

**UNIVERZITET U BEOGRADU
ELEKTROTEHNIČKI FAKULTET**

Katedra za elektroniku

Predmet: Računarska elektronika



Projekat u assembleru

13 projekat (G7)

Rok za predaju: 16.05.2016.

Projekat radili:

Ime	Prezime	broj indeksa
Radosav	Vulikić	552/2013
Nikola	Miljković	225/2012

Tekst zadatka: Napisati program koji iscrtava mali kvadrat na ekranu korišćenjem nekoliko blokova (ASCII) u boji koja se zadaje na početku programa na standardnom ulazu. Moguće je zadati tri boje unosom sledećih karaktera: *R-red*, *G-green*, *B-blue*. Pomerati kvadrat po ekranu na osnovu random generisanih smerova. koristiti fiksno kašnjenje od 50 ms.

Objašnjenje koda:

Na početku definišemo osnovne promenljive, kao što su veličina bafera, ivice prozora, string koji se ispisuje na početku izvršavanja programa, string koji sadrži naziv programa. Definišemo i neoznačene promenljive, na primer: `stdInHandle`, `stdOutHandle` - čuvaju informacije o tome da li se radi o upisu ili ispisu, memoriju rezervisanu za bafer, broj pročitanih bajtova...

Program očekuje unos za određivanje boje (r-crvena, g-zelena, b-plava) ili u slučaju da nije unet validan znak, zahteva ponovno unošenje. Bafer je veličine 1byte tako da je u baferu (memorija dodeljena baferu) sve vreme samo jedan karakter. Ukoliko je u baferu validan karakter prelazimo na postavljanje boje prozora i karaktera koji će biti ispisani. U suprotnom ukoliko karakter u baferu nije validan nakon provere vraćamo se na početak programa i ponovo tražimo od korisnika da unese karakter.

Unutar petlje `loop1` postavljamo "handle" za ispis, zatim vidljivost kursora na nevidljiv, postavljamo naziv prozora na vrednost stringa definisanog na početku. Postavljamo početnu poziciju kursora. Započinjemo iscrtavanje kvadrata tako što na početnoj poziciji iscrtamo jedan karakter ASCII vrednosti `DBh`, zatim pomerimo cursor na poziciju čija je x koordinata veća za jedan i ponovimo iscrtavanje karaktera vrednosti `DBh`. Nakon toga pozivamo funkciju koja pauzira izvršavanje programa za definisan broj mili sekundi (definisan na početku programa).

Nakon toga se obrnutim redom briše nacrtani kvadrat. Ispitujemo da li je korisnik pritisnuo taster `SPACE` kojim bi se program završio, ukoliko nije određuju se nove nasumične koordinate sledećeg kvadrata koje treba da budu u opsegu prozora. Nakon toga skačemo na početak petlje `loop1` i ponavljamo iscrtavanje kvadrata.

U trenutku kada korisnik pritisne taster `SPACE` skačemo na `loopexit` gde završavamo program.

Kod programa:

```
INCLUDE Irvine32.inc
INCLUDE macros.inc

;-----

buffer_size = 1      ;//Velicina bafera
drawDelay = 50       ;//Kasnjenje izmedju sukcesivnih iscrtavanja
square = 219 ;//Tip simbola koji se crta, njegova ASCII vrednost u decimalnim brojevima
;-----

;//Definisanje velicine prozora
xmin = 0             ;//leva ivica
xmax = 49            ;//desna ivica
ymin = 0             ;//gornja ivica
ymax = 29            ;//donja ivica
;-----

;//-----Promenljive koje koristimo u programu-----

.data
string byte "Welcome, this is the dancing square program.", 0dh, 0ah,
"To terminate this program press SPACE key or close the window.", 0dh, 0ah,
"You can choose your square color by typing one of the characters:", 0dh, 0ah,
"r- for red, b- for blue and g- for green.", 0dh, 0ah, 0; //String koji opisuje funkciju
;//programa

windowRect SMALL_RECT <xmin, ymin, xmax, ymax>;//Velicina prozora
wintitle byte "Dancing square", 0 ;//Naslov programa
cursorInfo CONSOLE_CURSOR_INFO <> ;//Informacije o kursoru

.data?
buffer byte buffer_size dup(?)
stdInHandle handle ?
bytesRead dword ?
stdOutHandle handle ?

;-----Pocetak programa-----

.code
main proc

;-----Ispisivanje teksta na pocetku programa-----

loopstr:
    call Clrscr; //Brisemo sve iz prozora (ukoliko je ovo druga iteracija)
    mov edx, offset string; //Postavljamo pokazivac na pocetak stringa
    call WriteString; //Ispisujemo string

    invoke getstdhandle, std_input_handle; // Postavlja handle za upis
    mov stdInHandle, eax

    invoke ReadConsole, stdInHandle, ADDR buffer,
    buffer_size, ADDR bytesRead, 0; // Cita podatak iz konzole, stavlja ga u bafer
```

```

mov esi, offset buffer;    //pokazivac na bafer
mov ecx, bytesRead;        //broj procitanih bajtova
mov ebx, type buffer;      //tip/velicina bafera

; //Odredjivanje boje kvadrata-----

.if buffer == 'r'          ; //Ispitujemo da li je u baferu vrednost 'r'
    mov ax, red + (white * 16) ; // AL koristimo za postavljanje boje kvadrata,
                                ; // AH za postavljanje boje pozadine
.elseif buffer == 'b'
    mov ax, blue + (white * 16)
.elseif buffer == 'g'
    mov ax, green + (white * 16)
.endif

jnz loopstr    ; //Ukoliko je unet karakter koji ne pripada skupu {'r','b','g'}
                ; //ponovo se vracamo na pocetak programa sve dok korisnik
                ; //ne unese korektnu vrenost

call SetTextColor ; //Postavljamo zeljenu boju kvadrata i pozadine
call Clrscr

; //-----

INVOKE GetStdHandle, STD_OUTPUT_HANDLE;    //Postavlja handle za ispis podataka
mov    stdOutHandle, eax

INVOKE GetConsoleCursorInfo, stdOutHandle, ADDR cursorInfo; //Cita trenutno stanje
                                                ; //kursora
mov    cursorInfo.bVisible, 0;    // Postavlja vidljivost kursora na nevidljiv
INVOKE SetConsoleCursorInfo, stdOutHandle,
    ADDR cursorInfo;    //Postavlja novo stanje kursora

; //----Postavlja ime prozora-----

INVOKE SetConsoleTitle, ADDR wintitle

; //----Pocetna pozicija kvadrata-----

mov dl, 20
mov dh, 10

; //-----

loop1:

; // iscrtavanje kvadrata
call gotoxy;    //Pomera kursor na pozicije zadate sa: x = dl, y = dh
mov al, square; //Upisuje vrednost simbola square (DBh=219dec) u al
call writechar; //Ispisuje simbol square(ASCII DBh) na poziciju kursora x,y
inc dl;        //Povecava koordinatu x za 1
call gotoxy;   //Pomera kursor na novu poziciju
mov al, square; //Upisuje vrednost simbola square (DBh=219dec) u al
call writechar; //Ispisuje simbol square(ASCII DBh) na poziciju kursora x+1,y
; //-----

mov    eax, drawDelay; //Upisuje duzinu pauze (u mili sekundama) u EAX
call Delay;    //Pauzira trenutni proces u trajanju definisanom vrednoscu u EAX

```

```

; //Brisanje kvadrata-----

call gotoxy;           //Pomera kursor na poslednju poziciju iscrtavanja (x=dl,y=dh)
mov al, ' ';           //U al upisuje vrednost simbola " " (ASCII 20h)
call writechar;        //Ispisuje simbol " " na poziciju kursora x,y
dec dl;                //Smanjuje vrednost koordinate x za 1
call gotoxy;           //Pomera kursor na poziciju x-1,y
mov al, ' ';
call writechar

; //Promena velicine prozora-----
INVOKE SetConsoleWindowInfo,
        stdoutHandle, TRUE, ADDR windowRect

; //Prekidanje rada programa-----
INVOKE GetKeyState, VK_SPACE; //Ukoliko je pritisnut taster SPACE, postavlja
                                ; //najvisi bit
test eax, 80000000h           ; // EAX registra na "1", test proverava da li je
                                ; //taj bit na jedinici
.if !zero? ; // i ako jeste skacemo na loopexit cime se rad programa završava
    jmp loopexit
.endif
; //-----

; //Dodeljivanje nasumicnih koordinata-----
mov eax, xmax;               //Opseg nasumicnog dodeljivanja rand(0,xmax)
call randomrange;            //Vraca nasumicnu vrednost iz dodeljenog opsega
mov dl, al;                  //Upisuje dobijenu vrednost u dl (x koordinata)
mov eax, ymax;               //Opseg nasumicnog dodeljivanja rand(0,ymax)
call randomrange;            //Vraca nasumicnu vrednost iz dodeljenog opsega
mov dh, al;                  //Upisuje dobijenu vrednost u dl (y koordinata)
; //-----

jmp loop1

loopexit:
    INVOKE ExitProcess,0

main ENDP
END main

```