



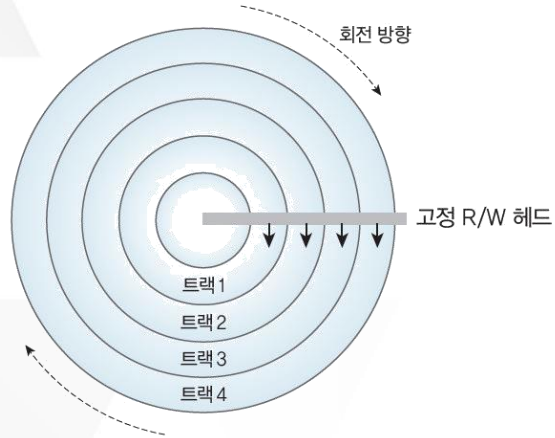
1

자기 디스크의 물리적 구조

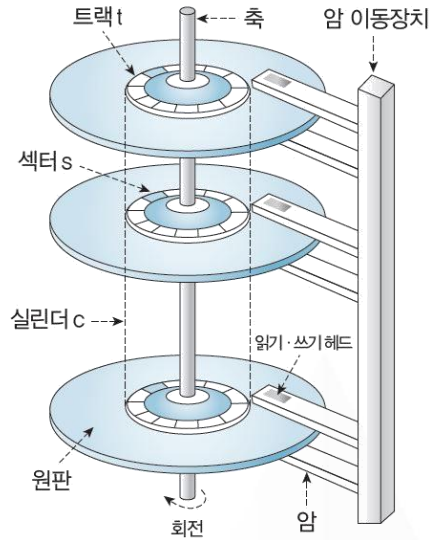
1 자기 디스크의 물리적 구조

1 디스크 구조

[디스크의 종류와 구조]



(a) 고정 헤드 디스크



(b) 이동 헤드 디스크장치

※ 출처: 그림으로 배우는 구조와 원리 운영체제, 구현회, 한빛아카데미, 2016

1 자기 디스크의 물리적 구조

2 디스크 시스템

- ▶ 디스크 드라이버, 프로세서, 디스크 제어기로 구분
 - 디스크 드라이버
: 구동 모터, 액세스 암 이동장치, 입출력 헤드 부분의 기계적인 부분 담당
 - 프로세서
: 컴퓨터의 논리적인 상호작용, 즉 데이터의 위치 (디스크 주소)와 버퍼, 판독, 기록 등 관리

1 자기 디스크의 물리적 구조

2 디스크 시스템

- ▶ 디스크 드라이버, 프로세서, 디스크 제어기로 구분
 - 디스크 제어기
 - 디스크 드라이버의 인터페이스 역할
 - 프로세서에서 명령을 받아 디스크 드라이버 동작, 디스크 드라이버는 탐색 seek, 기록, 판독 등 명령 수행

1 자기 디스크의 물리적 구조

2 디스크 시스템

- ▶ 디스크 드라이버, 프로세서, 디스크 제어기로 구분
 - 디스크의 정보는 드라이버 번호, 표면 번호, 트랙 번호 등으로 나누는 디스크 주소로 참조
 - 트랙
: 원형 평판 표면에 데이터를 저장할 수 있는 동심원을 가리킴, 자기장의 간섭을 줄이거나 헤드를 정렬하려고 트랙 사이에 일정한 공간을 두어 트랙 구분

1 자기 디스크의 물리적 구조

2 디스크 시스템

- ▶ 디스크 드라이버, 프로세서, 디스크 제어기로 구분
 - 디스크의 정보는 드라이버 번호, 표면 번호, 트랙 번호 등으로 나누는 디스크 주소로 참조
 - 실린더
: 동일한 동심원으로 구성된 모든 트랙, 즉 동일한 위치에 있는 모든 트랙의 집합을 의미, 헤드의 움직임 없이 액세스할 수 있는 드라이브의 모든 트랙에 해당

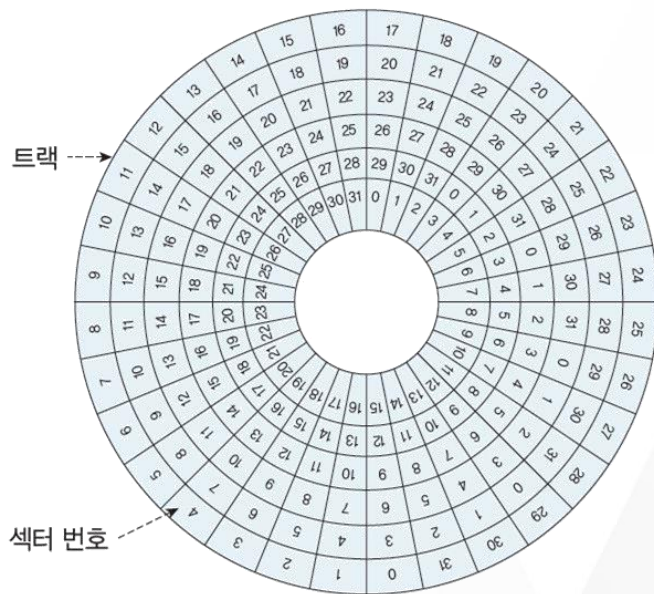
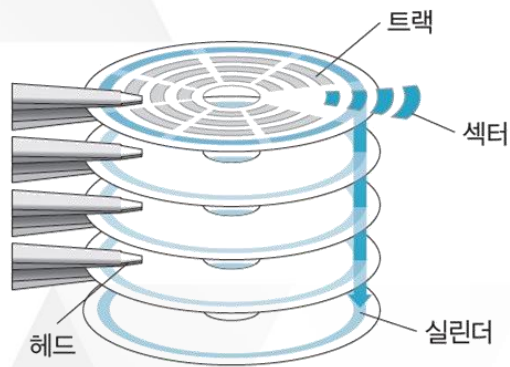
1 자기 디스크의 물리적 구조

2 디스크 시스템

- ▶ 디스크 드라이버, 프로세서, 디스크 제어기로 구분
 - 디스크의 정보는 드라이버 번호, 표면 번호, 트랙 번호 등으로 나누는 디스크 주소로 참조
 - 섹터
: 트랙을 부채꼴 모양으로 나눈 조각을 의미, 트랙 내의 정보는 블록을 구성하고, 이 블록이 하드웨어적으로 크기가 고정되었을 때가 바로 섹터임, 섹터는 데이터 기록이나 전송의 기본 단위로 일반적으로 512바이트의 데이터 영역으로 구성됨, 섹터는 고유 번호가 있어 디스크에 저장된 데이터의 위치 식별 가능

1 자기 디스크의 물리적 구조

3 디스크의 논리적 구조

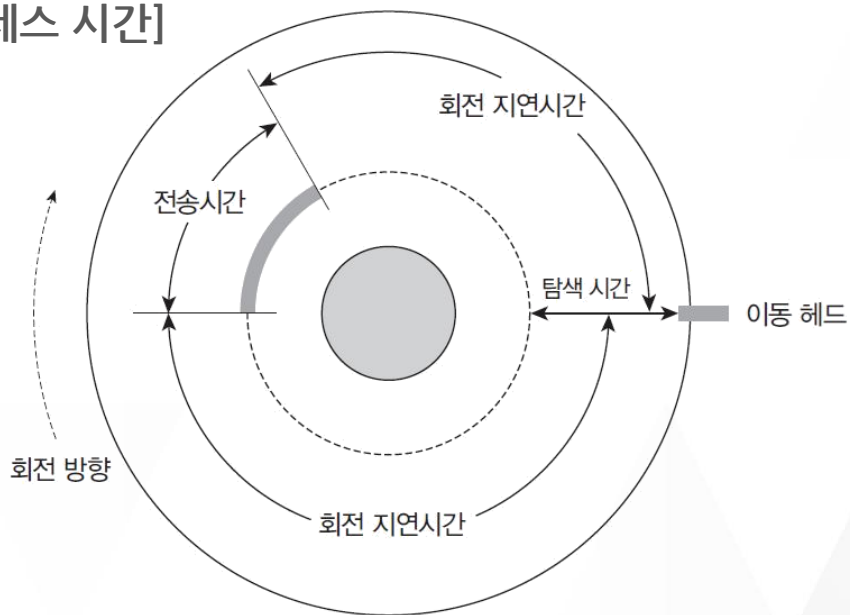


※ 출처: 그림으로 배우는 구조와 원리 운영체제, 구현회, 한빛아카데미, 2016

1 자기 디스크의 물리적 구조

4 디스크 액세스 시간의 개념

[디스크 액세스 시간]



※ 출처: 그림으로 배우는 구조와 원리 운영체제, 구현회, 한빛아카데미, 2016

1 자기 디스크의 물리적 구조

5 디스크 액세스 시간의 계산

- ▶ 이동 디스크 : 탐색 시간+회전 지연시간+전송시간
- ▶ 고정 헤드 디스크 : 회전 지연시간+전송시간

1 자기 디스크의 물리적 구조

5 디스크 액세스 시간의 계산

[디스크 액세스 시간을 계산하는 예]

- 탐색 시간 : 50밀리초
- 회전 지연시간 : 16.8밀리초
- 전송시간 : 0.00094밀리초/바이트 [1KB 전송시간 : $0.96256 (= 0.00094 \times 1,024)$]

(a) 탐색 시간, 회전 지연시간, 전송시간 예

- 이동 헤드 디스크의 데이터 액세스 시간 = $50 + 16.8 + 0.96256$
- 고정 헤드 디스크의 데이터 액세스 시간 = $16.8 + 0.96256$

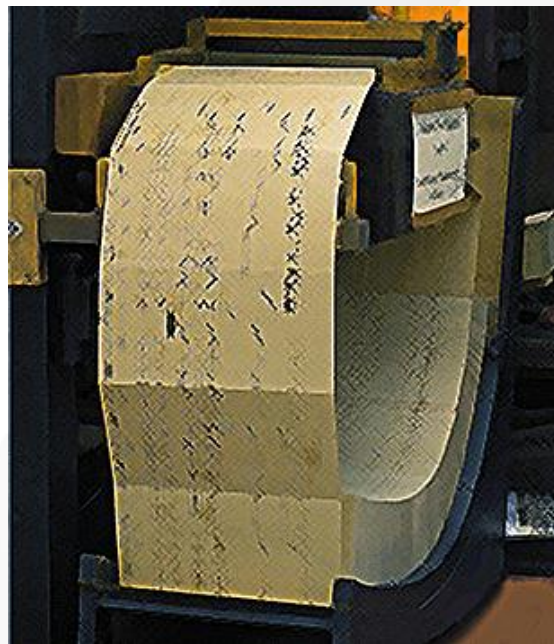
(b) 디스크 종류에 따른 데이터 액세스 시간

※ 출처: 그림으로 배우는 구조와 원리 운영체제, 구현회, 한빛아카데미, 2016

1 자기 디스크의 물리적 구조

6 초기 컴퓨터에서 사용 하던 천공카드

- ▶ 초창기 컴퓨터에서는 종이에 일정한 패턴의 구멍을 뚫어 데이터를 기록하는 종이 테이프나 천공카드를 주로 사용함, 하지만 이들은 많은 용량을 기록하기 어려우며 보관이 불편하다는 단점이 있어 그 대안으로 제시된 것이 자성 물질로 코팅한 플라스틱 테이프를 이용하는 자기 테이프 기록 장치인데, 이는 비교적 대용량이라는 장점이 있었으나 데이터를 읽어 들이는 속도가 너무 느려서 불편해서 용량이 크면서 속도도 빠른 데이터 저장장치가 절실히 필요한 상황이었음

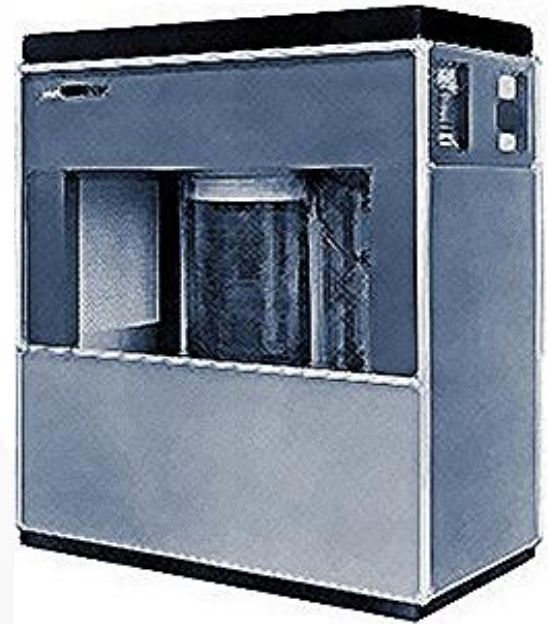


※출처: <http://it.donga.com/21796/>

1 자기 디스크의 물리적 구조

7 4.8MB의 대용량 저장장치

- ▶ 1956년, 미국 IBM사에서 IBM 305 RAMAC이라는 새로운 컴퓨터를 개발했는데, 여기에는 이전에는 볼 수 없던 전혀 새로운 형태의 저장 장치가 달려있었음, 이 장치는 전공카드나 테이프가 아니라 자성 물질로 덮인 플래터(Platter : 금속 디스크)를 여러 장 쌓아 올린 구조를 가지고, 이 장치는 당시에는 매우 큰 용량이었던 4.8MB의 데이터를 저장할 수 있었으며, 분당 1,200RPM으로 플래터를 회전시키며 고속으로 데이터를 읽거나 쓸 수 있어 화제를 불러일으켰음, 이것이 바로 세계 최초의 하드디스크드라이브(Hard Disk Drive, 이하 HDD) 임



※출처: <http://it.donga.com/21796/>

1 자기 디스크의 물리적 구조

7 4.8MB의 대용량 저장장치

- ▶ 개발 당초에 이 장치는 자기 디스크 기억장치, 혹은 단순히 디스크 장치로 불리곤 했음, 그리고 1973년에 IBM에서 내놓은 IBM 3340 저장장치의 개발코드명이었던 윈체스터(Winchester)가 유명해지면서 윈체스터 디스크라고 불리기도 했음, 하지만 1980년을 전후해 휴대용 저장 장치인 플로피(Floppy : 부드러운) 디스크가 많이 쓰이게 되면서 이와 대비되는 개념으로 하드(Hard: 단단한)디스크라는 용어가 등장하고 컴퓨터 내부에 설치되는 대용량 자기 디스크 기반 저장 장치를 지칭하는 일반 명사가 됨



※출처: <http://it.donga.com/21796/>

1 자기 디스크의 물리적 구조

8 표준화

- ▶ 개발 초기의 HDD는 기업용, 혹은 국가 기관용 대형 컴퓨터에만 쓰였기 때문에 가격이 자동차 몇 대 수준에 달할 정도로 비쌌으며 크기도 컸음, IBM 305 RAMAC에 탑재된 최초의 HDD는 지름 24인치(약 60cm) 짜리 플래터 50장으로 구성되어 있어서 장치 전체의 크기가 소형 냉장고나 세탁기와 비슷할 정도였음



※출처: <http://it.donga.com/21796/>

1 자기 디스크의 물리적 구조

8 표준화

- ▶ PC(개인용 컴퓨터)에 HDD의 탑재가 일반화 된 것은 1980년에 미국의 씨게이트(Seagate)사가 개발한 ST-506의 출시 이후로, ST-506은 지름 5.25인치 (약 13cm)의 플래터를 내장하고 5MB의 용량을 갖추고 있었음, ST-506는 PC에도 내장할 수 있을 정도로 크기가 작았고 가격도 1,500달러 정도로 비교적 저렴한 편이었음, ST-506이 인기를 끌면서 이후에 출시된 타사의 PC용 HDD도 ST-506와 호환되는 기본 설계 및 데이터 입출력 구조를 갖추게 되었고, 이는 HDD의 표준 규격으로 자리잡게 됨



※출처: <http://it.donga.com/21796/>

2 디스크 인터리빙

- ▶ 디스블록은 트랙과 섹터로 분할되기 때문에 바깥쪽 트랙의 블록은 안쪽 트랙의 블록보다 면적이 크며, 디스크가 항상 동일한 속도로 회전할 경우 바깥쪽 블록의 데이터 기록밀도는 그 블록이 가지고 있는 면적에 비해 적다는 의미
- ▶ 따라서 바깥쪽 트랙의 기억공간은 많이 낭비되고 있는 셈으로 이를 해결하기 위해 최근 바깥 트랙의 블록 수를 면적에 비례해 늘려주어 디스크 전체에 걸쳐 동일한 데이터 기록밀도를 유지하도록 하고 있고, 쓰기/읽기 헤드가 디스크의 바깥쪽 트랙으로 움직이면 디스크의 회전속도를 감속시키거나 데이터의 전송 속도를 증가시키도록 하고 있음

1 디스크 인터리빙 개념

- ▶ 디스크 인터리빙은 디스크의 입출력 성능에 중요한 역할을 하며 데이터가 CPU에 의해 디스크 제어기로부터 기억 장치로 전송되는 동안 다음 섹터는 디스크 헤드 밑을 통과해 버리고 디스크 제어기에는 새로운 비트가 도착하게 됨
- ▶ 단순한 디스크 제어기는 동시에 입력과 출력을 소화해 내지 못하므로 해서 기억 장치 전송이 일어나고 있는 동안 디스크 헤드 아래를 지나는 섹터는 잃어버리게 됨

- ▶ 그 결과 디스크 제어기는 매번 다른 블록을 읽음에 따라서 완전한 트랙을 읽기 위해서는 두 번의 디스크 플래터의 회전이 필요하게 됨
- ▶ 그 이유는 한번은 짝수 번 블록을 위한 회전이고 또 한번은 홀수 번 블록을 위한 회전이기 때문임

- ▶ 만약에 버스를 통해서 디스크 제어기로부터 주기억장치로 블록을 전송하는 시간이 디스크로부터 블록을 읽는 속도보다 오래 걸리면 한 블록을 읽어낸 후 두 블록을 건너뛰도록 해야 함
- ▶ 디스크 인터리빙은 디스크 제어기에게 기억 장치로 데이터를 전송하기 위한 시간을 주기 위해 블록을 건너뛰도록 하는 기법

- ▶ 즉 연속적으로 번호가 붙여진 디스크 블록을
순서적으로 읽을 때 디스크 제어기와 디스크 구동기
사이의 전송 부하 때문에 각 블록 사이에 지연 시간이
존재하게 되는데 지연 시간이 경과된 후에 헤드가
정확한 블록에 위치하도록 논리적으로 연속된 블록들을
디스크 표면상에서 일정한 간격으로 건너뛰게 함

3 디스크 가용공간

1 디스크 공간 할당

- ▶ 디스크에서는 일반적으로 블록의 크기를 4K로 사용하고 있음
- ▶ 예를 들어 블록의 크기가 4K이고 크기가 1K인 파일이 있다면 이 파일을 저장하기 위해서는 3개의 블록 할당 필요
- ▶ 이유는 각각 4K씩 2개의 블록과 마지막 남은 3K를 저장하기 위하여 1개의 블록이 필요하기 때문
- ▶ 여기서 마지막 4K 블록에 3K 를 할당하고 남은 1K를 디스크 기억 공간의 단편화라 부름

1 디스크 공간 할당

- ▶ 일반적으로 디스크에서 파일의 블록은 분산 할당 방법을 사용하고 있기 때문에 전체 디스크 상에 분산 적재되어 있음
- ▶ 따라서 시간이 지날수록 파일의 삭제와 삽입이 반복되면서 단편화된 조각들이 늘어나게 되므로 디스크 공간의 효율성이 떨어지므로 주기적으로 디스크 공간을 압축하는 과정이 필요

2 디스크 가용 공간 관리

- ▶ 디스크 공간은 한정되어 있기 때문에 새로운 파일들을 위하여 빈 공간을 효율적으로 관리할 필요가 있음
- ▶ 디스크상의 빈 공간의 상태를 관리하기 위해서 시스템은 가용 공간 리스트를 유지하여 비어 있는 모든 디스크의 블록들을 등록함
- ▶ 새로운 파일을 생성하기 위해서 시스템은 가용공간 리스트를 탐색하고 공간을 할당하여 새로운 파일들을 배정함
- ▶ 공간이 할당된 후 가용 공간 리스트를 갱신하고 파일이 삭제되면 이 디스크 공간은 다시 가용 공간 리스트에 추가