

再谈 C 编程中屏幕录入的 实用程序

田 军 (青岛市工商银行)

作者在《计算机系统应用》1995 年第 4 期上发表一文《一个 C 编程中屏幕录入的实用程序》(以下称前文)。文中设计了一个通用函数 Scrstr(),此函数结合调用程序 try.c 中给出的定位方法,可以一般性地完成多个字符串同屏录入时的全屏幕编辑修改功能。之后,陆续收到几封来信,探询对 C 语言中字符型数据以外的其它类型数据的处理办法。作者首先感谢各位同行对文章的厚爱,并在此进一步讨论各种基本类型数据的屏幕录入问题,敬请指教。

前文中单对字符型数据进行了较详细的讨论。全面地说,屏幕录入函数应是一组函数,即 C 语言中每种数据类型均有一个对应的类似 Scrstr()的函数,当遇到要录入某种类型的数据时取用对应的函数。然而在这组函数中以对应字符型数据的函数 Scrstr()为最基本的函数,对应其它数据类型的函数都可以以此函数为基础用限制录入字符和类型转换来取得。

C 语言中的基本数据类型很多,但大致可以划分为四类:字符型、无符号型、整型和浮点型。其中字符型就是前文中讨论的内容,无符号型是有符号型的特例,而整型中以长整型,浮点型中以双精度型最具有代表性,所以本文以对应长整型的函数 Scrln()和对应双精度型的函数 Scrdou()为例进行讨论。

1.如附程序所见,Scrln()是在 Scrstr()的基础上经过三点修改得到的:

(1)形式参数中 *inlon 为长整型,它取代了字符数组形式参数 instr[]和数组长度 n。在录入前 *inlon 的内容没有被赋予内部字符数组 vol,因而不能象字符型数据那样可以对某一位数进行修改,如果修改需重新输入整个数。

(2)函数对键盘的输入做了限制,只允许输入十、一和 0~9 这 12 个符号,它们在 ASCII 码表上的位置如下:

+ 43
- 45

(3)函数在每一个出口处用库函数 atol()将内部字符数组 val 的内容转换成整型的数 *inlon。

2.函数 Scrdou()与 Scrln()类似,也要对函数 Scrstr()做三点修改:

(1)以双精度型形式参数 *indou 取代 instr[]和 n。

(2)在允许输入+、-和 0~9 的同时,还允许从键盘输入小数点,小数点在 ASCII 码表上的位置为 46。

(3)在每个出口处用库函数 atof()将 val 的内容转换成双精度型数 *indou。(作者在调试中发现库函数 atof()用起来有不尽人意之处,故编制了一个函数 atofd()用来将字符数组转换成浮点数或双精度数)。对于其它类型的数据均可相仿地一一写出对应的函数,由此即可得到一组完整的屏幕录入实用函数。作者在银行会计程序中使用了这组函数,在 XENIX V 环境下运行,收到了予想的效果,受到用户好评。为便于验证,特意设计了一个实验程序 try3.c。

最后,在前文和本文中皆最小限度地利用了功能键。如果需要,可以扩展利用更多的功能键以达到更加方便有效的实用效果。

```

/* try3.c */
#include <stdio.h>
#include < curses.h>

int j,k;
char vol1[11];
long vol2;
double vol3;
main()
{
    int x,y;
    system("clear");
    x = 10; y = 10;
    for(j = 0; j < 11; j++) vol1[j] = ' / 0';
    strcpy(vol1, "abcd");
    vol2 = 1234;
    vol3 = 567.89;
    initscr();
    j = 1;
    while(j > 0)
        {switch(j){case 1:k = scrstr(x,y,"VOL1:",10);
                    if(k == 1)j = j-2;break;
                    case 2:k = scrln(x+2,y,"VOL2:",&vol2);
                    if(k == 1)j = j-2;break;
                    case 3:k = scrdou(x+4,y,"VOL3:",&vol3);
                    if(k == 1)j = j-2;break;
                    default:j = -1;break;}}
}

```

```

j++;}
endwin();
printf(" / n vol1 = %s / nvol2 = %d / n vol3 = %f / n",vol1,
                                             vol2,vol3);

return(0);
}
/* 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<nvol[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);
}
/* sclon.c */
sclon(lin,col,pristr,inlon)
int lin,col;
char pristr[];
long * inlon;
{
int i,j,len,n;
char c;
char vol[81];
for(i=0;i<81;i++)vol[i]=' / 0';
n=11;
printf(" / x1b[%d;%dH",lin,col);
printf("%s",pristr);
len=strlen(pristr)+1;
printf("%d",* inlon);
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");
if(vol[0]!=' / 0')* inlon=atol(vol);
printf("%d",* inlon);
return(1);
case 0x42:printf(" / x1b[%d;%dH",lin,col+len);
printf(" / x1b[k");

```

```

        if(vol[0]!='/'&'0')*inlon=atol(vol);
        printf("%d",*inlon);
        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&&(c==43||c==45||c>47&&c<58)){vol[i]=c;i++;}
else{printf("/b");printf("");printf("/b");}
}
printf("/x1b[%d;%dH",lin,col+len);
printf("/x1b[k");
if(vol[0]!='/'&'0')*inlon=atol(vol);
printf("%d",*inlon);
printf("%d",*inlon);
return(0);
}
/* sscrdu.c */
sscrdu(lin,col,pristr,indou)
int lin,col;
char pristr[];
double *indou;
{
    int i,j,len,n;
    char c;
    char vol[81];
    for(i=0;i<81;i++)vol[i]='/'&'0';
    n=21;
    printf("/x1b[%d;%dH",lin,col);
    printf("%s",pristr);
    len=strlen(pristr)+1;
    printf("%f",*indou);
    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");
            if(vol[0]!='/'&'0')atofd(vol,indou);
            printf("%f",*indou);
            return(1);
        case 0x42:printf("/x1b[%d;%dH",lin,col+len);
            printf("/x1b[k");
            if(vol[0]!='/'&'0')atofd(vol,indou);
            printf("%f",*indou);
            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&&(c==43||c==45||c==46||c>47&&c<58)){vol[i]=c;i+
+;}
else{printf("/b");printf("");printf("/b");}
}
printf("/x1b[%d;%dH",lin,col+len);
printf("/x1b[k");
if(vol[0]!='/'&'0')atofd(vol,indou);
printf("%f",*indou);
return(0);
}
/* atofd.c */
/*
atofd(vol,outdou)
char vol[];
double *outdou;
{
    char c;
    int i;
    double d;
    c='+';
    *outdou=0.0;
    d=-1.0;
    for(i=0;i<81;i++)
    {if(vol[i]='/'&'-'&&*outdou==0.0)c='-' ;
        if(vol[i]='/'&'.'&&d<0.0)d=1.0;
        if(vol[i]>47&&vol[i]<58){*outdou=*outdou*10+vol[i]-48;
            if(d>0.0)d=d*10;}
        if(vol[i]='/'&'0'){if(d>0.0)*outdou=*outdou/d;
            if(c='/'&'-'&) *outdou=-*outdou;
            return(0);}
    }
}

```