

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:

$$QM = HZ[0] - 160$$

$$WM = HZ[1] - 160$$

$$LOCAL = (QM - 1) * 94 + (WM - 1)$$

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;
    int 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("Exampi:gthz myhz.dat 汉字使用示例\n");
        exit(0);
    }
    if((fpin=fopen("hzkxsj","rb"))==NULL)
    {
        printf("Can not open hzk file!!\n");exit(0);
    }
    if((fpout=fopen(argv[1],"w"))==NULL)
    {
        printf("Can not open data file!!\n");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; /* 转换为 X,Y
坐 */
        y = 16-ByteLoop / 2; /* 标并写入文件 */
        sprintf(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(即绘制散点图)即可。绘出汉字后,用 Change 功能改变坐标轴、重框和网格的颜色为 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 (青岛市工商银行电脑部)

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

一个一般性的数据录入程序 scrstr()。该函数能自动识别来自键盘的上、下、左、右键及退格键，结合调用程序中简单明了的定位方法，就可以实现 C 编程中的全屏幕编辑修改功能。

scrstr()函数通过判断键入的上、下、左、右和退格键代码来识别键盘。这些特殊的键代码对应如下：

- ↑ ESC [A
- ↓ ESC [B
- ESC [C
- ← ESC [D
- 退格 0X08

当获得左、右、退格键代码时，做出相应的动作；当获得上移键时退出函数并返回 1；当获得回车键或下移键时退出函数并返回 0。

scrstr()函数必须有 curses 函数库的支持，因为库函数 curses 的初始化屏幕函数 initscr()使得 getchar()能读取一系列的单个字符，而不需要等待回车键。

为了说明函数 scrstr()的具体应用，笔者编制了一个 try.c 程序调用 scrstr()函数。它判断 scrstr()函数的返回值，以使光标移动到相应的录入项。上述程序 try.c 和 scrstr()已在 XENIX V 系统下编译运行通过，使用效果良好。

```
/* try.c */
#include <stdio.h>
#include <curses.h>
int j,k;
char vol1[9],vol2[9],vol3[9],vol4[9];
main()
{
short x,y;
for(j=0;j<9;j++)
{vol1[j]='^\0';vol2[j]='\0';vol3[j]='^\0';}
system("clear");
x=10;y=10;
initscr();
j=1;
while(j>0)
{switch(j)
  case 1:k=scrstr(10,10,"aaaa:",vol1,6);
    if(k==1)j=j-2;break;
  case 2:k=scrstr(12,10,"bbbb:",vol2,2);
    if(k==1)j=j-2;break;
  case 3:k=scrstr(14,10,"cccc:",vol3,3);
    if(k==1)j=j-2;break;
  case 4:k=scrstr(16,10,"dddd:",vol4,8);
    if(k==1)j=j-2;break;
  default:j=-1;break;
}j++;
endwin();
printf("\n vol1=%s \n vol2=%s \n vol3=%s \n vol4=%s
```

```
\n ",vol1,vol2,vol3,vol4);
}
/* scrstr.c */
scrstr(lin,col,pristr,instr,n)
int lin,col,n;
char pristr[],instr[];
{
int i,j,len;
char c;
char vol[81];
for(i=0,i<81;i++) vol[i] = '^0';
printf("\x1b[%d;%dH",lin,col);
printf("%s ",pristr);
len = strlen(pristr)+1;
strcpy(vol,instr);
printf("%s",vol);
printf("\x1b[%d;%dH",lin,col+len);
i=0;
while((c=getchar())!=^r)
{
  if(c==0x08)
    if<i<0){i--;
      j=i;
      while(j<n+1){vol[j]=vol[j+1];
        if(vol[j]=='^0'){printf("");
          else{printf("%c",vol[j]);
            j++;}
        printf("\x1b[%d;%dH",lin,col+len+i);
        continue;}
      else{printf("");
        continue;}
    }
  if(c==0x1b)
  { c=getchar();c=getchar();
    switch(c)
      case 0x41:printf("\x1b[%d;%dH",lin,col+len);
        printf("\x1b[K");
        strncpy(instr,vol,n);
        printf("%s",instr);
        return(1);
      case 0x42:printf("\x1b[%d;%dH",lin,col+len);
        printf("\x1b[K");
        strncpy(instr,vol,n);
        printf("%s",instr);
        return(0);
      case 0x43:if(i<n&&vol[i]=='^0'){i++;}
        else{printf("\b");}
        continue;
      case 0x44:if(i<0){i--;}
        else{printf("");}
        continue;
    }
  if(i<n){vol[i]=c;i++;}
  else{printf("\b");printf("");printf("\b");}
}
printf("\x1b[%d;%dH",lin,col+len);
printf("\x1b[K");
strncpy(instr,vol,n);
printf("%s",instr);
return(0);
}
```