E08:9AFA E9BC00     JMP     9BB9
4E08:9AFD 803E9598FF CMP     BYTE PTR [9895],FF ; 定义打印机过程Ctrl+F10处理完了吗?
4E08:9B02 7503      JNZ     9B07
4E08:9B04 E9090F     JMP     AA10          ; 没有处理完转AA10继续处理
4E08:9B07 3D0067     CMP     AX,6700        ; Ctrl+F10(进入定义打印机字号,行宽的过程)
4E08:9B0A 7503      JNZ     9B0F
-u
4E08:9B0C E9500E     JMP     A95F
4E08:9B0F 803ED695FF CMP     BYTE PTR [95D6],FF ; 改变字符颜色Ctrl+F6过程结束了吗?
4E08:9B14 752A      JNZ     9B40
4E08:9B16 3C0D      CMP     AL,0D
4E08:9B18 750D      JNZ     9B27
4E08:9B1A C606D69500 MOV     BYTE PTR [95D6],00
4E08:9B1F B200      MOV     DL,00
4E08:9B21 EB0C04     CALL    9F30

```

4E08:9B24	E99E02	JMP	9DC5	
4E08:9B27	8A1E7700	MOV	BL,[0077]	: 颜色寄存器控制字→BL
4E08:9B2B	80E30F	AND	BL,0F	
-u				
4E08:9B2E	FEC3	INC	BL	: 改变当前颜色码为下一个颜色码
4E08:9B30	80CB30	OR	BL,30	
4E08:9B33	881E7700	MOV	[0077],BL	: 改变后的颜色码送回颜色寄存器单元
4E08:9B37	B40B	MOV	AH,0B	: OB→AH改变当前CRT颜色
4E08:9B39	8700	MOV	BH,00	
4E08:9B3B	CD10	INT	10	
4E08:9B3D	E98502	JMP	9DC5	
4E08:9B40	3D0063	CMP	AX,6300	: 改变当前字符颜色 (Ctrl+F ₆)?
4E08:9B43	7517	JNZ	9B5C	
4E08:9B45	8A0000	MOV	DX,0000	
4E08:9B48	E8E503	CALL	9F30	: 转子程序, 清提示行
4E08:9B4B	8D1ED795	LEA	BX,[95D7]	
-u				
4E08:9B4F	32D2	XOR	DL,DL	
4E08:9B51	C606D695FF	MOV	BYTE PTR [95D6],FF	: FFH→改变字符颜色过程标志
4E08:9B56	B91100	MOV	CX,0011	: 提示行信息字节数→CX
4E08:9B59	E9ED06	JMP	A249	
4E08:9B5C	3D0062	CMP	AX,6200	: Ctrl+F ₅ (切换10H中断向量)?
4E08:9B5F	740B	JZ	9B6C	
4E08:9B61	26	ES:		
4E08:9B62	803E490006	CMP	BYTE PTR [0049],06	: 当前是640 × 200 图形方式
4E08:9B67	741F	JZ	9B88	: 若是, 转 9 B 88
4E08:9B69	EB4E	JMP	9B89	: 否则, 转 9 B B 9
4E08:9B6B	90	NOP		
4E08:9B6C	C606399610	MOV	BYTE PTR [9639],10	
-u				
4E08:9B71	B000F0	MOV	BP,F000	
4E08:9B74	8A65F0	MOV	DX,F065	
4E08:9B77	892E4596	MOV	[9645],BP	
4E08:9B7B	89164796	MOV	[9647],DX	
4E08:9B7F	BF4000	MOV	DI,0040	
4E08:9B82	E801FF	CALL	9A86	
4E08:9B85	E93D02	JMP	9DC5	
4E08:9B88	26	ES:		
4E08:9B89	803E490006	CMP	BYTE PTR [0049],06	: 当前是640 × 200 图形方式?
4E08:9B8E	7509	JNZ	9B99	
4E08:9B90	C706D02B0A00	MOV	WORD PTR [2BD0],000A	: 若是, 10→汉字重码缓冲区最大容量
-u				
4E08:9B96	EB07	JMP	9B9F	
4E08:9B98	90	NOP		
4E08:9B99	C706D02B0500	MOV	WORD PTR [2BD0],0005	: 否则, 彩色图形方式, 5→汉字重码缓冲区最
4E08:9B9F	3C00	CMP	AL,00	: 是功能键吗?
4E08:9BA1	741C	JZ	9BBF	大容量
4E08:9BA3	E9EA00	JMP	9C90	
4E08:9BA6	803EAC2BFF	CMP	BYTE PTR [2BAC],FF	: 不是, 转 9 C 90, ASC I 码汉字外吗处理模块
				: 当前是纯中文方式吗?

4E08:9BAB 750C	JNZ	9BB9	: 若不是纯中文方式, 转 9 B B 9 键盘码处理了程序返回
4E08:9BAD 8D3E9D95	LEA	DI,[959D]	: 若是, 汉字内码缓冲区首址→DI
4E08:9BB1 C606D59500	MOV	BYTE PTR [95D5],00	: 0 0 →内码计数器
-u			
4E08:9BB6 E97C0D	JMP	A935	: 转 A 9 35 纯中文方式处理
4E08:9BB9 C606D59501	MOV	BYTE PTR [95D5],01	
4E08:9BBE C3	RET		
4E08:9BBF 80FC68	CMP	AH,68	: Alt + F ₁ ~F ₆ 功能键吗?
4E08:9BC2 7203	JB	9BC7	
4E08:9BC4 E84F	JMP	9C15	: 是, 转 9 C 15 汉字输入方式选择处理
4E08:9BC6 90	NOP		
4E08:9BC7 80FC66	CMP	AH,66	
4E08:9BCA 7519	JNZ	9BE5	
4E08:9BCC B80F96	MOV	BX,960F	: “取消纯中文方式” 提示信息地址→BX
4E08:9BCF 8036AC2BFF	XOR	BYTE PTR [2BAC],FF	
4E08:9BD4 7403	JZ	9BD9	
-u			
4E08:9BD6 BB0196	MOV	BX,9601	
4E08:9BD9 B90E00	MOV	CX,000E	: 提示信息字节数(长度) →CX
4E08:9BDC E8EE00	CALL	9CCD	: 转子程序, 在提示行写提示信息
4E08:9BDF C606D59500	MOV	BYTE PTR [95D5],00	
4E08:9BE4 C3	RET		
4E08:9BE5 80FC65	CMP	AH,65	: Ctrl + F。(建立/取消自动光标)?
4E08:9BE8 7528	JNZ	9C12	: 不是, 转 9 C 12 进入汉字处理
4E08:9BEA BB2D96	MOV	BX,962D	: “取消自动光标” 提示信息地址→BX
4E08:9BED 80362C96FF	XOR	BYTE PTR [962C],FF	: 改变自动光标状态, 当前是自动建立光标状态吗?
4E08:9BF2 7505	JNZ	9BF9	
4E08:9BF4 B000	MOV	AL,00	
-u			
4E08:9BF6 EB06	JMP	9BFE	
4E08:9BF8 90	NOP		
4E08:9BF9 B001	MOV	AL,01	: 建立自动光标参数→AL
4E08:9BFB BB2096	MOV	BX,9620	
4E08:9BFE 53	PUSH	BX	
4E08:9BFF B413	MOV	AH,13	: 建立/取消自动光标
4E08:9C01 CD10	INT	10	
4E08:9C03 5B	POP	BX	
4E08:9C04 B90C00	MOV	CX,000C	: 提示信息字节数(长度) →CX
4E08:9C07 E8C300	CALL	9CCD	: 转子程序, 在提示行写提示信息
4E08:9C0A B00E9B95FF	OR	BYTE PTR [959B],FF	: 设置第一键标志为第一键状态
4E08:9C0F E9B301	JMP	9DC5	
4E08:9C12 EB7C	JMP	9C90	: 转 9 C 90, 键盘码处理子程序进入ASCII码汉字处理模块
4E08:9C14 90	NOP		
4E08:9C15 80FC68	CMP	AH,68	: Alt + F ₁ (区位码功能键)?
-u			
4E08:9C1B 750E	JNZ	9C28	
4E08:9C1A C6063A9601	MOV	BYTE PTR [963A],01	: 01(区位码) →汉字输入方式标志
4E08:9C1F BBED95	MOV	BX,95ED	: “区位:” 提示信息地址→BX
4E08:9C22 EBAC00	CALL	9CD1	: 转子程序, 清提示行, 转提示行信息, 第一键状态→第键标志

4E08:9C25 E99D01	JMP	9DC5	
4E08:9C28 80FC69	CMP	AH,69	: Alt+F2(首尾码功能键)?
4E08:9C2B 751B	JNZ	9C45	: 不是, 转9C45
4E08:9C2D C6063A9602	MOV	BYTE PTR [963A],02	: 02(首尾码)→汉字输入方式标志
4E08:9C32 8BE895	MOV	BX,95E8	: “首尾:”提示信息地址→BX
4E08:9C35 8026BB28FE	AND	BYTE PTR [28BB],FE	
-u			
4E08:9C3A 800EBB2810	OR	BYTE PTR [28BB],10	
4E08:9C3F E88F00	CALL	9CD1	: 清提示行, 写提示信息, 第一键状态→第一键标志
4E08:9C42 E98001	JMP	9DC5	
4E08:9C45 80FC6A	CMP	AH,6A	: Alt+F3(拼音码功能键)?
4E08:9C48 7513	JNZ	9C5D	
4E08:9C4A 8BF295	MOV	BX,95F2	: “拼音:”提示信息→BX
4E08:9C4D EBB100	CALL	9CD1	: 转子程序, 清提示行, 写提示信息, 第一键状态→第一键标志
4E08:9C50 C6063A9604	MOV	BYTE PTR [963A],04	
4E08:9C55 800EBB2B11	OR	BYTE PTR [28BB],11	
-u			
4E08:9C5A E96801	JMP	9DC5	
4E08:9C5D 80FC6B	CMP	AH,6B	: Alt+F4(快速码功能键)?
4E08:9C60 751B	JNZ	9C7A	
4E08:9C62 C6063A9608	MOV	BYTE PTR [963A],08	: 08(快速码)→汉字输入方式标志
4E08:9C67 8026BB28FE	AND	BYTE PTR [28BB],FE	
4E08:9C6C 800EBB2B10	OR	BYTE PTR [28BB],10	
4E08:9C71 8BF795	MOV	BX,95F7	
4E08:9C74 E85A00	CALL	9CD1	: 清提示行, 写提示信息, 第一键状态→第一键标志
4E08:9C77 E94B01	JMP	9DC5	
-u			
4E08:9C7A 80FC6D	CMP	AH,6D	: Alt+F5(ASCII功能键)?
4E08:9C7D 7511	JNZ	9C90	: 不是, 转9C90, ASCII码汉字处理模块
4E08:9C7F C6063A9610	MOV	BYTE PTR [963A],10	: 10H(ASCII码)→汉字输入方式标志
4E08:9C84 8B3B96	MOV	BX,963B	: “ASCII:”提示信息地址→BX
4E08:9C87 890600	MOV	CX,0006	: 提示信息字节数(长度)→CX
4E08:9C8A E84000	CALL	9CCD	: 清提示行, 写提示信息, 第一键状态→第一键标志
4E08:9C8D E93501	JMP	9DC5	: 键盘码处理子程序返回
4E08:9C90 803E9B95FF	CMP	BYTE PTR [959B],FF	: 当前是外码第一键?
4E08:9C95 742C	JZ	9CC3	: 是第一键, 转外码第一键处理入口
4E08:9C97 F6063A9620	TEST	BYTE PTR [963A],20	: 进入词组方式?
-u			
4E08:9C9C 7403	JZ	9CA1	: 是, 转词组输入方式处理
4E08:9C9E E9030B	JMP	A7A4	
4E08:9CA1 3C3B	CMP	AL,3B	: “:” (当前要进入词组方式)?
4E08:9CA3 7503	JNZ	9CA8	
4E08:9CA5 E93309	JMP	A5DB	: 是, 进入词组方式
4E08:9CAB F6063A9601	TEST	BYTE PTR [963A],01	: 当前是区位码方式?
4E08:9CAD 7511	JNZ	9CC0	: 是, 转区位码处理
4E08:9CAF F6063A9602	TEST	BYTE PTR [963A],02	: 当前是首尾码方式?
4E08:9CB4 7510	JNZ	9CC6	: 是, 转首尾码处理
4E08:9CB6 F6063A9604	TEST	BYTE PTR [963A],04	: 当前是拼音码方式?
4E08:9CB8 752E	JNZ	9CEB	: 是, 转拼音码处理

-u

4E08:9CB0 E9C103 JMP A081 : 转A 0 81快速码处理
 4E08:9CC0 E90801 JMP 9DCB : 转区位码处理
 4E08:9CC3 E9C401 JMP 9E8A : 转外码第一键处理入口
 4E08:9CC6 3C00 CMP AL,0D : “←” (回车键)?
 4E08:9CCB 752B JNZ 9CF2 : 不是, 转首尾, 拼音码非第一键处理
 4E08:9CCA E960FF JMP 9C2D : 是, 转首尾码初始化模块
 4E08:9CCD 50 PUSH AX : 保存AX
 4E08:9CCE EB05 JMP 9CD5
 4E08:9CD0 90 NOP
 4E08:9CD1 50 PUSH AX
 4E08:9CD2 B90500 MOV CX,0005 : 5→CX (显示字符字节数)
 4E08:9CD5 C6069C9500 MOV BYTE PTR [959C],00
 4E08:9CDA B80010 MOV AX,1000

-u

4E08:9CDD CD10 INT 10
 4E08:9CDF 32D2 XOR DL,DL
 4E08:9CE1 E86505 CALL A249 : 在提示行显示输入方式提示信息
 4E08:9CE4 800E9B95FF OR BYTE PTR [959B],FF
 4E08:9CE9 5B POP AX
 4E08:9CEA C3 RET
 4E08:9CEB 3C00 CMP AL,0D : “←” (回车键)?
 4E08:9CED 7503 JNZ 9CF2 : 不是, 转首尾, 拼音码非第一键处理
 4E08:9CEF E958FF JMP 9C4A : 是, 转拼音码初始化模块
 4E08:9CF2 3C08 CMP AL,08 : ←” (退格键)?
 4E08:9CF4 7434 JZ 9D2A : 是, 转9 D 2 A 退格键处理
 4E08:9CF6 3C20 CMP AL,20 : “ ” (空格键)?
 4E08:9CF8 7505 JNZ 9CFF
 4E08:9CFA B030 MOV AL,30 : 是, 空格键当作数字“0”, 用以选择重码
 4E08:9CFC EB09 JMP 9D07 : 转9 D 07数字键处理

-u

4E08:9CFE 90 NOP
 4E08:9CFF 3C30 CMP AL,30 : 是数字键 (“0”~“9”)?
 4E08:9D01 724E JB 9D51 : 若不是数字键, 转9 D 51继续判断键盘码
 4E08:9D03 3C39 CMP AL,39
 4E08:9D05 7F4A JG 9D51
 4E08:9D07 240F AND AL,0F : (重码指针) 数字字符转换成数字→AL
 4E08:9D09 3A06D22B CMP AL,[2BD2] : 挑选的重码指针超过重码区重码数?
 4E08:9D0D 7318 JNB 9D27 : 超过, 转挑选重码出错处理
 4E08:9D0F 50 PUSH AX
 4E08:9D10 53 PUSH BX
 4E08:9D11 33DB XOR BX,BX
 4E08:9D13 8ADB MOV BL,AL
 4E08:9D15 D0E3 SHL BL,1
 4E08:9D17 8AB78C2B MOV AL,[BX+2B8C] : 选中的汉字内码 (2字节) →AX
 4E08:9D1B 8AA7BD2B MOV AH,[BX+2B8D]

-u

4E08:9D1F A39D95 MOV [959D],AX : 选中的汉字内码→内码区
 4E08:9D22 5B POP BX

4E08:9D23 58	POP	AX	
4E08:9D24 E93F02	JMP	9F66	: 转 9 F 66, 02→内码计数器, 置第一键标志
4E08:9D27 E99200	JMP	9DBC	: 9 DBC, 挑选重码出错处理
4E08:9D2A E85000	CALL	9D7D	: 提示行设置光标
4E08:9D2D E80502	CALL	9F35	: 清提示行
4E08:9D30 FE0E9C95	DEC	BYTE PTR [959C]	: 外码计数器 - 1
4E08:9D34 7515	JNZ	9D4B	: 若外码计数器 ≠ 0, 转 9 D 4 A 执行
4E08:9D36 C6069C9501	MOV	BYTE PTR [959C],01	: 若外码计数器 = 0, 01→外码计数器
4E08:9D3B E83F00	CALL	9D7D	: 重新设置光标
4E08:9D3E C6069C9500	MOV	BYTE PTR [959C],00	: 0 0 →外码计数器
-u			
4E08:9D43 800E9B95FF	OR	BYTE PTR [959B],FF	: 设置第一键标志
4E08:9D48 EB78	JMP	9DC5	: 转键盘码处理子程序返回
4E08:9D4A 90	NOP		
4E08:9D4B E87202	CALL	9FC0	: 在对照表中查重码设置指针
4E08:9D4E E94F02	JMP	9FA0	: 转 9 F = A 0, 退格后, 重新开始外码查重
4E08:9D51 3C2C	CMP	AL,2C	: “.” (向前翻页)?
4E08:9D53 7436	JZ	9D8B	: 是, 转向前翻页处理
4E08:9D55 3C2E	CMP	AL,2E	: “.” (向后翻页)?
4E08:9D57 7455	JZ	9DAE	: 是, 转向后翻页处理
4E08:9D59 3C61	CMP	AL,61	: 小写字母 “a” ~ “z”?
4E08:9D5B 7C19	JL	9D76	: 若 < “a” 转 9 D 76, 做 “(” 键判断
4E08:9D5D 3C7A	CMP	AL,7A	
4E08:9D5F 7F5B	JG	9DBC	: 若 > “z”, 则为非法外码转 9 DBC 报警
4E08:9D61 803E9C9503	CMP	BYTE PTR [959C],03	: 外码已满 3 键?
-u			
4E08:9D66 7F54	JG	9DBC	: 已满, 第 4 键不应是小写字母, 否则转 9 DBC 报警
4E08:9D68 80069C9501	ADD	BYTE PTR [959C],01	: 不满 3 键, 外码计数器 + 1
4E08:9D6D E80D00	CALL	9D7D	: 调子程序在提示行设置光标
4E08:9D70 E81602	CALL	9FB9	: 外码存入外码区, 提示行显示外码
4E08:9D73 E92702	JMP	9F9D	
4E08:9D76 3C5B	CMP	AL,5B	: “(”?
4E08:9D78 74E7	JZ	9D61	: 是 “(”, 合法外码转回 9 D 61
4E08:9D7A EB40	JMP	9DBC	: 不是 “(”, 非法外码转 9 DBC 报警
4E08:9D7C 90	NOP		
4E08:9D7D B204	MOV	DL,04	: 04→DL
4E08:9D7F 02169C95	ADD	DL,[959C]	: 外码计数器 + 4 →DL (光标置)
4E08:9D83 50	PUSH	AX	
4E08:9D84 B80210	MOV	AX,1002	: 设置功能号
-u			
4E08:9D87 CD10	INT	10	: 10H 中断, 在提示行设置光标
4E08:9D89 58	POP	AX	
4E08:9D8A C3	RET		
4E08:9D8B E87302	CALL	A001	: 为反向查表设置指针
4E08:9D8E E88002	CALL	A011	: 根据外码计数器的值转 1 ~ 5 键查重子程序
4E08:9D91 E85902	CALL	9FED	: 反向查表结束, 为下次查表设置指针
4E08:9D94 E8CD03	CALL	A164	: 提示行显示重码及剩余重码数
4E08:9D97 803ED22800	CMP	BYTE PTR [2BD2],00	: 查到重码了吗?

4E08:9D9C 7503	JNZ	9DA1	: 查到, 转 9DA1
4E08:9D9E EB1C	JMP	9DBC	: 没查到, 转 9DBC报警, 键盘码处理子程序返回
4E08:9DA0 90	NOP		
4E08:9DA1 803ED22B01	CMP	BYTE PTR [2BD2],01	: 只有一个重码?
4E08:9DA6 7503	JNZ	9DAB	: 不只一个, 转 9DAB
-u			
4E08:9DAB E9D401	JMP	9F7F	: 只有一个, 转 9F7F做唯一重码处理
4E08:9DAB EB18	JMP	9DC5	: 因重码不只一个, 需挑选重码, 转 9DC5
4E08:9DAD 90	NOP		
4E08:9DAE EB1F02	CALL	9FD0	: 为正向查表设置指针
4E08:9DB1 EB6D02	CALL	A021	: 根据外码计数器的值转 1~5 键查重子程序
4E08:9DB4 EB2202	CALL	9FD9	: 正向查表结束, 为下次查表设置指针
4E08:9DB7 EBAA03	CALL	A164	: 提示行显示重码及剩余重码数
4E08:9DBA EBDB	JMP	9D97	: 查表结束处理
4E08:9DBC 52	PUSH	DX	
4E08:9DBD B207	MOV	DL,07	: 07→DL
4E08:9DBF 50	PUSH	AX	
4E08:9DC0 EB3809	CALL	A6FB	: 扬声器报警
4E08:9DC3 5B	POP	AX	
4E08:9DC4 5A	POP	DX	
4E08:9DC5 C606D59500	MOV	BYTE PTR [95D5],00	: 0→内码计数器(内码无数据)
-u			
4E08:9DCA C3	RET		
4E08:9DCB 3C00	CMP	AL,00	: “←”(回车键)?
4E08:9DCD 741B	JZ	9DEA	: 是, 经 9DEA 转区位码初始化模块
4E08:9DCF 3C08	CMP	AL,08	: “←”(退格键)?
4E08:9DD1 751A	JNZ	9DED	: 不是, 转 9DED 区位码处理
4E08:9DD3 B208	MOV	DL,08	: 08→DL: 清除字符
4E08:9DD5 EB2309	CALL	A6FB	
4E08:9DD8 FE0E9C95	DEC	BYTE PTR [959C]	: 外码计数器-1
4E08:9DDC 803E9C9500	CMP	BYTE PTR [959C],00	: 退格键处理的是第一键吗?
4E08:9DE1 7505	JNZ	9DE8	: 不是, 转 9DE8
4E08:9DE3 800E9B95FF	OR	BYTE PTR [959B],FF	: 是, 设置第一键标志
4E08:9DE8 EBDB	JMP	9DC5	
-u			
4E08:9DEA E92DFE	JMP	9C1A	
4E08:9DED 3C30	CMP	AL,30	: 数字字符“0”~“9”?
4E08:9DEF 7CCB	JL	9DBC	
4E08:9DF1 3C39	CMP	AL,39	
4E08:9DF3 7FC7	JB	9DBC	
4E08:9DF5 50	PUSH	AX	: 是数字字符, 则保存于 AX
4E08:9DF6 BAD0	MOV	DL,AL	: 区位码外码数字字符→DL
4E08:9DF8 EB0009	CALL	A6FB	
4E08:9DFB 5B	POP	AX	
4E08:9DFC 33DB	XOR	BX,BX	
4E08:9DFE 8A1E9C95	MOV	BL,[959C]	: 外码计数器值→BL:
4E08:9E02 8887B12B	MOV	[BX+2BB1],AL	
4E08:9E06 FEC3	INC	BL	
4E08:9E08 80FB04	CMP	BL,04	: 当前是第 4 键

```

-u
4E08:9E0B 7406      JZ      9E13          ; 是, 转 9 E 13完整区位码处理
4E08:9E0D 8B1E9C95  MOV     [959C],BL
4E08:9E11 EBB2      JMP     9DC5
4E08:9E13 EB1300    CALL    9E29          ; BX返回对应的汉字内码
4E08:9E16 C6069C9500  MOV     BYTE PTR [959C],00
4E08:9E1B 800E9B95FF  OR      BYTE PTR [959B],FF
4E08:9E20 86FB      XCHG   BL,BH
4E08:9E22 891E9D95  MOV     [959D],BX    ; 区位码存内码区
4E08:9E26 E93D01    JMP     9F66
4E08:9E29 BEB12B  MOV     SI,2BB1      ; 取外码区地址→SI
-u
4E08:9E2C 8B04      MOV     AX,[SI]      ; 区号外码→AX
4E08:9E2E EB3800    CALL    9E69          ; 将区号转换成BCD码→AH
4E08:9E31 8AFC      MOV     BH,AH
4E08:9E33 46        INC     SI            ; 调整外码区指针
4E08:9E34 46        INC     SI
4E08:9E35 8B04      MOV     AX,[SI]      ; 位号外码→AX
4E08:9E37 EB2F00    CALL    9E69          ; 将位号转换成BCD码→AH
4E08:9E3A 8ADC      MOV     BL,AH
4E08:9E3C 8AC7      MOV     AL,BH
4E08:9E3E E81000    CALL    9E51          ; BCD码转换成二进制码→AL
4E08:9E41 8AFB      MOV     BH,AL
4E08:9E43 8AC3      MOV     AL,BL
4E08:9E45 E80900    CALL    9E51
4E08:9E48 8AD8      MOV     BL,AL        ; 位号BCD码→BL
4E08:9E4A 81C32020  ADD     BX,2020      ; BX中的区位码转换成国标码
-u
4E08:9E4E EB14      JMP     9E64
4E08:9E50 90        NOP
4E08:9E51 8400      MOV     AH,00        ; AL中的BCD码转换成二进制码→AH
4E08:9E53 B104      MOV     CL,04
4E08:9E55 B606      MOV     DH,06
4E08:9E57 BAD0      MOV     DL,AL        ; +位×16+个位(BCD码)→DL
4E08:9E59 24F0      AND     AL,F0
4E08:9E5B D2E8      SHR     AL,CL
4E08:9E5D F6E6      MUL     DH
4E08:9E5F 2AD0      SUB     DL,AL
4E08:9E61 8AC2      MOV     AL,DL        ; 二进制码→AL
4E08:9E63 C3        RET
4E08:9E64 81CBB080  OR      BX,8080      ; 国标码转换成机内码
4E08:9E68 C3        RET
4E08:9E69 80FC39    CMP     AH,39        ; 若>“9”, 转 9 E 80
4E08:9E6C 7F12      JG      9E80
-u
4E08:9E6E 80EC30    SUB     AH,30        ; 若>“9”, 转 9 E 85
4E08:9E71 3C39      CMF     AL,39
4E08:9E73 7F10      JG      9E85

```

4E08:9E75 2C30	SUB	AL,30	: AL中的数字字符转换成相应的数字
4E08:9E77 B104	MOV	CL,04	
4E08:9E79 D2C0	ROL	AL,CL	
4E08:9E7B 0AE0	OR	AH,AL	
4E08:9E7D E80A	JMP	9E89	: AL高4位→AH高4位
4E08:9E7F 90	NOP		
4E08:9E80 80EC37	SUB	AH,37	
4E08:9E83 EBEC	JMP	9E71	
4E08:9E85 2C37	SUB	AL,37	
4E08:9E87 EBEE	JMP	9E77	
4E08:9E89 C3	RET		
4E08:9E8A 803E3A9610	CMP	BYTE PTR [963A],10	: ASCII码方式?
-u			
4E08:9EBF 7503	JNZ	9E94	
4E08:9E91 E912FD	JMP	98A6	: 是, 转9BA6, ASCII码处理
4E08:9E94 F6063A9620	TEST	BYTE PTR [963A],20	
4E08:9E99 7403	JZ	9E9E	
4E08:9E9B E99D08	JMP	A73B	
4E08:9E9E F6063A9608	TEST	BYTE PTR [963A],0B	: 快速码方式下的第一键?
4E08:9EA3 7403	JZ	9EAB	
4E08:9EA5 E99101	JMP	A039	: 是, 转A039快速码第一键处理
4E08:9EA8 803E9C9500	CMP	BYTE PTR [959C],00	: 外码计数器=0?
4E08:9EAD 7429	JZ	9ED8	
-u			
4E08:9EAF 80FC82	CMP	AH,82	: Alt+ "1" ~ Alt+ "0" ?
4E08:9EB2 7314	JNB	9EC8	
4E08:9EB4 80FC78	CMP	AH,78	
4E08:9EB7 721F	JB	9ED8	
4E08:9EB9 80EC77	SUB	AH,77	: Alt+ "1" ~ Alt+ "0" 转换成数字
4E08:9EBE 80FC0A	CMP	AH,0A	: 转换后的数字=10?
4E08:9EBF 7502	JNZ	9EC3	
4E08:9EC1 32E4	XOR	AH,AH	: 数字10作0处理
4E08:9EC3 BAC4	MOV	AL,AH	
4E08:9EC5 E93FFE	JMP	9D07	: 转9D07, 数字键挑选重码
4E08:9EC8 80FC82	CMP	AH,82	
4E08:9ECB 7503	JNZ	9ED0	
4E08:9ECD E9A100	JMP	9F71	
-u			
4E08:9ED0 80FC83	CMP	AH,83	
4E08:9ED3 7503	JNZ	9ED8	
4E08:9ED5 E9A000	JMP	9F78	
4E08:9ED8 3C30	CMP	AL,30	
4E08:9EDA 7C2A	JL	9F06	: < "0", 不是汉字外码, 作ASCII码处理
4E08:9EDC F6063A9601	TEST	BYTE PTR [963A],01	: 区位码方式下的第一键?
4E08:9EE1 7426	JZ	9F09	
4E08:9EE3 3C39	CMP	AL,39	
4E08:9EE5 7F1F	JG	9F06	: "区位:" 提示信息地址→BX
4E08:9EE7 BBED95	MOV	BX,95ED	
4E08:9EEA E8E4FD	CALL	9CD1	

4E08:9EED A2B12B	MOV	[2BB1],AL	: 区位码第一键外码→外码区
-u			
4E08:9EF0 C6069C9501	MOV	BYTE PTR [959C],01	
4E08:9EF5 8AD0	MOV	DL,AL	
4E08:9EF7 E80108	CALL	A6FB	
4E08:9EFA 80263A96DF	AND	BYTE PTR [963A],DF	
4E08:9EFF F6169B95	NOT	BYTE PTR [959B]	
4E08:9F03 E9BFFE	JMP	9DC5	: 键盘码处理子程序返回
4E08:9F06 E99DFC	JMP	9BA6	: 第一键非汉字外码作ASCII码处理
4E08:9F09 3C61	CMP	AL,61	: 小写字母“a”~“z”?
4E08:9F0B 7CF9	JL	9F06	
4E08:9F0D 3C7A	CMP	AL,7A	
4E08:9F0F 7FF5	JG	9F06	
-u			
4E08:9F11 8D1EE895	LEA	BX,[95EB]	
4E08:9F15 F6063A9602	TEST	BYTE PTR [963A],02	: 是首尾方式?
4E08:9F1A 7504	JNZ	9F20	: 是, 转 9 F 20处理
4E08:9F1C 8D1EF295	LEA	BX,[95F2]	
4E08:9F20 E8AEFD	CALL	9CD1	
4E08:9F23 C6069C9501	MOV	BYTE PTR [959C],01	
4E08:9F2B E852FE	CALL	9D7D	
4E08:9F2B EB28	JMP	9F55	: 转 9 F 55, 首尾, 拼音第一键查重处理入口
4E08:9F2D 90	NOP		
4E08:9F2E B205	MOV	DL,05	: 05→DL (光标位置)
4E08:9F30 B80210	MOV	AX,1002	
-u			
4E08:9F33 CD10	INT	10	
4E08:9F35 26	ES:		
4E08:9F36 803E490006	CMP	BYTE PTR [0049],06	: 640 × 200 图形方式
4E08:9F3B 7406	JZ	9F43	
4E08:9F3D B92300	MOV	CX,0023	
4E08:9F40 EB04	JMP	9F46	
4E08:9F42 90	NOP		
4E08:9F43 B94B00	MOV	CX,004B	
4E08:9F46 52	PUSH	DX	
4E08:9F47 B220	MOV	DL,20	
4E08:9F49 E8AF07	CALL	A6FB	
4E08:9F4C E2FB	LOOP	9F49	
4E08:9F4E 5A	POP	DX	
4E08:9F4F B80210	MOV	AX,1002	: 重新设置光标
4E08:9F52 CD10	INT	10	
-u			
4E08:9F54 C3	RET		
4E08:9F55 E83100	CALL	9F89	: 在提示行显示第一键外码
4E08:9F58 E86500	CALL	9FC0	: 为第一键在对照表查重设置指针
4E08:9F5B E83703	CALL	A295	: 一键查重
4E08:9F5E E87800	CALL	9FD9	: 正向查表结束, 为下次查表设置指针
4E08:9F61 E80002	CALL	A164	: 在提示行显示重码汉字

4E08:9F64 2B94	JMP	9EFA	: 键盘码处理子程序返回
4E08:9F6A 0A069B9502	MOV	BYTE PTR [95D5],02	: 02→内码区计数器
4E08:9F6B 899E4D95FF	OR	BYTE PTR [959B],FF	: 设置第一键标志
4E08:9F70 C3	RET		
4E08:9F71 FA169B75	MOV	BYTE PTR [959B]	: 改变第一键标志
-u			
4E08:9F75 E913FE	JMP	9D8B	
4E08:9F7B F6169B95	NOT	BYTE PTR [959B]	
4E08:9F7D E92FFE	JMP	9DAE	: 向后翻页处理入口
4E08:9F7F 8B0EBC2B	MOV	CX,[2B8C]	: 重码区第一个汉字内码→内码区
4E08:9F83 890E9D95	MOV	[959D],CX	
4E08:9F87 E0DD	JMP	9F66	
4E08:9F89 50	PUSH	AX	
4E08:9F8A 330B	XOR	CX,BX	: 0→BX
4E08:9F8C 8A1E9D95	MOV	BL,[959C]	: 外码键数→BL
4E08:9F90 FE0B	DEC	BL	: 外码键数-1→BL
4E08:9F92 0D97912B	MOV	[BX+2BB1],AL	: 以外码键数为位置指针, 将当前外码送外码区
-u			
4E08:9F93 0A00	MOV	DL,AL	
4E08:9F95 050007	CALL	A6FB	: 提示行显示当前外码
4E08:9F98 92	POP	AX	
4E08:9F9C C3	RET		
4E08:9F9D E93000	CALL	9FD0	: 为开始查重设置各种指针
4E08:9FA0 E87E00	CALL	A021	: 调用1~5键查重子程序
4E08:9FA3 E83300	CALL	9FD9	: 正向查表结束, 为下次查表设置指针
4E08:9FA6 E83901	CALL	A164	: 提示行显示重码汉字及剩余重码数
4E08:9FA9 003ED22B01	CMP	BYTE PTR [2BD2],01	: 重码区只有一个汉字重码?
4E08:9FAE 7500	JNZ	9FD0	: 不只一个, 经9FBD键盘码处理子程序返回
4E08:9FB0 812B8C	MOV	AX,[2B8C]	: 只有一个, 唯一重码→AX
4E08:9FB3 A39D95	MOV	[959D],AX	: 唯一重码→内码区
-u			
4E08:9FB6 C6069B95FF	MOV	BYTE PTR [959B],FF	: 设置第一键标志
4E08:9FB8 EBA9	JMP	9F66	: 02→内码计数器, 键盘码处理子程序返回
4E08:9FBD E905FE	JMP	9DC5	
4E08:9FC0 8D36D62B	LEA	SI,[2BD6]	: 对照表(词表外码区)首址→SI
4E08:9FC4 8936AD2B	MOV	[2BAD],SI	
4E08:9FC8 83C604	ADD	SI,+04	: SI指向第二个表项
4E08:9FCB 8936AF2B	MOV	[2BAF],SI	: 正向查表首指针指向第二个表项
4E08:9FCF C3	RET		
4E08:9FD0 8B364F2B	MOV	SI,[2BAF]	: 取正向查表指针
4E08:9FD4 8936AD2B	MOV	[2BAD],SI	
-u			
4E08:9FD8 C3	RET		
4E08:9FD9 8B36AF2B	MOV	SI,[2BAF]	
4E08:9FDD 83EE04	SUB	SI,+04	
4E08:9FE0 89364996	MOV	[9649],SI	
4E08:9FE4 8B36AD2B	MOV	SI,[2BAD]	
4E08:9FE8 8936AF2B	MOV	[2BAF],SI	
4E08:9FEC C3	RET		

4E08:9FED 8B364996	MOV	SI,[9649]	: 取反向查表首指针
4E08:9FF1 83C604	ADD	SI,+04	
4E08:9FF4 8936AF2B	MOV	[2BAF],SI	
-u			
4E08:9FFB 8B36AD2B	MOV	SI,[2BAD]	
4E08:9FFC 89364996	MOV	[9649],SI	
4E08:A000 C3	RET		
4E08:A001 8B364996	MOV	SI,[9649]	: 取反向查表首指针
4E08:A005 EBCD	JMP	9FD4	
4E08:A007 99	CWD		
4E08:A00B A22BA3	MOV	[A32B],AL	
4E08:A00B 85A3	MOV	CH,A3	
4E08:A00D 53	PUSH	BX	
4E08:A00E A4	MOVSB		
4E08:A00F 0E	PUSH	CS	
4E08:A010 A5	MOVSW		
4E08:A011 EB1900	CALL	A02D	
4E08:A014 8D3ED62B	LEA	DI,[2BD6]	
-d cs:a007			
4E08:A007 99-A2 2B A3 B5 A3 53 A4 0E			."+#5#5#.
4E08:A010 A5 EB 19 00 8D 3E D6 2B-B3 EF 04 FD FF 94 07 A0			Xh...>V+.o.)...
4E08:A020 C3 EB 09 00 8D 3E 96 95-FF 94 07 A0 C3 A0 9C 95			Ch...). C ..
4E08:A030 32 E4 FE C8 D1 E0 8B F0-C3 80 FC B2 73 0B 80 FC			2d*HQ'.pC.l.s..i
4E08:A040 78 72 03 E9 62 FE EB 0B-00 73 03 E9 77 FD E9 55			xr.ib*h..s.iw)iU
4E08:A050 FB 3C 61 72 2A 3C 7A 7F-26 8D 1E F7 95 EB 71 FC			(<ar*(z.&...w.hq!
4E08:A060 A2 B1 2B 80 26 3A 96 DF-C6 06 9B 95 00 C6 06 9C			"l+.&:_F....F..
4E08:A070 95 01 50 EB 8B FE 5B EB-03 FD EB 0C FF F9 C3 FB			..PhB*Xh.)h..yCx
4E08:A080 C3 3C 20 75 1F 80 3E			C< u..>
-ucs:a011			
4E08:A011 EB1900	CALL	A02D	
4E08:A014 8D3ED62B	LEA	DI,[2BD6]	
4E08:A018 83EF04	SUB	DI,+04	
4E08:A01B FD	STD		
4E08:A01C FF9407A0	CALL	[SI+A007]	
4E08:A020 C3	RET		
4E08:A021 EB0900	CALL	A02D	: 在SI中返回与外码键数相应的值
4E08:A024 8D3E9695	LEA	DI,[9596]	: 为正向查表, 将查表结束指针指向对照表尾
4E08:A028 FF9407A0	CALL	[SI+A007]	
4E08:A02C C3	RET		
4E08:A02D A09C95	MOV	AL,[959C]	: (外码计数器-1) × 2 → SI
4E08:A030 32E4	XOR	AH,AH	
-u			
4E08:A032 FECB	DEC	AL	
4E08:A034 D1E0	SHL	AX,1	
4E08:A036 BBF0	MOV	SI,AX	
4E08:A038 C3	RET		
4E08:A039 80FCB2	CMP	AH,B2	: Alt + "1" ~ Alt + "0" ?
4E08:A03C 7308	JNB	A046	

4E08:A03E 80FC78	CMP	AH,78	
4E08:A041 7203	JB	A046	
4E08:A043 E962FE	JMP	9EAB	
4E08:A046 E80B00	CALL	A051	: 转快速码处理子程序
4E08:A049 7303	JNB	A04E	
4E08:A04B E977FD	JMP	9DC5	
4E08:A04E E955FB	JMP	9BA6	: 非快速码合法外码作ASCII码处理
4E08:A051 3C61	CMP	AL,61	: 小写字母“a”~“z”?
-u			
4E08:A053 722A	JB	A07F	
4E08:A055 3C7A	CMP	AL,7A	
4E08:A057 7F26	JG	A07F	
4E08:A059 8D1EF795	LEA	BX,[95F7]	
4E08:A05D E871FC	CALL	9CD1	: 清提示行, 显示“快速:”提示信息
4E08:A060 A2B12B	MOV	[2BB1],AL	
4E08:A063 80263A96DF	AND	BYTE PTR [963A],DF	
4E08:A068 C6069B9500	MOV	BYTE PTR [959B],00	
4E08:A06D C6069C9501	MOV	BYTE PTR [959C],01	
4E08:A072 50	PUSH	AX	
-u			
4E08:A073 EBB8FE	CALL	9F2E	: 清提示信息后的提示行
4E08:A076 58	POP	AX	
4E08:A077 E803FD	CALL	9D7D	: 提示行设置光标。
4E08:A07A E80CFF	CALL	9F89	: 外码存外区, 提示行显示外码
4E08:A07D F9	STC		: CF = 1, 快速码外码已存入外码区
4E08:A07E C3	RET		
4E08:A07F F8	CLC		: CF = 0, 非合法外码
4E08:A080 C3	RET		
4E08:A0B1 3C20	CMP	AL,20	
4E08:A0B3 751F	JNZ	A0A4	
4E08:A0B5 803E9C9504	CMP	BYTE PTR [959C],04	: 外码已满回键?
4E08:A0BA 7418	JZ	A0A4	
4E08:A0BC E831FF	CALL	9FC0	
4E08:A0BF E88FFF	CALL	A021	
4E08:A092 803ED22B00	CMP	BYTE PTR [2BD2],00	: 查到重码了吗?
-u			
4E08:A097 7503	JNZ	A09C	: 有重码接A09C, 继续执行
4E08:A099 E9C000	JMP	A15C	
4E08:A09C C606D22B01	MOV	BYTE PTR [2BD2],01	
4E08:A0A1 E9DBFE	JMP	9F7F	: 转9F7F, 唯一重码处理
4E08:A0A4 3C08	CMP	AL,08	: “←”(退格键)?
4E08:A0A6 750E	JNZ	A0B6	
4E08:A0AB FE069C95	INC	BYTE PTR [959C]	: 是退格键, 为设置光标, 外码计数器+1
4E08:A0AC E8CEFC	CALL	9D7D	: 设置光标
4E08:A0AF FE0E9C95	DEC	BYTE PTR [959C]	
4E08:A0B3 E91DFD	JMP	9DD3	: 退格键处理
4E08:A0B6 3C0D	CMP	AL,0D	
-u			

4E0B:A0B8 7503	JNZ	A0BD	
4E0B:A0BA E9A5FB	JMP	9C62	; 回车键, 转9C26快速码初始化
4E0B:A0BD 803E9C9500	CMP	BYTE PTR [959C],00	; 外码计数器 = 0?
4E0B:A0C2 7404	JZ	A0CB	
4E0B:A0C4 3C5B	CMP	AL,5B	; " (" 键
4E0B:A0C6 740B	JZ	A0D0	; 是, 接A0D0继续处理
4E0B:A0C8 3C61	CMP	AL,61	; 小写字母 "a" ~ "z" ?
4E0B:A0CA 723C	JB	A108	
4E0B:A0CC 3C7A	CMP	AL,7A	
4E0B:A0CE 7F38	JG	A108	
4E0B:A0D0 803E9C9504	CMP	BYTE PTR [959C],04	; 外码已满回键?
4E0B:A0D5 7503	JNZ	A0DA	
4E0B:A0D7 EB4B	JMP	A124	; 满回键, 转A124 第五键快速处理
-u			
4E0B:A0D9 90	NOP		
4E0B:A0DA FE069C95	INC	BYTE PTR [959C]	; 外码存外码区, 提示行显示外码
4E0B:A0DE EBA8FE	CALL	9F89	
4E0B:A0E1 803E9C9504	CMP	BYTE PTR [959C],04	
4E0B:A0E6 7403	JZ	A0EB	
4E0B:A0E8 E9DAFC	JMP	9DC5	
4E0B:A0EB EBD2FE	CALL	9FC0	; 为开始查重码设置指针
4E0B:A0EE E830FF	CALL	A021	; 对照表查重码
4E0B:A0F1 803ED22B00	CMP	BYTE PTR [2BD2],00	; 查到重码了吗?
4E0B:A0F6 7503	JNZ	A0FB	; 查到, 接A0FB继续执行
4E0B:A0FB EB62	JMP	A15C	
-u			
4E0B:A0FA 90	NOP		
4E0B:A0FB 803ED22B01	CMP	BYTE PTR [2BD2],01	; 只有一个重码吗?
4E0B:A100 749A	JZ	A09C	
4E0B:A102 E85F00	CALL	A164	; 不只一个, 提示行显示重码
4E0B:A105 E9BDFC	JMP	9DC5	
4E0B:A108 803E9C9504	CMP	BYTE PTR [959C],04	
4E0B:A10D 7512	JNZ	A121	
4E0B:A10F 8A0ED22B	MOV	CL,[2BD2]	
4E0B:A113 80C130	ADD	CL,30	; 转换成数字字符→CL
4E0B:A116 3C30	CMP	AL,30	
4E0B:A118 720A	JB	A124	
-u			
4E0B:A11A 3AC8	CMP	CL,AL	; 键盘码 > 重码区重码区数?
4E0B:A11C 7206	JB	A124	
4E0B:A11E E9E6FB	JMP	9D07	; 在重码区重码数范围内, 转数字键挑重码区
4E0B:A121 EB39	JMP	A15C	
4E0B:A123 90	NOP		
4E0B:A124 880EBC2B	MOV	CX,[2B8C]	
4E0B:A128 B90E9D95	MOV	[959D],CX	
4E0B:A12C C6069B95FF	MOV	BYTE PTR [959B],FF	
4E0B:A131 E810FF	CALL	A051	
4E0B:A134 C606D59502	MOV	BYTE PTR [95D5],02	
4E0B:A139 7219	JB	A154	

```

-u
4E08:A13B 8D3E9D95 LEA DI,[959D]
4E08:A13F 83C702 ADD DI,+02
4E08:A142 803EAC2BFF CMP BYTE PTR [2BAC],FF
4E08:A147 7408 JZ A151
4E08:A149 2E CS:
4E08:A14A 8B05 MOV [DI],AL
4E08:A14C FE06D595 INC BYTE PTR [95D5]
4E08:A150 C3 RET
4E08:A151 EBE107 CALL A935
4E08:A154 803ED22B00 CMP BYTE PTR [2BD2],00
4E08:A159 7401 JZ A15C

-u
4E08:A15B C3 RET
4E08:A15C B207 MOV DL,07
4E08:A15E EB9A05 CALL A6FB : 调子程序, 报警
4E08:A161 E961FC JMP 9DC5
4E08:A164 B205 MOV DL,05
4E08:A166 EB16FC CALL 9D7F : 提示行设置光标
4E08:A169 EBC9FD CALL 9F35
4E08:A16C 33D2 XOR DX,DX
4E08:A16E 33DB XOR BX,BX
4E08:A170 26 ES: : ES = 0040
4E08:A171 B03E490004 CMP BYTE PTR [0049],04 : 640 × 200 图形方式, 接A 180 执行
4E08:A176 7508 JNZ A180
4E08:A178 B90500 MOV CX,0005

-u
4E08:A17B B20B MOV DL,0B
4E08:A17D EB06 JMP A185
4E08:A17F 90 NOP
4E08:A180 B90A00 MOV CX,000A
4E08:A183 B210 MOV DL,10
4E08:A185 3B0E9795 CMP CX,[9597] : 重码数超过提示行容量
4E08:A189 7307 JNB A192 : 没有超过, 接A 192 继续执行
4E08:A18B 290E9795 SUB [9597],CX
4E08:A18F EB07 JMP A19B
4E08:A191 90 NOP
4E08:A192 C70697950000 MOV WORD PTR [9597],0000
4E08:A198 B80210 MOV AX,1002 : 设置功能号, 提示行设置光标

-u
4E08:A19B CD10 INT 10
4E08:A19D 8A0ED22B MOV CL,[2BD2]
4E08:A1A1 32ED XOR CH,CH
4E08:A1A3 80F900 CMP CL,00 : 重码区有重码吗?
4E08:A1A6 7508 JNZ A1B0 : 有, 转A 1 B 0 提示行显示重码
4E08:A1A8 B207 MOV DL,07
4E08:A1AA EB4E05 CALL A6FB
4E08:A1AD EB27 JMP A1D6
4E08:A1AF 90 NOP

```

4E08:A1B0 B230	MOV	DL,30	
4E08:A1B2 52	PUSH	DX	
4E08:A1B3 E84505	CALL	A6FB	: 显示DL中的数字字符(重码号)
4E08:A1B6 B23A	MOV	DL,3A	: “.” → DL
4E08:A1B8 E84005	CALL	A6FB	: 数字后显示冒号(“.”)
-u			
4E08:A1BB 8A97BC2B	MOV	DL,[BX+2B8C]	
4E08:A1BF E83905	CALL	A6FB	: 显示汉字
4E08:A1C2 8A97BD2B	MOV	DL,[BX+2BB0]	
4E08:A1C6 E83205	CALL	A6FB	: 显示一个完整的汉字
4E08:A1C9 B220	MOV	DL,20	: “ ”(空格) → DL; 汉字显示后 全空格
4E08:A1CB E82D05	CALL	A6FB	
4E08:A1CE 5A	POP	DX	
4E08:A1CF FEC2	INC	DL	
4E08:A1D1 83C302	ADD	BX,+02	
4E08:A1D4 E2DC	LOOP	A1B2	
4E08:A1D6 B20B	MOV	DL,0B	
4E08:A1D8 E82005	CALL	A6FB	
-u			
4E08:A1DB B25B	MOV	DL,5B	: “(” → DL
4E08:A1DD E81B05	CALL	A6FB	
4E08:A1E0 8B0E9795	MOV	CX,[9597]	
4E08:A1E4 E80600	CALL	A1ED	
4E08:A1E7 B25D	MOV	DL,5D	: “)” → DL
4E08:A1E9 E80F05	CALL	A6FB	: 调A6FB子程序,显示“)”
4E08:A1EC C3	RET		
4E08:A1ED 51	PUSH	CX	
4E08:A1EE 52	PUSH	DX	
4E08:A1EF 53	PUSH	BX	
4E08:A1F0 50	PUSH	AX	
4E08:A1F1 E80500	CALL	A1F9	: 显示剩余重码数
4E08:A1F4 5B	POP	AX	
4E08:A1F5 5B	POP	BX	
4E08:A1F6 5A	POP	DX	
4E08:A1F7 59	POP	CX	
4E08:A1F8 C3	RET		
4E08:A1F9 33C0	XOR	AX,AX	
-u			
4E08:A1FB 8BD0	MOV	DX,AX	
4E08:A1FD 8BD9	MOV	BX,CX	
4E08:A1FF B91000	MOV	CX,0010	: 10H(16, BX的字节数) → CX
4E08:A202 D1E3	SHL	BX,1	
4E08:A204 92	XCHG	DX,AX	
4E08:A205 E83200	CALL	A23A	: 剩余重码数转换成BCD码 BCD码 → AX
4E08:A208 92	XCHG	DX,AX	
4E08:A209 E2F7	LODP	A202	: 转回A 202,直到BX(16位)全部移进CF
4E08:A20B B9101B	MOV	CX,1B10	
4E08:A20E 52	PUSH	DX	
4E08:A20F 8AD6	MOV	DL,DH	

4E08:A211 EB1000	CALL	A224	: 提示行显示剩余重码百位
4E08:A214 5A	POP	DX	
4E08:A215 8AF2	MOV	DH,DL	
4E08:A217 D0EA	SHR	DL,1	
4E08:A219 D0EA	SHR	DL,1	
-u			
4E08:A21B D0EA	SHR	DL,1	
4E08:A21D D0EA	SHR	DL,1	
4E08:A21F EB0200	CALL	A224	: 提示行显示剩余重码十位
4E08:A222 8AD6	MOV	DL,DH	
4E08:A224 80E20F	AND	DL,0F	: 屏蔽DL高4位
4E08:A227 7402	JZ	A22B	
4E08:A229 B100	MOV	CL,00	
4E08:A22B FECD	DEC	CH	
4E08:A22D 22CD	AND	CL,CH	
4E08:A22F 80CA30	OR	DL,30	: DL低4位转换成相应数字字符
4E08:A232 50	PUSH	AX	
4E08:A233 52	PUSH	DX	
4E08:A234 EB0E00	CALL	A245	: 提示行显示DL中的数字
4E08:A237 5A	POP	DX	
4E08:A238 5B	POP	AX	
4E08:A239 C3	RET		
4E08:A23A 12C0	ADC	AL,AL	: 进位加 (AL + AL + CF → AL)
-u			
4E08:A23C 27	DAA		
4E08:A23D 86C4	XCHG	AH,AL	
4E08:A23F 12C0	ADC	AL,AL	
4E08:A241 27	DAA		: 十进制调整, 得到千位、百位BCD码→AL
4E08:A242 86C4	XCHG	AH,AL	
4E08:A244 C3	RET		
4E08:A245 EBB304	CALL	A6FB	: 提示行显示DL中的字符
4E08:A248 C3	RET		
4E08:A249 B80210	MOV	AX,1002	: 设置功能号, 提示行设置光标
4E08:A24C CD10	INT	10	
4E08:A24E 8A17	MOV	DL,[BX]	: BX指针下的字符(汉字)→DL
4E08:A250 EBA804	CALL	A6FB	: 提示行显示DL中的字符
4E08:A253 43	INC	BX	
4E08:A254 E2FB	LOOP	A24E	
4E08:A256 C3	RET		
4E08:A257 50	PUSH	AX	
4E08:A258 52	PUSH	DX	
4E08:A259 56	PUSH	SI	
4E08:A25A 53	PUSH	BX	
4E08:A25B 8A1E022B	MOV	BL,[28D2]	
-u			
4E08:A25F 3A1E02B	CMP	BL,[28D0]	: 重码区已满?
4E08:A263 7324	JNB	A289	
4E08:A265 BBC6	MOV	AX,SI	: 当前查表指针→AX
4E08:A267 2DD62B	SUB	AX,28D6	: 当前指针下的表项在对照表的位置→AX

4E08:A26A D1E8	SHR	AX,1	
4E08:A26C D1E8	SHR	AX,1	
4E08:A26E B25E	MOV	DL,5E	: 5 EH (94) → DL (94个区, 94个位)
4E08:A270 F6F2	DIV	DL	: 转换当前表项为对应的汉字国标码
4E08:A272 053021	ADD	AX,2130	
4E08:A275 0D80B0	OR	AX,80B0	: 国标码转换成汉字内码→AX
4E08:A278 32FF	XOR	BH,BH	
4E08:A27A D0E3	SHL	BL,1	
4E08:A27C 8987BC2B	MOV	[BX+2BBC],AX	: 将汉字内码存入BX为位置指针的重码区
-u			
4E08:A280 D0EB	SHR	BL,1	
4E08:A282 FEC3	INC	BL	
4E08:A284 881ED22B	MOV	[2BD2],BL	
4E08:A288 F9	STC		
4E08:A289 5B	POF	BX	
4E08:A28A 5E	POP	SI	
4E08:A28B 5A	POP	DX	
4E08:A28C 58	POP	AX	
4E08:A28D C3	RET		
4E08:A28E BFD62B	MOV	DI,2BD6	? 对照表首址→查表结束指针
4E08:A291 FD	STD		: 设置方向标志, 反向查表
4E08:A292 EB05	JMP	A299	
4E08:A294 90	NOF		
4E08:A295 BF9695	MOV	DI,9596	: 对照表尾指针→查表结束指针
4E08:A298 FC	CLD		: 清除方向标志, 正向查表
4E08:A299 8B36AD2B	MOV	SI,[2BAD]	: 当前/下次查表指针→查表指针(SI)
4E08:A29D 8026BB2BDF	AND	BYTE PTR [2BBB],DF	
-u			
4E08:A2A2 33C0	XOR	AX,AX	
4E08:A2A4 A2D22B	MOV	[2BD2],AL	
4E08:A2A7 A39795	MOV	[9597],AX	
4E08:A2AA 33C9	XOR	CX,CX	
4E08:A2AC BA1EB12B	MOV	BL,[2BB1]	: 取外码区一键外码→BL
4E08:A2B0 80E31F	AND	BL,1F	
4E08:A2B3 3BF7	CMP	SI,DI	: 查表指针=查表结束指针?
4E08:A2B5 741C	JZ	A2D3	
4E08:A2B7 8A04	MOV	AL,[SI]	: 取查表指针(SI)下的对照表表项首尾码第一个字符→AL
4E08:A2B9 241F	AND	AL,1F	
4E08:A2BB F606BB2B01	TEST	BYTE PTR [2BBB],01	: 是拼音方式吗?
4E08:A2C0 7409	JZ	A2CB	
-u			
4E08:A2C2 8A4401	MOV	AL,[SI+01]	: 拼音方式, 取对照表拼音码第一个字符→AL
4E08:A2C5 247C	AND	AL,7C	
4E08:A2C7 D0EB	SHR	AL,1	
4E08:A2C9 D0EB	SHR	AL,1	
4E08:A2CB 3AC3	CMP	AL,BL	
4E08:A2CD 741C	JZ	A2EB	
4E08:A2CF AD	LODSW		: 根据方向标志, 调查表指针(SI)

4E08:A2D0 AD	LODSW		
4E08:A2D1 EBEO	JMP	A2B3	: 转回A 2 B 3, 继续查找
4E08:A2D3 0BC9	OR	CX,CX	
4E08:A2D5 7411	JZ	A2E8	
4E08:A2D7 890E9795	MOV	[9597],CX	
4E08:A2D8 3B0ED02B	CMP	CX,[2BD0]	
4E08:A2DF 7704	JA	A2E5	
4E08:A2E1 893EAD2B	MOV	[2BAD],DI	
-u			
4E08:A2E5 F9	STC		: CF = 1, 以示有重码
4E08:A2E6 FC	CLD		
4E08:A2E7 C3	RET		
4E08:A2E8 FB	CLC		
4E08:A2E9 FC	CLD		
4E08:A2EA C3	RET		
4E08:A2EB F606BB2B10	TEST	BYTE PTR [2BBB],10	
4E08:A2F0 7516	JNZ	A308	
4E08:A2F2 F606BB2B01	TEST	BYTE PTR [2BBB],01	
4E08:A2F7 7409	JZ	A302	
4E08:A2F9 F6440340	TEST	BYTE PTR [SI+03],40	
4E08:A2FD 7509	JNZ	A308	
4E08:A2FF EB1C	JMP	A31D	
4E08:A301 90	NOP		
4E08:A302 F6440380	TEST	BYTE PTR [SI+03],80	
-u			
4E08:A306 7415	JZ	A31D	
4E08:A308 E84CFF	CALL	A257	: 查到的表项转换成汉字内码
4E08:A30B 7210	JB	A31D	
4E08:A30D F606BB2B20	TEST	BYTE PTR [2BBB],20	
4E08:A312 7509	JNZ	A31D	
4E08:A314 8936AD2B	MOV	[2BAD],SI	
4E08:A318 800EBB2B20	OR	BYTE PTR [2BBB],20	
4E08:A31D 41	INC	CX	
4E08:A31E EBAF	JMP	A2CF	: 调整查表指针继续查找
4E08:A320 BFD62B	MOV	DI,2BD6	: 对照表首址→查表结束指针
4E08:A323 FD	STD		
4E08:A324 EB05	JMP	A32B	: 二键查重子程序
-u			
4E08:A326 90	NOP		
4E08:A327 BF9695	MOV	DI,9596	: 对照表尾指针→查表结束指针
4E08:A32A FC	CLD		: 清除方向标志, 正向查表
4E08:A32B 8B36AD2B	MOV	SI,[2BAD]	: 当前/下次查表指针→查表指针 (SI)
4E08:A32F 8026BB2BDF	AND	BYTE PTR [2BBB],DF	
4E08:A334 33C0	XOR	AX,AX	
4E08:A336 A2D22B	MOV	[2BD2],AL	: 0→重码区重码数
4E08:A339 A39795	MOV	[9597],AX	
4E08:A33C 33C9	XOR	CX,CX	
4E08:A33E 8B1EB12B	MOV	BX,[2BB1]	: 取外码区=键外码→BX

4E08:A342 80E31F	AND	BL,1F	: BL中的外码字母转换成数字 (1~26)
4E08:A345 80E71F	AND	BH,1F	
-u			
4E08:A348 50	PUSH	AX	
4E08:A349 8AE7	MOV	AH,BH	
4E08:A34B 32C0	XOR	AL,AL	
4E08:A34D D1EB	SHR	AX,1	
4E08:A34F D1EB	SHR	AX,1	
4E08:A351 D1EB	SHR	AX,1	
4E08:A353 8AFC	MOV	BH,AH	
4E08:A355 0ADB	OR	BL,AL	
4E08:A357 58	POP	AX	
4E08:A358 3BF7	CMP	SI,DI	
4E08:A35A 741E	JZ	A37A	
4E08:A35C 8B04	MOV	AX,[SI]	
4E08:A35E 25FF03	AND	AX,03FF	
4E08:A361 F606BB2B01	TEST	BYTE PTR [2BBB],01	: 是拼音方式吗?
4E08:A366 740A	JZ	A372	
-u			
4E08:A368 8B4401	MOV	AX,[SI+01]	: 拼音方式, 表项第2、3字节→AX
4E08:A36B 25FC0F	AND	AX,0FFC	
4E08:A36E D1EB	SHR	AX,1	
4E08:A370 D1EB	SHR	AX,1	
4E08:A372 3BC3	CMP	AX,BX	
4E08:A374 741C	JZ	A392	
4E08:A376 AD	LODSW		: 根据方向标志, 调整查表指针 (SI)
4E08:A377 AD	LODSW		
4E08:A378 EBDE	JMP	A35B	
4E08:A37A 0BC9	OR	CX,CX	
4E08:A37C 7411	JZ	A3BF	
4E08:A37E 890E9795	MOV	[9397],CX	
4E08:A382 3B0ED02B	CMC	CX,[2BD0]	
4E08:A386 7704	JA	A3BC	
-u			
4E08:A388 873EAD2B	MOV	[2BAD],DI	
4E08:A38C FC	CLD		
4E08:A38D F9	STC		
4E08:A38E C3	RET		
4E08:A38F FB	CLC		
4E08:A390 FC	CLD		
4E08:A391 C3	RET		
4E08:A392 E8C2FE	CALL	A257	: 查到的表项转换成汉字内码
4E08:A395 7210	JB	A3A7	: 若重码已存重码区转A 3 A 7 执行
4E08:A397 F606BB2B20	TEST	BYTE PTR [2BBB],20	
4E08:A39C 7509	JNZ	A3A7	
4E08:A39E 8936AD2B	MOV	[2BAD],SI	
4E08:A3A2 800EBB2B20	OR	BYTE PTR [2BBB],20	
4E08:A3A7 41	INC	CX	

-u

4E08:A3A8 EBCC	JMP	A376	
4E08:A3AA BFD62B	MOV	DI,2BD6	: 对照表首址→查表结束指针
4E08:A3AD FD	STD		: 设置方向标志, 反向查表
4E08:A3AE EB05	JMP	A3B5	: 转A 3 B 5 三键检查子程序
4E08:A3B0 90	NOF		
4E08:A3B1 BF9695	MOV	DI,9596	
4E08:A3B4 FC	CLD		
4E08:A3B5 8B36AD2B	MOV	SI,[2BAD]	
4E08:A3B9 8026BB2BDF	AND	BYTE PTR [2BBB],DF	
4E08:A3BE 33C0	XOR	AX,AX	
4E08:A3C0 A2D22B	MOV	[2BD2],AL	
4E08:A3C3 A39795	MOV	[9597],AX	
4E08:A3C6 33C9	XOR	CX,CX	

-u

4E08:A3C8 8B1EB12B	MOV	BX,[2BB1]	: 取外码在 1、2 键外码→BX
4E08:A3CC 80E31F	AND	BL,1F	: BL 中的外码字母转换成数字 (1~26)
4E08:A3CF 80E71F	AND	BH,1F	: BH 中的外码字母转换成数字 (1~26)
4E08:A3D2 50	PUSH	AX	
4E08:A3D3 BAE7	MOV	AH,BH	
4E08:A3D5 32C0	XOR	AL,AL	
4E08:A3D7 D1EB	SHR	AX,1	
4E08:A3D9 D1EB	SHR	AX,1	
4E08:A3DB D1EB	SHR	AX,1	
4E08:A3DD 8AFC	MOV	BH,AH	
4E08:A3DF 0AD8	OR	BL,AL	
4E08:A3E1 A0B32B	MOV	AL,[2BB3]	
4E08:A3E4 241F	AND	AL,1F	
4E08:A3E6 D0E0	SHL	AL,1	

-u

4E08:A3EB D0E0	SHL	AL,1	
4E08:A3EA 0AFB	OR	BH,AL	
4E08:A3EC 58	POP	AX	
4E08:A3ED 3BF7	CMP	SI,DI	
4E08:A3EF 7427	JZ	A41B	
4E08:A3F1 8B04	MOV	AX,[SI]	
4E08:A3F3 25FF7F	AND	AX,7FFF	
4E08:A3F6 F606BB2B01	TEST	BYTE PTR [2BBB],01	: 拼音方式?
4E08:A3FB 7413	JZ	A410	
4E08:A3FD 8B4401	MOV	AX,[SI+01]	: 拼音方式, 取表项 2、3 字节→AX
4E08:A400 25FCFF	AND	AX,FFFC	: 屏蔽与拼音码 1、2、3 字符无关位
4E08:A403 D1EB	SHR	AX,1	
4E08:A405 D1EB	SHR	AX,1	
4E08:A407 F6440301	TEST	BYTE PTR [SI+03],01	

-u

4E08:A40B 7403	JZ	A410	
4E08:A40D 80CC40	OR	AH,40	
4E08:A410 3BC3	CMP	AX,BX	
4E08:A412 741C	JZ	A430	

4E08:A414 AD	LODSM		
4E08:A415 AD	LODSM		
4E08:A416 EBD5	JMP	A3ED	: 转回A 3 ED继续查找重码
4E08:A418 0BC9	OR	CX,CX	
4E08:A41A 7411	JZ	A42D	
4E08:A41C 890E9795	MOV	[9597],CX	
4E08:A420 3B0ED02B	CMP	CX,[2BD0]	: 重码数超过重码容量?
4E08:A424 7704	JA	A42A	
4E08:A426 893EAD2B	MOV	[2BAD],DI	
4E08:A42A F9	STC		
-u			
4E08:A42B FC	CLD		
4E08:A42C C3	RET		
4E08:A42D FB	CLC		
4E08:A42E FC	CLD		
4E08:A42F C3	RET		
4E08:A430 E824FE	CALL	A257	: 查到的表项转换成汉字内码
4E08:A433 7210	JB	A445	: 若重码已存重码区转A 4 45
4E08:A435 F606BB2B20	TEST	BYTE PTR [28BB],20	
4E08:A43A 7509	JNZ	A445	
4E08:A43C 8936AD2B	MOV	[2BAD],SI	
4E08:A440 800EBB2B20	OR	BYTE PTR [28BB],20	
4E08:A445 41	INC	CX	
4E08:A446 EBCC	JMP	A414	: 转回A 414 继续三键查重
4E08:A448 8FD62B	MOV	DI,28D6	
-u			
4E08:A44B FD	STD		: 设置方向标志, 反向查表
4E08:A44C EB05	JMP	A453	: 转A 453 回键查重子程序入口
4E08:A44E 90	NOP		
4E08:A44F BF9695	MOV	DI,9596	
4E08:A452 FC	CLD		
4E08:A453 8B36AD2B	MOV	SI,[2BAD]	: 当前/下次查表指针→查表指针
4E08:A457 8026BB2BDF	AND	BYTE PTR [28BB],DF	: 设置剩余重码记录开关位
4E08:A45C 33C0	XOR	AX,AX	
4E08:A45E A2D22B	MOV	[2BD2],AL	
4E08:A461 A39795	MOV	[9597],AX	
4E08:A464 33C9	XOR	CX,CL	
4E08:A466 8B1EB12B	MOV	BX,[2BB1]	: 取外码区1、2键外码→BX
4E08:A46A 80E31F	AND	BL,1F	
-u			
4E08:A46D 80E71F	AND	BH,1F	
4E08:A470 50	PUSH	AX	
4E08:A471 8AE7	MOV	AH,BH	
4E08:A473 32C0	XOR	AL,AL	
4E08:A475 D1EB	SHR	AX,1	
4E08:A477 D1EB	SHR	AX,1	
4E08:A479 D1EB	SHR	AX,1	
4E08:A47B 8AFC	MOV	BH,AH	
4E08:A47D 0ADB	OR	BL,AL	: 取第3、4键外码→AX

4E0B:A47F A1B32B	MOV	AX,[2BB3]	
4E0B:A482 251F1F	AND	AX,1F1F	
4E0B:A485 D0E0	SHL	AL,1	
4E0B:A487 D0E0	SHL	AL,1	
4E0B:A489 0AF8	OR	BH,AL	
4E0B:A48B D0EC	SHR	AH,1	
-u			
4E0B:A48D 7303	JNB	A492	
4E0B:A48F 80CF80	OR	BH,80	
4E0B:A492 8AF4	MOV	DH,AH	
4E0B:A494 58	POP	AX	
4E0B:A495 3BF7	CMP	SI,DI	
4E0B:A497 743A	JZ	A4D3	
4E0B:A499 8B04	MOV	AX,[SI]	
4E0B:A49B 8A5402	MOV	DL,[SI+02]	
4E0B:A49E 80E20F	AND	DL,0F	: 屏蔽与首尾码 1、2 字符拼音码 1、2 字符无关位
4E0B:A4A1 F606BB2B01	TEST	BYTE PTR [2BBB],01	: 拼音方式?
4E0B:A4A6 741F	JZ	A4C7	
4E0B:A4AB 8B4401	MOV	AX,[SI+01]	: 拼音方式, 取表项 2、3 字节→AX
4E0B:A4AB 25FCFF	AND	AX,FFFC	
-u			
4E0B:A4AE D1E8	SHR	AX,1	
4E0B:A4B0 D1E8	SHR	AX,1	
4E0B:A4B2 F6440301	TEST	BYTE PTR [SI+03],01	: 拼音码第 3 字符 4 位 = 1?
4E0B:A4B6 7403	JZ	A4BB	
4E0B:A4B8 80CC40	OR	AH,40	: = 1, 第 4 位的 1 填 A × 第 14 位
4E0B:A4BB 8A14	MOV	DL,[SI]	: 表项第 1 字节→DL
4E0B:A4BD 80E21F	AND	DL,1F	: 首尾码第 1 字符作为第 4 字符
4E0B:A4C0 D0EA	SHR	DL,1	
4E0B:A4C2 7303	JNB	A4C7	没查到重码转 A 4 C 7
4E0B:A4C4 80CC80	OR	AH,80	
4E0B:A4C7 3BC3	CMP	AX,BX	: AX = BX?
4E0B:A4C9 7504	JNZ	A4CF	
4E0B:A4CB 3AF2	CMP	DH,DL	
4E0B:A4CD 741C	JZ	A4EB	
-u			
4E0B:A4CF AD	LODSW		: 依方向标志调整查表指针
4E0B:A4D0 AD	LODSW		
4E0B:A4D1 EBC2	JMP	A495	: 继续查
4E0B:A4D3 08C9	OR	CX,CX	
4E0B:A4D5 7411	JZ	A4EB	
4E0B:A4D7 890E9795	MOV	[9597],CX	
4E0B:A4DB 380ED02B	CMP	CX,[2BD0]	: 重码数超过重码区容量?
4E0B:A4DF 7704	JA	A4E5	
4E0B:A4E1 893EAD2B	MOV	[2BAD],DI	: 没有, 调整指针
4E0B:A4E5 F9	STC		: CF = 1, 有重码
4E0B:A4E6 FC	CLD		
4E0B:A4E7 C3	RET		

4E08:A4E8 F8	CLC		; CF=0, 无重码
4E08:A4E9 FC	CLD		
4E08:A4EA C3	RET		
4E08:A4EB EB69FD	CALL	A257	; 查到的表项转换成汉字内码
4E08:A4EE 7210	JB	A500	; 若重码已存重码区转A 500 执行
-u			
4E08:A4F0 F606BB2B20	TEST	BYTE PTR [2BBB],20	
4E08:A4F5 7509	JNZ	A500	
4E08:A4F7 B936AD2B	MOV	[2BAD],SI	
4E08:A4FB 800EBB2B20	OR	BYTE PTR [2BBB],20	
4E08:A500 41	INC	CX	
4E08:A501 EBCC	JMP	A4CF	
4E08:A503 BFD62B	MOV	DI,2BD6	
4E08:A506 FD	STD		; 设置方向标志, 反向查表
4E08:A507 EB05	JMP	A50E	
4E08:A509 90	NOP		
4E08:A50A BF9695	MOV	DI,9596	; 对照表尾地址→查表结束指针
4E08:A50D FC	CLD		
4E08:A50E BB36AD2B	MOV	SI,[2BAD]	; 当前/下次查表指针→查表指针
-u			
4E08:A512 B026BB2BDF	AND	BYTE PTR [2BBB],DF	
4E08:A517 33C0	XOR	AX,AX	
4E08:A519 A2D22B	MOV	[2BD2],AL	
4E08:A51C A39795	MOV	[9597],AX	; 0→重码计数器
4E08:A51F 33C9	XOR	CX,CX	
4E08:A521 8B1EB12B	MOV	BX,[2BB1]	; 取外码区1、2键外码→BX
4E08:A525 80E31F	AND	BL,1F	
4E08:A528 80E71F	AND	BH,1F	
4E08:A52B 50	PUSH	AX	
4E08:A52C 8AE7	MOV	AH,BH	
4E08:A52E 32C0	XOR	AL,AL	
4E08:A530 D1E8	SHR	AX,1	
-u			
4E08:A532 D1E8	SHR	AX,1	
4E08:A534 D1E8	SHR	AX,1	
4E08:A536 BAFC	MOV	BH,AH	
4E08:A538 0ADB	OR	BL,AL	
4E08:A53A A1B32B	MOV	AX,[2BB3]	; 取外码区3、4键外码→AX
4E08:A53D 251F1F	AND	AX,1F1F	
4E08:A540 D0E0	SHL	AL,1	
4E08:A542 D0E0	SHL	AL,1	
4E08:A544 0AF8	OR	BH,AL	
4E08:A546 D0EC	SHR	AH,1	
4E08:A548 7303	JNB	A54D	
4E08:A54A 80CF80	OR	BH,80	
4E08:A54D A0B52B	MOV	AL,[2BB5]	
4E08:A550 86C4	XCHG	AH,AL	
-u			
4E08:A552 80E41F	AND	AH,1F	

4E08:A555 D0E4	SHL	AH,1	: AH中的5键外码(数字)左移4位
4E08:A557 D0E4	SHL	AH,1	
4E08:A559 D0E4	SHL	AH,1	
4E08:A55B D0E4	SHL	AH,1	
4E08:A55D 7207	JB	A566	: 若5键外码第4位是1转A566
4E08:A55F 0AC4	OR	AL,AH	
4E08:A561 32E4	XOR	AH,AH	
4E08:A563 EB07	JMP	A56C	
4E08:A565 90	NOP		
4E08:A566 0AC4	OR	AL,AH	
4E08:A568 32E4	XOR	AH,AH	
4E08:A56A FEC4	INC	AH	
4E08:A56C 8BE8	MOV	BP,AX	
4E08:A56E 5B	POP	AX	: 对查表指针(查完吗)?
4E08:A56F 3BF7	CMP	SI,DI	
4E08:A571 7503	JNZ	A576	
-u			
4E08:A573 EB36	JMP	A5AB	: 查完转
4E08:A575 90	NOP		
4E08:A576 53	PUSH	BX	
4E08:A577 8B04	MOV	AX,[SI]	
4E08:A579 8B5402	MOV	DX,[SI+02]	
4E08:A57C 81E2FF01	AND	DX,01FF	
4E08:A580 F606BB2B01	TEST	BYTE PTR [2BBB],01	: 拼音方式?
4E08:A585 7417	JZ	A59E	
4E08:A587 BADC	MOV	BL,AH	: 拼音方式,表项第2字节→BL
4E08:A589 D1EA	SHR	DX,1	
4E08:A58B D0D3	RCL	BL,1	
4E08:A58D D1E8	SHR	AX,1	
4E08:A58F D1DA	RCR	DX,1	
4E08:A591 D0D3	RCL	BL,1	
-u			
4E08:A593 80E401	AND	AH,01	
4E08:A596 86C3	XCHG	BL,AL	
4E08:A598 86E2	XCHG	DL,AH	
4E08:A59A 86D6	XCHG	DH,DL	
4E08:A59C 86DA	XCHG	DL,BL	
4E08:A59E 5B	POP	BX	
4E08:A59F 3BC3	CMP	AX,BX	
4E08:A5A1 7504	JNZ	A5A7	
4E08:A5A3 3BD5	CMP	DX,BP	
4E08:A5A5 741C	JZ	A5C3	
4E08:A5A7 AD	LODSW		: 根据方向标志调整查表指针(SI):
4E08:A5A8 AD	LODSW		
4E08:A5A9 EBC4	JMP	A56F	
4E08:A5AB 0BC9	OR	CX,CX	
4E08:A5AD 7411	JZ	A5C0	
4E08:A5AF 890E9795	MOV	[9597],CX	
-u			

4E08:A5B3 3B0ED02B	CMP	CX,[2B00]	
4E08:A5B7 7704	JA	A5BD	
4E08:A5B9 893EAD2B	MOV	[2BAD],DI	
4E08:A5BD F9	STC		
4E08:A5BE C3	RET		
4E08:A5BF FC	CLD		
4E08:A5C0 F8	CLC		
4E08:A5C1 FC	CLD		
4E08:A5C2 C3	RET		
4E08:A5C3 E891FC	CALL	A257	: 查到的表项转换成汉字内码
4E08:A5C6 7210	JB	A5DB	
4E08:A5CB F606BB2B20	TEST	BYTE PTR [2BBB],20	
4E08:A5CD 7509	JNZ	A5DB	
4E08:A5CF 8936AD2B	MOV	[2BAD],SI	
-u			
4E08:A5D3 800EBB2B20	OR	BYTE PTR [2BBB],20	
4E08:A5D8 41	INC	CX	
4E08:A5D9 EBCC	JMP	A5A7	
4E08:A5DB 8D1EFC95	LEA	BX,[95FC]	: “词组:” 提示信息地址→BX
4E08:A5DF FF369C95	PUSH	[959C]	
4E08:A5E3 E8EBF6	CALL	9CD1	: 提示行初始化, 显示“词组”
4E08:A5E6 8F069C95	POP	[959C]	
4E08:A5EA E86B00	CALL	A658	: 在提示行显示外码区所有外码
4E08:A5ED E87A00	CALL	A66A	: 建立词组表寻址指针
4E08:A5F0 BB36A22B	MOV	SI,[2BA2]	
-u			
4E08:A5F4 EB1202	CALL	A809	: SI指针下的词组长度→AX
4E08:A5F7 03069A2B	ADD	AX,[2B9A]	
4E08:A5FB A39E2B	MOV	[2B9E],AX	
4E08:A5FE E8C3F9	CALL	9FC4	: 为开始在词组表查重码设计指针
4E08:A601 C606A52B00	MOV	BYTE PTR [2BA5],00	
4E08:A606 8B3EA02B	MOV	DI,[2BA0]	
4E08:A60A E84D02	CALL	A85A	: 在词组表查词组重码
4E08:A60D E80602	CALL	A816	
4E08:A610 B00E3A9620	OR	BYTE PTR [963A],20	: 设置词组标志, 以进入词组状态
-u			
4E08:A615 C6069B9500	MOV	BYTE PTR [959B],00	
4E08:A61A 83F900	CMP	CX,+00	
4E08:A61D 750B	JNZ	A62A	
4E08:A61F 820B	MOV	DL,0B	
4E08:A621 E80CF9	CALL	9F30	: 清提示行
4E08:A624 E8AFFB	CALL	A1D6	: 提示行显示(000)
4E08:A627 E932FB	JMP	A15C	
4E08:A62A 51	PUSH	CX	
4E08:A62B E87100	CALL	A69F	: 提示行显示剩余重码数、重码词组
4E08:A62E 59	POP	CX	: 恢复重码数
4E08:A62F 83E901	CMP	CX,+01	: 只有一个词组重码?
4E08:A632 7403	JZ	A637	
4E08:A634 E9BEF7	JMP	9DC5	: 不只一个, 转键盘码处理子程序返回

```

-u
4E08:A637 C6069B95FF  MOV  BYTE PTR [959B],FF
4E08:A63C 8BD9      MOV  BX,CX
4E08:A63E 4B        DEC  BX
4E08:A63F B003      MOV  AL,03
4E08:A641 F6E3      MUL  BL
4E08:A643 8BD8      MOV  BX,AX
4E08:A645 8BB74B96    MOV  SI,[BX+964B] : 选中的词组在词表内码区的地址→SI
4E08:A649 89366996    MOV  [9669],SI
4E08:A64D 8AB74D96    MOV  AL,[BX+964D]
4E08:A651 A26B96      MOV  [966B],AL
4E08:A654 E8AA00      CALL A701          : 选中词组处理

-u
4E08:A657 C3        RET
4E08:A658 33DB      XOR  BX,BX
4E08:A65A 8A97B12B    MOV  DL,[BX+2BB1]
4E08:A65E E89A00      CALL A6FB          : 提示行显示DL中的外码
4E08:A661 FEC3      INC  BL
4E08:A663 3A1E9C95    CMP  BL,[959C]
4E08:A667 72F1      JB   A65A
4E08:A669 C3        RET
4E08:A66A 803E6C9600  CMP  BYTE PTR [966C],00 : 是否加载了用户词组表?
4E08:A66F 7513      JNZ  A684
4E08:A671 8CC8      MOV  AX,CS
4E08:A673 A3982B      MOV  [2B98],AX
4E08:A676 8D066D96    LEA  AX,[966D]      : 系统词组表外码区首址→词表外码区指针

-u
4E08:A67A A3A22B      MOV  [2BA2],AX
4E08:A67D 8D06A196    LEA  AX,[96A1]      : 系统词组表内码区首址→词组表内码区指针
4E08:A681 A3A02B      MOV  [2BA0],AX
4E08:A684 8B2EA02B      MOV  BP,[2BA0]
4E08:A688 892E9A2B      MOV  [2B9A],BP
4E08:A68C C606A62B17    MOV  BYTE PTR [2BA6],17
4E08:A691 26        ES:
4E08:A692 803E490006  CMP  BYTE PTR [0049],06 : 是640 × 200 图形方式?
4E08:A697 7505      JNZ  A69E
4E08:A699 C606A62B3A    MOV  BYTE PTR [2BA6],3A

-u
4E08:A69E C3        RET
4E08:A69F B20B      MOV  DL,0B          : OBH →DL
4E08:A6A1 E8BCF8      CALL 9F30          : 清提示行
4E08:A6A4 8A16D22B    MOV  DL,[2BD2]
4E08:A6AB 32F6      XOR  DH,DH
4E08:A6AA 29169795    SUB  [9597],DX
4E08:A6AE E825FB      CALL A1D6          : 显示“(” 剩余重码数 “)”
4E08:A6B1 B220      MOV  DL,20
4E08:A6B3 E84500      CALL A6FB
4E08:A6B6 B230      MOV  DL,30
4E08:A6B8 8A0ED22B    MOV  CL,[2BD2]      : 重码区重码数→CX
4E08:A6BC 32ED      XOR  CH,CH

```


4E08:A6BE E80500	CALL	A6C6	: 提示行显示重码词组号及重码词组
4E08:A6C1 FEC2	INC	DL	
4E08:A6C3 E2F9	LOOP	A6BE	: 继续显示直到重码词组, 直到CX = 0
4E08:A6C5 C3	RET		
4E08:A6C6 52	PUSH	DX	
4E08:A6C7 E83100	CALL	A6FB	: 显示词组重码号
4E08:A6CA B23A	MOV	DL,3A	
4E08:A6CC E82C00	CALL	A6FB	: 词组重码号后显示一个帽号
4E08:A6CF 8A1ED22B	MOV	BL,[2BD2]	
4E08:A6D3 2AD9	SUB	BL,CL	
4E08:A6D5 B003	MOV	AL,03	
4E08:A6D7 F6E3	MUL	BL	
4E08:A6D9 8BDB	MOV	BX,AX	
4E08:A6DB 8BB74B96	MOV	SI,[BX+964B]	
-u			
4E08:A6DF 51	PUSH	CX	
4E08:A6E0 8ABF4D96	MOV	CL,[BX+964D]	
4E08:A6E4 A1982B	MOV	AX,[2B98]	: 词组表段码→AX
4E08:A6E7 8ED8	MOV	DS,AX	
4E08:A6E9 AC	LODSB		
4E08:A6EA 8AD0	MOV	DL,AL	: 显示词组
4E08:A6EC E80C00	CALL	A6FB	
4E08:A6EF E2FB	LOOP	A6E9	
4E08:A6F1 B220	MOV	DL,20	
4E08:A6F3 E80500	CALL	A6FB	
4E08:A6F6 0E	PUSH	CS	
4E08:A6F7 1F	POP	DS	
4E08:A6F8 59	POP	CX	
4E08:A6F9 5A	POP	DX	
4E08:A6FA C3	RET		
4E08:A6FB B80310	MOV	AX,1003	: 设置功能号, 显示DL中汉字
4E08:A6FE CD10	INT	10	
-u			
4E08:A700 C3	RET		
4E08:A701 A1982B	MOV	AX,[2B98]	: 词组表段址→AX
4E08:A704 8B366996	MOV	SI,[9669]	: 选中词组内码地址指针→SI
4E08:A708 8A0E6B96	MOV	CL,[966B]	
4E08:A70C 8D3E9D95	LEA	DI,[959D]	
4E08:A710 80F938	CMP	CL,38	
4E08:A713 720C	JB	A721	
4E08:A715 80E938	SUB	CL,38	
4E08:A718 880E6B96	MOV	[966B],CL	
4E08:A71C B138	MOV	CL,38	
4E08:A71E EB06	JMP	A726	
-u			
4E08:A720 90	NOP		
4E08:A721 C6066B9600	MOV	BYTE PTR [966B],00	
4E08:A726 880ED595	MOV	[95D5],CL	
4E08:A72A 8ED8	MOV	DS,AX	
4E08:A72C 06	PUSH	ES	

4E08:A72D 0E	PUSH	CS	
4E08:A72E 07	POP	ES	
4E08:A72F 32ED	XOR	CH,CH	
4E08:A731 F3	REPZ		: 将当前词组内码 (SI寻址)
4E08:A732 A4	MOVSB		
4E08:A733 07	POP	ES	
4E08:A734 0E	PUSH	CS	
4E08:A735 1F	POP	DS	
4E08:A736 89366996	MOV	[9669],SI	
4E08:A73A C3	RET		
4E08:A73B 80FC82	CMP	AH,82	: 键盘扩展码 ≥ 82H (Alt " + " -) ?
4E08:A73E 7314	JNB	A754	: ≥ 82H, 恢复前后页处理
-u			
4E08:A740 80FC78	CMP	AH,78	
4E08:A743 721C	JB	A761	
4E08:A745 80EC77	SUB	AH,77	
4E08:A748 80FC0A	CMP	AH,0A	
4E08:A74B 7502	JNZ	A74F	
4E08:A74D 32E4	XOR	AH,AH	
4E08:A74F 8AC4	MOV	AL,AH	
4E08:A751 EB11	JMP	A764	
4E08:A753 90	NOF		
4E08:A754 80FC82	CMP	AH,82	
4E08:A757 7503	JNZ	A75C	
4E08:A759 EB1B	JMP	A776	: 转A 776 恢复前页处理
4E08:A75B 90	NOF		
4E08:A75C 80FC83	CMP	AH,83	
4E08:A75F 742E	JZ	A78F	
-u			
4E08:A761 E93AF7	JMP	9E9E	
4E08:A764 3806D22B	CMP	[2802],AL	: 挑选重码的数字超过重码区重码数吗?
4E08:A768 7703	JA	A76D	
4E08:A76A E9EFF9	JMP	A15C	
4E08:A76D 32ED	XOR	CH,CH	
4E08:A76F 8ACB	MOV	CL,AL	
4E08:A771 FEC1	INC	DI	
4E08:A773 E9C1FE	JMP	A637	
4E08:A776 EBB800	CALL	A031	: 为反向查词表设置内、外码区指针
4E08:A779 8B3EA22B	MOV	DI,[2BA2]	
4E08:A77D 83EF04	SUB	DI,+04	
4E08:A780 FD	STD		
-u			
4E08:A781 C606A52BFF	MOV	BYTE PTR [2BA5],FF	
4E08:A786 E8D100	CALL	A85A	: 词表查重子程序, 反向查表
4E08:A789 E8AE00	CALL	A83A	: 为下次查词表设置内、外码区指针
4E08:A78C E981FE	JMP	A610	
4E08:A78F E8BF00	CALL	A851	: 为正向查词表设置内外码区指针
4E08:A792 8B3EA02B	MOV	DI,[2BA0]	
4E08:A796 C606A52B00	MOV	BYTE PTR [2BA5],00	
4E08:A79B E8BC00	CALL	A85A	
4E08:A79E E87500	CALL	A816	: 为下次查词表设置指针

-u

4E08:A7A1 E96CFE JMP A610
 4E08:A7A4 3C08 CMP AL,08
 4E08:A7A6 751E JNZ A7C6
 4E08:A7A8 FE0E9C95 DEC BYTE PTR [959C] : 是退格键, 外码计数器 - 1
 4E08:A7AC 803E9C9500 CMP BYTE PTR [959C]
 4E08:A7B1 7510 JNZ A7C3 : 不是, 重新进入词组方式
 4E08:A7B3 80263A96DF AND BYTE PTR [963A],DF
 4E08:A7B8 EB73F7 DALL 9F2E : 清提示行
 4E08:A7BB 800E9B93FF OR BYTE PTR [959B],FF
 4E08:A7C0 E902F6 JMP 9DC5

-u

4E08:A7C3 E915FE JMP A5DB : 重新进入词组方式
 4E08:A7C6 3C0D CMP AL,0D : "→"
 4E08:A7C8 7507 JNZ A7D1
 4E08:A7CA C6069C9500 MOV BYTE PTR [959C],00 : 是回车键, 0→外码计数器
 4E08:A7CF EBE2 JMP A7B3 : 转A 7 B 3, 取消词组方式标志设置第一键标志
 4E08:A7D1 3C2C CMP AL,2C
 4E08:A7D3 7502 JNZ A7D7
 4E08:A7D5 EB9F JMP A776
 4E08:A7D7 3C2E CMP AL,2E
 4E08:A7D9 74B4 JZ A7BF
 4E08:A7DB 3C30 CMP AL,30
 4E08:A7DD 7209 JB A7E8
 4E08:A7DF 3C39 CMP AL,39
 4E08:A7E1 7705 JA A7E8

-u

4E08:A7E3 2C30 SUB AL,30
 4E08:A7E5 E97CFF JMP A764
 4E08:A7E8 3C20 CMP AL,20 : " " 空格键?
 4E08:A7EA 7505 JNZ A7F1
 4E08:A7EC 32C0 XOR AL,AL : 是, 空格键作数字 0→AL
 4E08:A7EE E973FF JMP A764
 4E08:A7F1 7213 JB A806
 4E08:A7F3 803E9C9503 CMP BYTE PTR [959C],03
 4E08:A7F8 730C JNB A806
 4E08:A7FA FE069C95 INC BYTE PTR [959C]
 4E08:A7FE EB7CF5 CALL 9D7D : 设置光标
 4E08:A801 E885F7 CALL 9F89 : 外码存外码区, 提示行显示外码?

-u

4E08:A804 EBB9 JMP A78F
 4E08:A806 E953F9 JMP A15C
 4E08:A809 1E PUSH DS
 4E08:A80A A1982B MOV AX,[2B98] : 取词组表段码
 4E08:A80D BED8 MOV DS,AX
 4E08:A80F BA4403 MOV AL,[SI+03] : SI指针下的词组长度→AL
 4E08:A812 32E4 XOR AH,AH
 4E08:A814 1F POP DS
 4E08:A815 C3 RET
 4E08:A816 EBC0F7 CALL 9FD9 : 为下次查词表设置外码区指针
 4E08:A819 8B364996 MOV SI,[9649]
 4E08:A81D EBE9FF CALL A809 : SI指针下的词组长度→AX

4E08:AB20 29069E2B	SUB	[2B9E],AX	: 调整正向查词表内码区指针
-u			
4E08:AB24 A19E2B	MOV	AX,[2B9E]	
4E08:AB27 A39C2B	MOV	[2B9C],AX	
4E08:AB2A A19A2B	MOV	AX,[2B9A]	
4E08:AB2D A39E2B	MOV	[2B9E],AX	
4E08:AB30 C3	RET		
4E08:AB31 A19C2B	MOV	AX,[2B9C]	
4E08:AB34 A39A2B	MOV	[2B9A],AX	
4E08:AB37 E9C7F7	JMP	A001	: 为反向查词表设置外码区指针
4E08:AB3A 8B364996	MOV	SI,[9649]	
4E08:AB3E E8C8FF	CALL	A809	: SI指针下的词组长度→AX
4E08:AB41 03069C2B	ADD	AX,[2B9C]	
-u			
4E08:AB45 A39E2B	MOV	[2B9E],AX	
4E08:AB48 A19A2B	MOV	AX,[2B9A]	
4E08:AB4B A39C2B	MOV	[2B9C],AX	
4E08:AB4E E99CF7	JMP	9FED	: 设置外码区指针
4E08:AB51 A19E2B	MOV	AX,[2B9E]	
4E08:AB54 A39A2B	MOV	[2B9A],AX	
4E08:AB57 E976F7	JMP	9FD0	: 为正向查词表设置外码区指针
4E08:AB5A 8026BB2BDF	AND	BYTE PTR [2BBB],DF	
4E08:AB5F 32F6	XOR	DH,DH	
4E08:AB61 8B36D22B	MOV	[2BD2],DH	
-u			
4E08:AB65 A1B12B	MOV	AX,[2BB1]	: 1、2键外码→AX
4E08:AB68 8A16B32B	MOV	DL,[2BB3]	: 3键外码→DL
4E08:AB6C 8B36AD2B	MOV	SI,[2BAD]	: 当前查表指针→SI
4E08:AB70 8B2E9A2B	MOV	BP,[2B9A]	
4E08:AB74 8B0E982B	MOV	CX,[2B98]	: 取词表段码
4E08:AB78 8ED9	MOV	DS,CX	
4E08:AB7A 33C9	XOR	CX,CX	
4E08:AB7C 3BF7	CMP	SI,DI	
4E08:AB7E 7519	JNZ	A899	
4E08:AB80 0E	PUSH	CS	
4E08:AB81 1F	POP	DS	
4E08:AB82 890E9795	MOV	[9597],CX	: 重码数→重码计数器
-u			
4E08:AB86 A0D22B	MOV	AL,[2BD2]	: 重码区重码数→AL
4E08:AB89 32E4	XOR	AH,AH	
4E08:AB8B 38C8	CMP	CX,AX	
4E08:AB8D 7708	JA	A897	
4E08:AB8F 8936AD2B	MOV	[2BAD],SI	: 查表指针→当前查表指针
4E08:AB93 892E9A2B	MOV	[2B9A],BP	
4E08:AB97 FC	CLD		
4E08:AB98 C3	RET		
4E08:AB99 3A04	CMP	AL,[SI]	: 外码第1键查到词组重码吗?
4E08:AB9B 7520	JNZ	A8BD	
4E08:AB9D 2E	CS:		
4E08:AB9E 803E9C9501	CMP	BYTE PTR [959C],01	: 查到, 外码只有1键?
4E08:ABA3 7503	JNZ	A8A8	
4E08:ABA5 EB3A	JMP	A8E1	: 只有1键, 转查到词组重码处理

```

-u
4E08:ABA7 90      NOP
4E08:ABAB 3A6401  CMP     AH,[SI+01]      ; 外码 1、2 键查到词组重码了吗?
4E08:ABAB 7510    JNZ     ABB0
4E08:ABAD 2E      CS:
4E08:ABAE 803E9C9502 CMP     BYTE PTR [959C],02 ; 查到, 外码只有 2 键?
4E08:ABB3 7503    JNZ     ABB8
4E08:ABB5 EB2A    JMP     ABE1            ; 只有 2 键, 转查到词组重码处理
4E08:ABB7 90      NOP
4E08:ABBB 3A5402  CMP     DL,[SI+02]      ; 外码 1、2、3 键查到词组重码吗?
4E08:ABBB 7424    JZ      ABE1            ; 查到, 转查到词组重码处理
4E08:ABBD 8A5C03  MOV     BL,[SI+03]      ; 查表指针下(SI)的词组长度→BL
4E08:ABCO 32FF    XOR     BH,BH
4E08:ABC2 2E      CS:
4E08:ABC3 803EA52BFF CMP     BYTE PTR [2BA5],FF
-u
4E08:ABC8 7402    JZ      ABCC
4E08:ABCA 03EB    ADD     BP,BX
4E08:ABCC 50      PUSH    AX
4E08:ABCD AD      LODSW
4E08:ABCE AD      LODSW
4E08:ABCF 58      POP     AX
4E08:ABD0 2E      CS:
4E08:ABD1 803EA52BFF CMP     BYTE PTR [2BA5],FF
4E08:ABD6 7507    JNZ     ABDF
4E08:ABD8 8A5C03  MOV     BL,[SI+03]
4E08:ABD8 32FF    XOR     BH,BH
4E08:ABDD 2BEB    SUB     BP,BX
4E08:ABDF EB9B    JMP     AB7C            ; 词表内码区指针指向上一个词表内码
4E08:ABE1 41      INC     CX
4E08:ABE2 027403  ADD     DH,[SI+03]
4E08:ABE5 80C603  ADD     DH,03
-u
4E08:ABE8 2E      CS:
4E08:ABE9 3A36A62B CMP     DH,[2BA6]
4E08:ABED 760B    JBE     ABFA            ; 查到的重码词组超过提示行容量?
4E08:ABEF 2E      CS:
4E08:ABF0 803ED22B00 CMP     BYTE PTR [2BD2],00
4E08:ABF5 7403    JZ      ABFA
4E08:ABF7 EB22    JMP     A91B
4E08:ABF9 90      NOP
4E08:ABFA 2E      CS:
4E08:ABFB 8A1ED22B MOV     BL,[2BD2]
4E08:ABFF 50      PUSH    AX
4E08:A900 B003    MOV     AL,03
4E08:A902 F6E3    MUL     BL
4E08:A904 BBD8    MOV     BX,AX
4E08:A906 8A4403  MOV     AL,[SI+03]
-u

```

4E0B:A909 2E	CS:		
4E0B:A90A 88B74D96	MOV	[BX+964D],AL	
4E0B:A90E 2E	CS:		
4E0B:A90F 89AF4B96	MOV	[BX+964B],BP	
4E0B:A913 5B	POP	AX	
4E0B:A914 2E	CS:		
4E0B:A915 FE06D22B	INC	BYTE PTR [2B02]	
4E0B:A919 EBA2	JMP	ABBD	
4E0B:A91B 2E	CS:		
4E0B:A91C F606BB2B20	TEST	BYTE PTR [2BBB],20	: 测试剩余重码记录开关位
4E0B:A921 7510	JNZ	A933	: 如果开关已关上, 转A 9 33
4E0B:A923 2E	CS:		
4E0B:A924 8936AD2B	MOV	[2BAD],SI	
4E0B:A928 2E	CS:		
4E0B:A929 892E9A2B	MOV	[2B9A],BP	
-u			
4E0B:A92D 2E	CS:		
4E0B:A92E 800EBB2B20	OR	BYTE PTR [2BBB],20	
4E0B:A933 EBB8	JMP	ABBD	
4E0B:A935 3C0B	CMP	AL,0B	
4E0B:A937 750D	JNZ	A946	
4E0B:A939 BAE0	MOV	AH,AL	
4E0B:A93B 06	PUSH	ES	
4E0B:A93C 1E	PUSH	DS	
4E0B:A93D 07	POP	ES	
4E0B:A93E AB	STOSW		: AX送DI为位置指针的内码区
4E0B:A93F 07	POP	ES	
4E0B:A940 8006D59502	ADD	BYTE PTR [95D5],02	
4E0B:A945 C3	RET		
4E0B:A946 3C20	CMP	AL,20	
4E0B:A948 7502	JNZ	A94C	
4E0B:A94A EBED	JMP	A939	
4E0B:A94C 3C20	CMP	AL,20	: AL<20H?
-u			
4E0B:A94E 720B	JB	A958	: <20H, 非前显示字符, 转A 958
4E0B:A950 0C80	OR	AL,80	: 可显示的字符转换成国标03区的图形
4E0B:A952 B4A3	MOV	AH,A3	
4E0B:A954 B6E0	XCHG	AL,AH	
4E0B:A956 EBE3	JMP	A93B	: 转A 93B, AX存内码区
4E0B:A958 8B05	MOV	[DI],AL	
4E0B:A95A FE06D595	INC	BYTE PTR [95D5]	
4E0B:A95E C3	RET		
4E0B:A95F B200	MOV	DL,00	
4E0B:A961 EBCCF5	CALL	9F30	: 清提示行
4E0B:A964 C6069598FF	MOV	BYTE PTR [9895],FF	: 进入定义打印机过程
4E0B:A969 8D1E9698	LEA	BX,[9896]	: “打印字号(A—P):” →BX
4E0B:A96D B90F00	MOV	CX,000F	
-u			
4E0B:A970 32D2	XOR	DL,DL	

4E08:A972 E8D4F8	CALL	A249	
4E08:A975 B214	MOV	DL,14	: 14H → DL (光标位置)
4E08:A977 8D1EA598	LEA	BX,[98A5]	: "纸宽(80—134):" → BX
4E08:A97B B90E00	MOV	CX,000E	
4E08:A97E E8C8F8	CALL	A249	
4E08:A981 B210	MOV	DL,10	: 10H → DL
4E08:A983 B80210	MOV	AX,1002	
4E08:A986 CD10	INT	10	: 设置光标在"打印字号(A—P):"之后
4E08:A98B C6069C9500	MOV	BYTE PTR [959C],00	: 外码计数器 ≥ 5?
4E08:A98D E935F4	JMP	9DC5	
-u			
4E08:A990 803E9C9505	CMP	BYTE PTR [959C],05	
4E08:A995 7322	JNB	A9B9	
4E08:A997 2C30	SUB	AL,30	*
4E08:A999 3C09	CMP	AL,09	: 数字 0 ~ 9?
4E08:A99B 771C	J _A	A9B9	
4E08:A99D FE069C95	INC	BYTE PTR [959C]	是, 外码计数器 + 1
4E08:A9A1 8A1E9C95	MOV	BL,[959C]	
4E08:A9A5 80EB02	SUB	BL,02	
4E08:A9AB 32FF	XOR	BH,BH	
4E08:A9AA 8887B12B	MOV	[BX+2BB1],AL	
4E08:A9AE BAD0	MOV	DL,AL	: 数字 → DL
-u			
4E08:A9B0 80C230	ADD	DL,30	: 恢复数字为ASCII码字符
4E08:A9B3 E845FD	CALL	A6FB	: 提示行显示数字字符
4E08:A9B6 E90CF4	JMP	9DC5	
4E08:A9B9 EB67	JMP	AA22	
4E08:A9BB 90	NOP		
4E08:A9BC 3C08	CMP	AL,08	: "←" (退格键)?
4E08:A9BE 750A	JNZ	A9CA	
4E08:A9C0 803E9C9502	CMP	BYTE PTR [959C],02	: 是, 键入过数字字符吗?
4E08:A9C5 725B	JB	AA22	
4E08:A9C7 EB5C	JMP	AA25	: 键入过数字字符, 转AA 25退格键处理
4E08:A9C9 90	NOP		
4E08:A9CA 3C0D	CMP	AL,0D	: "←" (退格键)?
4E08:A9CC 75C2	JNZ	A990	
4E08:A9CE 803E9C9502	CMP	BYTE PTR [959C],02	: 是, 键入过数字字符吗?
-u			
4E08:A9D3 7503	JNZ	A9DB	: 键入过数字字符, 转回车键处理
4E08:A9D5 EB22	JMP	A9F9	
4E08:A9D7 90	NOP		
4E08:A9D8 B10A	MOV	CL,0A	
4E08:A9DA 8A2E9C95	MOV	CH,[959C]	: 定义纸宽数字字符数 → CH
4E08:A9DE 80ED02	SUB	CH,02	
4E08:A9E1 33DB	XOR	BX,BX	
4E08:A9E3 FEC3	INC	BL	
4E08:A9E5 B000	MOV	AL,00	
4E08:A9E7 F6E1	MUL	CL	
4E08:A9E9 0287B12B	ADD	AL,[BX+2BB1]	

4E0B:A9ED	FEC3	INC	BL	
4E0B:A9EF	FECD	DEC	CH	
4E0B:A9F1	75F4	JNZ	A9E7	
-u				
4E0B:A9F3	33D2	XOR	DX,DX	
4E0B:A9F5	B403	MOV	AH,03	: 设置功能号
4E0B:A9F7	CD17	INT	17	
4E0B:A9F9	B80010	MOV	AX,1000	: 设置功能号
4E0B:A9FC	CD10	INT	10	
4E0B:A9FE	C606959800	MOV	BYTE PTR [9895],00	
4E0B:AA03	C6069C9500	MOV	BYTE PTR [959C],00	
4E0B:AA0B	C6069B95FF	MOV	BYTE PTR [959B],FF	
4E0B:AA0D	E9B5F3	JMP	9DC5	: 转, 键盘码处理子程序返回
4E0B:AA10	803E9C9502	CMF	BYTE PTR [959C],02	: 外码计数器 ≥ 2 ?
-u				
4E0B:AA15	73A5	JNB	A9BC	: ≥ 2, 转A 9 BC定义打印机纸宽
4E0B:AA17	3C0B	CMF	AL,0B	: < 2, 接A A 17定义打印机字号
4E0B:AA19	752F	JNZ	AA4A	
4E0B:AA1B	803E9C9500	CMF	BYTE PTR [959C],00	
4E0B:AA20	7503	JNZ	AA25	
4E0B:AA22	E937F7	JMP	A15C	
4E0B:AA25	FE0E9C95	DEC	BYTE PTR [959C]	: 退格键, 外码计数器 - 1
4E0B:AA29	B208	MOV	DL,08	
4E0B:AA2B	EBCDFC	CALL	A6FB	: 光标退一格
4E0B:AA2E	E994F3	JMP	9DC5	
4E0B:AA31	245F	AND	AL,5F	
4E0B:AA33	3C41	CMF	AL,41	: < "A" ?
-u				
4E0B:AA35	72EB	JB	AA22	
4E0B:AA37	3C50	CMF	AL,50	: > "P" ?
4E0B:AA39	77E7	JA	AA22	
4E0B:AA3B	FE069C95	INC	BYTE PTR [959C]	: "A" ~ "P", 外码计数器 + 1
4E0B:AA3F	A2B12B	MOV	[2BB1],AL	: 定义打印机字号字母存外码区
4E0B:AA42	8AD0	MOV	DL,AL	
4E0B:AA44	E8B4FC	CALL	A6FB	: 提示行显示DL中的字母
4E0B:AA47	E97BF3	JMP	9DC5	
4E0B:AA4A	3C0D	CMF	AL,0D	
4E0B:AA4C	75E3	JNZ	AA31	
4E0B:AA4E	803E9C9501	CMF	BYTE PTR [959C],01	
4E0B:AA53	7515	JNZ	AA6A	: = 0, 转AA 6 A 执行
-u				
4E0B:AA55	33D2	XOR	DX,DX	
4E0B:AA57	B01B	MOV	AL,1B	: ESC (1 B) → AL
4E0B:AA59	32E4	XOR	AH,AH	
4E0B:AA5B	CD17	INT	17	ESC送打印机
4E0B:AA5D	B049	MOV	AL,49	
4E0B:AA5F	32E4	XOR	AH,AH	
4E0B:AA61	CD17	INT	17	: "I" 送打印机

4E08:AA63 A0B12B MOV AL,[2BB1]
4E08:AA66 32E4 XOR AH,AH
4E08:AA68 CD17 INT 17 ; 定义打印机字母→打印机
4E08:AA6A B223 MOV DL,23 ; 23H→DL
4E08:AA6C BB0210 MOV AX,1002
4E08:AA6F CD10 INT 10 ; 设置光标到纸宽提示信息之后
4E08:AA71 C6069C9502 MOV BYTE PTR [959C],02

-u

4E08:AA76 E94CF3 JMP 9DC5
4E08:AA79 63 DB 63
4E08:AA7A 63 DB 63
4E08:AA7B 6C DB 6C
4E08:AA7C 69 DB 69
4E08:AA7D 62 DB 62
4E08:AA7E 000A ADD [BP+SI],CL
4E08:AAB0 0D0A0D OR AX,0D0A
4E08:AAB3 2020 AND [BX+SI],AH
4E08:AAB5 2020 AND [BX+SI],AH
4E08:AAB7 2020 AND [BX+SI],AH
4E08:AAB9 2020 AND [BX+SI],AH
4E08:AABB 2020 AND [BX+SI],AH
4E08:AABD 2020 AND [BX+SI],AH
4E08:AABF 2020 AND [BX+SI],AH
4E08:AA91 20BFC6BA AND [BX+BAC6],BH
4E08:AA95 A3C5E0 MOV [E0C5],AX

-dcs:aa79

4E08:AA79 63 63 6C 69 62 00 0A cclib..
4E08:AAB0 0D 0A 0D 20 20 20 20 20-20 20 20 20 20 20 20 20 20 ...
4E08:AA90 20 20 BF C6 BA A3 C5 E0-D1 B5 D6 D0 D0 C4 20 20 科海培训中心
4E08:AAA0 20 20 20 BA BA D7 D6 B2-D9 D7 F7 CF B5 CD B3 20 汉字操作系统
4E08:AAB0 20 20 B0 E6 B1 BE 20 A2-B2 A3 B1 A3 B0 0A 0D 24 版本 2.1 0..\$
4E08:AAC0 1E 33 C0 50 0E 1F BD 16-79 AA B8 00 3D CD 21 73 .3EP...y嘴.=M?s
4E08:AAD0 03 EB 70 90 50 B8 1F 35-CD 21 BC C2 89 16 75 00 .kp.P8.5M?.B..u.
4E08:AAE0 BE DA BB D3 5B B9 00 B0-B4 3F CD 21 72 55 3D 00 .Z.SI9..4?M?rll=.
4E08:AAF0 00 74 0C BC DA B1 C2 00-08 .t..Z.B..

-u cs:aac0

4E08:AAC0 1E PUSH DS
4E08:AAC1 33C0 XOR AX,AX
4E08:AAC3 50 PUSH AX
4E08:AAC4 0E PUSH CS ; 改DS=CS
4E08:AAC5 1F POP DS
4E08:AAC6 8D1679AA LEA DX,[AA79] ; DS: DX指向CCLIB首字符
4E08:AACA B8003D MOV AX,3D00 ; 3DH号功能调用,打开汉字库文件供读
4E08:AACD CD21 INT 21
4E08:AACF 7303 JNB AAD4
4E08:AAD1 EB70 JMP AB43
4E08:AAD3 90 NOP
4E08:AAD4 50 PUSH AX ; 保存CCLIB文件把柄

4E08:AAD5 B81F35	MOV	AX,351F	: 取 1 F 类中断向量地址指针→EX; BX
4E08:AADB CD21	INT	21	
4E08:AADA 8CC2	MOV	DX,ES	
4E08:AADC 89167500	MOV	[0075],DX	: 汉字字模库首址段码送CS: 0075单元
-u			
4E08:AAE0 8EDA	MOV	DS,DX	: DS; DX指向汉字字模库首址
4E08:AAE2 8BD3	MOV	DX,BX	
4E08:AAE4 5B	POP	BX	: CCLIB文件FCB弹出到BX
4E08:AAE5 890080	MOV	CX,8000	: 从CCLIB中 (BX) 读8000H 个字节
4E08:AAE8 B43F	MOV	AH,3F	
4E08:AAEA CD21	INT	21	
4E08:AAEC 7255	JB	AB43	
4E08:AAEE 3D0000	CMP	AX,0000	: CCLIB文件全部读完了吗?
4E08:AAF1 740C	JZ	A AFF	: 若读完, 转A AFF
4E08:AAF3 BCD8	MOV	DX,DS	
4E08:AAF5 81C20008	ADD	DX,0800	
4E08:AAF9 8EDA	MOV	DS,DX	
4E08:AFB8 33D2	XOR	DX,DX	
4E08:AFD EBE6	JMP	AAE5	
4E08:AFF 0E	PUSH	CS	
-u			
4E08:AB00 1F	POP	DS	
4E08:AB01 B43E	MOV	AH,3E	: 关闭CCLIB文件
4E08:AB03 CD21	INT	21	
4E08:AB05 8D168B18	LEA	DX,[1888]	: 修改 1 DH类中断指针, 初始化数据表
4E08:AB09 B81D25	MOV	AX,251D	
4E08:AB0C CD21	INT	21	
4E08:AB0E 8D164818	LEA	DX,[1848]	: 修改 10H类中断指针, 指向CS: 1848
4E08:AB12 B81D25	MOV	AX,251D	
4E08:AB15 CD21	INT	21	
4E08:AB17 8D169827	LEA	DX,[2798]	: 修改 1 FH类中断指针, 指向CS: 2798
4E08:AB1B B81F25	MOV	AX,251F	
4E08:AB1E CD21	INT	21	
-u			
4E08:AB20 8D16B398	LEA	DX,[98B3]	: 修改 16H类中断指针, 指向CS: 98B 3
4E08:AB24 B81625	MOV	AX,2516	
4E08:AB27 CD21	INT	21	
4E08:AB29 B80600	MOV	AX,0006	: 将显示器设置成640 × 200 图形方式
4E08:AB2C CD10	INT	10	
4E08:AB2E 8D167FAA	LEA	DX,[AA7F]	: 取CCBIOS标题首址送DX
4E08:AB32 B409	MOV	AH,09	
4E08:AB34 CD21	INT	21	
4E08:AB36 8D1679AA	LEA	DX,[AA79]	
4E08:AB3A 81C20301	ADD	DX,0103	
4E08:AB3E 90	NOP		
4E08:AB3F 5B	POP	AX	
-u			

4E08:AB40	1F	POP	DS
4E08:AB41	CD27	INT	27
4E08:AB43	CB	RETF	
4E08:AB44	0000	ADD	[BX+SI],AL
4E08:AB46	0000	ADD	[BX+SI],AL
4E08:AB48	0000	ADD	[BX+SI],AL
4E08:AB4A	0000	ADD	[BX+SI],AL
4E08:AB4C	0000	ADD	[BX+SI],AL
4E08:AB4E	0000	ADD	[BX+SI],AL
4E08:AB50	0000	ADD	[BX+SI],AL
4E08:AB52	0000	ADD	[BX+SI],AL
4E08:AB54	0000	ADD	[BX+SI],AL
4E08:AB56	0000	ADD	[BX+SI],AL
4E08:AB58	0000	ADD	[BX+SI],AL
4E08:AB5A	0000	ADD	[BX+SI],AL
4E08:AB5C	0000	ADD	[BX+SI],AL
4E08:AB5E	0000	ADD	[BX+SI],AL

第六章 CC-BIOS 的显示器控制模块

6.1 总述

CC-BIOS中的显示器控制模块(亦称CRT控制模块),是对ROM BIOS中的显示器控制模块作了较大的扩充后形成的,主要是扩充了显示汉字的功能。由于在显示汉字时,显示器必须工作在图形方式下,而ROM-BIOS的显示器控制模块基本上是在字符方式下工作的,故除了扩充汉字显示功能外,对CRT控制模块中的其他一些功能块,也要进行扩充,使得CC-BIOS中的CRT控制模块可以在图形方式下模仿ROM-BIOS中的CRT控制模块的所有功能,从而使其能与西文软件兼容。由此可见,这部分的内容是相当丰富的。

实现汉字显示输出的过程,实际上是把汉字机内码转换成其对应的汉字字模,并把字模信息送屏幕显示的过程。在CC-DOS(V2.1)中,采用异形国标码(即国标码的最高位置1)来作为机内码,它用两个字节来表示一个机内码。汉字字模存放在汉字库中,其点阵为 16×16 的。字模采用横向点阵,即把表示一个汉字字模的32个字节,看成为16个字,每个字的16位依次表示汉字一行上的16个点的情况。这样的字模适用于在显示器上显示汉字。汉字库中的字模是按其对应的国标码的次序排列的。另外,还有一个字符字模库,它被用来存放字符的字模。在图形方式下,显示器的字符发生器是不起作用的。这时,如要显示字符,就要去字符字模库中取出其对应的字模,再把字模信息送屏幕显示。字符的字模点阵是 8×8 的,每个字符字模占8个字节。

对CRT控制模块的调用,是通过调用软中断10H来实现的,所以CRT控制模块亦被称为10H类中断程序。

6.2 显示器控制模块的结构

CRT控制模块(10H类中断程序)是由二十个独立的功能块(块号为0—19)组成的。它们分别完成规定的功能。表6-1列出了这二十个功能块的情况。

CRT控制模块的功能

表6-1

功能块号	入口地址	完 成 功 能	功能块号	入口地址	完 成 功 能
0	CS:18E0	对CRT进行初始化	10	CS:1EA9	在当前光标位置写字符
1	CS:1A48	置光标类型	11	CS:1CC2	设置屏幕彩色色调
2	CS:1A69	置光标位置	12	CS:1EEB	在指定坐标处写点
3	CS:1C65	读光标位置	13	CS:1EDA	在指定坐标处读点
4	CS:24B3	读光笔位置	14	CS:23FC	以TTY方式显示字符
5	CS:1C9E	选择有果显示页	15	CS:1CE8	取当前视屏状态
6	CS:1D49	屏幕向上滚动有效页	16	CS:24B8	提示行操作
7	CS:1DF1	屏幕向下滚动有效页	17	CS:25AD	修改汉字库字模
8	CS:1E30	在当前光标位置读字符/属性	18	CS:25C0	取汉字库字模到指定处
9	CS:1E76	在当前光选位置写字符/属性	19	CS:25D3	光标的建立和取消

注:本章中的CS均表示CRT控制模块的代码段值

调用其中的某个功能块，就能完成相应的功能。其调用方法是，把欲调用之功能块的块号送入 AH 寄存器，然后调用 10H 类软中断，即可实现对该功能块的调用。

6.3 几个重要的工作区

显示器屏幕上的显示信息是存放在 CRT 刷新区内的。CRT 刷新区地址为 B000:8000—9F3 (偶数线扫描区) 和 B000:A000—BF3F (奇数线扫描区)，共 16000 个字节。CC-DOS 自举时把显示器初始化成 200×640 图形方式，这时屏幕上共有 128000 个象元，每个象元对应于 CRT 刷新区中的一位。改变刷新区中的某一位的值，则改变了与其对应的屏幕象元的状态 (亮或黑)。

在 200×640 图形方式下，因为汉字的点阵是 16×16 的，再考虑到行的间隔，所以屏幕上只能显示 11 行信息。其中最后一行由 CC-DOS 规定作为系统提示行，所以用户能用的仅 10 行。而在字符方式下，通常屏幕能显示 25 行，每行 80 字符。不少系统软件和应用软件是按此方式编制的。为了做到能与西文软件兼容，在 CC-BIOS 的 CRT 控制模块中引入了两种屏幕概念，即实屏和虚屏。虚屏是面向用户的，每屏为 80 行。实屏是真正的屏幕，它是面向 CRT 控制模块的，每屏为 11 行。在引入虚屏概念后，用户能把显示器作为每屏 25 行来使用。一些要 25 行屏幕支持下运行的软件，也能在 CC-BIOS 的 CRT 控制模块支持下运行。

因实屏就是真正的屏幕，故实屏的显示内容被存在 CRT 刷新区内，这在上边已解释过了。虚屏的内容也要由刷新区来记录，这个刷新区称为虚刷新区，虚刷新区用来存放虚屏的 25×80 个字符，它的每个字节对应于虚屏上的一个字符。为叙述方便，我们把实屏的刷新区和虚屏的刷新区分别称为 RRAM 和 VRAM。VRAM 由三个分区组成，每个分区均为 25×80 字节，这三个分区中的字节是一一对应的，表 6-2 列出了 VRAM 分区的情况。

VRAM 的分区

表 6-2

分 区 名	地 址	存 放 内 容
字符 VRAM	CS:00B0—087F	字 符
属性 VRAM	CS:0880—104F	字符的属性字
标志 VRAM	CS:0050—081F	字符的标志字

我们再来解释一下 VRAM 与 RRAM 间的关系。面向用户的虚屏是 25 行，而实屏仅 11 行，故实屏显示的信息只是虚屏中的一部分内容。我们把 VRAM 中对应于 RRAM 的这一部分，称为 RRAM 在 VRAM 中的映象区。为了反映出 RRAM 对应于 VRAM 中的哪一部分，CRT 控制模块中有两个指针：RRAM 首指针和 RRAM 尾指针，其地址分别为 CS:00A5 和 CS:00A6，它们分别指向 RRAM 在 VRAM 中的映象区的首和尾。指针的内容是行序号 (0—24)。尾指针与首指针之差永远为 9，也就是说，两指针间的映象区为 10 行。通过改变两个指针之值，可实现 RRAM 在 VRAM 中的浮动。另外，因 CC-DOS 规定实屏的最后一行 (第 10 行) 为提示行，为减少屏幕的滚动次数，所以把虚屏中的第 24 行 (最后一行) 定为提示行的映象区，即实屏的第 10 行永远对应于虚屏的第 24 行。图 6-1 描述了 VRAM 与 RRAM 间的关系。还要指出一点，就是 VRAM 中存放的是字符信息 (即代码)，而 RRAM 中存放的是图形信息 (即字模)。

为了实现显示汉字，CRT 控制模块中还使用了一个字模缓冲区。它的作用是作为字模信息

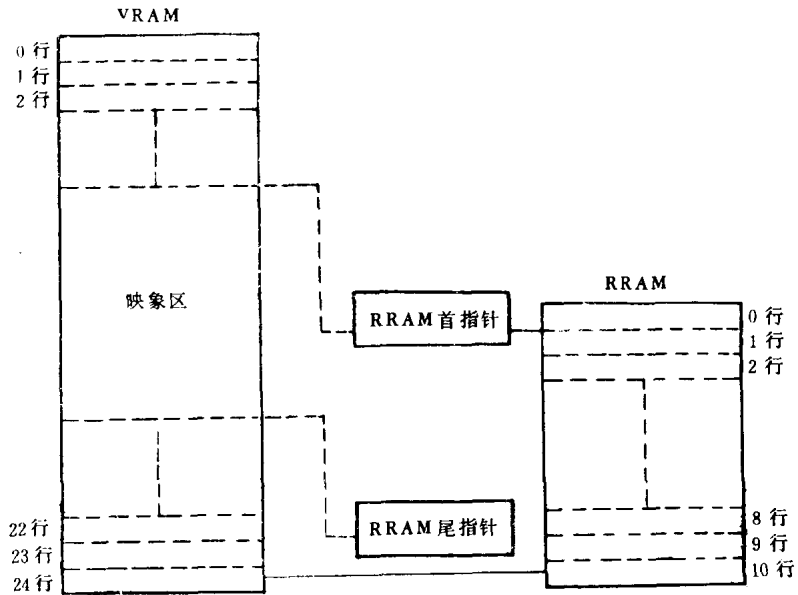


图 6-1 VRAM与RRAM之间的关系

的加工场所。它共有三十六个字节，地址为CS: 0078—009B。它分为左半区和右半区，每半区均为十八字节。字模信息在显示前先被送到该缓冲区进行排列和加工，然后再把该缓冲区中的内容复写到RRAM中去，以完成显示。

为了实现显示器的初始化，CRT控制模块中有两个参数区。一个是CRTC（CRT控制器）初始化参数区，存放着CRTC中十六个数据寄存器的初值。对应于显示器的每种工作方式均有一组这样的数据，因此这个数据区中存放了若干组这样的初始数据。该区的地址为CS: 1888—18C7。另一个是CRT控制字参数区，存放着对应于各种工作方式的CRT控制字。该区的地址为CS: 18D8—18DF。

其他一些工作区将在下面各部分中分别给予介绍。

6.4 CRT的初始化

CC-DOS自举时，在引导CC BIOS的过程中要对CRT进行初始化。这项工作是通过调用CRT控制模块中的0号功能块来完成的。对CRT初始化所做的主要工作是：根据指定的CRT工作方式，对CRT和CRTC（IBM PC系统的CRTC采用M6845）的寄存器置以相应的初值，对一些工作区的单元置以初始值。

在调用0号功能块对CRT进行初始化时，先要把CRT的工作方式号送入AL寄存器，然后再调用0号功能块，即可把CRT初始化成指定的工作方式。CRT工作方式与其方式号的对应关系如表6-3所示。

CRT工作方式

表6-3

方式号	CRT工作方式	方式号	CRT工作方式
0	25×40黑白字符方式	4	200×320彩色图形方式
1	25×40彩色字符方式	5	200×320黑白图形方式
2	25×80黑白字符方式	6	200×640黑白图形方式
3	25×80彩色字符方式		

表 6-3 中的 CRT 工作方式是对彩色图形显示器而言的。

另外, 单色字符显示器的工作方式是 25×80 字符方式, 其对应的方式号为 7。

0 号功能块的执行步骤如下:

1) 取 CRT C 的索引寄存器的 I/O 口地址, 并把该地址与 CRT 工作方式号一起保存到系统的 ROM 通讯区 (从 0040:0000 开始) 中。CRT C M6845 共有十六个数据寄存器, 它们合用一个 I/O 口地址。向 CRT C 的数据寄存器传送信息, 均是送往此 I/O 口。具体传送到哪个寄存器, 由 CRT C 索引寄存器中的内容指出。故在向 CRT C 的数据寄存器送信息前, 先要向 CRT C 索引寄存器送入欲送往的数据寄存器号。彩色显示器和单色显示器之 CRT C 索引寄存器的 I/O 地址分别为 3 D 4 和 3 B 4。

2) 把 1F H 类中断指针指向字符字模库的始址。在图形方式下输出字符时要用到它。

3) 根据 CRT 的工作方式号, 到 CRT C 初始化参数区中的相应区域取初始化数据, 并分别送入 CRT C M6845 的十六个数据寄存器, 从而实现了 CRT C 的初始化。彩色显示器和单色显示器之 CRT C 数据寄存器的 I/O 口地址分别为 3 D 5 和 3 B 5。

4) 对 CRT 的刷新区进行初始化。CRT 如是字符方式, 则往其刷新区的字符单元送入 20 H (空格), 往其属性字单元送入 07 H (表示正常的黑底白字显示)。CRT 如是图形方式, 则把刷新区的各单元全清为 0, 即清屏。

5) 根据 CRT 工作方式号, 去 CRT 控制字参数区的相应位置取数据, 并送入 CRT 控制寄存器中。CRT 控制寄存器的内容决定了 CRT 的工作方式。彩色显示器和单色显示器之控制寄存器的 I/O 口地址分别为 3 D 8 和 3 B 8。

6) 定义光标性质。光标的性质由光标的属性字来决定, 它描述光标的形状、光标的闪烁和光标的隐现等情况。这个字保存在系统的 ROM 通讯区内。

7) 将 CRT 控制字和 CRT 屏幕的列数保存到系统的 ROM 通讯区内。

8) 将当前光标位置置 0 行 0 列。有一个当前光标位置字专门记录当前光标的位置, 只要把该字清为 0 即可。然后把当前光标位置字的内容保存到系统的 ROM 通讯区中。

9) 定义彩色选择字。即把决定 CRT 彩色功能的彩色选择字置入 CRT 彩色选择寄存器中, CRT 彩色选择寄存器的 I/O 口地址为 3 D 9。

10) CRT 的工作方式如为字符方式, 则直接返回。如是图形方式, 则还要对 VRAM 和字模缓冲区进行初始化, 即把这些区的所有单元都清为 0, 然后返回。

6.5 光标的产生和定位

光标问题是 CRT 控制模块中要解决的一个大问题。在字符方式下, 光标的产生和定位是由 CRT C 来实现的。但是, 在图形方式下, CRT C 就不能实现这些功能了, 这时, 光标的产生和定位均要由 CRT 控制模块本身中程序来实现。这部分功能是由 2 号功能块来完成的。下面我们就对 2 号功能块进行一些分析。

1. 光标的产生

在字符方式下, 光标是由 CRT C 产生的。我们只要把 CRT 初始化时定义的光标属性字送入 CRT C 的光标开始寄存器和光标结束寄存器即可。这时, CRT C 就会形成由光标属性字定义的光标图形。关于光标属性字各位的具体意义, 请参阅 M6845 的有关技术手册。在图形方式下, 光标的产生由程序来实现, 这段程序被称为光标产生模块。此模块是 2 号功能块中的一个子程序。

它在执行中要对一些标志字节进行判断,对这些标志字节的不同内容,此模块将完成不同的功能。常用的功能是:清除光标、产生光标和直接返回。光标产生模块的具体执行步骤如下:(入口地址为CS:1AAD)

- 1) 对一些标志字节进行判别,以决定所要完成的功能。
- 2) 根据光标的当前位置参数,换算出光标在RRAM中的相应位置。
- 3) 在字模缓冲区中形成一个字符空格,并把字模缓冲区中的内容送到光标在RRAM中的位置,即清除了原来的光标图形。
- 4) 判别是否要建立光标。如不要建立光标,则立即返回。否则继续执行下面各步。
- 5) 根据光标的当前位置参数,换算出光标在RRAM中的相应位置。
- 6) 在字模缓冲区中形成光标图形(CC-DOS用的是下划线),并把字模缓冲区中的内容送到光标在RRAM中的位置,即建立了光标图形。
- 7) 保存光标位置参数并返回。

2. 光标的定位

所谓光标定位,就是要根据用户指定的位置,把光标移到该处。用户给出的光标位置是虚屏上的行号和列号,而光标在实屏上,故这里存在一个虚屏和实屏间的位置参数转换问题,即把虚屏上的行号和列号转换成RRAM中的相应地址。这些均由2号功能块来完成。在调用2号功能块实现光标定位前,要给出入口参数,即光标定位的位置参数(虚屏上的行号和列号)。我们把它称为新光标位置参数,而把原来屏幕上的光标的行号和列号称为原光标位置参数。

图6-2给出了2号功能块的流程。

在图6-2中,根据CRT的工作方式,分别转向字符方式光标定位模块和图形方式光标定位

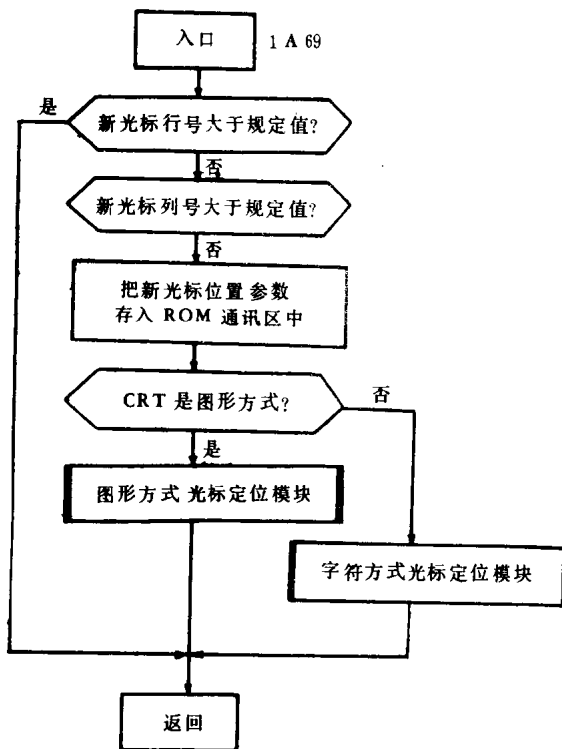


图 6-2 2号功能块流程

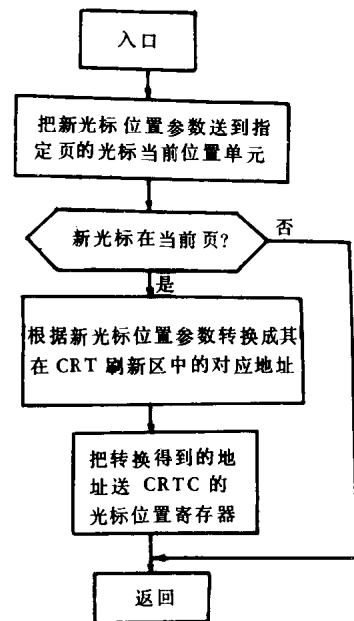


图 6-3 字符方式光标定位模块流程

模块，由它们分别去完成光标的定位工作。下面分析这两个模块。

先分析字符方式光标定位模块。在字符方式下对光标定位是较容易的。它是依靠 CRT C 来实现的。其执行流程如图 6-3 所示。

对图 6-3 要说明的是：所谓 CRT C 的光标位置寄存器，是 M6845 的十六个数据寄存器中的两个寄存器，即光标地址高位寄存器和光标地址低位寄存器，这两个寄存器的内容决定了光标在屏幕上的位置。

下面对图形方式光标定位模块进行分析。在图形方式下光标定位，要比字符方式下复杂得多，它的工作流程如图 6-4 所示。



图 6-4 圆形方式光标定位模块流程

下面对图 6-4 作一些解释。

一开始的删除原光标，是调用光标产生模块中的删除光标部分来完成的。因用户给出的新光

标位置参数是对虚屏而言的，而RRAM只是VRAM的一部分，所以有可能新光标位置不在RRAM中(即不在当前屏幕上)。因此要利用RRAM可以在VRAM中浮动的功能，通过修改RRAM首指针和RRAM尾指针之值，让RRAM在VRAM中上浮或下浮，使得新光标位置在RRAM中(即在屏幕上)。同时要把实屏上显示的内容上移或下移，这是通过移动RRAM中的内容来实现的。在完成这些工作后，新光标位置一定在实屏上。然后经过位置参数换算，再调用光标产生模块来形成光标，从而完成光标定位。

特别要指出的是，提示行中光标的产生和定位，不是本功能块实现的，而是由提示行管理模块中的有关子块来实现的。本章的第九节将对这个模块进行分析。

6.6 屏幕滚动的实现

CRT控制模块向用户提供了屏幕滚动的功能，用户可要求屏幕上的内容上滚或下滚，并可

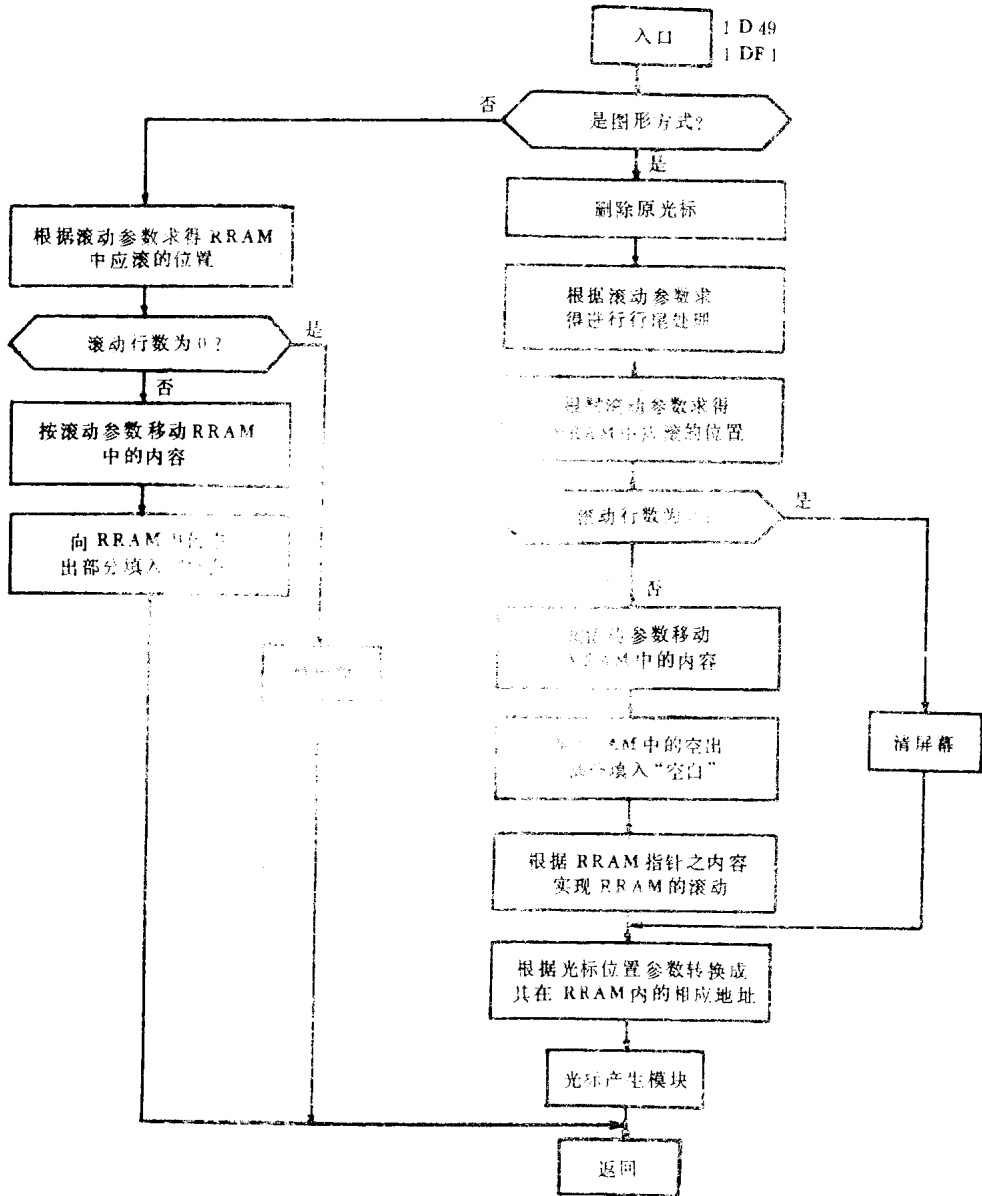


图 6 5 屏幕滚动的流程

指定欲滚动的区间和滚动的行数。由 6 号功能块和 7 号功能块分别完成屏幕的上滚或下滚。用户要求屏幕滚动, 先要提供滚动参数 (即滚动区间和滚动行数等), 然后再调用两个功能块中的某一个。下面对实现屏幕滚动的方法作一分析。

在字符方式下, 屏幕滚动是比较容易实现的, 把 R R A M 中的内容, 根据用户提供的滚动参数进行相应的移动即可。在图形方式下, 要实现屏幕的滚动就不是这么简单了。因用户给出的滚动参数是针对虚屏的, 故不能马上直接移动 R R A M 中的内容, 而要利用 V R A M 来模拟虚屏滚动的操作, 即先要根据滚动参数来滚动 V R A M 中的内容。当然, V R A M 的三个分区中的内容都要作相应的滚动。滚动后空出的部分要填入“空白”。然后, 根据 R R A M 首指针和 R R A M 尾指针的值, 算出 R R A M 要滚的参数, 根据这些参数来滚动 R R A M 中的内容。

在滚动的过程中要注意两点: 一点是不应把光标滚到不该到的地方, 故在滚动前要先删去原光标, 等滚动完成后再重新定位和建立光标。另一点是, 在滚动前要先进行行尾处理, 这是因为屏幕的滚动不一定是以整行来进行的, 所以滚动完成后就有可能出现汉字位置的错误, 即半个汉字显示在一行末, 而另外半个汉字显示在下行首。因此, 应在滚动前进行一次位置调整 (即行尾处理), 把应在行末显示的汉字, 调整到下一行的行首显示。经过这样调整之后, 就可避免在完成滚动以后, 出现汉字位置的错误。

图 6 5 给出了实现屏幕滚动流程

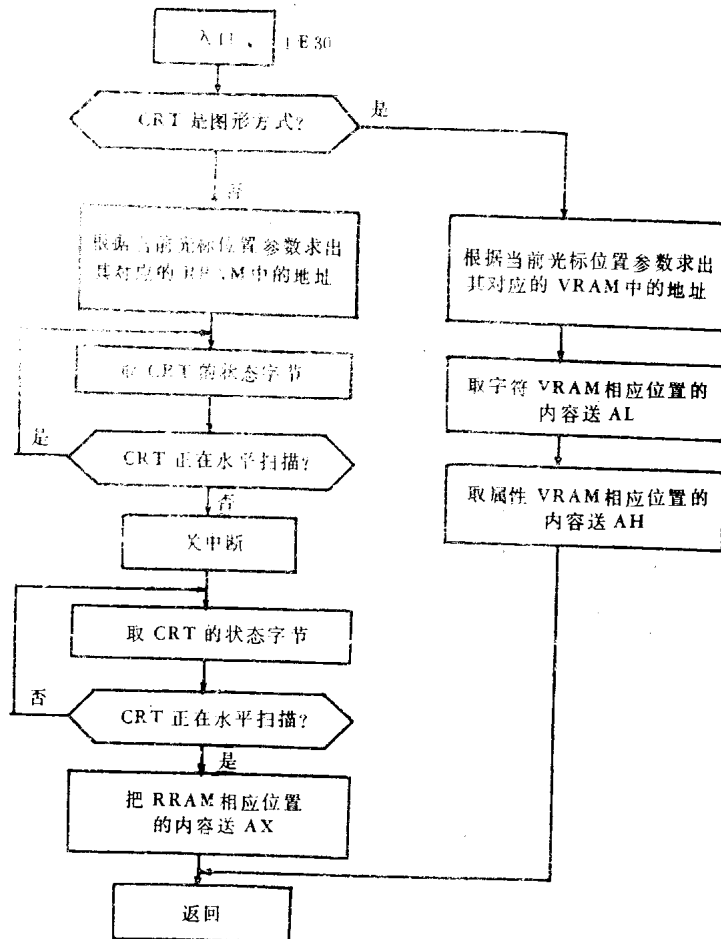


图 6 6 8 号功能块流程

6.7 读字符的实现

CRT控制块提供了读出当前光标位置上的字符内容的功能，这由8号功能块来实现。在字符方式下和图形方式下实现读字符的方法是不同的。在字符方式下读字符，只要把RRAM中的相应字符读出即可。不过，为防止RRAM中的内容在读出时发生变化，故在读字符之前要进行关中断，并要在CRT正处于水平扫描期间进行读取（因为向RRAM写字符，要在CRT处于水平回扫时进行）。在图形方式下读字符，不能去读RRAM中的内容，因为RRAM中内容是图形信息，而要把图形信息再转换成字符（代码）信息是不容易实现的。故应该去读VRAM中相应位置上的内容，VRAM中存放的是字符信息。在完成读字符操作后，要返回给调用者两个内容，一个是字符，另一个是它的属性字。这就要对字符VRAM和属性VRAM进行读操作。

8号功能块的流程如图6-6所示。由图可知，返回给调用者的参数是通过AX寄存器返回的。其中AL寄存器存放的是读到的字符，AH寄存器存放的是字符的属性字。

6.8 字符显示的实现

在屏幕的当前光标位置上显示字符，这是由10号功能块来完成的。该功能块所做的主要工作是，把汉字机内码转换成其对应的汉字字模，并把字模送屏幕上的相应位置显示。图6-7给出了该功能块的流程。下面对图6-7作几点说明。

1) 判别一个字符是否是汉字机内码，只要判其是否大于A0H即可。

2) 判别一个字符的相邻字符是否是机内码符，是通过判其相邻字符所对应的标志字来实现的（标志字存放在标志VRAM中）。当一个字符为机内码符时，则其对应的标志字为80H。

3) 为了避免一个汉字的一半显示在行末，而另一半显示在下一行的行首，故要进行行尾处理。其所做的主要工作是，当汉字显示的位置正好在行末时，则将其调整到下行行首去显示。

4) 因为汉字库中的字模是按其对应的国标码的次序来排列的，而本系统又采用异形国标码作机内码，故由机内码转换成其字模在汉字库中的地址是很容易的。

5) 汉字库的首址被保存在指定单元中（CS:0075—0076）。这是在CC-BIOS被引导时保存好的。

6) 字符字模库的首址由1FH类中断指针指出，这在CRT初始化时已置好。

7) 把汉字字模信息送入字模缓冲区时，为了保持屏幕的行间隔，故在最后要加送两行“空白”信息。当把字符字模信息送入字模缓冲区时，为了使字符与汉字保持一样的高度（字符高为8，汉字高为16），故在其字模信息前后均要送入几行“空白”信息。

6.9 TTY显示方式的实现

TTY显示方式是常用的显示方式，就是在当前光标位置上显示字符，然后光标自动后移一字符（如显示的是汉字，则要后移一汉字位置）。这个功能是由14号功能块实现的。如要显示一般字符，只要调用该功能块一次即可实现，如要显示汉字，则要调用该功能块两次才能实现，因为汉字机内码由两个字节组成。

14号功能块主要要完成下列工作：取当前光标的位置参数，为光标后移作准备；对特殊字符进行处理（包括退格符、回车符、换行符和报警符）；把字符送CRT当时光标位置显示；改变光标位置参数，实现光标后移，必要时还要进行滚行操作。

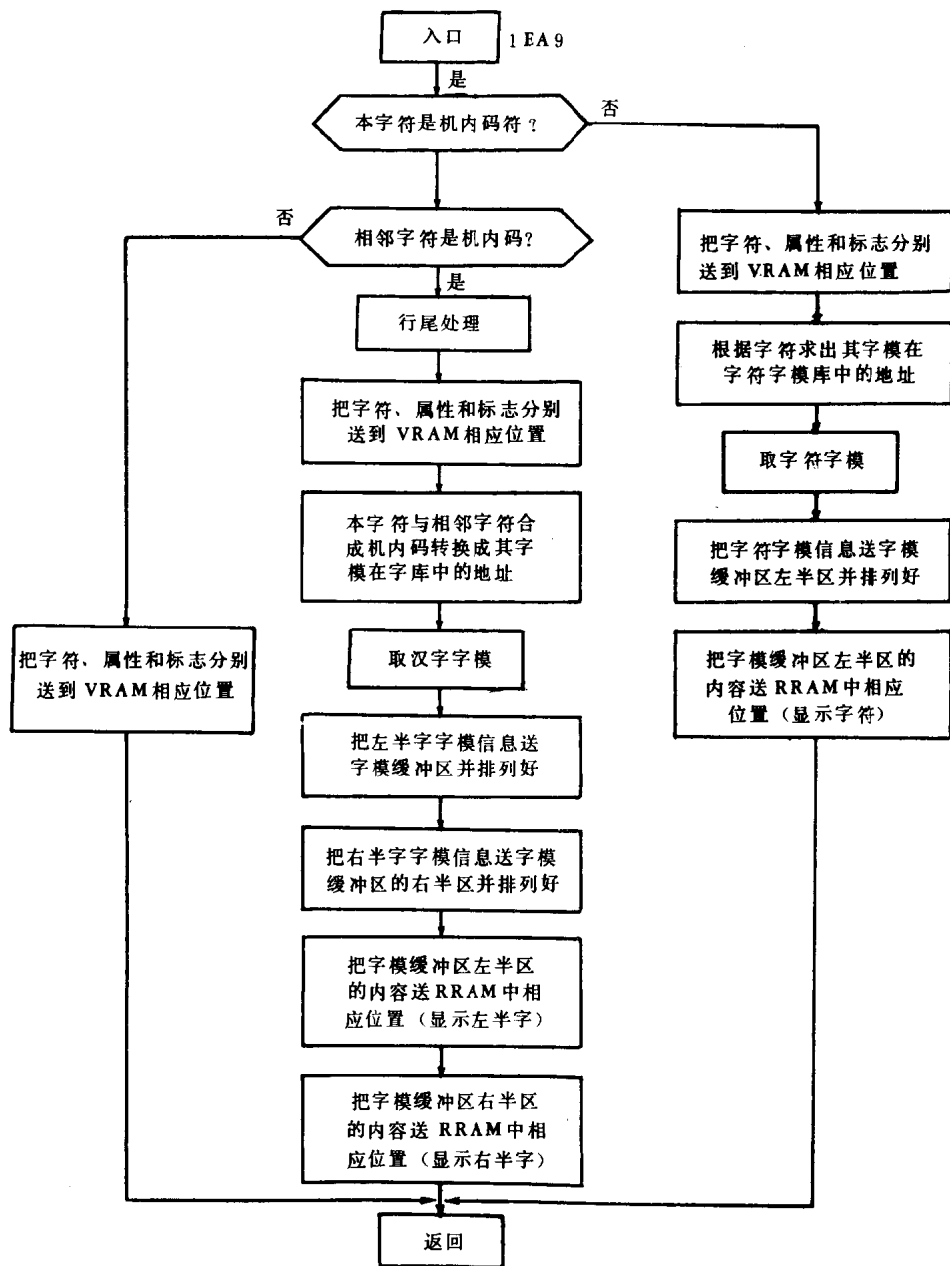


图 6-7 10号功能块流程

图 6-8 给出了 14 号功能块的流程。下面对图 6-8 作几点说明。

- 1) 光标定位，是通过调用 2 号功能块来实现的。
- 2) 所谓报警处理，就是使系统配置的喇叭发出一定频率的声音。其实现过程为：向定时器 8253 的 2 号通道置数，让它产生一定的脉冲，使可编程 I/O 口 8255 A 中 B 口的第 1 位和第 0 位置位，以选通 8253 向喇叭输出脉冲，并选通 1.19MHz 的时钟脉冲来调制 8253 产生的脉冲，从而使喇叭发出声音。隔一定时间后，使 8255 A 中 B 口的第 1 位和第 0 位复位，从而停止选通，喇叭发音结束。
- 3) 把字符送屏幕显示，是通过调用 10 号功能块来实现的。这个功能块已在 6.8 节中分析过了。

4) 屏幕向上滚动一行, 是通过调用 6 号功能块来实现的。这个功能块的情况已在 6.6 节中介绍过了。

6.10 提示行管理模块

CC DOS 把屏幕的最后一行规定为提示行, 用以显示提示信息和重码汉字。可见这一行是一特殊行。特殊行要特殊处理, 故在 CRT 控制模块中专门有一个提示行管理模块, 来实现对提

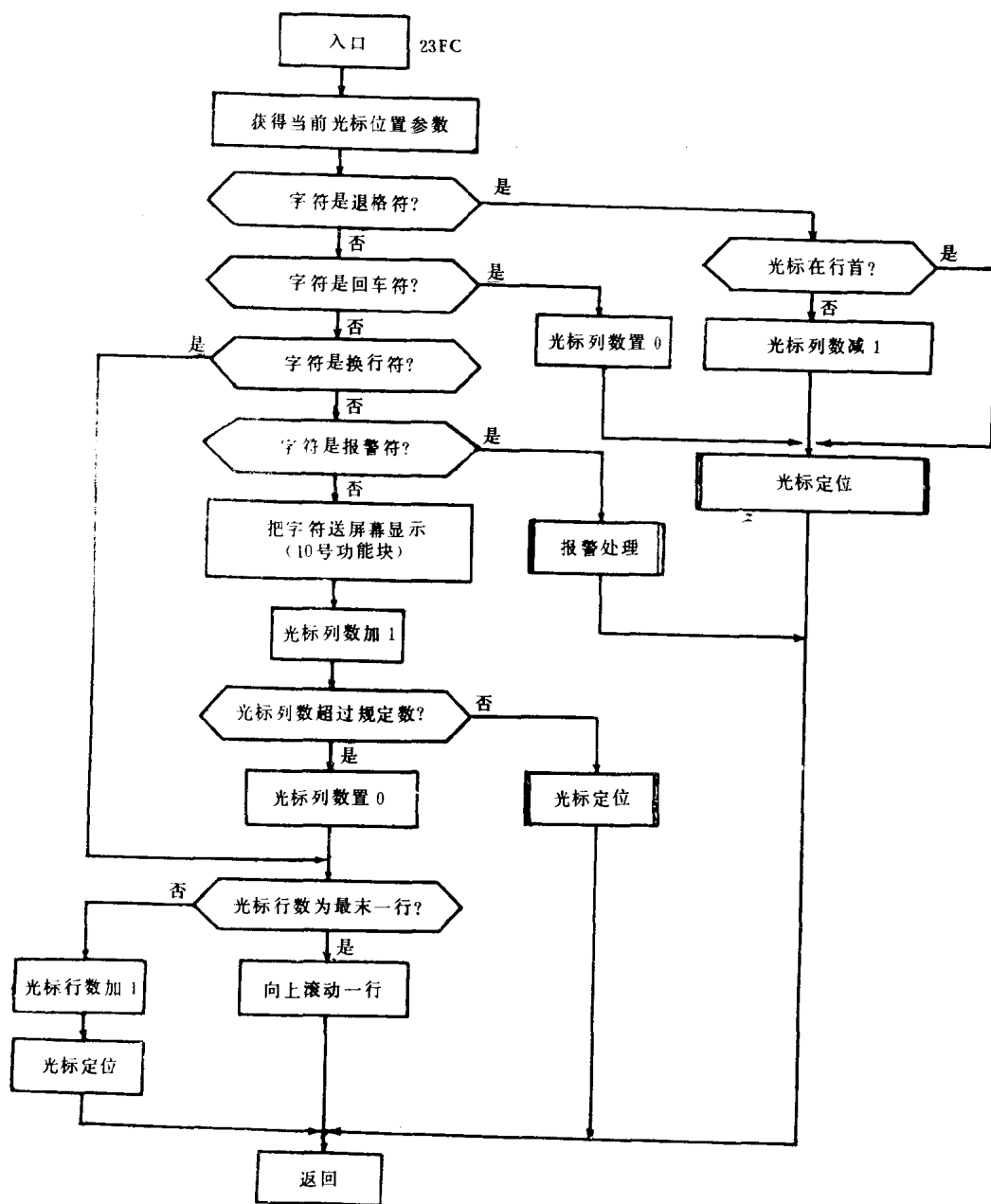


图 6-8 14号功能块流程

示行的管理。这个管理模块就是16号功能块。

提示行管理模块向用户提供了四种功能，每种功能均由16号功能块中的一个子块实现。所以，16号功能块中含有四个子块，它们的块号是0~3。每个子块所实现的提示行功能如表6-4所示。

提示行管理模块的子块

表6-4

子块块号	地 址	完 成 功 能	子块块号	地 址	完 成 功 能
0	CS:24C7	清除提示行中的内容	2	CS:24E9	提示行光标定位
1	CS:24F8	在提示行当前光标位置显示字符	3	CS:24FE	提示行TTY方式显示字符

用户要进行提示行操作，只要调用表6-4中的相应子块即可实现。调用方法是，把要调用的子块块号送入AL寄存器，把功能块号16送入AH寄存器中，再调用10H类软中断，即可实现对指定的子块的调用。

下面对这四个子块依次进行分析。

0号子块所完成的功能是：

- 1) 把提示行所对应的RRAM单元全部清为“空白”。
- 2) 在提示行首显示光标。提示行的光标是专门由提示行管理模块产生的，它的产生很简单，就是显示一下划线（其ASCII码为5FH）。只要把下划线的字模放到提示行行首所对应的RRAM的位置上，即完成了这一步的工作。
- 3) 在提示行的上方显示一条直线。这条直线把提示行与其他行分隔开来。只要把其在RRAM中的相应位全部置为1，即可实现。

1号子块所完成的功能是：

- 1) 清除原来的光标。只要把“空白”填入光标所在位置即可。
- 2) 调用提示行字符显示模块，实现字符的显示。

提示行字符显示模块是专门用来在提示行中的指定位置上实现字符显示的，其工作流程如图6-9所示。对图6-9要说明的是：所谓机内码标记，是用来记录前一字符是否是机内码符的；如前一字符是机内码符，机内码标记则置位。反之，则复位。所以，要判断前一字符是否为机内码符，只要判断机内码标记的值就可以了。

2号子块所完成的功能是：

- 1) 把原光标的位置参数转换成提示行在RRAM中对应的地址。提示行的原光标位置参数专门有两个单元来保存，它们的地址为CS:0056~0057。
- 2) 清除原光标。只要把“空白”送到RRAM中的光标对应位置即可。
- 3) 把新光标位置参数转换成提示行在RRAM中对应的地址，并保存新光标位置参数。新光标位置参数由调用者提供，它将被保存在CS:0056~0057中。
- 4) 在提示行的指定位置显示光标。即向RRAM中的相应位置送入下划线字符的字模。

3号子块所完成的功能是：

- 1) 对退格符进行退格处理，对报警符进行报警处理。
- 2) 在提示行的当前光标位置显示字符。即调用提示行显示字符模块（见图6-9）。
- 3) 光标后移一字符。先对光标位置参数中的列参数进行增值（如已到行末则不增），然后通过调用2号子块来实现光标的后移。

6.11 显示器控制块 INT 10H

INT 10H由CCDOS调入文件CCCC.exe执行后,修改为CCBIOS 10H,驻留内存。入口地址CS:1848,在此之前有显示器管理模块数据区与工作区(请参考前章)

CS: 0049 CRT工作方式单元

- = 00 40×25黑白字符 = 01彩色
- = 02 80×25黑白字符 = 03彩色
- = 04 320×200彩色图形 = 05黑白
- = 06 640×200黑白图形 = 07单色板

CS: 004A 屏幕列数单元

004C 刷新区长度单元

0050 光标单元(高字节行值,低字节列值)

0 页面

0052 1 页面

⋮

0060 7 页面

0075 汉字模库起始段地址

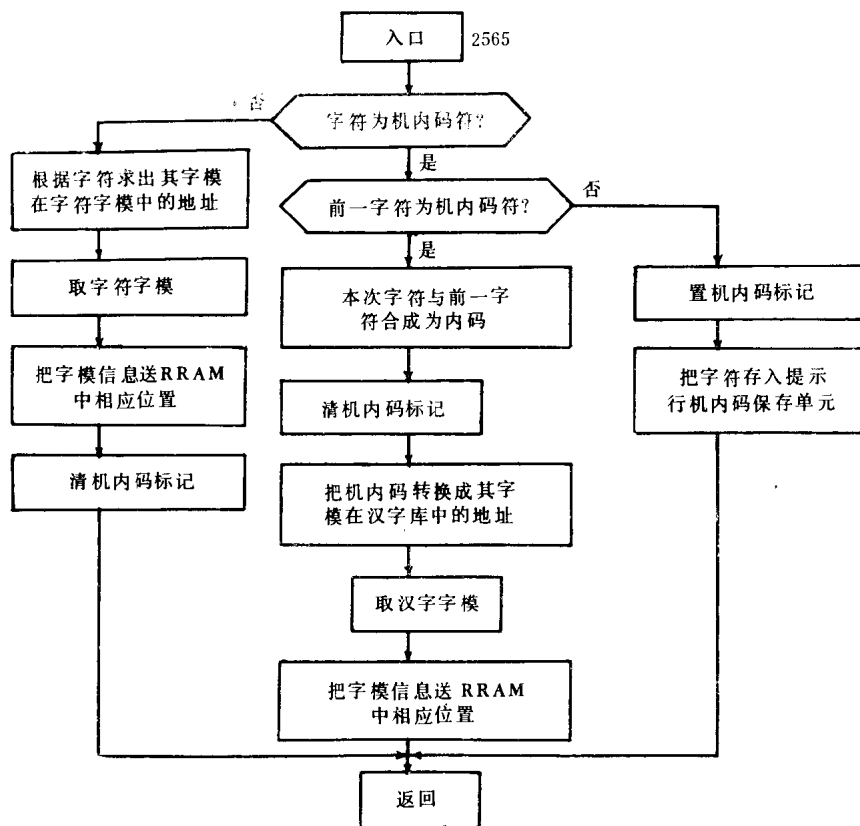


图 6-9 提示行字符显示模块流程

A:\c:\debug cccc.exe

-r

AX=0000 BX=0000 CX=AB44 DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=4DF8 ES=4DF8 SS=4E08 CS=4E08 IP=AAC0 NV UP DI PL NZ NA PO NC

4E08:AAC0 1E PUSH DS

-dcs:00

4E08:0000	00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00
4E08:0010	00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00
4E08:0020	00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00
4E08:0030	00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00
4E08:0040	00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00
4E08:0050	01 00 01 00 01 00 01 00-01 00 01 00 01 00 01 00
4E08:0060	00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00
4E08:0070	00 00 00 00 00 00 00 36-00 00 00 00 00 00 00 006.....

-g

4E08:0080	00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00
4E08:0090	00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00
4E08:00A0	00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00
4E08:00B0	00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00
4E08:00C0	00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00
4E08:00D0	00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00
4E08:00E0	00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00
4E08:00F0	00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00

-

-dcs:880 25行字符区属性

4E08:0880	00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00
4E08:0890	00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00
4E08:08A0	00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00
4E08:08B0	00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00
4E08:08C0	00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00
4E08:08D0	00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00
4E08:08E0	00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00
4E08:08F0	00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00

-

-dcs:1050 25行标志区

4E08:1050	00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00	内容与字符区对应
4E08:1060	00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00	80H没处理完汉字(第一字节)内码
4E08:1070	00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00	01处理完第一字节内码
4E08:1080	00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00	02处理完第2字节内码
4E08:1090	00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00
4E08:10A0	00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00
4E08:10B0	00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00
4E08:10C0	00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00

-d cs:1820 INT10H 功能块转移地址数据表—0—13H号功能入口地址

4E08:1820	E0 18 4B 1A 69 1A 65 1C-B3 24 9E 1C 49 1D F1 1D	.H.i.e.3\$.I.q.
4E08:1830	30 1E 76 1E A9 1E C2 1C-EB 1E DA 1E FC 23 EB 1C	0.v.).B.k.Z.1#h.

```

4E08:1840 B8 24 AD 25 C0 25 D3 25-FB FC 06 IE 52 51 53 56 B$-%%$%<!.RQSV
4E08:1850 57 55 50 8A C4 32 E4 D1-E0 8B F0 3D 28 00 72 04 WUP.D2dQ'.p=(.r.
4E08:1860 58 E9 DB 01 B8 08 4E 8E-DB 8B 00 8B 8B 3E 10 00 XiL.G.N.X8.B.>..
4E08:1870 81 E7 30 00 83 FF 30 75-03 8B 00 80 8E C0 58 8A .g0...0u.8.0.@X.
4E08:1880 26 49 00 2E FF A4 20 18-3B 2B 2D 0A 1F 06 19 1C &I...$.8(-.....
4E08:1890 02 07 06 07 00 00 00 00-71 50 5A 0A 1F 06 19 1C .....qPZ.....
-g
4E08:18A0 02 07 06 07 00 00 00 00-3B 2B 2D 0A 7F 06 64 70 .....8(-...dp
4E08:18B0 02 01 06 07 00 00 00 00-61 50 52 0F 19 06 19 19 .....aPR.....
4E08:18C0 02 0D 0B 0C 00 00 00 00-00 0B 00 10 00 40 00 40 .....@.@
4E08:18D0 28 28 50 50 28 28 50 50-2C 2B 2D 29 2A 2E 1E 29 ((PP((PP,(-)*..))
4E08:18E0 BA D4 03 B3 00 B3 FF 30-75 07 B0 07 BA B4 03 FE :T.3...0u.0.:4.*
4E08:18F0 C3 8A E0 A2 49 00 89 16-63 00 1E BE 40 00 8E DE C.'!...c...>@..^
4E08:1900 BE 49 00 8B 04 BE 63 00-89 14 50 52 83 C2 04 8A >I...>c...PR.B..
4E08:1910 C3 EE 5A 2B C0 8E DB 8D-06 9B 27 BE 7C 00 89 04 CnZ+@.X...'>!...

```

-ucs:1848

```

4E08:1848 FB          STI          : INT10H 入口
4E08:1849 FC          CLD          : 关中断
4E08:184A 06          PUSH        ES      : 清方向标志
4E08:184B 1E          PUSH        DS      : 保存各寄存器现场
4E08:184C 52          PUSH        DX
4E08:184D 51          PUSH        CX
4E08:184E 53          PUSH        BX
4E08:184F 56          PUSH        SI
4E08:1850 57          PUSH        DI
4E08:1851 55          PUSH        BP
4E08:1852 50          PUSH        AX
4E08:1853 8AC4         MOV        AL,AH    : 功能调用号→AL
4E08:1855 32E4         XOR        AH,AH    : 0→AH
4E08:1857 D1E0         SHL        AX,1     : 乘2转相应地址
4E08:1859 8BF0         MOV        SI,AX    : 存相应功能号地址
4E08:185B 3D2B00       CMP        AX,002B
4E08:185E 7204         JB         1864
4E08:1860 58          POP        AX       : 超值恢复AX转1A 3F 中断返回
4E08:1861 E9DB01       JMP        1A3F
4E08:1864 8B084E       MOV        AX,4E08  : 段地→DX
4E08:1867 8ED8       MOV        DS,AX

```

-j

```

4E08:1869 B800B8       MOV        AX,B800  : 彩显 口地址→AX
4E08:186C 8B3E1000    MOV        DI,[0010] : 设备设置参数→DI
4E08:1870 81E73000    AND        DI,0030
4E08:1874 83FF30       CMP        DI,+30   : 单色板? 彩显转187C
4E08:1877 7503         JNZ        187C
4E08:1879 8B00B0       MOV        AX,B000  : 单色口地址
4E08:187C 8EC0         MOV        ES,AX
4E08:187E 58          POP        AX

```

4E08:187F 8A264900 MOV AH,[0049] ; CRT工作方式→AH
 4E08:1883 2E CS: ;
 4E08:1884 FFA42018 JMP [SI+1820] ; 转向相应功能模块
 4E08:1888 3828 CMP [BX+SI],CH ; 1888起(6845初始化数据表)初始化40×25字符方式

INT10H 设置显示方式(0号功能)

-u18e0
 4E08:18E0 8AD403 MOV DX,03D4 ; 彩色口地址→DX
 4E08:18E3 B300 MOV BL,00 ; 彩色方式清BL
 4E08:18E5 83FF30 CMP DI,+30 ; 单色板? 不是转18F 1
 4E08:18E8 7507 JNZ 18F1 ; 是07→HL
 4E08:18EA B007 MOV AL,07 ;
 4E08:18EC 8AB403 MOV DX,03B4 ; 单色口地址
 4E08:18EF FEC3 INC BL ; 01→BL(黑白方式)
 4E08:18F1 8AE0 MOV AH,AL ; CRT工作方式→AH
 4E08:18F3 A24900 MOV [0049],AL ; CRT工作方式存相应单元
 4E08:18F6 89166300 MOV [0063],DX ; 插件板口地址→显示器插件单元
 4E08:18FA 1E PUSH DS ;
 4E08:18FB BE4000 MOV SI,0040 ; DS 指向DOS数据区
 4E08:18FE 8EDE MOV DS,SI ;
 -u
 4E08:1900 BE4900 MOV SI,0049 ;
 4E08:1903 8804 MOV [SI],AL ; 当前工作方式码送CRT方式单元
 4E08:1905 BE6300 MOV SI,0063 ;
 4E08:1908 8914 MOV [SI],DX ; 插件板地址送DOS数据区
 4E08:190A 50 PUSH AX ;
 4E08:190B 52 PUSH DX ;
 4E08:190C 83C204 ADD DX,+04 ; 保存插件板地址
 4E08:190F 8AC3 MOV AL,BL ;
 4E08:1911 EE OUT DX,AL ;
 4E08:1912 5A POP DX ; 恢复插件板地址
 4E08:1913 2BC0 SUB AX,AX ;
 4E08:1915 8ED8 MOV DS,AX ;
 4E08:1917 8D069827 LEA AX,[2798] ; 取字模库地址
 4E08:1918 BE7C00 MOV SI,007C ;
 4E08:191E 8904 MOV [SI],AX ; 取IFH中断向量指向ASC II字符模块
 -u
 4E08:1920 8CC8 MOV AX,CS ;
 4E08:1922 894402 MOV [SI+02],AX ;
 4E08:1925 C51E7400 LDS BX,[0074] ; 取10H中断向量指向DS:BX
 4E08:1929 58 POP AX ; 即CS:1888数据块CRT初始化
 4E08:192A 891000 MOV CX,0010 ; 16→CX,CRT初始化表每个表项字节数
 4E08:192D 80FC02 CMP AH,02 ;
 4E08:1930 7210 JB 1942 ; 模式0, 1, BX指向第一个表项, 转1942
 4E08:1932 03D9 ADD BX,CX ;
 4E08:1934 80FC04 CMP AH,04 ;
 4E08:1937 7209 JB 1942 ; 模式2, 3, BX指向第二个表项, 转1942执行
 4E08:1939 03D9 ADD BX,CX ;

```

4E08:193B 80FC07    CMP    AH,07
4E08:193E 7202    JB     1942      ; 模式 4, 5, 6, BX 指向第三个表项, 转1942
-u
4E08:1940 03D9    ADD    BX,CX
4E08:1942 50      PUSH   AX
4E08:1943 32E4    XOR    AH,AH
4E08:1945 8AC4    MOV    AL,AH
4E08:1947 EE      OUT    DX,AL    ; 内部寄存器号→AL
4E08:1948 42      INC    DX
4E08:1949 FEC4    INC    AH      ; 内部寄存器号+1
4E08:194B 8A07    MOV    AL,[BX]
4E08:194D EE      OUT    DX,AL
4E08:194E 43      INC    BX      ; BX 指向下一个表项数据
4E08:194F 4A      DEC    DX
4E08:1950 E2F3    LOOP  1945
4E08:1952 58      POP    AX
4E08:1953 1F      POP    DS
4E08:1954 33FF    XOR    DI,DI
4E08:1956 893E4E00  MOV   [004E],DI
4E08:195A C606620000  MOV   BYTE PTR [0062],00
4E08:195F B90020    MOV   CX,2000  ; 2000H →CX
-u
4E08:1962 90FC04    CMP    AH,04
4E08:1965 720C    JB     1973      ; 模式 1, 2, 3 转1973执行
4E08:1967 80FC07    CMP    AH,07
4E08:196A 7404    JZ     1970      ; 单色板, 转1970执行
4E08:196C 33C0    XOR    AX,AX
4E08:196E EB06    JMP    1976
4E08:1970 B90008    MOV   CX,0800  ; 0800H →CX
4E08:1973 B82007    MOV   AX,0720
4E08:1976 F3      REPZ
4E08:1977 AB      STOSW
4E08:1978 C70660006700  MOV   WORD PTR [0060],0067
4E08:197E A04900    MOV   AL,[0049] ; 当前CRT模式→AL
4E08:1981 32E4    XOR    AH,AH
-u
4E08:1983 8BF0    MOV    SI,AX
4E08:1985 8B166300  MOV   DX,[0063] ; 插件板地址→DX
4E08:1989 83C204    ADD    DX,+04
4E08:198C 2E      CS:
4E08:198D 8AB40818  MOV   AL,[SI+18D8]
4E08:1991 EE      OUT    DX,AL    ; 控制字送操作模式寄存器
4E08:1992 A26500    MOV   [0065],AL
4E08:1995 1E      PUSH   DS
4E08:1996 BF4000    MOV   DI,0040  ; DS = 0040
4E08:1999 8EDF    MOV   DS,DI
4E08:199B BF6500    MOV   DI,0065  ; 取DOS 数据区操作模式单元地址
4E08:199E 8B05    MOV   [DI],AL

```

4E08:19A0 1F	POP	DS	
4E08:19A1 2E	CS:		
4E08:19A2 8AB4D01B	MOV	AL,[SI+18D0]	
-u			
4E08:19A6 32E4	XOR	AH,AH	
4E08:19A8 A34A00	MOV	[004A],AX	
4E08:19AB B1E60E00	AND	SI,000E	: 为查表变换当前模式数字
4E08:19AF 2E	CS:		
4E08:19B0 8B8CC818	MOV	CX,[SI+18C8]	
4E08:19B4 890E4C00	MOV	[004C],CX	
4E08:19B8 BF5000	MOV	DI,0050	: DI指向0页面光标位置单元
4E08:19BB 33C0	XOR	AX,AX	
4E08:19BD 8905	MOV	[DI],AX	
4E08:19BF 884505	MOV	[DI+05],AL	
4E08:19C2 1E	PUSH	DS	
4E08:19C3 B84000	MOV	AX,0040	
-u			
4E08:19C6 8ED8	MOV	DS,AX	
4E08:19C8 C7050000	MOV	WORD PTR [DI],0000	
4E08:19CC 1F	POP	DS	
4E08:19CD 42	INC	DX	
4E08:19CE B030	MOV	AL,30	
4E08:19D0 803E490006	CMP	BYTE PTR [0049],06	
4E08:19D5 7503	JNZ	19DA	
4E08:19D7 A07700	MOV	AL,[0077]	
4E08:19DA EE	OUT	DX,AL	
4E08:19DB A26600	MOV	[0066],AL	: 存颜色寄存器单元
4E08:19DE 803E490004	CMP	BYTE PTR [0049],04	
4E08:19E3 725A	JB	1A3F	
4E08:19E5 803E490007	CMP	BYTE PTR [0049],07	
-u			
4E08:19EA 7453	JZ	1A3F	
4E08:19EC 33C0	XOR	AX,AX	
4E08:19EE F7D0	NOT	AX	
4E08:19F0 A37100	MOV	[0071],AX	
4E08:19F3 A3AD00	MOV	[00AD],AX	: FFFFH → 读操作时的内光标单元
4E08:19F6 F7D0	NOT	AX	
4E08:19F8 A3A700	MOV	[00A7],AX	: 0 → 25行区光标单元
4E08:19FB 3026730030	AND	BYTE PTR [0073],30	
4E08:1A00 B91200	MOV	CX,0012	: 18 → CX
4E08:1A03 1E	PUSH	DS	
4E08:1A04 07	POP	ES	
4E08:1A05 BF7B00	MOV	DI,007B	: 字模加工区18个字节清0
4E08:1A08 F3	REPZ		
4E08:1A09 AB	STOSW		
-u			
4E08:1A0A A29C00	MOV	[009C],AL	
4E08:1A0D B90400	MOV	CX,0004	

4E08:1A10 BFA100	MOV	DI,00A1	
4E08:1A13 F3	REPZ		: 清提示行内码缓冲区
4E08:1A14 AA	STOSB		
4E08:1A15 C606A50000	MOV	BYTE PTR [00A5],00	: 0 → 汉字屏起始行单元
4E08:1A1A C606A60009	MOV	BYTE PTR [00A6],09	: 9 → 汉字库结束行单元
4E08:1A1F A3AB00	MOV	[00AB],AX	: 0 → 读操作时的外光标单元
4E08:1A22 A3A900	MOV	[00A9],AX	: 0 → 写操作时的外光标单元
4E08:1A25 B9D007	MOV	CX,07D0	: 2000 → CX
4E08:1A28 BFB000	MOV	DI,00B0	
-u			
4E08:1A2B F3	REPZ		: 清25行字符区
4E08:1A2C AA	STOSB		
4E08:1A2D B9D007	MOV	CX,07D0	
4E08:1A30 BF5010	MOV	DI,1050	: 取25行标志区首址
4E08:1A33 F3	REPZ		: 清25行标志区
4E08:1A34 AA	STOSB		
4E08:1A35 B9D007	MOV	CX,07D0	
4E08:1A38 BFB008	MOV	DI,08B0	
4E08:1A3B B003	MOV	AL,03	
4E08:1A3D F3	REPZ		
4E08:1A3E AA	STOSB		
4E08:1A3F 5D	POP	BP	
4E08:1A40 5F	POP	DI	
4E08:1A41 5E	POP	SI	
4E08:1A42 5B	POP	BX	
4E08:1A43 59	POP	CX	
4E08:1A44 5A	POP	DX	
4E08:1A45 1F	POP	DS	
4E08:1A46 07	POP	ES	
4E08:1A47 CF	IRET		
4E08:1A48 B40A	MOV	AH,0A	: 设置6845内部寄存器号10
4E08:1A4A 890E6000	MOV	[0060],CX	
-u			
4E08:1A4E EB0200	CALL	1A53	: 调子程序写10号, 11号内部寄存器
4E08:1A51 EBEC	JMP	1A3F	
4E08:1A53 8B166300	MOV	DX,[0063]	
4E08:1A57 8AC4	MOV	AL,AH	: 6845索引寄存器指向AH号内部寄存器
4E08:1A59 EE	OUT	DX,AL	
4E08:1A5A 42	INC	DX	
4E08:1A5B 8AC5	MOV	AL,CH	
4E08:1A5D EE	OUT	DX,AL	
4E08:1A5E 4A	DEC	DX	: DX 指向索引寄存器端口
4E08:1A5F 8AC4	MOV	AL,AH	
4E08:1A61 FEC0	INC	AL	
4E08:1A63 EE	OUT	DX,AL	
4E08:1A64 42	INC	DX	: DX 指向索引寄存器端口
4E08:1A65 8AC1	MOV	AL,CL	
4E08:1A67 EE	OUT	DX,AL	

4E08:1A68 C3	RET		
4E08:1A69 80FE19	CMP	DH,19	: 设置光标行号超过范围
4E08:1A6C 7202	JB	1A70	
-u			
4E08:1A6E EBCF	JMP	1A3F	: 超过, 什么也不作, 转 1 A 3 F 中断返回
4E08:1A70 3A164A00	CMP	DL,[004A]	: 置光标列号超过范围?
4E08:1A74 7202	JB	1A7B	
4E08:1A76 EBC7	JMP	1A3F	: 超过, 什么也不作, 转 1 A 3 F 中断返回
4E08:1A7B B94000	MOV	CX,0040	
4E08:1A7B 1E	PUSH	DS	
4E08:1A7C 8ED9	MOV	DS,CX	
4E08:1A7E 89165000	MOV	[0050],DX	
4E08:1A82 1F	POP	DS	
4E08:1A83 803E490004	CMP	BYTE PTR [0049],04	: 哪种工作模式?
4E08:1A8B 720A	JB	1A94	: 字符方式, 转 1 A 94 执行
4E08:1A8A 803E490007	CMP	BYTE PTR [0049],07	
-u			
4E08:1A8F 7403	JZ	1A94	
4E08:1A91 E99F00	JMP	1B33	: 图形方式, 转 1 B 33 执行
4E08:1A94 8ACF	MOV	CL,BH	: 负号→CL
4E08:1A96 32ED	XOR	CH,CH	
4E08:1A98 D1E1	SHL	CX,1	: 为寻址, $CX \times 2 \rightarrow CX$
4E08:1A9A 8BF1	MOV	SI,CX	
4E08:1A9C 89945000	MOV	[SI+0050],DX	: 新光标位置→设置页面光标位置单元
4E08:1AA0 383E6200	CMP	[0062],BH	: 设置页面是当前显示页?
4E08:1AA4 7505	JNZ	1AAB	
4E08:1AA6 BBC2	MOV	AX,DX	: 是, 光标位置→AX
4E08:1AAB E8A901	CALL	1C54	: 将新光标写入光标寄存器
4E08:1AAB EB92	JMP	1A3F	
4E08:1AAD F6065500FF	TEST	BYTE PTR [0055],FF	
-u			
4E08:1AB2 747E	JZ	1B32	
4E08:1AB4 F6065400FF	TEST	BYTE PTR [0054],FF	: 测试自动光标标志
4E08:1AB9 7516	JNZ	1AD1	: 建立自动光标状态, 转 1 A D 1 处理光标
4E08:1ABB F606AF00FE	TEST	BYTE PTR [00AF],FE	
4E08:1AC0 7470	JZ	1B32	
4E08:1AC2 F606AF0001	TEST	BYTE PTR [00AF],01	
4E08:1AC7 750B	JNZ	1AD1	
4E08:1AC9 800EAF0001	OR	BYTE PTR [00AF],01	
4E08:1ACE EB62	JMP	1B32	
4E08:1AD0 90	NOP		
4E08:1AD1 A15000	MOV	AX,[0050]	: 当前内光标位置→AX
-u			
4E08:1AD4 50	PUSH	AX	
4E08:1AD5 A15200	MOV	AX,[0052]	: 原内光标外置→AX
4E08:1AD8 A35000	MOV	[0050],AX	
4E08:1ADB E8DC0B	CALL	23BA	: AX 返回原光标位置在刷新区的地址
4E08:1ADE 803E490006	CMP	BYTE PTR [0049],06	: 640 × 200 图形方式?

4E08:1AE3 7402	JZ	1AE7	: 是, 转 1AE7
4E08:1AE5 D1E0	SHL	AX,1	
4E08:1AE7 058002	ADD	AX,0280	
4E08:1AEA 8BF0	MOV	SI,AX	
4E08:1AEC 26	ES:		
4E08:1AED F604FF	TEST	BYTE PTR [SI],FF	: 原内光标位置有光标吗?
4E08:1AF0 7410	JZ	1B02	
4E08:1AF2 26	ES:		
4E08:1AF3 C60400	MOV	BYTE PTR [SI],00	
-u			
4E08:1AF6 803E490006	CMP	BYTE PTR [0049],06	: 640 × 200 图形方式?
4E08:1AFB 7405	JZ	1B02	
4E08:1AFD 26	ES:		
4E08:1AFE C6440100	MOV	BYTE PTR [SI+01],00	
4E08:1B02 8F065000	POP	[0050]	
4E08:1B06 83FD01	CMP	BP,+01	
4E08:1B09 7421	JZ	1B2C	
4E08:1B0B EBAC08	CALL	238A	: AX 返回新光标在刷新区的地址
4E08:1B0E 803E490006	CMP	BYTE PTR [0049],06	
4E08:1B13 7402	JZ	1B17	: 是, 转 1B17
4E08:1B15 D1E0	SHL	AX,i	
-u			
4E08:1B17 058002	ADD	AX,0280	
4E08:1B1A 8BF0	MOV	SI,AX	
4E08:1B1C 26	ES:		
4E08:1B1D C604FF	MOV	BYTE PTR [SI],FF	
4E08:1B20 803E490006	CMP	BYTE PTR [0049],06	
4E08:1B25 7405	JZ	1B2C	
4E08:1B27 26	ES:		
4E08:1B28 C64401FF	MOV	BYTE PTR [SI+01],FF	: 彩色图形方式, 再设置 1 个字节
4E08:1B2C A15000	MOV	AX,[0050]	: 新(当前)内光标位→原内光标位置单元
4E08:1B2F A35200	MOV	[0052],AX	
4E08:1B32 C3	RET		
4E08:1B33 BD0100	MOV	BP,0001	
4E08:1B36 EB74FF	CALL	1AAD	
-u			
4E08:1B39 80FE18	CMP	DH,18	: 新外光标行是 25 行区的提示行?
4E08:1B3C 7503	JNZ	1B41	
4E08:1B3E EB15	JMP	1B55	: 是, 转 1B55 执行
4E08:1B40 90	NDP		
4E08:1B41 32FF	XOR	BH,BH	
4E08:1B43 A0A500	MOV	AL,[00A5]	: 汉字屏起始行→AL
4E08:1B46 3AF0	CMP	DH,AL	
4E08:1B48 721C	JB	1B66	
4E08:1B4A A0A600	MOV	AL,[00A6]	: 汉字屏结束行→AL
4E08:1B4D 3AC6	CMP	AL,DH	
4E08:1B4F 725B	JB	1BAC	
4E08:1B51 2A36A500	SUB	DH,[00A5]	
4E08:1B55 89165000	MOV	[0050],DX	: 新外光标行转换成新内光标行


```

-u
4E08:1B59 C6065500FF  MOV  BYTE PTR [0055],FF      ; 设置光标标志 1, 允许处理光标
4E08:1B5E 33ED        XOR  BP,BP
4E08:1B60 EB4AFF      CALL 1AAD                    ; 抹原光标, 设置新光标
4E08:1B63 E9D9FE      JMP  1A3F
4E08:1B66 52         PUSH DX
4E08:1B67 2AC6        SUB  AL,DH
4E08:1B69 3C09        CMP  AL,09                  ; 滚动行数 ≥ 9?
4E08:1B6B 7202        JB   1B6F                    ; ≥ 9, 作 9 行处理
4E08:1B6D B009        MOV  AL,09
4E08:1B6F 33C9        XOR  CX,CX
4E08:1B71 B609        MOV  DH,09
4E08:1B73 8A164A00    MOV  DL,[004A]              ; 屏幕列数 - 1 → DL
4E08:1B77 FECA        DEC  DL

-u
4E08:1B79 50         PUSH AX
4E08:1B7A E8C005    CALL 213D                    ; 将不被滚出屏幕的行滚到屏幕下部
4E08:1B7D 5B         POP  BX                      ; 恢复向下滚动行数 → BL → BH
4E08:1B7E 8AFB        MOV  BH,BL
4E08:1B80 FECF        DEC  BH
4E08:1B82 5A         POP  DX                      ; 恢复新外光标行列值
4E08:1B83 52         PUSH DX
4E08:1B84 8BC2        MOV  AX,DX                   ; 新外光标行 → AX
4E08:1B86 8826A500    MOV  [00A5],AH
4E08:1B8A 80C409    ADD  AH,09
4E08:1B8D 8826A600    MOV  [00A6],AH
4E08:1B91 02F7        ADD  DH,BH
4E08:1B93 32D2        XOR  DL,DL
4E08:1B95 8BFA        MOV  DI,DX
4E08:1B97 EB5D00    CALL 1BF7                    ; 屏幕空留区滚动一行字符

-u
4E08:1B9A 80FF00    CMP  BH,00
4E08:1B9D 7408        JZ   1BA7
4E08:1B9F FECF        DEC  BH
4E08:1BA1 8BD7        MOV  DX,DI                   ; 屏幕空留区写字符指针 - 1
4E08:1BA3 FECE        DEC  DH
4E08:1BA5 EBEC        JMP  1B93
4E08:1BA7 5A         POP  DX                      ; 恢复新外光标的行列值
4E08:1BA8 32F6        XOR  DH,DH
4E08:1BAA EBA9        JMP  1B55
4E08:1BAC 52         PUSH DX
4E08:1BAD 2AF0        SUB  DH,AL
4E08:1BAF 8AC6        MOV  AL,DH
4E08:1BB1 3C09        CMP  AL,09                  ; 滚动行数超过 9 行?
4E08:1BB3 7202        JB   1BB7                    ; 0 → CX
4E08:1BB5 B009        MOV  AL,09                  ; 9 → DX
4E08:1BB7 33C9        XOR  CX,CX
4E08:1BB9 B609        MOV  DH,09

-u

```

4E08:18BB 8A164A00	MOV	DL,[004A]	
4E08:18BF FECA	DEC	DL	
4E08:18C1 50	PUSH	AX	
4E08:18C2 E8B604	CALL	204B	: 将不被滚出的行滚到屏幕上部
4E08:18C5 5B	POP	BX	
4E08:18C6 FECB	DEC	BL	
4E08:18CB B709	MOV	BH,09	
4E08:18CA 2AFB	SUB	BH,BL	
4E08:18CC 5A	POP	DX	
4E08:18CD 8BC2	MOV	AX,DX	
4E08:18CF 52	PUSH	DX	
4E08:18D0 8826A600	MOV	[00A6],AH	
4E08:18D4 80EC09	SUB	AH,09	
4E08:18D7 8826A500	MOV	[00A5],AH	
-u			
4E08:18DB 2AF3	SUB	DH,BL	
4E08:18DD 32D2	XOR	DL,DL	
4E08:18DF 8BFA	MOV	DI,DX	
4E08:18E1 EB1300	CALL	1BF7	
4E08:18E4 80FF09	CMP	BH,09	: 空留区全写完了吗?
4E08:18E7 7408	JZ	1BF1	: 写完, 转 1 B F 1
4E08:18E9 FEC7	INC	BH	: 没有, 屏幕向上滚动的行数 + 1
4E08:18EB 8BD7	MOV	DX,DI	
4E08:18ED FEC6	INC	DH	
4E08:18EF EBEC	JMP	1BDD	
4E08:18F1 5A	POP	DX	: 恢复新外光标在25行区的行列值
4E08:18F2 B609	MOV	DH,09	
4E08:18F4 E95EFF	JMP	1B55	
4E08:18F7 A15500	MOV	AX,[0055]	
4E08:18FA 50	PUSH	AX	
-u			
4E08:18FB C606550000	MOV	BYTE PTR [0055],00	
4E08:1C00 A1A100	MOV	AX,[00A1]	
4E08:1C03 A36C00	MOV	[006C],AX	
4E08:1C06 A1A300	MOV	AX,[00A3]	
4E08:1C09 A36E00	MOV	[006E],AX	
4E08:1C0C A17100	MOV	AX,[0071]	: 取写汉字光标
4E08:1C0F 50	PUSH	AX	: 压入堆栈保存
4E08:1C10 B90100	MOV	CX,0001	
4E08:1C13 32DB	XOR	BL,BL	
4E08:1C15 891E5000	MOV	[0050],BX	
4E08:1C19 8BC7	MOV	AX,BI	: 25行区取字符指针 → AX
-u			
4E08:1C1B E8F300	CALL	1D11	: AX 返回25行区取字符地址参数
4E08:1C1E D1FB	SAR	AX,1	
4E08:1C20 8BF0	MOV	SI,AX	
4E08:1C22 8A84B000	MOV	AL,[SI+00B0]	
4E08:1C26 8A9C8008	MOV	BL,[SI+08B0]	

4E08:1C2A B40A	MOV	AH,0A	: 10H 中断 0 AH 功能, 当前内光标位置写字符
4E08:1C2C CD10	INT	10	
4E08:1C2E 47	INC	DI	
4E08:1C2F A15000	MOV	AX,[0050]	: 取当前内光标位置
4E08:1C32 FE00	INC	AL	
4E08:1C34 3A064A00	CMP	AL,[004A]	: 一行写满了吗?
4E08:1C38 7215	JB	1C4F	
4E08:1C3A 58	POP	AX	: 一行写满, 恢复写汉字内光标
-u			
4E08:1C3B A37100	MOV	[0071],AX	
4E08:1C3E 58	POP	AX	
4E08:1C3F A35500	MOV	[0055],AX	
4E08:1C42 A16C00	MOV	AX,[006C]	: 恢复提示行内码缓冲区的数据
4E08:1C45 A3A100	MOV	[00A1],AX	
4E08:1C48 A16E00	MOV	AX,[006E]	
4E08:1C4B A3A300	MOV	[00A3],AX	
4E08:1C4E C3	RET		
4E08:1C4F A35000	MOV	[0050],AX	
4E08:1C52 EBC5	JMP	1C19	: AX 返回行列值在刷新区的地址位移
4E08:1C54 EBA300	CALL	1CFA	
4E08:1C57 8BC8	MOV	CX,AX	
4E08:1C59 030E4E00	ADD	CX,[004E]	
-u			
4E08:1C5D D1F9	SAR	CX,1	
4E08:1C5F B40E	MOV	AH,0E	
4E08:1C61 EBEFFD	CALL	1A53	: 将光标值 (CX) 写入光标地址寄存器
4E08:1C64 C3	RET		
4E08:1C65 803E490004	CMP	BYTE PTR [0049],04	: 哪种工作模式?
4E08:1C6A 721B	JB	1C87	
4E08:1C6C 803E490007	CMP	BYTE PTR [0049],07	
4E08:1C71 7414	JZ	1C87	
4E08:1C73 8B165000	MOV	DX,[0050]	: 图形方式, 当前内光标位置→DX
4E08:1C77 80FE17	CMP	DH,17	: 光标位置在提示行?
4E08:1C7A 7F04	JS	1C80	: 是, 转 1C 80
4E08:1C7C 0236A500	ADD	DH,[00A5]	
-u			
4E08:1C80 8916A700	MOV	[00A7],DX	
4E08:1C84 EB0B	JMP	1C91	
4E08:1C86 90	NOF		
4E08:1C87 8ADF	MOV	BL,BH	: 字符方式, 页号→BL
4E08:1C89 32FF	XOR	BH,BH	
4E08:1C8B D1E3	SHL	BX,1	
4E08:1C8D 8B975000	MOV	DX,[BX+0050]	
4E08:1C91 8B0E6000	MOV	CX,[0060]	: 光标类型→CX
4E08:1C95 5D	POP	BP	
4E08:1C96 5F	POP	DI	
4E08:1C97 5E	POP	SI	
4E08:1C98 5B	POP	BX	

4E08:1C99 58	POP	AX	
4E08:1C9A 58	POP	AX	
4E08:1C9B 1F	POP	DS	
4E08:1C9C 07	POP	ES	
4E08:1C9D CF	IRET		
4E08:1C9E A26200	MOV	[0062],AL	: 选择页号→当前显示单元
-u			
4E08:1CA1 8B0E4C00	MOV	CX,[004C]	
4E08:1CA5 98	CBW		: AL 中页号扩展成字
4E08:1CA6 50	PUSH	AX	
4E08:1CA7 F7E1	MUL	CX	
4E08:1CA9 A34E00	MOV	[004E],AX	
4E08:1CAC 8BC8	MOV	CX,AX	
4E08:1CAE D1F9	SAR	CX,1	
4E08:1CB0 840C	MOV	AH,0C	: 12→AH
4E08:1CB2 E89EFD	CALL	1A53	: 选择页号地址 (CX) → 起始地址寄存器
4E08:1CB5 58	POP	BX	
4E08:1CB6 D1E3	SHL	BX,1	: 对字计数, 乘 2
4E08:1CB8 8BB75000	MOV	AX,[BX+0050]	
4E08:1CBC EB95FF	CALL	1C54	: 写光标寄存器
4E08:1CBF E97DFD	JMP	1A3F	
-u			
4E08:1CC2 8B166300	MOV	DX,[0063]	: 插件板地址→ DX
4E08:1CC6 83C205	ADD	DX,05	
4E08:1CC9 A06600	MOV	AL,[0066]	: 取当前颜色寄存器的值→AL
4E08:1CCC 0AFF	OR	BH,BH	
4E08:1CCE 750E	JNZ	1CDE	
4E08:1CD0 24E0	AND	AL,E0	
4E08:1CD2 80E31F	AND	BL,1F	: 取新颜色值 (BL 的低 5 位)
4E08:1CD5 0AC3	OR	AL,BL	
4E08:1CD7 EE	OUT	DX,AL	: AL 的值写入插件板颜色寄存器
4E08:1CD8 A26600	MOV	[0066],AL	
4E08:1CDB E961FD	JMP	1A3F	
4E08:1CDE 24DF	AND	AL,DF	
4E08:1CE0 D0EB	SHR	BL,1	: 测试BL
-u			
4E08:1CE2 73F3	JNB	1CD7	: BL = 0 选择 0 色组
4E08:1CE4 0C20	OR	AL,20	: BL = 1, AL 第 5 位置 1
4E08:1CE6 EBEB	JMP	1CD7	
4E08:1CE8 8A264A00	MOV	AH,[004A]	: 屏幕列数→AH
4E08:1CEC A04900	MOV	AL,[0049]	: 当前工作模式→AL
4E08:1CEF 8A3E6206	MOV	BH,[0062]	: 当前正在显示的页号→BH
4E08:1CF3 5D	POP	BP	
4E08:1CF4 5F	POP	DI	
4E08:1CF5 5E	POP	SI	
4E08:1CF6 59	POP	CX	
4E08:1CF7 E949FD	JMP	1A43	
4E08:1CFA 803E490004	CMP	BYTE PTR [0049],04	

4E08:1CFF 7210	JB	1D11	: 字符方式, 转 1 D 11
4E08:1D01 803E490007	CMP	BYTE PTR [0049],07	
-u			
4E08:1D06 7409	JZ	1D11	
4E08:1D08 80FC17	CMP	AH,17	: 图形方式, 行值的最后一行?
4E08:1D0B 7F04	JG	1D11	: 是, 转 1 D 11
4E08:1D0D 0226A500	ADD	AH,[00A5]	
4E08:1D11 53	PUSH	EX	
4E08:1D12 8B08	MOV	BX,AX	: 行列值→BX
4E08:1D14 8AC4	MOV	AL,AH	: 行值→AL
4E08:1D16 F6264A00	MUL	BYTE PTR [004A]	
4E08:1D1A 32FF	XOR	BH,BH	
4E08:1D1C 03C3	ADD	AX,BX	: 加上列值
4E08:1D1E D1E0	SHL	AX,1	: 地址位移按字计算, 乘 2
4E08:1D20 5B	POP	BX	
4E08:1D21 C3	RET		
4E08:1D22 80FD17	CMP	CH,17	: 滚动窗口起始行≥23?
4E08:1D25 7202	JB	1D29	
-u			
4E08:1D27 B517	MOV	CH,17	: ≥23作23处理
4E08:1D29 80FE18	CMP	DH,18	: ≥24?
4E08:1D2C 7202	JB	1D30	
4E08:1D2E B618	MOV	DH,18	: ≥24作24处理
4E08:1D30 3A0E4A00	CMP	CL,[004A]	: 滚动窗口起始列值超出范围?
4E08:1D34 7206	JB	1D3C	
4E08:1D36 8A0E4A00	MOV	CL,[004A]	: 超出屏幕列数值-1→起始列值
4E08:1D3A FEC9	DEC	CL	
4E08:1D3C 3A164A00	CMP	DL,[004A]	
4E08:1D40 7206	JB	1D48	
4E08:1D42 8A164A00	MOV	DL,[004A]	: 超出, 屏幕列数-1→结束列值
4E08:1D46 FECA	DEC	DL	
-u			
4E08:1D48 C3	RET		
4E08:1D49 E8D6FF	CALL	1D22	: 判断滚动参数是否超出范围
4E08:1D4C 8ADB	MOV	BL,AL	: 滚动行数→BL
4E08:1D4E 80FC04	CMP	AH,04	: 字符方式
4E08:1D51 7208	JB	1D5B	
4E08:1D53 80FC07	CMP	AH,07	
4E08:1D56 7403	JZ	1D5B	
4E08:1D58 E9F601	JMP	1F51	: 图形方式, 转 1 F 51 执行
4E08:1D5B 53	PUSH	BX	
4E08:1D5C 8BC1	MOV	AX,CX	
4E08:1D5E EB3900	CALL	1D9A	: 为窗口滚动参数设置
4E08:1D61 7433	JZ	1D96	
4E08:1D63 03F0	ADD	SI,AX	
4E08:1D65 8AE6	MOV	AH,DH	: 滚动窗口中的行数→AH
4E08:1D67 2AE3	SUB	AH,BL	
-uu			

^ Error

~u

4E08:1D69 E87500	CALL	1DE1	: 在窗口中向上滚动一行
4E08:1D6C 03F5	ADD	SI, BP	
4E08:1D6E 03FD	ADD	DI, BP	
4E08:1D70 FECC	DEC	AH	
4E08:1D72 75F5	JNZ	1D69	
4E08:1D74 58	POP	AX	
4E08:1D75 B020	MOV	AL, 20	
4E08:1D77 E87000	CALL	1DEA	: 将滚动后窗口空留区清一行空白
4E08:1D7A 03FD	ADD	DI, BP	
4E08:1D7C FECE	DEC	BL	: 滚动行数 - 1
4E08:1D7E 75F7	JNZ	1D77	
4E08:1D80 8B084E	MOV	AX, 4E08	
4E08:1D83 8ED8	MOV	DS, AX	
4E08:1D85 803E490007	CMP	BYTE PTR [0049], 07	
~u			
4E08:1D8A 7407	JZ	1D93	
4E08:1D8C A06500	MOV	AL, [0065]	: 取显示器操作模式寄存器→AL
4E08:1D8F BAD803	MOV	DX, 03DB	
4E08:1D92 EE	OUT	DX, A	: 恢复操作模式寄存器
4E08:1D93 E9A9FC	JMP	1A3F	
4E08:1D96 8ADE	MOV	BL, 05	
4E08:1D98 E8DA	JMP	1D74	
4E08:1D9A 803E490002	CMP	BYTE PTR [0049], 02	: 哪种工作模式?
4E08:1D9F 7219	JB	1DBA	: 方式 0, 1 转 1 DBA 执行
4E08:1DA1 803E490003	CMP	BYTE PTR [0049], 03	
4E08:1DA6 7712	JA	1DBA	: 方式 4, 5, 6, 7 转 1 DBA 执行
4E08:1DAB 52	PUSH	DX	: 方式 2, 3 保存DX
4E08:1DA9 BADA03	MOV	DX, 03DA	: 插件板状态寄存器端口地址→DX
~u			
4E08:1DAC 50	PUSH	AX	
4E08:1DAD EC	IN	AL, DX	: 读状态寄存器
4E08:1DAE A808	TEST	AL, 08	
4E08:1DB0 74FB	JZ	1DAD	
4E08:1DB2 B025	MOV	AL, 25	
4E08:1DB4 BAD803	MOV	DX, 03DB	: 取操作模式寄存器端口地址
4E08:1DB7 EE	OUT	DX, AL	: 控制字→操作模式寄存器, 关闭显示
4E08:1DB8 58	POP	AX	
4E08:1DB9 5A	POP	DX	
4E08:1DBA E83DFF	CALL	1CFA	: AX 返回起始行列值在刷新区的位移地址
4E08:1DBD 03064E00	ADD	AX, [004E]	: 转换成当前页地址→AX
4E08:1DC1 8BF8	MOV	DI, AX	
4E08:1DC3 8BF0	MOV	SI, AX	
4E08:1DC5 2BD1	SUB	DX, CX	: 计算全窗口行列值→DX
4E08:1DC7 FEC6	INC	DH	
4E08:1DC9 FEC2	INC	DL	
4E08:1DCB 32ED	XOR	CH, CH	

```

-u
4E08:1DCD 8B2E4A00  MOV  BP,[004A]      ; 屏幕列数→BP
4E08:1DD1 03ED      ADD  BP,BP
4E08:1DD3 8AC3      MOV  AL,BL
4E08:1DD5 F6264A00  MUL  BYTE PTR [004A]
4E08:1DD9 03C0      ADD  AX,AX
4E08:1ddb 06        PUSH ES              ; 数据段指向CRT刷新区
4E08:1DDC 1F        POP  DS
4E08:1DDD 80FB00    CMP  BL,00
4E08:1DE0 C3        RET
4E08:1DE1 8ACA      MOV  CL,DL          ; 滚动窗口每行字节数→CL
4E08:1DE3 56        PUSH SI             ; 保存录址指针
4E08:1DE4 57        PUSH DI
4E08:1DE5 F3        REPZ
4E08:1DE6 A5        MOVSW              ; 滚动一行( SI ) 到窗口的指定位置
4E08:1DE7 5F        POP  DI
4E08:1DE8 5E        POP  SI
4E08:1DE9 C3        RET
4E08:1DEA 8ACA      MOV  CL,DL          ; 窗口每行字节数→CL
4E08:1DEC 57        PUSH DI             ; 保存寻址指针
-u
4E08:1DED F3        REPZ                ; AX →(DI)
4E08:1DEE AB        STOSW
4E08:1DEF 5F        POP  DI
4E08:1DF0 C3        RET
4E08:1DF1 EB2EFF    CALL 1D22           ; 判断滚动参数是否超出范围
4E08:1DF4 8AD8      MOV  BL,AL          ; 滚动行数-BL
4E08:1DF6 80FC04    CMP  AH,04
4E08:1DF9 7208      JB   1E03
4E08:1DFB 80FC07    CMP  AH,07
4E08:1DFE 7403      JZ   1E03
4E08:1E00 E9AE02    JMP  20B1
4E08:1E03 FD        STD
4E08:1E04 53        PUSH BX
4E08:1E05 8BC2      MOV  AX,DX
4E08:1E07 EB90FF    CALL 1D9A           ; 为窗口滚动设置参数
4E08:1E0A 7420      JZ   1E2C
4E08:1E0C 2BF0      SUB  SI,AX
-u
4E08:1E0E 8AE6      MOV  AH,DH
4E08:1E10 2AE3      SUB  AH,BL
4E08:1E12 EBCCFF    CALL 1DE1           ; 在窗口中向下滚动一行
4E08:1E15 2BF5      SUB  SI,BP
4E08:1E17 2BFD      SUB  DI,BP
4E08:1E19 FECC      DEC  AH
4E08:1E1B 75F5      JNZ  1E12
4E08:1E1D 58        POP  AX
4E08:1E1E B020      MOV  AL,20
4E08:1E20 EBC7FF    CALL 1DEA           ; 将滚动后窗口空留区清一行空白

```

4E08:1E23 28FD	SUB	DI,BP	
4E08:1E25 FECB	DEC	BL	
4E08:1E27 75F7	JNZ	1E20	
4E08:1E29 E954FF	JMP	1D80	
4E08:1E2C BADE	MOV	BL,DH	: 滚动窗口中的行数→BL
-u			
4E08:1E2E EBED	JMP	1E1D	: 转回 1 E 1 D , 将全窗口清成空白
4E08:1E30 80FC04	CMF	AH,04	: 哪种工作模式?
4E08:1E33 720B	JB	1E3D	: 字符方式, 转 1 E 3 D
4E08:1E35 80FC07	CMF	AH,07	
4E08:1E38 7403	JZ	1E3D	
4E08:1E3A E9AA05	JMP	23E7	: 图形方式, 转23E 7
4E08:1E3D E81A00	CALL	1E5A	: 返回BX 选择页号光标在刷新区的地址
4E08:1E40 8BF3	MOV	SI,BX	
4E08:1E42 8B166300	MOV	DX,[0063]	: DX 指向插件板状态寄存器
4E08:1E46 83C206	ADD	DX,+06	
4E08:1E49 06	PUSH	ES	
4E08:1E4A 1F	POP	DS	
4E08:1E4B EC	IN	AL,DX	: 读状态寄存器
4E08:1E4C A801	TEST	AL,01	: 扫描正程?
-u			
4E08:1E4E 75FB	JNZ	1E4B	
4E08:1E50 FA	CLI		
4E08:1E51 EC	IN	AL,DX	: 扫描正程, 屏蔽中断
4E08:1E52 A801	TEST	AL,01	
4E08:1E54 74FB	JZ	1E51	
4E08:1E56 AD	LDSW		: 读字符及属性→AX
4E08:1E57 E9E5FB	JMP	1A3F	
4E08:1E5A 8ACF	MOV	CL,BH	: 页号→CX
4E08:1E5C 32ED	XOR	CH,CH	
4E08:1E5E 8BF1	MOV	SI,CX	
4E08:1E60 D1E6	SHL	SI,1	: 为寻址, SI×2→SI
4E08:1E62 8B845000	MOV	AX,[SI+0050]	
4E08:1E66 33DB	XOR	BX,BX	
4E08:1E68 E306	JCXZ	1E70	
4E08:1E6A 031E4C00	ADD	BX,[004C]	
-u			
4E08:1E6E E2FA	LOOP	1E6A	
4E08:1E70 E887FE	CALL	1CFA	: AX 返回处理页面光标在刷新区的地址
4E08:1E73 03D8	ADD	BX,AX	
4E08:1E75 C3	RET		
4E08:1E76 80FC04	CMF	AH,04	: 字符工作方式?
4E08:1E79 720B	JB	1E83	: 字符方式, 转 1 E 83 执行
4E08:1E7B 80FC07	CMF	AH,07	
4E08:1E7E 7403	JZ	1E83	
4E08:1E80 E97D03	JMP	2200	: 图形方式, 转2200 执行
4E08:1E83 8AE3	MOV	AH,BL	
4E08:1E85 50	PUSH	AX	

4E08:1E86 51	PUSH	CX	
4E08:1E87 E8D0FF	CALL	1E5A	: BX 返回处理页面光标在刷新区的地址
4E08:1E8A 88FB	MOV	DI, BX	
4E08:1E8C 59	POP	CX	
4E08:1E8D 5B	POP	BX	
-u			
4E08:1E8E 8B166300	MOV	DX, [0063]	: 取插件板端口地址
4E08:1E92 83C206	ADD	DX, +06	
4E08:1E95 EC	IN	AL, DX	: 读状态寄存器
4E08:1E96 A801	TEST	AL, 01	
4E08:1E98 75FB	JNZ	1E95	
4E08:1E9A FA	CLI		
4E08:1E9B EC	IN	AL, DX	
4E08:1E9C A801	TEST	AL, 01	
4E08:1E9E 74FB	JZ	1E9B	
4E08:1EA0 88C3	MOV	AX, BX	
4E08:1EA2 AB	STOSW		
4E08:1EA3 FB	STI		
4E08:1EA4 E2EB	LOOP	1E8E	: CX - 1 → CX 若 CX ≠ 0 转回 1E8E 继续
4E08:1EA6 E996FB	JMP	1A3F	
4E08:1EA9 80FC04	CMP	AH, 04	: 字符方式?
4E08:1EAC 720B	JB	1EB6	: 字符方式。转 1EB6
-u			
4E08:1EAE 80FC07	CMP	AH, 07	
4E08:1EB1 7403	JZ	1EB6	
4E08:1EB3 E94A03	JMP	2200	: 图形方式转2200执行
4E08:1EB6 50	PUSH	AX	
4E08:1EB7 51	PUSH	CX	
4E08:1EB8 E89FFF	CALL	1E5A	: BX 返回选择页光标在刷新区的地址
4E08:1EBB 88FB	MOV	DI, BX	
4E08:1EBD 59	POP	CX	
4E08:1EBE 5B	POP	BX	
4E08:1EBF 8B166300	MOV	DX, [0063]	
4E08:1EC3 83C206	ADD	DX, +06	: DX 指向状态寄存器
4E08:1EC6 EC	IN	AL, DX	: 取状态字
4E08:1EC7 A801	TEST	AL, 01	
4E08:1EC9 75FB	JNZ	1EC6	
4E08:1ECB FA	CLI		
4E08:1ECC EC	IN	AL, DX	
4E08:1ECD A801	TEST	AL, 01	
-u			
4E08:1ECF 74FB	JZ	1ECC	
4E08:1ED1 8AC3	MOV	AL, BL	: 安写的字符→AL
4E08:1ED3 AA	STOSB		
4E08:1ED4 47	INC	DI	
4E08:1ED5 E2EB	LOOP	1EBF	: CX - 1 → CX, 若 CX ≠ 0 转回 1EBF 再写
4E08:1ED7 E965FB	JMP	1A3F	
4E08:1EDA EB3100	CALL	1F0E	: 设置像素点的各种参数

4E08:1EDD 26	ES:		
4E08:1EDE 8A04	MOV	AL,[SI]	
4E08:1EE0 22C4	AND	AL,AH	
4E08:1EE2 D2E0	SHL	AL,CL	
4E08:1EE4 BACE	MOV	CL,DH	
4E08:1EE6 D2C0	ROL	AL,CL	
4E08:1EE8 E954FB	JMP	1A3F	
4E08:1EEB 50	PUSH	AX	
4E08:1EEC 50	PUSH	AX	
4E08:1EED EB1E00	CALL	1F0E	: 设置象素点的各种参数, AL 高位返回象素点值
-u			
4E08:1EF0 D2E8	SHR	AL,CL	
4E08:1EF2 22C4	AND	AL,AH	
4E08:1EF4 26	ES:		
4E08:1EF5 8A0C	MOV	CL,[SI]	: 从刷新区读象素点所在的字节
4E08:1EF7 5B	POP	BX	
4E08:1EF8 F6C380	TEST	BL,B0	
4E08:1EFB 750B	JNZ	1F0A	
4E08:1EFD F6D4	NOT	AH	
4E08:1EFF 22CC	AND	CL,AH	
4E08:1F01 0AC1	OR	AL,CL	
4E08:1F03 26	ES:		
4E08:1F04 8B04	MOV	[SI],AL	: 刷新区写新象素点值
4E08:1F06 5B	POP	AX	
4E08:1F07 E935FB	JMP	1A3F	
4E08:1F0A 32C1	XOR	AL,CL	
4E08:1F0C EBF5	JMP	1F03	
4E08:1F0E 53	PUSH	BX	
4E08:1F0F 50	PUSH	AX	
-u			
4E08:1F10 B028	MOV	AL,28	: 40→AL
4E08:1F12 52	PUSH	DX	
4E08:1F13 80E2FE	AND	DL,FE	
4E08:1F16 F6E2	MUL	DL	
4E08:1F18 5A	POP	DX	
4E08:1F19 F6C201	TEST	DL,01	: 奇扫描行?
4E08:1F1C 7403	JZ	1F21	
4E08:1F1E 050020	ADD	AX,2000	: 奇扫描行, AX 指向奇扫描区象素点所在行的第一字节
4E08:1F21 8BF0	MOV	SI,AX	: 刷新区地址指针→SI
4E08:1F23 5B	POP	AX	
4E08:1F24 8BD1	MOV	DX,CX	
4E08:1F26 BBC002	MOV	BX,02C0	: 为彩色图形方式设置参数
4E08:1F29 B90203	MOV	CX,0302	
4E08:1F2C 803E490006	CMP	BYTE PTR [0049],06	: 640 × 200 图形方式
-u			
4E08:1F31 7206	JB	1F39	
4E08:1F33 8BB001	MOV	BX,0180	: 是, 为640 × 200 图形方式设置参数
4E08:1F36 B90307	MOV	CX,0703	

4E08:1F39 22EA	AND	CH,DL	
4E08:1F3B D3EB	SHR	BX,CL	
4E08:1F3D 03F2	ADD	SI,DX	: SI 指向 象 素 点 所 在 字 节 的 刷 新 区 地 址
4E08:1F3F BAF6	MOV	DH,DH	
4E08:1F41 2AC9	SUB	CL,CL	
4E08:1F43 D0CB	ROR	AL,1	: 要 写 的 象 素 点 值 移 到 AL 的 高 位
4E08:1F45 02CD	ADD	CL,CH	
4E08:1F47 FECF	DEC	BH	
4E08:1F49 75F8	JNZ	1F43	
4E08:1F4B 8AE3	MOV	AH,BL	: 屏 幕 码 参 数 → AH
4E08:1F4D D2EC	SHR	AH,CL	
4E08:1F4F 5B	POP	BX	
4E08:1F50 C3	RET		
-u			
4E08:1F51 32FF	XOR	BH,BH	
4E08:1F53 06	PUSH	ES	
4E08:1F54 50	PUSH	AX	
4E08:1F55 53	PUSH	BX	
4E08:1F56 51	PUSH	CX	
4E08:1F57 52	PUSH	DX	
4E08:1F58 50	PUSH	AX	
4E08:1F59 53	PUSH	BX	
4E08:1F5A 51	PUSH	CX	
4E08:1F5B 52	PUSH	DX	
4E08:1F5C B00100	MOV	BP,0001	
4E08:1F5F E848FB	CALL	1AAD	调 子 程 序 抹 去 原 光 标
4E08:1F62 5A	POP	DX	
4E08:1F63 59	POP	CX	
4E08:1F64 5B	POP	BX	
4E08:1F65 5B	POP	AX	
4E08:1F66 8AD8	MOV	BL,AL	
4E08:1F68 53	PUSH	BX	
4E08:1F69 8BC1	MOV	AX,CX	
4E08:1F6B E87200	CALL	1FE0	: 为 窗 口 滚 动 设 置 参 数
4E08:1F6E 744E	JZ	1FBE	
4E08:1F70 03F0	ADD	SI,AX	: 设 置 滚 动 原 指 针
-u			
4E08:1F72 8AE6	MOV	AH,DH	
4E08:1F74 2AE3	SUB	AH,BL	
4E08:1F76 7504	JNZ	1F7C	
4E08:1F78 5B	POP	AX	
4E08:1F79 EB18	JMP	1F93	
4E08:1F7B 90	NOP		
4E08:1F7C E88200	CALL	2001	: 将 不 被 滚 出 窗 口 的 行 滚 到 窗 口 的 上 部
4E08:1F7F 03F5	ADD	SI,BP	
4E08:1F81 03FD	ADD	DI,BP	
4E08:1F83 FECC	DEC	AH	
4E08:1F85 75F5	JNZ	1F7C	

4E08:1F87 58	POP	AX	
4E08:1F88 B000	MOV	AL,00	
4E08:1F8A E8A500	CALL	2032	
4E08:1F8D 03FD	ADD	DI,BP	
4E08:1F8F FECB	DEC	BC	
4E08:1F91 75F7	JNZ	1F8A	: 转回 1 F 8 A , 继续清空白区
-u			
4E08:1F93 5A	POP	DX	
4E08:1F94 59	POP	CX	
4E08:1F95 5B	POP	BX	
4E08:1F96 58	POP	AX	
4E08:1F97 07	POP	ES	
4E08:1F98 382EA600	CMP	[00A6],CH	: 汉字屏结束行 < 滚动窗口起始行?
4E08:1F9C 7235	JB	1FD3	
4E08:1F9E 8A26A500	MOV	AH,[00A5]	: 取汉字屏起始行
4E08:1FA2 3AF4	CMP	DH,AH	
4E08:1FA4 722D	JB	1FD3	
4E08:1FA6 3AE5	CMP	AH,CH	: 窗口起始行在汉字屏之内?
4E08:1FA8 7205	JB	1FAF	: 是, 转 1 FAF
4E08:1FAA 32ED	XOR	CH,CH	
4E08:1FAC EB03	JMP	1FB1	
4E08:1FAE 90	NOP		
4E08:1FAF 2AEC	SUB	CH,AH	: 修改窗口起始行, 改为汉字屏内的行号
4E08:1FB1 9A26A600	MOV	AH,[00A6]	: 取汉字屏结束行
-u			
4E08:1FB5 3AF4	CMP	DH,AH	
4E08:1FB7 7209	JB	1FC2	
4E08:1FB9 B609	MOV	DH,09	
4E08:1FBB EB09	JMP	1FC6	
4E08:1FBD 90	NOP		
4E08:1FBE BADE	MOV	BL,DH	
4E08:1FC0 EBC5	JMP	1FB7	
4E08:1FC2 2A36A500	SUB	DH,[00A5]	: 修改窗口结束行, 改为汉字屏内的行号
4E08:1FC6 3C09	CMP	AL,09	
4E08:1FC8 7202	JB	1FCC	
4E08:1FCA B009	MOV	AL,09	
4E08:1FCC 3AEE	CMP	CH,DH	
4E08:1FCE 7403	JZ	1FD3	
4E08:1FD0 E87800	CALL	204B	: 在刷新区实现窗口的向上滚动
4E08:1FD3 C6065500FF	MOV	BYTE PTR [0055],FF	
-u			
4E08:1FD8 33E0	XOR	BP,BP	
4E08:1FDA E8D0FA	CALL	1AAD	: 设置光标
4E08:1FDD E95FFA	JMP	1A3F	: 转 1 A 3 F , 中断返回
4E08:1FE0 EB2EFD	CALL	1D11	: AX 返回入口行列值在25行区的地址
4E08:1FE3 D1FB	SAR	AX,1	: AX 被除 2 , 得到25行的地址位移
4E08:1FE5 8BF8	MOV	DI,AX	
4E08:1FE7 8BF0	MOV	SI,AX	

4E08:1FE9 2BD1	SUB	DX,CX	: 窗口结束行列值—窗口起始行列→DX
4E08:1FEB FEC6	INC	DH	: DH + 1 → DH (全窗口行数)
4E08:1FED FEC2	INC	DL	: DL + 1 → DL (全窗口列数)
4E08:1FEF 32ED	XOR	CH,CH	
4E08:1FF1 8B2E4A00	MOV	BP,[004A]	: 屏幕列数→BP
4E08:1FF5 9AC3	MOV	AL,BL	
4E08:1FF7 F6264A00	MUL	BYTE PTR [004A]	: 滚动行数在25行区所占地址空间→AX
-u			
4E08:1FFB 1E	PUSH	DS	
4E08:1FFC 07	POP	ES	
4E08:1FFD 80FB00	CMP	BL,00	: 如果滚动行数 = 0, 置0标志
4E08:2000 C3	RET		
4E08:2001 BACA	MOV	CL,DL	
4E08:2003 56	PUSH	SI	
4E08:2004 57	PUSH	DI	
4E08:2005 81C6B000	ADD	SI,00B0	: 滚动原指针向25行字符区
4E08:2009 81C7B000	ADD	DI,00B0	: 滚动目的指针指向25行字符区
4E08:200D F3	REPZ		: 在25行字符区滚动一行
4E08:200E A4	MOVSB		
4E08:200F 5F	POP	DI	
4E08:2010 5E	POP	SI	
4E08:2011 56	PUSH	SI	
4E08:2012 57	PUSH	DI	
4E08:2013 81C5010	ADD	SI,1050	: 滚动指针指向25行标志区
4E08:2017 91C75010	ADD	DI,1050	
-u			
4E08:201B 8ACA	MOV	CL,DL	
4E08:201D F3	REPZ		: 在25行标志区滚动一行
4E08:201E A4	MOVSB		
4E08:201F 5F	POP	DI	
4E08:2020 5E	POP	SI	
4E08:2021 56	PUSH	SI	
4E08:2022 57	PUSH	DI	
4E08:2023 81C68008	ADD	SI,0880	
4E08:2027 81C78008	ADD	DI,0880	
4E08:202B 8ACA	MOV	CL,DL	
4E08:202D F3	REPZ		
4E08:202E A4	MOVSB		
4E08:202F 5F	POP	DI	
4E08:2030 5E	POP	SI	
4E08:2031 C3	RET		
4E08:2032 BACA	MOV	CL,DL	: 滚动窗口每行字符数→CL
4E08:2034 57	PUSH	DI	
4E08:2035 81C7B000	ADD	DI,00B0	
4E08:2039 F3	REPZ		: 在25行字符区清一行空白
4E08:203A AA	STOSB		
-u			
4E08:203B 5F	POP	DI	

4E08:203C 57	PUSH	DI	
4E08:203D 81C75010	ADD	DI,1050	
4E08:2041 50	PUSH	AX	
4E08:2042 32C0	XOR	AL,AL	
4E08:2044 8ACA	MOV	CL,DL	
4E08:2046 F3	REPZ		: 在25行标志区清一行标志
4E08:2047 AA	STOSB		
4E08:2048 5B	POP	AX	
4E08:2049 5F	POP	DI	
4E08:204A C3	RET		
4E08:204B 3AEE	CMP	CH,DH	: 滚动窗口起始行<滚动窗口结束行
4E08:204D 7201	JB	2050	
4E08:204F C3	RET		: 是, 什么也不做, 子程序返回
4E08:2050 1E	PUSH	DS	
4E08:2051 8ADB	MOV	BL,AL	
4E08:2053 8BC1	MOV	AX,CX	
4E08:2055 53	PUSH	BX	
4E08:2056 EB6403	CALL	23BD	: AX 返回窗口起始行列值扫描区的地址
4E08:2059 5B	POP	BX	
4E08:205A 8BF8	MOV	DI,AX	
-u			
4E08:205C 2BD1	SUB	DX,CX	
4E08:205E 81C20101	ADD	DX,0101	: DX + 0101H → DX
4E08:2062 50	PUSH	AX	
4E08:2063 B009	MOV	AL,09	
4E08:2065 F6E6	MUL	DH	
4E08:2067 8AF0	MOV	DH,AL	
4E08:2069 5B	POP	AX	
4E08:206A 803E490006	CMP	BYTE PTR [0049],06	: 640 × 200 图形方式?
4E08:206F 7304	JNB	2075	
4E08:2071 D0E2	SHL	DL,1	
4E08:2073 D1E7	SHL	DI,1	
4E08:2075 06	PUSH	ES	
4E08:2076 1F	POP	DS	
4E08:2077 2AED	SUB	CH,CH	
4E08:2079 B009	MOV	AL,09	
4E08:207B F6E3	MUL	BL	
-u			
4E08:207D 3C00	CMP	AL,00	
4E08:207F 742C	JZ	20AD	
4E08:2081 8ADB	MOV	BL,AL	
4E08:2083 B450	MOV	AH,50	
4E08:2085 F6E4	MUL	AH	
4E08:2087 8BF7	MOV	SI,DI	
4E08:2089 03F0	ADD	SI,AX	
4E08:208B 8AE6	MOV	AH,DH	
4E08:208D 2AE3	SUB	AH,BL	
4E08:208F EB1801	CALL	21AA	: 刷新区窗口向上滚动一行

4E08:2092 81EEB01F	SUB	SI,1FB0	: 调整滚动指针, 指向偶扫描区下一行
4E08:2096 81EFB01F	SUB	DI,1FB0	
4E08:209A FECC	DEC	AH	
4E08:209C 75F1	JNZ	20BF	
-u			
4E08:209E 8AC7	MOV	AL,BH	
4E08:20A0 E82001	CALL	21C3	: 刷新区滚动后空留区清一扫描行空白
4E08:20A3 81EFB01F	SUB	DI,1FB0	
4E08:20A7 FECB	DEC	BL	
4E08:20A9 75F5	JNZ	20A0	
4E08:20AB 1F	POP	DS	
4E08:20AC C3	RET		
4E08:20AD 8ADE	MOV	BL,DH	
4E08:20AF EBED	JMP	209E	
4E08:20B1 32FF	XOR	BH,BH	
4E08:20B3 06	PUSH	ES	
4E08:20B4 50	PUSH	AX	
4E08:20B5 53	PUSH	BX	
4E08:20B6 51	PUSH	CX	
4E08:20B7 52	PUSH	DX	
4E08:20B8 50	PUSH	AX	
4E08:20B9 53	PUSH	BX	
4E08:20BA 51	PUSH	CX	
4E08:20BB 52	PUSH	DX	
4E08:20BC BD0100	MOV	BP,0001	
-u			
4E08:20BF E8EBF9	CALL	1AAD	: 抹去原光标
4E08:20C2 5A	POP	DX	: 恢复滚动参数
4E08:20C3 59	POP	CX	
4E08:20C4 5B	POP	BX	
4E08:20C5 58	POP	AX	
4E08:20C6 FD	STD		
4E08:20C7 8ADB	MOV	BL,AL	
4E08:20C9 53	PUSH	BX	
4E08:20CA 8BC2	MOV	AX,DX	: 滚动窗口结束行列值→AX
4E08:20CC E811FF	CALL	1FE0	: 为窗口滚动设置参数
4E08:20CF 7441	JZ	2112	
4E08:20D1 2BF0	SUB	SI,AX	
4E08:20D3 8AE6	MOV	AH,DH	
4E08:20D5 2AE3	SUB	AH,BL	
4E08:20D7 7504	JNZ	20DD	
4E08:20D9 58	POP	AX	
4E08:20DA EB18	JMP	20F4	
4E08:20DC 90	NOP		
4E08:20DD E821FF	CALL	2001	
-u			
4E08:20E0 2BF5	SUB	SI,BP	
4E08:20E2 2BFD	SUB	DI,BP	

4E08:20E4 FECC	DEC	AH	
4E08:20E6 75F5	JNZ	20DD	
4E08:20E8 5B	POP	AX	
4E08:20E9 B000	MOV	AL,0	
4E08:20EB EB44FF	CALL	2032	: 在25行区滚动后空留区清一行空白
4E08:20EE 2BFD	SUB	DI, BP	
4E08:20F0 FECB	DEC	BL	
4E08:20F2 75F7	JNZ	20EB	
4E08:20F4 5A	POP	DX	
4E08:20F5 5F	POP	CX	
4E08:20F6 5B	POP	BX	
4E08:20F7 5B	POP	AX	
4E08:20F8 07	POP	ES	
4E08:20F9 3B2EA600	CMP	100A61, CH	
4E08:20FB 7231	JB	2130	
4E08:20FF 3A36A500	CMP	DH, [00A5]	: 滚动窗口结束行<汉字屏起始行
-u			
4E08:2103 722B	JB	2130	: 是, 整个滚动窗口在汉字屏之外
4E08:2105 8A26A500	MOV	AH, [00A5]	: 取汉字起始行
4E08:2109 3AE5	CMP	AH, CH	
4E08:210B 7209	JB	2116	
4E08:210D 32ED	XOR	CH, CH	
4E08:210F EB07	JMP	211B	
4E08:2111 90	NOP		
4E08:2112 8ADE	MOV	BL, DH	
4E08:2114 EBD2	JMP	20EB	
4E08:2116 2AEC	SUB	CH, BH	: 修改窗口起始行, 改为汉字屏内的行号
4E08:2118 8A26A600	MOV	AH, [00A6]	
4E08:211C 3AF4	CMP	DH, AH	
4E08:211E 7205	JB	2125	
4E08:2120 B609	MOV	DH, 09	
4E08:2122 EB03	JMP	2127	
-u			
4E08:2124 90	NOP		
4E08:2125 2AE6	SUB	AH, DH	: 汉字屏结束行一窗口结束行→AH
4E08:2127 3C09	CMP	AL, 09	
4E08:2129 7202	JB	212D	
4E08:212B B009	MOV	AL, 09	
4E08:212D EB0D00	CALL	213D	: 在刷新区实现窗口向下滚动
4E08:2130 C6065500FF	MOV	BYTE PTR [0055], FF	
4E08:2135 33ED	XOR	BP, BP	
4E08:2137 EB73F9	CALL	1AAD	: 设置光标
4E08:213A E902F9	JMP	1A3F	
4E08:213D 3AEE	CMP	CH, DH	
4E08:213F 7201	JB	2142	
4E08:2141 C3	RET		
4E08:2142 1E	PUSH	DS	
4E08:2143 FD	STD		

4E08:22A6 3C80	CMP	AL,80	: 扩展字符图形吗?
4E08:22A8 7306	JNB	22B0	
-u			
4E08:22AA BE9B27	MOV	SI,2798	: ASCII 码, SI 指向 ASCII 码字符字模库首址
4E08:22AD 0E	PUSH	CS	
4E08:22AE EB0B	JMP	22BE	
4E08:22B0 2C80	SUB	AL,80	
4E08:22B2 2BF6	SUB	SI,SI	
4E08:22B4 8EDE	MOV	DS,SI	
4E08:22B6 C5367C00	LDS	SI,[007C]	
4E08:22BA 1E	PUSH	DS	
4E08:22BB D1E0	SHL	AX,1	
4E08:22BD D1E0	SHL	AX,1	
4E08:22BF D1E0	SHL	AX,1	
4E08:22C1 03F0	ADD	SI,AX	: SI 指向入口字符码(AL) 的字符字模地址
4E08:22C3 1F	POP	DS	
4E08:22C4 07	POP	ES	: 附加段 ES 指向数据段
4E08:22C5 57	PUSH	DI	
4E08:22C6 51	PUSH	CX	
4E08:22C7 BF7800	MOV	DI,0078	: 字模加工区首址→DI
-u			
4E08:22CA B90300	MOV	CX,0003	: 3→CX
4E08:22CD 33C0	XOR	AX,AX	
4E08:22CF F6C370	TEST	BL,70	: 反向显示?
4E08:22D2 7402	JZ	22D6	
4E08:22D4 F7D0	NOT	AX	
4E08:22D6 F3	REPZ		
4E08:22D7 AB	STOSW		
4E08:22D8 B90400	MOV	CX,0004	: 字模库当前字模段取 2 字节字模→AX
4E08:22DB AD	LODSW		
4E08:22DC F6C370	TEST	BL,70	
4E08:22DF 7402	JZ	22E3	
4E08:22E1 F7D0	NOT	AX	: 反向显示, AX 中的字模按位取反
4E08:22E3 AB	STOSW		
4E08:22E4 E2F5	LOOP	22DB	
4E08:22E6 33C0	XOR	AX,AX	
4E08:22E8 F6C370	TEST	BL,70	
-u			
4E08:22EB 7402	JZ	22EF	
4E08:22ED F7D0	NOT	AX	
4E08:22EF AB	STOSW		
4E08:22F0 33C0	XOR	AX,AX	
4E08:22F2 AB	STOSW		: 再写 2 字节空白
4E08:22F3 59	POP	CX	
4E08:22F4 5F	POP	DI	
4E08:22F5 06	PUSH	ES	
4E08:22F6 1F	POP	DS	
4E08:22F7 07	POP	ES	

4E08:21A5 C3	RET		
4E08:21A6 BADE	MOV	BL,DH	
4E08:21A8 EBEC	JMP	2196	
4E08:21AA 8ACA	MOV	CL,DL	: 窗口内每行列数→CL
4E08:21AC 56	PUSH	SI	: 保存滚动指针
4E08:21AD 57	PUSH	DI	
4E08:21AE F3	REPZ		: 在偶扫描区行的窗口内滚动一行
4E08:21AF A4	MOVSB		
4E08:21B0 5F	POP	DI	: 恢复滚动指针
4E08:21B1 5E	POP	SI	
4E08:21B2 81C60020	ADD	SI,2000	: 滚动指针指向奇扫描行区
4E08:21B6 81C70020	ADD	DI,2000	
4E08:21BA 56	PUSH	SI	
4E08:21BB 57	PUSH	DI	
4E08:21BC 8ACA	MOV	CL,DL	
4E08:21BE F3	REPZ		: 在奇扫描行区的窗口内滚动一行
4E08:21BF A4	MOVSB		
4E08:21C0 5F	POP	DI	
4E08:21C1 5E	POP	SI	
4E08:21C2 C3	RET		
4E08:21C3 8ACA	MOV	CL,DL	: 窗口内每行列数→CL
-u			
4E08:21C5 57	PUSH	DI	: 保存清空白指针
4E08:21C6 F3	REPZ		: 在偶扫描行区窗口内清一行空白
4E08:21C7 AA	STOSB		
4E08:21C8 5F	POP	DI	
4E08:21C9 81C70020	ADD	DI,2000	
4E08:21CD 57	PUSH	DI	
4E08:21CE 8ACA	MOV	CL,DL	
4E08:21D0 F3	REPZ		
4E08:21D1 AA	STOSB		
4E08:21D2 5F	POP	DI	: 恢复清空白指针
4E08:21D3 C3	RET		
4E08:21D4 51	PUSH	CX	
4E08:21D5 8B365000	MOV	SI,[0050]	: 当前内光标位置→SI
4E08:21D9 890100	MOV	CX,0001	
4E08:21DC EB02	JMP	21E0	: 转21E 0, 清一字符位空白
4E08:21DE 90	NOP		
4E08:21DF 51	PUSH	CX	: 保存清空白计数
4E08:21E0 51	PUSH	CX	
4E08:21E1 BF7800	MOV	DI,0078	: 取字模加工区首址
4E08:21E4 06	PUSH	ES	
-u			
4E08:21E5 1E	PUSH	DS	
4E08:21E6 07	POP	ES	
4E08:21E7 32E4	XOR	AH,AH	
4E08:21E9 3C01	CMP	AL,01	: AL = 1?
4E08:21EB 7504	JNZ	21F1	

4E08:21ED B4FF	MOV	AH,FF	: AL = 1
4E08:21EF 32C0	XOR	AL,AL	
4E08:21F1 B91000	MOV	CX,0010	: 16→CX
4E08:21F4 F3	REPZ		: 在字模加工区左半区写16个字节的空白
4E08:21F5 AA	STOSB		
4E08:21F6 AB	STOSB		
4E08:21F7 07	POP	ES	
4E08:21F8 59	POP	CX	
4E08:21F9 8BFE	MOV	DI,S1	
4E08:21FB E6FA00	CALL	22FB	: 在刷新区光标位置开始处清CX 位置空白
4E08:21FE 59	POP	CX	
4E08:21FF C3	RET		
4E08:2200 50	PUSH	AX	: 保存要字的字符
4E08:2201 3C80	CMP	AL,60	
4E08:2203 7214	JB	2219	: ASCII 码, 转2219执行
-u			
4E08:2205 3CA0	CMP	AL,A0	: 汉 内码?
4E08:2207 7F10	JB	2219	
4E08:2209 8026AF00FE	AND	BYTE PTR [00AF],FE	
4E08:220E 8B3E5000	MOV	DI,[0050]	
4E08:2212 E8BD00	CALL	22A2	: 当前光标位置与扩展字符图形字模
4E08:2215 5B	POP	AX	
4E08:2216 E926F8	JMP	1A3F	
4E08:2219 3CFF	CMP	AL,FF	
4E08:221B 74EC	JZ	2209	: AL = FFH, 作扩展字符处理, 转2209
4E08:221D EBBF03	CALL	25DF	
4E08:2220 8BE8	MOV	BP,AX	: 字符、汉字处理标志→BP
4E08:2222 F6C407	TEST	AH,07	
-u			
4E08:2225 7504	JNZ	222B	
4E08:2227 5B	POP	AX	
4E08:2228 E914F8	JMP	1A3F	
4E08:222B F6C401	TEST	AH,01	: ASCII 码?
4E08:222E 7503	JNZ	2233	
4E08:2230 EB07	JMP	2239	: AH = 02, 2 字节汉字内码已全写入25行缓冲区
4E08:2232 90	NOP		
4E08:2233 E86C00	CALL	22A2	: 在当前内光标处写ASCII 码
4E08:2236 EB0A	JMP	2242	
4E08:2238 90	NOP		
4E08:2239 8BF2	MOV	SI,DX	
4E08:223B 8B3E7100	MOV	DI,[0071]	: 写汉字内光标→DI
4E08:223F E80900	CALL	224B	: 当前光标位置(DI) 写汉字字模
4E08:2242 33ED	XOR	BP,BP	
4E08:2244 E866F8	CALL	1AAD	
-u			
4E08:2247 5B	POP	AX	
4E08:2248 E9F4F7	JMP	1A3F	
4E08:224B 06	PUSH	ES	

4E08:224C 57	PUSH	DI	
4E08:224D 1E	PUSH	DS	: 字模加工区首址→DI
4E08:224E BF7800	MOV	DI,0078	
4E08:2251 07	POP	ES	
4E08:2252 06	PUSH	ES	
4E08:2253 8E0E	MOV	DS,SI	: 数据段指向要写的汉字字模段码
4E08:2255 33F6	XOR	SI,SI	
4E08:2257 B91000	MOV	CX,0010	
4E08:225A 32C0	XOR	AL,AL	: 0→AL
4E08:225C AD	LDSW		
4E08:225D F6C370	TEST	BL,70	
4E08:2260 7402	JZ	2264	
4E08:2262 F7D0	NOT	AX	
4E08:2264 AA	STOSB		
4E08:2265 26	ES:		
4E08:2266 886511	MOV	[DI+11],AH	
-u			
4E08:2269 E2F1	LOOP	225C	
4E08:226B 33C0	XOR	AX,AX	
4E08:226D AB	STOSW		: 左半区写2字节空白
4E08:226E 26	ES:		
4E08:226F 894510	MOV	[DI+10],AX	: 右半区写2字节空白
4E08:2272 1F	POP	DS	
4E08:2273 5F	POP	DI	
4E08:2274 07	POP	ES	
4E08:2275 BE7800	MOV	SI,0078	
4E08:2278 57	PUSH	DI	
4E08:2279 8BC7	MOV	AX,DI	
4E08:227B EB3F01	CALL	23BD	: AX返回入口光标(AX)地址位移
4E08:227E 88F8	MOV	DI,AX	: 刷新区光标地址→DI
4E08:2280 D90100	MOV	CX,0001	
4E08:2283 E88700	CALL	230D	: 当前光标位置写字模加工左及半区字模
4E08:2286 03C612	ADD	SI,+12	
-u			
4E08:2289 58	POP	AX	: 恢复写汉字内光标
4E08:228A FED0	INC	AL	
4E08:228C 3A064A00	CMP	AL,[004A]	: 光标超过屏幕的字符列数?
4E08:2290 7204	JB	2296	
4E08:2292 32C0	XOR	AL,AL	: 超过,修改光标值,指向下一行头
4E08:2294 FED4	INC	AH	
4E08:2296 EB2401	CALL	23BD	: AX返回内光标在刷新区的地址位移
4E08:2299 88F8	MOV	DI,AX	
4E08:229B D90100	MOV	CX,0001	
4E08:229E EB6C00	CALL	230D	: 在刷新区修改后的光标位置写字模加工区右半区字模
4E08:22A1 C3	RET		
4E08:22A2 32EA	XOR	AH,AH	: 0→AH,写ASCII码子程序入口
4E08:22A4 06	PUSH	ES	
4E08:22A5 1E	PUSH	DS	

4E08:22A6 3C80	CMP	AL,80	: 扩展字符图形吗?
4E08:22AB 7306	JNB	22B0	
-u			
4E08:22AA BE9B27	MOV	SI,2798	: ASCII 码、SI 指向 ASCII 码字符字模库首址
4E08:22AD 0E	PUSH	CS	
4E08:22AE EB0B	JMP	22BB	
4E08:22B0 2C80	SUB	AL,80	
4E08:22B2 2BF6	SUB	SI,SI	
4E08:22B4 8EDE	MOV	DS,SI	
4E08:22B6 C5367C00	LDS	SI,[007C]	
4E08:22BA 1E	PUSH	DS	
4E08:22BB D1E0	SHL	AX,1	
4E08:22BD D1E0	SHL	AX,1	
4E08:22BF D1E0	SHL	AX,1	
4E08:22C1 03F0	ADD	SI,AX	: SI 指向入口字符码(AL) 的字符字模地址
4E08:22C3 1F	POP	DS	
4E08:22C4 07	POP	ES	: 附加段ES 指向数据段
4E08:22C5 57	PUSH	DI	
4E08:22C6 51	PUSH	CX	
4E08:22C7 BF7800	MOV	DI,0078	: 字模加工区首址→DI
-u			
4E08:22CA B90300	MOV	CX,0003	: 3→CX
4E08:22CD 33C0	XOR	AX,AX	
4E08:22CF F6C370	TEST	BL,70	: 反向显示?
4E08:22D2 7402	JZ	22D6	
4E08:22D4 F7D0	NOT	AX	
4E08:22D6 F3	REPZ		
4E08:22D7 AB	STOSW		
4E08:22D8 B90400	MOV	CX,0004	
4E08:22DB AD	LODSW		: 字模库当前字模段取 2 字节字模→AX
4E08:22DC F6C370	TEST	BL,70	
4E08:22DF 7402	JZ	22E3	
4E08:22E1 F7D0	NOT	AX	: 反向显示, AX 中的字模按位取反
4E08:22E3 AB	STOSW		
4E08:22E4 E2F5	LDDP	22DB	
4E08:22E6 33C0	XOR	AX,AX	
4E08:22E8 F6C370	TEST	BL,70	
-u			
4E08:22EB 7402	JZ	22EF	
4E08:22ED F7D0	NOT	AX	
4E08:22EF AB	STOSW		
4E08:22F0 33C0	XOR	AX,AX	
4E08:22F2 AB	STOSW		: 再写 2 字节空白
4E08:22F3 59	POP	CX	
4E08:22F4 5F	POP	DI	
4E08:22F5 06	PUSH	ES	
4E08:22F6 1F	POP	DS	
4E08:22F7 07	POP	ES	

4E08:22F8 BE7800	MOV	SI,0078	: SI 指向字模加工区首址
4E08:22FB 8BC7	MOV	AX,DI	: 内光标位置→AX
4E08:22FD 80FC18	CMP	AH,18	
4E08:2300 7502	JNZ	2304	
4E08:2302 FEC7	INC	BH	
4E08:2304 E8B600	CALL	238D	: AX 符号入口光标在刷新区地址位移
4E08:2307 8BF8	MOV	DI,AX	: 刷新区光标地址→DI
4E08:2309 E80100	CALL	230D	: 字模加工右半区字模写入刷新区光标位置
-u			
4E08:230C C3	RET		
4E08:230D 52	PUSH	DX	
4E08:230E 50	PUSH	AX	
4E08:230F 53	PUSH	BX	
4E08:2310 803E490006	CMP	BYTE PTR [0049],06	: 哪种图形方式?
4E08:2315 722F	JB	2346	
4E08:2317 57	PUSH	DI	
4E08:2318 56	PUSH	SI	
4E08:2319 53	PUSH	BX	
4E08:231A B609	MOV	DH,09	
4E08:231C AC	LDSB		
4E08:231D F6C380	TEST	BL,80	
4E08:2320 7518	JNZ	233A	
4E08:2322 AA	STOSB		
4E08:2323 AC	LDSB		: 取字模加工区下字节一字模→AL
4E08:2324 26	ES:		
4E08:2325 88B5FF1F	MOV	[DI+1FFF],AL	: 刷新区奇扫描区写1字节字模
4E08:2329 83C74F	ADD	DI,+4F	: DI 指向下一个扫描
-u			
4E08:232C FECE	DEC	DH	
4E08:232E 75EC	JNZ	231C	
4E08:2330 5B	POP	BX	
4E08:2331 5E	POP	SI	
4E08:2332 5F	POP	DI	
4E08:2333 47	INC	DI	
4E08:2334 E2E1	LOOP	2317	: 将当前字模再写一次
4E08:2336 5B	POP	BX	
4E08:2337 58	POP	AX	
4E08:2338 5A	POP	DX	
4E08:2339 C3	RET		
4E08:233A 26	ES:		: 异或处理
4E08:233B 3205	XOR	AL,[DI]	
4E08:233D AA	STOSB		: 刷新区偶扫描区写1字节字模
4E08:233E AC	LDSB		: 取字模加工区下一字节字模→AL
4E08:233F 26	ES:		
4E08:2340 3285FF1F	XOR	AL,[DI+1FFF]	: 奇扫描区1字节字模与AL异或送回AL
4E08:2344 EBDE	JMP	2324	
4E08:2346 BAD3	MOV	DL,BL	: 字符属性→DL
4E08:2348 D1E7	SHL	DI,1	: 彩色图形方式,刷新区写字模首址×2
4E08:234A E83E00	CALL	238B	: 将BL低2位的颜色值扩展到整个BX

4E08:234D 5B	POP	AX	
4E08:234E 57	PUSH	DI	: 保存刷新区写字模指针
4E08:234F 56	PUSH	SI	: 保存字模加工区首址
4E08:2350 B609	MOV	DH,09	
4E08:2352 50	PUSH	AX	: AX 中的字模属性入栈
4E08:2353 AC	LODSB		
4E08:2354 E84B00	CALL	23A2	: 将AL 中的字模扩展成AX 的一个字的字模
4E08:2357 23C3	AND	AX,BX	: 字模值“与”上颜色值
4E08:2359 F6C280	TEST	DL,80	
4E08:235C 7403	JZ	2361	
4E08:235E 26	ES:		
4E08:235F 3305	XOR	AX,[DI]	
4E08:2361 26	ES:		
4E08:2362 8905	MOV	[DI],AX	: 字模的象素点写入刷新区偶扫描区当前位置
4E08:2364 AC	LODSB		
4E08:2365 E83A00	CALL	23A2	: AL 中的字模扩展成AX 中的一个字的字模
4E08:2368 23C3	AND	AX,BX	
4E08:236A F6C280	TEST	DL,80	
-u			
4E08:236D 7405	JZ	2374	
4E08:236F 26	ES:		
4E08:2370 33850020	XOR	AX,[DI+2000]	
4E08:2374 26	ES:		
4E08:2375 89850020	MOV	[DI+2000],AX	: 字模的象素点写入刷新区奇扫描区当前位置
4E08:2379 83C750	ADD	DI,+50	
4E08:237C FECE	DEC	DH	
4E08:237E 7503	JNZ	2353	
4E08:2380 5B	POP	AX	: 恢复 字符属性
4E08:2381 5E	POP	SI	: 恢复字符加工区首址
4E08:2382 5F	POP	DI	: 恢复刷新区写字模指针
4E08:2383 83C702	ADD	DI,+02	: 刷新区写字模指针指向下一个位置
4E08:2386 E2C6	LOOP	234E	: 当前字模再写一次
4E08:2388 50	PUSH	AX	
4E08:2389 50	JMP	2336	
4E08:238B 50	PUSH	AX	
4E08:238C 80E303	AND	BL,03	
-u			
4E08:238F 8AC3	MOV	AL,BL	
4E08:2391 51	PUSH	CX	
4E08:2392 B90300	MOV	CX,0003	
4E08:2395 D0E0	SHL	AL,1	
4E08:2397 D0E0	SHL	AL,1	
4E08:2399 0ADB	OR	BL,AL	
4E08:239B E2F8	LOOP	2395	
4E08:239D 8AFB	MOV	BH,BL	
4E08:239F 59	POP	CX	
4E08:23A0 5B	POP	AX	

4E08:23A1 C3	RET	
4E08:23A2 52	PUSH	DX ; 保存现场; AL 中与位象数点转换为 8 个彩色象数点
4E08:23A3 51	PUSH	CX
4E08:23A4 8AD0	MOV	DL,AL ; 字模→DL
4E08:23A6 8AF0	MOV	DH,AL ; 字模→DH
4E08:23AB 890800	MOV	CX,0008 ; 8→CX
4E08:23AB D0D2	RCL	DL,1
4E08:23AD D1D0	RCL	AX,1
-u		
4E08:23AF D0D6	RCL	DH,1
4E08:23B1 D1D0	RCL	AX,1
4E08:23B3 E2F6	LOOP	23AB ; 继续移位, 直列CX为0
4E08:23B5 59	POP	CX
4E08:23B6 5A	POP	DX
4E08:23B7 86E0	XCHG	AL,AH
4E08:23B9 C3	RET	
4E08:23BA A15000	MOV	AX,[0050] ; 取当前内光标位置
4E08:23BD 53	PUSH	BX
4E08:23BE 51	PUSH	CX
4E08:23BF 50	PUSH	AX
4E08:23C0 80FC18	CMP	AH,18 ; 光标在24行?
4E08:23C3 7502	JNZ	23C7
4E08:23C5 840A	MOV	AH,0A ; 是, 24行对应汉字屏提示行, 10→AH
4E08:23C7 B109	MOV	CL,09 ; 9→CL
4E08:23C9 32ED	XOR	CH,CH
4E08:23CB 8BD8	MOV	BX,AX ; 光标值→BX
4E08:23CD 8AC4	MOV	AL,AH ; 光标行值→AL
-u		
4E08:23CF F6264A00	MUL	BYTE PTR [004A]
4E08:23D3 52	PUSH	DX
4E08:23D4 F7E1	MUL	CX
4E08:23D6 5A	POP	DX
4E08:23D7 2AFF	SUB	BH,BH ; BH 中的光标行值清0, BL 为光标行值
4E08:23D9 03C3	ADD	AX,BX
4E08:23DB 59	POP	CX
4E08:23DC 80FD18	CMP	CH,18 ; 24行?
4E08:23DF 7203	JB	23E4
4E08:23E1 055000	ADD	AX,0050 ; 是, 地址再加80字节
4E08:23E4 59	POP	CX
4E08:23E5 5B	POP	BX
4E08:23E6 C3	RET	
4E08:23E7 A15000	MOV	AX,[0050]
4E08:23EA EB0DF9	CALL	1CFA ; AX 返回当前光标在25行区的地址参数
4E08:23ED D1F8	SAR	AX,1 ; 被2除, 得到地址位移
-u		
4E08:23EF 8BF0	MOV	SI,AX
4E08:23F1 8A84B000	MOV	AL,[SI+00B0] ; 当前外光标位置读字符→AL
4E08:23F5 8AA4B000	MOV	AH,[SI+00B0] ; 当前外光标位置读属性→AH

4E08:23F9 E943F6	JMP	1A3F	
4E08:23FC 800EAF00FE	OR	BYTE PTR [00AF],FE	: 14号功能块入口, TTY方式
4E08:2401 50	PUSH	AX	: 保存要写的字符
4E08:2402 50	PUSH	AX	: 再入栈保存
4E08:2403 B403	MOV	AH,03	
4E08:2405 CD10	INT	10	: 调用3号功能模块, DX返回当前外光标
4E08:2407 58	POP	AX	: 恢复要写的字符码
4E08:2408 3C08	CMP	AL,08	: 退格符?
4E08:240A 7503	JNZ	240F	
4E08:240C EB68	JMP	2476	
4E08:240E 90	NOP		
-u			
4E08:240F 3C0D	CMP	AL,0D	: 回车符?
4E08:2411 7503	JNZ	2416	
4E08:2413 EB6A	JMP	247F	
4E08:2415 90	NOP		
4E08:2416 3C0A	CMP	AL,0A	: 换行符?
4E08:2418 7503	JNZ	241D	
4E08:241A EB67	JMP	2483	
4E08:241C 90	NOP		
4E08:241D 3C07	CMP	AL,07	: 扬声器发声?
4E08:241F 7503	JNZ	2424	
4E08:2421 EB67	JMP	248A	: 扬声器发声, 转248 A
4E08:2423 90	NOP		
4E08:2424 BA3E6200	MOV	BH,[0062]	: 当前显示页→BH
4E08:2428 B40A	MOV	AH,0A	
4E08:242A B90100	MOV	CX,0001	
4E08:242D CD10	INT	10	: 在当前光标位置写字符
-u			
4E08:242F FEC2	INC	DL	: 光标列值+1
4E08:2431 3A164A00	CMP	DL,[004A]	: 光标列值
4E08:2435 7538	JNZ	2472	
4E08:2437 B200	MOV	DL,00	: 超出, 光标列值设置到行首
4E08:2439 B0FE17	CMP	DH,17	: 当前光标位置在23行?
4E08:243C 7232	JB	2470	
4E08:243E B402	MOV	AH,02	: 在, 功能号2→AH
4E08:2440 B700	MOV	BH,00	
4E08:2442 CD10	INT	10	: 调用2号功能块模, 设置光标
4E08:2444 A04900	MOV	AL,[0049]	
4E08:2447 3C04	CMP	AL,04	
4E08:2449 7206	JB	2451	
4E08:244B 3C07	CMP	AL,07	
4E08:244D B700	MOV	BH,00	
-u			
4E08:244F 7506	JNZ	2457	
4E08:2451 B408	MOV	AH,08	: 功能号8→AH
4E08:2453 CD10	INT	10	: 读字符及属性
4E08:2455 BAFC	MOV	BH,AH	: 属性→BH

4E08:2457 B80106	MOV	AX,0601	: 功能号 6 → AH, 滚动行数 1 → AL
4E08:245A B90000	MOV	CX,0000	
4E08:245D B617	MOV	DH,17	: 23 (17H) → PH
4E08:245F BA164A00	MOV	DL,[004A]	: 屏幕列数 → DL
4E08:2463 FECA	DEC	DL	
4E08:2465 CD10	INT	10	: 屏幕滚动一行
4E08:2467 B026AF0001	AND	BYTE PTR [00AF],01	
4E08:246C 5B	POP	AX	
4E08:246D E9CFF5	JMP	1A3F	
-u			
4E08:2470 FEC6	INC	DH	
4E08:2472 B402	MOV	AH,02	: 功能号 2 → AH
4E08:2474 EBEF	JMP	2465	: 转 2465 设置光标
4E08:2476 B0FA00	CMP	DL,00	: 光标在一行的行首
4E08:2479 74F7	JZ	2472	: 是, 不能再退, 转 2472 设置光标
4E08:247B FECA	DEC	DL	
4E08:247D EBF3	JMP	2472	
4E08:247F B200	MOV	DL,00	: 光标指向行首
4E08:2481 EBEF	JMP	2472	
4E08:2483 B0FE17	CMP	DH,17	: 当前行是 23 行?
4E08:2486 75E8	JNZ	2470	
4E08:2488 EBB4	JMP	243E	: 是, 转 243 E 屏幕向上滚动一行
4E08:248A B302	MOV	BL,02	: 2 → BL
4E08:248C E80200	CALL	2491	: 扬声器发声
4E08:248F EBD6	JMP	2467	
-u			
4E08:2491 B0B6	MOV	AL,B6	: 控制字 → AL
4E08:2493 E643	OUT	43,AL	: 写入定时器控制口
4E08:2495 B83305	MOV	AX,0533	: 确定频率
4E08:2498 E642	OUT	42,AL	
4E08:249A BAC4	MOV	AL,AH	
4E08:249C E642	OUT	42,AL	
4E08:249E E461	IN	AL,61	
4E08:24A0 BAE0	MOV	AH,AL	
4E08:24A2 0C03	OR	AL,03	
4E08:24A4 E661	OUT	61,AL	
4E08:24A6 2BC9	SUB	CX,CX	
4E08:24A8 E2FE	LOOP	24AB	
4E08:24AA FECD	DEC	BL	
4E08:24AC 75FA	JNZ	24AB	
4E08:24AE BAC4	MOV	AL,AH	
4E08:24B0 E661	OUT	61,AL	
-u			
4E08:24B2 C3	RET		
4E08:24B3 33C0	XOR	AX,AX	
4E08:24B5 E987F5	JMP	1A3F	
4E08:24B8 3C00	CMP	AL,00	: 清提示行?
4E08:24BA 740B	JZ	24C7	: 清提示行, 转 24C 7

4E08:24BC 3C01	CMP	AL,01	
4E08:24BE 7438	JZ	24F8	
4E08:24C0 3C02	CMP	AL,02	
4E08:24C2 7425	JZ	24E9	
4E08:24C4 E838	JMP	24FE	: AL = 3, 转24FE,TTY 方式写提示行
4E08:24C6 90	NOP		
4E08:24C7 8B0E4A00	MOV	CX,[004A]	: 16是功能: 提示行管理
4E08:24CB 32C0	XOR	AL,AL	
4E08:24CD B709	MOV	BH,09	
4E08:24CF B301	MOV	BL,01	
4E08:24D1 BE0018	MOV	SI,1800	
-u			
4E08:24D4 E808FD	CALL	21DF	: 清提示行
4E08:24D7 32D2	XOR	DL,DL	
4E08:24D9 E87800	CALL	2554	: 提示行设置光标
4E08:24DC BF203C	MOV	DI,3C20	
4E08:24DF B0FF	MOV	AL,FF	: FFH → AL
4E08:24E1 895000	MOV	CX,0050	: 80 → CX
4E08:24E4 F3	REPZ		: 在提示行之上写一条分隔线
4E08:24E5 AA	STOSB		
4E08:24E6 E956F5	JMP	1A3F	
4E08:24E9 52	PUSH	DX	: 提示行光标定位
4E08:24EA 8B165600	MOV	DX,[0056]	: 取提示行原光标
4E08:24EE E84E00	CALL	253F	: 提示行抹去原光标
4E08:24F1 5A	POP	DX	
4E08:24F2 E85F00	CALL	2554	: 提示行设置新光标
-u			
4E08:24F5 E947F5	JMP	1A3F	
4E08:24F8 E86A00	CALL	2565	: 在当前光标处写ASCII 码, 汉字
4E08:24FB E941F5	JMP	1A3F	
4E08:24FE 80FA08	CMP	DL,08	: 提示行TTY 方式显示符退格?
4E08:2501 741C	JZ	251F	
4E08:2503 80FA07	CMP	DL,07	: 发声?
4E08:2506 742F	JZ	2537	
4E08:2508 E85A00	CALL	2565	: 提示行当前光标位置写字符
4E08:250B 8B165600	MOV	DX,[0056]	
4E08:250F FEC2	INC	DL	
4E08:2511 3A164A00	CMP	DL,[004A]	: 光标超出范围?
-u			
4E08:2515 7502	JNZ	2519	
4E08:2517 FECA	DEC	DL	
4E08:2519 E83800	CALL	2554	: 提示行设置新光标
4E08:251C E920F5	JMP	1A3F	
4E08:251F 8B165600	MOV	DX,[0056]	: 提示行光标 → DX
4E08:2523 E81900	CALL	253F	: 抹去原光标
4E08:2526 8B165600	MOV	DX,[0056]	
4E08:252A 80FA00	CMP	DL,00	

4E08:252D	7402	JZ	2531	
4E08:252F	FECA	DEC	DL	
4E08:2531	E82000	CALL	2554	: 提示行设置新光标
4E08:2534	E908F5	JMP	1A3F	
-u				
4E08:2537	B302	MOV	BL,02	
4E08:2539	E855FF	CALL	2491	: 扬声器发声
4E08:253C	E900F5	JMP	1A3F	
4E08:253F	32C0	XOR	AL,AL	
4E08:2541	B301	MOV	BL,01	
4E08:2543	B618	MOV	DH,18	
4E08:2545	B9165600	MOV	[0056],DX	: 提示行光标单元
4E08:2549	B90100	MOV	CX,0001	
4E08:254C	B709	MOV	BH,09	
4E08:254E	8BF2	MOV	SI,DX	
4E08:2550	E8BCFC	CALL	21DF	: 刷新区提示行清当前位置(SI)
4E08:2553	C3	RET		
4E08:2554	E8E8FF	CALL	253F	: 清提示行当前位置(DL)
-u				
4E08:2557	B25F	MOV	DL,5F	
4E08:2559	A07300	MOV	AL,[0073]	
4E08:255C	50	PUSH	AX	
4E08:255D	E80500	CALL	2565	: 在当前位置写光标(字符“-”)
4E08:2560	58	POP	AX	
4E08:2561	A27300	MOV	[0073],AL	
4E08:2564	C3	RET		
4E08:2565	52	PUSH	DX	: 提示行字符显示块入口
4E08:2566	BB165600	MOV	DX,[0056]	: 提示行光标→DX
4E08:256A	E8D2FF	CALL	253F	: 清提示行当前光标位置
4E08:256D	5A	POP	DX	
4E08:256E	BAC2	MOV	AL,DL	
4E08:2570	BB0109	MOV	BX,0901	
4E08:2573	B90100	MOV	CX,0001	
4E08:2576	BB3E5600	MOV	DI,[0056]	
-u				
4E08:257A	A880	TEST	AL,80	: 汉字内码?
4E08:257C	7509	JNZ	2587	
4E08:257E	E821FD	CALL	22A2	: 在当前光标位置写字符(AL)
4E08:2581	80267300F0	AND	BYTE PTR [0073],F0	: 不是, 清提示行汉字内码标志
4E08:2586	C3	RET		
4E08:2587	F606730001	TEST	BYTE PTR [0073],01	: 前一个位置的码是第一字节汉字内码
4E08:258C	7508	JNZ	2596	: 是, 当前是第二字节内码, 转2596
4E08:258E	FE067300	INC	BYTE PTR [0073]	
4E08:2592	A2A100	MOV	[00A1],AL	
4E08:2595	C3	RET		
4E08:2596	BA36A100	MOV	DH,[00A1]	: 取第一字节内码→DH, 第二字节内码在DL中
-u				
4E08:259A	80267300F0	AND	BYTE PTR [0073],F0	

4E08:259F 81E27F7F	AND	DX,7F7F	: 汉字内码转换成国标码
4E08:25A3 E8C601	CALL	276C	: DX返回当前汉字字模起始段码
4E08:25A6 8BF2	MOV	SI,DX	
4E08:25A8 4F	DEC	DI	
4E08:25A9 E89FFC	CALL	2248	: 提示行DI 处写汉字字模
4E08:25AC C3	RET		
4E08:25AD E8BC01	CALL	276C	: DX 返回指定汉字内码对应的字模起始段码
4E08:25B0 8EC2	MOV	ES,DX	
4E08:25B2 8EDD	MOV	DS,BP	
4E08:25B4 8BF3	MOV	SI,BX	
4E08:25B6 33FF	XOR	DI,DI	
4E08:25BB B91000	MOV	CX,0010	
-u			
4E08:25BB F3	REPZ		: 把字模数据写入到字模库当前汉字内码对应的位置
4E08:25BC A5	MOVSW		
4E08:25BD E97FF4	JMP	1A3F	
4E08:25C0 E8A901	CALL	276C	: DX 返回指定汉字内码对应的字模起始段码
4E08:25C3 8EDA	MOV	DS,DX	
4E08:25C5 8EC5	MOV	ES,BP	: BP:BX 汉字内码字模数写入地址
4E08:25C7 8BFB	MOV	DI,BX	
4E08:25C9 33F6	XOR	SI,SI	
4E08:25CB B91000	MOV	CX,0010	
4E08:25CE F3	REPZ		
4E08:25CF A5	MOVSW		: 重复写入指定地址
4E08:25D0 E96CF4	JMP	1A3F	
4E08:25D3 A25400	MOV	[0054],AL	
4E08:25D6 BD0100	MOV	BP,0001	
4E08:25D9 E8D1F4	CALL	1AAD	
-u			
4E08:25DC E960F4	JMP	1A3F	
4E08:25DF 33D2	XOR	DX,DX	
4E08:25E1 51	PUSH	CX	
4E08:25E2 53	PUSH	BX	
4E08:25E3 8B2E5000	MOV	BP,[0050]	: 当前内光标位置→BP
4E08:25E7 A880	TEST	AL,80	: 要写的码是汉字内码?
4E08:25E9 751B	JNZ	2606	
4E08:25EB 50	PUSH	AX	
4E08:25EC 8BC5	MOV	AX,BP	
4E08:25EE E86901	CALL	275A	: DI 返回当前光标在25行区地址位移
4E08:25F1 5B	POP	AX	
4E08:25F2 5B	POP	BX	
4E08:25F3 59	POP	CX	
4E08:25F4 8B85B000	MOV	[DI+00B0],AL	
4E08:25F8 8B9DB008	MOV	[DI+08B0],BL	
-u			
4E08:25FC C685501000	MOV	BYTE PTR [DI+1050],00	
4E08:2601 B8FD	MOV	DI,BP	: 当前光标位置→DI
4E08:2603 B401	MOV	AH,01	

4E08:2605 C3	RET		
4E08:2606 50	PUSH	AX	: 保存要写的汉字的汉字内码
4E08:2607 E8DC00	CALL	26E6	: DL 返回当前光标位置参数
4E08:260A 58	POP	AX	
4E08:260B 0AD2	OR	DL,DL	
4E08:260D 7418	JZ	2627	
4E08:260F FECA	DEC	DL	
4E08:2611 7468	JZ	267E	
4E08:2613 E8AF00	CALL	26C5	: 测试前一个码的状态标志
4E08:2616 750A	JNZ	2622	
4E08:2618 8681	MOV	DH,BI	
4E08:261A 8BC8	MOV	CX,AX	
-u			
4E08:261C 5B	POP	BX	
4E08:261D E8DD00	CALL	26FD	: 内码写25行区, AH 返回写刷新区标志
4E08:2620 59	POP	CX	
4E08:2621 C3	RET		
4E08:2622 E8AC00	CALL	26D1	: 判断下一个位置码的标志, DH 返回写25行区标志
4E08:2625 EBF3	JMP	261A	
4E08:2627 803E510000	CMP	BYTE PTR [0051],00	
4E08:262C 7507	JNZ	2635	
4E08:262E 803EA50000	CMP	BYTE PTR [00A5],00	
4E08:2633 74E5	JZ	261A	
4E08:2635 50	PUSH	AX	
4E08:2636 8BC5	MOV	AX,BP	
4E08:2638 FECC	DEC	AH	
4E08:263A A04A00	MOV	AL,[004A]	
-u			
4E08:263D FECB	DEC	AL	
4E08:263F E82001	CALL	2762	: 测试上一行行尾码的状态标志
4E08:2642 7403	JZ	2647	
4E08:2644 58	POP	AX	
4E08:2645 EBD3	JMP	261A	
4E08:2647 8BC5	MOV	AX,BP	
4E08:2649 FECC	DEC	AH	
4E08:264B A04A00	MOV	AL,[004A]	
4E08:264E FECB	DEC	AL	
4E08:2650 A37100	MOV	[0071],AX	
4E08:2653 E80401	CALL	275A	: AX 返回上一行行尾行列值在25行区地址位移
4E08:2656 59	POP	CX	
4E08:2657 5B	POP	BX	
4E08:2658 88DB100	MOV	[DI+00B1],CL	
4E08:265C C78550100102	MOV	WORD PTR [DI+1050],0201	
-u			
4E08:2662 889D8108	MOV	[DI+0881],BL	
4E08:2666 8AD1	MOV	DL,CL	
4E08:2668 8AB5B000	MOV	DH,[DI+00B0]	: 上一行行尾汉字内码→DH
4E08:266C 8BC5	MOV	AX,BF	

4E08:266E B0FC00	CMP	AH,00	
4E08:2671 7407	JZ	267A	
4E08:2673 EBF600	CALL	276C	: DX 返回汉字内码对应的字模起始段码
4E08:2676 B402	MOV	AH,02	
4E08:2678 59	POP	CX	
4E08:2679 C3	RET		
4E08:267A 32E4	XOR	AH,AH	
4E08:267C 59	POP	CX	
4E08:267D C3	RET		
4E08:267E 50	PUSH	AX	
4E08:267F E84300	CALL	26C5	: 测试当前光标的前一个位置码标志
-u			
4E08:2682 7508	JNZ	268F	
4E08:2684 8BC5	MOV	AX,BP	
4E08:2686 B681	MOV	DH,B1	
4E08:2688 59	POP	CX	
4E08:2689 5B	POP	BX	
4E08:268A E87000	CALL	26FD	: 当前光标位置写25行区, AH 返回写刷新区标志
4E08:268D 59	POP	CX	
4E08:268E C3	RET		
4E08:268F 8BC5	MOV	AX,BP	
4E08:2691 80FC09	CMP	AH,09	
4E08:2694 7510	JNZ	26A6	
4E08:2696 803EA60017	CMP	BYTE PTR [00A6],17	
4E08:269B 7209	JB	26A6	
4E08:269D 59	POP	CX	
4E08:269E 5B	POP	BX	
4E08:269F 32F6	XOR	DH,DH	
4E08:26A1 E85900	CALL	26FD	: AH 返回写刷新区标志
-u			
4E08:26A4 59	POP	CX	
4E08:26A5 C3	RET		
4E08:26A6 8BC5	MOV	AX,BP	
4E08:26A8 32C0	XOR	AL,AL	
4E08:26AA FEC4	INC	AH	
4E08:26AC E8B300	CALL	2762	: 测试下一行行首码的状态标志
4E08:26AF 7402	JZ	26B3	
4E08:26B1 EBEA	JMP	269D	
4E08:26B3 59	POP	CX	
4E08:26B4 5B	POP	BX	
4E08:26B5 B682	MOV	DH,B2	
4E08:26B7 E84300	CALL	26FD	: AH 返回写刷新区标志 (02), DX 返回汉字字模起始段码
4E08:26BA 8BCD	MOV	CX,BP	
4E08:26BC 80FD09	CMP	CH,09	
4E08:26BF 7502	JNZ	26C3	
4E08:26C1 32E4	XOR	AH,AH	
4E08:26C3 59	POP	CX	
-u			

4E08:26C4 C3	RET		
4E08:26C5 53	PUSH	BX	
4E08:26C6 50	PUSH	AX	
4E08:26C7 8BC5	MOV	AX,BP	
4E08:26C9 FEC8	DEC	AL	
4E08:26CB E89400	CALL	2762	; CF 标志返回前一个码的状态情况
4E08:26CE 58	POP	AX	
4E08:26CF 58	POP	BX	
4E08:26D0 C3	RET		
4E08:26D1 53	PUSH	BX	
4E08:26D2 50	PUSH	AX	
4E08:26D3 8BC5	MOV	AX,BP	
4E08:26D5 FEC0	INC	AL	
4E08:26D7 E88800	CALL	2762	; CF 返回下一个位置码状态情况
4E08:26DA 7505	JNZ	26E1	
4E08:26DC B6B2	MOV	DH,B2	
4E08:26DE EB03	JMP	26E3	
4E08:26E0 90	NOP		
4E08:26E1 32F6	XOR	DH,DH	
4E08:26E3 58	POP	AX	
-u			
4E08:26E4 58	POP	BX	
4E08:26E5 C3	RET		
4E08:26E6 8BC5	MOV	AX,BP	
4E08:26E8 3C00	CMP	AL,00	
4E08:26EA 740E	JZ	26FA	
4E08:26EC B201	MOV	DL,01	
4E08:26EE 8A364A00	MOV	DH,[004A]	
4E08:26F2 FECE	DEC	DH	
4E08:26F4 3AC6	CMP	AL,DH	
4E08:26F6 7402	JZ	26FA	
4E08:26F8 FEC2	INC	DL	
4E08:26FA 32F6	XOR	DH,DH	
4E08:26FC C3	RET		
4E08:26FD 8BC5	MOV	AX,BP	
4E08:26FF F6C680	TEST	DH,B0	
4E08:2702 7513	JNZ	2717	
-u			
4E08:2704 EB5300	CALL	275A	
4E08:2707 88DB000	MOV	[DI+00B0],CL	
4E08:2708 889D8008	MOV	[DI+08B0],BL	
4E08:270F C685501080	MOV	BYTE PTR [DI+1050],B0	
4E08:2714 32E4	XOR	AH,AH	
4E08:2716 C3	RET		
4E08:2717 F6C601	TEST	DH,01	
4E08:271A 7520	JNZ	273C	
4E08:271C A37100	MOV	[0071],AX	
4E08:271F E83800	CALL	275A	; DI 返回当前光标位置在25行区的地址位移
4E08:2722 888DB000	MOV	[DI+00B0],CL	


```

-u
4E08:2726 C78550100102 MOV WORD PTR [DI+1050],0201
4E08:272C 889D8008 MOV [DI+0880],BL
4E08:2730 BAF1 MOV DH,CL
4E08:2732 8A95B100 MOV DL,[DI+00B1]
4E08:2736 EB3300 CALL 276C ; DX 返回汉字内码对应的字模起始段码
4E08:2739 B402 MOV AH,02
4E08:273B C3 RET
4E08:273C FEC8 DEC AL
4E08:273E A37100 MOV [0071],AX
4E08:2741 EB1600 CALL 275A ; DI 返回前一个位置在25行区的地址位移
4E08:2744 888DB100 MOV [DI+00B1],CL

-u
4E08:2748 C78550100102 MOV WORD PTR [DI+1050],0201
4E08:274E 889D8108 MOV [DI+08B1],BL
4E08:2752 BAD1 MOV DL,CL
4E08:2754 8AB5B000 MOV DH,[DI+00B0]
4E08:2758 EBDC JMP 2736
4E08:275A EB9DF5 CALL 1CFA ; AX 返回入口行列值25行区的地址参数
4E08:275D D1FB SAR AX,1 ; 被2除,得到地址位移
4E08:275F 8BF8 MOV DI,AX
4E08:2761 C3 RET
4E08:2762 EB5FF5 CALL 275A ; DI 返回入口行列值在25行区的地址位移
4E08:2765 8AB55010 MOV AL,[DI+1050]

-u
4E08:2769 3C80 CMP AL,80 ; 未完成状态,设置CF标志
4E08:276B C3 RET
4E08:276C B1E27F7F AND DX,7F7F ; 汉字内码转换成国标码
4E08:2770 50 PUSH AX
4E08:2771 80FE29 CMP DH,29 ; 09区的内码?
4E08:2774 7502 JNZ 277B
4E08:2776 B626 MOV DH,26 ; 09区,09区转换成06区
4E08:2778 80FE30 CMP DH,30
4E08:277B 7203 JB 2780
4E08:277D 80EE08 SUB DH,08
4E08:2780 80EE21 SUB DH,21 ; 国标码区号转换成区位码区号
4E08:2783 B05E MOV AL,5E
4E08:2785 F6E6 MUL DH ; 94×AL→AL
4E08:2787 80EA21 SUB DL,21

-u
4E08:278A 32F6 XOR DH,DH
4E08:278C 03C2 ADD AX,DX ; 区的起始地址参数+位的起始地址参数→AX
4E08:278E D1E0 SHL AX,1
4E08:2790 03067500 ADD AX,[0075] ; 加上汉字字模库起始段码
4E08:2794 8BD0 MOV DX,AX
4E08:2796 5B POP AX
4E08:2797 C3 RET
4E08:2798 0000 ADD [BX+SI],AL

```

```

4E08:279A 0000      ADD     [BX+SI],AL
4E08:279C 0000      ADD     [BX+SI],AL
4E08:279E 0000      ADD     [BX+SI],AL
4E08:27A0 7EB1      JLE     2723
4E08:27A2 A5        MOVSW
4E08:27A3 81BD99817E7E  CMP     WORD PTR [DI+8199],7E7E
4E08:27A9 FFDB      CALL    FAR BX

```

-d cs:2798 ASC I 字符字模库 (8 × 8) 共128 个字模

```

4E08:2798 00 00 00 00 00 00 00 00      .....
4E08:27A0 7E 81 A5 B1 BD 99 81 7E-7E FF DB FF C3 E7 FF 7E  ^X.=..^.L.Cg.^
4E08:27B0 6C FE FE FE 7C 38 10 00-10 38 7C FE 7C 38 10 00  1***!B...8!*!B..
4E08:27C0 38 7C 38 FF FE 7C 38 7C-10 10 38 7C FE 7C 38 7C  8!B**!B!..8!*!B!
4E08:27D0 00 00 18 3C 3C 18 00 00-FF FF E7 C3 C3 E7 FF FF    ...<<.....gCCg..
4E08:27E0 00 3C 66 42 42 66 3C 00-FF C3 99 BD BD 99 C3 FF    .<fBBf<..C.=.C.
4E08:27F0 0F 07 0F 7D CC CC CC 78-3C 66 66 66 3C 18 7E 18  ...}LLLx<fff<.^
4E08:2800 3F 33 3F 30 30 70 F0 E0-7F 63 7F 63 63 67 E6 C0  23?00pp`.c.ccqf@
4E08:2810 99 5A 3C E7 E7 3C 5A 99      .Zkqg<Z.

```

-d

```

4E08:2818 80 E0 F8 FE FB E0 80 00      .'x*x'..
4E08:2820 02 0E 3E FE 3E 0E 02 00-18 3C 7E 18 18 7E 3C 18  ..>*)....<^.*<.
4E08:2830 66 66 66 66 66 00 66 00-7F DB DB 7B 1B 1B 1B 00  fffff.f...[[<....
4E08:2840 3E 63 38 6C 6C 38 CC 78-00 00 00 00 7E 7E 7E 00  >c8!1BLx....***.
4E08:2850 18 3C 7E 18 7E 3C 18 FF-18 3C 7E 18 18 18 18 00  .<^.*<...<^.....
4E08:2860 18 18 18 18 7E 3C 18 00-00 18 0C FE 0C 18 00 00  ....*<.....^.....
4E08:2870 00 30 60 FE 60 30 00 00-00 00 C0 C0 C0 FE 00 00  .0`^0....@e@^..
4E08:2880 00 24 66 FF 66 24 00 00-00 18 3C 7E FF FF 00 00  .$.f$....<^....
4E08:2890 00 FF FF 7E 3C 18 00 00      ...*<...

```

-d

```

4E08:2898 00 00 00 00 00 00 00 00      .....
4E08:28A0 30 78 78 30 30 00 30 00-60 60 00 00 00 00 00  0xx00.0.111.....
4E08:28B0 6C 6C FE 6C FE 6C 6C 00-30 7C 00 78 00 F8 30 00  11*1*11.0!ex.x0.
4E08:28C0 00 C6 CC 18 30 66 C6 00-38 6C 38 78 5C CC 76 30  .fL.0ff.81Bv\Lv.
4E08:28D0 60 60 C0 00 00 00 00 00-18 3C 60 60 60 30 18 00  ``e.....0`^0..
4E08:28E0 60 30 18 18 18 30 60 00-00 66 3C FF 3C 66 00 00  `0...0`.f<.<f..
4E08:28F0 00 30 30 FC 30 30 00 00-00 00 00 00 00 30 30 60  .00!00.....00`
4E08:2900 00 00 00 FC 00 00 00 00-00 00 00 00 00 30 30 00  ...!.....00.
4E08:2910 06 0C 18 30 60 C0 80 00      ...0`e..

```

-d

```

4E08:2918 7C C6 CE DE F6 E6 7C 00      !FN^vf!.
4E08:2920 30 70 30 30 30 30 FC 00-78 CC 0C 38 60 CC FC 00  0p0000!.xL.B'Li.
4E08:2930 78 CC 0C 38 0C CC 78 00-1C 3C 6C CC FE 0C 1E 00  xL.B.Lx.<!L^...
4E08:2940 FC C0 FB 0C 0C CC 78 00-38 60 C0 FB CC CC 78 00  !ex..Lx.B'exLLx.
4E08:2950 FC CC 0C 18 30 30 30 00-78 CC CC 78 CC CC 78 00  !L..000.xLLxLLx.
4E08:2960 78 CC CC 7C 0C 18 70 00-00 30 30 00 00 30 30 00  xLL!..p..00..00.
4E08:2970 00 30 30 00 00 30 30 60-18 30 60 C0 60 30 18 00  .00..00'.0'@'0..
4E08:2980 00 00 FC 00 00 FC 00 00-60 30 18 0C 18 30 60 00  ...!...!..'0...0'.
4E08:2990 78 CC 0C 18 30 00 30 00      xL..0.0.

```

-d

每符号 8 字节
每行二个符号

```

4E08:2978 7C C6 DE DE DE C0 78 00          IF^@ex.
4E08:29A0 30 78 CC CC FC CC CC 00-FC 66 66 7C 66 66 FC 00 0xLLILL.Lfffiff.
4E08:29B0 3C 66 C0 C0 C0 66 3C 00-FB 6C 66 66 66 6C F8 00 <f@ef(xlfffflx.
4E08:29C0 FE 62 68 78 68 62 FE 00-FE 62 68 78 68 60 F0 00 ~bhxhb^."bhxh`p.
4E08:29D0 3C 66 C0 C0 CE 66 3E 00-CC CC CC FC CC CC CC 00 <f@NF>.LLLLLLL.
4E08:29E0 78 30 30 30 30 30 78 00-1E 0C 0C 0C CC CC 78 00 x00000x.....LLx.
4E08:29F0 E6 66 6C 78 6C 66 E6 00-F0 60 60 60 62 66 FE 00 fflxiff.p`bf^..
4E08:2A00 C6 EE FE FE D6 C6 C6 00-C6 E6 F6 DE CE C6 C6 00 Fh^VFF.Ffv^NFF.
4E08:2A10 38 6C C6 C6 C6 6C 38 00          BIFFF1B.

-d
4E08:2A18 FC 66 66 7C 60 60 F0 00          iffff`p.
4E08:2A20 78 CC CC CC DC 78 1C 00-FC 66 66 7C 6C 66 E6 00 xLLLx.....ffiff.
4E08:2A30 78 CC E0 70 1C CC 78 00-FC B4 30 30 30 30 78 00 xL`p.Lx.140000x.
4E08:2A40 CC CC CC CC CC CC FC 00-CC CC CC CC CC 78 30 00 LLLLLL.LLLLLx0.
4E08:2A50 C6 C6 C6 D6 FE EE C6 00-C6 C6 6C 38 38 6C C6 00 FFFV^nF.FF16B1F.
4E08:2A60 CC CC CC 78 30 30 78 00-FE C6 8C 18 32 66 FE 00 LLLx00x.^F..2f^..
4E08:2A70 78 60 60 60 60 60 78 00-C0 60 30 18 0C 06 02 00 x`^x.@`0.....
4E08:2A80 76 18 18 18 18 18 78 00-10 38 6C C6 00 00 00 00 x.....x..B1F....
4E08:2A90 00 00 00 00 00 00 00 FF          .....

-d
4E08:2A98 30 30 18 00 00 00 00 00          00.....
4E08:2AA0 00 00 78 0C 7C CC 76 00-E0 60 60 7C 66 66 DC 00 ..x.lLv.`fflx.
4E08:2AB0 00 00 78 CC C0 CC 78 00-1C 0C 0C 7C CC CC 76 00 ..xL@Lx.....llLv.
4E08:2AC0 00 00 78 CC FC C0 78 00-38 6C 60 F0 60 60 F0 00 ..xLl@x.B1`p`p.
4E08:2AD0 00 00 76 CC CC 7C 0C FB-E0 60 6C 76 66 66 E6 00 ..vLLl.x`lvfff.
4E08:2AE0 30 00 70 30 30 30 78 00-0C 00 0C 0C CC CC 78 0.p000x.....LLx
4E08:2AF0 E0 60 66 6C 78 6C E6 00-70 30 30 30 30 78 00 `flxiff.p00000x.
4E08:2B00 00 00 CC FE FE D6 C6 00-00 00 FB CC CC CC CC 00 ..L^VF...xLLLL.
4E08:2B10 00 00 78 CC CC CC 78 00          ..xLLLx.

-d
4E08:2B18 00 00 DC 66 66 7C 60 F0          ..\ffff`p
4E08:2B20 00 00 76 CC CC 7C 0C 1E-00 00 DC 76 66 60 F0 00 ..vLLl....\vf`p.
4E08:2B30 00 00 7C C0 78 0C F8 00-10 30 7C 30 30 34 18 00 ..fex.x..01004..
4E08:2B40 00 00 CC CC CC CC 76 00-00 00 CC CC CC 78 30 00 ..LLLLv...LLLx0.
4E08:2B50 00 00 C6 D6 FE FE 6C 00-00 00 C6 6C 38 6C C6 00 ..FV^*1...F1B1F.
4E08:2B60 00 00 CC CC CC 7C 0C FB-00.00 FC 98 30 64 FC 00 ..LLLl.x..f.0d!.
4E08:2B70 1C 30 30 E0 30 30 1C 00-18 18 18 00 18 18 18 00 ..00`00.....
4E08:2B80 E0 30 30 1C 30 30 E0 00-76 DC 00 00 00 00 00 00 `00.00`.v\.....
4E08:2B90 00 10 38 6C C6 C6 FE 00          ..B1FF^..

-d622:00
0622:0000 00 00 00 00 00 00 00 00-00 00 00 00 07 F0 0C C0 .....p.@
0622:0010 0C C0 07 80 0C 00 0F 80-1C E0 18 60 0C C0 07 80 .e.....`.e..
0622:0020 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
0622:0030 00 00 00 00 30 00 1E 00-0F 00 07 00 00 00 00 .....0.....
0622:0040 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
0622:0050 00 00 00 00 3C 00 66 00-66 00 66 00 3C 00 00 00 ....<.f.f.f.<...
0622:0060 00 00 00 00 00 00 00 00-00 00 00 00 01 80 03 C0 .....e
0622:0070 03 C0 01 80 00 00 00 00-00 00 00 00 00 00 00 00 .e.....

```

第七章 CC-BIOS的打印驱动模块

7.1 总述

CC BIOS中的打印驱动模块,是对ROM-BIOS的打印驱动模块进行扩充而成的。主要是进行了一些汉字适配工作,使其具有输出汉字的功能。汉字的打印输出是在图形方式下进行的,而ROM BIOS的打印驱动模块并不支持这种方式。因此,CC-BIOS的打印驱动模块,在ROM BIOS的打印驱动模块的基础上增加了图形方式的处理功能。另外,在CC-BIOS的其他模块中,有些模块与本模块间存在通讯关系,这些模块在工作中,通过通讯手段,向本模块发送一些控制命令,故本模块还要对这些发来的控制命令进行相应的解释处理。

实现汉字打印输出的过程,实际上是把汉字机内码转换成对应的打印数据,并把这些打印数据送打印机输出的过程。打印数据的形成要借助于汉字字库。CC-BIOS的汉字字库中的字模采用的是横向点阵,它适用于在屏幕上显示汉字。而打印机需要的是纵向点阵(即打印数据)。所以本模块还要把字库的横向点阵转换成打印数据。然后再向打印机输出。

在IBM PC系统中,用户调用打印机有两种方式,一种是调用打印机打印文件或输出结果;另一种是调用打印机打印当前屏幕上显示的内容。前者是通过调用17类中断程序来实现的,后者是通过调用5类中断程序来实现的。故CC-BIOS的打印驱动模块由打印机驱动程序(17类中断程序)和打印屏幕驱动程序(5类中断程序)组成。

打印机驱动程序与系统所配置的打印机有直接关系,一般有24针打印机和9针打印机。它们的驱动程序大体上是一致的,但在一些细节上(比如控制命令)会有些不同。因为IBM PC系统的原配打印机为9针打印机,所以下面以9针打印机为例来进行分析。

7.2 打印屏幕驱动程序

当需要在打印机上打印出当前屏幕上显示的内容时,只要按下“打印屏幕”键(Shift-PrtSc),就驱动了5类中断程序(打印屏幕驱动程序),来完成这项工作。

CC-BIOS的5类中断程序较简单,它由RAM部分和ROM部分组成。其中的RAM部分是新增部分,ROM部分是ROM-BIOS中的字符打印输出部分。

5类中断程序的入口地址为CS:1FD3,其执行过程如下:

首先判断CRT的当前工作方式,如是图形工作方式,则向17类中断程序发送ESC“W”控制命令,由17类中断程序中的有关模块(这部分内容在本章第六节中介绍)去解释这个命令,以实现把当前屏幕上显示的内容在打印机上以图形方式打印出来;如果CRT的工作方式为字符方式,则调用本中断程序中的ROM部分,实现把当前屏幕上显示的内容送打印机以字符方式打印出来。图7-1给出了5类中断程序ROM部分的执行流程。

根据图7-1可知,这部分程序所做的工作是把CRT刷新区中当前页的内容逐个字符地送打印机输出。其中的显示标志能反映出CRT当前是否正在进行显示。如果CRT当前正在进行显示,则其屏幕内容尚未稳定,故暂时不能向打印机输出内容,一定要在它显示完成后才能输出其内容到打印机。显示标志为一内存单元,其地址为0050:0000。当其内容为1时,表示CRT正在

显示；当其内容为 0 时，表示 CRT 完成显示。判断 CRT 是否正在显示，只要判断这个单元的内容即可。另外，在把当前屏幕显示的内容送打印机输出前，要置显示标志为“正在显示”，这是为了防止在这段时间中屏幕内容发生变化，同时也等于锁住了“打印屏幕”键。一旦打印屏幕内容完毕，要立即把显示标志置为“显示完成。”

7.3 打印机驱动程序的组成和输出字型

打印机驱动程序（17类中断程序）由功能块和工作区组成。它共有四个功能块，分别完成相应的功能。表 7-1 列出了它们的具体情况。调用这些功能块的方法是：把功能块号送入 AH 寄存器，再调用 17 类软中断。

功能块号	入口地址	完成功能
0	CS: 1 ABF	把字符送打印机输出
1	CS: 1 B 16	对打印机初始化后返回状态
2	CS: 1 ABA	取打印机状态
3	CS: 200 A	确定打印机的行宽

这些功能块中最重要的是 0 号功能块，CC-BIOS 中实现汉字打印输出的部分包含在此功能块中。我们将要重点对它进行分析。

在介绍 17 类中断程序的工作区和分析各功能块之前，先介绍一下 CC-BIOS 的汉字打印输出字型。因为有的工作区和程序与此有关。

CC-BIOS (V2.1) 向用户提供了十六种汉字打印输出字型。通过键盘命令（由 CC-BIOS 的键盘管理模块接收）可以指定打印输出的字型。以普通字型为基准，其他字型是密集打印、纵向扩展、横向扩展和横向字的实施，以及它们组合的实施。所谓密集打印，就是使打印的点阵更紧密些；所谓纵向扩展（纵扩），就是在垂直方向上把字扩展一倍；所谓横向扩展（横扩），就是在水平方向上把字扩展一倍；所谓横向字，就是把汉字按逆时针方向旋转 90 度打印出来。把这四种情况组合起来就构成了一些复合字型。

7.4 打印机驱动程序的工作区

现在介绍一下打印机驱动程序中的几个主要工作区和工作单元。

1. 打印数据缓冲区（简称缓冲区）

用于存放要送往打印机输出的打印数据，并用它来对字模点阵进行重新排列处理。在图形方式下（打印汉字就是在此方式下进行的）要用到此缓冲区。它共分成四个分区。即缓冲区 1、缓

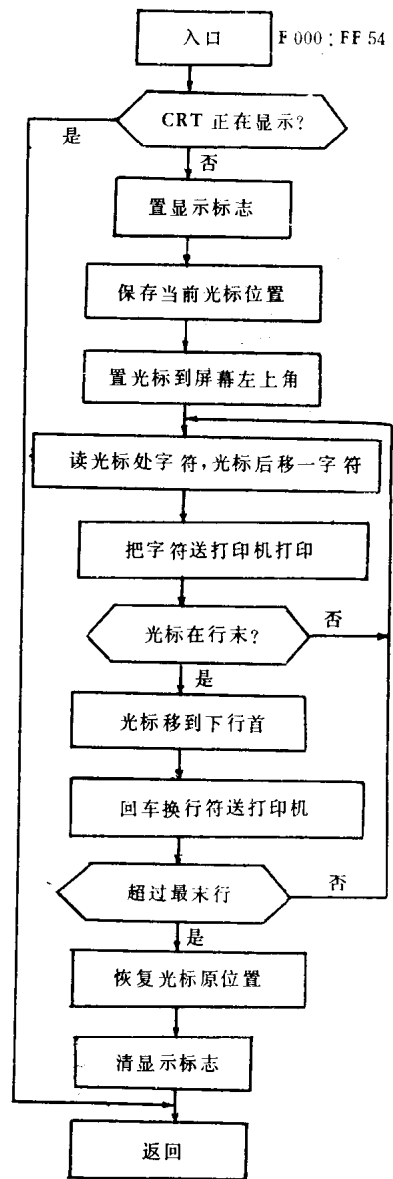


图 7-1 5 类中断程序 ROM 部分的流程图

缓冲区 2、缓冲区 3 和缓冲区 4，每个分区均为 816 个字。下面介绍这四个分区的作用。

缓冲区 1 和缓冲区 2：它们的地址分别为 CS:00E2~0741 和 CS:0742~0DA1，它们用于存放纵向扩展的汉字上半部之打印数据。

缓冲区 3 和缓冲区 4：它们的地址分别为 CS:0DA2~1401 和 CS:1402~1A61，它们用于存放纵向扩展的汉字下半部之打印数据。对于非纵向扩展的汉字，缓冲区 3 存放其上半部打印数据，缓冲区 4 存放其下半部打印数据。对于一般字符，其打印数据存放在缓冲区 4 中。

2. 属性工作区

地址为 CS:0016~00E1，用于存放要打印之字符的属性字（它表明该字符是按一般图形打印，还是按密集图形打印）。它共有 204 个字节，每一个字节对应于缓冲区中的八个字节（因一个字符有八个打印数据），即指明这八个字节的打印属性。

3. 状态字

地址为 CS:000F~0010，用于表示打印的几种状态。比如，是否密集打印，是否横向扩展，是否纵向扩展，是否为横向字，前一字符是否为 ESC，前两字符是否为 ESC“I”等等。它们均由状态字的指定位来表示。根据状态字的内容，可以判断出当前字符是否属于控制命令。以及当前要打印的是何种字型。表 7-2 列出了状态字的有关位号所表示之内容。

状态字各位情况

表 7-2

地址	位号	表示内容	地址	位号	表示内容
000F	0	当前行扩展	000F	5	纵向扩展
	1	密集打印		6	I 状态
	2	非图形		7	ESC 状态
	4	横向扩展	0010	5	横向字体

4. 缓冲区计数器

地址为 CS:000D—000E，它记录了缓冲区中当前所含打印数据的个数。

5. 汉字机内码标记

地址为 CS:0011，记录前一个输出字符是否为汉字机内码符。

6. 打印输出字符保存单元

地址为 CS:0007，用以保存当前打印输出的字符。

7. 打印机状态字保存单元

地址为 CS:0008，用以保存打印机的当前状态字。

另外还有一些工作单元，将在后面涉及到。

7.5 打印机驱动程序的总体

现在，我们对打印机驱动程序（17类中断程序）的总体进行一些分析。

17类中断程序的入口地址为 CS:1A62，入口参数为：AH 寄存器中是调用的功能块号，DX 寄存器中是所用打印机的编号。图 7-2 给出了 17类中断程序的总体流程。从图 7-2 可以看出，17类中断程序的总体流程主要完成一些初始化工作，具体工作由各功能块分别完成。另外，17类中断程序会返回打印字符（在 AL 寄存器中）和打印机当前状态（在 AL 寄存器中）给调用者。

17类中断程序的四个功能块中，0号功能块是最主要和最复杂的，我们把它放到后面对它进

行专门分析。其他三个功能块（1号—3号）都很简单，下面对它们作一剖析。

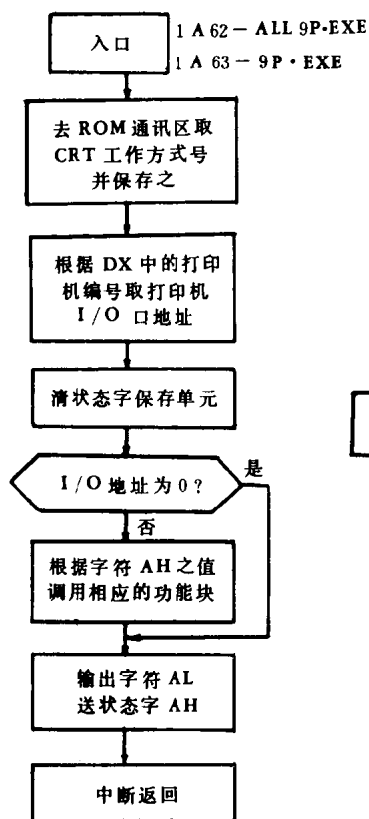


图 7-2 17类中断程序的流程

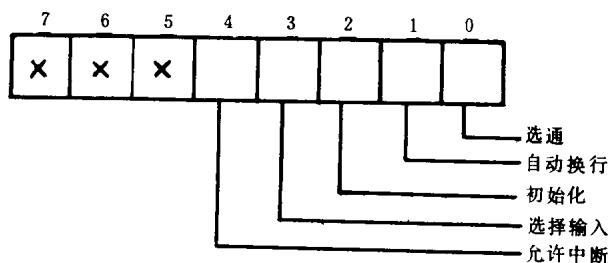


图 7-3 打印机的控制字

1号功能块

此功能块所做的工作是对打印机进行初始化，并返回打印机状态。它的执行过程如下：（入口地址为CS:1B15）

1) 向打印机的初始化线发一个正跳变信号，以进行初始化。其具体实现为：把打印机控制字的初始化位（第2位）清为0，经延时后，再把该位置成1，打印机控制字寄存器的I/O口地址为03BE。图7-3给出了打印机控性字各位所表示的内容。

2) 置打印机为不许中断和自动换行。即把打印机控制字的第1位和第4位清为0。

3) 取打印机的状态字。打印机状态字寄存器的I/O口地址为03BD。

4) 把状态字和输出字符分别存入状态字保存单元和输出字符保存单元。然后返回。

2号功能块:

此功能块所完成的工作是把打印机状态字返回给调用者。它的执行过程如下：（入口址为CS:1AB9）

1) 从打印机状态字寄存器取打印机状态字到AH寄存器。

2) 把AH寄存器内容的后三位清为0（状态字的后三位不用）。

3) 把AH寄存器内容送状态字保存单元，然后返回。

打印机的状态字通过AH寄存器返回给调用者。

3号功能块:

此功能块所完成的工作是确定打印机的行宽。CC-BIOS提供了定义打印机行宽的键盘命令，这个命令由键盘管理模块接收，然后由该模块调用本功能块来实现打印机行宽的定义。本功能块的入口地址为CS:200A。入口参数是，AL寄存器中存放定义的打印机行宽数。它的执行过程如下：

判别定义的行宽是否超出规定的最大行宽。如超出，则把规定的最大行宽存入行宽定义单元。如不超出，则把定义的行宽存入行宽定义单元。

行宽定义单元的地址为CS:000B。本单元用于存放行宽数，打印机在输出时，要按其中的行宽来完成行输出，从而实现了行宽的定义。

7.6 17类中断程序的0号功能块

0号功能块是17类中断程序的主功能块。它的功能是把字符送打印机打印输出。当然，对于不同的字符，它要进行相应的处理。图7-4给出了该功能块的工作流程。

从图7-4中我们可以看到，本功能块在一开始就根据CRT的工作方式分为两个分支，即字符处理分支和图形处理分支。字符处理分支较简单，只要把欲打印输出的字符直接送打印机输出即可。不过值得提一下的是，在本系统中，对于打印机没有采用常用的中断方式来实现字符的输

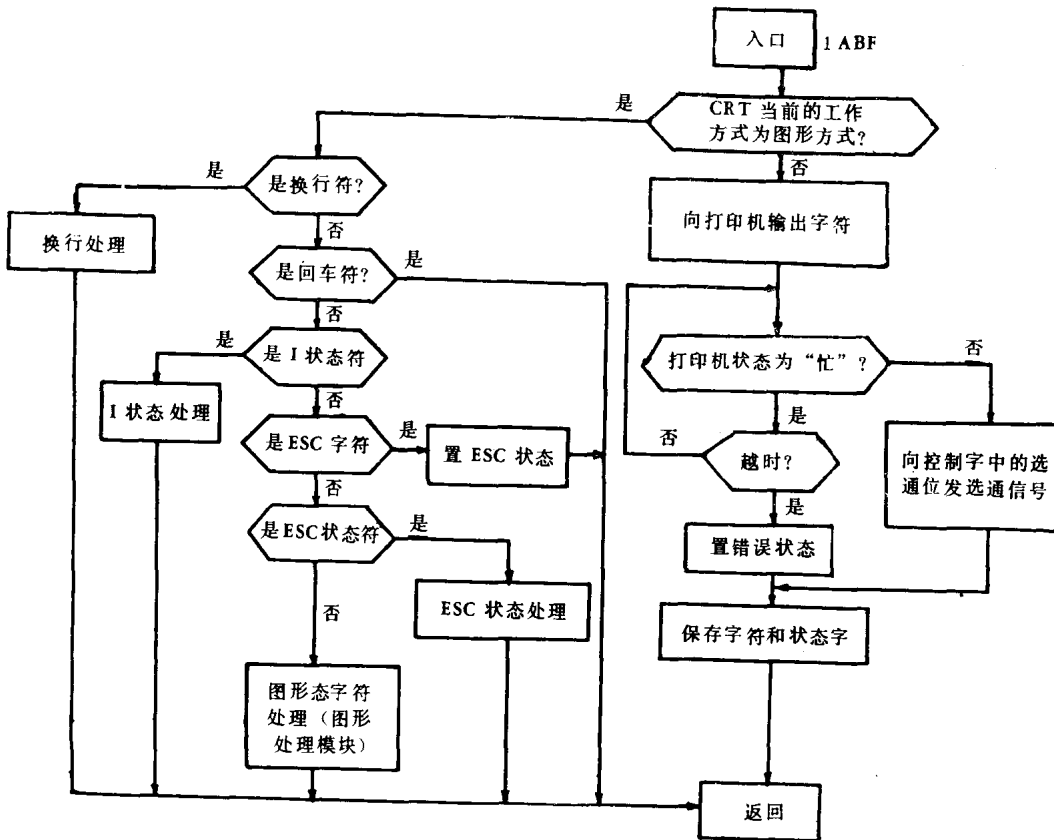


图 7-4 17类中断程序0号功能块流程

出（在初始化时，把打印机置成不许中断的目的就在于此），而是采用查询方式来实现的。所以要从打印机的状态寄存器中取状态字进行判断，如打印机为“忙”则等待，否则就向打印机控制字的选通位（见图 7-3）发选通信号，从而完成向打印机输出一个字符。图形处理分支则要复杂得多，下面对这部分流程作几点说明。

1) 换行处理模块要完成下列工作，判别缓冲区是否为空（即判别缓冲区计数是否为 0），若为空则进行行间距为 8/72 英寸的换行（图形方式下打印时，相邻点间的纵向间隔为 1/72 英寸。），若不空则调用输出缓冲区内容模块，把缓冲区中的全部内容送打印机输出，然后再进行行间距为 8/72 英寸的换行。

输出缓冲区内容模块的工作流程，如图 7-5 所示。

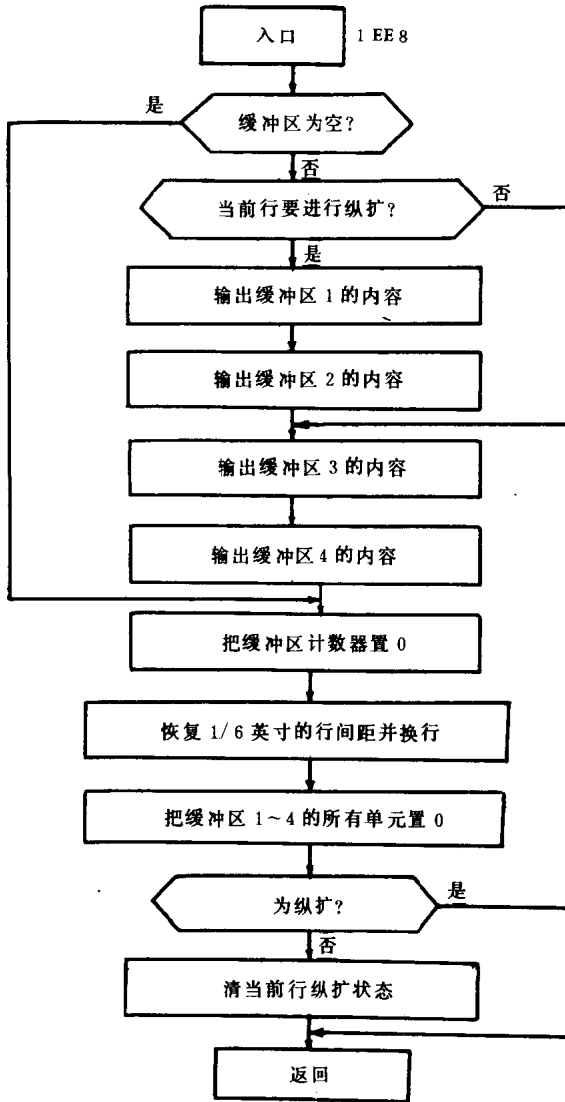


图 7-5 输出缓冲区内容模块流程

对图 7-5 要说明一点, 就是所谓当前行要进行纵扩, 是指某一行至少有一个要进行纵向扩展的字存在, 状态字的第 0 位反映了这一点 (请见表 7-2)。

2) I 状态处理模块所完成的工作是改变汉字的字型。在键盘管理模块 (16 类中断程序) 中, 向用户提供了一个改变打印输出的汉字字型的功能键“F10”。实际上, 16 类中断程序并没有对此功能进行处理。而只是当接收到此功能键的代码后, 用通讯的手段向 17 类中断程序发出 ESC “I” 命令, 后跟一个字母 (“A” ~ “P” 中之一) 代表字型的选择。I 状态处理模块就是对 ESC “I” 命令进行解释处理。它的主要工作是: 根据选择的字型, 分别把状态字中的有关位置以相应的值 (参阅表 7-2)。

3) ESC 态处理模块的功能是对一些控制命令进行处理。其中主要是处理由 5 类中断程序发来的 ESC “W” 控制命令 (参阅本章第二节)。实现在图形方式下, 把当前屏幕上显示的内容送打印机输出。本处理模块是这样来做的: 把 CRT 刷新区中的内容 (即当前屏幕上显示的内容) 转换成打印数据, 再送打印机输出。然后清除 ESC 状态和 I 状态 (把状态字的第 6 位和第 7 位清为 0)。CRT 刷新区的地址为 B000:8000~9F3F (偶数线扫描区) 和 B000:A000~BF3F (奇数线扫描区)。

4) 图形态字符处理模块完成在图形方式下对字符进行处理的工作, 是 0 号功能块的主体, 故在本章第七节单独对它进行分析。

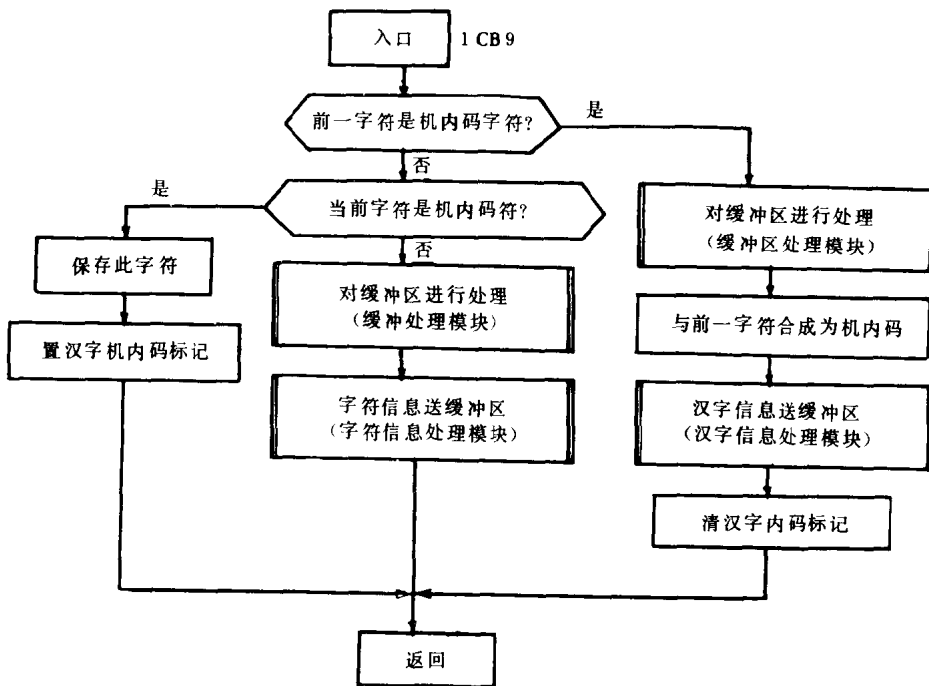


图 7-6 图形态字符处理模块流程

7.7 图形态字符处理模块

本模块是 0 号功能块中的一个重要子程序，它所完成功能是，对一般字符和汉字机内码分别进行处理，再把处理后产生的信息存入缓冲区。本模块的流程如图 7-6 所示。下面对图 7-6 中的有关部分作几点说明。

1) 要判别前一字符是否为机内码符，只要判别汉字机内码标记即可。

2) 在向缓冲区送入信息前，要先调用缓冲区处理模块。该模块的主要任务是判断缓冲区中是否能放下本次要放的信息，如放不下，则先把缓冲区中的内容送打印机输出，然后才能接收新的内容。如能放下，则直接接受新的内容。图 7-7 给出了缓冲区处理模块的详细流程。

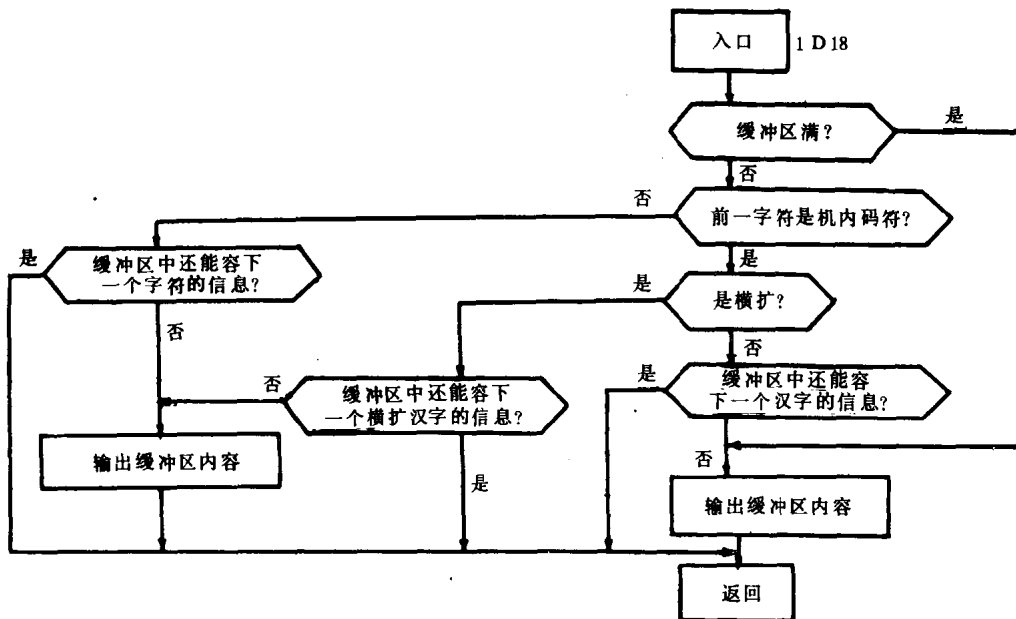


图 7-7 缓冲区处理模块流程

在图 7-7 中，把当前欲存入缓冲区的内容分为三种情况，即字符、汉字和横扩汉字。这三者所对应的打印数据量（即要存入缓冲区的信息量）是不同的。只有当缓冲区能容纳下它们所对应的全部信息量时，才能不输出缓冲区中内容，而把信息直接存入缓冲区。只有这样做了，才能保证输出信息的完整性。否则就有可能出现字的一部分在一行末，另一部分在下行首的不正常情况。

3) 汉字信息处理模块是一个很重要的模块，该模块的主要任务是：把汉字机内码转换成它所对应的打印数据，并把这些打印数据存入缓冲区。该模块的关键是获得打印数据。所谓打印数据，就是在图形方式下欲打印某个汉字或字符时，须送给打印机的数据组。打印数据是从汉字或字符的字模信息转换来的。把字库中的字模信息换成打印数据的过程，就是把横向点阵（适用于显示器）转换成纵向点阵（适用于打印机）的过程。图 7-8 给出了汉字信息处理模块的工作流程）下面对图 7-8 作一些解释。

(i) 因 CC-BIOS 中的汉字库是按国标码顺序排的，而机内码又是采用的异形国标码（国标码的最高位置 1），所以只要经过单简的计算，即可由机内码获得其对应汉字的字模在汉字库中的地址。根据这个地址，就可以取得字模信息。

(ii) 当输出横向字时，形成横向字的打印数据较容易。因为，如把字模信息直接当作打印数据送打印机输出，则打印出的汉字是按顺时针方向旋转了 90 度，而横向字正好与此相反。所以，

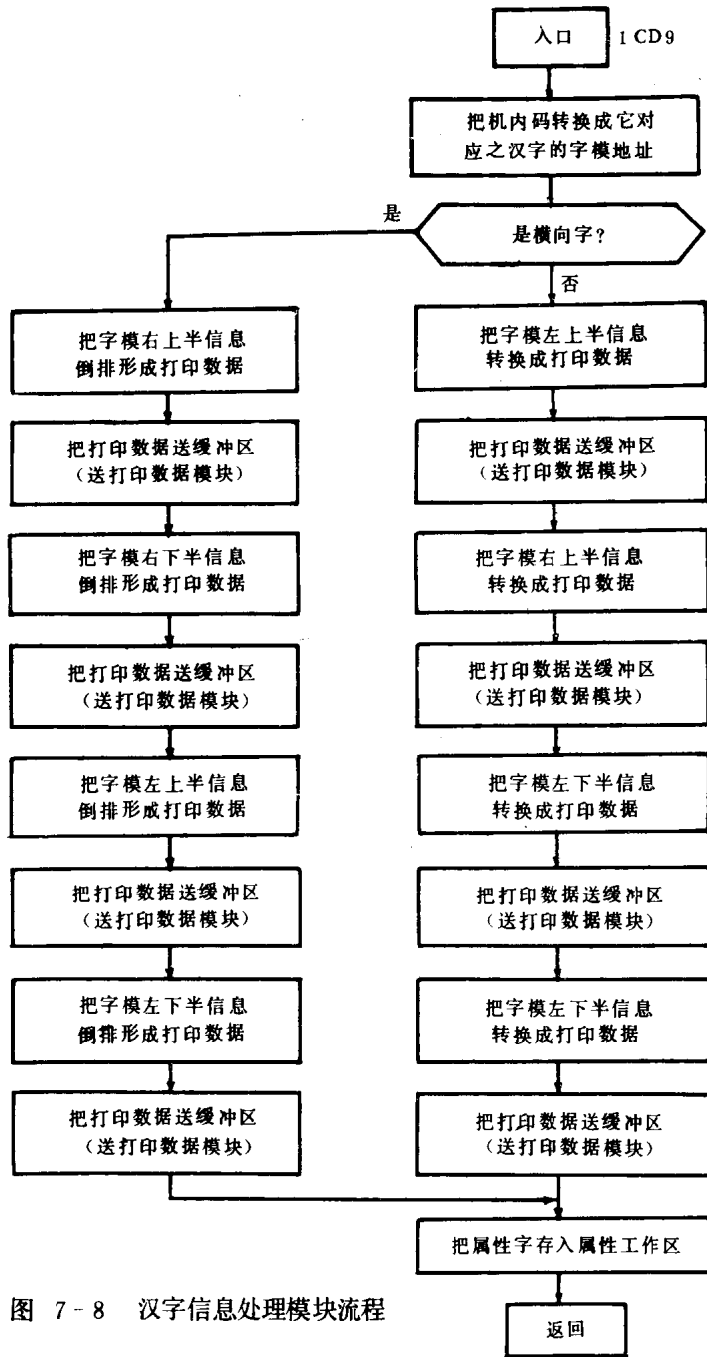


图 7-8 汉字信息处理模块流程

只要把字模信息逐个字节倒置，即可得到横向汉字的打印数据。所谓倒置，就是把字节的内容倒过来排列。如把10101010进行倒置，即为 01010101。

(iii) 非横向字的打印数据（也就是纵向点阵）的取得，是把横向点阵信息经移位和逻辑运算后取得。其实现方法并不难，这儿不去讨论了。

(iv) 所谓属性，实际上就是对打印机进行密集图形打印或进行一般图形打印的打印机控制代码。具体按何种代码作为属性字，这要根据状态字中密集打印位的值来决定。

(v) 把打印数据送缓冲区，是通过调用送打印数据模块来实现的。该模块的功能是按指定的字型，对打印数据进行相应的加工，并把它们送入缓冲区中。该模块的工作流程，如图 7-9 所示。

从图 7-9 中可以知道，对于纵向扩展的汉字，要对它的打印数据进行位重复处理。这是因为在进行纵扩时，在垂直方向上每行都要重复一次。例如，对 10101010 进行位重复后得到 1100110011001100，对于横向扩展的汉字，要对它的打印数据进行字节重复处理。这是因为在进行横扩时，在水平方向上每列都要重复一次。例如，就 10101010 进行字节重复后，得到 1010101010101010。经过这样处理后的打印数据被存入缓冲区中，以后把缓冲区中的内容送打印机输出时，就能获得指定字型的汉字。

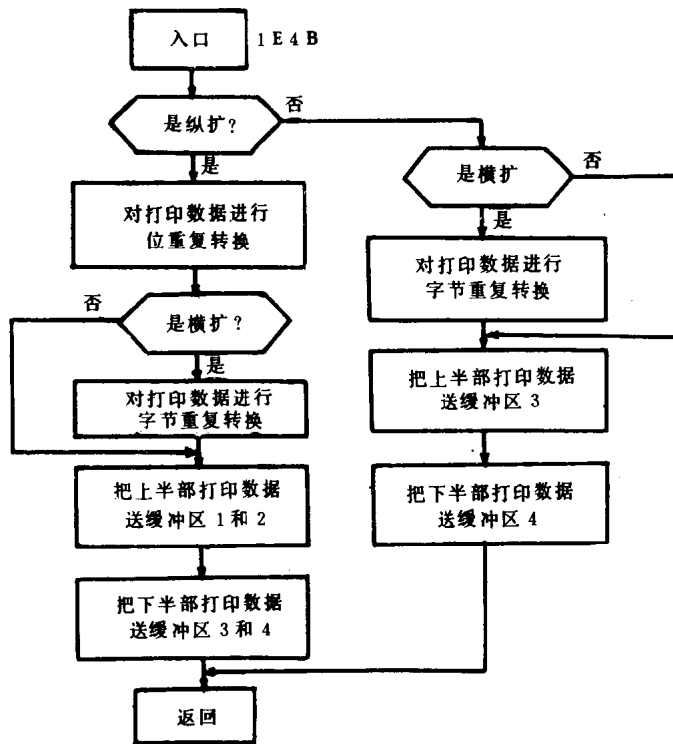


图 7-9 送打印数据模块流程

7.8 打印驱动模块源程序 INT 17H

西文 DOS INT 17H, INT 05H 程序清单可用 Debug 命令得到或查 IBMPC/XT 硬件技术参考手册，其 ROM BIOS 部份有详细文本注解。本节只对汉化后 CC-DOS 中新 INT 17, INT 05 举例如下：若框图与 9 P、.exe 文本有出入请查照 ALL9 P 子程。

C:\debug b:all9p.exe FX-100 A-P 16字型(大字)驱动程序

-r

AX=0000 BX=0000 CX=2080 DX=0000 SP=0000 BP=0000 SI=0000 DI=0000

DS=4DF8 ES=4DF8 SS=4E08 CS=4E08 IP=2038 NV UP DI PL NZ NA PO NC

4E08:2038 0E PUSH CS

-dds:2038

```

4DF8:2038 75 05 80 26 0F 00 FE C3          u..&..~C
4DF8:2040 53 33 DB 53 F8 D5 FE 33-C9 8A 05 3A 05 75 04 41  S3[ShU*3I...u.A
4DF8:2050 47 EB F8 D1 E1 D1 E1 D1-E1 5B 8B F3 5F 8B 16 09  6kxQaQaQa[.s...
4DF8:2060 00 2B D3 3B CA 72 02 8B-CA 03 D9 53 8B DF 57 EB  .+S;Jr..J.YS._Wh
4DF8:2070 0D 00 5B 5B 3B 1E 09 00-72 01 C3 50 53 EB C5 56  ..X[;...r.CPSkEV
4DF8:2080 53 03 F3 51 E8 0C 00 59-51 AC EB 47 FB 59 E2 F8  S.sQh..YQ,h6(Ybx
4DF8:2090 5B 5E C3 51 50 B0 1B EB-3A FB 5B EB 36 FB 59 51  [^CQP0.h:(Xh6{YQ
4DF8:20A0 8A C1 EB 2F FB 59 8A C5-E8 29 FB C3 B0 1B EB 23  .Ah/(Y.Eh)(CO.h#
4DF8:20B0 FB B0 41 EB 1E FB B0 0B          (0Ah.(Y.

```

-dds:00

```

4DF8:0000 CD 20 00 80 00 9A F0 FF-0D F0 42 02 F3 4A 70 02  M....p..pB.sjP.
4DF8:0010 F3 4A E2 04 42 05 F3 4A-01 01 01 00 02 FF FF FF  sjb.B.sj.....
4DF8:0020 FF FF FF FF FF FF FF FF-FF FF FF FF F5 4D C8 2A  .....uMH#
4DF8:0030 F3 4A 00 00 00 00 00 00-00 00 00 00 00 00 00  sj.....
4DF8:0040 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00  .....
4DF8:0050 CD 21 C8 00 00 00 00 00-00 00 00 00 20 20 20  M'K.....
4DF8:0060 20 20 20 20 20 20 20 20-00 00 00 00 00 20 20 20  .....
4DF8:0070 20 20 20 20 20 20 20 20-00 00 00 00 00 00 00 00  .....

```

-d

```

4DF8:0080 00 0D 62 3A 61 6C 6C 39-70 2E 65 7B 65 0D 7B 65  ..b:all9p.exe.exe
4DF8:0090 0D 54 49 4E 47 0D 00-00 00 00 00 00 00 00 00  .TING.....
4DF8:00A0 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00  .....
4DF8:00B0 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00  .....
4DF8:00C0 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00  .....
4DF8:00D0 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00  .....
4DF8:00E0 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00  .....
4DF8:00F0 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00  .....

```

-u

```

4E08:2038 0E          PUSH    CS          ; 改DS=CS
4E08:2039 1F          POP     DS
4E08:203A B81F35      MOV     AX,351F     ; 取1FH向量到ES:BX
4E08:203D CD21      INT     21
4E08:203F 891E0300   MOV     [0003],BX
4E08:2043 BCC3       MOV     BX,ES
4E08:2045 891E0100   MOV     [0001],BX
4E08:2049 B81035      MOV     AX,3510     ; 取10H向量到ES:BX
4E08:204C CD21      INT     21
4E08:204E BB7500      MOV     BX,0075
4E08:2051 26         ES:
4E08:2052 8B07      MOV     AX,[BX]
4E08:2054 2E         CS:
4E08:2055 A31020     MOV     [2010],AX   ; 16x16字模库首址段值→CS:2010

```

-u

```

4E08:2058 8D16621A LEA DX,[1A62]
4E08:205C 8B1725 MOV AX,2517
4E08:205F CD21 INT 21
4E08:2061 8D16D31F LEA DX,[1FD3]
4E08:2065 8B0525 MOV AX,2505
4E08:2068 CD21 INT 21
4E08:206A 8D163820 LEA DX,[2038]
4E08:206E 81C20301 ADD DX,0103
4E08:2072 CD27 INT 27
4E08:2074 0000 ADD [BX+SI],AL
4E08:2076 0000 ADD [BX+SI],AL

4E08:2006 CF IRET
4E08:2007 3E POP SI
4E08:2008 5A POP DX
4E08:2009 58 POP AX
4E08:200A 1F POP DS
4E08:200B EA54FF00F0 JMP F000:FF54
4E08:2010 0000 ADD [BX+SI],AL
4E08:2012 81E27F7F AND DX,7F7F
4E08:2016 50 PUSH AX
4E08:2017 80FE30 CMP DH,30
4E08:201A 7203 JB 201F

```

; 改 INT 17H 入口=CS : 1 A 62

; 改 INT 05H 入口=CS : 1 F D 3

; 新 INT 05H 入口

; ROM BIOS INT 05H 入口

; 字模库段地址单元

-u

```

4E08:201C 80EE08 SUB DH,08
4E08:201F 80EE21 SUB DH,21
4E08:2022 B05E MOV AL,5E
4E08:2024 F6E6 MUL DH
4E08:2026 80EA21 SUB DL,21
4E08:2029 32F6 XOR DH,DH
4E08:202B 03C2 ADD AX,DX
4E08:202D D1E0 SHL AX,1
4E08:202F 2E CS:
4E08:2030 03061020 ADD AX,[2010]
4E08:2034 8BD0 MOV DX,AX
4E08:2036 58 POP AX
4E08:2037 C3 RET
4E08:2038 0E PUSH CS
4E08:2039 1F POP DS
4E08:203A 8B1F35 MOV AX,351F

```

-u

```

4E08:203D CD21 INT 21
4E08:203F 891E0300 MOV [0003],BX
4E08:2043 8CC3 MOV BX,ES
4E08:2045 891E0100 MOV [0001],BX
4E08:2049 8B1035 MOV AX,3510
4E08:204C CD21 INT 21
4E08:204E 8B7500 MOV BX,0075

```

A>c:debug 9p.exe

FX - 100 (I - J) 小字型驱动程序

-r

AX=0000 BX=0000 CX=2100 DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=4DF8 ES=4DF8 SS=4E08 CS=4E08 IP=2081 NV UP DI PL NZ NA PD NC

```

4E08:2081 0E          PUSH    CS
4E08:2081 0E          PUSH    CS
4E08:2082 1F          POP     DS
4E08:2083 B81F35        MOV     AX,351F
4E08:2086 CD21          INT     21
4E08:2088 B91E0300      MOV     [0003],BX
4E08:208C 8CC3          MOV     BX,ES
4E08:208E B91E0100      MOV     [0001],BX
4E08:2092 B81035        MOV     AX,3510
4E08:2095 CD21          INT     21
4E08:2097 BB7500        MOV     BX,0075      ; 取10HCCLIB16×16字模库起始段址
4E08:209A 26          ES:      ; 在ES:BX = ES: 0075
4E08:209B 8B07          MOV     AX,[BX]
4E08:209D 2E          CS:
4E08:209E A35920        MOV     [2059],AX    ; 字模库起始段值→CS : 2059

```

-u

```

4E08:20A1 8D16631A      LEA     DX,[1A63]    ; 改INT 17新向量到CS : 1A 63
4E08:20A5 B81725        MOV     AX,2517
4E08:20A8 CD21          INT     21
4E08:20AA 8D161C20      LEA     DX,[201C]    ; 改INT 05新向量到CS : 201 C
4E08:20AE B80525        MOV     AX,2505
4E08:20B1 CD21          INT     21
4E08:20B3 8D168120      LEA     DX,[2081]    ; 新向量终止地址加点余量
4E08:20B7 81C20301      ADD     DX,0103
4E08:20BB CD27          INT     27          ; 驻留内存
4E08:20BD 0000          ADD     [BX+SI],AL
4E08:20BF 0000          ADD     [BX+SI],AL

```

-ucs:1a63

```

4E08:1A63 FC          CLD     ; INT 17H新向量入口
4E08:1A64 FB          STI
4E08:1A65 1E          PUSH    DS
4E08:1A66 52          PUSH    DX
4E08:1A67 56          PUSH    SI
4E08:1A68 51          PUSH    CX
4E08:1A69 53          PUSH    BX
4E08:1A6A 06          PUSH    ES
4E08:1A6B 57          PUSH    DI
4E08:1A6C 50          PUSH    AX
4E08:1A6D BE084E      MOV     SI,4E08
4E08:1A70 BEDE          MOV     DS,SI
4E08:1A72 BE4000      MOV     SI,0040
4E08:1A75 BEC6          MOV     ES,SI
4E08:1A77 BE4900      MOV     SI,0049

```


4E08:1A7A 26	ES:		
4E08:1A7B 8A04	MOV	AL,[SI]	
4E08:1A7D 2E	CS:		
4E08:1A7E A20000	MOV	[0000],AL	
4E08:1A81 D1E2	SHL	DX,1	
-u			
4E08:1A83 BE0800	MOV	SI,0008	
4E08:1A86 03F2	ADD	SI,DX	
4E08:1A88 26	ES:		
4E08:1A89 8B14	MOV	DX,[SI]	
4E08:1A8B 58	POP	AX	
4E08:1A8C 1E	PUSH	DS	
4E08:1A8D 07	POP	ES	
4E08:1A8E 89160500	MOV	[0005],DX	
4E08:1A92 C606080000	MOV	BYTE PTR [0008],00	
4E08:1A97 08D2	OR	DX,DX	
4E08:1A99 7413	JZ	1AAE	
4E08:1A9B 0AE4	OR	AH,AH	; AH = 0 0号功能, 字符打印转出
4E08:1A9D 7420	JZ	1ABF	
4E08:1A9F FECC	DEC	AH	; AH = 1 1号功能, 打印机初始化
4E08:1AA1 7473	JZ	1B16	
-u			
4E08:1AA3 FECC	DEC	AH	; AH = 2 2号功能, 读打印口状态
4E08:1AA5 7413	JZ	1ABA	
4E08:1AA7 FECC	DEC	AH	; AH = 3 3号功能, 确定行宽
4E08:1AA9 7503	JNZ	1AAE	
4E08:1AAB E95C05	JMP	200A	
4E08:1AAE 26	ES:		; 打印机状态字→AX, 中断返回
4E08:1AAF A10700	MOV	AX,[0007]	
4E08:1AB2 5F	POP	DI	
4E08:1AB3 07	POP	ES	
4E08:1AB4 5B	POP	BX	
4E08:1AB5 59	POP	CX	
4E08:1AB6 5E	POP	SI	
4E08:1AB7 5A	POP	DX	
4E08:1AB8 1F	POP	DS	
4E08:1AB9 CF	IRET		
4E08:1ABA EB4100	CALL	1AFE	; 2号功能, 读打印口状态入口
4E08:1ABD EBEB	JMP	1AAE	
4E08:1ABF 803E000004	CMP	BYTE PTR [0000],04	; 0号功能, 字符打印转出入口
-u			
4E08:1AC4 720A	JB	1AD0	
4E08:1AC6 803E000007	CMP	BYTE PTR [0000],07	
4E08:1ACB 7403	JZ	1AD0	
4E08:1ACD EB5B	JMP	1B2A	
4E08:1ACF 90	NOP		
4E08:1AD0 EB0200	CALL	1AD5	
4E08:1AD3 EBD9	JMP	1AAE	

4E08:1AD5 50	PUSH	AX	
4E08:1AD6 26	ES:		
4E08:1AD7 8B160500	MOV	DX,[0005]	
4E08:1ADB B30A	MOV	BL,0A	
4E08:1ADD 33C9	XOR	CX,CX	
4E08:1ADF EE	OUT	DX,AL	
4E08:1AE0 42	INC	DX	
4E08:1AE1 EC	IN	AL,DX	: 读打印机状态字
4E08:1AE2 8AE0	MOV	AH,AL	
-u			
4E08:1AE4 A8B0	TEST	AL,B0	
4E08:1AE6 750E	JNZ	1AF6	
4E08:1AE8 E2F7	LOOP	1AE1	
4E08:1AEA FECB	DEC	BL	
4E08:1AEC 75F3	JNZ	1AE1	
4E08:1AEE 80CC01	OR	AH,01	
4E08:1AF1 80E4F9	AND	AH,F9	
4E08:1AF4 EB15	JMP	1B0B	
4E08:1AF6 B00D	MOV	AL,0D	: 回车码输出
4E08:1AFB 42	INC	DX	
4E08:1AF9 EE	OUT	DX,AL	
4E08:1AFA B00C	MOV	AL,0C	: 换页码输出
4E08:1AFC EE	OUT	DX,AL	
4E08:1AFD 5B	POP	AX	
4E08:1AFE 50	PUSH	AX	: 读打印口状态
4E08:1AFF 26	ES:		
4E08:1B00 8B160500	MOV	DX,[0005]	
-u			
4E08:1B04 42	INC	DX	
4E08:1B05 EC	IN	AL,DX	
4E08:1B06 8AE0	MOV	AH,AL	
4E08:1B08 80E4FB	AND	AH,FB	: 取高 5 位
4E08:1B0B 5A	POP	DX	
4E08:1B0C 8AC2	MOV	AL,DL	
4E08:1B0E 80F44B	XOR	AH,4B	: 取AH 中位 3, 4 异或
4E08:1B11 26	ES:		
4E08:1B12 A30700	MOV	[0007],AX	
4E08:1B15 C3	RET		
4E08:1B16 83C202	ADD	DX,+02	: INT 17 1号功能
4E08:1B19 B00B	MOV	AL,0B	
4E08:1B1B EE	OUT	DX,AL	
4E08:1B1C 88E803	MOV	AX,03EB	
4E08:1B1F 4B	DEC	AX	
4E08:1B20 75FD	JNZ	1B1F	
4E08:1B22 B00C	MOV	AL,0C	
-u			
4E08:1B24 EE	OUT	DX,AL	
4E08:1B25 E8D6FF	CALL	1AFE	

4E08:1B28 E884	JMP	1AAE	
4E08:1B2A 3C0A	CMP	AL,0A	: 换行吗? 图形打印方式处理入口
4E08:1B2C 7518	JNZ	1B46	
4E08:1B2E 833E090000	CMP	WORD PTR [0009],+00	
4E08:1B33 7506	JNZ	1B3B	
4E08:1B35 E8BD04	CALL	1FF5	
4E08:1B38 EB04	JMP	1B3E	
4E08:1B3A 90	NOP		
4E08:1B3B E8A803	CALL	1EE6	
4E08:1B3E C606120000	MOV	BYTE PTR [0012],00	
4E08:1B43 E968FF	JMP	1AAE	
-u			
4E08:1B46 3C0D	CMP	AL,0D	: 回车?
4E08:1B48 7503	JNZ	1B4D	
4E08:1B4A E961FF	JMP	1AAE	
4E08:1B4D F6060F0040	TEST	BYTE PTR [000F],40	
4E08:1B52 7403	JZ	1B57	
4E08:1B54 E90301	JMP	1C5A	
4E08:1B57 3C1B	CMP	AL,1B	: esc ?
4E08:1B59 7503	JNZ	1B5E	
4E08:1B5B E9F400	JMP	1C52	
4E08:1B5E F6060F0080	TEST	BYTE PTR [000F],80	
4E08:1B63 7446	JZ	1BAB	
4E08:1B65 3C49	CMP	AL,49	: "I" ?
-u			
4E08:1B67 7512	JNZ	1B7B	
4E08:1B69 80260F0001	AND	BYTE PTR [000F],01	
4E08:1B6E C606100000	MOV	BYTE PTR [0010],00	
4E08:1B73 800E0F00C0	OR	BYTE PTR [000F],C0	
4E08:1B7B E933FF	JMP	1AAE	
4E08:1B7B F6060F0004	TEST	BYTE PTR [000F],04	
4E08:1B80 7406	JZ	1B88	
4E08:1B82 E84800	CALL	1BCD	
4E08:1B85 E948FF	JMP	1AD0	
-u			
4E08:1B88 3C4B	CMP	AL,4B	: "K" ?
4E08:1B8A 7506	JNZ	1B92	
4E08:1B8C EB2900	CALL	1BB8	
4E08:1B8F E91CFF	JMP	1AAE	
4E08:1B92 3C4C	CMP	AL,4C	: "L" ?
4E08:1B94 7508	JNZ	1B9E	
4E08:1B96 80260F003F	AND	BYTE PTR [000F],3F	
4E08:1B9B E91B01	JMP	1CB9	
4E08:1B9E 3C57	CMP	AL,57	: "W" ?
4E08:1BA0 7503	JNZ	1BA5	
4E08:1BA2 EB34	JMP	1BD8	
4E08:1BA4 90	NOP		
4E08:1BA5 EB1000	CALL	1B88	

```

-u
4E08:1B88 E903FF      JMP     1AAE
4E08:1B8B F6060F0004  TEST   BYTE PTR [000F],04
4E08:1BB0 7503         JNZ    1BB5
4E08:1BB2 E90401      JMP     1CB9
4E08:1BB5 E918FF      JMP     1AD0
4E08:1BB8 50          PUSH   AX
4E08:1BB9 B01B         MOV    AL,1B
4E08:1BBB EB17FF      CALL   1AD5
4E08:1BBE 5B          POP    AX
4E08:1BBF EB13FF      CALL   1AD5
4E08:1BC2 C6060F0004  MOV    BYTE PTR [000F],04
4E08:1BC7 C6060F0000  MOV    BYTE PTR [000F],00
-u
4E08:1BCC C3          RET
4E08:1BCD 80260F007F  AND    BYTE PTR [000F],7F
4E08:1BD2 B01B         MOV    AL,1B
4E08:1BD4 EBFEFE      CALL   1AD5
4E08:1BD7 C3          RET
4E08:1BD8 B800B8      MOV    AX,BB00 ; B 800 →DS
4E08:1BDB B8DB      MOV    DS,AX
4E08:1BDD 33F6      XOR    SI,SI
4E08:1BDF E8DF03      CALL   1FC1
4E08:1BE2 B91900      MOV    CX,0019
4E08:1BE5 51          PUSH   CX
4E08:1BE6 EB1200      CALL   1BFB
4E08:1BE9 56          PUSH   SI
4E08:1BEA EB2300      CALL   1C10
-u
4E08:1BED 5E          POP    SI
4E08:1BEE B1C64001  ADD    SI,0140
4E08:1BF2 E8CC03      CALL   1FC1
4E08:1BF5 59          POP    CX
4E08:1BF6 E2ED      LOOP   1BE5
4E08:1BF8 EB4B      JMP    1C42
4E08:1BFA 90          NOP
4E08:1BFB 51          PUSH   CX
4E08:1BFE B98002      MOV    CX,0280
4E08:1BFF B04C      MOV    AL,4C
4E08:1C01 26          ES:
4E08:1C02 F6060F0002  TEST   BYTE PTR [000F],02
4E08:1C07 7502         JNZ    1C0B
4E08:1C09 B04B      MOV    AL,4B
4E08:1C0B EB9A03      CALL   1FAB
-u
4E08:1C0E 59          POP    CX
4E08:1C0F C3          RET
4E08:1C10 51          PUSH   CX

```

4E08:1C11	B95000	MOV	CX,0050	
4E08:1C14	51	PUSH	CX	
4E08:1C15	B90800	MOV	CX,0008	
4E08:1C18	EB0800	CALL	1C23	
4E08:1C18	E2FB	LOOP	1C18	
4E08:1C1D	59	POP	CX	
4E08:1C1E	46	INC	SI	
4E08:1C1F	E2F3	LOOP	1C14	
4E08:1C21	59	POP	CX	
4E08:1C22	C3	RET		
4E08:1C23	51	PUSH	CX	
4E08:1C24	56	PUSH	SI	
4E08:1C25	BF0400	MOV	DI,0004	
4E08:1C28	8A24	MOV	AH,[SI]	
4E08:1C2A	D2EC	SHR	AH,CL	
4E08:1C2C	D0D0	RCL	AL,1	
-u				
4E08:1C2E	BAA40020	MOV	AH,[SI+2000]	
4E08:1C32	D2EC	SHR	AH,CL	
4E08:1C34	D0D0	RCL	AL,1	
4E08:1C36	83C650	ADD	SI,+50	
4E08:1C39	4F	DEC	DI	
4E08:1C3A	75EC	JNZ	1C28	
4E08:1C3C	EB96FE	CALL	1AD5	
4E08:1C3F	5E	POP	SI	
4E08:1C40	59	POP	CX	
4E08:1C41	C3	RET		
4E08:1C42	26	ES:		
4E08:1C43	C70609000000	MOV	WORD PTR [0009],0000	
4E08:1C49	26	ES:		
4E08:1C4A	80260F003F	AND	BYTE PTR [000F],3F	
-u				
4E08:1C4F	E95CFE	JMP	1AAE	
4E08:1C52	800E0F0080	OR	BYTE PTR [000F],80	
4E08:1C57	E954FE	JMP	1AAE	
4E08:1C5A	33DB	XOR	BX,BX	
4E08:1C5C	3C49	CMP	AL,49	: "I" ?
4E08:1C5E	720E	JB	1C6E	
4E08:1C60	3C50	CMP	AL,50	: "P" ?
4E08:1C62	7605	JBE	1C69	
4E08:1C64	B041	MOV	AL,41	
4E08:1C66	EB06	JMP	1C6E	
4E08:1C68	90	NOP		
4E08:1C69	80CB02	OR	BL,02	
4E08:1C6C	2C08	SUB	AL,08	
4E08:1C6E	3C45	CMP	AL,45	: "E" ?
-u				
4E08:1C70	7205	JB	1C77	

4E08:1C72 80CF20	OR	BH,20	
4E08:1C75 2C04	SUB	AL,04	
4E08:1C77 3C41	CMP	AL,41	: "A" ?
4E08:1C79 7703	JA	1C7E	
4E08:1C7B EB18	JMP	1C95	
4E08:1C7D 90	NOP		
4E08:1C7E 3C42	CMP	AL,42	: "B" ?
4E08:1C80 7506	JNZ	1C88	
4E08:1C82 80CB10	OR	BL,10	
4E08:1C85 E80E	JMP	1C95	
4E08:1C87 90	NOP		
4E08:1C88 3C43	CMP	AL,43	: "C" ?
4E08:1C8A 7506	JNZ	1C92	
4E08:1C8C 80CB21	OR	BL,21	
4E08:1C8F E804	JMP	1C95	
-u			
4E08:1C91 90	NOP		
4E08:1C92 80CB31	OR	BL,31	
4E08:1C95 80260F0001	AND	BYTE PTR [000F],01	
4E08:1C9A 8026100000	AND	BYTE PTR [0010],00	
4E08:1C9F 081E0F00	OR	[000F],BL	
4E08:1CA3 083E1000	OR	[0010],BH	
4E08:1CA7 F6C320	TEST	BL,20	
4E08:1CAA C606110000	MOV	BYTE PTR [0011],00	
4E08:1CAF 7405	JZ	1CB6	
-u			
4E08:1CB1 C606110003	MOV	BYTE PTR [0011],03	
4E08:1CB6 E9F5FD	JMP	1AAE	
4E08:1CB9 F606120008	TEST	BYTE PTR [0012],08	: 图形态字符处理
4E08:1CBE 7403	JZ	1CC3	
4E08:1CC0 EB17	JMP	1CD9	
4E08:1CC2 90	NOP		
4E08:1CC3 3CA1	CMP	AL,A1	
4E08:1CC5 720F	JB	1CD6	
4E08:1CC7 3CF8	CMP	AL,F8	
4E08:1CC9 730B	JNB	1CD6	
4E08:1CCB C606120008	MOV	BYTE PTR [0012],08	
4E08:1CD0 A21300	MOV	[0013],AL	
-u			
4E08:1CD3 E9D8FD	JMP	1AAE	
4E08:1CD6 EB21	JMP	1CF9	
4E08:1CDB 90	NOP		
4E08:1CD9 E83C00	CALL	1D18	: 汉字信息处理模块
4E08:1CDC 8A261300	MOV	AH,[0013]	
4E08:1CE0 257F7F	AND	AX,7F7F	: 取国标(除高位)
4E08:1CE3 8BD0	MOV	DX,AX	
4E08:1CE5 EB7303	CALL	205B	: 计算汉字内码的字模地址
4E08:1CE8 8EDA	MOV	DS,DX	

```

4E0B:1CEA 33F6      XOR     SI,SI
4E0B:1CEC EB6A00     CALL   1D59
4E0B:1CEF 06        PUSH   ES
4E0B:1CF0 1F        POP    DS
4E0B:1CF1 C606120000     MOV    BYTE PTR [0012],00
-u
4E0B:1CF6 E9B5FD      JMP    1AAE
4E0B:1CF9 EB1C00     CALL   1D1B
4E0B:1CFC 32E4      XOR    AH,AH
4E0B:1CFE 247F      AND    AL,7F
4E0B:1D00 BB160100     MOV    DX,[0001]
4E0B:1D04 BB360300     MOV    SI,[0003]
4E0B:1D08 BEBA      MOV    DS,DX
4E0B:1D0A D1E0      SHL    AX,1
4E0B:1D0C D1E0      SHL    AX,1
4E0B:1D0E D1E0      SHL    AX,1
4E0B:1D10 03F0      ADD    SI,AX
4E0B:1D12 EB9301     CALL   1EAB
4E0B:1D15 E996FD      JMP    1AAE
-u
4E0B:1D18 52        PUSH   DX                : 缓冲区处理模块
4E0B:1D19 BB160B00     MOV    DX,[000B]
4E0B:1D1D 39160D00     CMP    [000D],DX
4E0B:1D21 7207      JB     1D2A
4E0B:1D23 50        PUSH   AX
4E0B:1D24 EBBF01     CALL   1EE6
4E0B:1D27 5B        POP    AX
4E0B:1D28 5A        POP    DX
4E0B:1D29 C3        RET
4E0B:1D2A F606120078     TEST   BYTE PTR [0012],78
4E0B:1D2F 7412      JZ     1D43
4E0B:1D31 F6060F0010     TEST   BYTE PTR [000F],10    : 测状态位. 横扩?
4E0B:1D36 7516      JNZ    1D4E
-u
4E0B:1D38 B3EA10     SUB    DX,+10
4E0B:1D3B 39160D00     CMP    [000D],DX
4E0B:1D3F 77E2      JA     1D23
4E0B:1D41 EBE5      JMP    1D28
4E0B:1D43 B3EA08     SUB    DX,+08
4E0B:1D46 39160D00     CMP    [000D],DX
4E0B:1D4A 77D7      JA     1D23
4E0B:1D4C EBDA      JMP    1D28
4E0B:1D4E B3EA20     SUB    DX,+20
4E0B:1D51 39160D00     CMP    [000D],DX
4E0B:1D55 77CC      JA     1D23
4E0B:1D57 EBCF      JMP    1D28
-u
4E0B:1D59 26        ES:

```

4E08:1D5A F6061000A0	TEST	BYTE PTR [0010],A0
4E08:1D5F 756C	JNZ	1DCD
4E08:1D61 B8C00C	MOV	BX,0CC0
4E08:1D64 26	ES:	
4E08:1D65 8B3E0900	MOV	DI,[0009]
4E08:1D69 B90B00	MOV	CX,000B
4E08:1D6C 33F6	XOR	SI,SI
4E08:1D6E E84D00	CALL	1DBE
4E08:1D71 E8D700	CALL	1E4B
4E08:1D74 E2F6	LOOP	1D6C
4E08:1D76 26	ES:	
4E08:1D77 F606100020	TEST	BYTE PTR [0010],20
-u		
4E08:1D7C 754F	JNZ	1DCD
4E08:1D7E B90B00	MOV	CX,000B
4E08:1D81 33F6	XOR	SI,SI
4E08:1D83 46	INC	SI
4E08:1D84 E83700	CALL	1DBE
4E08:1D87 E8C100	CALL	1E4B
4E08:1D8A E2F5	LOOP	1D81
4E08:1D8C 26	ES:	
4E08:1D8D F606100020	TEST	BYTE PTR [0010],20
4E08:1D92 7539	JNZ	1DCD
4E08:1D94 BB2013	MOV	BX,1320
4E08:1D97 26	ES:	
4E08:1D98 8B3E0900	MOV	DI,[0009]
-u		
4E08:1D9C B90B00	MOV	CX,000B
4E08:1D9F 33F6	XOR	SI,SI
4E08:1DA1 46	INC	SI
4E08:1DA2 46	INC	SI
4E08:1DA3 E81600	CALL	1DBE
4E08:1DA6 E8A200	CALL	1E4B
4E08:1DA9 E2F4	LOOP	1D9F
4E08:1DAB B90B00	MOV	CX,000B
4E08:1DAE 33F6	XOR	SI,SI
4E08:1DB0 46	INC	SI
4E08:1DB1 46	INC	SI
4E08:1DB2 46	INC	SI
4E08:1DB3 E80B00	CALL	1DBE
4E08:1DB6 E89200	CALL	1E4B
4E08:1DB9 E2F3	LOOP	1DAE
4E08:1DBB EB45	JMP	1E02
-u		
4E08:1DBD 90	NOF	
4E08:1DBE B60B	MOV	DH,0B
4E08:1DC0 AC	LDSB	
4E08:1DC1 D2D8	RCR	AL,CL

20100

4E08:1DC3 D0D4	RCL	AH,1
4E08:1DC5 46	INC	SI
4E08:1DC6 46	INC	SI
4E08:1DC7 46	INC	SI
4E08:1DC8 FECE	DEC	DH
4E08:1DCA 75F4	JNZ	1DC0
4E08:1DCC C3	RET	
4E08:1DCD 26	ES:	
4E08:1DCE F606100080	TEST	BYTE PTR [0010],80
4E08:1DD3 755B	JNZ	1E30
4E08:1DD5 88C00C	MOV	BX,0CC0
4E08:1DD8 26	ES:	
4E08:1DD9 8B3E0900	MOV	DI,[0009]
-u		
4E08:1DD0 B91000	MOV	CX,0010
4E08:1DE0 33F6	XOR	SI,SI
4E08:1DE2 46	INC	SI
4E08:1DE3 32E4	XOR	AH,AH
4E08:1DE5 E85600	CALL	1E3E
4E08:1DE8 EB6000	CALL	1E4B
4E08:1DEB E2F8	LOOP	1DE5
4E08:1DED BB2013	MOV	BX,1320
4E08:1DF0 26	ES:	
4E08:1DF1 8B3E0900	MOV	DI,[0009]
4E08:1DF5 B91000	MOV	CX,0010
4E08:1DF8 33F6	XOR	SI,SI
4E08:1DFA EB4100	CALL	1E3E
-u		
4E08:1DFD EB4B00	CALL	1E4B
4E08:1E00 E2F8	LOOP	1DFA
4E08:1E02 8BC7	MOV	AX,DI
4E08:1E04 26	ES:	
4E08:1E05 8B1E0900	MOV	BX,[0009]
4E08:1E09 26	ES:	
4E08:1E0A B93E0900	MOV	[0009],DI
4E08:1E0E 2BC3	SUB	AX,BX
4E08:1E10 EB1E00	CALL	1E31
4E08:1E13 B24C	MOV	DL,4C
4E08:1E15 26	ES:	
4E08:1E16 F6060F0002	TEST	BYTE PTR [000F],02
4E08:1E18 7504	JNZ	1E21
-u		
4E08:1E1D D1E0	SHL	AX,1
4E08:1E1F B24B	MOV	DL,4B
4E08:1E21 26	ES:	
4E08:1E22 01060D00	ADD	[000D],AX
4E08:1E26 D1E9	SHR	CX,1
4E08:1E28 D1E9	SHR	CX,1

4E08:1E2A D1E9	SHR	CX,1	
4E08:1E2C 8AC2	MOV	AL,DL	
4E08:1E2E F3	REPZ		
4E08:1E2F AA	STOSB		
4E08:1E30 C3	RET		
4E08:1E31 B90306	MOV	CX,0003	
4E08:1E34 D3EB	SHR	BX,CL	
4E08:1E36 83C317	ADD	BX,+17	
4E08:1E39 8BFB	MOV	DI,BX	
4E08:1E3B 8BC8	MOV	CX,AX	
-u			
4E08:1E3D C3	RET		
4E08:1E3E 51	PUSH	CX	
4E08:1E3F B90B00	MOV	CX,000B	
4E08:1E42 AD	LODSW		
4E08:1E43 D0D8	RCR	AL,1	
4E08:1E45 D0D4	RCL	AH,1	
4E08:1E47 E2FA	LOOP	1E43	
4E08:1E49 59	POP	CX	
4E08:1E4A C3	RET		
4E08:1E4B 52	PUSH	DX	: 送打印数据模块
4E08:1E4C 1E	PUSH	DS	
4E08:1E4D 06	PUSH	ES	
4E08:1E4E 1F	POP	DS	
4E08:1E4F F6060F0020	TEST	BYTE PTR [000F],20	: 纵扩?
4E08:1E54 7514	JNZ	1E6A	
4E08:1E56 8BA1E300	MOV	[BX+DI+00E3],AH	
4E08:1E5A 47	INC	DI	
4E08:1E5B F6060F0010	TEST	BYTE PTR [000F],10	: 横扩?
-u			
4E08:1E60 7405	JZ	1E67	
4E08:1E62 8BA1E300	MOV	[BX+DI+00E3],AH	
4E08:1E66 47	INC	DI	
4E08:1E67 1F	POP	DS	
4E08:1E68 5A	POP	DX	
4E08:1E69 C3	RET		
4E08:1E6A 51	PUSH	CX	
4E08:1E6B 53	PUSH	BX	
4E08:1E6C B90B00	MOV	CX,000B	
4E08:1E6F D0D4	RCL	AH,1	
4E08:1E71 D1D2	RCL	DX,1	
4E08:1E73 8BDA	MOV	BX,DX	
4E08:1E75 D1DB	RCR	BX,1	
4E08:1E77 D1D2	RCL	DX,1	
4E08:1E79 E2F4	LOOP	1E6F	
4E08:1E7B 5B	POP	BX	
4E08:1E7C 53	PUSH	BX	
4E08:1E7D 81EBC00C	SUB	BX,0CC0	

```

-u
4E08:1EB1 D1C3      ROL    BX,1
4E08:1EB3 88B1E300  MOV    [BX+DI+00E3],DH
4E08:1EB7 F6D60F0010 TEST   BYTE PTR [000F],10
4E08:1EB8 7404      JZ     1E92
4E08:1EBE 88B1E400  MOV    [BX+DI+00E4],DH
4E08:1E92 88914307  MOV    [BX+DI+0743],DL
4E08:1E96 47          INC    DI
4E08:1E97 F6D60F0010 TEST   BYTE PTR [000F],10
4E08:1E9C 7405      JZ     1EA3
4E08:1E9E 88914307  MOV    [BX+DI+0743],DL

```

```

-u
4E08:1EA2 47          INC    DI
4E08:1EA3 5B          POP    BX
4E08:1EA4 59          POP    CX
4E08:1EA5 1F          POP    DS
4E08:1EA6 5A          POP    DX
4E08:1EA7 C3          RET
4E08:1EA8 26          ES:
4E08:1EA9 8B3E0900  MOV    DI,[0009]
4E08:1EAD BB2013  MOV    BX,1320
4E08:1EB0 26          ES:
4E08:1EB1 F6D6100020 TEST   BYTE PTR [0010],20
4E08:1EB6 7518      JNZ    1ED0
4E08:1EB8 B90800  MOV    CX,0008
4E08:1EBB 56          PUSH   SI
4E08:1EBC B608      MOV    DH,08
4E08:1EBE AC      LODSB
4E08:1EBF D208      RCR    AL,CL
4E08:1EC1 D0D4      RCL    AH,1

```

```

-u
4E08:1EC3 FECE      DEC    DH
4E08:1EC5 75F7      JNZ    1EBE
4E08:1EC7 5E          POP    SI
4E08:1EC8 E880FF  CALL  1E4B
4E08:1ECB E2EE      LODP  1EBB
4E08:1ECD E932FF  JMP    1E02
4E08:1ED0 B608      MOV    DH,08
4E08:1ED2 B90800  MOV    CX,0008
4E08:1ED5 AC      LODSB
4E08:1ED6 D0D8      RCR    AL,1
4E08:1EDB D0D4      RCL    AH,1
4E08:1EDA E2FA      LODP  1ED6
4E08:1EDC E86CFF  CALL  1E4B
4E08:1EDF FECE      DEC    DH
4E08:1EE1 75EF      JNZ    1ED2

```

```

-u
4E08:1EE3 E91CFF  JMP    1E02
4E08:1EE6 06          PUSH   ES

```

4E08:1EE7 1F	POP	DS	
4E08:1EE8 833E090000	CMP	WORD PTR [0009],+00	: 输出缓冲区内容模块入口
4E08:1EED 7433	JZ	1F22	
4E08:1EEF F6060F0010	TEST	BYTE PTR [000F],10	: 横扩?
4E08:1EF4 740C	JZ	1F02	
4E08:1EF6 BA0000	MOV	DX,0000	
4E08:1EF9 E81500	CALL	1F11	
4E08:1EFC BA0100	MOV	DX,0001	
4E08:1EFF E80F00	CALL	1F11	
4E08:1F02 BA0200	MOV	DX,0002	
-u			
4E08:1F05 EB0900	CALL	1F11	
4E08:1F08 BA0300	MOV	DX,0003	
4E08:1F0B EB0300	CALL	1F11	
4E08:1F0E EB12	JMP	1F22	
4E08:1F10 90	NOP		
4E08:1F11 B86006	MOV	AX,0660	
4E08:1F14 F7E2	MUL	DX	
4E08:1F16 05E300	ADD	AX,00E3	
4E08:1F19 8BD8	MOV	BX,AX	
4E08:1F1B E83700	CALL	1F55	
4E08:1F1E E8A000	CALL	1FC1	
4E08:1F21 C3	RET		
4E08:1F22 C70609000000	MOV	WORD PTR [0009],0000	: 缓冲区计数器清 0
-u			
4E08:1F2B B01B	MOV	AL,1B	
4E08:1F2A E8A8FB	CALL	1AD5	
4E08:1F2D B032	MOV	AL,32	
4E08:1F2F E8A3FB	CALL	1AD5	
4E08:1F32 B00A	MOV	AL,0A	
4E08:1F34 E89EFB	CALL	1AD5	
4E08:1F37 B80000	MOV	AX,0000	
4E08:1F3A B9C00C	MOV	CX,00CC	
4E08:1F3D BF E300	MOV	DI,00E3	
4E08:1F40 F3	REPZ		
4E08:1F41 AB	STOSW		
4E08:1F42 C7060D000000	MOV	WORD PTR [000D],0000	
-u			
4E08:1F48 F6060F0020	TEST	BYTE PTR [000F],20	: 纵扩?
4E08:1F4D 7505	JNZ	1F54	
4E08:1F4F B0260F00FE	AND	BYTE PTR [000F],FE	
4E08:1F54 C3	RET		
4E08:1F55 53	PUSH	BX	
4E08:1F56 33DB	XOR	BX,BX	
4E08:1F58 53	PUSH	BX	
4E08:1F59 E8D5FE	CALL	1E31	
4E08:1F5C 33C9	XOR	CX,CX	
4E08:1F5E BA05	MOV	AL,[DI]	

4E08:1F60 3A05	CMP	AL,[DI]
4E08:1F62 7504	JNZ	1F68
4E08:1F64 41	INC	CX
4E08:1F65 47	INC	DI
4E08:1F66 EBF8	JMP	1F60
-		
-uu		
^ Error		
-u		
4E08:1F68 D1E1	SHL	CX,1
4E08:1F6A D1E1	SHL	CX,1
4E08:1F6C D1E1	SHL	CX,1
4E08:1F6E 5B	POP	BX
4E08:1F6F 8BF3	MOV	SI,BX
4E08:1F71 5F	POP	DI
4E08:1F72 8B160900	MOV	DX,[0009]
4E08:1F76 2B03	SUB	DX,BX
4E08:1F78 3BCA	CMP	CX,DX
4E08:1F7A 7202	JB	1F7E
4E08:1F7C 8BCA	MOV	CX,DX
4E08:1F7E 03D9	ADD	BX,CX
4E08:1F80 53	PUSH	BX
4E08:1F81 8BDF	MOV	BX,DI
4E08:1F83 57	PUSH	DI
4E08:1F84 E80D00	CALL	1F94
4E08:1F87 58	POP	AX
-u		
4E08:1F88 5B	POP	BX
4E08:1F89 3B1E0900	CMP	BX,[0009]
4E08:1F8D 7201	JB	1F90
4E08:1F8F C3	RET	
4E08:1F90 50	PUSH	AX
4E08:1F91 53	PUSH	BX
4E08:1F92 EBC5	JMP	1F59
4E08:1F94 56	PUSH	SI
4E08:1F95 53	PUSH	BX
4E08:1F96 03F3	ADD	SI,BX
4E08:1F98 51	PUSH	CX
4E08:1F99 E80C00	CALL	1FAB
4E08:1F9C 59	POP	CX
4E08:1F9D 51	PUSH	CX
4E08:1F9E AC	LDSB	
4E08:1F9F E833FB	CALL	1AD5
4E08:1FA2 59	POP	CX
4E08:1FA3 E2FB	LOOP	1F9D
4E08:1FA5 5B	POP	BX
4E08:1FA6 5E	POP	SI
4E08:1FA7 C3	RET	
-u		

4E08:1FA8 51	PUSH	CX	
4E08:1FA9 50	PUSH	AX	
4E08:1FAA B01B	MOV	AL,1B	
4E08:1FAC E826FB	CALL	1AD5	
4E08:1FAF 58	POP	AX	
4E08:1FB0 E822FB	CALL	1AD5	
4E08:1FB3 59	POP	CX	
4E08:1FB4 51	PUSH	CX	
4E08:1FB5 BAC1	MOV	AL,CL	
4E08:1FB7 E81BFB	CALL	1AD5	
4E08:1FBA 59	POP	CX	
4E08:1FBB BAC5	MOV	AL,CH	
4E08:1FBD E815FB	CALL	1AD5	
4E08:1FC0 C3	RET		
4E08:1FC1 F60611007F	TEST	BYTE PTR [0011],7F	
4E08:1FC6 7411	JZ	1FD9	
-u			
4E08:1FC8 FE0E1100	DEC	BYTE PTR [0011]	
4E08:1FCC 7403	JZ	1FD1	
4E08:1FCE EB10	JMP	1FE0	
4E08:1FD0 90	NOP		
4E08:1FD1 C606110003	MOV	BYTE PTR [0011],03	
4E08:1FD6 EB1D	JMP	1FF5	
4E08:1FDB 90	NOP		
4E08:1FD9 F606100020	TEST	BYTE PTR [0010],20	
4E08:1FDE 7515	JNZ	1FF5	
4E08:1FE0 B01B	MOV	AL,1B	
4E08:1FE2 EBF0FA	CALL	1AD5	
4E08:1FE5 B033	MOV	AL,33	
4E08:1FE7 E8EBFA	CALL	1AD5	
-u			
4E08:1FEA B001	MOV	AL,01	
4E08:1FEC E8E6FA	CALL	1AD5	
4E08:1FEF B00A	MOV	AL,0A	
4E08:1FF1 E8E1FA	CALL	1AD5	
4E08:1FF4 C3	RET		
4E08:1FF5 B01B	MOV	AL,1B	
4E08:1FF7 E8DBFA	CALL	1AD5	
4E08:1FFA B041	MOV	AL,41	
4E08:1FFC E8D6FA	CALL	1AD5	
4E08:1FFF B008	MOV	AL,08	
4E08:2001 E8D1FA	CALL	1AD5	
4E08:2004 B00A	MOV	AL,0A	
4E08:2006 E8CCFA	CALL	1AD5	
4E08:2009 C3	RET		
-u			
4E08:200A B308	MOV	BL,08	: INT 17H 3号功能入口
4E08:200C F6E3	MUL	BL	

```

4E08:200E 3D6106    CMP     AX,0661
4E08:2011 7203      JB      2016
4E08:2013 886006    MOV     AX,0660
4E08:2016 A30B00    MOV     [000B],AX
4E08:2019 E992FA    JMP     1AAE
4E08:201C FB          STI
4E08:201D 1E          PUSH    DS
4E08:201E 50          PUSH    AX
4E08:201F 52          PUSH    DX
4E08:2020 56          PUSH    SI
4E08:2021 BE3100    MOV     SI,0031
4E08:2024 884000    MOV     AX,0040
4E08:2027 BED8      MOV     DS,AX
4E08:2029 8A04      MOV     AL,[SI]
-u
4E08:202B 0E          PUSH    CS
4E08:202C 1F          POP     DS
4E08:202D 33D2      XOR     DX,DX
4E08:202F A20000    MOV     [0000],AL
4E08:2032 803E000004  CMP     BYTE PTR [0000],04
4E08:2037 7217      JB      2050
4E08:2039 803E000007  CMP     BYTE PTR [0000],07
4E08:203E 7410      JZ      2050
4E08:2040 8B1B00    MOV     AX,001B
4E08:2043 CD17      INT     17
4E08:2045 B057      MOV     AL,57
4E08:2047 32E4      XOR     AH,AH
4E08:2049 CD17      INT     17
-u
4E08:204B 5E          POP     SI
4E08:204C 5A          POP     DX
4E08:204D 5B          POP     AX
4E08:204E 1F          POP     DS
4E08:204F CF          IRET
4E08:2050 5E          POP     SI
4E08:2051 5A          POP     DX
4E08:2052 5B          POP     AX
4E08:2053 1F          POP     DS
4E08:2054 EA54FF00F0  JMP     F000:FF54
4E08:2059 0000      ADD     [BX+SI],AL
4E08:205B 81E27F7F  AND     DX,7F7F
4E08:205F 50          PUSH    AX
4E08:2060 80FE30    CMP     DH,30
4E08:2063 7203      JB      2068
4E08:2065 80EE08    SUB     DH,08
4E08:2068 80EE21    SUB     DH,21

```

: 0660→行宽定义单元 (013)

: INT 05H 新向量入口

-uu
^ Error

-u

```
4E08:206B B05E      MOV     AL,5E
4E08:206D F6E6      MUL     DH
4E08:206F 80EA21     SUB     DL,21
4E08:2072 32F6      XOR     DH,DH
4E08:2074 03C2      ADD     AX,DX
4E08:2076 D1E0      SHL     AX,1
4E08:2078 2E          CS:
4E08:2079 03065920    ADD     AX,[2059]
4E08:207D 8BD0      MOV     DX,AX
4E08:207F 5B          POP     AX
4E08:2080 C3          RET
4E08:2081 0E          PUSH   CS
4E08:2082 1F          POP     DS
4E08:2083 B81F35     MOV     AX,351F
4E08:2086 CD21      INT     21
4E08:2088 B91E0300    MOV     [0003],BX
```

: 9 P.exe 驱动程序本身执行指令入口

-u

```
4E08:208C BCC3      MOV     BX,ES
4E08:208E 891E0100    MOV     [0001],BX
4E08:2092 B81035     MOV     AX,3510
4E08:2095 CD21      INT     21
4E08:2097 BB7500     MOV     BX,0075
4E08:209A 26          ES:
4E08:209B 8B07      MOV     AX,[BX]
4E08:209D 2E          CS:
4E08:209E A35920     MOV     [2059],AX
4E08:20A1 8D16631A   LEA     DX,[1A63]
4E08:20A5 B81725     MOV     AX,2517
4E08:20A8 CD21      INT     21
4E08:20AA 8D161C20   LEA     DX,[201C]
```

: ASCII 字符库起址→CS : 0001

: 双字节CCLIB字模库段址在

: ES : (BX) →CS : (2059)

-u

```
4E08:20AE B80525     MOV     AX,2505
4E08:20B1 CD21      INT     21
4E08:20B3 8D168120   LEA     DX,[2081]
4E08:20B7 81C20301   ADD     DX,0103
4E08:20BB CD27      INT     27
4E08:20BD 0000      ADD     [BX+SI],AL
```


-r

AX=0000 BX=0000 CX=0C00 DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=4DF8 ES=4DF8 SS=4E08 CS=4E08 IP=0B5A NV UP DI PL NZ NA PO NC
4E08:0B5A 0E PUSH CS

-dcs:00

4E08:0000 00 00 7E 00 00 00 00 00-00 00 00 00 00 00 00 CS:00汉字模库段址
4E08:0010 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 03第1字节汉字内码标志
4E08:0020 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 04第1字节汉字内码标志
4E08:0030 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 05第2字节汉字内码标志
4E08:0040 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 09打印机口地址
4E08:0050 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 0B打印机状态
4E08:0060 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 0D打印数据区,
4E08:0070 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00

-d cs:1ed

4E08:01ED 00 00 00 ...
4E08:01F0 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 01ED打印数据属性
4E08:0200 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00
4E08:0210 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00
4E08:0220 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00
4E08:0230 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00
4E08:0240 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00
4E08:0250 00 00 00 00 00 00 00 00-00 00 00 00 00-00 00 00
4E08:0260 00 00 00 00 00 00 00 00-00 00 00 00 00 00

-d cs:3cd

4E08:03CD 02 00 7E
4E08:03D0 09 7E 09 00 00 00 00 00-00 00 00 00 00 00 00
4E08:03E0 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00
4E08:03F0 00 00 00 00 00 00 00 00-00 00 B2 12 F6 E2 3D 7E2.vb=^
4E08:0400 09 72 03 B8 7E 09 26 A3-D1 03 EB 37 90 FB 1E 52 .r.B^.&00.k7.(.R
4E08:0410 56 51 53 06 57 0E 1F BE-40 00 BE C6 8B F2 D1 E6 VQS.W..>@..F.rBf
4E08:0420 26 8B 54 08 1E 07 89 16-09 00 C6 06 0C 00 00 0B &.T.....F.....
4E08:0430 D2 74 10 0A E4 74 23 FE-CC 74 38 FE CC 74 10 FE Rt..dt#*Lt8*Lt.^
4E08:0440 CC 74 B7 26 A1 0B 00 5F-07 5B 59 5E 5A Lt7k!...[Y*Z

-ucs:3fa

4E08:03FA B212 MOV DL,12 ; 18->DL
4E08:03FC F6E2 MUL DL ; 行宽字符数换成打印机针列数
4E08:03FE 3D7E09 CMP AX,097E ; 行宽列数最大值?
4E08:0401 7203 JB 0406
4E08:0403 B87E09 MOV AX,097E
4E08:0406 26 ES:
4E08:0407 A3D103 MOV [03D1],AX ; 行宽最大值存ES : 03D1
4E08:040A EB37 JMP 0443
4E08:040C 90 NOP
4E08:040D FB STI ; 新INT 17H
4E08:040E 1E PUSH DS
4E08:040F 52 PUSH DX
4E08:0410 56 PUSH SI

4E08:0411 51	PUSH	CX	
4E08:0412 53	PUSH	BX	
4E08:0413 06	PUSH	ES	
4E08:0414 57	PUSH	DI	
4E08:0415 0E	PUSH	CS	; DS 改为CS 值
4E08:0416 1F	POP	DS	
4E08:0417 BE4000	MOV	SI,0040	; DOS 数据段值40
-u			
4E08:041A BEC6	MOV	ES,SI	; ES =40
4E08:041C 8BF2	MOV	SI,DX	; 打印机编号采2
4E08:041E D1E6	SHL	SI,1	
4E08:0420 26	ES:		
4E08:0421 8B540B	MOV	DX,[SI+0B]	; 取打印机口地址
4E08:0424 1E	PUSH	DS	
4E08:0425 07	POP	ES	; ES 取DS 值
4E08:0426 89160900	MOV	[0009],DX	; 打印机口地址→(0009)
4E08:042A C6060C0000	MOV	BYTE PTR [000C],00	; 清打印机状态字
4E08:042F 0BD2	OR	DX,DX	
4E08:0431 7410	JZ	0443	; 打印机编号=0, 中断返回
4E08:0433 0AE4	OR	AH,AH	; 测试功能调用
4E08:0435 7423	JZ	045A	; AH = 0 转045 A 0号功能
4E08:0437 FECC	DEC	AH	
4E08:0439 743B	JZ	0473	; AH = 1 转0473 1号功能
-u			
4E08:043B FECC	DEC	AH	
4E08:043D 7410	JZ	044F	; AH = 2 转044 F 2号功能
4E08:043F FECC	DEC	AH	
4E08:0441 74B7	JZ	03FA	; AH = 3 转03F A 3号功能
4E08:0443 26	ES:		
4E08:0444 A10B00	MOV	AX,[000B]	; 打印机状态字→AX
4E08:0447 5F	POP	DI	; 恢复现场
4E08:0448 07	POP	ES	
4E08:0449 5B	POP	BX	
4E08:044A 59	POP	CX	
4E08:044B 5E	POP	SI	
4E08:044C 5A	POP	DX	
4E08:044D 1F	POP	DS	
4E08:044E CF	IRET		
4E08:044F B402	MOV	AH,02	
4E08:0451 E81000	CALL	0464	; AL 字符送打印子程序
4E08:0454 26	ES:		
4E08:0455 A30B00	MOV	[000B],AX	; 存状态字
4E08:045B EBE9	JMP	0443	
4E08:045A EB3B	JMP	0494	; 转0号功能
-u			
4E08:045C 90	NOP		
4E08:045D E80200	CALL	0462	

4E08:0460 EBE1	JMP	0443	
4E08:0462 32E4	XOR	AH,AH	
4E08:0464 52	PUSH	DX	
4E08:0465 33D2	XOR	DX,DX	
4E08:0467 9C	PUSHF		: 状态标志入栈
4E08:0468 9AD2EF00F0	CALL	F000:EFD2	: ROM INT 17H入口(原装机)
4E08:046D 2E	CS:		
4E08:046E A30B00	MOV	[000B],AX	: 存状态字→(000B)
4E08:0471 5A	POP	DX	
4E08:0472 C3	RET		
4E08:0473 B401	MOV	AH,01	
4E08:0475 EBECFF	CALL	0464	
4E08:047B B01B	MOV	AL,1B	: esc
4E08:047A EB55FF	CALL	0462	
-u			
4E08:047D B04B	MOV	AL,4B	: "H"
4E08:047F EBE0FF	CALL	0462	
4E08:0482 B83000	MOV	AX,0030	: "0"
4E08:0485 E8DCFF	CALL	0464	
4E08:048B B030	MOV	AL,30	: "0"
4E08:048A E8D5FF	CALL	0462	
4E08:048D B035	MOV	AL,35	: "5"
4E08:048F EB00FF	CALL	0462	
4E08:0492 E6AF	JMP	0443	
4E08:0494 C606D60306	MOV	BYTE PTR [03D6],00	: 0号功能 AL→打印机子程序清汉字内码标志
4E08:0499 F606030010	TEST	BYTE PTR [0003],10	: 收到第一字节汉字?
-u			
4E08:049E 7403	JZ	04A3	
4E08:04A0 E9D800	JMP	057B	: 是,当前第二字节内码转057 B
4E08:04A3 A8B0	TEST	AL,80	: 没有,当前是第一字节内码?
4E08:04A5 7403	JZ	04AA	
4E08:04A7 E9BB00	JMP	0565	: 是,转0565执行
4E08:04AA 3C0E	CMP	AL,0E	
4E08:04AC 7508	JNZ	04B6	: SO? 是没打印控制横扩位
4E08:04AE 806ECD0310	OR	BYTE PTR [03CD],10	
4E08:04B3 E9A400	JMP	055A	转055 A执行
4E08:04B6 3C0F	CMP	AL,0F	: SI? 取消横扩
4E08:04B8 7508	JNZ	04C2	
4E08:04BA 8026CD03EF	AND	BYTE PTR [03CD],EF	: 控制横扩位清0
-u			
4E08:04BF E99800	JMP	055A	
4E08:04C2 3C1B	CMP	AL,1B	: ESC 码?
4E08:04C4 7508	JNZ	04CE	
4E08:04C6 C606D7030F	MOV	BYTE PTR [03D7],0F	: 是,设ESC 标志
4E08:04CB E975FF	JMP	0443	
4E08:04CE 803ED7030F	CMP	BYTE PTR [03D7],0F	: 收到ESC?
4E08:04D3 7528	JNZ	04FD	
4E08:04D5 3C49	CMP	AL,49	: 收到I? 是转04F 5

4E08:04D7 741C	JZ	04F5	
4E08:04D9 50	PUSH	AX	: 不是保存数据
4E08:04DA B01B	MOV	AL,1B	: ESC →AL
4E08:04DC C606D603FF	MOV	BYTE PTR [03D6],FF	: 非汉字内码标志→03D6
-u			
4E08:04E1 EB1701	CALL	05FB	: 打印数据存打印区
4E08:04E4 5B	POP	AX	: 恢复当前打印数据
4E08:04E5 C606D603FF	MOV	BYTE PTR [03D6],FF	
4E08:04EA E80E01	CALL	05FB	
4E08:04ED C606D70300	MOV	BYTE PTR [03D7],00	: 清除esc 标志
4E08:04F2 E94EFF	JMP	0443	
4E08:04F5 B00ED703F0	OR	BYTE PTR [03D7],F0	: 设ESC + "I" 命令标志
4E08:04FA E946FF	JMP	0443	
4E08:04FD B03ED703FF	CMP	BYTE PTR [03D7],FF	: ESC + "I" 命令收到?
-u			
4E08:0502 753B	JNZ	053C	: 没有, 转053 C
4E08:0504 3C41	CMP	AL,41	: A 号字
4E08:0506 750D	JNZ	0515	
4E08:0508 B026CD03FB	AND	BYTE PTR [03CD],FB	: 设A 号字标志
4E08:050D B00ECD0301	OR	BYTE PTR [03CD],01	
4E08:0512 EB20	JMP	0534	
4E08:0514 90	NGP		
4E08:0515 3C42	CMP	AL,42	: B 号字
4E08:0517 750D	JNZ	052B	
4E08:0519 B026CD03FB	AND	BYTE PTR [03CD],FB	: 设B 号字标志
4E08:051E B00ECD0302	OR	BYTE PTR [03CD],02	
-u			
4E08:0523 EB0F	JMP	0534	
4E08:0525 90	NGP		
4E08:0526 3C43	CMP	AL,43	: C 号字
4E08:052B 750A	JNZ	0534	
4E08:052A B026CD03FB	AND	BYTE PTR [03CD],FB	: 设C 号字标志
4E08:052F B00ECD0324	OR	BYTE PTR [03CD],24	
4E08:0534 C606D70300	MOV	BYTE PTR [03D7],00	
4E08:0539 E907FF	JMP	0443	
4E08:053C 3C0D	CMP	AL,0D	: 回车符?
4E08:053E 7513	JNZ	0553	
4E08:0540 B03ECE0300	CMP	BYTE PTR [03CE],00	: 打印数据区有数据?
-u			
4E08:0545 7506	JNZ	054D	
4E08:0547 E818FF	CALL	0462	: 没有, 回车送打印
4E08:054A E9F6FE	JMP	0443	
4E08:054D E8AB00	CALL	05FB	: 有数据, 调打印子程序
4E08:0550 E9F0FE	JMP	0443	
4E08:0553 3C0A	CMP	AL,0A	: 换行符?
4E08:0555 7503	JNZ	055A	
4E08:0557 E9E9FE	JMP	0443	
4E08:055A C606D603FF	MOV	BYTE PTR [03D6],FF	

4E08:055F E83500	CALL	0597	: 非汉字标志FF → (03D6)
4E08:0562 E9DEFE	JMP	0443	
-u			
4E08:0565 3CA1	CMP	AL,A1	: 第一字节汉字内码? 是转056 E
4E08:0567 7705	JA	056E	
4E08:0569 247F	AND	AL,7F	: 不是, 高位清0, 回494, 0号入口
4E08:056B E926FF	JMP	0494	
4E08:056E C606030010	MOV	BYTE PTR [0003],10	: 是, 设第一字节汉字内码标志
4E08:0573 BB0400	MOV	BX,0004	: 第一字节汉字内码地址→BX
4E08:0576 8B07	MOV	[BX],AL	: 存第一字节汉字内码
4E08:0578 E9C8FE	JMP	0443	
4E08:057B C606030000	MOV	BYTE PTR [0003],00	: 清第一字节汉字内码标志
4E08:0580 BB0400	MOV	BX,0004	
4E08:0583 8B4701	MOV	[BX+01],AL	: 存第二字节汉字内码
-u			
4E08:0586 E80600	CALL	058F	: 调子程序AX 返回完整汉字内码
4E08:0589 E80B00	CALL	0597	: 完整汉字内码存打印数据区
4E08:058C E9B4FE	JMP	0443	
4E08:058F 8A260400	MOV	AH,[0004]	: 第一字节汉字内码→AH
4E08:0593 800500	MOV	AL,[0005]	: 第二字节汉字内码→AL
4E08:0596 C3	RET		
4E08:0597 B03ED603FF	CMP	BYTE PTR [03D6],FF	: 要存数据是汉字内码?
4E08:059C 7507	JNZ	05A7	
4E08:059E E84600	CALL	05E7	
4E08:05A1 E8E802	CALL	0B8C	: 不是, 一字节打印数据存打印区
4E08:05A4 EB29	JMP	05CF	: 修改列数计数器
-u			
4E08:05A6 90	NOP		
4E08:05A7 0DB0B0	OR	AX,B0B0	
4E08:05AA 8A1ECE03	MOV	BL,[03CE]	: 取写打印数据区指针
4E08:05AE B700	MOV	BH,00	
4E08:05B0 8BA70D00	MOV	[BX+000D],AH	: 2字节打印数据指针
4E08:05B4 8B870E00	MOV	[BX+000E],AL	
4E08:05B8 8A0ECD03	MOV	CL,[03CD]	: 取控制标志→CL
4E08:05BC 8B8FED01	MOV	[BX+01ED],CL	: 存打印数据属性区
4E08:05C0 8B8FEE01	MOV	[BX+01EE],CL	
4E08:05C4 FE06CE03	INC	BYTE PTR [03CE]	: 写打印数据指针+2
-u			
4E08:05C8 FE06CE03	INC	BYTE PTR [03CE]	
4E08:05CC E8BD02	CALL	0B8C	: 修改列数计数器
4E08:05CF 50	PUSH	AX	: 存AX
4E08:05D0 26	ES:		
4E08:05D1 A1D303	MOV	AX,[03D3]	: 取列计数器值
4E08:05D4 3B06D103	CMP	AX,[03D1]	: 列数超过?
4E08:05D8 5B	POP	AX	: 恢复AX
4E08:05D9 720B	JB	05E6	
4E08:05DB C606D803FF	MOV	BYTE PTR [03DB],FF	: 超过, 没超出标志
4E08:05E0 E81800	CALL	05FB	: 打印数据送打印机

4E08:05E3 EB7402	CALL	085A	: 初始化数据区各标志
4E08:05E6 C3	RET		
4E08:05E7 B700	MOV	BH,00	: 单字节(ASCII) 存在打印区
-u			
4E08:05E9 8A1ECE03	MOV	BL,[03CE]	: 存当前打印数据指针
4E08:05ED 88870D00	MOV	[BX+000D],AL	
4E08:05F1 C687ED0100	MOV	BYTE PTR [BX+01ED],00	: 清打印数据属性
4E08:05F6 FE06CE03	INC	BYTE PTR [03CE]	: 打印数据指针 + 1
4E08:05FA C3	RET		
4E08:05FB F606CD0320	TEST	BYTE PTR [03CD],20	: 数据送打印机子程序
4E08:0600 7407	JZ	0609	: 当前是C号字?
4E08:0602 E85300	CALL	065B	
4E08:0605 E85202	CALL	085A	: C号字, 纵扩方式打印
4E08:0608 C3	RET		: 初始化各标志单元
-u			
4E08:0609 C606D50300	MOV	BYTE PTR [03D5],00	: 清读打印数据指针
4E08:060E 8A1ED503	MOV	BL,[03D5]	: 读打印数据指针
4E08:0612 B700	MOV	BH,00	
4E08:0614 8BF3	MOV	SI,BX	: 送SI
4E08:0616 B8B40D00	MOV	AL,[SI+000D]	: 取数据→AL
4E08:061A 8AA4ED01	MOV	AH,[SI+01ED]	: 取属性→AH
4E08:061E B0FC00	CMP	AH,00	: 汉字内码? 是转0629
4E08:0621 7506	JNZ	0629	
4E08:0623 E85E02	CALL	088A	: 不是, AL送打印并调整指针
4E08:0626 EB16	JMP	063E	
4E08:0628 90	NOP		
-u			
4E08:0629 8826CD03	MOV	[03CD],AH	: 属性送控制标志
4E08:062D 8AA40E00	MOV	AH,[SI+000E]	: 第二节汉字内码→AH
4E08:0631 B6E0	XCHG	AL,AH	: 调整二字节内码位置
4E08:0633 E88202	CALL	08BB	: 取内码字模送打印机
4E08:0636 FE06D503	INC	BYTE PTR [03D5]	
4E08:063A FE06D503	INC	BYTE PTR [03D5]	: 读打印数据指针 + 2
4E08:063E 8A1ECE03	MOV	BL,[03CE]	: 取写打印数据指针
4E08:0642 381ED503	CMP	[03D5],BL	
4E08:0646 7402	JZ	064A	: 打印数据送完?
4E08:0648 EBC4	JMP	060E	: 未完继续
-u			
4E08:064A B00D	MOV	AL,0D	: 送完, 回车换行
4E08:064C E813FE	CALL	0462	
4E08:064F B00A	MOV	AL,0A	
4E08:0651 E80EFE	CALL	0462	
4E08:0654 E80302	CALL	085A	
4E08:0657 C3	RET		
4E08:0658 C606D50300	MOV	BYTE PTR [03D5],00	: C号字(纵扩)送打印子程序
4E08:065D 8A1ED503	MOV	BL,[03D5]	: 清读打印数据指针
4E08:0661 B700	MOV	BH,00	
4E08:0663 8BF3	MOV	SI,BX	: 读打印数据指针→SI

4E08:0665	BAA40D00	MOV	AH,[SI+000D]	: 打印数据→AH
4E08:0669	BAB4ED01	MOV	AL,[SI+01ED]	属性→AL
-u				
4E08:066B	A804	TEST	AL,04	: 汉字内码? 不是转 06DC
4E08:066F	746B	JZ	06DC	
4E08:0671	BAB40E00	MOV	AL,[SI+000E]	: 取第二字节内码→AL
4E08:0675	257F7F	AND	AX,7F7F	
4E08:0678	8BD0	MOV	DX,AX	: 2字节内码交位置→DX
4E08:067A	E82703	CALL	09A4	: DX 返回当前汉字内码字模段地址
4E08:067D	52	PUSH	DX	: 保存字模指针
4E08:067E	E84803	CALL	09C9	
4E08:0681	E8CE02	CALL	0952	: 若是横扩状态, 取消横扩
4E08:0684	5A	POP	DX	: 送图形打印命令到打印机
4E08:0685	1E	PUSH	DS	
4E08:0686	BEDA	MOV	DS,DX	: 保存DS, 改DS 为字模地址
4E08:0688	E85803	CALL	09E3	: 32字节字模送加工区
4E08:068B	8BDA03	MOV	BX,03DA	: 取字模加工区首址
-u				
4E08:068E	1F	POP	DS	
4E08:068F	1E	PUSH	DS	
4E08:0690	8BF3	MOV	SI,BX	: SI 指向字模加工区首址
4E08:0692	B91000	MOV	CX,0010	: 16→CX
4E08:0695	B602	MOV	DH,02	
4E08:0697	B000	MOV	AL,00	
4E08:0699	E8A302	CALL	093F	: 00送打印机第1组针头
4E08:069C	E8A002	CALL	093F	: 00送打印机第2组针头
4E08:069F	AC	LODSB		
4E08:06A0	E87103	CALL	0A14	: 1列字模头4位纵扩后送打印机3组针头
4E08:06A3	4E	DEC	SI	: 移回字模指针
4E08:06A4	FECE	DEC	DH	
4E08:06A6	75EF	JNZ	0697	: 当前字模列再送打印
4E08:06A8	46	INC	SI	: 调整字模指针到下一列
4E08:06A9	46	INC	SI	
4E08:06AA	E2E9	LOOP	0695	: 直到16列字模头4位纵扩后
4E08:06AC	EB16	JMP	06C4	(每列打二次)全部送完
-u				
4E08:06AE	90	NOP		: 转06C4
4E08:06AF	B90B00	MOV	CX,000B	: 此段指令跳过不执行
4E08:06B2	33F6	XOR	SI,SI	
4E08:06B4	E83301	CALL	07EA	
4E08:06B7	E2F9	LOOP	06B2	
4E08:06B9	B90B00	MOV	CX,000B	
4E08:06BC	BE0100	MOV	SI,0001	
4E08:06BF	E82801	CALL	07EA	
4E08:06C2	E2FB	LOOP	06BC	
4E08:06C4	1F	POP	DS	: 保存数据段
4E08:06C5	E80E03	CALL	09D6	: 若横扩送横扩指令到打印机
4E08:06C8	E88901	CALL	0854	: 各送2列空白

4E08:060B E88601	CALL	0854	
-u			
4E08:060E E88301	CALL	0854	
4E08:06D1 FE06D503	INC	BYTE PTR [03D5]	: 读打印数据指针+2
4E08:06D5 FE06D503	INC	BYTE PTR [03D5]	
4E08:06D9 EB1B	JMP	06F6	
4E08:06DB 90	NOF		
4E08:06DE A802	TEST	AL,02	: 读出属性是横扩汉字
4E08:06DE 7407	JZ	06E7	
4E08:06E0 FE06D503	INC	BYTE PTR [03D5]	: 是, 读打印数据指针+1
4E08:06E4 E86701	CALL	084E	: 18列空白送打印
4E08:06E7 FE06D503	INC	BYTE PTR [03D5]	
4E08:06EB E86001	CALL	084E	
-u			
4E08:06EE A801	TEST	AL,01	: 读出属性是A号字
4E08:06F0 7404	JZ	06F6	
4E08:06F2 FE06D503	INC	BYTE PTR [03D5]	: 读打印数据指针+1
4E08:06F6 8A1ED503	MOV	BL,[03D5]	: 一行打印完?
4E08:06FA 3A1ECE03	CMF	BL,[03CE]	
4E08:06FE 7403	JZ	0703	
4E08:0700 E95AFF	JMP	065D	: 未完继续
4E08:0703 B00D	MOV	AL,0D	: 一行完送回车
4E08:0705 E85AFD	CALL	0462	
4E08:0708 E87A02	CALL	0985	: 走纸
4E08:070B C604550300	MOV	BYTE PTR [03D5],00	: 读打印数据指针清0
-u			
4E08:0710 8A1ED503	MOV	BL,[03D5]	
4E08:0714 B700	MOV	BH,00	
4E08:0716 8BF3	MOV	SI,BX	
4E08:0718 8AA40D00	MOV	AH,[SI+000D]	: 当前指针数据→AH
4E08:071C 8AB4ED01	MOV	AL,[SI+01ED]	属性→AL
4E08:0720 A804	TEST	AL,04	: 打印数据是纵扩汉字内码?
4E08:0722 747E	JZ	07A2	
4E08:0724 8AB40E00	MOV	AL,[SI+000E]	: 是, 第二字节内码→AL
4E08:0728 257F7F	AND	AX,7F7F	: 二字节内码高位清0→DX
4E08:072B 8BD0	MOV	DX,AX	
4E08:072D E87402	CALL	09A4	: DX返回汉字对应字模起始段址
-u			
4E08:0730 52	PUSH	DX	: 存字模指针
4E08:0731 E89502	CALL	09C9	: 若是横扩取消之
4E08:0734 EB1B02	CALL	0952	: 送图形打印命令到打印机
4E08:0737 5A	POP	DX	: 恢复字模指针
4E08:0738 1E	PUSH	DS	
4E08:0739 8EDA	MOV	DS,DX	
4E08:073B E8A502	CALL	09E3	: 32字节字模存加工区
4E08:073E BBD403	MOV	BX,03DA	: 取字模加工区首址
4E08:0741 1F	POP	DS	
4E08:0742 1E	PUSH	DS	

4E08:0743 88F3	MOV	SI,BX	
4E08:0745 B91000	MOV	CX,0010	: 16→CX
4E08:0748 B602	MOV	DH,02	
4E08:074A AC	LODSB		: 字模加工区取 1 字节模 (汉字上半部)
4E08:074B D0E0	SHL	AL,1	: AL 中某列字模下 4 位 (汉字 5—8 行)
4E08:074D D0E0	SHL	AL,1	
4E08:074F D0E0	SHL	AL,1	: 上移到高 4 位
-u			
4E08:0751 D0E0	SHL	AL,1	
4E08:0753 E8BE02	CALL	0A14	: AL 高 4 位字模纵扩后送打印机 1 组针
4E08:0756 AC	LODSB		
4E08:0757 50	PUSH	AX	
4E08:0758 E8B902	CALL	0A14	: AL 中高 4 位 (汉字模 9—12 行) 送 2 组针
4E08:075B 58	POP	AX	
4E08:075C D0E0	SHL	AL,1	: AL 下半部低 4 位 (字模 13—16 行) 上移到高 4 位
4E08:075E D0E0	SHL	AL,1	
4E08:0760 D0E0	SHL	AL,1	
4E08:0762 D0E0	SHL	AL,1	
4E08:0764 EBAD02	CALL	0A14	: AL 高 4 位纵扩后送打印机 3 组针
4E08:0767 FECE	DEC	DH	
4E08:0769 7404	JZ	076F	
4E08:076B 4E	DEC	SI	: 调整字模指针
4E08:076C 4E	DEC	SI	
4E08:076D E8DB	JMP	074A	
4E08:076F E2D7	LOOP	0748	: 直到 16 列字模全部送完
-u			
4E08:0771 E817	JMP	078A	
4E08:0773 90	NOP		
4E08:0774 B90800	MOV	CX,0008	: 跳过此段直到 078 A
4E08:0777 BE1000	MOV	SI,0010	
4E08:077A E86D00	CALL	07EA	
4E08:077D E2F5	LOOP	0777	
4E08:077F B90800	MOV	CX,0008	
4E08:0782 BE1100	MOV	SI,0011	
4E08:0785 E86200	CALL	07EA	
4E08:0788 E2F8	LOOP	0782	
4E08:078A 1F	POP	DS	: 恢复 DS
4E08:078B E84B02	CALL	09D6	: 恢复 DS 打印机横扩状态
4E08:078E E8C300	CALL	0854	: 各送 2 列空白
-u			
4E08:0791 E8C000	CALL	0854	
4E08:0794 E8BD00	CALL	0854	
4E08:0797 FE06D503	INC	BYTE PTR [03D5]	: 读打印数据指针 + 2
4E08:079B FE06D503	INC	BYTE PTR [03D5]	
4E08:079F EB1F	JMP	07C0	
4E08:07A1 90	NOP		
4E08:07A2 3C00	CMP	AL,00	: 非汉字内码?
4E08:07A4 7508	JNZ	07AE	: 汉字内码转 07AE

4E08:07A6 B6C4	XCHG	AH,AL	: 非汉字, 打印数据送AL
4E08:07AB E8D900	CALL	0884	
4E08:07AB EB13	JMP	07C0	
4E08:07AD 90	NOP		
4E08:07AE A2CD03	MOV	[03CD],AL	: 读出属性→ 打印控制标志
-u			
4E08:07B1 8A840E00	MOV	AL,[SI+000E]	: 取第二节字内码
4E08:07B5 E80001	CALL	08BB	: 送汉字模到打印机
4E08:07BB FE06D503	INC	BYTE PTR [03D5]	: 读打印指针+ 2
4E08:07BC FE06D503	INC	BYTE PTR [03D5]	
4E08:07C0 8A1ECE03	MOV	BL,[03CE]	: 写打印指针→BL
4E08:07C4 3A1ED503	CMP	BL,[03D5]	
4E08:07C8 7403	JZ	07CC	: 读完?
4E08:07CA E943FF	JMP	0710	: 未完继续
4E08:07CD B00D	MOV	AL,0D	
4E08:07CF E890FC	CALL	0462	: 读完, 回车
-u			
4E08:07D2 B00A	MOV	AL,0A	换行
4E08:07D4 E89BFC	CALL	0462	
4E08:07D7 C3	RET		
4E08:07DB 803ED803FF	CMP	BYTE PTR [03D8],FF	: 此段指令未用
4E08:07DD 750A	JNZ	07E9	
4E08:07DF C606D80300	MOV	BYTE PTR [03D8],00	
4E08:07E4 B00A	MOV	AL,0A	
4E08:07E6 E879FC	CALL	0462	
4E08:07E9 C3	RET		
4E08:07EA E85801	CALL	0945	
4E08:07ED 8AC4	MOV	AL,AH	
4E08:07EF 8BE8	MOV	BP,AX	
4E08:07F1 EB0B00	CALL	07FF	
-u			
4E08:07F4 8BC5	MOV	AX,BP	
4E08:07F6 EB0600	CALL	07FF	
4E08:07F9 8BC5	MOV	AX,BP	
4E08:07FB EB0100	CALL	07FF	
4E08:07FE C3	RET		
4E08:07FF 51	PUSH	CX	
4E08:0800 EB1300	CALL	0816	
4E08:0803 8AC6	MOV	AL,DH	
4E08:0805 EB3701	CALL	093F	
4E08:0808 8BC7	MOV	AX,DI	
4E08:080A 86E0	XCHG	AL,AH	
4E08:080C EB3001	CALL	093F	
4E08:080F 8BC7	MOV	AX,DI	
4E08:0811 EB2B01	CALL	093F	
-u			
4E08:0814 59	POP	CX	
4E08:0815 C3	RET		

4E08:0816 B90800	MOV	CX,0008	
4E08:0819 B203	MOV	DL,03	
4E08:081B BAC4	MOV	AL,AH	
4E08:081D D0E0	SHL	AL,1	
4E08:081F D1D7	RCL	DI,1	
4E08:0821 D0D6	RCL	DH,1	
4E08:0823 FECA	DEC	DL	
4E08:0825 75F4	JNZ	081B	
4E08:0827 D0E4	SHL	AH,1	
4E08:0829 E2EE	LOOP	0819	: 以上未用
4E08:082B C3	RET		
4E08:082C B01B	MOV	AL,1B	: 空白列图形数据“esc”
4E08:082E EB0E01	CALL	093F	
4E08:0831 B049	MOV	AL,49	: “I”
4E08:0833 EB0901	CALL	093F	
-u			
4E08:0836 B000	MOV	AL,00	: “0”
4E08:0838 EB0401	CALL	093F	
4E08:083B BAC1	MOV	AL,CL	: CL 值送打印机
4E08:083D E8FF00	CALL	093F	
4E08:0840 B000	MOV	AL,00	: 00→AL
4E08:0842 E8FA00	CALL	093F	: 00送打印机第一组针头
4E08:0845 EBF700	CALL	093F	: 00送打印机第二组针头
4E08:0848 E8F400	CALL	093F	: 00送打印机第三组针头
4E08:084B E2F3	LOOP	0840	: 若CX ≠ 0 再送一列空白
4E08:084D C3	RET		
4E08:084E B112	MOV	CL,12	: 18→CL, 送18列空白到打印机
4E08:0850 E8D9FF	CALL	082C	
4E08:0853 C3	RET		
4E08:0854 B102	MOV	CL,02	: 送2列空白到打印机
-u			
4E08:0856 E8D3FF	CALL	082C	
4E08:0859 C3	RET		
4E08:085A C606CE0300	MOV	BYTE PTR [03CE],00	: 写打印数据指针清0
4E08:085F C606D50300	MOV	BYTE PTR [03D5],00	: 读打印数据指针清0
4E08:0864 C706D3030000	MOV	WORD PTR [03D3],0000	: 列计数器清0
4E08:086A B026CD0397	AND	BYTE PTR [03CD],97	: 初始化控制标志
4E08:086F C606030000	MOV	BYTE PTR [0003],00	: 第一字节汉字内码标志清0
4E08:0874 C606D70300	MOV	BYTE PTR [03D7],00	: ESC 命令标志清0
-u			
4E08:0879 C606D60300	MOV	BYTE PTR [03D6],00	: 汉字内码标志清0
4E08:087E C606D80300	MOV	BYTE PTR [03D8],00	: 超出标志清0
4E08:0883 C3	RET		
4E08:0884 E8DBFB	CALL	0462	送打印
4E08:0887 FE06D503	INC	BYTE PTR [03D5]	: 字模加工区指针+1
4E08:088B C3	RET		
4E08:088C B03ED603FF	CMF	BYTE PTR [03D6],FF	: 汉字内码? 不是转08B 2
4E08:0891 741F	JZ	08B2	

4E08:0893 F606CD0304	TEST	BYTE PTR [03CD],04	: 汉字内码, C 号字?
4E08:0898 740B	JZ	08A5	: 不是转08A 5
-u			
4E08:089A 8306D30336	ADD	WORD PTR [03D3],+36	: 是, 列计数器+54
4E08:089F 800ECD0320	OR	BYTE PTR [03CD],20	: 控制标志第5位置1
4E08:08A4 C3	RET		
4E08:08A5 F606CD0302	TEST	BYTE PTR [03CD],02	: B 号字? 不是转08B 2
4E08:08AA 7406	JZ	08B2	
4E08:08AC 8306D30324	ADD	WORD PTR [03D3],+24	: 是, 列计数器+36
4E08:08B1 C3	RET		
4E08:08B2 8306D30312	ADD	WORD PTR [03D3],+12	: A 号字, 列计数器+18
4E08:08B7 C3	RET		
4E08:08B8 8BD0	MOV	DX,AX	: 汉字内码→DX
-u			
4E08:08BA E8E700	CALL	09A4	: DX 返回内码在字模库段址
4E08:08BD 52	PUSH	DX	: 保存DX
4E08:08BE E80801	CALL	09C9	: 取消横扩
4E08:08C1 E8A300	CALL	0967	: 送图形打印命令
4E08:08C4 5A	POP	DX	: 恢复字模指针
4E08:08C5 1E	PUSH	DS	
4E08:08C6 BEDA	MOV	DS,DX	
4E08:08C8 B90800	MOV	CX,0008	
4E08:08CB 33F6	XOR	SI,SI	: 清 SI (取字模指针)
4E08:08CD E87500	CALL	0945	: AH 返回字模左上角一列字模
4E08:08D0 8AC4	MOV	AL,AX	: 字模→AL, AX→BP 暂存
4E08:08D2 8BE8	MOV	BP,AX	
4E08:08D4 E86800	CALL	093F	: 左上角一列字模送打印机第1组针
4E08:08D7 BE1000	MOV	SI,0010	
-u			
4E08:08DA E86800	CALL	0945	: AH 返回左下角一列字模
4E08:08DD 8AC4	MOV	AL,AX	
4E08:08DF 8BF8	MOV	DI,AX	
4E08:08E1 E85B00	CALL	093F	: 左下角一列字模送打印机第2组针
4E08:08E4 B000	MOV	AL,00	
4E08:08E6 E85600	CALL	093F	: 00送打印机第3组针
4E08:08E9 E83B00	CALL	0927	: 若是横扩, BP,DI 字模再打一次
4E08:08EC E2DD	LOOP	08CB	直到右半部打完
4E08:08EE B90800	MOV	CX,0008	
4E08:08F1 BE0100	MOV	SI,0001	
4E08:08F4 E84E00	CALL	0945	: AH 返回右下角一列字模
4E08:08F7 8AC4	MOV	AL,AX	
4E08:08F9 8BE8	MOV	BP,AX	
-u			
4E08:08FB E84100	CALL	093F	
4E08:08FE BE1100	MOV	SI,0011	: 17→SI
4E08:0901 E84100	CALL	0945	: AH 返回右下角一列字模
4E08:0904 8AC4	MOV	AL,AX	
4E08:0906 8BF8	MOV	DI,AX	

4E08:0908 E83400	CALL	093F	: 送打印机
4E08:090B B000	MOV	AL,00	
4E08:090D E82F00	CALL	093F	: 00送打印机第3组针
4E08:0910 EB1400	CALL	0927	: 若是横扩, BP,DI 中字模再打一次
4E08:0913 E2DC	LOOP	08F1	直到左半部字模全部打印
4E08:0915 1F	POP	DS	
4E08:0916 EBB000	CALL	09D6	
4E08:0919 F606CD0302	TEST	BYTE PTR [03CD],02	: 横扩?
-u			
4E08:091E 7403	JZ	0923	: 不是, 转子程序返回
4E08:0920 E831FF	CALL	0854	: 2列空白送打印
4E08:0923 E82EFF	CALL	0854	: 2列空白送打印
4E08:0926 C3	RET		
4E08:0927 26	ES:		
4E08:0928 F606CD0302	TEST	BYTE PTR [03CD],02	: 横扩?
4E08:092D 740F	JZ	093E	
4E08:092F 8BC5	MOV	AX,BP	: BP 中字模送打印
4E08:0931 E80B00	CALL	093F	
4E08:0934 8BC7	MOV	AX,DI	: DI 中字模送打印
4E08:0936 E80600	CALL	093F	
4E08:0939 B000	MOV	AL,00	
4E08:093B E80100	CALL	093F	
-u			
4E08:093E C3	RET		
4E08:093F 50	PUSH	AX	
4E08:0940 EB1FFB	CALL	0462	: AL 送打印机
4E08:0943 58	POP	AX	
4E08:0944 C3	RET		
4E08:0945 B608	MOV	DH,08	: 08→AH
4E08:0947 AC	LODSB		: 取一字字节字模SI+1→SI 移进AH
4E08:0948 D2D8	RCR	AL,CL	
4E08:094A D0D4	RCL	AH,1	
4E08:094C 46	INC	SI	: 调整字模指针
4E08:094D FECE	DEC	DH	
4E08:094F 75F6	JNZ	0947	: 直到8字节同一位字模移进AH
4E08:0951 C3	RET		
4E08:0952 B01B	MOV	AL,1B	: esc 送打印
4E08:0954 E8E8FF	CALL	093F	
4E08:0957 B049	MOV	AL,49	: I 送打印
4E08:0959 E8E3FF	CALL	093F	
4E08:095C B000	MOV	AL,00	: 00送打印
-u			
4E08:095E EBDEFF	CALL	093F	
4E08:0961 B020	MOV	AL,20	: 20H = 32送打印
4E08:0963 E8D9FF	CALL	093F	
4E08:0966 C3	RET		
4E08:0967 B01B	MOV	AL,1B	: esc
4E08:0969 E8F6FA	CALL	0462	

4E08:096C B049	MOV	AL,49	: " I "
4E08:096E EBF1FA	CALL	0462	
4E08:0971 B000	MOV	AL,00	: " 0 "
4E08:0973 EBECFA	CALL	0462	
4E08:0976 B010	MOV	AL,10	
4E08:0978 F606CD0302	TEST	BYTE PTR [03CD],02	: 横扩?
4E08:097D 7402	JZ	0981	
-u			
4E08:097F B020	MOV	AL,20	: 是, 20送打印
4E08:0981 EBDEFA	CALL	0462	
4E08:0984 C3	RET		
4E08:0985 B01B	MOV	AL,1B	: esc
4E08:0987 E8D8FA	CALL	0462	
4E08:098A B057	MOV	AL,57	: " W "
4E08:098C E8D3FA	CALL	0462	
4E08:098F B030	MOV	AL,30	: " 0 "
4E08:0991 EBCEFA	CALL	0462	
4E08:0994 B030	MOV	AL,30	: " 0 " 送打印
4E08:0996 E8C9FA	CALL	0462	
4E08:0999 B031	MOV	AL,31	: " I " 送打印
4E08:099B E8C4FA	CALL	0462	
4E08:099E B036	MOV	AL,36	: " b " 送打印
-u			
4E08:09A0 E8BFFA	CALL	0462	
4E08:09A3 C3	RET		
4E08:09A4 81E27F7F	AND	DX,7F7F	: 内码换国标
4E08:09AB 50	PUSH	AX	
4E08:09A9 80FE30	CMP	DH,30	: > 16区内码?
4E08:09AC 7203	JB	09B1	
4E08:09AE 80EE08	SUB	DH,08	: 是, 区号 - 8
4E08:09B1 80EE21	SUB	DH,21	: 国标区号换为区位码区号
4E08:09B4 B05E	MOV	AL,5E	: 94→AL
4E08:09B6 F6E6	MUL	DH	
4E08:09BB 80EA21	SUB	DL,21	: 国标位号换为区位码位号
4E08:09BB 32F6	XOR	DH,DH	
4E08:09BD 03C2	ADD	AX,DX	
4E08:09BF D1E0	SHL	AX,1	: AX 乘 2 作段址基值
-u			
4E08:09C1 03060000	ADD	AX,[0000]	: 加字模库起始段码
4E08:09C5 8BD0	MOV	DX,AX	: DX 得当前字模段址
4E08:09C7 5B	POP	AX	
4E08:09C8 C3	RET		
4E08:09C9 F606CD0310	TEST	BYTE PTR [03CD],10	: 横扩?
4E08:09CE 7405	JZ	09D5	
4E08:09D0 B00F	MOV	AL,0F	: 是SI. 送打印
4E08:09D2 E8BDFA	CALL	0462	: 取消横扩返回
4E08:09D5 C3	RET		
4E08:09D6 F606CD0310	TEST	BYTE PTR [03CD],10	: 横扩?

4E08:09D8 7405	JZ	09E2	
4E08:09DD B00E	MOV	AL,0E	: 是SO 送打印
4E08:09DF EB80FA	CALL	0462	
-u			
4E08:09E2 C3	RET		
4E08:09E3 5F	POP	DI	
4E08:09E4 07	POP	ES	
4E08:09E5 06	PUSH	ES	
4E08:09E6 57	PUSH	DI	
4E08:09E7 BFDA03	MOV	DI,03DA	: 字模加工区首址
4E08:09EA B90800	MOV	CX,0008	: 8→CX
4E08:09ED BE0000	MOV	SI,0000	: 读字模指针
4E08:09F0 EB1A00	CALL	0A0D	: 左上角一列字模存字模加工区
4E08:09F3 BE1000	MOV	SI,0010	: 16→CX
4E08:09F6 EB1400	CALL	0A0D	: 左下角一列字模存字模加工区
4E08:09F9 E2F2	LOOP	09ED	: 直到左半部字模全存加工区
4E08:09FB B90800	MOV	CX,0008	
4E08:09FE BE0100	MOV	SI,0001	: 1→SI 读字模指针
4E08:0A01 EB0900	CALL	0A0D	: 右上角一列字模存加工区
-u			
4E08:0A04 BE1100	MOV	SI,0011	: +16→SI
4E08:0A07 EB0300	CALL	0A0D	: 右下角一列字模加存工区
4E08:0A0A E2F2	LOOP	09FE	: 直到右半部字模全存加工区
4E08:0A0C C3	RET		
4E08:0A0D EB35FF	CALL	0945	: AH 返回DS 为起始段汉字模
4E08:0A10 8AC4	MOV	AL,AH	: (横排变纵排), 纵排字模→AL
4E08:0A12 AA	STOSB		: 调整存字模加工区指针
4E08:0A13 C3	RET		
4E08:0A14 51	PUSH	CX	: AL 中高4位字模纵扩为1字节字模
4E08:0A15 B90400	MOV	CX,0004	: 图形数据送打印子程序
4E08:0A18 BAE0	MOV	AH,AL	
4E08:0A1A B202	MOV	DL,02	
4E08:0A1C D0E0	SHL	AL,1	
4E08:0A1E D0D3	RCL	BL,1	
4E08:0A20 BAC4	MOV	AL,AH	
4E08:0A22 FECA	DEC	DL	
-u			
4E08:0A24 75F6	JNZ	0A1C	
4E08:0A26 D0E0	SHL	AL,1	
4E08:0A28 E2EE	LOOP	0A1B	
4E08:0A2A BAC3	MOV	AL,BL	: 字模图形数据→打印
4E08:0A2C EB10FF	CALL	093F	
4E08:0A2F 59	POP	CX	
4E08:0A30 C3	RET		
4E08:0A31 9C	PUSHF		
4E08:0A32 9A54FF00F0	CALL	F000:FF54	: ROM BIOS INT 05H 地址
4E08:0A37 C3	RET		: (F000:FF 54 原装机)
4E08:0A38 1E	PUSH	DS	: 新INT 05H入口

```

4E08:0A39 53          PUSH  BX
4E08:0A3A BB4000     MOV   BX,0040
4E08:0A3D 8ED8       MOV   DS,BX
4E08:0A3F BB4900     MOV   BX,0049
4E08:0A42 803F04     CMP   BYTE PTR [BX],04      : 图形方式?
-u
  ^ Error
-u
4E08:0A45 720B       JB    0A4F
4E08:0A47 803F07     CMP   BYTE PTR [BX],07      : 单色板?
4E08:0A4A 7403       JZ    0A4F
4E08:0A4C EB23       JMP   0A71                   : 是图形转A 77
4E08:0A4E 90         NOP
4E08:0A4F 56         PUSH  SI
4E08:0A50 33DB       XOR   BX,BX
4E08:0A52 8ED8       MOV   DS,BX
4E08:0A54 BE5C00     MOV   SI,005C
4E08:0A57 FF34       PUSH  [SI]
4E08:0A59 FF7402     PUSH  [SI+02]
4E08:0A5C C704D2EF     MOV   WORD PTR [SI],EFD2    : 原装机 BIOS INT 17H
4E08:0A60 C7440200F0   MOV   WORD PTR [SI+02],F000 : 地址F 000 : EFD 2
-u
4E08:0A65 EBC9FF     CALL  0A31                   : ROM BIOS INT 05H
4E08:0A68 8F4402     POP   [SI+02]                : 恢复原17H 中断指针
4E08:0A6B 8F04       POP   [SI]
4E08:0A6D 5E         POP   SI
4E08:0A6E 5B         POP   BX
4E08:0A6F 1F         POP   DS
4E08:0A70 CF         IRET
4E08:0A71 06         PUSH  ES                      : 图形方式打屏幕子程序
4E08:0A72 52         PUSH  DX
4E08:0A73 56         PUSH  SI
4E08:0A74 57         PUSH  DI
4E08:0A75 51         PUSH  CX
4E08:0A76 55         PUSH  BP
4E08:0A77 50         PUSH  AX
4E08:0A78 BB0088     MOV   AX,B800                 : 数据段指向彩显卡RAM 段
4E08:0A7B 8ED8       MOV   DS,AX
4E08:0A7D 33F6       XOR   SI,SI                    : SI 指向RAM 首址
4E08:0A7F B91000     MOV   CX,0010
4E08:0A82 EB00FF     CALL  09B5                     : 打印机走纸
-u
4E08:0A85 EB9500     CALL  0B1D                     : 打印机单向打印
4E08:0A88 EB1800     CALL  0AA3                     : RAM区0—91行送打印
4E08:0A8B EB7500     CALL  0B03                     : 送图形打印到打印机
4E08:0A8E EBA200     CALL  0B33                     : RAM区92—99行送打印
4E08:0A91 EB9400     CALL  0B2B                     : 双向打印指令→打印机
4E08:0A94 B00A       MOV   AL,0A

```


4E08:0A96 EBC9F9	CALL	0462	: 换行符送打印
4E08:0A99 58	POP	AX	
4E08:0A9A 5D	POP	BP	
4E08:0A9B 59	POP	CX	
4E08:0A9C 5F	POP	DI	
4E08:0A9D 5E	POP	SI	
4E08:0A9E 5A	POP	DX	
4E08:0A9F 07	POP	ES	
4E08:0AA0 5B	POP	BX	
4E08:0AA1 1F	POP	DS	
4E08:0AA2 CF	IRET		: 0—191 行图形送打印子程序
4E08:0AA3 51	PUSH	CX	
4E08:0AA4 56	PUSH	SI	
-u			
4E08:0AA5 B95000	MOV	CX,0050	
4E08:0AAB E85800	CALL	0B03	: 送图形打印命令到打印机
4E08:0AAB E80C00	CALL	0ABA	: 12行图形送打印
4E08:0AAE E8D4FE	CALL	0985	: 打印机走纸
4E08:0AB1 5E	POP	SI	
4E08:0AB2 59	POP	CX	
4E08:0AB3 81C6E001	ADD	SI,01E0	: RAM寻址指针加12条扫描行
4E08:0AB7 E2EA	LOOP	0AA3	
4E08:0AB9 C3	RET		
4E08:0ABA 51	PUSH	CX	
4E08:0ABB B90800	MOV	CX,0008	
4E08:0ABE 51	PUSH	CX	
4E08:0ABF 56	PUSH	SI	
4E08:0AC0 E80900	CALL	0ACC	: 12个扫描行同一列数据送打印
4E08:0AC3 5E	POP	SI	
4E08:0AC4 59	POP	CX	
-u			
4E08:0AC5 E2F7	LOOP	0ABE	: 直到8列数据全送完
4E08:0AC7 46	INC	SI	
4E08:0AC8 59	POP	CX	
4E08:0AC9 E2EF	LOOP	0ABA	: 直到80列数据全送完
4E08:0ACB C3	RET		
4E08:0ACC 51	PUSH	CX	: 12个扫描行同一列经纵扩送打印子程
4E08:0ACD E80600	CALL	0AD6	
4E08:0AD0 59	POP	CX	
4E08:0AD1 51	PUSH	CX	
4E08:0AD2 E80100	CALL	0AD6	: 4个扫描行同一列图形数据送打印
4E08:0AD5 59	POP	CX	
4E08:0AD6 8AD9	MOV	BL,CL	
4E08:0AD8 32ED	XOR	CH,CH	
4E08:0ADA B102	MOV	CL,02	
4E08:0ADC 86D9	XCHG	CL,BL	
4E08:0ADE 8A24	MOV	AH,CSII	: 取CRT RAM 偶扫描区一字节→AH
4E08:0AEC 8AFC	MOV	BH,AH	

4E0B:0AE2 D2EF	SHR	BH,CL	
4E0B:0AE4 D0D0	RCL	AL,1	: 纵扩 (将位数据再一次移进AL)
-u			
4E0B:0AE6 D2EC	SHR	AH,CL	
4E0B:0AEB D0D0	RCL	AL,1	
4E0B:0AEA BAA40020	MOV	AH,[SI+2000]	: 取CRT RAM 奇扫描区一字节→AH
4E0B:0AEE 0AFC	MOV	BH,AH	
4E0B:0AF0 D2EF	SHR	BH,CL	
4E0B:0AF2 D0D0	RCL	AL,1	
4E0B:0AF4 D2EC	SHR	AH,CL	
4E0B:0AF6 D0D0	RCL	AL,1	
4E0B:0AFB B3C650	ADD	SI,+50	
4E0B:0AFB B6D9	XCHG	CL,BL	
4E0B:0AFD E2DD	LOOP	0ADC	: 直到4个扫描行数据纵扩完→AL
4E0B:0AFF E860F9	CALL	0462	: AL送打印
4E0B:0B02 C3	RET		
ucs:0b03			
4E0B:0B03 B01B	MOV	AL,1B	: 640列图形打印子程序
4E0B:0B05 E85AF9	CALL	0462	: esc
4E0B:0B06 B049	MOV	AL,49	: "I"
4E0B:0B0A E855F9	CALL	0462	
4E0B:0B0D 51	PUSH	CX	
4E0B:0B0E B96002	MOV	CX,0280	
4E0B:0B11 BAC5	MOV	AL,CH	: "02"送打印
4E0B:0B13 E84CF9	CALL	0462	
4E0B:0B16 BAC1	MOV	AL,CL	: "08"送打印
4E0B:0B18 E847F9	CALL	0462	
4E0B:0B1B 59	POP	CX	
4E0B:0B1C C3	RET		
4E0B:0B1D B01B	MOV	AL,1B	: 单向打印子程序
4E0B:0B1F E840F9	CALL	0462	
4E0B:0B22 B03E	MOV	AL,3E	: 送 "<"到打印机
-u			
4E0B:0B24 E838F9	CALL	0462	
4E0B:0B27 C3	RET		
4E0B:0B28 B01B	MOV	AL,1B	: 双向打印子程
4E0B:0B2A E835F9	CALL	0462	
4E0B:0B2D B03C	MOV	AL,3C	: 送 ">"到打印机
4E0B:0B2F E830F9	CALL	0462	
4E0B:0B32 C3	RET		
4E0B:0B33 B95000	MOV	CX,0050	: 80→CX, 192—199扫描行图形数据送打印
4E0B:0B36 51	PUSH	CX	
4E0B:0B37 B90800	MOV	CX,0008	
4E0B:0B3A 56	PUSH	SI	
4E0B:0B3B 51	PUSH	CX	
4E0B:0B3C E897FF	CALL	0AD6	: 4个扫描行同一列数据打印
4E0B:0B3F 59	POP	CX	
4E0B:0B40 51	PUSH	CX	

4E08:0B41 E892FF	CALL	0AD6	: 下 4 个扫描行同一列数据打印
-u			
4E08:0B44 59	POP	CX	
4E08:0B45 5E	POP	SI	
4E08:0B46 32C0	XOR	AL,AL	
4E08:0B48 E817F9	CALL	0462	: 一个字节空白送打印
4E08:0B4B E2ED	LOOP	0B3A	: 直到 8 列全部打完
4E08:0B4D 59	POP	CX	
4E08:0B4E 46	INC	SI	
4E08:0B4F E2E5	LOOP	0B36	: 直到80×8 列全部打完
4E08:0B51 E831FE	CALL	0985	: 走纸
4E08:0B54 B00A	MOV	AL,0A	
4E08:0B56 E809F9	CALL	0462	: 换行送打印机
4E08:0B59 C3	RET		
4E08:0B5A 0E	PUSH	CS	: ALL 24P exe 执行指令入口
4E08:0B5B 1F	POP	DS	
4E08:0B5C B81035	MOV	AX,3510	
4E08:0B5F CD21	INT	21	
4E08:0B61 BB7500	MOV	BX,0075	
-u			
4E08:0B64 26	ES:		: cclib 16×16字模库段址→AX
4E08:0B65 8B07	MOV	AX,[BX]	
4E08:0B67 A30000	MOV	[0000],AX	
4E08:0B6A 8D160D04	LEA	DX,[040D]	: 改INT 17中断在CS : 04 0 D
4E08:0B6E 8B1725	MOV	AX,2517	
4E08:0B71 CD21	INT	21	
4E08:0B73 8D16380A	LEA	DX,[0A38]	: 改INT 05中断在CS : 0 A 38
4E08:0B77 B80525	MOV	AX,2505	
4E08:0B7A CD21	INT	21	
4E08:0B7C 8D165A0B	LEA	DX,[0B5A]	: DX 指向驱动程序地址
4E08:0B80 B401	MOV	AH,01	: 初始化打印机
4E08:0B82 CD17	INT	17	
-u			
4E08:0B84 B1C20401	ADD	DX,0104	: 加点余量
4E08:0B88 CD27	INT	27	: 驻留内存退出
4E08:0B8A 0000	ADD	[BX+SI],AL	
4E08:0B8C 0000	ADD	[BX+SI],AL	
4E08:0B8E 0000	ADD	[BX+SI],AL	
4E08:0B90 0000	ADD	[BX+SI],AL	
4E08:0B92 0000	ADD	[BX+SI],AL	
4E08:0B94 0000	ADD	[BX+SI],AL	
4E08:0B96 0000	ADD	[BX+SI],AL	
4E08:0B98 0000	ADD	[BX+SI],AL	

第八章 CC-BIOS 高级打印驱动模块

8.1 概述

CC-BIOS 的普通打印驱动模块能驱动打印机输出 16×16 点阵的汉字。CC-BIOS 还有另一种打印驱动模块，它能在24针打印机上输出 24×24 点阵的高质量汉字。我们把这个打印驱动模块称为高级打印驱动模块。这种打印驱动模块向用户提供了两种调用方式，一种方式是通过5号系统功能来调用该模块；另一种方式是通过调用17类软中断来实现对该模块的调用。不管采用哪种调用方式，该模块均需要另外三个附属文件的支持，才能正常运行。这三个文件是CLIB24、CLIB241和CLIB242。它们就是 24×24 点阵的字库。由于它们所含的信息量要比 16×16 点阵的字库(CCLIB)的信息量大得多，故没有象CCLIB那样常驻内存，这三个文件均驻在外存（一般是在硬盘上）。CLIB24中放的是正常字的 24×24 点阵字模，CLIB241中放的是纵扩字的上半字的 24×24 点阵字模，CLIB242中放的是纵扩字的下半字的 24×24 点阵字模。由于CC-BIOS不提供显示 24×24 点阵汉字的功能，所以这三个字库是高级打印驱动模块独用的。因此，它们中的字模，不是象CCLIB中的那样，采用横向点阵，而是采用适合于打印机输出的纵向点阵。其点阵信息可以直接作为打印数据送打印机输出，免去了点阵间转换的麻烦。另外，该模块的运行需要24针打印机支持，而各种24针打印机的控制命令是不同的。CC-DOS 2.13版 24×24 点阵打印有A—P 16种字型，宋、仿宋、黑、楷4种字体，其他功能也有所改进。24点阵打印必须注意机型与CC-DOS匹配，否则死锁不能运行。CC-DOS系统盘上提供的高级打印驱动模块的程序，是根据TH-3070 24针打印机编写的。下面我们就对这一模块进行全面分析。

8.2 高级打印驱动模块的生成

为了使用户通过5号系统功能和17类软中断调用高级打印驱动模块，故必须形成新的21类中断程序模块和17类中断程序模块，并修改21类中断指针和17类中断指针，使它们指向各自的新模块。在CC-DOS的系统盘上有一个D320.EXE，它就是用以完成这些工作的。表8-1列出了这个文件的具体结构。

D 320·EXE 文件的结构

表 8-1

段内地址	内 容	段内地址	内 容
(0000—003C) 0000—013C	17类中断程序的工作区 (一) (PSP)	05F5—0AEE (0700—0BF6)	17类中断程序的工作区 (二) (文件控制块)
003D—0059 (013D—0159)	新21中断程序	0BF A—0CBE	D320.EXE本身的执行代码
005A—05F4 (015A—06E4)	17类中断程序	0CBF—0D14	D320.EXE本身的工作区 (差错信息)

从表8-1中可以知道，D320.EXE文件中含有新的21类中断程序模块和17类中断程序模块，以及它本身的执行代码。当它被调入内存运行时，仅执行它本身的执行代码(0BF A—

0CBE),但是把新的21类和17类中断程序模块也带入了内存。

从上面可以知道,高级打印驱动模块是由21类中断的程序和17类中断程序组成的。

下面分析D320.E X E本身的执行代码的工作流程。它的流程如图8-1所示。

图8-1的流程中,对文件CLIB 24、CLIB 241和CLIB 242都进行了打开操作。但是,这不仅是为了打开这三个文件,同时也是为了检查这三个文件是否都存在。这三个文件中每个记录的内容就是一个字模的信息。因其点阵为 24×24 ,故一个字模需要 3×24 个字节,所以定义文件的记录长为72个字节。修改17类和21类中断指针,是让它们指向了由D320.E X E执行时代入内存的新的17类和21类中断程序模块。

8.3 21类中断程序

在CC-DOS系统中,用户是通过调用21类中断来使用系统功能调用的。为了用户能通过5号系统功能调用来实现 24×24 点阵汉字的输出,故必须在21类中断程序中增加有关内容。图8-2给出了21中断程序的流程。

从图8-2的流程中可以知道,只有在5号系统功能调用时才专门处理,其它系统功能调用均由原来的21类中断程序处理(在形成新的21类中断程序时,并没有破坏原来的21类中断程序)。对5号系统功能调用的处理是调用17类中断程序的0号功能块来实现将字符送打印机输出。

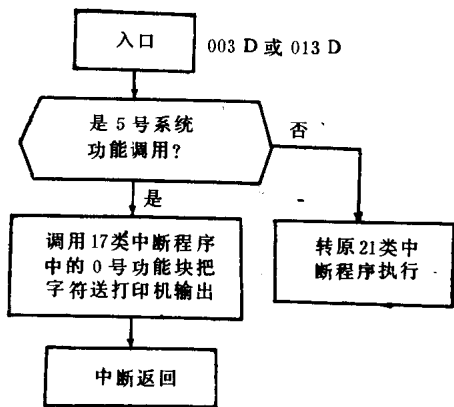


图8-2 新21类中断程序流程

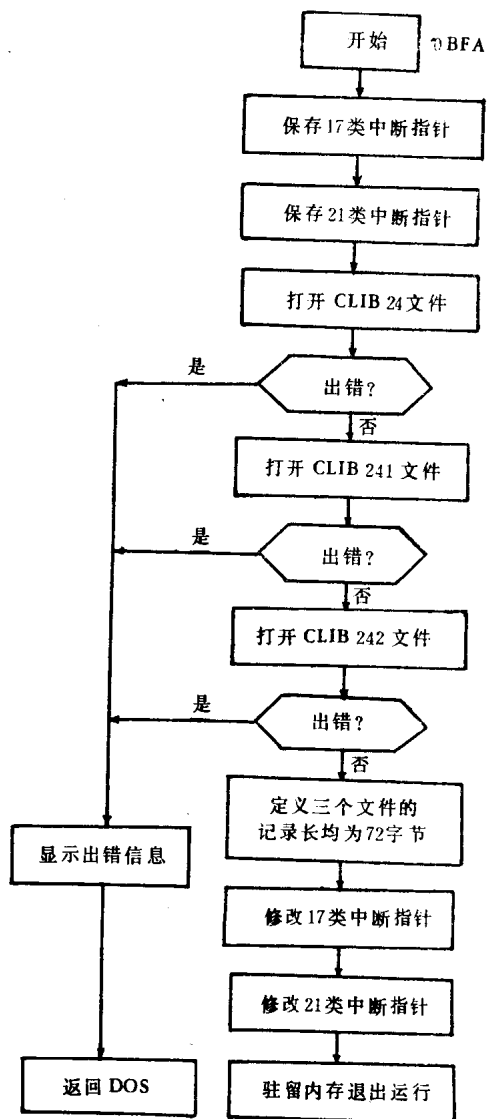


图8-1 D320.E X E的流程

8.4 17类中断程序

17类中断程序由功能块和工作区组成,共有四个功能块,具体情况如表8-2所示。

其中0号功能块最为重要,我们将在下面对它进行专门分析。1号、2号和3号功能块的流程较简单,与CC-BIOS的普通打印驱动模块中17类中断程序的相应功能块类似,所以这儿就不再重复了。

下面介绍几个17类中断程序中的主要工作区:(段地址有D320.E X E和D32024.E X E两种

块号	段内地址	完 成 功 能	块号	段内地址	完 成 功 能
0	0104—05F4 (020F—06F4)	向打印机输出字符	2	00E7—00ED (01E7—01ED)	取打印机的状态
1	00EE—010D (01EE—020D)	打印机初始化	3	0071—0081 (0171—0181)	确定打印机的行宽

以下为D32024. exe数据区)

1. 打印缓冲区 (简称缓冲区)

其起始地址为CS:06A8。用以存放要送打印机输出的当前行的代码及其属性字。要指出的是: 该区中放的是代码, 而不是打印数据。

2. 磁盘信息交换区 (简称交换区)

其地址为CS:0649—0691。它是读取磁盘信息的缓冲区, 存放从字库文件读出的一个记录的内容。要指出的是该区中放的是图形信息 (即打印数据)。

3. 文件控制块 0 (FCB 0)

其地址为CS:(05DA—05FD)是CLIB 24的文件控制块, 用于对该文件的操作。

4. 文件控制块 1 (FCB 1)

其地址为CS:(05FE—0623)。是CLIB 241的文件控制块, 用于对该文件的操作。

5. 文件控制 2 (FCB 2)

其地址为CS:(0624—0648)。是CLIB 242的文件控制块, 用于对该文件的操作。

6. 缓冲区标记

其地址为CS:06A3。记录缓冲区是否满。所谓缓冲区满, 是指把缓冲区中的全部代码转换成对应的点阵后, 点阵的总列数正好等于一行所能容纳的列数。该标记为00时, 表示未滿, 为FF时, 表示已滿。

7. 缓冲区计数器

其地址为CS:06A1。其记录的并不是缓冲区中已存入的代码个数, 而是记录的缓冲区中代码所对应点阵的总列数。

8. 机内码符保存单元

其地址为CS:06A4。用于保存汉字机内码的第一个字节内容。

9. 标志字节

其地址为CS:0699。该字节有关位所代表的内容如下:

(1) 第0位 图形打印状态。该位为0时, 表示目前未向打印机发图形打印控制命令。该位为1时, 表示目前已向打印机发了图形打印控制命令 (打印机的图形打印控制命令为ESC“1”)。

(2) 第1位 机内码状态。该位为0时, 表示前一字符非机内码符; 该位为1时, 表示前一字符为机内码符。

(3) 第2位 ESC状态。该位为0时, 表示前一字符不是ESC; 该位为1时, 表示前一字符是ESC (打印机控制命令的前导符)。

(4) 第3位 I状态。该位为0时, 表示前两个字符不是ESC和I; 该位为1时, 表示

前两个字符是ESC和I，即定义输出汉字字型的命令和图形打印命令。

(5) 第4位 打印数据状态。该位为0时，表示在图形打印命令后还未送打印数据数；该位为1时，表示在图形打印命令后已送打印数据数（占两字节）的高字节。

(6) 第5位 H状态。该位为0时，表示前两个字符不是ESC和H；该位为1时，表示前两个字符是ESC和H，即定义打印页长的命令。

(7) 第7位 当前行状态。该位为0时，表示当前行不要进行纵向扩展；该位为1时，表示当前行要进行纵向扩展，即行中至少有一个字符要进行纵向扩展。

10. 属性字节

其地址为CS: 069A。该字节有关位所代表之内容如下：

(1) 第0位 该位为0时，表示该字不进行横扩；该位为1时，表示该字要进行横扩。

(2) 第1位 该位为0时，表示该字不进行纵扩；该位为1时，表示该字要进行纵扩。

(3) 第2位 该位为0时，表示该字非横向字；该位为1时，表示该字为横向字。

(4) 第7位 该位为0时，表示该字为正常打印；该位为1时，表示该字为横向紧缩打印。另外还有一些工作单元，将在下面各部分中分别予以介绍。

8.5 17类中断程序的0号功能块

17类中断程序的0号功能块，是整个打印驱动模块中最主要的部分。它所完成的工作是：对要输出的字符按类分别进行处理，然后把其对应的字模信息组织成打印数据，并送打印机输出。

0号功能块的工作流程，如图8-3所示。下面对图8-3进行一些解释和说明。

1) 判断是否为图形打印状态，就是判断目前是否已向打印机发了图形打印控制命令ESC“I”。这只要判断标志字节第0位的值即可。

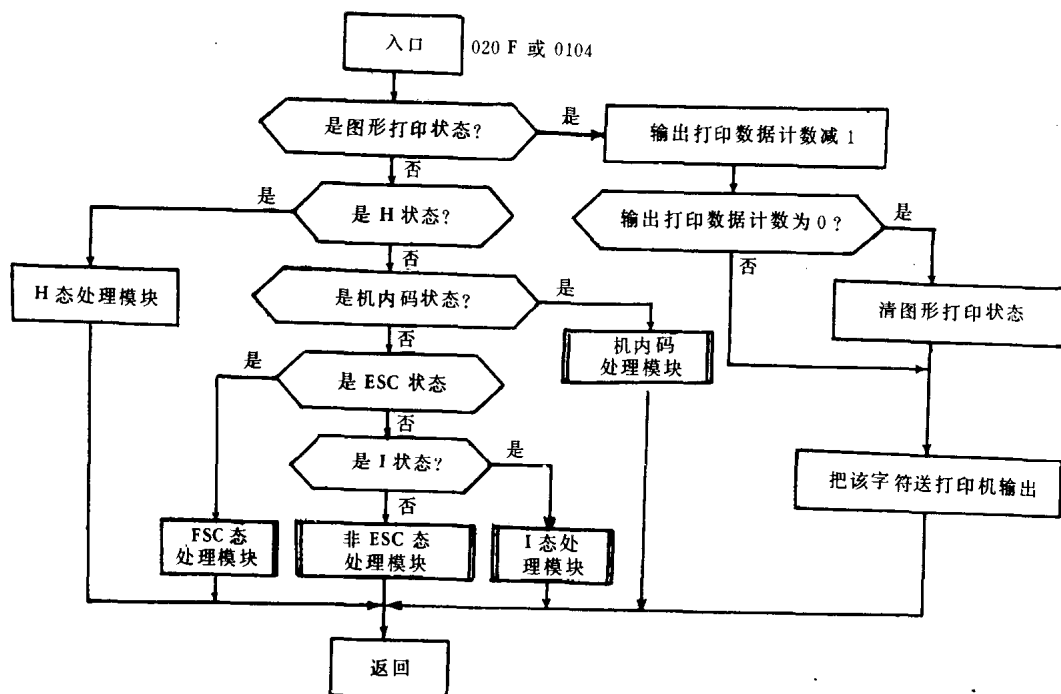


图 8-3 17类中断0号功能块流程

2) 输出打印数据计数器是记录本次共要输出打印数据数量的。在向打印机发出图形打印命令ESC“I”时,其后还要跟上两个字节的内容,用以表示本次图形打印共要输出多少个打印数据组(一组为三个打印数据,表示图形中的一列)。这两字节之值的三倍被保存在输出打印数据计数器中。它的地址为CS:069F。当计数器之值为0时,则表示本次要输出的打印数据已全部输出完毕,本次图形打印结束。

3) H状态表示当前已发定义页长命令ESC“H”,此命令后还有三个字符,表示一个三位十进制数,用以定义每页的行数(即页长)。H态处理模块所做的工作,就是把本次开始输入的三个字符转换成其对应的三位十进制数之值,并把此值存入页长记录单元和清H状态(标志字节的第5位),然后返回。该处理模块的入口地址为CS:058C。页长记录单元的地址为CS:0531。

4) 当前面一个输出字符是ESC时(即ESC状态),则进入ESC态处理模块。该模块要判断本次字符是否能与ESC组成有效命令(只有H和I才能与ESC组成有效命令)。如本次字符为I,则置I状态(标志字节的第3位置位);如本次字符为H,则置H状态(标志字节的第5位置位)。否则标志字节的第0~6位全部清0,然后返回。该模块的入口地址为CS:02A1。(D32024, EXE在CS:0196)。

5) 当前两个输出字符为ESC和I时即(即I状态)执行I态处理模块。I态处理模块的流程如图8-4所示。

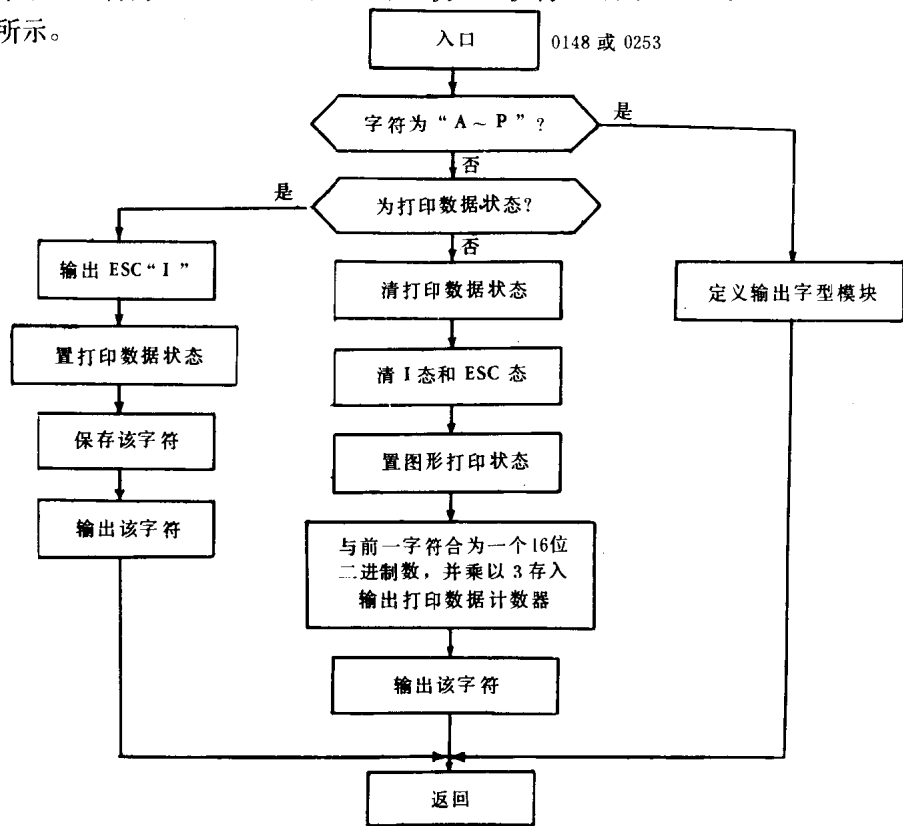


图8-4 I态处理模块流程

ESC“I”有两种含义:当其后续跟字母A~P中的一个时,则表示为定义输出字型的命令,这是一个系统规定的命令。如果其后不是上述的字母时,则表示为图形打印命令。这时,在ESC“I”后边还需要跟两个字节的内容,其意义已在本节的第2点中说明了,这儿不再重复。

下面对图8-4作一些说明。

(i) 定义输出字型模块所完成的工作是, 根据ESC “I” 命令后面跟的字母, 对属性字节中的有关位置以相应的值。实际上, 各种输出的字型均是正常打印、横扩打印、纵扩打印、横向字和紧缩打印的实施以及它们间的组合实施。本模块就是根据字型与实施打印方式的对应关系来置属性字中的相应位之值的。本模块为I态处理模块的一个子程序。它的入口地址为CS:02C6 (D32024、EXE在CS:01BB)。

(ii) 打印数据状态位就是标志字节的第四位。该位不为1时, 表示ESC “I” 命令后的两个用来表示打印数据量的字节均未收到, 故本次字符就是其中的第一个字节, 应把它保存起来, 这个保存单元就借用机内码保存单元。

6) 非ESC态处理模块和机内码处理模块的内容都比较丰富, 我们把它们分别放到本章的第六节和第七节中进行专门分析。

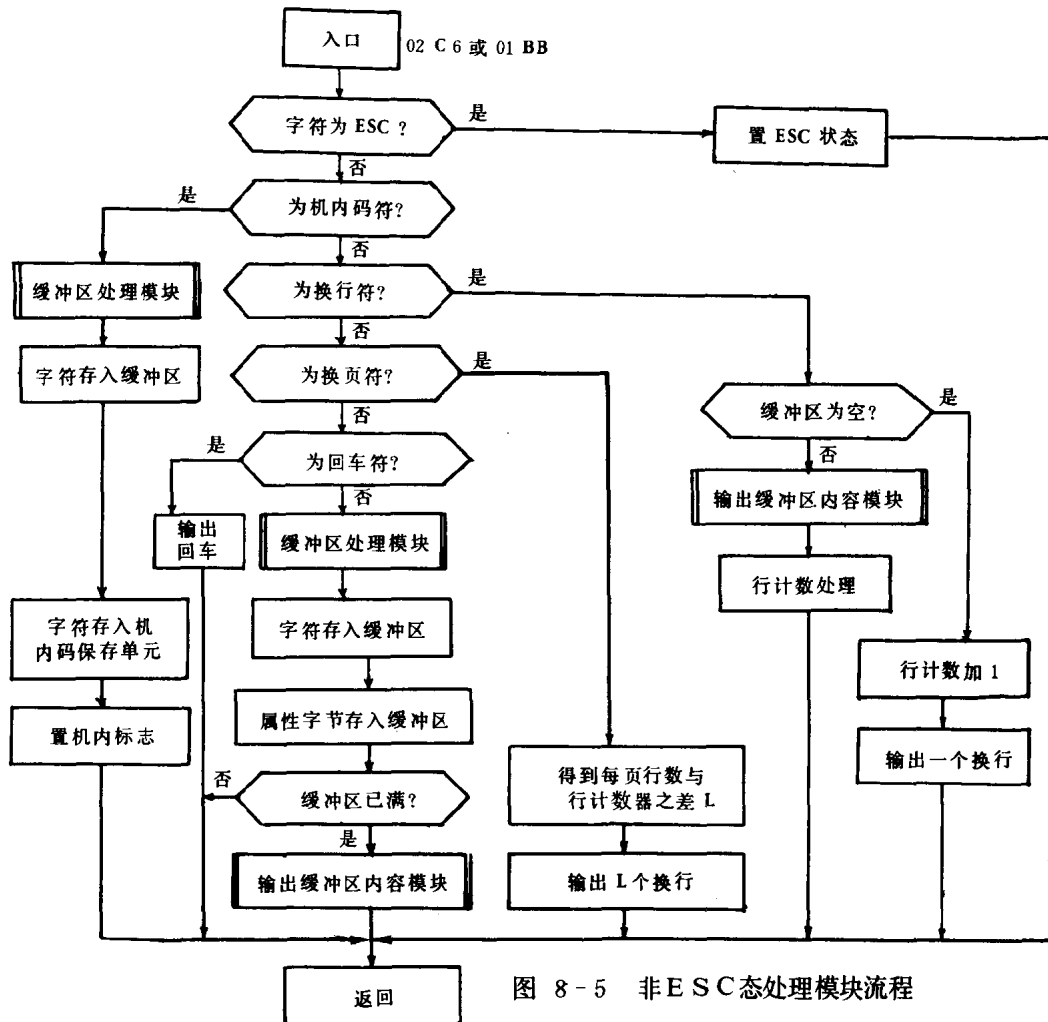


图 8-5 非ESC态处理模块流程

8.6 非ESC态处理模块

非ESC态处理模块是0号功能块中的一个子程序, 它的主要任务是: 对各种欲输出之字符进行相应的处理; 对于特殊字符(如换行符、回车符、换页符、机内码符和ESC符), 则实现它们所代表的功能或做出相应的标记。对于非特殊字符, 则将其存入打印缓冲区中, 必要时须输出这个缓冲区中的内容。

图8-5给出了非ESC态处理模块的流程。从图8-5中可看出, 流程的上半部是对特殊字

符的处理，下半部是把非特殊字符存入缓冲区内。下面对此流程作一些说明。

1) 在两种情况下需要输出缓冲区中的内容。一种是输出换行符时，缓冲区内已有内容。另一种是当要向缓冲区存入内容，而缓冲区已满。

2) 所谓行计数，即行计数器之内容。行计数器是用来记录当前页已打印的行数的。它的地址为C S: 0550。

3) 把字符存入缓冲区时，是与当时的属性字节一起存入的。属性字节描述了该字符的输出字型，我们把它称为该字符的属性字。以后输出该字符时，要用到它的属性字。

4) 输出缓冲区内容模块的功能是把缓冲区中的内容经处理后送打印机输出。该模块的详细情况，在本章第七节中进行介绍。

5) 缓冲区处理模块的功能是完成把字符存入缓冲区前的准备工作。图8-6给出了该模块的工作流程。从图8-6可以知道，该模块的主要任务是：在向缓冲区存入新内容前对缓冲区的剩

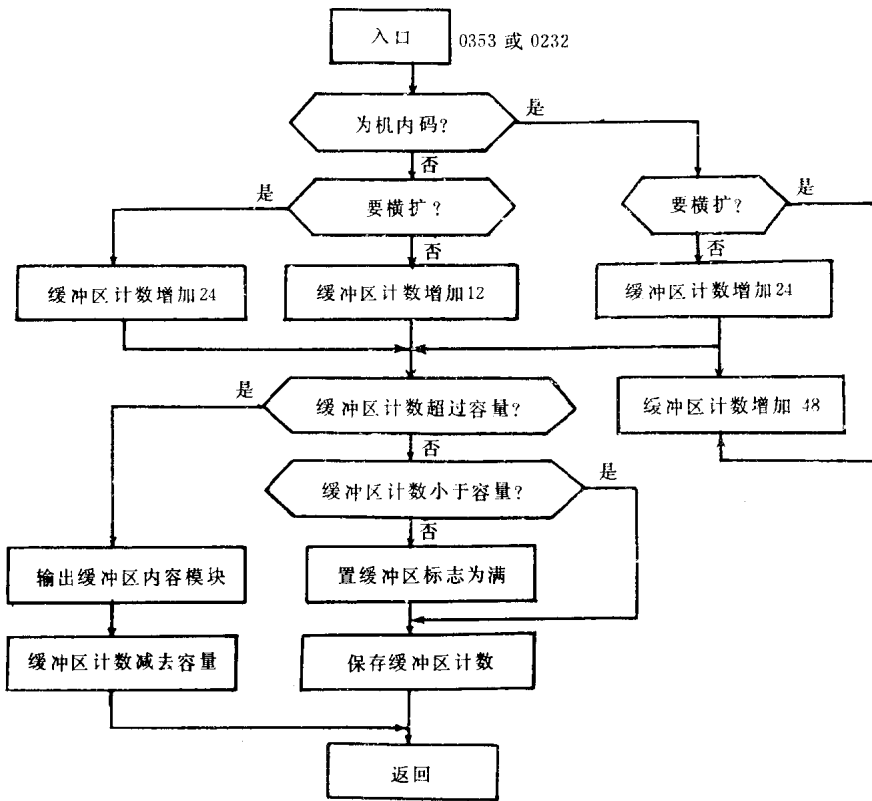


图 8-6 缓冲区处理模块流程

余空间进行判别，如能容下新的内容，则把新内容直接存入；否则就输出缓冲区内的内容。要注意的是，送入的内容须根据其属性字定义的字型给予相应的数据量。只有在缓冲区能容下它所对应的全部数据量时，才能直接存入缓冲区内。只有这样，才能保证今后输出时信息的完整性。

8.7 机内码处理模块

机内码处理模块也是 0 号功能块中的一个子程序。它所做的工作是：把前一次输出的机内码符与本次输出字符合成机内码，然后把它及其属性字存入缓冲区中。

该模块的流程如图 8-7 所示。对图 8-7 中的流程有以下几点解释和说明。

1) 在本流程中, 把当前字符和前一字符存入缓冲区时, 并没有调用缓冲区处理模块来做存入缓冲区前之准备工作。这是因为输出前一字符时 (该字符为机内码符), 已在执行非 E S C 态处理模块时调用过缓冲区处理模块了 (见图 8-5)。当时已把前一字符与当前字符的数据量算在一起了 (见图 8-6)。因此, 本流程中就可把它们直接存入缓冲区。

2) 缓冲区是否满, 只要判缓冲区标记的值即可 (见本章第四节)。

3) 输出缓冲区内容模块所做的工作是把缓冲区中的代码信息转换成图形信息 (即打印数据) 送打印机输出。所以该模块是一个很重要的模块, 是实现汉字打印输出的关键。

图 8-8 给出了该模块的工作流程。下面对图 8-8 进行一些说明。

(i) 判当前行是否要扩展, 只要判标志字节中的第 7 位即可。

(ii) 从图 8-8 可以知道, 对于扩展行 (即行中至少有一个纵扩字) 要分两次打印, 第一次是打印行中纵扩字的上半字, 对于行中的非纵扩字, 则打印相应大小的空白字; 第二次是打印行中纵扩字的下半字, 对于行中的非纵扩字, 则打印其整字样。由此可见, 对于纵扩字在本模块中是单独进行处理的, 而不是象横扩字、横向字和紧缩字那样在输出, 交换区内容模块中进行处理。

(iii) 所谓字库选定标记, 是用以记录当前选定哪个字库进行访问的。在进行纵扩处理时要用到它。其地址为 C S:06 B 8。当其值为 0 时, 表示选定上半字字库 (CLIB 241)。当其值为 1 时, 表示选定下半字字库 (CLIB 242)。

(iv) 在图 8-8 中, 当要输出一般字符时, 是先形成该字符的机内码, 然后再得到其字模。由字符形成其机内码是很简单的, 在国标码中, 字符均排在第三区中, 该区的代码为 23H, 又因为机内码为异形国标码 (最高位置 1), 所以区号应为 A 3H, 字符的 ASCII 码即为位号。所以只要在字符代码前加上一个字节 (A 3H), 即可形成字符的机内码。

(v) 指定单向打印和双向打印均是利用打印机本身的控制命令来实现的。TH-3070 打印机的这两个控制命令分别为 E S C “>” 和 E S C “<”。

(vi) 输出交换区内容模块所完成的工作是: 根据输出字的属性字, 把交换区中的图形信息进行相应的变换, 然后送打印机输出。这儿的变换是对横扩字、横向字和紧缩字进行的, 不包括纵扩字。经这样变换后, 打印机上输出的就为指定字型了。该模块的流程与 C C-BIOS 中的普通打印驱动模块的相应部分类似, 这儿不再重复。

(vii) 读字模到交换区模块的功能是: 把机内码转换成其对应字模在字库文件中的记录号, 再把指定字库文件中的记录 (字模信息) 读入交换区中。其流程如图 8-9 所示。

从图 8-9 中可以看到, 在纵扩时需要字库选定标记。另外, 把机内码转换成记录号是很简单的。因为字库中的记录是按国标码顺序排的, 而机内码为异形国标码, 所以很容易实现它们之间的转换。

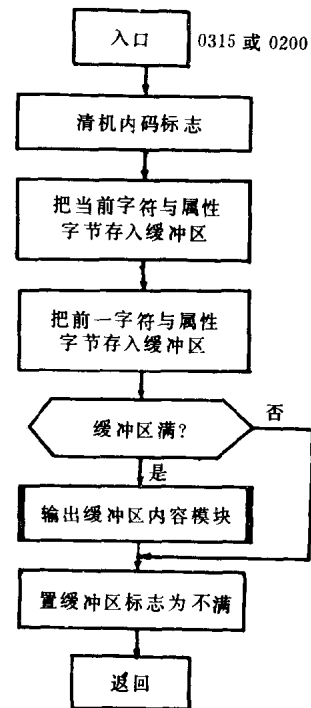


图 8-7 机内码处理流程

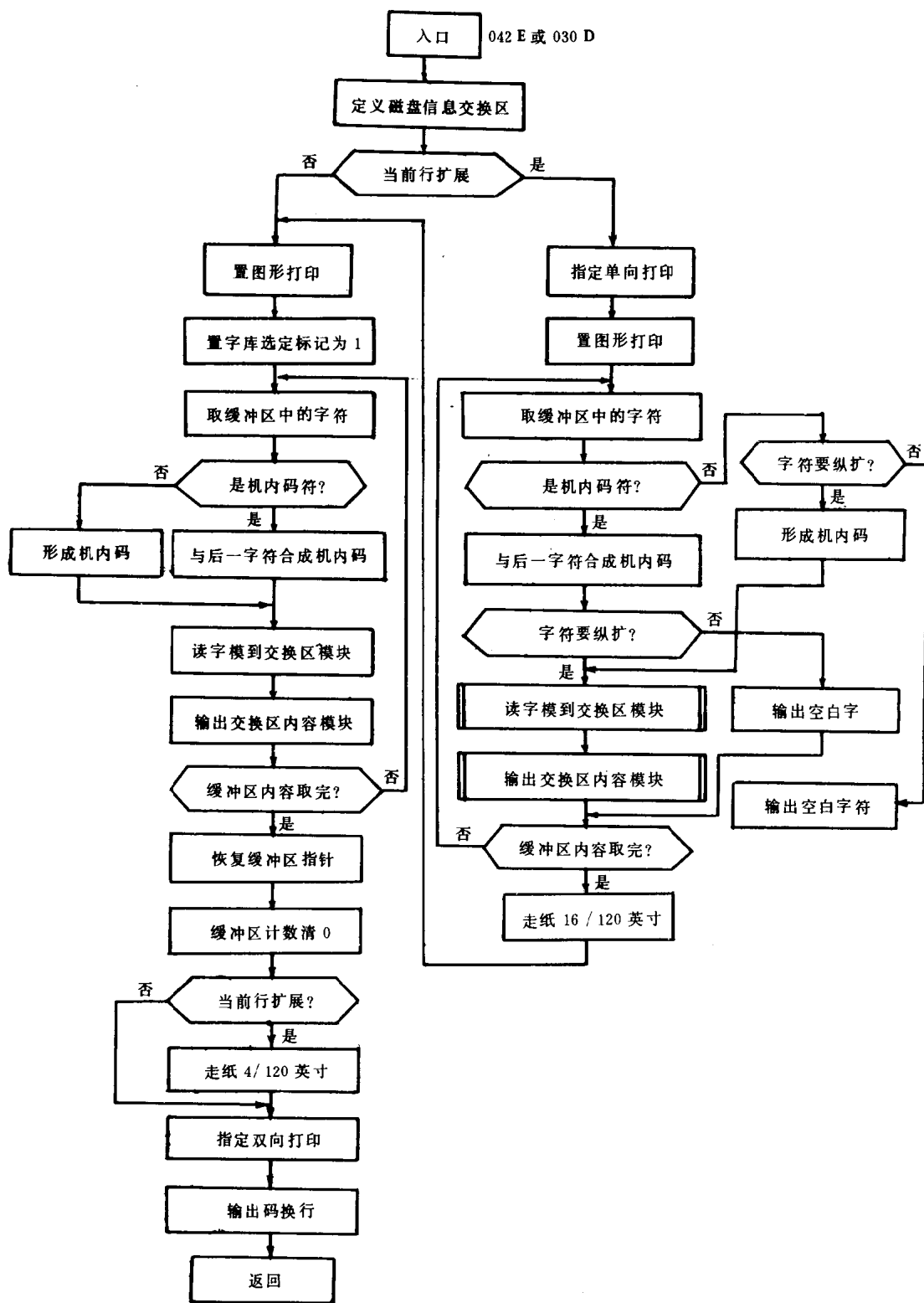


图 8 - 8 输出缓冲区内容模块流程

本文是根据CC-DOS (V2.1) 做上述分析的, 其他版本的CC-DOS 中, 这部分内容可能会有所不同。以上分析的模块是针对TH-3070 24针打印机的, 读者不准把它改成适用于其他24针打印机的驱动模块。一般说来, 只要改动其中的打印机控制命令部分即可。

通过以上分析可知道, 高级打印驱动模块虽然能打印出高质量的24×24点阵的汉字, 但是随之也带来一些缺点。因为它的字库信息量大, 故不能常驻内存, 而只能存在外存上。从而, 每取一个字模均要访问一次外存, 这势必影响打印驱动模块的运行速度。另外高级打印驱动模块不具备打印屏幕内容的功能。高级打印驱动模块常被一些需要打印出高质量汉字的应用程序所选用。CCDOS 2.13是CCDOS 2.0/2.1改进版, 24点阵打印效果显著提高。

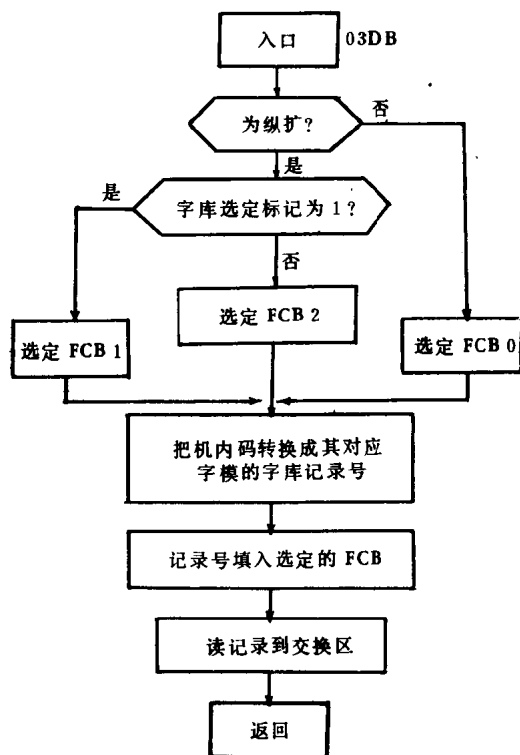


图 8-9 读字模到交换区模块流程

8.8 CC-BIOS 高级打印驱动模块 (一) (TH-3070打印机用)

24点阵打印驱动程序必需适配24点阵字库与硬件机型方能正常运行。若字库或机型不符, 常发生死机或打印错码等现象。

后面列出两种不同的打印机24点阵驱动程序文本差别:

	D320.exe	D32024.exe
执行指令入口	CS:0BFA	CS:OAD2
新INT 17-H入口	CS:0183	CS:0083
0号功能入口	CS:020F	CS:0104
1号功能入口	CS:01EE	CS:00EE
2号功能入口	CS:01E7	CS:00E7
3号功能入口	CS:0171	CS:0071
新INT 21H入口	CS:013D	CS:003D
打印机数据区	CS:105-11B	CS:0005-001B
INT17H程序数据区	CS:0771-07D6	CS:0649-06A8
机内码处理入口	CS:0315	CS:020D
缓冲区处理入口	CS:0353	CS:0232
读字模到交换区	CS:0510	CS:03DB
输出缓冲区内容	CS:042E	CS:030D
8.4节所给的INT 17程序工作区, 接近于D32024.exe。		
控制标志字节		CS:0699
控制属性字节		CS:069A

-r

AX=0000 BX=0000 CX=0A2A DX=0000 SP=FFFE BP=0000 SI=0000 DI=0000
 DS=4DF8 ES=4DF8 SS=4DF8 CS=4DF8 IP=0100 NV UP DI PL NZ NA PD NC
 4DF8:0100 BEFA0B MOV SI,0BFA

-d ds:00

4DF8:0000	CD 20 00 80 00 9A F0 FF-0D F0 42 02 F3 4A 70 02	Mp..pB.sJp.
4DF8:0010	F3 4A E2 04 42 05 F3 4A-01 01 01 00 02 FF FF FF	sJb.B.sJ.....
4DF8:0020	FF FF FF FF FF FF FF FF-FF FF FF FF F5 4D CB 2AuMH*
4DF8:0030	F3 4A 00 00 00 00 00 00-00 00 00 00 00 00 00	sJ.....
4DF8:0040	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00
4DF8:0050	CD 21 CB 00 00 00 00 00-00 00 00 00 00 20 20 20	M!K.....
4DF8:0060	20 20 20 20 20 20 20 20-00 00 00 00 00 20 20 20
4DF8:0070	20 20 20 20 20 20 20 20-00 00 00 00 00 00 00 00

段前缀 (PSP)

-d

4DF8:0080	00 0D 64 33 32 30 2E 65-7B 65 0D 0D 78 65 00 0D	..d320.exe..xe..
4DF8:0090	49 54 49 4E 47 0D 00 00-00 00 00 00 00 00 00 00	ITING.....
4DF8:00A0	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
4DF8:00B0	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
4DF8:00C0	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
4DF8:00D0	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
4DF8:00E0	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00sJp.
4DF8:00F0	00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00

键盘缓冲区

-u

4DF8:0100	BEFA0B	MOV	SI,0BFA	
4DF8:0103	FFE6	JMP	SI	
4DF8:0105	0010	ADD	[BX+SI],DL	: 工作区(打印状态字单元)
4DF8:0107	0000	ADD	[BX+SI],AL	
4DF8:0109	0000	ADD	[BX+SI],AL	
4DF8:010B	0000	ADD	[BX+SI],AL	: 打印口地址单元
4DF8:010D	0000	ADD	[BX+SI],AL	
4DF8:010F	0000	ADD	[BX+SI],AL	
4DF8:0111	0000	ADD	[BX+SI],AL	
4DF8:0113	0000	ADD	[BX+SI],AL	
4DF8:0115	0000	ADD	[BX+SI],AL	
4DF8:0117	0000	ADD	[BX+SI],AL	
4DF8:0119	0000	ADD	[BX+SI],AL	
4DF8:011B	0000	ADD	[BX+SI],AL	
4DF8:011D	80F905	CMF	CL,05	

-u

4DF8:0120	740B	JZ	012D
4DF8:0122	2E	CS:	
4DF8:0123	FF360F01	PUSH	[010F]
4DF8:0127	2E	CS:	
4DF8:0128	FF361B01	PUSH	[011B]
4DF8:012C	CB	RETF	
4DF8:012D	5B	POP	AX
4DF8:012E	5B	POP	AX

4DF8:012F 2E	CS:		
4DF8:0130 8F061901	POP	[0119]	
4DF8:0134 9C	PUSHF		
4DF8:0135 50	PUSH	AX	
4DF8:0136 2E	CS:		
4DF8:0137 FF361901	PUSH	[0119]	
4DF8:013B 8AE1	MOV	AH,CL	
4DF8:013D 80FC05	CMP	AH,05	: 新INT 21H 中断入口CS : 013 D
-u			
4DF8:0140 740B	JZ	014D	
4DF8:0142 2E	CS:		
4DF8:0143 FF361701	PUSH	[0117]	
4DF8:0147 2E	CS:		
4DF8:0148 FF361501	PUSH	[0115]	
4DF8:014C CB	RETF		
4DF8:014D 50	PUSH	AX	
4DF8:014E 52	PUSH	DX	
4DF8:014F BAC2	MOV	AL,DL	
4DF8:0151 32E4	XOR	AH,AH	
4DF8:0153 33D2	XOR	DX,DX	
4DF8:0155 CD17	INT	17	
4DF8:0157 5A	POP	DX	
4DF8:0158 5B	POP	AX	
4DF8:0159 CF	IRET		
4DF8:015A 32E4	XOR	AH,AH	: 17类中断打印公用子程入口
4DF8:015C 9C	PUSHF		CS : 015A
4DF8:015D 9AD2EF00F0	CALL	F000:EFD2	: ROM BIOS INT 17H 调用
-u			
4DF8:0162 A30501	MOV	[0105],AX	: 保留状态字
4DF8:0165 C3	RET		
4DF8:0166 5B	POP	AX	
4DF8:0167 9C	PUSHF		
4DF8:0168 0E	PUSH	CS	
4DF8:0169 1F	POP	DS	
4DF8:016A FF1E1101	CALL	FAR [0111]	
4DF8:016E EB6E	JMP	01DE	
4DF8:0170 90	NOP		
4DF8:0171 B21B	MOV	DL,1B	: 17类中断 3 号功能入口=CS : 0171
4DF8:0173 F6E2	MUL	DL	
4DF8:0175 3DBF09	CMP	AX,098F	
4DF8:0178 7203	JB	017D	
4DF8:017A B88F09	MOV	AX,098F	
4DF8:017D A3CC07	MOV	[07CC],AX	
4DF8:0180 EB47	JMP	01C9	
-u			
4DF8:0182 90	NOP		
4DF8:0183 FB	STI		: 17类中断主体入口=CS : 183
4DF8:0184 06	PUSH	ES	INT 16H 定义打印机结束调用

4DF8:0185 1E	PUSH	DS
4DF8:0186 52	PUSH	DX
4DF8:0187 56	PUSH	SI
4DF8:0188 51	PUSH	CX
4DF8:0189 53	PUSH	BX
4DF8:018A 57	PUSH	DI
4DF8:018B 55	PUSH	BP
4DF8:018C 50	PUSH	AX
4DF8:018D 33ED	XOR	BP, BP
4DF8:018F B8DD	MOV	DS, BP
4DF8:0191 2E	CS:	
4DF8:0192 A10F01	MOV	AX, [010F]
4DF8:0195 2E	CS:	
4DF8:0196 BB360D01	MOV	SI, [010D]
4DF8:019A B8DB	MOV	DS, AX
4DF8:019C AD	LDSW	
4DF8:019D 2E	CS:	
4DF8:019E A30701	MOV	[0107], AX
4DF8:01A1 AD	LDSW	
-u		
4DF8:01A2 2E	CS:	
4DF8:01A3 A30901	MOV	[0109], AX
4DF8:01A6 2E	CS:	
4DF8:01A7 BB360B01	MOV	SI, [010B]
4DF8:01AB AC	LDSB	
4DF8:01AC A8FF	TEST	AL, FF
4DF8:01AE 7402	JZ	01B2
4DF8:01B0 EBB4	JMP	0166
4DF8:01B2 5B	POP	AX
4DF8:01B3 BCCD	MOV	BP, CS
4DF8:01B5 B8DD	MOV	DS, BP
4DF8:01B7 B8C5	MOV	ES, BP
4DF8:01B9 0AE4	OR	AH, AH
4DF8:01BB 7452	JZ	020F
4DF8:01BD FECC	DEC	AH
4DF8:01BF 742D	JZ	01EE
4DF8:01C1 FECC	DEC	AH
-u		
4DF8:01C3 7422	JZ	01E7
4DF8:01C5 FECC	DEC	AH
4DF8:01C7 74A8	JZ	0171
4DF8:01C9 8B1E0F01	MOV	BX, [010F]
4DF8:01CD B8C3	MOV	ES, BX
4DF8:01CF 8B3E0D01	MOV	DI, [010D]
4DF8:01D3 A10701	MOV	AX, [0107]
4DF8:01D6 AB	STOSW	
4DF8:01D7 A10901	MOV	AX, [0109]
4DF8:01DA AB	STOSW	

测试功能调用

- : AH = 0 转 0 号功能
- CS : 020 F, 向打印机输出字符
- : AH = 1 转 1 号功能
- CS : 01E E 打印机初始化
- : AH = 2 转 2 号功能
- CS : 01E 7 取打印机状态字

- : AH = 3 转 3 号功能
- CS : 0171 确定打印机行宽

4DF8:01DB A10501	MOV	AX,[0105]	
4DF8:01DE 5D	POP	BP	
4DF8:01DF 5F	POP	DI	
4DF8:01E0 5B	POP	BX	
4DF8:01E1 59	POP	CX	
4DF8:01E2 5E	POP	SI	
-u			
4DF8:01E3 5A	POP	DX	
4DF8:01E4 1F	POP	DS	
4DF8:01E5 07	POP	ES	
4DF8:01E6 CF	IRET		
4DF8:01E7 B402	MOV	AH,02	: 2号功能入口=CS:01E7
4DF8:01E9 E970FF	CALL	015C	
4DF8:01EC E8DB	JMP	01C9	
4DF8:01EE B401	MOV	AH,01	: 1号功能入口=CS:01EE初始化码送打印
4DF8:01F0 E869FF	CALL	015C	
4DF8:01F3 B01B	MOV	AL,1B	: esc
4DF8:01F5 E862FF	CALL	015A	
4DF8:01F8 B048	MOV	AL,48	: "H"
4DF8:01FA E85DFF	CALL	015A	
4DF8:01FD B83000	MOV	AX,0030	: "0"
4DF8:0200 E859FF	CALL	015C	
-u			
4DF8:0203 B030	MOV	AL,30	: "0"
4DF8:0205 E852FF	CALL	015A	
4DF8:0208 B035	MOV	AL,35	: "5"送打印
4DF8:020A E84DFF	CALL	015A	
4DF8:020D EBBA	JMP	01C9	
4DF8:020F F606C10701	TEST	BYTE PTR [07C1],01	: 0号功能入口=CS:020F 控制标点=01,图形打印?
4DF8:0214 90	NOP		
4DF8:0215 7416	JZ	022D	
4DF8:0217 50	PUSH	AX	
4DF8:0218 A1C707	MOV	AX,[07C7]	
4DF8:021B 48	DEC	AX	
4DF8:021C A3C707	MOV	[07C7],AX	
4DF8:021F 7506	JNZ	0227	
4DF8:0221 8026C107FE	AND	BYTE PTR [07C1],FE	: 控制状态取高7位
-u			
4DF8:0226 90	NOP		
4DF8:0227 58	POP	AX	
4DF8:0228 E82FFF	CALL	015A	
4DF8:022B EB9C	JMP	01C9	
4DF8:022D F606C10720	TEST	BYTE PTR [07C1],20	: H态? (打印页长)
4DF8:0232 90	NOP		
4DF8:0233 7403	JZ	0238	
4DF8:0235 E97C04	JMP	06B4	
4DF8:0238 F606C10702	TEST	BYTE PTR [07C1],02	: 机内码?
4DF8:023D 90	NOP		

4DF8:023E 7403	JZ	0243	
4DF8:0240 E9D200	JMP	0315	
4DF8:0243 F606C10704	TEST	BYTE PTR [07C1],04	: esc ?
-u			
4DF8:024B 90	NOP		
4DF8:0249 747B	JZ	02C6	
4DF8:024B F606C1070B	TEST	BYTE PTR [07C1],0B	: esc + "I" ?
4DF8:0250 90	NOP		
4DF8:0251 744E	JZ	02A1	
4DF8:0253 3C41	CMP	AL,41	: "A" ? I 态处理模块入口
4DF8:0255 7207	JB	025E	CS : 0253
4DF8:0257 3C50	CMP	AL,50	: "Q" ?
4DF8:0259 7703	JA	025E	
4DF8:025B E95201	JMP	03B0	
4DF8:025E 50	PUSH	AX	
4DF8:025F B01B	MOV	AL,1B	: esc
4DF8:0261 E9F6FE	CALL	015A	
4DF8:0264 B049	MOV	AL,49	: "I" ?
4DF8:0266 EBF1FE	CALL	015A	
-u			
4DF8:0269 5B	POP	AX	
4DF8:026A F606C10710	TEST	BYTE PTR [07C1],10	
4DF8:026F 90	NOP		
4DF8:0270 740F	JZ	02B1	
4DF8:0272 800EC10710	OR	BYTE PTR [07C1],10	
4DF8:0277 90	NOP		
4DF8:027B A2C407	MOV	[07C4],AL	
4DF8:027B E8DCFE	CALL	015A	
4DF8:027E E948FF	JMP	01C9	
4DF8:0281 8026C107E3	AND	BYTE PTR [07C1],E3	
4DF8:0286 90	NOP		
4DF8:0287 800EC10701	OR	BYTE PTR [07C1],01	
-u			
4DF8:028C 90	NOP		
4DF8:028D 50	PUSH	AX	
4DF8:028E 8A26C407	MOV	AH,[07C4]	
4DF8:0292 590300	MOV	CX,0003	
4DF8:0295 F7E1	MUL	CX	
4DF8:0297 A3C707	MOV	[07C7],AX	
4DF8:029A 5B	POP	AX	
4DF8:029B E8BCFE	CALL	015A	
4DF8:029E E928FF	JMP	01C9	
4DF8:02A1 3C49	CMP	AL,49	: "I" ?
4DF8:02A3 7509	JNZ	02AE	
4DF8:02A5 800EC10708	OR	BYTE PTR [07C1],08	
4DF8:02AA 90	NOP		
4DF8:02AB EB14	JMP	02C1	
-u			

4DF8:02AD 90	NOF		
4DF8:02AE 3C48	CMP	AL,48	: "H" ?
4DF8:02B0 7509	JNZ	02BB	
4DF8:02B2 800EC10720	OR	BYTE PTR [07C1],20	
4DF8:02B7 90	NOF		
4DF8:02B8 EB07	JMP	02C1	
4DF8:02BA 90	NOF		
4DF8:02BB 8026C10780	AND	BYTE PTR [07C1],80	
4DF8:02C0 90	NOF		
4DF8:02C1 33C0	XOR	AX,AX	
4DF8:02C3 E903FF	JMP	01C9	
4DF8:02C6 3C1B	CMP	AL,1B	: 非 esc 处理模块入口
4DF8:02C8 750B	JNZ	02D5	
4DF8:02CA 800EC10704	OR	BYTE PTR [07C1],04	
-u			
4DF8:02CF 90	NOF		
4DF8:02D0 33C0	XOR	AX,AX	
4DF8:02D2 E9FAFE	JMP	01C9	
4DF8:02D5 3CA0	CMP	AL,A0	
4DF8:02D7 772E	JA	0307	
4DF8:02D9 3C0A	CMP	AL,0A	: 换行?
4DF8:02DB 7503	JNZ	02E0	
4DF8:02DD E93301	JMP	0413	
4DF8:02E0 3C0C	CMP	AL,0C	: 换页?
4DF8:02E2 7503	JNZ	02E7	
4DF8:02E4 E97603	JMP	065D	
4DF8:02E7 3C0D	CMP	AL,0D	: 回车?
4DF8:02E9 7506	JNZ	02F1	
4DF8:02EB EB6CFE	CALL	015A	
4DF8:02EE E9D8FE	JMP	01C9	
-u			
4DF8:02F1 247F	AND	AL,7F	
4DF8:02F3 EB5D00	CALL	0353	
4DF8:02F6 AA	STOSB		
4DF8:02F7 A0C207	MOV	AL,[07C2]	
4DF8:02FA AA	STOSB		
4DF8:02FB 893EC507	MOV	[07C5],DI	
4DF8:02FF EB9C00	CALL	039E	
4DF8:0302 33C0	XOR	AX,AX	
4DF8:0304 E9C2FE	JMP	01C9	
4DF8:0307 A2C407	MOV	[07C4],AL	
4DF8:030A 800EC10702	OR	BYTE PTR [07C1],02	
4DF8:030F 90	NOF		
4DF8:0310 33C0	XOR	AX,AX	
-u			
4DF8:0312 E984FE	JMP	01C9	
4DF8:0315 8026C107FD	AND	BYTE PTR [07C1],FD	: 机内码处理入口

4DF8:031A 90	NOP	
4DF8:031B 8A26C407	MOV	AH,[07C4]
4DF8:031F 3CA1	CMP	AL,A1
4DF8:0321 722B	JB	034E
4DF8:0323 3CFF	CMP	AL,FF
4DF8:0325 7427	JZ	034E
4DF8:0327 80FCF8	CMP	AH,F8
4DF8:032A 7322	JNB	034E
4DF8:032C 80FCA1	CMP	AH,A1
4DF8:032F 721D	JB	034E
4DF8:0331 50	PUSH	AX
-u		
4DF8:0332 E84E00	CALL	0353
4DF8:0335 58	POP	AX
4DF8:0336 AA	STOSB	
4DF8:0337 A0C207	MOV	AL,[07C2]
4DF8:033A AA	STOSB	
4DF8:033B 8AC4	MOV	AL,AH
4DF8:033D AA	STOSB	
4DF8:033E A0C207	MOV	AL,[07C2]
4DF8:0341 AA	STOSB	
4DF8:0342 893EC507	MOV	[07C5],DI
4DF8:0346 E85500	CALL	039E
4DF8:0349 33C0	XOR	AX,AX
4DF8:034B E97BFE	JMP	01C9
4DF8:034E 88A1A1	MOV	AX,A1A1
4DF8:0351 EBDE	JMP	0331
-u		
4DF8:0353 880EC907	MOV	CX,[07C9] ; 缓冲区处理入口
4DF8:0357 A8B0	TEST	AL,80
4DF8:0359 7528	JNZ	0383
4DF8:035B B30C	MOV	BL,0C
4DF8:035D F606C20701	TEST	BYTE PTR [07C2],01
4DF8:0362 90	NOP	
4DF8:0363 7402	JZ	0367
4DF8:0365 B318	MOV	BL,18
4DF8:0367 32FF	XOR	BH,BH
4DF8:0369 03CB	ADD	CX,BX
4DF8:036B 3B0ECC07	CMP	CX,[07CC]
4DF8:036F 7221	JB	0392
4DF8:0371 7423	JZ	0396
-u		
4DF8:0373 53	PUSH	BX
4DF8:0374 50	PUSH	AX
4DF8:0375 E8B600	CALL	042E
4DF8:0378 58	POP	AX
4DF8:0379 5B	POP	BX
4DF8:037A 891EC907	MOV	[07C9],BX

4DFB:037E 8B3EC507	MOV	DI,[07C5]	
4DFB:0382 C3	RET		
4DFB:0383 BB1800	MOV	BX,0018	
4DFB:0386 F606C20701	TEST	BYTE PTR [07C2],01	
4DFB:038B 90	NOP		
4DFB:038C 74D9	JZ	0367	
4DFB:038E B330	MOV	BL,30	
4DFB:0390 EBD5	JMP	0367	
4DFB:0392 8ED9	MOV	BX,CX	
-v			
4DFB:0394 EBE4	JMP	037A	
4DFB:0396 C606CB07FF	MOV	BYTE PTR [07CB],FF	
4DFB:039B 90	NOP		
4DFB:039C EBF4	JMP	0392	
4DFB:039E 803ECB07FF	CMP	BYTE PTR [07CB],FF	
4DFB:03A3 90	NOP		
4DFB:03A4 7503	JNZ	03A9	
4DFB:03A6 E88500	CALL	042E	
4DFB:03A9 C606CB0700	MOV	BYTE PTR [07CB],00	
4DFB:03AE 90	NOP		
4DFB:03AF C3	RET		
4DFB:03B0 C606C20700	MOV	BYTE PTR [07C2],00	
-u			
4DFB:03B5 90	NOP		
4DFB:03B6 3C41	CMP	AL,41	: "A" ?
4DFB:03B8 7503	JNZ	03BD	
4DFB:03BA EB4C	JMP	0408	: 是 "A" 转408
4DFB:03BC 90	NOP		
4DFB:03BD 3C42	CMP	AL,42	: "B "
4DFB:03BF 7509	JNZ	03CA	
4DFB:03C1 800EC20701	OR	BYTE PTR [07C2],01	
4DFB:03C6 90	NOP		
4DFB:03C7 EB3F	JMP	0408	
4DFB:03C9 90	NOP		
4DFB:03CA 3C43	CMP	AL,43	: "C "
4DFB:03CC 750F	JNZ	03DD	
4DFB:03CE 800EC20702	OR	BYTE PTR [07C2],02	
4DFB:03D3 90	NOP		
4DFB:03D4 800EC10780	OR	BYTE PTR [07C1],80	
-u			
4DFB:03D9 90	NOP		
4DFB:03DA EB2C	JMP	0408	
4DFB:03DC 90	NOP		
4DFB:03DD 3C44	CMP	AL,44	
4DFB:03DF 750F	JNZ	03F0	
4DFB:03E1 800EC20703	OR	BYTE PTR [07C2],03	
4DFB:03E6 90	NOP		
4DFB:03E7 800EC10780	OR	BYTE PTR [07C1],80	
4DFB:03EC 90	NOP		

```

4DF8:03ED EB19      JMP      0408
4DF8:03EF 90        NOP
4DF8:03F0 3C48      CMP      AL,48
4DF8:03F2 770A      JA       03FE
4DF8:03F4 800EC20704    OR       BYTE PTR [07C2],04
-u
4DF8:03F9 90        NOP
4DF8:03FA 2504      SUB      AL,04
4DF8:03FC EB88      JMP      03B6
4DF8:03FE 3C49      CMP      AL,49
4DF8:0400 7506      JNZ     0408
4DF8:0402 800EC20780    OR       BYTE PTR [07C2],80
4DF8:0407 90        NOP
4DF8:0408 33C0      XOR      AX,AX
4DF8:040A 8026C10780    AND      BYTE PTR [07C1],80
4DF8:040F 90        NOP
4DF8:0410 E9B6FD      JMP      01C9
4DF8:0413 833EC90700    CMP      WORD PTR [07C9],+00
4DF8:0418 7509      JNZ     0423
-u
4DF8:041A EB5A02      CALL    0677
4DF8:041D EB3AFD      CALL    015A
4DF8:0420 E9A6FD      JMP      01C9
4DF8:0423 EB6402      CALL    068A
4DF8:0426 EB0500      CALL    042E
4DF8:0429 33C0      XOR      AX,AX
4DF8:042B E99BFD      JMP      01C9
4DF8:042E 52        PUSH    DX
4DF8:042F BA7107      MOV     DX,0771
4DF8:0432 B41A      MOV     AH,1A
4DF8:0434 CD21      INT     21
4DF8:0436 5A        POP     DX
4DF8:0437 8B3EC507    MOV     DI,[07C5]
-u
4DF8:043B BED007      MOV     SI,07D0
4DF8:043E 8936C507    MOV     [07C5],SI
4DF8:0442 F606C10780    TEST    BYTE PTR [07C1],80
4DF8:0447 90        NOP
4DF8:0448 744E      JZ      0498
4DF8:044A B33E      MOV     BL,3E
4DF8:044C EBCE01      CALL    061D
4DF8:044F EBB201      CALL    0604
4DF8:0452 AD        LODSW
4DF8:0453 A880      TEST    AL,80
4DF8:0455 7420      JZ      0477
4DF8:0457 8ADB      MOV     BL,AL
4DF8:0459 AD        LODSW
4DF8:045A 8AF8      MOV     BH,AL

```

: 输出缓冲内容模块

: ">" 单向打印符
: 送打印机

```

-u
4DF8:045C F6C402    TEST    AH,02
4DF8:045F 7506      JNZ     0467
4DF8:0461 E87301    CALL    05D7
4DF8:0464 E807      JMP     046D
4DF8:0466 90        NOP
4DF8:0467 E8A600    CALL    0510      ; 读字模到交换区
4DF8:046A E8DF00    CALL    054C
4DF8:046D 3BF7      CMP     SI,DI
4DF8:046F 75E1      JNZ     0452
4DF8:0471 E8B401    CALL    0628
4DF8:0474 E822      JMP     0498
4DF8:0476 90        NOP
4DF8:0477 F6C402    TEST    AH,02
4DF8:047A 7505      JNZ     0481

-u
4DF8:047C E87701    CALL    05F6
4DF8:047F EBEC      JMP     046D
4DF8:0481 3C20      CMP     AL,20
4DF8:0483 7505      JNZ     048A
4DF8:0485 E86E01    CALL    05F6
4DF8:0488 EBEB      JMP     046D
4DF8:048A B7A3      MOV     BH,A3
4DF8:048C 0CB0      OR      AL,B0
4DF8:048E BAD8      MOV     BL,AL
4DF8:0490 E87D00    CALL    0510
4DF8:0493 E82B01    CALL    05C1
4DF8:0496 EBD5      JMP     046D
4DF8:0498 E86901    CALL    0604
4DF8:049B E8D007    MOV     SI,07D0

-u
4DF8:049E C606C30701  MOV     BYTE PTR [07C3],01
4DF8:04A3 90        NOP
4DF8:04A4 AD        LODSW
4DF8:04A5 A8B0      TEST    AL,B0
4DF8:04A7 7446      JZ      04EF
4DF8:04A9 BAD8      MOV     BL,AL
4DF8:04AB AD        LODSW
4DF8:04AC 8AF8      MOV     BH,AL
4DF8:04AE E85F00    CALL    0510
4DF8:04B1 E89B00    CALL    054C
4DF8:04B4 3BF7      CMP     SI,DI
4DF8:04B6 75EC      JNZ     04A4
4DF8:04B8 E8D007    MOV     SI,07D0
4DF8:04BB 8936C507  MOV     [07C5],SI

-u
4DF8:04BF C706C9070000  MOV     WORD PTR [07C9],0000
4DF8:04C5 F606C10780  TEST    BYTE PTR [07C1],B0

```

4DF8:04CA 90	NOP	
4DF8:04CB 7403	JZ	04D0
4DF8:04CD EB6101	CALL	0631
4DF8:04D0 F606C20702	TEST	BYTE PTR [07C2],02
4DF8:04D5 90	NOP	
4DF8:04D6 7506	JNZ	04DE
4DF8:04DB 8026C1077F	AND	BYTE PTR [07C1],7F
4DF8:04DD 90	NOP	
4DF8:04DE C606C30700	MOV	BYTE PTR [07C3],00

-u

4DF8:04E3 90	NOP	
4DF8:04E4 B33C	MOV	BL,3C
4DF8:04E6 EB3401	CALL	061D
4DF8:04E9 B00A	MOV	AL,0A
4DF8:04EB EB6CFC	CALL	015A
4DF8:04EE C3	RET	
4DF8:04EF 3C20	CMP	AL,20
4DF8:04F1 7505	JNZ	04F8
4DF8:04F3 EB0001	CALL	05F6
4DF8:04F6 EBBC	JMP	04B4
4DF8:04F8 F6C480	TEST	AH,80
4DF8:04FB 750E	JNZ	050B
4DF8:04FD B7A3	MOV	BH,A3
4DF8:04FF 0C80	OR	AL,80
4DF8:0501 BADA	MOV	BL,AL

: 换行

-u

4DF8:0503 EB0A00	CALL	0510
4DF8:0506 EB8800	CALL	05C1
4DF8:0509 EBA7	JMP	04B4
4DF8:050B EBC200	CALL	05D0
4DF8:050E EBA4	JMP	04B4
4DF8:0510 50	PUSH	AX
4DF8:0511 52	PUSH	DX
4DF8:0512 F6C402	TEST	AH,02
4DF8:0515 7506	JNZ	051D
4DF8:0517 BA0207	MOV	DX,0702
4DF8:051A EB0F	JMP	052B
4DF8:051C 90	NOP	
4DF8:051D BA2707	MOV	DX,0727
4DF8:0520 F606C30701	TEST	BYTE PTR [07C3],01

: 读字模到交换区模块入口

-u

4DF8:0525 90	NOP	
4DF8:0526 7403	JZ	052B
4DF8:0528 BA4C07	MOV	DX,074C
4DF8:052B 87DA	XCHG	DX,BX
4DF8:052D 81E27F7F	AND	DX,7F7F
4DF8:0531 81EA2121	SUB	DX,2121
4DF8:0535 B05E	MOV	AL,5E

4DF8:0537 F6E6	MUL	DH
4DF8:0539 32F6	XOR	DH,DH
4DF8:053B.03C2	ADD	AX,DX
4DF8:053D 894721	MOV	[BX+21],AX
4DF8:0540 87DA	XCHG	DX,BX
4DF8:0542 B427	MOV	AH,27
4DF8:0544 890100	MOV	CX,0001
-u		
4DF8:0547 CD21	INT	21
4DF8:0549 5A	POP	BX
4DF8:054A 58	POP	AX
4DF8:054B C3	RET	
4DF8:054C 8AFC	MOV	BH,AH
4DF8:054E 56	PUSH	SI
4DF8:054F B91800	MOV	CX,0018
4DF8:0552 F6C401	TEST	AH,01
4DF8:0555 7403	JZ	055A
4DF8:0557 893000	MOV	CX,0030
4DF8:055A BE7107	MOV	SI,0771
4DF8:055D 56	PUSH	SI
4DF8:055E AC	LDSB	
4DF8:055F F6C704	TEST	BH,04
4DF8:0562 7403	JZ	0567
4DF8:0564 0A4402	OR	AL,[SI+02]
-u		
4DF8:0567 EBFOFB	CALL	015A
4DF8:056A AC	LDSB	
4DF8:056B F6C704	TEST	BH,04
4DF8:056E 7403	JZ	0573
4DF8:0570 0A4402	OR	AL,[SI+02]
4DF8:0573 E8E4FB	CALL	015A
4DF8:0576 AC	LDSB	
4DF8:0577 F6C704	TEST	BH,04
4DF8:057A 7403	JZ	057F
4DF8:057C 0A4402	OR	AL,[SI+02]
4DF8:057F E8D8FB	CALL	015A
4DF8:0582 5E	POP	SI
4DF8:0583 F6C701	TEST	BH,01
4DF8:0586 7510	JNZ	0598
-u		
4DF8:0588 83C603	ADD	SI,+03
4DF8:058B F6C780	TEST	BH,80
4DF8:058E 7403	JZ	0593
4DF8:0590 83C603	ADD	SI,+03
4DF8:0593 E2C8	LOOP	055D
4DF8:0595 EB28	JMP	058F
4DF8:0597 90	NOP	
4DF8:0598 49	DEC	CX

4DF8:0599 AC	LDSB	
4DF8:059A F6C704	TEST	BH,04
4DF8:059D 7403	JZ	05A2
4DF8:059F 0A4402	OR	AL,[SI+02]
4DF8:05A2 EBB5FB	CALL	015A
4DF8:05A5 AC	LDSB	
4DF8:05A6 F6C704	TEST	BH,04
-u		
4DF8:05A9 7403	JZ	05AE
4DF8:05AB 0A4402	OR	AL,[SI+02]
4DF8:05AE EBA9FB	CALL	015A
4DF8:05B1 AC	LDSB	
4DF8:05B2 F6C704	TEST	BH,04
4DF8:05B5 7403	JZ	05BA
4DF8:05B7 0A4402	OR	AL,[SI+02]
4DF8:05BA EB9DFB	CALL	015A
4DF8:05BD E29E	LOOP	055D
4DF8:05BF 5E	POP	SI
4DF8:05C0 C3	RET	
4DF8:05C1 56	PUSH	SI
4DF8:05C2 B91800	MOV	CX,0018
4DF8:05C5 F6C401	TEST	AH,01
4DF8:05C8 7506	JNZ	05D0
-u		
4DF8:05CA B90C00	MOV	CX,000C
4DF8:05CD 80CC84	OR	AH,B4
4DF8:05D0 80E4FE	AND	AH,FE
4DF8:05D3 8AFC	MOV	BH,AH
4DF8:05D5 EB63	JMP	053A
4DF8:05D7 50	PUSH	AX
4DF8:05D8 B91800	MOV	CX,0018
4DF8:05DB F6C401	TEST	AH,01
4DF8:05DE 7403	JZ	05E3
4DF8:05E0 B93000	MOV	CX,0030
4DF8:05E3 32C0	XOR	AL,AL
4DF8:05E5 EB72FB	CALL	015A
4DF8:05E8 32C0	XOR	AL,AL
-u		
4DF8:05EA EB6DFB	CALL	015A
4DF8:05ED 32C0	XOR	AL,AL
4DF8:05EF EB68FB	CALL	015A
4DF8:05F2 E2EF	LOOP	05E3
4DF8:05F4 58	POP	AX
4DF8:05F5 C3	RET	
4DF8:05F6 50	PUSH	AX
4DF8:05F7 B91800	MOV	CX,0018
4DF8:05FA F6C401	TEST	AH,01
4DF8:05FD 75E4	JNZ	05E3

4DF8:05FF B90C00	MOV	CX,000C	
4DF8:0602 E8DF	JMP	05E3	
4DF8:0604 BB0EC907	MOV	CX,[07C9]	
4DF8:060B B01B	MOV	AL,1B	: esc
-u			
4DF8:060A E84DFB	CALL	015A	: "I "
4DF8:060D B049	MOV	AL,49	
4DF8:060F E848FB	CALL	015A	
4DF8:0612 8AC5	MOV	AL,CH	
4DF8:0614 E843FB	CALL	015A	
4DF8:0617 8AC1	MOV	AL,CL	
4DF8:0619 E83EFB	CALL	015A	
4DF8:061C C3	RET		
4DF8:061D B01B	MOV	AL,1B	
4DF8:061F E838FB	CALL	015A	
4DF8:0622 8AC3	MOV	AL,BL	
4DF8:0624 E833FB	CALL	015A	
4DF8:0627 C3	RET		
4DF8:0628 53	PUSH	BX	
4DF8:0629 BB3631	MOV	BX,3136	
-u			
4DF8:062C E80B00	CALL	063A	
4DF8:062F 5B	POP	BX	
4DF8:0630 C3	RET		
4DF8:0631 53	PUSH	BX	
4DF8:0632 BB3430	MOV	BX,3034	
4DF8:0635 E80200	CALL	063A	
4DF8:0638 5B	POP	BX	
4DF8:0639 C3	RET		
4DF8:063A B01B	MOV	AL,1B	
4DF8:063C E818FB	CALL	015A	
4DF8:063F B057	MOV	AL,57	
4DF8:0641 E816FB	CALL	015A	
4DF8:0644 B030	MOV	AL,30	
4DF8:0646 E811FB	CALL	015A	
4DF8:0649 B030	MOV	AL,30	
4DF8:064B E80CFB	CALL	015A	
-u			
4DF8:064E 8AC7	MOV	AL,BH	
4DF8:0650 E807FB	CALL	015A	
4DF8:0653 8AC3	MOV	AL,BL	
4DF8:0655 E802FB	CALL	015A	
4DF8:0658 C3	RET		
4DF8:0659 43	INC	BX	
4DF8:065A 0000	ADD	[BX+SI],AL	
4DF8:065C 00B80E59	ADD	[BP+DI+590E],CL	
4DF8:0660 06	PUSH	ES	
4DF8:0661 2B0E5B06	SUB	CX,[065B]	

```

4DF8:0665 B00A      MOV     AL,0A
4DF8:0667 33D2      XOR     DX,DX
4DF8:0669 EBEEFA    CALL   015A
4DF8:066C E2F7      LOOP   0665

```

: 换行送打印

-u

```

4DF8:066E C7065B060000  MOV     WORD PTR [065B],0000
4DF8:0674 E952FB      JMP     01C9
4DF8:0677 8B0E5B06     MOV     CX,[065B]
4DF8:067B 41             INC     CX
4DF8:067C 3B0E5906     CMP     CX,[0659]
4DF8:0680 7203          JB      0685
4DF8:0682 33C7          XOR     CX,CX
4DF8:0684 41             INC     CX
4DF8:0685 890E5B06     MOV     [065B],CX
4DF8:0689 C3            RET
4DF8:068A 51            PUSH   CX
4DF8:068B 8B0E5B06     MOV     CX,[065B]

```

-u

```

4DF8:068F 41             INC     CX
4DF8:0690 F606C10780    TEST   BYTE PTR [07C1],80
4DF8:0695 90            NOP
4DF8:0696 7401          JZ      0699
4DF8:0698 41             INC     CX
4DF8:0699 3B0E5906     CMP     CX,[0659]
4DF8:069D 720A          JB      06A9
4DF8:069F 33C9          XOR     CX,CX
4DF8:06A1 41             INC     CX
4DF8:06A2 F606C10780    TEST   BYTE PTR [07C1],80
4DF8:06A7 90            NOP
4DF8:06A8 41             INC     CX
4DF8:06A9 890E5B06     MOV     [065B],CX
4DF8:06AD 59            POP     CX
4DF8:06AE C3            RET

```

-u

```

4DF8:06AF 0000          ADD     [BX+SI],AL
4DF8:06B1 0000          ADD     [BX+SI],AL

```

-u cs:06b4

```

4DF8:06B4 8B1EAF06     MOV     BX,[06AF]
4DF8:06B8 FEC3          INC     BL
4DF8:06BA BEB106     MOV     SI,06B1
4DF8:06BD 8B40FF     MOV     [BX+SI-01],AL
4DF8:06C0 80FB03     CMP     BL,03
4DF8:06C3 7534          JNZ     06F9
4DF8:06C5 32DB          XOR     BL,BL
4DF8:06C7 891EAF06     MOV     [06AF],BX
4DF8:06CB AC          LODSB
4DF8:06CC 2C30          SUB     AL,30
4DF8:06CE B30A          MOV     BL,0A

```

: BCD 码转换为二进制

```

-u
4DF8:06D0 F6E3      MUL    BL
4DF8:06D2 8BD0      MOV    DX,AX
4DF8:06D4 AC         LODSB
4DF8:06D5 2C30      SUB    AL,30
4DF8:06D7 32E4      XOR    AH,AH
4DF8:06D9 03D0      ADD    DX,AX
4DF8:06DB 8BC2      MOV    AX,DX
4DF8:06DD F6E3      MUL    BL
4DF8:06DF 8AC4      MOV    AL,AH
4DF8:06E1 AC         LODSB
4DF8:06E2 2C30      SUB    AL,30
4DF8:06E4 02E0      ADD    AH,AL
4DF8:06E6 88265906   MOV    [0659],AH
4DF8:06EA C7065B060000 MOV    WORD PTR [065B],0000

```

```

-u
4DF8:06F0 8026C107DB AND    BYTE PTR [07C1],DB
4DF8:06F5 90         NOP
4DF8:06F6 E9D0FA      JMP    01C9
4DF8:06F9 891EAF06   MOV    [06AF],BX
4DF8:06FD E9C9FA      JMP    01C9
4DF8:0700 0000      ADD    [BX+SI],AL
4DF8:0702 03434C   ADD    AX,[BP+DI+4C]
4DF8:0705 49         DEC    CX
4DF8:0706 42         INC    DX
4DF8:0707 3234      XOR    DH,[SI]
4DF8:0709 2020      AND    [BX+SI],AH
4DF8:070B 2020      AND    [BX+SI],AH
4DF8:070D 2000      AND    [BX+SI],AL
4DF8:070F 0000      ADD    [BX+SI],AL

```

-d csl0705

```

4DF8:0705 49 42 32-34 20 20 20 20 20 00 00
4DF8:0710 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00
4DF8:0720 00 00 00 00 00 00 00 03-43 4C 49 42 32 34 31 20
4DF8:0730 20 20 20 00 00 00 00 00-00 00 00 00 00 00 00
4DF8:0740 00 00 00 00 00 00 00 00-00 00 00 00 03 43 4C 49
4DF8:0750 42 32 34 32 20 20-20 20-00 00 00 00 00 00 00
4DF8:0760 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00
4DF8:0770 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00
4DF8:0780 00 00 00 00 00

```

-d cs:0700

```

4DF8:0700 00 00 03 43 4C 49 42 32-34 20 20 20 20 20 00 00
4DF8:0710 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00
4DF8:0720 00 00 00 00 00 00 00 03-43 4C 49 42 32 34 31 20
4DF8:0730 20 20 20 00 00 00 00 00-00 00 00 00 00 00 00
4DF8:0740 00 00 00 00 00 00 00 00-00 00 00 00 03 43 4C 49
4DF8:0750 42 32 34 32 20 20 20 20-00 00 00 00 00 00 00

```

D3 20. exe 文件控制块

```

1B24 ..
.....
.....CLIB241
.....
.....CLI
B242 .....
文件控制块.....
以下工作区.....
.....

```

```

...CLIB24 ..
.....
.....CLIB241
.....
.....CLI
B242 .....

```

```

4DF8:0760 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
4DF8:0770 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
-d
4DF8:0780 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
4DF8:0790 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
4DF8:07A0 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
4DF8:07B0 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
4DF8:07C0 00 00 00 00 00 D0 07 00-00 00 00 00 BF 09 00 00 .....P.....
4DF8:07D0 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
4DF8:07E0 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
4DF8:07F0 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....

```

-u cs:bfa

```

4DF8:0BFA 1E          PUSH  DS          ;D320 . exe 执行指令入口
4DF8:0BFB 33D2        XOR   DX,DX
4DF8:0BFD 52          PUSH  DX
4DF8:0BFE BE8200      MOV   SI,0082
4DF8:0C01 AC          LODSB
4DF8:0C02 3C31        CMP   AL,31
4DF8:0C04 7509        JNZ   0C0F
4DF8:0C06 B84315      MOV   AX,1543
4DF8:0C09 BB3C15      MOV   BX,153C
4DF8:0C0C EB14          JMP   0C22
4DF8:0C0E 90          NOP
4DF8:0C0F 3C32        CMP   AL,32
4DF8:0C11 7509        JNZ   0C1C
4DF8:0C13 B8F300      MOV   AX,00F3
4DF8:0C16 BBEE00      MOV   BX,00EE
4DF8:0C19 EB07          JMP   0C22

```

-u

```

4DF8:0C1B 90          NOP
4DF8:0C1C B8DF00      MOV   AX,00DF
4DF8:0C1F B8DA00      MOV   BX,00DA
4DF8:0C22 0E          PUSH  CS
4DF8:0C23 1F          POP   DS
4DF8:0C24 A30D01      MOV   [010D],AX
4DF8:0C27 891E0B01   MOV   [010B],BX
4DF8:0C2B B81735      MOV   AX,3517
4DF8:0C2E E8E400      CALL  0D15
4DF8:0C31 891E1101   MOV   [0111],BX
4DF8:0C35 BC061301   MOV   [0113],ES
4DF8:0C39 B82135      MOV   AX,3521

```

-u

```

4DF8:0C3C E8D600      CALL  0D15
4DF8:0C3F 891E1501   MOV   [0115],BX
4DF8:0C43 BC061701   MOV   [0117],ES
4DF8:0C47 33C0        XOR   AX,AX
4DF8:0C49 BEC0        MOV   ES,AX
4DF8:0C4B BEC100      MOV   SI,00C1

```

4DF8:0C4E 26	ES:	
4DF8:0C4F 8B04	MOV	AX,[SI]
4DF8:0C51 A31B01	MOV	[011B],AX
4DF8:0C54 26	ES:	
4DF8:0C55 8B4402	MOV	AX,[SI+02]
4DF8:0C58 8BFE	MOV	DI,SI
4DF8:0C5A A30F01	MOV	[010F],AX
-u		
4DF8:0C5D 8D061D01	LEA	AX,[011D]
4DF8:0C61 AB	STOSW	
4DF8:0C62 8CC8	MOV	AX,CS
4DF8:0C64 AB	STOSW	
4DF8:0C65 8ED8	MOV	DS,AX
4DF8:0C67 8A0207	MOV	DX,0702
4DF8:0C6A B40F	MOV	AH,0F
4DF8:0C6C CD21	INT	21
4DF8:0C6E 0AC0	OR	AL,AL
4DF8:0C70 7539	JNZ	0CAB
4DF8:0C72 BA2707	MOV	DX,0727
4DF8:0C75 B40F	MOV	AH,0F
4DF8:0C77 CD21	INT	21
4DF8:0C79 0AC0	OR	AL,AL
4DF8:0C7B 7534	JNZ	0CB1
-u		
4DF8:0C7D BA4C07	MOV	DX,074C
4DF8:0C80 B40F	MOV	AH,0F
4DF8:0C82 CD21	INT	21
4DF8:0C84 0AC0	OR	AL,AL
4DF8:0C86 752F	JNZ	0CB7
4DF8:0C88 8B4B00	MOV	AX,004B
4DF8:0C8B A31007	MOV	[0710],AX
4DF8:0C8E A33507	MOV	[0735],AX
4DF8:0C91 A35A07	MOV	[075A],AX
4DF8:0C94 BAB301	MOV	DX,01B3
4DF8:0C97 B425	MOV	AH,25
4DF8:0C99 B017	MOV	AL,17
4DF8:0C9B CD21	INT	21
-u		
4DF8:0C9D BA3D01	MOV	DX,013D
4DF8:0CA0 B021	MOV	AL,21
4DF8:0CA2 B425	MOV	AH,25
4DF8:0CA4 CD21	INT	21
4DF8:0CA6 BAFD0B	MOV	DX,0BFD
4DF8:0CA9 CD27	INT	27
4DF8:0CAB BABF0C	MOV	DX,0CBF
4DF8:0CAE EBOA	JMP	0CBA
4DF8:0CB0 90	NOP	
4DF8:0CB1 BADB0C	MOV	DX,0CDB

```

4DF8:0CB4 EB04      JMP     OCBA
4DF8:0CB6 90          NOP
4DF8:0CB7 BAFB0C     MOV     DX,OCFB
4DF8:0CBA B409     MOV     AH,09
4DF8:0CBC CD21     INT     21

-u
4DF8:0CBE CB          RETF
4DF8:0CBF 0A0D     OR      CL,[DI]      ; 0CBF 以下字符块
4DF8:0CC1 43          INC     BX
4DF8:0CC2 41          INC     CX
4DF8:0CC3 4E          DEC     SI
4DF8:0CC4 4E          DEC     SI           ^C

```

```

-d cs:cbf
4DF8:0CBF 0A
4DF8:0CC0 0D 43 41 4E 4E 4F 54 20-4F 50 45 4E 20 46 49 4C .CANNOT OPEN FIL
4DF8:0CCD 45 20 43 3A 43 4C 49 42-32 34 24 0A 0D 43 41 4E E C:CLIB24$.CAN
4DF8:0CE0 4E 4F 54 20 4F 50 45 4E-20 46 49 4C 45 20 43 3A NOT OPEN FILE C:
4DF8:0CF0 43 4C 49 42 32 34 31 24-0A 0D 43 41 4E 4E 4F 54 CLIB24$.CANNOT
4DF8:0D00 20 4F 50 45 4E 20 46 49-4C 45 20 43 3A 43 4C 49 OPEN FILE C:CLI
4DF8:0D10 42 32 34 32 24 1E 32 E4-D1 E0 D1 E0 BB F0 33 C0 B242$.2dQ`Q'.p3@
4DF8:0D20 8E D8 AD BB DB AD 8E C0-1F C3 00 00 00 00 00 00 .X-.X-.@.C.....
4DF8:0D30 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....

```

-d
D 32024.exe 是2024打印机24点阵驱动程序，它适配10片5"盘的硬盘字库，装完字库若先用2024P引导，A~P 16种字型只能打出A~C 3种，所以它不如后来改进版C C—DOS 2.13字库。

C>debug d32024.exe

```

-r
AX=0000 BX=0000 CX=0C00 DX=0000 SP=0000 BP=0000 SI=0000 DI=0000
DS=4DF8 ES=4DF8 SS=4E08 CS=4E08 IP=0000 NV UP DI PL NZ NA PO NC
4E08:0000 BED20A      MOV     SI,0AD2

-d ds:00
4DF8:0000 CD 20 00 80 00 9A F0 FF-0D F0 42 02 F3 4A 70 02 M ....p..pB.sJp.
4DF8:0010 F3 4A E2 04 42 05 F3 4A-01 01 01 00 02 FF FF FF sJb.B.sJ.....
4DF8:0020 FF FF FF FF FF FF FF FF-FF FF FF FF F5 4D CB 2A .....uMH*
4DF8:0030 F3 4A 00 00 00 00 00 00-00 00 00 00 00 00 00 sJ.....
4DF8:0040 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
4DF8:0050 CD 21 CB 00 00 00 00 00-00 00 00 00 20 20 20 M!K.....
4DF8:0060 20 20 20 20 20 20 20 20-00 00 00 00 20 20 20 .....
4DF8:0070 20 20 20 20 20 20 20 20-00 00 00 00 00 00 00 .....

程序段前缀(PSP)

```

```

-u
4E08:0000 BED20A      MOV     SI,0AD2
4E08:0003 FFE6      JMP     SI
4E08:0005 0010      ADD     [BX+SI],DL      ; 工作区(打印状态字)
4E08:0007 0000      ADD     [BX+SI],AL
4E08:0009 0000      ADD     [BX+SI],AL      ; 打印口地址单元
4E08:000B 0000      ADD     [BX+SI],AL      ; 打印机状态字单元
4E08:000D 0000      ADD     [BX+SI],AL      ; 打印数据区

```


4E08:000F 0000	ADD	[BX+SI],AL	
4E08:0011 0000	ADD	[BX+SI],AL	
4E08:0013 0000	ADD	[BX+SI],AL	
4E08:0015 0000	ADD	[BX+SI],AL	
4E08:0017 0000	ADD	[BX+SI],AL	
4E08:0019 0000	ADD	[BX+SI],AL	
4E08:001B 0000	ADD	[BX+SI],AL	
4E08:001D 80F905	CMP	CL,05	
-u			
4E08:0020 740B	JZ	002D	
4E08:0022 2E	CS:		
4E08:0023 FF360F00	PUSH	[000F]	
4E08:0027 2E	CS:		
4E08:002B FF361B00	PUSH	[001B]	
4E08:002C CB	RETF		
4E08:002D 5B	POP	AX	
4E08:002E 5B	POP	AX	
4E08:002F 2E	CS:		
4E08:0030 8F061900	POP	[0019]	
4E08:0034 9C	PUSHF		
4E08:0035 50	PUSH	AX	
4E08:0036 2E	CS:		
4E08:0037 FF361900	PUSH	[0019]	
4E08:003B 8AE1	MOV	AH,CL	
4E08:003D 80FC05	CMP	AH,05	: 新INT 21H 入口CS : 003 D
-u			
4E08:0040 740B	JZ	004D	
4E08:0042 2E	CS:		
4E08:0043 FF361700	PUSH	[0017]	
4E08:0047 2E	CS:		
4E08:004B FF361500	PUSH	[0015]	
4E08:004C CB	RETF		
4E08:004D 50	PUSH	AX	
4E08:004E 52	PUSH	DX	
4E08:004F 8AC2	MOV	AL,DL	
4E08:0051 32E4	XOR	AH,AH	
4E08:0053 33D2	XOR	DX,DX	
4E08:0055 CD17	INT	17	
4E08:0057 5A	POP	DX	
4E08:0058 5B	POP	AX	
4E08:0059 CF	IRET		
4E08:005A 32E4	XOR	AH,AH	: 打印公用子程序
4E08:005C 9C	PUSHF		
4E08:005D 9AD2EF00F0	CALL	F000:EF02	: ROM BIOS INT 17H 调用
-u			
4E08:0062 A30500	MOV	[0005],AX	
4E08:0065 C3	RET		
4E08:0066 5B	POP	AX	

4E08:0067 9C	PUSHF	
4E08:0068 0E	PUSH	CS
4E08:0069 1F	POP	DS
4E08:006A FF1E1100	CALL	FAR [0011]
4E08:006E EB6E	JMP	00DE
4E08:0070 90	NOP	
4E08:0071 B218	MOV	DL,18 : 3号功能定义打印行宽
4E08:0073 F6E2	MUL	DL
4E08:0075 3D8F09	CMP	AX,098F : 行宽最大值?
4E08:0078 7203	JB	007D
4E08:007A B8BF09	MOV	AX,098F
4E08:007D A3A406	MOV	[06A4],AX : 行宽值→06A 4
4E08:0080 EB47	JMP	00C9
-ucs:82		
4E08:0082 90	NOP	
4E08:0083 FB	STI	: 新INT 17H中断主体入口CS : 0083
4E08:0084 06	PUSH	ES (INT 16H定义打印机结束调用)
4E08:0085 1E	PUSH	DS
4E08:0086 52	PUSH	DX
4E08:0087 56	PUSH	SI
4E08:0088 51	PUSH	CX
4E08:0089 53	PUSH	BX
4E08:008A 57	PUSH	DI
4E08:008B 55	PUSH	BP
4E08:008C 50	PUSH	AX
4E08:008D 33ED	XOR	BP,BP
4E08:008F 8EDD	MOV	DS,BP
4E08:0091 2E	CS:	
4E08:0092 A10F00	MOV	AX,[000F]
4E08:0095 2E	CS:	
4E08:0096 BB360D00	MOV	SI,[000D]
4E08:009A BEDB	MOV	DS,AX
4E08:009C AD	LODSW	
4E08:009D 2E	CS:	
4E08:009E A30700	MOV	[0007],AX
4E08:00A1 AD	LODSW	
-u		
4E08:00A2 2E	CS:	
4E08:00A3 A30900	MOV	[0009],AX
4E08:00A6 2E	CS:	
4E08:00A7 8B360B00	MOV	SI,[000B]
4E08:00AB AC	LODSB	
4E08:00AC A8FF	TEST	AL,FF
4E08:00AE 7402	JZ	00B2
4E08:00B0 EBB4	JMP	0066
4E08:00B2 5B	POP	AX
4E08:00B3 BCCD	MOV	BP,CS
4E08:00B5 BEDD	MOV	DS,BP

4E08:00B7 8EC5	MOV	ES, BP	
4E08:00B9 0AE4	OR	AH, AH	: AH = 0 转 0 号功能
4E08:00BB 7447	JZ	0104	向打印机输出字符CS : 104
4E08:00BD FECC	DEC	AH	: AH = 1 转 1 号功能
4E08:00BF 742D	JZ	00EE	打印机初始化CS : 00EE
4E08:00C1 FECC	DEC	AH	: AH = 2 转 2 号功能
-u			取打印机状态字CS : 00E7
4E08:00C3 7422	JZ	00E7	
4E08:00C5 FECC	DEC	AH	: AH = 3 转 3 号功能
4E08:00C7 74AB	JZ	0071	定打印机行宽CS : 0071
4E08:00C9 8B1E0F00	MOV	BX, [000F]	: 取打印机状态数据地址
4E08:00CD 8EC3	MOV	ES, BX	
4E08:00CF 8B3E0D00	MOV	DI, [000D]	
4E08:00D3 A10700	MOV	AX, [0007]	
4E08:00D6 AB	STOSW		: 存AX → ES:DI 即 (000F) ; (000D) 处
4E08:00D7 A10900	MOV	AX, [0009]	
4E08:00DA AB	STOSW		: 存AX → ES:DI
4E08:00DB A10500	MOV	AX, [0005]	
4E08:00DE 5D	POP	BP	: 恢复寄存器
4E08:00DF 5F	POP	DI	
4E08:00E0 5B	POP	BX	
4E08:00E1 59	POP	CX	
4E08:00E2 5E	POP	SI	
-u			
4E08:00E3 5A	POP	DX	
4E08:00E4 1F	POP	DS	
4E08:00E5 07	POP	ES	
4E08:00E6 CF	IRET		
4E08:00E7 B402	MOV	AH, 02	: 2 号功能取打印机状态字
4E08:00E9 E870FF	CALL	005C	
4E08:00EC EBD8	JMP	00C9	
4E08:00EE B401	MOV	AH, 01	: 1 号功能打印机初始化
4E08:00F0 E869FF	CALL	005C	
4E08:00F3 B01B	MOV	AL, 1B	: esc
4E08:00F5 EB62FF	CALL	005A	
4E08:00F8 B042	MOV	AL, 42	: "B"
4E08:00FA E850FF	CALL	005A	
4E08:00FD B035	MOV	AL, 35	: "5"
4E08:00FF E858FF	CALL	005A	
4E08:0102 EBC5	JMP	00C9	
-u			
4E08:0104 F606990601	TEST	BYTE PTR [0699], 01	: 0 号功能AL → 打印子程入口
4E08:0109 90	NOP		打印控制标志
4E08:010A 7416	JZ	0122	
4E08:010C 50	PUSH	AX	
4E08:010D A19F06	MOV	AX, [069F]	: 是, 打印数据计数 - 1
4E08:0110 4B	DEC	AX	
4E08:0111 A39F06	MOV	[069F], AX	

4E08:0114 7506	JNZ	011C	
4E08:0116 B0269906FE	AND	BYTE PTR [0699],FE	: 控制状态字取高7位
4E08:011B 90	NOP		
4E08:011C 5B	POP	AX	: 输出打印
4E08:011D E83AFF	CALL	005A	
4E08:0120 EBA7	JMP	00C9	
4E08:0122 F606990620	TEST	BYTE PTR [0699],20	: H 态? (打印页长)
-u			
4E08:0127 90	NOP		
4E08:0128 7403	JZ	012D	
4E08:012A E95F04	JMP	058C	: BCD 换二进制码定页长处理转058 C
4E08:012D F606990602	TEST	BYTE PTR [0699],02	: 机内码?
4E08:0132 90	NOP		
4E08:0133 7403	JZ	0138	
4E08:0135 E9D500	JMP	020D	
4E08:0138 F606990604	TEST	BYTE PTR [0699],04	: esc ?
4E08:013D 90	NOP		
4E08:013E 747B	JZ	018B	
4E08:0140 F606990608	TEST	BYTE PTR [0699],08	: esc + "I" ?
4E08:0145 90	NOP		
4E08:0146 744E	JZ	0196	
-u			
4E08:0148 3C41	CMP	AL,41	: "A" ? I 态处理模块入口
4E08:014A 7207	JB	0153	
4E08:014C 3C50	CMP	AL,50	: "Q"
4E08:014E 7703	JA	0153	
4E08:0150 E93C01	JMP	028F	: 字型选择
4E08:0153 50	PUSH	AX	
4E08:0154 B01B	MOV	AL,1B	
4E08:0156 E801FF	CALL	005A	: esc
4E08:0159 B049	MOV	AL,49	
4E08:015B E8FCFE	CALL	005A	: "I"
4E08:015E 5B	POP	AX	
4E08:015F F606990610	TEST	BYTE PTR [0699],10	: 打印数据状态, 已送数据?
4E08:0164 90	NOP		
4E08:0165 740F	JZ	0176	
4E08:0167 B00E990610	OR	BYTE PTR [0699],10	
-u			
4E08:016C 90	NOP		
4E08:016D A29C06	MOV	[069C],AL	
4E08:0170 E8E7FE	CALL	005A	
4E08:0173 E953FF	JMP	00C9	: 存打印机状态数据
4E08:0176 B0269906E3	AND	BYTE PTR [0699],E3	
4E08:017B 90	NOP		
4E08:017C B00E990601	OR	BYTE PTR [0699],01	
4E08:0181 90	NOP		
4E08:0182 50	PUSH	AX	
4E08:0183 8A269C06	MOV	AH,[069C]	

4E08:0187 B90300	MOV	CX,0003	
4E08:018A F7E1	MUL	CX	
-u			
4E08:018C A39F06	MOV	[069F],AX	
4E08:018F 58	POP	AX	
4E08:0190 E8C7FE	CALL	005A	
4E08:0193 E933FF	JMP	00C9	
4E08:0196 3C49	CMP	AL,49	: "I" ?
4E08:0198 7509	JNZ	01A3	
4E08:019A 800E990608	OR	BYTE PTR [0699],08	
4E08:019F 90	NOP		
4E08:01A0 EB14	JMP	01B6	
4E08:01A2 90	NOP		
4E08:01A3 3C48	CMP	AL,48	: "H" ?
4E08:01A5 7509	JNZ	01B0	
4E08:01A7 800E990620	OR	BYTE PTR [0699],20	
-u			
4E08:01AC 90	NOP		
4E08:01AD EB07	JMP	01B6	
4E08:01AF 90	NOP		
4E08:01B0 8026990680	AND	BYTE PTR [0699],80	
4E08:01B5 90	NOP		
4E08:01B6 33C0	XOR	AX,AX	
4E08:01B8 E90EFF	JMP	00C9	
4E08:01BB 3C1B	CMP	AL,1B	: 非esc 处理模块入口
4E08:01BD 750B	JNZ	01CA	
4E08:01BF 800E990604	OR	BYTE PTR [0699],04	: 是esc . 置汉字图形标志
4E08:01C4 90	NOP		
4E08:01C5 33C0	XOR	AX,AX	
4E08:01C7 E9FFFE	JMP	00C9	
4E08:01CA 3CA0	CMP	AL,A0	: 汉字机内码?
-u			
4E08:01CC 772E	JA	01FC	
4E08:01CE 3C0A	CMP	AL,0A	: 换行?
4E08:01D0 7503	JNZ	01D5	
4E08:01D2 E91D01	JMP	02F2	
4E08:01D5 3C0C	CMP	AL,0C	: 换页?
4E08:01D7 7503	JNZ	01DC	
4E08:01D9 E95903	JMP	0535	
4E08:01DC 3C0D	CMP	AL,0D	
4E08:01DE 7506	JNZ	01E6	
4E08:01E0 E877FE	CALL	005A	
4E08:01E3 E9E3FE	JMP	00C9	
4E08:01E6 247F	AND	AL,7F	
4E08:01E8 E84700	CALL	0232	
4E08:01EB AA	STOSB		
-u			
4E08:01EC A09A06	MOV	AL,[069A]	

4E08:01EF AA	STOSB		
4E08:01F0 893E9D06	MOV	[069D],DI	
4E08:01F4 E88600	CALL	027D	
4E08:01F7 33C0	XOR	AX,AX	
4E08:01F9 E9CDFE	JMP	00C9	
4E08:01FC E83300	CALL	0232	: 缓冲区处理子程
4E08:01FF A29C06	MOV	[069C],AL	
4E08:0202 800E990602	OR	BYTE PTR [0699],02	
4E08:0207 90	NOP		
4E08:0208 33C0	XOR	AX,AX	
4E08:020A E9BCFE	JMP	00C9	
-u			
4E08:020D 80269906FD	AND	BYTE PTR [0699],FD	: 机内码处理入口
4E08:0212 90	NOP		
4E08:0213 8B3E9D06	MOV	DI,[069D]	
4E08:0217 0C80	OR	AL,80	
4E08:0219 AA	STOSB		
4E08:021A A09A06	MOV	AL,[069A]	: 属性字节
4E08:021D AA	STOSB		
4E08:021E A09C06	MOV	AL,[069C]	
4E08:0221 AA	STOSB		: 机内码字节
4E08:0222 A09A06	MOV	AL,[069A]	
4E08:0225 AA	STOSB		
4E08:0226 893E9D06	MOV	[069D],DI	: 把当前字符与属性存缓冲区
4E08:022A E85000	CALL	027D	: 缓冲区满?
-u			
4E08:022D 33C0	XOR	AX,AX	
4E08:022F E997FE	JMP	00C9	
4E08:0232 8B0EA106	MOV	CX,[06A1]	: 缓冲区处理入口, 总列数→CX
4E08:0236 A880	TEST	AL,80	: 机内码?
4E08:0238 7528	JNZ	0262	
4E08:023A B30C	MOV	BL,0C	
4E08:023C F6069A0601	TEST	BYTE PTR [069A],01	: 横扩?
4E08:0241 90	NOP		
4E08:0242 7402	JZ	0246	
4E08:0244 B318	MOV	BL,18	: 24→BL
4E08:0246 32FF	XOR	BH,BH	
4E08:0248 03CB	ADD	CX,BX	
4E08:024A 3B0EA406	CMP	CX,[06A4]	
-u			
4E08:024E 7221	JB	0271	
4E08:0250 7423	JZ	0275	
4E08:0252 53	PUSH	BX	
4E08:0253 50	PUSH	AX	: 输出缓冲区内容子程
4E08:0254 E8B600	CALL	030D	
4E08:0257 5B	POP	AX	
4E08:0258 5B	POP	BX	

4E08:0259	891EA106	MOV	[06A1],BX	
4E08:025D	8B3E9D06	MOV	DI,[069D]	
4E08:0261	C3	RET		
4E08:0262	BB1800	MOV	BX,0018	
4E08:0265	F6069A0601	TEST	BYTE PTR [069A],01	: 横扩?
4E08:026A	90	NOP		
4E08:026B	74D9	JZ	0246	
4E08:026D	B330	MOV	BL,30	: 48→BL
-u				
4E08:026F	EBD5	JMP	0246	
4E08:0271	8BD9	MOV	8X,CX	
4E08:0273	E8E4	JMP	0259	
4E08:0275	C606A306FF	MOV	BYTE PTR [06A3],FF	: 缓冲区满标志FF
4E08:027A	90	NOP		
4E08:027B	EBF4	JMP	0271	
4E08:027D	803EA306FF	CMP	BYTE PTR [06A3],FF	: 缓冲区满?
4E08:0282	90	NOP		
4E08:0283	7503	JNZ	028B	
4E08:0285	E88500	CALL	030D	
4E08:0288	C606A30600	MOV	BYTE PTR [06A3],00	
4E08:028D	90	NOP		
4E08:028E	C3	RET		
-u				
4E08:028F	C6069A0600	MOV	BYTE PTR [069A],00	: 字型选择,清属性
4E08:0294	90	NOP		
4E08:0295	3C41	CMP	AL,41	: "A" ?
4E08:0297	7503	JNZ	029C	
4E08:0299	EB4C	JMP	02E7	
4E08:029B	90	NOP		
4E08:029C	3C42	CMP	AL,42	: "B" ?
4E08:029E	7509	JNZ	02A9	
4E08:02A0	800E9A0601	OR	BYTE PTR [069A],01	
4E08:02A5	90	NOP		
4E08:02A6	EB3F	JMP	02E7	
4E08:02A8	90	NOP		
4E08:02A9	3C43	CMP	AL,43	: "C" ?
4E08:02AB	750F	JNZ	02BC	
4E08:02AD	800E9A0602	OR	BYTE PTR [069A],02	: 加纵扩
-u				
4E08:02B2	90	NOP		
4E08:02B3	800E990680	OR	BYTE PTR [0699],80	: 加纵扩
4E08:02B8	90	NOP		
4E08:02B9	EB2C	JMP	02E7	
4E08:02BB	90	NOP		
4E08:02BC	3C44	CMP	AL,44	: "D" ?
4E08:02BE	750F	JNZ	02CF	
4E08:02C0	800E9A0603	OR	BYTE PTR [069A],03	: 是,加横扩纵扩
4E08:02C5	90	NOP		

4E08:02C6	800E990680	OR	BYTE PTR [0699],80	: 加纵扩
4E08:02CB	90	NOP		
4E08:02CC	EB19	JMP	02E7	
4E08:02CE	90	NOP		
4E08:02CF	3C48	CMF	AL,48	: "H" ?
4E08:02D1	770A	JA	02DD	
-u				
4E08:02D3	800E9A0604	OR	BYTE PTR [069A],04	
4E08:02D8	90	NOP		
4E08:02D9	2C04	SUB	AL,04	
4E08:02DB	EB88	JMP	0295	
4E08:02DD	3C49	CMF	AL,49	: "I" ?
4E08:02DF	7506	JNZ	02E7	
4E08:02E1	800E9A0680	OR	BYTE PTR [069A],80	
4E08:02E6	90	NOP		
4E08:02E7	33C0	XOR	AX,AX	
4E08:02E9	8026990680	AND	BYTE PTR [0699],80	
4E08:02EE	90	NOP		
4E08:02EF	E9D7FD	JMP	00C9	
4E08:02F2	833EA10600	CMF	WORD PTR [06A1],+00	: 列数为 0 ?
-u				
4E08:02F7	7509	JNZ	0302	
4E08:02F9	E85302	CALL	054F	
4E08:02FC	E85BFD	CALL	005A	
4E08:02FF	E9C7FD	JMP	00C9	
4E08:0302	E85D02	CALL	0562	
4E08:0305	E80500	CALL	0300	
4E08:0308	33C0	XOR	AX,AX	
4E08:030A	E98CFD	JMP	00C9	
4E08:030D	52	PUSH	DX	: 输出缓冲区内容
4E08:030E	BA4906	MOV	DX,0649	: 定磁盘缓冲在DS : 0649
4E08:0311	B41A	MOV	AH,1A	
4E08:0313	CD21	INT	21	
4E08:0315	5A	POP	DX	
4E08:0316	8B3E9D06	MOV	DI,[069D]	: 缓冲区指针
-u				
4E08:031A	BEA806	MOV	SI,06A8	: 打印缓冲代码属性 → (069D)
4E08:031D	89369D06	MOV	[069D],SI	
4E08:0321	F606990680	TEST	BYTE PTR [0699],80	: 纵扩?
4E08:0326	90	NOP		
4E08:0327	744C	JZ	0375	
4E08:0329	E8AE01	CALL	04DA	
4E08:032C	AD	LODSW		
4E08:032D	A880	TEST	AL,80	
4E08:032F	7423	JZ	0354	
4E08:0331	BADB	MOV	BL,AL	
4E08:0333	AD	LODSW		
4E08:0334	8AFB	MOV	BH,AL	

4E08:0336 F6C402	TEST	AH,02	
4E08:0339 7506	JNZ	0341	
-u			
4E08:033B EB6F01	CALL	04AD	
4E08:033E EB07	JMP	0347	
4E08:0340 90	NOP		
4E08:0341 EB9700	CALL	03DB	: 读字模到交换区子程
4E08:0344 E8DB00	CALL	0422	
4E08:0347 3BF7	CMP	SI,DI	
4E08:0349 75E1	JNZ	032C	
4E08:034B E8B501	CALL	0503	
4E08:034E E8B901	CALL	04DA	
4E08:0351 EB25	JMP	0378	
4E08:0353 90	NOP		
4E08:0354 F6C402	TEST	AH,02	
4E08:0357 7505	JNZ	035E	
4E08:0359 E87001	CALL	04CC	
-u			
4E08:035C EBE9	JMP	0347	
4E08:035E 3C20	CMP	AL,20	
4E08:0360 7505	JNZ	0367	
4E08:0362 EB6701	CALL	04CC	
4E08:0365 EBEO	JMP	0347	
4E08:0367 B7A3	MOV	BH,A3	
4E08:0369 0CB0	OR	AL,80	
4E08:036B 8AD8	MOV	BL,AL	
4E08:036D E86B00	CALL	03DB	
4E08:0370 EB2401	CALL	0497	
4E08:0373 EBD2	JMP	0347	
4E08:0375 EB6A01	CALL	04E2	
4E08:037B BEA806	MOV	SI,06A8	
4E08:037B C6069B0601	MOV	BYTE PTR [069B],01	
-u			
4E08:0380 90	NOP		
4E08:0381 AD	LDSW		
4E08:0382 A880	TEST	AL,80	
4E08:0384 7434	JZ	03BA	
4E08:0386 8AD8	MOV	BL,AL	
4E08:0388 AD	LDSW		
4E08:0389 8AF8	MOV	BH,AL	
4E08:038B E84D00	CALL	03DB	
4E08:038E EB9100	CALL	0422	
4E08:0391 3BF7	CMP	SI,DI	
4E08:0393 75EC	JNZ	0381	
4E08:0395 BEA806	MOV	SI,06A8	
4E08:0398 B9369D06	MOV	[069D],SI	
4E08:039C C706A1060000	MOV	WORD PTR [06A1],0000	: 缓冲区代码列数清 0

```

-u
4E08:03A2 EB7001      CALL    0515
4E08:03A5 F6069A0602  TEST    BYTE PTR [069A],02      : 纵扩?
4E08:03AA 90          NOP
4E08:03AB 7506          JNZ     03B3
4E08:03AD 802699067F  AND     BYTE PTR [0699],7F      : 取状态低7位
4E08:03B2 90          NOP
4E08:03B3 C6069B0600  MOV     BYTE PTR [069B],00
4E08:03BB 90          NOP
4E08:03B9 C3          RET
4E08:03BA 3C20          CMP     AL,20
4E08:03BC 7505          JNZ     03C3
4E08:03BE E80B01      CALL    04CC
4E08:03C1 EBCE          JMP     0391

-u
4E08:03C3 F6C480      TEST    AH,80
4E08:03C6 750E          JNZ     03D6
4E08:03C8 B7A3      MOV     BH,A3
4E08:03CA 0C80      OR      AL,80
4E08:03CC BADB      MOV     BL,AL
4E08:03CE E80A00      CALL    03DB
4E08:03D1 E8C300      CALL    0497
4E08:03D4 EBBB      JMP     0391
4E08:03D6 E8CD00      CALL    04A6
4E08:03D9 EBB6      JMP     0391
4E08:03DB 50          PUSH   AX                        : 读字模到交换入口
4E08:03DC 52          PUSH   DX
4E08:03DD F6C402      TEST    AH,02
4E08:03E0 7506          JNZ     03EB
4E08:03E2 BADA05      MOV     DX,05DA

-u
4E08:03E5 E80F      JMP     03F6
4E08:03E7 90          NOP
4E08:03EB BAF05      MOV     DX,05FF
4E08:03EB F6069B0601  TEST    BYTE PTR [069B],01
4E08:03F0 90          NOP
4E08:03F1 7403          JZ      03F6                      : CLIB 242 FCB
4E08:03F3 BA2406      MOV     DX,0624
4E08:03F6 87DA      XCHG   DX,BX
4E08:03FB 81E27F7F  AND     DX,7F7F
4E08:03FC 81EA2121  SUB     DX,2121
4E08:0400 B05E      MOV     AL,5E
4E08:0402 F6E6      MUL     DH
4E08:0404 32F6      XOR     DH,DH

-u
4E08:0406 03C2      ADD     AX,DX
4E08:0408 894721      MOV     [BX+21],AX
4E08:040B 87DA      XCHG   DX,BX
4E08:040D B427      MOV     AH,27                    : 读记录, CX = 1 一个记录

```

4E08:040F B90100	MOV	CX,0001
4E08:0412 CD21	INT	21
4E08:0414 0AC0	OR	AL,AL
4E08:0416 7503	JNZ	041B
4E08:0418 5A	POP	DX
4E08:0419 58	POP	AX
4E08:041A C3	RET	
4E08:041B 8B26A606	MOV	SP,[06A6]
4E08:041F E9A7FC	JMP	00C9
4E08:0422 8AFC	MOV	BH,AH
4E08:0424 56	PUSH	SI
4E08:0425 B91800	MOV	CX,0018
-u		
4E08:0428 F6C401	TEST	AH,01
4E08:042B 7403	JZ	0430
4E08:042D B93000	MOV	CX,0030
4E08:0430 BE4906	MOV	SI,0649
4E08:0433 56	PUSH	SI
4E08:0434 AC	LODSB	
4E08:0435 F6C704	TEST	BH,04
4E08:0438 7403	JZ	043D
4E08:043A 0A4402	OR	AL,[SI+02]
4E08:043D EB1AFC	CALL	005A
4E08:0440 AC	LODSB	
4E08:0441 F6C704	TEST	BH,04
4E08:0444 7403	JZ	0449
4E08:0446 0A4402	OR	AL,[SI+02]
-u		
4E08:0449 E80EFC	CALL	005A
4E08:044C AC	LODSB	
4E08:044D F6C704	TEST	BH,04
4E08:0450 7403	JZ	0455
4E08:0452 0A4402	OR	AL,[SI+02]
4E08:0455 E802FC	CALL	005A
4E08:0458 5E	POP	SI
4E08:0459 F6C701	TEST	BH,01
4E08:045C 7510	JNZ	046E
4E08:045E B3C603	ADD	SI,+03
4E08:0461 F6C780	TEST	BH,80
4E08:0464 7403	JZ	0469
4E08:0466 B3C603	ADD	SI,+03
-u		
4E08:0469 E2C8	LOOP	0433
4E08:046B EB28	JMP	0495
4E08:046D 90	NOP	
4E08:046E 49	DEC	CX
4E08:046F AC	LODSB	
4E08:0470 F6C704	TEST	BH,04

4E08:0473 7403	JZ	047B	
4E08:0475 0A4402	OR	AL,[SI+02]	
4E08:0478 E8DFFB	CALL	005A	
4E08:047B AC	LODSB		
4E08:047C F6C704	TEST	BH,04	
4E08:047F 7403	JZ	0484	
4E08:0481 0A4402	OR	AL,[SI+02]	
4E08:0484 E8D3FB	CALL	005A	
4E08:0487 AC	LODSB		
4E08:048B F6C704	TEST	BH,04	
-u			
4E08:048B 7403	JZ	0490	
4E08:048D 0A4402	OR	AL,[SI+02]	
4E08:0490 E8C7FB	CALL	005A	
4E08:0493 E29E	LODP	0433	
4E08:0495 5E	POP	SI	
4E08:0496 C3	RET		
4E08:0497 56	PUSH	SI	
4E08:049B B91800	MOV	CX,001B	
4E08:049B F6C401	TEST	AH,01	
4E08:049E 7506	JNZ	04A6	
4E08:04A0 B90C00	MOV	CX,000C	
4E08:04A3 80CC84	OR	AH,84	
4E08:04A6 80E4FE	AND	AH,FE	
4E08:04A9 8AFC	MOV	BH,AH	
-u			
4E08:04AB EBB3	JMP	0430	
4E08:04AD 50	PUSH	AX	
4E08:04AE B91800	MOV	CX,001B	
4E08:04B1 F6C401	TEST	AH,01	
4E08:04B4 7403	JZ	04B9	
4E08:04B6 B93000	MOV	CX,0030	
4E08:04B9 32C0	XOR	AL,AL	: 输出空白
4E08:04BB E89CFB	CALL	005A	
4E08:04BE 32C0	XOR	AL,AL	: 输出空白
4E08:04C0 E897FB	CALL	005A	
4E08:04C3 32C0	XOR	AL,AL	: 输出空白
4E08:04C5 E892FB	CALL	005A	
4E08:04C8 E2EF	LOOP	04B9	: 直到CX = 0
4E08:04CA 5B	POP	AX	
-u			
4E08:04CB C3	RET		
4E08:04CC 50	PUSH	AX	
4E08:04CD B91800	MOV	CX,001B	
4E08:04D0 F6C401	TEST	AH,01	
4E08:04D3 75E4	JNZ	04B9	
4E08:04D5 B90C00	MOV	CX,000C	
4E08:04DB EBDF	JMP	04B9	: 转送空白

4E08:04DA 53	PUSH	BX	
4E08:04DB B347	MOV	BL,47	: "G"
4E08:04DD E80A00	CALL	04EA	
4E08:04E0 5B	POP	BX	
4E08:04E1 C3	RET		
4E08:04E2 53	PUSH	BX	
4E08:04E3 B334	MOV	BL,34	: "4" 送打印
4E08:04E5 E80200	CALL	04EA	
4E08:04E8 5B	POP	BX	
4E08:04E9 C3	RET		
4E08:04EA 8B0EA106	MOV	CX,[06A11]	: 打印列数→CX
-u			
4E08:04EE B01B	MOV	AL,1B	
4E08:04F0 E867FB	CALL	005A	
4E08:04F3 BAC3	MOV	AL,BL	
4E08:04F5 E862FB	CALL	005A	
4E08:04F8 BAC5	MOV	AL,CH	
4E08:04FA E85DFB	CALL	005A	
4E08:04FD BAC1	MOV	AL,CL	
4E08:04FF E858FB	CALL	005A	
4E08:0502 C3	RET		
4E08:0503 53	PUSH	BX	
4E08:0504 BB1200	MOV	BX,0012	
4E08:0507 E81700	CALL	0521	
4E08:050A E80200	CALL	050F	: 换行
4E08:050D 5B	POP	BX	
-u			
4E08:050E C3	RET		
4E08:050F B00A	MOV	AL,0A	: 换行
4E08:0511 E846FB	CALL	005A	
4E08:0514 C3	RET		
4E08:0515 53	PUSH	BX	
4E08:0516 BB1400	MOV	BX,0014	
4E08:0519 E80500	CALL	0521	
4E08:051C E8F0FF	CALL	050F	
4E08:051F 5B	POP	BX	
4E08:0520 C3	RET		
4E08:0521 B01B	MOV	AL,1B	: esc
4E08:0523 E834FB	CALL	005A	
4E08:0526 B04A	MOV	AL,4A	: "J"
4E08:0528 E82FFB	CALL	005A	
4E08:052B BAC3	MOV	AL,BL	
4E08:052D E82AFB	CALL	005A	
-u			
4E08:0530 C3	RET		
4E08:0531 43	INC	BX	: 页长记录单元CS : 531
4E08:0532 0000	ADD	[BX+SI],AL	
4E08:0534 008B0E31	ADD	[BP+DI+310E],CL	

4E08:0538	052B0E	ADD	AX,0E2B	
4E08:053B	3305	XOR	AX,[DI]	
4E08:053D	B00A	MOV	AL,0A	: 换行送打印
4E08:053F	33D2	XOR	DX,DX	
4E08:0541	EB16FB	CALL	005A	
4E08:0544	E2F7	LOOP	053D	
4E08:0546	C70633050000	MOV	WORD PTR [0533],0000	
4E08:054C	E97AFB	JMP	00C9	
4E08:054F	BB0E3305	MOV	CX,[0533]	
-u				
4E08:0553	41	INC	CX	
4E08:0554	3B0E3105	CMP	CX,[0531]	
4E08:0558	7203	JB	055D	
4E08:055A	33C9	XOR	CX,CX	
4E08:055C	41	INC	CX	
4E08:055D	890E3305	MOV	[0533],CX	
4E08:0561	C3	RET		
4E08:0562	51	PUSH	CX	
4E08:0563	BB0E3305	MOV	CX,[0533]	
4E08:0567	41	INC	CX	
4E08:0568	F606990680	TEST	BYTE PTR [0699],80	: 纵扩?
4E08:056D	90	NOP		
4E08:056E	7401	JZ	0571	
4E08:0570	41	INC	CX	
4E08:0571	3B0E3105	CMP	CX,[0531]	
-u				
4E08:0575	720A	JB	0581	
4E08:0577	33C9	XOR	CX,CX	
4E08:0579	41	INC	CX	
4E08:057A	F606990680	TEST	BYTE PTR [0699],80	: 纵扩?
4E08:057F	90	NOP		
4E08:0580	41	INC	CX	
4E08:0581	890E3305	MOV	[0533],CX	
4E08:0585	59	POP	CX	
4E08:0586	C3	RET		
4E08:0587	0000	ADD	[BX+SI],AL	
4E08:0589	0000	ADD	[BX+SI],AL	
4E08:058B				
4E08:058F				
4E08:0592				
-u cs:58c				
4E08:058C	8B1E8705	MOV	BX,[0587]	: 定页长处理BCD码转换为二进制数
4E08:0590	FEC3	INC	BL	
4E08:0592	BE8905	MOV	SI,0589	
4E08:0595	8B40FF	MOV	[BX+SI-01],AL	
4E08:0598	80FB03	CMP	BL,03	
4E08:059B	7534	JNZ	05D1	
4E08:059D	32DB	XOR	BL,BL	

```

4E08:059F 891E8705    MOV     [0587],BX
4E08:05A3 AC          LODSB
4E08:05A4 2C30        SUB     AL,30
4E08:05A6 B30A        MOV     BL,0A
4E08:05A8 F6E3        MUL     BL
4E08:05AA 8BD0        MOV     DX,AX
-u
4E08:05AC AC          LODSB
4E08:05AD 2C30        SUB     AL,30
4E08:05AF 32E4        XOR     AH,AH
4E08:05B1 03D0        ADD     DX,AX
4E08:05B3 8BC2        MOV     AX,DX
4E08:05B5 F6E3        MUL     BL
4E08:05B7 BAC4        MOV     AL,AH
4E08:05B9 AC          LODSB
4E08:05BA 2C30        SUB     AL,30
4E08:05BC 02E0        ADD     AH,AL
4E08:05BE 88263105    MOV     [0531],AH      : 存页长记录
4E08:05C2 C70633050000  MOV     WORD PTR [0533],0000
4E08:05C8 80269906DB    AND     BYTE PTR [0699],DB
-u
4E08:05CD 90          NOP
4E08:05CE E9F8FA        JMP     00C9           : 取存打印机状态字节
4E08:05D1 891E8705    MOV     [0587],BX
4E08:05D5 E9F1FA        JMP     00C9           : 取存打印机状态字节
4E08:05D8 0000        ADD     [BX+SI],AL
4E08:05DA 03434C        ADD     AX,[BP+DI+4C]
4E08:05DD 49          DEC     CX
4E08:05DE 42          INC     DX
4E08:05DF 3234        XOR     DH,[SI]
4E08:05E1 2020        AND     [BX+SI],AH
4E08:05E3 2020        AND     [BX+SI],AH
4E08:05E5 2000        AND     [BX+SI],AL
4E08:05E7 0000        ADD     [BX+SI],AL
4E08:05E9 0000        ADD     [BX+SI],AL
4E08:05EB 0000        ADD     [BX+SI],AL
-d cs:05dB
4E08:05DB 00 00 03 43 4C 49 42 32    ...CLIB2
4E08:05E0 34 20 20 20 20 20 00 00-00 00 00 00 00 00 00 00    4 .....
4E08:05F0 00 00 00 00 00 00 00 00-00 00 00 00 00 00 03    .....
4E08:0600 43 4C 49 42 32 34 31 20-20 20 20 00 00 00 00 00    CLIB241 .....
4E08:0610 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00    .....
4E08:0620 00 00 00 00 03 43 4C 49-42 32 34 32 20 20 20 20    ....CLIB242
4E08:0630 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00    .....
4E08:0640 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00    .....
4E08:0650 00 00 00 00 00 00 00 00    .....
-d
4E08:0658 00 00 00 00 00 00 00 00    .....

```

```

4E08:0660 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
4E08:0670 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
4E08:0680 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
4E08:0690 00 00 00 00 00 00 00 00-00 00 00 00 00 AB 06 00 .....
4E08:06A0 00 00 00 00 BF 09 00 00-00 00 00 00 00 00 00 .....
4E08:06B0 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
4E08:06C0 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 .....
4E08:06D0 00 00 00 00 00 00 00 00 .....

```

-u cs:0ad2

: D 32024 . exe 指令执行入口

```

4E08:0AD2 1E      PUSH    DS
4E08:0AD3 33D2     XOR     DX,DX
4E08:0AD5 52      PUSH    DX
4E08:0AD6 BEB200  MOV     SI,00B2
4E08:0AD9 AC      LODSB
4E08:0ADA 3C31     CMP     AL,31
4E08:0ADC 7509     JNZ     0AE7
4E08:0ADE B84315  MOV     AX,1543
4E08:0AE1 BB3C15  MOV     BX,153C
4E08:0AE4 EB14     JMP     0AFA
4E08:0AE6 90      NOP
4E08:0AE7 3C32     CMP     AL,32
4E08:0AE9 7509     JNZ     0AF4
4E08:0AEB B8F300  MOV     AX,00F3
4E08:0AEE BBEE00  MOV     BX,00EE
4E08:0AF1 EB07     JMP     0AFA

```

-u

```

4E08:0AF3 90      NOP
4E08:0AF4 B8DF00  MOV     AX,00DF
4E08:0AF7 BBDA00  MOV     BX,00DA
4E08:0AFA 0E      PUSH    CS
4E08:0AFB 1F      POP     DS
4E08:0AFC A30D00  MOV     [000D],AX
4E08:0AFF 891E0B00 MOV     [000B],BX
4E08:0B03 B81735  MOV     AX,3517
4E08:0B06 CD21     INT     21
4E08:0B08 891E1100 MOV     [0011],BX
4E08:0B0C 8C061300 MOV     [0013],ES
4E08:0B10 B82135  MOV     AX,3521

```

-u

```

4E08:0B13 CD21     INT     21
4E08:0B15 891E1500 MOV     [0015],BX
4E08:0B19 8C061700 MOV     [0017],ES
4E08:0B1D 33C0     XOR     AX,AX
4E08:0B1F BEC0     MOV     ES,AX
4E08:0B21 BEC100  MOV     SI,00C1
4E08:0B24 26      ES:
4E08:0B25 8B04     MOV     AX,[SI]
4E08:0B27 A31B00  MOV     [001B],AX

```


4E08:0B2A 26	ES:		
4E08:0B2B 8B4402	MOV	AX,[SI+02]	
4E08:0B2E 8BFE	MOV	DI,SI	
4E08:0B30 A30F00	MOV	[000F],AX	
-u			
4E08:0B33 8D061D00	LEA	AX,[001D]	
4E08:0B37 AB	STOSW		: 存AX → ES:DI
4E08:0B38 BCC8	MOV	AX,CS	
4E08:0B3A AB	STOSW		
4E08:0B3B 8ED8	MOV	DS,AX	
4E08:0B3D BADA05	MOV	DX,05DA	
4E08:0B40 B40F	MOV	AH,0F	
4E08:0B42 CD21	INT	21	
4E08:0B44 0AC0	OR	AL,AL	
4E08:0B46 7539	JNZ	0BB1	
4E08:0B48 BAF05	MOV	DX,05FF	
4E08:0B4B B40F	MOV	AH,0F	
4E08:0B4D CD21	INT	21	
4E08:0B4F 0AC0	OR	AL,AL	
4E08:0B51 7534	JNZ	0BB7	
-u			
4E08:0B53 BA2406	MOV	DX,0624	
4E08:0B56 B40F	MOV	AH,0F	
4E08:0B58 CD21	INT	21	
4E08:0B5A 0AC0	OR	AL,AL	
4E08:0B5C 752F	JNZ	0BB0	
4E08:0B5E B84800	MOV	AX,0048	
4E08:0B61 A3E805	MOV	[05E8],AX	
4E08:0B64 A30D06	MOV	[060D],AX	
4E08:0B67 A33206	MOV	[0632],AX	
4E08:0B6A B8B300	MOV	DX,00B3	: 改 INT 21 中断在CS : 003 D
4E08:0B6D B425	MOV	AH,25	
4E08:0B6F B017	MOV	AL,17	
4E08:0B71 CD21	INT	21	
-u			
4E08:0B73 BA3D00	MOV	DX,003D	: 改INT 17H 中断在CS : 0083
4E08:0B76 B021	MOV	AL,21	
4E08:0B78 B425	MOV	AH,25	
4E08:0B7A CD21	INT	21	
4E08:0B7C BAD50A	MOV	DX,0AD5	
4E08:0B7F CD27	INT	27	
4E08:0B81 BA9508	MOV	DX,0B95	: 差错信息
4E08:0B84 EB0A	JMP	0B90	
4E08:0B86 90	NOP		
4E08:0B87 BAB108	MOV	DX,0BB1	: 差错信息
4E08:0B8A EB04	JMP	0B90	
4E08:0B8C 90	NOP		
4E08:0B8D BACE0B	MOV	DX,0BCE	: 差错信息

```

4E08:0B90 B407      MOV     AH,09
4E08:0B92 CD21      INT     21
-u
4E08:0B94 CB        RETF
4E08:0B95 0A0D      OR      CL,[DI]
4E08:0B97 43        INC     BX
4E08:0B98 41        INC     CX
4E08:0B99 4E        DEC     SI
4E08:0B9A 4E        DEC     SI
4E08:0B9B 4F        DEC     DI
4E08:0B9C 54        PUSH   SP
4E08:0B9D 204F50      AND     [BX+50],CL
4E08:0BA0 45        INC     BP
4E08:0BA1 4E        DEC     SI
4E08:0BA2 204649      AND     [BP+49],AL
4E08:0BA5 4C        DEC     SP
4E08:0BA6 45        INC     BP
4^C

```

-dcs:0b95

```

4E08:0B95 0A 0D 43-41 4E 4E 4F 54 20 4F 50
4E08:0BA0 45 4E 20 46 49 4C 45 20-43 3A 43 4C 49 42 32 34
4E08:0BB0 24 0A 0D 43 41 4E 4E 4F-54 20 4F 50 45 4E 20 46
4E08:0BC0 49 4C 45 20 43 3A 43 4C-49 42 32 34 31 24 0A 0D
4E08:0BD0 43 41 4E 4E 4F 54 20 4F-50 45 4E 20 46 49 4C 45
4E08:0BE0 20 43 3A 43 4C 49 42 32-34 32 24 00 00 00 00 00
4E08:0BF0 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
4E08:0C00 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
4E08:0C10 00 00 00 00 00

```

-d

```

4E08:0C15 00 00 00-00 00 00 00 00 00 00 00 00 00 00
4E08:0C20 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00
4E08:0C30 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00
4E08:0C40 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00
4E08:0C50 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00
4E08:0C60 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00
4E08:0C70 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00
4E08:0C80 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00
4E08:0C90 00 00 00 00 00

```

差错信息

```

..CANNOT OP
EN FILE C:CLIB24
$.CANNOT OPEN F
ILE C:CLIB241$.
CANNOT OPEN FILE
C:CLIB242$.
.....
.....
.....

```

D 32024 , exe 文件控制块

```

.....
.....
.....
.....
.....
.....
.....
.....
.....

```

第九章 CC-DOS使用说明

9.1 系统启动

使用CC-DOS时,有两种方式可以启动系统,即冷启动和热启动。冷启动的过程是:将载有CC-DOS的软盘插入A驱动器,然后接通电源,并打开显示器开关和主机开关,显示器左上角出现闪动光标,大约5秒钟后,A驱动器鸣叫开始工作,同时A驱动器的指示灯开亮,把CC-DOS读入内存,当A驱动器的指示灯熄灭,程序读入完毕,显示器就会显示系统题头和提示行:

A> _

这就表示CC-DOS操作系统已启动成功,可继续进行正常工作。

当用户工作过程中,发现自己的程序死锁或希望重新启动CC-DOS时,可用热启动的方式启动。热启动的过程是:将载有CC-DOS的软盘插入A驱动器,同时按下CTRL键、ATL键和DEL键,A驱动器鸣叫开始工作,然后和冷启动一样,显示器上显示出系统题头和提示行:

A> _

这时,便可以输入命令或建立用户自己的文件。

9.2 汉字输入

汉字操作系统CC-DOS 2.0/2.1支持四种汉字输入方式,可以通过功能键进行选择,功能键定义示于表9.1。

CC-DOS (V 2.1) 功能键定义

表 9.1

ALT + F 1	区位码输入方式
ALT + F 2	首尾码输入方式
ALT + F 3	拼音码输入方式
ALT + F 4	快速码输入方式
ALT + F 6	ASCII码输入方式
CTRL + F 5	改变BIOS中10类中断程序的模式(西文屏幕与汉字屏幕转换)
CTRL + F 6	改变显示字符的颜色(再打一次回车退出)
CTRL + F 7	纯西文和纯中文方式转换
CTRL + F 8	自动光标的建立和取消(应用程序缺光标用)
CTRL + F 9	纯中文方式的建立和取消
CTRL + F 10	定义打印机的字型和行宽

若需要退出汉字输入模式,同时按下ALT键和F6键,即可恢复西文输入模式。

1. 区位码法汉字输入操作

区位码法汉字输入的特点是无重码且输入的范围比较大。除了可以输入国标GB 2312-80所规定的一、二级汉字(共6763个)外,还可以输入:

各种符号202个(如间隔符、运算符、制表符、标点等)

序号60个 (1~20, (1)~(20), ①~⑩, (一)~(十))

数字22个 (0~9, I~XII)

英文大、小写字母52个

日文假名169个, 其中平假名83个, 片假名86个。

希腊文大、小写字母48个

俄文大、小写字母66个。

这些字符, 根据其排列位置可分94个区, 每区94个字符 (94位)。所以区位码中, 区码在前取01~94, 位码在后取01~94。选择区位码法输入汉字时, 同时按下ALT键和F1键, 此时屏幕底部提示行是:

区位: _

当键入4位区位码时, 屏幕光标处即显示出该区位码所代表的汉字或字符。

例如, 当键入1903时, 屏幕显示出“常”。

A > 常_

区位: 1903

✓ 2. 首尾码法汉字输入操作

首尾码法汉字输入的依据是把汉字分解, 归纳出汉字字母, 并把这些字母与键盘上的英文字母建立起一一对应关系, 或者把这些字母直接刻在键盘上, 形成中文键盘。当需要用首尾码法输入汉字时, 同时按下ALT键和F2键, 此时屏幕底部提示行显示:

首尾: _

首先确定待输入汉字的字形, 然后取该字形左上角的汉字字母为首码, 取该字形右下角的汉字字母为尾码。输入首尾码后, 如果重码很大, 还可以再输该汉字的首音码 (该汉字汉语拼音的第一个字母), 如果仍然有多于10个的重码, 可用“>”键把接下去的10个重码汉字显示出来, 直至找到该汉字。用“<”键可退回到前10个重码汉字显示。

例如, 要输入汉字“视”, 它的首码是“M”, 尾码是“T”, 首音码为“U”。键入“M”, 屏幕提示行为:

首尾: M 0: 扶1: 办2: 邦3: 帮4: 惫5: 被6: 博7: 勃8: 补9: 布 [274]

其中 [274] 为剩余重码字数。再键入“T”, 提示行为:

首尾: MT 0: 规 1: 轨2: 辊3: 兢4: 九5: 克6: 礼7: 龙8: 轮9: 袍 [024]

因为重码数为24, 不算很多, 我们可以利用“>”键找到它。按下“>”键, 则提示行为:

首尾: MT 0: 七 1: 乾2: 翹3: 衫4: 视5: 屯6: 丸7: 旭8: 尧9: 尤 [014]

按下“4”键, 即可在屏幕光标处显示出所选的汉字。在键入“MT”后, 我们也可以再键入首音码“U”, 此时提示行为:

首尾: MTU 0: 衫1: 视 [000]

这表示首尾码为“MTU”的汉字只有两个：衫、视。按下“1”键，即可得“视”字。

3. 拼音码法汉字输入操作

拼音码法汉字输入的特点是易学、好掌握，因此应用极广。但它的缺点是重码多。为了既减少击键次数，又不增加重码，采用了紧缩汉字拼音的方法，即将一些联用的声母和韵母紧缩为一个字母，示于表9-2。

表 9-2

拼 音	对 应 键	拼 音	对 应 键
zh	A	ai	L
ch	I	en	F
sh	U	eng	G
an	J	ing	Y
ang	H	ong	S
ao	K	ii	V

有了这12个紧缩汉字拼音码后，所有汉字输入的击键次数都不超过三次。

例如，“王”字的汉语拼音为“Wang”，输入码为“WH”。同时按下ALT键和F3键，提示行为：

拼音： []
键入“W”，提示行显示：

拼音：W 0:挖1:哇2:蛙3:洼4:娃5:瓦6:袜7:歪8:外9:腕 [195]
再键入“H”，提示行显示：

拼音：WH 0:汪1:王2:亡3:枉4:网5:往6:旺7:望8:忘9:妄 [005]
按下“1”键，便可得到“王”。

4. 快速汉字输入

所谓快速输入法，实际上就是首尾码的盲打法。当操作员能熟记汉字的首尾码时，可以用一键加空格、二键加空格或三键加空格快速输入汉字。当出现重码时，用“0~9”键选择，如果不加以选择，而进行下一次输入，则隐含选择重码汉字的第一个。

9.3 汉字打印输出

CC-DOS支持并行打印机。由于所配置的打印机不同，使用的控制代码也不同，所以CC-DOS无法同时支持各类并行打印机，而需要根据打印机型号，装入相应的汉字打印驱动程序，才具备打印汉字能力。例如：

对于FZ-100、CP-80等九针打印机，需装入ALL9P或NEW9P汉字打印驱动程序；

对于TH3070、LQ-1500等24针打印机，需装入ALL24P汉字打印驱动程序；

对于M2024打印机，可装入2024P汉字打印驱动程序。

在汉字打印输出前，可先用控制键CTRL+F10来设置打印参数，同时按下这两个键，屏

幕底部的提示行显示:

打印字号 (A~P):__ 纸宽 (80~134):

这时根据打印文件的需要,选择所使用的汉字字号,按回车键后,光标移向纸宽选择,再键入所希望的纸宽数据,按回车键,提示行消失,打印参数设置完毕。

CC-DOS提供了以下打印命令。

(1) 屏幕打印

CC-DOS与PC-DOS一样,也具有屏幕硬拷贝功能。在汉字处理作业中,当需要把屏幕上显示的汉字文件,原样不动从打印机上输出时,只要同时按下SHIFT键和PrtSc键即可实现。屏幕上的所有汉字、字符都作为图形处理,从打印机上打印出来。

(2) 用“CTRL-P”命令进行汉字打印

在PC-DOS中,常用“CTRL-P”键来达到边显示边打印的目的。而在CC-DOS中,同样也可以利用该键,把屏幕显示的信息同时从打印机上打印出来。这对于使用DIR命令列出带汉字的文件目录以及使用TYPE命令打印出汉字文件都是十分方便的。

(3) 用“PRINT”命令输出汉字文件

在CC-DOS下,可以利用假脱机输出打印命令“PRINT”,使计算机在打印文件的同时又可进行其它作业处理。这一点也和PC-DOS完全一样。由于汉字打印速度较低,所以利用“PRINT”命令是提高系统运行效率的有效方法。

如果硬盘中预先装入了 24×24 点阵的汉字库,则可以在使用“PRINT”命令打印之前,先键入D320或D32024,这样就可以打印出 24×24 点阵的高精度汉字文件。

除上述操作打印命令外,CC-DOS还支持各种高级语言的打印语句进行汉字打印输出。例如BASIC中的LPRINT语句、FORTRAN中的WRITE语句、PASCAL中的WRITELN语句以及DBASE中的REPORT FORM <文件名> TO PRINT语句等。在程序执行过程中,打印不同字号、字形的汉字,是通过输出转移符“ESC”所引导的控制码来进行转换的。字形分为两类,A~H为一类,是标准字形;I~P为另一类,输出的是横向宽度比标准字形小一半的浓缩字形。每一行中,同类字形不同字号的汉字可以混排,但是不同类字形的汉字不能出现在同一行。“ESC”的控制格式为:

“ESC”+“I”+“X”

其中X所选择(A~P)中的任一种。“ESC”控制序列规定如下:

“ESC”+“I”+“A”打印 16×16 标准点阵汉字;

“ESC”+“I”+“B”打印 16×32 点阵横向扩大一倍的汉字;

“ESC”+“I”+“C”打印 32×16 点阵纵向扩大一倍的汉字;

“ESC”+“I”+“D”打印 32×32 点阵,横向、纵向都扩大一倍的汉字;

“ESC”+“I”+“E”到“ESC”+“I”+“H”是将上述4种分别对应的字号转体 90° ,以纵排方式打印输出;

“ESC”+“I”+“I”到“ESC”+“I”+“P”是将上述8种不同的字号分别转变成浓缩体。

下面的BASIC程序就是一个用例。

CCDOS 2.1 配FX-100 9针打印机

10 WIDTH "LPT1:", 80

15 LPRINT

20 LPRINT CHR\$(27)+CHR\$(73)+CHR\$(65): "汉";

30 LPRINT CHR\$(27)+CHR\$(73)+CHR\$(66): "字";

40 LPRINT CHR\$(27)+CHR\$(73)+CHR\$(67): "系";

50 LPRINT CHR\$(27)+CHR\$(73)+CHR\$(68): "统";

60 END

执行这个程序，在同一行中打出不同字型号的4个汉字。还有一种情况是：同样A~P16种字型，不同的驱动程序打印出字型并不相同。

表9-3列出市售最常见几种打印机驱动程序以供参考。

打印驱动程序清单

表9-3

驱动文件	打印机型号	字模库点阵	输出字型种数	适配汉字库版本
ALL24P.exe	3070型	16×16		CC-DOS2.0/2.1
3070P.exe	"	"		"
ALL9P.exe	FX-100型	"	A~P16种大字	"
2024P.exe	2024型	"	A~C3种	"
Brothe16.exe	2024型/1724	"	A~P16种	"
3.COM	3070	"		要图形软件GRD·SYS
9P.exe	FX-100	"	A, B, I, J 4种	CC-DOS2.0/2.1
9999P3.exe	2024/1724	24×24	A~P16种	CC-DOS2.13
OKI8320.exe	OKI8320型	24×24	A~P16种	CC-DOS2.0/2.1
D320.exe	3070型	"		CLLB241-2
D32024.exe	2024/1724	"		CLLB241-2
OPD3202.exe	"	"		CLLB24
9999D4.exe	"	"	A~P16种	CC-DOS2.13
P3070.COM	3070机(带字库)	--		
H3070C.exe	3070型	48×48		硬盘1M字库

24点阵打印时驱动程序必须配适当的字库文件，如果字库文件对不上，就不能正确打印。CC-DOS版本必须与打印机型号适配。

改进版CC-DOS2.13汉字系统对2024/1724打印机适配能打宋，仿宋，黑，楷四种不同字体各4种字型（共A~P16种），具有目前较好的汉字库。（24×24，16×16兼有）

在普通CC-DOS2.0/2.1支持下用R2463高速打印机，不用驱动程序设定就能打精美的A~P16字型，因为该机本身带24点阵字库ROM。

日本OKI8320机是一种灵巧的宽行打印机，在普通汉字库CC-DOS2.0/2.1支持下亦能打A~P16字型但其字体与字型尺寸不同于M2024或1724打印机。

所以说打印输出的字体与字型能否打好汉字决定于三因素：CC-DOS版本，打印机型号，驱动程序。通常打印机厂商都为用户适配好，普通用户不一定深究此点。只有在非原配的条件下，用户要另配打印机时，才必须考虑打印效果，字体与字型种类。

9.4 在CC-DOS支持下的软件使用

CC-DOS是PC-DOS的汉化版，因此，一般西文软件不作改动就可以在CC-DOS下

运行，并具有汉字信息处理能力。但有些软件，因为有些特殊约定所以必须经过改造才能在CC-DOS下运行，如C-COBOL、C-DATABASE II/III等。

1. BASIC 语言

BASIC 语言作为一种交互式语言，简单易学，得到广泛应用。在CC-DOS支持下，具有汉字信息、处理功能之后，交互式功能更直观明了。深受用户欢迎。

(1) 怎样进入汉字BASIC状态

在CC-DOS启动系统后，将装有BASIC解释程序的软盘插入A驱动器（假设使用高级BASIC 2.0），然后打入命令BASICA并按回车键。即：

```
A>BASICA <CR>
```

BASIC的解释程序就调入了内存并在屏幕顶部显示：

```
The IBM Personal Computer Basic
Version A2.0 Copyright IBM Corp. 1981, 1982, 1983
60865 Bytes free
OK
```

这时，便可以输入命令或编写程序了。以后的编程格式，命令语句以及操作方法完全与PC-DOS下的BASIC一样。只是在西文字符串出现的地方，根据用户的需要可换成汉字。

(2) BASIC程序中汉字的使用场合

在CC-DOS支持下的BASIC程序，凡是出现字符串处，需要时原则上都可以使用汉字串。具体说有如下场合。

①注释语句可使用汉字注释，使程序更容易理解。

②PRINT/LPRINT语句中，可直接使用汉字作为显示或打印输出的信息。如：

```
LPRINT "The first"; N; "Prime numbers are:";
```

可改写成：

```
LPRINT "前"; N; "个质数是:"
```

③INPUT和LINE INPUT语句中，要求用户输入数据时，可用汉字提示。如：

```
INPUT "The original data:"
```

可改写成：

```
INPUT "初始数据:"
```

④为字符串变量赋值时，可把汉字赋给字符串变量。如：READ DATA语句、INPUT\$(n)等。

下面是一个用汉字建立通讯录的例子。这是某人为他50个朋友建立的通讯录，共有姓名、工作单位、家庭住址三项内容。

```
10 "EX 1-1 建立通讯录
20 DIM ADDRBK$(49, 2)
30 PRINT "请输入通讯录数据"
40 PRINT "姓名,      工作单位, 家庭住址 <CR>"

50 FOR I = 0 TO 49
60 INPUT ":", ADDRBK$(1, 0), ADDRBK$(1, 1), ADDRBK$(1, 2)
```


70 NEXT I

80 END

执行这个程序时，屏幕上的提示信息及用户输入的部分数据为：

请输入通讯录数据

姓 名 工作单位 家庭住址

：张 强，北京计算机二厂，北京和平路20号

：李伟男，北京化工三厂，北京人民路35号

：刘大华，中国科学院五所，北京上海路50号

2. dBASE II/III 关系型数据库

dBASE II 关系型数据库，以其简单明了，功能齐全，而成为世界上最流行、最受欢迎的数据库，享有“大众数据库”的美称。dBASE III 是在 dBASE II 基础上改进的新一代关系型数据库，功能更加齐全，越来越被广泛应用。尤其具有汉字信息处理能力的 C-dBASE II 和 C-dBASE III 更为人们所熟悉和欢迎。

(1) C-dBASE II 启动

在 C C-DOS 启动系统后，将装有 C-dBASE II 的软盘插入 A 驱动器。然后打入命令 DBASE 并按回车键。即：

A>DBASE <CR>

就进入了 C-dBASE II 状态，并显示提示行：

C-dBASE II Version 2.41, August 1984

其中“·”是 C-dBASE II 的提示符，它表示此时用户可以用 C-dBASE II 提供的命令语句进行各种操作，或者执行由 C-dBASE II 命令语句所编制成的数据库应用程序。

(2) C-dBASE II 中汉字使用的场合

C-dBASE II 是在西文 dBASE II 的基础上，对其中 DBASE·COM 模块中的常驻模块以及 DBASEMSG·OVR 覆盖模块，作了能适应汉字处理的修改而实现的。但功能完全一样，而且全部沿用了西文 dBASE II 的命令。编写 C-dBASE II 应用程序时，下列情况可使用汉字：

- ① 凡使用文件名作参数的地方，均可使用汉字文件名，但最多允许四个汉字。
- ② 字段名及存贮变量名可使用汉字，但最多允许五个汉字
- ③ 数据库文件的数据输入和文件修改命令，可以在字符型字段中输入汉字。
- ④ 字符串型存贮变量的赋值 (STORE) 和输入 (INPUT, ACCEPT) 命令可以使用汉字。
- ⑤ 注释语句 (NOTE, *) 可用汉字注释。
- ⑥ 格式输入命令 (@...GET) 和格式输出命令 (@...SAY) 可以使用汉字字符串作为提示符，并输出和输入汉字信息。
- ⑦ 宏替换函数 "&" 可用来替换一个汉字字符串。

(4) C-dBASE III 启动

在 C C-DOS 启动系统后，将装有 C-dBASE III 的软盘插入 A 驱动器，键入 DBASE 命令并按回车键。即：

A > D B A S E < C R >

就进入了 C-dBASE III 状态, 屏幕显示:

C-dBASE III 版本 1.0 A C C - D O S * * *

电子工业部第六研究所版权所有!

:

按下 F1 键作为帮助

输入一个命令并且按下回车键

其中 “.” 为 C-dBASE III 提示符, 它表示已进入 C-dBASE III 状态, 用户可以键入 C-dBASE III 命令或运行 C-dBASE III 程序。如果是初学者, 对 C-dBASE III 不熟悉, 可以按下 F1 键, 求得帮助, 此时屏幕显示:

```
-----最大帮助-----                                主菜单
dBASE III 主菜单
-----
1. 开始                : 启动
2. 什么是...          : 名词解释
3. 怎么做              : 操作步骤
4. 建立数据库          : 建库命令
5. 使用现有的数据库   : 增删修改已建库
6. 命令和函数          : 命令和功能函数
命令 >
```

用户可以根据以上提示, 提出自己的问题, 求得帮助, 从而熟悉 C-dBASE III。在这里只能查询命令解释, 打 <esc> 退回原点, 才能执行。

C-dBASE III 应用汉字的场合与 C-dBASE II 一样, 这里不再介绍。

3. 汉字 WORDSTAR

WORDSTAR 软件是当今微机上流行的一种文字处理软件, 具有很强的字处理功能, 深受广大用户喜爱。汉字 WORDSTAR 是在西文 WORDSTAR 基础上汉化成功的汉字文字处理软件, 在 C C - D O S 支持下运行。启动过程如下:

首先用 C C - D O S 启动系统, 屏幕上出现提示符 “A >” 后, 再将存有汉字 WORDSTAR 的软盘插入 A 驱动器, 然后键入 “WS” 命令并按回车键。即:

A > W S < C R >

此时, 屏幕显示出菜单式起始命令:

: 《起始命令》

```
-----
D 进入编辑          E 更换文件名
P 打印文件          Q 拷贝文件
R 运行程序          Y 删除文件
N 编辑非文书文件   X 退出
```

用户可根据以上提示, 选择命令, 进行自己的工作。如要建立一个新文件或编辑一个已存放于软

盘的旧文件，按下D键，屏幕显示：

使用本命令建立新文书文件或更改现存文件，文件名前是一个驱动器字母 A — B 及冒号，如省略则隐含当前驱动器；文件名取 1 — 4 个汉字，扩展名取一个汉字。

文件名是：文件名？

如此下去，用户可通过人机对话完成自己的工作。

附录 I 软中断程序简介

类型	地址	子功能	名称 / 说明
0	0-3	None	DIVIDE - BY - 0 / Automatically taken upon division over flow
1	4-7	None	SINGLE - STEP / taken after every instruction when CPU Trap Flag indicates single - step mode (bit 8 of FLAGS is 1)
2	8-B	None	NON - MASKABLE / vector not disabled via CLI. Taken when hard memory error occurs. Displays message, halts system
3	C-F	None	BREAKPOINT / taken when CPU executes the breakpoint opcode (0CCH)
4	10-13	None	OVERFLOW / taken when OF = 1 and INTO opcode is executed
5	14-17	None	PRINT - SCREEN / service dumps the screen to the printer Invoked by KBD - INT shifted key 55 (PrtSc)
6	18-1B	None	reserved
7	1C-1F	None	reserved
8	20-23	None	TIMER - INT / 55ms timer "tick" taken 18.2 times per second Updates BIOS clock and turns off diskette drive motors after 2 seconds of inactivity.
9	24-27	None	KBD - INT / taken whenever a key is pressed or released Stores characters and scan - codes in buffer at 0040:001E Updates shift key status at 0040:0017, 18
A	28-2B	None	reserved
B	2C-2F	None	reserved
C	30-33	None	reserved
D	34-37	None	reserved
E	38-3B	None	DISKETTE - INT / indicates that a seek is in progress (sets bit 0 of 0040:003F)
F	3C-3F	None	reserved
10	40-43		VIDEO - IO / services handle video output

类 型	地 址	子功能	名称 / 说明
-----	-----	-----	---------

- | | | | |
|---|--|--|--|
| 0 | | SET - MODE / initialize for one of seven text or graphics modes
AL = 0: 40 × 25 BW
AL = 1: 40 × 25 color
AL = 2: 80 × 25 BW
AL = 3: 80 × 25 color graphics
AL = 4: 320 × 200 color graphics
AL = 5: 320 × 200 BW graphics
AL = 6: 640 × 200 BW graphics | |
| 1 | | SET - CUR - TYPE / set the size of the cursor or turn it off CH = start line (bit 5 = no cursor), CL = end line | |
| 2 | | SET - CUR - POS / set the coordinates of The cursor DH = row (0 - 24), DL = column (0 - 79 or 0 - 39), BH = video page | |
| 3 | | READ - CUR - POS / return the position of the cursor BH = video page
returns: DH = row DL = column, CX = cursor size | |
| 4 | | READ - LIGHT - PEN / fetch light pen information
returns: AH = 0: light pen not triggered
AH = 1: DH, DL = character row, column
CH, CL = Graphics | |
| 5 | | SELECT - ACTIVE - PAGE / set page numbet for services 6 and 7
AL = new page (0 - 7 for modes 0 and 1, 0 - 3 for modes 2 and 3) | |
| 6 | | SCROLL - PAGE - UP / scroll up or clear a display window
AL = number of lines to scroll (0 = blank entire window)
CH, CL = row, column of top left corner of window
DH, DL = row, column of lowest right corner of window
BH = video attribute for blanked lines | |
| 7 | | SCROLL - PAGE - DOWN / scroll down or clear | |

			a display "window" (see subfunction 6 for parameters)
		8	READ - CHAR - ATTR / fetch a character from the cursor position BH = page returns: AL = character, AH = attribute
		9	WRITE - CHAR - ATTR / display character (s) and attribute at cursor BH = display page, CX = repeat count, AL = character to display BL = attribute (graphics modes: BL = color (add 80H for XOR)
		A	WRITE - CHAR / display character (s) only (use current attribute. See subfunction 9 for parameters)
		B	SET - COLOR - PALETTE / set palette for graphics or border for text BH = 0: select border (text mode) BL = color (0 - 31 [16 - 31 is high intensity]) BH = 1: select graphics palette BL = palette code (0 = green / red / yellow, 1 = cyan / magenta / white)
		C	WRITE - DOT / plot 1 graphics pixel DX = row, CX = column, AL = color (bit 7 to XOR the dot)
		D	READ - DOT / determine the color of 1 graphics pixel DX = row, CX = column, returns: AL = color of dot
		E	WRITE - TTY / write one character and update cursor. Also handles CR (0DH), beep (07H), and scrolls screen on if needed. AL = char so write. BH = page, BL = foreground (in graphics mode)
		F	VIDEO - STATE / fetch the mode and width (columns) of the screen returns: AL = mode (see subfunction 0), AH = width, BH = page
11	44 - 47	None	EQUIPMENT - CHECK / fetch a code describing

类型	地址	子功能	名称 / 说明
----	----	-----	---------

active peripherals
returns: AX = equipment code
bits: FP - JSSS DDVVRN!
PP = number of printers
J = joystick
SSS = serial devices (RS - 232 ports)
DD = number of disk drives (00 = 1, 01 = 2, 10 = 3, 11 = 4)
VV = video (11 = BW card, 01 and 10 = color / graphics card)
RR = RAM in mother - board (00 = 16K....11 = 64K).
N = numeric coprocessor (8087 chip)
I = disk - existence (0 = no drives in system)

12	48 - 48	None	MEMORY - SIZE / fetch AX = count of contiguous 1K RAM blocks
13	4C - 4F		DISKETTE - IO / access the diskette drives
		0	RESET / reset the disk controller chip (MPD765)
		1	GET - STATUS / fetch AL = status of most recent operation
		2	READ - SECTORS / read one or more sectors from diskette

DH = head DL = drive (0 = A, 1 = B, etc.)
CH = track CL = sector (1 - 8, 1 - 9 for DOS 2.0)
AL = sectors to read
ES:BX = address to store / fetch data
[0000:0078] = doubleword pointer to diskette paramters
returns: Carry Flag = NC (0) for successful
Carry Flag = CY (1) failure
AH = 磁盘状态字 (差错原因)
80H = timeout: disk failed to respond
40H = seek operation failure
20H = NEC disk drive controller failed
10H = bad cyclical redundancy check (CRC error)
09H = attempted access across 64K

类型	地址	子功能	名称 / 说明
----	----	-----	---------

			boundary
			08H = DMA overrun
			04H = record not found
			03H = write protect error
			02H = sector ID error (can't find address IUBK)
			01H = unknown command
		3	WRITS - SECTORS / write from memory to disk (parameters as with subfunction 2)
		4	VERIFY / verify that a write operation was successful (parameter as with subfunction 2 but ES:BX not needed)
		5	FORMAT - TRACK / write sector - ID bytes for one track ES:BX points to 8 (or 9) 4-byte sector ID marks: byte 0 = track number byte 1 = head number byte 2 = sector number byte 3 = bytes in sector (0 = 128, 1 = 259, 2 = 512, 3 = 1024)
14	50 - 53		RS232 - IO / initialize and access serial communications port
		0	INIT - COM / initialize port named by DX (0 or 1) AL has initialization bit pattern: BBBPPSLL BBB = baud rate: 110, 150, 300, 600, 1200, 2400, 4800, 9600 PP = parity: 00 = none, 01 = odd, 11 = even S = stop bits: 0 = 1, 1 = 2 LL = word length: 10 = 7 bits, 11 = 8 bits
		1	SEND - CHAR / send character to comm port DX (0 or 1) AL = character returns: bit 7 of AH = 1 if error
		2	AWAIT - COMM - CHAR / wait for a character from comm port DX return: AL = character

类型	地址	子功能	名称 / 说明
----	----	-----	---------

			AH = error code (0 for no error)
		3	COMM - STATUS / fetch the status of comm port DX (0 or 1) returns: comm - line status in AH, modem status in AL AH bit 7 = timeout bit 6 = empty transmit shift register bit 5 = empty transmit holding register bit 4 = break detected ("long space") bit 3 = framing error bit 2 = parity error bit 1 = overrun error bit 0 = data ready AL bit 7 = received line signal detect bit 6 = ring indicator bit 5 = data set ready bit 4 = clear to send bit 3 = delta receive line signal detect bit 2 = trailing edge ring detector bit 1 = delta data set ready bit 0 = delta clear to send
15	54 - 57		CASSETTE - IO / access cassette tape drive (NOP for IBM / XT)
		0	CASS - ON / start the motor
		1	CASS - OFF / turn the motor off
		2	CASS - READ / read CX bytes into buffer at ES:BX
		3	CASS - WRITE / write CX bytes to tape from buffer at ES:BX
16	58 - 5B		KBD - IO / access the keyboard buffer
		0	AWAIT - CHAR / read the next character in keyboard buffer. If no key is ready, wait for one. returns: AL = ASCII character, AH = scan code
		1	PREVIEW - KEY / see if key is ready returns: ZF = ZR (1) if no key is ready, else ZF = NZ and AX = character and scan

类型	地址	子功能	名称 / 说明
----	----	-----	---------

			code
		2	SHIFT - STATUS / fetch bit flags indicating shift status returns: AL = bit codes (same as 0040:0017) bit 7 = insert state bit 6 = CapsLock state bit 5 = NumLock state bit 4 = ScrollLock state bit 3 = alternate shift (Alt key) bit 2 = control shift (Ctrl key) bit 1 = left shift (left caps - shift key) bit 0 = right shift (right caps - shift key) note: other codes found at 0040:0013 bit 7 = insert shift (Ins key pressed) bit 6 = caps shift (CapsLock key pressed) bit 5 = num shift (NumLock key pressed) bit 4 = scroll shift (ScrollLock key pressed) bit 3 = hold state (Ctrl - NumLock is in effect)
17	5C - 5F		PRINTER - IO / access the parallel printer (s)
		0	PRINT - CHAR / send AL to printer DX (0, 1, or 2) returns: AH = 1 if unable to print, else AH = status (as below)
		1	INIT - PRINTER / set unit line low, send 0CH to printer DX returns: status as below
		2	读打印机状态字 / read status of printer DX into AH returns: bit flags: bit 7 = printer is busy bit 6 = acknowledge line state bit 5 = out - of - paperline state bit 4 = printer selected line state bit 3 = I / O error bit 2 = unused

类型	地址	子功能	名称 / 说明
			bit 1 = unused bit 0 = timeout error
18	60 - 63	None	ROM - BASIC / execute nondisk BASIC at F600:0000
19	64 - 87	None	BOOT - STRAP / read track 0, sector 1 into 0000:7C00, then transfer control to that address. If no diskette drive available, take INT 18H
1A	68 - 68		TIME - OF - DAY / access the PC internal clock
		0	READ - CLOCK / fetch count of 55ms "tick" since power up returns: CX = high word of count DX = low word of count AL = 0 when no 24 - hour overflow since power up
		1	SET - CLOCK / set number of 55ms in clock variable CX = high word of timer ticks DX = low word of timer ticks note: the clock ticks are incremented by TIMER - INT at about 18.2 times per second Therefore: counts per second = 18(12H) counts per minute = 1,092(444H) counts per hour = 65,543(10011H) counts per day = 1,573,040(1800B0H)
1B	6C - 6F	None	KEYBOARD - BREAK / routine taken when KBD - INT senses Ctrl - Break
1C	70 - 73	None	USER - TIMER - INT / taken 18.2 times per second; invoked by the TIMER - INT; normally vectors to dummy IRET
1D	74 - 77	None	VIDEO - PARMS - PTR / vector of video initialization parameters: This doubleword address points to three sets of 16 bytes containing data used to initialize for

类型	地址	子功能	名称 / 说明
----	----	-----	---------

video modes.

parameter table format corresponds to the Motorola 6845 CRT controller registers:

R0 = horizontal total (horizontal sync. in characters)

R1 = horizontal displayed (characters per line)

R2 = horizontal sync position (move display left or right)

R3 = sync. width (vertical and horiz. pulse: 4 bits each)

R4 = vertical total (total character lines)

R5 = vertical adjust adjust for 50 or 60 Hz refresh)

R6 = vertical displayed (lines of chars displayed)

R7 = vertical sync. position (lines shifted up or down)

R8 = interlace (bits 4 and 5) and skew (bits 6 and 7)

R9 = max scan line address (scan lines per character)

R10 = cursor start (starting line of cursor)

R11 = cursor stop (ending scan line of cursor)

R12 = display memory start address high byte (6 bits)

R13 = display memory start address low byte (8 bits)

R14 = cursor address high byte (6 bits)

R15 = cursor address low byte (8 bits)

1E 78 - 7B None

DISKETTE - PARMs - PTR / vector of diskette controller

parameters:

This doubleword address points to a data table used by

DISKETTE - 10 (INT 13H). 11 - byte table format:

byte 0 = 4 - bit step rate, 4 - bit head unload time

byte 1 = 7 - bit head load time, 1 - bit DMA flag

byte 2 = 55ms counts - delay until motor off

类型	地址	子功能	名称 / 说明
----	----	-----	---------

byte 3 = sector size (0 = 128, 1 = 256, 2 = 512, 3 = 1024)

byte 4 = last sector on track

byte 5 = gap between sectors

byte 6 = data length for DMA transfers

byte 7 = gap length for format

byte 8 = fill byte for format

byte 9 = head settle time (in milliseconds)

byte 10 = motor start time (in 1/8 - th second intervals)

1F	7C - 7F	None	<p>GRAPHICS - TBL - PTR / vector of data used by VIDEO - IO to display characters above ASCII 127 in graphics modes This doubleword address points to a 1k table of 128 8 - byte character definition bit patterns. The first byte of each entry is top row, last byte is bottom row</p>
20	80 - 83	None	<p>TFRMINATE / exit from the PC - DOS program that has a PSP located at CS:0000. DOS 2.0 EXIT (INT 21H, service 4CH) is similar but provides a means to leave an exit code and does not require CS to point to the PSP (program segment prefix)</p>
21	84 - 87		DOS - SERVICES / invoke DOS service named by AH (see Appendix E - 2)
22	88 - 8B		DOS - TERMINATE - ADDR / vector points to address of parent process, bumped to by TERMINATE (INT 20H)
23	8C - 8F		DOS - BREAK - ADDR / vector points to address of Ctrl - Break handling routine for currently executing program
24	90 - 93		DOS - ERROR - ADDR / vector points to address of critical error handling routine for currently executing program
25	94 - 97		<p>DOS - DISK - READ / transfers control to DOS disk driver logic Expects: AL = drive number (0 = A, 1 = B, etc.) CX = count of sectors</p>

类型	地址	子功能	名称 / 说明
			DX = beginning logical sector number DS:BX = address of data buffer Returns: Carry Flag = CY (1) if error and AH = error code warning: leaves 1 word (old flags) on stack
26	98 - 9B		DOS - DISK - WRITE / transfers control to DOS disk driver logic (see DOS - DISK - READ, INT 25H, for parameters)
27	9C - 9F	None	FIX - IN - MEMORY / exit to DOS, but leave program and/ or data resident. Expects CS:DX to point first to byte available for subsequent programs. DOS 2.0 KEEP (service 31H) is similar, but provides a means to leave an exit code.
28 - 3F	A0 - FF		Reserved by DOS
40	100 - 103		Reserved: XT fixed disk BIOS
41 - 5F	104 - 17F		Reserved
41 - 7F	104 - 1FF		Not used, available for applications
80 - 85	200 - 217		Reserved by BASIC
86 - F0	218 - 3C3		used by BASIC interpreter (BASIC exit / patch vectors)
F1 - FF	3C4 - 3FF		Not used, available for applications

附录2 PC - DOS INT 21H 系统调用简介

功能码

名称 / 说明

-
- 0 TERMINATE / terminate the program that has a PSP at CS:0000. Works exactly like INT 20H.
 - 1 KEYBOARD - INPUT / read (wait for) a character from the standard input and echo that character to the standard output.
returns:AL = character read
 - 2 DISPLAY - OUTPUT / send a character to the standard output device.
expects:DL = character to output
 - 3 AUX - INPUT / read (wait for) a character from the asynchronous communications adapter (COMn or AUX).
returns:AL = character read
 - 4 AUX - OUTPUT / send a character to the asynchronous communications adapter (COMn or AUX).
expects:DL = character to output
 - 5 PRINTER - OUTPUT / send a character to the printer (LPTn or PRN.)
expects:DL = character to output
 - 6 CONSOLE - IO / receive a character from standard input or send a character to the Standard output.
expects:DL = FFH to fetch waiting input character from input.
returns:ZF = ZR(1) if not ready, else AL = character if DL is not FFH, then the character in DL is output.
 - 7 CONSOLE - INPUT / read (wait for) a character from the standard input.
The character is not displayed.
returns:AL is input character.
 - 8 NO - ECHO - INPUT / identical to function 1, but the character is not output.
 - 9 PRINT - STRING / send a string of characters to the standard output.
expects:DS:DX:points to the string to print.
The string must be terminated by dollar sign (" \$" = \$ ASCII 24H)
 - A BUFFERED - INPUT / read a string of characters from the standard input.
expects:DS:DX points to input buffer. First byte is maximum number

- of characters desired.
 returns: second byte [DS: DX + 1] is actual length of input—less the terminating carriage return (ASCII 0DH) —the rest of the buffer contains the characters read.
- B INPUT - STATUS / see if the standard input has a character ready.
 returns: AL = FFH if a character ready, AL = 00 if no character
 Note: DOS 1.0 and DOS 1.1 : the standard input device is always the keyboard and standard output device is always the screen.
- C CLEAR - AND - INPUT / clear the standard input device buffer of any pretyped keys and then invoke an input function.
 expects: AL = input function 1, 6, 7, 8, or 0AH
- D RESET - DISK / flush all file buffers.
- E SELECT - DISK / select new default disk drive, query valid drives.
 expects: DL = new default drive (0 = A, 1 = B, etc.)
 returns: AL = total number of drives (floppy and fixed disk drives)
- F OPEN - FILE / open a file with unopened FCB pointed to by DS: DX.
 returns: AL = FFH if file not found, AL = 00 if file found.
- 10 CLOSE - FILE / close file with opened FCB pointed to by DS:DX .
- 11 SEARCH - FIRST / search the current directory for first file that matches the name in the unopened FCB pointed to by DS:DX.
 returns: AL = FFH if no match
 AL = 0 if match found and
 DTA is filled with a 32 - byte directory entry
- 12 SEARCH - NEXT / after calling function 11H, this will search for other files that match the specified wildcard (global) filename.
 returns: same as SEARCH - FIRST (function 11H)
- 13 DELETE - FILE / delete file named by the FCB pointed to by DS: DX.
- 14 READ - SEQ / read sequential data from the file named by the opened FCB pointed to by DS: DX. Count of bytes is in FCB RECORD - SIZE field.
 returns: one record is read into DTA (see DOS function 1AH) sets FCB CURRENT - RECORD field for the next read operation.
 AL = 0 if transfer was successful (no error)
 AL = 1 if end of file and record is empty
 AL = 2 if DTA segment was overrun (can't cross 64 - k boundary)

- AL=3 if end of file and partial record is read.
- 15 WRITE-SEQ/write sequential data to the file named by the opened FCB pointed to by DS:DX. One record is transferred from the DTA to the disk and the FCB CURRENT-RECORD field is updated.
returns:AL=0 if transfer was successful(no error)
AL=1 if diskette is full or access denied(read-only file)
AL=2 if end of DTA segment was encountered(overrun error)
- 16 CRRATE-FILE/like OPEN-FILE except the file is truncated to length 0.
- 17 RENAME-FILE/special FCB format is used to change the name of a file
returns:AL=FFH if file not found or new name already exists
AL=0 if no error
- 18 Used internally by DOS
- 19 CURRENT-DISK/return the current default disk in AL (0=A, 1=B, etc.)
- 1A SET-DTA/establish a Disk Transfer Address. In DOS 1.0 and 1.1, all diskette data transfers pass through this area.
expects:DS:DX points to the new buffer for diskette transfers (DTA).
- 1B FAT-DATA/in DOS 1.0 and 1.1, returns DS:BX =address of the file allocation table of the default disk. DOS 2.0 returns address of the FAT identification byte for the default disk.
- 1C DISKTTE-FAT-DATA/like FAT-DATA (function 1BH), but will return information about a specific diskette drive. (DOS 2xx only.)
expects:DL= the number of the drive (0=default, 1=A, 2=B, etc.)
- 1D Used internally by DOS
- 1E Used internally by DOS
- 1F Used internally by DOS
- 20 Used internally by DOS
- 21 READ-RANDOM/read one disk record into the DTA.
expects: RANDOM-RECORD field of FCB to identify the record.
DS:DX points to FCB of an opened file.
- 22 WRITE-RANDOM/like subfunction 21H, but writes a record.
- 23 FILE-SIZE/Search the directory for file named in FCB pointed to by DS:DX, then set the FCB RANDOM-RECORD field to the number of records in the file; i.e., total bytes/ (bytes per record field)

- returns: AL = FFH if file not found, AL = 0 if file found
- 24 SET - RANDOM - FIELD / sets the RANDOM - RECORD field of the FCB pointed to by DS:DX to agree with the CURRENT + BLOCK and CURRENT RECORD fields.
- 25 SET - INT VECTOR / change doubleword address in interrupt vector table.
expects: AL = interrupt type
DS: DX = new address to place in vector table.
- 26 CREATE - PROG - SEGMENT / build a PSP at the paragraph named by DX.
- 27 READ - BLOCK / read one or more records from an open FCB (at DS:DX) into the DTA.
expects: CX = count of blocks to read.
- 28 WRITE - BLOCK / write one or more records from the DTA to the open FCB pointed to by DS:DX.
expects: CX = count of blocks to read.
- 29 PARSE - PILENAME / scan a command line and create an unopenec FCB
Note. this call is not usefu. for command lines with path namen.
expects: DS:SI points to the command line to parse
ES:DI points to an area to build an unopened FCB
AL = bit flags for special options
- 2A GET - DATE / fetch the date according to the DOS internal clock.
returns: CX = year (1980-2099)
DH = month (1-12), DL = day (1-31)
- 2B SET - DATE / set the dste for the DOS internal clock.
expects: parameters as GET - DATE (function 2AH)
returns: AL = FFH if invalid date, AL = 0 if operation successful
- 2C GET - TIME / fetch the time of day according to the DOS internal clock.
returns: CH = hours (0-23), CL = minutes (0-59)
DH = seconds (0-59), DL = hundredths of a second
- 2D SET - TIME / set the time of day for the DOS internal clock.
expects: parameters as GET - TIME (function 2CH)
returns: AL = FFH if invalid date, AL = 0 if operation successful
- 2E VERIFY - SWITCH / sets or resets the flag that DOS checks to see if a diskette write operation should be followed by a verify operation
expects: DL = 0, AL = 1 (turn verify on)
DL = 0, AL = 0 (turn verify off)

.....

The following DOS services are new to PC-DOS version 2.00. Error handling for these functions has been standardized. When a function returns with the Carry Flag set (CF=CY=1), an error has occurred and an error code has been placed into the AX register. The following table names the error condition that is associated with the error code:

error return

code (hex) meaning

.....

- | | |
|----|--|
| 1 | Invalid function number |
| 2 | File not found |
| 3 | Path not found |
| 4 | Too many open files (no handles available) |
| 5 | Access denied |
| 6 | Invalid handle |
| 7 | Memory control blocks destroyed |
| 8 | Insufficient memory |
| 9 | Invalid memory block address |
| A | Invalid environment |
| B | Invalid format |
| C | Invalid access code |
| D | Invalid data |
| E | (not used) |
| F | Invalid drive specified |
| 10 | Attempted to remove the current directory |
| 11 | Not same device |
| 12 | No more files |
-
- | | |
|----|---|
| 2F | GET - DTA / fetch the current DTA into ES: BX. |
| 30 | GET - VERSION / fetch AL = DOS Version number, AH = subrelease. |
| 31 | KEEP - PROCESS / terminate a program, leaving its code and data resident
expects: DX = number of paragraphs to remain resident (starting at the paragraph of the PSP)
AL = exit code (may be interrogated via ERRORLEVEL) |
| 32 | Used internally by DOS. |
| 33 | BREAK CHECK / set or query the level of DOS Ctrl-Break checking.
expects: AL = subfunction: |

- 0 = fetch switch into DL (0 = OFF, 1 = ON).
1 = set switch to DL (0 = OFF, 1 = ON)
- 34 Used internally by DOS.
- 35 GET - INT - VECTOR / fetch the doubleword value of an entry in the interrupt vector table. Vector of the interrupt type in AL is returned in ES:BX. Note: returns original values of vectors changed via SET - VECTOR (function 25H).
- 36 GET - FREE - SPACE / determine the amount of free space on a disk
expects: DL = drive number (0 = default, 1 = A, 1 = B, etc.)
returns: AX = FFFFH if invalid drive number; otherwise.
AX = sectors per cluster
BX = free clusters
CX = bytes per sector
Note: total free bytes = BX * AX * CX
- 37 SWITCHAR / set or query the current command - line parameter delimiter (usually slash ('/ ') ASCII 2FH)
if AH = 0, then set the switch character to the value of DL
if AH = 1, then return the current switch character in DL
note: This is an undocumented DOS service
- 38 COUNTRY - INFO / fetch country - dependent data (currency symbol, etc)
- 39 MKDIR / create a subdirectory.
expects: DS:DX points to ASCIIZ string with path name of directory.
returns: error code in AX if CF = CY (codes: 3 or 5)
- 3A RMDIR / remove a sub - directory.
expects: DS:DX points to ASCIIZ string with path name of directory.
returns: error code in AX if CF = CY (codes: 3 or 5)
- 3B CHDIR / change current directory to that named by ASCIIZ string
expects: DS:DX points to ASCIIZ string with path name of directory.
returns: error code in AX if CF = CY (code: 3)
- 3C CREATE / create a file (supersedes CREATE FILE, function 16H)
Open the file, truncating it to length 0.
expects: DS:DX points to ASCIIZ string with drive, path, and name.
CX = file attribute:
01H = read - only
02H = hidden
04H = system

- 06H = entry contains VOLUME label
 10H = entry is sub - directory
 20H = archive bit (used by BACKUP utility)
 returns: AX = error code if CF = CY (codes:3, 4, or 5); otherwise,
 AX = file handle
- 3D OPEN / open a file (supersedes OPEN - FILE, function 0FH).
 expects: DS:DX points to ASCII string of path and filepec.
 AL = access code:
 0 = open for reading
 1 = open for writing
 2 = open for both reading and writing
 returns: AX = error code if CF = CY (codes:3, 4, 5, or 12); otherwise
 AX = file handle
- 3E CLOSE - HANDLE / close a file (supersedes CLOSE - FILE, function 10H)
 expects: BX is 16-bit file handle returned by OPEN or CREATE
 returns: Ax = error code if CF = CY (code:6)
- 3F READ - HANDLE / read bytes from a file or device.
 expects: BX = 16 - bit file handle.
 CX = count of bytes to read
 DS:DX points to buffer to receive data.
 returns: AX = error code if CF = CY (codes:5 or 6); otherwise
 AX = actual unnumber of bytes read
 note: AX = 0 means end of file reached
- 40 WRITE - HANDLE / write bytes to a file or device.
 expects: BX = 16 - bit file handle
 CX = count of bytes to write
 DS:DX points to buffer containing data to write.
 returns: AX = error code if CF = CY (codes: 5 or 6); otherwise
 AX = actual number of bytes written
 note: error occurred if AX not equal to CX after call
- 41 UNLINK / delete a file from a specified directory.
 expects: DS:DX points to ASCII string of path and filespec
 returns: AX = error code if CF = CY (codes:2 or 5)
- 42 LSEEK / move file read / write pointer according to method and offset
 expects: AL = method code:
 0 = pointer is moved to offset from start of file

- 1 = pointer is increased by offset
 2 = pointer is moved to end of file + offset
 CX:DX = offset value (CX is most significant word)
 BX = 16-bit file handle
- returns: AX = error code if CF = CY (codes:1 or 6); otherwise
 DX:AX = new value of file pointer (DX is most significant)
- 43 CHMOD / change or query file mode (alter the attribute of a file
 expects: DS:DX points to ASCII string of drive, path, and file name
 AL = subfunction code:
 0 = return file attribute in CX
 1 = set file attribute to that named by CX
 returns: AX = error code if CF = CY (codes:3 or 5)
- 44 IOCTL / input and output device control writes control information to
 or reads status from devices that support IOCTL requests.
- 45 DUP - HANDLE / duplicate a file or device handle This function creates
 a new handle that may be used to refer to an existing file or device.
 expects: BX = handle of device or file to duplicate
 returns: AX = error code if CF = CY (codes: 4 or 6); otherwise.
 AX = new handle for same device.
- 46 REDIRECT / force an existing file or device to refer to the data stream
 defined by another file or device.
 expects: CX = handle of current file / device
 BX = handle of file / device that will take over the data I/O.
 returns: AX = error code if CF = CY (code:6)
- 47 GET - DIR / fetch the pathspec of the current default directory.
 expects: DS:SI points to buffer to receive ASCII string pathspec
 DL = drive number (0 = default, 1 = A, etc.)
 returns: AX = error code if CF = CY (code:15)
- 48 ALLOCATE - MEM / request DOS to reserve bytes for a process.
 expects: SX = number of paragraphs requested
 returns: AX = error code if CF = CY (codes:7 and 8)
 and BX is number of free paragraphs; otherwise
 AX:0000 is address of allocated block
- 49 FREE - MEM / return an allocated block of memory back to the common
 pool
 expects: ES is the segment of a block allocated by ALLOCATE - MEM
 returns: AX = error code if CF = CY (codes: 7 or 9)

-
- 4A SETBLOCK / change the size of an allocated block of memory.
 expects: ES is the segment of the block to change.
 BX = the desired new size of the block (in paragraphs)
 returns: AX = error code if CF = CY (codes: 7, 8, or 9)
 and BX is number of available paragraphs
- 4B EXEC / load and execute a program. Create a "child" process.
 expects: DS:DX points to ASCII string of path and filename
 ES:BX points to a parameter block (see below)
 AL = method code
 0 = create a PSP, load, and execute a program
 ES:BX points to parameter block:
 WORD segment address of environment
 DWORD points to command line for program
 DWORD points to first FCB to be passed
 DWORD points to second FCB to be passed
 3 = load only
 ES:BX points to parameter block:
 WORD segment address to load file
 WORD relocation factor
 returns: AX = error code if CF = CY (codes: 1, 2, 5, 8, 10, or 11)
 note: this function may need to load the transient part of
 COMMAND.COM. Therefore, COMMAND.COM must be available
 at the path of the COMSPEC defined in the environment.
- 4C EXIT / terminate the current process, transferring control to the parent
 process. Supersedes TERMINATE (function 0). CS need not point to
 the base segment of the terminating process.
 expects: AL = exit code (may be interrogated by WAIT (function 4DH)
 or by ERRORLEVEL batch commands).
- 4D WAIT / fetch the exit code returned by a terminated process
 returns: AX = return code of most recently exited process
 AH = 0 = normal termination
 1 = terminated by Ctrl - Break
 2 = terminated by critical error handler
 3 = terminated by KEEP (function 31H.)
 AL = 1 - byte return code (ERRORLEVEL code)
 note: This works only once per termination.
- 4E FIND - FIRST / find a file that matches a path and filespec.

- Supersedes SEARCH-FIRST (function 11H)
 expects: DS:DX points to ASCII string of drive, path, and filespec
 CX is the attribute to be used to find the match.
 (see function 3CH)
 returns: AX = error code if CF = CY (codes: 2 or 18); otherwise
 DTA (set via function 1AH) is filled with 43 bytes:
 21 byte---used for subsequent searches
 1 byte---attribute found
 2 bytes---time stamp of file
 2 bytes---date stamp of file
 2 bytes---low word of file size
 2 bytes---high word of file size
 13 bytes---ASCII string "filename.ext", 0
- 4F FIND-NEXT / after a call to FIND-FIRST, this function returns other
 directory entries that also match a global filespec.
 Supersedes SEARCH-NEXT (function 11H)
 expects: must be called after FIND-FIRST (parameters are the same)
 returns: AX = error code 18 and CF = CY when no match is found
- 50 Used internally by DOS.
- 51 Used internally by DOS.
- 52 Used internally by DOS.
- 53 Used internally by DOS.
- 54 GET-VERIFY-STATE / find whether the write-verify switch is OFF
 or ON.
 returns: AL = 0 if verify is OFF; AL = 1 if verify is ON
- 55 Used internally by DOS.
- 56 RENAME / change a directory entry to give a file a new name and
 optionally, move a file to a different sub-directory (same drive).
 expects: DS:DX points to ASCII string of existing path and file
 ES:DI points to ASCII string with new path and filename
 returns: AX = error code if CF = CY (codes: 3, 5, or 17)
- 57 TIME-STAMP / fetch or change the date and time stamp of a file.
 expects: BX = the file handle
 AL = subfunction code:
 0 = change date to DX, time to CX
 1 = fetch date into DX and get time into CX
 returns: AX = error code if CF = CY (codes: 1 or 6)

附录3 CC-DOS其他版本使用简介

3.1 LLDOS双拼双音汉字系统

一、简介

LLDOS采用的是双音编码计算机汉字输入方案，是以汉语拼音为基础的，单字和词汇融为一体的词处理系统，利用双音编码按照汉语拼音（全拼或双拼）击键，不需要耗费眼力的提示进行选择，可输入数万条词汇和六千七百多个单字，双音编码由一整套输入方法组成，可以满足不同人员的使用要求，只要汉语拼音有一定基础，就可以不经培训直接输入汉字，也可以稍加练习达到高速盲打的水平。

二、功能

LLDOS最主要的特点是采用了双音编码汉字输入法，双音编码共有五种输入方法，可分为三大类：

- (1) 双字词汇方式的“双拼双音”和“全拼双音”；
- (2) 单字提示方式的“声韵双拼”和“汉语拼音”；
- (3) “多字词汇”方式。

以上五种方法即可以独立使用还可以互相调用，其中“多字词汇”还可以让用户自己扩充词汇。双音编码采用先进的数据压缩技术，使得含有数万条词汇的码表能够在内存中运行，赢得了时间和空间上的高效率。

除了双音编码方式外，LLDOS还有以下几种输入方式：

- 基本区位码和扩充区位码（ALT/F1）；
- 电报码和反查电报码（ALT/F5）；
- ASCII码（ALT/F10）；

三、运行环境

D型版适用于IBM-PC系列机

I型版为长城0520CH机使

四、系统的文件组成

- (1) D型版（两片盘）

系统盘：

```
COMMAND.COM HSYS.EXE AUTOEXEC.BAT HANZIVD.EXE  
HANZIKD.EXE ANSI.SYS CONFIG.SYS HAN.SYS
```

数据盘：

```
DBINT.COM CHINT.COM BMSJD.BAT BMSJC.BAT CHBM.EXE
```

- (2) I型版（一片盘）

```
COMMAND.COM DBINT.COM REALTIME.EXE REALTIME.TXT  
CHBM.EXE
```

```
PRPT.EXE BMSJC.BAT AUTOEXEC.BAT
```

```
LLINT16I.EXE BMSJD.BAT
```

```
ANSI.SYS GRD.SYS CHINT.COM CONFIG.SY
```

五、系统的安装和启动

(1) D型版 (两片盘)

- 可将两片盘的内容全部拷入硬盘中，直接用硬盘启动。
- 如果没有硬盘或不想拷入硬盘者，请将系统盘插入 A 驱动器，把数据盘插入 B 驱动器，启动即可。

注意：使用之前须修改 AUTOEXEC. BAT 文件。

(2) I 型版 (一片盘)

- 直接将系统盘插入 A：启动；
- 或将软盘上的内容拷入硬盘后，用硬盘启动。

六、使用说明

下面主要把双音编码体系作一个简单说明，双音编码体系包括：汉语拼音，声韵双拼，全拼双音和多字词汇等五种输入方法。

(一) 汉语拼音

汉语拼音简称全拼，用小写字母输入。

编码选择键：ALT/F4 两键同时按下。(与声韵双拼互为反复键)

提示行显示：汉语拼音

汉字输入码：法定的汉语拼音 (全拼音)

打完了一个汉字的汉语拼音以后，如果不能增加字母变成别的拼音音节，则不需要结束键，一般情况下以空格键作为结束键。

打完结束键以后，提示行中便出现同音汉字供您选择。

“，”键为向前翻页，“，”键为向后翻页。

此方法没有记忆负担，适用于一般工程技术人员。

(二) 全拼双音

这里介绍利用汉语拼音输入双字词汇和单个汉字的方法。 编码选择键：ALT/F2 两键同时按下。(与双拼双音互为反复键)

提示行显示：全拼双音

汉字输入码：双字词汇的法定汉语拼音 (全拼音)

拼音结束键：如果第二个汉字的汉语拼音不能再增加字母变成别的拼音，则不需要结束键，一般情况下以空格键做为结束键。

如果在打完第一个汉字的汉语拼音以后，在第二个汉字的汉语拼音还没有打完时，打入一个倒斜杠键，则系统自动转入“汉语拼音”的提示行选择方式，选取一个汉字以后自动回到“拼音双拼”状态，这种方法不需要耗费眼力的提示行选择，可以唯一确定所需要的汉字。

(三) 声韵双拼

声韵双拼是利用声韵双拼代码和提示行输入单个汉字的方法。

编码选择键：ALT/F4 两键同时按下 (与汉语拼音互成反复键)

提示行显示：声韵双拼

汉字输入码：汉语拼音的声母和韵母双拼代码，等长两键，无结束键，声母和韵母的代码表请查阅参考资料。

(四) 双拼双音

双拼双音既利用声韵双拼代码输入双字词汇和单个汉字的方法。

编码选择键: ALT/F 2 两键同时按下, (同拼音双音互成反复键)

提示行显示: 双拼双音

汉字输入码: 双字词汇汉语拼音的声母和韵母的双拼代码, 等长 A 键, 声韵代码请查阅参考资料。

举例: 输入“中央”二字, 需键入 vgye (其中: v zh, e ong, y -y, c ang)

敲入 vgye 四键后, “中”字自动被选入, 如果“央”字也要, 须再按一个空格。

(五) 多字词汇

多字词汇是相对双字词汇而言, 是指 3 至 7 个汉字的词汇, 进入多字词汇有两种方法:

1. 直接进入多字词汇

编码选择键: ALT/F 3 两键同时按下。

2. 从其他方式下转入多字词汇:

只要一个汉字或词汇的输入已经完成, 这时打入一个倒斜杠键, 便进入“多字词汇”的输入状态, 输入一条词汇后, 系统又自动回到原来的输入状态, 如果进入“多字词汇”以后又不想用“多字词汇”输入, 可再打一个倒斜杠。

提示行显示: 多字词汇

词汇输入代码: 对系统提供的通用词汇, 取每个汉语拼音的第一个字母, 一字一码, 最多 7 码 (位)

结束键: 不足 7 位时以空格结束, 达到 7 位时不结束键。

举例: 输入“中国共产党”一词, 打入 zggcd (空格) 即可。

七、参考资料

(1) 双音编码使用手册。(2) 双音编码操作手册。

3. 2 CCDOS V 2.4 词典功能汉字系统

一、简介

CCDOS 2.40 的主要特点是词典管理和词组调用, 这个特点对于那些常用自己专业术语而苦于烦琐输入的用户, 提供了很大便利。

二、功能

除了与 CCDOS 2.10 的一般功能相同外, 还增加了词典管理和词组调用两项新功能。其功能键分别为:

ALT + F 5 —— 词典管理 ALT + F 9 —— 词组 ALT + F 10 —— 输入方式提示

三、运行环境

IBM-PC 及兼容机。

四、系统的文件组成

IBMBIO.COM IBMDOS.COM COMMAND.COM CLIBLD.EXE
LOCK.COM CCLD.EXE CCLIB DICTION.DAT M3070A.EXE
ALL24P.EXE AUTOEXEC.BAT

五、系统的安装和启动

本系统只有一张盘片，使用时只要把盘片插入 A 驱动器，然后冷热启动均可。

六、使用说明

1. 词典管理功能: (按 ALT + F 5 键)

CRT 下方提示行出现:

词典管理: (盘 A) INS: 插入 DEL: 删除 C: 紧缩词典 D: 设置词典盘号
ESC: 退出

(1) 增加新词组:

按 INS 键, CRT 提示行出现:

新词组名: KHPZ (“KHPZ” 为用户敲入的新词组名), 然后回车。

CRT 提示行变换为:

拼音: _ 新词组: KHPZ: 科海培中 | 回车结束

上一行中的“科海培中”四个字是由用户按拼音方式逐个送入, 用户可以根据自己的需要定义任何词组, 结束后按回车键, 新词组即存入盘内, 词组定义完毕, 按 ESC 键退出, 系统回到词典管理提示行。

(2) 删除老词组:

按 DEL 键, 提示行出现:

词组名: KHPZ (用户敲入) 然后回车, CRT 出现:

词组名: KHPZ: 科海培中 删除吗? (Y/N)

删除打 Y, 否则打 N

2. 词组功能: (按 ALT + F 9)

CRT 提示:

词组: _

如同选择汉字一样, 给出自己定义的词组名称, 如刚才举例的: “KHPZ”

词组: KHPZ 科海培中 [KHPZ] 回车选择好

此时回车该词组就进入你所编辑的文本中, 系统回到进入词组方式前的状态。

注: 按一次 ALT + F 9 只能选择一个词组, 如需再选, 还需再按 ALT + F 9 键盘。

七、参考资料

无。

3.3 CC DOS V 2.13 16/24 点阵汉字系统

一、简介

本系统共 8 片软盘, 包括十六点阵及二十四点阵字库, 其中二十四点阵字库包括: 宋、仿宋、黑、楷等四种字体, 二十四点阵字库占七张盘片。

二、功能

能打印宋, 仿宋, 黑及楷等多种字体, 支持实线报表打印。

三、运行环境

IBM PC/XT 及兼容机。

四、系统的文件组成

0 # 盘 (系统盘):

AUTOEXEC. BAT 启动批处理文件 WORK1.COM 建二级字库扇区索引程序
 MENU.COM 11/16行显示选择菜单 FILE4.COM 检索16*16字库中断程序
 FILE2.COM 为一级字库申请内存程序 HZK1 16*16点阵一级字库
 CCP1.EXE 调一级字库及拼音码表, 显示和键盘处理模块程序 (11行显示)
 C16P1.EXE 功能同上 (16行显示)
 WORK24.COM 建24*24字库扇区索引程序 FILE24.COM 检索24*24字库
 中断程序

五、安装及启动

1. 安装

本系统由 IBM-PC DOS 2.10支持, 安装步骤:

- A. 在硬盘根目录下装入 PC DOS 2.10系统模块。
- B. 将本系统 0 # 盘插入驱动器 A: A>COPY *, * C;
- C. 装载字库: 将 7 # 盘插入驱动器 A: , A>ZK (CR), 按提示操作即可, 最后将 7 # 盘上的 HZK 2 拷贝到 C 盘。
- D. 显示二级库和 24×24点阵字库均须建扇区索引文件: HZK 2. IND长777字节, HZK 24T. INT长597字节, HZK 24S. IND—HZK 24K. IND长度均为2856字节。

建立索引文件: 将 0 # 盘插入驱动器 A:

A>WORK 1 (CR) 再 A>WORK 24 (CR) 若某索引文件长度不为以上数值, 则应删除重装。

2. 启动直接用硬盘冷、热启动均可, 系统自动装入字库, 挂打印机默认为24点阵打印。

六、使用说明

1. 功能键

常用功能键同 CCDOS 2.10例如 ALT + F 2 区位码输入……

本系统特有的功能键:

CTRL + F 1	转16×16点阵打印	CTRL + F 2	转24×24点阵打印
CTRL + F 4	改变纸宽 (1—306)	CTRL + F 5	特殊打印功能开关
CTRL + F 10	改变字号和行距		

2. 特殊打印功能

标识符: \ (ASCII码为60H, 与~共在一键)

格式: \&行距 功能: 置行距 (1—255, 单位 1/120英寸)

 \@字号\ 置字号 (A—P)

3. 字号

16×16点阵 A—P大小及转90度角共16种字形。

24×24点阵 A—P

其中 A, B, C, D为宋体大小四种型号; E, F, G, H为仿宋体大小四种型号
 I, J, K, L为黑体大小四种型号; M, N, O, P为楷体大小四种型号。

4. 打印参考数据

90年11月27日

		M2024	M3079
16×16点阵:	汉字占16列 每行	136个	153个
	字符占8列 每行	272个	306个
24×24点阵:	汉字占24列 每行	90个	102个
	字符占12列 每行	100个	204个

可使两行竖线相连的最大行距:

16×16	A—H型	12	10
16×16	I—P型	24×24	10
			16

选择打印机: 将7#盘插入驱动器A: A:\QDCX(CR) (替换C盘驱动程序, 原版为2024)

注: CCDOS启动时默认行距为20, 其他详细说明请参阅CCDOS 2.13使用说明。

七、参考资料

CCDOS 2.13使用说明 (0#盘2.13S文件)

参 考 资 料

- <1> Peter Abd. Assembler for IBMPC/XT Reston Publishing CO.1984
- <2> 钱培德著: CCB10S分析
- <3> CCDOS 2.13使用说明 (0#盘2.13S文件)
- <4> IBMPC硬件技术参考手册: ROM Bios Listing
- <5> DOS3.0操作说明

[General Information]

书名=PC-DOS系统汉化和CC-DOS分析与使用

作者=

页数=1000

SS号=

DX号=

出版日期=

出版社=

书名
前言
目录
目录

- 第一章 PC-DOS 2.0系统分析
 - 1.1 概论
 - 1.2 PC-DOS的组成与功能
 - 1.3 PC-DOS的命令处理实例分析
- 第二章 PC-DOS操作说明
 - 2.1 系统启动
 - 2.2 软盘的初始化和复制
 - 2.3 键盘命令
 - 2.4 文件及文件操作命令
- 第三章 CC-DOS的概述和初步分析
 - 3.1 系统概述
 - 3.2 CC-DOS的自举原理
 - 3.3 CC-DOS中汉字显示输出的实现
 - 3.4 CC-DOS中汉字输入的实现
 - 3.5 CC-DOS中汉字打印输出的实现
- 第四章 CC-BIOS的引导过程
 - 4.1 概述
 - 4.2 BIOS的结构与修改
 - 4.3 CC-DOS的自举过程
 - 4.4 CC-BIOS的引导过程
 - 4.5 对FILE1.EXE的分析
 - 4.6 对CCCC.EXE的分析
 - 4.7 CC-BIOS打印模块的形成
 - 4.8 引导源程序
- 第五章 CC-BIOS的键盘管理模块
 - 5.1 引言
 - 5.2 键盘中断的处理
 - 5.3 键盘输入请求程序的结构
 - 5.4 代码转换和扫描表
 - 5.5 汉字输入功能的实现
 - 5.6 读入字符处理模块
 - 5.7 输入码处理模块
 - 5.8 键盘管理模块的源程序
- 第六章 CC-BIOS的显示器控制模块
 - 6.1 总述
 - 6.2 显示器控制模块的结构
 - 6.3 几个重要的工作区
 - 6.4 CRT的初始化
 - 6.5 光标的产生和定位
 - 6.6 屏幕滚动的实现
 - 6.7 读字符的实现

- 6.8 字符显示的实现
- 6.9 TTY显示方式的实现
- 6.10 提示行管理模块
- 6.11 显示器控制模块的源程序
- 第七章 CC-BIOS的打印驱动模块
 - 7.1 总述
 - 7.2 打印屏幕驱动程序
 - 7.3 打印机驱动程序的组成和输出字型
 - 7.4 打印机驱动程序的工作区
 - 7.5 打印机驱动程序的总体
 - 7.6 17类中断程序的0号功能块
 - 7.7 图形态字符处理模块
 - 7.8 打印驱动模块的源程序
- 第八章 CC-BIOS的高级打印驱动模块
 - 8.1 概述
 - 8.2 高级打印驱动模块的生成
 - 8.3 21类中断程序
 - 8.4 17类中断程序
 - 8.5 17类中断程序的0号功能块
 - 8.6 非ESC态处理模块
 - 8.7 机内码处理模块
 - 8.8 高级打印驱动模块的源程序
- 第九章 CC-DOS使用说明
 - 9.1 系统启动
 - 9.2 汉字输入
 - 9.3 汉字打印输出
 - 9.4 在CC-DOS支持下的软件使用
- 附录1 PC-DOS软中断程序简介
- 附录2 PC-DOS INT21H系统调用简介
- 附录3 CC-DOS其它版本使用简介