第一部分　C 语言编程技巧

1. 保存与恢复汉字屏幕窗口的方法

方法(一)

通过分配内存来实现屏幕和窗口的保存与恢复，见程序(一)。内存的大小由屏幕窗口的大小决定，通过 malloc()函数分配，调用 save _ video()函数存贮，调用 restore _ video()函数恢复屏幕窗口。该方法适用于对部分屏幕或窗口的保存与恢复。

方法(二)

利用显示缓冲区中的空余页面进行存贮，见程序(三)。调用 save _ video _ padge()函数存贮，调用 restore _ video _ padge()函数恢复。适用于对整个视屏的存贮与恢复，改变函数的段地址可改变页面号，最多可保存 6 个视屏。该方法的特点是速度快，不需另行分配内存。

方法(三)

把汉字屏幕的整个视屏或窗口永久地保存在文件中，见程序(二)。由于要进行磁盘的读写操作，速度稍慢。

演示程序分别用三种方法对屏幕和窗口进行存贮、恢复和移动操作，采用快速的直接视屏存取，对中西文屏幕都适用。一些没有提供对汉字屏幕存贮和恢复功能的高级语言，也可调用方法(二)或方法(三)的功能。程序用 Turbo C 2.1 编译，在浪潮 286(CEGA 卡)和东海 286(CEGA 卡)上运行通过。

程序清单

```
#include<stdlib.h>
#include<stdio.h>
#include<conio.h>
#include<mem.h>
#include<string.h>
main()
{
  intsize,startx=10,starty=19,endx=13,endy=33;
  int startx1=15, starty1=40, endx1=18, endy1=54;
  char *p1, *p2;
  FILE *fd;
  size=(endx-startx+1) * (endy-starty+1) * 2;
  p1=malloc (size);
  p2=malloc (size);
  display()
  getch();
/*程序(一)*/
  save _ video(startx, starty, endx, endy, p1, p2);
  clrscr();
  restore _ video(startx, starty, endx, endy, p1, p2);
  getch();
```

```
/*程序(二)*/
  fd=fopen ("scr1.xwf", "w+b");
  save_video(startx, starty, endx, endy, p1, p2);
  fwrite(p1, size, 1, fd);
  fwrite(p2, size, 1, fd);
  fclose (fd);
  fd=fopen("scr1.xwf", "r+b");
  fread(p1, size, 1, fd);
  fread(p2, size, 1, fd);
  restore_video(startx1, starty1, endx1,endy2,p1,p2);
  fclose(fd);
  getch();
/*程序(三)*/
  save_video_padge();
  clrscr();
  getch();
  restore_video_padge();
  free(p1);
  free(p2);
}
display()
{
  register int i, j;
  for (i =1, j=15 ,i<26; i++)
  {
    gotoxy (j,i);
    printf("THIS IS THE TEST OF THE CHINESE SCREEN PROGRAM");
  }
  gotoxy(20 ,11);
  printf("中国计算机用户");
  gotoxy(20, 12);
  printf("中国计算机用户");
  gotoxy(20, 13);
  printf("中国计算机用户");
  gotoxy(20, 14);
  printf("中国计算机用户");
}
save_video(int startx,int starty, int endx,int endy, char * p1,char *p2)
{
  register int i, j;
  char far * v1=(char far * ) 0xB8000000; char far *t1;
  char far * v2=(char far * )0xB8000000;char far *t2;
  for (i=startx; i<endx+1; i++)
  for (j=startx; j<endx+1; j++)
```

```
    {
      t1=(v1+(i*160)+j*2);
      *p1++=*t1++;
      *p1++=*t1;
      t2=(v2+(i*160)+j*2);
      *p2++=*t2++;
      *p2++=*t2;
    }
  }
restore_video (int startx, int starty, int endx,int endy,char *p1, char *p2)
{
  register int i , j;
  char far *t1=(char far *) 0xB8000000;char far * v1;
  char far *t2=(char far *) 0xB8000000; char far *v2;
  for(i=startx; i<endx+1; i++)
  for(j=starty; j<endy+1; j++)
  {
    v1=t1;
    v1+=(i*160) +j*2;
    *v1++=*p1++;
    *v1=*p1++;
    v2=t2;
    v2+=(i*160)+j* 2;
    *v2++=* p2++;
    *v2=*p2++;
  }
}
save_video_padge ()
{
  movedata (0xb800, 0x0000,0xb900, 0000,4000);
  movedata (0xb000, 0x0000,0xb100, 0000,4000);
}
restore_video_padge ()
{
  movedata (0xb900, 0x0000, 0xb800, 0000, 4000);
  movedata (0xb100, 0x0000 ,0xb000, 0000, 4000);
}
```

徐卫飞

2. 在 Turbo C 2.0 图形方式下显示彩色汉字

Turbo C 2.0 提供了丰富的图形函数库，因而用它来开发用户界面相当完美。然而，由于它是西文软件，不可避免地在汉字处理方面就要有许多不足之处。

Turbo C 2.0 在图形方式下，提供了 outtextxy 函数，其主要功能是将一串字符送到指定

位置。但此函数不能输出汉字字符串，一般用户只能用 printf 函数来显示汉字串。因此，就不能像 outtextxy 函数那样在图形下正确地定出汉字串的显示位置，也不能显示出彩色汉字。为此，研究了一套方法，不但能完成 outtextxy 函数的一切功能，而且还能显示出彩色汉字。经过一些处理后，还能放大和产生美术字体，从而大大地增强了用户界面的完善性。

程序清单

```
# include <graphics.h>
# include <stdio.h>
# include <stdlib.h>
# include <conio.h>
# include <dos.h>
# include <string.h>
void drawhz(int x,int y,char finame [15], int color) /* 输出汉字串函数 */
{
  short int coordx, coordy,i;
  FILE *fp;
  if((fp=fopen(finame, "rb"))== =0)     /* 判断汉字串文件存在否? */
  {
    printf("Can not open the file\n");
    return;
  }
  while(! feof(fp))              /* 读入汉字串文件内容 */
  {
    i=fscanf(fp, "%d, %d" ,&coordx, &coordy);
    if (i<=0) {fclose (fp);return;}
    coordx=x+coordx; coordy=y+coordy;
    putpixel(coordx,coordy,col); /* 显示汉字串 */
  }
  fclose(fp);
}
main()
{
  int maxx,maxy,x1,y1,x2,y2,errorcode;
  int graphdriver=DETECT,graphmode;
  struct arccoordstype arcinfo;
  initgraph(&graphdriver,&graphmode,"");  /* 设置图形显示器方式 */
  errorcode=graphresult();
  if (errorcode! =0) /* 判断图形文件存在否? */
  {
    gotoxy(10,10);printf ("图形文件错误:%s\n",grapherrormsg(errorcode));
    gotoxy(10,11);printf("按任一个键退出...");
    getch();
    closegraph(); exit (1);
  }
```

```
    drawhz (80,130," jcqxxt.hzc",4);/* 调用显示汉字串函数 */
  /* 注:从坐标(80,130)处开始显示一串汉字,"jcqxxt.hzc"为汉字串文件,存的为汉字串的点阵坐标值,"
4"为要显示汉字串的颜色值(此处为红色)。 */
    getch(); closegraph() ;exit(0);
}
```

汉字串文件的实现方法是在 BASIC 或 C 等高级语言下,先显示一串汉字,然后取出此串汉字的点阵坐标,存于汉字串文件中,该汉字串文件就可使用了。

此方法在 AST 386 等微机上运行通过。

王斌

3. 用 C 语言实现的文本文件阅读器

为了对文件进行归类或要查询某一个文件,经常需要阅读各类文本文件,目前尚未见到阅读文本文件的专用软件,比较流行的是采用 Borland 公司的 Turbo 系列软件上所附的文本阅读器软件 README. COM,或 MS DOS 系统提供的 TYPE 命令实现。前者是为西文系统设计的阅读软件,是比较完善的文本阅读器,具有上下翻行、翻页、打印某页、显示移到文件头和字符搜索等功能;后者可用于文本文件的显示,但功能极弱,不能适应日常工作中文本文件的显示与阅读。由于 README. COM 是针对西文的 ASCII 码文件设计的,将其应用于中文文件的阅读时存在许多问题,尽管国内近来对 README. COM 的汉化做了一些工作,但由于汉化工作是面对可执行文件而非源程序进行,因此,始终存在一些难以解决的问题。主要是以下三个问题:

①不支持带统配符"?"或"*"的文件名,一次只能对一个文件进行阅读,而不能对一批文件名有规律的文本文件进行依次逐个阅读。

②难以辩识出文件中的各种控制符(在各类文字处理系统建立的中文文件中,含有各种用于分页、分行、字体选择和字型变化的控制符,这些控制符的 ASCII 码值通常介于 80H—A0H 之间,在阅读时,西文软件将其作为扩展 ASCII 码中的图形符显示)。由于受文本文件中控制符的影响,极易使显示的文件发生混乱。

③不能正常显示用 SUPER WPS 系统的 D 命令编辑的文书文件。

鉴于汉化 README. COM 存在诸多不足,面此用户只好使用文字处理系统来阅读文件,但该方法又因频繁进入与退出文字处理系统,不但需要频繁的读写磁盘操作,而且需不断键入新的文件名才能完成多个文件的查询与阅读。为此,用 Turbo C 2.0 编制了一个程序(READFILE.C),它可以实现文本阅读的基本功能。将该程序与文字处理系统结合起来,可以完成一个增强的文本阅读功能,即将 READFLLE.C 程序用于文件的搜索、查询与阅读,而将字处理系统作为搜索某一字符串、打印文件的某一页和修改文件,互相取长补短,使功能倍增。

1. 程序的功能

READFILE 程序具有如下功能:

①支持带统配符"?"、"*"的文件名,实现了用统配符一次阅读一批文件的功能,程序根据输入的文件名(可含盘符、路径和文件名)自动在指定范围内搜索满足条件的文件,并依次逐个读入内存,在屏幕上显示出来供用户阅读。

②在显示完文件的第一屏内容(22 行文本)后,屏幕底部出现提示信息和命令行,此时可键入下列命令实现阅读文件所需的不同功能:

PGUP:向前翻一屏,即重新显示前一屏的内容。若当前显示的是文件第一屏的内容,向前翻屏仍显示原内容。

F:将文件指针移到文件开头,即重新显示文件第一屏的内容。

E:当前显示的文件结束.开始显示下一个满足条件的文件的第一屏内容。

ESC:结束文件的阅读,返回 DOS 状态。

其它任意键:继续显示文件下一屏的内容。

③可以正常显示 SUPER WPS 的文书文件与非文书文件。程序通过判别文件头是否为"01FF",以区别 WPS 系统的二类文件。若判断文件为文书文件,则自动将文件的读写指针移到从文件头开始偏移地址 1024 的位置(即文件内容开始存放的位置),以避开文件头。若为非文书文件,则从文件头开始显示。

④解决了汉字的显示问题,杜绝了将一个汉字分别显示在二行的情况。该项功能是通过判断用双字节(其机内码均大于 A0H)表示的汉字的前一字节的显示位置,使其不显示在屏幕的第 79 列(屏幕显示列为 0～79)来实现的。

⑤程序可滤掉原文件中含有的字型控制符(ASCII 码大于 7FH 小于 A0H)、分页符(ASCII 码为 0C0AH)、硬回车(ASCII 码为 0D0AH)和软回车(ASCII 码为 8D8AH)等特殊意义的符号,并将后三种符号作为统计显示行数的标准。由于成功地滤掉了文件中的控制符,从而彻底地解决了由这些符号引起的汉字误组合使显示文件内容混乱的问题。

⑥可由用户根据需要指定在屏幕上显示文件的行数,实现了部分文件内容查询的功能。若指定的显示行数小于屏幕允许的显示行数,则将不停地依次显示满足条件的文件的内容。

2. 程序的使用

该程序可放在硬盘根目录下作为 DOS 的外部命令使用,也可以在 SPDOS、2.13 和 UCDOS 等中文系统下使用,它具有比 TYPE 命令强大得多的功能。另外,尽管该程序是针对 SUPER WPS 系统下文件的显示和查询,但由于解决了汉字文件显示中的常见问题,也完全适用于其它系统文本文件的显示,只是因为各文字处理系统的分页符、软回车和硬回车的编码可能存在差异,使屏幕显示的行数与要求的行数有一定的误差。

程序可以按以下方式使用:

①C>READFILE 文件名 行数<回车>

②C>READFILE 文件名 <回车>

③C>READFILE <回车>

后两种使用方式不指定显示文件的行数,则隐含为显示文件的全部内容。第一种使用方式指定显示行数,则每个文件显示完指定的行数后就自动结束,系统将继续显示下一个满足条件的文件。另外,程序还能根据是否输入了文件名而进行提示。文件名中可以包括驱动器、路径和统配符"*"和"?",此时为一批文件的阅读。

程序在 AST 386SX/20 微机上调试通过,也适用于其它型号的微机,程序编译的软件环境为 Turbo C 2.0。

程序清单

```
# include "stdio.h"
# include "dos.h"
# include "dir.h"
# include "ctype.h"
```

```
main(argc,argv)
int argc;
char *argv[];
{
  char ch1,ch3,dir[80];
  FILE *fp;
  struct ffblk f;
  long int ll[50];
  int i=0,i1=0,ch,k1=0,k2=0,k3=0,l1,l2,p1,ch2;
  printf("\n 中西文文本文件阅读器\n");
  printf(" 版本 V1.0\n");
  if (argc<2)
  {
    prntf("\n 请输入文件名:");
    scanf("%s",argv[1]);
    ch2=getchar();
  }
  if (argc=3) p1=atoi(argv[2]);
  else
  p1=1000;
  getcwd(dir,80);
  if ((dir[0]! =argv[1][0]&& (argv[1][1]=':'))
  {
    ch3=tolower(argv[1][0]-'a';
    setdisk(ch3);
  }
  l1=findfirst(argv[1],&f,0);
  if (l1<0) printf("文件名输入错,磁盘上无此文件\n");
  while(! l1)
  {
    k1=0;
    k3=0;
    if((fp=fopen(f.ff-name,"rb"))=NULL)
    {
      printf("文件 f.ff-name 不能打开! \n");
      exit(-1);
    }
    printf("文件 %s 的内容为:\n",f.ff-name);
    ll[0]=0;;
    if (((getc(fp)=0X01) &&. (getc(fp) =0xff)))
    {
      fseek(fp,1024L,0);
      ll[0]=ftell(fp);
      ch1='y';
```

```
        }
        else fseek(fp,0L,0);
        while ((ch=getc(fp))!=EOF)
        {
          if (ch>=0xa0)
          {
            if (++il==1)
              {
                if (i>=79)
                {
                  printf("\n%c",ch);
                  i=1;
                  continue;
                }
              }
              else
              il=0;
          }
          else
        {
          switch(ch)
          {
            case 0x0a:
            case 0x8d:
            case 0x0d:if (((ch=getc(fp))=0x8a)||(ch=0x0a))
            {
              printf("\n");
              i=0;
              if (++k1>=p1)
              {
                k2=0;
                fseek(fp,0L,2);
              }
              if (++k2>=22)
              {
                k2=0;
                printf("\n 输入命令:PgUp 前一页;F:文件头;E:下一个文件;Esc:到 DOS;其它键下一
页.Command>");
                ch2=bioskey(0);
                printf(\"n");
                ll[++k3]=ftell(fp);
                switch(ch2)
                {
                  case 0x4900:if(--k3=0)
```

```
                {
                  fseek(fp,ll[0],0);
                  printf("this is first page\n");
                }
                else
                  fseek(fp,ll[-k3],0);
                  break;
                case 0x1265;
                case 0x1245;fseek(fp,0L,2);
                  break;
                case 0x2146;
                case 0x2166,k3=0;
                  fseek(fp,ll[0],0);
                  break;
                case 0x011b;fclose(fp);
                  printf("\n 正常返回到 DOS 状态);
                  setdisk(dir[0]-'A');
                  ch=chdir(dir);
                exit(0);
              }
            }
          continue;
        }
      )
      if ((ch< 0xa0) &&(ch>0x7f)) continue;
    }
    printf("%c",ch);
    ++i;
  }
  fclose(fp);
  printf("\n OK!%c;",ch3);
  ch1='y'? printf("文书文件");printf("非文书文件");
  printf("%s 显示完毕\n",f.ff-name);
  ll=findnext(&f);
}
setdisk(dir[0]-'A');
ch=chdir(dir);
}
```

邓波　蒲昌平

4. 磁盘文件的递归显示与查询实用程序

在 DOS 的树型目录文件管理系统中，DIR 命令仅能显示当前目录下的可见文件，不能显示隐含文件或子目录下的文件。为了使用多级子目录下的一个文件，经常需要查询其全路径

名，但用 DIR 命令难以满足要求，不得不频繁使用子目录转换命令进入各级子目录后再用 DIR 命令查询。此外对于硬盘的管理来讲，了解磁盘上的全部文件是非常重要的。为此，用 Turbo C 2.0 编写了一个增强显示和查询磁盘文件的实用程序 DIR1.C，它具有类似 UNIX 操作系统的 LC 命令的功能。其功能为：

①当按"DIR1 盘符:"时，将递归显示指定磁盘的各级子目录和各子目录下的文件名，并在隐含文件名后显示"＊"号，与普通文件区别，在子目录后则显示一个"\"号。

②当按"DIR 盘符:文件名"时，将作为查询该文件的全路径名使用，当查找到与指定文件名匹配的文件后，将显示该文件的全路径名和其它属性。

③当按"DIR1 盘符:＊.扩展名"时，将递归显示各级子目录下指定扩展名的文件的全路径名和其它属性。

程序经编译成可执行文件 DIR1.EXE 后，可放在硬盘根目录下作为 DIR 命令的补充。该程序在 AST 386SX/20 上调试通过，也完全适合于其它型号微机。

程序清单

```
#include "dos.h"
#include "dir.h"
#include "string.h"
main (argc,argv)
char * argv [];
{
  struct ffblk f;
  register int do1,d2;
  char path [80]="\0000",path1 [80] [80], * p;
  int level=0,d1=0 ,d3=0;
  path [0]='A' +getdisk ();
  if (argv [1] [1] = = ':')
  {
    path [0] =argv [1] [0] ;
    argv [1] +=2;
  }
  if ((p=strchr(argv [1] ,' * '))= =argv [1]
  {
    argv [1] +=2;
    strupr (argv [1]);
  }
  strcat (path,"\");
  do
  {
    if (1 (strlen (argv [1] )>0)) printf ("\n %s - s \n ",path);
    d1=0;
    strcat (path,"\\*");
    strcpy (path1 [d3], path);
    strcat (path ,"*.*");
    do1=findfirst (path, & f, 23);
```

```
    while (! do1)
    {
      if (stricmp(f.ff_name,".")&&stricmp(f.ff_name,".."))
      {
        if (! stricmp (f.ff_name,argv [1]))
        {
          printf ("\n%s%s %ld\n",path1[d3],f.ff_name,f.ff_fsize);
        }
        if ( (p=strstr(f.ff_name,argv[1]))&&(*(-p)==  '.') )
        {
          printf ("\n %s%s %ld" ,path1[d3],f.ff_name,f.ff_fsize);
        }
        if ( (f.ff_attrib& 0x10)==FA_DIREC)
        {
          strcpy (path1 [d3+1], path1 [d3]);
          strcat (path1 [d3++],f.ff_name);
        }
        if (! strlen (argv [1]))
        {
          printf ("%-s",f.ff_name);
          if ( (f.ff_attrib& 0x10)==FA_DIREC) printf( "\\");
          else ((f.ff_attrib&0x02)==FA_HIDDEN)?
          printf (" *") ;printf ("");
          for (d2=1; d2<(13-strlen(f.ff_name));++d2)
          printf ("");
          if (++d1>5)
          {
            printf ("\n");
            d1=0;
          }
        }
      }
      do1=findnext(&f);
    }
    strcpy (path,path1[level++]);
  }
  while (level <=d3);
}
```

邓波　蒲昌平

5. DOS 环境下的文件连接程序

在DOS环境下,多个文件连接是比较困难的。有的程序运行时,必须将许多文件链接在一起,形成一个文件,作为程序的输入文件,否则程序就无法运行。为此,用C语言编了一个

"CAT"连接程序。

程序清单

```
# include "stdio.h"
main (argc, argv) / * cat: concatenate files * /
int argc;
char * argv []
FILE * fp, * fopen();
if (argc = = 1) / * no args: copy standard input * /
  filecopy (stdin);
else
  while (-- argc >0)
    if ( (fp = fopen ( * + + argv,"r"))==NULL)
      fprintf (stderr, "cat: can' t open %s\n", * argv);
      exit (1);
    } else
        filecopy (fp);
        fclose (fp);
  }
  exit (0);
}
filecopy (fp) / * copy file fp to standard output * /
FILE * fp;
  int c;
  while ((c=getc (fp)) ! =EOF)
  putc(c,stdout);
}
```

使用方法

C>cat　文件1　文件2　文件3　文件4　文件5　文件6　文件7…>文件n

本程序用 Turbo C 2.0 编译，在 IBM PC/XT、GW0520CH、GW286B、Compaq 386/25e 微机上运行通过。

宋广元

6. 用 Turbo C 编写的一条 move 命令

本文介绍的程序可以识别"*"和"?"匹配符，可以成批的搬动文件，给文件管理带来了方便。除路径必须要以"/"结束外，程序使用与 COPY 命令相同。其格式为：

[d_i:][path]move　[[path]oldfilename　[path]newfilename]

程序用 Turbo C 2.0 编写，用小型模式编译，在 IBM PC/XT 机上运行通过，DOS 版本为 3.2。

程序清单

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

```
# include<alloc.h>
int fnum,value;
char fname [1000] [30];
char * subspath [2];
char drive [MAXDRIVE];
char dir [MAXDIR];
char file [MAXFILE];
char ext [MAXEXT];
int flag;
void getname(char str [30])
{
  struct ffblk ff;
  int j=0,done=findfirst (str,& ff,0);
  while (! done)
  {
    j=j+1; strcpy (fname [j],ff.ff_name);
    done=findnext (&ff);
  }
  fnum=j;
}
void getcd (char arg1[],char arg2[])
{
  int i;
  for (i=0;i<2; i++) subspath [i]=(char *)
  malloc (sizeof(char) * 80);
  fnsplit (arg1, drive, dir, file, ext);
  strcpy (subspath [0],drive);strcat(subspath [0],dir);
  strcat (subspath [0],"\0");
  if (arg2)
  {
    flag=fnsplit(arg2, drive,dir, file, ext);
    strcpy (subspath [1], drive);
    strcat (subspath[1],dir);
  }
  else getcwd (subspath [1],80);
  if (subspath[1][strlen(subspath[1])-1]! ='\\') strcat(subspath[1],"\\");
  strcat (subspath[1],"\0");
}
main (int argc,char * argv[])
{
  int k;
  char source[40],dest[40];
  if (argc<2)
  {
```

```
    printf("\n Usage: [d:][path]MV  [[path]oldfilename  [path]newfilename]");
    return;
  }
  else
  {
    getname (argv [1];geted (argv [1], argv [2]);
  }
  for (k=1; k<fnum - 1;k++)
  {
    strcpy (source,subspath [0]);
    streat (source, fname [k]; streat (source,"\0");
    strcpy (dest,subspath [1]);
    if (fnum == =1&&(flag & FILENAME)) streat (dest,file);
    else streat (dest,fname[k]);
    if (rename (source, dest) ==0)
    printf ("File %s move to%s\n",source,dest);
    else
    printf("error file or error path! \n");
  }
}
```

王定安

7. Turbo C 语言可变参数的引用

Turbo C 的 STDARG.H 头文件中，提供了一组实现可变参数引用的工具，一个类型 va_list；一组宏(va_start、va_arg、va_end)。只需简单地应用它们即可实现可变参数的引用。

va_list，将引用可变参数的变量 variation 的类型。

va_start (va_list variation，lastvar)；使 variation 变量指向可变参数表的第一个参数，在调用 va_arg 和 va_end 之前执行，其中 lastvar 为传递给被调用函数的参数表中的最后一个固定参数的变量名。

va_arg (va_list，variation，type)；利用变量类型 type，定义下一个将被传递参数的类型，并将该参数传递给 variation 变量。

va_end (va_list variation)：在 va_arg 读完所有可变参数后，调用 va_end，使被调用函数产生一个正常返回。

利用 va_…系列工具，可以十分方便地完成可变参数的引用。下面以实现任意个整数的算术平均值函数为例，具体说明 va_…系列工具的使用。

程序清单

```
#include"stdio.h"
#include"stdarg.h"
float average (char * message,…)
{
  int count;
  float total=0,avernum;
```

```
    va_list variation;
    int tmp;
    va_start (variation,message);
    while ( (tmp=va_arg (variation, int)) !=0)
    {           /* 可变参数以数字 0 表示结束 */
      total+=tmp;
      count++;
    }
    avernum=total / count
    return (average)
}
main ()
{
    float x;
    char * msg;
    * msg="12,23,34,46,56,76";
    x=average(msg,12,23,34,45,56,76,0);
    printf ("%s 的算术平均数是:%f\n",msg,x);
}
```

罗辉

8. 开发 TVGA 彩卡高分辨(1024×768)图形功能

TVGA 彩卡不仅支持 CGA、EGA、VGA 图形标准，而且提供了比 VGA(640×480)标准更高的视频分辨率(1024×768)和图形功能。采用 1024×768 高分辨显示模块进行 CAD 一类的图形处理工作，无论是图形质量还是工作效率都会成倍提高。

只有配有 512K DRAM 的 TVGA 卡支持 1024×768 高分辨图形模式，而各种图形演示软件虽然画面十分漂亮，但大都是采用 800×600 的分辨率，因此所配置的 TVGA 卡是否具有 1024×768 的高分辨率功能，如何自行开发使用这一功能已成为用户关心的两个问题。

下面介绍用 Turbo C 编写的一个 TVGA(1024×768)图形显示程序，运行此程序可在屏幕上观察到一个十分美观、细腻并经消隐处理后的三维曲面。按照屏幕提示可以从 16 种颜色选择不同的背景色和图形色(当图形色选 100 时，曲面图形将由 16 种颜色的彩带组成)。输入不同的 W 值，可以改变曲面起伏变化的颜率，从面构造不同形态的三维曲面。程序中提供上述各种变化的目的是便于了解和使用 TVGA 卡高分辨图形的各种功能，以便开发自己的图形显示程序。

说明

①程序中使用的 line 函数可在 Turbo C 软件的 BAR.C 中找到。

②如果使用的是 VGART 1024 Prisma 彩卡，仅需将 Setmode 函数中的 5F 改为 5D 即可。

③如果需要在 AUTO CAD 中使用 1024×768 高分辨功能，可利用 TVGA 卡配带的驱动程序 DSVGA.EXE，并按 TVGA 卡手册说明进行显示器配置。

④如果程序不能正常运行，请检查所用的 TVGA 卡是否配有 512K DRAM 以及 DIP 开关的设置是否正确。

⑤本程序产生的图形还可用于观察显示器的分辨率、聚焦、闪烁程度等性能情况。

程序清单

```
/* C program used to display high resolution graphics for TVGA(1024×768)board. */
# include<stdio.h>
# include<float.h>
# include<dos.h>
# include<math.h>
/* Set mode */
int setmode(int bkcolor)
{
  union REGS regs;struct SREGS sregs;
  int ret ;
  regs.x.ax=0x005F;
  sregs.ds=_DS;
  ret=int86x(0x10,&regs,&regs,&sregs);
  /* Set palette registers */
  regs.h.bl=0x00;
  regs.h.bh=bkcolor;
  regs.x.dx=0x0000;
  regs.x.cx=0x0040;
  regs.x.ax=0x1000;
  sregs.ds=_DS;
  ret=int86x(0x10,&regs,&regs,&sregs);
}
/* Write point */
writep (int x, int y, int color)
{
  union REGS regs ;struct SREGS sregs;
  int ret;
  regs.x.dx=y;
  regs.x.cx =x;
  regs.h.ah =0x0c;
  regs.h.al =color;
  int86x(0x10,&regs, &regs,&sregs);
}
main ()
{
  int i,x,y,x1,y1,x2,y2,xc,yc,xr,zr,h,xa,spa,z,lx;
  int xp1,xp2,yp1,yp2;
  float w,xx,yy,zrzr,zz;
  int err,color,color1,bkcolor;
  int ub [701], lb [701];
  printf(" -TVGA high resolution(1024×768)graphics demonstrate-\n\n");
  printf ("0--black          1--blue          3--green\n");
  printf("3--cyan           4--red           5--magenta\n");
```

```
printf("6——brown          7——lightgray      8——darkgray\n");
printf("9——lighthlue      10——lightgreen    11——lightcyan\n");
printf("12——lighred       13——lightmagenta  14——yellow\n");
printf("15——white\n");
printf ("\n Please select backcolor:\n\nBackcolor=?");
scanf("%d",&bkcolor);
printf("\nPlease select graphic color:\n");
printf(" if color=100 ,select all 16 colors to display .\n\nColor=?");
scanf("%d ,&color1);
printf("\nPlease input W\n");
printf(" W=0.02,     2 peaks in surface;\n");
printf(" W=0.044,    3 peaks in surface;\n");
printf(" W=0.07,     4 peaks in surface;\n\n W=?");
scanf(" %f,&w);
setmode(bkcolor);
printf("TVGA(1024×768) graphics demonstrate program");
xc=400;
yc=230;
xr=250;
zr=240;
h=80;
xa=50;
for (i=1;i<701;i++)
{
  ub[i]=0;
  lb[i]=1000;
}
spa=2;
z=-zr+1;
while (z<=zr-1)
{
  color=(color1==100) ? z/5:color1;
  zz=(float)z * (float)z;
  zrzr=(float)zr * (float)zr;
  lx=(int)(xr * sqrt(1.0-(zz/zrzr))+0.5);
  x=-lx;
  xx=(float)x * (float)x;
  y=(int)((float)h * (sin(w * sqrt(xx+zz))));
  x1=x+xc+z;
  y1=yc+y+z/2+0.5;
  for (x=-lx+1;x<=lx-1;x++)
  {
    xx=(float)x * (float)x;
    zz=(float)z * (float)z;
```