

SADRŽAJ

9.1 Kriterijumi za raspoređivanje zahteva za rad sa Hard Disk-om

9.2 Algoritmi za raspoređivanje zahteva

FCFS, SSTF, SCAN, C-SCAN

LOOK, C-LOOK

9.3 RAID strukture

9.4 Napredne strukture

9.1-Kriterijumi za raspoređivanje zahteva

- Operativni sistem je odgovoran **za efikasno korišćenje hardvera**.
- HD predstavljaju **usko grlo računarskog sistema** po pitanju performansi
- **Ukupno vreme** predstavlja period koji protekne između prvog zahteva i kompletiranja poslednjeg transfera podata i predstavlja zbir:
Ukupno vreme = vreme pristupa + vreme prenošenja podataka.
- **Vreme pristupa** HD (*Access time*) sastoji se od 2 komponente:
 1. **Vreme pozicioniranja** (*Seek time*) – vreme potrebno za postavljanje glave za upis i čitanje na određeni cilindar,
 2. **Vreme rotacionog kašnjenje** (*Rotational latency*) – dodatno vreme za čekanje da odgovarajući sektor dođe pod glavu za upis i čitanje.
- U jednom trenutku postoji **više zahteva za rad sa HD**
- Ukupno vreme **može se skratiti raspoređivanjem** ovih zahteva
- Postoje algoritmi **koji minimizuju obe mehaničke komponente** – vreme pozicioniranja i vreme rotacionog kašnjenja i uzimaju u obzir i keširanje na samom disk uređaju.

Jednostavnosti radi, analiziraćemo samo algoritme koji minimizuju vreme pozicioniranja jer oni predstavljaju osnov za sve ostale algoritme

9.2- Algoritmi za pozicioniranje glave na HD

1. **FCFS** (*First Come, First Served*) je najjednostavniji algoritam, koji zahteve prosleđuje u onom redosledu kako su stigli. FCFS obezbeđuje krajnje fer odnose prema prispelim zahtevima, ali i loše performanse.
2. **SSTF** (*Shortest Seek Time First*) izvršava zahteve tako što od prispelih zahteva prvo uzima onaj koji izazva najmanji pomeraj glava (*seek time*)
3. **SCAN** naizmenično pomera glave **od početka do kraja diska i nazad**, i na tom putu izvršava zahteve koji se nalaze na tekućem cilindru.
4. **C-SCAN** (*Circular SCAN*) je varijanta SCAN algoritma koja rešava problem favorizovanja unutrašnjih cilindara jer se **samo u jednom smeru izvršavaju zahtevi**. Kada glave dođu do poslednjeg cilindra, pomeraju se na početak, ne izvršavajući zahteve na tom putu.
5. **LOOK** je modifikacija SCAN algoritma jer se glave **ne pomeraju do kraja ili početka diska**, nego do poslednjeg zahteva koji se nalazi u redu čekanja u smeru kretanja glave HD – **dvosmerno izvršavanje**.
6. **C-LOOK** je modifikacija C-SCAN algoritma. C-LOOK izvršava zahteve **samo u rastućem smeru do poslednjeg zahteva u redu**, nakon toga se vraća na zahtev najbliži početku diska-**jednosmerno izvršavanje**

9.2- Algoritmi za pozicioniranje glave na HD

PRIMER:

Neka se u redu čekanja nalaze sledeći zahtevi za rad sa diskom koji su došli u FIFO poretku:

{98, 183, 37, 122, 14, 124, 65, 67}

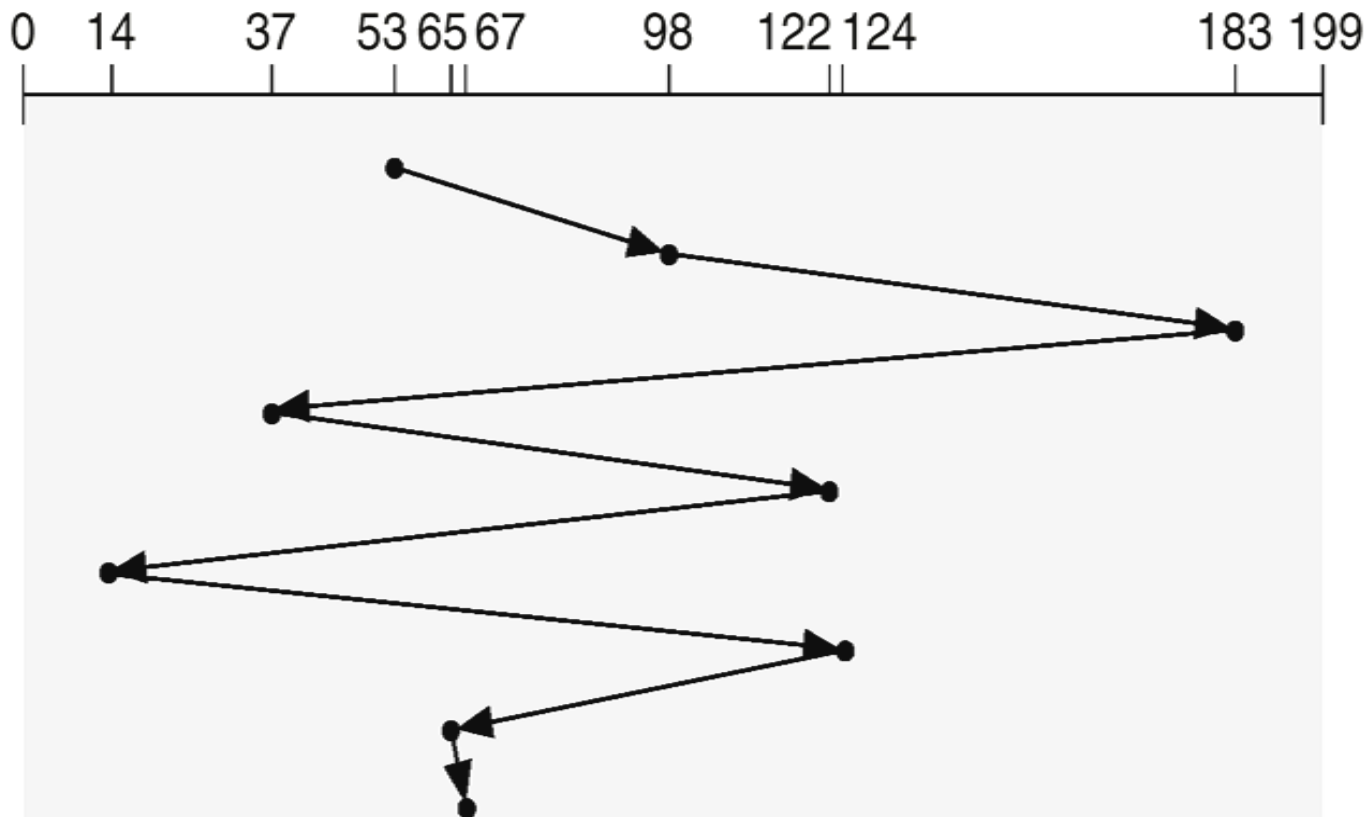
Pretpostavimo da disk ima 200 cilindara (0-199) i da se glave za upis i čitanje trenutno nalaze na cilindru 53, a prethodno su bile na cilindru 100.

9.2 - FCFS (First Come - First Served)

- Ovaj algoritam zahteve prosleđuje **po redosledu pristizanja zahteva**.
- Brojevi u zagradi predstavljaju **pomeraj koji glave diska prave pri prelasku na sledeći cilindar**, izražen u broju cilindara.

Trenutna pozicija glave je na cilindru 53

U redu čekanja nalaze se zahtevi: 98, 183, 37, 122, 14, 124, 65, 67



98	(45)
183	(85)
37	(146)
122	(85)
14	(108)
124	(110)
65	(59)
67	(2)

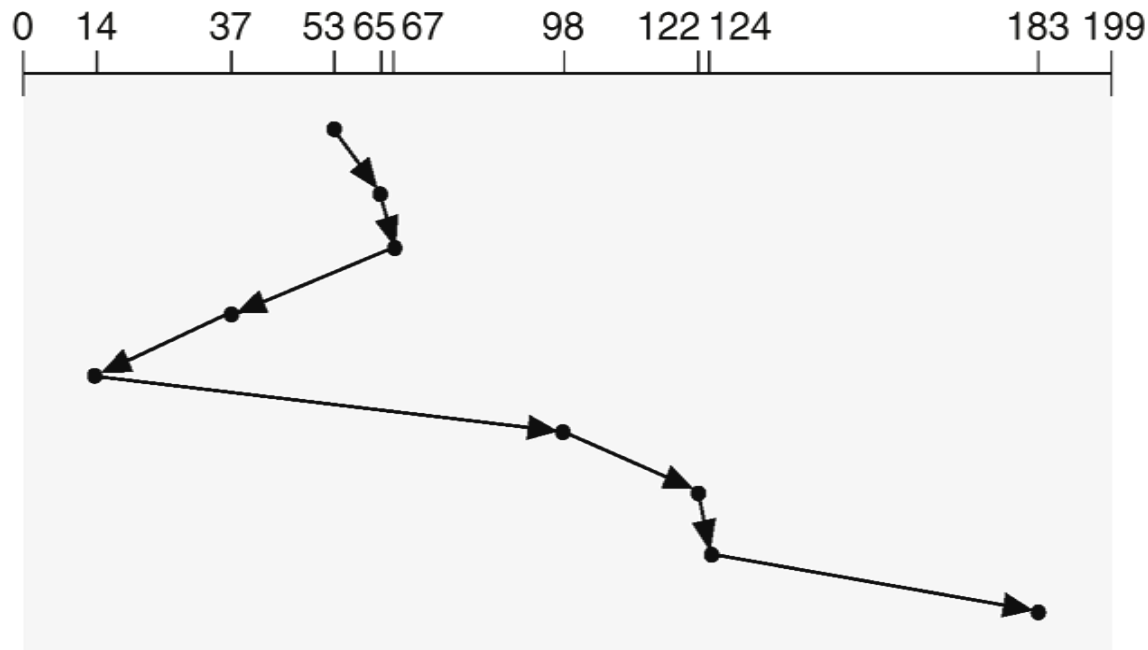
Ukupno **640**

9.2 - SSTF (Shortest Seek Time First)

- SSTF izvršava zahtev **sa najmanjim vremenom traženja od tekuće pozicije glave** i ovo raspoređivanje predstavlja vrstu SJF raspoređivanja
- Može dovesti do “**izgladnjivanja**” (*starvation*) nekih zahteva.
- Glave mogu ostati veoma dugo u jednoj zoni **opslužujući zahteve koji unose male pomeraje**, pa zahtevi čiji su cilindri daleko od tekuće

Trenutna pozicija glave je na cilindru 53

U redu čekanja nalaze se zahtevi: 98, 183, 37, 122, 14, 124, 65, 67



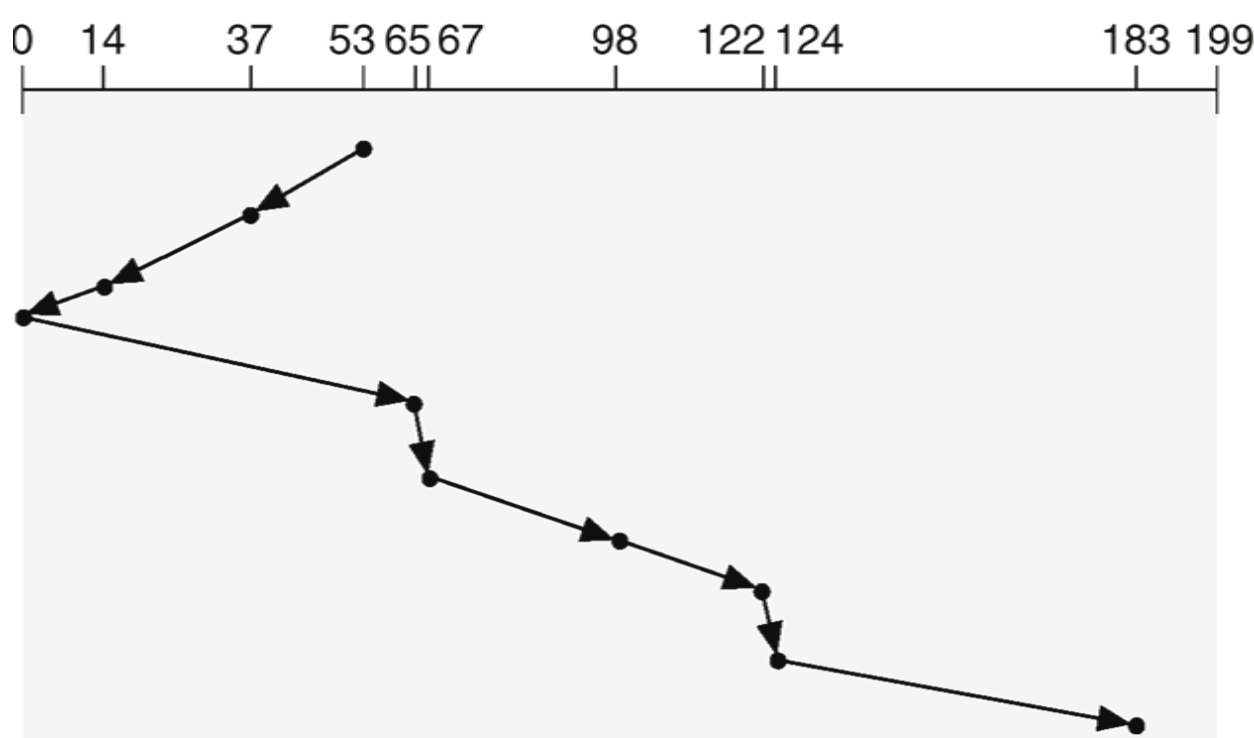
65	(12)
67	(2)
37	(30)
14	(23)
98	(84)
122	(24)
124	(2)
183	(59)
Ukupno	- <u>236</u>

9.2 - SCAN (Elevator Algorithm)

- Glava diska startuje sa jednog kraja diska i pomera se ka drugom kraju servisirajući redom zahteve.
- Sa drugog kraja diska kretanje glave vrši se u suprotnom smeru i nastavlja se sa servisiranjem zahteva.

Trenutna pozicija glave je na cilindru 53

U redu čekanja nalaze se zahtevi: 98, 183, 37, 122, 14, 124, 65, 67



37	(16)
14	(23)
0	(14)
65	(65)
67	(2)
98	(31)
122	(24)
124	(2)
183	(59)

Ukupno - **236**

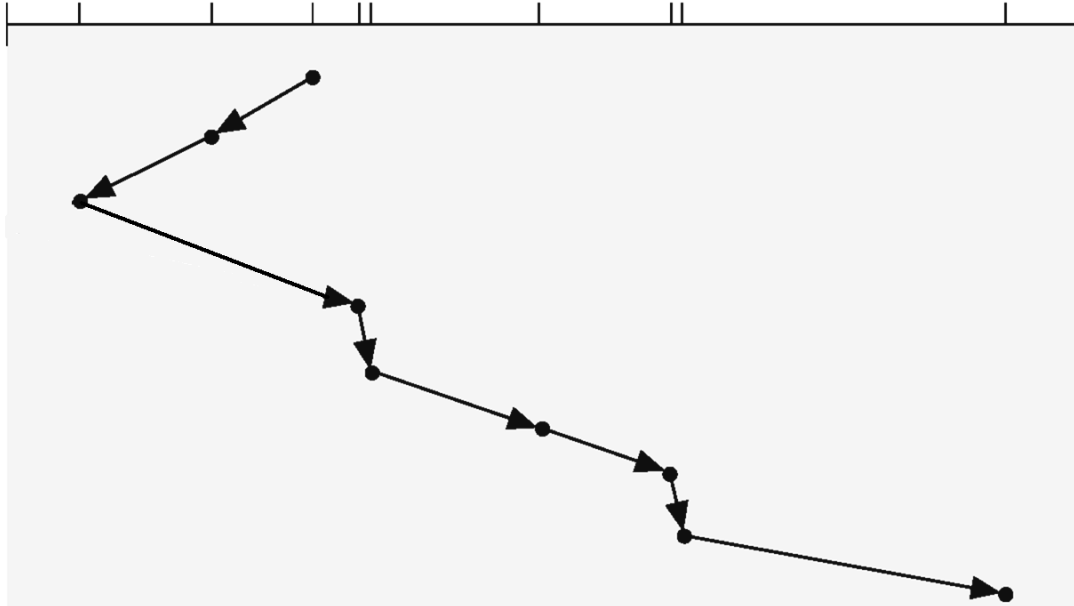
9.2 - LOOK

- Algoritam LOOK predstavlja **modifikaciju algoritma SCAN**.
- Glave se ne pomeraju do kraja ili početka diska nego **do poslednjeg zahteva** koji se nalazi u redu čekanja na tom smeru.
- Pri tome LOOK **opslužuje zahteve u oba smeru**.
- U našem primeru imamo pomeranje za 208 cilindara.

Trenutna pozicija glave je na cilindru 53

U redu čekanja nalaze se zahtevi: 98, 183, 37, 122, 14, 124, 65, 67

0 14 37 53 65 67 98 122 124 183 199



37 (16)

14 (23)

65 (51)

67 (2)

98 (31)

122 (24)

124 (2)

183 (59)

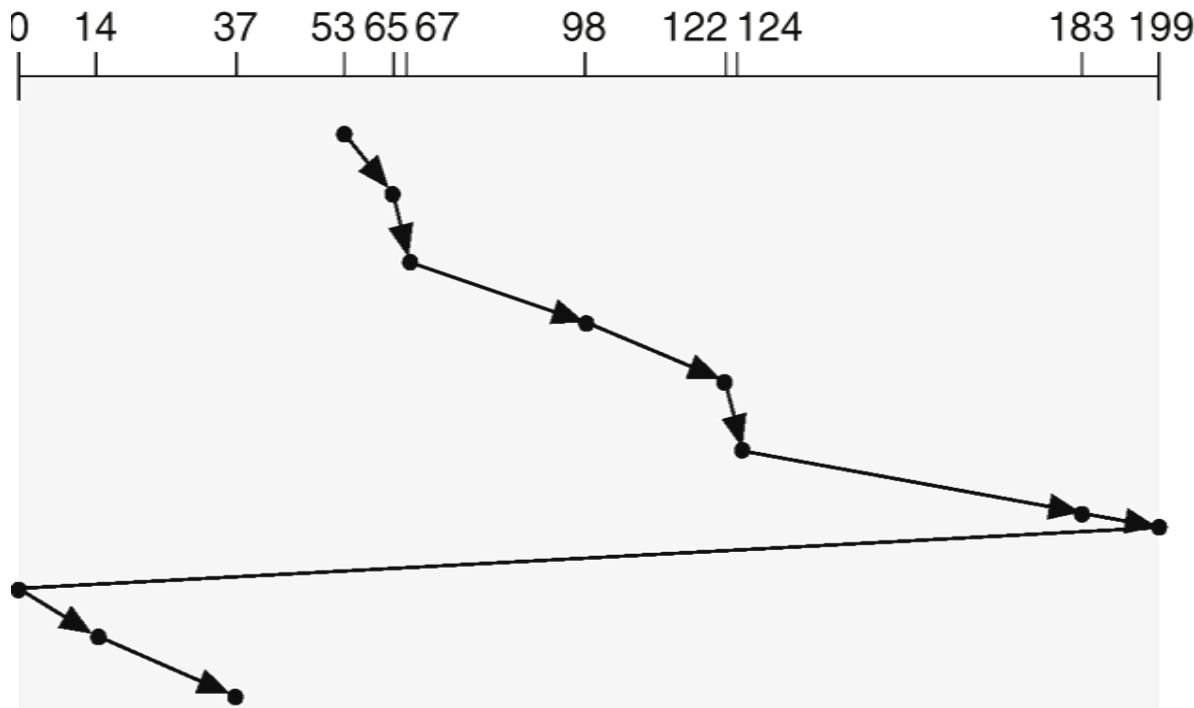
Ukupno - **208**

9.2 - C-SCAN (Circular SCAN)

- Algoritam C-SCAN omogućava uniformnije vreme čekanja od SCAN
- Glava se pomera sa jednog na drugi kraj diska uz servisiranje zahteva
- Kada stigne do kraja, vraća se na početak diska bez servisiranja zahteva
- Cilindri se tretiraju kao kružna lista, tj. posle poslednjeg ponovo se prelazi na prvi cilindar.

Trenutna pozicija glave je na cilindru 53

U redu čekanja nalaze se zahtevi: 98, 183, 37, 122, 14, 124, 65, 67



65	(12)
67	(2)
98	(31)
122	(24)
124	(2)
183	(59)
199	(16)
0	(199)
14	(14)
37	(23)

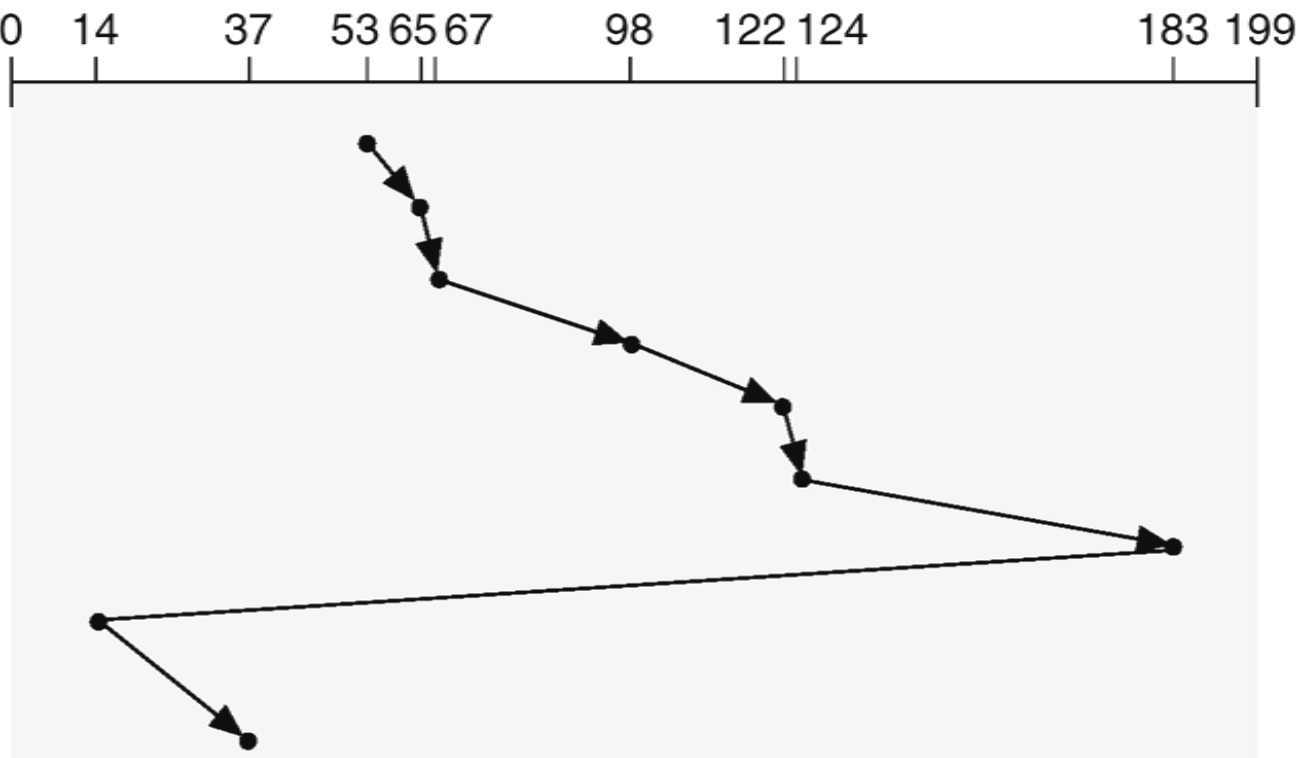
Ukupno - **382**

9.2 - C-LOOK

➤ Algoritam C-LOOK podrazumeva da **glava ide jedino do poslednjeg zahteva**, a zatim po potrebi menja smer kretanja bez prethodnog odlaska do kraja diska.

Trenutna pozicija glave je na cilindru 53

U redu čekanja nalaze se zahtevi: 98, 183, 37, 122, 14, 124, 65, 67



65	(12)
67	(2)
98	(31)
122	(24)
124	(2)
183	(59)
14	(169)
37	(23)

Ukupno - **322**

9.2- Algoritmi za pozicioniranje glave na HD

- Performanse **zavise od ukupnog pomeranja glava** za čitanje i pisanje pri opsluživanju zahteva iz reda i one **slabe sa većim pomeranjima**.
- **Relativne performanse** izrazili smo ukupnim brojem cilindara koje glave za čitanje i pisanje prelaze pri opsluživanju zahteva.
- **FCFS** je najjednostavniji algoritam koji je krajnje fer prema prispelim zahtevima, ali daje znatno lošije performanse (**640** pomeraja glave).
- **SSTF**-optimalan u pogledu vremena pozicioniranja ali nastaje problem zakucavanja (*starvation*). Glave mogu ostati veoma dugo u jednoj zoni, opslužujući zahteve koji unose male pomeraje. (**236**)
- **SCAN** - rešava problem zakucavanja. Prilikom obrade zahteva, SCAN daje prednost unutrašnjim cilindrima u odnosu na periferne. (**236**)
- **LOOK** – u praksi uvek menja SCAN algoritam (**208**)
- **C-SCAN** ima mnogo bolju raspodelu opsluživanja i ne izaziva problem zakucavanja, koji postoji pri radu sa algoritmom SSTF (**382**)
- **C-LOOK** je najbolje rešenje za jako opterećene sisteme. (**322**)

SCAN, C-SCAN i C-LOOK za jako opterećene sisteme

SSTF i C-LOOK razuman izbor za default algoritam

9.2- Algoritmi za pozicioniranje glave na HD

- Prethodno pomenuti algoritmi su **zastareli**
- Modernije varijante ovih algoritama **minimizuju obe mehaničke komponente** - i **pozicioniranje** i **rotaciono kašnjenje** (pri čemu rotaciono kašnjenje ima dominantan uticaj na performanse savremenih hard diskova)
- Jedan takav algoritam je **SATF** (*Shortest Access Time First*)
- On radi na **principu algoritma SSTF**, ali pri izboru sledećeg zahteva iz reda, on u obzir uzima obe mehaničke komponente.
- Najsavremeniji algoritmi uzimaju **u obzir i keširanje** na samom disku.
- Algoritam C-LOOK, **u kombinaciji sa ugrađenim keširanjem diska**, daje najbolje rezultate.

9.3 - RAID strukture

- Koliko god da su savremeni diskovi brzi i kvalitetni, po performansama daleko zaostaju za procesorima - **brzina i pouzdanost**.
- Jedna od tehnika kojima se poboljšavaju performanse sekundarne memorije podrazumeva korišćenje više diskova koji rade kooperativno, tj. **grupa diskova koji se koriste kao jedan uređaj** (*Disk striping*).
- Postojanje više diskova na istom računaru dozvoljava da se oni iskoriste, **ne samo zbog povećanja prostora**, već i za:
 1. **povećanje pouzdanosti**, zbog postojanja **redundantnih diskova**
 2. **poboljšanje performansi**, zbog mogućeg **paralelizma u radu**
- Različite tehnike organizacije sekundarne memorije koje **upotrebom više (jeftinih) diskova umesto jednog skupog** ostvaruju gornje ciljeve, nazivaju se **RAID-Redundant Arrays of Inexpensive/Independent Disks**
- RAID je tehnologija koja je **ušla u upotrebu 1987** godine i u početku je razvijena isključivo za potrebe serverskih sistema.
- Ona je razvijena zbog potreba **većih performansi i veće sigurnosti**, bez velikih troškova (zato se nalazi *inexpensive* u nazivu).

9.3 - RAID strukture

RAID struktura može da se ostvari putem:

1. **softverskog RAID**—diskovi se povezuju na standardnu U/I magistralu na uobičajeni način a OS implementira RAID funkcionalnost
2. **hardverskog RAID** - preko posebnog hardverskog kontrolera koji implementira RAID funkcije, tako da je RAID transparentan za OS
 - **Softverski RAID kontroleri** pripremaju podatke o tome kako bi taj niz trebao da funkcioniše, pa ih prosleđuju procesoru (CPU).
 - Procesor **radi kalkulaciju dobijenih podataka**: na koliko delova se deli fajl, veličinu trake (*stripe*), gde koji deo fajla treba da ide i vraća podatke kontroleru koji ih nakon toga šalje na prethodno određena mesta
 - Izračunavanje podataka potrebnih za RAID 0 i 1 nizove **nisu mnogo procesorski zahtevne operacije**, pa su oni ovde primenjivi.
 - **RAID 5 je procesorski zahtevan**, pa ako bi kalkulacije njegovih podataka prepustili CPU, ne bi smo mogli da koristimo računar.
 - Tu na scenu dolaze **hardverski RAID kontroleri** koji na sebi imaju, **specijalizovani procesor** koji isključivo radi kalkulacije potrebne za RAID niz, kao i **dodatnu keš memoriju**.

9.3 - RAID strukture - performanse

- Ako bi se N diskova koristilo samo za povećanje prostora N puta, bez redundanse podataka, **onda se verovatnoća otkaza sistema povećava N puta** u odnosu na verovatnoću otkaza jednog diska
- Višestruke diskove **koristimo za smeštanje redundantnih podataka**, tako da u slučaju otkaza nekog diska ne dolazi **do gubitka informacija**
- **Ogledanje** (*mirroring*): jedan logički disk sastoji se od **dva fizička diska sa identičnim kopijama**; svaki upis je dupliran i vrši se identično na oba diska; ako jedan otkáže drugi je raspoloživ do popravke/zamene.
- **Tehnika pruga** (*data striping*) – podaci se sukcesivno smeštaju:
 - 1. bit-level striping**: 1 bit svakog bajta raspoređuje se na 8 diskova
 - 2. block-level striping**: susedni blokovi fajla su raspoređeni na susedne diskove - **najčešće primenjivana tehnika**
- Sa N diskova, **povećava se prostor N puta i poboljšavaju performanse**:
 - ✓ za više pristupa malim delovima, **povećava se propusna moć**, jer se više zahteva može paralelizovati na više diskova
 - ✓ za jedan pristup većem delu, **smanjuje se vreme odziva** jer se deli pristup na više diskova koji rade paralelno

9.3 - Veličina trake kod RAID-a

- Veličina trake (*stripe size*) je **izuzetno bitna** za veće performanse RAID
- Trake (*stripes*) su **blokovi jednog fajla koji su razbijeni u manje delove**
- **Veličinu** u koju se podaci razbijaju (*stripe size*) **definiše korisnik** u rasponu od 1KB do 1024KB pa i više.
- Princip rada je takav da **RAID kontroler deli podatke u blokove** (jedan ili više u zavisnosti od veličine trake i fajla) i zatim ih distribuira na diskove u nizu (različiti delovi fajla idu na različite diskove).
- Podatak se **brže upisuje** zato što hard disk upisuje manje fajlove, koji su u stvari deo jednog većeg fajla. **Čitanje je takode brže**, jer podaci se preuzimaju istovremeno sa HD u nizu, a onda se sklapaju u kontroleru

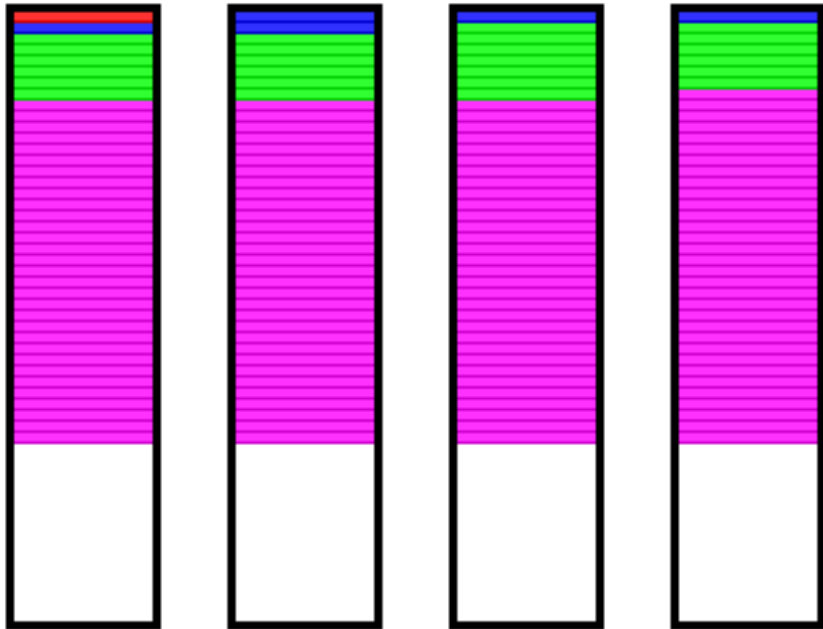
Koja je najbolja veličina trake za najbolje performanse?

Primer: *Potrebno je smestiti 2 fajla (128 i 4096 kB) na RAID 0 (min. potrebno dva diska) a) traka je 64kB i b) traka je 1024kB.*

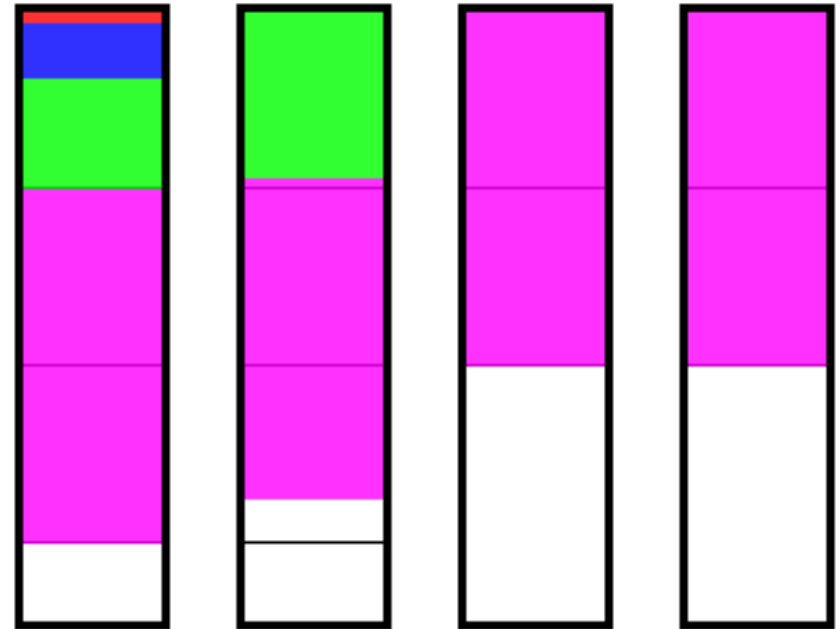
- za fajl od 128 kB **ima ubrzanja** (duplo brže) a za fajl od 4096 kB ne (64 puta pristupa diskovima – **znatno usporenje**)
- za fajl od 128 kB **nema promene** (traka je veća od fajla) dok za fajl od 4096 kB imamo **znatno ubrzanje** (dva pristupa po disku)

9.3-Upoređivanje veličina trake kod RAID

Veličina trake 4 kiB



Veličina trake 64 kiB



Posmatramo 4 fajla različite veličine:

1. 4 kiB – crvena,
2. 20 kiB – plava,
3. 100 kiB – zelena
4. 500 kiB – magenta

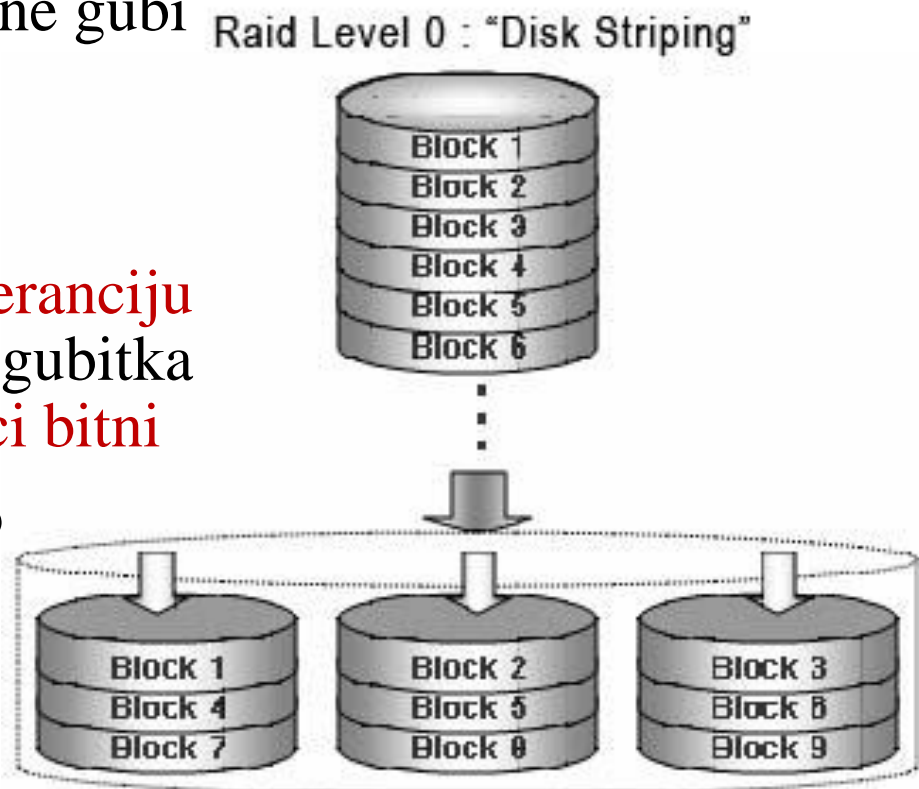
9.3 - RAID 0 (*Striping*)

- RAID 0 ili *Striped Set without Parity* predstavlja konfiguraciju u kojoj je **traka na nivou jednog ili više blokova** podataka (*block striping*).
- Veoma **brza konfiguracija**, ima najbolje performanse čitanja i upisa, ali **bez ikakve redundanse**, otkaz ma kog diska znači gubitak svih podataka
- U suštini to i **nije validan RAID** niz, jer nema toleranciju greške.
- Pogodno rešenje za zahtevne aplikacije kod kojih **pouzdanost nije bitna**

Prednosti: nema pravljenja parnosti (ne gubi se na brzini usled toga), **lak za implementaciju, jeftin, maksimalno iskorišćenje mesta** na diskovima

Mane: nije pravi RAID niz, **nema toleranciju greške**, otkaz jednog diska dovodi do gubitka podataka, **loš za servere gde su podaci bitni**

Upotreba: priprema za štampu, video montaža i produkcija, graficki dizajn, svi koji **žele veće performanse**, a rade redovan **back-up**



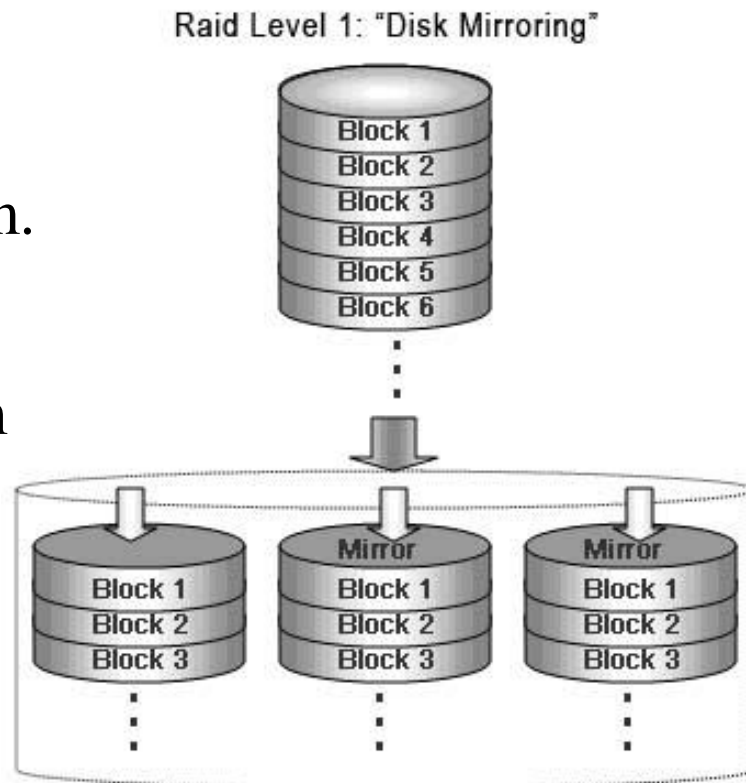
9.3 - RAID 1 (*Mirroring*)

- Svaki disk **ima svoje ogledalo** (*mirroring*) bez ikakve paralelizacije
- **Jeftina, jednostavna** konfiguracija za servere koji zahtevaju pouzdanost
- Sadržaj jednog diska **ima identičnu kopiju na drugom disku** u nizu.
- Koristimo ga kada nam je **važnost podataka daleko važnija od brzine**
- RAID 1 je možda **sporiji u upisu podataka**, čak i od jednog diska
- Dobra osobina je što u slučaju otkaza jednog diska **ne gubimo podatke**, a takođe **ni vreme za zamenu diska**.

Prednosti: nema generisanja parnosti, laka implementacija, **visoka tolerancija grešaka**, koristi **pun kapacitet diska**, 2 diska minimum.

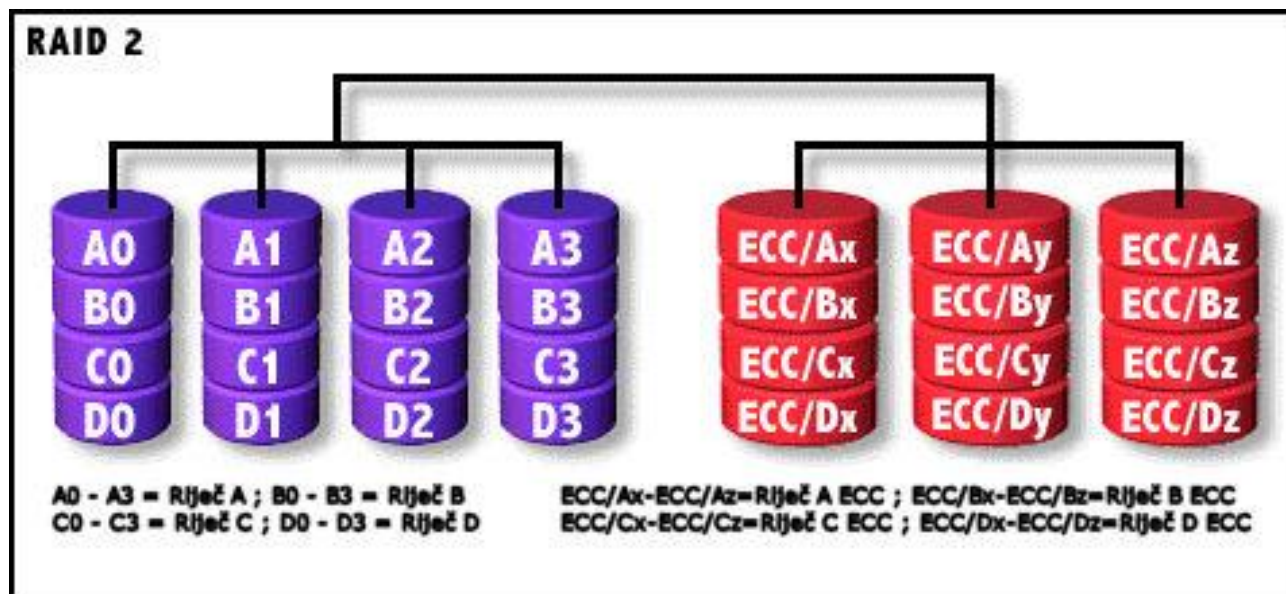
Mane: **nekorisno iskorišćavanje mesta** na diskovima, **visoki troškovi** po upotrebljivom GB, **duplo veći broj upisa**.

Upotreba: web serveri, knjigovodstvo, banke, svuda gde je sigurnost podataka na prvom mestu



9.3 - RAID 2

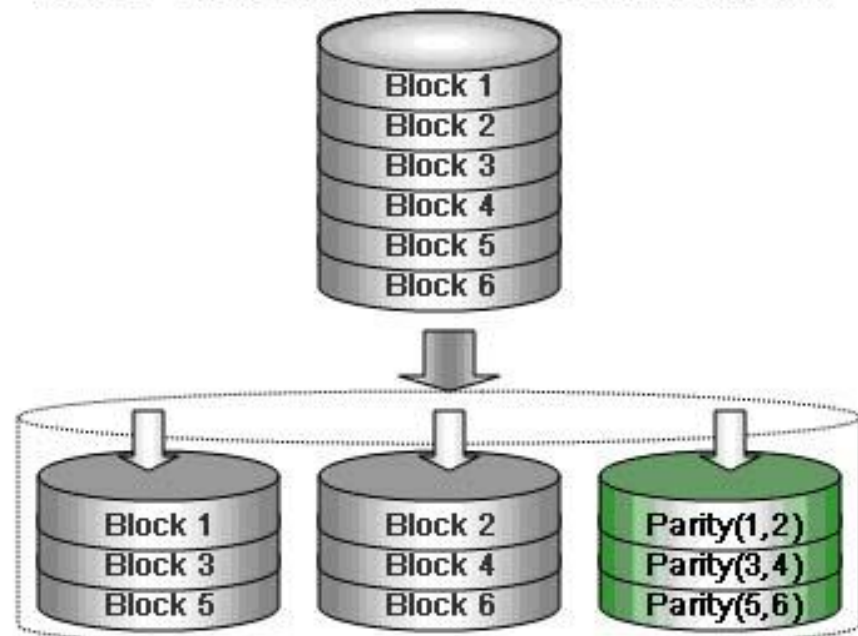
- Poznata je pod nazivom **RAID sa memorijskim stilom korekcije** (*memory-style error correcting code organization*)
- Memorije imaju ECC algoritam koji za **svaki bajt ima 3 ekstra bita**, potrebna za detekciju i korekciju jednobitnih grešaka.
- RAID 2 ima organizaciju **deljenja podataka na bit** (*bit-striping*) ili bajt nivou, a bez obzira na broj diskova podataka, potrebna su **još tri diska za ECC** koja mogu sačuvati podatke u slučaju otkaza bilo kog diska.
- RAID 2 je dobar po pitanju paralelizma, bolji je od RAID 0 po pitanju utroška diskova. Praktično se **ne koristi** (prevaziđen od strane RAID 3)



9.3 - RAID 3

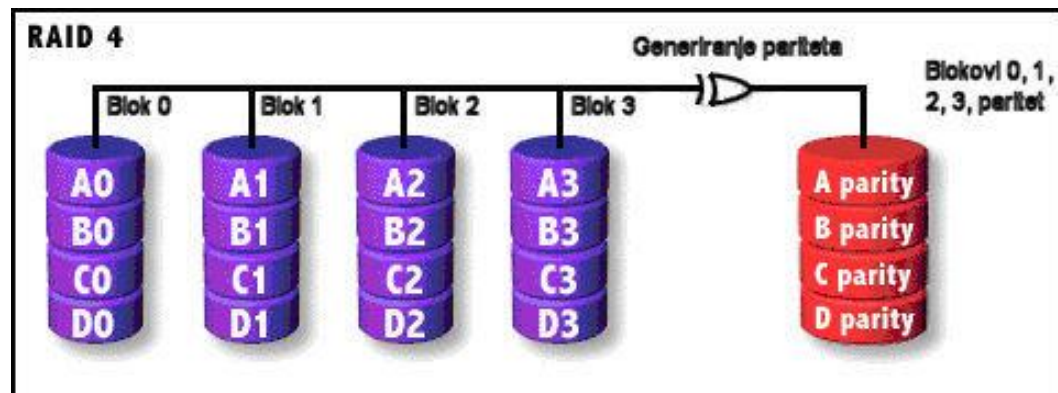
- Deljenje podataka na **nivou bita ili bajta**, a po pitanju redundanse prvi put se **uvodi parnost za diskove** (*bit-interleaved parity organization*).
- Ukoliko dođe do otkaza jednog diska **svi podaci su i dalje dostupni**.
- Za razliku od memorije u kojoj je teško odrediti tačnu poziciju greške, kod diskova **svaki pojedinačni disk-kontroler** zna da li je njegov pročitani sektor korektan ili ne.
- Zbog toga je **bit parnosti** (*parity bit, P*) dovoljan i za **detekciju i za korekciju greške**, pošto se tačno zna koji bit je pogrešan.
- Za sve diskove podataka **dovoljan je jedan bit parnosti**.
- Za oštećeni bit, dovoljan je jedan bit da čuva parnost i na bazi te parnosti **može da se rekonstruiše oštećeni bit**.
- Svako zapisivanje podataka zahteva **4 pristupa diskovima** (čitanje starih podataka i pariteta i ponovni upis)

Raid 3 - Disk Striping with Dedicated Parity Disk



9.3 - RAID 4

- Deljenje podataka **na nivou bloka** (*block striping*), a po pitanju redundanse **koristi se parnost za diskove na nivou bloka** (*block-interleaved parity organization*).
- čitanje jednog bloka obrađuje jedan disk, ostali mogu da obrađuju druge zahteve u paraleli
- čitanje i upis velikih segmenata obavlja se paralelizovano
- upis jednog bloka ili nezavisni upisi po jedan blok jesu problem: ne mogu se obavljati u paraleli jer svaki **zahteva čitanje bloka, izmenu i upis**, i to i za podatke i za blok parnosti
- za n blokova **dovoljan je jedan blok parnosti** koji je jednak veličini trake diska.
- Velika mana ovog rešenja je što dovodi **do preopterećenja diska parnosti** koji učestvuje u svakom ciklusu upisa pa tako postaje usko grlo



9.3 - RAID 5 (Rotating Parity Array)

- Veoma je slična RAID 4, samo što podatke i bitove parnosti **rasipa po svim diskovima**-za svaki paket od N blokova, **jedan disk(bilo koji)čuva parnost**, ostali podatke, ciklično (*block-interleaved distributed parity*)
- **Najčešće korišćeno rešenje u serverima**, jer pokušava da objedini brzinu RAID 0 (*striping*) i sigurnost RAID 1 (*mirroring*).
- Ovaj niz se naziva **rotirajući**, jer se informacija o parnosti ne upisuje uvek na isti disk, već svaki put na drugi.
- U ovom nizu uloga diskova se stalno smenjuje, time se **eliminise usporavanje** do koga bi došlo da se informacija o parnosti konstantno upisuje na isti disk (RAID 4).
- Zbog toga je **potreban hardverski kontroler**, jer on ne samo da generiše informaciju o parnosti, već on određuje smenu diskova, kao i to gde će koji podatak biti upisan.
- U slučaju otkaza bilo kog diska, **niz i dalje može da nastavi da radi**.
- Mana je ta što će **raditi sporije**, a i samo rekreiranje niza, posle zamene pokvarenog diska, **traje znatno duže** nego kod RAID 1
- Istovremeno se može odvijati **više upisa i čitanja** bez zagušenja

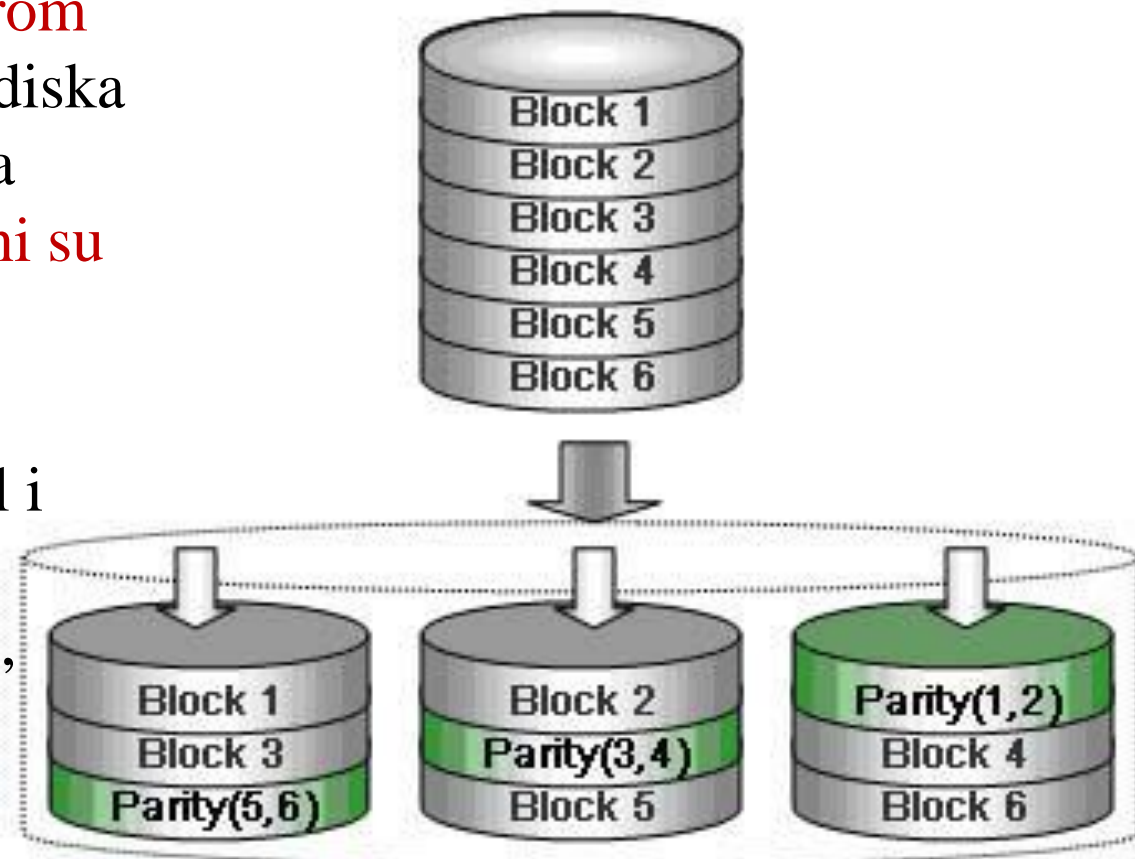
9.3 - RAID 5 (Rotating Parity Array)

Prednosti: najfleksibilniji od svih nizova, najbolji balans cena/performance/sigurnost od svih RAID, mogućnost više simultanih upisa, visoka brzina čitanja, osrednja brzina upisivanja, visoka efikasnost, za kreiranje potrebna minimalno tri diska

Mane: neefikasan sa transferom velikih fajlova, otkazivanje diska ima uticaj na performance, za najbolje performance potrebni su skupi kontroleri

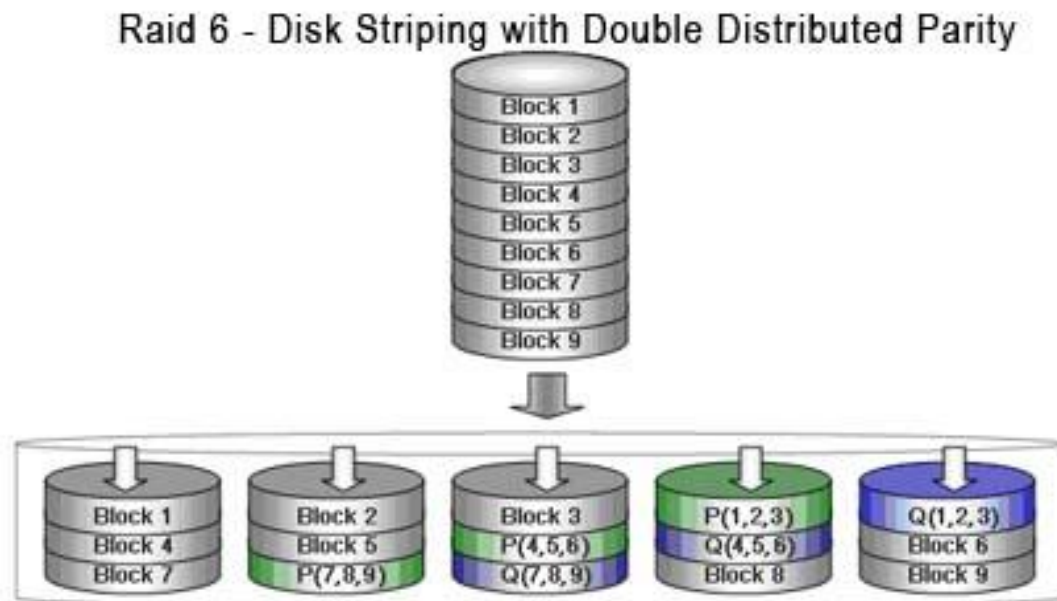
Primena: baze podataka, fajl i print serveri, web, e-mail i news serveri, intranet serveri, radne stanice

Raid 5 - Disk Striping with Single Distributed Parity



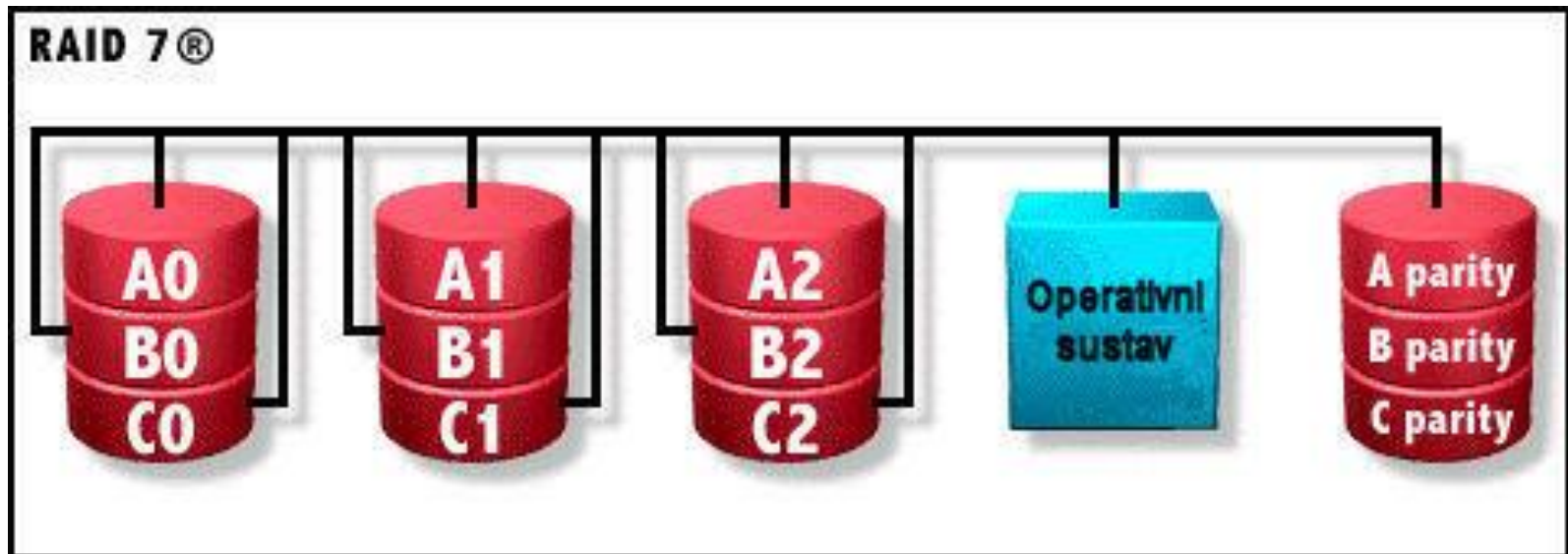
9.3 - RAID 6

- **RAID 6** (P+Q *redundancy scheme*) predstavlja **jedinu RAID kombinaciju koja može razrešiti problem u slučaju otkaza više diskova.**
- Radi kao RAID 5, ali umesto parnosti, **čuva dodatne redundantne ECC kodove** kako bi se zaštitio od otkaza više od jednog diska
- Rezultat ovog dvostrukog pariteta je svakako **povećanje tolerancije**, tako da RAID 6 može toleriš **kvar bilo koja dva diska.**
- RAID 6 je **malo lošiji** od RAID 5 zbog izračunavanja i zapisivanja više paritetnih informacija, dok je **kod čitanja podataka malo bolja situacija** zbog čitanja podataka sa jednog dodatnog diska.



9.3 - RAID 7

- RAID 7 se zasniva na konceptu korišćenom kod RAID 3 i RAID 4 niza ali sa velikim poboljšanjima.
- Uključuje korišćenje velike, brze memorije raspoređene u više nivoa i koristi specijalni *Real-time procesor* za asinhrono upravljanje nizom što omogućava veliki broj istovremenih operacija.
- Poboljšane su performanse proizvoljnog čitanja i zapisivanja u odnosu na RAID 3 i RAID 4, jer ona u velikoj meri zavise od diska pariteta, a to je ovde izbegnuto primenom brze memorije



9.3 - RAID structure (0-6)



(a) RAID 0: non-redundant striping



(b) RAID 1: mirrored disks



(c) RAID 2: memory-style error-correcting codes



(d) RAID 3: bit-interleaved Parity



(e) RAID 4: block-interleaved parity

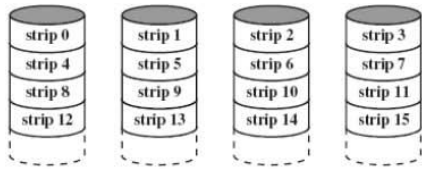


(f) RAID 5: block-Interleaved distributed parity

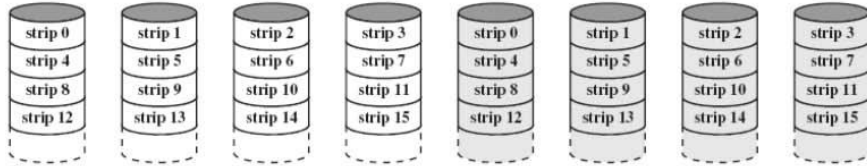


(g) RAID 6: P + Q redundancy

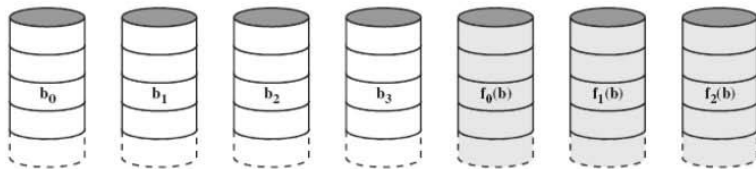
9.3 - RAID structure (0-6)



(a) RAID 0 (Nonredundant)



(b) RAID 1 (Mirrored)



(c) RAID 2 (Redundancy through Hamming code)

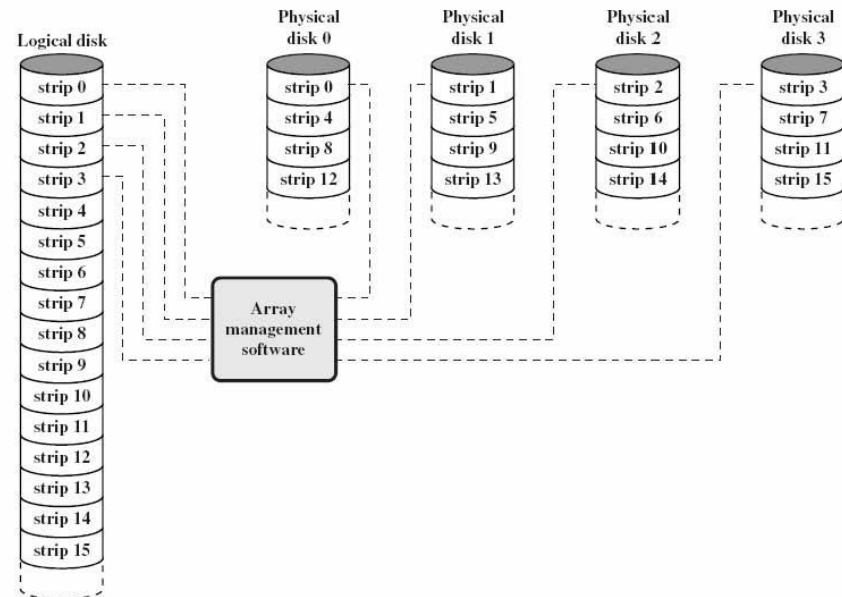
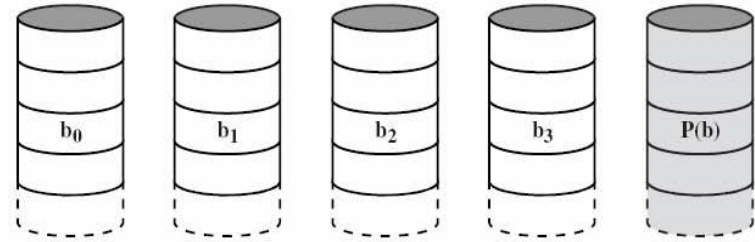
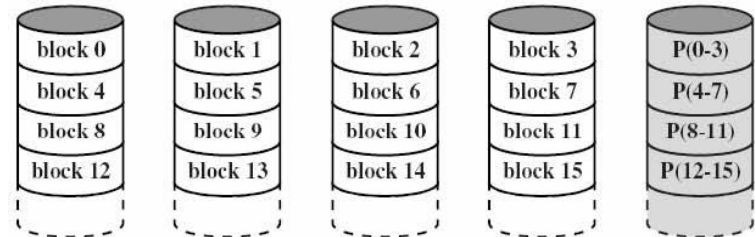


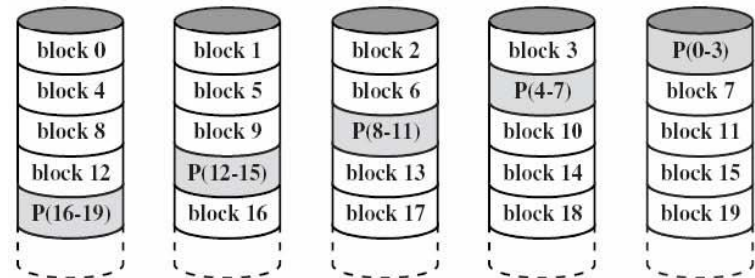
Figure 69 Data Mapping for a RAID Level 0 Array



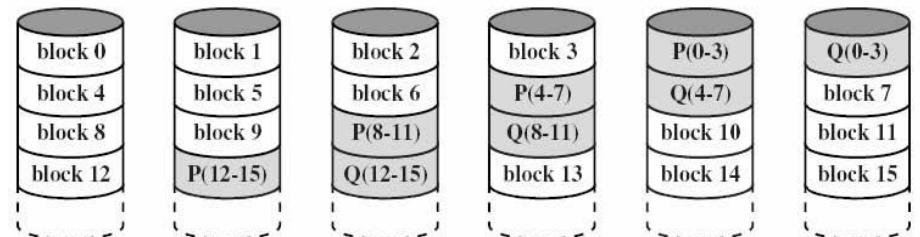
(d) RAID 3 (Bit-interleaved parity)



(e) RAID 4 (Block-level parity)



(f) RAID 5 (Block-level distributed parity)



(g) RAID 6 (Dual redundancy)

9.3 - RAID strukture

Prilikom projektovanja RAID sistema, projektant treba da odgovori na **nekoliko pitanja**:

1. Koji će RAID nivo da se primeni?
2. Koliko diskova će biti u RAID sistemu?
3. Koliko grupa parnosti će se napraviti?
4. Koliko će biti veličina trake?

- Potrebno je odabrati **kvalitetan RAID kontroler**.
- Svi hard diskovi koji se nalaze u RAID nizu **treba da budu istog kapaciteta i performansi**
- Većina novih matičnih ploča **ima ugrađene RAID kontrolere** koji podržavaju konfiguracije RAID 0,1,0+1 i 5 – ovi kontroleri nisu hardverski, već softverski.
- Hardverski RAID **sam deli podatke u trake**, dok u slučaju softverskog RAID-a **to obavlja procesor**.

9.4 - Višestruki RAID nizovi

- Pojedine RAID strukture imaju **određene prednosti i nedostatke**
- Vršiti se **kombinovanje prednosti više različitih RAID polja** tako da se dobiju neke nove (bolje) karakteristike novonastalog niza.
- Tako su nastala **višestruka, ili ugnježdjena RAID polja** (*multiple, nested, multi-RAID*). Ponekad se nazivaju **dvo-dimenzionalna polja**, zbog šematskog izgleda prikaza na kome su vidljiva **dva polja diskova**.
- Višestruka RAID polja se najčešće koriste zbog svojih poboljšanih karakteristika jer ona **imaju bolje performanse** nego pojedinačna RAID polja od kojih su sastavljena.
- Najčešće korišćeno polje, u kombinaciji s ostalima, je **RAID 0 polje**, koje se najčešće kombinuje sa redundantnim poljima kao što su RAID 1, RAID 3 ili RAID 5, koja **osiguravaju toleranciju** dok RAID 0 polje **osigurava visoke performanse**.
- **Troškovi su povećani** u odnosu na troškove jednostrukih RAID polja.
- **Povećan je broj potrebnih diskova**, a kako je polje složenije veći troškovi nastaju i **potrebom za boljim upravljanjem i održavanjem**

9.4 - RAID 0+1 i 1+0

Prednosti:

- nema generisanja parnosti
- lak za implementaciju
- koristi pun kapacitet diskova
- visoka tolerancija greške
- odlične performanse pri upisu i čitanju podataka

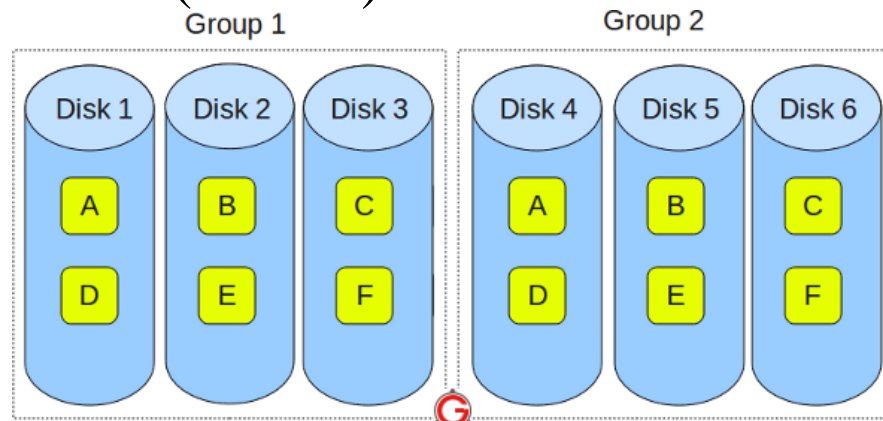
Mane:

- neefikasno iskorišćenje mesta na diskovima
- visoka cena
- mala skalabilnost

Upotreba:

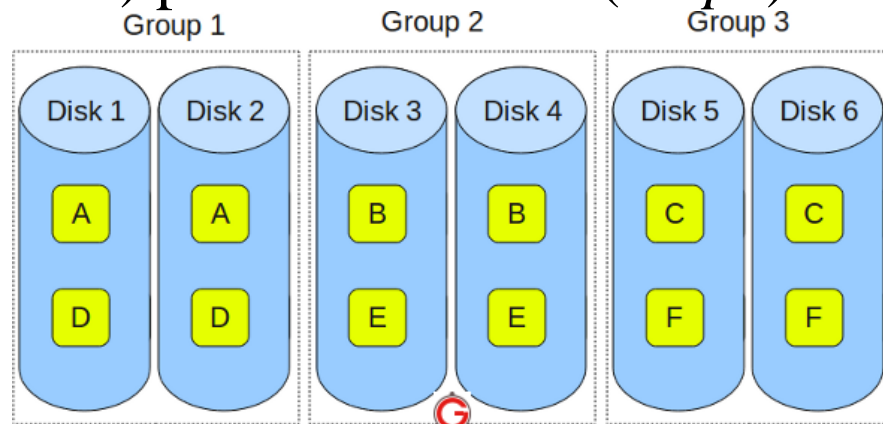
- fajl serveri
- baze podataka
- gde se zahtevaju viske performanse sa visokom sigurnošću

RAID 0+1: N diskova u traci (*stripe*), i još N takvih u drugoj traci koja je ogledalo (*mirror*)



RAID 01 – Blocks Striped. (and Blocks Mirrored)

RAID 1+0: N parova diskova u ogledalu (*mirrored*) povezani u traku (*stripe*)



RAID 10 – Blocks Mirrored. (and Blocks Striped)

Hvala na pažnji !!!



Pitanja

? ? ?