

## 1. İŞLETİM SİSTEMİNE GİRİŞ

İşletim sistemleri konusu, bilgisayar bilimleri kapsamındaki en temel konulardan birini oluşturmaktadır. İşletim sistemleri, bilgisayar sistemlerinin gelişmesine paralel olarak gelişme göstermiştir. Çünkü yeni gelişen bilgisayar mimarisi, yeni istekler ve ihtiyaç duyulan güvenliğe göre işletim sistemleri gelişmiştir. Bu nedenle, kullanıcı ile bilgisayar arasında bir köprü görevi yürüten ve donanıma en yakın yazılım birimi olan işletim sisteminin ayrıntılarını incelemeye geçmeden önce, bir bilgisayar sisteminin yapısını genel olarak ele almak gerekmektedir.

Bir bilgisayar sisteminin genel olarak 4 bileşeni vardır.

1. Donanım (İşlemci (CPU), bellek ve I/O üniteleri gibi)
2. İşletim Sistemi (Ms-Dos, Unix, OS/2 gibi)
3. Sistem Yazılımları (Derleyiciler, Veritabanı ve Network Yazılımları)
4. Uygulama Yazılımları (Kullanıcıların kendi geliştirdikleri yazılımlar)

Bu noktada şu hususu açıklamak gerekir ki CPU (Central Processor Unit) bilindiği gibi bir bilgisayar sisteminin en temel bileşeni olup, aynı şekilde bilgisayar sistemlerindeki disk, printer, disket, terminal (ana makineye bağlı, sıradan uç kullanıcılar) vs. gibi I/O (Input/Output) üniteleri donanım kısmı olmaktadır.

Yazılım (software) ise, hem bilgisayar sistemini oluşturan donanım birimlerinin yönetimini hem de kullanıcıların işlerini yapmak için gerekli olan programlardır. Yazılım olmaksızın bir bilgisayar sistemi, bir takım elektronik kartlar, kablolar ve mekanik bazı parçalardan ibaret bir cihazdır. Bir bilgisayar sistemi, üzerine işletim sistemi (Operating Systems) ve onun üzerine de diğer yazılımların yüklenmesi ve çalıştırılmasından sonra gerekli işlevleri yerine getirebilmektedir.

Bilgisayar yazılımları genel olarak 2 ana grupta incelenebilir.

- Sistem Yazılımları (System Software)
- Uygulama Yazılımları (Application Software)

**Sistem Yazılımları (System Software);** bilgisayarın kendisinin işletilmesini sağlayan, işletim sistemi, derleyiciler (compilers) (Yazılım programında, yazılan programı makine diline çeviren program), çeşitli donatılar (facility) gibi yazılımlardır.

**Uygulama Yazılımları (Application Software);** bu kullanıcıların işlerine çözüm sağlayan örneğin çek, senet, stok kontrol, bordro, kütüphane kayıtlarını tutan programlar, bankalardaki müşterilerin para hesaplarını tutan programlar vs. gibi yazılımlardır.

Bütün sistem programları içinde en temel yazılım işletim sistemidir ki, bilgisayarın bütün donanım ve yazılım kaynaklarını kontrol ettiği gibi, kullanıcılara ait uygulama yazılımlarının da çalıştırılmalarını ve denetlenmelerini sağlar.

Modern bir bilgisayar sistemi, bir veya birden fazla işlemci (ya da diğer bir söylemle "CPU"), gerçek bellek (RAM), saatler, terminaller, diskler, bilgisayar ağı (network) birimleri, yazıcı üniteleri, CD sürücüsü, disket ve teyp üniteleri gibi I/O ünitelerinden oluşmaktadır. Doğal olarak bir bilgisayar sistemi oldukça karmaşık bir yapıdadır.

Programcıları, donanımın bu karmaşık yapısından etkilenmemelerini sağlamak ve disk gibi donanım ünitelerinin nasıl çalıştıklarını anlamak zorunda bırakılmaları için, donanımın üzerine ilave edilen yazılımların katmanlar şeklinde (layered system) oluşturulmaları ve bu sayede çok daha kolay bir şekilde, sistemin bütün parçalarının yönetilebilmesi şeklinde bir yapılanma, uzun yıllar önce geliştirilmiş bir yaklaşımdır.

Bu yapının en alttaki üç katmanı donanımı oluşturmaktadır. En alttaki katman, fiziksel üniteler, entegre devreler, kablolar, power (elektrik destek) üniteleri, disket sürücüleri, disk üniteleri ve diğer benzeri donanım birimlerinden oluşmaktadır. Bu katmanın mimari yapısı ile ilgilenmek ve bunları çalışma prensiplerini geliştirmek elektronik mühendislerinin işidir.

- Uygulama Yazılımları
- Sistem Yazılımları
- İşletim Sistemleri
- Makine Dili
- Mikroprogram
- Fiziksel Birimler

Fiziksel üniteleri (donanım) doğrudan kontrol eden ve fiziksel katmanın bir üstündeki en ilkel yazılım düzeyini oluşturan katman, "*microprogram*" dır. Bu katman genellikle "read-only" (yalnızca okunabilir) bellek (ROM) sahasında bulunur. Microprogram da ekleme (add), taşıma (move), karşılaştırma (compare) gibi makine diline ait temel komutları adım adım yerine getirir. Microprogram, örneğin, Add işlemini yerine getirmek için eklenecek sayının nerede yer aldığını saptar ve üzerine eklenecek sayıyı ilave ettikten sonra sonucu elde eder.

Microprogramın yorumladığı komut seti (instruction set), makine dili (machine language) katmanını oluşturur ki, bu gerçekte makinenin gerçek bir donanım parçasını

oluřturmakta ve bir bilgisayar da donanımın gerek bir parasıymıř gibi nitelenebilmektedir. Bu nedenle bazı makinelerde Microprogram, donanımın iinde varsayıldığından ayrı bir katman řeklinde bulunmaz.

Makine dili, genel olarak 50 ile 300 arasında komuta (intruction) sahiptir. Veri tařıma, aritmetik iřlem yapma ve deęerleri kıyaslama gibi iřlevleri yerine getirir. Makine dili, yazıcı ve disket sőrücü gibi I/O (input/output) ünitelerinin denetimini özel yüklenmiř bilgiler sayesinde yapmaktadır.

Bir bilgisayar sisteminin dördüncü katmanında yer alan iřletim sisteminin temel iřlevi, donanımın karmařıklığına kullanıcıya yansıtılmamak ve daha elverişli ortam hazırlayıp, kullanıcının kolayca iřini yapmasını saęlamaktır.

İřletim Sistemi, üzerinde yer alan bazı yazılımlar "*Sistem Yazılımı*" olarak anılır. Örneęin derleyiciler (compiler); yazdığımız programı makine diline eviren ara program, editörler, yararlı programlar (utility) ; virüs temizleyen programlar gibi gerek iř için yardımcılarıdır, veritabanı yönetim sistemleri (database management system) ve bilgisayar aęı yazılımları (network software) yine birer sistem yazılımlarıdır. Ancak bu yazılımlar İřletim Sisteminin kendi öz paraları deęildir.

Özet olarak İřletim Sistemi, aynı zamanda donanım üreticisi olan veya yalnızca yazılım geliřtiren özel bir firma tarafından yazılıp pazarlanan ve bir bilgisayar sisteminin donanım ve yazılım kaynaklarını kontrol eden ve kullanıcılarında kendi özümlelerini geliřtirebildikleri ortamı hazırlayan bir sistem yazılımıdır.

## **2. İŞLETİM SİSTEMİ TANIMI**

En büyüğünden en küçüğüne, bütün genel amaçlı bilgisayarlarda çalışan programlar, bir işletim sistemine gereksinim duyarlar. Bu yüzden bilgisayarlarda herhangi bir program çalıştırılmadan önce İşletim Sistemi ile programların ana belleğine (RAM) yerleştirilmeleri gerekir. Bu işlem genellikle bilgisayar ilk açıldığı zaman otomatik olarak yapılır ve HDD' deki İşletim Sistemi ana belleğe yüklenir.

Bir işletim sisteminden beklenen hizmet, donanım ve yazılım kaynaklarının uyumlu ve verimli bir şekilde birlikte işletilmesidir. Örneğin kullanıcı Cobol veya Pascal dili ile geliştirdiği uygulama programını, bir bilgisayar sisteminde çalıştırabilmesi için, uygulama programı ve verilerini yazabileceği bir disk ünitesi, verilerini yazdıracığı yazıcı ünitesi, bu programı işletecek işlemci (CPU) ve gerçek bellek gibi donanım birimlerinin yanı sıra derleyici (compiler), yükeyici (loader) ve network yazılımları gibi yazılım birimlerine de ihtiyaç vardır.

**İşletim sistemi, bir bilgisayar sisteminde kullanıcı ile iletişim kurarak, donanım ve yazılım nitelikli kaynakların kullanıcılar arasında adil bir biçimde paylaşılmasını ve donanım ile yazılım birimlerinin etkin bir biçimde kullanılmasını sağlayan sistem programları topluluğuna denir.**

İşletim Sistemi, bilgisayar donanımı ile bilgisayar kullanıcısı arasında bir arayüz (interface) görevini gören programlar topluluğudur. Bu programlar topluluğunun genel amacı, bilgisayar kullanıcılarına programlarını çalıştırabilecekleri ortamı yaratmak ve bilgisayar sisteminin etkin ve verimli olarak kullanılmasını sağlamaktır.

### **3. İŞLETİM SİSTEMİNİN TARİHÇESİ**

İlk gerçek "Sayısal Bilgisayar" İngiliz matematikçilerinden Charles Babbage (1792-1871) tarafından tasarlanmıştır. Ancak onun yaşadığı yıllarda teknoloji yetersizliklerinden, tasarladığı makinelerde işletim sistemleri mevcut değildi.

#### **a) Birinci Nesil İşletim Sistemleri (1945-1955)**

Babbage' in başarısızlıkla sonuçlanan çalışmalarından sonra, II. Dünya savaşına kadar olan dönemde yok denecek kadar az bir gelişme olmuştur. 1940' lı yıllarda ise, Harvard Üniversitesinde Howard Aiken; Princeton Üniversitesinde, John Von Neumann ve Amerika ile Almanya'daki bazı diğer araştırmacıların çalışmaları sonucunda vakum tüpleri kullanılarak sayısal bazı makinelerin geliştirilmesi mümkün olabilmiştir. Ancak bu geliştirilen makineler son derece büyük ve odalar dolusu on binlerce vakum tüplerinden yapılmış ve bugün evlerde kullanılan bilgisayarlardan yüzlerce kez daha yavaş çalışmaktaydılar.

Bu dönemde makinenin hem tasarımını yapan, hem imalatını yapan, hem programlayan, hem işleten ve hem de bakımını yapan hep aynı küçük bir gruptu. Bütün programlama, kontrol panelindeki ilgili yerlere, ilgili kabloları takarak makine dili ile yapılırdı. İşletim sisteminin ise adı bile anılmamaktaydı. Sonraları 1950' li yılların başında kartlı makinelerin gelişmesi ile programların kartlara yazılıp buradan okutulması sağlanmakla beraber, diğer olaylar tümüyle aynıydı.

#### **b) İkinci Nesil İşletim Sistemleri (1955-1965)**

1950' li yılların ortasında transistörlerin geliştirilmesi ile büyük bir devrim oldu. Bu dönemde bilgisayarlar müşterilerin işlerini yapabilecekleri düzeye geldiği için üretici firmalar tarafından satılmaya başladılar. Bu yıllarda, bilgisayar tasarımcıları, üreticileri, operatörler, programcılar ve bakım personeli kesin olarak birbirinden ayrıldılar.

Bu makineler yine de çok büyük ve çok pahalı olduklarından, çok büyük kapasiteli klima cihazları ile soğutma gerektirdiğinden ve çok büyük devlet daireleri ya da çok büyük özel sektör kuruluşları tarafından satın alınabildiler. Bu nesil bilgisayarlarda, kullanıcı her bir satırını bir karta yazdığı programını getirip eliyle sistem operatörüne verirdi.

Operatör kartları kart okuyucu cihazında okutur ve okunmuş seklini teyp bandına aktarırdı. Sonra sisteme derleyici bandını yükler ve arkasından da kullanıcının programının bulunduğu bandı yükleyerek derleme işlemini yapardı. Bu derleme işlemi tamamlandıktan sonra programın çalıştırılabilir halini 3. banda çıkar ve bunu tekrar sisteme götürüp çalıştırarak programın sonucunu yazıcıdan yazdırırdı.

Bu dönemde bundan sonra sağlanan en büyük aşama, derleyicinin bir defa yüklenmesinden sonra, çok sayıda farklı programcının programlarının 1 bant üzerine arka arkaya yüklenip çalıştırılması olanağı ile Yığın İşlem (*Batch Processing*) kavramının getirilmesi ve uygulamaya koyulmasıdır. Bundan önce bilindiği gibi her programcının programı için derleyici bandını da bir defa yükleme zorunluluğu vardı. Bu nesil bilgisayarlar bilimsel ve mühendislik işleri için ve FORTRAN dili ile kullanılırdı. İşletim sistemi ise IBM' in geliştirdiği ve 7094 makinelerin de kullanılan IBSYS' di.

### **c) Üçüncü Nesil İşletim Sistemleri (1965-1980)**

1960' lı yılların başına kadar üretici firmalar iki farklı üretim çizgisinde gittiler. Bir taraftan mühendislik ve bilimsel işlerde kullanılan bilgisayarlar, diğer taraftan da bankacılık ve sigortacılık şirketleri gibi ticari kuruluşlar tarafından kullanılan bilgisayarlar üretildi. Ancak bu durum çeşitli sorunlar yarattığından IBM firması bu iki farklı yaklaşımı tek bir yapı üzerinde birleştirmek ve sorunları gidermek amacı ile 360 mimarisini duyurdu.

Bu nesil bilgisayarların mimari yapısındaki en önemli yenilik transistörlerin yerine entegre devlerin kullanılmış olması idi. Böylece makinelerin boyutları küçülürken, çıkardıkları sıcaklıkta binlerce kat azalmıştı. Bununla beraber kullanım açısından bu mimari yapının getirdiği en önemli yenilik ise "*mutiprogramming*" tekniği idi. Eski nesil bilgisayarlarda, kart ya da bant okuma süresi boyunca CPU tamamen boş olarak beklemekte iken, bu nesilde belleğin parçalara ayrılıp, her parçada başka bir programın çalıştırılması sayesinde, örneğin bir program teypten okuma yaparken CPU atıl (boş) olarak durmamakta ve diğer programın gereksindiği hesaplama işini yapmakta idi.

Üçüncü nesil bilgisayarların getirdiği bir diğer önemli özellik, aynı anda gelen çok sayıda program destelerinin, kendinden önce gelenin çalışıp bitmesini beklemeden arka arkaya okutulup disk üzerinden sıra ile çalışmayı beklemelerinin sağlanması idi. Bu olanağa "*SPOOLING*" (Simultaneous Peripheral Operation On Line) adı verilmiştir. Spooling tekniği, yazıcı gibi paylaşımlı kullanıma uygun olmayan ünitelerin kullanıcılar tarafından hiç beklemeksizin kullanabilmelerine olanak sağlamıştır.

Örneğin var sayalım ki, aynı bir yazıcıda yazılmak üzere aynı anda 3 farklı kullanıcı programı tarafından 3 tane çıktı gönderilse ne olur? Eğer işletim sistemi ve onun kaynakları yöneten fonksiyonları olmasaydı, kağıt üzerinde ilk 5 satır mesela 1. kullanıcının, sonraki bazı satırlar 2. kullanıcının ve diğer bazı satırlar da 3. kullanıcının olurdu ki bu tam bir kaos yaratırdı. İşte işletim sistemi örneğin sahip olduğu Spooling mekanizması sayesinde bu kullanıcılar tarafından gönderilen işleri disk üzerinde sıra ile biriktirir ve yazıcı ünitesinden de sıra ile birbirine karışmadan yazdırır.

Özet olarak Spooling;

Paylaşımli kullanıma uygun olmayan çevre ünitelerinin, kullanıcılar arasında birbirlerini beklemelerine gerek olmaksızın paylaşıyorlarmış gibi kullanmalarını sağlar.

1. Paylaşımli kullanıma uygun olmayan çevre ünitelerinin, kullanıcılar arasında birbirlerini beklemelerine gerek olmaksızın paylaşıyorlarmış gibi kullanmalarını sağlar.
2. Hız bakımından birbirinden çok farklı üniteleri arasındaki bilgi transferinin etkin bir şekilde yapılabilmelerini sağlar.

Yine üçüncü nesil bilgisayarlarla gelen diğer bir özellik zaman paylaşımıdır (*Time-Sharing*). Bu yazılım teknolojisi ile de, aynı anda çok sayıda kullanıcının terminaleri başındayken çalıştırdıkları işlere yada terminal vasıtası ile olmasa da sistem üzerinde yığın işlem "Batch Processing" olarak çalıştırılan işlere CPU' nun sıra ile ve kısa sürelerle tahsis edilmesi sağlanabilmiştir. Bu sayede hem sistemde çalıştırılan işlerin hepsi CPU' yu kısa aralıklarla kullanabilmiş olmakta, hem de sistemde çalışan örneğin ekran başında oturan kullanıcılar CPU' nun yalnızca kendilerine servis verdikleri hisine sahip olurlar.

#### **d) Dördüncü Nesil İşletim Sistemleri (1980-....)**

LSI (Large Scale Integration circuits) entegre devrelerinin gelişmesi ile ve binlerce transistörü ihtiva eden chiplerin 1 cm<sup>2</sup> üzerine yerleştirilmesi ile kişisel bilgisayar (PC – Personal Computer) devri doğmuş oldu.

O dönemdeki kişisel bilgisayarlar mimari bakımından mini bilgisayarlardan farklı olmamakla beraber, fiyatı bakımından çok daha ucuzdular. PC'lerin gelişmesi ve bunlar üzerinde çalışabilecek yazılımların, hiç bilgisayar bilgisi olmayan kişiler tarafından da kullanılabilir olması bu nesil bakımından evrim olmuştur. Bu nesilde iki tane işletim sistemi sektöre hakim olmuştur. Bunlardan bir tanesi Ms-Dos, diğeri de Unix'tir. 1980' li yılların ortalarında ilginç bir teknolojik yapılanmada başlamıştır. PC'lerin Ağ İşletim Sistemleri (Network Operating System) ve Dağıtık İşletim Sistemleri (Distributed Operating System) ile kullanılmaya başlamasıdır.

Bir ağ işletim sisteminde, kullanıcılar ortamda çok sayıda bilgisayarın mevcut bulunduğunun farkında olurlar ve aynı zamanda uzaktaki başka bilgisayarlara Uzaktan Bağlanma (Remote Login) olabildikleri gibi dosyalarını bir bilgisayardan diğerine kopya edebilirler. Ağ işletim sistemindeki, en önemli özelliklerinde biri de, her makinenin kendi yerel işletim sistemi tarafından işletilmesi ve her makinenin kendi kullanıcılarına sahip olmasıdır.

Dağıtık işletim sistemlerin de, bunun tersine, gerçekte ortamda çok sayıda CPU, olduğu halde, ortamın kullanıcıya sadece geleneksel tek işlemcili gibi görünmesidir. Bir gerçek dağıtık sistemde, kullanıcılar programlarının nerede çalıştırıldığıнын ve dosyalarının nerede yerleşmiş olduğunun farkında olmazlar. Bu işlemlerin hepsi otomatik olarak ve etkin olarak işletim sistemi tarafından gerçekleştirilir.

#### 4. İŞLETİM SİSTEMİNİN KATMANLARI

Bir işletim sisteminin yazılım tasarımında ele alınması gereken iki önemli konu bulunmaktadır;

**1. Performans:** İşletim sistemi, makine kaynaklarını (özellikle MİB zamanı ve bellek alanı) en etkili şekilde kullanılmasını sağlayacak şekilde tasarlanmalıdır. Makinenin donanımsal performansını en iyi şekilde kullanabilmelidir.

**2. Kaynakların özel kullanımı:** İşletim sistemi, kaynakların yalıtımını sağlamalıdır, diğer bir deyişle bir işlemin diğer işleme ait kaynaklara olan müdahalesine veya bu işleme ait bilgilerin silinmesine izin vermeyen bir koruma mekanizması geliştirmelidir. Her işletim sisteminin kaynaklara ulaşımı yönetmede kullandığı stratejiyi belirleyen bir güvenlik politikası vardır.

Her işletim sisteminin tasarımında olan üç temel unsur ise şunlardır;

1. İşlemci modları
2. Çekirdek (Kernel)
3. Sistem servislerini uyarma metodu

##### **İşlemci Modları**

İşlemcide, bir programın çalışma yeteneğini gösteren bir mod biti bulunmaktadır. Bu bit 'supervisor (kernel)' veya 'kullanıcı' modunu belirlemede kullanılır. İşlemci supervisor modda iken donanımsal her tür komutu çalıştırırken kullanıcı modunda ise bazı komutları çalıştırabilir. Supervisor moda çalışan komutlara supervisor, öncelikli veya korunmuş komutlar denilmektedir. İşletim sistemi programları supervisor moda çalışırken diğer tüm yazılımlar kullanıcı modunda çalışmaktadır. Örneğin giriş/çıkış işlemleri supervisor moda çalışmakta, kullanıcı modunda yer alan bir program herhangi bir giriş/çıkış işlemi yapılmasını istediğinde bunu işletim sisteminin yapmasını istemektedir.

Sistem aynı zamanda mod bitini kullanarak bellek alanları tanımlar. Eğer mod biti supervisor modda olacak şekilde ayarlandığında işlemcide çalışan işlem hem belleğin supervisor hem de kullanıcı alanlarına ulaşabilir. İşlemci kullanıcı modunda çalışırken ise bu işlem bellekte sadece kullanıcı alanına ulaşabilir. İşletim sistemlerinde bu alanlara kullanıcı ve sistem (supervisor, çekirdek veya korunmuş) alanı denilmektedir.

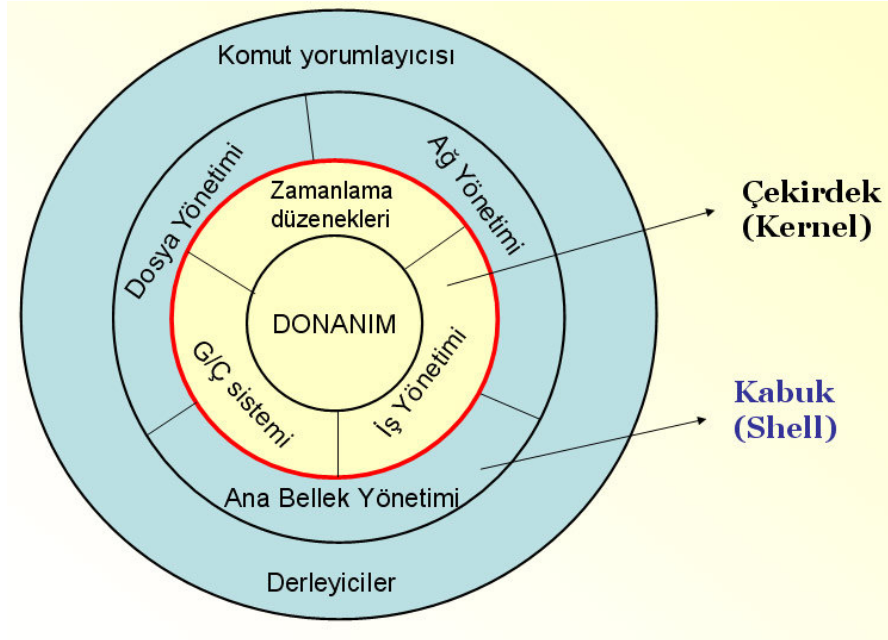
Genel olarak; mod biti işletim sisteminin koruma haklarından biridir. İşletim sistemi supervisor modda çalışmakta ve kullanıcı moduna göre belleğe ve öncelikli komut kümesine ulaşmakta daha fazla haklara sahip olmaktadır. İşlemci supervisor moda geçtiğinde işletim sisteminin kodlarını çalıştırmaktadır.

Kullanıcı modundaki bir işlem işletim sistemini çağırdığında işlemci hemen supervisor moda, mod bitini kullanarak geçer ki bu duruma supervisor çağrı (veya sistem çağrısı) denilmektedir. Örneğin; Word'de büyük bellek gerektiren bir dosya açınca başka işlemlerin alanlarına müdahale edilir. Bunu önlemek amacıyla yeni bir alan bu dosya için eklenmelidir. Burada supervisor çağrı yapılmıştır.

8086/8088 gibi eski işlemcilerde mod biti bulunmadığı için supervisor ve kullanıcı komutları arasında bir ayırım yapılmamaktadır. Bu da kaynakların paylaşımını ve yalıtımını güçleştirmekteydi.

### Çekirdek

İşletim sisteminin supervisor modda çalışan ve diğer parçaları için temel servisleri sağlayan en önemli parçasıdır. İşletim sisteminin uzantıları kullanıcı modunda çalışır ve daha sınırlı haklara sahip olur. Çekirdekte çalışan işletim sistemi fonksiyonları ise belleğe ve çekirdeğin diğer bölümlerine ulaşmada daha fazla haklara sahiptir. Kabuk (shell) veya diğer adıyla komut yorumlayıcısı ise kullanıcının sisteme verdiği komutları anlayan ve çalıştıran bir programdır. Kabuğun genellikle bir arayüzü bulunmaktadır; örneğin DOS'taki C:> nin görüldüğü komut istemi arayüzü ve kullanıcının girdiği DIR komutu. Çekirdek ve kabuk bazı işletim sistemlerinde ayrı iken bazılarında da sadece kavramsal olarak ayrılmıştır.



Monolitik çekirdekler (monolithic kernel), 1970–1990 arasında kullanılan ilk çekirdeklerdir. Burada tüm yazılımlar, sürücüler işletim sisteminin çekirdeğinde yer almaktadır. Örnek olarak Unix verilebilmektedir. Çekirdek büyük olmasına karşın her tür fonksiyonu içerdiği için genelde hızlıdır. Monolitik çekirdeklerin boyutlarının çok büyük olduğu düşüncesi modüler yapıda olan mikro çekirdekleri (microkernel) yaratmıştır. Bu

çekirdeklerde sadece en önemli işletim sistemi fonksiyonları bulunduğu için oldukça küçük boyutta olmaktadır. Yeni bir donanım eklendiğinde onun sürücüsü de çekirdeğe tanıtılmaktadır. Bu çekirdeklere örnek olarak MS-DOS verilebilir.

### **Sistem servislerini uyarma metodu**

Kullanıcı işlemlerinin işletim sisteminden belli servisleri (program çalıştırma, giriş/çıkış ve dosya işlemleri, ağ erişimi gibi) sağlaması istendiğinde oluşan bir durumdur. Bu bir sistem fonksiyonunun çağırılması veya MİB'ne bir mesaj gönderilmesi (message passing) ile gerçekleşmektedir. Sistem çağrıları, işletim sistemi ve işlemler arasında bir arayüzdür. Bu çağrılar genellikle Assembly dili komutları şeklindedir. C ve C++ gibi bazı programlama dilleri bunu direkt olarak yapabilmektedir. Microsoft Windows ise bunu Win32 API ile gerçekleştirmektedir.

### **UNIX İşletim Sistemi Katmanları**

UNIX, 1969 yılında, Ken Thompson tarafından Bell laboratuvarlarında geliştirilmiş, çok kullanıcı, çok görevli yapıyı destekleyen bilgisayar işletim sistemidir. Günümüzde Windows tabanlı sistemlerden sonra en çok kullanılan ve kökleri UNIX'e dayanan işletim sistemi GNU/Linux'tur.

Unix işletim sistemi çekirdeğinde, işletim sisteminin temel fonksiyonlarını yapması için gerekli işlem, bellek, dosya yönetimi bulunmaktadır. Aygıt yönetimi öncelikli olarak çekirdeğin içinde yer almaktaydı. Ancak daha sonra tek bir donanımın eklenmesinin bile bütün çekirdeğin tekrar derlenmesini gerektirmesi nedeni ile aygıt yönetimi yine çekirdeğin içinde olmak üzere ayrı bir bölüm haline getirilmiştir. Sistem çağrı arayüzü yani kabuk kullanıcının uygulama katmanında gerçekleştirdiği işlemleri ve komutları alıp yorumlayarak çekirdeğe iletmekte ve çekirdekte direkt olarak donanım ile iletişime geçerek gerekli işlemleri yapmaktadır.

### **Windows NT/2000/XP Katmanları**

Windows NT mimarisi donanım soyutlama katmanı, NT çekirdeği, NT yürütücüsü ve birçok alt sistemden oluşmaktadır. Donanım soyutlama katmanı, bir donanımın birçok detayının soyutlanması (örneğin kesme adresleri) ile işletim sisteminin donanımsal adresleri kullanması yerine farklı soyutlamaları kullanmasını sağlamaktadır. NT çekirdeği donanıma en yakın işletim sistemi mekanizmalarını örneğin kesmelerin ele alınması ve iş parçacıklarının zamanlanması gibi işlemleri yürütmektedir. NT yürütücüsü ise NT çekirdeğinin en üstünde işlemler ve kaynaklarla uğraşan ve işlem, bellek ve dosya yönetimini yapan ve aygıt alt sistemini içeren bölümdür. Sistem çağrı yorumlayıcısı Win32 API tarafından gerçekleştirilmektedir. G/Ç alt sistemi, sistem çekirdek fonksiyonlarından ayrı bir alt sistem olarak tanımlanmış ve aygıt sürücülerini

içermektedir. NT alt sistemleri ise kullanıcı tarafında NT çekirdeğinin bazı işlemleri yapabilmesi için gerekli servisleri sağlamaktadır.

### **Sistem Kaynakları**

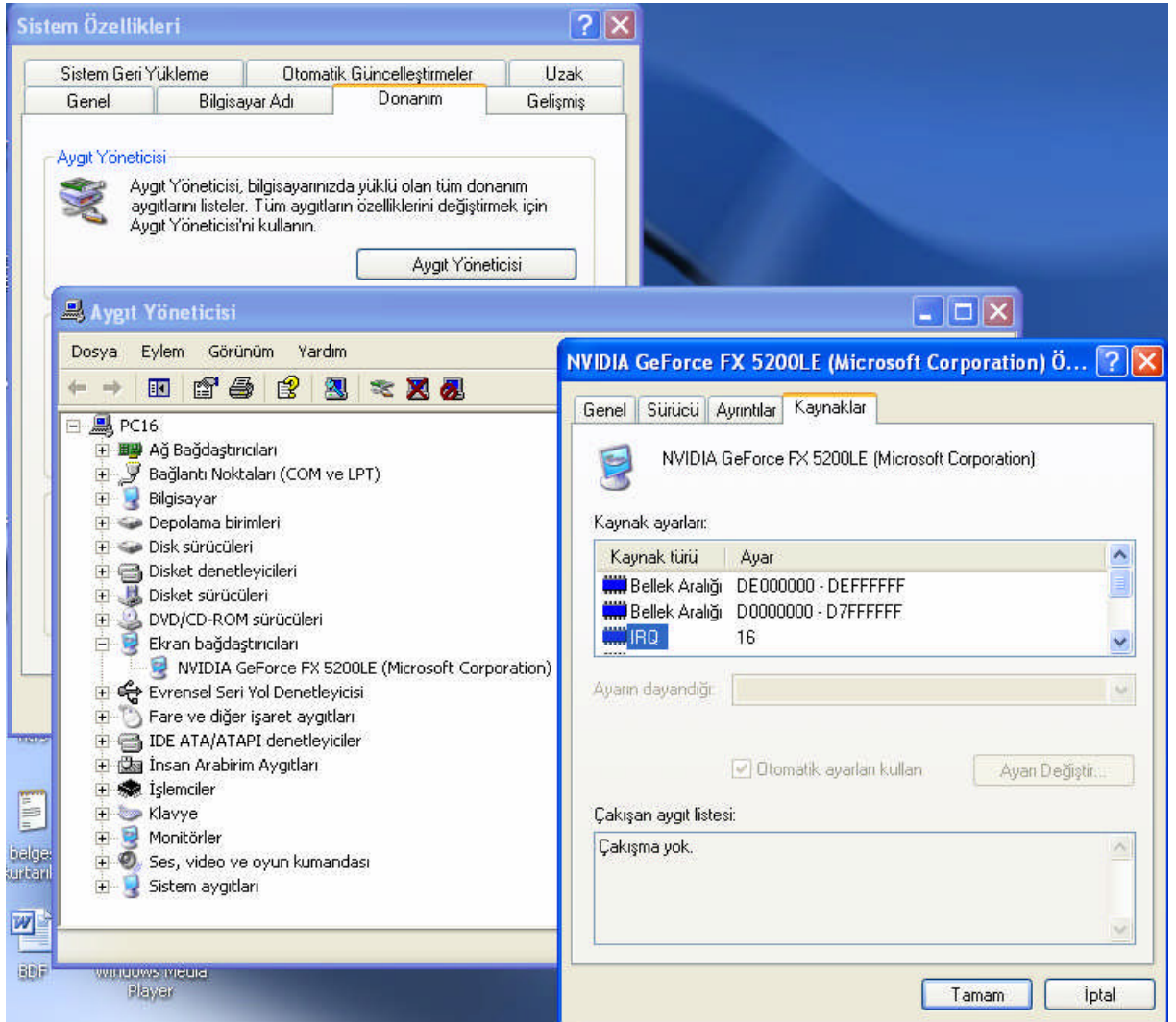
Sistem kaynakları, bilgisayar sistemi içerisinde kullanılan aygıtların (seri, paralel, usb port, fare v.s.), programların kontrol edilebilmesi, kullanıcılara hizmet edebilmesi için gerekli mekanizmaları anlatmak için kullanılan kelimelerdir. Sistem kaynakları, sistem içerisindeki donanım elemanlarının CPU ile haberleşebilmesi için paylaşılır. Sistem kaynakları iki veya daha fazla donanımın aynı zamanda haberleşmeye çalışmasını engeller. CPU'nun sistem aygıtlarını tanımlayabilmesini ve onlar ile haberleşebilmesini sağlar.

### **Kesme İstekleri (IRQ)**

IRQ İngilizce karşılığı Interrupt Request, yani kesme isteği kelimelerinin kısaltmasıdır. IRQ ile donanımlar sistem işlemcisi ile iletişim kurarlar. Bir nevi her donanımın özel adresi denebilir. Bilgisayarımızın merkezi işlem ünitesi olarak CPU çevre birimleri ile olan tüm iletişimleri başlatır, onların yönetimini elinde tutar. Peki herhangi bir çevre birimi CPU'nun kendisi ile ilgilenmesini nasıl sağlayacak, onun dikkatini nasıl çekecek. İşte bu noktada IRQ (Donanım kesmeleri) devreye girer. IRQ, çevre birimlerinin dikkat çekmek için kullandığı bir yöntemdir.

#### **- Kesmeler (IRQ) Nasıl Çalışır?**

Bilgisayarımızdaki kesmeler Intel 8259 öncelikli kesme denetleyicisi (PIC) tarafından sağlanır. Önceleri ayrı bir yonga olarak bulunan bu kesme denetleyicisi, günümüz bilgisayarlarında anakartın yonga setinde yerleşik olarak bulunmaktadır. Bir kesme talebi geldiğinde 8259 CPU'yu elindeki işi geçici olarak durdurmaya ve hemen bu kesmeyi yönetmesine olanak sağlayan bir programı çalıştırmaya iter. CPU kesme hizmet programına dallanırken dönüş adresini yığın hafızada saklar ve işi bitince işleme yine kaldığı yerden devam eder. Birçok aygıt aynı anda kesme hizmeti isteyebilir. Sistem tarafından belirlenmiş öncelik sırasına göre talepler karşılanır. Genelde kesme hizmet programı yüksek önceliğe sahip bir işlem tarafından kesilebilir. Ama daha öncelikli veya eşit seviyedeki bir cihazdan kesme talebi gelirse o anki kesme programı bitene kadar bu istek saklanır.



Eğer bilgisayarımızda bir donanım cihazımız doğru çalışmıyor ise aygıt yöneticisi penceresinden donanım elemanını seçerek **çift tıklarız** ve de açılan sekmede **kaynaklar** sekmesi ile boş olan bir IRQ seçmeliyiz. Ancak unutmamak gerekir ki, bu ayarlar için bilgi düzeyimiz yeterli değil ise müdahale etmememiz gerekir.

### **Doğrudan Bellek Erişimi (DMA)**

DMA İngilizce karşılığı Direct Memory Access anlamına gelen direkt hafıza erişimi kelimelerinin kısaltmasıdır. Özellikle disk sürücüler ve benzeri cihazlar için bu seçeneğin aktif halde olması belli bir performans artışı sağlamaktadır. Çünkü bu durum sayesinde cihaz gerek duyduğu bilgileri işlemciye uğramadan direk olarak sistem belleğinden elde edebilir. Bu kanallar sistem belleğine bazı aygıtların (ses kartı, ethernet kartı gibi) erişimini hızlandırmak için kullanılırlar. Bir sabit disk disk denetleyicisi sabit diskten bazı verileri aldıktan sonra bunları RAM'e depolamak ister. Aynı şekilde yerel iletişim ağı (ethernet) kartından da veri geldiğinde bunların RAM'e depolanması gerekebilir. Bunları I/O adresleri üzerinden CPU'ya oradan da RAM'e göndermek yerine bazı kartların

kullanabildiği DMA (Direct Memory Access - direk bellek erişimi) kanalları vasıtasıyla daha hızlı ve CPU'yu da meşgul etmeden direk RAM'e ulaştırmak mümkün. Bu sayede CPU meşgul edilmemiş olacak ve de bizim isteğimiz daha hızlı bir şekilde yerine getirilmiş olacaktır.

Tekrar özetlersek DMA verileri bir çevre biriminden RAM'e veya RAM'den çevre birimine CPU'nun müdahalesine gerek kalmadan aktarabilmeyi sağlar. Çevre birimlerinin birbirine direk ulaşmasına imkan sağlayamaz. Sisteminize DMA kullanmak üzere kaç tane kart takılabileceği sınırlıdır. Hafıza erişim bilgilerini de **kaynaklar** sekmesinden görebiliriz. Ancak erişim adres bilgileri çoğunlukla bizim değiştirebileceğimiz bilgiler değildir. Bütün donanım kartları ile ilgili erişim adresleri bilgilerini bilmemiz gerekir ki bu da çok düşük bir olasılıktır.

### **Giriş/ÇıkışAdresleri (I/O)**

CPU'nun çevre aygıtlarıyla ve devre kartları (ses kartı, ethernet kartı vs.) ile iletişim kurmak ve bu aygıtları birbirinden ayırt edebilmek için kullandığı Giriş/Çıkış (Input/Output) adresleridir. Bu adresler "port adresleri" veya "donanım adresleri" olarak da bilinir. Zaten CPU'nun dışdünya ile iletişim kurmak için kullandığı iki yol vardır denilebilir. Bunlardan biri bilgisayarımızın ana belleğinin adresleri diğeri de bahsedildiği üzere I/O adresleridir.

### **I/O Adres Çakışmaları**

Her kartın mikroişlemci ile haberleşmesi için farklı bir I/O adresi vardır. Birden fazla kartın aynı adresi kullanması durumuna çakışma denir. İki kartın aynı adresi kullanması durumunda mikroişlemci tarafından gönderilen komutlar bu kartlar tarafından doğru algılanmaz. Bu durum kartların çalışmamasına ya da hatalı çalışmasına neden olur.

Çoğu çevre birimi ve kartlar tek bir I/O adres aralığını kullanır. En basit şekliyle klavyenin kullandığı I/O adres aralığını başka bir kart kullanmaya kalkarsa, bu kart çalışmayacak, bununla birlikte klavyeniz de devre dışı kalacaktır. Zaten kart üretilirken klavyenin I/O adresini kullanacak bir kart tasarımı yapılmaz. Çünkü bu adres sabittir, klavye denetleyicisi tarafından kullanılmaktadır ve bir standart haline gelmiştir. Kartlar üretilirken bunlar göz önünde bulundurulmuş önemli kriterlerdir. "Peki, o zaman I/O çakışmaları nasıl olabilir?" diye bir soru gelebilir aklınıza. Bazı I/O değerleri standart değildir, sorunlar da zaten bu aralık değerlerini kullanan kartlarda görülmektedir. Şayet aynı adresi birden fazla kart için ayarlarsanız çakışmaya sebep olacağı için kartlar görevlerini yapamayacaktır. Giriş-çıkış adresleri bilgilerini de **kaynaklar** sekmesinden görebiliriz.