



# Bölüm 4: Değişkenler

## Mikroişlemciler



# Değişkenler

- Bir değişken, bir bellek konumunu temsil eder.
- Programcı için, *var1* değişkeninin
  - *5A73:235B* adresi yerine kullanılması çok daha kolaydır.
- Derleyici iki tür değişkeni destekler.
  - BYTE: Bir bayt (8 bit) değerini temsil eder.
    - name DB value (*Define Byte*).
  - WORD: Bir sözcük (16 bit) değerini temsil eder.
    - name DW value (*Define Word*).
- counter DB 10h (counter adında bir BYTE değişkeni).
- sum DW 0FFFFh (sum adında bir WORD değişkeni).



# Değişkenler

- Değişken adları,
  - harfle başlamalıdır,
  - sonrasında harf veya rakam kombinasyonları içerebilir.
- İsim belirtilmeyen değişkenler,
  - isim belirtilmemiş olabilir ancak bir adresleri vardır.
- Herhangi bir sayısal değer (ikili, onlu, onaltılı) alabilir.
- İlk değer atanmamış değişkenler için "?" sembolü kullanılır.



# Örnek Kod

```
ORG 100h
MOV AL, var1 ; AL, var1 deęişkeninin deęerini alır.
MOV BX, var2 ; BX, var2 deęişkeninin deęerini alır.
RET ; Programı sonlandırır.
VAR1 DB 7 ; BYTE türünde var1 deęişkeni, deęeri 7.
var2 DW 1234h ; WORD türünde var2 deęişkeni, deęeri 1234h.
```



# Örnek Kod

memory (1K) at: 0B56 : 0100 Disassemble from: 0B56 : 0100

0100: A0 160 á	MOV AL, [00108h]
0101: 08 008	MOV BX, [00109h]
0102: 01 001 ☹	RET
0103: 8B 139 i	POP ES
0104: 1E 030 ▲	XOR AL, 012h ?
0105: 09 009	ADD [BX + SI], AL
0106: 01 001 ☹	ADD [BX + SI], AL
0107: C3 195 †	ADD [BX + SI], AL
0108: 07 007 •	ADD [BX + SI], AL
0109: 34 052 4	ADD [BX + SI], AL
010A: 12 018 †	ADD [BX + SI], AL
010B: 00 000	ADD [BX + SI], AL
010C: 00 000	ADD [BX + SI], AL
010D: 00 000	ADD [BX + SI], AL
010E: 00 000	ADD [BX + SI], AL
010F: 00 000	ADD [BX + SI], AL
0110: 00 000	ADD [BX + SI], AL
0111: 00 000	ADD [BX + SI], AL
0112: 00 000	ADD [BX + SI], AL

variables



# Örnek Kod

- Derleyici,
  - Kaynak kodu bir dizi bayta dönüştürür.
  - Değişken adlarını bellek konumlarıyla değiştirir.
  - Büyük-küçük harfe duyarsızdır ("VAR1" ve "var1" aynı değişken).
- COM dosyaları yüklendiğinde DS, CS yazmacının değerini alır.
- Bağlı konum (*offset*), bellek konumunun başlangıcından olan kaydırmadır.
- VAR1'in bağlı konumu 0108h, tam adresi 0B56:0108'dir.
- var2'nin bağlı konumu 0109h, tam adresi 0B56:0109'dur.
  - WORD türünde olduğu için 2 BYTE kaplar.
  - Düşük bayt düşük adreste saklanır, 34h, 12h'den önce yer alır.



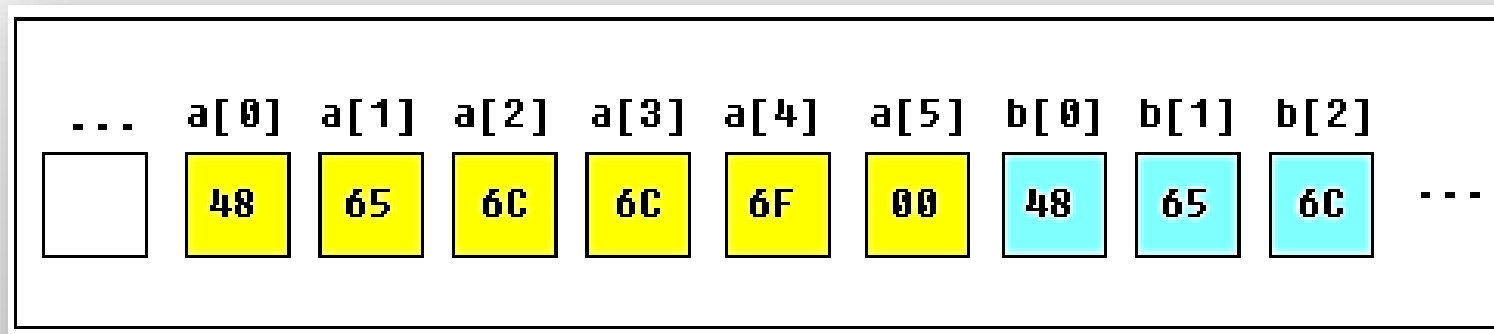
# ORG 100h Direktifi

- Derleyiciye yürütülebilir dosyanın yükleneceği adresi söyler.
  - Değişken adları bellek adresleriyle değiştirilirken dikkate alınır.
- COM dosyalarında kullanılır.
- İşletim sistemi, CS'nin ilk 256 baytında programla ilgili bazı verileri tutar.
  - Komut satırı parametreleri gibi bilgiler bu alanda saklanır.
- EXE dosyaları 0000 ofsetinde yüklenir ve
  - Değişkenler için özel bir kesim (*segment*) kullanır.
- Direktifler, gerçek makine koduna dönüştürülmez.



# Diziler

- Diziler, değişken zincirleri olarak düşünülebilir.
- Diziler, ardışık bellek konumlarında saklanan veri gruplarıdır.
- Tırnak içindeki metin otomatik olarak bayt dizisine dönüştürür.
- Dizi elemanlarına bellek adresi üzerinden erişilebilir.
- a DB 48h, 65h, 6Ch, 6Ch, 6Fh, 00h
- b DB 'Hello', 0.







# Diziler

- Dizinin herhangi bir elemanına,
  - Köşeli parantez kullanarak erişilebilir.
    - Örneğin: `MOV AL, a[3]`
    - Bellek indis yazmaçları `BX`, `SI`, `DI`, `BP` kullanarak erişilebilir.
      - Örneğin: `MOV SI, 3` ve `MOV AL, a[SI]`
  - Büyük bir dizi tanımlamak için `DUP` işleci kullanılır.
    - `number DUP ( value(s) )`
    - `c DB 5 DUP(9) → c DB 9, 9, 9, 9, 9`
    - `d DB 5 DUP(1, 2) → d DB 1, 2, 1, 2, 1, 2, 1, 2, 1, 2`
  - `DW` dizi tanımlamak için kullanılamaz.!



# Değişkenin Adresini Alma

- LEA ve OFFSET, değişkenin adresini almak için kullanılır.
- LEA, indisli değişkenlerin adresini almak için de kullanılabilir.
- Değişkenin adresi, prosedüre parametre geçirmek için çok kullanışlıdır.



# Değişkenin Adresini Alma

```
ORG 100h
MOV     AL, VAR1           ; VAR1 değerini AL yazmacına kopyala.
LEA     BX, VAR1          ; VAR1 adresini BX yazmacına kopyala.
MOV     BYTE PTR [BX], 44h ; VAR1 değerini güncelle.
MOV     AL, VAR1          ; VAR1 değerini AL yazmacına kopyala.
RET
VAR1    DB    22h
END
```



# Değişkenin Adresini Alma

```
ORG 100h
MOV     AL, VAR1           ; VAR1 değerini AL yazmacına kopyala.
MOV     BX, OFFSET VAR1   ; VAR1 adresini BX yazmacına kopyala.
MOV     BYTE PTR [BX], 44h ; VAR1 değerini güncelle.
MOV     AL, VAR1           ; VAR1 değerini AL yazmacına kopyala.
RET
VAR1    DB    22h
END
```



# Değişkenin Adresini Alma

- LEA BX, VAR1 ve MOV BX, OFFSET VAR1 aynı makine koduna derlenir:
  - MOV BX, num
  - num, değişkenin bağıl konum değerinin 16 bitlik temsili.
- Sadece BX, SI, DI, BP yazmaçları köşeli parantez içinde kullanılabilir.
- LEA (Load Effective Address), değişkenin adresini yazmaca yükler.
- OFFSET, değişkenin bağıl konum değerini sağlar.



# Sabitler (Constants)

- Değişkenlere benzer.
- Ancak program derlendiğinde var olurlar.
- Bir sabitin değeri tanımlandıktan sonra değiştirilemez.
- Sabitler, programın farklı bölümlerinde aynı değer kullanılmasını sağlar.
- Sabit tanımlamak için EQU direktifi kullanılır:
  - name EQU <herhangi bir ifade>.
  - k EQU 5
  - MOV AX, k



# Değişkenler Arasında Veri Taşıma

```
jmp start
```

```
value1 DW 10 ; İki byte'lık bir değişken tanımlama
```

```
value2 DW 0 ; İkinci bir değişken tanımlama
```

```
start:
```

```
MOV AX, value1 ; value1 değişkeninin değerini AX yazmacına taşı
```

```
MOV value2, AX ; AX yazmacındaki değeri value2 değişkenine taşı
```

```
end start
```



# Değişkenler Arasında Veri Taşıma

emulator: noname.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	0C
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		0700:0100	
07100:	EB 235 6	JMP 0106h	
07101:	04 004	OR AL, [BX + SI]	
07102:	0A 010 NEWL	ADD [BX + SI], AL	
07103:	00 000 NULL	MOV AX, [00102h]	
07104:	00 000 NULL	MOV [00104h], AX	
07105:	00 000 NULL	NOP	
07106:	A1 161 í	NOP	
07107:	02 002 0	NOP	
07108:	01 001 0	NOP	
07109:	A3 163 ú	NOP	
0710A:	04 004	NOP	
0710B:	01 001 0	NOP	
0710C:	90 144 é	NOP	
0710D:	90 144 é	NOP	
0710E:	90 144 é	NOP	
0710F:	90 144 é	NOP	
07110:	90 144 é	NOP	
07111:	90 144 é	NOP	
07112:	90 144 é	NOP	
07113:	90 144 é	NOP	
07114:	90 144 é	NOP	
07115:	90 144 é	...	

screen source reset aux vars debug stack flags





# Değişkenler Arasında Veri Taşıma

emulator: noname.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	0C
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:	EB	235	6
07101:	04	004	◆
07102:	0A	010	NEWL
07103:	00	000	NULL
07104:	00	000	NULL
07105:	00	000	NULL
07106:	A1	161	i
07107:	02	002	0
07108:	01	001	0
07109:	A3	163	ú
0710A:	04	004	◆
0710B:	01	001	0
0710C:	90	144	É
0710D:	90	144	É
0710E:	90	144	É
0710F:	90	144	É
07110:	90	144	É
07111:	90	144	É
07112:	90	144	É
07113:	90	144	É
07114:	90	144	É
07115:	90	144	É

07 00: 01 06

```
MOU AX, [00102h]
MOU [00104h], AX
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...
```

screen source reset aux vars debug stack flags



# Değişkenler Arasında Veri Taşıma

emulator: noname.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	0A
BX	00	00
CX	00	0C
DX	00	00
CS	07 00	
IP	01 09	
SS	07 00	
SP	FF FE	
BP	00 00	
SI	00 00	
DI	00 00	
DS	07 00	
ES	07 00	

07 00: 01 09

07100:	EB	235	δ
07101:	04	004	◆
07102:	0A	010	NEWL
07103:	00	000	NULL
07104:	00	000	NULL
07105:	00	000	NULL
07106:	A1	161	í
07107:	02	002	⊖
07108:	01	001	⊖
07109:	A3	163	ú
0710A:	04	004	◆
0710B:	01	001	⊖
0710C:	90	144	É
0710D:	90	144	É
0710E:	90	144	É
0710F:	90	144	É
07110:	90	144	É
07111:	90	144	É
07112:	90	144	É
07113:	90	144	É
07114:	90	144	É
07115:	90	144	É

07 00: 01 09

```
MOU AX, [00102h]
MOU [00104h], AX
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...
```

screen source reset aux vars debug stack flags



# Değişkenler Arasında Veri Taşıma

emulator: noname.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	0A
BX	00	00
CX	00	0C
DX	00	00
CS	07 00	
IP	01 0C	
SS	07 00	
SP	FF FE	
BP	00 00	
SI	00 00	
DI	00 00	
DS	07 00	
ES	07 00	

07 00: 01 0C

07100:	EB	235	δ
07101:	04	004	◆
07102:	0A	010	NEWL
07103:	00	000	NULL
07104:	0A	010	NEWL
07105:	00	000	NULL
07106:	A1	161	í
07107:	02	002	⊖
07108:	01	001	⊖
07109:	A3	163	ú
0710A:	04	004	◆
0710B:	01	001	⊖
0710C:	90	144	É
0710D:	90	144	É
0710E:	90	144	É
0710F:	90	144	É
07110:	90	144	É
07111:	90	144	É
07112:	90	144	É
07113:	90	144	É
07114:	90	144	É
07115:	90	144	É

07 00: 01 0C

```
MOV AX, [00102h]
MOV [00104h], AX
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...
```

screen source reset aux vars debug stack flags



# Değişkenleri Toplama İşlemi

jmp start

value1 DW 105h ; İki byte'lık değişken tanımla, başlangıç değeri 105h

value2 DW 103h ; İkinci bir değişken tanımla

result DW ? ; Sonucu saklayacak değişken, başlangıç değeri belirsiz (?)

start:

MOV AX, value1 ; value1 değişkeninin değerini AX yazmacına yükle

ADD AX, value2 ; value2 değişkeninin değerini AX yazmacına ekle

MOV result, AX ; AX yazmacındaki değeri result değişkenine taşı

end start





# Değişkenleri Toplama İşlemi

emulator: noname.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	12
DX	00	00
CS	07 00	
IP	01 08	
SS	07 00	
SP	FF FE	
BP	00 00	
SI	00 00	
DI	00 00	
DS	07 00	
ES	07 00	

07 00: 01 08

07100:	EB	235	δ
07101:	06	006	♣
07102:	05	005	♣
07103:	01	001	⊙
07104:	03	003	♥
07105:	01	001	⊙
07106:	00	000	NULL
07107:	00	000	NULL
07108:	A1	161	i
07109:	02	002	⊙
0710A:	01	001	⊙
0710B:	03	003	♥
0710C:	06	006	♣
0710D:	04	004	♦
0710E:	01	001	⊙
0710F:	A3	163	ú
07110:	06	006	♣
07111:	01	001	⊙
07112:	90	144	É
07113:	90	144	É
07114:	90	144	É
07115:	90	144	É

07 00: 01 08

```
MOU AX, [00102h]
ADD AX, [00104h]
MOU [00106h], AX
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...
```

screen source reset aux vars debug stack flags



# Değişkenleri Toplama İşlemi

emulator: noname.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	01	05
BX	00	00
CX	00	12
DX	00	00
CS	07 00	
IP	01 0B	
SS	07 00	
SP	FF FE	
BP	00 00	
SI	00 00	
DI	00 00	
DS	07 00	
ES	07 00	

07 00: 01 0B

07100:	EB	235	δ
07101:	06	006	♣
07102:	05	005	♣
07103:	01	001	⊙
07104:	03	003	♥
07105:	01	001	⊙
07106:	00	000	NULL
07107:	00	000	NULL
07108:	A1	161	í
07109:	02	002	⊙
0710A:	01	001	⊙
0710B:	03	003	♥
0710C:	06	006	♣
0710D:	04	004	♦
0710E:	01	001	⊙
0710F:	A3	163	ú
07110:	06	006	♣
07111:	01	001	⊙
07112:	90	144	É
07113:	90	144	É
07114:	90	144	É
07115:	90	144	É

07 00: 01 0B

```
MOU AX, [00102h]
ADD AX, [00104h]
MOU [00106h], AX
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...
```

screen source reset aux vars debug stack flags



# Değişkenleri Toplama İşlemi

emulator: noname.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	08
BX	00	00
CX	00	12
DX	00	00
CS	07 00	
IP	01 0F	
SS	07 00	
SP	FF FE	
BP	00 00	
SI	00 00	
DI	00 00	
DS	07 00	
ES	07 00	

07 00: 01 0F

07100:	EB	235	δ
07101:	06	006	♣
07102:	05	005	♣
07103:	01	001	⊙
07104:	03	003	♥
07105:	01	001	⊙
07106:	00	000	NULL
07107:	00	000	NULL
07108:	A1	161	í
07109:	02	002	⊙
0710A:	01	001	⊙
0710B:	03	003	♥
0710C:	06	006	♣
0710D:	04	004	♦
0710E:	01	001	⊙
0710F:	A3	163	ú
07110:	06	006	♣
07111:	01	001	⊙
07112:	90	144	É
07113:	90	144	É
07114:	90	144	É
07115:	90	144	É

07 00: 01 0F

```
MOV AX, [00102h]
ADD AX, [00104h]
MOV [00106h], AX
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...
```

screen source reset aux vars debug stack flags





# Değişkenleri Toplama İşlemi

emulator: noname.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	08
BX	00	00
CX	00	12
DX	00	00
CS	07 00	
IP	01 12	
SS	07 00	
SP	FF FE	
BP	00 00	
SI	00 00	
DI	00 00	
DS	07 00	
ES	07 00	

07 00: 01 12

```
07100: EB 235 6
07101: 06 006
07102: 05 005
07103: 01 001
07104: 03 003
07105: 01 001
07106: 08 008 BACK
07107: 02 002
07108: A1 161 i
07109: 02 002
0710A: 01 001
0710B: 03 003
0710C: 06 006
0710D: 04 004
0710E: 01 001
0710F: A3 163 u
07110: 06 006
07111: 01 001
07112: 90 144 E
07113: 90 144 E
07114: 90 144 E
07115: 90 144 E
```

07 00: 01 12

```
MOV AX, [00102h]
ADD AX, [00104h]
MOV [00106h], AX
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...
```

screen source reset aux vars debug stack flags



# Değişkenleri Toplama İşlemi

Random Access Memory

0700:0100 update  table  list

0700:0100	EB	06	05	01	03	01	08	02-A1	02	01	03	06	04	01	A3	δ↑↑↑↑↑.0í00↑↑↑↑
0700:0110	06	01	90	90	90	90	90	90-90	90	90	90	90	90	90	90	↑↑↑↑↑↑↑↑↑↑↑↑↑↑
0700:0120	90	90	90	90	90	90	F4	00-00	00	00	00	00	00	00	00	↑↑↑↑↑↑↑. . . . .
0700:0130	00	00	00	00	00	00	00	00-00	00	00	00	00	00	00	00	. . . . .
0700:0140	00	00	00	00	00	00	00	00-00	00	00	00	00	00	00	00	. . . . .
0700:0150	00	00	00	00	00	00	00	00-00	00	00	00	00	00	00	00	. . . . .
0700:0160	00	00	00	00	00	00	00	00-00	00	00	00	00	00	00	00	. . . . .
0700:0170	00	00	00	00	00	00	00	00-00	00	00	00	00	00	00	00	. . . . .



# İki Değişkenin Değerini Karşılaştırma

START:

MOV AX, value1 ; Birinci değişkenin değerini AX yazmacına yükle

CMP AX, value2 ; AX yazmacındaki değeri ikinci değişken ile karşılaştır

JGE greater\_or\_equal ; value1, value2'den büyük veya eşitse atla

MOV AX, value2 ; İkinci değişkenin değerini AX yazmacına yükle

JMP store\_max ; Karşılaştırma sonrası büyük değeri saklayan kısma atla

greater\_or\_equal:

MOV AX, value1 ; Birinci değişkenin değerini AX yazmacına yükle

store\_max:

MOV max\_value, AX ; AX yazmacındaki değeri max\_value değişkenine taşı

value1 DW 10 ; Birinci değişken, başlangıç değeri 10

value2 DW 5 ; İkinci değişken, başlangıç değeri 5

max\_value DW ? ; Büyük değeri tutacak değişken, ilk değeri belirsiz (?)

END START



# İki Değişkenin Değerini Karşılaştırma

emulator: noname.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	1A
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: A1 161 í
07101: 14 020 ¶
07102: 01 001 ©
07103: 3B 059 ;
07104: 06 006 ♣
07105: 16 022 =
07106: 01 001 ©
07107: 7D 125 >
07108: 05 005 ♣
07109: A1 161 í
0710A: 16 022 =
0710B: 01 001 ©
0710C: EB 235 δ
0710D: 03 003 ♡
0710E: A1 161 í
0710F: 14 020 ¶
07110: 01 001 ©
07111: A3 163 ú
07112: 18 024 †
07113: 01 001 ©
07114: 0A 010 NEWL
07115: 00 000 NULL
```

0700:0100

```
MOU AX, [00114h]
CMP AX, [00116h]
JNL 010Eh
MOU AX, [00116h]
JMP 0111h
MOU AX, [00114h]
MOU [00118h], AX
OR AL, [BX + SI]
ADD AX, 00000h
ADD [BX + SI] + 09090h, I
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...
```

screen source reset aux vars debug stack flags



# İki Değişkenin Değerini Karşılaştırma

emulator: noname.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	0A
BX	00	00
CX	00	1A
DX	00	00
CS	07 00	
IP	01 03	
SS	07 00	
SP	FF FE	
BP	00 00	
SI	00 00	
DI	00 00	
DS	07 00	
ES	07 00	

07 00: 01 03

```
07100: A1 161 i
07101: 14 020 9
07102: 01 001 0
07103: 3B 059 ;
07104: 06 006 +
07105: 16 022 -
07106: 01 001 0
07107: 7D 125 >
07108: 05 005 *
07109: A1 161 i
0710A: 16 022 -
0710B: 01 001 0
0710C: EB 235 6
0710D: 03 003 v
0710E: A1 161 i
0710F: 14 020 9
07110: 01 001 0
07111: A3 163 u
07112: 18 024 +
07113: 01 001 0
07114: 0A 010 NEWL
07115: 00 000 NULL
```

07 00: 01 03

```
MOV AX, [00114h]
CMP AX, [00116h]
JNL 010Eh
MOV AX, [00116h]
JMP 0111h
MOV AX, [00114h]
MOV [00118h], AX
OR AL, [BX + SI]
ADD AX, 00000h
ADD [BX + SI] + 09090h, 1
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...
```

screen source reset aux vars debug stack flags



# İki Değişkenin Değerini Karşılaştırma

emulator: noname.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	0A
BX	00	00
CX	00	1A
DX	00	00
CS	07 00	
IP	01 07	
SS	07 00	
SP	FF FE	
BP	00 00	
SI	00 00	
DI	00 00	
DS	07 00	
ES	07 00	

07 00: 01 07

```
07100: A1 161 i
07101: 14 020 9
07102: 01 001 0
07103: 3B 059 ;
07104: 06 006 +
07105: 16 022 -
07106: 01 001 0
07107: 7D 125 >
07108: 05 005 +
07109: A1 161 i
0710A: 16 022 -
0710B: 01 001 0
0710C: EB 235 6
0710D: 03 003 v
0710E: A1 161 i
0710F: 14 020 9
07110: 01 001 0
07111: A3 163 u
07112: 18 024 +
07113: 01 001 0
07114: 0A 010 NEWL
07115: 00 000 NULL
```

07 00: 01 07

```
MOV AX, [00114h]
CMP AX, [00116h]
JNL 010Eh
MOV AX, [00116h]
JMP 0111h
MOV AX, [00114h]
MOV [00118h], AX
OR AL, [BX + SI]
ADD AX, 00000h
ADD [BX + SI] + 09090h, 1
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...
```

screen source reset aux vars debug stack flags



# İki Değişkenin Değerini Karşılaştırma

emulator: noname.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	0A
BX	00	00
CX	00	1A
DX	00	00
CS	07 00	
IP	01 0E	
SS	07 00	
SP	FF FE	
BP	00 00	
SI	00 00	
DI	00 00	
DS	07 00	
ES	07 00	

07 00: 01 0E

```
07100: A1 161 i
07101: 14 020 q
07102: 01 001 @
07103: 3B 059 ;
07104: 06 006 +
07105: 16 022 =
07106: 01 001 @
07107: 7D 125 >
07108: 05 005 +
07109: A1 161 i
0710A: 16 022 =
0710B: 01 001 @
0710C: EB 235 d
0710D: 03 003 v
0710E: A1 161 i
0710F: 14 020 q
07110: 01 001 @
07111: A3 163 u
07112: 18 024 ^
07113: 01 001 @
07114: 0A 010 NEWL
07115: 00 000 NULL
```

07 00: 01 0E

```
MOV AX, [00114h]
CMP AX, [00116h]
JNL 010Eh
MOV AX, [00116h]
JMP 0111h
MOV AX, [00114h]
MOV [00118h], AX
OR AL, [BX + SI]
ADD AX, 00000h
ADD [BX + SI] + 09090h, 1
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...
```

screen source reset aux vars debug stack flags



# İki Değişkenin Değerini Karşılaştırma

emulator: noname.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	0A
BX	00	00
CX	00	1A
DX	00	00
CS	07 00	
IP	01 11	
SS	07 00	
SP	FF FE	
BP	00 00	
SI	00 00	
DI	00 00	
DS	07 00	
ES	07 00	

0700:0111

```
07100: A1 161 í
07101: 14 020 ¶
07102: 01 001 ©
07103: 3B 059 ;
07104: 06 006 †
07105: 16 022 =
07106: 01 001 ©
07107: 7D 125 >
07108: 05 005 †
07109: A1 161 í
0710A: 16 022 =
0710B: 01 001 ©
0710C: EB 235 δ
0710D: 03 003 ♡
0710E: A1 161 í
0710F: 14 020 ¶
07110: 01 001 ©
07111: A3 163 ú
07112: 18 024 †
07113: 01 001 ©
07114: 0A 010 NEWL
07115: 00 000 NULL
```

0700:0111

```
MOV AX, [00114h]
CMP AX, [00116h]
JNL 010Eh
MOV AX, [00116h]
JMP 0111h
MOV AX, [00114h]
MOV [00118h], AX
OR AL, [BX + SI]
ADD AX, 00000h
ADD [BX + SI] + 09090h, 1
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...
```

screen source reset aux vars debug stack flags





# Double Word Değişken Tanımlama

```
jmp start
mydouble dd 12345678h ; double word değişken tanımlama
mywords  dw 5678h ; 2 ayrı word değişken tanımlama
          dw 1234h
mybytes  db 78h ; 4 ayrı byte değişken tanımlama
          db 56h
          db 34h
          db 12h
data dd 00010010001101000101011001111000b ; 32 bits
start: ; double word değişkeni dx:ax 'e yükle
      mov ax, data
      mov dx, [data+2]
```



# Double Word Değişken Tanımlama

emulator: double\_word.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	1A
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:	EB	235	6
07101:	10	016	▶
07102:	78	120	x
07103:	56	086	U
07104:	34	052	4
07105:	12	018	‡
07106:	78	120	x
07107:	56	086	U
07108:	34	052	4
07109:	12	018	‡
0710A:	78	120	x
0710B:	56	086	U
0710C:	34	052	4
0710D:	12	018	‡
0710E:	78	120	x
0710F:	56	086	U
07110:	34	052	4
07111:	12	018	‡
07112:	A1	161	i
07113:	0E	014	¶
07114:	01	001	⊙
07115:	8B	139	i

0700:0100

```
JMP 0112h
JS 015Ah
XOR AL, 012h
JS 015Eh
XOR AL, 012h
JS 0162h
XOR AL, 012h
JS 0166h
XOR AL, 012h
MOV AX, [0010Eh]
MOV DX, [00110h]
RET
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...
```

screen source reset aux vars debug stack flags



# Double Word Değişken Tanımlama

emulator: double\_word.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	1A
DX	00	00
CS	07 00	
IP	01 12	
SS	07 00	
SP	FF FE	
BP	00 00	
SI	00 00	
DI	00 00	
DS	07 00	
ES	07 00	

07 00: 01 12

Address	Hex	Dec	Op
07100:	EB	235	6
07101:	10	016	▶
07102:	78	120	x
07103:	56	086	U
07104:	34	052	4
07105:	12	018	‡
07106:	78	120	x
07107:	56	086	U
07108:	34	052	4
07109:	12	018	‡
0710A:	78	120	x
0710B:	56	086	U
0710C:	34	052	4
0710D:	12	018	‡
0710E:	78	120	x
0710F:	56	086	U
07110:	34	052	4
07111:	12	018	‡
07112:	A1	161	i
07113:	0E	014	µ
07114:	01	001	⊖
07115:	8B	139	i

MOU AX, [0010Eh]  
MOU DX, [00110h]  
RET  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
...

screen source reset aux vars debug stack flags



# Double Word Değişken Tanımlama

emulator: double\_word.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	56	78
BX	00	00
CX	00	1A
DX	00	00
CS	07 00	
IP	0115	
SS	07 00	
SP	FFFE	
BP	0000	
SI	0000	
DI	0000	
DS	07 00	
ES	07 00	

0700:0115

```
07100: EB 235 6
07101: 10 016 >
07102: 78 120 x
07103: 56 086 U
07104: 34 052 4
07105: 12 018 †
07106: 78 120 x
07107: 56 086 U
07108: 34 052 4
07109: 12 018 †
0710A: 78 120 x
0710B: 56 086 U
0710C: 34 052 4
0710D: 12 018 †
0710E: 78 120 x
0710F: 56 086 U
07110: 34 052 4
07111: 12 018 †
07112: A1 161 i
07113: 0E 014 j
07114: 01 001 @
07115: 8B 139 i
```

0700:0115

```
MOU AX, [0010Eh]
MOU DX, [00110h]
RET
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...
```

screen source reset aux vars debug stack flags



# Double Word Değişken Tanımlama

emulator: double\_word.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	56	78
BX	00	00
CX	00	1A
DX	12	34
CS	07 00	
IP	01 19	
SS	07 00	
SP	FF FE	
BP	00 00	
SI	00 00	
DI	00 00	
DS	07 00	
ES	07 00	

0700:0119

07119: C3 195
0711A: 90 144 É
0711B: 90 144 É
0711C: 90 144 É
0711D: 90 144 É
0711E: 90 144 É
0711F: 90 144 É
07120: 90 144 É
07121: 90 144 É
07122: 90 144 É
07123: 90 144 É
07124: 90 144 É
07125: 90 144 É
07126: 90 144 É
07127: 90 144 É
07128: 90 144 É
07129: 90 144 É
0712A: 90 144 É
0712B: 90 144 É
0712C: 90 144 É
0712D: 90 144 É
0712E: F4 244

0700:0115

```
MOV AX, [0010Eh]
MOV DX, [00110h]
RET
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...
```

screen source reset aux vars debug stack flags



SON