

Programlama Dilleri

2.1 Programlama Dilleri ve C

Bölüm başlığının aksine, burada programlama dillerinin çok uzun ve ilginç gelişim öykülerine girmeyecek ve onların tam sınıflandırmasını yapmak gibi zor bir işe girişmeyeceğiz.

Onun yerine, asıl konumuz olan, C programlama dilinin kısa geçmişi-
şini açıklayacak ve onun programlamadaki yerini belirtmeye çalışacağız.
Bunları bilmek, gitmek istediğimiz yolu daha iyi aydınlatacaktır.

Programlama dilleri, tarihi gelişimleri, makina diline yakın ya da uzak oluşları, yapısal olup olmadıkları, veri tipi bildiren mi, bildirmeyen mi, derlenen mi yoksa yorumlayıcı mı, genel amaçlı mı yoksa özel amaçlı mı oldukları, iş yapma yöntemleri gibi çok farklı bakış açılarına göre sınıflandırılır.

Programlama dillerini eksiksiz sınıflandırmak çok zordur. İlgilenenler [2] web sayfasına bakabilirler. Yapılan sınıflandırmalar, programlama dillerini kesin çizgilerle birbirlerinden ayırmaz. Başka bir deyişle, bir dil birden çok sınıfa girebilir.

İşin nasıl yapıldığı açısından sınıflandırmada *yordamsal* (procedural) diller önemli bir sınıf oluşturur. Yordamsal dillerde, belli işleri yapan belli yordamlar (procedure) vardır. Yordamı bir iş yapan deyimler topluluğu (blok) olarak tanımlayacağız. Çoğu dilde bu bloklara *fonksiyon* ya da *procedure* denilir. C dili *yordamsal* bir dildir; yani işleri fonksiyonlarla yapar.

Yordamsal dillerin önemli bir alt sınıfı *buyrukçu* (*imperative*) dillerdir. Buyrukçu diller komut yönelteli dillerdir. Her bir işi yaptıran belirli komutlar

vardır.

C dili buyrukçu (imperative) bir dildir.

Yapısal Programlama

1960ların sonuna kadar programlama karmaşık, güncellenmesi ve düzeltilmesi zor ve çok maliyetli bir iş idi. İlk programlama dillerinde array dışında bir yapı yoktur. Her öznitelik değişkenlerle ve array ile tanımlanırdı. Bu kısıtlı yapıda, örneğin basit bir personel uygulaması yazmak bile çok zordu. Çünkü bir kişiye ait *ad*, *soyad*, *sicilno*, *doğum_tarihi* vb bilgileri bir arada tutan bir yapı yoktu. 1960'lı yıllar yeni arayışlarla geçti. 1971 yılında *Niklaus Wirth* ilk yapısal programlama dili sayılan *Pascal* dilini ortaya koydu. Hiç bir dil hemen ortaya çıkmaz. Her dilin öncüleri vardır. Pascal dili de kendi öncülerinin başlattığı işi noktalandı. Birbirleriyle ilişkili olan değişkenleri bir arada tutan ve adına *record* denilen yapıyı tanımladı. *Sıralama*, *tekrarlama* ve *program akışını yönlendirme* (karar verme) eylemlerini yapan deyimlerden oluşan yapıları kurdu. Pascal, kendisinden önceki kuşak olan FORTRAN soyundan gelen programlama kavramını değiştirdi. Uzun süre, bilgisayar programcılığını öğretmek için üniversitelerde kullanılan başlıca araç oldu.

C dili de Pascal gibi *yapısal* bir dildir. Pascal'daki *record* terimi yerine C dili *struct* (structure) terimini kullanıyor.

2.2 C Dilinin Kısa Tarihçesi

C dili ABD'de AT&T şirketinin *Bell* laboratuvarlarında 1969-1973 yıllarında *Dennis Ritchie* tarafından yazılmıştır.

C dili genel amaçlı *buyrukçu* (*imperative*) bir programlama dilidir. Yapısaldır. Değişken bildirim ve değişkenlerin kapsam alanları, program akışının denetimi için kendine özgü sözdizimi (syntax) kuralları vardır.

UNIX İşletim Sisteminin ilk sürümü, alt-düzeyli PDP-7 assembler dili için yazıldı. R.McClure PDP-7 için TMG adını verdiği bir dil yazdı. TMG kullanarak FORTRAN dili için bir derleyici yazmaya başlayan Ken Thompson, sonunda, derleyici yerine adına *B* dediği bir İşletim sistemi yarattı. *B* dili UNIX İşletim Sistemlerinin geliştirilmesinde önemli bir araç oldu.

Bell Laboratuvarlarına yeni alınan PDP-11 makinası için bir işletim sistemi geliştirmek üzere *Dennis Ritchie* ve ekibi işe koyuldu. *B* diline dayalı olarak yeni bir dil yarattılar ve adına *C* dediler. O zamanki UNIX işletim

sisteminin hemen her komutunu yeniden *C* dili ile yazdılar. Bu yeni UNIX dilinin iki önemli niteliği vardı:

Taşınabilirlik (portability) Yaratılan UNIX İşletim sistemi, mevcut olanlar yanında sonradan ortaya çıkacak yeni makinalarda da kullanılabilir olacak şekilde tasarlandı. Bunu başarmak için üç işin yapılması gerekli idi.

1. Yaratılan yeni UNIX dilini ve onun uygulamalarını yapmaya yarayan *C* dilini, her makinanın assembly diline dönüştüren derleyiciler yazılması gerekti.

Üretici firma, kendi makinası için bu işi yapan derleyiciyi yazmak ya da yazdırmak zorunda kaldı. Tabii, bu zorlama yasayla yapılmadı. Yeni makinanın piyasaya girebilmesi için üretici firmanın yapmak zorunda kaldığı sıradan bir iş oldu. Yani UNIX derleyicisi hazırlamak makina üretiminin bir parçası haline geldi. Bu uygulama günümüze dek uzandı. Yeni makina üreten şirket, onun yaygın dillerdeki derleyicilerini de ortaya koymazsa, makinasını satamaz. Tabii, yaygın diller için, üçüncü şahıs ve şirketler de derleyici yazabilirler ve yazıyorlar. O nedenle, örneğin, Unix ya da Windows işletim sistemi altında çalışan çok sayıda *C* derleyicisi vardır.

2. Sonuçta, söz konusu derleyici kullanılarak, *C* dilinde yazılmış bir program, büyük ölçüde her yeni makinanın assembly diline dönüştürülebilir hale geldi.
3. *C* dilinden yeni makinanın assembly diline dönüşemeyen az sayıda kodlar varsa, onlar için özel kodlar yazılabiliyordu.

Geliştirilebilirlik (improvability) Böylece, *C* dilini öğrenen programcı, her yeni makina için program yazabilir hale geldi. Böyle oluşu, çok sayıda sistem yöneticisinin ve programcının *C* diline yönelmesine neden oldu. *C* diline yönelenler, kendi sistemleri için, gerekli fonksiyonları ekleyerek *C* dilini geliştirmeye başladılar.

Bu kadar etkili olan *C* dilini Bell'in duvarları içine hapsetmek olanaksızdı. Kısa zamanda, *C* nin öncüleri Bell Laboratuvarından dışarıya sızmaya başladı. Her yerdeki programcılar, Bell'den çıkan kodları kullanarak program yazmaya başladılar. Doğal olarak, kurumlar ve programcılar, kendilerine uygun olanları seçerek, birbirlerinden farklı *C* dili yarattılar. Böyle olması, her yeni makina alımında sistem yöneticilerinin önüne büyük zorluklar

koymaya başladı. Ortaya çıkan sorunu çözmek için ABD Ulusal Standartlar Enstitüsü ANSI (American National Standards Institute) C dili için standartları oluşturmak üzere bir komite kurdu. Bu komite 1989 yılında ANSI C adıyla anılan işletim sistemini standart olarak ortaya koydu. Bazı istisnalar dışında, ANSI C dili, UNIX İşletim sistemleri için standart olarak kabul edildi.

1990 yılında ANSI C, Uluslararası Standartlar Örgütü ISO (International Standards Organization) tarafından da standart C dili olarak kabul edildi.

Doğal olarak, donanımın yetenekleri artarken diller de ona paralel gelişiyor. C dili standartları, 1999 yılında yeniden ele alındı. C99 adı verilen standartlar, C89 standartları ile ona yapılan eklerden oluşuyor. Daha önce yazılmış derleyiciler ve programlar, C89 standardı ile sınırlıdır. Bazı yazılım şirketleri c99 standardına uyan derleyicileri piyasaya sürmüşlerdir. Programcıların bazıları, C89 standardı ile yazmayı sürdürmektedirler. Bu derste C89 standardına uyan derleyiciler yeterli olacaktır.

2.3 Neden C ?

C dilinin ortaya çıktığı zamanlarda ve sonraki zamanlarda yazılan çok iyi diller ve işletim sistemleri vardır. Örneğin *Pascal*, *Basic*, *Java* kendi amaçları doğrultusunda çığır açan önemli dillerdir. Ama C dili, bütün zamanların vazgeçilmez dili olarak kalmayı başarmıştır. Bunun nedenleri vardır:

- C dili makina dili ile üst-düzey diller arasında bir yerdedir. Hem sistem ile ilgili işleri, hem üst-düzey ile ilgili işleri yapabilir.
- C ile yazılan programlar, makina diline yakın olduğu için hızlı çalışır.
- C güçlü, esnek ve gelişebilir bir dildir. Bilgisayar ile ilgili her alana uygulanabilir: *İşletim sistemi*, *ağ*, *web*, *grafik*, *hesap tabloları*, *derleyici yazmak*, *başka dil yazmak*, *vb* işler için başvurulmuş ilk araçtır. Bu gün kullanılan bir çok *uygulama programı*, *işletim sistemi*, *dil* ve *derleyici* C dili ile yazılmıştır. Örneğin, platform bağımsızlığı yaratan *Java* dili C ile yazılmıştır. Nesne tabanlı dillerin son aşamaları sayılan *Python* ve *Ruby* dilleri C ile yazılmıştır.
- C çok yaygınlık kazandığı için, hemen her platform için yazılmış derleyicilere ve zengin kütüphanelere sahiptir.

- C ile yazılan kaynak programlar taşınabilir. Örneğin, Windows işletim sistemi altında standart C dili ile yazılan bir C programı DEC VAX sisteminde (aynen ya da küçük değişikliklerle) derlenip çalıştırılabilir.
- C dilinin en önemli özelliklerinden birisi, makina diline yakın oluşu nedeniyle, kendi kendisini genişletebilmesidir. Programcı, standart C dilinde olmayan kodları kendi başına yaratabilir ve onları C dilinin parçaları imiş gibi çalıştırabilir.
- C dili ile yazılmış bir program, esasta C ile yazılmış fonksiyonlardan oluşan bir kümedir. Standart hale gelen fonksiyonlar arasında birbirleriyle ilişkili olanlar bir araya getirilerek *header files* (başlık fonksiyonları) denilen paketler oluşturulur. Bunlar C kütüphanesinde dirler. Programcı, başlık fonksiyonlarından istediğini `#include ...` deyimi ile kendi programına katabilir. Bu şekilde programa çağrılan fonksiyonlar, sanki o program içinde imişler gibi işlerler.

2.4 C++ Nedir?

C++, C dilinin bir üst kümesidir, C den türetilmiş ve yeni özelliklere sahip bir işletim sistemidir. Buna NYP (Nesne Yönelimli Programlama) [OOP-Object Oriented Programming] deniliyor. C++, C dilinin yaptığı her işi yapabilir. Ayrıca NYP olma özeliğine sahiptir. NYP alanındaki en önemli gelişmelerden birisi *Java* dilidir. Java dili C ile yazılmıştır. NYP olması yanında platform bağımsızlık kavramını ilk getiren dildir. Bu gün bilgisayar teknolojisinin her alanına uygulanabilen genel amaçlı, platform bağımsız dillerin lideridir. Ticari uygulamalarda ve üniversitelerdeki öğretim programlarında yakın zamanlara kadar egemenliğini sürdürdü.

C dilinin ortaya çıkışı gibi, C++ dili de AT&T laboratuvarlarında ortaya çıkmıştır. 1980 yılında Bjarne Stroustrup, C dili üzerine oturan C++ dilini yarattı.

C++ dilinin başlangıcını 1979 yılına kadar götürmek mümkündür. O yıllarda *Bjarne Stroustrup*, benzetim (simulasyon) konulu doktora tezini hazırlıyordu. Kullandığı dil *Simula 67* idi. Bu dilin öncüsü olan *Simula* dili, nesne yönelimli dil paradigmasının atası sayılır. Bjarne Stroustrup, *Simula*'nın programcılıkta yararlı olabileceğini sezdi. Ama *Simula*'nın çok yavaş kaldığını da görüyordu. O nedenle, C dilini kullanarak *Simula*'dakine benzer sınıflar (classes) yaratmaya başladı. Giderek, C dilinin bir üst kümesi olan yeni bir dil ortaya çıktı. C'ye ekler anlamında *C++* denilen yeni

dil, nesne yönelimli programlamanın temel niteliklerini de ortaya koydu.

Aslında C++ dili karma (hybrid) bir dildir. Aynı zamanda Nesne Yönelimlidir, yapısaldır ve yordamsal bir dildir. Bu nitelikleri yanında veri tipi soyutlamaya (data abstraction) yatkın oluşu onun programcılar arasında hızla yayılmasını sağladı. Bu gün, C++ her platform için derleyicilere ve zengin bir kütüphaneye sahiptir. Son yıllarda yapılan istatistiklerde Java ve Python dilinin gerisine düşmüş görünse de, programcılıktaki önemini korumaya devam etmektedir.