

1)

a) Code Segment

Code segment registeri olan CS icra gören programı içeren segmentin başlangıç adresini gösterir. Diğer bir deyişle, CS ile adreslenen segment içinde makine dili komutları yer almaktadır. Stack segment registeri olan SS, geçerli stack segmentin başlangıç adresini içerir. Stack segmenti t,çeşitli verilerin geçici olarak saklandığı bir bellek bloğudur.

b) Data Segment

Data segment registeri olan DS ,geçerli data segmentinin başlangıç adresidir. Bu segment genel olarak program değişkenleri tarafından kullanılır. Diğer bir deyişle, bilgi işlem operasyonlarına tabi tutulan bilgiler, data segment içindeki lokasyonlarda yer alırlar.

c) Extra Segment

Ekstra segment registeri olan ES geçerli ekstra segmentin başlangıç adresini içerir. Ekstra segment, genel olarak string operasyonları için kullanılır. Yeterli bellek büyüklüğüne sahip olan sistemlerde bu 4 segment, belleğin farklı parçalarıdır. Geniş belleğe sahip olmayan sistemlerde ise, bu segmentler üstüste gelebilir.

d) Stack Segment

Stack segment registeri olan SS, geçerli stack segmentin başlangıç adresini içerir. Stack segmenti t,çeşitli verilerin geçici olarak saklandığı bir bellek bloğudur.

e) IP

Çeşitli bellek bölgelerindeki verilere ulaşabilmek için adres göstericisi olarak kullanılır. Bellek içindeki bir lokasyonun adresini gösterebilmek için, iki ayrı adres değeri kullanılır. İlk değer, işaret edilen lokasyonu içinde barındıran segmentin adresidir . bu noktada segment registerlerinden yararlanılır. Lokasyonun segment içindeki yerini gösteren ikinci adres değeri ise offset adresi olarak bilinir. Offset adreslerinin gösterilmesi için de, pointer ve indeks registerlerinden yararlanılır.

2)

- **R (Register)** : Bu komut o andaki register değerlerini ve bayrakların durumunu belirlemekte kullanılır.

Ör:

```
-R
AX=0000 BX=0000 CX=0000 DX=0000 SP=FFEE BP=0000 SI=0000 DI=0000
DS=2AF2 ES=2AF2 SS=2AF2 CS=2AF2 IP=0100 NV UP EI PL NZ NA PO NC
2AF2:0100 8B24      MOV    SP,[SI]          DS:0000=20CD
```

-

Bu komut yardımı ile register'lara istenilen değerler (hexadecimal olmak şartıyla) verilebilir.

Ör:

```
-R AX
AX 0000
```

- **F (Fill)** : Bellek başlangıç ve bitiş adresleri belirlenen bir alanı belirlenen bir sayı ile doldurmak amacıyla kullanılır.

Ör:

-F DS:100,120,AB

F komutundan yararlanarak DS'in 100H-120H arasındaki kısım ABH değeri ile doldurulacaktır. Bu işlemden sonra D komutu ile DS:100H adresinden itibaren incelersek işlemin olması gibi yapıldığını kontrol etmiş oluruz.

- **D (Dump / Display):** Bu komut ile belleğin istediğimiz segment'inin istediğimiz offset'inden itibaren bilgileri görebiliriz. İstersek görmek istediğimiz alanı sınırlama imkanına da sahibiz.

Ör:

```
-D DS:0
2AF2:0000 CD 20 FF 9F 00 9A EE FE-1D F0 4F 03 5A 25 8A 03 . .....O.Z%..
2AF2:0010 5A 25 17 03 5A 25 14 24-01 01 01 00 02 FF FF FF Z%..Z%.$.....
2AF2:0020 FF FF FF FF FF FF FF FF-FF FF FF FF 3B 25 4E 01 .....;%N.
2AF2:0030 16 2A 14 00 18 00 F2 2A-FF FF FF FF 00 00 CE 0B .*.....*.....
2AF2:0040 06 1E 00 00 00 00 00 00-00 00 00 00 00 00 00 00 .....
2AF2:0050 CD 21 CB 00 00 00 00 00-00 00 00 00 00 20 20 20 .!.....
2AF2:0060 20 20 20 20 20 20 20 20-00 00 00 00 00 20 20 20 .....
2AF2:0070 20 20 20 20 20 20 20 20-00 00 00 00 00 00 00 00 .....
-
```

Burada Data segment'in 0000H offset'inden itibaren olan kısmı gösterilmiştir. Eğer herhangi bir aralık verilmemiş ise 128 byte gösterir. Hexadecimal olarak belleğin dökümünü verdikten sonra yanında gösterilebilir ASCII karakter karşılıklarına da yer verilir.

Ör:

```
-D DS:100,115
2AF2:0100 8B 24 17 5A 18 BA 4A 29-A2 4F F6 C9 07 AE 07 09 .$.Z..J).O.....
2AF2:0110 20 8C 07 BA 4F 80 .....O.
-
```

- **M (Move) :** Bir yerdeki bir blok bilgiyi başka bir yere taşımak için kullanılır

Ör:

- M DS:100,110,CS:500

Bu komut ile DS:100H ile DS:110H arasında kalan bilgi CS:500H adresinden itibaren yerleştirilecektir.

- **C (Compare) :** Belirlenen aralıktaki bilgileri karşılaştırmak amacıyla kullanılan bir komuttur.

```
-C 100,110,10
2AF2:0100 AB 5A 2AF2:0010
2AF2:0101 AB 25 2AF2:0011
2AF2:0102 AB 17 2AF2:0012
2AF2:0103 AB 03 2AF2:0013
2AF2:0104 AB 5A 2AF2:0014
2AF2:0105 AB 25 2AF2:0015
2AF2:0106 AB 14 2AF2:0016
2AF2:0107 AB 24 2AF2:0017
2AF2:0108 AB 01 2AF2:0018
2AF2:0109 AB 01 2AF2:0019
2AF2:010A AB 01 2AF2:001A
2AF2:010B AB 00 2AF2:001B
2AF2:010C AB 02 2AF2:001C
2AF2:010D AB FF 2AF2:001D
2AF2:010E AB FF 2AF2:001E
2AF2:010F AB FF 2AF2:001F
```

2AF2:0110 AB FF 2AF2:0020

Bu komut 100H-110H adresindeki bilgiler ile 10H adresinden başlayan bilgileri karşılaştırır ve farklı olanları listeler.

-C 100,110,100

-

100H-110H adresindekileri yine 100H adresindekiler ile karşılaştırmaktadır. Herhangi bir farklılık bulunmadığı için ekranda hiçbir bilgi görünmez.

S (Search) : Hafıza bölgesinde istenilen adres aralığında belli bir değeri arar.

Eğer Sonuç bulursa adres olarak ekranda gözlenir.

Ör:

-S CS:100,110,65

0100 ile 0110 offsetleri arasında 65 bilgisin arar.