OPERATIVNI SISTEMI

VIII - Ulazno-izlazni podsistem

VIII - Ulazno-izlazni podsistem

S A D R Ž A J

Uvod

- posmatrajmo teorijski model OS-a
- **do sada su definisani**
  - pojam i funkcije jezgra OS-a
  - funkcije sloja upravljanja memorijom
- **time smo obezbedili**
  - okruženje u kojem se mogu formirati procesi koji međusobno mogu komunicirati
  - da procesima može biti dodeljena određena količina memorija na korišćenje
- **još uvek nije obezbeđeno sledeće**
  - korisnik procesima ne može zadati ulazne podatke
  - korisnik od procesa ne može preuzeti rezultate obrade
  - proces nije u mogućnosti da rezultate obrade sačuva na disku
  - proces ne može da ostvari komunikaciju sa drugim računarom
- **drugim rečima**
  - u OS treba ugraditi podsistem koji će obezbediti **komunikaciju sa U/I uređajima**

8.1 Funkcije ulazno-izlaznog podsistema

- većina uređaja koji se priključuju na računar može se svrstati u neku **opštu kategoriju**
- opšte kategorije su formirane prema **nameni uređaja**
  - uređaji za **dugotrajno skladištenje podataka** (HDD, SSD)
  - uređaji za **prenos podataka** (mrežne kartice)
  - uređaji koji obezbeđuju **interfejs prema korisniku** (monitori, tastature i miševi)
- osim namene, uređaje karakteriše još i sledeće
  - skup operacija koje se nad podacima mogu izvesti
  - brzina prenosa podataka na relaciji računar-uređaj
  - količina podataka kojima se istovremeno pristupa (jedinična količina transfera)
  - način pristupa podacima
  - deljivost uređaja itd...

8.1 Funkcije ulazno-izlaznog podsistema

- **funkcije U/I podsistema**
  - upravlja i kontroliše U/I uređaje i operacije koje ti uređaji izvršavaju
  - obezbeđuje što jednostavniji interfejs prema korisniku i prema ostatku sistema
- OS mora da obezbedi podršku za rad sa širokim spektrom ulazno-izlaznih uređaja
- karakteristike uređaja variraju sa
  - vrstom uređaja
  - konkretnim modelom određenog proizvođača hardvera
- raznovrsnost uređaja otežava konstrukciju U/I podsistema koji karakterišu
  - izvestan nivo **uniformnosti**
  - **izolacija karakteristika specifičnih** za konkretne uređaje

8.1 Funkcije ulazno-izlaznog podsistema

- konfliktni trendovi
  - standardizacija softverskih i hardverskih interfejsa
  - povećanje broja U/I uređaja različitih karakteristika
- rezultat: podrška za konkretne uređaje ili grupe srodnih sadržana je u **drajverima** *(drivers)*

8.1 Funkcije ulazno-izlaznog podsistema

Nezavisnost uređaja

- postoje dva aspekta nezavisnosti uređaja
  - **različiti modeli** uređaja istog tipa moraju se sa aspekta programa jednako posmatrati
    - primer: program pristupa DVD uređaju bez obzira na model
  - nezavisnost programa od **konkretne vrste uređaja**
    - primer: program na što istovetniji način može čita podatke sa SSD ili DVD uređaja
- posledica
  - programi ne rade sa realnim (konkretnim) uređajima
  - sve U/I operacije obavljaju pomoću virtuelnih uređaja
  - primer: **/dev/sda1**
  - prilikom čitanja vas ne interesuje da li je to
    - HDD, SSD, DVD, USB fleš memorija ...
    - koji je model uređaja

8.2 Klasifikacija U/I uređaja

- kriterijum: **namena uređaja**
  - uređaji za dugotrajno skladištenje podataka (*storage devices)* - diskovi, CD-i...
  - uređaji za prenos podataka (*transmission devices)* - mrežne kartice, modemi...
  - uređaji koji obezbeđuju interfejs ka korisniku (*human-interface devices)* - monitori, tastature, miševi...
- kriterijum: **smer transfera**
  - ulazni uređaji (miš, skener)
  - izlazni uređaji (štampač)
  - U/I uređaji (mrežna kartica)
- kriterijum: **jedinična količina transfera**
  - blok uređaji (HDD, jedinična količina je blok)
  - karakter (znak) uređaji (tastatura, jedinična količina je karakter)
  - mrežni uređaji

8.2 Klasifikacija U/I uređaja

- kriterijum: **metod pristupa**
  - uređaji sa direktnim pristupom (HDD, SDD)
  - uređaji sa sekvencijalnim pristupom (magnetna traka)
- kriterijum: **deljivost uređaja**
  - deljivi uređaji (HDD)
  - nedeljivi, odnosno posvećeni (tastatura)
- kriterijum: **mogućnost upisa**
  - uređaji za čitanje i upis (HDD)
  - uređaji samo za čitanje (CD-ROM)
  - uređaji samo za pisanje (grafička kartica)

8.2 Klasifikacija U/I uređaja

Časovnik i tajmer kao specijalna klasa uređaja

- **hardverski časovnik** i **tajmer** obezbeđuju tri osnovne funkcije
  - prikazivanje tekućeg vemena
  - prikazivanje proteklog vremena
  - tajmerski okidač za operaciju X u trenutku T
- predstavljaju specijalnu klasu uređaja
  - pristupa im se pomoću specifičnih sistemskih poziva
- tajmeri koji mere proteklo vreme ili obezbeđuju okidanje su **programibilni intervalski tajmeri**
  - oni se programiraju tako da nakon određenog vremena generišu **prekidni signal**
  - koriste se za realizaciju tehnike **pretpražnjena** (*preemption*) pri raspoređivanju procesora

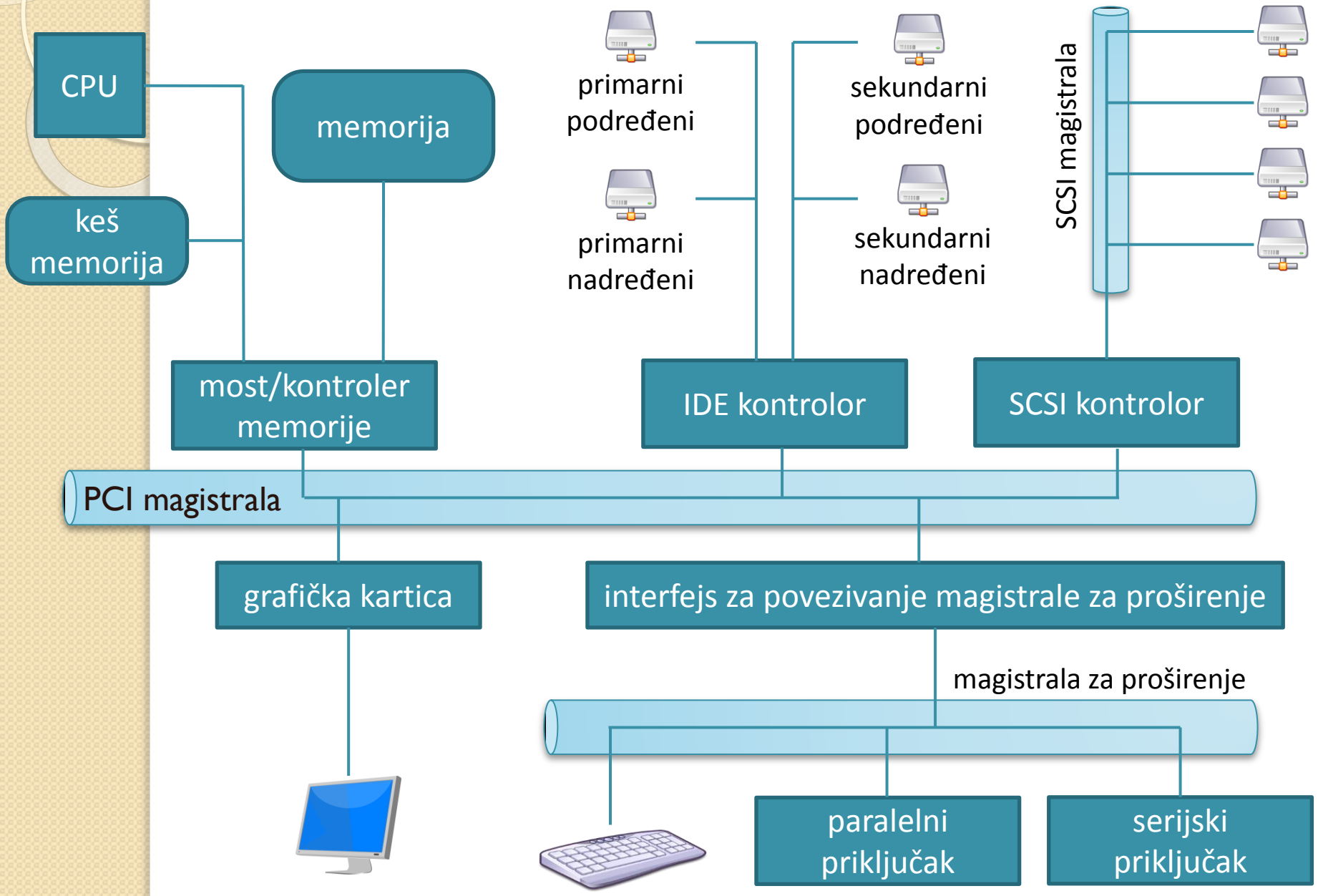
8.3 Hardver od značaja za U/I podsistem

- uređaji komuniciraju sa računarskim sistemom putem signala koje šalju preko žičnih ili bežičnih prenosnih medija
- princip povezivanja uređaja prikazan je na primeru PC arhitekture

Magistrala (*bus*)

- magistrala predstavlja vezu sa računarom kod koje više uređaja deli zajednički skup žica sa strogo definisanim protokolom koji specificira skup poruke koje se mogu poslati
  - **sistemska magistrala** PCI spaja procesorsko-memorijski podsistem sa brzim U/I uređajima (HDD, SSD...)
  - **magistrala za proširenje** (*expansion bus*) je specijalna magistrala koja sadrži serijske i paralelne priključke (*ports*) za spore uređaje (tastatura, miš...)

8.3 Hardver od značaja za U/I podsistem



8.3 Hardver od značaja za U/I podsistem

Kontroleri

- kontroler je kolekcija elektronike koja radi na tri načina
  - kao **port**
    - serijski kontroler je port
  - kao **magistrala**
    - SCSI kontroler se priključuje na PCI bus i formira novi SCSI bus
  - kao **uređaj**
    - grafička kartica je kontroler koji se priključuje na PCI magistralu, a radi kao uređaj
- svaki kontroler ima jedan ili više **registara**
- komunikacija između procesora i kontrolera se obavlja putem upisa i očitavanja vrednosti registara

8.3 Hardver od značaja za U/I podsistem

Kontroleri

- pristup registrima je moguć na dva načina
  - ako su memorijski i U/I prostor **razdvojeni**
    - procesor ima posebne instrukcije za upis i čitanje sa adrese na kontroleru
  - ako se U/I prostor **tretira kao memorija**
    - koriste se instrukcije za rad sa memorijom (memorijski mapirane U/I operacije)
- nekim uređajima se može pristupiti na oba načina
  - to zavisi od operacije koju treba obaviti
  - primer - **grafička kartica**
    - U/I portovima kojima se mogu obaviti osnovne upravljačke operacije pristupa se posebnim instrukcijama
    - memorija na grafičkim karticama koja služi za smeštaj slike tretira se kao memorija

8.3 Hardver od značaja za U/I podsistem

Kontroleri

- tipičan U/I port se sastoji iz četiri vrste registara
- **kontrolni registar**
  - služi za postavljanje režima rada uređaja (brzina porta i tip komunikacije)
  - u ovaj registar procesor isključivo upisuje podatke
- **statusni registar**
  - opisuje status (da li je komanda izvršena, da li podatak spreman ...)
  - procesor iz statusnog registra isključivo čita podatke
- **registar podataka za ulazni režim** (*data-in)*
  - *data-in* registar služi za čitanje podataka sa ulaznih uređaja
  - procesor iz *data-in* registra isključivo čita podatke
- **registar podataka za izlazni režim** (*data-out)*
  - *data-out* registar služi za upis podataka na izlazni uređaj
  - iz ovog registra procesor isključivo čita informacije

8.3 Hardver od značaja za U/I podsistem

Tehnika prozivanja (*polling*)

- kompletan protokol za interakciju između računara i kontrolera može biti dosta komplikovan
- računar i kontroler funkcionišu po principu **proizvođač-potrošač**
- ovaj princip se realizuje pomoću dva bita
- ***busy bit***
  - opisuje stanje kontrolera u **statusnom registru**
    - ukoliko je bit setovan, kontroler je zauzet, odnosno nešto radi
    - ukoliko je bit resetovan, kontroler je slobodan i spreman da prihvati komandu
- ***command-ready bit***
  - nalazi se u **komandnom registru** kontrolera
  - računar ukazuje kontroleru na **prisustvo nove komande**
    - kada računar setuje ovaj bit to znači da je kontroler dobio komandu i da izvršavanje može da počne

8.3 Hardver od značaja za U/I podsistem

Tehnika prozivanja (*polling*)

- tehnika sporazumevanja (*handshake)* prikazana je putem primer slanja jednog bajta na serijski kanal
  1. računar čita statusni registar dok vrednost *busy* bita ne bude 0
  2. računar setuje *write* bit u komandnom registru i upisuje bajt u *data-out* registar
  3. računar setuje *command-ready* bit
  4. kontoler setuje *busy* bit
  5. kontroler čita svoj komandni registar i detektuje *write* komandu
     - kontroler čita *data-out* registar
     - kontroler organizuje ciklus za slanje sadržaja *data-out* registra na serijski kanal
  6. ako je sve u redu kontroler briše
     - *command-ready* bit
     - *error bit* u statusnom registru koji ukazuje na pojavu greške
     - *busy* bit koji ukazuje da je komanda završena
- počev od kraja koraka 4 računar stalno proverava da li je busy bit 0, kako bi znao kada je komanda završena

8.3 Hardver od značaja za U/I podsistem

Tehnika prozivanja (*polling*)

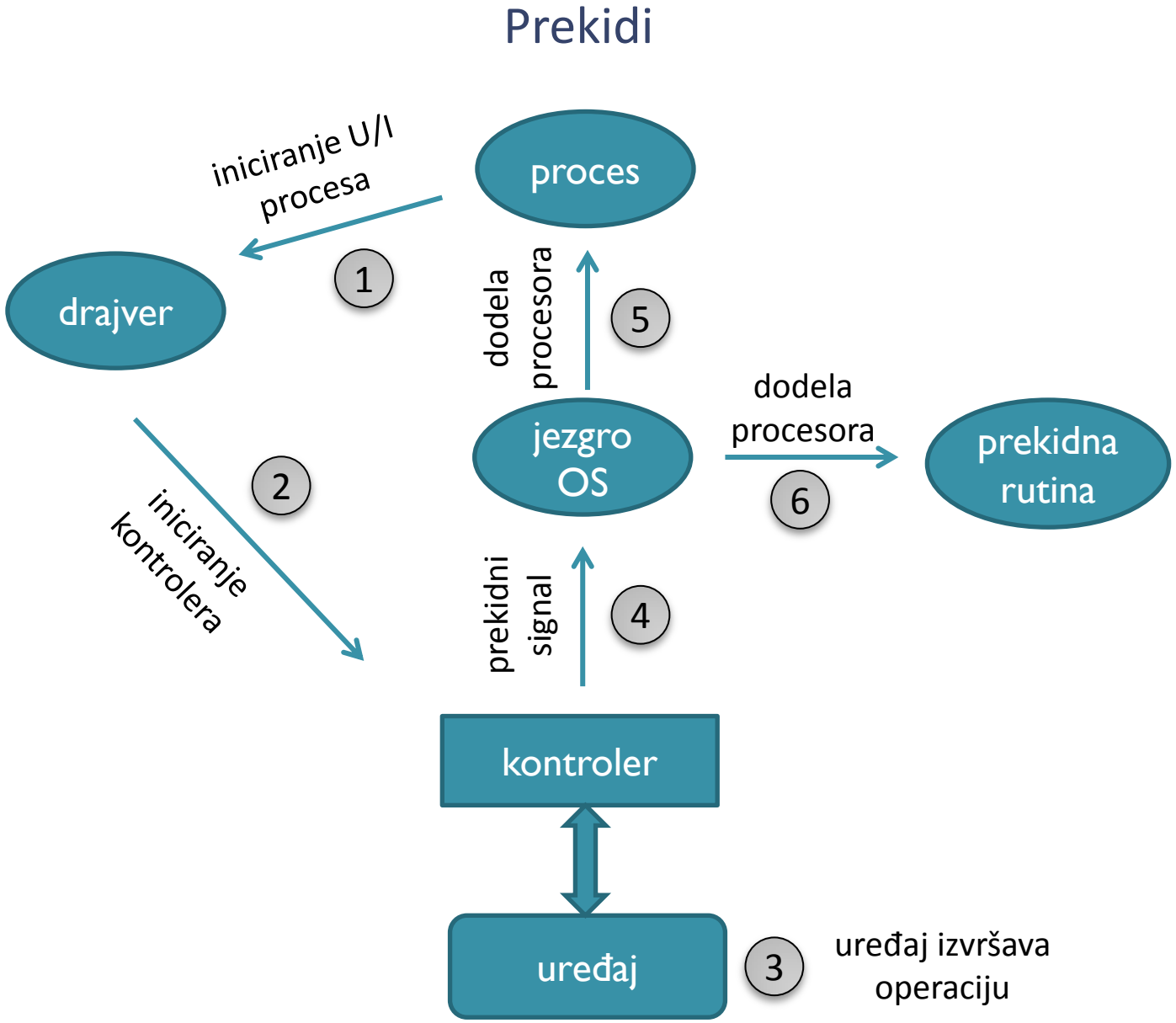
- ovakva petlja mora da se ponovi za svaki ciklus
- u koraku 1 imamo tehniku **prozivanja** (*polling)*
  - ova tehnika se takođe naziva **zaposlen čekanjem** (*busy-waiting)*
    - računar ponavlja čitanje statusnog registra sve dok vrednost busy bita ne postane 0
- tehnika prozivanja se na mnogim CPU obavlja u sledećoj **petlji**
  - čitanje statusnog registra
  - ekstrakcija i analiza *busy* bita
  - odluka na osnovu vrednosty *busy* bita
    - povratak u petlju ako je vrednost busy bita 1
    - izlaz iz petje ako je vrednost busy bita 0
- u slučaju sporih uređaja procesor ostaje zaposlen čekanjem u krajnje dugačkim petljama
- osnovni nedostatak: **trošenje procesorskog vremena na petlju prozivanja**
- nedostatak se može ukloniti uvođenjem **mehanizma prekida** (*interrupt)*

8.3 Hardver od značaja za U/I podsistem

Prekidi

- hardverski mehanizam
- omogućava uređaju da označi procesoru kada je **komanda izvršena**
- funkcioniše na sledeći način
  - kada uređaj završi operaciju, kontroler šalje prekidni signal procesoru preko **prekidne linije** (IRQ - *interrupt request line)*
  - procesor će završiti tekuću instrukciju
  - čuva se **kontekst procesa** (vrednosti programskog brojača, registara procesora ...)
  - prekida se sekvencijalno izvršavanje programa kako bi se obradio prekid
  - poziva se **rutina za obradu prekida** (*interrupt handler)* koja
    - određuje uzrok prekida
    - opslužuje prekid
  - posle obrade prekida, procesor nastavlja izvršavanje programa

8.3 Hardver od značaja za U/I podsistem



8.3 Hardver od značaja za U/I podsistem

Prekidi

1. proces najpre inicira ulazno-izlazni ciklus koji se prosleđuje drajveru
2. drajver inicira kontroler
3. uređaj zatim izvršava operaciju
4. nakon završetka kontroler postavlja prekidni signal
5. prekida se izvršenje tekućeg procesa i predaje kontrola rutini za obradu prekida
6. nakon završene prekidne rutine, obnavlja se kontekst prekinutog procesa i predaje kontrola prekinutom procesu

- ovakav mehanizam omogućava procesoru da odgovori na svaki asinhroni događjaj

8.3 Hardver od značaja za U/I podsistem

Prekidi

- u modernim OS-ima tehnika prekida uključuje
  - mogućnost **odlaganja** (*deffer)* obrade prekida dok je proces u kritičnoj sekciji
  - brzu i efikasnu tehniku određivanja uređaja koji je postavio prekid
  - višeslojni prekidni sistem koji će razlikovati prioritet prekidnih signala i rešavati prekide po prioritetu
- većina procesora umesto jedne ima dve posebne linije za prekidne signale
- **nemaskirajuća** (*non-masking)* linija
  - prekidni signal uvek može da prekine izvršenje tekućeg procesa
  - koristi se za slanje prekidnih signala usled kritičnih hardverskih grešaka (greške u memoriji)
- **maskirajuća** (*masking)* linija
  - prekidni signal ne prekida izvršenje procesa sve dok se proces npr. nalazi u kritičnoj sekciju
  - koristi se za slanje prekidnih signala koji potiču od normalnih operacija i standardnih grešaka koje se javljaju na U/I uređajima

8.3 Hardver od značaja za U/I podsistem

DMA (*Direct Access Memory*)

- kada uređaj pripremi podatke procesor treba da preuzme svaki bajt iz kontrolera
  - procesor svaki put proverava statusni bit što znači da se izvršavaju dva U/I ciklusa
  - ova tehnika čitanja podataka se naziva **programirani ulaz-izlaz** (PIO - *Programmed Input–Output*)
- u slučaju masovnih transfera podataka PIO tehnika izaziva veliki gubitak vremena
  - primer: čitanje sa HDD
- zato se koristi **DMA kontroler** (*direct memory access)*
  - obavlja transfer podataka na relaciji kontroler - operativna memorija i obrnuto
  - ima signale za upravljenje transferom sa kontrolera
  - ima signale kojima upravlja memorijskim ciklusima

8.3 Hardver od značaja za U/I podsistem

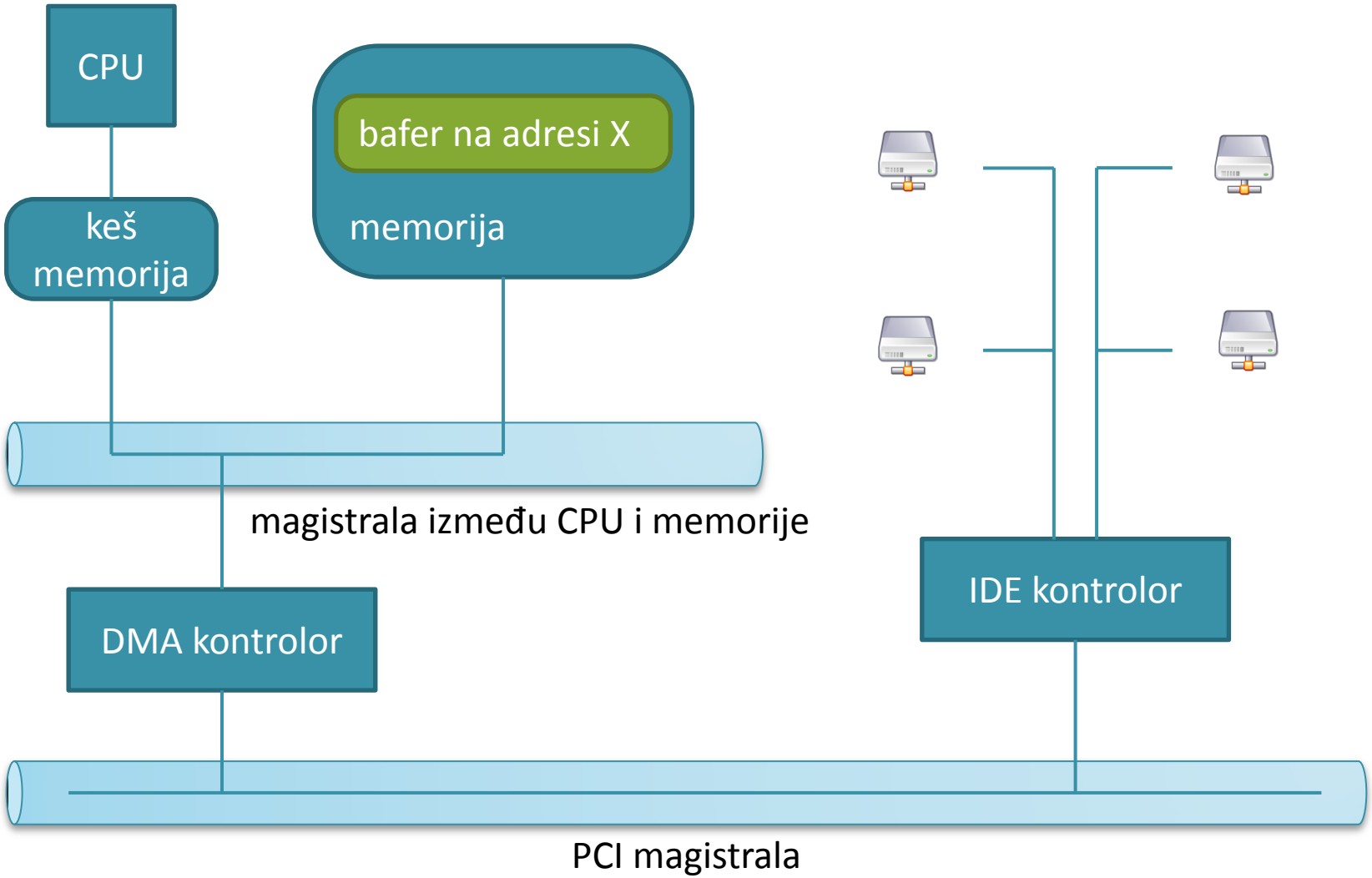
DMA (*Direct Access Memory*)

- uprošćeni primer prenosa podataka na relaciji disk-memorija preko DMA kontrolera

1. proces nalaže drajveru prenos podatke sa diska u bafer koji je na memorijskoj adresi X
2. drajver nalaže disk kontroleru da izvrši transfer C bajtova sa diska u bafer na adresi X
3. disk kontroler inicira DMA transfer
4. disk kontroler redom šalje bajtove podataka DMA kontroleru
5. DMA kontroler prenosi podatke u bafer
   - DMA uvećava memorijsku adresu X nakon svakog prenešenog bajta
   - DMA smanjuje broj bajtova C nakon svakog prenešenog bajta
6. kada je prenešen poslednji bajt (C=0) DMA šalje prekidni signal procesoru
   - prekidni signal je obaveštenje o završenom transferu

8.3 Hardver od značaja za U/I podsistem

DMA (*Direct Access Memory*)



8.3 Hardver od značaja za U/I podsistem

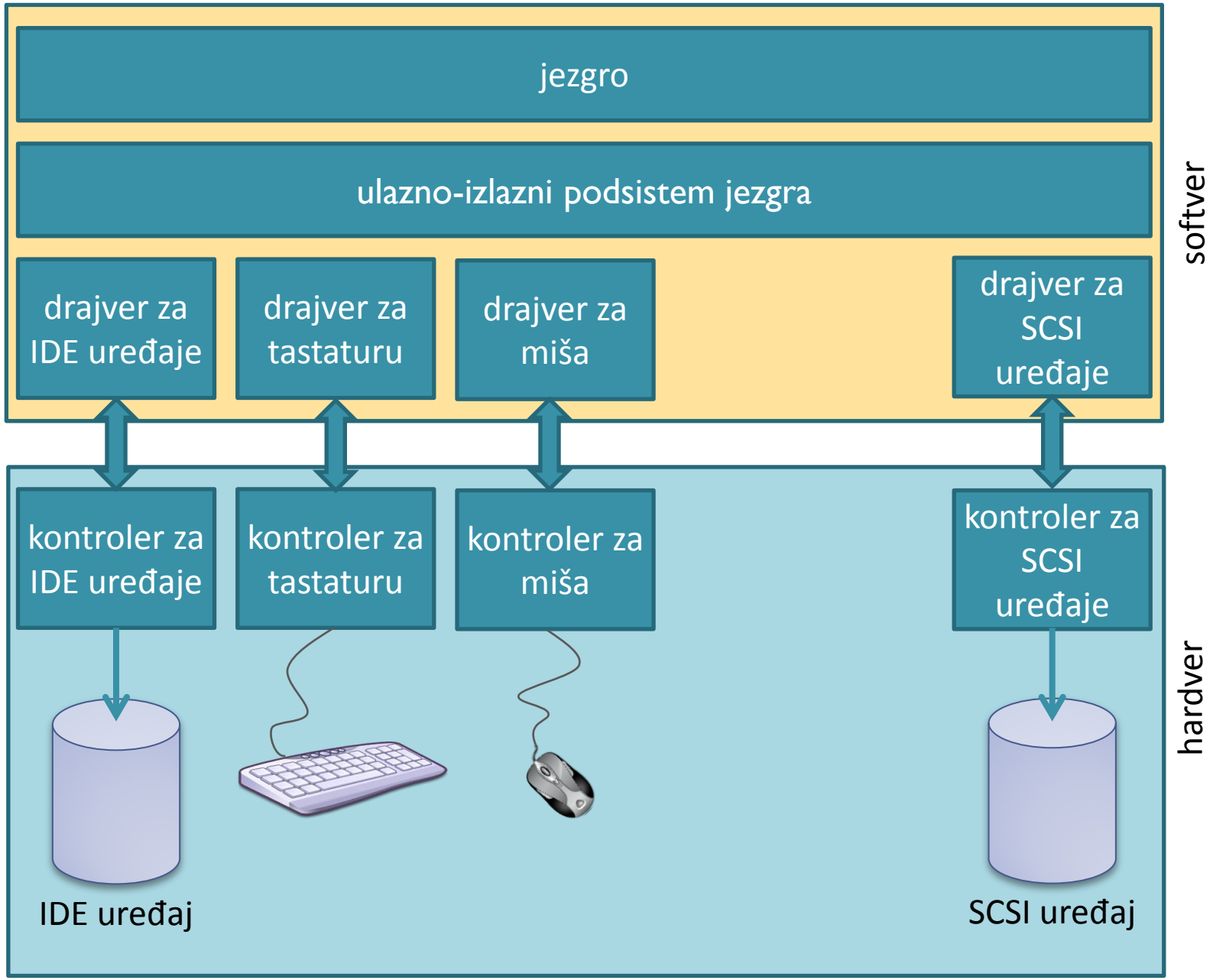
DMA (*Direct Access Memory*)

- iniciranje DMA transfera zahteva da se definiše
  - početna memorijska adresa
  - blok bajtova koji treba da se prenese
  - smer transfera
- U/I kontroler se ne navodi kao izvorište jer je DMA fizički povezan sa njim

8.4 Uniformni interfejs ka aplikacijama

- srodni uređaji (na primer, diskovi) grupišu se u **opšte klase uređaja**
- svakoj klasi uređaja pristupa se preko standardnog skupa funkcija (tzv **interfejs**)
- razlike između uređaja iste klase sakrivene su u specijalnim modulima jezgra OS
  - ti moduli se nazivaju **drajveri** (*drivers)*
  - drajveri postoje za svaki uređaj
- iznad svih drajvera nalazi se sloj U/I podsistema jezgra praktično nezavisan od hardvera
  - svi detalji vezani za konkretan hardver prepušteni su drajverima
- ova slojevita organizacija olakšava posao
  - dizajnerima OS-a
  - proizvođačima uređaja i kontrolera
    - proizvođači uređaja za svoje uređaje pišu drajvere za razne OS
    - plus: moguće je priključenje uređaja bez intervencije projektanata samog OS-a
    - minus: za svaki OS neophodan je poseban drajver za konkretan uređaj

8.4 Uniformni interfejs ka aplikacijama



8.4 Uniformni interfejs ka aplikacijama

- deo jezgra zadužen za rad sa U/I uređajima koordiniše široku kolekciju servisa koji su na raspolaganju aplikacijama i ostalim delovima kernela, kao što su
  - upravljanje imenima datoteka i uređaja (*name space),*
  - kontrola pristupa datotekama i uređajima
  - kontrola operacija
  - dodela uređaja procesima na korišćenje
  - raspoređivanje ulazno-izlaznih operacija
  - baferovanje, keširanje i spul tehnika
  - kontrola statusa uređaja
  - konfiguracija i inicijalizicija drajvera

8.4 Uniformni interfejs ka aplikacijama

Blokirajući i neblokirajući sistemski pozivi

- sistemski pozivi dozvoljavaju izbor između blokirajućih i neblokirajućih U/I operacija
- proces koji izdaje **blokirajući sistemski poziv** za izvršavanje operacije
  - blokira se i čeka da se operacija izvrši
  - oslobađa procesor i vraća se u red čekanja na procesor tek onda kada se operacija završi
- proces koji izdaje **neblokirajući sistemski poziv** za izvršavanje operacije
  - ne blokira, nego nastavlja da radi
  - primer
    - komandni interpreter koji kao proces koji obezbeđuje korisnički interfejs
    - podaci se primaju sa tastature i istovremeno prikazuju na ekranu
    - interpreter inicira zadatu komandu tek kada se svi podaci prime sa ulaznog uređaja

8.5 Usluge koje obezbeđuje U/I podsistem

- jezgro operativnog sistema obezbeđuje više usluga koje se odnose na ulazno-izlazne operacije
  - raspoređivanje ulazno-izlaznih operacija
  - baferovanje
  - keširanje
  - spuler (*spooler)*
  - prevođenje zahteva u U/I operacije

8.5 Usluge koje obezbeđuje U/I podsistem

Raspoređivanje ulazno-izlaznih operacija (*sheduling*)

- podrazumeva obezbeđivanje dobrog poretka izvršavanja U/I operacija u cilju postizanja optimalnih performansi
- poredak u kome su zahtevi za izvršavanjem U/I operacija pristigli u sistem
  - po pravilu je strogo stohastički (slučajni)
  - gotovo nikada nije optimalan
- raspoređivanje operacija se zasniva na **redu čekanja**
  - definiše se za svaki uređaj posebno
  - red se formira na na osnovu nekog **kriterijuma**
    - primer: kriterijum kojim se smanjuje vreme potrebno za pozicioniranje glava diska
  - prilikom formiranja kriterijuma mora se sprečiti pojava zastoja i zakucavanja

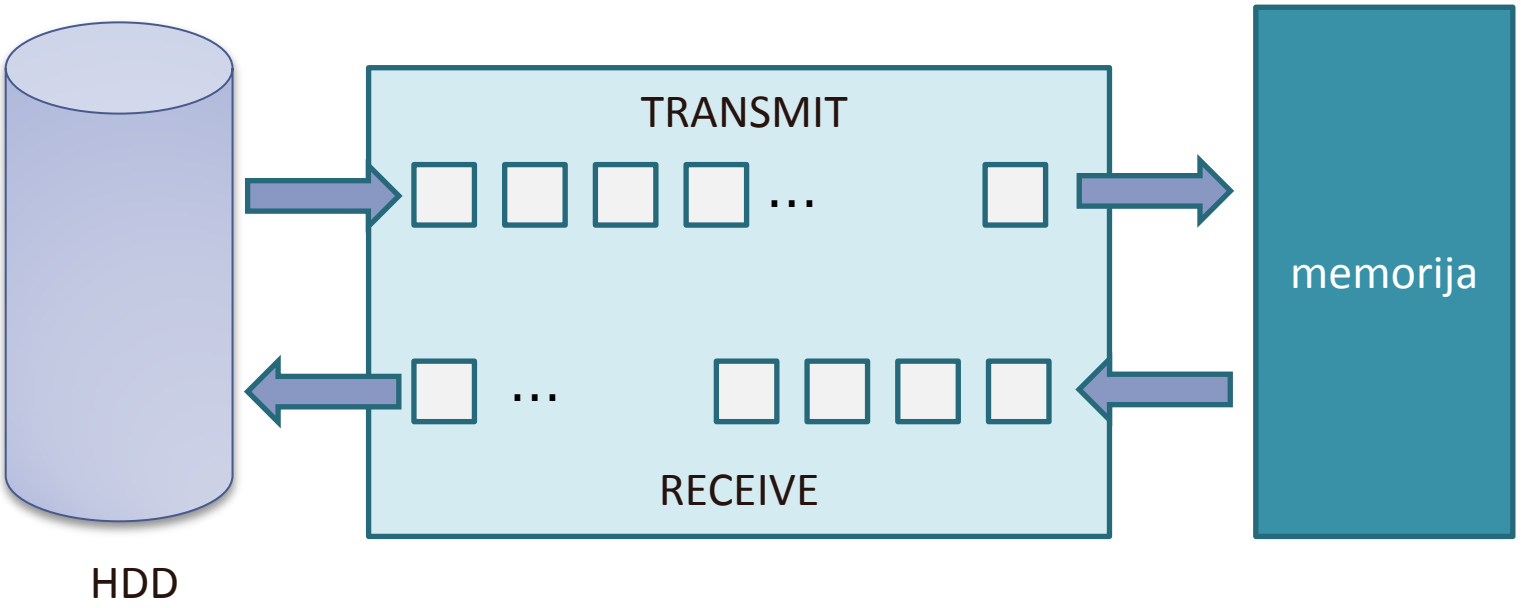
8.5 Usluge koje obezbeđuje U/I podsistem

Baferovanje (*buffering*)

- bafer je deo memorije koji funkcioniše na principu proizvođač-potrošač
- bafer čuva privremene podatake prilikom transfera između dva uređaja ili uređaja i aplikacije
- razlog baferovanja je **usklađivanje različitih** brzina između potrošača i proizvođača
- primer
  - **dvostruko baferovanje** (*double-buffering)*
  - tehnika koja se koristi pri keširanju hard diskova
  - postoje dva bafera
    - TRANSMIT, koristi se za baferovanje podataka na relaciji memorija-disk
    - RECEIVE, koristi se za baferovanje podataka na relaciji disk-memorija
  - time se premošćava razlika u brzini između memorije i diska

8.5 Usluge koje obezbeđuje U/I podsistem

Baferovanje (*buffering*)



Primena dvostrukog baferovanja prilikom keširanja hard diskova

8.5 Usluge koje obezbeđuje U/I podsistem

Keširanje (*caching*)

- **keš** (*cache)* je region brze sistemske memorije koji čuva kopiju podataka, obično sa diska
- keširanje (*caching)* je tehnika kopiranja delova diska u keš memoriju
  - pristup podacima u kešu je znatno brži u odnosu na pristup podacima na U/I uređajima
  - time se osetno povećavaju performanse disk U/I sistema
- razlika između keša i bafera
  - bafer čuva **trenutno aktuelne podatke**
  - keš čuva **bilo koju kopiju sa diska**
- isti memorisjki prostor se može koristiti i za baferovanje i za keširanje
- keširanje je jako značajno i realizuje se u više nivoa
  - pri tome se koriste različite tehnike za popunu i zamenjivanje podataka u kešu

8.5 Usluge koje obezbeđuje U/I podsistem

Spuler (*spooler*)

- **spuler** je bafer koji privremeno čuva izlazne podatke namenjene nedeljivom uređaju
- spul tehnika omogućava istovremeni pristup uređajima koji nisu deljivi na sledeći način
  - procesi upisuju podatke namenjene uređaju na disk
  - OS upravlja tim spul baferom tako što opslužuje jedan po jedan zahtev
- primer
  - svaki proces koji želi nešto da štampa ostavlja svoj zathev u spul bafer na disku
  - **print spuler** (proces koji radi u pozadini) obavlja štampu jednog po jednog zahteva

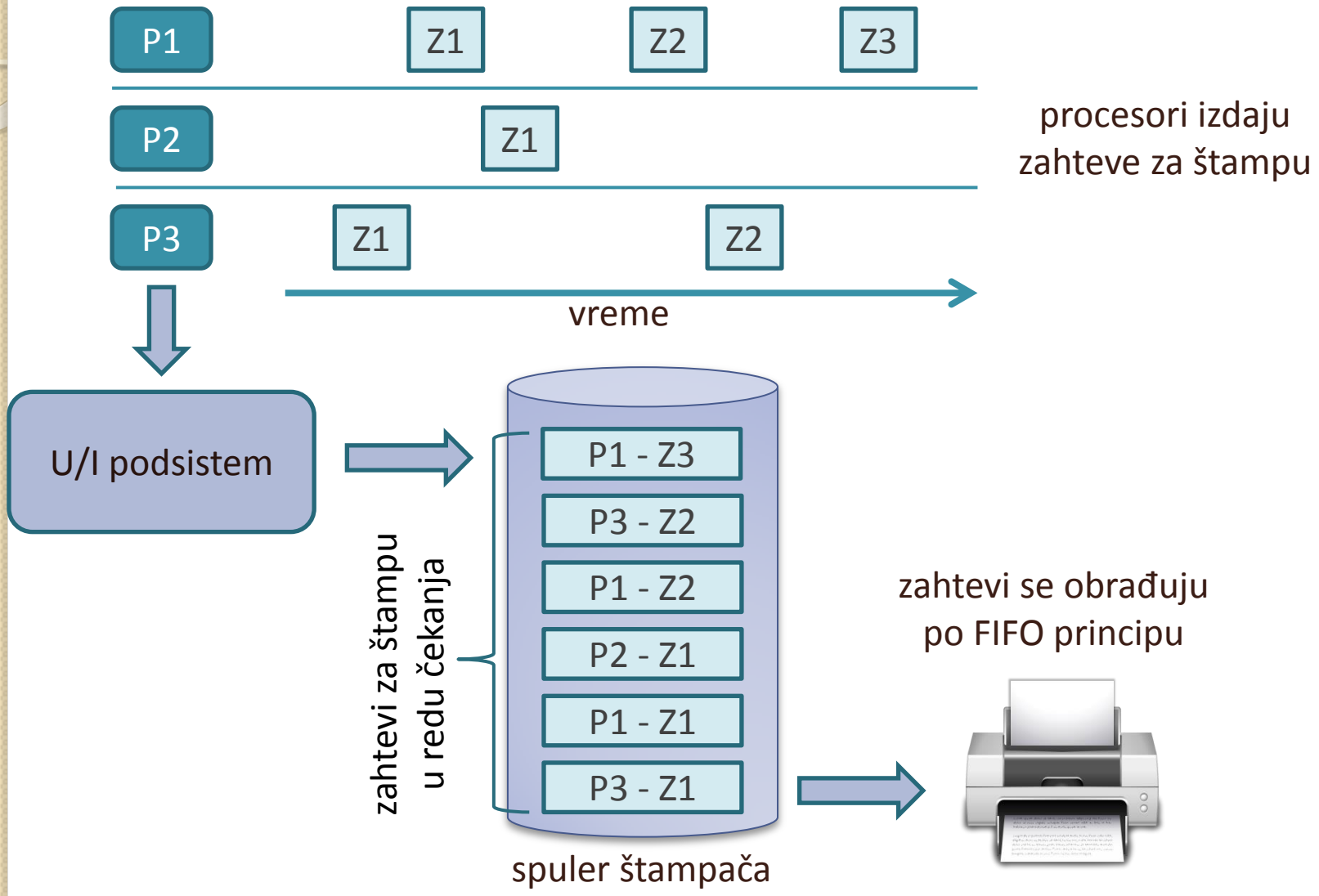
8.5 Usluge koje obezbeđuje U/I podsistem

Spuler (*spooler*)

- prednosti
  - proces relativno brzo obavlja postavljanje svog zahteva u spul bafer
    - nakon toga je slobodan da dalje obavlja svoje aktivnosti
    - primer: štampanje dokumenta od 1000 stranica
  - nedeljivi uređaji se koriste kao prividno deljivi
  - time se omogućava većem broju procesa da kvazi-istovremeno koriste uređaj

8.5 Usluge koje obezbeđuje U/I podsistem

Spuler (*spooler*)



8.5 Usluge koje obezbeđuje U/I podsistem

Prevođenje zahteva u U/I operacije

- primer: čitanje datoteke sa diska
  - korisnik zadaje ime datoteke koju želi da pročita
  - OS prevodi logičko ime u fizičku adresu prvog bloka datoteke na uređaju (disku)
  - ignorisaćemo sam proces prevođenja imena u fizičku adresu prvog bloka datoteke
- slučaj blokirajućeg čitanja jednog bloka sa diska
  1. proces šalje blokirajući sistemski poziv read()
  2. sistemski poziv proverava da li su ulazni parametri korektni i ako jesu nastavlja dalje
  3. proverava se stanje keša
     - ako zahtev može tako da se zadovolji podaci se vraćaju procesu iz keša
     - time se ciklus čitanja završava bez upotrebe samog uređaja
  4. ako podaci nisu u kešu, izvršiće se pravi U/I ciklus, što znači da proces mora da se blokira
     - pre blokade se poziva drajver

8.5 Usluge koje obezbeđuje U/I podsistem

Prevođenje zahteva u U/I operacije

5. drajver alocira kernelski bafer za prijem podataka i ubacuje zahtev za izvršenjem operacije u red čekanja
   - po nekom kriterijumu zahtev dolazi na red
   - drajver programira odgovarajući kontroler
6. kontroler obavlja U/I operaciju i odgovarajući transfer podataka
7. transfer se odvija ili u PIO režimu ili preko DMA kontrolera
8. po okončanju transfera kontroler postavlja prekidni signal koji se obrađuje
9. drajver prima prekidni signal koji ukazuje da je transfer završen, obrađuje status i obaveštava jezgro da je posao završen
10. jezgro prebacuje podatke u korisnički prostor i potom ukida blokadu sa procesa (tranzicija WAIT-READY u dijagramu stanja procesa)
11. proces nastavlja sa izvršenjem aktivnosti kada mu dispečer dodeli procesor na korišćenje

8.6 Performanse U/I podsistema

- performanse ulazno-izlaznog podsistema mogu se povećati ukoliko se
  - balansirano koristi procesor, memorija i U/I podsistem
    - preopterećenje u jednom delu dovodi do besposlenosti u drugom delu
  - koriste DMA kontroleri i keš mehanizmi
  - smanji broj prekida povećanjem jedinične količine transfera ili uvođenjem tehnike prozivanja (ukoliko se zaposlenost čekanjem može minimizovati)
  - smanji broj prebacivanja konteksta procesa
  - smanji broj kopiranja istih podataka na putu od aplikacije do U/I uređaja
- uopšteno, performanse se mogu uvećati na sledećim nivoima
  - nivo aplikativnog softvera
  - nivo drajvera
  - nivo hardvera