

## **PROGRAMLAMA DİLLERİ**

**Arş. Gör. Levent ELDENİZ**  
**MARMARA ÜNİVERSİTESİ**  
İletişim Fakültesi

Bilgisayar bilimi bilgi işlemin araçlarını ve gelişim sürecini içerir. Bilgisayarlardan problem çözümünde yararlanırken kullanıcılarla bilgisayarlar arasında bir iletişim kurulması zorunluluğu vardır. Bilgisayarlar 0 ve 1 sembolleri ile ifade edilen elektriksel bir dille konuşurlar. Bu makine dili'nin ayrıntıları her makinenin tasarlanması sırasında üretici tarafından belirlenir. Kullanıcılar da elbette kendi doğal dillerini kullanırlar. Bu da bilgisayar ve kullanıcı arasındaki iletişim aralığını oldukça daraltır. Kullanıcılar çok geniş bir alana yayılan problemlerle bilgisayarlarının başına otururlar. Ancak mevcut makine dili ile bu problemlerin bilgisayara aktarılması oldukça güçtür. Bu güçlük programlama ve programlama dili kavramlarının ortaya çıkmasına yol açmıştır (1).

Programlama dillerini sınıflandırmanın iki yolu vardır: Seviyelerine ve uygulama alanlarına göre.

Programlama dillerinin dört seviyede toplayabiliriz.

1. Konuşma diline yakın programlama dilleri
2. Yüksek düzey programlama dilleri
3. Assembly dilleri
4. Makine dilleri

Konuşma diline yakın programlama dilleri diğer dillerle karşılaştırıldıklarında en üst düzeydedirler ve bu dillerdeki ifadeler İngilizce gibidir. Bu diller esas olarak "nasıl yapılır"dan çok "ne yapılacak" ifadelerinin hakim olduğu komut dilleridirler. İstatistik dilleri olan SPSS ve SAS, veritabanı dilleri NATURAL ve IMS bunlara örnek verilebilir. Bu diller profesyonellerin

kendi işlerini yaparken kullandıkları terminolojiye uygun bir şekilde, programlama tecrübesi veya programcı desteği gerekmeden programlama yapabilmeleri düşünülerek geliştirilmiştir.

Yüksek düzey programlama dilleri en çok kullanılan programlama dilleridir. Böyle olmakla beraber aslında konuşma diline yakın değildirler. Bu diller diğer programcıların kolayca okuyup anlayabileceği yazım şekillerine ve algoritmalara izin verirler. Ayrıca yüksek düzey diller genellikle taşınabilirlik özelliğine sahiptirler. Bu da farklı makinelerde kullanılacakları anlamına gelir. Yani bir program bir makineden diğerine metin düzeltimi gerekmeden kolayca taşınabilir. Bu anlamda yüksek düzey diller "makineden bağımsız" olarak adlandırılırlar. Yüksek düzey dillere örnek olarak Pascal, APL ve FORTRAN (bilimsel uygulamalarda), COBOL (veri işleme uygulamalarında), SNOBOL (metin işleme uygulamalarında), LISP ve PROLOG (yapay zeka uygulamalarında) ve PL/I (genel amaçlı uygulamalarda) verilebilir (2).

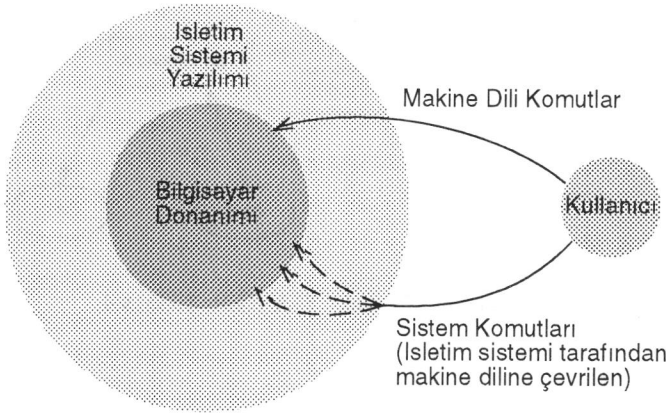
### **Assembly ve Makine Dilleri**

Assembly ve Makine dilleri donanım bağımlı programlama dilleridirler. Her makine kendi makine ve assembly diline sahiptir. Makine dili bir CPU tarafından komut olarak alınıp uygulanan bellek veri guruplarını temsil eden ikili kod dizilerinden oluşur. Assembly dili makine dili komutlarının sembolik ifadelerini içeren bir programlama dilidir. Makine diline kıyasla daha az sıkıcı bir programlamaya izin verir. Temel makine mimarisi bu dilleri etkili programlamada kaçınılmaz kılar.

Aşağıdaki üç benzer program parçası yüksek düzey, assembly ve makine dilleri arasındaki farkları göstermektedir (3).

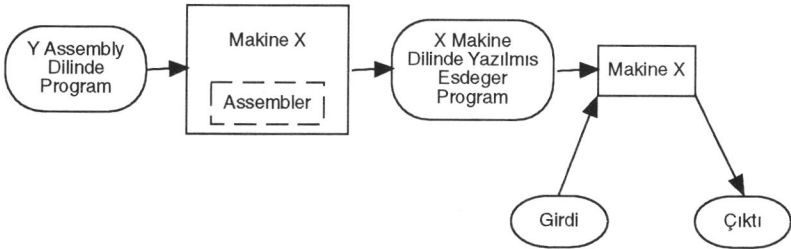
<b>Pascal</b>	<b>Assembly Dili</b>	<b>Makine Dili</b>
Z:=W+X*Y	L 3,X M 2,Y A 3,W ST3,Z	41 3 0C1A4 3A 2 0C1A8 1A 3 0C1A0 50 3 0C1A4

Örnekte görüldüğü gibi bire bir karşılaştırmada assembly ile makine dili arasında yakın bir ilişki olduğu görülebilir.



**Şekil 1.** Bir bilgisayar sisteminin katmanları. Kaynak : GEAR, C. William, **Computer Organization and Programming With an Emphasis on The Personal Computer**, 1985, s.7.

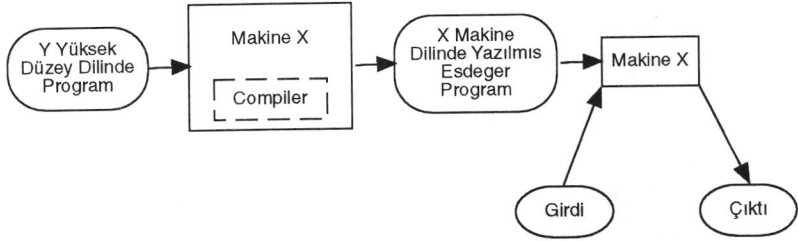
Yüksek düzey bir dilde veya bir assembly dilde programın makine koduna çevrilip çalışabilmesi için bir arabirime ihtiyaç vardır. Bu arabirimlerden en çok kullanılanları "assembler, compiler ve interpreter"dır.



**Şekil 2.** Assembler arabirimi. Kaynak : TUCKER, 1986, s.4.

## Compiler

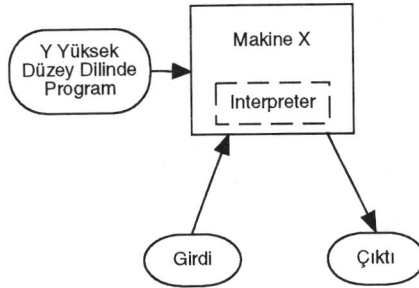
Yüksek düzeyli program komutlarını CPU'nun doğrudan doğruya anlayıp yerine getireceği ikili kodlara dönüştüren bir çeviri programıdır. Derleyici programın tümünü ikili koda dönüştürerek sırayla yürütülmesini sağlar. Programda herhangi bir değişiklik yapıldığında programın kaynak kodunun tümünün yeniden derlenmesi gerekir (4).



**Şekil 3.** Compiler arabirimi. Kaynak : TUCKER, 1986, s.4.

### Interpreter

Yüksek düzeyli bir dilde ifade edilmiş olan deyimlerin uygulanması için kullanılan çeviri programıdır. Programın her deyimini işlem sırasında ikili koda çevirir ve hemen icra eder. Derlenen programlara göre yorumlanan programlar daha yavaş işler (5).



**Şekil 4.** Assembler arabirimi. Kaynak : TUCKER, 1986, s.4.

Assembler ve Compiler arabirimleri programın kaynak kodunu kullandığımız makinenin makine diline çevirirler. İnterpreter ise programın her deyimini çalışma esnasında anında çevirir ve çalıştırır. Compiler ile programın bir defa makine diline çevrilmesi çalışması için yeterlidir. Ancak interpreter ile programın her çalışmasında tekrar çevrilmesi gereklidir.

Bazı diller APL, PROLOG ve LISP gibi, sadece interpreter edilen/yorumlanan dillerdir. FORTRAN, Pascal, COBOL, PL/I, SNOBOL, C, Ada ve Modula-2 gibi diller ise compiler edilen/derlenen dillerdir. Interpreter tarafından çalıştırılan diller hataların bulunması ve düzeltilmesi açısından com-

piller tarafından çalıştırılan dillere göre daha kullanışlıdır. Deneysel ve eğitim amaçlı programcılık çalışmalarında interpreter diller daha sık kullanılır.

Programlama dillerinin karşılaştırılması ve değerlendirilmesinde gözönünde bulundurulması gereken dokuz kriter vardır. Bunlar şunlardır (6):

1. Anlamlılık
2. İyi tanımlama
3. Veri tipleri ve yapıları
4. Modülerlik
5. Giriş-Çıkış imkanları
6. Taşınabilirlik
7. Etkinlik, verimlilik
8. Eğitilebilirlik
9. Genellik

## **PASCAL**

Pascal 1970'lerin başında algoritma tasarımı ve bilgisayar biliminin ilkelerini öğretmek için tasarlanmıştır. 1960'larda aynı amaçlar doğrultusunda geliştirilen ALGOL (Algoritmik Dil)'ün devamı niteliğindedir. Pascal matematiğin, veri işlemin ve yapay zekanın pratik uygulamalarında kullanılmaktadır. ALGOL 1960'larda programlama dili tasarımı kullanılmıştır. Pascal da 1970'li ve 80'li yıllarda aynı görevi yerine getirmiştir (7).

## **FORTRAN**

FORTRAN 1954'te geliştirilmiştir. Dolayısı ile halen kullanılmakta olan en eski ve en güçlü yüksek düzey dillerdendir. FORTRAN (Formula Translating System - Formül Çevrim Sistemi) en çok bilimsel ve mühendislik programlama çalışmalarında kullanılmaktadır (8).

## **COBOL**

1959'un Mayıs ayında veri işlem gruplarının ortak bir veri işlem diline ihtiyaç duyması ile ortaya çıkmıştır. 1960'ın Nisan ayında endüstriden ve kamudan bilgisayar üreticilerinin ve kullanıcılarının biraraya gelmesi ile ortaya çıkan ve düzenlenen CODASYL (Conference on Data Systems Languages)'de ilk versiyonu COBOL-60 adıyla yayınlanmıştır. COBOL (Common Business Oriented Language) en çok veri işleme uygulamalarında kullanılır.

maktadır. Programların ve programlama tekniklerinin bir makineden diğeri- ne kolaylıkla aktarılabilmesi ve paylaşılabilmesi amacıyla bir ortak dil olarak geliştirilmiştir. Bu açıdan COBOL İngilizce konuşma diline benzer bir programlama dilidir. Sözdizimi dikkatsiz bir gözlemcinin bile bir programcı gibi programı okuyup anlayabileceği gibi tasarlanmıştır (9).

## C

C programlama dili çok geniş bir alanda çalışabilmek için tasarlanmıştır. C'nin gelişimi ile UNIX işletim sisteminin gelişimi paralellik izler. 1969'da UNIX Bell Laboratuvarlarında tamamı assembler ile yazılarak geliştirilmiştir. Aynı zamanda Kenneth Thompson tarafından B adında deneysel bir dil geliştirilmiştir. 1972'de C B'nin bir genişletmesi olarak tasarlanmıştır. 1973'te UNIX bu sefer yüzde doksanı C ile tekrar yazılmıştır (10).

Günümüzde ise kişisel bilgisayarların yaygınlaşması ile bir çok yeni programlama dili ortaya çıkmıştır. Özellikle son yıllarda yaygın kullanım bulan dBase, FoxPro, Progress, Paradox gibi veri tabanı dillerini, VisualBasic gibi Multimedya programlama dillerini, Turbo Pascal ve C++ gibi genel amaçlı programlama dillerini bunlara örnek olarak verebiliriz.

## DİPNOTLAR

- (1) SCHEID, Francis, **Theory and Problems of Computers and Programming**, 1983, s. 1.
- (2) TUCKER, Allen B. Jr., **Programming Languages**, 1986, ss. 2-3.
- (3) Ibid., s. 3.
- (4) AYDIN, Emin D., **Bilişim Sistemleri Sözlüğü Bilgisayar Bilgişlem ve Telekomünikasyon**, 1992, s. 121.
- (5) Ibid., s. 333.
- (6) TUCKER, Op.cit., s. 11.
- (7) Ibid., s. 16.
- (8) Ibid., s. 63.
- (9) Ibid., ss. 102-103.
- (10) Ibid., ss. 412-413.