

# 디스크 성능 최적화



13<sup>th</sup> Week  
Kim, Eui-Jik

# Contents

- 2차 저장소의 발전
- 무빙 헤드 디스크 저장소의 특징
- 디스크 스케줄링이 필요한 이유
- 디스크 스케줄링 전략
- 회전 최적화

## 2차 저장소의 발전

- 마그네틱 저장소의 개발
  - 읽기/쓰기 헤드를 통한 데이터 접근
  - (1951년) 마그네틱 테이프 저장소
    - 영구적이면서 재기록 가능한 저장 매체
    - 일종의 순차 접근 저장소
    - 트랜잭션을 처리하는 응용에 적합하지 않음
  - (1957년) 최초의 하드 디스크 드라이브
    - RAMAC(random access method of accounting and control)
    - 임의 접근(직접 접근) 장치
    - 순차적으로 접근해야 한다는 제한이 없음

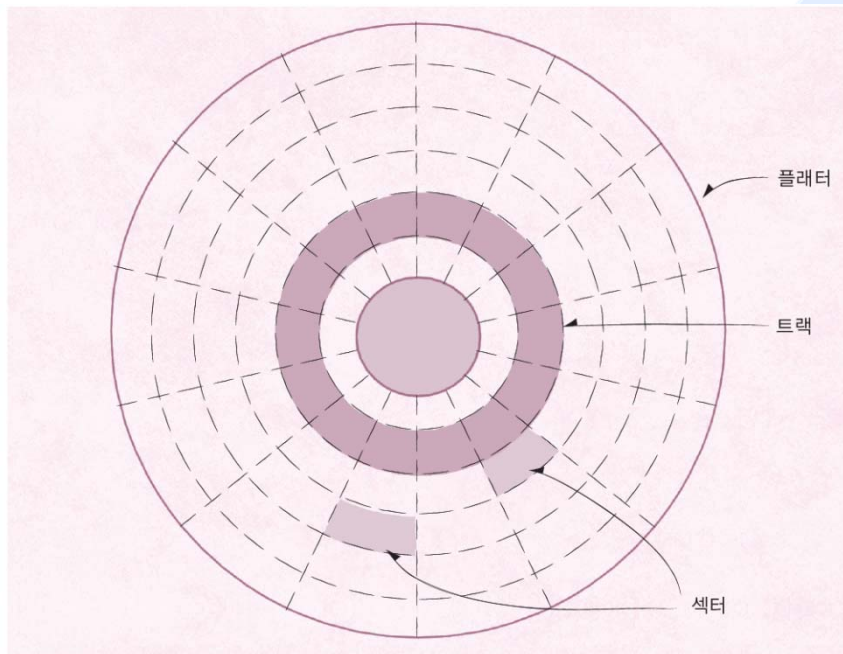
# 무빙 헤드 디스크 저장소의 특징

- 데이터
  - 마그네틱 디스크 또는 플래터에 기록
  - 디스크 표면에 있는 데이터
    - 읽기/쓰기 헤더를 통해 접근
- 플래터
  - 고속으로 회전하는 스피들에 연결
  - 회전시 읽기/쓰기 헤드는 디스크 표면에 있는 데이터의 원형 트랙을 훑음
- 읽기/쓰기 헤드
  - 액츄에이터에 달려 있는 디스크 암 끝에 위치
- 디스크암
  - 디스크의 표면에 평행하게 움직임
  - 읽기/쓰기 헤드를 새로운 위치로 옮길 시 실린더에 접근
  - 새로운 실린더로 옮기는 과정(탐색)

# 무빙 헤드 디스크 저장소의 특징

## ■ 섹터

- 작은 단위의 데이터에 접근하기 위해 트랙을 나눔
- 운영체제는 데이터가 위치한 헤드와 실린더, 섹터를 명시
  - 특정 데이터의 위치를 나타냄

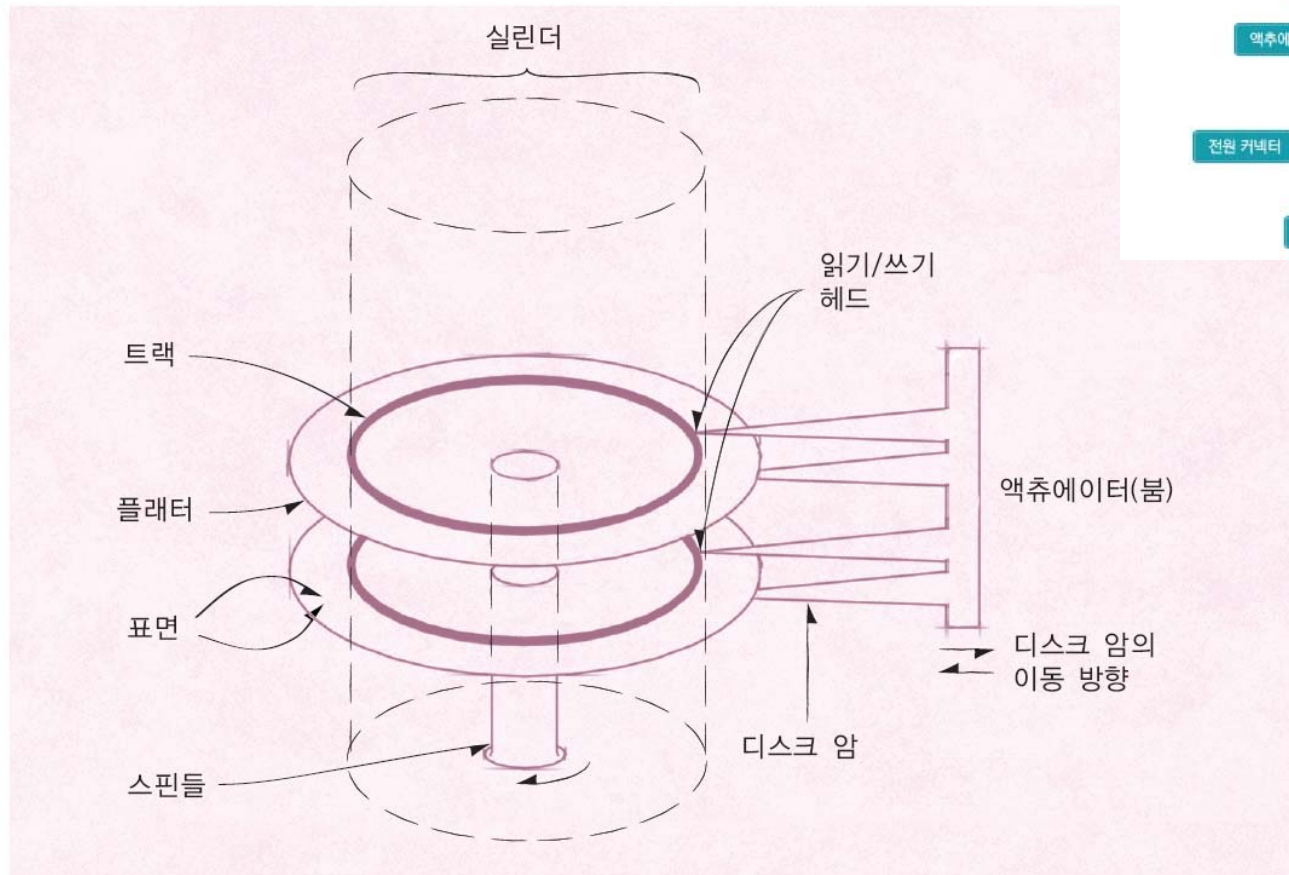


[그림 12-2] 윗면에서 본 디스크 표면의 구조



[그림 12-3] 디스크 접근의 구성 요소

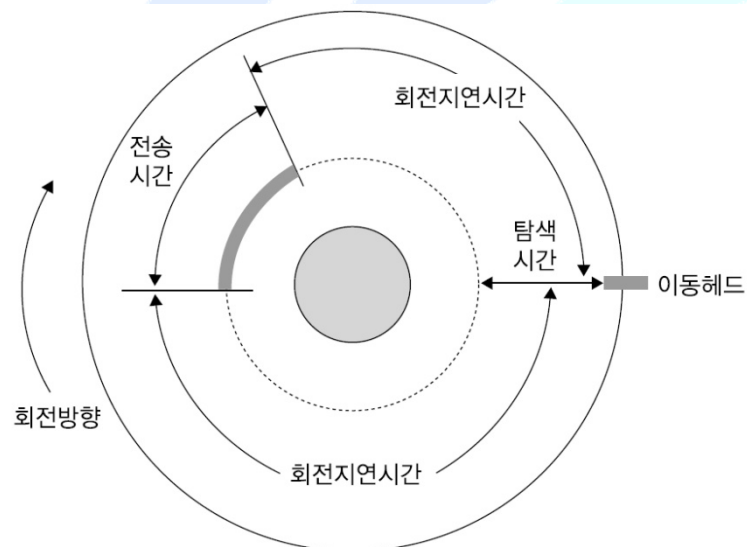
# 무빙 헤드 디스크 저장소의 특징



[그림 12-1] 무빙 헤드 디스크의 측면 도해

# 디스크 액세스 시간

- 디스크 액세스 시간
  - 탐색(Seek)
    - 디스크상의 원하는 섹터에 액세스하기 위해 헤드를 해당 트랙 또는 실린더로 이동해 위치시킴
  - 탐색 시간(Seek Time)
    - 탐색에 걸리는 시간
    - 디스크의 헤드가 움직이는 시간에 좌우되므로 멀리 떨어진 트랙 탐색 시 탐색시간이 길어짐
  - 회전지연시간(Rotational Latency Time)
    - 원하는 섹터가 입출력 헤드 아래로 회전할 때를 기다리는 시간
  - 전송시간(Transmission Time)
    - 디스크와 메인 메모리 간의 섹터를 주고받는 데 걸리는 시간



디스크 액세스 시간

# 디스크 스케줄링이 필요한 이유

- 여러 프로세스가 동시에 디스크의 데이터를 읽거나 쓰려는 요청
  - 디스크가 서비스할 수 있는 속도보다 빠르게 요청을 발생
  - 디스크 요청을 보관할 수 있는 큐 사용
- 초기의 컴퓨팅 시스템
  - 단순 선착 우선 서비스 (FCFS) 방식
  - FCFS
    - 공평한 방식
    - 요청률, 시스템 부하가 커질 경우
      - 대기 시간이 길어지는 단점



# 디스크 스케줄링 전략

## ■ 디스크 스케줄링 전략

- 입출력장치(디스크 드라이버)는 요청(Request)을 위한 큐를 가짐
  - 프로세스는 입출력이 필요할 때마다 운영체제에 시스템 호출을 보냄
  - 요청은 다음과 같은 정보를 포함함
    - 입력 동작인지 출력 동작인지에 대한 정보
    - 디스크 주소(구동기, 실린더, 표면, 블록)
    - 메모리 주소
    - 전송할 정보의 총량(바이트 또는 단어의 수)
  - 다른 프로세스가 둘 중 하나라도 사용하고 있다면 요청은 디스크 대기 큐에 저장됨
  - 디스크 대기 큐에서 시스템의 스케줄링은 아래의 몇 가지 기준으로 평가할 수 있음
    - 처리량 단위 : 시간당 처리한 서비스 요청의 수
    - 탐색시간 : 디스크 헤드(암) 이동시간
    - 평균반응시간 : 요청 후 서비스될 때까지의 대기시간
    - 반응(응답)시간 변화 : 반응시간에 대한 예측 정도 (적절한 시간 안에 서비스하여 요청이 무기한 연기되지 않도록 방지함)

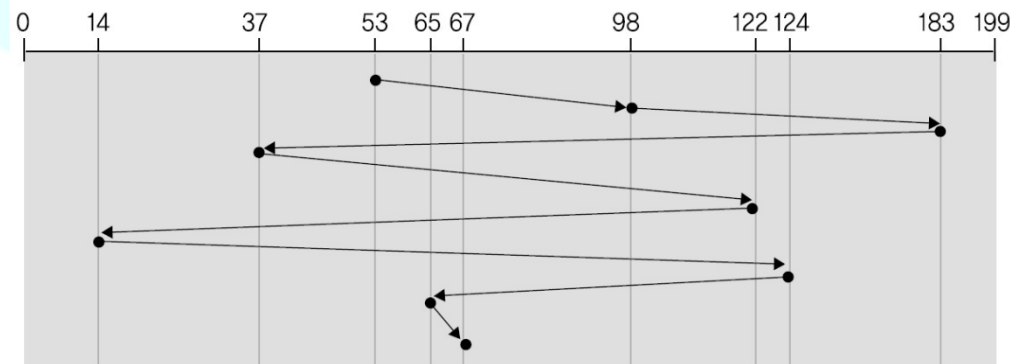
※ 스케줄링 정책은 처리량을 최대화하고 평균반응시간과 탐색시간을 최소화해야 함

※ 처리량 및 평균반응시간 최적화로 시스템 성능 향상은 가능하나 개별 요청에 지연 발생 가능

# 디스크 스케줄링

- 선입 선처리 스케줄링(FCFS, First-Come-First-Served)
  - 디스크 스케줄링의 가장 간단한 형태
    - 요청이 도착한 순서에 따라 처리
    - 프로그램하기 쉽고 어떤 요청도 무기한 연기되는 경우가 없으며 본질적으로 공정성이 유지됨
  - FCFS 스케줄링 예
    - 처음 요청이 트랙 98, 마지막은 트랙 67로 되는 선형 요청 디스크 큐
  - 문제점
    - 디스크 요청이 흩어져 있는 경우 실행시간 오버헤드는 적으나 탐색시간이 오래 걸려 처리량이 감소함

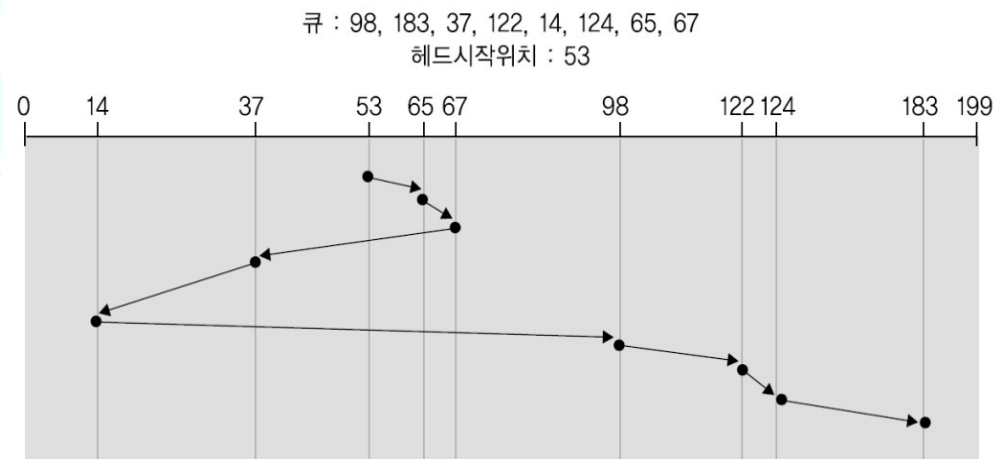
큐 : 98, 183, 37, 122, 14, 124, 65, 67  
헤드시작위치 : 53



FCFS 스케줄링

# 디스크 스케줄링

- 최소 탐색시간 우선 스케줄링(SSTF, Shortest Seek Time First)
  - 디스크 요청을 처리하기 위해서 헤드가 먼 곳까지 이동하기 전에 현재 헤드 위치에 가까운 모든 요구를 먼저 처리하는 방법
  - 최소탐색 우선 알고리즘 예
  - 디스크 처리시간을 실질적으로 줄일 수 있으나, 디스크 요구의 기아상태 발생 가능
    - [그림]에서 트랙 14와 186에 대한 요청이 있는 경우
      - 트랙 186을 기다리게 하고 트랙 14를 처리할 경우, 트랙 14를 처리하는 동안 가까운 곳에서 다른 요청이 계속 발생할 수 있음
      - 이론적으로 트랙 186을 무한히 기다리도록 하는 연속적인 요청이 있을 가능성이 있음



# 디스크 스케줄링

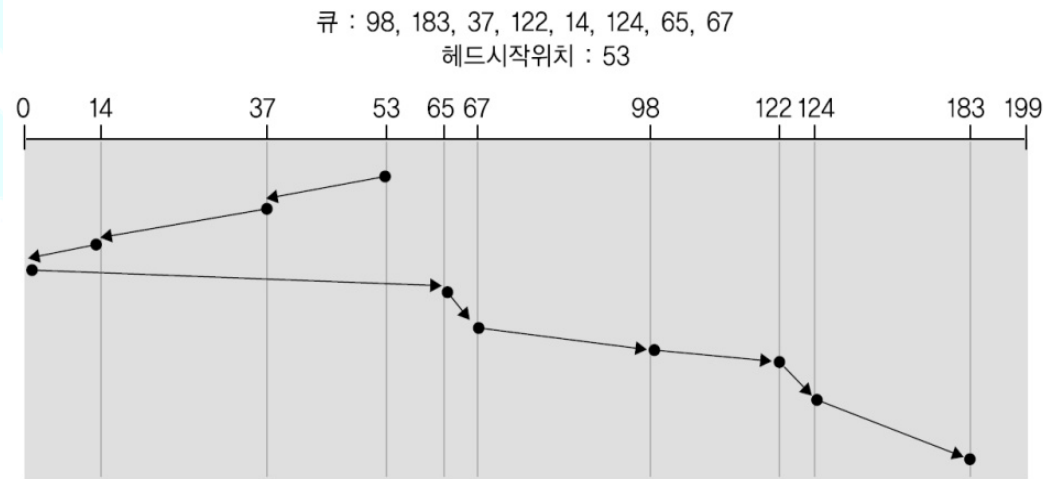
## ■ 스캔 (SCAN) 스케줄링

### ■ 요청 큐의 동적 특성 반영

- 입출력 헤드가 디스크의 한 끝에서 다른 끝으로 이동
- 한쪽 끝에 도달했을 때는 역방향으로 이동하면서 요청한 트랙을 처리
  - 헤드는 디스크의 한 끝과 다른 끝 사이를 계속해서 왕복함
  - 새로운 디스크 요청이 헤드의 바로 앞 요청의 큐에 도착하면 즉시 처리되나, 헤드 뒤에 있는 경우 헤드가 디스크의 끝까지 이동하고 돌아올 때까지 기다려야 함

### ■ 스캔 알고리즘 예

- 현재 헤드의 위치와 이동방향 파악 필요



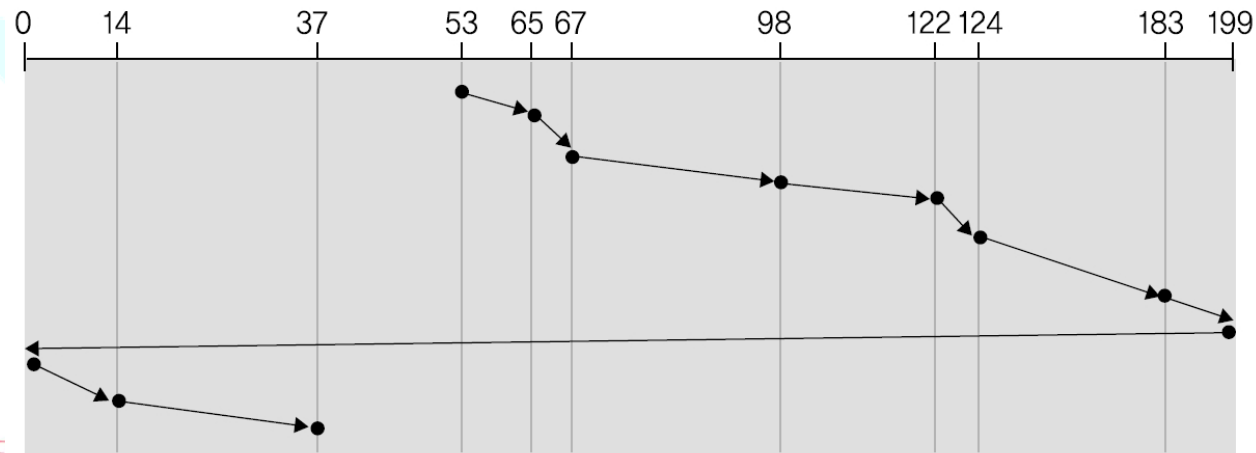
스캔 알고리즘

# 디스크 스케줄링

- 순환 스캔 스케줄링(C-SCAN, Circular SCAN scheduling)
  - 스캔 알고리즘을 변형, 대기시간을 균등하게
    - 스캔 스케줄링처럼 헤드는 한쪽 방향으로 이동하면서 요청을 처리하지만, 한쪽 끝에 다다르면 반대 방향으로 헤드가 이동하지 않고 다시 처음부터 요청을 처리
    - 처음과 마지막 트랙을 서로 인접시킨 것과 같은 원형처럼 디스크를 처리, 처리량을 향상
    - 바깥 트랙과 안쪽 트랙에 대한 차별이 없어 반응시간의 변화를 줄임
    - 동일한 실린더(트랙)에 대한 요청이 연속적으로 발생되면 처리가 무기한 연기될 수 있음

큐 : 98, 183, 37, 122, 14, 124, 65, 67

헤드시작위치 : 53



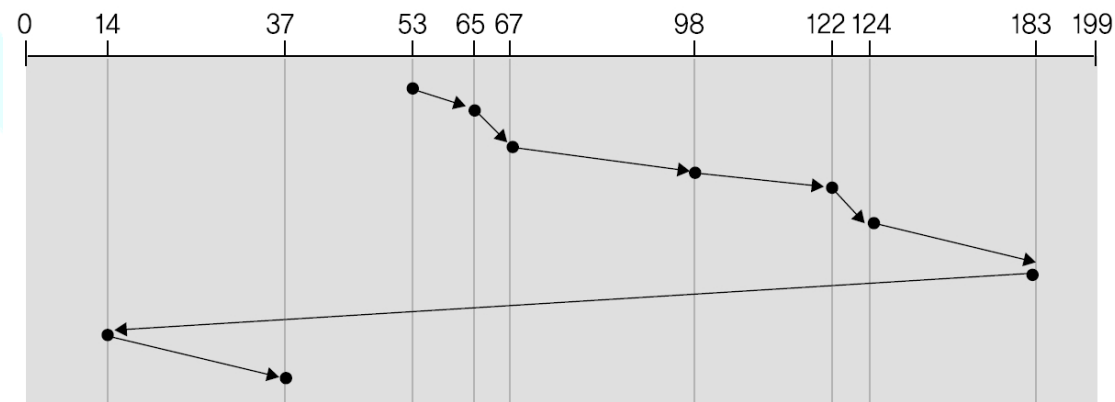
# 디스크 스케줄링

## ■ 룩(Look) 스케줄링

### ■ 스캔 또는 순환 스캔 방법 원리와 달리 구현됨

- 헤드는 요청에 따라 각 방향으로 이동, 현재 방향에 더 이상의 요청이 없을 때 이동 방향을 바꾼다
- 스캔과 순환 스캔의 이런 형태를 룩(Look), 또는 순환 룩(C-Look)라 부름
  - Look : 진행 방향으로 움직이기 전에 먼저 요청이 있는지 검사함을 의미함

큐 : 98, 183, 37, 122, 14, 124, 65, 67  
헤드시작위치 : 53



순환 룩 알고리즘

# 회전 최적화

- 초기 하드 드라이브
  - 접근 시간을 좌우하는 것
    - 탐색 시간
    - 탐색 최적화에 관한 연구
- 현재 하드 디스크
  - 탐색 시간과 평균 지연 시간의 자릿수 같은 정도
  - 회전 최적화로도 성능을 개선 할 수 있음을 의미

# 회전 최적화

- 최단 지연 시간 우선 SLTF 스케줄링
  - **Shortest-Latency-Time-First**
  - 디스크 암이 특정 실린더에 도달하면, 해당 실린더의 트랙에 대한 여러 요청이 대기
  - 모든 요청 중 회전 지연 시간이 가장 짧은 요청을 먼저 서비스
  - **섹터 큐잉(Sector Queueing)** 알고리즘이라 표현
    - 요청들을 섹터 위치에 따라 큐에 넣고, 가장 가까운 섹터에 대한 요청을 먼저 서비스

