بنام خداوند بخشنده ی مهربان

گرافیک در C بخش ۲

گردآوری : سعدالله ویسی

SaadolaH@Gmail.com

۱۳۸۹

WWW.MAGAIZNEPRO.MIHANBLOG.COM

تقدیم به همسر عزیزم

.....

مقدمه

همانطور که میدانید هر زبان برنامه نویسی دارای قابلیت گرافیکی است چرا که امروز هرچه نرم افزار شما قوی هم باشد اما دارای UI(User Interface) زیبایی نباشد محبوبیت چندانی نخواهد داشت

اما در این بخش گرافیک شما باید با ساختمان داده ها آشنایی داشته باشید لذا کاربر گرامی درصورت توانایی کار با ساختمان ها در زبان C به مطالعه این بخش بپردازید در صورت لزوم کتاب الکترونیکی ساختمان ها در C/C++ که در وبلاگ نیز موجود است را بخوانید همچنین این بخش دارای توابعی از اسمبلی است زیرا پیش نهاد می شود با مراجعه به وبلاگ بخش توابع اسمبلی و زبان C را دانلود کرده و مطالعه بفرمائید

با تشکر سعدالله ویسی

گرافیک

همانطور که گفتم در هر زبانی قدرت گرافیکی نهفته است مطمئنن ویژوال بیسیک نامی آشناست برای شما که قدرت گرافیکی خود را دارد اما در تکنولوژی دات نت تمامی زبان های برنامه نویسی در یک سطح قرار دارد و سرعت اجرای کدها در تکنولوژی کاملا مساوی است.

برای گرافیک برنامه هاییکه می نویسیم اولین شرط یا قدم طراحی گرافیک تعییت وجه گرافیکی مناسب برای صفحه نمایش است.

رنگ های موجود در بوردهای CGAوVGA برای هر پیکسل چهار سیگنال(آبی،سبز،قرمز،شدت) به صفحه نمایش ارسال می شود.

کد رنگ	رنگ	کد رنگ	رنگ
۰	سیاه	۸	خاکستری تیره
۱	آبی	۹	آبی روشن
۲	سبز	۱۰	سبز روشن
۳	کبود	۱۱	کبود روشن
٤	قرمز	۱۲	قرمز روشن
٥	بنفش	۱۳	بنفش روشن
٦	قهوه ای	١٤	زرد
۷	خاکستری روشن	۱۵	سفید

خوب موقع آن رسیده یک سری به توابع اسمبلی بزنیم حالا ما احتیاج به اطلاعاتی درباره union REGS داریم.

بصورت ساده تر بگویم شما با این ساختمان میتوانید محتوای ثبات های را تغییر دهید

```
union REGS {
struct WORDREGS x;
struct BYTEREGS h;
```

```
}
```

در فایل <dos.h> قرار دارد و اکثرا برای تابع های زیر بیشترین استفاده را دارد.

int86 int86x intdos intdosx

یک مثال کوچک این یک تابع است که چگونگی استفاده از این ساختمان را به شما نشان خواهد داد

```
Void mv() {
union  REGS regs;
regs.h.ah=2; تغییر محتوی ثبات ah
regs.x.ax=0;
regs.h.dh=1;
regs.x.cx=0;
}
```

برای دسترسی به AX,BX,CX,DX می توانید بصورت زیر عمل کنید

```
union REGS r;
r.x.dx=01;
```

در مثال بالا باید دیگر کار با union REGS را یاد گرفته باشید .

اما در زبان اسمبلی وقفه ۱۰ تابع ۰۰ صفحه نمایش را در حالت گرافیکی قرار میدهد برای این کار از تابع int86 استفاده می کنیم

با استفاده از int86 در فایل <dos.h> میتوان وقفه ای در برنامه ایجاد کرد همانطور که گفتیم وقفه ۱۰ تابع ۰۰ صفحه نمایش را به حالت گرافیکی می برد برنامه زیر نحوه استفاده از این تابع را به شما نشان میدهد.

```
#include <iostream.h>
#include <conio.h>
#include <dos.h>
#define VIDEO 0x10
void movetoxy(int x, int y)
```

```
{
union REGS regs;
regs.h.ah = 2;  /* set cursor position */
regs.h.dh = y;
regs.h.dl = x;
regs.h.bh = 0;  /* video page 0 */
int86(VIDEO, &regs, &regs);
}
```

فکر نکنم نکته ای باشد بجر اینکه وقفه ۱۰ تابع ۰۲ باعث انتقال مکان نما از نقطه ای به نقطه ای دیگر می شود دقت کنید این تابع همان کار تابع gotoxy در زبان C++ است.

برای درک بیشتر این مطلب کمی در مورد زبان برنامه نویسی اسمبلی توضیح می دهیم(مختصر)

وقفه ۱۰ تابع ۰۲

با این تابع میتوان مکان را به هر مکانی که می خواهیم منتقل کنیم که باید ثبات ها بصورت زیر مقدار دهی شوند.

BH	DH	DL
شماره صفحه(۰،۱،۲،۳)	شماره سطر	شماره ستون

خب برنامه زیر مکان نما را به سطر ۱۰ و ستون ۲۰ منتقل می کند.

```
#include <conio.h>
#include <iostream.h>
#include <dos.h>
int main() {
union REGS r;
r.h.ah=2;
r.h.bh=0;
r.h.dh=10;
r.h.dl=20;
int86(0x10,&r,&r);
Getch();
Return 0;
}
```

فکر نکنم در برنامه مقابل چیزی ابهام داشته باشد فقط در قسمت int86 من نوشتم 0x10 خیلی واضح است چون این عدد باید در مبنای Hex باشد و باید 0x حتما جلوی آن نوشته شود.

در پایین همین برنامه را به زبان اسمبلی برای شما نوشته ام.

```
mov  ah,02h          ;function
mov  bh,0        ;page=0
mov  dh,10           ;row
mov dl,20            ;column
int 10h    ;call inttrupt
```

خب البته این برنامه بالا به زبان اسمبلی بصورت مختصر نوشته شده و برای راحتی کار تعریف سگمنت کد و ... را ننوشته ام.

حالت گرافیکی

آداپتورها دو حالت دارند متنی و گرافیکی.

برای تعیین حالت گرافکی از تا h۰۰ از وقفه ۱۰ استفاده می کنیم در حالت گرافیکی صفحه نمایش از مجموعه نقاط پیکسلی تشکیل شده است حالتهای گرافیکی را همراه با دقت آن در جدول پایین صفحه مشاهده خواهید کرد.(در این جدول من فقط ۲ حالت را برای مثال نوشته ام).

حالت	دقت
04h	(رنگی)۲۰۰*۳۲۰
05h	(تک رنگی)۲۰۰*۳۲۰
0DH	(رنگی)۴۸۰*۳۴۰

خوب نوبت مثال:در این مثال من میخواهم برنامه گرافیکی خود را در حالت ۲۰۰*۳۲۰ قرار دهم.

```
#include <conio.h>

#include <dos.h>
#include <iostream.h>

int main() {

union REGS r;

r.h.ah=00;

r.h.al=04;
```

```
int86(0x10,&r,&r);
getch();
return 0;
}
```

خب تا حالا شاید اگر در مورد زبان اسمبلی نظر خوبی نداشتید باید عوض شده باشه و به قدرت این زبان پی برده باشید.

نوشتن پیکسل

در کتاب آقای جعفر نژاد قمی نوشتن پیکسل توضیح داده شده من میخواهم کمی کامل تر این بحث را برای شما باز کنم.

ما می دانیم هر شکلی از کنارهم قرار گرفتن پیکسل ها به وجود آمده است پس در گرافیک برنامه نوشتن پیکسل اصل مهمی است که به دوصورت همانطور که در کتاب آقای قمی نوشته شده است میتوان در پیکسل نوشت.

۱-استفاده از روال های بایاس

۲-نوشتن پیکسل بطور مستقیم در حافظه

در روش اول سرعت کار پایین است

توضیح:

در زبان اسمبلی وقفه ۱۰ تابعی به اسم 0Ch داریم که برای نوشتن پیکسل استفاده می شود خوب حتما می دانید C همان عدد ۱۲ است اما در مبنای ۱٦

A	B	C	D	E	F
۱۰	۱۱	۱۲	۱۳	۱٤	۱۵

ما در زبان برنامه نویسی اسمبلی باید بصورت 0CH بنویسیم چون باید مبنای ۱٦ باشد

اما برای نوشتن پیکسل باید ثبات ها را بصورت زیر مقدار دهی کرد.

AL	AH	BH	DX	CX

ستون	سطر	شماره صفحه	تابع	رنگ پیکسل

*نکته:سطرو ستون را در محدوده برنامه انتخاب کنید(۲۰۰*۳۲۰)

خوب حالا میخواهیم یک برنامه نویسی هم به زبان C++ و هم به زبان اسمبلی

```
#include <conio.h>
#include <dos.h>

int main() {

union REGS r;

r.h.ah=12;                    //0CH(HEX)=>12

r.h.al=4;

r.x.dx=20;

r.x.cx=15;

int86(0x10,&r,&r);

getch();

return 0;

}
```

خب موقع آن رسیده یک مثال هم با زبان اسمبلی برای شما بنویسم.

```
mov ah,0ch

mov al,03;

mov bh,0

mov cx,150

mov dx,100

int 10h
```

خب اینم برنامه به زبان اسمبلی امید وارم تا حالا مشکلی نداشته باشید.

تمرین:با تابع getch() در C++ کار کردید منتظر فشار دادن کلید می ماند ولی آنرا در صفحه نمایش ظاهر نمی کند خب من میخوام این برنامه رو با روال های بایاس طبق اطلاعات زیر که به شما میدم بنویسید.

وقفه ۲۱ تابع ۰۸(برای یادگیری: در این تابع کلیدی که شما فشار داده اید در داخل ثبات al ذخیره میشود)

خوب امید وارم بتونید بدون مشکل این برنامه رو بنویسید.

روش دوم:

(برگرفته از کتاب آقای جعفر نژاد قمی).سرعت نوشتن این روش میتوان گفت بطور تقریبی چیزی نزدیک به ١٤ الی ١٥ برابر روال های بایاس است.خب برای این کار شما باید آدرس وجوه گرافیکی را بدانید در بورد های CGA از B8000000H شروع می شود پیکسل های طر زوج در B8000000 و فرد در B8002000H قرار میگیرد.

برای نوشتن از تابع memponit() استفاده می شود.

۱.پیدا کردن بایتی از صفحه نمایش که پیکسل مورد نظرمان آنجا باشد برای نوشتن هر پیکسل از دوبایت استفاده می شود لذا هر پیکسل میتواند $2^2=4$ رنگ داشته باشد و هربایت میتواند شامل ٤ پیکسل باشد به دستورات زیر دقت فرمائید.

```
index=x*40 + y /4;
if (x%2)
        index+=8512;
```

توضیح:

دستور اول x در ٤٠(تعداد ستون ها) ضرب شده و y بر ٤(تعداد پیکسلها در بایت)تقسیم شده و حاصل آنها با یکدیگر جمع می شود اگر سطری که پیکسل در آنجا نوشته می شود فرد باشد باید مقدار ثابت به دست آمده در دستور اول،اضافه گردد این مقدار بین B8002000 تا B8000000 یعنی 2000H و یا ۸۱۹۲ که از این مقدار باید یک سطر (٤٠) کسر گردد.

۲.تعیین محل پیکسل در بایتی که آدرس آن بایت در متغیر index قرار دارد و نوشتن آن در حافظه صفحه نمایش پیکسل ها در یک بایت از سمت چپ به راست قرار دارد بعنوان مثال پیکسل شماره صفر در بیتهای ٦و ۷ و پیکسل شماره ۳ در بیتهای ۰ و۱ قرار میگیرند چون در هر بایت ٤ عدد پیکسل قرار میگیرند وقتی آنها تغییر می کنند باید مقدار بقیه پیکسل ها باید حفظ گردد که بهترین راه برای حفظ این پیکسل ها استفاده از یک نقاب(mask)است.بطوریکه همه ی بیت های آن بجز بیتهایی که باید تغییر کنند برابر با ۱ باشد این بیتها با بیتهای موجود در حافظه نمایش AND می شوند سپس با اطلاعات جدید OR می گردند این وضعیت با حالتی که در آن مقادیر قبلی با مقدار صفر OR می شود و سپس با مقادیر جدید XOR می گردد متفاوت است زیرا در این حالت کافی است که مقادیر قبلی با مقدار صفر OR شود و سپس با مقدار جدید OR شود.برای پیدا کردن موقعیت پیکسل در بایت،کافی است که باقیمانده Y بر ٤ محاسبه گردد

.

*.کاربر گرامی ادامه بحث گرافیک چون دارای منابع محدودی بوده است و بنده ادامه بحث گرافیک را پیشنهاد می کنم از کتاب 'برنامه نویسی به زبان C 'آقای جعفر نژاد قمی مطالعه بفرمائید.

با تشکر سعدالله ویسی

منابع

۱.جعفرنژاد،قمی،برنامه نویسی به زبان **C**،تهران:علوم رایانه،۱۳۸۸

2.robert lafore,interactive course(C)

3.ulla krich-prinz and peter prinz,A Complete Guide to Programing in C++

All right reserved By MagazinePro