用 C 语言开发 FoxBASE+的图形功能

耿 磊 (北京经济学院)

一、引 言

自从 Fox 引入我国后,以其编程方便直观、开发周期短、运行速度极快而深受广大软件开发者的喜爱,因此已被广泛应用于开发信息管理系统中。但是,由于它自身内部没有提供任何的图形功能,使得进一步开发较高质量的系统受到很大限制。因此,如何开发 FoxBASE+图形功能就成为软件开发者非常关心的一个问题。

在高级语言如 TurboC、PASCAL 等,都提供了非常强的图形功能,利用这些函数,人们可以很方便地设计出二维甚至三维的界面以及制作动画。那么,能否将这些图形功能用于 FoxBASE+的开发呢?

下文以 TurboC 为例,采用将 C 语言程序驻留于内存中,以数据库为接口的方法,使得在 FoxBASE+中可以方便地使用 C 语言提供的图形函数,有效地解决了 FoxBASE+缺乏图形功能的不足。

二、实现原理

为了使 Fox 程序能任意调用 C 语言的图形函数,先编制 C 语言的图形函数功能库(见函数:LIB( )),将所有欲调用的 C 语言图形函数都集中在该库中,为调用方便,分别予以编号,该函数功能库能够接受并分析由 Fox 接口函数库(FUNLIB、DBF,详见下面的说明)传递过来的编号及参数,执行相对应的图形函数,从而实现 Fox 程序可以随机调用 C 图形函数的目的。

在 FoxBASE+中,用于实现与其它语言接口的方法有两种:一种是使用 RUN 或"!",即在 FoxBASE+中直接运行扩展名为.EXE、.COM 或.BAT 的可执行文件;另一种是使用函数命令 CALL,调用二进制文件(扩展名为.BIN)。其中,第一种方法常因内存不够而运行失败;而第二种方法只要所调用的二进制文件的长度不超过 32K,文件即可成功运行,极少有因内存不够而失败的情况。因此,本文采用了第二种方法。CALL 函数的调用格式为:CALL<文件名>WITH<参数>,其中,参数为字符串变量或字符串常量,用来向所调用的程序传递所需要的参数。由于该参数传递的内容远远不能满足程序设计的要求,为此,本文提出了一种以数据库为接口的方法,来解决它同 C 语言之间参数传递。该接口数据库(FUNLIB.DBF)结构为:

```
Structure for database:  D:FUNCLIB.DBF
Number of data records:  10
Data of Last update:     3/2/94
Field  FieldName    Type      Width    Dec
  1    CODE         Numeric   2
  2    PALATTE      Numeric   2
  3    COLOR        Numeric   2
  4    PARA1        Numeric   3
  5    PARA2        Numeric   3
  6    PARA3        Numeric   3
  7    PARA4        Numeric   3
  8    PARA5        Numeric   3
  9    PARA6        Numeric   3
 10    PARA7        Numeric   3
 11    PARA8        Numeric   3
 12    PARA9        Numeric   3
```

其中,字段 1 为欲调用函数的编号;其它字段为参数字段,用以存放相应的参数。注意:参数字段的数目不应少于 C 图形函数中参数个数的最大值。

在 Fox 程序中,当要使用 C 语言的图形函数时,根据需要在该库中填写图形函数的相应编号及参数,例如调用 C 画线函数(Line( ))的方法如下:

```
use funclib.dbf
appen blan
repl CODE with 6,color with3
repl PARA1 with 20,PARA2 with20
repl PARA3 with 200,PARA4 with 200
use
```

填写完后,调用图形函数功能库,就可实现在 FOX 系统下对 C 画线函数的调用。同样的方法,一次可完成一个或多个 C 函数的调用。

前面提到,函数 CALL 要求调用的文件不能超过 32K,而丰富的 C 图形函数积量很可能使图形函数功能库

© 中国科学院软件研究所 http://www.c-s-a.org.cn

的长度超过 32K,因此,本文采用中断的方式,将图形函数库驻留在内存中。这样不仅提高了运行速度,并且还可以使 FOX 程序象调用自身的内部函数一样快速、自由、方便。

在 DOS 操作系统中,DOS 以中断的方式对异步事件进行响应。在内存最低端 0000:0000～0000:03FF 处共有 1K 的内存空间,用来存放 256 个中断向量,虽然 DOS 及其它应用程序已占用了许多向量,但是仍有一些中断向量空闲出来,供开发者使用。本文取未使用的中断向量 80H(在内存偏移量 0000:0200H 处的四个字节),作为图形函数库的中断入口。设置中断和驻留内存用 C 语言的 setvect( )及 keep( )两个函数极易实现(详见所附程序 CLIB.CPP)。因此,便可以调用名形函数功能库,而该文件又不会超过 32K,其建立方法可以用下面简单的方法实现:

```
C:>DEBUG
-A
  XXXX:0100  INT 80
  XXXX:0102  RETH    ;远程调用方式返回
  XXXX:0103
-N  FUNLIB.BIN
```

三、结　论

用小模式将 CLIB.CPP 编译为可执行文件 CLIB.EXE。使用时先运行 CLB.EXE,将图形函数功能库驻留在内存中,然后在运行汉字系统及相应的 FPX 程序,即可在 FoxBASE+系统下方便的调用 C 语言的图形函数。所附程序在 COMPAQ386 上 WPS、2.13K、中国龙、天汇、启明星等汉字系统下运行通过。

```
*程序名:SAMPLE.PRG
*功能:调用图形函数功能库示例
set safe off
set talk off
set stat off
Load call clib.bin
use funclib.dbf
appe blan
*调用 C 屏幕图形初始化函数
repl code with 1
appe blan
*调用 C 画线函数
repl code with 6,color with 3
repl para1 with 20, para2 with20
repl para3 with 200, para4 with 200
use
call callclib.bin
release module callclb.bin
set safe on
set status 0n
set talk on
RETU
```

```
程序名:CLIE.CPP
功能:开发 FoxBSAE-图形功能示例
编者:耿　磊
日期:1994.3.2
#include <dos.h>
#include <stdio.h>
#include <graqhics.h>
#include <lib.h>//函数功能库索引表
#ifdef__cplusplus
    #define__CPPARGS...
#else
    #define__CPPARGS
#endif
void interrupt Newint80handler(__CPPARGS);
char FileName[ ]="UNCLIB.DBF";//接口库
//接口库参数结构
struct _Record
{
    int code;//c 图形函数编号
    int palatte;//以下为该函数相应的参数
    int color;
    int para1;
    int para2;
    int para3;
    int para4;
    int para5;
    int para6;
    int para7;
    int para8;
    int para9;
}Parameters;
enum BOOLEAN {FALSE,TRUE};
unsigned int RecordNumber;
unsigned int RecordLength;
unsigned int StructureLength;
unsigned char str[35];
FILE *fp;
//根据 FUNCLIB.DBF 文件头,取结构信息
void GetHandle( )
{
    fread (str,32,1,fp);
    RecordNumder=str[5]*256+str[4];
    StructureLength=str[9]*256+str[8];
    RecordLength=str[11]*256+str[10];
}
int Change(int i,unsigned char c1,
unsigned char c2,unsigned char c3)
    unsigned int a1,a2;
    if (i<=3)
    {
```


© 中国科学院软件研究所 http://www.c-s-a.org.cn

```
        a1=(c1==0x20)?0:(c1-0x30)*10;
        return (a1+c2-0x30);
    }
    else {
        a1=(c1==0x20)?0:(c1-0x30)*100;
        a2=(c2==0x20)?0:(c2-0x30)*10;
        return (a1+a2+c3-0x30);
    }
}
  取出接口库中每一条记录的信息
void GetFieidData(int Re)
{
  fseek(fp,Long(StructureLength+1+Re*RecordLength),SEEK_SET
);
  fread(strR,Recordiength,1,fp)
  Parameters.code =Change(1,str[0],str[1],″);
  Parameters.palatte=Change(2,str[2],str[3],″);
  Parameters.color=Change(3,str[4],str[5],″);
  Parameters.para1=Change(4,str[6],str[7],str[8]);
  Parameters.para2=Change(5,str[9],str[10],str[11]);
  Parameters.para3=Change(6,str[12],str[13],str[14]);
  Parameters.para4=Change(7,str[15],str[16],str[17]);
  Parameters.para5=Change(8,str[18],str[19],str[20]);
  Parameters.para6=Change(9,str[21],str[22],str[23]);
  Parameters.para7=Change(10,str[24],str[25],str[26]);
  Parameters.para8=Change(11,str[27],str[28],str[29]);
  Parameters.para9=Change(12,str[30],str[31],str[32]);
}
//图形函数功能库,仅举几个为例
void lib(int CODE)
{
    switch(CODE)
    {
      case_INITGRAPH:
      int gdriver=DETECT,gmode;
      initgraph(&gdriver,&gmode,″″);
      if(graphresult( )!=grok)
       {
        printf(″Graphics error:%s\ n″,grapherrormsg(graphresult(
)));
        return;
       }
      break;
    case_BAR:
      setfilstyle(Parameters.palatte,Parameters.color);
      bar(Parameters.para1,Parameters.para2,
      Parameters.para3,Parameters.para4);
      break;
    case_SETCOLOR:
      setcolor(Parameters.color);
      break;
    case_LINE:
      setcolor(Parameters.color);
      line(Parameters.para1,Parameters.para2,
      Parameters.para3,Parameters.para4);
      break;
    case_PUTPIXEL:
      putpixel(Parameters.para1,Parameters.para2,
      Parameters.color);
      break;
    case_CLOSEGRAPH:
      closegraph( );
      break;
    default:break;
    }
}
void ActiveLib( )
{
    int Record;
    for(Record=O;Record<RecordNumber;Record++)
    {
      GetFieldData(Record);
      Lib(Parameters.code);
        }
    }
    void interrupt NewInt80handler(_CPPARGS)
    {
      if((fp=fopen(FileName,″rb″))==NULL)
      {
        printf(″Cannot open file funclib.dbf″);
        return;
      }
      GetHandle( );
      ActiveLib( );
      fclose(fp);
}
int main( )
{
    //设置 80h 中断向量入口地址为新的中断例程
    setvect(0x80,NewInt80handler);
    //将新的中断例程驻留在内存中,然后返回
    keep(0,(_SS+(_SP/16)-_psp));
    return 0;
}
//头函数:lib.h
//功能:定义 C 图形函数的索引编号
#define    _INITGRAPH1
#define    _BAR2
#define    _CIRCLE3
#define    _SETCOLOR4
#define    _SETFILLSTYLE5
#define    _LINE6
#define    _PUTPIXEL7
#define    _BAR3D8
#define    _OUTBAR9
#define    _SETVISUALPAGE12
#define    _CLOSEGRAPH14
```


© 中国科学院软件研究所 http://www.c-s-a.org.cn