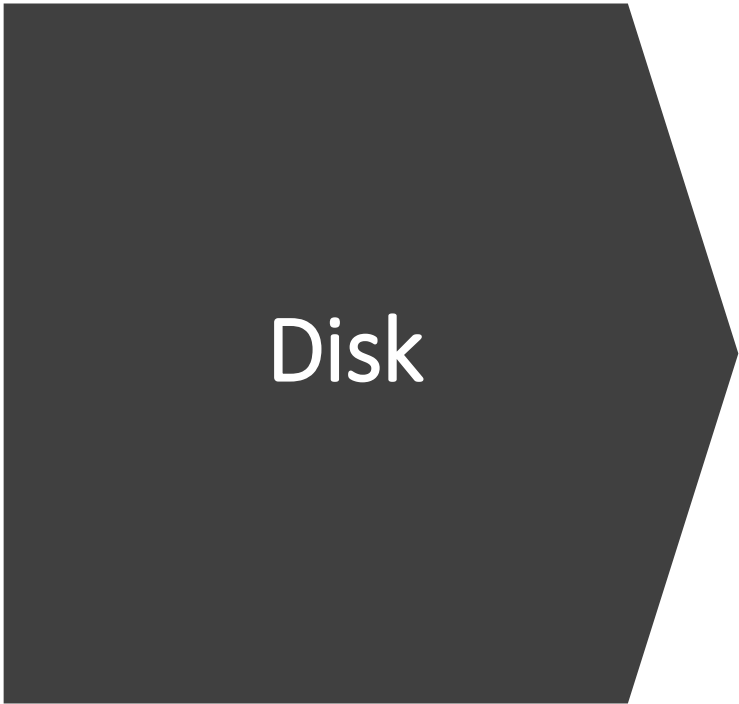




OPERACIJSKI SUSTAVI

Algoritmi raspoređivanje poslova
diska



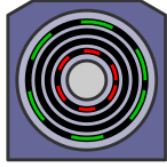
5 1/4" hard drive (CAV)



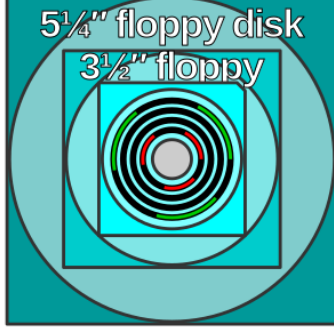
MiniDisc (CLV)



Iomega Zip (CAV)



8" floppy disk (CAV)



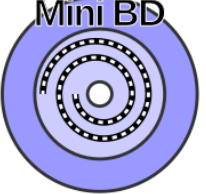
Audio Compact Disc (CLV*)



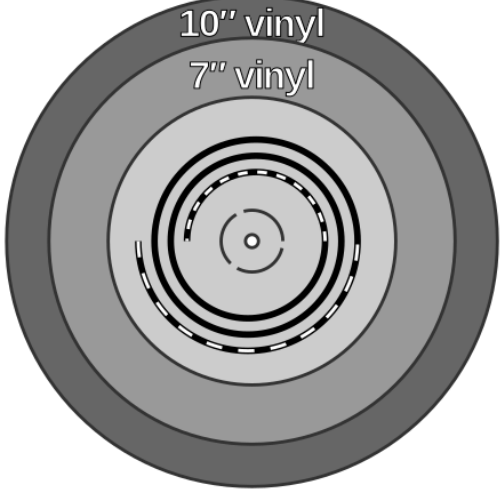
Digital Versatile Disc (CLV*)



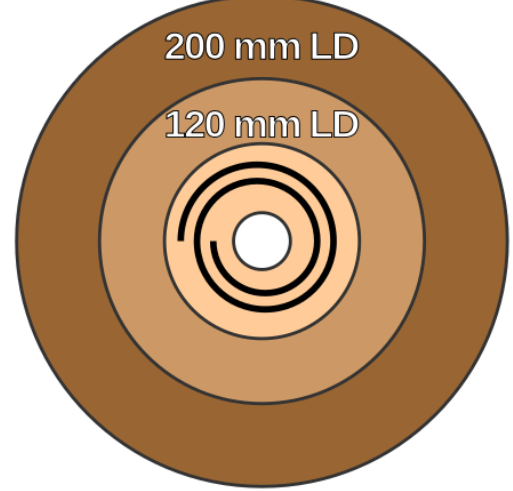
Blu-ray Disc (CLV)



12" vinyl record (CAV)

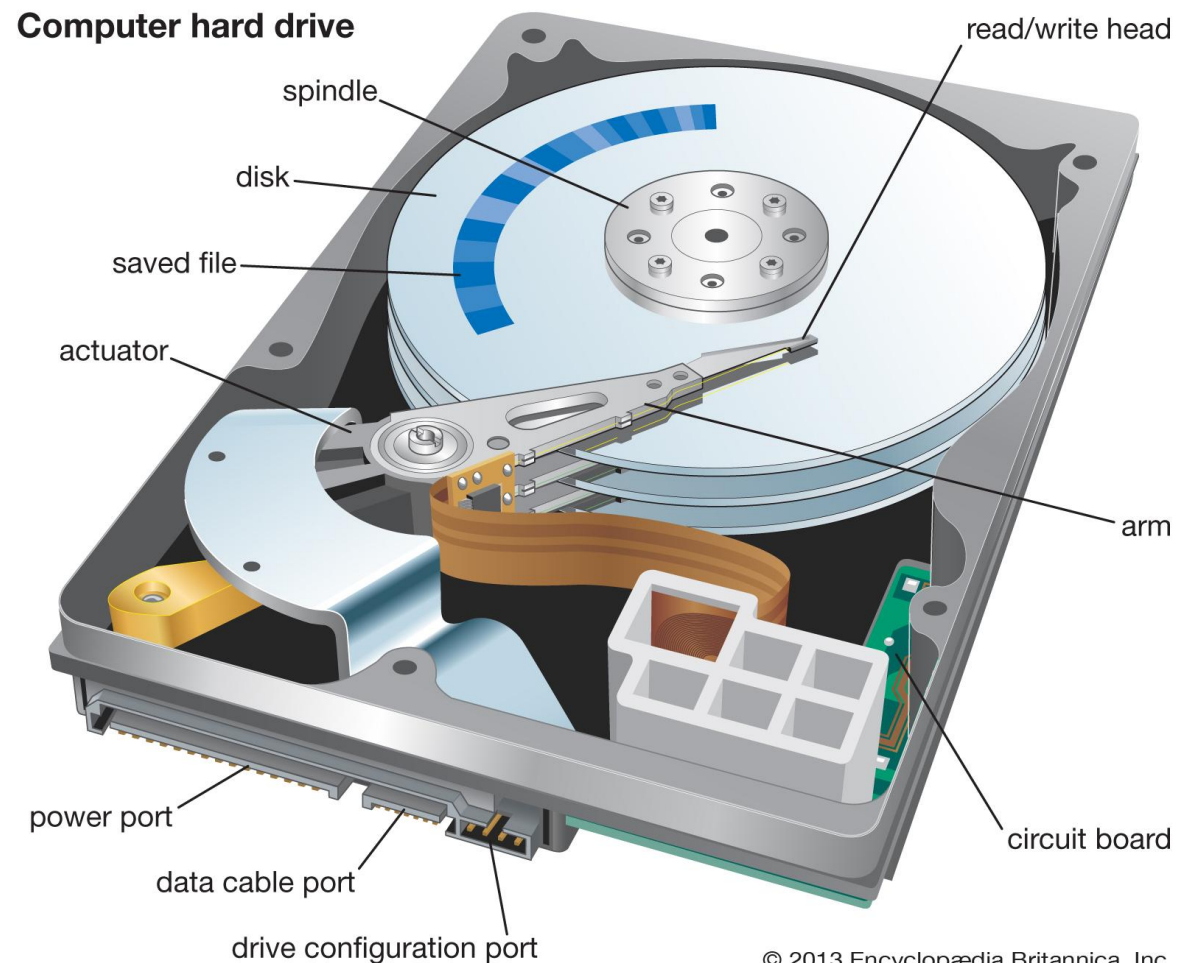


LaserDisc (CLV/CAA/CAV)



Djelovi diska

- Jedna ili više okruglih ploča presvučenih magnetskim materijalom
 - koje se vrte **konstantnom brzinom** s mehanizmom magnetskih glava
 - koje se mogu pomicati približno radijalno iznad ploča
- Upravljačkog sklopa, spremnika, sučelja prema elektromehaničkom dijelu
- Sučelje prema sabirnici računala



© 2013 Encyclopædia Britannica, Inc.

How does it works...

- How does HDD work:

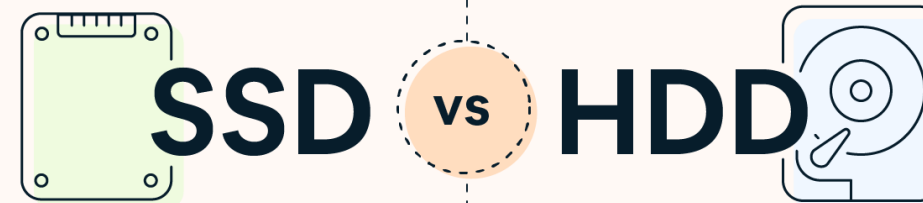
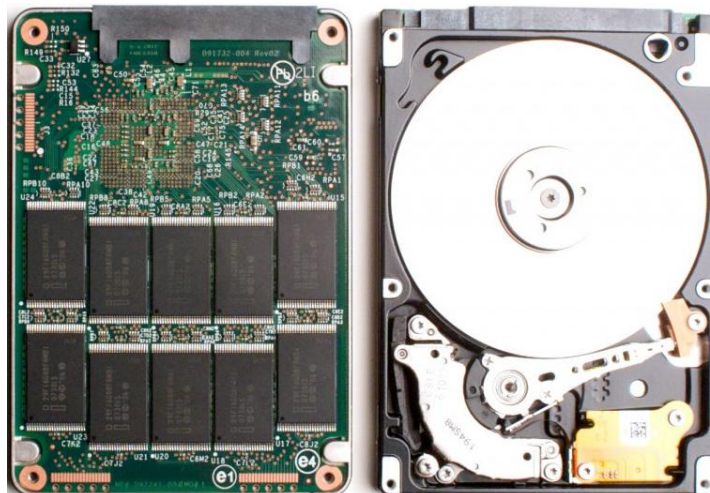
<https://www.youtube.com/watch?v=n6uPALWAYxc&t>

- How does SSD work:

<https://youtu.be/E7Up7VuFd8A?si=1MsUacYk4o-4IT3p>

HDD vs. SSD

- Hard disk drive
 - Koristi diskove
- Solid-state drive
 - Koristi flash memoriju (čipove)



faster	✓	✗	slower
shorter lifespan	✗	✓	longer lifespan
more expensive	✗	✓	cheaper
non-mechanical (flash)	✓	✗	mechanical (moving parts)
shock-resistant	✓	✗	fragile
best for storing operating systems, gaming apps, and frequently used files			best for storing extra data, such as movies, photos, and documents

Izvor: Avast

Algoritmi raspoređivanje poslova diska:

- FCFS
- SSTF
- SCAN (elevators/lift)
- LOOK
- C-SCAN (one-way elevator/jednosmjerno dizalo)
- C-LOOK
- RSS (Random Scheduling) – kao što ime kaže
- LIFO (Last-In First-Out) - najnoviji poslovi se servisiraju prije postojećih
- N-STEP SCAN - svi zahtjevi koji pripadaju međuspremniku bit će servisirani u jednom potezu
- F-SCAN - koristi dva pod-reda čekanja
- Completely Fair Queuing (CFQ) – upotrijebljeno u Linux 2.0
- Noop scheduler – slično kao FIFO
- Budget Fair Queueing (BFQ) – koristi se od Linux 5.0
- **Seek distance** – vrijeme pretraživanja
 - vrijeme potrebno za pomicanje ruke (*arm*) diska na određenu stazu gdje se podaci trebaju čitati ili pisati.
- Referentni string - Struktura podataka
 - lista brojeva sektora kojima različite zadaće pristupaju.

Zašto koristimo algoritme?

- Višestruki I/O zahtjevi mogu stizati od strane različitih procesa i samo jedan I/O zahtjev može istovremeno poslužiti diskovni kontroler. Stoga drugi I/O zahtjevi moraju čekati u redu čekanja i trebaju biti zakazani.
- Dva ili više zahtjeva mogu biti udaljeni jedan od drugog pa mogu rezultirati većim pomakom kraka diska.
- Tvrdi diskovi su jedan od najsporijih dijelova računalnog sustava i stoga im je potrebno pristupati na učinkovit način.

Detaljno o Algoritmima:

- ALgoritmi:
 - FCFS
 - SSTF
 - SCAN (elevator/dizalo)
 - LOOK
 - C-SCAN (one-way elevator/jednosmjerno dizalo)
 - C-LOOK
- Važne informacije:
 - Koji je trenutni položaj glave
 - Referentni niz – I/O zahtjevi
 - Koja je veličina diska
- **Seek distance**
 - Vrijeme pretraživanja (pomicanje glave za čitanje/pisanje)

FCFS – First Come First Serve

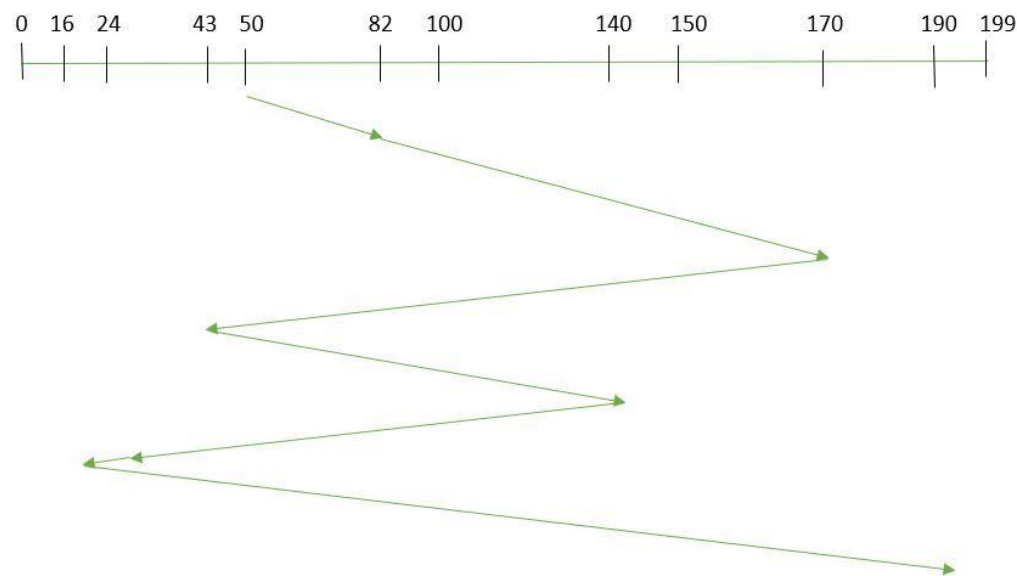
- Tj. FIFO
- Zadaćama se dodjeljuje pristup disku onim redoslijedom kojim pristižu
- **Prednosti:**
 - Nema izgladnjivanja
 - Jednostavan za implementaciju
- **Nedostak:**
 - Slučajan pristup disku
 - Nema uštede vremena

FCFS - zadatak

- Referentni niz: **82,170,43,140,24,16,190**
- Glava diska je na polju/nalazi se na: **50**

Korak	Trenutna zadaća	Sljedeća zadaća	Razlika
1	50	82	32 ABS(50-82)
2	82	170	88 ABS(82-170)
3	170	43	127 ...
4	43	140	97
5	140	24	116
6	24	16	8
7	16	190	174

SUM: 642



SSTF - Shortest Seek Time First

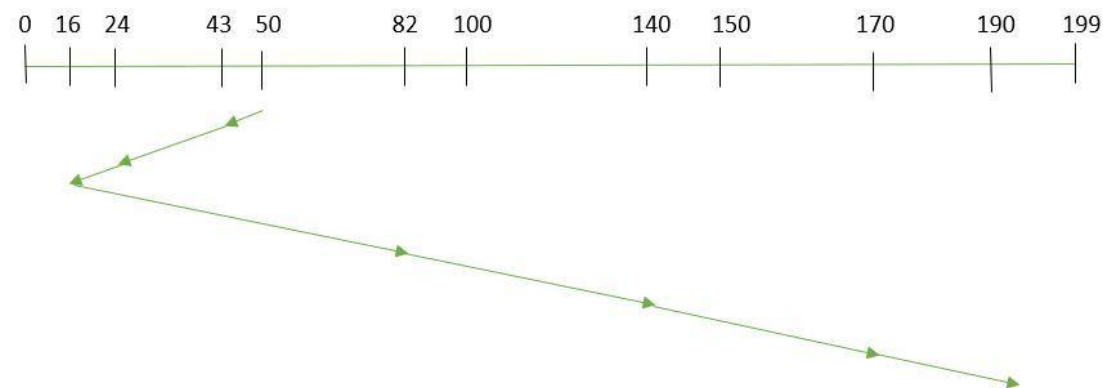
- Najmanje vrijeme traženja
- Algoritam:
 - Kada nova zadaća dođe izračunaj udaljenost od zadaće koja trenutno pristupa disku
 - Stavi zadaću na odgovarajuće mjesto u redu (sortiranom po udaljenosti)
 - Kad zadaća završi **odaberi najbližu**
 - Uvijek se mora krenuti od jedne zadaće, a mi ćemo uzeti da je prva pozicija u nizu ili gdje se trenutno glava nalazi

SSTF - zadatak

- Referentni niz: **82,170,43,140,24,16,190**
- Glava diska je na polju/nalazi se na: **50**

Korak	Trenutna zadaća	Sljedeća zadaća	Razlika
1	50	43	7
2	43	24	19
3	24	16	8
4	16	82	66
5	82	140	58
6	140	170	30
7	170	190	20

SUM: 208



Redosljed: 50, 43, 24, 16, 82, 140, 170, 190

SSTF – prednosti/mane

- **Prednosti:**

- Prosječno vrijeme odgovora se smanjuje
- Povećava se propusnost

- **Nedostaci:**

- Potrebno računanje vremena udaljenosti unaprijed
- Može uzrokovati gladovanje za zahtjev ako ima duže vrijeme traženja u usporedbi s dolaznim zahtjevima
- Velika varijacija vremena odgovora jer SSTF favorizira samo neke zahtjeve

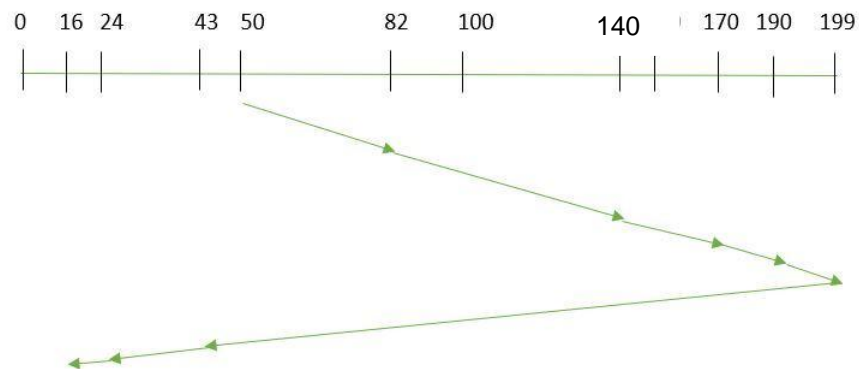
SCAN

- Tzv. „Dizalo”
 - Kada nova zadaća dođe izračunaj udaljenost od zadaće koja trenutno pristupa disku
 - Stavi zadaću na odgovarajuće mjesto u redu (sortiranom po udaljenosti)
 - Kad zadaća završi **odaberi najbližu u jednom smjeru**
 - Kada **završimo put** u jednom smjeru **do kraja vraćamo se u suprotnom smjeru**

SCAN - zadatak

- Referentni niz: **82,170,43,140,24,16,190**
- Glava diska je na polju/nalazi se na: **50**
 - Prvi sektor diska je na **0**-tom, a zadnji na **199**-tom sektoru

Korak	Trenutna zadaca	Sljedeća zadaca	Razlika
1	50	82	32
2	82	140	58
3	140	170	30
4	170	190	20
5	190	199	9
6	199	43	156
7	43	24	19
8	24	16	8



Redosljed: 50, 82, 140, 170, 190, **199**, 43, 24, 16

SUM: 332

Brzo računanje: $ABS(50-199) + ABS(199-16) = 332$

SCAN – prednosti/mane

- **Prednosti:**

- Visoka propusnost
- Mala varijacija vremena odgovora
- Prosječno vrijeme odgovora

- **Nedostaci:**

- Problem sa pravednošću, zadaće koje pristupaju sredini diska će biti dva puta brže posluživane od ostalih
- Dugo vrijeme čekanja na zahtjeve za lokacije koje je upravo posjetio disk

LOCK

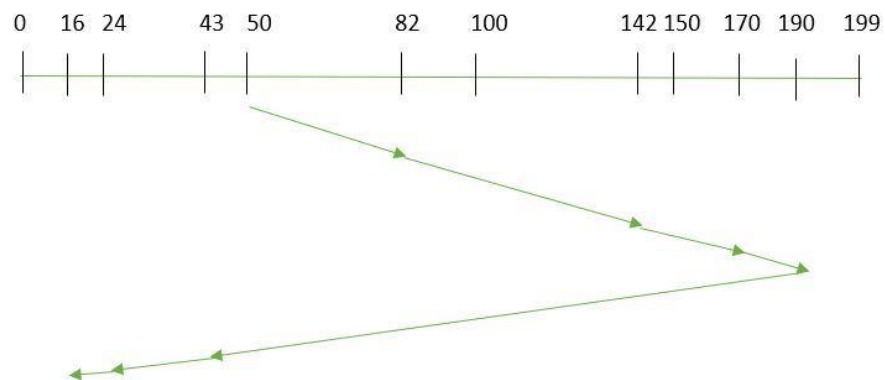
- Isto kao i SCAN ali **ignorira** prvi i zadnji sektor na disku (*ako nije u referentnom stringu*)

LOCK - zadatak

- Referentni niz: **82,170,43,140,24,16,190**
- Glava diska je na polju/nalazi se na: **50**
 - Prvi sektor diska je na **0**-tom, a zadnji na **199**-tom sektoru

Korak	Trenutna zadaća	Sljedeća zadaća	Razlika
1	50	82	32
2	82	140	58
3	140	170	30
4	170	190	20
5	190	43	147
6	43	24	19
7	24	16	8

SUM: 314



Redosljed: 50, 82, 140, 170, 190, 43, 24, 16

Brzo računanje: $ABS(50-190) + ABS(190-16) = 314$

LOCK – prednosti/mane

- **Prednosti:**

- Omogućuje ujednačenije vrijeme čekanja u usporedbi sa SCAN-om

- **Nedostaci (isto kao i LOOK):**

- Problem sa pravednošću, zadaće koje pristupaju sredini diska će biti dva puta brže posluživane od ostalih
- Dugo vrijeme čekanja na zahtjeve za lokacije koje je upravo posjetio disk

C-SCAN (jednosmjerno dizalo)

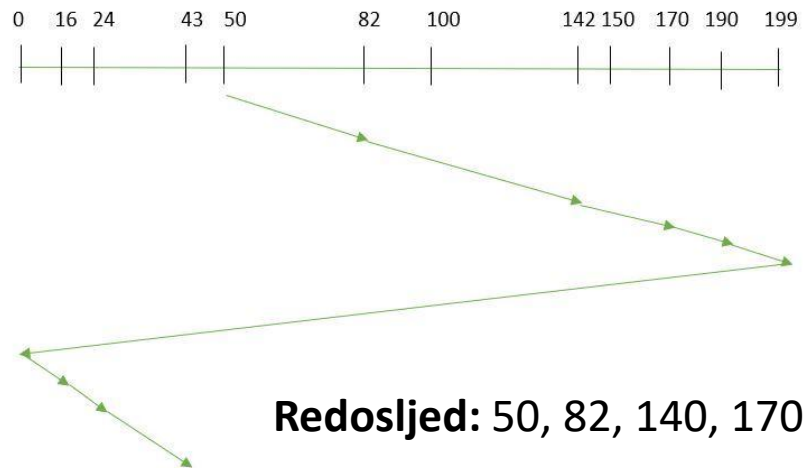
- Circular SCAN
- Jednosmjerno dizalo – glava za čitanje ide samo u **jednom smjeru**, kada dođe do **kraja – ide na početak**

C-SCAN

- Referentni niz: **82,170,43,140,24,16,190**
- Glava diska je na polju/nalazi se na: **50**
 - Prvi sektor diska je na **0**-tom, a zadnji na **199**-tom sektoru

Korak	Trenutna zadaća	Sljedeća zadaća	Razlika
1	50	82	32
2	82	140	58
3	140	170	30
4	170	190	20
5	190	199	9
6	199	0	199
7	0	16	16
8	16	24	8
9	24	43	19

SUM: 391



Brzo računanje: $ABS(50-199) + ABS(199-0) + ABS(0-43) = 391$

C-SCAN – prednosti/mane

- Prednosti:

- Poboljšanje vremena u odnosu na FIFO
- Nema izgladnjivanja
- Riješen problem češće posluživanja sredine diska

- Nedostaci:

- Dugo vrijeme čekanja na zahtjeve za lokacije koje je upravo posjetio disk

C-LOOK

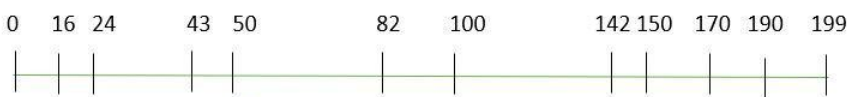
- Radi isto kao i C-SCAN uz iznimku da **ne posjećuje prvi odnosno zadnji sektor** ako nije u referentnom stringu.

C-LOOK - zadatak

- Referentni niz: **82,170,43,140,24,16,190**
- Glava diska je na polju/nalazi se na: **50**
 - Prvi sektor diska je na **0**-tom, a zadnji na **199**-tom sektoru

Korak	Trenutna zadaća	Sljedeća zadaća	Razlika
1	50	82	32
2	82	140	58
3	140	170	30
4	170	190	20
5	190	16	174
6	16	24	8
7	24	43	19

SUM: 341



Redosljed: 50, 82, 140, 170, 190, 16, 24, 43

Brzo računanje: $ABS(50-190) + ABS(190-16) + ABS(16+43) = 341$

C-LOCK – prednosti/mane

- **Prednosti:**

- Isto kao i SCAN
- Ne mora ići na početak/kraj diska

- **Nedostaci:**

- Dugo vrijeme čekanja na zahtjeve za lokacije koje je upravo posjetio disk

Usporedbe

Algoritam	Rezultat
SSTF	208
LOOK	314
SCAN	332
C-LOCK	341
C-SCAN	391
FIFO	642

- SSTF – izglednjanje (drugi nemaju taj nedostatak)
- LOOK i C-LOOK ne vraćaju glavu za čitanje na početak/kraj
- LOOK i SCAN – pravednost (podaci na sredini diska biti će prije posluženi od onih na kraju)
- LOOK je sličan kao SCAN isto kao što je i C-LOOK sličan C-SCAN

Zadatak

- Referentni niz: 1, 2, 3, 4, 12, 11, 12, 11, 10, 5, 35, 11, 10, 1, 35, 30, 29, 12, 13
- Prvi sektor je 0, zadnji sektor je sektor 50
- Glava diska je na sektoru: 12 i glava se kreće u lijevo (prema padajućem sektoru)
- Upotrijebiti algoritme:
 - A) FIFO
 - B) SSTF,
 - C) SCAN
 - D) LOOK
 - E) C-SCAN
 - F) C-LOOK
- Za svaki algoritam napraviti tablicu sa koracima i izračunati seek distance.

Rješenja:

- A) FIFO = 147
- B) SSTF = 51
- C) SCAN = 87 ili 47
- D) LOOK = 72 ili 35
- E) C-SCAN = 99 ili 125
- F) C-LOOK = 67

Napomena: Možda su neka rješenja kriva!?



Hvala na pažnji!

