

## Giriş

- Bir bilgisayarın iki temel işlevi vardır: **I/O ve processing.**
- **İşletim sisteminin I/O bileşenleri üzerindeki rolü, I/O cihazlarını denetlemek ve I/O işlemlerini yürütmektir.**
- I/O cihazları, **işlevleri, hızları, kontrol yöntemleri açısından farklılık gösterir.**
- I/O cihaz teknolojisinde artan **standartlaşma ile birlikte I/O cihazlarında gelişme hızlanmıştır.**
- I/O cihazlarındaki **çeşitlilik artarken yeni donanım ve yazılım tekniklerine ihtiyaç duyulmaktadır.**
- **Temel I/O donanım bileşenleri, port'lar, bus'lar ve cihaz denetleyicileridir.**
- **İşletim sistemi kernel'ı, cihaz sürücülerini kullanacak şekilde yapılandırılır.**

3

## Konular

- Giriş
- **I/O donanımı**
- I/O arayüz uygulaması
- I/O isteklerinin donanımsal işlemlere dönüştürülmesi
- Performans

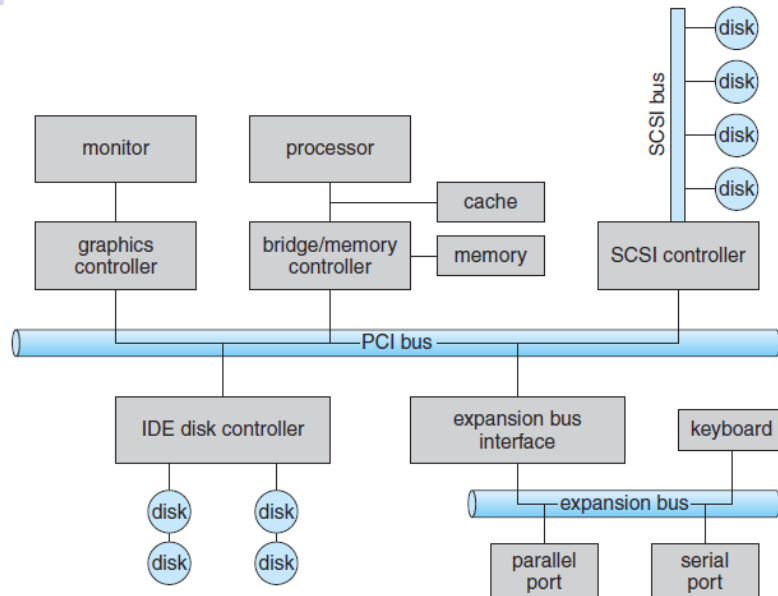
4

## I/O donanımı

- Bilgisayar çok farklı türdeki cihazla çalışabilir.
- Bu cihazlar, **depolama cihazları** (disk, tape), **haberleşme cihazları** (ağ bağlantısı, Bluetooth), **kullanıcı arayüz cihazları** (klavye, mouse, ekran, ses giriş/çıkış) olabilir.
- Farklı kullanım ortamlarında, **girişler/çıkışlar** ve **kullanılan I/O cihazları farklı olabilir**. (Örn. Uçak kontrol bilgisayarında pedal, joystick kullanılır.)
- Bir cihaz, **bilgisayara kablo veya hava aracılığıyla sinyal göndererek haberleşir**.
- Bir **cihaz**, bilgisayarla **bağlantı noktası (port)** aracılığıyla **haberleşir**.
- Cihazların paylaşarak kullandığı **iletim yollarına bus** denir.
- Bus'ların  **sinyalleşme yöntemleri, bant genişlikleri ve bağlantı yöntemleri birbirinden farklıdır**.

5

## I/O donanımı



6

## I/O donanımı

- **PCI (Peripheral Component Interconnection) bus**, CPU ve hafıza ile hızlı birimlerin bağlantısını sağlar.
- **Expansion bus**, daha yavaş birimlerin (seri port, USB, klavye, mouse) bağlantısını sağlar.
- **SCSI (Small Computer System Interface) bus**, disklerin SCSI denetleyiciye bağlantısını sağlar.
- **PCIe (PCI Express) bus** ile 16 GB/s hızında transfer yapılabilmektedir.
- **HyperTransport 3.1 bus** ile 51 GB/s hızında transfer yapılabilmektedir.
- Bir **controller**, port, bus veya cihazla işlem yapan elektronik bileşendir.
- **IDE (Integrated Drive Electronics)**, harddisk, CD-ROM/DVD arayüzüdür.
- Seri port denetleyici basittir ve bir chip içerir.
- SCSI denetleyici karmaşıktır ve ayrı board (**host adapter**) üzerindedir.

7

## I/O donanımı

- Bazı cihazların denetleyicileri kendi üzerindedir (disk sürücü).
- Disk denetleyicileri, kendi mikrokodlarına ve işlemcilerine sahiptir.
- CPU ile I/O cihazı arasındaki haberleşme iki şekilde yapılabilir:
  - Her denetleyici veri yazmak ve okumak için bir grup register'a sahiptir.
    - Bilgisayar I/O cihazıyla iletişim yapmak için bu register'ları kullanır.
    - Özel I/O komutları ile register'lara erişilebilir.
  - Başka bir yöntemde ise, cihaz denetleyici register'ları işlemcinin adres aralığına eşleştirilir (**memory-mapped I/O**).
    - Memory-mapped I/O yönteminde, CPU fiziksel hafızada cihazın register'ları ile eşleştirilen adreslerden okuma ve yazma yapar.
- Bazı sistemler bu iki yöntemi de kullanabilir.

8

## I/O donanımı

- Şekilde bilgisayar için genellikle kullanılan I/O port adresleri görülmektedir.

I/O address range (hexadecimal)	device
000-00F	DMA controller
020-021	interrupt controller
040-043	timer
200-20F	game controller
2F8-2FF	serial port (secondary)
320-32F	hard-disk controller
378-37F	parallel port
3D0-3DF	graphics controller
3F0-3F7	diskette-drive controller
3F8-3FF	serial port (primary)

9

## I/O donanımı

- Bir I/O port genellikle 4 register'dan oluşur: **status, control, data-in ve data-out.**
  - **Data-in register:** Host tarafından giriş almak için okunur.
  - **Data-out register:** Host tarafından çıkış göndermek için yazılır.
  - **Status register:** Cihazın durumunu bildiren bitlere sahiptir.
  - **Control register:** Cihazın başlatılması veya çalışma durumunun değiştirilmesi için kullanılır (seri port için -> parity bit, word length, full/half duplex, transfer hızı).
- **Data register'ları** genellikle **1-4 byte** boyutundadır.
- Bazı denetleyiciler FIFO yapısına sahip chip'ler kullanarak veri tutarlar.
- **FIFO chip'ler burst data'yı tutmak için faydalıdır.**
- CPU ile I/O cihazı arasındaki haberleşme **polling** veya **interrupt** yöntemi ile yapılabilir.

10

## I/O donanımı

### Polling

- Host ile controller arasındaki protokol çok karmaşıktır, ancak **handshaking basittir.**
- Host ile controller arasındaki ilişki **producer-consumer** şeklinde modellenebilir.
- Controller kendi durumunu status register ile gösterir.
- Controller meşgul ise **busy bit'i set (1)** eder boş ise **clear (0)** yapar.
- Host, command register içindeki **command-ready bit** ile controller'a komut gönderdiğini iletir.
- Controller, sürekli command register'ı kontrol eder ve gelen komutun işlemini gerçekleştirir.

11

## I/O donanımı

### Polling

- Aşağıda host bir port aracılığıyla controller'a çıkış göndermektedir:
  1. Host, status register'daki busy bit'i clear yapıncaya kadar okuyarak bekler.
  2. Host, command register'daki write bit'i set eder, bir byte'ı data-out register'ına yazar.
  3. Host, command-ready bit'i set eder.
  4. Controller, command-ready bit'in set edildiğini görür ve busy bit'i set eder.
  5. Controller, command register'ı okur ve write komutu olduğunu görür.
  6. Controller, data-out register'ını okur ve ilgili cihaza yazma işlemini yapar.
  7. Controller, command-ready bit'ini clear yapar, status register'daki error bit'ini clear yapar (işlemin başarılı bittiğini gösterir) ve status register'daki busy bit'i clear yapar.
- Yukarıdaki döngü her byte için sürekli tekrar eder.

12

## I/O donanımı

### Polling

- Host'un **busy bit'i** sürekli kontrol etmesi **busy-waiting** veya **polling** diye adlandırılır.
- **Device controller ve cihaz çok hızlı ise yöntem uygulanabilir.**
- Eğer **bekleme süresi artarsa**, host **başka bir göreve geçiş** yapar.
- **Çoğu bilgisayar mimarisinde** bir cihazın durumunun sorgulanması (poll) **3 cycle ile tamamlanır.**
  - Cihaz register'ının okunması
  - Durum bitinin mantıksal işlemi (AND)
  - Branch if not zero (busy bit 1 ise döngü tekrar eder.)
- Bazı cihazlarda hızlı dönüş yapılmazsa veri kaybolur (seri port, klavye).
- Sıklıkla **meşgul olan cihazlarda polling etkin yöntem değildir.**

13

## I/O donanımı

### Interrupt

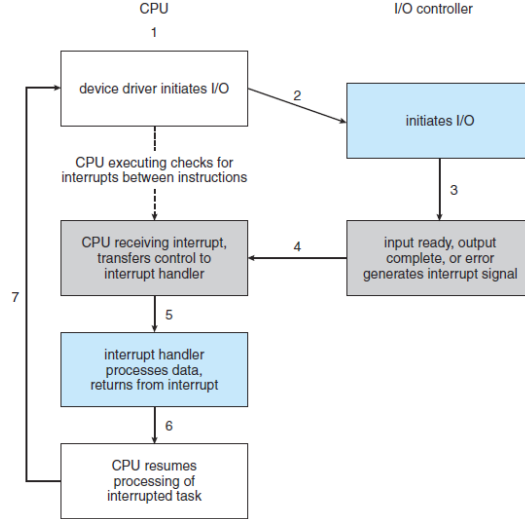
- CPU donanımı **bir bit interrupt-request line'a sahiptir.**
- CPU, her **instruction cycle'dan** hemen sonra **interrupt bit'ini kontrol eder.**
- CPU **interrupt algıladığı anda**, **process'in durumunu saklar** ve **interrupt-handler routine'e geçer.**
- Interrupt-handler routine hafızada sabit bir adrese sahiptir.
- Tek kullanıcı bilgisayarlar bile saniyede yüzlerce interrupt yönetir.
- **Sunucu sistemleri saniyede yüzbinlerce interrupt yönetir.**

14

## I/O donanımı

### Interrupt

- Interrupt tarafından başlatılan I/O döngüsü aşağıda görülmektedir.



15

## I/O donanımı

### Interrupt

- CPU'lar genellikle **nonmaskable** ve **maskable** olarak iki tür interrupt'a sahiptir.
- **Nonmaskable** interrupt'lar geldiğinde **engellenemez** (hafıza hatası).
- **Maskable** interrupt'lar CPU tarafından **kapatılabilir** (program hatası).
- Interrupt mekanizması bir adres (numara) kabul eder ve **interrupt-handling routine'i** seçer.
- Tüm adresler **interrupt vector** tablosunda saklanır.

16

## Interrupt

- Intel Pentium işlemciler için interrupt vector tablosu aşağıdadır.

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

17

## Interrupt

- 0-31 arasındaki olaylar nonmaskable tanımlanmıştır ve hata sinyalleri için kullanılır.
- 32-255 arasındaki olaylar maskable tanımlanmıştır ve cihazların ürettiği hatalar için kullanılır.
- Interrupt mekanizması önceliklendirme için de kullanılabilir.
- Yüksek öncelikli bir interrupt geldiğinde düşük öncelikli interrupt çalışıyorsa kesilir.
- Interrupt'lar sıfıra bölme hatası gibi çok farklı istisnaların (exception) yönetiminde de kullanılmaktadır.

18



## I/O donanımı

### Direct memory access

- Büyük boyutta veri transfer eden cihazlarda, genel amaçlı mikroişlemciyi kullanmak etkin çözüm değildir.
- Genel amaçlı mikroişlemci, controller register'ı aracılığıyla her byte'ı ayrı aktarır (programmed I/O, PIO).
- Çoğu bilgisayar bu tür veri aktarımları için özel amaçlı işlemci (direct-memory-access, DMA) kullanır.
- Host, hafızaya DMA komut bloğunu yazarak DMA ile transferi başlatır.
- Yazılan blok, kaynak pointer, hedef pointer ve aktarılabilecek byte sayısını içerir.
- CPU tarafından komut bloğunun adresi DMA controller'a iletilir.
- DMA controller, aktarımı memory bus üzerinden gerçekleştirir.

19

## I/O donanımı

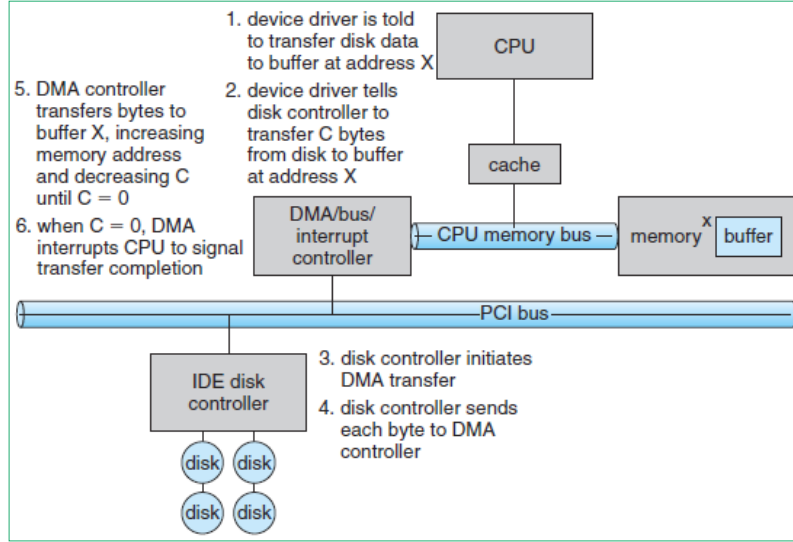
### Direct memory access

- DMA controller ile device controller arasındaki handshaking DMA-request ve DMA-acknowledge bağlantılarıyla sağlanır.
- Device controller, DMA-request sinyali ile istek iletir.
- DMA-controller,
  - Memory bus'ın kullanımını alır,
  - Hafıza adresini memory adres bus'a yerleştirir,
  - DMA-acknowledge sinyalini gönderir.
- Device controller, DMA-acknowledge sinyalini alınca veri transferine başlar ve DMA-request sinyalini kaldırır.
- Veri aktarımı tamamlandığında, DMA controller tarafından CPU'ya interrupt iletir.

20

## I/O donanımı

### Direct memory access



21

## I/O donanımı

### I/O donanımı

- İşletim sisteminin I/O donanım bileşenleri ve özellikleri:
  - Bus
  - Controller
  - I/O port ve register'lar
  - Host ve device controller arasında handshaking
  - Handshaking işleminin polling veya interrupt ile gerçekleştirilmesi
  - Büyük boyuttaki transferler için DMA controller kullanılması
- İşletim sistemi çok farklı cihazlarla çalışabilecek şekilde tasarlanmalıdır.
- İşletim sistemi, yeni cihazlarla işlem yapabilir olmalıdır.

22

## Konular

- Giriş
- I/O donanımı
- I/O arayüz uygulaması
- I/O isteklerinin donanımsal işlemlere dönüştürülmesi
- Performans

23

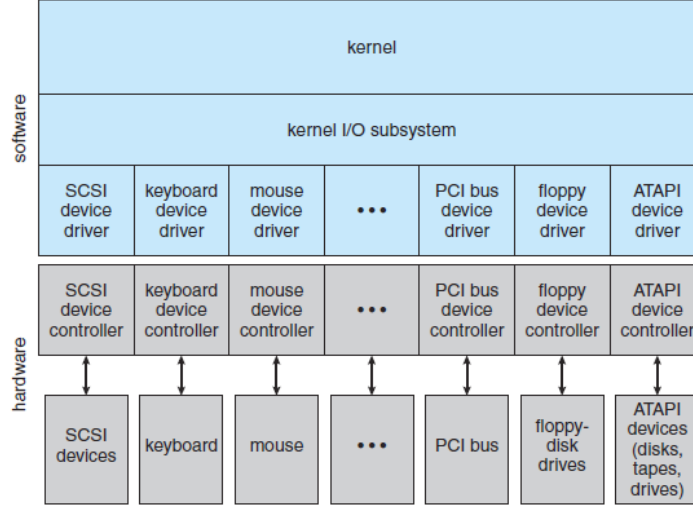
## I/O arayüz uygulaması

- Bir uygulamanın, **bir dosyayı açarken hangi tür disk üzerinde olduğunu bilmesi gerekmemelidir.**
- **Yeni disk ekleme** veya farklı cihazları eklemenin işletim sisteminde herhangi bir **soruna/hataya yol açmaması** gereklidir.
- **Karmaşık diğer problemlerde olduğu gibi, işletim sistemi bu tür işlemlerde abstraction, encapsulation ve software layering yöntemlerini kullanır.**
- **Soyutlama** için arayüz (interface) oluşturulur.
- Cihazların birbirinden farklı özellikleri device driver'larda **encapsulate** edilir.

24

## I/O arayüz uygulaması

- **Device driver katmanı**, device controller'lardaki farklılıkları kernel I/O alt sisteminden gizler, **I/O sistem çağrıları cihaz farklılığından etkilenmez.**



25

## I/O arayüz uygulaması

- **Device driver kullanılması, işletim sistemi geliştiricisinin de cihaz üreticisinin de işini kolaylaştırır.**
- Cihaz üreticisi **ya host controller ile uyumlu cihaz tasarımı yapar, ya da geliştirdiği cihazın sürücüsünü yazar.**
- Böylelikle yeni geliştirilen **donanım sisteme doğrudan takılarak kullanılabilir.**
- **Her işletim sistemi, device driver için kendi standartlarına sahiptir.**
- Bu yüzden, **her cihaz farklı işletim sistemleri için sürücü sağlamalıdır (Windows, Linux, Mac OS X).**

26

## I/O arayüz uygulaması

- Cihazlar farklı açılardan birbirine göre farklı özelliklere sahiptir:
  - **Karakter stream/blok:** Cihaz **karakter karakter** veya **blok şeklinde veri transfer eder.**
  - **Sıralı/rastgele erişim:** Cihaz belirli bir **sırada veri aktarımı yapar** veya **rastgele konumlanarak veri aktarımı yapar.**
  - **Senkron/asenkron:** **Senkron** cihazlarda **response time öngörülebilir.** **Asenkron** cihazlarda başka cihazlara/olaylara bağlı olduğundan **response time öngörülemez.**
  - **Paylaşılabilir/adanmış:** **Paylaşılabilir cihaz çok sayıda process tarafından kullanılabilir.** **Adanmış olan sadece bir cihaz tarafından kullanılabilir.**
  - **Hız:** Cihazlar **Byte/s** seviyesinden **GB/s** seviyesine kadar farklı hızlarda olabilir.
  - **Read-write, read only, write only:** Cihazlardaki **işlem türü farklı olabilir.**

27

## I/O arayüz uygulaması

### *Clock ve timer*

- Çoğu bilgisayar **clock ve timer'a sahiptir** ve aşağıdaki işlevleri sağlarlar:
  - **Şimdiki zamanı verir.**
  - **Geçen süreyi verir.**
  - **T zamanında X işleminin tetiklenmesini sağlar.**
- Zaman hassasiyeti olan uygulamaların yanı sıra **işletim sistemi de clock ve timer'ı sıklıkla kullanılır.**
- **Uygulamalar belirli zaman aralıklarında belirli işleri başlatmak için timer kullanırlar.**
- Ağ alt sistemi, başlatılan işlemlerin belirli süre sonunda **(timeout)** iptal edilmesi için kullanır.

28

## I/O arayüz uygulaması

### *Blocking ve nonblocking I/O*

- Bir uygulama **blocking I/O** başlattığında, uygulamanın çalışması **askıya alınabilir**.
- Uygulama işletim sisteminin **run kuyruğundan wait kuyruğuna alınabilir**.
- Bu tür uygulamaların **çalışma süresi öngörülemezdir**.
- **Nonblocking I/O** işlemlerinde işlem başladıktan sonra **kesilemez veya askıya alınamaz**.
- **Nonblocking I/O** olan **klavye ve mouse** girişleri anında ekrana yansıtılmalıdır.
- **Audio/video** uygulamaları da **nonblocking I/O** şeklinde çalıştırılırlar.

29

## Konular

- Giriş
- I/O donanımı
- I/O arayüz uygulaması
- **I/O isteklerinin donanımsal işlemlere dönüştürülmesi**
- Performans

30

## I/O isteklerinin donanımsal işlemlere dönüştürülmesi

- Device driver ile device controller handshaking yaptıktan sonra, **fiziksel olarak istek yapılan işlemin nasıl gerçekleştirileceğinin belirlenmesi gereklidir.**
- Bir **disk okuma işleminde, uygulama dosya adı ile veriyi okumak ister.**
- **File sistem, dosya adı ile dosyaya ayrılmış alanı eşleştirir.**
- MS-DOS işletim sisteminde, **file-access table, dosya adına tahsis edilmiş disk bloklarını gösterir.**
- Dosya adı ile disk controller arasındaki bağlantı için **donanımın port adresi** veya **memory-mapped register** kullanılır.
- **MS-DOS'ta dosya adındaki iki nokta üst üste işaretinden önceki harf cihazı tanımlar (c:, d:).**
- **c: veya d: cihaz tablosundan özel port adresine eşleştirilir.**

31

## I/O isteklerinin donanımsal işlemlere dönüştürülmesi

- Bir **blocking read isteğinin yaşam döngüsü** aşağıdadır:
  1. **Process, blocking read() sistem çağrısını başlatır.**  
Dosya daha önce açılmıştır.
  2. **Sistem çağrı kodu parametrelerin doğruluğunu ve hazır olduğunu kontrol eder. Hazırsa, I/O isteği tamamlanmıştır.**
  3. **Hazır değilse, fiziksel I/O gerçekleştirilir.**  
Process **run kuyruğundan wait kuyruğuna alınır ve I/O isteği yapılır.**  
I/O alt sistemi **device driver'a isteği iletir.**
  4. **Device driver, kernel buffer'ından yer ayırır ve veriyi buraya alır.**  
**Driver, device controller'a control register aracılığıyla gerekli komutları gönderir.**
  5. **Device controller, veri transferini donanım üzerinde gerçekleştirir.**
  6. **Driver, durum bilgisini ve veriyi, poll yöntemiyle veya DMA ile hafızaya alır. DMA ile yapılırsa bir interrupt oluşturulur.**

32

## I/O isteklerinin donanımsal işlemlere dönüştürülmesi

7. **Interrupt-vector tablosu** ile doğru **interrupt handler routine'i** seçilir, gerekli **veri okunur**, **device driver'a** sinyal ile bildirilir ve **interrupt routine'e** dönülür.
8. **Device driver**, I/O işleminin tamamlandığını **kernel'a** bildirir.
9. **Kernel**, alınan **veriyi process'e** aktarır ve **process'i** wait kuyruğundan **ready kuyruğına** aktarır.
10. **Scheduler**, **process'i** CPU'ya atadığında ise çalışmasına kaldığı yerden devam eder.

33

## Konular

- Giriş
- I/O donanımı
- I/O arayüz uygulaması
- I/O isteklerinin donanımsal işlemlere dönüştürülmesi
- **Performans**

34



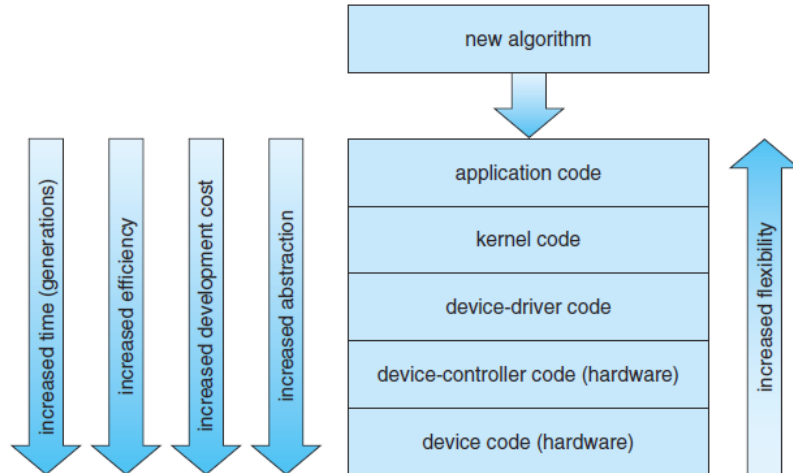
## Performans

- I/O, sistem performansını etkileyen en önemli faktördür.
- CPU, device driver kodunu yoğunlukla çalıştır, block ve unblock process'ler arasında etkin çalışmayı sağlar.
- Modern bilgisayarlar her saniye binlerce interrupt yürütür.
- Her interrupt context switch gerektirir ve maliyeti yüksek bir iştir.
- I/O etkinliğini artırmak için farklı yöntemler kullanılabilir:
  - Context switch sayısını azaltmak.
  - Cihaz ve uygulama arasında aktarım yaparken hafızadaki kopyalama işlemlerini azaltmak (kernel buffer-application adres space).
  - Büyük veri aktarımları yaparak interrupt sıklığını azaltmak.
  - DMA kullanımını artırmak.
  - Basit işlemleri donanıma aktararak controller tarafından yapılmasını sağlamak.
  - CPU, memory, bus ve I/O performansını dengelemek.

35

## Performans

- I/O işlemlerinin hangi düzeyde gerçekleştirildiği, **geliştirilme süresi, etkinliği, geliştirilme maliyeti, soyutlama düzeyi ve esnekliği** açısından önemlidir.



36