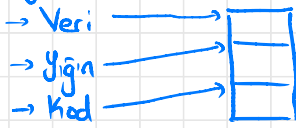


.EXE

3 ayrı kesim alanı tanımlanır.



FAR tipinde ana yardımcı olmalı

Program ilk çalıştığında
DS = OS' a döniş kesim değeri
SS = Stack (Yığın)
CS = Kod [END Etiket]

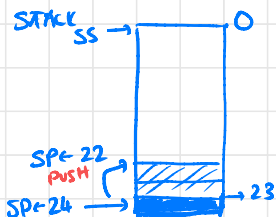
Baş belleğin tamamı kullanılabilir
(Kod olarak)

Yığın boyutu tanımlanan alan kadar

NEAR ve FAR tipinde alt yardımcı çağırabilir.

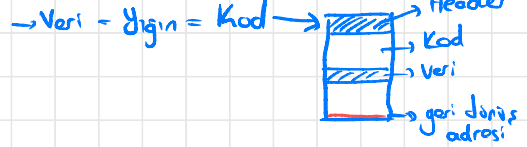
$IP \Rightarrow$ [END Etiket]
IP her zaman 0 değeri olmak zorunda değil

$SP \leftarrow$ Yığın boyutu kadar.
STK SEGMENT PARA 'ST'
DW 12 DUP(10)
STK ENDS



.COM

3 katışık kesim alanı mevcut



NEAR tipinde ana yardımcı olmalı.

DS = CS = SS

64 KB ile sınırlı
(64 KB - (Header + Döniş adresleri) : kod boyutu

64 KB - (Header + Veri + Kod) : yığın boyutu.

Sadece NEAR alt yardımcı çağırabilir.

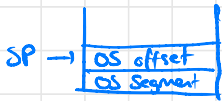
$IP \leftarrow$ 100H' ten başlamalı.
ORG 100H

$SP \leftarrow$ FFFEH
FFFFEH adresindeki word OS döniş adresi

Tanımlı değişkenlere erişim için
(DS ← tanımlı veri bölümünü göstermesi için)
(OS'a dönüş işlemleri burada yapılır)
MOV AX, veri_bölümü_ismi
MOV DS, AX

OS'a dönebilmek için
OS (DS: 0000H) dönüş bekler.

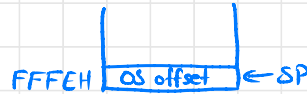
PUSH DS
XOR AX, AX ; AX ← 0
PUSH AX



RET

Tanımlı verilere erişim için özel bir işleme
ihtiyaç yoktur.

OS'a dönüş için bir işleme gerek yok.
OS, COM tipi programı çalıştırdığında
yığına (FFFFH adresine) geri dönüş off-
set değerini otomatik olarak yazar.
RET



Diskte kapladığı alan

toplam program boyutu + 512 Byte header

Diskte kapladığı alan toplam program
boyutu kadar.

= HIZLAMA TIPLERİ =

KETA

PUSH DS
XOR AX, AX
PUSH AX

• Dikkat kapladığı alan
toplam 1000 byte + 52 byte kadar

FILE

• Dikkat kapladığı alan
toplam 1000 byte + 52 byte kadar

Hızlama tipleri:

BYTE ← segment sayıdan 16 bit, adresin ikili olarak gösterilmesi.

WORD ← segment sayıdan 16 bit, 2 nin katı boyutlu adresin gösterilmesi.

PARA ← " " " 16'ın katı " "

PAGE ← " " " 256'nin katı " "

... 0000
... 0001

00000H → LSB
00001H → MSB

256

TAB

GA 85

GA 85

16 bitlik konular

16 bit

16B → 16 x 8

⇒ 512 x 16

TAB ⇒ TABS + GABS

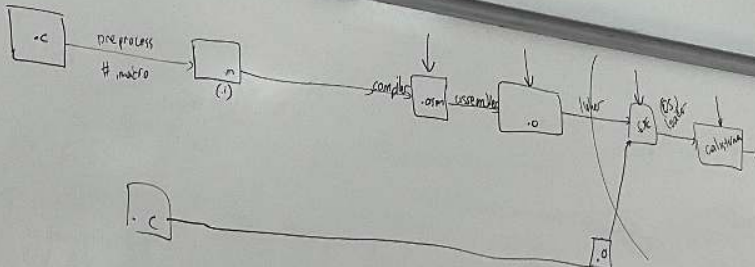
LSB ← 00001H ... 0001

MSB ← 00002H ... 0010

TAB ⇒ TABS + GABS

LSB ← 00001H ... 0001

MSB ← 00002H ... 0010



=DOSBOX =

mount c c:\dosbox => dosbox options
[autoexec] en altına yazılırsa her seferinde yazmana gerek yok.

c:

Link
masm } mount edilen dosyanın root'una konulacak.
debug

C:\> MASM.EXE TOPLA16.EXE } derleme
linker... > TOPLA16
_lst... > TOPLA16 } uzatılı dosyalar
_obj
_crf
_lst

C:\> LINK.EXE TOPLA16.OBJ > Linklene
map -> TOPLA16

C:\> TOPLA16.EXE => Execute.

C:\> DEBUG.EXE TOPLA16.EXE => Debug

- r : registers (registerlerin değerleri)

- r ax

- r F

- D CS:0 Kod segmentin machine kodunu dump eder
CS:0,25 ilk 25 byte yazdır

- A enter dedikten sonra assembly görebiliriz.

- q çıkış
- u cs:0 belirtilen segment: unassemble
- t programı 1 adım gösterme (trace)
- ? commands
- go verilen adrese kadar programı çalıştırır.

