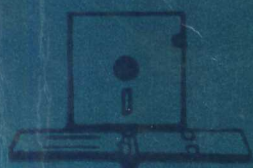


DOS

中文程序开发技巧集锦

天方图书创作室 陶 伟 刘 洋 编著

DOS中文程序
开发技巧集锦



北京航空航天大学出版社

DOS 中文程序 开发技巧集锦

天方图书创作室 陶 伟 刘 洋 编著

北京航空航天大学出版社

(京)新登字 166 号

内 容 简 介

本书针对中文应用软件在发展中出现的问题,向程序员介绍应采用的方法及技术背景。内容有:中文操作系统原理及中西文兼容技术;汉字字模与显示;语言接口(dBase、FoxBase 与 C; TSR 程序的驻留、激活、卸载;汇编语言与 C);EMS 和 XMS 的用法;使用程序自动生成器开发应用软件;程序发生技术和方法;中文 API;用面向对象方法开发中文窗口系统;DOS 操作系统常用的小技巧和小工具。书中所有程序用 C、C++、汇编、FoxBase 编写。

供青年学生、初级程序员及中级程序员阅读使用。

● 书 名: DOS 中文程序开发技巧集锦

DOS ZHONGWEN CHENGXU KAIFA JIQIAO JIJIN

● 编著者:天文图书创作室 陶 伟 刘 洋

● 责任编:许传安

● 出版者:北京航空航天大学出版社

● 发行者:新华书店总店科技发行所

● 排版者:天方科技公司

● 印装者:通县兔子店印刷厂

● 开 本:787×1092 1/16

● 印 张:21.75

● 字 数:556 千字

● 定 价:17.50 元

● 印 数:8000 册

● 书 号:ISBN 7-81012-480-3/TP·114

前言

如果您想开发一流的中文应用软件,那么本书正是您所需要的。

程序员是现代最令入羡慕的职业,成为一名优秀的程序员需要经历艰苦的磨炼。一位优秀的程序员应该具备以下的素质:他不仅充分了解某种应用的背景知识,而且完全熟悉整个计算机环境,包括操作系统和硬件本身,同时能充分理解用户的需求。只有完全掌握这些东西的程序员才能编写出漂亮的用户界面,才可以向用户提供最大的灵活性,充分发挥计算机的效用。

美国计算机界把计算机和半导体技术能做的事称为计算机的“能力(Power)”,而把程序员的想象力和软件工程能让计算机做到的事称为计算机的“效用(Utility)”。目前,计算机界的情况是计算机的能力远远超过其效用。换言之,相对于硬件的高速发展,软件尚待开发的地方还很多,如何有效地使用计算机比怎样制造计算机更为复杂。所以,去年美国《哈佛商业周刊》发表一篇引起轰动的文章《不生产计算机的计算机公司》,文中作者指出:“凡是能真正了解硬件和软件之间关系的计算机公司将是最能获利和最易立足的公司。”显然,Microsoft 公司就是一个名副其实的“不生产计算机的计算机公司”。发挥计算机的“效用(Utility)”是每一位程序员最重要的任务,本书的主线索就是硬件(Hardware)、软件(Software)、应用(Application)之间的相互关系,把效用(Utility)放在了首位。

本书并不想成为一本软件或硬件的技术参考大全,而仅仅是针对中文应用软件在开发过程中出现的具体问题,向读者介绍一位专业程序员应采用的方法,以及他必须拥有的技术背景。

本书涉及的理论和技术领域包括以下几个方面:

- 中文操作系统原理(包括支持直接写屏和不支持直接写屏的中文操作系统)
- 中文操作系统中西文兼容技术
- 汉字的字模与显示
- dBase、FoxBase 与 C 语言接口
- TSR 程序的驻留、激活、卸载,及与 C 语言接口
- 汇编语言与 C 语言接口
- EMS 和 XMS 的用法
- 如何使用程序自动生成器开发应用软件

- 程序发声的技术和方法
- 中文 API(Application Program Interface)
- 如何用面向对象的方法开发中文窗口系统
- DOS 操作系统常用的小技巧和小工具

本书所有的程序用 C、C++、汇编、FoxBase 编写。我们认为学习本书的最好的方法是通过实践来学习。最好您能拥有一台 286 以上带 VGA 卡的 PC 机,这样您可以通过实践来真正掌握书中谈到的一些技术,理解书中倡导的一些思想。正如《The UNIX Programming Environment》一书的作者 Brian W. Kernighan 所说:“Read a bit, try it out, then come back and read some more”。不要害怕摸机器它不会咬你,最好的学习方式就是大胆地尝试。

最后向在本书编著过程中,给予我们许多帮助的北京天方科技公司、OK 软件实验室及柏彦软件实验室的全体程序员表示感谢。在编著过程中我们也参考了大量的资料,在此也向这些资料的作者表示感谢。

陶 伟 刘 洋

1993 年 10 月于北京 OK Soft Lab.

目 录

第一章 中文操作系统概述

1.1 中文操作系统的发展史	1
1.2 中文操作系统的组成	2
1.3 中文操作系统的原理	4
1.3.1 中文系统的显示原理	4
1.3.2 中文系统的输入接口	11
1.3.3 中文系统的打印接口	11
1.3.4 中文系统的系统服务	12
1.4 中文系统的性能评价	12
1.5 中文系统的发展趋势	13

第二章 汉字的字模与显示

2.1 汉字的代码体系	14
2.1.1 汉字的输入码	14
2.1.2 汉字机内码	14
2.1.3 汉字地址码	15
2.1.4 汉字交换码	16
2.1.5 汉字字形码	16
2.2 字库与字模	16
2.3 汉字字模的读取	18
2.4 建立自己的中文字库	25
2.5 汉字的放大显示	33

第三章 如何用 C 语言对 .DBF 文件进行处理

3.1 .DBF 文件的物理结构	40
3.2 在 DOS 下显示 .DBF 文件的库结构和内容	42
3.3 用 C 实现 .DBF 文件的通用报表生成	44

第四章 用 FoxView 开发中文数据库软件

4.1 建立一个中文数据库	51
4.2 FoxView 格式视图和表格视图	53
4.3 用 FoxView 产生程序和运行	54
4.3.1 产生程序	54
4.3.2 运行程序	56
4.3.3 数据库文件的使用、映射与进入	57
4.3.4 FoxView 中的数据项	60
4.4 FoxView 中各菜单选项的用法	62

4.4.1	视图	62
4.4.2	Menu System	62
4.4.3	磁盘菜单	64
4.4.4	装入菜单	65
4.4.5	Gen Menu	68
4.4.6	字段菜单	72
4.5	视图	73
4.5.1	方框、颜色与文本	73
4.5.2	屏幕格式视图	77
4.6	一个通用的数据库管理例程	84
4.7	通用报表例程	102

第五章 如何在 dBase 和 FoxBase 程序中调用 Turbo C 语言图形库

第六章 如何为应用程序配上音响效果

6.1	PC 发音的原理	121
6.2	警笛和“激光爆炸”	122
6.3	产生太空音响	124
6.4	在后台演奏音乐	125

第七章 DOS 内存驻留技术(TSR)

7.1	DOS 中断向量	128
7.1.1	计算机的中断结构	128
7.1.2	IBM PC 提供的中断	129
7.1.3	键盘输入的方法	131
7.1.4	改变中断向量	131
7.1.5	检查中断向量	133
7.1.6	Vector.asm 显示中断向量源程序	134
7.2	最简单的 TSR 程序	142
7.2.1	一个最基本的 COM 程序	143
7.2.2	超小型的 TSR 的程序	144
7.2.3	改进的 TSR 程序	145
7.2.4	减少 TSR 程序的内存额外负担	146
7.2.5	TSR 程序的激活机制	146
7.2.6	把 TSR 程序与中断处理程序连接	148
7.2.7	一个完整的实用的 TSR 程序	150
7.3	TSR 程序应用之一:键的宏定义程序	152
7.3.1	基本的键宏程序	154
7.3.2	多键宏程序	157

7.3.3	单键宏源程序 Macos.asm	159
7.3.4	多键宏源程序 Macom.asm	162
7.4	检查 TSR 程序是否已驻留	166
7.4.1	通过扩展的功能号作为判断 TSR 驻留的标志	166
7.4.2	通过 MCB 链来寻找 TSR 程序驻留标志	167
7.4.3	使用多路中断 INT 2FH 来判断驻留	170
7.5	如何卸载 TSR 程序	174
7.5.1	COM 程序和 EXE 程序	175
7.5.2	DOS 内存管理	178
7.5.3	程序段前缀(PSP)	178
7.5.4	DOS 功能调用 31H 终止进程并保持驻留	183
7.5.5	DOS 中断 27H 结束但驻留内存	184
7.6	TSR 程序与 C 语言接口	188
7.7	TSR 程序应用之二——应用程序的自动演示工具	198

第八章 EMS 和 XMS 的使用方法

8.1	DOS 的内存管理	206
8.2	DOS 下内存管理的几个发展阶段	207
8.3	扩展内存(Extended Memory)	208
8.4	XMS 内存管理规范	209
8.5	EMS 内存管理规范	211
8.6	EMS 应用实例	212
8.6.1	检测 EMS 是否存在	212
8.6.2	EMS 编程的几点注意事项	214
8.6.3	使用 EMS 装载汉字库	214
8.7	XMS 应用实例	220
8.7.1	检查 XMS 是否安装	220
8.7.2	XMS 编程的注意事项	221
8.7.3	使用 XMS 装载汉字库	221

第九章 中文 API

9.1	数据结构说明	228
9.1.1	宏定义	228
9.1.2	数据结构	228
9.2	库函数使用说明	230
9.2.1	HZValue()宏	230
9.2.2	InitSYSDisp()函数	230
9.2.3	ExitSYSDisp()函数	231
9.2.4	InitSpecDot()函数	231

9.2.5	ExitSpecDot()函数	232
9.2.6	InitInput()函数	232
9.2.7	Register AsciiFont()函数	232
9.2.8	RemoveAsciiFont()函数	233
9.2.9	SetAsciiFont()函数	234
9.2.10	SetHZFont()函数	234
9.2.11	SetScreenAttr()函数	235
9.2.12	SetTextAttr()函数	235
9.2.13	SetBackAttr()函数	235
9.2.14	SetShowBack()函数	235
9.2.15	SetScreenShow()函数	236
9.2.16	GetScreenAttr()函数	236
9.2.17	GetScrAsciiFont()函数	236
9.2.18	GetScrHZFont()函数	237
9.2.19	GetShowBack()函数	237
9.2.20	GetScreenShow()函数	237
9.2.21	GetWindowSize()函数	237
9.2.22	GetScreenUnit()函数	238
9.2.23	GetUnitSize()函数	238
9.2.24	PielXY()函数	239
9.2.25	XYPixel()函数	239
9.2.26	ShowUnit()函数	239
9.2.27	ShowLine()函数	240
9.2.28	ShowWindow()函数	240
9.2.29	ChangeAttr()函数	240
9.2.30	ReadWindow()函数	241
9.2.31	WriteWindow()函数	242
9.2.32	WriteChar()函数	242
9.2.33	WriteUnit()函数	243
9.2.34	WriteString()函数	243
9.2.35	Clear Window()函数	244
9.2.36	ScrollUpWindow()函数	244
9.2.37	ScrollDownWindow()函数	245
9.2.38	SetScreenXY()函数	245
9.2.39	GetScreenX()函数	246
9.2.40	GetScreenY()函数	246
9.2.41	OpenCursor()函数	246
9.2.42	CloseCursor()函数	246
9.2.43	GetCursorStyle()函数	247

9.2.44	SetCurosrStyle()函数	247
9.2.45	GetSpecDotSize()函数	247
9.2.46	GetSpecDot()函数	248
9.2.47	DispSpecDot()函数	248
9.2.48	OpenMouse()函数	249
9.2.49	CloseMouse()函数	249
9.2.50	MoveMouse()函数	249
9.2.51	GetMouseStatue()函数	250
9.2.52	AbsReadKey()函数	250
9.2.53	PressKey()函数	250
9.2.54	ClearKey()函数	251
9.2.55	ReadKeyMouse()函数	251
9.2.56	打印字库点阵提取程序	252

第十章 用面向对象的方法开发中文窗口系统

10.1	程序设计语言的演变	252
10.2	面向对象程序设计语言中的术语	254
10.2.1	对象	254
10.2.2	类、对象、实例变量、方法	254
10.2.3	封装	254
10.2.4	继承	255
10.3	面向对象的窗口系统	255
10.4	面向对象的中文窗口系统源程序	257

第十一章 DOS 操作系统的使用技巧与小工具

11.1	286/386 中的 Shadow RAM 功能应用	292
11.2	巧用 DOS 设备名	293
11.3	DOS 三种设备文件名的使用技巧	293
11.4	如何保存和恢复硬盘正常主引导记录	294
11.5	巧用 CHKDSK 列出所有隐藏子目录	296
11.6	Whereis 磁盘文件查找工具文件	298
11.7	GREP——功能强大的匹配查找工具	299
11.7.1	所有的开关及表示的含义	299
11.7.2	如何使用 GREP 查找串	300
11.7.3	使用 GREP 的例子	300
11.8	DOSKEY——DOS 命令行记忆编辑工具	302
11.9	Touch——更新文件的生成时间和日期	303
11.10	WC——统计文件的行数和单词数	303
11.11	Timer——实时时钟	303

11.12	DT ——目录服务程序	303
11.13	Unti ——引导区病毒清除程序	304
11.14	KEY ——键盘监测程序	304
11.15	MFONT ——中西文字模编辑器	305
11.15.1	性能概述	305
11.15.2	启动	305
11.15.3	菜单操作	305
11.15.4	功能键及鼠标操作	306
11.16	Speedkey ——键盘加速程序	307
11.17	TDMEM ——查看内层空间,检查驻留程序	308
11.18	LHA ——功能强大的压缩工具	308
附录一 技术词汇表		310
附录二 EMS 功能调用参考		313
附录三 XMS 功能调用参考		324
附录四 《启明星中文操作系统简版》使用说明		329

第一章

中文操作系统概述

近十年来,随着国内微型计算机的普及,它的应用领域也在逐步地拓宽。DOS 下的中文操作系统也经历了从无到有,从单一类型到多种类型并存的过程,并使之成为微机设备中不可缺少的一部分。而如何更好地了解和使用的中文系统,早已成为广大用户所关注的主要问题。本章将着重介绍 DOS 下中文操作系统的发展历史、工作原理,以及性能的评估,使读者能更好地驾驭所使用的中文操作系统,使其发挥更大的效益。

1.1 中文操作系统的发展历史

1984 年,原电子工业部推出 CC-DOS2.1,适用于 IBM-PC 系列机。毫无疑问,CC-DOS2.1 的推出,才真正使中文信息在微型机上处理成为可能。可以说,CC-DOS2.1 是我国第一个实用的中文操作系统,虽然它只是针对 DOS 在 IBM 系列的微型机上实现,但其意义在于证明了中文信息同样可以用计算机来进行处理,为我国中文信息处理作了巨大的贡献。

CC-DOS 以 2.1 版本为起点,陆续改进升级,直到 CC-DOS4.0 的推出,中文操作系统可以说进入了第二个阶段。但是,CC-DOS2.1 仅达到了一个可行的目标,高效和可靠的目标远未达到。在三个 CC-DOS2.1 的基本模块中,都表现出效率和可靠性方面的弱点。例如,在键盘输入方面不够方便和快速,在显示输出上适用面不广,在打印输出上字体字形不丰富等等的弱点。另外在一些系统资源的利用和管理上不尽合理。到 CC-DOS4.0 时,对上述弱点作了或多或少的改进。CC-DOS4.0 初步向上们证明了,中文信息不仅可以在计算机上进行处理,而且可以比较高效地在计算机上处理,甚至于可以更有利于西文字符信息的处理。

以 CC-DOS 的各种版本为基础,中文操作系统的开发者们在不同的角度和侧面进行了开发,并且也得到了广泛的应用。例如,2.13 系列的操作系统对 CC-DOS 的打印输出功能作了极大的改进,受到用户的青睐。然后,GWBIOS、UCDOS、联想系统 etc 以其丰富的字符功能、较强的汉字输入能力、彩色输出功能等吸引了一批用户。接着,金山 DOS 以其简洁有效的桌面系统获得用户信赖。

1990 年之后,大陆的软件工作者才在吸取台湾倚天中文系统的基础之上又推出了一系

列的支持文本直接与屏的中文系统,其中较为著名的中文软件系统有:天汇标准中文系统、超想中文系统、中国龙、PTDOS、启明星等等。而汉卡类的硬中文系统数不胜数。这些中文系统固然在速度上、软件收益上有所保证,但由于价格高、升级慢、中西文兼容差等原因,很难大规模推广。我们认为硬中文系统是没有前途的,故本书不愿在这个方面花更多的笔墨。

总之,中文 DOS 经过 10 多年的发展,涌现出许多优秀的中文 DOS 系统。它们为微机在国民经济的各个领域的普及起到了关键性的作用,同时大大促进了中文电脑的发展。

但是,纵观市场上的中文系统,也存在着一些问题:(1)没有统一的标准和规范;(2)应用软件兼容性差;(3)不是彻底的中文 DOS 系统,目前汉字 DOS 大多用“外挂”的办法;(4)重复开发,各自为政。据一些人士统计,当前市场上出售的汉卡不下几百种,软中文系统也有几十个版本。这些问题,实际已经成为中文电脑进一步发展的障碍,因此在 1993 年 9 月 28 日,大陆成立了“中文平台技术委员会”,其 DOS 专委会提出了中文 DOS 标准应该具备的 4 个特征。

- 制定统一的 API 规范,确保应用软件能在不同厂商的汉字环境下运行。
- 设定输入法、提示行、字形、打印命令等模块接口,允许用户或软件开发者挂新的模块。
- 设定统一操作系统接口,保证汉字开发商在显示、键盘、打印、监控等模块的接口一致。
- 该标准应能容纳台湾有关中文 DOS 的接口规范,以利海峡两岸的文化交流。

可以预计,中文操作系统的春秋战国时期不会持续很久,它将向着标准化、通用化的方向发展,最终形成几个主流版本。

1.2 中文操作系统的组成

在介绍中文系统的基本组成之前,首先介绍几个有关的基本概念。

中文系统 DOS 下中文系统是指附加在微机西文 DOS 环境之下的一层汉字“外壳”。这层“外壳”由纯软件或软件、硬件组成。通过修改 ROM BIOS 中有关输入、输出的中断来“截获”西文环境下的输入、输出,并加入汉字的输入、输出功能,使之可以实现汉字的输入、显示及打印。这样就把原来只能处理西文的 DOS 操作系统转化为可以处理汉字的操作系统,如图 1.1 所示。

汉字交换码、外码和内码 汉字的交换码是指作为信息交换用的汉字编码。外码是指汉字的输入码,由于在汉字输入时还是使用西文的键盘,因此,每输入一个汉字需要若干个键值,这组键值就称为对应汉字外码,例如,在拼音输入法中“汉”对应的外码为“HAN”。内码是指机器内部的汉字编码,在国内现在一般所使用的汉字内码为双字节码,它是汉字交换码 GB2312-80 的变型码,即每个字节的最高位为“1”,作为汉字的标识。例如,汉字“啊”的汉字内码为“B0A1”。

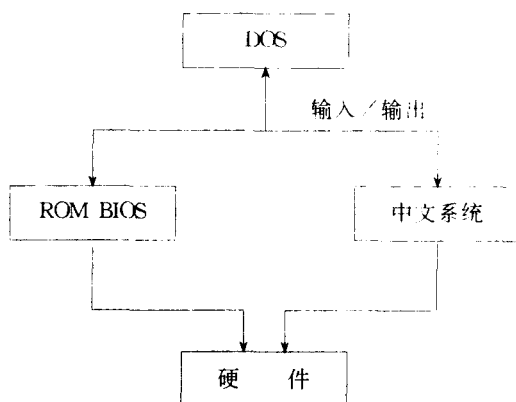


图 1.1 中文 DOS 示意图

中文系统的基本组成主要分以下 5 个方面。

1. 键盘管理模块

键盘管理模块是负责管理计算机键盘的功能模块，它的主要功能有两项：一是对外部输入法的接口，外部输入法将应用程序输入的汉字外码转换为计算机的内码，通过这个接口传送给系统，另一功能是解释系统的功能键并调用相应的系统服务功能。

2. 显示管理模块

显示管理模块负责解释所要显示的汉字内码或 ASCII 字符，并将它们显示在计算机屏幕上，是中文系统的主要模块，它代表着这个中文系统的实现类型和技术水平。

3. 字模管理模块

字模管理模块负责提供屏幕显示和打印所需的汉字点阵。它根据要求对汉字字模的原始信息进行加载、处理，最终提供相应的输出点阵。例如：如果需要提供 40×40 点阵的空心字，字模管理块就会从字库中获取该字的信息，生成相应的点阵，并对它进行空心处理而后返回。早期的汉字系统这种模块比较简单，随着用户对输出质量逐渐提高，汉字的矢量字库便应运而生了，这就使得字模管理模块的功能更加趋于复杂。

4. 打印管理模块

打印管理模块是支持汉字打印的打印驱动模块。它接收需要打印的汉字内码和打印属性，再通过字模管理模块提取相应的点阵，最后通过打印机的图形方式控制打印机，使汉字从打印机上输出。

5. 系统服务模块

系统服务模块是中文系统的服务性支持模块。它为中文系统和在此系统之上开发的应用程序提供相应的低层支持。包括各种实用程序，诸如造字、系统配制管理，以及系统的内核调用，以中断或高级语言库方式提供。如果在中文系统下开发的应用程序中充分利用系统服务模块，那么将会得到事半功倍的效果，即所谓的 API。但是需要提醒注意的是，它可能会影响应用程序在其他汉字系统上的工作。

各模块关系示意如图 1.2 所示。

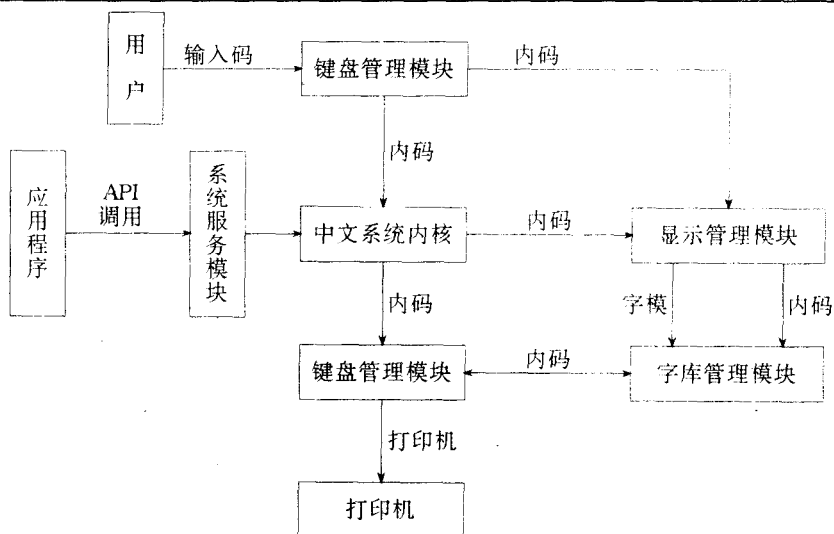


图 1.2 中文 DOS 组成

1.3 中文操作系统的原理

IBM PC 机的 BIOS 存放在主机板的 ROM 中,所以一般称其为 ROM BIOS。它由若干独立的外部设备驱动模块组成,向系统程序员和高层应用程序提供对外部设备的控制,包括加电自检、显示器、键盘、通讯、打印机和磁带机等的信息传送等,这些操作均无需用户考虑其细节。ROM BIOS 是高层软件与外部设备间的桥梁。

组成 ROM BIOS 的各模块的入口地址存放在内存最低端的中断向量表中,并以直接调用或者间接调用的方式向高层软件提供对外部设备的控制,其中断号为 10H~1AH。CPU 接收到中断后,就根据中断向量转向 ROM BIOS 中相应的服务程序执行。

因此,汉字系统应扩充或修改 ROM BIOS 中与汉字输入输出相关的模块,它们是:显示器管理模块(10H 号中断)、键盘管理模块(16H 号中断)、打印驱动模块(17H 号中断)以及屏幕拷贝模块(5H 号中断)。在系统启动时将它们调入内存并常驻,然后修改中断向量表,使相应中断向量,指向内存中的新的模块。以后当相应的 BIOS 中断调用发生时,就不再执行原 ROM BIOS 中的模块,而转向内存中的相应模块。这就是汉字系统实现的基本方法。

下面,我们从中文系统显示、输入接口、打印、系统服务等四个方面详细加以论述。

1.3.1 中文系统的显示原理

一、西文的显示原理

西文显示环境将显示模式分为两类:一类是西文的文本状态,另一类是西文的图形显示状态。

在标准西文的文本显示模式中,显示屏幕被分成 25 个文本行,每行 80 个文本列,屏幕上能显示 2000 个 ASCII 字符,每个字符由显示存储器中的两个字节来决定。首字节被映射在偶数内存地址,包含所要显示的 ASCII 字符码,而第二字节被映射为一个奇数内存地址,

包含被称为字符属性的彩色信息。为了定义 25 行 80 列的页需要 4000 个字节的显示内存。

当显示环境为单色显示卡时,文本显示存储区开始于 B000:0000。而彩色显示卡的文本显示存储区开始于 B800:0000。

为了将一个 ASCII 字符转换成屏幕上的一个象素阵列,就要使用到显示卡上的发生器,通过字符发生器产生点阵信息,再将点阵信息连同字符发生信息一起送入视频控制器,由视频控制器产生输出信号给显示器,这样就可以在屏幕上看到所要显示的 ASCII 字符了。

西文的图形显示(以 VGA 卡的 12H 方式为例)在 VGA 的 12H 方式下,将屏幕在竖直方向上分为 480 个象素,水平方向上分为 640 个象素。屏幕上的每个点都可以选择 16 种颜色值(0~15),用 4 位表示,它们被按顺序分别记录在显示缓冲区中相应的 4 个位平面上,通过显示控制线路将显示缓冲区中的信息送到显示器进行显示。

二、西文文本显示的基本方法

在西文 DOS 环境下,显示西文文本一般有三种不同级别的显示方法,即高级显示方法、中级显示方法和低级显示方法。一种方法是通过 ROM BIOS 中的显示方面的功能调用来显示西文 ASCII 字符,这是早期软件所使用的方法,由于它是通过 ROM BIOS 完成显示工作,所以显示速度受到影响但软件兼容性好。另一种方法是通过了解显示缓冲区的位置后,绕过 ROM BIOS 直接将要显示的 ASCII 字符和属性送入显示缓冲区的相应位置,从而实现文本的高速显示。这种显示方法称为“直接写屏”,它是现在常用的显示方法。最简单最高级的方法是通过 DOS 调用,它完全与显示硬件独立,可保证在所有运行 DOS 的机器上运行。下面将对这三种级别的显示和中文系统的关系加以具体论述。

1. 高级显示方法(DOS 系统调用)

在 DOS 的系统调用里,有几个可用于显示字符,并与显示硬件完全独立,可保证在所有运行 DOS 的机器上运行。

这里的系统调用有 2H 号(显示单个字符)、9H 号(显示字符串),由于 DOS 已将文件分配指针 1 给了标准输出设备(显示器),所以也可用 40H 号调用来显示字符。

对这些采用高级显示输出的软件的汉化,已不需要修改原来的程序本身,就可完成汉化工作。这是因为汉字系统已在比 DOS 更底层的 BIOS 级上解决了显示汉字的问题,用户只需直接修改原来的西文字符串为汉字即可显示出汉字信息。但有时西文软件还综合了其他方法显示输出,下面详细说明。

对西文显示信息的汉化,已有一些现成的汉化工具软件,其原理是将软件中的原西文字符串提取出来,将字符串及位置存入一中间文件,用户使用各种字处理软件将其中的西文改成汉字,再将其按原位置写回软件即可。

2. 中级显示方法(ROM BIOS 调用)

中级显示方法是指调用 IBM 的 BIOS 显示器管理模块实现的显示功能。通过 BIOS 调用还可实现设置显示方式、选择显示页、读显示状态、置调色板、设置字符前景与背景颜色以及画点读点等功能,显示速度比高级显示方法快。可使显示输出更加灵活多样、丰富多彩。

在汉字系统下,BIOS 功能调用的 9H 号、10H 号和 14H 号已能够完成汉字的正常显示。然而在许多软件里,往往和其他调用一起来完成字符显示,从而影响到正常的汉字显示:

(1)设置显示方式

这在西文软件里是常见的,就是利用 BIOS 中断 INT10 的 0 号功能调用,将显示方式设置为 3(80×25 彩色字符方式)或 7(单色字符方式),有的利用此功能清屏幕。然而多数汉字系统是工作在图形方式下的,这里应改为其所需的显示方式

(2)对输出字符进行预处理

有的西文软件在显示输出字符时,先对它进行预处理,对 ASCII 码小于 20 号和大于 7F 号的字符进行转换或屏蔽高位。如动态调试程序 DEBUG.COM 就是将欲显示字符的最高位进行屏蔽再显示的。但是这最高位是汉字内码区别于西文字符的标志,所以将这个处理指令去掉即可显示汉字了。

在一些西文软件中,字节的最高位被用来作奇偶校验或加亮标志,若对最高位不作处理,将会失去这些功能,如文字编辑软件 WordStar。

3. 低级显示方法(直接写屏)

在相当多的西文软件里,是通过直接写显示控制器的寄存器和显示缓冲区(或称显示器刷新存储区)的,以获得最佳显示效果,即所谓的“直接写屏”技术。已经计算好并置于 DI 寄存器中,显示缓冲区段地址置于 ES 中,所要显示的字符及属性置于 AX 中,只用了一条 STOSW 指令就完成了字符的显示。

这种采用低级显示方法的软件的汉化,比前两种方法的汉化更为复杂,尤其在使用图形方式的汉字系统中的汉化。因为这里只有图形显示缓冲区,而且不同的显示方式显示缓冲区段也有所不同,只有将显示方法改为中级显示方法才行。然而这不仅需要对原来的程序进行大规模的修改,而且很可能失去原有的一些功能。所以这类软件常常在字符方式显示汉字的机器上首先进行汉化,如长城机和联想汉卡等。即使这样,原程序也要作很大修改,因为这些机器原显示缓冲区里字符区和属性区往往和西文显示缓冲区的结构不同。

为了克服上述汉化过程中的缺陷,可采用一种类似于中级显示的方法,即用软中断实现写汉字显示缓冲区的技术来完成汉化工作。这样不但可保留原软件的全部功能,而且能加快汉化速度。另外,对显示缓冲区结构不同的软件汉化,只要使用不同的中断服务程序即可。

在汉化时,应选用合适的中断号,避免与 DOS 系统、汉字系统和应用程序中所使用的中断号发生冲突。在汉化过程中,只用一条中断指令来替换原程序中直接写显示缓冲区的指令,就可达到显示汉字的目的。

例:foxBase 中级显示字符程序段

```
;AL 为待显示字符,ES 已设置为 0B800H
MOV AH, CL           ;显示属性
MOV ES:[BX], AX      ;写显示缓冲区
ADD BX, 2             ;指针加 2
```

采用此法,设置一中断 INT6AH,再将指令 MOV ES:[BX],AX(三个字节)改为两条指令 INT6AH 和 NOP,就完成了对 foxBase 的汉化。下一项任务就是编写中断服务程序,用以将主程序里的寄存器 AX 中的字符及属性在指定位置上显示出来。

由上面汉化过程可见,只要找出原程序中显示模块就可立即进行汉化工作,无需再去分析整个程序去寻找可供插入修改代码的位置。

使用低级显示方法的软件,其清屏方式往往也是采用直接写显示缓冲区的方法进行,如 REP STOSW 指令。另外,在使用弹出式或下拉式菜单时,常常用保存和恢复显示缓冲区内