



# Bölüm 8: Yığın

## Mikroişlemciler



# Yığın (Stack)

- Yığın, geçici verileri saklayan bellek alanıdır.
- CALL komutu tarafından kullanılır ve prosedürün dönüş adresini saklar.
- RET komutu, bu değeri yığından alır ve belirtilen adrese döner.
- INT komutu bir kesmeyi çağırdığında benzer bir işlem gerçekleşir;
  - Durum yazmacı, kod kesimi ve bağıl konum değeri yığına saklanır
  - IRET komutu, kesme çağrısından dönmek için kullanılır.
- PUSH: 16 bit değeri yığına koyar.
- POP: 16 bit değeri yığından alır.



# Örnek Kod Parçası

```
ORG    100h

MOV    AX, 42    ; AX register'ına 42 değerini ata.
PUSH   AX        ; AX değerini yığına koy.

POP    BX        ; Yığından değeri BX yazmacına al.

RET    ; İşletim sistemine dön.

END
```



# PUSH Komutu

- PUSH REG
  - PUSH SREG
  - PUSH memory
  - PUSH immediate
- 
- REG: AX, BX, CX, DX, DI, SI, BP, SP.
  - SREG: DS, ES, SS, CS.
  - memory: [BX], [BX+SI+7], 16 bit variable, gibi ..
  - immediate: 5, -24, 3Fh, 10001101b, gibi ..



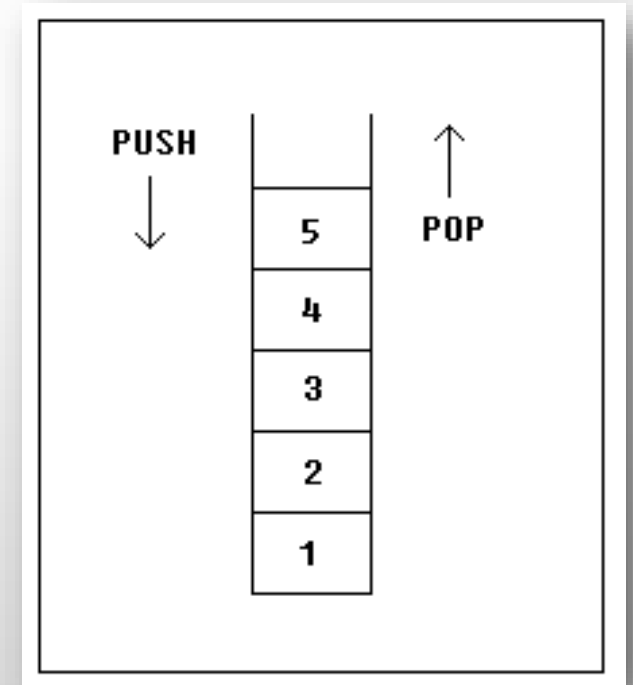
# POP Komutu

- POP REG
  - POP SREG
  - POP memory
- 
- REG: AX, BX, CX, DX, DI, SI, BP, SP.
  - SREG: DS, ES, SS, (CS hariç).
  - memory: [BX], [BX+SI+7], 16 bit variable, gibi ..



# LIFO Algoritması

- LIFO, "Last In First Out"un kısaltmasıdır.
- Yığına eklenen son öge, yığından çıkan ilk öge olur.
- Yığına koyulan değerler sırasıyla (1, 2, 3, 4, 5) olsun,
  - Yığından ilk alınan değer 5 olur,
  - Ardından sırasıyla 4, 3, 2 ve en son 1 alınır.





# Örnek Kod Parçası

```
ORG    100h
MOV    AX, 1          ; AX yazmacına 1 değerini ata.
PUSH   AX             ; 1 değerini yığına koy.
MOV    AX, 2          ; AX yazmacına 2 değerini ata.
PUSH   AX             ; 2 değerini yığına koy.
MOV    AX, 3          ; AX yazmacına 3 değerini ata.
PUSH   AX             ; 3 değerini yığına koy.
POP    BX             ; Yığından BX yazmacına al. (3)
POP    BX             ; Yığından BX yazmacına al. (2)
POP    BX             ; Yığından BX yazmacına al. (1)
RET    ; İşletim sistemine dön.
END
```



# Yığın Güvenliği

- PUSH ve POP komutları,
  - programın çalışma süresince geçici verileri saklar.
- Yığın bütünlüğü,
  - eşit sayıda PUSH ve POP işlemi ile korunmalıdır.
- Eğer eşit sayıda PUSH ve POP yapılmazsa,
  - yığın bozulabilir,
  - işletim sistemine geri dönüş yapılamayabilir.
- RET komutu, yığında bir dönüş adresi bekler (genellikle 0000h).





# Örnek Kod Parçası

```
ORG    100h
MOV    AX, 1234h
PUSH   AX           ; store value of AX in stack.
MOV    AX, 5678h   ; modify the AX value.
POP    AX           ; restore the original value of AX.
RET
END
```



# Örnek Kod Parçası

- `ORG 100h`
- `MOV AX, 1212h` ; store 1212h in AX.
- `MOV BX, 3434h` ; store 3434h in BX
- `PUSH AX` ; store value of AX in stack.
- `PUSH BX` ; store value of BX in stack.
- `POP AX` ; set AX to original value of BX.
- `POP BX` ; set BX to original value of AX.
- `RET`
- `END`





# Örnek Kod Parçası

emulator: micro-os\_kernel.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	12	12
BX	34	34
CX	00	0B
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:	B8	184	3
07101:	12	018	3
07102:	12	018	3
07103:	BB	187	7
07104:	34	052	4
07105:	34	052	4
07106:	50	080	P
07107:	53	083	S
07108:	58	088	X
07109:	5B	091	[
0710A:	C3	195	†
0710B:	90	144	E
0710C:	90	144	E
0710D:	90	144	E
0710E:	90	144	E
0710F:	90	144	E
07110:	90	144	E
07111:	90	144	E
07112:	90	144	E
07113:	90	144	E
07114:	90	144	E
07115:	90	144	E

07 00: 01 06

```
MOU AX, 01212h
MOU BX, 03434h
PUSH AX
PUSH BX
POP AX
POP BX
RET
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...
```

screen source reset aux vars debug stack flags



# Örnek Kod Parçası

emulator: micro-os\_kernel.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	12	12
BX	34	34
CX	00	0B
DX	00	00
CS	07 00	
IP	01 08	
SS	07 00	
SP	FF FA	
BP	00 00	
SI	00 00	
DI	00 00	
DS	07 00	
ES	07 00	

07 00: 01 08

07100:	B8	184	3
07101:	12	018	3
07102:	12	018	3
07103:	BB	187	7
07104:	34	052	4
07105:	34	052	4
07106:	50	080	P
07107:	53	083	S
07108:	58	088	X
07109:	5B	091	L
0710A:	C3	195	↓
0710B:	90	144	E
0710C:	90	144	E
0710D:	90	144	E
0710E:	90	144	E
0710F:	90	144	E
07110:	90	144	E
07111:	90	144	E
07112:	90	144	E
07113:	90	144	E
07114:	90	144	E
07115:	90	144	E

07 00: 01 08

```
MOU AX, 01212h
MOU BX, 03434h
PUSH AX
PUSH BX
POP AX
POP BX
RET
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...
```

screen source reset aux vars debug stack flags



# Örnek Kod Parçası

emulator: micro-os\_kernel.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	34	34
BX	34	34
CX	00	0B
DX	00	00
CS	07 00	
IP	01 09	
SS	07 00	
SP	FF FC	
BP	00 00	
SI	00 00	
DI	00 00	
DS	07 00	
ES	07 00	

07 00: 01 09

07100:	B8	184	3
07101:	12	018	4
07102:	12	018	4
07103:	BB	187	7
07104:	34	052	4
07105:	34	052	4
07106:	50	080	P
07107:	53	083	S
07108:	58	088	X
07109:	5B	091	L
0710A:	C3	195	T
0710B:	90	144	E
0710C:	90	144	E
0710D:	90	144	E
0710E:	90	144	E
0710F:	90	144	E
07110:	90	144	E
07111:	90	144	E
07112:	90	144	E
07113:	90	144	E
07114:	90	144	E
07115:	90	144	E

MOU AX, 01212h  
MOU BX, 03434h  
PUSH AX  
PUSH BX  
POP AX  
POP BX  
RET  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
...

screen source reset aux vars debug stack flags



# Örnek Kod Parçası

emulator: micro-os\_kernel.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	34	34
BX	12	12
CX	00	0B
DX	00	00
CS	07 00	
IP	01 0A	
SS	07 00	
SP	FF FE	
BP	00 00	
SI	00 00	
DI	00 00	
DS	07 00	
ES	07 00	

07 00: 01 0A

07100:	B8	184	↵
07101:	12	018	↵
07102:	12	018	↵
07103:	BB	187	↵
07104:	34	052	4
07105:	34	052	4
07106:	50	080	P
07107:	53	083	S
07108:	58	088	X
07109:	5B	091	[
0710A:	C3	195	
0710B:	90	144	É
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

MOU AX, 01212h  
MOU BX, 03434h  
PUSH AX  
PUSH BX  
POP AX  
POP BX  
RET  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
NOP  
...

screen source reset aux vars debug stack flags



# Yığın Belleği ve İşleyişi

- Yığın belleği,
  - SS (Yığın Kesimi) ve SP (Yığın İşaretçisi) yazmaçlarını kullanır.
- İşletim sistemi genellikle bu yazmaçların başlangıç değerlerini belirler.
- PUSH kaynak:
  - SP yazmacından 2 çıkarılır.
  - Kaynak değeri SS:SP adresine yazılır.
- POP hedef:
  - SS:SP adresindeki değer hedefe yazılır.
  - SP yazmacına 2 eklenir.





# Yığın Kullanımı

```
mov ax, 2
```

```
mov bx, 4
```

```
mov cx, 8
```

```
push ax
```

```
push bx
```

```
push cx
```

```
pop ax
```

```
pop bx
```

```
pop cx
```



# Yığın Kullanımı

emulator: micro-os\_kernel.bin\_

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	00
DX	00	00
CS	0100	
IP	0000	
SS	0100	
SP	FFFE	
BP	0000	
SI	0000	
DI	0000	
DS	0100	
ES	0100	

0100:0000 0100:0000

Address	Hex	Char	Instruction
01000:	B8 184	7	MOV AX, 00002h
01001:	02 002	0	MOV BX, 00004h
01002:	00 000	NULL	MOV CX, 00008h
01003:	BB 187	7	PUSH AX
01004:	04 004	◆	PUSH BX
01005:	00 000	NULL	PUSH CX
01006:	B9 185	5	POP AX
01007:	08 008	BACK	POP BX
01008:	00 000	NULL	POP CX
01009:	50 080	P	NOP
0100A:	53 083	S	NOP
0100B:	51 081	Q	NOP
0100C:	58 088	X	NOP
0100D:	5B 091	I	NOP
0100E:	59 089	Y	NOP
0100F:	90 144	É	NOP
01010:	90 144	É	NOP
01011:	90 144	É	NOP
01012:	90 144	É	NOP
01013:	90 144	É	NOP
01014:	90 144	É	NOP
01015:	90 144	É	...

screen source reset aux vars debug stack flags



# Yığın Kullanımı

emulator: micro-os\_kernel.bin\_

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	02
BX	00	00
CX	00	00
DX	00	00
CS	0100	
IP	0003	
SS	0100	
SP	FFFE	
BP	0000	
SI	0000	
DI	0000	
DS	0100	
ES	0100	

0100:0003

01000:	B8	184	ı
01001:	02	002	ı
01002:	00	000	NULL
01003:	BB	187	ı
01004:	04	004	ı
01005:	00	000	NULL
01006:	B9	185	ı
01007:	08	008	BACK
01008:	00	000	NULL
01009:	50	080	P
0100A:	53	083	S
0100B:	51	081	Q
0100C:	58	088	X
0100D:	5B	091	I
0100E:	59	089	Y
0100F:	90	144	É
01010:	90	144	É
01011:	90	144	É
01012:	90	144	É
01013:	90	144	É
01014:	90	144	É
01015:	90	144	É

0100:0003

```
MOU AX, 00002h
MOU BX, 00004h
MOU CX, 00008h
PUSH AX
PUSH BX
PUSH CX
POP AX
POP BX
POP CX
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...
```

screen source reset aux vars debug stack flags



# Yığın Kullanımı

emulator: micro-os\_kernel.bin\_

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	02
BX	00	04
CX	00	00
DX	00	00
CS	0100	
IP	0006	
SS	0100	
SP	FFFE	
BP	0000	
SI	0000	
DI	0000	
DS	0100	
ES	0100	

0100:0006

01000:	B8	184	ı
01001:	02	002	0
01002:	00	000	NULL
01003:	BB	187	ı
01004:	04	004	♦
01005:	00	000	NULL
01006:	B9	185	ı
01007:	08	008	BACK
01008:	00	000	NULL
01009:	50	080	P
0100A:	53	083	S
0100B:	51	081	Q
0100C:	58	088	X
0100D:	5B	091	I
0100E:	59	089	Y
0100F:	90	144	É
01010:	90	144	É
01011:	90	144	É
01012:	90	144	É
01013:	90	144	É
01014:	90	144	É
01015:	90	144	É

0100:0006

```
MOU AX, 00002h
MOU BX, 00004h
MOU CX, 00008h
PUSH AX
PUSH BX
PUSH CX
POP AX
POP BX
POP CX
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...
```

screen source reset aux vars debug stack flags



# Yığın Kullanımı

emulator: micro-os\_kernel.bin\_

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	02
BX	00	04
CX	00	08
DX	00	00
CS	0100	
IP	0009	
SS	0100	
SP	FFFE	
BP	0000	
SI	0000	
DI	0000	
DS	0100	
ES	0100	

0100:0009

01000:	B8	184	7
01001:	02	002	0
01002:	00	000	NULL
01003:	BB	187	7
01004:	04	004	♦
01005:	00	000	NULL
01006:	B9	185	7
01007:	08	008	BACK
01008:	00	000	NULL
01009:	50	080	P
0100A:	53	083	S
0100B:	51	081	Q
0100C:	58	088	X
0100D:	5B	091	I
0100E:	59	089	Y
0100F:	90	144	E
01010:	90	144	E
01011:	90	144	E
01012:	90	144	E
01013:	90	144	E
01014:	90	144	E
01015:	90	144	E

0100:0009

```
MOV AX, 00002h
MOV BX, 00004h
MOV CX, 00008h
PUSH AX
PUSH BX
PUSH CX
POP AX
POP BX
POP CX
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...
```

screen source reset aux vars debug stack flags



# Yığın Kullanımı

emulator: micro-os\_kernel.bin\_

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	02
BX	00	04
CX	00	08
DX	00	00
CS	0100	
IP	000A	
SS	0100	
SP	FFFC	
BP	0000	
SI	0000	
DI	0000	
DS	0100	
ES	0100	

0100:000A

```
01000: B8 184 7
01001: 02 002 0
01002: 00 000 NULL
01003: BB 187 7
01004: 04 004 4
01005: 00 000 NULL
01006: B9 185 7
01007: 08 008 BACK
01008: 00 000 NULL
01009: 50 080 P
0100A: 53 083 S
0100B: 51 081 Q
0100C: 58 088 X
0100D: 5B 091 I
0100E: 59 089 Y
0100F: 90 144 E
01010: 90 144 E
01011: 90 144 E
01012: 90 144 E
01013: 90 144 E
01014: 90 144 E
01015: 90 144 E
```

0100:000A

```
MOV AX, 00002h
MOV BX, 00004h
MOV CX, 00008h
PUSH AX
PUSH BX
PUSH CX
POP AX
POP BX
POP CX
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...
```

screen source reset aux vars debug stack flags



# Yığın Kullanımı

emulator: micro-os\_kernel.bin\_

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	02
BX	00	04
CX	00	08
DX	00	00
CS	0100	
IP	000B	
SS	0100	
SP	FFFA	
BP	0000	
SI	0000	
DI	0000	
DS	0100	
ES	0100	

0100:000B

01000:	B8	184	7
01001:	02	002	0
01002:	00	000	NULL
01003:	BB	187	7
01004:	04	004	♦
01005:	00	000	NULL
01006:	B9	185	7
01007:	08	008	BACK
01008:	00	000	NULL
01009:	50	080	P
0100A:	53	083	S
0100B:	51	081	Q
0100C:	58	088	X
0100D:	5B	091	I
0100E:	59	089	Y
0100F:	90	144	E
01010:	90	144	E
01011:	90	144	E
01012:	90	144	E
01013:	90	144	E
01014:	90	144	E
01015:	90	144	E

0100:000B

```
MOV AX, 00002h
MOV BX, 00004h
MOV CX, 00008h
PUSH AX
PUSH BX
PUSH CX
POP AX
POP BX
POP CX
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...
```

screen source reset aux vars debug stack flags



# Yığın Kullanımı

emulator: micro-os\_kernel.bin\_

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	02
BX	00	04
CX	00	08
DX	00	00
CS	0100	
IP	000C	
SS	0100	
SP	FFF8	
BP	0000	
SI	0000	
DI	0000	
DS	0100	
ES	0100	

0100:000C

```
01000: B8 184 7
01001: 02 002 8
01002: 00 000 NULL
01003: BB 187 7
01004: 04 004 4
01005: 00 000 NULL
01006: B9 185 7
01007: 08 008 BACK
01008: 00 000 NULL
01009: 50 080 P
0100A: 53 083 S
0100B: 51 081 Q
0100C: 58 088 X
0100D: 5B 091 L
0100E: 59 089 Y
0100F: 90 144 E
01010: 90 144 E
01011: 90 144 E
01012: 90 144 E
01013: 90 144 E
01014: 90 144 E
01015: 90 144 E
```

0100:000C

```
MOV AX, 00002h
MOV BX, 00004h
MOV CX, 00008h
PUSH AX
PUSH BX
PUSH CX
POP AX
POP BX
POP CX
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...
```

screen source reset aux vars debug stack flags





# Yığın Kullanımı

emulator: micro-os\_kernel.bin\_

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	08
BX	00	04
CX	00	08
DX	00	00
CS	0100	
IP	0000	
SS	0100	
SP	FFFA	
BP	0000	
SI	0000	
DI	0000	
DS	0100	
ES	0100	

0100:0000

01000:	B8	184	ı
01001:	02	002	0
01002:	00	000	NULL
01003:	BB	187	ı
01004:	04	004	♦
01005:	00	000	NULL
01006:	B9	185	ı
01007:	08	008	BACK
01008:	00	000	NULL
01009:	50	080	P
0100A:	53	083	S
0100B:	51	081	Q
0100C:	58	088	X
0100D:	5B	091	L
0100E:	59	089	V
0100F:	90	144	É
01010:	90	144	É
01011:	90	144	É
01012:	90	144	É
01013:	90	144	É
01014:	90	144	É
01015:	90	144	É

0100:0000

```
MOV AX, 00002h
MOV BX, 00004h
MOV CX, 00008h
PUSH AX
PUSH BX
PUSH CX
POP AX
POP BX
POP CX
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...
```

screen source reset aux vars debug stack flags



# Yığın Kullanımı

emulator: micro-os\_kernel.bin\_

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	08
BX	00	04
CX	00	08
DX	00	00
CS	0100	
IP	000E	
SS	0100	
SP	FFFC	
BP	0000	
SI	0000	
DI	0000	
DS	0100	
ES	0100	

0100:000E

01000:	B8	184	ı
01001:	02	002	0
01002:	00	000	NULL
01003:	BB	187	ı
01004:	04	004	♦
01005:	00	000	NULL
01006:	B9	185	ı
01007:	08	008	BACK
01008:	00	000	NULL
01009:	50	080	P
0100A:	53	083	S
0100B:	51	081	Q
0100C:	58	088	X
0100D:	5B	091	L
0100E:	59	089	Y
0100F:	90	144	É
01010:	90	144	É
01011:	90	144	É
01012:	90	144	É
01013:	90	144	É
01014:	90	144	É
01015:	90	144	É

0100:000E

```
MOV AX, 00002h
MOV BX, 00004h
MOV CX, 00008h
PUSH AX
PUSH BX
PUSH CX
POP AX
POP BX
POP CX
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...
```

screen source reset aux vars debug stack flags



# Yığın Kullanımı

emulator: micro-os\_kernel.bin\_

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	08
BX	00	04
CX	00	02
DX	00	00
CS	0100	
IP	000F	
SS	0100	
SP	FFFE	
BP	0000	
SI	0000	
DI	0000	
DS	0100	
ES	0100	

0100:000F

Address	Hex	Dec	Symbol
01000:	B8	184	ı
01001:	02	002	ö
01002:	00	000	NULL
01003:	BB	187	ı
01004:	04	004	♦
01005:	00	000	NULL
01006:	B9	185	ı
01007:	08	008	BACK
01008:	00	000	NULL
01009:	50	080	P
0100A:	53	083	S
0100B:	51	081	Q
0100C:	58	088	X
0100D:	5B	091	I
0100E:	59	089	Y
0100F:	90	144	É
01010:	90	144	É
01011:	90	144	É
01012:	90	144	É
01013:	90	144	É
01014:	90	144	É
01015:	90	144	É

0100:000F

```
MOV AX, 00002h
MOV BX, 00004h
MOV CX, 00008h
PUSH AX
PUSH BX
PUSH CX
POP AX
POP BX
POP CX
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
NOP
...
```

screen source reset aux vars debug stack flags



SON