

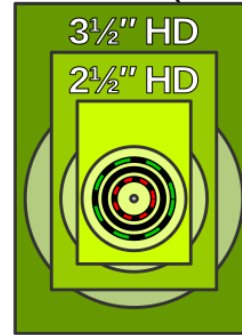


OPERATING SYSTEMS

Disk scheduling algorithms

Disk

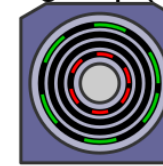
5¼" hard drive (CAV)



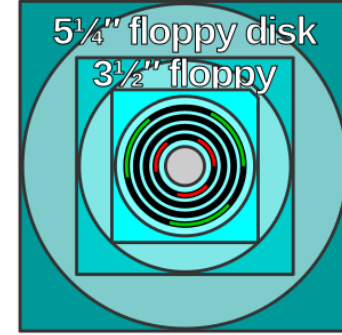
MiniDisc (CLV)



Iomega Zip (CAV)



8" floppy disk (CAV)



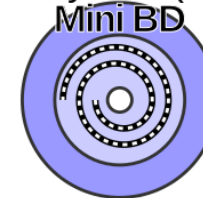
Audio Compact Disc (CLV*)



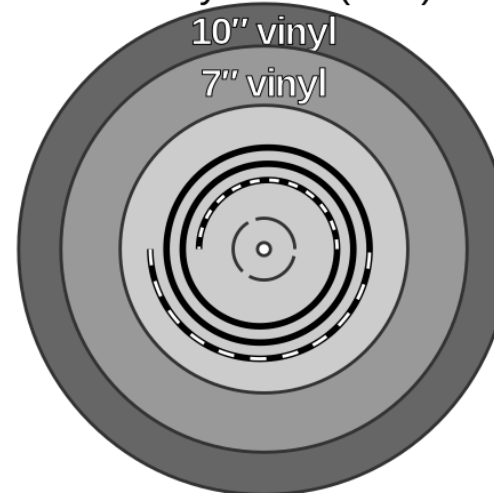
Digital Versatile Disc (CLV*)



Blu-ray Disc (CLV)



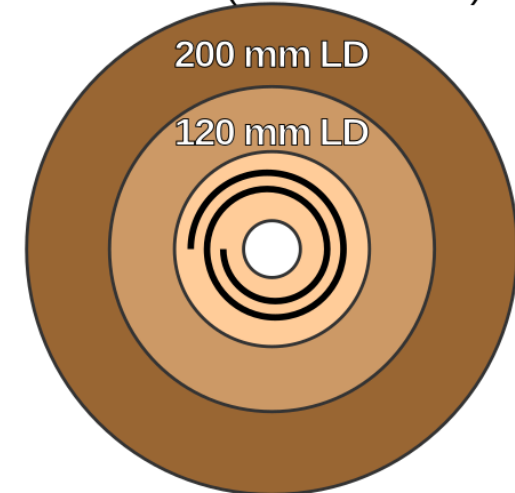
12" vinyl record (CAV)



10" vinyl

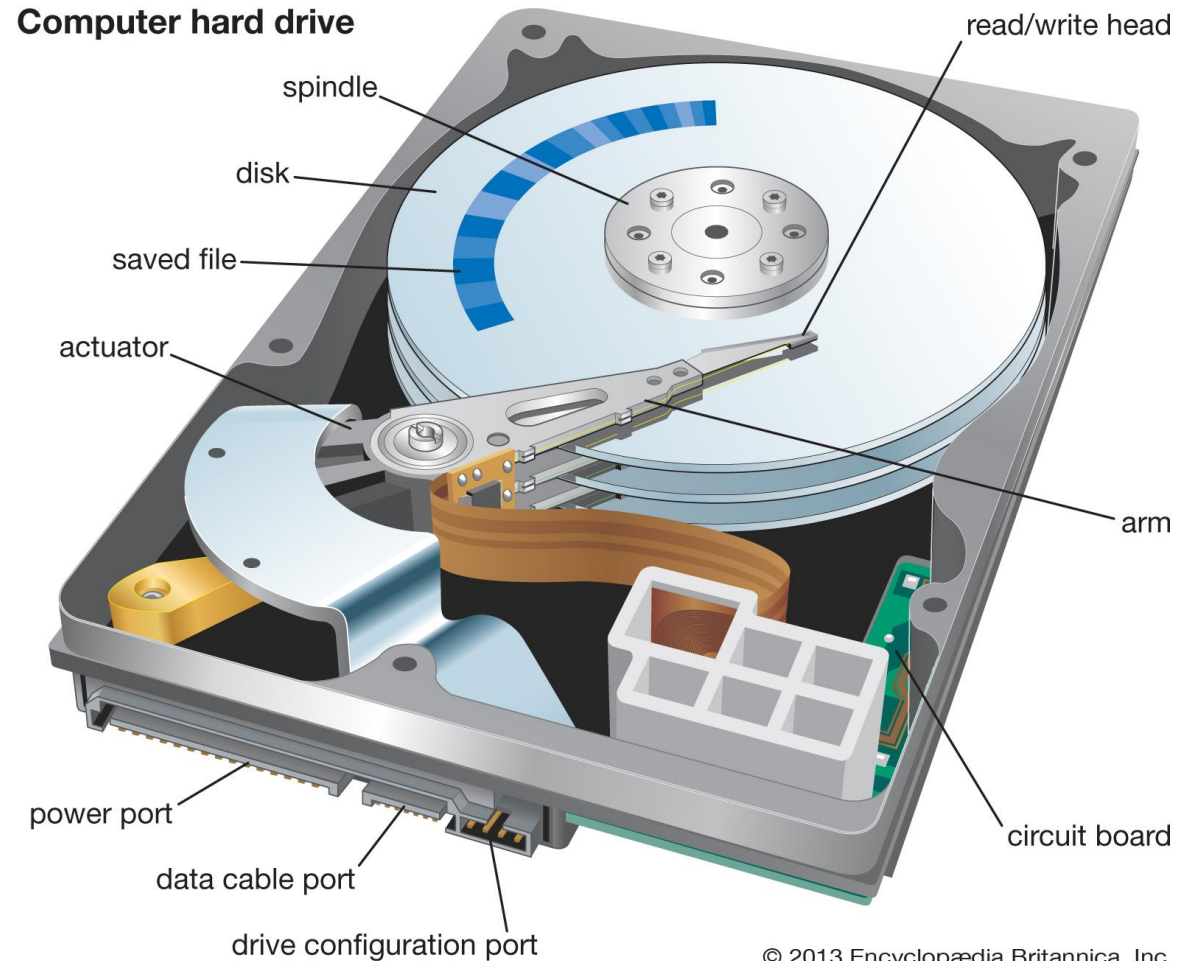
7" vinyl

LaserDisc (CLV/CAA/CAV)



Disk Parts

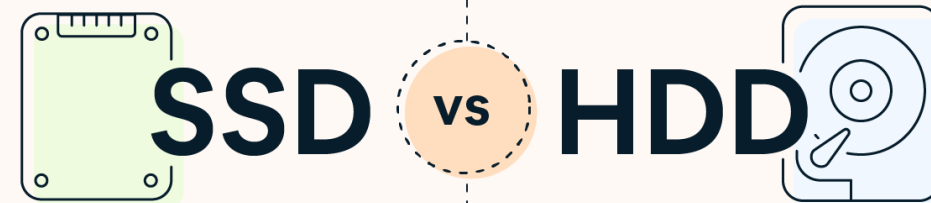
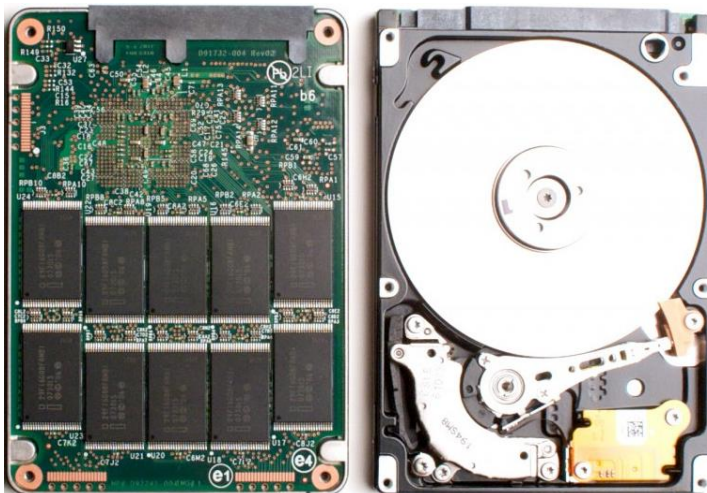
- One or more round plates coated with magnetic material
 - spinning at a constant speed with the magnetic head mechanism
 - which can be moved approximately radially above the plates
- Control assembly, interface to the electromechanical part
- System Bus Interface



© 2013 Encyclopædia Britannica, Inc.

HDD vs. SSD

- Hard disk drive
 - Use Disks
- Solid-state drive
 - Use flash memory (chips)



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

Source: Avast

Algorithms for scheduling disk jobs:

- FCFS
- SSTF
- SCAN (elevator)
- LOOK
- C-SCAN (one-way elevator)
- C-LOOK
- **Seek distance** – search time
 - the time it takes to move the hand (arm) of the disk to a specific path where the data should be read or written.
- Reference String - Data Structure
 - a list of sector numbers to which different tasks approach.

Why do we need algorithms?

- Multiple I/O requests can be made by different **processes** and only one I/O request can be served by a disk controller at a time. Therefore, other I/O requests must wait in the queue and should be scheduled.
- Two or more requests can be away from each other, so they can result in a larger shift of the disk arm.
- Hard drives are one of the slowest parts of the computer system and therefore need to be accessed in an effective way.

FCFS – First Come First Serve

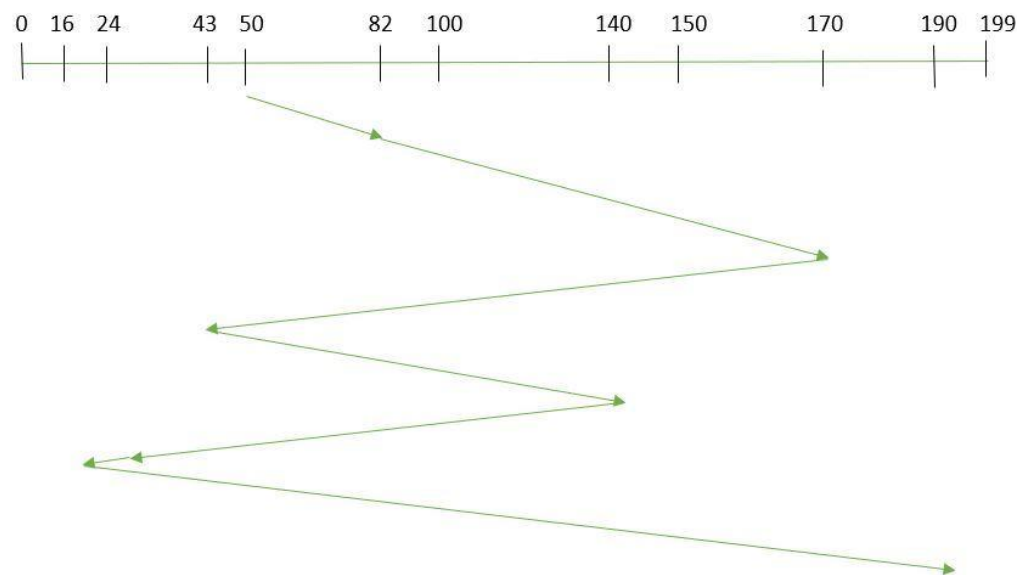
- Aka. FIFO
- Tasks are granted access to the disk in the order in which they arrive
- **Advantages:**
 - No starvation (all will be served)
 - Easy to implement
- **Disadvantage:**
 - Accidental disk access
 - No time savings

FCFS - task

- Reference String: **82,170,43,140,24,16,190**
- The disc head is currently on the sector: **50**

Korak	Trenutna zadaca	Sljedeća zadaca	Razlika
1	50	82	32
2	82	170	88
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

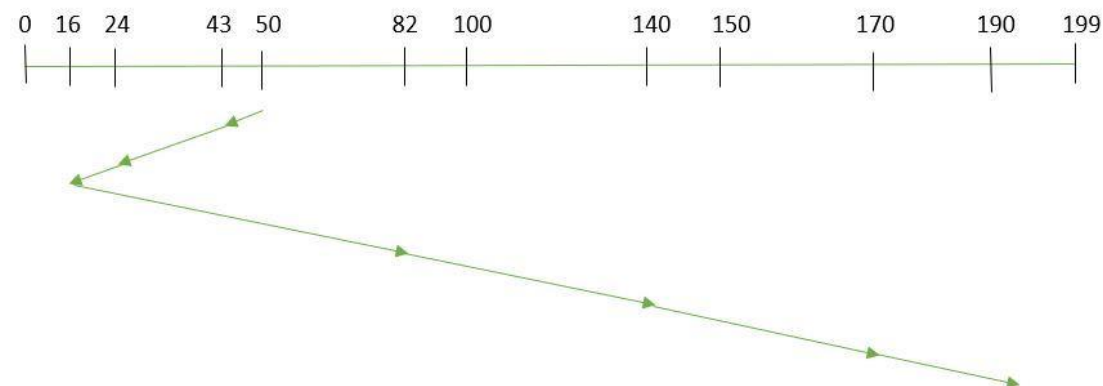
- Minimum Search Time
- Algorithm:
 - When the new task comes, calculate the distance from the task currently accessing the disk
 - Place the task in the appropriate place in the queue (sorted by distance)
 - When the task is complete, select the **nearest**
 - It must always start from one task, and we will take that first position in a row or where the head is currently located

SSTF - task

- Reference String: **82,170,43,140,24,16,190**
- The disc head is on: **50**

Korak	Trenutna zadaca	Sljedeća zadaca	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



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

SSTF – pros/cons

- Advantages:

- The average seek time is decreasing
- Throughput increases

- Disadvantages:

- Overhead to calculate seek time in advance
- Can cause Starvation for a request if it has higher seek time as compared to incoming requests
- High variance of response time as SSTF favours only some requests

SCAN

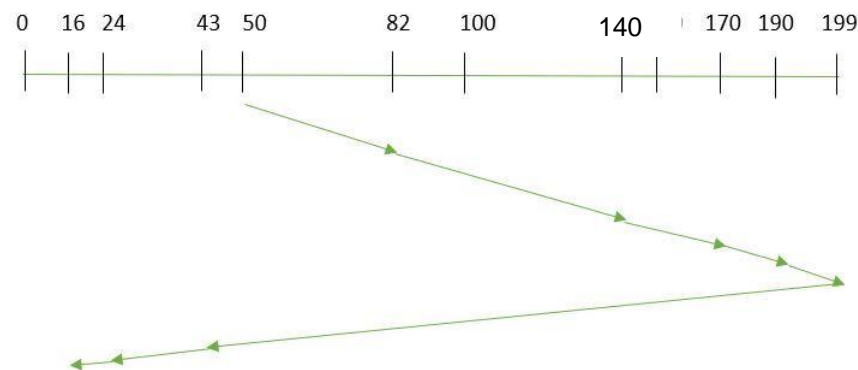
- Aka. „Elevator”
 - When the new task comes, calculate the distance from the task currently accessing the disk
 - Place the task in the appropriate place in the queue (sorted by distance)
 - When the task is over, select the nearest **one way**
 - When we finish the one-way trip we **go back in the opposite direction**

SCAN - zadatak

- Reference String: **82,170,43,140,24,16,190**
- The disc head is on: **50**
 - The first sector of the disk is on **0**, and the last in the **199th** sector

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

SUM: 332



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

Quick Computing: $ABS(50-199) + ABS(199-16) = 332$

SCAN – pros/cons

- Advantages:

- High throughput
- Low variance of response time
- Average response time

- Disadvantage:

- The problem with fairness, the tasks that access the middle of the disk will be served twice as quickly as the rest
- Long waiting time for requests for locations just visited by disk arm

LOCK

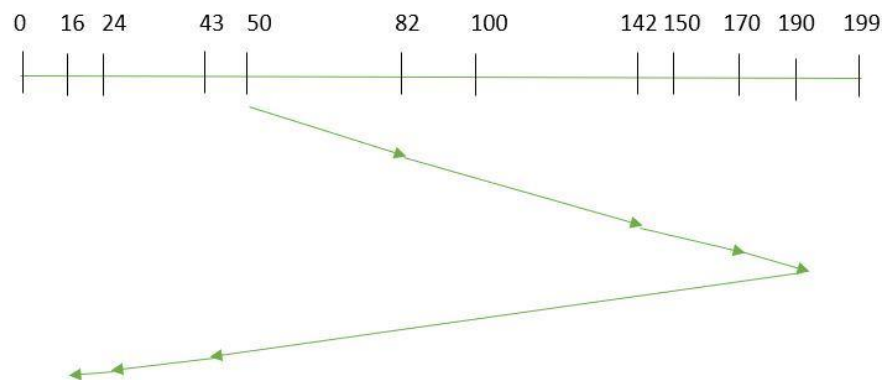
- Same as SCAN but **ignores** the first and last sectors on the disk (if not in the reference string)

LOCK - task

- Reference String: **82,170,43,140,24,16,190**
- The disc head is on: **50**
 - The first sector of the disk is on **0**, and the last in the **199th** sector

Korak	Trenutna zadaca	Sljedeća zadaca	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



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

Quick Computing: $ABS(50-190) + ABS(190-16) = 314$

LOCK – pros/cons

- Advantages:

- Allows for a more uniform wait time compared to SCAN

- Disadvantage (the same as SCAN):

- The problem with fairness, the tasks that access the middle of the disk will be served twice as quickly as the rest
- Long waiting time for requests for locations just visited by disk arm

C-SCAN (one-way elevator)

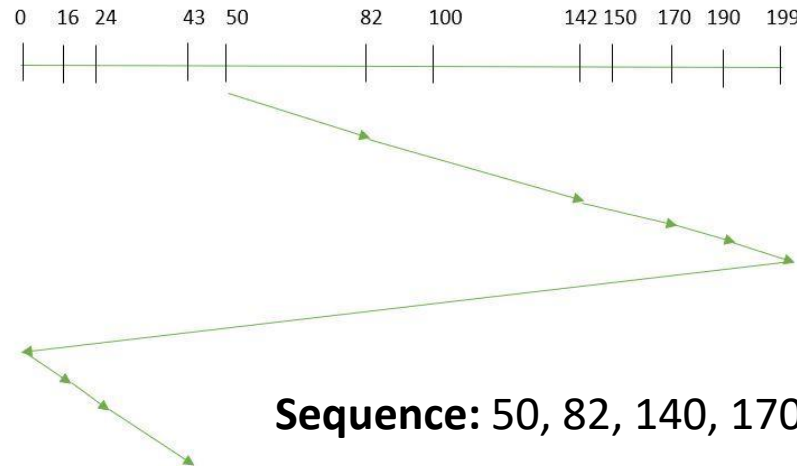
- Circular SCAN
- One-way elevator – the reading head only goes **in one direction**, when it reaches the **end – it goes to the beginning**

C-SCAN

- Reference String: **82,170,43,140,24,16,190**
- The disc head is on: **50**
 - The first sector of the disk is on **0**, and the last in the **199th** sector

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	0	199
7	0	16	16
8	16	24	8
9	24	43	19

SUM: 391



Quick Computing: $ABS(50-199) + ABS(199-0) + ABS(0-43) = 391$

C-SCAN – prednosti/mane

- **Advantages:**
 - Improving weather compared to FIFO
 - No starvation
 - Addressed issue of serving the middle of the disk more frequently
- **Disadvantage:**
 - Long wait time for requests for locations just visited by the disk

C-LOOK

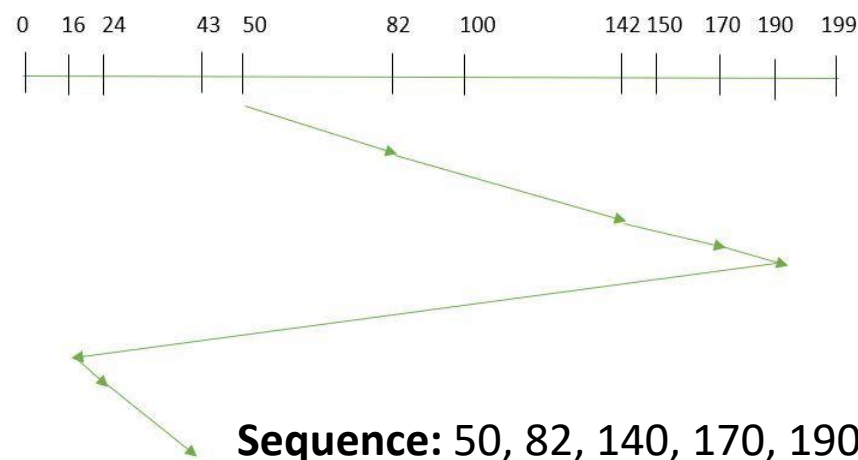
- It does a table like C-SCAN with the exception that it **does not visit the first or last sector** if it is not in the reference string.

C-LOOK - zadatak

- Reference String: **82,170,43,140,24,16,190**
- The disc head is on: **50**
 - The first sector of the disk is on **0**, and the last in the **199th** sector

Korak	Trenutna zadaca	Sljedeća zadaca	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**



Quick Computing: $ABS(50-190) + ABS(190-16) + ABS(16+43) = 341$

C-LOCK – prednosti/mane

- **Advantages:**
 - Same as C-SCAN
 - It doesn't have to go to the beginning/end of the disk
- **Disadvantage:**
 - Long wait time for requests for locations just visited by the disk

Comparison

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

- SSTF – **starvation** (others do not have this disadvantage)
- LOOK & C-LOOK do not return the reading head/arm to the beginning/end
- LOOK & SCAN – fairness (the data in the middle of the disk will be served rather than those at the end)
- LOOK is similar to SCAN algorithm, as is CLOOK similar to CSCAN

Pitanje

Consider an operating system capable of loading and executing one consecutive user process at a time.

The algorithm used to arrange the disk head is First Come First Served (FCFS).

If FCFS is replaced with the shortest search time (SSTF), which the supplier claims gives **50% better comparison results**, what is the expected improvement in user program I/O performance?

Odaberite odgovor:

- A) More than 50%
- B) 50%
- C) Less than 50%
- D) 0%

Answer:

D) 0%

Since the operating system can perform **one successive** user process at a time (1 processor with 1 core/core), the disk is always accessed in an **FCFS** way. The OS never has the choice to choose an IO from multiple IOs because it always takes one at a time...

Single-processor vs. Multi-processor
Single-core vs. Multiple-core
(accessing the disk is a process!)

Single-processor – home computers, mobile devices (1-2 or more core)
Multi-processor - blade servers

[illegible]

Task

- Reference String: 12, 11, 12, 11, 10, 5, 35, 11, 10, 1, 35, 30, 29, 12, 11, 14, 15, 10, 30, 31, 35, 1, 2, 3, 4
- The first sector is 1, the last sector is sector 35
- The disc head is on the sector: 12
- Use algorithms:
 - A) FIFO
 - B) SSTF
 - C) SCAN
 - D) LOOK
 - E) C-SCAN
 - F) C-LOOK
- For each algorithm make a table with steps and calculate seek distance.

Solutions:

- A) FIFO = 202
- B) SSTF = 55
- C) SCAN = 57
- D) LOOK = 57
- E) C-SCAN = 67
- F) C-LOOK = 67

**Thank you for your
attention!**

