

15장 파일 시스템

파일

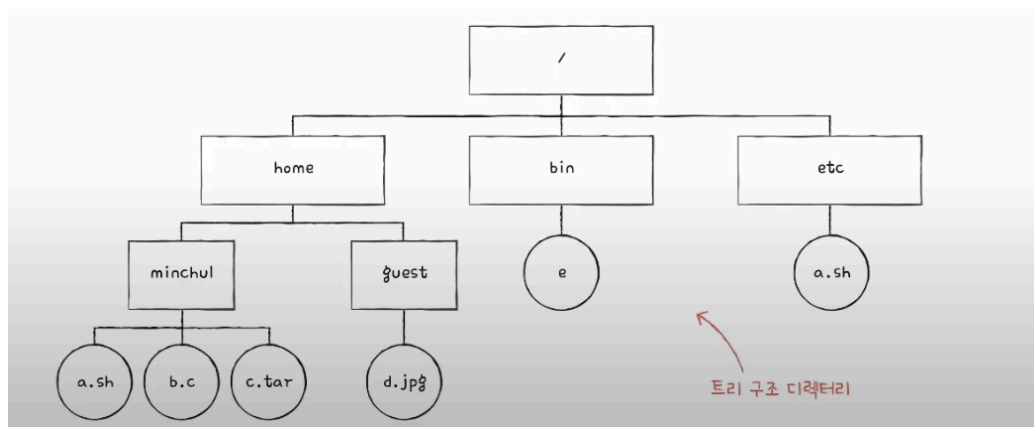
- 하드 디스크나 SSD와 같은 보조기억장치에 저장된 관련 정보의 집합
 - 메타데이터

속성 이름	의미
유형	운영체제가 인지하는 파일의 종류를 나타낸다.
크기	파일의 현재 크기와 허용 가능한 최대 크기를 나타낸다.
보호	어떤 사용자가 해당 파일을 읽고, 쓰고, 실행할 수 있는지를 나타낸다.
생성 날짜	파일이 생성된 날짜를 나타낸다.
마지막 접근 날짜	파일에 마지막으로 접근한 날짜를 나타낸다.
마지막 수정 날짜	파일이 마지막으로 수정된 날짜를 나타낸다.
생성자	파일을 생성한 사용자를 나타낸다.
소유자	파일을 소유한 사용자를 나타낸다.
위치	파일의 보조기억장치상의 현재 위치를 나타낸다.

→ 확장자

파일 유형	대표적인 확장자
실행 파일	없는 경우, exe, com, bin
목적 파일	obj, o
소스 코드 파일	c, cpp, cc, java, asm, py
워드 프로세서 파일	xml, rtf, doc, docx
라이브러리 파일	lib, a, so, dll
멀티미디어 파일	mpeg, mov, mp3, mp4, avi
백업/보관 파일	rar, zip, tar

디렉토리



- 여러 계층으로 파일 및 폴더를 관리하는 **트리 구조 디렉토리**
- **절대 경로** : 루트 디렉토리에서 자기 자신까지 이르는 고유한 경로
- **상대 경로** : 현재 디렉토리부터 시작하는 경로
- **디렉토리 엔트리**
 - 디렉토리 내부에 해당 디렉토리에 담겨 있는 대상과 관련된 정보
 - 즉, 보조기억장치에서 저장된 위치를 알 수 있음

파일 시스템

파티셔닝과 포매팅

- **파티셔닝** : 저장 장치의 논리적인 영역을 구획하는 작업
 - 파티션마다 다른 파일 시스템을 설정할 수도 있음
- **포매팅** : 파일 시스템을 설정하여 어떤 방식으로 파일을 저장하고 관리할 것인지를 결정하고, 새로운 데이터를 쓸 준비를 하는 작업을 의미
- 파티셔닝과 포매팅이 완료되면 파일 시스템을 이용할 수 있음

파일 할당 방법

- 운영체제는 파일과 디렉터리를 블록 단위로 읽고 씀
- 하드 디스크의 가장 작은 저장 단위는 **섹터**이지만, 운영체제는 하나 이상의 섹터를 블록이라는 단위로 묶은 뒤 **블록** 단위로 파일과 디렉터리를 관리함.
- **연속 할당**
 - 파일이 저장 장치 내에서 연속적인 공간을 차지하도록 블록을 할당하는 방법



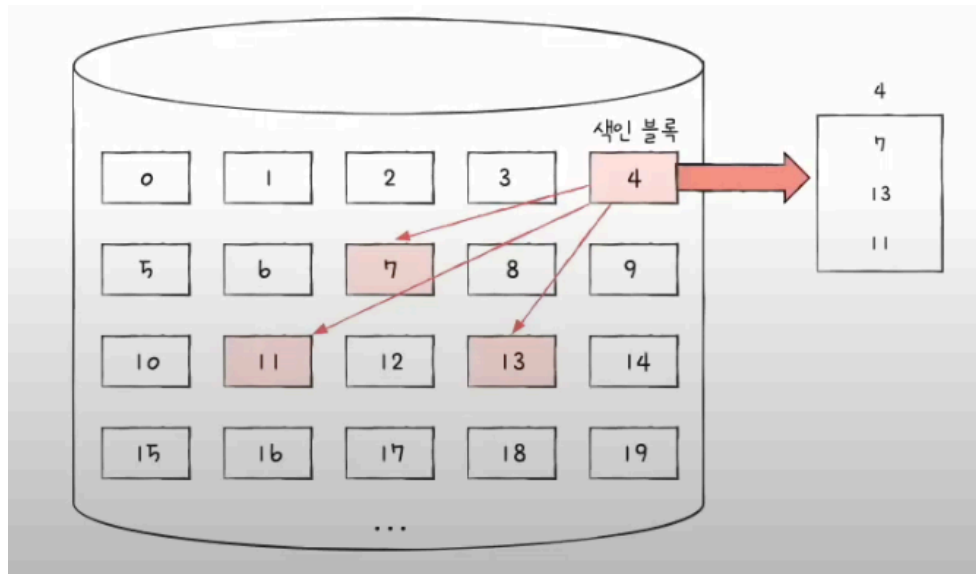
- 디렉토리 엔트리 : 파일 이름 & 첫 번째 블록 주소 & 블록 단위 길의 명시
- 장점 : 연속적으로 저장하는 방식이기에 구현이 단순
- 단점 : 외부 단편화를 야기함
- **연결 할당**
 - 각 블록 일부에 다음 블록의 주소를 저장하여 각 블록이 다음 블록을 가리키는 형태로 할당하는 방식
 - 즉, 파일을 이루는 데이터를 연결 리스트로 관리

- 단점

- 반드시 첫 번째 블록부터 하나씩 차례대로 읽어야 함
- 하드웨어 고장이나 오류 발생 시 해당 블록 이후 블록은 접근할 수 없음

- 색인 할당

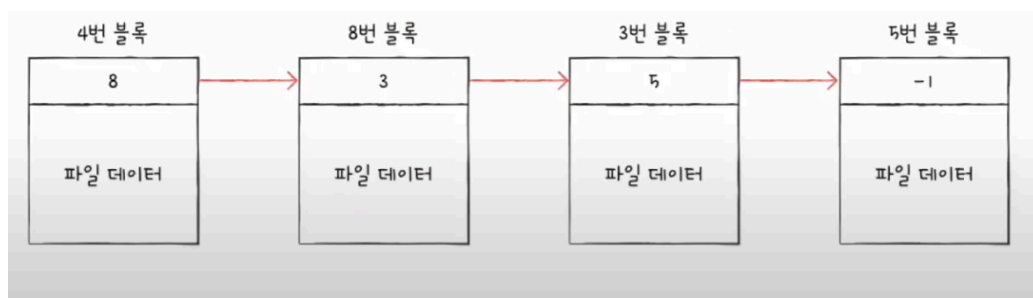
- 파일의 모든 블록 주소를 색인 블록이라는 하나의 블록에 모아 관리하는 방식



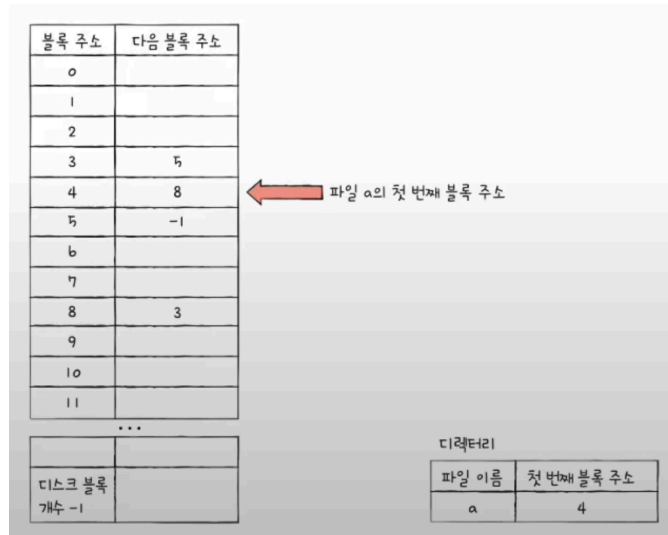
- 파일 내 임의의 위치에 접근하기 용이

FAT 파일 시스템

- 연결 할당 기반 파일 시스템
- 연결 할당의 단점을 보완

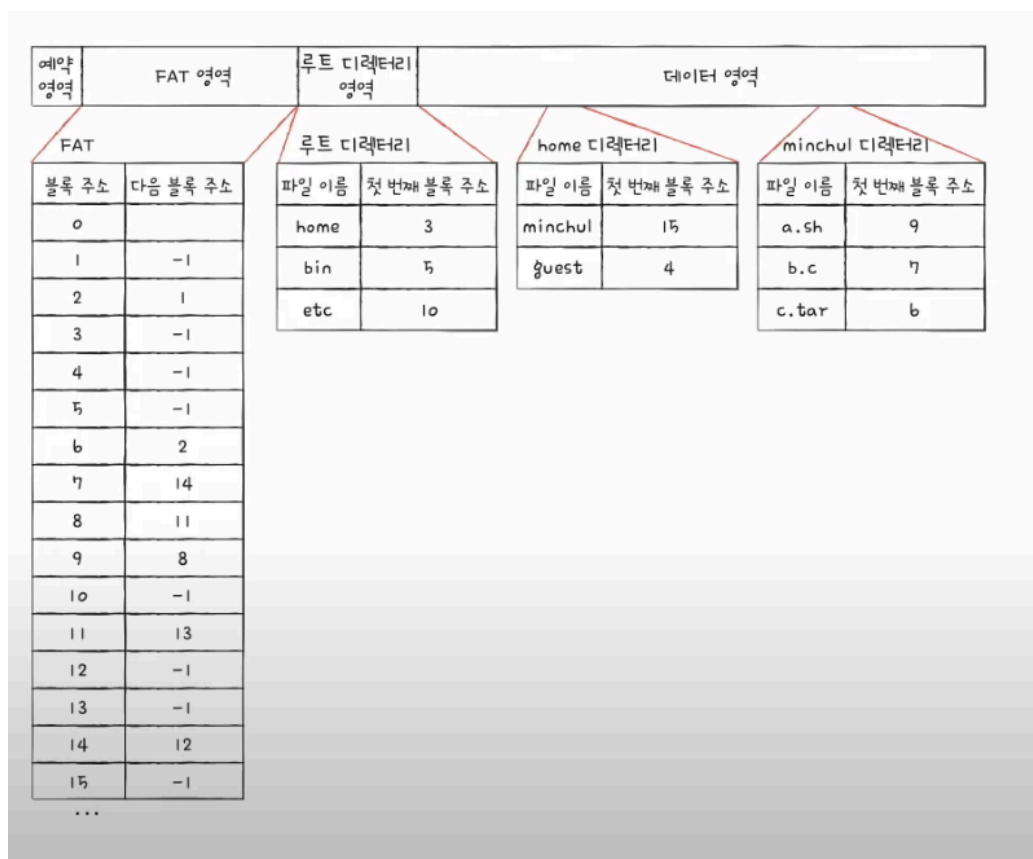


- 각 블록에 포함된 다음 블록 주소를 한데 모아 테이블(FAT: File Allocation Table)로 관리



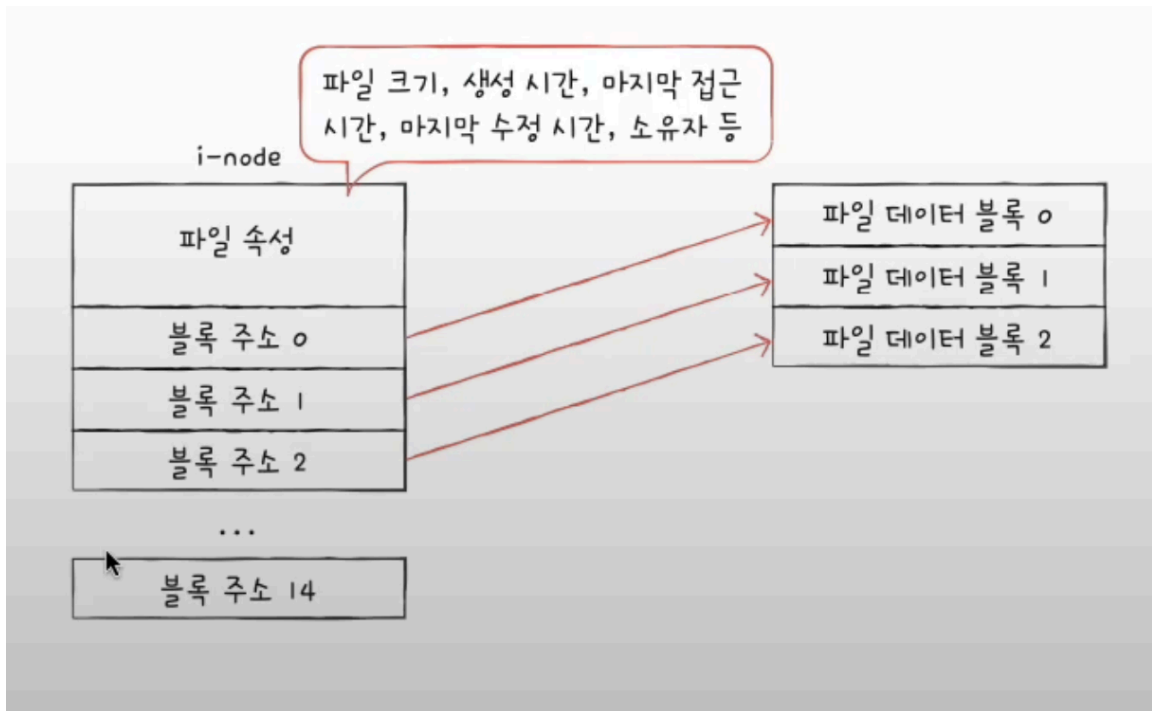
- FAT가 메모리에 적재된 채 실행되면 임의 접근의 성능이 개선됨
- 디렉토리 엔트리

파일 이름	확장자	속성	예약 영역	생성 시간	마지막 접근 시간	마지막 수정 시간	시작 블록	파일 크기
-------	-----	----	-------	-------	-----------	-----------	-------	-------

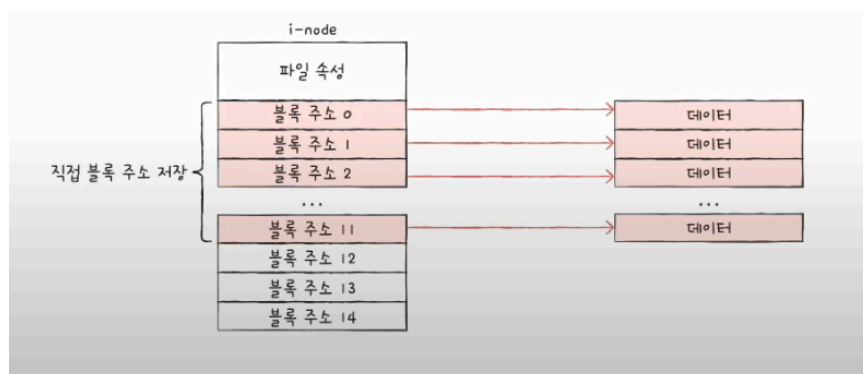


유닉스 파일 시스템

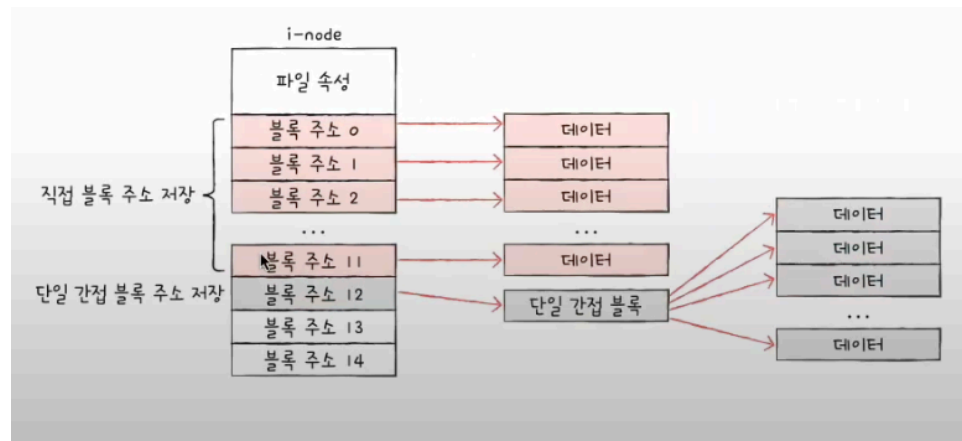
- 색인 할당 기반 파일 시스템
- 색인 블록을 i-node라고 부름
 - 파일의 속성 정보와 15개의 블록 주소 저장 가능



- 15개보다 큰 파일을 저장해야 하는 경우
 1. 블록 주소 중 12개에는 직접 블록 주소 저장
 - 직접 블록 : 파일 데이터가 저장된 블록

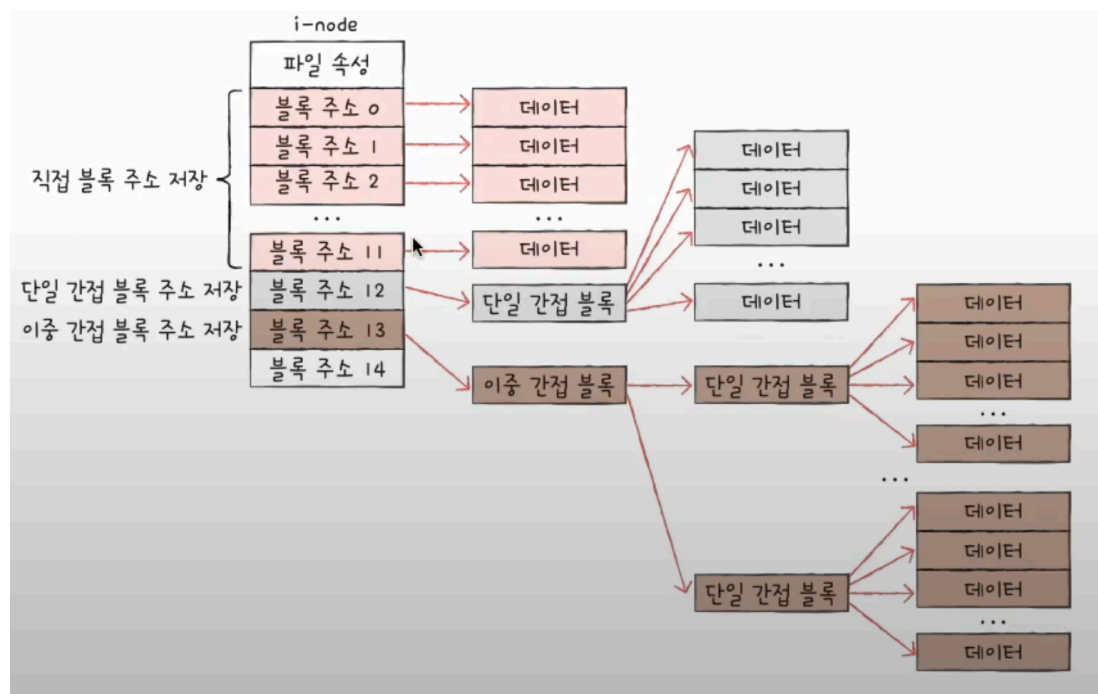


2. 1번으로 충분하지 않다면 13번째 주소에 단일 간접 블록 주소 저장
 - 단일 간접 블록 : 파일 데이터를 저장한 블록 주소가 저장된 블록



3. 2번으로 충분하지 않다면 14번째 주소에 이중 간접 블록 주소 저장

- 이중 간접 블록 : 단일 간접 블록들의 주소를 저장하는 블록



4. 3번으로 충분하지 않다면 열다섯 번째 주소에 삼중 간접 블록 주소를 저장

- 삼중 간접 블록 : 이중 간접 블록 주소가 저장된 블록
- 디렉터리 엔트리



