

UNIVERZITET U BEOGRADU ELEKTROTEHNIČKI FAKULTET

Katedra za elektroniku

Predmet: Računarska elektronika



Projekat 9: Analiza ulaznog fajla

Projekat radili:

Ime	Prezime	broj indeksa
Aleksandra	Brkić	2013/0550
Nevena	Stojnić	2013/0245

ZADATAK: *Napisati program koji će analizirati ulazni fajl. Svaki red u ulaznom fajlu je sekvenca karaktera: $x0\ y0\ x1\ y1$ boja, gde su $x0, y0$ koordinate gornjeg desnog ugla kvadrata, $x1, y1$ koordinate donjeg desnog ugla kvadrata, boja – boja kvadrata. Na osnovu sadržaja fajla iscrtava se odgovarajući broj kvadrata na ekranu.*

Kako bi se ostvarile tražene funkcionalnosti izvršavanje programa trebalo bi da sadrži sledeći niz koraka:

1. Učitavanje fajla u kome su sprecificirane koordinate i boje kvadrata koje treba iscrtati:
 - 1.1 Korisnik sa standardnog ulaza unosi ime ulaznog fajla
 - 1.2 Proveravamo da li je fajl ispravan i da li je bafer dovoljno veliki da se učita čitav fajl
 - 1.3 Pravila za pisanje ulaznog fajla shodno zahtevanim funkcionalnostima i usvojenim projektnim odlukama data su u prilogu 2.
2. Ukoliko je fajl ispravan i čitanje je prošlo uspešno, potrebno je pristupiti obradi učitanoog fajla. To se obavlja sledećim nizom koraka:
 - 2.1 U eksternoj petlji analiziraće se jedna po jedna linija ulaznog fajla kako bi se dobile sve potrebne informacije za iscrtavanje kvadrata
 - 2.2 Kako bismo dobili koordinate i boju kvadrata potrebna nam je funkcija koja će vršiti konverziju ASCII karaktera u int i 5 bajtovskih lokacija u kojima će se čuvati podaci dobijeni iz linije.
3. Nakon što smo dobili potrebne informacije za crtanje kvadrata potrebno je izvršiti iscrtavanje kvadrata sledećim nizom koraka:
 - 3.1 Podešavanje boje teksta korišćenjem procedure SetTextColor iz Irvine32 biblioteke
 - 3.2 Pomeranjem kursora na određeno mesto i ispisivanjem solid block karaktera zarad formiranja kvadratne structure. (Napomena. Zbog činjenice da solid block nije kvadratnog oblika ni ova struktura neće biti potpuno kvadratna.)
 - 3.3 Dužina stranice kvadrata određuje se kao $y1 - y0 + 1$
4. Kada smo završili sa iscratavanjem potrebno je vratiti se na korak 2.2 osim ukoliko smo došli do kraja ulaznog fajla, u tom slučaju potrebno je završiti sa izvršavanjem programa tako da se čeka da korisnik unese neki karakter sa standardnog ulaza čime će se zatvoriti prozor.

U prlogu 1 nalazi se asemblerski kod koji prikazuje realizovane funkcionalnosti. Detaljniji komentari dati su u kodu.

U prilogu 3 nalazi se igled konzolnog prozora nakon izvršavanja programa.


```

file_opened:
;Ucitavanje fajla u bafer
    mov edx, OFFSET buffer
    mov ecx, BUFFER_SIZE
    call ReadFromFile
    mov bytesInFile, eax           ;cuvamo broj procitanih bajtova
    jnc check_buffer_size         ;proveravamo da li je doslo do greske pri citanju
    mWrite "Greska: Doslo je do greske pri citanju fajla!";ako jeste - prikazuje se poruka
    call WriteWindowsMsg
    jmp close_file

check_buffer_size:
    cmp eax, BUFFER_SIZE         ;proveravamo se da li je ulazni bafer dovoljno veliki
    jb drawing                   ;ako jeste - pocinje iscertavanje!
    mWrite "Greska: Broj karaktera u fajlu prevazilazi velicinu bafera! Ucitajte manji fajl"
    call WriteWindowsMsg         ;ukoliko nije ispisuje se greska
    jmp quit

;zatvorimo file
close_file:
    mov eax, fileHandle
    call CloseFile

;iscrtavanje!
drawing:
    mov eax,0
    mov ebx,0
    mov ecx,0
    mov edx,0

    ;brisanje sadrzaja konzole i podesavanja kursora
    call Clrscr
    INVOKE GetStdHandle, STD_OUTPUT_HANDLE
    mov outHandle, eax
    INVOKE GetConsoleCursorInfo, outHandle, ADDR cursorInfo
    mov cursorInfo.bVisible,0
    INVOKE SetConsoleCursorInfo, outHandle, ADDR cursorInfo

    mov eax,0
    mov esi, OFFSET buffer           ;u ESI se stavlja adresa buffer-a

    ReadOneLine:
                                ;citamo koordinate jednog kvadrata
        call convert_ascii_to_int    ;ucitavamo koordinate
        cmp bl, ' ' ;svaki put kada se skoci na ReadLine ocekuje se da je u bl cifra osim ako
                                ;je kraj fajla
        jz ifNotEnd ;ukoliko nije kraj fajla nakon izlaska iz procedure u bl ce se nalaziti
        jmp quit ;' ', u suprotnom smo dosli do kraja i treba završiti program

    ifNotEnd:
        mov x0_cord, al             ;ucitavamo x0
        inc esi

        call convert_ascii_to_int    ;ucitavamo y0
        mov y0_cord, al
        inc esi

        call convert_ascii_to_int    ;ucitavamo x1
        mov x1_cord, al
        inc esi

        call convert_ascii_to_int    ;ucitavamo y1
        mov y1_cord, al
        inc esi

```

```

        call convert_ascii_to_int          ;ucitavamo boju
        mov color, al
        inc esi

        mov bl, [esi]                    ; u bl ucitavamo ono sto je na adresi na koju ukazuje esi
        cmp bl, ' '                      ; uporedjuje bl sa razmakom
        jz ReadSpecialCharacters
        cmp bl, 0ah
        jz ReadSpecialCharacters
        cmp bl, 0dh                      ;uporedjuje bl sa 0ah
        jz ReadSpecialCharacters

ReadSpecialCharacters:
        inc esi                          ;posto je u bl ili ' ' ili \r ili \n inkrementiramo esi
        mov bl, [esi]                    ;pomeramo u bl, [esi]

        cmp bl, 0ah                      ;proveravamo da li je \n
        jz ReadSpecialCharacters

        cmp bl, 0dh                      ;proveravamo da li je \r
        jz ReadSpecialCharacters

        cmp bl, ' '                      ;proveravamo da li je ' '
        jz ReadSpecialCharacters

        jmp DrawRect                     ;kada smo ucitali liniju, idemo na iscrtavanje

DrawRect:
        mov dh, y0_cord                  ;u dh stavljamo y cursor position

        DrawVertical:                     ;iscrtavanje po vertikali
            mov eax, 0
            mov al, color                  ;postavljamo boju kvadrata koji treba iscrtati
            call SetTextColor

            mov eax, 0
            mov dl, x0_cord                 ;u dl stavljamo x cursor position
            mov al, y1_cord
            sub al, y0_cord
            add al, 1
            mov ecx, eax ;u brojac stavljamo duzinu stanice po x osi (jer je ista kao po y)
            mov al, 0DBh ;solid-block

            DrawHorizontal:                ;iscrtavamo po x osi ecx puta
                call Gotoxy
                call WriteChar
                inc dl
                loop DrawHorizontal
            cmp dh, y1_cord
            jz ReadOneLine

            inc dh
            jmp DrawVertical

quit:
        mov al, 0ah                      ;zavrsavamo program
        mov eax, 15                       ;menjamo broju teksta na belu
        call SetTextColor
        call ReadChar                      ;ceka na karakter sa ulaza da bi zavrsio izvorsavanje
        exit

main endp
end main

```

PRILOG 2. Pravila za pisanje ulaznog fajla

- Linija ulaznog fajla treba da bude oblika:

x0 y0 x1 y1 boja

pri čemu su x0 i x1 koordinate gornjeg i donjeg desnog ugla i po pravilu bi trebalo da budu jednake.

- Boju u ulaznom fajlu treba specificirati prema sledećoj tabeli:

KOD	BOJA	KOD	BOJA
0	crna	8	siva
1	plava	9	svetlo plava
2	zelena	10	svetlo zelena
3	cijano plava	11	svetlo cijano plava
4	crvena	12	svetlo crvena
5	magenta	13	svetlo magenta
6	braon	14	žuta
7	svetlo siva	15	bela

PRILOG 3. Izgled konzolnog prozora nakon isrtavanja kvadrata specificiranih u TEST00.txt

- U prilogu se nalazi test fajl TEST00.txt, a rezultat izvršavanja programa kada se za ulazni fajl prosledi TEST00.txt prikazan je na slici ispod.



