

# Ulazno/Izlazni (I/O) Sistemi

- **I/O Ulazno/Izlazni Hardver**
- **Aplikacijski I/O Interfejs**
- **Kernelnski I/O Podsystem**
- **Pretvaranje I/O Zahteva u Hardverske Operacije**
- **Tokovi Podataka (Streams)**
- **Performanse**

# I/O Hardver

## ■ Opšte **kategorije** I/O uređaja:

- ☞ **memorijski uređaji**
- ☞ **uređaji za prenos podataka**
- ☞ **uređaji za korisnički intrefejs**

## ■ **Opšti koncepti:**

- ☞ **Controller** (host adapter) realizovan kao:

- ☞ **port** (4 vrste registara :kontrolni, statusni, ulazni za podatke, izlazni za podatke )

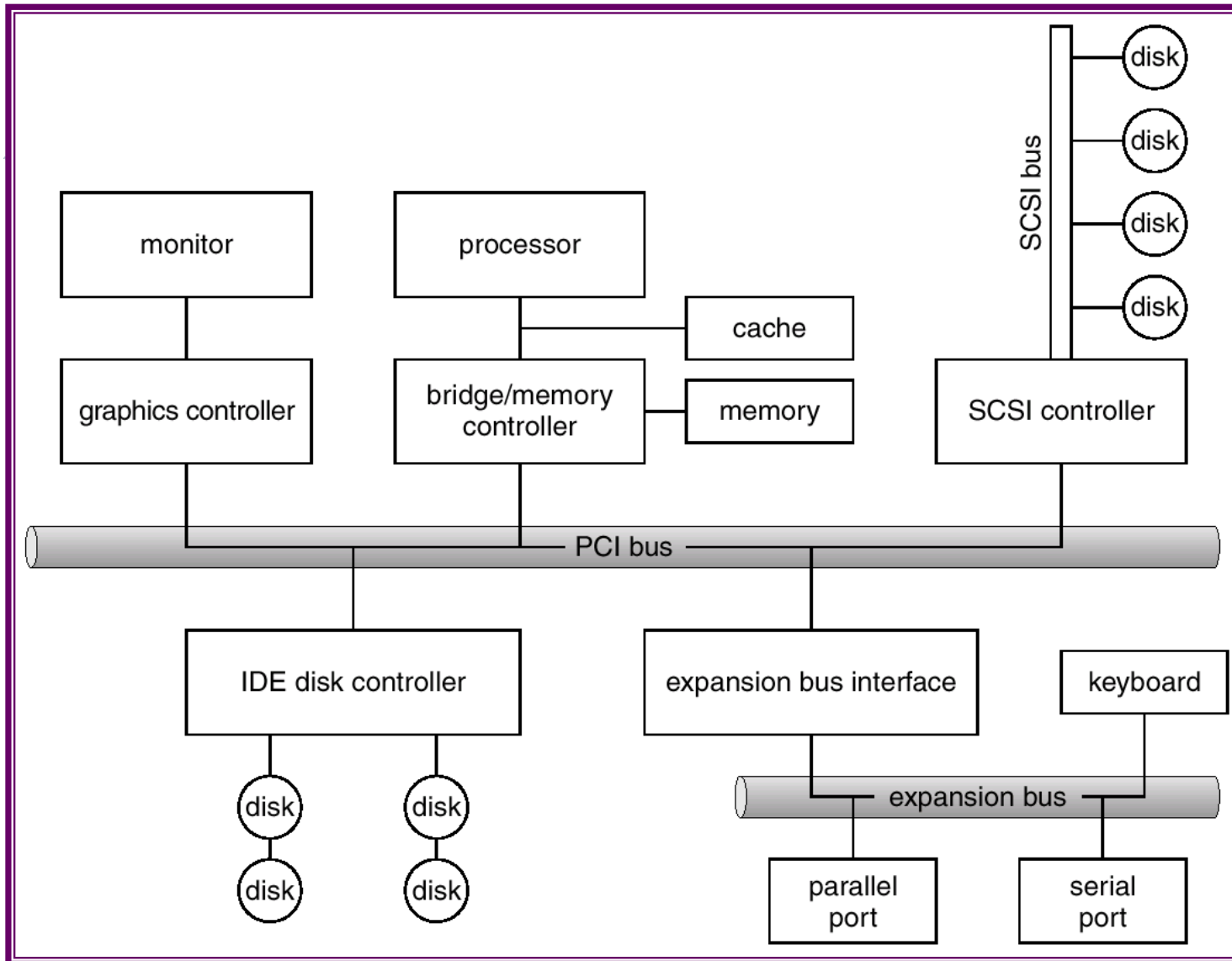
- ☞ **magistrala** (jedan uređaj priključen na sistem-daisy chain ili direktno deljenje podataka)

## ■ **I/O instrukcije** za kontrolu uređaja

## ■ **I/O uređaji** poseduju **adrese**, koje se koriste za:

- ☞ **direktne I/O instrukcije**
- ☞ **memorijski mapirane I/O instrukcije**

# Tipična Struktura Magistrale ko PC-a



# Lokacije Portova I/O Uređaja Kod PC-a (partial)

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)

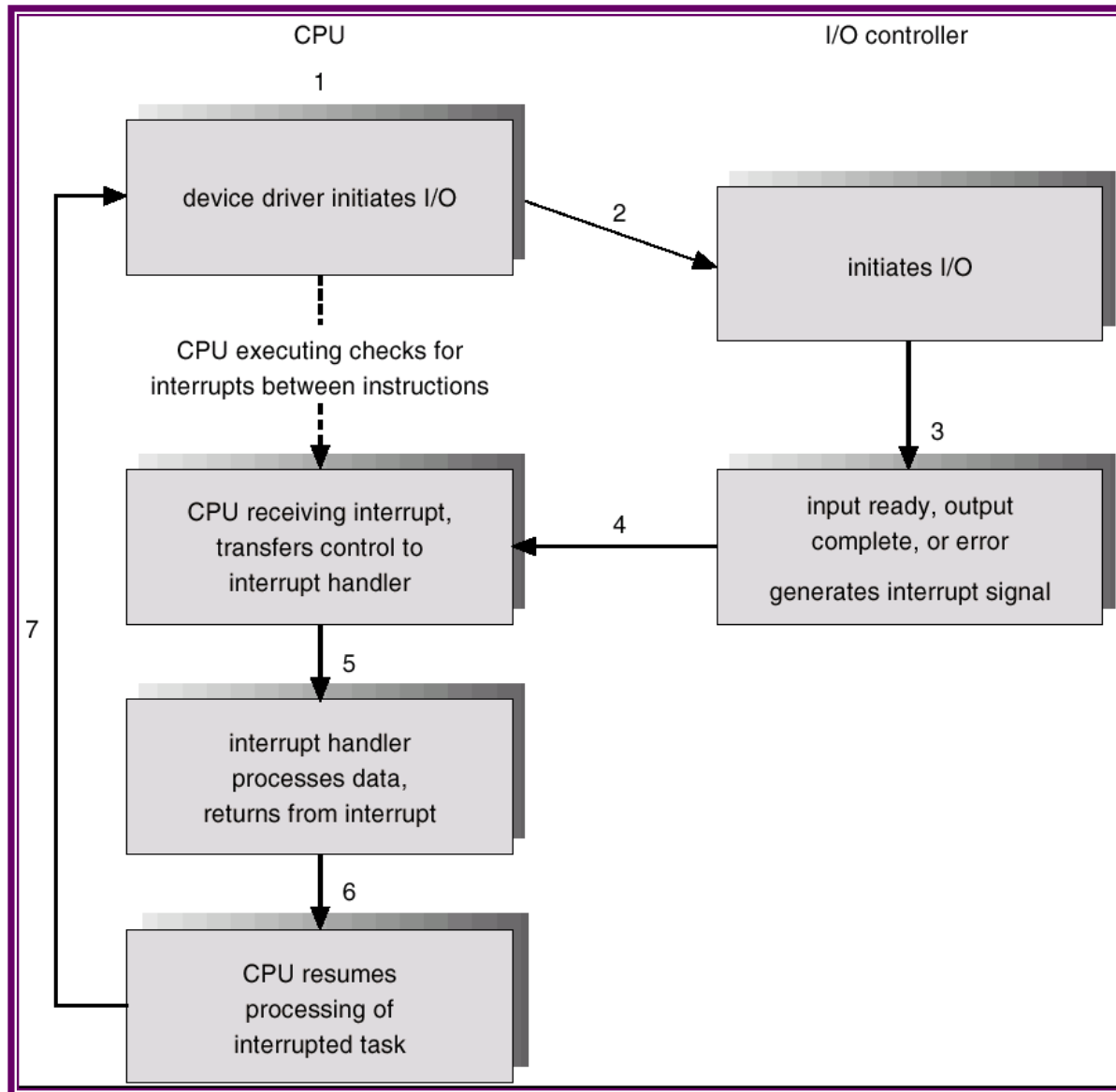
# Tehnika Prozivanja (polling) PIO

- Određivanje stanja korišćenjem statusnog registra, bits:
  - ☞ zauzeto(busy)
  - ☞ greska(error)
- Čekanje dok kontroler ne bude spreman-slobodan (free)
  - ☞ busy bit
- Zadavanje komandi:
  - ☞ postavljanje komandnog koda u komandni registar
  - ☞ postavljanje podatka u izlazni registar (ako je izlazna komanda-write)
  - ☞ postavljanje command-ready bita
- Izvršenje komande (sa mogućim transferom podataka)
  - ☞ busy bit je postavljen svo vreme
- Čekanje na kraj komande
  - ☞ busy bit
- Prozivanje = Busy-čekanje u petlji:
  - ☞ Da kontroler bude spreman
  - ☞ Izvršenje komande

# Prekidi (Interrupts)

- **CPU** poseduje **prekidnu liniju**
  - ☞ koja se aktivira
  - ☞ od strane I/O uređaja
- **ISR: Rutina za obradu prihvata prekidne signale**
- **Maskirajući IRQ se mogu**
  - ☞ ignorisati
  - ☞ odložiti (defer)
- **IVT: Vektor tabela prekida**
  - ☞ Poziva odgovarajuće rutine za obradu **prioritetnih prekida**
    - 📄 zasnovana na prioritetu
    - 📄 izvršava neke nemaskirane prekide
- **Prekidni mehanizam takođe se koristi za:**
  - ☞ izuzetke (exception)
  - ☞ Software interrupts

# Prekid-Odvijanje I/O Ciklusa



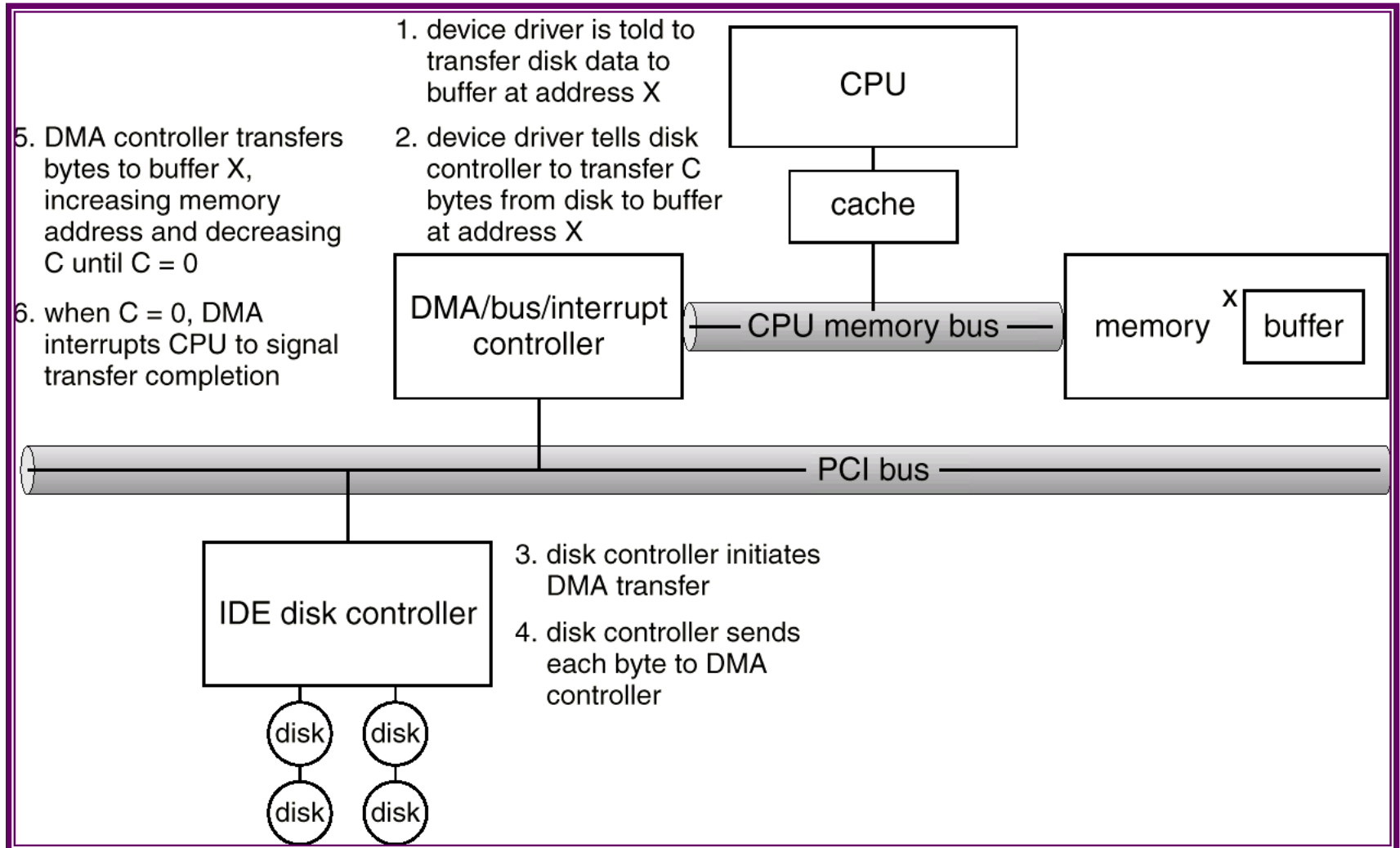
# Vektor Tabela Dogadaja Kod Intel Pentium Procesora

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

# Direktan Pristup Memoriji (DMA)

- **PIO=Programirani I/O**
- **Za svaki transfer podataka, PIO zahteva dve faze:**
  - ☞ čekanje u petlji da podaci budu spremni
  - ☞ transfer podataka
- **DMA:Koristi se da bi se izbegao**
  - ☞ programirani I/O
  - ☞ za veće transfere podataka
- **Zahteva DMA controller**
- **Oslobađa CPU**
  - ☞ od prenosa podataka
  - ☞ DMA prenosi direktno podatke
  - ☞ između I/O uređaja i memorije

# Šest Koraka Izvođenja DMA Transfera



# Aplikacijski I/O Interfejs

## ■ I/O sistemski pozivi su:

- ☞ visokog nivoa, generalni



## ■ Device-driver layer

- ☞ Karakteristike uređaja su sakrivene

- ☞ u specijalnim **modulima kernela** koji se nazivaju **drajveri**

## ■ Podela uređaja:

- ☞ **Po načinu transfera podataka** - na blok i karakter uređaje

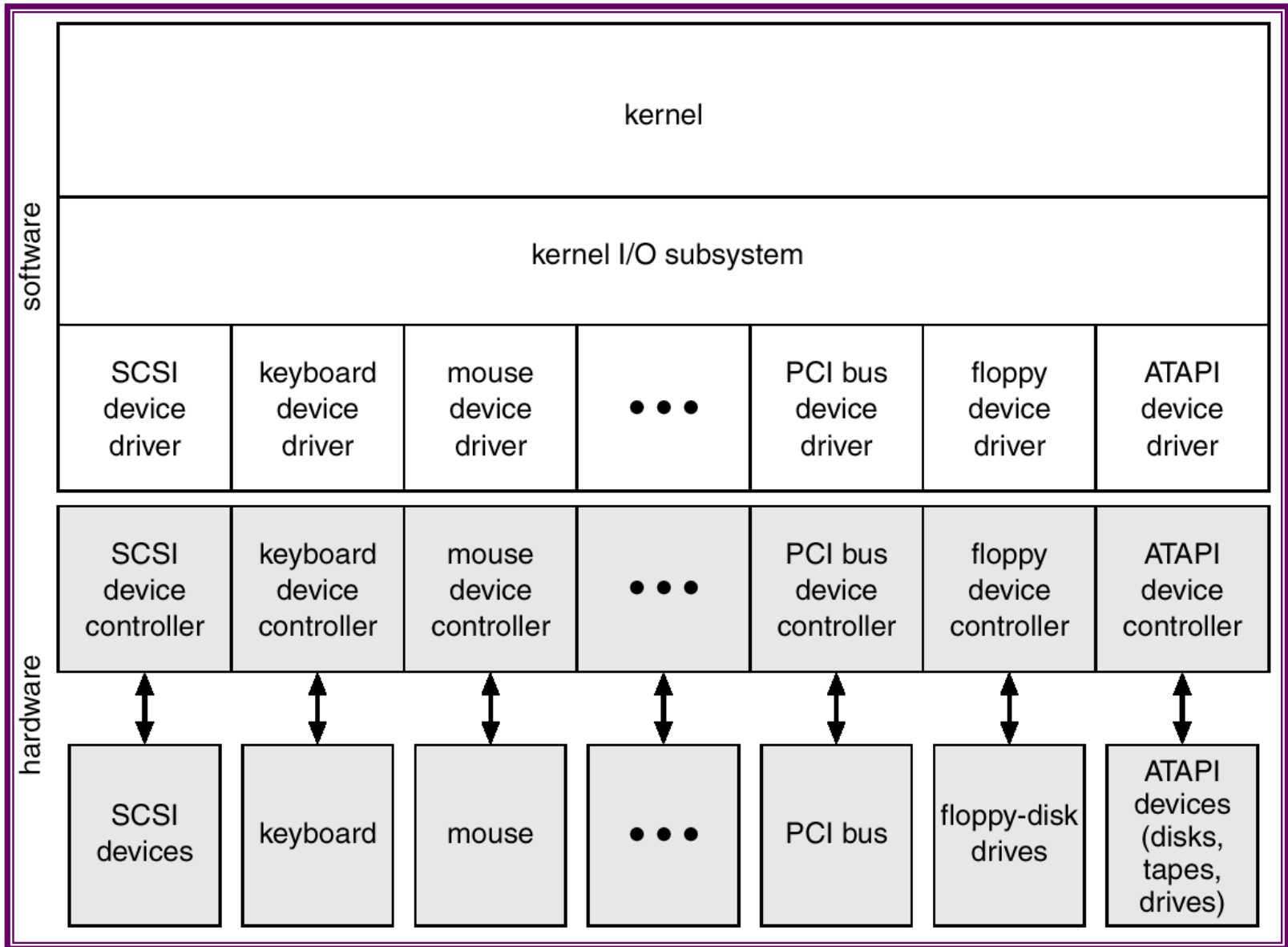
- ☞ **Po metodi pristupa** - sekvencijalni i uređaje sa direktnim ili slučajnim pristupom

- ☞ **Po deljivosti** - na deljive i nedeljive

- ☞ **Po brzini** - brzi i spori

- ☞ **Po mogućnosti upisa** - read-write, read only, or write only

# Struktura I/O Kernela



# Karakteristike I/O Uređaja

aspect	variation	example
data-transfer mode	character block	terminal disk
access method	sequential random	modem CD-ROM
transfer schedule	synchronous asynchronous	tape keyboard
sharing	dedicated sharable	tape keyboard
device speed	latency seek time transfer rate delay between operations	
I/O direction	read only write only read&write	CD-ROM graphics controller disk

# Blok i Karakter Uređaji

## ■ Blok uređaji se odnose na diskove

- ☞ rade sa blokovima podataka
- ☞ koriste komande: read, write, seek
- ☞ koriste sirov(raw) I/O pristup sistemu ili pristup preko systemske cache memorije
- ☞ memorija-moguć je mapirani pristup datotekama

## ■ Karakter uređaji koriste tastaturu, miš, serijske portove

- ☞ rade u režimu bajt po bajt
- ☞ koriste komande: get, put
- ☞ biblioteke
  - 📄 smeštene su na vrhu
  - 📄 dozvoljavaju editovanje po liniji

# Mrežni Uređaji

## ■ Razlikuju se od

- ☞ blok i karakter uređaja
- ☞ zato što poseduju **sopstveni interfejs socket**

## ■ Unix i Windows NT/9//2000 sadrže socket interfejs

- ☞ razdvajaju mrežni protokol
- ☞ od mrežnih operacija
- ☞ sadrže `select` funkcionalnost

## ■ Mehanizmi za mrežne interfejse

- ☞ pipes
- ☞ FIFO
- ☞ streams
- ☞ redovi za poruke (message queue)
- ☞ mailboxes

# Časovnik i Tajmer

## ■ Obezbeđuju:

- ➔ trenutno vreme
- ➔ proteklo vreme
- ➔ tajmerski okidač (trigger function)

## ■ Programibilni intervalski tajmeri se koriste za:

- ➔ merenje proteklog vremena
- ➔ periodične prekide

## ■ `ioctl` (na UNIX-u) pokriva

- ➔ sporedne aspekte I/O
- ➔ kao što su clock i tajmeri

# Blokirajuće i Ne-blokirajuće I/O Operacije

## ■ **Blokirajući:**

### ■ **proces se blokira i čeka sve dok se ne kompletira I/O operacija**

- ☞ **Lako za korišćenje i razumevanje**
- ☞ **Nedovoljan za neke potrebe**

## ■ **Neblokirajući:**

### ■ **I/O poziv vraća onoliko bajtova koliko je dostupno**

- ☞ **korisnički interfejs, kopiranje podataka (buffered I/O)**
- ☞ **ostvareno preko višenitnih procesa (multi-threading)**
- ☞ **brzo odgovara sa brojem pročitanih ili upisanih bajtova**

## ■ **Asinhroni:**

### ■ **I/O pozivi vraćaju kontrolu procesu u istom trenutku**

- ☞ **Proces se odvija dok se I/O izvršava**
- ☞ **teško za korišćenje**
- ☞ **I/O podsistem označava procesu kada je I/O završen**

# Kernelski I/O Podsystem

## ■ 1. Raspoređivanje (Scheduling)

- ☞ poredak I/O zahteva prispelih u per-device reda čekanja
- ☞ operativni sistemi optimizuju red izvršenja

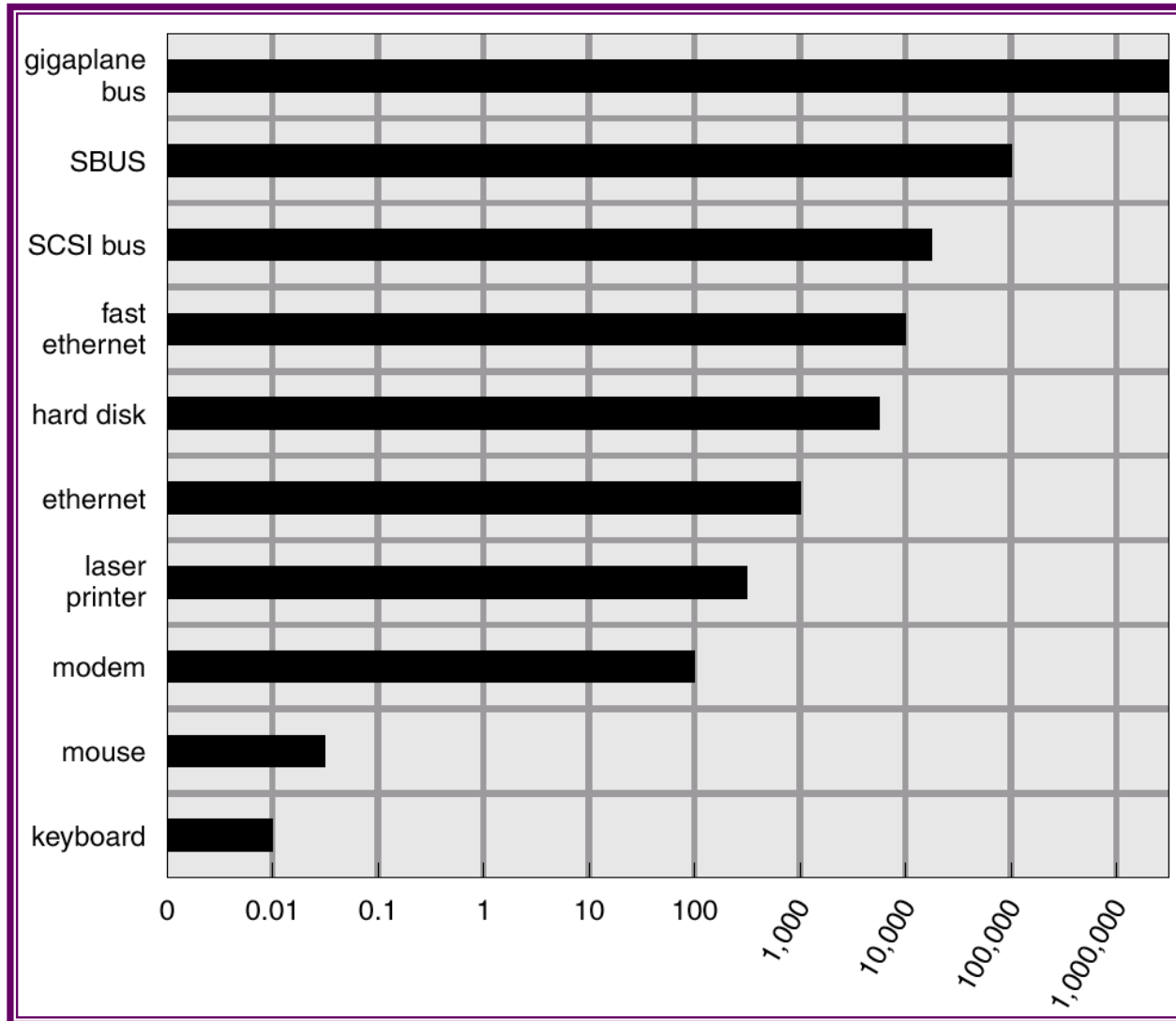
## ■ 2. Baferisanje (Buffering) –

- ☞ smešta privremeno podatke u memoriju
- ☞ dok se obavlja
- ☞ njihov transfer između uređaja

## ■ Baferisanje se koristi da bi se :

- ☞ prevazišle **razlike u brzini uređaja**
- ☞ prevazišle **razlike u veličini transfera** između uređaja
- ☞ održala semantika kopiranja **“copy semantics”**

# Sun Enterprise 6000 Device-Transfer Rates



KB/s

# Kernelski I/O Podsystem

- **3. Keširanje:**
- **brza memorija koja čuva kopije podataka**
  - ☞ Uvek samo kopije
  - ☞ Ključno je za performanse
  
- **4. Spool tehnika:**
- **bafer koji čuva izlazne podatke za uređaj**
  - ☞ ako uređaj može da pošalje samo jedan zahtev u tom trenutku
  - ☞ npr. Štampač
  
- **5. Rezervacija uređaja –**
- **omogućava ekskluzivni pristup uređaju**
  - ☞ **Sistemske pozive za dodeljivanje i razdeljivanje**
  - ☞ **Motrenje na zastoje**

# Upravljanje greškama

- OS može da se oporavi **od greške**:
  - ➔ čitanja diska
  - ➔ nedostupnosti uređaja
  - ➔ privremenog neuspelog upisa
- **Uglavnom vraća**:
  - ➔ broj greške
  - ➔ ili
  - ➔ kod
- **kada I/O zahtev neuspe**
- **System error logs**
  - ➔ čuvaju izveštaje o problemima (problem reports)

# Kernelske Strukture Podataka

- **Kernel mora da čuva informacije o stanju I/O komponenti,**

- ☞ kao što su:

- ☞ **tabela otvorenih datoteka**

- ☞ **mrežne konekcije**

- ☞ **stanja kareakter uređaja**

- **Većina kompleksnih struktura podataka čuvaju informacije za:**

- ☞ **baferisanje**

- ☞ **dodeljivanje memorije**

- ☞ **“prljavi” blokovi**

- **neki OS koriste**

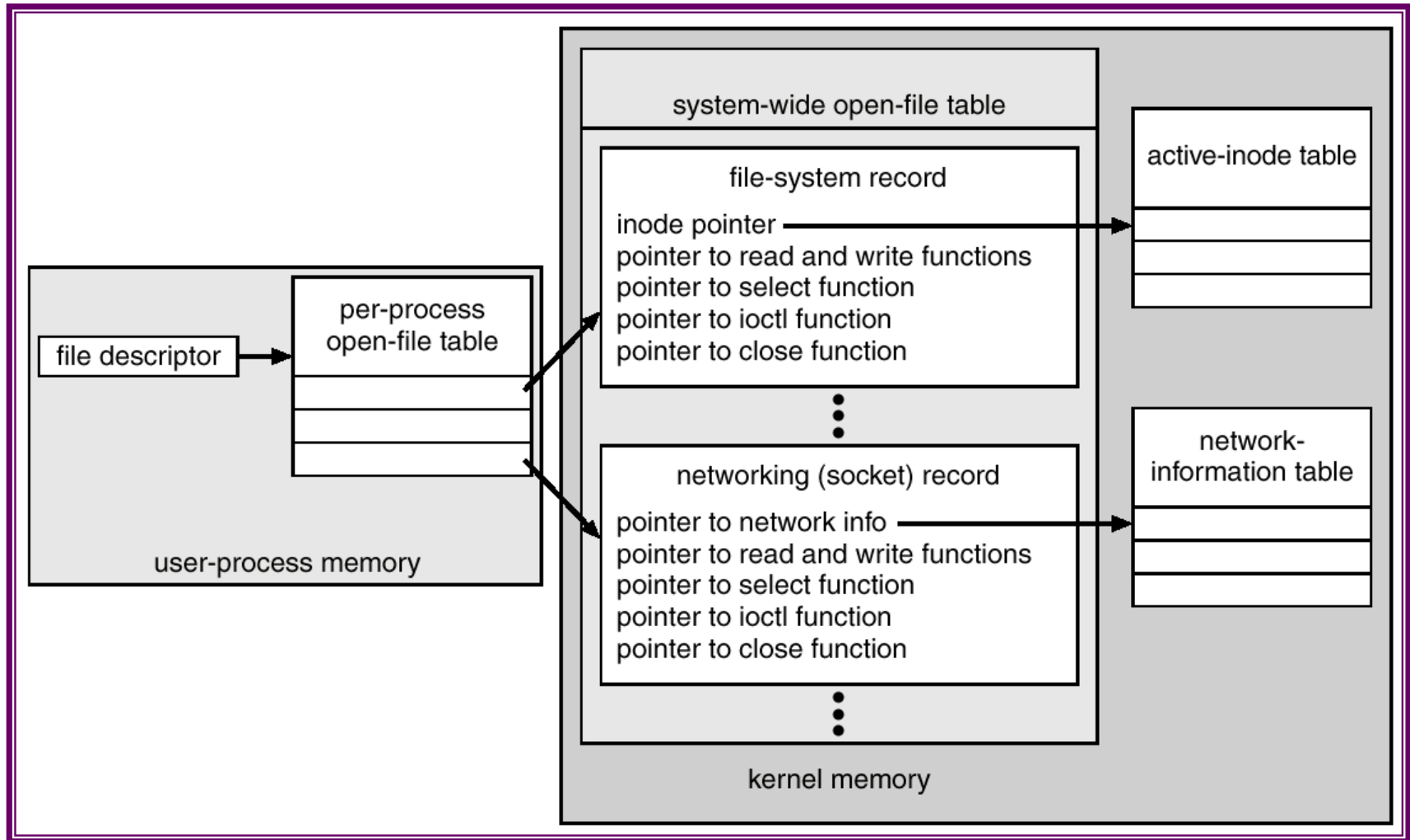
- ☞ **objektno orijentisane metode**

- ☞ **i**

- ☞ **Message-passing**

- ☞ **za implementaciju I/O**

# UNIX I/O Struktura Kernela

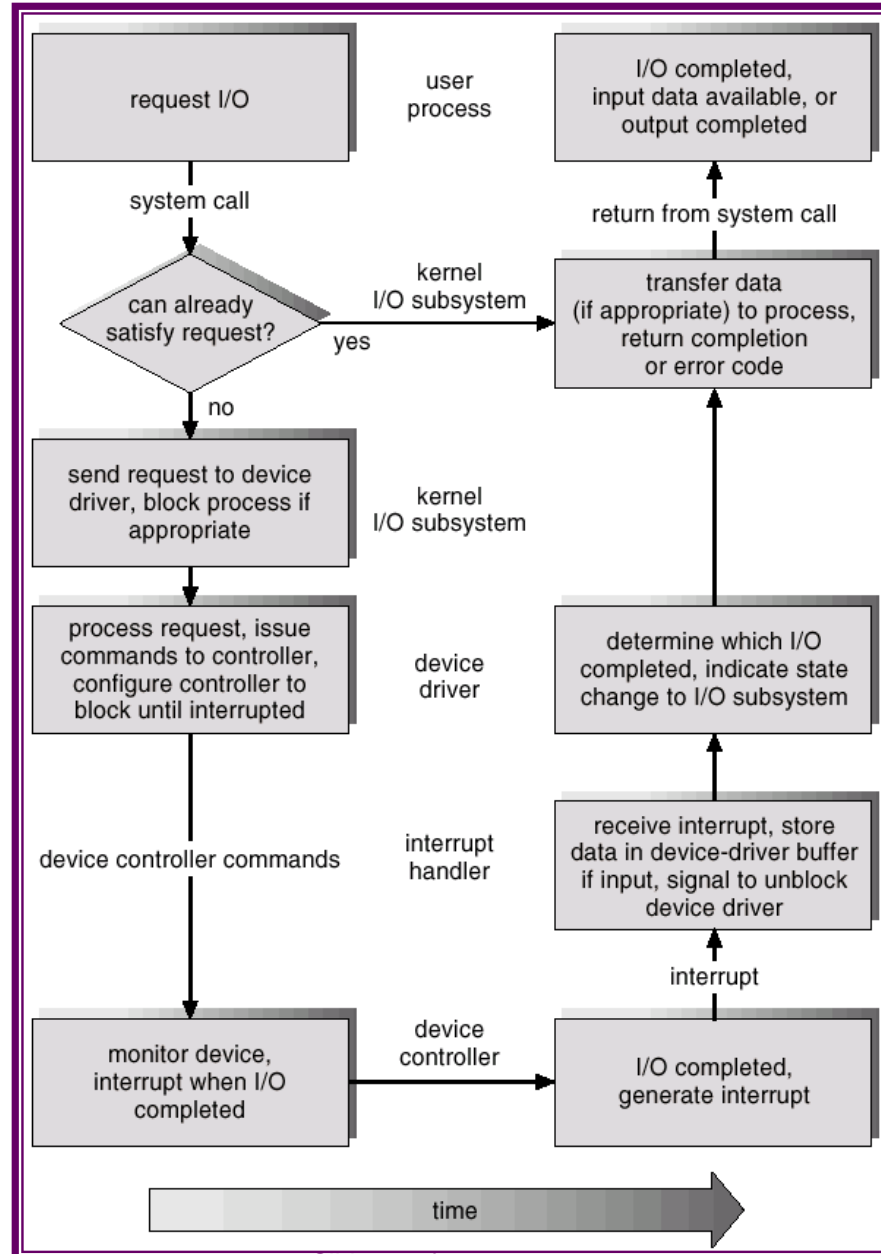


# Transformisanje I/O Zahteva u Hardverske Operacije

## ■ Abalizirajmo **učitavanje datoteke sa diska**:

- ☞ **Određivanje uređaja koji čuva datoteku**
- ☞ **Prevođenje imena koje se prikazuje uređaju**
- ☞ **Fizičko učitavanje podataka sa diska u bafer**
- ☞ **Priprema podataka dostupnih za proces**
- ☞ **Vraćanje kontrole procesu**

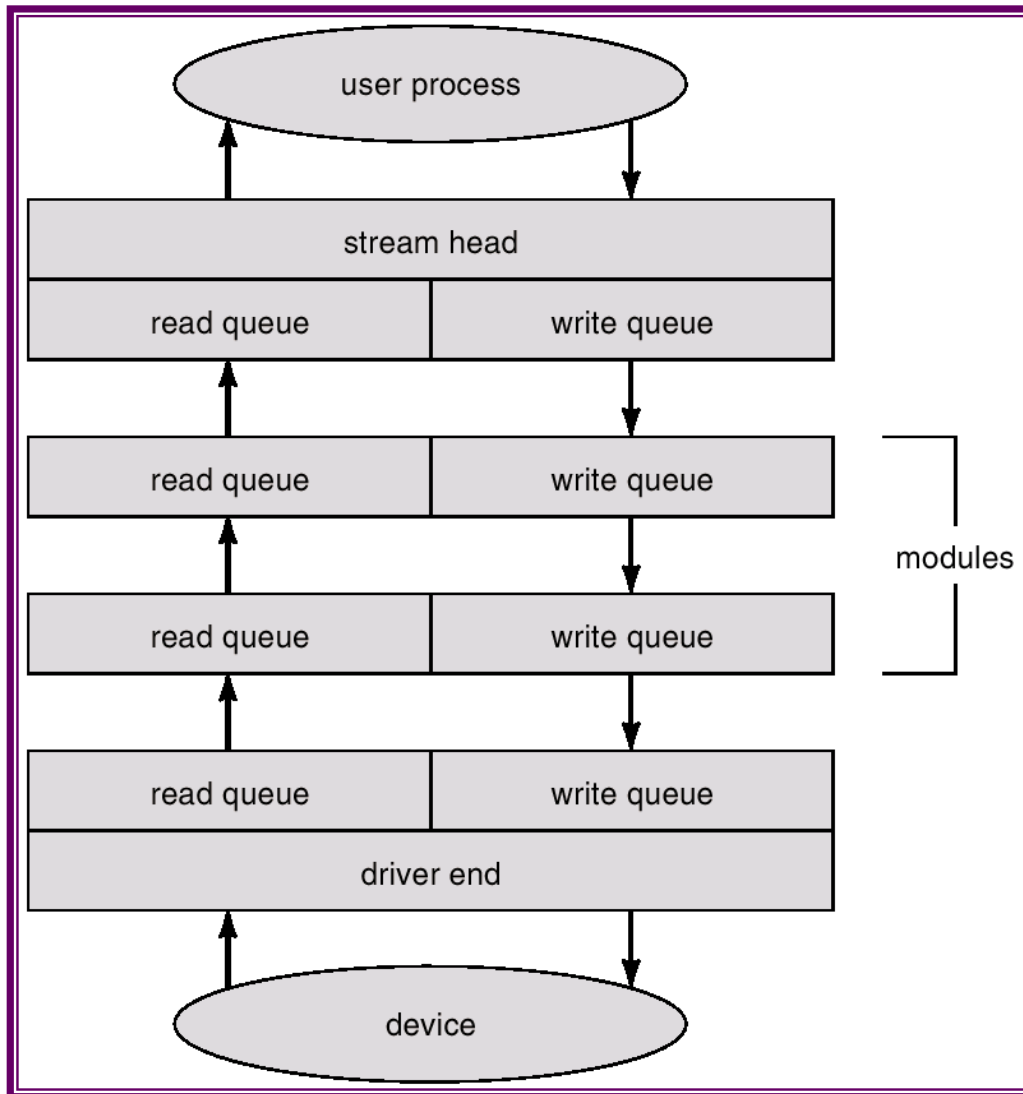
# Životni Ciklus I/O Zahteva



# STREAMS

- **stream:**
- **full-duplex komunikacioni kanal**
- **Između:**
  - ☞ **procesa** na korisničkom nivou
  - ☞ **i**
  - ☞ **uređaja**
- **(stream) se sastoji od:**
  - ☞ **glave (stream head) koja komunicira sa korisničkim procesom**
  - ☞ **drajvera i interfejsa sa uređaja**
  - ☞ **proizvoljnog broja STREAM modula između njih**
- **Svaki modul sadrži**
  - ☞ **red za čitanje**
  - ☞ **red za pisanje**
- **Prosleđivanje poruka**
  - ☞ **se koristi**
  - ☞ **za komunikaciju**
  - ☞ **između redova**

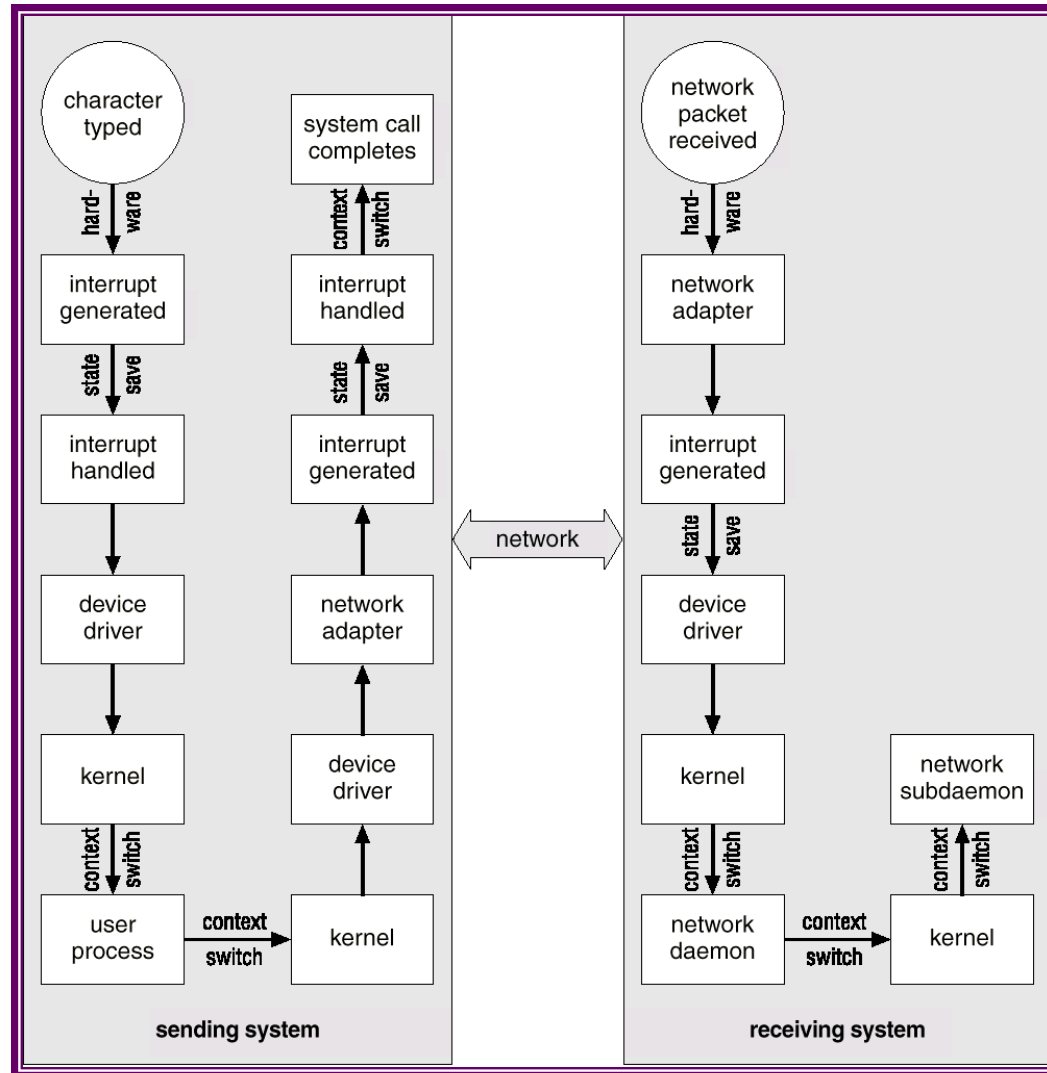
# Struktura STREAM-ova



# Performanse

- **I/O performanse jako utiču na opšte performanse sistema:**
  - ☞ **Zahtev od CPU da izvrši drajverski i kernelski I/O kod**
  - ☞ **prebacivanja konteksta i rukovanja prekidima**
  - ☞ **kopiranje podataka**
  - ☞ **Mrežni saobraćaj takođe izaziva veliki broj prekida**

# Komunikacija Unutar Računara



# Poboljšanje Performansi

- **Smanjenje broja prebacivanja sadržaja**
- **Smanjenje kopiranja podataka**
- **Smanjenje prekida**
  - ☞ korišćenjem
  - ☞ velikih transfera
  - ☞ pametnih kontrolera,
  - ☞ tehnikom pozivanja polling
- **Korišćenjem direktnog pristupa memoriji (DMA)**
- **Balansirano korišćenje:**
  - ☞ CPU, memorije, magistrale, I/O performansi
  - ☞ za najveću propusnu moć

# Način Postizanja Najbolje Funkcionalnosti Uređaja

