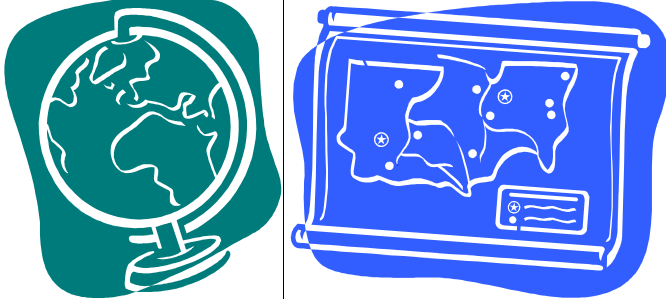


## Modelleme

**Modelleme**, gerçek dünya nesnesinin yaklaşık olarak ifade edilmesidir. Bir model, bütünü oluşturan parçaları ve bu parçalar arası etkileşimin nasıl olduğunu gösterir. Yaklaşık olarak ifade, gerçek sistemin sadece bizim istediğimiz özelliklerini almamız yüzündendir. Mesela dünya maketi, gerçek dünyanın bir modelidir. Bu model kıtaları, denizleri ve dünyanın şeklini ifade eder. Eğer sadece bu bilgilerle ilgileniyorsak bu model bizim için faydalıdır. Fakat dünyanın uzaydaki konumu, hızı, muazzam büyüklüğü bu model tarafından ifade edilmez. Kağıt üzerindeki bir harita da dünyanın bir modelidir. Fakat burada dünyanın fiziksel şekil özelliği de gözardı edilmiştir.



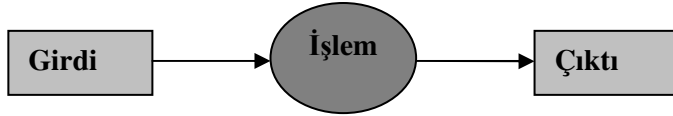
**Şekil 1.5** Dünyanın maket ve harita olarak iki değişik modeli

Soyutlamanın, modelleme için temel araç olduğunu görebiliriz. Modelleme sırasında, gerçek dünya nesnesi bir bütün olarak görülür ve sadece ilgi duyulan özellikleri alınır. Bu yüzden bir sistemin, her defasında değişik özelliklerini alarak çok sayıda modeli oluşturulabilir. Mesela bir şehir için, sadece yer şekillerini (yükselti ve yeşillik alanları), bina ve tesisleri, turistik yerleri gösterir haritalar çizilebilir.

Artık bilgisayarın modelini (veya haritasını) çıkarabiliriz. Bilgisayara değişik açılardan bakıp, değişik özelliklerini yakalayabiliriz. Buna göre de değişik modellerini oluşturabiliriz.

## Girdi-İşlem-Çıktı Modeli

Bilgisayar için en bilinen ve temel modellerden birisi **Girdi-İşlem-Çıktı (Input-Process-Output)** modelidir.



**Şekil 1.6** En basit bilgisayar modeli, girdi-işlem-çıktı

Bu modelde odaklandığımız özellik veridir ve bilgisayarla veri arasındaki ilişki ifade edilir. Mesela 2 ve 3 sayıları girdi olarak verilir. Toplama işlemi yapılır. Çıktı olarak 5 sayısı üretilir.

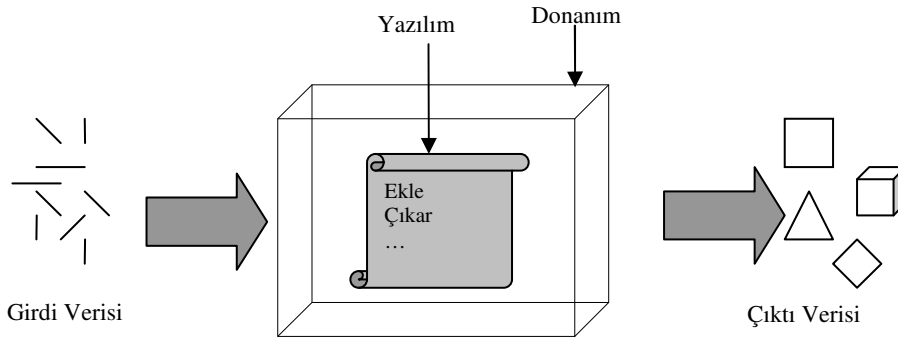


**Şekil 1.7 Ot yiyip süt veren bir inek, çok faydalıdır ve girdi-işlem-çıkı modeline oturur**

Bu modeli anlamak zor olmamalıdır. Çünkü etrafımızda bu modele göre çalışan pek çok şey görebiliriz. Mesela bir inek girdi olarak ot yer. Bunu işler ve çıktı olarak süt üretir.

Bu modelin kötü tarafı çok genel olmasıdır. Çünkü bir inek bile modelin içerisine dahil olmaktadır! İşlem basamağını yazılım ve donanım olarak açarsak biraz daha gelişmiş bir model elde ederiz. Bu modele şimdilik veri-donanım-yazılım modeli diyelim.

### Veri-Donanım-Yazılım Modeli



**Şekil 1.8 Veri-donanım-yazılım modeli**

Bu modele göre, bir bilgisayar sisteminin 3 ana bileşeni vardır.

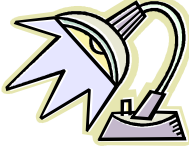
- **Donanım (hardware)**
- **Yazılım (software)**
- **Veri (data)**

**Donanım**, bilgisayarın fiziksel olarak mevcut, elle dokunulabilir tüm bileşenleridir. Bilgisayarın kasası ve içindeki elektronik parçalar, monitör, klavye gibi tüm cihazlar bu gruba girer. Bu bileşenlerin bir araya gelip oluşturduğu programlanabilir elektronik cihaza bilgisayar diyoruz. Bütün donanım bileşenlerini 5 grupta toplayabiliriz.

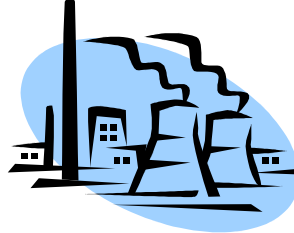
1. **CPU** (Central Processing Unit / Merkezi İşlem Birimi -MİB): Komutların işletildiği donanım parçasıdır.
2. **Bellek**: Veri ve komutlar bellekte *birlikte* tutulurlar. Veri ve komutların CPU tarafından erişilebilmesi için mutlaka bellekte olmaları gerekir. Elektriğe bağımlıdır.
3. **Girdi Cihazları**: Verilerin girildiği klavye ve fare gibi cihazlardır.
4. **Çıktı Cihazları**: Sonuç verilerinin sunulduğu ekran ve yazıcı gibi cihazlardır.
5. **Depolama Cihazları**: Veri ve yazılımların *uzun süreli saklanması* için kullanılan sabitdisk ve CD sürücü gibi cihazlardır. Elektrikten bağımsızdır.

CPU ve bellek, bilgisayarın çalışması için olmazsa olmaz donanımlardır. Fakat diğer gruplardaki parçalar (mesela ekran ve klavye) olmadan da bilgisayar çalışmaya devam edebilir. Bu son üç gruptaki cihazlar, **çevresel cihazlar (peripheral devices)** olarak bilinirler.

**Yazılım** bu donanımı bir işi başarmak üzere yönlendiren komutlar topluluğudur. Mesela Internet tarayıcınız (Internet Explorer ve Firefox gibi) modem, bellek, CPU ve ekran kartı gibi donanım cihazlarını kullanarak uzaktaki bilgisayarlarda bulunan web sayfalarını bilgisayarınızın ekranında görüntüler. Klavye ve fare vasıtasıyla sizden yeni komutları alır.



**Uygulama Programları** (lamba gibi) doğrudan bize hizmet ederler



**Sistem Yazılımları** ise (elektrik santralının ev aletlerine hizmet etmesi gibi) uygulama yazılımlarına hizmet ederler.

### Şekil 1.9 Sistem ve Uygulama yazılımı arasındaki fark

Yazılım, uygulama yazılımları ve sistem yazılımları olarak ikiye ayrılabilir. **Uygulama yazılımları** doğrudan kullanıcı isteklerini karşılamak için üretilmiş yazılımlardır. Mesela kelime işlem ve e-posta yazılımları gibi. **Sistem yazılımları** ise, uygulama yazılımlarının *üretilmesine* ve *çalıştırılmasına* hizmet ederler. Mesela programlama dillerinin derleyicileri, uygulama programı yazmak içindir. Doğrudan son kullanıcıya hitap etmezler. İşletim sistemleri (Linux, Windows XP...) ise, uygulamaların çalıştırılması için oluşturulmuş en önemli sistem yazılımlarıdır. Sistem ve uygulama yazılımları arasındaki farkı anlamak için, masa lambasıyla elektrik santralını

düşünebiliriz. Masa lambası doğrudan bizim ihtiyacımıza cevap verdiği için uygulama yazılımı olarak düşünülebilir. Elektrik santralinden ise, bizim doğrudan faydalanmamız mümkün değildir. Bunun yerine kullandığımız elektrikli aletlerin çalışmasına hizmet eder.

Donanım yazılıma hizmet ettiği gibi, yazılım da verinin işlenmesine hizmet eder. Bir yazılım, faydalı bir iş üretebilmek için dış dünyayla etkileşim içinde olmalıdır. Bu yüzden çalışması sırasında **girdi** verisi kabul eder. Yapılan işlem sonrası bir **çıkıtı** verisi oluşturur. Mesela hesap makinesi yazılımı, kullanıcıdan sayıları ve yapılacak işlemi alabilmeli ve sonuç bilgisini yine kullanıcıya ulaştırabilmelidir. Bir kelime işlem yazılımı yazdığımız metni ve yaptığımız font ayarlarını girdi olarak kabul eder. Çıkıtı olarak ekran görüntüsü, basılı kağıt veya disk üzerinde dosya üretebilir. Oyunlar tuş ve fare hareketlerini girdi verisi olarak kullanırlar. Çıkıtı olarak ekranda oyun karakterlerinin yeni durumunu yansıtan çizimlerini üretebilirler. Demek ki, girdi ve çıkıtı verileri sadece metin formatında veya sayısal olmak zorunda değildir. Burada belirtmemiz gereken önemli bir husus girdi verisinin bir kısmının program kodu içine gömülmüş olabileceğidir. Mesela çemberin çevresini hesaplayan bir programda Pi sayısı program kodu içerisinde bulunabilir.

Temel bileşenleri gördükten sonra, bunların birlikte çalışma şeklini gerçek dünya örnekleri üzerinde anlamaya çalışalım.

Bir bilgisayarın çalışma şeklini yemek yapma eylemine benzetebiliriz. Aşçı, bir kağıt üzerinde yazılı yemek tarifine göre un, yağ ve şeker gibi malzemeleri kullanarak pasta yapar. Yemeği hazırlama esnasında kaşık, fırın ve çeşitli kaplar kullanır. Bu analogide un ve yağ gibi malzemeler girdi verisi olarak düşünülebilir. Aşçı ve yemek hazırlama esnasında kullanılan gereçler donanımdır (Daha ince düşünürsek aşçı CPU'yu, tarifin yazılı olduğu kağıt belleği, kap-kacak diğer çevre cihazlarını temsil edecektir). Yemek tarifi yazılımı ifade eder. Sonuçta oluşan pasta ise çıkıtı verisine denk gelecektir.



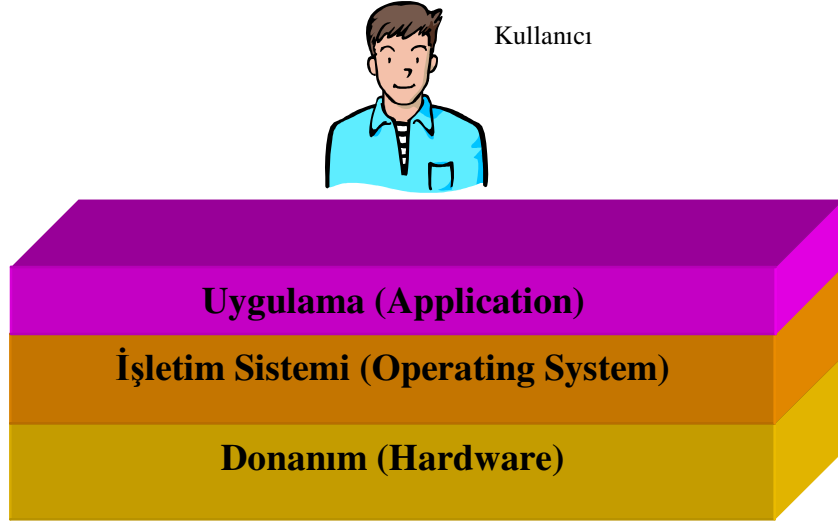
**Şekil 1.10 Bilgisayarın çalışma şekli ve yemek yapma eylemi benzerlik gösterir**

Bilgisayarı anlamak için kendimize de bakabiliriz. Bedenimizi donanım olarak düşünebiliriz. Ruhumuz ise yazılım olur. Bedenimiz ruhumuzun direktifleri doğrultusunda hareket ettiği gibi, donanım da yazılımın direktifleri doğrultusunda hareket eder. Ruhumuz bu hareketleri belirlerken, duyu organlarımız vasıtasıyla (bunlar girdi cihazları olarak düşünülebilir) çevreden gelen uyarıları

dikkate alır. Dış dünyadan gelen bu bilgiler/uyarılar ruhumuz tarafından işlenir ve cevap olarak bir hareket yapılması sağlanır. Bunlar da çıktı cihazları olarak düşünebileceğimiz, ağızımız, kollarımız gibi organlarımız vasıtasıyla olur. Mesela bir köpeği görmemiz girdi verisi, bacakların koşmaya başlaması çıktı verisi olarak düşünülebilir. Yazılım da girilen veriyi işleyip çıktı cihazlarında yeni bir veri oluşturur. Bu örnek bize yazılımın, tıpkı ruh gibi, elle tutulur, gözle görülür bir şey olmadığını da gösterir.

## Kullanıcı Uzaklık Modeli

Bizim bu kitap boyunca takip edeceğimiz model ise kullanıcı uzaklık modelidir. Bu modelde bilgisayar kullanıcıya yakın ve uzak olan katmanlar şeklinde gösterilir. Bunun soyutlama konusunda, bilgisayar katmanlar şeklinde ifade etmemize çok benzediğine dikkat edin.



**Şekil 1.11 Kullanıcı Uzaklık Modeli**

Şekilden, bilgisayarın katmanlar halinde inşa edildiğini görüyoruz. Neden böyle bir yapıya gidildiğini, bilgisayarı daha kolay anlayabileceğimiz bir örnekle izah edelim.

Farz edelim ki gözleri görmeyen, burnu koku almayan ve kulakları duymayan bir köpeğiniz var. Doğuştan böyle, bu yüzden etrafındaki dünya hakkında hiçbir fikri yok! Herkes gibi, köpeğinizin bu eksileriyle beraber artıları da var. Bazı basit komutları, ileriye adım at, sağa dön, ayağını kaldır, başını indir ve ısır gibi, emrinizle yapabiliyor. Fakat ona emir verebilmenin tek yolu sırtına hafifçe vuruşlarda bulunmak. Köpeğinizin anladığı dil, mors alfabesinde olduğu gibi bir dizi halinde, sırtına hafifçe vuruş ve duraklardan oluşuyor. Mesela “ısır” komutunu vermek için (. . . . . -.-) şeklinde vuruş ve duraklarda bulunmanız gerekiyor. Diğer komutlar da bu tip

vuruş ve duraklardan oluşuyor. Ayrıca köpeğinizin çok kuvvetli bir belleği var ve çok uzun vuruş ve durak dizilerini hafızasında tutabiliyor. Şimdi mutfaktan terliklerinizi köpeğinize getirtmek istediğinizi varsayalım. Mutfağa gidinceye kadar ki tüm adımları ve terliği alabilmesi için başını aşağıya indirip terliği ısırmasını ve tekrar geri dönüp önünüzde bırakmasını sağlamak için, sizce çok karmaşık ve çok uzun olan sırtına vuruşlarla bu emirleri köpeğe vermeniz gerekiyor. Halbuki bu uzun ve sizce karmaşık anlatım yerine “mutfağa git ve terliklerimi getir” demeniz ne kadar da güzel olurdu. Köpeğinizin becerileri, arabaya atlayıp işyerinize gidip CD kutunuzu alıp getirmeye bile müsait fakat sizin için gayet kolay olan bu iş için köpeğinize, sizce çok karmaşık bir iş olan, saatlerce emir vermeniz gerekecektir. Hem de sırtına vurarak..!

Fakat siz yılmıyorsunuz ve köpeğinizle daha iyi işler yapabilmek için düşünüyorsunuz. Aklınıza parlak bir fikir geliyor; Köpekle aranıza anlaşması daha kolay bir şey/katman koymak! Ve 5 yaşındaki çocuğunuza köpeğin anladığı bu komutları öğretiyorsunuz. Artık karşınızda sizinle aynı seviyede olmasa da daha zeki bir muhatap var. Siz çocuğunuzun anlayacağı seviyede emirlerle çocuğunuzu yönlendiriyorsunuz ve çocuğunuz da köpeğe emirler veriyor.

İşte bu örnekte köpeğin sadece sırtına vuruş ve durakları anlayabilmesi gibi, elektronik devrelerden oluşan bilgisayar da sadece bir devredeki voltaj seviyesinin belli bir değer altında (ki buna devrede elektrik yok diyoruz. Ve 0 değeri ile ifade ediyoruz) veya üstünde (ki buna da devrede elektrik var diyoruz. Ve 1 değeri ile ifade ediyoruz) olması durumunu fark edebilir. Eğer devredeki voltaj değeri yerine 1 ve 0 değerlerini düşünürsek bu devrelerde 1 veya 0 değerlerinin saklandığını söyleyebiliriz. **Bit**, içerisinde 1 veya 0 değerini tutabilen en küçük depolama birimidir. RAM bellekte bu devrelerden milyonlarcası bulunur. Köpeğe bu vuruş ve duraklar çok basit komutları ifade edebileceği gibi, bit dizisi de bilgisayara çok basit komutları ifade edecektir. Bu komutlara **makine komutları** diyoruz. CPU'nun görevi bu komutları icra etmektir. Bit dizileri bir veriyi de ifade edebilirler. İlk bilgisayarlar ortaya çıktığında, bilgisayar donanımının anladığı tek dil olan makine dili ile bilgisayara iş yaptırılmaya çalışılmıştır. Fakat bilgisayarın bu yapısı bizim için çok karmaşıktır. Bu yüzden araya, çalışması daha kolay katmanlar koyulmuştur. Bunlardan en önemlisi işletim sistemi katmanıdır. En yukarıda uygulama programları doğrudan bizimle ilişki kurarak, bilgisayarın bizim istediğimiz işleri yapmasını sağlarlar.

Bu kitap boyunca yukarıda sayılan katmanları kitabın bölümleri halinde inceleyeceğiz. Bu katmanlara ek olarak, bilgisayar ağları da bir bölüm olarak incelenecek.

## Donanım

Donanımdan veri-donanım-yazılım modelinde bahsettik. Bu yüzden burada ayrıca bahsetmeyeceğiz.

Bu katmanla ilgili, kitapta 3 bölüm işlendi.

1. *Matematik ve Elektronik temeller* bölümünde, bilgisayarın dili olan ikili sayı sistemini ve CPU, bellek gibi bileşenlerin temel yapı taşları olan mantık kapılarını göreceğiz.

2. *Gerçek dünya – sanal dünya* bölümünde, gerçek dünyadaki metin, ses, resim ve sayı gibi verilerin bilgisayar ortamında nasıl ifade edildiğini öğreneceğiz.
3. *Donanım* bölümünde, bilgisayarı oluşturan bileşenleri ayrıntılı olarak inceleyeceğiz. En son kısımda performans değerlendirme testleri konusunda bilgileneceğiz.

## İşletim Sistemi

**İşletim sistemi** uygulamalara iki konuda hizmet eder.

- Bütün uygulamaların kullandığı kaynakları, yani alt katmanda bulunan donanım cihazlarını yönetmek.
- Yukarı katmanda bulunan uygulama programları için, donanımın karmaşıklığını gizleyerek, çalışması daha kolay bir platform sunmak.

*Donanım cihazlarını yönetme* görevini, trafik polisliğine benzetebiliriz. Trafik polisinin kavşakta durup, çok sayıda aracın sınırlı olan yolları verimli kullanmasını sağlaması gibi, işletim sistemi de, aynı anda çalışan pek çok programın, disk, bellek ve CPU gibi donanım kaynaklarını verimli kullanmasını sağlar. Mesela Windows ve Linux üzerinde aynı anda birçok uygulama birlikte çalışır. Bunların CPU'yu hangi sırada ve ne kadar süre kullanacakları işletim sistemi tarafından belirlenir. Ayrıca bu uygulamalar bellekte birlikte buldukları için birbirlerinin alanlarına müdahale etmelerinin de önlenmesi gerekecektir. Yine, yazıcı gibi bir çevre cihazını aynı anda birden çok uygulama kullanmak istediğinde, karışık bir çıktının oluşmasını önlemek için, işletim sistemi bu yazdırma isteklerinin bir sıra dahilinde yazıcıyı gönderilmesini sağlar.

*Daha kolay programlanabilir platform sunma* görevi, soyutlama tekniğinin bir kullanımudur. Doğrudan makine komutlarıyla bilgisayarı programlamamız, özellikle çevre cihazlarıyla çalışmamız oldukça zordur. Mesela, disket sürücünden bir dosya okuyacağımızı farz edelim. Bunun için uygulamanın disket sürücü denetleyici ile konuşabilecek tüm yeteneklere sahip olması gerektirir. Bunlar, disket sürücü motorunu açıp kapama, dosyanın disket üzerindeki konumunu belirleme, okuma işlemi için gerekli olan komutu disket sürücüye gönderme ve disket sürücünün cevabını anlayabilmek gibi karmaşık işlemleri içeriyor. Buna sabitdisk ve CD sürücü gibi ortamları da ekleyelim. Programlama işinin ne kadar zor olduğunu anlayabiliriz.

İşletim sistemi bizi bu karmaşıklıktan izale eder ve çalışmak için fonksiyonlar/servisler sunar. Mesela, Windows işletim sistemi üzerinde uygulama geliştiren bir programcı, işletim sistemine, dosya yol tanımını (dosya adı, dizini ve sürücü harfi), okunacak veri miktarını ve okunan verinin bellekte nereye koyulacağını parametre olarak verip okuma isteği yapabilir. Bu sayede disk sürücü, disket sürücü veya başka bir depolama ortamının özelliklerini bilmeden de program yazabilir.

İşletim sisteminin pek çok fonksiyon ve prosedürden oluşan bu hizmet setine, **Uygulama Programlama Arabirimi (Application Programming Interface -API)** diyoruz. Bu hizmet setinin içinde neler var diye bakarsak; Diskten dosya okuma, ekrana yazı yazma, klavyeden girilen

bir tuşu okuma, yazıcıya veri gönderme gibi, donanım cihaz tipi ve bunlarla yapılabilecek işlerin çeşitliliği kadar çok fonksiyonu görebiliriz. Programların işletim sisteminden yaptıkları isteklere, Windows dünyasında **API çağrısı (API call)** veya UNIX dünyasında **Sistem çağrısı (System call)** diyoruz. Her işletim sisteminin API'si farklıdır. Bu yüzden bir işletim sisteminde çalışan uygulama, özel yazılımlar ve teknikler kullanılmadığı müddetçe, diğerinde çalışmayacaktır. API çağrılma şekilleri de değişiklik gösterebilir: DOS API kesmeler şeklinde kullanılırken, Windows API fonksiyon çağrıları şeklindedir. Programlar isteklerini bu çağrılar vasıtasıyla işletim sistemine iletirler. İşletim sistemi bu isteği donanım cihazlarının anlayabileceği çok sayıda emirlere bölerek işin yapılmasını sağlar.

Bu katmanla ilgili, kitapta *işletim sistemleri* bölümü işlenmiştir.

Bu bölümde, işletim sisteminin görevlerini daha ayrıntılı olarak göreceğiz. Her görevi, Linux ve Windows XP işletim sistemleri üzerinde uygulamalı olarak inceleyeceğiz. Kurumlarda, işletim sistemleri yönetimi konusunda çalışan bilgisayar mühendisleri, **sistem yöneticisi** unvanını alırlar. Bu yüzden bu bölümde bu görevi de ana hatlarıyla inceleyeceğiz.

## Uygulama

Bizler kullanıcı olarak, her zaman uygulama katmanında çalışan programları kullanırız. Oynadığımız oyunlar, kelime işlemciler (Microsoft Office Word, OpenOffice writer), veritabanı yazılımları, İnternet tarayıcıları, film oynatıcı programlar, veri iletişim programları ve dil eğitim programları bu kategoriye girer. *Altındaki katmanlar sadece uygulama programlarının çalışması için hizmet verirler.*

Bir **uygulama** veya **program**, bilgisayarın beyni olarak niteleyebileceğimiz CPU'nun neler yapması gerektiğini söyleyen komutlar topluluğudur. Programı oluşturan bu komutlara makine komutu diyoruz. Programın çalışması, bu komutların bellekten alınıp CPU tarafından işletilmesi demektir.

Uygulamalar, insanların ihtiyaçları kadar çeşitlilik gösterir. Binlerce kullanıcıya hizmet veren bankacılık sistemlerinden kişisel ofis yazılımlarına ve çocuklar için oyunlardan NASA'da uzay mekiklerinin rotasını tayin eden uygulamalara kadar geniş bir aralık vardır. Bunların hepsini elbette incelememiz mümkün değildir. Uygulamalar gerçek dünya problemlerini çözmek için geliştirilirler. Bazı problemler ve çözümleri, bilgisayar biliminde bir alan olacak kadar gelişme göstermiştir. Örnek olarak simülasyon, bilgisayar grafikleri, yapay zeka, bilgi sistemleri ve veritabanlarını verebiliriz.

Bu katmanla ilgili, kitapta *uygulama programları* bölümü işlenmiştir.



## Profesyonelliğe Doğru

Mesleki noktada daha iyi konuma gelebilmek için nasıl bir yol izlemeliyiz?

- Öncelikle bilgisayar temel konuları hakkında genel fikir sahibi olmalıyız. Donanım, programlama, işletim sistemi, ağ, veritabanı, güvenlik konularını örnek olarak verebiliriz. Elinizdeki kitap bu amaç için iyi bir seçimdir. Pratik ağırlıklı CompTIA sertifikasyonu için hazırlanan kitaplar veya teori ağırlıklı bilgisayar bilimine giriş kitapları bu konuda faydalıdır.
- İngilizce olması gereken şartlardandır. Çünkü İnternet ortamı için neredeyse resmi dil İngilizcedir. İngilizce bilmemek tonlarca bilgiden, İnternet haber listelerinden mahrum kalmak demektir.
- Bu seviyeden sonra uzmanlaşmak istediğiniz konuyu seçip üzerinde çalışmanız gerekiyor. Seçimde sevdiğiniz bir konu olması ilk esastır. Bundan sonra piyasada aranılan bir konu olmasına özen gösterin. Uzmanlaşmak için ilk iş, konu hakkında yazılmış çok satan (İngilizce!) birkaç kitabı bulup okumaktır. Bunun için [www.amazon.com](http://www.amazon.com) adresindeki çok satanlar listesinden ve kitap eleştirilerinden faydalanabilirsiniz. Bunları kağıt ortamda satın alın. Böylece kitapları karalayarak verimli çalışabilirsiniz.
- Konunuzla ilgili bir haber grubuna üye olun. [groups.google.com](http://groups.google.com) veya [groups.yahoo.com](http://groups.yahoo.com) gibi adreslerden böyle grupları bulabilirsiniz. Böylece o konuda karşılaşılan problemleri ve çözümlerini yakından takip edebilirsiniz.
- Kurumlar için ISO 9000, ürünler için CE (Avrupa Normlarına Uygunluk) artık istenir hale geldiği gibi, çalışanlar için de sertifikasyon gittikçe daha istenir bir hale gelecektir. Bu daha kaliteli işgücü için gereklidir<sup>2</sup>. Eğer konunuzla ilgili bir sertifikasyon varsa, onu almaya çalışın. Mesela
  - Teknikerlik/teknisyenlik: CompTIA A+, server+
  - Windows sistem yöneticiliği: Microsoft MCSE (Microsoft Certified System Engineer)
  - Ağ: CCNA (Cisco Certified Network Associate)
  - Veritabanı yöneticiliği: OCA (Oracle Certified Associate)
  - Programlama: SCJP (Sun Certified Programmer for the Java 2 Platform), MCSD (Microsoft Certified Solution Developer)
  - Güvenlik: CompTIA Security+
  - Linux: RHCE (RedHat Certified Engineer), LPI (Linux Professional Institute)

Bu bölümde incelenen alanlar bilgi sistemleri ve yapay zekadır.

<sup>2</sup> Burada güngörmüş tecrübeli fakat sertifikasız bilgi teknolojileri çalışanlarını düşününce, “Sertifika gümüşse tecrübe altındır” sözünü belirtmeden geçemeyeceğim.

**Bilgi Sistemleri (Information Systems)** bir kurum içerisinde, verinin toplanması, depolanması, işlenmesi ve bilgi olarak alıcılara dağıtılması amacıyla kurulmuştur. Bir kurumda değişik seviye yöneticiler ve veri giriş elemanları bulunabilir ve her grubun bilgi ihtiyacı farklıdır. Mesela veri giriş elemanı bir müşterinin borcuyla ilgilenirken orta seviye yönetici günlük toplam tahsilatla ilgilenebilir. En tepedeki yönetici ise kurumun yıllık karının geçmiş yıllara göre nasıl değiştiğini görmek isteyebilir. İşte bilgi sistemleri, her seviye kurum personelinin ihtiyaç duyduğu bilgiye ulaşılabilmesini amaçlar.

**Yapay Zeka (Artificial Intelligence)**, insanın bilgisayardan üstün olduğu noktalarda bilgisayar kabiliyetlerini iyileştirmeye çalışır. Bilgisayarlar hız ve doğruluk isteyen matematiksel hesaplama temelli işlemlerde çok başarılı olurlarken, muhakeme etme, görülen cisimleri ayırt etme, konuşulanları anlama gibi alanlarda oldukça geridirler. Yapay zekanın çalışma alanları da insana ait bu zeki özelliklerdir. Yani insana ait zeki özelliklerin bilgisayara kazandırılmasını amaçlar.

## Kapasite ve Hız Birimleri

Kitap boyunca hız ve kapasite bildirimleri sırasında sık sık karşımıza çıkacak olan birimlere bakmamız faydalı olacaktır

**Bit**, Binary digIT kelimelerinin kısaltmasıdır ve “**ikili sayma sistemindeki rakam**” anlamına gelir. Bilgisayarda en küçük depolama birimidir. 1 Bit, üzerinde 2 durumdan birini (1 veya 0) saklayabilen depolama hücresidir.

**1 Byte: 8 Bit.**

Birim	Üslü ifade	Açık ifade (Byte)	Yaklaşık Byte olarak ifade
KiloByte (KB)	$2^{10}$ Byte	1024	~ Bin Byte
MegaByte(MB)	$2^{20}$ Byte	1,048,576	~ 1 Milyon Byte
GigaByte(GB)	$2^{30}$ Byte	1,073,741,824	~ 1 Milyar Byte
TeraByte(TB)	$2^{40}$ Byte	1,099,511,627,776	~ 1 Trilyon Byte
PetaByte(PB)	$2^{50}$ Byte	1,125,899,906,842,624	~ 1 Katrilyon Byte
ExaByte(EB)	$2^{60}$ Byte	1,152,921,504,606,846,976	~ 1 Kentilyon Byte

Gündelik hayatta metrik sistemi, bin ve milyon şeklinde artan bir sistem olarak kullanıyoruz. Mesela Kilogram, bin gramdır. Fakat bilgisayarlar onlu sistem yerine ikili sistemi kullandıkları için 1 KiloByte,  $10^3$  Byte yerine  $2^{10}$  Byte anlamına gelir.

Günümüz sabitdiskleri üzerinde 100 GB veri normal bir kapasite haline gelmiştir. RAM bellekler 256 MB ve daha üstü kapasitelerde olabilirler. Disket 1.44 MB kapasitededir.