

# Simuliranje C-SCAN (Circular SCAN) algoritma za upravljanje diskom

Napisati konkurentni program koji simulira raspoređivanje diska po C-SCAN algoritmu. Algoritam C-SCAN omogućava uniformnije vreme čekanja od SCAN algoritma tj. lift algoritma. Glava diska se pomera **sa jednog na drugi kraj diska** uz servisiranje UI zahteva. Kada stigne do kraja, vraća se na početak diska **bez servisiranja zahteva**. Staze se tretiraju kao kružna lista, tj. posle poslednje staze ponovo se prelazi na prvu stazu diska.

Klasa OS predstavlja deo operativnog sistema koji je zadužen za raspoređivanje zahteva. Metodu uputi\_ui\_zahtev pozivaju niti koje predstavljaju procese. U ovoj metodi postavlja se zahtev raspoređivaču diska da izvrši UI operaciju na određenoj stazi tj. cilindru diska. Proces mora da sačeka da zahtev bude izvršen, pre nego nastavi dalje izvršavanje.

Metodu obradi\_zahtev poziva nit koja predstavlja raspoređivač diska. Metoda je zadužena da obradi jedan zahtev iz skupa zahteva. Zahtev za obradu se iz skupa zahteva bira po C-SCAN principu. Obrada zahteva traje 300 ms. Rezultat metode je broj staze kojoj je disk pristupio pri obradi zahteva.

## PRIMER RADA C-SCAN ALGORITMA:

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

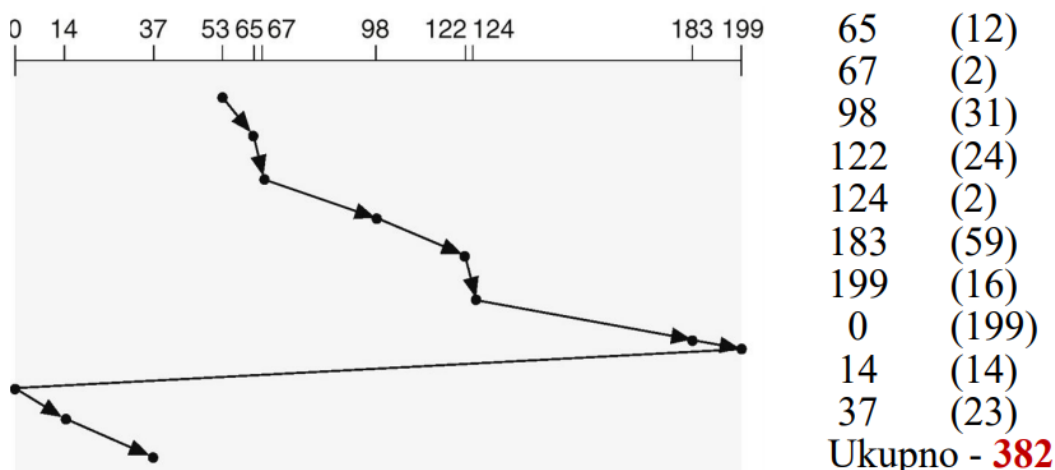
{ 98, 183, 37, 122, 14, 124, 65, 67 }

Pretpostavimo da disk ima 200 staza, od 0 do 199. Glava za upis i čitanje se trenutno nalazi na stazi 53, a prethodno je bila na stazi 100.

Brojevi u zagradi predstavljaju pomeraj koji glava diska pravi pri prelasku na sledeću stazu, izražen u broju staza.

Trenutna pozicija glave je na stazi 53

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



Slika 1. Primer C-SCAN algoritma

Na Slika 1. prikazan je redosled obrade zahteva primenom C-SCAN algoritma. Može se uočiti da glava diska polazi od trenutnog položaja (u navedenom primeru to je 53. staza) i ide do poslednje staze, nakon čega se vraća na početak i kreće sa obradom zahteva iz intervala od 0 do 53 staze.

### Napomene:

1. Ne pokušavati rešavati ceo zadatak odjednom! Prvo krenuti od skice problema na papiru. Papir ćete dobiti od asistenata. Dobro i sa razumevanjem pročitati tekst zadatka. Realizovati deo po deo zadatka.
2. Funkcija main sadrži test scenario koji treba da bude od pomoći u proveru validnosti algoritma. Test scenario se može menjati po želji ali je napravljen tako da se lako može istestirati da li vam program radi ispravno. Pre predaje zadatka trebalo bi vratiti test scenario u originalni oblik (tj. funkciju main), ako su u međuvremenu rađene neke promene kako bi zadaci bili efikasno pregledani.
3. Komentari su obavezni.