

BİLGİSAYAR DİLLERİ VE DERLEYİCİLER

Elekt. Müh. Emrehan HALICI

Bilgisayarların önemi, kendilerinden istenilen işlemleri hatasız ve çok hızlı bir şekilde yerine getirmelerinden gelmektedir. Bu işlemlerin bilgisayar tarafından, dolaylı ya da dolaysız şekilde anlanacak bir biçimde yazılmasına, "program" veya "izlençe" adı verilir.

Hatırlanacağı gibi, bilgisayar içinde tüm işlemlerin bir kod numarası bulunur. (Örneğin: Toplama yapma, bir adresten diğerine kopyalama yapma vb.) Yapılacak işlemlerin, bu kod numaralarıyla (ikili sayı sisteminde) gösterilmiş şekline "makine dili" denir. Kuşkusuz programlar makine dilinde yazılabilir; ancak bu şekilde program yazmak, hem zordur hem de yazdıktan sonra programda hatalar aramak ve değişiklikler yapmak güçlükler çıkarır. Arzu edilen, karışık numaralar yerine, insanların doğal dillerine daha yakın komutlarla bilgisayara hükmetmektir. Bu amaçla, "proglamlama dili" adı verilen birçok bilgisayar dili oluşturulmuştur. Bunlardan en yaygın olanları, **FORTRAN, COBOL, PL/I, PASCAL, ALGOL** ve **BASIC** dilleridir.

Doğal dillerde olduğu gibi, bilgisayar dillerinin de kendilerine özgü dilbilgisi ve yazım kurları bulunur. Değişik uygulamalar için, bu dillerden bazıları diğerlerine göre daha kolaylıklar sağlar ve tercih edilir. Örneğin, bilimsel uygulamalar için genellikle FORTRAN, ticari uygulamalar için de COBOL dili kullanılır. PL/I ve PASCAL, genel amaçlı programlarda kullanılan dillerdir. Mikrobilgisayarların yaygınlaşmasıyla birlikte BASIC dili büyük önem kazanmıştır. Büyük boy bilgisayarlardan, mikrobilgisayarlara kadar her tip makinede kullanım alanı bulan BASIC dili, en standartlaşmış dil olarak kabul edilebilir.

Üstte bahsolunanların veya diğer bilgisayar dillerinin herhangi birinde yazılmış olan programa "kaynak program" (source program) adı verilir. Kaynak programlar, bilgisayar tarafından doğrudan doğruya anlaşılabilir. Bu programları bilgisayar tarafından anlaşılıp uygulanabilmesi için, "amaç program"a (object program) çevi-

Kullanıcılarının elinde harikalar yaratan bilgisayarlar, henüz mantık yürütecek ve yorum yapacak akıla sahip değiller. Çözmekle yükümlü oldukları problemlerin her aşamasında karşılaşacakları değişik durumlarda, bilgisayarların ne yapacağı daha önceden kendilerine verilmiş olmalıdır.

rilmeleri gerekir. Bu çevirim işlemi, "derleyici" adı verilen bir program kullanarak gerçekleştirilir.

Şimdi, BASIC proglama dilinde yazılmış çok kısa bir kaynak programı inceleyelim: Komutları Türkçe'ye çevirerek, programı Türkçe olarak inceleyebiliriz:

10 A'YI OKU	10 INPUT A
20 B'YI OKU	20 INPUT B
30 C = A + B	30 C = A + B
40 C'YI YAZ	40 PRINT C

Görüldüğü gibi program, insanların rahatlıkla anlayabileceği bir yapıya sahiptir. Her satırın başındaki sayı, satır numarasını göstermektedir. BASIC dilinde, her komuta bir satır numarası vermek gerekmektedir. Satır numaraları, artmak koşulu ile istenilen biçimde seçilebilir.

Yukarıda verilen program, giriş biriminden (örn. klavyeden) girilecek olan iki sayıyı toplayacak ve çıkışı birimine (örn. ekrana) yazacaktır. Ancak, bu kaynak programın bilgisayar tarafından anlaşılabilmesi için makine diline çevirilmesi gerekmektedir. Bu işlemi de BASIC derleyicisi gerçekleştirecektir. Programın makine diline çevirilmesi şu şekilde olacaktır:

10 A'YI OKU : Bu komutun makine koduna çevirilmesi için, oku işlemine karşılık gelen işlem kodunun ve okunan sayının saklanacağı adresin, yani "A"nın adresinin bilinmesi gerekir. Biz makinemizde, OKU komutuna karşılık gelen işlem kodunun 63 ve "A"nın değerinin saklanacağı adresin 0095 olduğunu kabul edelim. Bu durumda bu komuta karşılık gelen makine kodu şöyle olacaktır :

63 00 95
işlem adres
kodu

Bu sayılar, bilgisayara ne yapacağını söyleyen şifrelerdir ve program işlerken bilgisayar, 63 kodu ile klavyeden bir sayı okuyacağını, peşinden gelen 0095 sayısıyla da, okuyacağı sayıyı 0095 adresinde saklayacağını anlayacaktır.

20 B'YI OKU : Bu komutu makine koduna



çevirmek için, "OKU" ya karşılık gelen işlem kodunun 63 olduğunu daha önce kabul etmiş-tik. Şimdi de "B" nin adresinin 0096 olduğunu kabul edelim. Bu durumda makine dilindeki kodumuz,

63 00 96

olacaktır. Program işlerken bu komuta sıra geldiğinde, bilgisayar 63 yayısında, yine klavyeden bir sayı okuyacağını anlayacak, ancak bu sefer okuduğu sayıyı 0096 adresinde saklayacaktır.

30 C = A + B : Bu işlemi makine dilinde tanımlamak için, parçalara ayıracak, daha sonra makine koduna çevireceğiz.

1. Kayıtlayıcıya A'yı yükle
2. Kayıtlayıcıya B'yi topla
3. Kayıtlayıcının değerini C'de sakla

Daha önceki sayılarda, bilgi üzerinde işlem yapılan yerin merkezi işlem birimi olduğunu ve burada kayıtlayıcıların bulunduğunu söylemiş-tik. Matematiksel işlemler, genellikle bu kayıtlayıcılar üzerinde yapılır.

1. İşlem için, bellekten kayıtlayıcıya kopyalama işlemi yapmaya yarayan işlem kodunu, 2. işlem için, bellekteki sayıyı kayıtlayıcıya toplamak için gerekli olan işlem kodunu ve 3. İşlem için de, kayıtlayıcıdaki sayının belleğe kopyalanmasını sağlayan işlem kodunu bilmemiz gerekiyor. Bu kodların, sırasıyla 19, 29 ve 40 olduğunu kabul edelim. Daha önceki işlemlerimizde, "A" nin adresini 0095, "B" nin adresini 0096 olarak kullanmış-tık. "C" nin adresi de 0097 olsun. Bu durumda, $C = A + B$ işlemine karşılık gelen makine kodları, sırasıyla aşağıdaki gibi olacaktır:

19 00 95

29 00 96

40 00 97

40 C'YI YAZ : "YAZ" işlemine karşılık gelen makine kodunun 64 olduğunu kabul ederek, bu komutun makine dilindeki karşılığı,

64 00 97

olacaktır.

Program parçamızın makine diline çevrilmiş hali aşağıda görülmektedir :

10 INPUT A	53 00 95
20 INPUT B	63 00 96
30 C = A + B	19 00 95
40 PRINT C	29 00 96
	40 00 97
	64 00 97

Kuşkusuz bu sayılar, daha sonra da ikili sayılara dönüştürülecek ve bilgisayar içinde o şekilde yorumlanacaklardır.

Bu yazımızda, bir kaynak programın makine koduna nasıl dönüştürüleceğini anlatmaya çalış-tık. Bu dönüştürme ve ikili sayılara çevirme iş-lemi insanlar tarafından yapılacak olsaydı, bü-yük sorunlarla karşılaşılırdı. Cysa bu amaçla kullanılan derleyiciler, kaynak programları alıp, doğrudan doğruya ikili sayılı sisteminde makine kodu üretirler. O halde, yazdığımız programlar iki aşamadan geçmektedir :

1. Derleme (Compilation)
2. Yürütme (Execution)

Derleme aşamasında, kaynak programlar derlenerek, amaç programlar elde edilir. Yürüt-me aşamasında ise, elde edilen amaç program-lar bilgisayar çalıştırılır.

Programlama dilleri ve derleyiciler sayesinde bilgisayar kullanıcıları, karmaşık makine kod-larıyla ilgilenmeden, kendi dillerine çok benzer bir şekilde programlarını yazarlar. Bu insan di-line daha yakın olan programları kendi anlaya-cağı biçimde tercüme etmek görevi, yine çalış-kan dostumuz BİLGİSAYAR'ın sırtındadır. ■



"Bu elmas bir yüzük değil, 1.300 ke-limelik teklifimi içeren bir 64K RAM çip-dir."