بنام خداوند بخشنده ی مهربان

C\C++

www.magazinepro.mihanblog.com

گرافیک در C++C,

گردآوری و ترجمه : سعدالله ویسی

دانشجوی رشته نرم افزار کامپیوتر

در زبان C هیچگونه تابعی در مورد گرافیک کامپیوتری پیش بینی نشده است.اما در بورلند c دارای چندین تابعی هستیم.که هدرفایل h.graphics و تعدادی دیگر در فایل h.conio نیز می باشد.

قبل از هرچیز ما شما را با مفهوم پنجره آشنا می کنم پنجره محدوده ای از صفحه نمایش که خرجی یک برنامه ما در آنجا دیده می شود.ما می توانیم اطلاعات خود را در دو صورت نمایش دهیم

1. **در صورت متنی(Text)**
2. **در صورت گرافیکی(graphics)**

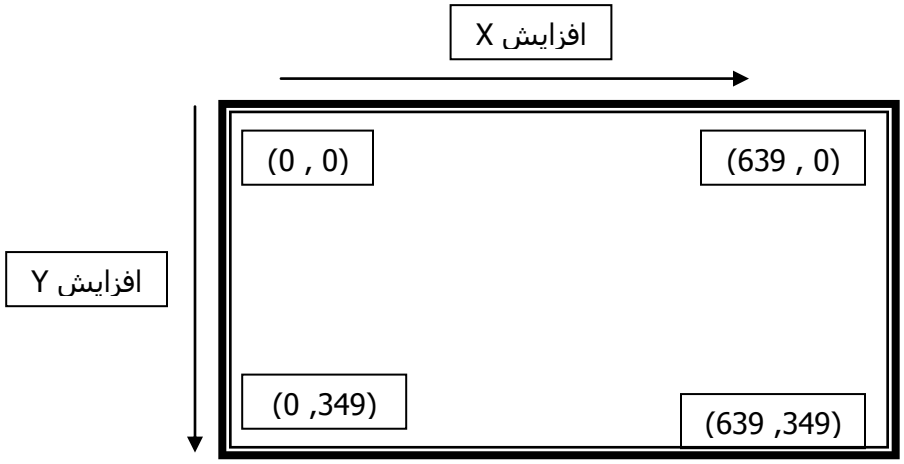
که هریک میتوانند در بوردهای گرافیکی مختلف وضعیت های متفواتی داشته باشند مثال:

VGA,EGA,CGA و ....

خب به اصل مطلب می پردازیم

پیکسل:کوچکترین واحد قابل دسترس در وجه گرافیکی پیکسل نامیده می شود.

مختصات در صفحه نمایش بصورت x,y نمایش داده می شود که در آن x محور افقی و y محور عمودی است.مختصات گوشه ی بالا سمت چپ صفحه نمایش بصصورت (0و0) است.



در ادامه بحث گرافیک به بوردهای گرافیکی و جعبه رنک های آن می پردازیم

تابع initgraph()

تابع گرافیگی initgraph() برای انتقال یک مبدل گرافیکی مناسب به حافظه استفاده می شود این مبدل گرافیکی باید به حافظه انتقال داده شود تا توابع گرافیکی کار کنند.

Void far initgraph(int far *driver,int far *mode,char far *path)

Driver به مبدل گرافیکی که دارای پسوند BGI است اشاره میکند که در فایل h.graphics وجود دارد

اما اگر DETECT=driver دقت کنید باید با حرف بزرگ نوشته شود استفاده شود حالا شما به جای driver اسم متغیری که تعریف کرده اید بنویسید موجب می شود تا تابع بطور اتوماتیک بورد گرافیکی را تشخیص داده و روی بالاترین دقت بگذارد در مثال زیر دقت کنید

قبل از پرداختن به مثال من از turbo c ++ DOS.3 0 استفاده می کنم که در درایو سی به آدرس زیر قرار دارد

C:\TC\BIN

C:\TC\BGI

از c:\tc\bgi بعد از تشخیص برای پیدا کردن مبدل مورد نیاز است که در ایت پوشه قرار دارد.

Int ndrv=DETECT,mode;

Initgraph(&ndrv,&mode,"c:\\tc\\bgi");

دقت کنید بعلت استفاده از کاراکتر های کنترلی در سی پلاس پلاس باید برای درج \ از \\ استفاده کرد تا آدرس تولید شود وگرنه کامپایلر خطا می گیرد.در این دستور وجه گرافیکی VGA به حافظه منتقل می شود.

تابعی نیز وجود دارد که به شما اجازه میدهد تا بتوانید وجه گرافیکی را تعیین کنید

توابع setgraphmode وgetgraphmode این توابع برای تعیین و بدست آوردن حالت گرافیکی مورد استفاده قرار می گیرند

تابع setgraphmode)) است که شکل کلی آن بصورت

void far setgraphmode(int mode)

در دستور فوق mode یک وجه گرافیکی است وجه های گرافیکی بصورت زیر است.

دستور getgrapgmode الگوی آن بصورت زیر است

```
int far getgraphmode(void);
```

در الگوی فوق mode **یکی از حالت های گرافیکی صفحه رو در بر می گیره مثل اندازه صفحه یا تعداد پیکسل های صفحه که بر اساس ماکروهایی که برای راه اندازهای گرافیکی مطرح کردم کار می کند . معمولان مقدار صفر به** mode **میدن یا کلان بهش مقدار نمیدن که همون صفر منظور میشه ! تابع** getgraphmode **هم حالت گرافیکی فعلی را تشخیص داده و ماکروی مربوط به آن را برمی گرداند . مثلان از حالت گرافیکی** CGA **باشد این تابع عدد** 4 **را برمی گرداند .جدول زیر لیست ماکروها را برای شما نمایش میدهد**

نام ماکرو	شماره
DETECT	0
CGA	1
MCGA	2
EGA	3
EGA64	4
EGAMONO	5
IBM 8514	6
HERCMONO	7
ATT400	8
VGA	9
PC3270	10

بفرض مثال setgraphmode(VGA) **وجه گرافیکی** VGA **را در دقت بالا قرار میدهد.**

تابع restorecrtmode()

تابع initgraph() **موجب تغییر حالت صفحه نمایش می شود این تابع حالت صفحه نمایش را به حالت قبل از** initgraph() **بر میگرداند.**

Void far restorecrtmode()

تابع moveto

این تابع باعث می شود که از یک مختصات در صفحه به مختصات دیگه ای در صفحه برید . (اگه با تابع gotoxy **کار کردید یه چیزی شبیه همونه ولی تو محیط گرافیک)**

الگوی کلی این تابع به این صورت است :

```
void far moveto(int x, int y);
```

کـه در این جا x, y مختصات نقطه ای هستند که ما می خواهیم به اونجا بریم . در ضمن توجه داشته باشید که این تابع مختصات داده شده رو از نقطه 0و 0 محاسبه می کنه !

تابع moverel

این تابع کلان شبیه تابع moveto **است ولی با یه تفاوت کلی! انتقال موقعیت جاری به یک نقطه دلخواه نسبت به موقعیت فعلی نشان میدهد ساده تراینه که وقتی شما یه مختصات بهش میدید اون این مختصات جدید رو با مختصات قبلی جمع می کنه و به نقظه جدید میره . مثلان اگه شما روی مختصات 10و 10 باشد و به این نابع مقدار 5و 10 رو بفرستید این مقدار با 10و 10 شما جمع میشه و در مجموع شما به نقطه 15و 20 میرید ولی اگه از** moveto **استفاده کنید شما به نقطه 5و 10 میرید یعنی از 0و 0 محاسبه شده**

شکل کلی تابع

```
void far moverel(int dx, int dy);
```

یک مثال ساده برای این تابع

```
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

int main)void(
{
    */request auto detection /*
   int gdriver  =DETECT, gmode, errorcode;
   char msg]80[;

    */initialize graphics and local variables /*
   initgraph(&gdriver, &gmode, "");

    */read result of initialization /*
   errorcode  =graphresult();
   if (errorcode  =!grOk)   */an error occurred /*
   {
      printf("Graphics error% :s\n",
grapherrormsg)errorcode));
      printf("Press any key to halt":);
      getch();
      exit(1(;  */terminate with an error code /*
   }

    */move the C.P .to location )20, 30/* (
   moveto(20, 30);

    */plot a pixel at the C.P/* .
   putpixel(getx(), gety(), getmaxcolor());

    */create and output a message at )20, 30/* (
   sprintf)msg, %) "d, %d"(, getx(), gety((;
   outtextxy)20, 30, msg(;

    */move to a point a relative distance /*
    */away from the current value of C.P/* .
   moverel)100, 100(;

    */plot a pixel at the C.P/* .
   putpixel)getx(), gety(), getmaxcolor((;

    */create and output a message at C.P/* .
   sprintf)msg, %) "d, %d"(, getx(), gety((;
   outtext)msg(;

    */clean up /*
   getch();
   closegraph();
   return 0;
}
```

تابع line

این تابع همین طور که از اسمش پیداست برای کشیدن یک خط به کار میره و الگوی استفاده ازش هم به این صورته که :

```
void far line(int x1, int y1, int x2, int y2);
```

این تابع از نقطه (x1,y1) شروع میشه و تا (x2,y2) یک خط رسم میکنه

فکر نکنم این تابع نیاز باشه زیاد در مورد حرف بزنیم یک خط رسم میکنه.

تابع lineto

این تابع خطی را از موقعیت جاری تا یک نقطه دلخواه رسم میکند.

```
void far lineto(int x,int y);
```

این تابع بعد از رسم خط موقعیت جاری را به x,y انتقال میدهد.

اما تابع linerel **از موقعیت جاری تا نقطه ای که مختصات آن نسبت به موقعیت جاری سنجیده می شود خطی رسم میکند**

```
Line(100,100,200,200);
Lineto(100,80);
Linerel(30,40);
```

تابع getmoderange **پایین ترین و بالاترین وجه گرافیکی را در نظر میگیرد**

شکل کلی تابع

```
void far getmoderange(int driver,int far *low,int far *high);
```

```
#include <iostream.h>
#include <conio.h>
int main (){
int ndriver=DETECT,gmode;
int far *high,*low;
initgraph)&ndriver,gmode,””(;
getmoderange)driver,low,high(;
cout<<”low:\t”<<low<<endl<<”high:\t”<<high;
return 0;
}
```

نام ماکرو	شماره
DETECT	0
CGA	1
MCGA	2
EGA	3
EGA64	4
EGAMONO	5
IBM 8514	6
HERCMONO	7
ATT400	8
VGA	9
PC3270	10

این برنامه بورد گرافیکی را تشخیص داده و محدوده و وجه گرافیکی را مشخص می کند

تابع getgraphmode()

این تابع وجه گرافیکی فعلی سیستم را نمایش میدهد.

int far getgraphmode()

این تابع فقط مقداری را که بیانگر وجه گرافیکی است بصورت عددی نمایش میدهد بعنوان مثال بورد گرافیکی CGA برای این بورد کرافیکی عدد 4 را برمیگرداند.

تابع setpalette

برای تعیین رنگ سیستم نمایش رنگ در وجه گرافیکی

void far setpalette(int index,int color)

index **عنصری از جدول زنگ اشاره است.**

نام ماکرو	شماره معدل	رنگ
BLACK	0	سیاه
BLUE	1	آبی
GREEN	2	سبز
CYAN	3	کبود
RED	4	قرمز
MAGENTA	5	بنفش
BROWN	6	قهوه‌ای
LIGHTGRAY	7	خاکستری روشن
DARKGRAY	8	خاکستری تیره
LIGHTBLUE	9	آبی روشن
LIGHTGREEN	10	سبز روشن
LIGHTCYAN	11	کبود روشن
LIGHTRED	12	قرمز روشن
LIGHTMAGENTA	13	بنفش روشن
YELLOW	14	زرد
WHITE	15	سفید

در بورد CGA **میتوان رن را تغییر داد که** index **برابر با صفر است دستور زیر را در نظر بگیرید**

Setpalette(0,GREEN)

تابع setbackcolor()

همانطور که از نامش پیداست برای تغییر رنگ ضمینه بکار می رود

void far setbackcolot(int color);

در بعضی از کامپایلرها این دستور کار نمی کند درصورتیکه این دستور کار نکرد از این دستور استفاده کنید

Setbkcolor(int color);

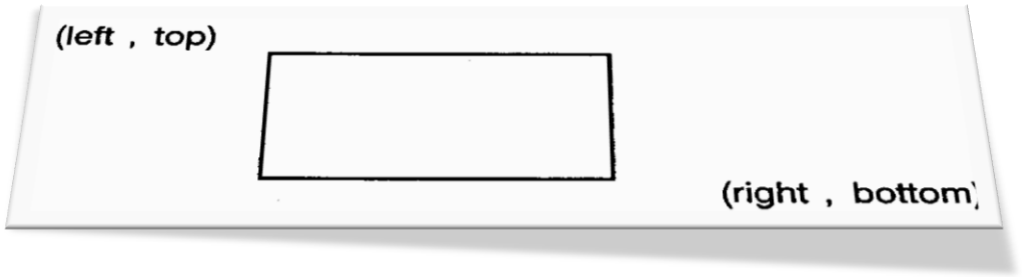
بجای color یا نام رنگ و یا ثابت آنرا می نویسید که در جدول زیر آمده است.

نام ماکرو	شماره معدل	رنگ
BLACK	0	سیاه
BLUE	1	آبی
GREEN	2	سبز
CYAN	3	کبود
RED	4	قرمز
MAGENTA	5	بنفش
BROWN	6	قهوه‌ای
LIGHTGRAY	7	خاکستری روشن
DARKGRAY	8	خاکستری تیره
LIGHTBLUE	9	آبی روشن
LIGHTGREEN	10	سبز روشن
LIGHTCYAN	11	کبود روشن
LIGHTRED	12	قرمز روشن
LIGHTMAGENTA	13	بنفش روشن
YELLOW	14	زرد
WHITE	15	سفید

تابع rectangle

از این تابع برای رسم مستطیل استفاده می شود.

void far rectangle(int left,int top,int right,int bottom)



تابع setfillpattern()

الگویی برای پر کردن اشکال گرافیکی الگوی تابع بصورت زیر است .

void far setfillpattern(char far *pattern,int color)

pattern **آرایه ای حداقل 8 بایتی است که الگوی مورد نظر را مشخص می کند**

تذکر: اگر بیتی برابر 1 **باشد رنگ مشخص شده توسط پارامتر** color **منظور خواهد شد**

```
#include <iostream.h>
#include <conio.h>
#include <graphics.h>

int main] ()
char p]8=[{10,20,30,40,50,60,70,80};
int driver,mode;
driver=DETECT;
initgraph)&driver,&mode,:("""
setcolor)GREEN(;
rectangle)100,200,200,300(;
setfillpattern)p,RED(;
floodfill)150,250,GREENo(;
getch();
return o;
}
```

اما تابع floodfill که در بالا می بینید این تابع داخل یک شکل گرافیکی بسته را با الگو و رنگ تعیین شده توسط تابع setfillpattern() پر می کند:

void far floodfill(int x,int y,int border);

در الگوی فوق (x,y) نقطه ای در داخل شکل گرافیکی بسته می باشد و border رنگ محیط شکل را تعیین می کند.

تابع setfillstyle()

برای تعیین رنگ و سبک پر شدن شکل های گرافیکی استفاده می شود.

void far setfillstyle(int pattern,int color)

ماکرو	معادل عددی	مفهوم
EMPTY_FILL	0	پرکردن شکل با رنگ زمینه
SOLID_FILL	1	پرکردن شکل با رنگی پررنگ
LINE_FILL	2	پرکردن شکل توسط خطوط
LTSLASH_FILL	3	پرکردن شکل توسط / روشن
SLASH_FILL	4	پرکردن شکل توسط /
BKSLASH_FILL	5	پرکردن شکل توسط \
LTBKSLASH_FILL	6	پرکردن شکل توسط \ روشن
XHATCH_FILL	8	پرکردن شکل توسط هاشورزنی
INTERLEAVE_FILL	9	پرکردن شکل از کاراکترهای گوناگون
WIDE_DOT_FILL	10	پرکردن شکل توسط نقاط توخالی
CLOSE_DOT_FILL	11	پرکردن شکل با نقاط پر
USER_FILL	12	الگوی انتخابی توسط برنامه‌نویس با استفاده از تابع setfillpattern()

توابع bar,bar3d

از این دو تابع برای رسم هیستوگرام استفاده می شود.

void far bar(int left,int Top,int right,int bottom)

در این الگو left,top نقطه بالای سمت چپ و right,bottom نقطه پایین سمت راست را مشخص می کند هیستوگرام رسم شده با الگوی تعیین شده توسط تابع setfillpattern() پر می شود.

تابع bar3d مانند تابع bar میباشد با این تفاوت که هستوگرام 3 بعدی را رسم می کند.

void far bar(int left,int Top,int right,int bottom,int depth,int topflag)

که depth میزان بعد سوم را مشخص می کند.

اگر برابر با صفر باشد دوبعدی رسم می کند اما اگر topflag صفر باشد هیستوگرام دارای قله خواهد بود.

Bar(100,100,120,200);

bard3d(200,100,220,200,10,1)

تابع arc()

از این تابع برای رسم کمان استفاده می شود

و دارای الگوی زیر است

void far arc(int x,int y,int start,int end,int radius)

در الگوی فوق x,y **مرکز کمان را مشخص می کنند** start **نقطه شروع و** end **نقطه پایین و** radius **شعاع کمان(دقت کنید**start,end **بر حسب درجه است).**

کد زیر یک کمان 0 **تا** 90 **درجه با مرکز** 100**و**100 **رسم میکند**

arc(100,100,0,90,20)

تابع circle

این تابع برای رسم دایره استفاده می شود

void far circle(int x,int y,int radius);

x,y **مرکز دایره و** radius **شعاع آن است.**

تابع ellipse

برای رسم بیضی و یا قسمتی از بیضی

void far ellipse(int x,int y,int start,int end,int xradius,int yradius)

نقطه x,y **مرکز بیضی،**xradius **شعاع افقی،**yradius **شعاع عمودی**

اما برای رسم قسمتی از بیضی باید نقطه شروع و پایان را در start,end **مشخص کنید**

(start,end **برحسب درجه میباشد)**

تابع setviewport

این تابع برای ایجاد یک محدوده گرافیکی مورد استفاده می کند.

void far setviewport(int left,int top,int right,int bottom,int clip)

بازهم left,top **گوشه ی سمت چپ و** right,bottom **گوشه پایین سمت راست است و اگر** clip **برابر با یک باشد شکل گرافیکی یا متن نمی تواند از محدوده گرافیکی خارج شود.**

توابع outtext,outtextxy

از تابع outtext **برای نمایش متن در وجه گرافیکی در موقعیت جاری صفحه نمایش است.**

void far outtext(char *str)

که *str **متنی است که باید نمایش داده شود.**

تابع outtextxy

این تابع یک متن گرافیکی را در مختصاتی که به آن میدهیم نمایش میدهد.

outtextxy)int x,int y,char str(

تابع settextstyle

با استفاده از این تابع در وجه گرافیکی اندازه کاراکترهایی را که توسط دو تابع outtext,outtextxy **بروی صفحه نمایش نمایش داده می شود تغییر دهیم.**

void far settextstyle(int font,int direction,int size);

در الگوی فوق font **اندازه کاراکتر را تعیین می کند.که مقادیر آن در جدول زیر آمده است.**

نام ماکرو	معادل عددی	مفهوم
DEFAULT_FONT	0	8×8bit_mapped
TRIPLEX_FONT	1	stroked triplex
SMALL_FONT	2	small stroked font
SCANS_SERIF	3	stroked scan serif
GOTHIC_FONT	4	stroked gothic

اما direction نحوه نمایش متن را مشخص می کند که در جدول زیر آمده است

نام ماکرو	معادل عددی	مفهوم
HORIZ_DIR	0	از چپ به راست
VERT_DIR	1	از بالا به پایین

تذکر:font در size ضرب شده و اندازه کاراگتر را افزایش میدهد مقادیر معتبر برای size از 0 تا 10 میباشد.

تابع gettextsettings()

الگوری متنی را که توسط توابع صفحه نمایش مورد استفاده قرار گرفته است را مورد استفاده قرار میگیرد.

void far gettextsettings(struct textsettingstype far* info)

در الگوی فوق info ساختاری از textsettingstype که الگوری متن را نگهداری می کند است و دارای ساختار زیر است:

```
struct textsettingstype (){
int font;
int direction;
int charsize;
int horize;
```

int vert;

}

در ساختمان فوق font میتواند یکی از مقادیر زیر را بپذیرد.

نام ماکرو	معادل عددی	مفهوم
defalt_font	0	8*8 bit_mapped
TRIPLEX_FONT	1	stroked triplex
SMALL_FONT	2	stroked small
SCANS_SERIF	3	stroked scans
GOTHIC_FONT	4	stroked gothic

مقادیر معتبر direction

نام ماکرو	معادل عددی	مفهوم
HORIZ_DIR	0	از چپ به راست
VERT_DIR	1	از بالا به پایین

charsize **ضریبی است که اندازه کاراکتر تاثیر می گذارد و مقادیر** horiz **و** vert **مشخص می کند که متن مورد نظر نسبت به موقعیت جاری مکان نما چگونه باشد.**

ماکرو	مقدار عددی	مفهوم
LEFT_TEXT	0	موقعیت جاری در سمت چپ متن باشد
CENTER_TEXT	1	موقعیت جاری در وسط متن باشد
RIGHT_TEXT	2	موقعیت جاری در سمت راست متن باشد
BOTTOM_TEXT	3	موقعیت جاری در پایین متن باشد
TOP_TEXT	4	موقعیت جاری در بالای متن باشد

توابع getmaxx **و** getmaxy

تابع هرکدام از این دوتابع به ترتیب مقدار X,Y **را در وجه گرافیکی جاری مشخص می کند.**

int far getmaxx()

int far getmaxy()

تابع getpixel()

توسط این تابع رنگ پیکسل موجود را در نقطه ای مشخص باز می گرداند.

int far getpixel(int x,int y);

تابع settextjustify

این تابع چگونگی نمایش متن را در وجوه گرافیکی نسبت به موقعیت جاری تعیین می کند.

void far settextjustify(int horiz,int vert);

نام ماکرو	معادل عددی	مفهوم
LEFT_TEXT	0	**موقعیت جاری در سمت چپ متن باشد**
CENTER_TEXT	1	**موقعیت جاری در وسط متن باشد**
RIGHT_TEXT	2	**موقعیت جاری در سمت راست متن باشد**
BOTTOM_TEXT	3	**موقعیت جاری در پایین باشد**
TOP_TEXT	4	**موقعیت جاری در بالا باشد**

تابع cleardevice()

این تابع در وجوه گرافیکی موجب پاک شدن صفحه می شود و موقعیت مکان نما را به 0،0 تغییر میدهد.

void far cleardevice(void);

تابع clearviewport()

این تابع برای پاک کردن یک محدوده گرافیگی مورد استفاده قرار میگیرد

void far clearviewport(void);

تابع getviewsettings

مشخصات محدوده گرافیکی موجود در صفحه نمایش را در اختیار کاربر قرار میدهد

void far getviewsettings(struct viewporttype fa *info)

viewporttype ساختمانی است که مشخصات محدوده گرافیکی در آن قرار دارد

```
struct viewporttype {
    int left,right,top,bottom;
    int clipflag;
};
```

فکر نکنم دیگر نیازی باشد به توضیح left,top و right,bottom

توابع getx,gety

این دو تابع در وجوه گرافیکی موقعیت مکان نما را بر میگردانند

```
int far getx(void);
int far gety(void);
```

تابع closegraph()

این تابع سیستم را از حالت گرافیکی خارج می کندو کلیه حافظه ای را که در اختیار مبدل گرافیکی و font داده است به سیستم عامل باز می گرداند.

```
void far closegraph(void);
```

تابع detectgraph()

نوع بورد گرافیکی را مشخص می کند.

```
void far detectgraph(int *far driver,int *far mode);
```

در تابع فوق driver شماره مبدل گرافیکی را بر میگرداند و بالاترین دقتی که بورد گرافیکی موجود در کامپیوتر که می تواند پشتیبانی کند در پارامتر mode قرار میدهد.

تابع drawpoly

برای رسم چند ضلعی بکار می رود

void far drawpoly(int numpoints,int far *points);

در الگوی بالا numpoints تعداد اضلاع را مشخص میکند و point به آرایه ای که مختصات گوشه های چند ضلعی در آن قرار دارد اشاره می کند.

```
int shape=[]{10,10,100,80,200,200,350,90,0,0}
drawpoly(5,shape);
```

تابع fillpoly

با توجه به نام تابع معلوم است داخل آن را با یک الگو و رنگ جاری پر می کند.

void far fillpoly(int numpoints,int far *points);

تابع getarccoords()

این تابع مختصات آخرین کمانی که توسط تابع arc رسم شده است را تشخیص و را بر می گرداند.

void far getarccoord(struct arccoordstype far *cords)

ساختمان acrcoordstype

```
struct arccoordtype{
int x,y;
int xstart,ystart,xend,yend;
}
```

ساختمان بالا x,y مرکز کمان،نقطه (xstart,ystart) نقطه شروع و نقطه (xend,yend) نقطه پایان کمان را مشخص می کند.

تابع getbkcolor()

از این تابع برای تشخیص رنگ زمینه استفاده می شود.

int far getbkcolor(void);

تابع getcolor

این تابع رنگی را که اکنون در رسم گرافیکی مورد استفاده قرار میگیرد تشخیص میدهد.

تابع getfillpattern()

الگویی را که اکنون در رسم اشکال گرافیگی مورد استفاده قرار میگیرد را مشخص می کند

void far getfillpattern(char far *pattern);

توابع getimage,imagesize,putimage

از تابع getimage **برای ذخیره کردن قسمتی از صفحه نمایش حاوی گرافیک در حافظه مورد استفاده قرار میگیرد.**

void far getimage(int left,int top,int right,int bottom,void *far buf);

با استفاده از left,top **و** right,bottom **دو سرقطب مستطیل از صفحه نمایش را مشخص می کند و** buf **حافظه ایست که برای ذخیره کردن شکل استفاده می شود.**

برای تعیین میزان حافظه لازم جهت ذخیره کردن شکل از تابع imagesize **با الگوری یز استفاده می شود.**

unsigned far imagesize(int left,int top,int right,int bottom);

این تابع با توجه به مختصات میزان حافظه لازم جهت ذخیره کردن تصویر را مشخص می کند.

unsigned size;

size=imagesize(10,10,100,100);

تابع putimage **شکلی را که تابع** getimage **در حافظه قرار داده است در نقطه ای دیگر از حافظه کپی می کند.**

void far putimage)int x,int y,void far *buf,int op(;

buf **به شکلی که توسط تابع** getimage **ذخیره شده است اشاره می کند و نقطه** x,y **محلی از صفحه نمایش را مشخص می کند که شکل موجود در حافظه** buf **با شروع از این نقطه در صفحه نمایش ذخیره می کند اما پارامتر** op **نحوه یا چگونگی ظاهر شدن شکل را در صفحه نمایش مشخص می نماید.به توجه به مقادیر زیر:**

نام ماکرو	مقدار	مفهوم
COPY_PUT	0	شکل به همانصورت کپی میشود.
XOR_PUT	1	XOR
OR_PUT	2	OR
AND_PUT	3	AND
NOT_PUT	4	NOT

پایان بخش اول

با تشکر سعدالله ویسی مدیر وبلاگ

www.magazinepro.mihanblog.com

منابع

1.Herbert schildt,C the complete refrence)C++,ANSI(

2.Herbert Schildt,Turbo C The Complete Reference

3.Herbert Schildt,Advanced Turbo C

4.Jafar Nezhad Ghomi