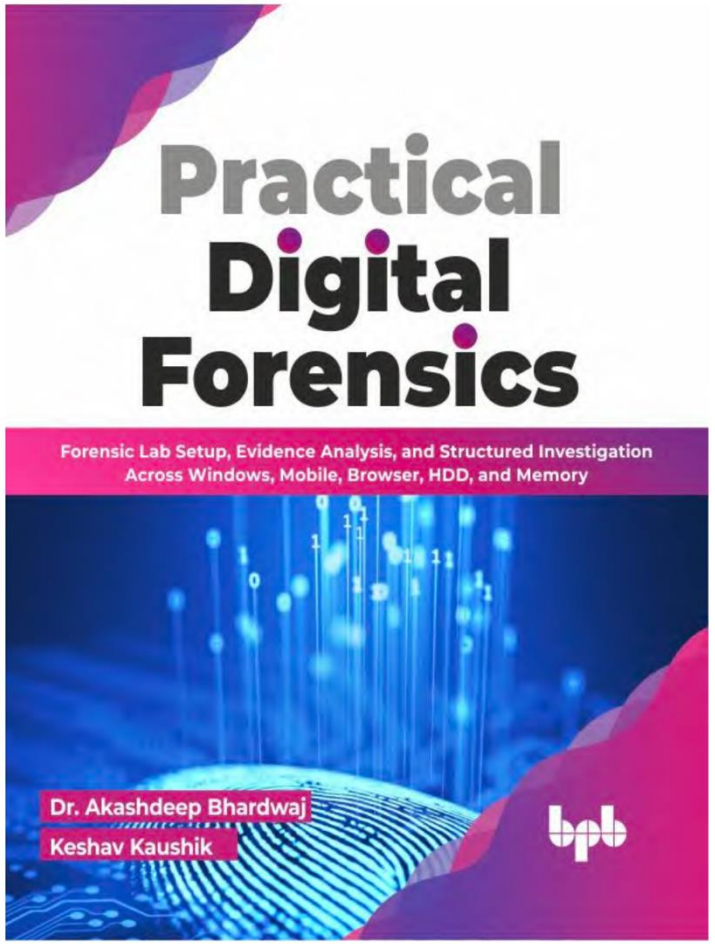
Hard Disks and File Systems

유튜브 주소: https://youtu.be/vKCwTLCR3D0

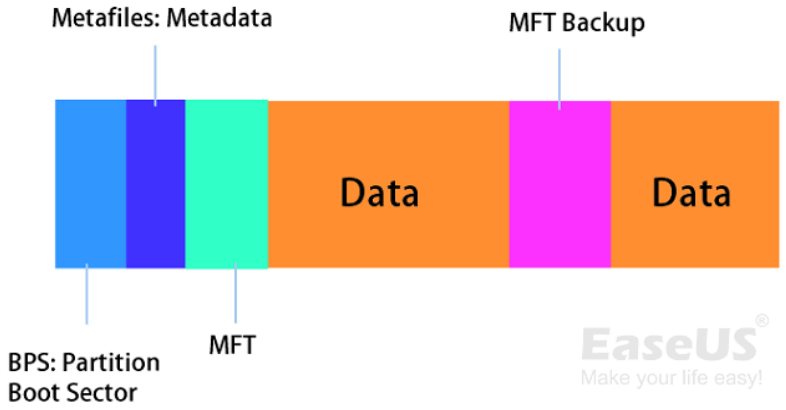
CHAPTER3: Hard Disks and File Systems

- **포렌식을 위해 필요한 기본적인 개념들에 대한 이해를 돕는 챕터**
  - 파일시스템, 파일 구성 방식, 저장 장치 종류 등에 대해 학습
- 학습 항목
  - 숫자 체계의 종류 (Different number system)
  - 인코딩 방식 (Encoding schema)
  - **파일 구조와 복원 (File carving and structure)**
  - **파일 메타데이터 (File metadata)**
  - **해시 분석 (Hash analysis)**
  - 시스템 메모리 (System memory)
  - **저장 장치 (Storage)**
  - **파일 시스템 (Filesystem)**
  - 클라우드 컴퓨팅 (Cloud computing)
  - **윈도우 운영체제 (Windows OS)**
  - 네트워킹 (Networking)



Hard disk and file systems

- 파일 시스템
  - **파일 시스템(FS)은 데이터가 저장되는 방식과 위치를 결정**
  - 논리적인 저장 단위를 사용하여, 파일을 디렉터리로 그룹화(폴더 구조 생성)
  - 운영체제와 사용자가 파일을 구조적으로 관리하고 접근할 수 있도록 지원
  - 대표적인 파일 시스템: NTFS, FAT32, HFS+



NTFS 파일 시스템 구조

FAT32 파일 시스템 구조

Hard disk and file systems

- 파일 시스템 핵심 역할
  - 메타데이터 관리
    - 파일 이름, 생성일, 크기 등
  - 저장소 관리
    - 데이터 블록 및 섹터의 위치 지정
  - 디렉터리 관리
    - 계층적 폴더 구성
  - 접근 제어
    - 사용자 및 앱이 데이터에 접근하는 방식 제어
    - E.g. 파일 시스템 없이는 프로그램 제거, 파일 복구, 이름 중복 방지 등이 불가능

Hard disk and file systems

- 포렌식 관점에서 본 파일 시스템
  - **삭제된 데이터 복구 시 파일 시스템 분석이 핵심**
    - 삭제된 데이터의 위치와 상태를 추적하기 위해선 파일 시스템 분석 필요
    - 블록/섹터 수준의 구조 분석을 통해 조각난 파일도 복원 가능
    - 디스크 캐시, 커널 연동, 사용자 API도 분석의 대상(고급 분석)
  - **즉, 파일 시스템 내부 분석을 통해 삭제된 파일 추적 가능**
    - **파일이 지워져도, 파일 시스템 내부에 일부 정보가 남아있을 수 있음**
    - E.g. 메타데이터, 블록 위치 정보, 파일 할당 테이블 등
    - 해당 정보들을 분석하여 파일 복구 시도 가능

File systems

- 파일 시스템 구조
  - **파일 시스템은 계층적 구조로 구성**
  - 상위 폴더 -> 하위 폴더 or 파일로 연결되는 트리 구조로 구성
  - 운영체제는 이 구조를 기반으로 파일 저장, 탐색, 접근 권한 관리 등을 수행
  - 사용자는 루트(최상위 폴더)부터 경로 기반으로 파일에 접근

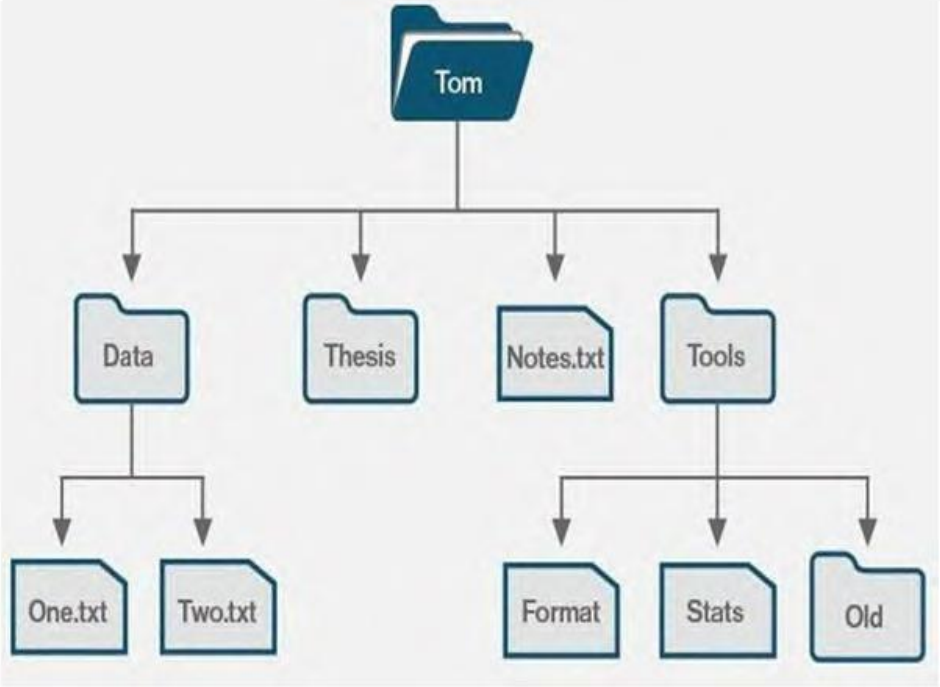
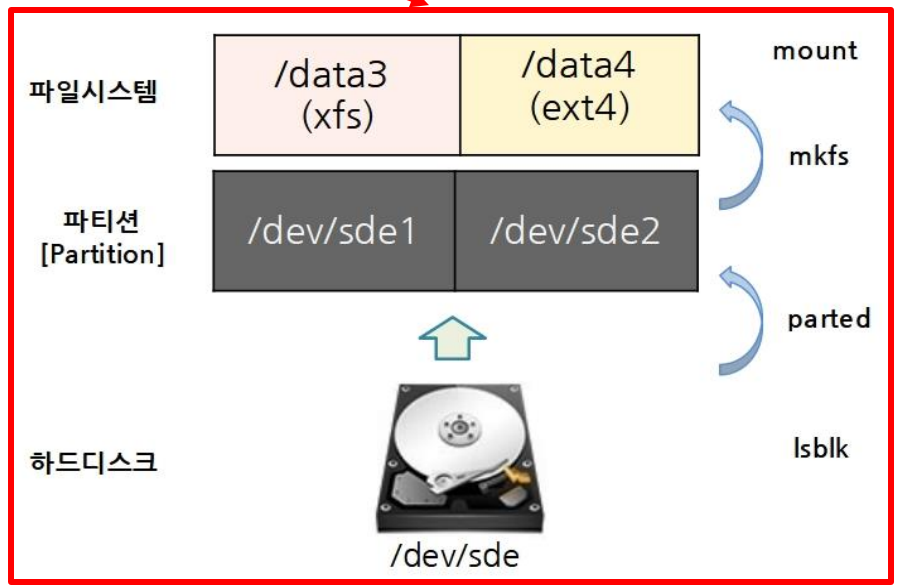


Figure 3.1: *File system*

File systems

- 파일 시스템과 파티션의 관계
  - **파일 시스템은 논리적 구조를 지님(운영체제가 파일을 다루는 기준이 되는)**
  - 하나의 물리 디스크에 여러 개의 파티션 존재 가능
  - **각 파티션은 고유한 파일 시스템(NTFS, FAT 등)을 지님**
  - E.g. 2개의 파티션에 각각 다른 파일 시스템 적용 가능
    - BUT 일반적인 설정은 아님(Windows의 경우)
    - 리눅스에선 파티션 별로 다른 파일 시스템을 적용하는 경우 있음
  - 대부분의 응용 프로그램은 파일 시스템이 없으면 실행 불가
  - 운영체제별 파일 시스템 호환성 차이 존재
    - E.g. MacOS용 프로그램은 Windows에서 실행 불가



File systems

- 주요 파일 시스템 비교

파일 시스템	주요 특징	사용 환경
FAT32	4GB 제한, 폭넓은 호환성, 오래된 구조	USB, 카메라, 게임기
exFAT	FAT32 개선, 대용량 지원, 경량 구조	최신 USB, SD카드
NTFS	높은 보안성, 복구 가능성 높음, 대용량 파일/디스크 지원	Windows 시스템 디스크
HFS+	MacOS 전용, 안정적인 트리 구조	구형 Mac 저장 장치
UDF	광학 매체(CD/DVD) 호환, 범용성	디스크, 멀티 플랫폼
GFS	공유 저장소, 다중 서버 간 파일 접근 지원	고가용성 클러스터 시스템

- MacOS는 HFS+를 사용했으나, 현재는 APFS로 넘어가는 추세
  - MacOS 10.13 High Sierra부터 APFS 도입(2017년)
- UDF는 광학 매체(CD, DVD)용, GFS는 서버 간 파일 공유 시스템으로 활용

File systems

- 주요 파일 시스템 외 특수 파일 시스템
  - 일반적인 파일 시스템 외에도 특수 목적의 파일 시스템들이 존재
  - 저장 매체, 네트워크, 트랜잭션, DB 등 용도에 따라 구조가 다름

유형	설명
디스크	일반적인 HDD, SSD 파일 시스템(E.g. NTFS, FAT32, ext4)
테이프 기반	고전 방식, 마그네틱 테이프에서 사용, 데이터는 순차 접근만 가능(랜덤 엑세스 불가) -> 매우 느림
네트워크 기반	원격 서버 파일 접근 방식(NFS, GFS 등), 파일이 로컬 디스크에 있는게 아니라 원격 서버에 저장되어있음
트랜잭션 기반	여러 파일을 한꺼번에 안전하게 변경할 필요가 있을 시 사용 (문제가 생기면 전부 취소, 성공하면 전부 저장)
데이터베이스 형	메타데이터(속성) 기반 파일 관리(E.g. IBM DB2 for i5), 파일 이름이 아니라 '작성자', '태그', '주제' 등으로 파일 탐색 가능
특수 목적형	운영체제 내부 정보 및 구조를 파일처럼 사용(E.g. 리눅스의 /proc, /dev)

File systems

- 파일 유형 및 확장자
  - **파일 시스템은 확장자(extension)을 통해 파일의 역할을 구분**
  - 포렌식 분석 시, 확장자를 분석하여 파일의 기능 및 위험성 판단 가능
    - E.g. .exe 확장자는 실행 파일 -> 악성코드가 포함되었을 가능성 있음

유형	예시 확장자	설명
실행 파일	.exe, .com, .bin, none	실행 가능한 프로그램
오브젝트 파일	.obj, .o	컴파일된 중간 결과물(링크 전)
소스 코드	.c, .py, .pas, .asm, .a	다양한 언어의 소스 코드
배치 파일	.bat, .doc	명령어 스크립트
텍스트 파일	.txt, .doc	일반 텍스트 문서
워드 문서	.wp, .tex, .rrf	워드 프로세서 포맷 문서
라이브러리	.lib, .a	코드 루틴 모음(프로그램에서 링크)
출력/보기용	.ps, .dvi, .gif, .pdf	인쇄 or 미리보기용 ASCII/바이너리 파일
압축 파일	.arc, .zip, .tar, .gz	여러 파일을 하나로 묶거나 압축한 파일

File systems

- 저장장치 계층 구조
  - 컴퓨터 저장장치는 속도, 용량 비용에 따라 계층적으로 나뉨
  - 상위 계층일수록 속도는 빠르지만 용량이 작음(그리고 비쌈)
  - 하위 계층일수록 속도는 느리지만 용량이 큼(그리고 저렴함)

계층	예시	용량
1차 저장소 (primary)	레지스터, 캐시, 메인메모리(RAM)	KB~GB
2차 저장소 (secondary)	SSD, HDD 등	GB~TB
3차 저장소 (Tertiary)	광 디스크, 마그네틱 테이프	TB 이상 가능

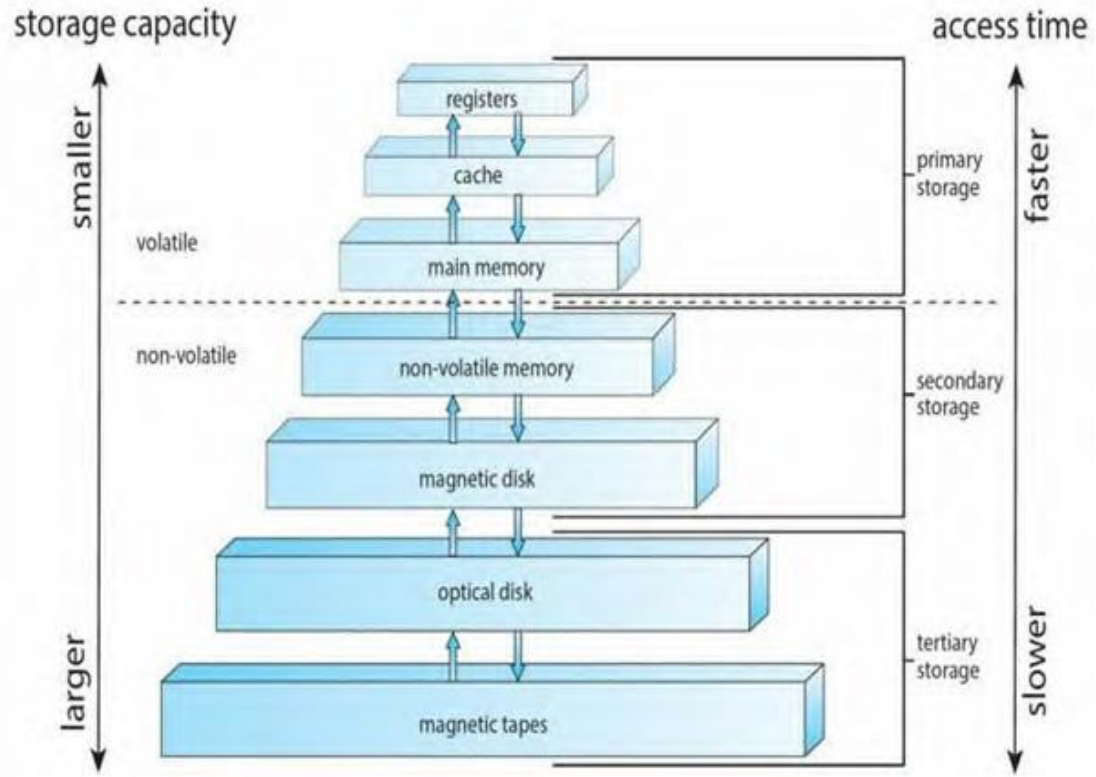
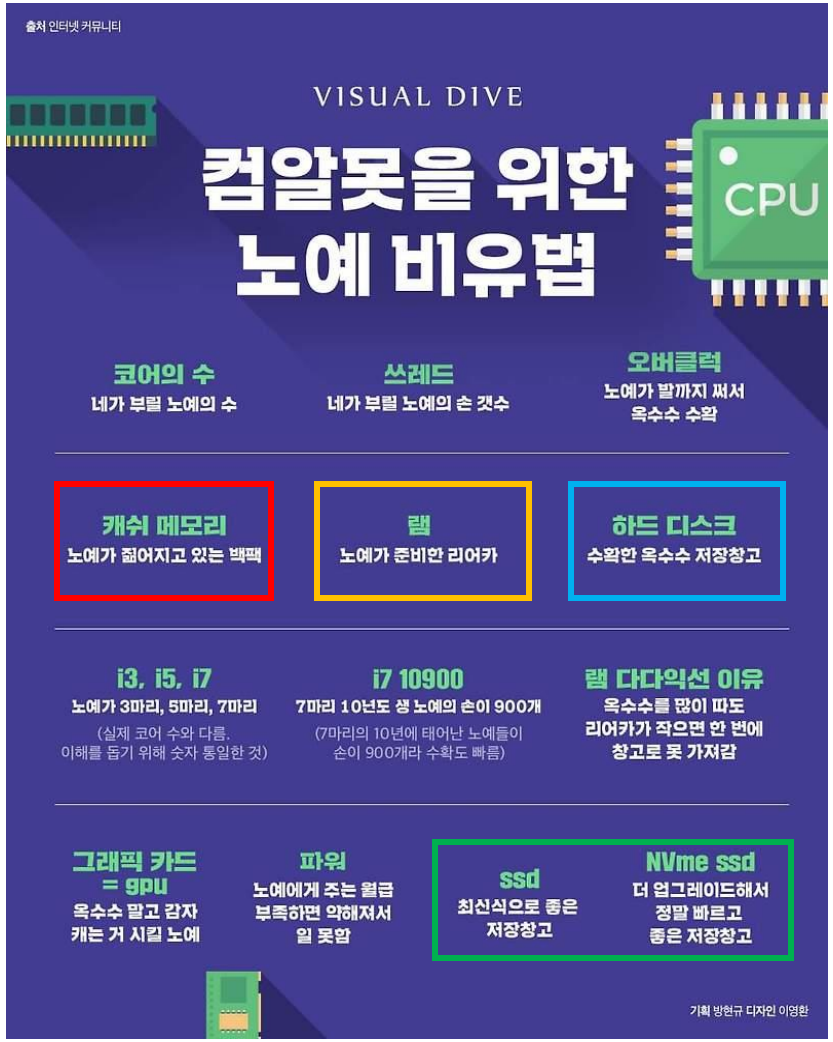


Figure 3.3: Memory hierarchy of storage

File systems

- 저장장치 계층 구조 이해를 돕기 위한, 유명한 옥수수 설명법



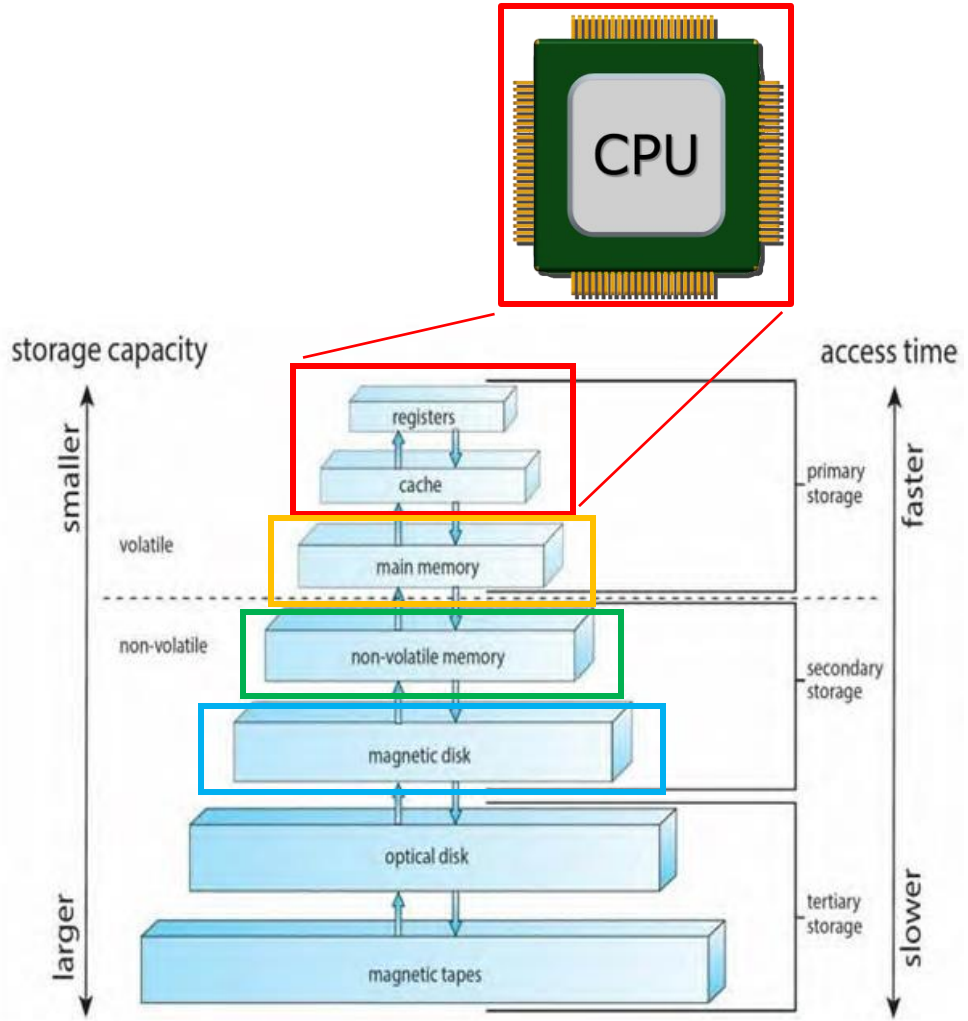


Figure 3.3: Memory hierarchy of storage

Hard disk

- 하드디스크(HDD)는 데이터를 자기 디스크에 저장하는 장치
- 내부는 다음과 같은 주요 부품으로 구성
- Platter(원판)
  - 디스크를 이루는 원형 판(데이터 저장)
- Head
  - 플래터의 자기 정보를 R/W
- VCM
  - 전자석 원리로 헤드를 움직임
- Spindle Motor
  - 플래터를 고속 회전 시킴
    - 읽기, 쓰기가 가능하게 함
- Data Track
  - 플래터 표면에 데이터를 저장할 때 사용하는 단위(논리적 개념)

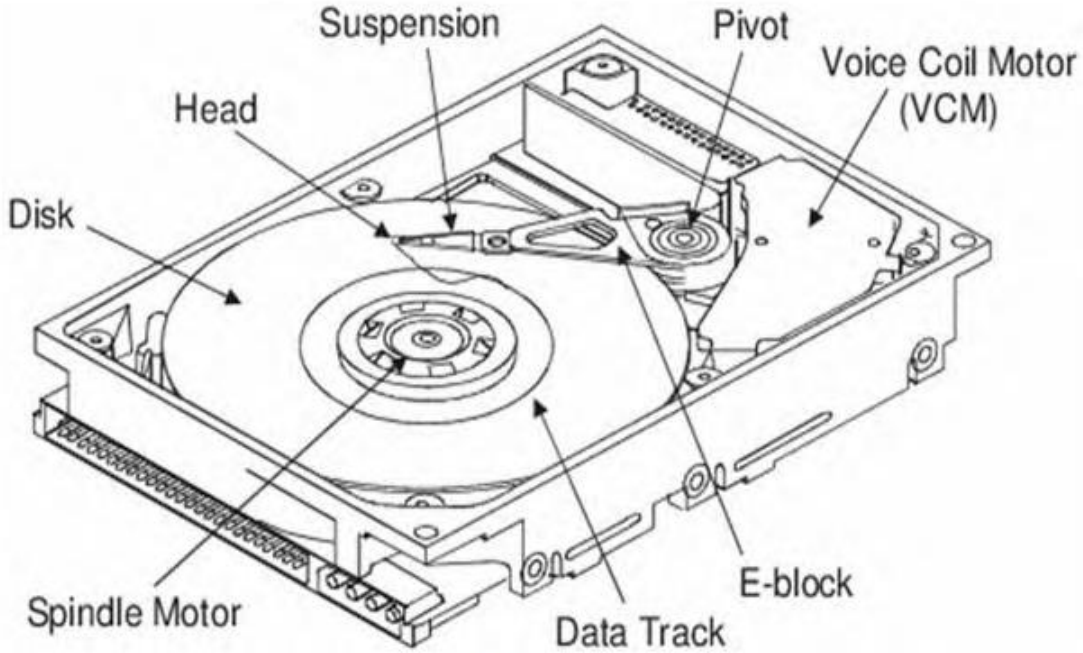


Figure 3.4: *Parts of hard disk*

Hard disk

- 하드디스크의 데이터 저장 구조
  - 플래터(원판)의 표면 위에 데이터를 저장
  - 저장 시 논리적인 단위로 영역을 나눠서 저장

유형	설명
트랙	디스크 표면 위의 원 모양 경로, 데이터를 원형으로 저장
섹터	트랙을 나눈 조각, 보통 512Byte 크기, **물리적 최소 저장 단위**
클러스터	여러 섹터를 묶은 단위, 파일 시스템이 실제로 사용하는 저장 단위, **논리적 최소 저장 단위**
실린더	여러 원판에서 같은 위치의 트랙들을 수직으로 연결한 개념

  - 클러스터는 항상 2의 배수 크기(1,2,4,8…섹터)
  - 하나의 파일이 한 클러스터에 다 안들어갈 경우, 여러 클러스터에 나누어 저장됨
    - **이 때 해당 클러스터들이 물리적으로 연속되지 않은 위치에 저장되면 단편화(Fragmentation) 발생**
    - 단편화 발생 시, 파일을 읽는 속도가 느려짐(헤드가 여러 위치를 이동해야 하기 때문)
    - 이러한 이유로, HDD에 Windows를 설치하던 시절에는 디스크 조각 모음 기능을 수행해줘야 했음

Hard disk forensics

- 하드디스크 포렌식이란?
  - **하드디스크 내부에서 삭제되거나, 파괴된 데이터를 복구하고 분석하는 과정**
  - 주로 복구한 데이터를 법적 증거로 활용
  - 포렌식 기법을 통해 암호화된 정보도 추출 가능
    - 단, 암호화된 정보를 복호화하는 건 별개의 문제
  - USB, 외장하드, CD/DVD, 플로피디스크, 모바일 기기 등이 대상
  - **원본에는 절대 포렌식 작업을 수행하지 않는다는 원칙 존재**
    - 실제 포렌식에서는 원본 데이터를 훼손하지 않기 위해 복제본을 분석함

Hard disk forensics

- 하드디스크 포렌식의 7단계 절차

단계	설명
1. 식별	범죄 현장 등에서 하드디스크, USB, CD 등 저장장치를 식별
2. 확보 및 획득	**장치를 확보하고 원본에 대한 해시값 생성, 복제본 생성 및 복제본의 해시값 생성** **-> 분석 중 데이터가 조작 및 손상되지 않았다는 무결성을 확보하는 핵심 단계**
3. 검증	생성한 복제본과 원본의 해시값 비교(동일성 검증)
4. 보존	원본은 안전한 장소에 보관, 복제본은 추가로 생성하여 백업
5. 분석	**복제본을 분석하여 삭제, 은폐, 암호화된 데이터 복구** **-> 가장 많은 시간과 전문성이 요구되는 단계**
6. 보고서 작성	수집 및 분석한 내용을 요약하여 정리
7. 문서화	법적 효력을 갖는 공식 보고서 작성

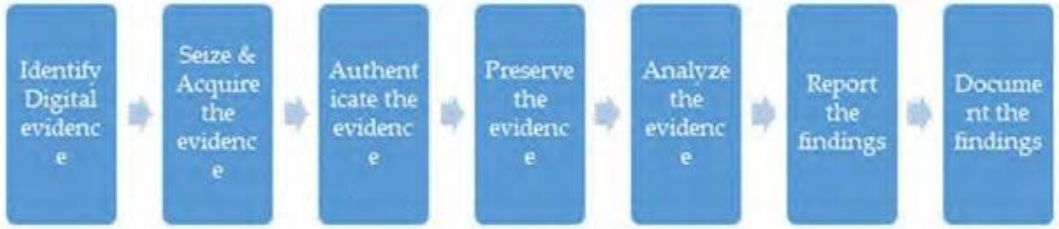


Figure 3.5: Steps involved in hard disk forensics

Analyzing the registry files

- Windows 레지스트리 분석
  - **레지스트리: Windows 시스템의 핵심 설정 정보가 저장되는 DB**
    - OS의 설정, 사용자 활동, 설치 프로그램, 로그인 정보 등을 저장
    - **시스템의 일기장이라고 볼 수 있음**
  - 사용자의 활동 흔적이 레지스트리에 남을 확률이 높음
    - 대부분 사용자는 레지스트리를 직접적으로 조작하지 않기 때문
  - 주요 정보 예시
    - 최근 사용된 파일 목록(MRU 리스트)
    - 자동 실행 프로그램(Run 키)
    - 설치된 소프트웨어 및 경로
    - 마지막 로그인 한 사용자 및 시간
  - 단점: 구조가 매우 복잡하여 수작업으로 분석하기 어렵고, 시간이 오래 걸림
  - RegAlyzer 도구 활용 가능
    - 기존 regedit의 기능적 한계를 극복한 레지스트리 분석 도구
    - 패턴 검색(정규 표현식 사용), 히스토리 기능, 탭 사용, 북마크 기능 등 사용 가능

Analyzing the registry files

- Windows 시스템 파일 분석(NTFS)
  - **NTFS는 파일 삭제 시, 파일 본문은 유지된 채 인덱스 정보만 제거됨**
    - 이를 통해 포렌식 도구를 사용하여 파일의 내용, 삭제 시간, 사용자 정보 등을 복구 가능
    - E.g. 로그인 사용자와 삭제 시간 비교를 통한 행위자 추정
  - Alternate Data Streams(ADS)
    - NTFS는 파일 하나에 여러 데이터 스트림을 저장할 수 있는 ADS 기능 제공
    - Mac과의 파일 호환을 목적으로 생긴 기능
    - But, 텍스트 및 실행 파일의 데이터를 은닉할 수 있어 악용 가능
    - E.g. document.txt:hidden.exe -> 탐색기에는 보이지 않지만, 실제 exe 파일 존재
  - 자동화 도구를 사용하여 NTFS 분석 가능
    - 삭제된 파일의 시간 정보 및 사용자 추정
    - ADS 스트림 자동 탐지 및 위치 추적
  - **단, 고의 삭제(덮어쓰기, 파괴 등)가 이루어진 경우는 복구가 불가능할 수 있음**

Q & A