

# Upravljanje diskovima



# Kategorije ulazno-izlaznih uređaja

- Ulazni-izlazni uređaji mogu se grupirati u tri kategorije:

## Ljudski čitljivi

- Pogodno za komunikaciju s korisnikom računala
- Pisači, terminali, video prikaz, tipkovnica, miš

## Strojno čitljivi

- Pogodno za komunikaciju s elektroničkom opremom
- Diskovni pogoni, USB ključevi, senzori, kontroleri

## Komunikacijski

- Pogodno za komunikaciju s udaljenim uređajima
- Modemi, upravljački programi za digitalne linije

# Razlike u ulazno-izlaznim uređajima

- Uređaji se razlikuju u brojnim područjima:

## *Brzina prijenosa podataka*

- Postoje razlike između brzina prijenosa podataka

## *Primjena*

- Ovisi o uređaju na koji se stavlja

## *Složenost kontrole*

- Učinak na operativni sustav ovisi i složenosti ulazno-izlaznog modula koji kontrolira uređaj

## *Jedinica prijenosa*

- Podaci se mogu prenositi kao tok ili u većim blokovima

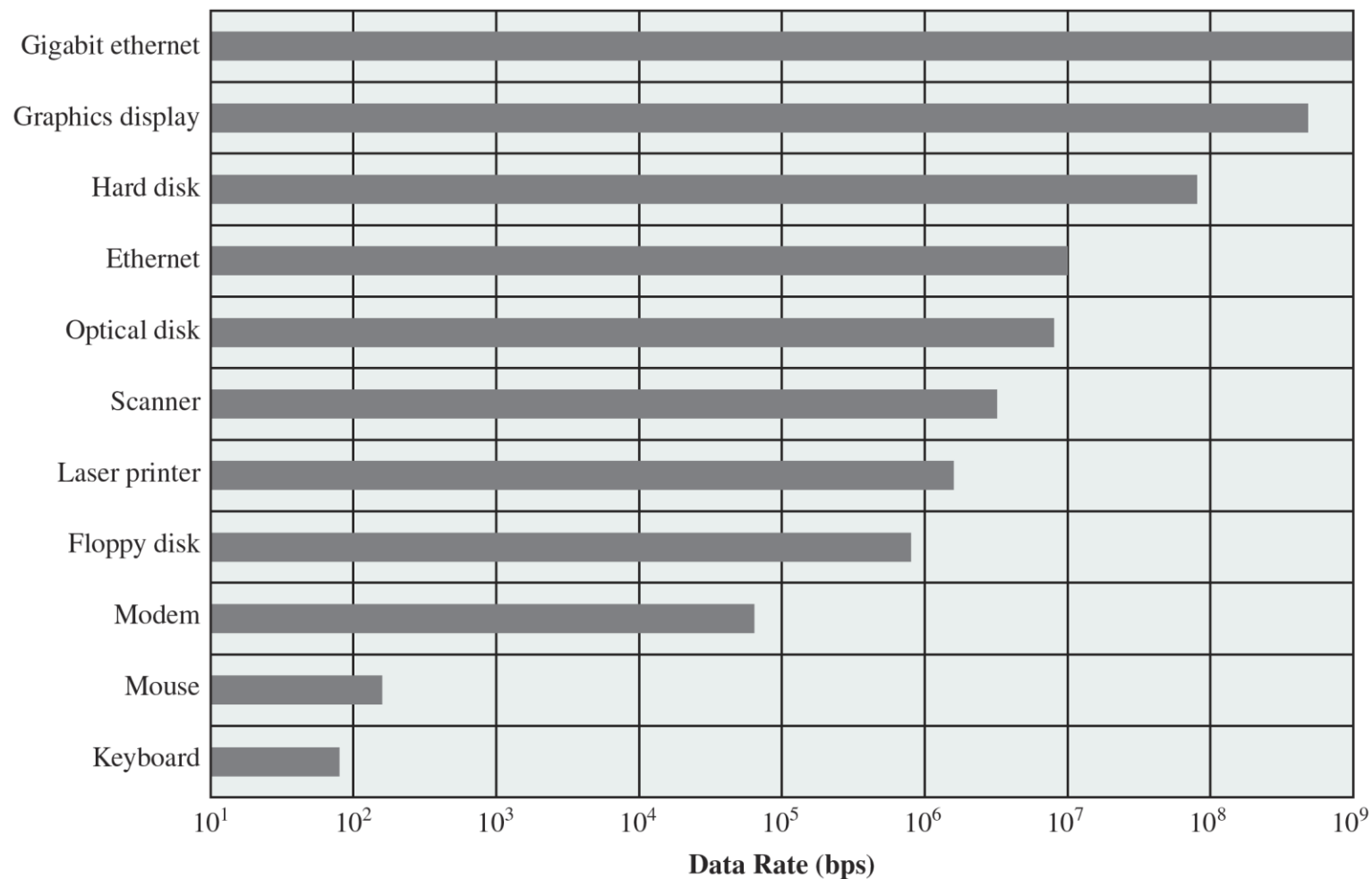
## *Prikaz podataka*

- Različiti uređaji koriste različite sheme kodiranja podataka

## *Uvjeti pogreške*

- Priroda pogrešaka, način na koji se prijavljuju

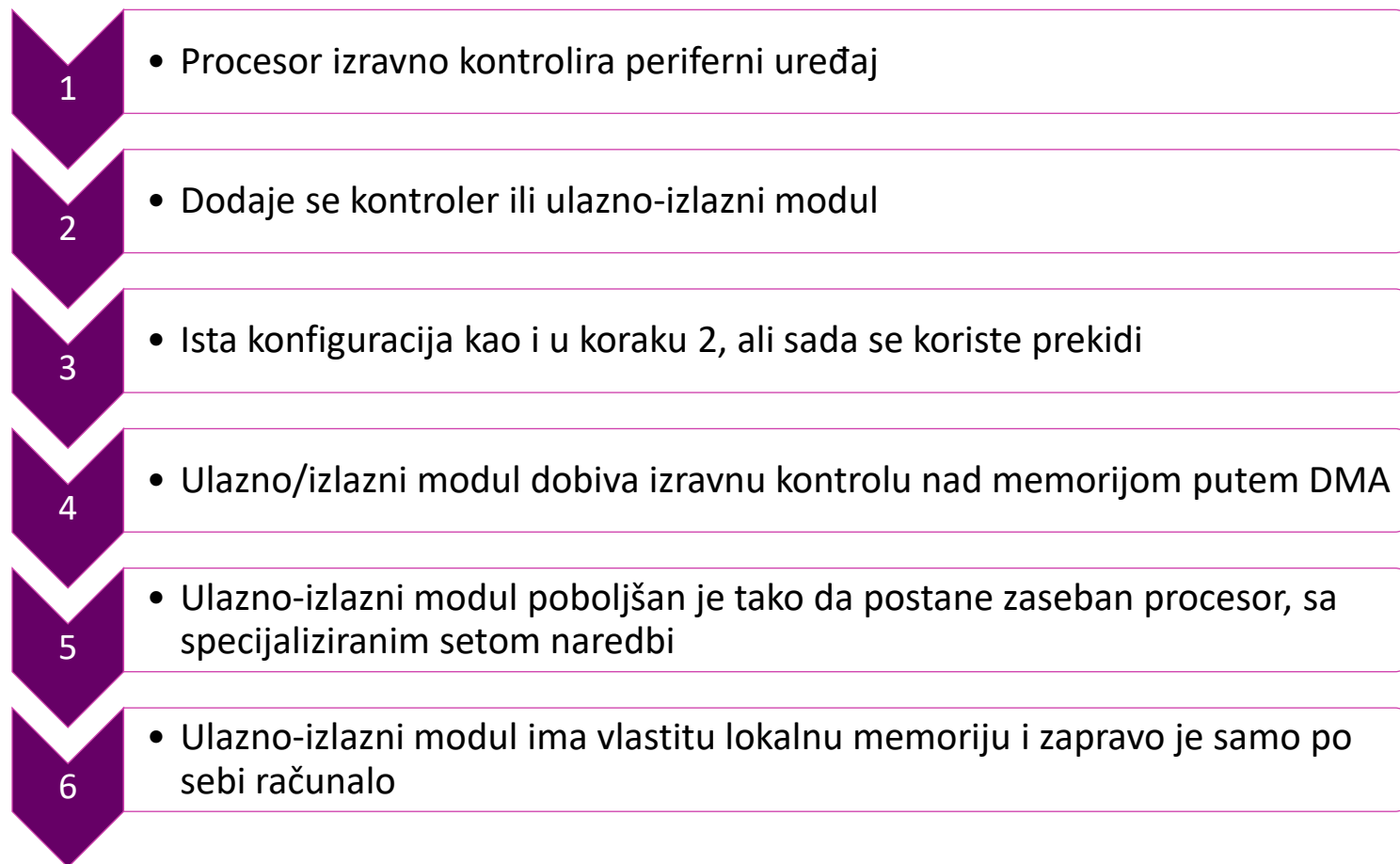
# Uobičajene brzine prijenosa podataka



# Organizacija ulazno/izlaznog sučelja

- Programirano ulazno/izlazno sučelje
  - Procesor izdaje ulazno/izlaznu naredbu u ime procesa ulazno-izlaznom modulu; taj proces tada zauzet čeka da se operacija dovrši prije nastavka
- Prekidno ulazno/izlazno sučelje
  - Procesor izdaje ulazno-izlaznu naredbu u ime procesa
    - Ako se ne blokira – procesor nastavlja izvršavati upute iz postupka koji je izdao ulazno-izlaznu naredbu
    - Ako se blokira - sljedeća uputa koju procesor izvršava je iz OS-a, što će trenutni proces staviti u blokirano stanje i izvršavati drugi proces
- Izravnog pristupa memoriji (DMA)
  - DMA modul kontrolira razmjenu podataka između glavne memorije i ulazno-izlaznog modula

# Evolucija ulazno/izlazne funkcije



# Ciljevi dizajna

## Efikasnost

- Veliki naponi u I/O dizajnu
- Važno jer su ulazno-izlazne operacije često usko grlo
- Većina I/O uređaja izuzetno je spora u usporedbi s glavnim memorijom i procesorom
- Područje na kojem se najviše radi je komunikacija s diskovima

## Općenitost

- Poželjno je postupati sa svim uređajima na ujednačen način
- Odnosi se na način na koji procesi prikazuju ulazno-izlazne uređaje i način na koji operativni sustav upravlja ulazno-izlaznim uređajima
- Raznolikost uređaja otežava postizanje istinske općenitosti
- Korištenje hijerarhijskog, modularnog pristupa dizajnu funkcije ulazno-izlazne funkcije
-

# Međupohranjivanje

- Kako bi se izbjegli gubitci i neučinkovitost, ponekad je prikladno izvršiti prijenose ulaza prije podnošenja zahtjeva i izvršiti izlazne prijenose neko vrijeme nakon podnošenja zahtjeva

## Uređaj orijentiran blokove

- Pohranjuje informacije u blokove koji su obično fiksne veličine
- Prijenosi se vrše jedan po jedan blok
- Moguće je referencirati podatke prema broju bloka
- Primjeri su diskovi i USB ključevi

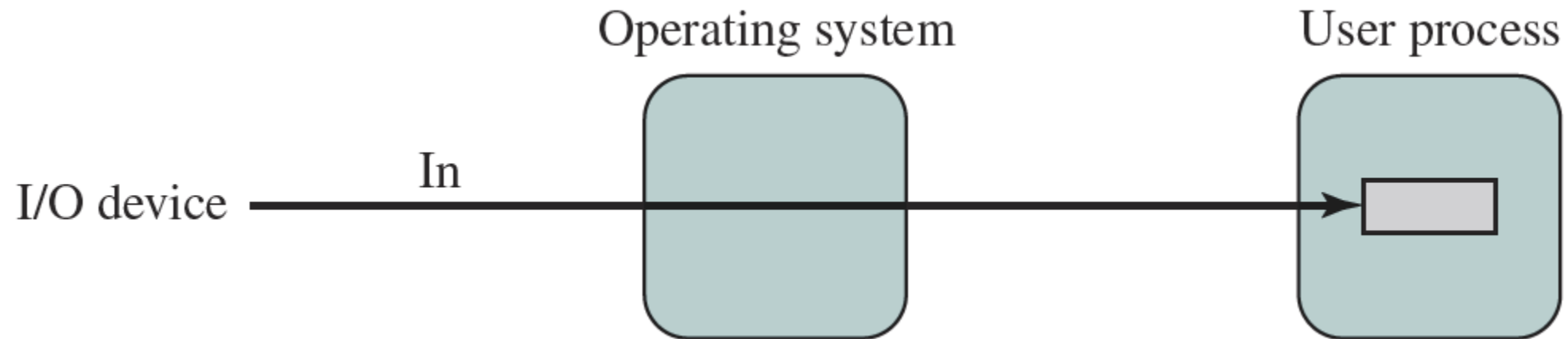
## Uređaj orijentiran na tokove

- Prenosi podatke kao tok bajtova
- Nema strukture bloka
- Primjeri su terminali, pisači, komunikacijski priključci, miš i drugi pokazivački uređaji te većina drugih uređaja koji nisu sekundarna pohrana



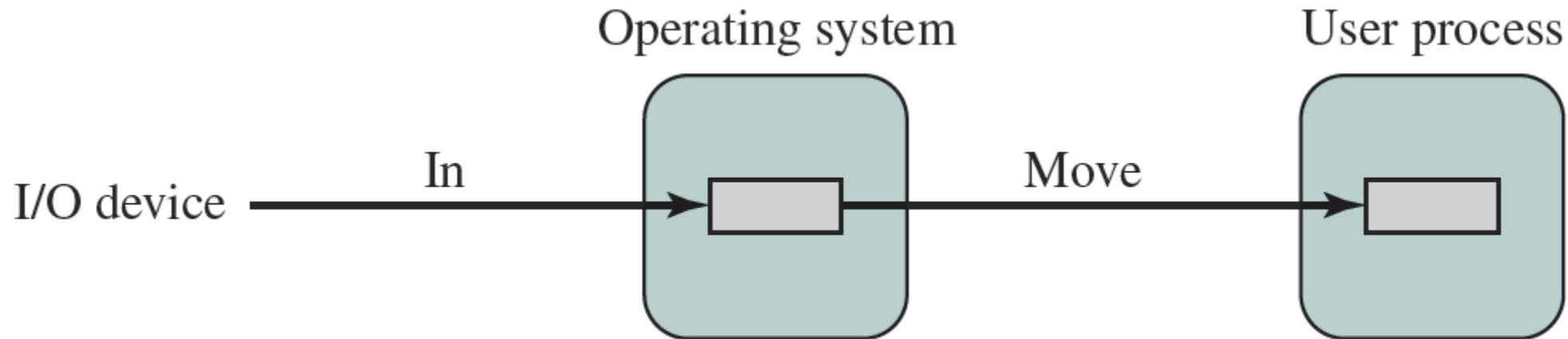
# Bez međuspremnik

- Bez međuspremnik, OS izravno pristupa uređaju kada mu je to potrebno



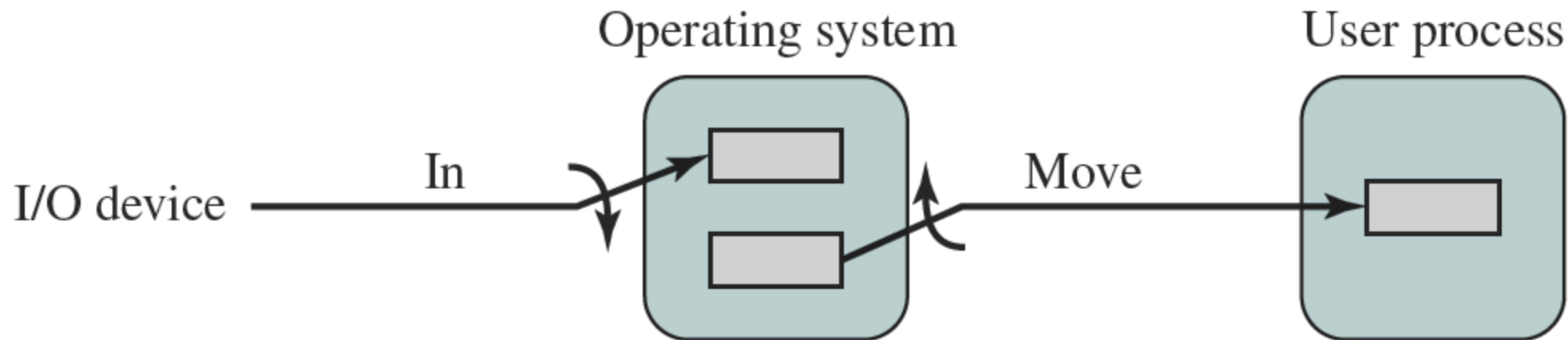
# Jedan međuspremnik

- Najjednostavnija vrsta podrške koju operativni sustav može pružiti
- Kada korisnički proces izda ulazno/izlazni zahtjev, OS operaciji dodjeljuje međuspremnik u sistemskom dijelu glavne memorije



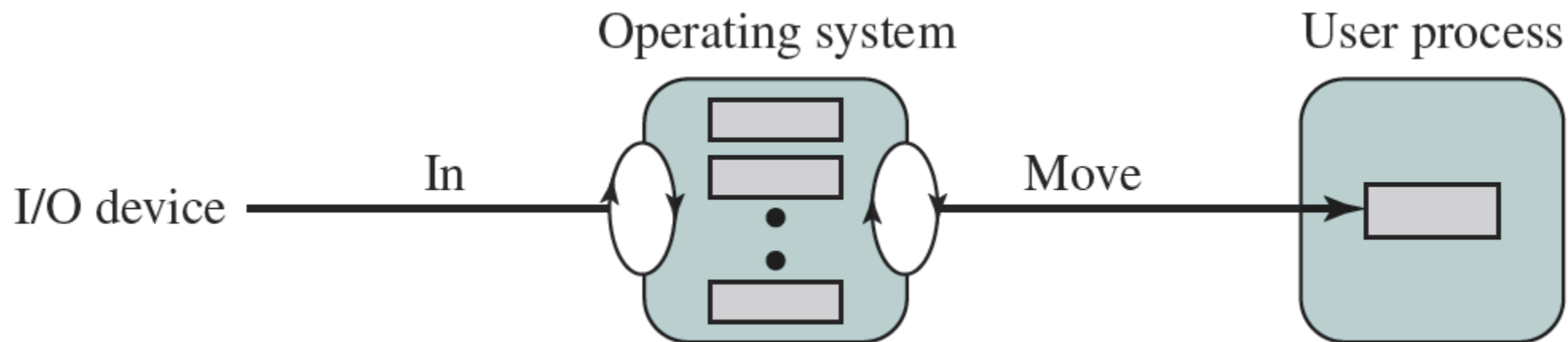
# Dvostruki međuspremnik

- Dodjeljivanje dva systemska međuspremnika operaciji
- Proces sada prenosi podatke u ili iz jednog međuspremnika dok operativni sustav prazni ili ispunjava drugi međuspremnik
- Poznato i kao zamjena međuspremnika



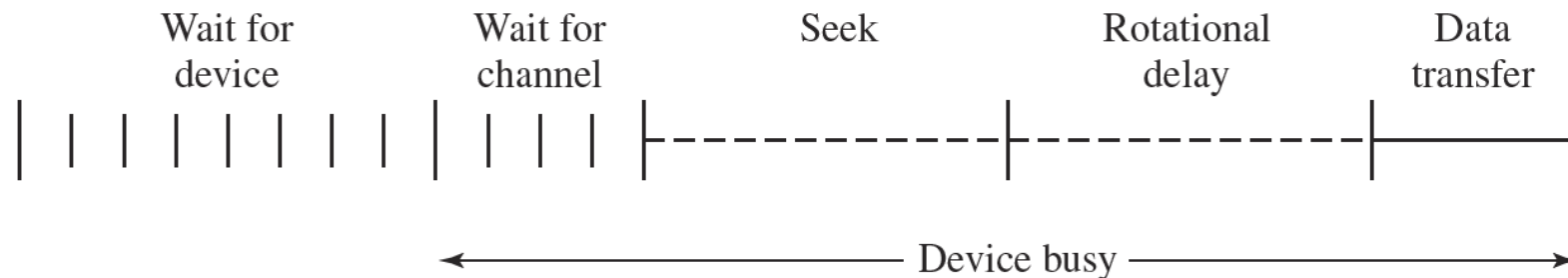
# Kružni međuspremnik

- Kada se koristi više od dva međuspremnika, prikupljanje međuspremnika naziva se kružnim zaštitnim slojem
- Svaki pojedinačni međuspremnik je jedna jedinica u kružnom međuspremniku



# Parametri performansi diska

- Stvarni detalji ulazno-izlazne operacije diska ovise o:
  - Računalnom sustavu
  - Operativnom sustavu
  - Prirodi ulazno-izlaznog kanala i hardvera kontrolera diska



# Parametri performansi diska

- Kada diskovni pogon radi, disk se okreće konstantnom brzinom
- Za čitanje ili pisanje glave mora biti postavljena na željenoj stazi i na početku željenog sektora na toj stazi
- Odabir staze uključuje pomicanje glave u pokretnom sustavu ili elektronički odabir jedne od glava na sustavu s fiksnom glavom
- Na pokretnom sustavu vrijeme potrebno za pozicioniranje glave na stazi poznato je kao vrijeme traženje (*seek time*)
- Vrijeme potrebno da početak sektora dođe do glave poznato je kao rotacijsko kašnjenje
- Zbroj vremena traženja i kašnjenja rotacije jednak je vremenu pristupa

# Vrijeme traženja

- Vrijeme potrebno za premještanje diskne ruke na traženi zapis
- Sastoji se od dvije ključne komponente:
  - Početno vrijeme pokretanja
  - Vrijeme potrebno za prelazak staze koje se moraju prijeći nakon što je pristupna ruka u stazi
- Vrijeme poravnivanja
  - Vrijeme nakon pozicioniranja glave iznad ciljne staze dok se ne potvrdi identifikacija
- Mnogo poboljšanja dolazi od manjih i lakših komponenti diska
- Tipičan prosjek vremena traženja na suvremenim tvrdim diskovima je ispod 10ms

# Performanse diska

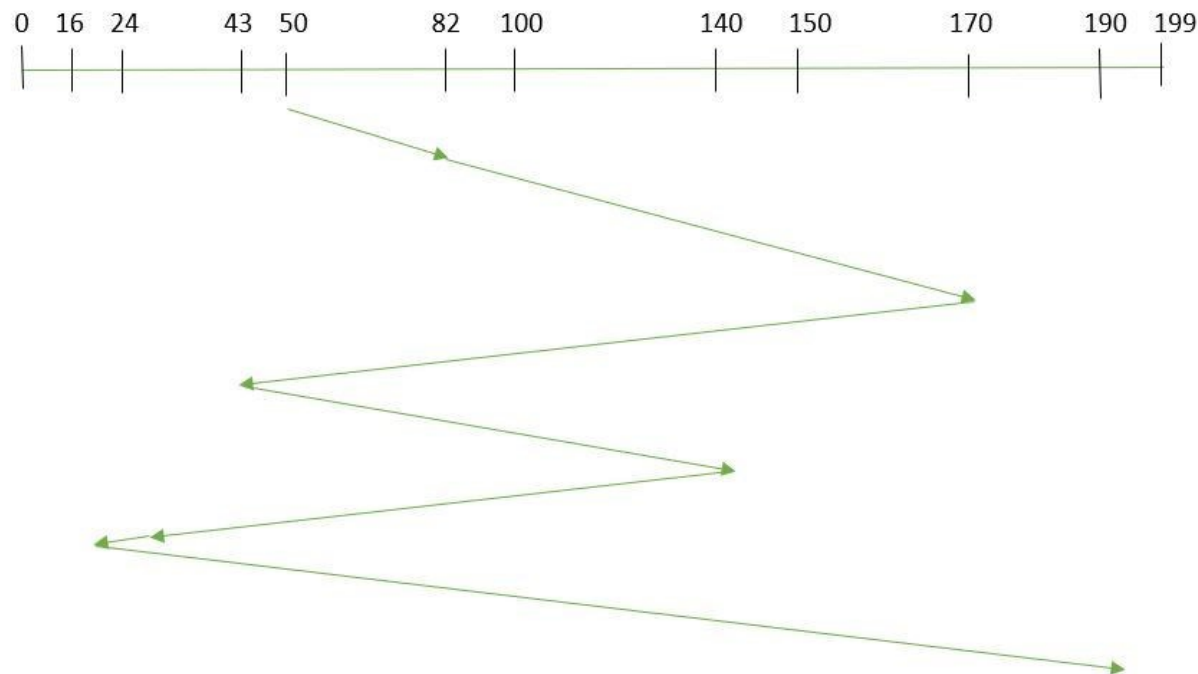
- Rotacijsko kašnjenje
  - Vrijeme potrebno da se adresirano područje diska rotira u položaj na kojem mu može pristupiti glava za čitanje/pisanje
  - Diskovi se okreću brzinom od 3.6000 o/min do 15.000 o/min



# First Come – First Serve (FCFS – FIFO)

- Procesi u sekvencijalnom redoslijedu
- Pošteno prema svim procesima
- Približno nasumično zakazivanje performansi ako postoji mnogo procesa koji se natječu

Trenutna pozicija glave za čitanje/pisanje je: 50  
Redoslijed zahtjeva je: 82, 170, 43, 140, 24, 16, 190



Ukupno vrijeme traženja  $= (82-50) + (170-82) + (170-43) + (140-43) + (140-24) + (24-16) + (190-16) = 642$  | Prosječna duljina traženja = 91.7

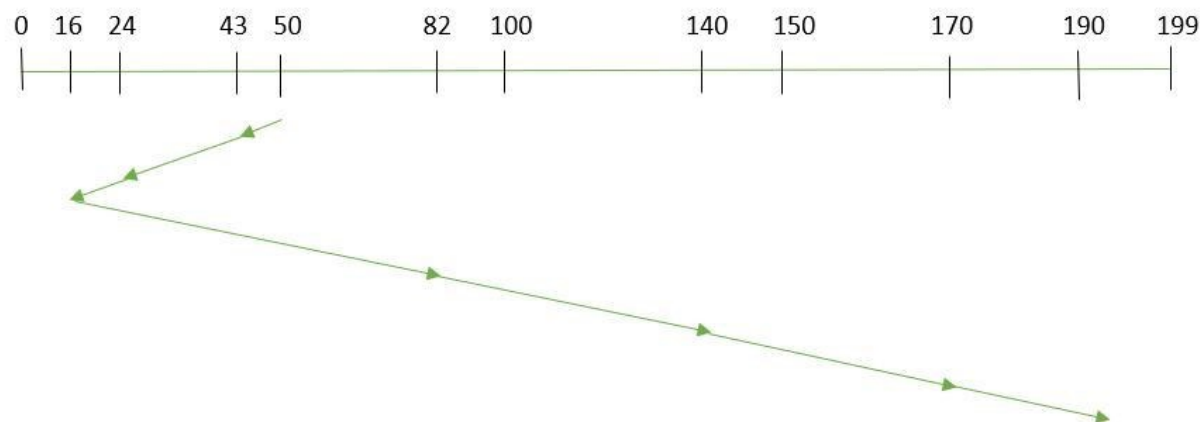
# Prioritet (PRI)

- Kontrola zakazivanja je izvan kontrole softvera za upravljanje diskovima
- Cilj nije optimizirati korištenje diska, već ispuniti druge ciljeve
- Kratke obrade i interaktivni poslovi imaju veći prioritet
- Pruža dobro interaktivno vrijeme odziva
- Dulji poslovi možda će morati čekati pretjerano dugo
- Loša politika za sustave baza podataka

# Shortest Seek Time First (SSTF)

- Odabere ulazno/izlazni zahtjev diska koji zahtijeva najmanje pomicanje ruke diska s trenutnog položaja
- Uvijek odabere minimalno vrijeme traženja
- Ako je isto na obje stranenastavlja se kretati u istom smjeru

Trenutna pozicija glave za čitanje/pisanje je: 50  
Redoslijed zahtjeva je: 82,170, 43,140, 24,16, 190

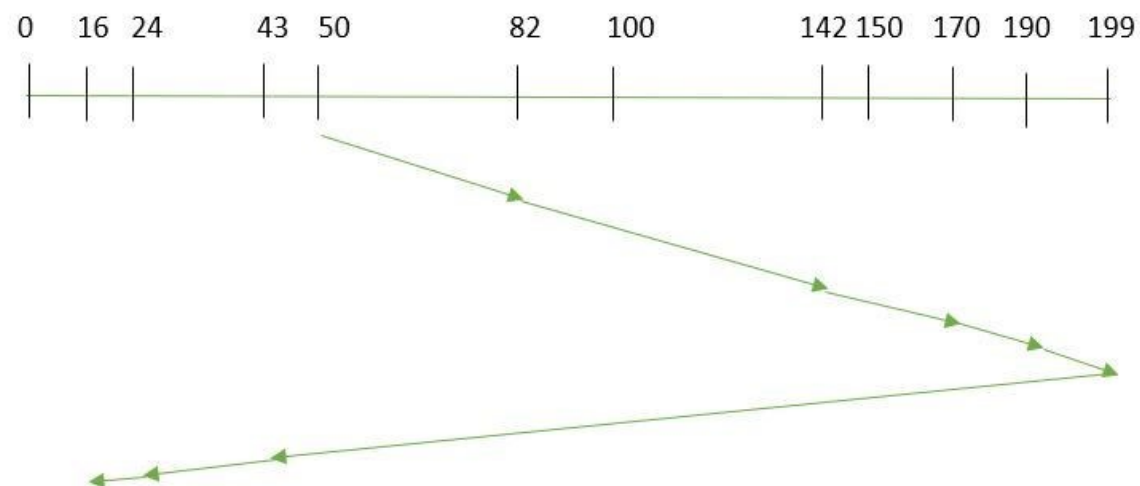


Ukupno vrijeme traženja =  $(50-43)+(43-24)+(24-16)+(82-16)+(140-82)+(170-140)+(190-170) = 208$  | Prosječna duljina traženja = 29.7

# SCAN

- Poznat i kao algoritam dizala
- Ruka se kreće samo u jednom smjeru
  - Zadovoljava sve nepodmirene zahtjeve dok ne dođe do posljednjeg zapisa na disku, a zatim kreće u suprotnom smjeru
- Favorizira mjesta čiji su zahtjevi za staze najbliže najdubljim i najudaljenijim stazama i pogoduje najnovijim zahtjevima

Trenutna pozicija glave za čitanje/pisanje je: 50 kreće se prema kraju  
Redoslijed zahtjeva je: 82.170, 43.140, 24.16.190

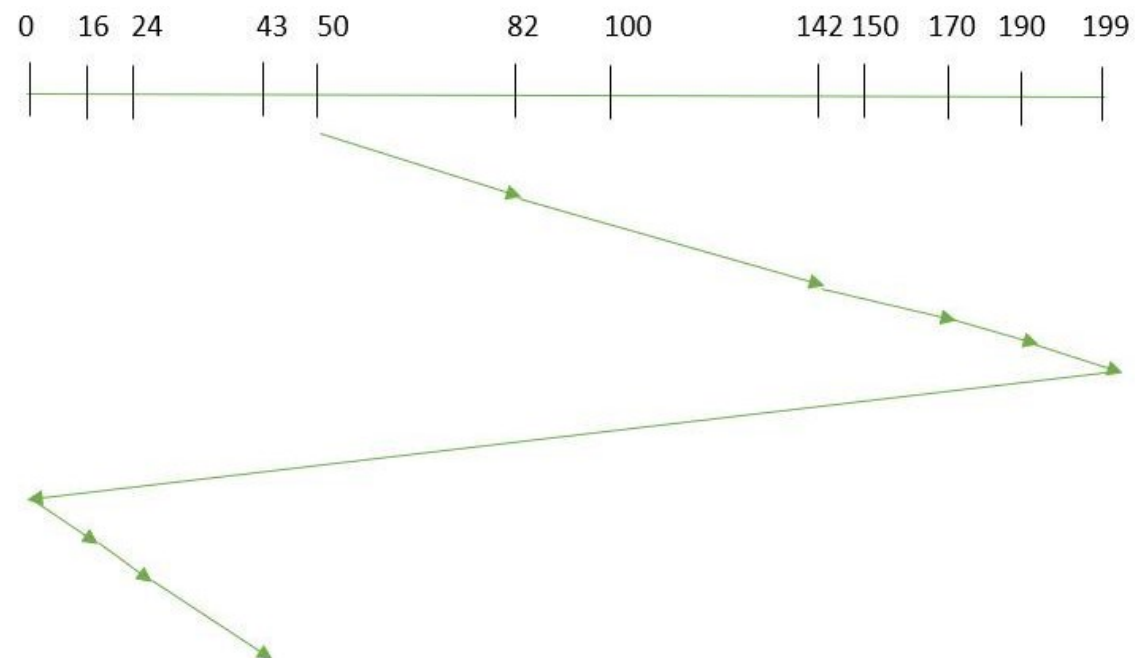


Ukupno vrijeme traženja =  $(199-50)+(199-16) = 332$  | Prosječna duljina traženja = 47.4

# C-SCAN (Circular SCAN)

- Ograničava skeniranje samo u jednom smjeru
- Kada je posljednji blok pročitao u jednom smjeru, ruka se vraća na suprotni kraj diska i ponovno čita podatke u istom smjeru

Trenutna pozicija glave za čitanje/pisanje je: 50 kreće se prema kraju  
Redoslijed zahtjeva je: 82.170, 43.140, 24.16.190

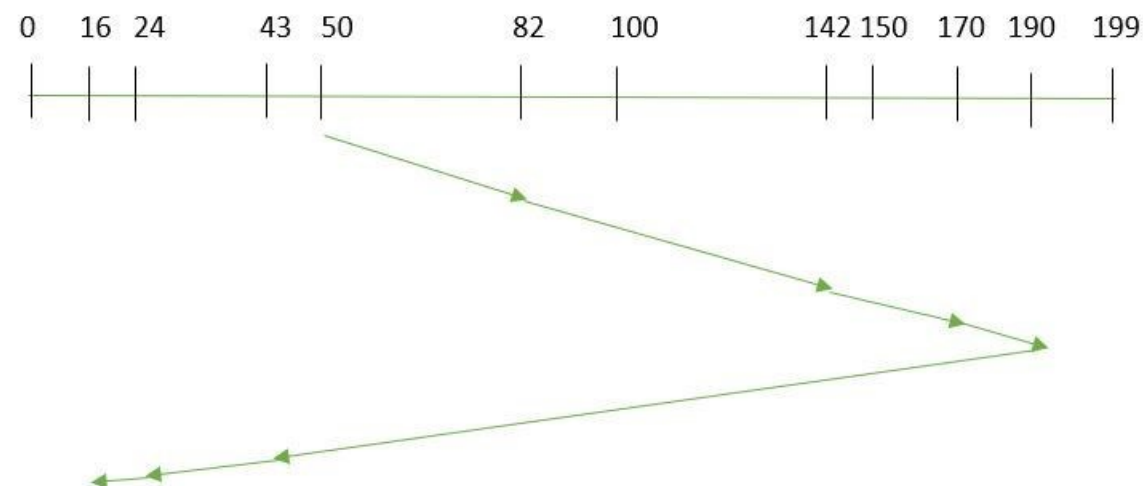


Ukupno vrijeme traženja =  $(199-50)+(199-0)+(43-0) = 391$  | Prosječna duljina traženja = 55.8

# LOOK

- Sličan je algoritmu SCAN, osim razlike da ruka unatoč odlasku na kraj diska ide samo na posljednji zahtjev koji treba pročitati i zatim mjenja svoj smjer

Trenutna pozicija glave za čitanje/pisanje je: 50 kreće se prema kraju  
Redoslijed zahtjeva je: 82.170, 43.140, 24.16.190

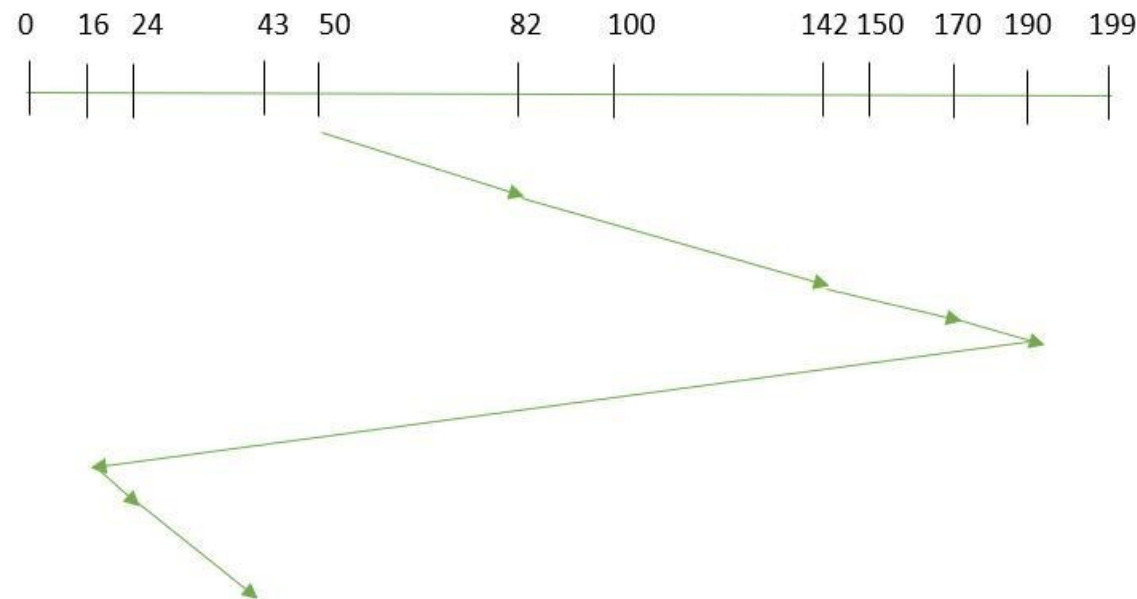


Ukupno vrijeme traženja =  $(190-50)+(190-16) = 314$  | Prosječna duljina traženja = 44.8

# C-LOOK

- C-LOOK je sličan C-SCAN algoritmu.
- U C-LOOK-u ruka unatoč odlasku na kraj ide samo do posljednjeg zahtjeva koji treba pročitati, a zatim ide na skače na suprotni zahtjev na drugom kraju diska.

Trenutna pozicija glave za čitanje/pisanje je: 50 kreće se prema kraju  
Redoslijed zahtjeva je: 82.170, 43.140, 24.16.190



Ukupno vrijeme traženja =  $(190-50)+(190-16)+(43-16) = 341$  | Prosječna duljina traženja = 48.7

# N-Step-SCAN

- Segmentira red čekanja zahtjeva za diskom u više redova čekanja duljine  $N$
- Redovi čekanja se obrađuju jedan po jedan pomoću SCAN-a
- Dok se red čekanja obrađuje, novi zahtjevi dodaju se u neki drugi red čekanja
- Ako je na kraju skeniranja dostupno manje od  $N$  zahtjeva, svi se obrađuju sljedećim skeniranjem

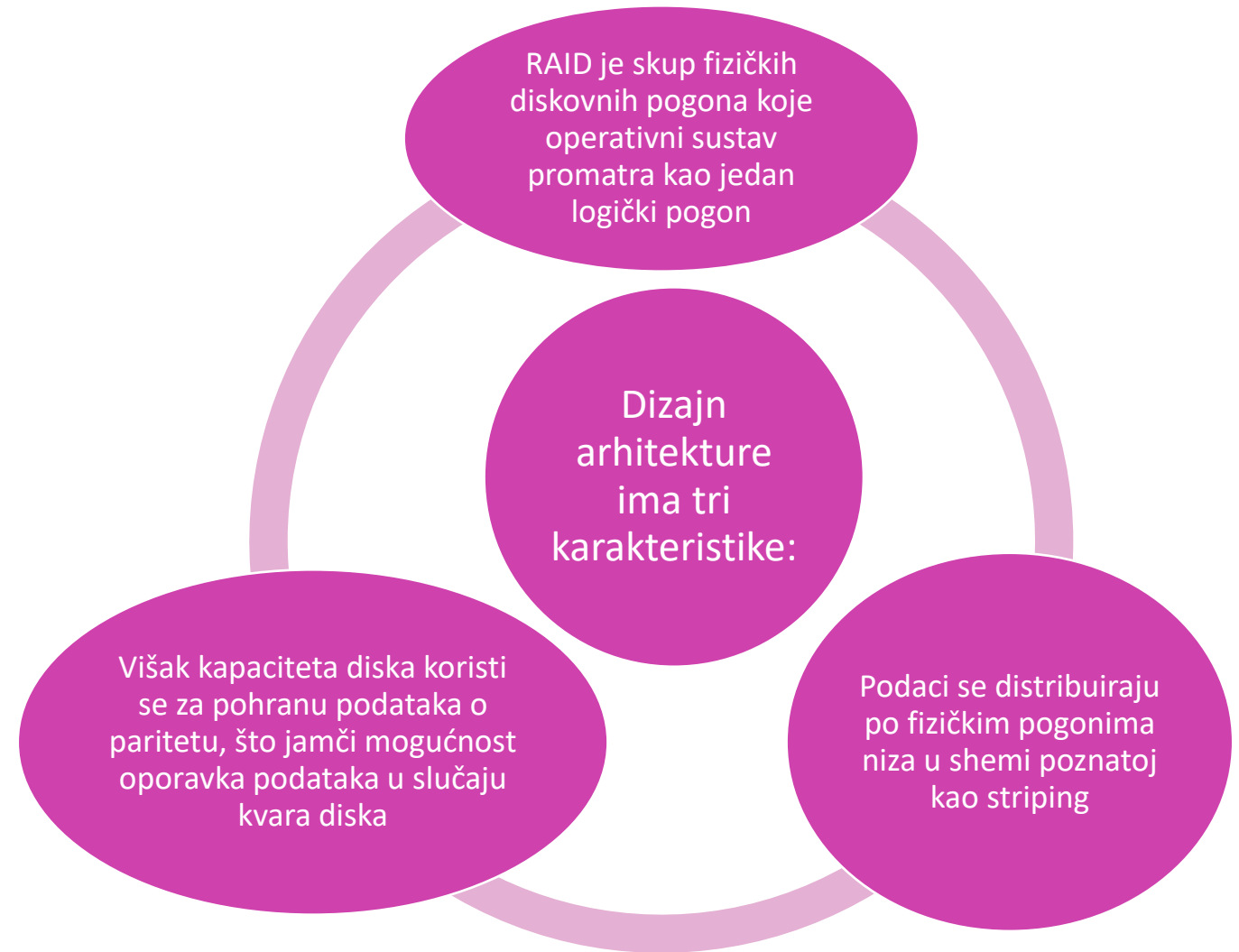


# FSCAN

- Koristi dva reda čekanja
- Kada skeniranje započne, svi zahtjevi su u jednom od redova čekanja, dok je drugi prazan
- Tijekom skeniranja svi novi zahtjevi stavljaju se u drugi red čekanja
- Dostava novih zahtjeva odgađa se dok se ne obrade svi stari zahtjevi

# RAID

- Redundantni niz neovisnih diskova
- Sastoji se od sedam razina, od nule do šest

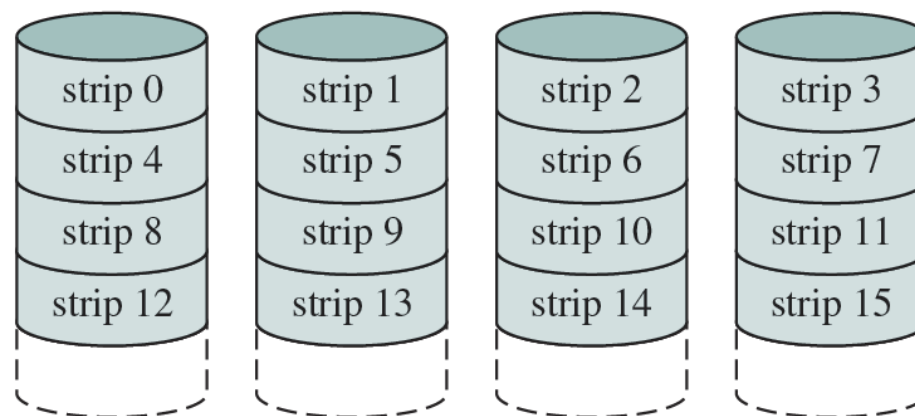


# RAID

- Izraz je izvorno skovan u radu skupine istraživača sa Sveučilišta California na Berkeleyu.
  - U radu su navedene različite konfiguracije i aplikacije te uvedene definicije RAID razina
- Strategija koristi više diskovnih pogona i distribuira podatke na takav način da omogućuje istovremeni pristup podacima s više pogona
  - Poboljšava ulazno-izlazne performanse i omogućuje lakše povećanje kapaciteta
- Jedinostveni doprinos je učinkovito rješavanje potrebe za redundancijom
- Koristi pohranjene podatke pariteta koji omogućuju oporavak podataka izgubljenih kvarom na diska

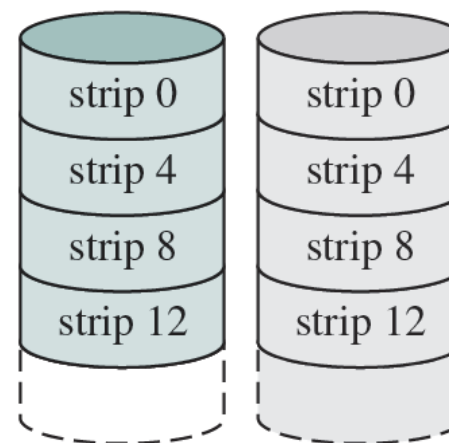
# RAID Level 0

- Nije pravi RAID jer ne uključuje redundanciju za pružanje zaštite podataka
- Podaci o korisniku i sustavu raspoređeni su na svim diskovima u polju
- Logički disk podijeljen je na trake



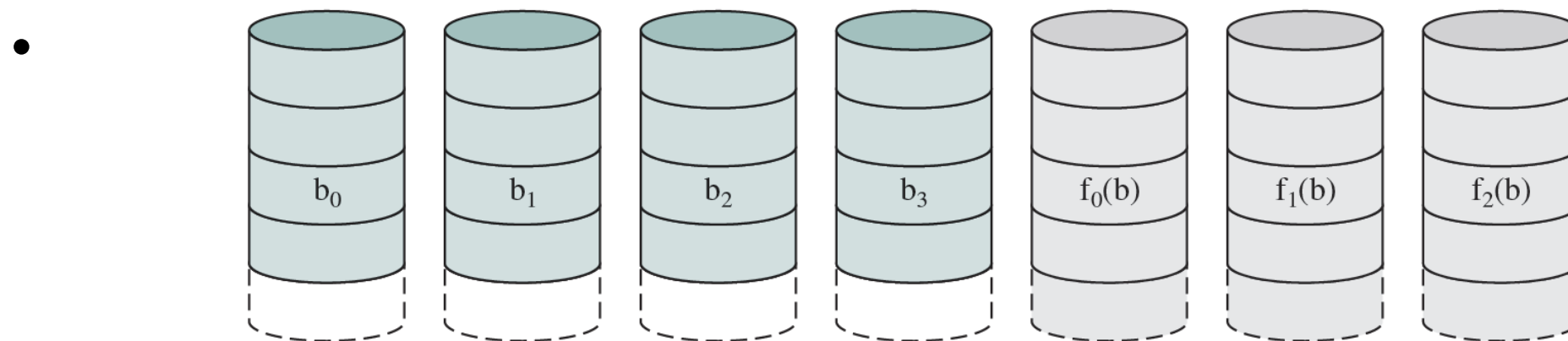
# RAID Level 1

- Redundancija se postiže jednostavnim dupliciranjem svih podataka
- Ne postoji "kazna za pisanje"
- Kada se disk pokvari, podacima se i dalje može pristupiti s drugog diska
- Glavni nedostatak je trošak



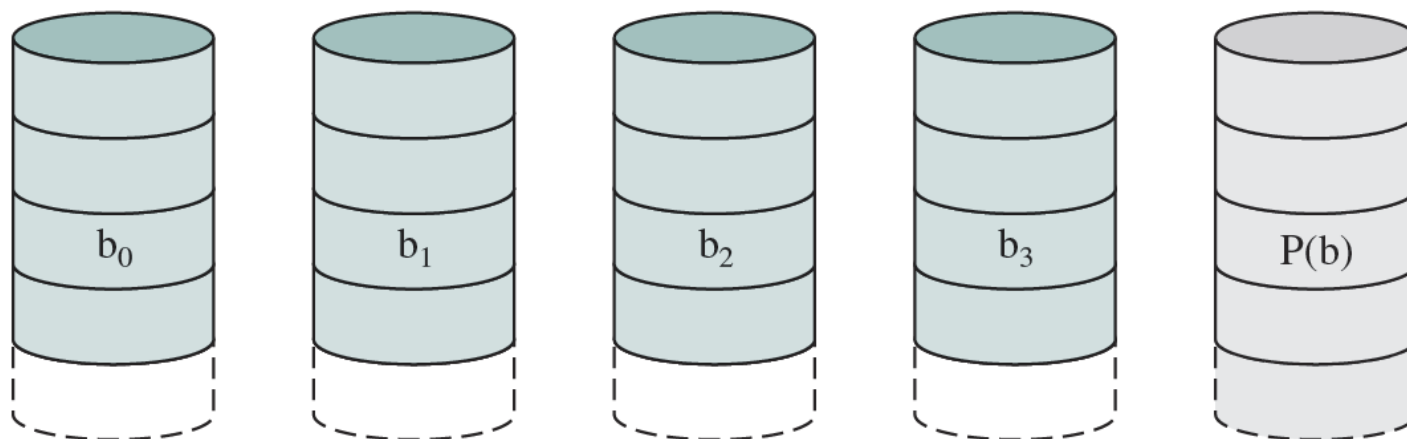
# RAID Level 2

- Koristi tehniku paralelnog pristupa
- Koriste se podatkovne pruge
- Obično se koristi hamming kôd
- Učinkovit izbor u okruženju u kojem se pojavljuju dosta grešaka na diskovima



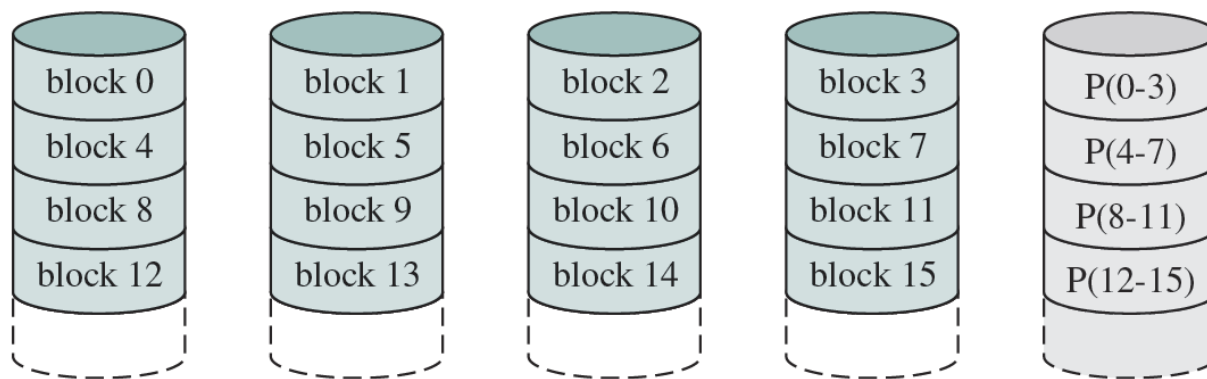
# RAID Level 3

- Potreban je samo jedan redundantni disk, bez obzira na veličinu polja diskova
- Koristi paralelni pristup, s podacima distribuiranim u malim trakama
- Može postići vrlo visoke brzine prijenosa podataka



# RAID Level 4

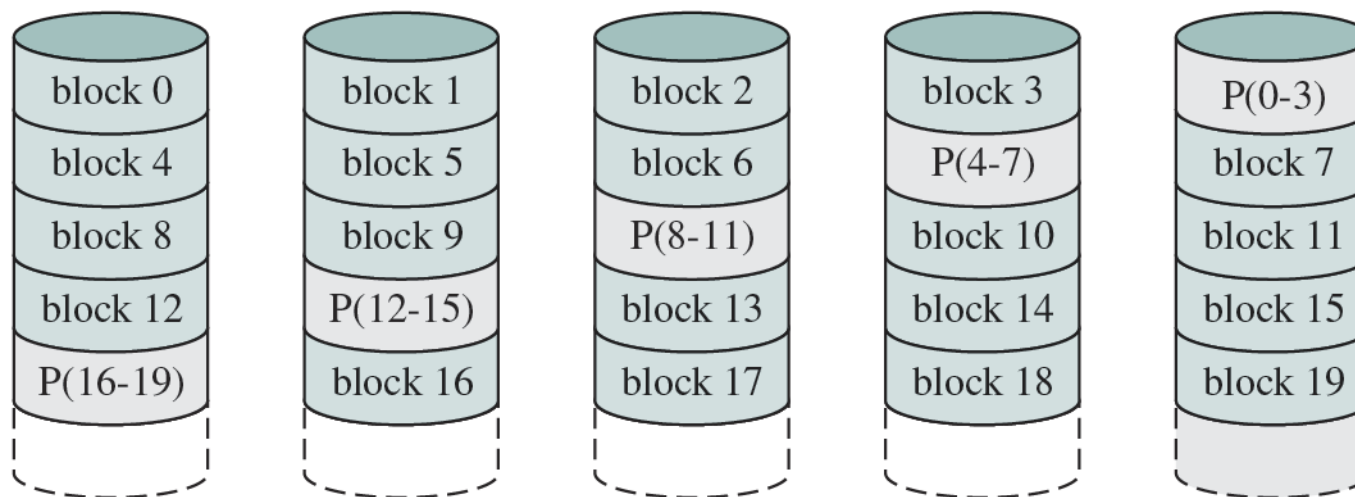
- Koristi neovisnu tehniku pristupa
- Bit-po-bit paritetna traka izračunava se na odgovarajućim trakama na svakom podatkovnom disku, a paritetni bitovi pohranjuju se u odgovarajuću traku na paritetnom disku
- Uključuje kaznu za pisanje kada se izvrši zahtjev za pisanjem male veličine





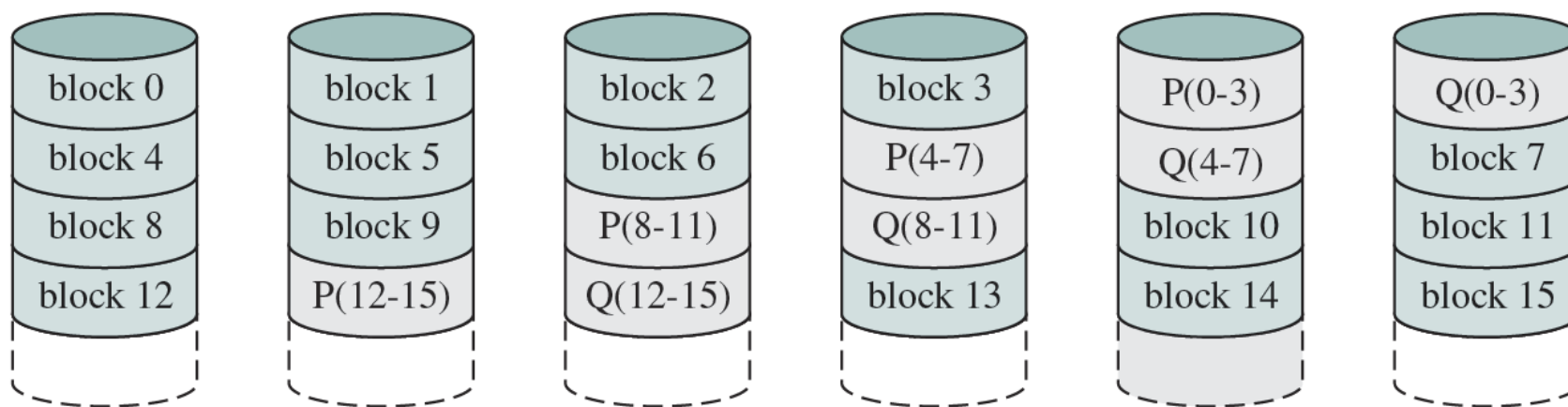
# RAID Level 5

- Slično RAID-4, ali distribuira paritetne bitove na svim diskovima
- Tipična dodjela je shema kružnog pisanja
- Ima obilježje da gubitak bilo kojeg diska ne rezultira gubitkom podataka



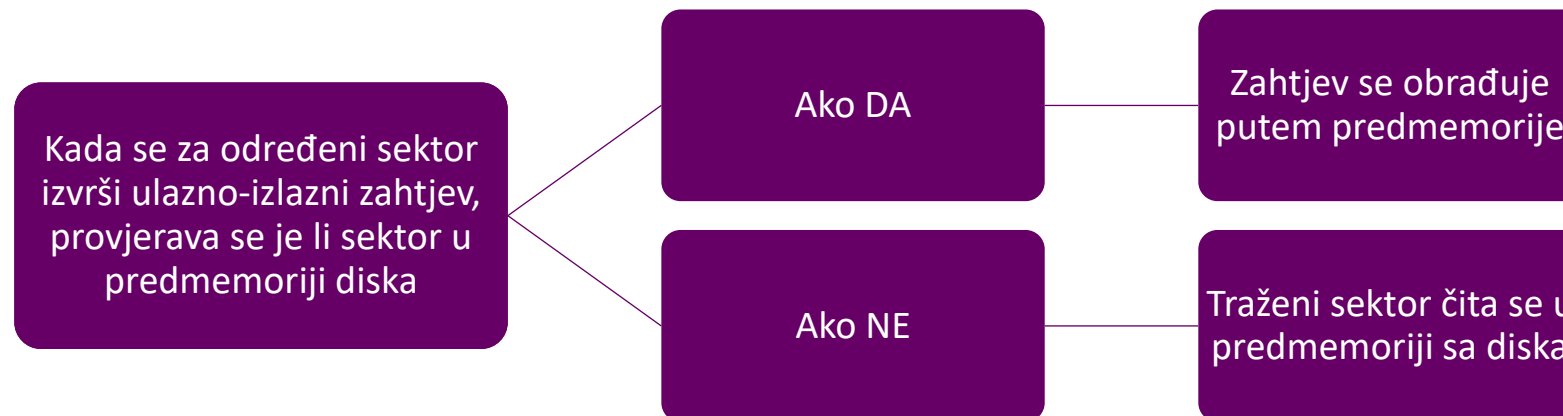
# RAID Level 6

- Dva različita izračuna pariteta provode se i pohranjuju u zasebne blokove na različitim diskovima
- Pruža izuzetno visoku dostupnost podataka
- Snosi značajnu kaznu za pisanje jer svako pisanje utječe na dva bloka pariteta



# Predmemorija diskova

- Predmemorija je manja i brža od glavne memorije i postavljena je između glavne memorije i procesora
- Smanjuje prosječno vrijeme pristupa memoriji iskorištavanjem načela lokaliteta
- Predmemorija diska je međuspremnik u glavnoj memoriji za diskovne sektore
- Sadrži kopiju nekih sektora na disku



# Least Recently Used (LRU)

- Najčešće korišten algoritam koji se bavi zamjenom
- Zamijenjen je blok koji je najduže bio u predmemoriji bez njegova korištenja
- Pokazivači upućuje na predmemoriju
  - Posljednji referentni blok nalazi se na vrhu stoga
  - Kada se blok referencira ili unosi u predmemoriju, on se postavlja na vrh stoga

# Least Frequently Used (LFU)

- Zamijenjen je blok koji ima najmanje korištenja
- Brojač je povezan sa svakim blokom
- Brojač se povećava svaki put kada se pristupi bloku
- Kada je potrebna zamjena, odabire se blok s najmanjim brojem

# UNIX predmemorija međuspremnik

- U osnovi je predmemorija diska
  - Ulazno/izlazne operacije s diskom obrađuju se kroz predmemoriju međuspremnik
- Prijenos podataka između predmemorije međuspremnik i korisničkog procesnog prostora uvijek se događa pomoću DMA-a
  - Ne troši nikakve cikluse procesora
  - Koristi cikluse sabirnice
- Održavaju se tri popisa:
  - Besplatni popis
    - Popis svih dijelova predmemorije dostupnih za dodjelu
  - Popis uređaja
    - Popis svih međuspremnik koji su trenutno povezani sa svakim diskom
  - Ulazno/izlazni red upravljačkog programa
    - Popis međuspremnik koji se izvršavaju ili čekaju ulazno/izlazno operacije na uređaju

# Red čekanja znakova

Koriste ga uređaji orijentirani na znakove

Terminali i pisači



Zapisan ulazno-izlaznim uređajem i pročitano procesom ili obrnuto

Koristi se model proizvođača/potrošača



Redovi čekanja znakova mogu se čitati samo jednom

Kako se svaki znak pročita, briše se

# Bez međuspreminika

- DMA između uređaja i procesnog prostora
- To je uvijek najbrža metoda za proces izvođenja
- Proces je zaključan u glavnoj memoriji i ne može se zamijeniti
- Ulazno-izlazni uređaj povezan je s postupkom za vrijeme trajanja prijenosa, što ga čini nedostupnim za druge procese



# Linux

- Vrlo slično implementacijama UNIX-a
- Povezuje posebnu datoteku sa svakim upravljačkim programom ulazno-izlaznog uređaja
- Prepoznaju se blok, znakovi i mrežni uređaji
- Zadani raspored diska u Linuxu 2.4 je Linux Elevator

Za Linux 2.6 algoritam Dizala povećan je s dva dodatna algoritma:

- Posljednim rokom (Deadline) ulazno/izlaznim planerom
- Iščekivanim (Anticipatory) ulazno/izlaznim planerom

# Elevator planer

- Održava jedan red čekanja za zahtjeve za čitanje i pisanje na disku te obavlja funkcije sortiranja i spajanja u redu čekanja
- Kada se novi zahtjev doda u red čekanja, četiri operacije razmatraju se redoslijedom:
  - Ako se zahtjev odnosi na isti sektor na disku ili neposredno susjedni sektor na neriješeni zahtjev u redu čekanja, tada se postojeći zahtjev i novi zahtjev spajaju u jedan zahtjev
  - Ako je zahtjev u redu čekanja dovoljno star, novi se zahtjev umeće na rep reda čekanja
  - Ako postoji odgovarajuće mjesto, novi zahtjev se umeće sortiranim redoslijedom
  - Ako nema odgovarajuće lokacije, novi zahtjev postavlja se na rep reda čekanja

# Deadline planer

- Dva problema manifestiraju se shemom dizala:
  - Zahtjev za udaljenim blokom može se odgoditi na dulje vrijeme jer se red čekanja dinamički ažurira
  - Niz zahtjeva za pisanje može blokirati zahtjev za čitanje na duže vrijeme i tako blokirati proces
- Deadline planer razvijen je 2002. godine
  - Ovaj planer koristi dva para redova čekanja
  - Osim što je svaki dolazni zahtjev smješten u red sortiranog dizala kao i prije, isti se zahtjev postavlja na repu pročitanoog FIFO reda čekanja za zahtjev za čitanje ili red čekanja FIFO-a za pisanje za zahtjev za pisanje
  - Kada je zahtjev izvršen, uklanja se s iz oba reda čekanja

# Anticipatory planer

- Elevator and deadline planer mogu biti kontraproduktivno ako postoje brojni sinkroni zahtjevi za čitanje
- U Linuxu se iščekivajući planer postavlja na deadline planer
- Kada se pošalje zahtjev za čitanje, iščekivajući planer uzrokuje kašnjenje sustava zakazivanja
- Velika je vjerojatnost da će aplikacija koja je izdala posljednji zahtjev za čitanjem izdati još jedan zahtjev za čitanje na istom području diska
- Taj će se zahtjev odmah upotpuniti
- U suprotnom, planer nastavlja pomoću deadline planer

# NOOP planer



Ovo je najjednostavnije među Linux I/O planerima

To je minimalni planer koji umeće ulazno/izlazne zahtjeve u FIFO red i koristi spajanje

Njegove glavne namjene uključuju blok uređaje koji se ne temelje na disku, kao što su memorijski uređaji, i specijalizirana softverska ili hardverska okruženja koja rade vlastito zakazivanje i trebaju samo minimalnu podršku jezgre OS-a

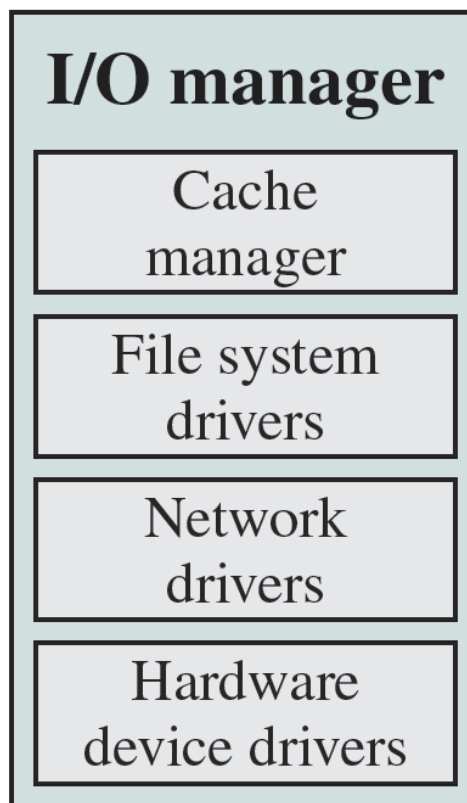
# Potpuno pošten I/O planer čekanja (CFQ)

- Razvijen je 2003. godine
- Zadani je I/O planer na Linuxu
- CFQ planer jamči pravednu raspodjelu ulazno-izlazne propusnosti diska među svim procesima
- Održava se po ulazno-izlaznim redovima čekanja procesa
  - Svakom se procesu dodjeljuje jedan red čekanja
  - Svaki red čekanja ima dodijeljeni vremenski isječak
  - Zahtjevi se šalju u ove redove čekanja i obrađuju se u kružnom rasporedu dodjele
- Kada planer obrađuje određeni red čekanja, a u tom redu čekanja više nema zahtjeva, čeka u stanju mirovanja unaprijed definirani vremenski interval za nove zahtjeve, a ako nema zahtjeva, nastavlja se na sljedeći red čekanja

# Predmemorija stranica – Linux

- Za Linux 2.4 i noviju postoji jedinstvena predmemorija stranica za sav promet između diska i glavne memorije
- Kada dođe vrijeme za pisanje prijavih stranica na disk, više njih se može posložiti i učinkovito zapisati
- Zbog načela vremenskog lokaliteta, stranice u predmemoriji stranice vjerojatno će se ponovno referencirati prije nego što se izbace iz predmemorije, čime će se smanjiti broj ulazno-izlaznih operacija sa diskom

# Windows ulazno/izlazni upravitelj

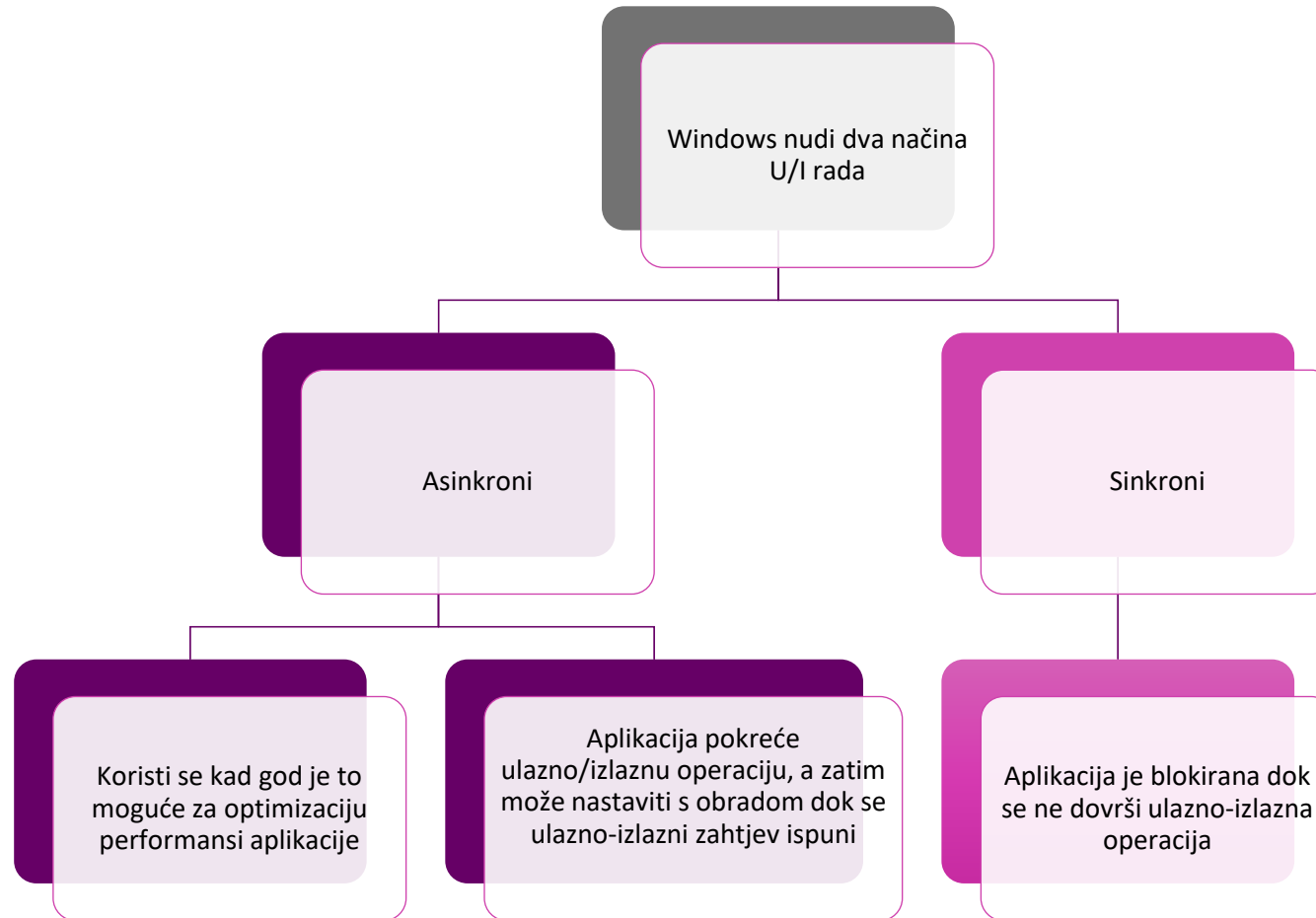




# Osnovne ulazno-izlazne funkcije

- Upravitelj predmemorije
  - Mapira područja datoteka u kernelsku virtualnu memoriju, a zatim se oslanja na upravitelj virtualne memorije za kopiranje stranica u datoteke na disku i iz njih
- Upravljački programi datotečnog sustava
  - Šalje ulazno-izlazne zahtjeve upravljačkim programima softvera koji upravljaju prilagodnikom hardverskog uređaja
- Mrežni upravljački programi
  - Windows uključuje mogućnosti integriranog umrežavanja i podršku za udaljene datotečne sustave
  - Objekti se implementiraju kao softverski upravljački programi
- Upravljački programi hardverskih uređaja
  - Izvorni kod upravljačkih programa uređaja sa sustavom Windows prenosiv je u različitim vrstama procesora

# Asinkroni i sinkroni I/O



# Završetak ulazni/izlazne operacije

- Windows nudi pet različitih tehnika za signalizaciju završetak ulazno/izlazneoperacije:

- 1 • Signalizacija objekta datoteke
- 2 • Signalizacija objekta događaja
- 3 • Poziv asinkronog procesa
- 4 • Ulazno-izlazni priključci za dovršetak
- 5 • Prozivanje

# Konfiguracije RAID-a u sustavu Windows

- Windows podržava dvije vrste RAID konfiguracija:
- 

## Hardverski RAID

Odvojeni fizički diskovi koje je kontroler za pohranu diska spojio u jedan ili više logičkih diskova

## Programski RAID

Nprekinuti prostor na disku kombiniran u jednu ili više logičkih particija pomoću upravljačkog programa diska otpornog na greške

# Volume Shadow Copies i Volume Encryption

- Volume Shadow Copies

- Učinkovit način izrade dosljednih snimaka volumena kako bi se mogli sigurnosno kopirati
- Također korisno za arhiviranje datoteka po volumenu
- Implementirao programski kroz upravljački program koji izrađuje kopije podataka na disku prije nego što je izbrisan

- Volume Encryption

- Windows koristi BitLocker za šifriranje cijelih jedinica
- Sigurnije od šifriranja pojedinačnih datoteka
- Omogućuje više međusobno povezanih slojeva sigurnosti

# Datoteke

- Zbirke podataka koje su stvorili korisnici
- Datotečni sustav je jedan od najvažnijih dijelova OS-a
- Poželjna svojstva datoteka:

## Dugoročno postojanje

- Datoteke se pohranjuju na disk ili drugu sekundarnu pohranu i ne nestaju kada se korisnik odjavi

## Djeljive između procesima

- Datoteke imaju nazive i mogu imati povezane dozvole pristupa koje dopuštaju kontrolirano dijeljenje

## Struktura

- Datoteke se mogu organizirati u hijerarhijsku strukturu kako bi odražavale odnose među datotekama

# Datotečni sustavi

- Pruža način za pohranu podataka organiziranih kao datoteke i zbirku funkcija koje se mogu izvršiti nad njima
- Održavanje skupa atributa pridruženih datoteci
- Uobičajene operacije uključuju:
  - Stvaranje
  - Brisanje
  - Otvaranje
  - Zatvaranje
  - Čitanje
  - Pisanje

# Struktura datoteke

Četiri pojma obično se koriste  
pri raspravi o datotekama:

Polje

Zapis

Datoteka

Baza  
podataka



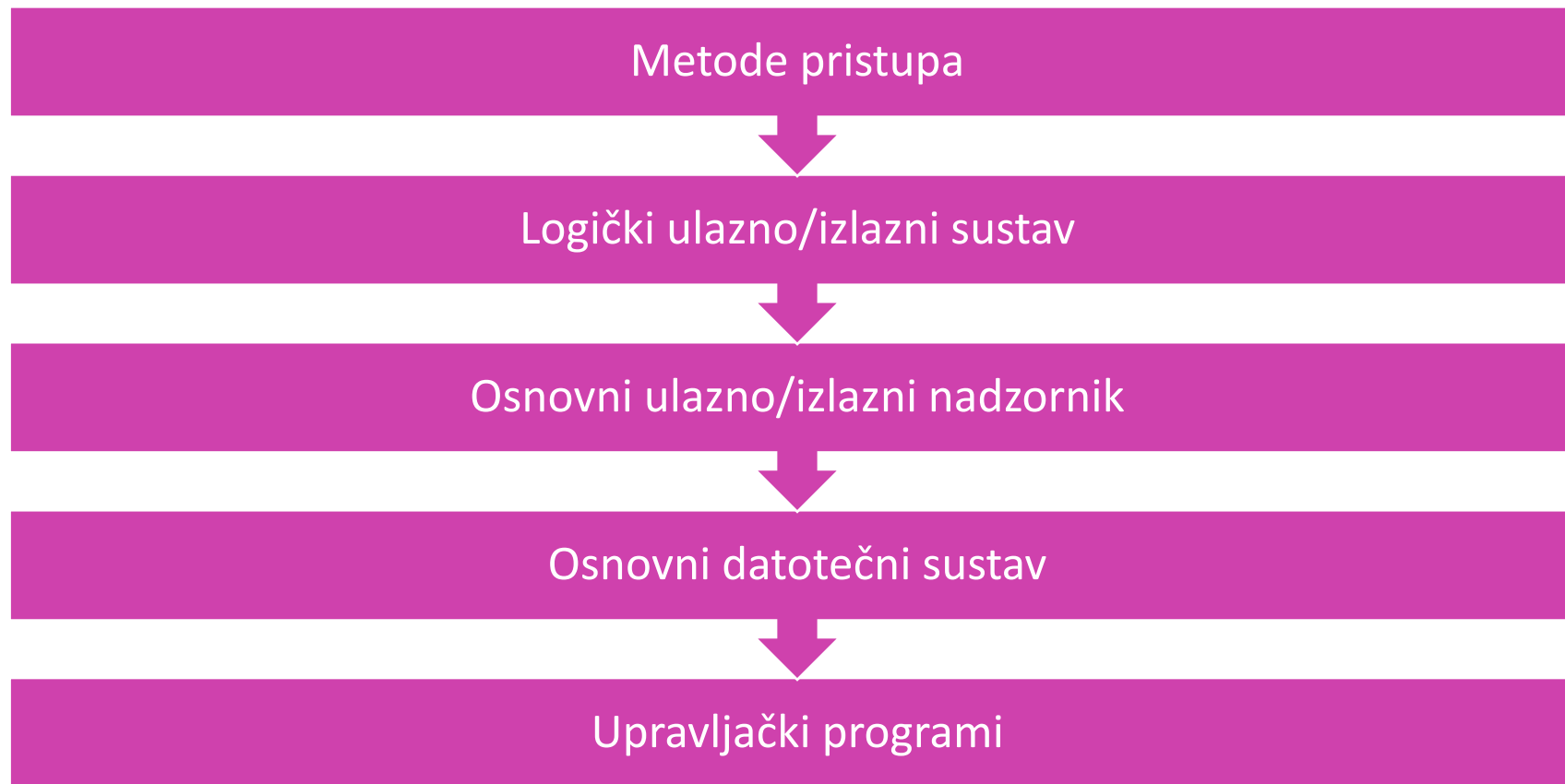
# Pojmovi strukture datotečnog sustava

- Polje
  - Osnovni element podataka
  - Sadrži jednu vrijednost
  - Fiksna ili promjenjiva duljina
- Baza podataka
  - Prikupljanje povezanih podataka
  - Odnosi među elementima podataka su eksplicitni
  - Dizajniran za upotrebu u brojnim različitim aplikacijama
  - Sastoji se od jedne ili više vrsta datoteka
- Datoteka
  - Prikupljanje sličnih zapisa
  - Tretira se kao jedan entitet
  - Može se pozivati po nazivu
  - Ograničenja kontrole pristupa obično se primjenjuju na razini datoteke
- Zapis
  - Zbirka povezanih polja koja neka aplikacija može tretirati kao jedinicu
  - Fiksna ili promjenjiva duljina

# Sustav upravljanja datotekama

- Zadovoljiti potrebe korisnika za upravljanjem podacima
- Jamstvo da su podaci u datoteci valjani
- Optimizirati performanse
- Pružanje ulazno-izlazne podrške za razne vrste uređaja za pohranu
- Minimizirati mogućnost gubitka podataka
- Pružanje standardiziranog skupa rutina ulazno-izlaznog sučelja korisničkim procesima
- Pružanje ulazno-izlazne podrške većem broju korisnika u višekorisničkim okruženjima

# Arhitektura datotečnog sustava



# Upravljački programi

- Najniža razina
- Izravno komunicira s perifernim uređajima
- Odgovoran za pokretanje ulazno-izlaznih operacija na uređaju
- Obraduje dovršetak ulazno-izlaznog zahtjeva
- Smatra se dijelom operativnog sustava

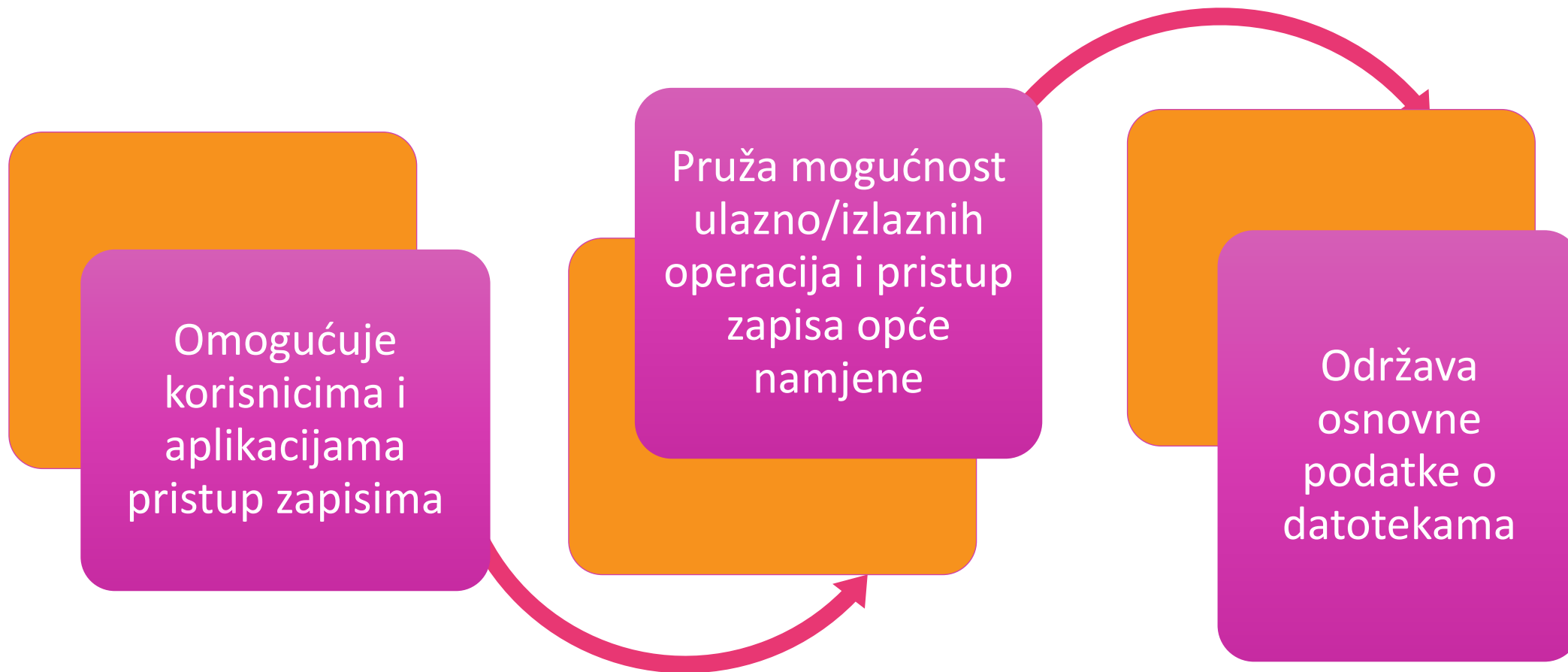
# Osnovni datotečni sustav

- Naziva se i fizička ulazno-izlazna razina
- Primarno sučelje s okruženjem izvan računalnog sustava
- Bavi se blokovima podataka koji se razmjenjuju s diskovnim sustavima
- Upravlja postavljanje blokova na sekundarni uređaj za pohranu
- Upravlja međuspremnikom blokova u glavnoj memoriji
- Ne razumije sadržaj podataka ili strukturu uključenih datoteka
- Smatra se dijelom operativnog sustava

# Osnovni ulazno/izlazni nadzornik

- Odgovoran za pokretanje i zatvaranje svih ulazno-izlaznih datoteka
- Na ovoj razini održavaju se kontrolne strukture koje se bave ulazno-izlaznim uređajima, planiranjem i statusom datoteka
- Odabire uređaj na kojem će se izvršiti ulazno/izlazna operacija
- Bavi se zakazivanjem pristupa disku radi optimizacije performansi
- Dodjeljuje ulazno-izlazne međuspremnik i sekundarnu memoriju
- Dio operativnog sustava

# Logički ulazno/izlazni sustav



# Metode pristupa

- Razina datotečnog sustava najbliža korisniku
- Pruža standardno sučelje između aplikacija i datotečnih sustava i uređaja koji sadrže podatke
- Različite metode pristupa odražavaju različite strukture datoteka i različite načine pristupa i obrade podataka

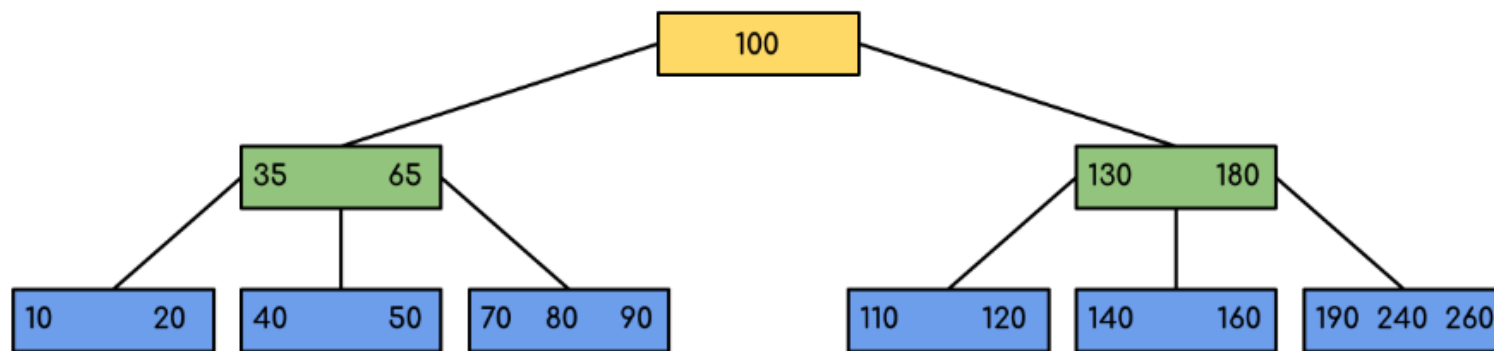


# Organizacija datoteka i pristup

- Organizacija datoteka logično je strukturiranje zapisa kako je određeno načinom na koji im se pristupa
- Pri odabiru organizacije datoteka važno je nekoliko kriterija:
  - Kratko vrijeme pristupa
  - Jednostavnost ažuriranja
  - Ekonomičnost skladištenja
  - Jednostavno održavanje
  - Pouzdanost
- Prioritet kriterija ovisi o aplikaciji koja će koristiti datoteku

# B-stabla

- Uravnotežena struktura stabla sa svim granama jednake duljine
- Standardna metoda organiziranja indeksa za baze podataka
- Obično se koristi u datotečnim sustavima OS-a
- Omogućuje učinkovito pretraživanje, dodavanje i brisanje stavki

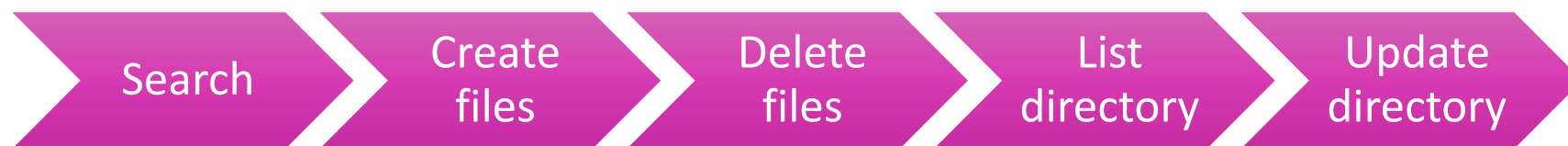


# Karakteristike B-stabla

- B-stablo karakterizira minimalni stupanj  $d$  i zadovoljava sljedeća svojstva:
  1. Svaki čvor ima najviše  $2d - 1$  ključa i  $2d$  djece ili, ekvivalentno,  $2d$  pokazivača
  2. Svaki čvor, osim korijena, ima najmanje  $d - 1$  ključ i  $d$  pokazivača, zbog čega je svaki unutarnji čvor, osim korijena, barem napola pun i ima najmanje  $d$  djece
  3. Korijen ima najmanje 1 ključ i dvoje djece
  4. Svi listovi pojavljuju se na istoj razini i ne sadrže nikakve informacije.
  5. Neleaf čvor s  $k$  pokazivačima sadrži  $k - 1$  ključ

# Operacije izvršene u imeniku

- Da biste razumjeli zahtjeve za strukturu datoteke, korisno je razmotriti vrste operacija koje se mogu izvoditi u direktoriju:



# Shema na dvije razine

Za svakog korisnika postoji jedan direktorij i glavni direktorij

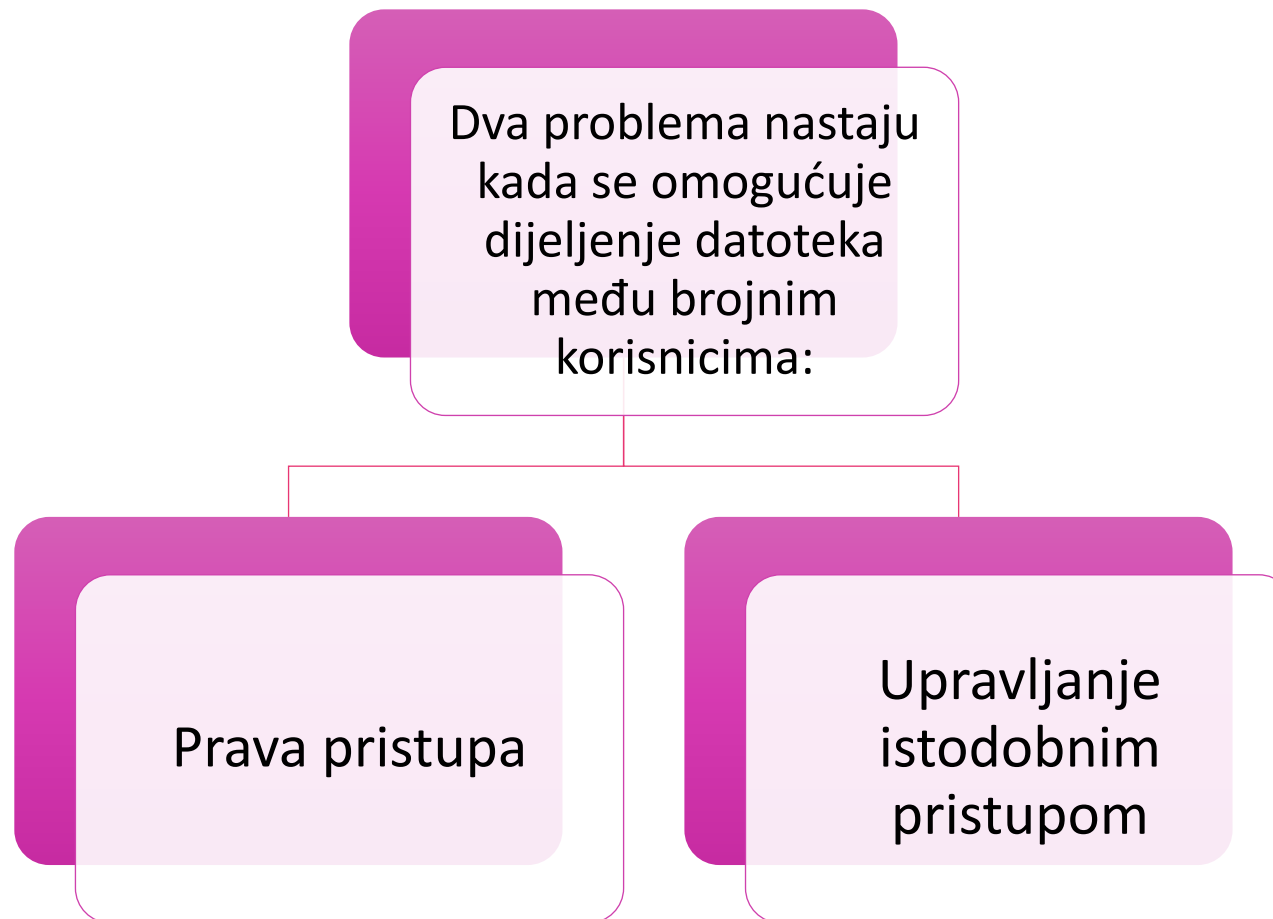
Glavni direktorij ima unos za svaki korisnički imenik koji pruža podatke o adresi i kontroli pristupa

Svaki korisnički direktorij jednostavan je popis datoteka tog korisnika

Imena moraju biti jedinstvena samo unutar zbirke datoteka jednog korisnika

Datotečni sustav može jednostavno nametnuti ograničenje pristupa direktorijima

# Zajedničko korištenje datoteka



# Prava pristupa

- Niko
  - Korisniku neće biti dopušteno čitanje korisničkog direktorija koji uključuje datoteku
- Čitanje metapodataka
  - Korisnik može utvrditi postoji li datoteka i tko je njezin vlasnik, a zatim može zatražiti od vlasnika dodatna prava pristupa
- Izvršavanje
  - Korisnik može učitati i izvršiti program, ali ga ne može kopirati
- Čitanje
  - Korisnik može pročitati datoteku u bilo koju svrhu, uključujući kopiranje i izvršavanje
- Dodavanje
  - Korisnik može dodati podatke u datoteku, ali ne može izmijeniti ili izbrisati sadržaj datoteke
- Ažuriranje
  - Korisnik može mijenjati, brisati i dodavati podatke datoteke
- Promjena zaštite
  - Korisnik može promijeniti prava pristupa dodijeljena drugim korisnicima
- Brisanje
  - Korisnik može izbrisati datoteku iz datotečnog sustava

# Prava korisničkog pristupa

## Vlasnik

Obično početni autor datoteke

Ima puna prava

Može dodijeliti prava drugima

## Određeni korisnici

Pojedinačni korisnici koji su određeni korisničkim ID-om

## Grupe korisnika

Skup korisnika koji nisu pojedinačno definiran

## Svi

Svi korisnici koji imaju pristup ovom sustavu

Ovo su javne datoteke



# Dodjela prostora datotekama

- Na sekundarnoj pohrani datoteka se sastoji od zbirke blokova
- Operacijski sustav ili sustav za upravljanje datotekama odgovoran je za dodjelu blokova datotekama
- Pristup koji se primjenjuje za dodjelu datoteka utječe na upravljanje slobodnim prostorom
- Prostor se dodjeljuje datoteci kao jedan ili više dijelova (susjedni skup dodijeljenih blokova)
- Tablica za dodjelu datoteka (FAT)
- Struktura podataka koja se koristi za praćenje dijelova dodijeljenih datoteci

# Veličina blokova

- Pri odabiru veličine blokova dolazi do kompromisa između učinkovitosti sa stajališta jedne datoteke u odnosu na ukupnu učinkovitost sustava
- Stavke koje treba uzeti u obzir:
  - Kontinuitet prostora povećava performanse, posebno za Retrieve\_Next operacije, a uvelike i za transakcije koje se odvijaju u operativnom sustavu orijentiranom na transakcije
  - Velikim brojem malih blokova povećava se veličina tablica potrebnih za upravljanje informacijama o dodjeli
  - Imajući dijelove fiksne veličine pojednostavljuje preraspodjelu prostora
  - S promjenjivom veličinom ili malim dijelovima fiksne veličine smanjuje se neiskorišteni dio prostora nastalog zbog alokacije

# Alternative

- Dvije glavne alternative:

## Promjenjivi, veliki susjedni dijelovi

- Pruža bolje performanse
- Promjenjiva veličina izbjegava gubitke prostora
- Tablice za dodjelu datoteka su male

## Blokovi

- Mali fiksni dijelovi pružaju veću fleksibilnost
- Zahtijevaju velike tablice ili složene strukture
- Blizina je napuštena kao primarni cilj
- Blokovi se dodjeljuju po potrebi

# Upravljanje slobodnim prostorom

- Kao što se mora upravljati dodijeljenim prostorom, tako mora i neraspoređeni prostor
- Za izvršavanje dodjele prostora datotekama potrebno je znati koji su blokovi dostupni
- Uz tablicu za dodjelu datoteka potrebna je i tablica raspodjele diska

# Bit tablice

- Ova metoda koristi vektor koji sadrži po jedan bit za svaki blok na disku
- Svaki unos 0 odgovara slobodnom bloku, a svaki 1 odgovara bloku koji se koristi

## Prednosti:

- Dobro funkcionira s bilo kojim načinom dodjele prostora datotekama
- Najmanje moguće prostora

# Lančani slobodni dijelovi

- Slobodni dijelovi mogu se vezati lancima pomoću pokazivača i vrijednosti duljine u svakom slobodnom dijelu
- Zanimarivi dodatni prostor jer nema potrebe za tablicom raspodjele diska
- Prilagođeno svim metodama dodjele prostora datotekama

## Nedostaci:

- Dovodi do fragmentacije
- Svaki put kada dodijelite blok, prvo morate pročitati blok da biste pronašli pokazivač na novi prvi slobodni blok prije nego što napišete podatke u taj blok

# Indeksiranje

- Tretira slobodan prostor kao datoteku i koristi tablicu indeksa kao i za dodjelu datoteka
- Za učinkovitost, indeks bi se trebao temeljiti na dijelovima promjenjive veličine, a ne na blokovima
- Ovaj pristup pruža učinkovitu podršku za sve metode dodjele datoteka

# Popis praznih blokova

Svakom bloku dodjeljuje se broj uzastopno

Popis brojeva svih slobodnih blokova održava se u rezerviranom dijelu diska

Ovisno o veličini diska, za pohranu broja jednog bloka bit će potrebno 24 ili 32 bita

Veličina praznog popisa blokova je 24 ili 32 puta veća od odgovarajuće tablice bitova i mora se pohraniti na disk

Postoje dvije učinkovite tehnike za spremanje malog dijela praznih blokova u glavnu memoriju:

Popis se može tretirati kao push-down stog s prvih nekoliko tisuća elemenata koji se čuvaju u glavnoj memoriji

Popis se može tretirati kao FIFO red čekanja, s nekoliko tisuća unosa s početka i kraja reda čekanja u glavnoj memoriji



# Volumeni

- Zbirka adresabilnih sektora u sekundarnoj memoriji koje OS ili aplikacija mogu koristiti za pohranu podataka
- Sektori u obujmu ne moraju biti uzastopni na fizičkom uređaju za pohranu
- Oni se samo tako trebaju pojaviti OS-u ili aplikaciji
- Volumen može biti rezultat sastavljanja i spajanja manjih volumena

# UNIX upravljanje datotekama

- U datotečnom sustavu UNIX razlikuje se šest vrsta datoteka:

## Regularne - obične

- Sadrži proizvoljne podatke u blokovima podataka

## Direktorij

- Sadrži popis naziva datoteka i pokazivača na pridružene inode (čvorovi indeksa)

## Specijalne

- Ne sadrži podatke, ali pruža mehanizam za mapiranje fizičkih uređaja u nazive datoteka

## Imenovane cijevi

- Međuprocesni komunikacijski objekt

## Linkovi

- Alternativni naziv datoteke za postojeću datoteku

## Simboličke veze

- Podatkovna datoteka koja sadrži naziv datoteke s kojom je povezana

# Inod - čvor indeksa

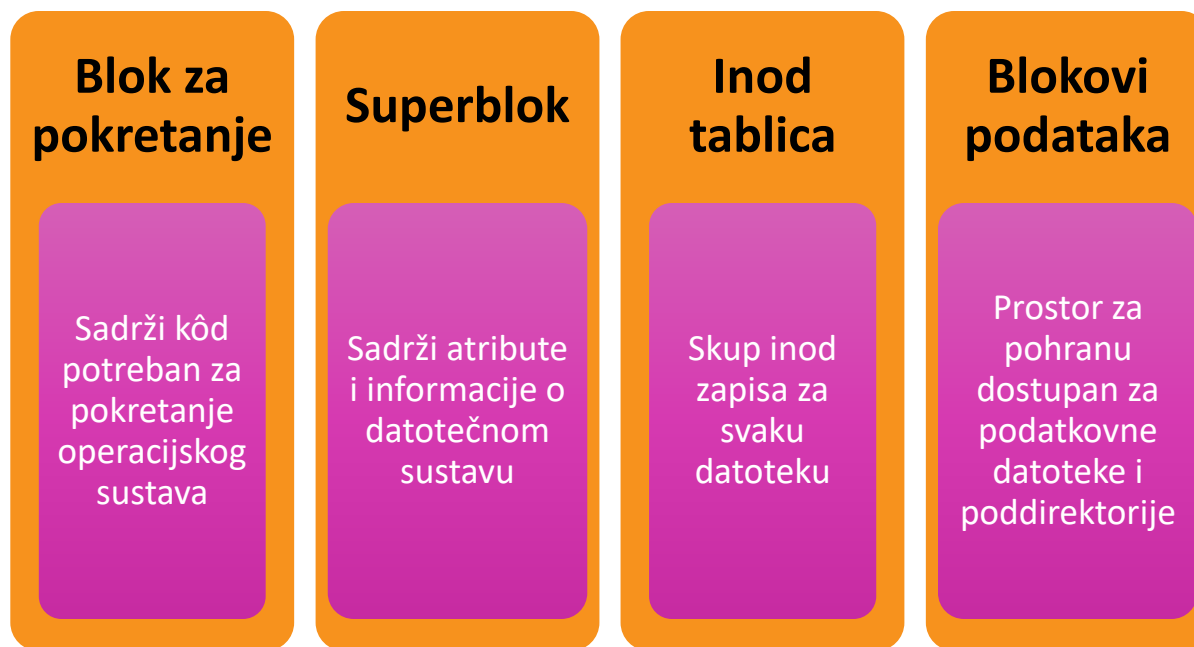
- Svim vrstama UNIX datoteka upravlja OS pomoću inod
- Inod (čvor indeksa) je kontrolna struktura koja sadrži ključne informacije potrebne operacijskom sustavu za određenu datoteku
- Nekoliko naziva datoteka može biti povezano s jednim inodom
  - Aktivni inod povezana je s točno jednom datotekom
  - Svaku datoteku kontrolira točno jedan inode

# Dodjela datoteka

- Dodjela datoteka vrši se na temelju bloka
- Dodjela je po potrebi dinamična, umjesto da se koristi predalokacija
- Za praćenje svake datoteke koristi se indeksirana metoda, a dio indeksa pohranjen je u inodu za datoteku
- U svim implementacijama UNIX-a inod uključuje niz izravnih pokazivača i tri neizravna pokazivača (jednostruki, dvostruki, trostruki)

# Struktura volumena

- Datotečni sustav UNIX nalazi se na jednom logičkom disku ili particiji diska i postavljen je sa sljedećim elementima:



# Datotečni sustav Windows

- Programeri sustava Windows NT dizajnirali su novi datotečni sustav, Datotečni sustav Nove tehnologije (NTFS) koji je namijenjen ispunjavanju vrhunskih zahtjeva za radne stanice i poslužitelje
- Ključne značajke NTFS-a:
  - Oporavivost
  - Sigurnost
  - Veliki diskovi i velike datoteke
  - Više tokova podataka
  - Journaling
  - Kompresija i šifriranje
  - Tvrde i simboličke veze

# NTFS volumen i struktura datoteke

- NTFS koristi sljedeće koncepte pohrane diska:

## Sektor

- Najmanja jedinica fizičke pohrane na disku
- Veličina podataka u bajtovima potencija je broja 2 i gotovo uvijek je 512

## Klaster

- Jedan ili više susjednih sektora
- Veličina klastera u sektorima je potencija broja 2

## Volumen

- Logička particija na disku, koja se sastoji od jednog ili više klastera i koju datotečni sustav koristi za dodjelu prostora
- Može biti cijeli ili dio jednog diska ili se može proširiti na više diskova
- Maksimalna veličina volumena za NTFS je  $2^{64}$  klastera

# Glavna tablica datoteka (MFT)

- Srce datotečnog sustava Windows je MFT
- MFT je organiziran kao tablica redaka od 1.024 bajta, nazvana zapisi
- Svaki redak opisuje datoteku na ovoj jedinici, uključujući sam MFT, koji se tretira kao datoteka
- Svaki zapis u MFT-u sastoji se od skupa atributa koji služe za definiranje karakteristika datoteke (ili mape) i sadržaja datoteke



# SQLite

- Najčešće implementirani modul SQL baze podataka na svijetu
- Na temelju strukturiranog jezika upita (SQL)
- Dizajniran za pružanje pojednostavljenog sustava upravljanja bazama podataka temeljenog na SQL-u prikladnog za ugrađene sustave i druge ograničene memorijske sustave
- SQLite biblioteka može se implementirati u manje od 400 KB
- Za razliku od drugih sustava za upravljanje bazama podataka, SQLite nije zaseban proces kojem se pristupa iz klijentske aplikacije
  - Biblioteka SQLite povezana je i time postaje sastavni dio aplikacijskog programa

# Summary

- Ulazno-izlazni uređaji
- Organizacija funkcije ulazno/izlazne funkcije
  - Evolucija I/O funkcije
  - Izravan pristup memoriji
- Problemi s dizajnom operacijskog sustava
  - Ciljevi dizajna
  - Logička struktura ulazno-izlazne funkcije
- Ulazno/izlazni međuspremnik
  - Jednostruki/dvostruki/kružni međuspremnik
  - Korisnost međuspremnika
- Planiranje diska
  - Parametri performansi diska
  - Pravila zakazivanja diska
- RAID
- Predmemorija diska
- UNIX SVR4 I/O
- Linux I/O
- Windows ulazno/izlazno
- Struktura datoteke
- Sustavi za upravljanje datotekama
- B-stabla
- Direktoriji datoteka
  - Sadržaj
  - Struktura
  - Imenovanja
- Zajedničko korištenje datoteka
- Prava pristupa
- Simultani pristup
- UNIX upravljanje datotekama
- Linux virtualni datotečni sustav
- Datotečni sustav Windows
- SQLite



**Thank you for  
your attention!**