

10주차 I/O system ~ RAID Architecture

I/O System (HW)

I/O Mechanisms

Pooling (Programmed I/O)

Interrupt

Direct Memory Access (DMA)

I/O Services of OS

Disk Scheduling

Optimizing seek time

First Come First Service (FCFS)

Shortest Seek Time First (SSTF)

Scan Scheduling

C-Scan Scheduling

Look Scheduling

Optimizing rotational delay

Shortest Latency Time First (SLTF)

Shortest Positioning Time First (SPTF)

RAID Architecture

RAID 0

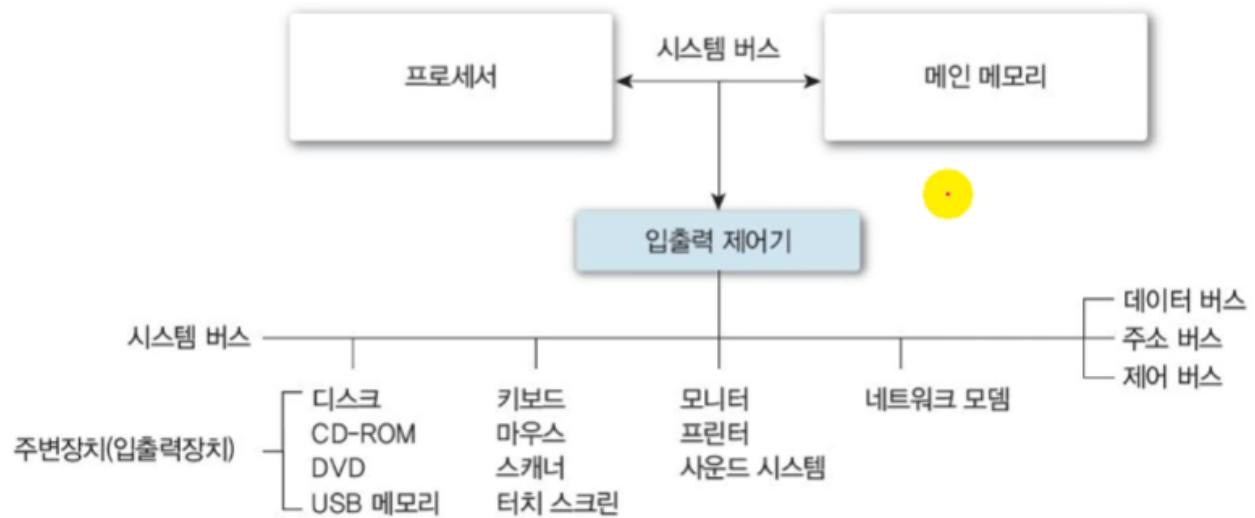
RAID 1

RAID 3

RAID 4

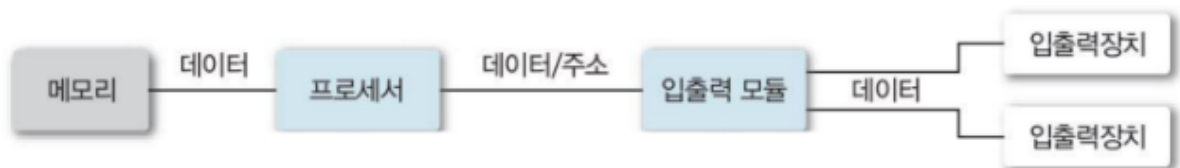
RAID 5

I/O System (HW)



I/O Mechanisms

- 프로세서와 입출력 장치 간의 데이터 전송 방법
- Processor controlled memory access
 - Polling (Programmed I/O)
 - Interrupt



- Direct Memory Access (DMA)

Pooling (Programmed I/O)

- Processor가 주기적으로 I/O 장치의 상태 확인
 - 모든 I/O 장치를 순환하면 확인
 - 전송 준비 및 전송 상태 등
- 장점
 - Simple

- I/O 장치가 빠르고, 데이터 전송이 잦은 경우 효율적
- 단점
 - Processor의 부담이 큼
 - Pooling overhead (I/O device가 느린 경우)

Interrupt

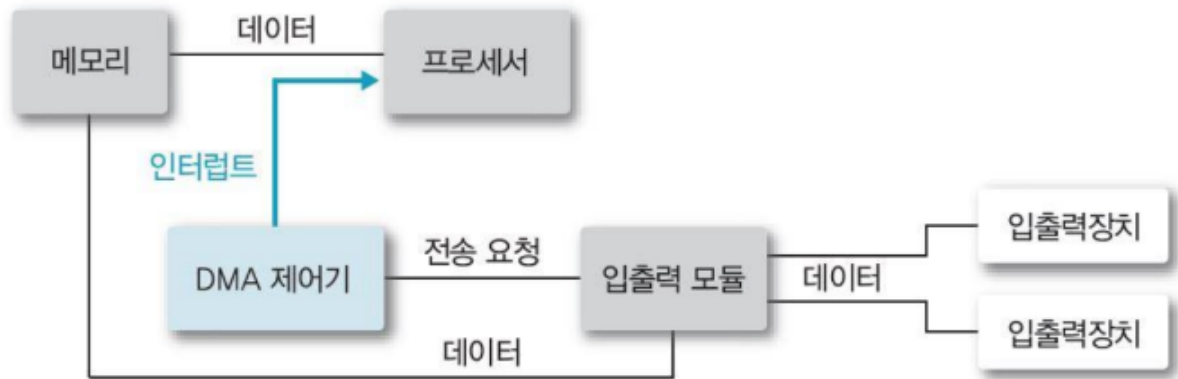
- I/O 장치가 작업을 완료한 후, 자신의 상태를 Processor에게 전달
 - Interrupt 발생 시, Processor는 데이터 전송 수행



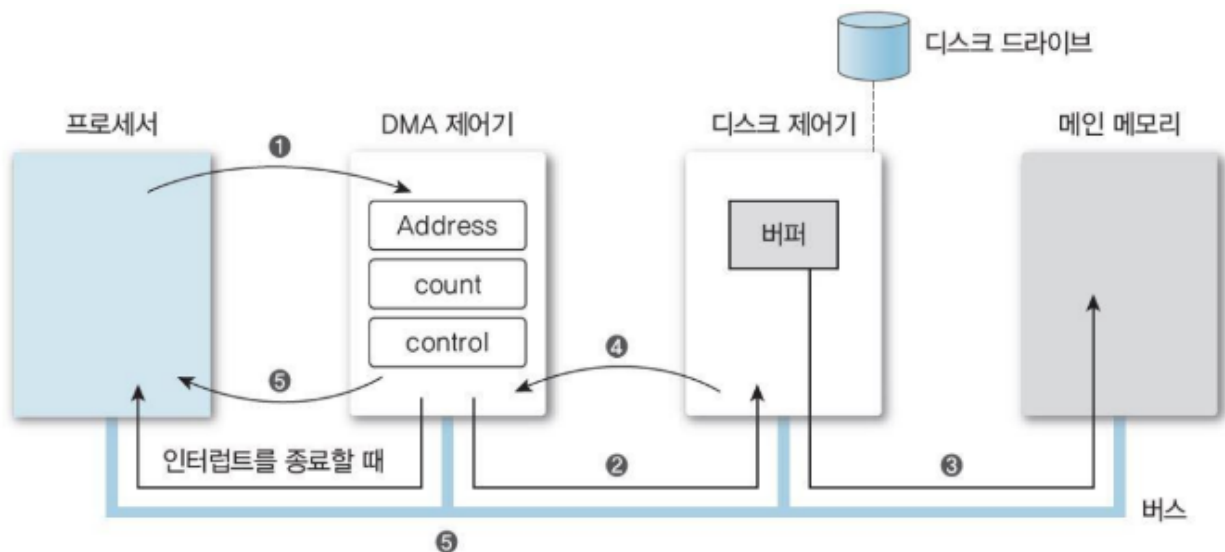
- 장점
 - Pooling 대비 low overhead
 - 불규칙적인 요청 처리에 적합
- 단점
 - Interrupt handling overhead

Direct Memory Access (DMA)

- Processor controlled memory access 방법
 - Processor가 모든 데이터 전송을 처리해야 함
 - High overhead for the processor
- Direct Memory Access
 - I/O 장치와 Memory 사이의 데이터 전송을 Processor 개입 없이 수행



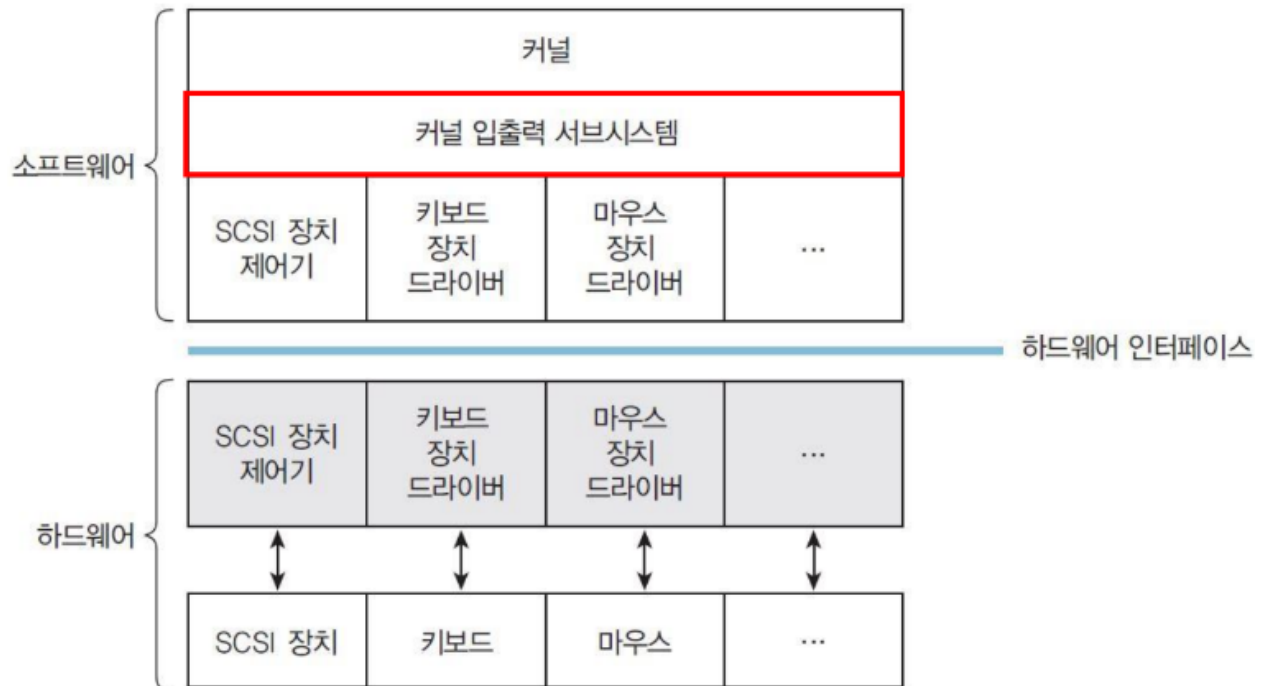
- Processor는 데이터 전송의 시작/종료 만 관여



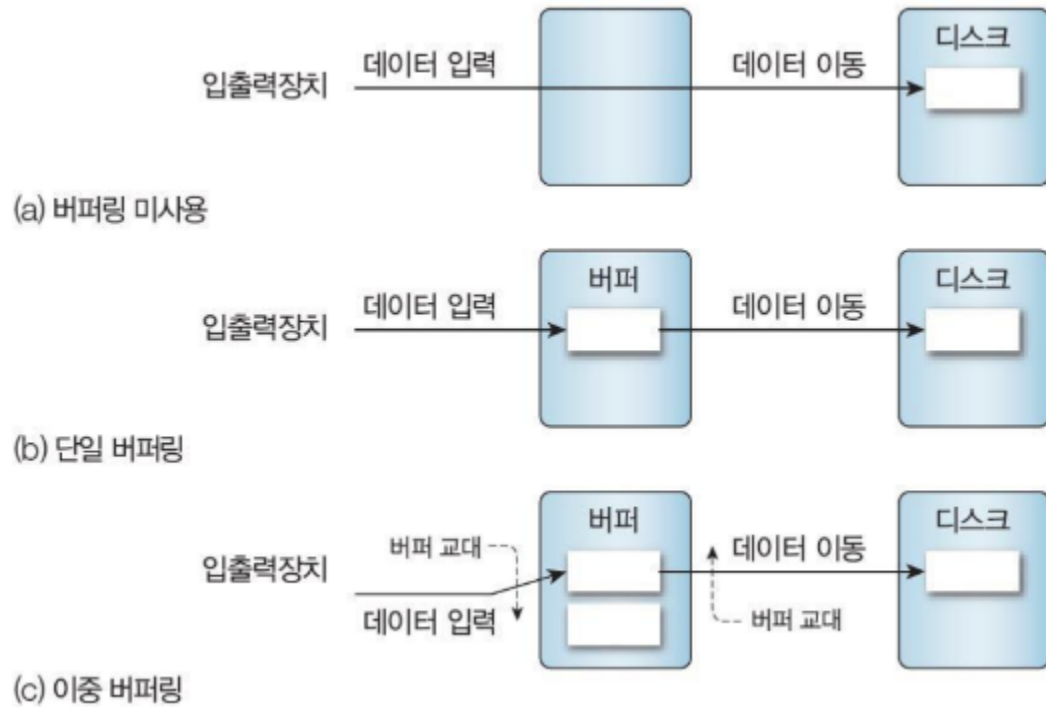
- ① 프로세서가 전송 방향, 전송 바이트 수, 데이터 블록의 메모리 주소 등을 DMA 제어기에 보낸다.
- ② DMA 제어기는 디스크 제어기에 데이터를 메인 메모리로 전송하라고 요청한다.
- ③ 디스크 제어기가 메인 메모리에 데이터를 전송한다.
- ④ 데이터 전송을 완료하면 디스크 제어기는 DMA 제어기에 완료 메시지를 전달한다.
- ⑤ DMA 제어기가 프로세서에 인터럽트 신호를 보낸다.

I/O Services of OS

- 입출력 효율성을 높이기 위해 OS가 지원하는 것



- I/O Scheduling
 - 입출력 요청에 대한 처리 순서 결정
 - 시스템의 전반적 성능 향상
 - Process의 요구에 대한 공평한 처리
 - E.g., Disk I/O scheduling
- Error handling
 - 입출력 중 발생하는 오류 처리
 - E.g., disk access fail, network communication error 등
- I/O device information managements
- Buffering
 - I/O 장치와 Program 사이에 전송되는 데이터를 Buffer에 임시 저장
 - 전송 속도 (or 처리 단위) 차이 문제 해결



- Caching
 - 자주 사용하는 데이터를 미리 복사해 둠
 - Cache hit시 I/O를 생략 할 수 있음
- Spooling
 - 한 I/O 장치에 여러 Program이 요청을 보낼 시, 출력이 섞이지 않도록 하는 기법
 - 각 Program에 대응하는 disk file에 기록 (spooling)
 - Spooling이 완료 되면, spool을 한번에 하나씩 I/O 장치로 전송

Disk Scheduling

- Disk access 요청들의 처리 순서를 결정
- Disk system의 성능을 향상
- 평가 기준
 - Throughput (단위 시간당 처리량)
 - Mean response time (평균 응답 시간)

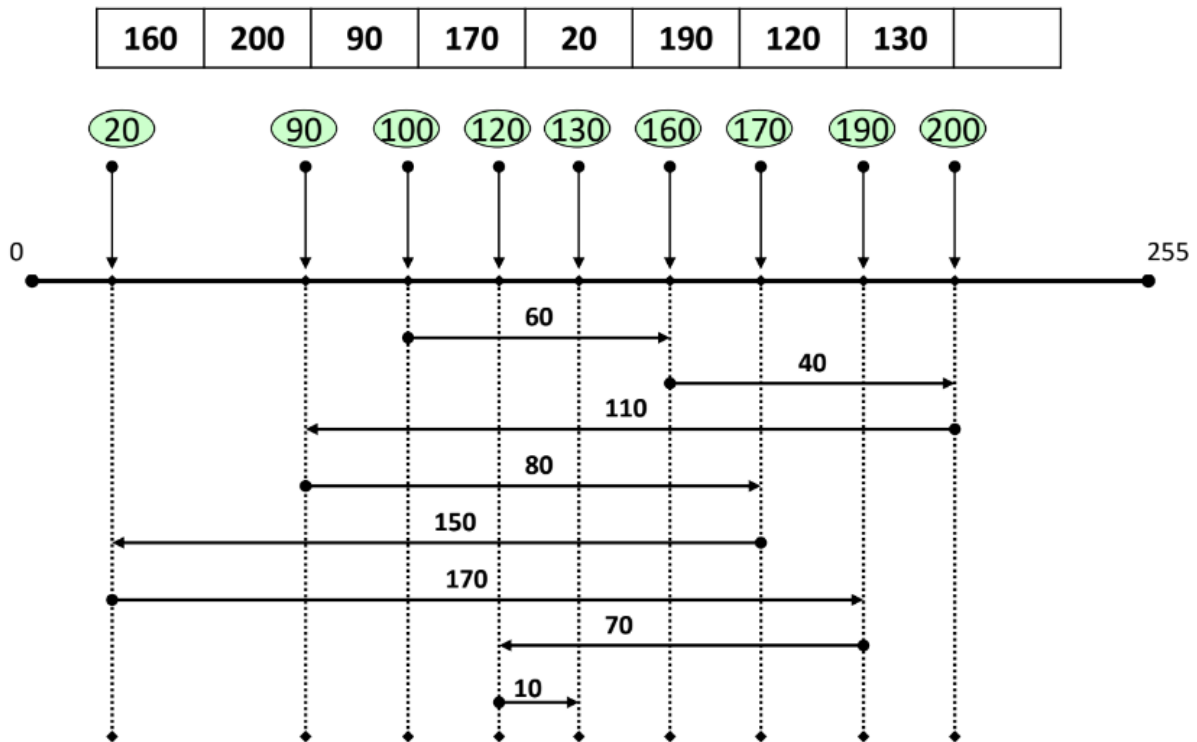
- Predictability (응답 시간의 예측성)
 - 요청이 무기한 연기(starvation)되지 않도록 방지
- 1. Seek time
 - 디스크 head를 필요한 cylinder로 이동하는 시간
- 2. Rotational delay
 - 1 이후에서 부터,
 - 필요한 sector가 head 위치로 도착하는 시간
- 3. Data transmission time
 - 2 이후에서 부터,
 - 해당 sector를 읽어서 전송(or 기록)하는 시간
- $1+2+3 = \text{Data access time}$
- Optimizing seek time
 - FCFS (First Come First Service)
 - SSTF (Shortest Seek Time First)
 - Scan
 - C-Scan (Circular Scan)
 - Look
- Optimizing rotational delay
 - Sector queueing (SLTF, Shortest Latency Time Frist)
- SPTF (Shortest Positioning Time First)

Optimizing seek time

First Come First Service (FCFS)

- 요청이 도착한 순서에 따라 처리
- 장점
 - Simple

- Low scheduling overhead
 - 공평한 처리 기법 (무한 대기 방지)
- 단점
 - 최적 성능 달성에 대한 고려가 없음
- Disk access 부하가 적은 경우에 적합
- Example
 - 총 256개의 cylinder으로 구성
 - Head의 시작 위치 : 100번 cylinder
 - Access request queue

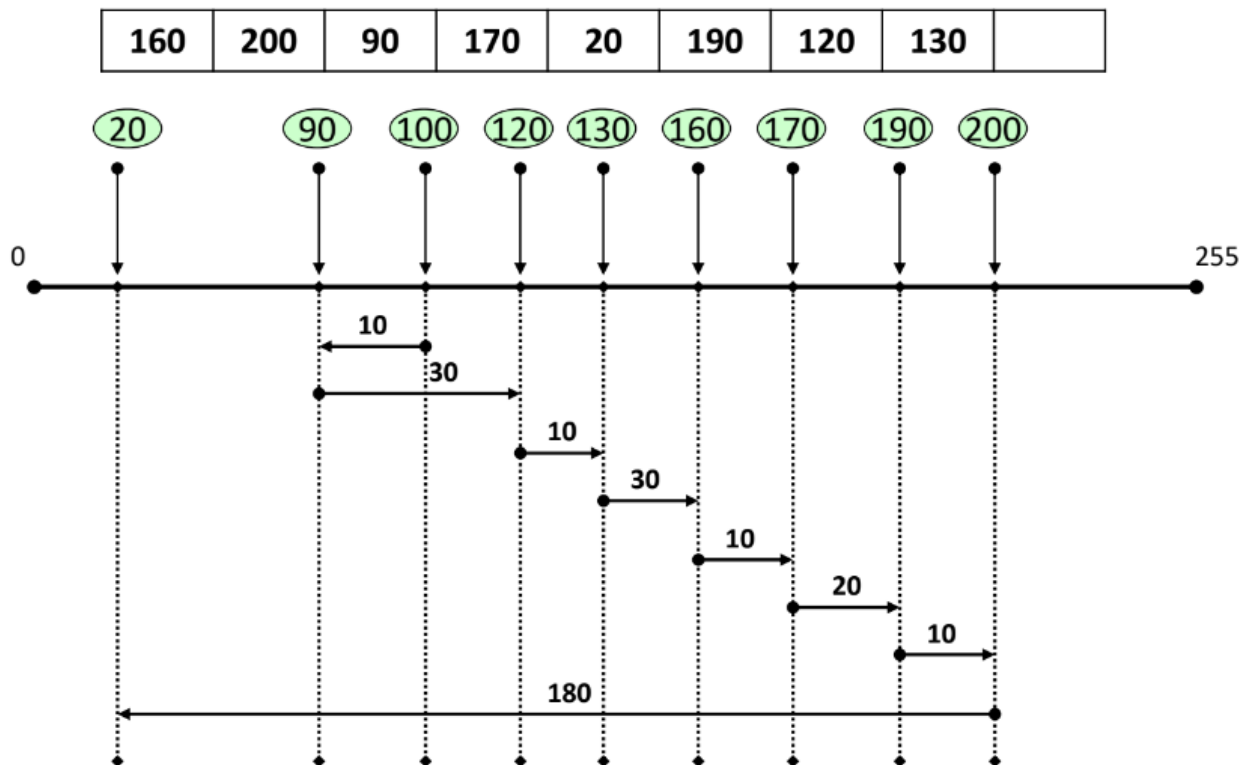


Total seek distance = 690

Shortest Seek Time First (SSTF)

- 현재 head 위치에서 가장 가까운 요청 먼저 처리
- 장점

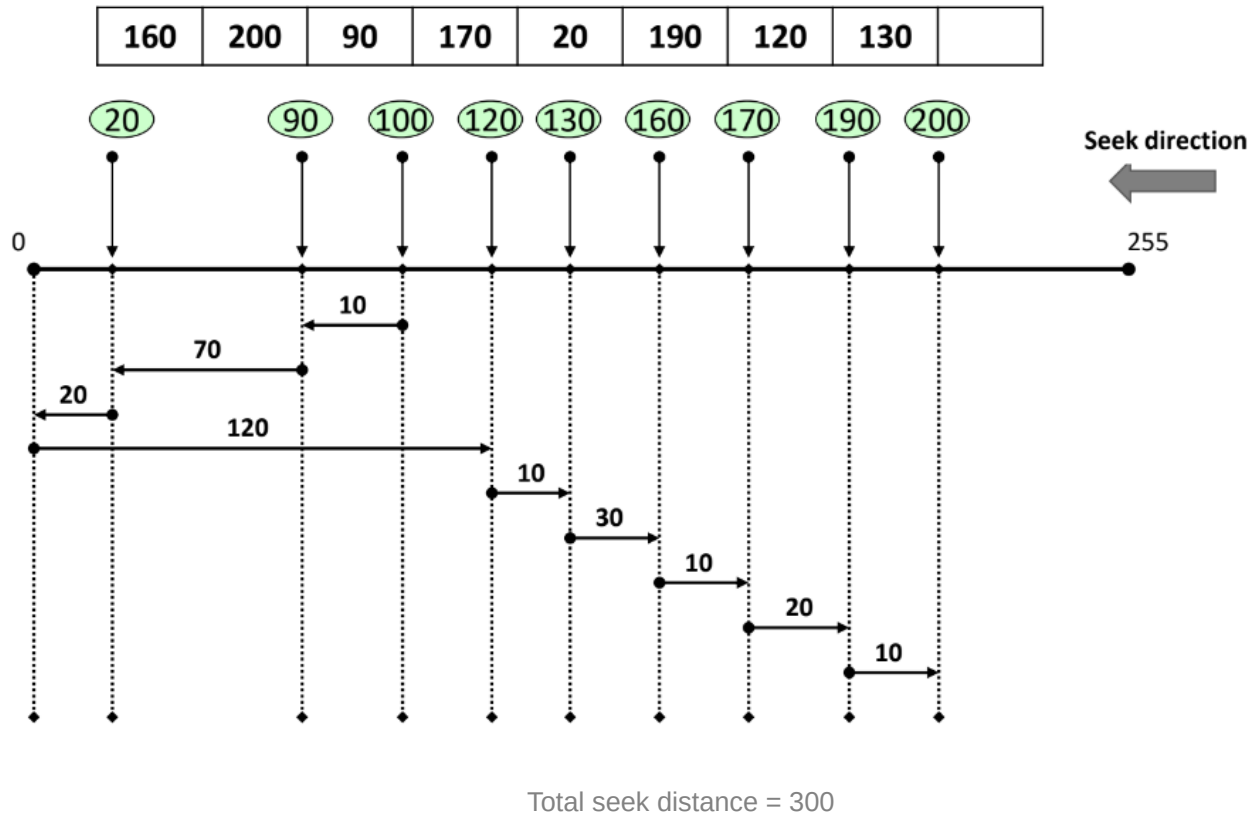
- 이동거리 ↓ ⇒ Throughput ↑
- 평균 응답 시간 ↓
- 단점
 - Predictability ↓
 - Starvation 현상 발생 가능
- 일괄처리 시스템에 적합



Scan Scheduling

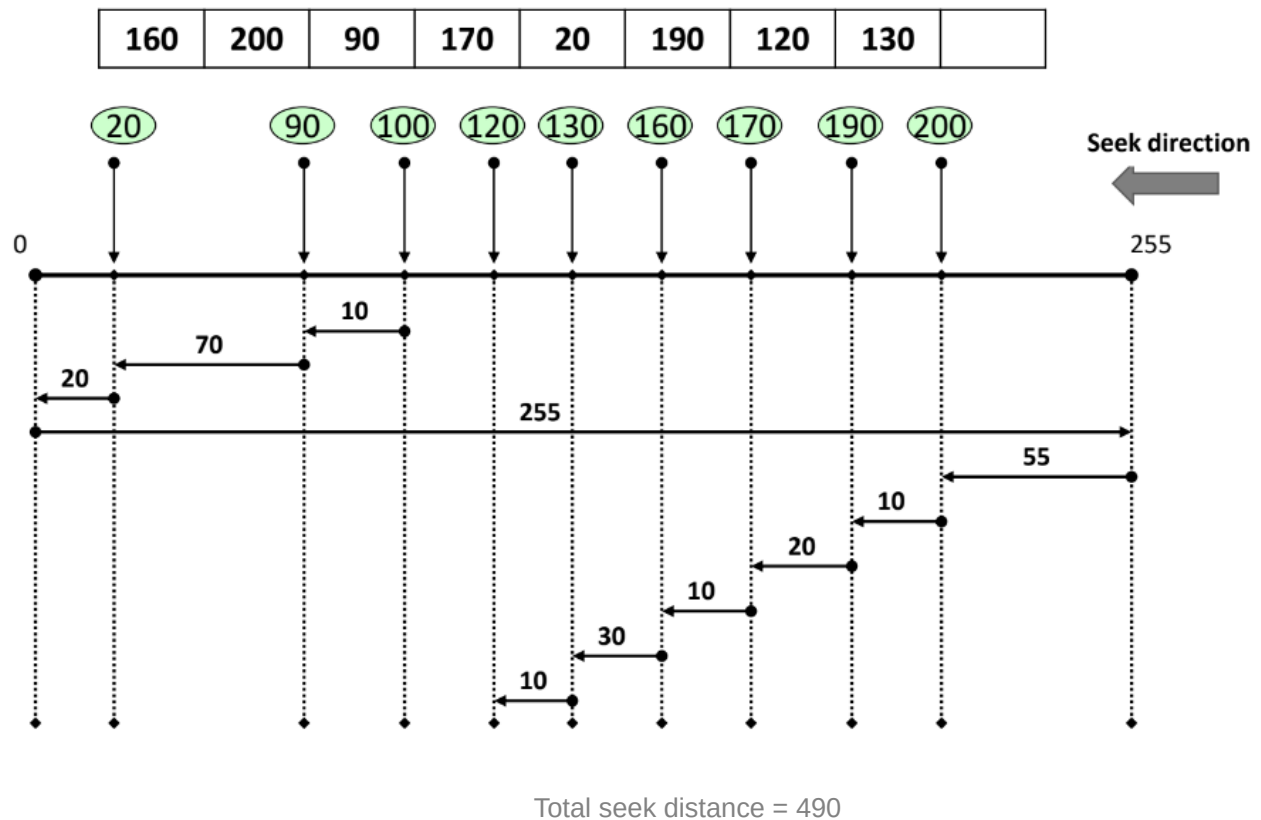
- 현재 head의 진행 방향에서, head와 가장 가까운 요청 먼저 처리
 - (진행방향 기준) 마지막 cylinder 도착 후, 반대 방향으로 진행
- 장점
 - SSTF의 starvation 문제 해결

- Throughput 및 평균 응답시간 우수
- 단점
 - 진행 방향 반대쪽 끝의 요청들의 응답시간 ↑



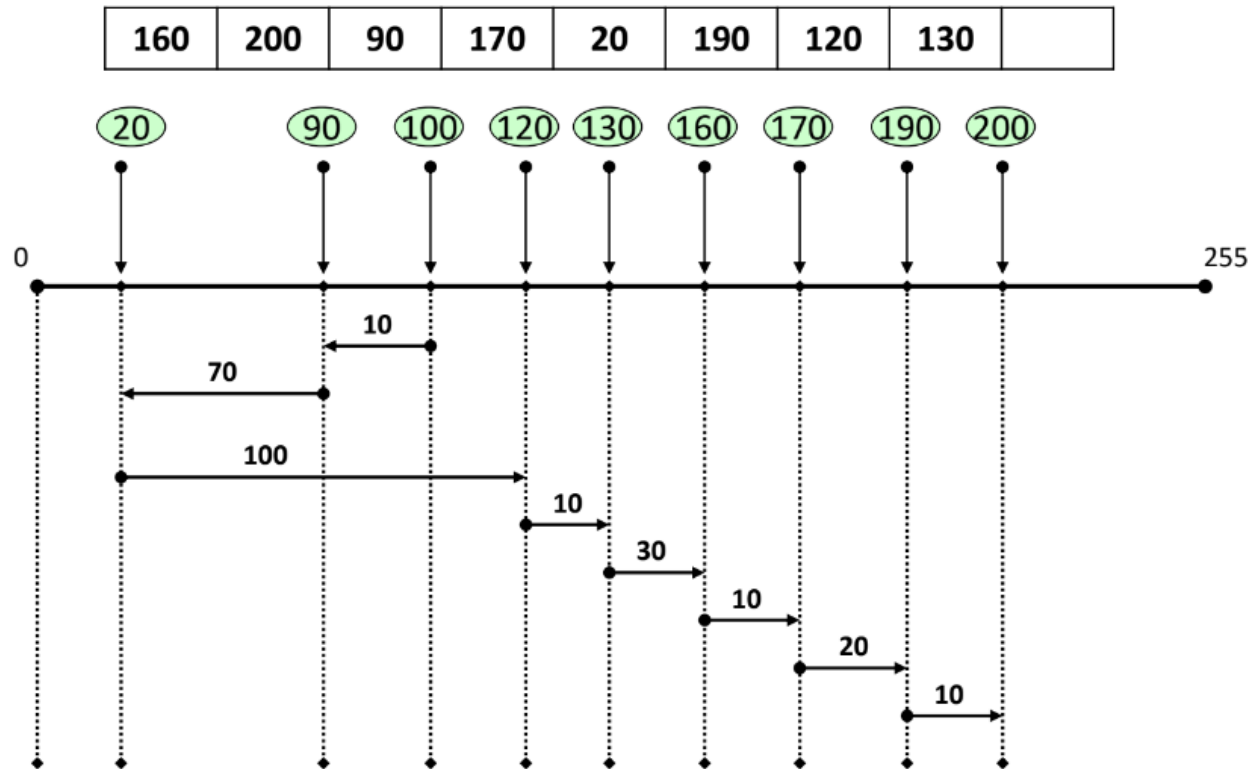
C-Scan Scheduling

- SCAN과 유사
- Head가 미리 정해진 방향으로만 이동
 - 마지막 cylinder 도착 후, 시작 cylinder로 이동 후 재시작
- 장점
 - Scan대비 균등한 기회 제공



Look Scheduling

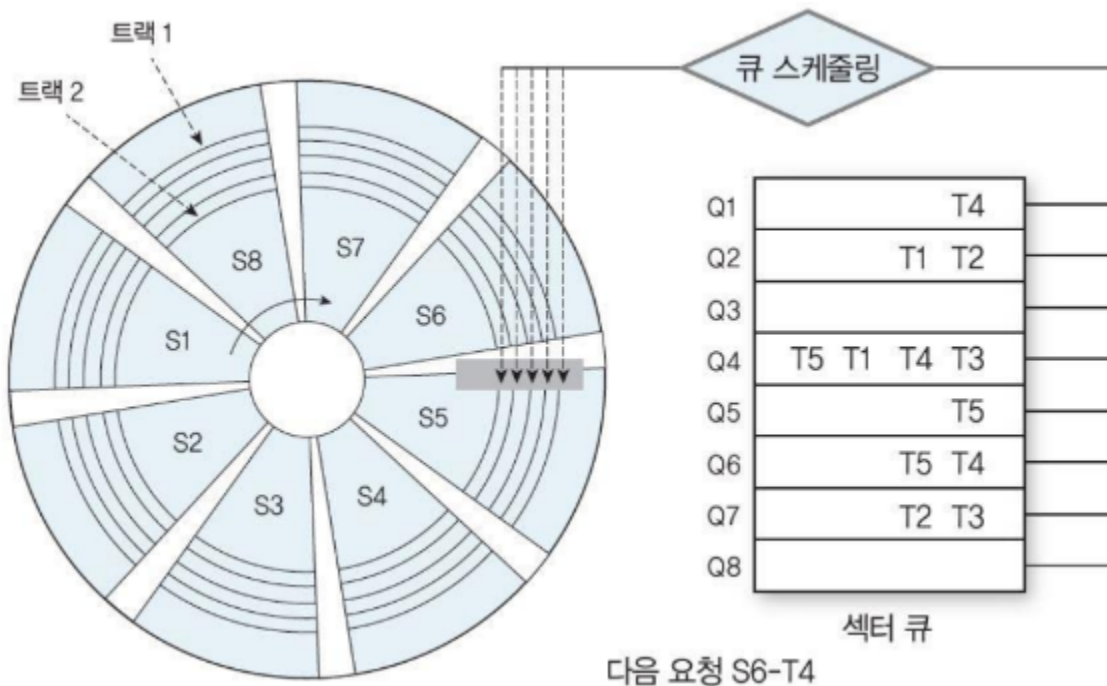
- Elevator algorithm
- Scan (C-Scan)에서 현재 진행 방향에 요청이 없으면 방향 전환
 - 마지막 cylinder까지 이동하지 않음
 - Scan (C-Scan)의 실제 구현 방법
- 장점
 - Scan의 불필요한 head 이동 제거



Optimizing rotational delay

Shortest Latency Time First (SLTF)

- Fixed head disk 시스템에 사용
 - 각 track마다 head를 가진 disk
 - e.g., drum disk
 - Head의 이동이 없음
- Sector queuing algorithm
 - 각 sector별 queue 유지
 - Head 아래 도착한 sector의 queue에 있는 요청을 먼저 처리 함



- Moving head disk의 경우, 같은 cylinder에 여러 개의 요청 처리를 위해 사용 가능
 - Head가 특정 cylinder에 도착하면, 고정 후 해당 cylinder의 요청을 모두 처리

Shortest Positioning Time First (SPTF)

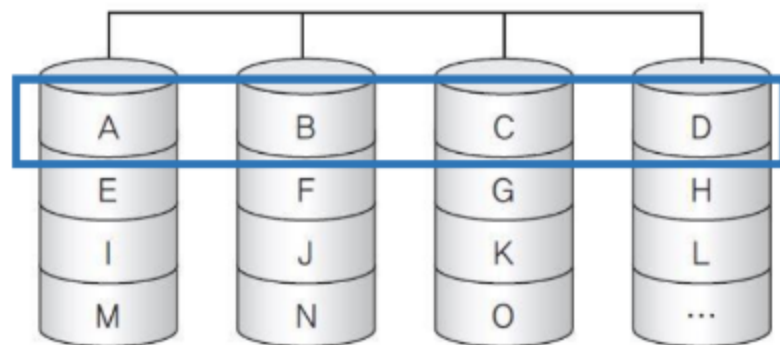
- Positioning time = Seek time + rotational delay
- Positioning time이 가장 작은 요청 먼저 처리
- 장점
 - Throughput ↑, 평균 응답 시간 ↓
- 단점
 - 가장 안쪽과 바깥쪽 cylinder의 요청에 대해 starvation 현상 발생 가능
- Eschenbach scheduling
 - Positioning time 최적화 시도
 - Disk가 1회전 하는 동안 요청을 처리할 수 있도록 요청을 정렬
 - 한 cylinder내 track, sector들에 대한 다수의 요청이 있는 경우, 다음 회전에 처리 됨

RAID Architecture

- Redundant Array of Inexpensive Disks (RAID)
- 여러 개의 물리 disk를 하나의 논리 disk로 사용
 - OS support, RAID controller
- Disk system의 성능 향상을 위해 사용
 - Performance (access speed)
 - Reliability

RAID 0

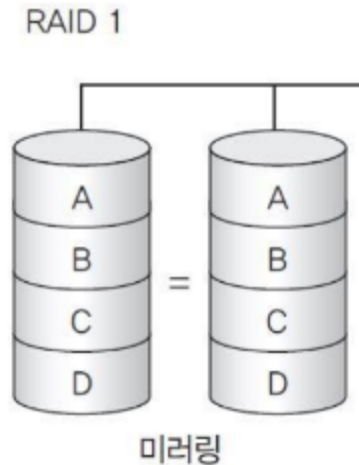
- Disk striping
 - 논리적인 한 block을 일정한 크기로 나누어 각 disk에 나누어 저장
- 모든 disk에 입출력 부하 균등 분배
 - Parallel access
 - Performance 향상
- 한 Disk에서 장애 시, 데이터 손실 발생
 - Low reliability



RAID 1

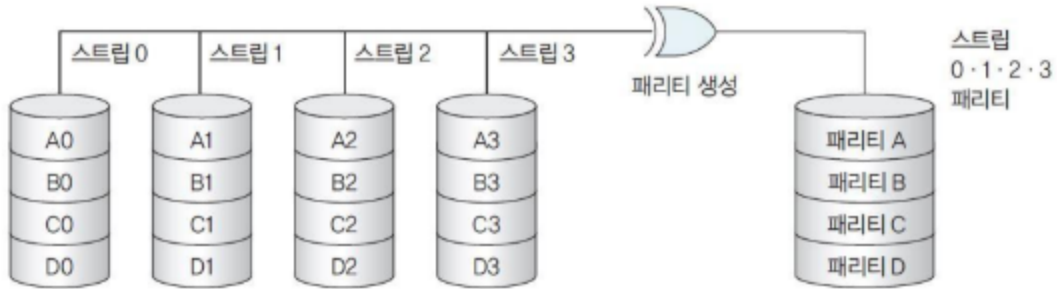
- Disk mirroring
 - 동일한 데이터를 mirroring disk에 중복 저장

- 최소 2개의 disk로 구성
 - 입출력은 둘 중 어느 disk에서도 가능
- 한 disk에 장애가 생겨도 데이터 손실 X
 - High reliability
- 가용 disk 용량 = (전체 disk 용량/2)



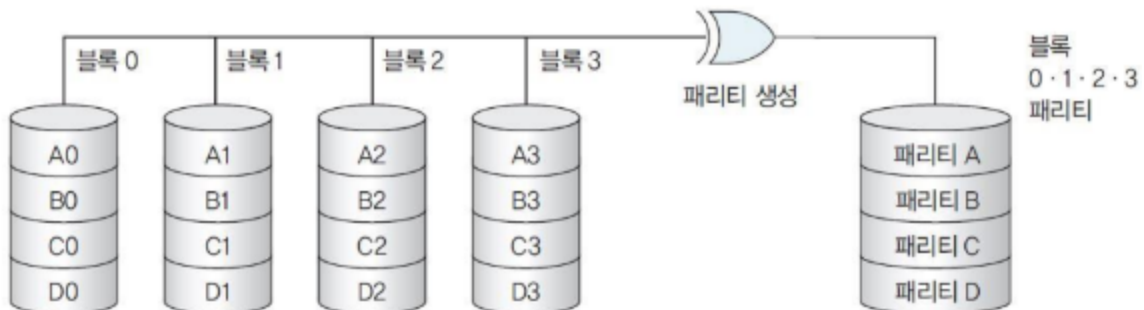
RAID 3

- RAID 0 + parity disk
 - Byte 단위 분할 저장
 - 모든 disk에 입출력 부하 균등 분배
 - Parallel access, Performance 향상
- 한 disk에 장애 발생 시, parity 정보를 이용하여 복구
- Write 시 parity 계산 필요
 - Overhead
 - Write가 몰릴 시, 병목현상 발생 가능



RAID 4

- RAID 3과 유사, 단 Block 단위로 분산 저장
 - 독립된 access 방법
 - Disk간 균등 분배가 안될 수도 있음
 - 한 disk에 장애 발생 시, parity 정보를 이용하여 복구
 - Write 시 parity 계산 필요
 - Overhead / Write가 몰릴 시 병목현상 발생 가능
- 병목 현상으로 성능 저하 가능
 - 한 disk에 입출력이 몰릴 때



RAID 5

- RAID 4와 유사
 - 독립된 access 방법
- Parity 정보를 각 disk들에 분산 저장

- Parity disk의 병목현상 문제 해소
- 현재 가장 널리 사용 되는 RAID level 중 하나
 - High performance and reliability

