

电脑

(月刊)

1994年第3期

总第69期

主 办:电子工业部中国软件行业协会
编 辑:《电脑》编辑部
出 版:电脑杂志社
地 址:广州市石牌华南师范大学内
邮政编码:510631
电 话:(020)5516911-3273
总发行处:韶关市邮电局
国外发行:中国国际图书贸易总公司
(北京 399 信箱 邮政编码:100044)
国外发行代号:M4190
印 刷:韶关曲江印刷厂
定 阅 处:全国各地邮电局、所
定 价:3.00 元
出版日期:1994年2月25日
刊 号:ISSN1002-9613
CN44-1188TP
邮发代号:46-115
广告经营许可证:粤工商广字 01090 号
主 编:吴 军
副 主 编:林 林
责任编辑:萨成照

奖

有奖征名

为了把《电脑》办得更生动活泼,内容既充实又具有现代气息。《电脑》今年开展栏目有奖征名活动,对现有的栏目,你有更贴切更醒目的名称取代,请来信。一经采用,奖励50元/个。并在杂志上公布。

多媒体

2/ 多媒体声霸卡软件的主要功能特点

先睹为快

6/ CPU+RAM+FBGA

电脑与法律

7/ 用户界面的版权保护讨论

电脑应用

10/ 电子数据交换(EDI)面面观(四)、(五)

13/ EDI在船舶代理行业中的应用与发展

软件纵横

15/ DOS 4B 01 功能分析及应用

网络与通信

19/ NOVELL 网络设计与应用

22/ 高级 UNIX 连网技术讲座 第五讲 TLI(上)

使用与维修

25/ 彩色显示器场不同步及无光栅故障的检修

26/ LQ-1600K 打印机检修一例

NEW

27/ 新一代的远程访问产品——NetWare Connect

IDEA

28/ FOXBASE 与 COBOL 语言相互读取数据的一种方法

30/ 为菜单的每一选项提供相应的 HELP

31/ 利用反汇编模块实现转换键的动态模拟

中文信息处理

32/ 自然码——一种优秀的汉字输入法

ABC

33/ 用 C 语言实现的一个洗牌发牌程序

34/ CPDOS 使用经验点滴

35/ MS-DOS6.0 下多种系统软件的应用

36/ 用 NORTON 对经过加密的深层子目录解密

37/ PROMPT 命令的特殊用途

游戏乐园

39/ 《笑傲江湖》攻略

41/ 魔法门外传存盘数据文件分析

42/ 游戏《组织》的心得与解拆

病毒防治

43/ 硬盘写保护防病毒新法

44/ 神秘的“V888”病毒

48/ 一种通用的消毒方法和一个奇妙的
解毒工具 RCOPY03

单片机与单板机

51/ 单片微机与模糊控制讲座

第九讲 用 MC6805R2 控制的电饭锅(上)

电脑英豪录

53/ “2.13”——吴晓军

软件廊

54/ 改进汉字输入的探索

56/ 五笔字型自学速成软件

57/ 中外软件廊部分软件报价

摩卡

59/ 综合驱动电子器件卡

电脑用户

60/ 汉化 README.COM 程序开发与功能扩充(中)

64/ XENIX 下磁带备份和恢复的实用程序

66/ 西文 DOS 下汉字高速显示

69/ 利用 XMS 功能使金山 DOS 与 DOS5.0 兼容

71/ 提取并修改或汉化程序中
英文文本信息的工具软件

73/ 再谈单显作图问题

74/ 几种图象文件格式浅析

75/ 搜索回溯原理及它的递归实现

简讯

77/ DEC 公司与广东“京粤”
携手合作发展电脑产业

小辞典

77/ 小辞典

CONTENTS

2/ Characters of the "Sound Blater" multimedia card

6/ CPU+RAM+FBGA

7/ The copyright of user interface

8/ What is EDI?

15/ The analysis of DOS 4B 01

19/ Applications of NOVELL

22/ Advanced UNIX networking techniques (5) TLI

25/ Repair for field asynchronism and without grating in
a color monitor

26/ Repair for LQ-1600K

28/ An access method between FOXBASE and COBOL

30/ HELP for every selecting in a menu

32/ ZRM — a fine Chinese character input code

33/ A program for shuffling and sending playing cards
in C

34/ Experience of using SPDOS

35/ Applications of system software based upon
MS-DOS

36/ Diciphering for an enciphered deep subdirectory
with NORTON

37/ Special uses of the command PROMPT

39/ Games

43/ Virus precaution for hard disks

44/ Virus "V888"

48/ A method of elimination of viruses and its tool
RCOPY03

51/ Single chip computer and fuzze control (9)
controlling by MC6805R2

53/ Mr. Wu and his "2.13"

58/ README.COM in Chinese characters and its
extened function

62/ The magnetic tape backup and restore utility file
under XENIX

71/ Graphing in a monochrome monitor

72/ analysis of the formats of image files

73/ Search backtraeking and its recursive process

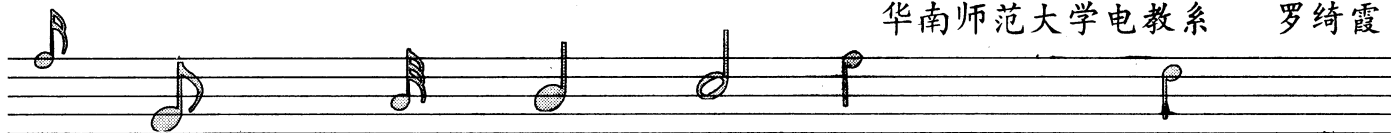
广告索引

1. 广东省罗定市无线电厂
2. 华楠建筑新材料公司
3. 都帮机电有限公司
4. 广州市科立特技术发展公司
5. 广东省天地广告有限公司
6. 香港现代电子出版社
7. 广州国际电脑电子地博览中心中外软件廊
8. 华粤电脑工程公司
9. 广州市海谊电子仪器实业公司
10. 广州宏辉机电工程公司
11. 广州袖珍计算机技术服务中心
12. 广州白云山电源设备厂
13. 电脑软件法律保护咨询部
14. 南方网络计算机有限公司
15. 宏图电脑
16. 广州蓝深计算机网络系统公司
17. 中美合资广州安斯电子设备厂
18. 广州中联电脑公司
19. 清华大学科学馆
20. 广东省佛山通用电器厂



多媒体声霸卡软件的主要功能特点

华南师范大学电教系 罗绮霞



当今计算机领域的热点技术和市场走向是多媒体计算机技术。随着多媒体技术的发展,出现愈来愈多的多媒体硬、软件产品。其中新加坡创新公司的声霸卡系列产品是目前国际上较为流行的多媒体电脑的标准音响设备。它不但具有优良稳定的硬件特性,还具有很强、很丰富的软件功能。因此,我们有必要深入了解和掌握声霸卡的软件功能特点,以便有效地应用声霸卡的声音处理技术。

一、声霸卡软件的安装

声霸卡软件安装过程如下:

1. 插入第一张声霸卡软盘到驱动器
2. 敲入 Install 命令,将出现文字说明,按任一键,出现主菜单。
3. 在主菜单选择“Begin Install”,出现提示后,选择软件安装在哪个硬盘。
4. 安装程序将在选择的硬盘建立一个子目录\SBPRO,并复制文件到该子目录。
5. 文件复制完,安装程序自动在 AUTOEXEC. BAT 文件加入设置声霸卡运行环境的参数。
6. 用户确认开机启动的驱动器。
7. 用户确认机内 Windows 3.1 系统所在的地方,以便安装程序自动在 WIN. INI 文件加入一条命令,这样,下次启动 Windows 时会自动建立声霸卡的窗口及应用程序图符。
8. 用户按照屏幕提示选择三个

硬件参数: I/O 地址 220H 或 240H; 中断号 2,3,5,7; DMA 通道号。当选择的参数与实际硬件不符时,安装程序提示你重新选择。选择好参数后,安装程序自动在 AUTOEXEC. BAT 文件设置环境变量,指明这些参数的设置情况。

9. 重新启动系统,完成安装。

安装完毕后,在指定硬盘建立了如图 1 所示的目录结构,声霸卡软件程序安装在这些子目录中。

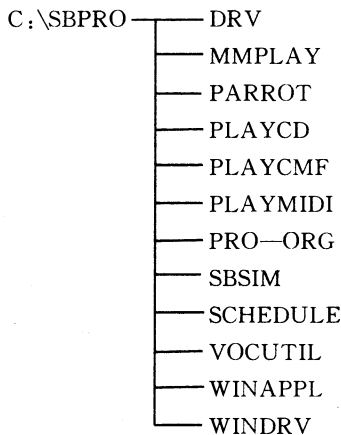


图 1 声霸卡软件目录结构

安装完毕后,为了检验一下声霸卡硬、软件安装的有效性,并对声霸卡的声音、音乐和声音效果有个初步的印象,我们可首先调用声霸卡的多媒体演示软件(进入 SBPRO\MM-PLAY 子目录,敲入 MMDEMO 和按回车)。从中可看到该演示软件集合了图像动画制作和声霸卡的声音功能(如:调频音乐、数字化声音和 CD 音频),且声音和画面同步,支持

渐显和混合等。

二、声霸卡在 DOS 环境下的应用软件的功能特点

声霸卡的一些应用软件是在 DOS 环境下运行调用的。声霸卡在 DOS 环境下运行的主要应用软件的功能特点如下:

(一) 会说话的鹦鹉 (Talking Parrot) 软件

软件的调用过程:

1. 进入\SBPRO\PARROT 子目录。
2. 敲入 PARROT 和按 Enter。
3. 按 Enter 和输入一个数字 (140~230), 该数字与声音噪声电平有关。

这时,在屏幕上出现一个在讲话的鹦鹉图像。当鹦鹉停下来时,你对着插在声霸卡上的麦克风讲话,鹦鹉会重复你刚才所说的话,十分有趣。按 Esc 退出应用程序返回 DOS。

这个应用软件一方面表明了声霸卡的声音处理能力;另一方面,我们可将说话的鹦鹉客户化。即我们可通过建立自己的图像和声音来取代鹦鹉的图像和声音。因为鹦鹉图像由四个图像文件组成,其发出的声音由 21 个声音文件组成,我们只需创建四个 .PCX 格式的图像文件取代鹦鹉图像文件,建立 21 个 .VOC 格式的声音文件取代鹦鹉说话的 21 个声音文件并存储于声音库中,就可实现鹦鹉客户化。这种功能,可帮助我们容易地

设计出有声音、有图像的简单应用软件。

(二)调频智能风琴软件(PRO—ORG)

1. 功能

帮助用户用 PC 键盘演奏出象管弦乐队效果的音乐;帮助用户掌握风琴的使用;如果有附加的 MIDI 键盘,还可以自动加入背景伴奏和节奏。

2. 调用过程

①进入 SBPRO\PRO—ORG 子目录。

②敲入 PRO—ORG 和按 Enter,出现风琴键盘画面和主菜单。

③选歌曲:按 F6,出现可演奏的乐曲名单,通过←、→、↑、↓等键选择,选中按 Enter。

④选择乐器和节奏:按 F2,出现 CMS'S 的智能风琴控制板且在演奏。这时,选择 F1~F4 改变演奏乐器,F5~F9 改变演奏节拍。

⑤按 F2 以伴奏形式演奏,按 F4 则可以听到乐曲的旋律。

(三)声音编辑 II (Voice Editor II) 软件

1. 功能

该声音编辑软件支持立体声录音、压缩和无限长的声音文件的编辑;它可对声音波形进行显示、放大、粘贴和剪辑;它可使声音生成回波和混响、淡入和淡出等特殊效果;可重放声音文件;可控制 CD 音频等。

2. 进入声音编辑 II 软件

①进入 SBPRO\VOCUTIL 子目录

②敲入 VEdit2[/B:nn]和按 Enter

其中,命令行参数/B:nn 设置缓冲区大小,nn 默认值为 16,表示 32K 的缓冲区。声音编辑软件有 7 个菜单命令:File、Record、Play、Pack(压缩)、Edit(编辑)、CDisc(CD 盘)和 Volume(音量)。File 菜单中的命令主要对声音文件进行装载、存盘等操作;Record 菜单完成录音参数的设置和声音文件的录制;PLAY 菜单命令

实现对声音文件的播放;PACK 菜单命令完成对声音文件的压缩;EDIT 菜单命令完成声音数据块的插入、删除、修改、复制、连接等操作;CDisc 单播放 CD 光盘;Volume 菜单调节各种音源的音量。

3. 录制一个声音文件(格式为.VOC)

(1)选择 Record 菜单中的 Settings 命令设置录音参数。

(2)设置声音存放的地方。在 Record 菜单中选择 To Memory 则存于内存;To Disk 则以.VOC 文件格式存于磁盘。

4. 声霸卡支持的.VOC 声音文件的组成

声霸卡支持.VOC 格式的声音文件。一个.VOC 声音文件由一些信息块组成:Header(文件头)、Voice(声音块)、Silence(无声块)、ASCII Text(ASCII 文本块)、Marker(标记块)、Repeat(重复块)、End Repeat(结束重复)、Terminator(文件终止)。当我们选择 File 菜单的 Load 命令装载声音文件后,就会在屏幕出现该文件的块信息窗口。我们可方便地用鼠标在块信息窗口选择一个或多个连续信息块。在希望选择的某一信息块按一下鼠标,则可选择该信息块;在希望选择的起始信息块按住鼠标左键,然后拖动 鼠标到终止块后松手,则可选择连续的多个信息块。

5. 编辑声音文件

我们可用软件中 Edit 菜单的编辑命令对声音文件的信息数据块进行编辑,来实现声音文件的编辑。假如我们要插入信息块,首先要用 File 菜单的 Load 命令装载文件;在块信息窗口选择希望插入的地方;在 Edit 菜单选择 Onsert 命令,出现对话框,在对话框选择插入块的类型,选择 OK,则将插入一个信息块,插入的信息块显示在块信息窗口。类似地,我们可以删除、移动、复制、压缩声音文件的信息块。

6. 改变声音波形特性和特殊的

效果处理

使用 Edit 菜单中的 Modify 命令,就可进入对声音波形的操作。可对声音波形进行分解、切割、粘贴和存贮;还可完成放大、回声、淡入/淡出等声音效果的处理。

如上所述,声音编辑软件功能强且丰富,操作简单,能有效地帮助我们实现声音的输入、输出和充分地利用声霸卡的声音处理技术。

(四)声音实用软件(Voice Utilities)

声霸卡软件在 SBPRO\VOCUTIL 子目录下,还给出一些 DOS 命令(以可执行文件.EXE 形式给出),以实现.VOC 文件的播放、录音、连接以及与.WAV 格式的声音文件进行转换等。对这些命令的有关参数进行设置,还可实现背景声音。

1. 给非.VOC 格式的声音文件加上.VOC 格式的文件头

命令格式:VOC—HRD(源文件)〈目标文件.VOC〉

该命令实质上是将其它格式的源声音文件转换为.VOC 格式的声音文件

2. 播放.VOC 格式声音文件的命令

命令格式:VPLAY 文件名.VOC[/B:XX[/T:XX[/Q[/X=“DOS 命令”]

其中,参数 B 设置堆栈尺寸;T 以秒为单位设置播放时间;Q 表示执行命令时不给出屏幕信息,除非是错误信息;X 表示播放声音的同时执行其它 DOS 命令。显然,参数 Q 使我们能保持屏幕画面不改变;参数 X 使我们能实现背景声音。这在多媒体软件的开发中很有用。

3. 录制.VOC 声音文件的命令

格式:VREC 文件名.VOC[/B:XX[/S:XX[/T:XX[/Q[/X=“DOS 命令”]

其中,参数 S 设置采样频率;T 设置录音时间,最大录音时间约 18~20 小时;其它参数和含义同前述。这

个录音命令比前述声音编辑软件的录音功能强。在编辑软件中,录音时间很短(几分钟),这在某些场合是不够的。录音命令可克服这种不足。

4. 连接多个.VOC格式的声音文件的命令

格式:Jointvoc/T<目标文件名.VOC>[文件1.VOC][/SXX][/MXX][/RXX][/RE]...

其中,参数S表示在目标文件中加上静止块,静止时间由十进制数值XX设定;M表示加上标记块,标记值由数值XX给出;R设定重复次数;RE结束重复。

例:Jointvoc/Tbbc5.voc BBC1.VOC/R2 BBC2.VOC/RE

执行该命令生成的目标文件bbc5.voc是:

文件BBC1.VOC+重复2次BBC2.VOC文件+BBC2.VOC文件

由前述知,用声音编辑软件录制声音时,录音时间受到限制,我们可用Jointvoc命令弥补这种不足。即我们可用声音编辑软件录制多个声音文件,然后用Jointvoc命令将它们连接成一个完整的声音文件。

5. .VOC文件与.WAV声音文件互相转换

.WAV格式的声音文件,可在Windows中播放,是Windows支持的声音文件格式。声霸卡软件设置了两个DOS命令实现.VOC与.WAV文件的转换。

(五)音乐实用软件(Music Utilities)

音乐实用软件给出两个DOS命令,用于演奏.MID和.CMF格式的音乐文件。很多多媒体软件都支持这两种格式的音乐文件,并建有不少基于这两种格式的音乐库。因此,音乐实用软件中的这两个DOS命令,能帮助我们直接方便地调用现成的优美音乐和实现背景音乐,加上音乐效果。

1. 演奏.MID文件的DOS命令和使用方法

命令格式:PALYMIDI<.MID文件名>[/FMT:TYPE][/DRUM:CHANNEL]

其中,参数/FMT:TYPE设置用基本型还是扩展型演奏,TYPE的值为“BASIC”或“EXTENDED”; /DRUM:CHANNEL定义打击乐通道。

演奏.MID文件的过程是:首先进入\SBPRO子目录,敲入SBMIDI和按Enter,以装入SBMIDI软驱动器;然后进入SBPRO\PLAYMIDI子目录,敲入演奏.MID文件的PLAYMIDI命令并按回车。按Esc中止。当不演奏时,回到\SBPRO子目录,敲入SBMIDI/U和按Enter,可从内存移去SBMIDI软驱。

2. 演奏.CMF文件的DOS命令和使用方法

命令格式:PLAYCMF<文件名.CMF>[/Q][/S]

其中,参数Q表示执行命令时不给出屏幕信息;S表示执行特定的DOS命令,这使我们能实现背景音乐。

演奏.CMF文件的过程是:进入\SBPRO子目录,敲入SBFMDRV和按Enter,以将SBFMDRVV软驱动器装入内存;进入SBPRO\PLAYCMF子目录,执行PLAYCMF命令。按Esc中止。当不演奏时,回到\SBPRO子目录,敲入SBFMDRV/U和按Enter,从内存移走SBFMDRV软驱。

(六)声霸卡标准接口解调器(SBSIM)软件

SBSIM是一个驻留内存程序,该驻留程序使用户能够通过一些命令在多媒体应用中加入声音和音乐。特别地,能实现背景音乐和背景声音及声音和音乐的一些特殊效果。

1. 在声音处理程序(VOICE.EXE)中使用SBSIM

运行VOICE.EXE处理程序的一些命令前,首先要装入SBFMDRV和SBMIDI这两个软驱动器,它们是

内存驻留程序。装入方法是在\SBPRO子目录下敲入SBFMDRV和按回车,接着敲入SBMIDI和按回车;然后要装入SBSIM软件,装入方法是在\SBPRO\SBSIM子目录下,敲入SBSIM和按回车。

当完成SBFMDRV和SBMIDI软驱的安装及SBSIM软件的安装后,我们可以执行VOICE.EXE处理程序中的一些DOS命令。

(1)播放声音文件

首先执行VOICE/SPEAKER:ON命令,打开喇叭;然后执行VOICE/START:<.VOC声音文件>命令。则计算机开始播放指定的.VOC文件。在播放声音的同时,计算机返回处于DOS状态,你可以执行其它的DOS命令。

(2)停止、暂停和继续播放

在DOS状态下敲入VOICE/STOP和按Enter则停止播放;敲入VOICE/PAUSE和按Enter则暂停;敲入VOICE/CONTINUE和按Enter则继续播放。

(3)使用声音文件中的标记(Marker)块

当我们在声音文件中设计了标记块并使用Wait命令,就可在播放声音文件的同时,并不是马上返回DOS,而是当碰到标记块或当声音文件结束时才返回DOS。这种功能,使我们可在某些设计中加入标记块来控制程序的执行,即碰到声音标记块后才执行某一特定功能程序或命令。

例如:首先执行VOICE/SPEAKER:ON,然后执行

VOICE/START:BBC5.VOC /WAIT:| | |(| | |是在文件BBC5.VOC中设置的标记块),则计算机在播放此声音文件并碰到标记块或声音结束后才返回DOS状态,这时我们再根据需要安排执行特定的应用程序或DOS命令。

2. 在音乐处理程序(Music.exe)中使用SBSIM

在运行Music.exe之前,也要先

装入 SBFMDRV 和 SBMIDI 软驱动器及 SBSIM 软件。然后才能在 DOS 状态下执行 Music.exe 中的 DOS 命令,如演奏.CMF 或.MID 格式的音乐文件的命令:Music/START:<.CMF 或.MID 文件>;MUSIC/STOP 则是停止演奏;MUSIC/PAUSE 则是暂停等。使用这些命令播放音乐的同时,计算机处于 DOS 状态,因此我们可以在播放音乐的同时执行其它 DOS 命令,从而实现背景音乐。

(七)其它在 DOS 环境下的应用软件

声霸卡有两个实用程序控制声霸卡混合器,从而可调节声霸卡支持的各种声源的音量、增益和音质。这两个应用程序就是在\SBPRO 子目录下的 SBP-SET 和 SBP-MIX;此外,声霸卡还配有 CD 激光唱机驱动软件,使用户能够控制和播放 CD-ROM 驱动器上的音频压缩光盘。播放 CD 光盘的方法是:在\SBPRO 子目录下敲入 CDPLYR 并按回车,这时屏幕上出现控制板;再用鼠标操作控制板按键进行播放、暂停、停止等。

前述即为声霸卡在 DOS 环境下的主要软件功能特点。下面是声霸卡在 Windows 环境下的主要软件功能、特点。

三、声霸卡在 Windows 环境下的应用软件功能特点

在安装声霸卡软件时,计算机自动在 Windows 程序管理器内建立了一个声霸卡窗口,在该窗口建立了 JukeBox、Creative WareStudio、Soundo'LE 等应用程序的图符。进入 Windows 后,打开声霸卡窗口,在该窗口的某个图符处按两下鼠标左键,即可进入该应用程序。

JukeBox 应用程序使我们在 Windows 环境下播放 MIDI 文件;而且可将一系列.MID 文件排序、连续播放。

Creative WareStudio 应用程序使我们在 Windows 环境下实现对.WAV 格式的声音文件的操作,如录制.WAV 格式的声音文件;播放.WAV 格式的声音文件;对.WAV 文件进行编辑;对.WAV 文件的波形进行切割、移动、粘贴、混合、放大、渐入、渐出和回声等特殊效果处理。

WareStudio 应用程序必须在 Windows3.1 或更高版本的环境下运行,且 Windows 要设置成加强模式,并要求计算机有足够的内存,才能对声音波形进行编辑和录制。当波形文件太大时,不能调入内存编辑。

WaveStudio 交互界面好,其交互接口界面有五大部分:菜单棒、按钮棒、编辑窗、预览窗和状态棒。菜单棒有 7 个菜单:File、Edit、View、Special、Window 和 Info。File 菜单的命令主要对声音文件进行装载、存盘、关闭等操作;Edit 菜单完成对声音波形的切割、复制、粘贴和混合等操作;Special 菜单使用户能对声波加入回声、淡入淡出、放大、复制和倒转播放等特殊效果处理。另外,执行 Edit 菜单或 Special 菜单命令之前,必须先使用 File 菜单中的 OPEN 命令装载声音文件;View 菜单使用户能根据需要关闭或打开预览窗、按钮棒、状态棒或调节波形放大比例;Windows 菜单用于安排窗口。按钮棒有一些按钮实现对.WAV 文件的操作,如 PLAY、STOP、PAUSE 和 RECORD 按钮实现.WAV 文件的播放、停止、暂停和录音操作,在使用 PLAY 播放之前,首先要用 File 菜单的命令装载文件;按钮棒还有三个控制框实现对波形的操作,Start 框显示所选波形起始位置,Size 框显示所选波形尺寸,Zoom 框显示波形放大比;编辑窗显示所装载的.WAV 文件的波形,在该窗口我们可选择和编辑一段或全部波形;预览窗显示当前编辑窗

的整段波形;状态棒显示当前波形文件的播放时间、波形类型、波形频率等信息。

另一应用程序 Creative Soundo'LE 使用户能够播放和录制声音文件,支持目标连接和嵌入,即我们可以在其它一些应用程序如 Write 文本、Cardfile 文本中插入声音波形文件。

Soundo'LE 有 File、Edit、Options、About 四个菜单,File 菜单完成装载文件、存盘等操作;Copy 菜单命令用于将声音文件数据复制到书写板(Clipboard),以便能进行嵌入或连接;Options 菜单命令可设置录音参数、调节音量等;About 菜单显示有关版权信息。Soundo'LE 还设有六个按钮用于对声音文件进行播放、暂停、前进、停止和录音等操作。使用播放按钮播放前,首先要执行 File 菜单命令装载文件,使用录音按钮录音之前,首先要执行 Options 菜单的录音设置命令设置参数和使用 File 菜单的 New 命令清除内存中的其它声音数据。Soundo'LE 可使声音波形文件嵌入或连接到其它一些应用程序中,嵌入方法有两种,具体的操作过程与 Windows3.1 手册介绍的嵌入方法类似,故不再详细介绍。

还有几个在 Windows 环境下的应用程序,如 Sound Blaster Pro Mixer 应用程序,使我们在 Windows 环境下调节各种音源的音量;Talking Scheduler 应用程序,可帮助用户快速容易地安排生动的约会,如安排约会日期和时间,约会人物,听到约会问候语等;游戏程序 Creative Mosaic,还可测试人们的智力才能和灵活技巧性。

092

正大数据修复

电话:(020)5516961-5931

BB 机:(020)3322322 呼 4364



CPU+RAM+FPGA

——《PLD 应用设计技术》、《FPGA 原理和应用设计》两书简介

80 年代,数字电子系统的结构模式,可概括为:CPU+RAM+PLD。PLD(Programmable Logic Devices 可编程逻辑器件)汇集了大规模集成电路的众多优点,改变了传统系统设计的实现方法,设计者可随心所欲地用 PLD 完成各种规模系统的设计,推动了数字系统的开发应用。

随着现代数字系统大容量、高速度、现场灵活编程设计的要求,80 年代中期一种新型的用户现场可编程的集成电路 FPGA(Field Programmable Gate Array)也就应运而生了。

有人预言,90 年代的先进数字电子系统将以 CPU+RAM+FPGA 为其结构特征。所以,许多权威人士指出,如今不懂可编程逻辑器件的工程师,犹如当年搞电子技术不懂晶体管一样。

今天,由于改革开放,引进先进器件和技术的工作变得容易些。为此,希望凡是搞硬件电路设计的工程师们,都要学习和应用 PLD 和 FPGA 器件,这是使我们开发的产品赶上和超过世界先进水平的必由之路。

电子工业出版社广州科技公司组织编写了《PLD 应用设计技术》和《FPGA 原理和应用设计》。二本书的作者都是有关领域开展研究和应用的中青年技术骨干。前者为中国科技大学六系朱世鸿老师,后者为深圳大学电子系朱明程老师。

以上二书将由电子工业出版社于 94 年 3 月和 5 月正式出版发行,与广大读者见面。

·PLD 实用设计技术 16 开本 558 页

定价:38.50 元

该书主要章节如下:

第一章 PLD 原理和应用;第二章 PLD 编程

器设计;第三章 PLD 应用设计软件包;第四章 PAL、GAL 应用开发实例;第五章 EPROM 在数字逻辑系统中的应用;第六章数字系统的可靠性技术等。

全书选用大量具有典型代表性的应用实例,各例稍加或无需修改即可实用于各种规模的数字系统中。

该书还从工程技术的经验角度,详细讨论如何抑制数字系统中的干扰问题。

·FPGA 原理及应用设计

估价:28.00 元

该书系统介绍九十年代国际电子应用设计领域广为注目的新型器件——FPGA(用户现场可编程门阵列)集成电路的原理、结构及应用设计技术。本书第一、二章,主要介绍和讨论了各种可编程器件的原理和发展,并着重介绍和分析了 FPGA 的概念,结构,参数特性及工作原理。

本书第三、四、五、六、七章主要介绍 FPGA 的应用设计方法和过程,着重介绍了在国际上应用最广的 Xilinx FPGA 开发系统,讨论了设计输入,设计实现,设计仿真的具体方法和过程及注意要点。第八章是应用设计的实例介绍。

以上二书的相继配套出版,包括了 PLD 和 FPGA 的全部实用技术,对于航空航天,电子通讯,计算机智能控制,仪器仪表,自动化应用设计领域,从事电子系统,尤其是数字应用系统,ASIC 单片系统,程控产品设计的专业科技人员,应用设计工程师,大专院校电子专业师生是一本具有实用价值的新技术应用参考书。本书可作为大学电子学科高年级电子系统选修教材和技术推广培训教材使用。(电子工业出版社广州科技公司供稿)

电脑与法律

用户界面的版权保护讨论

· 王桂海 ·

用户界面(user interface)的版权保护是软件保护中很引人注意的问题。1990年 Lotus 公司诉 Paperback 公司销售 PV-planner 软件侵犯了其电子数据表格程序 Lotus1-2-3 的外观与感觉(look & feel)版权一案,就是比较典型的案例。此案以原告胜诉终结。作者在《电脑》杂志 1991 年第 5 期曾撰文对此案作过分析,并介绍了案中原告、被告、美国版权局,美国联邦法院几个方面的意见^[1]。近几年来,在外国和我国软件保护研究论著中,仍然对用户界面的版权问题展开讨论^{[2]~[6]},到现在也还没有统一的想法。本文打算就这个问题,对流行的几种观点进行介绍和分析。由于篇幅所限,对下文提到各案例的案情,不作详细介绍,有兴趣的读者可以参阅本文列出的参考文献或其它资料。文中给各种论点冠以称谓,只是为了行文的方便,未必是恰当的。

对用户界面版权保护看法,总起来无非是两大方面:一是坚持不予保护(或只给很弱的很有限的保护);另一方面则认为应给予足够的保护。前者主要有列几种论点:

一、“再创作论”

这种论点认为,对已有的某种界面形式,如果以另一个不同的程序来实现,那末,后一程序是属于再创作。其创作所注入的智力劳动远远大于对界面形式的设计,因此,不应因利用了某个已有界面,便认为是对这个界面的侵权。这种论点出自 1978 年美国的 Synercom 公司诉 University Computing 公司一案,这也是最早的关于计算机界面保护的诉讼案。原告控告被告复制了它的输入格式(input formats,一种用于计算机的穿孔卡片),尤其是它的数据顺序。但法院却认为,即使是被告确实编了一程序去接收原告的输入格式也不算侵权。法官 Higgenbotham 指

出:编写程序与对问题的描述,不是一回事,是一种创作。从对问题的一般描述到问题的解决,不管用哪一种语言去编计算机程序,都与一般的文字作品的翻译有很大的不同,也不同于把某种语言的程序翻译成另一种语言的程序。在大多数情况下,某个问题即使有了足够详细的精确的公式,但要保证它能转换为一系列明确的计算机指令,仍需要有丰富的想象力、创造力、独立思考和单独地作出试验。这样得到的程序决不能把它看作只是一个复制品或是问题说明书的译本。程序与说明书,不论是物理特性还是现实的用途,都是完全不同的,它们实际上是两种不同的表达形式而不是同一形式的不同版本。这样的观点是说,如果只继承已有的用户界面而进行程序的另外编制,基本上不同于复制支持这个界面的程序的内在结构,因而不是侵权。所以界面本身不应受保护,只有实现该界面的程序才是应受版权保护的。

二、“同一性论”

也是在同上的案中,被告坚持认为,它确实借用了原告的思想原理(这不算侵权),但也仅仅只借用了思想原理而未涉及原告所采用的表达,问题是原告的表达和思想是分不开的。因为原告的接口所采用的原理,恰恰就是这些数据的特定的序列,没有办法把后者和前者分离出来,这就是思想/表达的同一性,所以,不能认为是侵权。法院最后承认了这一点。(按“思想/表达同一性”,也可用于其它作品的判断)。

三、“制约论”

这一论点出现在关于 Lotus 案以及其它一些案中,代表者是 Lotus 案的被告和美国版权局。他们认为,有些人机界面(例如屏幕),结构过于简单,而且,用户友好的要求又往往是共同的,通用的。屏幕设计者在独立设计的情况

下,由于受到制约,可选择因素及范围极少,极易出现雷同。因而,“一般说来,在一个特定形式中,由词组或简单短语构成的屏幕菜单及类似的功能接口是不能作版权登记的”。如果把程序看作文学作品来谋求保护,那末,“在菜单中的短语,算不上是文学作品”。因而用户界面不应列入保护范围,这一观点没有被法庭接受。

四、“标准化论”

这也是 Lotus 案中被告坚持的一种观点。这种观点认为,对于已经成为大多数用户熟悉的,习惯用的技术,应把它们看作是一种工业标准。Lotus 1-2-3 就已经成熟到这个地步,如果不允许自由采用这种方式,实际上是强化了原作者的垄断地位,不利于社会公众利益。但因为美国在法律条款上还没有这样的先例,把未经法律承认的事实上的标准化宣布为为了公共利益而可以随意采用。所以,这一论点也没有被法庭接受。

五、“文字保护论”

这一观点认为,程序是一个作品,按作品版权划分的范围即思想/表达的原则,作品的表达形式是文字。因此,程序的保护范围应就是相当于文字的编码,对于非文字部分(nonliteral elements of computer program),那是属于思想的范畴,不应给予保护。而人机界面,则是非文字性的东西(例如,在屏幕中表现出来的是带有某种功能的菜单),而相雷同的样式可以由完全不同的程序生成,那末,只须保护程序本身,而不应延及这些程序产生的结果。也就是说,只要程序本身不是抄袭的,则即使屏幕出现雷同,也不应视为抄袭。这个观点和“再创作论”很相似,但立论的角度不同,而且对保护范围提得更具体。

当然,还有其它一些论点,如为了利于技术的发展,鼓励兼容,鼓励竞争,应当适当限制对用户界面的保护范围。

持相反观点的,除 Lotus 案的原告外,还有 Broderbund 公司诉 Unison 公司, Digital 公司诉 Softklone 公司, Apple 公司诉 Microsoft 公司与 HP 公司, Xerox 公司诉 Apple 公司等案的原告。这些案件,包括 Lotus 案中,多数是原告(要求取得界面保护)获胜。这也反映了法官的观点。他们致胜的最主要根据,是基于能从非文字部分中分离出一种对思想来说是非本质性的因素(nonessential elements)。这种非本质因素,则是属于表达的形式,如果是原创的,那就应受到版权保护。

这一论点,有人经过考察,认为最早不是来源于人机界面的纠纷案而是源出于 1986 年那个著名的 Whelan 诉 Jaslow 案。通过严格地分析 Whelan 案中法院的推理步骤,便可以看出这个思路,这些步骤是^[7]:

1. 版权法是保护文字作品的;

2. 计算机程序属于文字作品;

3. 因此,版权法可以扩充到保护计算机程序;

4. 版权法的目的是要对作品实行保护,在保护发明、创造和允许推广、扩散之间,找出某个平衡点,既要保护作者的积极性,不允许随意扩散;又要鼓励技术的改进、革新和推广;

5. 版权法并不保护创作文字作品时所必须采取的手法、情节。因为保护这些东西,就会形成给第一个作者以垄断权,打破了上述的平衡;

6. 对一个程序来说,反映思想本质的非文字性成份,就象是作品的情节和手法那样,不应给予保护;但是,有一些非文字成份,却不属于思想的本质因素;

7. 给程序的非文字部分中,对思想来说是非本质因素(not essential to the program's idea)给予保护是最好的平衡,在这前提下允许技术推广、扩散,这与版权法的目的是是一致的;

8. 计算机工艺、技术的进步与科学、艺术等其它领域的进步没有本质上的不同;

9. 因而,用处理其它领域中版权原则去处理计算机程序是适当的。

这几个步骤中,最关键的是第 7 步,这一步把非文字性的东西划分为属于思想本质因素和非思想本质的因素。这样,便把思想表达的分界线从文字表达层次推到非文字表达层次上去。也就是说,作品的文字表达固然要得到保护,但文字表达之上,如果能在思想层次中分离出一些非本质因素,那末,这些非本质因素也可以是属于表达的范畴,可以得到保护。这其实正是人们较熟知的 SSO 准则。

随后的几个关于人机界面的案子,便是接受了这一原则而进行判决的。法官能够在非文字的部分中,仍然分离出一种不是属于思想的因素,如菜单的整体结构、每个菜单行中的命令的次序、代表每个命令的字母或词的符号标记的选择、这些符号在屏幕上显示的方式(亮度、字母采用数、简化方法,是否划线等)、使用的菜单结构种类及提示等等。把这些因素结合起来,尽管当中某些因素选择的自由度很小,但整体来看却可以有多种的表达形式。因而,综合起来便可以受到版权的保护。

从这一论点出发,结论便是:虽然人机界面是通过程序来实现,但仅仅程序的不同还不能取得版权,还应进一步看看属非文字部分的界面本身是不是充分反映了作者的原创性。在 Lotus 案中,法院引用了专家的论述:“大量的原创性工作是在计算机程序的概念化阶段及用户界面设计时,而不是在编码时,创作一个合适的用户界面比将界面转化成机器指令(按指编程序)更加复杂,需要更大

的创造力,原创力和洞察力。”法院认为,若只保护计算机程序的非文字部分就只能保护计算机程序的原创性开发工作的很少一部分,而将更加重要的开发过程的原创性部分排除于保护之外,其结果将为版权法的目的——鼓励创造——相违背。

这就是后来的几个案子受 Whelan 案影响的结果。有人甚至认为,如果 Synercon 案真是 Lotus 案的唯一先例,对后者的判决肯定不会是现在这个样子。

对上面两种对立的观点,从判案的实例可以看到,多数支持“接口应受保护”的主张。但作者则有以下看法:

1. 认为大量的原创性工作是在“在计算机程序化概念阶段”,便很可能推论为程序的概念化工作要受到版权保护。

2. 把非文字部分再划出属于思想的基本因素部分加以保护,实际上是把思想/表达分界线向抽象层次推进。

以上两点,其结果是要扩大软件版权的保护范围。这样去扩大,不一定很是恰当。事实上,不论在美国还是其它国家,对 Whelan 案的非议并不少,而且美国也有完全不按其准则办案的实例。

3. 笼统地认为计算机技术的进步和其它科技、艺术的进步发展没有什么差异,这也是不适当的。计算机科学的进步的一个特点是技术更新极为迅速。看不到这一点,把它当作一般作品去扩大保护范围,又由于版权保护的期限较专利保护长得多,按照这样去处理,会给技术的发展带来不利影响;

4. 人机界面是一种程序运行的结果,其表达形式受到技术的限制,某些状态(例如菜单形式)显得比较简单。这种状态下表达形式的定义应从严,给予保护的应相应地小。而另外一些状态如目前已发展起来的多媒体技术,则可以放宽保护范围。虽然同是界面,但因为后者独创的余地很宽,应给予较大范围的保护。所以,对界面版权保护的纠纷,应在全面考虑技术所能提供的支持后才作判决。

5. 要及时使某些成熟的技术变为“标准化”。由于有些界面已普遍被用户接受,产品已在市场上形成主流,为防止造成垄断局面,及时确定这些成果为标准化,促进技术发展,是十分必要的。目前,对一些专业应用的菜单格式,就可以这样做。

6. 可以考虑在版权保护中另外采取一些新措施。如美国有一种意见认为“同一权利人所拥有的项计算机程序只能形成一项作品,只能在程序编码或屏幕显示映象两者中选择其一进行版权登记”。这是一种折衷的,并未确定的观点。有人还建议应当让美国版权作品新技术委员会(National Commission on New Technological Users

of Copyrighted works,简称 CONTU)专门来研究这些问题,为用户界面以及其它程序中的非文字成份探求一种新的保护形式。这些都值得我们重视。

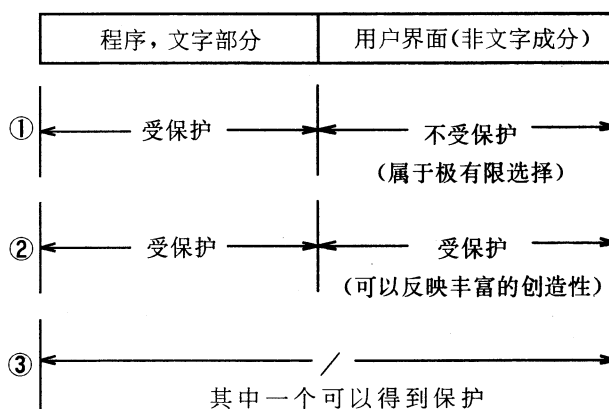
为了对上述问题有个较清晰的描述,我们可以用下图来表示各种观点。图的左边是表示为取得界面效果而制作的程序;右面则是界面自身。

①认为,程序应受到保护;界面不应受保护,因为它既不属于文字部分,又受到很大限制。

②认为,程序无疑要受保护,但界面也反映了作者的独创性,应当受到保护。

③认为,界面是程序产生的,是程序结果及数据的表达形式。程序和界面只应有一个受到版权保护(有人建议界面划入专利的外观设计保护范围)。

以上只是反映了对版权的保护范围的各种观点。一个具体的程序或界面能否得到保护,最终取决于其所具备的独创性程度。



参考文献:

- [1] 王桂海,“Look and Feel”的争论及其影响,电脑,1991(5)
- [2],[7] Nelson R.Capes, The Louts Decision: Analysis and Recommendation for Non-Copyright Protection, Software Protection, July 1991
- [3] 应明, 计算机软件的版权保护, 北京, 北京大学出版社, 1991, 145-159
- [4] 中国软件登记中心等编, 计算机软件著作权保护工作手册, 北京, 电子工业出版社, 1993, 168-178
- [5] 王桂海 张甫筠, 软件法律保护中的几个“模糊”问题, 北京, 张明主编, 中国计算机应用论集, 1992, 45-49
- [6] 张柳坚, 计算机软件版权原理与实务, 安徽, 中国科学技术大学出版社, 166-170

电子数据交换(EDI)面面观

(四)、(五)

海关管理干部学院 李凤仪

招商局集团 EDI 应用 已初具规模

(1) 1986 年 6 月开始陆续安装了 IBMSYSTEM/38、IBMAS/400 等 11 台电脑。

(2) 建立了招商局集团局域网 (CMHLAN)，它通过专线 (LEASED LINE) 和交换线路 (SWITCHED LINE) 与香港地区网 (DIALCOM)、中国公用分组交换数据通信网 (CNPAC)、美国麦克唐纳—道格拉斯资讯系统公司国际网 (TYMNET) 和美国通用电气国际业务公司国际网 (GEIS) 等连接，实现了与世界各地的电子数据交换。

(3) 已建成招商局集团 EDI 系统 (CMHEDI)，该系统根据 COSCO (中国远洋公司) 货物运输单证 EDI 标准。

1988 年 3 月 21 日，招商局集团航运有限公司 (CMCL) EDI 系统正式投入运行，通过 TYMNET 的 EDI 信箱与 COSCO 的美、日代理交换舱单。从 1990 年 4 月开始，用点对点异步通信方式与香港地区揽货公司传送提单。1990 年 9 月使用香港国际数据互传服务 (IDAS) 和 CNPAC 与上海外代交换集装箱货运舱单。1991 年 5 月与 GEIS 共同开发的货运单证 EDI 系统通过技术鉴定。1992 年通过香港公用数据交换网 DATAPACK 和其他专用通信线路与集装箱堆场、码头进行货运文件电子数据交换，可与 GEIS 的集装箱管理信息系统 (EMS) 交换数据。

(4) 计划安装 MULTINET 多用

户 EDI 系统，可支持各种 EDI 标准，能自动生成、处理、收、发、检查报文，可实现与政府机构的电子数据交换。

中国远洋运输 (集团) 公司是我国最大的运输企业，以国际航运为中心业务，有 600 余艘船舶，航行于 50 余条国际航线和世界 1100 多个港口之间，直属 200 余家国内外代理机构，与世界上数以万计的公司和客户进行商业往来，每天信息量达百万字以上，其 EDI 应用已全面展开。

(1) 80 年代初已应用计算机管理，使用中、大型机 IBMAS/400 和 IBMES9000 系列。

(2) 与美国 GE 公司合作开发 SHIPDES 网络通讯，实现国内外代理间传送数据。也选用邮电部 CNPAC 网开展网络通讯。按 ISO/OSI7489 开放系统的互联标准开发“中远货运单证交换系统”，通信环境为电话网 (IDD 方式) 和 CNPAC，成为 EDI 服务网络系统。

(3) 开发了靠拢 EDIFACT 标准的“货运单证 EDI 报文标准 (3.1 版)”、“货物追踪信息 EDI 报文标准”和“船期表 EDI 报文标准”，还计划 93 年底完成如下标准制定：

- 国际航运集装箱尺寸和类型代码
- 国际航运条款、代码
- 国际航运货物包装和分类代码
- 国际航运运费和费用项目代码
- 国际航运提单 EDI 报文格式

(4) 《代理公司出口货运单证 EDI 应用系统》已运行 (由上海外轮代理公司开发)，使用后使中远船队得以在美航线上立足发展，具有与先进国家

船队竞争的能力，在日本航线方面应用后可为中远船队减少支出近 70 万美元。

上海外轮代理公司是对外贸易运输中联系船、货、箱、港、海关、银行、保险公司、商检等方面的信息枢纽，外贸运输中占最大比重的主要是船与货，及作为包装或运输工具的箱 (集装箱)。当这些货与箱装卸或装拆时，就会发生频繁的与国内外方方面面的交换行为，交接的凭证主要是各式各样签字有效的单据，如提单、舱单、装箱单等。货运单证必须在船抵港前一天到达，而特快快递和 FAX 传真方式不能保证航运单证的质量和时效，又要重新制成向港口、海关申报的格式。该公司和远洋公司的其他国内外代理一起从 1990 年起开始使用计算机处理单证，协助中国远洋运输总公司制定货运单证数据通讯格式和代码，即中远代理行业的 EDI 雏形；1991 年 5 月起正式使用国内的 CNPAC 网和美国 CE 公司的网络，开展代理公司之间的电子数据交换，所有中远集装箱航线上的舱单都可采用电子数据交换的方式进行国际交流，取得较好效益。以日本航线为例，近百票提单的货的数据大约 2M 字节，从转换生成到通讯完毕，对方代理接收到自己的计算机系统里去，大约只要一小时左右，质量高，安全迅速，不要人工干预，大大提高了工作效率，初步显示出采用 EDI 方式的优越性。

《代理公司进口单证 EDI 应用系统》由天津外代和上海外代合作开发，

1993年下半年可投入运转,此系统可进行转船处理,可解决转船单证重新输入问题,由计算机来进行转船分配。

《远洋系统船舶运费舱单 EDI应用系统》可对代理公司代算代收的运费进行监督、复核及进行成本核算。1992年6月由上海远洋公司开发,1993年下半年可部分投入试运行,全部投产后,可节省舱单邮寄费44%,提高传送速率81%,使EDI应用进入一个更高层次。

《出口理货单证数据 EDI应用系统》由上海外代和上海理货公司合作开发,应用上海市邮电局MHS服务,1993年下半年可实用化,应用后将提高集装箱出口船舶的数据准确性,核对船图和理货清单的工作量可全部取消,节约人工每月约200人天,避免错港、漏船等责任事故和倒箱费的经济损失。

(5)组织落实和普及教育、培训,派人参加联合国贸发会组织的亚太地区EDI学习班,派员外出考察、交流,邀请美国、新加坡等EDI专家前来讲课,大力促进EDI的意识和知识在中远的普及和推广。

中国外运集团是我国最大的国际货运代理企业,承担我国进出口贸易三分之二以上的货物代理业务和三分之一的货物运输业务。在全国各地有分公司,在海外有400多个业务网点,年平均营业额50亿元。目前计算机应用面已达90%以上,在海上已用于订舱、船舶管理、集装箱跟踪管理和租赁管理;在空运上已用于空运快件管理。自90年代以来,逐步与日本、美国、新加坡、香港等地的航运界EDI协会、企业进行了接触,下一步完成海运业务管理EDI系统和空运业务管理EDI系统,再与海外代理、国内有关外贸公司、供货公司、海关、银行等进行电子数据交换。系统实现后,可为客户提供电子订舱、查询服务,可加强船舶调度管理,可提供准确的货物跟踪,可加快运输,预计年经济效益1.5亿元,大大增强企业在国际航运界的竞争能力,开拓了巨大的货运市场,使企业成为全面走向现代化、国际化、实业化的国际运输集团。

中国电子进出口总公司是国家大型外贸专业总公司,是电子行业对外贸易的主渠道。公司在“七五”期间

已初步建立起内部的计算机管理信息系统,建立了30多家分公司间的数据通信网络,成为世界贸易中心(WTC)商业信息网络的会员,并联入了北京公用电子信箱系统和国家外汇管理局的中国金融外汇信息咨询网络,为开发EDI应用系统打下了基础。通过开发和建立公司的EDI应用开发平台,在已有局域网基础上扩充和升档,完成合同、单据等贸易文件的收集、整理、发送、追踪、反馈、制单和自动检测;通过与公共数据网络的接口,实现网内各结点间的信息交换;通过电话网络、CNPAC公共数据网及国际公共数据网,实现与国内有关单位、国外客商直接进行信息传递,最终达到建立一个高效、安全、稳定、可不停顿运行的一体化网络,支持我国进出口贸易高层次地与国际水平同步发展。预计公司每年可增加上千万元的经济效益。

中国化工进出口总公司是一个有70多个海外分支机构的跨国公司,年创汇100亿美元,作为我国国际化经营外贸企业的试点单位,为我国EDI的发展创出了一条新路,该公司开发化工进出口贸易EDI系统,1992年7月已实行了进口业务、运输、保险、财务、计划统计、投资项目管理的内部数据交换,开发了进口业务网络系统和财务业务网络。公司总部与国内各直属公司、子公司、投资企业、海外公司和海外投资企业实行了联网,实现了贸易单证由计算机处理与生成。最近该公司已加入EDI国际网络,即已基本完成EDI的第一阶段——内部主机和终端装置的联网,和EDI的第二阶段——同行业间的联网,以及EDI的第三阶段——跨行业的联网。今后还要和海关、银行、保险公司、港务局、航运公司、商检机关、口岸仓库、促贸会、进出口审批机关等联网。该公司还同美国GE公司签订了合作开展电子政务服务的协议。

中国抽纱品进出口(集团)公司山东抽纱进出口公司是一家有七个直属企业和十个中外合资企业、年出口额上亿美元的公司,1991年11月,山东抽纱公司与美国通用电气国际业务公司在京签署了EDI订单系统测试协

议,用这个系统接收国外客户的订单,生成有关贸易的EDI单据,用EDI方式传给美国客户,用EDI方式向海关报关、与银行结算,预计年经济效益可达百万美元,还带动了贸易链上相关业务部门,形成青岛口岸跨行业协调、合作的局面。

湖南省外经委计算中心与湖南粮油食品进出口公司决定开发鲜活产品报关EDI系统。

4. 海关EDI工作概况

海关一直关注EDI发展的趋势,积极派员参加了我国有关方面组织的EDI研讨等各项活动。海关总署明确提出:EDI的应用对促进我国对外经济贸易事业的发展,对贯彻海关工作“促进为主”方针和加强宏观监控的职能都具有重要的战略意义。

海关85规划期间将制定报关、报文标准并引进DEC公司的EDI软件,在上海虹桥机场、北京首都机场进行EDI应用试点。

为实施EDI,海关在以下方面已经创造和正在创造必要的技术环境:

(1)海关已基本建成了海关业务计算机管理体系。

至1992年4月,整个海关系统已装备中小型VAX系列计算机系统107套,微型计算机系统1000套;终端设备1800台,技术干部1312人,设备大部分具有八十年代中期的国际先进水平。计算机设备已在167个海关单位中普及使用,全国各海关已建立了大大小小的众多计算机应用系统,已开发或移植运行的应用软件达七百多个,几乎所有海关业务环节都有了比较适用的技术项目,已进入由单机应用向系统化、网络化,从单项业务管理向综合业务管理方向发展的阶段。1992年,业务覆盖面包括100%的海关已实现用计算机进行海关贸易统计的数据收集处理,80%的海关已实现用计算机进行进出口货物征税、税收稽查及其他数十种单项应用项目,九龙等9个主要海关实现了报关自动化综合处理。

1991年,报关自动化系统—H883通过国家技术鉴定并开始推广使用,H883所处理报关单已占全国海关报关单总量的一半,这为EDI的研究和

开发准备了条件。H883 系统有与外单位,如经贸部贸易局、民航机场、港务局、铁路局、外运公司、外轮代理公司、报关公司和银行等部门和单位的接口,为与这些单位联网打下基础。

(2) 建立全国海关计算机网络—909 工程。

全国海关网络分为三级,即全国海关网、地区网和以直属海关为中心的关区网,目前广州、九龙、天津、北京等海关已建立了本关的关区网,以广东分署为中心的连接九个海关的广东地区网建成并投入使用,总署规划,近期要建成 20 个以上关区网以及总署至 30 个直属海关的全国初级网建设,并逐步升级为中上水平的计算机网络,以总署为中心的全国海关网拟于 1994 年建成,这样,实施 EDI 就有了通讯基础。

(3) 改革现有报关制度。

正在着手取消目前这种海关对数千家企业报关的做法,建立专业报关行向海关办理报关手续的制度,为计算机报关作好组织准备。

(4) 开展了报关单电脑预录入业务。

报关自动化系统 H883 应用后,开展了报关单电脑预录入业务。该项业务是由预录入公司通过海关终端将报关单数据在正式报关前录入海关计算机系统。目前,全国已有 45 个预录入公司向海关办理了数据预录入业务。这是电脑报关的前期准备工作。H883 系统的预申报子系统的设计,是迈向 EDI 的第一步。

(5) 编制新的报关单。

海关正在以海关合作理事会推荐的单一报关单(SGD)格式为基本,设计和印制新一代的报关单,以适应 EDI 的实施。

(6) 1992 年采用了《商品名称及编码协调制度》,以适应今后开发 EDI 的需要,这个制度是一种新型的、系统的、多用途的国际贸易商品分类制度,前面说过,外贸部门许可证商品亦已采用这个制度。

(7) 正在开发电脑自动审单系统。

(8) 改变了传统的只在进出口岸设海关的原则,设关延伸到内地,使企业可在当地报关,为实施 EDI 提供了极大方便。

(9) 组织力量正在制订海关 EDI 报文标准。例如,1993 年 3 月推出进(转)口货物舱单标准(CNC101)CUSCAR 等。

5、保险业动向

保险业正为实施 EDI 积极创造条件。

(1) 使用中型计算机处理业务,且与中国银行、工商总局等单位联网。

(2) 与美国邓白氏集团的信息网络联通,可通过香港、美国的资料库查询世界各国主要贸易公司的有关情报。

(3) 正在编制《保险业务分类与代码》及《保险业务凭证格式及数据元》两个国家标准,推动管理工作规范化、电子化。

(4) 在条件较成熟的地区已开展了 EDI 应用试点工作,初见成效。

6、抓住机遇,掀起 EDI 应用热潮。

发展 EDI 是大势所趋,在我国已开始由上至下在各领域重视起来,并开始起步。目前的发展大体上是先外后内,从流通到生产;通信基础设施先行,外贸、海关为先导;从简单起步,向标准发展;逐个应用领域集中力量打歼灭战。

目前,全球贸易额已上升到万亿美元的数量级,在世界各国贸易中我国位居 11 位,我国进出口总额已占国民生产总值的 30% 以上,外国评论认为,大中华经济圈包括大陆、台湾和香港,其经济实力和发展潜力使它到 2000 年将成为继美国、日本、德国之后的第四大贸易支柱。港台 EDI 已接近实用化,我国只有大力发展 EDI 才能顺应潮流,如果跟不上全球 EDI 应用步伐,势必使我国在国际贸易中处于劣势,并将直接阻碍我国经济发展,无论是从创造国际贸易伙伴条件、提高通关效率、维护国家的经济和贸易地位还是从抓住机遇、有效地参与国际商业竞争和创造巨大商贸效益来考虑,加快 EDI 在我国的应用都是十分迫切和需要的。随着我国社会主义市场经济的发展、恢复关贸总协定缔约国地位以及与国际市场接

轨,国内市场将逐步国际化,加快推广 EDI 更显急切,电子工业部已把 EDI 作为电子部的一项重大信息系统工程决心抓紧抓好,在国家“八五”科技攻关中正式立项并投资 500 万元予以支持,计划经过 3~5 年的努力,在一些重要的经济和工业部门中,有计划、有步骤地建立起具有初步规模和功能的 EDI 系统,力争到 2000 年初步建立起覆盖全国的通用 EDI 系统。中央各部门、各行各业、社会各界均要关注、参与、支持和配合,掀起一个 EDI 应用的热潮,架起通向世界市场的金桥。

五、EDI 发展趋势

未来贸易发展的先决条件是谁拥有 EDI 手段谁才有资格构成国际贸易伙伴。1998 年前北美将继续是 EDI 的最大的市场,但欧洲的增长速度将更高,亚太和世界其余地区增长速度会最高,虽然到 90 年代末占有的市场份额依然很小。

据估计,EDI 市场的未来发展可如下表所示:

年	营业额 (亿美元)	增长 (%)	使用单位数 (万个)
1987	1.300		
1988	2.080		
1991	2.506	54.9	
1992			10.0
1993	10.000	40.0	
1995	19.000	56.0	40.0

EDI 市场数年间可增长 8 倍,翻译软件也将很快发展,1991~1998 年间将从 6000 万美元增长到 3.21 亿美元。世界电子数据交换市场将大步前进,随着时间的推移,将从大企业向着与之有业务往来的中小公司扩散,但要达到成功的应用,还有若干路程要走。

095



广州白云山电源设备厂

CWY 系列

高抗干扰稳压电源

地址:510515 广州市沙河同和 电话:7714403 FAX:7705761

EDI 在船舶代理行业中的应用与发展

船舶代理公司在运输行业中一直发挥着重要的作用,招商局货柜航运有限公司(以下简称 CMCL)作为中国远洋运输(集团)总公司(以下简称 COSCO)在香港地区集装箱船舶的总代理,承担 COSCO 十二条国际航线的船舶代理业务,贸易伙伴遍及全球。

1992 年 CMCL 代理 COSCO 船舶 1099 艘次,代理箱量 45 万个标准箱(93 年已达 49 万箱),支线船舶营运 524 航次,支线船舶载箱 74354 个标准箱。几年来公司的电脑应用和 EDI(电子数据交换)技术的推广是伴随着业务的发展而不断完善的,实践证明:EDI 在改善客户服务;加强企业在市场上竞争的能力;提高工作效率;减少差错率等方面具有广泛的发展前景。

一、EDI 系统的建立

1. 建立 EDI 是业务发展的需要

CMCL 成立于 1987 年 9 月,现有职工约 200 人,成立之初美洲线的运输单证就面临着美国海关要求对采用 EDI 方式报关的舱单文件优先处理和审批,否则将被列入黑名单。换句话说:如果不改变传统的 DHL 和随船携带文件的传递方式,因传送和处理资料所导致的时间延误,势必对船东及货主都造成经济上的损失和不利影响。

为了在激烈的市场竞争掌握主动权,并应付不断增长的业务量,CMCL 从高起点出发,在香港招商局电脑部的技术支持下,积极引进移植 COSCO 日本代理在 IBM S/36 机上开发运行的单证制作和通讯软件,经过与美洲线上的有关代理开会,共同商定并确认了单证通讯格式及实施方案。经过四个月的修改与测试,于 1988 年 3 月 21 日正式投入运行。

该方案中,数据发送方根据业务要求,使用 S/36 机上的单证制处理系统输入单证数据,检查无误后,按规定的通讯格式生成顺序文件,再利用美国 TYMNET 网络送到接收方。由于 TYMNET 网在香港没有结点,因此 CMCL 通过 IDD 电话直接由日本进网收发数据。数据接收方从网上取走数据后,再按通讯格式及自身应用系统的要求进行逻辑校验,通过检查后即可装到应用数据库中,从而大大减少了单证数据重新输入的工作量。COSCO 北美代理同时按美国海关要求的格式进行转换后再用本地网送到海关,达到了预期的目标。另外由于采用了 TYMNET 的网络信箱,也解决了不同地点的时差问题。

从 CMCL 在短期内成功地建立 EDI 系统的过程中,我们可以看到:技术和业务之间的关系是相辅相成的,技术的研究和发展应具有超前性,但为了使技术变为生产力并产生经济效益,还必须靠业务来推动。人们通常认为 EDI 是电脑人员的事情,并将大量时间花费在技术实现方面,而忽略了业务部门的作用和实际需要。一个企业内部的 EDP 部门与电脑公司的工作目标有着本质的区别,前者以提供技术服务为主,后者则以销售产品赢利为主,因此以业务为主导,技术做保证,并把重点放在推广和应用他人成功的经验和技术上,这应是企业加速 EDI 系统建立的捷径。

2. 企业内部的电脑化是建立 EDI 的基础

EDI 是 EDP(电子数据处理)的延伸,是一种数据交换技术,对 EDI 的研究主要侧重于通讯格式、转换软件和网络通路等方面,如果接收或发送的 EDI 数据不与应用系统相衔接,就会使 EDI 变为 FAX 的功能,从而失去了发展 EDI 系统的意义。为了巩固和扩大已取得的成果,CMCL 在实现了美洲线的单证 EDI 后,从公司整体业务的角度进行了规划,并开发了一套集成度较高,能实现数据共享的管理信息系统(MIS),该系统的主要子系统包括:代理船舶动态;进出口单证的制作与通讯;运费发票的制作;本地区集装箱动态跟踪管理;全程货物跟踪与通讯;集装箱堆场管理与通讯;财务。

企业内部的应用系统越完善,数据越齐备,实现 EDI 就越容易,效果也越好,这一点往往容易被忽视。

二、EDI 的推广与应用

1. 如何推广 EDI

首先要正确地选择合作伙伴。CMCL 的贸易伙伴大致可分为本系统和外系统二部分,本系统贸易伙伴的特点是业务性质相同,但分布在不同地点,如与 CMCL 各航线有关的 COSCO 代理等。外系统贸易伙伴的特点是分属不同的业务范围,但分布在同一地点,如香港海关、海事处、码头、堆场、货运代理和银行等。本系统的 EDI 一般是由总公司负责推动,外系统的 EDI 则本着互利互惠的原则相互推动。

CMCL 在应用系统基本成形之后,对于本系统、通讯格式、网络投资和代码管理等问题已由总公司统一解决,推广的难点在于其它代理的应用系统是否已经建立,以及是否有足够的技术力量开发通讯收发及数据转换软件。尽管

COSCO 各代理所使用的机型比较统一,但由于应用系统数据结构的设计千差万别,因此数据的转换和分配软件不可能统一,这是影响 EDI 推广进度的“瓶颈”。对于外系统,由于业务范围和性质不同,所使用的代码系统、机型也可能不同,所以必须由双方的电脑技术人员和业务人员坐在一起开会,商定通讯格式、代码转换和移行推广双方的权利与义务等问题,然后分头实施。

例如,CMCL 与香港国际货柜码头有限公司(以下简称 HIT)的 EDI 合作应用中,HIT 使用 HP 主机,前端处理机用 SUN 工作站,操作系统为 UNIX,对外通讯全部由 SUN 工作站经 AT&T 的 HANS 网完成。CMCL 在 AS/400 主机上利用 X.400 通讯软件每 20 分钟上网收发一次数据,从而真正解决了异种机的实时通讯问题,这一成功合作,被 IBM 香港公司称之为开创了亚太地区成功应用的先例。

除 HIT 码头外,香港地区还有 3 个与 CMCL 有业务往来的码头,其特点是本身有电脑人员和设备,也有一套规模较小的应用系统,CMCL 利用与 HIT 合作成功的经验,很快与其建成了 EDI 系统,取得了立杆见影的效果。

综上所述,选择合作伙伴时应考虑如下因素:

- 合作各方都有积极性
- 合作各方的信息交换量大且频繁
- 至少有一个合作伙伴具有使用 EDI 的实际经验
- 合作各方能够负担自身的通讯设备及通讯软件的费用
- 合作各方都有一批可依靠的、熟悉各自电脑设备和应用系统的技术力量
- 合作各方都有良好的通讯线路和一套可用的、能与 EDI 衔接并尽可能完善的应用系统,同时为 EDI 所需的所有数据已经包含在上述系统中。

如合作的一方没有电脑设备和人员,而另一方能够提供一套包括应用系统在内的 EDI 软件产品,这也是借用 EDI 来推动小公司电脑化的一种捷径。

2. 通讯方案的选择

总的原则是:在技术可行的前提下尽可能降低成本,让用户操作方便,减少人工干预,提高自动化程度。香港地区可供选择的方案比较多,其中包括点到点经 MODEM 利用普通电话线路的通讯,通过专线进入国防数据网的通讯以及利用本地公共数据交换网的通讯等等。点到点的通讯方式比较简单成本低,而网络通讯方式成本高。

CMCL 与 HIT 的通讯是利用 AS/400 的通讯卡和 X.400 通讯软件经专线实时地将数据送到本地的 X.25 网络信箱中,同时收取由 HIT 发出的数据。

CMCL 与 COSCO 各有关代理的通讯是利用 AS/400 的通讯卡和 GE 的通讯软件将数据经本地的 GE 网络结点送到 GE 网在美国的信箱中,同时收取从其它代理发来的数据。

CMCL 与本地各堆场和运费代理的通讯是利用 AS/400 的 ASCII 控制器及普通的电话线,在 PC/SUPPORT

软件的支持下,实现主机与远程数据的自动 DOWN/UP LOAD 交换。由于香港本地电话收费很低,而堆场都使用 PC 软件管理箱动态,故此方案的收效最明显。

3. 代码管理的重要性

推广 EDI 的应用,离不开代码的统一管理问题。EDI 使用的代码一般只供通讯格式中使用,用户的应用系统则采用另外一套代码,两者之间的转化可用一个对照表实现,对照表则由专人维护,并保证对照表中的代码是一一对应的。

一般来说,EDI 的最大受益者是接收方,但在开发 EDI 应用的过程中,实现数据发送比较简单,而实现数据接收比较困难,因为发送方只要从本身的应用系统按通讯格式生成数据并发送出去即可,发送之前的逻辑校验也容易通过。但对处于分布式系统中的各合作伙伴来说,由于时间差或种种原因而造成数据不一致性的问题是无法避免的,因此接收方在接收时必须对代码数据和通讯格式再次进行逻辑检查,以防止将错误数据装入到数据库中。

以单证通讯为例,尽管已投产使用了多年,各种代码也由总公司统一管理,但接收数据时港口代码的不一致性问题依然会发生,从而增加接收处理的工作量,使 EDI 的效益无法得到全面发挥,其原因主要是有关代理的代码维护不及时,由此可见:EDI 的代码维护及管理必须由专人负责,不能交给一般用户去做。

4. 发展 EDI 所需的投资

发展 EDI 的投资一般包括如下几个方面:

- 进网费,如申请安装专线及使用数据网等(一次性)
- 购买通讯卡和 MODEM 等硬件费用(一次性)
- 购买通讯软件费用(一次性)
- 开发或购买转换软件的费用(一次性)
- 测试用费(一次性)
- 专用通讯线路租用费(每月)
- 网络通讯或电话费用,一般按字符数和时间计算,由发方支付。

由于主机的通讯软件比 PC 贵,为了降低成本,在速度、容量与自动化程度许可的情况下,应尽可能考虑以 PC 作为前置通讯处理机。

三、EDI 的进一步发展

CMCL 目前美洲线的单证通讯已全部转用 GE 网,在欧洲线上也与新加坡、汉堡和国内主要外代实现了单证 EDI 通讯,其它航线及尚未实现单证 EDI 通讯的代理还需进一步的推动,同时为配合中远香港协调予配中心的工作,CMCL 在原美洲线 BAYPLAN EDI 的基础上,正组织力量按 BAPLIE 1.5 版本要求开发 EDIFACT 的转换软件,以解决多套通讯格式及代码的转换与维护问题。

另一方面对香港本地的贸易伙伴和政府机构,也要通过宣传加强联系,随着 EDIFACT 技术的发展以及 EDI 用户的增多,在新的 EDI 开发项目中将尽可能采用国际标准。

DOS 使用 EXEC 功能(4B 号功能)来装入或执行外部命令,4B 号功能共有五个子功能,它们分别是:

当 AH=4BH 时:

AL=00 装入且执行程序

AL=01 装入但不执行程序

AL=03 装入覆盖程序(.OVL)

AL=04 装入且在后台执行程序(仅 DOS 4.0 中)

AL=05 进入 EXEC 状态(DOS 5.0 以上版本)

其中 00 号和 03 号子功能早已见诸文档,余下三个子功能均未见公开文档。01 号子功能只装入但不执行程序,

0EH 双字 保存返回的子进程的 SS:SP。

12H 双字 保存返回的子进程的 CS:IP。

调用前由 ES:BX 指向参数块,DS:DX 指向子进程名,返回时若成功则进位标志清零,且 AX 中含有子进程代码第一条指令代码的偏移值,BX 和 DX 被破坏,子进程的堆栈指针填于参数块的 0EH~11H 处,子进程代码指针填于参数块的 12H~15H 处。若失败则 AX 中返回错误代码。在 4B01 执行前应由 4AH 号功能释放当前所有可用内存,以便装入尽可能大的子进程。

4B01 功能也为子进程建立程序段前缀 PSP,其内容大

本文详细分析了 DOS 的 4B01 功能调用的方法,说明了该功能对自覆盖 .EXE 文件加密的步骤,给出了实例。

DOS4B01 功能分析及应用

武汉建筑高等专科学校 廖金祥

用在调试程序如 DEBUG,SYMDEB 等中来装入被调试程序。我们可以利用 01 号子功能这个特性来解决目前软件加密中的一个难题,即具有自覆盖功能的 .EXE 程序或特大的(数百 K 乃至超过基本内存 640K).EXE 程序的加密问题。

一、4B01 功能分析

4B01 功能要求的参数块与 4B00 号功能稍有不同,它在 4B00 功能参数块后面新增了两个双字指针,它们用来存放 4B01 功能调用的返回参数,第一个指针存放被装入程序(此程序称为子进程,下同)的 SS:SP,第二个指针存放 CS:IP。参数块详细格式如下:

偏移 尺寸 说明

00H 字 复制给子进程的环境块的段指针,填于 PSP 的偏移 2CH 处。

02H 双字 复制给子进程的命令行尾指针。

06H 双字 复制给子进程的第一个 FCB 指针,填于 PSP 偏移 5CH 处。

0AH 双字 复制给子进程的第二个 FCB 指针,填于 PSP 偏移 6CH 处。

部分由父进程的 PSP 复制而来,但在其偏移 0AH~0DH 中填入了子进程返回父进程地址,这个地址便是父进程中功能调用指令 INT 21 的下一条指令的地址,即是说子进程的自然出口在父进程中 4B01 功能调用的下一条指令处。

1. 当前 PSP 的切换

执行了 4B01 功能后,控制仍在父进程手中,但当前 PSP 已经指向了子进程,而 DOS 中的当前进程是唯一的,若此时父进程中执行了与文件或设备有关的输入输出操作,就会产生一些麻烦,因为父进程使用了子进程的 PSP,而文件或设备句柄只与当前进程有关,所以应该在 4B01 执行前保存父进程 PSP 段址,执行后保存子进程 PSP 的段址,在 4B01 执行后且控制交给子进程前,利用 DOS 的 50H 号功能将当前 PSP 指向父进程自己,以便于父进程工作,而当需要将控制交给子进程时再将当前 PSP 指向子进程,当从子进程返回时,再将 PSP 指向父进程。

2. 子进程返回地址的重置

由前述可知,在子进程的 PSP 中,出口地址固定指向 4B01 调用的下一条指令处,这并不是我们希望的。在实际

应用中,并不需要 4B01 执行后立即返回父进程,而是要等到子进程执行完毕后再返回到父进程,这就要求我们对子进程 PSP 中偏移 0A~0DH 处的出口地址重新设置。当我们用远跳指令将控制交给子进程时,子进程返回父进程的正确地址是远跳指令的下一条指令处。

3. 子进程的命令行参数

我们也可以让子进程带自己的命令行参数,这些参数是由父进程传递给子进程的。4B01 在执行时为子进程建立了 PSP,且将父进程的 PSP 偏移 80H 处的命令行参数复制到子进程 PSP 的偏移 80H 处,这样子进程便可象其它程序一样接收命令行参数了。这就是说,子进程的命令行参数可以当作父进程的命令行参数键入。

二、4B01 功能在加密中的应用

在软件加密中,最难解决的问题之一是自覆盖的。EXE 文件的加密。这类程序往往较大,甚至可以超过 640K,它由文件头、映象及覆盖部分组成。DOS 在加载它们时,先装入映象部分,在映象部分执行时逐步装入覆盖部分。此类程序的覆盖部分并不另存于 OVL 中,而是文件本身的一部分,这就是自覆盖。

1. 自覆盖程序加密的难点

由于通用加密工具将加密代码部分添加在被加密程序尾部,致使 DOS 企图将映象、覆盖部分及添加的加密代码一同装入内存,这往往造成内存不够用,另外,即使三者能够一次性装入内存,由于加密工具在加密时误将覆盖部分作为映象一起作了加密变换,致使映象装入覆盖部分时,由于映象并不负责对覆盖部分作解密变换,故此时的覆盖部分已面目全非,无法执行。一个最佳的方案是将加密代码插入映象和覆盖部分之间,使 DOS 顺利地装入映象和加密代码,由加密代码对映象解密,再由映象装入覆盖部分。但这谈何容易,一则通用的加密工具很难确定映象部分尾部的位置,二则由于加密代码的插入使覆盖部分在文件中的位置发生了改变,映象将无法找到覆盖部分,于是覆盖部分得不到执行。

2. 自覆盖程序的加密

我们可以利用 4B01 功能对自覆盖程序加密,方法是写一个通用的装入程序(称为父进程),由父进程装入被加密程序(子进程)的映象部分,对映象部分中加密的代码段作解密变换,执行映象部分,读入覆盖部分。具体步骤如下:

①产生一个密钥,密钥可以确定,也可以读 KEY 盘(钥匙盘)产生,用此密钥对子进程映象的某段代码进行加密变换。其方法是:读入子进程文件头的 1BH 个字节,将文

件头长度(偏移 08~09H 的值)乘以 10H;再将代码段对于起始段的偏移量(偏移 16H~17H 的值)乘以 10H;将这两个结果相加,再加上代码段的偏移 IP 的值(偏移 14H~15H 的值),其和便是映象部分第一条指令在子进程文件中的位置,有了这个位置,再来确定加密变换的程序段就更好办了。

②为 4B01 调用准备好参数块,保存父进程的 PSP 段址。

③执行 4B01 调用准备好参数块,保存子进程 PSP 段址,将当前 PSP 指向父进程。

④执行较强的反跟踪程序段。

⑤在子进程 PSP 偏移 0A~0DH 处填入子进程的出口地址,产生正确的密钥,对子进程的加密段进行解密变换。

⑥当前 PSP 指向子进程,DS 和 ES 指向 PSP 段,CS:IP 置为参数块中 12~15H 的值,SS:SP 置为参数块中 0E~11H 的值,用远跳指令将控制交给子进程。

⑦当前 PSP 指向父进程,清除内存中子进程的映象,以防从内存中破获子进程的代码。以上第一步见本文所附 JAMI. ASM 清单,为了简便,对子进程映象加密的起始位置假定为第一条指令的位置,密钥假定为 5AH,加密变换的字节数为 100H。其余各步参见 4B01. ASM 清单。

3. 替换中断 21 调用

为了加强加密解密过程的封闭性,使解密者不能用中断驻留的方法截获正确的程序代码,应避免使用 INT 21 指令。DOS 在每个 PSP 的偏移 05~09 处固定设置为一条远调用指令:CALL F01D:FEF0,由此地址的翻转即等同于指令 CALL 0000:00C0,而在 0:00C0~0:00C4 处是一条远跳指令 JMP XXXX:YYYY,XXXX 即是 INT 21 向量的段地址,而 YYYY 与 INT 21 向量的偏移地址有一个差值,通过分析 DOS3.3 以上版本可知,在 DOS4.0 以上版本中这个差值是 25H,在 DOS3.3 中差值是 1AH。在父进程中反跟踪程序段前截获 DOS 版本号,记下相应的差值,在反跟踪程序段后,在 0:00C1 字节中加上这个差值。为了能正确地返回父进程,还要在 0:00C5 字节中填入“RET”指令的指令码 C3H,经过以上处理即可用:

PUSHF

CALL DWORD PTR DS:[06]

代替 INT 21 指令了。

三、使用方法

将 JAMI. ASM 和 4B01. ASM 分别汇编、连接,再将 4B01. EXE 转换成 4B01. COM。被加密的带自覆盖的 EXE 程序(例子中假定为 FILE. EXE)经过 JAMI 进行加密预处理,不能直接运行,必须由 4B01. COM 对它加载运行。

四、进一步讨论

华南计算机公司

地址:广州市沙河太公路京溪桥 电话:(020)7705696

本文中提及对子进程映象某段代码进行加密变换,然而此段代码的确定并非易事,因为此段代码中不能含有未确定的项,即是说不能含有重定位字。由于 4B01 功能装入映象时对其中的重定位字进行了重定位,而对变换后的重定位字进行重定位势必产生错误的结果。我们知道,文件头的偏移 18H~19H 指出了重定位表在文件中起始位置,06H~07H 指出了重定位表项的个数,每一表项由四个字节组成,前二字节为偏移,后二字节的段值,此四字节合起来指向文件中某个字,此字称为重定位字,该字在映象被装入内存后要被修改,方法是将该字值加上起始段值(起始段值等于 PSP 段值加上 10H),其和放回原处,此过程称为重定位。可以在父进程(4B01)中对重定位字进行处理,比如在加密步骤第五步后插入一程序段,该段程序的流程是:

1. 读入子进程文件头的 1BH 个字节,其中第 08~09H 指出了文件头的长度,再读入整个文件头,在重定位表中找到重定位表项。

2. 将该表项的段值加上子进程 PSP 段值,再加上 10H,再加上该表的偏移,设其和为 A。

3. 将子进程第一指令地址(码段加偏移)加上加密变换的字节数,设其和为 B。

4. 若 $A \leq B$,说明该处的重定位字在加密变换范围内,需要修改,转 6,否则转 5。

5. 一般说来重定位表项是升序存放在重定位表中的,当重定位字地址超过加密范围时,流程结束。

6. 由地址 A 找到重定位字,将其值与密钥作 XOR 运算(假定加密时也是 XOR 运算),结果减去子进程 PSP 段,再减去 10H,其差与密钥作 XOR 运算,结果加上子进程 PSP 段,再加上 10H,其和放回原处。

7. 找到下一重定位表项,转 2。

限于篇幅,在例子 4B01. ASM 中没有给出详细的程序段,有兴趣者可自己解决。

JAMI. ASM 程序清单

```
cseg segment para
    assume cs:cseg,ds:cseg
begin:push cs
    pop ds
    mov dx,offset fname;指向被加密程序名
    mov ax,3d02h;打开文件
    int 21h
    push cs
    pop ds
    jc exit
    mov handle,ax
    mov bx,ax
    mov cx,1bh;读入文件头的 1BH 个字节
    mov dx,offset rwbuff
    mov ah,3fh
    int 21h
    xor dx,dx
```

```
    mov cx,0010h;计算代码段第一条指令条
    mov si,offset rwbuff;在文件中的
    mov ax,[si+08h];位置,并保存
    mul cx;在 CX:DX 中
    push ax
    mov ax,[si+16h]
    mul cx
    pop cx
    clc
    adc ax,cx
    jnc timel
    inc dx
    clc
timel: adc ax,[si+14h]
    jnc time2
    inc dx
    clc
time2: mov cx,dx
    mov dx,ax
    push cx
    push dx
    mov ax,4200h;读写指针移到第一条指令处
    int 21h
    mov ah,3fh
    mov cx,100H;读入 100h 个字节
    mov dx,offset rwbuff
    int 21h
    mov al,5ah;假定密钥为 5AH
    mov bx,offset rwbuff
lop: xor byte ptr [bx],al;加密变换
    inc bx
    loop lop
    pop dx
    pop cx
    mov bx,handle
    mov ax,4200h;移动读写指针
    int 21h
    mov ah,40h;写入加代码段
    mov dx,offset rwbuff
    mov cx,20h
    int 21h
    mov ah,3eh;关闭文件
    int 21h
exit: mov ah,4ch
    int 21h
handle dw 0
fname db 'file.exe',0
rwbuff db 100h dup(0)
cseg ends
end
```

电子工业出版社

广州科技公司

电话: 7588476, 7588494

4B01. ASM 程序清单

```

cseg      segment para
          assume cs:cseg,ds:cseg
          org 100h

start:    jmp begin

parentpsp dw 0
childpsp  dw 0
fname     db 'file.exe',0
kss       dw 0
ksp       dw 0
block     dw 7 dup(0)
newstack  dd 0
newcseg   dd 0
incc      db 0

begin:    mov ax,cs
          mov ds,ax
          mov es,ax
          mov bx,100h      ;释放内存
          mov ah,4ah
          int 21h
          mov di,2ch
          mov bx,offset block ;指向参数块
          mov si,es:[di]
          mov [bx],si
          mov word ptr [bx+2],0080h
          mov [bx+4],es
          mov word ptr [bx+6],005ch
          mov [bx+8],es
          mov word ptr [bx+10],006ch
          mov [bx+12],es
          mov cs:kss,ss      ;保存父进程堆栈
          mov cs:ksp,sp
          mov ah,51h        ;取父进程 PSP 段保存
          int 21h
          mov parentpsp,bx
          mov ah,30h        ;检查 DOS 版本
          int 21h
          xchg ah,al
          cmp ax,031eh      ;是 DOS3.3 以上版本吗?
          jae con0          ;是,转
          jmp exit

con0:     cmp ax,0400h      ;是 DOS4.0 以上版本吗?
          jae con1          ;是,转
          mov cs:incc,1ah   ;DOS3.3 版增量是 1AH
          jmp con

con1:     mov cs:incc,25h   ;DOS4.0 以上版增量 25H
con:      mov dx,offset fname ;DS:DX 指向子进程名
          mov bx,offset block ;ES:BX 指向参数块
          mov ax,4b01h

          int 21h          ;4B01 调用
          jnc con2          ;成功转
          jmp exit

con2:     mov ah,51h        ;保存子进程 PSP
          int 21h
          mov childpsp,bx
          mov bx,parentpsp ;PSP 指向父进程
          mov ah,50h
          int 21h
          retrace: nop      ;此处插入反跟踪程序段
          push ds
          xor ax,ax
          mov ds,ax
          mov al,cs:incc
          add ds:[0c1h],al   ;加上 INT21 偏移的差值
          mov byte ptr ds:[0c5h],0c3h ;'RET' 的指令码
          pop ds
          mov si,offset newcseg ;准备对子进程代码
          mov bx,[si]        ;段解密
          mov ax,[si+2]
          mov ds,ax
          mov cx,100h        ;假定解密字节数是 100H
          mov al,5ah         ;作为例子,假定密钥为 5AH
          lop: xor [bx],al
          inc bx
          loop lop
          mov bx,cs:childpsp ;在子进程 PSP 中填入
          mov ds,bx          ;返回地址
          mov bx,offset return
          mov ds:[0ah],bx
          mov ds:[0ch],cs
          mov bx,offset newstack
          mov ax,cs:[bx]     ;置子进程堆栈
          mov sp,ax
          mov ax,cs:[bx+2]
          mov ss,ax
          mov bx,cs:childpsp ;当前 PSP 指向子进程
          mov ah,50h
          pushf              ;INT 21 调用
          call dword ptr ds:[06h]
          mov ds,bx          ;ES,DS 指向子进程 PSP
          mov es,bx
          jmp dword ptr cs:newcseg ;控制转子进程
return:   mov bx,cs:parentpsp ;返回后当前 PSP 指向父进程
          mov ah,50h
          pushf              ;INT 21 调用
          call dword ptr ds:[06h]
          mov ss,cs:kss
          mov sp,cs:ksp
exit:     mov ah,4ch
          int 21h
cseg      ends
end start

```

北大方正系列产品代理
广州市大恒科技

地址:广州市越秀北路 133 路二楼 电话:3327850

本文着重阐述 NOVELL 公司的网络操作系统 NETWARE 3.11 的功能,软硬件要求,结合本所网络的设计,安装调试及应用作一个概要总结。

网络与通信

NOVELL 网络设计与应用

交通部广州信息技术研究所 饶汉村

近年来,美国 NOVELL 公司以“压倒群芳,一花独秀”之势占有国际局部网络市场的绝对优势,国内使用 NETWARE 网的单位也日益增多,而且也逐渐代替原来的 3+网,IBM 令牌环网等多种 LAN 网络,究其原因是因为 NOVELL 公司发表的 NETWARE 网络操作系统是世界上第一个独立于硬件并真正文件共享的 LAN 网络,由于它采用了开放式协议及其它先进技术,NETWARE 可以运行在 3 COM,ETHERNET,AST PC II,OMMINET,IBM TOKING ESTAR PLAN 2000 等几十种网络上运行,并支持一百余种不同厂家及型号的网板,在 NETWARE 的管理下,不同机种,不同操作系统的工作站可以在同一网络中共享网络资源,通过大型机网的连接,NETWARE 还可以组成复杂的广域混合网络系统。

一、NOVELL 公司网络产品的发展概况及功能

NOVELL 公司自 1985 年推出 NETWARE 286 局部网络操作系统以来,以后陆续推出了 ADVANCENETWARE 286 V2.1X, SFT NETWARE 286 V2.1X, ELS LEVEL IELS LEVEL II,发展到今天更是推出了 NETWARE SFT 386 V3.11, NETWARE 386 V4.0 由于 4.0 版本推出不久,还没见到更详细的资料,本文以 NETWARE V3.11 版本讨论为主。

NETWARE V3.11 是一个全 32

位的网络操作系统,它支持所有重要的台式操作系统(例如:DOS, WINDOWS, OS/2, UNIX, MACINTOSH),以及 IBM SAA 环境, NETWARE V3.11 为众多厂商提供的网络硬件高性能的综合平台。

NETWARE V3.11 提供了集中的网络管理,其版本有 20 用户, 100 用户, 250 用户三种,不管机构范围多大,它都提供了标准的高性能的联网方法。

NETWARE V3.11 支持 OS/2 V1.3, MACINTOSH, NFS UNIX, OSI FTAM, SPX/IPX, NETBIOS, TCP/IP, APPLE TALK OS/2 命名管道,动态组合,工作组管理,NETWARE 远程管理,支持 MCA, ISA, EISA 总线,支持最大磁盘容量是 32TB,文件最大长度是 4GB,可打开文件 100000 个。

NETWARE V3.11 的核心是实时操作系统,操作系统分配内存,允许访问网络文件系统,为系统里的每个部分调度任务,由于是在 32 位的 80386 和 80486 环境下设计的,这个实时操作系统为 NETWARE 的速度和可靠性奠定了基础。任何网络服务基于服务器的应用程序和服务器的实用程序都是模块化的,任何时间都可在服务器不关闭的情况下加载或卸载, NLM 可以通过 C—Library (CLIB) 调用系统功能, C—LIB 是一个开发工具,它应许 NOVELL 公司和第三方的开发人员为 NETWARE V3.11 基于服务器的 NLM 应用程序。NETWARE V3.11 的

模块化结构如下图:

NETWARE V3.11 操作系统分成以下三部分:

1、服务器操作系统是网络的心脏,提供了最基本的网络操作。例如文件管理,内存管理。

2、客户——服务器应用程序,运行在服务器操作系统为用户提供一切服务。例如:实用程序:LOAD, DOWN, EXIT, INSTALL, BIND, MONITOR, UNLOAD 等。

3、工作站连接软件和工作站操作系统。例如:DOS, WINDOWS, OS/2, UNIX 等一起驻留在工作站的内存中,使用这些操作系统的系统调用,应用程序能通过连接软件获得网络服务。

NETWARE 支持的协议如下图所示:

NETWARE 的安全系统和可靠性

NETWARE 为保护网络资源提供了多级安全系统,这些安全系统包括文件级到网络目录级,用户和口令级,用户组级,网络超级用户可以使 NETWARE 安全系统根据安全需要变得复杂或简单,可以限制用户对指定文件,指定目录,指定工作站或指定时间进行操作,口令是加密的,超级用户可以指定用户周期性的改变自己的口令,非法入侵侦测和封锁可以让超级用户指定用户错误登录的次數。

NETWARE 网的可靠性主要表现在 NETWARE 磁盘及文件管理技术上体现,主要在下几方面:

1、Hot Fix 功能

为防止硬盘表面故障引起的读写错误而导致数据丢失,NETWARE 每次向硬盘写完数据后,会随即而读出并和原来的数据比较,若有错;即将该区域做上坏标志,同时将原来写入硬盘的数据重新写入 HotFix 预留的区域,该区域是在安装时硬盘上保留的特定空间,约占总硬盘空间的 2%,以备损坏转向使用,当要读取坏区时,Hot Fix 将其转向相应的预留区域读取。

2. 双重文件定位及目录区表

NETWARE 将文件目录及文件定位表形成两个副本,分别存放于磁盘的不同磁片中,NETWARE 的自动目录区管理功能发现主目录区损坏时,自动转到后备目录区,并使用 Hot Fix 功能处理被损坏的目录区,从而实现错误的自动恢复。另外在启动服务器自检时,NETWARE 会比较两份目录区及 FAT 表是否相同,若不同将会发出错误提示。

3. 磁盘同步映象

NETWARE 采用双磁盘同步映象技术,由两个同步作读写,当发现某一硬盘出错时,另一台立即自动接管全部作业,而不影响网络的正常服务,更不会丢失数据,损坏的磁盘可卸下修复或更换,然后重新装上继续同步映象功能,保证了网络的容错运行。

4. 磁盘双工容错

NETWARE 允许由两组独立控制器,电源的硬盘子系统组成双工硬盘系统,NETWARE 可对磁盘 I/O 进行检测并可自动校正每个磁区不多于 11 个 BIT 的错误及硬盘控制器的轻微错误,与映象系统一样当某一组硬盘系统出现不可恢复的错误时,另一组硬盘立即全部接管整个网络的操作,由于有两组完全独立的硬盘系统,当有多个读请求时,NETWARE 会按两组硬盘的空闲情况或磁头离要读出区域的距离,将读请求分配给不同的硬盘组,这样两组硬盘就可以并行地读出所需的数据,提高整个系统的效率。

5. 交易跟踪系统(TTS)

NETWARE 的 TTS 可跟踪整个交易的完成情况,对于某一交易由于意外中断而造成的文件破坏,TTS 可将文件恢复为交易前的状态。

6. 恢复被删除文件功能

NETWARE 386 为用户提供了被删除文件的恢复功能,除非用户设定了自动清除已删除文件标志,否则文件被删除后它所占用的硬盘空间并不立即释放,而是直到文件服务器已用完所有空间后,再按文件被删的顺序开始清除已删的文件,在文件被正式清除之前,用户随时可恢复其状态,从而免除了用户的误删文件的危险。

二、NOVELL 网络的硬件要求

NOVELL 网的硬件组成主要有四大基本硬件组成:

1. 文件服务器

2. 工作站或其它智能设备

3. 网络接口卡

4. 电缆连接

文件服务器是 NOVELL 网络系统的主要组成部分,许多网络的性能及操作都和网络服务器有关,网络越大,用户越多,更要选择高性能的微机系统作为文件服务器,使用 NETWARE V3.11 版本,它需要带 4MRAM 最小配置基于 386 的硬件,下面公式可以用来计算文件服务器所需的内存:

对于每一个 DOS 卷,所需内存 $M1 = 0.023 \times \text{卷大小 (M) / 块大小}$

对于每一个带名字空间的卷 $M2 = 0.032 \times \text{卷大小 (M) / 块大小}$

文件服务器总的所需内存 $= M1 + M2 + 2MB$

工作站可以是 286,386 微机或无盘工作站。

网络接口卡要根据你所选则的网络拓扑结构来选择相应的网络接口卡因为 NOVELL 网支持总线型,星型,令牌环型的网络拓扑结构,也可以根据需要各种拓扑结构的综合使用,然而各种结构及使用不同电缆所支持的网络总长度是不同的。

在 ETHERNET 网中:

粗电缆(90 欧姆)的限制是:

干线段的最大数是:5

最大干线长度:500 米

最大网络干线电缆长度:2500 米

连接到一条干线段的最大站数:100 个

发送接收器间最小距离:2.5 米

发送接收器间最大长度:50 米

细电缆(50 欧姆)的限制是:

干线段的最大数是:5

最大干线长度:185 米

最大网络干线电缆长度:925 米

连接到一条干线段的最大站数:30 个

BNC T 连接接收器间最小距离:0.5 米

粗细电缆结合的干线中细电缆的最大长度 L:

$L = (1640 - L1) / 3.28$ 英尺

其中:L1 是要构造的干线段长度

支持的网络接口卡有:NE1000,NE2000,NE3200,NE/2,3C501,3C503,3C523 等及其兼容卡。

在 IBM TOKENING 网中的限制是:

站的最大数是:96 个

8228 部件最数目:12 个

8228 部件和站间最大连接电缆距离:45 米

两个 8228 部件间最大电缆距离:45 米

连接所有 8228 部件的最大连接电缆距离:120 米

8228 部件是一个多路存取部件

支持的网卡有 PC ADAPTER 卡等。

在 PC-NET 网中(星型网)限制是:

站点数目最大数:72个

8路分路器和站点间的最大距离:60米

两个站间的最大距离:600米

支持的网卡有 IBM PC ADAPTER II ARC NET 卡等

三、网络设计和安装

我们考虑了经费和用户的要求,采用 ETHERNET 总线结构,连接多种网络卡,使用 HUB 群集器,网络拓扑结构如下图所示:

采用细电缆,网络总长度约 150 米,16 个工作站,AST PP486 作为服务器内存 4MB 硬盘 210MB,网络接口卡有:二合一的:HT200C,EN3360,3C503 三合一的:HT200CT,EN7016

安装步骤如下:

1、服务器程序的安装

(1)首先在服务器创建一个 DOS 分区(最少 4MB),把必要的网络文件拷贝到 DOS 分区上,即把 NETWARE 3.11 的 OPERATING SYSTEM-1,SYSTEM-2,SYSTEM-3 软盘上的所有文件拷贝到硬盘上,然后转换到包含有 SERVER.EXE 的目录上,运行 SERVER.EXE 程序,键入 SERVER <Enter>出现提示请输入文件服务器名,文件服务器唯一的一个内部网络号。

(2)装入磁盘驱动程序

LOAD ISADISK <Enter>

(3)装入网络管理程序

LOAD NMAGENT <Enter>

(4)装入网络接口卡驱动程序

LOAD <LAN 驱动程序> <Enter>

(5)将网络协议和 LAN 驱动程序连接

BIND IPX TO <LAN 驱动程序> <Enter>

(6)装入安装程序

LOAD INSTALL <Enter>

2、工作站软件的生成

(1)有盘工作站

把 DOS ODI WORKSTATION SERVICES 软盘拷贝到 WSGEN 子目录下,输入下面命令:

C:>WSGEN <Enter>

选择你的工作站所插入的网络接口卡的 LAN 驱动程序,输入网络卡的端口地址,中断号等,生成适合本工作站网络卡的 IPX.COM 程序,生成后,若要启动工作站,则在启动完 DOS 后,打入:

IPX <Enter>

NET3 <Enter>对应 DOS3.0,DOS3.1,DOS3.31

或 NETX <Enter>对应 DOS5.0,DOS6.0

若连网成功,则可以 LOGIN 注册进入服务器管理员分配给你的用户名和口令。

(2)无盘工作站

无盘工作站的驱动程序要建一个远程引导影像文件。首先在网络卡上要插入 DOS 引导程序的 BIOS EPROM 芯片,确认你的 DOS 版本,在一台有工作站上,以 SUPERVISOR 用户的身份进行注册,将远程工作站的引导软盘插入 A 盘驱动器,做如下步骤:

1)将驱动器 F:映象到 SYS:SYSTEM

MAP F:=SYS:SYSTEM <ENTER>

2)将驱动器 G:映象到 SYS:LOGIN

MAP G:=SYS:LOGIN <Enter>

3)运行 DOSGEN 程序

F:>DOSGEN <Enter>

将 DOS 的三个驱动程序:IBMBIO.COM,IBMDOS.COM,COMMAND.COM 和 IPX.COM,NET3.COM,NETX.COM 拷贝到 NET\DOS.SYS 的盘上。

(4)把 SYS:LOGIN 中的 NET\DOS.SYS 文件标志为可共享的

F:>FLAG NET\DOS.SYS S <Enter>

5)把 SYS:LOGIN 中给远程引导用户授与修改权限

F:>GRANT M TO <用户名> <Enter>

这样无盘工作站就可以连网了。

在服务器 NETWARE 软件安装完成,工作站软件也生成后,可以启动 NOVELL 网络了,若工作站连网不成功,首先要检查网络接口卡地址,中断号等是否正确,再检查电缆连接有无接触不良,终止器是否 50 欧姆,反复检查和分段测试,这样可以查出问题所在,连网也较容易成功。值得一提的是各种兼容机,兼容网络接口卡并不一定能够在 NOVELL 网连网成功,它要求主机板和 IDE 卡比较严格。

四、管理信息系统在网络上的应用

随着计算机应用的日益普及,单台微机的使用已经不能满足企事业单位更高级管理上的需要,单个部门微机的独立使用使数据的重复冗余,不准确性也显得日益突出了,因此必然从分散式微机管理向数据集中式分散管理靠拢,在企事业单位整体信息管理要考虑总体要求,部门的要求,分清数据在哪一层次上共享,设置各层次的数据共享权限,统计和计算信息管理系统的信息量和速度要求,选择适当的数据库软件,选择适当的网络拓扑结构和连接方法,决定服务器的内存,硬盘的大小。

在系统设计时要特别注意数据的共享和排它性,设置多级的保密字,授予用户对数据库的访问权限,数据的安全保护,禁止未被授权者进入系统感用户进行越权访问,设计好数据的后备和恢复程序。

098

万利科技广场

广州市国际路 25 号(中央酒店侧)

网络与通信

高级 UNIX 连网技术讲座

第五讲 TLI(上)

5.1 ISO—OSI 模型

国际标准化组织(ISO)已经提出著名的开放性系统互连参考模型(ISO—OSI)。这一模型在概念上把互连分为七个层次:

- * 应用层
- * 表示层
- * 会话层
- * 传输层
- * 网络层
- * 数据链路层
- * 物理层

并用报文在这些层之间的传递和在这些层中发生的动作来描述一个互连过程。

传输层接口 Transport Layer Interface, 或 TLI 是位于传输层顶部的一个接口。它在会话层与传输层之间。层与层之间通信的 ISO 术语描述信息在两层实体间的经过。较高的一层是用户, 较低的一层叫支撑块。有四类报文传递:

在客户/讲述者一方:

1. 请求: 用户向支撑块请求完成某些任务。

2. 确认: 支撑块向用户提供任务执行的结果。

在服务器/倾听者一方:

1. 信号: 支撑块向用户描述某些发生的结果。

2. 响应: 用户向支撑块响应信号。

例如, 建立传输层用户 A 与 Z 间点对点连接的过程可以描写如下:

1. 用户 A 请求支撑块 a 与用户 Z 建立连接。

A. 支撑块 a 按请求建立连接。

B. 支撑块 z 作为用户 Z 的支

撑块。

2. 支撑块 z 向用户 Z 发信号: 用户 A 希望连接。

3. 用户 Z 响应支撑块 z: 连接将被接受。

A. 支撑块 z 按请求建立与支撑块 a 的连接。

4. 支撑块 z 向用户 Z 确认用户 Z 已连接上。

用标准的术语来说, 一旦 A 和 Z 连接上, 它们就是用户层的通信进程, 支撑块 a 和 z 是传输层的通信进程。

我们将尽量用这些术语介绍相互间的作用。注意一个请求可以引发控制或数据信息在网上的传输, 但同样可能一个支撑块可以在没有联络其通信进程的情况下确认一个请求。一个支撑块发出的信号通常是由支撑块通信进程发起的, 但同样也可由本地支撑块发起。

5.2 传输层接口

对于点对点(主机对主机)通信有兴趣的人来说, 传输层也许是最有兴趣的, 而且是最重要的, 它是从底层算起保证可靠的通信的第一层; 从顶层算起需要考虑两端动作发生的第一层。ISO 开发了传输层标准(通常称为 ISO/TPn, 这里 n 可以是 0, 1, 2, 3, 或 4, 代表标准指定的服务类型)。其它著名的标准有 TCP/IP 协议组。如 X. 25 的各种公共标准; 如 AT&T Starlan 传输层支持块, 各种 Appletalk 协议, 或 Microsoft Server Message Block (SMB) 协议。

受 ISO 模型的影响, AT&T 建立了 UNIX 系统的传输层接口 Transport Level Interface (TLI) 作为对任何符合 ISO—OSI 标准的传输层支撑块的编程接口。UNIX 系统 TLI

的目的是让较高的层块, 特别是应用层的开发和实现能独立于下一层支撑块的细节。这样我们可以期望符合 TLI 的编码可在任何支撑块上运行, 只需把 TLI 库与支撑块作适当的连接。进一步, 如果是由 UNIX V Release 3.2 以上版本提供的基于标准 STREAMS 的 TLI 库, 符合 TLI 的编码可用任何基于 STREAMS 的支撑块运行而无需重新连接。这样我们就在支撑块上执行看起来是普通 UNIX 的 open 系统调用。

一个为 TLI 子程序库而准备的支撑块所必须满足的要求是很小的。有一份叫传输层支撑块规范的文件说明这些要求。如果支撑块满足这些规范, 程序员就可省去与支撑块细节的对话。当需要这些细节时, 如要求一个面向报文而不是面向字节流的接口时, TLI 接口提供来自支撑块的必要信息的确和所需选项的设定。

TLI 用户仔细了解这些符号是很重要的。TLI 是一个面向传输层支撑块的接口, 并非传输层本身。TLI 用户指定使用传输层支撑块, 支撑块可以是运行在任何支持 IP 网络的 TCP/IP 协议组, 可以是运行在 Starlan 局部网上的 AT&T Universal Receiver Protocol, 或象 AT&T Network Service Library 那样的 STREAMS 库, stream 的库模块可被上推。

5.3 认识支撑块

虽然我们已指出 TLI 的编码是独立于支撑块的, 但这并不排除我们利用支撑块的一些具体特性。为保持用户与支撑块界面的统一, TLI 提供了适当的数据结构 (TINFO) 和子程序 (t—getinfo)。这一机制的原理是一个 TLI 程序可以通过所提供的参数

适当地组织其活动,以至单一个程序就能自身适应大范围的传输层。

TINFO 数据结构包含了所有 TLI 给程序员的有关支撑块的信息:

```
{ addr; /* allowed size of the transport address */
  options; /* allowed size of the options field */
  tsdu; /* allowed size of a transport service data unit */
  etsdu;
  /* allowed size of an expedited transport servicedata unit */
  connect; /* allowed size of a connect request */
  discon; /* allowed size of a disconnect request */
  servtype; /* type of service provided */
};
```

请求这些信息的原语是 t—getinfo:

```
int
t—getinfo(tep, infores—p)
TINFO *infores—p;
```

通常结果结构 infores—p 的域带有定义所示的长度限制。但有可能有一个域不适合于当前所用的支撑块,或者是根本没有限制。表 5—1 是解释。

infores—p 的连接/断接长度域表示是否数据可以伴随连接/断接请求;这对告诉一个通信进程请求的原因是有用的。这里所用的实际值取决于所实现的系统。

服务类型表示支撑块是只对于端对端的数据报还是提供真正的传输连接。在后一情况下是提供延缓的还是非延缓的连接。

服务类型的代码有:

- * T—COTS: 数据流传输, 非延缓的断接。
- * T—COTS—ORD: 数据流传输, 延缓的断接。
- * T—CLTS: 数据报传输(也叫非连接的)。

表 5—1. TINFO 结构值

变量	数值	含义
	>=0	此域值的最大长度
addr	-1	没有最大长度
	-2	地址长度不能由用户设置
	>=0	此域值的最大长度
options	-1	没有最大长度
	-2	选项不能由用户设置
	0	不支持 tsdu
tsdu	>0	此域值的最大长度
	-1	没有最大长度
	-2	此支撑块并不传输正常的数
	0	据不支持 etsdu
	>0	此域值的最大长度
etsdu	-1	没有最大长度
	-2	此支撑块并不传输期待的数据
	>=0	此域值的最大长度
connect	-1	没有最大长度
	-2	连接不能携带数据
	>=0	此域值的最大长度
disconnect	-1	没有最大长度
	-2	断接不能携带数据

服务类型的变体由打开不同的设备来提供。例如, 基于

AT& T3B1/3B2Starlan 的 TLI 有/dev/starlan 提供数据流传输, /dev/starlandg 提供数据报传输; 基于 AT& T3B15 TWG/TCP/IP 的 TLI 有/dev/tcp 提供数据流传输, /dev/udp 提供数据报传输; 基于 Micom/Interlan TCP/IP UNIX V. 3 有提供数据流传输的/dev/it 和提供数据报的/dev/is; 基于 UNIX386 Excelan EXOS TCP/IP 网络服务的/dev/xt 和/dev/xttd。

表 5—2 给出 T—INFO 代表的值。

表 5—2. TINFO 服务值

TINFO	AT& T3B15 WIN/TCP	AT& T3B2 STARLAN	AT& T3B1 STARLAN
元素	/dev /tcp	dev /dev /starlan	dev /dev /starlandg
addr	16	16	255
option	-2	-2	32
tsdu	0	0	1462
etsdu	1	-2	-2
connect	-2	-2	128
discon	-2	-2	128
stvttype	2	3	1

本讲结尾有一个 TLI 程序的例子(showtli.c)。它从一个实际的传输层支撑块提取适当的信息。

5.4 支撑块的选项

由于 TLI 代码要对多种系统具有可移植性并且独立于其支撑块, TLI 要提供在运行时适应支撑块能力的原语就不足为奇了。在 TLI 有一个服务原语 t—optmgmt, 和读写 T—OPTMGMT 结构的选项过程:

```
typedef t—optmgmt
{
  NETBUF opt;
  long flags;
}
T—OPTMGMT;
int
t—optmgmt(tep, reqop—p, resop—p)
T—OPTMGMT reqop—p;
T—OPTMGMT resop—p;
```

交接过程如下:

1. reqop—p→flags 指定一个请求,
2. resop—p→flags 指定一个信息或确认。

T—NEGOTIATE 支撑块读 reqop—p→opt, 返回 resop—p→opt。

T—CHECK 支撑块通过把 resop—p→flags 设为 T—SUCCESS/T—FAILURE 表示对选项的支持。

T—DEFAULT 支撑块用 resop—p→opt 返回默认。

作为选项交接过程的例子, AT& TWIN/TCP 1.1 插座对熟悉的 UNIX 的 Berkeley 变体的 sockets 选项的交接过程作了些修改。opt. buf 指向

```
struct
{
  unsigned short name;
  unsigned short val;
```

这里 name 可以是 BSD 插座选项值之一:

SO—DEBUG
SO—REUSEADDR
SO—KEEPALIVE
SO—DONTROUTE
SO—USELOOPBACK
SO—LINGER
SO—DONTLINGER

除非 name 是 SO—LINGER 否则 val 值是没用的。

5.5 状态和事件

TLI 是用传输层端点(transport end—points TEP)描述传输层连接的。一个 TEP 有一个当前状态(用户可使用其值)和一个当前事件(用户也可使用其值)。TEP 的特性完全可用“有限状态自动机”来描述。这样 TLI 规范描述事件如何使 TEP 改变其状态以及 TEP 在任何具体状态下可能发生的事件。程序员用 TLI 子程序请求和响应传输层支撑块状态的改变。表 5—3 列出 TEP 的状态。

表 5—3. TEP 状态

TEP 状态	含 义
T—UNINT	初始状态,在此没有用户
T—UNBND	TEP 有一个 用户,但 TEP 没有名字
T—IDLE	TEP 有名字,但不做事
T—INCON	TEP 正等待连接另一个 TEP
T—OUTCON	TEP 正试图连接另一个 TEP
T—DATA XFER	TEP 正与另一个 TEP 通信
T—OUTREL	TEP 正试图与另一个 TEP 断接
T—INREL	TEP 正等待与另一个 TEP 断接

任何时候 TEP 的状态可在原语 t—getstate 返回的值中找到:

t—getstate();

表 5—4 列出一个 TEP 可能发生的事件。

表 5—4. TEP 事件

TEP 事 件	含 义
T—LISTEN	一个连接信号已到
T—CONNECT	一个连接确认已到
T—DATA	一个数据信号已到
T—EXDATA	一个期待的数据信号已到
T—ORDREL	一个释放信号已到
T—DISCONNECT	一个断接信号已到
T—UDERROR	一个错误出现在数据报
T—ERROR	一个错误出现在支撑块

当前的事件任何时候都可以在 t—look 原语返回的值中找到:

t—look();

5.6 状态的改变(延缓的)

表 5—5 中列出了 12 个直接改变状态的 TLI 原语。

表 5—5. 状态改变原语

TEL 函数	动 作
t—open()	请求一个 TEP
t—bind()	TEP 的命名
t—listen()	等待来自某一 TEP 的连接信号
t—accept()	响应某一 TEP 的连接信号
t—connect()	请求与指定的 TEP 连接
t—snd()	请求把数据发送到其它 TEP
t—rcv()	等待来自其它 TEP 的数据信号
t—sndrel()	请求与其它的 TEP 断接
t—rcvrel()	等待来自其它 TEP 的断接信号。
t—unbind()	放弃 TEP 名字
t—close()	放弃 TEP

表 5—6 的状态表说明事件如何使状态改变。事件在左边,状态在顶部,表中的条目是当 TEP 处在该列顶部的状态而遇到事件发生时的下一个状态。如果在一系列上没有条目,与该列状态对应的事件将不会发生;这样当 TEP 处在这些状态时程序员就不应发出相应的函数请求。

表 5—6 状态的改变如下图:

TLI 函数	T—UNINT	T—UNBND	T—IDLE	T—INCON	T—DATA XFER	T—SNDREL	T—RCVREL
t—open()	T—UNBND						
t—bind()		T—JDEL					
t—listen()			T—INCON				
t—connect()				T—DATA XFER		T—RCVREL	
t—accept()				T—DATA XFER	T—SNDREL		
t—snd()				T—IDLE			
t—rcv()							
t—revrel()							
t—unbind()				T—UNBND			
t—close()				T—UNINT			

彩色显示器场不同步及无光栅故障的检修

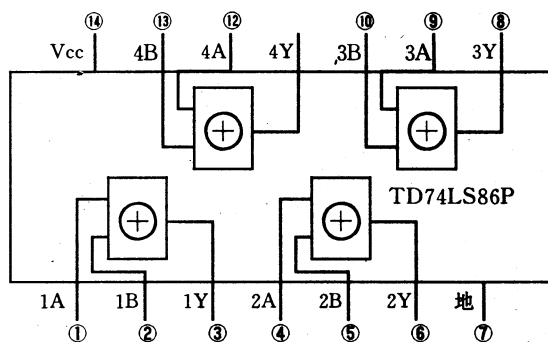
郑州轻工业学院 王一 王阳

1. 故障现象:一台 AST(VGA)彩色 CRT 显示器开机后字符上下滚动。

分析与检修:

显示器通电后,光栅正常,有字符显示,但字符稳不住,向上或向下滚动,这种现象称为场不同步。从故障现象分析,该机电源电路,行扫描电路,显象管电路及视频驱动电路工作基本正常,故障在场扫描电路,字符稳不住与场频有关,当场频高时,字符向下移动,反之,向上移动,该机故障现象即表现为场不同步。场不同步一般有两种原因引起:①场同步信号没有加到场振荡管的基极,引起无法控制同步;②场振荡电路本身出故障,造成场振荡频率偏移,超出了同步范围,而无法控制同步。

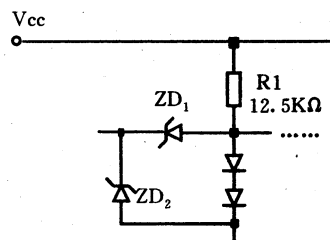
通过以上分析,就可进行检修了,首先检查各元器件有无松动,脱焊等,经检查没有发现异常。接着调节场频电位器来确定故障范围,当调节场频电位器时,字符能够在小范围内停住,但很不稳定,一松开就跑,这表明行振荡级电路基本正常,故障在场同步信号脉冲输入级或积分电路。因无图纸,只好顺着实际线路板进行检查,从计算机主机板输出的场同步信号 V_s 通过信号电缆线送至显示器视频驱动板上,又从视频板上用一根白色导线引至显示器主机板 IC601 的第⑨、⑫脚,经检查各插件无松动现象,排除了因脱焊引起信号丢失的可能性,IC601 型号为 TD74LS86P,经查有关资料得知,该 IC 为 2 输入四异或门电路,其内部电路如图 1,其真值表为图 2。其输入端等效电路为图 3。关机,先试着将同步信号 V_s 导线一端取掉,即人为不送场同步信号,开机故障依旧,这说明原故障确因同步信号没有加至场振荡管基极所致。故障部位在 IC601 或其外围电路,经检查其外围元件 C302、R303、R304 等均正常。又用静态测阻值的方法检查发现:同种机型正常的显示器该集成块⑨



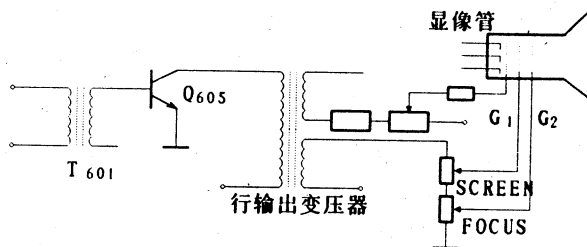
(图 1) $Y = A \oplus B = \bar{A}B + A\bar{B}$

输入		输出
A	B	Y
L	L	L
L	H	H
H	L	H
H	H	L

真值表(图 2)



图(3)



LQ-1600k 打印机检修一例

厦门集美航海学院 陈旭阳

故障现象:一台型号为 LQ-1600K 的打印机加电后初始化正确,没有装入打印纸纸尽灯不亮,在纸选择开关放在单张状态时,进纸操作正常,而在联张状态下,一按“进纸/退纸”键,打印机退纸动作,不能进纸。

分析与维修:在正常情况下,LQ-1600K 打印机加电后所有面板上的灯都亮,然后除电源灯外其他灯都熄灭,字车归原点,如果机器里还没有装好打印纸,纸尽灯点亮的同时蜂鸣器鸣响,如果原来已装好纸则字车归原点,蜂鸣器不鸣叫。在联纸状态时的正常情况下,当机器有打印纸时,一按“进纸/退纸”键,打印机就执行退纸操作;当没有打印纸时,一按“进纸/退纸”键,打印机就执行进纸操作。由于该打印机能正常执行初始化,而且输纸机构能工作,从而说明打印机的驱动电路和控制电路基本正常,故障一般只能出现在纸尽检测通路中。

该打印机的纸尽检测器是一个机械触点式开关,此开

关的一端通过电阻 R66 接+5 伏电源,并通过电阻 R65 接到 μ PD7810【CPU】的 2 脚(\square PA1),另一端通过 CN9 插件接地。当有纸时,开关断开,+5 伏通过电阻 R65 送到 PA1,CPU 确认打印纸装好,纸尽灯不亮,一按“进纸/退纸”键,打印纸退出;当打印纸完全退出后,纸尽检测开关接通,PA1 电平变低,CPU 确认打印机缺纸,纸尽灯亮并使蜂鸣器鸣叫,这时再按“进纸/退纸”键,打印机自动进入到打印位置。

由上述的故障现象和分析,可判断是纸尽检测开关损坏(开路),进行进一步检查。首先关断电源,移去打印机上盖,发现器机内有一层厚厚的灰尘,用吸尘器吸去机内的灰尘,移去牵引输纸器,拔下 CN9 插头,用万用表 $\times 1\Omega$ 档测量 CN9 插头两端,发现不论有无纸欧姆表的阻值总为 ∞ ,说明开关已开路。因此,当机器加电后,PA1 总为高电平,CPU 判断纸装好,纸尽灯不亮,一按“进纸/退纸”键,打印机执行退纸操作【由于在单纸状态下,纸尽检测开关已经断开,因此,在此状态下,进纸正常】。拆下纸尽检测开关,发现开关机械弹片上布满灰尘,用细毛刷打扫干净,再用细沙纸打磨开关的触点(不能用力过度,以防变形),再用干净的布将开关擦干净,用万用表欧姆档测量,开关恢复正常。重新将上纸尽检测开关,插好 CN9 插头,将上牵引输纸器,盖好外壳,加电装纸试机,一切恢复正常。

最后需要说明的是此故障是由于客户在长期使用过程中不注意保养而引起。

101

脚在路对地电阻值为 40Ω ,而该机为 5Ω ,说明该集成块已损坏,通过对其输入端等效电路分析认为可能是 ZD1 击穿或 ZD2 反向电阻值变小导致该 IC 损坏,更换一块新的进口 TD74LS86P,开机,字符显示稳定,场幅可调节,故障排除。

2. 故障现象:一台 AST(VGA)彩色 CRT 显示器,开机后屏幕无显示,电源指示灯亮,荧光屏有高压静电反映。

分析与检修:

引起屏幕无显示的原因有很多,一般大致有以下几个方面:①开关电源电路出故障;②行扫描电路有故障;③显象管供电电路有故障,或显象管损坏。从故障现象上分析,电源指示灯亮,表明电源电路工作基本正常;屏幕上有高压静电反映,则说明行扫描电路正常,没有光栅,则说明显象

管供电电路或显象管损坏,一般情况下显象管损坏的可能性很小。观察显象管灯丝亮,检查各极电压情况,发现加速极电压为 0V,其余电压均正常。由于无加速极电压,使控制阴极发射的电子束到达荧光屏的速度减慢电子束不能打到荧光屏上去,因此无光栅。该加速极电压大小是通过一体化行输出变压器上的电位器来调节的,变压器上标有 SCREEN 的即为加速极电位器,稍微调节一下该电位器,屏幕上立刻出现光栅,检查中发现该电位器为顺时针起始位置,说明被人调动过,经过询问用户得知,用户觉得光栅有点暗,自己调动过,因调范围过大,开机后十几秒钟后,光栅立刻消失,这是由于电压过高,引起显象管保护电路保护,由于不懂电路工作原理,只好又向相反方向旋动,导致无光栅故障。注意:该电位器的调节范围一次不能调节太大,否则会因为电压过高,而出现满屏回扫亮线,或引起显象管保护电路保护自动关机。奉劝计算机用户,显示器与电视机一样,机内有許多可调电位器其作用各不相同,出厂前都已经过厂家用仪器调试好,是不能随意调动的,一旦自己动手调乱,再想恢复就很困难了,而且还会因为调节不当,造成人为损坏机器的可能。切不可无根据地乱调。

100

软件法保护咨询热线

(020)7504151

逢星期五下午 2:30~6:00



一、NetWare Connect 简介

Novell 将各种远程访问方式结合在一起, 形成一个简单而可管理的平台, 网络管理员只需要这种产品就可以支持移动用户、外部通讯和各种连接方式。这就是 NetWare Connect。

移动型计算机的用户正在迅速增长, 其中包括出差在外的专业人士和那些必须离开办公室工作的人们, 他们需要远程访问企业网络, 而 NetWare Connect 正满足他们的这种需求。NetWare Connect 允许多个远程用户拨入网络并访问网络所有的资源(包括文件、数据库、应用程序、电子邮件和小型机服务进程); 也支持网络本地用户连接到广告栏、X. 25 和 ISDN 服务进程以及异地主机。

NetWare Connect 提供下列三种服务方式:

1. AppleTalk 远程访问服务 (ARAS)
2. NASI 连接服务 (NCS)
3. 远程节点服务 (RNS)

RNS 或 ARAS 支持远程的 DOS 工作站、Windows 工作站或 Macintosh 工作站拨入网络并且象本地用户一样访问资源。每一个远程工作站都作为网络的一个节点完全地访问服务进程和应用程序。NCS 支持远程控制中的内部连接。用户可以使用第三方的远程控制程序拨入成为远程控制工作站; 用户服务器可将 NetWare Connect 作为通讯前端, 本地用户可以使用第三方的应用程序拨出。

二、NetWare Connect 的特点

1. 基于 NetWare

NetWare Connect 具有 NetWare 高性能、安全 and 好管理的特点。它可以与文件、打印和其它 NLM 服务在服务器上同时运行。支持 IPX/SPX、AppleTalk 和 TCP/IP 网络协议。适于任何 NetWare 支持的网络拓扑。

2. 通讯资源共享

通讯资源指 MODEM、通讯线路、FAX 服务、异地主机、X. 25 服务和 ISDN 服务。通过 NetWare Connect, 用户在远程拨入或拨出网络时可以共享使用这些资源。通讯设备的共享可以减少设备的需要, 从而减低费用、增加安全性和简化管理。它也允许其它网络服务, 包括 NetWare Global MHS 和第三方的 NLM 共享通讯资源。

3. 远程节点连接

NetWare Connect 支持远程节点连接, 允许用户拨入并作为网络上的节点。允许基于 DOS 和 Windows 的 NetWare 和 LAN WorkPlace 客户机通过 IPX 或 TCP/IP 建立远程节点连接。支持 Macintosh 计算机的 AppleTalk 远程访问 (ARA)。

4. 远程控制连接

NetWare Connect 为远程控制用户提供可得到的远程控制宿主机器的清单。支持第三方远程控制程序, 象 pcANYWHERE、Co / Session、Carbon Copy 和 ReachOut。支持 Novell 和第三方应用服务器软件, 象 NetWare Access Services 和 Citrix System 的 WinView。

5. 支持外部通讯访问

本地用户可以使用异步、X. 25 或 ISDN 连接通过 NetWare Connect 拨出网络, 直接访问远程资源, 包括广告栏服务和异地主机 (DEC、Prime、Data General 和 HP)。在工作站一端, DOS 和 Windows 用户可以运行第三方的客户机程序, 只要它们兼容工业标准 NetWare 异步服务接口 (NASI) 或 INT 14。

6. 广泛的安全和管理功能

NetWare Connect 继承了 NetWare 的安全性优点, 包括登录、密码和审计功能。它也允许对用户或用户组定义权限限制访问通讯资源, 可以限定用户访问特定的资源和连接服务类型。它可以禁止未授权用户入网, 并具有特别的安全回叫功能。它的服务器工具提供端口的配置、处理和管理功能, 帐户跟踪显示并记录端口和用户的状态。NetWare Connect 还支持向 NMS 和 NetView Consoles 传递 SNMP 信息。

7. 基于工业标准

NetWare Connect 采用工业标准的硬件规格, 包括开放数据链路接口 (ODI)、异步输入输出 (AIO) 和广域网接口规格 (WANIS)。它支持异步、X. 25、ISDN 和其它连接类型的第三方通讯适配卡。同时, NetWare Connect 也支持 INT 14 软件接口和 NASI。

8. 灵活的配置

NetWare Connect 包括 2 口、8 口和 32 口三种包装。在一个服务器内可安装多种包装, 最多不超过 64 口。

新一代的远程访问产品

——NetWare Connect

广州蓝深计算机网络系统公司 王健斌

FOXBASE 与 COBOL 语言 相互读取数据的一种方法

人民银行在汇总全辖各专业银行报表的软件设计中最常遇到的一个难题是：有些专业银行的报表程序是用 COBOL 语言编写的，产生的数据存于 COBOL 数据库中，而有些专业银行的报表程序是用 FOXBASE 语言编写的，产生的数据存于 FOXBASE 数据中。笔者在实际设计汇总软件时，找到了一种在 FOXBASE 中读取 COBOL 数据文件中数据及在 COBOL 中读取 FOXBASE 数据库中数据的一种方法，请广大读者指正。

下面以本人编的一组程序举例说明：

假定：在 COBOL 中，有一个数据库 COB. DAT，它在数据部的描述为

```
02  YB-Y1  PIC  9(7)
02  YB-Y2  PIC  X(16)
02  YB-Y3  PIC  X(16)
02  YB-Y4  PIC  X(1)
02  YB-Y5  PIC  9(7)V9
02  YB-Y6  PIC  9(7)V9
02  YB-Y7  PIC  9(7)V9
02  YB-Y8  PIC  9(5)V9
02  YB-Y9  PIC  9(5)V9
02  YB-Y10 PIC  9(5)V9
02  YB-Y11 PIC  9(5)V9
02  YB-Y12 PIC  9(7)V9
```

在 FOXBASE 中，有一个数据库 FOX. DBF，它有 6 个字段，即：

字段名	字段类型	字段长度	小数点位数
DH	N	7	
A1	C	20	
A2	C	20	
A3	N	8	
A4	N	9	1
A5	N	9	1
A6	N	9	1

现在，要分别以 COB. DAT 中的数据项 YB-Y1 与

FOX. DBF 中的字段 DH 为关键字：(1)从 FOX. DBF 中导入 A5、A6 两字段数据库记录到 COB. DAT 中的 YB-Y5 和 YB-Y6 中；(2)从 COB. DAT 中导入两数据项 YB-Y5、YB-Y6 数据到数据库 FOX. DBF 中的 A5、A6 两字段的记录项中。由于在 COBOL 数据文件中，小数点不占位置，因此，有小数点的 FOXBASE 数据库的相应字段长度应比 COBOL 中的相应数据项长度要多一位。程序设计思路与程序清单分别如下：

■ 在 FOXBASE 程序中，从 COB. DAT 中，当 DH 与 YB-Y1 相同时，读取 YB-Y5、YB-Y6 到数据库 FOX. DBF 中的 A5、A6 记录项内，分以下几步进行：

1. 建立中间数据库 AA. DBF 包括 12 个字段，即

字段名	字段类型	字段长度	小数点位数
DH	N	7	
A1	C	16	
A2	C	16	
A3	C	1	
A4	N	8	0
A5	N	8	0
A6	N	8	0
A7	N	8	0
A8	N	8	0
A9	N	8	0
A10	N	8	0
A11	N	8	0
A12	N	8	0

2. 将 COB. DAT 读入 AA. DBF 中。

3. 以 DH 为关键字，将 AA. DBF 中字段 A5、A6 导入 FOX. DBF 中的字段 A5、A6 中。

4. 由于 COBOL 中小数点隐含性，所以最后，还要将导入的数据除以相应小数点位数的 10 的倍数。

程序清单如下：

```
sele 1
use aa
```

```

zap
appe from cob.dat sdf
select 2
use fox
index on dh to ii.ndx
update on dh from a replace a5 with a->a5,a6
with a->a6 rand
repl all a5 with a5/10
repl all a6 with a6/10
close database
retu

```

■ 从 FOXBASE 中的数据 FOX.DBF 读入 DH 与 YB-Y1 相同的字段 A5,A6 中的数据到 COBOL 中的 YB-Y5,YB-Y6 中,步骤为:

1. 先在 FOXBASE 中建立中间数据库 BB.DBF,有三个字段,将 FOX.DBF 中 DH,A5,A6 数据传入 BB.DBF 中。

字段名	字段类型	字段长度	小数点位数
DH	N	7	
A5	N	8	0
A6	C	8	0

2. 建立中间文本文件 DD.TXT,长度为 23,由于 COBOL 数据文件的数值型数据项不认识“.”,若不将小数点事处理,在 COBOL 数据文件中就会出现非 COBOL 数值型数据项符号“>”,在 FOXBASE 程序设计中,要事先处理小数点,笔者用一个 FOXBASE 程序实现上述设想,以得到文本文件 DD.TXT。

3. 设计 COBOL 程序,将 DD.TXT 中的数据读入 COB.DAT 中。

在操作时,先在 FOXBASE 中运行程序 P2.PRG,再将 DD.TXT 文件考入 COBOL 中,运行 COBOL 程序 DR.INT。

FOXBASE 语言中的程序清单如下:

```

***程序名 P2.PRG
sele 1
use fox
sele 2
use bb
index on dh to ii.ndx
update on dh from a replace x1 with a->a5*10 rand
update on dh from a replace x2 with a->a6*10 rand
copy to dd.txt fields dh,x1,x2 sdf
close database
retu

```

COBOL 源程序清单如下:

```

IDENTIFICATION DIVISION
PROGRAM-ID DR.INT
ENVIRONMENT DIVISION

```

INPUT-OUTPUT SECTION

FILE-CONTROL

```

SELECT DD-FILE ASSIGN TO "DD.TXT"
ORGANIZATION LINE SEQUENTIAL
SELECT YB-FILE ASSIGN TO "YBB.DAT"
ORGANIZATION IS INDEXED
ACCESS MODE IS DYNAMIC
RECORD KEY IS SYB-Y1

```

DATA DIVISION

FILE SECTION

FD DD-FILE

01 DD-REC

```

02 DH PIC 9(7)
02 X1 PIC 9(7)V9
02 X2 PIC 9(7)V9

```

FD YB-FILE

01 YB-REC

```

02 SYB-Y1 PIC 9(7)
02 YB-Y2 PIC X(16)
02 YB-Y3 PIC X(16)
02 YB-Y4 PIC X(1)
02 YB-Y5 PIC 9(7)V9
02 YB-Y7 PIC 9(7)V9
02 YB-Y8 PIC 9(5)V9
02 YB-Y9 PIC 9(5)V9
02 YB-Y10 PIC 9(5)V9
02 YB-Y11 PIC 9(5)V9
02 YB-Y12 PIC 9(7)V9

```

WORKING-STORAGE SECTION

```
77 I PIC 9(7)V9
```

```
77 J PIC 9(7)V9
```

PROCEDURE DIVISION

START-RUN

```
OPEN INPUT DD-FILE I-O YB-FILE
```

```
P1
```

```

READ DD-FILE AT END GO TO P2
MOVE DH TO SYB-Y1
READ YB-FILE INVALID KEY PERFORM P1
MOVE X1 TO YB-Y5
MOVE X2 TO YB-Y6
REWRITE YB-REC
GO TO P1P2
CLOSE DD-FILE YB-FILE
STOP RUN

```

本文所列程序在微机 BEIJING 0540 386、DOS3.3 操作系统、单用户 FOXBASE2.0、COBOL LERVER II 上通过运行。

103

为菜单的每一选项提供相应的 HELP

长沙国防科技大学 吴志

在利用 FoxBASE 和 FoxPRO 进行 MIS 设计时,可以设计出形象直观的下拉和光带式混合的功能菜单,如果能在每一项菜单上提供相应的功能、操作和注意事项的说明,那将给操作使用者提供非常方便的自学习手段,使一些用户完全不需培训就可直接上机操作。

本人在多个 MIS 中已实现了:把光带停于任一功能菜单上,按回车键或功能代号则执行相应功能模块,按 F2 键就把该功能模块的全部说明信息以开窗口的形式显示在屏幕上,操作者可以方便地利用方向键阅读所有信息,按 ESC 键关闭 HELP 窗口,回到原菜单选择状态。用户一致反映这一设计很好。下面把该实现方法介绍一下。

做过 MIS 软件设计的人大概都知道,我们可以利用 FoxBASE、FoxPRO 中提供的 ON key=316 do <子程序名> 这一设置语句来实现在 @... get <变量> read 接收语句的执行期间,按 F2 键转去执行<子程序名>所指的子程序;子程序可以完成帮助信息的显示。做到这点并不难,但要实现前面所说的功能则不能利用 ON key... 语句的设置。

我们先编一个开窗口显示帮助说明信息的子程序,假设子程序取名为 MHELP.PRG,该子程序实现的功能和步骤如下:

1. 在调用语句所指定的位置上开一个适当大小的窗口;
2. 打开含有说明信息的文字信息库(文字信息库中应带有标明功能菜单号的字段);
3. 查找定位到相应功能菜单的说明信息处,并显示它们;
4. 利用方向和翻页键控制所显示信息上下移动以便阅读;
5. 按 ESC 键恢复调用本子程序前的状态,即返回。

该子程序的清单比较长,考虑到篇幅,在此不详细列出。有了子程序 MHELP 之后,我们可以用类似下面一段程序来实现:

此程序摘自湖南省高考招生管理系统,实际运行通过。

..... * 程序头,略

DIME MENU0(7,2), MENU1(6), MENU2(6), MENU3(8),
MENU4(11), MENU5(10), MENU6(5), MENU7(7)

* 定义功能菜单所用数组

.....

< 定义各项功能菜单的名称 >

...

BAR1=1

BAR2=1

clear

clear typeahead

set func "F2" to chr(13)+"H" && 设置功能键 F2 为回车和 H 键

DO WHILE .T.

on key

MENU BAR MENU0,7 && 建立七项下拉菜单

MENU 1,MENU1,6,6

MENU 2,MENU2,6,6

MENU 3,MENU3,8,8

MENU 4,MENU4,11,10

MENU 5,MENU5,10,10

MENU 6,MENU6,5,5

MENU 7,MENU7,7,6

READ MENU BAR TO BAR1,BAR2 SAVE

if bar1<=0.or.bar2<=0

exit

endif

BAR12="CJ"+STR(BAR1,1)+LTRIM(STR(BAR2,2))

* 所选功能菜单号

keyboard "E" && 往键盘缓冲区送 E 字符

wls=""

@ 24,0 get wls pict "@"

read

* 读一字符到变量 wls 中

@24,0 say ""

if wls="H" && 若 wls 变量值为 H,则认为按过 F2 键

do mhelp with sys(5)+bar12

* 调用子程序,传递路径和功能菜单号

clear typeahead

else

on key=316 do mhelp with sys(16)

* 用于 @... get ... read 语句的设置

set cons on

DO & BAR12 && 转去执行相应的功能模块

endif

clear

ENDDO

..... && 程序尾,略

利用反汇编模块实现转换键的

IDEA

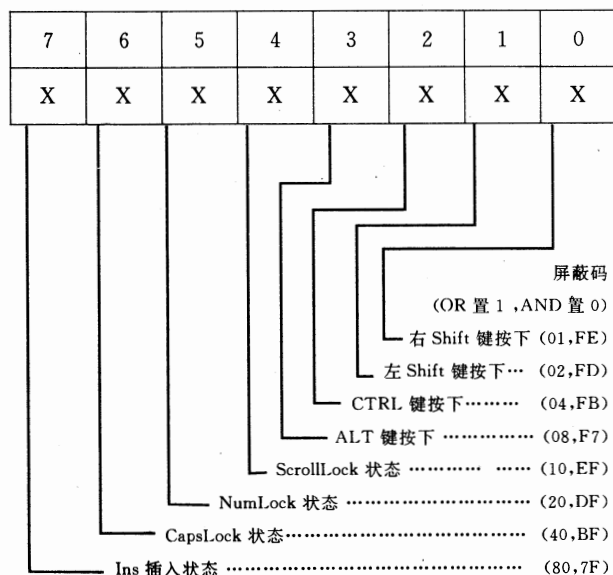


乌鲁木齐海关技术处

熊海鹏

键盘上的 Capslock, Numlock 等转换键在汉字和数字等输入的实际操作中,往往需要动态的转换,我们知道在磁盘操作系统引导过程中,由 ROM BIOS 初始化生成了以段地址 0040H 开始的 ROM 通讯区

ROM 通讯区中包含有键盘输入缓冲区,打印机接口地址,内存容量和转换键状态等重要内容。其中 0040H:0017H 处的 8 位字节就代表转换键的状态,该字节和利用 DOS 中断 INT16H 功能读出的转换键状态相同。利用反汇编修改该字节,就可实现转换键的动态模拟。下面是该字节各位代表的具体含义及屏蔽码:0040:0017H



如果 0040:0017H 字节为 60H(0110 0000),则表示键盘处于 NumLock 数字锁定状态和 CapsLock 字符大写状态。下面介绍一种动态模拟控制转换键 NumLock 的实例:

C>DEBUG

-A100

```
1219:0100  PUSH DS          ;将数据段 DS 堆栈
1219:0101  MOV AX,0040      ;将数据段地址 0040
1219:0104  MOV DS,AX       ;赋予 DS
1219:0106  MOV AH,[0017] ;取转换键内容
1219:010A  MOV BH,AH   ;将转换键内容存于 BH
1219:010C  AND BH,20   ;判断第五位是否为 1?
1219:010F  JZ 0116    ;不为 1,则执行 116
1219:0111  AND AH,DF   ;为 1,则将第五位置 0
1219:0114  JMP 0119    ;执行 119
1219:0116  OR AH,20    ;将第五位置 1
1219:0119  MOV [0017],AH ;改变状态内容
1219:011D  POP DS
1219:011E  INT 20
1219:0120  INT 21
1219:0122
```

-RCX

CX 0022

:22

-NC:\ZHMN.COM

-W

Writing 0022 bytes

-Q

注:程序中 20 和 DF 是第五位的屏蔽码,它控制 NumLock 状态,每执行一次 ZHMN.COM,NumLock 状态就改变一次,其它转换键状态保持不变。这里实现了 NumLock 键的模拟,如果修改程序中相应的屏蔽码,就可实现其它转换键的模拟。

105

买正版软件

中外软件廊

地址:广州流花路 119 号锦汉大厦 11 楼 电话:6689457

自然码

——一种优秀的汉字输入法

湖北省远安县财政局

唐银红



前国内开发的汉字已有数百种，能够高效率地输入汉字的也有几十种，但是大都不能适应于写作或中西文混合录入，当从1993年3期《电脑》杂志上看到北京电影学院刘一兵老师《爬格子动物的新生》一文，使我对自然码汉字输入法产生了莫大的兴趣，因此，从北京超想电脑技术开发公司购买了一套5.0版自然码汉字输入系统，经过一段时间的使用，觉得它用起来确实得心应手，是个值得非专业录入人员试一试的汉字输入法。

准确的说，自然码不是一种汉字系统，而仅仅是一种汉字输入编码方法，它采用了独特的万能悬挂方法，直接挂靠在各种汉字系统上，如2.13H、UCDOS、SPDOS等，成为各种汉字系统上的外挂输入法，因此，自然码只需要在硬盘上安装一次就能实现在各种汉字系统上使用，而其它的汉字输入法，如五笔字型，每种汉字系统都有自己的编码文件和词库文件，硬盘同时安装多种汉字系统时，就占用较大的硬盘空间。采用自然码汉字输入法，对小硬盘（40M以下）用户非常有利，可节约较大的硬盘空间。此外，自然码仅在硬盘上安装一次，其词组库也只有一套，用户在一种汉字系统下定义的词组，在另一种汉字系统下也能使用，给用户带来了方便，而不像五笔字型的词组，在2.13H系统中定义的词组就不能在SPDOS中使用。

正如刘老师文中所述，自然码是一种压缩双拼输入法，非常适用于电脑进行创作的人员，它以词组输入为基础，其内部固化了1万5千多条基本词组，使词组输入简洁、高效、并且其词组的搭配，采用了独特的智能相关的处理方法，它能根据上下文意义，自动从同音词组中挑选出一个合适的词出来，这在拼音输入法中独树一帜。而对于单字，自然码采用声韵义形的输入方式，以及一级、二级简码方式，无论什么汉字，只要按声韵义形方式输足四码，其编码对应的是唯一的单字，不用挑选，从而改善了拼音中同音字的挑选问题，基本上没有重码，使汉字的输入速度大大地提高。

此外，自然系统中的自定义词组，可在汉字的输入过程中即时造词，即不退出汉字编辑软件（WPS、PE）的情况下，直接定义词组，且定义的词组，造完后就可以使用，在有磁盘操作的时候，自动存入到硬盘中。

毋庸置疑，刘老师已对自然码汉字输入方法作了较详细的介绍，但刘老师作为一个文学工作者，所使用的只不过是纯中文的录入，其实，作为科技工作者和电脑爱好者，用自然码来输入科技论文和源程序，较其它汉字输入法更具有优越性。这是因为：一是自然码汉字输入法中的自定义词除了能够实时造词和自动存盘外，其词组编码和词组中各组成要素的编码没有任何联系，即：自定义的词组可以是纯西文字的字符串、各种符号或是与其它编码方式毫不相干的中文词组，例如：可以定义CCTV作为“中央电视台”的输入码，定义fox'作为Foxbase的输入代码，定义xgm'作为符号“Σ”的输入代码，这样，在进行中西文混合录入时，不必频繁地转换大小写状态和进行中、英文方式的转换了。二是标点符号的输入极为方便。一般的中文系统中，只有两种标点符号的方式：半角（英文）和全角字符，半角方式下不能输入中文标点符号，全角方式下，标点符号是中文方式，但字母和数字也变成了全角，这又是我们所不希望的，而自然码中，提供了第三种标点符号方式——中文标点，在此方式下，输入的标点符号为全角字符，是符合中国人习惯的标点符号，而输入的字母、数字又是半角字符，这种方式更符合人们的书写习惯。如果将键盘置为大写状态，输入的字符为半角字符，不必进行全角、半角方式的转换。

当然，自然码汉字输入法并不是完美无缺的，就是速度较别的系统稍慢，如遇到不会读音的汉字，采用其形义方式输入时，其速度完全不能让人忍受，即使在AST PP3微机上，有的字也需要上十分钟才能够确切地找到。然而，瑕不掩玉，我认为自然码的确是一个非常优秀的汉字输入法，特别适合于非专业录入人员的使用，或中西文混合录入。

在

```
void fenhpai (pai)
struct card pai[]:
{ int t=0, k;
do
{ for (k=0; k<4; k++)
    pai[t++].position=k+1;
} while (t<52);
}

void xipai (pai)
struct card pai[]:
{ int i, j;
```


MS-DOS 6.0 下多种系统软件的应用

广州钢铁有限公司计控处 周诚

MS-DOS6.0 以其优越的磁盘和内存管理赢得了广大计算机用户的喜爱。一部计算机上可能配制了多种系统软件,如:金山 DOS,213,网络软件,以及占内存很大的应用软件,如:AUTOCAD,FOXBASE2.10+等,这些软件在使用内存上往往会有冲突。如 MS-DOS6.0 可以把自己的很大一部分放在内存高端,在基本内存中留下约 620K 给用户,使许多软件运行速度大大提高。但是金山 DOS6.0 不允许其使用 EMM386,如果 DOS6.0 不用内存高端的话,运行金山 DOS 后仅给用户留下不到 400K,给运行 FOXBASE 带来很大的困难,大一点的软件将无法运行。

多种系统软件有的需要用扩展内存,有的不能用,用低一点的 DOS 版本,每次转换都需要修改系统配置文件 CONFIG.SYS 然后再启动给用户带来很大的不便。MS-DOS6.0 在这方面有一个突出的特点,可以对系统的配制用菜单进行选择!在 CONFIG 当中新增加了一些命令,如:

```
MENUITEM=CON-BLOCK,CON-TEXT
MENUDEFAULT=CON-BLOCK[.....]
等如在 CONFIG.SYS 中[MENU]
MENUITEM=CON-BLOCK1,CON-TEXT1
MENUITEM=CON-BLOCK2,CON-TEXT2
MENUITEM=CON-BLOCK3,CON-TEXT3
MENUDEFAULT=CON-BLOCK2,3
```

[CON-BLOCK1]

.....

[CON-BLOCK2]

.....

[CON-BLOCK3]

.....

启动后将显示配制菜单:

MS-DOS 6 STARTUP MENU

1 CON-TEXT1

2 CON-TEXT2

3 CON-TEXT3

ENTER A CHOICE:2 TIME REMAINING 03

3 秒钟内不选择的活,缺省为 CON-BLOCK2,系统将

按 CON-BLOCK2 块进行配制同样 在 AUTOEXEC.BAT 中也可以按不同的需要进行选择,加入 GOTO %CONFIG%将按不同的选择分别转向

:CON-BLOCK1

.....

:CON-BLOCK2

.....

:CON-BLOCK3

.....

这样就给多种软件在同一部机上的使用创造了有利条件,同时也为许多计算机知识不是很丰富的同志自动进入自己所需要的系统带来了很大方便,特别值得一提的是 2.13 汉字系统也可以在 MS-DOS6.0 下运行,只在 CONFIG.SYS 中不用 DOS6 的 ANSI.SYS 而选 DOS3.3 的该文件,按本文提供的系统配制文件 CONFIG.SYS 配制后修改 2.13 启动批处理文件,将 FILE1A.COM,INT10F.COM,FILE16B.COM 加载到内存高端,并去掉一些不常用的模块 FILE24A.COM,FILE40A.COM 等,能为常规内存区剩下约 530K 的内存,如同低版本的 DOS 一般!

下面附有在笔者所用的 386/DX 33 上所用的 CONFIG.SYS 和 AUTOEXEC.BAT 以供参考。

CONFIG.SYS

[MENU]

MENUITEM=CAD,CAD

MENUITEM=WPS,WPS

MENUITEM=213,213

MENUDEFAULT=WPS,1

[WPS]

BUFFERS=15,0

FILES=30

LASTDRIVE=I

FCBS=4,0

DEVICE /L:1,5888 =C:\DOS\RAMDRIVE.SYS 3000512256 /E

SHELL=C:\DOS\COMMAND.COM C:\DOS\p

DEVICE=C:\DOS\DBLSPACE.SYS /MOVE

用 Norton 对经过加密

的深层子目录 解密

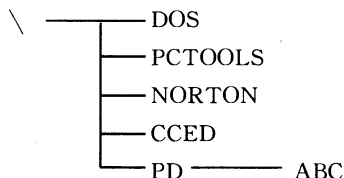
青海水泥厂技术科 李学峰 赵玉霞

《电脑》杂志 93 年第 9 期 IDEA 栏目中,广东的储汉宾发表了《谈 Pctools 对深层子目录的加密》一文,用该方法对深层子目录加密后,用 DOS 的 Dir 命令和 Pctools 软件确实不能发现,起到了保护数据的作用,但对有心人或好事者,用 Norton 软件对目录树进行观察,被加密的子目录便一目了然,而且很容易解密。

下面将我用 Norton 4.5 软件,对用该方法加密后的深层子目录进行解密的步骤提供给各位同仁,以便交流。

- 1、启动 Norton 的 NU 文件,进入主菜单 Main menu;
- 2、选择主菜单中 Explore disk 项,然后选择下级菜单的 Choose item 选项;

3、接着选择 Change directory 选项。此时,屏幕上呈现的目录树中,被加密的子目录同其他子目录一并显示,选择用 Dir 或 Pctools 未曾发现的子目录(该子目录就是加密后的子目录),例如下面目录树中 ABC 子目录。



- 4、按 Esc 键,并选择 Edit/display item,用光标键找到准备解密的子目录(如 ABC),按 F9 键,并用空格键编辑 Hid Dir Vol 为 Dir。

- 5、按 F2 键,将对应的目录登记表中第 32 个字节的 02 改为 00。按 Esc 键后,选择 Write the changed data,并退出 Norton。

至此,解密过程结束。由此可见,该加密方法并非安全可靠,更完美的加密方法有待有一步探讨。

110

```
[CAD]
DEVICE=C:\DOS\HIMEM.SYS
DEVICE=C:\DOS\EMM386.EXE RAM noems
BUFFERS=15,0
FILES=30
DOS=high,UMB
LASTDRIVE=I
FCBS=4,0
DEVICEHIGH /L:1,5888=C:\DOS\RAMDRIVE.SYS 3000 512 256/E
SHELL=C:\DOS\COMMAND.COM C:\DOS\p
DEVICEHIGH=C:\DOS\DBLSPACE.SYS /MOVE
```

```
[213]
DEVICE=C:\213\ANSI.SYS
DEVICE=C:\DOS\HIMEM.SYS
DEVICE=C:\DOS\EMM386.EXE RAM noems
BUFFERS=15,0
FILES=30
DOS=high,UMB
LASTDRIVE=I
FCBS=4,0
DEVICEHIGH /L:1,5888=C:\DOS\RAMDRIVE.SYS 3000512256 /E
SHELL=C:\DOS\COMMAND.COM C:\DOS\p
DEVICE=C:\DOS\DBLSPACE.SYS /MOVE
```

```
AUTOEXEC.BAT
@ECHO OFF
PROMPT $P$G
D:\PCTOOLS\BOOTS SAFE/T14
PATH C:\DOS;C:\;C:\WPS;E:\FOX;C:\213
GOTO %CONFIG%
GOTO ENDA
;WPS
SMARTDRV.EXE
FASTOPEN C:=50
DOSKEY
CALL XS
GOTO ENDA
;213
LH /L:0;1,42384/S C:\DOS\SMARTDRV.EXE
LH /L:1,12928 C:\DOS\FASTOPEN C:=50
LH /L:1,6400 C:\DOS\DOSKEY
CALL 213
GOTO ENDA
;CAD
LH /L:0;1,42384 /S C:\DOS\SMARTDRV.EXE
LH /L:1,12928 C:\DOS\FASTOPEN C:=50
LH /L:1,6400 C:\DOS\DOSKEY
;ENDA
ECHO ON
```

109

键与组合键的扫描码如下表:

各键的扫描码值

键名	单独键	SHIFT +	Ctrl +	Alt +
F1	59	84	94	104
F2	60	85	95	105
F3	61	86	96	106
F4	62	87	97	107
F5	63	88	98	108
F6	64	89	99	109
F7	65	90	100	110
F8	66	91	101	111
F9	67	92	102	112
F10	68	93	103	113

查表方法:先查纵座标后查横座标。

例:F6 键扫描码为 64,shift + F6 的扫描码为 89,
Ctrl+F6 的扫描码为 99 等。

定义功能键举例

例 1:PROMPT \$e F1 键:为 COPY 命令

例 2:PROMPT \$ep F2 键:为 FORMAT 命令

例 3:PROMPT \$eY"p F4 键:为 DISKCOPY 命令

例 4:PROMPT \$e3p F5 键:为 DIR/W 命令

例 5:PROMPT \$e F6 键:为启动 DBASE 命令

例 6:PROMPT \$e F7 键:为启动 EDLIN 命令

例 7:PROMPT \$e F8 键:为启动 debug 命令

例 8:PROMPT \$e.ASM"p

F9 键:为启动 MASM ZH.ASM 命令

例 9:PROMPT \$e"cdps";13;"splib";13;"spdos";13;
13;"wbx"13p

转 C 盘进入 WPS 子目录启动金山 DOS5.10

为了操作方便可以建立一个批处理文件:KEY.BAT,
其内容如下:

PROMPT \$e

PROMPT \$ep

PROMPT \$eY"p

PROMPT \$e3p

PROMPT \$e

PROMPT \$e

PROMPT \$e

PROMPT \$e.ASM"p

PROMPT \$e"cdps";13;"splib";13;"spdos";13;
wbx"13p

三、利用 PROMPT 命令可暂时将失灵的键符用其它键代替使用。

例 1:B 键失灵了,可以用 Z 键代替,方法如下:

C>PROMPT \$e 定义为 B 键

例 2:Q 键失灵了,可以用 A 键代替

C>PROMPT \$e 定义为 A 键

如此类推,其它字母键失灵也可找一个不常用键代替。

四、用 PROMPT 可以改变屏幕的颜色

格式 ESC[#;#m

第一个#为前景码值,第二个#为背景码值,m 必须为小写。

常用屏幕彩色代码

彩色码	前景码	彩色码	背景码
30	黑色	40	黑色
31	红色	41	红色
32	绿色	42	绿色
33	黄色	43	黄色
34	蓝色	44	蓝色
35	紫红色	45	紫红色
36	青蓝色	46	青蓝色
37	白色	47	白色

根据此表选择适当彩色,就能将屏幕颜色设置得非常协调。

例 1:PROMPT \$e 置为黄色,背景为蓝色的显示方式。

例 2:PROMPT \$e 置为红色,背景为绿色的显示方式。

五、几种符号规定

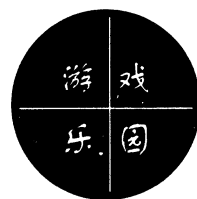
1. ESC:是一个控制代码,其 ASCII 码值是 IBH;
2. ;[:是重新定义的开始标志;
3. P :是表示此命令执行结束,P 必须小写;
4. 13 :是表示回车;
5. 13P:是表示不用手敲回车键,便执行命令;
6. \$e :等价 ESC 控制码;
7. \$- :相当于(Enter)作用。

六、注意问题

1. 上述的特殊用法,在启动的系统盘中的根目录下的 CONFIG.SYS 文件中,必须含有 DEVICE=ANSI.SYS。

2. 命令中的 e,p,m 必须是小写。

(本文的方法在神州 286、LEO386 及 PC 机上通过)。



沧海一声笑.....

《笑傲江湖》攻略

○宋巍○

《笑傲江湖》这本书大家都很熟悉,但是《笑傲江湖》系列电脑游戏软件,恐怕知道的人就不多了。我从朋友处拷来这套游戏,立刻被它的情节所吸引,该软件根据金庸名著改编,剧情逼力十足;超豪华动画,流畅的电影分镜,开创电脑游戏动画新风格;全场语音支援;人物出招时的特殊场景及过关动画等均有丰富数码语音;游戏中有数十首名曲贯穿全场;主题曲:“沧海一声笑”更激荡人心;游戏史上首创“多重动作控制模式”(MULT-ACTIONMODE),令人大开眼界。因而废寝忘食,锲而不舍,终于通关。现将玩这套游戏的一点心得奉献给大家,供大家参考。

《笑傲江湖》是其系列中的第一部,其余两部是《东方不败》与《风云再起》,该软件共 16957254 个字节,经压缩成 7 片三寸盘。

一、操作说明

1、键使用

①用小键盘数字键控制

7—左上 9—右上

1—左下 3—右下

0—跳

②用功能键控制

HOME—左上 PGUP—右上 END—左下

PGDW—右下 INS—跳

2、特殊键使用

F1—取物或询问

F2—物品使用

F10—载入进度、储存进度、退出游戏、语音播放

SPACEBACK—挥剑

3、物品用途

①华山剑谱—基本武功

②大还丹—一次补满内力

③猴儿酒—一次补一半内力

④熊胆丸—三次补满内力

⑤天香断续胶—一次补满生命值

⑥金创药—一次补一半生命值

⑦大补丸—三次补满生命值

⑧曲谱—可以听到主题曲“沧海一声笑”

以上的物品只能补充到 10

4、特殊武功招式(用大键盘数字键控制)

『1』数字键—离剑式 抛出飞剑适合远攻,宜对放飞镖和发气功之敌人,可以在途中破掉飞镖和气功,并射杀敌人,威力中等。

『2』数字键—浪剑式 抛出许多小剑,适合对付群敌,可以防范飞刀和气功的攻击,使敌人不能进身,但威力不够。

『3』数字键—荡剑式 人剑合一,威力强大适合对付强敌,如田伯光、封不平。

『4』数字键—撩剑式 空中劈剑威力最大适合对付超凡高手,如左冷禅、黑白子、岳不群。

(以上四种剑式只能在第二关思过崖学艺后才能使用)

二、过关说明

第一关 金盆洗手、英雄救美、拔刀相助、扶贫救弱。

主人公令狐冲接受师命,参加刘正飞大侠金盆洗手聚会。路上有许多宝箱,须站到箱上用 F1 键拿取。还会遇到发剑和放飞刀的敌人,对付挥剑的只须面对面按下空格键

和他比剑就可将他杀死。放飞刀的敌人比较难斗，必须围着他转，抽冷给一剑，他出刀很快，千万不可面对。第一关比较容易过，但要注意补上生命值。

第二关 违反派规，遭师处罚，面壁思过，名师传功。

令狐冲回归师门，师傅新收弟子林平之，并罚令狐冲思过涯面壁。他巧遇一代宗师风清扬，传他独孤九剑，学会四招：离剑式、浪剑式、荡剑式、撩剑式。回归途中又遇武林淫贼田伯光，注意田伯光刀法凶狠，答完话后迅速转身，拉开距离，转身用三式对付他两次即可打败。回禀师傅后又领命出战封不平，封不平剑法毒辣，宜于远攻，同样用第三式打败他。

第三关 尊师听令，送弟还家，路途艰险，奋不顾身。

令狐冲送师弟回家，必须要走在师弟前面，否则师弟被杀将要自杀谢罪，所以要速战速决。过了树林，家里有钉板，最好把进度储存一下，以免前功尽弃。杀死最后一个发气功的敌人，要在他身上用 F1 取回辟邪剑谱然后才能过关，空箱里装的是猴儿酒。回去的路上，丢失了剑谱，恰巧英雄救美，救了任盈盈。

第四关 丢失剑谱，逐出山门，仗剑江湖，智救教主。

令狐冲被逐出华山派，义不容辞的承担了恒山派掌门之职，与魔教中人殊死搏斗，向问天杀入重围救出令狐冲，为报恩去梅庄救魔教原教主任我行。梅庄机关重重，只可智取，不可力拼。宜在开阔地决战。要注意走视线所能及的墙角，切不可走看不见的墙边，因为那里可能有暗藏的钉板。这一关将面对两名强敌，首先黑白子出场，擅长霹雳掌，威力奇大，并有两名气功高手护驾。一上来必须用第四式先解决掉两个副手，断其左右臂，然后退到开阔地，最好把进度储存一下，以免成果付之东流。对付黑白子宜用威力大的第三式、第四式，内力、生命值注意补充，秘诀其一，把黑白子

引入死角，即用剑劈时，他无处后退，而霹雳掌的威力就无法发挥，打不到你，一鼓作气劈死。其二，和他绕圈子，冷不防用第四式袭击他，打完就跑，消灭他的有生力量。打死黑白子后即可进入梅庄腹地。房屋排布如图 1，各种用品取用，请做好统筹兼顾，武功宜用一、二式。过去后将遇到难以对付的敌人丹青子，擅长武功：旋风指，旋风一出，随人追踪，触身炸裂，威力很大。这时要用对付黑白子的方法，即先除去“两臂”。然后拉开距离绕圈子，把握机会冲入窄巷，转身用第四式——撩剑式劈他，多重复几次（5~6 次）就可送他上西天，救出任我行。屋子里有两宝箱，左一天山断续胶 5、右一大还丹 10。切记杀死强敌后就取不着宝物了。

第五关：五岳争盟，替师拒敌，忽明真象，怒斥乃师。

由于五岳门派争夺盟主，岳不群夺得盟主，左冷禅不服大打出手。令狐冲不计恩怨替师杀敌。左冷禅嵩山派掌门人，擅长火焰掌、柔光指。掌风一起遍地火焰；指风一点，内力光球就会跟踪而来，可用剑劈裂它。火焰掌较难对付，只有拉开距离，用荡剑式、撩剑式远距离攻击。当物品快用完时，可以走到开阔地的右上角，有一条小道，走到尽头可取得多种宝物，然后再回来对付左冷禅。左冷禅倒下，说出一段话，揭开了丢失剑谱之迷，原来是岳不群偷走了剑谱，诬陷徒弟，师徒反目成仇。岳不群华山派掌门、五岳派盟主，武功毒辣，内力生命充沛，擅长霞光剑，剑势一起，金光遍地。对付他有一个窍门，先把他的手下全部杀死，然后走到树林右下角存放宝物的地方，把岳不群引诱过来，劈他一剑，使他背靠宝箱无法后退，然后冲上去用剑劈使他没有机会发出剑式，大约要劈一百多下。要先储存进度再多练几次定能过关。方位见图 2。打败岳不群后，这套软件暂时结束，请接着玩《东方不败》。

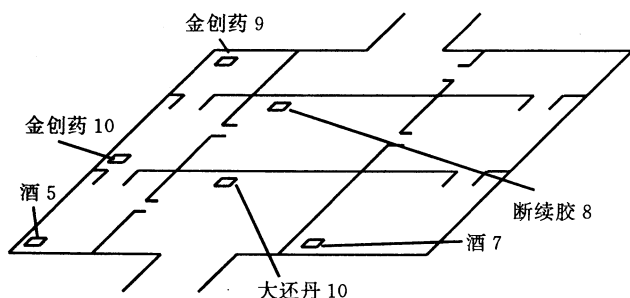
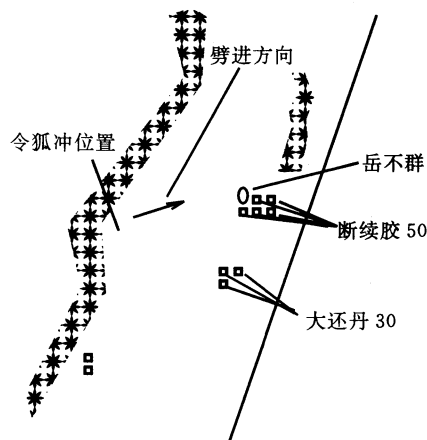


图 1



作者地址：(211451)江苏仪征化纤涤纶三厂聚酯



魔法门外传

存盘数据文件分析

广州华南师范大学计算机九〇级 梁宇珊

欢迎你来到这个几乎完全被魔王统治的大地来,你所背负的任务是十分艰巨的,除了必须洞察“第六面魔镜”的玄机,还得建立一座属于自己的城堡,进而向魔王挑战。

这是一个庞大而杰出的典型西方 PRG 游戏。在魔法门外传的世界里,你将率领六名队员所组成的冒险队伍进行探险。这个游戏是如此的激动人心:庞大而危险难测的野外地形,数不清的怪兽及物品,上百种咒语及魔法物品,几十万种装备,考验你智力的谜题,多不胜数任务,3D 立体的画面,绝不含糊的音效搭配及人物表情……这一切都将为你留下极其难忘的印象。

当你在原野上冒险时,别忘了去探索其它有文明存在的地区(城镇或地下城),在这个世界上有各种不同区域,如洞穴、高塔、神殿等,但队伍最常去的地方应该算是城镇了,在这里你可以找到各种不同的商店,得到各种不同的东西。下面这几个地方对你非常重要:

旅馆:可以重整队伍、买卖干粮、探听消息等。

武器店:可以买卖、鉴别、修复各种装备等。

训练场:可以增加等级(等级影响生命、法力等值)。

公会:可以购买、查看魔法。

神庙:可以医疗或得到祝福。

银行:存放金钱、宝石,可以得到利息。

请多用搜寻功能,即用鼠标在画面上点一点或敲击空格键,你将会有许多意想不到的收获。

如果你决心玩这个游戏,就要准备好花上近一个月或几个月的时间。

然而,当你玩到一半由于一时的不慎而玩不下去时,那将是一件非常痛苦的事情。这时候,你不妨将游戏一些参数作些修改继续你的奋斗。

笔者分析了魔法门外传存盘数据文件 XEEN*.SAV,找到了其中一些存放重要参数的地方,在此列出,以飨读者。你可以利用 PC-TOOLS 的 E 编辑文件功能修改对应的存盘数据文件,将相应的参数改成你所需要的数值。注意!可不要太贪心了哦。

1. 金钱、宝石

两者各点四个字节。一共八个字节,起始位置在该文件逻辑第 26 扇区中第 387 字节(简称为 S26D387,在 PCTOOLS 中也是这样表示的)。

2. 经验

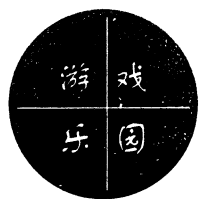
占四个字节,起始位置在(按六个人的顺序,以下同):

S5D229、 S17D457、 S15D65、 S13D27、 S6D71、 S9D305

3. 人物属性及等级

人物属性分为:力量、智慧、人格、耐力、速度、准确、运气七项。其各项及等级各占两个字节。等级存放在人物属性 14 个字节之后,两者之间相隔一个字节(值为 0)。人物属性及等级起始位置在:

S4D413、 S17D129、 S14D249、 S12D211、 S5D255、 S8D489



游戏《组织》的 心得与拆解

广东顺德大良灵智电脑 陈力智

目前的电脑游戏越出越多，新 GAME 如潮水般涌来，SYNDICATE (中文译名：组织、团伙) 是其中的佼佼者。运行 SYNDICATE 需要四兆内存、386 机器、SVGA、10 兆硬盘空间和一个精确的鼠标。

在游戏中，你是一个军事组织的头头，指挥手下在世界各地完成各种各样的任务，如：闯入兵营干掉叛变的上校；在人来人往的城市中杀死恐怖分子，救出人质；在敌国强行登陆利用先进的武器、周详的策略摧毁敌人的军事力量……

在 SYNDICATE 中，动作与谋略并重，人物的动作和武器的运用都令玩家得心应手，并叹为

观止。如配上音效卡，那么各种枪声、爆炸声、撕心裂肺的惨叫声将回荡在您的梦中。

钱在这个 GAME 中至关重要，因您需要大量的金钱来研制各样武器，及将手下逐渐改造成钢铁战士。笔者打爆机后总共耗费了 12 亿元。而您开头却只有 2 万元资金。

您只要将存盘文件 00. GAM 的第 22 个字节改成“10”，那您就会惊喜地发现财产值跃升到 21 亿元，打爆机的机会大了许多。

笔者在电脑公司工作，玩过许多电脑游戏，最欣赏的就是 SYNDICATE，您可千万不要错过啊！

114

4. 法术

每个人都会 39 种法术，在 SAV 文件中，值为 1 表示学会了该项法术，反之为 0。每个人法术都占 39 个字节，起始位置在：

S5D2、 S17D230、 S14D350、 S12D312、 S5D356、 S9D78

但第 2.4 位并非施法者，你可以将其法术后跟的数据串 1C 12 04 00 FF 中的 FF 修改一下使其成为一名施法者。

5. 装备

装备共分武器、护具、配件、杂项四大类，其中以武器种类最多，可达近三十万种。这四大类在 SAV 文件中是连着存放的，每人每类最多可放九样，每样占四个字节，即是说装备每人占 144 字

节。

装备四个字节属性一般为：

加强属性	装备种类	魔法属性	装配方式
------	------	------	------

其中魔法属性对武器有效，装配方式为 0 时表示该物品没有被装配上。

现给出起始地址：

S5D47、 S17D275、 S14D395、 S12D357、 S5D401、S9D123

另外，如果你看惯了版头，想尽快进入游戏的话，请将 XEEN.DAT 改名为 X.EXE，以后运行这个文件就可以了。

113

硬盘写保护防病毒新法

华南理工大学化学工程系

何健明 何启珍

以

往,许多电脑杂志上都介绍过多种硬盘写保护的方法,但都存在着不同程度的缺点:有的必须用软盘启动,有的以 A 盘启动即可进入硬盘进行读写,而且都比较容易解开。本文介绍一种比较完善的硬盘写保护方法,用此法,具有从 A 盘启动不能进入硬盘,从硬盘启动进入写保护状态,拒绝硬盘格式化操作,防止病毒入侵硬盘,和不易被解开等优点。

硬盘写保护,方法多种多样,但归根结底,都是要修改 BIOS 中断 INT13 的写功能,使其对硬盘的写操作无效。根据这一基本思路,很容易把 INT13 的中断向量复原,或根据其中断向量找到中断处理程序进行修改。于是这一基本思路同时也成为了硬盘写保护的致命弱点。本文介绍的硬盘写保护方法虽然同样要修改 INT13,但同时又利用 BIOS 定时器中断 INT1C 设置监视程序。INT1C 为定时器信号中断,在每次系统时钟信号(每秒 18.206481 次)出现时都执行这个中断向量指向的中断处理程序。原 INT1C 为一空中断,中断处理程序中只有一条 IRET 返回指令,现在我们把 INT1C 改成一监视 INT13 中断向量和中断处理程序的监视程序,如中断向量或中断处理程序一旦被修改则立即复完,因而使硬盘写保护更安全地运行。

本方法要修改硬盘的主引导。硬盘主引导位于 0 柱 0 面 1 扇区,包含主引导程序和硬盘分区表。硬盘分区表的四个分区记录从主引导记录偏移 1BEH 处连续存放,每个分区表占 16 个字节。当主机通电,完成硬件设备的自检后,先检查 A 驱动器中是否有系统盘,如有则把其引导扇区(0 面 0 道 1 扇区)读入,然后由引导程序读入系统文件 IBMBIO.COM 和 IBMDOS.COM,并把控制权交给它们,去完成系统初始化过程。如 A 驱动器中没磁盘,便读硬盘主引

导记录到内存 000:7C00H 处,并由引导程序读出活跃分区(分区表中第一字节为 80H)的引导记录,再由分区的引导程序读出系统文件 IBMBIO.COM 和 IBMDOS.COM 完成系统初始化。无论从软盘还是从硬盘启动,系统初始化过程中 IBMBIO.COM 都会读 0 柱 0 面 1 扇区主引导记录的分区表,所以在初始化过程中 0 柱 0 面 1 扇区必须有分区表初始化后才能进入硬盘。为了使从 A 盘启动不能进入硬盘和从硬盘启动能顺利进行,我们可采用交换扇区内容的方法。具体为先把完好的主引导写入 0 柱 0 面 2 扇区(该扇区一般用作主引导备份),然后修改 0 柱 0 面 1 扇区的引导程序,并把该扇区的分区表抹掉。这样,当从软盘启动时便找不到分区表,从而无法进入硬盘。当从硬盘启动时,先把 0 柱 0 面 1 扇区的假主引导读入内存前执行之。执行的结果为:真假主引导交换,即把原来的主引导写回 0 柱 0 面 1 扇区;关键盘中断;把原主引导程序读入内存 0000:7C00H 并执行。由于已把原来的主引导写回 0 柱 0 面 1 扇区,所以在其后的初始化过程中 IBMBIO.COM 便可从 0 柱 0 面 1 扇区中读出分区表,从而顺利启动。系统初始化最后一步是从根目录中找出 AUTOEXEC.BAT 并执行之,于是我们可在硬盘根目录的 AUTOEXEC.BAT 中加入:再交换 0 柱 0 面 1、2 扇区内容的程序 EXCHANGE.COM、写保护程序 PROTECT.COM、监视程序 WATCH.COM 和开键盘中断的程序 OPENKEYB.COM。由于从硬盘启动,一开始时我们已关键盘中断,所以用机者无法用同时按下 CTRL-BREAK 键来终止 AUTOEXEC.BAT 的执行,因此,从硬盘启动后便可顺利进入写保护状态。请注意,EXCHANGE、PROTECT、WATCH、OPENKEYB 四条命令必须在 AUTOEXEC.BAT 中按顺序存放,才能达

神秘的“V888”病毒

广东省湛江市坡头区南油特普公司 唐剑

最近,本单位几台微机在使用过程中遭到一种新的未知病毒的攻击。笔者对其进行了深入的分析,弄清了它的运行机制及其特点,在此介绍给大家,以便及时发现和清除。

1. 此种病毒为文件型病毒,被感染的 COM 文件长度增加 888 个字节,故笔者称其为“V888”病毒,被感染的 EXE 文件长度视原文件长度不同而增加 888~903 个字节。

2. 病毒修改了 INT21H 的入口,使其指向病毒体。当系统中有 4BH(装入并执行)功能调用时,病毒被激活,执行传染模块,若当前目录下找不到相应的 COM 及 EXE 文件或找到但当前磁盘剩余空间不足 4 簇,则转去执行破坏模块。

3. 破坏模块。当系统当前时钟分钟(8 位)的第 4 位不为 1 时,删去当前目录下的“.C”及“.BAK”文件。(事实上,笔者当初发现此种病毒也是因为调试 C 语言程序时,发现源程序突然丢失。)

4. 一般的病毒驻留主要是通过修改 0:0413H 中的值来实现,而 V888 病毒没有采用此种方法,而是通过直接修改 MCB(内存控制块)中的内容来减少系统的可用内存。故 CPAV1.4 及 SCAN106 等均查不出来。只有 PCTOOLS 运行时发现可用内存减少了 2K。

5. 病毒的主体部分都经过加密处理,只有在程序动态执行过程中解密后才能执行。另外病毒修改了 INT24H 中

到写保护的目的。为使大家易于理解,我们把写保护过程分成 EXCHANGE.COM、PROTECT.COM、WATCH.COM 和 OPENKEYB.COM 四部分,读者可将其合成为一个程序缩短运行时间。同时,由于程序短小所以我们用把程序置于中断向量表中空闲部分以实现常驻内存。读者也可改用 INT27 或 DOS 功能调用 31H 使中断处理程序常驻内存。

本方法实现的硬盘写保护运行时和未加写保护硬盘出现的信息完全相同,即拷文件进硬盘显示: Files copied; 删除硬盘文件后,用 DIR 命令亦找不到被删除文件; 格式化硬盘,同样显示一个个柱面的格式化,但由于已进入写保护状态,这些操作全是无效的,硬盘内的文件完好无损,重新启动后用 DIR 命令便又可看到了。

由于有比较完善的写保护措施,所以一般病毒无法入侵硬盘。要想解开写保护,只要运行 JM.COM 即可,但这只能暂时解开写保护,重新启动后,硬盘又会进入写保护状态。本文介绍的硬盘写保护在 PC 机及其兼容机、286、386 等微机上都实现。自在我系计算机室使用此法以来效果良好,

即免除了学生的误操作所造成的文件丢失或硬盘格式化,又可防治病毒侵入(DIR2 病毒除外,此病毒可用 KILL70 软件清除)。但在运行 213H 时因也要读取 0 柱 0 面 1 扇区的内容,所以在运行 213H 时要执行 JM.COM 与 EXCHANGE.COM 把主引导复原。运行后再执行 PROTECT.COM 与 WATCH.COM 即可。只要把 JM.COM, EXCHANGE.COM 与 PROTECT.COM 加入 213H.BAT 就可以很方便地使用 213H 了。

(注:对此法感兴趣者,请与广州五山华南理工大学化学工程系计算机室联系,索取程序,电话:(020)7111884, 邮政编码:510641)

115

KEDA
科达电源
急您所急 想你所想

地址:519000 珠海翠香二路 34 号红海工业楼三楼
电话:(0756)220324 FAX:(0756)231980

断(严重错误处理中断),使其指向一条 IRET 指令,增加了病毒传染过程的隐蔽性。

受到此种病毒攻击的系统,主要症状是许多.C 文件及.BAK 文件莫名其妙的消失,尤其在使用 TURBO C 及 BORLAND C 等集成开发环境中开发调试应用程序时,危害性最大。

病毒的检测并不难,借助 PCTOOLS 和 DEBUG 等工具可如下进行:

1. 用 DIR 命令发现.COM 文件或.EXE 文件增加 888 个字节左右。

2. 用 DEBUG 查看 INT21H 中断向量入口地址为 9FB0:0198。

3. 用 PCTOOLS 发现 DOS 可用内存空间减少 2K。可搜索特征字符串“56 8A 84 75 04 8D BC 4A 04 8D B4 0C 01 30 04 46 3B F7 75 F9 5E C3 33 F6”。

如果具有以上特征,则可肯定系统已感染了 V888 病毒。

对病毒的消除可采用以下方法进行:

一、对于.COM 文件

原.COM 文件的头三个字节(第一条指令)被病毒移到病毒体的最后三个字节。原指令被修改为指向病毒体。

对.COM 文件的解毒只需恢复头三个字节,并减少文件长度即可。如下例:

C:>DEBUG 800.COM

-R

AX=0000 BX=0000 CX=6AC8

-D

XXXX:6BC5 (100+6AC8-3)

XXXX:6BC5 E9 0D 66

-E

XXXX:0100 E9 E9 4D 0D 67 66

-RCX

CX:6AC8

:6750 (6AC8-378)

-W

-Q

二、对于.EXE 文件

EXE 文件的文件头恢复比较麻烦。原文件头的四个数据:堆栈模块相对于加载时的段地址的偏移量;相对装入

模块给出控制时的 SP 值;相对装入模块给出控制时的 IP 值和装入模块中代码段的偏移量。也就是原文件头的 0E~0FH;10~11H;14~15H;16~17H 八个字节的内容。它们分别保存在病毒部分的 7F,90,97,99 处,且其后均对其进行加密处理。加密的关键字存放在病毒体的倒数第三个字节。对其解密时分别取出这些值经解密(与关键字异或)后回填到文件头。

另外,文件头的 02~05 四字节的值(标识文件长度)的修改可采用如下方法进行:

文件长度减去 888 后 MOD 512,若余数为 0,则 02~03H 的值为 200H,04~05H 的值为(文件长度-888-1)/512。否则:02~03H 中的值为(文件长度-888) MOD 512。04~05H 中的值为(文件长度-888)/512+1。具体用手工方法消除该病毒这里就不再一一介绍,下面给出用 BORLAND C++2.0 编写的消毒程序,该程序在 AST 386,联想 486,SUPER 286 上通过,可以消除指定目录下所有.COM 文件及.EXE 文件所感染的病毒。

```
# include <dos.h>;
# include <stdio.h>;
# include <conio.h>;
# include <string.h>;
# include <dir.h>;
# include <io.h>;
# include <fcntl.h>;
void clearmem();
void kill(char curdir[80]);
void dispmess(char *filename);
int killem(struct fblk fblk);
int killexe(struct fblk fblk);
void cursorreset();
union hexword{
    unsigned int ii;
    char ch[2];
};
int filecleaned=0;
int oldrow,oldcol;
long filelen;
char virusmess[24]={ 0x56,0x8a,0x84,0x75,0x04,0x8d,
    0xbc,0x4a,0x04,0x8d,0xb4,0x0c,
    0x01,0x30,0x04,0x46,0x3b,0xf7,
    0x75,0xf9,0x5e,0xc3,0x33,0xf6
};
unsigned int oldint21hoff,oldint21seg;
unsigned int far *int21hpointer,far *newint21hpointer;
union hexword ss,sp,cs,ip,pagesize,bytesize;
void main(int argc, char * argv[])
{
    char buffer[80],olddirectory[80],curdir[80],ss[20],filename[15];
    int olddr,dr;
    union REGS r;
```

```

int21hpointer=(unsigned int far * )MK—FP(0x0000,0x0084);
printf("\n XiaoHe software inc. (C) 1993.11");
printf("\n V888 virus killer!");
printf("\n All rights reserved!");
printf("\n You are welcome to contact with me.");
printf("\n      Tel:(0759)950888—26417");
r.h.ah=0xf0;
int86(0x21,&r,&r);
if(r.h.ah==0xe0){
    if((*(int21hpointer==0x019b)&&
        (*(int21hpointer+1)==0x9fb0)){
        printf(" found virus v888 in your memory!\n");
        printf(" Now cleared!\n");
        clearmem();
    }
}
if (argc==1){
    printf("Usage: [kill888] [drive:]\n");
    exit(1);
}
getcwd(olddirectory,60);
olddr=getdisk();
if (argv[1][1]==' '){
    dr=toupper(argv[1][0])-'A';
    setdisk(dr);
    if(dr!=getdisk()){
        printf("\n Invalid disk driver!");
        exit(1);
    }
    if(argv[1][2]=='\0')
        strcat(argv[1],"\\");
    strcpy(curdir,argv[1]);
}
printf("\n");
kill(curdir);
setdisk(olddr);
chdir(olddirectory);
if(filecleaned==0){
    printf("\n No virus found!");
}
else {
    printf("\n   Altogether %d files cleaned",filecleaned);
}
}
void kill(char curdir[80])
{
    char curd[80],curd2[80],curd3[80];
    struct ffbk ffbk;
    int done;
    strcpy(curd,curdir);
    if(chdir(curd)==-1){
        printf("\n Can not enter into the directory %s",curd);
        exit(-1);
    }
    done=findfirst("*.com",&ffbkl,FA-RDONLY|FA-HIDDEN|
        FA-SYSTEM|FA-ARCH|0);
    while(!done){
        cursorreset();

```

```

        clreol();
        printf("Path: %s  Filename: %s",curd,ffbkl.ff-name);
        killcom(ffblk);
        done=findnext(&ffbkl);
    }
    done=findfirst("*.exe",&ffbkl,FA-RDONLY|FA-HIDDEN|
        FA-SYSTEM|FA-ARCH|0);
    while(!done){
        cursorreset();
        clreol();
        printf("Path: %s  Filename: %s",curd,ffbkl.ff-name);
        killexe(ffblk);
        done=findnext(&ffbkl);
    }
    done=findfirst("*.","&ffbkl,FA-DIREC);
    while(!done){
        if(ffblk.ff-attrib & FA-DIREC){
            if(ffblk.ff-name[0]!='.'){
                strcpy(curd2,curd);
                if(strlen(curd2)!=3)strcat(curd2,"\\");
                strcat(curd2,ffbkl.ff-name);
                kill(curd2);
                chdir(curd);
            }
        }
        done=findnext(&ffbkl);
    }
}
int killcom(struct ffbk ffbkl)
{
    char ch1,ch2,ch3,ch;
    int handle;
    int i=0,succeed=1;
    char buffer[30];
    if((handle=open(ffblk.ff-name,
        O-RDWR|O-BINARY))!=-1){
        lseek(handle,-0x2c,SEEK-END);
        read(handle,buffer,24);
        while((i<24)&& (succeed)){
            if(virusmess[i]!=buffer[i]) succeed=!succeed;
            i++;
        }
        if(!succeed){
            close(handle);
            return (0);
        }
        dispmess(ffblk.ff-name);
        lseek(handle,-3,SEEK-END);
        read(handle,&ch1,1);
        read(handle,&ch2,1);
        read(handle,&ch3,1);
        lseek(handle,0,SEEK-SET);
        write(handle,&ch1,1);
        write(handle,&ch2,1);
        write(handle,&ch3,1);
        ffbkl.ff-fsize-=888;
        chsize(handle,ffbkl.ff-fsize);
        close(handle);
    }
}

```

```

    printf("\n Virus v888 killed!\n");
    return (0);
}
else {
    printf("\n Can not open the file %s", ffbk.ff-name);
    return (-1);
}
}

int killexe(struct ffbk ffbk)
{
    char ch1, ch2, ch3, ch4, key;
    FILE *fp;
    int i=0, succeed=1;
    int handle, pages;
    long iii;
    char buffer[30];
    if((handle=open(ffbk.ff-name,
        O_RDWR|O_BINARY))!=-1)
    {
        iii=ffbkk.ff-fsize;
        lseek(handle, -0x2c, SEEK-END);
        read(handle, buffer, 24);
        while((i<24)&&.(succeed)){
            if(virusmess[i]!=buffer[i]) succeed=!succeed;
            i++;
        }
        if(!succeed){
            close(handle);
            return (0);
        }
        dispmess(ffbk.ff-name);
        lseek(handle, -3, SEEK-END);
        read(handle, &.key, 1);
        lseek(handle, -0x02e8, SEEK-END);
        read(handle, &.sp.ch[0], 1);
        read(handle, &.sp.ch[1], 1);
        sp.ch[0]^=.key;
        sp.ch[1]^=.key;
        lseek(handle, 16, SEEK-SET);
        write(handle, &.sp.ch[0], 1);
        write(handle, &.sp.ch[1], 1);
        lseek(handle, -0x2f9, SEEK-END);
        read(handle, &.ss.ch[0], 1);
        read(handle, &.ss.ch[1], 1);
        ss.ch[0]^=.key;
        ss.ch[1]^=.key;
        lseek(handle, 14, SEEK-SET);
        write(handle, &.ss.ch[0], 1);
        write(handle, &.ss.ch[1], 1);
        lseek(handle, -0x2df, SEEK-END);
        read(handle, &.cs.ch[0], 1);
        read(handle, &.cs.ch[1], 1);
        cs.ch[0]^=.key;
        cs.ch[1]^=.key;
        lseek(handle, 22, SEEK-SET);
        write(handle, &.cs.ch[0], 1);
        write(handle, &.cs.ch[1], 1);
        lseek(handle, -0x2e1, SEEK-END); /* 197 ip */

        read(handle, &.ip.ch[0], 1);
        read(handle, &.ip.ch[1], 1);
        ip.ch[0]^=.key;
        ip.ch[1]^=.key;
        lseek(handle, 20, SEEK-SET);
        write(handle, &.ip.ch[0], 1);
        write(handle, &.ip.ch[1], 1);
        lseek(handle, 2, SEEK-SET);
        read(handle, &.bytesize.ch[0], 1);
        read(handle, &.bytesize.ch[1], 1);
        read(handle, &.pagesize.ch[0], 1);
        read(handle, &.pagesize.ch[1], 1);
        iii-=0x378;
        lseek(handle, 2, SEEK-SET);
        if((pages=iii%512)==0){
            pagesize.ii=iii/512;
            bytesize.ii=0x200;
            write(handle, &.bytesize.ch[0], 2);
            write(handle, &.pagesize.ch[0], 2);
        }
        else{
            pagesize.ii=iii/512+1;
            bytesize.ii=iii%512;
            write(handle, &.bytesize.ch[0], 2);
            write(handle, &.pagesize.ch[0], 2);
        }
        ffbk.ff-fsize-=888;
        chsize(handle, ffbk.ff-fsize);
        close(handle);
        printf("\n Virus v888 killed!\n");
        return(0);
    }
    else {
        printf("\n Can not open the file %s", ffbk.ff-name);
        return (-1);
    }
}

void clearmem()
{
    newint21hpointer=(unsigned int far *)MK-FP(0x9fb0, 0x01f2);
    *int21hpointer++=*newint21hpointer++;
    *int21hpointer=*newint21hpointer;
}

void dispmess(char *filename)
{
    filecleaned++;
    printf("\n The file %s contain v888 virus!", filename);
}

void cursorreset() /* move cursor to begin of line */
{
    int row;
    row=wherey();
    gotoxy(1, row);
}

```

注:本期程序盘中有该程序文本。

一种通用的消毒方法和一个奇妙的解毒工具 RCOPY03

佛山教育学院兆华科技开发公司 熊 焰

当今计算机世界有两类工具软件吸引着许多软件爱好者的关注。一类是不断升级的消、防病毒软件,如国外的 CPAV V1.0、V1.2、V2.0……,SCAN V102、V104、V109 等等。国内的如 KILL V61、V62、V68 等。另一类是层出不穷的具有极高价格/字节比的加密工具如 BITLOCK、LOCK93、中国锁、软件狗等等。本文从另一个角度提出一种新的、通用而统一的消除病毒及解密的方法,该方法已在笔者最新研制成功的通用工具软件 RCOPY03 中得到实现,近半年的试用证明,该方法是十分有效的。

一、染毒及加密的基本原理

众所周知,当我们用常规的 C、PASCAL、汇编等语言编制了一个应用程序之后,最终都将编译成一个可执行的 COM 或 EXE 文件,这种文件的执行流程若非人工精心地“打补丁”,很难修改。这样,当一些病毒或加密工具等对其感染或加密时,只能是先将这种二进制代码进行复杂的变换,然后另外加上一段自己的用于逆变换的程序,这段程序通常是防跟踪的。这样,你要执行原来的程序,必须得先执行这段附加的程序。我们把这段附加的程序称之为“外壳”。对病毒程序而言,这种“外壳”的作用通常在于设置一些驻留中断,以便于往下感染其它的可执行文件,或干脆在运行“外壳”时就感染其它的文件。而对于加密工具所加密的可执行文件来说“外壳”的作用就在判断软、硬

盘上的“指纹”,各种软件狗或加密卡上的“密码”等,待判别通过,再对原来的可执行文件进行逆变换解码,这样才能使原来的可执行文件正常运行。

可见无论是可执行文件染上病毒或是经过加密,其原理基本上相同,即只是对程序包上一层“外壳”。

二、当前消病毒及解密的一般方法

按照传统的方法,当要对一个被感染病毒的文件进行杀毒时,必需先对该病毒程序的“外壳”进行细致地调试、分析,找出其感染及还原文件的规律,然后才能有针对性地编出对应于该病毒的杀毒程序,这也就是说,只有知道一个病毒名(即知道其感染及还原文件的过程),才能消除该病毒。迄今为止,在已推出的形形色色的各种杀毒软件及防毒卡中,尚找不到一种产品能在未知病毒名的情况下消除该病毒。这就导致这样一种普遍的现象,即每隔一段时间,出现了一些新的病毒,则流行的杀毒软件就要更新一个版本。如此此消彼长,不知何时有个尽头。许多用户手上的某个重要软件染上了某种未知名的病毒时,往往自己无法进行分析、消除,只得眼巴巴地等待着新版本的杀毒软件出现后,才能进行杀毒,使用。

同样一个令人感兴趣的领域是加密和解密。如前所述,当今的加密工具软件层出不穷,而且往往售价都很昂贵,几十 K 程序售价即达上千元,这种程序的核心部分,每一个字节都凝聚着作者

的精心谋划,其作用便是防止解密者的跟踪、分析。从这一意义上说,试图通过调试、分析被加密文件的“外壳”来解密文件,远较分析病毒程序困难得多。尽管如此,解密工作仍吸引着许多爱好者的关注。目前,通行的解密方法大多是通过繁琐而细致地逐步分析“外壳”指令来解码,散见于各种报刊上关于如何分析、破解一些著名加密软件“外壳”的文章,充分说明了这一点。这种解密方法,要求解密者具有丰富的系统知识和极大的耐心。笔者曾认识一位爱好者,为了破解 LOCK89 加密程序,足足花费了一个多月的时间,并打印了一大叠程序清单,才获得成功。当然,在这过程中,他也确实学到了不少知识与技巧,为他当前推出的另一个加密工具,打下了基础。

加密和解密这对矛盾的消长,促进着我国加密工具软件的研制、发展。国内一些著名的加密工具软件,设计之精巧,已达到了相当高的水平,并已出口到国外。

三、病毒及加密外壳的缺陷

静态地来看,要将一个被病毒或加密工具包裹了“外壳”的可执行文件解码还原,确实不易(尽管如此,仍然有许多不同的,用于还原被不同“外壳”包裹的工具,如各种杀毒软件、UNLOCK89、UNLOCK93 等等),而要设计一个通用的、可以解码被各种不同“外壳”包裹的工具,似乎更不可能。但动态地来看,则又不然。实际上我们可以注意到,不管“外壳”包得如何严密,它最终在内存中必需老老实实地把原程序自动还原,否则,被“加壳”程序便不能运行。而且从根本上说,被还原的原程序与“外壳”程序之间不存在任何因果联系,“外壳”程序只是人为地包裹上去的。因而只要我们取得了被“外壳”程序自动解码还原的原程序,往后便可以直接使用这种原程序而无须再运行“外壳”程序了,从而达到杀毒或解密的目的。

显然,作为“外壳”程序的作者来说,为了使其“加壳”程序通用,不可能改变被“加壳”程序的结构,也不可能对被“加壳”的程序提出过高的环境要求,否则被“加壳”的病毒便难以流行、传染,被

“加壳”的加密程序运行便受到极大的限制。这就是这类“加壳”程序的固有缺陷,它为我们对该程序“去壳”、还原提供了方便。

四、消病毒及解密的通用方法

前面我们已经谈到,不管“外壳”包得如何严密,其最终必需自动地解除被“加壳”的程序。我们实际上只要在原程序还原以后,从内存中截取到原程序,便可以达到“去壳”的目的。采用这种策略,笔者早年在江西曾推出过 RCOPY02 通用工具软件,并在全国各地得到了广泛的应用。但是,由于 RCOPY02 保存现场的时刻是由人为敲键控制,并且是单纯地保留现场,因而被其保留的现场只能在相同的软、硬件环境下使用,而且由于其把所有内存现场全保留下来,也就把病毒的驻留程序也保留下来了,并不能消除病毒。从而使 RCOPY02 的应用受到了很大的限制。

我们知道 DOS 操作系统为我们提供了丰富的中断功能。一般应用程序必须输入、输出或显示,这些操作通常都是使用 DOS 或 BIOS 的中断来实现。我们只要选择好一个目标程序使用的中断,采用“特洛伊木马”术,由目标程序调用该中断的事件激活,便可以在内存中截获由“外壳”自行解码还原后的原程序。然后,再把这些内存中的二进制代码整理,并建立一个可重定位的 EXE 文件头,便可得到一个可执行的、不含“外壳”的原文件了。这些,就是笔者新近推出的通用工具软件 RCOPY03 的基本工作原理。

RCOPY03 实际上是一个用于将内存中的二进制代码重建为 EXE 文件的通用工具软件。它通过截获原程序使用的中断来对被“加壳”程序进行“去壳”处理。经 RCOPY03“去壳”解码以后的程序,是一个正常的 EXE 文件,其运行不再像 RCOPY02 那样依赖于解码时的软、硬件环境,而可以在和原程序一样的其它任何环境下使用。因而其不仅可用于消除病毒、解密,还可用于解开一些经压缩以后使用的可执行文件。经 RCOPY03 解码以后的程序,可以很方便地进行修改、再加密、压缩等处理。

细心的读者可能注意到,如何选择目标程序使用的中断?如何判别处理目标程序设置并已链接到本身的原中断向量?……等等这些问题都不太好处理,为此,RCOPY03采用了一些新颖的特殊算法,使这些问题都得到了完满的解决。RCOPY03的多级下拉菜单中也提供了观察目标程序对任一中断向量调用序列的手段及选择截获点的方法。有趣的是,某类应用程序对一些中断向量的调用有着固定的序列。例如,由应用广泛的Turbo C v2.0所编译的应用程序中对DOS中断INT 21H的调用,在程序的开始部份就有如下固定的调用序列:

- | | |
|-----------|---------|
| ① AX=3000 | INT 21H |
| ② AX=3500 | INT 21H |
| ③ AX=3504 | INT 21H |
| ④ AX=3505 | INT 21H |
| ⑤ AX=3506 | INT 21H |
| ⑥ AX=2500 | INT 21H |
| ⑦ AH=4A | INT 21H |

其中,第一条中断调用指令,即AX=3000H,取DOS版本号的中断调用,其位置都是固定在IP=000AH的地方,这也是整个原EXE程序的第四条指令,如果通过RCOPY03截取这条指令,所获得解码以后的EXE文件和原程序几乎完全一样。

当然,在RCOPY03中采取这种截获、去壳的解码方法选择截获点时,截获点的选择并不是唯一的。如在上例中,即使选择第六条、第七条中断指令作为截获点,所得到的程序也可以一样执行,因为RCOPY03能够自动处理一些已执行过的中断。这样,RCOPY03就为用户选择截获点提供了极大的自由。

另外,许多病毒及当前国内一些著名的加密工具所包裹的“外壳”程序,对一些中断的调用,也

有着固定的序列。尽管有些“外壳”程序调用中断时并不采用常规的INT指令,其调用序列及调用时的入口参数等,仍然可被RCOPY03一清二楚地记录并显示出来,这为判别“加壳”母程序及区分“壳”程序和被“加壳”的原程序,提供了一个鲜明的依据。

五、结束语

采用上述基本原理研制成的RCOPY03当然还解决并处理了许多技术细节,限于篇幅,此处不能一一详述。RCOPY03研制成功后,笔者精心收集了许多种病毒及市面上流行的几乎所有加密工具对各种目标代码加密的可执行文件试用,证明这种方法是成功的。以较为难处理的加密“外壳”为例,笔者曾用93年推出的某一著名加密工具,对Norton v7.0中的NDD.EXE文件加密(该文件长度近400K),然后让一并不熟悉加、解密计算机爱好者参照说明书试解密,结果头尾不到半小时便解密还原。当然,熟悉加、解密或中断调用的爱好者,用RCOPY03去为一些“加壳”程序“去壳”,更是“弹指一挥间”的事了。

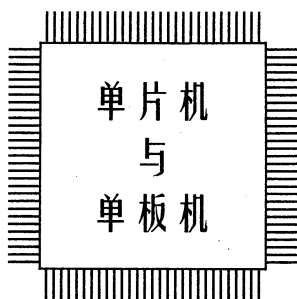
熟谙解密的爱好者可能注意到,有些加密工具还提供了供用户链接到源文件中的OBJ加密模块。这种程序已不单纯地是一个“加壳”程序,不能企图通过RCOPY03一次性地“去壳”解密。但是,由于这种程序在经RCOPY03“去壳”之后,一般不再具有防跟踪的性能(OBJ模块内除外),因而对这种程序,只要先用RCOPY03“去壳”,然后再用调试工具进行调试,找到OBJ模块的调用口,轻易地绕过去即可。这对贯于跟踪、分析那种防跟踪“外壳”的老手,实在是“小菜一碟”。

正像早年RCOPY02推出不久,防RCOPY02的加密软件也出现了一样(这很容易,只要屏蔽INT 5即可),相信此后不久,防RCOPY03的加密软件也必然会出现,这是很自然的事。如此,促进我国加密工具软件的研制,也是作者的最大心愿。当然,如前所述,RCOPY03不仅仅是一个解密工具。

广州袖珍计算机技术服务中心

地址:510080 广州市东风东路745号

电话:(020)7662683 FAX:(020)7758117



单片机与模糊控制讲座

第九讲 用 MC6805R2 控制的电饭锅(上)

广东工学院 余永权

单片机 MC6805R2 控制的电饭锅是一种有一定人类思维的电饭锅,它采用模糊推理进行煮饭的过程控制。这种电饭锅可以进行米量的自动识别和烹调过程的自动调节,并以“煮饭专家”的思想控制全过程,从而煮出来的饭无论从糖化程度,粘度,口感等方面进行量度,都比传统的电饭锅要好。从而适应人们随生活水平提高而对米饭质量要求提高的变化情况。

MC6805R2 控制的模糊电饭锅有可靠性高,结构简单,烹调方式多,控制质量高等一系列优点。从而向人们提供了一种极好的烹调工具。

一、煮饭的工艺过程

从煮饭专家的角度分析煮饭的过程,可以把这个工艺过程分成吸水,加热,沸腾,焖饭,膨胀和保温阶段,如图 1 所示,各个阶段的工作和意义如下:

1、吸水阶段

大米在正常状态的含水量较低,吸水阶段的工作目的就是使大米的含水量升高,以便在加热阶段能够使大米的加热趋于均匀,热量易透到大米的芯部,提高烹煮焖饭阶段的目的就是要使热量能透入到米饭的芯部,使米饭充分受热而内部的质量和外部质量趋于一致。也就是使米芯的淀粉的 α 化,还原糖等和外部即大米表面一样。焖饭阶段还使大米外部的水份一部分深透入米芯,促进内部的成熟变化,另一部分水份蒸发掉。这样,就使到整粒米饭外内一样。

在焖饭阶段,饭锅内部的温度会上升到较高的温度,通常达到 145°C 左右。这样,不但有利于米饭的成熟,而且,使锅底的米饭产生特殊香味。在到达 145°C 左右时,停止加热,由饭锅的余热对米饭进行热焖。

5、膨胀阶段

膨胀阶段是一个使米饭松化的过程。这个过程在焖饭

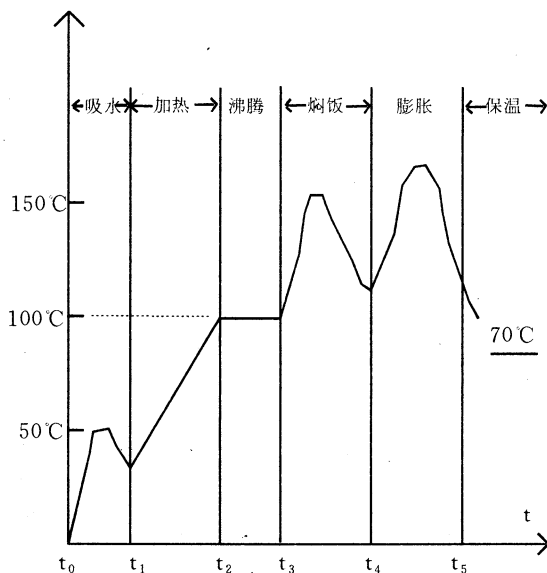


图 1

温度下降到一定程度时,马上又进行加热,这次加热会使米饭的水份进一步蒸发,由于蒸发水份对米饭的作用,使米饭进一步变得松软。加热温度高到一定程度停止,米饭又进行放热。米饭处于这种加放热状态;就可以变得充分松软。

6、保温阶段

保温阶段在饭锅的温度下降到 70°C 时开始执行,通过保持米饭处于恒定的 70°C 温度,就能在 12 小时左右保持米饭的质量完好如新。

传统的电饭锅以磁钢对饭锅进行控制,根本无法完成这个煮饭的工艺过程。用 MC6805R2 单片机控制的电饭

锅,以模糊控制实现煮饭专家的工艺过程控制,故可以煮出质量优良的米饭。

二、单片机控制的逻辑结构

由单片机 MC6805R2 控制的电饭锅的电路原理如图 2 所示。它包括电源、MC6805R2 单片机、检测信号输入电路、功能输入电路、显示电路和加热控制电路等 6 种主要电路组成。在这些电路中,MC6805R2 是系统的控制核心,所有的控制功能都是由 MC6805R2 的控制软件实现的。

1、电源电路

电源采用无变压器电路,直接由稳压管产生稳定 5V 左右的电压。在稳压电路中采用二级并联稳压电路,第一级

号。

过零信号通过对电源进行半波整流后,由晶体管 T_{r1} 放大后产生方波信号而得出。由 T_{r1} 输入到 MC6805R2 的方波信号的上升沿或下降沿就是过零信号。

温度信号是指电饭锅的水温和锅盖温度,温度由检测器 T_{r2} 、 T_{r3} 进行检测。 T_{r2} 、 T_{r3} 是半导体管温敏器件 MST102。由于它检测的温度信号为几百毫伏级的直流电压,为了把其放大成 0—5V 的直流电压,故采用二级运算放大器处理;第一级用于进行阻抗隔离,第一级用于信号放大。

4、功能输入电路

功能输入电路由按键 K_1 — K_3 组成,它们用于输入工作的命令。 K_1 用于启动或停止烹调的控制工作,使电饭锅只处于等待或控制状态。 K_2 用于选择电饭锅的工作方式,工作方式有保温、煮饭、煮粥、煮汤和锅内肉五种。 K_3 用于进行定时信号输入,定时时间为半小时到 3 小时。

5、显示电路

显示电路由 6 个 LED 二极管组成,它有两种显示功能,一种是工作状态显示,一种是定时显示。用 K_2 选择电饭锅工作方式

时,第一次按下 K_2 时,显示原有工作状态,从第二次开始按 K_2 时,则可改变工作方式。用 K_3 决定定时时,第一次按 K_3 时,则可从工作方式转入定时方式显示,第二次后,可改变定时时间;定时时间显示时以一个 LED 亮表示半个小时,6 个 LED 亮则表示 3 个小时。考虑到用水泡米的时间太长,会使到煮饭质量变劣,则和提高米饭质量的希望相违背,故不作太长时间的定时。

6、加热控制电路

加热控制电路由双向晶闸管组成。加热电路有两个加热丝 L_1 、 L_2 ,其中 L_1 是锅底主加热丝, L_2 是锅边副加热丝。 L_1 和 L_2 都由双向晶闸管所控制,双向晶闸管 T_{A1} 、 T_{A2} 采用第 I、II 象限触发的方式;控制采用移相控制方式。

这种逻辑结构有采用元件少,经济性好,可靠性高的特点;并且,它有较良好的控制质量和较齐全的功能。(待续)

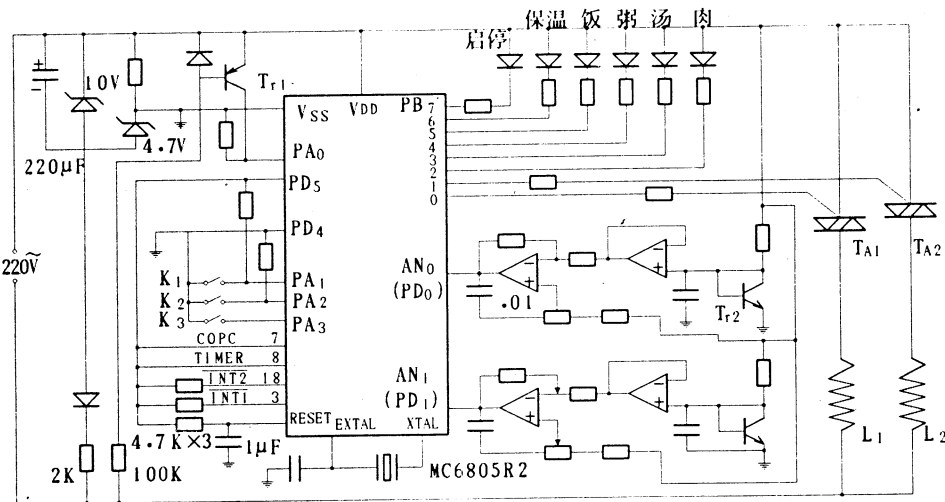


图 2

是 10V 稳压电路,第二级是 5V 稳压电路。

2、MC6805R2 单片机

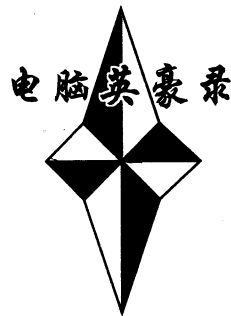
MC6805R2 单片机是控制电饭锅的心脏,它接收来自按键 K_1 — K_3 的状态,过零检测晶体管 T_{r1} 的状态,以及温度检测器 T_{r2} 、 T_{r3} 的状态;然后,决定控制的过程。控制过程的有关状态则送到 PB 口的显示器去进行显示。而控制信号则由 PB₁、PB₀ 去控制双向晶闸管,从而控制加热丝 L_1 、 L_2 的加热状态。整个过程由存储在单片机 MC6805R2 中的软件执行。

3、检测信号输入电路

检测信号有两种,一种是过零信号,另一种是温度信

人算不如电算
EASY IS EASY
拓展财务网络软件 EASY

电话:(020)7501451



“2.13”——吴晓军

当人们打开微电脑让它在中文环境下运行的时候，首先映入眼帘的，往往是“HHBIOS 2.13. 吴晓军”等字样。吴晓军是什么人呢？为什么他制作的 2.13 汉字系统会进入千万家计算机用户呢？吴晓军，是历史不长但已经有相当知名度的北京晓军电脑公司的董事长，是在我国软件园地上耕耘了十多年，并取得了丰硕果实的一名软件科技工作者。他集企业家和软件专家于一身，得力于他敏锐的经济目光和从不停止思索的脑子。

严格地说，吴晓军并不是电脑“科班”出身。他在北京八中毕业之后，正赶上“文化大革命”，被分配到外地去当工人，后来又回北京，当上化工三厂的工人。就在这段时间里，好学不倦的他抓紧时间完成了电大的电子专业课，并以优异的成绩毕业。也在这段学习期间，他第一次接触到计算机。他很快地认识到计算机确实是信息时代人们所不可或缺的应用工具，使用计算机是大势所趋；但要真正使人们用好计算机却仍然存在许多障碍，仅仅要让中国人在计算机上使用汉字便是一大难题。他敏锐地感到在这个领域中，大有“英雄用武之地”。“一定要在这个领域里有所作为”的念头，驱使他下苦工夫去钻研电脑技术，特别是解决汉字运行环境的技术。他边读书边试验，排除其它杂念，一干就是几年。1985 年，经过在 8 位微机上的一些实践经验积累之后，他试着采用 16 位 PC 机，利用电子部 6 所的 CCDOS，闯开自己的路子。他利用 debug 创造了一套汉字系统软件。这套软件的 2.11、2.12 版先后问世，并立即受到人们的注意。到 1986 年，2.13A 版问世了，这是他个人生涯中一个小小的里程碑：2.13A 版具有一级字库进入内存，二级放在硬盘，还具有联想词组功能，并可输出 4 种字体到打印机上，是当时汉字系统环境中比较突出的一种支持软件。2.13 受到用户的欢迎。成为微机汉字系统的代名词，并由此发展成 2.13 系列。当他研制成功可以移植到当时国内流行的长城微机上去的 2.13A 版，就被调到电子部 6 所平安里门市部去搞软件工作，他沿着“2.13”的方向走下去；1987 年开发出 2.13C；88 年，又做成能支持 44 种字体功能的 2.13D 和 E；89 年 9 月，2.13F 正式问世，能把五笔字型接上去，90 年 1 月 2.13H 版推出，功能不断充实，形成了倍受用户喜爱的 2.13 系列。随着 2.13 软件的流行，吴晓军逐渐为人们所熟悉，他的贡献。也到受计算机界人士的肯定。他似乎可以在较顺利的环境下继续开展他的软件研究工作了。就在这个时候，他经过对我国改革开放形势的分析，意识到市场经济在中国未来的重大作用，便当机立断，作出了人生道路上一次重大抉择：抛开铁饭碗，

自己办一个高科技的企业。1989 年 7 月，吴晓军在朋友的帮助下，借用别人的执照帐户，在饭店租了一房间，开始了艰难的创业。经过一年多的周折，他认为自己已经从一个专业的软件开发人员逐步转变为同时也粗懂市场规律的经营管理者，已经可以而且应该直接置身于市场经济的大海洋中去拼搏了。

就这样，他于 90 年 10 月成立了晓军电脑公司，这是他生涯的第二个里程碑。公司成立时只有 15 个人，集资 1 万元。但就凭 2.13 系列的过人技术和良好的服务态度，一年之后，累积了 100 多万元资金。这些钱，反过来支持他后续的研究。到 1991 年，晓军电脑公司研制成 2.13K I 型汉卡，除了原有的功能继续完善外，还能和西文图形方式兼容。2.13K I 型汉卡当年便销售了 400 多套，获利 170 万元。公司的发展进入了良性循环：人员增加了，技术力量加强了。但吴晓军从不停息他的脚步，率领技术人员在 92 年之内连续推出 2.13K I+型，2.13K II 型汉卡和 2.13K 网络汉字系统（他们管这叫 2.13 三兄弟）。这“三兄弟”具有极高的通用性：支持 PC-DOS 和 MS-DOS 各个时期的版本，支持各种分辨率的显示器，有直接写屏功能，西文软件毋需汉化便可支持汉字，支持各种 9 针、24 针、激光、喷墨打印机，并具有办公排版，西文原版 AutoCAD 使用汉字等功能。这“三兄弟”还采用积木式编程，由用户按实际应用情况拼装内存；同时还适应各种网络系统等等。

2.13 系列产品，从 91 年到 93 年，连续三年销售量翻番。产品的畅销给晓军公司带来更大的发展。现在，公司已搬到北京中关村，拥有上百名员工，几百万的资产，他的“2.13 集成办公系统”被评为北京市新技术开发区的“拳头产品”。在技术开发的同时，吴晓军狠抓产品质量和售后服务，用户提出的技术要求，往往在新的产品中得到体现。他认为这是他能不断更新的动力之一。

吴晓军，一个四十多岁的憨厚朴实的中年知识分子，一个在经济大潮中指挥着自己航船的船长，他没有什么豪言壮语，只是紧紧地抓住按科学规律办事，按经济规律办事这一诀窍，他便能把握着自己的命运一步一步迈向他的新目标：把“2.13”继续向网络、通讯、智能等方面深化以更臻美的产品提供给用户，让公司向国际大环境进军。

吴晓军属于中国科技界出现“断层”现象的这代人。但他终于通过自己的努力，向上承继老一辈人的技术，向下带领新一代人去开拓和拼搏。他认为，这正是他这一代科技工作者的历史使命。

编号:0027 “陆码”

本文分析对汉字键盘编码输入方法的需求,归纳出6条规律;论述了作者努力实现这6条规律而开发的一种四维形声码(“陆码”)输入法所做的技术工作,以及达到的效果。

改进汉字输入的探索

广东省电力一局电脑中心 陆国华

当今世界,已进入了信息时代。以计算机为代表的信息处理技术,在我国的四个现代化宏伟事业中,起着十分重要的作用。大量汉字信息输入计算机的问题成了推广使用的关键。随着计算机的普及和日益深入人们的日常生活,解决汉字输入难的问题更加迫切。经过科学工作者多年不懈的努力,这一“瓶颈”问题已经基本得到解决。

汉字输入计算机的途径主要有三条:声音输入、手写、图象输入和键盘输入。这三种途径中,最基本的是键盘输入。虽然声音和手写输入的技术正面临着重大的突破,已经接近实用化的程度,但即使将来技术过关达到实用化,也免不了要用键盘输入来修正个别的错误,许多特定的场合还只能使用键盘输入。因此,键盘输入的方法还要大力完善。

凭着人的聪明才智,设计出的各种编码方案已达到几百种,且商品化、实际运用的微机汉字操作系统也不下几十种,许多优秀的操作员输入汉字的速度已经超过了英文,使外国人惊叹不已。但是真正要找到一种既简单易学,又高速高效的令人满意的普及型输入法,仍很艰难。现在,几百种编码方案,使人眼花缭乱,新的方案仍然不断涌现,大有万“码”奔腾之势。这说明现有的所有方案都还不是最佳、最合理的,汉字是形和声的美妙结合,是元素最少、效率最高、最适合编码的信息载体。我们必须发挥创造力,共同努力,争取在九十年代统一到最合理、最简练、效率最高、最实用的方案上来。

各种输入方案无非是利用汉字的某一类特征,编排在对应的键面上。作者对较为流行的优秀方案进行深入分

析,发现它们都具有一些共性,呈现出一种规律性,正趋向于统一

一、主要的几点归纳如下:

1. 编码代码基元数趋向26个,同汉语拼音、英文、通用键盘一致,键面分布也已有一部分相当一致,尽量做到或形象或声似。
2. 码长趋向四键等长码,四键应该达到无重码。严格地算,三键已有 $26^3=17576$ 种组合,理论上可以包容二级汉字库6768个字无重码。但汉字同音、形似的字相当多,三键重码率要求降至1.5~3%的实用程度,势必附加许多规则,增加用户的负担,不利于普及。
3. 编码的信息来源应主要来自汉字的字形和读音本身,如果还要附加含义、分类等其他因素的方案都很难有长久的生命力。要充分利用声调和使用频度等高效率信息,高频字要简码先见。
4. 编码的目的是区分汉字,而不是组成汉字,所以要提取汉字最显著的特征,如首尾、首尾声等等,省略其它成分,中间不用拆分。
5. 音码趋向于两键,如双拼等方案,但复韵母代码不要太多。
6. 词组趋向四键自动词组,是提高输入效率的重要手段。

近年来计算机硬件和软件的飞速发展,对各种中文编码的汉字系统,又可出了许多更高的要求。通过对许多用户的调查,我们总结认为,现在对中文输入法主要追求的是:

1. 首要的是码元要少,简单明了,易学易记,又科学合理,高速高效。

2. 音形兼容输入,知音打音,知形打形,不用切换;能模糊查询。

3. 字词兼容输入,用户自动现场造词,用户词组库存盘和调入方便。

4. 兼容西文原版软件,少占或不占主内存,支持网络共享。

5. 兼容各种流行的中文排版软件,能外挂运行。

6. 能在中文 WINDOWS 环境下运行,方便输入汉字。

按照上述理论思想为指导,作者经过几年的艰辛努力,充分利用汉字的形、声、频度、结构等四维信息优势,集中了各种编码方案的优点,研究一种“四维形声码”(简称“陆码”)输入法,愿为编码的统一做一颗小小的铺路石。1991年通过省级鉴定,专家鉴定意见:“编码技术研制达到了国内先进水平。”已在广州一些单位推广使用,普遍反映易学易记,方便实用。曾获1992年能源部北京“双新会”银奖和广东省电力局科技进步一等奖,已申请专利(中国专利公开号:CN 1048613 A)。

主要特点介绍如下:

1. 可以在 MS-WINDOWS 3.1 中文版上运行,首尾声三键一字,不会的键可用?代替,三键词组按首尾首或声声输入均可。如 JSJ=计算机。

2. 可外挂金山 WPS、超想 CXDOS、2.13 等汉字系统下运行;充分发挥优秀的打印和显示功能,加上陆码的方便输入,可谓珠联璧合。

3. 支持 DOS6.0 等 HIMEM 设置,可做到不占用主内存。

4. 支持 NOVELL 等网络,原版西文软件如 FOX-PRO 等勿需汉化直上汉字。

5. 输入法仅一句口诀:

“按照笔顺打首尾,第三键打拼音头,兼容拼音尾韵调,词组连续打首尾”。

如打“微”字,只要取首码“J”打“J”,取尾码交叉打“X”即可,中间不用拆分,接打拼音“W”三键已基本唯一;按拼音打 WZ—也可,不用切换;打“微机”只要连续打首尾首尾“JXMU”即可。

(1) 基本码元(字根)仅 40 个,在各种编码中最少,或形象、或声似,如口打 O、尸打 P;人打 R、木打 M 等等,见字识码,四键唯一无重码。

(2) 全国首创单字和词组均可以按字音或字形兼容输入不用切换,既可以象双拼、自然码那样打拼音,又

可以象五笔字型、表形码那样按字形输入。如:学习=XX,中文=OIBX,计算机=JSJ 或 IKMU 均可。

(3) 强大的“现场造词”功能:除了五千常用词外,随时可添加用户词组,即刻可用(如打 DXPY 接着输入“邓小平”三个字,以后再打“DXPY”都有效);所造的新词库随时可存盘,一共可存 3 万多条专用词组。

(4) 用同一种编码可统一处理简体、繁体和日文汉字,已有繁体版本,可适用于单显和各种彩显的 IBM PC 到 486 和便携机等各种机型。

(5) 增强了区位翻页检字、字典学习、100 种常用标点符号、声响提示、提示编码、对折打印、改变文件属性、消毒等功能,人机界面友好。

(6) 附“陆码学习软件”,有系统示范教学等功能,自学一天即可掌握。

陆码汉字系统设计的主要对象是不用专门训练的普通操作者,一般工作人员、科技人员、作家、记者、教师、学生等,尤其是工作繁忙的各级领导干部。由于编码简单无重码、见字识码效率高、高频简码兼词组,对连续文本平均每个汉字仅须击 2 键,短期自学后每分钟可以输入 60 个汉字,可以用电脑写文章报告。熟练可超 200 字。

下面是用“陆码”和“音首”两种方法输入的实例,47 个汉字共打 90 键:

陆码的希望是让每一个中国人都能方便地用电脑输入汉字,

PUSY J XGCE Q IYR JQM Y R7 OIOD R T3 A8 BURX
T U QLYU XJJC DABW.

LUMA D XIWH U RH MZJ Y G VSGO R DU8 N6 FH-
BJ D2 Y3 DJNK UURU HJZI.

迎接 21 世纪信息时代对我们中华民族的严重挑战!

CWFZ 21 HLWL RIJD QSRC A JXRU O R P B J YTJT
FVIX!

YYJE 21 UIJI XNXI UIDL D5 WOMF V H M Z D YJVS
TKVJ!

注:陆码在软件廊的编号为 0027。陆码输入法软件 120 元,包括教材一本和 LCDOS 陆码汉字系统;2.13, WPS, CXDOS 等外挂陆码输入法模块;陆码学习软件;MS-WINDOWS 3.1 的陆码输入法 WINLM 等全套软件。普及版赠送,包括 LCDOS 1.1 和陆码学习软件等,只收磁盘和邮寄费共 20 元。

(有意使用陆码的读者请与本刊联系)

软件廊
软件廊
软件廊
软件廊
软件廊
软件廊

编号:0028 《五笔字型自学速成》

售价: 80.00 元/套

五笔字型自学速成软件

华南师范大学计算机系 梁宇种

你想无师自通五笔字型吗?

你想以最短的时间、最高的效率速成五笔字型吗?

你想大大提高自己的汉字输入速度和准确率吗?

那么,请使用五笔自学速成软件 WBSC 吧!

自从中国迈入电脑时代以来,汉字的输入就成为一大难题。可是自从一九八三年五笔字型输入技术诞生以来,问题迎刃而解。五笔字型以其多学科集成创造的学术优势和巧妙设计,最先突破了每分钟输入一百个字的大关,结束了汉字输入不能与西文相比的历史。五笔字型符合人们的写字习惯,即使你对某个字不认识,仍然可以利用五笔进行输入。如果你学会五笔字型,汉字输入速度一般能达到每分钟 40 字;熟练的,速度能达到每分钟 160 字以上。目前最高的速度是每分钟正确输入 270 字。

五笔输入虽然被广泛应用,但并不意味着学习五笔时一步就能登天。初学者在使用五笔时感到困难的是要把 130 多个字根背熟,每个字根还要对照键盘上的位置,分别放到各区中。因此,用传统方法,依然要经过个把月的强化记忆和训练才能掌握并达到每分钟 40 至 80 字水平。所以有许多人说,五笔用起来虽快,但学起来太难了!

为了使大家能更快地学会五笔,经过一段长时间的酝酿之后,94 年正式推出五笔字型自学速成软件 WBSC。WBSC 最大的特点就是可以无师自通五笔字型,但更重要的是能迅速地熟练掌握它。经过一段时间的试用发现,利用 WBSC 可以使初学者在极短的时间内完全掌握五笔。

为什么 WBSC 有如此让人吃惊的效果呢?先让我们来看一看掌握五笔并达到快速输入的三个关键:

(1) 快速的标准指法输入

(2) 熟记字根

(3) 熟练输入汉字的方法

熟记字根,是五笔学习的第一关,也是最难通过的一关。五笔字根比较难记忆,而且较枯燥,记忆量也很大。WBSC 就是要将学习的难度减少到可能的最低限度。自五笔诞生以来阻碍大多数人学习进程的关卡,在 WBSC 的陪伴下,如今你却能够轻而易举地通过了。

五笔字型方案所选用的字根,是严格按首笔型代号分区的,它按严谨的科学体系划组的首笔代号建立序列,按照字根的形、音、义间类同形相近的关系互相挂钩,是很有规律的。笔者通过对这些规律作了深入的研究,并通过对大量五笔初学者的调查,以及效率学、学习心理学的运用得出了一套五笔字型的速成方法,这就是 WBSC。

WBSC 适应能力强,支持单显、EGA、VGA 显示器。是一套适用于家庭、学校、办公室等地的教学辅助软件。

WBSC 采用了目前流行的下拉式菜单,界面及提示非常友好,一改以往各类五笔学习软件气氛沉闷的局面,并在各方面都比老一代学习软件有很大突破,使人有目一新之感。

WBSC 是中文“五笔速成”四个字的缩写,其核心就在于速成两字。WBSC 以字根学习为重点,其功能有自学速成、快速记忆、快速查询、强化训练,另外还有寓教于乐的字根大战游戏。这是根据人脑记忆中交错感维以及紧张与松弛相结合才能达到最佳记忆强度等特点安排的,可使人在不知不觉中,甚至于游戏中通过五笔中字根这最重要也是最难的一关。另外,WBSC 在单字、词、文章等方面也有其独创性。例如,强化训练环境就是一个小型编辑器,在里面可以不受限制地使用各种编辑功能,并且还允许使用其它汉字输入法。

在你单人学习之时,你也不是寂寞的。在你阅读教材之时,会出现一位“学友”,这个机器中的人物对你进行各种启发和诱导,从而达到达到你思维中最大灵动性的一面。

**欲免海盗软件之害
请到中外软件廊**

地址:广州流花路 119 号锦汉大厦 11 楼 电话:6689457

软件廊
软件廊
软件廊
软件廊
软件廊
软件廊
软件廊

中外软件廊部分软件报价

【编者】在读者、作者和编辑部的共同努力下，软件廊终于搭建起来了。软件廊既是软件沙龙又是软件市场。在软件廊里有两类软件，一类是自由软件，何谓自由软件，自由者可自由拷贝也，也就是说这类软件开发者为推广普及而放弃了版权的软件，或者作者允许非获商业得润的使用。这类软件前均有“☆”符号。另一类是有版权的软件，使用这类软件，必须遵守我国“软件保护条例”的规定。这方面的知识可参阅本刊93年12月《重视BSA的忠告》一文的介绍。邮购软件方法：

1、邮局汇款地址：广州市德政北路393号电脑杂志社科技开发经营部 邮码：510055

2、银行汇款：开户行：农行广州德政北办 帐号：80120003-71 户头：电脑杂志社科技开发经营部

美国 MICROSOFT 公司软件

编号	产品名称	版本	单价
029	DOS6.0	6.0	1099.00 元
030	WINDOWS3.1	3.10	1638.00 元
031	WINDOWS3.1(简体中文版)	3.10	1430.00 元
032	WINDOWS3.1(简体中文版 UPGRADE)	3.10	760.00 元
	用户必须有原版的中文 WINDOWS3.1 才能用此软件升级		
033	WINDOWS FOR WORKGROUPS	3.10	4397.00 元
034	WINDOWS NT	3.10	5442.00 元
035	WINDOWS NT ADVANCED SERVER	3.10	32927.00 元
036	SQL FOR NT(WORKGROUP SYSTEM 10 USERS)	4.20	32927.00 元
037	SQL FOR NT(DEPARTMENT SYSTEM 64 USERS)	4.20	87898.00 元
038	SQL FOR NT(ENTERPRISE SYSTEM UNLIMITED USERS)	4.20	164856.00 元
014	EXCEL FOR WINDOWS	5.00	5442.00 元
016	WORD FOR WINDOWS	3.00	5442.00 元
039	POWERPOINT FOR WINDOWS	3.00	5442.00 元
015	OFFICE FOR WINDOWS	4.00	8246.00 元
040	PROJECT FOR WINDOWS	3.00	7641.00 元
041	WORKS FOR WINDOWS	2.00	2188.00 元
042	VIDEO FOR WINDOWS	1.00	2188.00 元
043	ACCESS FOR WINDOWS	1.10	5442.00 元
044	ACCESS DISTRIBUTION KIT	1.10	5442.00 元
045	FOXPRO FOR DOS	2.50	5442.00 元
046	FOXPRO FOR WINDOWS	2.50	5442.00 元
047	FOXPRO FOR DOS DISTRIBUTION KIT	2.50	5442.00 元

编号	产品名称	版本	单价
048	FOXPRO FOR WINDOWS DISTRIBUTION KIT	2.50	5442.00 元
049	FOXPRO LIBRARY CONSTRUCTION KIT	2.50	5442.00 元
050	FOXPRO CONNETIVITY KIT	2.50	5442.00 元
051	FOX GRAPH	1.00	3249.00 元
052	C/C++ COMPILER (INCLUDE WINDOWS SDK)	7.00	5486.00 元
053	MACRO ASSEMBLER	6.00	1649.00 元
054	WINDOWS SDK	3.10	3838.00 元
055	WINDOWS DDK	3.10	5497.00 元
056	WIN32 SDK	3.10	5442.00 元
057	WINDOWS NT DDK	3.10	5442.00 元
058	MDK FOR WINDOWS	1.00	4343.00 元
059	VISUAL BASIC FOR WINDOWS PROTOOLKIT	2.00	5442.00 元
060	VISUAL C++ PRO	1.00	5486.00 元
061	TESTING TOOLS FOR WINDOWS	2.00	5442.00 元
062	WINDOWS SOUND SYSTEM	3.10	3200.00 元
022	DOS 6.2 STEP UP	6.20	150.00 元

用户必须有原版的 DOS6.0 才能用此软件升级

国内软件公司开发的软件

编号	产品名称	版本	单价
063	长安金山汉卡 III 型	6.1F	780.00 元
064	2.13K 办公排版系统(2.13K - II 汉字系统+2.13K 办公排版系统)		2800.00 元
065	2.13K - II 汉字系统+AUTOCAD 支持		2800.00 元
066	2.13K - II 汉字系统+办公排版系统+AUTOCAD 支持		3200.00 元
067	2.13K 网络汉字系统+网络办公排版系统(10 用户)		5000.00 元
068	2.13K - II 汉字系统+办公排版系统+超级数据库工具包		4300.00 元
069	天汇标准汉字系统 TECHWAY V1.5		680.00 元
070	长城图文编排系统		500.00 元
071	长城 TIDE/C 多媒体数据库系统(单色图象、纯软件)		800.00 元
072	长城 TIDE/C 多媒体数据库系统(彩色图象、带视频卡)		6700.00 元
073	长城五合一办公通讯系统(带 FAX/MODEM 卡)		3380.00 元
074	长城 9000B 字符汉卡		1600.00 元
075	长城 805 图形加速卡		2800.00 元
076	万达财务核算系统		2800.00 元
077	万达通用数据库输入输出系统		1800.00 元
078	万达应收款核算系统		1800.00 元
079	万达应付款核算系统		1800.00 元
080	万达物资管理系统		1800.00 元
081	万达现金出纳管理系统		680.00 元
082	万达通信录及名片管理系统		180.00 元
083	万达全国航班列车轮船时刻表		180.00 元
084	万达用户菜单自生成系统		888.0

联系地址: 广州流花路 119 号锦汉大厦 11 楼中外软件廊 咨询电话: 6689457

综合驱动电子器件卡

♥ 波仔 ♥

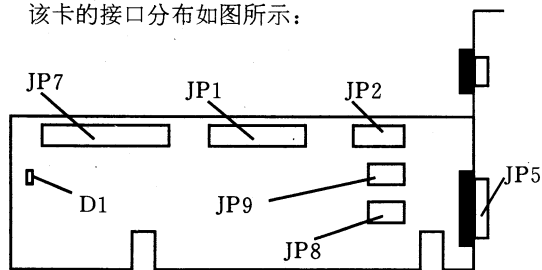
不要被这古怪的名称所骗,这种卡你的电脑里也插着,其实就是我们常说的 IDE(Integrated Drive Electronic)卡的中文直译。

以前一台电脑上是要插很多卡的,配软驱要软盘驱动卡,接硬盘要硬盘驱动卡,带打印机要有打印卡,做通信或装鼠标要 I/O 卡,……加上显示卡,一台电脑里面就要插五块卡。

但现在,打开你的电脑看一看,你会发现,你的电脑里并没有插那么多的卡。“哗,一定被卖电脑的坑了,找他要回那些卡去!”且慢!别激动,因为现在的 Super IDE 卡,是集软盘驱动、硬盘驱动、串行接口、并行打印接口、游戏杆接口于一体的。也就是说那一大堆卡的功能都已集中到 Super IDE 卡的身上了。

我们习惯把只有软硬盘驱动的叫 IDE 卡,而带上串、并行等接口的叫 Super IDE 卡。IDE 卡的售价并不贵,一般十六位的 Super IDE 卡只售 75 到 100 元左右。但麻雀虽小,五脏俱全,你的电脑内数据的“进出口”,全靠 IDE 卡来管理。IDE 卡的生产厂家在国内外都可谓星罗棋布,型号也五花八门,找来一块市场上常见的十六位 Super IDE 卡,把它的功能做一个简单的介绍。

该卡的接口分布如图所示:



一、D1

这是硬盘指示灯的接口,接到机箱的发光二极管(注意正负极)上,当硬盘读写时,发光二极管就会亮。

二、JP1

这是一个三十四针的接口,软驱的驱动由它来完成。它可以驱动两个软驱,用一条三十四线的扁平电缆把它与软驱连接起来。值得注意的是,接线时,电缆不要接反,按习

惯,软驱的电缆上,有红色花边的一端应该接到卡上标有“1”或“2”或“▲”符号的一端(后面接口的接法相同)。当你开机时,如果发现两个软驱和硬盘的指示灯都亮的话,很可能是电缆接反了,赶快关机检查一下。

三、JP7

这是硬盘驱动的接口,它是一个 40 针的接口,也可以驱动两个硬盘。但接两个硬盘时要注意把二号硬盘上的 C/D 接口短接。

四、JP2

它有十四根针,把它接到购买 IDE 卡时所附的接口板上的十四针插座上,再接上 PC 专用的游戏操纵杆,玩游戏时,就不会再跟键盘过不去了。

五、JP8 和 JP9

它们都是标准 RS232 串行接口,主要用来接鼠标和做各种的异步通信,当然也可以接到串行打印机上。一般都配有九针和二十五针的插座,你可以按自己的需要去接。出厂时,这两个接口分别被接成 COM1 和 COM2,如果有必要的话,你可以把它接成 COM3 和 COM4,详情可参阅随卡的说明。

六、JP5

这是并行打印口,用来连接打印机。

七、几点有关说明:

1、以上的所有接口出厂时全部允许使用,可以通过不同的跳线按你的要求选择允许或禁止使用,但每一块卡的跳线方法都不同,跳线之前,先阅读一下随卡说明。

2、硬盘接口只可以驱动 AT 总线的硬盘,以前的 MFM 和 IBM 的 SCSI 制式的硬盘都不能用它来驱动。

3、打印机接口可以选择不同的地址接成一口、二口或三口。

4、串行接口的地址也可以按实际情况选择。

5、有的主机板把 IDE 卡也做到主板上去了,当这部分出故障时,可以将它设置成“禁止使用”状态,另外插上好的 IDE 卡使用。

6、近年推出的 VESA 总线的高速 32 位微机,应使用 32 位的 Super IDE 卡。不但 RAM 会影响机器的运行速度,IDE 卡对机器也有很大的影响。

汉化 README.COM 程序开发与功能扩充

(中)

广东物价局 刘川

1、命令行参数处理功能的扩充

原程序的命令行参数处理程序只有处理文件名参数的功能,当命令行上有文件名参数时将其滤去空格之后取默认的文件名 README。由于未作文件名有效长度检验,因而当命令行参数无效时得到的文件名字长度为0,其结果当然是 File not found。为了克服上述不足,通过修改命令行参数处理程序(入口地址在 XXXX:0BE2),同时增加一个可选的/XX 命令行参数用于改变程序的屏幕显示行数,这使程序具有更好的适应能力。这时的调用命令只需在原命令行参数之后加上/XX 参数即可改变程序显示行数,行数 XX 的值可在 10—25 之间任选。当命令行/参数无效时还可以显示命令行帮助信息。扩充的命令行处理见程序 11。命令行参数处理功能扩充之后,程序的调用命令格式为:

README [[d:][path]name][[/xx]]

修改程序清单之二;

11. 命令行参数处理扩充程序

—A 0BF3

XXXX:0BF3 CALL 1352 ;调用扩充的命令行处理程序

XXXX:0BF6 CMP DI,0DFA ;文件名有效长度检查

XXXX:0BFA JZ 0C03

XXXX:0BFC NOP

XXXX:0BFD NOP

XXXX:0BFE NOP

XXXX:0BFF

—A 1352

XXXX:1352 LODSB ;读命令行一个字符到 AL

XXXX:1353 CMP AL,20 ;空格?

XXXX:1355 JZ 1360 ;Yes,滤去

XXXX:1357 CMP AL,0D ;回车?

XXXX:1359 JZ 1362 ;Yes,结束

XXXX:135B CMP AL,2F ;'/'字符?

XXXX:135D JZ 1363 ;Yes,转/命令处理

XXXX:135F STOSB ;写文件名

XXXX:1360 LOOP 1352

XXXX:1362 RET

XXXX:1363 LODSW ;取 XX 参数

XXXX:1364 XCHG AL,AH

XXXX:1366 SUB AL,30 ;XX 字符必须是'0'—'9'

XXXX:1368 JS 139D

XXXX:136A CMP AL,09

XXXX:136C JA 139D

XXXX:136E SUB AH,30

XXXX:1371 JS 139D

XXXX:1373 ADD ;BCD=>BIN

XXXX:1375 CMP AL,0A

XXXX:1377 JB 139D

XXXX:1379 CMP AL,19

XXXX:137B JA 139D

XXXX:137D MOV [0363],AL;XX=10—25 时修改屏幕行参数

XXXX:1380 MOV [0B5E],AL

XXXX:1383 MOV [0B76],AL

XXXX:1386 DEC AL

XXXX:1388 MOV [0638],AL

XXXX:138B MOV [0AD8],AL

XXXX:138E MOV [0B6E],AL

XXXX:1391 MOV [0DBA],AL

XXXX:1394 DEC AL

XXXX:1396 MOV [0662],AL

XXXX:1399 MOV [0AC5],AL

XXXX:139C RET

XXXX:139D MOV DX,13A3 ;/参数无效

XXXX:13A0 JMP 0304 ;显示命令行 Help

XXXX:13A3

—E 13A3 0D 0A

—E 13A5 'README.COM 2.03F Help 1993.9 Liu Chuan' 0D 0A 0D 0A

—E 13D1 'Command:' 0D 0A 09

—E 13DC 'README / for help' 0D 0A 09

—E 13F3 'README [[d:][path]name][[/xx]]' 0D 0A

自然码

最自然的汉字输入法

广东代理:电脑杂志社 电话:5516911—3273

-E 1411 'Examples:' 0D 0A 09
 -E 141D 'README *. * for read all files' 0D 0A 09
 -E 143E 'README for read default file README' 0D 0A 0D 0A
 -E 146A '/xx for change display line (xx=10-25,default 25)\$'
 -W

Writing 1668 Bytes

III、阅读 WPS 文本的程序设计

一、WPS 文本的自动识别与解密程序设计

WPS 文本的特征是正文之前多了一个 1K 字节的文件头,用于保存 WPS 文本标识、密码以及编辑状态等信息,文件头的第一、二字节为 01FFH(也有文章指出第一字节值为 01-03 之间)。根据这一标识可以实现文件类型的自动识别。在确认为 WPS 文本之后,设置文件标志 1 和文件头偏移量 400H,同时从文件头中取出密码。WPS 对文本加密时可选择 0-8 位密码口令,经运算之后形成加密密码存放在文件头偏移 02DDH 处开始的单元中,并以 ASCII 0 作为结束符,对于无加密文本,相当于密码有效长度为零位。在取出密码的同时将密码有效长度不足八位的以 0H 补足八位数,然后以固定八字节长度分组与八位密码作异或运算进行解密。注意到任何数与 0H 异或运算的结果不变,因而不论文本是否加密都可以使用同样的异或解密运算对读入的文本数据进行处理,这可简化编程而不必检测文本是否有加密。由于对正文的排版控制字符过滤时需要检测下一个字符的值,因而数据处理时需要同时解密当前字符和下个字符,解密之后再对正文的排版控制字符进行过滤处理(与 WS 文本过滤相同),从而实现 WPS 文本的自动识别与自动解密阅读。WPS 文本的自动识别与数据解密处理见程序 12-15。

由于在编辑的文本中既可以使用编辑软件的打印控制字符集,也可使用打印驱动程序的控制字符集,因而难以将这些控制字符彻底滤去,故本文未讨论对这类控制字符的过滤处理。WPS 文本的字体格式控制字符的值在 80H-A0H 之间,如需滤去,可在解密后进行检测过滤。

二、修改程序清单之三

;12. WPS 文本自动识别与数据处理程序接口

-A 075E
 XXXX:075E CALL 1218 ;WPS 文本识别
 XXXX:0761
 -A 07AE
 XXXX:07AE CALL 1254 ;WPS 数据处理
 XXXX:07B1

;13. 读文件头 CX 字节子程序

XXXX:11FE MOV BX,[0185] ;取文件句柄
 XXXX:1202 PUSH CX
 XXXX:1203 XOR CX,CX

XXXX:1205 XOR DX,DX
 XXXX:1207 MOV AX,4200 ;文件指针指向文件首
 XXXX:120A INT 21
 XXXX:120C POP CX
 XXXX:120D SUB SI,SI
 XXXX:120F MOV DS,[0183] ;指向 Text Buffer 段
 XXXX:1213 MOV AH,3F ;读文件到 Text Buffer
 XXXX:1215 INT 21
 XXXX:1217 RET

;14. WPS 文件识别处理程序

XXXX:1218 CMP BYTE PTR [1189],00 ;已有特殊文本标志?
 XXXX:121D JNZ 1253 ;Yes,返回
 XXXX:121F PUSH DS
 XXXX:1220 MOV CX,0400 ;读出文件头 400H 字节
 XXXX:1223 CALL 11FE
 XXXX:1226 LODSW ;查 WPS 文本标识 01FFH
 XXXX:1227 CMP AL,01
 XXXX:1229 JNZ 1230
 XXXX:122B CMP AH,FF
 XXXX:122E JZ 1233
 XXXX:1230 POP DS
 XXXX:1231 JMP 1253 ;无 WPS 标识,返回
 XXXX:1233 MOV SI,02DD ;指向 WPS 密码首址
 XXXX:1236 MOV DI,1191 ;指向密码变量首址
 XXXX:1239 MOV CX,0008 ;共 8 位密码
 XXXX:123C CMP AL,00 ;密码结束符?
 XXXX:123E JZ 1241 ;Yes,存 0H
 XXXX:1240 LODSB ;No,读一位密码
 XXXX:1241 STOSB ;存入密码变量中
 XXXX:1242 LOOP 123C
 XXXX:1244 POP DS
 XXXX:1245 MOV [1190],AL
 XXXX:1248 MOV BYTE PTR [1189],01 ;置 WPS 文本标志 1
 XXXX:124D MOV WORD PTR [118A],0400 ;置文件头偏移量
 XXXX:1253 RET

;15. WPS 解密过滤处理程序

XXXX:1254 PUSH DS
 XXXX:1255 PUSH CS
 XXXX:1256 POP DS
 XXXX:1257 CMP BYTE PTR [1189],01 ;WPS 文本?
 XXXX:125C JNZ 126A ;No,返回
 XXXX:125E MOV BX,[118C] ;取字节计数低 8 位
 XXXX:1262 AND BX,0007
 XXXX:1266 XOR AX,[BX+1190] ;异或解密 AX
 XXXX:126A POP DS
 XXXX:126B RET
 -W

Writing 1668 Bytes

宏图电脑

地址:广州市文德路 84 号 邮政编码:510030 电话:3325843

IV、阅读 DBF 数据库文件的程序设计

一、DBF 数据库文件的识别与数据格式转换程序设计

DBF 数据库文件开头由结构说明表(32 字节,见表 2)、字段说明表(每个字段占 32 字节,见表 3)和结束符(0DH 或 0D00H)三部分组成文件头,文件头之后才是数据记录,最后以 1AH 作为文件结束符。文件头的第一字节为 DBF 文件标识 03 或 83H。因此,可以根据文件的扩展名是否为 DBF 和文件第一个字节的值自动识别 DBF 文件

表 2、DBF 结构说明表

字节位移	字节数	含义
0	1	标志 03/83H
1—3	3	最后修改日期
4—7	4	记录数
8—9	2	文件头长度
10—11	2	记录宽度
12—31	20	保留

表 3、DBF 字段说明

字节位数	字节数	含义
0—9	10	字段名,不足部分填 0
10	1	分隔符
11	1	字段类型
12—15	4	字段数据地址
16	1	字段宽度
17	1	小数位数
18—31	14	保留

通过改变文件头偏移量设置方式来实现是否显示 DBF 的文件结构——当设置文件头偏移量等于 0 时,处理包括整个文件,因而可阅读 DBF 文件结构和数据记录;当设置文件头偏移量等于 DBF 文件头长度时,处理仅仅是 DBF 的数据区,因而阅读的只是 DBF 文件的数据记录。M 开关可以任意切换,但对于非 DBF 文件则为无效键。

由于 DBF 文件中既有 ASCII 字符也有二进制数据,需要根据当前处理的字节在文件中的位置作不同的处理,使数据能够转换为标准文本显示的格式。

1、文件头的前 32 字节结构说明表替换成“字段名称 类型 宽度 小数”结构标题字符串,未用的字节改为 0H 滤

去。

2、字段说明从文件偏移 32 字节处开始,按每个字段显示一行进行处理。每个字段中的字段名称和类型中的 ASCII 0 必须换为 20H(空格)以免被滤去而影响显示格式,其后的字段宽度和小数位数为二进制数,需要转换为十进制 ASCII 字符,其余字节换为 0H 滤掉,但字段最末一个字符改为 0AH 作为换行结束一个字段。经过上述处理,可以使显示 DBF 数据库文件结构与 FOXBASE/DBASE 的 DISPLAY STRUCTURE 命令显示的结果相同。

3、在文件头之后至文件结束符 1AH 之间为数据记录(注意,DBF 数据记录被删除 pack 之后并没有将文件作实际的截短,以至于 1AH 结束符并非文件最后字节)。由于每一个记录的第一个字符为删除标志,只要将该字符换为 0AH 字符即可实现从数据记录到文本格式的转换,从而使阅读时可以按记录分行格式化显示。

程序修改时已按阅读 1K 字节进行加宽,可以满足绝大部分 DBF 文件阅读的需要。如果需要阅读记录宽度大于 1K 的文件,可以按表 1 的方法将 Line Buffer 的长度加大到满足要求。

DBF 数据库文件的自动识别与数据格式转换处理见程序 16—20。

二、修改程序清单之四

;16. 程序接口

—A 0761

XXXX:0761 CALL 126C ;DBF 文件识别

XXXX:0764

—A 07B1

XXXX:07B1 CALL 12E5 ;DBF 数据处理

XXXX:07B4

;17. 修改 M 命令为显示 DBF 结构 ON/OFF 开关

—A 040C

XXXX:040C MOV AX,[1188]

XXXX:040F CMP AH,02 ;文件类型=DBF?

XXXX:0412 JNZ 03C6 ;No,无效返回

XXXX:0414 XOR AL,01 ;改变 DBF 显示方式

XXXX:0416 SUB AH,AH ;清文件类型

XXXX:0418 MOV [1188],AX

XXXX:041B JMP 051F

XXXX:041E NOP

XXXX:041F

;18. DBF 数据库文件识别程序

XXXX:126C CMP BYTE PTR [1189],00 ;已有特殊文本标志?

XXXX:1271 JNZ 12BA ;Yes,返回

XXXX:1273 MOV BX,[0173] ;指向文件名

XXXX:1277 MOV DL,0B

XXXX:1279 MOV DI,1168 ;指向'.DBF'串首

XXXX:127C MOV SI,BX ;文件名查找起址

**美国惠普(HP) 美国保时(Pulse)特约代理
海谊电子仪器实业公司**

电话:4420788 转 8912、8910、8916

```

XXXX:127E  MOV CX,0004    ;在文件名中查找'.DBF'
XXXX:1281  REPZ
XXXX:1282  CMPSB
XXXX:1283  JZ 128C
XXXX:1285  INC BX
XXXX:1286  DEC DL
XXXX:1288  JNZ 1279
XXXX:128A  JMP 12BA      ;查找不到
XXXX:128C  PUSH DS      ;查到
XXXX:128D  MOV CX,0020    ;读出文件头 20H 字节
XXXX:1290  CALL 11FE
XXXX:1293  MOV AX,[SI+08] ;取出 DBF 文件头长度
XXXX:1296  MOV BX,[SI+0A] ;取出 DBF 记录宽度
XXXX:1299  AND BYTE PTR [SI],7F
XXXX:129C  CMP BYTE PTR [SI],03 ;查 DBF 标识 03/83H
XXXX:129F  POP DS
XXXX:12A0  JNZ 12BA      ;无 DBF 标识,返回
XXXX:12A2  MOV [1184],AX ;存 DBF 文件头长度
XXXX:12A5  MOV [1186],BX ;存记录宽度
XXXX:12A9  CMP BYTE PTR [SI] ;显示 DBF 结构?
XXXX:12AE  JNZ 12B2      ;No
XXXX:12B0  XOR AX,AX      ;Yes
XXXX:12B2  MOV [118A],AX ;置文件头偏移量
XXXX:12B5  MOV BYTE PTR [1189],02 ;置 DBF 文件标志 2
XXXX:12BA  RET

```

;19. 字段宽度,小数位数转换为 4 位十进制字符子程序

```

XXXX:12BB  PUSH AX      ;保护当前字符
XXXX:12BC  MOV AX,[SI+04] ;取字段宽度,小数位数
XXXX:12BF  PUSH AX
XXXX:12C0  MOV BX,2020
XXXX:12C3  MOV [SI],BX
XXXX:12C5  MOV [SI+06],BX ;前导 0 为空格
XXXX:12C8  SUB AH,AH      ;转换字段宽度
XXXX:12CA  CALL 11CE
XXXX:12CD  MOV [SI+02],DX ;高二位
XXXX:12D0  MOV [SI+04],AX ;低二位
XXXX:12D3  POP AX
XXXX:12D4  CHG AL,AH      ;转换小数位数
XXXX:12D6  SUB AH,AH
XXXX:12D8  MOV BL,20      ;前导 0 为空格
XXXX:12DA  CALL 11CE
XXXX:12DD  MOV [SI+07],DX ;高二位
XXXX:12E0  MOV [SI+09],AX ;低二位
XXXX:12E3  POP AX
XXXX:12E4  RET

```

;20. DBF 文件数据格式转换处理程序

```

XXXX:12E5  PUSH DS
XXXX:12E6  PUSH CS
XXXX:12E7  POP DS
XXXX:12E8  CMP BYTE PTR [1189],02 ;DBF 文件?
XXXX:12ED  JNZ 1350      ;No,返回
XXXX:12EF  MOV BX,[118C] ;取字节计数指针到 DX:BX
XXXX:12F3  MOV DX,[118E]
XXXX:12F7  CMP DX,+00     ;DX>0 为数据记录区

```

```

XXXX:12FA  JZ 1316
XXXX:12FC  PUSH AX      ;记录格式转换
XXXX:12FD  MOV AX,BX
XXXX:12FF  SUB AX,[1184] ;减去文件头长度
XXXX:1303  SBB DX,+00    ;计算当前字符在记录中位置
XXXX:1306  DIV WORD PTR [1186]
XXXX:130A  POP AX
XXXX:130B  CMP DX,+01    ;是记录第一个字节?
XXXX:130E  JNZ 1350      ;No,字符输出
XXXX:1310  CMP AL,1A     ;文件结束符?
XXXX:1312  JZ 1350       ;Yes,结束
XXXX:1314  JMP 1336      ;No,改删除标志为 0AH 换行
XXXX:1316  CMP BX,[1184] ;BX>文件头长度?
XXXX:131A  JA 12FC       ;Yes,仍为数据记录区
XXXX:131C  CMP BX,+20    ;结构说明表?
XXXX:131F  JNB 132E      ;NO
XXXX:1321  MOV AL,[BX+116B] ;Yes,置换为结构标题
XXXX:1325  CMP BL,19     ;第 19H 字节之后改为 0H 滤掉
XXXX:1328  JB 1350
XXXX:132A  MOV AL,00
XXXX:132C  JMP 1350
XXXX:132E  AND BL,1F     ;字段说明处理
XXXX:1331  CMP BL,00     ;字段内字符指针 BL
XXXX:1334  JNZ 133A
XXXX:1336  MOV AL,0A     ;字段最后一个字节改为 0AH 换行
XXXX:1338  JMP 1350
XXXX:133A  CMP BL,18     ;第 18H 字节之后改为 0H 滤掉
XXXX:133D  JNB 132A
XXXX:133F  CMP BL,0C
XXXX:1342  JNZ 134A
XXXX:1344  POP DS
XXXX:1345  CALL 12BB      ;字段宽度和小数位数转换处理
XXXX:1348  JMP 1351
XXXX:134A  CMP AL,20     ;字段说明中<20H 的字符改为空格
XXXX:134C  JNB 1350
XXXX:134E  MOV AL,20
XXXX:1350  POP DS
XXXX:1351  RET

```

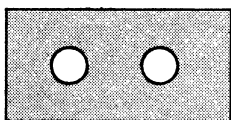
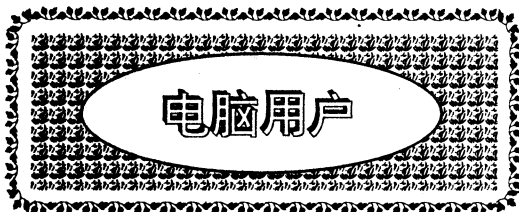
-W

Writing 1668 Bytes

124

简讯

美国著名的网络公司 Novell 公司拟于 94 年 6 月 18 日在广州中国大酒店举办“90 年代网络平台”展示和研讨会,届时将有 Novell 有关人员介绍网络技术的发展,展示 Novell 网络技术产品,同时还有国内几家最有实力的网络公司展出其它网络产品。由于参加的人数有所限制,请尽快向有关代理索取请柬。



XENIX



磁带备份和恢复的实用程序

浙江省台州地区人民银行 林荣庆

数据日常备份是计算机使用中不可缺少的操作,以便在系统发生意外事故后进行数据文件的恢复。磁带容量大,适合于每天批量大的数据文件的连续备份,磁带在查阅和存档方面比起软盘来说有其独到的好处。XENIX 系统为磁带操作已提供二个命令,一个是 tar 命令用于磁带的显示、拷入和拷出,另一个是 tape 命令用于磁带的维护和管理,由于磁带的特殊结构,tar 命令不能选择磁带的定位,而定位则要另外使用 tape rfm 命令,但 tape rfm 命令每次只能移动一批数据,对于备份天数较多的磁带来说要反复使用 tape rfm 命令,显然很不方便,容易出现张冠李戴,使备份和恢复的数据文件混乱,甚至造成无法挽回的损失。

今笔者用 C 语言编写了一个磁带备份和恢复程序 cd-tape.c,利用菜单选择方式,自动执行磁带有关功能的操作,并有状态文件显示磁带使用的情况,便于用户选择,该程序用于一合磁带连续备份一个月内的每天的数据文件。

菜单和状态表显示如下。

磁 带 备 份					
记录号	日期	记录号	日期	记录号	日期
1	921101	2	921102	6	----
7	----	8	----	12	----
13	----	14	-----	18	----
19	----	20	----	24	----
25	----	26	----	30	----
31	----				

- 1、当天磁带备份
 - 2、磁带备份满,清状态文件
 - 3、指定记录号磁带备份
 - 4、打印磁带备份表
 - 5、磁带恢复
 - 6、显示磁带备份目录文件
 - 7、退出
- 请输入选择:

程序中 dbfile 是状态文件,初次使用时会自动产生,此文件中每条记录存有记录号及相应日期,已备份的记录号标有某天的日期,程序中假定要备份和恢复的数据文件在 /usr/lrQ/back 目录下。当选择备份、恢复及显示功能时,

程序先读状态文件,根据用户输入的记录号和日期,通过系统调用 tape rfm 命令执行磁带定位,而后调用 tar 命令执行磁带的显示、拷入和拷出,然后修改状态文件。

当天磁带备份,指把某天的数据文件紧跟在上次备份位置之后复制到磁带上。

指定记录号磁带备份,指当天或其前面连续几次备份无效而要删除时采用此功能,输入的记录号受到限制,并有出错警告信息显示。

磁带恢复,指把某记录号下的数据文件从磁带复制到 back 目录中。显示磁带备份目录文件,指显示某记录号下磁带某段的文件目录。

打印磁带备份表,可把状态文件的每条记录打印出来,供查看磁带备份情况和存档备案。

磁带备份满、清状态文件,指状态表已用满一个月进行清除,产生原始状态表,以便供下个月使用另一合磁带。

程序已在长城、东海 386 机上通过,使用简便可靠,效果显著,今提供给广大用户参考。

```
#include "stdio.h"include "ctype.h"
#include "string.h"
#define dbname "dbfile"
main()
{
    int k,i,j,xz,jlh;
    char name1[6];
    typedef struct
    {
        char name[6];
        int age;
    }record;
    FILE *database,*fopen();
    record person[32];
    while(xz != 7)
    {
        system("clear");
        if((database=fopen(dbname,"r"))==NULL)
            goto csh;
        fread(person,sizeof(record),32,database);
        fclose(database);
```

```

printf(" 磁带备份\n");
printf("-----\n");
printf("记录号 日期;记录号 日期;记录号 日期;记录号 日期;记录号 日期;");
printf("记录号 日期\n");
printf("-----\n");
k=1;
for(j=0;j<6;j++)
{
    printf("\n");
    for(i=k;i<k+6;i++)
    {
        if(i==32)
            break;
        printf("%4d%8s",person[i].age, person[i].name);
    }
    k=k+6;
}
printf("\n");
printf("-----\n");
printf("1. 当天磁带备份 2. 磁带备份满,清状态文件 3. 指定记录号磁带备份\n");
printf("4. 打印磁带备份表 5. 磁带恢复 6. 显示磁带备份目录文件\n");
printf("7. 退出.\n");
printf("\n");
printf("输入选择:");
scanf("%d",&xz);
switch(xz)
{
    case 1:{
        printf("请输入日期(yymmdd):");
        scanf("%6s",name1);
        database=fopen(dbname,"r");
        fread(person,sizeof(record),32,database);
        fclose(database);
        for(i=1;i<32&& person[i].name[5]!='-',i++)
            system("tape rfm");
        system("tar cv8/usr/lrq/back");
        for(j=0;j<6;j++)
            person[i].name[j]=name1[j];
        database=fopen(dbname,"w");
        fwrite(person,sizeof(record),32,database);
        fclose(database);
        break;
    }
    case 2:{
        for(i=0;i<32;i++)
        {
            person[i].age=i;
            for(j=0;j<6;j++)
                person[i].name[j]='-';
        }
        database=fopen(dbname,"w");
        fwrite(person,sizeof(record),32,database);
        fclose(database);
        break;
    }
    case 3:{
        database=fopen(dbname,"r");
        fread(person,sizeof(record),32,database);

```

```

        fclose(database);
        k=0;
        for(i=1;i<32&& person[i].name[5]!='-',i++)
            k=i;
        printf("请输入记录号:");
        scanf("%d",&jlh);
        if(jlh>k+1)
        {
            printf("记录号太大,重选!");
            break;
        }
        printf("请输入日期(yymmdd):");
        scanf("%6s",name1);
        for(i=1;i<jlh;i++)
            system("tape rfm");
        system("tar cv8/usr/lrq/back");
        for(j=0;j<6;j++)
            person[i].name[j]=name1[j];
        for(i=jlh+1;i<32;i++)
            for(j=0;j<6;j++)
                person[i].name[j]='-';
        database=fopen(dbname,"w");
        fwrite(person,sizeof(record),32,database);
        fclose(database);
        break;
    }
    case 4:{
        database=fopen(dbname,"r");
        fread(person,sizeof(record),32,database);
        fclose(database);
        database=fopen("/dev/lpo","w");
        for(i=1;i<32;i++)
            fprintf(database,"%4d%8s\n",person[i].age,
                person[i].name);
        fclose(database);
        break;
    }
    case 5:{
        printf("请输入记录号:");
        scanf("%d",&jlh);
        for(i=1;i<jlh;i++)
            system("tape rfm");
        system("tar xv8 /usr/lrq/back");
        break;
    }
    case 6:{
        printf("请输入记录号:");
        scanf("%d",&jlh);
        for(i=1;i<jlh;i++)
            system("tape rfm");
        system("tar tv8");
        break;
    }
}
}

```

最近,许多计算机杂志及报纸上有许多文章介绍在西文 DOS 下显示汉字的方法。这些方法基本上可分为三类:(1)直接从硬盘中读出所要的汉字点阵,在图形方式下绘出汉字。这虽然可以显示所有汉字但是速度太慢,对于不要求高速度且每次显示少量汉字的应用还是可以使用的。(2)组成一个专用小字库,使用前调入内存,然后在图形方式下以描绘点阵的方法显示出来。同第一种方法类似,这种方法具有较高速度,但更换汉字要重新组织小字库,比较繁琐。由于内存有限,能显示的汉字数目有限。故这种方法适用于要显示汉字不是经常变化且仅需要图形方式的场所。(3)在 EGA/VGA 及兼容适配器利用 BIOS 的 10H 的 11H 号子功能将所用的汉字作为扩展字符集装入视屏缓冲区的第二页。这种显示方法顾及了显示的速度,和西文高度兼容。这时汉字是作为字符显示的,可选背景和属性,与英文不同的是每个汉字占用两个 ASCII 编码且要分两次显示。故这种方法编码不方便,更换显示的汉字比较麻烦,且每屏同时显示的汉字有限(对 EGA 最多为 256 个),对显示的内容是经常更换的应用是不适用的。

可见,这三种方法都没法高速显示任意汉字而又不占用主存,解决这个问题,笔者在雷达数字化系统软件中采用了如下方法:将字库装入扩展内存(EXT),显示时从扩展内存中读出点阵信息,然后直写显示存储器绘出汉字。由于现在主流微机是 286,扩展内存至少有的 384K,而最新 DOS 版本仍然无法突破臭名昭著的“640K 屏障”,将数据装入扩展内存,节约宝贵的主存,充分利用系统资源,具有重要意义。

所附的程序解释了这种方法的基本过程。程序以汇编语言写成,可

湖北鄂西州气象局 郑天波

西文 DOS 下 汉字高显示

以在屏幕任意位置显示 16 色汉字或西文。过程稍加修改即可供 C 语言程序调用。主程序首先调用过程 OPEN—CCLIB 将 16 点阵字库全部传输到扩展内存的从 1M+64K 开始的区域,再设置字符串首址,光标位置,字符颜色等入口参数后调用过程 WR—CHR 显示汉字字符串,然后退出。

本文的目的主要是给出这种方法的完整的实现过程,关于汉字的显示原理,EGA/VGA 编程及扩展内存使用已有大量的文章可以参考,在此就不再多加以解释了。程序中包含有详细的中文注释,故下面仅对程序及使用作几点说明:

(1)考虑只有 286 以上机器才有扩展内存,程序使用了 286 指令,建议使用 MASM5.X 或 TASM2.X 汇编。

(2)过程说明:

过程 MOVE—BLOCK 用于在存储器任意两个区域间传输数据块,即这两个区域既可是主存也可是扩展内存。

入口:

SR—LW:

源地址低字(BIT0—BIT15)。

SR—HW:

源地址高字节(BIT16—BIT23)。

DE—LW:

目的地址低字(BIT0—BIT15)。

DE—HW:

目的地址高字节(BIT16—BIT23)。

BYTES—COUNT:

数据块字节数。

注意这里源和目的地址为 24BIT,不要与段和偏移混淆,16 位到 24 地址的变换由过程 FORM—24BIT—ADDRESS 完成。

过程 FORM—24BIT—ADDRESS 入口:

DX:段址

CX:偏移

返回:

DX:地址低字

CL:地址高字节

过程 OPEN—CCLIB 将汉字显示字库调入扩展内存,本文使用的汉字库长度为 261696 字节,使用其他字库应适当的修改过程 CCHAR—DISPLAY。

过程 WR—CHR 支持汉字和英文混合字符串显示,其中显示汉字调用了 CCHAR—DISPLAY。

过程 CCHAR—DISPLAY 采用直接写视频存储器的方法在当前光标处绘出汉字。

入口:

AL:汉字内码高字节(=区码+0A0H)。

BL:汉字内码低字节(=位码+0A0H)。

BUFFER:汉字点阵数据(共 32 字节)。

(3)安全使用扩展内存。

为安全的使用扩展内存,CONFIG.SYS 中应有如下设置:

DEVICE = HIMEM. SYS /
INT15=384

则系统启动时将为 INT15 服务预留 384K 的扩展内存供装载字库,且不会与支持 XMS 规范的虚拟盘和高速磁盘缓存发生冲突。当然也可以

修改 MOVE—BLOCK 为支持 XMS 规范的版本,关于 XMS 规范的讨论已超出了本文的范围。

给出的程序适用于各种配有 E-GA/VGA 兼容显示适配器的 IBM PC/AT 兼容机。

```
;使用 286 指令
.286C
;保护方式下段描述符结构
DESC STRUC
DESC—LMT DW 0 ;段限
DESC—BASEL DW 0 ;地址低位字
DESC—BASEH DB 0 ;地址高位字节
DB 0
DW 0
DESC ENDS
.MODEL SMALL
.STACK 128
.DATA
;保护方式下全局描述符表
GDT LABEL BYTE
DESC <>
DESC <>
SRC DESC<,,,93h,>
TGT DESC<,,,93h,>
DESC<>
DESC<>

SEG—KP DW 0 ;保存分配内存段址
MSG1 DB 0dh,0ah
db '高速显示汉字',0ah,0dh
DB '(C)鄂西州气象局雷达站',0dh,0ah
DB '郑天波 1992 年 7 月',0dh,0ah
DB ' 邮编:445000',0dh,0ah
MSG2 db 0ah,0dh,'请按任意键','$'
color DB 0 ;字符颜色
sr—lw DW 0 ;源地址低字节
sr—hw DB 0 ;源地址高字节
de—lw DW 0 ;目的地址低字节
de—hw DB 0 ;目的地址高字节
bytes—count DW 0 ;每次传送的字节数
buffer DB 32 DUP(0ah) ;显示缓冲区
cclib DB 'cclib.dat',0 ;字库名
handle DW 0 ;文件句柄
PG—END DW ZSEG ;程序结束地址

;主程序
START:
mov dx,ⓐATA
mov ds,dx

;VGA:al=12h,EGA:al=10h
mov ah,00h
mov al,12h ;VGA:640*480
int 10h
;调字库入扩展内存
```

```
call OPEN—CCLIB
lea si,MSG2
mov color,12
call WR—CHR
mov ah,01h
int 21h
lea si,MSG1
mov color,10
call WR—CHR
mov ah,01h
int 21h
mov ah,00h
mov al,03h
int 10h
mov ah,4ch
int 21h

;读字库到扩展内存
OPEN—CCLIB PROC
push es
push ds
pusha
;取程序的 PSP 段址
;调整内存大小
push es
mov ah,62h
int 21h
mov es,bx
mov ah,4ah
mov bx,PG—END
mov cx,es
sub bx,cx
inc bx
int 21h
pop es

;为读字库分配内存
mov ah,48h
mov bx,1000h
int 21h
jc jerror
mov SEG—KP,ax
mov di,0FFFFh
jmp openf
jerror: mov ah,48h
int 21h
shl bx,4
mov di,bx
mov SEG—KP,ax

;打开字库文件
openf: mov dx,offset cclib
mov cx,0
mov al,0
mov ah,3dh
int 21h
mov handle,ax
mov bx,ax

JMPP2:
;置字库在扩展内存的 24 位首址
MOV de—lw,0
mov de—hw,11h
```

```
;形成字库缓冲区 24 位地址
mov dx,SEG—KP
mov cx,0000h
call form—24bit—address
mov sr—lw,dx
mov sr—hw,cl
mov bytes—count,di

;读字库到内存缓冲区
READ: mov cx,di
push ds
mov ax,SEG—KP
mov ds,ax
mov dx,0000h
mov ah,3fh
int 21h
pop ds
add ax,0
jz close
cmp ax,di
jz moveto
mov bytes—count,ax

;汉字点阵数据传送到扩展内存
moveto: call MOVE—BLOCK
clc
adc de—lw,ax
adc de—hw,0
jmp read

;关闭文件
CLOSE: mov ah,3eh
mov bx,handle
int 21h

;释放所分配的内存
mov ah,49h
mov dx,SEG—KP
mov es,dx
int 21h
popa
pop ds
pop es
ret

OPEN—CCLIB ENDP
;存储器任意区域间数据块传送
MOVE—BLOCK PROC NEAR
pusha
push es
push ds
mov cx,sr—lw
mov SRC.DESC—BASEL,cx
MOV CL,sr—hw
mov SRC.DESC—BASEH,CL
mov SRC.DESC—LMT,0FFFFh
MOV cx,de—lw
mov TGT.DESC—BASEL,CX
MOV CL,de—hw
mov TGT.DESC—BASEH,CL
MOV TGT.DESC—LMT,0FFFFh
MOV CX,bytes—count
shr cx,1
push ds
pop es
```



```

mov si,offset GDT
mov ah,87H
int 15H
pop ds
pop es
popa
ret
MOVE—BLOCK ENDP
;16 位地址到 24 位地址变换
FORM—24BIT—ADDRESS PROC NEAR
push ax
rol dx,4
mov ax,dx
and dl,0f0h
and ax,0fh
add dx,cx
mov cx,ax
adc cl,ch
pop ax
ret
FORM—24BIT—ADDRESS ENDP
;显示汉字
;入口:AX=内码高位
; BX=内码低位
CCHAR—DISPLAY PROC NEAR
pusha
;获得汉字点阵在字库中的偏移
sub ax,0a0h
sub ax,1
sub bx,0a0h
mov cx,94
mul cx
sub bx,1
add ax,bx
;得到汉字在扩展内存中地址
mov cx,32
mul cx
add dx,11h;从 1M+64K 开始
MOV sr—lw,ax
mov sr—hw,dl
;置点阵缓冲区地址
mov dx,seg buffer
mov cx,offset buffer
call form—24bit—address
mov de—lw,dx
mov de—hw,cl
mov bytes—count,32;16 点阵汉字
;读点阵数据到主存缓冲区
call MOVE—BLOCK
;显示汉字
call WRITE—VRAM
diwri: popa
ret
CCHAR—DISPLAY ENDP
;直写显示存储器显示汉字于当前光标处
WRITE—VRAMPROC NEAR
pusha
push es
;获得光标位置
push ds

```

```

mov ax,0000h ;DS=BIOS 数据段
mov ds,ax
mov ax,DS:[0450h]
;光标移动两个字符
cmp al,76
jbe inc01
inc byte ptr ds:[0451h]
mov byte ptr ds:[0450h],0
jmp inc02
inc01: inc byte ptr DS:[0450h]
inc byte ptr DS:[0450h]
inc02: pop ds
;获得字符偏移
mov bl,al
mov bh,00h
mov al,ah
mov ah,00h
mov cx,80*16
mul cx
mov di,ax
add di,bx ;DI=字符偏移
;设置颜色
mov ah,color
mov al,02h
mov dx,3c4h
out dx,ax
;直写显存
mov ax,0a000h
mov es,ax
mov si,offset buffer
mov cl,0
xxxx: mov ax,[si]
mov es:[di],ax
inc cl
add si,2
add di,80
cmp cl,16
jb xxxx
;开放四个位面
mov ah,0fh
mov al,02h
mov dx,3c4h
out dx,ax
pop es
popa
ret
WRITE—VRAM ENDP
;显示字符串(包括汉字和英文)
;入口:SI=字符缓冲区首地址
; 字符串以$结尾
WR—CHR PROC NEAR
PUSH AX
PUSH BX
PUSH CX
PUSH DX
WR—CHR1:
MOV AL,[SI]
CMP AL,0DH
JE WR—CHR2
CMP AL,0AH

```

```

JE WR—CHR2
CMP AL,'$'
JE WR—CHR0
;字符高位=1
goto12: cmp al,0a0h
jb not—cchar
mov al,[si+1]
cmp al,0a0h
jb not—cchar1
;显示汉字
mov al,[si]
mov ah,0 ;区码
mov al,al
mov bl,[si+1]
mov bh,0 ;位码
call cchar—display
inc si
inc si
jmp WR—CHR1
;ASCII 值小于 128 的英文字符
not—cchar1:
mov al,[si]
not—cchar:
mov cx,1
MOV BL,color
mov bh,0
mov ah,09h
int 10h
;移动光标
push ds
mov ax,0000h ;DS=BIOS 数据段
mov ds,ax
mov ax,DS:[0450h]
;光标移动两个字符
cmp al,79
jbe winc01
inc byte ptr ds:[0451h]
mov byte ptr ds:[0450h],0
jmp winc02
winc01: inc byte ptr DS:[0450h]
winc02: pop ds
inc si
JMP WR—CHR1
;回车与换行
WR—CHR2:MOV AH,2
MOV DL,AL
INT 21H
inc si
JMP WR—CHR1
;结束字符'$'
WR—CHR0:mov color,0
POP DX
POP CX
POP BX
POP AX
RET
WR—CHR ENDP
ZSEGEQU $
END START

```

Microsoft DOS5.0 引入了扩充内存规范 (XMS) 功能。本文谈谈如何利用 XMS, 修改金山 DOS 使之与 MS DOS5.0 兼容为例, 介绍如何利用 XMS 功能扩充内存作数据存取区, 从而节约“宝贵”的常规内存。

在使用 XMS 之前, 首先利用中断 2FH 来检验 XMS 的存在及入口。利用 MOV AX, 4300H 和 INT 2FH 两条语句来检测 XMS 的存在。若返回 AL=80H 则说明 XMS 存在。再执行 MOV AX, 4310H 和 INT 2FH, 此时 XMS 的入口地址返回在 ES:BX 中, 以后可调用这个入口来完成 XMS 功能, (参见附录中 CHLIB.COM 修改部分的 5200H 至 5215H 的内容)。

使用 XMS 扩充内存时, 很象对文件的操作。先申请扩充内存 (类似文件的打开), 若成功的话, 返回一个句柄, 以后申请扩充内存, 就是通过此句柄。下面仅介绍申请扩充内存及内存块的移动。若对其它功能有兴趣的话, 可参见求伯君所著的《深入 DOS 编程》一书。

申请分配扩充内存块: 功能号是 09H。调用它之前设置 AH=09H, DX=所需内存的大小 (以 K 为单位), 若分配成功, AX 返回 0001H, DX 返回扩充内存的句柄; 若失败 AX 被置成 0000, BL 是错误代码 (参见修改部分的 5218H~5225H)。

移动扩充内存块: 是使用 XMS 的关键功能之一。该功能可以将指定大小的内存块在扩充内存之间或扩充内存与常规内存之间移动。调用它之前令 AH=0BH, DS:SI 指向一个移动结构表, 该表如下所示:

相对该表偏移量	大小	作用
00	双字	要移动的字节数
04	字	源句柄
06	双字	源内偏移量
0A	字	目的句柄
0C	双字	目的内偏移量

这里说明几点: 首先要移动的字节数必须是偶数, 其次, 要移动的字节数、源内偏移量、目的偏移量都是一个长整数 (有一个例外, 见下); 第三, 源内偏移量、目的偏移量, 都是相对句柄指向的扩充内存块的起点, 而非扩充内存的绝对位置; 第四, 若句柄置为 0, 则表示对常规内存移动, 偏移量不是长整数, 而是一个实际的地址 (段内偏移量, 段)。利用该功能我们可以在大量数据处理时, 把有用数据从扩充内存取出, 而将暂时不用数据放入到扩充内存, 这样可以方便地处理更大量的数据。

需要指出: 0AH, 是释放扩充内存块调用前 DX 中放入所要释放内存块的句柄。

利用 XMS 功能

使金山 DOS 与 DOS 5.0 兼容

北京大学 物理系 易鹤龄

还有一点要注意的是: 调用 XMS 功能前应保证有 256 字节的自由栈空间, 否则后果不可预料。

下面简述一下对金山 DOS 5.0 的 CHLIB.COM 的修改, 使之将字库放入扩充内存而不与 MS DOS 5.0 相冲突。CHLIB.COM 涉及到扩充内存使用的有二处: 4E09H-4E20H 检测扩充内存是否足够; 49F4H-4A26H 读扩充内存字库点阵及 4D21H-4D9BH 将盘中字库放入到扩充内存中去。结合附录给出的源程序与 XMS 的使用方法, 我们对其它版本的金山 DOS 也可依此方法修改。

附录: 给出以上所述的程序清单。

```

-u 4e09, 4e20
1B58:4E09 8EC0      MOVES, AX
1B58:4E0B BBFEFF    MOV BX, FFEF
1B58:4E0E 26        ES;
1B58:4E0F 803FFC    CMP BYTE PTR [BX], FC
1B58:4E12 750B      JNZ 4E1F
1B58:4E14 B488      MOV AH, 88
1B58:4E16 CD15      INT 15
1B58:4E18 3DEE00    CMP AX, 00EE
1B58:4E1B 7202      JB 4E1F
1B58:4E1D 33C0      XOR AX, AX
1B58:4E1F 07        POP ES
1B58:4E20 C3        RET

-u 49f4 4a26
1B58:49F4 81E27F7F  AND DX, 7F7F
1B58:49F8 81EA2121  SUB DX, 2121
1B58:49FC 80FE09    CMP DH, 09
1B58:49FF 7203      JB 4A04
1B58:4A01 80EE06    SUB DH, 06
1B58:4A04 8BC2      MOV AX, DX
1B58:4A06 B05E      MOV AL, 5E
1B58:4A08 F6E4      MUL AH
1B58:4A0A 32F6      XOR DH, DH
1B58:4A0C 03C2      ADD AX, DX
1B58:4A0E BA2000    MOV DX, 0020
1B58:4A11 F7E2      MUL DX
1B58:4A13 A31949    MOV [4919], AX
1B58:4A16 80C210    ADD DL, 10
1B58:4A19 88161B49  MOV [491B], DL
1B58:4A1D B91000    MOV CX, 0010
1B58:4A20 BE0749    MOV SI, 4907
1B58:4A23 B487      MOV AH, 87
1B58:4A25 CD15      INT 15

-u 4d21 4d9b

```

```

1B58:4D21 BEBE1D      MOV SI,1DBE
1B58:4D24 D1E6        SHL SI,1
1B58:4D26 8CC8        MOV AX,CS
1B58:4D28 B104        MOV CL,04
1B58:4D2A 8BD0        MOV DX,AX
1B58:4D2C D3E2        SHL DX,CL
1B58:4D2E D3C0        ROL AX,CL
1B58:4D30 240F        AND AL,0F
1B58:4D32 01161949    ADD [4919],DX
1B58:4D36 1400        ADC AL,00
1B58:4D38 A21B49      MOV [491B],AL
1B58:4D3B 56          PUSH SI
1B58:4D3C B90008      MOV CX,0800
1B58:4D3F 81FE0008    CMP SI,0800
1B58:4D43 7302        JNB 4D47
1B58:4D45 8BCE        MOV CX,SI
1B58:4D47 BAA851      MOV DX,51A8
1B58:4D4A D1E1        SHL CX,1
1B58:4D4C D1E1        SHL CX,1
1B58:4D4E D1E1        SHL CX,1
1B58:4D50 D1E1        SHL CX,1
1B58:4D52 B43F        MOV AH,3F
1B58:4D54 CD21        INT 21
1B58:4D56 BE0749      MOV SI,4907
1B58:4D59 8BC8        MOV CX,AX
1B58:4D5B D1E9        SHR CX,1
1B58:4D5D B487        MOV AH,87
1B58:4D5F CD15        INT 15
1B58:4D61 810621490080 ADD WORD PTR [4921],8000
1B58:4D67 8016234900  ADC BYTE PTR [4923],00
1B58:4D6C 5E          POP SI
1B58:4D6D 81EE0008    SUB SI,0800
1B58:4D71 7204        JB 4D77
1B58:4D73 7402        JZ 4D77
1B58:4D75 EBC4        JMP 4D3B
1B58:4D77 B43E        MOV AH,3E
1B58:4D79 CD21        INT 21
1B58:4D7B C70617492000  MOV WORD PTR [4917],0020
1B58:4D81 C7061F492000  MOV WORD PTR [491F],0020
1B58:4D87 8CCB        MOV BX,CS
1B58:4D89 B104        MOV CL,04
1B58:4D8B D3E3        SHL BX,CL
1B58:4D8D 891E2149    MOV [4921],BX
1B58:4D91 8CCB        MOV BX,CS
1B58:4D93 D3C3        ROL BX,CL
1B58:4D95 80E30F      AND BL,0F
1B58:4D98 881E2349    MOV [4923],BL
-d4907 4936
1B58:4900              00-00 00 00 00 00 00 00 00
1B58:4910 00 00 00 00 00 00 00 00-80 A8 51 00 93 00 00 00
1B58:4920 80 00 00 10 93 00 00 00-00 00 00 00 00 00 00 00
1B58:4930 00 00 00 00 00 00 00 00
-A4A11.4A23
1B58:4A11 MOV [4913],DX
1B58:4A15 MUL DX
1B58:4A17 MOV [4919],AX
1B58:4A1A MOV [491B],DL
1B58:4A1E MOV SI,4913
1B58:4A21 MOV AH,0B
1B58:4A23 CALL FAR [4907]

```

```

-A4E14.4E1A
1B58:4E14              CS:
1B58:4E15              CMP WORD PTR [490B],+01
1B58:4E1A              NOP
-A4D26
1B58:4D26              MOV [491B],CS
1B58:4D2A              JMP 4D3B
-A4D56
1B58:4D56              MOV SI,4913
1B58:4D59              MOV [SI],AX
1B58:4D5B              MOV AH,0B
1B58:4D5D              CALL FAR [4907]
1B58:4D61              ADD WORD PTR [491F],8000
1B58:4D67              ADC BYTE PTR [4921],00
-A4D7B
1B58:4D7B              MOV WORD PTR [491D],8000
1B58:4D81              MOV AX,[490B]
1B58:4D84              MOV [4917],AX
1B58:4D87              MOV [4921],CS
1B58:4D8B              WORD PTR [491F],0000
1B58:4D91              WORD PTR [491B],0000
1B58:4D97              NOP
1B58:4D98              NOP
1B58:4D99              NOP
1B58:4D9A              NOP
1B58:4D9B              NOP
-A5200
1B58:5200              MOV AX,4300
1B58:5203              INT 2F
1B58:5205              CMP AL,80
1B58:5207              JNZ 522E
1B58:5209              MOV AX,4310
1B58:520C              INT 2F
1B58:520E              MOV [4907],BX
1B58:5212              MOV [4909],ES
1B58:5216              PUSH CS
1B58:5217              POP ES
1B58:5218              MOV DX,00EE
1B58:521B              MOV AH,09
1B58:521D              CALL FAR [4907]
1B58:5221              CMP AX,0001
1B58:5224              JNZ 522E
1B58:5226              MOV [490B],DX
1B58:522A              MOV [491D],DX
1B58:522E              JMP 4C30
-A 100
1B58:0100              JMP 5200
-d 4907 4936
1B58:4900              00-00 00 00 00 00 00 00 00
1B58:4910 00 00 00 00 00 00 00 00-00 A8 51 00 00 00 00 00
1B58:4920 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
1B58:4930 00 00 00 00 00 00 00 00
-R CX
CX 5100
:5131
-W
Writing 05131 bytes
-Q

```

提取并修改或汉化程序中 英文文本信息的工具软件

新疆乌鲁木齐市第八中学 苏哲

为了修改或汉化程序中的用于显示的英文文本信息,通常的做法是使用 PCTOOLS 一类工具软件的编辑功能,将需要修改的程序打开,寻找需要修改的字符串,在编辑状态下,直接修改程序中的十六进制码或 ASCII 字符,这种操作方法比较麻烦,对于修改少量信息还可以,但对于信息量较大的修改或者是信息的汉化就无能为力了。

针对这一情况,本人用 C 语言编写了一个专用小软件,用于提取、显示、修改或者汉化可执行程序中的英文文本信息。运行该软件,可以将程序中的英文文本信息搜索并提取,以信息文件的形式记录下来,为便于查找和修改,在信息文件中,对每一条信息都自动编号。本软件利用 MS-DOS 提供的命令行编辑功能,对需要修改的信息进行编辑,如果是进行汉化,本软件可以在中文环境下运行,在软件的提示下,直接逐条用中文信息替换原来的英文信息。为了方便查询信息修改情况,本软件将程序原有的信息、修改后的信息和正在修改的信息同时显示,在退出运行时,保存原有信息和修改后的信息,以便下次查询和修改。

在修改或者汉化可执行文件时,为了不破坏原文件的结构,本软件对新写入的字符串长度与原字符串长度进行比较,自动截去超长部分,对长度不足部分,自动用空格填满。为了使修改或者汉化后的程序在运行时,屏幕显示的信息美观,本软件在添加空格时,还提示用户是否需要删除字符串尾部的显示空格。

本软件的命令行格式是:HHMSG [path]filename1 [path]filename2。[path]filename1 是需要修改或者汉化的文件的位置和名称;[path]filename2 是从原文件中提取的信息文件的位置和名称。

在运用本软件修改或者汉化程序中的英文文本信息时,有以下几点要十分小心:

1、在修改或者汉化可执行文件的文本信息时,要严格区分提取的信息中那些不是文本信息的字符,对这些字符的位置和大小写都不能改变,否则后果不可预料。

2、在修改或者汉化程序的文本信息时,必须按软件的提示正常退出,不能中途使用 CTRL+C 键强行终止,否

则可能会破坏被修改的程序。

本程序用 C 语言编写,在 TURBOC 环境下编译,在 PC-XT/AT 机上运行通过。本人用这个工具软件汉化了 MS-DOSV5.0 系统的全部文本提示信息:

```
/* HHMSG.C Designed by SuZhe 1992.12 */
/* ----- */
/* Usage: */
/* HHMSG name1 name2 */
/* name1 Specified the filename that you want to edit. */
/* name2 Specified the filename to save the text messages. */
/* ----- */
```

```
#include<stdio.h>
#include<dos.h>
#include<bios.h>
#include<string.h>
#include<ctype.h>
#include<stdlib.h>
#define MSG—LONG 126
void read—msg(void);
void change—msg(int);
void load—msg(int);
void save—msg(int);
void edit—msg(void);
struct message
{
    unsigned long addr;
    char textc[MSG—LONG+1],texte[MSG—LONG+1];
    unsigned int size,flag;
}msg;
FILE *infile,*outfile;
char infile[80],outfile[80];
short nmsg;
char str[MSG—LONG+1],chr;
unsigned long addr;
void main(int argc,char *argv[])
{
    short flag;
    if(argc>2){
        strcpy(infile,argv[1]);
        strcpy(outfile,argv[2]);
    }
    else{
        printf("\nSource file: ");
        gets(infile);
        printf("\nMessage file: ");
        gets(outfile);
    }
}
```

```

if((infp=fopen(infile,"r+b"))==NULL){
    printf("\nCannot open file: %s !\n",infile);
    exit(1);
}
if((outfp=fopen(outfile,"r+b"))==NULL){
    if((outfp=fopen(outfile,"w+b"))==NULL){
        printf("\nCannot open file: %s !\n",outfile);
        exit(1);
    }
    else flag=0;
}
else flag=1;
printf("\nPlease wait...\n");
if(!flag) read—msg();
else{
    fseek(outfp,0,SEEK—END);
    nmsg=(ftell(outfp)+1)/sizeof(struct message);
}
rewind(infp);rewind(outfp);
edit—msg();
fclose(infp);
fclose(outfp);
printf("\nGood Bye !\n");
}

void read—msg()
{
    register short a,b;
    chr=getc(infp);
    while(!feof(infp)){
        if(isprint(chr) && chr!=' '){
            addr=ftell(infp)-1;
            for(a=0,b=0;a<MSG—LONG && !feof(infp);a++){
                str[a]=chr;
                chr=getc(infp);
                if(chr==' ') b++;
                else b=0;
                if(!isprint(chr) || b>2) break;
            }
            str[a+1]=NULL;
            if(a>3){
                msg.addr=addr;
                strcpy(msg.textc,str);
                strcpy(msg.texte,str);
                msg.size=strlen(str);
                msg.flag=0;
                fwrite(&msg,sizeof(struct message),1,outfp);
                nmsg++;
            }
        }
        chr=getc(infp);
    }
}

void load—msg(int n)
{
    fseek(outfp,(long)(sizeof(struct message))* (long)n,SEEK—SET);
    fread(&msg,sizeof(struct message),1,outfp);
}

void edit—msg()
{
    short a,start,end;
    char c,num[10];
    printf("\n\n\n\n\n");

```

```

while(1){
    while(1){
        printf("\nThe first msg No. (1 to %d): ",nmsg);
        gets(num);
        start=atoi(num);
        if(start>0 && start<=nmsg) break;
    }
    while(1){
        printf("\nThe last msg No. (%d to %d): ",start,nmsg);
        gets(num);
        end=atoi(num);
        if(end>0 && end<=nmsg && end>=start) break;
    }
    for(a=start-1;a<end;a++){
        load—msg(a);
        change—msg(a);
        if(msg.flag) save—msg(a);
    }
    printf("\n\n\n\n");
    printf("\n\nDo you want to edit other messages (Y/N)? Y\b");
    c=toupper(bioskey(0));
    putchar(c);
    if(c=='N') return;
}

void change—msg(int a)
{
    short b,sz;
    printf("\n\n\nSource MSG  %03d: %s",a+1,msg.texte);
    printf("\n\nTarget MSG  %03d: %s",a+1,msg.textc);
    for(;;){
        printf("\n\nInput A New MSG: ");
        gets(str);
        printf("\n\nConfirm (Y/N)?");
        if(tolower(bioskey(0))!='n') break;
    }
    if((sz=strlen(str))!=0){
        strcpy(msg.textc,str);
        msg.flag=1;
        if(msg.size>sz){
            printf("\n\nDel the spaces at the end(Y/N)?");
            if(tolower(bioskey(0))=='y'){
                for(b=sz;b<msg.size;b+=2){
                    msg.textc[b]=' ';
                    if(b+1<msg.size)
                        msg.textc[b+1]='\b';
                }
            }
        }
        else{
            for(b=sz;b<msg.size;b++){
                msg.textc[b]=' ';
            }
        }
    }
    else msg.flag=0;
}

void save—msg(int n)
{
    fseek(outfp,(long)(sizeof(struct message))* (long)n,SEEK—SET);
    fwrite(&msg,sizeof(struct message),1,outfp);
    fseek(infp,msg.addr,SEEK—SET);
    fwrite(msg.textc,sizeof(char),msg.size,infp);
}

```

再谈单显作图问题

◇ 四川省武隆县教委信息站 刘蓉丰 ◇

《电脑》杂志 93 年 1 期和 4 期“大学生之页”栏两次刊登了单显微机作图问题。确切地说，是单显模拟 CGA 模式 4 的问题。第 1 期文章有明显错误；第 4 期文章称能产生上下两幅相画面，如同使用两台显示器一般，很不理想。为解决这一问题，本人用 MASM 5.0V 编写了下面的程序，先将 BIOS 的配置单元改成 CGA，再用 INT 10H 置成模式 4，最后根据 CGA 卡的参数要求 M6845 的方式寄存器设置进行修改，并根据 CGA 卡的参数要求重新编程，取得了满意的效果。除了画面只比屏幕稍小一点外，绝无闪烁、重影、图像拉断，同时出现两幅画面等问题。本人还试玩了吃豆子，打飞机、天外来客……等游戏，画面音响均正常。需指出的是，若将送往 M6845 寄存器的第 1 个控制参数改为上述两文中使用的 35H 或 38H，就会出现两幅图象，且有重影、闪烁。

MDA 卡已很少见，但使用 HGC 卡的高分单显则在国内有相当数量，由于 HGC 卡高度兼容 MDA 卡，故本程序也可使高分单模拟 CGA 模式 4，从而使许多不能在 HGC 卡上运行的图形软件能够顺利运行，经实际运行验证，效果也非常好，这给拥有高分单显的朋友确实带来方便。

本程序经汇编连接成 .exe 文件执行，不能转为 .com 文件。

附源程序如后：

```
.MODEL small
.STACK 100h
.DATA
cga db 40h,28h,2dh,0ah,7fh,06h,64h,70h,2,1,6,7,0,0,0,0
.CODE
main PROC far
push ds
xor ax,ax
push ax
mov ax,40h
mov ds,ax
mov ax,ds:[10h]
and ax,00cfh
or ax,20h
```

```
mov ds:[10h],ax
mov ax,0004h
mov ax,@DATA
mov ds,ax
mov al,3
mov dx,3bfh
out dx,al
push es
push ds
mov al,2
mov dx,3b8h
out dx,al
push ds
pop es
lea si,cga
mov dx,3b4h
mov cx,10h
xor ax,ax
lp:
out dx,al
inc dx
lodsb
out dx,al
inc ah
mov al,ah
dec dx
loop lp
cld
mov cx,2000h
mov ax,0b800h
mov es,ax
xor di,di
xor ax,ax
rep stosw
mov al,8ah
mov dx,3b8h
out dx,al
pop es
pop ds
ret
main ENDP
end main
```

几种图象文件格式 浅析

铁道部株洲电力机车厂技校 王大银

目前较流行的图形文件格式有 PCX、GIF、BMP、TIF、IMG 等,下面分别就 PCX、GIF、BMP 格式的文件头作一个介绍。其中 BMP 为 Windows 专用格式。另外,PCX 采用同值长度编码方案压缩,GIF 采用 LZW 压缩方案压缩。

一、PCX 格式文件表头:

偏移	字节数	内容
00H	1B	0AH, PCX 文件标志
01H	1B	0,2,3,5 为 PCX 文件版本号(5 表示 256 色图象)
02H	1B	01H, 用同值长度编码压缩(0 表示未压缩)
03H	1B	每个像素的位数,其加一后的 2 的幂为颜色数
04H	2B	图象最小宽度(一般为 0)
06H	2B	图象最小高度(一般为 0)
08H	2B	图象最大宽度(像点数)
0AH	2B	图象最大高度
0CH	4B	图象设备分辨率
10H	48B	16 色调色板值(按红绿蓝顺序排列)
40H	1B	0
41H	1B	位平面数
42H	2B	每行图象数据所需字节数
44H	1B	为 1 则按灰度比例显示,为 2 则按彩色显示
45H	59B	保留
80H	?	图象数据
-301H	1B	0CH, 256 色调色板标志标记
-300H	768B	256 色调色板

注:(1)第 03H 字节表示颜色数,例如:第 03H 字节若为 7,则颜色数等于 $2^{(7+1)} = 256$

(2)每行像点数(第 08H 字节)和每行图象所需字节数并不相等,且每行实际像点数 = 第 08H 字节 + 1

(3)对 256 色文件,第-301H 字节(从文件尾部开始倒数第 769 字节)必需为 0CH,否则是一错误的图象文件。

(4)-300H 开始的调色板数据应向右移 2 位。

二、GIF 格式 256 色文件表头

偏移	字节数	内容	
00H	3B	"GIF", GIF 文件标志	整体数据
03H	3B	版本号,常为"87a"	
06H	2B	屏幕宽度	
08H	2B	屏幕高度	
0AH	1B	颜色数和整体标志	
0BH	1B	背景颜色号	
0CH	1B	0,表示表头(整体数据)结束	局部数据
0DH	768B	调色板数据(16 色则为 48B)	
30DH	1B	逗号,表示这是一图象块	
30EH	2B	图象在屏幕左边的位置	
310H	2B	图象在屏幕顶部的位置	
312H	2B	图象宽度	
314H	2B	图象高度	
315H	2B	颜色数和局部标志	
316H	?	图象数据	

注:(1)第 0AH 字节低三位加 1 表示整体颜色数(2 的幂),若该字节最高位(7 位)为 1,表示有整体调色板数据。

(2)同 PCX 格式一样,第 0DH 开始的调色板数据需右移 2 位才得到真实调色板数据。

(3)第 315H 字节含义同解释(1),但第 7 位若为 1,则图象的数据是按交替方式存储,即第 1 行数据对应屏幕第 1 行,第 2 行数据对应屏幕第 8 行等。

(4)第 30DH 字节若为"! ",则表示扩充号,";"则定义为结束。

(5)这个 GIF 格式假设一个 GIF 文件只包含一幅图象,即只有一个局部数据块。

三、BMP 格式文件表

偏移	字节数	内容
00H	2B	"BM", BMP 文件标志
02H	4B	BMP 文件尺寸(字节)
06H	2B	0,保留
08H	2B	0,保留
0AH	4B	图象数据起始位置
0EH	4B	28H,表头长度
12H	4B	图象宽度
16H	4B	图象高度
1AH	2B	1,图象设备位平面数
1CH	2B	每个像素的位数(1,2,4,8,24)
1EH	4B	0,1,2 压缩方式
22H	4B	图象大小
26H	8B	图象设备水平、垂直分辨率
2EH	4B	实际使用的颜色总数
32H	4B	重要的颜色数
36H	?	调色板数据(蓝,绿,红,0,蓝……)

这些图象格式现已为越来越多的图形软件所使用,并已成为图象文件格式的标准,对这些格式的识别,将给我们使用或编写图形软件提供极大的帮助。

搜索回溯原理及它的递归实现

韶关大学

王逸明

一、引言

在人工智能领域中,大多数问题都可归入非计算性问题一类。解决这类问题的最有效手段是搜索。人们已经对搜索技术进行了广泛深入的研究,得到了很多好的搜索方法,如深度优先搜索,广度优先搜索,启发式搜索,双向搜索等。在人工智能语言的发展中,被看作比 LISP 更为高级的逻辑编程语言 PROLOG 系统就是基于模式匹配和搜索回溯的设计原理,而后者正是深度优先的搜索回溯。

在一般情况下,深度优先搜索(搜索回溯)是较好的搜索方法。

在结构化的程序设计语言中,允许使用递归。例如在 C 语言中,如果一个函数内部的语句调用了函数自己,则称这个函数是递归的。递归函数的一个简单例子就是计算整阶乘的函数 `factr()`:

```
factr(int n)
{
    if(n==1)return(1);
    return(factr(n-1)*n);
}
```

当我们用 `factr(10)` 调用函数 `factr()` 时,函数返回 `factr(9)*10`,因而引起下一次用 9 对自身的调用,一直到用 1 调用此函数后,递归返回。

通过对搜索回溯的原理分析,我们发现它实际上也是一个递归过程,因此可以把这两种程序设计方法结合起来,即用递归实现搜索回溯,这是一种很有启发性的方法。

二、搜索回溯原理

一般可用树的中序遍历(inorder traversal)来说明搜索回溯的原理。我们在这里通过一个具体实例来分析搜索回溯的过程。

图形覆盖问题:下面有 12 个顺序编号的块(图 1),每块都是由 5 个单位面积的正方形(称为格)邻接而成。能否只通过平移,将这 12 块拼成一个高为 6,宽为 10 的矩形?图 1 给定的 12 块及编号。

为解决这个问题,设有一个 6×10 的矩形,它被划分为 60 个格。我们每一步选择一个适当的块,并把它放到矩形中恰当的位置,如此经过 12 步就完成了对矩形的覆盖。

为了使每一块都能放到恰当的位置,我们为每一步选择一个覆盖目标,这个目标就是矩形中按先上后下,自左向右的顺序第一个未被覆盖的格。我们用所选择的块去覆

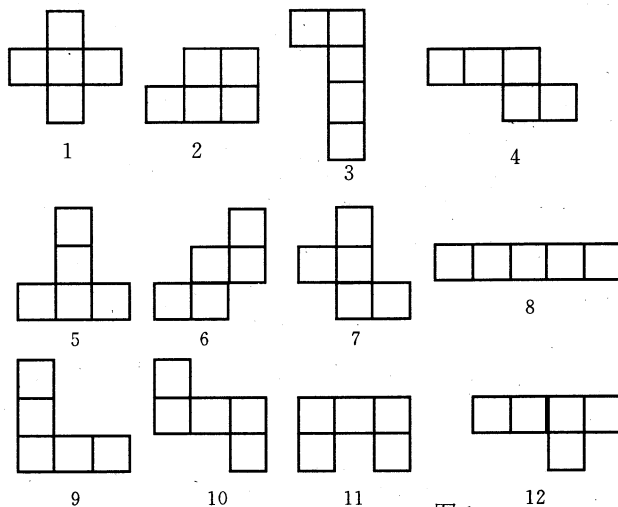


图 1

盖目标时,应使该块的最上方靠右的格恰好对准目标,而该块其余的格也全都对准矩形中未被覆盖的格。

我们使用搜索回溯的方法,并假设通过前面的 $n-1$ 步,已经有 $n-1$ 个块放到了恰当的位置。下面是进行第 n 步的过程(6 个步骤):

1. 确定第 n 步的覆盖目标。
2. 选择第一个搜索方向(即未就位的编号最小的块)。
3. 如果可以覆盖目标,则将该块就位,然后转到下一步。
4. 如果不能覆盖目标,则判断是否还有下一个搜索方向(即未就位的编号较大的块)。
5. 如果选择了下一个搜索方向,则转步骤 3。
6. 如果不能再选择下一个搜索方向,则回溯到上一步,将上一步就位的块恢复,并判断是否还有下一个搜索方向,然后转步骤 5。

下面是用搜索回溯法解题的程序框图(图 2):

三、搜索回溯的递归实现

从以上的分析中我们发现,在搜索回溯过程中,每向前搜索一步都是重复上述的 6 个步骤,而步骤 3 中的“转向下一步”,正是建立递归关系的基础,我们使用允许递归的 Pascal 或 C 语言,就可能用递归的程序结构来实现搜索回溯过程了。

这样设计在递归函数中将包含有回溯,因此递归的次数是不确定的。根据搜索要求,递归可以在第一次达到目标后结束,也可以再次回溯以进行穷尽搜索,此时递归在

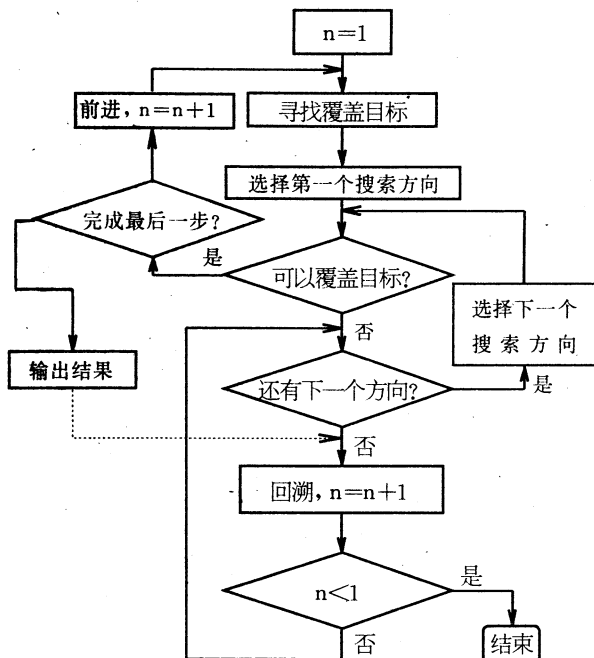


图2 搜索回溯方法程序框图

从第1步的最后一个搜索方向回溯后结束。

下面我们给出图形覆盖问题的C程序,作为用递归实现搜索回溯的一个实例。

程序中每个块的形状用五个格的坐标来描述,第一个格即最上方靠右的格的坐标为(0,0),并以此确定其余四个格的坐标。

主函数 main()中只有一个语句 find(0)。

find()函数正是一个递归函数,其功能是完成整个搜索回溯过程中的一步,并通过自我调用转到下一步。

在 find()函数中,还先后调用了下面三个函数:

pass()函数检验所选的块是否能覆盖目标。

next()函数选择下一个搜索方向,必要时进行回溯。

pr()函数打印出覆盖结果。

为简单起见,我们只允许对每个块进行平移,此时对 6×10 的矩形进行覆盖只有一个解。如果允许对每个块进行旋转和反射,则需对程序进行扩充,增加搜索方向。此时对 $6 \times 10, 5 \times 12, 4 \times 15$ 的矩形进行覆盖均有多个本质不同的解,对 3×20 的矩形也有1个解。当然,需要对程序进行扩充修改。

我们发现,用递归实现搜索回溯的源程序,确实具有简洁,清晰的特点。同时,它也形象化地体现了搜索回溯的过程。

以下是用 TurboC2.0 写的源程序。

```
#define ROW 6
#define COL 10
#define SIZE 5
#define NUM 12
```

```
int mark[NUM],order[NUM],board[ROW][COL],x[NUM][SIZE],y
[NUM][SIZE];
int block[NUM][SIZE][2]={
0,0, 1,-1, 1,0, 1,1, 2,0,
0,0, 0,1, 1,-1, 1,0, 1,1,
0,0, 0,1, 1,1, 2,1, 3,1,
0,0, 0,1, 0,2, 1,2, 1,3,
0,0, 1,0, 2,-1, 2,0, 2,1,
0,0, 1,-1, 1,0, 2,-2, 2,-1,
0,0, 1,-1, 1,0, 2,0, 2,1,
0,0, 0,1, 0,2, 0,3, 0,4,
0,0, 1,0, 2,0, 2,1, 2,2,
0,0, 1,0, 1,1, 1,2, 2,2,
0,0, 0,1, 0,2, 1,0, 1,2,
0,0, 0,1, 0,2, 0,3, 1,2};
main()
{int find(),pass(),next(),pr();
find(0);
}find(int i)
{int m,n;
order[i]=0;
for(m=0;m<ROW;m++)
for(n=0;n<COL;n++)
if(board[m][n]==0){x[i][0]=m;y[i][0]=n;goto begin;}
begin:
while(pass(i)==0)i=next(i);
if(i==NUM-1)pr();
if(i<0)exit();
find(i+1);
}
pass(intp)
{int k,a,b;
if(mark[order[p]]==1)return 0;
for(k=1;k<SIZE;k++)
{a=x[p][k]=x[p][0]+block[order[p]][k][0];
b=y[p][k]=y[p][0]+block[order[p]][k][1];
if(a<0||a>=ROW||b<0||b>=COL||board[a][b]>0)return 0;
}
for(k=0;k<SIZE;k++)board[x[p][k]][y[p][k]]=order[p]+1;
mark[order[p]]=1;
return 1;
}
next(intp)
{int k;
for(k=order[p]+1;k<NUM;k++)if(mark[k]==0){order[p]=k;return p;}
p--;
for(k=0;k<=SIZE;k++)board[x[p][k]][y[p][k]]=0;
mark[order[p]]=0;
}
}pr()
{int m,n;
printf("\nTheboardis:\n");
for(m=0;m<ROW;m++)
{for(n=0;n<COL;n++)
{textcolor(board[m][n]);
cprintf("%2d",board[m][n]);
printf("\n");
}
}
```



DEC 公司与广东“京粤” 携手合作发展电脑产业

[本刊讯]世界著名计算机公司美国电脑数据设备公司(英文: Digital Equipment Corporation)主管商用计算机生产及全世界市场销售副总裁 Weiss 先生一行,于3月初抵达广州进行市场调查,了解广东市场的潜力和中国高科技及商用计算机产业发展前景,并商讨进一步与广东京粤电脑中心合作,在广东建立合资、合作企业。随同 Weiss 先生前来访问考察的有 DEC 公司负责中国大陆、香港、台湾业务的总经理陈绍升先生,负责亚洲区零售、批发业务的商务经理冯羽军先生和 DEC 中国有限公司副总经理赵国豪先生等。

DEC 公司在广州活动时间,全部由京粤电脑中心安

排。京粤电脑中心谢太俭总裁等全程陪同并安排在广州的活动,还将安排贵宾去北京参观京粤收款机的用户,进一步落实合作事宜。

美国 DEC 公司是世界大计算机公司,有举足轻重地位。DEC 公司生产各种大、中、小、微型计算机及网络产品,在商业领域方面的计算机及收款机产业也很著名。京粤电脑中心同 DEC 公司已有两年多的合作关系,该中心是 DEC 公司小型计算机和微机计算机的增值代理,和收款机中国南方总经销,广州市赛马场和山西电子系统的计算机管理工程,也是京粤电脑中心同 DEC 公司的合作项目之一。

DEC 公司副总裁此次前来广东考察,显然注意到中国大陆市场潜力,并为其将来推广 POS 系统战略的需要,希望选择在广东投资设厂,并与京粤电脑中心就商用收款机方面建立合资、合作企业。

DEC 公司副总裁一行还受到广东省、广州市领导的接见。

132

alarm manager 报警管理器

alarm transfer 报警传输

ALE (Asian Language Environment) 亚洲语言环境

ALGaAs 砷化铝镓

algorithmic state machine (ASM) 算法状态机

alias 别名,亦称,又叫

alignment 对位,定位,调整,对准

alphanumeric 字母数字的

alphanumeric pager 显示字母数字的无线寻呼机

alpha test α 测试,内部测试

ALS (Automatic Location System) 自动定位系统

alter image in window 窗口中的交变图象

alternate 交替,更迭,轮流,相同,区别

alternative 替代

alternative algorithms 交替算法

alter switch 变更开关

alter window 交变窗口

ALU (Arithmetic and Logic Unit) 算术逻辑部件,算术逻辑单元

aluminium electrolytic capacitor 铝电解电容器

AM (Address Maintenance code) 地址修改码

AM (Application Model) 应用模型

小辞典

AMAE (Application Management Application Entity) 应用管理应用实体

AMAP (Application Management Application Process) 应用管理应用进程

ambient 环境光,环境

ambiguity 歧义,多义性;含糊,模糊点

ambiguous phrase of combination type 多义组合型歧义切分字段

ambiguous phrase of overlap type 交叠型歧义切分字段 CA (Apple Media Control Architecture) 苹果媒体控制体系结构

AME (Advanced Modeling Extend) 先进建模扩展

进建模扩展

AME (Advanced Modeling Extension) 先进建模扩充,高级建模[实体模型]扩展

American federation of information processing science (AFIP) 美国信息处理科学联合会

American federation of information processing society (AFIPS) 美国信息处理联合会

American national standards institute (ANSI) 美国国家标准学会

American national standards institute / systems planning and requirements committee (ANSI / SPARC) 美国国家标准学会 / 系统计划与要求委员会

133