

## Genel bilgiler

### Ders içeriği

1. İşletim Sistemlerine Giriş
2. İşletim Sistem Yapıları
3. Process'ler
4. Thread'ler
5. Process Senkronizasyonu
6. CPU Scheduling
7. Deadlocks
8. Ana Bellek
9. Sanal Bellek
10. Kütük İşlemleri
11. Dosya Sistemi Arayüzü ve Uygulaması
12. I/O Sistemleri
13. Koruma ve Güvenlik

3

## Konular

- İşletim sistemi ne iş yapar?
- Bilgisayar sistemi organizasyonu
- Bilgisayar sistemi mimarisi
- İşletim sistemi yapısı
- İşletim sistemi işlemleri
- Process yönetimi
- Memory yönetimi
- Storage yönetimi
- Koruma ve güvenlik
- Kernel veri yapıları
- Hesaplama ortamları
- Açık kaynak işletim sistemleri

4

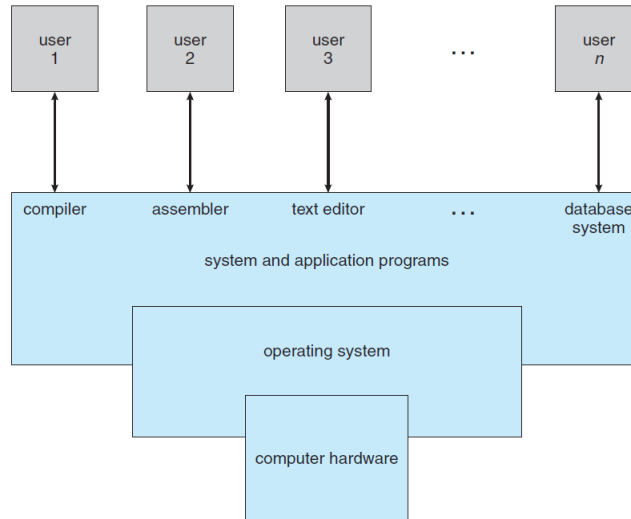
## İşletim sistemi ne iş yapar?

- **İşletim sistemi bilgisayar donanımını yöneten bir programdır.**
- Kullanıcı ile bilgisayar donanımı arasında aracı olarak görev yapar.
- Bazı işletim sistemlerinde (server) **etkinlik**, bazılarında ise **kullanılabilirlik** (mobil, client) önemlidir.
- **Bir bilgisayar sistemi genel olarak 4 bileşene ayrılabilir:**
  - Donanım
  - İşletim sistemi
  - Uygulama programları
  - Kullanıcılar
- **Donanım** (CPU, memory, I/O cihazları) temel kaynakları sağlar.
- **Uygulama programları** (kelime işlemciler, derleyiciler, Web tarayıcılar) bu kaynakların kullanıcı problemlerinde nasıl kullanılacağını belirler.

5

## İşletim sistemi ne iş yapar?

- **İşletim sistemi, farklı kullanıcılar için farklı kaynakların uygun kullanımını koordine eder.**



6

## İşletim sistemi ne iş yapar?

### *Kullanıcı açısından bakış*

- Kullanıcı bakışı **kullanılan arayüze bağlı olarak değişmektedir.**
- **Çoğu kullanıcı,** monitöre, klavyeye, fareye ve sistem birimine sahip bir **kişisel bilgisayar kullanır.**
- Bu durumda, **işletim sistemi** çoğunlukla **kolay kullanım için tasarlanır, kaynakların** nasıl paylaşıldığı ve **verimliliği,** çok kullanıcı yerine **tek kullanıcıya göre tasarlanır.**
- Diğer bir durumda ise, **çok sayıda kullanıcı bir ana bilgisayara bağlanır ve kaynakları paylaşırlar.**
- Burada, işletim sistemi **kaynak kullanım oranını maksimize edecek şekilde tasarlanır.**
- **Mobil cihazlar ve gömülü sistemler (ev cihazları, otomobil) için** tasarlanan işletim sistemlerinin de kendine özgü özellikleri vardır.

7

## İşletim sistemi ne iş yapar?

### *Sistem açısından bakış*

- Bilgisayar açısından işletim sistemi, **donanımla çok yakından ilişkiye sahip olan bir programdır.**
- Bu açıdan işletim sistemi, **kaynak** (CPU time, hafıza, depolama birimi, I/O cihazları, ...) **kullanımını planlayan programdır.**
- İşletim sistemi, **I/O cihazlarını ve kullanıcı programlarını kontrol eder ve** programların çalışması sırasındaki **hataları önlemeye yönelik işlemleri yönetir.**

8

## İşletim sistemi ne iş yapar?

### İşletim sistemi tanımı

- Bilgisayarların temel amacı, kullanıcı programlarının çalıştırılması ve kullanıcı problemlerinin kolay ve hızlı bir şekilde çözülmesidir.
- Bilgisayar donanımlarının tek başına kullanımı çok zordur, bu yüzden uygulama yazılımları geliştirilir.
- Yaygın kabul edilen tanımlamada, işletim sistemi bilgisayarda sürekli çalışan programdır ve kernel (çekirdek) olarak adlandırılır.
- Mobil işletim istemleri sadece kernel'a sahip değildir, middleware'e de sahiptirler.
- Middleware, uygulama geliştiricilere ek servisler sağlayan framework (platform) yazılımlarıdır.
- Apple iOS ve Google Android, middleware yazılımları ile veritabanı, multimedya ve grafik desteği sağlayan mobil işletim sistemleridir.

9

## Konular

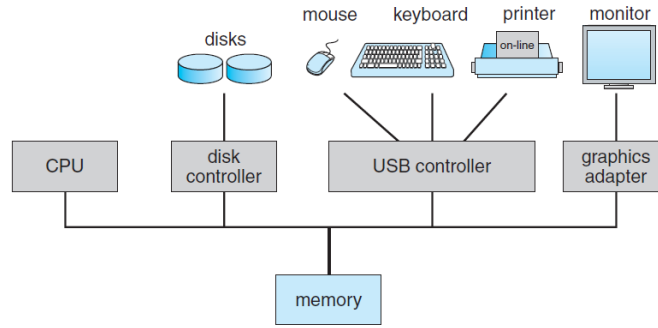
- İşletim sistemi ne iş yapar?
- Bilgisayar sistemi organizasyonu
- Bilgisayar sistemi mimarisi
- İşletim sistemi yapısı
- İşletim sistemi işlemleri
- Process yönetimi
- Memory yönetimi
- Storage yönetimi
- Koruma ve güvenlik
- Kernel veri yapıları
- Hesaplama ortamları
- Açık kaynak işletim sistemleri

10

## Bilgisayar sistemi organizasyonu

### Bilgisayar sisteminin çalışması

- Günümüz genel amaçlı bilgisayarları **bir veya birden fazla CPU'ya**, ortak bus üzerinden kontrol edilen **cihazlara** ve **paylaşılmış hafızaya sahiptir.**



- CPU ile cihaz denetleyicileri eş zamanlı çalışırlar ve paylaşılmış hafızaya aynı anda erişmek isteyebilirler.
- Hafıza denetleyicisi cihazların hafızaya erişimini yönetir.

11

## Bilgisayar sistemi organizasyonu

### Bilgisayar sisteminin çalışması

- Bilgisayar çalışmaya başladığında, **başlangıç programı** olarak **bootstrap programını kullanır.**
- Bootstrap programı oldukça basittir ve **ROM** (Read-Only Memory) veya **EEPROM** (Electrically Erasable Programmable Read-Only Memory) içerisinde saklanır.
- **Bootstrap programlarına firmware** adı verilir.
- Firmware programı bilgisayarın tüm bileşenlerini (CPU register, cihaz denetleyicileri, hafıza içeriği) başlatır.
- Bootstrap programı, **işletim sistemi kernel'inin bulunduğu konumu bilmek** ve **hafızaya yüklemek zorundadır.**
- Kernel hafızaya yüklendikten sonra sisteme ve kullanıcılara servis sağlamaya başlar.

12

## Bilgisayar sistemi organizasyonu

### Bilgisayar sisteminin çalışması

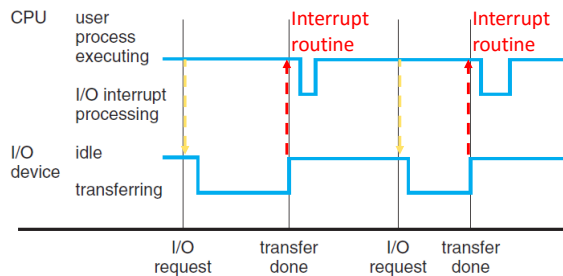
- Bazı servisler kernel dışındaki sistem programları (**system daemons, system processes**) tarafından sağlanır.
- UNIX üzerinde **init** ilk sistem prosesidir ve çok sayıda daemon başlatır. Bundan sonra sistem tümüyle **boot** edilmiş olur.
- İşletim sistemi **boot** edildikten sonra bir olay (**event**) gerçekleşmesi için beklemeye başlar.
- Bilgisayar sistemlerinde **bir olayın oluştuğu yazılım veya donanım tarafından interrupt** kullanılarak bildirilir.
- **Donanımlar**, CPU'ya bir sinyal ile interrupt bildirimi yapar.
- **Yazılımlar**, **system call** işlemlerini çalıştırarak interrupt başlatır.

13

## Bilgisayar sistemi organizasyonu

### Bilgisayar sisteminin çalışması

- CPU bir interrupt aldığı anda çalışmasını bulunduğu yerde keser ve **belirtilen diğer noktaya geçiş yapar**.
- Geçiş yaptığı yer, gelen kesmeyle ilişkilendirilmiş **service routine**'nin başlangıç adresidir.
- Interrupt'a ait **service routine bittiğinde ise önceki yere geçiş yaparak çalışmasına devam eder**.



14

## Bilgisayar sistemi organizasyonu

### Bilgisayar sisteminin çalışması

- **Interrupt routine adresleri pointer ile gösterilir.**
- Bu pointer'lar tablo halinde (**interrupt vector**) hafızanın 0-100 adresleri aralığında saklanır.
- **Interrupt pointer'ları her cihaz için ayrı adresi gösterir** ve interrupt gönderen cihazın routine'inin adresini sağlarlar.
- Interrupt geldiğinde çalışan komutun (**instruction**) adresinin ve **CPU'nun konfigürasyonunun (register değerleri)** saklanması gereklidir (**context switch**).
- Komut adresi ve CPU konfigürasyonu **stack (yığın)** üzerinde saklanmaktadır.

15

## Bilgisayar sistemi organizasyonu

### Depolama yapısı

- **CPU, programlara ait komutları hafızadan yükler.** Bu yüzden, çalışacak programların önce hafızaya alınması gereklidir.
- **Genel amaçlı bilgisayarlar programları çalıştırmak için ana hafızayı (main memory) kullanır.**
- Ana hafıza, **RAM (random-access memory)** olarak da adlandırılır.
- Ana hafıza yarı iletken teknolojisi kullanılarak oluşturulur (**DRAM-dynamic RAM**).
- **von Neumann mimarisi'**ne sahip sistemlerde, komutlar **fetch** ile (**hafızadan CPU içerisindeki register'a alınması**) çalıştırmaya başlanır.
- Fetch aşaması sonucunda komut **instruction register'a** alınmış olur.
- Komut çözülür, çalıştırılır ve **sonucu (varsa) hafızaya/reg aktarılır.**

16

## Bilgisayar sistemi organizasyonu

### Depolama yapısı

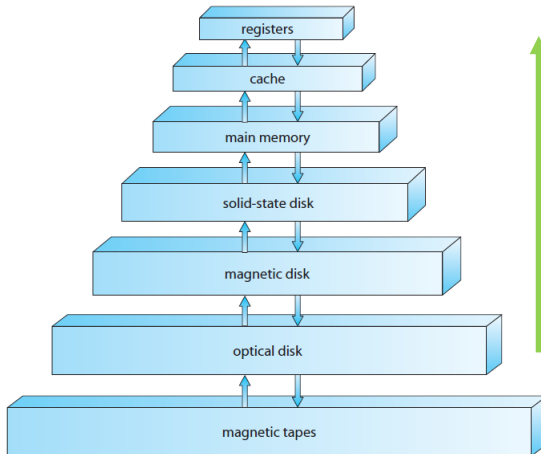
- **Tüm programların hafızada saklanması istenir** ancak iki neden dolayı mümkün değildir:
  - **Hafıza sınırlı kapasiteye sahiptir.** Veri ve programlar çok büyük boyuttadır.
  - **Enerji kesildiğinde hafızadaki veri kaybolur (volatile).**
- **Tüm bilgisayar sistemleri, programları ve verileri kalıcı saklamak için ikincil depolama (secondary storage) birimlerine sahiptir.**
- En yaygın kullanılan ikincil depolama birimi manyetik disklerdir (magnetic disk, hdd).
- Programlar hafızaya yüklenmeden önce manyetik disklerde tutulur.
- Veri saklama birimleri arasında, **hız, maliyet, boyut ve saklamanın kalıcılığı** açısından farklılıklar vardır.

17

## Bilgisayar sistemi organizasyonu

### Depolama yapısı

- Depolama birimleri arasında hiyerarşik bir ilişki vardır.



- Yukarı çıktıkça erişim hızı artar.
- Yukarı çıktıkça bit başına saklama maliyeti artar.
- Yukarı çıktıkça toplam kapasite azalır.
- Yukarı çıktıkça CPU tarafından kullanılma sıklığı artar.

18



## Bilgisayar sistemi organizasyonu

### Depolama yapısı

- **Main memory, cache ve register'lar** veriyi geçici saklama (**volatile**) birimleridir.
- **Solid-state disk, magnetic disk, optical disk ve magnetic tape** **kalıcı saklama (nonvolatile)** birimleridir.
- Solid-state disklerin farklı versiyonları vardır:
  - **DRAM ile manyetik disk birlikte kullanılır.** Normal işlem sırasında **DRAM** kullanılır daha sonra manyetik diske aktarma yapılır (dahili batarya kullanılır).
  - DRAM'den daha yavaş **flash memory** kullanılır (dahili batarya gerektirmez).
  - **NVRAM'de (Nonvolatile RAM)** ise **DRAM** ile **batarya** kullanılır ve kalıcı saklama yapılır.

19

## Bilgisayar sistemi organizasyonu

### I/O yapısı

- İşletim sistemleri kodunun büyük bir bölümü I/O yönetimine ayrılır.
- **I/O cihazlarında güvenilirlik (reliability) ve performans çok önemlidir.**
- Genel amaçlı bilgisayarlarda CPU ile çok sayıda I/O cihazı bus üzerinden bağlantıya sahiptir.
- Denetleyiciye bağlı olarak birden fazla cihaz bağlanabilir.
- **SCSI (small computer-systems interface)** denetleyiciye 7 veya daha fazla cihaz bağlanabilir.
- İşletim sistemi her cihaz denetleyicisi için cihaz sürücüsüne (**device driver**) sahiptir.

20

## Bilgisayar sistemi organizasyonu

### I/O yapısı

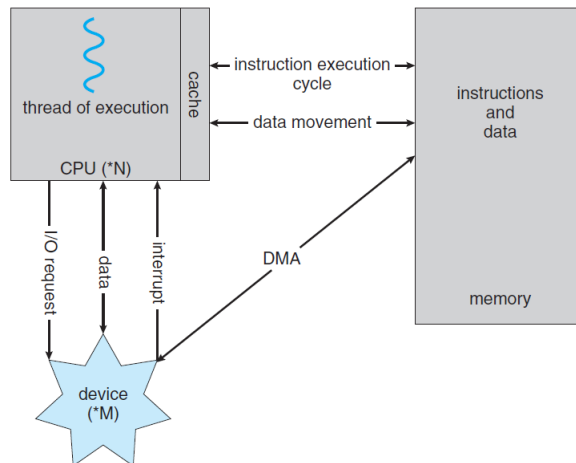
- I/O işlemini başlatmak için **device driver** uygun **register içeriğini device controller'a** aktarır.
- İşlemin tamamlandığı **device controller** tarafından **device driver'a interrupt** ile bildirilir.
- Bu şekilde veri aktarımında **overhead** fazladır ve aktarım işlemi yavaştır.
- **DMA (direct memory access)** ile blok veri cihaz ile hafıza arasında doğrudan aktarılır.
- DMA kullanıldığında I/O cihazı için buffer, pointer'lar ve sayıcılar oluşturulur ve device controller tüm aktarımı gerçekleştirilir.
- İşlemin bittiği device driver'a interrupt ile bildirilir. **Bu işlem süresince CPU diğer işleri gerçekleştirir.**

21

## Bilgisayar sistemi organizasyonu

### I/O yapısı

- Modern bilgisayar sistemleri DMA ile veri aktarımı yapar ve paylaşılmış bus kullanılmaz.



22

## Konular

- İşletim sistemi ne iş yapar?
- Bilgisayar sistemi organizasyonu
- **Bilgisayar sistemi mimarisi**
- İşletim sistemi yapısı
- İşletim sistemi işlemleri
- Process yönetimi
- Memory yönetimi
- Storage yönetimi
- Koruma ve güvenlik
- Kernel veri yapıları
- Hesaplama ortamları
- Açık kaynak işletim sistemleri

23

## Bilgisayar sistemi mimarisi

### *Tek işlemcili sistemler*

- Tek işlemciye sahip sistemde, **komut kümesindeki tüm komutlar bir işlemci tarafından çalıştırılmaktadır.**
- Bu sistemler disk, klavye, grafik denetleyici gibi bileşenlere sahiptir.
- **Tek işlemcili sistemlerin yönetimi** (sonraki görevin bildirilmesi, durumun izlenmesi) **işletim sistemi tarafından yapılmaktadır.**
- **Bu sistemler I/O cihazlarına özel işlemcilere de sahip olabilmektedir.**
- Örneğin, disk denetleyici işlemcisi, ana CPU'dan gelen isteklerin kuyruk yönetimi ve planlamasını gerçekleştirir.
- **Diğer sistemlerde** bu tür **cihaz işlemcileri donanımların içerisinde yer almaktadır** ve işletim sistemi bu işlemcilerle haberleşmez.
- **Genel amaçlı tek işlemciye sahip olan sistemler tek işlemcili olarak adlandırılır.**

24

## Bilgisayar sistemi mimarisi

### Çok işlemcili sistemler

- Son birkaç yıldır **çok işlemcili sistemler** (**multiprocesser systems, parallel systems, multicore systems**) yoğun kullanılmaya başlanmıştır.
- **Çok işlemcili sistemler, iki veya daha fazla CPU'ya sahiptir ve bus, clock, memory ve çevre birimlerini paylaşırlar.**
- Çok işlemcili sistemler **önce sunucu sistemlerinde** kullanılmıştır.
- **Daha sonra masaüstü ve dizüstü bilgisayarlarda** kullanılmıştır.
- **Son yıllarda ise mobil cihazlarda** kullanılmaya başlanmıştır.

25

## Bilgisayar sistemi mimarisi

### Çok işlemcili sistemler

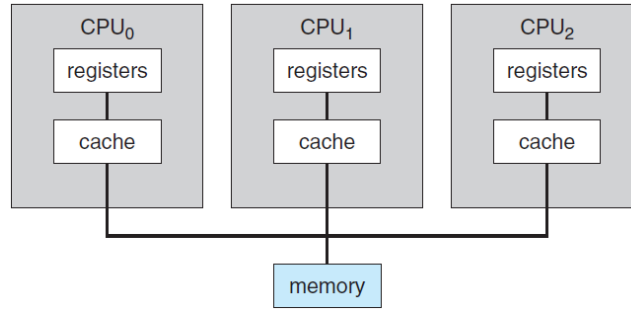
- Çok işlemcili sistemlerin temel olarak 3 avantajı vardır:
  - **Yüksek throughput:** İşlemci sayısı arttıkça daha kısa sürede daha fazla iş yapılır.  
**Hızlanma oranı işlemci sayısı ile doğru orantılı değildir!**
  - **Ekonomik ölçeklendirme:** Aynı sayıda tek işlemcili sisteme göre **daha ekonomiktir.** Çevre birimlerini, depolama birimlerini ve güç birimlerini paylaşırlar.
  - **Yüksek güvenilirlik:** Bir işlemcide oluşan hata sistemin tümünü çalışmaz hale getirmez. **Performans azalır! (graceful degradation)**
- Asimetrik çok işlemcili sistemlerde (**AMP-asymmetric multiprocessing**), **her işlemci bir işe atanmıştır** ve tüm işlemciler başka bir işlemci tarafından denetlenir.
- Simetrik çok işlemcili sistemlerde (**SMP-symmetric multiprocessing**), **her işlemci işletim sistemindeki tüm işleri yapabilir.** Tüm işlemciler eş düzey (**peer**) olarak çalışır.

26

## Bilgisayar sistemi mimarisi

### Çok işlemcili sistemler

- SMP mimarisinde, **her CPU kendi register'larına ve lokal cache'e sahiptir** ancak **hafıza paylaşılmaktadır**.



- **Windows, Mac OS X ve Linux** işletim sistemleri SMP mimarisini desteklemektedir.

27

## Bilgisayar sistemi mimarisi

### Çok işlemcili sistemler

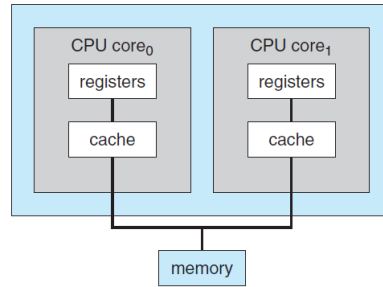
- RAM'e erişim süresi tüm işlemcilerde aynı olan modele **UMA (uniform memory access)** denilmektedir.
- RAM'e erişim süresi tüm işlemcilerde farklı olan modele **NUMA (nonuniform memory access)** denilmektedir.
- İşletim sistemi, kaynak yönetimi ile NUMA'nın erişim süresi dezavantajını ortadan kaldırabilir.

28

## Bilgisayar sistemi mimarisi

### Çok işlemcili sistemler

- Son yıllarda **bir chip üzerinde birden fazla işlemci (multicore) kullanılmaktadır.**
- **Multicore sistemler** birden fazla chip'e sahip çok işlemcili sistemlere göre **daha hızlıdır ve daha az enerji tüketirler.**
- Her core kendi **register'larına ve önbelleğine sahiptir, ancak hafızayı paylaşırlar.**



29

## Bilgisayar sistemi mimarisi

### Çok işlemcili sistemler

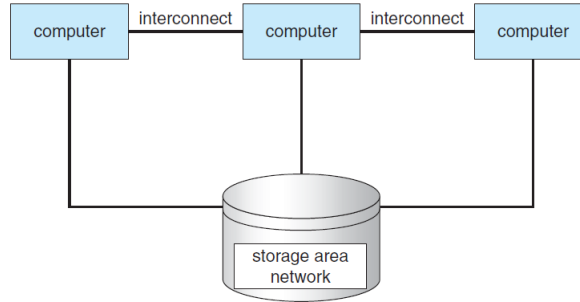
- **Blade sunucular**, çok işlemci board'ları, I/O board'ları ve ağ bağlantı board'larını **aynı kasada bulundurlar.**
- Her blade işlemci board'u ayrı boot yapılır ve kendi işletim sistemini çalıştırır.
- Bazı blade sunucularda, birden fazla çok işlemcili board kullanılabilir.

30

## Bilgisayar sistemi mimarisi

### Kümelenmiş (clustered) sistemler

- **Clustered sistemler** bağımsız iki veya daha fazla sistemden oluşurlar.
- Bu sistemler, depolama birimlerini paylaşırlar ve **LAN (local area network)** üzerinden haberleşirler.
- Bu sistemler **loosely coupled** (gevşek bağlı) olarak adlandırılırlar.



31

## Bilgisayar sistemi mimarisi

### Kümelenmiş (clustered) sistemler

- **Clustered sistemler, high-availability sağlarlar.**
- Her node, bir veya birkaç node'u izler hata oluşması durumunda o node'un görevlerini üstlenir.
- **Asymmetric clustering** yapısında, **bir sistem aktif çalışır diğeri bekleme modundadır (hot-standby mode)** ve çalışan sistemi izler. Hata olması halinde aktif çalışmaya başlar.
- **Symmetric clustering** yapısında, iki veya daha fazla sistem aktif olarak uygulamaları çalıştırır ve birbirlerini izlerler.

32

## Bilgisayar sistemi mimarisi

### Kümelenmiş (clustered) sistemler

- **Clustered sistemlerde**, bir program parçalara bölünerek eş zamanlı çalıştırılabilir (**parallelization**).
- Her sistemden elde edilen sonuçlar birleştirilerek sonuç çözüm elde edilir.
- Diğer clustered yapısında ise sistemler arasında iletişim **WAN (wide-area network)** üzerinden sağlanır.
- Bu sistemlerde işlem yapılan veride çakışmayı önlemek için **dağıtık kilitleme yönetimi (DLM-distributed lock manager)** yapılır.

33

## Konular

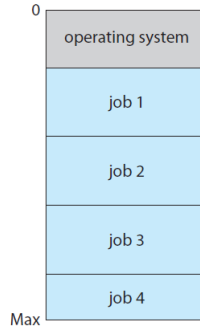
- İşletim sistemi ne iş yapar?
- Bilgisayar sistemi organizasyonu
- Bilgisayar sistemi mimarisi
- **İşletim sistemi yapısı**
- İşletim sistemi işlemleri
- Process yönetimi
- Memory yönetimi
- Storage yönetimi
- Koruma ve güvenlik
- Kernel veri yapıları
- Hesaplama ortamları
- Açık kaynak işletim sistemleri

34



## İşletim sistemi yapısı

- İşletim sistemi, programların çalıştırılması için ortam sağlamaktadır.
- İşletim sistemleri birden çok programı çalıştırabilir (multiprogramming).
- Multiprogramming çalışabilen işletim sistemi çok sayıda işi aynı anda hafızada tutar.
- Tüm işler disk üzerindeki job pool içinde tutulur.)



35

## İşletim sistemi yapısı

- Multiprogramming işletim sistemi bir işi alır ve çalıştırmaya başlar.
- Çalışan işte **bekleme olduğunda** başka bir işe geçiş yaparak çalışmaya devam eder.
- **Multitasking (time sharing)** işletim sistemlerinde CPU işler arasında çok hızlı geçişler yapar. (Geçiş için işte bekleme oluşması gerekli değildir.)
- Multitasking işletim sistemlerinde kullanıcı herhangi bir iş ile etkileşime geçebilir. **Tepki süresinin çok kısa olması gereklidir!**
- Hafızaya yüklenen ve çalıştırılmakta olan programa **process** denilir.
- Eğer hafızada ayrılan yerden daha çok sayıda iş hafızaya alınmak için hazır ise, **hafızaya alınacak olanı seçmeye job scheduling** denir.
- Aynı anda hafızada birden fazla iş hazır ise, **hangisinin ilk önce çalışacağına karar vermeye CPU scheduling** denilmektedir.

36

## Konular

- İşletim sistemi ne iş yapar?
- Bilgisayar sistemi organizasyonu
- Bilgisayar sistemi mimarisi
- İşletim sistemi yapısı
- **İşletim sistemi işlemleri**
- Process yönetimi
- Memory yönetimi
- Storage yönetimi
- Koruma ve güvenlik
- Kernel veri yapıları
- Hesaplama ortamları
- Açık kaynak işletim sistemleri

37

## İşletim sistemi işlemleri

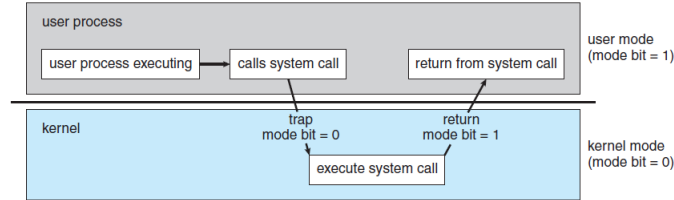
- **Modern işletim sistemleri, kesilmelerle yönetilirler (interrupt driven).**
- **Eğer çalışan process yoksa**, hiçbir I/O cihazı servis sağlamıyorsa, kullanıcılardan etkileşim yoksa, **işletim sistemi bekleme durumundadır** ve hiçbir iş yapmaz.
- Bir **trap** (veya **exception**), yazılım tarafından üretilen **interrupt**'tır ve işletim sisteminin iş gerçekleştirmesini sağlar.
- **Bir işletim sisteminde çalışan programlardan birisi hata ürettiğinde sadece o programın etkilenmesi istenir.**
- Ancak, bazı durumlarda **diğer programların çalışma hızı etkilenebilir, verileri değişebilir** veya **işletim sisteminin kendisi bile çalışmaz hale gelebilir.**
- **İyi tasarlanmış işletim sistemleri** bu şekilde hatalı programların **(malicious) diğerlerini etkilemesini engeller.**

38

## İşletim sistemi işlemleri

### Dual mode ve multimode işlem

- İşletim sisteminin doğru çalışmasını sağlamak için, **işletim sistemi kodu ile kullanıcı programının kodunun ayırt edilmesi gereklidir.**
- **Mode bit** kullanılarak, kullanıcı modu (**user mode = 1**) ve kernel modu (**supervisor, system, privileged = 0**) ayrımı (**dual mode**) yapılabilirler.



- Sistem boot edildiğinde kernel moddadır ve uygulama programı çalışmaya başlayınca user moda geçer.
- **Birden fazla bit kullanılarak multimode oluşturulabilir** (test mode, ...).

39

## İşletim sistemi işlemleri

### Timer

- Bir kullanıcı programının **sonsuz döngüye girmesi** veya hata oluşması durumunda, **sistem servislerini çağırnamaması sonucunda işletim sistemine dönülemez.**
- Bu sorunu gidermek için **timer** kullanılır.
- Timer her programa geçildiğinde set edilir ve aşağıya doğru sayar.
- **Timer 0 değerine ulaşınca kontrol işletim sistemine alınır.**
- İşletim sistemleri her program için belirlenen **timer süresini sabit veya değişken alabilmektedirler.**

40

## Konular

- İşletim sistemi ne iş yapar?
- Bilgisayar sistemi organizasyonu
- Bilgisayar sistemi mimarisi
- İşletim sistemi yapısı
- İşletim sistemi işlemleri
- **Process yönetimi**
- Memory yönetimi
- Storage yönetimi
- Koruma ve güvenlik
- Kernel veri yapıları
- Hesaplama ortamları
- Açık kaynak işletim sistemleri

41

## Process yönetimi

- CPU tarafından çalıştırılmakta olan program **process** olarak adlandırılır.
- Bir **process** yapması gereken işi tamamlamak için, **CPU süresine**, **hafızaya**, **dosyalara**, **I/O cihazlarına** ihtiyaç duyar.
- Process çalıştığı sürece bu kaynaklardan ihtiyaç duyduğunu kullanır.
- Çalışması sonlanınca işletim sistemi ayrılmış kaynakları serbest bırakır.
- Bir **program** pasif varlıktır (**passive entity**), bir **process** ise aktif varlıktır (**active entity**).
- **Program counter (PC)**, CPU içerisinde **register**'dir ve sonraki çalıştırılacak komutun adresini tutar.
- **Single-threaded process** bir PC'ye sahiptir, **multithreaded process** birden çok PC'ye sahiptir.

42

## Process yönetimi

### ■ Bir işletim sistemi process yönetiminde aşağıdaki işlerden sorumludur:

- CPU üzerindeki process ve thread'lerin zamanlaması,
- Kullanıcı ve sistem process'lerinin oluşturulması ve silinmesi,
- Process'lerin askıya alınması ve devam ettirilmesi,
- Process'lerin senkronizasyonu,
- Process'lerin haberleşmesi.

43

## Konular

- İşletim sistemi ne iş yapar?
- Bilgisayar sistemi organizasyonu
- Bilgisayar sistemi mimarisi
- İşletim sistemi yapısı
- İşletim sistemi işlemleri
- Process yönetimi
- **Memory yönetimi**
- Storage yönetimi
- Koruma ve güvenlik
- Kernel veri yapıları
- Hesaplama ortamları
- Açık kaynak işletim sistemleri

44

## Memory yönetimi

- Hafıza, modern işletim sistemlerinde işlemlerin gerçekleşmesinde temel elemandır.
- **CPU, tüm programları hafıza üzerinden çalıştırır.**
- CPU, disk üzerindeki bir program parçasına ihtiyaç duyduğunda, I/O çağrısı ile önce hafızaya aktarır.
- Genel amaçlı bilgisayarlar **CPU verimliliğini artırmak** için birden **çok programı hafızada tutarlar** ve hafıza yönetimi gerçekleştirirler.
- İşletim sistemi **hafıza yönetiminde aşağıdaki işlerden sorumludur:**
  - **Hafızanın hangi kısmının kullanıldığının** ve kimin tarafından kullanıldığının izlenmesi,
  - **Hangi process'in** (veya process parçasının) **hafızaya alınacağına** veya hafızadan atılacağına **karar verilmesi**,
  - Hafızadaki **boş alanların tahsis edilmesi** veya serbest bırakılması.

45

## Konular

- İşletim sistemi ne iş yapar?
- Bilgisayar sistemi organizasyonu
- Bilgisayar sistemi mimarisi
- İşletim sistemi yapısı
- İşletim sistemi işlemleri
- Process yönetimi
- Memory yönetimi
- **Storage yönetimi**
- Koruma ve güvenlik
- Kernel veri yapıları
- Hesaplama ortamları
- Açık kaynak işletim sistemleri

46

## Storage yönetimi

### *Dosya sistemi yönetimi*

- Her depolama birimi, hız, kapasite, veri aktarım oranı ve erişim yöntemi gibi farklı özelliklere sahiptir.
- İşletim sistemi, **depolama biriminin özelliklerini soyutlamak için** mantıksal depolama birimi olarak **file (dosya) tanımlar**.
- Dosyalar, sayısal, alfabetik, alfanümerik veya binary veri bulundurabilir.
- Dosyaya birden fazla kullanıcı erişebilir. **Her kullanıcı için erişim denetiminin (okuma, yazma, ekleme) yapılması gereklidir.**
- **İşletim sistemi dosya yönetiminde aşağıdaki işlerden sorumludur:**
  - Dosya oluşturma ve silme,
  - Dizin oluşturma ve silme,
  - Dosya ve izin manipülasyon işlemleri.

47

## Storage yönetimi

### *Mass-storage yönetimi*

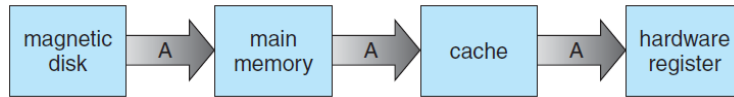
- **Tüm programlar hafızaya alınmadan önce disk üzerinde saklanır.**
- Disk yönetimi işletim sisteminin temel görevlerindendir.
- Manyetik tape, CD ve DVD sürücüler, **üçüncül (tertiary) depolama cihazlarıdır.**
- Bu cihazlar, **WORM (write-once, read-many-times)** veya **RW (read-write)** olarak farklı özelliklere sahip olabilirler.
- **İşletim sistemi disk yönetiminde aşağıdaki işlerden sorumludur:**
  - Boş alan yönetimi,
  - Depolama alanı tahsisi,
  - Disk kullanım zamanlaması.

48

## Storage yönetimi

### Cache bellek

- Cache bellek, hafızadan daha hızlı ve CPU'ya daha yakın saklama birimidir.
- CPU, bir veriye ihtiyaç duyduğunda hafızadan alır ve bir kopyasını cache bellek üzerine aktarır.
- Tekrarlı isteklerde cache bellek üzerindeki kullanır.



- Cache bellek kapasitesi çok küçük olduğundan yönetimi çok önemlidir.
- Cache bellekte tutulacak veya atılacak verilerin belirlenmesi için replacement algoritmaları kullanılır.

49

## Storage yönetimi

### Cache bellek

- Cache bellekler hafızadan çok küçük kapasiteye sahip olan ancak register'lardan daha fazla kapasiteye sahip olan depolama birimleridir.
- Cache belleklerde erişim adrese göre, register'larda isme göre yapılır.

| 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     |

- Birden fazla işlemcili sistemlerde cache tutarlılığının (cache coherence) sağlanması zorunludur.

50



## Storage yönetimi

### I/O sistemleri

- İşletim sistemlerinin amaçlarından birisi de **donanımların özelliklerinden kullanıcıyı soyutlamaktır.**
- Sadece **device driver** kendisine atanmış olan **cihazın özelliklerini bilir.**
- **I/O sistemi aşağıdaki bileşenlere sahiptir:**
  - Hafıza yönetim bileşeni (buffering, caching ve spooling),
  - Device driver arayüzü,
  - Donanımlar için driver.

51

## Konular

- İşletim sistemi ne iş yapar?
- Bilgisayar sistemi organizasyonu
- Bilgisayar sistemi mimarisi
- İşletim sistemi yapısı
- İşletim sistemi işlemleri
- Process yönetimi
- Memory yönetimi
- Storage yönetimi
- **Koruma ve güvenlik**
- Kernel veri yapıları
- Hesaplama ortamları
- Açık kaynak işletim sistemleri

52

## Koruma ve güvenlik

- Bir bilgisayar sistemi **birden fazla kullanıcının erişimine açıksa ve birden çok işlemcinin eş zamanlı işlemine izin veriyorsa, verilere erişimin düzenlenmesi zorunludur.**
- İşlemciye, dosyalara, hafıza segment'lerine ve diğer kaynaklara sadece **yetkisi olan processlerin erişimine izin verilmelidir.**
- Bilgisayar sistemindeki kaynaklara kullanıcıların veya process'lerin **erişiminin denetlenmesine koruma (protection) denilmektedir.**
- **Protection oluşabilecek hataları arayüzde iken algılar ve sistemin güvenilirliğini artırır.**
- Koruma altındaki bir sistem yetkili ve yetkisiz kullanıcıları birbirinden ayırt eder.

53

## Koruma ve güvenlik

- **Bir sistem yeterli korumaya sahip olsa da hatalara ve uygun olmayan erişimlere elverişli olabilir.**
- Örneğin, sisteme erişim yetkisi olan bir **kişinin bilgileri çalınabilir ve bilgileri silinebilir, kopyalanabilir, dosyaları kalıcı hasara uğrayabilir.**
- **Güvenlik (security), bir sistemi dışarıdan veya içeriden saldırılara karşı korumayı amaçlar.**
- Bu saldırılar, **virüsler, worm'lar, DoS, ...** gibi çok farklı şekillerde olabilir.
- **Bu saldırılardan korunmayı bazı işletim sistemleri görev olarak düşünürken, bazı işletim sistemleri bu işleri diğer yazılımlara bırakır.**
- Protection ve security, sistemdeki **tüm kullanıcıların birbirinden ayırt edilebilmesini gerektirir.**
- Çoğu işletim sistemi bunu **kullanıcı kimlikleri (user ID) ile yapar.**
- Grup ID veya sistem yöneticisi (**admin**) tanımlamaları da yapılabilir.

54

## Konular

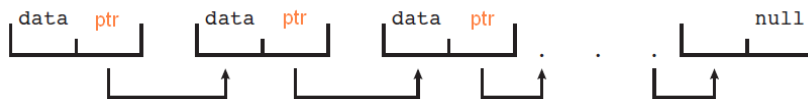
- İşletim sistemi ne iş yapar?
- Bilgisayar sistemi organizasyonu
- Bilgisayar sistemi mimarisi
- İşletim sistemi yapısı
- İşletim sistemi işlemleri
- Process yönetimi
- Memory yönetimi
- Storage yönetimi
- Koruma ve güvenlik
- **Kernel veri yapıları**
- Hesaplama ortamları
- Açık kaynak işletim sistemleri

55

## Kernel veri yapıları

### Bağlı listeler

- Bir **dizi (array)** basit bir veri yapısıdır ve **elemanlara doğrudan erişim sağlar**.
- Dizilerde **elemanların boyutları sabittir** ve bir elemana doğrudan erişmek için önündeki eleman sayısı ile bir elemanın boyutu çarpılır.
- Ancak çoğu uygulamalarda **elemanlar farklı boyutlarda olabilir**.
- **Bu durumda, bağlı listeler (linked lists) kullanılır.**
- Bağlı listeler, bir grup veriyi art arda sıralı halde tutar ve doğrudan erişime olanak sağlar.

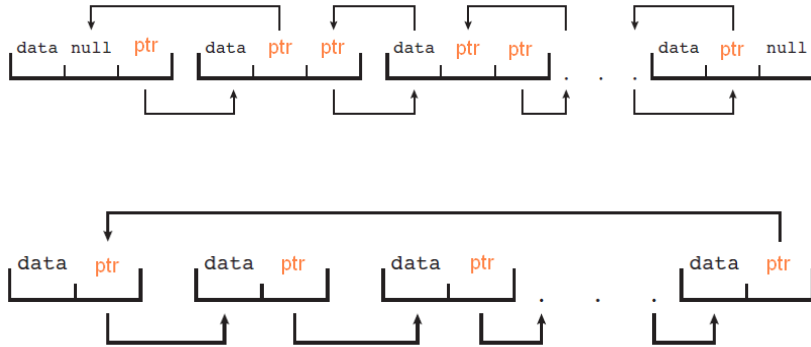


56

## Kernel veri yapıları

### Bağlı listeler

- Bağlı listeler bir bağlı (*singly linked list*), iki bağlı (*doubly linked list*) veya dairesel bağlı (*circularly linked list*) olabilir.

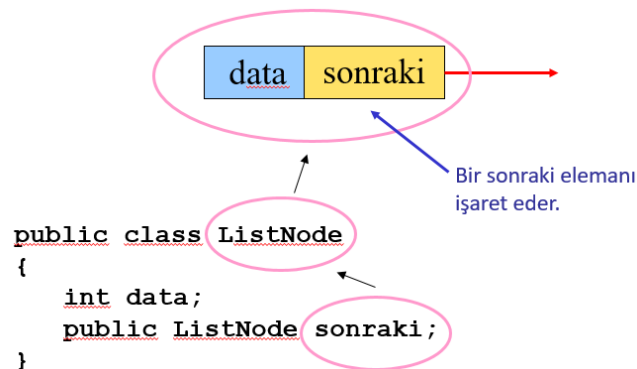


57

## Kernel veri yapıları

### Bağlı listeler

- Bağlı listelerde bir elemana erişim performansı  $O(n)$ 'dir (*worst case*).
- Tek bağlı listenin tanımı aşağıdaki gibi yapılabilir.



58

## Kernel veri yapıları

### Yığın

- Yığınlar (**stack**) son gelen ilk çıkar (**last in first out - LIFO**) şeklinde çalışan veri yapılarıdır.
- Bir stack üzerine yeni eleman eklemek için **push**, bir stack üzerinden son eklenen elemanı almak için **pop** işlevleri kullanılır.
- İşletim sistemleri, iç içe fonksiyon çağırımlarında stack yapısını kullanır.
- Yeni fonksiyona geçiş yaparken ve geri dönerken, interrupt altıyordamına geçiş yaparken ve geri dönerken, parametreler, lokal değişkenler ve dönüş adresi stack üzerine saklanır.

59

## Kernel veri yapıları

### Kuyruk

- Kuyruklar (**queue**) ilk gelen ilk çıkar (**first in first out - FIFO**) şeklinde çalışan veri yapılarıdır.
- Bir kuyruk üzerine yeni eleman eklendiğinde en sona kaydedilir, bir kuyruk üzerinden eleman alındığında en baştaki alınır.
- İşletim sistemleri, yazıcıya iş gönderirken, işlemci tarafından çalıştırılmak için bekleyen görevlerin yönetiminde kuyruk yapısını kullanır.

60

## Kernel veri yapıları

### Ağaçlar

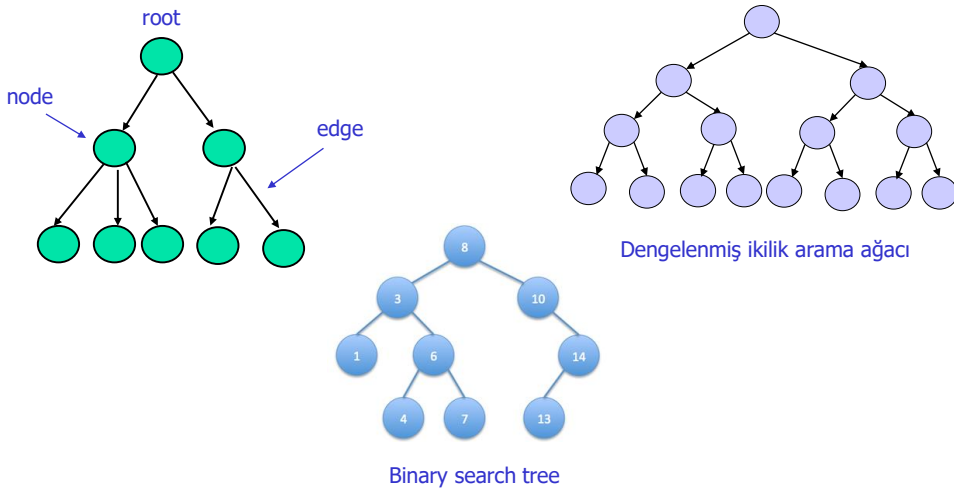
- Ağaçlar (**tree**) veriyi hiyerarşik şekilde göstermek için kullanılır.
- Ağaçlarda veriler **parent-child** ilişkisiyle tanımlanır.
- Genel olarak bir düğümde istenildiği kadar child olabilir.
- İkilik ağaçlarda (**binary tree**) ise bir düğüm iki child düğüme sahip olabilir.
- Binary arama ağaçlarda (**binary search tree**) bir elemana erişim performansı  $O(n)$ 'dir (**worst case**).
- Dengelenmiş binary arama ağaçlarda (**balanced binary search tree**) bir elemana erişim performansı  $O(\log n)$ 'dir (**worst case**).

61

## Kernel veri yapıları

### Ağaçlar

- Bir ağaç (**tree**) veriyi hiyerarşik şekilde göstermek için kullanılır.

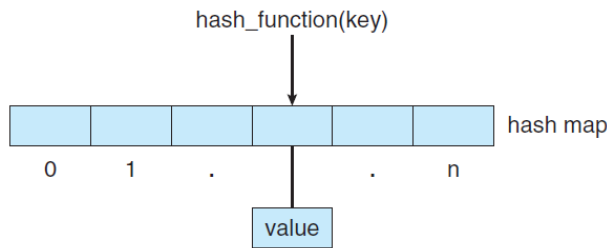


62

## Kernel veri yapıları

### Hash fonksiyonları

- Bir hash fonksiyonu (**hash function**) giriş olarak veri alır, bu veri üzerinde sayısal bir işlem yapar ve bir sayısal değer döndürür.
- Hash fonksiyonlarında veriye erişim performansı  $O(1)$ 'dir.
- Hash map, bir anahtar ile değer eşleştirmesi yapar.
- Eşleştirme genellikle tekildir.



63

## Kernel veri yapıları

### Bitmap

- Bitmap'ler  $n$  adet binary bit ile oluşturulan dizgidir (string).
- Hash fonksiyonlarında veriye erişim performansı  $O(1)$ 'dir.

001011101

- Çok sayıdaki kaynağın durumları ile ilgili bilgi (meşgul, kullanılabilir) bitlerle tutulabilir.
- Yukarıdaki bitler için 0, 1, 3 ve 7.kaynaklar kullanılabilir; 2, 4, 5, 6 ve 8.kaynaklar meşgul durumdadır.
- İşletim sistemi, disk bloklarının durumunu tutmak için bitmap kullanır.
- İşletim sistemleri kernel algoritmalarında veri yapılarını sıklıkla kullanır.

64

## Konular

- İşletim sistemi ne iş yapar?
- Bilgisayar sistemi organizasyonu
- Bilgisayar sistemi mimarisi
- İşletim sistemi yapısı
- İşletim sistemi işlemleri
- Process yönetimi
- Memory yönetimi
- Storage yönetimi
- Koruma ve güvenlik
- Kernel veri yapıları
- Hesaplama ortamları
- Açık kaynak işletim sistemleri

65

## Hesaplama ortamları

### Geleneksel hesaplama

- Birkaç yıl öncesine kadar ofisteki bir bilgisayar ağa bağlanmakta, yazıcı veya diğer kaynakları kullanmaktaydı.
- Günümüzde **Web teknolojileri** ve **WAN (Wide Area Network)** geleneksel **hesaplama ortamlarının sınırlarını genişletmiştir.**
- Firmalar **portal** oluşturmakta ve Web erişimiyle istemcilere kaynaklara erişim sağlamaktadır.
- **Mobil cihazlar kablosuz ağlar (wireless networks)** ile **Web portal'e bağlanırlar.**
- Konut kullacılarının bant genişliği günümüzde hala yeterli düzeyde değildir, ayrıca ağlarının güvenliği için **firewall** kullanırlar.
- Firewall, **IP filtreleme, port filtreleme, Web filtreleme, içerik filtreleme** yapabilir.

66



## Hesaplama ortamları

### *Mobil hesaplama*

- **Mobil hesaplama (mobile computing), akıllı telefonlar ve tablet bilgisayarlar ile yapılan işlemleri ifade eder.**
- Mobil cihazların özellikleri (ekran boyutu, hafıza kapasitesi ve performansı) son yıllarda önemli ölçüde gelişmiştir.
- Günümüzde **mobil cihazlar** sadece Web ve e-posta uygulamaları için değil, **tüm işlemler için kullanılır hale gelmiştir.**
- Mobil ortamlar için günümüzde **Apple iOS** ve **Google Android** işletim sistemleri yaygın olarak kullanılmaktadır.

67

## Hesaplama ortamları

### *Dağıtık sistemler*

- Bir dağıtık sistem, **fiziksel olarak ayrı, heterojen bilgisayar sistemidir.**
- Bir dağıtık sistem, **sahip olduğu çok sayıdaki kaynağa kullanıcıların erişimini sağlar.**
- **Bazı işletim sistemleri, sadece dosya erişimine yöneliktir ve FTP (File Transfer Protocol) ve NFS (Network File System) protokollerini kullanırlar.**
- **Bazı işletim sistemleri ise ağ fonksiyonlarını (kullanıcı yönetimi, kaynak atama, ...) kullanıcıların kullanmasına izi verir.**
- Bir ağ (**network**), iki veya daha fazla sistemin iletişimini sağlar.
- **Ağlar kullandıkları protokollere göre farklılık gösterirler.**
- Günümüzde **TCP/IP (Transmission Control Protocol/Internet Protocol)** en yaygın kullanılan protokol yığındır.

68

## Hesaplama ortamları

### Dağıtık sistemler

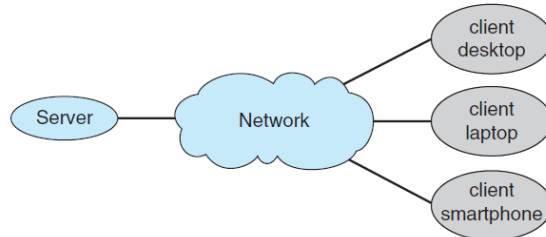
- Bilgisayar ağları kapsadıkları alana göre, **LAN (local-area network)**, **MAN (metropolitan-area network)** ve **WAN (wide-area network)** olarak üç gruba ayrılır.
- Bir bilgisayar ile laptop veya akıllı telefon arasında oluşturulan ağ, **PAN (personel-area network)** olarak adlandırılır.
- **WLAN (wireless LAN)** ise **IEEE 802.11** ve **Bluetooth** teknolojileriyle oluşturulan yaklaşık 300 metre kapsama alanına sahip ağıdır.
- Bir ağ işletim sistemi, ağdaki kaynakların yönetimini ve farklı bilgisayarlar üzerinde çalışan process'ler arasında iletişimi sağlar.

69

## Hesaplama ortamları

### İstemci-sunucu mimarisi

- Günümüzde birçok sunucu (**server**) sistemi, istemci (**client**) bilgisayarların isteklerini karşılamak üzere tasarlanır.
- **Compute-server** sistemlerinde, istemciler bir işlemin yapılması için arayüz üzerinden istek gönderirler.
- **File-server** sistemlerinde, istemciler dosya oluşturma, silme veya okuma işlemlerini yapabilirler.
- Sunucu, **yüksek konfigürasyona**, istemci ise **düşük konfigürasyona** sahiptir.

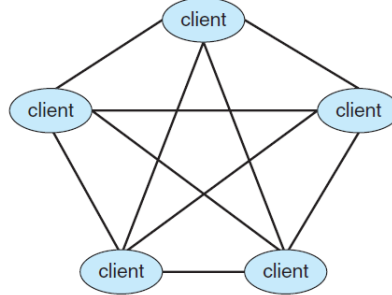


70

## Hesaplama ortamları

### Peer-to-peer mimarisi

- Eş düzey (**peer-to-peer, P2P**) mimarisinde istemci ve sunucu ayrımı yoktur. Tüm birimler aynı işlem kapasitesine ve yetkisine sahiptir.
- Kaynaklara erişim de dağıtık bir şekilde gerçekleştirilir.
- Napster, Gnutella gibi dosya paylaşım servisleri P2P mimarisine sahiptir.
- VoIP (voice over IP) teknolojisi kullanan **Skype**, P2P mimarisine sahiptir.



71

## Hesaplama ortamları

### Sanallaştırma

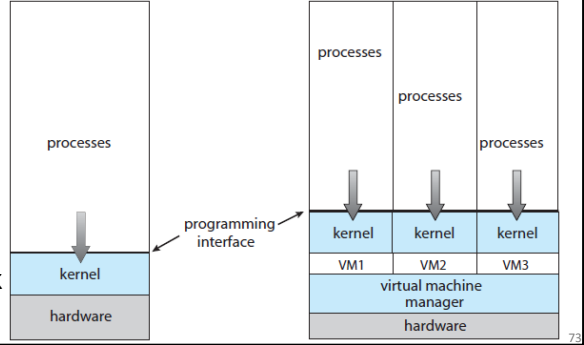
- Sanallaştırma (**virtualization**), işletim sistemlerinin uygulamaları başka işletim sistemlerinde çalıştırmasına izin verir.
- Sanallaştırma, **emülatör** olarak adlandırılan yazılımı içerir.
- Emülatörler, kaynak CPU ile hedef CPU'nun farklı olduğu durumlarda kullanılır.
- Apple, IBM CPU için derlenen bir programı Intel CPU'da çalıştırmak isterse, **Rosetta** isimli emülatörü (dinamik binary çevirici) kullanır.
- Yorumlayıcılar (**interpreter**), emülatör yazılımlarıdır ve yüksek seviyeli dilde yazılan programları makine koduna (**native code**) çevirmeden çalıştırır. **Makine koduna çevirme derleyicilerle (compiler) yapılır.**
- Basic, derleme de yapabilir, yorumlama da yapabilir. Java, her zaman yorumlayıcıdır. **JVM (Java Virtual Machine) bir emülatör yazılımıdır.**

72

## Hesaplama ortamları

### Sanallaştırma

- **VMware**, bir işletim sistemi üzerinde farklı işletim sistemlerinin misafir (**guest copy**) olarak çalışmasına ve kendi uygulamalarını çalıştırmasına izin verir.
- Şekilde Windows host işletim sistemi, VMware uygulaması ise sanal makine yöneticisidir (**virtual machine manager - VMM**).
- **VMM**, farklı işletim sistemlerini çalıştırır, kaynak kullanımlarını yönetir ve kullanıcıların birbirini etkilemesini önler.
- **VMware ESXi** ve **Citrix XenServer**, host olarak çalışır.



## Hesaplama ortamları

### Bulut bilişim

- **Bulut bilişim (cloud computing)**, hesaplama, depolama ve uygulamaları bir ağ aracılığıyla servis olarak dağıtır.
- Bulut hesaplama, sanallaştırmanın mantıksal uzantısı olarak kabul edilebilir.
- **Amazon Elastic Compute Cloud (EC2)**, binlerce sunucuya, milyonlarca sanal makineye ve petabyte depolama alanına sahiptir.
- EC2 bu kaynakları İnternet üzerinden kullanıcılara sunar ve **kullanıcılar kullandıkları kaynak oranınca aylık ücretlendirilirler**.

## Hesaplama ortamları

### Bulut bilişim

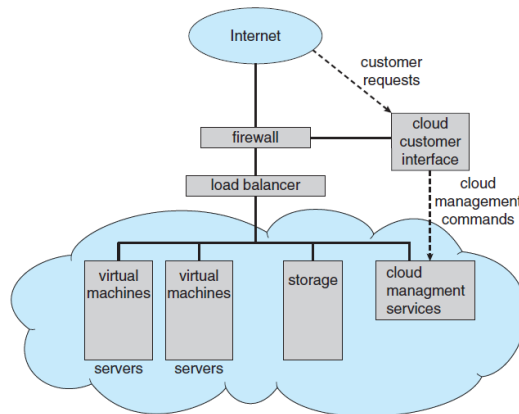
- Farklı bulut hesaplama türleri vardır:
  - **Public cloud** – İnternet üzerinden herkesin kullanımına açıktır.
  - **Private cloud** – bir firmanın sahibi olduğu buluttur.
  - **Hybrid cloud** – public ve private kullanılan bulut bileşenlerine sahiptir.
  - **Software as a service (SaaS)** – bir veya daha fazla uygulama (Word, Excel, ...) İnternet aracılığıyla kullanıma açıktır.
  - **Platform as a service (PaaS)** – Bir yazılım yığını (veritabanı sunucusu) uygulamalar için İnternet aracılığıyla kullanıma açıktır.
  - **Infrastructure as a service (IaaS)** – Sunucular veya depolama birimleri (üretilen verinin yedeklenmesi) İnternet aracılığıyla kullanıma açıktır.
- Bir bulut ortamı yukarıdaki türlerden birden fazlasını sağlayabilir.

75

## Hesaplama ortamları

### Bulut bilişim

- **Bulut hesaplama**, VMM yapısının ötesinde, kullanıcı process'lerinin çalıştığı **sanal makineleri yönetir**.
- VMM'ler bulut yönetim araçları (**Vware vCloud Director, Eucalyptus**) ile yönetilirler.



76

## Hesaplama ortamları

### Gerçek zamanlı gömülü sistemler

- Gömülü (**embedded**) sistemler, günümüzde **araç motorları, üretim robotları, mikro dalga fırınlar, uçaklar, teknolojik silahlar, ...**, gibi çok farklı yerlerde kullanılmaktadır.
- **Gömülü sistemler özel amaçlar için geliştirilirler ve sahip oldukları işletim sistemleri sınırlı özelliklere sahiptir.**
- Genellikle **arayüz gerektirmezler, donanımların izlenmesi ve yönetimini gerçekleştirirler. Farklı türleri vardır:**
  - İşletim sistemine sahiptirler ve özel amaçlı uygulamaları çalıştırır.
  - Özel amaçlı gömülü işletim sistemine sahiptirler ve istenen işlevleri yerine getirir.
  - Uygulamaya özel bütünleşik devreye (**application specific integrated circuits - ASICs**) sahiptirler ve işletim sistemine sahip değildirler. Bu tür sistemler **hardwired systems** olarak da adlandırılırlar.

77

## Hesaplama ortamları

### Gerçek zamanlı gömülü sistemler

- **Gömülü sistemler, gerçek zamanlı işletim sistemlerini (**real-time operating systems**) çalıştırır.**
- **Gerçek zamanlı işletim sistemlerinde zaman gereksinimi çok hassastır.**
- **İstenen zaman aralıklarında veya belirlenen anda işlemlerin gerçekleştirilmesi zorunludur.**
- **Otomobil enjeksiyon sistemleri, medikal uygulamalar, silah sistemleri** başlıca uygulama alanlarıdır.

78

## Konular

- İşletim sistemi ne iş yapar?
- Bilgisayar sistemi organizasyonu
- Bilgisayar sistemi mimarisi
- İşletim sistemi yapısı
- İşletim sistemi işlemleri
- Process yönetimi
- Memory yönetimi
- Storage yönetimi
- Koruma ve güvenlik
- Kernel veri yapıları
- Hesaplama ortamları
- Açık kaynak işletim sistemleri

79

## Açık kaynak işletim sistemleri

- Açık kaynak ([open-source](#)) işletim sistemlerinde, derlenmiş binary kod yerine **kaynak kodu da kullanılabilir durumdadır.**
- **Linux** açık kaynak işletim sistemlerine, **Windows** ise kapalı kaynak ([closed-source](#)) işletim sistemlerine örnek olarak verilebilir.
- **Apple Mac OS X** ve **iOS** hibrit işletim sistemleridir. Açık kaynak kernel'a ([Darwin](#)) sahiptir aynı zamanda kapalı kaynak bileşenlere de sahiptir.
- Kapalı kaynak işletim sistemlerinde **tersine mühendislik** ([reverse engineering](#)) kullanılarak binary kod oluşturulabilir.
- **Açık kaynak işletim sistemlerinde programcılar geliştirmeye katkı sağlayabilmektedirler.**
- **Açık kaynak**, kapalı kaynağa göre **daha güvenlidir.** Çünkü daha çok kişi tarafından kod görülmektedir.
- **Linux**, **BSD Unix** ve **Solaris** açık kaynak işletim sistemleridir.

80



## Ödev

- İşletim sistemlerinin kernel fonksiyonları hakkında araştırma ödevi hazırlayınız.