

Borland C++ 与 Turbo C 开发汉字应用软件技术

王小华 著



陕西电子编辑部

前 言

自1990年Borland 公司推出C++语言以来, 1991年2月推出了Borland C++ 2.0 和 Turbo C 2.0. 最近, Borland公司又推出了功能更强的Borland C++ 3.0. 它的突出优点是: 比以前版本有更快的编译速度, 几乎是Turbo C 2.0 版本的两倍; 增加了对应用程序代码的优化功能; 增强了 Windows 资源的编辑工具; 提供了更丰富的函数资源. 由此, Borland C++ 与 Turbo C 被称为九十年代最流行的编程软件之一。

时代在发展, 微机已越来越普及化, 在国内, 对汉字应用软件的需求也越来越强烈. 所以开发汉字应用软件有着广泛的前景. 本书主要讲述了 Borland C++ 与 Turbo C 在开发汉字应用软件方面的一些常用方法和技巧, 旨在充分利用“九十年代最流行的编程软件——C语言”开发有中国特色的汉字应用软件. 通过对汉字的输入、显示、打印等原理的分析, 讲述了实现这些功能的C程序设计方法, 尤其是对较为流行的汉字菜单的设计, 给出了详细的介绍. 本书的最大特点就是讲述了丰富的子程序设计方法, 并且首次在国内公开了一份汉字编辑软件的源程序。

目 录

第一章 DOS与CC-DOS系统分析	(1)
§1.1 DOS的组成与功能	(1)
§1.2 DOS的丰富资源	(3)
§1.3 CC-DOS的组成与功能	(11)
§1.4 CC-DOS的汉字输入原理	(14)
§1.5 CC-DOS的汉字显示原理	(16)
§1.6 CC-DOS的汉字打印原理	(21)
第二章 键盘管理的C程序设计	(25)
§2.1 对CC-BIOS的INT 16H的认识	(25)
§2.2 getkey() 函数	(25)
§2.3 Spkey() 函数	(26)
§2.4 Getkey() 函数	(26)
§2.5 条件分支语名	(27)
§2.6 开关分支语名	(30)
§2.7 几个应用实例	(32)
第三章 显示管理的C程序设计	(35)
§3.1 对CC-BIOS INT 10H的认识	(35)
§3.2 显示器的初始化	(35)
§3.3 光标的设置	(38)
§3.4 汉字的显示	(42)
§3.5 读字符的方法	(44)
§3.6 显示汉字串的函数	(45)
§3.7 屏幕滚动的实现	(48)
§3.8 颜色的设置	(50)
§3.9 提示行管理函数	(51)
第四章 用户界面的设计	(53)
§4.1 画点函数	(53)
§4.2 画线函数	(55)
§4.3 窗口技术	(57)
§4.4 条形菜单	(58)
§4.5 获得输入字符串的函数	(61)
§4.6 下拉式菜单的设计	(65)
§4.7 亮条的设计	(67)

§4.8 亮条移动的设计	(68)
§4.9 一个循环式下拉菜单组的程序设计	(69)
第五章 汉字编辑软件的设计	(77)
§5.1 初始化	(77)
§5.2 文件操作	(79)
【1、读取文件】	(79)
【2、文件存盘】	(82)
【3、重新命名】	(85)
【4、重新编辑】	(85)
【5、存盘退出】	(85)
【6、退出编辑】	(86)
§5.3 块操作	(86)
【1、定义块首】	(86)
【2、定义块尾】	(89)
【3、块复制】	(90)
【4、块移动】	(92)
【5、块删除】	(93)
【6、取消块】	(96)
【7、读块】	(97)
【8、写块】	(99)
§5.4 光标移动	(101)
【1、上移一行】	(101)
【2、下移一行】	(103)
【3、左移一列】	(104)
【4、右移一列】	(105)
【5、移至行首】	(105)
【6、移至行尾】	(106)
【7、移至块首】	(107)
【8、移至块尾】	(109)
§5.5 字符的处理	(110)
【1、回车处理】	(113)
【2、删除一字】	(115)
【3、删除光标前一个字符】	(117)
【4、删除一行】	(119)
【5、删至行尾】	(120)
§5.6 卷页滚动	(121)
【1、上翻一页】	(122)
【2、下翻一页】	(123)

【3、光标移至文件首部】	(125)
【4、光标移至文件尾部】	(126)
\$5.7 寻找替换	(126)
【1、寻找字符】	(127)
【2、替换字符】	(128)
【3、删除字符】	(131)
【4、继续寻找】	(133)
\$5.8 编辑控制	(134)
【1、设置宽度】	(134)
【2、插入与改写的转换】	(135)
【3、版面重排】	(136)
第六章 汉字编辑软件的有关函数介绍	(141)
Bar()	(141)
Block()	(142)
Putch()	(142)
Savefile()	(143)
Getfile()	(143)
Cmpk()	(148)
putstr()	(148)
Ptline()	(149)
Strr()	(149)
Putsheet3()	(150)
Putcur()	(151)
setbottom()	(151)
curx()	(152)
lastpt()	(152)
nextpt()	(152)
getkey()	(153)
Findy()	(153)
Klen()	(153)
Gotoxy()	(154)
Ptinform()	(154)
Graph()	(154)
Putma()	(155)
Getkey()	(155)
pause()	(155)
Wherex()	(156)
Wherey()	(156)

Uproll()	(156)
Downroll()	(156)
Cursor()	(157)
Putc()	(157)
Putcm()	(158)
Clreol()	(158)
Point()	(158)
Color()	(158)
Temp()	(159)
Gettext()	(160)
Puttext()	(160)
Line()	(161)
Col()	(161)
Clrblock()	(161)
Stripe()	(162)
String()	(162)
Cursoroff()	(163)
Cursoron()	(163)
Ptm()	(163)
Clrnote()	(164)
Inform()	(164)
Content()	(164)
Menu()	(165)
Editor()	(166)
第七章 汉字编辑软件SKY的源程序	(196)
§7.1 SKY0.C	(196)
§7.2 SKY1.C	(211)
§7.3 SKY2.C	(213)
§7.4 SKY3.C	(214)
§7.5 SKY4.C	(216)
§7.6 SKY5.C	(217)
§7.7 SKY6.C	(223)
§7.8 SKY7.C	(234)
§7.9 汉字编辑软件的头文件 SKY.H	(262)
附一 汉字编辑软件SKY的介绍	(266)
附二 键值表一	(268)
附三 键值表二	(270)

第一章 DOS与CC-DOS系统分析

§1.1 DOS的组成与功能

PC-DOS是个人计算机磁盘操作系统(Personal Computer Disk Operating System)的缩写。它是目前计算机通用的操作系统,由于采用分散模块结构,且配置灵活,资源丰富,深得用户欣赏。

PC-DOS主要由四部分组成:引导部分、BIOS.COM、DOS.COM、COMMAND.COM。引导块装在盘的0道1扇区,系统启动时,执行ROM起始地址FFFF:0000的启动程序。启动程序首先对系统进行初始化和自检,然后进入ROMBIOS的软中断(INT 19H)检查驱动器状态,若驱动器有盘插入,则读入DOS引导块,否则进入ROM-BIOS。读入引导块放在起始地址为0000:7C00处,且按一定的过程执行。其执行顺序为:查看目录块(0道5扇区),是否有BIOS.COM与DOS.COM两个文件(这两个文件以隐含的形式放在磁盘上,DOS的DIR命令不会发现,但可以通过PC-TOOLS或者别的应用软件发现)。若有,则将两文件读入60段0偏移位置处,并且转到6000:0000执行BIOS.COM。否则,等待插入系统盘。现在,一般微机配有两个驱动器与一个硬盘,在系统引导自举的过程中,先查看A盘,若无系统文件,再查看B与C,以至最终没有发现系统文件时,屏幕显示提示信息且进入等待状态。

在系统引导的过程中,若由于自检出错,则显示一系列错误信息,并导致死机现象,几种常见的出错代码及意义如下:

- 101 —— 系统板错。
- 201 —— RAM检查出错。
- 301 —— 键盘出错。
- 601 —— 软盘出错。
- 1701 —— 硬盘出错。
- 1801 —— 扩展箱出错。
- ROM —— ROM检查出错。

系统文件BIOS.COM是基本输入输出处理程序,起初始化作用。首先,它建立一张磁盘参数表,初始化串行口(RS232)与并行口(LPT1:打印机口),建立1,3,4,1BH的口断入口,为DOS建立通讯区,将DOS从当前位置移至BF段,完成这些操作之后,将自动覆盖掉部分BIOS程序,确定软盘驱动器数及RAM大小,调DOS初始化程序,建立用户区段,填写中断25、26H的入口。然后将COMMAND文件读到用户区100以后处,并建立磁盘数据传送区(DTA)【偏移为80H】,执行COMMAND。

DOS.COM文件是整个操作系统的核心,它提供了丰富的子程序,这些程序包含在中断INT 21H中,通过调用这些子程序,可以实现输入输出、文件管理、目录管理、内存管理等功能。

COMMAND.COM 文件可以看作是PC-DOS 的外壳,是用户与操作系统的接口,承担着分析和解释执行用户的各种命令。它主要由三部分组成:

(1) 常驻内存部分

该部分在系统被引导时,装入内存底部,它包含的子程序用以处理Ctrl_C与 Ctrl_Break 命令,处理磁盘输入输出错误及其它暂存程序的执行和退出。如我们常常看到的信息:Abort, Retry, Ignore? (放弃、再试一试,忽略?),便是COMMAND.COM 常驻内存部分查错而发出的信息。另外,常驻内存部分还包括必要时重新装入COMMAND.COM 暂存部分的指令。

(2) 初始化部分

在系统被引导时,该部分被装在常驻部分之后,当有AUTOEXEC.BAT 批文件存在时,则对该文件进行处理,AUTOEXEC.BAT 是用户所编的一个批处理文件,它是一系列命令的组合,在系统被引导时,该文件首先被执行,执行完后,若再执行其它程序,则暂存该部分的内存随即被覆盖掉。

(3) 暂存部分

COMMAND.COM 的暂存部分装在内存的高端,这部分程序所占用的内存随时可能被其它程序覆盖。提示符A> 或C> 等,即由该部分产生。由键盘或批文件读入的命令,亦由该部分解释并使之得到执行。在解释命令时,只对含有EXE、COM为扩展名的文件加载并执行它,对其它含有非EXE、COM为扩展名的文件则不作任何处理。COMMAND.COM 有一个再定位装入程序,它决定将用户的程序装在何处执行。当执行用户程序时,暂存部分在COMMAND.COM 常驻部分之后构造一个程序前缀PSP (Program Segment Prefix),然后将用户的可执行程序 (COM或COM文件) 装在PSP的下边,离PSP偏移100H (PSP为256字节长),接着将系统控制权交给用户程序。当用户程序执行完后,COMMAND.COM收回控制权。

COMMAND.COM 能接收和处理用户的三种命令:

【1】 内部命令

它是系统提供的固有的命令,这些命令的处理程序在COMMAND.COM 的暂存部分,如COPY, TYPE, DIR, REN, DEL, ERASE, VER等。

【2】 外部命令

外部命令以可执行文件 (COM或EXE文件) 的形式存放在磁盘上,执行这些命令时COMMAND.COM 先将该文件调入程序暂存区,然后开始执行它。执行完这些程序后,所占用的内存空间全部释放,控制权交给COMMAND.COM。

【3】 批处理命令

批处理文件是一个包含多个命令 (外部命令与内部命令) 集合的文件,以文本方式存放在磁盘上,具有扩展名BAT。执行该文件时,按批文件的命令组合顺序执行,执行完一个文件后,COMMAND.COM 收回控制权,然后再执行第二个、第三个……,直到将批命令全部执行完。

COMMAND.COM 对用户键入的命令执行时, 首先看是否为内部命令, 若是则执行之。否则看是否COM文件, 否则看是否EXE文件, 否则看是否BAT文件, 若均未找到, 则显示 Bad command or file name 的信息。

§1.2 DOS 的丰富资源

DOS 提供了丰富的中断子程序的功能调用子程序, 这些为程序的开发带来了极大的方便, 认识和利用这些资源, 对学习任何一种高级语言都有很大的帮助。

下面介绍几种常用的中断:

【1】 INT 5H

屏幕打印中断。调用该中断可实现屏幕上的字符打印, 若要实现屏幕上的图形打印, 需修改该中断向量, 自行设计一图形打印程序。

【2】 INT 10H

显示管理中断。该中断提供了如下子功能:

【1】功能 00H: 置显示方式。

入口: AL = 显示方式, 其值与下列各方式相对应

- 00H: 40×25 黑白字符方式
- 01H: 40×25 彩色字符方式
- 02H: 80×25 黑白字符方式
- 03H: 80×25 彩色字符方式
- 04H: 320×200 4色图形方式 (调色板0)
- 05H: 320×200 4色图形方式 (调色板1)
- 06H: 640×200 2色图形方式
- 07H: 单色字符卡
- 08H: 160×200 16色图形方式 (pcjr)
- 09H: 320×200 16色图形方式 (pcjr)
- 0AH: 640×200 4色图形方式 (pcjr)
- 0DH: 320×200 16色图形方式 (EGA)
- 0EH: 640×200 16色图形方式 (EGA)
- 0FH: 640×350 单色图形方式 (EGA)
- 10H: 640×350 4色或16色图形方式 (EGA)

(取决于可用的RAM)

【2】功能 01H: 置光标类型。即选择字符方式下硬件光标的开始和结束行。

入口: CH = 光标开始行

CL = 光标结束行

注释: 在字符方式下, 光标是由硬件产生的, 并不能用软件的方法消除光标, 光标开始和结束行的值由显示卡的类型决定。关闭光标的方法是: 将光标定位到一个不可显示的位置, 如 (X, Y) = (0, 25)。

【3】功能 02H : 置光标位置。

入口: BH=页号

DH=行置 (即Y坐标)

DL=列置 (即X坐标)

【4】功能 03H : 在字符方式下, 读光标位置。

入口: BH=页号

返回值: CH=光标开始行

CL=光标结束行

DH=光标所在行值 (Y坐标值)

DL=光标所在列值 (X坐标值)

【5】功能 04H : 读光标当前状态和位置。

返回值: CH=象素行值 (即Y坐标值, 0-199)

BX=象素列值 (即X坐标值, 0-319或0-639)

DH=字符行值 (即Y坐标值, 0-24)

DL=字符列值 (即X坐标值, 0-79或0-39)

【6】功能 05H : 选择活动的显示页。

入口: AL=页数, 由下列方式确定

0-7: 方式00H和01H (对CGA)

0-3: 方式02H和03H (对CGA)

0-7: 方式02H和03H (对EGA)

0-7: 方式0DH (对EGA)

0-3: 方式0EH (对EGA)

0-1: 方式0FH (对EGA)

0-1: 方式10H (对EGA)

【7】功能 06H : 屏幕上窗口内容上滚。

入口: AL=滚动行数 (如果AL=0, 则整个窗口内容被消除)

BH=空白区的属性

CH=窗口左上角的Y坐标

CL=窗口左上角的X坐标

DH=窗口右下角的Y坐标

DL=窗口右下角的X坐标

【8】功能 07H : 屏幕上窗口内容下滚。

入口: AL=滚动行数 (如果AL=0, 则整个窗口内容被消除)

BH=空白区的属性

CH=窗口左上角的Y坐标

CL=窗口左上角的X坐标

DH=窗口右下角的Y坐标

DL=窗口右下角的X坐标

【9】功能 08H：读光标位置的字符和属性。

入口：BH=显示页

返回值：AH=字符属性

AL=字符的ASCII码

【10】功能 09H：在当前光标位置写字符和属性。

入口：AL=字符的ASCII码

BH=显示页

BL=字符方式的属性与图形方式的颜色

CX=写字符的重复次数

【11】功能 0AH：不改变属性，只在光标位置写字符。

入口：AL=字符的ASCII码

BH=显示页

BL=图形方式的属性

CX=写字符的重复次数

【12】功能 0BH：置彩色调色板。

入口：BH=调色板

BL=颜色值

注 释：此功能只对标准彩色图形卡的中分辨率彩色图形显示方式（04H）

有效，在pcjr中只对方式04H至06H和08H至0AH有效。

在图形方式下，若BH=0，BL=背景和色板的颜色（0-15）。

在字符方式下，BL控制板的颜色；字符的背景色由字符的高四位

控制。若BH=1，BL=选择的调色板（在标准卡中为0和1）。

其值如下：

调色板	象素值	颜 色
0	0	与背景相同
0	1	绿 色
0	2	红 色
0	3	棕 色
1	0	与背景相同
1	1	青 色
1	2	洋 红 色
1	3	白 色

【13】功能 0CH：在提定的图形坐标位置画一点。

入口：AL=象素值。

对显示方式04H和05H，合法的象素值是0-3；对显

示方式06H，合法的象素值是0-1；若AL的第7位为

1, 新的象素值将与当前象素值的内容异或。

CX=列数(坐标X值)

DX=行数(坐标Y值)

X的取值范围取决于当前所选择的图形方式, 在方式04H、05H、0DH下, 取值范围为0-329, 在方式06H和0EH下, 取值范围为0-639。Y在方式04H、05H、0DH、0EH下, 取值范围为0-199, 在方式和0FH和10H下, 取值范围为0-349。

【14】功能 0DH: 取指定坐标位置的当前象素值。

入 口: CX=列数(坐标X值)

DX=行数(坐标Y值)

返回值: AL=象素值(见功能0CH的有关说明)。

【15】功能 0EH: 以电传打字方式(TTY方式)在屏幕上写字符。

入 口: AL=字符的ASCII码

BH=字符方式的显示页

BL=图形方式的背景色

注 释: 该功能实现对某些控制符的识别, 如铃响(07H)、回车(0DH)、换行(0AH)、制表(09H), 同时对这些控制符作相应的处理。当光标在屏幕最后一行时, 在回车换行和自动换行中实现屏幕上滚。

【3】INT 13H

磁盘管理中断。该中断提供了如下子功能:

【1】功能 00H: 重新启动软盘系统。

【2】功能 01H: 取软盘系统状态。

返回值: AH=状态字节。各位的意义如下:

- 第0位: 非法命令
- 第1位: 盘写保护
- 第2位: 申请的扇区未找到
- 第3位: DMA超载运行
- 第4位: 读盘数据错
- 第5位: 控制器错
- 第6位: 随机转移失败
- 第7位: 盘超时

【3】功能 03H: 读软盘。

入 口: AL=传送扇区数(1-9)

ES: BX=用户盘I/O缓冲段地址: 偏移地址

CH=磁道号(0-39)

CL=扇区号(1-9)

DH=磁头号(0-1)

DL=驱动器号(0-3)

返回值: 成功时, 进位清0, AH=0, AL=实际传送的磁盘扇区数。

失败时, 进位标志置1, AH=状态字节, 其各位的意义如下:

第0位: 送给驱动器的命令为非法命令

第1位: 盘写保护

第2位: 申请的扇区未找到

第3位: DMA超载运行

第4位: 读盘数据错

第5位: 控制器错

第6位: 随机转移失败

第7位: 盘超时

【4】功能 03H : 写盘。

入口: AL=传送扇区数(1-9)

ES: BX=用户盘I/O缓冲的段地址: 偏移地址

CH=磁道号(1-9)

CL=扇区号(1-39)

DH=磁头号(0-1)

DL=驱动器号(0-3)

返回值: 成功时, 进位清0, AH=0, AL=实际传送的磁盘扇区数。

失败时, 进位标志置1, AH=状态字节, 其各位的意义如下:

第0位: 送给驱动器的命令为非法命令

第1位: 盘写保护

第2位: 申请的扇区未找到

第3位: DMA超载运行

第4位: 读盘数据错

第5位: 控制器错

第6位: 随机转移失败

第7位: 盘超时

【5】功能 04H : 确证软盘指定扇区的地址。

入口: AL=确证扇区数(1-9)

CH=磁道号(0-39)

CL=扇区号(1-9)

DH=磁头号(0-1)

DL=驱动器号(0-3)

返回值: 成功时, 进位清0, AH=0, AL=实际传送的磁盘扇区数。

失败时, 进位标志置1, AH=状态字节, 其各位的意义如下:

第0位: 送给驱动器的命令为非法命令

- 第1位: 盘写保护
- 第2位: 申请的扇区未找到
- 第3位: DMA超载运行
- 第4位: 读盘数据错
- 第5位: 控制器错
- 第6位: 随机转移失败
- 第7位: 盘超时

【6】功能 05H : 格式化盘磁道 (软盘地址域和数据扇区的初始化)。

入口: ES: BX = 地址区域列段地址: 偏移地址

【4】 INT 16H

键盘管理中断。该中断提供了如下子功能:

【1】功能 00H : 从键盘读一字符, 同时返回该键盘的扫描码。

返回值: AH = 键盘扫描码

AL = 字符的ASCII码 (只对键盘上的字符和数字键)

【2】功能 01H : 读键盘状态。

返回值: 如果键等待被输入, 则清零标志且有AH = 扫描码, AL = 字符的ASCII码。如果没有键等待, 则零标志置1。

【3】功能 02H : 返回键盘标准, 它可以描绘键盘上每个键的状态。

返回值: AL = ROM BIOS键盘的标志字节, 各位的意义如下:

第0位: RIGHT-SHIFT键按下

第1位: LEFT-SHIFT键按下

第2位: CTRL键按下

第3位: ALT键按下

第4位: SCOLL LOCK打开

第5位: NUM LOCK打开

第6位: CAPS LOCK打开

第7位: INSERT打开

【5】 INT 17H

打印管理中断。该中断提供了如下子功能:

【1】功能 00H : 送一字符到指定的打印口, 同时返回打印机的状态。

入口: AL = 字符的ASCII码

DX = 打印机号 (0-2)

返回值: AH = 打印机状态字节, 其各位的意义如下:

第0位: 超时

第1位: 无意义

第2位: 无意义

第3位: I/O出错

第4位: 打印机选择

- 第5位: 无纸
- 第6位: 确认
- 第7位: 打印机不忙

【2】功能 01H : 初始化打印口, 同时返回打印机的状态。

入口: DX=打印机号(0-2)

返回值: AH=打印机状态字节, 其各位的意义如下:

- 第0位: 超时
- 第1位: 无意义
- 第2位: 无意义
- 第3位: I/O出错
- 第4位: 打印机选择
- 第5位: 无纸
- 第6位: 确认
- 第7位: 打印机不忙

【3】功能 02H : 测试打印状态。

入口: DX=打印机号(0-2)

返回值: AH=打印机状态字节, 其各位的意义如下:

- 第0位: 超时
- 第1位: 无意义
- 第2位: 无意义
- 第3位: I/O出错
- 第4位: 打印机选择
- 第5位: 无纸
- 第6位: 确认
- 第7位: 打印机不忙

【6】INT 20H

程序结束中断。

入口: CS=PSP的分段值

【7】INT 21H

功能调用中断, 参考有关功能调用手册。

【8】INT 22H

建立并管理程序中止。

【9】INT 27H

结束并驻留内存。

中断所提供的丰富功能, 给编程带来了很大的方便, 除了用汇编语言能直接调用这些中断之外, 其它高级语言如C、PASCAL等, 亦能方便地调用系统所提供的中断资源。准确熟练地掌握调用中断技术, 不失为编程的一条捷径。

CC-DOS就是在PC-DOS的基础之上修改了部分中断向量，添加了部分适应汉字处理的中断服务子程序。通过DEBUG可以看出，CC-DOS是在PC-DOS的基础上修改了INT 10H与INT 16H两个中断向量。有的版本也修改了INT 5H与INT 17H两个中断向量。查看中断向量的方法很简单，在DEBUG状态下，用D0000:0000命令，显示0段0偏移128字节内容。如：

在DOS状态下

C>debug

-d 0000:0000 0100

```
0000:0000 EA 56 70 02 C0 01 70 00-16 00 BC 0B 00 9F 70 00
0000:0010 5C 07 70 00 54 FF 00 F0-23 FF 00 F0 23 FF 00 F0
0000:0020 E1 02 00 9F 24 01 BC 0B-23 FF 00 F0 23 FF 00 F0
0000:0030 23 FF 00 F0 B1 03 BC 0B-39 04 BC 0B 5C 07 70 00
0000:0040 65 F0 00 F0 4D F8 00 F0-41 F8 00 F0 01 10 70 00
0000:0050 39 E7 00 F0 59 F8 00 F0-2E E8 00 F0 D2 EF 00 F0
0000:0060 00 00 00 F6 8A 19 70 00-6E FE 00 F0 56 07 70 00
0000:0070 49 FF 00 F0 A4 F0 00 F0-22 05 00 00 00 00 00 00
0000:0080 3F 14 70 02 C0 01 00 9F-F4 02 AA 0D 2F 03 AA 0D
0000:0090 BC 02 AA 0D DC 15 70 02-1F 16 70 02 68 63 70 02
0000:00A0 45 14 70 02 9E 06 70 00-45 14 70 02 45 14 70 02
0000:00B0 45 14 70 02 45 14 70 02-81 02 8A 0C B7 18 70 00
0000:00C0 EA 46 14 70 02 14 70 02-45 14 70 02 45 14 70 02
0000:00D0 45 14 70 02 45 14 70 02-45 14 70 02 45 14 70 02
0000:00E0 45 14 70 02 45 14 70 02-45 14 70 02 45 14 70 02
0000:00F0 45 14 70 02 45 14 70 02-45 14 70 02 45 14 70 02
0000:0100 59
```

在CCDOS状态下

C>debug

-d 0000:0000 0100

```
0000:0000 EA 56 70 02 C0 01 70 00-16 00 BC 0B 00 9F 70 00
0000:0010 5C 07 70 00 54 FF 00 F0-23 FF 00 F0 23 FF 00 F0
0000:0020 F5 1F BD 4D 24 01 BC 0B-23 FF 00 F0 23 FF 00 F0
0000:0030 23 FF 00 F0 B1 03 BC 0B-39 04 BC 0B 5C 07 70 00
0000:0040 D0 11 BD 4D 4D F8 00 F0-41 F8 00 F0 01 10 70 00
0000:0050 39 E7 00 F0 59 F8 00 F0-0C 25 BD 4D D2 EF 00 F0
0000:0060 00 00 00 F6 8A 19 70 00-6E FE 00 F0 56 07 70 00
0000:0070 49 FF 00 F0 A4 F0 00 F0-22 05 00 00 00 00 00 00
0000:0080 3F 14 70 02 C0 01 00 9F-F4 02 AA 0D 2F 03 AA 0D
0000:0090 BC 02 AA 0D DC 15 70 02-1F 16 70 02 68 63 70 02
```

```

0000:00A0 45 14 70 02 9E 06 70 00-45 14 70 02 45 14 70 02
0000:00B0 45 14 70 02 45 14 70 02-81 02 8A 0C B7 18 70 00
0000:00C0 EA 46 14 70 02 14 70 02-45 14 70 02 45 14 70 02
0000:00D0 45 14 70 02 45 14 70 02-45 14 70 02 45 14 70 02
0000:00E0 45 14 70 02 45 14 70 02-45 14 70 02 45 14 70 02
0000:00F0 45 14 70 02 45 14 70 02-45 14 70 02 45 14 70 02
0000:0100 59

```

可以看出,在DOS下,中断INT 10H的向量为F000:F065

INT 16H的向量为F000:E821

在CC-DOS下INT 10H的向量为4DBD:11D0

INT 16H的向量为4DBD:25C0

A2

§1.3 CC-DOS的组成与功能

CC-DOS (China Character-Disk Operating System) 是中文磁盘操作系统的简称, 配置在大多数PC机上, 它是在PC-DOS的基础上设计的, 主要增加了汉字处理功能, 支持在PC-DOS下运行的多种软件。

CC-DOS在字符的输入与显示方面, 对PC-DOS作了一定的改进, 它修改了ROM-BIOS所提供的软中断子程序中的两个中断向量INT 10H和INT 16H。那么系统在运行过程中, 每调用这两个中断, CC-DOS使其指向自己所设计的子程序的入口。在字符处理方面, CC-DOS把汉字作为和西文字符一样的处理, 有些高级语言不需汉化即可以在CC-DOS下运行, 如BASIC语言, 可以直接用 PRINT “中同-西安”的语句将汉字显示在屏幕。但有些语言由于有自己的适用环境, 未经汉化便不能直接在CC-DOS下运行, 而且有些汉化的软件失掉原有软件的大部分功能, 稳定性也不是那么好, 所以, 需要在DOS下设计一些能在CC-DOS下应用的软件, 这就是本书所要讨论的内容。如C语言, 它处理汉字有自己独特的方式, 且它是一种编译性很强的程序语言, 它兼有许多高级语言的优点, 以自己所具有的模块化结构风格和强大的编程能力及可移植性好等特点为广大软件人员所喜爱。所以, 用C语言处理汉字对于设计汉字应用软件已成为一种趋势, 越来越被广大的语言爱好者所接受。

CC-DOS的显示控制模块是对ROM-BIOS的显示控制模块作了修改与扩充之后而形成的能显示中西文的模块。显示汉字时, 显示器应工作在图形模式下, 以点阵的形式将汉字字模显示在屏幕相应的位置上。

在这里, 首先对汉字字模及汉字区位码与汉字内码作一些说明:

按照GB2312-80《信息交换用汉字编码字符集-基本集》及GB5007-85《信息交换用汉字24×24点阵字模集》所规定的标准, 汉字是由一系列点阵构成的, 有16×16与24×24两种基本的点阵形式。当然, 有的扩充字库还有48×48以及更多的点阵形式, 这对于提高字形质量有一定的好处。在造字上, 汉字图形字符的点阵字模是由置于栅格内的若干点确定的, 栅格由若干条等距离垂直线与水平线交叉而成。如果采用16×16点阵字模标准, 栅格为横向16格, 竖向16格, 方格的中心