4. 디스크 스케줄링(disk scheduling)

디스크 스케줄링은 디스크 장치에서 신속한 자료 입출력에 대한 문제를 다루는 것이다.
여러 사용자들이 디스크의 서로 다른 트랙에 있는 데이터 블록에 대한 자료 입출력을 요구할 때 운영체제는 효율적인 입출력 처리를 할 수 있어야 한다.

1. 선입선출(FCFS; First Come First Served) 스케줄링

① 입출력 요청이 디스크 큐에 들어온 순서대로 서비스하는 **가장 단순한 스케줄링**이다.
② 공평하며, 간단하지만 신속하고 효율적인 서비스를 제공할 수 없다.
③ 트랙 탐색을 위한 최적화 작업이 없다.
④ 디스크에 대한 입출력 요청이 적을 때 사용할 수 있는 스케줄링 기법이다.

[예] 현재 헤드 위치가 실린더 50일 때

요청 큐에 들어온 순서 : 80, 60, 0, 100, 70
헤드 이동거리 : 30 + 20 + 60 + 100 + 30 = **240**
→ 즉, 헤드가 240개의 실린더를 왔다갔다 가로질렀다는 것이다.

2. 최소탐색시간우선(SSTF; Shortest Seek Time First) 스케줄링

① 현재 헤드 위치에서 **탐색거리가 가장 짧은 입출력 요청**을 우선적으로 처리한다.
② 새로운 요청이 들어오게 되면 이를 포함시켜서 입출력 서비스를 한다.
→ 어떤 요청은 무한정 기다리게 되는 **기아(starvation)** 현상이 일어날 수 있다.
→ 하지만, FCFS보다 평균적으로 우수하다(짧은 응답시간, 많은 처리량)
③ 가운데 트랙이 보다 많은 서비스를 받게 되는 **탐색 편중 현상**이 일어난다.
④ 응답시간 편차로 대형 시스템에는 적합하지 않다.

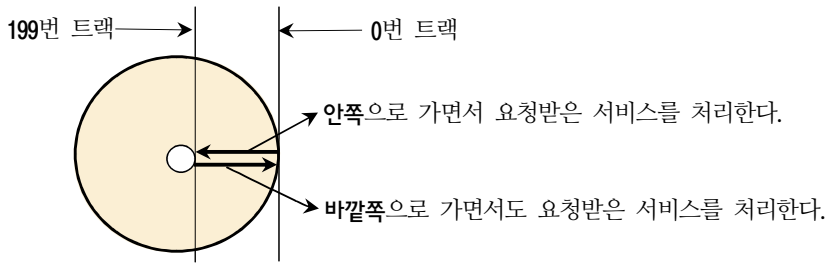
[예] 현재 헤드 위치가 실린더 50일 때 서비스 순서

요청 큐에 들어온 순서 : 80, 90, 0, 45, 70, 100, 30, 54
서비스 순서 : 54, 45, 30, 0, 70, 80, 90, 100
헤드 이동거리 : 4 + 9 + 15 + 30 + 70 + 10 + 10 + 10 = **158**

• SSTF는 FCFS보다는 개선된 알고리즘이지만 매우 좋은 알고리즘은 아니다.

3. SCAN 스케줄링

헤드가 트랙을 "왔다갔다" 하면서 입출력 요청이 있는 트랙에 대한 요구를 서비스한다.

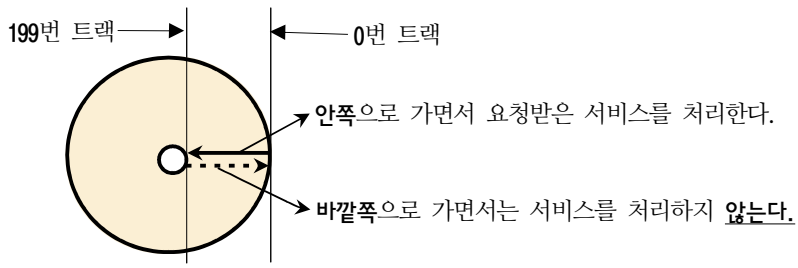


---

① 일명, 엘리베이터 알고리즘이다. → 엘리베이터처럼 **"왔다갔다"** 하면서 서비스를 처리한다.
② SCAN 기법에서 헤드 이동은 트랙 처음부터 마지막까지 이동한다.(서비스 요청이 없어도)
→ 즉, 헤드 이동은 트랙 0에서 199까지 이동된다.
③ SCAN은 SSTF에서 발생되는 트랙에 대한 차별이 없다.("왔다갔다" 하면서 서비스하므로)
④ 하지만, SCAN도 최적의 알고리즘은 아니다.
→ 서비스 진행 중에 입출력 요청이 많으면, 방금 지나간 트랙에 대한 서비스는 지연된다.

4. C-SCAN(Circular-SCAN, 순환-SCAN) 스케줄링

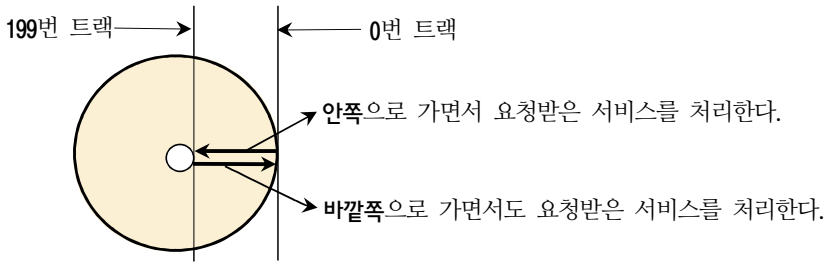
C-SCAN은 한쪽 방향으로 가면서만 입출력 요청에 대한 서비스를 처리한다.
C-SCAN은 반대 방향으로 가면서는 입출력 요청에 대한 서비스를 처리하지 않는다.



---

① SCAN을 변형한 기법으로 각 트랙(요청)에 대한 균등한 대기시간을 제공할 수 있다.
② C-SCAN은 SCAN처럼 헤드 이동은 트랙 0에서 199까지 이동된다.

[예제] 디스크에 대한 입출력 요청 서비스 운행 중이다. 현재, 디스크의 서비스 요청 대기 큐에 도착한 요청이 다음과 같을 때 SCAN 스케줄링 알고리즘에 의한 헤드의 총 이동거리는 얼마인가? (단, 현재 헤드의 위치는 트랙 50에 있고, 헤드의 이동은 0에서 199 방향이다)



요청 대기열의 순서 : 60, 100, 40, 20, 90, 150, 30, 180

---

① 168　　② 198
③ 238　　④ 328

♤ **SCAN 스케줄링**

- SCAN은 헤드가 트랙을 "**왔다갔다**" 하면서 입출력 요청을 서비스한다.
- SCAN 기법에서 헤드 이동은 트랙 처음부터 마지막까지 이동한다.(서비스 요청이 없어도)
- 즉, 헤드 이동은 트랙 0에서 199까지 이동된다.

- 서비스 순서 : (50) → 60 → 90 → 100 → 150 → 180 → 40 → 30 → 20
  (60 → 90 → 100 → 150 → 180 : 헤드가 트랙 0으로 가면서 서비스 ↰)
  (40 → 30 → 20 : ↳ 헤드가 트랙 199로 가면서 서비스)

- 헤드 이동 : 50 → 60 → 90 → 100 → 150 → 180 → 199 → 40 → 30 → 20
  (199 : ↳ 헤드가 트랙 끝까지 이동하므로)

- 헤드 이동거리 : (199 − 50) + (199 − 20) = 149 + 179 = **328**

정답 : ④

5. LOOK / C-LOOK 스케줄링

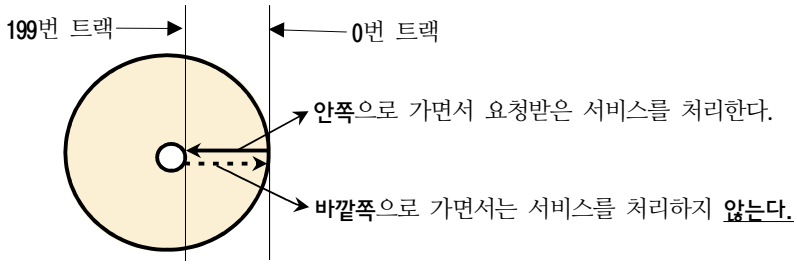
① LOOK은 각 방향에서 마지막 요청이 있는 트랙까지만 헤드를 이동하면서 입출력 서비스를 한다. → 주어진 방향으로 가면서 요청을 살피므로 LOOK이다. 요청이 없으면 더 가지 않는다.

② LOOK 기법에서는 헤드가 무조건 디스크 전체 폭을 가로지르지는 않는다.

→ SCAN과 C-SCAN에서는 디스크 전체 폭을 가로지르게 된다.(불합리)

③ LOOK에서도 LOOK과 C-LOOK으로 구별한다.

[예제] 디스크에 대한 입출력 요청 서비스 운행 중이다. 현재, 디스크의 서비스 요청 대기 큐에 도착한 요청이 다음과 같을 때 C-LOOK 스케줄링 알고리즘에 의한 헤드의 총 이동거리는 얼마인가? (단, 현재 헤드의 위치는 트랙 50에 있고, 헤드의 이동은 199에서 0 방향이다)



요청 대기열의 순서 : 60, 100, 40, 20, 90, 170, 150, 30, 180

---

① 160 ② 190 ③ 230 ④ 310

♤ C-LOOK 스케줄링

• C-LOOK은 한쪽 방향으로만 입출력 요청에 대한 서비스를 처리한다.
→ 반대 방향으로 헤드가 이동할 때는 서비스를 실시하지 않는다.
→ 그리고, C-LOOK은 마지막 요청이 있는 트랙까지만 헤드가 이동된다.

┌ 트랙 20에서 180까지 서비스를 처리한다.

• 서비스 순서 : 50 → 20 → 30 → 40 → 60 → 90 → 100 → 150 → 170 → 180

└ 서비스 없이 헤드가 되돌아가는 거리

• 헤드 이동거리 : (50 − 20) + (180 − 20) = 30 + 160 = **190**

정답 : ②

기출문제 분석

1. 다음에서 설명하는 디스크 스케줄링은? [2018년 지방 9급]

디스크 헤드가 한쪽 방향으로 트랙의 끝까지 이동하면서 만나는 요청을 모두 처리한다. 트랙의 끝에 도달하면 반대 방향으로 이동하면서 만나는 요청을 모두 처리한다. 이러한 방식으로 헤드가 디스크 양쪽을 계속 왕복하면서 남은 요청을 처리한다.

---

① 선입 선처리(FCFS) 스케줄링 ② 최소탐색시간우선(SSTF) 스케줄링
③ 스캔(SCAN) 스케줄링 ④ 라운드 로빈(RR) 스케줄링

♤ **디스크 스케줄링 – SCAN**

• SCAN은 헤드가 트랙을 "왔다갔다" 하면서 입출력 요청에 대한 서비스를 처리한다.
• SCAN은 양쪽 트랙 마지막 끝까지 헤드가 이동된다.

정답 : ③

2. 〈보기〉는 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 방식은 무엇인가? 단, 〈보기〉의 숫자는 입출력할 디스크 블록들이 위치한 트랙 번호를 의미하며, 현재 디스크 헤드의 위치는 트랙 50번이라고 가정한다. [2012년 계리]

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

① FCFS ② C-SCAN
③ SSTF ④ SCAN

♤ **C-SCAN**

┌ 서비스 순서가 처음부터 **다시 증가**한다.
• 서비스 순서 : 64, 66, 99, 121, 125, 182, **12, 35**
└ 서비스 순서가 계속 증가

정답 : ②

3. 운영체제의 디스크 스케줄링 기법에 대한 설명으로 옳은 것은? [2014년 국가 9급]

① FCFS(First-Come-First-Served)는 현재의 판독/기록 헤드위치에서 대기 큐 내 요구들 중 탐색시간이 가장 짧은 것을 선택하여 처리하는 기법이다.

② N-Step-SCAN은 대기 큐 내에서 디스크 암(disk arm)이 외부 실린더에서 내부 실린더로 움직이는 방향에 있는 요구들만을 처리하는 기법이다.

③ C-LOOK은 디스크 암(disk arm)이 내부 혹은 외부 트랙으로 이동할 때, 움직이는 방향에 더 이상 처리할 요구가 없는 경우 마지막 트랙까지 이동하지 않는 기법이다.

④ SSTF(Shortest-Seek-Time-First)는 각 요구 처리에 대한 응답시간을 항상 공평하게 하는 기법이다.

♤ **디스크 스케줄링**

- FCFS(First-Come-First-Served)는 입출력 요청 순서대로 서비스한다.
- SSTF(Shortest-Seek-Time-First)는 현재 헤드 위치에서 탐색거리가 가장 짧은 입출력 요청을 우선적으로 처리한다.

◉ **N-Step-SCAN 스케줄링 : SCAN 기법의 변형**

- 헤드가 진행 방향을 바꾸는 시점 기준으로 큐에 대기 중인 요청들만을 서비스한다.
- 서비스 진행 중에 **추가로 도착한 요청은 다음 진행 방향을 바꾼 후에 처리**한다.
- SCAN에 비해 응답시간 분산이 줄고, 특정 트랙의 무한 지연의 가능성을 차단한다.
- 즉, 특정 트랙에 많은 요청이 발생되어도 반대 방향의 요청이 무한 지연되지 않는다.

[예제] SCAN / N-step-SCAN 스케줄링 서비스

- 디스크가 총 256개의 트랙으로 구성되어 있고
- 현재, 대기큐에 도착된 요청 : 200, 70, 130, 60, 20, 170, 40, 80
- 헤드가 트랙 90번에서 0번으로 이동 중에 **50번 트랙의 요청이 추가**로 **도착**된 경우

↓ **풀이**

SCAN	90 → 80 → 70 → 60 → **50** → 40 → 20 → (0) → 130→ 170 → 200 ↳ scan은 진행 중에 추가로 도착한 요청도 같이 처리
N-step SCAN	90 → 80 → 70 → 60 → 40 → 20 → (0) → **50** → 130→ 170 → 200 ↓ N-step SCAN은 진행 중에 **추가로 도착**한 요청 **50**은 진행 방향을 바꾼 후에 처리

정답 : ③

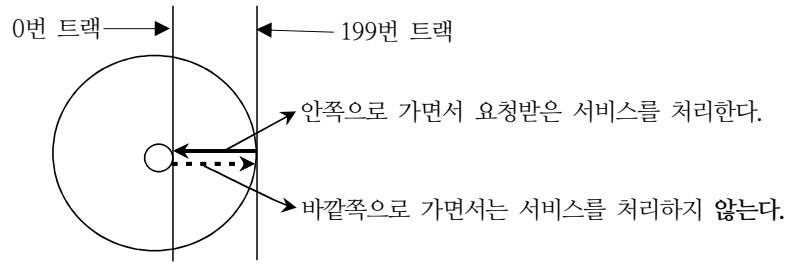
4. 디스크 헤드 위치가 55이고 0의 방향으로 이동할 때, C-SCAN 기법으로 디스크 대기 큐 25, 30, 47, 50, 63, 75, 100을 처리한다면 제일 마지막에 서비스 받는 트랙은? [2017년 국가 9급]

① 50 ② 63
③ 75 ④ 100

♤ C-SCAN(Circular-SCAN, 순환-SCAN) 스케줄링

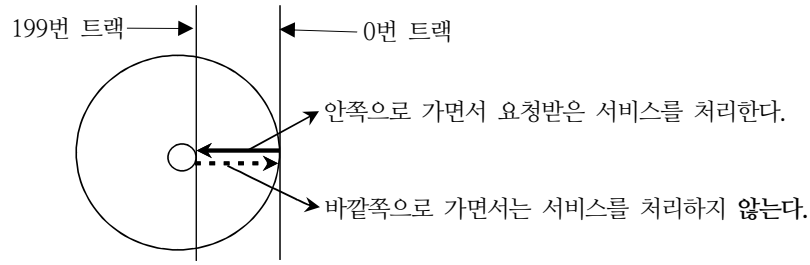
• 먼저, 이 문제는 정답이 2개가 될 수 있다.(그런데, 이의신청을 하지 못했음)
• 이유는 디스크 구조 및 서비스 처리 원리를 정확하게 제시하지 않았으므로

[첫번째] 출제자 정답에 따른 디스크 구조 및 서비스 처리 - 정답 ②



↱ 50에서 25까지 감소하면서 서비스 처리
디스크 대기 큐 : 25, 30, 47, 50, 63, 75, 100
↳ 100에서 63까지 감소하면서 서비스 처리

[두번째] 디스크 구조 및 서비스 처리가 다음과 같은 경우(실제로 바깥 트랙이 0)



디스크 대기 큐 : 25, 30, 47, 50, 63, 75, 100
↳ 25에서 100까지 차례로 증가하면서 서비스 처리

• 이유는 헤드 위치가 55이고, 0의 방향으로 가면서는 서비스 처리 안함
• 해서, 서비스 처리없이 그냥 헤드가 트랙 25까지 이동하므로

정답 : ②

5. 디스크 큐에 다음과 같이 I/O 요청이 들어와 있다 최소탐색시간우선(SSTF) 스케줄링 적용 시 발생하는 총 헤드 이동거리는? (단, 추가 I/O 요청은 없다고 가정한다. 디스크 헤드는 0부터 150까지 이동 가능하며, 현재 위치는 50이다) [2022년 국가 9급]

큐 : 80, 20, 100, 30, 70, 130, 40

---

① 100
② 140
③ 180
④ 430

♤ **최소탐색시간우선(SSTF)**

- 현재 헤드 위치에서 **탐색거리가 가장 짧은 입출력 요청**을 우선적으로 처리한다.
- 새로운 요청이 들어오게 되면 이를 포함시켜서 입출력 서비스를 한다.
- 어떤 요청은 무한정 기다리게 되는 기아(starvation) 현상이 일어날 수 있다.
- 하지만, FCFS보다 평균적으로 우수하다(짧은 응답시간, 많은 처리량)
- 가운데 트랙이 보다 많은 서비스를 받게 되는 **탐색 편중 현상**이 일어난다.
- 응답시간 편차로 대형 시스템에는 적합하지 않다.

큐 : 80, 20, 100, 30, 70, 130, 40

↓ **현재 헤드 위치는 50이므로**

서비스 순서 : 50 → 40, 30, 20, 70, 80, 100, 130

헤드 이동거리 : (50 − 20) + (130 − 20) = 30 + 110 = **140**

- SSTF는 FCFS보다는 개선된 알고리즘이지만 매우 좋은 알고리즘은 아니다.

정답 : ②