

دانشگاه بوعلی سینا

پروژه پایانی ریزپردازنده واسمبلی

استاد: وکیلان

دانشجویان:

علی امامی (9912358005) ، مهدی براتی (9912358008)

مقدمه

در این کد تعدادی رنگ در صفحه نمایش نشان داده میشوند کاربر میتواند یکی از رنگ هارا انتخاب کند و با موس در هر نقطه از صفحه که بخواهد میتواند با رنگ مورد نظر تایپ کند.

شرح بدنه کد:

در ابتدا نوع برنامه را مشخص میکنیم .

استک ۶۴ تایی تعریف می کنیم و در ادامه متغیری برای قرار دادن رنگ و دو متغیر برای قرار دادن موقعیت موس به صورت گرافیکی و دو متغیر برای قرار دادن موقعیت موس به صورت کاراکتری قرار داده شده است.

```
.MODEL SMALL
.STACK 64

.DATA                                ;DATA SEGMENT

    COLOR DB  07                    ;07 = HIGH INTENSIITY WHITE

    POSX DB 00
    POSY DB 00

    G_POSX DW ?
    G_POSY DW ?

    CLICK DB ?
```

در main برنامه نوع متنی ۸۰ در ۲۵ را مشخص می کنیم

```
.CODE

MAIN PROC    FAR

    MOV AX,@DATA
    MOV DS,AX
    MOV AL,03                ;TEXT MODE 80*25 & 16 COLORS
    MOV AH,00
    INT 10H

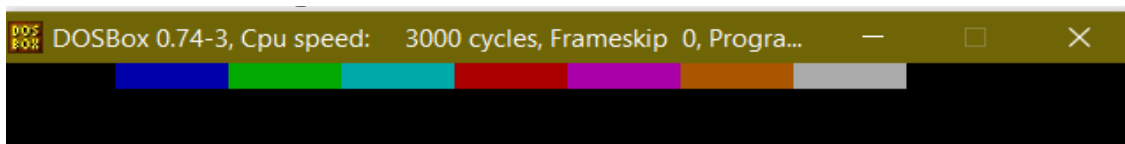
    ;CALL COLOR_PALETTE
```

رنگ‌ها را در آغاز صفحه نمایش می‌دهیم برای این کار نشانگر را به خط اول می‌بریم و در انتهای ستون از موقعیت (0,56) به ترتیب رنگ در نتیجه نمایش داده می‌شود

```
;-----SHOW COLOR PALET-----  
  
    MOV DL,56  
    MOV DH,00  
    MOV BH,00          ;PAGE NUMBER  
    MOV AH,02  
    int 10H            ;SET CURSOR POSITION  
  
    MOV AH,09  
    MOV AL,00  
    MOV BH,00  
    MOV BL,01110000B  
    MOV CX,07
```

این حلقه برای چاپ کردن رنگ‌ها می‌باشد

```
BACK:  
    PUSH CX  
    MOV CX,08          ;LENGTH OF COLORS IN THE PALET  
    INT 10H  
  
    SUB BL,00010000B  
  
    POP CX  
    PUSH AX  
    PUSH BX  
  
    MOV AX,CX  
    MOV BL,08  
    MUL BL  
    POP BX  
  
    SUB AL,08  
    MOV DL,AL  
    MOV DH,00  
    MOV BH,00  
    MOV AH,02  
    INT 10H            ;SET CURSOR POSITION  
  
    POP AX  
    LOOP BACK
```



در قسمت بعد موس را تعریف می‌کنیم و سپس در قسمت `click_check` در ابتدای محل قرارگیری موس را که به صورت گرافیکی است در متغیرهای گرافیکی قرار می‌دهیم (`G_posx,G_posy`)

```
;-----INSTALL MOUSE-----
```

```
MOV AX,00  
INT 33H
```

```
;-----CHECK FOR CLICK-----
```

```
CLICK_CHECK:
```

```
MOV AX,03  
INT 33H  
  
MOV CLICK,BL  
MOV G_POSX,CX  
MOV G_POSY,DX
```

تابع `CONVERT` مختصات را به صورت کاراکتری تبدیل می‌کند سپس در قسمت `cmp` چک می‌کنیم که آیا کلیک صورت گرفته یا نه اگر کلیک شده به تابع `CLICKED` می‌رویم در غیر این صورت تابع `click_check` اجرا می‌شود

```
CALL CONVERT          ;CALCULATE POSITION  
  
CMP CLICK,00          ;IS CLICKED  
JNE CLICKED           ;NOTHING IS CLICKED  
  
JMP CLICK_CHECK
```

در قسمت `CLICKED` چک می‌کنیم که راست کلیک نشده باشد و فقط کلیک چپ فشرده شده باشد.

```
CLICKED:              ;WHEN SOMETHING IS CLICKED
```

```
CMP CLICK,00000010B   ;CHECK IF IT IS RIGHT CLICK  
JNE SELECT  
  
JMP CLICK_CHECK
```

در **SELECT** اگر کلیک چپ در سطر اول فشرده شده باشد به تابع انتخاب رنگ **SELECT_COLOR** می‌رویم در این قسمت رنگ کاراکتری که روی آن فشار داده شده است در متغیر **color** ذخیره می‌شود

```
SELECT:                                ;SELECT CLOLR OR READ CHARACTER
```

```
    CMP POSY,00  
    JE COLOR_SELECT
```

```
    CALL READ
```

```
    JMP CLICK_CHECK
```

```
COLOR_SELECT:
```

```
    MOV AH,08                        ;READ CHARECTAR FROM CURSOR POSITION  
    INT 10H
```

```
    MOV CL,04  
    SHR AH,CL  
    MOV COLOR,AH                    ;AH HOLD COLOR  
    JMP CLICK_CHECK
```

چنانچه اگر در سطر اول کلیک نشده بود به بخش گرفتن از کیبورد و چاپ در صفحه می‌رویم.

```
    CALL READ
```

در این قسمت از کیبورد ورودی گرفته میشود و چک میشود ک اگر **enter** بود یعنی متن نوشته شده به پایان رسیده و موس آزاد میشود تا در نقطه دیگری کلیک کنیم و در انجا بنویسیم

در مرحله بعد چک میشود که اگر **back space** زده شده بود نشانگر یک کاراکتر عقب میرود و ان حرف را پاک میکند.

اگر دو شرط بالا صدق نکرد یعنی ورودی مشکلی ندارد و باید در صفحه با رنگ انتخاب شده نوشته شود

READ	PROC	;READ CHARACTER FORM KEYBOARD
------	------	-------------------------------

	MOV AH,00	
--	-----------	--

	INT 16H	
--	---------	--

	CMP AL,13	
--	-----------	--

	JZ BREAK	
--	----------	--

	CMP AL,08	
--	-----------	--

	JZ DETELE	
--	-----------	--

	MOV AH,09	
--	-----------	--

	MOV BH,00	
--	-----------	--

	MOV CX,01	
--	-----------	--

	MOV BL,COLOR	
--	--------------	--

	INT 10H	
--	---------	--

	MOV AH,02	
--	-----------	--

	MOV BH,00	
--	-----------	--

	ADD DL,01	
--	-----------	--

	INT 10H	
--	---------	--

	JMP READ	
--	----------	--

DETELE:		
---------	--	--

	MOV AH,02	
--	-----------	--

	MOV BH,00	
--	-----------	--

	SUB DL,01	
--	-----------	--

	INT 10H	
--	---------	--

	MOV AH,09	
--	-----------	--

	MOV BH,00	
--	-----------	--

	MOV CX,01	
--	-----------	--

	MOV BL,00	
--	-----------	--

	INT 10H	
--	---------	--

	JMP READ	
--	----------	--

BREAK:		
--------	--	--

	RET	
--	-----	--

تابع تبدیل موقعیت گرافیکی به کاراکتری

```
CONVERT PROC                                ;CHANGE GRAPHIC POSITION TO CHARACTER POSITION

    MOV AX,G_POSX
    MOV BL,08
    DIV BL
    MOV POSX,AL
    MOV AX,G_POSY
    DIV BL
    MOV POSY,AL
    MOV DL,POX
    MOV DH,POY
    MOV BH,00
    MOV AH,02
    INT 10H

    RET

CONVERT ENDP
```

نحوه اجرا :

- 1) رنگ مورد نظر را انتخاب میکنیم
- 2) روی یک نقطه روی صفحه کلیک میکنیم متن مورد نظر را تایپ میکنیم (قابلیت حذف با کلید Back space وجود دارد)
- 3) بعد از نوشتن متن کلید enter را برای تایید متن و رها شدن موس میزنیم

