

- 1. Bilgisayar Mühendisliğine Giriş
 - 1.1. Temel Kavramlar
 - 1.2. Algoritma ve Akış Şemaları
 - 1.2.1. Algoritmada Olması Gereken Özellikler
 - 1.2.2. Akış Şeması Simgeleri
 - 1.3. Programlama Dilleri
 - 1.3.1. Yazılım Geliştirme Süreci
 - 1.3.2. Programlama Dilinin Önemi
 - 1.3.3. Nesneye Yönelik Programlama
 - 1.3.4. Programlama Ortamı
 - 1.4. İşletim Sistemleri
 - 1.4.1. Bilgisayar Sistemi Bileşenleri
 - 1.4.2. İşletim Sistemlerinin Temel İşlevleri
 - 1.4.3. Sistem Performansı
 - 1.4.4. Çekirdek Sistem
 - 1.4.5. Proses Yönetimi
 - 1.4.6. Bellek Yönetimi
- 2. Bilgisayar Mimarisi ve Tasarım
- 3. Bilgisayar Ağları
 - 3.1. Ağ Yapısına Giriş
 - 3.1.1. Veri İletimi
 - 3.1.2. Veri İletim Tipleri
 - 3.1.3. Veri İletim Yöntemleri
 - 3.1.4. Veri İletim Ortamları
 - 3.1.5. Haberleşme Yöntemleri
 - 3.1.6. Ağ Mimarileri
 - 3.1.7. Sunucu Türleri
 - 3.1.8. Coğrafi Açıdan Ağ Türleri
 - 3.1.9. Topolojiler
 - 3.2. Ağ Bağlantı Cihazları
 - 3.3. İnternet ve TCP/IP
 - 3.3.1. Katmanlar
 - 3.3.2. TCP
 - 3.3.3. UDP
 - 3.3.4. IP
 - 3.4. Ağ Güvenliği
 - 3.4.1. Bilgi Güvenliği
 - 3.4.2. IPsec
 - 3.4.3. Saldırı Tespit Sistemleri (IDS-Intrusion Detection Systems)
 - 3.4.4. Şifreleme
- 4. Veri Yapıları ve Algoritmalar
 - 4.1. Bilgisayar Yazılım Dünyası
 - 4.2. Veri Yapıları ve Veri Modelleri
 - 4.2.1. Veri Modeli Türleri
 - 4.3. Sıralama Algoritmaları
 - 4.4. Arama Algoritmaları

- 5. Java ile Programlama
 - 5.1. Veri Tipleri ve Değişkenler
- 6. İşletim Sistemleri
 - 6.1. Bilgisayarlar ve İşletim Sistemleri
 - 6.2. Bilgisayar Mimarileri
 - 6.2.1. CISC (complex instruction set computer) Mimarisi
 - 6.2.2. RISC (reduced instruction set computer) Mimarisi
 - 6.2.3. İşletim Sistemleri Ağ Özellikleri
- 7. Veri Tabanı Yönetim Sistemleri
 - 7.1. Veritabanı Yönetim Sistemleri
 - 7.1.1. Kavramlar
 - 7.1.2. Uygulamalar
 - 7.1.3. Yaklaşımlar
 - 7.2. Veri Modelleri
- 8. Yazılım Mühendisliği
- 9. Web Programlama
 - 9.1. HTML5
 - 9.1.1. Neden Önemli
 - 9.1.2. Formlar
 - 9.1.2.1. Özellikleri
 - 9.1.3. Elemanlar
 - 9.1.3.1. Input Types
 - 9.2. CSS3
 - 9.2.1. CSS Pseudo Classes & Elements
 - 9.2.2. EcmaScript 6
- 10. Yazılım Proje Yönetimi

1. Bilgisayar Mühendisliğine Giriş

1.1. Temel Kavramlar

Birleşik mantık devreleri VE, VEYA gibi kapılardan oluşur ve devrenin çıktısı sadece o anki girdilere bağlıdır. **Sıralı/Ardışıl mantık devreleri (flip-flop)** hem o andaki girdilere hem de bir önceki çıktıya bağlıdır. Bu devreler saat darbeleri ile çalışır.

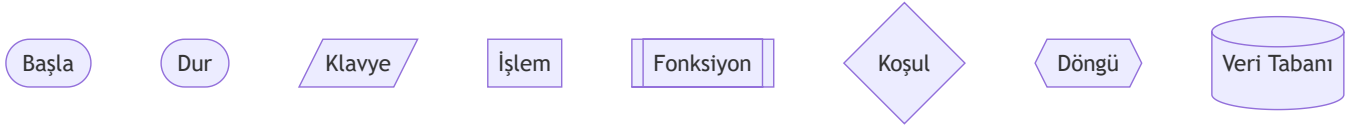
Assembly diliyle yazılmış programların çalışması için makina koduna dönüştürülmesi gerekir. Her işlemci ailesinin kendine has makina kodu ve assembly dili vardır.

1.2. Algoritma ve Akış Şemaları

1.2.1. Algoritmada Olması Gereken Özellikler

1. Etkin ve Genel olma
2. Sonlu olma
3. Yanılmazlık
4. Giriş/Çıkış tanımlı olma
5. Başarım (Bellek gereksinimi ve çalışma süresi arasındaki denge)

1.2.2. Akış Şeması Simgeleri



1.3. Programlama Dilleri

1.3.1. Yazılım Geliştirme Süreci

1. Gereksinim analizi
2. Yazılım Tasarımı
3. Kodlama
4. Sertifikasyon
5. Bakım

1.3.2. Programlama Dilinin Önemi

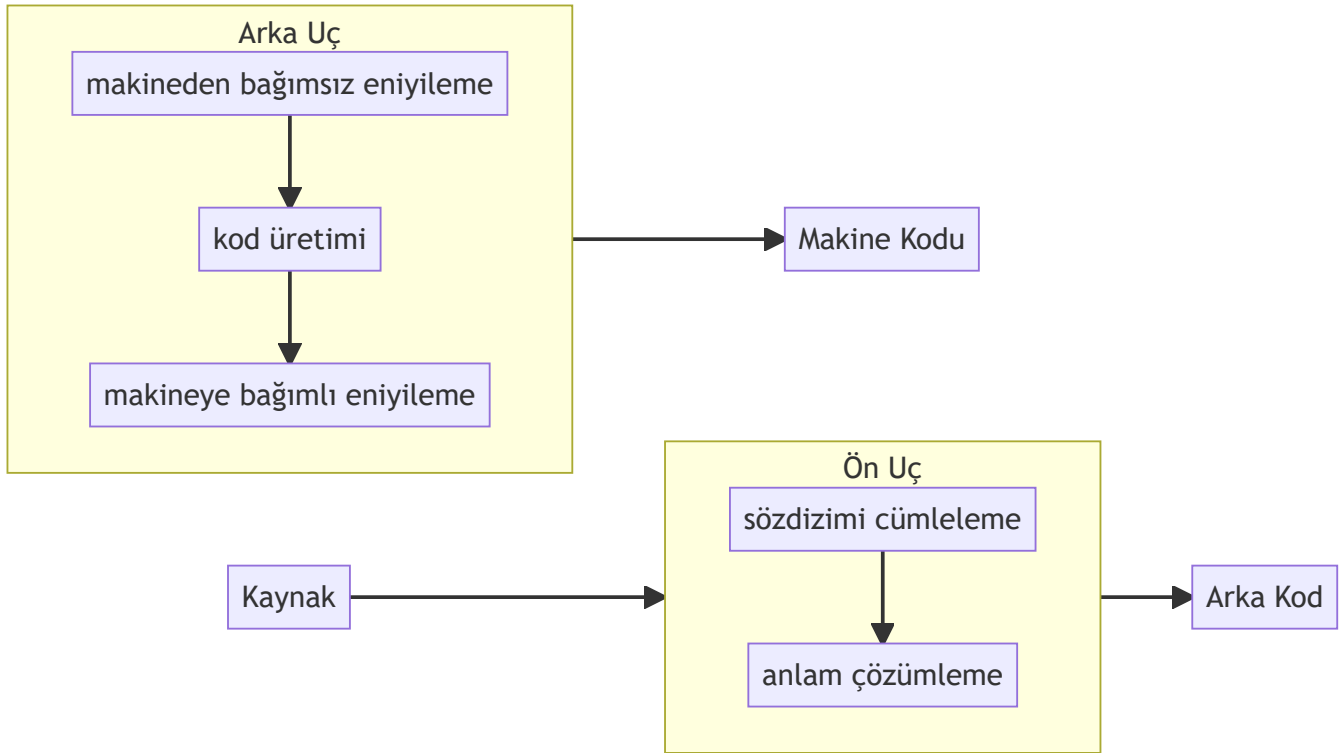
1. Güvenilir olmalı
 1. Yazılabilirlik
 2. Okunabilirlik
 3. Sıra dışı durumları karşılayabilme
2. Bakıma elverişli olmalı
3. Verimli çalışmalı

1.3.3. Nesneye Yönelik Programlama

1. **Veri soyutlama:** Yeni veri türlerini modelleyen sınıflar yaratması
2. **Kalıtım:** Sınıfları genişleterek ve özelleştirerek yeni sınıflar türetilmesi
3. **Çok biçimlilik:** Aynı isimdeki işlemlerin değişik nesne sınıfları için farklı algılanması

1.3.4. Programlama Ortamı

1. Editör
2. Derleyici (Compiler)
3. Kütüphane (Librarian): Nesne dosyaları
4. Bağlayıcı (Linker): Tüm nesne dosyalarını birleştirerek tek bir yürütülebilir dosya haline getirir
5. Yükleyici (Loader): Yürütülebilir dosyayı diskten belleğe kopyalar
6. Hata ayıklayıcı
7. Yorumlayıcı (Interpreter): Bir programın kaynak kodunu doğrudan satır satır yürüten program



1.4. İşletim Sistemleri

1.4.1. Bilgisayar Sistemi Bileşenleri

1. Donanım
2. İşletim sistemi
3. Sistem ve uygulama programları
4. Kullanıcılar

1.4.2. İşletim Sistemlerinin Temel İşlevleri

1. Yazılım donanım bütünlüğünün sağlanması
2. Kaynakların yönetimi
 1. Görev (task) > İş (job)
3. Kullanıcı ile sistem arasındaki ilişki, uyum ve düzen
 1. Adanmış işlem (dedicated)
 2. Toplu işlem (batch)
 3. Etkileşimli işlem (interactive, time sharing)

1.4.3. Sistem Performansı

1. **Verimlilik:** Sistem kaynaklarının düşük kullanımı
2. **Güvenilirlik:** Donanım ve yazılım yüzünden doğacak sorunları sezebilmeli ve önlemeli
3. **Koruyuculuk:** Bir kullanıcının yaptığı hatadan diğer kullanıcı etkilenmemeli
4. **Sezdiricilik**
5. **Elverişlilik:** Kullanıcı sistemin kaynaklarını kendi isteğine göre değil sistemin izin verdiği ölçüde kullanmalı

1.4.4. Çekirdek Sistem

1. İşlemciye prosesleri atamak
2. Kesmeleri yönetmek
3. Prosesler arasında iletişimi sağlamak

1.4.5. Proses Yönetimi

Proses: kendi veri şablonu olan ve kendi başına bütünlüğü olan kod parçası

1. Çalışıyor (running)
2. Askıda (blocked)
3. Hazır (ready)

Zaman Çizelgeleyici (scheduler): İşlemciye atanacak olan prosesi ve prosesin hangi koşullar altında işlemciyi kullanacağını belirler.

1. Proses kuyruğundan yürütülecek prosesi seçer
2. Prosesin kullanım zaman dilimini ayarlar

Deadlock: proseslerin hiç bir zaman ele geçiremeyecekleri bir birim veya kaynağa ihtiyaç duymaları sebebiyle sürekli askıda kalması

Semafor: Bir çeşit değişken. Signal ve Wait'tir.

1.4.6. Bellek Yönetimi

Bellek	Statik/Dinamik	Gerçek/Sanal
Tekli kesintisiz	S	G
Bölümlenmiş	S	G
Yer değiştirilebilir bölümlenmiş	D	G
Sayfalı	D	G
İsteğe bağlı sayfalı	D	S
Dilimli	D	S
Dilimli ve isteğe bağlı sayfalı	D	S

2. Bilgisayar Mimarisi ve Tasarım

3. Bilgisayar Ağları

3.1. Ağ Yapısına Giriş

3.1.1. Veri İletimi

1. Paralel
2. Seri
3. Asenkron: *START, STOP ve Eşik Biti vardır.*
4. Senkron: *Saat bilgisi*

Bilgisayar ağları üzerindeki iletişim **seri** iletişimidir.

3.1.2. Veri İletim Tipleri

1. BaseBand
2. BroadBand

3.1.3. Veri İletim Yöntemleri

1. Simplex (Tek Yön)
2. Half Duplex (Yarı Çift Yön)
3. Full Duplex

3.1.4. Veri İletim Ortamları

1. Koaksiyel Kablo
 1. **10Base5-Thicknet:** 10Mbps, BaseBand, 500m, 10mm
 1. Yellow ethernet
 2. 2.5m'de bir siyah bant
 3. Transceiver, AUI bağlantı
 2. **10Base2-Thinnet:** 10Mbps, BaseBand, 185m, 4mm
 1. Black ethernet
 2. 0.5m'de bir ve en fazla 30 cihaz
 3. BNC bağlantı, sonlandırıcı
2. Çift Burgulu (Twisted Pair) Kablo
 1. Kablo tipi
 1. UTP (Unshielded) *100m, 4-6 burğu**
 2. STP (Shielded) *Topraklanmalı*
 3. FTP (Foiled)
 4. S/FTP
 2. Konnektör
 1. RJ-11: *4 pin*
 2. RJ-45: *8 pin; T-568A, T-568B*
 3. Bağlama şekli
 1. Düz (Straight-through)
 1. Hub/Switch ve PC
 2. Switch ve Router
 3. Hub ve Hub *Uplink*
 4. Switch ve Switch
 5. Yazıcı ve Hub/Switch
 6. PC ve Kablo/DSL Modem
 2. Çapraz (Cross-through)
 1. Hub ve Hub

2. Pc ve Pc
3. Switch ve Switch
4. Switch ve Router
5. Yazıcı ve Hub/Switch

3. Fiber

1. Fiber Tipine göre
 1. Tek Mod (Single Mode Fiber - SMF): *3km*
 2. Çok Mod (Multi Mode Fiber - MMF): *Dereceli, Kademeli, daha kısa mesafeler*
2. Kablo Tipine göre
 1. Loose Tube
 2. Tight Buffer
3. Konnektör
 1. Düşük Insertion Loss: *Konnektörün bağlantısından dolayı sinyal gücü kaybı*
 2. Yüksek Return Loss: *Konnektörden kaynaklı geri yansıma*

3.1.5. Haberleşme Yöntemleri

1. Yayın (Broadcast)
 1. Unicast
 2. Multicast
 3. Broadcast
2. Anahtarlama (Switching)
 1. Devre Anahtarlama
 2. Paket anahtarlama (Routing)
 1. Statik yönlendirme
 1. En kısa yol
 1. Dijkstra
 2. Bellman-Ford
 2. Taşma: *gelen paketin kopyası geldiği düğüm hariç her yere*
 3. Rastgele: *ilgili yol veri hızı/tüm yol hızı*
 4. Akış durumu
 2. Dinamik yönlendirme
 1. Uzaklık vektörü: *Komşulara gönderir*
 2. Hat durumu: *Herkese gönderir*

3.1.6. Ağ Mimarileri

1. Peer-to-Peer
2. Client-Server
 1. 2-tier
 2. 3-tier

3.1.7. Sunucu Türleri

1. File server
2. Database Server
3. Transaction Server
4. Web Server

5. ISA (Internet Security and Acceleration Server) / TMG (Threat Management Gateway)
6. Proxy Server

3.1.8. Coğrafi Açıdan Ağ Türleri

1. PAN (Personal)
2. LAN
3. MAN (Metropolitan)
4. WAN
5. VPN
6. CAN (Controller)
7. SAN (Storage)

3.1.9. Topolojiler

1. Fiziksel
 1. Ortak yol (Bus)
 2. Halka (Ring): *MAU, token*
 3. Yıldız
 4. Örgü (Mesh)
 5. Ağaç (Hierarchical)
2. Mantıksal
 1. Yayın (Broadcast)
 2. Jetonlu Geçiş (Token Passing)

3.2. Ağ Bağlantı Cihazları

1. Ağ Arayüz Kartı (NIC-Network Interface Card): *MAC 48 bit*
2. HUB: *Gelen veriyi iletir*
 1. Aktif
 2. Pasif
3. Switch: *Gelen veriyi ilgilisine iletir*
 1. Cut Throught
 2. Store and Forward
4. Repeater
5. Bridge: *Ethernet-Token Ring*
6. Router: *Akıllı switch*
7. Gateway: *Farklı protokolleri bağlar*
8. Transceiver
9. Firewall
10. Access Point: *Kablosuza dönüşüm*
11. Modem

3.3. İnternet ve TCP/IP

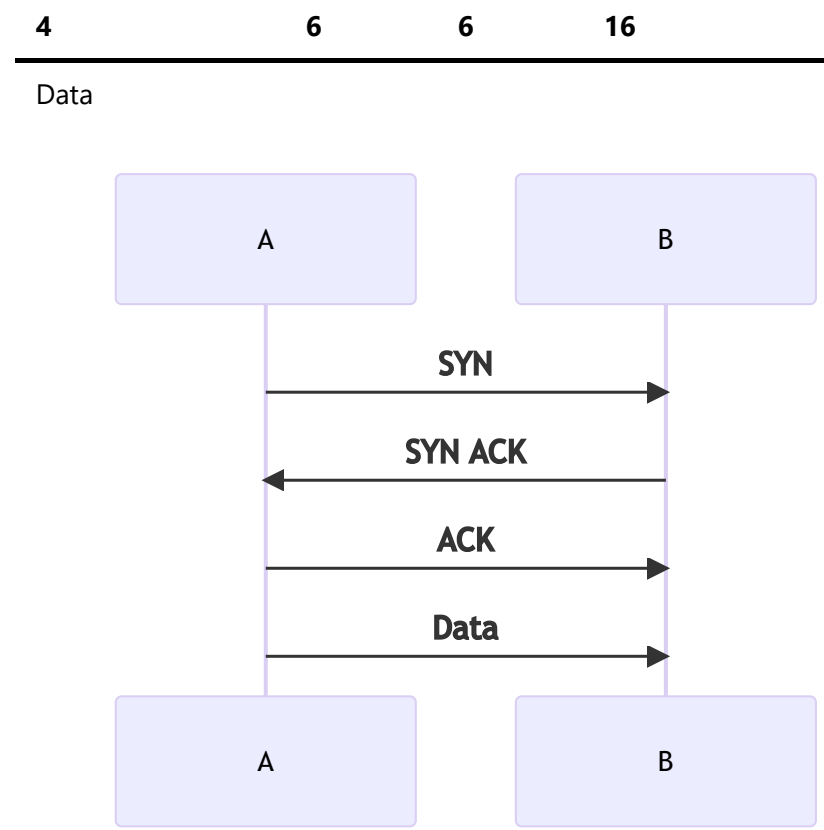
3.3.1. Katmanlar

1. Uygulama Katmanı

1. DHCP (Dynamic Host Configuration)
 2. DNS
 3. HTTP/HTTPS
 4. FTP/TFTP (Trivial)
 5. SMTP (Simple Mail Transfer)
 6. POP (Post Office)
 7. IMAP (Internet Message Access)
 8. IRC (Internet Relay Chat)
 9. SOCKS (SOCKEt Secure)
 10. TLS/SSL (Transport Layer Security/Secure Socket Layer)
 11. SSH (Secure Shell)
 12. TELNET
 13. SNMP (Simple Network Management)
 14. NNTP (Network News Transfer)
 15. RMON (Remote Monitoring)
 16. RPC (Remote Procedure Call)
 17. RTP (Remote Transport)
 18. LDAP
 19. NTP (Network Time)
2. Taşıma Katmanı
 1. TCP
 2. UDP
 3. İnternet Katmanı
 1. IP
 2. ICMP (Internet Control Message): *ping, traceroute*
 3. IPsec
 4. RIP (Routing Information)
 4. Uygulama Katmanı
 1. ARP/InARP (Address Resulation)
 2. RARP (Reverse)
 3. NDP (Neighbour Discovery): *IPv6*
 4. PPP (point-to-point)
 5. SLIP (Serial Line Interface)

3.3.2. TCP

4	6	6	16
Source Port		Destination Port	
Sequence Number			
Ack Number			
Header Length	Reserved	TCP Flag	Window Size
Checksum		Urgent Pointer	
Options			



3.3.3. UDP

16	16
Source Port	Destination Port
Length	Checksum
Data	

	TCP	UDP
Uygulama Katmanı	Stream	Message
Taşıma Katmanı	Segment	Packet
İnternet Katmanı	Datagram	Datagram
Fiziksel Katman	Frame	Frame

3.3.4. IP

4	4	6	2	3	13
Version	Internet Header Length	DSCP	ECN	Total Length	
Identification				Flags	FO
Time to Live		Protocol		Checksum	
Source IP Address					

4	4	6	2	3	13
Destination IP Adress					
Options (if header length > 5)					
Data					

Differentiated Services Code Point: Type of service

Explicit Congestion Notification: Ağ tıkanıklığı

Identification: veri parçalandığında

Flags: DF(Don't fragment): 1,0 ; MF(More Fragments): Arkadan başka veri geliyor mu

Fragment Offset: parçanın bütündeki yeri

Time to Live: Jump count

Protocol: TCP, UDP

3.4. Ağ Güvenliği

3.4.1. Bilgi Güvenliği

1. Gizlilik
2. Bütünlük
3. Doğruluk
4. Ulaşılabilirlik
5. Kanıt

3.4.2. IPsec

1. Protokol
 1. Authentication Header
 2. Encapsulation Security Payload
 3. AH + ESP
2. Bütünlük
 1. MD5: 128 bit anahtar, 512bitlik şifreler
 2. SHA1: 160 bit anahtar, 512bitlik şifreler
3. Kimlik Doğrulama
 1. Preshared Key
 2. Kerberos (Windows AD)
 3. Sertifika Yetkilisi
4. Gizlilik
 1. DES (Data Encryption Standart): 64 bit anahtar, 64bitlik şifreler
 2. 3DES
 3. AES (Advanced Encryption Standart): 128, 192 ve 256 bit anahtar
 4. Seal

3.4.3. Saldırı Tespit Sistemleri (IDS-Intrusion Detection Systems)

1. NIDS (Network)
2. HIDS (Host)
3. SIDS (Stack)

3.4.4. Şifreleme

1. Simetrik
 1. DES
 2. 3DES
 3. AES
2. Asimetrik
 1. Diffie-Hellman
 2. RSA
3. Anahtarsız
 1. MD5
 2. SHA

4. Veri Yapıları ve Algoritmalar

4.1. Bilgisayar Yazılım Dünyası

Genel olarak çalışma hızı ile bellek ihtiyacı ters orantılıdır. Hızlı algoritma yavaş olana göre daha fazla bellek ihtiyacı duyar. Algoritma karmaşıklığı iki açıdan ele alınır birisi **zaman karmaşıklığı** diğeri **bellek karmaşıklığı**dır.

4.2. Veri Yapıları ve Veri Modelleri

4.2.1. Veri Modeli Türleri

1. Bağlantı liste veri modeli
 - Ardışıl erişim
 - Dinamik yaklaşım
 - Arama maliyeti $O(n)$
 - Ekleme maliyeti $O(1)$
 - Silme maliyeti $O(1)$ veya $O(n)$
 - Esnek bellek kullanımı
2. Ağaç veri modeli
 - Hızlı arama
 - Algoritma karmaşıklığı logaritmik
 - Farklı problemlere model olması
 - Dinamik bellek kullanımı
 - Tasarım esnekliği
 - Kod ağacı
3. Graf veri modeli
 - Modelleme esnekliği

- En kısa yol
 - Greedy yöntemi
 - Kruskal algoritması
 - Graf renklendirme
 - Yol ağacı
 - DFS ve BFS yaklaşımları
 - Sosyal ilişki grafiği
4. Durum makinası veri modeli
- Davranış modelleme
 - Ardışıl yaklaşım
 - Desen uyuşması
 - Sözce arama
 - Gramer çözümü
 - Turing makinası
5. Veri tabanında ilişkisel veri modeli
- Bilgilerin düzenli saklanması
 - Hızlı arama/sorgulama
 - Veriler arasında ilişki oluşturulması
 - İnternet tarayıcı ortamında bilgi sorgulama
 - Koruma
 - Verinin arşivlenmesi
6. Ağ veri modeli
- Ağ üzerinden karşılıklı çalışma
 - IP Paketi
 - İnternet
 - Protokol kümeleri
 - Ağ üzerinden veri aktarımı
 - TCP/IP protokol kümesi

4.3. Sıralama Algoritmaları

1. **Araya sokma:** Sıralama süreci boyunca dizi iki parça gibi düşünülür. Sıralı olan ön taraf ve henüz sıralanmamış arka taraf
2. **Seçmeli sıralama:** Önce dizinin ilk elemanı alınır ve daha sonra küme içerisindeki kalan elemanların en küçüğü arınır. Bulunduğu zaman ilk eleman ile karşılıklı yer değiştirilir. En olumlu yanı eğer herhangi bir eleman gerçek yerinde ise yer değiştirme işlemi yapılmamasıdır.
3. **Kabarcık sıralaması:** Sıralanacak elemanlar üzerinde bir yönden diğer yöne doğru ilerlerken komşu iki elemanın yer değiştirmesi işlemine dayanır. Tasarımı basit, ancak etkin bir algoritma değildir.
4. **Birleşmeli sıralama:** Böl ve yönet yaklaşımına dayanır. Rekürsif tasarlanması doğasına uygundur.
5. **Kümeleme sıralaması:** İkili Kümeleme ağacı kurulmasına dayanır. İki yolu vardır birisi bir başka dizi gerektirmeksizin sırasız elemanların bulunduğu dizi üzerinde çalışır diğeri sıralı elemanların olacağı yeni bir dizi gerektirir.
6. **Hızlı sıralama:** Böl ve yönet yaklaşımına dayanır.

4.4. Arama Algoritmaları

1. **Ardışıl veya doğrusal arama:** Aranana bulunana kadar komisyon çalışır.

2. **İkili arama:** Köşeye sıkıştırma yöntemi.
3. **Çırpı fonksiyonu:** Çırpı tablosu, çatışmaların çözülmesi
4. **Metin içerisinde sözcük arama**

5. Java ile Programlama

5.1. Veri Tipleri ve Değişkenler

Her sınıf bir veri tipi, her veri tipi bir sınıftır

Ram'e direk kaydedebildiğim veriler ilkeldir. Küçük harf ile başlar.

```
void topla(){
    System.out.println("toplama işlemi yapılır...");
}

int topla(){
    return 5;
}

Car car = new Car();
System.out.println(car.door)
/* Null */

System.out.println(-4 / 0)
/* -Infinity */

System.out.println(0 / 0)
/* NaN */

public class Demo {
    static final double AVOGADRO; /* Yanlış */
    static final double PI = 3.14 /* 3,14 değil */
    static int a; /* 0 */
    int b = 1;

    public static void main(String[] args) {
        /* Java kutsal kitabının emri */
        static int c = 3; /* Yanlış */
        b = 4; /* Yanlış */
    }
}

/*Recursive*/
public static int sum(int k) {
    if (k > 0) {
        return k + sum(k - 1);
    } else {
        return 0;
    }
}
/* 10 + sum(9) ... 10 + 9 + 8 + 7 + 6 + 5 + 4 + 3 + 2 + 1 + 0 */
```

```
/* Construction, Inheritance, Polymorphism, Interface*/
interface SporPaket{
    void otomatikVites();
    void farSensoru();
    void yagmurSensoru();
    void startStop();
}

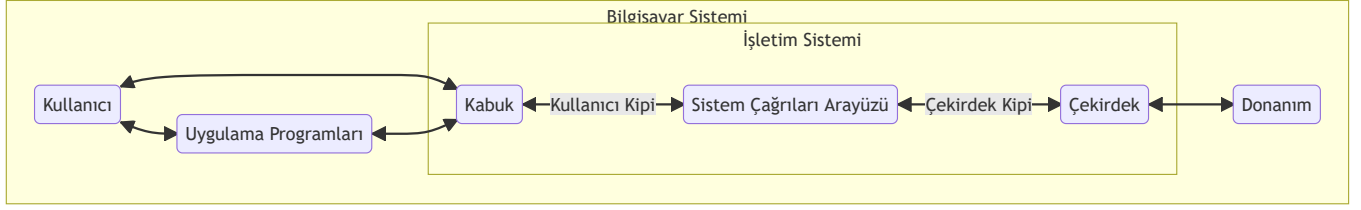
abstract class Renault implements SporPaket{
    public void otomatikVites(){
        System.out.println("P, N, D, R");
    }
    abstract void havaYastigi();
}

class Megane extends Renault{
    public void farSensoru(){
        System.out.println("Aydınlık, Karanlık");
    }
    public void yagmurSensoru(){
        System.out.println("Çise, Normal, Sağanak");
    }
    public void startStop(){
        System.out.println("Çalıştır, Durdur");
    }
    void havaYastigi(){
        System.out.println("Kaza");
    }
}

class TestDrive{
    public static void main(String args[]){
        Renault r = new Megane();
        r.otomatikVites();
        r.farSensoru();
        r.yagmurSensoru();
        r.startStop();
        r.havaYastigi();
    }
}
```

6. İşletim Sistemleri

6.1. Bilgisayarlar ve İşletim Sistemleri



6.2. Bilgisayar Mimarileri

	Harvard Mimarisi	von Neumann Mimarisi
Düzenleme	ROM komutlar, RAM veri için. Belleğe erişim yolları bağımsız.	ROM bilgisayarın açılışı, RAM veri ve kodlar için. Belleğe erişim yolları ortak.
Donanım Gereksinimi	Karmaşık	Basit
Alan Gereksinimi	Çok	Az
Komut Çalıştırma Hızı	Yüksek	Düşük
Bellek Alan Kullanımı	Çok	Az ve efektif
Kontrol Karmaşıklığı	Yüksek	Az

6.2.1. CISC (complex instruction set computer) Mimarisi

CISC mimarisi yüksek seviyeli programlama dillerindeki ifadelerin makine dilinde benzer şekilde doğrudan ifade edilmesine imkan sağlayacak şekilde tasarlanmıştır. Bunun gereği olarak CISC mimarisinde komut yapısı operantlardan en az bir tanesinin birlikte olmasını öngörür ve komutlar birden fazla alt operasyondan oluşurlar.

6.2.2. RISC (reduced instruction set computer) Mimarisi

CISC mimarisi kullanıcı odaklı iken RISC mimarisi donanım odaklı bir tasarıma sahiptir. RISC mimarisinde alt operasyonların böl ve yönet mantığına göre bellekten okuma, aritmetik lojik işlem yapma ve belleğe yazma olmak üzere birbirinden bağımsız üç ayrı komik şekilde ifade edilmesi ile ortadan kaldırmıştır. Bu yaklaşımın sonucu RISC mimarisindeki bütün komutların uzunluğu birbirine eşittir ve aritmetik lojik işlemler sadece saklıcılar içinde gerçekleştirilir.

```
/* CISC */
MULT Addr1, Addr2
```



```

/* RISC */
LOAD A, Addr1
LOAD B, Addr2
PROD A, B
STORE Addr1, A

```

CISC	RISC
donanıma yük biner	yazılıma yük biner
çok saat döngüsü ile çalışan karmaşık komutlar	tek saat döngüsünde tamamlanabilen basit komutlar
az birleştirici kodu	çok birleştirici kodu
transistörler karmaşık kod tasarım için kullanılır	transistörler saplayıcılar için kullanılır

6.2.3. İşletim Sistemleri Ağ Özellikleri

Linux: Kullanıcı uygulamaları tüm isteklerini soket arabirimi üzerinden gerçekleştirir. Linux ağ katmanı BSD soket katmanı alt yapısı üzerine inşa edilmiştir. Herhangi bir ağ verisi bu katmana bir uygulamanın soketinden veya bir ağ aygıtı sürücüsünden geldiğinde verilerin hangi ağ protokolünü içerdiklerini belirten bir tamamlayıcıyla etiketlenmiş olması beklenir.

Windows: Ağ arttırım protokollerini sürücüler üzerinden uygulamaktadır.

1. Sunucu ileti blogu (SMB) protokolü: Sistem üzerinden giriş çıkış isteklerinin gönderilmesi, oturumunda denetimi, sunucudan kaynana yönlendirme bağlantısını başlatan ve sonlandıran, yeniden yönlendirme süreçlerini yöneten, dosyalara erişim kontrollerini yapan, ağ üzerindeki yazılara veri gönderim süreçleri yöneten protokol. Yeni adı Ortak İnternet Dosya Sistemi (CIFS)
2. İletim Kontrol Protokolü: Basit ağ Yönetim Protokolü (SNMP), dinamik ana bilgisayar yapılandırma protokolü (DHCP) ve Windows İnternet Ad (WINS) gibi hizmetleri içerir.
3. Noktadan noktaya tünel protokolü (PPTP)
4. HTTP
5. Uzaktan Yordan Çağrılar (RPC)/(VPN)
6. Bileşen nesne modeli (COM)/(DCOM)

7. Veri Tabanı Yönetim Sistemleri

7.1. Veritabanı Yönetim Sistemleri

7.1.1. Kavramlar

1. Ana veri tabloları (Master)
2. İşlem tabloları (Transaction)
3. Alan (Field) *col*
 1. Anahtar alan (Key)
 2. İkincil anahtar alan(Foreign key)

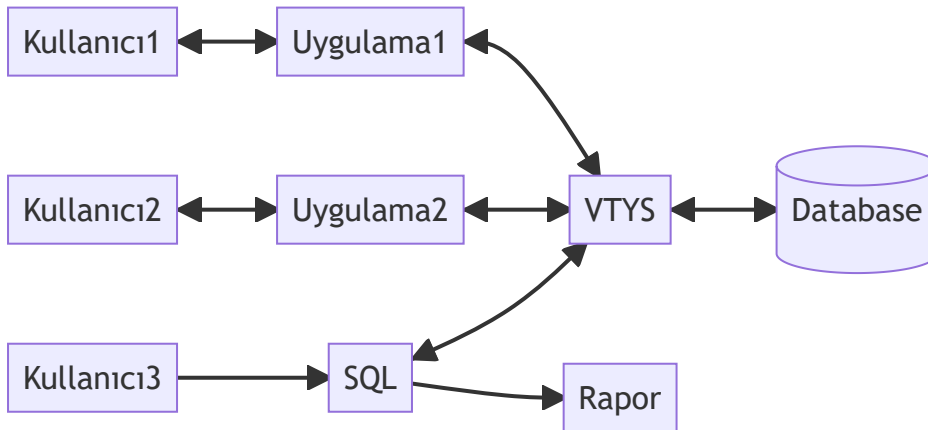
4. Kayıt (Record) *row*
5. Dataset
6. Index: Tabloların belirtilen alanlara göre sıralanması ile oluşan data setleri ifade eder.

7.1.2. Uygulamalar

Uygulama	Ortam	Boyut	Yenilik, Özellik
MS Access	Windows (Lokal)	25 MB	
MySQL	Unix/Linux (Web)	2 GB	Trigger, Stored Procedure
IBM DB2	IBM		+ Transaction Locking, Yüksek performans, yüksek maliyet
Informix	Postgres		
MS SQL	Windows	4 TB	Maliyetli, Güvenli
Postgres		64 TB	Veri güvenliği ön planda, açık kaynak
Oracle		Sınırsız	Yüksek güvenlik, yüksek maliyet
Sybase			Orta ve büyük ölçekli banka ve kamusal alan
BerkeleyDB	Gömülü	256 TB	
SQLite	Native		
NoSQL			İlişkisel olmayan veriler için (Not Only SQL) Çok yüksek performans

7.1.3. Yaklaşımlar

1. Geleneksel yaklaşım (*dosyalama*)
2. Veritabanı yaklaşımı



7.2. Veri Modelleri

1. Hiyerarşik *bire-çok*

2. Ağ *çok*-*çok*
3. Nesneye yönelik: Karmaşık verileri kolay saklar, Büyük verinin işlenmesi yavaş. *Java*
4. İlişkisel
5. Varlık-İlişki modeli

8. Yazılım Mühendisliği

9. Web Programlama

9.1. HTML5

9.1.1. Neden Önemli

1. Yeni standartlar
2. Hızlı kolay kodlama
3. Modern
4. Mobil cihaz desteği
5. Tarayıcı desteği
6. Temiz kodlama
7. Semantik ve genişletilebilir
8. Daha iyi veri girişi
9. Video ve ses desteği
10. CSS3 ile daha zengin içerikler
11. Çevrimdışı uygulama yeteneği
12. API desteği
13. Yerel veri depolama
14. Oyun programlama
15. Mobil uygulama geliştirme

9.1.2. Formlar

```
<form>
  <label for="fname">First name:</label><br />
  <input type="text" id="fname" name="fname" />
</form>
```

9.1.2.1. Özellikleri

Attributes	Açıklama
accept-charset	Specifies the character encodings used for form submission
action	Specifies where to send the form-data when a form is submitted
autocomplete	Specifies whether a form should have autocomplete on or off

Attributes	Açıklama
enctype	Specifies how the form-data should be encoded when submitting it to the server (only for method="post")
method	Specifies the HTTP method to use when sending form-data
name	Specifies the name of the form
novalidate	Specifies that the form should not be validated when submitted
rel	Specifies the relationship between a linked resource and the current document
target	Specifies where to display the response that is received after submitting the form

9.1.3. Elemanlar

```
<!--açılır liste-->
<label for="cars">Choose a car:</label>
<select id="cars" name="cars">
  <option value="volvo" selected>Volvo</option>
  <option value="saab">Saab</option>
  <option value="fiat">Fiat</option>
  <option value="audi">Audi</option>
</select>
```

```
<!--büyük metin girişi-->
<textarea name="message" rows="10" cols="30">
The cat was playing in the garden.
</textarea>
```

```
<button type="button" onclick="alert('Hello World!')">Click Me!</button>
```

```
<form action="/action_page.php">
  <fieldset>
    <legend>Personalia:</legend>
    <label for="fname">First name:</label><br />
    <input type="text" id="fname" name="fname" value="John" /><br />
    <label for="lname">Last name:</label><br />
    <input type="text" id="lname" name="lname" value="Doe" /><br /><br />
    <input type="submit" value="Submit" />
  </fieldset>
</form>
```

```
<form action="/action_page.php">
  <input list="browsers" />
  <datalist id="browsers">
    <option value="Internet Explorer"></option>
    <option value="Firefox"></option>
    <option value="Chrome"></option>
    <option value="Opera"></option>
    <option value="Safari"></option>
  </datalist>
</form>
```

9.1.3.1. Input Types

```
<input type="button" />
<input type="checkbox" />
<!--checkbox name attribute aynı olmalı-->
<input type="color" />
<input type="date" />
<input type="datetime-local" />
<input type="email" />
<input type="file" />
<input type="hidden" />
<input type="image" />
<input type="month" />
<input type="number" />
<input type="password" />
<input type="radio" />
<!--radio name attribute aynı olmalı-->
<input type="range" />
<input type="reset" />
<input type="search" />
<input type="submit" />
<input type="tel" />
<input type="text" />
<input type="time" />
<input type="url" />
<input type="week" />
```

9.2. CSS3

```
* {
  color: red;
  /*herşey*/
}

p {
  color: red;
}
```

```
.class {
  color: red;
}

#id {
  color: red;
}

p.class {
  color: red;
  /*sınıfı 'class' olan paragraflar*/
}

.class1,
.class2 {
  color: red;
  /*hem birinci hem ikinci sınıf ortak*/
}

.class1 > .class2 {
  color: red;
  /*birinci sınıfın hemen altındaki ikinci sınıf, çocuğu*/
}

.class1 .class2 {
  color: red;
  /*birinci sınıfın oğlu yada torunu olan ikinci sınıf*/
}

@font-face {
  font-family: Sansation;
  src: url(sansation_light.woff);
}

@keyframes myexample {
  from {
    top: 0px;
  }
  50% {
    top: 100px;
  }
  to {
    top: 200px;
  }
}
```

9.2.1. CSS Pseudo Classes & Elements

www.w3schools.com/css/css_pseudo_classes.asp

9.2.2. EcmaScript 6

1. Scope
2. Arrow fonksiyon
3. For of döngüsü
 1. String
 2. Array
 3. Map
 4. Set
4. Parametreler
 1. Default
 2. Rest (Dizi)
 3. Template
5. Map
6. Set
7. Built-in metotlar
 1. Objeye obje ekleme `let depo = Object.assign({}, obje1, obje2);`
 2. Array eleman arama
 3. String tekrarlama
 4. String arama
 5. Number güvenlik kontrolü
 6. Number işaret saptaması
8. Modüller
9. Class
10. Promise

10. Yazılım Proje Yönetimi
