

# Bilgisayar Programlama I

## Bilgisayar Bilimine Giriş

## Bilgisayar Tanımı ve Yapısı

Genel anlamı ile bilgisayar, verileri okuyabilen, bunlar üzerinde önceden tanımlanmış işlemleri gerçekleştiren ve sonucunda da istenen bir çıktıyı üreten elektronik bir cihazdır. Aynı zamanda bir bilgisayar, uygulama programları vasıtasyyla girilen verileri çok hızlı ve hatasız bir şekilde işleyerek bilgiye dönüştürme yeteneğine sahip bir makinedir. Bilgisayarlar kullanım amaçlarına göre farklı tür ve şekillerde (masaüstü, dizüstü, iş istasyonu, sunucu gibi) olabilmektedir. Ancak bir bilgisayar temel yapı itibarıyle donanım (hardware) ve yazılım (software) olmak üzere iki kısımdan oluşmaktadır.



Masaüstü (Desktop)

Dizüstü (Notebook / Laptop)

Bir bilgisayarın bütün fiziksel parçaları donanım olarak adlandırılır. Fakat bir bilgisayarın temel işlevlerini yerine getirmesi ve kullanıcının ihtiyaçlarını karşılaması bakımından donanım kısmı genel olarak dört ana bölüme ayrılabilir:

### 1. Merkezi İşlem Birimi (Central Processing Unit - CPU),

Bilgisayarın "beyni" olarak da bilinir. Tüm hesaplama ve işlem gücünü sağlar. İşlemci hızını ve performansını belirleyen çekirdek ve saat hızları gibi bileşenleri içerir.

### 2. Bellek-hafıza (Random Access Memory - RAM),

Geçici veri depolama birimidir. Bilgisayarın aktif olarak kullandığı verileri ve işlemleri saklar. Veri işlemlerinin hızlı gerçekleştirilmesini sağlar.

### 3. Depolama Birimleri,

- **Sabit Disk Sürücüsü (Hard Disk Drive - HDD):** Manyetik diskler üzerinde veri depolar. Daha büyük depolama kapasitesine sahiptir.
- **Katı Hal Sürücüsü (Solid State Drive - SSD):** Daha hızlı veri erişimi ve okuma/yazma hızları sağlar. Daha dayanıklı ve sessizdir.

### 4. Giriş/Çıkış Birimleri (Input/Output Devices),

- **Giriş Birimleri:** Kullanıcıdan veri girişi sağlar. Klavye, fare, mikrofon, tarayıcı gibi.
- **Çıkış Birimleri:** İşlenmiş verileri kullanıcıya sunar. Monitör, yazıcı, hoparlör gibi.

Bilgisayarın yazılım kısmı, donanımı çalıştırın ve kullanıcıların belirli görevleri gerçekleştirmesine olanak tanıyan programlar ve uygulamalardan oluşur. Yazılım, donanımın işlevlerini yönlendiren talimatlar bütündür ve iki ana kategoriye ayrıılır:

#### \* Sistem Yazılımı

- **İşletim Sistemi (Operation System - OS):** Bilgisayarın temel işlevlerini yönetir. Donanım kaynaklarını (CPU, bellek, depolama) kontrol eder ve uygulamalar arasında bu kaynakların dağıtımını sağlar. Kullanıcı ile donanım arasındaki etkileşimi yönetir. Windows, macOS, Linux, Android gibi.
- **Sistem Yardımcı Programları:** Bilgisayarın bakımını ve performansını iyileştirmek için kullanılır. Disk temizleme, virüs tarama, sistem geri yükleme gibi görevleri gerçekleştirir.

#### \* Uygulama Yazılımı

- **Ofis Yazılımları:** Belge oluşturma, hesap tabloları, sunumlar ve e-posta gibi günlük iş görevlerini gerçekleştirir. Microsoft Office, Google Workspace benzeri uygulamalar.
- **Medya ve Eğlence Yazılımları:** Müzik dinleme, video izleme, fotoğraf düzenleme ve oyun oynama gibi eğlence amaçlı kullanılır. VLC Media Player, Adobe Photoshop, Steam gibi.
- **Internet ve İletişim Yazılımları:** Web tarayıcıları ve e-posta istemcileri gibi internet üzerinde gezinme ve iletişim kurma araçlarını içerir. Google Chrome, Mozilla Firefox, Microsoft Outlook gibi.
- **Eğitim ve Öğretim Yazılımları:** Öğrenme ve öğretme süreçlerini destekleyen yazılımlar. Duolingo, Khan Academy, Coursera ve Udemy gibi.

Bilgisayar yazılımı, kullanıcıların ihtiyaçlarını karşılamak ve donanımın tam potansiyelini kullanabilmek için hayatı öneme sahiptir. Yazılım olmadan, bilgisayar donanımı işlevsel olmaz ve kullanıcılar belirli görevleri gerçekleştiremez. Bu bakımdan yazılımlarda olması gereken özellikleri genel olarak şu şekilde özetleyebiliriz:

- **Kullanıcı Dostu Arayüz:** Kolay ve sezgisel kullanımı sağlayan grafiksel kullanıcı arayüzleri (GUI).
- **Uyumluluk:** Farklı donanım ve diğer yazılım bileşenleriyle uyumlu çalışma yeteneği.
- **Güncellenebilirlik:** Yazılımın güvenlik yamaları, hata düzeltmeleri ve yeni özellikler eklemek için güncellenebilmesi.
- **Güvenlik:** Kullanıcı verilerini koruma ve yetkisiz erişimi engelleme yetenekleri.
- **Performans:** Verimli çalışma ve hızlı işlem yapma kapasiteleri.
- **Esneklik:** Kullanıcı ihtiyaçlarına göre özelleştirilebilir ve genişletilebilir olma.

## Bilgisayar Donanım

Bir bilgisayarın kapasitesini, kabiliyetini ve gücünü veren onun donanımıdır. Bu bakımından kullanım alanına ve amacına göre çok farklı donanımlar mevcuttur. Kişisel amaçlı veya işyerinde kullanılan bilgisayarlarda farklı donanımlara ihtiyaç duyulurken uluslararası büyük kurumsal şirketlerde ise çok farklı donanımlara sahip bilgisayarların kullanılması söz konusudur. Özel amaçlar ve ihtiyaçlar için kullanılan donanımlarında haricinde kişisel ve iş bilgisayarlarında kullanılan donanımlar en genel haliyle şu şekilde gruplandırılabilir.

1. Merkezi İşlem Birimi (CPU)
2. Bellek (RAM)
3. Anakart
4. Ekran Kartı (GPU)
5. Depolama Birimleri
6. Giriş/Çıkış Aygıtları
7. Güç Kaynağı Ünitesi (PSU)
8. Soğutma Sistemleri

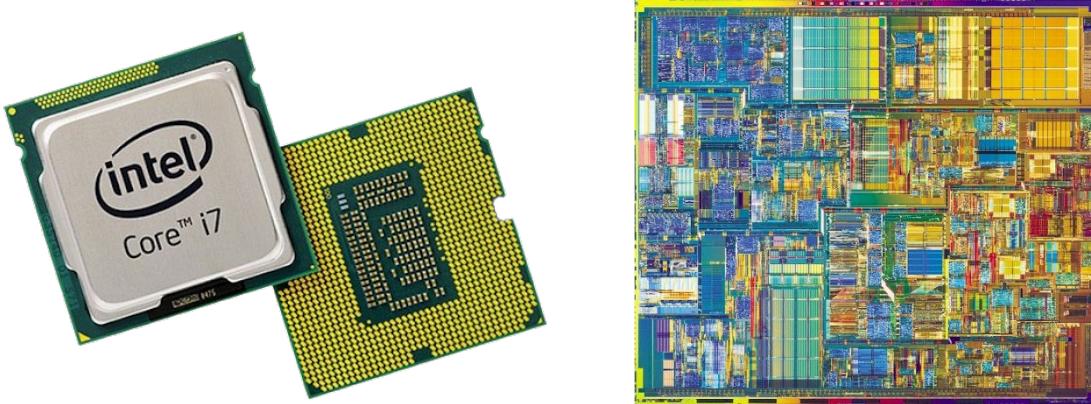
Bütün bu donanımların bir arada olduğu bir masaüstü bilgisayarı kasasının iç kısmı örnek olarak aşağıda gösterilmektedir. Bu donanım grupları bir bilgisayarın çalışmasını ve performansını belirleyen temel bileşenlerdir. Bu donanımların birbirine uyumlu olması, bilgisayarın stabilitesi ve performansı açısından kritiktir. Örneğin, anakartın CPU ve RAM ile uyumlu olması, bu bileşenlerin tam kapasiteyle çalışmasını sağlar. Ancak, uyumsuz donanımlar, sistemin çalışmaması, performans sorunları ve potansiyel donanım hasarları gibi ciddi problemlere yol açabilir. Bu nedenle, bilgisayar bileşenleri seçilirken birbirleriyle uyumluluk durumlarının kontrol edilmesi gereklidir.



Bir masaüstü bilgisayarı kasasının genel görünümü

## **Merkezi İşlem Birimi (CPU)**

Merkezi İşlem Birimi, ya da CPU (Central Processing Unit), bilgisayarın kalbi ya da beyni olarak kabul edilir. Tüm işlemlerin ve komutların verimli bir şekilde gerçekleştirilmesini sağlar. CPU, aritmetik ve mantıksal işlemleri gerçekleştirir, komutları yürütür ve diğer bileşenlerle veri alışverişini yönetir. Modern CPU'lar genellikle birden fazla çekirdeğe sahiptir, bu da aynı anda birden çok işlemi gerçekleştirebilmelerini sağlar. Bir CPU'nun genel görünümü ve iç mimarisi alttaki resim gösterilmiştir.



**Bir CPU'nun genel görünümü**

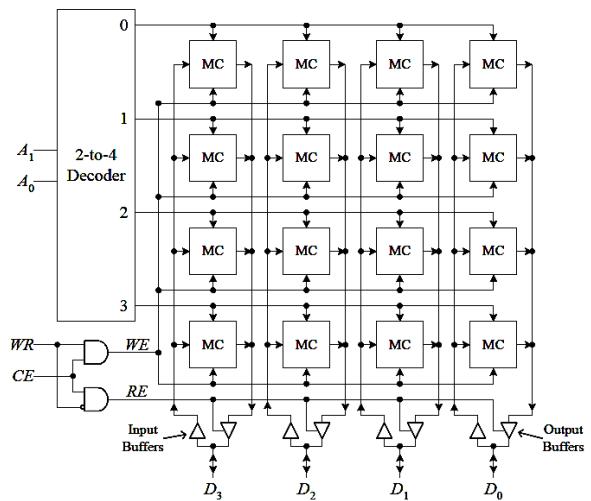
1971 yılında Intel'in ürettiği ilk işlemci 4 bitlik bir veri yoluna, 740 kHz'lık saat hızına ve 2.300 transistör içeriyordu. 1974 yılında üretilen Intel 8080 işlemcisi 8 bitlik veri yoluna ve 2 MHz saat hızına sahipti. 1980 yılında 16 bit, 1990'larda 32 bitlik işlemciler üretildi. 2000'lerden itibaren 64 bit mimari ve çok çekirdekli (dual-core, quad-core) işlemciler yaygınlaşmıştır. Günümüzde, modern CPU'lar milyarlarca transistör içerir, gigahertz (GHz) hızında çalışır ve gelişmiş çoklu çekirdek yapılarına sahiptir.

CPU'lar bir bilgisayarın genel performansını ve verimliliğini belirleyen en önemli bileşenlerden biri olması nedeniyle bazı özellikleri ön plana çıkmaktadır. Bu özellikler şu şekilde özetlenebilir:

- Saat Hızı (Clock Speed):** CPU'nun saniyede kaç işlem yapabileceğini belirler. Daha yüksek saat hızı, daha hızlı işlem anlamına gelir. Ölçüm Birimi Gigahertz (GHz) olarak verilir.
- Çekirdek Sayısı (Number of Cores):** Modern CPU'lar genellikle birden fazla çekirdeğe sahiptir. Her çekirdek bağımsız olarak işlem yapabilir, bu da aynı anda birden fazla görevin daha verimli bir şekilde gerçekleştirilmesini sağlar. Dual-core (2 çekirdek), quad-core (4 çekirdek), hexa-core (6 çekirdek), octa-core (8 çekirdek) gibi adlandırılır.
- İş Parçacıkları (Threads):** Çekirdeklerin aynı anda yürütüleceği bağımsız işlem parçacıklarıdır. Hyper-Threading gibi teknolojilerle her çekirdek birden fazla iş parçacığını işleyebilir.
- Önbellek (Cache):** CPU'nun hızlıca erişmesi gereken verilere kısa süreli erişim sağlamak için kullanılan bellek türüdür. L1, L2 ve L3 olarak sınıflandırılan farklı seviyeleri vardır. Daha büyük önbellek, daha hızlı işlem performansı sağlar.

## Bellek (RAM)

Rastgele Erişim Belleği (RAM), geçici veri depolama birimidir. Bu nedenle bilgisayar kapatıldığında içindeki veriler silinir. Bilgisayarın kısa süreli belleği olarak çalışır ve işlemci tarafından hızlıca erişilmesi gereken verileri depolar. RAM, bilgisayarın performansını doğrudan etkiler; daha fazla RAM, aynı anda daha fazla programın sorunsuz çalışmasını sağlar. İç yapısı, milyonlarca küçük kapasitör ve transistörden oluşur. Bu kapasitörler, elektrik yükünü tutarak 0 ve 1 şeklinde ikili (binary) verileri temsil eder. Transistörler ise kapasitörlerin durumunu okuma ve yazma işlemlerini kontrol eder. RAM, hücre dizileri şeklinde düzenlenmiş bu kapasitör ve transistörlerden oluşan matris yapısıyla, belirli bir adresdeki veriye doğrudan erişim sağlar. Bazı RAM bellek kartları ve iç mimarisi alttaki şekilde gösterilmiştir.



Çeşitli RAM bellekleri ve bir belleğinin iç mimarisi

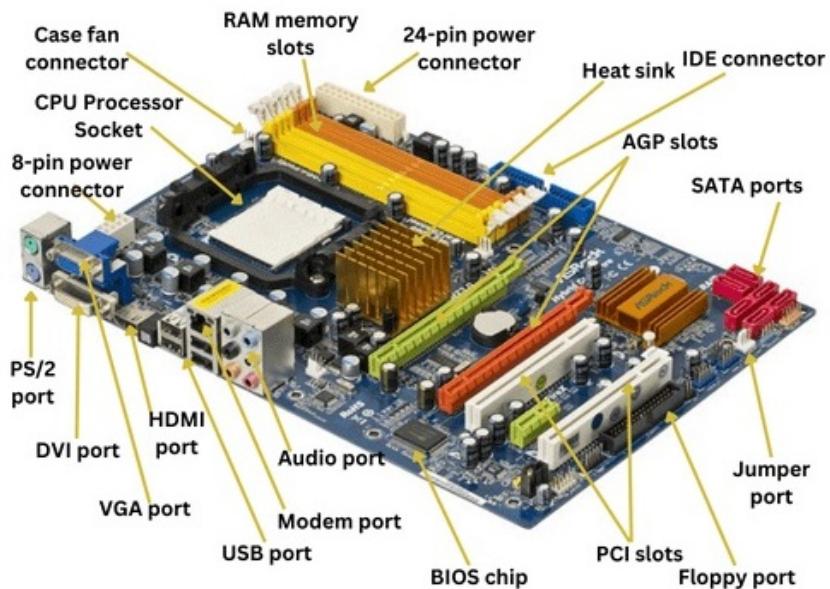
Bir RAM belleğinin performans hızına (MHz veya veri transfer hızları), kapasitesine (GB cinsinden) ve gecikme süresine (daha düşük gecikme, daha hızlı performans) bağlıdır. Yüksek kapasite, daha fazla veri ve uygulamanın aynı anda işlenmesine olanak tanırken, daha yüksek hız ve düşük gecikme süresi daha hızlı veri erişimi ve sistem performansı anlamına gelmektedir. 1970'lerde birkaç kHz ve 1KB ile başlayan hız ve kapasiteleri 2000'li yıllarda 100 MHz'e ve 1 GB'a günümüzde ise 4800 MHz'in ve 128 GB'ın üstüne çıkmıştır.

Çeşitli sistemleri için üretilmiş farklı RAM türleri bulunmaktadır. Özetlemek gerekirse;

- DDR (Double Data Rate), DDR2, DDR3, DDR4, DDR5:** Her bir nesil, önceki nesle göre daha yüksek hız ve verimlilik sunar. DDR4 ve DDR5 şu anda en yaygın kullanılan türlerdir.
- SDRAM (Synchronous Dynamic RAM):** CPU saat hızına senkronize olarak çalışan RAM türüdür. DDR RAM de bir tür SDRAM sayılır.
- DRAM (Dynamic RAM):** Elektrik yüklerini sürekli yenilemek zorunda olan ve dolayısıyla sürekli enerji tüketen RAM türüdür. Modern bilgisayarlarda yaygın olarak kullanılır.
- SRAM (Static RAM):** Daha hızlı ve enerji verimli ancak daha pahalı olan RAM türüdür. Genellikle CPU önbelleği gibi küçük ve hızlı bellek alanlarında kullanılır.

## Anakart

Anakart, tüm donanım bileşenlerinin bağlandığı merkezi devre kartıdır. Bir bilgisayarın omurgası olarak tüm bileşenlerin iletişim kurmasını ve uyumlu çalışmasını sağlar. CPU, RAM, GPU ve diğer bileşenlerin uyumlu çalışmasını garanti eder, bu da sistemin performansını doğrudan etkiler. Anakartın sunduğu genişleme yuvaları ve bağlantı noktaları, gelecekteki yükseltme ve genişletmeler için olanak tanır, böylece sistemin ömrü ve işlevselligi artırılır. Ayrıca, modern anakartlar gelişmiş teknolojileri destekleyerek daha hızlı veri transferi, güvenilir bağlantılar ve genel olarak üstün bir kullanıcı deneyimi sunar. Bu nedenle, anakart seçimi, bilgisayarın kararlılığı, performansı ve genişletilebilirliği açısından kritik bir rol oynar. Örnek bir anakart ve bileşenleri alttaki resimde gösterilmiştir.



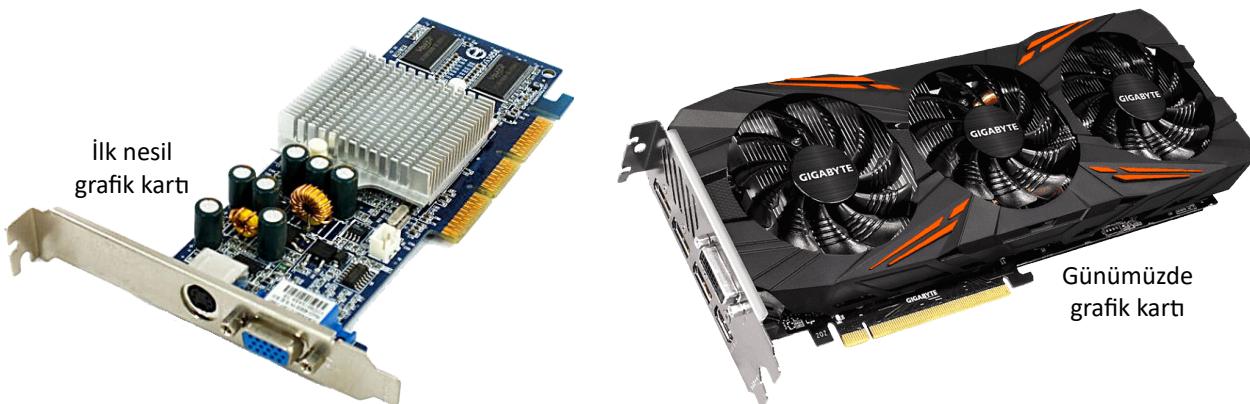
Bir anakartın genel mimarisi

Bir anakartın genel olarak belirli özelliklere sahiptir. Bu özellikleri arasında CPU'nun bağlandığı soket tipi, veri yollarını ve bileşenleri yöneten yonga seti, RAM modüllerinin takıldığı bellek yuvaları, ek bileşenler için genişleme yuvaları (PCIe), depolama cihazlarını bağlamak için SATA ve NVMe bağlantı noktaları, dış bağlantılar için I/O portları (USB, HDMI, Ethernet gibi) ve güç bağlantıları (24-pin ATX, CPU güç konektörü) yer alır.

Anakartlar kullanım alanlarına göre farklılık göstermektedir. **ATX (Advanced Technology eXtended)** türü anakartlar, geniş genişleme yuvaları ve tam boyutyla masaüstü bilgisayarlar için yaygın olarak kullanılır. **Micro-ATX** anakartlar ise ATX'e göre daha kompakt bir tasarıma sahip olup, daha az genişleme yuvası sunar ve kompakt masaüstü bilgisayarlarda tercih edilir. **Mini-ITX** anakartlar en küçük form faktörlerden biri olup genellikle tek bir genişleme yuvasına sahiptir, bu da onu küçük form faktörlü bilgisayarlar için ideal kılar. **E-ATX (Extended ATX)** anakartlar ise, ATX'in genişletilmiş versiyonudur ve daha fazla genişleme yuvası ile yüksek performanslı iş istasyonları ve oyun bilgisayarları için uygundur. Bu çeşitler, farklı kullanıcı ihtiyaçlarına ve sistem gereksinimlerine göre özelleştirilmiş çözümler sunar.

## Ekran/Grafik Kartı (GPU)

Grafik İşlemci Birimi (GPU), görüntü işlemleri ve grafiklerin işlenmesinden sorumlu olan bileşendir. Yüksek performanslı ekran kartları, oyunlar, video düzenleme ve grafik tasarımları gibi grafik yoğun işlemleri hızlı ve akıcı bir şekilde gerçekleştirir. Ayrıca, yüksek çözünürlük ve detaylı görüntüler sunarak görsel deneyimi artırır. Modern ekran kartları, birden fazla monitör desteği sunar ve gelişmiş grafik teknolojileri (örneğin, ray tracing, AI tabanlı işlemciler) ile daha gerçekçi ve etkileyici görseller sağlar. Bu özellikler, kullanıcıların verimliliğini artırır ve görsel deneyimlerini iyileştirir. Alttaki resimde ilk nesil bir grafik kartı ile günümüzdeki son nesil bir grafik kartı örnek olarak gösterilmiştir.



## Ekran/Grafik kart örnekleri

Ekran kartının temel özellikleri, GPU (grafik işlemci birimi) ve VRAM (video RAM) ile belirlenir; GPU, grafik işleme gücünü sağlarken, VRAM görsel verileri depolar ve hızlı erişim sağlar. Ekran kartları genellikle soğutma sistemleri içerir, bu da GPU'nun aşırı ısınmasını önler ve performansını korur. Çıkış bağlantıları (HDMI, DisplayPort, DVI) ise monitörlere yüksek kaliteli görüntü iletimi sağlar. GPU'nun hızı (GHz cinsinden) ve çekirdek sayısı, grafik işleme kapasitesini etkiler; daha yüksek hız ve çekirdek sayısı, daha iyi performans anlamına gelir. Bu özellikler, ekran kartının genel performansını, görüntü kalitesini ve çoklu monitör desteğini belirleyerek, oyun, video düzenleme ve grafik tasarımları gibi uygulamalarda kritik bir rol oynar.

Ekran kartı çeşitleri, farklı kullanıcı ihtiyaçlarına ve sistem gereksinimlerine göre değişim göstermektedir. Genel olarak üç farklı grafik kartından bahsetmek mümkündür.

- Gömülü/Entegre Ekran Kartları:** CPU veya anakart üzerinde entegre olarak bulunan grafik işlemcileridir. Temel grafik görevlerini yerine getirir ve harici ekran kartlarına göre daha düşük performans sağlar. Örnek olarak Intel UHD Graphics, AMD Radeon Vega kartları verilebilir.
- Ayrı Ekran Kartları (Discrete Graphics Cards):** Kendi bağımsız GPU'su ve VRAM'i olan, genellikle PCIe yuvasına takılan ekran kartlarıdır. Daha yüksek performans sunar ve oyunlar ile grafik yoğun uygulamalar için uygundur. NVIDIA GeForce RTX serisi, AMD Radeon RX serisi gibi.
- Profesyonel Ekran Kartları:** Yüksek performanslı ve özel grafik işleme yeteneklerine sahip ekran kartlarıdır, genellikle CAD, 3D modelleme ve video üretiminde kullanılır. NVIDIA Quadro serisi, AMD Radeon Pro serisi kartları.

## Depolama Birimleri

Depolama birimleri, kullanıcıların verilerini, programlarını ve işletim sistemlerini güvenli bir şekilde saklamalarını sağlar. Yeterli kapasiteye sahip olmaları, bilgisayarın genel kullanımını ve verimliliğini artırırken, hızlı depolama birimleri, sistemin performansını doğrudan etkiler. Bu cihazlar dahili (internal) olarak kullanılabildiği gibi harici (external) cihazlar olarak da kullanılabilmektedir. Harici cihazlar çoğunluklar yedekleme amacıyla ya da verinin taşınması (mobilize) amacıyla kullanılmaktadır.

Verilerin kayıt şekline göre iki çeşit depolama birimi vardır. Manyetik diskler kullanarak veri depolayan geleneksel depolama birimleri olan **Sabit Disk Sürücüler (HDD)** ve flash bellek teknolojisi kullanarak veri depolayan modern depolama birimleri olan **Katı Hal Sürücüler (SSD)**. HDD'ler 500 GB ile 10 TB arasında değişen büyük kapasitelere sahip olmalarına karşın, kapasiteleri 128 GB ile 4 TB arasında değişen SSD'lere kıyasla daha yavaş çalışmaktadır. HDD'lerde okuma/yazma hızı 80-160 MB/s iken SSD'lerde bu 200-550 MB/s olmaktadır. Yeni nesil **NVMe (Non-Volatile Memory Express Express – Dengesiz olmayan (Kararlı) Hızlı Bellek)** SSD'lerde ise bu hızlar 7300-6350 MB/s kadar çıkabilmektedir. Bu depolama birimlerinin genel görünümü alttaki resimde gösterilmiştir.



Depolama birimlerinin genel görünümü

Depolama birimlerinin anakarta bağlantı tipi aynı zaman veri transfer hızının bir göstergesidir. 1980'lerde PATA (Parallel Advanced Technology Attachment) yada yaygın adıyla IDE (Integrated Drive Electronics) ve SCSI (Small Computer System Interface) bağlantı tipleri yaygın iken 2003'lerde SATA (Serial Advanced Technology Attachment) bağlantı tipi yaygınlaşmıştır. Günümüzde NVMe SSD'lerin kullanımıyla birlikte mSATA (mini-SATA), M.2 ve PCIe bağlantıları kullanılmaya başlanmıştır.

Depolama birimleri kullanım amaçlarına ve kullanım yerlerine göre farklı büyülüklerde olmaktadır. HDD'lerde büyülüklük disk plakalarının çapıyla verilmektedir. İlk dönemlerde masaüstü bilgisayarlar ağırlıklı olarak 3.5 inch HDD'ler kullanılırken 2.5 inch HDD'ler dizüstü bilgisayarlarda kullanılmaktaydı. Günümüzde SSD'ler ve NVMe'ler hem masaüstü hem de dizüstü bilgisayarlarda performans açısından daha çok tercih edilmektedir.



## Giriş/Çıkış Aygıtları (Input/Output Devices)

Giriş/Çıkış aygıtları, bilgisayarın kullanıcı ile etkileşimi ve veri alışverişini mümkün kılar. Kullanıcılar, giriş aygıtları sayesinde bilgisayara veri ve komut girerken, çıkış aygıtları sayesinde bilgisayarın ürettiği bilgileri alır. İyi tasarlanmış ve uygun seçilmiş G/C aygıtları, kullanıcı deneyimini iyileştirir ve bilgisayar sisteminin performansını artırır. Giriş-Çıkış aygıtları bilgisayarların nerede, hangi amaç için kullanıldığına göre oldukça çeşitlilik göstermektedir. Ancak genel olarak bu aygıtların bazıları şu şekilde verilebilir.

### Giriş Aygıtları

1. Klavye : Metin yazma, komut verme ve veri girişi için temel aygittır.
2. Fare (Mouse) : Grafik arayüzlerle hızlı ve kolay etkileşim sağlar.
3. Tarayıcı (Scanner) : Belgelerin ve resimlerin dijitalleştirilmesini sağlar.
4. Mikrofon : Sesli komutlar, ses kayıtları ve iletişim için kullanılır.
5. Web Kamerası : Görüntülü iletişim ve yayıcılık için önemlidir.



Klavye



Fare



Tarayıcı



Mikrofon



Kamera

### Çıkış Aygıtları

1. Monitör : Görsel bilgi sunar ve kullanıcıların bilgisayar arayüzü ile etkileşimde bulunmasını sağlar.
2. Yazıcı : Belgelerin ve resimlerin fiziksel çıktısını sağlar.
3. Hoparlörler : Sesli bildirimler, müzik, filmler ve medya içerikleri için kullanılır.
4. Projeksiyon Cihazı : Sunumlar ve büyük grup gösterimleri için kullanılır.



Monitör



Yazıcı



Hoparlör



Projektör

### Hem Giriş Hem Çıkış Aygıtları

Dokunmatik Ekran : Hem giriş hem de çıkış işlevi görür. Kullanıcılar, ekrana dokunarak veri girebilir ve ekranada gösterilen bilgileri görebilir.

Harici Depolama Aygıtları : USB bellekler, harici HDD'ler ve SSD'ler hem veri girişine (yazma) hem de veri çıkışına (okuma) imkân tanır.

## Güç Kaynağı Ünitesi (Power Supply Unit - PSU)

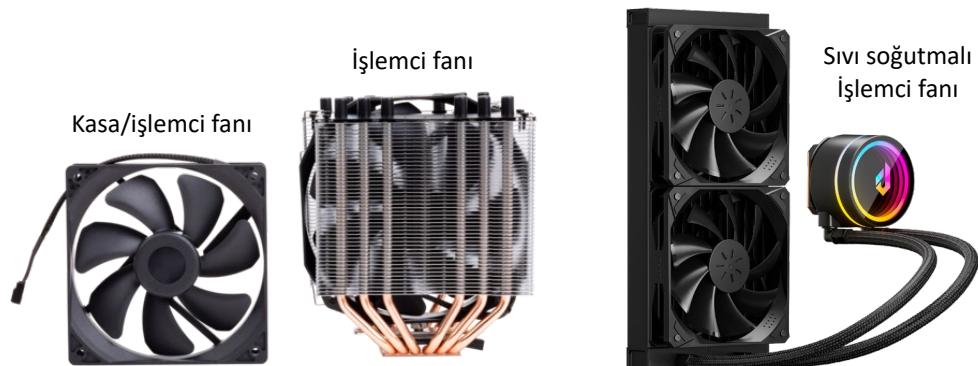
Güç kaynağı ünitesi (PSU), bir bilgisayarın tüm bileşenlerine ihtiyaç duydukları doğru akımı (DC) sağlamak için şehir elektriğini (alternatif akım - AC) dönüştüren cihazdır. PSU, bilgisayarın stabil ve güvenli bir şekilde çalışabilmesi için gerekli voltaj ve akımı sağlar. Güç kaynağı ünitesi, bilgisayar sisteminin güvenilirliği ve performansı için kritik öneme sahiptir. Yetersiz veya kalitesiz bir PSU, sistemde kararsızlığa, performans düşüşüne ve hatta donanım hasarına yol açabilir. Doğru seçilmiş bir PSU, bilgisayarın tüm bileşenlerine yeterli ve kararlı güç sağlar. Örnek güç kaynakları alttaki resimde gösterilmiştir.



Güç kaynaklarının maksimum güç kapasitesi 300W ile 1000W arasında olup, genellikle +3.3V, +5V ve +12V arasında voltaj çıkış hatlarına sahiptirler. Farklı bilgisayarlar için kullanılan ATX, SFX, Modüler ve modüler olmayan çeşitleri bulunmaktadır.

## Soğutma Sistemleri

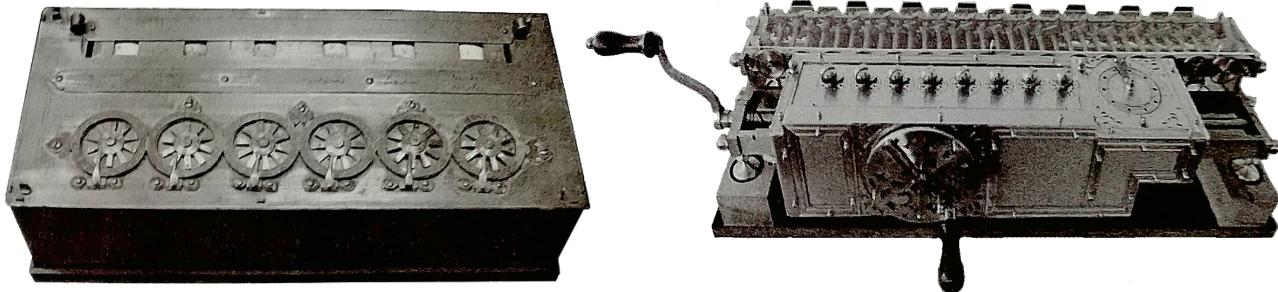
Bilgisayarın bileşenleri, özellikle CPU ve GPU, çalışırken ısı üretir. Bu ısının etkili bir şekilde dağıtılması ve bileşenlerin aşırı ısınmasının önlenmesi için soğutma sistemleri kullanılır. Soğutma sistemleri genellikle fanlar ve ısı emicilerden oluşur, ancak bazı yüksek performanslı sistemlerde sıvı soğutma çözümleri de kullanılabilir. **Hava Soğutma (Air Cooling)**, ısı dağıticılar (heatsinks) ve fanlar kullanarak bileşenlerden yayılan ısıyı dağıtır. CPU ve GPU için bu hava soğutucular yaygındır. **Sıvı Soğutma (Liquid Cooling)**, ısı dağıtımında sıvı kullanarak daha etkili soğutma sağlar. Su blokları, radyatörler, pompalar ve sıvı boruları kullanılarak sıcaklık kontrol edilir. **Pasif Soğutma (Passive Cooling)**, Fan veya hareketli parça kullanmadan doğal hava akışı ile ısı dağılığını sağlar. Isı dağıticılar ve büyük metal yüzeyler kullanılır.



Soğutma birimlerinin genel görünümü

## Bilgisayarın Tarihçesi

Bilgisayarlar ve programcılık ile ilgilenen herkesin bilgisayarların gelişimi konusunda az da olsa bir bilgi sahibi olması gereklidir. Tarihte bilinen ilk hesap makinesinin, Asya'da icat edilen abaküs olduğu kabul edilir. Orta Çağ'ın sonlarına kadar kullanılan abaküs basit toplama ve çıkarma işlemleri yapmak için kullanılmıştır. Fransız filozof ve matematikçi Blaise Pascal'ın 1642 yılında icat ettiği ve Pascaline ismini verdiği hesap makinesi, sekiz basamağa kadar toplama işlemi yapmakta kullanılabiliyordu. Alman matematikçi ve filozof Leibniz, 1673 yılında, Pascal'ın makinesini daha da geliştirerek çarpma ve bölme işlemlerini de yapabilen hale getirmiştir.

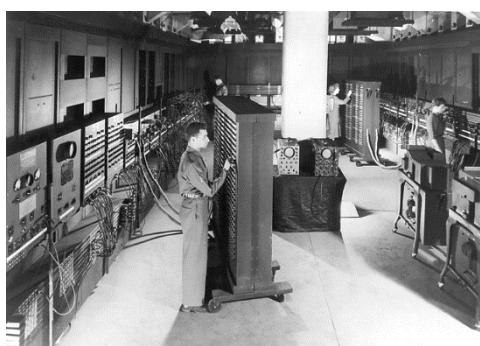


Pascaline ve Leibniz hesap makinesi

1819 yılında bir Fransız dokumacı Joseph Jacquard, dokuma tezgahlarında delikli kartlar kullanmak suretiyle iğnelerin istenen sırayı izlemesini ve kumaşın istenilen renk ve desene sahip olmasını sağladı. Kartlardaki deliklerin yerleri değiştirilerek kumaşın renk ve desenleri de değiştirilebiliyordu. Yirminci yüzyılda geliştirilen ilk bilgisayarlar da delikli kartlar kullanılarak programlanıyordu. Bu açıdan, Joseph Jacquard'ın icadı bilgisayarların gelişimi için son derece önemli bir adım olmuştur. İngiliz matematikçi Charles Babbage, bugünkü anlamdaki bilgisayarları ilk kez tasarlayan kişi olarak kabul edilmekle birlikte, Babbage, tasarladığı bilgisayarları hayatı geçirememiştir. Babbage'ın 1937'lerde tasarladığı bilgisayarlarda girdi, hafıza, işlemci ve çıktı üniteleri yer almaktaydı.

Yirminci yüzyılda bilgisayarın gelişimi açısından en büyük aşama 1947 yılında ENIAC (Electronic Numerical Integrator and Calculator) isimli bilgisayarın yapılması olmuştur. John W. Mauchly ve J. Presper Eckert tarafından yapılan ENIAC, yaklaşık 18.000 vakum tüpü ile çalışıyordu, bir oda büyüklüğünde ve 30 ton ağırlığındaydı. Önceki hesaplama makinelerinden farklı olarak 10 basamağa kadar sayıları hafızasında tutabiliyordu. 1950'lerden sonra Bell Laboratuvarları tarafından geliştirilen transistörler, vakum tüplerinin yerini almaya başladı. Bu gelişme bilgisayarların çok daha hızlı çalışmasına olanak verdi. 1951 yılında UNIVAC 1 (Universal Automatic Computer) isimli ilk ticari amaçlı bilgisayar tanıtıldı.

ENIAC



UNIVAC



Entegre devrelerin icadıyla birlikte 1960'lardan itibaren üçüncü nesil bilgisayarlar üretilmeye ve kullanılmaya başlandı. Bu gelişme sayesinde bilgisayarlar çok daha küçük, güçlü ve birden fazla programı aynı anda çalıştırabilir hale geldiler. Intel ilk mikroişlemcisini 1971 yılında tanıttı. İlk Intel mikroişlemcisi olan Intel 4004, saniyede 90.000 işlem gerçekleştirebiliyordu.

1980 yılında Microsoft Disk Operating System (MSDOS) tanıtıldı. İzleyen yılda IBM ev ve ofis kullanımı için ilk kişisel bilgisayarı tanıttı. 1984 yılında Apple Macintosh, grafik ara yüzlü ilk bilgisayarı tanıttı. Bunu da 90'larda Microsoft tarafından sunulan Windows işletim sistemi izledi. 1992'de grafik arayüzlü işletim sistemi Windows 3.1 kullanıma sunuldu. 1995'de Windows 95 olarak ve 2001'de Windows XP olarak güncellenmiştir.



**İlk kişisel bilgisayarlar ve işletim sistemi**

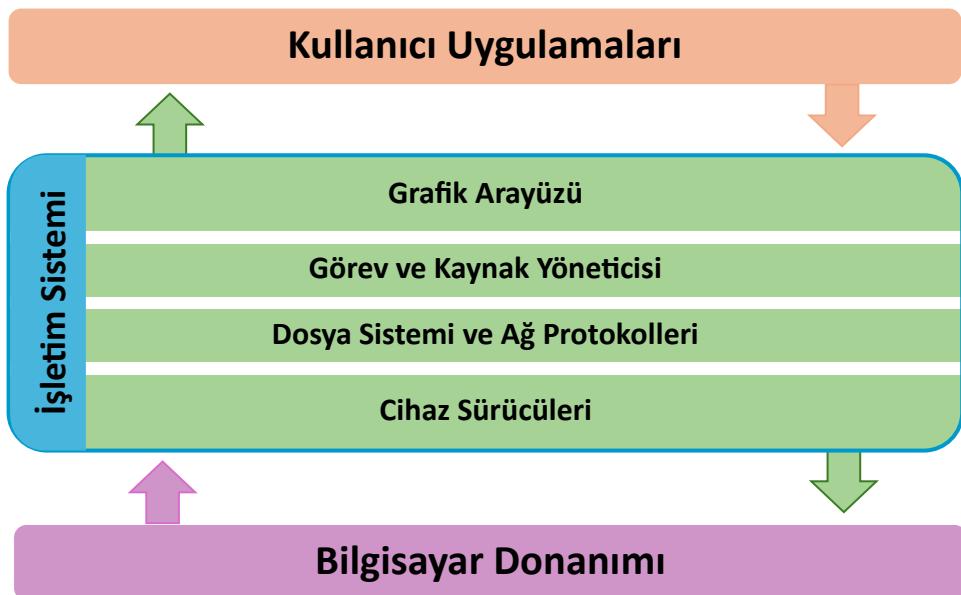
Doksanlı yıllarda sonra bilgisayarların gelişimi ve kullanımı hızlanarak arımıştır. Bu dönemde bilgisayarların giderek güçlenip hızlanırken aynı zamanda fiziksel olarak küçülmüştürler. Küçülen bilgisayarlar ile yavaş yavaş masaüstü bilgisayarların yerini dizüstü ve tablet bilgisayarlar almaya başlamıştır. İnternet kullanımının yaygınlaşmasıyla birlikte daha önceden sadece bilgisayarlar ile yapılabilen işlemler günümüzde akıllı telefonlarla yapılabilmektedir.

Bilgisayarları tarihsel gelişimlerine göre şu şekilde sınıflandırmak mümkündür.

- Birinci Nesil Bilgisayarlar (1945 - 1956) : vakum tüplü devreler
- İkinci Nesil Bilgisayarlar (1956 - 1964) : transistörlü devreler
- Üçüncü Nesil Bilgisayarlar (1964 - 1970) : bütünüleşik devreler
- Dördüncü Nesil Bilgisayarlar (1970 - 2000) : mikro işlemcili devreler
- Beşinci Nesil Bilgisayarlar (Günümüzden - Geleceğe) : paralel işlemcili devreler

## İşletim Sistemleri

İşletim sistemleri (OS), bilgisayar donanımı ile kullanıcılar arasında bir grafiksel arayüz ile bağlantı sağlayan ve bilgisayarın tüm donanım ve yazılım kaynaklarını yöneten temel sistem yazılımıdır. İşletim sistemleri, bilgisayarın çalışmasını sağlar ve kullanıcıların uygulamaları etkin bir şekilde kullanabilmesi için gerekli olan temel hizmetleri sunar. Bu açıdan işletim sistemleri, modern bilgisayarların vazgeçilmez bir parçasıdır.



Bir işletim sisteminin genel yapısı

Bir işletim sisteminin temel özellikleri kaynak yönetimi, dosya yönetimi, kullanıcı arayüzü, güvenlik ve erişim kontrolü olarak ifade edilir. İyi tasarlanmış bir işletim sistemi, bilgisayarın verimli ve stabil çalışmasını sağlar, kaynakları etkin bir şekilde kullanarak performansı artırır. Ayrıca, kullanıcıların çeşitli uygulamaları sorunsuz bir şekilde çalıştırabilmelerine olanak tanır. Günümüze kadar çeşitli kuruluşlar tarafından belirli sistemleri için oluşturulmuş birçok işletim sistemi mevcuttur. Her bir işletim sistemi kullanım amacına göre belirli donanımlara sahip olup bunlara uygun ayarlamalar içermektedir. Yaygın olarak kullanılan işletim sistemlerin bazıları şunlardır.

- Microsoft Windows 1985
- MS-DOS 1981
- Apple MacOS 1984
- IBM OS/2 1987
- UNIX 1969
- Linux 1991
- Solaris 1992

## **Microsoft Windows**

Kullanıcı dostu bir grafiksel arayüz (GUI) sunan ve kişisel bilgisayarlarda en yaygın kullanılan işletim sistemidir. Temel özellikleri arasında çoklu görev desteği, geniş donanım uyumluluğu, çeşitli yazılım ve uygulama desteği, gücü güvenlik özellikleri ve kolay ağ bağlantıları yer alır. İlk çıkış tarihi 1985 olan sistemin ilk sürümü Windows 1.0'dır. Daha sonraki yıllarda Windows 3.1, Windows 95, Windows XP, Windows 7, Windows 10, Windows 11 olarak sistem güncellenmiştir.



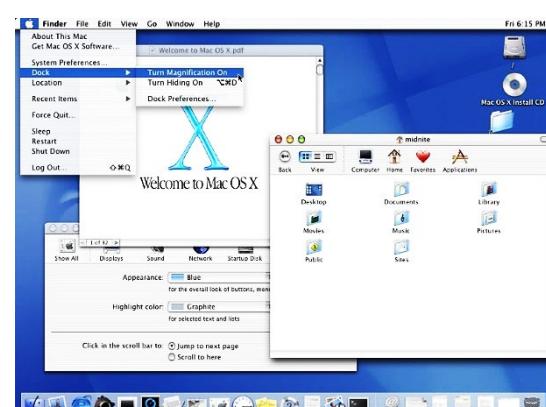
## **MS-DOS (Microsoft Disk Operation System)**

1981 yılında IBM PC'ler için Microsoft tarafından geliştirilen ve 1980'ler ile 1990'lara kadar yaygın olarak kullanılan bir komut satırı tabanlı işletim sistemidir. Kullanıcılar, komut satırı aracılığıyla dosya yönetimi, program çalışma ve disk işlemleri gibi temel işlevleri yerine getirebilir. MS-DOS, basit ve düşük kaynak tüketimi ile dikkat çeker, bu da onu dönemin sınırlı donanım kapasiteleri için ideal kılmıştır. MS-DOS 1.0, 5.0, 6.22 gibi sistem güncellemeleri olmuştur.



## **Apple MacOS (Macintosh Operation System)**

Apple'ın Macintosh bilgisayarları için geliştirdiği işletim sistemidir. MacOS, güçlü bir UNIX tabanına sahiptir ve kullanıcı dostu bir grafik arayüzü sunar. MacOS, yüksek performanslı ve güvenilir bir işletim sistemi olarak bilinir. Özellikle kreatif profesyoneller ve geliştiriciler için popülerdir çünkü grafik ve medya işleme uygulamaları için optimize edilmiştir. MacOS, güçlü dosya yönetimi, çoklu masaüstü desteği, Spotlight arama özelliği gibi özellikler sunar. İlk sistem sürümü MacOS daha sonra MacOS X, Mac OS Big Sur, MacOS Monterey olarak güncellenmiştir.



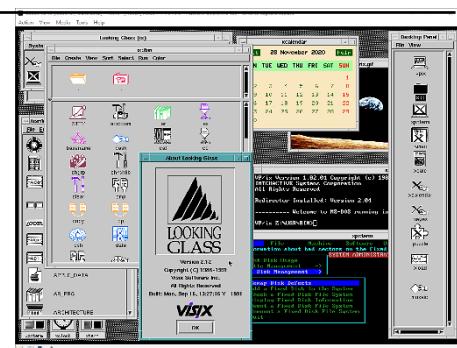
IBM OS/2

IBM OS/2, 1987 yılında IBM ve Microsoft'un ortaklığıyla geliştirilen, özellikle kurumsal kullanıcılar ve sunucular için tasarlanmış bir işletim sistemidir. OS/2, Grafiksel kullanıcı arayüzü (GUI)'ye sahip olup çoklu görev yetenekleri, gelişmiş bellek yönetimi ve yüksek güvenilirlik özellikleri ile dikkat çeker. OS/2, güçlü ağ özellikleri sayesinde geniş ağ yapılandırmalarını ve istasyonlarını destekler. İlk sürüm OS/2 1.0'dan sonra OS/2 2.0 ve Warp olarak güncellenmiştir.



UNIX

AT&T Bell Laboratuvarlarında 1969 yılında C programlama dilile yazılan UNIX işletim sistemi, çoklu kullanıcı ve çoklu görev yetenekleri ile tanınan güçlü ve güvenilir bir işletim sistemidir. Özellikleri arasında taşınabilirlik, modülerlik, esneklik ve güvenlik mekanizmaları bulunur. Özellikle sunucular, ana bilgi-sayarlar ve iş istasyonları gibi kritik sistemlerde yaygın olarak kullanılır. AIX, HP-UX, BSD gibi türevleri bulunmaktadır.



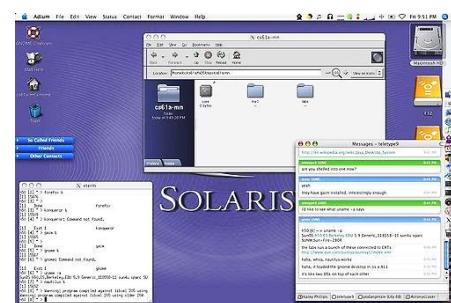
Linux

1991 yılında Linus Torvalds tarafından Unix tasarım ilkelerinden yararlanılarak geliştirilmiştir. O dönemde lisanslı ve pahalı olan işletim sistemlerine alternatif olarak açık kaynaklı ve özgür bir yazılım olarak GNU General Public License ile yayınlandı. Geliştiriciler ve sistem yöneticileri için güçlü komut satırı araçları ve geniş yazılım depoları sunar. Linux'un modüler yapısı, kullanıcıların kendi sistemini özelleştirmesine olanak tanır. Linux'in, Ubuntu, Fedora, Debian gibi çeşitli dağıtımları mevcuttur.



Solaris

1992 yılında Sun Microsystems tarafından geliştirilen ve şimdi Oracle tarafından yönetilen UNIX tabanlı bir işletim sistemidir. Solaris, güçlü ağ ve güvenlik özellikleri, yüksek performans, ZFS dosya sistemi, DTrace (dynamik izleme), ve Containers (hafif sanallaştırma) gibi ileri düzey özellikler sunar. Yüksek erişilebilirlik ve veri bütünlüğü sağlayarak, özellikle büyük kurumsal ağları yönetimi ve güvenliği için idealdir. İlk sürümden sonra Solaris 2.x, Solaris 10, Solaris 11 olarak güncellenmiştir.



## Bir İşletim Sisteminin İşlevleri

Uygulama yazılımları ile kullanıcılar arasında bir arayüz bağlantısı sağlayan ve bilgisayarın donanım kaynaklarını yöneten bir işletim sisteminin işlevleri genel olarak kaynak yönetimi, dosya yönetimi, kullanıcı arayüzü sağlama ve güvenlik ve erişim kontrolü şeklinde sıralanabilir. Ayrıca, işletim sistemi, bilgisayarın çalışma zamanını yöneterek, çoklu görev yürütme, çeşitli işlemleri eşzamanlı olarak çalıştırma ve uygulamalar arası iletişim kolaylaştırma gibi yetenekler de sunmaktadır. Temel işlevler;

**1. Kaynak Yönetimi:** İşletim sisteminin en kritik işlevlerinden biri olarak bilgisayarın verimli ve etkili çalışmasını sağlar. Optimum kaynak kullanımını sağlayarak, donanımın performansını en üst düzeye çıkarır. Bilgisayarın çoklu görev yeteneklerini geliştirir; farklı işlemler arasında CPU zamanını dağıtarak kullanıcıya sorunsuz bir çalışma ortamı yaratır.

- |                                   |   |
|-----------------------------------|---|
| <b>CPU Yönetimi</b>               | : İşlemlerin CPU'da yürütülme sırasını belirler.                              |
| <b>Bellek Yönetimi</b>            | : RAM'ın etkin kullanımını sağlar ve belleği programlar arasında tahsis eder. |
| <b>Depolama Yönetimi</b>          | : Dosya sistemlerini yönetir ve veri depolama birimlerine erişimi denetler.   |
| <b>Giriş/Çıkış (I/O) Yönetimi</b> | : Giriş ve çıkış aygıtlarının etkin kullanımını sağlar.                       |
- 

**2. Dosya Yönetimi:** Bilgisayarda verilerin düzenli ve güvenli bir şekilde saklanması sağlar. Bu işlev, verilerin kolay erişilebilir olmasını, düzenli bir yapı içinde organize edilmesini ve veri güvenliğinin sağlanmasını temin eder. Dosya yönetimi, ayrıca veri yedekleme ve kurtarma işlemleri için gerekli araçları da bünyesinde tutar.

- |                        |   |
|------------------------|---|
| <b>Dosya Sistemi</b>   | : Dosyaların oluşturulması, silinmesi, kopyalanması, taşınması ve yeniden adreslendirilmesi gibi işlemleri yönetir. |
| <b>Erişim Kontrolü</b> | : Kullanıcıların dosyalara okuma, yazma ve yürütme erişim izinlerini yönetir.                                       |
- 

**3. Kullanıcı Arayüzü:** İşletim sisteminin kullanıcı ile bilgisayar arasında etkili bir etkileşim sağlamasını temin eder ve bilgisayarın işlevlerini kolayca kullanabilmesini sağlar. İyi bir kullanıcı arayüzü, kullanıcıların işlemleri hızlı ve verimli bir şekilde gerçekleştirmelerini sağlar, iş akışını iyileştirir ve bilgisayarı daha erişilebilir hale getirir.

- |  |   |
|--|---|
| <b>Komut Satırı Arayüzü (CLI)</b>        | : Kullanıcıların komut yazarak bilgisayarla etkileşime girmesini sağlar.                          |
| <b>Grafiksel Kullanıcı Arayüzü (GUI)</b> | : Grafiksel simgelerin ve pencerelerin kullanılmasıyla daha kullanıcı dostu bir etkileşim sağlar. |
- 

**4. Güvenlik ve Erişim Kontrolü:** Bir işletim sisteminin kritik işlevlerinden olup, veri ve sistemlerin yetkisiz erişimlerden ve güvenlik tehditlerinden korunmasını sağlar.

- |   |   |
|---|---|
| <b>Kullanıcı Hesapları ve Parolalar</b> | : Çoklu kullanıcı desteği ve her kullanıcının verilerinin korunmasını sağlar. |
| <b>Yetkilendirme</b>                    | : Kullanıcıların bilgisayarda hangi işlemleri yapabileceğini kontrol eder.    |

## Bilgisayar Programlama

Programlama, bilgisayarların belirli görevleri yerine getirmesini sağlamak için yazılım oluşturma sürecidir. Bu süreç, bir bilgisayara nasıl çalışması gerektiğini anlatan talimatları yazmayı içerir. Programlamanın daha iyi anlaşılabilmesi için bazı temel kavramların bilinmesi zorunludur:

1. **Kod Yazma:** Bilgisayarın belirli görevleri yerine getirmesini sağlamak için talimatlar dizisi oluşturmakta. Bu talimatlar, belirli kuralları ve sözdizimini takip eden programlama dilleri kullanılarak yazılmıştır. Kod yazma süreci, bir problemi çözmek veya belirli bir işlevi gerçekleştirmek amacıyla algoritmaları ve mantıksal adımları düzenlemeyi içerir. Bu talimatlar bir bilgisayara ne yapması gerektiğini ve nasıl yapması gerektiğini anlatır.
2. **Programlama Dilleri:** Programlama dilleri, bilgisayarlarla iletişim kurmak ve onlara belirli görevleri gerçekleştirmelerini söylemek için kullanılan özel dilleridir. Her programlama dili, belirli bir sözdizimi ve kurallar bütünü içerir ve farklı ihtiyaçlara ve uygulamalara yönelik olarak tasarlanmıştır. Programlama dilleri, yazılımların geliştirilmesi, oyun yapımı, bilimsel hesaplamalar ve daha birçok alanda önemli bir rol oynar, bu nedenle doğru dil seçimi başarı için kritik bir öneme sahiptir. Örneğin, C++, Java ve Python gibi popüler programlama dilleri vardır.
3. **Algoritma:** Bir problemi çözmek için izlenen adımların toplamı "algoritma" olarak adlandırılır. Algoritmalar, problemi adım adım çözmek için gerekli olan tüm işlemleri ve bu işlemlerin sırasını belirler. Bu adımlar, problemi en etkili ve verimli şekilde çözmek için mantıksal bir sıra ile düzenlenir. Bir programlama diliyle kod haline getirilen algoritmalar, bilgisayara adım adım ne yapması gerektiğini açıklar.
4. **Değişkenler ve Veri Türleri:** Programlarda kullanılan veriler, "değişkenler" aracılığıyla saklanır ve işlenir. Değişkenler, programın çalışması sırasında değerlerin geçici olarak depolanmasını sağlayan isimlendirilmiş bellek alanlarıdır. Bu değişkenler, belirli türde verileri saklar; örneğin, sayısal veriler (tam sayılar, ondalıklı sayılar), metinsel veriler (karakterler ve dizeler), mantıksal veriler (doğru veya yanlış değerler) gibi. Bu sayede, programlar daha esnek ve güçlü hale gelir, çünkü değişkenler aracılığıyla farklı türde verilerle çalışmak mümkün olmaktadır.
5. **Fonksiyonlar ve İşlevler:** Fonksiyonlar, belirli bir işi yerine getiren kod bloklarıdır. Programlama dillerinde, kodu organize etmek ve tekrar kullanılabilir hale getirmek için fonksiyonlar kullanılır. Her fonksiyon belirli bir işlevi yerine getirir ve gerektiğinde çağrılarak tekrar kullanılabilir.
6. **Hata Ayıklama:** Programlamada hata ayıklama, yazılım geliştirme sürecinin kritik bir aşamasıdır ve kodda bulunan hataların (bug) bulunması ve düzeltilmesi işlemlerini kapsar. Bu süreç, yazılımın beklenmeyen davranışlar sergilemesine veya yanlış sonuçlar üretmesine neden olan sorunların tespit edilmesi ve giderilmesi için yapılan çalışmaları içerir.

Programlama, bilgisayarlara ve diğer teknolojilere çeşitli işlevler kazandırmak için temel bir beceridir. Yazılım geliştirme, oyun yapımı, veri analizi ve daha birçok alanda kullanılır. Bu süreç, mantıksal düşünmeyi, problem çözmeyi ve yaratıcılığı teşvik eder.

Assembly dili, makine diline göre daha açık ve anlaşılır olmakla birlikte yine de öğrenmesi ve program yazması uzun süren bir dildir. Ayrıca basit işlemler için bile uzun kodlar yazmak gerekmektedir. Bunun üstesinden gelebilmek için çok daha anlaşılır ve basit olan yüksek seviye programlama dilleri geliştirilmiştir. Makine dilinden, yüksek seviyeli dillere gidildikçe program konuşma diline daha çok benzemeye başlar. Python, C, C++, Java, Fortran gibi diller buna örnek verilebilir. Yukarıdaki kodla aynı işi yapan Python programı aşağıdaki gibidir.

```
while n > 0:
```

```
    sum += n
```

```
    n -= 1
```

Bu koddan da anlaşılacağı üzere bir yüksek seviyeli dil ile program yazmak makine dili ve assembly diline kıyasla çok daha kolaydır ve kodun anlaşılmasımda da bir zorluk yoktur. Bu kodlardan da görüleceği üzere 1. kuşak dil olan makine dillerinden sonra assembly dili 2. kuşak dil olarak adlandırılmaktadır. Assemblyden sonraki diller 3. kuşak diller ve görsel bir ortamda yazılan diller ise 4. kuşak diller olarak bilinirler.

Günümüze kadar kullanım amaçlarına göre çok çeşitli programlama dilleri geliştirilmiştir. Geçmişte bir dönem ve günümüzde yaygın olarak kullanılan programlama dillerinden bazıları şu şekilde verilebilir:

## COBOL

1950'li yıllarda Amerikan Savunma Bakanlığının bir projesi kapsamında geliştirilen dil “**Common Business Oriented Language**” kelimelerinden türetilmiştir. Daha çok iş dünyasında veri işleme ve finansal uygulamalar için geliştirilen bir yüksek seviyeli programlama dilidir ve özellikle büyük ölçekli ana bilgisayar (mainframe) sistemlerinde popülerdir. Dil, okunabilirliği ve anlaşılabilirliği artırmak için İngilizcaye benzer bir sözdizimi kullanır, bu da programcılar karmaşık iş mantığını kolayca yazmasına ve bakımını yapmasına olanak tanır.

```
EDIT      Z57658.SOURCE.COBOL(EDITCHAR) - 01.00
Command ==> _____
***** ***** Top of Data ****
000100  IDENTIFICATION DIVISION.
000200  PROGRAM-ID. EDITCHAR.
000300  DATA DIVISION.
000400  WORKING-STORAGE SECTION.
000500  01 NUNZZ      PIC ZZZZ9.
000600  01 NUMDOLL   PIC $99999.
000700  01 NUNSTAR   PIC *99999.
000900  PROCEDURE DIVISION.
000910  MOVE 00005  TO NUMZZ
000920  MOVE 00055  TO NUMDOLL
000930  MOVE 00055  TO NUMSTAR
001000  DISPLAY 'NUMZZ : ' NUMZZ
001100  DISPLAY 'NUMDOLL : ' NUMDOLL
001200  DISPLAY 'NUMSTAR : ' NUMSTAR.
001400  STOP RUN.
```

## FORTRAN

Fortran (**Formula Translation**), 1950'lerin başında IBM tarafından geliştirilen ve özellikle bilimsel ve mühendislik hesaplamaları için tasarlanmış yüksek seviyeli bir programlama dilidir. Fortran, matematiksel hesaplamalar ve sayısal analizler için optimize edilmiştir ve bu nedenle bilim insanları ve mühendisler arasında popüler olmuştur. Dil, güçlü sayısal işlem yetenekleri ve taşınabilirliği ile bilinir, bu da onu karmaşık hesaplamaların ve büyük ölçekli veri işlemlerinin yapıldığı alanlarda ideal kılar.

```
1 program main
2  implicit none
3  ! declaring variables
4  complex :: a, b, c, z_1, z_2, delta, sqr_delta
5  print *, "Please enter coefficients-->(a, b, c):"
6  read *, a
7  read *, b
8  read *, c
9  delta = cmplx(b ** 2 - 4.0 * a * c)
10  sqr_delta = csqrt(delta)
11  z_1 = (-b + sqr_delta) / (2.0 * a)
12  z_2 = (-b - sqr_delta) / (2.0 * a)
13  print *, "z_1:", z_1, "z_2:", z_2
14 end program main
```

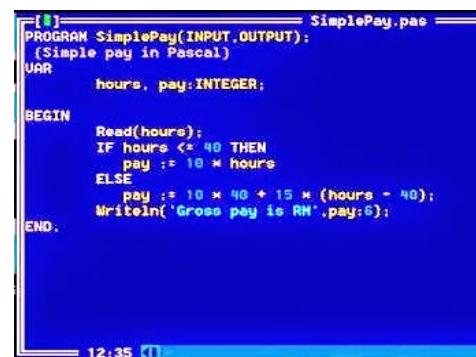
## BASIC

BASIC (Beginner's All-purpose Symbolic Instruction Code), 1960'ların ortalarında Dartmouth Koleji'nde John G. Kemeny ve Thomas E. Kurtz tarafından geliştirilen, özellikle programlamaya yeni başlayanlar için tasarlanmış, kullanıcı dostu bir programlama dilidir. BASIC, sadeliği ve anlaşılır sözdizimi sayesinde geniş bir kullanıcı kitlesine hitap eder ve programlama eğitimi için ideal bir araç olmuştur. Özellikle kişisel bilgisayarların yaygınlaşmasıyla 1970'ler ve 1980'lerde birçok bilgisayar sistemi için çeşitli türevleri geliştirilmiştir.

```
10 INPUT "What is your name: "; U$  
20 PRINT "Hello "; U$  
25 REM  
30 INPUT "How many stars do you want: "; N  
35 S$ = ""  
40 FOR I = 1 TO N  
50 S$ = S$ + "*"  
55 NEXT I  
60 PRINT S$  
65 REM  
70 INPUT "Do you want more stars? "; A$  
80 IF LEN(A$) = 0 THEN GOTO 70  
90 A$ = LEFT$(A$, 1)  
100 IF (A$ = "Y") OR (A$ = "y") THEN GOTO 30  
110 PRINT "Goodbye";  
120 FOR I = 1 TO 200  
130 PRINT U$; " "  
140 NEXT I  
150 PRINT
```

## PASCAL

1970 yılında Niklaus Wirth tarafından geliştirilmiş, özellikle eğitim amacıyla tasarlanmış bir yüksek seviyeli programlama dilidir. Yapısal programlamayı teşvik eden Pascal, temiz ve okunabilir bir sözdizimine sahiptir ve programcıların mantıksal ve düzenli kod yazmalarına olanak tanır. Pascal, güçlü tip denetimi ve prosedürel programlama özellikleriyle bilinir ve özellikle akademik çevrelerde algoritma ve veri yapıları öğretiminde yaygın olarak kullanılmıştır.



```
[1] PROGRAM SimplePay(INPUT,OUTPUT);  
  (Simple pay in Pascal)  
VAR  
  hours, pay:INTEGER;  
  
BEGIN  
  Read(hours);  
  IF hours <= 40 THEN  
    pay := 10 * hours  
  ELSE  
    pay := 10 * 40 + 15 * (hours - 40);  
  WriteLn('Gross pay is RM',pay:6);  
  
END.  
12:35
```

## C / C++

C programlama dili, 1970'lerde Dennis Ritchie tarafından geliştirilen ve bilgisayar bilimlerinde önemli bir yere sahip olan bir dildir. C, düşük seviyeli sistem programlamasından yüksek seviyeli uygulama geliştirmeye kadar geniş bir kullanım yelpazesine sahiptir. C++, 1980'lerde Bjarne Stroustrup tarafından geliştirilen güçlü ve esnek bir programlama dilidir. C dilinin üzerine nesne yönelimli programlama (OOP) özellikleri ekleyerek, daha karmaşık ve büyük yazılım projelerinde yapı ve organizasyon sağlamayı amaçlar.

```
#include <math.h>  
#include <iostream>  
#include <span>  
#include <vector>  
  
struct Circle {  
    float r;  
};  
  
void PrintTotalArea(std::span<Circle> circles) {  
    float area = 0;  
    for (const Circle& c : circles) {  
        area += M_PI * c.r * c.r;  
    }  
    std::cout << "Total area:" << area << "\n";  
}
```

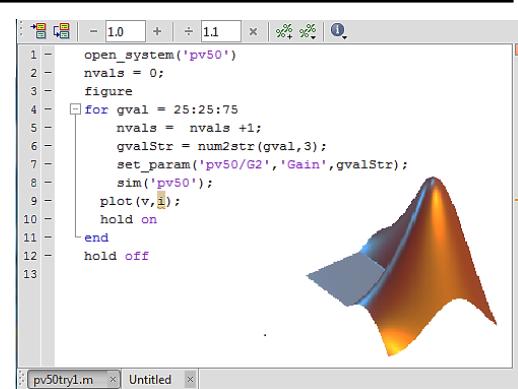
## JAVA

1995 yılında Sun Microsystems tarafından geliştirilen ve günümüzde geniş bir kullanım alanına sahip olan nesne yönelimli bir programlama dilidir. Platform bağımsızlığı ve taşınabilirliği ile bilinen Java, "Bir kez yaz, her yerde çalıştır" felsefesiyle tasarlanmıştır; bu, bir Java programının farklı işletim sistemlerinde ve cihazlarda çalışmasını sağlar. Java, zengin kütüphane desteği, güçlü güvenlik özellikleri ve geniş topluluk desteği ile tanınır. Genellikle büyük ölçekli kurumsal uygulamalarda, mobil uygulamalarda ve web uygulamalarında kullanılır.



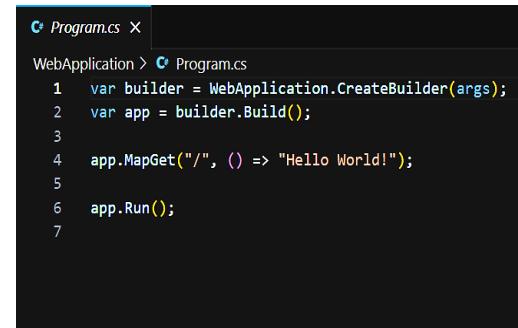
## MATLAB

Matrix Laboratory (**Matris Laboratuvar**) sözlerinden türetilmiştir. Mathworks firması tarafından, matematiksel hesaplamalar, veri analizi, algoritma geliştirme ve modelleme için yaygın olarak kullanılan yüksek seviyeli bir programlama dilidir. Özellikle mühendislik, bilim ve matematik alanlarında kompleks hesaplamaları kolaylaştırmak amacıyla tasarlanmıştır. Kullanıcı dostu bir arayüze sahip olan MATLAB, geniş bir kütüphane ve fonksiyon seti sunar.



## C#

C# (C-Sharp), Microsoft tarafından geliştirilen ve .NET Framework üzerinde çalışan, modern ve nesne yönelimli bir programlama dilidir. C#, güçlü tür denetimi, basit sözdizimi ve geniş kütüphane desteği ile bilinir, bu da onu yazılım geliştirme, web uygulamaları, masaüstü uygulamaları ve oyun geliştirme gibi birçok farklı alanda popüler bir seçenek haline getirir. C#, Java'ya benzer bir sözdizimine sahip olup, güçlü bellek yönetimi ve güvenlik özellikleri sunar.



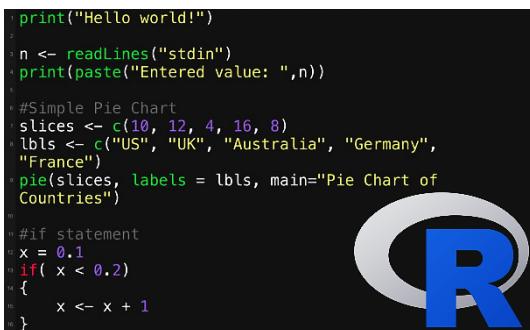
## PYTHON

Guido van Rossum tarafından 1991 yılında geliştirilen Python, yüksek seviyeli, açık kaynaklı ve çok amaçlı bir programlama dilidir. Python, sade ve okunabilir bir sözdizimi ile tanınır, bu da yeni başlayanlar için öğrenmeyi kolaylaştırır. Dinamik tip kontrolü, otomatik bellek yönetimi ve geniş standart kütüphanesi sayesinde Python, web geliştirme, veri analizi, yapay zeka ve bilimsel hesaplamalar gibi alanlarda yaygın olarak kullanılır.



## R

1993 yılında Auckland Üniversitesi'nden Ross Ihaka ve Robert Gentleman tarafından geliştirilmiştir. Özellikle veri analizi, istatistiksel hesaplamalar ve veri görselleştirme için geliştirilmiş güçlü bir dildir. Açık kaynaklı ve geniş bir kullanıcı topluluğuna sahip olan R, istatistiksel modeller oluşturmak, veri setlerini analiz etmek ve grafikler üretmek için çeşitli kütüphaneler ve paketler sunar. Gelişmiş grafik ve görselleştirme yetenekleriyle verilerin anlamlı ve etkileyici bir şekilde sunulmasını sağlar.



## Algoritma Kavramı

Belirli bir problemi çözmek veya bir görevi yerine getirmek için izlenen adımların sıralı bir planıdır. Bir algoritma, problemin çözümü için gereken işlemleri düzenler ve bu işlemlerin hangi sırayla yapılacağını belirler. Mesela algoritmayı bir yemek tarifi gibi düşünebilirsiniz; tarife göre malzemeleri eklemek, karıştırmak ve pişirmek gibi adımlar, problemi (yemek yapmayı) çözmek için izlenir. Programlamada ise algoritmalar, bilgisayarın veya yazılımın hangi adımları takip edeceğini ve işlemleri nasıl gerçekleştireceğini belirler. Algoritmalar, genellikle bir dizi mantıksal adım ve karar noktasından oluşur.

Genellikle bir problemin çözümü için tek bir algoritmadan bahsedilemez. Doğru, çözüm sağlayan ve birbirine göre belki daha üstün yanları bulunan birden fazla çözüm yolu her zaman olabilir. Ancak bir algoritmanın geliştirmesinde de temel bir metodoloji bulunmaktadır. Çözüm yolları farklıda olsa çözümün oluşturulma şekli aynıdır. Algoritma geliştirmede temel kavamlar şöyle verilebilir:

1. Çözümün Tasarlanması,
2. Değişkenlerin Tanımlanması,
3. Algoritmanın Görselleştirilmesi: Akış Diyagramı

## Çözümün Tasarlanması

Bir problemin bilgisayarla çözümünün doğru bir şekilde tasarlanabilmesi için **istenen**, **verilen** ve verilerden istenen elde etmek için yapılması **gerekilen işlemleri** tam ve doğru bir şekilde belirlemek gereklidir. **Genellikle verilenler veri ile, istenilenler de bilgi ile temsil edilir.** Bilginin işlenmiş veri olduğunu bir kez daha hatırlatmakta fayda vardır. Veri ile bilgileri hem algoritmada hem programlamada değişkenler ile ifade etmekteyiz. Verilenler ve istenilenler belirlendikten sonra, bunları temsil edecek değişkenleri ve yapılması gereken işlemleri tanımlamak gereklidir. Verilenden istenene doğru giderken verinin de bilgiye dönüşüm işlemi gerçekleşmiş olur. Bu dönüşüm, veriler üzerinde yapılan bir ya da birden fazla işlem ile gerçekleşir. Doğru ve anlaşılır bir şekilde tanımlanmış bir problemde verilenler (**veriler**), istenilenler (**bilgiler**) ve yapılması gereken **işlemler** kolayca tespit edilebilir.

Örneğin klavyeden bir kenar uzunluğu ve o kenara ait yüksekliği girilen üçgenin alanını hesaplayan programı dikkate alalım. Bu örnekte;

1. **İstenen**, üçgenin alanının bulunmasıdır.
2. **Verilenler** ise, bir kenar uzunluğu ve o kenara ait yüksekliktir.
3. **Yapılması gereken işlemler** ise,
  - a- Kullanıcıdan öncelikle **Üçgenin bir kenarını ve bu kenara ait yüksekliği girmesini istemektir.**
  - b- İstenen ve verilenleri temsil edecek **değişkenler tanımladıktan sonra**,
  - c- İsteneni elde etmek için yapılması **gerekilen işlemler tespit edilir.**

1. Başla
2. Kenar ve yüksekliği gir
3. Alanı, alan =  $(a*ha)/2$  formülüne göre hesapla
4. Alanı ekrana yazdır
5. Algoritma sonu

## Değişkenlerin Tanımlanması

Programlamada verilere erişimi kolaylaştırmak için verileri ve bilgileri isimlendirme metodu kullanılmaktadır. Hem programcı hem de derleyiciler veri ve bilgilerin değerlerine bu isimlerle erişir, bu isimleri kullanarak veri ve bilgilerin değerlerini, programda belirtilen işlemlerde kullanır. Bu isimler programcı tarafından oluşturulur ve mümkün olduğu kadar temsil ettiği veriyi veya bilgiyi hatırlatacak şekilde isimlendirme yapılır. İşte, veri ve bilgileri isimlendirmek için kullanılan bu karakter veya karakter gruplarına **değişken (variable)** adı verilir. Değişkenler içerdikleri verilerin türüne göre tamsayı, ondalıklı sayı veya alfabetik veri olarak isimlendirilmektedir.

Nasıl bir tür program hazırlanıyor olursa olsun bir programın tasarılanması aşamasında en çok dikkat etmesi gereken faktörlerden biri değişken tanımlamalarıdır. Değişkenin tipinin ve isminin yanlış seçilmesi bir an için bir problem teşkil etmese de sonraki aşamalarda seçilen yanlış değişken türleri hatalı sonuçlara sebep olabilmektedir. Ayrıca değişkenlere anlamsız isimler verildiğinde de ilk aşamada bir sorun çıkmasa da üzerinde bir zaman geçtikten sonra hangi değişkenin ne amaçla kullanıldığını hatırlamak zor olacaktır. Bunları sonradan tekrar anlamak için programı başından sonuna kadar tekrar gözden geçirmek gerekebilir. Bu durumlarla karşılaşmamak için değişkenlere her zaman anlamlı isimler vermek ve program içinde gerekli açıklama satırlarını eklemek daha doğru olacaktır. Örneğin, “**x toplamı**” diye aklınızdan geçen bir şey varsa o **değişkene “x” adını vermek yerine “toplam” demek** sonrası için daha iyi olacaktır.

Algoritma ve değişken isimlendirmesi ile ilgili bir örnek: Klavyeden girilen N adet tamsayının en küçüğünü ekrana yazacak programın algoritma adımlarını değişken tanımlamaları ile birlikte gösterelim.

1. Girilecek sayı adedini (N) klavyeden giriniz.
2. Bir sayı (S) giriniz.
3. En küçük sayı ( E ) olsun.
4. Kaçıncı sayıyı girdiğimizi gösteren K, 1 olsun.
5. K değeri N'den küçük mü? Hayır ise 11. adıma git.
6. Bir sayı giriniz.
7. Girilen sayı, en küçük sayıdan küçük mü? Hayır ise 9. adıma git.
8. En küçük sayı (E), girilen sayı (S)'dir.
9. K değerini 1 artır.
10. 5. adıma git.
11. En küçük sayıyı ekrana yaz.
12. Algoritma sonu.

## Algoritmanın Görselleştirilmesi: Akış Diyagramı

Programlama ile uğraşan herkesin aynı anlamı verebileceği bazı geometrik semboller kullanılarak bir algoritmanın daha teknik çizimlerle gösterilme biçimine **akış diyagramı (flowchart)** adı verilir. Bu diyagramlar, karmaşık süreçleri basitleştirmek anlamayı kolaylaştırır ve iletişimini güçlendirir. Akış diyagramları, bir problemi veya süreci adım adım çözümlemek ve görsel olarak temsil etmek için önemli bir araçtır, çünkü karmaşıklığı azaltır ve iş akışını anlamayı kolaylaştırır. Programlama, mühendislik, iş süreçleri yönetimi gibi birçok alanda kullanılırlar. Ayrıca bir akış şeması hazırlanırken sembollerin iç içe girmemesi, işlem akış yönü çizgilerinin birden fazla büklüm bulundurmaması ve birbirlerini kesmemesi, başta karşılaşılma işlem kutusu olmak üzere bütün kutulara dörtten fazla giriş/çıkış bağlanması gibi kurallara dikkat etmek gereklidir. Bir akış diyagramında sıkılıkla kullanılan şekiller şöyledir:

