

Ve`ba 8

PREKIDI TASTATURE O^ITAVANJE STATUSA TASTATURE

Uvek kada pritisnemo ili otpustimo neki taster na tastaturi ona generi`e jednobajtni broj, skan kod, koji jednozna`no defini`e taster. Tastatura generi`e razli`it skan kod kada se pritisne ili otpusti razli`iti taster. Uvek kada pritisnemo neki taster, skan bajt uzima vrednost od 1 do 83, a kada ga otpustimo tastatura generi`e skan kod za 128 ve`i od skan koda pritisnutog tastera. Tako naprimer kada se pritisne slovo Z, tastatura generi`e skan kod 44, a kada ga otpustimo taj kod `e biti $44+128=172$.

Tastatura sa ROM BIOS-om komunicira preko porta za prekide i to preko prekida broj 9. O`ekivani skan kod se {alje do BIOS-a preko uslu`nog programa za tastaturu u okviru dvo bajtnog koda, pri `emu ni`i bajt sadr`i ASCII vrednost tastera, a vi`i njegov skan kod vode`i i ra`una koji je od funkcijski tastera aktivan.

Postoje dva standardna status-bajta tastature sme`tena na memorijskim lokacijama 417H i 418H. Oni sadr`e informacije koji je od funkcijskih tastera pritisnut.

Na adresi 0417H nalazi se statusni bajt tastature `iji bitovi imaju slede`a zna`enja:

Bitovi	Zna`enje
7	INSERT taster
6	CAPS LOCK
5	NUM LOCK
4	SCROLL LOCK
3	pritisnut ALT
2	pritisnut CTRL
1	pritisnut levi SHIFT
0	pritisnut desni SHIFT

Kada je neki od tastera uklju`en na tom mestu biti `e postavljen 1 ina`e je nula.

Prekidi tastature vr`e se upotrebom interapta 16H sa slede`im njegovim funkcijama:

- 00H `itanje znaka sa tastature
- 01H `itanje statusa tipke na tastaturi
- 02H o`itavanje statusa tastature
- 03H postavljanje zvu`nog efekta kod pritiska tastera
- 05H slanje SCAN CODA i ASCII koda znaka
- 10H `itanje karaktera iz pro`irene tastature
- 11H `itanje statusa pro`irene tastature
- 12H `itanje indikatora pro`irene tasture

Poziv ovog interapta i odgovaraju`e funkcije vr`i se na taj na`in {to se u registar AH upi`e funkcija koju `elimo i instrukcijom INT 16H pozivamo ovaj prekid. Obi`no se rezultat operacije vra`a u registar AL.

Da bi upoznali navedene funkcije re{imo slede}i zadak:

ZADATAK:

Napisati program kojim se prikazuje koji je od funkcijski tastera pritisnut na tastaturi.

RE{ENJE:

Iz navedenih funkcija vidimo da funkcijom 02H mo`emo o-itati indikatore tastature odnosno sadr`aj indikator bajta tastature sa adrese 417H bi}e preme{ten u registar AL sa bitovima ~ija su zna-enja data prethodnom tabelom. Su{tina daljeg re{enja da pravilno tuma~imo pojedine bite ovog registra.

U data segmentu smo uveli tabelu statusa (STAT) ~ija je du`ina 8 bajta. Da bi o-itali bit po bit registra AL koristi}emo instrukciju {iftovanja u levo (SHL) kojom izba~eni bit ulazi u fleg prenosa CF. Ispitivanje statusa ovog fleg bita vr{imo instrukcijom JNC (skok ako CF nije jednako nuli) upisujemo u tabelu statusa 1 ili 0, pri ~emu moramo dodati ASCII '0' kako bi se broj mogao prikazati na ekranu.

Nakon osam pomeranja (ispitivanja svih osam bita) pozivamo proceduru PRIKAZ1 kako bi dobijene rezultate prikazali na ekranu.

Primetite da smo u programu koristili i funkciju Oh ovog interapta. U slu~aju da je pritisnut ESC taster ~iji je ASCII kod 27 zavr{avamo program (JMP KRAJ) ina-e nastavljamo dalje ispitivanje.

Rezultat:

OCITAVANJE STATUSA TASTATURE

```
Ukljucen INSERT      : 1
Ukljucen CAPS LOCK   : 1
Ukljucen NUM LOCK    : 1
Ukljucen SCROLL LOCK : 0
Pritisnut ALT        : 0
Pritisnut CTRL       : 0
Pritisnut levi SHIFT : 1
Pritisnut desni SHIFT: 0
```

```

TITLE prog8 ;Ocitavanje indikatora status tastature
;
sseg segment stack
    db 256 dup (?) ;Rezervisanje prostora za stek
sseg ends
;
dseg segment
    poruka    db 'OCITAVANJE STATUSA TASTATURE$'
    poruka1   db 'Ukljucen INSERT      : '
    pp11      db 1 dup(?)
    pp12      db '$'
    poruka2   db 'Ukljucen CAPS LOCK    : '
    pp21      db 1 dup(?)
    pp22      db '$'
    poruka3   db 'Ukljucen NUM LOCK      : '
    pp31      db 1 dup(?)
    pp32      db '$'
    poruka4   db 'Ukljucen SCROLL LOCK : '
    pp41      db 1 dup(?)
    pp42      db '$'
    poruka5   db 'Pritisnut ALT          : '
    pp51      db 1 dup(?)
    pp52      db '$'
    poruka6   db 'Pritisnut CTRL          : '
    pp61      db 1 dup(?)
    pp62      db '$'
    poruka7   db 'Pritisnut levi SHIFT : '
    pp71      db 1 dup(?)
    pp72      db '$'
    poruka8   db 'Pritisnut desni SHIFT: '
    pp81      db 1 dup(?)
    pp82      db '$'
    odg       db 1 dup (?)
    stat      db 8 dup (?)
dseg ends
;
cseg segment
;
assume cs:cseg, ds:dseg, ss:sseg
;
;Pocetak glavnog programa
;
status proc far
;
    push ds                ;Povratna adresa segmenta ide u stek
    mov ax, 0
    push ax                ;Nula-povratna adresa ide u stek
    mov bx, dseg
    mov ds, bx

    mov ah, 00            ;Postavljanje video moda
    mov al, 3
    int 10h

    mov ah, 11           ;Postavljanje palete boja
    mov bh, 0
    mov bl, 4
    int 10h
    call prikaz1
pon:
    mov al, 0
    mov ah, 02h          ;Poziv interapta za ocit. statusa
    int 16h

```

```

        mov cx, 8                ;Postavljanje brojaca
        mov si, 1
pon1:   shl  al,1
        jnc lb1
        mov stat[si], 1         ;Ispitivanje bita po bit
        jmp lb2
lb1:    mov  stat[si], 0
lb2:    add  stat[si], '0'       ;Pretavaranje u ASCII kod
        inc  si
        loop pon1
        call prikaz1

        mov dl, 40
        mov dh, 19
        call pozicija
        mov ah, 0h
        int 16h
        mov odg, al
        mov al, odg
        cmp al, 27
        je  kraj
        jmp pon

kraj:   ret
status endp
include bibl.asm
cseg ends
        end status

```