

ALT SEVİYE PROGRAMLAMA

Hafta 5

Dr. Öğr. Üyesi Erkan USLU

Assembly Program Tipleri

- Assembly dilinde programlar
 - Kodlama şekli
 - Kesim tanımlamaları
 - Kesim düzenine

göre iki farklı şekilde yazılabilir.

- Derlenen programın uzantısına göre isimlendirme yapılır:
 - EXE
 - COM

EXE Tipi Programlar

EXE Tipindeki Programlar

```
STACKSG SEGMENT PARA STACK 'STACK'  
    DW 32 DUP(?)  
STACKSG ENDS
```

```
DATASG SEGMENT PARA 'DATA'  
SAYI DB ?  
ELEMEN DW ?  
DATASG ENDS
```

```
CODESG SEGMENT PARA 'CODE'  
    ASSUME CS:CODESG, DS:DATASG, SS:STACKSG
```

```
BASLA PROC FAR  
    PUSH DS  
    XOR AX,AX  
    PUSH AX
```

```
    MOV AX, DATASG  
    MOV DS, AX
```

```
    RETF  
BASLA ENDP  
CODESG ENDS  
    END BASLA
```

EXE Tipindeki Programlar

```
STACKSG SEGMENT PARA STACK 'STACK'  
    DW 32 DUP(?)  
STACKSG ENDS
```

Stack Segment



```
DATASG SEGMENT PARA 'DATA'  
SAYI DB ?  
ELEMEN DW ?  
DATASG ENDS
```

```
CODESG SEGMENT PARA 'CODE'  
    ASSUME CS:CODESG, DS:DATASG, SS:STACKSG
```

```
BASLA PROC FAR  
    PUSH DS  
    XOR AX,AX  
    PUSH AX
```

```
    MOV AX, DATASG  
    MOV DS, AX
```

```
    RETF  
BASLA ENDP  
CODESG ENDS  
END BASLA
```

EXE Tipindeki Programlar

```
STACKSG SEGMENT PARA STACK 'STACK'
```

```
    DW 32 DUP(?)
```

```
STACKSG ENDS
```

```
DATASG SEGMENT PARA 'DATA'
```

```
SAYI  DB ?
```

```
ELEMAN DW ?
```

```
DATASG ENDS
```

```
CODESG SEGMENT PARA 'CODE'
```

```
    ASSUME CS:CODESG, DS:DATASG, SS:STACKSG
```

```
BASLA  PROC FAR
```

```
    PUSH DS
```

```
    XOR AX,AX
```

```
    PUSH AX
```

```
    MOV AX, DATASG
```

```
    MOV DS, AX
```

```
    RETF
```

```
BASLA  ENDP
```

```
CODESG ENDS
```

```
    END BASLA
```

Stack Segment
Adı

EXE Tipindeki Programlar

```
STACKSG SEGMENT PARA STACK 'STACK'  
    DW 32 DUP(?)  
STACKSG ENDS
```

```
DATASG SEGMENT PARA 'DATA'  
SAYI DB ?  
ELEMEN DW ?  
DATASG ENDS
```

```
CODESG SEGMENT PARA 'CODE'  
    ASSUME CS:CODESG, DS:DATASG, SS:STACKSG
```

```
BASLA PROC FAR  
    PUSH DS  
    XOR AX,AX  
    PUSH AX
```

```
    MOV AX, DATASG  
    MOV DS, AX
```

```
    RETF  
BASLA ENDP  
CODESG ENDS  
    END BASLA
```

Segmentin Stack
Segment
olduğunu belirler

EXE Tipindeki Programlar

```
STACKSG SEGMENT PARA STACK 'STACK'
```

```
    DW 32 DUP(?)
```

```
STACKSG ENDS
```

```
DATASG SEGMENT PARA 'DATA'
```

```
    SAYI DB ?
```

```
    ELEMAN DW ?
```

```
DATASG ENDS
```

```
CODESG SEGMENT PARA 'CODE'
```

```
    ASSUME CS:CODESG, DS:DATASG, SS:STACKSG
```

```
BASLA PROC FAR
```

```
    PUSH DS
```

```
    XOR AX,AX
```

```
    PUSH AX
```

```
    MOV AX, DATASG
```

```
    MOV DS, AX
```

```
    RETF
```

```
BASLA ENDP
```

```
CODESG ENDS
```

```
END BASLA
```

DW kullanılmalı
veya çift sayı ile
DB kullanılmalı

EXE Tipindeki Programlar

```
STACKSG SEGMENT PARA STACK 'STACK'  
    DW 32 DUP(?)
```

```
STACKSG ENDS
```

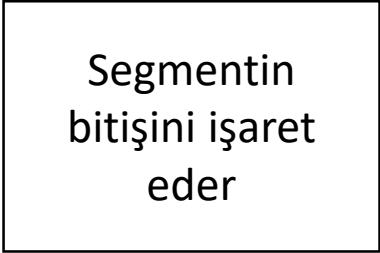
```
DATASG SEGMENT PARA 'DATA'  
SAYI DB ?  
ELEMEN DW ?  
DATASG ENDS
```

```
CODESG SEGMENT PARA 'CODE'  
    ASSUME CS:CODESG, DS:DATASG, SS:STACKSG
```

```
BASLA PROC FAR  
    PUSH DS  
    XOR AX,AX  
    PUSH AX
```

```
    MOV AX, DATASG  
    MOV DS, AX
```

```
    RETF  
BASLA ENDP  
CODESG ENDS  
    END BASLA
```



Segmentin
bitişini işaret
eder

EXE Tipindeki Programlar

```
STACKSG SEGMENT PARA STACK 'STACK'  
    DW 32 DUP(?)  
STACKSG ENDS
```

```
DATASG SEGMENT PARA 'DATA'  
SAYI DB ?  
ELEMEN DW ?  
DATASG ENDS
```

```
CODESG SEGMENT PARA 'CODE'  
    ASSUME CS:CODESG, DS:DATASG, SS:STACKSG
```

```
BASLA PROC FAR  
    PUSH DS  
    XOR AX,AX  
    PUSH AX
```

```
    MOV AX, DATASG  
    MOV DS, AX
```

```
    RETF  
BASLA ENDP  
CODESG ENDS  
    END BASLA
```

EXE Tipindeki Programlar

```
STACKSG SEGMENT PARA STACK 'STACK'  
    DW 32 DUP(?)  
STACKSG ENDS
```

```
DATASG SEGMENT PARA 'DATA'  
SAYI DB ?  
ELEMEN DW ?  
DATASG ENDS
```

Data Segment



```
CODESG SEGMENT PARA 'CODE'  
    ASSUME CS:CODESG, DS:DATASG, SS:STACKSG
```

```
BASLA PROC FAR  
    PUSH DS  
    XOR AX,AX  
    PUSH AX
```

```
    MOV AX, DATASG  
    MOV DS, AX
```

```
    RETF  
BASLA ENDP  
CODESG ENDS  
END BASLA
```

EXE Tipindeki Programlar

```
STACKSG SEGMENT PARA STACK 'STACK'  
    DW 32 DUP(?)  
STACKSG ENDS
```

```
DATASG SEGMENT PARA 'DATA'  
SAYI DB ?  
ELEMEN DW ?  
DATASG ENDS
```

```
CODESG SEGMENT PARA 'CODE'  
    ASSUME CS:CODESG, DS:DATASG, SS:STACKSG
```

```
BASLA PROC FAR  
    PUSH DS  
    XOR AX,AX  
    PUSH AX
```

```
    MOV AX, DATASG  
    MOV DS, AX
```

```
    RETF  
BASLA ENDP  
CODESG ENDS  
    END BASLA
```



Değişken tanımları

EXE Tipindeki Programlar

```
STACKSG SEGMENT PARA STACK 'STACK'  
    DW 32 DUP(?)  
STACKSG ENDS
```


```
DATASG SEGMENT PARA 'DATA'  
SAYI DB ?  
ELEMEN DW ?  
DATASG ENDS
```

```
CODESG SEGMENT PARA 'CODE'  
    ASSUME CS:CODESG, DS:DATASG, SS:STACKSG
```

```
BASLA PROC FAR  
    PUSH DS  
    XOR AX,AX  
    PUSH AX
```

```
    MOV AX, DATASG  
    MOV DS, AX
```

```
    RETF  
BASLA ENDP  
CODESG ENDS  
END BASLA
```



Data segment
ismi

EXE Tipindeki Programlar

```
STACKSG SEGMENT PARA STACK 'STACK'  
    DW 32 DUP(?)  
STACKSG ENDS
```

```
DATASG SEGMENT PARA 'DATA'  
SAYI DB ?  
ELEMEN DW ?  
DATASG ENDS
```

```
CODESG SEGMENT PARA 'CODE'  
    ASSUME CS:CODESG, DS:DATASG, SS:STACKSG
```

```
BASLA PROC FAR  
    PUSH DS  
    XOR AX,AX  
    PUSH AX
```

```
    MOV AX, DATASG  
    MOV DS, AX
```

```
    RETF  
BASLA ENDP  
CODESG ENDS  
    END BASLA
```

EXE Tipindeki Programlar

```
STACKSG SEGMENT PARA STACK 'STACK'  
    DW 32 DUP(?)  
STACKSG ENDS
```

```
DATASG SEGMENT PARA 'DATA'  
SAYI DB ?  
ELEMEN DW ?  
DATASG ENDS
```

```
CODESG SEGMENT PARA 'CODE'  
    ASSUME CS:CODESG, DS:DATASG, SS:STACKSG  
  
BASLA PROC FAR  
    PUSH DS  
    XOR AX,AX  
    PUSH AX  
  
    MOV AX, DATASG  
    MOV DS, AX  
  
    RETF  
BASLA ENDP  
CODESG ENDS  
END BASLA
```



Code Segment

EXE Tipindeki Programlar

```
STACKSG SEGMENT PARA STACK 'STACK'  
    DW 32 DUP(?)  
STACKSG ENDS
```

```
DATASG SEGMENT PARA 'DATA'  
SAYI DB ?  
ELEMEN DW ?  
DATASG ENDS
```

```
CODESG SEGMENT PARA 'CODE'  
    ASSUME CS:CODESG, DS:DATASG, SS:STACKSG
```

```
BASLA PROC FAR  
    PUSH DS  
    XOR AX,AX  
    PUSH AX  
  
    MOV AX, DATASG  
    MOV DS, AX  
  
    RETF  
BASLA ENDP  
CODESG ENDS  
END BASLA
```

EXE programda
en az 1 tane FAR
tipinde prosedür
olmalı

EXE Tipindeki Programlar

```
STACKSG SEGMENT PARA STACK 'STACK'  
    DW 32 DUP(?)  
STACKSG ENDS
```

```
DATASG SEGMENT PARA 'DATA'  
SAYI DB ?  
ELEMEN DW ?  
DATASG ENDS
```

```
CODESG SEGMENT PARA 'CODE'  
    ASSUME CS:CODESG, DS:DATASG, SS:STACKSG
```

```
BASLA PROC FAR
```

```
PUSH DS  
XOR AX,AX  
PUSH AX
```

```
MOV AX, DATASG  
MOV DS, AX
```

```
RETF
```

```
BASLA ENDP  
CODESG ENDS  
END BASLA
```

EXE'yi çağıran
kodun kesim
değeri DS'de,
ofset değeri 0,
dönüş için bu
değerler stack'e
atılır

EXE Tipindeki Programlar

```
STACKSG SEGMENT PARA STACK 'STACK'  
    DW 32 DUP(?)  
STACKSG ENDS
```

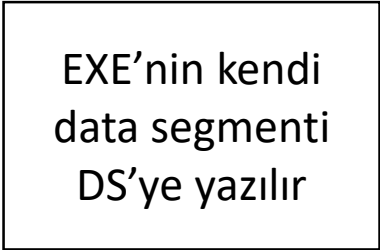
```
DATASG SEGMENT PARA 'DATA'  
SAYI DB ?  
ELEMEN DW ?  
DATASG ENDS
```

```
CODESG SEGMENT PARA 'CODE'  
    ASSUME CS:CODESG, DS:DATASG, SS:STACKSG
```

```
BASLA PROC FAR  
    PUSH DS  
    XOR AX,AX  
    PUSH AX
```

```
    MOV AX, DATASG  
    MOV DS, AX
```

```
    RETF  
BASLA ENDP  
CODESG ENDS  
    END BASLA
```



EXE'nin kendi
data segmenti
DS'ye yazılır

EXE Tipindeki Programlar

```
STACKSG SEGMENT PARA STACK 'STACK'  
    DW 32 DUP(?)  
STACKSG ENDS
```

```
DATASG SEGMENT PARA 'DATA'  
SAYI DB ?  
ELEMEN DW ?  
DATASG ENDS
```

```
CODESG SEGMENT PARA 'CODE'  
    ASSUME CS:CODESG, DS:DATASG, SS:STACKSG
```

```
BASLA PROC FAR  
    PUSH DS  
    XOR AX,AX  
    PUSH AX
```

```
    MOV AX, DATASG  
    MOV DS, AX
```

```
    .....
```

```
    RETF
```

```
BASLA ENDP  
CODESG ENDS  
    END BASLA
```

Assembly
kodunu
yazacağımız blok

EXE Tipindeki Programlar

```
STACKSG SEGMENT PARA STACK 'STACK'  
    DW 32 DUP(?)  
STACKSG ENDS
```

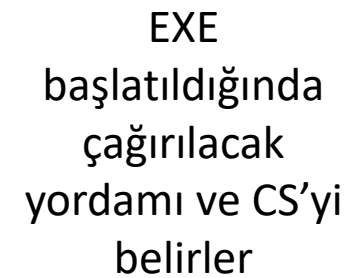
```
DATASG SEGMENT PARA 'DATA'  
SAYI DB ?  
ELEMEN DW ?  
DATASG ENDS
```

```
CODESG SEGMENT PARA 'CODE'  
    ASSUME CS:CODESG, DS:DATASG, SS:STACKSG
```

```
BASLA PROC FAR  
    PUSH DS  
    XOR AX,AX  
    PUSH AX
```

```
    MOV AX, DATASG  
    MOV DS, AX
```

```
    RETF  
BASLA ENDP  
CODESG ENDS  
END BASLA
```



EXE
başlatıldığında
çağırılacak
yordamı ve CS'yi
belirler

The diagram shows a box on the right containing text that explains the function of the 'BASLA' label. A line connects the 'END BASLA' label in the assembly code on the left to the box. The text in the box states that the EXE file, when executed, will call the procedure named 'BASLA' and determine the CS register.

EXE Tipindeki Programlar

```
STACKSG SEGMENT PARA STACK 'STACK'  
    DW 32 DUP(?)  
STACKSG ENDS
```

```
DATASG SEGMENT PARA 'DATA'  
SAYI DB ?  
ELEMEN DW ?  
DATASG ENDS
```

```
CODESG SEGMENT PARA 'CODE'  
    ASSUME CS:CODESG, DS:DATASG, SS:STACKSG
```

```
BASLA PROC FAR  
    PUSH DS  
    XOR AX,AX  
    PUSH AX
```

```
    MOV AX, DATASG  
    MOV DS, AX
```

```
    RETF  
BASLA ENDP  
CODESG ENDS  
    END BASLA
```

PAGE 60,80

TITLE ornek_____

STEK SEGMENT PARA STACK 'STK'

DW 30 DUP(?)

STEK ENDS

DSG SEGMENT PARA 'DTS'

SAYI DB ?

ELEMAN DW 10

DIZI DW 12 DUP(0)

DIZI2 DD 10 DUP(?)

DSG ENDS

CSG SEGMENT PARA 'CODE'

ASSUME CS:CSG, DS:DSG, SS:STEK

BASLA PROC FAR

PUSH DS

XOR AX,AX

PUSH AX

MOV AX, DSG

MOV DS, AX

RETF

BASLA ENDP

CSG ENDS

END BASLA

COM Tipi Programlar

COM Tipi Programlar

```
CODESG SEGMENT PARA 'CODE'
```

```
    ORG 100H
```

```
    ASSUME CS:CODESG, SS:CODESG, DS:CODESG, ES:CODESG
```

```
BASLA  PROC NEAR
```

```
    RET
```

```
BASLA  ENDP
```

```
SAYIB  DB ?
```

```
SAYIW  DW ?
```

```
CODESG ENDS
```

```
    END BASLA
```


COM Tipi Programlar

```
CODESG SEGMENT PARA 'CODE'  
    ORG 100H  
    ASSUME CS:CODESG, SS:CODESG, DS:CODESG, ES:CODESG  
  
BASLA PROC NEAR  
  
    RET  
BASLA ENDP  
  
SAYIB DB ?  
SAYIW DW ?  
CODESG ENDS  
END BASLA
```

COM programlar için tek bir segment tanımı yapılır; data, stack ve code segment olarak aynı segment kullanılır

COM Tipi Programlar

```
CODESG SEGMENT PARA 'CODE'
```

```
  ORG 100H
```

```
  ASSUME CS:CODESG, SS:CODESG, DS:CODESG, ES:CODESG
```

```
BASLA PROC NEAR
```

```
  RET
```

```
BASLA ENDP
```

```
SAYIB DB ?
```

```
SAYIW DW ?
```

```
CODESG ENDS
```

```
  END BASLA
```

COM programların ilk 256 byte'lık kısımları header bilgisi içerir. Assembly komutlarının ilk 100H'tan başlayabilmesi için ORG 100H kullanılır

COM Tipi Programlar

```
CODESG SEGMENT PARA 'CODE'
```

```
ORG 100H
```

```
ASSUME CS:CODESG, SS:CODESG, DS:CODESG, ES:CODESG
```

```
BASLA PROC NEAR
```

```
RET
```

```
BASLA ENDP
```

```
SAYIB DB ?
```

```
SAYIW DW ?
```

```
CODESG ENDS
```

```
END BASLA
```

COM program
tek segment
kullandığı için en
az 1 tane near
prosedür
içermesi gerekir

COM Tipi Programlar

```
CODESG SEGMENT PARA 'CODE'
```

```
    ORG 100H
```

```
    ASSUME CS:CODESG, SS:CODESG, DS:CODESG, ES:CODESG
```

```
BASLA PROC NEAR
```

```
    RET
```

```
BASLA ENDP
```

```
SAYIB DB ?
```

```
SAYIW DW ?
```

```
CODESG ENDS
```

```
    END BASLA
```



Değişken
tanımları

COM Tipi Programlar

```
CODESG SEGMENT PARA 'CODE'
```

```
    ORG 100H
```

```
    ASSUME CS:CODESG, SS:CODESG, DS:CODESG, ES:CODESG
```

```
BASLA PROC NEAR
```

```
    RET
```

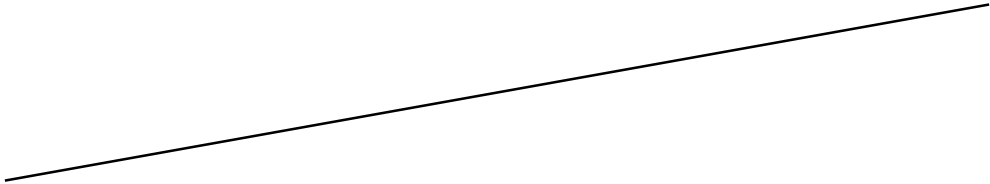
```
BASLA ENDP
```

```
SAYIB DB ?
```

```
SAYIW DW ?
```

```
CODESG ENDS
```

```
END BASLA
```



COM program
çalıştırıldığında
çalıştırılacak
yordamı belirtir

COM Tipi Programlar

```
CODESG SEGMENT PARA 'CODE'
```

```
    ORG 100H
```

```
    ASSUME CS:CODESG, SS:CODESG, DS:CODESG, ES:CODESG
```

```
BASLA PROC NEAR
```

```
    ...
```

```
    RET
```

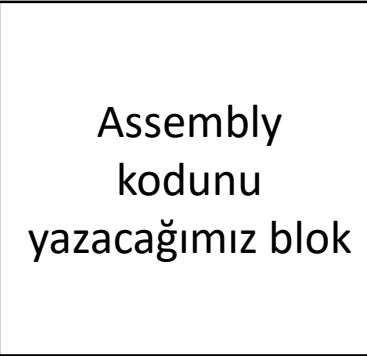
```
BASLA ENDP
```

```
SAYIB DB ?
```

```
SAYIW DW ?
```

```
CODESG ENDS
```

```
    END BASLA
```



Assembly
kodunu
yazacağımız blok

COM Tipi Programlar

```
CODESG SEGMENT PARA 'CODE'
```

```
    ORG 100H
```

```
    ASSUME CS:CODESG, SS:CODESG, DS:CODESG, ES:CODESG
```

```
BASLA PROC NEAR
```

```
    RET
```

```
BASLA ENDP
```

```
SAYIB DB ?
```

```
SAYIW DW ?
```

```
CODESG ENDS
```

```
    END BASLA
```

PAGE 60,80

TITLE ornek_____

CODESG SEGMENT PARA 'CODE'

ORG 100H

ASSUME CS:CODESG, SS:CODESG, DS:CODESG, ES:CODESG

BASLA PROC NEAR

RET

BASLA ENDP

SAYIB DB ?

SAYIW DW ?

CODESG ENDS

END BASLA

COM Şablon 1

PAGE 60,80

TITLE ornek_____

CODESG SEGMENT PARA 'CODE'

ORG 100H

ASSUME CS:CODESG, SS:CODESG, DS:CODESG, ES:CODESG

BILGI: JMP BASLA

SAYIB DB ?

SAYIW DW ?

BASLA PROC NEAR

RET

BASLA ENDP

CODESG ENDS

END BILGI

COM Şablon 2

COM EXE Karşılaştırma

	COM Programı	EXE Programı
Kullanılabilen Bellek	64KB - Header – Dönüş adresi	Boş bellek
Başlangıçta CS	Header'ın bulunduğu kesim	END komutundan sonraki etiketin bulunduğu kesim
Başlangıçta IP	100H	END komutundan sonraki etiketin bulunduğu ofset
Başlangıçta DS	Header'ın bulunduğu kesim	Dönüş için gerekli kesim değeri
Değişkenlere Erişim İçin Yapılması Gerekenler	Yok	MOV AX, verikesimismi MOV DS, AX
Değişkenlerin Tanımlandığı Yer	ENDP'den sonra	Veri kesiminde

COM EXE Karşılaştırma

	COM Programı	EXE Programı
Başlangıçta SS	Header'ın bulunduğu kesim	Yığın kesimi adresi
Başlangıçta SP	OFFFEH	Yığın kesimi boyutu
Başlangıçta Yığın	0000H	Boş
Yığın Büyüklüğü	64KB – 256 (header) – X byte Kod – Y byte Değişken	Yığın kesimi boyutu
Ana Prosedür Tipi	NEAR	FAR
Dönüş İçin Yapılması Gereken	Yok	PUSH DS XOR AX, AX PUSH AX