

令进行特殊处理的过程,该功能类似于屏幕设备驱动程序 ANSI.SYS。UCDOS V3.1 的特殊显示功能提供了相当丰富的功能:作图、图像操作、后台演奏和一些系统命令的转换等。利用这些功能,程序员可以方便的实现各种图像、后台音乐和图文操作。UCDOS V3.1 领先于其它中文 DOS 平台的技术之一就是:UCDOS V3.1 支持扩展显示模式的直接写屏功能,也就是说用户在 UCDOS V3.1 状态下,不仅可以得到丰富的色彩、超高的分辨率,而且还可以实现 24 点阵的汉字显示及在同一屏幕上完成汉字及 256 色图像的显示,这样的话 UCDOS V3.1 就可以部分地替代一些图文数据库的功能,而且希望公司也希望自己的产品能起到这种效果,他们在 UCDOS V3.1 的用户手册及 README 中都提供了一个用 FoxBASE 编写的人事档案查询程序,以进一步指导用户对特殊显示功能的认识。

的确,UCDOS V3.1 的特殊显示功能是一个相当完善、完善的功能,利用它用户可以在自己的程序中实一些编程语言无法完成或较难完成的功能。笔者在使用 UCDOS V3.1 的特殊显示功能时,发现了一些用户应该注意的事项:

1.使用特殊显示功能有两种方法:一是缺省方法;二是通过打印口 3,笔者推荐大家使用第二种方法去调用特殊显示功能(UCDOS V3.1 本身提供的程序也是使用了这种调用方法),这是因为现在流行的软件大部分是西文直接写屏的,如 C++、FOXPRO 等等,缺省的特殊显示格式肯定已经无法满足许多软件的需求了,所以说第二种方法更可靠;

2.确定自己的机器的显示内存,确保它们有足够的内存去支持扩展显示模式;

3.必须在 CONFIG.SYS 文件中加入 QEMM . SYS。

四、使用过程中的常见问题

1.无法正常启动 UCDOS。首先检查自己的机器查阅 UCDOS V3.1 的支持机型中是否有此类型的微机,不过一般来说,UCDOS V3.1 可以支持任何类型的机,包括长城机器,所以用户应该把目标转向 UCDOS V3.1 的本身配置或参数。

2.某些软件无法运行。如某些中文软件是针对某个

特定的汉字系统编制的;某些西文软件有自带的西文字库(如 AUTOCAD),不能直接使用汉字;某些西文软件(如 PCSHELL V8.0、NORTON V7.0)直接修改了 EGA 和 VGA 的英文字模,应禁止它们修改字模集:PCSHHELL / NF、NORTON / GO。

3.某些功能无法实现。首先检查自己的硬件与 UCDOS V3.1 中的配置是否相符;另外在可能的情况下优先选择使用北京希望电脑公司为 UCDOS V3.1 专门推出的扩充内存驱动程序—QEMM.SYS。

* * * * *

用 C 语言实现在 GRAPHTOOL 中使用汉字

杨 军 (中国运载火箭技术研究院)

GRAPHTOOL 是一个用于高级图形分析的功能强大的二、三维图形处理软件,它具有集成的处理和表达方式,可以处理 13 种基本图形和 26 种图形类型,并允许对任一种图形进行数据分析和元素编辑,可供要求高级图形分析的用户使用。

在 GRAPHTOOL 的图形编辑过程中,一般需要添加注解(如标题、轴标号等说明性文字),用以说明该图形的名称和用途,尤其是当一个图形文件中含有多个图形时,必须使用注解来区分不同的数据。但是,同大多数西文软件一样,GRAPHTOOL 不支持汉字的显示,只能添加西文注解。笔者在使用该软件的过程中,发现其中的散布图将数据文件中的点绘出,而在数据点之间不连接任何线段,这样,若能把汉字的点阵信息转换成 GRAPHTOOL 中的数据文件,便能利用散布图绘制所需要的汉字了。散布图是由基本图形 X-Y 图的数据文件由数据头和数据块两部分组成:数据头是数据文件的注释部分,每行以反斜杠“\”开始;数据块的第一行为用于绘图的数据点的数目及维数,其后便是数据矩阵。下面是一个 X-Y 图的数据文件的例子:

```
\MyPlot.data
\X—Time(s),Y—Height(m)
200      2
0.0      0.0
0.1      524.8
;        ;
```

读取点阵汉字库需知道汉字区位码与机内码、区位码与 DOS 记录位置之间的关系,这种关系可以通过观察已知区位码的汉字字符串的 ASCII 码来获得。比如读取启明星汉字系统中的 16×16 点阵字为 HZKXSJ 时,很容易测得下面的关系式:

QM = HZ[0]-160

$$WM = HZ[1] - 160$$
$$\text{LOCAL} = (\text{QM}-1) * 94 + (\text{WM}-1)$$

其中,QM 和 WM 是汉字的区码和位码,HZ[0]和 HZ[1]是汉字机内码的高位字节和低位字节,LOCAL 是汉字点阵在文件中的记录位置。下面的 C 语言源程序将用户输入的汉字字符串转换成 X-Y 图的数据文件,本程序在 Turbo C2.0 下运行通过。

/ * GTHZ.C将汉字点阵信息转换成 * /
 / * GRAPHTOOL 中的X-Y图数据文件 * /
 / * 中国运载火箭研究院一部五室 * /
 / * 杨军 一九九四年四月 * /

```
#include <stdio.h>
#include <string.h>
```

```
main(int argc,unsigned char * argv[])
```

```
FILE * fpin, * fpout;
```

```
int i,j,x,y;
```

```
itn ByteLoop, BitLoop;
```

```
unsigned int qm,wm,M = 0;
```

unsigned long int local;

```
unsigned char hzd[32],point;
```

```
if (argc != 3)
```

```
{
printf("Notice: You must use your file name\n");
printf("and needed chinese words\n");
printf("Examp1: gthz myhz.dat 汉字使用示例\n");
exit(0);
}
```

```
if((fpin = fopen("hzkxsi","rb")) == NULL)
```

```
{printf("Can not open hzk file!!");exit(0)}
```

```
if((fpout = fopen(argv[1], "w")) == NULL)
```

```
{printf("Can not open data file!!");exit(0)}
```

```
fprintf(fpout, "\n");
```

```
for(i = 0; i < strlen(argv[2]); i += 2)
```

```

{
    if( * (argv[2]+i) == 0) break;
}

```

```
qm = * (argv[2])+i-160;
```

```
wm = * (argv[2]+i+1)-160;
```

```
local=0; /* 计算汉字点阵信息位置 */
```

```
for(j=0;j<32;j++) /* 并读取字模 */
```

```
local += ((qm-1) * 94 + wm-1);
```

```
fseek(fpin,local,0):
```

```
fread(hzd,1,32,fpin);
```

```
for(ByteLoop = 0;ByteLoop < 32;ByteLoop++)
```

```
{
point = 0x80
```

```
for(BitLoop=0;BitLoop<8;BitLoop++)
```

```

{
    f((hzd[ByteLoop]&point)!=0)
    {
        x = 10 * i + 8 * (ByteLoop%2) + BitLoop; /* 转换成 X,Y
坐 * /
        y = 16 - ByteLoop / 2; /* 标并写入文件 * /
        fprintf(fpout, "%d %d\n", x, y);
        M++;
    }
    point = point > 1;
}
}
}
fclose(fpin);
fseek(fpout, 0, 0);
fprintf(fpout, "%d 2", M);
fclose(fpout);
}

```

生成了数据文件后,就可以在所编辑的图形中添加注解了。调出需要添加注解的图形后,选择 Add\Graph\X-Y Plot,根据对话框的提示键入汉字数据文件名,并把线型参数设置为 0(即绘制散布图)即可。绘出汉字后,用 Change 功能改变坐标轴、重框和网络的颜色为 0,隐去不需要的元素,并调节其大小、位置,直到满意为止。

* * * * *

一个 C 编程中屏幕录入的 实用程序

田 军 彭文澍 (青島市工商银行电脑部)

C 语言是当今计算机领域中最普遍使用的机器语言之一。但是单纯用 C 语言编写程序往往会遇到一个麻烦的问题,即屏幕录入数据的处理。

在多个录入项同时录入时,用 C 编程会有许多不足之处,其中最大的问题是在录入错误时无法有效地运用上、下、左、右键和退格键返回到上面已录入过的数据项进行修改,然而这种全屏幕编辑修改在其它一些高级语言中却是可以实现的。C 语言编程中解决此类问题没有一个确定的方法可循,较多的是利用具体问题的特殊性来处理录入过程,或全部重新录入。这使数据录入成为一项十分艰苦的工作。

本文探讨了解决这问题的一条有效途径,并编制了