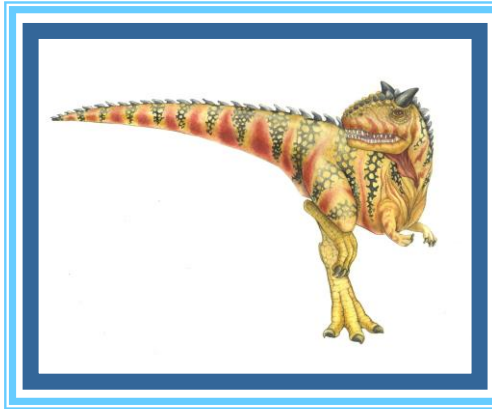


# 1.Hafta: Giriş

---





# İçindekiler

---

- ❑ İşletim sistemi nedir? Ne yapar? -What Operating Systems Do-
- ❑ Bilgisayar sistemi organizasyonu -Computer-System Organization-
- ❑ Bilgisayar sistemi mimarisi -Computer-System Architecture-
- ❑ İşletim sistemi yapısı -Operating-System Structure-
- ❑ İşletim sistemi işlevleri -Operating-System Operations-
- ❑ Proses yönetimi -Process Management-
- ❑ Bellek yönetimi -Memory Management-
- ❑ Depolama yönetimi -Storage Management-
- ❑ Koruma ve güvenlik -Protection and Security-
- ❑ Hesaplama ortamları -Computing Environments-





# Amaçlar (Purposes)

---

- Temel bir bilgisayar sisteminin çalışma mantığını kavramak
- İşletim sistemi bileşenlerini tanımak
- Hesaplama ortamları hakkında genel bilgiye sahip olma
- Açık kaynaklı işletim sisteminlerini keşfetmek





# Bir işletim sistemi nedir?

---

- Bilgisayar kullanıcısı ile bilgisayar donanımı arasında bir aracı gibi hareket eden bir programdır
  
- İşletim sistemi hedefleri (Goals):
  - Kullanıcı programlarını çalıştırmak ve kullanıcı problemlerini çözmek, (Solve user prb.)
  - Bilgisayar sisteminin rahatlıkla kullanılabilmesini sağlamak (Use conveniently)
  - Bilgisayarın donanımı verimli bir şekilde kullanmasını sağlamak (Efficiency)

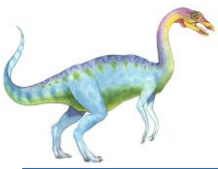




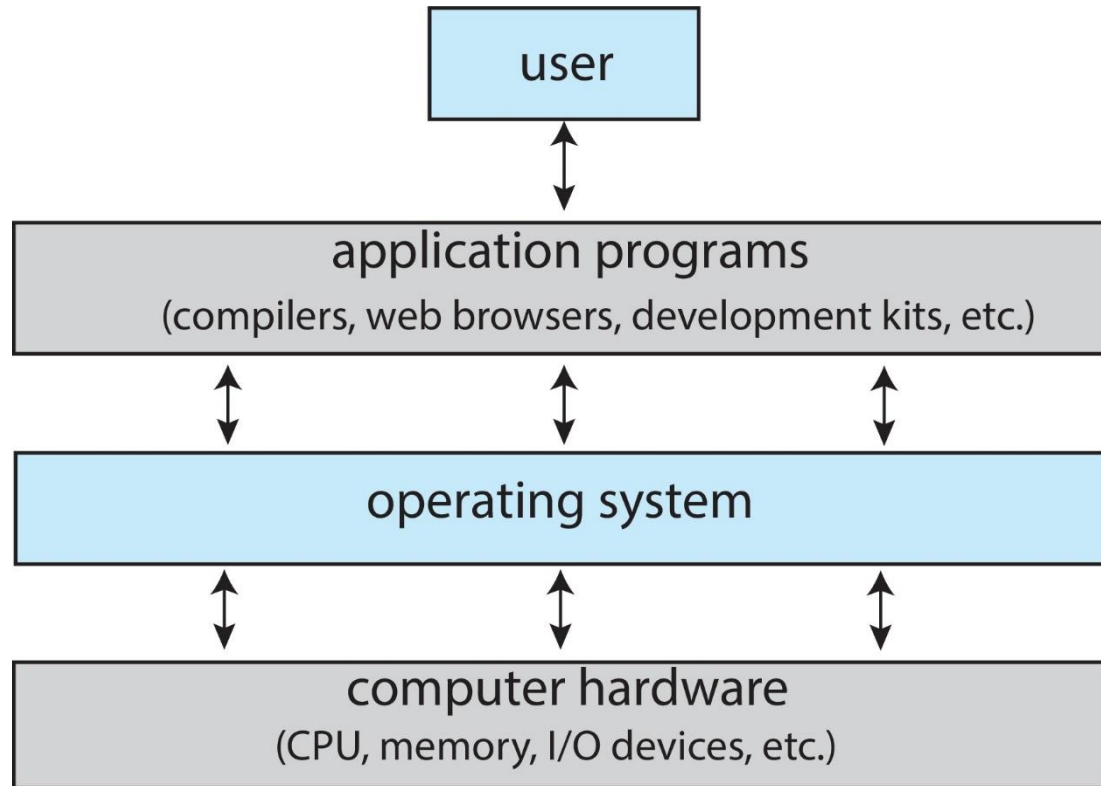
# Bilgisayar sistemi yapısı

- Bir bilgisayar sistemi dört ana bileşene bölünebilir (Computer Components)
  - Donanım – temel hesap kaynakları (Hardware)
    - ▶ MİB, bellek, giriş/çıkış birimleri
  - İşletim sistemi (OS)
    - ▶ Çeşitli uygulamalar ve kullanıcı arasında donanımın kullanımını kontrol ve koordine eder.
  - Uygulama programları (App) – sistem kaynaklarının kullanıcının isteklerini yerine getirmek için kullanılmasını sağlar
    - ▶ Kelime işlemciler, derleyiciler, tarayıcılar, veri tabanı sistemleri, oyunlar, vs.
  - Kullanıcılar (Users)
    - ▶ İnsanlar, makinalar, diğer bilgisayarlar





# Bir bilgisayar sisteminin bileşenleri





# İşletim sistemi ne yapar?

- ❑ Bakış açısına göre değişir
- ❑ Kullanıcıyı kolay kullanım ve iyi performans ilgilendirir (Kaynak kullanımına bakmaz)
- ❑ Paylaşılan bilgisayarlar(mainframe & minicomputer ) bütün kullanıcılarını memnun etmeli
- ❑ Bir göreve tahsis edilmiş bilgisayarların işletim sistemi görevlerine odaklanmalı (Workstation)
- ❑ Elle tutulur sistemler (Cep telf., tablet) kaynakları kısıtlıdır, uzun kullanım için optimize edilmişlerdir.
- ❑ Bazı bilgisayarların arayüzleri yoktur veya ufaktır (Aygıtlardaki gömülü bilgisayarlar örn: arabalardaki gömülü mini sist. )





# İşletim sistemi tanımı

---

- ❑ İşletim sistemi bir **kaynak yöneticisidir**.

(OS is a **resource allocator**)

- ❑ Tüm kaynakları organize eder.
- ❑ Verimli ve adil bir kaynak kullanımı için kaynaklara yapılan taleplerin çakışmaları durumunda karar verir.

- ❑ İşletim sistemi bir kontrol programıdır

(OS is a **control program**)

- ❑ Hataları ve bilgisayarın uygun olmayan kullanımını önlemek için programların çalışmasını denetler

- ❑ Sonuç olarak, bir yönetici olmazsa sistem bozulur 😊





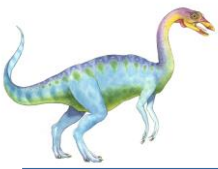


# İşletim sistemi tanımı (OS Definition )

---

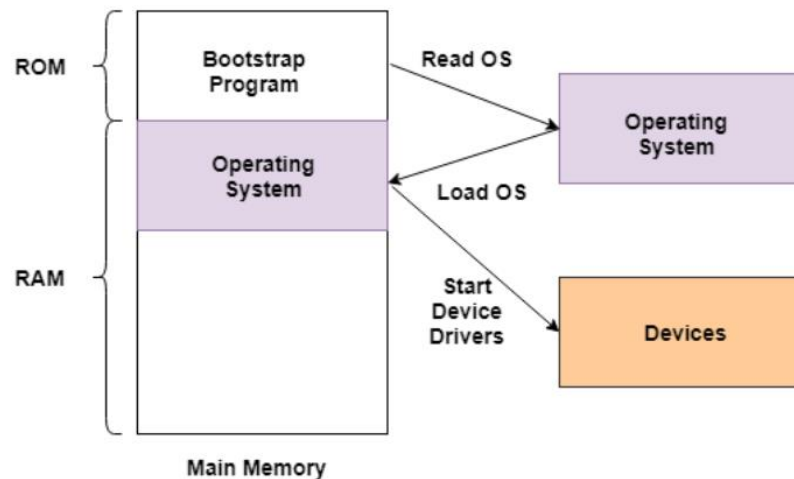
- Evrensel bir tanımı yok.
- ☺Sipariş ettiğinizde herşeyi getirebilen bir satıcı gibi
- “Bilgisayarda sürekli çalışan bir program”
- Program **geliştirme** ve çalıştırma ortamı sunar.
- Donanımın kullanımı zor ve anlaşılamaz detaylarını gizler

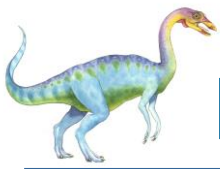




# Bilgisayar açılışı

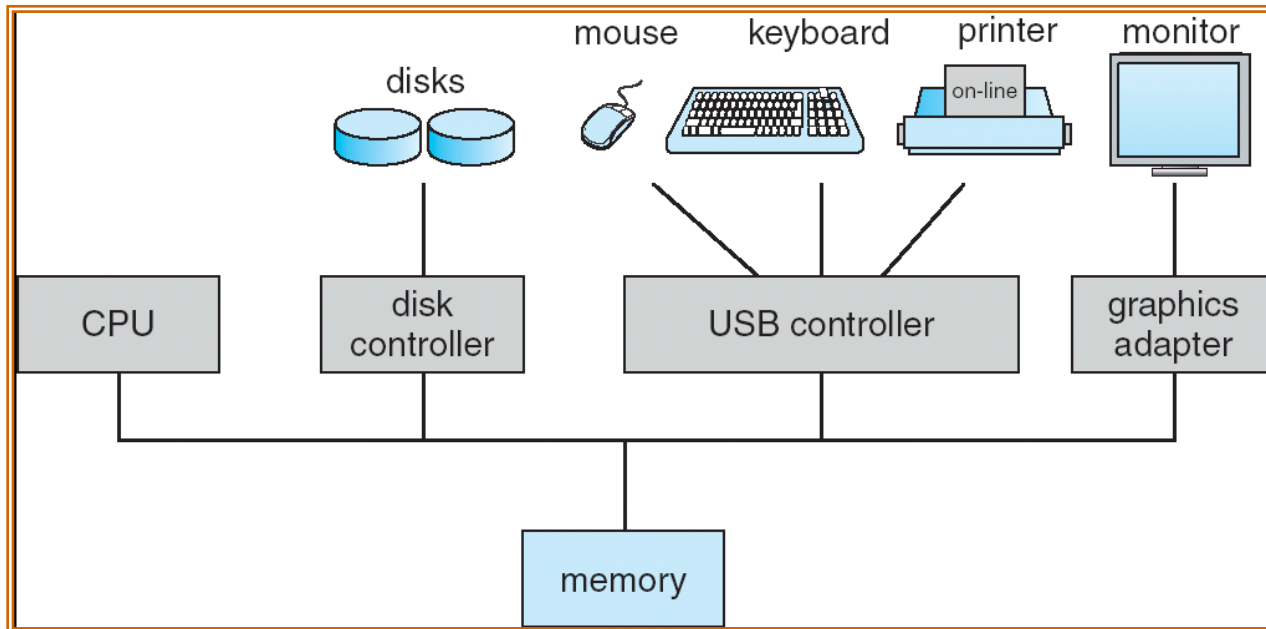
- Bilgisayarı açarken veya yeniden başlatırken bir **önyükleme programı (bootstrap program)** yüklenir
  - Genellikle ROM veya EPROM içinde tutulur ve **donanım yazılımı** –firmware olarak adlandırılır
  - Sistemin bütün bileşenlerini kurar
  - İşletim sistemi çekirdeğini yükler ve çalıştırır.





# Bilgisayar sistemi organizasyonu

- Bilgisayar sistemi çalışması
  - Bir veya daha fazla işlemci, aygıt yöneticileri **ortak bir veri yolu** üzerinden belleğe bağlanır
  - İşlemcilerin ve aygıtların eş zamanlı çalışması bellek çevrimleri için **yarışma/çekişme** sonucunu doğurur.





# Bilgisayar sistemi çalışması

- Giriş/çıkış aygıtları ve işlemci eş zamanlı olarak çalışabilir
- Her aygıta ait bir yönetici bulunur.
- Her aygıt yöneticisi özel bir tampon belleğe sahiptir.
- İşlemci veriyi ana bellekten tampon belleklere veya tersi yönde hareket ettirir.
- Giriş/çıkış, aygıt ile yöneticinin tampon belleği arasında gerçekleşir.
- Aygıt yöneticileri, *kesme – interrupt sonucu oluşan* işlemlerini tamamladıktan sonra işlemciyi haberdar ederler.





# Kesmelerdeki genel fonksiyonlar

---

- ❑ İşletim sistemi kesme tabanlı çalışır.
- ❑ Kesme mimarisi kesilen komutun adresini saklamalıdır.
- ❑ Olumsuz durumları önlemek için bir kesme işlenirken diğerleri geldiğinde bekletilir.
- ❑ Kesme, kontrolü **kesme vektörü** (tüm servis rutinlerinin adreslerini barındırır) içinden **kesme hizmet rutinine** devreder
- ❑ Yakalama (Trap: breakpoint, 0'a bölme, geçersiz bellek) veya istisna (Exception) yazılımın ürettiği kesmelerdir





# Intel Pentium processor event-vector table

vector number	description
0	divide error
1	debug exception
2	null interrupt
3	breakpoint
4	INTO-detected overflow
5	bound range exception
6	invalid opcode
7	device not available
8	double fault
9	coprocessor segment overrun (reserved)
10	invalid task state segment
11	segment not present
12	stack fault
13	general protection
14	page fault
15	(Intel reserved, do not use)
16	floating-point error
17	alignment check
18	machine check
19–31	(Intel reserved, do not use)
32–255	maskable interrupts





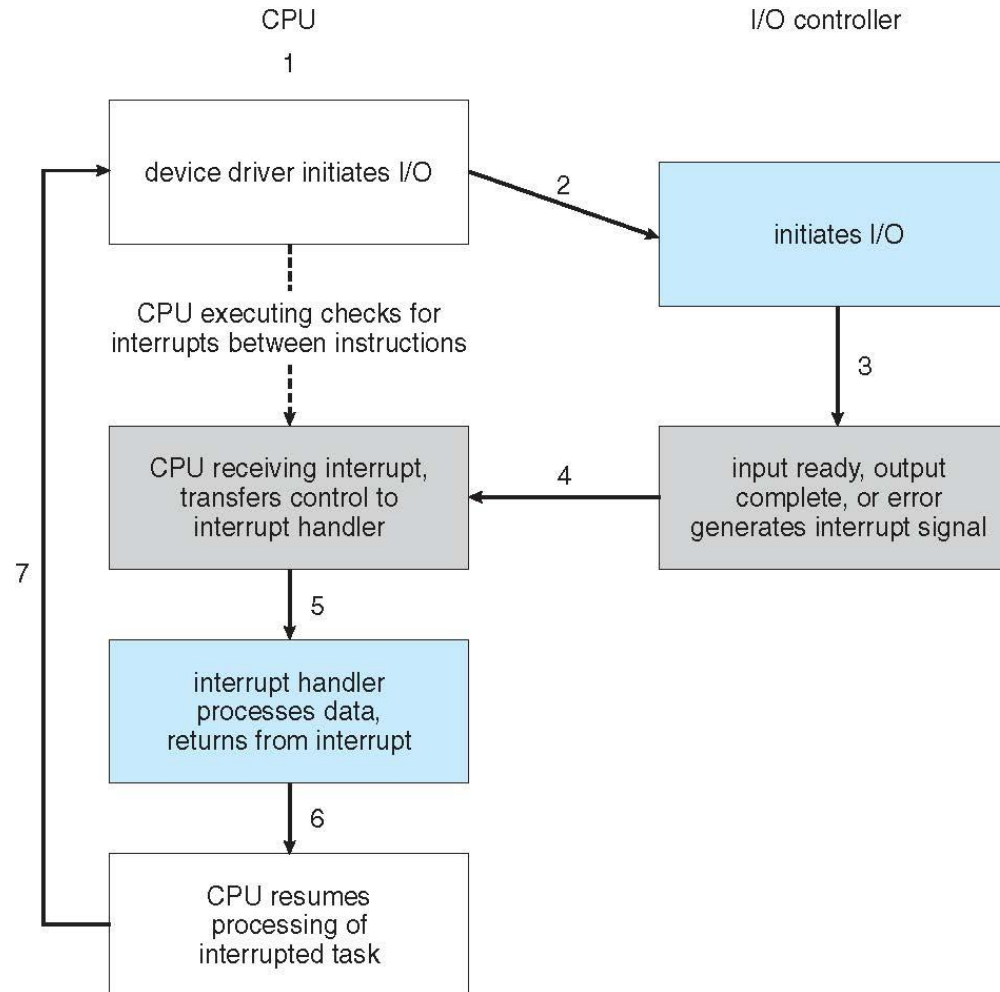
# Kesme kullanımı

- İşletim sistemi kaydediciler ve program sayacı yardımıyla İşlemcinin durumunu saklar
- Ne tür bir kesme meydana geldiğini tanımlar:
  - Yoklama, sorgu yoluyla (Polling). Kesme denetleyicisinin, isteği yapan aygıtı belirlemek için her aygıtın yoklanması (bir sinyal gönderilmesi) gerekir.
  - Kesme Vektörü tarafından. Kesme sinyalini gönderen cihazın kimliğini içeren bir kesme sinyali.
- Ayrı kod bölümleri, her bir kesme türü için hangi eylemin yapılması gerektiğini belirler





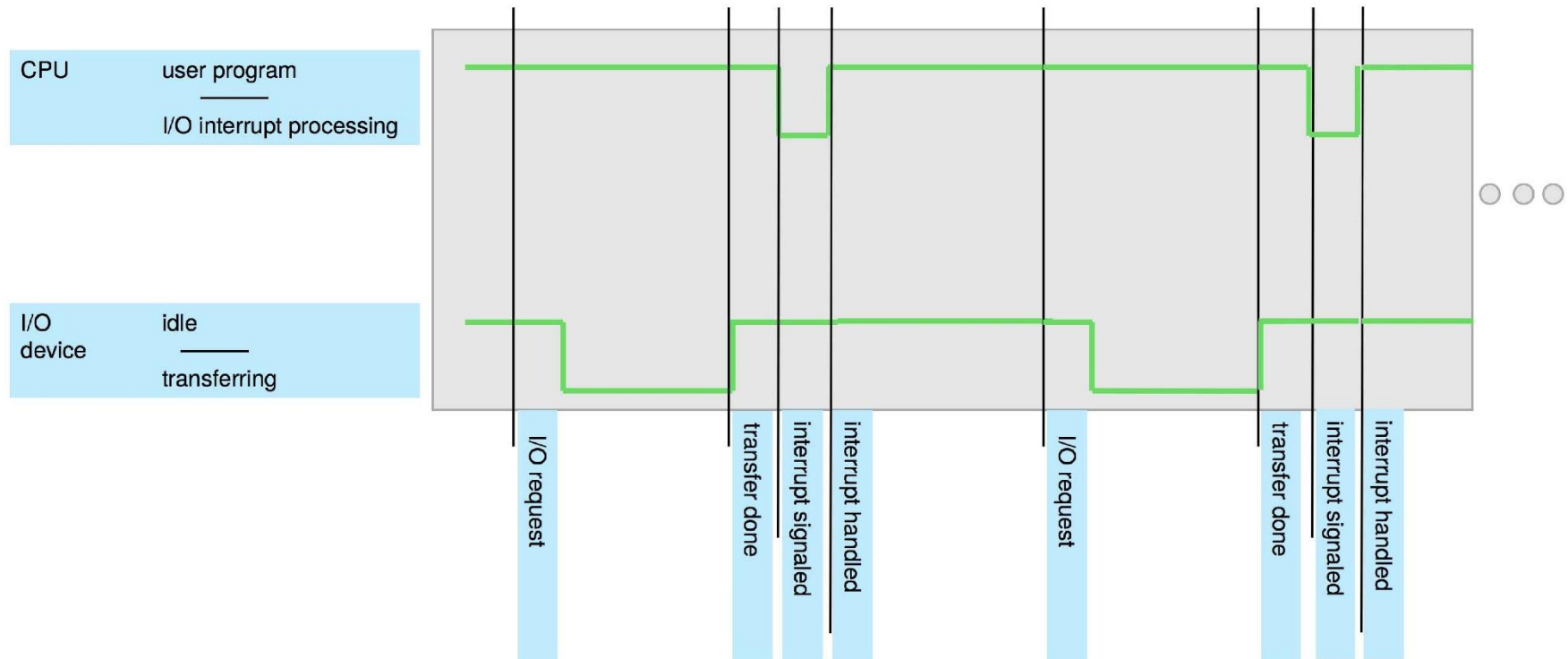
# Interrupt-driven I/O cycle.







# Kesme zaman çizelgesi

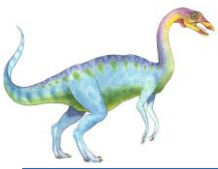




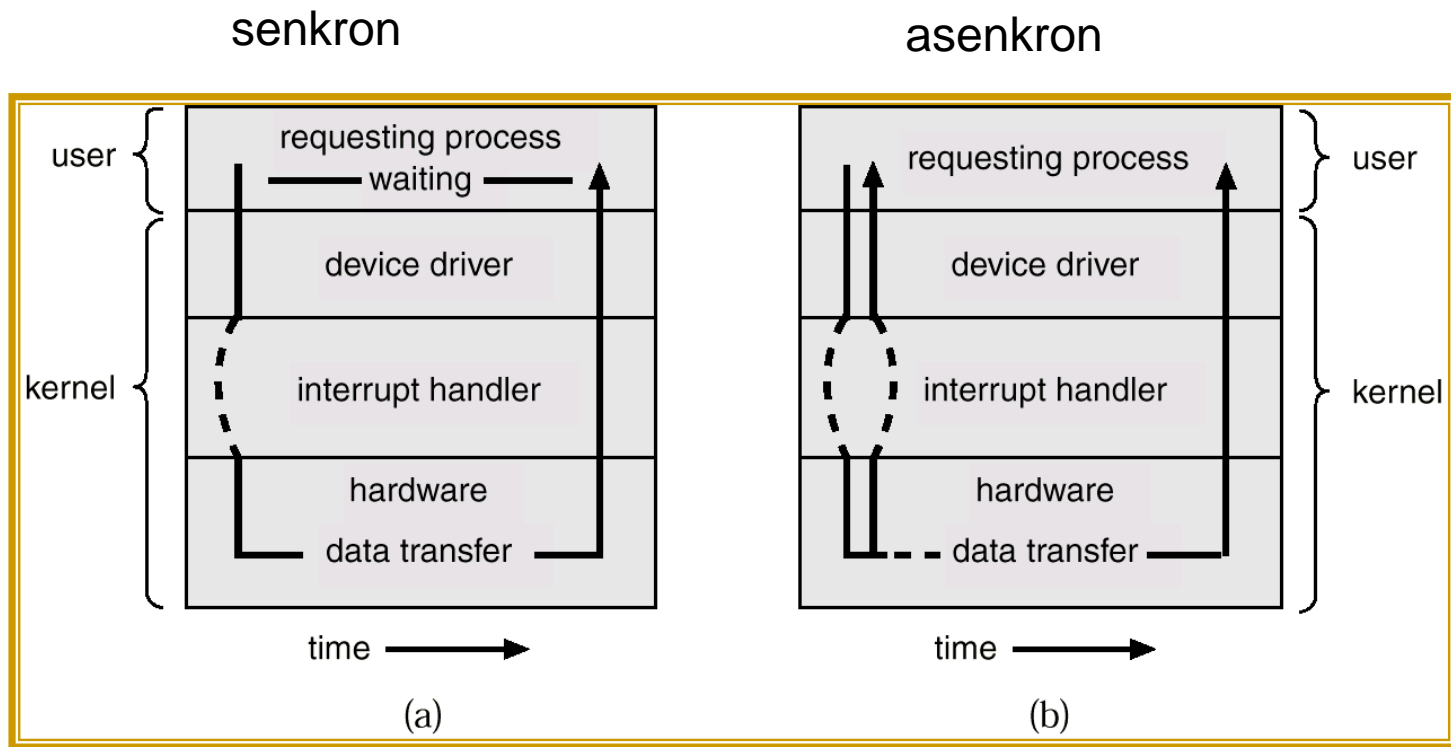
# Giriş/çıkış yapısı

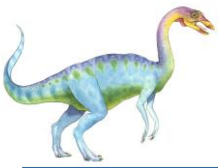
1. Giriş / çıkış işlemi başladıktan sonra, işlem tamamlanmadan kontrol kullanıcı programına aktarılmaz;
  - Bekle komutu bir sonraki kesmeye kadar işlemciyi boşa bekletir
  - Bekleme döngüsü (bellek erişimi için çekişme)
  - Herhangi bir anda en fazla bir giriş/çıkış isteği olur, eş zamanlı giriş/çıkış mümkün değildir. (Bir yoldan erişim olduğu için)
2. Giriş / çıkış işlemi başladıktan sonra, işlemin tamamlanmasını beklemeden kontrol kullanıcı programına geçebilir;
  - **Sistem çağırısı** – kullanıcının giriş/çıkış işlemini tamamlamayı beklemesini sağlamak için işletim sistemine yapılan bir istek
  - **Aygıt–durum tablosu** her bir giriş/çıkış aygıtının tipini, adresini ve durumunu tutar.
  - İşletim sistemi aygıt durumunu belirlemek ve değiştirmek için **tabloyu** indisler.



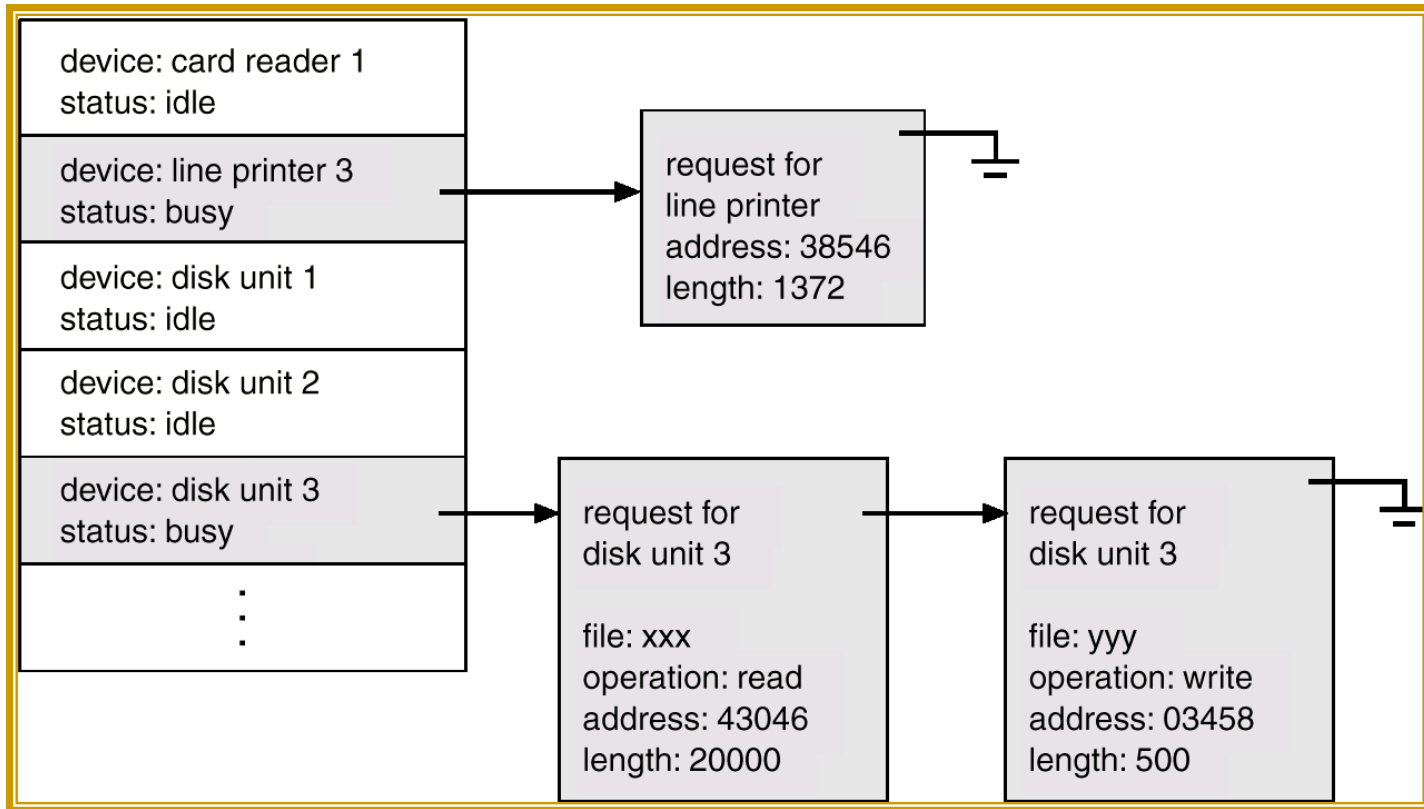


# İki farklı giriş/çıkış yöntemi





# Aygıt-durum tablosu





# Storage Structure

- Ana bellek - CPU'nun doğrudan erişebileceği tek büyük depolama ortamı
  - **Rasgele erişim / Random access**
  - Genellikle uçucu / **volatile**
- İkincil depolama - büyük kalıcı depolama kapasitesi sağlayan ana belleğin genişletilmesi / **nonvolatile** storage capacity
  - Sabit diskler - manyetik kayıt malzemesiyle kaplı sert metal veya cam plakalar
    - ▶ Disk yüzeyi mantıksal olarak bölümlere ayrılmış olan parçalara bölünmüştür
    - ▶ Disk denetleyicisi, aygıtla bilgisayar arasındaki mantıksal etkileşimi sağlar.
  - **Solid-state disks** – sabit disklerden daha hızlı, kalıcı
    - ▶ Çeşitli teknolojiler
    - ▶ Daha popüler





# Storage Definition

---

- Temel bilgisayar depolama birimi **bittir**. Bir bit iki değerden birini içerebilir, 0 ve 1. Bir bilgisayardaki diğer tüm depolama bit koleksiyonlarına dayanır.
- Bir **bayt** 8 bittir ve çoğu bilgisayarda bu en küçük depolama alanıdır.
- Daha az yaygın bir terim, verilen bir bilgisayar mimarisinin yerel veri birimi olan **kelime**dir (word). Bir kelime bir veya daha fazla byte'tan oluşur.





# Storage Definition (Cont.)

- Bilgisayar depolama, çoğu bilgisayar verimi ile birlikte genellikle bayt ve bayt koleksiyonlarında ölçülür ve işlenir.
  - A **kilobyte**, or **KB**, is  $1,024$  bytes
  - a **megabyte**, or **MB**, is  $1,024^2$  bytes
  - a **gigabyte**, or **GB**, is  $1,024^3$  bytes
  - a **terabyte**, or **TB**, is  $1,024^4$  bytes
  - a **petabyte**, or **PB**, is  $1,024^5$  bytes
  - exabyte, zettabyte, yottabyte
- Bilgisayar üreticileri genellikle bu sayıları toplar ve bir megabaytın 1 milyon bayt ve bir gigabaytın 1 milyar bayt olduğunu söyler. Ağ ölçümleri bu genel kural için bir istisnadır; bunlar bit cinsinden verilmiştir





# Depolama yapısı

---

- ❑ Ana bellek –işlemcinin doğrudan erişebildiği tek büyük kayıt ortamıdır.(RAM-geçici)
- ❑ İkincil bellek- daha büyük (geçici olmayan) bir kayıt ortamı sağlayan yardımcı bir bellektir.
- ❑ Manyetik diskler – manyetik kayıt malzemesi ile kaplı katı metal veya cam diskler
- ❑ Solid-state disks, SSD





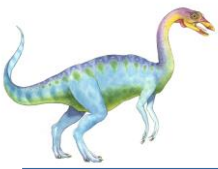


# Depolama hiyerarşisi

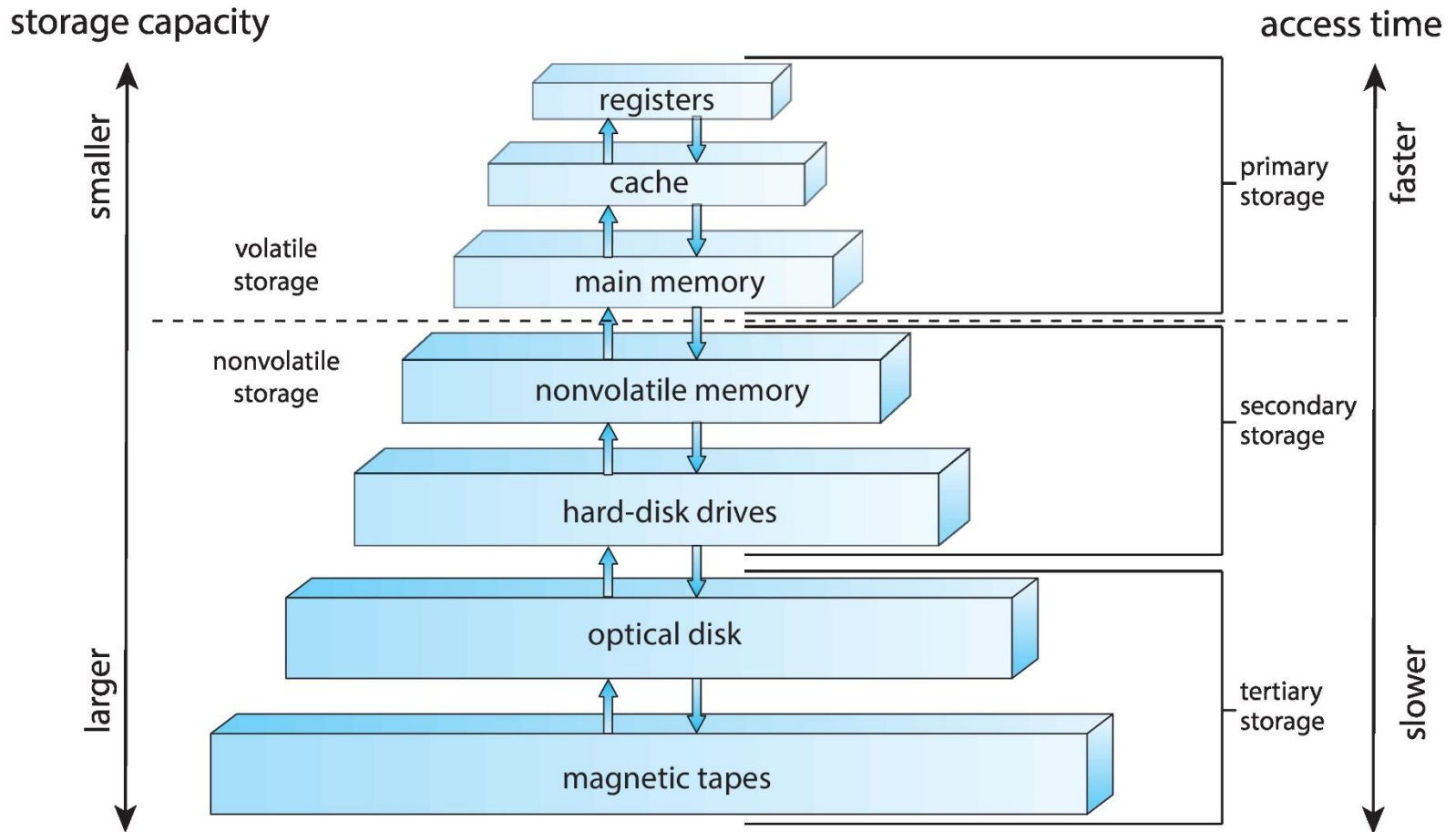
---

- Depolama sistemleri aşağıdaki kriterlere göre hiyerarşik bir yapıda organize edilirler:
  - Hız
  - Maliyet
  - Geçici olma durumuna göre
- *Önbellek*– bilgiyi daha hızlı bir kayıt ortamına kopyalamak; ana bellek hard disk için son ön bellektir.
- Aygıt Sürücüsü: G/Ç'yi yönetmek için her aygıtta özgü denetleyici
  - Denetleyici ve çekirdek arasında düzgün bir arabirim/arayüz sağlar





# Depolama hiyerarşisi





# Önbellek

- ❑ Çok önemli bir prensip
- ❑ Bilgisayarda pek çok alanda kullanılır (donanım, işletim sistemi, yazılım)
- ❑ Daha yavaştan daha hızlı kayıt ortamına veri kopyalanırken kullanılır
- ❑ Veri kopyalanmaya başlamadan önce daha hızlı kayıt ortamı (önbellek) kontrol edilir
  - ❑ Eğer önbellekte kopyalanacak veri bulunuyorsa, doğrudan hızlı bir şekilde oradan kullanılır.
  - ❑ Eğer yoksa, veri önbelleğe kopyalanır ve oradan kullanılır.
- ❑ Önbellek önbelleğe alınacak veriden daha küçüktür.
  - ❑ Önbellek yönetimi önemli bir tasarım problemidir
  - ❑ Önbellek boyutu ve maliyet





# Doğrudan bellek erişimi (DMA) yapısı

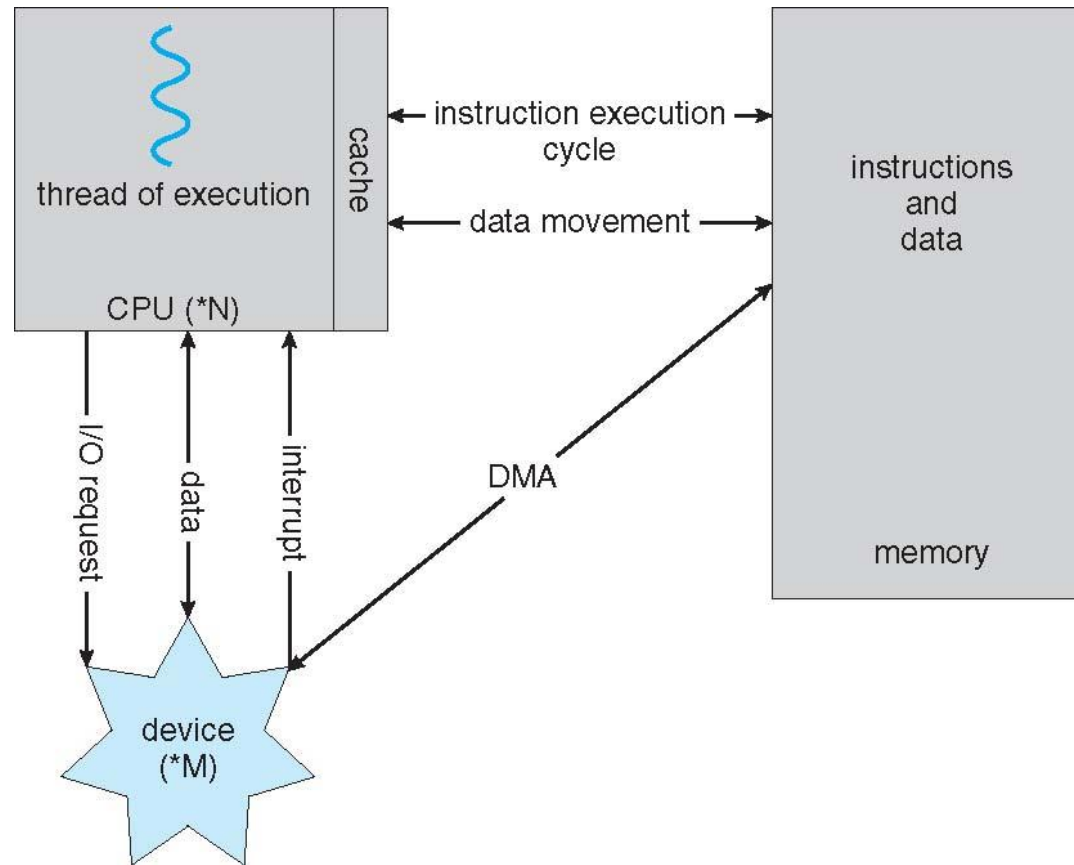
---

- ❑ Veriyi bellek hızına yakın bir hızda iletebilmek için yüksek hızlı giriş/çıkış aygıtları tarafından kullanılır
- ❑ Aygıt yöneticisi veri bloklarını tampondan doğrudan ana belleğe herhangi bir MİB müdahalesi olmaksızın gönderir.
- ❑ Kesme, Bayt başına değil blok başına bir adet üretilir.





# Modern Bilgisayarlar nasıl çalışır?



*A von Neumann architecture*





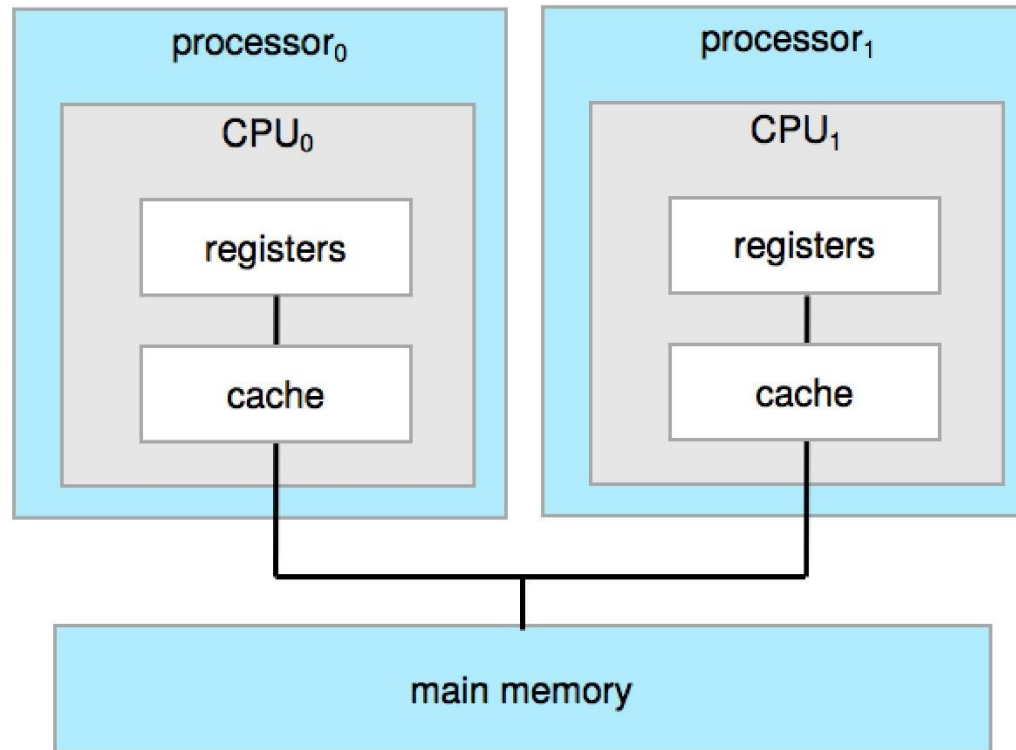
# Bilgisayar-Sistem Mimarisi

- Eskiden çoğu sistem tek bir genel amaçlı işlemci kullanılırdı. Fakat artık çok işlemcili sistemlere geçiliyor;
  - Avantajları;
    - ▶ Artan iş hacmi / **throughput**
    - ▶ Ekonomik
    - ▶ Artan güvenilirlik - bozulma veya hata az
- 2 tip:
  1. **Asimetrik Multiprocessing** – Her işlemciye bir özel görev atanır.
  2. **Simetrik Multiprocessing** – her işlemci tüm görevleri yerine getirir





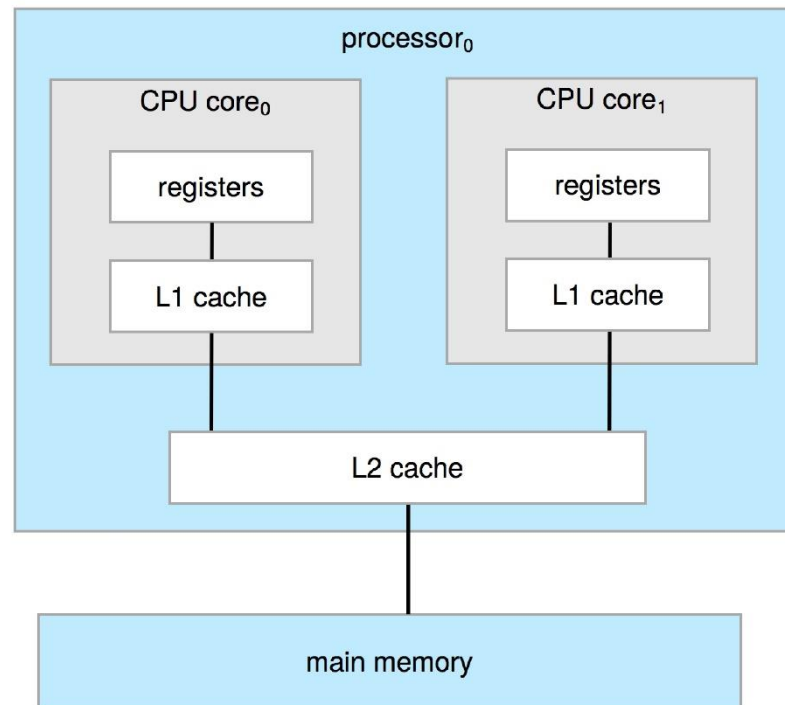
# Symmetric Multiprocessing Architecture





# A Dual-Core Design

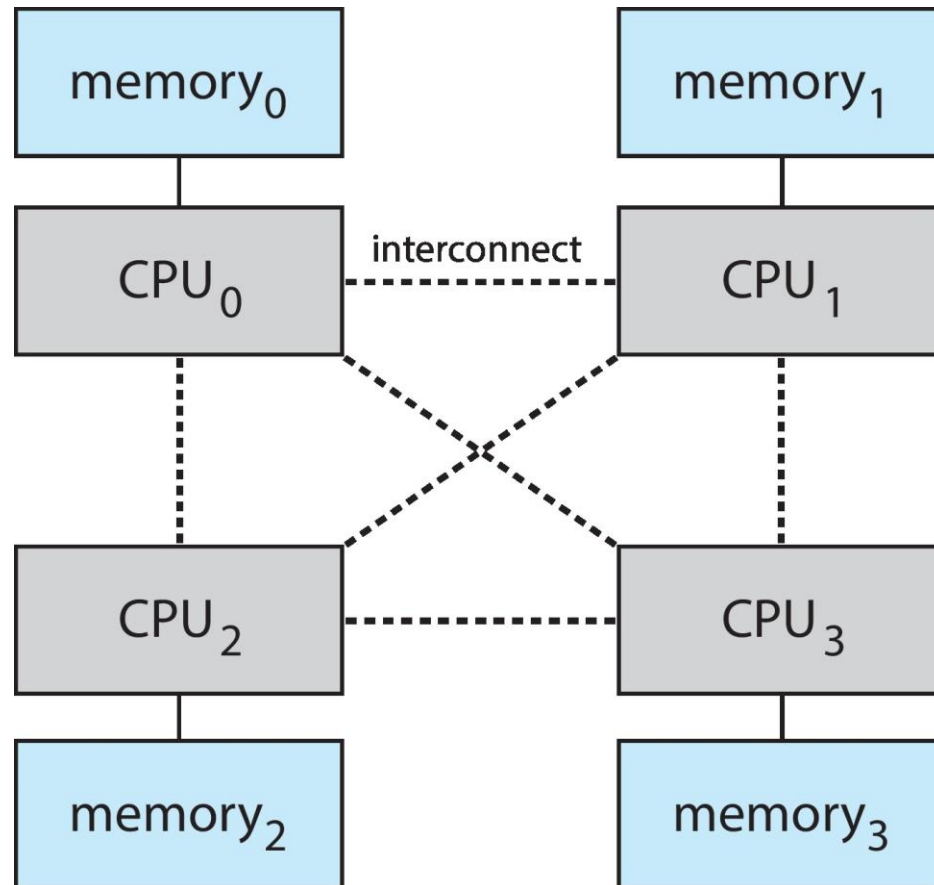
- ❑ 1 işlemci 2 çekirdek
- ❑ Tek-yonga ve çok çekirdek
- ❑ • **İşlemci** — Bir veya daha fazla CPU içeren bir fiziksel yonga.
- ❑ • **Çekirdek** — CPU'nun temel hesaplama birimi.

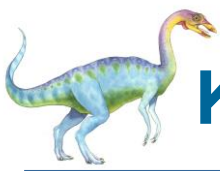






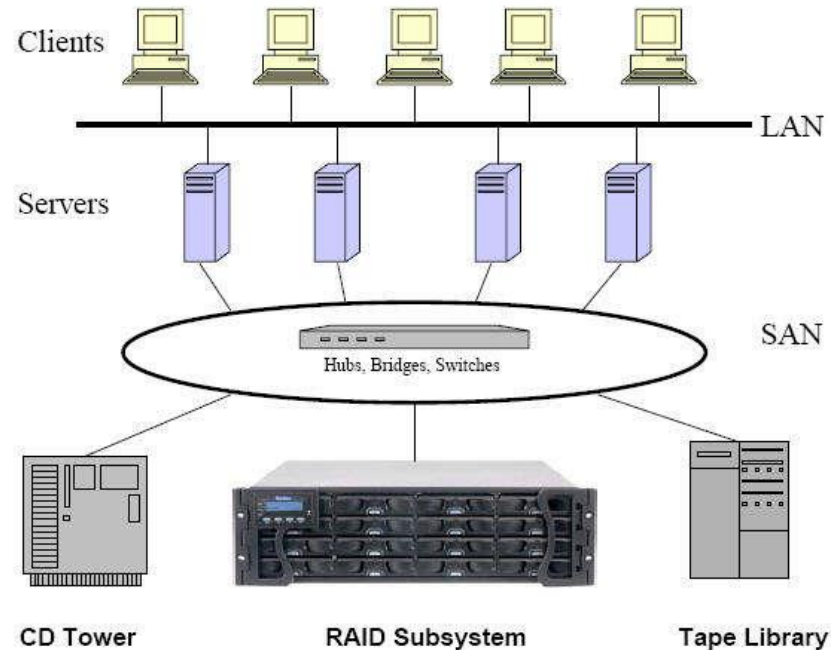
# Non-Uniform Memory Access System

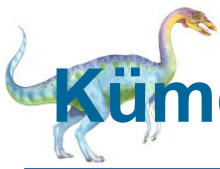




# Kümelenmiş Sistemler (Clustered Systems)

- Çok işlemcili sistemler gibi, ama birlikte çalışan çoklu sistemler
  - Genellikle depolama alanı paylaşılır (**storage-area network (SAN)**)
  - Arızalardan kurtulabilen yüksek kullanılabilirlikli bir hizmet sunar.
    - ▶ **Asimetrik kümeleme**, otomatik yedekleme (hot-standby) modunda bir makine çalışır diğeri yedek
    - ▶ **Simetrik kümeleme**, uygulamaların çalıştığı ve birbirini izleyen birden çok düğüm içerir.



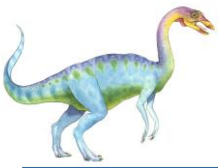


# Kümelenmiş Sistemler (Clustered Systems)

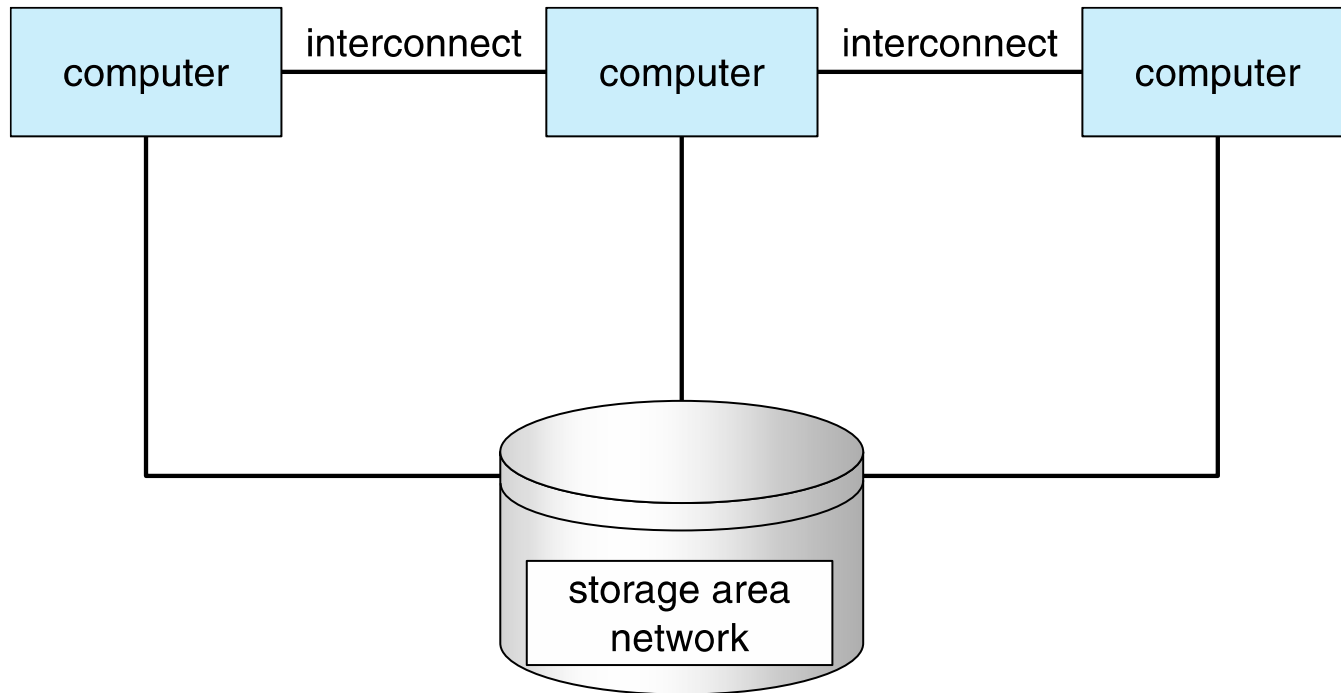
---

- Bazı kümeler yüksek özelliklere sahip **high-performance computing (HPC)** içindir.
  - ▶ Paralel çalışma için uygun yazılımlar gerekli
- Çakışan işlemlerden kaçınmak için kilit yöneticisi (DLM/**distributed lock manager**) uygulanabilir.





# Clustered Systems





# İşletim sistemi yapısı

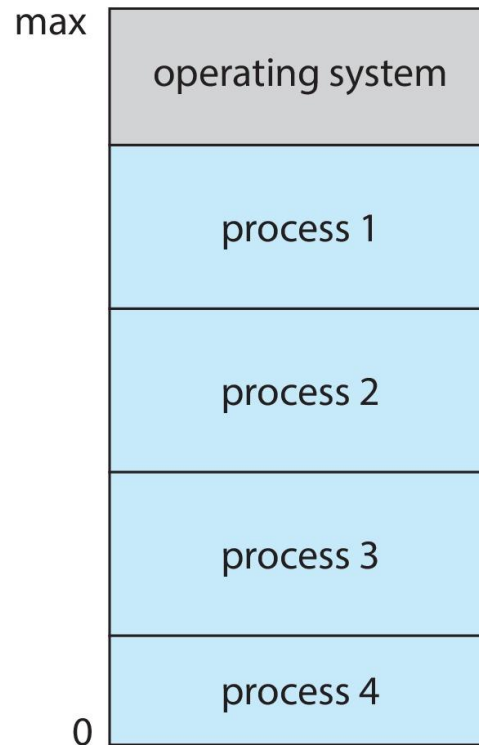
---

- **Verimlilik için gerekli (Multiprogramming)-**
  - Tek kullanıcı CPU ve I / O cihazlarını her zaman meşgul edemez.
  - Çoklu programlama işleri (kod ve veri) düzenler, böylece CPU'nun her zaman bir tane yürüttüğü kod vardır.
  - Sistemdeki toplam işlerin bir alt kümesi bellekte tutulur
  - **İş çizelgeleyicisi** tarafından bir iş seçilir ve çalıştırılır
  - Beklemesi gerektiğinde (örneğin I / O için), OS başka bir işe geçer





# Çok programlamalı bir sistemin bellek yerleşimi





# İşletim sistemi yapısı

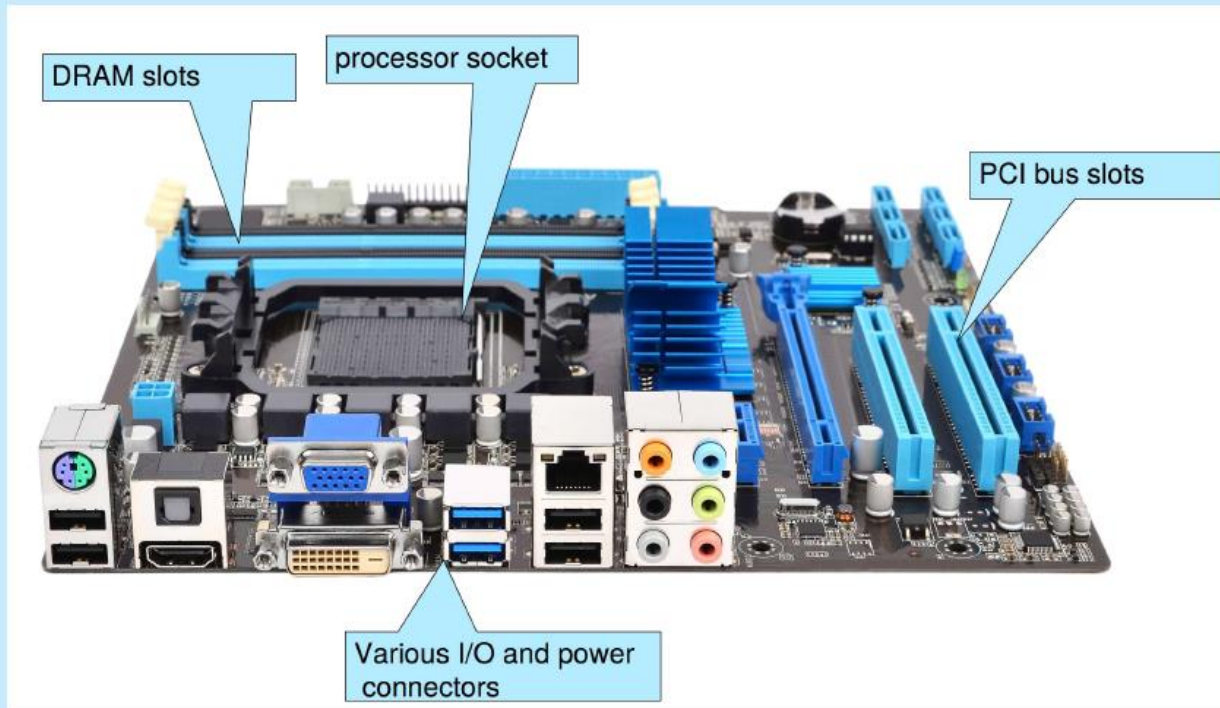
- **Zaman paylaşımı (Timesharing (multitasking) )**- işlemci işler arasında sürekli değişir, böylece kullanıcı aynı anda birden fazla program kullanabilir.
  - **Tepki zamanı** 1 sn den küçük olmalıdır
  - Her kullanıcı bellekte en az bir adet çalışan programa sahiptir- **proses**
  - Eğer aynı zamanda birden fazla iş çalışmaya hazırsa – **CPU çizelgeleyici**
  - Eğer prosesler belleğe sığmaz ise, **takas işlemi/swap** çalışma esnasında bellek içine veya dışına prosesi taşır
  - **Sanal bellek** belleğe sığmayan proseslerin çalışabilmelerine olanak tanır





# PC Motherboard

Consider the desktop PC motherboard with a processor socket shown below:



This board is a fully-functioning computer, once its slots are populated. It consists of a processor socket containing a CPU, DRAM sockets, PCIe bus slots, and I/O connectors of various types. Even the lowest-cost general-purpose CPU contains multiple cores. Some motherboards contain multiple processor sockets. More advanced computers allow more than one system board, creating NUMA systems.







# İşletim sistemi işlemleri

- **Bootstrap programı** - sistemi başlatmak için basit bir kod, çekirdeği yükler
- Çekirdeği yükler
- **system daemons/sistemin diğer bileşenleri başlatılır:**  
işletim sisteminde geri planda çalışan programlar (çekirdek dışında sağlanan hizmetler)
- Kesme tabanlı işlemler (donanımsal veya yazılımsal)
  - Donanımsal Kesme herhangi **donanım** tarafından başlatılabilir
  - Yazılım kesmesi ( yazılım hatası veya **istisna** )
    - ▶ Sıfıra bölme
    - ▶ İşletim sistemi servisi için istek
    - ▶ Diğer proses problemleri; sonsuz döngüleri vb.





# İşletim sistemi işlemleri

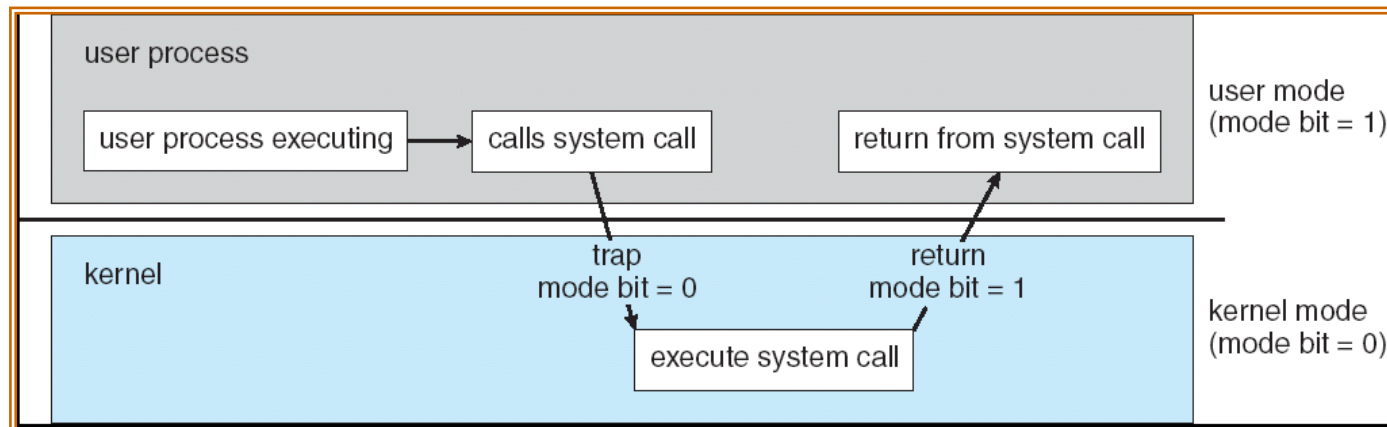
- **Çift-modlu** işlem işletim sistemine kendisini ve diğer bileşenleri korumasını sağlar
  - **Kullanıcı modu ve çekirdek modu**
  - **Mod biti** donanım tarafından sağlanır
    - ▶ Sistemin hangi modda çalıştığını belirlemeye yarar
    - ▶ İmtiyazlı bazı komutlar sadece çekirdek modunda çalışabilir
    - ▶ Sistem çağırısı, modu çekirdeğe çevirir ( kullanıcıdaki)
  - CPU'lar giderek daha çok modlu işlemleri destekliyor
  - i.e. misafir VM'ler için sanal makine yöneticisi (VMM) modu

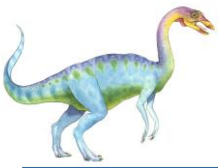




# Kullanıcı modundan çekirdek moduna geçme

- Sonsuz döngüler / aç gözlü davranan prosesler Zamanlayıcı ile engellenir
  - Belirli bir zaman periyodundan sonra kesme ayarlanır
  - İşletim sistemi sayacı saydırır
  - Sayaç sıfır olduğunda kesme üretir
  - Proses kontrolü yeniden ele almadan tespit edilir ve tahsis edilmiş zaman aşıldığında sonlandırılır.





# Proses yönetimi

- ❑ Proses çalışan bir programdır. (A process is a program in execution). Program **pasif bir varlıktır**, proses ise **aktif bir varlıktır**.
- ❑ Proses kendi görevini yerine getirmek için kaynaklara ihtiyaç duyar.
  - ❑ İşlemci, bellek, giriş/çıkış, dosyalar
  - ❑ Başlangıç verisi
- ❑ Prosesin sonlanması kullanılan kaynakların iadesini gerektirir.
- ❑ Tek akışlı proses bir sonraki çalışacak komutunun yerini gösteren bir **program sayacına** sahiptir.
  - ❑ Proses herhangi bir anda sadece tek bir komutu çalıştırır.
- ❑ Birden fazla iş akışına sahip proseslerde her bir iş akışına ait bir program sayacı vardır.
- ❑ Tipik olarak sistem, işlemci üzerinde aynı anda çalışan birçok kullanıcı ve işletim sistemi proseslerine sahiptir.





# Proses yönetim aktiviteleri

---

İşletim sistemi proses yönetimi yaparken aşağıdaki aktivitelerden sorumludur:

- Kullanıcı ve sistem proseslerini oluşturma ve silme
- Prosesleri askıya alma ve kaldığı yerden başlatma
- Proses senkronizasyon mekanizmalarını sağlama
- Proses haberleşme mekanizmalarını sağlama
- Ölümcül kilitlenme yönetimi mekanizmalarını sağlama





# Bellek yönetimi

---

- Proses çalışırken tüm veri ve komutlar bellekte olmalı
- Bellek yönetimi, işlemci kullanımını optimize ederken ve kullanıcıya cevap verirken bellekte neyin ve ne zamanda olması gerektiğini belirler
- Bellek yönetim aktiviteleri
  - Belleğin hangi parçalarının ve kim tarafından kullanılacağını izlemek
  - Hangi proseslerin veri belleğine taşınacağını veya silineceğini belirlemek
  - İstenildiğinde bellek alanını tahsis etme veya alma





# Depolama yönetimi

- İşletim Sistemi, depolama bilgisinin mantıksal bir görünümünü sağlar
  - Fiziksel özellikleri mantıksal depolama birimi olan **dosya** ya dönüştürür
  - Her ortam bir aygıt ile kontrol edilir (Disk sürücüsü, vb.)
    - ▶ Erişim hızı, kapasite, veri transfer oranı, erişim metodu gibi çeşitli özellikler
- Dosya sistemi yönetimi
  - Dosyalar genellikle klasörler halinde sınıflandırılır
  - Erişim kontrolü, güvenliği sağlar (Kim nereye erişecek ?)
  - İşletim sistemi aktiviteleri:
    - ▶ Dosya ve klasörleri silmek
    - ▶ Dosya ve klasörleri kullanmak
    - ▶ Dosyaları yardımcı belleğe taşımak
    - ▶ Dosyaları yedeklemek





# Yığın bellek yönetimi

- ❑ Genellikle diskler ana belleğe sığmayan veya kalıcı olarak tutulması gereken veriyi saklamak için kullanılır
- ❑ Sağlam bir veri yönetimi çok önemlidir
- ❑ Bilgisayarın hızı disk alt sisteminin hızına ve teknolojisine bağlıdır.
- ❑ İşletim sistemi aktiviteleri:
  - ❑ Takma ve sökme / Mounting and unmounting
  - ❑ Boş alan yönetimi
  - ❑ Depolama tahsisi
  - ❑ Disk çizelgeleme
  - ❑ Bölümleme/Partitioning
  - ❑ Koruma
- ❑ Depolama işlemleri hızlı olmalıdır
  - ❑ **Tertiary (üçüncül)** depolama, optik depolama, manyetik bant içerir
  - ❑ Yine de - işletim sistemi veya uygulamalar tarafından yönetilmeli







# Ön Belleğe alma / Caching

- Bir bilgisayarda birçok seviyede gerçekleştirilen önemli ilke (donanım, işletim sistemi, yazılım)
- Kullanılacak veriler geçici olarak yavaştan daha hızlı depolama alanına kopyalanır.
- Hızlı depolama alanı(cache) veri aranırken ilk kontrol edilir
  - Eğer veri varsa, doğrudan önbellekten alınır(fast)
  - Değilse, veriler önbelleğe kopyalanır ve oradan kullanılır
- Önbellek, Önbelleğe alınan Depolama alanından küçüktür
  - Önbellek yönetimi önemli tasarım sorunu
  - Önbellek boyutu ve değiştirme politikası





# Çeşitli Depolama ortamlarının performans analizi

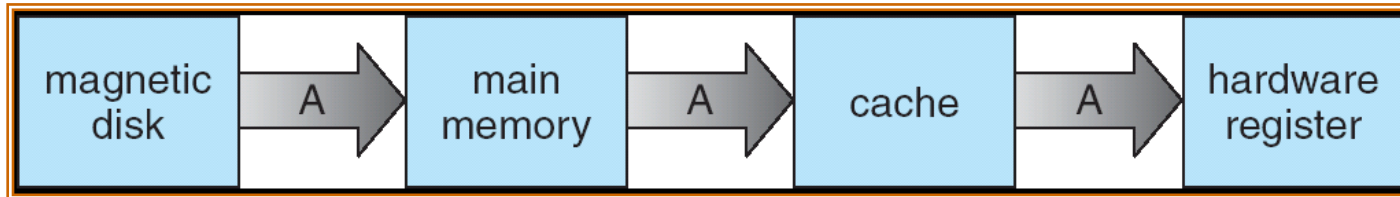
Level	1	2	3	4	5
Name	registers	cache	main memory	solid-state disk	magnetic disk
Typical size	< 1 KB	< 16MB	< 64GB	< 1 TB	< 10 TB
Implementation technology	custom memory with multiple ports CMOS	on-chip or off-chip CMOS SRAM	CMOS SRAM	flash memory	magnetic disk
Access time (ns)	0.25-0.5	0.5-25	80-250	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





## Bir A tamsayısının hardiskten kaydediciye hareketi

- Hangi kayıt birimi olursa olsun, çoklu-görev (Multi-tasking) alabilen ortamlar en son değeri kullanırken çok dikkatli olmalıdırlar.



- Çok-işlemcili ortam önbellek tutarlılığını sağlamalıdır (son değerler güncellenmeli)
- Dağıtık ortamlar daha karmaşıktır
  - Bir referansın birkaç kopyası olabilir





# Giriş/Çıkış alt sistemi

---

- İşletim sisteminin temel amaç ve işlevlerinden biri de donanım aygıtlarının zorluklarını kullanıcıdan gizlemektir
- Giriş / çıkış sistemi sorumlulukları:
  - Giriş/çıkışa ait bellek yönetimi- tamponlama, önbelleğe alma vb.
  - Genel bir aygıt sürücüsü arayüzü
  - Özel donanım aygıtları için sürücüler





# Koruma ve Güvenlik

- **Koruma** – proseslerin veya kullanıcıların kaynaklara erişimini kontrol etme mekanizması
- **Güvenlik** – harici ve dahili düşmanlara karşı sistemi savunma
  - Hizmeti engelleme saldırılarından, virüsler ve veri hırsızlığına kadar geniş bir alan
- Sistemler genellikle kimin ne yapabileceğini belirlemek ve tespit edebilmek için kullanıcıları sınıflandırır
  - Kullanıcı kimliği – **kullanıcı ID** si – her bir kullanıcı için isim ve numara içerir
  - Kullanıcı ID'si daha sonra ilgili kullanıcının tüm proses ve dosyalarıyla iliştilir- erişim kontrolü
  - Grup belirleyicisi (grup ID) herhangi bir dosya ve proses için imtiyazlı kullanıcı kümeleri oluşturur
  - İmtiyaz yönetimi kullanıcıların erişim haklarını belirlemeyi olanaklı kılar





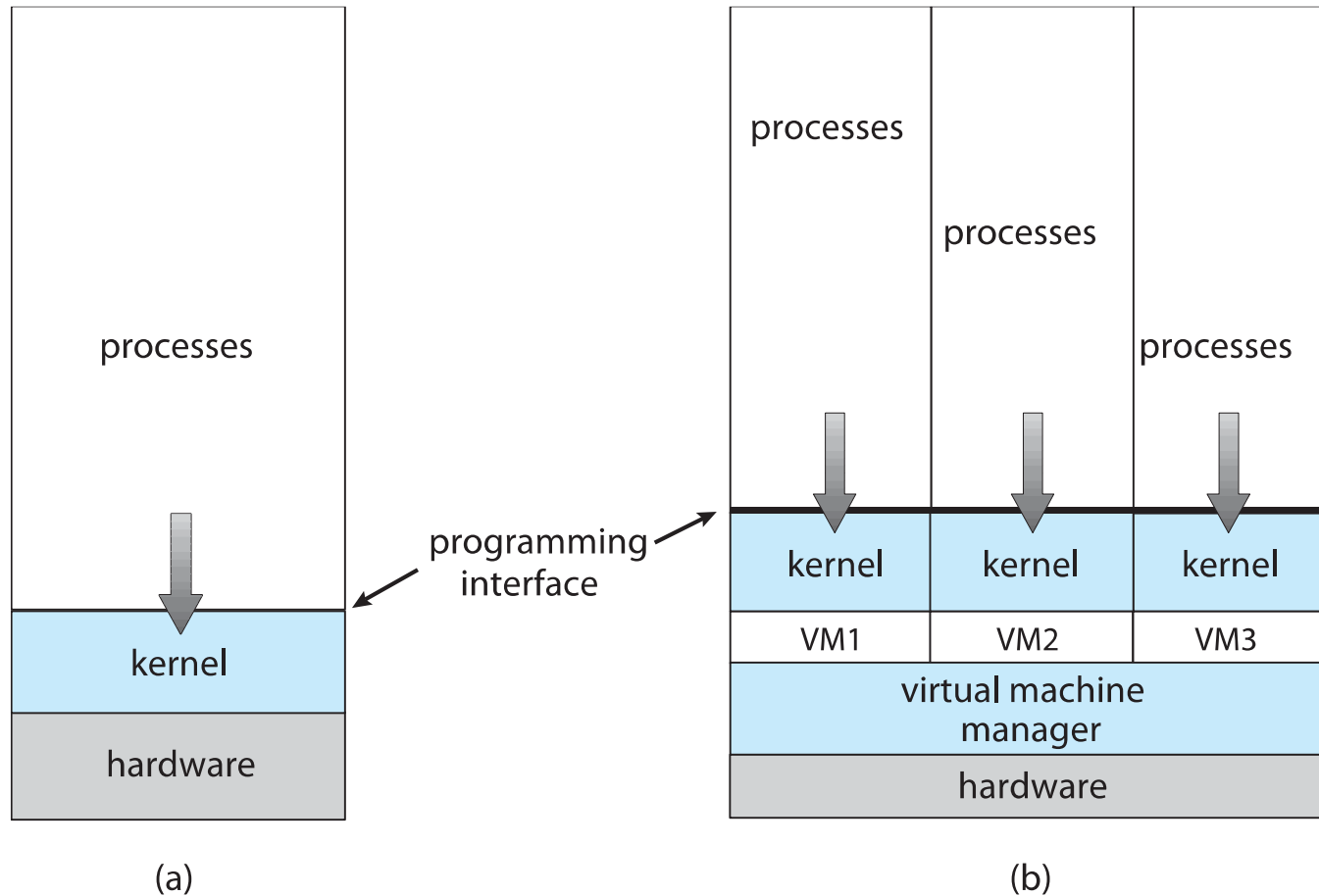
# Sanallaştırma (Virtualization)

- Diğer İşletim sist. Üzerinde çalışma imkanı
  - Geniş ve büyüyen endüstri
- **Emulation** (Emülasyon) Kaynak CPU tipi hedef türden farklı olduğunda kullanılır (i.e. PowerPC to Intel x86)
  - Yavaş bir metottur
  - Bilgisayar dili yerel kodla derlenemediğinde – Yorumlama/**Interpretation**
- **Virtualization (Sanallaştırma)** – işletim sistemi CPU için derlenmiştir ve diğer misafir kullanıcıların uyg. da derler.
  - VMM (sanal makine Yöneticisi) sanallaştırma hizmetlerini sağlar
  - Consider VMware running WinXP guests, each running applications, all on native WinXP host OS
- Kullanım durumları, keşif veya uyumluluk için birden fazla işletim sistemi çalıştıran bilgisayarları için olabilir.





# Computing Environments - Virtualization





# Virtualization (cont.)

- Kullanım durumları; keşif veya uyumluluk için birden fazla işletim sistemi kullanan dizüstü ve masaüstü bilgisayarları içerir
  - Mac OS X sunucusunu çalıştıran Apple dizüstü bilgisayar, Misafir oturum olarak Windows
  - Birden fazla sistem kullanmadan birden fazla işletim sistemi için uygulama geliştirme
  - Birden fazla sistem olmadan Kalite Güvencesi (kalite güvencesi/quality assurance) testi uygulamaları
  - Veri merkezlerinde bilgi işlem ortamlarını yürütme ve yönetme
- VMM yerel olarak çalışabilir, bu durumda onlar da ana/host bilgisayardır
  - Genel amaçlı bir ana bilgisayar yok (VMware ESX ve Citrix XenServer)







# Distributed Systems

## □ Dağıtık Hesaplama

### □ Birbirine bağlı muhtemelen heterojen sist. toplanması

▶ Ağ bir iletişim yoludur, TCP/IP en yaygın olanıdır

– Local Area Network (LAN)

– Wide Area Network (WAN)

– Metropolitan Area Network (MAN)

– Personal Area Network (PAN)

### □ Ağ İşletim Sistemi, ağdaki sistemler arasında özellikler sağlar

▶ İletişim şeması, sistemlerin mesaj alışverişi yapmasına izin verir

▶ Tek bir sistemin yansıması

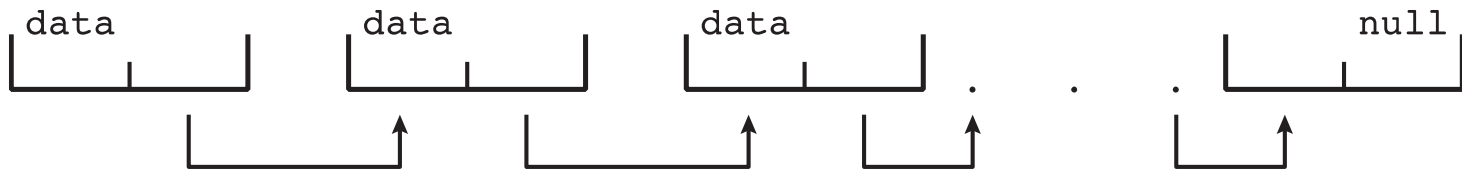




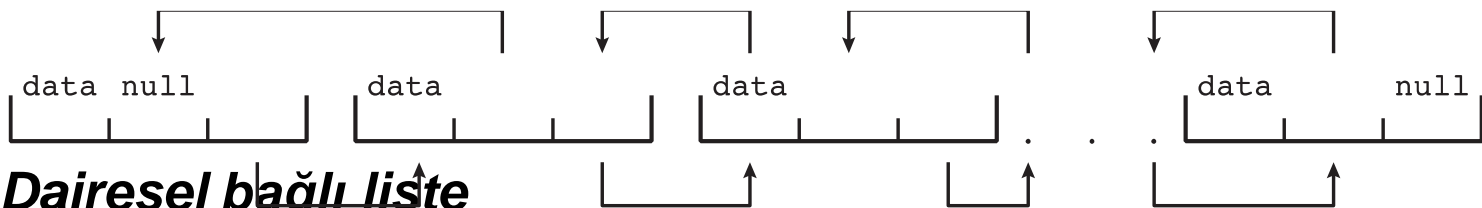
# Kernel Data Structures

n Standart programlama veri yapılarına çok benzer

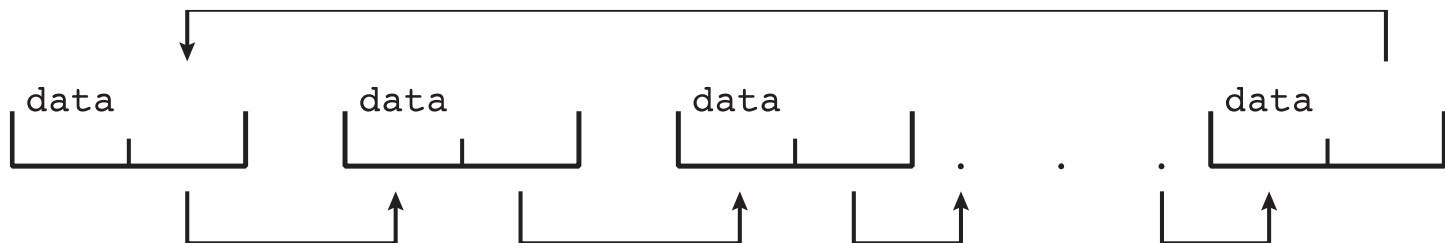
n **Tek yönlü bağlı liste**



n **Çiftyönlü bağlı liste**



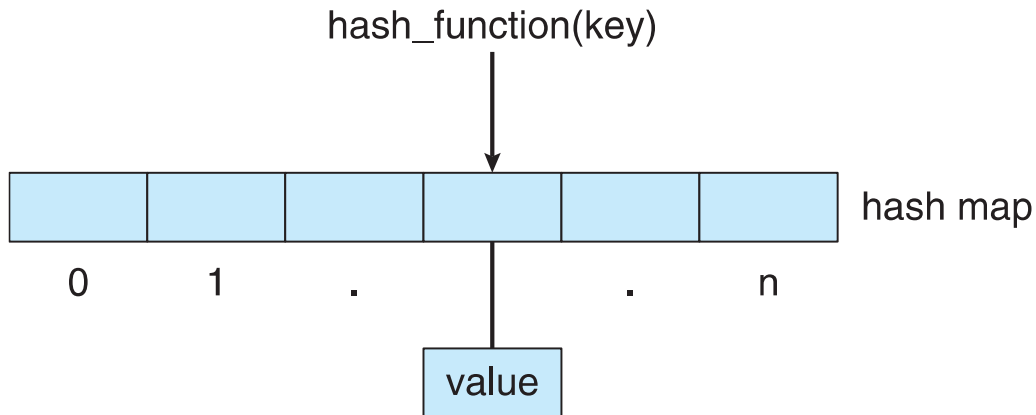
n **Dairesel bağlı liste**





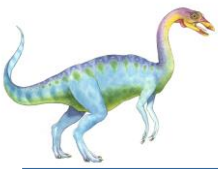
# Kernel Data Structures

- **Hash function** can create a **hash map**



- **Bitmap** – string of  $n$  binary digits representing the status of  $n$  items
- Linux data structures defined in ***include*** files `<linux/list.h>`, `<linux/kfifo.h>`, `<linux/rbtree.h>`





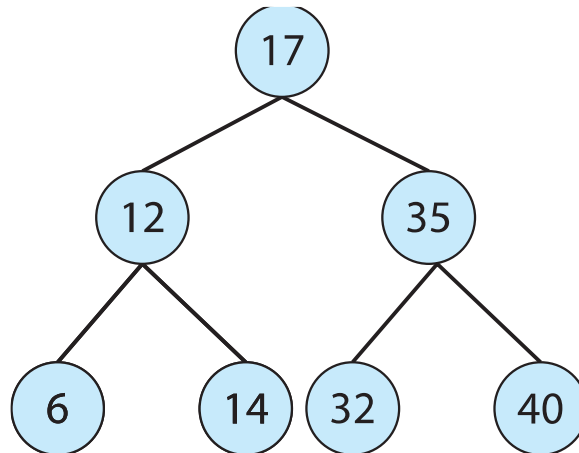
# Kernel Data Structures

## □ İkili arama ağacı

sol <= sağ

□ Arama performansı  $O(n)$

□ Dengeli ikili arama ağacının karmaşıklığı  $O(\lg n)$





# Hesaplama ortamları-Geleneksel

---

- Yalnız başına genel amaçlı makineler (Klasik bilgisayar)
- **Portals** /Portallar (Modem vb.) Web erişimi sağlar.
- **Network computers** / Ağ bilgisayarları (ufak istemciler)
- Mobil bilgisayarlar (Dizüstü Bilg.)
- Ağa bağlı bilgisayarları internetten gelen tehlikeleri korumak için güvenlik duvarı gibi sist. Kullanılır.





# Computing Environments - Mobile

---

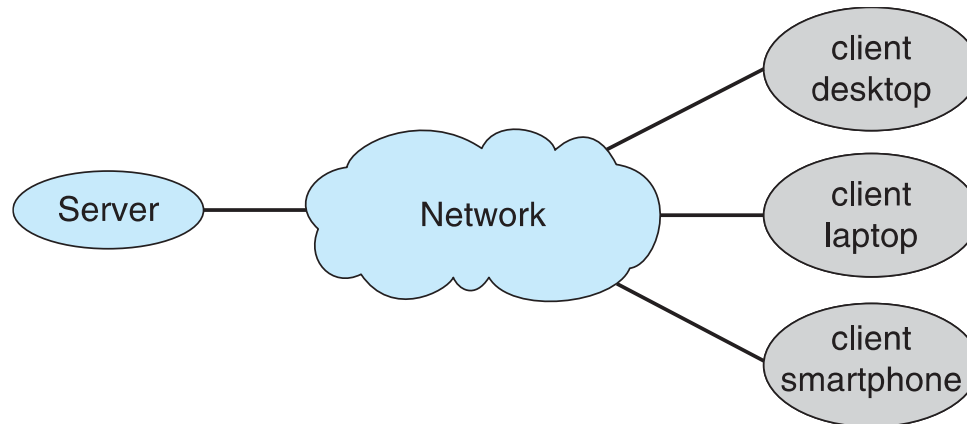
- ❑ akıllı telefonları, tabletler vb.
- ❑ “Geleneksel” bir dizüstü bilgisayar ile aralarındaki fonksiyonel fark nedir?
- ❑ Ekstra özellik - daha fazla OS özelliği (GPS, jiroskop)
- ❑ ***augmented reality*** /Artırılmış gerçeklik gibi yeni uygulama türlerine izin verir
- ❑ Bağlantı için IEEE 802.11 kablosuz veya hücresel veri ağlarını kullanır
- ❑ Liderler **Apple iOS** ve **Google Android**





# Hesaplama ortamları (Sunucu-İstemci)

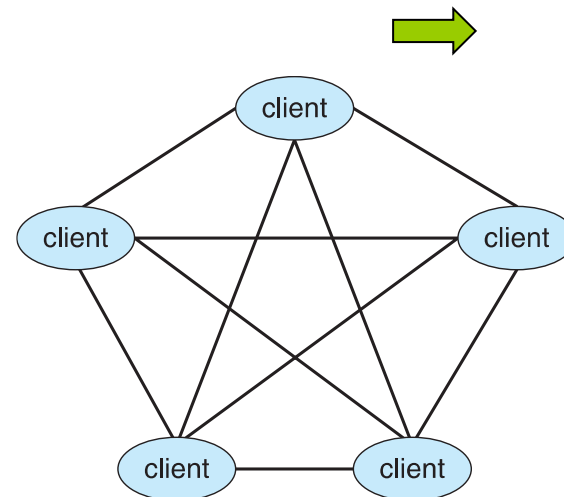
- Sunucu – istemci ortamı
  - Az gelişmiş terminaller akıllı ve gelişmiş bilgisayarlar ile yer değiştiriyor
  - Hizmet birimleri istemcilerden gelen isteklere cevap verir
    - Sunucudaki hesaplama hizmeti kullanıcılara istekleri için bir arayüz sağlar (Veritabanı)
    - Sunucudaki dosyalama hizmeti kullanıcılar dosyalarını saklayıp alma altyapısı sağlar (Bulut)





# Eşler arası hesaplama (Peer-to-Peer & P2P)

- Bir dağıtık sistem modelidir
- Sunucu ve istemci yoktur
  - Tüm düğümler eş olarak adlandırılır
  - Her biri hem istemci hem de sunucu olarak çalışabilir
  - Düğümler bir P2P ağına katılmalı
  - Naster, Gnutella, Voice over IP (VoIP) Skype



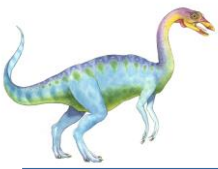




# Computing Environments – Cloud Computing

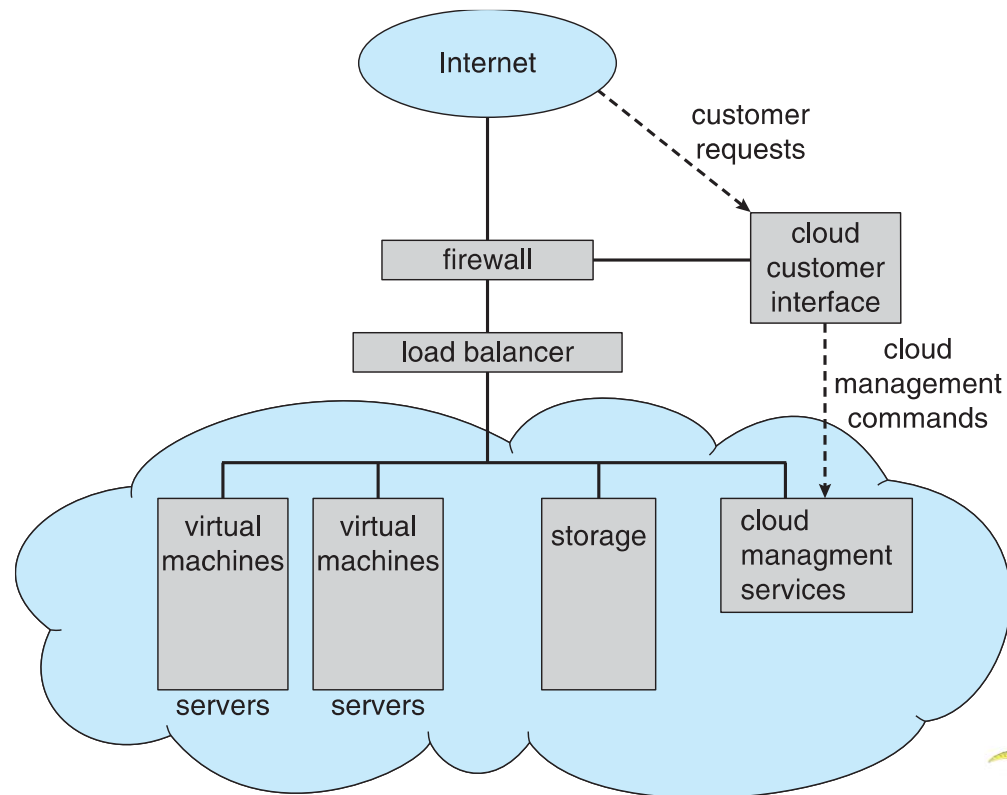
- Ağ üzerinden bir bilgisayar olarak hesaplama, depolama ve hatta uygulamalar sunar
- Sanallaştırmanın mantıksal uzantısı, çünkü sanallaştırmayı işlevsellik tabanı olarak kullanır.
  - Amazon EC2, binlerce sunucuya, milyonlarca sanal makineye, internete bağlı petabaytlarca depolama alanına, kullanıma göre ödeme yapıyorsunuz
- Birçok çeşit
  - Genel bulut - İnternet üzerinden herkes erişebilir
  - Özel bulut - şirketin kendi kullanımı için bir şirket tarafından işletilmektedir.
  - Karma bulut - hem genel hem de özel bulut bileşenlerini içerir.
  - Hizmet Olarak Yazılım (Software as a Service) - İnternet üzerinden kullanılabilen bir veya daha fazla uygulama (ör. Kelime işlemci)
  - Hizmet Olarak Platform (Platform as a Service) - İnternet üzerinden uygulama kullanımı için hazır yazılım yığını (ör. Bir veritabanı sunucusu)
  - Hizmet Olarak Altyapı (Infrastructure as a Service) - İnternet üzerinden kullanılabilen sunucular veya depolama (yani, yedekleme için kullanılabilir depolama alanı)





# Computing Environments – Cloud Computing

- Bulut hesaplama ; Geleneksel işletim sistemleri artı VMM'ler ve bulut yönetim araçları
  - İnternet bağlantısı firewall gibi güvenlik gerektirir
  - Yük dengeleyicileri, birden çok uygulama arasında trafiği dengelerler.





# Computing Environments – Real-Time Embedded Systems

- Gerçek zamanlı gömülü sistemler bilgisayarların en yaygın biçimi
  - özel amaçlı, sınırlı amaçlı işletim sistemi, gerçek zamanlı işletim sistemi gibi...
- Diğer birçok özel hesaplama ortamları
  - Bazıları OS'lere sahip, bazıları işletim sist. olmadan görevleri gerçekleştiriyor
- Gerçek zamanlı işletim sistemi iyi tanımlanmış sabit zaman kısıtlamalarına sahiptir
  - İşlemler zaman kısıtlaması içinde yapılmalıdır
  - İşlemler kısıtlamalara uymak zorunda





# Free and Open-Source Operating Systems

---

- ❑ Hem özgür işletim sistemleri ve açık kaynak işletim sistemleri, derlenmiş ikili kod yerine kaynak kodu biçiminde kullanılabilir.
- ❑ Diğer işletim sist. sadece ikili kaynak kodlu ve kaynak kod formatında kullanılabilir
- ❑ Kopya koruma ve Dijital Hak Yönetimi (DRM) bu işletim sist. korur
- ❑ GNU Public License (GPL) ve Free Software Foundation (FSF)
  - ❑ Özgür yazılım ve açık kaynak yazılım, farklı insan grupları tarafından desteklenen iki farklı fikirdir.
    - ▶ <http://gnu.org/philosophy/open-source-misses-the-point.html/>





# Free and Open-Source Operating Systems

---

- Örnekler arasında GNU / Linux ve BSD UNIX (Mac OS X'in çekirdeği dahil) ve daha birçok
- VMM'yi VMware Player (Windows'ta ücretsiz), Virtualbox (açık kaynak ve birçok platformda ücretsiz) - <http://www.virtualbox.com>) gibi kullanabilir
  - Araştırmak için misafir işletim sistemlerini çalıştırmak için kullanılabilir





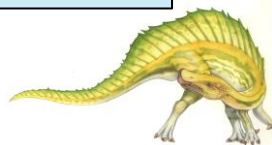
# The Study of Operating Systems

There has never been a more interesting time to study operating systems, and it has never been easier. The open-source movement has overtaken operating systems, causing many of them to be made available in both source and binary (executable) format. The list of operating systems available in both formats includes Linux, BSD UNIX, Solaris, and part of macOS. The availability of source code allows us to study operating systems from the inside out. Questions that we could once answer only by looking at documentation or the behavior of an operating system we can now answer by examining the code itself.

Operating systems that are no longer commercially viable have been open-sourced as well, enabling us to study how systems operated in a time of fewer CPU, memory, and storage resources. An extensive but incomplete list of open-source operating-system projects is available from [https://curlie.org/Computers/Software/Operating\\_Systems/Open\\_Source/](https://curlie.org/Computers/Software/Operating_Systems/Open_Source/)

In addition, the rise of virtualization as a mainstream (and frequently free) computer function makes it possible to run many operating systems on top of one core system. For example, VMware (<http://www.vmware.com>) provides a free “player” for Windows on which hundreds of free “virtual appliances” can run. Virtualbox (<http://www.virtualbox.com>) provides a free, open-source virtual machine manager on many operating systems. Using such tools, students can try out hundreds of operating systems without dedicated hardware.

The advent of open-source operating systems has also made it easier to make the move from student to operating-system developer. With some knowledge, some effort, and an Internet connection, a student can even create a new operating-system distribution. Just a few years ago, it was difficult or impossible to get access to source code. Now, such access is limited only by how much interest, time, and disk space a student has.



# 1. Bölümün sonu

