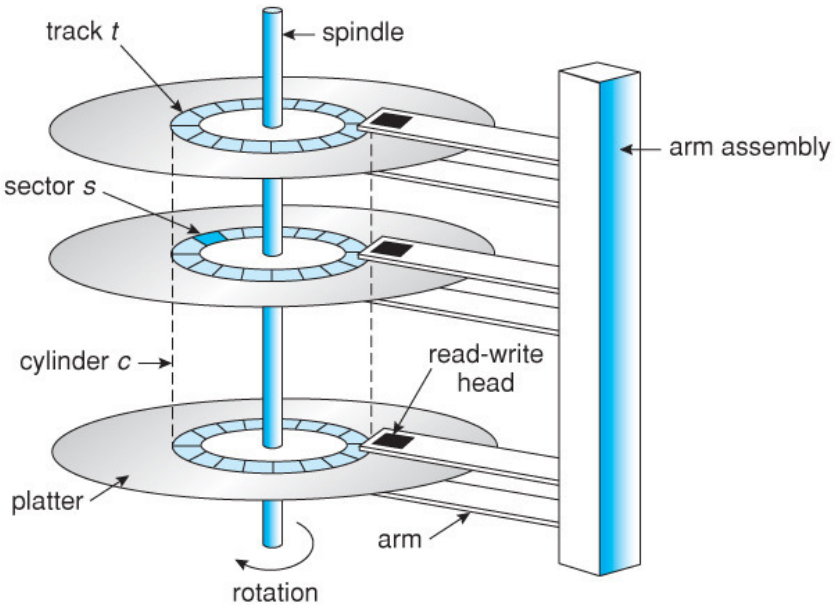
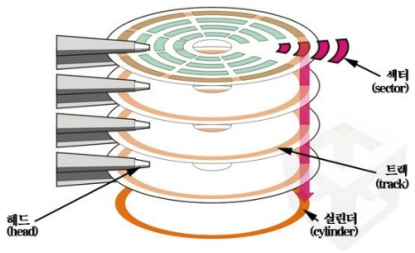
디스크 스케줄링

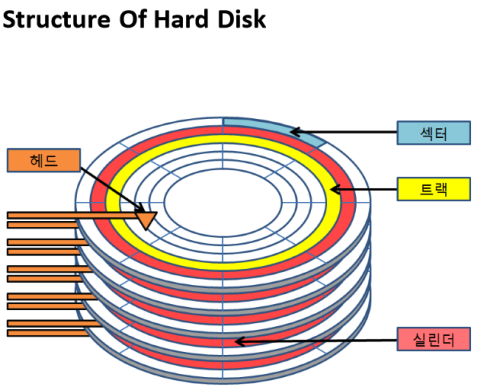
태그 운영체제

디스크 스케줄링

디스크의 구조







헤드(head)

- 디스크 암에 붙어있는 가벼운 물체
- 데이터를 읽거나 쓸 때 사용

플래터(platter)

- 0, 1의 데이터 저장
    - 표면의 자성체와 자기 이용

섹터(sector)

- 하드 디스크의 가장 작은 저장 단위
- 트랙의 일부를 작은 단위로 쪼갠 것

블록(block)

- 논리적인 개념에서의 가장 작은 단위
- 데이터를 전송하는 단위, 메모리에서는 블록마다 주소가 배정

⇒ 하드디스크에서는 섹터가 가장 작은 저장단위, 운영체제에서는 블록이 가장 작은 저장 단위

트랙(track)

- 플래터에서 회전축을 중심으로 데이터가 기록되는 동심원
    - 동심원 상의 섹터 집합

실린더(sylinder)

- 디스크 암의 헤드들은 각 플래터의 같은 위치의 트랙을 읽거나 쓰게됨. 이 때의 트랙 집합을 의미

목적

스케줄링은 트랙의 이동을 최소화하여 **탐색 시간**을 최소화하기 위해 수행하며 3가지의 목적을 가짐

- **처리량 최대화** : 일정 시간에 디스크 입출력 요구를 서비스 해주는 수를 최대화한다.
- **응답 시간의 최소화** : 어떤 요청이 있은 후 결과가 나올 때까지 걸리는 시간을 최소화한다.
- **응답 시간 편차의 최소화** : 각 요청의 응답 시간과 평균 응답 시간의 편차를 최소화한다.

디스크 접근 시간

탐색시간(seek time) + 회전 지연 시간(rotational delay) + 데이터 전송 시간(transfer time)

- 탐색시간
  - 현재 위치에서 목표 트랙까지 이동하는 시간
- 회전 지연 시간
- 데이터 전송 시간

디스크 스케줄링의 종류

비교를 위한 예시

- 성능 비교 지표: **트랙의 총 이동 거리**
- 트랙은 `0` ~ `24` 번까지 총 **25**개
- 트랙 접근 순서 표

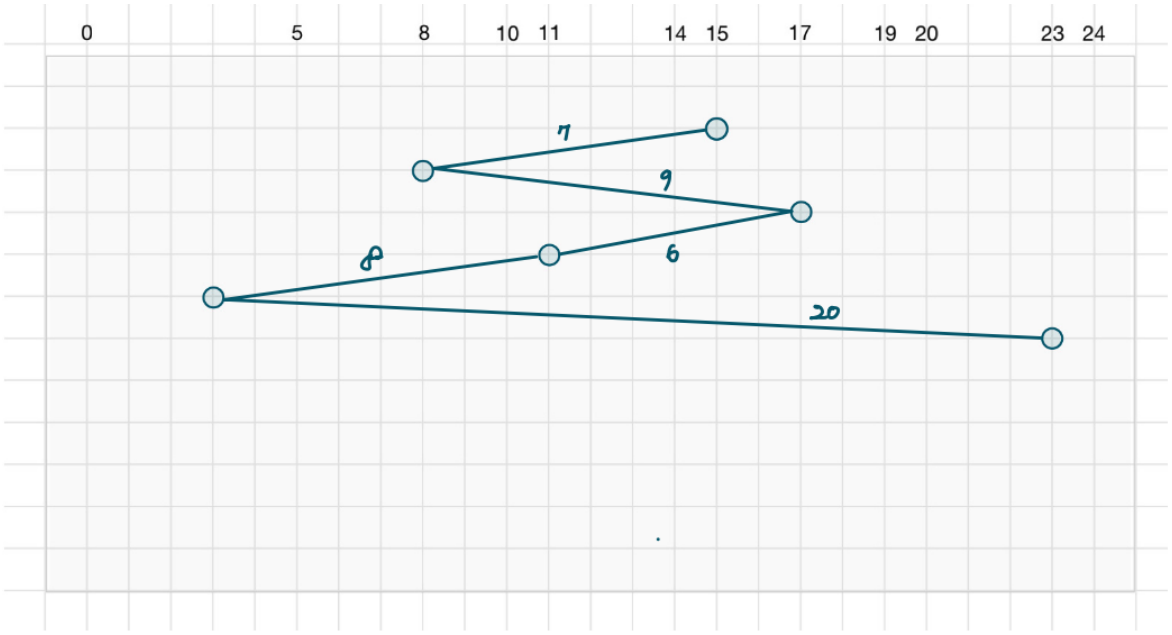
순번	1	2	3	4	5	6
트랙번호	15	8	17	11	3	23

스케줄링 종류	탐색시간
FCFS	50
SSTF	37
Block SSTF	42
SCAN	39
C-SCAN	46
LOOK	32
C-LOOK	38

해당 예시에서만 봤을 때, 탐색시간
LOOK < SSTF < C-LOOK < SCAN < Block SSTF < C-SCAN < FCFS

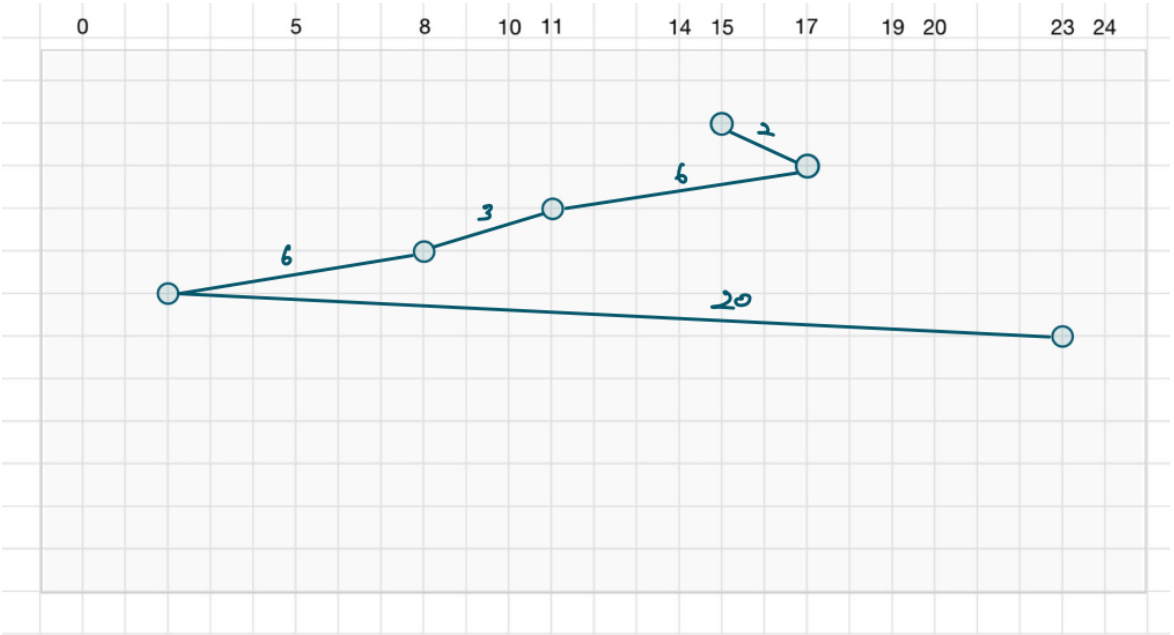
1. FCFS(First Come, First Service)

- 가장 단순한 디스크 스케줄링 방식
- **요청이 들어온 트랙 순** 서비스
- 특별한 기법 사용 X



2. SSTF(Shortest Seek Time First)

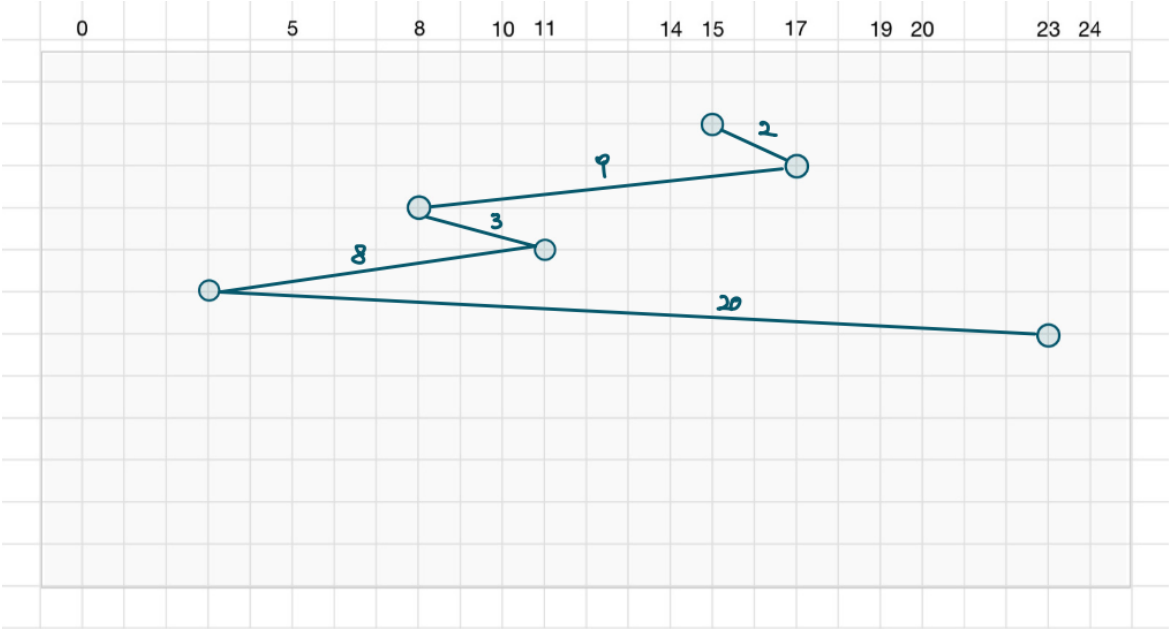
- 현재 헤드가 있는 위치에서 **가장 가까운 트랙부터 서비스**
  - 가장 가까운 트랙 2개 이상이라면, 먼저 요청받은 트랙을 서비스
- 효율성 높음
- **기아 현상(starvation)** 발생 가능성
  - 헤드가 중간에 위치하면 가장 안쪽이나 가장 바깥쪽에 있는 트랙을 서비스 받을 확률이 낮아짐
- 공평성 위배로 인해, SSTF 디스크 스케줄링은 잘 사용되지 않음
- 예전에는 일괄 작업 시스템처럼 다른 프로세스에 영향을 주지 않고 빠른 응답 성능을 필요로 하는 시스템에 사용되기도 했음



3. 블록 SSTF(block Shortest Seek Time First)

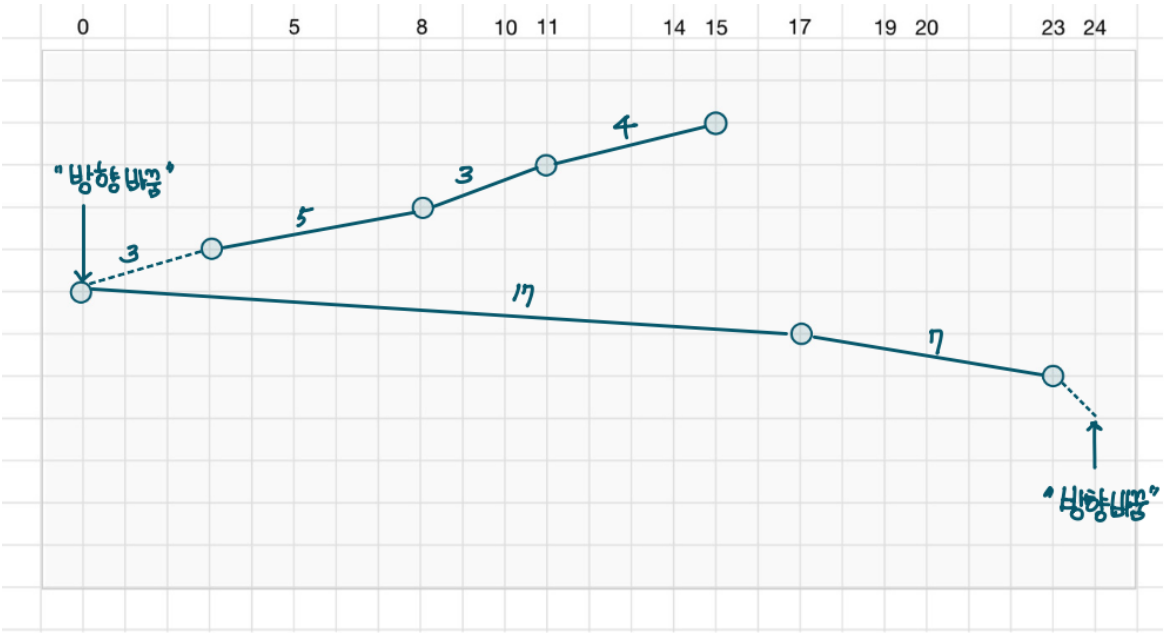
- SSTF의 공평성 위배 문제를 어느정도 해결한 방법

- 큐에 있는 트랙 요청을 일정한 **블록형태**로 묶음
  - SSTF에서 가장 먼 트랙은 큐의 맨 끝으로 이동
  - 블록 SSTF에서는 블록안에서 비교하여, 블록에서 가장 먼 트랙은 블록의 끝으로 이동
    - 몇 번만 양보하면 서비스를 받을 수 있음
  - **에이징**을 적용한 것
- 성능은 FCFS만큼 좋지 않음



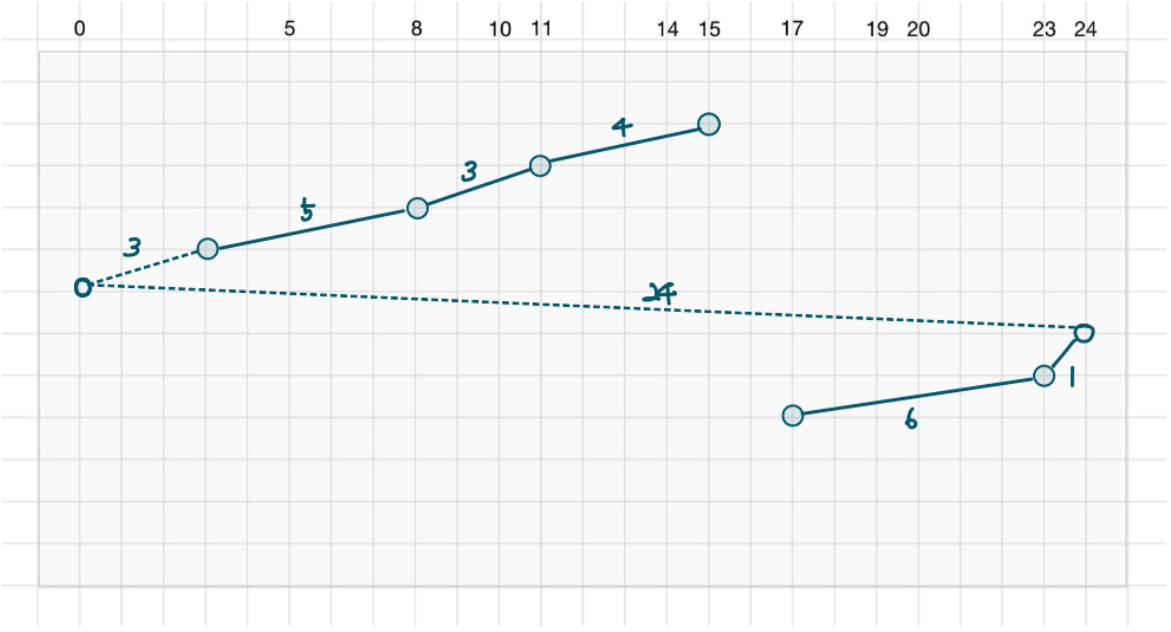
4. SCAN(SCAN disk)

- SSTF의 공평성 위배 문제를 완화하기 위해 만들어진 기법
- 헤드가 한 방향으로만 움직이면서 서비스함
  - 한쪽 끝에 도달하면 역방향으로 이동하면서 오는 길에 있는 요청을 처리함
  - 엘리베이터 기법이라고도 함
- 성능은 SSTF보다 좋지 않지만, FCFS보다 좋음
  - 많이 사용되는 기법
- 기아 현상 발생 가능성
  - 동일한 트랙이나 실린더 요청이 연속적으로 발생하면, 헤드가 더 이상 나아가지 못하고 제자리에 머물게 되어 바깥쪽 트랙이 서비스를 받지 못할 수 있음



5. C-SCAN(Circular SCAN)

- SCAN의 공평성 위배
  - 가장 바깥쪽 트랙을 한 번씩 방문하는 동안, 안쪽 트랙은 2번씩 방문하므로
- 헤드가 한쪽 방향으로 움직일 때는 요청받은 트랙을 서비스하고, 반대 방향으로 돌아올 때는 서비스하지 않고 이동
- SCAN 보다 성능이 좋지 않음
- 기아 현상 발생 가능성
  - 동일한 트랙(실린더) 요청이 연속적으로 발생 시, 바깥쪽 트랙 서비스 받지 못함



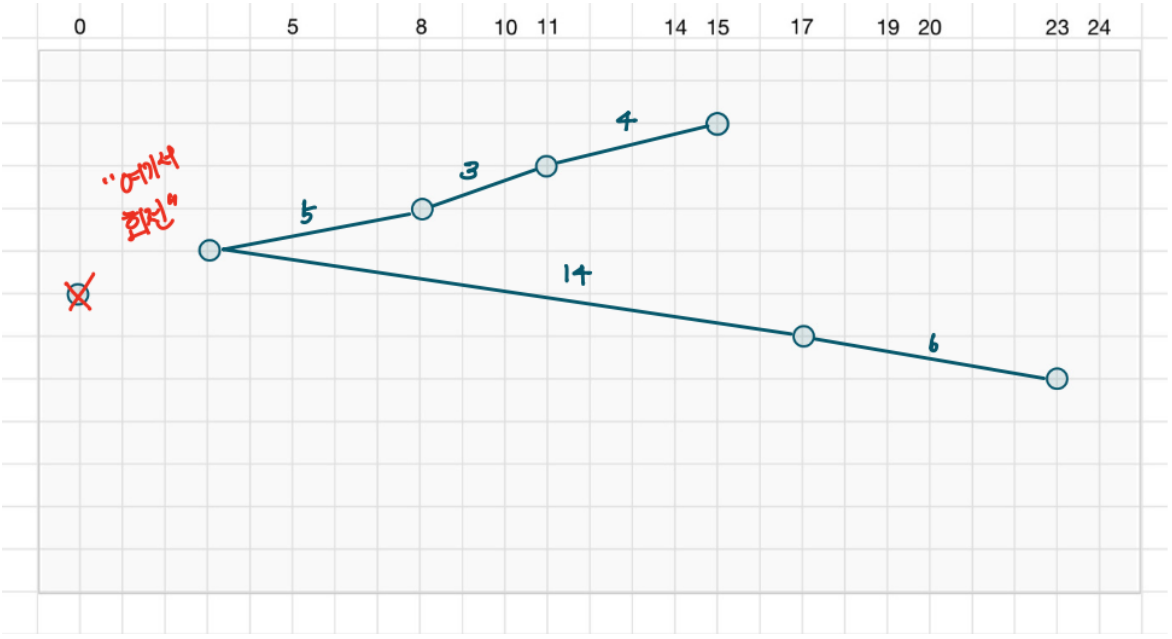
6. N-SCAN(N-step SCAN)

- N-step SCAN은 SCAN 기법의 무한 대기 발생 가능성을 제거한 것으로, 어떤 방향의 진행이 시작될 당시에 대기 중이던 요청들만 서비스하고, 진행 도중 도착한 요청들은 한데 모아서 다음의 반대 방향 진행 때 서비스하는 기법

- SSTF나 SCAN 기법보다 응답 시간의 편차가 적음
- 특정 방향에 많은 수의 요청이 도착할 경우 반대 방향에서의 무한 지연 발생을 방지할 수 있음
- 진행 도중 도착한 요청은 반대 방향 진행 시 서비스하기 위해 디스크 대기 큐에 저장함

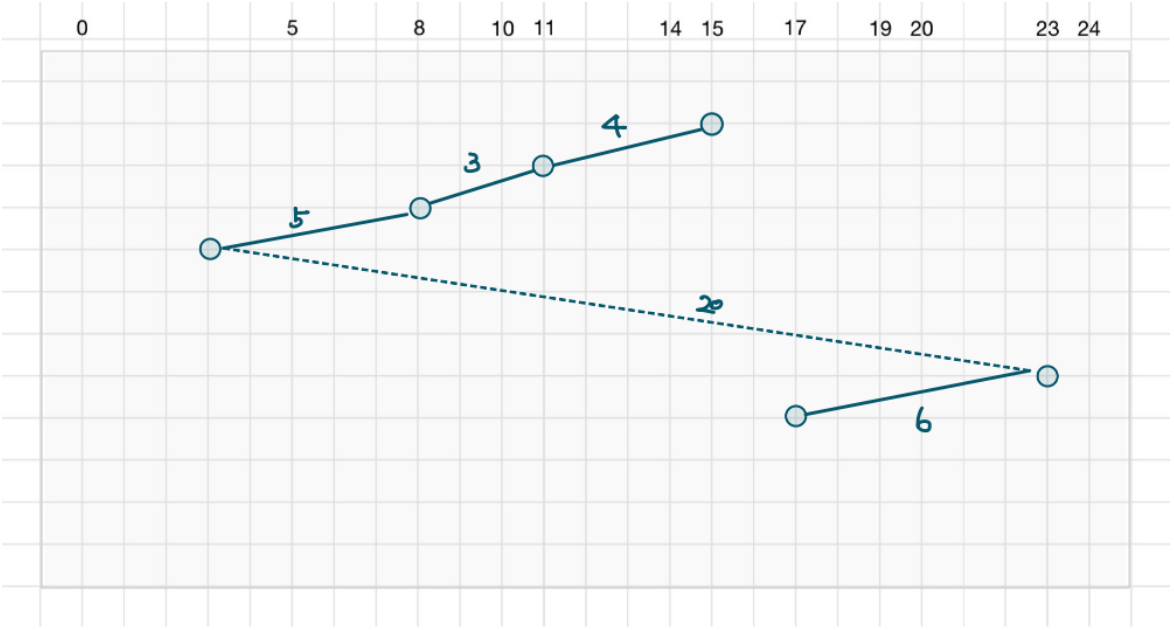
7. LOOK

- SCAN의 불필요한 부분을 제거하여 효율을 높인 기법
  - SCAN: 트랙 요청이 없어도 헤드가 맨 마지막 트랙에 도착한 후에야 방향을 바꿈
- 더 이상 서비스할 트랙이 없으면 헤드가 중간에서 방향을 바꿈
- SCAN보다 성능이 좋음



8. C-LOOK(Circular LOOK)

- 한 쪽 방향으로만 서비스하는 C-SCAN과 비슷하지만, 중간에 방향을 바꿀 수 있음



9. SLTF(Shortest Latency Time First)

- SLTF는 섹터 큐잉(Sector Queuing)이라고 함
- 회전 지연 시간의 최적화를 위해 구현된 기법
- 디스크 대기 큐에 있는 여러 요청을 섹터 위치에 따라 재정렬하고, 가장 가까운 섹터를 먼저 서비스 함
- 헤드의 이동이 거의 없고 고정 헤드 장치인 드럼과 같은 장치에서 사용된다.
    - 비용이 높아 사용되지 않음

참고자료

https://jess2.tistory.com/83

https://wansook0316.github.io/cs/os/2020/04/06/운영체제-정리-20-디스크-스케줄링-알고리즘.html

https://kjhoon0330.tistory.com/entry/운영체제OS-디스크-구조와-디스크-스케줄링

https://velog.io/@icnwpe/운영체제-HDD