BMB204. Veri Yapıları

Ders 1.
Temel Bilgiler

Erdinç Uzun
NKÜ Çorlu Mühendislik Fakültesi
Bilgisayar Mühendisliği Bölümü

Dersin Hedefi

- Mevcut veri yapıları ve veri modellerini anlamak
- Algoritmaları incelemek, mevcut çözümleri anlamak
- Bilgisayar yazılım dünyasına ait bazı kavramlar, terimler ve sözcükleri
- Bağlantılı liste, ağaç, graf gibi çeşitli veri modellerini
- Programların bellek gereksinimi ve çalışma hızlarının hesabını
- Arama ve sıralama algoritmalarını

Dersin Değerlendirilmesi

- Uygulama + Ödev : %20
  - Ödev, proje şeklinde olacak.
- Ara Sınav : %30
- Final : %50

Derse Çalışma

- Kitaplar
  - Veri Yapıları ve Algoritmalar, Dr. Rifat Çölkesen, Papatya Yayıncılık
- Slaytlar
  - Anahtar Kelimeler (İnternet)
- İnternet
  - http://en.wikibooks.org/wiki/Data_Structures
  - https://www.cs.auckland.ac.nz/~jmor159/PLDS210/ds_ToC.html
  - http://yzgrafik.ege.edu.tr/~ugur/13_14_Fall/DS/index13.html
  - http://www.mmdemirbas.com/2009/05/

Ders Planı

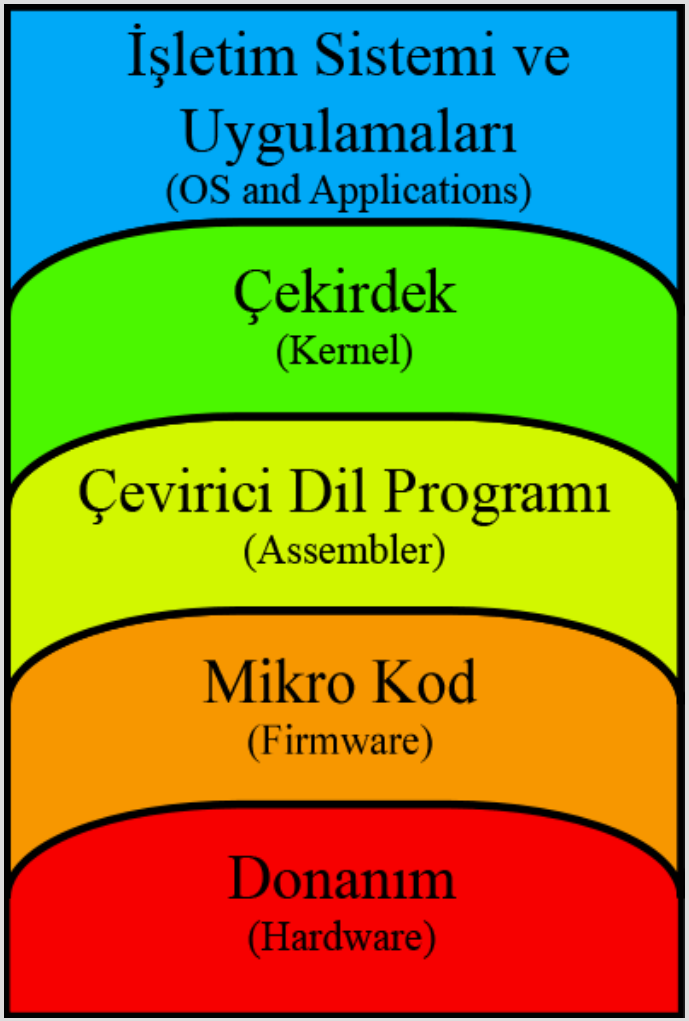
- Donanım, Yazılım ve Program
- Bilgisayar Mimarisi
- İşletim Sistemleri
- Veri Yapısı
- Veri Modeli
- Algoritma
- Programlama Dilleri
- IDE
- Veritabanı ve SQL
- Parçala ve Fethet Yöntemi
- İstemci / Sunucu
- Kıyaslama

Donanım, Yazılım ve Program

- Bilgisayarın oluşmasında, çalışmasında kullanılan somut parçaların tamamına donanım ismi verilmektedir.
- Bilgisayar donanımının çalışmasını olanaklı hale getiren programların tamamına yazılım adı verilmektedir.
- Program, bilgisayara verilen komutlar topluluğu olmakla birlikte, bu komutların, bir işi gerçekleştirme işlemidir.
- Yazılım programlar topluluğundan oluşmaktadır. Her bir programın yazılıma ait bir elemandır.

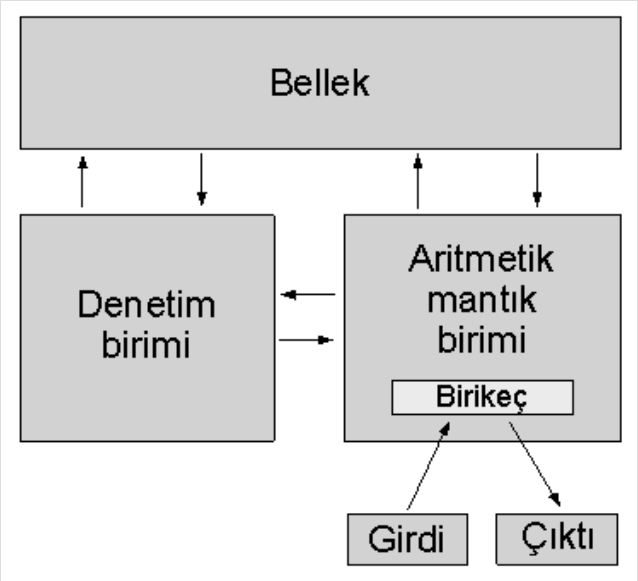
Bilgisayar Mimarisi

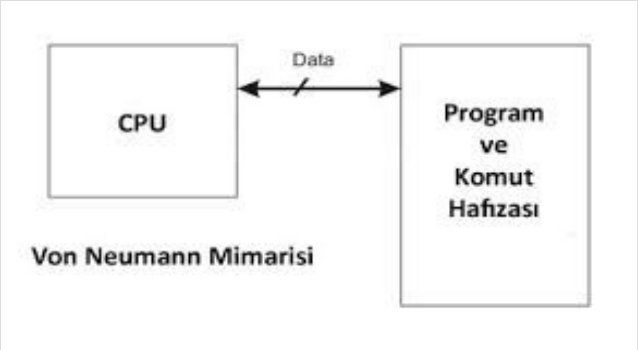
- Bilgisayar Mimarisi, bilgisayar parçalarının iç yapıları ve aralarındaki haberleşme bağlantıları ile ilgilidir.
- Bilgisayar donanımı, bir bilgisayarı oluşturan fiziksel parçaların genel adıdır.
- Mikro kod, mikroişlemcinin komut seti ile yazılmış olan yazılımlardır.
- Çevirme Dili (Assembly Language) bilgisayar programlarını yazmak için kullanılan düşük seviyeli bir programlama dilidir.
- Çekirdek, işletim sisteminin kalbidir. Uygulamalar ve donanım seviyesindeki bilgi işlemleri arasında bir köprü görevi görür.
- İşletim sistemi (Operating System), bilgisayarda çalışan, bilgisayar donanım kaynaklarını yöneten ve çeşitli uygulama yazılımları için yaygın servisleri sağlayan bir yazılımlar bütünüdür.



İlk bilgisayar mimarisi (Von Neumann)

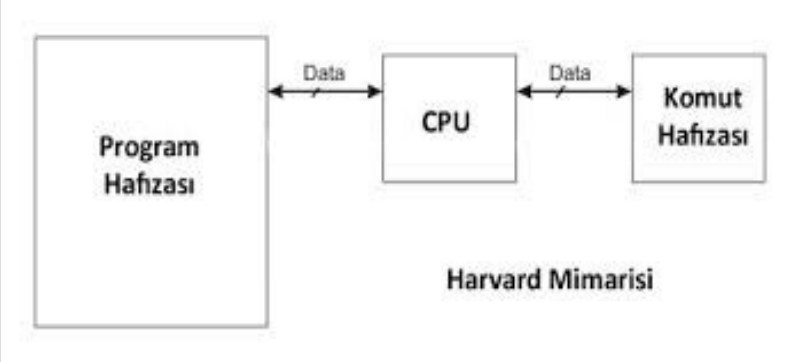
- Bugünkü bilgisayarların mimarisinin modeli ENIAC (Electronic Numerical Integrator And Computer - Elektronik sayısal entegreli hesaplayıcı) üzerinde çalışmış olan John von Neumann tarafından geliştirilmiştir ve von Neumann modelinde mantıksal olarak bilgisayar sistemi tam olarak tanımlanmıştır.
- Bilgisayar sisteminin; bellek, veriyolu, giriş, çıkış ve merkezî işlem biriminden ibaret olduğu düşünülmüştür.
- Von Neumann mimarisine sahip bilgisayarlarda gerçekleştirilen adımlar:
  - program sayacının gösterdiği adresten komut getirilir,
  - program sayacı 1 arttırılır,
  - kontrol birimi getirilen kodun komutunu çözer ve tekrar ilk adıma dönülür.





Harvard Mimarisi

- Harvard mimarili bilgisayar sistemlerinde veri ve buyruklar ayrı belleklerde tutulurlar.
- Komutla beraber veri farklı iletişim yollarını kullanarak ilgili belleklerden alınıp işlemciye getirilebilir. Getirilen komut işlenip gerekli verisi veri belleğinden alınırken sıradaki komut, komut belleğinden alınıp getirilebilir. Bu da hızı arttıran bir etkendir.

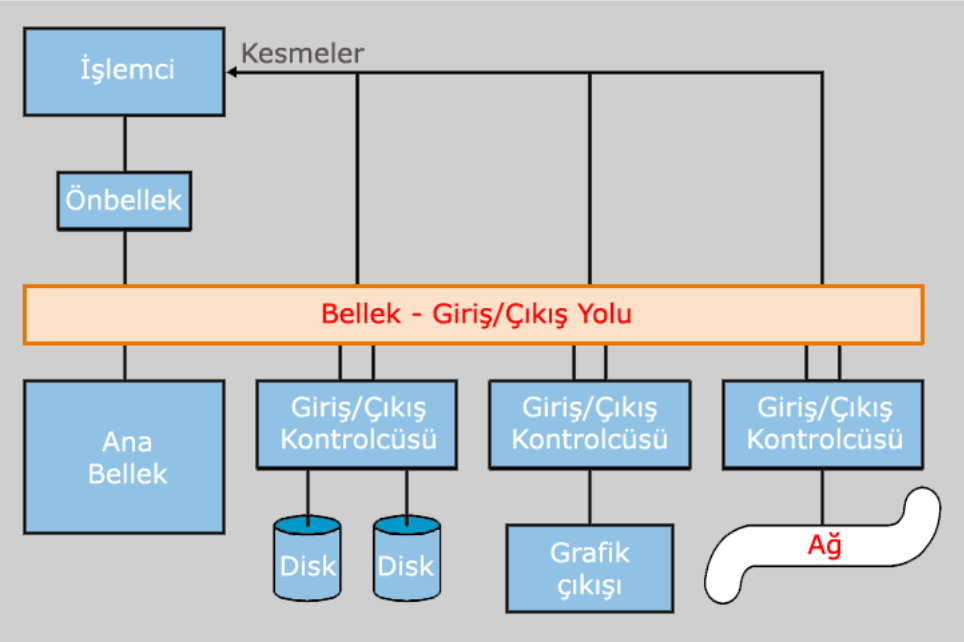


Önbellek

- Günümüz bilgisayarlarında, ön bellek kullanılarak bellekle tek yoldan iletişim ve buyrukla verinin aynı bellekte bulunma sorunu çözülmüştür.
- Ana bellek ile merkezi işlem birimi arasında görev yapan ve ana bellekten çok daha hızlı olan bir bellektir.
- Bu bellek birimi; işlem esnasında çok sık kullanılan bilgisayar talimatları ve geçici olarak tutulan bilgiler için bir “yaz-boz tahtası” olarak kullanılır.
- CPU ’nun ana bellekten veri alırken harcadığı zamanı azaltır; bu da bilgisayarı hızlandırır.

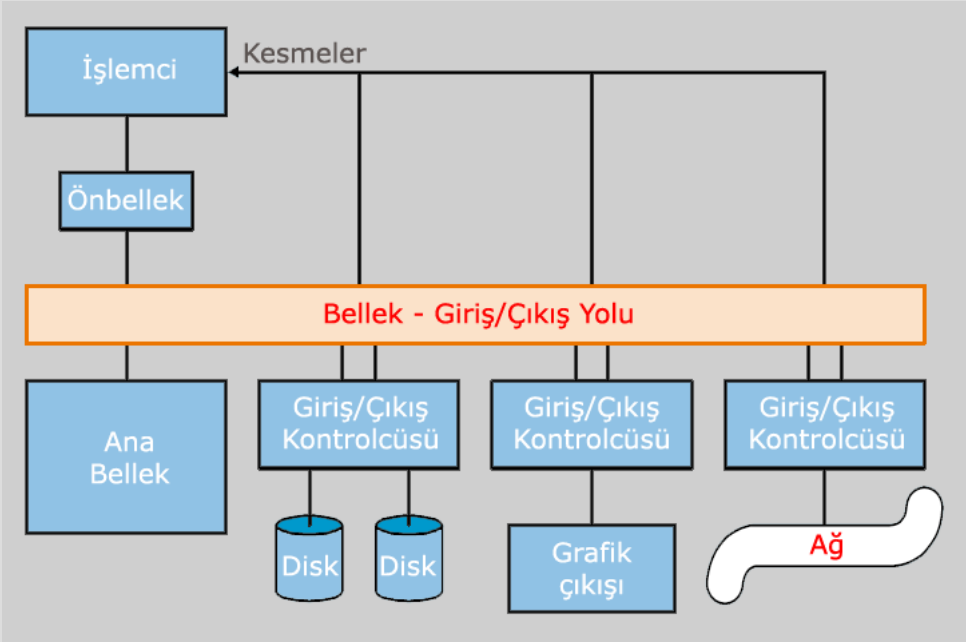
Temel bilgisayar yapısı Mikroişlemci

- Bilgisayarın kalbidir. İşlemcinin görevi, buyrukların bellekten getirilmesi, çözülmesi ve çalıştırılması, sonuçların gözlenmesi, program işlenirken diğer donanım birimlerinden gelen kesme (interrupt) isteklerine cevap vermesi gibi işlemlerdir.
  - Buyrukların yakalanması: Buyrukların programın saklandığı yerden alınması
  - Buyrukların çözülmesi: Gerekli işlemlerin ve buyruğun büyüklüğünün belirlenmesi
  - İşlenenlerin okunması: İşlem yapılacak verinin bulunması ve alınması
  - Yürütme: Sonucun ya da durumun hesaplanması
  - Sonucun saklanması: Sonuçların daha sonra yeniden kullanılmak üzere saklanması
  - Sonraki Buyruk: Bir sonraki buyruğun okunması için program sayacının değiştirilmesi



Temel bilgisayar yapısı
Mikroişlemci

- Günümüzde bu yapıya uygun ili bellek tipi vardır: ROM (Read Only Memory) ve RAM (Random Access Memory).
  - ROM üzerindeki bilgiler kalıcıdır. Mikrokodların bulunduğu bellektir. Firma tarafından yüklenir.
  - RAM üzerindeki bilgiler ise istenildiği zaman okunabilir ve yazılabilir. Elektrik kesintilerinde RAM üzerindeki tüm bilgiler silinir.
  - RAM bellekler de SRAM (Statik RAM) ve DRAM (Dynamic RAM) olmak üzere 2 çeşittir. SRAM çok pahalı ve hızlıdır, önbellek olarak kullanılırlar.



BIOS

- BIOS, (Basic Input-Output System) (Temel Giriş-Çıkış Sistemi).
  - ROM Bellek (Read Only Memory, tr: Salt Okunur Bellek) içinde yer alan bir tür yazılımdır.
  - Bilgisayar kapatılsada ROM içinde tutulur.
  - Bilgisayar açıldığı anda işlemciye tüm diğer donanımları sırasıyla tanıtır.
    - Donanımların temel iletişim protokollerini belirler.
    - İşletim sisteminin başlangıç öğelerinin Herhangi bir sürücüden (H.d.d, CD-ROM vb.) yüklenmesini sağlar.
    - İşletim sistemi çalışırken donanım ve işletim sistemi arasındaki ilişkileri düzenler.

Assembler

- Assembly dili programlar yazmak için kullanılan düşük seviyeli bir programlama dilidir.
- Bir assembly dil programı çevirici “Assembler” olarak adlandırılan faydalı bir program tarafından hedef bilgisayarın makine koduna çevrilir.
- Assembly insanlar tarafından anlaşılması zor olan makina dilinin sayısal ifadelerini, insanlar tarafından anlaşılarak programlanması daha kolay olan alfabetik ifadelerle değiştirerek düşük seviyede programlama için bir ortam oluşturur.
- Assembly kullanmanın amacı, ilk bilgisayarlarda yazılan programların daha az hata içermesi ve daha az zaman almasını sağlamaktır.
- Assembly dili programlarının yazılımında insan dostu sembollerin “mnemonics” kullanılması, daha fazla hataya yatkın ve zaman alıcı ilk bilgisayarlarda kullanılmış olan bir hedef bilgisayarının sayısal makine kodunda doğrudan programlama çalışmasının yerine geçmiştir.

Assembly (Hello World!)

```
dosseg
.model small
.stack 100h

.data
hello_message db 'Hello World!',0dh,0ah,'$'

.code
main  proc
      mov    ax,@data
      mov    ds,ax

      mov    ah,9
      mov    dx,offset hello_message
      int    21h

      mov    ax,4C00h
      int    21h
main  endp
end   main
```

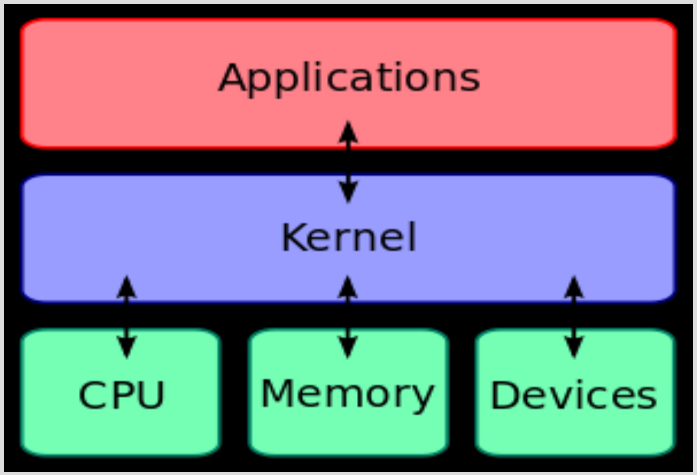
C (Hello World!)

```
#include<stdio.h>

int main()
{
   printf("Hello World\n");
   return 0;
}
```

Çekirdek (Kernel)

- İşletim sistemi çekirdeği, kısaca çekirdek (kernel), işletim sisteminin kalbidir.
  - Uygulamalar ve donanım seviyesindeki bilgi işlemleri arasında bir köprü görevi görür.
  - Çekirdeğin görevleri sistemin kaynaklarını yönetmeyi de kapsamaktadır.
  - İşletim sistemi görevleri, tasarımları ve uygulanmalarına göre farklı çekirdekler tarafından farklı şekillerde yapılır.
  - Sistem açılırken belleğe yüklenir ve sistem kapatılıncaya kadar bellekte kalır.
- Linux çekirdeği açık kaynak kodlu olduğu için üzerine yeni işletim sistemleri geliştirilebilir. (Pardus)



İşletim Sistemleri

- İşletim sistemi, bilgisayarda çalışan, bilgisayar donanım kaynaklarını yöneten ve çeşitli uygulama yazılımları için yaygın servisleri sağlayan bir yazılımlar bütünüdür.
- İşletim sistemleri sadece bilgisayar, video oyun konsolları, cep telefonları ve web sunucularında değil; arabalarda, beyaz eşyalarda hatta kol saatlerinin içinde bile yüklü olabilir.
- İşletim sistemleri işlevsellerinin genişliği ile değil, donanımı belli bir amaç doğrultusunda programlayabilme nitelikleriyle değerlendirilmelidir.

İşletim sistemleri

- Microsoft Windows
  - Kişisel Bilgisayarlar için: Windows 3.1, 95, Me, XP, Vista, 7, 8 Sunucular için ise Windows NT serisi (Bitti), Windows Server serisidir.
  - Microsoft Windows, 1981 yılında eski MS-DOS işletim sistemi üzerine IBM PC eklentisi yapılarak piyasaya sürülmüştür.
  - İlk olarak 1985 yılında yayımlanan Windows, kişisel bilgisayarların iş dünyasına hakim olmuştur.
  - Windows XP ile başlayarak tüm modern versiyonları Windows NT çekirdeği üzerine kurulmuştur.
- Mac OS X
  - Apple'ın 1984 yılında oluşturduğu ilk işletim sistemi olan Mac OS' in son sürümüdür.
  - Fakat Mac OS 8 ve 9 sürümlerinin aksine, Mac OS X, NeXT şirketi tarafından geliştirilmiş bir teknoloji üzerine kurulmuş UNIX tabanlı bir işletim sistemidir.

İşletim sistemleri

- Linux
  - Linux, Unix'e benzeyen ancak tamamen orijinal kod ile ücretsiz ve açık bir işletim sistemi yaratmaya çalışan bir programcı kitlesi işbirliğidir.
  - 1983 yılında Richard Stallman tarafından başlatılan projeye 1991 yılında Linus Torvalds çekirdeği tasarlayarak destek olmuştur. Bu nedenle Linux çekirdeği ve GNU yazılım koleksiyonunun kullanıldığı bu işletim sistemine Linux denir.
- Android
  - Android, Google, Open Handset Alliance ve özgür yazılım topluluğu tarafından geliştirilen, Linux tabanlı, mobil cihaz ve cep telefonları için geliştirilmekte olan, açık kaynak kodlu bir mobil işletim sistemidir.
- İOS
  - Apple'ın başlangıçta iPhone için geliştirdiği ancak daha sonra iPod Touch ve iPad'de de kullanılan mobil işletim sistemidir. Mac OS X'den türetilmiştir.

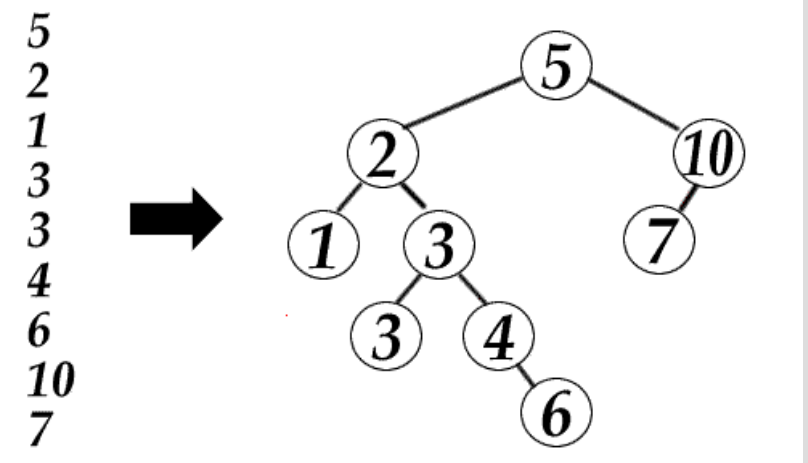
Veri Yapıları (Data Structures)

- Veri yapısı verinin veya bilginin bellekte tutulma şeklini veya düzenini gösterir.
- Tüm programlama dillerinin,
  - Tamsayı (int),
  - kesirli sayı (float),
  - karakter (char) ve
  - sözcük (string) saklanması için temel veri yapıları vardır.
- Programcı bu veri yapılarını, bunların bellekte nasıl saklandığı konusuyla ilgilenmeksizin bolca kullanır.
- Ayrıca, programcıda temel veri yapıları dışında yeni veri yapıları tanımlanması için programlama dillerinde bir çok özellik vardır. Örneğin:
  - Struct
  - Class

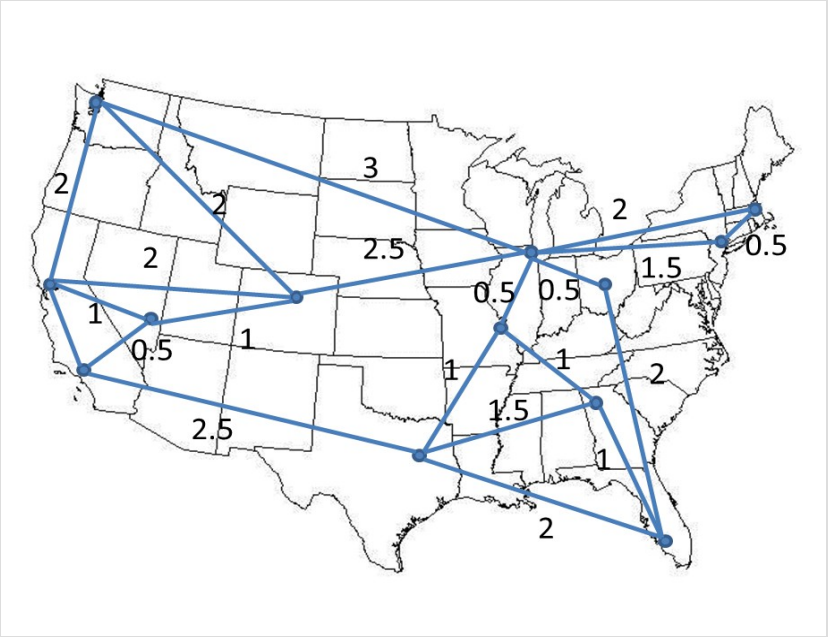
Veri Modeli (Data Model)

- Veri modeli, verilerin birbirleriyle ilişkisel veya sırasal durumunu gösterir.
- Bilgisayar ortamında uygulanacak tüm matematik ve mühendislik problemleri bir veri modeline yaklaştırılarak veya yeni veri modelleri tanımlaması yapılarak çözülebilmektedir.
- Örneğin,
  - Veriniz üzerinde hızlı arama yapmak istiyorsunuz. Yandaki gibi bir ağaç modeli kullanabilirsiniz.
  - Şehirler arasındaki mesafeleri hesaplayan bir yazılım geliştireceğiniz durumda ise graf modelini kullanabilirsiniz.

Binary Tree (İkili ağaç) Örneği



Graph Model (Graf Modeli) Örneği

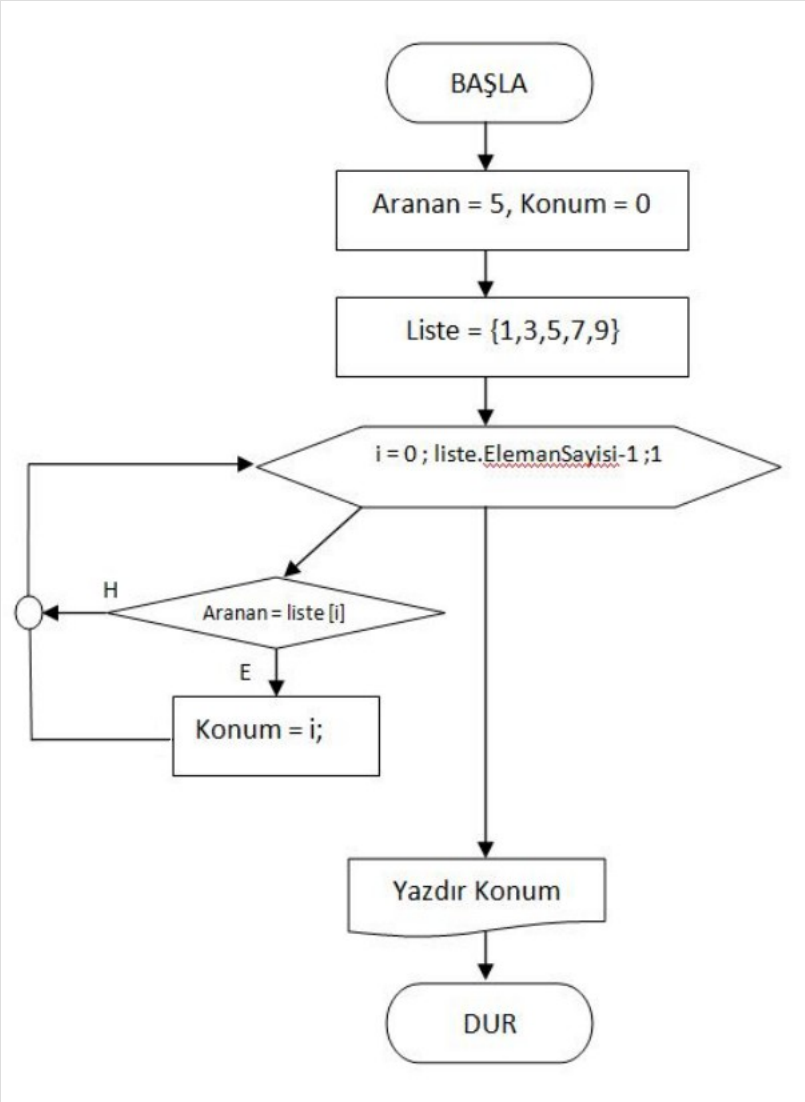


Algoritma

- Algoritma, matematikte ve bilgisayar biliminde bir işi yapmak için tanımlanan, bir başlangıç durumundan başladığında, açıkça belirlenmiş bir son durumunda sonlanan, sonlu işlemler kümesidir.
- Başka bir deyişle, belli bir problemi çözmek veya belirli bir amaca ulaşmak için çizilen yola algoritma denir.
- Bir algoritma buldum!!! Patent yapabilir miyim?
  - Algoritmalar, tek başlarına, genellikle patent verilebilir değildirler.
  - Yazılım patenti son derece tartışmalıdır ve algoritmaları içeren birçok eleştirilmiş patent vardır, özellikle veri sıkıştırma algoritmaları, Unisys' LZW patentinde olduğu gibi.

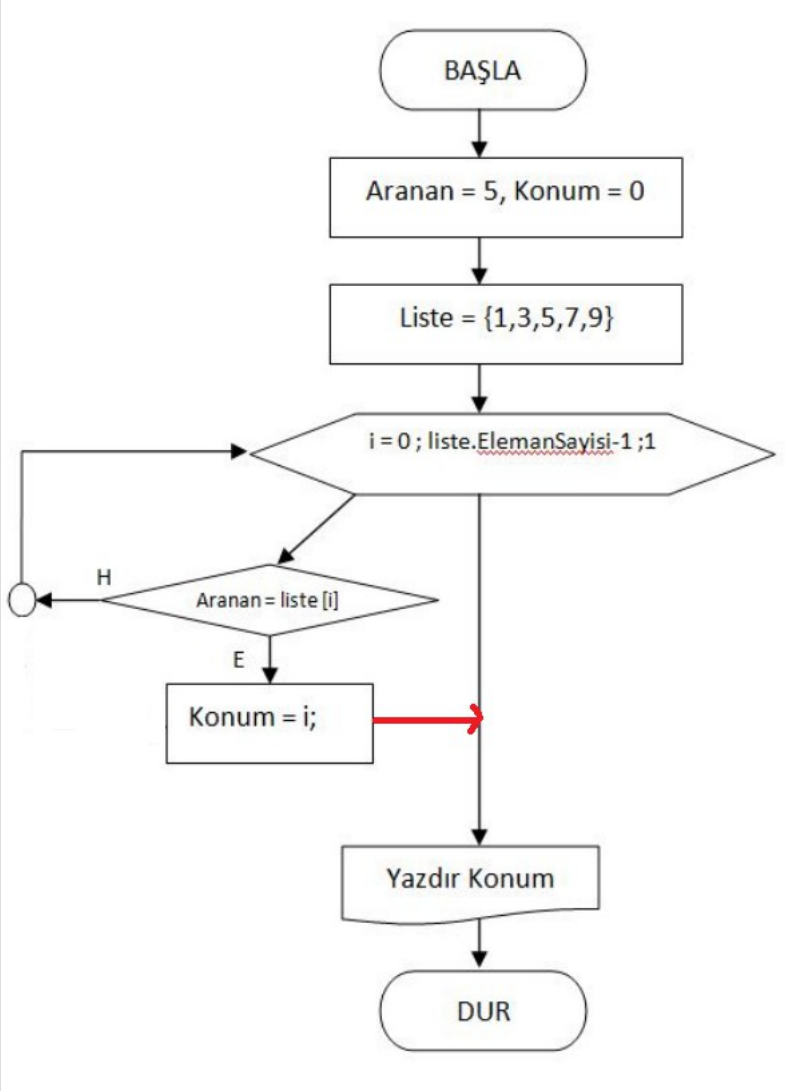
Bir algoritma örneği

- Yanda bir dizi içinde arama yapan bir algoritma vardır.
- Sizce algoritmada hata var mıdır?
- Daha iyi duruma getirilebilir mi?



Çözüm

- Sayı bulununca arama bitirilebilir. Bu sayede zaman kazanılmış olur.



Programın bellek kullanımı ve çalışma hızı

- Programın çalışma hızı karmaşıklıkla ifade edilir; bu kavram zaman birimiyle ifade edilmeyip doğrudan işlem adedi veya döngü sayısıyla ifade edilir.
- Çünkü, programın çalışma hızında zaman miktarı donanıma çok bağlıdır; dolayısıyla algoritmaları birbiriyle karşılaştırmak için zaman miktarını kullanmak gerçekçi olmayıp yanılgılara neden olmaktadır.
- Bunun yerine, ilgili algoritmanın bilgisayar donanımından bağımsız olarak kaç adet işlem veya döngüyle gerçekleştirilebileceği hesaplanır. Algoritma karmaşıklığı iki açıdan ele alınır.
  - Zaman karmaşıklığı (Time complexity), algoritmanın sonuca ulaşması için gerekli zaman hakkında bilgi verir.
  - Bellek (Alan) karmaşıklığı (memory(space) complexity) ise algoritmanın ihtiyaç duyacağı bellek miktarı hakkında bilgi verir.

Programlama Dilleri

- Programlama dili, yazılımcının bir algoritmayı ifade etmek amacıyla, bir bilgisayara ne yapmasını istediğini anlatmasının tek tipleştirilmiş yoludur.
- Programlama dilleri, yazılımcının bilgisayara hangi veri üzerinde işlem yapacağını, verinin nasıl depolanıp iletileceğini, hangi koşullarda hangi işlemlerin yapılacağını tam olarak anlatmasını sağlar.

Programlama Dilleri

- Programalama diliyle düz metin şekinde olan programa 'kaynak kodu' (source code) denir.
- Makine dilinde olan koda da "ikili kod" (binary code) denir.
- Programlama dili uygulamasında iki yaklaşım vardır:
  - Derleme (Compilation): programlama dilindeki ifadelerin çalıştırılmadan önce makine diline çevrilmesi, sonra da çalıştırılması anlamına gelir.
  - Yorumlama (Interpretation): programa dilindeki ifadeleri bir yandan okuyup bir yandan makine diline çevrilmesi anlamına gelir.
  - Derleme'nin faydası daha hızlı olmasıdır. Çünkü makine diline çevirme sadece ve sadece bir kere yapılmaktadır.
  - Derlemenin zararı da, programdaki her değişiklikte önce derlemeyi sonra çalıştırmayı gerekli kılmasıdır. Sık değişiklik yapılan durumlarda bu, programcı için ciddi bir sorundur.
  - Perl, Basic ve PHP gibi diller yorumlamalı iken C ve Pascal gibi diller derlemeli dillerdir.

Programlama dilleri

- Acaba Java ve C# yorumlamalı mı yoksa derlemeli bir dil midir?
  - Bu diller hem 'derlemeli' bir dildir, hem de 'yorumlamalı'.
  - Bu programlama dilleriyle yazılmış kaynak kodu, sanal bir işlemcinin anlayabileceği makine koduna çevrilir. Bu kod gerçek bir makine kodu değildir.
  - Sanal işlemci Java veya C# çatısını (framework) kullanarak makine koduna çevirip çalıştırır.
  - Eğer işletim sisteminde framework yok ise uygulama çalışmaz. Diğer taraftan C uygulamasının çalışması için uygun bir işletim sistemi yeterlidir.
  - Hatta Framework'ün versiyonu bile çalışmayı etkileyebilir.

IDE(Entegre Geliştirme Ortamı)
Integrated Development Environment

- Bilgisayar programcılarının hızlı ve rahat bir şekilde yazılım geliştirebilmesini amaçlayan, geliştirme sürecini organize edebilen birçok araç ile birlikte geliştirme sürecinin verimli kullanılmasına katkıda bulunan araçların tamamını içerisinde barındıran bir yazılım türüdür.
- Tümleşik geliştirme ortamlarında olması gerekli en temel özellikler:
  - Programlama diline göre sözdizimi renklendirmesi yapabilen kod yazım editörü.
  - Kod dosyalarının hiyerarşik olarak görülebilmesi amacıyla hazırlanmış gerçek zamanlı bir dizelge.
  - Tümleşik bir derleyici, yorumlayıcı ve hata ayıklayıcı.
  - Yazılımın derlenmesi, bağlanması, çalışmaya tümüyle hazır hale gelmesi ve daha birçok ek işi otomatik olarak yapabilmek amacıyla küçük inşa araçları.
- En bilinen tümleşik geliştirme ortamlarına örnek olarak Eclipse, Microsoft Visual Studio, Dev-C++, NetBeans gibi ortamlar verilebilir.

Veritabanı ve SQL

- Veritabanı, bilgilerin belirli bir disiplin altında saklanması ve gerektiğinde hızlı bir şekilde aranıp bulunması/sorgulanması için geliştirilmiş bir saklama/sorgulama sistemidir.
- SQL (Structured Query Language) ise bir sorgulama dilidir; veritabanı ile etkileşimin kotarılması, veritabanı veri yapısının tanımlanması, veritabanına veri yazma/sorgulama işlemlerinin yapılmasını sağlar.
- VeriTabanı Yönetim Sistemi, bu işlemleri yönetir.
- VTYS amacı kullanıcının belirlediği verileri hızlı ve doğru bir şekilde fiziksel ortama yazmak ve aynı ortamdan hızlı ve doğru bir şekilde tekrar elde etmektir.

Veritabanı ve SQL

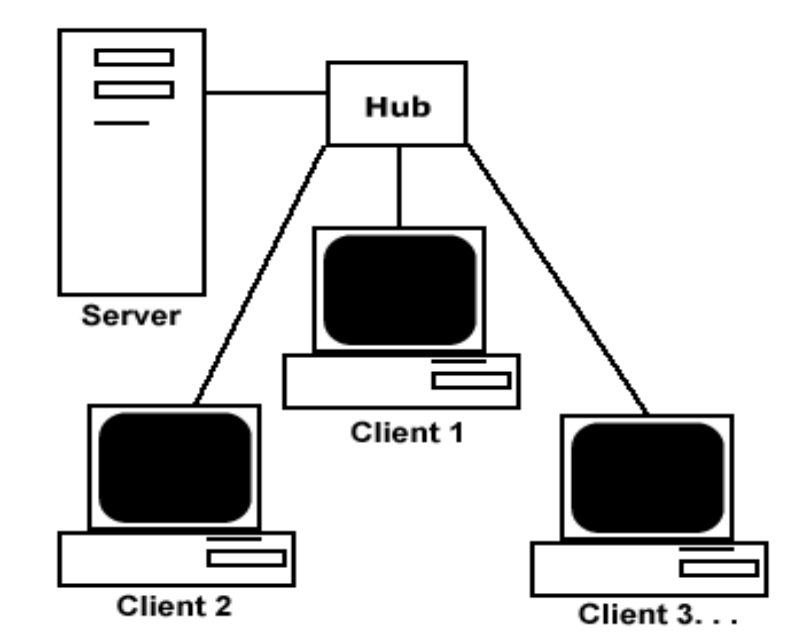
- SQL standart bir sorgulama dili olup Access, Microsoft SQL Server, MySQL, ORACLE, SYBASE, Ingres gibi birçok VTYS tarafından desteklenmektedir;
- ORACLE, SYBASE ve Ingres profesyonel anlamda veritabanı yönetim sistemleridir; hem tasarım ortamı sunarlar hem de uygulamanın etkin bir şekilde çalışmasını sağlarlar.
- MySQL açık kaynak kodlu bir VTYS'dir. Ancak Oracle tarafından satın alınmıştır.

Parçala Fethet Yöntemi (Divide and Conquer)

- Bu yöntem algoritma analizinde çok kullanılan, bir algoritmayı tahlil etmek veya yeni bir algoritma oluşturmak için kullanılan yaklaşımlardan birisidir.
- Bu yaklaşıma göre problem ufak ve çözülmesi nispeten daha kolay olan parçalara bölünür. Her parça ayrı ayrı çözüldükten sonra sonuçlar birleştirilerek genel problemin çözümü elde edilir.

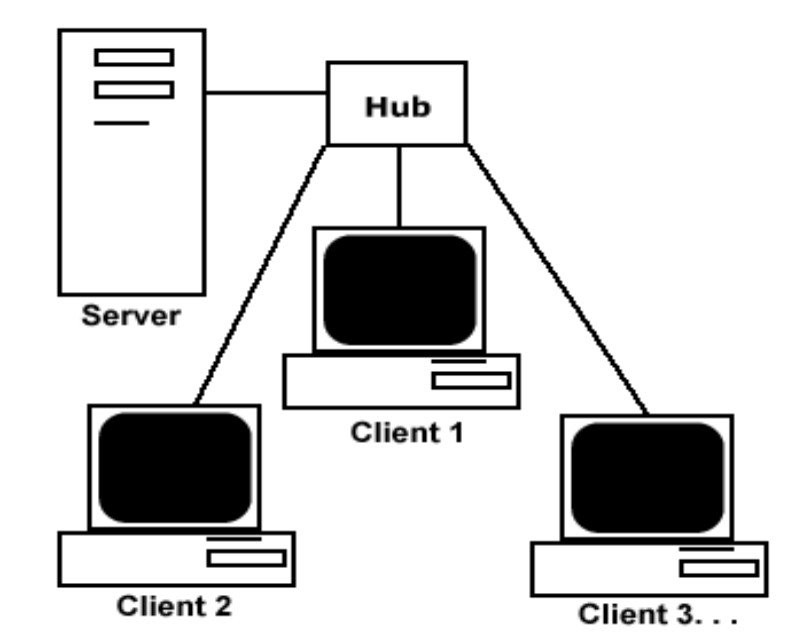
İstemci / Sunucu

- İstemciyi, sunucudan ayıran bir ağ mimarisidir. Ağdaki tüm kaynaklar server'a bağlıdır ve ağ kaynaklarının tek sorumlusu server'dır.
- Eğer server isterse, bu sorumluluğu ağdaki diğer makinalara verebilir veya hepsini kendi kullanır.
- Ağdaki diğer makinalar ancak server üzerinden ağa bağlanabilir ve server'ın verdiği erişim izinleri dahilinde ağ kaynaklarından faydalanabilir.



İstemci / Sunucu

- Bu fikrin pek çok çeşitli uygulaması olmasına karşın, en güzel örneği İnternet üzerindeki Web sayfalarıdır. Bir web sayfası incelenirken, bilgisayar ve web tarayıcısı istemci olarak adlandırılır.
- Web sayfasını depolayan gelişmiş bilgisayarlar, veritabanları ve uygulamalar da sunucu olarak adlandırılır.
- Web tarayıcısı, web sitesinden bir istekte bulunur ve sunucu istenen bilgileri toplar ve onu bir web sitesi şekline getirerek web tarayıcısına geri yollar, kullanıcılar da ekranda web sitesini görmüş olur.



Kıyaslama Örneği

- Yüz tanıma uygulaması yazsanız.
  - CPU ve RAM kullanma miktarı.
  - Test verisi üzerinde yüz tanıma sayısı
    - Örneğin 100 tane yüz göster, kaçını tanıyabildi.
  - Işık etkisi (Işıktan ne kadar etkileniyor, bu probleme ait özel bir kıyaslama)
  - Kamera etkisi (Dış etmen, acaba kamera markası, türü yüz tanıma işlemini etkiler mi?)

Kıyaslama (Benchmarking)

- Kıyaslama, aynı işi yapan iki programın veya yazılımın evrensel anlamda örnek veriler üzerinde çalıştırılarak başarımlarını karşılaştırmaktır; kıyaslama donanım için de yapılabilir. Bir yazılımda kıyaslama kriterleri:
  - Kullandığı CPU miktarı
  - Kullandığı RAM miktarı
  - Kullandığı Disk miktarı
  - Yanıt verme süresi
  - Aynı anda yapabileceği maksimum işlem sayısı
  - Ağın band genişliğine etkisi

Bir kıyaslama örneği

- Sıkıştırma programlarından WinRAR mı WinZip mi iyidir?
  - Hangisi daha iyi sıkıştırıyor. (Disk)
  - Hangisi daha kısa sürede sıkıştırıyor. (Yanıt verme süresi)
  - Hangisi daha sistem kaynaklarını (CPU, RAM) kullanıyor.
  - 5'ten fazla sıkıştırma işlemi yapınca yukarıdaki üç testi tekrarla.
  - Aynı işlemler açma işlemi içinde tekrarlanır.