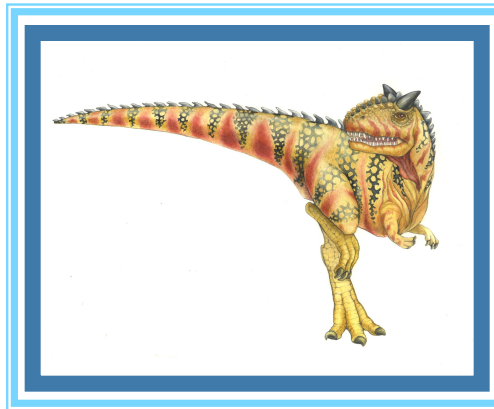


# Chapter 1: Introduction





# Chapter 1: Introduction

---

- What Operating Systems Do (İşletim Sistemleri Nedir)
- Computer-System Organization (Bilgisayar Organizasyonu)
- Computer-System Architecture(Bilgisayar Mimarisi)
- Operating-System Structure( İşletim Sistemi Yapısı)
- Operating-System Operations(İşletim Sistemi Operasyonları)
- Process Management( İşlem Yönetimi)
- Memory Management(Bellek Yönetimi)
- Storage Management (Depolama Yönetimi)
- Protection and Security (Koruma ve Güvenlik)
- Kernel Data Structures (Çekirdek Veri Yapıları)
- Computing Environments(Çalışma Ortamları)





# What is an Operating System?

---

- A program that acts as an intermediary between a user of a computer and the computer hardware (**Bir bilgisayar kullanıcısı ve bilgisayar donanımı arasında aracı olarak görev yapan bir program**)
- Operating system goals(**hedef**):
  - Execute user programs and make solving user problems easier (**Kullanıcı programlarını çalıştırma ve kullanıcı problemlerini çözme**)
  - Make the computer system convenient to use(**sistemi uygun hale getirme**)
  - Use the computer hardware in an efficient manner(**donanımı verimli şekilde kullanma**)





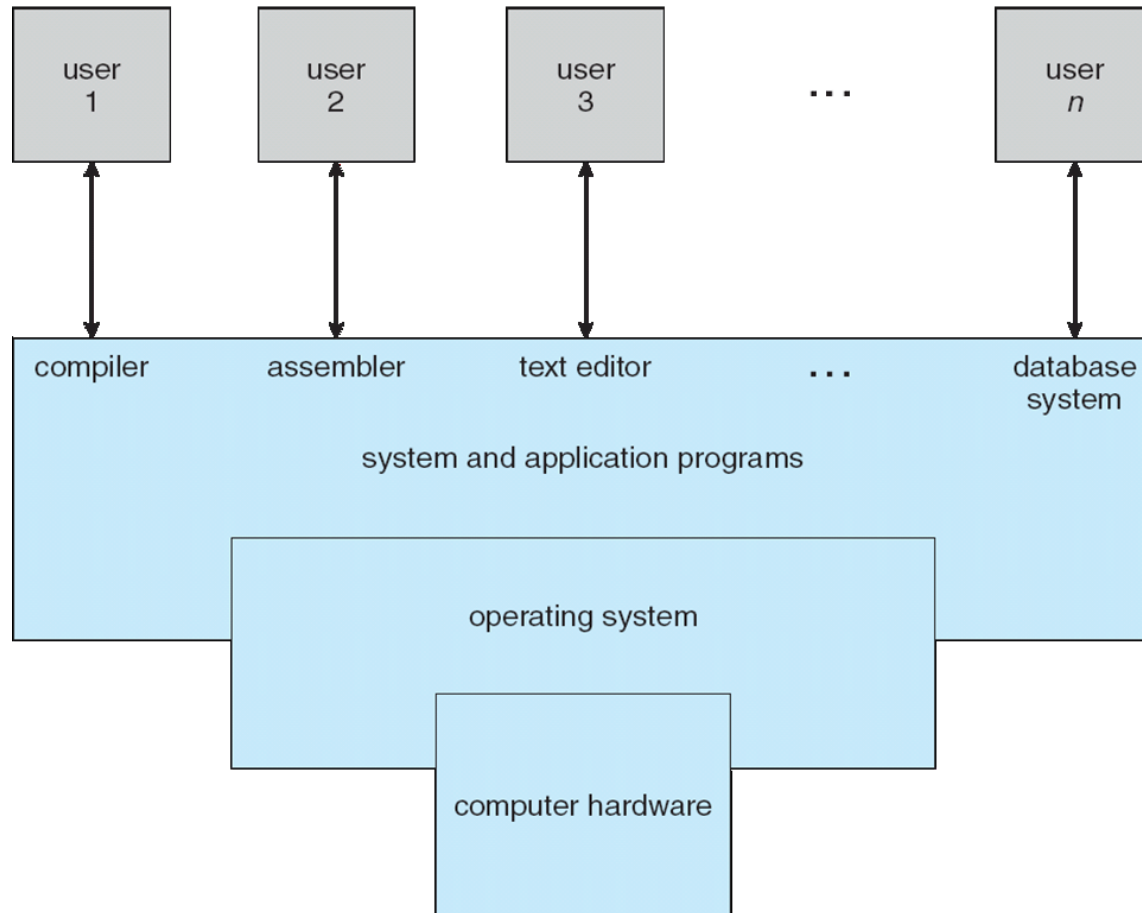
# Computer System Structure

- Computer system can be divided into four components(**sistem 4 bileşene ayrılabilir**):
  - Hardware – provides basic computing resources
    - ▶ CPU, memory, I/O devices
  - Operating system
    - ▶ Controls and coordinates use of hardware among various applications and users(**Çeşitli uygulamalar ve kullanıcılar arasında donanım kullanımını kontrol eder ve koordine eder**)
  - Application programs – define the ways in which the system resources are used to solve the computing problems of the users (**kullanıcıların bilgi işlem problemlerini çözmek için sistem kaynaklarının nasıl kullanıldığını tanımlar**)
    - ▶ Word processors, compilers, web browsers, database systems, video games
  - Users
    - ▶ People, machines, other computers





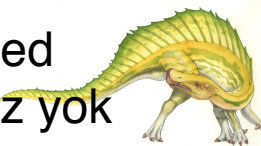
# Four Components of a Computer System





# What Operating Systems Do

- Depends on the point of view (**Göreceli**)
- Users want convenience, **ease of use** and **good performance**(**Kolaylık ve kullanım kolaylığı ve iyi performans**)
  - Don't care about **resource utilization** (**Kullanıcılar Kaynak kullanımı ile ilgilenmesin**)
- But shared computer such as **mainframe** or **minicomputer** must keep all users happy( **paylaşımlı bilgisayarlar herkesi mutlu etmek zorunda**)
- Users of dedicate systems such as **workstations** have dedicated resources but frequently use shared resources from **servers**(**İş istasyonları gibi adanmış sistem kullanıcılarına özel kaynaklara sahip olmakla birlikte sıklıkla sunuculardan paylaşılan kaynakları kullanır**)
- Handheld computers are resource poor, optimized for usability and battery life(**Mobil veya el bilgisayarı kaynak yönünden zayıf:kullanılabilirlik ve batarya için**)
- Some computers have little or no user interface, such as embedded computers in devices and automobiles(**Bazı bilgisayarlarda arayüz yok ya da az, gömülü sistem örneği**)





# Operating System Definition

---

- OS is a **resource allocator (kaynak ayırıcı)**
  - Manages all resources (kaynakları yönetir)
  - Decides between conflicting requests for efficient and fair resource use( **Çakışan istekleri için etkin ve adil kullanım**)
- OS is a **control program (kontrol program)**
  - Controls execution of programs to prevent errors and improper use of the computer (**Hatalar ve yürütülen programlar denetlenir**)





# Operating System Definition (Cont.)

- No universally accepted definition ( Genel bir tanım yok)
- “Everything a vendor ships when you order an operating system” (**Bir işletim sistemi siparişi verdiğinizde bir satıcı tarafından gönderilen her şey**)
  - But varies wildly (Tanımlar hızlıca değişir)
- “The one program running at all times on the computer” is the **kernel**. (**Çekirdek her zaman çalışır**)
- Everything else is either
  - a system program (ships with the operating system) , or
  - an application program.







# Computer Startup

- **bootstrap program** is loaded at power-up or reboot
  - Typically stored in ROM or EPROM
  - Initializes all aspects of system
  - Loads operating system kernel and starts execution
  
- (Bir işletim sistemi , bilgisayarın başlamasından sorumlu mudur?)
- BIOS anakarta ait ROM ve EPROMda.
- Tüm donanımlar anakarta işaret gönderir.
- Sonrasında işletim sistemi çekirdeği çalışmaya başlar.(diskteki özel dosyaları arar. Bunu bios söyler)

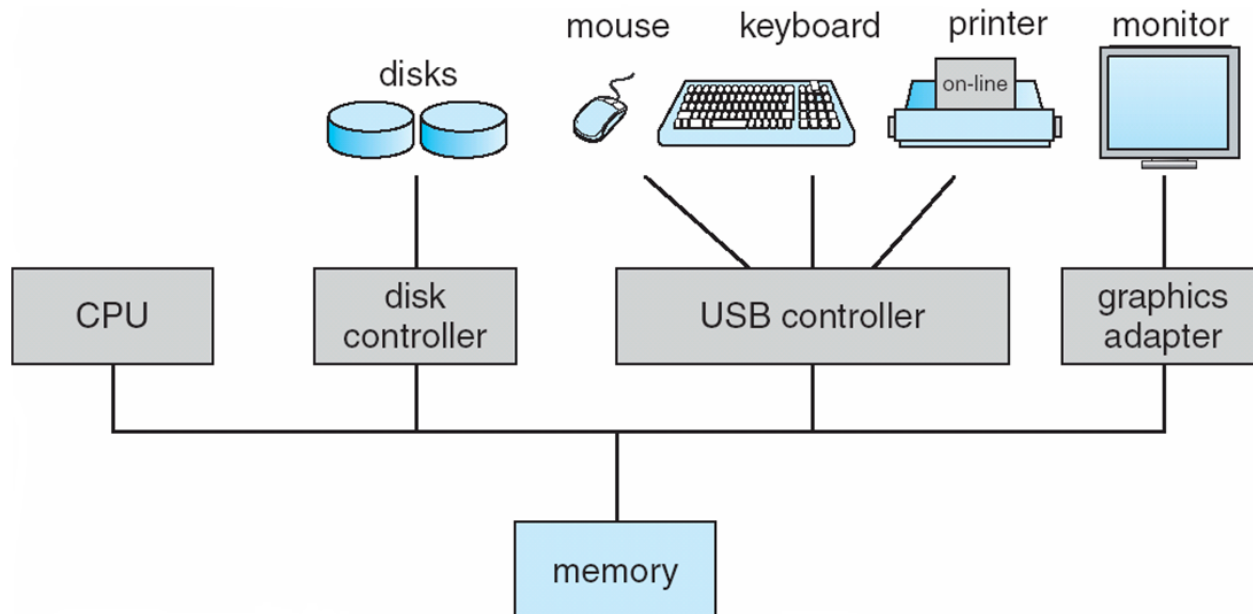




# Computer System Organization

## ■ Computer-system operation

- One or more CPUs, device controllers connect through common bus providing access to shared memory(**Bir veya daha fazla işlemci, aygıt denetleyicileri, paylaşılan belleğe erişim sağlayan ortak bir veri yolu vasıtasıyla bağlanır**)
- Concurrent execution of CPUs and devices competing for memory cycles(**Bellek çevrimleri için yarışan CPU'ların ve aygıtların eş zamanlı yürütülmesi**)





# Computer-System Operation

- I/O devices and the CPU can execute concurrently
- Each device controller is in charge of a particular device type
- Each device controller has a local buffer
- CPU moves data from/to main memory to/from local buffers
- I/O is from the device to local buffer of controller
- Device controller informs CPU that it has finished its operation by causing an **interrupt**
- **(Tüm giriş çıkış cihazları ve CPU eşzamanlı çalışır**
- **Her cihaz için denetleyici var,**
- **Her denetleyici lokal belleğe sahiptir.**
- **işlemci veriyi bellekten lokal tamponlara taşır.**
- **I/O cihazdan denetleyici tampona taşır.**
- **Aygıt denetleyicisi kesme oluşturarak CPU'ya işlemin bittiğini haber verir)**





# Common Functions of Interrupts

- Interrupt transfers control to the interrupt service routine generally, through the **interrupt vector**, which contains the addresses of all the service routines
- Interrupt architecture must save the address of the interrupted instruction
- A **trap** or **exception** is a software-generated interrupt caused either by an error or a user request
- An operating system is **interrupt driven**
- Kesintiler kontrolü, o kesintiye ait **kesinti servis rutinine (interrupt service routine)** yönlendirir
- Servis rutinleri, kesinti sonucu yapılması gereken işi gerçekleştiren yazılım parçacıklarıdır
- Hangi servis rutininin hangi hafıza adresinde bulunduğu **kesinti vektöründe (interrupt vector)** bulunmaktadır
- Bilgisayar, kesinti sonunda yarıda kesilen işleme geri dönebilmek için, kesilen işlemin işletilen son komutunun adresini saklamalıdır
- Kayıp kesintilere engel olmak için kesinti işletildiği sürece yeni kesinti gönderimine izin verilmez
- **Tuzak (trap)** yazılım tarafından oluşturulan kesintilerdir
- Tuzaklara yazılım hataları ya da kullanıcı istekleri neden olur
- İşletim sistemleri kesintilerle yönlendirilirler (interrupt driven)





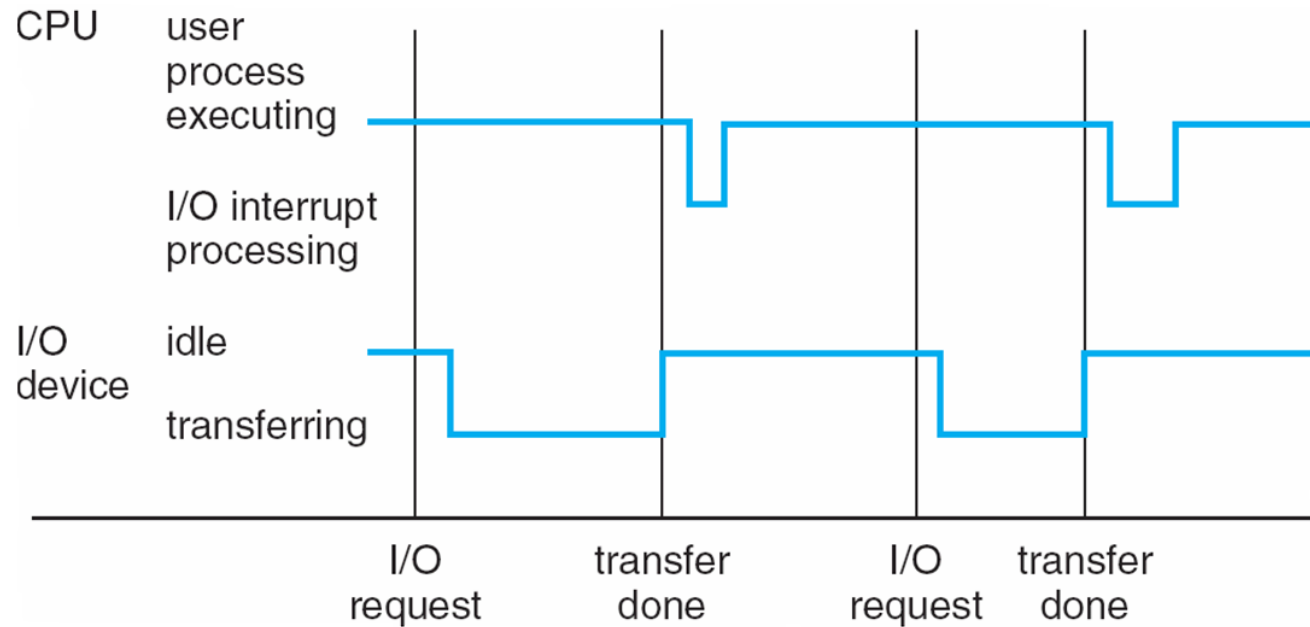
# Interrupt Handling

- The operating system preserves the state of the CPU by storing registers and the program counter
- Determines which type of interrupt has occurred:
  - **vectored** interrupt system
- Separate segments of code determine what action should be taken for each type of interrupt
- İşletim sistemi CPU'nun durumunu kaydeder: **yazmaçlar (registers)** ve **program sayacı (program counter)**
- Hangi tür kesintinin gerçekleştiğini belirler:
  - **sorgulama (polling)** – hangi cihazdan gerçekleştiği bulunmalıdır
  - **vektör kesinti sistemi (vectored interrupt system)** – cihazı belirten kod, kesinti ile birlikte gönderilir
- Her bir kesinti için hangi işlemin gerçekleştirileceğini ayrı bir kod parçası belirler





# Interrupt Timeline





# Storage Definitions and Notation Review

The basic unit of computer storage is the **bit**. A bit can contain one of two values, 0 and 1. All other storage in a computer is based on collections of bits. Given enough bits, it is amazing how many things a computer can represent: numbers, letters, images, movies, sounds, documents, and programs, to name a few. A **byte** is 8 bits, and on most computers it is the smallest convenient chunk of storage. For example, most computers don't have an instruction to move a bit but do have one to move a byte. A less common term is **word**, which is a given computer architecture's native unit of data. A word is made up of one or more bytes. For example, a computer that has 64-bit registers and 64-bit memory addressing typically has 64-bit (8-byte) words. A computer executes many operations in its native word size rather than a byte at a time.

Computer storage, along with most computer throughput, is generally measured and manipulated in bytes and collections of bytes.

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

Computer manufacturers often round off these numbers and say that a megabyte is 1 million bytes and a gigabyte is 1 billion bytes. Networking measurements are an exception to this general rule; they are given in bits (because networks move data a bit at a time).





# Storage Structure

- Main memory – only large storage media that the CPU can access directly
  - Random access
  - Typically **volatile**(geçici)
- Secondary storage – extension of main memory that provides large **nonvolatile** storage capacity
- Hard disks – rigid metal or glass platters covered with magnetic recording material
  - Disk surface is logically divided into **tracks**, which are subdivided into **sectors**
  - The **disk controller** determines the logical interaction between the device and the computer
- **Solid-state disks** – faster than hard disks, nonvolatile
  - Various technologies
  - Becoming more popular







# Storage Hierarchy

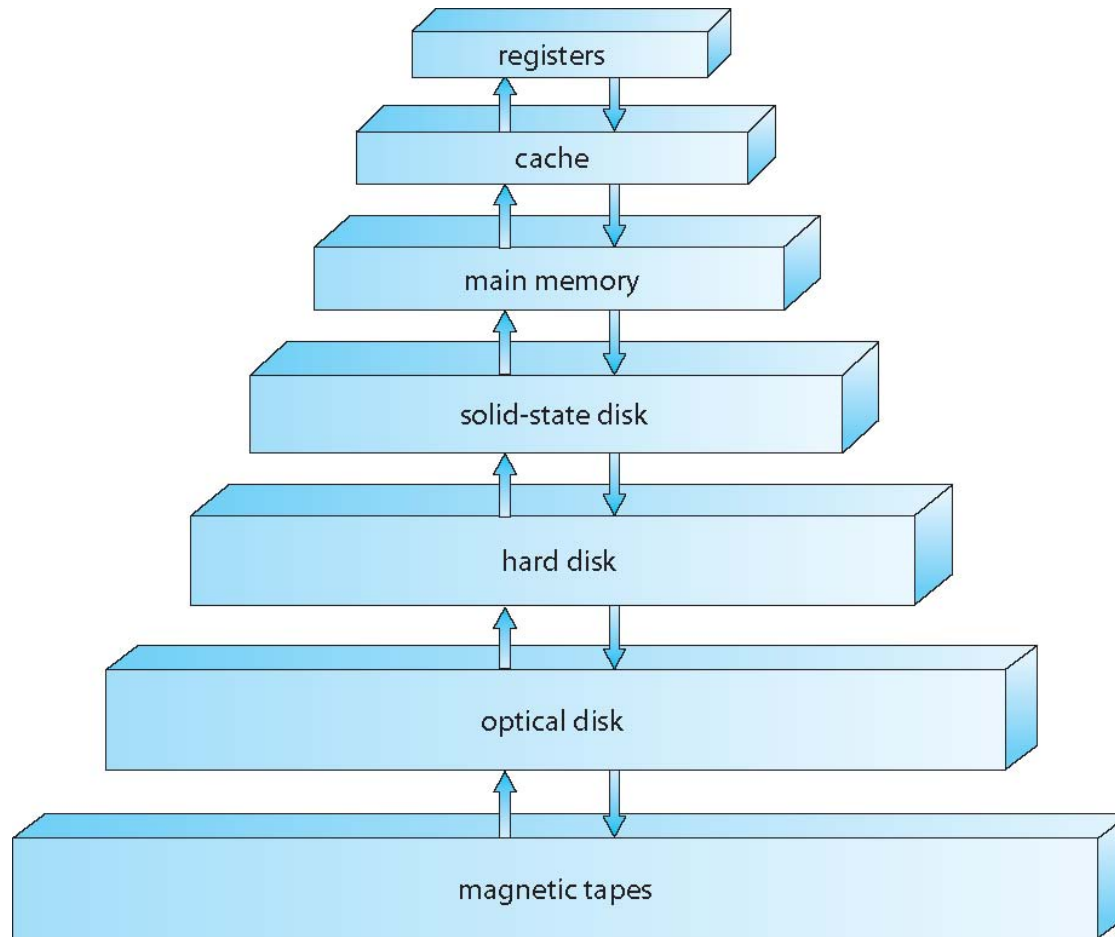
---

- Storage systems organized in hierarchy
  - Speed
  - Cost
  - Volatility
- **Caching** – copying information into faster storage system; main memory can be viewed as a cache for secondary storage
- **Device Driver** for each device controller to manage I/O
  - Provides uniform interface between controller and kernel





# Storage-Device Hierarchy





# Caching

- Important principle, performed at many levels in a computer (in hardware, operating system, software)
- Information in use copied from slower to faster storage temporarily
- Faster storage (cache) checked first to determine if information is there
  - If it is, information used directly from the cache (fast)
  - If not, data copied to cache and used there
- Cache smaller than storage being cached
  - Cache management important design problem
  - Cache size and replacement policy
- **Aranan bilgi öncelikle daha hızlı depolama biriminde mi (ön bellek) kontrol edilir**
  - Eğer oradaysa, bilgi direk ön bellekten alınır (hızlı)
  - Eğer değilse, ön belleğe alınır ve oradan kullanılır
- **Ön bellek**, ön belleğe alınacak bilgiden daha küçük ise
  - Ön bellek yönetimi önemli bir tasarım problemidir
  - Ön bellek boyutu ve **yenileme politikası (replacement policy)**





# Direct Memory Access Structure

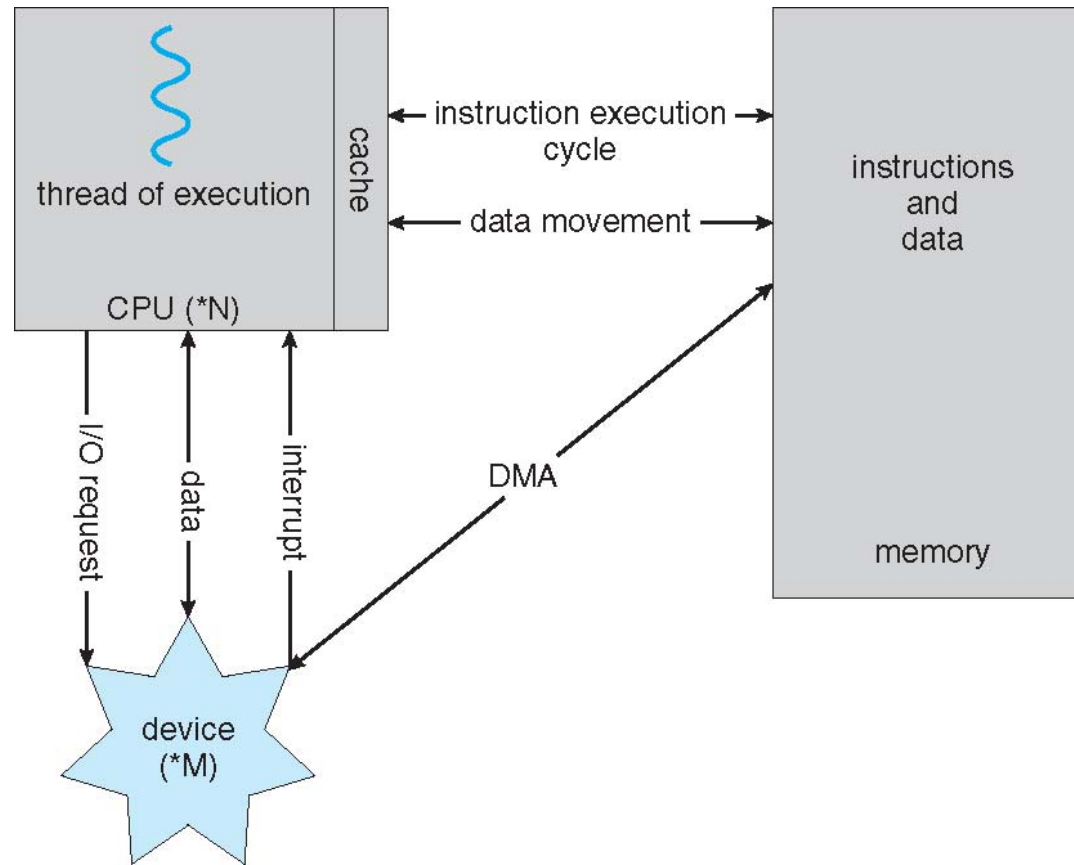
---

- Used for high-speed I/O devices able to transmit information at close to memory speeds
- Device controller transfers blocks of data from buffer storage directly to main memory without CPU intervention
- Only one interrupt is generated per block, rather than the one interrupt per byte
- **Hafıza hızına yakın bilgi aktarması yapabilen yüksek hızlı I/O cihazları için kullanılır**
- **Cihaz denetleyicisinin, CPU'nun çalışmasını bölmeden, veri bloklarını cihazın tampon belleğinden direk olarak hafızaya aktarmasıdır**
- **Her byte için kesinti göndermek yerine, her bir blok için bir kesinti gönderilir**
- **belleğe erişim**





# How a Modern Computer Works



*A von Neumann architecture*





# Computer-System Architecture

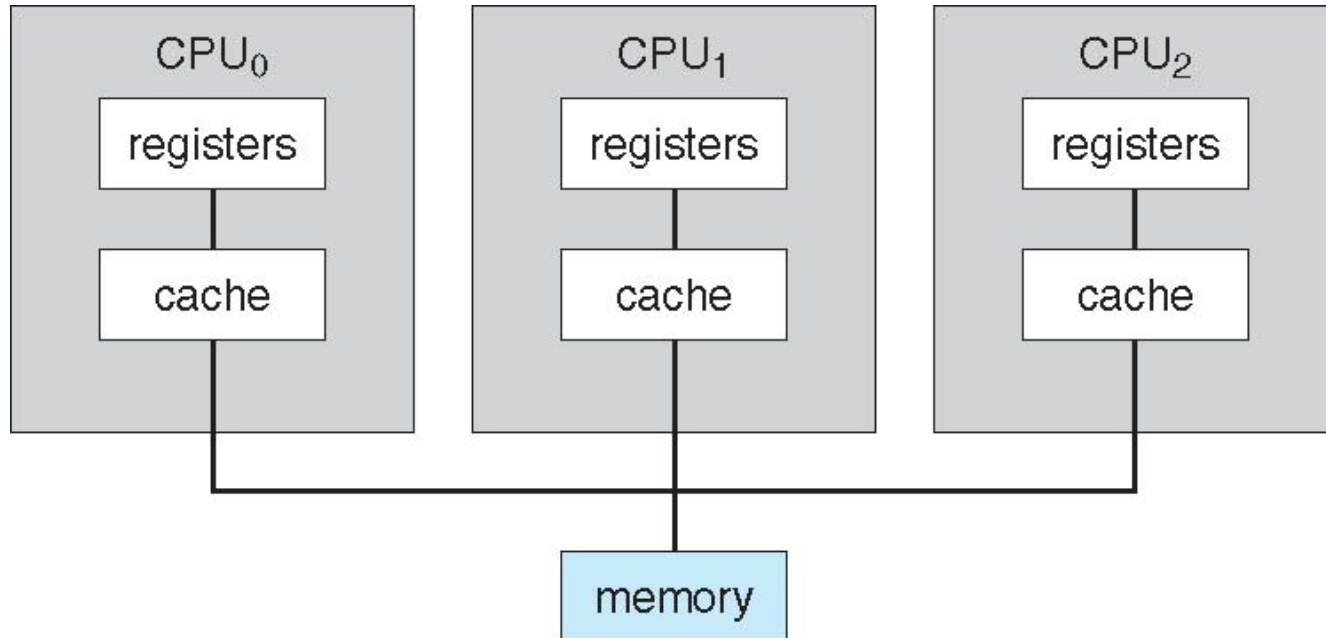
---

- Most systems use a single general-purpose processor
  - Most systems have special-purpose processors as well
- **Multiprocessors** systems growing in use and importance
  - Also known as **parallel systems**, **tightly-coupled systems**
  - Advantages include:
    1. **Increased throughput**
    2. **Economy of scale**
    3. **Increased reliability** – graceful degradation or fault tolerance
  - Two types:
    1. **Asymmetric Multiprocessing** – each processor is assigned a specific task.
    2. **Symmetric Multiprocessing** – each processor performs all tasks





# Symmetric Multiprocessing Architecture





# Operating System Structure

- **Multiprogramming (Batch system)** needed for efficiency (Verimlilik için çoklu programlama) (bir problemi çok parçaya böl)
  - Single user cannot keep CPU and I/O devices busy at all times (tek kullanıcı CPU yu meşgul edemez)
  - Multiprogramming organizes jobs (code and data) so CPU always has one to **execute** (çoklu programlama görevleri organize eder)
  - A subset of total jobs in system is kept in memory ( Tüm işler hafızada)
  - One job selected and run via **job scheduling (Bir iş koşturulur)**
  - When it has to wait (for I/O for example), OS switches to another job ( O beklerken diğer işler koşturulur)







# Operating System Structure

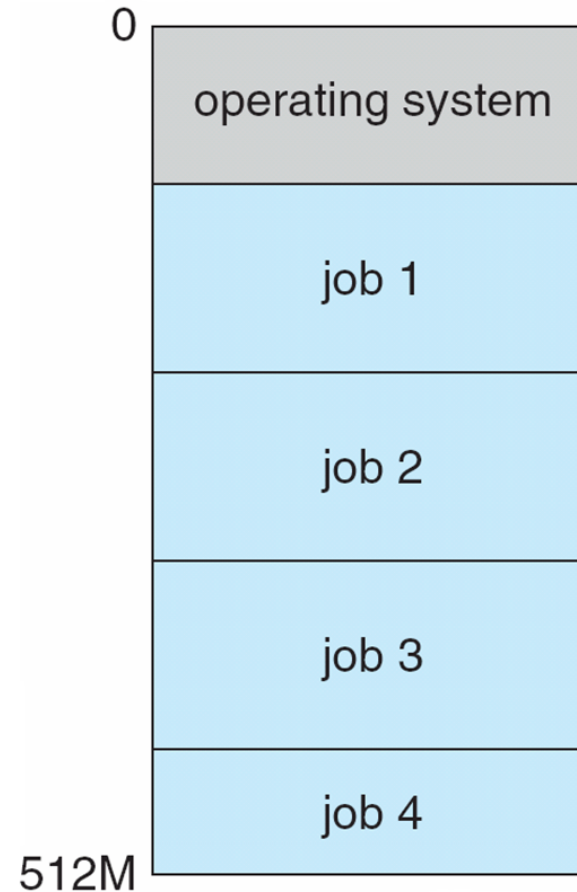
---

- **Timesharing (multitasking)** is logical extension in which CPU switches jobs so frequently that users can interact with each job while it is running, creating **interactive** computing (Birden fazla problemi zamana yay)
  - **Response time** should be  $< 1$  second
  - Each user has at least one program executing in memory  $\Rightarrow$  **process**
  - If several jobs ready to run at the same time  $\Rightarrow$  **CPU scheduling**
  - If processes don't fit in memory, **swapping** moves them in and out to run
  - **Virtual memory** allows execution of processes not completely in memory





# Memory Layout for Multiprogrammed System





# Operating-System Operations

- **Interrupt driven** (hardware and software)
  - Hardware interrupt by one of the devices
  - Software interrupt (**exception** or **trap**):
    - ▶ Software error (e.g., division by zero)
    - ▶ Request for operating system service
    - ▶ Other process problems include infinite loop, processes modifying each other or the operating system

İşletim sistem kesme yönelimlidir:

- ▶ Aygıtlardan biri tarafından yapılan donanım kesmesi
- ▶ Yazılım kesmesi (istisna veya tuzak):
  - Yazılım hatası (ör. Sıfıra bölünme)
  - İşletim sistemi hizmeti isteği
  - Diğer işlem sorunları arasında sonsuz döngü, birbirlerini değiştiren işlemler...





# Operating-System Operations (cont.)

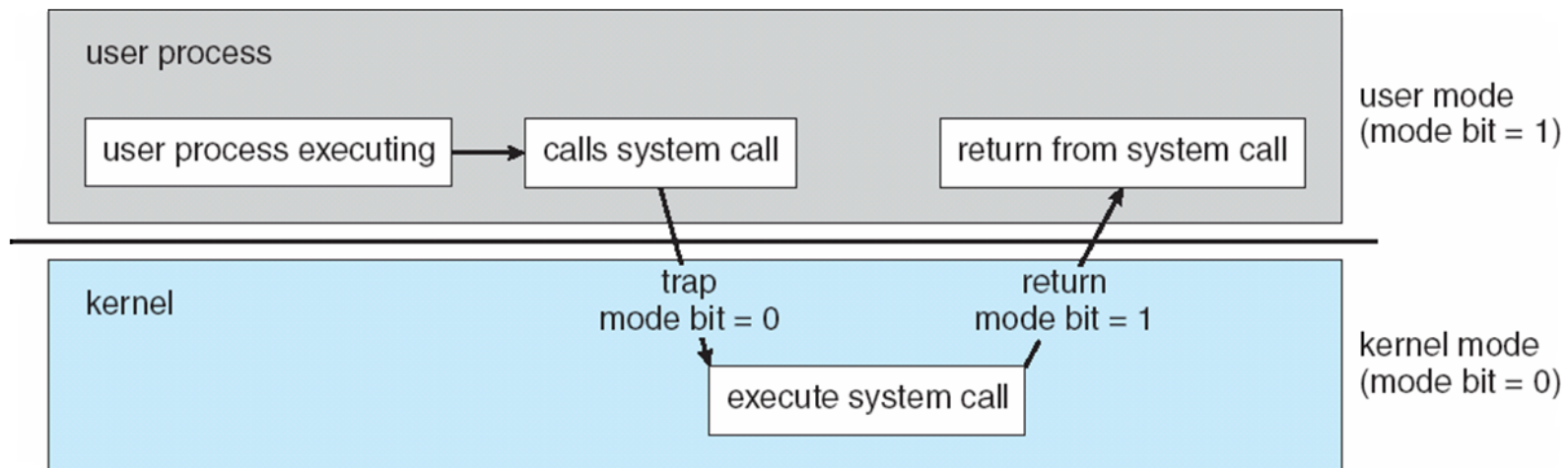
- **Dual-mode** operation allows OS to protect itself and other system components
  - **User mode** and **kernel mode**
  - **Mode bit** provided by hardware
    - ▶ Provides ability to distinguish when system is running user code or kernel code
    - ▶ Some instructions designated as **privileged**, only executable in kernel mode
    - ▶ System call changes mode to kernel, return from call resets it to user
  - Donanım tarafından sağlanan **mod biti (mode bit)**
    - ▶ Sistemin kullanıcı kodu mu yoksa çekirdek kodu mu çalıştırdığını ayırt etmekte kullanılır
    - ▶ Bazı komutlar **ayrıcalıklı** olarak tanımlıdır ve sadece çekirdek modunda çalıştırılabilirler
    - ▶ Sistem çağrılarını modu, çekirdek moduna çevirir.
    - ▶ Sistem çağrısı bittiğinde mod, kullanıcı moduna çevrilir





# Transition from User to Kernel Mode

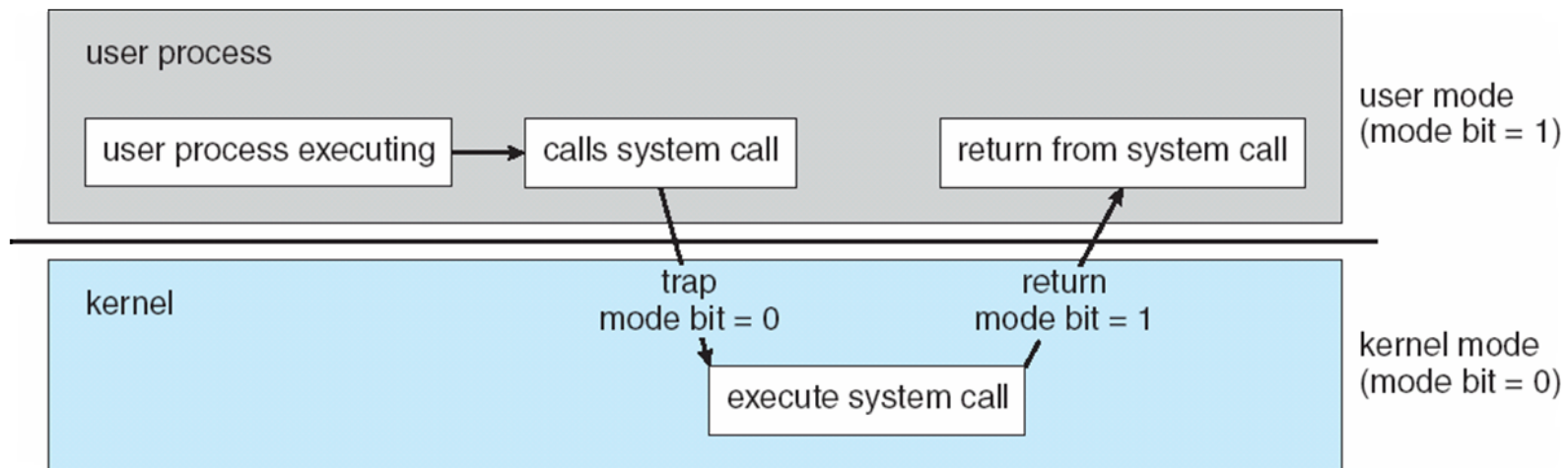
- Timer to prevent infinite loop / process hogging resources
  - Timer is set to interrupt the computer after some time period
  - Keep a counter that is decremented by the physical clock.
  - Operating system set the counter (privileged instruction)
  - When counter zero generate an interrupt
  - Set up before scheduling process to regain control or terminate program that exceeds allotted time





# Transition from User to Kernel Mode

- **Zamanlayıcı (timer)** sonsuz döngülere ve işlemci kilitlenmelerine engel olur
  - Belli bir zaman diliminden sonra kesme gönderilir
  - İşletim sistemi sayacı azaltır
  - Sayaç sıfırlandığında kesme oluşturulur
  - Zamanlayıcı program devreye girmeden sorun çıkaran işlem devre dışı bırakılır veya sonlandırılır





# Process Management

- A process is a program in execution. It is a unit of work within the system. Program is a ***passive entity***, process is an ***active entity***.
- Process needs resources to accomplish its task
  - CPU, memory, I/O, files
  - Initialization data(Başlangıç verisi)
- Process termination requires reclaim of any reusable resources
  
- **İşlem (process)** çalışmakta olan programdır
- Program *pasif* bir şeyken, işlem *aktif* bir şeydir
- İşlemler görevlerini yerine getirmek için kaynaklara ihtiyaç duyarlar
  - CPU, hafıza, I/O, dosyalar
  - Başlangıç verisi
- İşlemin sonlandırılması kullanılan kaynakların sisteme iade edilmesini gerektirir





# Thread Management

- Single-threaded process has one **program counter** specifying location of next instruction to execute
- Process executes instructions sequentially, one at a time, until completion
- Multi-threaded process has one program counter per thread
- Typically system has many processes, some user, some operating system running concurrently on one or more CPUs
- Concurrency by multiplexing the CPUs among the processes / threads

**İş parçacığı (thread)** bir program çalışırken aynı anda yapılması gereken başka işler varsa bunları çalıştırmak için kullanılır

Tek iş parçacıklı (single-threaded) işlemler, çalıştırılacak bir sonraki komutun hafızadaki konumunu belirten tek bir **program sayacına (program counter)** sahiptir

İşlem sonlanana kadar, komutları tek tek sırayla çalıştırır

**Çok iş parçacıklı (multi-threaded)** işlemler her bir iş parçacığı için ayrı bir program sayacına sahiptir

Tipik olarak sistemlerde, pek çok işlem, birkaç kullanıcı ve pek çok işletim sistemi işlemi aynı anda bir veya birden fazla işlemcide çalıştırılır

**Aynı anda kullanım (concurrency)** işlemcilerin birden fazla işlem veya iş parçacığı arasında ortak kullanımını gerektirir







# Process Management Activities

The operating system is responsible for the following activities in connection with process management:

- Creating and deleting both user and system processes
  - Suspending and resuming processes
  - Providing mechanisms for process synchronization
  - Providing mechanisms for process communication
  - Providing mechanisms for deadlock handling
- 
- Kullanıcı ve sistem işlemlerinin oluşturulması ve bitirilmesi
  - İşlemlerin duraklatılması ve devam ettirilmesi
  - İşlemlerin senkronizasyonu için mekanizmalar sağlaması
  - İşlemlerin birbiri ile iletişim kurabilmesi için mekanizmalar sağlaması
  - **Kilitlenmelerin (deadlock)** sağlıklı yönetilmesi için mekanizmalar sağlaması





# Memory Management

---

- To execute a program all (or part) of the instructions must be in memory
- All (or part) of the data that is needed by the program must be in memory.
- Memory management determines what is in memory and when
  - Optimizing CPU utilization and computer response to users
- Memory management activities
  - Keeping track of which parts of memory are currently being used and by whom
  - Deciding which processes (or parts thereof) and data to move into and out of memory
  - Allocating and deallocating memory space as needed

Tüm veriler bellekte: Hangi bölümler kullanılmakta, kime tahsis edildi, hangi veriler belleğe aktarılabilir, bellek tahsisi ve ayırma işlemleri





# Storage Management

---

- OS provides uniform, logical view of information storage
  - Abstracts physical properties to logical storage unit - **file**
  - Each medium is controlled by device (i.e., disk drive, tape drive)
    - ▶ Varying properties include access speed, capacity, data-transfer rate, access method (sequential or random)
- File-System management
  - Files usually organized into directories
  - Access control on most systems to determine who can access what
  - OS activities include
    - ▶ Creating and deleting files and directories
    - ▶ Primitives to manipulate files and directories
    - ▶ Mapping files onto secondary storage
    - ▶ Backup files onto stable (non-volatile) storage media





# Mass-Storage Management

---

- Usually disks used to store data that does not fit in main memory or data that must be kept for a “long” period of time
- Proper management is of central importance
- Entire speed of computer operation hinges on disk subsystem and its algorithms
- OS activities
  - Free-space management
  - Storage allocation
  - Disk scheduling
- Some storage need not be fast
  - Tertiary storage includes optical storage, magnetic tape
  - Still must be managed – by OS or applications
  - Varies between WORM (write-once, read-many-times) and RW (read-write)





# Performance of Various Levels of Storage

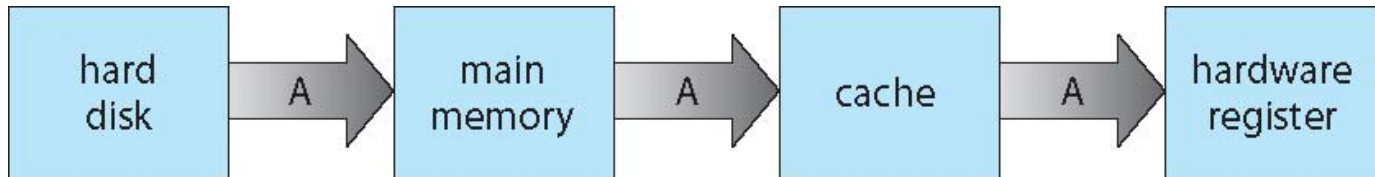
Level	1	2	3	4	5
Name	registers	cache	main memory	solid state disk	magnetic disk
Typical size	< 1 KB	< 16MB	< 64GB	< 1 TB	< 10 TB
Implementation technology	custom memory with multiple ports CMOS	on-chip or off-chip CMOS SRAM	CMOS SRAM	flash memory	magnetic disk
Access time (ns)	0.25 - 0.5	0.5 - 25	80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





# Migration of data “A” from Disk to Register

- Multitasking environments must be careful to use most recent value, no matter where it is stored in the storage hierarchy  
Çok işlemlili ortamlar, en güncel değeri kullanmak konusunda dikkatli olmalıdır (depolama hiyerarşisinin neresinde tutuluyorsa tutulsun)



- Multiprocessor environment must provide **cache coherency** in hardware such that all CPUs have the most recent value in their cache
- Çok işlemcili sistemlerde **ön bellek tutarlılığı** donanım seviyesinde sağlanmalı ve tüm işlemciler en güncel değere sahip olmalıdır





# I/O Subsystem

- One purpose of OS is to hide peculiarities of hardware devices from the user

İşletim sisteminin amaçlarından biri donanım cihazlarının karmaşıklıklarını kullanıcıdan gizlemektir

- I/O subsystem responsible for
  - Memory management of I/O including buffering (storing data temporarily while it is being transferred), caching (storing parts of data in faster storage for performance), spooling (the overlapping of output of one job with input of other jobs)
  - General device-driver interface
  - Drivers for specific hardware device
    - ▶ **Tampon bellek işlemleri (buffering)** – veriyi bir yerden diğer yere aktarırken geçici olarak saklamak
    - ▶ **Ön bellek işlemleri (caching)** – veriyi geçici olarak daha hızlı depolama birimine aktarmak
    - ▶ **Kuyruklama (spooling)** – bir işin çıktısını diğer işin girdisi haline getirmek





# Protection and Security

---

- **Protection** – any mechanism for controlling access of processes or users to resources defined by the OS
- **Security** – defense of the system against internal and external attacks
  - Huge range, including denial-of-service, worms, viruses, identity theft, theft of service
- Systems generally first distinguish among users, to determine who can do what
  - User identities (**user IDs**, security IDs) include name and associated number, one per user
  - User ID then associated with all files, processes of that user to determine access control
  - Group identifier (**group ID**) allows set of users to be defined and controls managed, then also associated with each process, file
  - **Privilege escalation** allows user to change to effective ID with more rights







# Koruma ve Güvenlik

- **Koruma (protection)** – İşlemlerin veya kullanıcıların herhangi bir kaynağa erişiminin işletim sistemi tarafından kontrol edilmesi
- **Güvenlik (security)** – sistemin içerden ve dışardan gelen saldırılara karşı savunulması
  - Geniş kapsamlı: DoS saldırıları, virusler, solucanlar, kimlik bilgileri hırsızlığı
- Sistemler öncelikle kullanıcıları, kimin ne yapabileceğine göre sınıflandırır
  - **Kullanıcı adı (user IDs, security IDs)** her kullanıcı için isim ve ilişkili numarayı içerir
  - Kullanıcı adı daha sonra, erişim kontrolü amacıyla, kullanıcının sahip olduğu tüm dosya ve işlemlerle ilişkilendirilir
  - **Grup adı (group ID)** da benzer şekilde bir grup kullanıcıyı belli işlem ve dosyalarla ilişkilendirmek ve erişim kontrolü sağlamak amacıyla kullanılır
  - **Ayrıcalık artışı**, kullanıcının daha fazla hakla etkin kimliğe geçmesine izin verir.





**Koruma (protection)** – İşlemlerin veya kullanıcıların herhangi bir kaynağa erişiminin işletim sistemi tarafından kontrol edilmesi

**Güvenlik (security)** – sistemin içerden ve dışardan gelen saldırılara karşı savunulması

Geniş kapsamlı: DoS saldırıları, virusler, solucanlar, kimlik bilgileri hırsızlığı

Sistemler öncelikle kullanıcıları, kimin ne yapabileceğine göre sınıflandırılır

**Kullanıcı adı (user IDs, security IDs)** her kullanıcı için isim ve ilişkili numarayı içerir

Kullanıcı adı daha sonra, erişim kontrolü amacıyla, kullanıcının sahip olduğu tüm dosya ve işlemlerle ilişkilendirilir

**Grup adı (group ID)** da benzer şekilde bir grup kullanıcıyı belli işlem ve dosyalarla ilişkilendirmek ve erişim kontrolü sağlamak amacıyla kullanılır

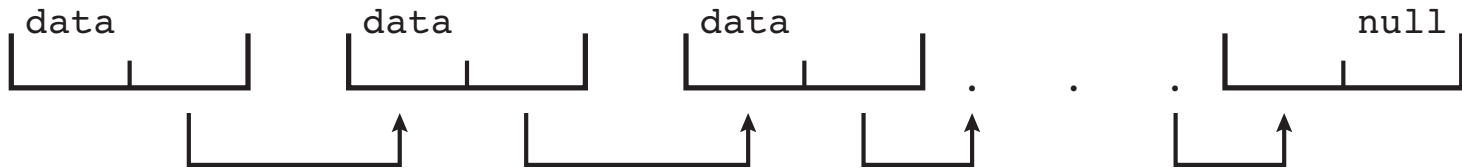




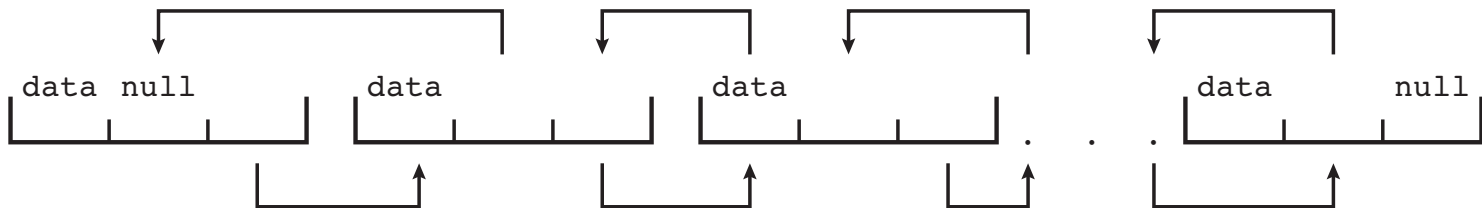
# Kernel Data Structures

- Many similar to standard programming data structures

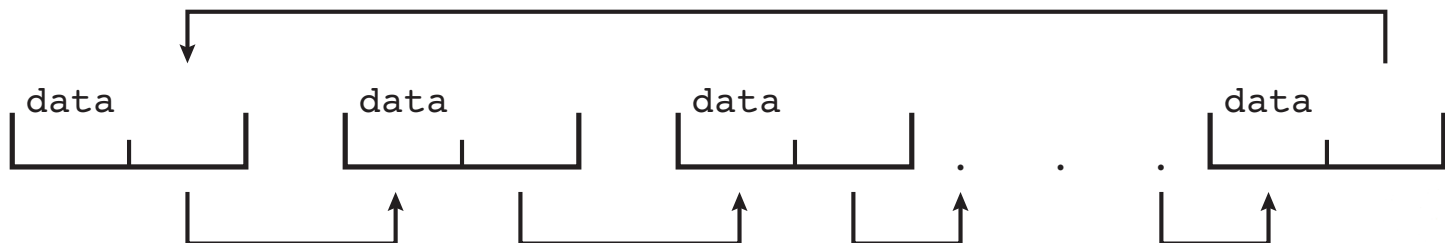
- ***Singly linked list***



- ***Doubly linked list***



- ***Circular linked list***



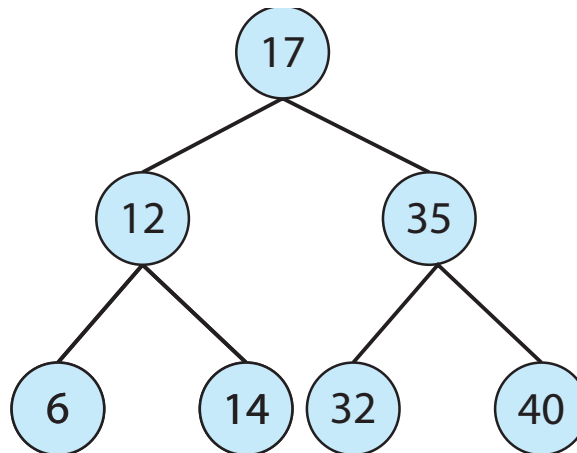


# Kernel Data Structures

## ■ Binary search tree

left  $\leq$  right

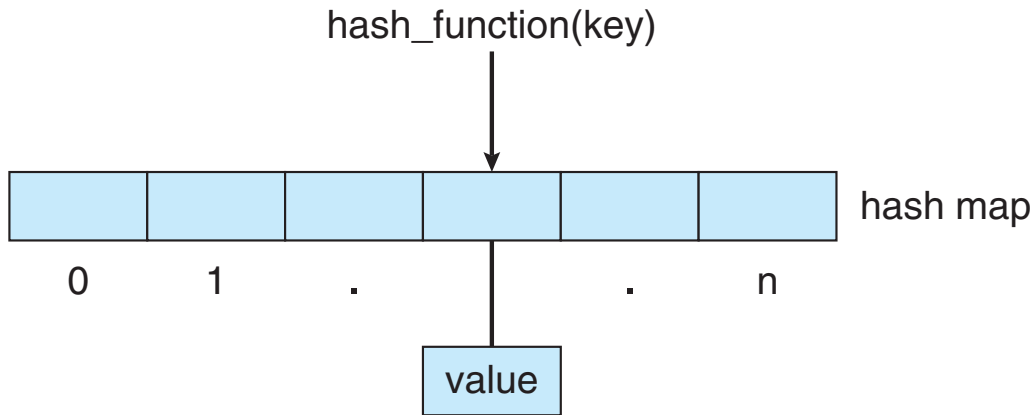
- Search performance is  $O(n)$
- **Balanced binary search tree** is  $O(\lg n)$





# Kernel Data Structures

- **Hash function** 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>`



# End of Chapter 1

---

