



Bölüm 5: Kesmeler

Mikroişlemciler



Kesmeler (Interrupts)

- Programlamayı kolaylaştıran bir dizi komutu ifade eder.
- Bir karakteri yazdırmak için kod yazmak yerine bir kesme çağrılabilir.
- Disk sürücüsü ve diğer donanımlarla çalışan kesme işlevleri de vardır.
- Yazılım kesmeleri, belirli görevleri yerine getiren, kesme hizmet yordamları
 - ISR (*interrupt service routine*).
 - Karakter yazdırma, bellek yönetimi gibi işlevler.
- Donanım tarafından tetiklenen kesmeler de vardır.



Yazılım Kesmeleri (Software Interrupts)

- Yazılım kesmesi çağırarak için INT komutu kullanılır.
 - INT değeri
 - değeri, 0'dan 255'e (0'dan 0FFh'ye) kadar bir sayıdır.
- Kesme, alt fonksiyonlara sahip olabilir.
- Kesme çağrılmadan, AH yazmacı ayarlanarak alt fonksiyonu belirtilir.
- Her bir kesmenin 256 adet alt fonksiyonu olabilir ($256 * 256 = 65536$).
- Diğer yazmaçlar, alt fonksiyonlara parametre iletmek için kullanılabilir.



Ekran Yazısı (Screen Text) Yazma

```
ORG    100h
MOV    AH, 0Eh    ; alt fonksiyon seç.
MOV    AL, 'H'    ; ASCII kodu: 72
INT    10h        ; yazdır!
MOV    AL, 'e'    ; ASCII kodu: 101
INT    10h        ; yazdır!
MOV    AL, 'l'    ; ASCII kodu: 108
INT    10h        ; yazdır!
MOV    AL, 'l'    ; ASCII kodu: 108
INT    10h        ; yazdır!
RET    ; işletim sistemine geri dön.
```



Ekran Yazısı (Screen Text) Yazma

- INT 10h,
 - ekrana karakter yazdırmak ve
 - imleci hareket ettirmek için kullanılır.
- 0Eh alt fonksiyonu,
 - AL yazmacındaki ASCII kodunu ekrana yazdırır.
- Her bir karakter için,
 - AL yazmacına ASCII kodu atanır.
 - INT 10h çağrısı ile karakter ekrana yazdırılır.



Klavye Tuşuna Basılmasını Bekleme

```
ORG 100h
```

```
; wait for any key press:
```

```
MOV AH, 0
```

```
INT 16H
```

```
RET
```



Toplama Çıkarma ve Sonucu Yazdırma

org 100h

mov al, 5 ; bin=00000101b

mov bl, 10 ; hex=0ah or bin=00001010b

; 5 + 10 = 15 (decimal), 0fh, 00001111b

add bl, al

; 15 - 1 = 14 (decimal), 0eh, 00001110b

sub bl, 1



Toplama Çıkarma ve Sonucu Yazdırma

```
; sonucu ikilik tabanda yazdır  
mov cx, 8  
print: mov ah, 2 ; yazdırma fonksiyonu  
       mov dl, '0'  
       test bl, 10000000b ; ilk biti test et  
       jz zero  
       mov dl, '1'  
zero:  int 21h  
       shl bl, 1  
loop print
```




Toplama Çıkarma ve Sonucu Yazdırma

; b (binary suffix) yazdır

```
mov dl, 'b'
```

```
int 21h
```

; klavye tuşu basılmasını bekle

```
mov ah, 0
```

```
int 16h
```

```
ret
```



Toplama Çıkarma ve Sonucu Yazdırma

emulator: add-sub.com_

file math debug view external virtual devices virtual drive help

Load reload step back single step run step delay ms: 0

registers

	H	L
AX	00	00
BX	00	00
CX	00	26
DX	00	00
CS	07 00	
IP	01 00	
SS	07 00	
SP	FF FE	
BP	00 00	
SI	00 00	
DI	00 00	
DS	07 00	
ES	07 00	

0700:0100

```
07100: B0 176
07101: 05 005
07102: B3 179
07103: 0A 010 NEWL
07104: 02 002
07105: D8 216
07106: 80 128
07107: EB 235
07108: 01 001
07109: B9 185
0710A: 08 008 BACK
0710B: 00 000 NULL
0710C: B4 180
0710D: 02 002
0710E: B2 178
0710F: 30 048
07110: F6 246
07111: C3 195
07112: 80 128
07113: 74 116
07114: 02 002
07115: B2 178
```

0700:0100

```
MOV AL, 05h
MOV BL, 0Ah
ADD BL, AL
SUB BL, 01h
MOV CX, 00008h
MOV AH, 02h
MOV DL, 030h
TEST BL, 080h
JZ 0117h
MOV DL, 031h
INT 021h
SHL BL, 1
LOOP 010Ch
MOV DL, 062h
INT 021h
MOV AH, 00h
INT 016h
RET
NOP
NOP
NOP
...
```

screen source reset aux vars debug stack flags



Toplama Çıkarma ve Sonucu Yazdırma

emulator: add-sub.com_

file math debug view external virtual devices virtual drive help

Load reload step back single step run step delay ms: 0

registers

	H	L
AX	00	05
BX	00	00
CX	00	26
DX	00	00
CS	07 00	
IP	01 02	
SS	07 00	
SP	FF FE	
BP	00 00	
SI	00 00	
DI	00 00	
DS	07 00	
ES	07 00	

07 00: 01 02

07100:	B0	176	///
07101:	05	005	▲
07102:	B3	179	
07103:	0A	010	NEWL
07104:	02	002	0
07105:	D8	216	±
07106:	80	128	Ç
07107:	EB	235	δ
07108:	01	001	0
07109:	B9	185	
0710A:	08	008	BACK
0710B:	00	000	NULL
0710C:	B4	180	
0710D:	02	002	0
0710E:	B2	178	///
0710F:	30	048	0
07110:	F6	246	÷
07111:	C3	195	
07112:	80	128	Ç
07113:	74	116	t
07114:	02	002	0
07115:	B2	178	///

07 00: 01 02

```
MOU AL, 05h
MOU BL, 0Ah
ADD BL, AL
SUB BL, 01h
MOU CX, 00008h
MOU AH, 02h
MOU DL, 030h
TEST BL, 080h
JZ 0117h
MOU DL, 031h
INT 021h
SHL BL, 1
LOOP 010Ch
MOU DL, 062h
INT 021h
MOU AH, 00h
INT 016h
RET
NOP
NOP
NOP
...
```

screen source reset aux vars debug stack flags



Toplama Çıkarma ve Sonucu Yazdırma

emulator: add-sub.com_

file math debug view external virtual devices virtual drive help

Load reload step back single step run step delay ms: 0

registers

	H	L
AX	00	05
BX	00	0A
CX	00	26
DX	00	00
CS	07 00	
IP	01 04	
SS	07 00	
SP	FF FE	
BP	00 00	
SI	00 00	
DI	00 00	
DS	07 00	
ES	07 00	

07 00: 01 04

```
07100: B0 176  |
07101: 05 005  |
07102: B3 179  |
07103: 0A 010  NEWL
07104: 02 002  |
07105: D8 216  |
07106: 80 128  |
07107: EB 235  |
07108: 01 001  |
07109: B9 185  |
0710A: 08 008  BACK
0710B: 00 000  NULL
0710C: B4 180  |
0710D: 02 002  |
0710E: B2 178  |
0710F: 30 048  |
07110: F6 246  |
07111: C3 195  |
07112: 80 128  |
07113: 74 116  |
07114: 02 002  |
07115: B2 178  |
```

07 00: 01 04

```
MOV AL, 05h
MOV BL, 0Ah
ADD BL, AL
SUB BL, 01h
MOV CX, 00008h
MOV AH, 02h
MOV DL, 030h
TEST BL, 080h
JZ 0117h
MOV DL, 031h
INT 021h
SHL BL, 1
LOOP 010Ch
MOV DL, 062h
INT 021h
MOV AH, 00h
INT 016h
RET
NOP
NOP
NOP
...
```

screen source reset aux vars debug stack flags



Toplama Çıkarma ve Sonucu Yazdırma

emulator: add-sub.com_

file math debug view external virtual devices virtual drive help

Load reload step back single step run step delay ms: 0

registers

	H	L
AX	00	05
BX	00	0F
CX	00	26
DX	00	00
CS	07 00	
IP	01 06	
SS	07 00	
SP	FF FE	
BP	00 00	
SI	00 00	
DI	00 00	
DS	07 00	
ES	07 00	

07 00: 01 06

```
07100: B0 176 █
07101: 05 005 ↑
07102: B3 179 |
07103: 0A 010 NEWL
07104: 02 002 @
07105: D8 216 †
07106: 80 128 C
07107: EB 235 6
07108: 01 001 @
07109: B9 185 |
0710A: 08 008 BACK
0710B: 00 000 NULL
0710C: B4 180 |
0710D: 02 002 @
0710E: B2 178 █
0710F: 30 048 0
07110: F6 246 ÷
07111: C3 195 †
07112: 80 128 C
07113: 74 116 t
07114: 02 002 @
07115: B2 178 █
```

07 00: 01 06

```
MOV AL, 05h
MOV BL, 0Ah
ADD BL, AL
SUB BL, 01h
MOV CX, 00008h
MOV AH, 02h
MOV DL, 030h
TEST BL, 080h
JZ 0117h
MOV DL, 031h
INT 021h
SHL BL, 1
LOOP 010Ch
MOV DL, 062h
INT 021h
MOV AH, 00h
INT 016h
RET
NOP
NOP
NOP
...
```

screen source reset aux vars debug stack flags



Toplama Çıkarma ve Sonucu Yazdırma

emulator: add-sub.com_

file math debug view external virtual devices virtual drive help

Load reload step back single step run step delay ms: 0

registers

	H	L
AX	02	05
BX	00	0E
CX	00	08
DX	00	30
CS	07 00	
IP	01 10	
SS	07 00	
SP	FF FE	
BP	00 00	
SI	00 00	
DI	00 00	
DS	07 00	
ES	07 00	

0700:0110

07100:	B0	176	///
07101:	05	005	↑
07102:	B3	179	
07103:	0A	010	NEWL
07104:	02	002	⊖
07105:	D8	216	÷
07106:	80	128	Ç
07107:	EB	235	δ
07108:	01	001	⊖
07109:	B9	185	
0710A:	08	008	BACK
0710B:	00	000	NULL
0710C:	B4	180	
0710D:	02	002	⊖
0710E:	B2	178	///
0710F:	30	048	0
07110:	F6	246	÷
07111:	C3	195	
07112:	80	128	Ç
07113:	74	116	t
07114:	02	002	⊖
07115:	B2	178	///

0700:0110

```
MOV AL, 05h
MOV BL, 0Ah
ADD BL, AL
SUB BL, 01h
MOV CX, 00008h
MOV AH, 02h
MOV DL, 030h
TEST BL, 080h
JZ 0117h
MOV DL, 031h
INT 021h
SHL BL, 1
LOOP 010Ch
MOV DL, 062h
INT 021h
MOV AH, 00h
INT 016h
RET
NOP
NOP
NOP
```

screen source reset aux vars debug stack flags



Toplama Çıkarma ve Sonucu Yazdırma

emulator: add-sub.com_

file math debug view external virtual devices virtual drive help

Load reload step back single step run step delay ms: 0

registers

	H	L
AX	02	05
BX	00	0E
CX	00	08
DX	00	30
CS	07 00	
IP	01 17	
SS	07 00	
SP	FF FE	
BP	00 00	
SI	00 00	
DI	00 00	
DS	07 00	
ES	07 00	

0700:0117

```
07117: CD 205 =
07118: 21 033 !
07119: D0 208 u
0711A: E3 227 r
0711B: E2 226 r
0711C: EF 239 n
0711D: B2 178 s
0711E: 62 098 b
0711F: CD 205 =
07120: 21 033 !
07121: B4 180 j
07122: 00 000 NULL
07123: CD 205 =
07124: 16 022 -
07125: C3 195 j
07126: 90 144 e
07127: 90 144 e
07128: 90 144 e
07129: 90 144 e
0712A: 90 144 e
0712B: 90 144 e
0712C: 90 144 e
```

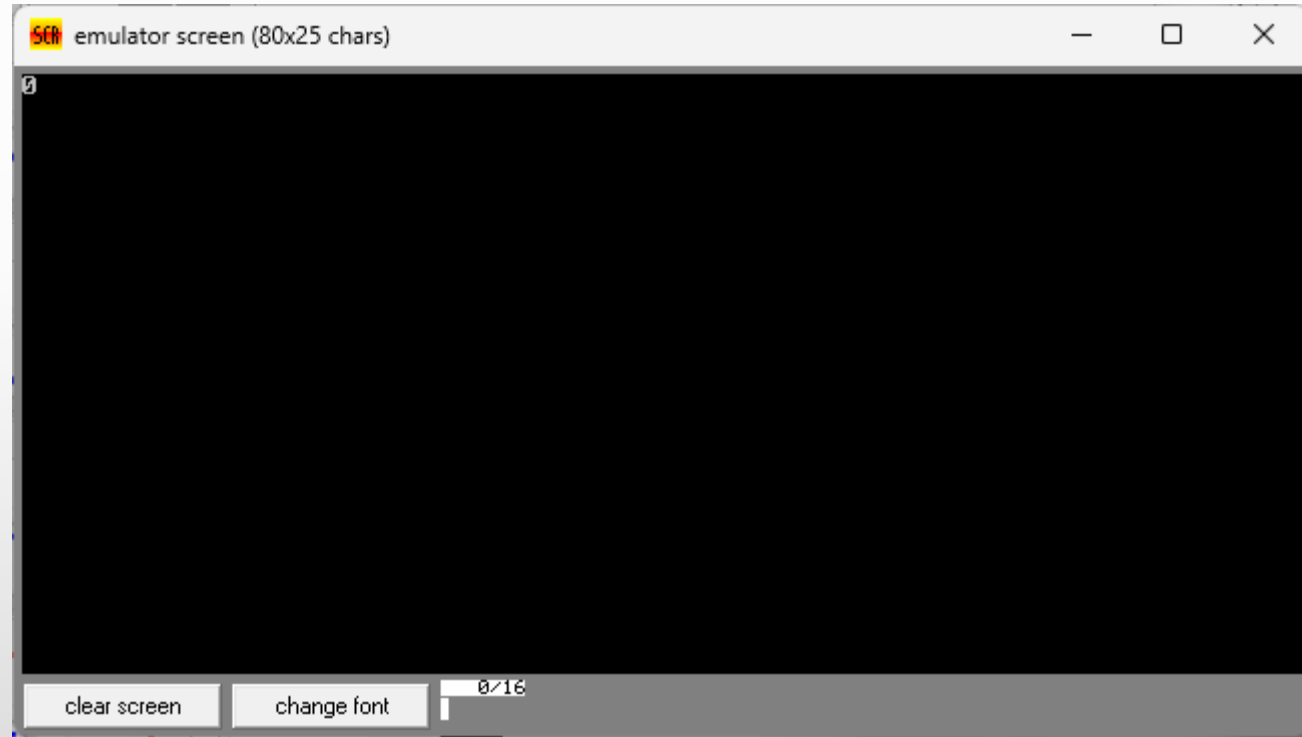
0700:0113

```
MOV AL, 05h
MOV BL, 0Ah
ADD BL, AL
SUB BL, 01h
MOV CX, 00008h
MOV AH, 02h
MOV DL, 030h
TEST BL, 080h
JZ 0117h
MOV DL, 031h
INT 021h
SHL BL, 1
LOOP 010Ch
MOV DL, 062h
INT 021h
MOV AH, 00h
INT 016h
RET
NOP
NOP
NOP
...
```

screen source reset aux vars debug stack flags



Toplama Çıkarma ve Sonucu Yazdırma





Toplama Çıkarma ve Sonucu Yazdırma

```
scr emulator screen (80x25 chars)
00001110
```

clear screen change font 0/16





SON