

**UNIVERZITET U BEOGRADU**  
**ELEKTROTEHNIČKI FAKULTET**

*Katedra za elektroniku*

*Predmet: Računarska elektronika*



**Projekat 11: Pomeranje kursora**  
**2015/2016**

**Čarna Đoković 300/2012**

**Jelena Živković 134/2012**

## 1. Tekst projekta:

Koristeći tastere na tastaturi pomerati kursor u četiri pravca. Pritiskom tastera B na tastaturi kursor se pomera na donju ivicu po sredini prozora, tastera U na gornju ivicu po sredini, L levo po sredini i R desno po sredini. Pritiskom tastera ENTER završava se izvršavanje programa. Sve pritisnute tastere je potrebno čuvati u memoriji i po izvršavanju programa sve akcije zapisati u odgovarajući tekstualni log fajl.

## 2. Realizacija projekta:

Pisanje programa smo počele od glavnog zahteva projekta koji je da se pritiskom na taster kursor pomera u željenom smeru u zavisnosti od toga koji je taster pritisnut. Počele smo sa pisanjem procedure *moveit* koja obuhvata nekoliko funkcija koje smo više puta koristile. Prethodno smo definisale širinu i visinu prozora kao i početnu poziciju kursora na (0,0) (što odgovara koordinatama x i y). Sada ćemo reći nešto malo više o svakoj od ovih funkcija.

**Get key:** ova funkcija služi za kontrolu tasterima, poredi uneti karakter sa tastature sa sadržajem registra al i u zavisnosti od toga poziva posebnu funkciju za pomeranje u smeru koji je odabran.

**moveup, movedown, moveleft, moveright:** Ovo su funkcije koje vrse pomeranje kursora i rade na istom principu. U registre dl i dh upisujemo trenutnu poziciju kursora, umanjujemo ili uvećavamo register za 1 u zavisnosti od toga da li je potrebno da idemo gore ili dole, odnosno levo i desno. Nakon svake promene registra za koordinatu koja nam je bitna za taj pravac (kod pomeranja na gore i dole to je y a kod pomeranja u levo i desno to je x) proveravamo da li je unutar granica prozora, ako izlazi iz granica tu koordinatu ostavljamo na tom mestu sve dok ne dođe do drugog tastera koji će je pomeriti u nekom drugom smeru a ne van granica. Koristili smo funkciju *gotoxy* koja kao argumente prima dl i dh i postavlja kursor na željenu poziciju.

**exit1:** Služi za izlazak iz programa u slučaju pritiska tastera ENTER.

**setCursor:** Postavlja kursor.

Tokom celog procesa pomeranja, na svaki pritisak tastera, on se upisuje karakter po karakter u output.txt fajl sve dok ne dođe do izlaza iz programa.

### 3. Kod projekta

```
;;;;;;;;;;;;; PROJEKAT IZ RACUNARSKE ELEKTRONIKE ;;;;;;;;;;;;;;
;;;;;;;;;;;;; Carna Djokovic 300/2012
;;;;;;;;;;;;; Jelena Zivkovic 134/2012
;;;;;;;;;;;;; maj 2016
```

```
INCLUDE Irvine32.inc
```

```
; Velicina prozora
xmin = 0
xmax = 79
ymin = 0
ymax = 24
```

```
BUFFER_SIZE = 501
```

```
.DATA
```

```
row db ?
```

```
col db ?
```

```
buffer BYTE BUFFER_SIZE DUP( ? )
```

```
filename BYTE "output.txt", 0
```

```
fileHandle HANDLE ?
```

```
stringLength DWORD ?
```

```
bytesWritten DWORD ?
```

```
str1 BYTE "Cannot create file", 0dh, 0ah, 0
```

```
titleStr BYTE "Projekat iz racunarske elektronike: 11", 0
```

```
.code
```

```
main PROC
```

```
INVOKE SetConsoleTitle, ADDR titleStr
```

```
; Pravljenje novog fajla
```

```
mov edx, OFFSET filename
```

```
call CreateOutputFile
```

```
mov fileHandle, eax
```

```
; Proverava greske pri kreiranju fajla
```

```
cmp eax, INVALID_HANDLE_VALUE
```

```
jne setCursor
```

```
mov edx, OFFSET str1; prikazi gresku
```

```
call WriteString
```

```
jmp quit
```

```
setCursor :
```

```
mov dh, 0; postavlja kursor na nulu
```

```
mov dl, 0
```

```
call gotoxy
```

```
mov row, dh
```

```
mov col, dl
```

```
call moveit
```

```
exit
```

```

main ENDP

moveit proc
Get_key :
call readchar
cmp al, 'u'
je moveup
cmp al, 'b'
je movedown
cmp al, 'l'
je moveleft
cmp al, 'r'
je moveright
cmp al, 0Dh; 0D je oznaka za enter
je exit1
jmp get_key

moveup :
call log_action

mov dh, row; dh je za y koordinatu
sub dh, 1; kada se pomera na gore mora da se smanji
mov dl, col
.IF dh < ymin || dh > ymax
mov dh, ymin
call gotoxy
mov row, dh; stavi sada nove vrednosti u row i col
mov col, dl
jmp get_key
.ENDIF
call gotoxy
mov row, dh
mov col, dl
jmp get_key; vrati se da cekas sledeci taster

movedown :
call log_action
mov dh, row
add dh, 1; kada se pomera na dole mora da se uvecava
mov dl, col
.IF dh <= ymin || dh >= ymax
mov dh, ymax
call gotoxy
mov row, dh; stavi sada nove vrednosti u row i col
mov col, dl
jmp get_key
.ENDIF
call gotoxy
mov row, dh
mov col, dl
jmp get_key

```

```

moveright :
call log_action
mov dl, col
add dl, 1
mov dh, row
.IF dl <= xmin || dl >= xmax
mov dl, xmax
call gotoxy
mov row, dh
mov col, dl
jmp get_key
.ENDIF
call gotoxy
mov row, dh
mov col, dl
jmp get_key

moveleft :
call log_action
mov dl, col
sub dl, 1
mov dh, row
.IF dl <= xmin || dl >= xmax
mov dl, xmin
call gotoxy
mov row, dh
mov col, dl
jmp get_key
.ENDIF
call gotoxy
mov row, dh
mov col, dl
jmp get_key

exit1:
call CloseFile
ret
moveit endp
log_action proc

mov edi, OFFSET buffer
stosb; upisuje karakter u memoriju iz al registra
mov eax, fileHandle
mov edx, OFFSET buffer
mov ecx, 1; jedan po jedan karakter
call WriteToFile

ret
log_action endp

quit :
exit

END main

```