

??? ?????

“磁盘调度 (Disk Scheduling) 是指，在磁盘 I/O 请求到达时，按照一定的策略来安排磁头的移动，以减少寻道时间，提高磁盘 I/O 效率。”

- 磁盘调度算法的目标是：在满足 I/O 请求的前提下，尽量减少磁头的移动次数，从而减少寻道时间，提高磁盘 I/O 效率。
- 寻道时间 (Seek Time) 是指磁头从当前位置移动到目标位置所需的时间。寻道时间是磁盘 I/O 时间的重要组成部分。

常见的磁盘调度算法有：

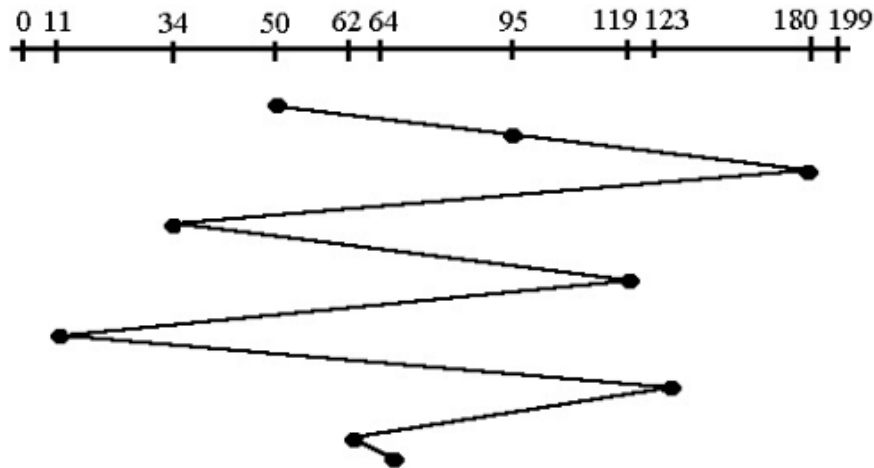
- FCFS (First Come First Served)
- SSTF (Shortest Seek Time First)
- SCAN
- C-SCAN
- LOOK
- C-LOOK
- Eschenbach (电梯调度算法)
- N-step SCAN



1.FCFS(First Come First Served)

假设磁盘 I/O 请求序列为：

- 请求序列：10, 20, 30, 40, 50, 60, 70, 80, 90, 100
- 磁头初始位置：50
- 按照 FCFS 算法，磁头将按照请求序列的顺序依次访问每个请求。
- 寻道时间 (Seek Time) 是指磁头从当前位置移动到目标位置所需的时间。寻道时间是磁盘 I/O 时间的重要组成部分。



※ FCFS(First Come First Served)

01. 디스크 입/출력 요청 대기 큐에 다음과 같은 순서로 기억되어 있다. 현재 헤드가 53에 있을 때, 이들 모두를 처리하기 위한 총 이동 거리는 얼마인가? (단, FCFS 방식을 사용한다.)

대기큐:

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

53

45 85 146 85 108 110 59 2

[정답] 640

- □ □ □ : 53
- □ □ □ : 98, 183, 37, 122, 14, 124, 65, 67
- □ □ □ :

1. $|53 - 98| = 45$
2. $|98 - 183| = 85$
3. $|183 - 37| = 146$
4. $|37 - 122| = 85$
5. $|122 - 14| = 108$
6. $|14 - 124| = 110$
7. $|124 - 65| = 59$
8. $|65 - 67| = 2$

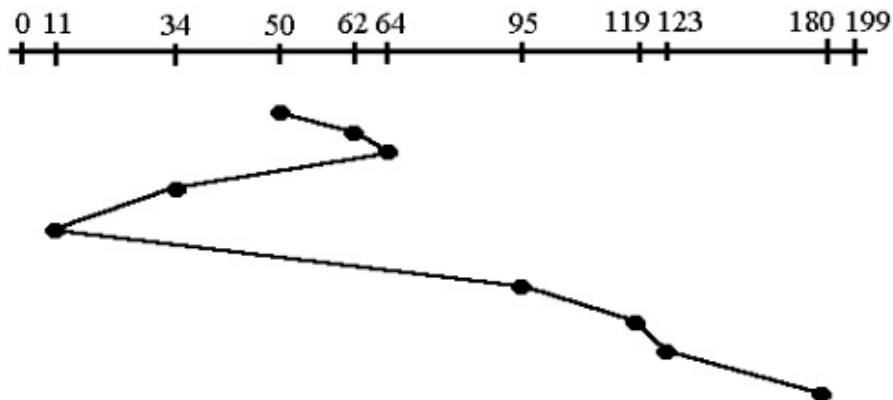
- □ □ □ □ : 640 (□ □ □ □ □ □ □ □ □ □)

$$45 + 85 + 146 + 85 + 108 + 110 + 59 + 2 = 640$$

2.SSTF(Shortest Seek Time First)

□□□ □

- □□ □□□, □□□□□ □□□□ □□□ □□□□□ □□□□ □□
- □□□□ □□□□ □□ □□ □□ □□ □□ □□□□ .
- □□ □ □□□□ □□ □□ □ □□□ □□□□□□ □ □□□□ .



※ SSTF(Shortest Seek Time First)

01. 초기 헤드 위치가 50이며 트랙 0 방향으로 이동 중이다. 디스크 대기 큐에 다음과 같은 순서의 액세스 요청이 대기 중일 때, SSTF 스케줄링을 사용하여 모든 처리를 완료하고자 한다. 가장 먼저 처리되는 트랙을 쓰시오. 초기 위치 50 → 각 요청과의 거리 비교 이동 방향 같은 X. 50보다 작은 쪽 우선
- (단, 가장 안쪽 트랙 0, 가장 바깥쪽 트랙 200)

대기 큐: X 0
100, 180, (40), 120, 0, 130, 55, 80, (51), 200 50과 가장 가까운 트랙인 51 먼저 처리

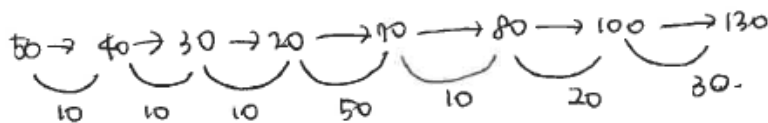
[정답] 51

SSTF □ □□□ .

☐ ☒ ☐

02. 디스크 큐에 다음과 같이 I/O 요청이 들어와 있다. 최소탐색시간 우선(SSTF) 스케줄링 적용 시 발생하는 총 헤드 이동 거리를 구하시오. (단, 추가 I/O 요청은 없다고 가정한다. 디스크 헤드는 0부터 150까지 이동 가능하며, 현재 위치는 50이다.)

80, 20, 100, 30, 70, 130, 40



- $\alpha_0 \alpha_1 \alpha_2$: 50
- $\alpha_3 \alpha_4$: 0 ~ 150 (α_3 SSTF α_4 α_3 α_4)
- α_5 : 80, 20, 100, 30, 70, 130, 40
- α_6 : SSTF
- α_7 : $\alpha_0 \alpha_1 \alpha_2 \alpha_3 \alpha_4 \alpha_5 \alpha_6$

- $\square \square \square \square$: 140

$$10 + 10 + 10 + 50 + 10 + 20 + 30$$

03. 현재 헤드의 위치가 50에 있고 트랙 0번 방향으로 이동하며, 요청 대기 열에는 아래와 같은 순서로 들어 있다고 가정할 때 SSTF(Shortest Seek Time First) 스케줄링 알고리즘에 의한 헤드의 총 이동거리를 구하시오.

대기큐:

100, 180, 40, 120, 0, 130, 70, 80, 150, 200

50 → 40 → 70 → 80 → 100 → 120 → 130 → 150 → 180 → 200 → 0 .
 10 30 10 20 20 10 20 30 20 200 .

[정답] 370

04. 사용자가 요청한 디스크 입·출력 내용이 다음과 같은 순서로 큐에 들어 있을 때 SSTF 스케줄링을 사용한 경우의 처리 순서를 쓰시오.
 (단, 현재 헤드 위치는 53 이고, 제일 안쪽이 1번, 바깥쪽이 200번 트랙이다.)

대기큐:

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

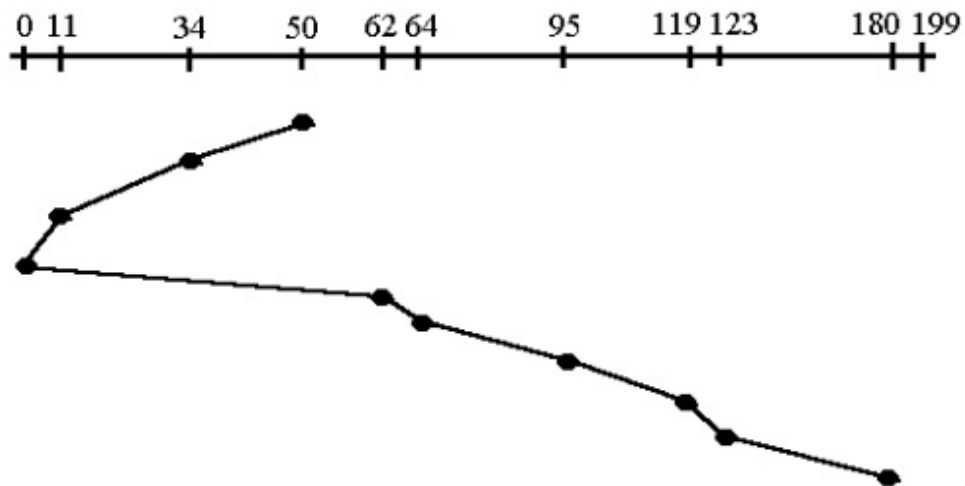
53 → 65 → 67 → 37 → 14 → 98 → 122 → 124 → 183
 12 2 30 23 84 24 2 59

[정답] 53-65-67-37-14-98-122-124-183

3. SCAN(?????? ?????)

□□ □□

- □□ □□□, □□□□□ □□□
- □□ □□ □□□□□ □□□□□□□□□□ □□ ,
- □□ □□□□□□□□□□□□□□ □□ .
- **SCAN** □□□□□□□□□□ (0□□□) □□ □□ (C-SCAN □□□□□□□□□□ .)
- □ : □□ 50□□ → 0 □□□□ □□ □□
 → 50□□ □□ □□□□ □□□□ , 0□□□□□□□□□□□□ □□ □□ 50□□
 □□ □□□□ □□□□ .



01. 디스크 스케줄링에서 SCAN기법을 사용할 경우, 다음과 같은 작업대기 큐의 작업들을 수행하기 위한 헤드의 총 트랙 이동 거리는?
(단, 초기 헤드의 위치는 30이고, 현재 0번 트랙으로 이동 중이다.) 76

대기큐:
7, 46, 15, 38, 3

30 → 15 → 7 → 3 → 0 → 38 → 46
15 8 4 3 38 8

[정답] 76

02. 디스크 스케줄링 기법 중 SCAN을 사용하여 다음 작업대기 큐의 작업을 모두 처리하고자 할 경우, 가장 최후에 처리되는 트랙은?
(단, 현재 디스크 헤드는 50 트랙에서 40 트랙으로 이동해 왔다고 가정한다.) 55

대기큐:
7, 55, 15, 38, 3

50 → 40 → 38 → 15 → 7 → 3 → 0 → 55

[정답] 55

03. 디스크에서 헤드가 70트랙을 처리하고 60트랙으로 이동해 왔다. 디스크 스케줄링 기법으로 SCAN 방식을 사용할 때 다음 디스크 대기큐에서 가장 먼저 처리되는 트랙은? 50

대기큐:
20, 50, 95, 100

70 → 60 → 50

[정답] 50

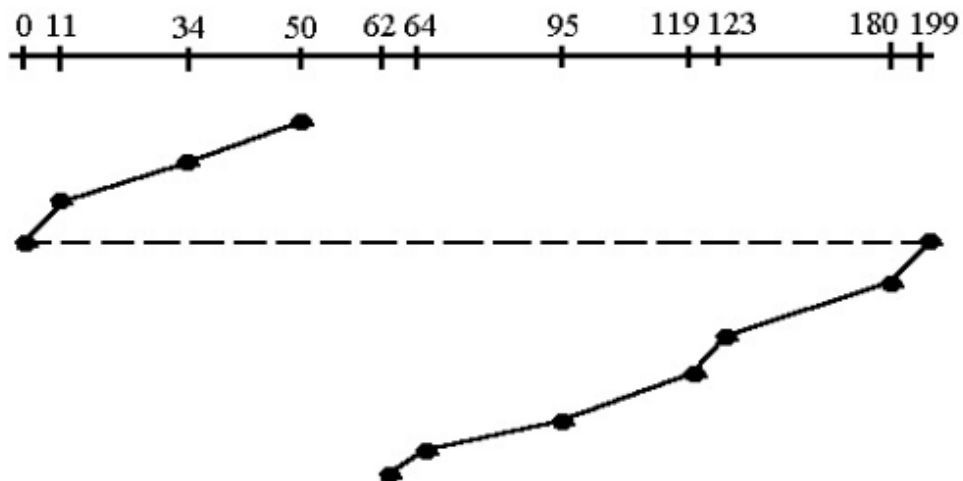
4. C-SCAN(Circular-Scan)

문제

- 0, 11, 34, 50, 62, 64, 95, 119, 123, 180, 199
- C-SCAN 알고리즘은 헤드 위치가 0일 때, 0을 방문한 후 "0"을 방문한 후 11을 방문한다.
- 11을 방문한 후 34, 50, 62, 64, 95, 119, 123, 180, 199를 순서대로 방문한다.
- C-SCAN 알고리즘은 헤드 위치가 199일 때, 199를 방문한 후 0을 방문한다.
- 0을 방문한 후 11을 방문한다.

C-SCAN 알고리즘은 헤드 위치가 "0"일 때 "0"을 방문한다.

- 0을 방문한 후 11을 방문한다.
- 11을 방문한 후 34, 50, 62, 64, 95, 119, 123, 180, 199를 순서대로 방문한다. → 0을 방문한다



01 트랙 번호가 0부터 199인 200개의 트랙을 가진 디스크가 있다. 디스크 스케줄링 기법 중 C-SCAN을 사용하여 다음과 같은 작업 대기 큐(디스크 큐)의 작업을 처리하고자 하는 경우, 처리되는 트랙의 순서를 바르게 나열 하시오.

(단, 현재 디스크 헤드는 트랙 35에서 트랙 47로 이동해 왔다고 가정한다.)

대기큐:

47 → 86 → 139 → 158 → 175 → 199 → 0 → 13 → 22 [정답] 47 → 86 → 139 → 158 → 175 → 199 → 0 → 13 → 22

02. 현재 헤드의 위치가 50에 있고, 요청 대기열의 순서가 다음과 같을 경우, C-SCAN 스케줄링 알고리즘에 의한 헤드의 총 이동거리는 얼마인가?
(단, 현재 헤더의 이동 방향은 안쪽이며, 안쪽의 위치는 0으로 가정한다.)

대기큐:

50 → 40 → 0 → 200 → 180 → 150 → 130 → 100 → 80 → 70

10 40 200 20 80 20 80 20 10

03. 표의 내용은 0 ~ 199번의 200개 트랙으로 이루어진 디스크 시스템에서, 큐에 저장된 일련의 입출력 요청들과 어떤 디스크 스케줄링(disk scheduling) 방식에 의해 처리된 서비스 순서이다. 이 디스크 스케줄링 방식을 쓰시오.
(단, 표의 숫자는 입출력할 디스크 블록들이 위치한 트랙 번호를 의미하며, 현재 디스크 헤드의 위치는 트랙 50번이라고 가정한다.)

- 요청 큐 : 99, 182, 35, 121, 12, 125, 64, 66
- 서비스 순서 : 64, 66, 99, 121, 125, 182, 12, 35

50

5. LOOK

□ □ □	□ □
-------	-----

-   
-           

SCAN vs LOOK

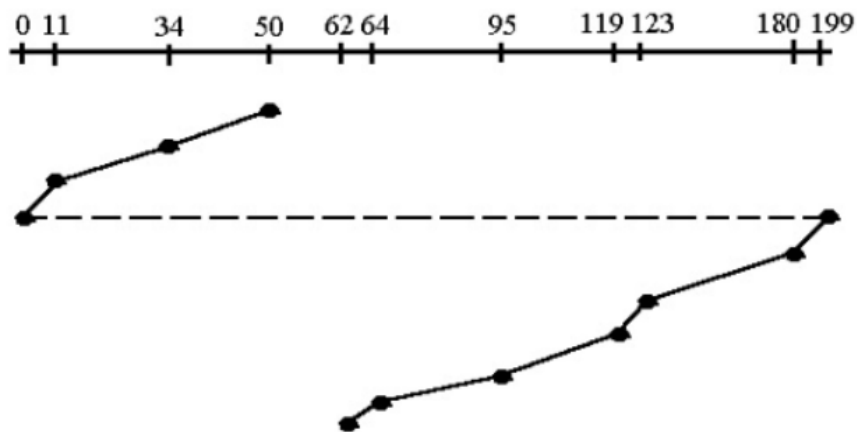
“SCAN 扫描头从磁盘一端向另一端移动时，如果磁头经过磁道时，发现磁道上没有要访问的数据，则磁头继续向另一端移动，直到遇到下一个要访问的数据为止。这种扫描方式称为 SCAN 扫描方式。

LOOK vs C-SCAN

“LOOK 扫描头从磁盘一端向另一端移动时，如果磁头经过磁道时，发现磁道上没有要访问的数据，则磁头继续向另一端移动，直到遇到下一个要访问的数据为止。这种扫描方式称为 LOOK 扫描方式。C-SCAN 扫描头从磁盘一端向另一端移动时，如果磁头经过磁道时，发现磁道上没有要访问的数据，则磁头继续向另一端移动，直到遇到下一个要访问的数据为止。这种扫描方式称为 C-SCAN 扫描方式。

LOOK vs C-LOOK

“LOOK 扫描头从磁盘一端向另一端移动时，如果磁头经过磁道时，发现磁道上没有要访问的数据，则磁头继续向另一端移动，直到遇到下一个要访问的数据为止。这种扫描方式称为 LOOK 扫描方式。C-LOOK 扫描头从磁盘一端向另一端移动时，如果磁头经过磁道时，发现磁道上没有要访问的数据，则磁头继续向另一端移动，直到遇到下一个要访问的数据为止。这种扫描方式称为 C-LOOK 扫描方式。



01. 디스크 스케줄링 방법 중 LOOK 방식을 사용할 때 현재 헤드가 60에서 50으로 이동해 왔다고 가정할 경우 다음과 같은 디스크 큐에서 가장 먼저 처리되는 것은?

70

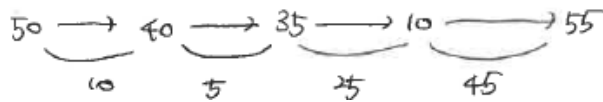
대기큐:
70, 80, 100, 90

[정답] 70

02. 다음과 같은 트랙이 요청되어 큐에 도착하였다. 모든 트랙을 서비스하기 위하여 LOOK 스케줄링 기법이 사용되었을 때 모두 몇 트랙의 헤드 이동이 생기는가?
(단, 현재 헤드의 위치는 50 트랙 이고 헤드는 트랙 0 방향으로 움직이고 있다.)

85

대기큐:
10, 40, 55, 35



[정답] 85

6. C-LOOK

□□ □□

- □□ □□ □□ □□ □□ □□□□ .
- □□□ □□ □□□□ □□ □□ □□□□□□ □□ .
- □□□ □□□ □□ □□ , □□□ □□ □□□ □□□□ .
- □□ □ □□□ □□□ □□□ □□□□□□□□□□□□ □□ □□ □□ □□□□ □□□□ .
- □ , □□□ □□□ □□ , □□□ □□ □□□ □□ □□ □□□□ □□□□ .

[illegible]

8. N-setp SCAN

“SCAN 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 ,
SCAN 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 0000 0000 0000



- $100 \text{ N} - 100 \quad 100 \quad 100$