

Bilgisayar Mühendisliğine Giriş

1. Temel Kavramlar

- Bilgisayar mühendisinin yapacağı tek şey program yazmak değildir. Bunun yerine programcının program yazmak için kullanacağı araçları tasarlamak ya da bir projede koordinasyonu sağlamak ve yönetmektir.
- Bununla birlikte tüm bunları yapabilmesi için programcılık mezunu birinden çok daha iyi program yazabilmelidir.
- Program tasarımı bir sanattır, diğer sanat dalları gibi belirli bir temel eğitim alındıktan sonra kişinin kendi bilgi, deneyim, beğeni ve duygularını yansıttığı bir ürünü ortaya çıkardığı bir çalışma şeklidir.
- Bu sunumda bilgisayar mühendislerinin görevleri açıklanacak, bilgisayar mühendisliğinin temel konuları ve tanımları verilecektir.

Bilgisayar Mühendisliği Eğitimi ve Görevleri

- Bilgisayar mühendisliği bilgisayar programları yazmayı öğrenmekten öte, bir problemi çözecek sistemlerin tasarlanmasını, sistem tasarlayacak yeni araçların ve teknolojilerin geliştirilmesini sağlayacak bilgi ve beceriyi kazandırmayı amaçlar.
- Bilgisayar mühendisliğinin bilimsel bir tanımı yapılacak olursa: Veriyi bilgiye dönüştüren algoritmik süreçlerin sistematik olarak incelenmesi ve geliştirilmesini amaçlayan bir disiplindir. Bu tip bir eğitim, kişinin sorunlara farklı gözle bakabilmesini ve farklı çözümler üretebilmesini sağlar.
- Günümüzde, yaygın kanının aksine, sadece programcılık eğitimi olarak bu konuda bir uzmanın niteliklerine sahip olmak mümkün değildir. Çünkü bilgisayar mühendisliği eğitiminin kazandıracığı en önemli özellik algoritmik düşünme yeteneğidir.
- Bilgisayar mühendisliği eğitimi alan birisi donanım yazılım geliştirme, hazır donanımlar üzerinde çalışacak sistem ve kullanıcı yazılımları geliştirme, veritabanı uygulamaları yazma, bilgisayar ağlarını projelendirme ve günümüzde öne çıkan bilgisayar ve bilgi güvenliği konularında önleyici sistemler tasarlamaktır.
- Kamu ve özel sektörde bilgisayar mühendislerine olan gereksinim artarak sürmektedir. Bu yüzden iş bulma konusunda bilgisayar mühendisleri sıkıntı çekmemektedir.
- Diğer taraftan teknoloji hızla geliştiği için mühendislik alanında çalışan insanların yeni teknolojilere adapte olabilmeleri rekabetçi bir piyasada yerlerini korumalarını sağlayacaktır. Örneğin bundan 5-10 yıl öncesine ait bilgisayar dergileri incelendiğinde piyasadaki donanım ürünleri ve fiyatları tebessüm ettirecek türdendir.
- Günümüzde bilgisayar mühendisliğini meslek olarak seçmemiş insanların çoğu bilgisayar teknolojisine aşina olması gerekir. Örneğin veritabanı çalışma mantığını anlayan bir banka müdürü, şubesinin verimliliğini nasıl arttırabileceğini daha kolay bulabilecektir. Sayısal imzanın çalışma mantığını anlayan bir avukat karşılaştığı sahtekârlık davalarını daha kolay çözebilecek ve müvekkilini daha kolay savunabilecektir.

Bilgisayar Mühendisliği Konuları

- Bilgisayar mühendisliği temelde iki konudan oluşmaktadır.
- Donanım, Yazılım ve Bilgisayar Bilimleri

Donanım

- Donanım alanına sayısal sistemlerin tasarlanması ve programlanması girmektedir. Örneğin cep telefonları için yazılan işletim sistemleri ve yazılımları ya da gömülü sistemler bu alana girmektedir. Yine bu alana sağlık ve güvenlik alanlarında kullanılan gerçek zamanlı işletim sistemleri ve sayısal sinyal işleme konuları girmektedir.

Yazılım

- Yazılım geliştirmek için algoritma tasarımı yazılım alanında bilgisayar mühendislerinin en çok yaptığı işlerdir. Tasarlanan algoritmalar daha sonra bilgisayar mühendislerinin aralarında bulunduğu yazılımcılar tarafından programa dönüştürülürler.
- Bir bilgisayar mühendisi olarak yoğunlukla kullanacağınız bir diğer teknoloji veritabanıdır. Veritabanı tasarımı, sorgulamalar, raporlamalar ve arayüzler veritabanıyla ilgili çalışma alanlarıdır.
- Bilgisayar ağları ve veri güvenliği de yine yazılım alanına giren konulardır. Bilgisayar ağlarının tasarlanması, yönlendirici, switch gibi bileşenlerin en uygun şekilde konfigüre edilmesi, ağ güvenliğinin sağlanması bu konu içindedir. Yine veri güvenliği alanında şifreleme, veri ve bil-

gisayar güvenliği, bilgisayar saldırılarına karşı programların geliştirilmesi bu yazılım konusu altında değerlendirilebilir.

- Bu konuların hepsi aslında kendi içinde bir uzmanlık alanıdır ve iş hayatına atılan bir bilgisayar mühendisinin bu alanların tümüne hakim olması beklenemez. Bunun yerine bilgisayar mühendisleri bu alanlardan bir veya bir kaçında uzmanlaşırlar.

Bilgisayar Donanımı

- Donanım elektronik devre elemanlarında oluşan ve birleştiğinde bilgisayar sistemini oluşturan bileşenlere verilen genel addır. Bu bileşenleri öğrenmeden önce bazı terimlerin anlamına bakılmalıdır. Örneğin bilgisayar dilinde en küçük veri birimi bittir ve farklılık göstermekle birlikte genel 0V ile 0, 5V ile 1 ifade edilerek saklanacak tüm bilgiler 0 ve birlerle ifade edilir. Sekiz bitin birleştirilmesiyle byte, iki baytın birleşmesiyle Word (sözcük) oluşur.

Merkezi işlem birimi:

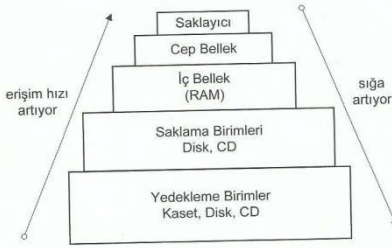
- Bilgisayarı oluşturan en önemli birim merkezi işlem birimidir kısaca MİB ya da central processing unit'in kısaltması olarak CPU adı ile anılır. Bilgisayarınızda yapılan tüm işlemler aslında MİB de gerçekleşmektedir. Veri okunması, veriler üzerinde işlem yapılması ve elde edilen sonucun saklanması hep MİB'in yaptığı işlemler sonucundadır. MİB içinde temelde aritmetik, lojik ve karşılaştırma işlemleri yapılır. Yazdığınız programlarda da aslında temelde bu işlemlerin ard arda anlamlı bir şekilde sıralanmasıyla ifade edilerek işlemci üzerinde yürütülür.
- Farklı firmalar farklı güçte ve özellikte işlemciler üretmektedir. Örneğin Intel firmasının bir modeli olan Pentium işlemciler çoğu kişisel bilgisayarda ilk tercih edilen model olmuştur, bunun yanı sıra daha küçük gömülü sistemlerde kullanılan Motorola firmasının 68040 modeli de bir diğer işlemci çeşidi olarak verilebilir. Teknolojinin ve işlemci üretim süreçlerinin gelişmesiyle günümüzde çok daha güçlü işlemciler çok daha ucuza imal edilebilmektedir.
- İşlemciler temelde transistörlerden oluşurlar. Transistörler aritmetik ve mantık işlemlerini yapacak şekilde organize olmuşlardır. Aritmetik işlemler toplama, çıkarma gibi işlemlerdir. Bazı işlemciler çarpma ve bölme işlemi için de bu iki işlemi kullanırlar. Mantık işlemleri ise VE, VEYA, DEĞİL gibi işlemlerdir. Mantık işlemlerinin matematiksel olarak ifade edildiği sisteme Boole cebri denir. Boole cebrinde verilen bir fonksiyonun doğruluk tablosu üzerinde sadeleştirme işlemi uygulanarak, bu fonksiyonu gerçekleştirecek devrenin daha sade ve daha az devre elemanı ile gerçekleştirilmesi sağlanır.
- Mantık devrelerini birleşik ve sıralı olarak ikiye ayrılırlar. Bu iki tür devrenin temel farkı hafızanın varlığıdır. Birleşik mantık devreleri girişine ne uygulanırsa sonucu ona göre üretir, bu devreler genelde ve veya gibi mantık kapılarından oluşur.
- Sıralı mantık devrelerinin çıkışı hem girişlerine hem de devrelerin o an hafızada tuttukları durumlarına bağlıdır. Bu devreler saat darbeleriyle çalışır. Flip-floplar bu devrelere örnek olarak gösterilebilir.
- MİB'in, anabellek, ikincil bellek ve çevre birimleriyle haberleşmesini, kontrol etmesini tanımlayan modele «bilgisayar mimarisi» denir. Bilgisayarlar farklı mimariyle sahip olma da genel olarak temel bileşenlerin hepsine sahiptir.
- MİB'e ne yapacağını söylemenin en alt düzeyde ve en zor yolu makine kodudur. Makine kodu 1 ve 0'lardan oluşur ve MİB bu kodu ne anlamak için bir yorumlayıcıya ve derleyiciye ihtiyaç duymaz. Fakat her işlemci modelinin veya mimarisinin makine kodu farklı olacağı için bir MİB'e yazdığını makine kodu başka bir MİB'de çalışmayabilir.
- Makine koduyla program yazmak zor olduğu için, işlemci komutu düzeyinde simgeleştirme yapan assembly dili geliştirilmiştir. Örneğin toplama işleminin makine kodu 0xFB iken, assembly dilinde «add» kelimesidir.

- MİB'in içinde bundan başka saklayıcılar kontrol birimleri ve Arimetik Lojik Birim (ALU) bulunmaktadır.
- Saklayıcılar verilerin geçici olarak saklanması için kullanılan hızlı bellek parçalarıdır. Bu birimlerin kapasiteleri çok küçüktür.
- Kontrol birimleri, işlemci ile çevre birimlerini ve bellekleri birbirine bağlayan veriyolu arasında koordinasyonu ve veri alışverişini kontrol eden birimlerdir.

Bellek

- Bellek çalıştırılacak programın, üzerinde işlem yapılacak verilerin ve üretilen sonuçların içinde saklandığı donanımdır. Bilgisayarda en çok kullanılan bellek türü rastgele erişimli bellek (RAM) dir. Bu belleklerin özelliği elektrik kesilince içindeki bilgileri kaybetmesidir.
- Sadece yazılabilir bellekler ise yazma işlemin sadece bir kez yapılabilirdiği fakat elektrik kesintilerinden etkilenmeyen belleklerdir. Bu yüzden bilgisayarın ilk açılışında çalıştırılacak ve tüm donanımı kontrol edecek program olan BIOS bu tür bir bellekte saklanmaktadır. Kızıl ötesi elektrik kullanarak silinebilen (EPROM) ve elektrik akımı ile tekrar programlanabilen (EEPROM) çeşitleri de vardır. EEPROM lar sıkça kullandığımız USB belleklerin temel bileşenleridir.
- RAM belleğin pahalı olması, MİB'in adresleyebileceği bellek alanının sınırlı olması nedeniyle programların verilerinin saklanması için ikincil bellek kullanılır.
- İkincil belleğe örnek olarak CD, DVD ve sabit diskler gösterilebilir. İkincil belleğin en önemli özelliği elektrik kesintisinden etkilenmemeleri ve kapasitelerinin büyük olmasıdır. Bunun bir dezavantajı olarak da erişim hızları düşüktür.
- Diskler manyetik belleklerdir. Veriler manyetik plakalar üzerindeki alanların farklı polarize olması ile saklanırlar. Okuyucu kafa manyetik diskte ilgili alana gelir bilgiyi okur ve ana belleğe aktarır. Bu okumanın olabilmesi için önce okuyucu kafa ilgili ize gelir sonra disk dönerek okunacak bölgeyi kafanın altına getirir.

- CD ve DVD gibi optik bellekler ise verilerin bir madde üzerine optik izler bırakarak saklanması esasına dayanır. Yine bu tür belleklerin de bir kez yazılabilenleri olduğu gibi birden çok yazılabilenleri de vardır.
- Belleklerin sığası yani kapasiteleri arttıkça erişim süreleri artmakta bu yüzden hızları düşmektedir. Buna göre bir hiyerarşi yapılacak olursa en hızlı ve kapasitesi en az bellek türü saklayıcılar, en yavaş ve kapasitesi en çok olan bellek türü de yedekleme birimleridir.



Giriş Çıkış Birimleri

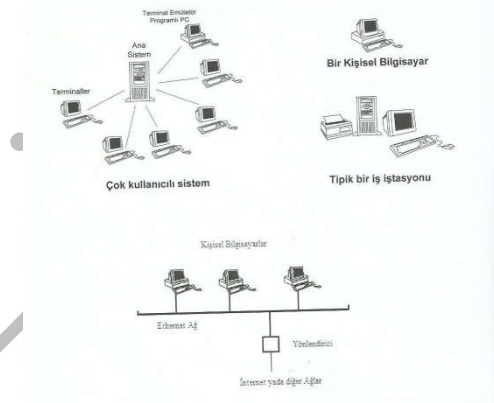
- Bilgisayarınızda işlem yapabilmeniz için ona komut verebilmeli ve işleyeceği verileri girebilmelisiniz. Bu yüzden klavye, fare, mikrofon gibi bilgisayara veri girişi yapabilen aygıtlar tasarlanmıştır. Bu aygıtlar, sahip oldukları kontrol kartları ve sürücülerle kullanıcıdan aldıkları bilgileri, işletim sistemi aracılığı ile, gerekli programa iletirler.
- İşlenen verilerden elde edilen sonucun gösterilebilmesi için monitör, yazıcı gibi çıkış cihazlarına da ihtiyaç vardır. Yine bu cihazlarda da kontrol birimleri ve sürücü yazılımlarıyla bilgisayarın gerekli parçalarıyla haberleşirler.
- Manyetik ve optik diskler de aslında birer giriş çıkış aygıtlarıdır.
- Giriş çıkış birimlerinin birbiriyle haberleşebilmesi için temelde iki bağlantı türü vardır. Bunlar seri iletişim ve paralel iletişimdir. Seri iletişimde bilgi bit-bit iletilirken, paralel iletişimde aynı anda birden fazla bitlik bilgi iletilir.
- Günümüzde yaygın olarak kullanılan bağlantı türü olan USB (Evrensel Seri Yol) aslında bir seri iletişim çeşididir. En büyük özelliği bu şekilde

iletişime geçecek bir cihazın ayrı bir güç kaynağına ihtiyaç duymamasıdır.

- Giriş çıkış cihazlarının denetlenmesi işletim sistemlerinin görevidir.

Bilgisayar Ağları

- Bilgisayar ağları, bilgisayar ve diğer cihazların ağ üzerinden haberleşmelerini sağlamak için kullanılır. Bu sayede ağ üzerinden veri ve kaynak paylaşımı yapılabilir. Ağda kullanılan veri aktarma kurallarına protokol denir. En yaygın olarak kullanılan protokollerden biri, İnternetin de kullandığı, TCP/IP (Transmission Control Protocol / Internet Protocol)dir.
- Bilgisayarların birbiriyle haberleşebilmesi için istemci sunucu mimarisi geliştirilmiştir. Bu sayede çok fazla işlem gücü ve güvenlik gerektiren veritabanı programları sunucuda tutulmakta, yapılan sorguların sonucu da istemcilere dönmektedir.
- Bilgisayar ağlarının kullanım alanlarından biri de bir uygulamanın farklı sistemlerde çalıştırılabilmesidir.
- Bilgisayar ağına yönelik yazacağınız programda ağın mimarisi ve ağda kullanılan protokol çok önemlidir. Örneğin TCP/IP protokolü için yazdığınız bir programı farklı bir protokolle çalışan bir ağa götürdüğünüzde çalışmayacaktır.
- NETBUI, NETWARE, AppleTalk diğer iletişim protokolleridir.



Bilgisayar Yazılımı

- Bilgisayar yazılımlarını işletim sistemleri gibi sistem yazılımları ve bir kelime işlemci gibi son kullanıcıya hitap uygulama yazılımları olmak üzere iki sınıfta ele alabiliriz. Her iki yazılım türü de, belirlenene işleri yapmak üzere bir programlama dilinde yazılmış amaca uygun program ve verilerden oluşan komutlar dizisidir.

Yazılım

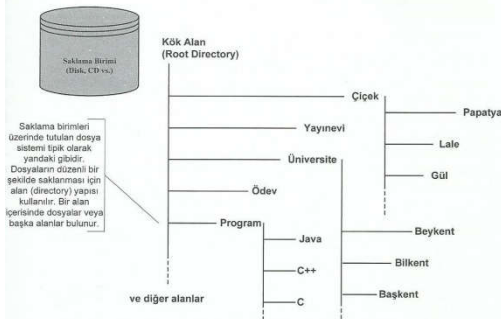
- Program belirli bir işi yerine getiren, kendi içinde bir bütün olan, algoritmik ifadelerdir. Program günümüzde kod yazılarak yapılmaktadır ama geçmiş yıllarda, özellikle bilgisayar tarihinin ilk yıllarında programlar donanımsal olarak yazılıyordu.
- Bilgisayar programcıları için sadece yazılım tarafıyla ilgilenirken, sistem geliştiriciler hem yazılım hem de bu yazılımın üzerinde çalışacağı donanım konularının her ikisiyle birden ilgilenmek zorundadırlar.
- Yazılımın tanımına gelecek olursak, yazılım bir işin program kodu üretilerek yapılmasıdır. Bu durumda donanım dışında kalan, programcının istenen bir işi yerine getirmek üzere yazdığı kod ve veri kümesidir.
- Yazılım bir veya birden çok program kümesinin bir araya gelmesinden oluşur. Özellikle büyük ölçekli yazılımlar birçok program, veri ve dokümanın bir araya gelmesiyle oluşur.
- Biri işin yapılması için tasarlanmış algoritmik ifadenin bir programla dili seçilerek (ör C, Java, C#), seçilen dilin özellikleri uygun bir şekilde kodlanması sonucu ortaya çıkan metne program kodu denir. Bir satırlık bir program olabileceği gibi milyonlarca satırdan oluşan programlar da vardır.
- Bir programda saklanması gereken değerler olabilir. Bu bilgilerin tutulduğu bellek gözlerine verilen simgesel isimlere de-

ğişken denir. Bir değışkenden sakladığınız bilgi aslında bilgisayarınız bel-
leğinde saklanmaktadır. Yani aslında değışkenler, bilgisayar belleğinde
belli bir alanı işaret eden etiketlerdir. Değışkenlere verilecek isimler,
programlama diline has kurallara uyduktan sonra, tamamen programcıya
bırakılmıştır.

- Değışkenlerde saklanmış olan bilgiler üzerinde işlem yapmaya yarayan
simgelere operatör denir (ör: toplama, çıkarma, and, or). Programlama
dillerinde operatörler önceden tanımlanmışlardır.
- Bilgisayar ortamında metinlerde aslında sayılar şeklinde tutulur. Hangi
sayının hangi harfe karşılık geldiği ASCII veya UNICODE tablosu de-
diğimiz tablolardan bakılabilir. Örneğin BABA sözcüğünün bellekte
kapladığı değer ASCII tablosuna göre 66 65 66 65 şeklinde olacaktır.
- Sözcükler aritmetik veya lojik işlemlere sokulamazlar. Fakat günümüz
gelişmiş programlama dillerinde toplama ve karşılaştırma işlemlerine di-
rek tabi tutulabilirler.
- Atama işlemi bir değerin bir değışkene ya da değışkenin gösterdiği ha-
fıza alanına yazılması işlemidir. Programlama dillerine göre farklı şekil-
lerde ifade edilebilirler.
- Yazılım hakkında buraya kadar ifade edilen tüm bilgiler bir program-
lama dilinden diğerine farklılık gösterebilir.

İşletim Sistemi

- İşletim sistemi, bilgisayarın donanımı ile, kullanıcı programı arasında
etkileşimi sağlayan, bu tür programlara hazır imkânlar sunan, sistemin
donanımsal ve yazılımsal kaynaklarının paylaşılmasından ve yönetil-
mesinden sorumlu yazılımdır. Bir bilgisayar sistemi, işletim sistemi ol-
madan transistör yığınından başka bir şey değildir. En çok bilinen işle-
tim sistemleri Windows, Linux, Mac OS ve günümüzde telefonların do-
nanımlarını yöneten Android ve IOS işletim sistemleridir.
- Çok kullanıcı ve çok görevli işletim sisteminin en büyük yeteneği bir
bilgisayarda aynı anda birden fazla görevi paralel olarak çalıştırabilme-
sidir bunu işlemcisi değişik görevlere sırayla atayarak gerçekleştirir.
Bu sayede sistem kaynaklarını verimli bir şekilde kullanmış olur.
- İşletim sisteminin bir diğer görevi bellek yönetimidir. Bazı programla-
rın bilgisayarın sahip olduğu boş fiziksel bellekten fazlasına ihtiyaç du-
yarlar. Bu durumda kullanılmayan bir bellek sayfası diske yazılır ve bo-
şalan sayfa bu programa atanır. Tüm sayfalar dolduğunda ne yapılacağı-
na veya hangi sayfaların diske yazılacağına işletim sistemi karar verir.
- Sonradan kullanmak üzere dosyaları diske saklamak da yine işletim sis-
teminin görevidir. Disklerin saklayabileceği birim kapasiteler bellidir.
Birim kapasitenin üzerinde boyutu olan dosyaların nasıl saklanacağına
yine işletim sistemi ve kullandığı dosya sistemi karar verir.
- Dosyaların saklanması klasörler ve hiyerarşik yapılar kullanılır. Bu
sayede birbiri ile ilişkili veriler bilgisayarda gruplanarak bir organizas-
yon sağlanır. Örneğin Windows işletim sistemini ilgilendiren dosyaların
hepsi C:\Windows klasöründe, kullanıcı programlarından 32 bit olanlar
C:\Program Files (x86), 64 bit olanlar da C:\Program Files klasöründedir.



- Giriş çıkış birimlerini kontrol etmek de yine işletim sistemlerinin göre-
vidir. Bu sistemleri kontrol edecek sürücü yazılımlarının çoğu artık işle-
tim sistemi ile birlikte gelmektedir.

Sayı Sistemleri:

- Sayı sistemleri, bir değeri değışik biçimlerde ifade etmeye yarayan
aritmetiksel ifadelerdir. Kullanılan tabana göre kullanılacak rakamlar ve
basamak değeri de farklılık gösterecektir. Farklı sayı tabanlarının
farklı kullanım alanları vardır. Örneğin günlük hayatımızda onlu sayı

sistemine alışmış olmamıza rağmen bilgisayar sistemlerinde
ikili ve onaltılı sayı sistemleri büyük avantaj sağlar.

- Değışik sayı sistemleri için matematiktekinden farklı olarak
değışik kodlama teknikleri geliştirilmiştir.

Algoritma

- Bir problemin elle veya bilgisayarla çözülebilmesi için önce-
likle analize edilmesi gerekir. Algoritma bu analiz sonucu üreti-
len ve problemin çözümüne yönelik net, bir programla dili ile
ifade edilebilir, sıralı adımlar seti olarak tanımlanabilir. Bir
problemin çözümüne yönelik bu özellikleri taşıyan bir algorit-
ma oluşturulabiliyorsa bu problem bilgisayar yardımıyla çözü-
lebilir demektir.
- Algoritmanın ifade edilebilmesi için temel adımlar, akış şema-
ları veya yalancı kod (pseudocode) diye ifade edilen biçimleyici
kurallar kullanılır. Bu biçimleyici kuralların standartları vardır
ve bu sayede bu dillerle ifade edilen algoritmalar kolayca pro-
gramcı tarafından anlaşılabilir kodu yazılabilir. Örneğin «a>b ise
c=10» şeklinde ifade edilen yalancı kod herhangi bir program-
lama dilince kolayca kodlanabilir.
- Çevrimli yada iteratif bir yaklaşım çoğu problemi çözerken
uygulanması gereken bir yoldur. Bu ihtiyaç programlama dille-
rinin hemen hemen hepsinde kendine karşılık bulmuştur. Örne-
ğin C dilinde for, while gibi deyimlerle ifade edilen yapılar bu
şekilde tekrarlı çalışma gerektiren kodları ifade etmek için kul-
lanılır.
- Özyinelemeli (rekürsif) yaklaşım ise, bu şekilde tekrarlamalı
çalışma gerektiren kodları döngü ile değil de kodun kendini ça-
ğırması ile çözmeye çalışmaktadır. Rekürsif yaklaşım daha hızlı
olduğu için değil, bazı problemlerin doğası gereği kullanılmak-
tadır.
- Tasarlanan algoritmanın verimli olması bir diğer tasarım prob-
lemidir. Algoritmanın verimliliği zaman karmaşıklığı ve alan
karmaşıklığı olmak üzere iki alanda incelenir. Zaman karmaşı-
klığı algoritmanın, parametrelerine bağlı olarak ne sürede bitece-
ğinin analiz edilmesi, alan karmaşıklığı ise algoritmanın çalış-
ması için bellekte ne kadar alana ihtiyaç duyacağı ile ilgilidir.

Veri Yapıları

- Veri yapıları bilginin bellekte tutulma şeklini ve düzenini gös-
terir. Programlama dillerinde genel olarak tam sayılı, kesirli sa-
yı, karakter ve sözcük saklanması için temel veri yapıları zaten
vardır. Programcının kendi ihtiyacına göre kendi veri yapıları
tanımlayabilir. Diziler listeler ikili ağaçlar en yaygın kullanılan
veri yapılarıdır. Veri yapıları tanımlamak kullanıcı bilgisayarın
donanımını ayrıntılarıyla uğraşmaktan kurtararak, verileri kav-
ramsal olarak göstermesini sağlamaktadır.

Programlama dilleri

- Programlama dilleri bilgisayarın donanım ayrıntılarını bilmeye
gerek kalmadan program yazmayı sağlayan araçlardır doğal dil-
lerde (Türkçe, Fransızca gibi) olduğu gibi programlama dille-
rinde de kendi yazın kuralları ve gramerleri vardır. Her biri
farklı bir alanı yönelmiş diller olduğu gibi genel amaçlı Basic,
Fortran, Pascal, C# ve Java gibi programlama dilleri vardır.
Programlama dillerinin ortak özellikleri olduğu gibi birbirlerin-
den farklılıklarda göstermektedirler ama tüm programlama dil-
lerinde bilgi giriş ve çıkış için komutlar hesaplama komutları ve
kontrol komutları bulunmaktadır. Kontrol komutları çalıştırılan
bir komutun sonucuna göre programın hangi satırdan devam
edeceğini karar vermek amacıyla kullanılır. Programlama dille-
rini sınıflandıracak olursak Assembly gibi makine dillerinin ya-
nı sıra nesne tabanlı, yapısal, mantıksal ve fonksiyonel diler
olarak sınıflanabilirler. Örneğin C Dili yapısal bir dil iken Java
nesne tabanlı bir dildir.

- Nesne tabanlı programlamada sınıf nesne belli özellikleri tem-
sil eden kavramlardır. Örneğin insan canlılar içinde bir sınıfı
temsil ederken Hasan insan sınıfından bir nesnedir.

•Bununla birlikte bir nesne farklı sınıflara dahil olabilir. Örneğin Hasan öğretim üyesi, erkek ve insan sınıfındandır ve bu sınıfların tüm özelliklerini gösterir.

•Yapısal dillerde ise sınıflar yerine alt programlar kullanılır. Örneğin prosedür, fonksiyon gibi. Bunlar belli bir işi yapmaya odaklanmış bağımsız kod parçalarıdır. Tam olarak işi kendisine parametre olarak gelen veriyi gerekli işlemleri yaptıktan sonra ürettiği sonucu kendisini çağıran programa göndermektir. Bir fonksiyon dışarıdan başka bir fonksiyon tarafından çağrılabilir gibi kendi kendini de çağırabilir. Bu şekilde kendini çağıran fonksiyonları rekürsif ya da özyinelemeli fonksiyon denir.

Veritabanları

•Veritabanları bilgilerin belirli bir disiplin altına saklanması ve gerektiğinde hızlı bir şekilde bulunması için geliştirilmiş bir saklama ve sorgulama sistemidir. Veritabanında bilgiler tablolar halinde tutulur. Veriye ulaşmak için her bir tablonun anahtar dediğimiz ayırt edici bir sütunu vardır. Sorgulamalar, bu alanlar kullanılarak ve birden çok tablodan oluşan sorgular bu alanlarla ilişkilendirilerek yapılır. Veritabanlarını etkin bir şekilde kullanabilmek için veritabanı yönetim sistemleri (VTYS) geliştirmiştir. Veritabanı ulaşmak isteyen uygulama programları öncelikle bu yönetim sisteminin sağladığı arayüzü kullanmak zorundadır. Bu yaklaşımın en büyük avantajı veri bağımsızdır. Bu sayede veritabanının yapısı uygulama yazılımında değişiklik yapmadan değiştirilebilir.

•Veritabanı tasarımında da programlama dillerinde olduğu gibi değişik yaklaşımlar bulunur. Örneğin ilişkisel veri tabanları en yaygın kullanılanlardır. Bu modelde tablolar arasında ilişkiler kurularak verilere erişilir ve veri tutarlılığı sağlanır.

SQL () bir veri sorgulama dildir. Bu dil sayesinde VTYS ile iletişime geçilerek veritabanı veri yapısının tanımlanması, veritabanlarında veri yazma ve sorgulama işlemlerin yapılmasını gibi işlemlerin yapılmasını sağlar. SQL standart bir sorgulama dilidir ve Oracle, Access, SQL server gibi birçok farklı veritabanı ile etkileşimde bulunmayı sağlar.

Veri güvenliği

•Veri güvenliği konusu bilgisayardan çok daha önce ortaya çıkmıştır örneğin kâğıt ve diğer ortamlar üzerindeki bilgilerin saklanması her zaman önem arz etmiştir. Bilgisayarın ortaya çıkmasıyla bu konu daha fazla önem arz etmiştir. Bilgisayar alanında veri güvenliği iletişim anında, yedekleme üniteleri veya herhangi bir ortamda tutulan bilginin korunması anlamına gelmektedir.

Bilgi güvenliğinin temel amaçları gizlilik, bütünlük, erişebilirlik, inkâr edilememezlik ve izleme olarak verilebilir.

•Verilerin yetkisiz kişilerin eline geçmesi, geçse bile anlaşılabilmesi veri güvenliğinin ana prensiplerindendir. Bunun için veriler bilgisayarda bile makul sürelerde kırılamayacak şekilde şifrelenmektedir. Bilgisayara karşı olan saldırılar virüsler solucanlar ve genellikle servisi engelleyen saldırılar olarak gösterilebilir. Bu gibi saldırıların ortaya çıkmasında şirket çalışanlarının bilinçsizliği öne çıkmaktadır.

1.5.Özet

•Bilgisayar mühendisliği eğitiminin temel amacı algoritmik düşünmeyin üretebilmektir bununla birlikte bilgisayarın temellerini oluşturan donanım ve yazılım konularında bilgi sahibi olmalı temel ilkeleri de öğrenmelidir.

•Bilgisayar donanımı; merkezi işlem birimi, bellek, giriş çıkış üniteleri ve bilgisayar ağları gibi konuları kapsamaktadır. Bu donanımların tasarlanması, üzerlerinde çalışacak yazılımların geliştirilmesi bilgisayar mühendisliğinin görevleri arasındadır.

•Yazılım bir amaca yönelik olarak hazırlanmış programlar kümesi olarak tanımlanabilir. Program ise daha dar bir kapsamda bir görevi yerine getirmek için tasarlanmış koda denir.

•Bu şekilde bir mühendislik eğitimi alan kişi özel ya da genel amaçlı sistemler tasarlayın problemlere çözüm üretecektir.

2. BÖLÜM

Algoritmalar ve Akış Diyagramları

İnsan günlük hayatında herhangi bir işi gerçekleştirmeden önce plan yapar. Plan işin nasıl yapılacağını, hangi aşamalardan geçileceğini, ne tür

adımlar izleneceğini düzenli bir şekilde belirtir ve hedefe ulaşmada yol göstererek büyük kolaylık sağlar. Planlı bir şekilde gerçekleştirilen işlerde başarıya ulaşmak çok daha kolaydır.

İnsanın yaşamı boyunca her işinde başvurduğu plan kavramının bilgisayar dünyasındaki karşılığı algoritmadır. Bilgisayarda işlemleri gerçekleştirebilmek için izlenecek adımlar dizisine algoritma denir.

Bir diğer deyişle algoritma, işlemleri yaptırabilmek için, bilgisayara öğretilen işlem basamaklarıdır. Algoritmanın özel geometrik şekillerle çizilmiş haline de akış diyagramı adı verilir.

Algoritma kelimesi Orta Asya'daki Harezmi kentinde doğan matematikçi Ebu Abdullah Muhammed bin Musa el Harezmi'nin isminden gelmektedir. 9. yy'da cebir alanında yayınladığı «Hisab el-cebir ve el-mukabala» kitapları cebir alanındaki ilk kitaplardandır. Bu kitabının Latinceye çevrilmesiyle büyük ilgi görür ve Avrupalılar el- Harezmi yerine (Arapçada Al-Khwârizmi) algorizm kelimesini kullanırlar ve zamanla bu kelime algoritmaya dönüşür.

Program yazma adımları

- 1.Problemin analiz edilmesi
2. Uygun çözüm yollarından duruma en uygununun belirlenmesi
- 3.Algoritma veya akış diyagramlarının hazırlanması
- 4.Algoritma veya akış diyagramlarının kodlanması
- 5.Hataların düzeltilip test edilmesi

Algoritmada kullanılan operatörler

İşlemleri belirten simgelere bilgisayar terminolojisinde operatör denir.

OPERATÖRLER

Matematiksel İşlem Operatörleri

- ^ üs alma
- * çarpma
- / bölme
- + toplama
- çıkarma
- . tam ve ondalık kısım ayırma

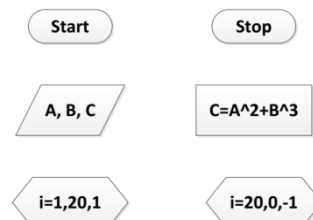
Karşılaştırma Operatörleri

- = veya == eşittir
- <> veya != eşit değildir
- < küçük
- > büyük
- <= küçük eşit
- >= büyük eşit

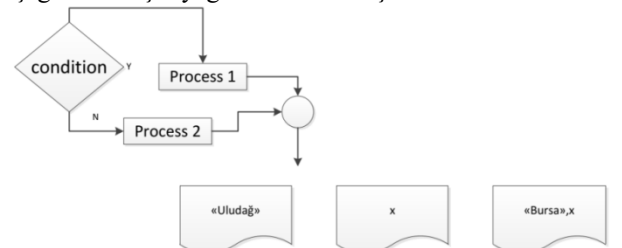
Örnekler: Aşağıdaki denklemleri bilgisayar dilinde kodlayınız

$$y1 = 4 \frac{a}{\sqrt[3]{b^2}} \quad y2 = \frac{a^2 + b^3}{\frac{ab}{a^{(2+b)}} + 3a^2} \quad y3 = \sqrt{\frac{1+a}{1+\frac{1}{a^2}}} + \frac{1}{a\sqrt{a}}$$

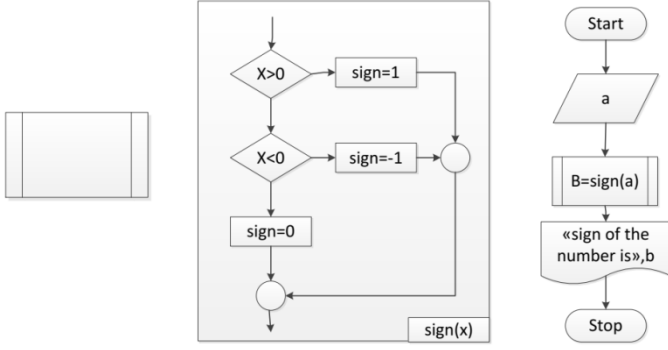
Akış Diyagramları



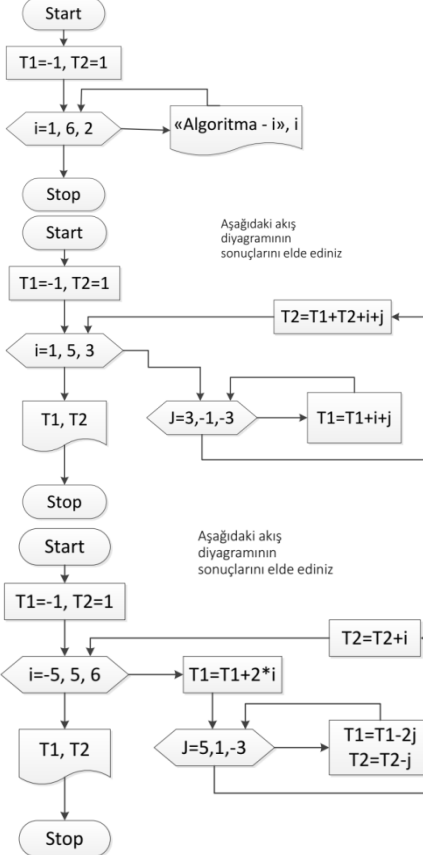
Aşağıdaki akış diyagramının ekran çıktısını elde ediniz.



Aşağıdaki akış diyagramının sonuçlarını elde ediniz.



Aşağıdaki akış diyagramının sonuçlarını elde ediniz.



Örnekler

Klavyeden girilen N sayısına göre 1 den N e kadar tek sayıların toplamını çift sayıların ise çarpımını bulan akış diyagramını çiziniz.
Girilen üç sayıdan en büyüğünü bulan programın akış diyagramını çiziniz.
Bir sayının bölenlerinin bölenlerini bulan programın akış diyagramını çiziniz.

3. BÖLÜM Boole Cebri

- Sayısal devre tasarımı için kullanılan Boole cebri cebrideki gibi sayısal nicelikleri değil, doğruluk değerlerini yani mantıksal önermeleri kullanır. Bu sayede Boole cebri sadece doğruluk değeri 1'i doğru 0'ı yanlış kabul ederek önermelerle işlem yapılmasını sağlar.
- Ayrıca bazı programlama dillerinde bool tipinde değişkenler de vardır.
- Örneğin iki önerme x ve y olsun. Doğru önermeleri «doğru» yanlış önermeleri «yanlış» sözcüğü ile gösterelim. Buna göre bu iki önermenin VE(\wedge) ve VEYA(\vee) bağlaçlarıyla olan doğruluk tablosu:

x	y	$x \wedge y$	$x \vee y$
DOĞRU	DOĞRU	DOĞRU	DOĞRU
DOĞRU	YANLIŞ	YANLIŞ	DOĞRU
YANLIŞ	DOĞRU	YANLIŞ	DOĞRU
YANLIŞ	YANLIŞ	YANLIŞ	YANLIŞ

- Bu önermeleri bilgisayarda olduğu gibi doğruyu 1 ile yanlış da 0 ile gösterecek olursak.

x	y	$x \wedge y$	$x \vee y$
1	1	1	1
1	0	0	1
0	1	0	1
0	0	0	0

- Cümleleri simgelerle kısaltarak belirli doğruları matematiksel formüllerle ortaya çıkaran Boole cebri, hesaplamaların yanı sıra felsefe alanında da oldukça fazla kullanılmaktadır.

Boole Cebri Esasları

- Boole cebrinde VE sembolü çarpımla, VEYA sembolü toplama ifade edilir. Bir değişkenin tümleyeni ise NOT (değil) işlemleri ile gösterilir.

Boole cebri teoremleri ve Örnekler:

- Değişme Kuralı: $X + Y = Y + X$ $XY = YX$
- Birleşme Kuralı: $X + Y + Z = (X + Y) + Z = X + (Y + Z)$
 $XYZ = X(YZ) = (XY)Z$
- Dağılma Kuralı: $X(Y + Z) = XY + XZ$
- Özdeşlik Kuralı: $X + X = X$ $XX = X$
- VE Kuralı: $X * 1 = X$ $X * 0 = 0$
- VEYA Kuralı: $X + 0 = X$ $X + 1 = 1$
- Tamamlayıcı Kuralı: $X + \bar{X} = 1$ $X \bar{X} = 0$
- Tersin Ters Kuralı:
 $\bar{\bar{X}} = X$ $\overline{X + Y} = \bar{X} * \bar{Y}$ $\overline{XY} = \bar{X} + \bar{Y}$
- De Morgan Kuralı: $\overline{XY} = \bar{X} + \bar{Y}$ $\overline{X + Y} = \bar{X} * \bar{Y}$
- Yutma Kuralı: $X + X * Y = X$ $X * (X + Y) = X$
- $F = A(A + \bar{B})$ ifadesinin eşdeğerini teoremler kullanarak bulunuz.
 $F = A(A + \bar{B})$ $F = A(1 + \bar{B})$
 $F = A * A + A * \bar{B}$ $F = A * 1$
 $F = A + A * \bar{B}$ $F = A$

- $F = AB + \bar{A}\bar{B}$ ifadesinin eşdeğerini teoremler kullanarak bulunuz.

$$F = AB + \bar{A}\bar{B} \quad F = A * 1$$

$$F = A(B + \bar{B}) \quad F = A$$

Doğruluk Tabloları ve Boole Cebri İşlemi Basitleştirme

- Boole cebri girişlerin alabileceği değişik durumlara göre çıkışların alacağı değerlerin gösterildiği tablolara doğruluk tabloları denir.

Aşağıda VE, VEYA, DEĞİL işlemleri için doğruluk tabloları gösterilmiştir.

X	Y	$X.Y$ (VE)	$X+Y$ (VEYA)	\bar{X} (DEĞİL)
1	1	1	1	0
1	0	0	1	0
0	1	0	1	1
0	0	0	0	1

- $F = XY + X$ fonksiyonunun doğruluk tablosunu bulunuz.

X	Y	$X * Y$	$X * Y + X$
0	0	0	0
0	1	0	0
1	0	0	1
1	1	1	1

Bu tabloya göre X ile $X * Y + X$ ifadesinin doğruluk tablosu aynı olduğu için ifade $F = X$ şeklinde sadeleştirilebilir.

$$F = XY + \bar{X}Y + \bar{Y} \quad \text{ifadesini sadeleştiriniz.}$$

$$F = XY + \bar{X}Y + \bar{Y} \quad F = Y + \bar{Y}$$

$$F = Y(X + \bar{X}) + \bar{Y} \quad F = 1$$

$$F = Y * 1 + \bar{Y}$$

- $F = XYZ + XY\bar{Z} + \bar{X}Y$ ifadesini sadeleştiriniz

$$F = XYZ + XY\bar{Z} + \bar{X}Y \quad F = Y(X + \bar{X})$$

$$F = X * Y(Z + \bar{Z}) + \bar{X} * Y \quad F = Y * 1$$

$$F = X * Y * 1 + \bar{X} * Y \quad F = Y$$

$$F = X * Y + \bar{X} * Y$$

- $F = X * (\bar{X} + Y)$ ifadesini sadeleştiriniz.

$$F = X * (\bar{X} + Y) \quad F = 0 + XY$$

$$F = X * \bar{X} + XY \quad F = XY$$

Lojik Kapılar ve Doğruluk Tabloları:

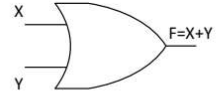
• Sayısal devrelerin temelini lojik kapılar, diyot, transistor, direnç, kondansatör gibi elemanlar oluşturur. Entegre devreler ise bu gibi devre elemanlarını tek cip içinde birleştirerek daha az güçle, daha hızlı ve daha az yer kapayacak şekilde çalıştırmayı amaçlamaktadır. Bu bölümde lojik kapılar ve bu kapılara ait doğruluk tabloları sunulmuştur.

Ve Kapısı (AND)



X	Y	F
1	1	1
1	0	0
0	1	0
0	0	0

Veya Kapısı (OR)

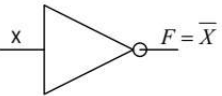


X	Y	F
1	1	1
1	0	1
0	1	1
0	0	0

- Ör: 3 girişli veya kapısını ve doğruluk tablosunu oluşturunuz:

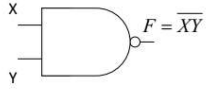
A	B	C	F
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	1

Değil Kapısı (NOT)



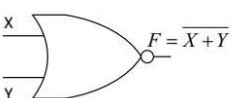
X	F
1	0
0	1

Ve Değil Kapısı (NAND)



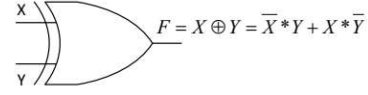
X	Y	F
1	1	0
1	0	1
0	1	1
0	0	1

Veya Değil Kapısı (NOR)



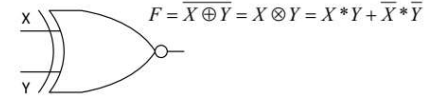
X	Y	F
1	1	0
1	0	0
0	1	0
0	0	1

Ya da Kapısı (XOR)



X	Y	F
1	1	0
1	0	1
0	1	1
0	0	0

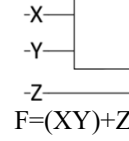
Ya da Değil Kapısı (XNOR)



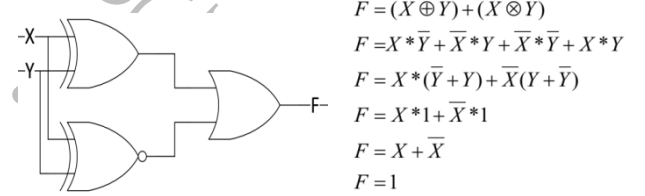
X	Y	F
1	1	1
1	0	0
0	1	0
0	0	1

1.3. Lojik Diyagramdan Matematiksel İfadenin Elde Edilmesi:

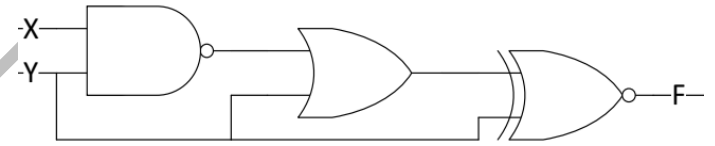
Aşağıdaki devrenin matematiksel karşılığını bulunuz.



Aşağıdaki devrenin matematiksel karşılığını bulunuz.



Aşağıdaki devrenin matematiksel karşılığını bulunuz.



$$F = (\bar{X}Y + Y) \otimes Y$$

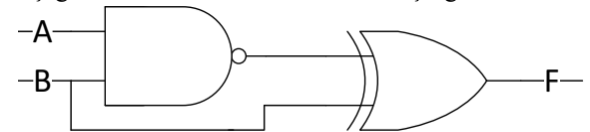
$$F = (\bar{X} + \bar{Y} + Y) \otimes Y \quad F = 1 * \bar{Y} + 1 * Y$$

$$F = (\bar{X} + 1) \otimes Y \quad F = 0 + Y$$

$$F = 1 \otimes Y \quad F = Y$$

Ödev:

Aşağıdaki devrenin matematiksel karşılığını bulunuz.



Ödev: Fonksiyonunu sadeleştirdikten sonra lojik diyagramını çiziniz.

$$F = (\bar{x} + xy)(\bar{y} + xy)$$

4. BÖLÜM

Programlama Dilleri

- Programlama dilleri teorisi bilgisayar mühendisliğinin bir koludur.
- İçerisinde
- Programlama dillerinin genel özelliklerini
- Programlama dillerinin sınıflandırılması
- Tasarım uygulama ve analiz konularını içerir.
- Programlama dilleri matematik yazılım mühendisliği dil bilimi ve hatta eğitim birimleri arasında yer alan disiplinler arası bir konudur.

- Programlama dilleri bilgisayar mühendisliğinin en temel konularının başında gelir

1. Yazılım geliştirme süreci

- Programlama dilleri yazılım geliştirme araçları olduğu için öncelikle yazılım geliştirme sürecini ele alalım
- Yazılım geliştirme süreci beş evrede incelenebilir.

Gereksinim analizi

- Bir yazılım belirli bir kullanıcı grubunun ortaya koyduğu ihtiyaçları karşılamak amacıyla geliştirmiştir. Kullanıcı gereksinimlerinin tasarlanacak ve geliştirilecek yazılım sistemi tarafından eksiksiz bir biçimde karşılanması beklenir.
- Gereksinimlerin açıkça ortaya koyabilmesi için kullanıcı grubu tarafından uygun bir formda net bir biçimde bir metin haline getirilmelidir.
- Bu aşamanın verimli geçebilmesi için yazılmış kullanacak ve geliştirecek olanların ortak bir çalışma yapması uygun olacaktır
 - Sistemin başarısı önceden ortaya konulan bu ihtiyaçların
 - yazılım tarafından ne kadar iyi karşılandığı,
 - kullanıcıların ihtiyaçlarını ne kadar doğru ortaya koydukları ve
 - ortaya konulan bu ihtiyaçların gereksinimleri ne kadar iyi yansıttığı sorularının cevaplanması ile ölçülebilir
- Bu evrenin çıktısı olarak beklenenlerin
- yazılı bir metne dökülmesi,
- kullanıcı el kitaplarının taslak halleri,
- maliyet ve fizibilite analizleri ve
- baş başam gereksinimleri olarak sıralanabilir.

Yazılım tasarımı

- Yazılım tasarımcıları birinci evredeki dokümanları ortaya koyduktan sonra sistem tasarımı ile ilgili tanımların verildi dokümanları ortaya koyar
- Bu dokümandan beklenen sistemi oluşturan tüm modüller ve bunların tüm arayüzlerinin tanımlanmış olmasıdır.
- Bu aşamada seçilen tasarım yönteminin kullanılacak olan programlama dilinin seçiminde büyük etkisi vardır

Kodlama

- İkinci evredeki tanımlamaların gerçekleştirilerek kodlandığı evredir
- Bu adımda programlama dilinin doğrudan uygulandığı tek adımdır. Çıktı olarak tümüyle çalışan bir yazılım sistemi beklenir.

Sertifikasyon

- Bu adımda ortaya konan yazılım kalite denetimi yapılarak son kullanıcı sunulur
- Sertifikaşyon işlemi tüm yazılım için tek seferde yapılabileceği gibi uygun olanı her modülün tek tek bu işleme tabi tutulmasıdır.
- Sertifikaşyon da dikkat edilecek husus yazılım gereksinimlerinde ortaya konulan ihtiyaçların ve beklentilerin karşılanıp karşılanmadığı ve arayüzlerin sorunsuz bir şekilde çalışıp çalışmadığıdır.

1.5.Bakım

- Bu adım belirlenen hataların giderilmesi ve yeni bileşenler eklenmesini kapsamaktadır.
- Bakım adımı maliyetler açısından ele alındığında önemli bir adımdır Çünkü deneyimler ortaya koymuştur ki bakım maliyetleri kimi zaman toplam yazılım maliyetini geçebilmektedir.

2. Yazılım geliştirme sürecinde programlama dilinin önemi

- Yazılım geliştirme sürecinin başarısı açısından seçilecek programı dili en önemli etkenlerden biridir. Bu nedenle seçilecek programlama dilinin ve bu dilde yazılmış yazılımın bazı temel kriterleri sağlaması gerekir bu kriterler şu şekildedir

Güvenilirlik

- Kullanıcılar kullandıkları yazılıma güvenmelidirler. Nadiren de ortaya çıksa oluşacak yazılım yada donanım ve ilgili hata mesajlarında dahi yazılımı olan güvenlerini kaybetmemelidirler. Bir yazılımın güvenliğinin oluşturulmasında programlama dillerinin kalite ölçütleri önem kazanır. Bu ölçütler şu şekilde sıralanabilirler:

Yazılabilirlik

- Ölçülebilmesi zor bir özelliktir. Problemin gerektirdiği veri giriş çıkış akış denetimi gibi unsurların programlama dilinde doğal olarak karşılanabilir olması ile ilgilidir.

- Programcı problemin çözümündeki önemli noktaların dışındaki detaylarla vakit kaybetmemelidir.
- Nesnel bir kriter olmamasına rağmen yüksek seviyeli dillerin düşük seviyeli dillere göre daha fazla yazılabilirlik özelliğine sahip olduğu söylenebilir.

Okunabilirlik

- Programın mantığını takip etmek, sınamak ve hata ayıklamak mümkün olabilmelidir. Programcının tarzına fazlaca bağlı olduğundan, yazılabilirlik gibi okunabilirlik de nesnel olmayan bir kriterdir. Bununla birlikte, dil ne kadar basit ise ve algoritmaların doğal olarak ifadesine ne ölçüde izin veriyor ise, kodun incelenerek programın hangi işlemleri gerçekleştirdiğinin anlaşılması o ölçüde kolay olacaktır. Örneğin, «goto» ifadesinin kullanmanın programın okunabilirliğinde güçlük yaratacağı açıktır.

Sıra dışı Durumları Karşılayabilme

- Programlama dili, hatalı giriş, aritmetik taşıma gibi arzu edilmeyen durumlara hazırlıklı olmalı ve bu durumlarda uygun çözümlen içermelidir.
- Bu özellikleri içeren bir programlama dili ile yazılmış bir sistemin, normal olmayan durumlarda dahi davranışı tam olarak tahmin edilebilir

Yazılım Bakıma Elverişli Olmalıdır

- Geliştirilmiş bir yazılımın hatalarının ayıklanması ve giderilmesi, yeni modüller eklenmesi veya çıkartılması, kullanıcı ihtiyaçları doğrultusunda geliştirilmesi, yazılım bakımının temel unsurlarıdır.
- Bir yazılımın bakımının yapılabilmesi için, bu yazılımın geliştirildiği programlama dilinin, güvenilirlik için de bir kriter olan okunabilirliğiyle birlikte değiştirilebilme yeteneğinin yüksek düzeyde olması gereklidir.

Yazılım Verimli Çalışmalıdır

- Yazılımın verimliliği öteden beri, hem algoritmaların hem de programlama dilinin seçiminde en önemli unsurlarında birisi olmuştur.
- Yazılımın verimliliği kullanılan programlama dilinin verimliliğine doğrudan bağlıdır.
- Verimlilik esas olarak programlama dilinin tasarım aşamasındaki en önemli hedeflerden biridir.
- Erken dönem programlama dillerinde, verimlilik programlama dilinin mevcut donanımı etkin ve hızlı bir şekilde kullanması ile ölçülen bir kavram olarak görülmüştür.
- Örneğin: FORTRAN esas olarak IBM 704 makinesi için tasarlanmıştır ve birçok kısıtlamaları (dizi boyutları, ifade şekilleri gibi) bu makineye göre ayarlanmıştır. Dolayısıyla FORTRAN IBM 704 üzerinde mümkün olan en yüksek verimlilikle çalışacak şekilde tasarlanmıştır. Verimlilik kavramı günümüzde bir hayli değişikliğe uğramış ve sadece işletim hızı ve bellek gereksinimleri ölçülür olmaktan çıkmıştır.
- Verimlilik ölçümünde bir program ya da sistemin oluşturulabilmesinde ve bakımının yapılmasında harcanacak emek ve zaman da önemli bileşenler olmuştur.

3. Programlama Dillerinin Tarihçesi ve Sınıflandırılması

- Charles Babbage'nin 1822'de fark makinesini keşfetmesinden bu yana bilgisayarların belirli bir görevi yapmalarına yönelik talimatları anlamaları ihtiyacı doğmuştur.
- Bu ihtiyaç programlama dili kavramını doğurmuştur.
- Programlama dilleri ilk olarak belirli bir programa belli bir işi yapmak için talimatların adım adım bilgisayara girilmesi şeklinde ortaya çıkmıştır.
- Daha sonraki zamanlarda programlama dilleri mantıksal dalanma, nesne yönelimi gibi özellikler kazanarak gelişim göstermiştir.
- Halen kullanılmakta olan programlama dilleri son 50 yıllık süreçte iki önemli evre geçirmişleridir. Bunlar erken dönem prog-

ramlama dilleri ve modem programlama dilleri olarak iki ana grupta toplanabilirler.

- Charles Babbage'nin fark makinesi sadece dişli çarkların ayarlanması yoluyla bazı aritmetik hesaplamalar yapabilen bir mekanik sistem olmasına karşılık, bu ayarlamalar ilk programlama dili olarak kabul edilmiştir.
- Zamanla bu fiziksel hareket şekli yerini elektrik işaretlere bırakmıştır. Bu şekilde bir programlama dili ilk olarak ABD hükümeti tarafından geliştirilen ENIAC bilgisayarı kullanılmıştır (1942).
- 1945 yılında, John Von Neumann bilgisayar programcılığının yönünü belirleyen iki önemli kavram geliştirmiştir.
- Bunlardan ilki, basit bir donanımın, karmaşık komutlar ile kontrol edilmesini öngören ve yeniden programlamayı basitleştirerek, programa göre donanım değişikliğini ortadan kaldıran paylaşımlı programlama tekniğidir.
- İkinci kavram ise programlama dilleri açısından oldukça önemli olan, koşullu denetim transferi kavramıdır.
- Bu kavram bir program kodunun yazılış sırasına göre işletilmesi yerine alt programların ve kısa kod parçalarının istenen herhangi bir sıraya göre işletilmesini öngörmektedir.
- Bu fikir program kodunun dallanabilmek için "if-then" gibi mantıksal ifadeler veya "for" gibi döngü deyimlerine göre oluşturulması gerektiğini ifade etmektedir.
- 1949'da Short Code adı verilen programlama dili ortaya çıkmıştır.
- Bu elektronik cihazları programlamaya yarayan ilk programlama dilidir.
- Bu dilde programcı, 0 ve 1'ler kullanarak program komutlarının değiştirilmesini sağlamaktaydı.
- Bu dil, karmaşık programlama dillerinin ilk örneği olarak kabul edilir.
- 1951'de Grace Hopper ilk derleyiciyi yazmıştır. Derleyici, programlama dilindeki ifadelerin bilgisayarın anlayabileceği 0 ve 1'lere dönüştüren bir programdır. Bu gelişme daha hızlı programlamaya imkân vermiştir.
- 1957 yılında ilk gelişmiş programlama dili olan FORTRAN ortaya çıkmıştır. Bu dil IBM tarafından bilimsel hesaplamalar için geliştirilmiştir.
- Dili içeriği oldukça basit olup kullanıcıya bilgisayarın donanımına alt seviyeden erişim imkânı sağlar.
- Günümüzde bu dil sadece "IF " "GOTO" "DO" ifadelerini içerdiği için tercih edilmemekle birlikte, o dönemde bu ifadeler büyük bir gelişim olarak kabul edilmiştir.
- Günümüzde kullanılan basit veri türleri, kaynağını FORTRAN dilinden almıştır.
- FORTRAN dili sayısal işlemlerde avantajlı bir dil olmasına rağmen giriş çıkış işlemlerin kontrolü açısından yeterli bir dil değildir.
- Bu nedenle giriş çıkış düzenlemelerinin oldukça önemli olduğu ticari uygulamalar için 1959'da COBOL dili geliştirilmiştir.
- COBOL özellikle iş dünyası için geliştirilmiş bir dil olup, sadece sayı ve metin türü veri türlerini içermekteydi.
- Aynı zamanda bu dil yukarıdaki veri türlerinin diziler ve kayıtlar gibi veri grupları halinde organize edilmesine olanak sağlamıştır.
- 1958'de MIT'de çalışan John McCarthy, yapay zeka araştırmaları için LISP dilini geliştirmiştir.
- Üstün uzmanlık isteyen bir alan için tasarlanması sebebiyle kendine özgü bir sözdizimi (syntax) yapısına sahiptir.
- Bu dilin diğer dillerden en büyük farkı içerdiği tek veri türünün "liste" olmasıdır.
- Liste veri türü parantezler ile sınırlandırılmış bir öğeler dizisini ifade eder.
- LISP dilinin kendi kendini değiştirme ve hatta geliştirme gibi kendine özgü bir yeteneği vardır.
- ALGOL dili 1958 yılında bir araştırma grubu tarafından bilimsel amaçlı kullanım için geliştirilmiştir. Bu dilin programlama açısından önemi,

- PASCAL, C, C++, JAVA gibi dillerin gelişmesi için ilk adım olması ve aynı zamanda BNF (Backus-Naar Form) olarak bilinen bir formal dilbilgisine sahip ilk programlama dili olmasıdır.
- ALGOL dili bir sonraki nesli olan ALGOL 68 ile birlikte, kendi kendini çağıran (recursive) fonksiyonlar gibi özellikler kazanmasına rağmen, bu özellikler kullanım zorluklarının da beraberinde getirmiş, bu nedenle PASCAL gibi daha kolay ve küçük programlama dillerinin geliştirilmesine neden olmuştur.
- PASCAL dili düzenli bir yaklaşım izlenerek, o dönem kullanımda olan COBOL, FORTRAN ve ALGOL gibi dillerin en iyi özellikleri birleştirilerek Niklaus Wirth tarafından 1968 yılında geliştirilmiştir.
- Diğer dillerdeki giriş çıkış işlemleri, matematiksel fonksiyonlar gibi özelliklerin en iyilerinin seçilmesi ile oluşturulan PASCAL dili oldukça başarılı bir dil olmuştur.
- Bunun yanı sıra PASCAL ile birlikte işaretçi veri türü de geliştirilmiştir.
- Ayrıca PASCAL dili "case" kontrol yapısını içeren ilk dil olma özelliğini taşımaktadır. PASCAL dili dinamik değişkenlerin kullanılması konusunda da yenilikler içermektedir.
- Dinamik değişkenler program çalışırken "new" ve "dispose" komutları ile oluşturulabilen ve silinebilen değişkenlerdir.
- Ancak PASCAL dili dinamik dizileri veya dinamik değişken gruplarını desteklememekte idi.
- PASCAL bu özellikleri bir sonraki nesli sayılan PASCAL-MODULA 2 sayesinde kazanmıştır. Ancak bu dönemlerde C dili kullanıcılar arasında popülerlik kazanmaya başlamıştır.
- C programlama dili New Jersey'deki Bell Laboratuvarı'nda Dennis Ritchie tarafından 1972'de geliştirilmiştir.
- İlk gelişmiş dillerden günümüzün modern dillerine geçiş PASCAL'dan C'ye geçiş ile olmuştur.
- C dilinin ataları B ve BCPL dilleri olmasına rağmen, C dili ile PASCAL dilleri arasındaki benzerlikler oldukça belirgindir.
- PASCAL'ın içerdiği bütün özellikler hatta "case" gibi yeni özellikler bile C dili içerisinde bulunmaktadır. C dili hızlı ve güçlü olmak adına işaretçi yapısını oldukça kapsamlı kullanmakta, ancak yine bu sebepten dolayı okunabilirliği zor bir dildir. C dili PASCAL dilindeki hataların büyük bir kısmını giderdiği için PASCAL kullanıcılarını kazanmayı başarmıştır.
- Ritchie, C dilini yine aynı dönemde üzerinde çalışılan Unix sisteminde kullanılmak üzere geliştirmiştir. Bu nedenle, C ve Unix verimlilik açısından en uyumlu platformları oluşturmaktadır. C dili Unix, Windows, MacOS ve çeşitli Linux sürümleri gibi işletim sistemlerinin geliştirilmesinde kullanılmıştır.
- 80'li yıllara doğru yeni bir programlama yaklaşımı olan "nesneye yönelik programlama" ortaya çıkmıştır. Bu yöntemde verimin parçaları programcı tarafından paketlenilebilen ve yönlendirilebilen birer nesne olarak ele alınmıştır. Bu yaklaşımdan etkilenen Bjarne Stroustrup "sınıf" veri türü içeren bir C uyarlaması (C with Classes) geliştirmiştir.
- Bu eklentiler grubu, yeni bir programlama dili olan C++ dilinin gelişiminin yolunu açmıştır. Böylece C++ dili 1983 yılında kullanıma sunulmuştur. C++ dili, C dilinin gücünü nesneye yönelik programlama kullanılarak organize etmek için geliştirilmiştir. Ancak nesneye yönelik programlama metodu kullanılırken C dilini üstün kılan hızı ve taşınabilir olma özellikleri korunmuştur.
- 1990'ların öncesinde interaktif TV geleceğin teknolojisi olarak görülüyordu. SUN Microsystems interaktif TV'nin bir özelliğe ihtiyacı olduğuna karar verdi. Bu özellik, bütün makinelerde sorunsuz çalışma veya taşınabilirlik idi. 1994'de Java bu ihtiyaç üzerine geliştirilmiştir. İnteraktif TV'nin başarısızlıkla sonuçlanmasından sonra, Java proje takımı giderek kullanımı artan ve popülerleşen Web üzerine yoğunlaştı. Bir yıl sonra Netscape kendi İnternet gezginini geliştirmek için Java'nın lisansını

nesne yönetim özelliği ile şu anda bilgisayar dünyasının en çok kullanılan dillerinden biridir.

- **C#**
- Nesneye dayalıdır. C++ m ve Java'nın pozitif yönlerini bünyesinde birleştirmiş yeni bir
- Programcıya Internet uygulamaları ve yerel uygulamalar yazmakta bazı kolaylıklar getirmiştir.
- İleriye dönük olarak Microsoft'un Java teknolojisine rakip olarak ortaya sürdüğü bir programlama dilidir ve Microsoft'un bu konulardaki (Internet uygulamaları) yelpazesini genişletmeye yönelik bir atılımdır.
- Microsoft teknolojileri kullanacak programcıların C#'ı öğrenmeleri zamanla gerekecektir ve öncelikle öğrenilmeye başlanması da avantaj getireceği açıktır.

• **Java**

- Nesneye yönelik dillerdendir. Son yıllarda geliştirilmiş bir dil olup modern ve yenilikçi altyapısı ve görsel özellikleri ve sürekli gelişen kütüphane (library) desteği ile gün geçtikçe kullanımı artan bir dil olmuştur.
- Java dili hemen hemen her alanda kullanılabilen esnek ve güçlü bir dildir.

• **VB.NET**

- Nesneye dayalı bir dildir. Visual Basic (VB) teki birçok özellik bu dilde yeniden yapılandırılarak değişmiştir.
- Yapısına bakılırsa VB den ayrı yeni bir dil geliştirilmiş denilebilir. Eklenen bazı özellikler ile VB de yapılamayan birçok işlem artık yapılabilmektedir ve OOP'nin özellikleri desteklenerek daha verimli kod yazmaya olanak sağlanmıştır.
- VB programcıların VB.NET'e geçişleri kolay olmayacak olsa da VB yerine VB.NET kullanımı gün geçtikçe artacaktır.
- VB.NET İnternet uygulamalarından yerel uygulamalara kadar kullanım imkanı geniş bir dildir.

• **Delphi**

- Pascal tabanlı bir dil olup nesneye yönelik programlama yapabilmek özelliği taşır.
- Öğrenilmesinin zor olmaması nedenleriyle çoğu bilgisayar programlama öğrencisinin tercih ettiği bir dildir. Görsel programlama özelliği taşır.

• **Pascal**

- Pascal Yapısal bir dildir, C 'ye benzerlik gösterir.
- Öğrenilmesinin zor olmayışı ve bilgisayar eğitimi veren okullarda okutulan bir ders olması sebebiyle kullanım alanı genelde üniversiteler ve bilimsel hesaplamalar yapan kurumlardır.

• **Visual Basic**

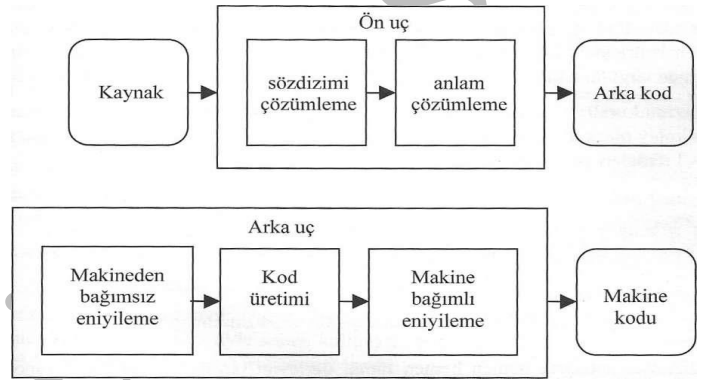
- Basic tabanlı bir dil olup öğrenilmesi kolay, kullanım alanı geniş bir dildir.
- Özellikle görsel uygulamalarda projenin arabiriminin hızlı yazılmasını sağladığı için genelde kullanıcı arabirimi tasarımlarında kullanılır.
- Kapsamlı veya çok kullanıcı uygulamalarda kullanılmaz. Kullanıcı sayısı az olan veya kısa sürede bitmesi gereken küçük ölçekli projelerde tercih edilen bir programlama aracıdır. Yoğun olarak kullanılmaktadır.

4. Programlama Ortamı

- Bir programlama ortamı programlama dili ile birlikte, birçok bileşenden oluşur.
- Bu bileşenler, sembolik olarak kodlanmış programın bilgisayar donanımı tarafından istenilen işlevleri yerine getirmesi için gereken tüm unsurlardır.
- Aslında programlama dili tek başına, ardışık olarak dizili olan sembollerden başka bir şey değildir.
- Aşağıda, bir programlama ortamının temel unsurları sıralanmıştır.
- Editör (Editör): Kaynak kodu oluşturmak ve gerektiğinde değişiklik yapmak için gerekli olan araçtır. Editörde yazılanlar, seçilen dilin komutlarından oluşan metinlerdir.
- Derleyici (Compiler): Editör tarafından bir bilgisayar dilinde yazılmış olan kaynak kodu, makine koduna çeviren bir bilgisayar programıdır.

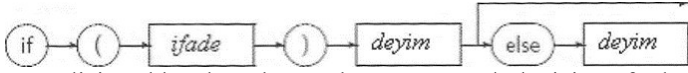
- Şekil 2 de bir derleyicinin genel yapısı verilmiştir. Ön uç (front end), kodun söz diziminin ve anlam yapısının, kullanılan dile uygunluğunu kontrol ederken,
- arka uç makine kodunu üretir. Ön uç, dile bağımlı iken, son uç makineye bağımlıdır.
- Kütüphane (Librarian): Nesne dosyalarından (object files) oluşan kütüphanedir.
- Bağlayıcı (Linker) : Programın içerdiği tüm nesne dosyalarını birleştirerek tek bir yürütülebilir dosya haline getirir.
- Yükleyici (Loader): Yürütülebilir dosyayı, diskten belleğe kopyalar.
- Hata ayıklayıcı (Debugger): Programcının hataları anlayabilmek için programın yürütülmesini adım adım kontrol edebilmesini sağlar.
- Yorumlayıcı (Interpreter): Bir programın kaynak kodunu doğrudan satır satır yürüten bir programdır.

Şekil 2. Bir derleyicinin genel yapısı.



5. Programlama Dillerinin Elemanları

- Programlama dili birçok bileşenin uyum içinde çalışması ile algoritmayı çalıştırılabilir koda dönüştürmektedir.
- **Söz Dizimi (Syntax)**
- Sıradan dillerde olduğu gibi, programlama dillerinin de bir söz dizimi (syntax) vardır.
- Bir programın söz dizimi, simgelerin geçerli olarak kabul edilebilmesi için hangi düzende yazılması gerektiğini belirleyen kurallar kümesidir.
- Söz dizimi için en bilinen formal notasyon, "Extended Backus Naur Form" (EBNF) olarak bilinir. Sözdizimi olarak doğru yazılmış bir kod anlamsal olarak doğru olmayabilir.
- C'de bir if deyiminin sözdizimi aşağıdaki kuralla verilir:
- if-deyimi ::=if (ifade) deyim [else deyim]
- Burada eğik yazılı olanlar, sözdizimsel ifadeler olup koyu yazılı ile yazılan isim ya da simgeler programda görünmesi gereken gerçek karakterlerdir.
- Tüm kurallar "::=" simgesini içerir. Bu simge "-den oluşmuştur" anlamını taşımaktadır. Diğer çeşitli simgeler de kuralları daha kısa ve anlaşılır yapmak için kullanılır.
- [] simgesi içerdiği ifadelerin seçimlik olduğunu, Böylece "else" kısmının seçimlik olduğu anlaşılmaktadır.
- { } simgesi sıfır ya da daha fazla tekrarları belirtir ;
- | sembolü de veya anlamına gelmektedir.
- Parantezler, değişken listesini tanımlamak için kullanılır.
- Değişken-tanıtımı ::= tür belirleyici {,belirleyici};
- Yukarıdaki ifade şu şekilde yorumlanır: Bir değişken tanıtımı, tür belirleyici ve takip eden bir belirleyiciden (örneğin bir değişken adından) oluşur.
- Seçimlik olarak birbirini aralarında virgüllü takip eden birden fazla belirleyici olabilir ve ";" ile tanıtım sona erer.
- Sözdizimi kuralları Şekil 3'teki sözdizimi diyagramları şeklinde verilirse anlaşılması daha kolay olacaktır.
- Daire ve oval şekiller gerçek karakterleri, dikdörtgenler ise sözdizimsel ifadeleri göstermektedir.



- Sözdizimsel hataların hemen hemen tümü, derleyici tarafından yakalanıp raporlanacaktır.
- Ancak bazı hatalar derleyicinin yakalayamayacağı türden olup çalışma esnasında hatalara neden olabilir. Aşağıda bu tür hatalar ile ilgili bazı notlar verilmiştir:
- Belirleyicilerin uzunlukları ile ilgili sınırlamalara dikkat edilmelidir. Örneğin ilk 10 karakter dikkate alınıyorsa «öğrencinin_numarası» ile «öğrencinin_notu» değişkenleri aynı değişkenler olarak alınacaktır.
- Birçok dilde küçük büyük harf ayırımı yoktur. Buna göre "SAYI" ile "sayi" aynı değişkenlerdir.
- C dilinde küçük büyük harf ayırımı olduğundan belirtilen iki değişken farklı değişkenler anlamına gelir.
- Bu tür dillerde hataya neden olmamak için genel bir kullanım belirlemek gerekir.
- Örneğin C dilinde değişkenler küçük harfle, sabitler büyük harfle yazılır.
- Açıklamalar için iki kullanım tarzı vardır. Fortran, Ada ve C++ açıklamalar sırasıyla C, ~ ve // ile başlar ve satırın sonuna kadar devam ederken, C ve Pascal'da sembolle başlar ve sembolle biter.
- C'de /*...*/ şeklinde, Pascal'da {...} veya (*...*) şeklindedir.
- Birbirine benzeyen ama farklı olan simgelere dikkat edilmelidir.
- Matematikten bilindiği gibi, "=" sembolü C ve Fortran'da atama operatörü olarak kullanılırken, C'deki "=" ve Fortran'daki «eq.» sembolü eşitlik operatörü olarak kullanılır.
- Anlamı doğru gibi görünse de, kullanımı yanlış olan aşağıdaki yazım şekline insanların eğilimi daha fazladır:
- If(a = b)
Anlambilim (Semantics)
- Anlambilim, bir programlama dilindeki bir ifadenin ne anlama geldiğidir.
- Dilin söz dizimi çok iyi anlaşılır olsa da, anlamını kavramak daha zordur.
- Aşağıdaki İngilizce cümlelerde, pen kelimesinin ilkinde kümes, diğerinde dolmakalem olduğunun bilinmesi ve buna göre cümlelerin anlamının çıkarılması gerekir.
The pig is in the pen. The ink is in the pen.
- Veri (Data)**
- Bir dili öğrenirken, insanların iki şeye odaklanmaları gibi bir eğilim vardır: deyimlere ve komutlara.
- Bu konular üzerinde çalışıldığında ve öğrenildiğinde, verinin yapısı ile ilgili konulara eğilim gösterilmelidir. Programlamadaki asıl kavramı tanımlayalım:
- Tür, bir değerler ve bu değerler üzerinde yapılabilecek işlemler kümesidir.
- C'de int'in gerçek anlamı şu şekilde ifade edilebilir: int sonlu sayıda değer içeren ve üzerinde birtakım aritmetiksel işlemlerin yapılabildiği bir türdür.
- Ada ve C++ gibi moderne dillerde, dilin türleriyle sınırlı kalmayıp, çözmeye çalıştığımız probleme uygun kendi türlerimizi oluşturma imkânımız vardır.
- Atama Deyimi (Assignment Statement)**
- Tüm sıradan programlama dillerinde bulunan deyim atama deyimidir. Atama deyimi genel olarak üç işin yürütülmesini sağlar:
- 1. Deyimin sağ tarafındaki ifadenin değeri hesaplanır.
- 2. Deyimin sol tarafındaki ifadeyi hesapla. İfade bir bellek hücresinin adresini değerlendirmelidir.
- 3. Adım (1) de hesaplanan değeri, adım (2) de elde edilen bellek hücresine kopyala.
- Tür Kontrolü (Type Checking)**
- Atamanın sonunda, sol taraftaki hesaplama bir bellek hücresinin adresini üretirken, ifade belirli bir türde bir değer üretir.
- Adresin ilişkilendirilmiş olduğu değişkenin türüyle, ifade türünün aynı olacağı konusunda bir garanti yoktur. Hatta, üretilen değer, değişkenin değer sınırları içinde olacağının da garantisi yoktur.

- Tür kontrolü, hedef değişkenin türüyle, ifade türünün uyumlu olup olmadığının kontrol edilmesidir.
- Bu, bir alt program çağrıldığında, gerçek parametrenin bir formal parametreye atanması durumunda da geçerlidir.
- Tür kontrolü ile ilgili olası yaklaşımlar:
- Hiçbir şey yapma; atamanın anlamlı olmasını sağlamak, programcının sorumluluğundadır.
- İfadeden ortaya çıkan değeri, sol tarafın gerektirdiği türe dönüştür.
- Güçlü tür kontrolü: Türler uyumlu değilse, atamayı yapma.
- Burada esneklik ile güvenilirlik arasında bir çelişkili durum bulunmaktadır.
- Ne kadar çok tür kontrolü yapılırsa, program o denli güvenilir olacak ancak, uygun bir tür kümesi tanımlamak için daha fazla çaba sarf etmek gerekecektir.
- Ters düşünülecek olursa, daha az tür kontrolü, daha kolay program yazılmasını sağlayacak ancak, hataları bulmak zorlaşacak ve programın güvenilirliği azalacaktır.
- Kontrol Deyimleri (Control Statements)**
- Atama deyimleri normal olarak yazıldığı şekilde yürütülürken, kontrol deyimleri yürütmenin sırasını değiştirmek maksadıyla kullanılır.
- Yapısal programlama, kontrol deyimlerinin sınırlandırıldığı programlama stiline verilen addır. İyi yapılandırılmış bir kontrol deyimi, iki şekilde olabilir:
- İki ya da daha fazla seçenekten birinin seçilmesi, if ya da case deyimlerinde olduğu gibi.
- Tekrarlı ifadelerin çevrim durumları, for ya da while deyimlerinde olduğu gibi.
- Döngü deyimleri özellikle iki sebepten dolayı önemlidir: Yürütme zamanının çoğu, döngü deyimlerinde harcanır ve, çoğunlukla döngünün başında ya da sonunda yanlış kodlama yapılmış olabilir.

Alt Programlar (Subprograms)

- Alt program, programın farklı yerlerinden defalarca çağrılabilen, içinde veri tanımlamalarının ve yürütülebilir deyimlerin bulunduğu program birimleridir.
- Alt programlar "Başlangıçta aynı program parçalarını tekrar kullanılması gerektiğinde kullanılırken, modern bakış açısına göre programlamanın temel elemanlarından biri olarak kullanılır olmuştur.
- Bir alt program çağrıldığında, parametre olarak isimlendirilen değerler alt programa gönderilmiş olur.

Modüller (Modules)

- Buraya kadar ele alınan elemanlar, program yazmak için yeterli olan ancak, yazılım sistemi oluşturmak için yeterli olmayan elemanlardır.
- Büyük programlar, bir programcı takımı tarafından geliştirilirler. Bir alt program 40-50 satırdan fazla olmamalıdır. Aynı şekilde 1600-2500 satırdan daha büyük kodları kontrol etmek de anlamak da zordur.
- Modern programlama dilleri, fazla miktarda veri ve alt program içeren modüller içerir.
- Hataların önlenmesi ve yanlış anlamalara yol açmayı önlemek için, derleme süresince modüller arasındaki ara yüzlerin kontrolü avantaj sağlar.

6. Özet

- Bilgisayarların gelişim sürecinde programlama dilleri de pekçok aşama ve evrim sonucunda bugünkü şekillerini almışlardır. Bu açıdan, programlama dillerinin tarihçesi bilgisayar mühendisliği açısından önemli bir konudur.
- Programlama dilleri, genel olarak, uygulama alanına göre ve seviyelerine göre sınıflandırılabilir. Nesneye yönelik programlama modern programlama dillerinin tümünde yer alan önemli bir özelliktir.

- Nesne kendisini tanımlayan veriler ve bu veriler üzerinde yapılacak tüm işlemler ile bir bütün olarak düşünülür.
- Aynı gruptan nesnelere genel bir isim verilip yeni bir veri türü yaratılmaktadır. Bir genelleme olan bu yapı "sınıf" olarak adlandırılır.
- Modern programlama dilleri bir programlama ortamı üzerinde oluşturulur. Programlama ortamında editör, derleyici, kütüphane, bağlayıcı, yükleyici, hata ayıklayıcı, yorumlayıcı bileşenleri bulunmaktadır.
- Bir programın söz dizimi, simgelerin geçerli olarak kabul edilebilmesi için hangi düzende yazılması gerektiğini belirleyen kurallar kümesidir.
- Söz dizimi için en bilinen formal notasyon, Extended Backus Naur Form (EBNF) olarak bilinir.
- Anlambilim bir programlama dilindeki bir ifadenin ne anlama geldiğidir.

Ayrıca bir programlama dilinin temel elemanları arasında atama deyimi, tür kontrolü, alt programlar gibi kavramlar da bulunmaktadır.

YUKARI TARAF VİZEYE KADAR Kİ BÖLÜMDÜR.

VİZE SONRASI KONULAR

5. BÖLÜM

İşletim Sistemleri

- İşletim sistemi, en kısa açıklama ile bilgisayarın sahip olduğu kaynaklar ile bilgisayar kullanıcısı arasında arayüz görevi yapan programlar topluluğudur;
- amacı ise, bilgisayar kullanıcılarına programları çalıştırabilecekleri bir ortam hazırlamak ve bilgisayar kaynaklarını donanım olsun yazılım olsun etkin bir şekilde kullanılmasını / paylaşılmasını sağlamaktır.
- İşletim sistemi konusu genel olarak bilgisayar mühendisliği alanında geniş yer tutar; hem yazılımcılar için hem sistem mühendisleri için iyice bilinmesi gereken bir konudur;
- Teknik açıdan, işletim sistemi tanımı, "bilgisayar sistemini oluşturan donanım ve yazılım nitelikli kaynakları kullanıcılar arasında kolay, hızlı ve nitelikli bir işletim hizmetine olanak verecek biçimde paylaşırken, bu kaynakların kullanım verimliliğini en üst düzeyde tutmayı amaçlayan bir yazılım sistemi" olarak açıklamıştır [Saatçi-1993].
- Bu doğru bir tanımdır; ancak bir şey daha eklemek gerekir ki "İşletim sistemi yalnızca kullanıcılara hizmet etmez; bir de diğer programlara da hizmet edebilir."
- Bilgisayar sisteminin genel olarak dört ayrı bileşeni vardır:
 1. Donanım (işlemci, bellek, saklama birimleri, G/Ç birimleri)
 2. İşletim sistemi
 3. Uygulama programları ve
 4. Kullanıcılarıdır
- İşletim sistemi, kullanıcı açısından bir arayüzden oluşur; bu kullanıcılara ilgili programları çalıştırma, etkileşimde bulunma ve sonuçları görme imkanı verir.
- Bu arayüz aracılığıyla, kullanıcı bilgisayarı kolayca kullanır.
- Çünkü, artık günümüzde, görsel arayüz ortamları de facto olmuştur. Teknik açıdan işletim sistemi ise birçok kavramı bilmeyi gerektirir.
- Bunlar kısaca
 - tek-kullanıcı,
 - çok-kullanıcı,
 - giriş/çıkış birimleri,
 - görev yönetimi,
 - birlikte çalışma, kilitlenme,
 - bellek yönetimi,
 - dosya yönetimi,
 - güvenlik,
 - ağ iletişimi,
 - sistem çağrıları vb. verilebilir.

1. Bilinen İşletim Sistemleri

- Genel amaçlı ve özel amaçlı olarak tasarlanmış birçok işletim sistemi vardır; örneğin, Windows, Linux, UNIX, Macintosh hemen akla gelen işletim sistemleridir.
- Bu bölümde bilinen işletim sistemlerine kısaca değinilmiştir.

WINDOWS AİLESİ

- "Windows", kullanıcıya grafik arayüzler sunarak etkileşim sağlayan Microsoft firmasının geliştirdiği bir işletim sistemleri ailesidir.
- Microsoft'un ilk işletim sistemi olan MS-DOS'tan farklı olarak Windows'ta aynı anda çok sayıda programla çalışılabilir.
- Windows ailesinin son üyesi Windows 10'dur. Microsoft Windows ailesi, ilk zamanlar IBM PC için geliştirilen MS-DOS üzerine bir grafik arayüz koyularak başlamıştır.
- Daha sonra kendi Windows çekirdeği üzerine sürümleri geliştirilmiştir.

LINUX AİLESİ

- Linux ilk olarak Linus Torvalds tarafından yazılmaya başlanmıştır. Daha sonra birçok katılımcı sayesinde artarak gelişmiştir.
- Linux, aslında komple bir sistem değildir; işletim sistemi çekirdeğine verilen isimdir. İşletim sistemi çekirdeği üzerine eklenen uygulamalarla paketlenerek kurulumu hazır hale getirilmesine "linux dağıtımı" adı verilmektedir.
- Linux çekirdeği kullanılarak oluşturulan işletim sistemlerine linux dağıtımları adı verilir. Linux, oluşturulan bu dağıtımların çekirdeğidir.
- Kullanım amacına göre dağıtımlar şekillenmiştir. Bu dağıtımlarda değişmeyen tek nokta çekirdektir;
- dağıtımlar içinde farklılıklar yapılandırma araçları, paketleme ve güncelleme yöntemleri, kitaplık ve geliştirme dizinlerinde değişiklikler biçiminde olmaktadır.
- Günümüzde yaygın olarak kullanılan ve bilinen dağıtımlar:
 - Pardus
 - Debian
 - Ubuntu
 - Suse
 - Slackware

UNIX İŞLETİM SİSTEMİ

- Unix işletim sisteminin ilk uyarlaması 1969 yılında BELL laboratuvarlarında çalışan "Ken THOMPSON" tarafından geliştirildi daha sonra gruba "Dennis RITCHIE"nin katılımıyla ilk uyarlaması üretildi.
- Ritchie ve Thompson Unix üzerinde uzun süre çalıştılar; daha sonra ikinci uyarlaması üretildi. Üçüncü uyarlaması da assembly dili yerine genel olarak C programlama diliyle yazıldı.
- C programlama dili de, ilk olarak, Bell laboratuvarlarında Unix'i desteklemek için geliştirildi.
- Unix, C diliyle yazıldığında sisteme çoklu-programlama ve diğer gelişmeler de eklendi.
- Unix işletim sistemi, aynı zamanda, birçok teknik özelliğiyle diğer işletim sistemlerine öncü olmuştur.
- UNIX Tasarım ilkeleri
 - Unix, zaman-paylaşımı ilkesi göre çalışır;
 - dosya sistemi ağaç yapısındadır;
 - kullanıcıların kendi alt dizinleri oluşturulur.
 - Unix işletim sistemi, günümüzde de yaygın olarak kullanılmaktadır. Özellikle kurumlardaki büyük ölçekli sistemlerde bulunan işletim sistemi Unix'tir.
 - Masaüstü kullanıcıları için o kadar yaygınlaşmamıştır; ancak, kurumsal uygulamalarda bir sunucu işletim sistemi olarak ve çok kullanıcı bir sistem olarak genişçe bir kullanımı vardır.

VM İŞLETİM SİSTEMİ

- VM terimi, "Virtual Machine" sanal makine sözcüklerinin İngilizce baş harflerinden türetilmiştir; yani VM bir tür sanal bilgisayar veya görüntü bilgisayar olarak adlandırılabilir.
- Bir görüntü bilgisayar gerçek bir bilgisayarın yanıltıcı bir görüntüsüdür; tek bir gerçek sistemi birden fazla gerçek bilgisayarmış gibi gösteren bir sanal/görüntü bilgisayar, işletim sistemi tarafından sağlanır.
- Aynı gerçek sistem üzerinde işletim sistemi tarafından oluşturulan sanal/görüntü bilgisayar kullanıcıları, kendilerinin gerçek bir sistemde çalıştığını sanırlar.
- 1940 ve 1950'li yıllarda, bilgisayar sistemleri tek kullanıcı idi.
- Bilgisayar kullanıcısı, gerçek makinenin konsolunda, görünür makinenin tüm olanakları ve kullanıcıya uygunluğu ile otururdu.
- Bir iş işletimi, günümüzün kişisel bilgisayarı gibiydi. Eğer bilgisayar bir cevaba/girdiye ihtiyaç duyar ve kullanıcı onu hemen girmezse, sistem boş duruma düşer.
- O sırada gerçek sistemin kaynakları başka kullanıcı için kullanılabilir ve böylece gerçek sistemin kullanım verimi artmış olur.

2. İşletim Sisteminin Görevleri

- Bilgisayar işletim sistemlerinin görevleri, en genel anlamda, bilgisayar sisteminin sahip olduğu donanımsal kaynakları ve onun üzerine daha önceden yüklenmiş/kurulmuş olan yazılım parçalarını, kullanıcılara ve yine ihtiyaç duyulduğunda sistemin kendisine paylaşmak, verimli kullanılması sağlamaktır.
- Bir bilgisayar ile kullanıcı arasındaki iletişim süreci aşağıdaki şekilde gösterilebilir.
- Görüleceği gibi hiyerarşinin en üstünde olan insan ile en altta olan bilgisayar donanımı arasında bellek yönetimi, işlemci yönetimi, çevre birimlerin yönetimi, derleyiciler, editörler vs gibi birçok birim vardır.



- Temel bilgisayar kaynakları donanım tarafından sağlanmaktadır;
- Bunlar
- saklama birimi
- bellek,
- işlemci,
- çevre birimleri
- olarak sıralanabilir. Bu kaynakların kullanımı, uygulama programlarıyla aracılığıyla gerçekleştirilmektedir.
- İşletim sistemi, uygulama programlarıyla donanım arasındaki iletişimi sağlamaktadır.
- İşletim sistemi bir kontrol programıdır denilebilir. Kontrol programının amacı, kullanıcı programlarının çalışmasını sağlamak, bilgisayarın uygunsuz kullanımına ve hatalara yol açmasına engel olmaktır.

İşletim Sistemlerinde Temel İşlevler:

- Yazılım-donanım bütünlüğünün sağlanması
- Kaynakların yönetimi
- Kullanıcı ile sistem arasındaki ilişki, uyum düzeninin kurulması.

3. İşletim Sistemlerinin Gelişim Evresi

- 1940'lı yıllardan günümüze kadar olan süreç içerisinde donanımsal alt yapının gelişmesini paralel olarak işletim sistemlerinin mimarisi ve kullanıcı arayüzlerinde büyük değişiklikler olmuştur.

- İlk başlarda kullanıcı arayüzleri basit, satır tabanlı iken günümüzde grafik tabanlı görsel arayüzler geliştirilmiştir.
- Örneğin, Windows işletim sisteminin pencereleri de bu görsel tabanlı çözümlerden birisidir.
- Halbuki, önceleri, siyah bir pencere içerisinde fare olmaksızın komutların yazılarak yürütüldüğü bir yapı vardı bu durum hala bir seçenek olarak işletim sistemleri tarafından sağlanmaktadır. Örneğin, Windows ortamında «command prompt» penceresi böylesi bir ortam sunmaktadır.
- İşletim sistemlerinin kullanım yönünden geçirmiş oldukları evreler 1970'li yıllara kadar aşağıdaki gibi verilebilir;
- 1946 – 1952 Komut komut yürütme {Instruction-by-instruction processing}
- 1952 – 1957 İşten işe yürütme (Job-by-job processing)
- 1957 – 1962 Toplu işleme (Batch processing)
- 1962 – 1967 Çoklu programlama (Multi-programming)
- 1967 - Zaman paylaşımı (Time sharing)
- daha sonra görsel arayüzle etkileşim ortaya çıkmıştır; ve sanki tüm evrelerin özellikleri, yeni eklemelerle modern bilgisayarlarda var olmaya başlamıştır.
- "Komutla komut yürütmede bilgisayarın yapacakları işler el ile denetlenmekteydi.
- Şöyle ki, günümüzde «operatör» olarak bilinen kişi, o zamanlar iyi bir programcı veya bir bilgisayar mühendisi olmak zorundaydı; çünkü yapacağı iş bayağı teknikti.
- Arşivleme birimi olan teyplere kasetleri takma, kartları okuyucusuna yerleştirme vs gibi hazırlıkları yapıp programı başlatırdı.
- Operatörün bir iş için gereken hazırlıkları yapma süresi yerleştirme zamanı (set-up- time), başlatılan işin bitimine kadar olan süreye de çalışma zamanı (runtime) denir.
- Komutla komut yürütmede yerleştirme zamanı çalışma zamanından da oldukça uzundu. Dolayısıyla, bilgisayar sistemlerinin daha verimli bir şekilde kullanılabilmesi için "bir işten diğer bir işe" geçiş işlemlerinin otomatikleştirilmesi yolları aranmaya başlanmıştır.
- Toplu işlemiden sonra en büyük gelişme çoklu-programlama sistemleridir.
- Bu gelişmenin temelini kesme (interrupt) kavramı oluşturmuştur.
- Kesme, işlemcinin görevini sürdürürken, gelen işaretlerle yaptığı görevi bırakıp yeni bir göreve daldanmasıdır.
- Ancak, bu işlemi yaparak eski görevine ait bilgileri bir yerde saklar ve kesme işi bittikten sonra eski görevine kaldığı yerden devam eder.
- Önceleri kesme mekanizması daha çok endüstriyel kontrol uygulamalarında ve çevre birimlerinin bağlantısında kullanılmaktaydı.
- Daha sonraları, 1960'ın ilk yarısında böyle bir mekanizma yardımı ile birçok programın aynı anda aynı işlemciyi (CPU) kullanabileceği düşünülmüştür.
- Şöyle ki, programlardan birisi bir G/Ç işlemi beklemek zorunda kalacak olursa, o G/Ç işleminin yapılması beklenirken diğer bir program işlemciyi kullanabilir ve bu program bir giriş çıkış işlemi yapma durumuna geldiğinde işlemciyi üçüncü bir programa verilebilir.
- İşte bilgisayar sistemlerinin bu şekilde kullanılmaları bugün çoklu programlama (multiprogramming) olarak adlandırılmaktadır.
- Çoklu-programlama yönteminden başka şekilde de yararlanılmaktadır. Birçok program aynı işlemciyi belirli sürelerde (quantum) ele geçirip kullanabilirler.
- Bu şekilde, o anda sistemde bulunan bütün programların paralel olarak ilerledikleri izlenimi verilebilir.
- Özellikle bu programların her biri sisteme yakın veya sisteme uzak terminallerle ilişkili olursa, terminal kullanıcılarının her

birine sanki o sistemi yalnız onlar kullanıyormuş izlenimi verilebilir.

- İşte bu çalışma şeklinin uygulandığı sistemlere "çoklu-erişim" (multi-access) ve çalışma şekline de etkileşimli (interactive) veya "zaman paylaşımı" (time-sharing) adı verilmektedir.

4. İşletim Sistemi Türleri

- İşletim sistemleri çeşitli açılardan sınıflandırılabilirler. Örneğin;
 - a) Kullanıcılara sağladıkları çalışma ortamı
 - b) Kullanıcıların sisteme erişim biçimleri
 - c) Tasarım ve mimarisinde izlenen yaklaşımlara göre
- birbirine dolaylı olarak bağımlı üç boyut üzerinde incelenebilir.
- Bir işletim sistemi, ilk iki boyuttaki özelliklerinden yalnız birini taşıyabileceği gibi bunlardan çalışmayan birkaçını da birlikte bulundurabilir.
- Bir işletim sistemi, aynı anda yalnız bir görüntü sistem kurma olanağını sağlıyorsa, bu sistemin "tekli programlama" düzeninde çalıştığı, eş zamanlı birçok görüntü sisteminin kurulmasına olanak sağlıyorsa, "çoklu programlama" düzeninde çalıştığı söylenir.
- Birlikte çalışan görüntü sistemleri, işlemci dışındaki kaynaklar için keşifbiliyorsa, işletim sistemi eşzamanlı kaynak paylaşımına olanak tanımaktadır.

Tekli Programlama (Monoprogramming)

- Tekli programlamaya dayalı çalışılan bir sistemde, aynı anda yalnızca bir görüntü ortam kurulduğundan, kullanıcı sistemin tüm kaynaklarını kullanabilir.
- Çalışma sürecinde oluşabilecek hatalar başka bir kullanıcıya yansımayaacağı için, korunma önlemleri, sadece işletim sistemi ile kullanıcı arasında olacağı öngörülür.
- Dolayısıyla tekli programlama düzeninde kaynak atama, sistem bütünlüğünü koruma vs. gibi sorunlar kolayca çözülebilir.

Çoklu Programlama (Multiprogramming)

- Çoklu programlamaya dayalı sistemler, ilk zamanlar, işlemcinin boş olarak beklediği süreleri değerlendirmek için tasarlanmıştır.
- Sistemde çalışan herhangi bir iş/program parçası giriş/çıkış, senkronizasyon vs gibi nedenlerle bekleme durumuna geçtiğinde işlemcinin başka bir işe başlaması ve böylece işlemcinin kullanım verimliliğinin yükseltilmesi amaçlanmıştır.
- Genel olarak işlemci ile giriş/çıkış birimlerinin çalışma hızları arasındaki fark büyüktür; dolayısıyla işlemci birim giriş/çıkış işlemini beklemek yerine başka bir işi yürütür. Bu da sistemin kullanım verimliliğini artırır.

Çok Görevli İşlem (Multitasking)

- Bir sistemde bağımsız çalışabilen en küçük işletim birimi olarak tanımlanan görev, bilgisayar ortamında işlerin yürütülmesini sağlayan temel araçtır.
- Bir işin sistemde çalışması, işletim sisteminin bu işe en az bir görevi ataması ve yürütmesiyle gerçekleşir.
- Adımları sırayla uygulanacak bir iş, sırası ile derleme, bağlama, uygulama adımlarını karşılayan programlar çalıştıracak tek bir görev tarafından yürütülebilir.
- Ancak, işteki bazı adımları paralel olarak uygulamaya yöneldiğimizde, birlikte yürütülecek her adım içi ayrı bir görevin kullanılması gerekir.
- Böyle bir çalışma ortamını kurma olanağını sağlayan işletim sistemleri, çok görevli işlem yapan sistemler olarak tanımlanır.

Sistem Kullanım Biçimine / Erişime Göre Sınıflama

- Bir bilgisayar sisteminde hizmet üretim süreci,
 - hazırlık + sunuş + işletim + sonuçlama
 - olarak tanımlanan evrelerden oluşur.
- İşletim sistemlerinde, işletim dışında kalan evrelerin düzenleniş biçimleri, kullanıcıların bilgisayar sistemine nasıl erişeceklerini, gereksedikleri hizmeti alırken nasıl davranacaklarını, doğrudan belirleyen etmenlerdir.

- Bir işte, çalışma ortamının hazırlanması, uygulanacak programın işletim sistemine aktarılması ve sonuçların kullanıcıya iletilmesinde benimsenen yaklaşımlara göre işletim sistemleri;
 - Adanmış işlem (Dedicated processing)
 - Toplu İşleme (Batch processing)
 - Etkileşimli işlem (Interactive processing)
 - yapan sistemler biçiminde sınıflanmaktadır.

Adanmış İşlem (Dedicated Processing)

- Adanmış işlem, bilgisayar sisteminin belirli bir süre için, tümüyle bir kullanıcının hizmetine verildiği çalışma türüdür. İlk kuşak bilgisayar sistemlerinde ve yetmişli yılların minibilgisayarlarında yaygın olarak kullanılmıştır.

Toplu İşleme (Batch Processing)

- Bir bilgisayar sisteminde verimi arttıracak en etkin yol uygulama sürecinin her adımında yinelenen hazırlanış ve sunuş evrelerinde harcanan süreleri kısaltmaktır.
- Bu da ön gereksinimlerin önceden planlanıp, düzenlenmesi ve doğrudan insan etkeninin katılmadığı otomatik bir iş akış mekanizmasının kurulmasıyla gerçekleştirilebilir.
- Toplu işlem bu ilkelerin uygulandığı ilk çalışma düzenidir.
- Toplu işlem, bilgisayar sistemlerinin daha verimli kullanılmasını sağlayarak, iş başına düşen sistem giderlerinin azalmasına yol açmıştır.
- Bu olumlu yönün yanısıra, toplu işlemin sakıncalı sayılabilecek iki yönü bulunmaktadır. Bunlardan ilki, iş yönetiminin durgun ve iş denetim dilinin olanaklarıyla sınırlı bulunmasıdır.
- Kullanıcı, işletimde oluşan ve öngörmediği durumları çözümlemek için, işin sonuçlanıp kendisine ulaşmasını beklemek zorundadır. Doğru bir sonuç alabilmek için, işi yeniden sisteme vermesi gerekmektedir.
- İkinci sakınca, sonuçlanma evresinden kaynaklanmaktadır.

Etkileşimli İşlem (Interactive processing):

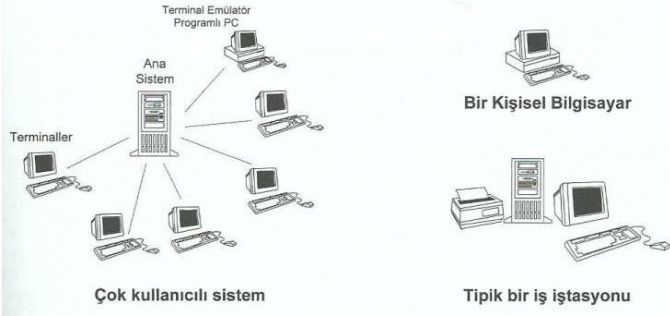
- Etkileşimli işlem kullanıcılara işlerini anında yönlendirme sonu anında aracısız elde edip, hızla önlem alma olanağını sağlayan çalışma türüne verilen addır
- Zaman paylaşımı (time sharing), «kullanıcılara bir bilgisayar sistemini, aynı anda birçok kişi kullanmasına karşın tek başına çalışıyormuş izlenimi veren ve bu sistemi birçok uygulama/kullanıcı arasında bölüştüren, etkileşimli işlem yaklaşımına» verilen isimlendirmedir.

Yapılarına Göre Sınıflama

- İşletim sistemlerini gerçekleştirirken benimsenen ilkeler, bu sistemleri sınıflamada kullanılabilecek ölçütlerdir. Örneğin, sistemde görevler arası zaman uyumunda uygulanan yaklaşımlara göre, işletim sistemlerini yapılarına göre iki grupta toplanabilir:
 - a. İletiyeye dayalı sistemler (message oriented systems)
 - b. Yordam çağırma dayalı sistemler (procedure oriented systems)
- Bunun dışında sistemde temel birim varsayılan öğeler üzerinde (örneğin görev, iş, dosya, program vb.) yapılan işlemler de, ayırıcı etken sayılmaktadır.

Kullanıcı Sayısına Göre Sınıflama

- Kullanıcı sayısına göre tek-kullanıcı ve çok-kullanıcı; kullanım amacına yönelik olarak genel amaçlı ve özel amaçlı; proses işleme şekline göre
 - tek prosesli (single processing),
 - çok prosesli (multi processing);
- zamana duyarlılığı açısından gerçek zamanlı (real time) ve
- gerçek zamanlı olmayan işletim sistemi olarak sınıflanabilir.



- Bir işletim birden çok sınıflamaya girebilmektedir.
- Örneğin DOS, genel amaçlı, tek kullanıcıli, tek prosesli ve gerçek zamanlı olmayan bir işletim sistemi iken Unix yine genel amaçlı, ancak, çok kullanıcıli, çoklu prosesli ve gerçek zamanlı işletim sistemidir.
- Çok kullanıcıli sistemlerde, yukarıdaki şekilde görüldüğü gibi bir ana sistem olur; terminal olarak adlandırılan ve aslında klavye ve monitörden oluşan bir erişim birimi bağlanır.
- Kullanıcılar, ana sistemi kendilerine tanınan erişim hakları çerçevesinde zaman paylaşımı ilkesine göre kullanırlar.
- Ana sisteme bağlanan kullanıcıların sayısı veya kullanıcıların başlattıkları program sayısı arttıkça paylaşımda ayrılan zaman miktarı azalır.
- Diğer bilgisayarlar üzerinden de ana sisteme bir terminal bağlantı yapılır.

5. İşletim Sistemi Yapısı/Mimarisi

- İşletim sistemi bilgisayarın donanımsal ve yazılımsal kaynaklarıyla sıkı sıkıya bağlıdır; bir yandan kaynakları adil bir şekilde kullanırken, bir yandan da, işlevini yerine getirebilmesi için o kaynakları kendisi kullanılır.
- En genel olarak bir işletim sistemi yapısında aşağıda listelenen birimler vardır:

- donanım ve çevre birimler
- bellek yönetim birimi
- dosya ve G/Ç yönetim birimi
- çekirdek katman
- sistem çağrıları
- kullanıcı arayüzü
- uygulama programları

Kaynak Paylaşımı

- Bilgisayar sisteminin sahip olduğu kaynaklar, çoğu zaman tek bir kullanıcı tarafından tümüyle tüketilemez.
- Boşta kalan böylesi kaynaklar, sistemin ortalama veriminin düşürdüklerinden, bunların eşzamanlı çalışacak başka kullanıcıların hizmetine kullanılması, sistem yapımcılarının çözmesi gerek başlıca sorunlardan biri olmuştur.
- İşlem birimi ve bellek, altmışlı yıllardan başlayarak yetmişlerin ortalarına kadar, toplam eder içindeki oranları nedeniyle, bir sistemde paylaşılması amaçlanmış ilk kaynaklardır.
- Bilgisayar sistemlerindeki kaynak paylaşım sorunu, ekonomik gerekçeler kadar uygulamanın zorunlu kıldığı nedenlere de dayanmaktadır.
- Örneğin rezervasyon, stok denetimi, banka hesapları vb. uygulamalarda, birçok kullanıcının ortak bir veritabanını bölüşmesi gereklidir.
- Süreç denetimi gibi, eşzamanlı işlevlerin birlikte yürütülmesi gereken uygulamalarda ise bilgisayar sistemi, paralel çalışan bağımlı/bağımsız birçok görev arasında paylaşılır.

İş (Job)

- Bağımsız bir bütün olarak gerçekleştirilmesi istenen; işletim sisteminin de diğerleriyle ilişkilendirmeden ele alacağı hizmet kümesi "iş" olarak adlandırılır.
- İşler, bir veya daha çok programın ayrı ayrı uygulanacağı alt adımlardan oluşabilir.
- İşler, genellikle adımların ard arda uygulanacağı biçimde düzenlenir. Her adım, bir öncekinin sonuçlanması üzerine devreye girer.

• Ancak, kullandıkları fiziksel kaynaklar yönünden çelişmeyen bağımsız adımların, paralel olarak uygulanabilmeleri de olanaklıdır.

Görev (Task) ve İşlemcinin Anahtarlanması

- Bir bilgisayar sisteminde hizmet sağlama, bağımlı veya bağımsız birçok işlevin birlikte yürütülmesiyle gerçekleştirilir.
- Çeşitli nedenlerle, işlemcinin yürütmekte olduğu işi zaman zaman bırakıp, daha öncelikli yeni bir işlere yönelmesi, sonramda bıraktığı işe dönüp, yarıda bıraktığı noktadan sürdürmesi gerekebilir.
- Bu işleme işlemcinin anahtarlanması denilir.
- Anahtarlanmayı yapabilmek için,
 - a. İşlemcinin o andaki durumunun saklanması
 - b. İşlemci içerisindeki birtakım saklayıcıların yeni işle ilgili verilerle güncellenmesi
- gerekir. İşletim sistemleri bu amaçla, birlikte yürütülecek her işlevin çalışma ortamını tanımlayan bir yapı kurar. Bu yapıda işlemcinin anahtarlamada kullanılacak verilerin yanısıra, işi yürütmek için gereksinim duyulan kaynaklarla ilgili bilgi tutar.

Bilgisayar Sistemi Kaynakları

- İşletim sistemleri açısından bir bilgisayar sisteminin kaynakları dört ayrı grupta toplanabilir
- Bellek
- İşlemci (CPU)
- Çevre birimleri (yazıcılar, terminaller, diskler vb.)
- Veri (Dosya, veritabanı vs.)
- Bir bilgisayar sisteminin kaynaklarının, o sistemin performansını en yüksek düzeyde tutabilecek bir şekilde paylaşımlı olarak kullanmasını sağlayan mekanizmaya «yönetim» adı verilir.
- Bir sistemin performansı temel olarak beş ayrı kriterle ölçülür:
- **Verimlilik:** İşletim sistemleri genellikle karmaşık programlardan oluşmuştur. Bu programların, yönetimini üstlendikleri sistem kaynaklarını kullanımına oranı ne kadar düşük olursa sistemin verimliliği de o derece yüksek olur.
- **Güvenilirlik:** Sistem yönetimi, en azından üstünde kurulu olduğu donanım kadar güvenli olmalıdır. Donanım veya yazılım yönünden doğacak sorunları çözebilmeli ve bu sorunlardan doğacak zararları en aza indirmelidir.
- **Koruyuculuk:** Yönetim bir kullanıcının, ki bu program da olabilir, yapacağı hataların diğer kullanıcıları hiçbir şekilde etkilememesini sağlayabilmelidir.
- **Sezdiricilik:** Genel olarak kullanıcıların sistemden isteklerinin neler olduğunu sezmek veya kestirmek kolay olmamaktadır. Ancak, kullanıcıların isteklerinin belirli süreler içinde büyük farklılıklar göstermediği de gözlenmiştir.
- Örneğin, kullanıcı sisteme bir iş girdiğinde, o işin kabaca ne zaman sonuçlanabileceğini kestire- bilmelidir.
- **Elverişlilik:** Kullanıcıların bir sistemin kaynaklarını paylaşımlı olarak kullanmaları, onların kendi istekleriyle değil de sistemin onlardan istediği ekonomik bir zorunluluktur.

Çekirdek Sistem

- Donanım üzerindeki denetim işlevleri ve fiziksel birimlerin yönetimi çekirdek sistem tarafından gerçekleştirilmektedir.
- Çekirdek sistemin yürüttüğü işlevler arasında
 - kesme yönetimi,
 - görev yaratma ve yok etme,
 - G/Ç işlemlerinin sağlanması
- sayılabilir.
- Çekirdek işletim sisteminde önemli bir yer tutar; bilgisayar açıldığında belleğe alınır ve bilgisayar çalıştığı sürece bellekte tutulur.

- Bir işletim sisteminde çekirdeğin işlevi çok geniştir, çekirdeğin görevleri aşağıdaki gibi özetlenebilir:
- 1. İşlemciye prosesleri atamak
- 2. Kesmeleri yönetmek
- 3. Prosesler arasındaki iletişimi sağlamak.

Kesmelerin Yönetimi

- Kesme, yazılım veya donanımdan kaynaklanan bir uyarı sonucunda işlemcinin yürütmekte olduğu görevi bırakıp, uyarıyı üreten yere hizmet verecek göreve anahtarlanmasına yol açan olaydır.
- Çoğu bilgisayar sisteminde, kesmeyi işleyecek görevin anahtarlanması, donanım tarafından başlatılan, yazılım yolu ile tamamlanan bir süreçtir.
- Kesme üreten uyanlar, bilgisayar sisteminin yapısı içinde tür ve sayıca dondurulmuş bulunduklarından bunları, işleyecek görevlerin sayısı da belirlidir.

Giriş/Çıkış Donanımı Yönetimi

- İşletim sistemlerinde, çekirdeği oluşturan katmanın yürüttüğü en karmaşık görev, hiç kuşkusuz, giriş/çıkış donanımı yönetimidir.
- Giriş/çıkış birimlerinin farklı yapılarda olmaları, bu birimleri ana sisteme bağlarken izlenen yolların çeşitliliği, yönetim işlevini zorlayan başlıca unsurlardır.
- İşletim sistemindeki çekirdek yazılım katmanı, bir yandan bellek ile çevre birimleri arasındaki fiziksel veri akışını yönetirken diğer yandan da, bu birimlerin fiziksel farklılıklarının, daha üst düzeyde yer alan yazılıma yansıtmayan, tekdüze bir erişim mekanizması sağlamakla yükümlüdür.
- Bilgisayarlar çevre ile olan bağlantılarını G/Ç birimleri yoluyla sağlar. İşlenecek veriler ve kullanılacak programlar giriş birimlerinden işlemciye iletilir; işlem sonuçları bu kez çıkış birimleri aracılığıyla kullanıcıya yansıtılır. Burada iki-yönlü iletişimin sağlanmasında kullanılan cihazlar G/Ç birimleri olarak adlandırılır.

Bağlantı Arabirimi

- Çok hızlı çalışan işlemciye karşın, G/Ç birimlerinin birçoğu elektromekaniktir ve göreceli olarak oldukça yavaşırlar. Böylesi farklı hızlarda çalışan birimler arasında bağlantı arabirimleri kullanılır.

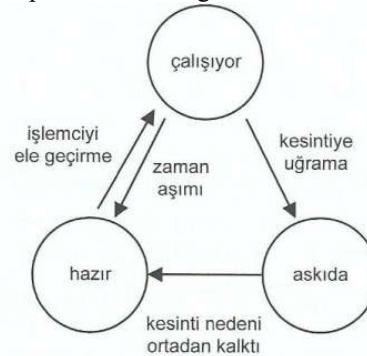
Dosya Yönetim Sistemi

- "Dosya yönetim sistemi daha üst düzeydeki yazılım katmanlarının simgesel olarak tanımlanan dosyalar ile sistemdeki fiziksel G/Ç birimleri arasındaki eşlemeyi yapan, bu mimlerin etkin ve verimli biçimde kullanılmalarını sağlayan işlevlerin toplandığı kesimdir.
- Programlama dillerinde yer alan dosya işlemleri kullanıcılara, gerek yapı gerekse adlandırma yönlerinden gerçek sistemden olduğunca bağımsız dosyalar oluşturma olanağı sağlar.
- Örneğin üst düzey bir programlama dilinde, ardışıl erişimli bir dosya, ard arda kesintisiz olarak gelen verilerden oluşan bir yapıdadır. Oysa bu dosya, bir mıknaatıslı teker üzerinde sektör, iz, silindir gibi fiziksel kalıplara bölünecek, belki de ard arda gelmeyen konumlara yazılacaktır.
- Bu yapıyı kullanıcıya, programlama dili düzeyinde algıladığı gibi yansıtmak, dosya yönetim sisteminin birincil işlevidir. Böylece bir dosya için gerekli alanın atanması, atamaların mıknaatıslı tekerin en fazla veri saklayabilecek ve G/Ç başarımı arttıracak biçimde yapılması, bu katmanın, çevre birimlerinin etkin kullanımı için üstlendiği işlevlerden birkaç tanesidir.
- Dosya kimliği {file identifier}, dosyayı bir sistem içinde ötekilerden ayıran simgedir. Örneğin, kurumlardaki sicil veya öğrenci numaralan bu nitelikte biricik (unique) tanımlardır.
- Dosya tanımlayıcı {file descriptor} ise, dosyaya ilişkin fiziksel ve mantıksal özelliklerin tutulduğu veri kümesidir. Dosyalara fiziksel olarak ulaşabilmek için, burada bulunan bilgilerden yararlanır.
- Günlük yaşamımızda adres, iş yeri, doğum tarihi vb bilgileri içeren kimlik kartları, işlev açısından dosya tanımlayıcılara benzetilebilir.

6. Prosesler ve Proses Yönetimi

- Proses, kendi veri şablonu olan ve kendi başına bütünlüğü olan kod parçasına verilen isimlendirmedir.
- Proses kavramı bilgisayar işletim sistemlerinde oldukça önemlidir; çünkü, bilgisayarın ana işlemcisine verilecek/yaptırılacak işler prosesler düzeyinde yapılır.
- İşlemci aynı anda bir proses yürütür; o prosesin işi bittiğinde veya herhangi bir nedenle o proses askıya alındığında bir başka processe geçer.
- Dolayısıyla işletim sistemlerinde yürütülmeye hazır olan proseslerin tutulduğu bir liste vardır.
- Yürütülmesi için hazır olan prosesler, hemen işlemciye ele geçirilemezler (kullanamazlar), "hazır prosesler listesine/kuyruğuna öncelik bilgisine göre eklenirler;
- İşlemci o anda yürüttüğü prosesini bıraktıktan sonra listedeki bir sonraki prosesini alır ve yürütmeye başlar.
- İşlemci yönetimindeki ana amaç, sistemdeki çalışma verimini en yüksek tutacak şekilde işlemciyi proseslere paylaştırmaktır. Genel olarak bir proses aşağıdaki üç durumdan birinde bulunabilir.
- **Çalışıyor** {Running;}: O anda işlemciyi kullanan, ele geçirmiş olan processtir.
- **Askıda** (Blocked): Herhangi bir nedenle işlemciyi bırakmış, bir ihtiyacının sağlanmasını bekleyen proses; örneğin, bir veriye ihtiyacı vardır ve henüz o veri hazır değildir.
- **Hazır** (Ready): Her şeyiyle yürütülmeye hazır olan proseslerdir; hazır olan prosesler işletilme önceliğine göre hazır prosesler kuyruğuna eklenirler.
- Bir proses çalışıyor durumunda iken askıda durumuna geçmesi için harici birtakım olayların oluşması gerekir:
- Örneğin, G/C işlemi veya o anda herhangi bir kaynak yetersizliği gibi. Böylesi durumlar işlemcinin kontrolü dışında ortaya çıkar.
- Hazır durumuna geçiş iki şekilde olur; ya çalışırken o proses ayrılan zaman dilimi dolmuştur ya da askıda iken daha önce kesintiye uğrama nedeni ortadan kalkmıştır.
- Örneğin veri girilmesi gerekiyordur ve veri girilmiştir. Bir proses varlığı süresince bu üç durum arasında gidip gelir.
- Proseslerin hangi durumda bulunduğu bilgisi işletim sistemlerince kullanıcılara gösterilir.
- Örneğin Unix işletim sisteminde her processe tamsayıdan oluşan bir kimlik numarası verilir.
- Bu kimlik numarası üzerinden proseslere çeşitli müdahaleler de bulunulabilir.
- Örneğin proses yok edilebilir. Unix işletim sisteminde ps {process status} komutu "prosesleri listeler; «kill» komutu yanında kimlik numarası verilen prosesini yok eder; zaman zaman bir proses kilitletiğinde bu komuta başvurulur.

Bir prosesin herhangi bir andaki durumu



Zaman Çizelgeleyici (Scheduler)

- Çizelgeleyici, işlemciye atanacak olan prosesini ve prosesin hangi koşullar altında işlemciyi kullanacağını belirler; iki ana görevi vardır:
- 1. Proses kuyruğundan yürütülecek prosesini seçmek
- 2. Proses ait kullanım zaman dilimini ayarlamak.

- Zaman dilimi, bir prosesin işlemciyi ele geçirdikten sonra en fazla ne kadar meşgul edeceğini gösterir; eğer zaman dilimi aşılsa proses yine hazır kuyruğuna girer ve tekrar kendisine sıra gelmesini bekler.
- Çizelgeliyicinin hangi mantığa göre hazır listesinden proses seçeceğini önemlidir; bu seçme işini, genel olarak, daha önce her bir iş için atanan öncelik değerlerine göre yapar.

Ölümcul Kilitlenme (Deadlock)

- Ölümcul kilitlenme, proseslerin hiçbir zaman ele geçiremeyecek bir birime ya da kaynağı ihtiyaç duymaları durumunda sürekli olarak askıda kalmalarıdır. Ayrıca bir proses bir diğerinin işini bitirmesini o da onun bitirmesinin bekleyen bir durum olursa yine ölümcul kilitlenme oluşur.
- Örneğin, paralel çalışan iki iş kaynakları çapraz istekte bulunursa kilitlenme oluşacaktır:

A işi B işi

x	y
y	x

Semaforlar

- Semafor, bir çeşit değişken türüdür; bu değişken üzerinde işlem yapılacak iki temel işlem tanımlanmıştır. Bu işlemlerden birisi özgün adıyla signal, diğeri wait'tir;
- bazı kaynaklarda ilk tanımlandığı zaman ki adları olan P ve V olarak ta anılırlar. Signal işlemi semaforun içeriğini bir artırır; ve signal (s) şeklinde gösterilir. Wait işlemi de semaforun içeriğini bir azaltır; ve wait (s) şeklinde gösterilir. Burada semafor değişkendir ve artı bir sayıdır.
- En çok 1 değerini alan semafora "ikili semafor" adı verilir. Semafor değişkenler, genel olarak, kaynakların kritik bölgelerde kilitlenme olmaması için kullanılır:

wait (s)
kritik kaynak signal(s)

7. Bellek Yönetimi

- Bilgisayarın ana belleğini belirli bir düzen içerisinde kullandıran işlevlerin hepsine "bellek yönetimi" denilir:
- İhtiyaç duyulan belleği vermek,
- bununla ilgili bilgileri tutmak
- kullanılmayan bellek alanını serbest bırakmak,
- görüntü/sanal bellek yönetimi mekanizması oluşturmak ve
- belleğin verimli kullanılmasını sağlamak gibi birçok işlevi vardır.

Ana Belleğin Düzenlenmesi

- Ana bellek, yönetim ve kullanım şekli açısından çeşitli yapılarda düzenlenebilmektedir:
- 1. Tekli kesintisiz (Single contiguous)
- 2. Bölümlenmiş (Partitioned)
- 3. Yer-değiştirilebilir bölümlenmiş (Relocatable partitioned)
- 4. Sayfalı (Paged)
- 5. İsteğe-bağlı sayfalı (Demand-Paged)
- 6. Dilimli (Segmented)
- 7. Dilimli ve isteğe-bağlı sayfalı (Segmented And Demand Paged).
- Bu tekniklerden ilk ikisi statik diğerleri dinamik yöntemlerdir. Ayrıca, ilk dört bellek yönetimi "gerçek", diğerleri "görüntü" bellek yönetimi olarak sınıflandırılır.
- Tekli-kesintisiz yapıda bellek lineer şekilde tek bir bütün olarak, bellek alanı olarak kullanılır.
- Bu durum daha çok mikrobilgisayarlarda böyledir. Hiçbir parçalama ve dilimleme yoktur, bellekte her gözün bir adresi vardır ve doğrudan belleğin fiziksel adresi kullanır. Bu belleğin en yalın kullanım şeklidir.
- Bölümlenmiş bellek yapısında lineer olarak düzenlenmiş bellek alanı farklı amaçlarla bölümlenir ve her bölüm bir amaca uygun olarak kullanılır.
- Sayfalı bellek yapısının temeli şöyledir: Adresleme alanı ve fiziksel alan eşit uzunlukta küçük parçalara bölünür; adresleme alanının küçük

parçalarına "sayfa", fiziksel alanın küçük parçalarına ise "blok" adı verilir.

- Böyle bir yapı içinde bir öğenin adresi, içinde bulunduğu sayfa no ve sayfa içindeki konumudur.
- Bellek yönetiminde böylesi yapıların olmasındaki amaç, belleği olabildiğince israf etmeden kullanmak ve çoklu göreve imkan verecek görüntü/sanal bellek yapısının kurulmasını sağlamak içindir.

Görüntü/Sanal Bellek

- Görüntü bellek, bellek biriminin doğrudan fiziksel adresleme yapısını kullanmak yerine dolaylı olarak kullanmaktır.
- Şöyle ki, görüntü bellek sisteminde bellek alanı dilimlenir ve sayfalanır; bir bellek gözüne erişmek için hem dilim hem sayfa bilgisi gerekir. Görüntü bellek yapısı çok-kullanıcı sistemlere imkan vermiştir.

8. Özet

- Teknik açıdan, işletim sistemi, "bilgisayar sistemini oluşturan donanım ve yazılım nitelikli kaynakları kullanıcılar arasında kolay, hızlı ve nitelikli bir işletim hizmetine olanak verecek biçimde paylaştırırken, bu kaynakların kullanım verimliliğini en üst düzeyde tutmayı amaçlayan bir yazılım sistemi" olarak tanımlanmıştır.
- Bir bilgisayar sisteminin genel olarak dört ayrı bileşeni vardır:
- Donanım
- işletim sistemi
- uygulama programları ve
- kullanıcılarıdır.
- işletim sistemi, kullanıcı açısından bir arayüzden oluşur; bu kullanıcılara ilgili programları çalıştırma, etkileşimde bulunma ve sonuçları görme imkanı verir. Bu arayüz aracılığıyla, kullanıcı bilgisayarı kolayca kullanır.
- Genel amaçlı ve özel amaçlı olarak tasarlanmış birçok işletim sistemi vardır; örneğin, Windows, Linux, Unix, Macintosh hemen akla gelen işletim sistemleridir.
- İşletim sistemlerinde en önemli unsurlar proses yönetimi, bellek yönetimi, giriş/çıkış işlemleri, dosya yönetimi olarak sıralanabilir.
- Çekirdek işletim sisteminin en önemli parçasıdır ve donanım ile uygulama programları arasındaki işlerin yapılmasını sağlar.

6. BÖLÜM

Bilgisayar ağları ve İnternet

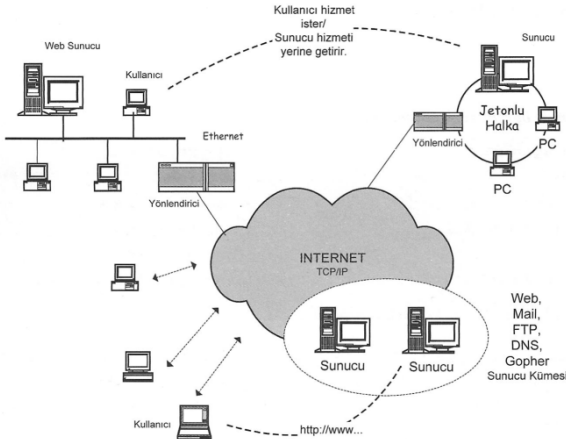
Komple ağı parçaları	OSI başvuru modeli
LAN, MAN	Uzak bağlantı
Kampüs Ağ	İnternet'in tarihçesi
İnternet Hizmetleri	İnternet tekniği
TCP/IP	İnternet alan adı
Kullanıcı/sunucu mimarisi	Kablosuz ağlar
Ethernet	Ağ cihazları

- Bilgisayar ağı bilgisayar ve benzeri sayısal sistemlerin belirli bir protokol altında birbirleriyle iletişimde bulunmasını sağlayan sistemdir.
- Ağ üzerindeki bilgisayarlar, birbirlerinden çok uzakta olsalar bile aynı protokol sayesinde karşılıklı veri alış-verişinde bulunabilir.
- Bilgisayar ağı, genel olarak geniş bir anlama sahiptir.
- Küçük bir ofis içerisindeki birkaç bilgisayar ve bir yazıcının bir HUB cihazı üzerinden bağlanıp belirli bir protokol altında haberleşmeleri de bir ağ oluşturur, tüm dünyaya yayılmış İnternet de bir bilgisayar ağıdır.
- Ağın büyüklüğü ne olursa olsun, ağ kapsamında, iletişimde bulunacak uç sistemler, ağı oluşturan HUB, anahtar, yönlendi-

rici gibi ağ cihazları ve kablolu alt yapısı / veya kablosuz iletim ortamı vardır.

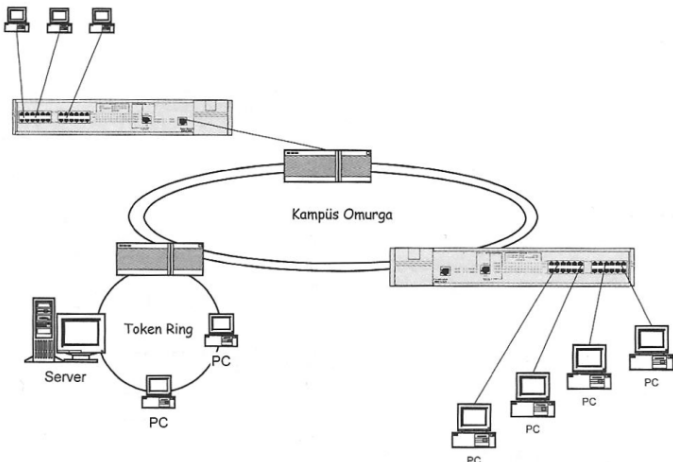
1. Komple Ağın Parametreleri

- Komple bir ağ genel olarak, LAN, Kampüs Ağı, WAN ve Uzak Bağlantı olarak adlandırılan dört farklı parçadan oluşur.
 - Herbir parçanın kendisine yönelik özellikleri ve standartları vardır. Dolayısıyla bir parçaların her biri ayrı ayrı göz önüne alınmalıdır.
 - Bilgisayar ağları konusuna başlarken herşeyden bu dört parçanın özellikleri ve ağ konusunda referans nokta olan OSI başvuru modeli kavramları öğrenilmelidir. Komple bir bilgisayar ağının topolojisi tipik olarak Şekil.1 deki gibi gösterilebilir ve dört farklı parçadan oluşur.
 - LAN, Kampüs Ağı, WAN, Uzak Bağlantı
- Şekil 1. Komple bir ağın topolojisi



1. Parça: Yerel Alan Ağı - LAN (Local Area Network)

- LAN'larda temel özellik, sistemlerin aynı ortamda veya birbirlerine yakın mesafede olmalarıdır. Bu nedenle sistemlerin arasında kullanılacak kabloların seçiminde büyük bir esneklik vardır ve kablolu alt yapısı bir kez kurulduktan sonra maliyetsiz yüksek hızlı bir iletim ortamı sağlar; veya kablosuz iletim ortamı kolayca kurulabilir.
 - En basitinden 1 HUB ile LAN kurulabilir. Şekil.2 de 1 HUB ve anahtar cihazından oluşan bir yerel alan. ağı görülmektedir.
 - LAN kurulmasında ofis uygulamalarında "Ethernet" teknolojisi, endüstriyel uygulamalarda ise "Jeton Halka" (Token Ring) teknolojisi tercih edilmektedir.
 - Yukarıdaki şekilde sol-üste Ethernet ve sağ-üste de Jetonlu Halka topolojisi görülmektedir. Görüldüğü gibi farklı teknolojilerden oluşan alt ağlar birbirlerine bağlanabilmektedir.
- Şekil 2. Kampüs uygulaması



2. Parça: Kampüs Ağı

- Kampüs ağ, LAN ile benzer özelliklere sahiptir; farkı, daha uzak mesafe gereksinimi ve birden çok LAN içerebilmesidir. Adı üzerinde üniversite kampüsleri gibi sınırlı bir alana dağılmış binalar içerisindeki bilgi-

sayarları, alt-ağ kapsamında yerel alan ağlarını birbirine bağlamak için kullanılır.

- Omurga ağ, kampüs ağ oluşturmak için kurulan ve tüm kampüsü dolaşan çatı durumundadır. Daha uzak mesafelere bağlantılar yapılacağı için bakır kabloların yetmediği yerlerde fiber optik kablolar veya noktadan noktaya kablosuz bağlantılar yapılabilir.

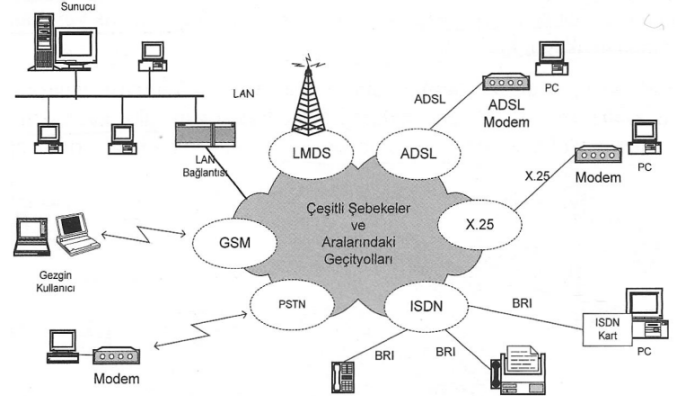
3. Parça: Geniş Alan Ağı (WAN: Wide Area Network)

- WAN bağlantısı, birbirinden çok uzakta olan kampüs türü ağları ve yerel alan ağlarını, birbirine bağlar. WAN uygulamasında anahtar sözcükler mesafe, band genişliği ve aradaki iletişim ortamının telekom şirketinden kiralanmasıdır. WAN uygulamasında iletişim ortamı band genişliği sınırlıdır ve genişliğine göre ücret ödenir; dolayısıyla en verimli şekilde kullanılması gerekir.

4. Parça: Uzak Bağlantı (Remote Connection)

- Uzak bağlantı bir veya birkaç bilgisayarın veya küçük bir ofiste bulunan bilgisayar grubunun merkezi yere bağlanması için kullanılır. Örneğin günümüzde yaygın olarak kullanılan ADSL bağlantıları uzak bağlantı olarak düşünülebilir.
- WAN'ın bir parçası olduğu varsayılabilir; ancak temel özellik birkaç kullanıcı olması ve iletim ortamı olarak büyük bir band genişliğine ihtiyaç olmamasıdır.
- Ayrıca uzak bağlantı bir çeşit tüketici bağlantısıdır. Yani daha çok veri alış yapısı. Veri gönderme oranı daha azdır.
- Uzak bağlantılarda, genel olarak, iletişim ortamı olarak Türk Telekom veya GSM operatörlerinin sunduğu telefon iletişimi kullanılır; örneğin ADSL ve GSM altyapısı uzak bağlantı için kullanılabilir; bkz. Şekil.3 .

Şekil 3. Uzak bağlantı uygulaması



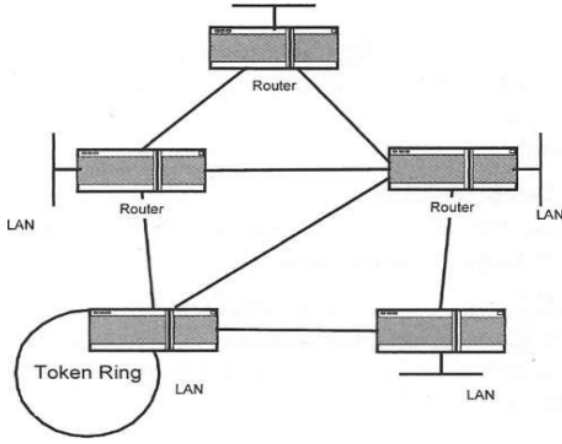
2. Band Genişliği ve İletişim Hızı

- Band genişliği, bir iletim ortamının birim zaman da aktardığı veri miktarıdır; birim zaman da, genel olarak 1 saniyedir. Dolayısıyla band genişliği saniyede aktarılan veri miktarıdır; ve bps (bit per second) ile gösterilir.
- Örneğin 1 saniyede 100 bit veri aktarılıyorsa band genişliği 100 bps'dır.
- İletişim hızı da, band genişliği ile yakından ilişkilidir; kanalın band genişliğine kadar büyüksse iletişim hızı da o kadar yüksek olur.
- Yani karşı tarafa aktarılacak istenen veri daha çabuk gönderilir. Günümüzde, band genişliği yerel alan ağlarında 100 Mbps ile 10 Gbps arasında, geniş alan ağlarında ise 0.5 Mbps ile 1 Gbps arasında değişmektedir. 100 Mbps, 12.5 MBps eder; buradaki büyük B harfi Byte anlamına gelir

3. OSI Başvuru Modeli

- OSI başvuru modeli ISO tarafından tanımlanmış ve ağ uygulamasında kullanılan örnek bir modeldir; ve ağdaki bir bilgis-

bağlantılarında kullanılan yönlendiriciler "IP yönlendirici" olarak adlandırılır.



• Geçityolu Gateway

- Geçityolları, adı üzerinde kapalı bir alandan dışarıya açılma olanağı sunan cihazlardır. Ağ uygulamasında iki farklı şekilde kullanılır. Birisi, yerel alan ağların İnternet'e veya başka ağlara bağlanmasında kullanılır; bu şekilde kullanım yönlendiriciler aracılığıyla yapılır. Bu durumda yönlendirici aynı zamanda geçityolu olmuş olur.
- İkinci kullanım şekli farklı protokol kümesine dayalı ağların birbirine bağlanmasını sağlamaktır. Çünkü protokol kümeleri arasında ciddi bir dönüşüm yapılmalıdır; protokol dönüşümü, OSI başvuru modeline göre 6-7 katmana gereksinim duyar.

• Güvenlik Duvarı FireWall

- Güvenlik duvarı özel ağ ile İnternet gibi genel ağ arasına konulan ve istenmeyen erişimleri engelleyen bir sistemdir; bununla ağ güvenliği sağlanmaya çalışılır ve erişim hakları düzenlenir.
- Güvenlik duvarının sistem üzerinde tam olarak etkili olabilmesi için, ağ ortamı ile İnternet arasındaki tüm trafiğin güvenlik duvarı üzerinden geçirilmesi gerekir.
- Güvenlik duvarlarının tercih edilmesi için en büyük nedenlerden biri de adres dönüşüm (NAT, Network Address Translation) özelliğidir. Sadece tek bir IP adresi ile tüm ağ kullanıcıları İnternet'e çıkabilir ve yerel ağ ortamındaki IP adresleri tamamen İnternet ortamından yalıtılmış şekilde kullanılabilir.
- Güvenlik duvarı görevi gören sistemlere, virüs tehlikesini algılayacak ve bunları engelleyecek yazılımlar da yüklenebilir. Böylece e-mail veya başka şekillerde ağa gelebilecek virüs tehlikesi için koruma sağlanmış olur.
- Güvenlik duvarının konfigürasyonu belirli bir stratejiye göre hazırlanır; kurulmadan önce ne tür bilgilerin korunacağı, ne derecede bir güvenlik uygulanacağı ve kullanılacak güvenlik algoritmaları önceden belirlenir. Genel yaklaşım engelleme ve serbest bırakma şeklindedir.

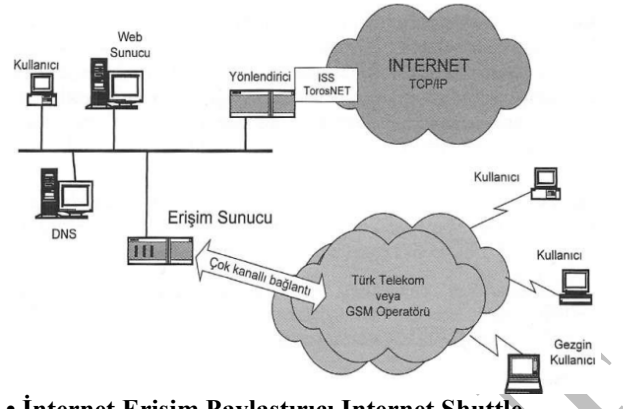
• Modem Cihazı

- Modem cihazı, analog hatlar üzerinden sayısal veri aktarımı yapılmasını sağlayan bir ağ cihazıdır. Bir PC'nin telefon hattı üzerinden merkezi bir yere bağlanması için de kullanılır.
- Bir LAN'ın analog bir hat üzerinden WAN bağlantısının yapılmasında da kullanılır. Komple bir LAN'ın WAN bağlantısında analog hat kullanılıyorsa, hat ile yönlendirici arasına bir modem yerleştirilmelidir.

• Erişim Sunucu Access Server

- Erişim sunucu, LAN 'lara uzak bağlantıların gerçekleştirilmesi için kullanılır; iki nokta arasında bir iletişim kanalı oluşturur. Çevrim-içi bağlantı ile yapılan İnternet bağlantıları erişim sağlayıcılar üzerinden olmaktadır; bkz. aşağıdaki Şekil.7 ye. Aşağıdaki şekilde İnternet bulutu, bir yerel alan ağı, Türk Telekom şebeke bulut ve erişim sunucunun yeri gösterilmiştir.

Şekil 7. Erişim sunucu uygulaması örneği



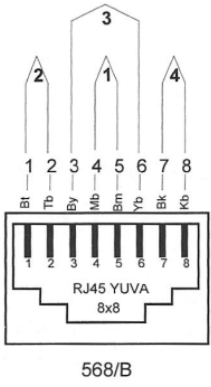
• İnternet Erişim Paylaştırıcı Internet Shuttle

- İnternet erişim paylaştırıcılar yerel ağa bağlı bilgisayarların tek bir İnternet bağlantısıyla İnternet'e erişmelerini sağlarlar; küçük ofisler veya birden çok bilgisayarın olduğu ev kullanıcıları için uygun bir çözüm sunarlar; bir bağlantıyla birden çok kullanıcı İnternet'e çıkabilir.
- Kısaca İnternet erişim paylaştırıcıyla, telefon hattı üzerinden modem bağlantısıyla birden çok kullanıcı İnternet'e çıkabilir; tüm bilgisayarlarda tek bir IP adresi kullanılarak erişim yapılabilir. Günümüzde ADSL modemler de bir erişim paylaştırıcı gibi çalışmaktadır.
- Aslında, çok daha fazla ağ cihazı vardır; ancak verilenler en çok kullanılanlar olup bilgisayar ağlarının temelini oluştururlar.

5. Kablolama ve Kablosuz Bağlantı

- Ağ uygulamalarında sistemler arasındaki elektriksel bağlantı iletişimin temel taşıdır; bu, kablolama yapılarak veya gerektiğinde kablosuz ortamlar kullanılarak sağlanır. Kablolama ihtiyacı, sistemler arasındaki mesafeye ve istenen hıza bağlı olarak bakır veya fiber optik kablolarla sağlanır.
- **Kablolama Alt Yapısı**
- Kablolama altyapısında en önemli anahtar sözcük yapısal kablolamadır; belirli bir standarda göre yapılır.
- Bu standartlardan en ünlüleri EIA-568 ailesi ve ISO 11801'dir. Bu standartlarla kablolar çeşitli sınıflara ayrılmış ve uygulama alanına göre tanımlar yapılmıştır.
- Yapısal kablolamada daha çok bakır UTP ve fiber optik (FO) kablolar kullanılır. UTP kablolar, daha çok ağ cihazları ile bilgisayarlar arası bağlantılarda kullanılırken, FO kablolar ağ cihazlarının birbirine bağlantısında, mesafe sorunu olduğu durumlarda ve elektromanyetik gürültünün çok olduğu yerlerde kullanılır.
- UTP ağ kablosu, sekiz telden oluşur ve teller birbirine ikiyeşerli gruplar haline dolandırılmıştır; sonlandırılması için RJ45 konnektörü kullanılır.
- Kablolama alt yapısı kurulurken, kullanılan kablo, sonlandırma konnektörleri, ara bağlantı birimleri vs hepsi aynı sınıftan olmalı veya bir üst sınıftan olmalıdır. Örneğin UTP Cat 5e kablo kullanılıyorsa, priz, RJ45 fiş hepsi Cat 5e standardında olmalıdır.
- Aşağıda UTP kablonun RJ45 konnektörü ile düz sonlandırılması için gerekli EIA-568/B standardı verilmiştir. RJ45 konnektörü 8-uçludur ve bu uçların hepsine birer UTP kablo teli bağlanır. Hangi rengin hangi uca bağlanacağı aşağıda görülmektedir.

Şekil 8. Kablo sonlandırma; RJ45 konnektörü

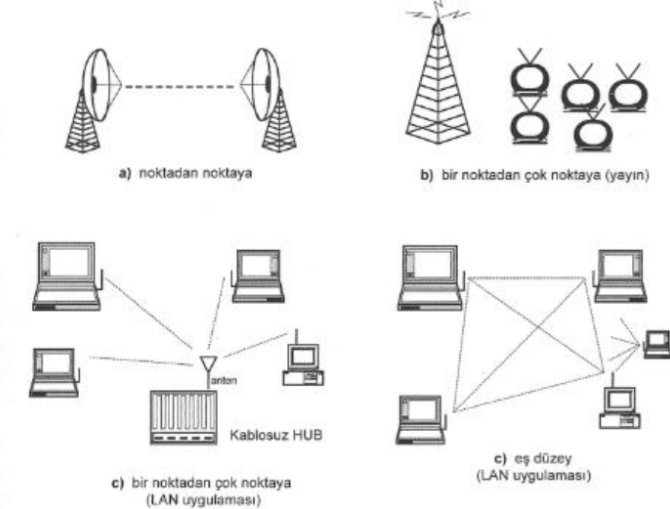


Uç	EIA-568/B Renkleri
1	Beyaz/Turuncu - Bt
2	Turuncu/Beyaz - Tb
3	Beyaz/Yeşil - By
4	Mavi/Beyaz - Mb
5	Beyaz/Mavi - Bm
6	Yeşil/Beyaz - Yb
7	Beyaz/Kahve - Bk
8	Kahve/Beyaz - Kb

• Kablosuz Bağlantı Wireless Connection

- Kablo döşemesinin zor/imkansız olduğu durumlarda veya kablo döşenmesinin istenmediği durumlarda kablosuz iletim yapılabilir; yani bir-biriyle iletişimde bulunacak sistemler, havadan kablosuz olarak RF işaretler veya kızılötesi ışınları kullanarak iletişim yapar.
- Kablosuz aktarım uygulama gereksinimine göre çeşitli şekiller yapılabilmektedir:
- Noktadan noktaya (point-to-point)
- Bir noktadan çok noktaya (point-to-multipoint)
- Eş düzey (peering)
- Noktadan noktaya kablosuz iletim daha çok ana hatların bağlantısı için kullanılırken bir noktadan çok noktaya kablosuz dağıtım için kullanılır.
- Ethernet tabanlı kablosuz LAN'da bir noktadan çok noktaya kablosuz iletimle yapılmaktadır. Eş düzey kablosuz bağlantıda her bilgisayar aynı özelliiktir.
- Aşağıdaki şekilde her üç kablosuz bağlantı şekli gösterilmektedir. Kablosuz LAN uygulamasında yoğun olarak IEEE 802.11 ailesi kullanılmaktadır.

Şekil 9. Kablosuz bağlantı yöntemleri



- **Bluetooth**, kablosuz iletişim ortamı sunan bir teknolojidir; kısa mesafede klavye, fare, telefon vs gibi sayısal giriş/çıkış birimlerinin ara kablolar olmadan ana sisteme bağlantı yapılmasını kolaylaştıran bir teknolojidir.
- Hücresel ağ bağlantısının çeşitli cihazlara paylaştırılmasını sağlayacak bir teknolojidir denilebilir. Kişisel alan ağlarına (PAN: Personel Area Network) veya ailesel alan ağlarına (FAN: Family Area Network) geçiş de hızlandıran bir teknolojidir.

6. Ağ Üzerine Çeşitli Kavramlar

- Bilgisayar ağı hem çok basit hem de karmaşık bir konudur ve bilgisayar mühendisliğinde önemli yere sahiptir.
- Eğer küçük bir ofiste yerel alan ağı kurulacaksa basit bir konudur; bir HUB ve birkaç bilgisayar ile kolayca kurulur. Ancak, unutulmamalıdır ki, İnternet'in alt yapısı, Türk Telekom'un alt yapısı ve GSM operatörlerin alt yapısı da birer bilgisayar ağıdır.

Protokol ve Birlikte Çalışabilme

- Ağ üzerinde iki bilgisayarın karşılıklı veri aktarabilmesi ve ortak süreçler yürütebilmesi için bilgisayarların birlikte çalışabilme (interoperability) yeteneğinin olması gerekir.
- Birlikte çalışabilme, verici ve alıcı arasında kullanılacak işaretler, veri formatları ve verinin değerlendirme yöntemleri üzerinde anlaşmayla mümkün olur.
- Bunu sağlayan kurallar dizisi "protokol" olarak adlandırılır. Protokol, ağı farklı parçalarının birbirleriyle nasıl etkileşimde ve iletişimde bulunacağını belirler.

Standart

- Bir standart, her üreticinin uyduğu ortak tanımlamalardır. Tasarım ve üretim aşamasında standarda uygun yapılsa, o ürün marka bağımsız olarak kullanılabilir.
- Örneğin RS-232-C seri bağlantının 9 veya 25 uçlu olması bir standarttır;

Protokol Kümeleri

- Halihazırda birçok protokol kümesi geliştirilmiştir. Bunlardan bazıları yalnızca onu geliştiren üreticiler tarafından kullanılırken birçoğu açık sistem haline gelmiştir.
- Örneğin DECnet, SPX/IPX, SNA ve XNS protokol kümeleri sırasıyla Digital, Novell, IBM ve Xerox firmaları tarafından geliştirilmişlerdir ve yine bu firmalar tarafından kullanılmaktadır.
- TCP/IP gibi bazı protokol kümeleri ise bütün üreticiler tarafından desteklenen, tartışılmaz genel standart olmuşlardır. TCP/IP, İnternet uygulamasından dolayı gündemde olan protokol kümesidir.

Hizmet Kalitesi (QoS: Quality of Service)

- Kısaca QoS diye anılan hizmet kalitesi, ağı bir araç olarak kullanan uç sistemlerin gereksinim duyduğu trafik türünün ağ tarafından karşılanmasıdır.
- Örneğin iki bilgisayar arasından dosya aktarımı, iki PBX santaralı arasındaki ses bilgisi aktarımı veya video konferans uygulamaları ayrı ayrı trafik kalitesine gereksinim duyarlar.
- Video konferans gibi gerçek zaman uygulamaları, ağdan, gecikme ve iletim yolu band genişliği açısından belirli bir garanti beklerken, dosya aktarımı böyle bir garantiye gereksinim duymaz.
- Kısacası hizmet kalitesi öngörülen gereksinimlerin zamanında ve başarımla karşılanmasıdır. Uygulama için gerekli hizmet kalitesi alınmaz ise, uygulamanın çalışmasında kesintiler oluşur.

Topoloji ve Ağ Haritası

- Topoloji, bilgisayar ağını oluşturan bileşenlerin ara bağlantılarını, sistemin işlevini ve coğrafi konum açısından şeklini belirler.
- Yerel alan ağlarında yoğun olarak ortak yol, yıldız veya halka topolojileri kullanılırken, geniş alan ağlarında ağaç ve örgü topolojisi kullanılır.
- Ağ haritası da topolojiye çok yakındır; ağ üzerindeki sistemleri ve onların birbirlerine bağlantısını gösterir.
- Ortak yol topolojisinde ağdaki tüm sistemler aynı iletim kanalı üzerinden haberleşirler. Her sistemin bir adresi vardır; sistemler yol üzerindeki her mesajı okurlar ve yalnızca kendilerini adresleyen işlerler.
- Ortak yola yeni düğüm ekleme kolay olur; ancak her yeni eklenen, düğüm başına düşen yol kapasitesini azaltır.
- Halka topolojisinde her düğüm komşu iki düğüme bağlıdır; iletim yolu halka biçimindedir.
- Yolu ele geçirilmesi için en yaygın kullanılan yöntem jetonu halka üzerinde dolaştırmaktır. Yolu kimin kullanacağını, yolda dolaşan bir jeton belirler; jetonu ele geçiren düğüm, onu yoldan alıp yola verisini koyar.

• Göndereceği verisi bittikten sonra jetonu yeniden yola koyar. Aynı düğüm tekrar veri göndermek isterse jetonu ele geçirene kadar beklemelidir.

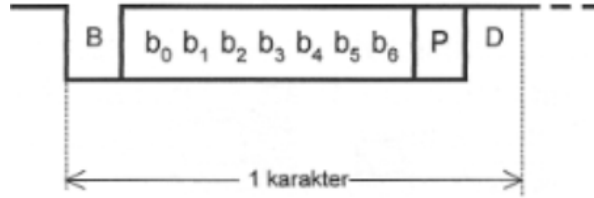
- Yıldız topolojide düğümlerin tamamı merkezi noktadaki bir cihaza bağlıdır.
- Ağ üzerindeki tüm trafik, bu merkezi noktadan geçer. Günümüzdeki ağ uygulamalarında yoğun olarak yıldız topoloji kullanılmaktadır.
- Ortak yol ve halka uygulamaları bile, mantıksal olarak ortak yol ve halka biçimde olsa dahi fiziksel bağlantıları yıldız şeklinde olabilmektedir. Yıldız topolojide de ağa yeni düğüm kolayca eklenebilir.

Çevirmeli Bağlantı Dial-Up

- Çevirmeli bağlantıda, iletişimde bulunacak iki düğüm arasında, iletişimden önce dinamik olarak bir iletişim yolu kurulur.
- İletişimi başlatmak isteyen karşı sisteme çağrı gönderir; karşı taraf çağrıyı kabul edince her iki taraf iletişimin nasıl yapılacağı üzerine el sıkışırlar; veri iletişimi el sıkışmanın ardından başlar.

Seri ve Paralel İletim

- Seri iletim aktarılabilecek verinin bit bit aktarılmasıdır. Bilgisayar ağları üzerinden veri alışı verışı genel olarak seri iletime dayanır. Seri iletim kendi içerisinde asenkron, senkron ve isokron olarak üçe ayrılır.
- Asenkron iletimde gönderilecek verinin önce başla (start) biti; sonuna da hata sezmede kullanılan eşlik (parity) ve dur (stop) bitleri eklenir.
- Bilgisayarların seri portları asenkron seri iletimdir.

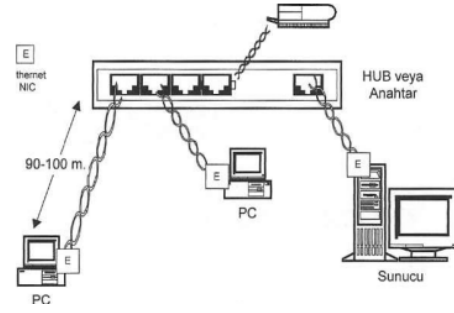


7. Ağ Teknolojileri

- Ağ Teknolojisi ağı dört farklı parçasına göre farklılık gösterir; yerel alan, geniş alan, kampüs ağların ve uzak bağlantıların herbirinin farklı teknolojileri vardır. Çünkü herbirinin gereksinimleri farklıdır, ona göre çözüm yapılmalıdır.
- Yerel alan ağı kurulabilecek teknolojilerin başında Ethernet gelir; hem daha az maliyetli hem de kurulumu kolaydır; ancak çeşitli gereksinimlerden dolayı Jetonlu Halka (Token Ring), FDDI ve ATM teknolojileri de seçilmektedir.
- Jetonlu Halka, daha çok endüstriyel uygulamalar için uygun olurken, FDDI ve ATM orta veya büyük ölçekli yerel alan ağı uygulamalarında "omurga" oluşturmak için güçlü bir yapıya sahiptir.
- Geniş alan ağı komple bir ağı önemli bir parçasını oluşturur; teknolojileri denilince, akla hemen çevirmeli modem, kiralık hat, X.25, FR, ISDN, xDSL, LMDS gelir; ancak ATM ve B-ISDN gibi teknolojiler de uygulamalarında görülmektedir.
- Her teknolojinin kendisine has, uygulamada seçimi olabilecek özellikleri vardır.

Ethernet ve Türevleri

- Ethernet ilk olarak 2,94 Mbps hızında tasarlandı; ancak daha sonra, artan gereksinimi karşılamak üzere 10 Mbps, 100 Mbps, 1000 Mbps ve Gbps hızlarında çalışan türevleri geliştirilmiştir. Ethernet ve türevleri olan Fast Ethernet, Gigabit Ethernet yerel alan ağları kurulmasında "vazgeçilemez bir standarttır" denilebilir.
- Ethernet (2.94 Mbps, 5 Mbps, 10 Mbps)
- Fast Ethernet (100 Mbps)
- Gigabit Ethernet (1 Gbps)
- 10G Ethernet (10 Gbps)
- Her Ethernet kartının MAC adresi olarak adlandırılan 48 bitlik bir adresi vardır ve tipik olarak 00-23-c3-45-00-b3 şeklinde gösterilir; bu adres tektir. LAN içerisindeki yerel erişimler, gerçekte bu adresler kullanılarak gerçekleştirilir.



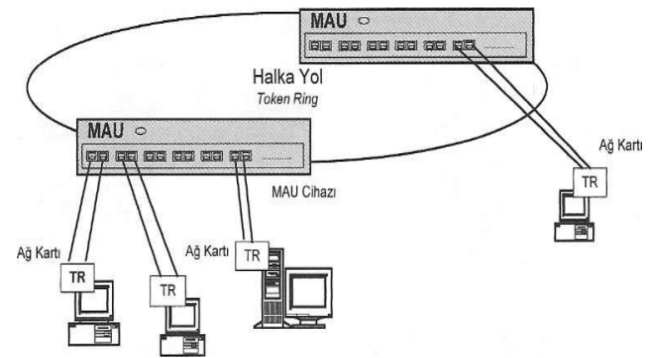
Yola Erişim - CSMA/CD

- CSMA/CD, Ethernet'te yola erişim yöntemidir; bu yöntemde bir düğüm veri aktarmadan önce yolu dinler, eğer yol, o anda diğer düğümler tarafından veri aktarmak için kullanılıyorsa, yolda bir taşıyıcı olduğunu sezer ve kendi verisini yola çıkarmaz, bir süre bekler.
- Eğer bir düğüm o anda yolda taşıyıcı olduğunu sezdiği halde, yola verisini çıkarırsa çatışma oluşur; veri aktarımı gerçekleşmez.
- Çatışmanın çok olduğu durumlarda ağ başarımını düşürür; azaltılması gerekir.

Jetonlu Halka ve Türevleri

- Jetonlu Halkada düğümler birbirlerine Şekil.12 de görüldüğü gibi halka biçiminde bağlanırlar. Aktarım hızı olarak çeşitli türleri vardır: 4, 16, 100 Mbps gibi. Jetonlu Halka teknolojisi ofis uygulamalarında pek kullanılmaz.
- Düğümlerin birbirlerine halka biçimde bağlanmasından dolayı, her düğüm fiziksel olarak komşu 2 düğüme bağlıdır.
- Jetonlu Halka ağının kurulması MAU (Multistation Access Unit) olarak adlandırılan ve üzerinde uç sistemlerin bağlanması için birden çok port olan cihazlar kullanılır.
- Jetonlu Halkada, biri veri aktarımı, diğeri jeton aktarımı için iki tür çerçeve kullanılır. Veri çerçevesi, bir düğümden diğerine bilgi aktarılan çerçevedir; Jeton çerçevesi ise, halkaya veri çerçevesi çıkarmak isteyen düğüme o hakkı vermeyi sağlayan özel bir kısa çerçevedir.
- Çerçeve başlığı içindeki bir bit, çerçevenin hangisi olduğunu ayırt etmeye yarar.

Şekil 12. Jetonlu Halka Uygulaması



Bazı WAN ve Uzak Bağlantı Teknolojileri

- Çok çeşitli uzak bağlantı ve WAN bağlantı teknolojileri vardır; ilk akla gelen Frame Relay, X.25, ISDN ve ISDN sıralanabilir. Herbirinin farklı hizmet türleri ve kalitesi vardır. Uygulama gereksinimine en uygun olanı seçilir.
- **Frame Relay & X.25**
- Frame Relay, paket anahtarlama bir WAN teknolojisidir; X.25, bulut teknolojisine dayanan ve paket anahtarlama bir ağ (Packet Switching Network, PSN) üzerinden senkron veri aktarımı yapılmasını sağlayan bir arayüz tanımlamasıdır. Hizmet kalitesinin (QoS) fazla önemli olmadığı uygulamalarda en ekonomik aktarım ortamını sunar.
- **ISDN Integrated Services Digital Network**

- ISDN (Tümleşik Hizmetler Sayısal Şebekesi), aynı anda ses, veri, resim ve görüntü aktarma olanağı veren; bulut teknolojisine dayanan ve sayısal iletişim ortamı sunan bir ağ sistemidir.
- ISDN, telefon şebekesinde olduğu gibi çevrimli sayısal bağlantı kurma olanağı sunar; numarası bilinen bir aboneye, iletişime önce çağrı yapılarak bağlantı kurulur ve iletişim gerçekleştirildikten sonra bağlantı koparılır.
- ISDN, BRI ve PRI olarak adlandırılan iki tür hizmet sunar:
 - **BRI hizmeti**
 - Kullanıcı verisi için 128 Kbps'lik band genişliğine sahip olup, 2B+D olarak da gösterilir; herbiri 64 Kbps olan 2 tane B kanalı, 16 Kbps olan bir tane D kanalı içerir.
 - **PRI hizmeti**
 - Kullanıcı verisi için 2 Mbps'lik band genişliğine sahip olup, herbiri 64 Kbps olan 30 tane B kanalı ve D kanalı (30B+D) içerir.
 - **ADSL Asymmetric Digital Subscriber Line**
 - ADSL (Asimetrik Sayısal Abone Hattı), kullanıcılara bakır telefon hattı üzerinden konuşmanın yansıya yüksek hızlarda veri haberleşmesi ortamı sağlayan bir teknolojidir.
 - Ortamın alış yönündeki hızı 1,5 Mbps'ten başlayıp 8 Mbps'e, gönderiş yönünde ise 16 Kbps'ten başlayıp 576 Kbps'e kadar olabilmektedir.
 - Bir ADSL bağlantı üzerinde 3 temel iletim kanalı vardır: alış, gönderiş ve POTS kanalı olarak adlandırılır. Alış kanalı yüksek band genişliği sunar; kapasitesi Mbps'ler mertebesinde.
 - Gönderiş kanalı Kbps'ler düzeyindedir. POTS (Plain Old Telephone Service) kanalı, ADSL iletim ortamı üzerinden telefon konuşmasını aktarmak için kullanılır.

8. İnternet ve TCP/IP

- Web sayfaları, e-posta, dosya aktarımı, müzik parçalarını dinlenmede İnternet hizmetlerinden birkaç tanesidir.
- İnternet'te bu tür uygulamalarının herbirinin kendisine özgü protokolü vardır ve bu protokol uyarınca iletişim yapılır. Bu bölümde İnternet'e teknik açıdan bir bakış yapılmakta ve temel kavramlar sunulmaktadır.

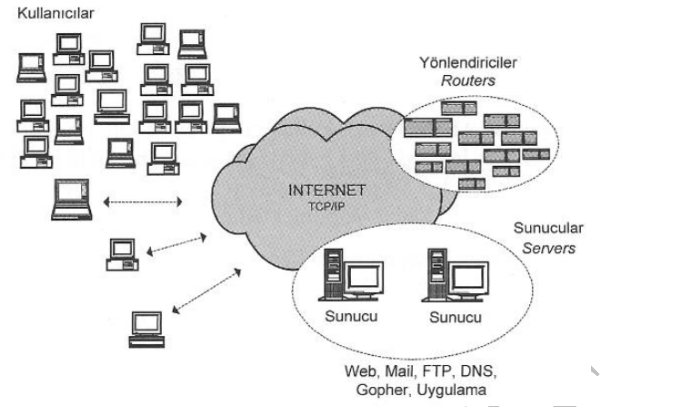
TCP/IP Protokol Kümesi

- TCP/IP bir protokol kümesidir; marka bağımsız bilgisayar sistemlerinin birbirleriyle karşılıklı çalışabilmesi için en yaygın kullanılan protokol kümesidir.
- İnternet'te de TCP/IP protokol kümesi kullanılır; bu nedenledir ki, TCP/IP kullanımı çok yaygınlaşmıştır. İnternet bir "geniş alan ağı"dır; ağların ağı da denilebilir.
- Bünyesinde binlerce yerel alan ağı, milyonlarca bilgisayar sistemi vardır ve bu bilgisayarların donanımları, işletim sistemleri birbirinden farklı olabilmektedir.
- Teknik açıdan bakıldığında, TCP/IP ve İnternet hemen hemen aynı anlama gelmeye başlamıştır.
- İnternet'in, teknik açıdan görüntüsü çizilmek istenirse, aşağıdaki gibi şekil ortaya çıkar; milyonlarca kullanıcı bilgisayarı, binlerce yönlendirici cihazı, binlerce sunucu sistem...
- İnternet ağ mimarisi katmanlı yapıdadır; ancak OSI de olduğu gibi 7 katman değil de yalnızca 4 katman tanımlıdır. İletişim için gerekli bütün iş dört katmana ayrıştırılmıştır. Her katmanda yapılacak görevler protokol tanımlamaları yapılarak belirlenmiştir.

Dört-Katmanlı İnternet Mimarisi

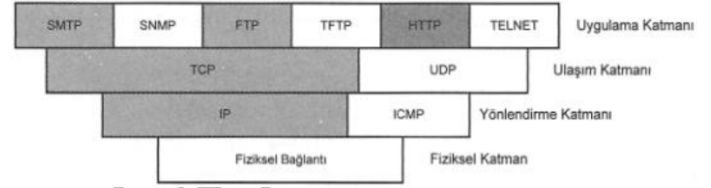
- TCP/IP, bu protokollerin oluşturduğu kümeye verilen genel isimlendirmedir; TCP ve IP isimleri yalnız başlarına küme içinde birer protokol adıyla, ikisi birlikte protokol kümesine verilen adlandırma değildir.
- TCP/IP'de IP protokolünün biri IPv4, diğeri IPv6 olarak adlandırılan iki uyarlaması vardır. İlk göze çarpan IPv6'nın adres uzayının genişliğidir.

Şekil 13. İnternetin tipik bir görünümü



Dört-Katmanlı İnternet Mimarisi

- TCP/IP, Şekil.14'de görüldüğü gibi dört-katmanlı yapıdadır; OSI başvuru modeli 7- katmanlı idi. En üstte uygulama programları vardır; bunlar genel olarak kullanıcının doğrudan etkileşimde bulunduğu programlardır.



- Uygulama katmanı altında da sırasıyla ulaşım, yönlendirme ve fiziksel katmanlar vardır. Ulaşım katmanında TCP ve UDP protokolleri, yönlendirme katmanında IP ve ICMP protokolleri tanımlıdır.
- Fiziksel katman içinse, herhangi bir tanımlama yapılmayıp var olan diğer fiziksel katman tanımlamaları geçerlidir.
- Görüldüğü gibi her katmanda birden çok protokol vardır; ancak uygulama programları tarafından istenen bir iş yerine getirilirken, her katmandaki protokollerden yalnızca biri kullanılır.
- Uygulama programları, uygulama katmanındaki protokoller aracılığıyla iletişimde bulunabilirler. Örneğin, e-posta SMTP ile tanımlanan kurallara uyarınca aktarılır; Web sayfaları da HTTP ile aktarılır.

Soket ve Port Numarası

- Uygulama ile ulaşım katman protokolleri arasında port olarak adlandırılan bir geçit tanımlıdır; bu iki katman arasındaki iletişim bu portlar aracılığıyla gerçekleştirilir; her portun 16 bitlik bir numarası vardır. Bu 16 bitlik port no ve 32 bitlik IP adresi beraberce kullanıldığında ortaya çıkan adrese soket numarası denir. TCP bağlantıları soketler üzerinden sağlanır.
- 0 ile 255 arasındaki port numaraları, standart uygulama katmanı hizmetlerine erişim için ayrılmıştır. Örneğin, FTP için port 21, TELNET için port 23 gibi birçok port numarası belirli uygulamalar için saklı tutulmuştur.

İnternet Adresleri ve DNS

- TCP/IP protokol kümesinin kullanıldığı ağlarda, adresleme IP adreslere dayanarak gerçekleştirilir. Ağ üzerinde olup da iletişimde bulunacak her sistemin birer IP adresi olmalıdır.
- IP adresler sistemlere ağ üzerinde birer adresleme kimliği kazandırır.
- IP adresleme, TCP/IP protokol kümesinin 3'üncü katmanı olan yönlendirme katmanı protokolü uyarınca kullanılır.
- IP adresler 32 bitlik birer sayıdır; bu sayı, yazımı ve gösterimi kolay olması açısından, ve aynı zamanda ağ yönetimi için sağladığı kolaylıklardan dolayı, 8 bitlik dört parçaya bölünmüştür.
- Herbir parça 0 ile 255 (28-1=255) arasında bir sayı olabilir ve parçalar 199.23.0.0, 172.8.5.3 adreslerinde görüldüğü gibi birbirlerinden noktalar ile ayrılır: a.b.c.d, 0 < a, b, c ve d < 255.
- Aşağıda, Şekil-7.15 de örnek bir yerel alan ağı, İnternet bağlantısı ve sistemlere atanan IP adresleri görülmektedir:

The diagram illustrates a network topology for a DMZ. A central router (TCP/IP A8) connects the internal network to the Internet. The internal network consists of a server (Sunucu) at 10.15.20.2 and several workstations (Kullanıcı) at 10.15.20.1, 10.15.20.5, 10.15.20.17, and 10.15.20.22. The DMZ section includes a firewall (FW) at 10.15.20.25, a proxy (Yönlendirici) at 10.15.20.21, and a web server at 10.15.20.9. The Internet is represented by a cloud.

- Sayfa **24 / 24**