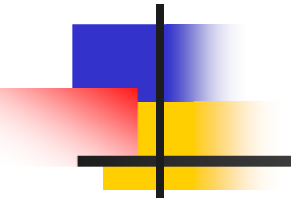
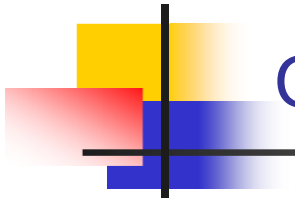


Operating Systems
(Mass-Storage Structure)

Chapter 11

These lecture materials are modified from the lecture notes written by A. Silberschatz, P. Galvin and G. Gagne.

August, 2022

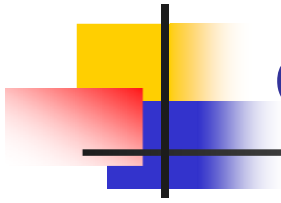


Outline

- Overview of Mass-Storage Structure
- DISK Scheduling
- Reliability

Overview of Mass-Storage Structure

Overview of Mass-Storage Structure



- 최신 컴퓨터의 주요 대용량 저장 시스템은 일반적으로 HDD(Hard Disk Drive)나 NVM(Non-Volatile Memory)등의 보조 저장 장치로 구성됨

- 컴퓨터 시스템에서 가장 일반적이고 중요한 저장 장치가 HDD와 NVM 장치이기 때문에, 두 가지 유형의 저장 장치에 대해 설명

HDD(Hard Disk Drive)

- HDD는 크게 spindle, platter, arm, arm assembly, read/write head로 이루어짐
- platter안이 여러개의 track으로 구성되어짐
- track의 최소 단위를 sector라고 함.

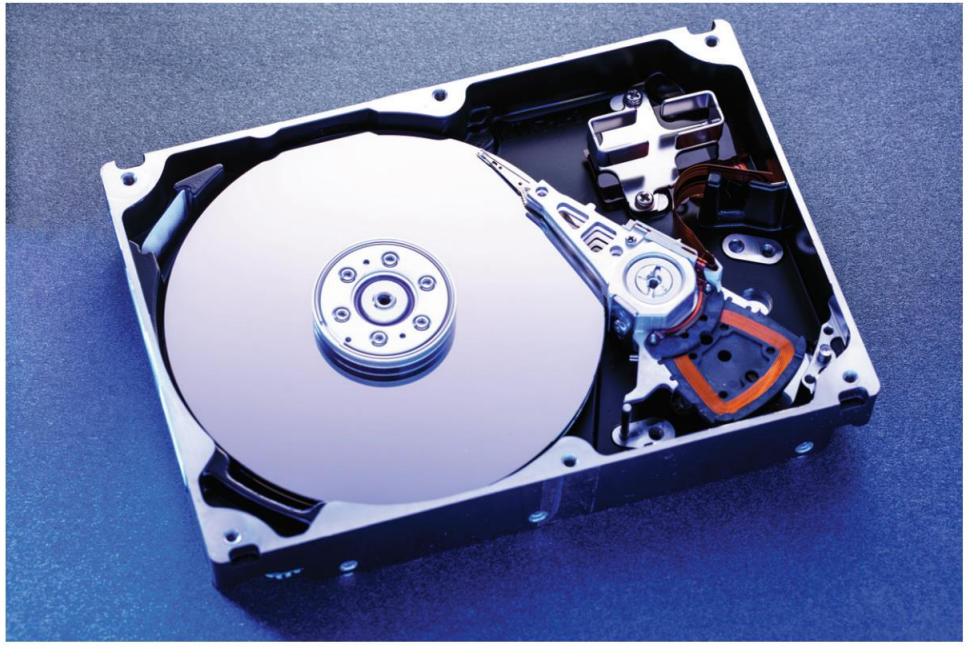


Figure 11.2 A 3.5-inch HDD with cover removed.

HDD(Hard Disk Drive)

- **Spindle :** 디스크를 회전시켜 내용을 읽고 쓰는 역할로, 흔히 들은 RPM이 스핀들 모터의 회전 속도를 의미
- **Platter :** 실제 데이터가 저장되는 위치, 여러 트랙으로 구성
  - 일반적으로 물리적인 손상 시 복구 불가능
- **Track :** 데이터가 저장되는 공간
- **Sector :** 데이터 저장의 최소 단위, 하나의 파일만 저장 가능
- **Cylinder :** 플래터 표면 동일 트랙들의 집합
- **Arm :** 헤드를 움직이는 역할
- **Read-Write Head :** 데이터를 저장 혹은 삭제하며, 데이터를 읽어들이는 역할도 수행
  - 가끔 데이터를 손상시킬 때가 있으며 이를 head crash라고 함.
- **Block :** 여러 섹터를 하나로 묶은 것

HDD(Hard Disk Drive)

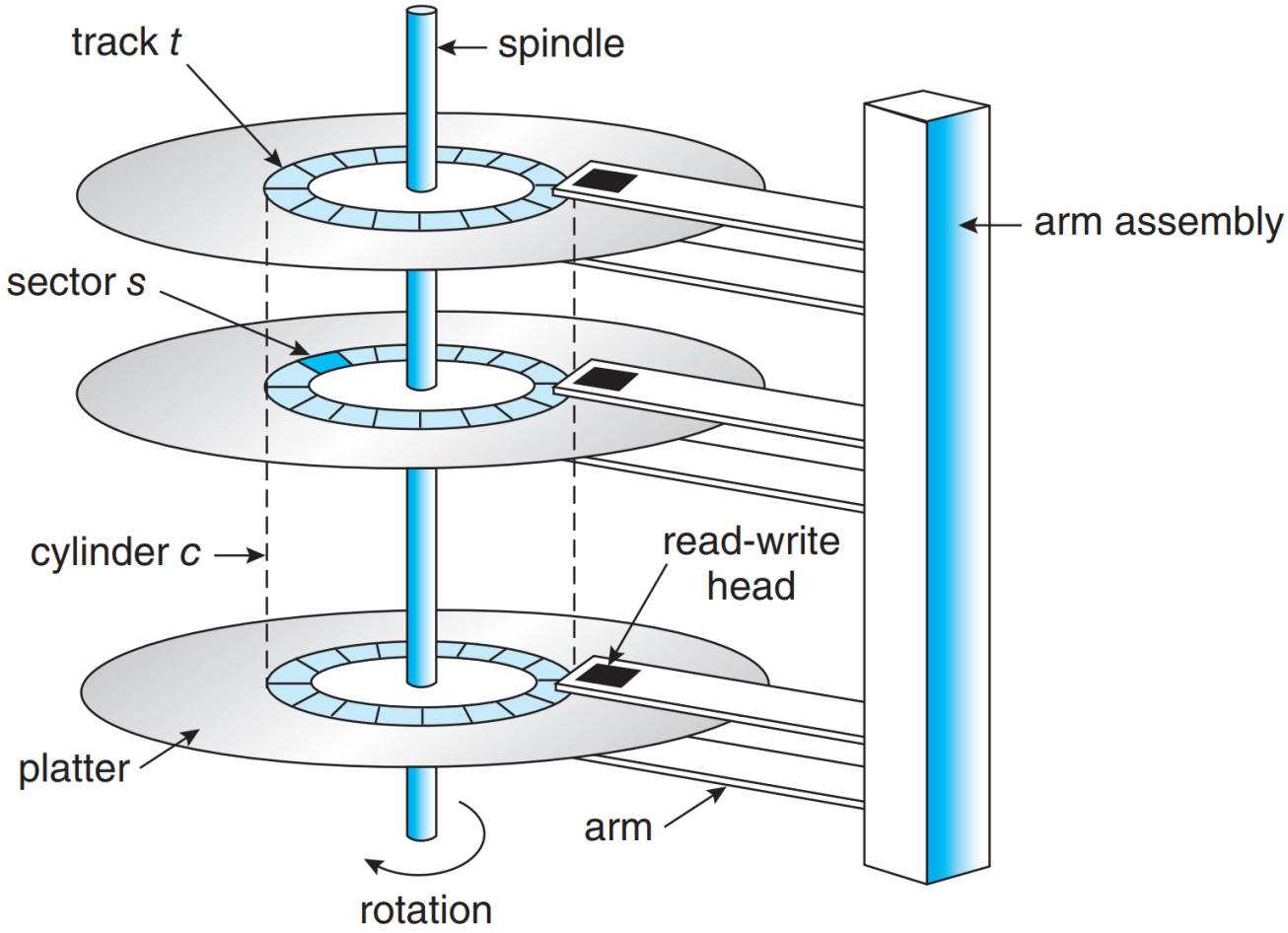
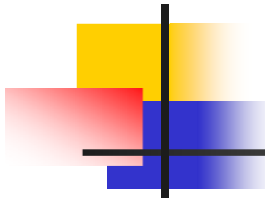


Figure 11.1 HDD moving-head disk mechanism.

HDD(Hard Disk Drive)

- **접근 시간 및 속도**
  - **전송 속도 (transfer rate) :**
    스토리지와 컴퓨터 간의 데이터 전송 흐름의 속도
  - **위치 지정 시간 (Positioning time) :**
    seek time + rotational latency
  - **탐색 시간 (seek time) :**
    Disk arm을 원하는 Cylinder로 이동하는데 필요한 시간
  - **회전 지연 시간 (rotational latency) :**
    원하는 Sector가 디스크 헤드 위치까지 회전하는데 걸리는 시간

NVM(Nonvolatile Memory Devices)



- 기계식이 아닌 전기식.
- 플래시 메모리 기반 NVM을 SSD (solid-state disk)라고 함.
- NVM 장치는 움직이는 부분이 없기 때문에 HDD보다 더 안정적, 탐색 시간이나 회전 대기 시간이 없기 때문에 더 빠를 수 있음.
  - 전력소비량이 적음
- HDD보다 용량 대비 가격이 비쌈
  - 그러나 NVM 장치의 용량 대비 가격이 하락하여 사용량이 급증

NVM(Nonvolatile Memory Devices)

Figure 11.3 A 3.5-inch SSD circuit board.

NAND semiconductors

- NAND 반도체는 아래의 특성 때문에 자체적인 저장 및 신뢰성 문제를 가짐.
  - 섹터와 유사한 ‘Page' 단위로 읽고 쓸 수 있지만 데이터를 덮어쓸 수 없음
  - 덮어쓰기 위해서는, 먼저 NAND 셀을 지워야함.
    삭제는 여러 페이지로 구성된 'Block' 단위로 이루어지며 읽기 또는 쓰기보다 시간이 더 걸림.
  - 쓸 때마다 마모가 발생함.
    그러므로 NAND NVM 수명은 연 단위가 아니라 DWPD (Drive Writes Per Day)로 측정됨.

NAND semiconductors

- NAND는 데이터를 한 번 쓴 후에 덮어쓸 수 없음.
- Erase가 발생하지 않는 동안,
  NAND 블록은 valid 페이지와 invalid 페이지를 포함함.

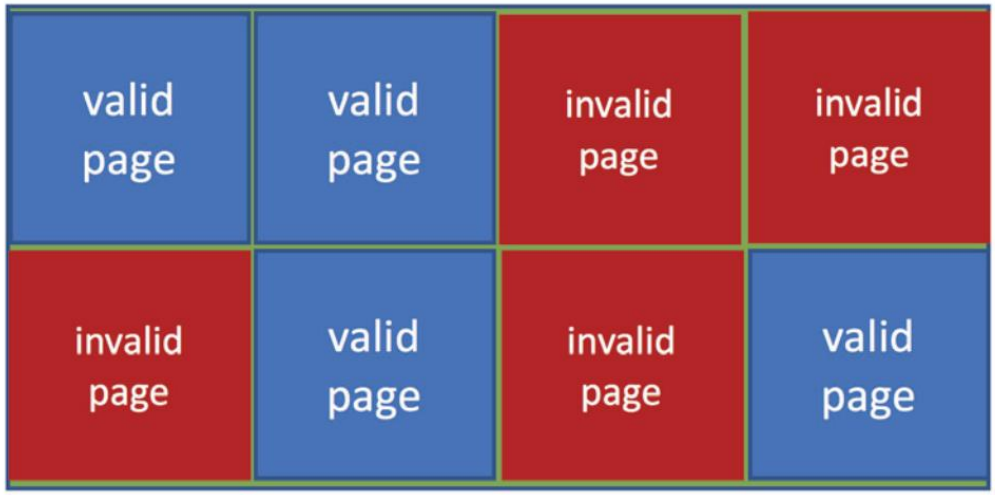
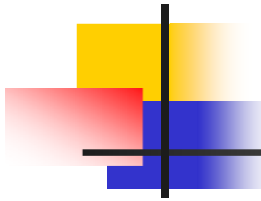


Figure 11.4 A NAND block with valid and invalid pages.

NAND semiconductors

- **FTL(Flash Translation Layer):**
  - SSD를 구성하는 NAND는 Block으로 구성되므로, OS의 논리적 주소와 NAND의 물리적 주소의 주소 매핑이 필요함.
  - FTL는 이러한 주소 매핑을 수행함.



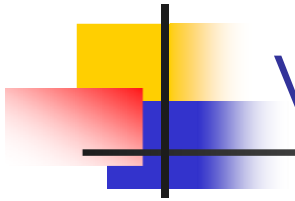
NAND semiconductors

- **Garbage-collection(GC):**
  - SSD의 모든 페이지가 기록된 경우를 가정
  - 이 때 Write는 Erase 가 발생할 때까지 기다린 다음 수행가능. 그러나 Erase는 매우 느림.
  - Invalid 페이지라고 체크만 해둔 뒤, 나중에 한번에 GC를 수행하여 invalid 페이지들을 소거하여 빈 페이지로 만듦.
  - 그럼 이 경우 페이지는 어디에 Write 하는가?

- **over-provisioning 공간**:
  - 장치가 꽉 찼을 경우, 페이지를 저장할 곳이 필요함. 이를 수행하기 위한 임시 작업 공간.

NAND semiconductors

- **Wear leveling**
  - 반복 쓰기로 인한 메모리 수명 단축 방지를 위해,
    FTL은 플래시 장치의 모든 블록에 데이터를 균등하게 사용되도록 함



Volatile Memory

- 저장된 정보를 유지하기 위해서는 지속적으로 전기가 필요한 컴퓨터 메모리.
  - 전기가 없으면 데이터가 휘발됨
  - 보통 메인 메모리로 사용함.

- SSD에 있는 DRAM에 FTL의 매핑 테이블을 저장함.
  - DRAM이 있는 SSD가 더욱 빠른 속도를 냄.

DISK Scheduling

HDD Scheduling

- HDD를 효율적으로 사용하기 위해서는 접근 시간을 최소화하고 대역폭을 최대화해야함.
- 데이터 I/O 요청들은 디스크의 큐에 들어감. 이 큐에서 어떤 요청부터 처리할 지를 스케쥴링함으로써 효율성을 높임.
- **대역폭(bandwidth) :** 시간당 한 번에 전송할 수 있는 크기

$$\frac{\text{전송된 총 바이트 수}}{\text{첫 번째 서비스 요청과 마지막 전송 완료 사이의 전체 시간}}$$

FCFS Scheduling

- 먼저 들어온 요청 섹터를 먼저 처리하는 방법 (FIFO)

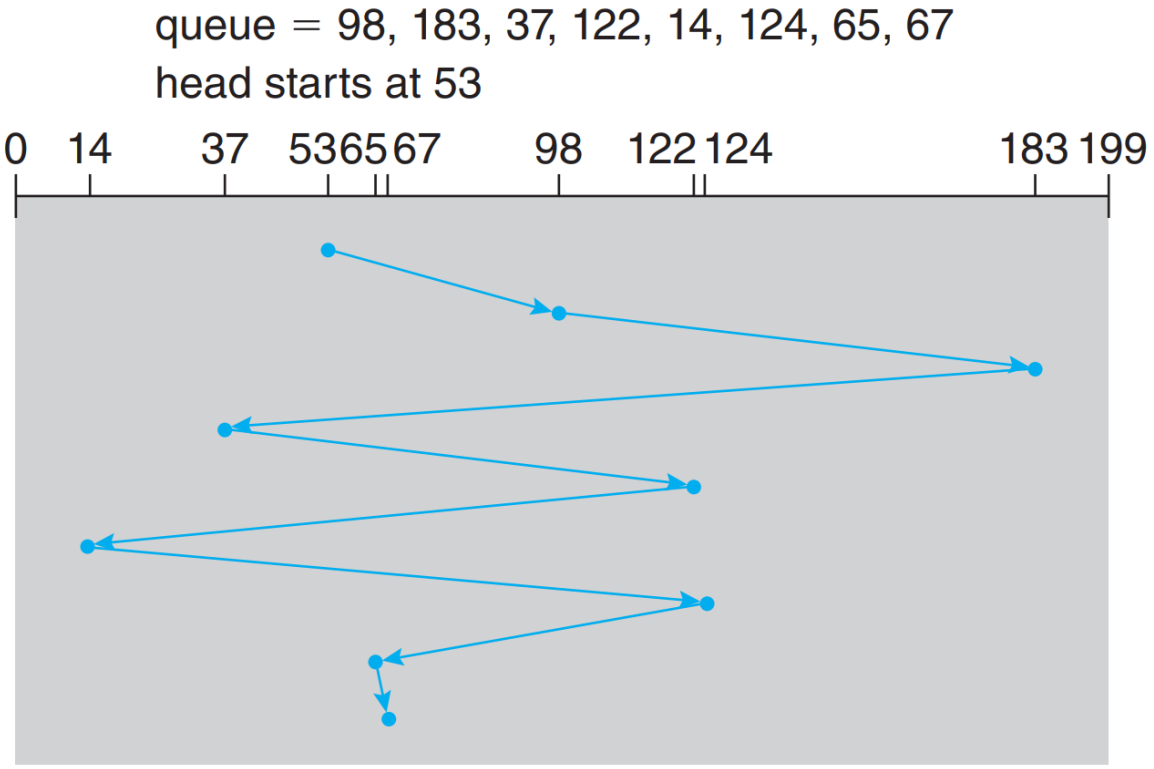
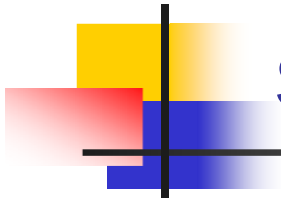


Figure 11.6 FCFS disk scheduling.



SCAN Scheduling

- Disk arm이 디스크의 한 끝에서 시작해 다른 끝으로 이동하여 가는 길에 있는 요청을 모두 처리함.
- 다른 한쪽 끝에 도달하면 역방향으로 이동하면서 오는 길에 있는 요청을 모두 처리함.
- 헤드는 디스크 양쪽을 왕복. 엘리베이터 알고리즘이라고도 부름.

SCAN Scheduling

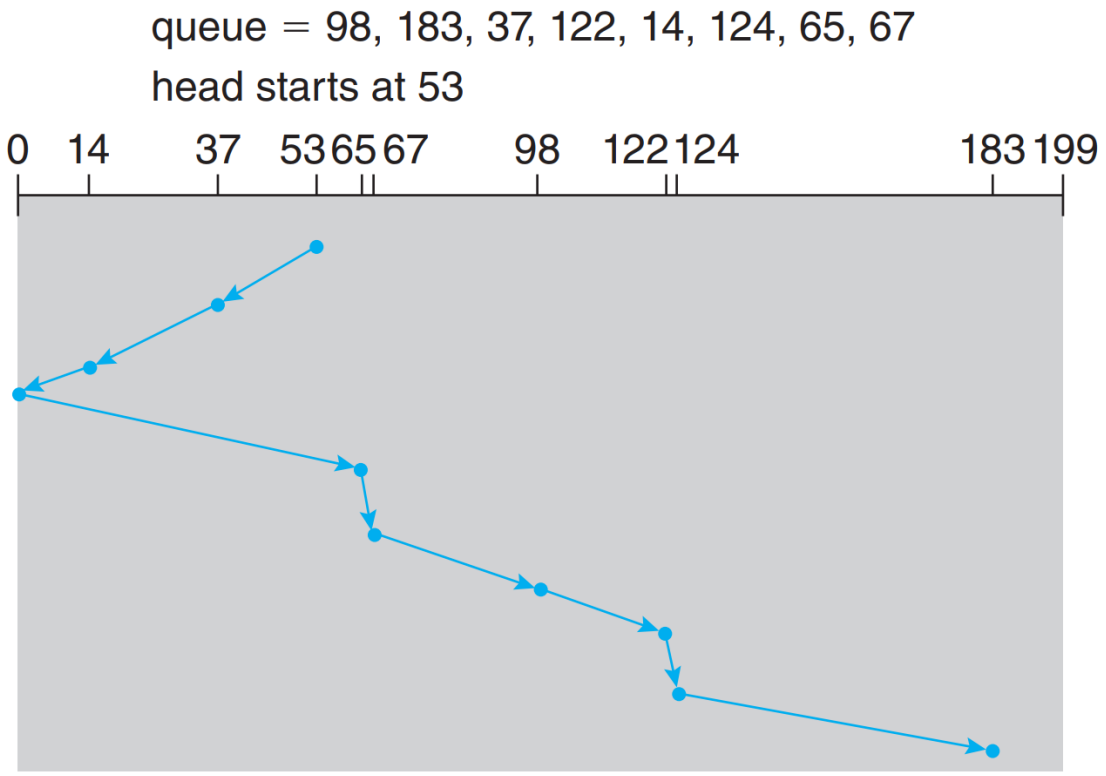
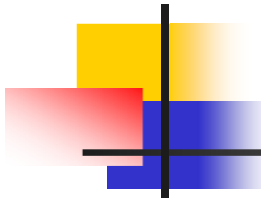


Figure 11.7 SCAN disk scheduling.



C-SCAN Scheduling(circular-SCAN)

- 각 요청에 걸리는 시간을 좀 더 균등하게 하기 위한 SCAN의 변형.
- Disk arm이 디스크의 한 끝에서 시작해 다른 끝으로 이동하여 가는 길에 있는 요청을 모두 처리하지만,
- 다른 한쪽 끝에 도달하면 처음 시작했던 자리로 다시 되돌아간 후, 요청을 다시 처리함.

C-SCAN Scheduling(circular-SCAN)

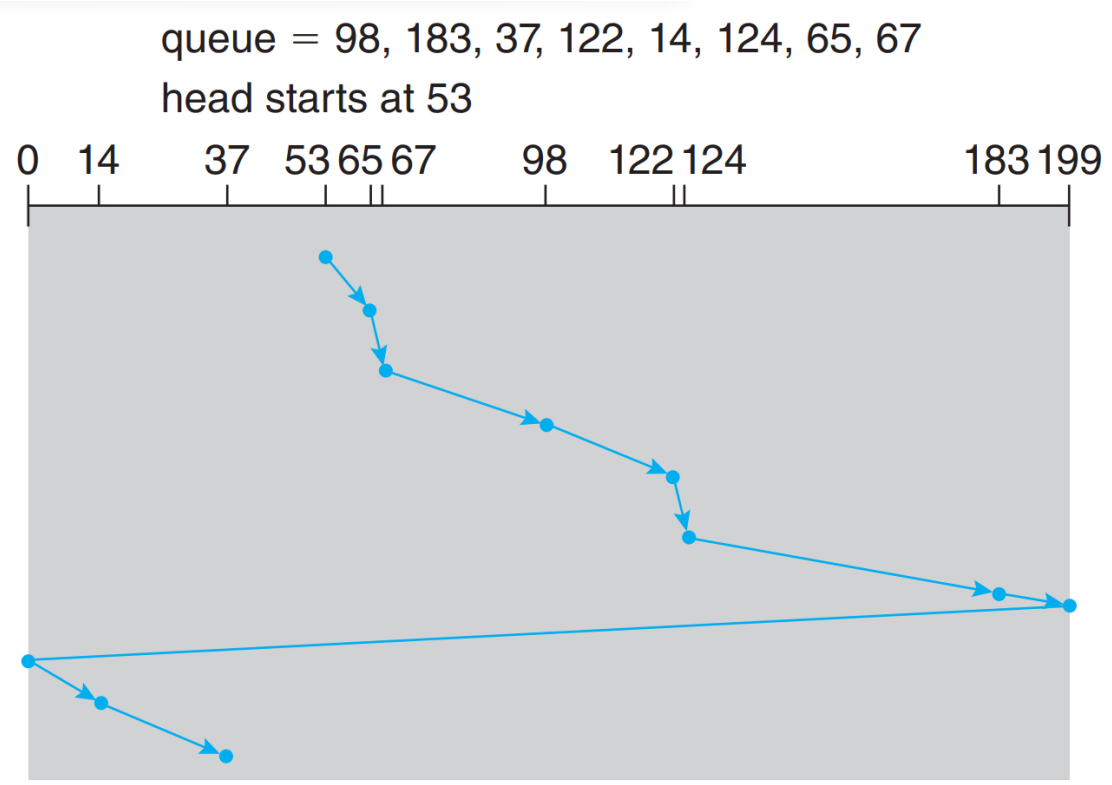
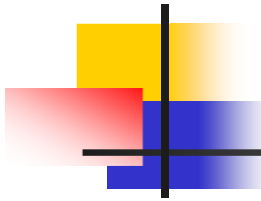
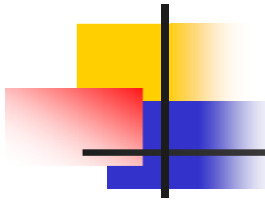


Figure 11.8 C-SCAN disk scheduling.

C-SCAN Scheduling(circular-SCAN)



- 각 요청에 걸리는 시간을 좀 더 균등하게 하기 위한 SCAN의 변형.
- Disk arm이 디스크의 한 끝에서 시작해 다른 끝으로 이동하여 가는 길에 있는 요청을 모두 처리하지만,
- 다른 한쪽 끝에 도달하면 처음 시작했던 자리로 다시 되돌아간 후, 요청을 다시 처리함.

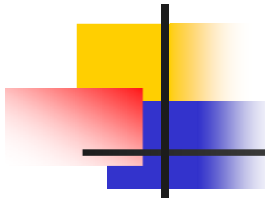


NVM Scheduling

- HDD 스케쥴링 알고리즘은 Positioning time을 최소화하는데 중점을 둠.
- 하지만 NVM 장치에는 움직이는 부분이 없으므로 일반적으로 간단한 FCFS 정책을 사용함.

- 예) **리눅스의 NOOP scheduler**
  - FCFS가 기본이지만, 인접한 쓰기 요청을 병합하도록 수정
  - NVM 장치의 쓰기 시간은 보통 플래시 메모리의 속성 때문에 일정하지 않으므로

Reliability



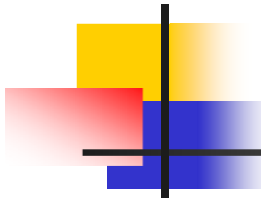
Parity bit

- 데이터는 시스템 내부에서 계속 전송되고 업데이트 됨. 이 과정에서 오류가 생길 수 있음.

- **패리티 비트 (parity bit)**를 사용하여 오류를 감지함.
- 고정 길이 워드의 값을 계산, 저장 및 비교하기 위해 나머지(%) 연산을 수행하는 체크섬(checksum)의 한 형태임.
- 패리티 비트는 오류 발생 여부는 알 수 있으나 오류 수정은 불가능.

Parity bit

- 짝수(even) 패리티
  - 전체 비트에서 1의 개수가 짝수가 되도록 패리티 비트를 정하는 방법
  - 1의 개수가 홀수이면 패리티 비트는 1
  - 1의 개수가 짝수이면 패리티 비트는 0
- 홀수(odd) 패리티
  - 전체 비트에서 1의 개수가 홀수가 되도록 패리티 비트를 정하는 방법
  - 1의 개수가 홀수이면 패리티 비트는 0
  - 1의 개수가 짝수이면 패리티 비트는 1
- 각 바이트당 1의 개수가 짝수/홀수가 되지 않으면 데이터의 변조가 생긴 것으로 판단

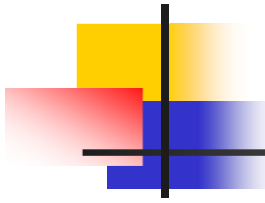


RAID Structure

- **RAID :** Redundant Arrays of Independent Disks
  (복수 배열 독립 디스크)

- 여러 디스크를 묶어 하나의 디스크처럼 사용하는 디스크 구성 기술
  - 병렬로 작동시켜 데이터의 읽기 및 쓰기 속도를 향상

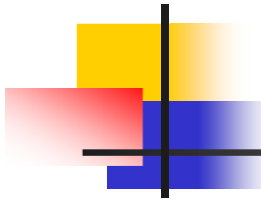
- **신뢰성을 높여 데이터 보호 능력을 향상함.**



RAID Structure

- **Mirroring:**
  한 디스크의 데이터를 다른 디스크로 중복 저장함.
  어떤 디스크가 잘못되어도 데이터를 보존할 수 있게 함.
  신뢰성을 높여주지만 비용이 비쌈.

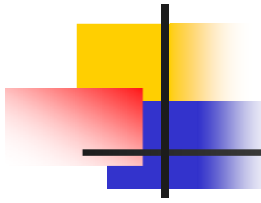
- **Stripping:**
  하나의 데이터를 여러 디스크에 나누어 저장함.
  저장되는 속도 및 공간 효율이 좋음.
  그러나 디스크 중 하나가 고장나면 데이터 전체를
  못 쓰므로 신뢰도를 낮춤.



RAID Structure

- RAID Levels : 비용 대비 성능에 따라 레벨로 분류함

- **RAID 0 :** 데이터를 여러 디스크에 스트라이핑하는 방식
  한 디스크에서 장애 발생 시 데이터가 모두 손실됨
- **RAID 1 :** 디스크에 기록된 정보를 모두 미러링하여 저장함
- **RAID 4 :** 데이터를 여러 디스크에 스트라이핑 함
  패리티 디스크를 추가하고, 패리티 비트를 넣어 디스크에 에러가 발생하지 않았는지 검사함
  - 최소 3 개의 디스크 필요



RAID Structure

- RAID Levels : 비용 대비 성능에 따라 레벨로 분류함

- **RAID 5 :** 각 디스크에 패리티 비트를 추가함
  - 최소 3 개의 디스크 필요
- **RAID 6 :** 각 디스크에 패리티 비트 뿐만 아니라 2차 패리티 비트 역시 추가하여 더 정교하게 에러를 감지함
  - 최소 4개의 디스크 필요

RAID Structure

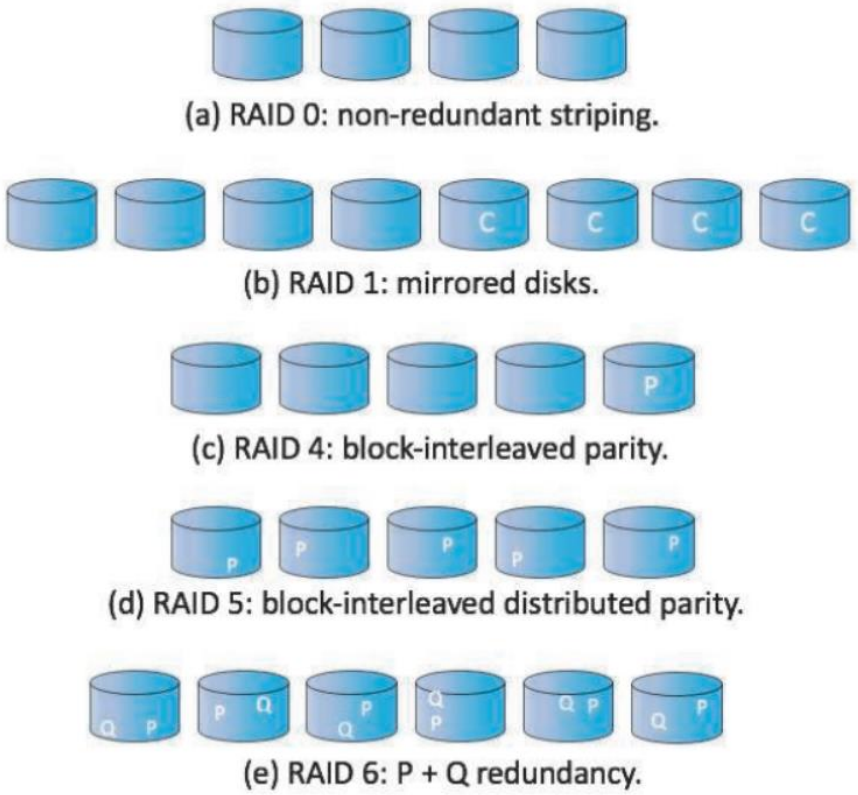
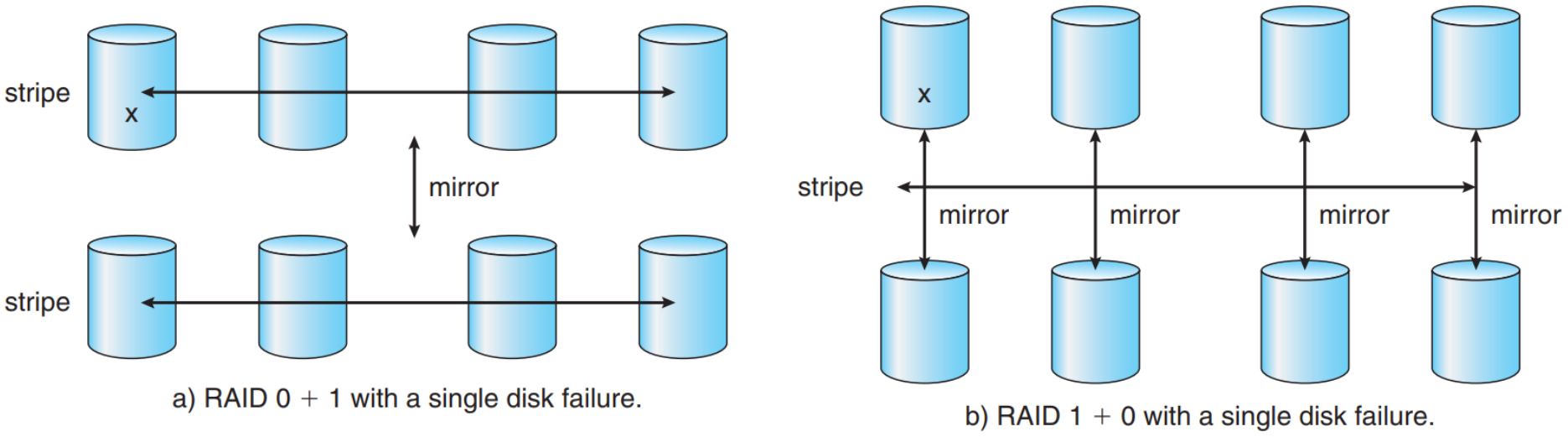


Figure 11.15 RAID levels.

- RAID Levels :
  비용 대비 성능에 따라 레벨로 분류함

RAID Structure

- **RAID 0 + 1 :** 스트라이핑한 디스크를 미러링 하는 방식.
- **RAID 1 + 0 :** 미러링한 디스크를 스트라이핑 하는 방식
  - 안정성: RAID 1 + 0 > RAID 0 + 1



a) RAID 0 + 1 with a single disk failure.

b) RAID 1 + 0 with a single disk failure.

Figure 11.16 RAID 0 + 1 and 1 + 0 with a single disk failure.

Chapter 11
Finish