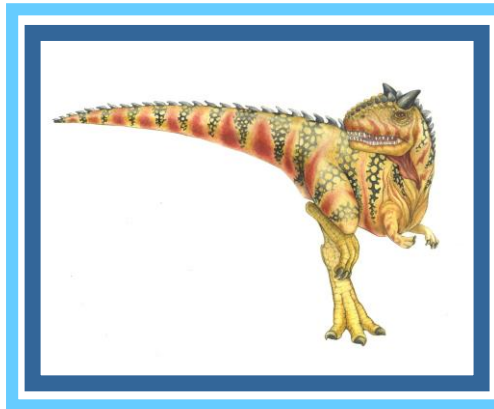




# 1. Bölüm

## İşletim Sistemlerine Giriş

Fırat Üniversitesi Bilgisayar Mühendisliği





# Ders Kaynakları

---

- Ders kitabı
  - Operating System Concepts, Silberschatz, Galvin ve Gagne
  - <https://codex.cs.yale.edu/avi/os-book/OS10/index.html>
- Ders sunumları
  - <https://codex.cs.yale.edu/avi/os-book/OS10/slide-dir/index.html>





# 1. Bölüm: Giriş

---

- İşletim Sistemleri Ne Yapar (What Operating Systems Do)
- Bilgisayar-Sistem Organizasyonu (Computer-System Organization)
- Bilgisayar-Sistem Mimarisi (Computer-System Architecture)
- İşletim-Sistemi Operasyonları (Operating-System Operations)
- Kaynak Yönetimi (Resource Management)
- Güvenlik (Security and Protection)
- Sanallaştırma (Virtualization)
- Dağıtık Sistemler (Distributed Systems)
- Çekirdek Veri Yapıları (Kernel Data Structures)
- Hesaplama Ortamları (Computing Environments)
- Özgür Açık Kaynak İşletim Sistemleri (Free/Libre and Open-Source Operating Systems)





# 1. Bölüm Hedefler

---

- Bir bilgisayar sisteminin genel organizasyonunu ve kesmelerin (interrupts) rolünü tanımlamak
- Modern, çok işlemcili bir bilgisayar sistemindeki bileşenleri tanımlamak
- Kullanıcı modundan çekirdek (kernel) moduna geçişi örneklemek
- Çeşitli hesaplama ortamlarında işletim sistemlerinin nasıl kullanıldığını anlamak
- Özgür ve açık kaynaklı işletim sistemi örneklerine değinmek





# İşletim Sistemi Nedir?

---

- Arabalardan, "Nesnelerin İnterneti" aygıtlarını içeren ev aletlerine, akıllı telefonlara, kişisel bilgisayarlara, kurumsal bilgisayarlara ve bulut bilişim ortamlarına kadar işletim sistemleri her yerdedir.
- **İşletim Sistemi**, bir bilgisayarın donanımını yöneten yazılımdır, bilgisayar kullanıcısı ile bilgisayar donanımı arasında aracı görevi gören bir programdır.





# İşletim Sisteminin Amaçları

---

- **İşletim sisteminin üç ana amacı:**
  1. Bir bilgisayar kullanıcısının bilgisayar donanımı üzerindeki programları uygun ve verimli bir şekilde yürütebilmesi için bir ortam sağlamak.
  2. Gerekli görevleri gerçekleştirmek için bilgisayarın kaynaklarını mümkün olduğunca adil ve verimli tahsis etmek.
  3. Bir kontrol programı olarak iki ana işleve hizmet eder:
    1. Bilgisayarın uygunsuz kullanımını ve hataları önlemek için kullanıcı programlarının yürütülmesini denetler.
    2. I/O cihazlarla ilgili işlemleri yönetir ve I/O cihazlarının kontrolünü sağlar.





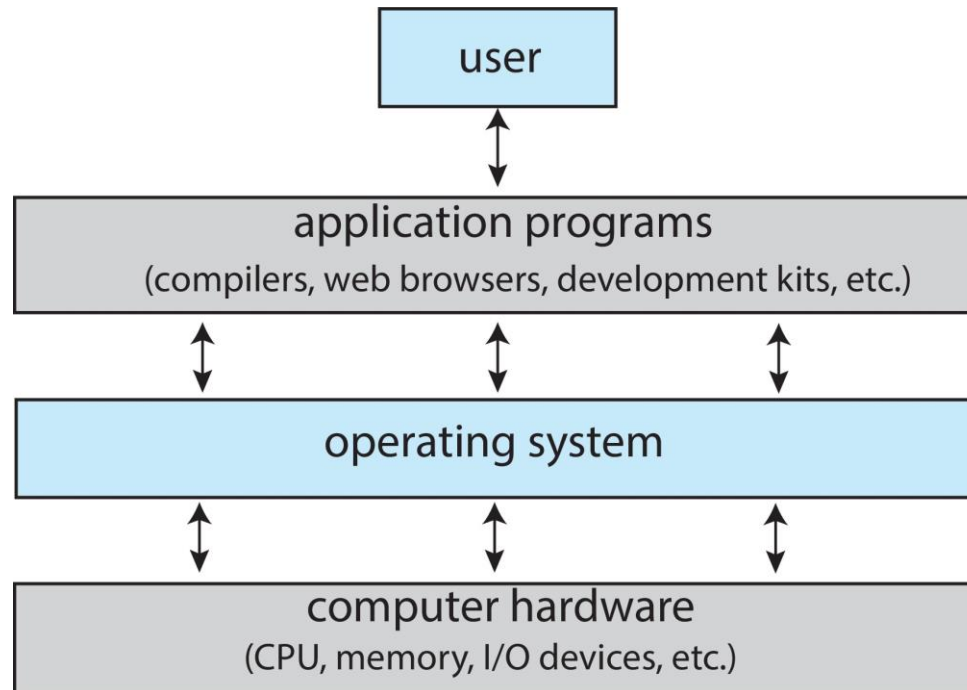
# Bilgisayar Sistem Yapısı

- Bilgisayar sistemi dört bileşene ayrılabilir:
  - Donanım - temel hesaplama kaynaklarını sağlar
    - ▶ CPU, memory, I/O aygıtları...
  - İşletim Sistemi
    - ▶ Çeşitli uygulamalar ve kullanıcılar arasında donanım kullanımını kontrol eder ve koordine eder.
  - Uygulama programları - kullanıcıların bilgisayar sisteminde yapmak istedikleri işler için hangi sistem kaynaklarının nasıl kullanılacağını belirler.
    - ▶ Metin işleyicisi, derleyici, web tarayıcı, veritabanı sistemleri, video oyunları ...
  - Kullanıcılar
    - ▶ İnsanlar, makineler, diğer bilgisayarlar





# Bilgisayar Bileşenlerinin Soyut Görünümü







# İşletim Sistemleri Ne Yapar

- İşletim sistemi ne yapar? Sorusunun cevabı biraz da bakış açısına bağlıdır;
- Kullanıcılar rahatlık, **kullanım kolaylığı** ve **iyi performans** ister
  - Kaynak kullanımını (**resource utilization**) önemsemez
- Ancak **mainframe** veya **minicomputer** gibi paylaşılan bilgisayarlar tüm kullanıcıları mutlu etmelidir.
  - İşletim sistemi, donanımın verimli bir şekilde kullanılmasını sağlayan ve kullanıcı programlarının yürütülmesini yöneten bir **kaynak ayırıcı (resource allocator)** ve **kontrol programıdır (control program)**.
- İş istasyonları (**workstations**) gibi sistemlerin (dedicate systems) kullanıcıları özel kaynaklara sahiptir, ancak sık sık sunuculardaki paylaşılan kaynakları kullanırlar.
- Akıllı telefonlar ve tabletler gibi mobil cihazlar kaynak açısından zayıftır, kullanılabilirlik ve pil ömrü için optimize edilmiştir.
- Cihazlarda ve otomobillerde yerleşik bilgisayarlar (**embedded computers**) gibi bazı bilgisayarların çok az kullanıcı arayüzü vardır veya hiç yoktur
  - Kullanıcı müdahalesi olmadan öncelikli olarak çalışacak şekilde tasarlanmıştır.





# İşletim Sistemlerinin Tanımı

- Bilgisayarların sayısız tasarımı ve kullanımı nedeniyle işletim sistemi terimi, birçok rolü ve işlevi kapsar.
- Bilgisayarlar; ekmek kızartma makinelerinden gemilere, arabalardan uzay araçlarına, oyun makinelerinden endüstriyel kontrol sistemlerine kadar bir çok yerde vardır.
- Bu çeşitliliği anlamak için bilgisayarların tarihine bakılabilir: Bilgisayar, neyin yapılabileceğini belirlemek için bir deney olarak başladı ve hızlı bir şekilde belli amaçlı sistemlerin kullanımına açıldı.
  - ▶ Bilgisayarların ilk kullanım amaçları: kod kırma ve yörünge planlaması gibi askeri kullanımlar, nüfus sayımı hesaplaması gibi kurumsal kullanımlar içindir.





# İşletim Sistemlerinin Tanımı

- İşletim sistemlerinin evrensel olarak kabul edilmiş bir tanımı yoktur.
- **Genel bir tanım olarak** işletim sistemi bilgisayarda her zaman çalışan çekirdek (**kernel**) olarak adlandırılan bir programdır.
- Çekirdekle birlikte iki tür program daha vardır:
  - **Sistem programları** (işletim sistemiyle birlikte gönderilir, ancak çekirdeğin bir parçası değildir)
  - **Uygulama programları**, işletim sistemiyle ilişkili olmayan tüm programlardır.
- İşletim sistemlerinin özellikleri günümüzde artmaktadır.
  - Mobil işletim sistemleri yalnızca bir çekirdekten ibaret değil, aynı zamanda uygulama geliştiricilerine veritabanları, multimedya, grafikler gibi ek hizmetler sağlayan bir dizi yazılım frameworkleri sağlayan bir ara yazılım (**middleware**) içerir.





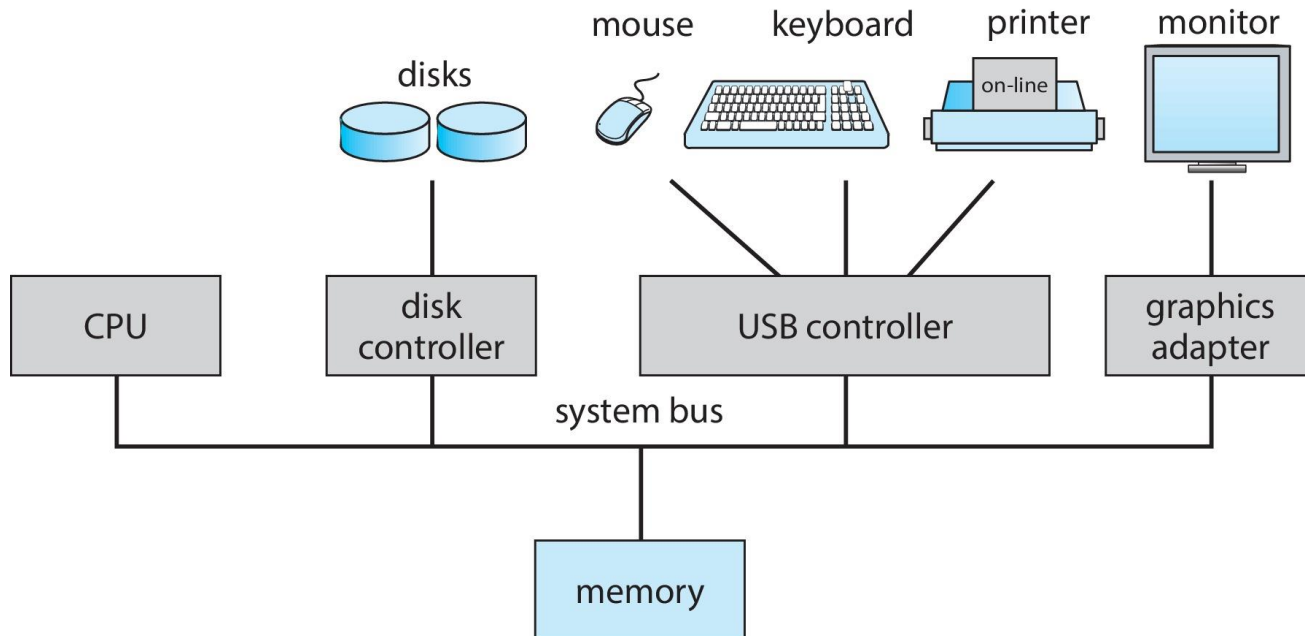
# Bilgisayar Sistem Yapısına Genel Bakış





# Bilgisayar Sistem Organizasyonu

- Modern bir genel amaçlı bilgisayar sistemi, bir veya daha fazla CPU ve bir dizi aygıt denetleyicinin ortak bir veri yolu (**bus**) üzerinden bağlanmasından oluşur. Ortak veri yolu, bileşenlerle paylaşılan bellek arasında erişimi sağlar.
- CPU ve aygıt denetleyiciler paralel çalışabilir, bellek kullanımında rekabet içinde olabilirler.





# Bilgisayar Sistem Organizasyonu

---

- I/O aygıtları ve CPU aynı anda çalışabilir.
- Her bir aygıt denetleyicisi, belirli bir aygıt türünden sorumludur.
- Her aygıt denetleyicisinin lokal bir bufferi vardır
- İşletim sistemlerinin her bir aygıt denetleyicisi için bir aygıt sürücüsü (**device driver**) vardır.
- CPU, verileri lokal bufferlerden ana belleğe, ana bellekten lokal bufferlere taşır.
- I/O, aygıttan aygıt denetleyicisinin lokal bufferine doğrudur.
- Aygıt denetleyicisi, CPU'ya bir kesmeye (**interrupt**) neden olan işlemin bittiğini bildirir.





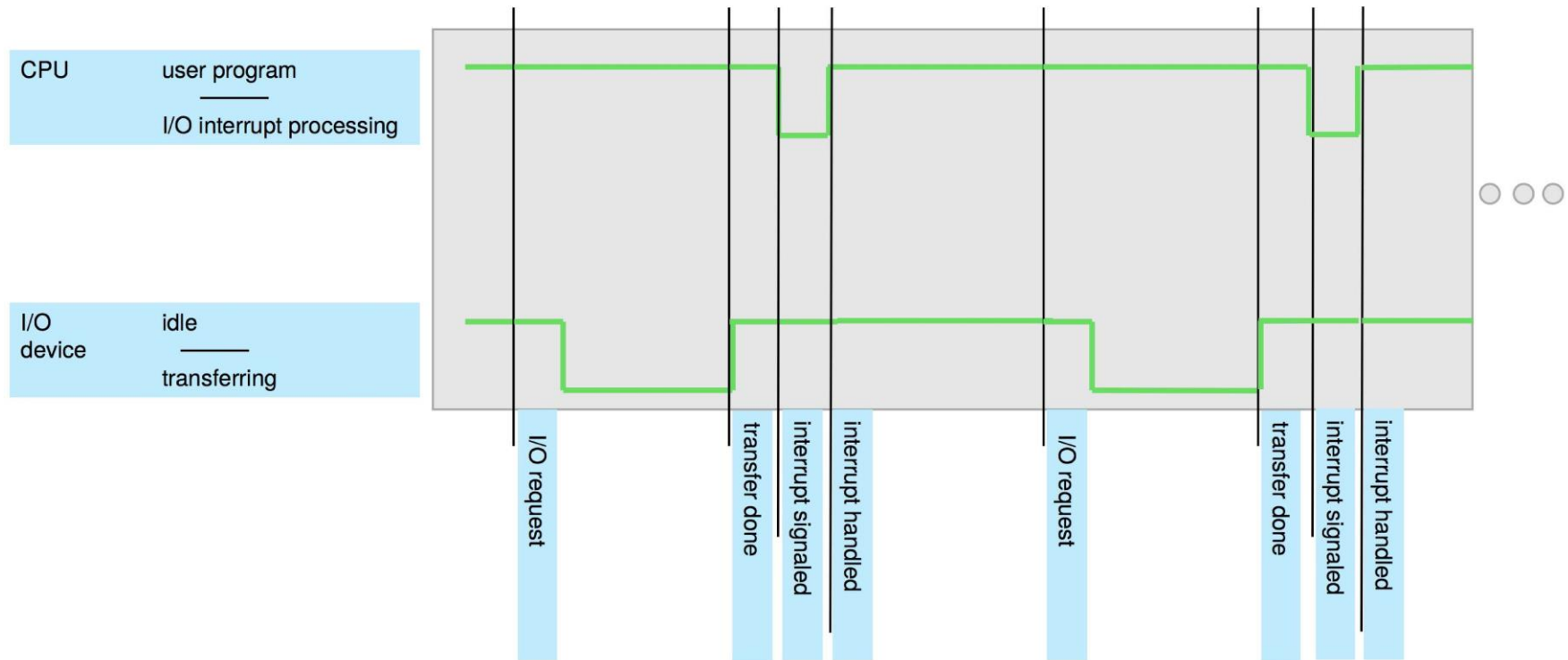
# Interruptlerin Yaygın İşlevleri

- **Kesmeler, bir bilgisayar mimarisinin önemli bir parçasıdır.** Bir işletim sistemi kesmelere dayalıdır (**interrupt driven**). Her bilgisayar tasarımının kendi kesme mekanizması vardır, ancak birkaç işlevi ortaktır.
- Temel interrupt(kesme) mekanizması, bir aygıt denetleyicisi hizmete hazır hale geldiğinde CPU'nun asenkron bir olaya yanıt vermesini sağlar.
- Kesme, uygun **kesme servis rutinine (interrupt service routine ISR)** diğer bir adıyla **kesme işleyicisine (interrupt handler)** aktarılır. Gelen kesmenin id sine interrupt vektörde bakılarak ilgili interrupt service rutinene hızlıca transfer edilir.
- Kesme mimarisi kesme komutunun adresini kaydetmelidir.
- **trap** ya da **exception**, yazılım tarafından oluşturulan bir hatadan ya da kullanıcının isteğinden kaynaklanan kesmelerdir. Örneğin sıfıra bölme hatası, geçersiz bir bellek adresine erişme gibi.





# Interrupt Timeline







# Interrupt Handling (Kesme İşleyicisi)

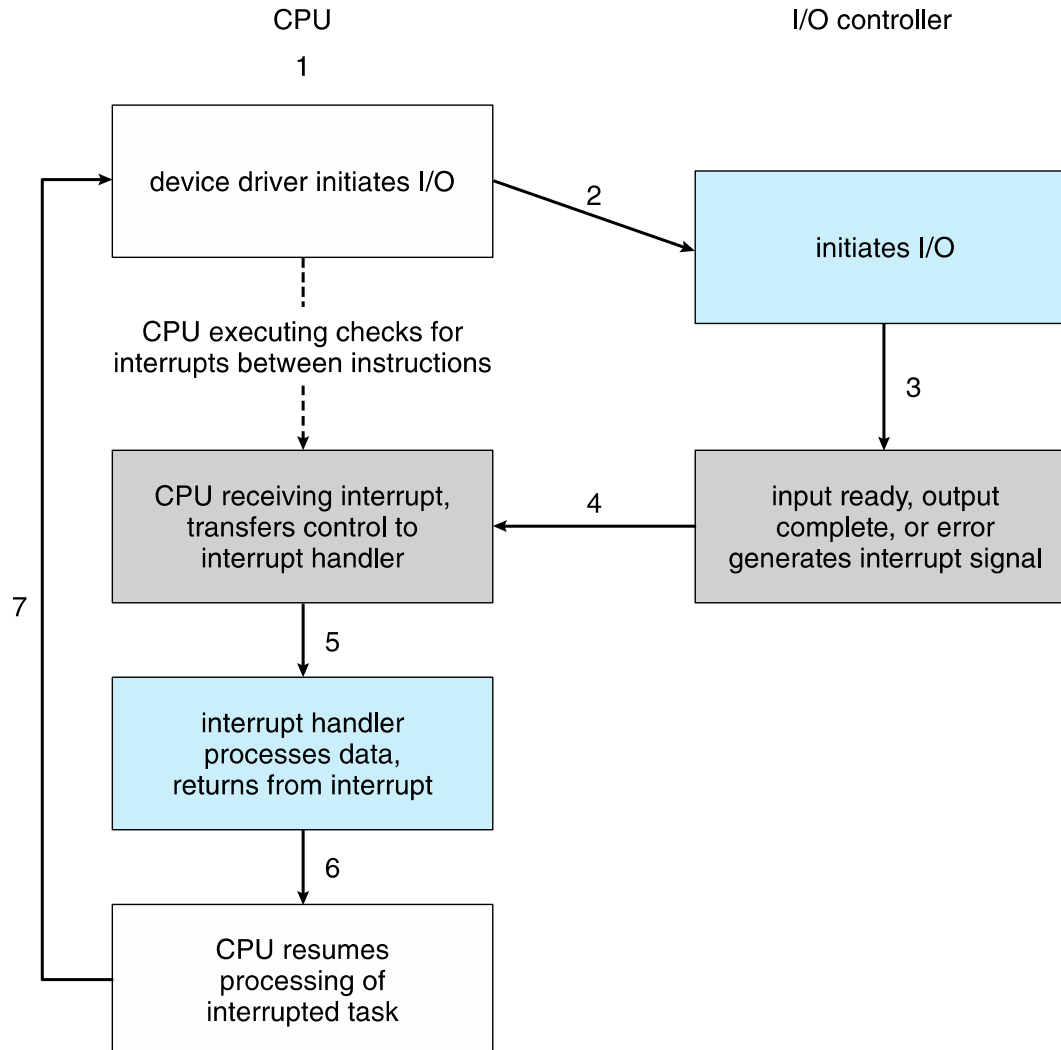
---

- Interrupt handling mekanizması yürütülürken CPU'nun durum bilgisi registerlere ve program counter'lere kaydedilerek korunur.
- Interrupt işlemi bittikten sonra CPU interrupt gelmeden önceki yürütme durumuna (execution state) döndürülür.
- Interrupt handling hangi tür kesmelerin oluştuğunu belirler.
  - Komutu segmentlerine ayırır her bir interrupt türü için ne yapılması gerektiğini belirler.





# Interrupt-drive I/O Cycle





# I/O Yapısı

---

- I/O yu işlemek (I/O handling) için iki yöntem vardır:
  1. I/O başlar, kontrol yalnızca I/O tamamlandığında kullanıcı programına döner.
  2. I/O başlar, kontrol I/O tamamlanmasını beklemeden kullanıcı programına döner.





# I/O Yapısı

1. I/O başlar, kontrol yalnızca I/O tamamlandığında kullanıcı programına döner:
  - Bir sonraki kesmeye kadar CPU'yu boşa bırakır.
  - Belleğe erişimde bekleme döngüsü olur.
  - Bir seferde en fazla bir I/O isteği ele alınır, eşzamanlı I/O işleme yoktur.
2. I/O başlar, kontrol I/O tamamlanmasını beklemeden kullanıcı programına döner:
  - Sistem çağrısı (**System call**) – ile kullanıcının I/O nun tamamlanmasını beklemesine izin vermek için işletim sistemine istek yapılır.
  - Aygıt durumu tablosu (**Device-status table**) her I/O aygıtı için tipini, adresini ve durumunu gösteren bilgiler içerir.
  - İşletim sistemi, her bir aygıtın durumunu belirlemek ve tabloda interrupt bilgilerini güncellemek için I/O aygıt tablosunu indeksler.





# Bilgisayar Açılışı

---

- Bilgisayar açılırken veya yeniden başlatılırken başlangıç programı olarak **bootstrap programını** kullanır.
  - Genellikle ROM veya EPROM'da depolanır, **firmware** (bir cihazın belirli donanımı için alt seviyeli kontrol sağlayan yazılımdır) olarak bilinir.
  - CPU register'lerinden aygıt denetleyicilerine ve bellek içeriklerine kadar sistemin tüm yönlerini başlatır.
  - İşletim sistemi çekirdeğini (kernel) yükler ve işletim sisteminin çalışmasını başlatır.





# Depolama Yapısı (Storage Structure)





# Depolama Yapısı

- CPU komutları yalnızca bellekten yükleyebilir, bu nedenle herhangi bir programın çalışması için önce belleğe yüklenmesi gerekir.
- Genel amaçlı bilgisayarlar, programlarının çoğunu ana bellek (rastgele erişimli bellek veya RAM olarak da adlandırılır) adı verilen yeniden yazılabilir bellekten çalıştırır.
- Ana bellek - yalnızca CPU'nun doğrudan erişebileceği büyük depolama ortamıdır
  - **Rasgele/doğrudan erişimlidir (Random Access)**
  - **Geçicidir/uçucudur (volatile)** - güç kesildiğinde içeriğini kaybeder.
  - Tipik olarak **random-access memory, Dynamic Random-access Memory (DRAM)** adı verilen bir yarı iletken teknolojisinde uygulanır.





# Depolama Yapısı

- İdeal olarak, programların ve verilerin kalıcı olarak ana bellekte kalmasını isteriz. Ancak bu genellikle iki nedenden dolayı çoğu sistemde mümkün değildir:
  - Ana bellek genellikle gerekli tüm programları ve verileri kalıcı olarak saklamak için çok küçüktür.
  - Ana bellek, belirtildiği gibi geçicidir; güç kapatıldığında veya başka bir şekilde kaybolduğunda içeriğini kaybeder.
- Bu nedenle, çoğu bilgisayar sistemi, ana belleğin bir uzantısı olarak ikincil depolama (Secondary storage) sağlar, büyük **kalıcı (nonvolatile)** depolama kapasitesi sağlar.
- İkincil depolamaya duyulan gereksinim, büyük miktarda veriyi kalıcı olarak tutabilmesinden dolayıdır.







# Depolama Yapısı

- En yaygın ikincil depolama aygıtları, hem programlar hem de veriler için depolama sağlayan sabit disk sürücüler (HDD'ler) ve kalıcı bellek (NVM) aygıtlarıdır.
  - Çoğu program (sistem ve uygulama), belleğe yüklenene kadar ikincil depoda saklanır.
- **Hard Disk Drives (HDD)** – manyetik kaydedici malzeme ile kaplı sert metal veya cam plakalardan oluşur.
  - Disk yüzeyi mantıksal olarak **tracklar**, tracklar da içinde sektörler (**sector**) bölünmüştür.
  - Disk denetleyicisi (**disk controller**), cihaz ve bilgisayar arasındaki mantıksal etkileşimi belirler.
- **Kalıcı bellek aygıtları (Non-volatile memory (NVM))**– kalıcı hard disklerden daha hızlıdır. (Flash memory, SSD, vb.)
  - Çeşitli teknolojileri vardır.
  - Kapasite ve performansları arttıkça ve fiyatları düştükçe daha popüler hale geliyor.





# Depolama Hiyerarşisi

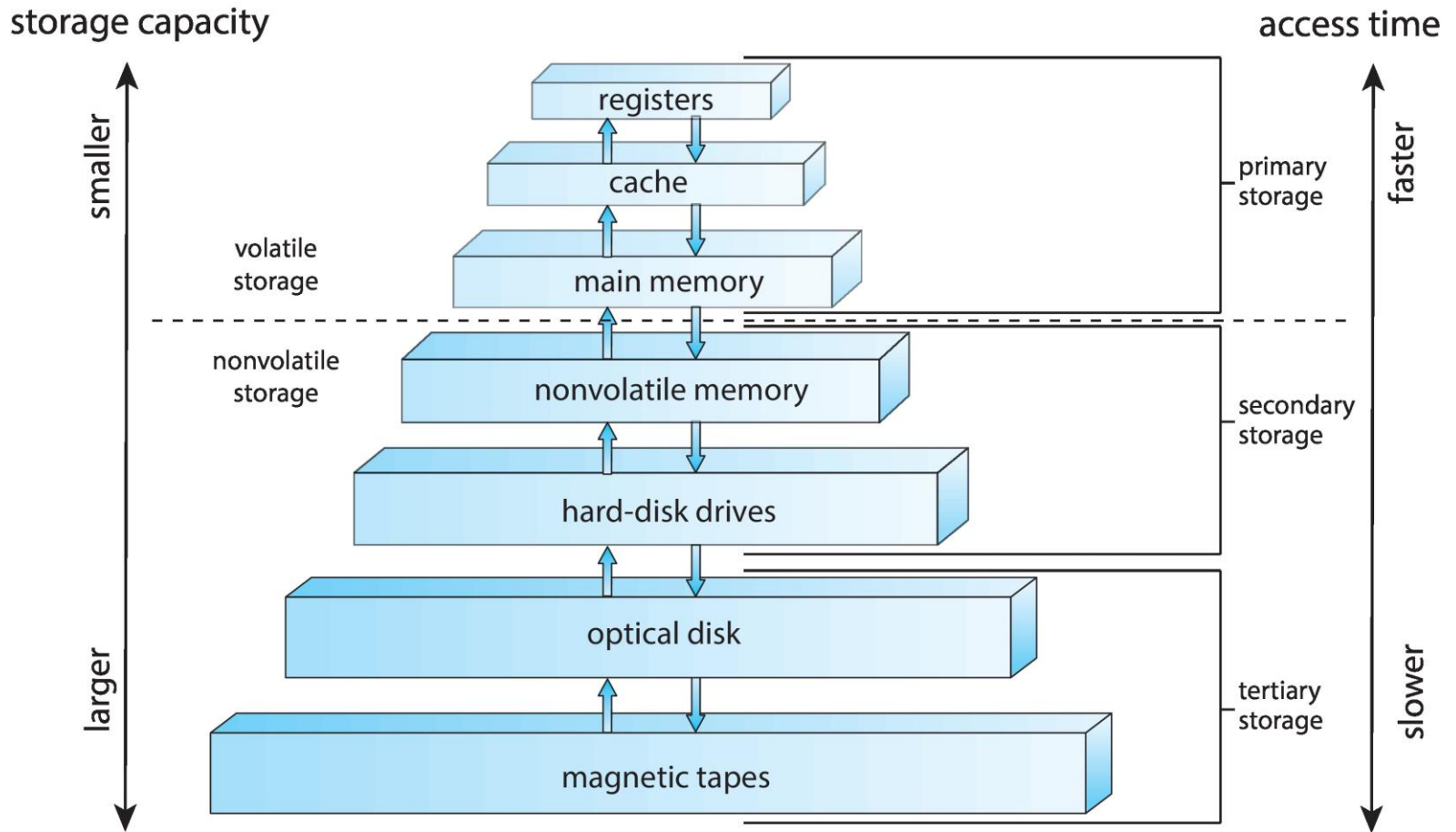
---

- Depolama sistemleri hiyerarşik bir biçimde organize edilmiştir:
  - Hız
  - Maliyet
  - Uçuculuk
- **Caching** – bilgiyi daha hızlı depolama sistemine kopyalamaktır; ana bellek, ikincil depolama için bir cache olarak görülebilir.
- **Aygıt Sürücüsü (Device Driver)** I/O yu yönetmek için her bir aygıt denetleyicisinin bir aygıt sürücüsü vardır.
  - Aygıt sürücüsü, aygıt denetleyicisi ve çekirdek arasında tekdüze bir arayüz sağlar.





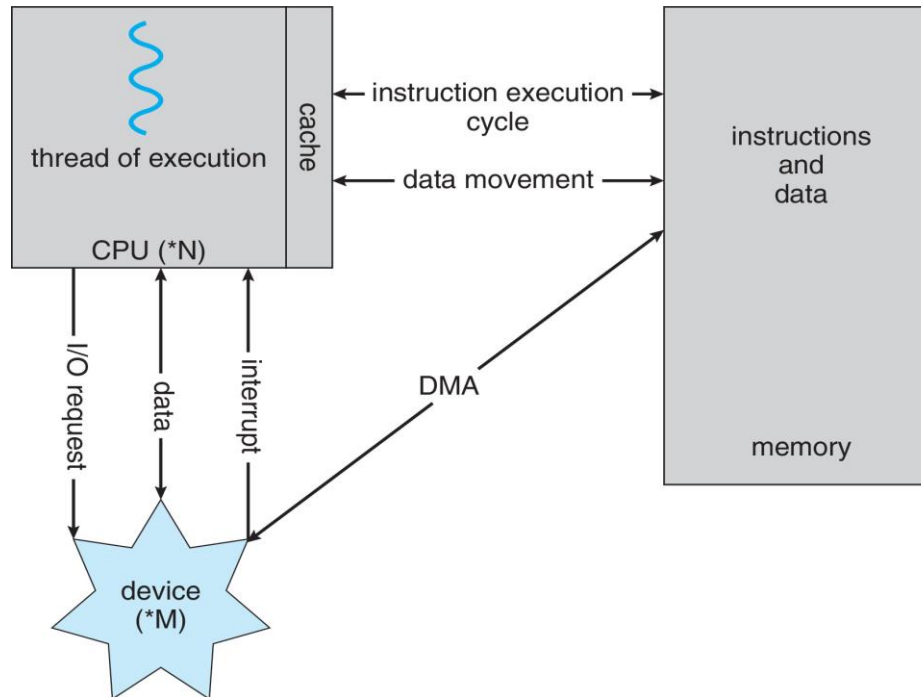
# Depolama Aygıtlarının Hiyerarşisi





# Modern Bir Bilgisayar Nasıl Çalışır?

- Şekilde bir bilgisayar sisteminin tüm bileşenlerinin etkileşimini gösterilmektedir.
- Von Neumann mimarisinde komut yürütme döngüsü (instruction execution cycle) şöyledir: bellekten bir komut alınır ve instruction registerine kaydedilir, daha sonra bu komut çözülür (decode) ve çalıştırılır, belki sonuç değeri yenden belleğe yazılır.



*A von Neumann architecture*





# Doğrudan Bellek Erişim Yapısı (Direct Memory Access Structure)

- Interrupt tabanlı I/O formu, küçük miktarlardaki verileri taşımak için uygundur ancak NVS (Nonvolatile storage) I/O gibi toplu veri taşınması için kullanıldığında yüksek ek yük üretebilir.
- Bu sorunu çözmek için bellek hızlarına yakın hızlarda bilgi iletebilen yüksek hızlı doğrudan bellek erişimi (DMA) aygıtları kullanılır.
- I/O aygıtı için arabellekleri, işaretçileri ve sayaçları ayarladıktan sonra, aygıt denetleyicisi, CPU'nun herhangi bir müdahalesi olmaksızın, tüm bir veri bloğunu doğrudan aygıta ve ana belleğe aktarır.
- Düşük hızlı aygıtlar için oluşturulan bayt başına bir kesme yerine, aygıt sürücüsüne işlemin tamamlandığını söylemek için blok başına yalnızca bir kesme oluşturulur.
- Aygıt denetleyicisi bu işlemleri gerçekleştirirken, CPU diğer işleri gerçekleştirmek için kullanılabilir.





# İşletim Sistemi Operasyonları

## (Operating-System Operations)

- Bootstrap program – sistemi başlatan, kerneli yükleyen basit bir kod parçasıdır.
- Kernel yüklenir.
- Sistem arka plan yordamları (**system daemons**) başlatılır (kernel dışında sağlanan hizmetler)
- Kernel, yazılımsal ve donanımsal olarak interrupt tabanlıdır (**interrupt driven**).
  - Donanım interruptları donanımsal aygıtlardan kaynaklanır.
  - Bir başka interrupt biçimi de bir hata (örneğin sıfıra bölme veya geçersiz bellek erişimi) ya da bir kullanıcı programının sistem çağrısı (**system call**) adı verilen özel bir işlemi yürüterek bir işletim sistemi hizmetinin gerçekleştirilmesini istemesi nedeniyle yazılım tarafından oluşturulan bir kesme olan **exception** ya da **trap'tir**.





# Multiprogramming (Batch system)

- İşletim sistemlerinin en önemli yönlerinden biri, birden çok programı çalıştırabilmeleridir. Tek bir kullanıcı her zaman CPU ve I/O aygıtlarını meşgul edemez.
- **Multiprogramming** ile process'ler organize edilerek CPU'nun her zaman yürütecek bir process'e sahip olması sağlanır, böylece hem CPU'nun verimliliği artırılır hem de kullanıcılar (*genellikle birden fazla program çalıştırmak isterler*) memnun edilir.
- **Multiprogramming** işletim sistemlerinde birden fazla program aynı anda bellekte tutulur, işletim sistemi bu programlardan birini yürütmek için **job scheduling** ile seçer.
- Yürütülen process'in tamamlanması için I/O gibi bazı işlemler gerekebilir. **Multiprogramming** olmayan bir sistemde process'in beklediği işlemler tamamlanana kadar CPU boşta bekler.
- **Multiprogramming** sistemde ise yürütülmek üzere başka bir process seçilir, bu process de tamamlanmak için beklemek zorunda kalırsa yürütülmek üzere başka bir process seçilir.





# Multiprogramming (Batch system)

---

- Process'in beklediği işlemler tamamlandığında OS tarafından tekrar yürütülür.
- En az bir process yürütülmek üzere hazır olarak beklediği müddetçe, CPU asla boşta kalmaz.
- Multicore bir sistemde, yürütülmeye hazır process olduğu müddetçe CPU'yu meşgul tutma kavramı, sistemdeki tüm işlemci çekirdeklerine genişletilir.







# Multitasking (Timesharing)

- **Multitasking, multiprogramming**'in genişletilmiş halidir. CPU aralarında geçiş yaparak birden fazla process'i yürütür.
- Ancak bu geçişler o kadar sık olur ki kullanıcı tek işlemcili bir sistemde bile birden çok işin aynı anda yürütüldüğünü sanır.
- Böylece hem CPU verimliliği hem de sistemin kullanıcıya cevap verebilirliği artmış olur.
  - Cevaplama zamanı 1 saniyeden küçük olmalıdır.
  - Her kullanıcının bellekte yürütülen en az bir programı vardır.
  - Eğer işler belleğe sığmazsa, **swapping** ile işler belleğe alınır, yürütülür ve bellekten çıkarılır.
  - **Sanal bellek (Virtual memory)** belleğe sığmayan işlemlerin yürütülmesine izin verir.





# Multitasking (Timesharing)

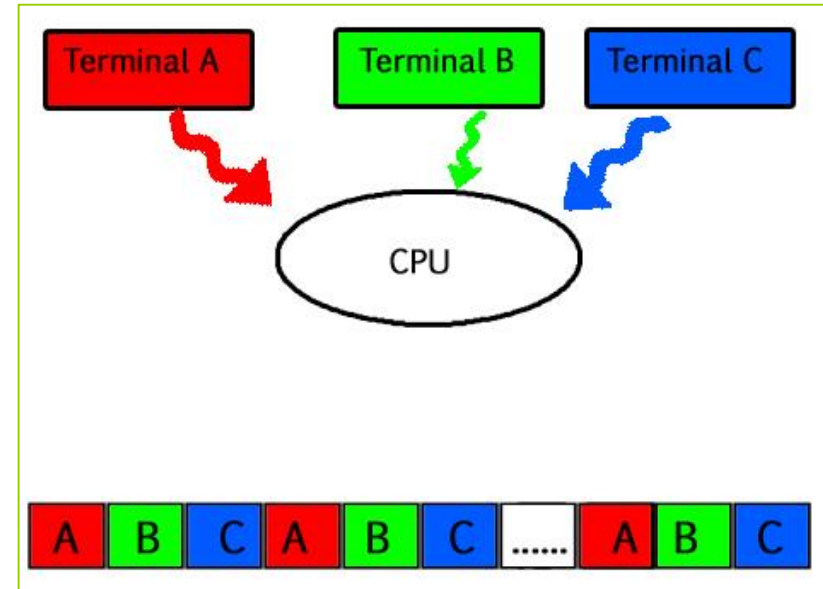
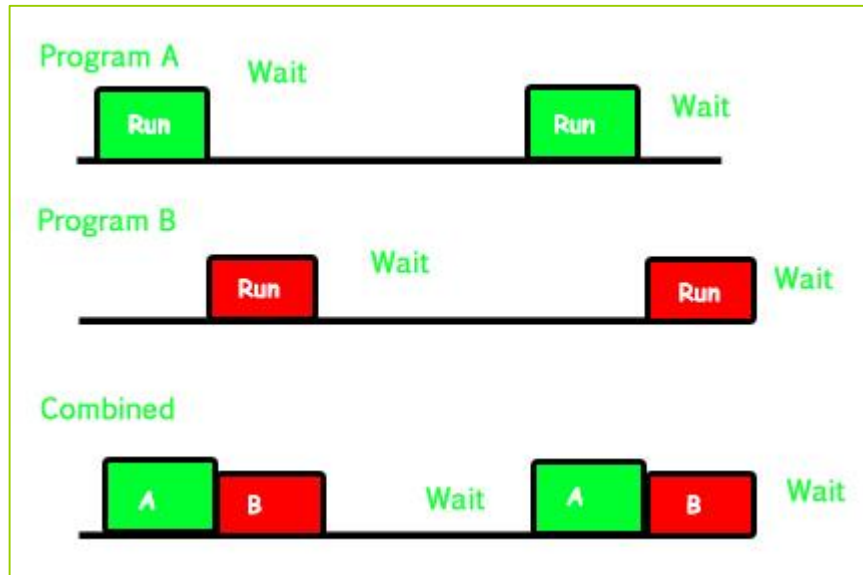
---

- **Multitasking** sistemlerde bir process genellikle tamamlanmadan veya bir I/O'ya ihtiyaç duymadan önce kısa bir süreliğine çalıştırılır, sonra başka bir process yine kısa bir süreliğine çalıştırılır.
- Birden fazla process aynı anda çalışmaya hazırsa, işletim sistemi hangi process'in çalışacağını **CPU Scheduling** ile belirler.



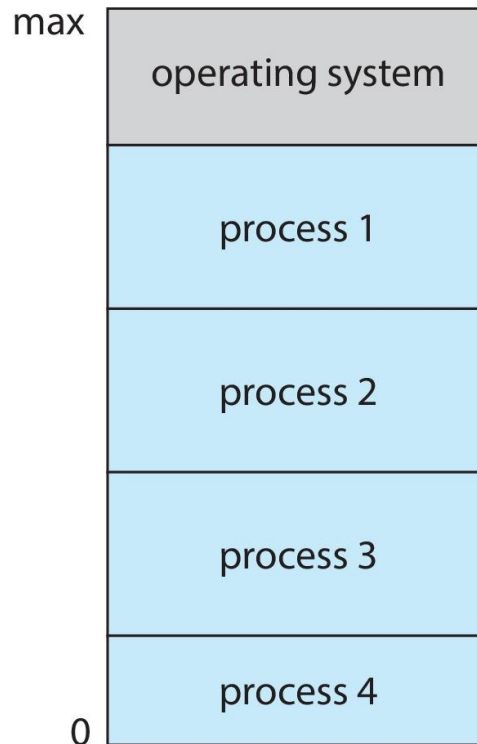


# Multiprogramming vs Multitasking





# Multiprogram Sistemin Bellek Görünümü (Memory Layout for Multiprogrammed System)





# Dual-mode İşlemi

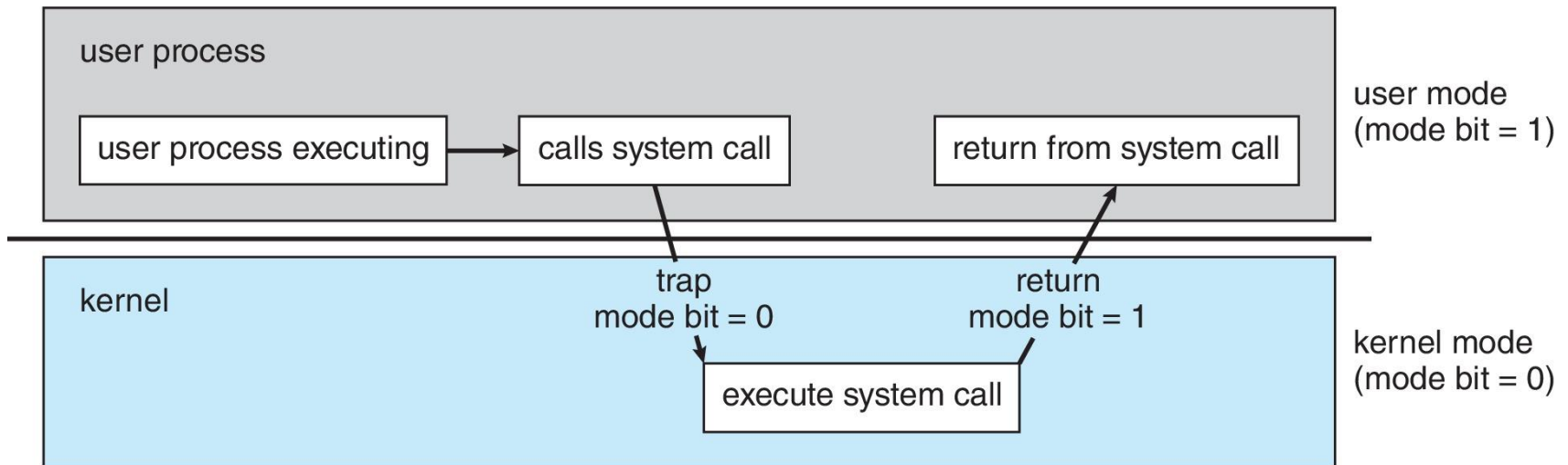
- **Dual-mode** işlemi, işletim sisteminin kendisini ve diğer sistem bileşenlerini korumasına izin verir.
  - **User mode** ve **kernel mode**
- Donanım tarafından sağlanan **Mode bit** ile
  - Sistemin kullanıcı modunu veya kernel modunu ne zaman çalıştırdığını ayırt etmeyi sağlar.
  - Bir kullanıcı kodu yürütüldüğünde  $\Rightarrow$  mode biti “user”
  - Çekirdek kodu yürütüldüğünde  $\Rightarrow$  mode biti “kernel”
- Öncelikli (**privileged**) olarak tasarlanmış bazı komutlar, yalnızca kernel modunda çalıştırılabilir.





# Dual-mode İşlemi

- Kullanıcının mod bitini açıkça "kernel" olarak ayarlamayacağını nasıl garanti ederiz?
  - Kullanıcının kernel modu tarafına göndereceği istekler System call ile yapılır, System call başlatılır, mode kernel yapılır, kernel modunda system call yürütülür, geri dönecek değer kullanıcı tarafında döndürülür, tekrar user moduna geçilir.





# Zamanlayıcı (Timer)

- Timer sonsuz bir döngüye girilmesini ve kaynakların tüketilmesini önlemek için kullanılır.
  - Timer, bir süre sonra bilgisayarı interrupta uğratacak şekilde ayarlanır.
  - Fiziksel saat tarafından azalan bir sayaç bulundurur.
  - İşletim sistemi sayacı ayarlar.
  - Sayaç sıfırlandığında bir interrupt oluşturur.
  - Kontrolü yeniden kazanmak veya ayrılan süreyi aşan programı sonlandırmak için scheduling process ten önce set edilir.





# Process Yönetimi (Process Management)

- Process yürütülmekte olan bir programdır. Sistem için bir iş birimidir. Program ***pasif*** bir varlıktır; process ***aktif*** bir varlıktır.
- Process görevini yerine getirmek için kaynaklara ihtiyaç duyar:
  - CPU, memory, I/O, files
  - Başlatma verileri
- Processin sonlandırıldıktan sonra yeniden kullanılabilir kaynakların process'ten geri alınmasını gerekir.
- Tek iş parçacıklı process (Single-threaded), bir sonraki yürütülecek komutun yerini belirten bir **program counter'e** sahiptir
  - Process tamamlanana kadar komutlar sırayla, birer birer yürütülür.
- Multi-threaded processlerde thread başına bir **program counter** vardır.
- Tipik olarak sistemde birçok processes vardır: bir veya daha fazla CPU üzerinde aynı anda çalışan bazıları kullanıcı processleri, bazıları da işletim sistemi processleri şeklindedir.







# Process Yönetimi Faaliyetleri

## (Process Management Activities)

İşletim sistemi, process yönetimi ile bağlantılı olarak aşağıdaki faaliyetlerden sorumludur:

- Hem kullanıcı hem de sistem processlerinin oluşturulması ve silinmesi
- Processlerin askıya alınması ve devam ettirilmesi
- Process senkronizasyon mekanizmasının sağlanması
- Process iletişim mekanizmasının sağlanması
- Deadlock işleyici (handling) mekanizmasının sağlanması





# Bellek Yönetimi

---

- Bir programı çalıştırmak için komutların tamamı (veya bir kısmı) bellekte olmalıdır.
- Programın ihtiyaç duyduğu verilerin tamamı (veya bir kısmı) bellekte olmalıdır.
- Bellek yönetimi, bellekte neyin ne zaman olacağını belirler.
  - CPU kullanımını ve bilgisayarın kullanıcılara cevap vermesini optimize eder.
- Bellek yönetimin faaliyetleri
  - Hafızanın hangi bölümlerinin kimler tarafından kullanıldığının anlık takip etme
  - Hangi processlerin (veya bunların bölümlerinin) ve verilerin belleğe girip çıkacağına karar verme
  - Bellek alanını gerektiği gibi ayırma/tahsis etme ve bırakma





# Dosya-Sistem Yönetimi

- İşletim sistemi, bilgi depolamanın aynı tipte, mantıksal bir görünümünü sağlar.
  - Depolama aygıtlarının fiziksel özelliklerini soyutlamak için mantıksal bir depolama birimi olan **dosya (file)** tanımlanmıştır.
  - Her ortam bir aygıt tarafından kontrol edilir. (örneğin disk sürücü, bant(sürücü))
    - ▶ Herbirinin kendi karakteristiği vardır, erişim hızı, kapasite, veri aktarım hızı, erişim yöntemi (sequential or random) gibi özellikleri değişkenlik gösterir.





# Dosya-Sistem Yönetimi

---

- Dosyalar genellikle klasörlerde organize edilir.
- Kimin neye erişebileceğini belirlemek için çoğu sistemde erişim kontrolü vardır.
- İşletim sisteminin dosya sistemi faaliyetleri
  - Dosya ve izinleri değiştirmek için yollar sunmak
  - Dosyaları ikincil depoya eşlemek
  - Dosyaları kalıcı depo ortamlarında yedeklemek





# Yığın-Depolama Yönetimi (Mass-Storage Management)

- Diskler belleğe sığmayan veya uzun süre saklanması gereken verileri tutmakta kullanılır. (HDD, flash memory)
- İkincil depolama sıklıkla ve yaygın olarak kullanıldığından verimli kullanılması gerekir.
- Bir bilgisayarın tüm çalışma hızı, ikincil depolama alt sisteminin hızlarına ve bu alt sistemi yöneten algoritmalara bağlı olabilir.
- İşletim sistemi, ikincil depolama yönetimi ile bağlantılı olarak aşağıdaki faaliyetlerden sorumludur.
  - Mounting and unmounting
  - Boş alan yönetimi
  - Depo alanı ayırma
  - Disk planlama
  - Bölmelendirme
  - Koruma





# Caching (Ön belleğe alma)

- Bilgisayarda birçok düzeyde gerçekleştirilen önemli bir prensiptir. (donanım, işletim sistemi, yazılım)
- Kullanımda olan bilgiler geçici olarak daha yavaş depolamadan daha hızlı depolamaya kopyalanır
- Bilginin önbellekte olup olmadığı önce kontrol edilir.
  - Bilgi mevcutsa cache tarafından hızlıca kullanılır.
  - Mevcut değilse veri önce önbelleğe kopyalanır sonra kullanılır.
- Önbellek depolama aygıtlarından oldukça küçüktür.
  - Ön bellek **sınırlı** bir büyüklüğe sahip olduğundan önbellek yönetimi **önemli** bir **tasarım** problemidir.
  - Önbellek büyüklüğünün ve yerdeğiştirme politikasının dikkatli bir şekilde belirlenmesi performansın önemli bir ölçüde artmasını sağlayabilir.





# Çeşitli Depolama Türlerinin Özellikleri

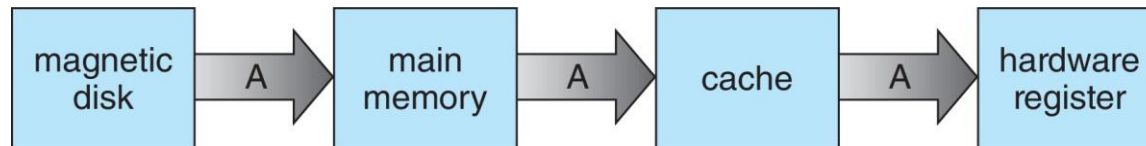
Level	1	2	3	4	5
Name	registers	cache	main memory	solid-state disk	magnetic disk
Typical size	< 1 KB	< 16MB	< 64GB	< 1 TB	< 10 TB
Implementation technology	custom memory with multiple ports CMOS	on-chip or off-chip CMOS SRAM	CMOS SRAM	flash memory	magnetic disk
Access time (ns)	0.25-0.5	0.5-25	80-250	25,000-50,000	5,000,000
Bandwidth (MB/sec)	20,000-100,000	5,000-10,000	1,000-5,000	500	20-150
Managed by	compiler	hardware	operating system	operating system	operating system
Backed by	cache	main memory	disk	disk	disk or tape





# “A” Verisinin Diskten Registera Göçü

- Multitasking ortamlar **en güncel değeri** (değer depolama hiyerarşisinin neresinde olursa olsun) kullanmaya dikkat etmelidir.



- Multiprocessor ortamda cache tutarlılığı(coherency) sağlanmalıdır ki bütün CPU'lar cache'lerinde en güncel değere sahip olsunlar.
- Dağıtık ortamdaki durum daha da karmaşıktır.
  - Bir verinin çok sayıda kopyası bulunabilir.
  - Kitabın 19. bölümünde çeşitli çözümler ele alınmıştır.







# I/O Altsistemi

- OS'in bir amacı donanım cihazlarının kendilerine has özelliklerini kullanıcıdan gizlemektir.
- I/O altsistemi şunlardan sorumludur:
  - Bir verinin çok sayıda kopyası bulunabilir.
  - I/O bellek yönetimi kapsamında:
    - buffering-arabelleğe kaydetme (veri aktarılırken geçici olarak saklama),
    - caching (verinin parçalarını performans için daha hızlı depoda saklama),
    - spooling (bir işin çıktısının başka bir işin girdisiyle örtüşmesi)
  - Genel aygıt sürücü arayüzü
  - Özel donanım aygıtları için sürücüler





# Koruma ve Güvenlik

- **Koruma (Protection)** – Process'lerin veya kullanıcıların işletim sistemi tarafından tanımlanan kaynaklara erişimini kontrol etmek için kullanılan mekanizmalardır.
- **Güvenlik (Security)** – sistemin iç ve dış saldırılara karşı savunulması
  - Çok çeşitli saldırılar: hizmet engelleme (denial-of-service), solucanlar (worms), virüsler, kimlik hırsızlığı (identity theft), hizmet hırsızlığı vs.
- Sistemler öncelikle kullanıcıları ayırır ve kimin ne yapabileceğini belirler.
  - Kullanıcı kimlikleri (user IDs, security IDs) her kullanıcı için bir isim ve numara içerir.
  - User ID, erişim denetimi için o kullanıcının bütün dosya ve process'leriyle ilişkilendirilir.
  - Grup kimliği (group ID) kullanıcı gruplarının tanımlamasına, süreç ve dosyalarla ilişkilendirilerek denetimin yönetilmesine imkan verir.
  - Yetki yükseltme (Privilege escalation) kullanıcının daha fazla hakka sahip ID'ye geçmesini sağlar.





# Sanallaştırma (Virtualization)

---

- Sanallaştırma, tek bir bilgisayarın donanımını, birkaç farklı yürütme ortamına soyutlamamıza izin veren bir teknolojidir.
- Böylece her bir ortamın kendi özel bilgisayarında çalıştığı yanılsaması yaratılır.
- Sanallaştırma, işletim sistemlerinin diğer işletim sistemleri içinde uygulamalar olarak çalışmasını sağlar.
  - Sanallaştırma endüstrisi geniş bir alandır ve gittikçe büyümektedir.





# Sanallaştırma (Virtualization)

- **Emulation**, bilgisayar donanımının yazılımla simüle edilmesidir, kaynak CPU türü hedef CPU türünden farklı olduğunda kullanılır.
- Örneğin, Apple masaüstü ve dizüstü bilgisayarları için IBM Power CPU'dan Intel x86 CPU'ya geçtiğinde, IBM CPU için derlenen uygulamaların Intel CPU üzerinde çalışmasına izin veren "Rosetta" adlı bir emülator özelliği içeriyordu.
  - Genellikle yavaş bir metottur.
  - Bilgisayar dili makine diline derlenerek çevrilmediğinde – Interpretation (yorumlama) kullanılır.
- **Virtualization** –Sanallaştırma ile, belirli bir CPU mimarisi için yerel olarak derlenen bir işletim sistemi, yine o CPU için yerel olan başka bir işletim sistemi içinde çalışır.
  - VMM (Virtual Machine Manager – sanal makine yöneticisi) sanallaştırma servislerini sunar.





# Sanallaştırma (Virtualization)

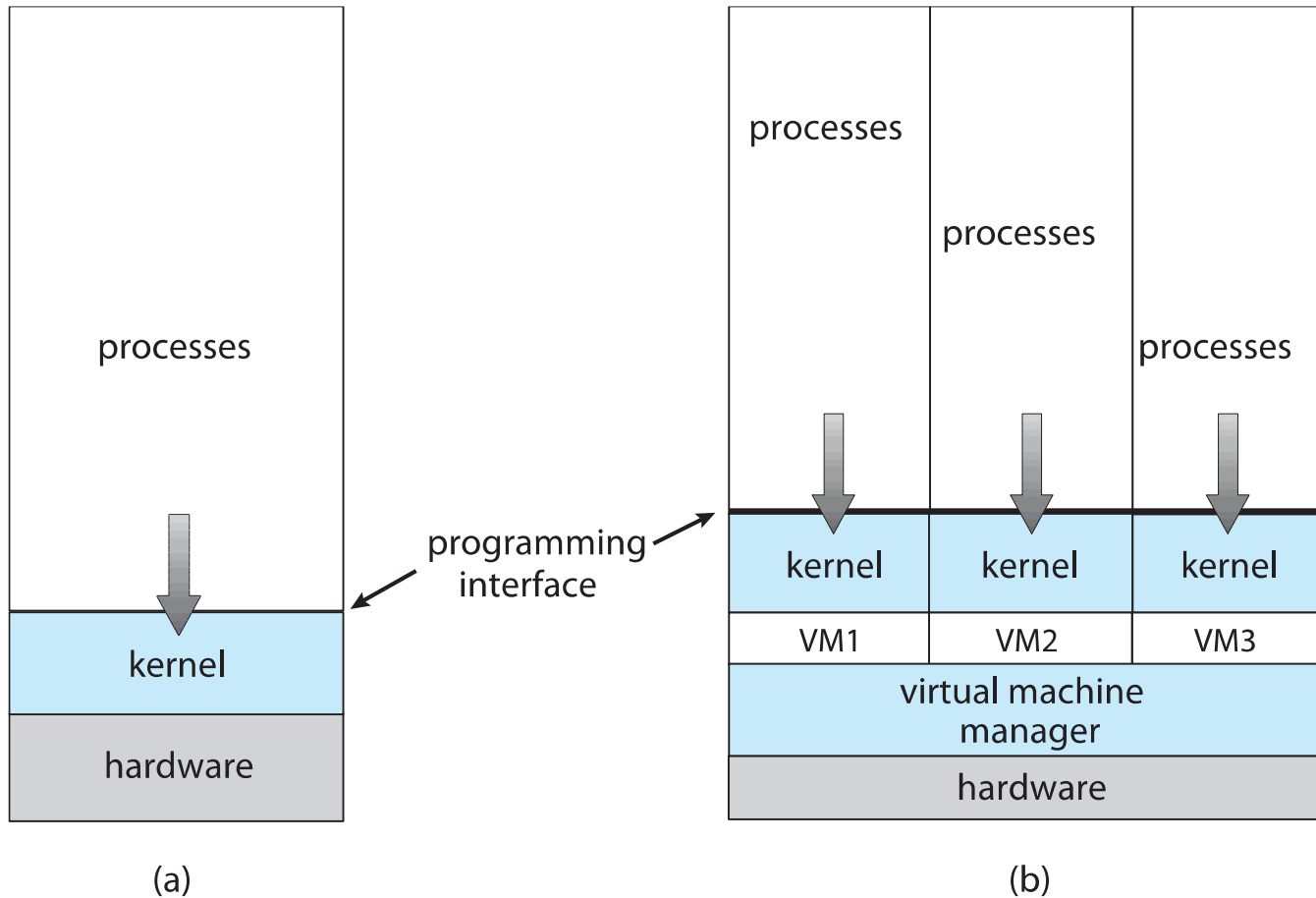
---

- Kullanım alanı olarak araştırma veya uyumluluk çalışmaları için farklı OS'lerin bilgisayarlarda çalıştırılması gösterilebilir.
  - macOS işletim sisteminin çalıştığı bir Apple dizüstü bilgisayarda, Windows uygulamalarının yürütülmesine izin vermek için bir Windows 10 konuk olarak çalıştırabilir.
  - Birden çok sisteme sahip olmadan birden çok işletim sistemi için uygulama geliştirilebilir.
  - Birden fazla sisteme sahip olmadan kalite güvence testi uygulamaları yapılabilir.
  - Veri merkezlerinde hesaplama ortamlarını yürütmek ve yönetmek için kullanılabilir.





# Sanallaştırma (Virtualization)





# Dağıtık Sistemler (Distributed Systems)

- Dağıtık sistem, kullanıcıların çeşitli kaynaklara erişimini sağlamak için bir network ile birbirine bağlı, fiziksel olarak ayrı, muhtemelen heterojen bilgisayarlardan oluşan bir sistemdir.
  - Network bir iletişim yoludur, TCP/IP en yaygın olanıdır
    - ▶ **Local Area Network (LAN)** -bir oda, bina veya kampüs içindeki bilgisayarları birbirine bağlar.
    - ▶ **Wide Area Network (WAN)** -binaları, şehirleri veya ülkeleri birbirine bağlar.
    - ▶ **Metropolitan Area Network (MAN)** - kullanıcıları bir metropol alanı büyüklüğündeki bir coğrafi bölgedeki bilgisayar kaynaklarıyla birbirine bağlayan bir bilgisayar ağıdır.
    - ▶ **Personal Area Network (PAN)** - bireysel bir kişinin çalışma alanındaki elektronik cihazları birbirine bağlamak için bir bilgisayar ağıdır.





# Dağıtık Sistemler

---

- **Network Operating System**, farklı bilgisayarlardaki farklı işlemlerin mesaj alışverişi yapmasına izin veren, ağ üzerinden dosya paylaşımı gibi özellikler sağlayan bir işletim sistemidir.
  - İletişim şeması, sistemlerin mesaj alışverişine izin verir.
  - Tek bir sistemin illüzyonudur.







# Bilgisayar Sistem Mimarisi (Computer System Architecture)





# Bilgisayar Sistem Mimarisi

---

- **Çekirdek(Core)** : CPU'nun temel hesaplama birimidir.
- **CPU**: komutları çalıştıran donanımdır.
- **İşlemci (Processor)**: Bir veya daha fazla CPU içeren fiziksel bir yongadır.
- **Çok çekirdekli (Multicore)**: Aynı CPU üzerinde birden fazla çekirdek içerir.
- **Çok işlemcili (Multiprocessor)**: Birden fazla processor içerir.





# Bilgisayar Sistem Mimarisi

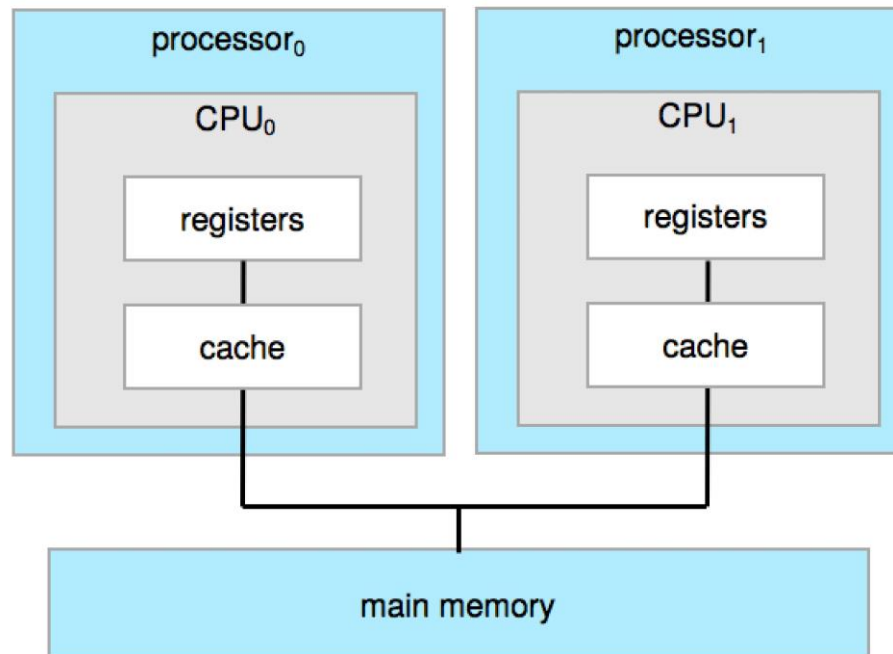
- Çoğu sistem tek bir genel amaçlı processor kullanır
  - Çoğu sistemin özel amaçlı işlemcileri de vardır.
- **Multiprocessors** sistemlerin kullanım ve önemi giderek artmaktadır.
  - **parallel systems, tightly-coupled (sıkı bağlı sistemler)** olarak da bilinirler.
  - Avantajları:
    1. **Increased throughput:** işlemci sayısının artmasıyla, daha kısa sürede daha fazla iş yapılıyor.
    2. **Economy of scale:** kaynak paylaşımı nedeniyle, çok işlemcili sistemler birden çok tek işlemcili sistemlerden daha ucuzdur.
    3. **Increased reliability** – bir işlemcinin arızalanması tüm sistemi durdurmaz
  - İki türüdür:
    1. **Asymmetric Multiprocessing** – her işlemci özel spesifik bir göreve atanır
    2. **Symmetric Multiprocessing** – her işlemci tüm görevlerde kullanılır.





# Symmetric Multiprocessing Architecture

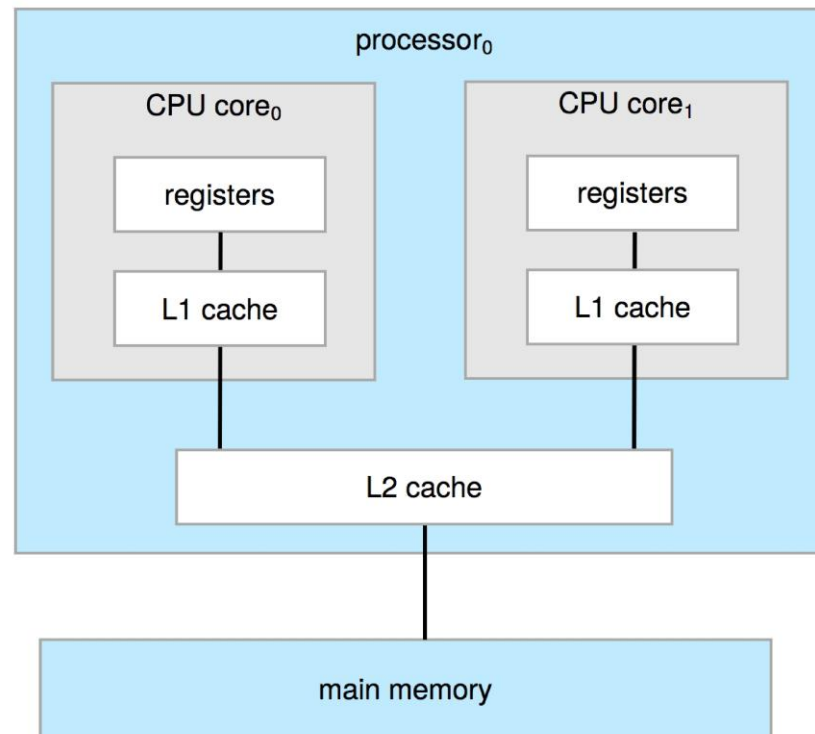
- Şekilde her biri kendi CPU'suna sahip iki işlemcili tipik bir **Symmetric Multiprocessing** mimarisi gösterilmektedir.
- Her CPU işlemcisi kendi register kümesinin yanı sıra lokal ön belleğe sahiptir. Ancak, tüm işlemciler fiziksel belleği sistem veri yolu (bus) üzerinden paylaşır.

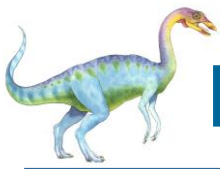




# Dual-Core Design

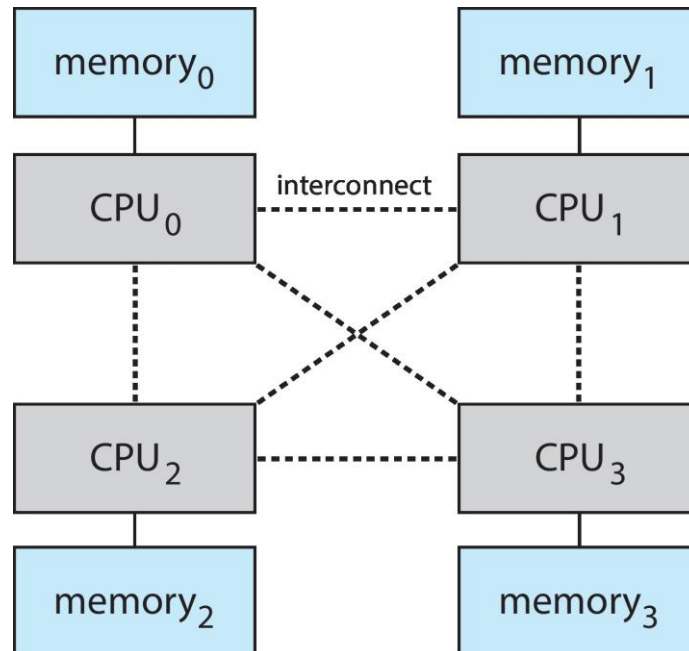
- **multicore** : birden çok çekirdeğinin tek bir yonga üzerinde bulunduğu çok çekirdekli sistemler.
- Birden çok ayrı sistem tek çatı altında toplanır.
- Her core kendi register'larına ve önbelleğine sahiptir, ancak hafızayı paylaşırlar.





# Non-Uniform Memory Access System

- Çok işlemcili bir sisteme ek CPU'lar eklemek, hesaplama gücünü artırır.
- Ancak, sistem çok iyi ölçeklenemediğinde çok fazla CPU eklenmesi sistem veriyolu için çekişme oluşturur, bir darboğaz haline gelir ve performans düşmeye başlar.
- Bunun yerine alternatif bir yaklaşım, her bir CPU'nun kendi yerel belleğinin olduğu yaklaşımdır, CPU küçük, hızlı bir yerel veriyolu aracılığıyla kendi yerel belleğine erişir.





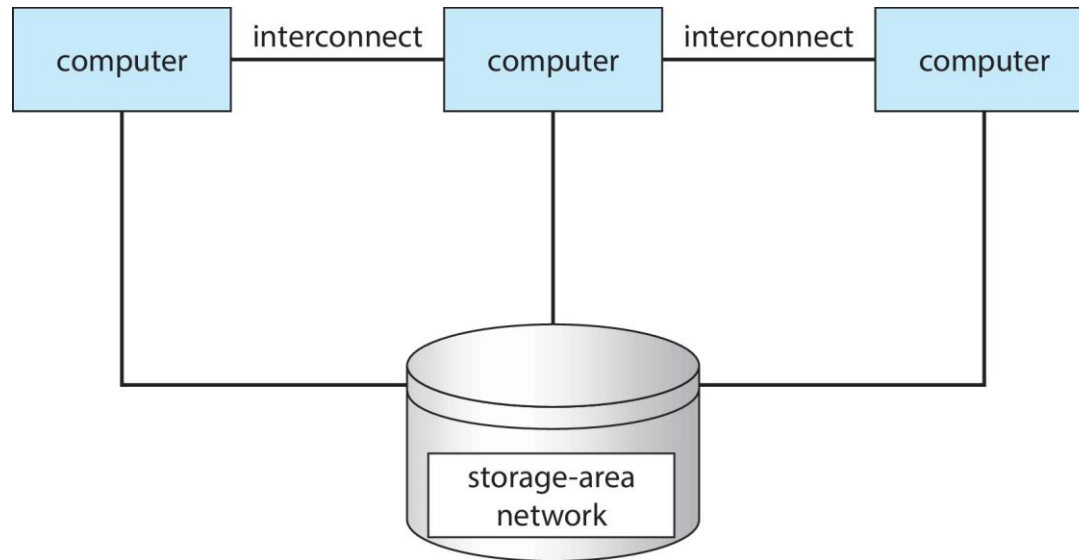
# Küme Sistemleri (Clustered Systems)

- Çok işlemcili sistemlere benzer, ancak birden çok sistem birlikte çalışır
  - Genellikle bir **storage-area network (SAN)** ile depolama paylaşılır.
  - Yüksek kullanılabilirlik (**high-availability**) hizmeti sağlar, bir veya birden fazla sistem başarısız olsa bile hizmet devam eder.
    - ▶ **Asymmetric clustering** bekleme modunda (hot-standby mode) bir makine bulunur.\*
    - ▶ **Symmetric clustering** uygulamaların çalıştırıldığı birden çok node bulunur, bunlar birbirlerini izlerler.
  - Bazı küme sistemleri **high-performance computing (HPC)** için tasarlanmıştır
    - ▶ Uygulamaların paralel çalıştırılabilir (**parallelization**) şekilde yazılması gerekir.
  - Bazı küme sistemleri karışıklıkları önlemek için **distributed lock manager (DLM)** e sahiptirler.





# Küme Sistemleri (Clustered Systems)







# Bilgisayar Sistem Ortamları (Computer System Environments)





# Hesaplama Ortamları

- Bir bilgisayar sistemi, birçok sorunu çözmek için farklı şekillerde düzenlenmiş birçok cihaz kullanır.
- Bu durum birçok bilgisayarın birden çok sorunu ele almak için **hesaplama yaptığı** ve aralarında **bilgi alışverişinde** bulunduğu **hesaplama ortamlarını** (**computing environments**) gerektirir.
- Çeşitleri
  - Geleneksel
  - Mobil
  - İstemci Sunucu (Client Server)
  - Eşten eşe (Peer-to-Peer)
  - Bulut Bilişim (Cloud computing)
  - Gerçek Zamanlı Gömülü Sistemler (Real-time Embedded)





# Hesaplama Ortamları - Geleneksel

---

- Bağımsız, tek çalışan genel amaçlı makinelerdir.
- Bilgisayar ve iletişim teknolojilerinin gelişmesiyle geleneksel hesaplama ortamlarının çoğunu ayıran çizgiler bulanıklaştı, artık birçok sistem birbirine bağlıdır (Internet).
- **Portal'lar** iç sistemlere web erişimi sağlar.
- **Ağ bilgisayarları - Network computers (thin clients)** web terminallerine benzer.
- Mobil bilgisayarlar **wireless networks** ile birbirine bağlanır.
- Networking artık her yerde – ev sistemlerini dahi Internet saldırılarından korumak için **firewall (güvenlik duvarı)** kullanılır.





# Hesaplama Ortamları - Mobil

---

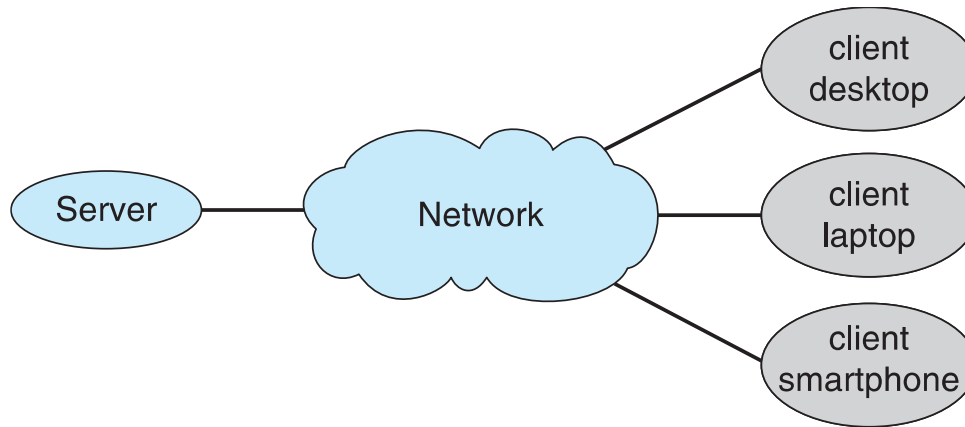
- Akıllı telefonlar, tabletler, vb.
- Mobil cihazlar bir masaüstü veya dizüstü bilgisayarda kullanılamayan veya pratik olmayan işlevselliklerin sağlamasına izin verir. (GPS, gyroscope, accelerometer)
- Yeni uygulama türlerine imkan tanır (örn. ***augmented reality – artırılmış gerçeklik***)
- IEEE 802.11 kablosuz veya hücresel veri ağları ile bağlantı sağlanıyor.
- En önde gelenleri **Apple iOS** ve **Google Android**





# Hesaplama Ortamları – İstemci Sunucu

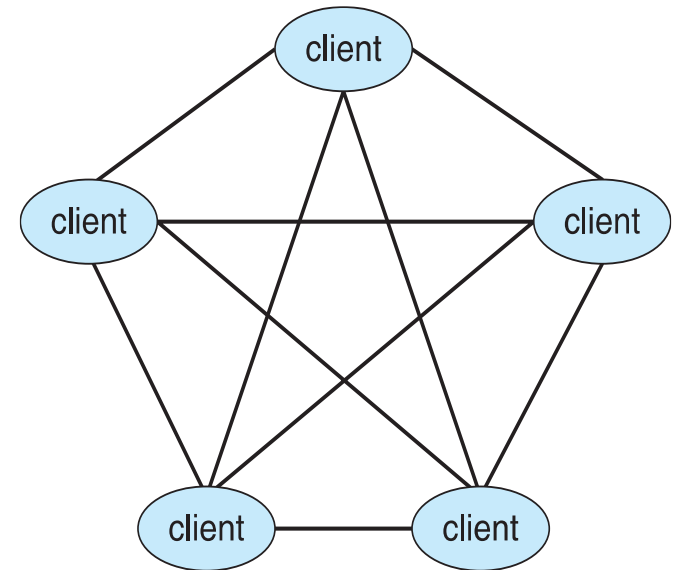
- İstemci-sunucu yapısı
  - Akıllı PC'lerle desteklenen, kullanıcı isteklerinin yapıldığı, yetenekleri kısıtlı terminaller ve
  - Kullanıcılardan gelen isteklere cevap veren sunucular (**servers**) vardır
    - ▶ **Compute-server system** istemciye servis isteyebileceği bir arayüz sunar. (i.e., veritabanı)
    - ▶ **File-server system** istemcilerin dosya saklayıp alabilmesi için bir arayüz sunar.





# Hesaplama Ortamları - Peer-to-Peer

- Bir dağıtık sistem modelidir.
- P2P (peer-to-peer, eşler arası), istemci sunucu ayrımı yapmaz.
  - Bütün düğümler eş kabul edilir.
  - Her biri istemci, sunucu veya ikisi de olabilir.
  - Düğüm P2P ağına katılmalıdır.
    - ▶ Servisini ağdaki merkezi arama servisine kaydettirir, veya
    - ▶ **Keşif protokolü** ile servis isteğini broadcast yapar, servis isteklerine yanıt verir.
  - Örneğin Bittorrent, Napster and Gnutella, **Voice over IP (VoIP)** Skype





# Hesaplama Ortamları – Bulut Bilişim

---

- Hesaplama, depolama, uygulama vb. servisleri ağ üzerinden sunar.
- Sanallaştırmayı temel alarak işlevsellik sunduğu için sanallaştırmanın uzantısı olarak kabul edilebilir.
  - Amazon **EC2**, kullanıma göre ödeme sisteminin olduğu binlerce sunucu, milyonlarca sanal makine ve petabyte'larca depolama alanı sunar.



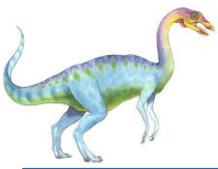


# Bulut Bilişim

- Farklı türleri vardır.
  - **Public cloud** – Internet üzerinden parasını veren herkese açık
  - **Private cloud** – Şirketin kendi kullanımı için yönettiği
  - **Hybrid cloud** – hem açık hem özel bulut bileşenleri içerir.
  - Servis olarak Yazılım (Software as a Service (**SaaS**)) – Internet üzerinden kullanılan uygulamalar (örn. ofis uygulamaları)
  - Servis olarak Platform (Platform as a Service (**PaaS**)) – yazılım geliştirici veya uygulamalar tarafından kullanıma hazır (örn. veritabanı sunucusu)
  - Servis olarak Altyapı (Infrastructure as a Service (**IaaS**)) – Internet üzerinden kullanıma açık sunucu ve depolama alanları.

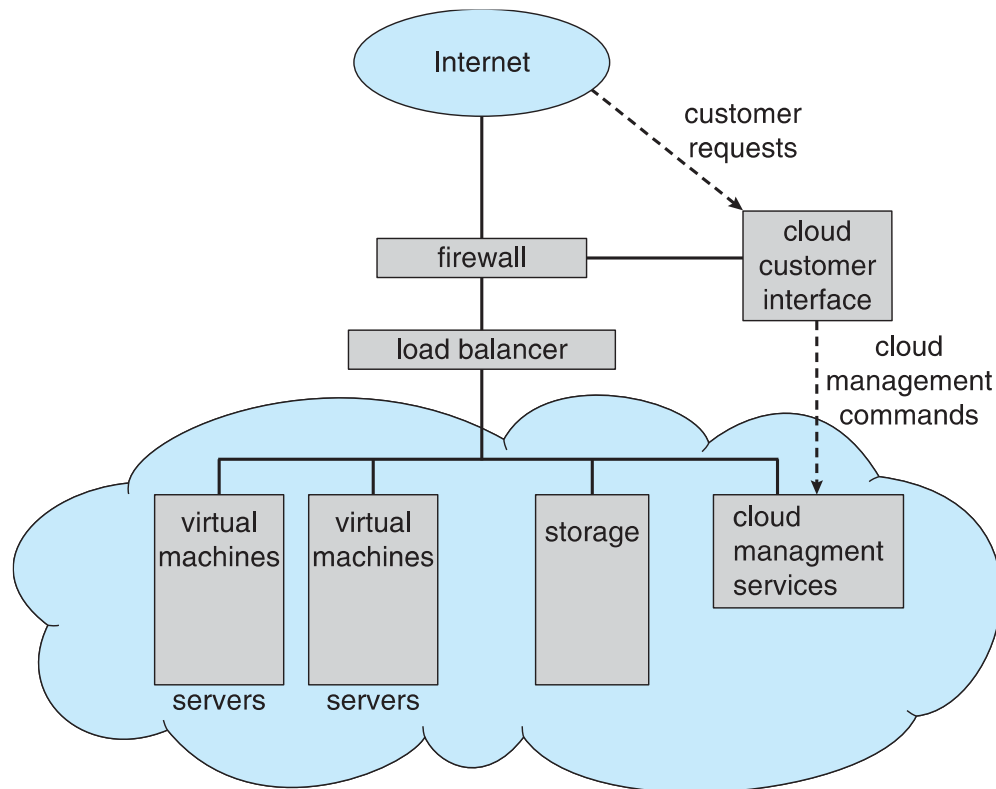






# Bulut Bilişim

- Bulut bilişim ortamları geleneksel işletim sistemleri, VMM'ler ve bulut yönetim araçlarından oluşur
  - Internet bağlantısı firewall gibi güvenlik önlemlerini gerektirir.
  - Yük dengeleyiciler (Load balancers) birden çok uygulama arasında trafiği yayar.





# Gerçek Zamanlı Gömülü Sistemler

- Gerçek zamanlı gömülü sistemler en yaygın bilgisayar biçimidir,
- Gömülü bilgisayarlar araba motorlarından üretim robotlarına, optik sürücülerden mikrodalga fırınlarına kadar her yerde bulunur.
  - Çok çeşitlidir: özel amaçlı gömülü OS'lerin olduğu cihazların yanısıra görevlerini OS olmadan gerçekleştiren uygulamaya özel entegre devrelere sahip donanım cihazları da vardır
- Hem bağımsız birim olarak hem de bu cihazların web'in ögesi olarak kullanılabilmesi gittikçe yaygınlaşacaklardır.
  - Süt bittiğinde markete süt sipariş verebilen akıllı buzdolabı teknolojileri hayata geçmiştir.





# Gerçek Zamanlı Gömülü Sistemler

- Gömülü sistemler neredeyse her zaman gerçek zamanlı işletim sistemlerini (**real-time OS**) çalıştırırlar.
- Gerçek zamanlı sistem, bir işlemcinin bir işi yapmasının veya bir veri akışının gerçekleşmesinin ancak **katı zaman gereksinimleri** doğrultusunda geçerli kabul edildiği durumlarda kullanılır; bu nedenle, özel amaçlı bir uygulamada genellikle bir kontrol cihazı olarak kullanılır.
- Tıbbi görüntüleme sistemleri, endüstriyel kontrol sistemleri, silah sistemleri ve otomobil motoru yakıt enjeksiyon sistemleri gerçek zamanlı sistemlerdir.
- Gerçek zamanlı OS (**real-time OS**) iyi tanımlı sabit zaman kısıtlarına tabidir.
  - İşlem, kısıta uygun olarak yapılmalıdır.
  - Yalnızca zaman kısıtlamaları dahilinde doğru sonucu döndürürse doğru çalışır.





# Özgür Açık-Kaynak İşletim Sistemleri

- Hem özgür işletim sistemleri (**free OS**) hem de açık kaynaklı işletim sistemleri (**open-source OS**) derlenmiş binary formatın yanı sıra kaynak kodu biçiminde de sunulurlar.
- Lisanslı bir eserin izinsiz kopyalanmasını önleme mekanizmaları olan **copy protection** ve **Digital Rights Management (DRM)** hareketlerine karşı oluşturulmuştur.
- Yazılım kullanımını ve yeniden dağıtımını sınırlama hareketine karşı koymak için, Richard Stallman 1984'te GNU adlı özgür, UNIX uyumlu bir işletim sistemi geliştirmeye başlamıştır.
  - GNU isminin açılımı "**G**NU's **N**ot **U**nix!" tir. Bu ismi almasındaki sebep de tasarımının Unix'e benzerken kendisinin özgür yazılım olması ve herhangi bir UNIX kodunu içermemesidir.
  - Stallman'a göre "free" fiyat değil kullanım özgürlüğü anlamına gelir.





# Özgür Açık-Kaynak İşletim Sistemleri

- Özgür yazılımı oluşturanlar özgür yazılımın desteklenmesi ve geliştirilmesi için lisanslar oluşturmuşlardır: **Free Software Foundation (FSF)**, **GNU General Public License (GPL)**
- Özgür yazılım ve açık kaynaklı yazılım, farklı gruplar tarafından desteklenen iki farklı fikirdir:
  - Özgür yazılım yalnızca kaynak kodunu sağlamakla kalmaz, aynı zamanda ücretsiz kullanıma, yeniden dağıtım ve değiştirmeye izin verecek şekilde lisanslanır.
  - Açık kaynaklı yazılımın bu tür bir lisanslama sunmayabilir. Bu nedenle, tüm özgür yazılımlar açık kaynaklı olsa da, bazı açık kaynaklı yazılımlar "ücretsiz" değildir.





# Özgür Açık-Kaynak İşletim Sistemleri

---

- GNU/Linux, bazı dağıtımları ücretsiz, bazıları ise yalnızca açık kaynaklı olan en ünlü açık kaynaklı işletim sistemidir. Windows ise kapalı kaynak bir işletim sistemidir.
- Apple'ın mac OS X işletim sistemi hibrit bir yaklaşım içerir. mac OS X in çekirdeği olan Darwin açık kaynaktır ancak tescilli, kapalı kaynaklı bileşenler de vardır.
  - Darwin, BSD Unix tabanlıdır
- Linux'u bir Windows (veya başka) sisteminde Virtualbox kullanarak çalıştırabilirsiniz.





# GNU-GNU/Linux

---

- 1984'te geliştirmeye başlanan GNU işletim sisteminin resmî çekirdeği GNU Hurd'dür, ancak Hurd henüz bitmemiş, geliştirilmesi aşamasında bazı lisans sorunları yaşanmış ve yapısal değişikliğe gidilmiştir.
- 1991'de Finlandiya'da bir öğrenci olan Linus Torvalds, GNU derleyicilerini ve araçlarını kullanarak ilkel UNIX benzeri bir çekirdek yayınlamış ve katkıda bulunmaları için internet üzerinden insanları davet etmiştir.
- Linux işletim sistemi kısa sürede binlerce programcı tarafından hızlıca geliştirilmeye başlanmış, birçok GNU kullanıcısı Linux çekirdeğine geçiş yapmış böylece Linux çekirdeği, Hurd'ın yerini almıştır.





# GNU-GNU/Linux

- Linux çekirdeğini kullanan kullanıcılar sistemlerine genel olarak "Linux" demektedir, ancak GNU Projesi, "Linux" yerine "GNU/Linux" denmesini önerir. Çünkü Linux, bünyesinde GNU araçlarını barındırmaktadır.
- Birçok GNU yazılımları diğer işletim sistemlerinde de kullanılmıştır (Windows, BSD, Solaris ve Mac OS gibi).
- GNU/Linux un başlıca dağıtımları Red Hat, SUSE, Fedora, Debian, Slackware ve Ubuntu'dur.
- Dağıtımlar; işlev, yardımcı program, kurulu uygulamalar, donanım desteği, kullanıcı arabirimi ve amaç bakımından farklılık gösterir.
- Örneğin, Red Hat Enterprise Linux, büyük ticari kullanıma yöneliktir. PCLinuxOS, sistemin önyükleme diskine kurulmadan bir CD-ROM'dan başlatılabilen ve çalıştırılabilen bir işletim sistemidir.

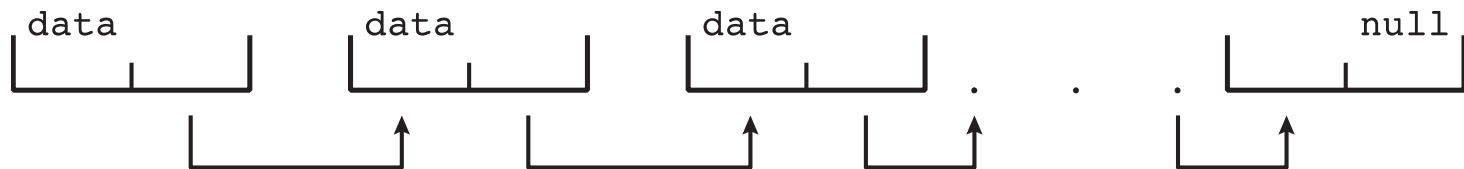




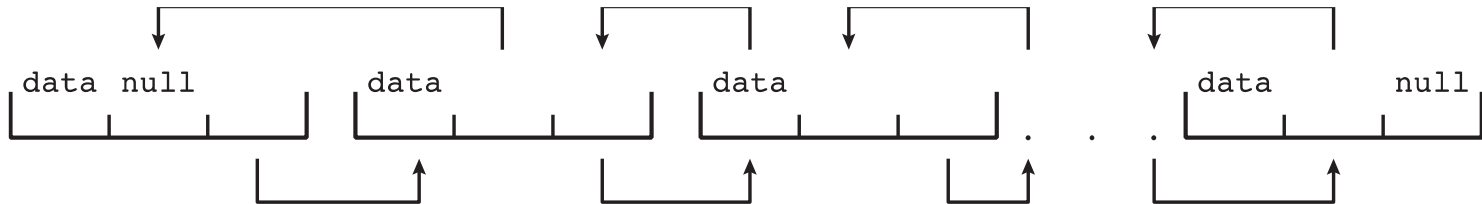


# Kernel Data Structures

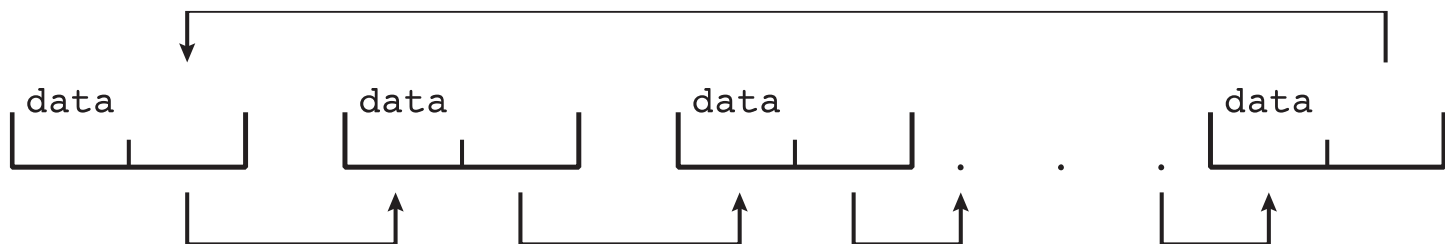
- OS'lerin gerçeştiriminde standart programlamadaki veri yapıları kullanılır.
- ***Singly linked list***



- ***Doubly linked list***



- ***Circular linked list***



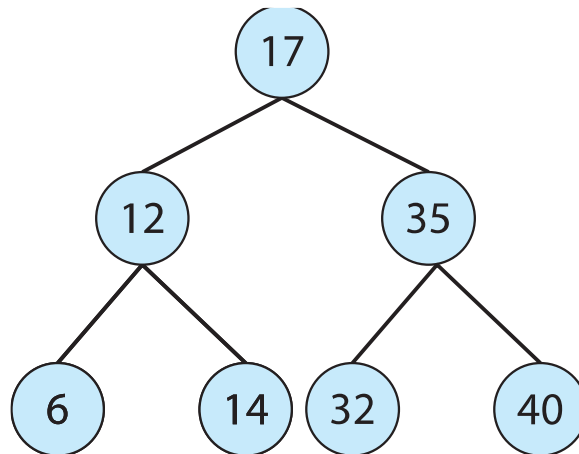


# Kernel Data Structures

## ■ Binary search tree

left  $\leq$  right

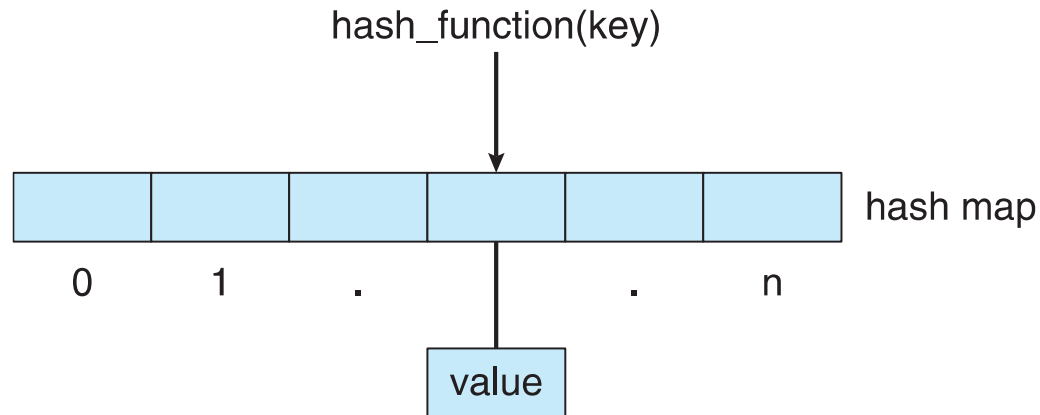
- Search performance is  $O(n)$
- **Balanced binary search tree** is  $O(\lg n)$





# Kernel Data Structures

- **Hash function** ile **hash map** oluşturulur.



- **Bitmap** – n tane elemanın durumu, n uzunluğunda binary değerlerden oluşan bir string içinde tutulur. Sistemdeki kaynakların müsaitlik durumu böyle çok az yer kaplayan bir veri yapısında tutulabilir. Örneğin i. Sıradaki bit değeri 0 ise i. kaynak müsait (kullanılabilir), 1 ise müsait değil.
- Linux'te kullanılan bazı veri yapıları `<linux/list.h>`, `<linux/kfifo.h>`, `<linux/rbtree.h>`





# End of Chapter 1

