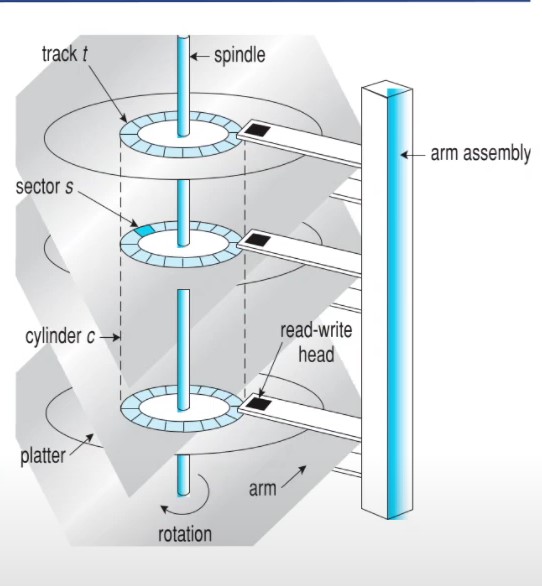
**Storage System**

*Disku* përbëhet nga *pllaka* të cilat janë të vendosura njër mbi tjetrën në një mbajtës që rrotullohet, mbi secilën *pllak* gjendet një *kokë e shrkim/leximit* , kët koka janë të ngjitur për një krahu mbajtës. *Pllakat* janë të ndara në mënyr logjike në *shirita*, secili *shitir* është i ndarë tutje në *sektorë*. Bashkësia e renditur e *shiritave* përbëjn një cilindër.

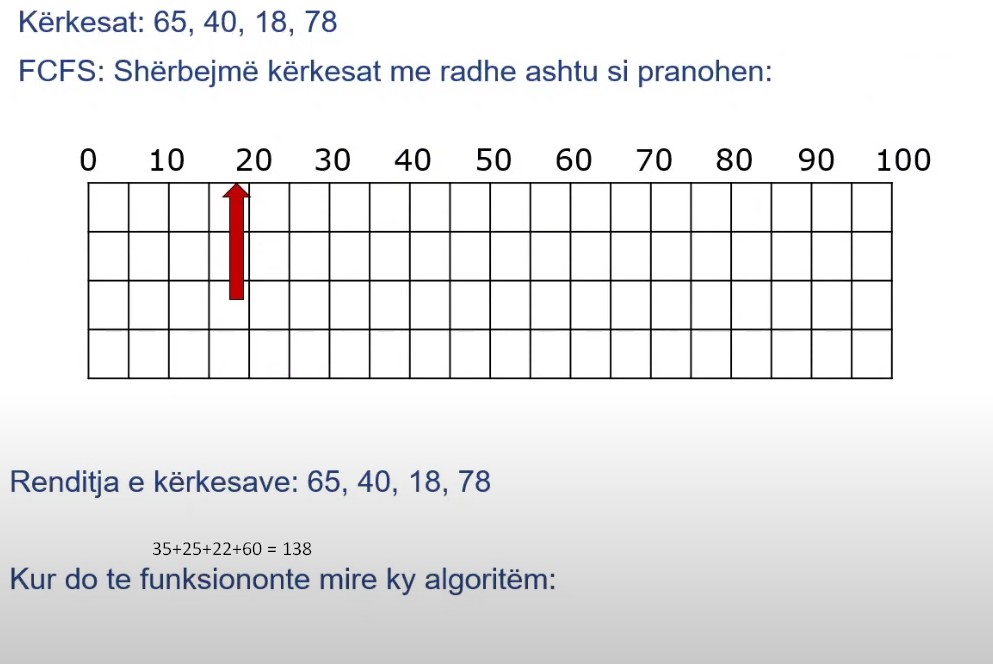


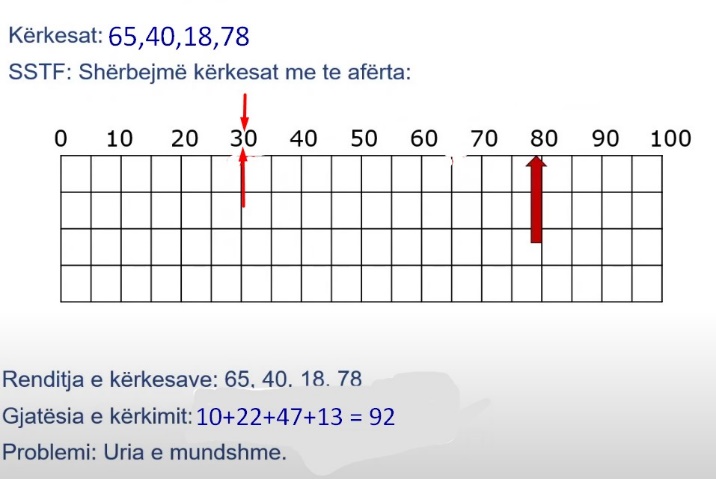
Disk I/O = Koha e kërkimit+Vonesa e rrutullimeve+Transferimi.

Për të fituar shpejtësi në *kohën e qasjes(Disk I/O)* duhet të minimizohet koha e kërkimit dhe vonesa e rrotullimit.

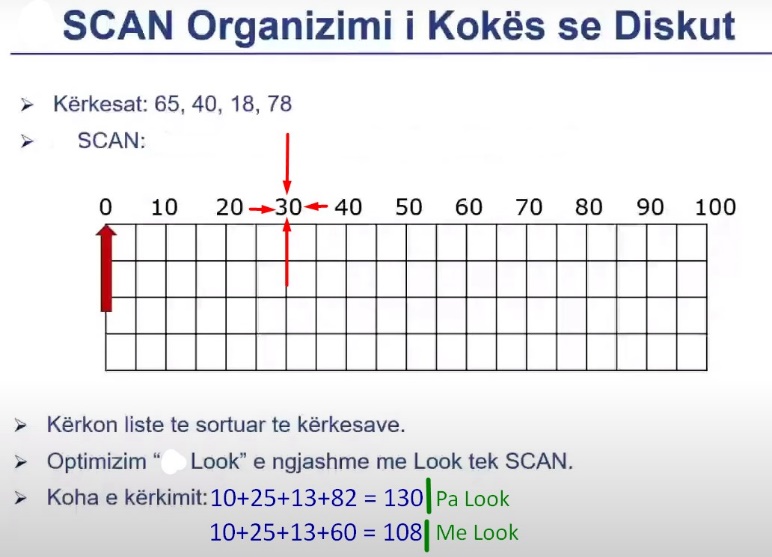
Në *kohën e qasjes* ndikojnë edhe faktorë të tjerë si : *Madhësia e disqeve* – Sa më të vogla që janë disqet aq më pak rrotullime duhet bërë. *Shpejtësia e rrotullimeve*. *Organizimi i operacioneve në disk* ashtu që të minimalizohet lëvizja e kokës. *Vendosja e të dhënave* të njëjta në sektorë të afërt. *Të dhënat* që përdoren më shpesh të vendosen më afër kokës së lexim/shkrimit. *Madhësia e sektorit* (bllokut) nëse kemi blloqe të vogla të dhënat e njëjta do duhet shpërndarë në shumë blloqe kjo do rezultoj në më shumë rrotullime, nëse kemi blloqe të mëdha në tjetrën anë do të shfaqet fragmentim i brendshëm. SSD nuk e kanë problemin me kohën e qasjes.

Ekzistojnë 4 algoritme që përdoren për ta bër *schedul kokën e disku* (për të përzgjedhur se në cilin seksion të kërkohet) :

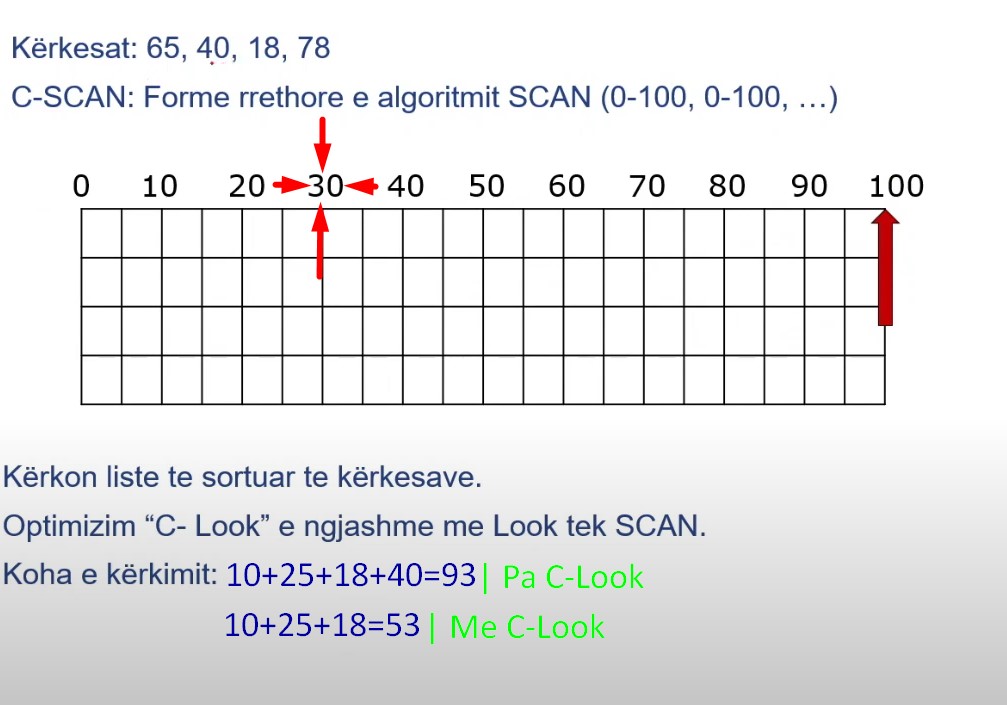
**1**.*First-come-first-served* (FCFS) : Lexon në bazë të rradhës që i kanë ardhur në kërkesa. Funksionon mirë vetëm nëse kemi të bëjm me ngarkesa të vogla ose jemi duke përdorur SSD.

**2.***Shortest seek time first (SSF)* : Lexon nga sektorët që janë më afër kokës së lexim/shkrimit.

**3**.*Scan* : Koka lëviz nga fillimi deri në fund dhe nga fundi deri në fillim duke lëvizur duke lexuar nga sektorët për të cilët ka kërkesa. Për me kanë ma efiqient e kërkon që lista e kërkesave të jet e sortuar. Mund të optimizohet tutje nëse përdorim *Look* – E kufizon lëvizjen e kokës duke paracaktuar pik fillestare sektorin më të vogël nga lista e kërkesave dhe kufi max vlerën e fundit të listës së kërkesave.



**4**.*C-San* është formë rrethore e algoritmit *Scan*. Shkon nga fillimi deri në fund e pastaj kthehet në fillim instant. Edhe ky algoritëm mund të optimizohet nëse përdorin *C-Look* që ka logjikën e njëjtë me *Look*.



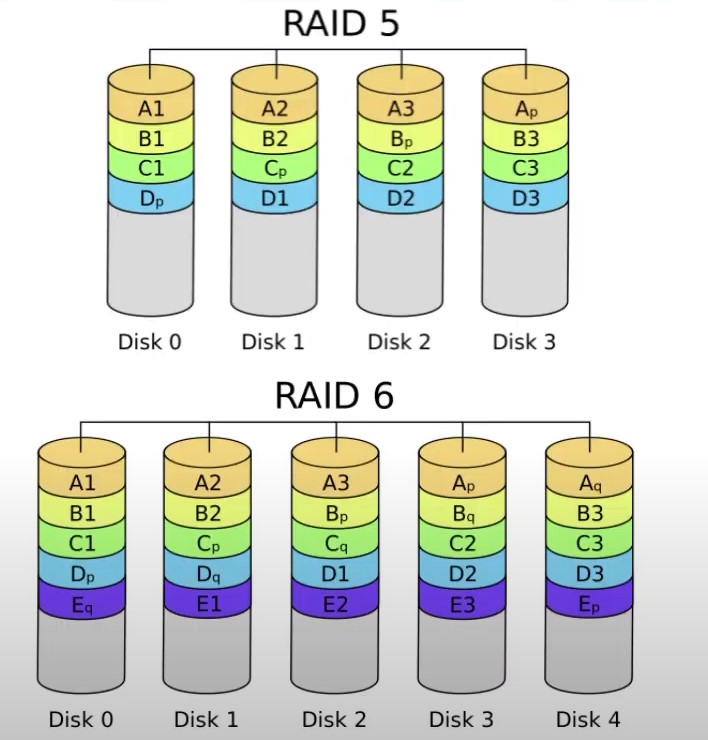
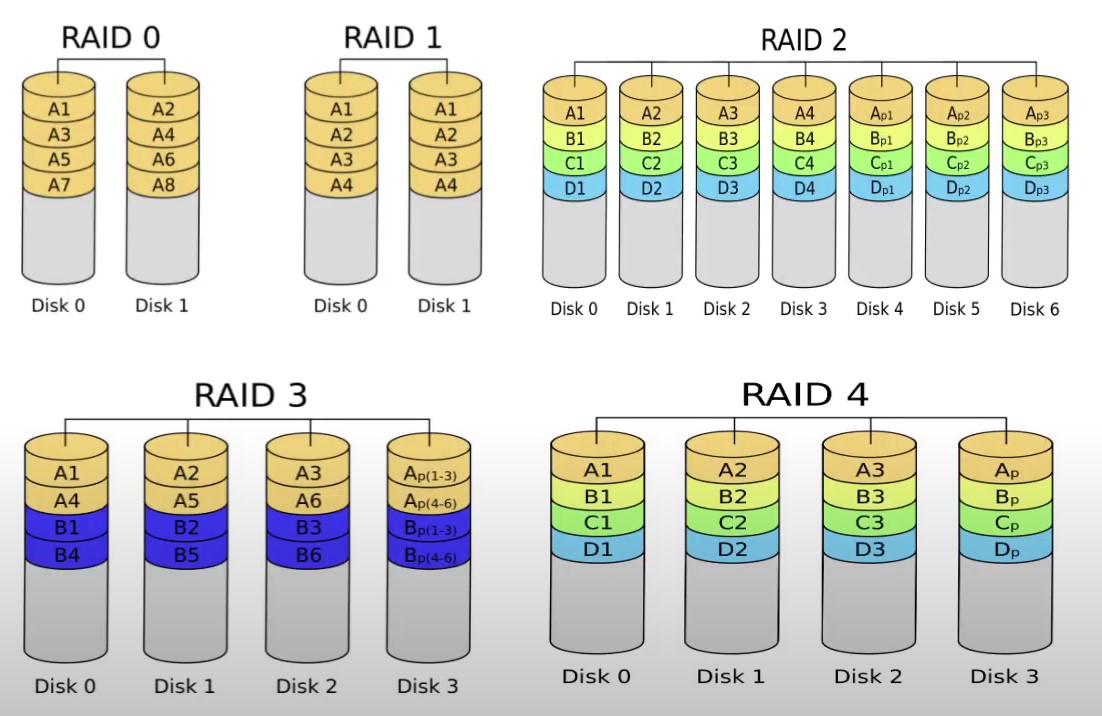
OS duhet të kujdeset edhe për sinkronizimin e leximit dhe përpunimit të të dhënave që merren nga Disk. Për këtë i përdor dy mekanizma: **1**.*Interleaving* – Len blloqe të zbrazta për arsye se mund të lexohet një e dhënë nga disku e ajo të dërgohet për përpunim e derisa OS ta përpunon krahu i leixm/shkrimit mund të kaloj blloqe pa lexuar e në të cilat mund të ket të dhëna që duhet lexuar në atë moment ose pak sekonda më vonë. Nëse lihen hapsira të zbrazta(interleav) krau kalon vetëm hapsira e zbrazta deri sa OS i përpunon të dhënat paraprake. **2**.*Read ahead* – Lexohen disa të blloqe nga disku dhe vendosen në *bufferin* e kontrollerit të disku, blloqet e lexuara parashukohen nga OS, jo gjithmon parashikimet janë të sakta.

*SSD* ska koka magnetike, qasja është direkte duke përodrur numrin e bllokut. Lexim është shumë i shpejtë, shkrimi është më i ngadal se leximi shkaku se nuk mund ta bëj mbishkrimin direkte më vonë largohen mbeturinat. Blloqet e SSD kanë jetëgjatësi të shkurtë dhe për këtë përdoren mekanizma si lëvizja e të dhënave nëpër blloqe për ta kursyer jetë gjatësin e blloqeve.

Nëse duam hapsira me jetë të gjatë, të lira dhe të ngadalta mund të përdorim hapsirat terciare si : Tapes(shiritat) dhe disqet optike.

*RAID* (Redudant array of idepedent disk) – Mund të implementohet në dy forma softuerike dhe harduerike. Nëse e implementojmë në mënyrë softuerike duhet pasur një disk jashta *array* të cilën e përdorim për OS dhe me anë të OS ne e konfigurojmë RAID, nëse dëmtohet disku ku gjendet OS të gjitha të dhënat në RAID do të dëmtohen. Kjo mënyrë ka kosto më të ultë dhe nuk ka nevoj për kartel. Nëse e implementojmë me haruder na duhet edhe një *RAID Kartel*, e cila është e veqant për secilin nivel të RAID që duam ta aplikojmë. Këto kartela na ofrojnë një konzollë e cila hapet vetëm me ndihmën e BIOS pa pasur nevoj për OS. Në kët konzolë mund të implementojmë RAID në disqet që i zgjedhim.

Llojet e RAID : RAID-0 – ndryshe njihet si *striped* dy disqe trajtohen si një disk, bë të dy disket shkruhet dhe lexohet një kohësisht. RAID-1 – Ndryshe njihet si *mirroring* të dhënat ruhen backup nga njëri disk në tjetrin, vetëm njëri disk përdoret realisht tjt mbushet automatikisht me të njëjtat të dhëna. RAID-5 : Ruhen parity bit në secilin disk, me këtë bit mund të kthehen të dhënat e humbura nga njëri disk i cili mund të dështoj, toleron vetëm një prishje të disku. Mund ta përdorim 2/3 e disku për të dhëna 1/3 përdoret për parity data.

*Hot-spare* disk janë disqe që nuk alokohen por rrin të gatshme për të zëvendësuar ndonjë disk i cili mund të dështoj.

X+Y = Z (Z – parity bit) Nëse humbin X mund ta fitojmë me X = Z – Y.