

1) Segment yazmaçlarının hepsi 16 bittir ve hafızanın segment olarak adlandırılan kısmını adreslemede kullanılırlar.

Code Segment Yazmacı (CS) : MS-DOS işletim sisteminde, programları oluşturan kodlar Code Segment'e yüklenir. Yani komutlar Code Segment yazmaçlarında saklanır. Bu kodlar Instruction Pointer (IP) yardımıyla bulunur. IP işlenecek komutun adres bilgisini saklar. Böylece IP, işlenecek CS'i işaret ettiğinde oradaki komutlar çalıştırılır.

Stack Segment Yazmacı (SS) : Programlar, Stack Segment (SS) denilen bir hafıza alanını geçici depolama alanı olarak kullanırlar. Örneğin bir dallanma veya interrupt olduğunda, o andaki bilgiler SS'e kaydolur. Bu yığıtlar "Last In First Out" tekniğiyle çalışır. Stack Pointer (SP) ise SS' in adresini tutar. Program gerekli interrupt'ı gerçekleştirdikten sonra, SP yardımıyla SS'deki dosyalar yerinden çıkarılır. Program kaldığı yerden devam eder.

Data Segment Yazmacı (DS) : .exe türündeki bir programda kullanılacak olan veriler Data Segment' te tutulur. Programın veri bölümünün başlangıç adresi DS içinde tutulur.

Extra Segment Yazmacı (ES) : Çeşitli string işlemlerinde kullanılan bellek bölgesine Extra Segment adı verilir. Bu bellek adresi DI ve ES içinde tutulur.

Instruction Pointer (IP) : O anda işletilen komuttan sonraki komutun offset adresini içinde tutar. Her kod işleminden sonra mikroişlemci tarafından 1 arttırılır ve işlem bu şekilde devam eder. Bu register'ın değerini, stack üzerindeki bilgiler ile dolaylı olarak değiştirmek mümkündür.

2)

YAZMAÇ (REGISTER) : Bu komut CPU' nun kullandığı yazmaçların görülmesi ve değiştirilmesi için kullanılmaktadır. Eğer tüm yazmaçlar görüntülenmek istenirse, aşağıdaki gibi bir sonuç elde edilir;

-R

AX=0000 BX=0000 CX=0000 DP=0000 SP=FFEE BP=0000 SI=0000 DI=0000

DS=1787 ES=1787 SS=1787 CS=1787 IP=0100 NV UP EI PL NZ NA PO NC

1787:01000 0000

ADD [BX+SI], AL

Eğer üzerinde değişme yapılacak bir yazmaç içinse;

-R AX

AX 0000

: 0200 şeklinde olur.

DOLDUR (FILL) : Belirlediğimiz bir bellek bölgesini, gireceğimiz bir data ile doldurmak için F komutu kullanılır. Kullanımı şu şekildedir;

-F DS:100,110,AB Bu işlem yapılırken DS'nin 100 – 110 adresi arası AB ile doldurulur.

DÖKÜM (DUMP) : Bellek bölgesinin istenilen kısmını göstermek için kullanılır.

-D komutu girildiğinde en son görüntülenen bellek bölgesinden sonraki 128 byte görüntülenir.

Bu alanı küçültmek için;

-D (offset) komutu girilir. Böylece başlangıç adresi offset olarak alınır ve sonraki 128 byte görüntülenir.

-D (başlangıç) (bitiş) komutu girildiğinde ise yalnızca belirlenen yerler arası görüntülenir.

TAŞI (MOVE) : Bu komut yardımıyla, belirli bir bölgeyi belleğin başka bir yerine taşıyabiliriz. Bunun komut aşağıdaki gibi kullanılabilir;

- M DS:100,110,CS:500 Bu komut ile DS'nin 100-110 arası bölgesi CS:500 e taşınır.

ARAMA (SEARCH) : Bir bellek bölgesinde istenilen veriyi arar. Bunun için aşağıdaki kod parçası kullanılabilir;

-S (başlangıç) (bitiş) (liste)

KARŞILAŞTIR (COMPARE) : Belirli iki aralık içindeki verileri karşılaştırmak için kullanılır.

-C 100,110,10 kodu girildiğinde 100-110 adresindeki veriler ile 10 adresinden başlayan bilgiler karşılaştırılır ve farklı olanları listelenir. Eğer bu veriler aynıysa ekrana hiçbir şey yazılmaz.

ÇEVİR (ASSEMBLE) : Bu kod yardımıyla Assembly dilinde kod yazılabilir.

- A [segment:offset] şeklinde kodlar Assambly dilinde yazılabilir.

GERİ ÇEVİR (UNASSEMBLE) : Bu kod yardımıyla, Assembly dilinde yazılmış kodlar makine diline çevrilir. Böylece ayrı bir derleyici kullanmadan makine dilinde programlar yazılabilir. Kullanım şekli aşağıdaki gibidir:

-U (başlangıç) (bitiş)

Bu başlangıç ve bitiş adresleri segment veya offset olarak verilebilir. Burada dikkat edilmesi gereken nokta, başlangıç ve bitiş noktaları seçilirken bir kod parçasının bölünmeden alınması gerektiğidir.

İZLEME (TRACE) : Bellekte bulunan komutların istediğimiz bir kısmını çalıştırmaya yarar. Bunu tek-tek yapabileceğimiz gibi bir bütün halinde de yapabiliriz. Yani;

-T (adım sayısı) kodu girildiğinde girilen adım sayısı kadar komut çalıştırılır.

-T (başlangıç) (adım sayısı) kodu girildiğinde ise başlangıç adresinden sonraki adım sayısı kadar komut çalıştırılır.