

# BİL 362 Mikroişlemciler

---

Hazırlayan: M.Ali Akcayol  
Gazi Üniversitesi  
Bilgisayar Mühendisliği Bölümü



## Konular

### Mikroişlemci Programlama

- Makrolar
- BIOS ve DOS Fonksiyonları
- Klavye Kullanımı
  - Klavyeden bir karakter alma
  - Klavyeden bir string alma
- Ekran Fonksiyonları
  - Ekrana bir karakter yazma
  - Ekrana bir string yazma
  - Ekran ayarları
- Mouse Kullanımı

## Makrolar

- Makrolar prosedürler gibi bir işi yapmak için kullanılırlar.
- Prosedürler **call** ile çağrılır makrolar sadece ismi yazılarak çağrılır.
- Prosedürler programın sonunda bulunur, makrolar kullanıldığı yerden önce bulunur.
- Bir makro **MACRO** deyimiyle **ENDM** deyimi arasındaki komutlardan oluşur.
- Prosedürlerde olduğu gibi **MACRO** deyiminden önce bir isim girilmelidir.
- **MACRO** deyiminden sonra dışarıdan giriş yapmak için parametreler tanımlanabilir.

## Makrolar - Örnek

```
02  TOPLA MACRO SAYI1,SAYI2,SAYI3,SAYI4,SAYI5,SONUC
03      XOR AX,AX
04      ADD AX,SAYI1
05      ADD AX,SAYI2
06      ADD AX,SAYI3
07      ADD AX,SAYI4
08      ADD AX,SAYI5
09      MOV SONUC,AX
10  ENDM
11
12  org 0100h
13  TOPLAM dw 0000h
14
15  TOPLA 10,20,30,40,50,TOPLAM
16  mov cx,TOPLAM
17  mov [0110h],cx
18
19  TOPLA 12,23,34,45,56,TOPLAM
20  mov cx,TOPLAM
21  mov [0112h],cx
22
23  TOPLA 13,25,46,57,68,TOPLAM
24  mov cx,TOPLAM
25  mov [0114h],cx
26
27  TOPLA 11,22,33,44,55,TOPLAM
28  mov cx,TOPLAM
29  mov [0116h],cx
30
31  TOPLA 98,87,76,65,54,TOPLAM
32  mov cx,TOPLAM
33  mov [0118h],cx
34
35  hlt
```

## BIOS ve DOS Fonksiyonları

- INT komutuyla bazı özel işlemler için gerekli BIOS ve DOS fonksiyonları kullanılabilir.
- INT komutu çalıştırıldığında önce CS:IP ve flag değerlerini stack'a push eder.
- Bir interrupt çalıştırılmadan önce kullanacağı register'lara ilgili değerlerin aktarılması gerekir.
- Her bir interrupt için çok sayıda alt fonksiyon tanımlıdır. Hangi fonksiyonun çalışacağı AH register'ının değerine göre belirlenir.

## Klavye Kullanımı

Ekrana yazdırarak (echo) bir karakter okuma

- INT 21H / AH=01h alt fonksiyonuyla kullanılır.
- Okunan karakter AL register'ına kaydedilir.

```
; program 'd' veya 'D' girene kadar calisir
org 0100h
mov ah,1 ; klavyeden bir karakter okuma
yenikarakter:
    int 21h
    cmp al,44h ; D
    je dogru
    cmp al,64h ; d
    je dogru
    jmp yenikarakter
dogru:
hlt
```

## Klavye Kullanımı

### Ekrana yazmadan bir karakter okuma

- INT 21H / AH=06h alt fonksiyonuyla kullanılır. Okuma işlemi için dl register'ına 0FFh değeri aktarılır.
- Okunan karakter AL register'ına kaydedilir. Ancak ekranda gösterilmez. AH=1 deki gibi beklenmez işlemci bekleme sırasında başka iş yapabilir.

```
; sifre girisi
org 0100h
yenikarakter:
    mov ah,6      ; klavyeden bir karakter okuma
    mov dl,0FFh
    int 21h
    cmp al,00h    ; bos ise yazma
    je yenikarakter
    mov ah,2      ; int 21h, ah=2 fonksiyonu dl'yi yazar
    mov dl,2Ah
    int 21h
    jmp yenikarakter
hlt
```

## Klavye Kullanımı

### Ekrana yazmadan bir karakter okuma - devam

```
; kucuk harfi buyuk harfe ceviris
org 0100h
yenikarakter:
    mov ah,6      ; klavyeden bir karakter okuma
    mov dl,0FFh
    int 21h
    cmp al,61h    ; a
    jb yaz
    cmp al,7Ah    ; z
    jbe buyuk     ; kucuk harf buyuk yap
    jmp yenikarakter
    buyuk:
        sub al,20h ; AL-20h ayni karakterin buyugu
    yaz:
        ; ekrana yazdirma islemleri
        cmp al,00h ; bos ise yazma
        je yenikarakter
        mov ah,2   ; int 21h, ah=2 fonksiyonu dl'yi yazar
        mov dl,al
        int 21h
        jmp yenikarakter
hlt
```

## Klavye Kullanımı

### Klavyeden string okuma

- INT 21H / AH=0Ah alt fonksiyonuyla kullanılır. Okunan string en çok 255 karakter olabilir.
- Enter yapılana kadar veya 255 karakter olana kadar alınan string'i DS:DX adresine kaydeder. İlk byte buffer boyutunu ikinci byte girilen karakter sayısını saklar.

```
org 100h
buffer db 10,?,10 dup(' ') ; buffer boyutu 10
mov dx, offset buffer      ; ds:dx degeri buffer yapildi
mov ah, 0Ah                ; int 21h, ah=0Ah,
int 21h                    ; ds:dx'e string okur
xor bx, bx
mov bl, buffer[1]          ; karakter sayisi aliniyor
mov buffer[bx+2], '$'      ; int 21h, ah=9, ds:dx adresinde
mov dx, offset buffer+2    ; $ okuyana kadar ekrana yazar
mov ah, 9
int 21h
hlt
```

## Klavye Kullanımı

### Klavyede bir tuşa basıldığını kontrol etmek

- INT 16H / AH=01h alt fonksiyonuyla bir tuşa basıldığında buffer'a alınır.
- INT 16H / AH=00h alt fonksiyonuyla buffer'daki karakter AL'ye alınır.

```
org 0100h
mov dx, offset mesaj      ; int 21h, ah=09h fonksiyonu
mov ah, 9                 ; $ okuyana kadar ekrana yazar
int 21h
basadon:
mov ah, 01                ; int 16h, ah=01 tusa basildigini kontrol eder
int 16h
mov ah, 00                ; int 16h, ah=00 karakteri buffer'dan al'ye aktarin
int 16h
jz cikis
jmp basadon
cikis:
mesaj db "cikis icin bir tusa basiniz...$"
hlt
```

## Ekran Fonksiyonları

### Ekrana bir karakter yazdırma

- INT 21H / AH=02h alt fonksiyonuyla kullanılır. Yazılacak karakter DL register'ındadır.

```
org 0100h
mov ah, 2
mov dl, 'I' ; ekrana 'I' karakteri yazar
int 21h
mov dl, 'N' ; ekrana 'N' karakteri yazar
int 21h
mov dl, 'T' ; ekrana 'T' karakteri yazar
int 21h
mov dl, 'E' ; ekrana 'E' karakteri yazar
int 21h
mov dl, 'L' ; ekrana 'L' karakteri yazar
int 21h
mov dl, ' ' ; ekrana ' ' karakteri yazar
int 21h
mov dl, '8' ; ekrana '8' karakteri yazar
int 21h
mov dl, '0' ; ekrana '0' karakteri yazar
int 21h
mov dl, '8' ; ekrana '8' karakteri yazar
int 21h
mov dl, '6' ; ekrana '6' karakteri yazar
int 21h
hlt
```

## Ekran Fonksiyonları

### Ekрана string yazdırma

- INT 21H / AH=09h alt fonksiyonuyla kullanılır.
- Yazılacak string'in offset adresi dx register'ındadır.
- String sonunun \$ ile bitirilmesi gerekir.

```
org 0100h
mov dx, offset kitapadi ; int 21h, ah=09h fonksiyonu
mov ah, 9 ; $ okuyana kadar ekrana yazar
int 21h
ret
kitapadi db "Ah Istanbul $" ; buffer boyutu 10
hlt
```



## Ekran Fonksiyonları

### Ekran ayarları - ekran boyutu değiştirme

- INT 10H / AH=00h alt fonksiyonuyla kullanılır.
- İstenen ekran boyutu AL register'ındadır.  
AL = 00h için, boyut=40x25, renk=16 renk, sayfa sayısı=8 sayfa.  
AL = 03h için, boyut=80x25, renk=16 renk, sayfa sayısı=8 sayfa.  
AL = 13h için, boyut=40x25, renk=256 renk, 320x200 piksel, 1 sayfa.

```
org 0100h
mov al, 13h      ; 40x25, 256 renk, 320x200 piksel
mov ah, 0
int 10h

mov dx, offset kitapadi      ; int 21h, ah=09h fonksiyonu
mov ah, 9                    ; $ okuyana kadar ekrana yazar
int 21h
ret
kitapadi db "Ah Istanbul $" ; buffer boyutu 10
hlt
```



## Ekran Fonksiyonları

### Ekran ayarları - imleç (cursor) yeri

- INT 10H / AH=02h alt fonksiyonuyla cursor pozisyonu girilir.
- INT 10H / AH=03h alt fonksiyonuyla cursor pozisyonu okunur.
- Her ikisinde de DH=satır, DL=sütun ve BH=sayfa sayısı.

```
org 0100h

mov ah, 3          ; cursor pozisyonu al (dh,dl)
int 10h
add dh, 10          ; satir
add dl, 10          ; sutun
mov ah, 2
int 10h
mov dx, offset kitapadi      ; int 21h, ah=09h fonksiyonu
mov ah, 9                    ; $ okuyana kadar ekrana yazar
int 21h

mov ah, 3          ; cursor pozisyonu al (dh,dl)
int 10h
add dh, 3           ; satir
add dl, 3           ; sutun
mov ah, 2
int 10h
mov dx, offset kitapadi      ; int 21h, ah=09h fonksiyonu
mov ah, 9                    ; $ okuyana kadar ekrana yazar
int 21h

ret
kitapadi db "Ah Istanbul $" ; buffer boyutu 10
hlt
```

## Ekran Fonksiyonları

### Ekran ayarları - imleç renk özelliği

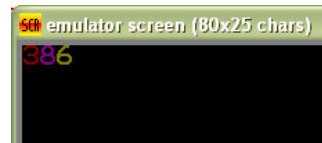
- INT 10H / AH=09h alt fonksiyonuyla cursor özelliği girilir.
- AL=karakter, BL=renk, CX=yazma adedi ve BH=sayfa sayısı.

```
org 0100h
mov dl,0      ; en sol pozisyon
mov cx,1      ; karakter yazım tekrar sayısı
mov bh,0

mov ah,09h    ; int 10h, ah=09h, bl ile belirtilen
mov al,'3'    ; renkte al'yi ekrana yazar
mov bl,4
int 10h
mov ah,2      ; cursor'u saga kaydirmek için
inc dl        ; dl sutun sayısı bir artıriliyor
int 10h

mov ah,09h
mov al,'8'
mov bl,5
int 10h
mov ah,2
inc dl
int 10h

mov ah,09h
mov bl,6
mov al,'6'
int 10h
hlt
```



## Mouse Kullanımı

### Mouse başlatma

- INT 33H / AX=0000h alt fonksiyonuyla mouse başlatılır.
- Başarılı olursa AX=0FFFFH olur ve BX buton sayısını gösterir.
- Başarısız olursa AX=0 gösterir.

```
org 0100h
mov ax,0      ; int 33h, ax=0 mouse baslatir
int 33h      ; mouse baslatildi
hlt
```



## Mouse Kullanımı

### Mouse pointer gösterme

- INT 33H / AX=0001h alt fonksiyonuyla mouse pointer görüntülenir.

```
org 0100h
mov ax, 1    ; int 33h, ax=1 mouse pointer'i görüntüler
int 33h
hlt
```

### Mouse pointer gizleme

- INT 33H / AX=0002h alt fonksiyonuyla mouse pointer gizlenir.

```
org 0100h
mov ax, 2    ; int 33h, ax=1 mouse pointer'i gizler
int 33h
hlt
```

## Mouse Kullanımı

### Mouse butonları ve pozisyonu algılama

- INT 33H / AX=0003h alt fonksiyonuyla mouse butonları ve pozisyonu bilgilerine ulaşılır.
- Sol butona basıldıysa BX=1, sağ butona basıldıysa BX=2, ikisine birden basıldıysa BX=3 olur.
- CX register'ı x pozisyonunu, DX register'ı y pozisyonunu gösterir.
- Scrool butonu için BX=4 olur.
- Scrool butonuyla sol ve sağ butona basıldığında sırasıyla BX=5 ve BX=6 olur.

```
org 0100h
mov ax, 1    ; int 33h, ax=1 mouse pointer'i gizler
int 33h
dongu:
mov ax, 3    ; int 33h, ax=3 buton ve pozisyon algılama
int 33h
jmp dongu
hlt
```



## Ödev

- Hafızada bulunan 5000 byte (karakter) blok içerisinde dışarıdan girilen bir string'i (boyutu değişebilir - enter'a basana kadar) arayan program yazınız.
- String bulunursa ilk karakterinin sıra numarasını ekrana yazacak, bulunamazsa kullanıcıya mesaj verecektir.
- Arama işlemi için SCAS komutu kullanılmayacaktır. Karakter bazında kontrollerle arama işlemi yapılacaktır.
- Boyer-Moore arama algoritması kullanılacaktır.
- Program emu8086 ile yazılacaktır.
- Ödevin çıktısı teslim edilecek ve kaynak kodu [akcayol@gazi.edu.tr](mailto:akcayol@gazi.edu.tr) adresine e-posta ile gönderilecektir.