



# Bölüm 1: Giriş

## İşletim Sistemleri



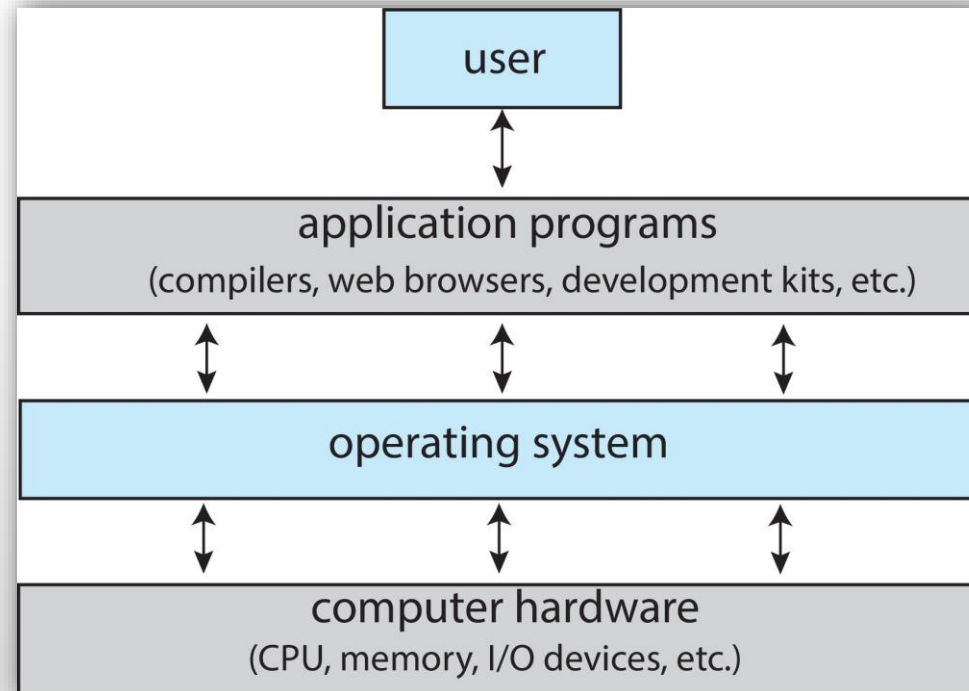
# İşletim Sistemi

- Modern bilgisayarlar çok karmaşıktır.
- Uygulama geliştiricisinin her detayı bilmesi imkansızdır.
- Kaynakları daha iyi ve daha basit bir şekilde yönetebilmek için bir katman
- Çeşitli işletim sistemleri;
  - *Windows, Linux, MacOS, Android, IOS, Contiki, VxWorks*
- Kullanıcılar,
  - **Kabuk (Shell)** veya
  - **GKA (GUI)** sayesinde bilgisayar ile etkileşime girer.



# Modern Bilgisayarın Bileşenleri

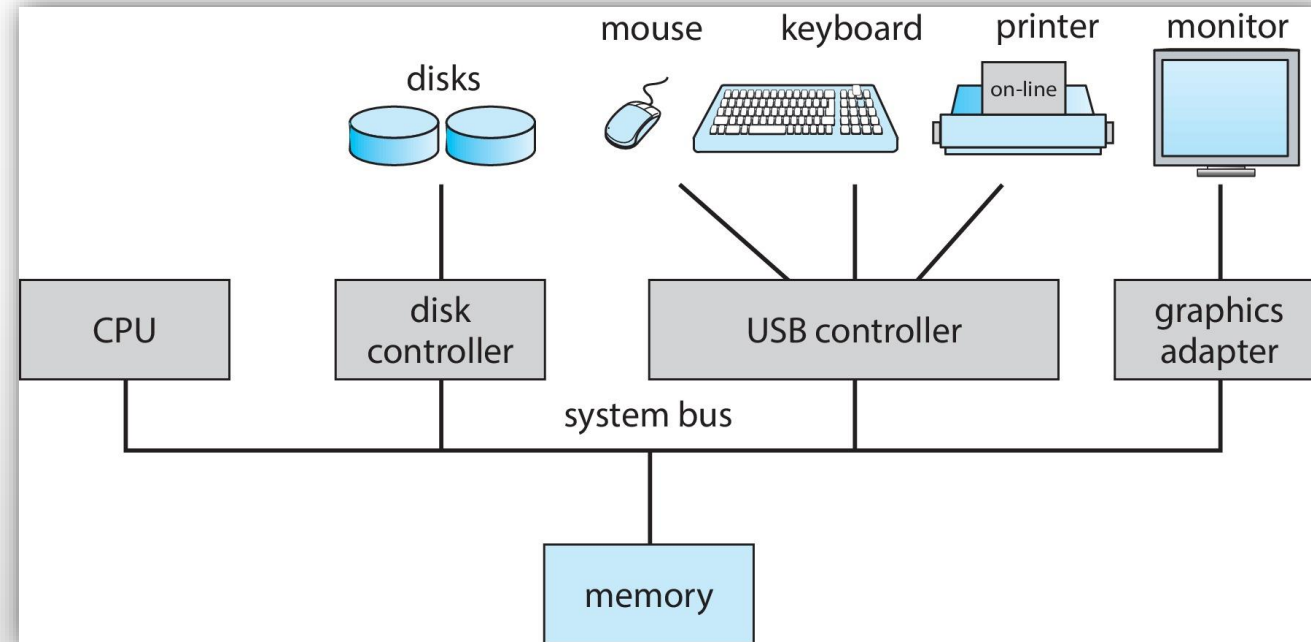
■ .





# Modern Bilgisayarın Donanım Bileşenleri

- İşlemci
- Ana bellek
- Disk
- Yazıcı
- Klavye
- Fare
- Ekran
- Ağ arayüzleri
- G/Ç cihazları





# Kabuk (Shell) ve GKA (Grafik Kullanıcı Arayüzü)

- **Kabuk:** İşletim sistemiyle etkileşimini sağlayan komut satırı arayüzü.
- **GKA:** Simge, pencere, fare işaretçisi gibi grafik öğeleri sayesinde işletim sistemi ile etkileşimi sağlayan görsel arayüzü.
- Kabuk ve GKA,
  - işletim sisteminin çekirdeği ile alakalı değil,
  - temel sistem işlevlerini veya hizmetlerini sağlamazlar.
  - kullanıcı arayüzünün bir parçası.
  - kullanıcının işletim sistemiyle etkileşime girmesi için bir araç sağlar.



# Kabuk (Shell) ve GKA (Grafik Kullanıcı Arayüzü)

- İşletim sisteminde aynı anda birden çok kabuk ve GKA kullanılabilir.
- Kişiselleştirme sağlar.
- Özel ihtiyaçların daha verimli bir şekilde yerine getirilmesini sağlar.
- Kabuk, belirli görevleri gerçekleştirmek için verimli ve etkili bir yol sağlar.
- GKA, daha kullanıcı dostu ve sezgisel bir yol sağlar.
- Geniş bir kullanıcı kitlesi tarafından kullanılabilir olmasını sağlar.

# Kabuk



```
Command Prompt

Media State . . . . . : Media disconnected
Connection-specific DNS Suffix . :

Wireless LAN adapter Wi-Fi:

Connection-specific DNS Suffix . : local
Link-local IPv6 Address . . . . . : fe80::beb9:b82:dc20:c3a2%12
IPv4 Address. . . . . : 192.168.1.89
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . : 192.168.1.1

Ethernet adapter Bluetooth Network Connection:

Media State . . . . . : Media disconnected
Connection-specific DNS Suffix . :

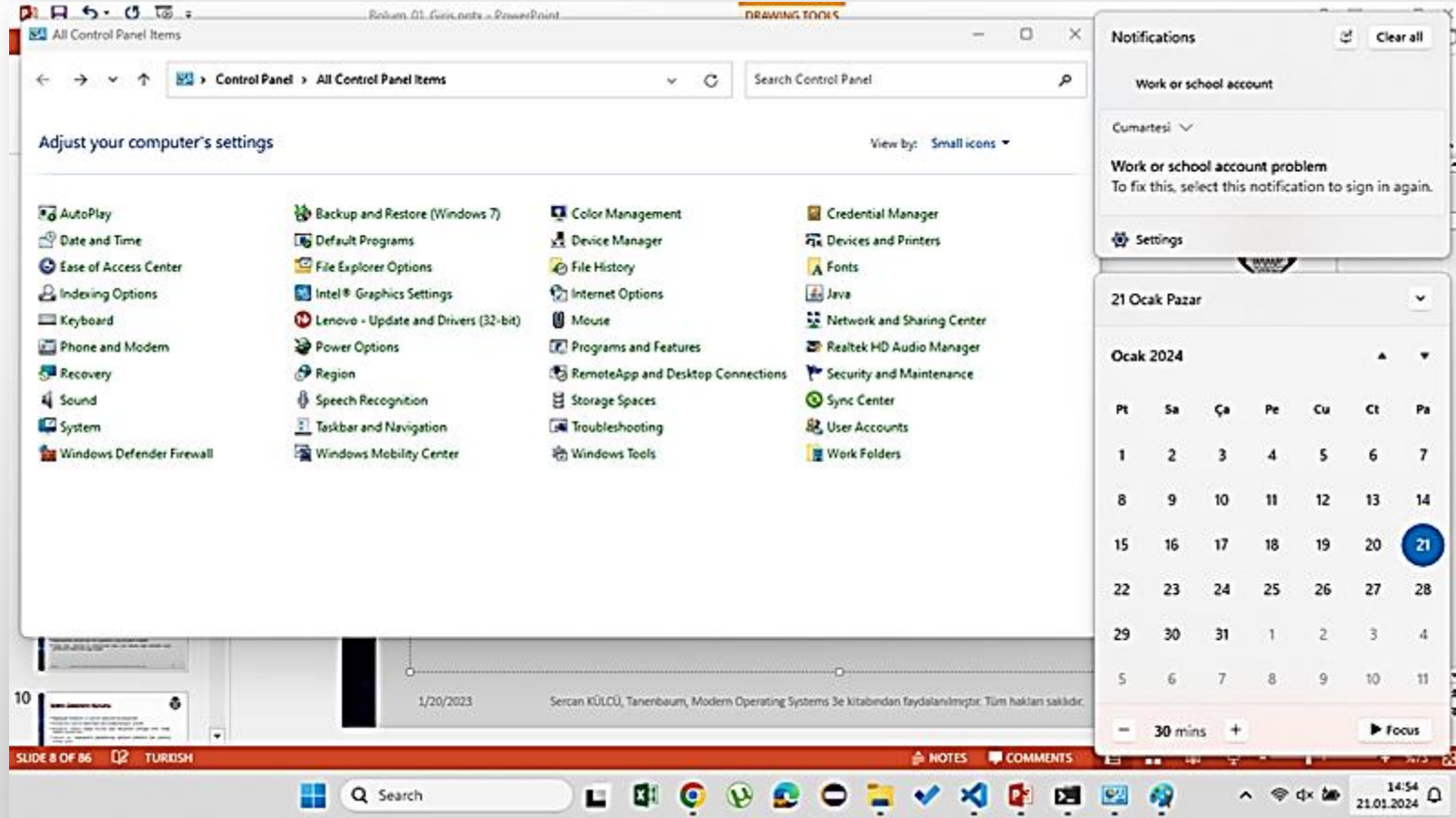
C:\Users\sercan>netstat -h

Displays protocol statistics and current TCP/IP network connections.

NETSTAT [-a] [-b] [-e] [-f] [-i] [-n] [-o] [-p proto] [-r] [-s] [-t] [-x] [-y] [interval]

-a          Displays all connections and listening ports.
-b          Displays the executable involved in creating each connection or
           listening port. In some cases well-known executables host
           multiple independent components, and in these cases the
           sequence of components involved in creating the connection
           or listening port is displayed. In this case the executable
           name is in [] at the bottom, on top is the component it called,
```





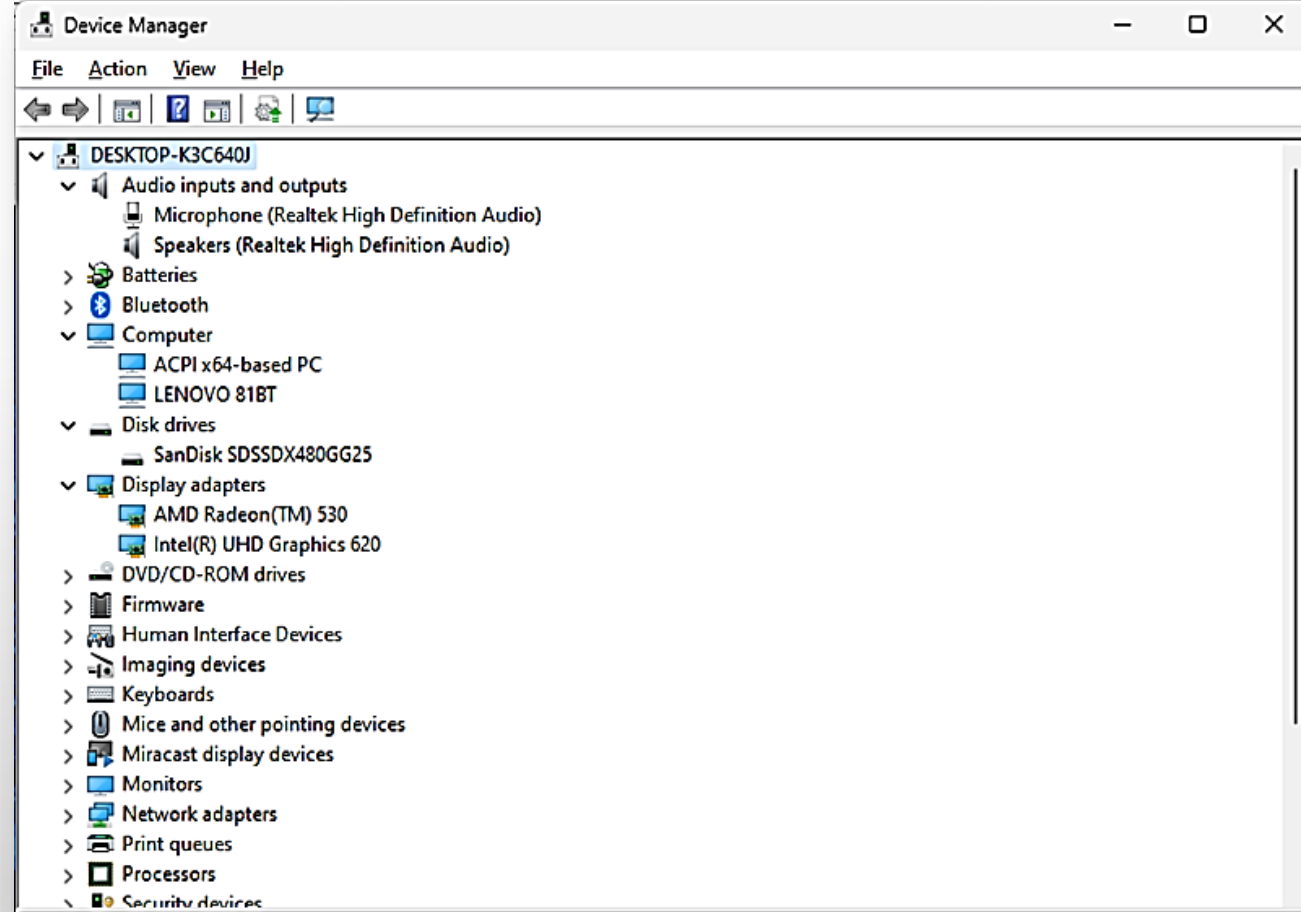




# Aygıt Yöneticisi

- Bilgisayara bağlı donanım aygıtlarını yönetir.
- Cihazların yapılandırılmasına, güncellenmesine olanak tanır.
- Donanım aygıtlarının işletim sistemi ile uyumlu çalışmasında kilit rol oynar.
- Bilgisayarda çalışan ayrı bir uygulama veya program değildir.
- Cihazın hakkında (türü, üreticisi, sürümü) bilgi sağlar.
- Cihaz ile ilgili sorun ve problemler hakkında bilgi sağlar.

# Aygıt Yöneticisi



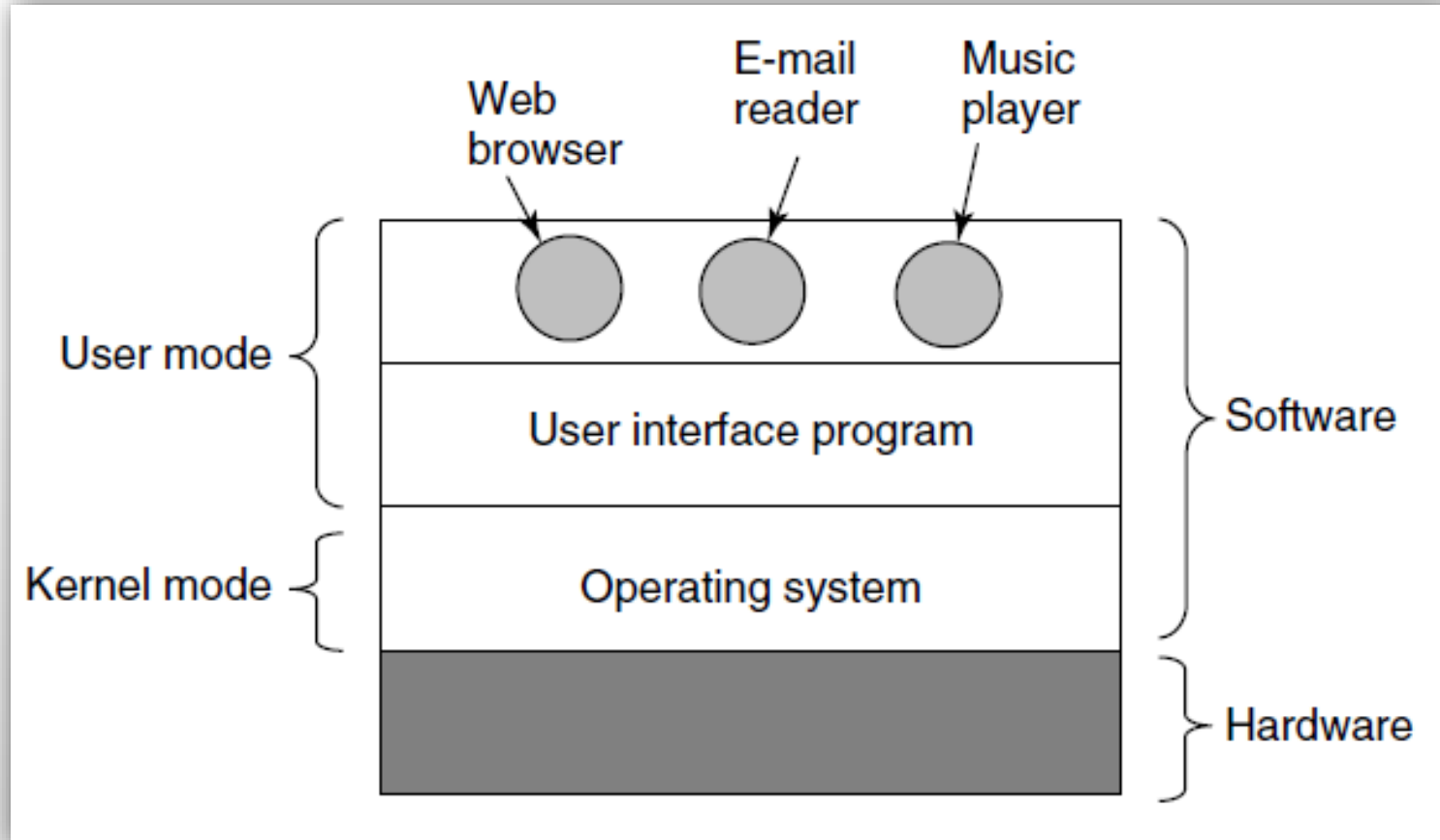


# İşletim Sisteminin Konumu

- Donanım ve yazılım arasında bir arayüzdür.
- Donanımın yazılım tarafından nasıl kullanılacağını yönetir.
- **Donanım,**
  - fiziksel olarak mevcut bileşenleri (CPU, RAM, disk) temsil eder.
- **Yazılım,**
  - bilgisayarın yapabileceği işlemleri yürütmek için yazılmış kodları içerir.
- **İşletim sistemi,**
  - yazılımın donanımı kullanmasını kontrol eder,
  - donanımın kullanımını optimize eder ve
  - sistemin güvenliğini sağlar.



# İşletim Sistemi Konumu





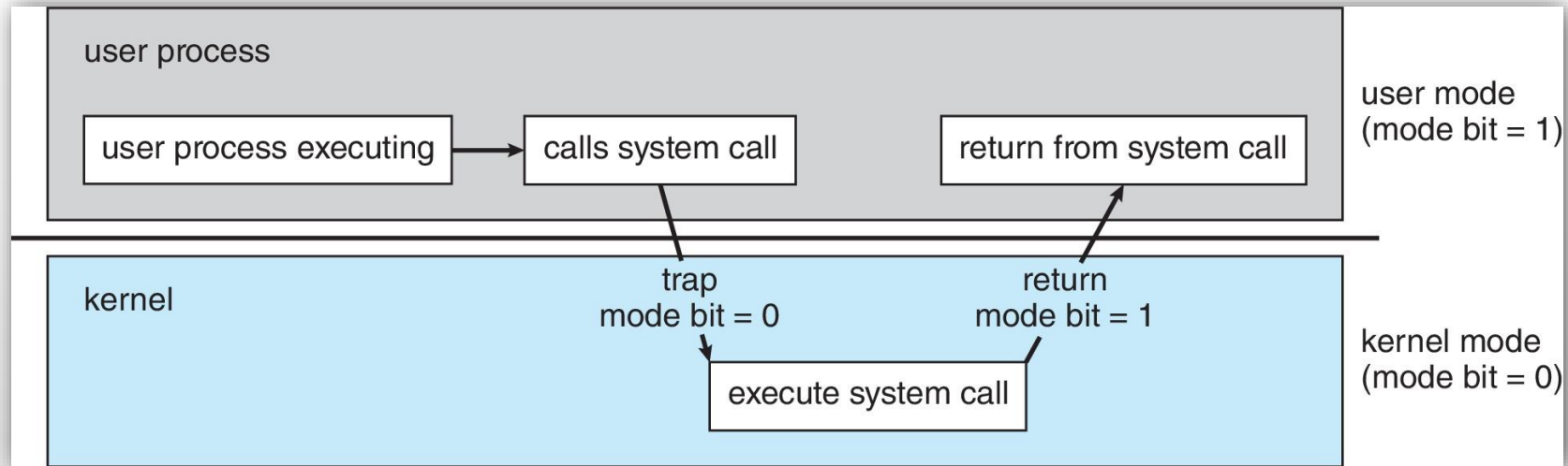
# Çekirdek Modu ve Kullanıcı Modu

- Çoğu bilgisayarın iki çalışma modu vardır:
  - Çekirdek modu, tüm donanıma tam erişime sahip olunur ve herhangi bir talimat yürütülebilir.
  - Kullanıcı modu, sınırlı erişim ve kapasiteye sahip olunur.
- Kabuk ve GKA, kullanıcı modu yazılımının en düşük seviyesidir.



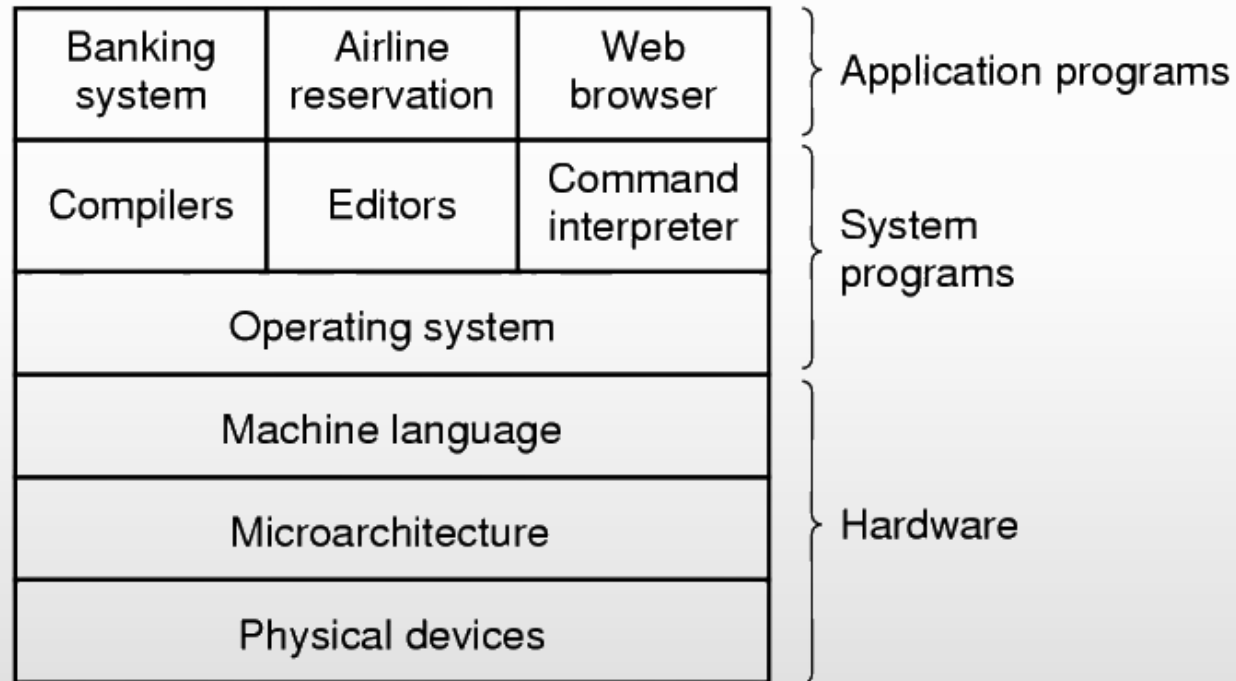
# Çekirdek Modu ve Kullanıcı Modu

■ .





# İşletim Sistemi Konumu







# Fiziksel Aygıtlar, Mikro Mimari, Makine Dili

- **Fiziksel aygıtlar**, merkezi işlem birimi (CPU), bellek, depolama ve giriş/çıkış aygıtları gibi donanım bileşenleridir.
- **Mikro mimari**, veri yolu, bellek hiyerarşisi ve kontrol birimi dahil olmak üzere bilgisayar işlemcisinin özel tasarımıdır.
- **Makine dili**, doğrudan donanım tarafından yürütülebilen ikili koddan oluşan en düşük seviyeli programlama dilidir.
- İşletim sistemi, aygıt sürücülerini ile fiziksel aygıtlarla iletişim kurar, yönetir.
- Aygıt sürücülerini, üst katmandan gelen isteklerini fiziksel aygıtlar tarafından anlaşılacak ve yürütülebilecek düşük düzeyli komutlara çevirir.



# Fiziksel Aygıtlar, Mikro Mimari, Makine Dili

- İşletim sistemi, uygulamalar ile mikro mimari arasında arayüz sunar.
- Uygulamaların kaynaklara verimli ve etkin bir şekilde erişmesine ve kullanılmasına izin verir.
- Üst düzey uygulamaları doğrudan donanım tarafından yürütülebilen makine diline çevirir.
- Donanım yapılandırması veya mikro mimarisi ne olursa olsun uygulamaların bilgisayarda yürütülmesine izin verir.
- İşletim sistemi tasarlanırken, bilgisayarın mikro mimarisi ve fiziksel cihazları dikkate alınır.

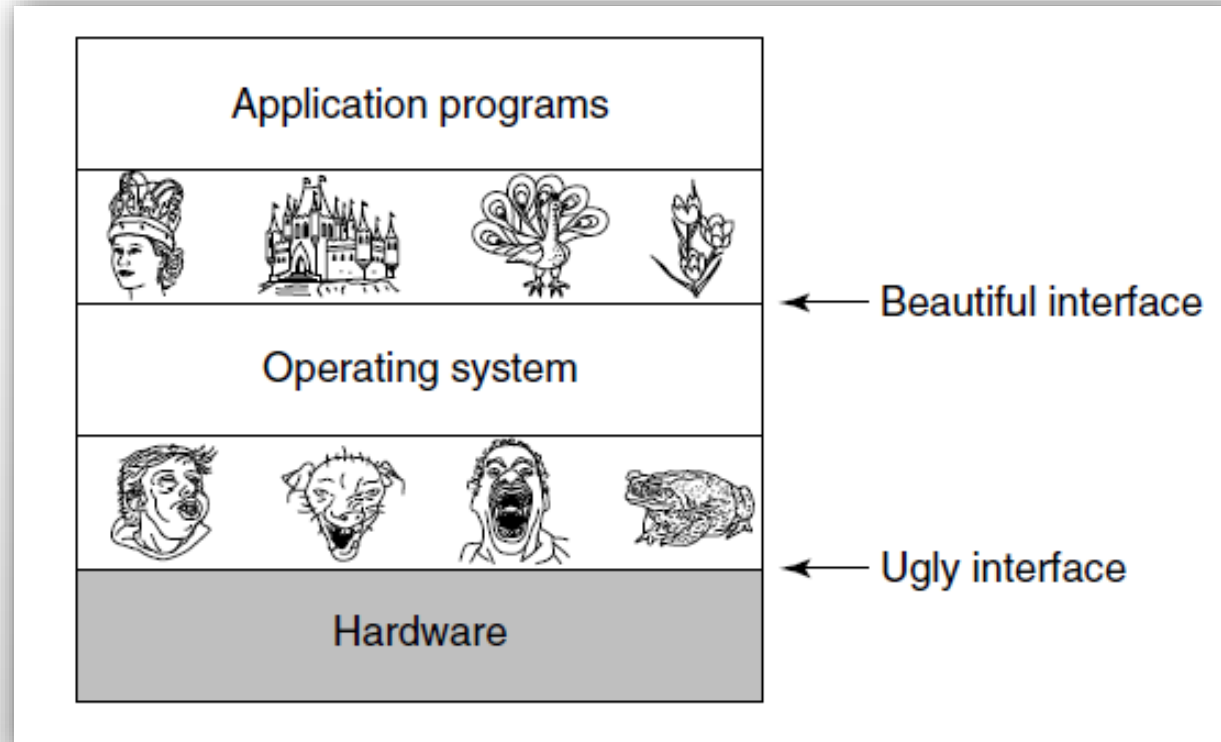


# Genişletilmiş Makine Olarak İşletim Sistemi

- Donanımın üstüne inşa edilmiş bir yazılımdır.
- Bilgisayar donanımını kullanmayı kolaylaştırır.
- Donanımın özelliklerini ve yeteneklerini kullanılabilir hale getirir.
- Donanımın özelliklerini gizler ve direkt kullanmasını engeller.
- İşletim sistemi ara yüzünü kullanmak daha kolaydır.
- İşletim sistemleri çirkin donanımları güzel soyutlamalara dönüştürür.



# Genişletilmiş Makine Olarak İşletim Sistemi





# Genişletilmiş Makine Olarak İşletim Sistemi

- Soyutlama:
  - İşlemci – Süreç
  - Depolama – Dosya
  - Bellek – Adres uzayı
- 4 tip rol var:
  - Donanım tasarımcısı
  - Çekirdek tasarımcısı
  - Uygulama geliştirici
  - Son kullanıcı



# Kaynak Yöneticisi Olarak İşletim Sistemi

- İşletim sistemi, donanım kaynaklarını etkili bir şekilde yönetir.
- Kaynak kullanımını optimize eder.
- Kaynakların uygulamalar arasında adil bir şekilde dağıtımını sağlar.
- Üstten aşağıya bakış açısı:
  - Uygulama programları için *soyutlamalar* sağlar
- Aşağıdan yukarıya bakış açısı:
  - Karmaşık sistemin parçalarını yönetir
- Alternatif bakış açısı:
  - Kaynakların düzenli ve kontrollü dağıtımını sağlar.



# Kaynak Yöneticisi Olarak İşletim Sistemi

- Birden çok programın aynı anda çalışmasına izin verir.!
- Bellek, G/Ç cihazları ve diğer kaynakları yönetir ve korur.
- Kaynakları iki farklı şekilde paylaşır.
  - *Zaman* (time)
  - *Alan* (space)
- Birçok program aynı anda belge yazdırmak isterse ne olur?
- Her sürecin kaynak kullanımı/ihtiyacı nasıl hesaplanır?
- Kaynaklar çoklanırsa, adalet ve verimlilik nasıl sağlanır?



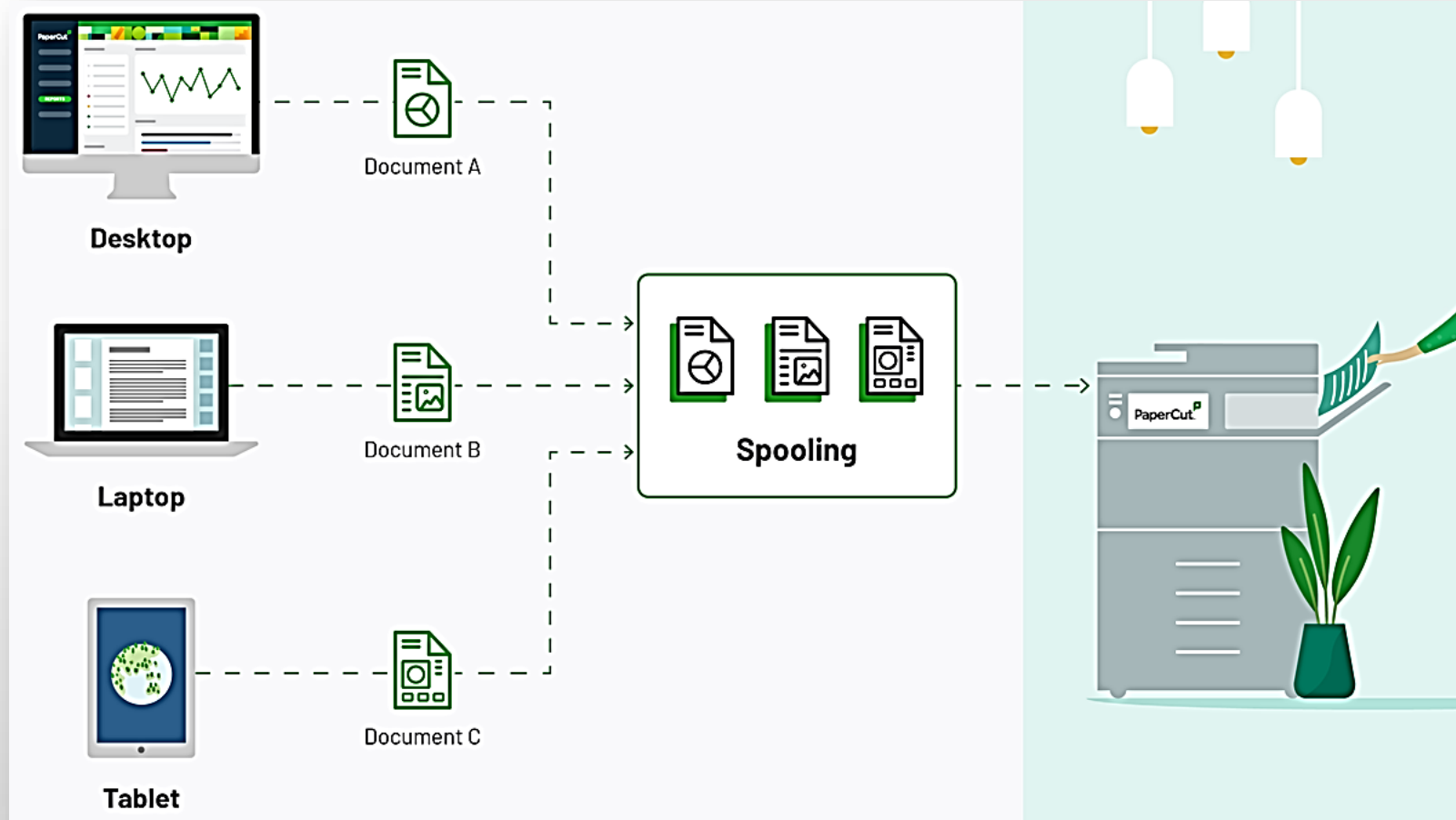


# Birçok Program Aynı Anda Yazdırmak İsterse

- Programlar, yazdırma isteklerini işletim sistemine gönderir.
- İşletim sistemi, yazdırma isteklerini koordine eder ve önceliklendirir.
- İşletim sistemi, kaynakları tahsis etmekten ve isteklerin yürütülme sırasını belirlemekten sorumludur.
- İşletim sistemi, eşzamanlı yazdırma isteklerini yönetmek için *İlk Gelen, İlk Hizmet Alır (FCFS)*, *Önce En Kısa İş (SJF)*, *Round Robin* ve *Priority Scheduling* gibi çeşitli zamanlama algoritmaları kullanır.
- Eşzamanlı yazdırma işlemini optimize etmek için, işletim sistemi *ara belleğe alma* ve *biriktirme (spooling)* tekniklerini kullanır.



# Birçok Program Aynı Anda Yazdırmak İsterse





# Süreçlerin Kaynak Kullanımının Hesaplanması

- Bir sürecin kaynak kullanımı, görevi tamamlaması için ihtiyaç duyulan CPU zamanı, bellek miktarı ve G/Ç işlem sayısı üzerinden tanımlanır.
- Windows'ta performans sayaçları, Linux'ta /proc dosya sistemi gibi performans izleme araçları ile sürecin **işlemci kullanımı** ölçülebilir.
- Bellek tahsis etme ve serbest bırakma modelleri izlenerek ve sürecin *çalışma kümesinin* boyutu izlenerek **bellek kullanımı** ölçülebilir.
- Sistem çağrıları ve G/Ç istekleri izlenerek gerçekleştirilen **G/Ç işlemlerinin sayısı** ölçülebilir.



# Kaynaklar Çoklandığında Adalet ve Verimlilik

- Kaynaklardaki artış,
  - performans ve hızlı yanıt verebilirlik gibi **faydalar** sağlayabilir,
  - kaynakların adil ve verimli kullanımını sağlamak gibi **zorluklar** getirir.
- Adalet ve verimliliği sağlamak için kaynak *çizelgeleme (scheduling)* algoritmaları, *önceliklendirme*, kaynak *kotaları* gibi yaklaşımlar kullanılır.
- Kullanılacak yaklaşım,
  - sistemin ve kullanıcının özel gereksinimlerine ve hedeflerine,
  - çalışan süreçlerin doğasına ve davranışına bağlıdır.



# İşletim Sistemlerinin Tarihi

- İlk jenerasyon (1945-55)
  - Vakum tüplü sistemler
- İkinci jenerasyon (1955-65)
  - Transistörlü ve toplu işlem (*batch*) sistemler
- Üçüncü jenerasyon (1965-1980)
  - Entegre yongalar (IC) ve çoklu programlama
- Dördüncü jenerasyon (1980-günümüz)
  - Kişisel bilgisayarlar
- Beşinci jenerasyon (1990-günümüz)
  - Mobil bilgisayarlar



# Vakum Tüplü Sistemler

- Büyük ve yavaş
- Mühendisler tasarlar, inşa eder, çalıştırır ve bakımını yapar
- Makine diliyle veya kablolar kullanılarak programlanır
- Takılabilir kartlar ile çalışır
- Ağırlıklı olarak sayısal hesaplamalar yapar



# Transistörlü ve Toplu İşlem (batch) Sistemler

- Transistörler, vakum tüplerin yerini aldılar
  - daha küçük, daha güvenilir ve daha enerji verimli
- İşlemler işlem kuyruğuna eklenir ve işletim sistemi sırayla yürütür.
  - İşlemlerin paralel olarak yürütülmesine izin vermez.
  - İşlemlerin manuel olarak yürütülmesini gerektirir.
  - Veri işleme, hesaplama ve raporlama gibi işlemler için kullanılır
  - Gerçek zamanlı işlemler için uygun değildir.



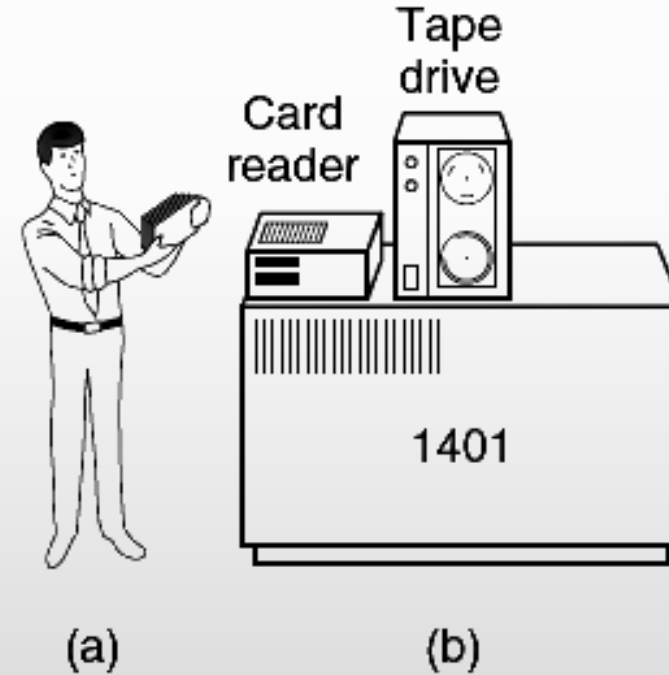


# Transistörlü ve Toplu İşlem Sistemler

- Eski bir toplu sistem.

(a) Programcılar 1401'e kartlar getirir.

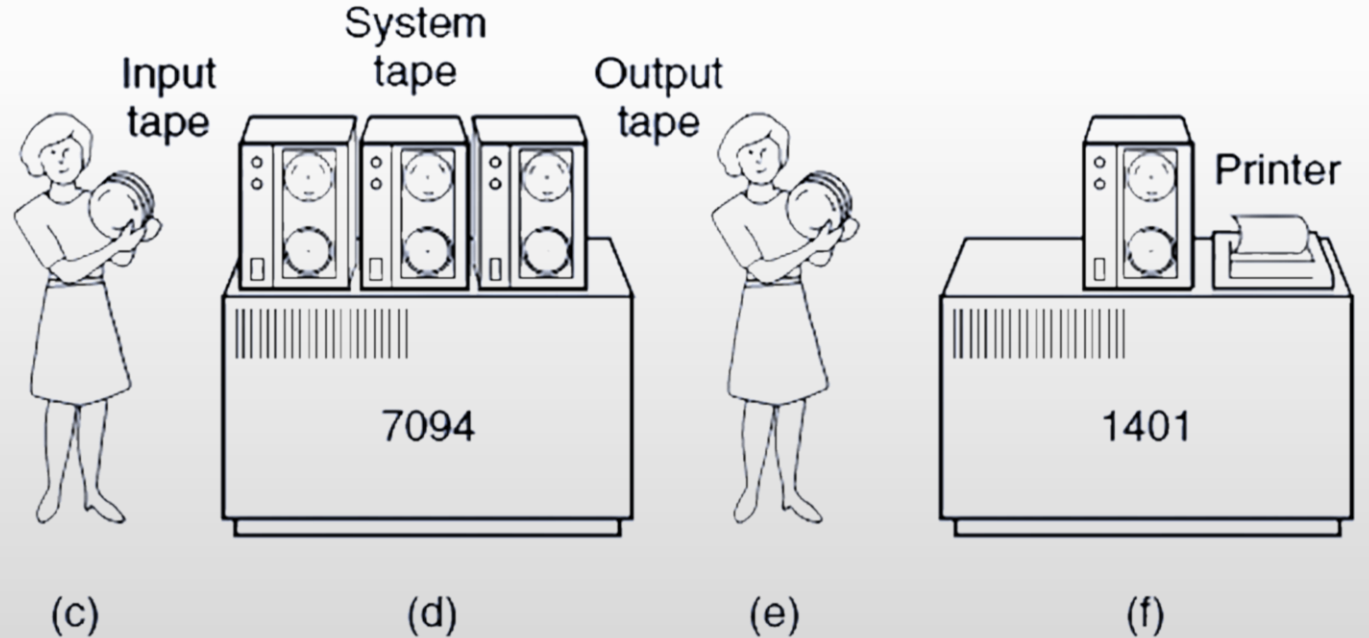
(b) 1401, iş kartlarını kasede (*teyp*) okur.





# Transistörler ve Toplu Sistemler

- (c) Girdi bandının 7094'e taşınması.
- (d) 7094 hesaplamaları yapar.
- (e) Çıktı bandının 1401'e taşınması.
- (f) 1401 çıktıyı yazdırır.





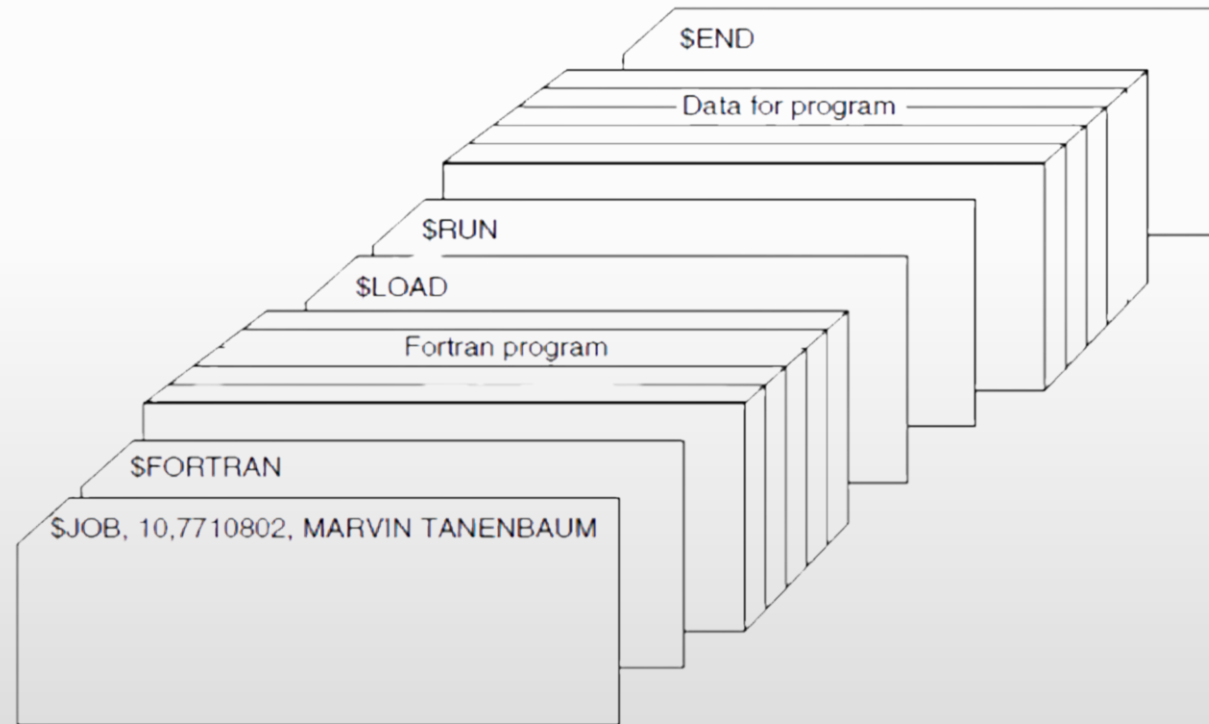
# IBM 1401 ve 7094, System/360

- IBM 1401, 1959'da tanıtılan ticari ve bilimsel uygulamalarda yaygın olarak kullanılan ilk bilgisayarlardan.
- IBM 7094, 1962'de tanıtılan 1401'in daha güçlü bir versiyonu. Bilimsel ve teknik alanlarda kullanıldı. Zaman paylaşımını destekleyen ve çok kullanıcının aynı anda sisteme erişebildiği ilk bilgisayarlardan.
- Hem 1401 hem de 7094, ilk büyük ölçekli işletim sistemlerinden biri olan IBM System/360 İşletim Sistemi (OS/360) altında çalışıyordu.
- Birden çok uygulamayı aynı anda çalıştırmayı ve yazıcılar, teypler ve disk sürücüler gibi kaynakları birden çok kullanıcı arasında paylaşmayı mümkün kıldı.



# Tipik bir FMS İşinin Yapısı

- FMS (Flexible manufacturing system)





# FMS Esnek Üretim Sistemi

- Fabrikada üretim sürecini otomatikleştiren bilgisayar kontrollü bir sistemdir.
- Malzeme taşıma, işleme ve montaj gibi bir dizi üretim sürecini tek bir otomatik sistemde birleştirir.
- İşletim sistemleri, FMS'nin çalışmasında çok önemli bir rol oynamaktadır.
- FMS ortamında işletim sistemi, güvenilir, verimli, ve ölçeklenebilir olmalıdır.
- FMS ortamlarında özel işletim sistemlerinin kullanılması,
  - üretim verimliliği,
  - gerçek zamanlı kontrol,
  - geliştirilmiş veri yönetimi sağlar.



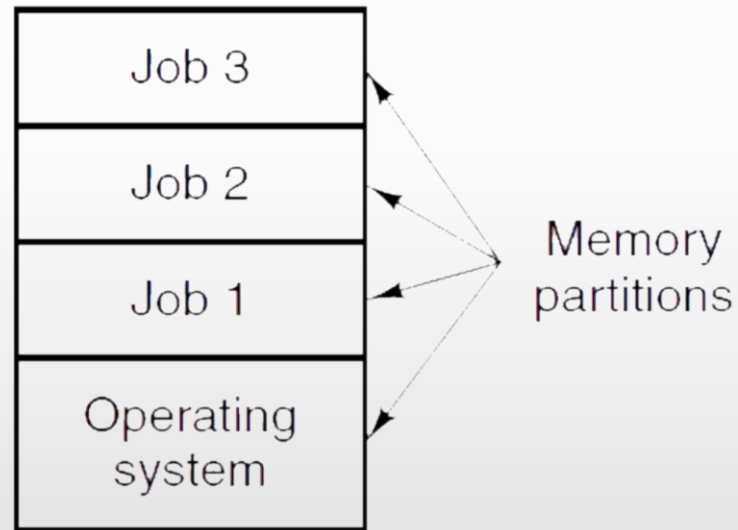
# Bütünleşik Devreler ve Çoklu Programlama

- Bütünleşik devreler transistörlerin yerini aldı.
  - Daha küçük, daha güvenilir ve daha enerji verimli
- Çoklu programlama, birden fazla işlemi aynı anda yürütmek için kullanılır.
  - Dinamik olarak işlemlerin ağırlıklarının ayarlanmasını sağlar.
  - İşlemler arasında eşitliği sağlar
  - İşlemlerin paralel olarak yürütülmesini sağlar.
  - Gerçek zamanlı işlemler için uygun.



# Bütünleşik Devreler ve Çoklu Programlama

- Bellekte üç işi olan bir çoklu programlama sistemi.





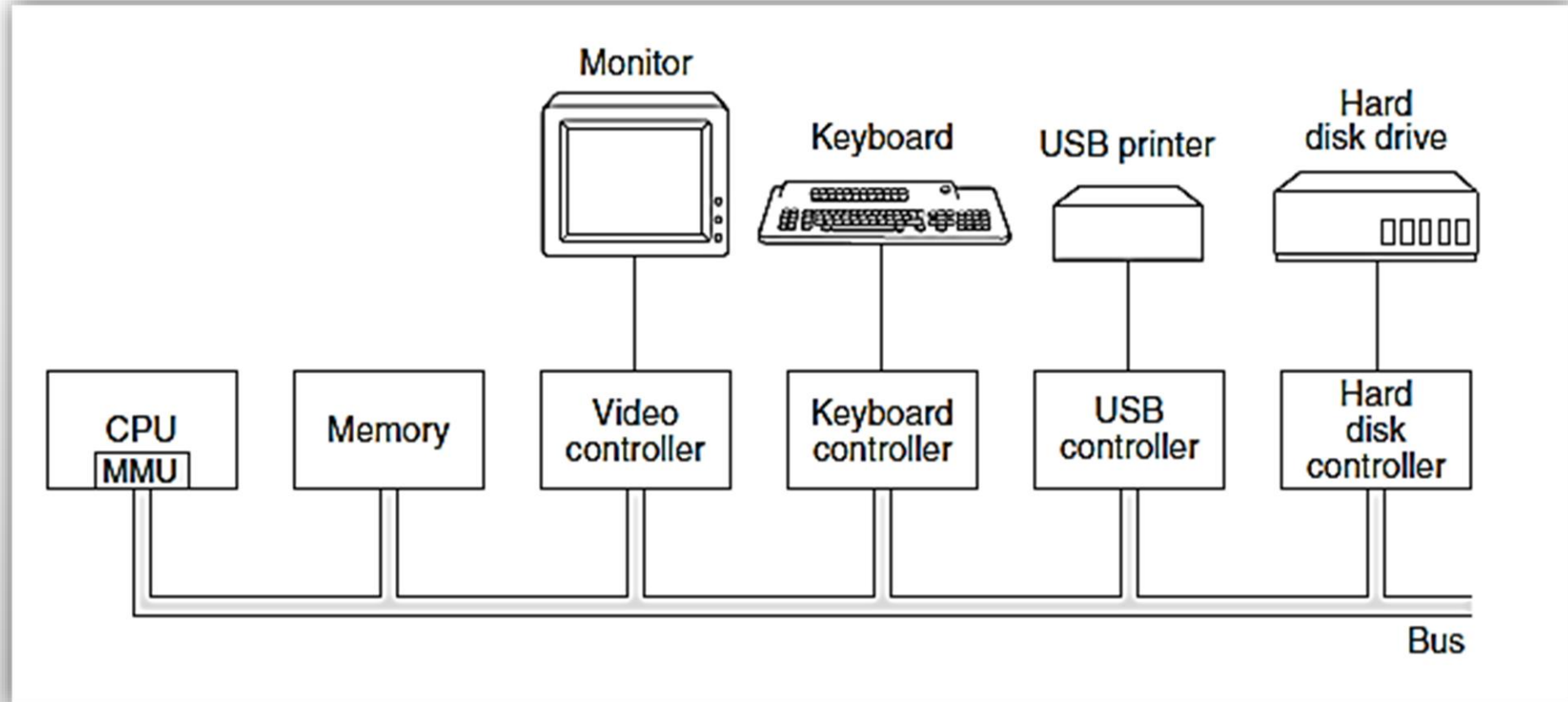


# Kişisel Bilgisayarlar

- Bilgisayarların fiyatları büyük ölçüde düşmüştür
- CP/M: İlk disk tabanlı işletim sistemi
- 1980: IBM PC, Basic dili yorumlayıcı, DOS, MS-DOS
- GUI, Lisa, Apple: kullanıcı dostu
- Grafik arayüzlü MS-DOS, Win95/98/ME, winNT/XP

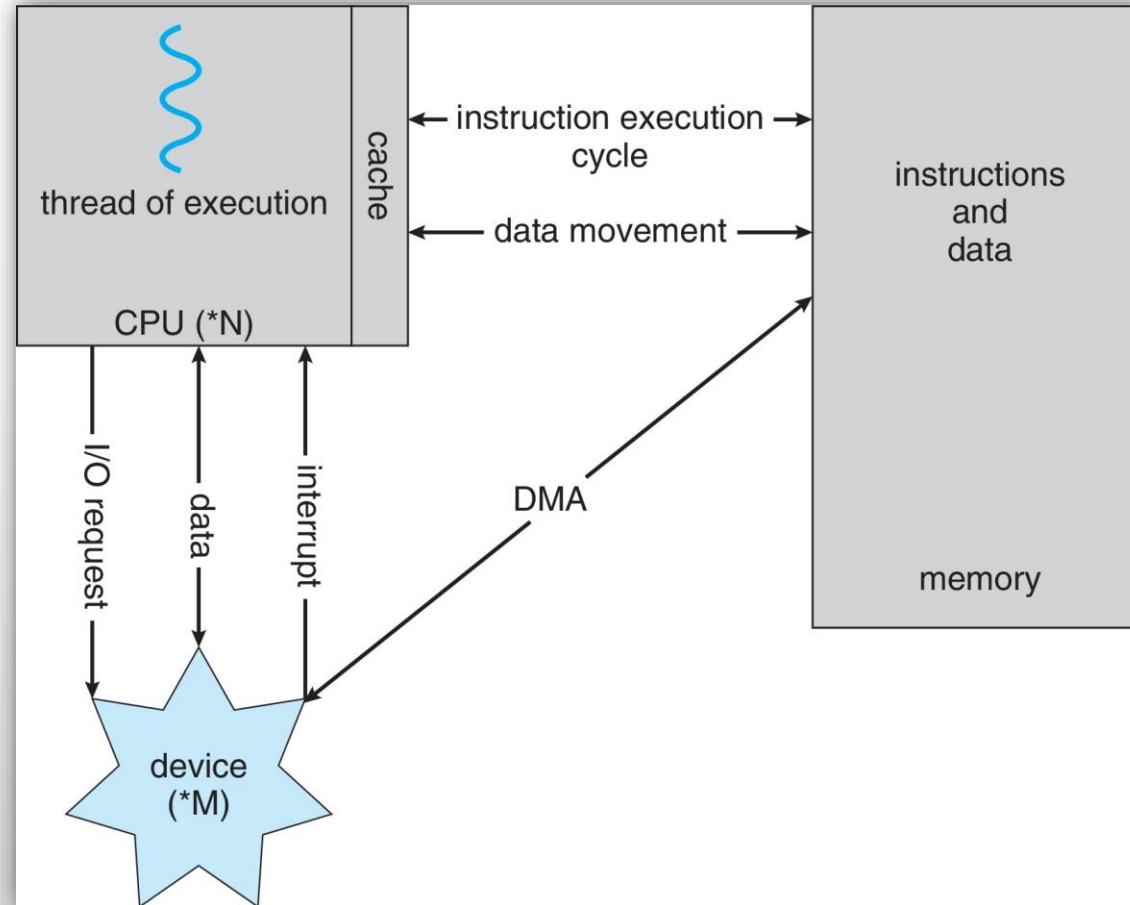


# Kişisel Bilgisayarın Bazı Bileşenleri





# Von Neumann Mimarisi





# İşlemciler

- Tüm işlemler işlemci tarafından yürütülür.
- İşlemci,
  - Komutu anlar
  - Komutu yürütmek için gerekli olan işlemleri gerçekleştirir.
  - Çok çekirdekli yapıda olabilir
  - Birden fazla işlemi aynı anda yürütebilir.
  - Hız, çekirdek sayısı, önbellek boyutu, veri yolu genişliği gibi özelliklere göre farklı performans sergiler.



# İşlemciler

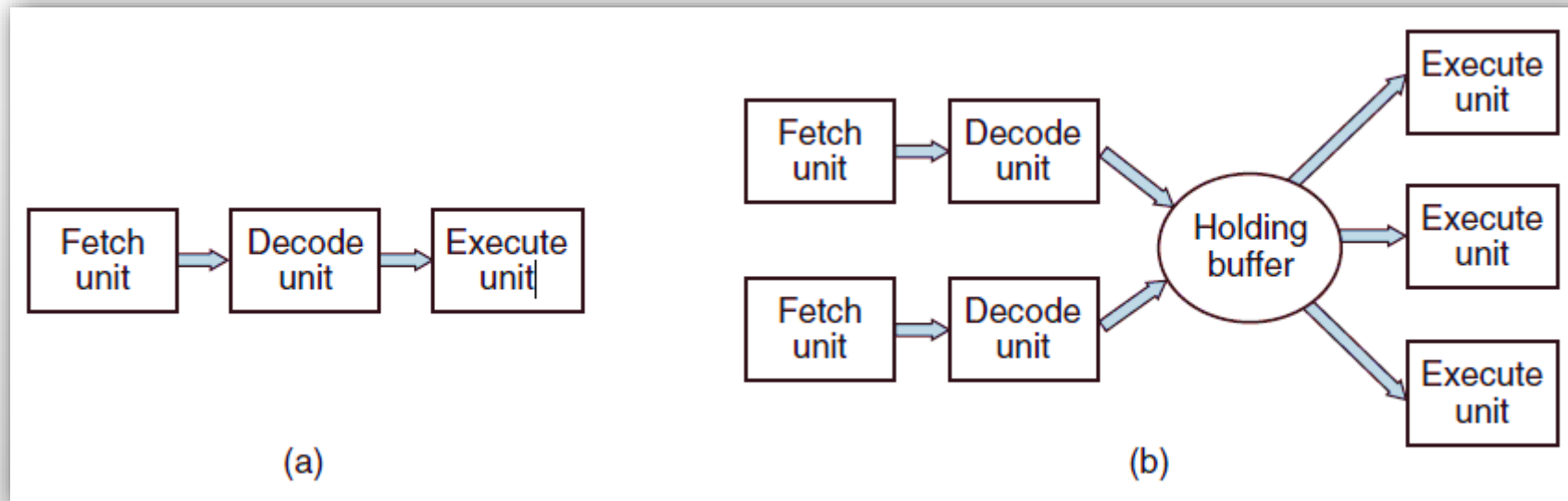
- Bilgisayarın beyni olarak anılır
- Komutu bellekten alır ve yürütür
- **CPU Döngüsü:**
  - Getir (*fetch*), kodu çöz (*decode*), yürüt (*execute*)
- Değişken ve geçici sonuçları saklamak için yazmaçlara sahiptir:
  - Bellekten yazmaca *yükler*
  - Yazmaçtan belleğe *saklar*
- *Program sayacı*: işletilecek bir sonraki komutu gösterir.
- *Yığıt işaretçisi*: geçerli yığıtın en üstünü gösterir.
- *PSW*: program durum sözcüğü, öncelik, mod, ...

# İşlemciler



(a) Üç aşamalı bir boru hattı (*pipeline*).

(b) Bir *superscalar* işlemci





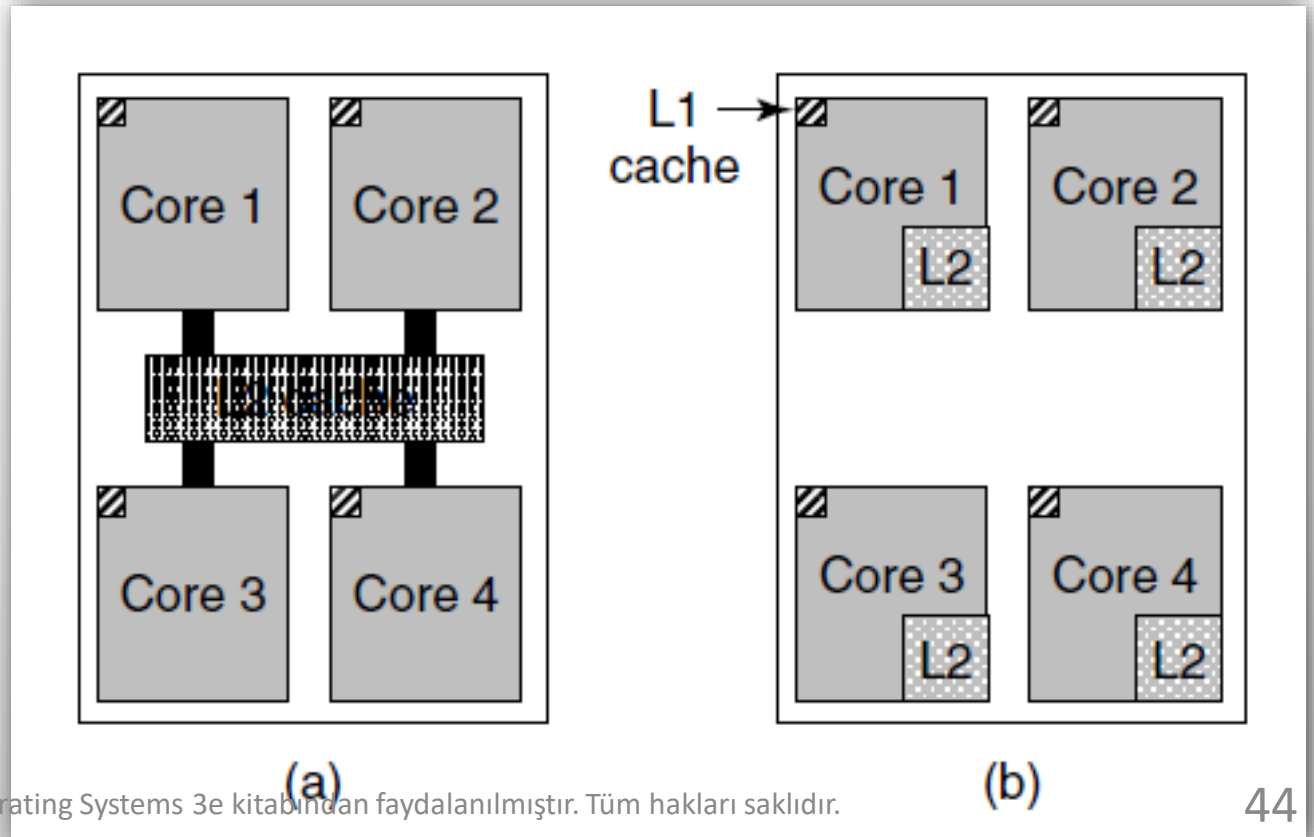
# Bellek

- RAM (*Random Access Memory - Rastgele erişimli bellek*)
- Verileri geçici olarak saklar.
- Verilere işlemci tarafından erişilir.
- İşlemler arasında verileri paylaşmayı ve hızlı erişimi sağlar.
- Bellek boyutu, bilgisayarın performans ve kullanılabilirliğini etkiler.



# Bellek

- (a) Paylaşımlı L2 önbellekli dört çekirdekli bir yonga (*chip*).
- (b) Ayrı L2 önbellekli dört çekirdekli bir yonga (*chip*).







# Önbellek (cache)

- İşlemci tarafından sıklıkla erişilen verileri geçici olarak depolar.
- Yüksek hızlı bellek birimleridir.
- Verilere erişmek için hızlı, düşük gecikmeli (*latency*) bir yol sağlar.
- **L1 önbellek:** En küçük ve en hızlı önbellek türüdür, işlemciye entegredir.
- **L2 önbellek:** İşlemci ile aynı yongadadır, daha büyük, daha yavaştır.
- **L3 önbellek:** En büyük ve en yavaş önbellek türüdür, çip dışında bulunur.

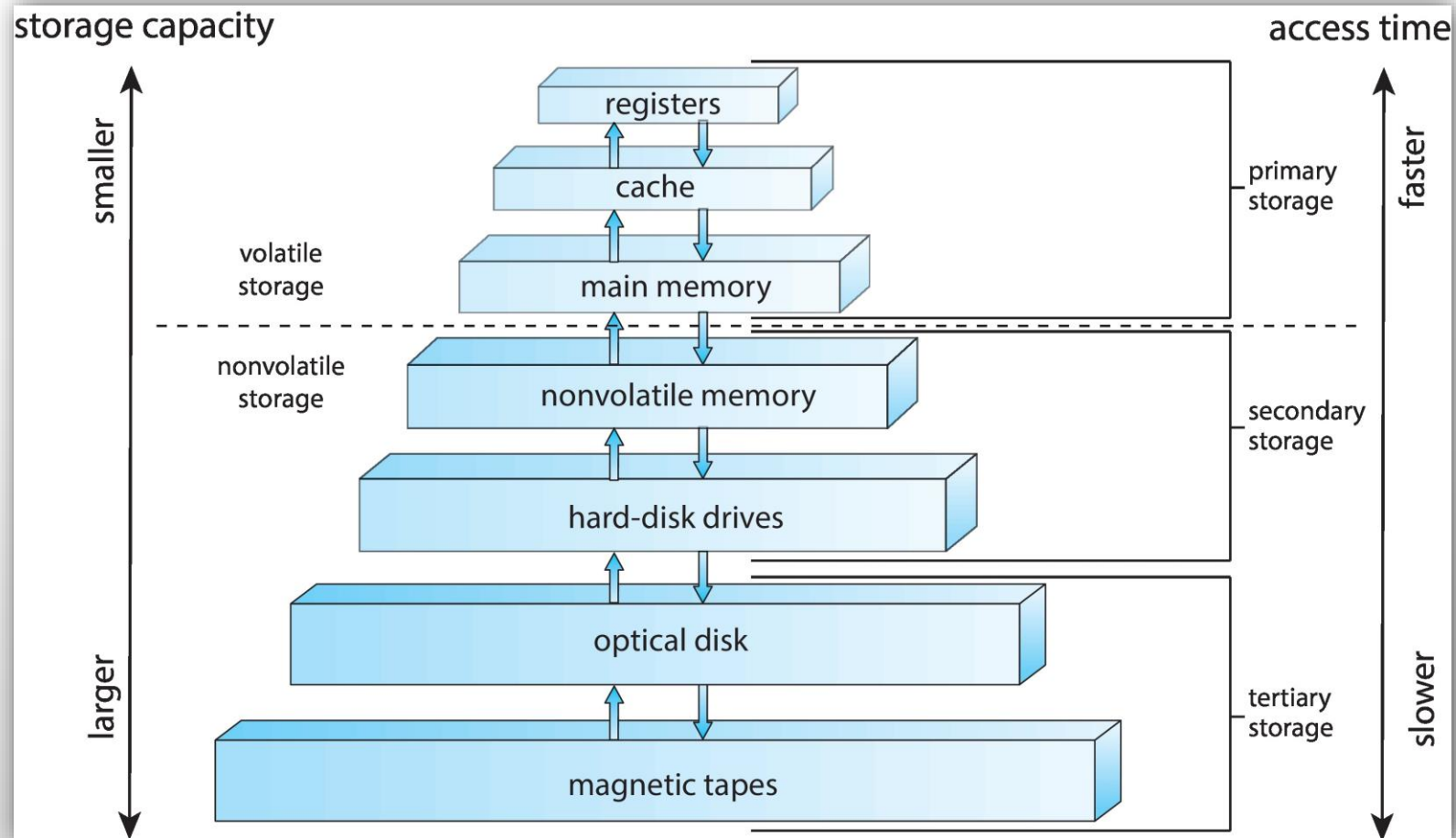


# Ana Bellek

- **RAM**, (*Random access memory*)
  - Rastgele erişimli bellek
  - Değiştirilebilir, hızlı, pahalı
- **ROM**, (*Read only memory*)
  - Salt okunur bellek
  - Değiştirilemez, hızlı, ucuz
  - BIOS, İşletim sistemi yükleyici ..
- **EEPROM**, (*Electrically Erasable Programmable ROM*)
  - Elektriksel olarak silinebilir ve yeniden programlanabilir
  - Yeniden yazılabilir, yavaş
  - Taşınabilir müzik oynatıcılarındaki diskler ..

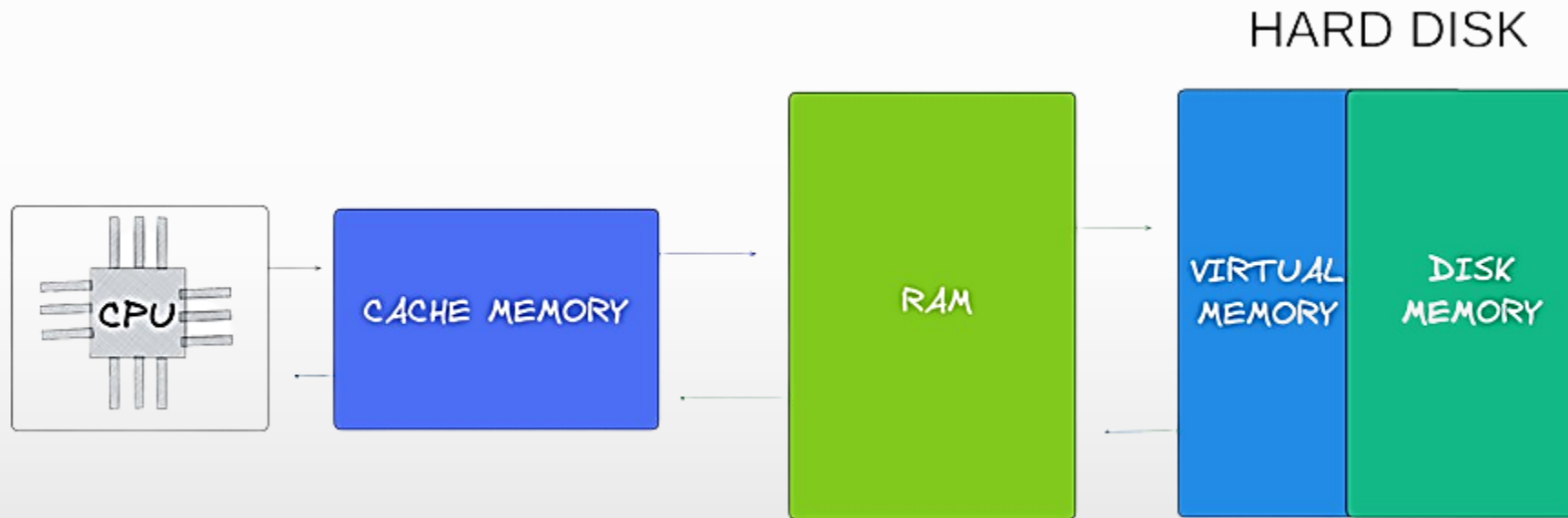


# Bellek Hiyerarşisi





# Bellek Hiyerarşisi





# Bellek Hiyerarşisi

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



# Bellek

- Bilgisayarın verileri alıp depoladığı yerdir.
- İdeal olarak, yonga şeklinde ve büyük olmalıdır.
- Bellek hiyerarşisi göz önünde bulundurulmalıdır.
- **Önbellek satırları:**
  - Bellek, önbellek satırlarına bölünür; sık kullanılanlar önbellekte saklanır
  - *Cache hit/miss*, aranan verinin önbellekte bulunup/bulunamaması.
  - Performansı artırmak için kullanılır.



# Önbellek

- Ana bellek, önbellek satırlarına bölünmüştür (64 bayt).
  - 1. satırda 0-63, 2. satırda 64-127
- Program bir sözcük (word) okuyacağında, donanım verinin önbellekte olup olmadığını kontrol eder.
  - Önbellekte bulunursa, *cache hit* olur (2 döngü (*cycle*)).
  - Bulunamazsa, veri yolu üzerinden ana bellekten talep edilir (maliyetli).
- Önbellek pahalı olduğu için boyutu sınırlıdır.
- Önbellek hiyerarşilere sahip olabilir.



# Önbellek Sorunları

- Yeni bir öge, önbelleğe ne zaman yerleştirilmeli?
- Yeni öge, hangi önbellek satırına konulmalı?
- Yer açmak gerektiğinde, önbellekten hangi öge çıkarılmalı?
- Çıkarılan öge, bellekte nereye yerleştirilmeli?





# Önbellek Sorunları

- Önbelleği yönetirken çeşitli faktörler göz önünde bulundurulur:
- **Boyutu:** ne kadar veri depolanabileceğini, dolayısıyla performansın ne kadar iyileştirilebileceğini etkiler.
- **Değiştirme ilkesi:** önbellek dolduğunda hangi verilerin yer değiştireceğine karar verilmeli, performans ile verimliliği dengeleyen bir ilke seçilmelidir.
- **Tutarlılık:** birden çok işlemcinin, eşzamanlı erişimlerinde önbellek tutarlı ve güncel kalmalıdır.



# Disk

- Verileri uzun süreli, kalıcı olarak saklar.
- Okuma ve yazma işlemleri için veriler disk plakaları üzerinde saklanır.
- Disk sürücüleri, farklı boyutlarda ve kapasitelerde olabilir.
- Disk plakası, okuyucu/yazıcı kafa, motor ve kontrol elemanlarından oluşur.
  - **Disk plakası**, verileri saklamak için kullanılan alandır.
  - **Okuyucu/yazıcı kafası**, veri okuma ve yazma işlemleri için kullanılır.
  - **Motor**, disk plakasını döndürür ve yazıcı kafasını hareket ettirir.
  - **Kontrol elemanları**, disk sürücüsünün işlemlerini yönetir.

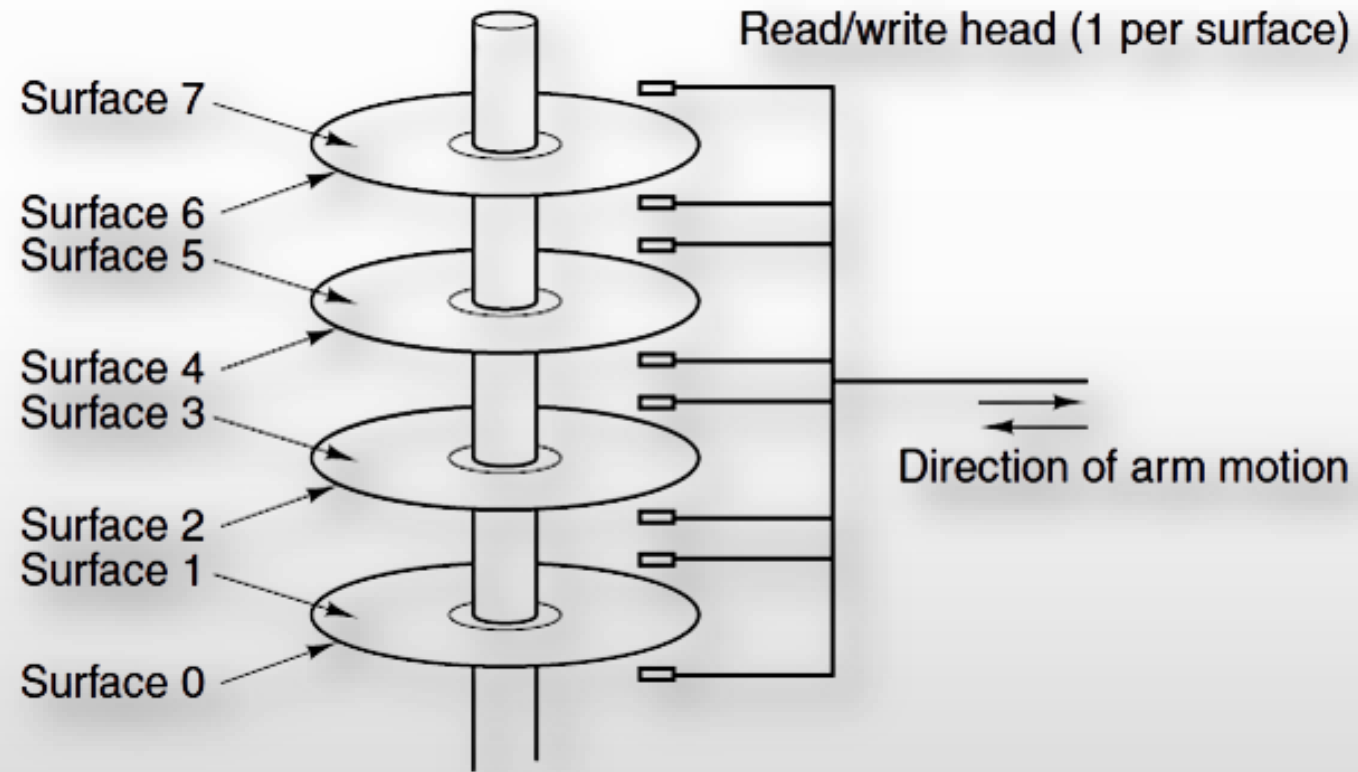


# Disk

- Ucuz, yüksek kapasiteli, ancak yavaştır.
- Mekanik hareketlere ihtiyaç duyar.
- *Arm, track, cylinder, sector, head, checksum*
  - *x diskinde y sektörünü oku* komutunu alır.
  - x ve y bilgisini [cylinder, sector, head] *adres* şekline çevirir.
  - Kolu *doğru silindire* hareket ettirir.
  - Kafanın *doğru sektör* üzerine gelmesini bekler.
  - Sürücüden gelen *bitleri* okur ve saklar.
  - Sağlama yapar. Okunan bitleri *sözcük* olarak bellekte saklar.
- Disk, *Sanal belleğin* uygulanmasına yardımcı olur
  - Bellek yetersiz kaldığında, depolama alanı olarak diskler kullanılır.



# Disk Sürücüsünün Yapısı





# G/Ç Cihazları

- Veri alma/gönderme işlemlerini gerçekleştirir.
- Dış dünya ile bilgisayar arasındaki veri transferini sağlar.
- Klavye, fare, ekran, yazıcı, tarayıcı, ses kartı, kameralar, vb.
- İşletim sistemi tarafından yönetilir.
- Kullanıcının cihazları kullanmasına izin verilir.
- Bilgisayarın performansını ve kullanılabilirliğini etkiler.



# G/Ç Cihazları

- İki parçadan oluşur: *denetleyici* ve *aygıt*
- **Denetleyici:** işletim sistemine basit bir arayüz sağlar.
- **Aygıt sürücüsü:** denetleyiciyle konuşur, komut verir ve yanıt alır.
- İletişim modları:
  - *Meşgul bekleme (busy waiting)*
  - *Kesme (interrupt)*
  - *DMA (direct memory access)*



# Aygıt Sürücüsü

- İşletim sistemi denetleyiciyle konuşur. (komut verir, yanıt alır)
- Denetleyici üreticileri, her işletim sistemi için bir *sürücü* sağlar.
- Sürücü, *çekirdek modunda* çalışır.
- Denetleyici, sürücüyle iletişim kurmak için *yazmaçları (register)* kullanır.
- Üç iletişim modu
  - Sorgulama (*polling*)
  - Kesmeler (*interrupt*)
  - DMA



# Sorgulama, Kesilme ve Doğrudan Bellek Erişimi

- **Sorgulama** (polling), aygıtın hazır olup olmadığını anlamak için aygıtın durumu kontrol edilir.
  - Döngü veya zamanlayıcı kullanılarak aygıt sürekli sorgulanır.
- **Kesilmeler** (interrupt), aygıtın ilgilenilmesi gerektiğini bildirme yöntemidir.
  - Aygıt kesme oluşturduğunda, işletim sistemine bir sinyal gönderir, işletim sistemi mevcut görevini durdurur ve kesmeyi işler.
  - Sorgulamadan daha verimli ve duyarlı bir yöntemdir.
- **DMA**, işlemciyi dahil etmeden doğrudan cihaz ile ana bellek arasında veri aktarma yöntemidir.
  - Büyük miktarda veri aktarımı için hızlı ve verimli bir yöntem sağlar.





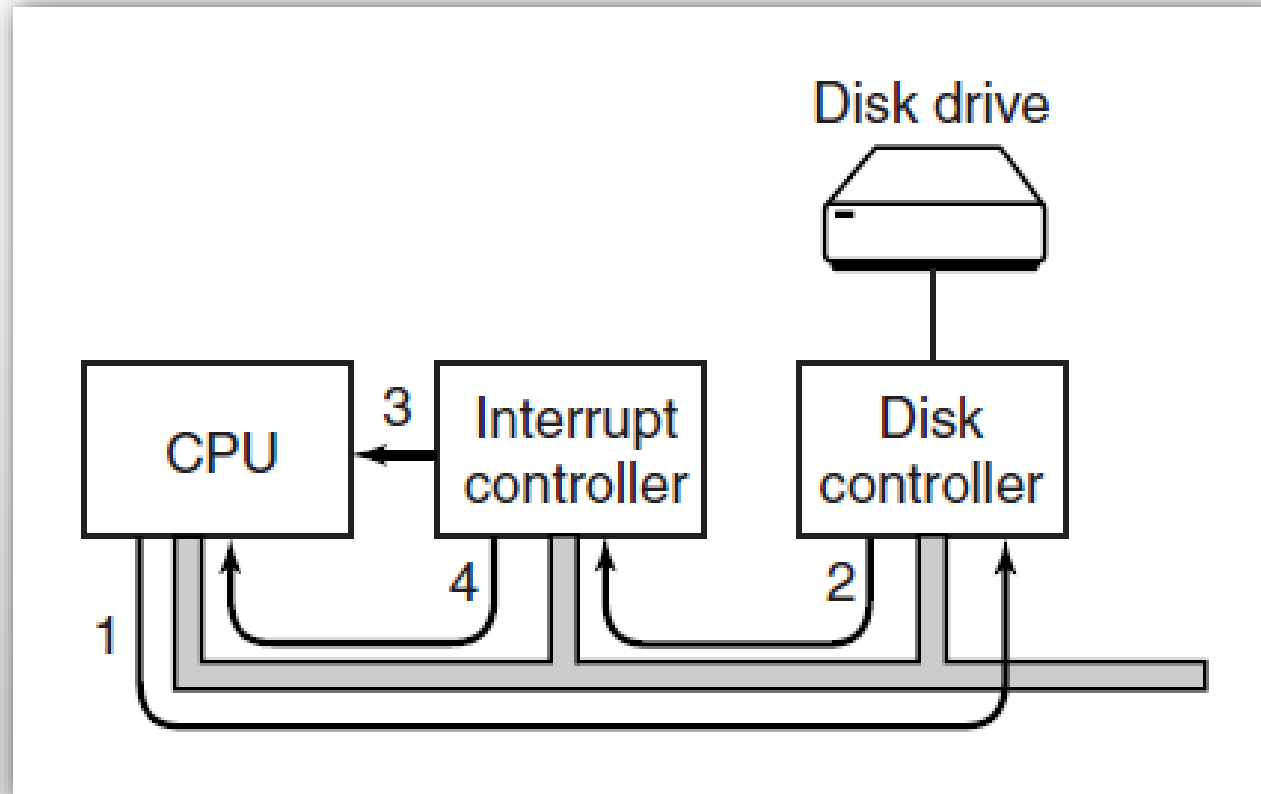
# G/Ç Cihazları - Sorgulama

- Sürücü, denetleyiciye komut verir.
- Sürücü, aygıt hazır olana kadar sorgular. Örneğin,
  - hazır olana kadar yazıcı denetleyicisini sorgula, bir karakter gönder.
- Yüksek işlemci kullanımına neden olur.
- Programlanmış G/Ç olarak adlandırılır, artık kullanılmıyor.



# G/Ç Cihazları - Kesme

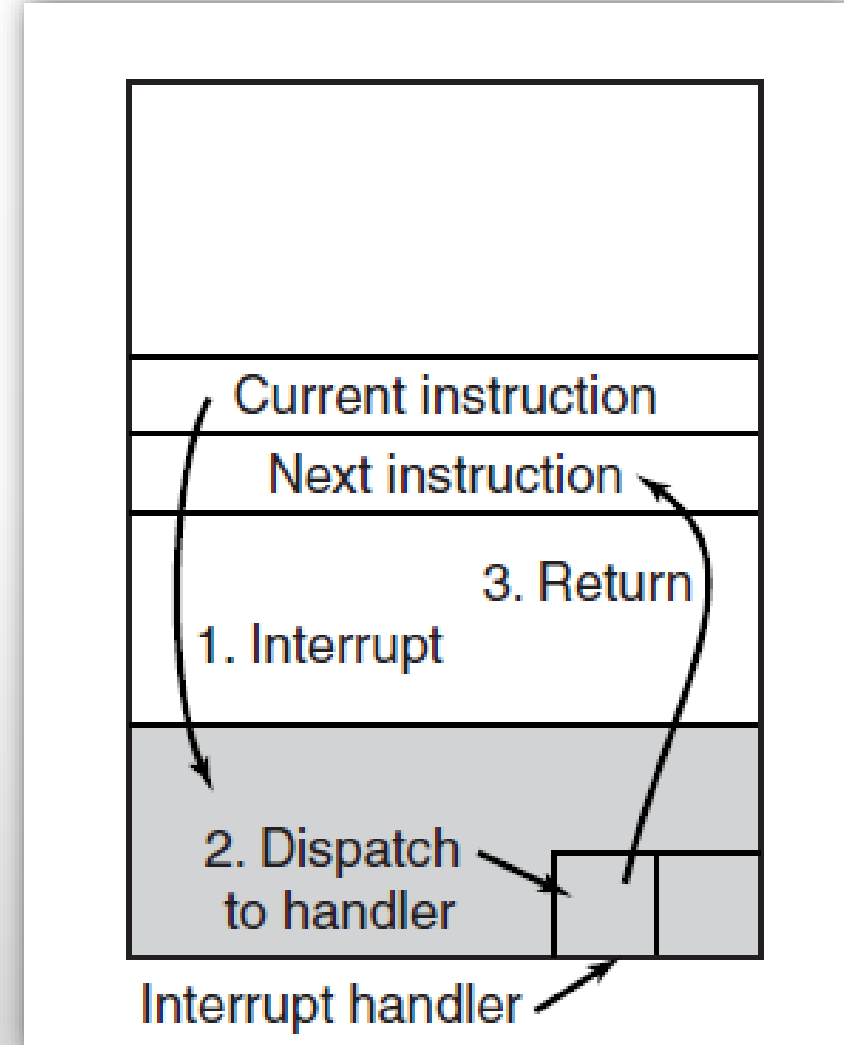
- G/Ç cihazını başlatma ve bir kesme alma adımları





# G/Ç Cihazları - Kesme

- Kesme işleme,
  - kesmeyi alma
  - kesme işleyicisini çalıştırma
  - kullanıcı programına geri dönme
- G/Ç işlemi bittiğinde kesme üretilir
- Bir işlem yapılırken işlemcinin başka işler yapmasına izin verir.





## G/Ç Cihazları - DMA

- Özel (denetleyici) yonga gereklidir.
- Bellek ile veri transferinde işlemci kullanmaktan kaçınır.
- İşlemci, DMA yongasına aktarım hakkında gerekli bilgileri verir.
- Yonga işlem bittiğinde kesme üretir.

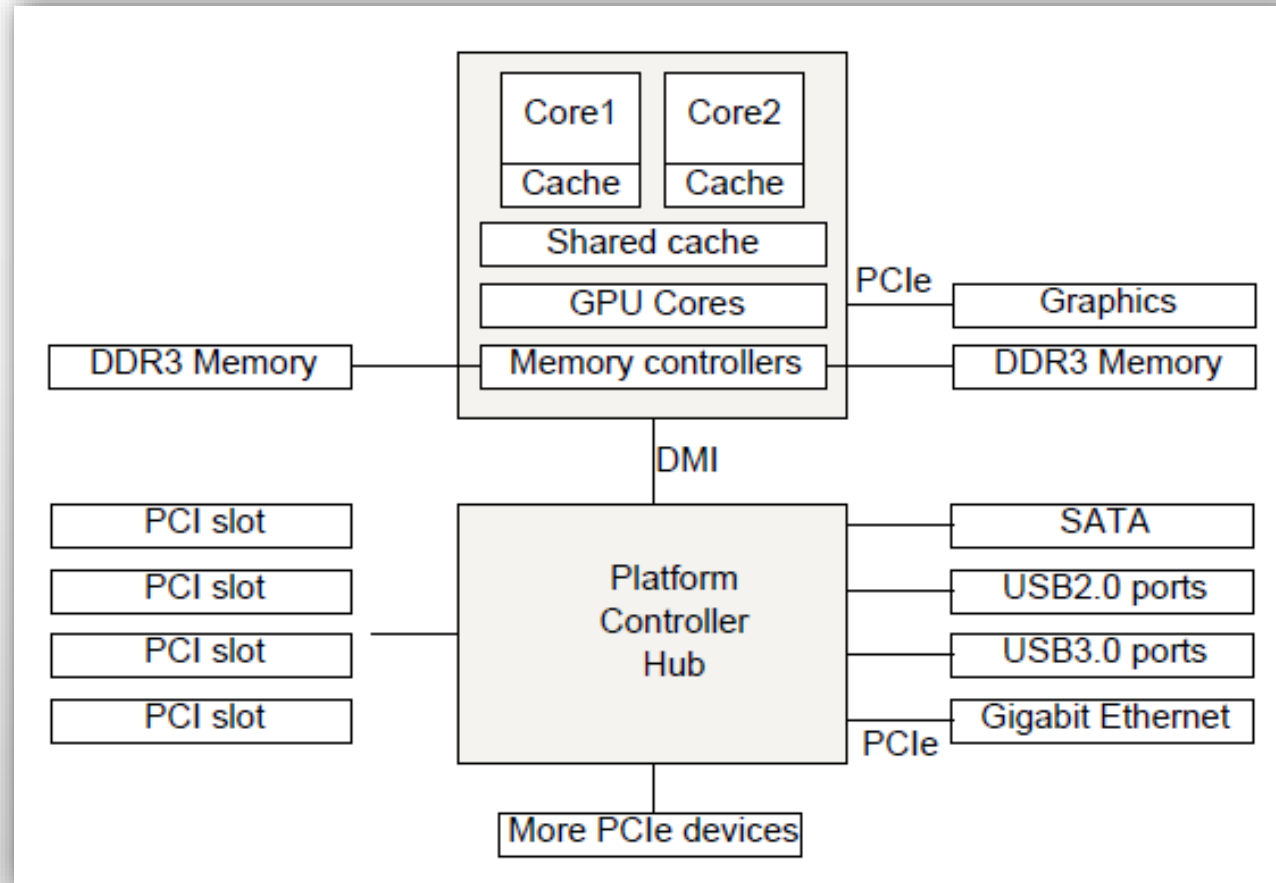


# Veriyolları

- Bilgisayar bileşenleri arasında veri ve sinyallerin taşınması için kullanılır.
  - Örneğin, işlemci, bellek, G/Ç cihazları arasında veri taşır.
- Farklı bant genişliği ve hız değerlerine sahip olabilir.
  - Örneğin, PCI, PCI-Express, USB gibi.
- Veri taşıma yönetiminden işletim sistemi sorumludur.
- Bilgisayarın performansını ve kullanılabilirliğini etkiler.
- Zamanla, daha hızlı (PCI), özelleştirilmiş (SCSI, USB) veri yolları çıktı.

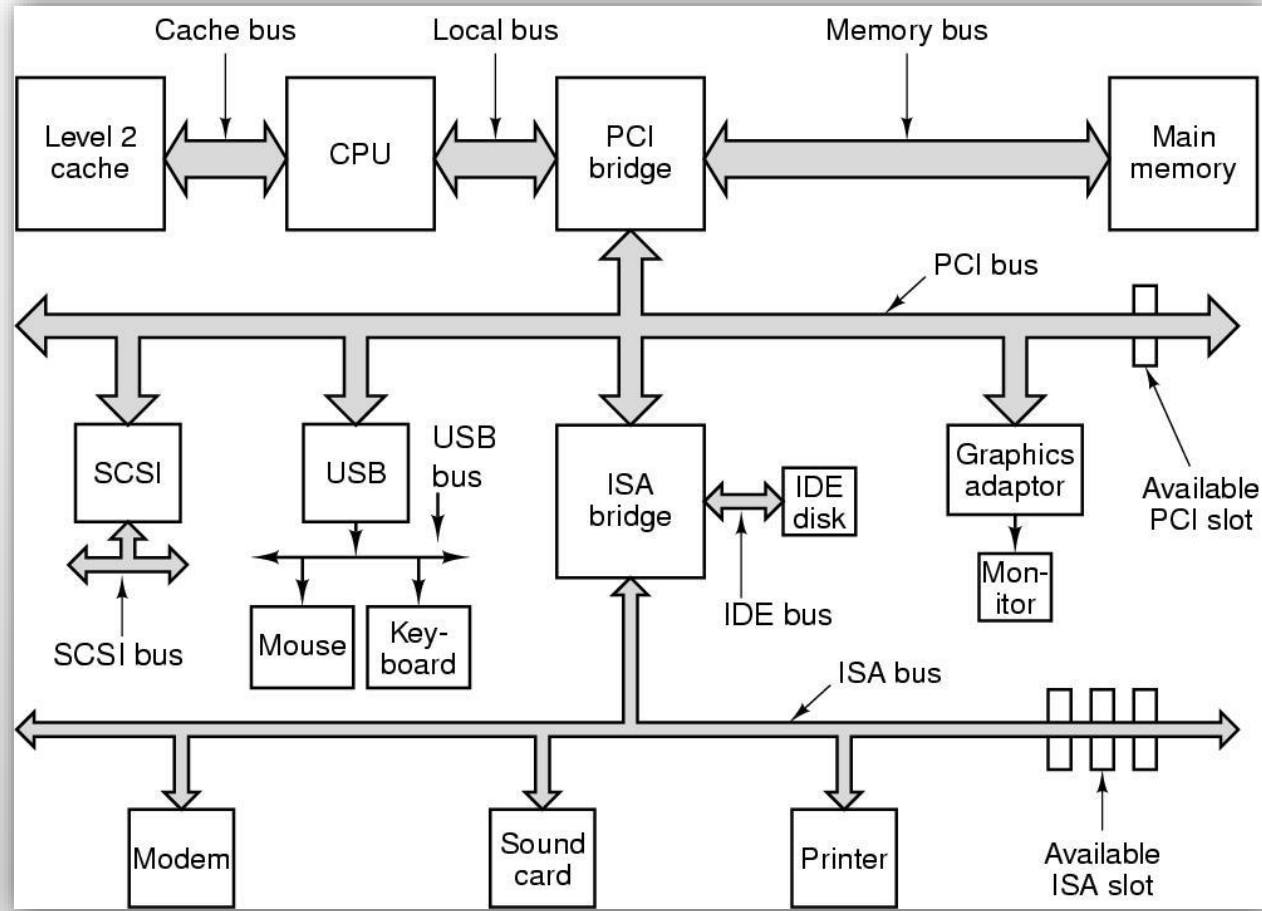


# X86 Sistem Yapısı





# Pentium Sistem Veriyolları





# Veriyolları

- **PCI**, ağ kartı, ses kartı, grafik kartı gibi çevre birimleri ana karta bağlar. Yüksek hızlı bir yerel veri yoludur. Sisteme yeni çevre birimleri eklemek için esnek ve ölçeklenebilir bir çözüm sunar.
- **ISA**, eskiden yaygın olarak kullanılan, PCI'ye kıyasla daha yavaş ve kısıtlı bir veri yolu standardıdır.
- **SCSI**, sabit sürücüler ve teyp gibi depolama aygıtlarını bağlamak için kullanılır. Tek bir veri yolunda birden çok depolama aygıtını bağlayabilen hızlı ve esnek bir çözüm sunar.
- **USB**, klavye, fare, flash sürücüler gibi çevre birimlerini bağlamak için yaygın olarak kullanılan yüksek hızlı bir veri yoludur.
- **IDE**, sabit sürücüler ve CD-ROM gibi depolama aygıtlarını bağlamak için kullanılır. Basit ve uygun maliyetli bir çözüm sunar.





# Bilgisayarın Ayağa Kalkması

- **BIOS** (*basic input output system*): temel giriş/çıkış sistemi.
- Ana kartta yer alır, düşük seviye G/Ç yazılımı.
  1. Bellek, klavye ve diğer temel cihazları kontrol eder.
  2. Önyükleme aygıtını belirler (disket, CD-ROM, disk).
  3. Önyükleme aygıtının ilk sektörü belleğe okunur.
    - Sektör, hangi bölümün aktif olduğunu kontrol eden bir program içerir.
  4. Ardından, ikincil bir önyükleyici belleğe okunur.
  5. İşletim sistemi aktif bölümden okunur.



# İşletim Sistemi Çeşitleri

- Anabilgisayar (*mainframe*)
- Sunucu (*server*)
- Çok işlemcili (*multiprocessor*)
- Kişisel (*personal*)
- Mobil (*handheld*)
- Gömülü (*embedded*)
- Algılayıcı düğüm (*sensor node*)
- Gerçek zamanlı (*real-time*)
- Akıllı kart (*smart card*)



# Anabilgisayar (mainframe) İşletim Sistemi

- Büyük ve karmaşık bilgisayar sistemleri için tasarlanmıştır.
- Çoklu kullanıcı ve çoklu işlem desteği sunar.
- Yüksek performans, yüksek güvenilirlik ve yüksek kullanılabilirlik sunar.
- Büyük veri setleri ve yüksek trafik gerektiren işlemler için kullanılır.
- Endüstriyel ve kurumsal uygulamalar için kullanılır.
- IBM z/OS, Unisys MCP, Fujitsu BS2000/OSD ...



# Sunucu (server) İşletim Sistemi

- Çoklu kullanıcı, çoklu işlem, yüksek kullanılabilirlik için optimize edilmiştir.
- Kurumsal ve endüstriyel ortamlarda yaygın olarak kullanılır.
- Kuruluşların veri merkezi, bulut bilişim gibi servisleri sağlar.
- Sunucu cihazlarında veri depolama, dosya paylaşımı, veritabanı işlemleri, web sunucusu hizmetleri gibi hizmetleri sağlar.
- Windows Server, Linux, UNIX ...



# Çoklu İşlemci (multiprocessor) İşletim Sistemi

- Birden fazla işlemciye sahip bilgisayarlarda kullanılır.
- İşlemlerin paralel olarak gerçekleştirilmesini destekler.
- İşlemlerin işlemci üzerinde aynı anda çalışmasını sağlar.
- İşlemlerin verimli bir şekilde yürütülmesini sağlar.
- Linux, UNIX, Windows ...



# Kişisel (personal) İşletim Sistemi

- Ev kullanıcıları ve küçük işletmeler için tasarlanmıştır.
- Uygulama yüklemek, internette gezinmek, dosyaları yönetmek gibi basit işlemleri gerçekleştirir.
- Kullanıcı dostu arayüzler ve kolay kullanım sunar.
- Windows, MacOS, Linux ...



# Mobil (handheld) İşletim Sistemi

- Taşınabilir cihazlar için tasarlanmıştır. (akıllı telefonlar, tabletler)
- İnternet erişimi, e-posta, sosyal medya, navigasyon, müzik ve video oynatma gibi hizmetler sağlar.
- iOS, Android, Windows Phone ...



# Gömülü (embedded) İşletim Sistemi

- Otomatikleştirilmiş sistemler, cep telefonları, ev otomasyonu, araba ve hava taşıtları gibi cihazlar için tasarlanmıştır.
- Sistem ve cihazların özelliklerini optimize etmek için tasarlanmıştır.
- Linux, VxWorks, QNX ...





# Algılayıcı düğüm (sensor node) İşletim Sistemi

- Algılayıcı, IoT, M2M ağları için tasarlanmıştır.
- Algılayıcı verilerini toplamak, işlemek ve iletmek için kullanılır.
- Enerji verimliliği ve güç tüketimi için optimize edilmiştir.
- TinyOS, Contiki, RIOT ...



# Gerçek Zamanlı (real-time) İşletim Sistemi

- Zaman kritik işlemlerin zamanında yerine getirilmesi için gerçek zamanlı uygulamalar tarafından kullanılır.
- Ses, video, hareket ve diğer algılayıcı verilerini işlemek için kullanılır.
- Tahmin ve kontrol uygulamaları, otomatik sistemler, tren kontrol sistemleri, askeri araçlar gibi alanlarda kullanılır.
- VxWorks, QNX, RTLinux ...



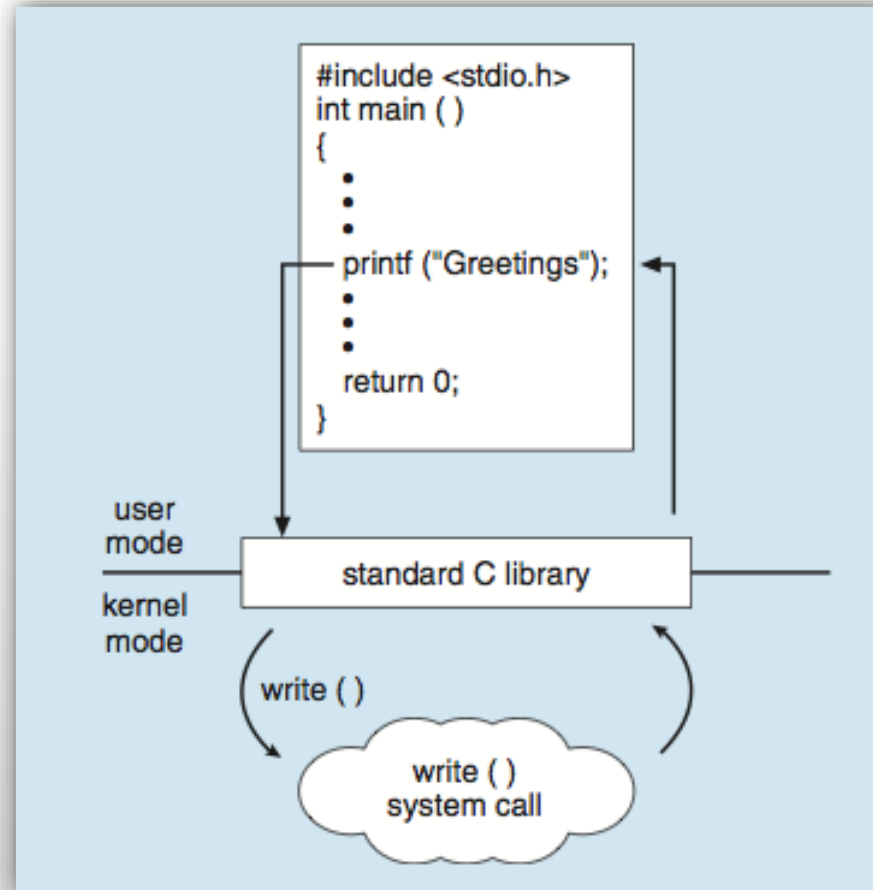
# Akıllı Kart (smart card) İşletim Sistemi

- Akıllı kartlar, küçük boyutlu ve güvenli cihazlar için tasarlanmıştır.
- Kimlik doğrulama, para transferi, elektronik para, kriptografik işlemler ve güvenli veri depolama için kullanılır.
- Kredi kartları, banka kartları, yolcu uçuş kartları, kimlik kartları ve diğer kartlar için kullanılır.
- JavaCard, MULTOS ...



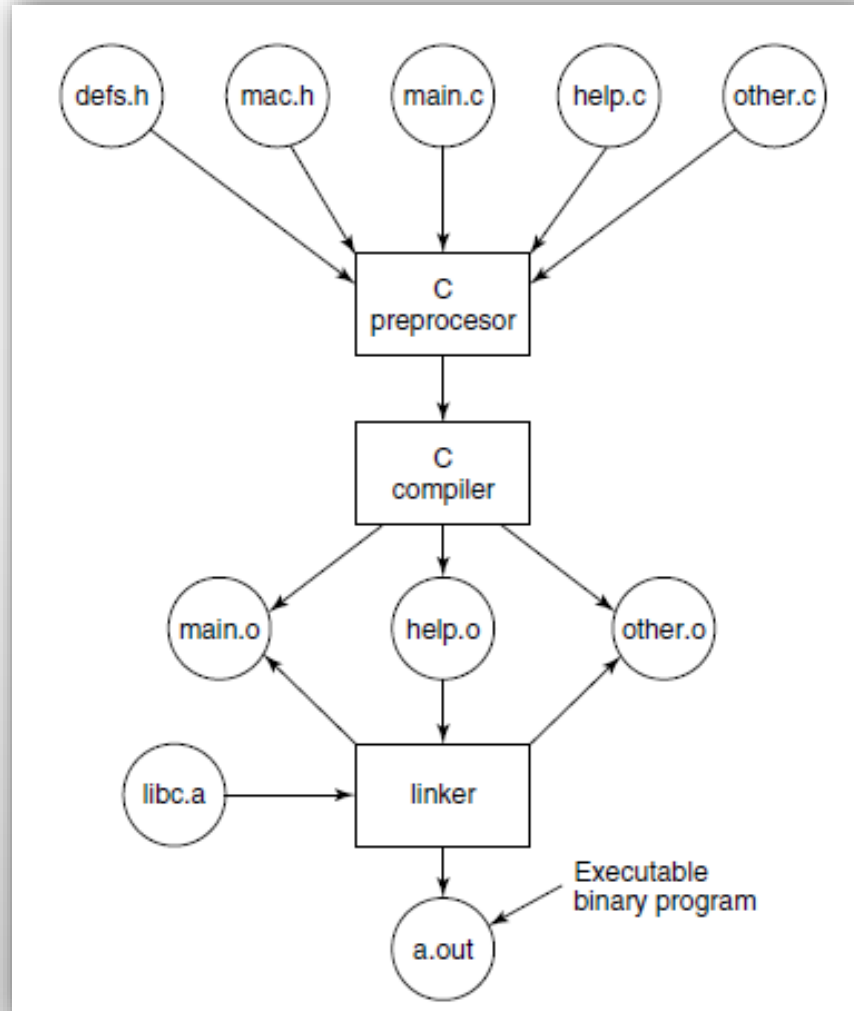
# Standart C Kütüphanesi

■ .





# Yürütülebilir Dosya Oluşturma



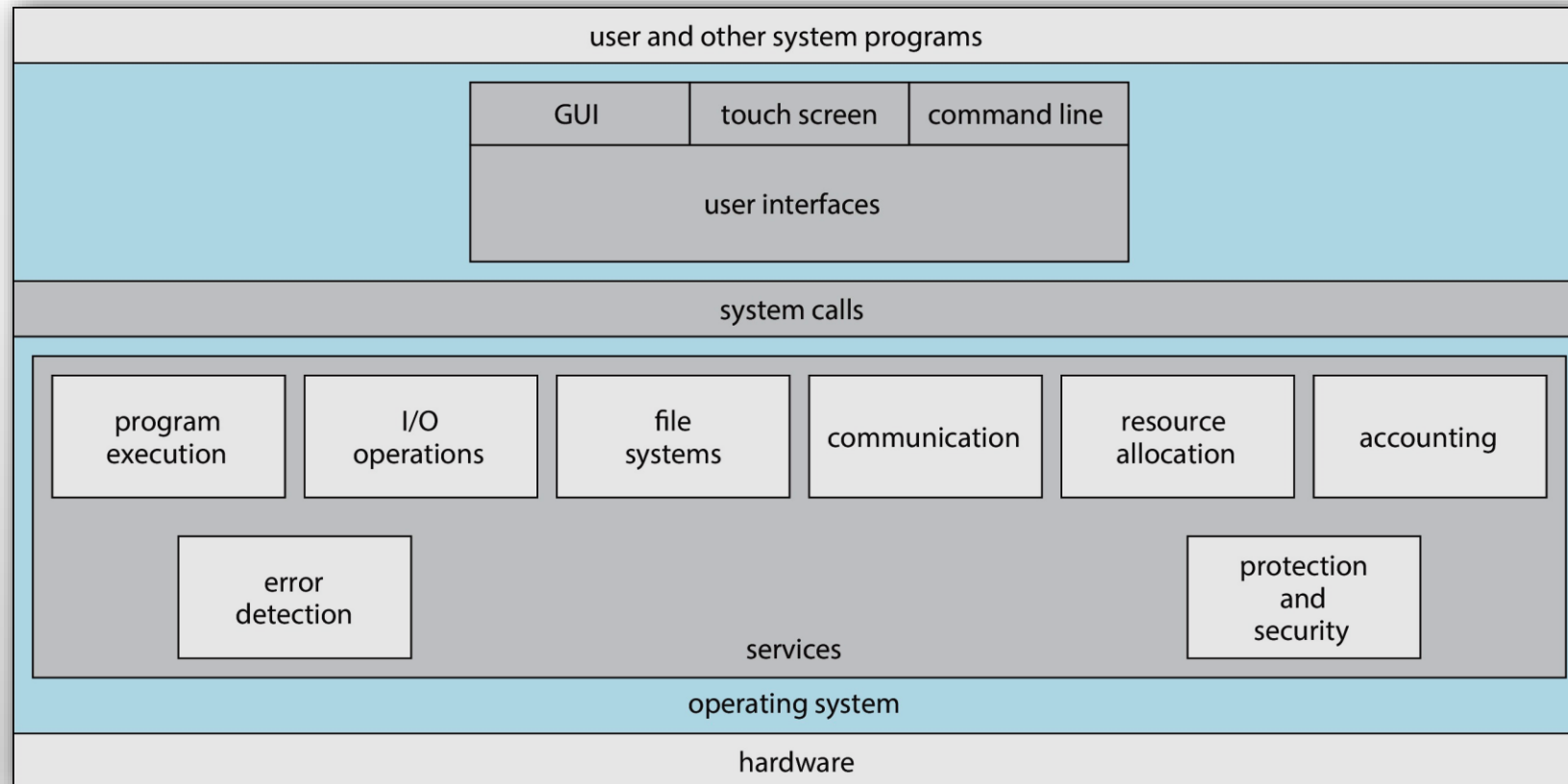


# Yürütülebilir Dosya Oluşturma

- Önışlemci (*preprocessor*)
  - Başlık dosyalarını alır, makroları genişletir, koşullu derlemeyi ele alır.
- Derleyici (*compiler*)
  - `.c` → `.o` , kaynak (*source*) koda göre nesne (*object*) dosyasını oluşturur.
- Bağlayıcı (*linker*)
  - `.o` uzantılı dosyaları birleştirerek yürütülebilir (`exe`) dosyayı oluşturur.



# İşletim Sistemi Servisleri





# Politika ve Mekanizma

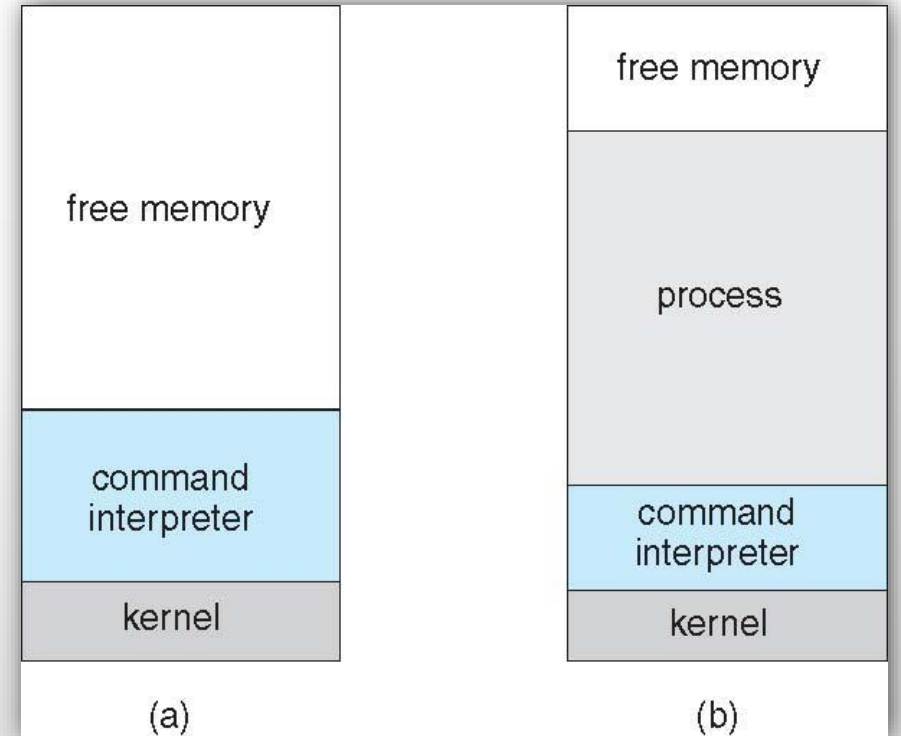
- Politika (*policy*): Ne yapılacak?
- Mekanizma (*mechanism*): Nasıl yapılır?
- Politikalar ne yapılacağını, mekanizma ise nasıl yapılacağını belirler.
- Politikanın mekanizmadan **ayrılması** çok **önemli** bir ilkedir.
- Bu ilke politika kararları daha sonra değiştirilecekse maksimum esneklik sağlar (örneğin, çizelgeleme, önceliklendirme)
- İşletim sistemi tasarlamak, oldukça yaratıcılık gerektirir.





# MS-DOS

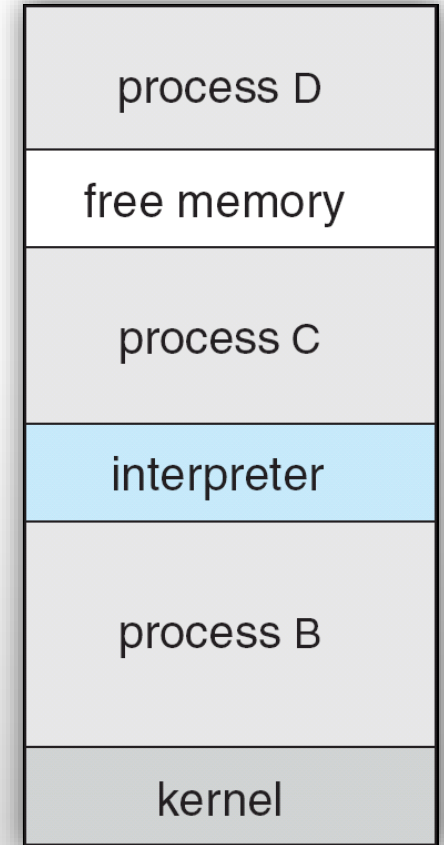
- Tek görevli.
- Sistem ayağa kalktığında kabuk çağrılır.
- Süreç yaratılmaz.
- Tek bellek alanı.
- Program belleğe üzerine yazılarak yüklenir.
- Program sonlandığında kabuk yeniden yüklenir.
- (a) Sistem ayağa kalkarken
- (b) Program çalışırken





# FreeBSD

- UNIX türevi.
- Çok görevli.
- Kullanıcı giriş yapar ve seçilen kabuk yüklenir.
- Kabuk *fork()* komutu ile süreç yaratır.
- *exec()* komutu ile program süreç içerisine yüklenir.
- Kabuk programın sonlanmasını bekler veya kullanıcı komutları ile çalışmaya devam eder.





# Metrik ve Birimleri

Exp.	Explicit	Prefix	Exp.	Explicit	Prefix
$10^{-3}$	0.001	milli	$10^3$	1,000	Kilo
$10^{-6}$	0.000001	micro	$10^6$	1,000,000	Mega
$10^{-9}$	0.000000001	nano	$10^9$	1,000,000,000	Giga
$10^{-12}$	0.000000000001	pico	$10^{12}$	1,000,000,000,000	Tera
$10^{-15}$	0.000000000000001	femto	$10^{15}$	1,000,000,000,000,000	Peta
$10^{-18}$	0.000000000000000001	atto	$10^{18}$	1,000,000,000,000,000,000	Exa
$10^{-21}$	0.000000000000000000001	zepto	$10^{21}$	1,000,000,000,000,000,000,000	Zetta
$10^{-24}$	0.000000000000000000000001	yocto	$10^{24}$	1,000,000,000,000,000,000,000,000	Yotta



SON