파일시스템 포렌식

2024. 1

이창하

목차

파일시스템 포렌식

학습목표

- 파일시스템 분석관련 기본 용어
- 파일시스템의 기본 구조 및 메타 데이터의 종류와 역할
- 아이노드와 디렉토리 엔트리의 차이
- 파일시스템의 복원 과정
- 주요 모바일 파일시스템의 기본 구조
- 비할당 영역을 식별하는 정확한 과정

왜 파일시스템을 공부하는가?

왜 파일시스템을 공부하는가?

사례

- exFAT의 경우 복사된 파일과 레코딩한 파일의 저장구조가 다름

왜 파일시스템을 공부하는가?

사례

- exFAT의 경우 복사된 파일과 레코딩한 파일의 저장구조가 다름

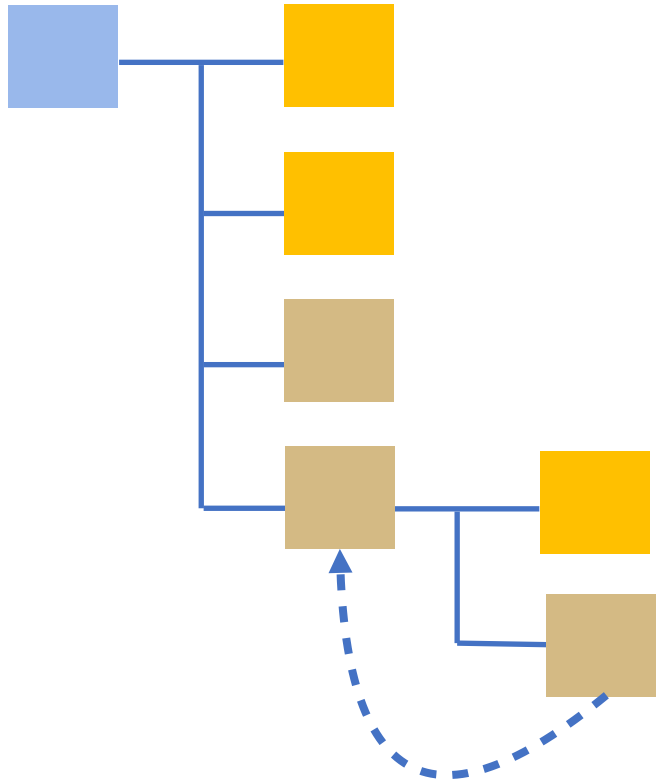
"파일 저장 패턴 자체가 포렌식 아티팩트(artifact)"

왜 파일시스템을 공부하는가?

사례

- JTAG, Chip-Off시 BitFlip 현상 발생

  F/S 메타 데이타 손상 가능



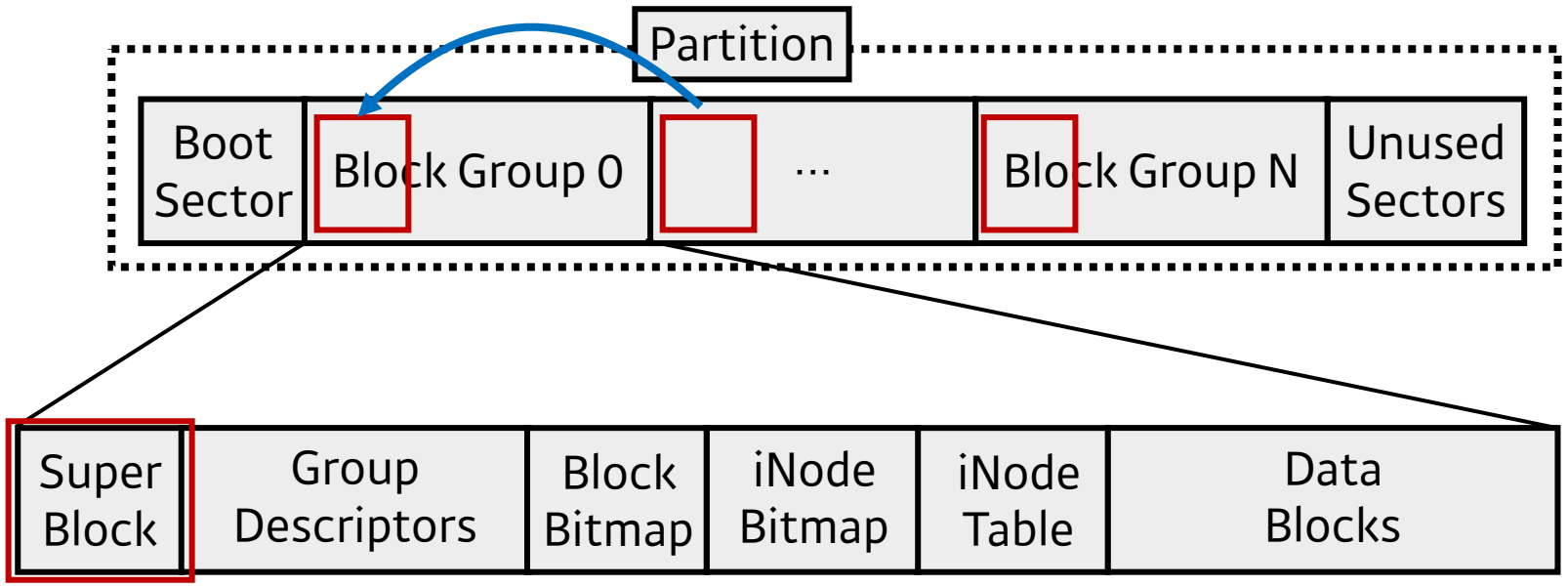
Infinite loop!

왜 파일시스템을 공부하는가?

사례

- JTAG, Chip-Off시 BitFlip 현상 발생

  F/S 메타 데이타 손상 가능



왜 파일시스템을 공부하는가?

사례

- 256G iPhone의 FFS (Full File System) 획득 결과 이미지가 4TB

주요 용어 및 개념

주요 용어 및 개념

파일시스템

- 정의 - The **structure and logic rules** used to manage the groups of information and their names.

주요 용어 및 개념

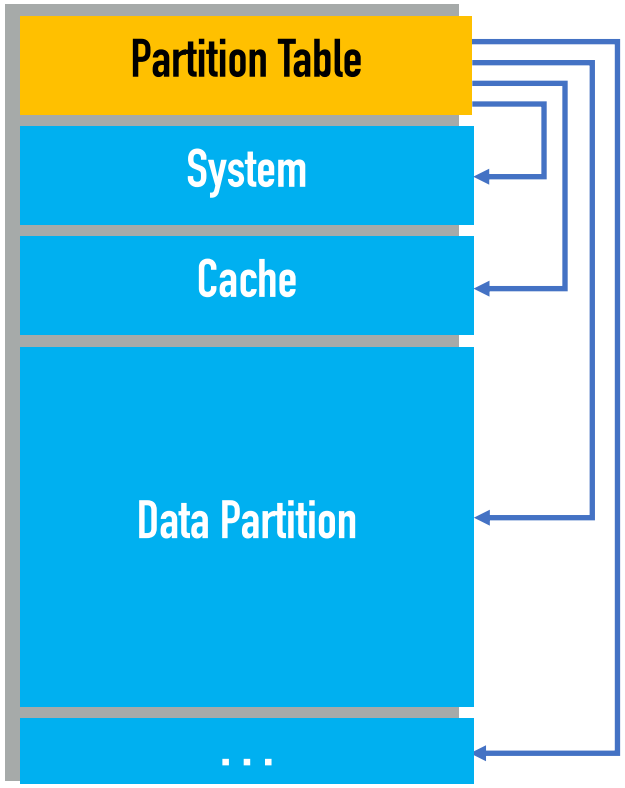
파일시스템

- 정의 - The **structure and logic rules** used to manage the groups of information and their names.

- 종류
  - ext3/4, hfs+, ntfs
  - fat 12/16/32, exFAT
  - efs2, tfs4, yaffs, jffs2
  - f2fs, apfs, vdfs

주요 용어 및 개념

파일시스템

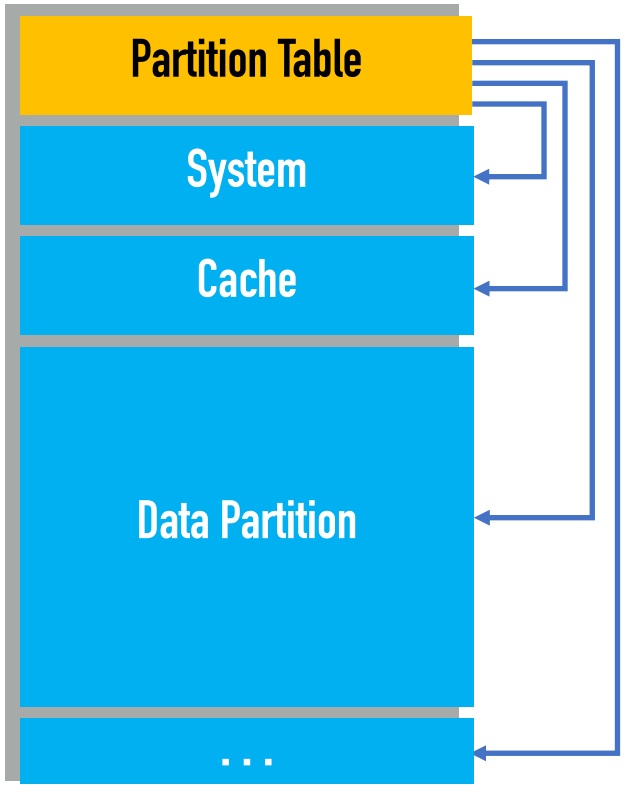


- 파티션 테이블

  파티션의 크기, 위치 및 이름을 기록하고 있는 테이블

주요 용어 및 개념

파일시스템



- 파티션 테이블

  파티션의 크기, 위치 및 이름을 기록하고 있는 테이블

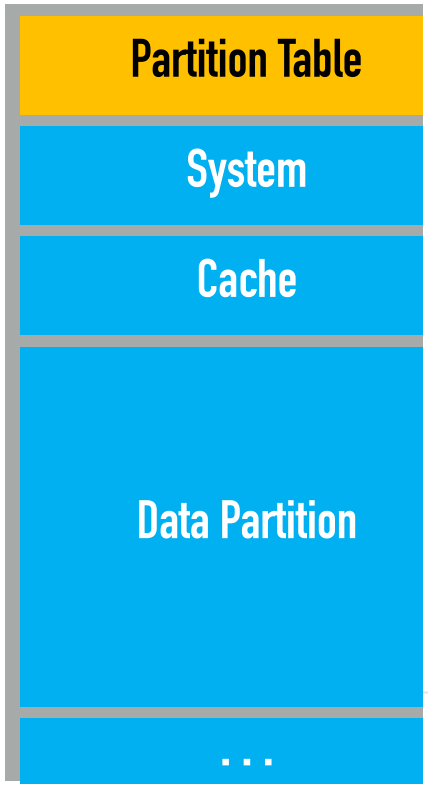
- 종류

  MBR(Master Boot Record),

  GPT(GUID partition table), EPK 등

주요 용어

파일시스템



는 테이블

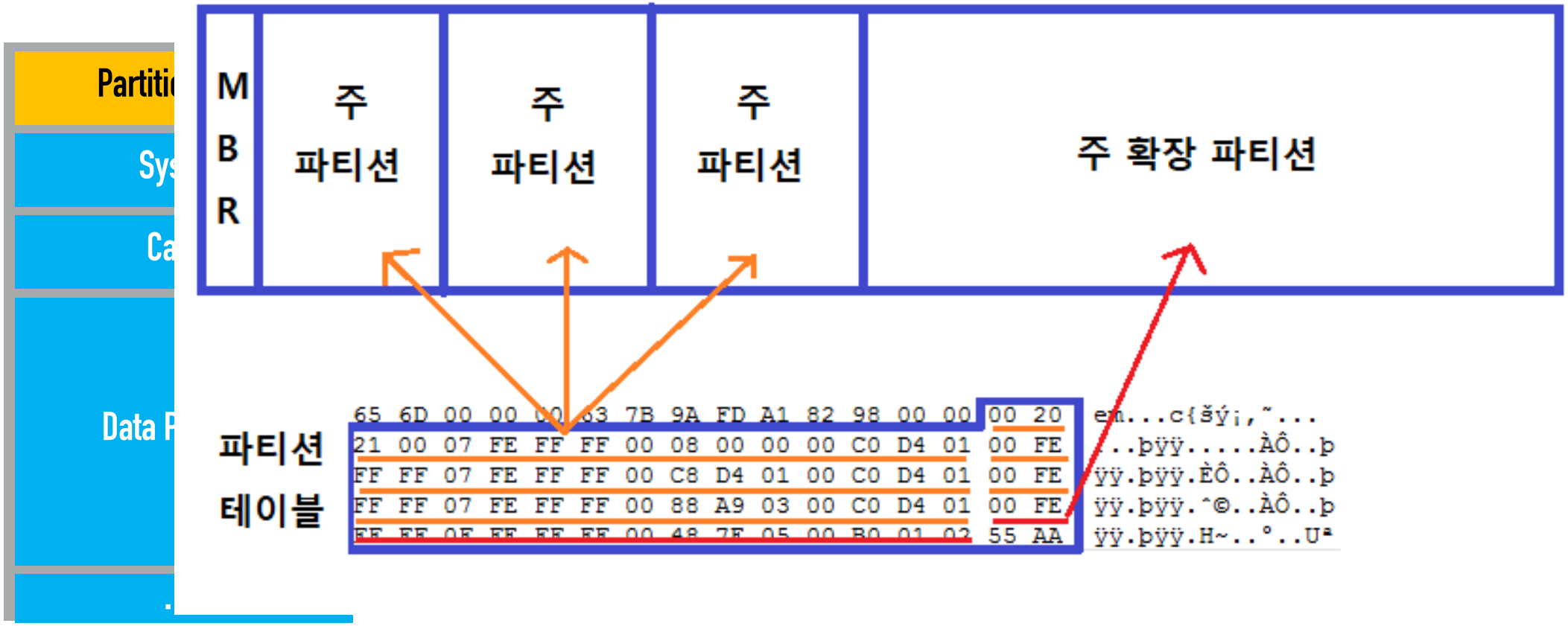
FAT32.001

```
Offset(h)  00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
00000000   33 C0 8E D0 BC 00 7C 8E C0 8E D8 BE 00 7C BF 00  3ÀŽÐ¼.|ŽÀŽØ¾.|¿.
00000010   06 B9 00 02 FC F3 A4 50 68 1C 06 CB FB B9 04 00  .¹..üó¤Ph..Ëû¹..
00000020   BD BE 07 80 7E 00 00 7C 0B 0F 85 0E 01 83 C5 10  ½¾.€~..|.....ƒÅ.
00000030   E2 F1 CD 18 88 56 00 55 C6 46 11 05 C6 46 10 00  âñÍ.ˆV.UÆF..ÆF..
00000040   B4 41 BB AA 55 CD 13 5D 72 0F 81 FB 55 AA 75 09  ´A»ªUÍ.]r..ûUªu.
00000050   F7 C1 01 00 74 03 FE 46 10 66 60 80 7E 10 00 74  ÷Á..t.þF.f`€~..t
00000060   26 66 68 00 00 00 00 66 FF 76 08 68 00 00 68 00  &fh....fÿv.h..h.
00000070   7C 68 01 00 68 10 00 B4 42 8A 56 00 8B F4 CD 13  |h..h..´BŠV.‹ôÍ.
00000080   9F 83 C4 10 9E EB 14 B8 01 02 BB 00 7C 8A 56 00  ŸƒÄ.žë.¸..».|ŠV.
00000090   8A 76 01 8A 4E 02 8A 6E 03 CD 13 66 61 73 1C FE  Šv.ŠN.Šn.Í.fas.þ
000000A0   4E 11 75 0C 80 7E 00 80 0F 84 8A 00 B2 80 EB 84  N.u.€~.€.„Š.²€ë„
000000B0   55 32 E4 8A 56 00 CD 13 5D EB 9E 81 3E FE 7D 55  U2äŠV.Í.]ëž.>þ}U
000000C0   AA 75 6E FF 76 00 E8 8D 00 75 17 FA B0 D1 E6 64  ªunÿv.è..u.ú°Ñæd
000000D0   E8 83 00 B0 DF E6 60 E8 7C 00 B0 FF E6 64 E8 75  èƒ.°ßæ`è|.°ÿædèu
000000E0   00 FB B8 00 BB CD 1A 66 23 C0 75 3B 66 81 FB 54  .û¸.»Í.f#Àu;f.ûT
000000F0   43 50 41 75 32 81 F9 02 01 72 2C 66 68 07 BB 00  CPAu2.ù..r,fh.».
00000100   00 66 68 00 02 00 00 66 68 08 00 00 00 66 53 66  .fh....fh....fSf
00000110   53 66 55 66 68 00 00 00 00 66 68 00 7C 00 00 66  SfUfh....fh.|..f
00000120   61 68 00 00 07 CD 1A 5A 32 F6 EA 00 7C 00 00 CD  ah...Í.Z2öê.|..Í
00000130   18 A0 B7 07 EB 08 A0 B6 07 EB 03 A0 B5 07 32 E4  . ·.ë. ¶.ë. µ.2ä
00000140   05 00 07 8B F0 AC 3C 00 74 09 BB 07 00 B4 0E CD  ...‹ð¬<.t.»..´.Í
00000150   10 EB F2 F4 EB FD 2B C9 E4 64 EB 00 24 02 E0 F8  .ëòôëý+Éädë.$.àø
00000160   24 02 C3 49 6E 76 61 6C 69 64 20 70 61 72 74 69  $.ÃInvalid parti
00000170   74 69 6F 6E 20 74 61 62 6C 65 00 45 72 72 6F 72  tion table.Error
00000180   20 6C 6F 61 64 69 6E 67 20 6F 70 65 72 61 74 69   loading operati
00000190   6E 67 20 73 79 73 74 65 6D 00 4D 69 73 73 69 6E  ng system.Missin
000001A0   67 20 6F 70 65 72 61 74 69 6E 67 20 73 79 73 74  g operating syst
000001B0   65 6D 00 00 00 63 7B 9A 38 57 61 78 00 00 00 02  em...c{š8Wax....
000001C0   03 00 0C 61 1B 06 80 00 00 00 00 90 01 00 00 00  ...a..€.........
000001D0   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000001E0   00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000001F0   00 00 00 00 00 00 00 00 00 00 00 00 00 00 55 AA  ..............U ª
```

CHS Address
Partition Type
Ending CHS Address
Staring LBA Address
Size in Sector
Signature
Boot Flag

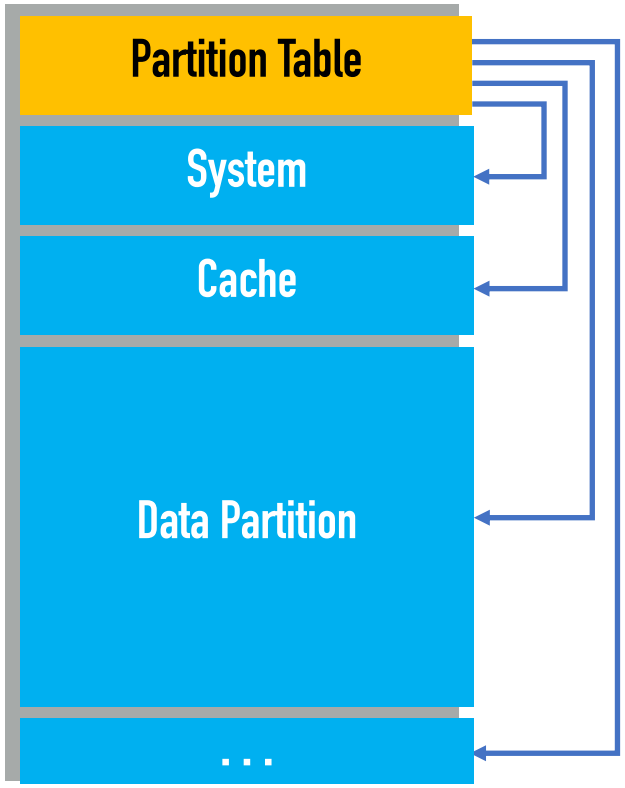
주요 용어 및 개념

파일시스템



주요 용어 및 개념

파일시스템

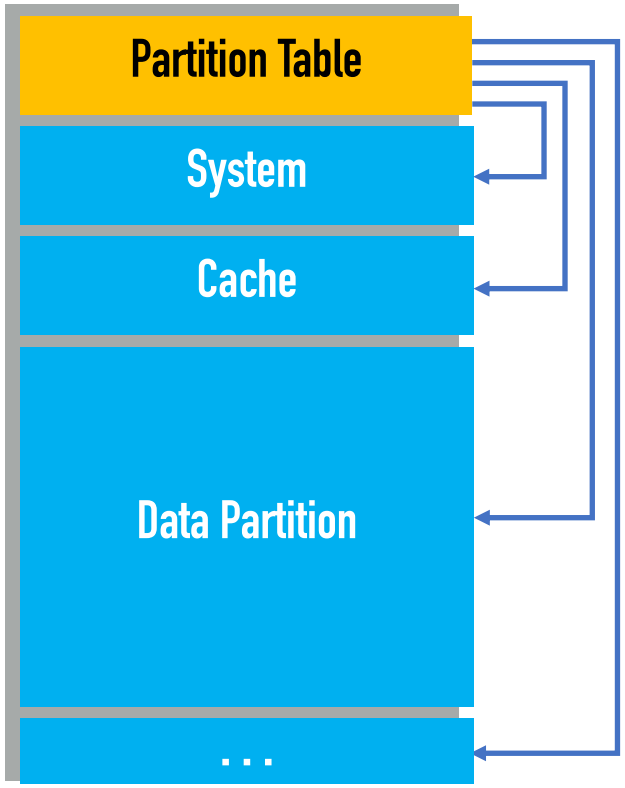


- 파티션

  물리적으로 연속된 섹터의 모임

주요 용어 및 개념

파일시스템



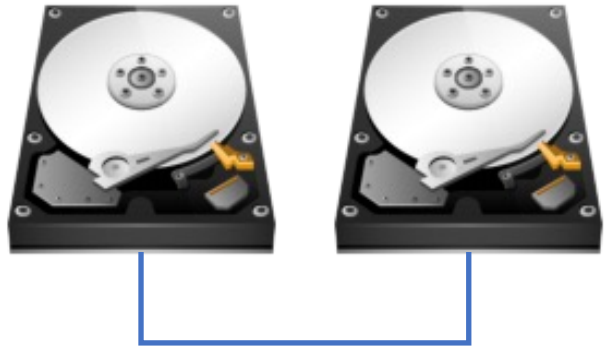
- 파티션

  물리적으로 연속된 섹터의 모임

- 볼륨

  논리적인 섹터의 모임

  파티션 ⊆ 볼륨



볼륨

주요 용어 및 개념

파티션 / 볼륨

- 파티션의 이점
  - 시스템 파티션과 구분하여 데이터 저장, **백업**
  - 다양한 운영체제 설치
  - 파일 탐색 시간 향상 (헤드 움직임 감소)

주요 용어 및 개념

파티션 / 볼륨

- 파티션의 이점
  - 시스템 파티션과 구분하여 데이터 저장, 백업
  - 다양한 운영체제 설치
  - 파일 탐색 시간 향상 (헤드 움직임 감소)
- 동적 볼륨의 이점
  - 동적으로 사용량 증가, 저장 매체 추가/제거 → 오라클
  - 대용량 저장소

주요 용어 및 개념

섹터, 클러스터, 블록, 페이지

- 섹터

  하드 디스크에서 원자성을 지원하는 I/O 단위

  0x200?

- 클러스터

  마이크로소프트에서 만든 파일시스템의 I/O 단위

- 블록 / 페이지

주요 용어 및 개념

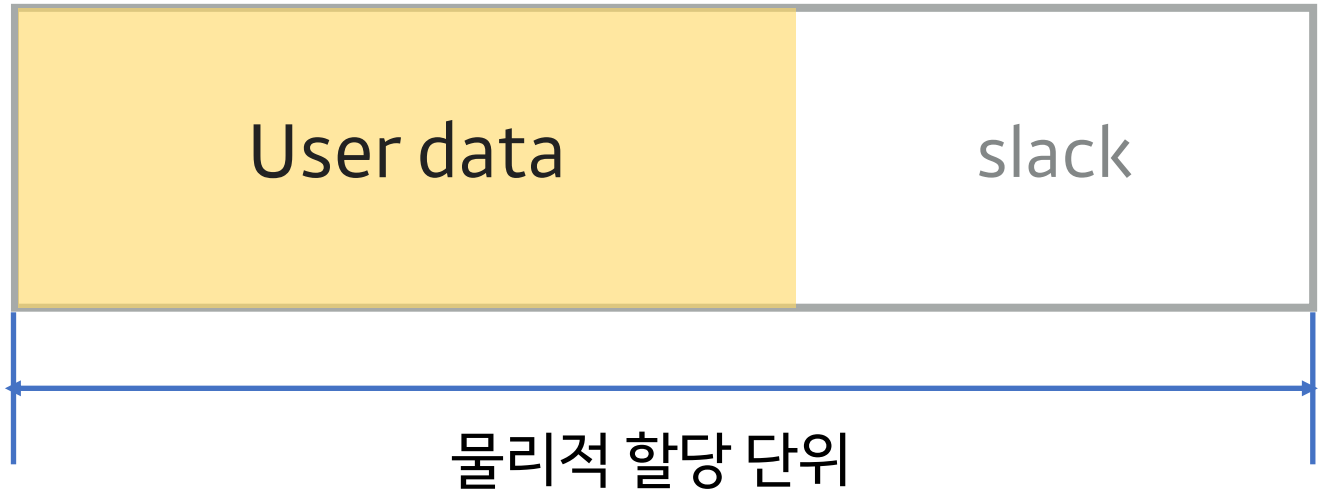
슬랙 (Slack)

- 저장매체의 물리적 구조와 논리적 구조의 차이로 낭비되는 공간

주요 용어 및 개념

슬랙 (Slack)

- 저장매체의 물리적 구조와 논리적 구조의 차이로 낭비되는 공간



주요 용어 및 개념

슬랙 (Slack)

- 저장매체의 물리적 구조와 논리적 구조의 차이로 낭비되는 공간
  - 삭제 데이터 복구
  - 안티 포렌식
  - 안티 안티 포렌식

주요 용어 및 개념

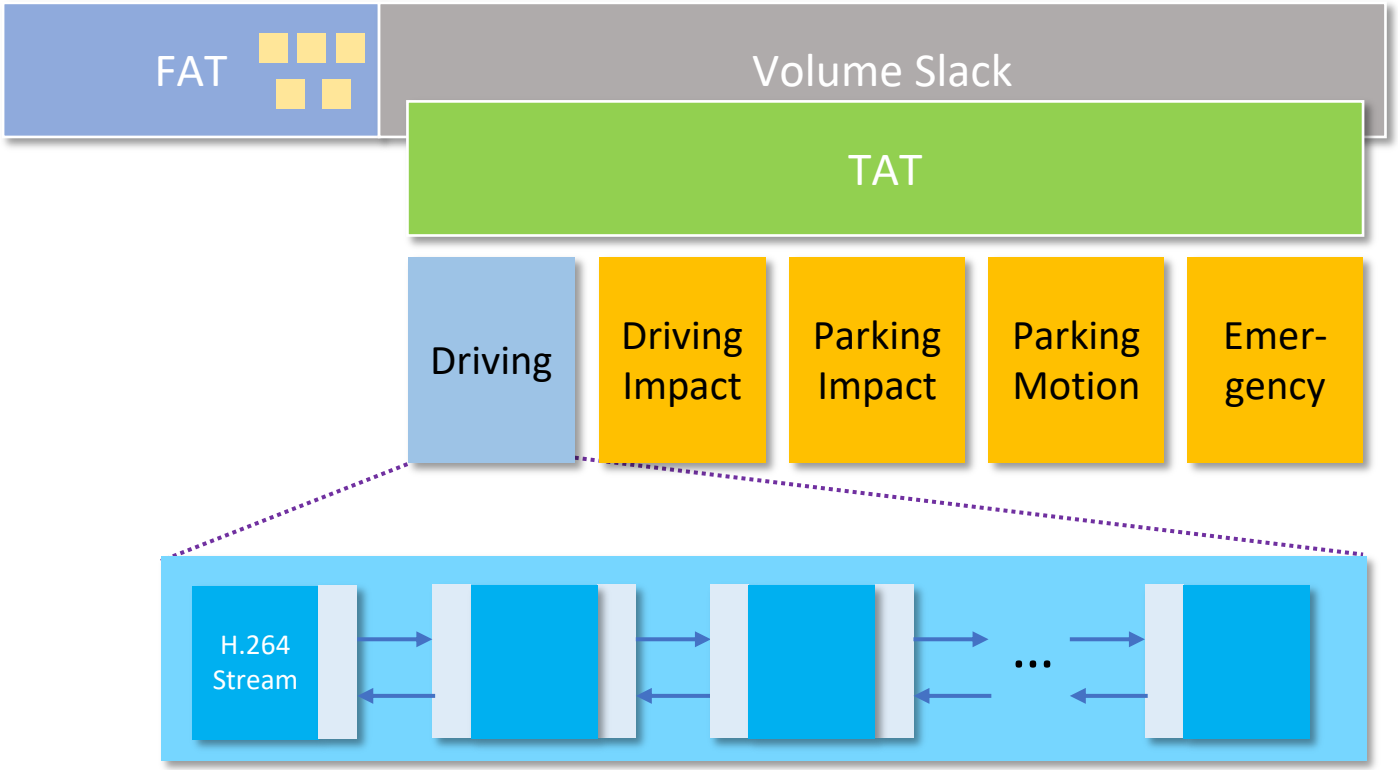
슬랙 (Slack)

- 저장매체의 물리적 구조와 논리적 구조의 차이로 낭비되는 공간

주요 용어 및 개념

슬랙 (Slack) - 볼륨 슬랙 예제

- 저장매체의 물리적 구조와 논리적 구조의 차이로 낭비되는 공간



주요 용어 및 개념

슬랙 (Slack) - 볼륨 슬랙 예제

- 저장매체의 물리적 구조와 논리적 구조의 차이로 낭비되는 공간

주요 용어 및 개념

슬랙 (Slack) - 볼륨 슬랙 예제

- 저장매체의 물리적 구조와 논리적 구조의 차이로 낭비되는 공간

주요 용어 및 개념

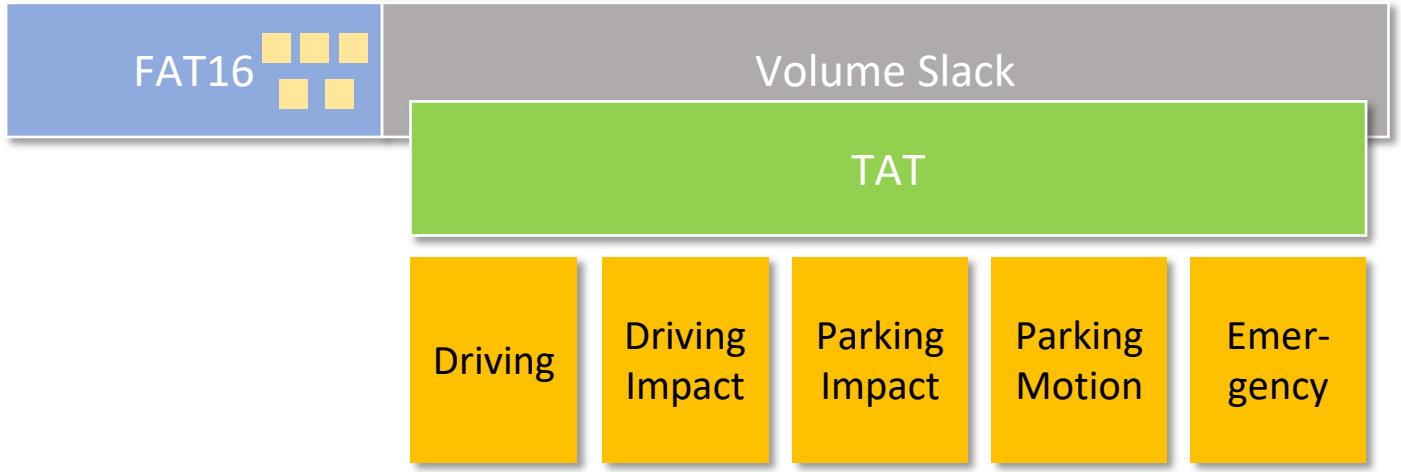
슬랙 (Slack) - 볼륨 슬랙 예제

- 저장매체의 물리적 구조와 논리적 구조의 차이로 낭비되는 공간

주요 용어 및 개념

슬랙 (Slack) - 볼륨 슬랙 예제

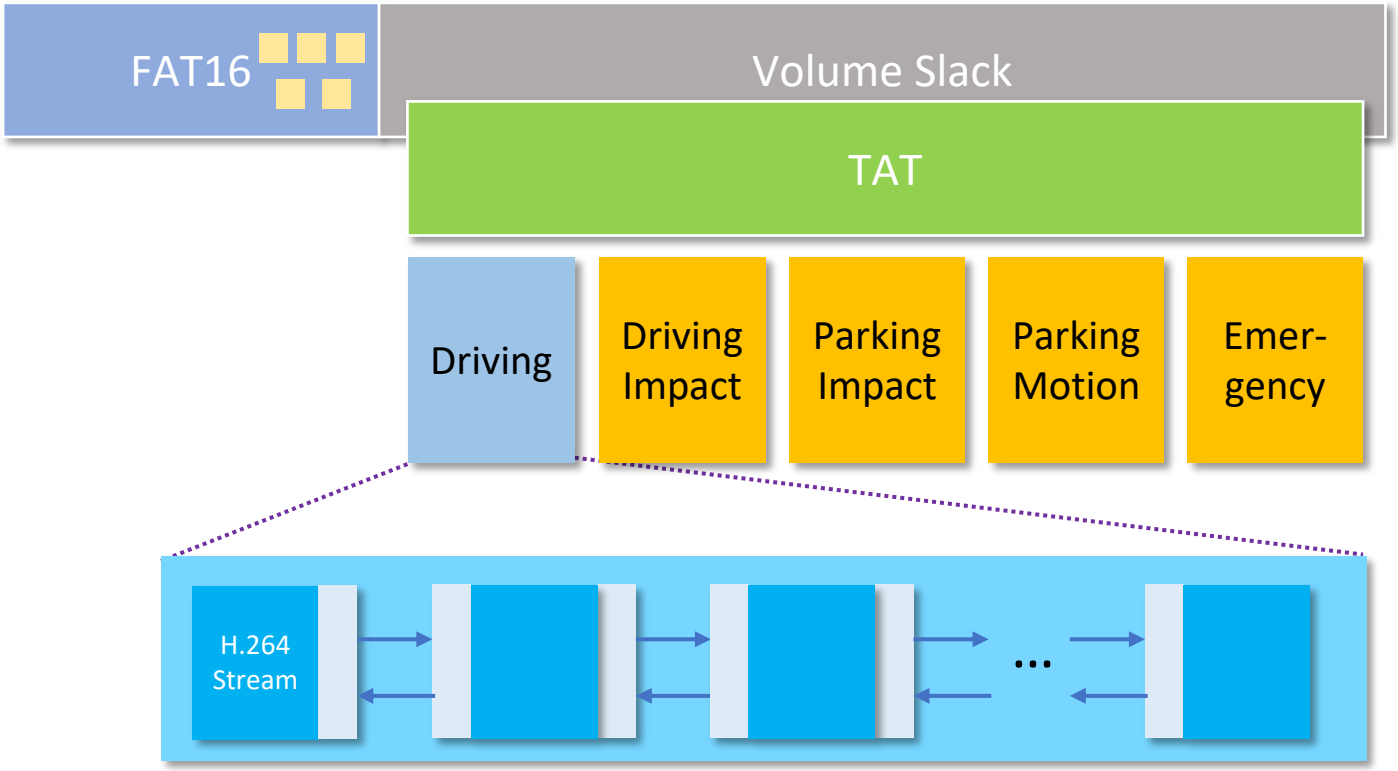
- 저장매체의 물리적 구조와 논리적 구조의 차이로 낭비되는 공간



주요 용어 및 개념

슬랙 (Slack) - 볼륨 슬랙 예제

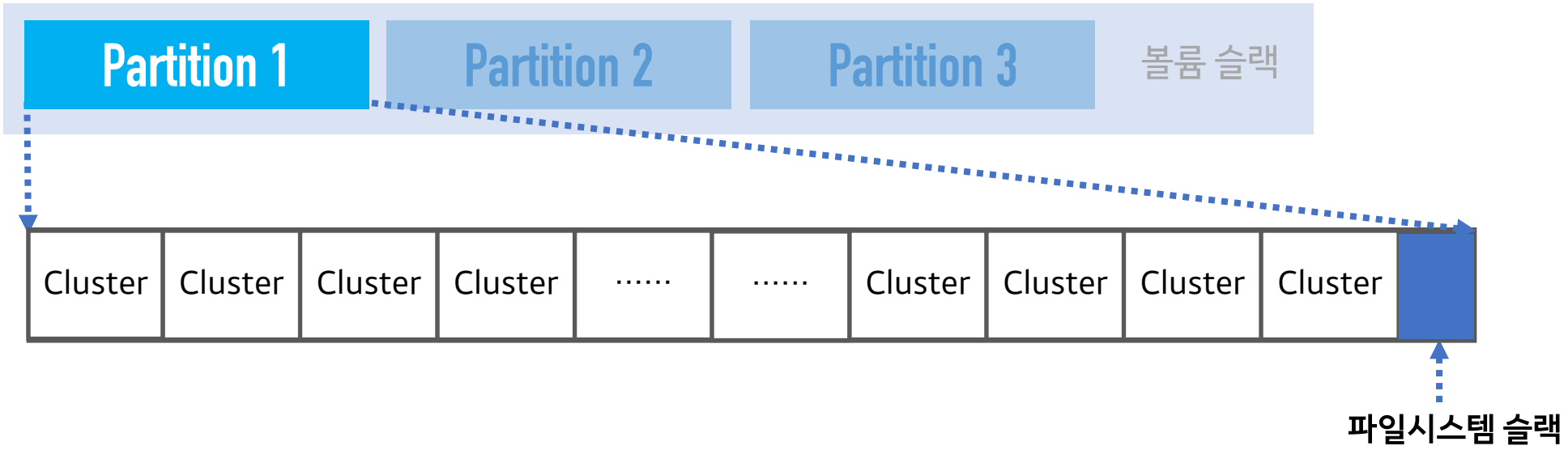
- 저장매체의 물리적 구조와 논리적 구조의 차이로 낭비되는 공간



주요 용어 및 개념

슬랙 (Slack)

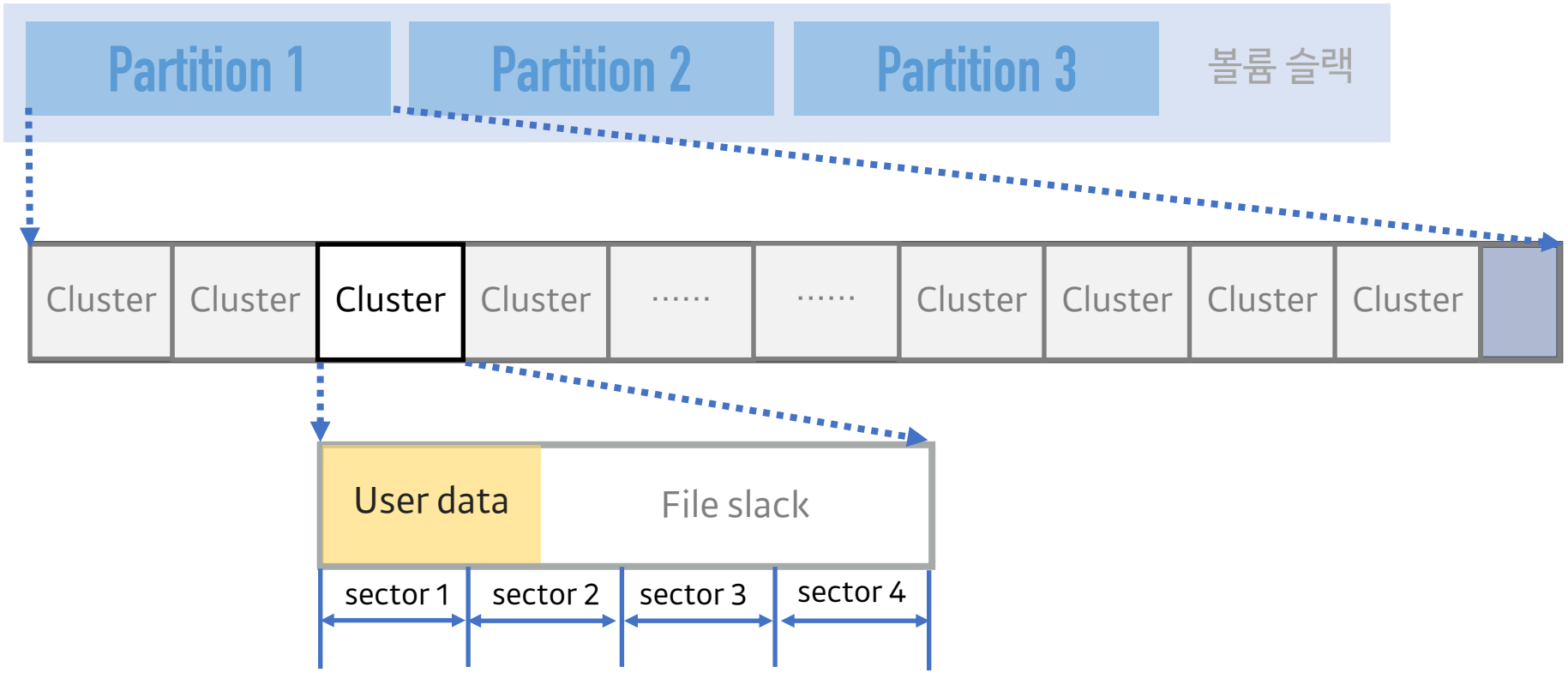
- 저장매체의 물리적 구조와 논리적 구조의 차이로 낭비되는 공간



주요 용어 및 개념

슬랙 (Slack)

- 저장매체의 물리적 구조와 논리적 구조의 차이로 낭비되는 공간



주요 용어 및 개념

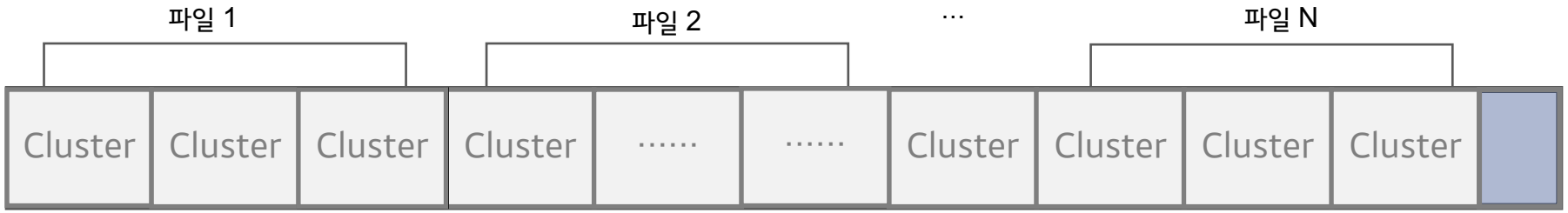
슬랙 (Slack) - 포맷프리 파일시스템 예제

- 저장매체의 물리적 구조와 논리적 구조의 차이로 낭비되는 공간
- MS에서 만들어진 FS는 Cluster 단위로 I/O 지원. 그 결과 파일의 크기도 클러스터의 배수

주요 용어 및 개념

슬랙 (Slack) - 포맷프리 파일시스템 예제

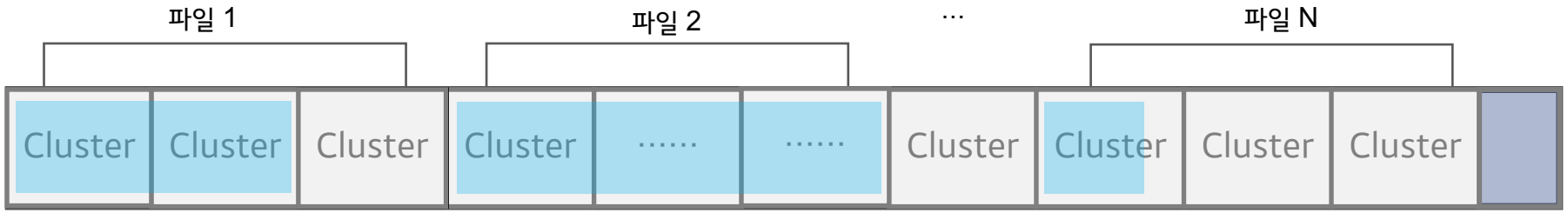
- 저장매체의 물리적 구조와 논리적 구조의 차이로 낭비되는 공간
- MS에서 만들어진 FS는 Cluster 단위로 I/O 지원. 그 결과 파일의 크기도 클러스터의 배수



주요 용어 및 개념

슬랙 (Slack) - 포맷프리 파일시스템 예제

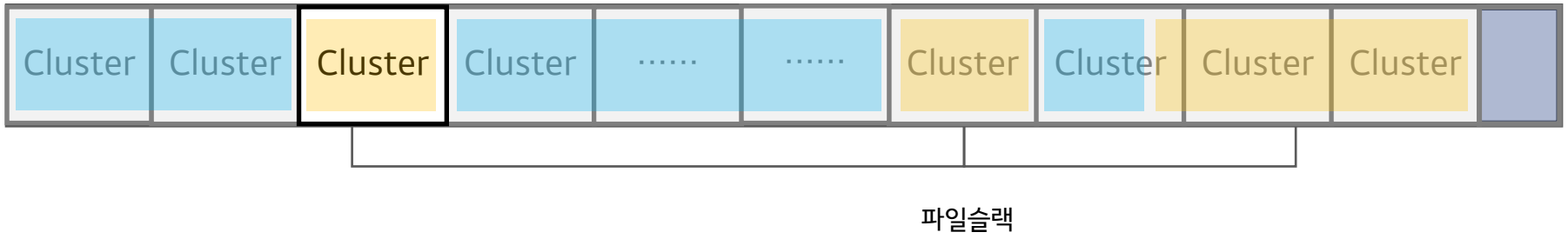
- 저장매체의 물리적 구조와 논리적 구조의 차이로 낭비되는 공간
- MS에서 만들어진 FS는 Cluster 단위로 I/O 지원. 그 결과 파일의 크기도 클러스터의 배수



주요 용어 및 개념

슬랙 (Slack) - 포맷프리 파일시스템 예제

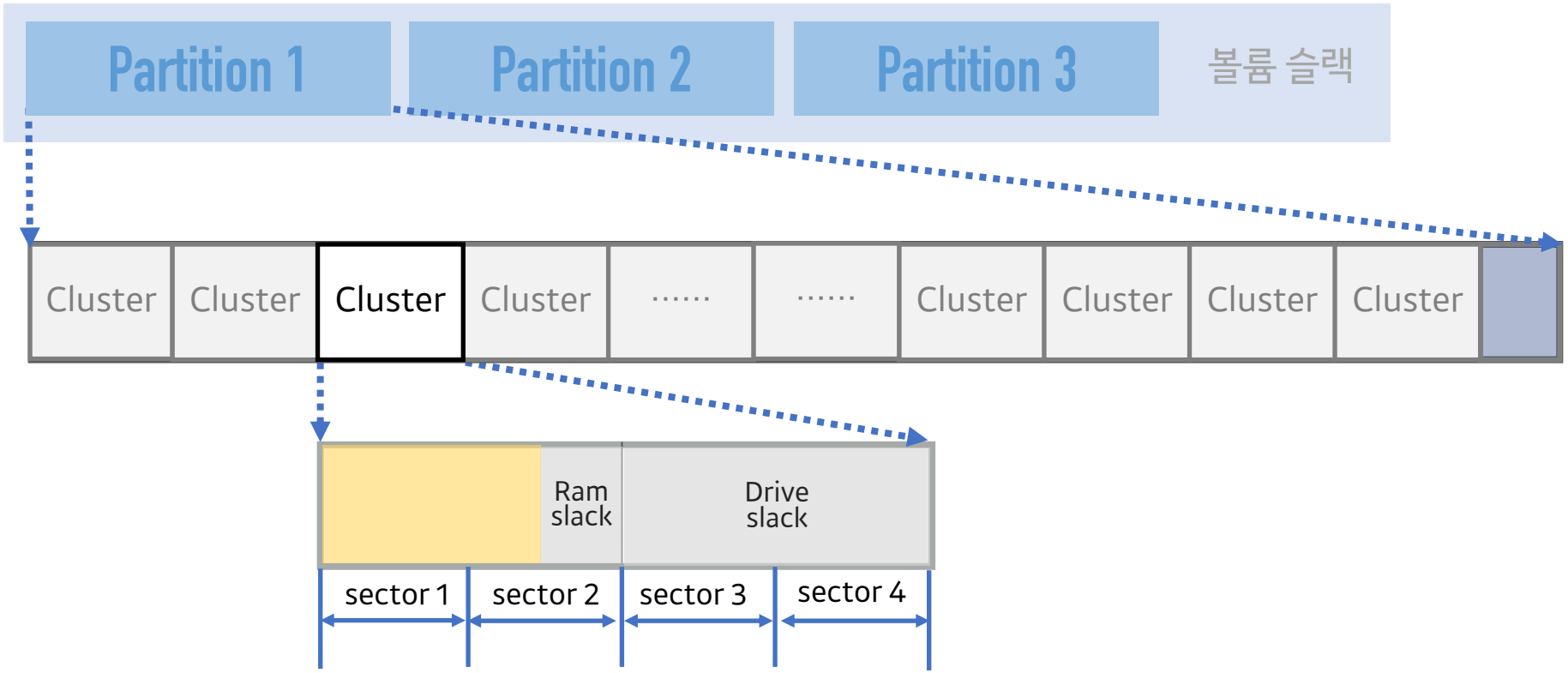
- 저장매체의 물리적 구조와 논리적 구조의 차이로 낭비되는 공간
- MS에서 만들어진 FS는 Cluster 단위로 I/O 지원. 그 결과 파일의 크기도 클러스터의 배수



주요 용어 및 개념

슬랙 (Slack)

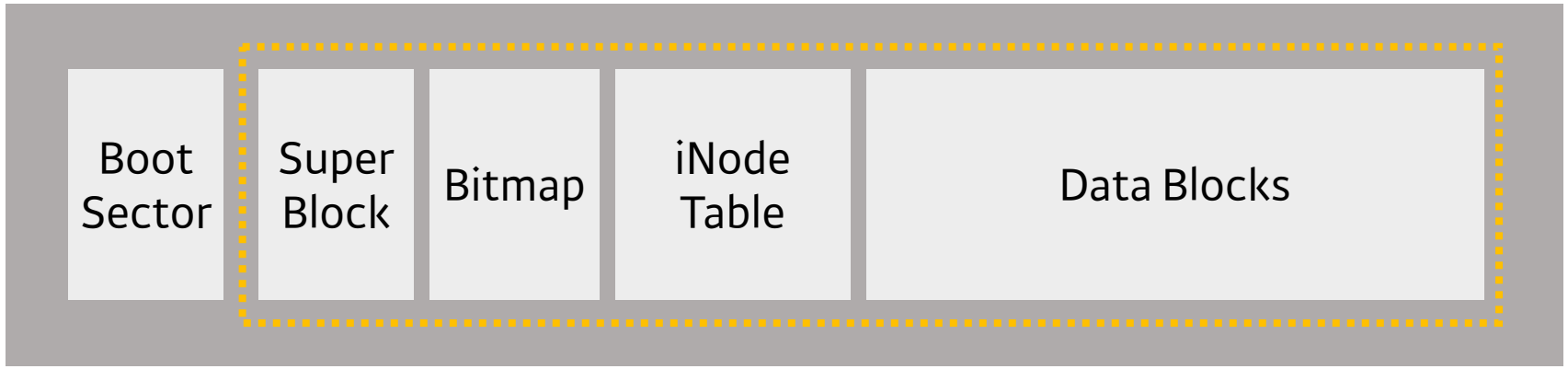
- 저장매체의 물리적 구조와 논리적 구조의 차이로 낭비되는 공간



주요 용어 및 개념

파티션 구조

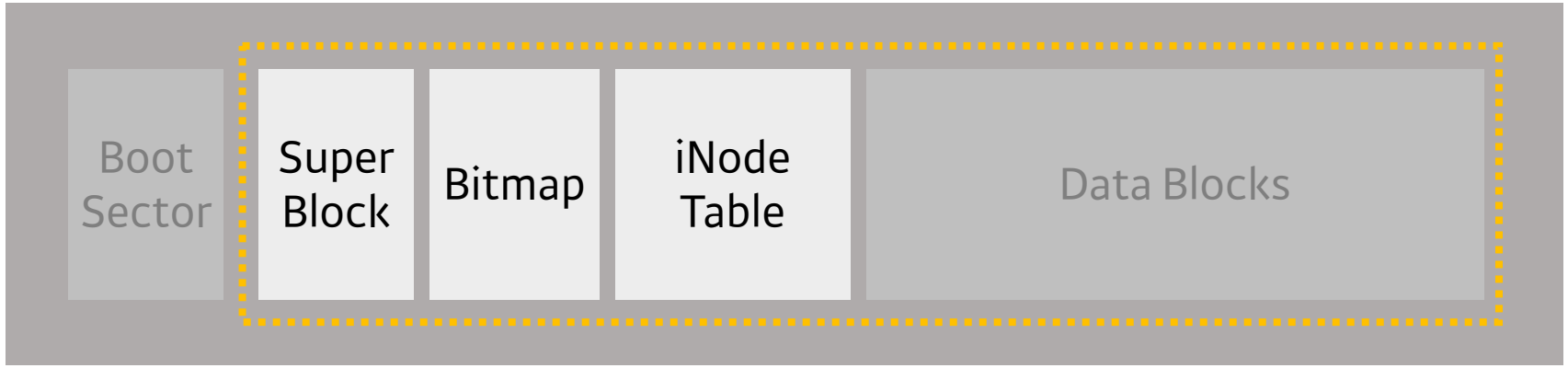
- 일반적인 파티션의 구조



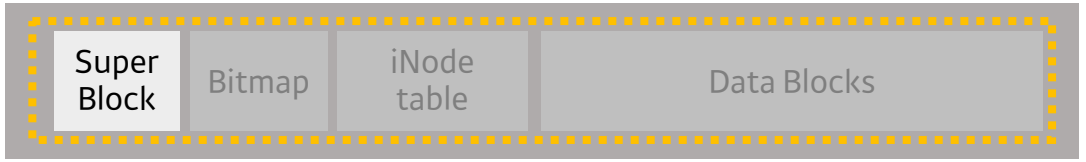
주요 용어 및 개념

파티션 구조

- 일반적인 파티션의 구조
- 파티션 메타 데이터



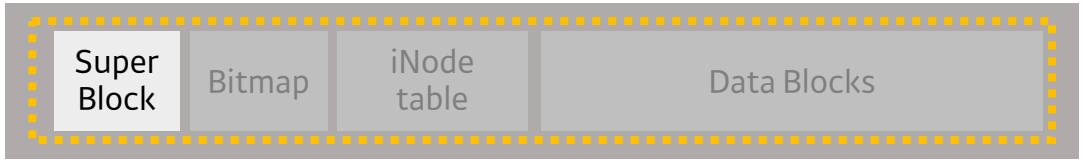
주요 용어 및 개념



슈퍼블록

- 파일시스템의 가장 기본이 되는 정보, 통계치를 저장하는 자료 구조
  - 블록의 크기
  - 각 영역의 위치 (저널, 비트맵, 아이노드 테이블)
  - 루트 아이노드 번호
  - 사용 중인 아이노드 개수
  - F/S 크기

주요 용어 및 개념

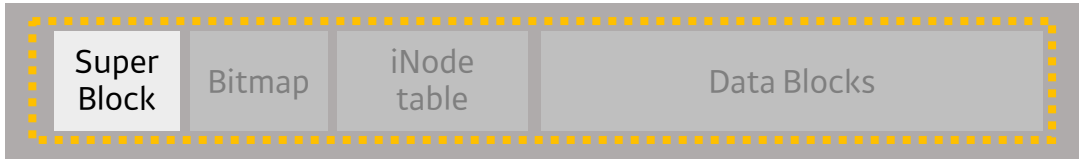


슈퍼블록

- 파일시스템의 가장 기본이 되는 정보, 통계치를 저장하는 자료 구조
  - 블록의 크기
  - 각 영역의 위치 (저널, 비트맵, 아이노드 테이블)
  - 루트 아이노드 번호
  - 사용 중인 아이노드 개수
  - F/S 크기

"F/S Geometry"

주요 용어 및 개념

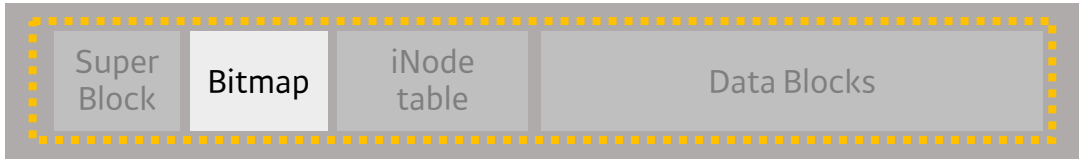


슈퍼블록

- 파일시스템의 가장 기본이 되는 정보, 통계치를 저장하는 자료 구조
  - 블록의 크기
  - 각 영역의 위치 (저널, 비트맵, 아이노드 테이블)
  - 루트 아이노드 번호
  - 사용 중인 아이노드 개수
  - F/S 크기

“F/S Statistics”

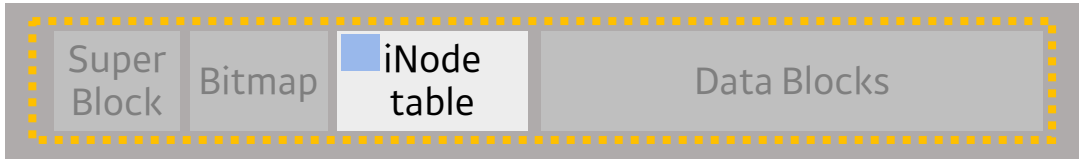
주요 용어 및 개념



비트맵

- 데이터 블록의 각 블록의 사용 여부를 비트맵 영역의 해당 위치의 비트로 표현
  - 비할당 영역 식별

주요 용어 및 개념



아이노드

- 파일 혹은 디렉토리

기본 정보

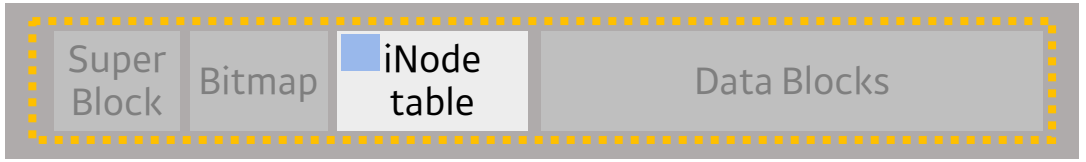
- 시각 (생성, 수정)
- 크기
- 속성 (파일, 폴더)
- 이름

인라인 데이터

혹은

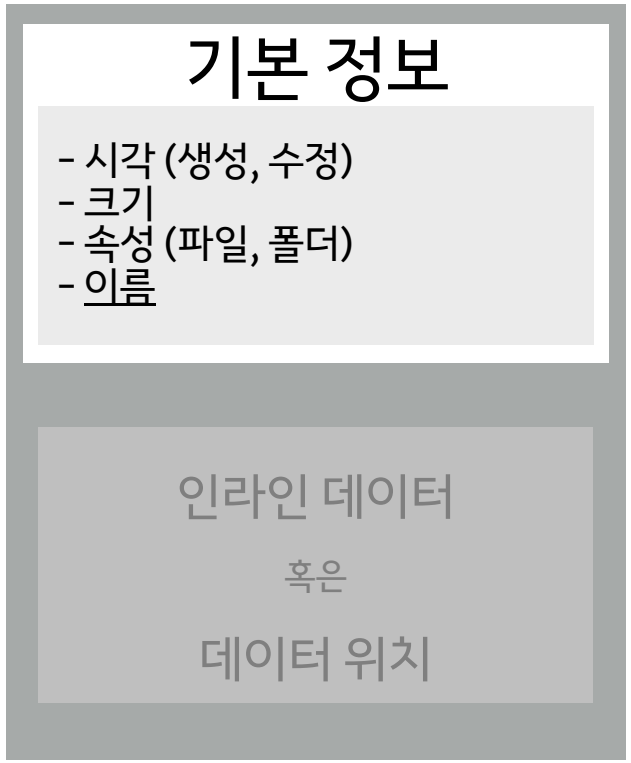
데이터 위치

주요 용어 및 개념

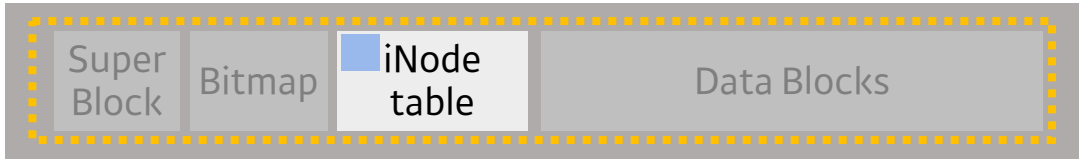


아이노드

- 파일 혹은 디렉토리

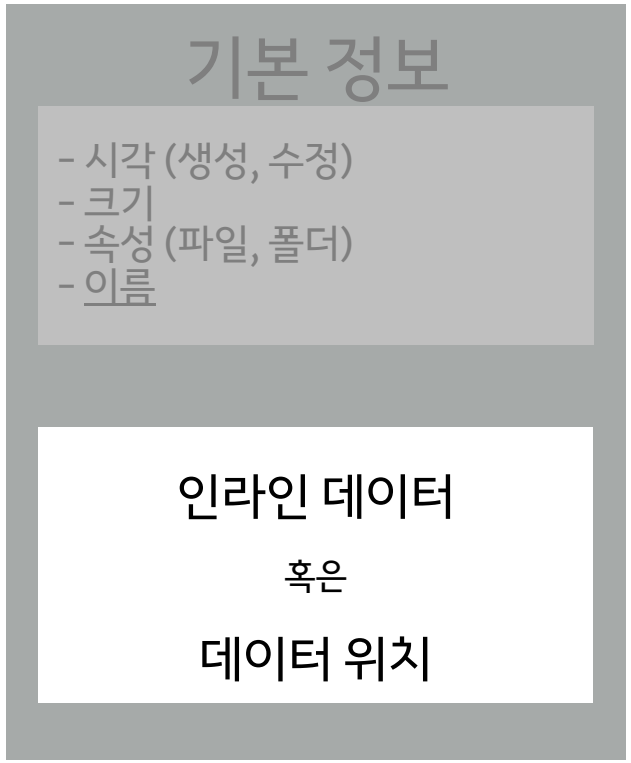


주요 용어 및 개념

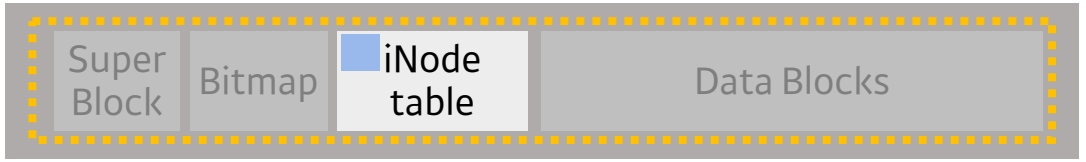


아이노드

- 파일 혹은 디렉토리, 고정크기

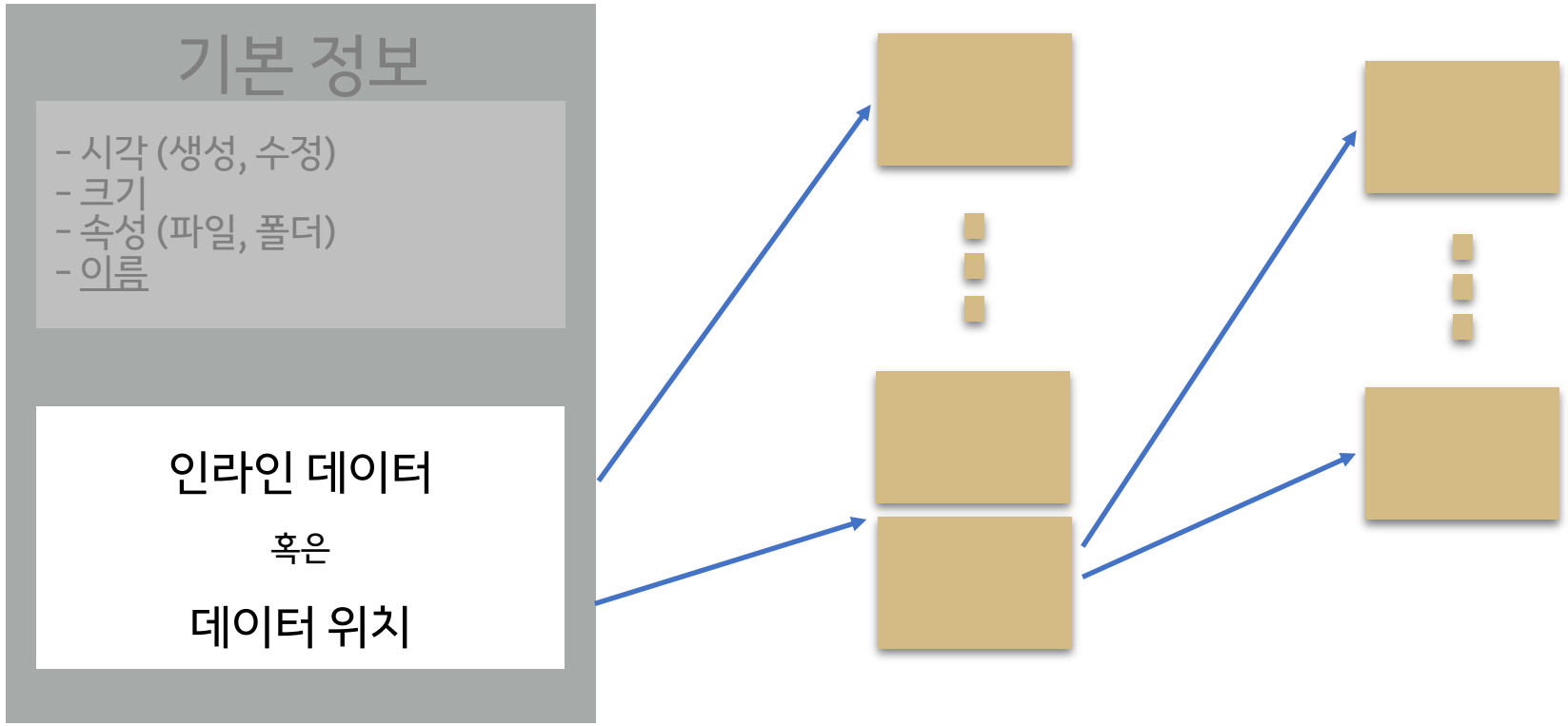


주요 용어 및 개념



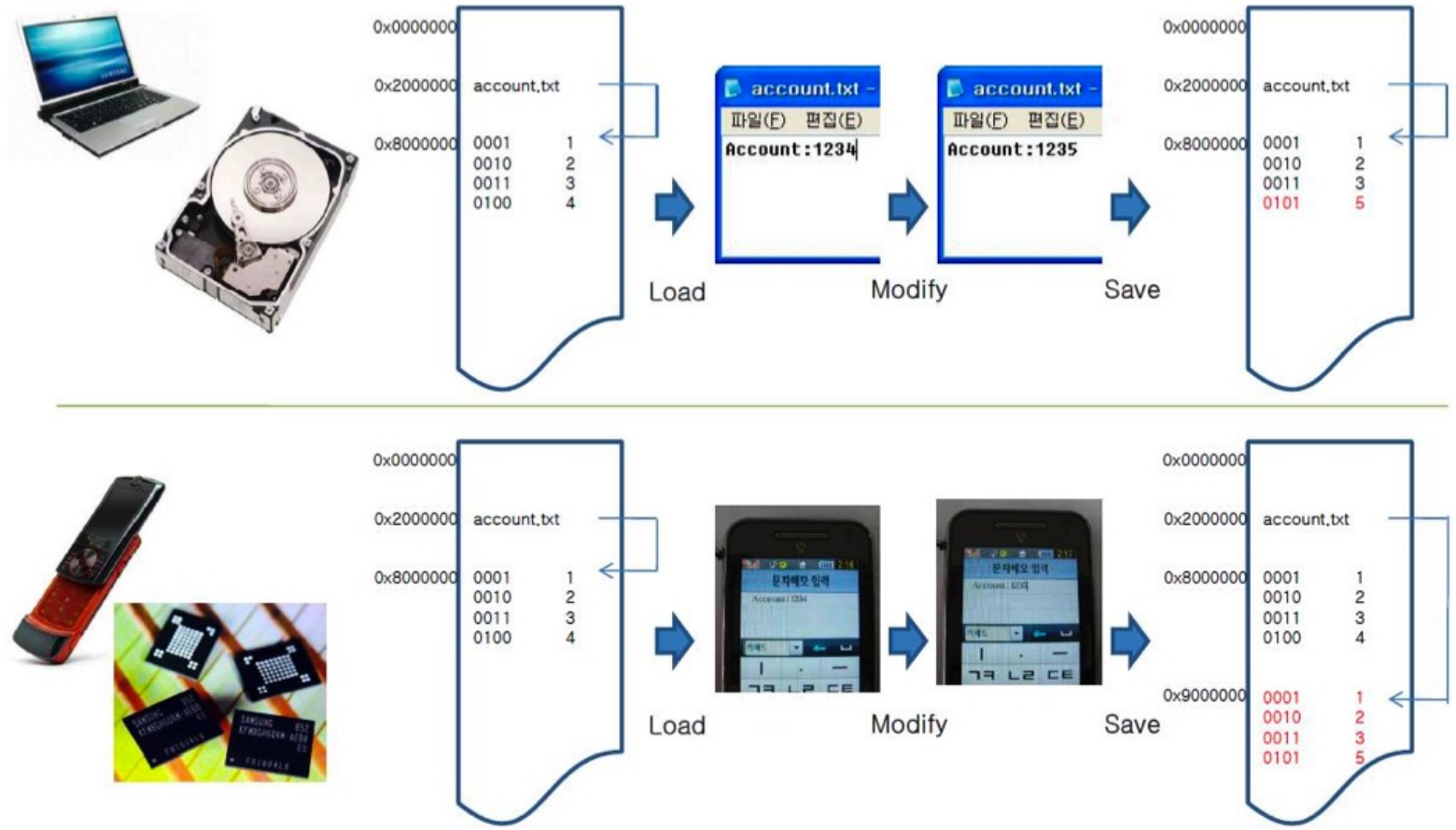
아이노드

- 파일 혹은 디렉토리, 고정크기

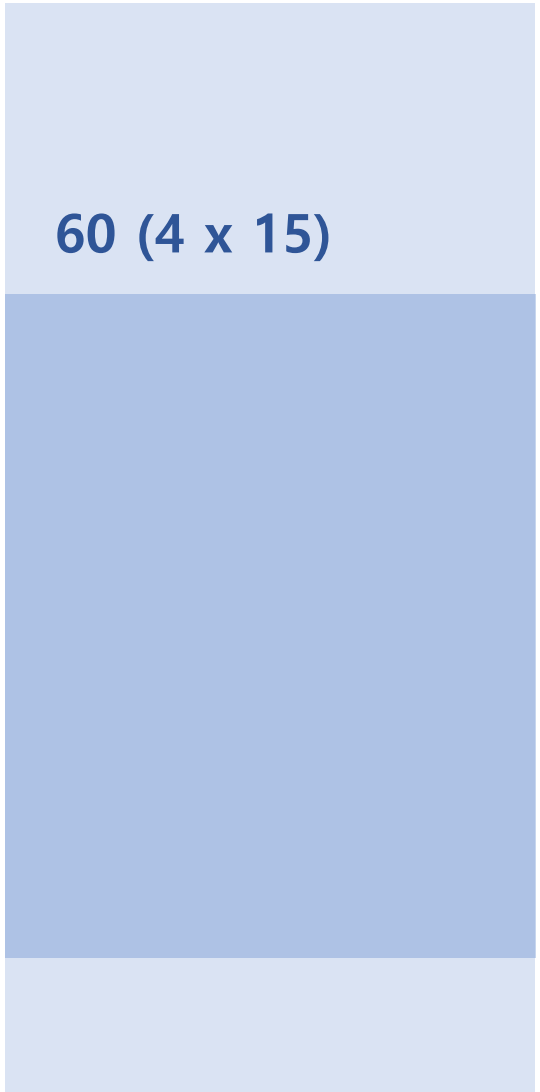


주요 용어 및 개념

참고) Wear Leveling

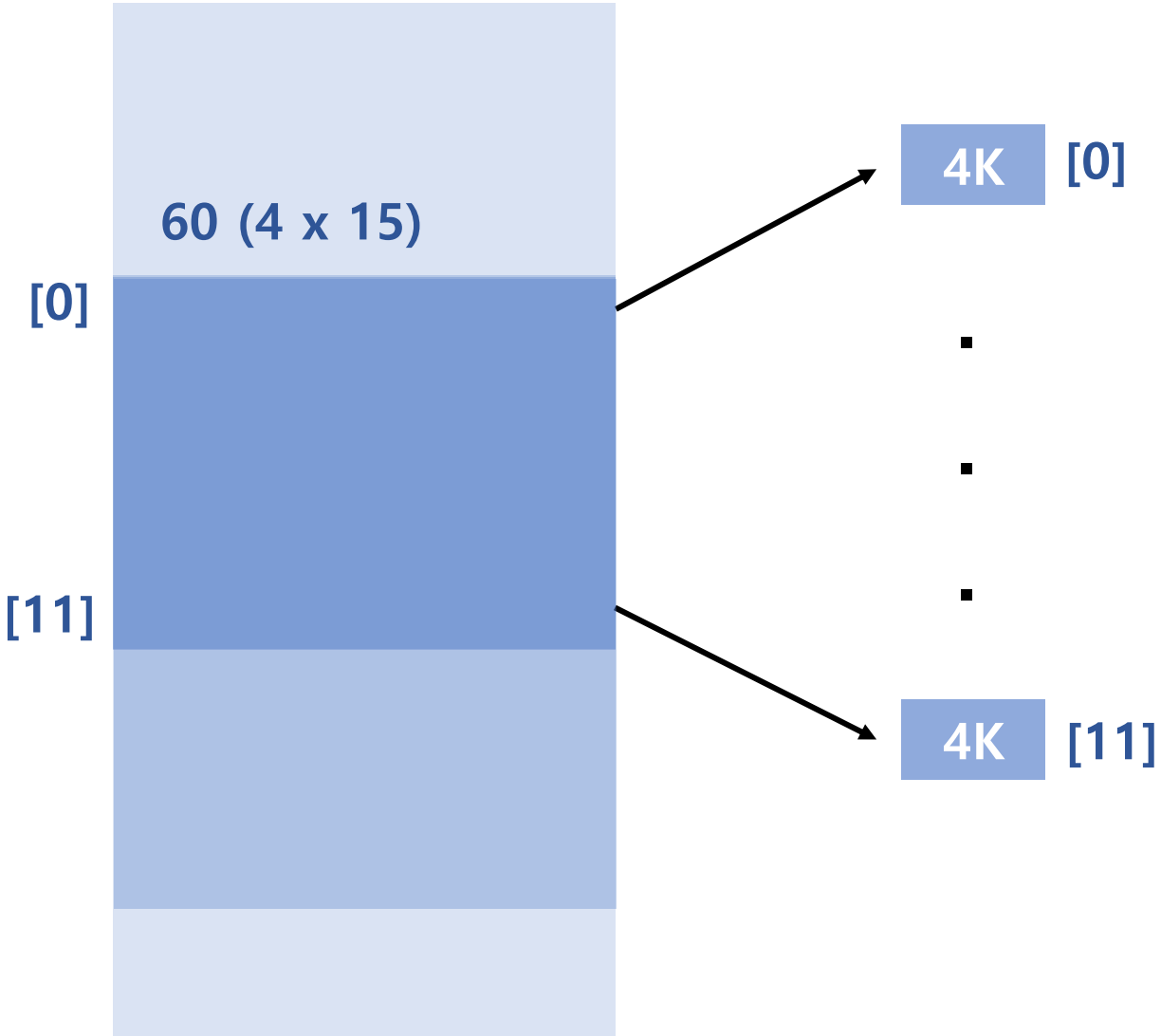


주요 용어 및 개념 -

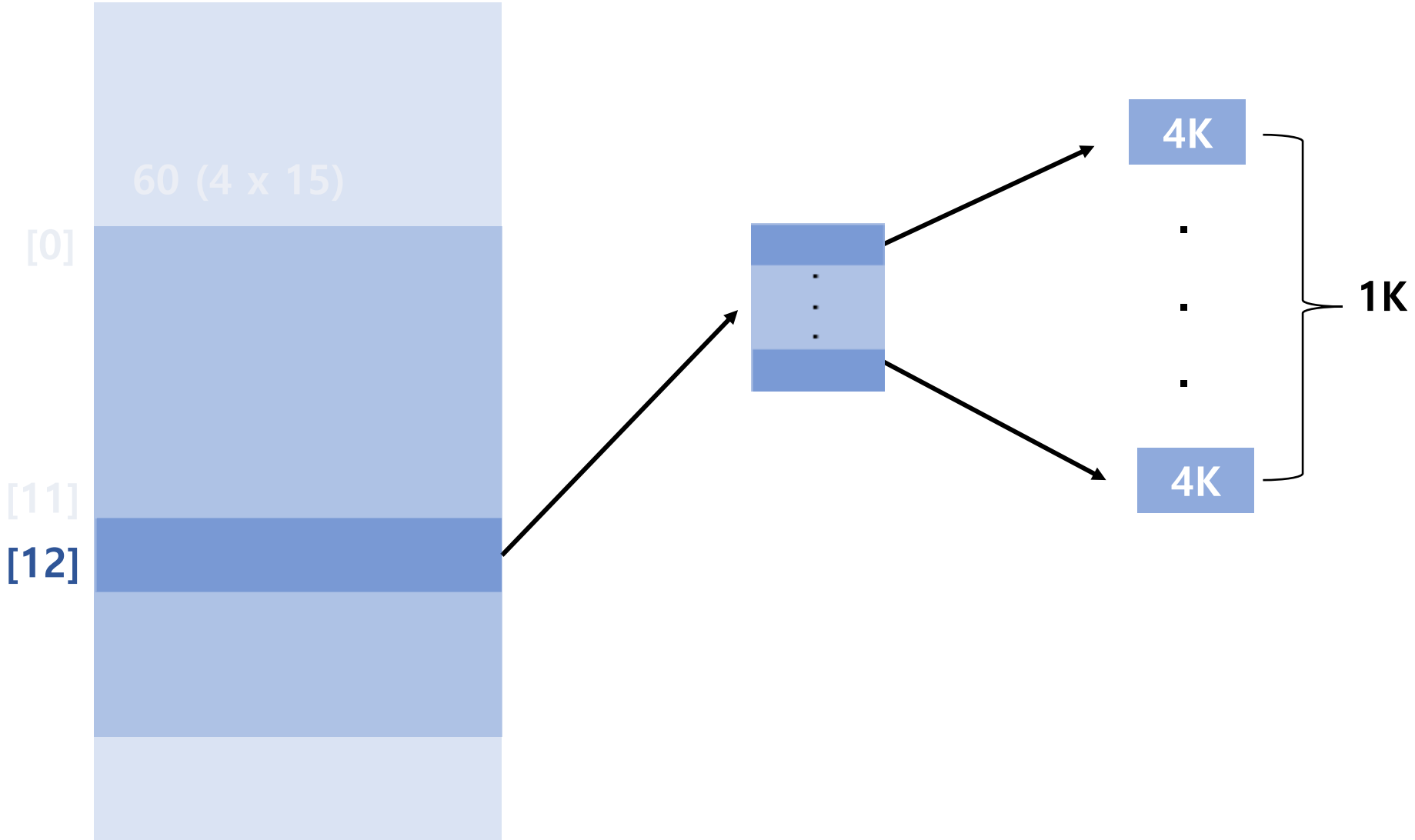


ex) Ext3 4k block size
triple-indirection

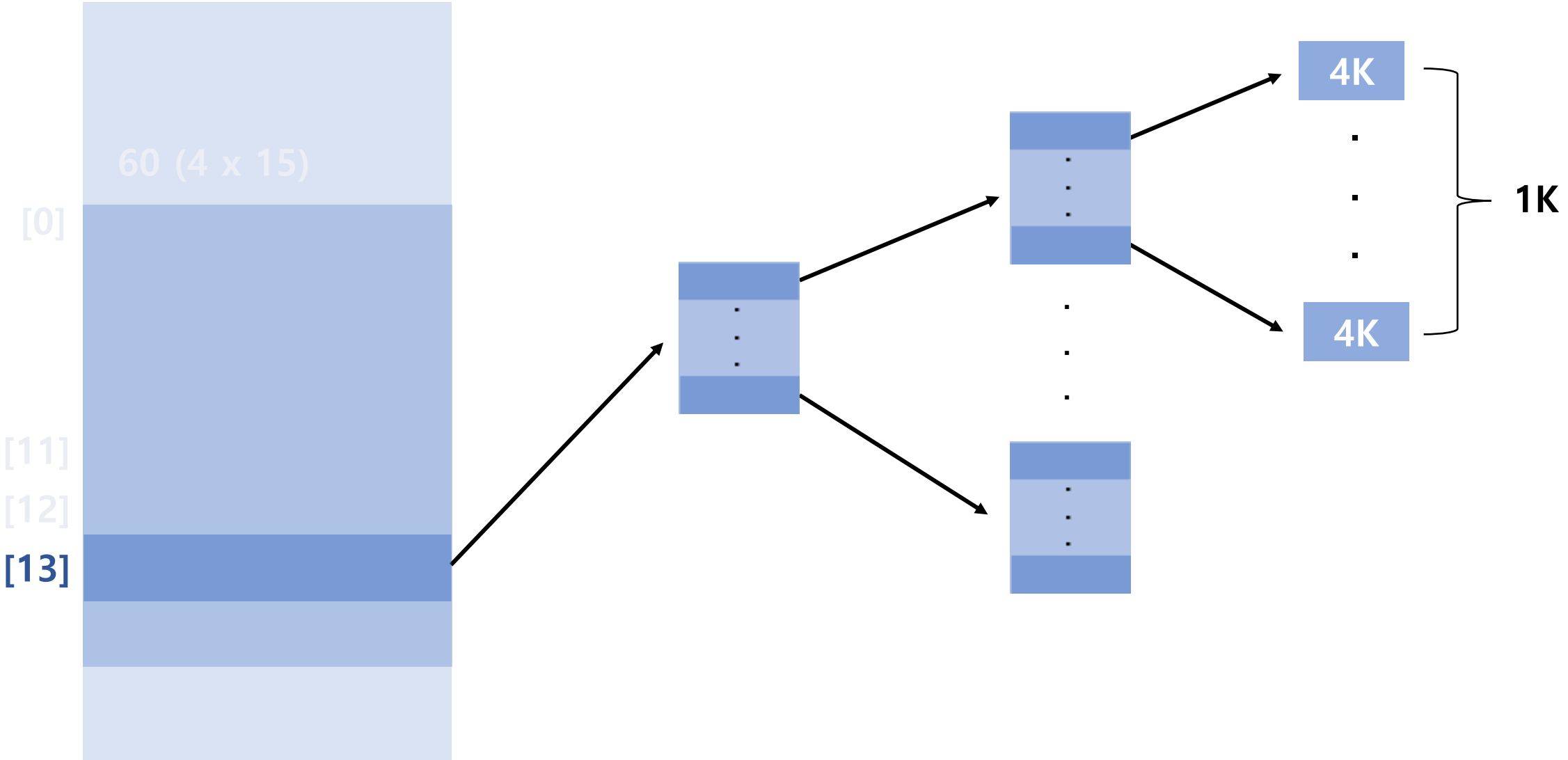
주요 용어 및 개념



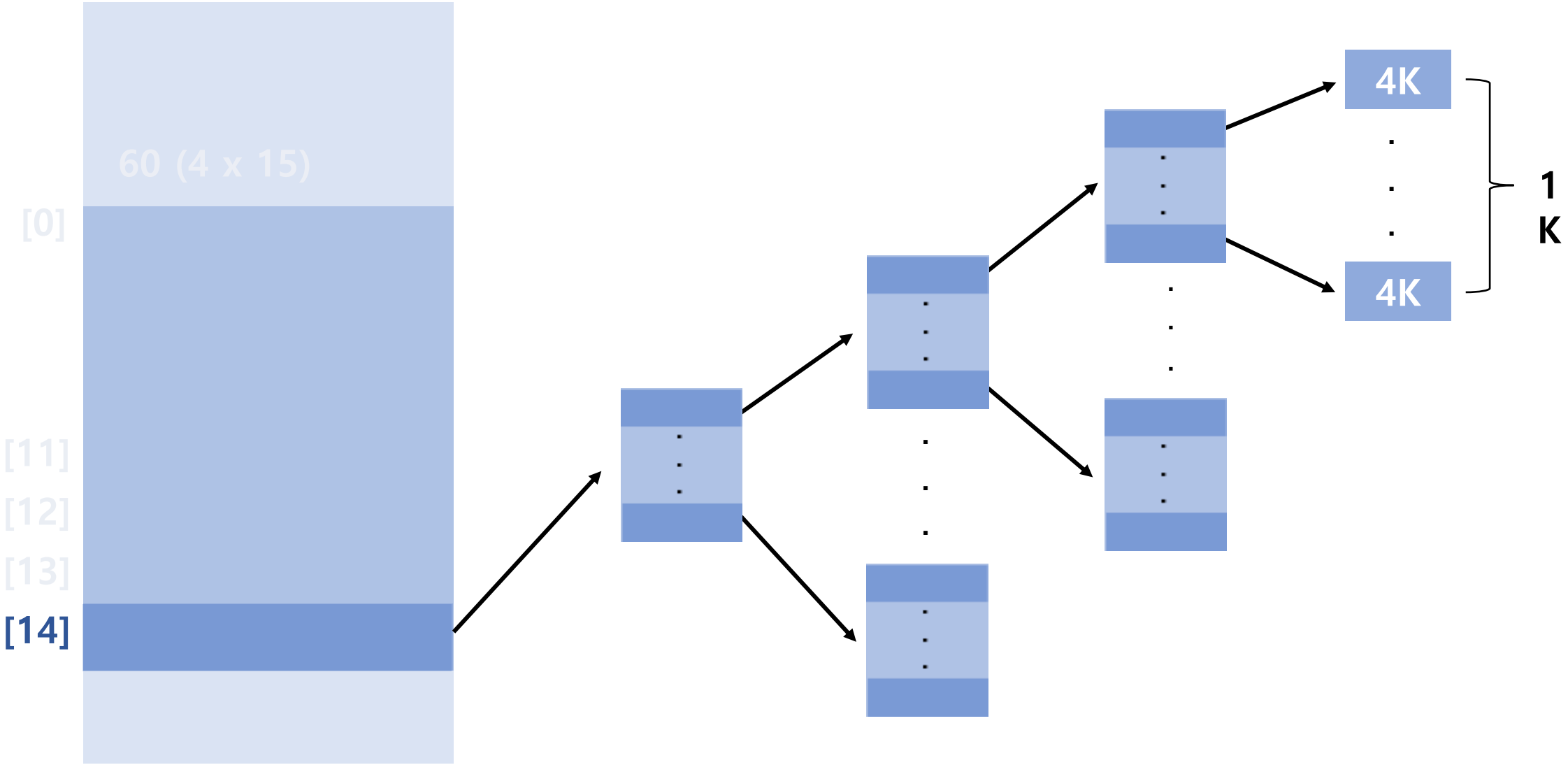
주요 용어 및 개념



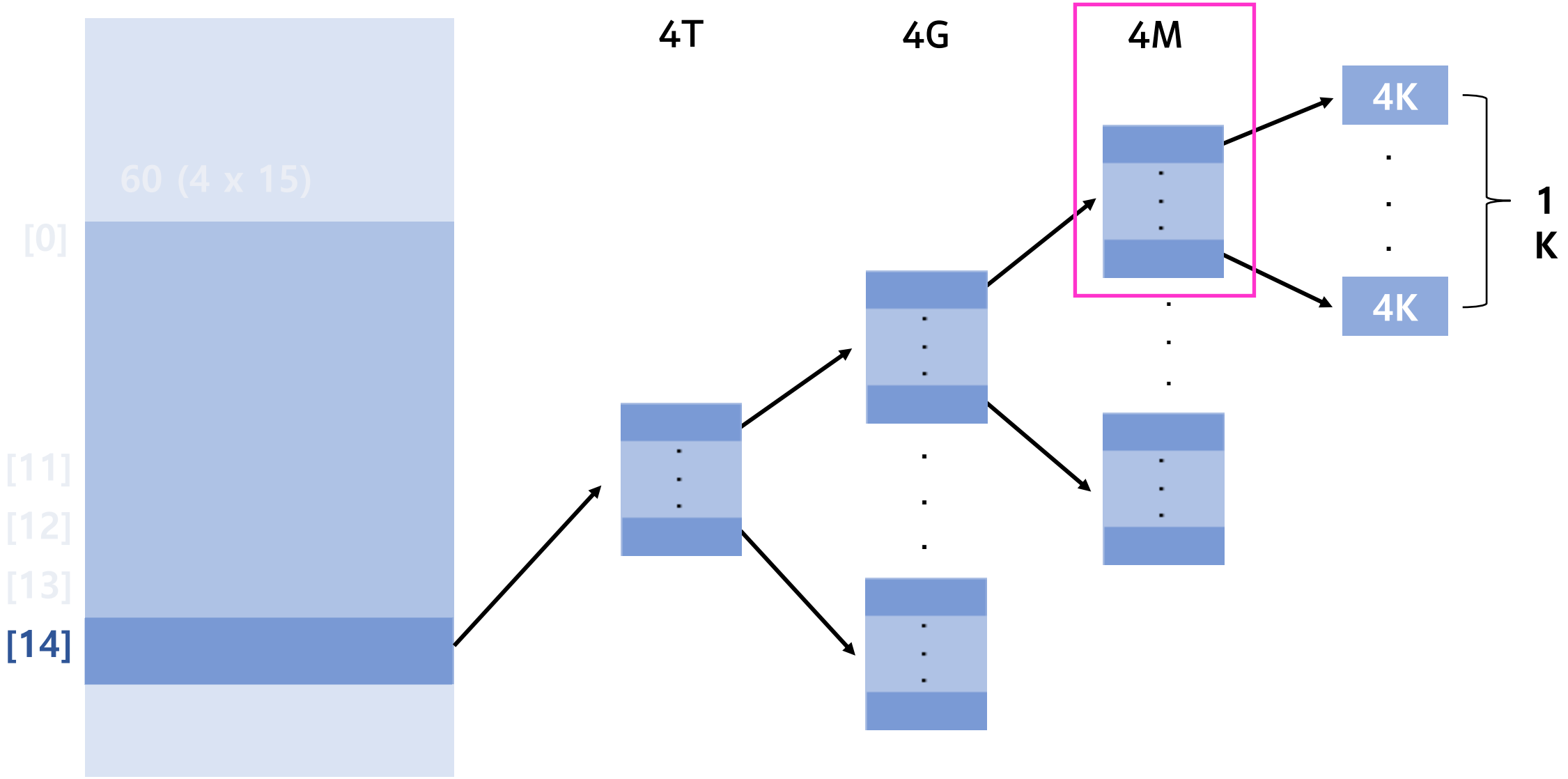
주요 용어 및 개념



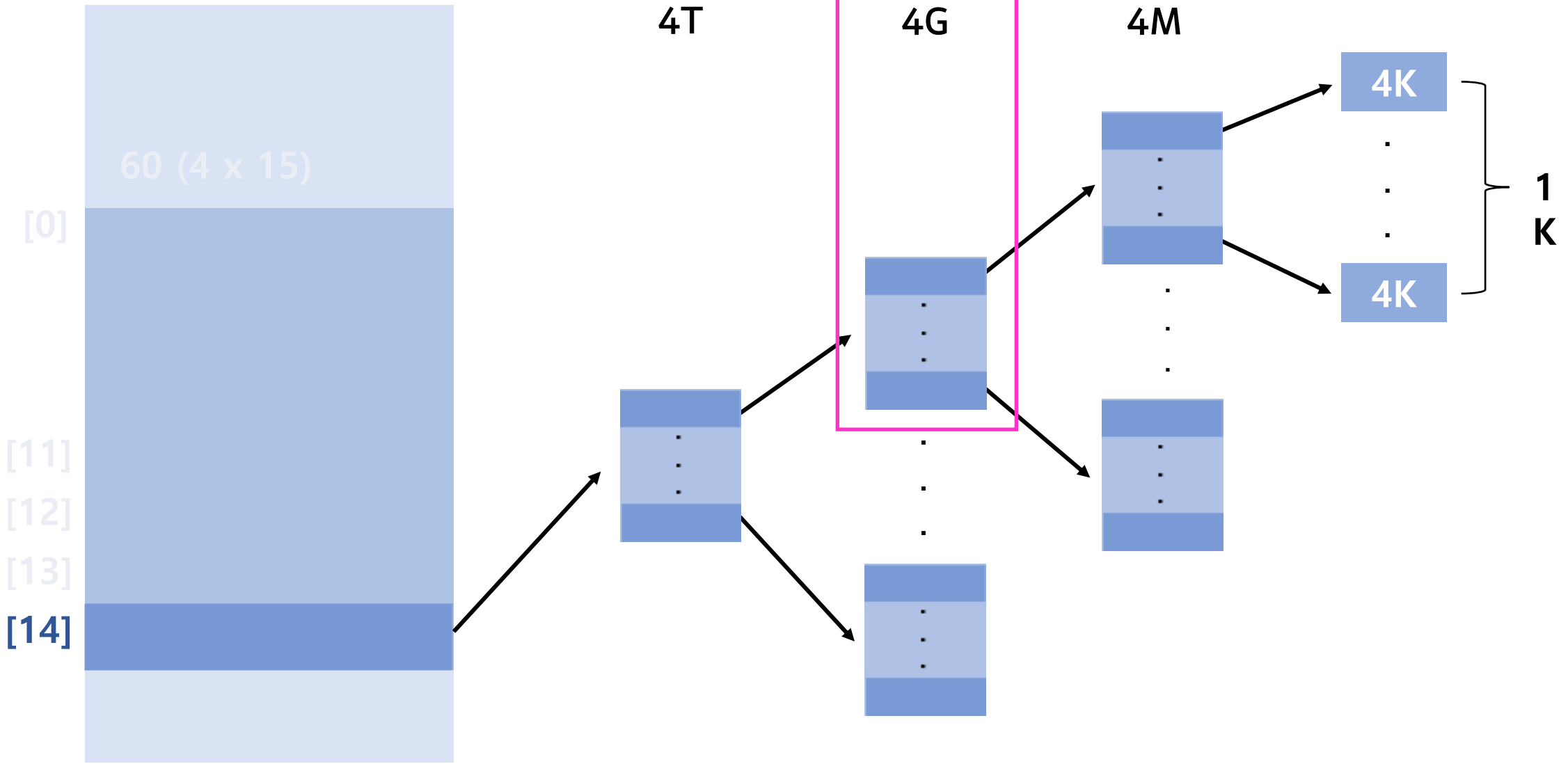
주요 용어 및 개념



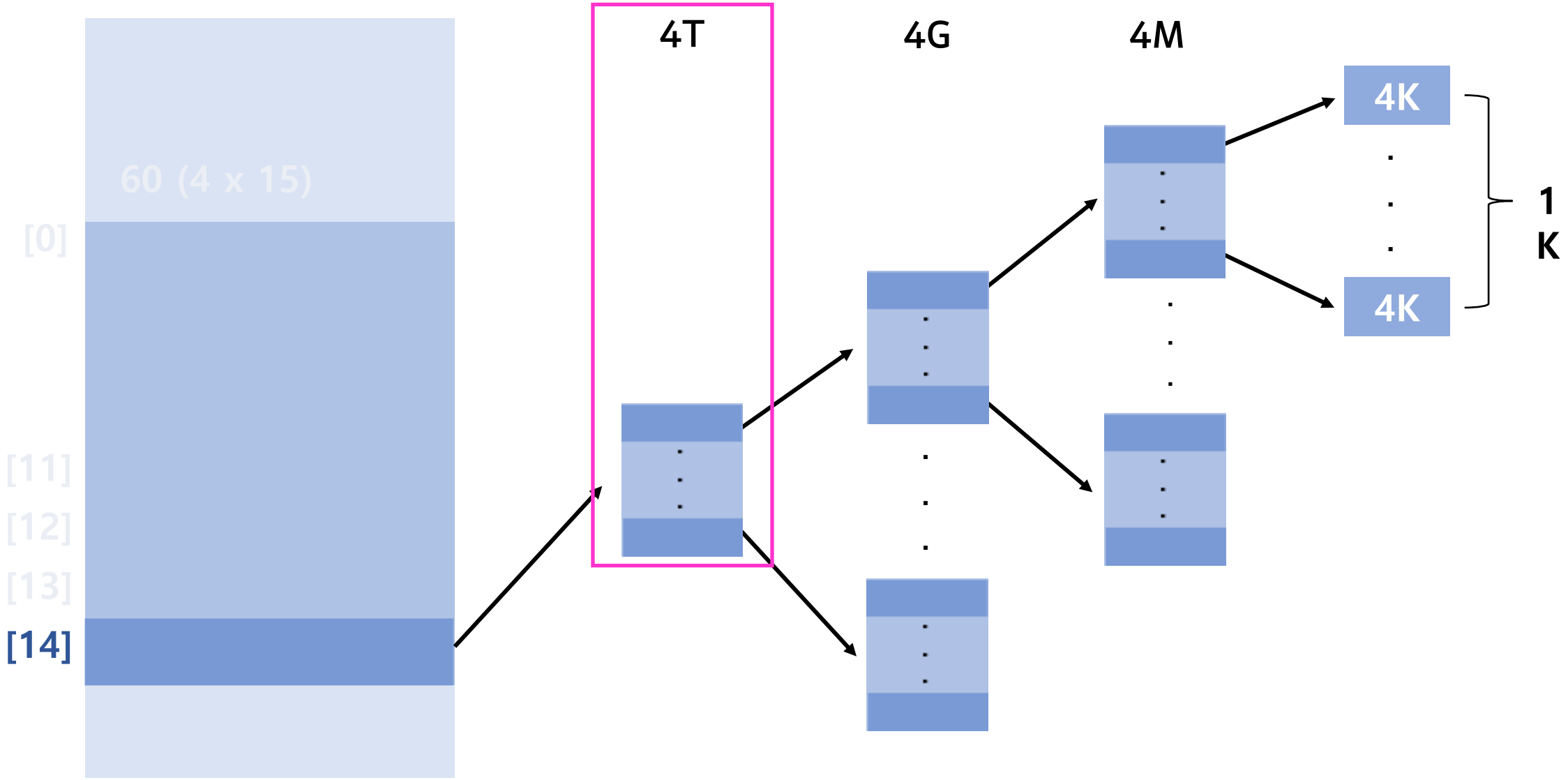
주요 용어 및 개념



주요 용어 및 개념



주요 용어 및 개념



주요 용어 및 개념

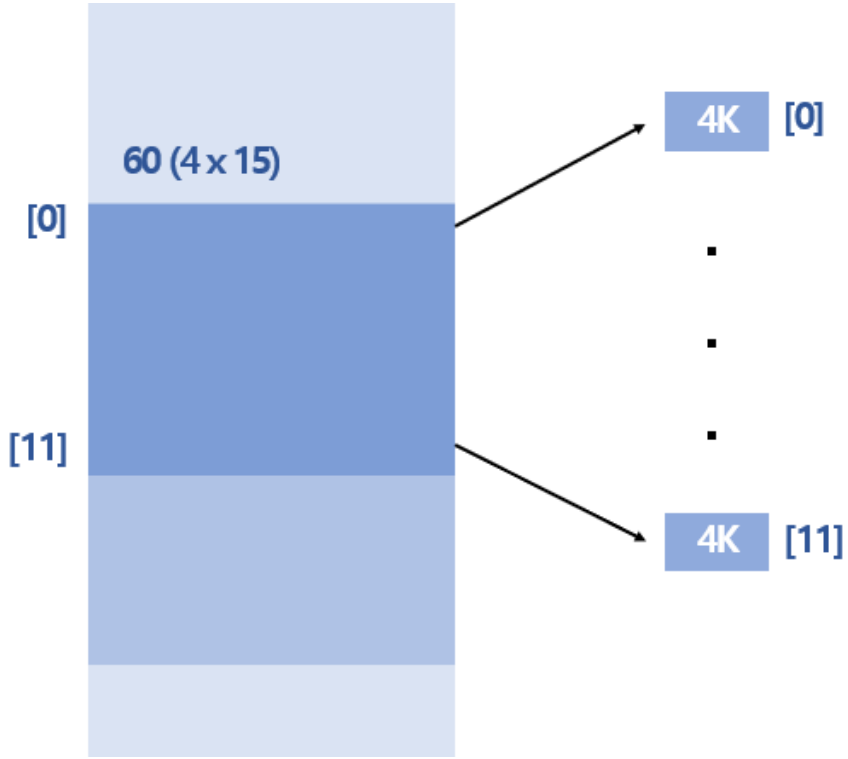
ex) 52K

주요 용어 및 개념

ex) 52K

52/4 = 13

1) [0, 11] 블록은 Direct Blocks에 저장



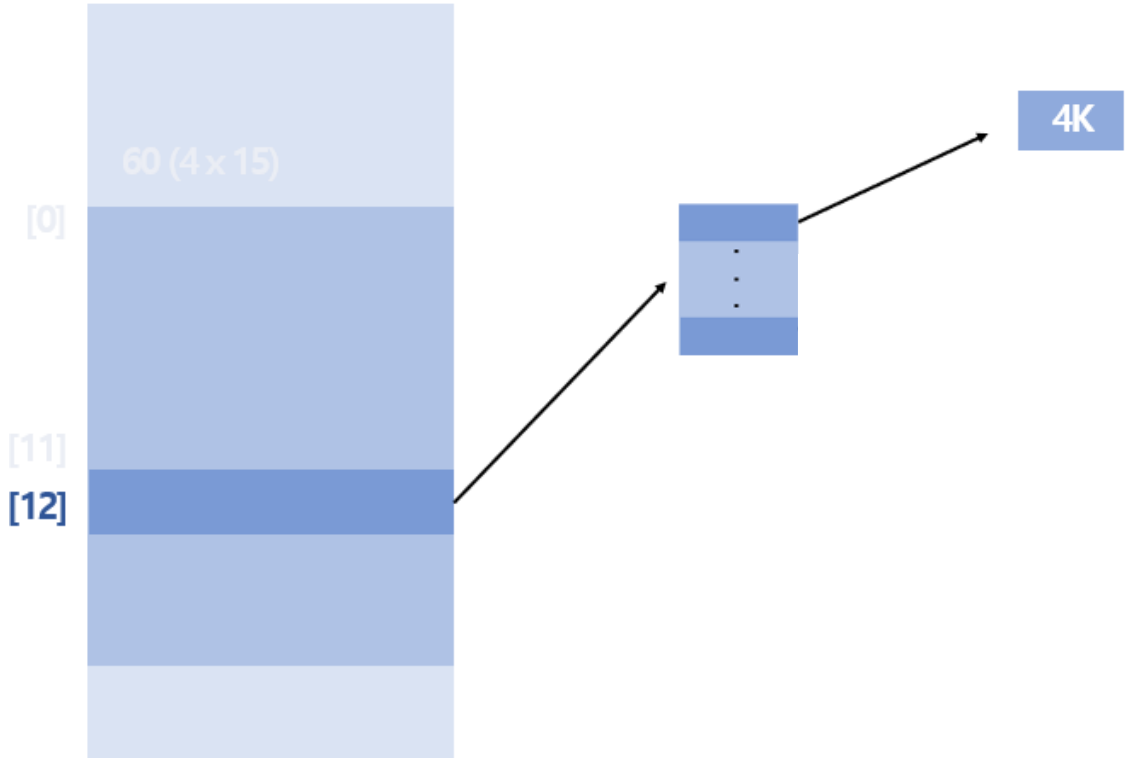
주요 용어 및 개념

ex) 52K

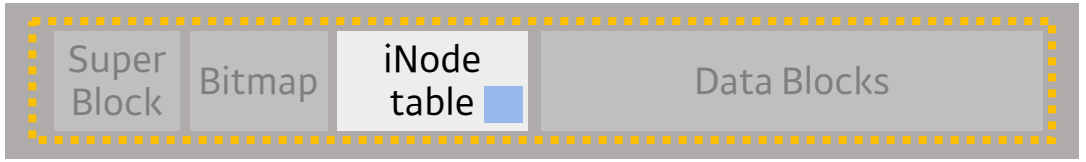
52/4 = 13

1) [0, 11] 블록은 Direct Blocks에 저장

2) 13번째 블록은 Indirect Blocks에 저장

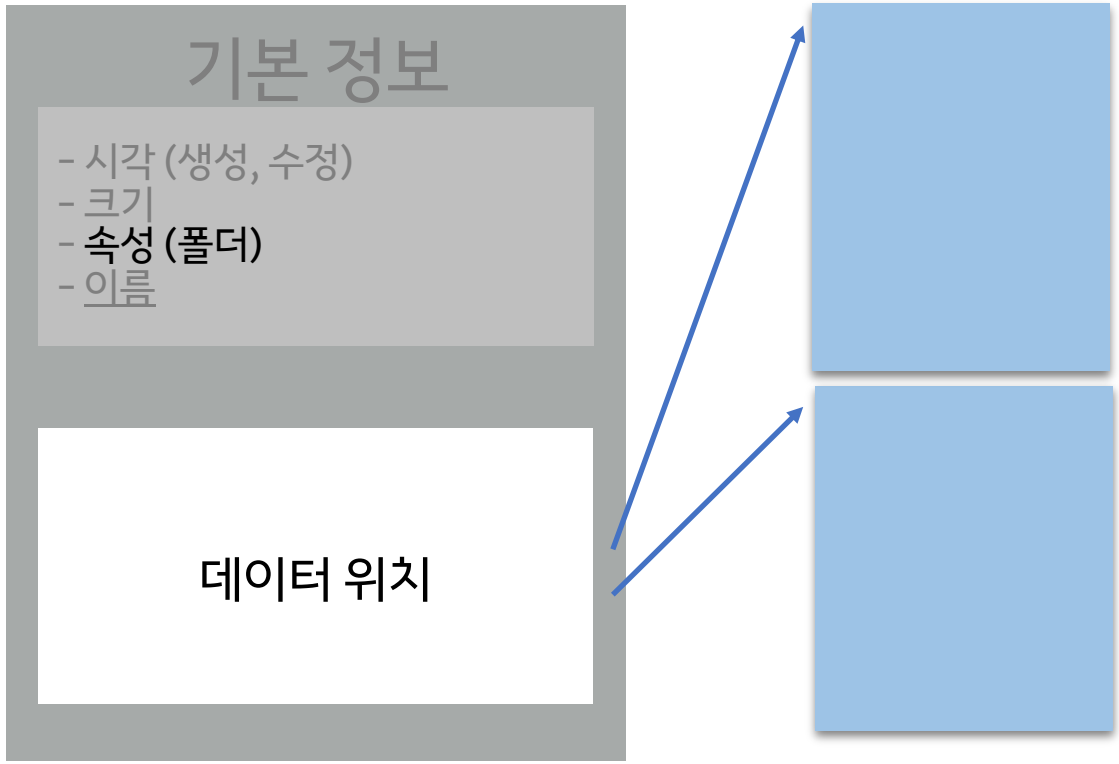


주요 용어 및 개념

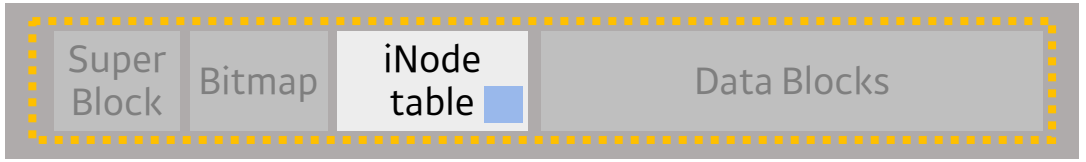


아이노드

- 파일 혹은 디렉토리

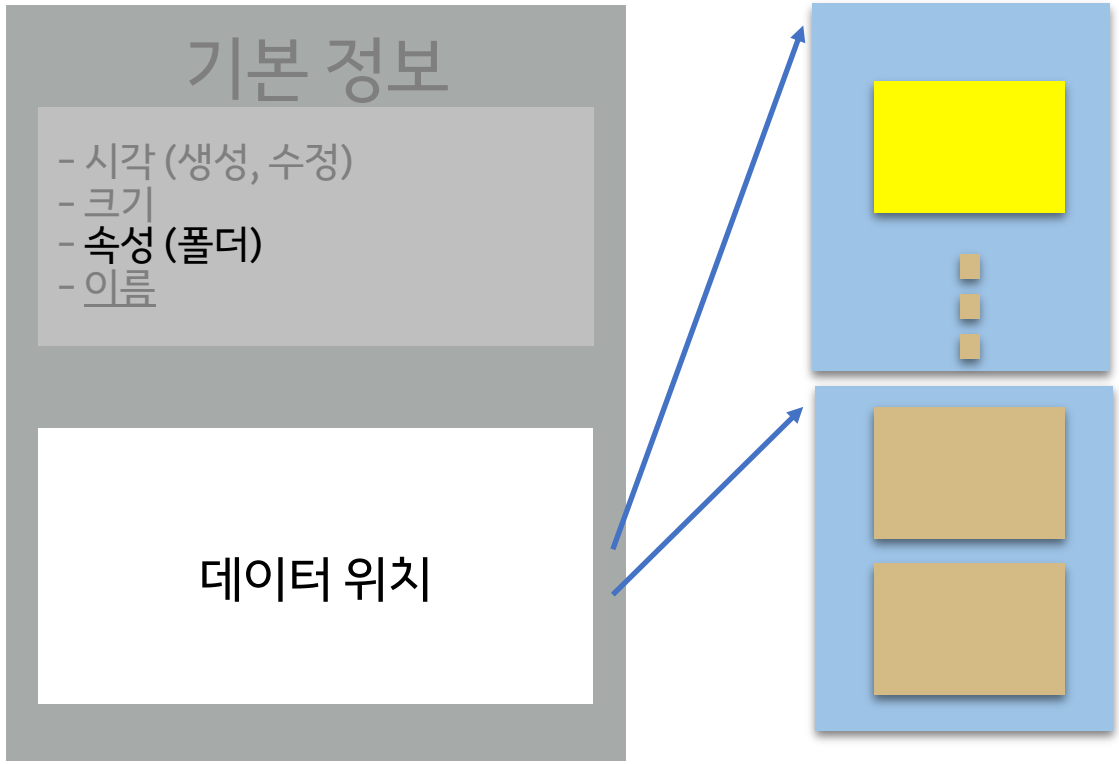


주요 용어 및 개념

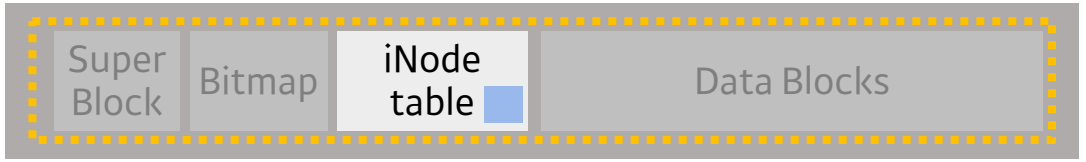


아이노드

- 파일 혹은 디렉토리

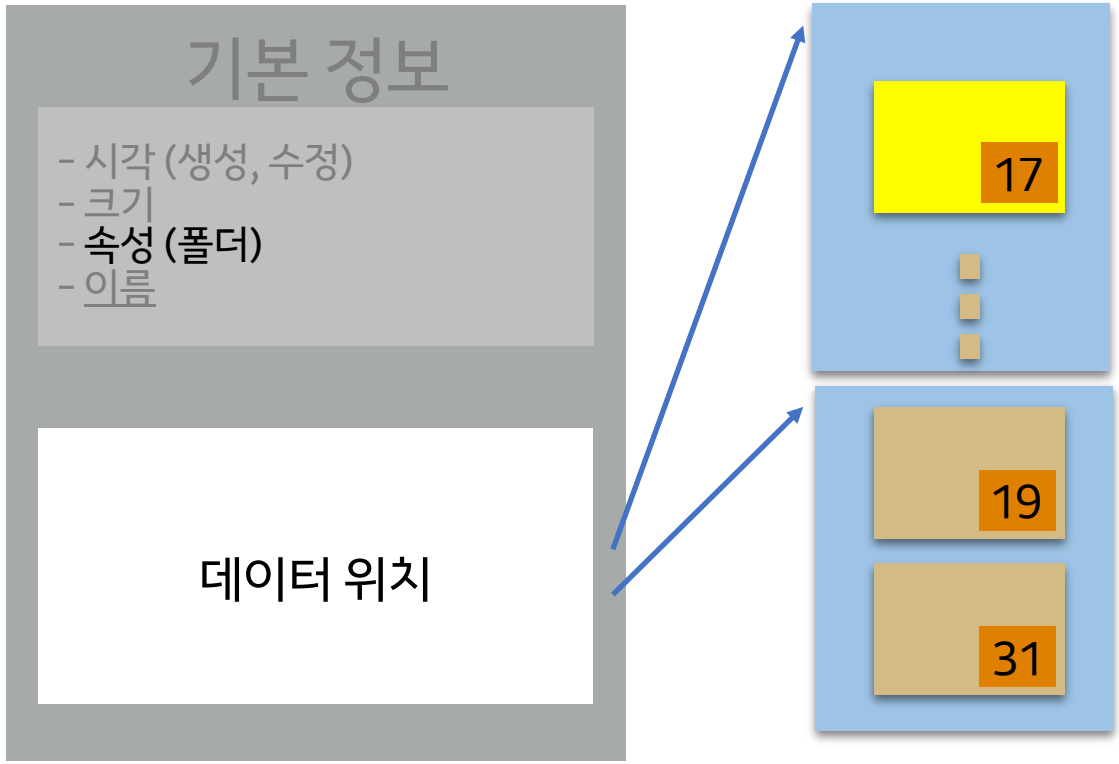


주요 용어 및 개념

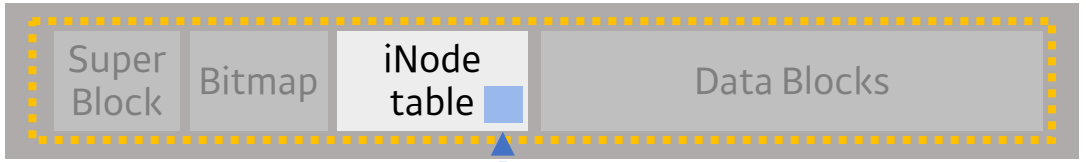


아이노드

- 파일 혹은 디렉토리

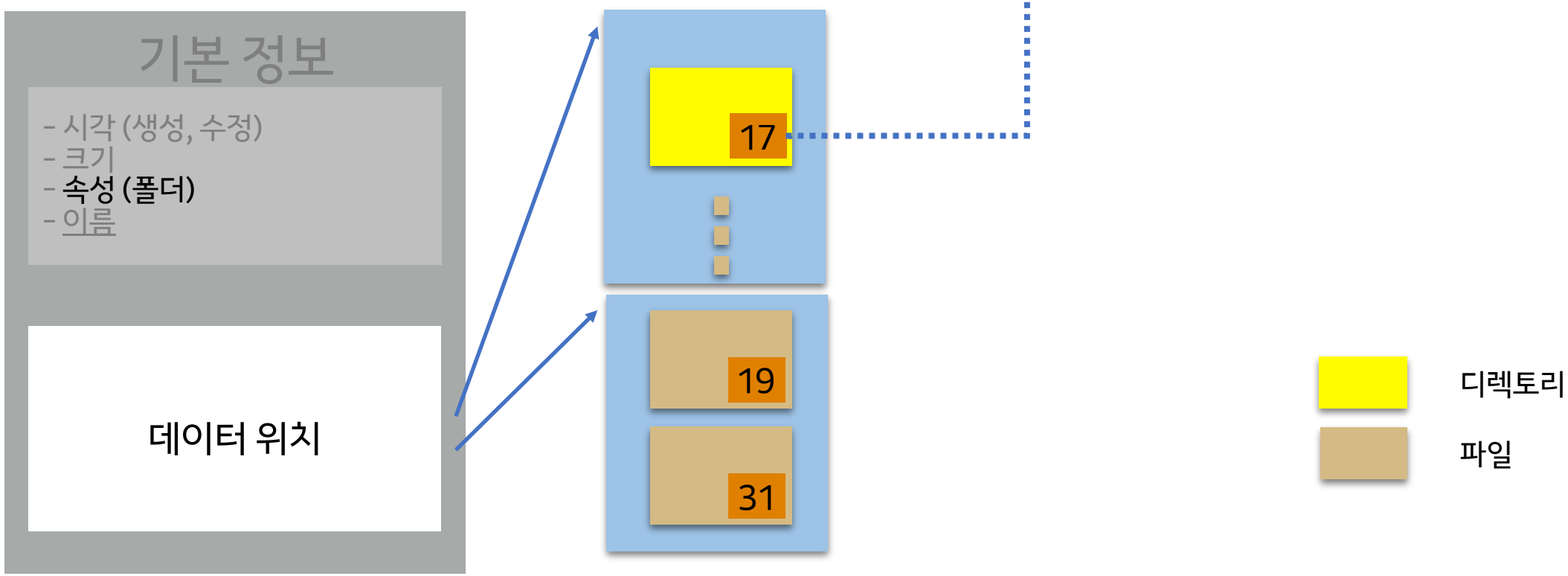


주요 용어 및 개념

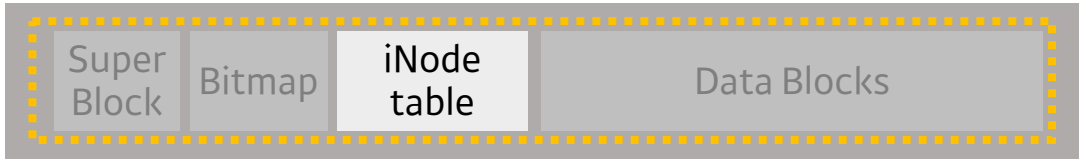


아이노드

- 파일 혹은 디렉토리

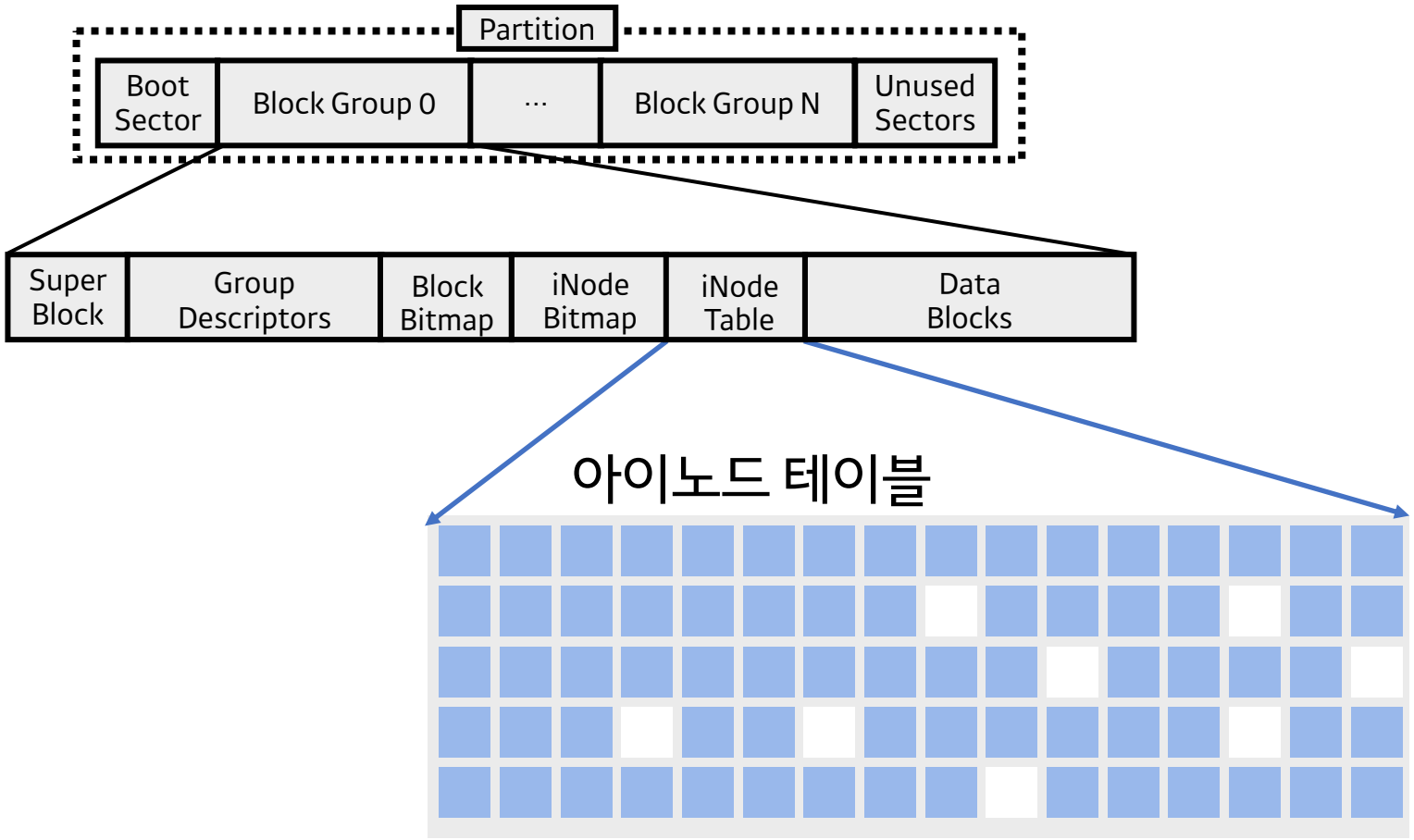


주요 용어 및 개념

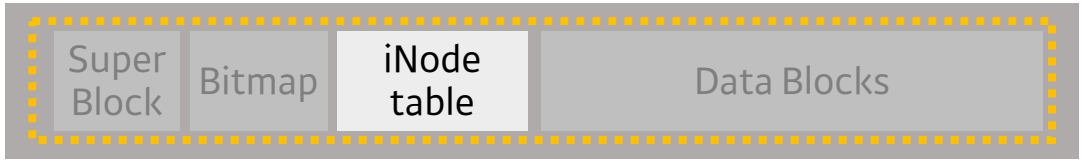


아이노드 테이블

- 아이노드 테이블

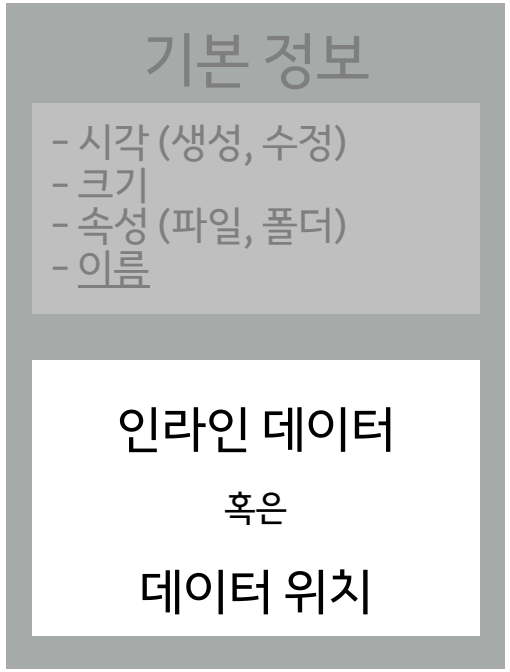


주요 용어 및 개념

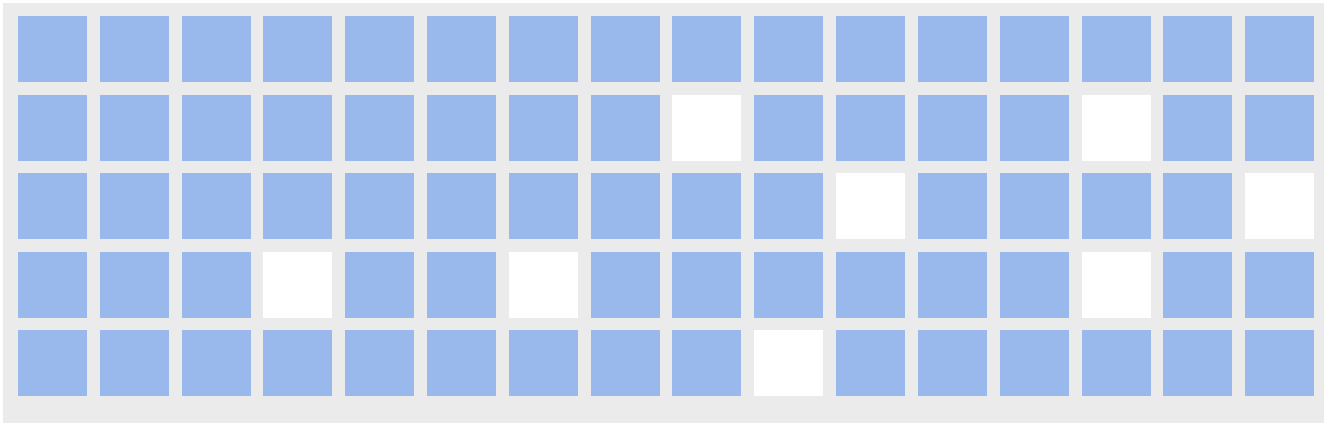


아이노드 테이블

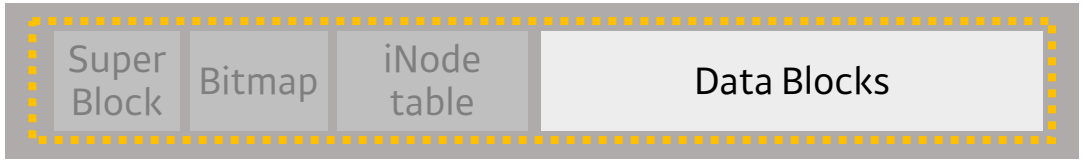
- 아이노드 테이블
- 아이노드 번호를 어떻게 알 수 있을까?



아이노드 테이블



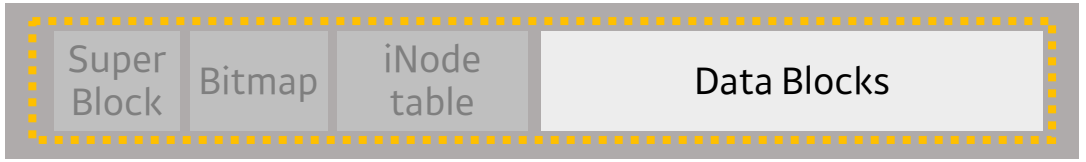
주요 용어 및 개념



데이터 블록

- 실제 데이터가 존재하는 영역
  - 파일 데이터

주요 용어 및 개념

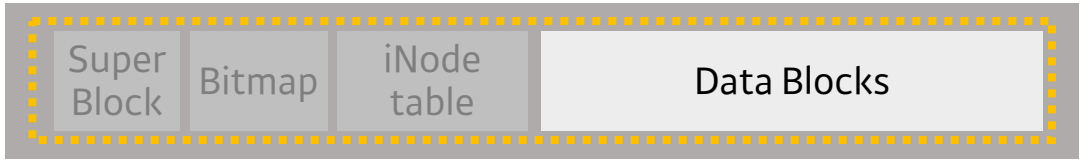


데이터 블록

- 실제 데이터가 존재하는 영역
  - 파일 데이터
  - 디렉토리 엔트리 정보

    (아이노드 번호 + 파일 메타 데이터)

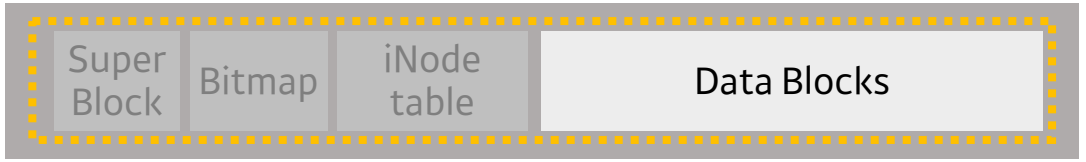
주요 용어 및 개념



데이터 블록

- 실제 데이터가 존재하는 영역
  - 파일 데이터
  - 디렉토리 엔트리 정보
  - 저널 데이터

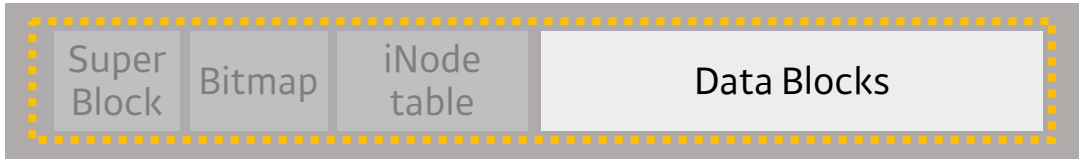
주요 용어 및 개념



데이터 블록

- 실제 데이터가 존재하는 영역
  - 파일 데이터
  - 디렉토리 엔트리 정보
  - 저널 데이터
  - 비할당 영역

주요 용어 및 개념

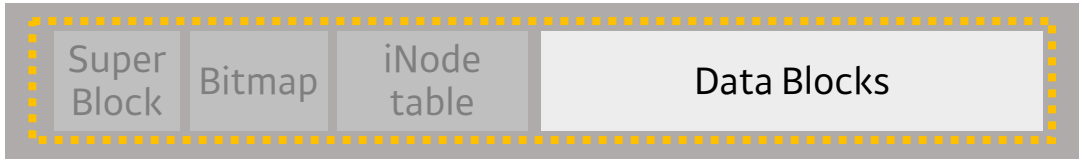


데이터 블록

- 실제 데이터가 존재하는 영역
  - 파일 데이터
  - 디렉토리 엔트리 정보
  - 저널 데이터
  - 비할당 영역

    [ff d8 ff ⋯ ff d9]

주요 용어 및 개념



데이터 블록

- 실제 데이터가 존재하는 영역
  - 파일 데이터
  - 디렉토리 엔트리 정보
  - 저널 데이터
  - 비할당 영역

    [ff d8 ff ⋯ ff d9]

    디렉토리 엔트리 카빙 어디서?

주요 용어 및 개념

저널

- 파일시스템의 변경을 저장하고 있는 파일시스템의 영역
  - 메타 데이터 or 파일 데이터
  - 환형 큐(circular queue)
  - ext3/4, hfs+, ntfs, f2fs, apfs(⇔fat)

주요 용어 및 개념

저널

- 파일시스템의 변경을 저장하고 있는 파일시스템의 영역
- 파일시스템 장애 복원을 위한 정보

주요 용어 및 개념

저널

- 파일시스템의 변경을 저장하고 있는 파일시스템의 영역
- 파일시스템 장애 복원을 위한 정보
- **파일시스템 성능 향상** (improves write performance on disk by turning random I/O into sequential I/O)

주요 용어 및 개념

저널

- 파일시스템의 변경을 저장하고 있는 파일시스템의 영역
- 파일시스템 장애 복원을 위한 정보
- 파일시스템 성능 향상
- 복호화 키 획득

주요 용어 및 개념

Soft (symbolic) Link

- 기존에 존재하는 파일 및 디렉토리에 대한 포인터

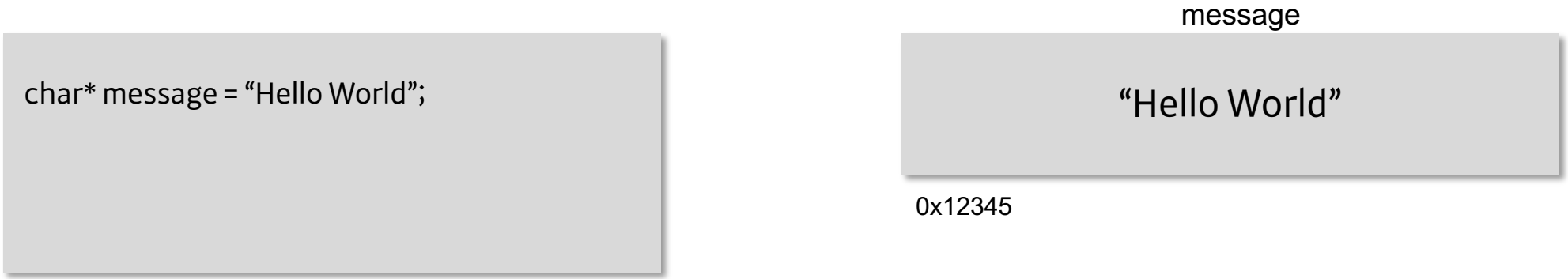
  원본 파일에 대한 경로만 존재

주요 용어 및 개념

Soft (symbolic) Link

- 기존에 존재하는 파일 및 디렉토리에 대한 포인터

  원본 파일에 대한 경로만 존재



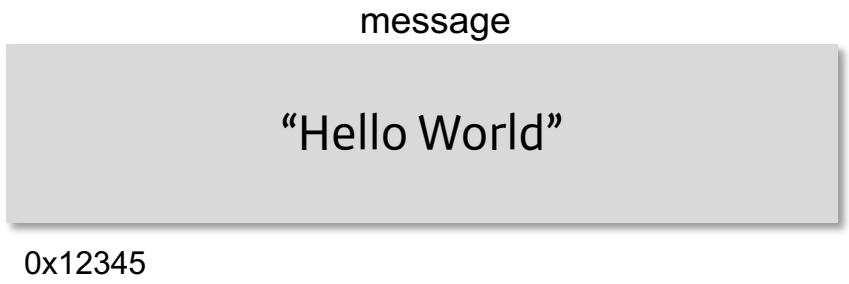
주요 용어 및 개념

Soft (symbolic) Link

- 기존에 존재하는 파일 및 디렉토리에 대한 포인터

  원본 파일에 대한 경로만 존재

```
char* message = “Hello World”;
char* link = message;
```



주요 용어 및 개념

Soft (symbolic) Link

- 기존에 존재하는 파일 및 디렉토리에 대한 포인터

  원본 파일에 대한 경로만 존재

```
char* message = “Hello World”;
char* link = message;
```

주요 용어 및 개념

Soft (symbolic) Link

- 기존에 존재하는 파일 및 디렉토리에 대한 포인터

  원본 파일에 대한 경로만 존재

```
char* message = “Hello World”;
char* link = message;
```

주요 용어 및 개념

Soft (symbolic) Link

- 기존에 존재하는 파일 및 디렉토리에 대한 포인터

  원본 파일에 대한 경로만 존재

```
char* message = “Hello World”;
char* link = message;
delete message;
```

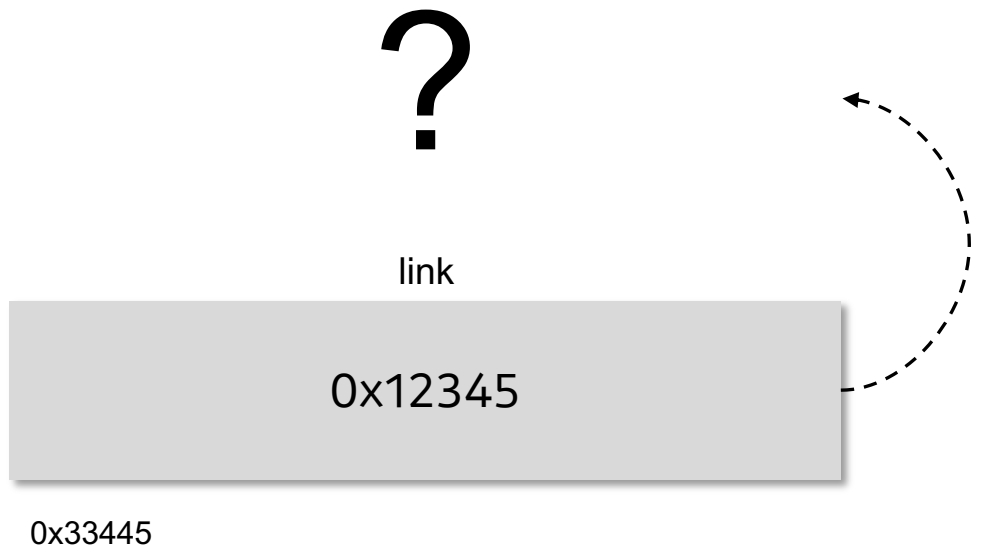
주요 용어 및 개념

Soft (symbolic) Link

- 기존에 존재하는 파일 및 디렉토리에 대한 포인터

  원본 파일에 대한 경로만 존재

```
char* message = “Hello World”;
char* link = message;
delete message;
```



주요 용어 및 개념

Soft (symbolic) Link

- 기존에 존재하는 파일 및 디렉토리에 대한 포인터
- 독립된 iNode 번호

```
$ cat source.file
Welcome to OSTechNix
```

```
$ ln -s source.file softlink.file
```

```
$ ls -lia
```

```
total 12
11665675 drwxrwxr-x  2 sk sk 4096 Oct 17 11:39 .
 4325378 drwxr-xr-x 37 sk sk 4096 Oct 17 11:39 ..
11665731 lrwxrwxrwx  1 sk sk   11 Oct 17 11:39 softlink.file -> source.file
11665692 -rw-rw-r--  1 sk sk   21 Oct 17 11:39 source.file
```

주요 용어 및 개념

Soft (symbolic) Link

- 기존에 존재하는 파일 및 디렉토리에 대한 포인터
- 독립된 iNode 번호
- 파일시스템에 걸쳐 생성 가능
- 원본 파일에 대한 경로만 가지고 있음

주요 용어 및 개념

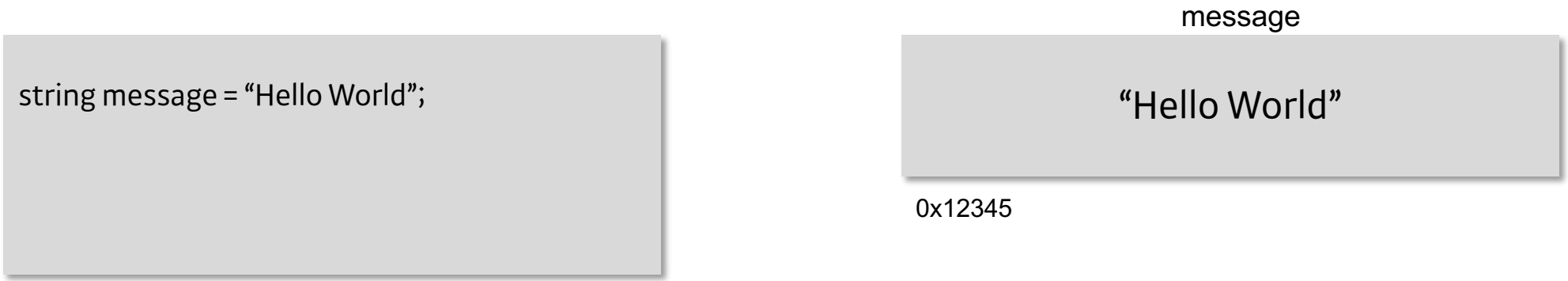
Hard Link

- 기존에 존재하는 파일에 대한 alias

주요 용어 및 개념

Hard Link

- 기존에 존재하는 파일에 대한 alias



주요 용어 및 개념

Hard Link

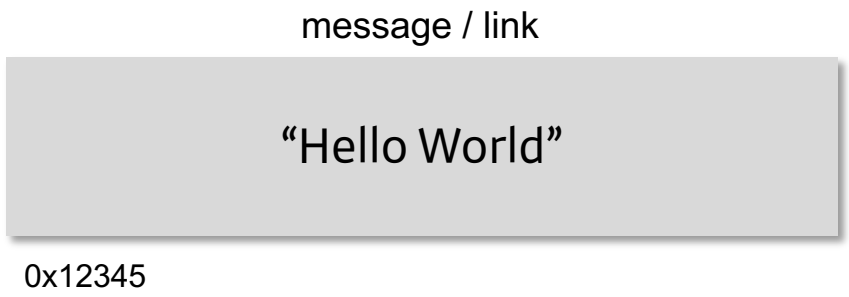
- 기존에 존재하는 파일에 대한 alias

주요 용어 및 개념

Hard Link

- 기존에 존재하는 파일에 대한 alias

```
string& link = message;
```



주요 용어 및 개념

Hard Link

- 기존에 존재하는 파일에 대한 alias
- iNode 공유, 참조 카운트

```
$ ln source.file hardlink.file
```

```
$ ls -lia
```

```
total 16
11665675 drwxrwxr-x 2 sk sk 4096 Oct 17 11:58 .
4325378 drwxr-xr-x 37 sk sk 4096 Oct 17 11:39 ..
11665692 -rw-rw-r-- 2 sk sk 21 Oct 17 11:57 hardlink.file
11665692 -rw-rw-r-- 2 sk sk 21 Oct 17 11:57 source.file
```

주요 용어 및 개념

Hard Link

- 기존에 존재하는 파일에 대한 alias
- iNode 공유, 참조 카운트

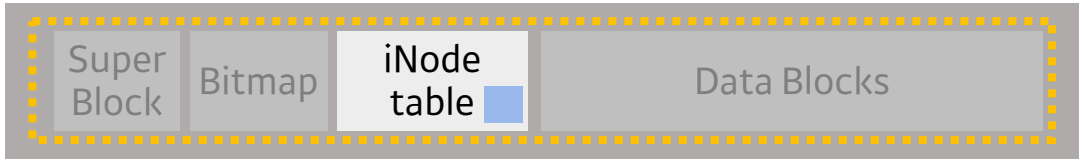
```
$ ln source.file hardlink.file
```

```
$ ls -lia
```

```
total 16
11665675 drwxrwxr-x 2 sk sk 4096 Oct 17 11:58 .
4325378 drwxr-xr-x 37 sk sk 4096 Oct 17 11:39 ..
11665692 -rw-rw-r-- 2 sk sk 21 Oct 17 11:57 hardlink.file
11665692 -rw-rw-r-- 2 sk sk 21 Oct 17 11:57 source.file
```

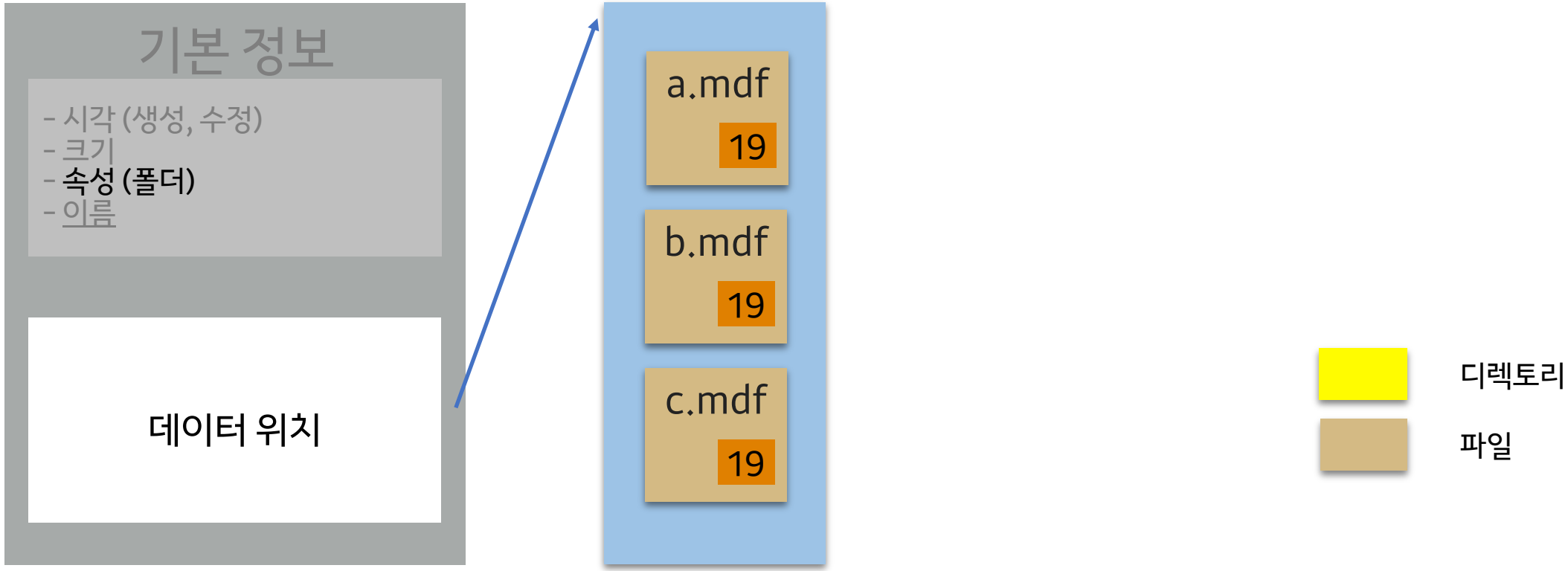
Hard Link는 어떻게 표현되는가?

주요 용어 및 개념



Hard Link

- 서로 다른 디렉토리 엔트리가 동일 inode 번호 공유



주요 용어 및 개념

Hard Link

- 기존에 존재하는 파일에 대한 alias
- iNode 공유, 참조 카운트
- 단일 파일시스템 내에서만 가능

주요 용어 및 개념

Special file

- 일반 파일로 간주된 다양한 외부 장치 인터페이스

  printer, tape drive, named pipe

- stream interface

  open, seek, read, write, close

- char / block device

주요 용어 및 개념

Pipe

- 두 개 이상의 프로세스가 입출력을 연결하여 데이터를 주고 받을 수 있게 만든 파일
- ls -al | wc -l

주요 용어 및 개념

Socket

- 두 개 이상의 네트워크 프로세스의 입출력을 파일로 표현

주요 용어 및 개념

Mount

- 디바이스, 파일, 원격 노드 등을 파일시스템의 특정 디렉토리에 위치

```
$ mount /dev/hda2 /media/PHOTOS
```

```
$ umount /dev/hda2
```

```
$ mount –t ext4 –o loop ./ext.bin /mnt/guest#
```

주요 용어 및 개념

man

- 유닉스-like 운영체제의 커맨드에 대한 help
- man man
- RTFM

파일시스템 복원

파일시스템 복원

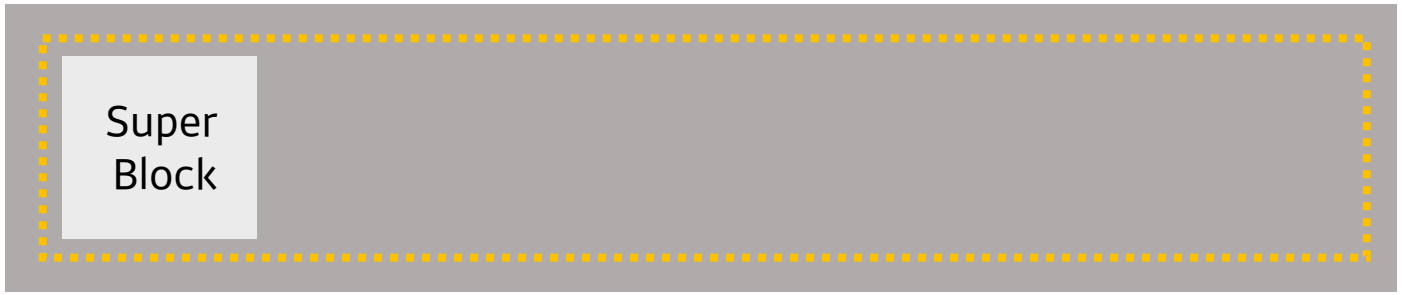
파일시스템 복원

- 획득 이미지에서 활성과 삭제 영역을 식별
- 파일시스템 복호화

파일시스템 복원

파일시스템 복원 절차

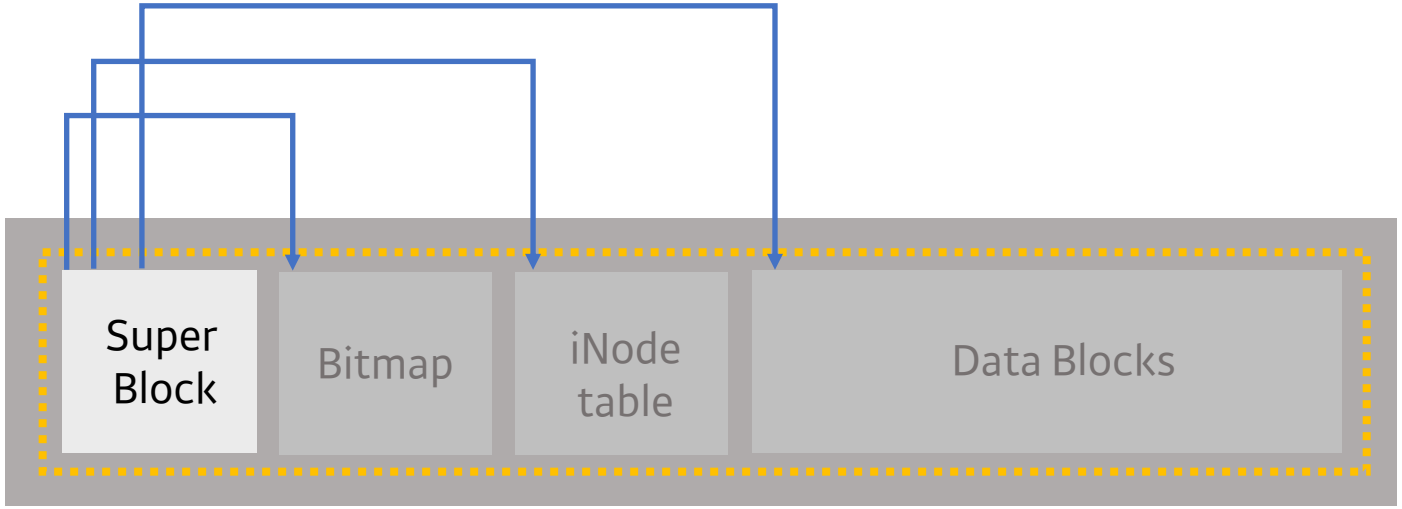
- 슈퍼 블록에서 F/S geometry 식별



파일시스템 복원

파일시스템 복원 절차

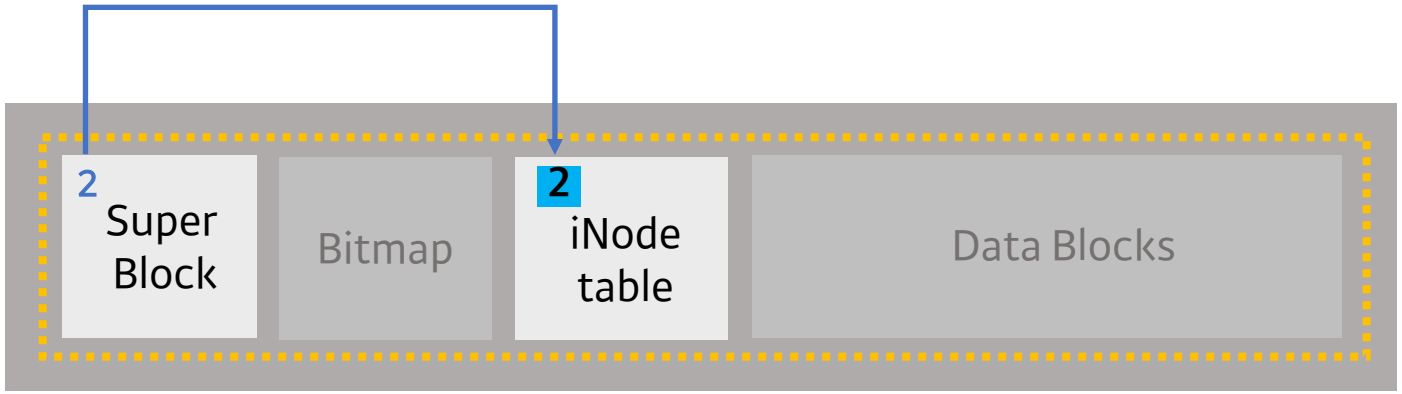
- 슈퍼 블록에서 F/S geometry 식별



파일시스템 복원

파일시스템 복원 절차

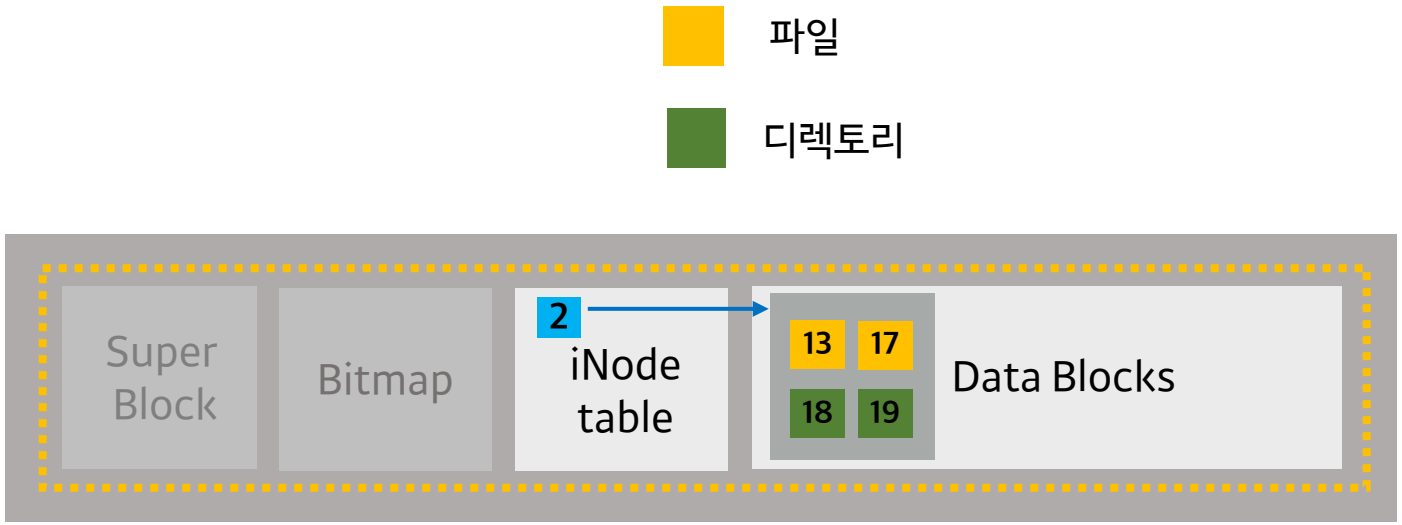
- 슈퍼 블록에서 F/S geometry 식별
- 루트 iNode → 정보 가져오기 iNode table에서 읽기



파일시스템 복원

파일시스템 복원 절차

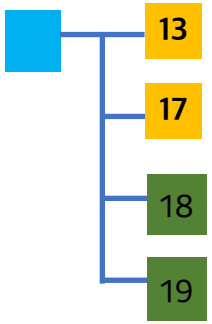
- 슈퍼 블록에서 F/S geometry 식별
- 루트 iNode → 정보 가져오기 iNode table에서 읽기
- 루트 iNode의 자식을 읽음

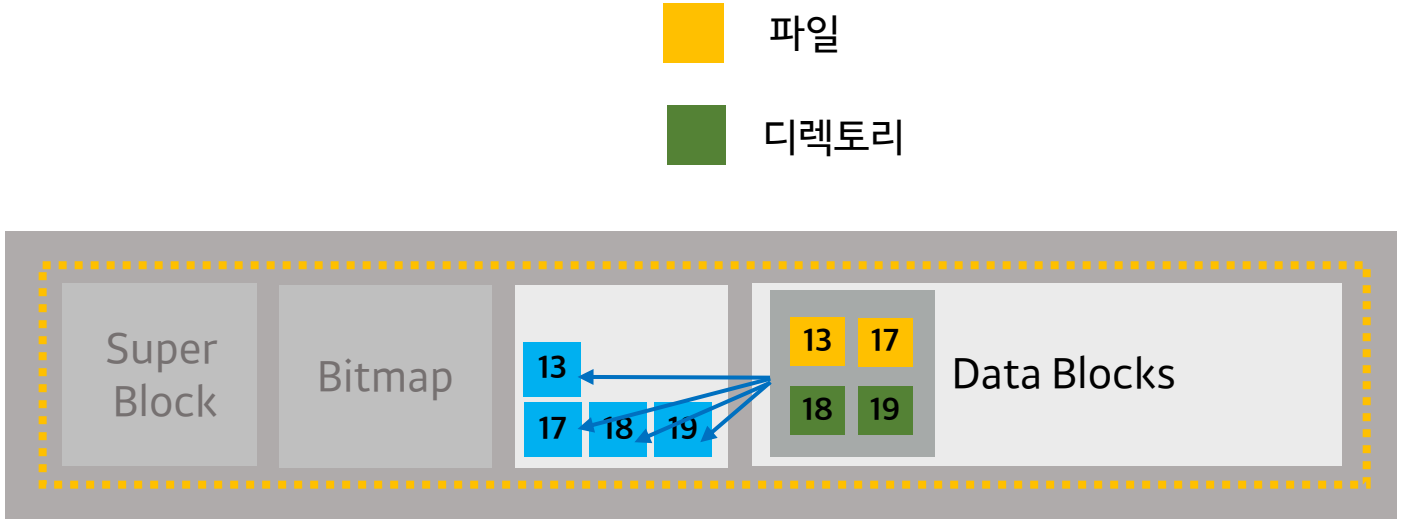


파일시스템 복원

파일시스템 복원 절차

- 슈퍼 블록에서 F/S geometry 식별
- 루트 iNode → 정보 가져오기 iNode table에서 읽기
- 루트 iNode의 자식을 읽음

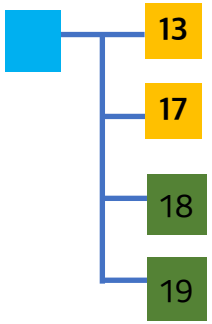


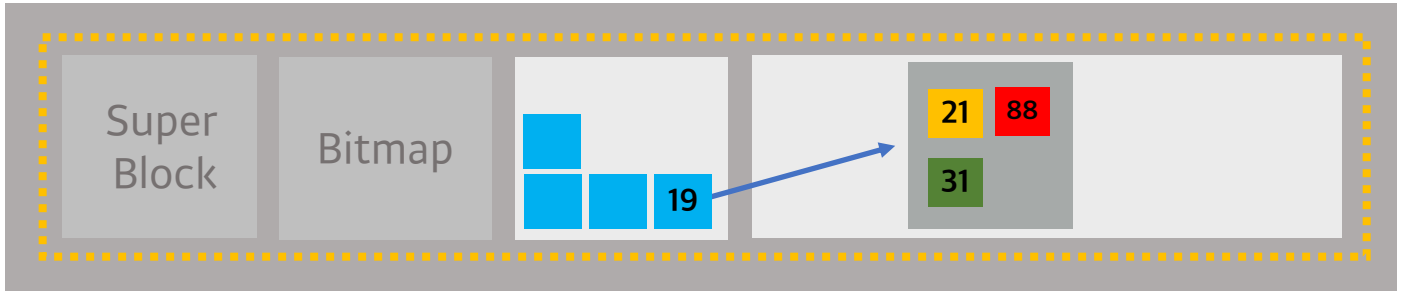


파일시스템 복원

파일시스템 복원 절차

- 슈퍼 블록에서 F/S geometry 식별
- 루트 iNode → 정보 가져오기 iNode table에서 읽기
- 루트 iNode의 자식을 읽음
- 자식 디렉토리의 iNode를 읽음

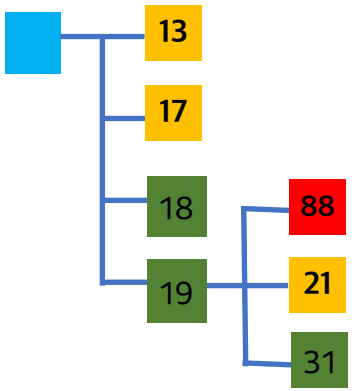


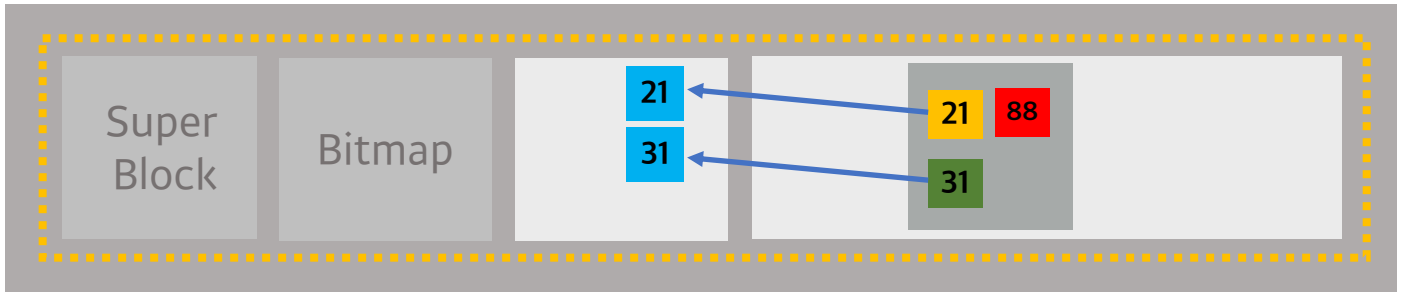


파일시스템 복원

파일시스템 복원 절차

- 슈퍼 블록에서 F/S geometry 식별
- 루트 iNode → 정보 가져오기 iNode table에서 읽기
- 루트 iNode의 자식을 읽음
- 자식 디렉토리의 iNode를 읽음

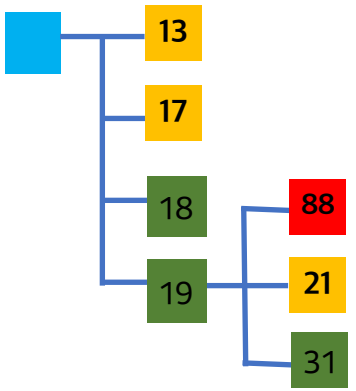


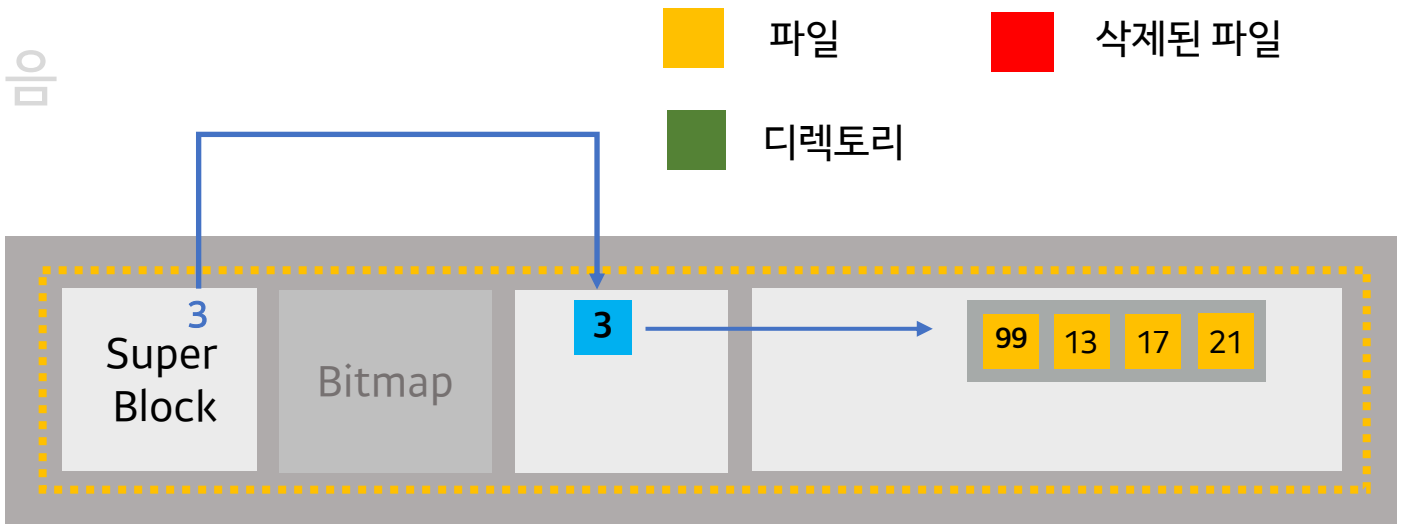


파일시스템 복원

파일시스템 복원 절차

- 슈퍼 블록에서 F/S geometry 식별
- 루트 iNode → 정보 가져오기 iNode table에서 읽기
- 루트 iNode의 자식을 읽음
- 자식 디렉토리의 iNode를 읽음
- 저널을 읽음

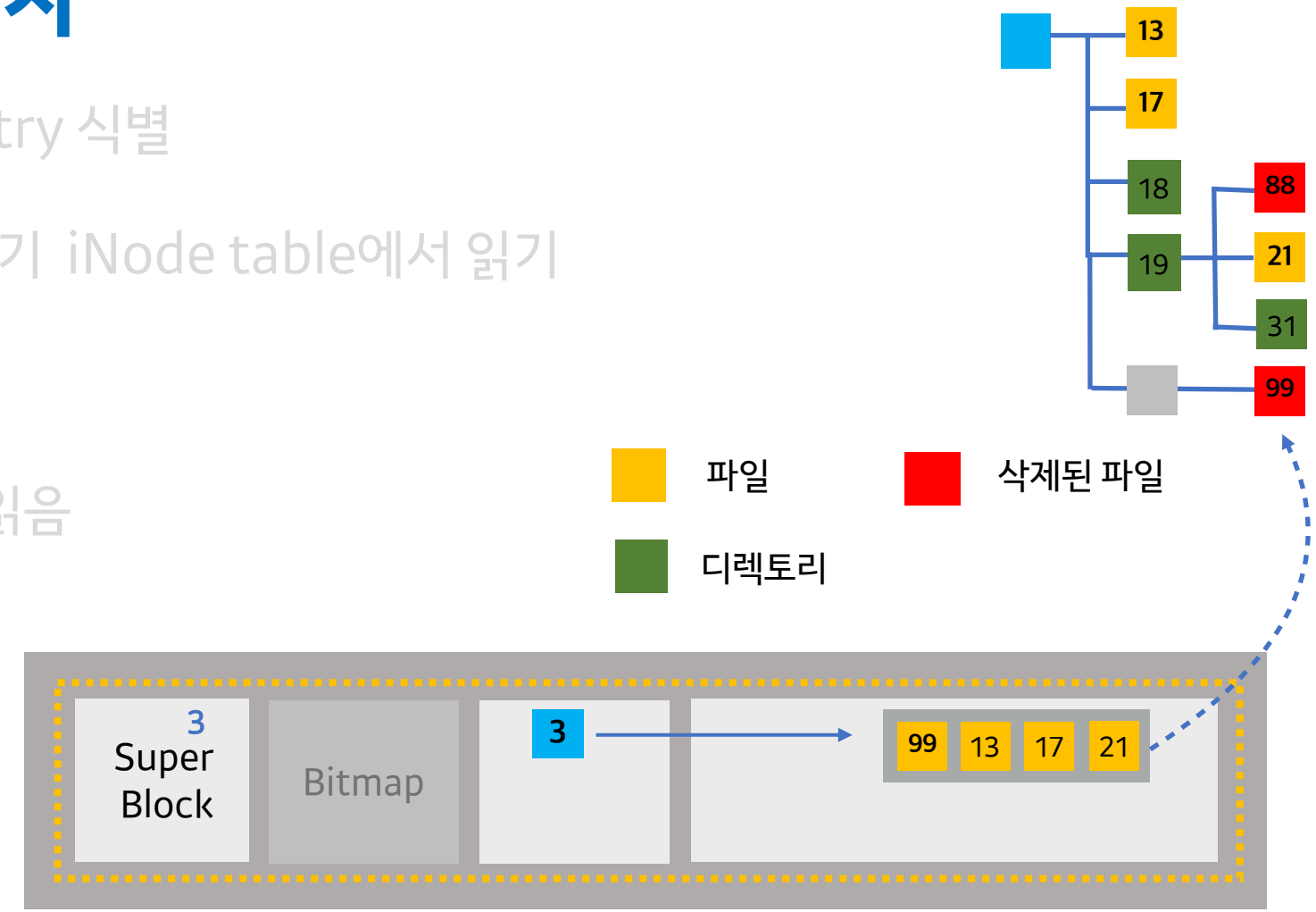




파일시스템 복원

파일시스템 복원 절차

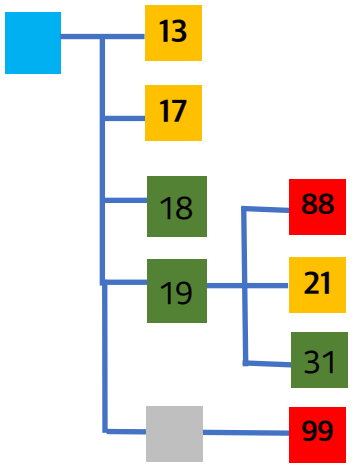
- 슈퍼 블록에서 F/S geometry 식별
- 루트 iNode → 정보 가져오기 iNode table에서 읽기
- 루트 iNode의 자식을 읽음
- 자식 디렉토리의 iNode를 읽음
- 저널을 읽음

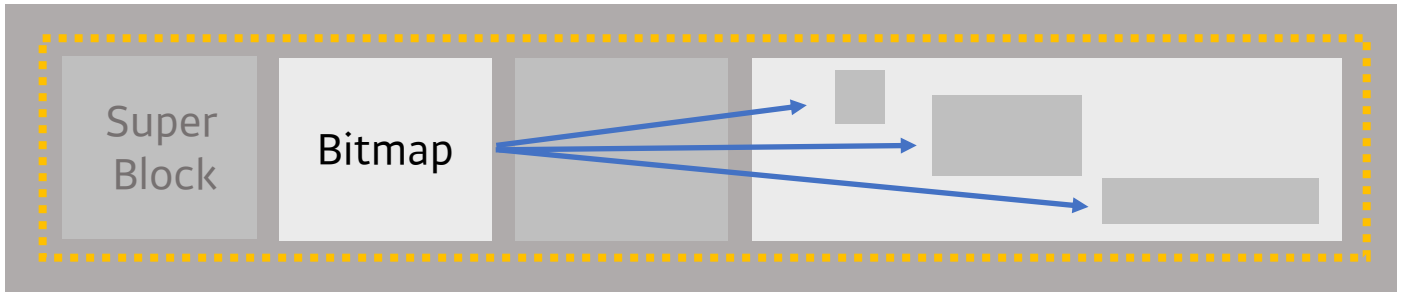


파일시스템 복원

파일시스템 복원 절차

- 슈퍼 블록에서 F/S geometry 식별
- 루트 iNode → 정보 가져오기 iNode table에서 읽기
- 루트 iNode의 자식을 읽음
- 자식 디렉토리의 iNode를 읽음
- 저널을 읽음
- 비트맵을 읽음

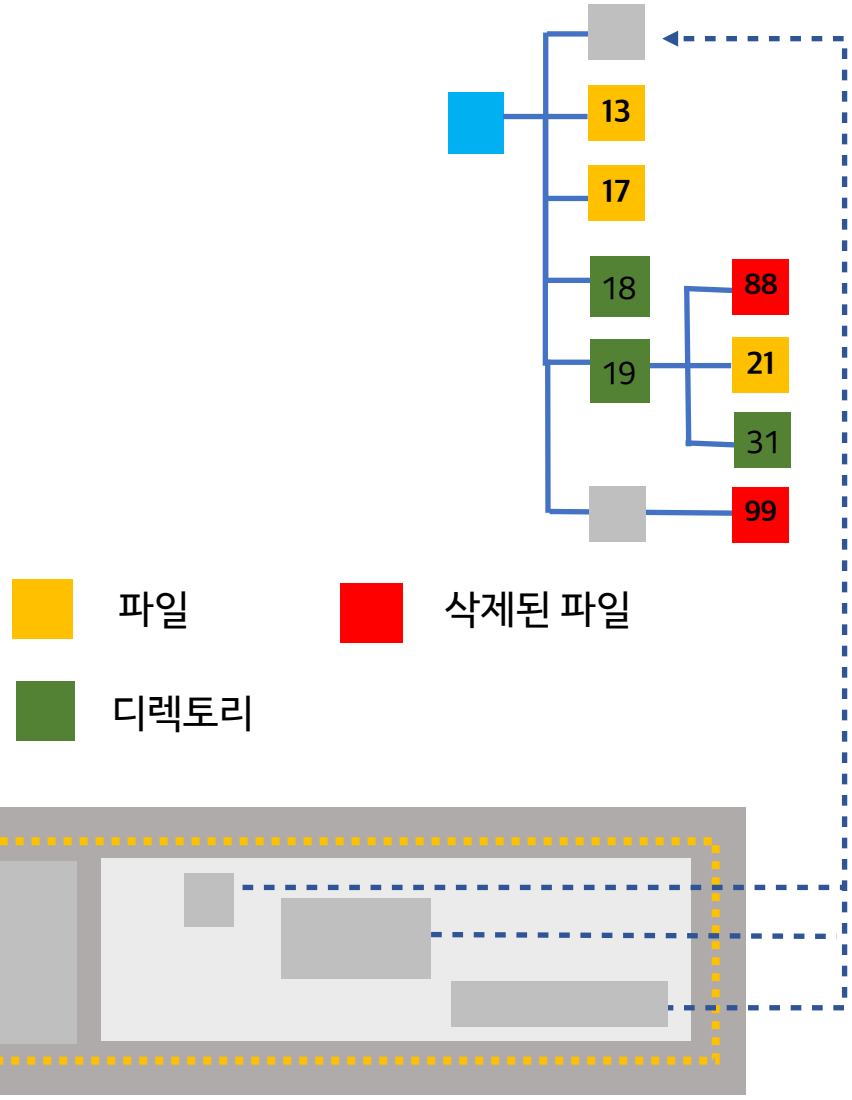




파일시스템 복원

파일시스템 복원 절차

- 슈퍼 블록에서 F/S geometry 식별
- 루트 iNode → 정보 가져오기 iNode table에서 읽기
- 루트 iNode의 자식을 읽음
- 자식 디렉토리의 iNode를 읽음
- 저널을 읽음
- 비트맵을 읽음



파일시스템 복원

파일시스템 복원 절차

획득 시 파일시스템 메타 정보 일부가 손상되면

어떻게 활성 / 비할당 영역을 식별할 수 있을까요?

파일시스템 복원

파일시스템 복원 절차

획득 시 파일시스템 메타 정보 일부가 손상되면

비할당 영역 = 전체 - 활성 영역

모바일 파일시스템

모바일 파일시스템

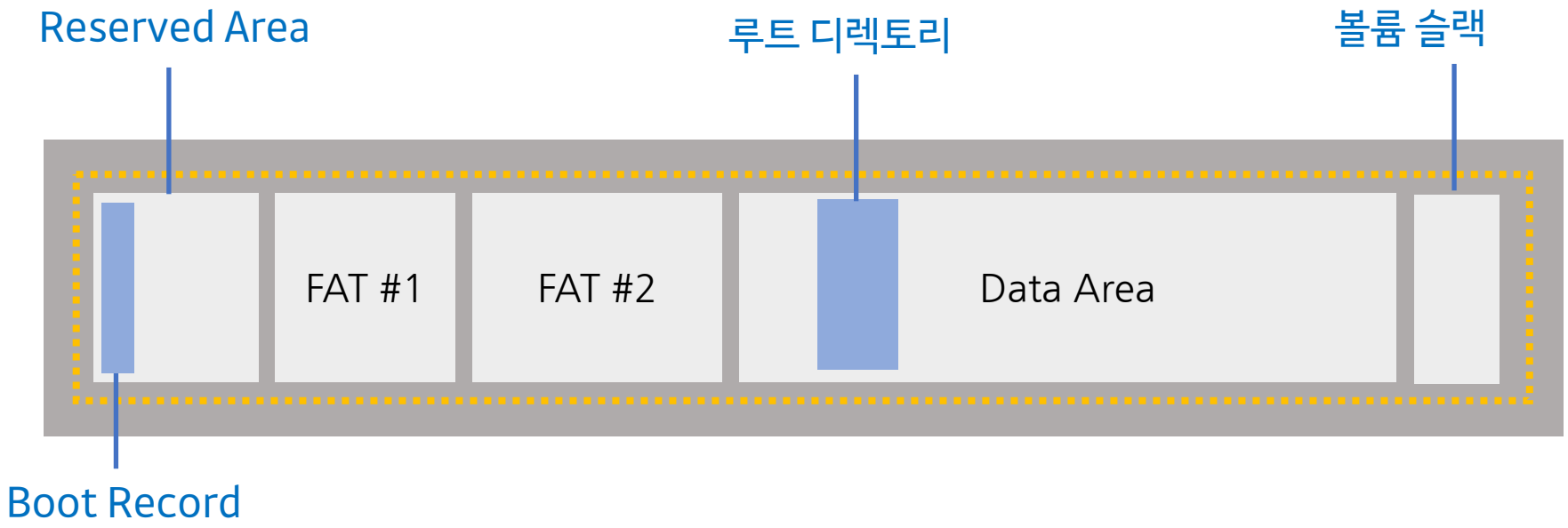
FAT 12/16/32 - 파일시스템 복원

- SD 카드에서 사용, TRIM이 적용되지 않음

모바일 파일시스템

FAT 12/16/32 - 파일시스템 복원

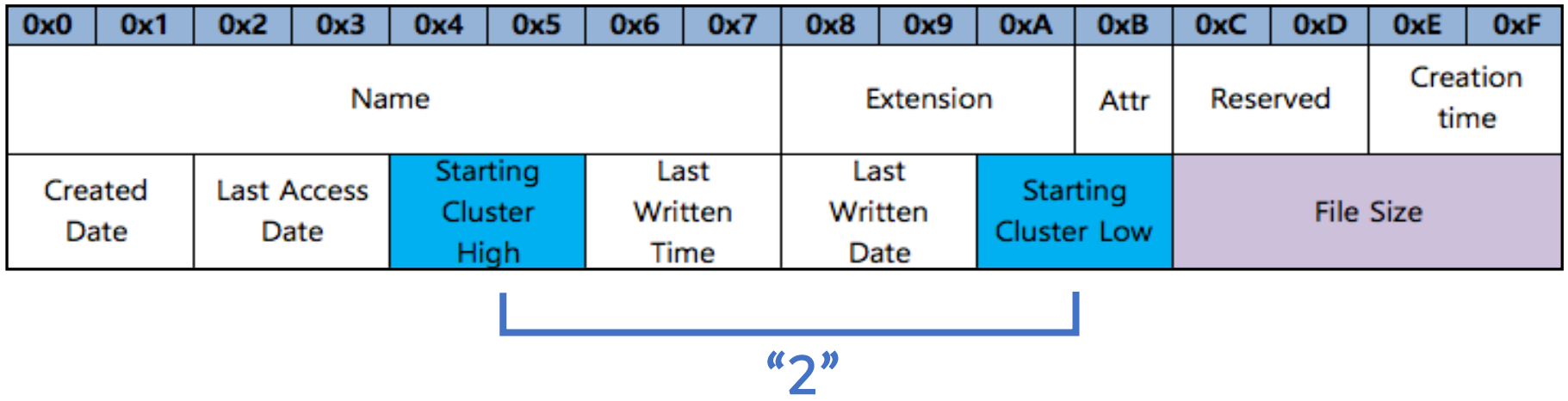
- SD 카드에서 사용, TRIM이 적용되지 않음
- Boot Record - Root 디렉토리 위치, 클러스터 크기, …



모바일 파일시스템

FAT 12/16/32 - 파일시스템 복원

- SD 카드에서 사용, TRIM이 적용되지 않음
- Boot Record
- 디렉토리 엔트리 – 메타 데이터, **데이터 위치**



모바일 파일시스템

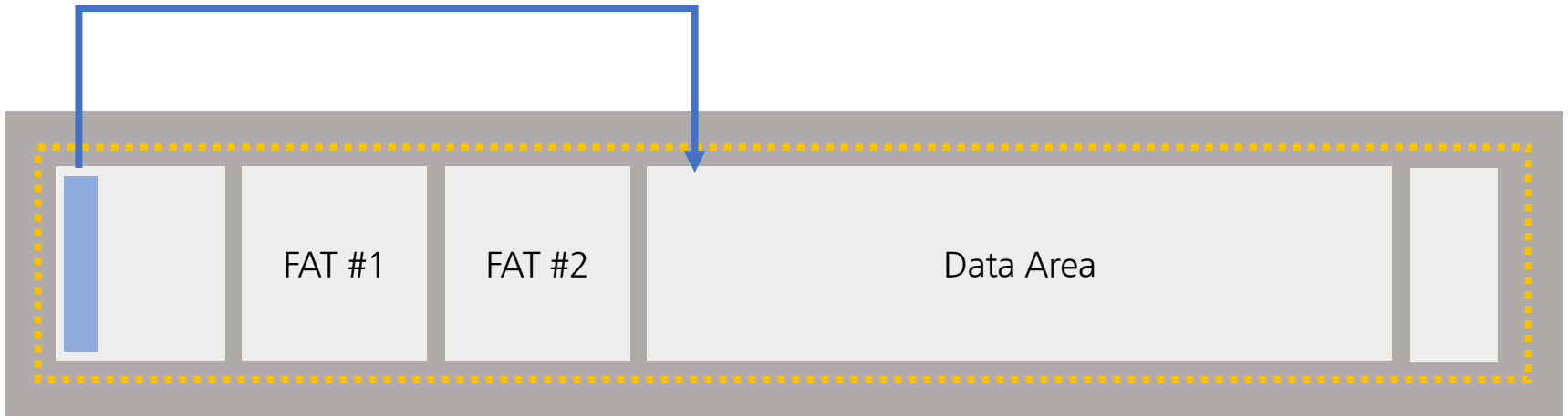
FAT 12/16/32 - 파일시스템 복원

- SD 카드에서 사용, TRIM이 적용되지 않음
- Boot Record
- 디렉토리 엔트리 - 메타 데이터, 데이터 위치
- FAT (File Allocation Table) - 실제 데이터의 위치를 연결 리스트로 표현

모바일 파일시스템

FAT 12/16/32 - 파일시스템 복원

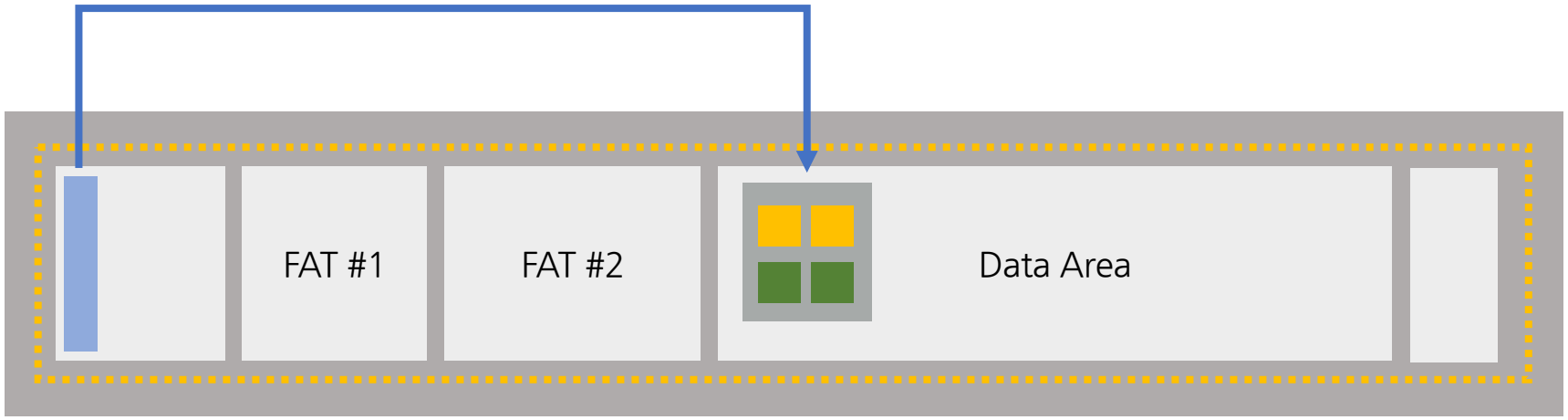
- 부트 레코드에서 루트 디렉토리 위치 찾기



모바일 파일시스템

FAT 12/16/32 - 파일시스템 복원

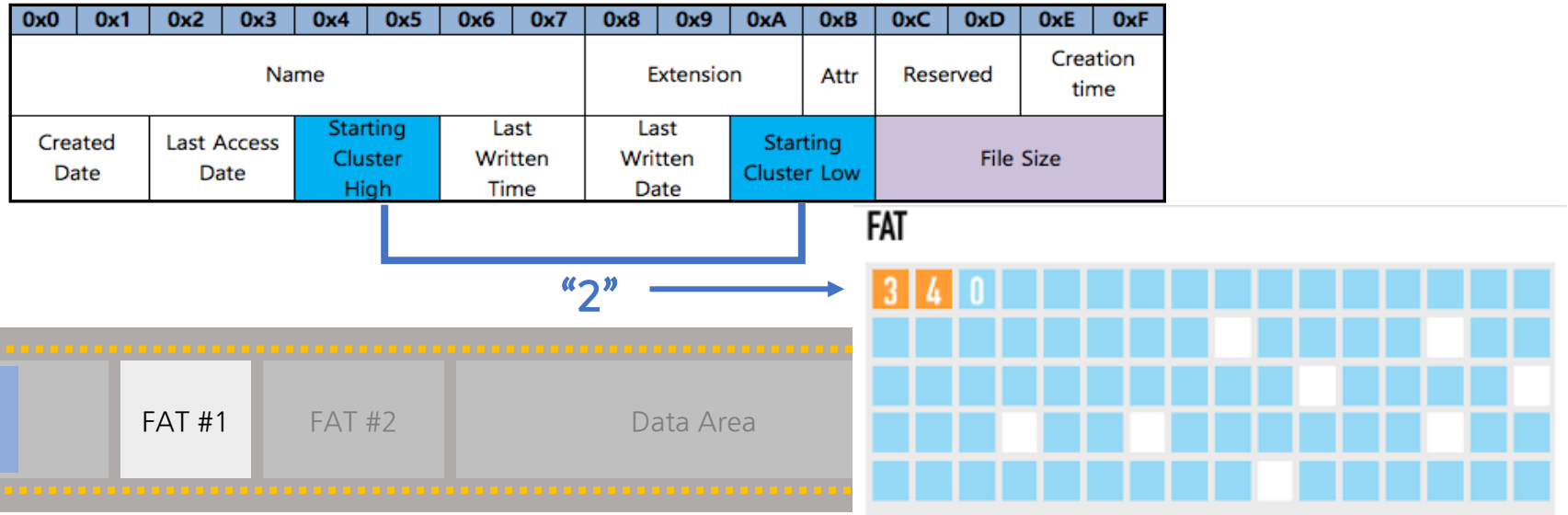
- 부트 레코드에서 루트 디렉토리 위치 찾기
- 루트 디렉토리에서 디렉토리 엔트리 읽기



모바일 파일시스템

FAT 12/16/32 - 파일시스템 복원

- 부트 레코드에서 루트 디렉토리 위치 찾기
- 루트 디렉토리에서 디렉토리 엔트리 읽기
- 클러스터 체인을 통해 현재 디렉토리 엔트리의 데이터 읽기



모바일 파일시스템

FAT 12/16/32 - 파일시스템 복원

- 부트 레코드에서 루트 디렉토리 위치 찾기
- 루트 디렉토리에서 디렉토리 엔트리 읽기
- 클러스터 체인을 통해 현재 디렉토리 엔트리의 데이터 읽기
- **파일**인 경우 - 체인의 번호에 해당하는 클러스터의 내용을 모아 컨텐츠 복원

  디렉토리인 경우 - 체인의 번호에 해당하는 클러스터의 내용을 모아 디렉토리 엔트리 목록 구성

  → 디렉토리 엔트리 목록 내의 각 엔트리에 대해 위 과정을 반복

모바일 파일시스템

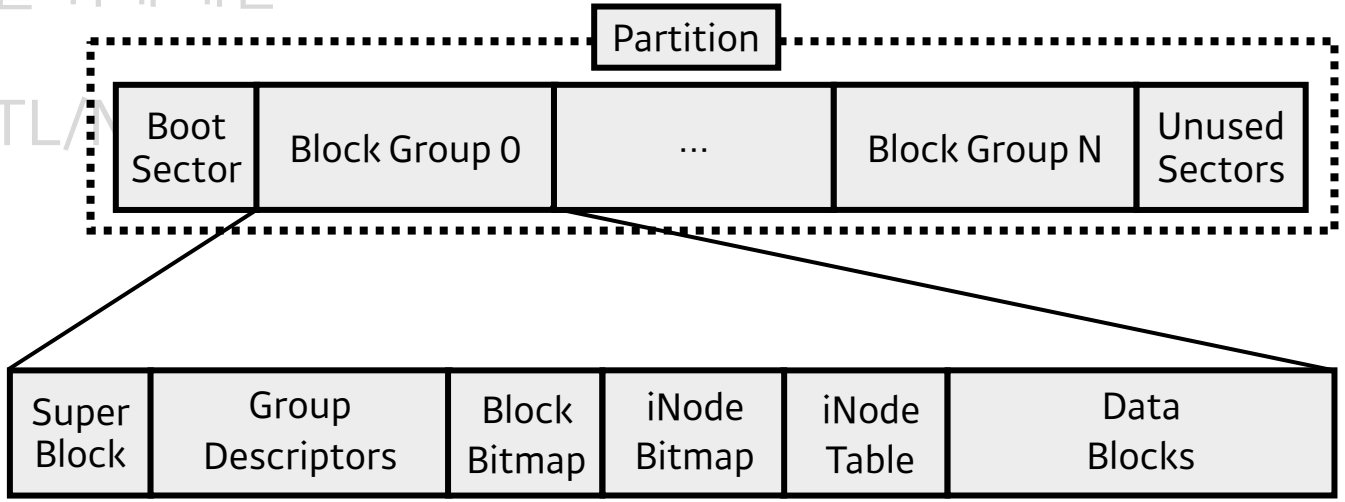
Ext3/4

- Extended File System
- Linux kernel을 위해 개발
- eMMC 혹은 FTL/NAND에서도 동작

모바일 파일시스템

Ext3/4

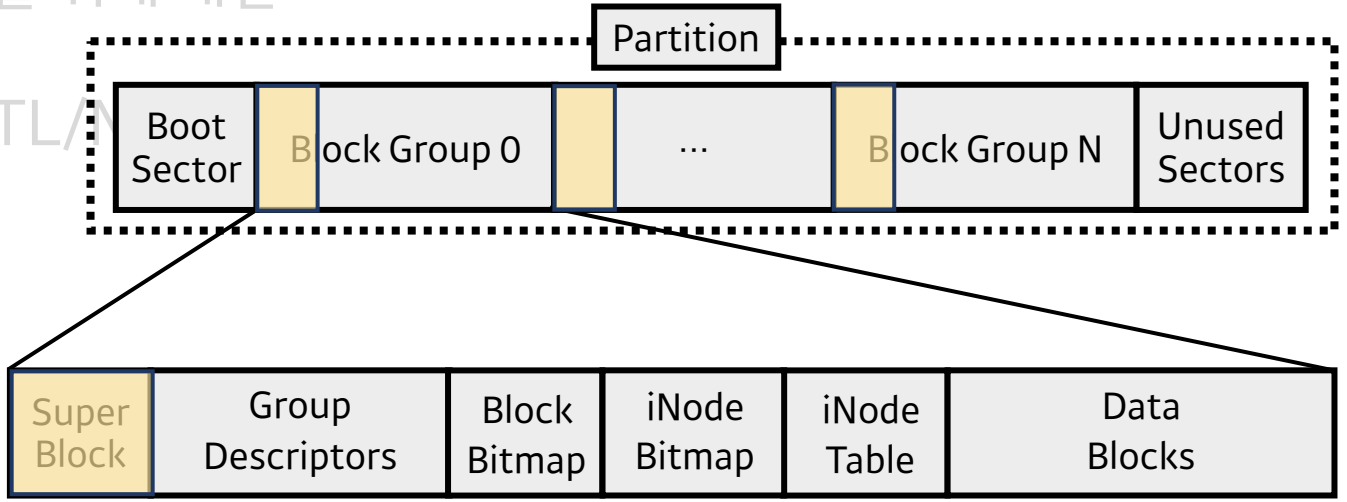
- Extended File System
- Linux kernel을 위해 개발
- eMMC 혹은 FTL/N



모바일 파일시스템

Ext3/4

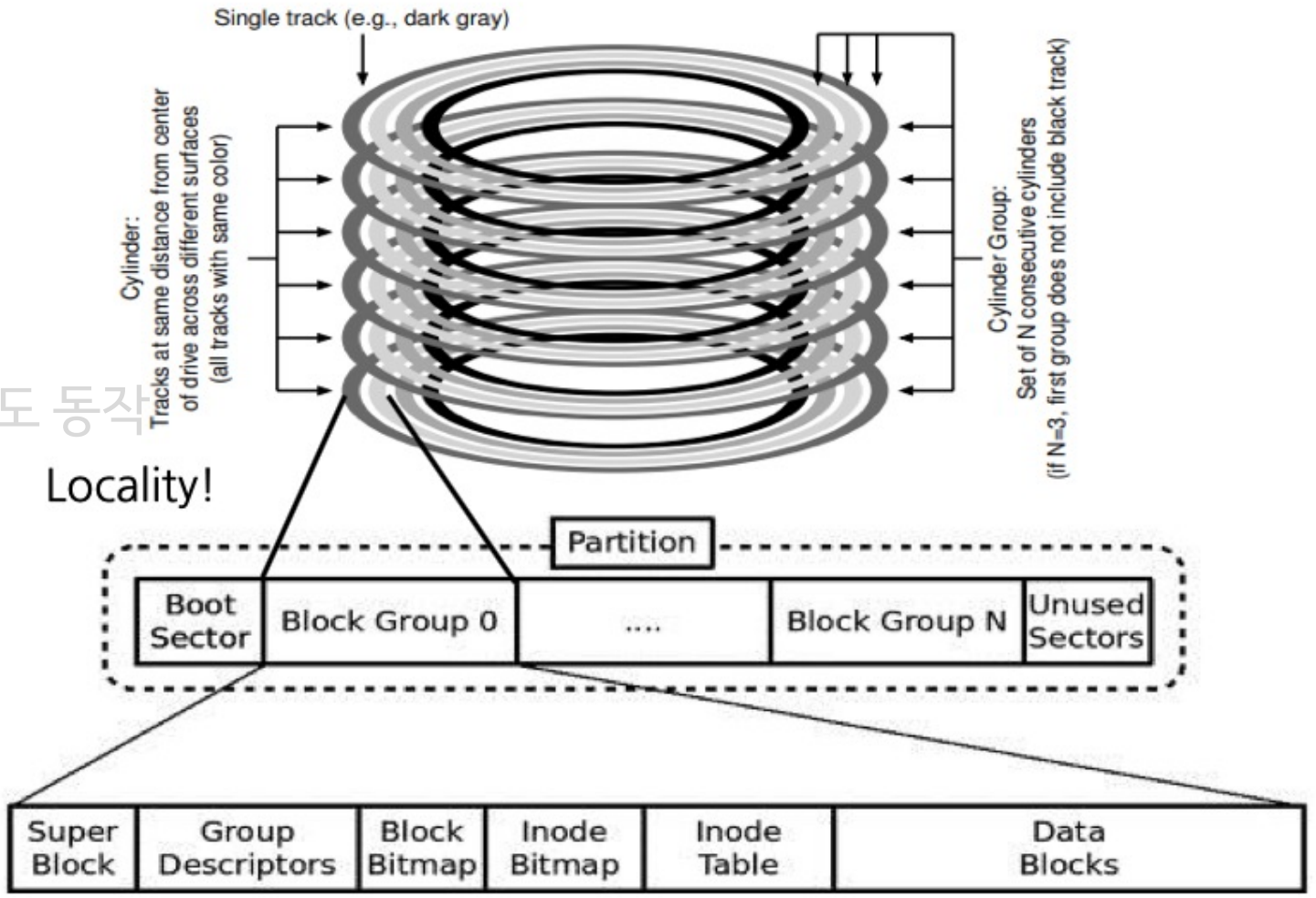
- Extended File System
- Linux kernel을 위해 개발
- eMMC 혹은 FTL/N



모바일 파일시스템

Ext3/4

- Extended File System
- Linux kernel을 위해 개발
- eMMC 혹은 FTL/NAND에서도 동작
- FFS (Fast File System)



모바일 파일시스템

Ext3

- 데이터 표현

  iNode에서 직접~3중 간접 포인터로 파일 내용 표현

- Ext3부터 저널 제공

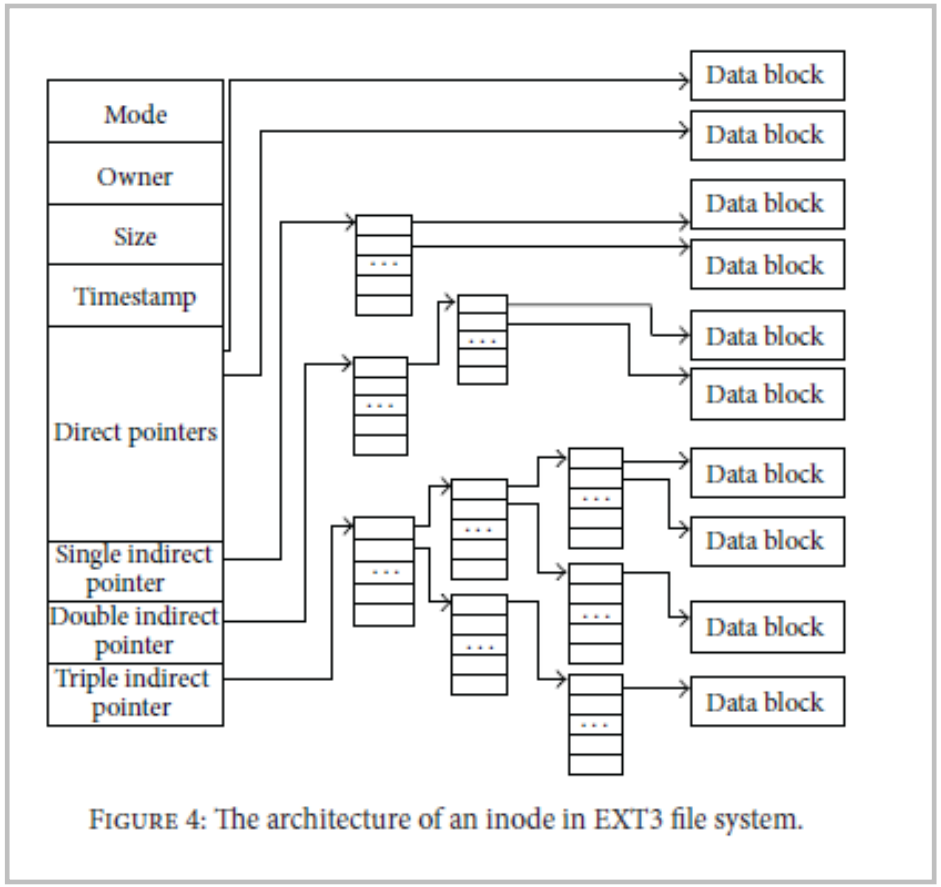


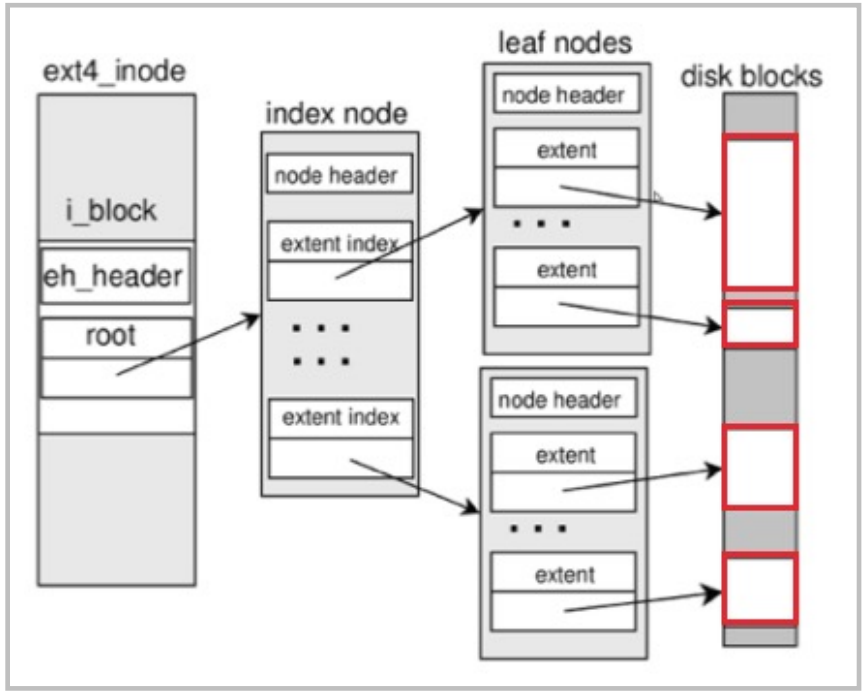
FIGURE 4: The architecture of an inode in EXT3 file system.

모바일 파일시스템

Ext4

- 데이터 표현

블록 포인터 대신 extent (start, size)로 데이터 표현

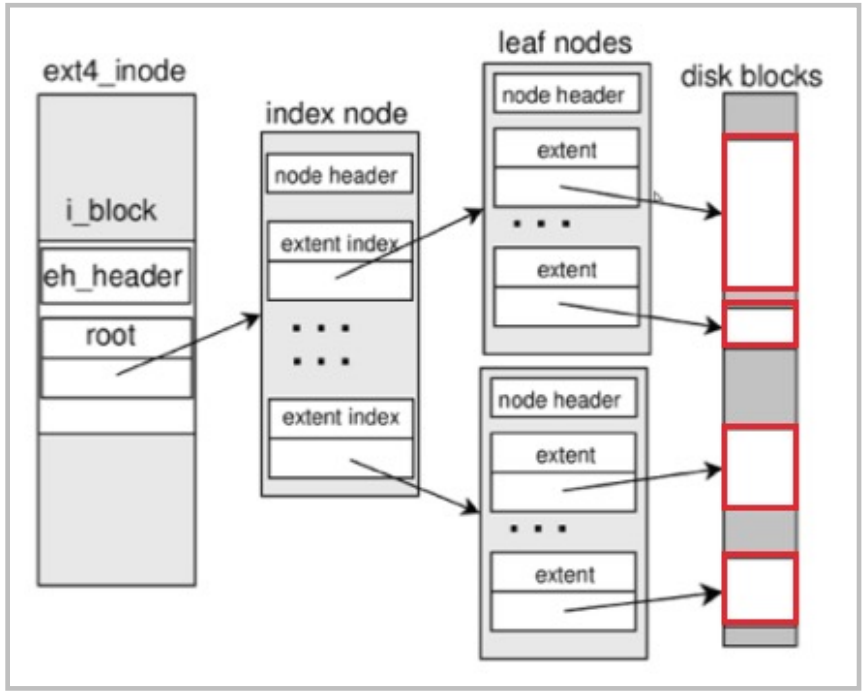


모바일 파일시스템

Ext4

- 데이터 표현

블록 포인터 대신 extent (start, size)로 데이터 표현 (hfs+, ntfs)



모바일 파일시스템

Ext4

- 데이터 표현

  블록 포인터 대신 extent (start, size)로 데이터 표현

- Sparse File 지원 (allocate-on-flush)

- 1 엑사 바이트 볼륨

  16 테라 바이트 파일

모바일 파일시스템

HFS+

- Hierarchical File System (HFS+, HFS/X)
- Apple사에서 macOS, iOS 사용을 위해 개발
- 파일시스템 암호화 지원 (FBE)

모바일 파일시스템

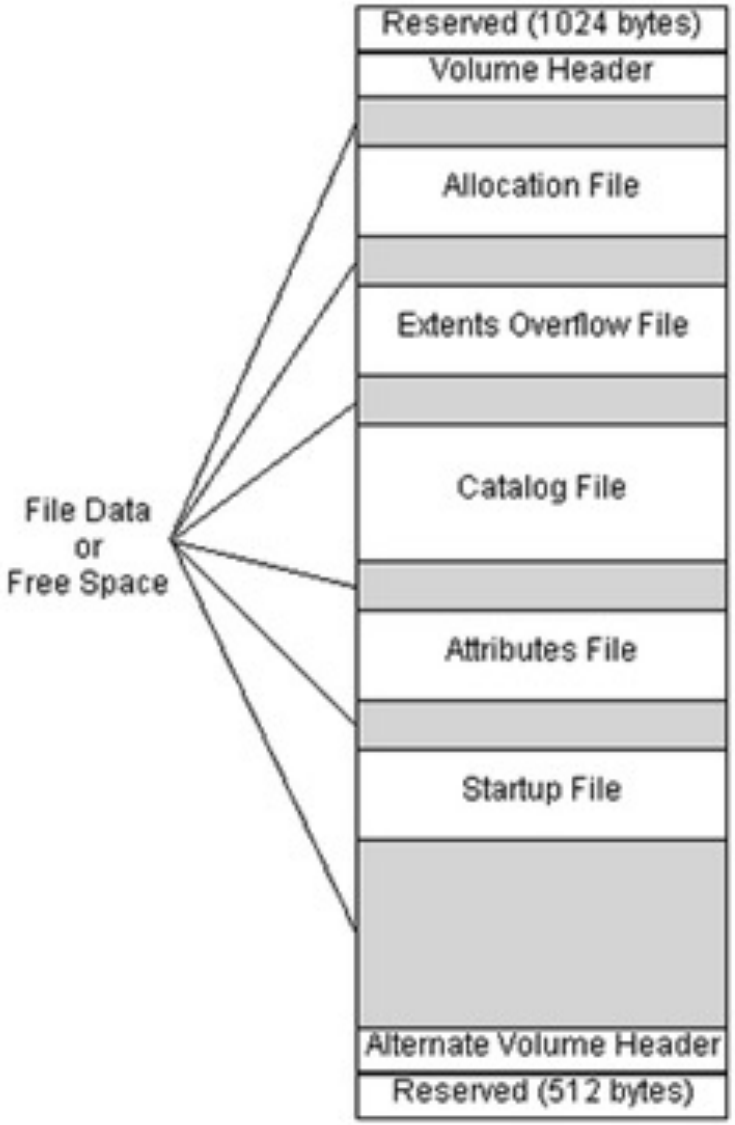
HFS+

- Hierarchical File System (HFS+, HFS/X)
- Apple사에서 macOS, iOS 사용을 위해 개발
- 파일시스템 암호화 지원 (FBE)
- 저널 제공 (8M / 100 G)
- 8 엑사 바이트 볼륨/파일

모바일 파일시스템

HFS+

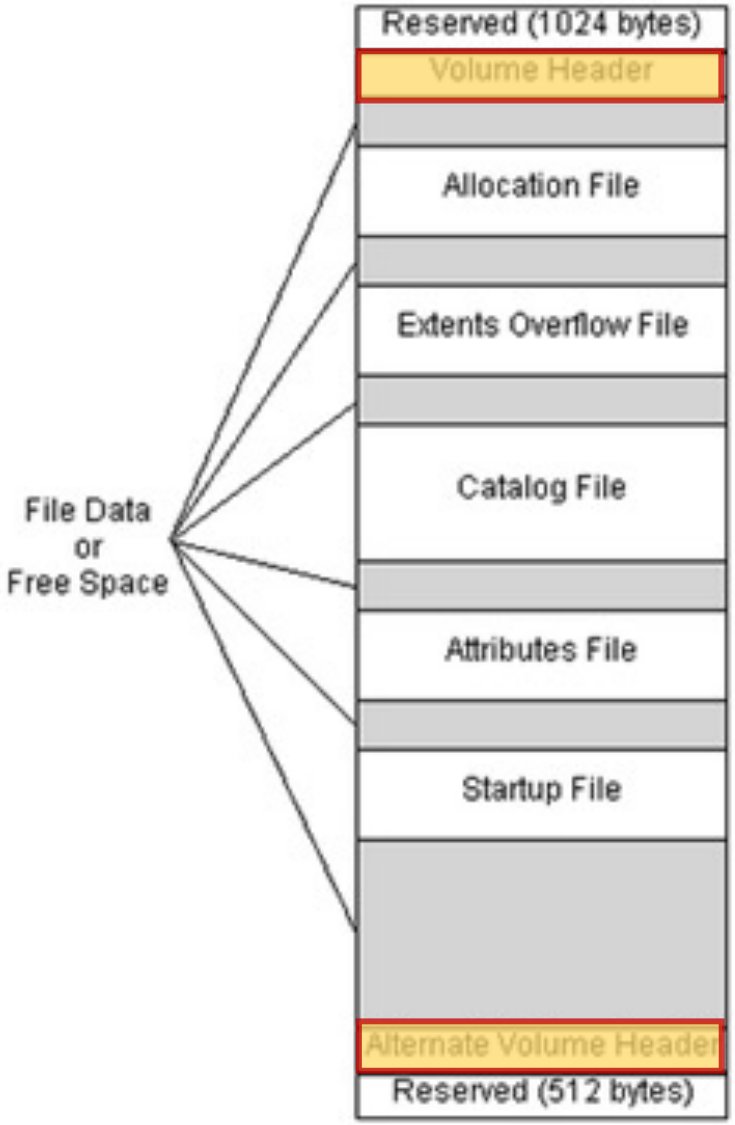
- Hierarchical File Sy
- Apple사에서 macOS
- 파일시스템 암호화 지
- 저널 제공 (8M / 100
- 8 엑사 바이트 볼륨/파



모바일 파일시스템

HFS+

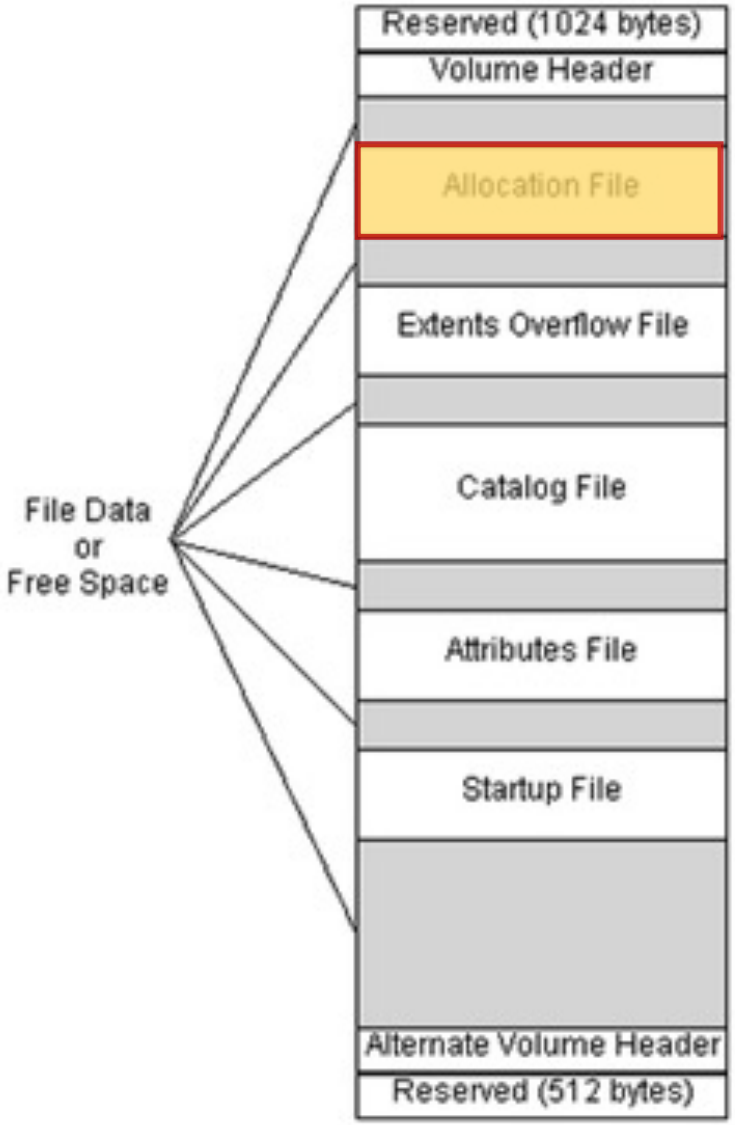
- Hierarchical File Sy
- Apple사에서 macOS
- 파일시스템 암호화 지
- 저널 제공 (8M / 100
- 8 엑사 바이트 볼륨/파



모바일 파일시스템

HFS+

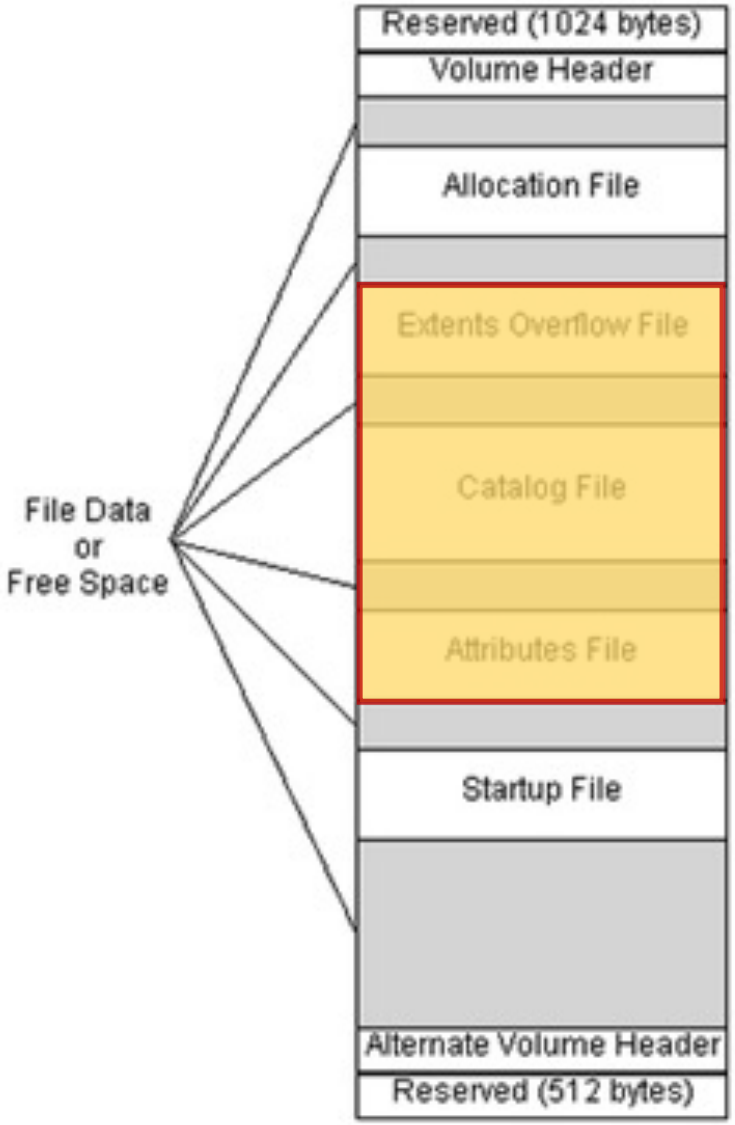
- Hierarchical File Sy
- Apple사에서 macOS
- 파일시스템 암호화 지
- 저널 제공 (8M / 100
- 8 엑사 바이트 볼륨/파



모바일 파일시스템

HFS+

- Hierarchical File Sy
- Apple사에서 macOS
- 파일시스템 암호화 지
- 저널 제공 (8M / 100
- 8 엑사 바이트 볼륨/파



모바일 파일시스템

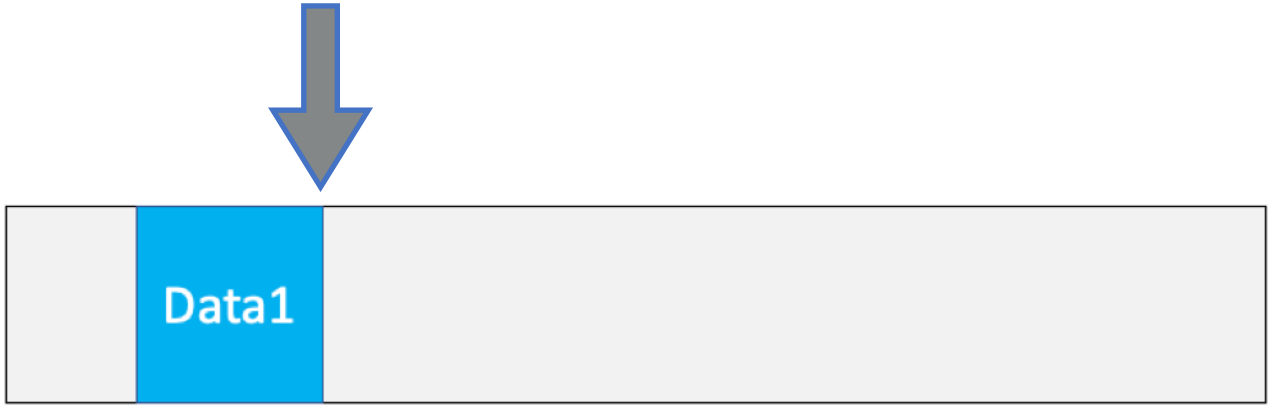
f2fs

- Flash Friendly File System
- 삼성에서 낸드 플래시를 위해 개발 (Log File System)

모바일 파일시스템

f2fs

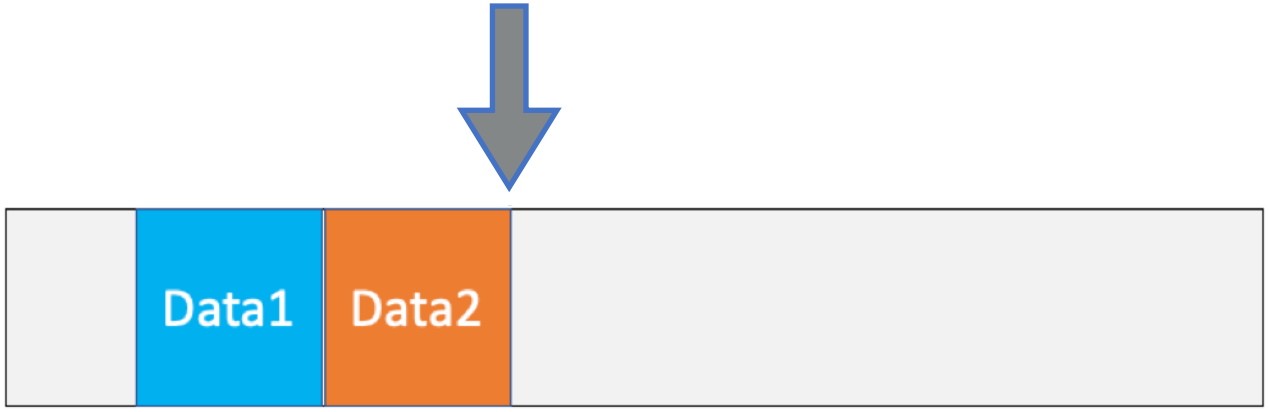
- Flash Friendly File System
- 삼성에서 낸드 플래시를 위해 개발 (Log File System)



모바일 파일시스템

f2fs

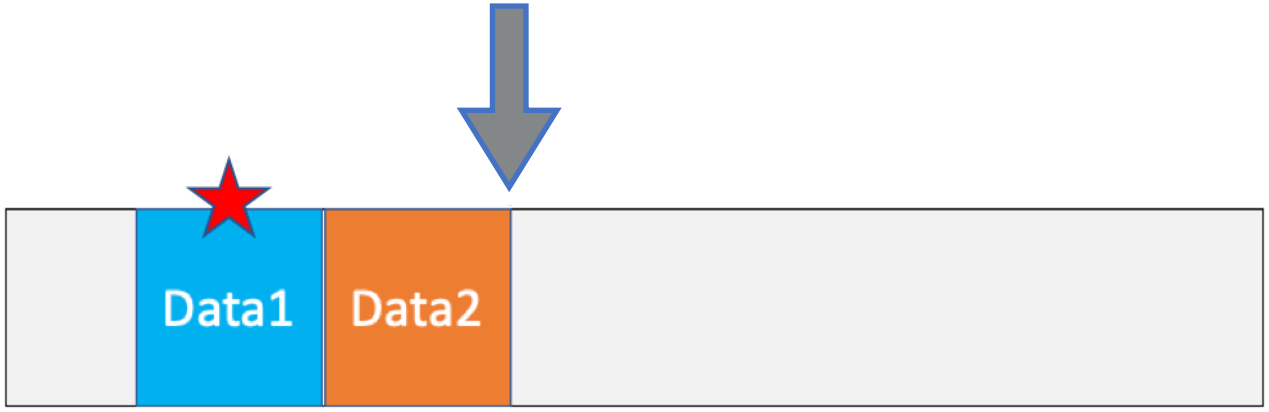
- Flash Friendly File System
- 삼성에서 낸드 플래시를 위해 개발 (Log File System)



모바일 파일시스템

f2fs

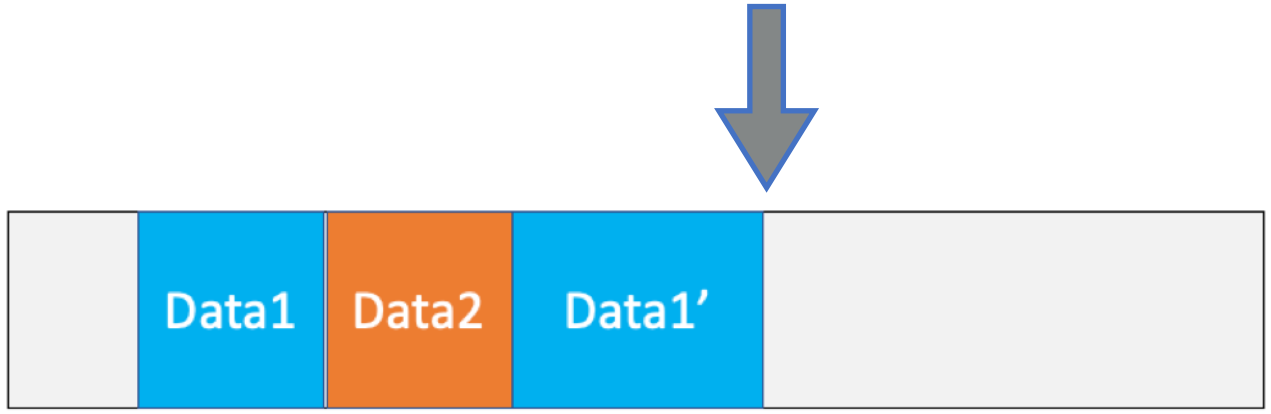
- Flash Friendly File System
- 삼성에서 낸드 플래시를 위해 개발 (Log File System)



모바일 파일시스템

f2fs

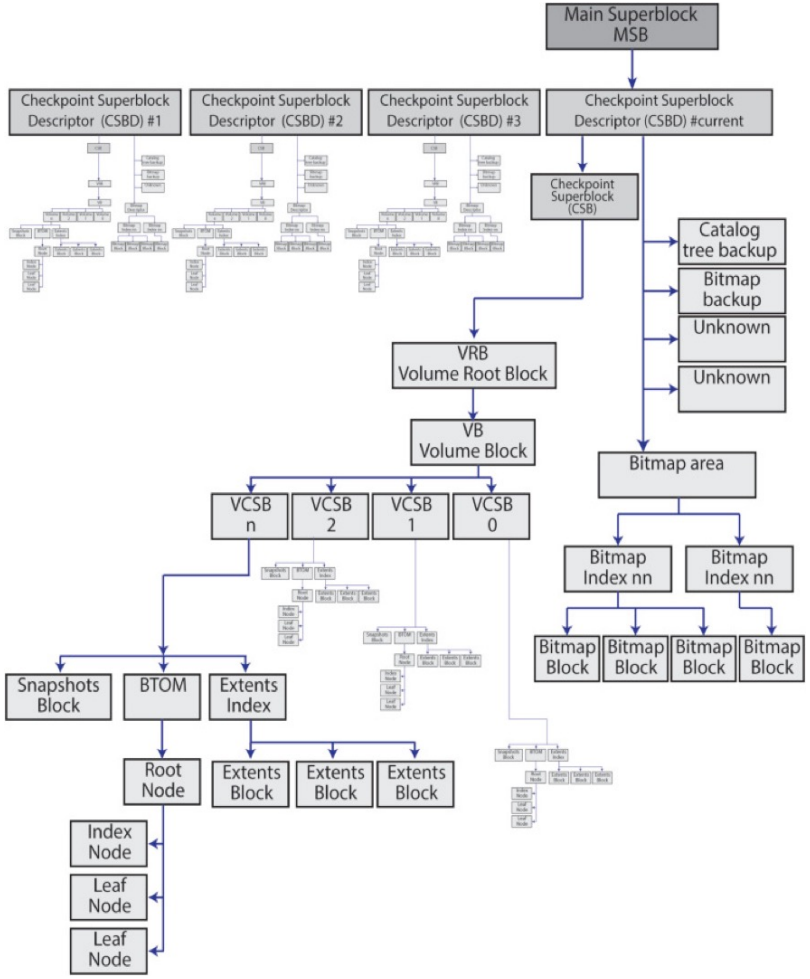
- Flash Friendly File System
- 삼성에서 낸드 플래시를 위해 개발 (Log File System)



모바일 파일시스템

f2fs

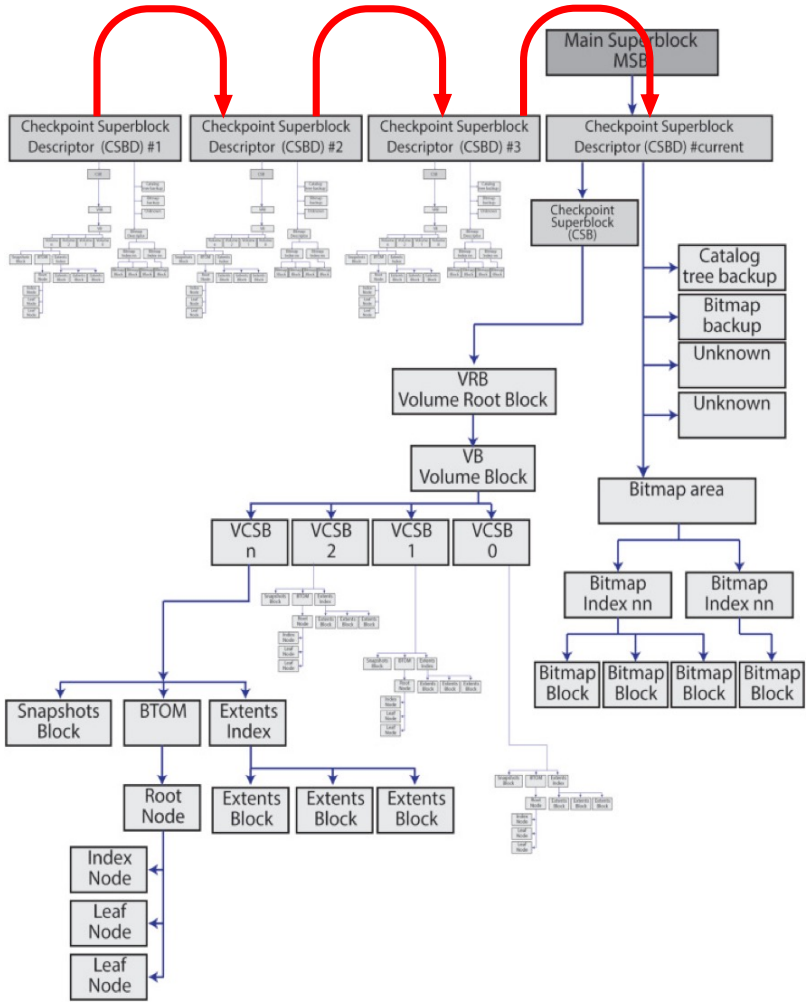
- Flash Friendly File System
- 삼성에서 낸드 플래시를 위해 개발 (Log File System)



모바일 파일시스템

f2fs

- Flash Friendly File System
- 삼성에서 낸드 플래시를 위해 개발 (Log File System)



모바일 파일시스템

f2fs

- Flash Friendly File System
- 삼성에서 낸드 플래시를 위해 개발 (Log File system)
- HWAWEI (P9, P10, P12, …), Motorola (Moto G,E,X,Z),

  OPPO (R11s(t)/R11st Plust), Google (Nexus 9, Pixel 3, Pixel XL…), OnePlus (3, 3T)

모바일 파일시스템

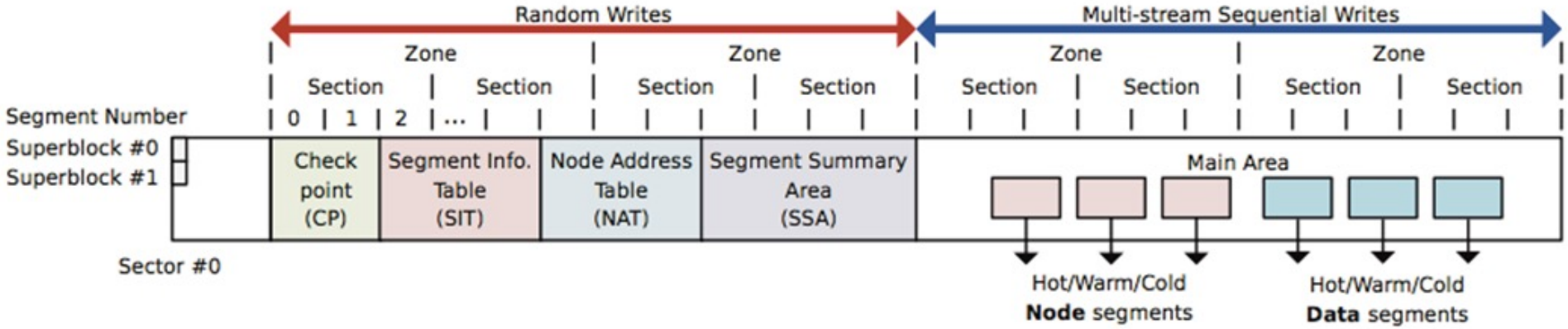
f2fs

- Flash Friendly File System
- 삼성에서 낸드 플래시를 위해 개발 (Log File system)
- HWAWEI, Motorola, OPPO, Google, OnePlus
- 오래된 갤럭시, Nexus에 f2fs를 설치하는 사용자 증가 (2.5 ~ 3 배 이상 빨라짐)

모바일 파일시스템

f2fs

- Flash Friendly File System
- 삼성에서 낸드 플래시를 위해 개발 (Log File system)



모바일 파일시스템

f2fs

- Flash Friendly File System
- 삼성에서 낸드 플래시를 위해 개발 (Log File system)

모바일 파일시스템

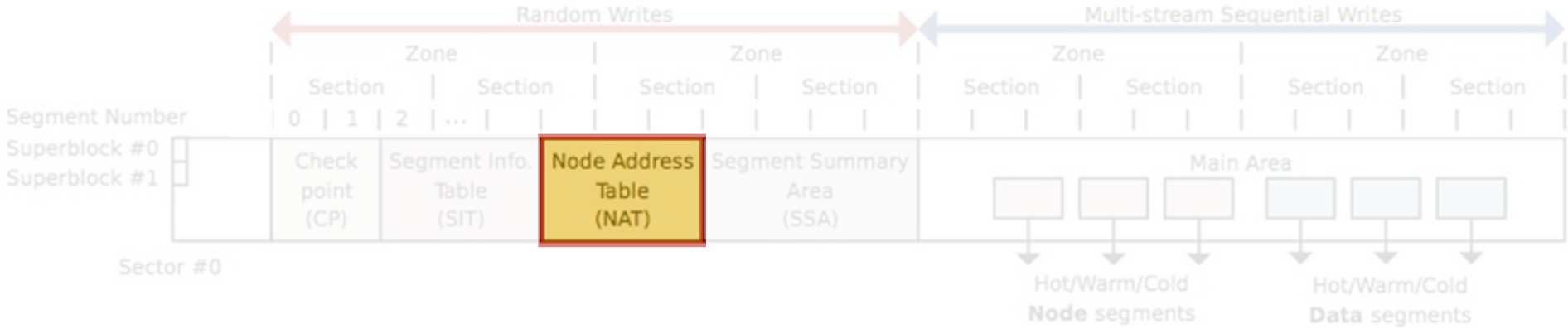
f2fs

- Flash Friendly File System
- 삼성에서 낸드 플래시를 위해 개발 (Log File system)

모바일 파일시스템

f2fs

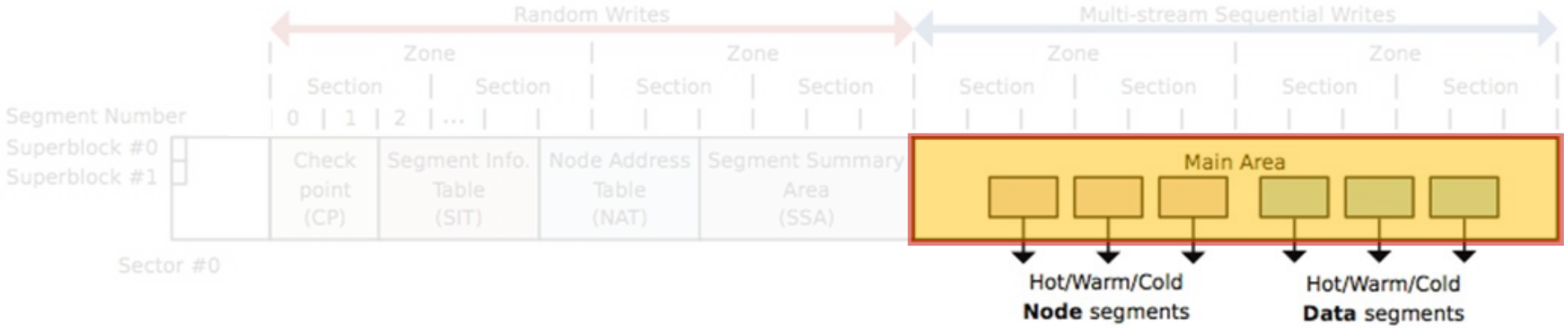
- Flash Friendly File System
- 삼성에서 낸드 플래시를 위해 개발 (Log File system)



모바일 파일시스템

f2fs

- Flash Friendly File System
- 삼성에서 낸드 플래시를 위해 개발 (Log File system)



모바일 파일시스템

f2fs – iNode

- 고전적인 UNIX iNode 구조

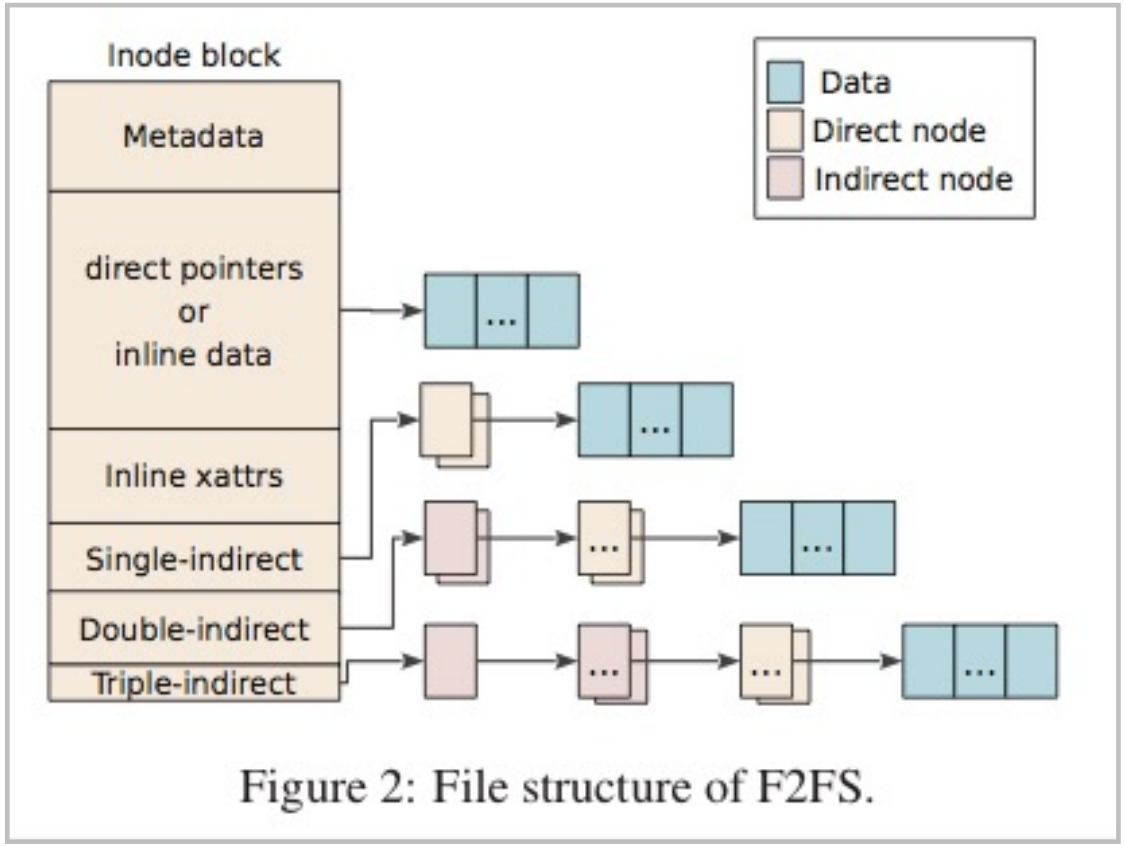


Figure 2: File structure of F2FS.

모바일 파일시스템

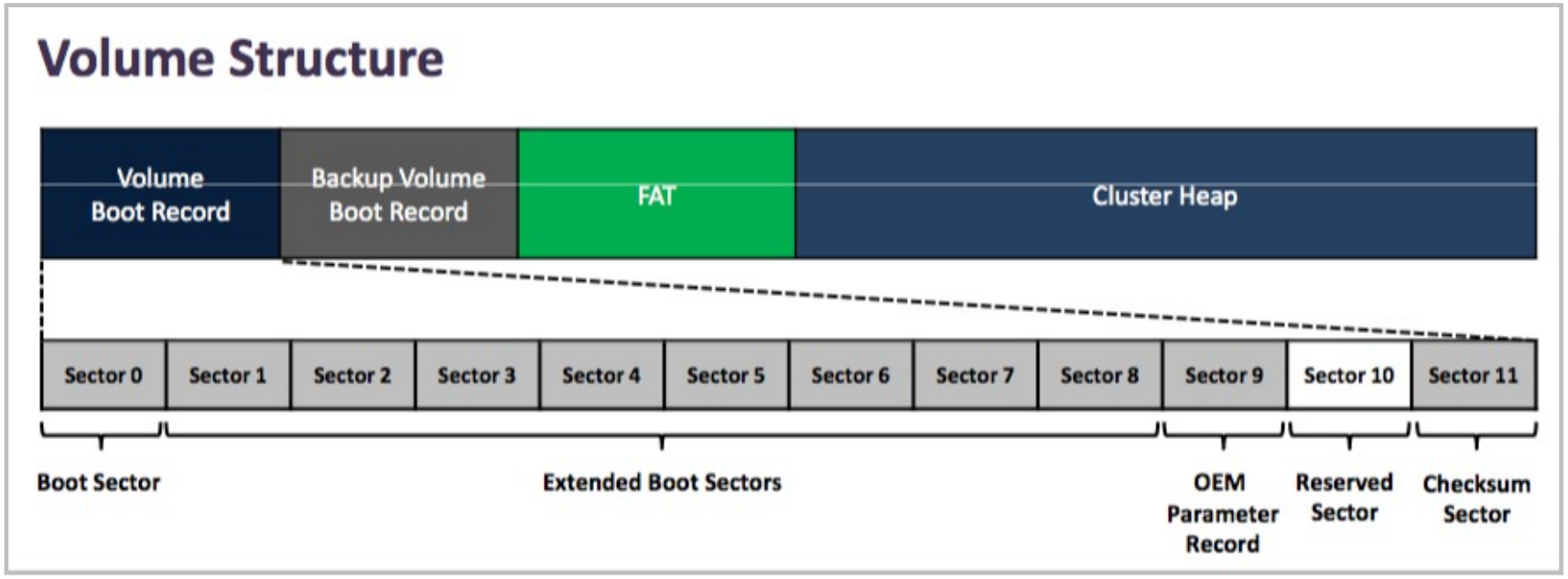
exFAT

- SD 카드에서 널리 사용, TRIM이 적용되지 않음

모바일 파일시스템

exFAT

- SD 카드에서 널리 사용, TRIM이 적용되지 않음
- 볼륨 구조



모바일 파일시스템

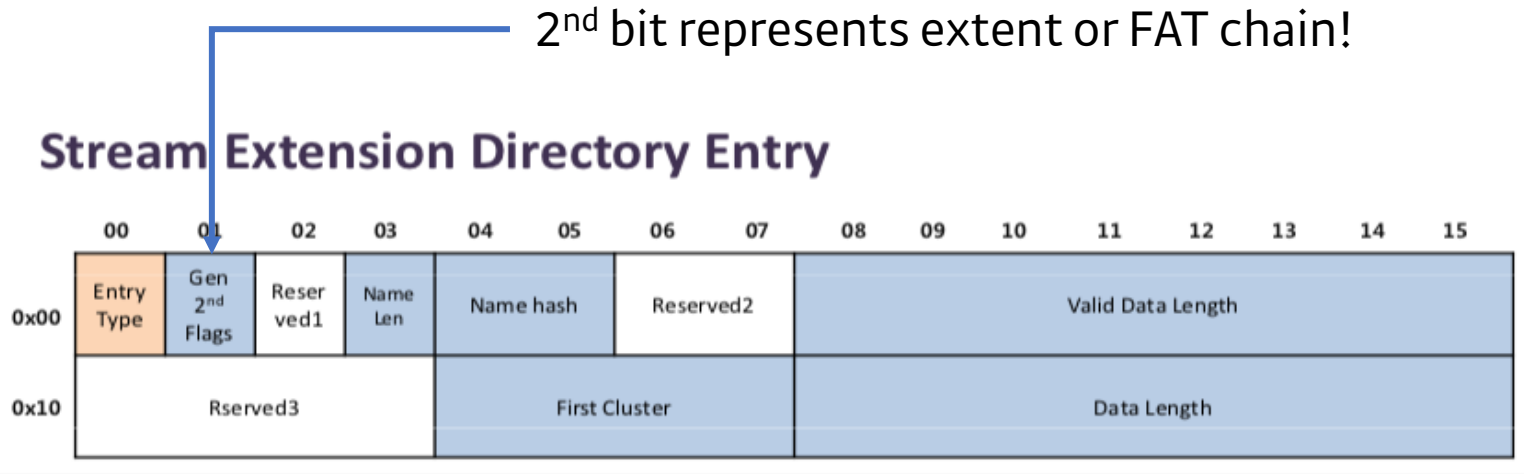
exFAT

- SD 카드에서 널리 사용, TRIM이 적용되지 않음
- 볼륨 구조
- fat 체인과 extent 동시에 사용

모바일 파일시스템

exFAT

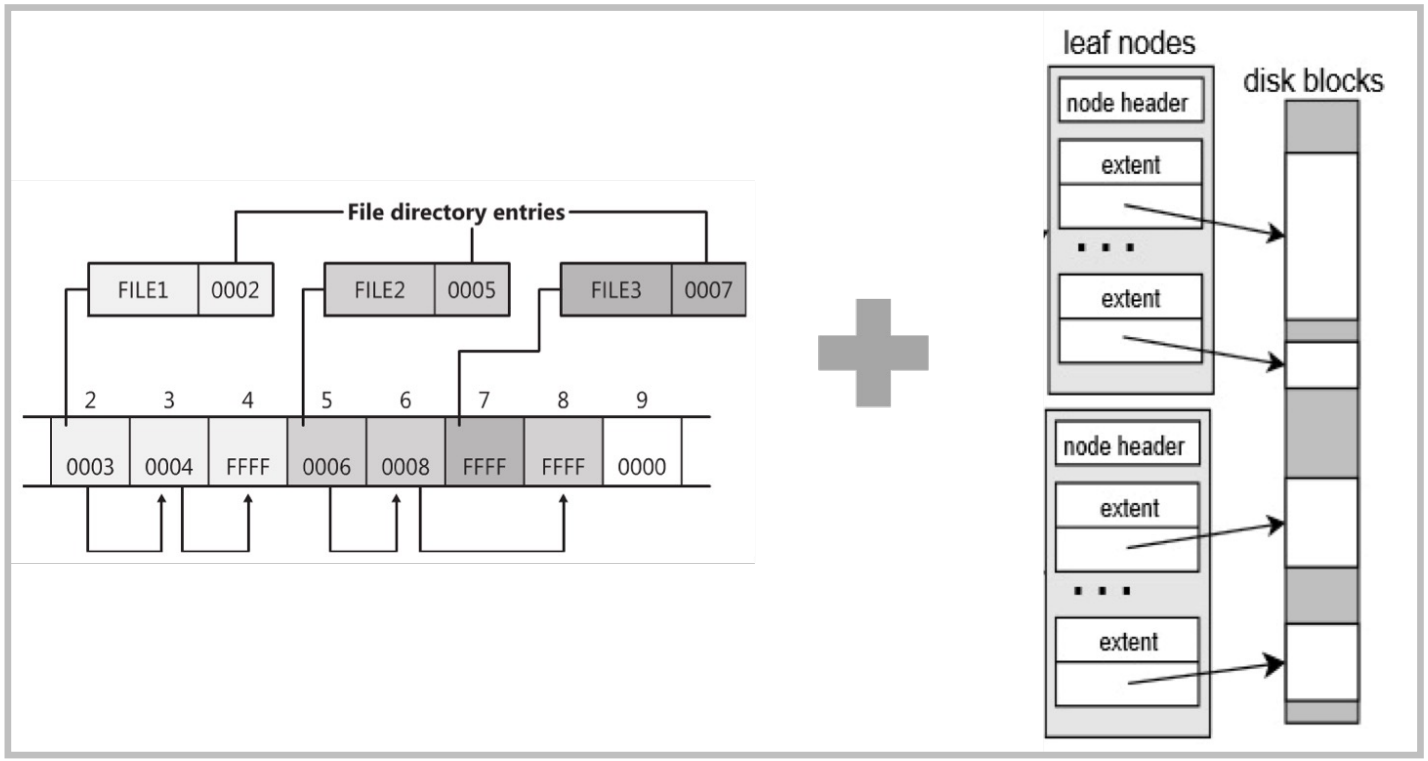
- SD 카드에서 널리 사용, TRIM이 적용되지 않음
- 볼륨 구조
- fat 체인과 extent 동시에 사용



모바일 파일시스템

exFAT

- SD 카드에서 널리 사용, TRIM이 적용되지 않음
- 볼륨 구조
- fat 체인과 extent 동시에 사용



가상 파일시스템

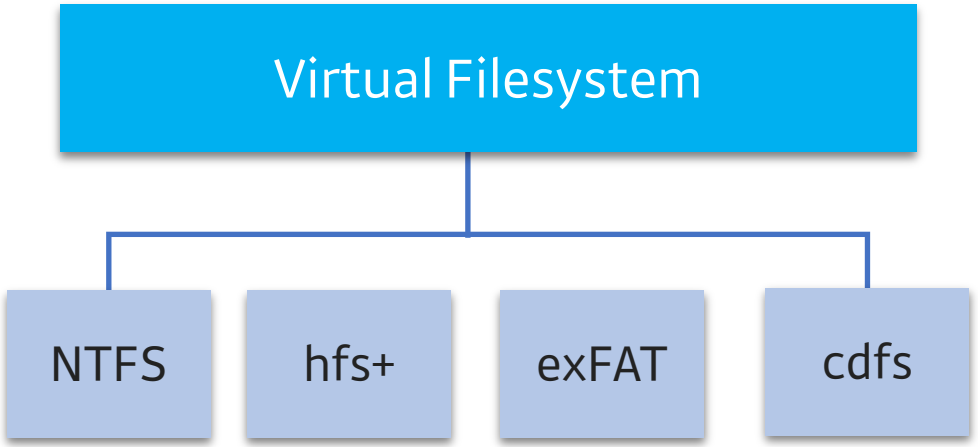
가상 파일시스템

- 서로 다른 파일시스템에 대해 사용자에게 동일한 인터페이스를 제공하는 모듈

가상 파일시스템

가상 파일시스템

- 서로 다른 파일시스템에 대해 사용자에게 동일한 인터페이스를 제공하는 모듈



가상 파일시스템

가상 파일시스템

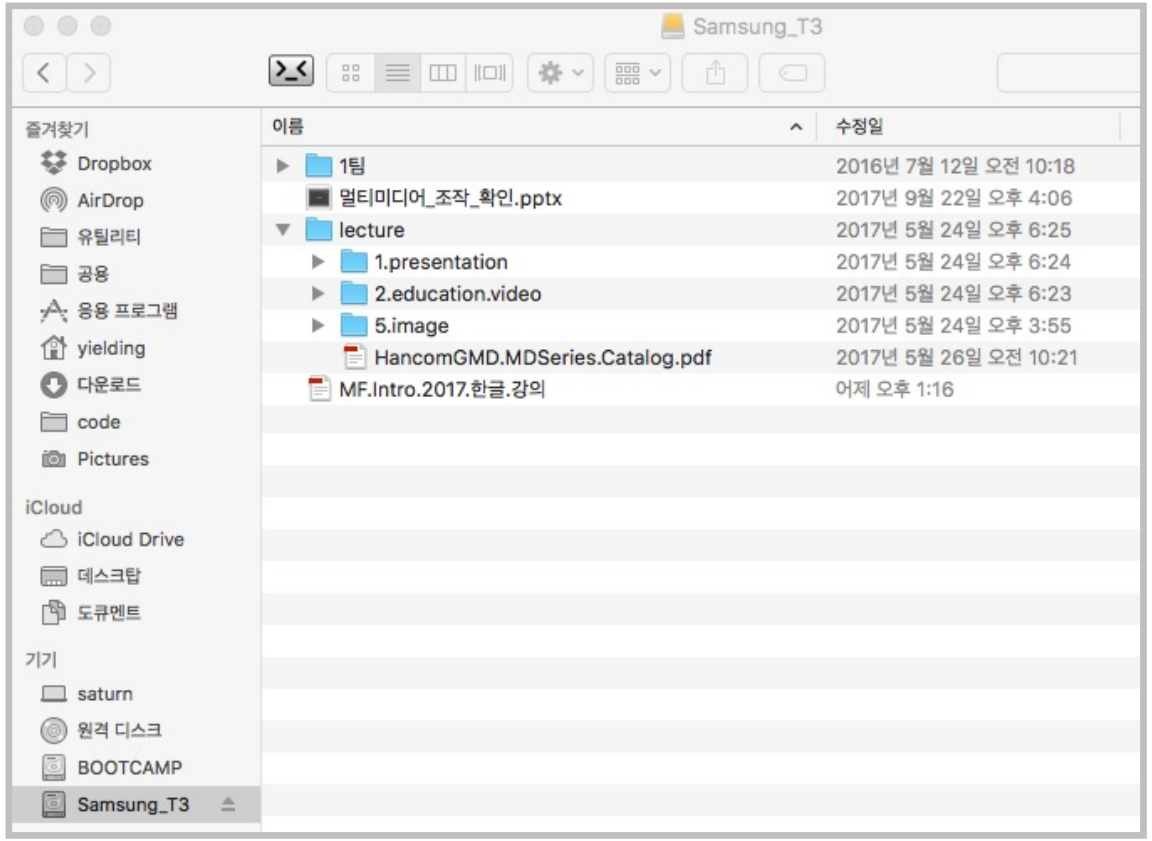
- 서로 다른 파일시스템에 대해 사용자에게 동일한 인터페이스를 제공하는 모듈

- Ex) macOS

  hfs+

  ntfs

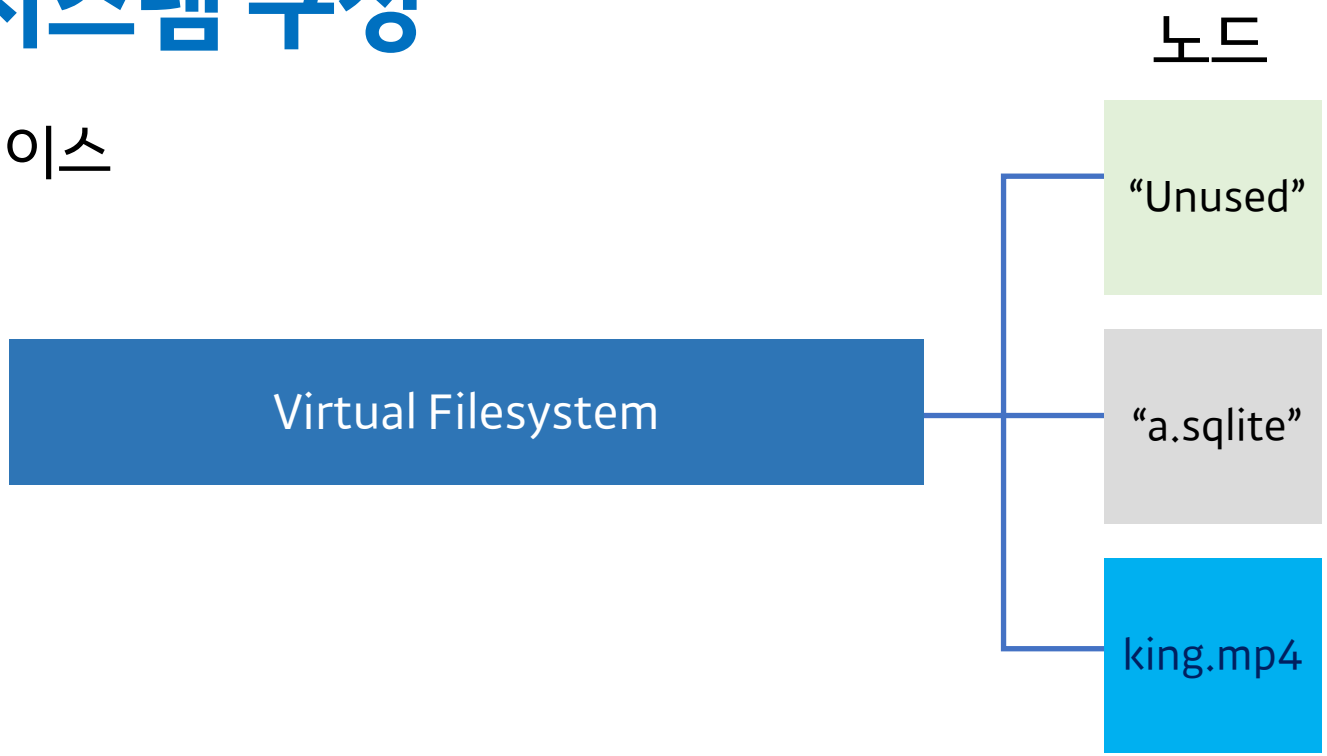
  exFAT



가상 파일시스템

가상 파일시스템 구성

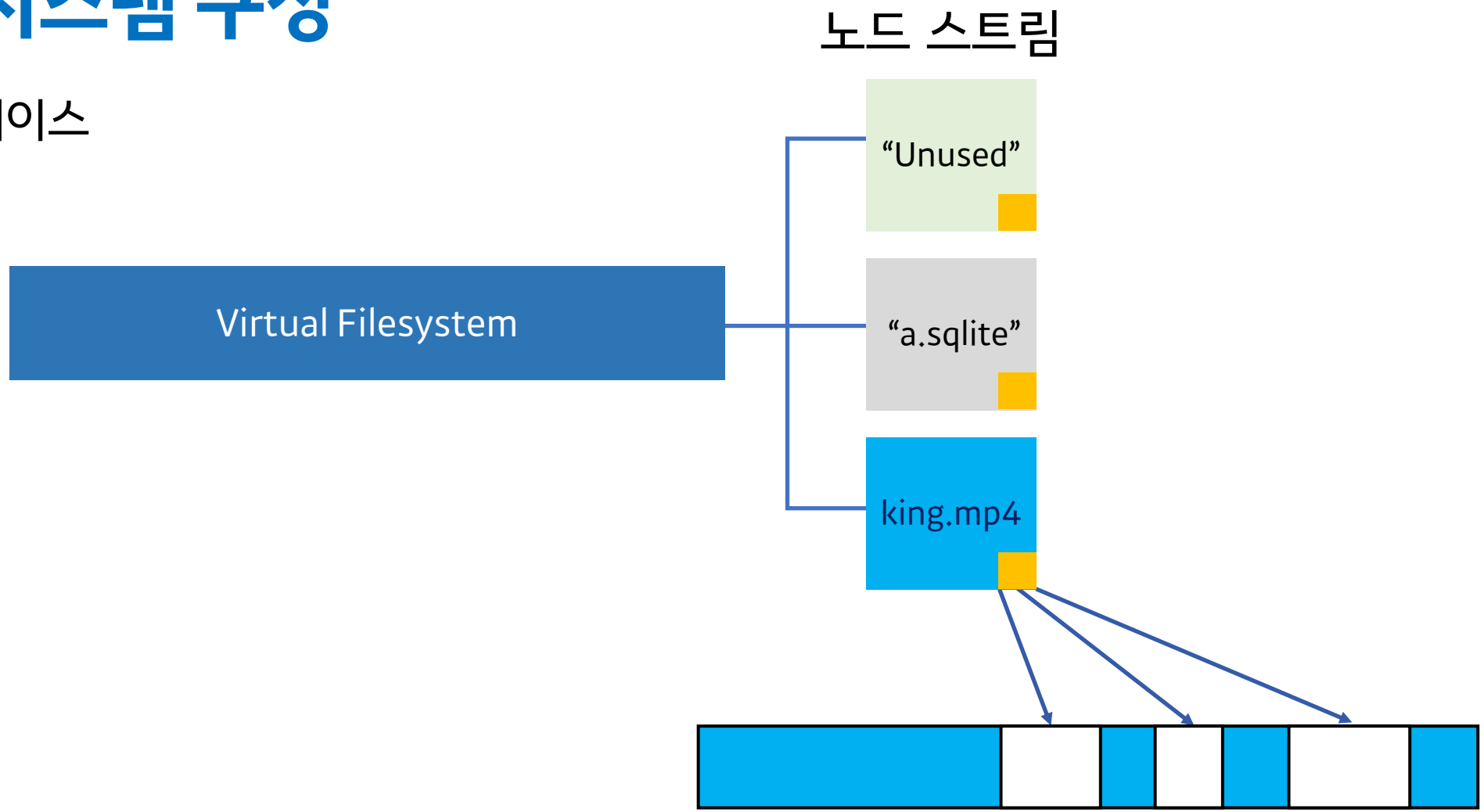
- 사용자 인터페이스



가상 파일시스템

가상 파일시스템 구성

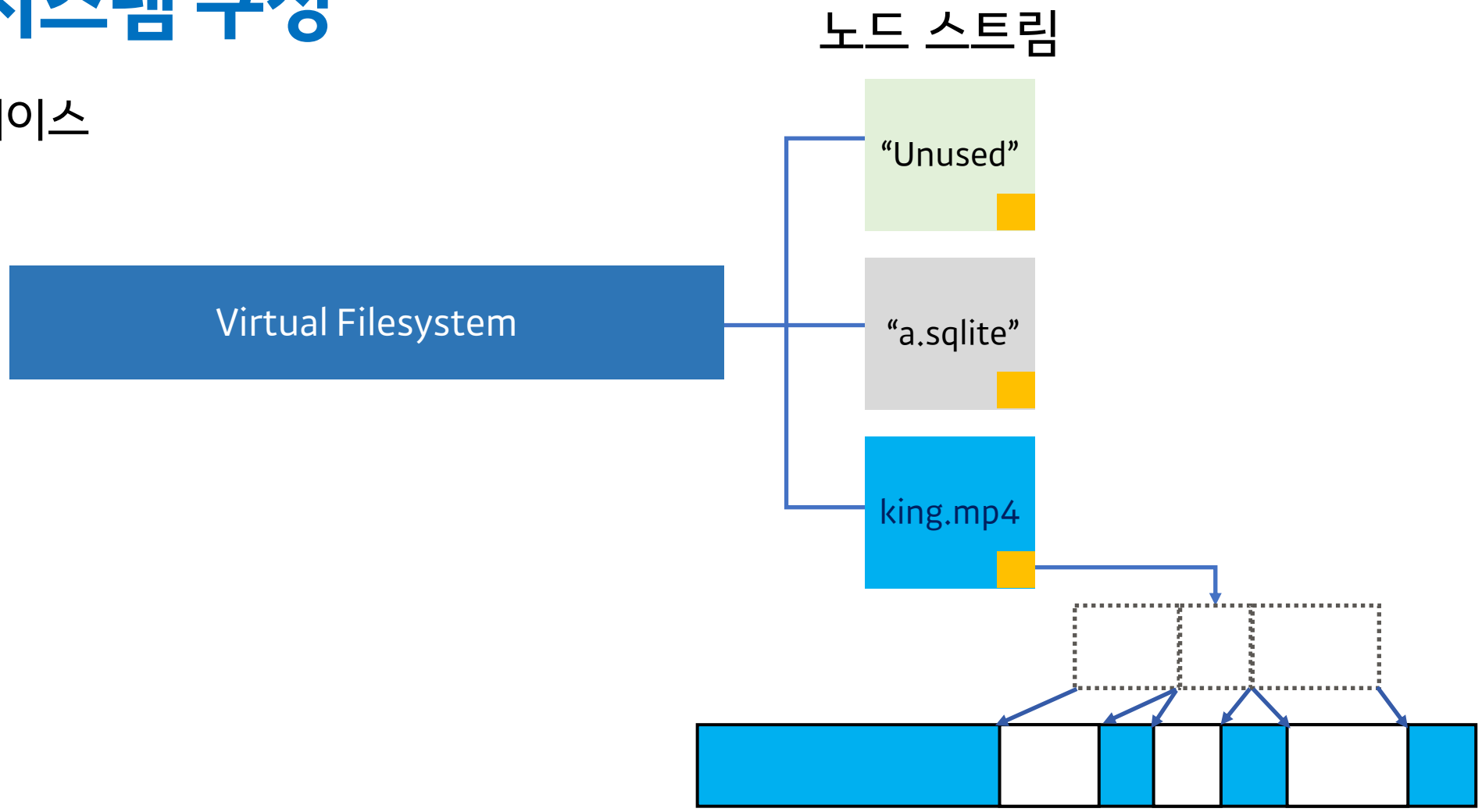
- 사용자 인터페이스



가상 파일시스템

가상 파일시스템 구성

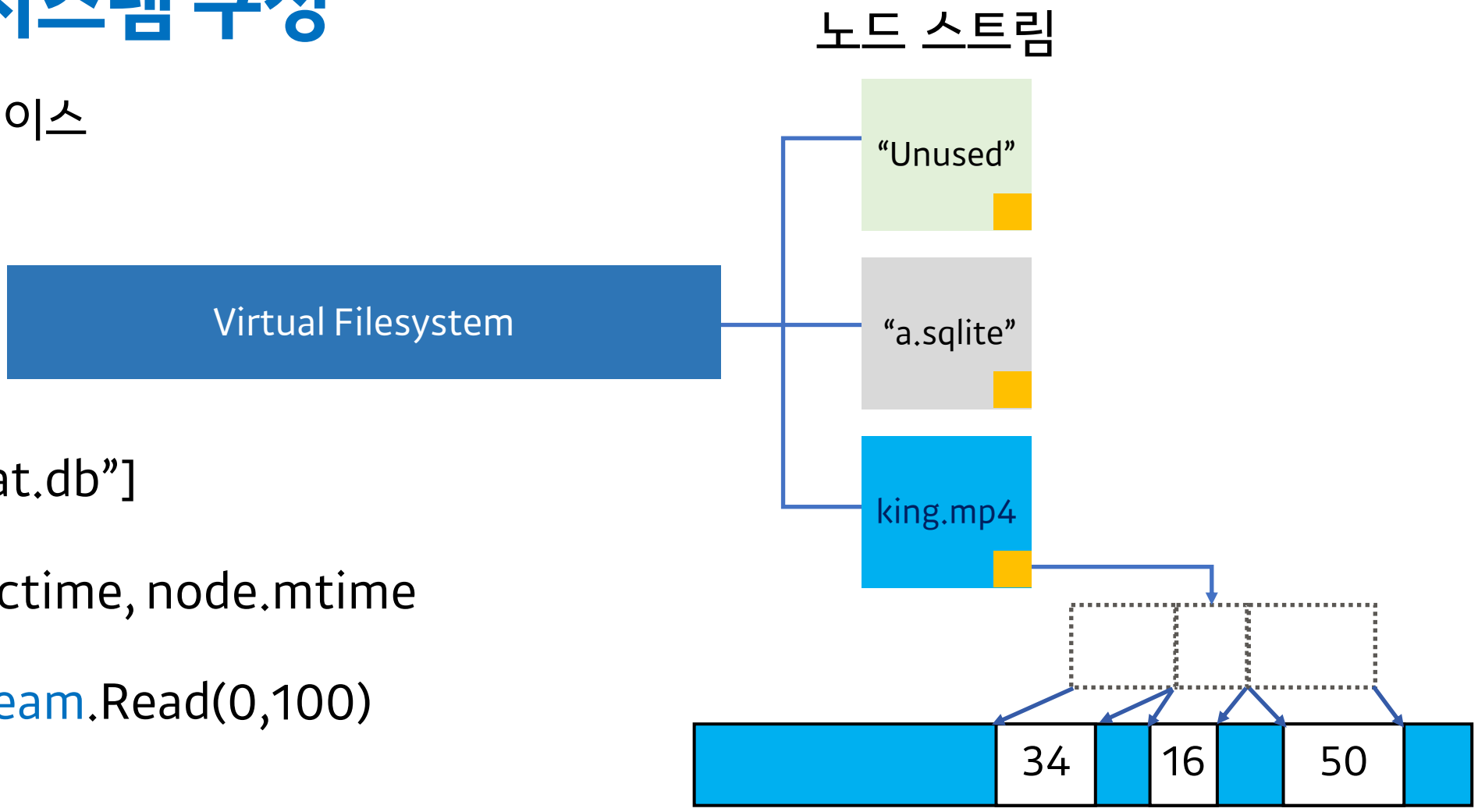
- 사용자 인터페이스



가상 파일시스템

가상 파일시스템 구성

- 사용자 인터페이스



node = fs["chat.db"]

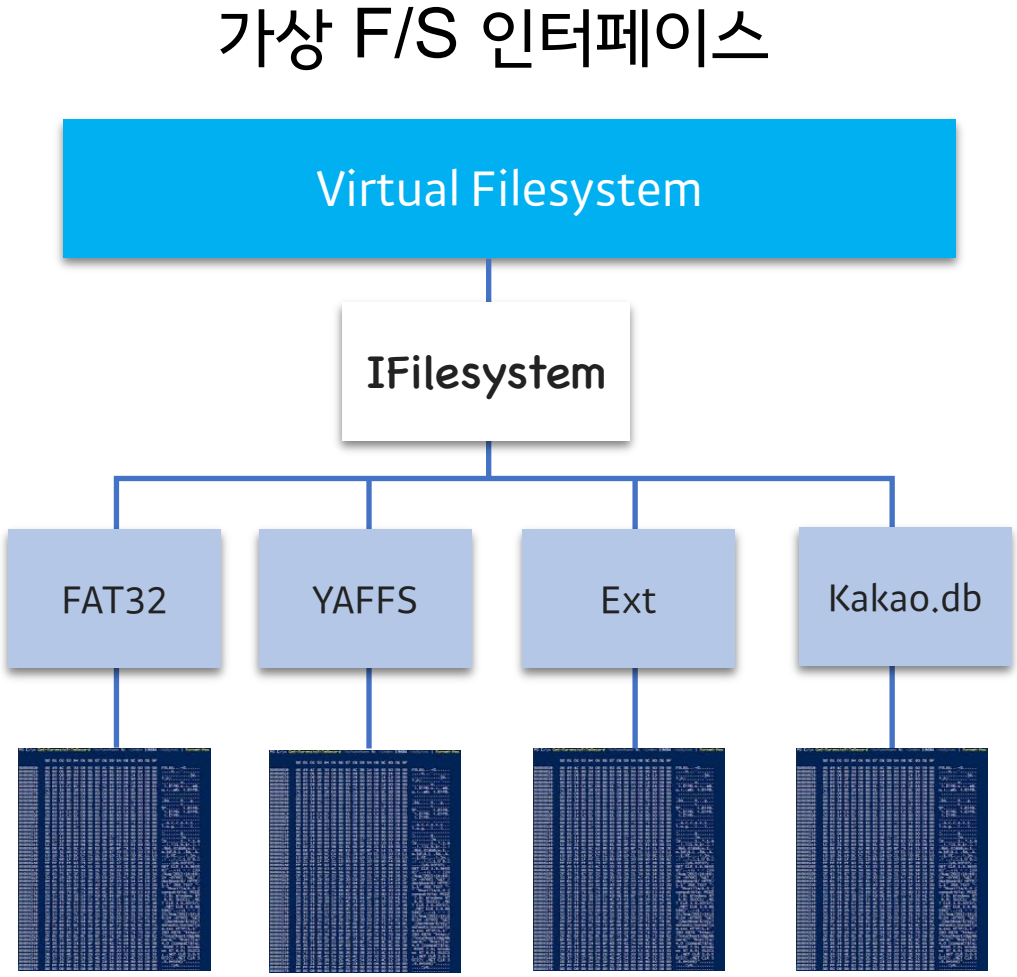
ct, mt = node.ctime, node.mtime

sig = node.stream.Read(0,100)

가상 파일시스템

가상 파일시스템 구성

- 사용자 인터페이스
- F/S 인터페이스

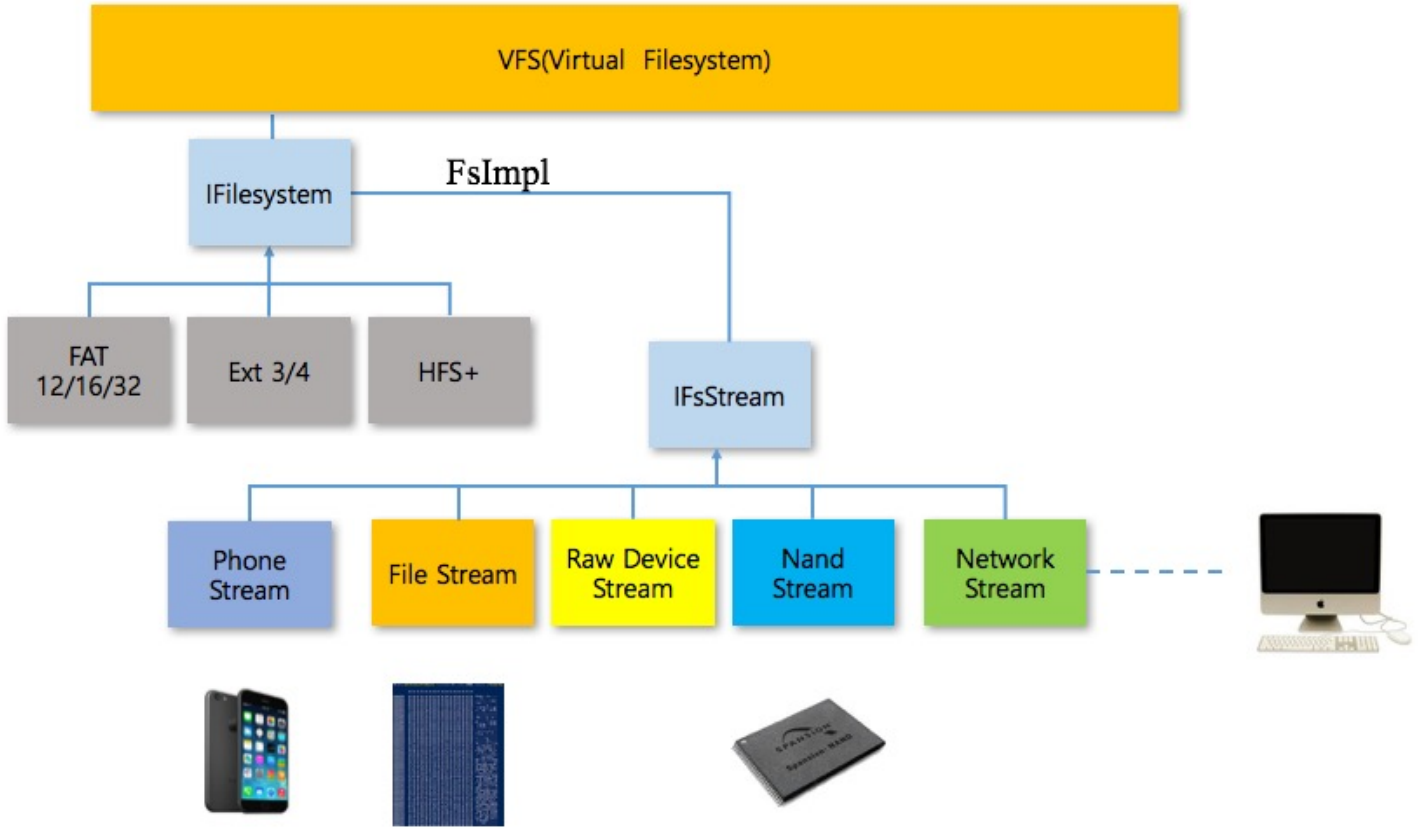


가상 파일시스템

가상 파일시스템 구성

- 사용자 인터페이스
- F/S 인터페이스
- F/S 스트림 인터페이스

F/S 스트림 인터페이스



Q & A

파일시스템 포렌식

테스트

- 슬랙이란 무엇인가? 어떤 종류의 슬랙이 있는가? (10)
- 아이노드에 파일의 경로가 없는 이유는 무엇인가? (5)
- 아이노드에 아이노드 번호가 없는 이유는 무엇인가? (5)
- Ext 파일시스템에서 블록 포인터 방식과 익스텐트 방식의 장단점을 비교하시오.(20)
- 디렉토리 엔트리가 표현하는 가장 중요한 정보는 무엇인가? (5)
- 저널을 분석해서 얻을 수 있는 정보는 무엇인가? (10)
- 일반적인 파일시스템의 수퍼블록에 존재하는 정보에 대해서 기술하시오. (30)
- exFAT에서 발견된 파일이 핸드폰에서 직접 녹화한 것인지 다른 곳에서 복사한 것인지를 어떻게 판단할 수 있는가? (20)
- 피지컬 획득 시 아이노드 테이블을 읽는 중 오류가 발생해서 일부 파일이 복원되지 않은 경우가 생겼다. 이 경우 어떻게 대처해서 누락되는 활성 파일을 분석할 수 있는가? (30)
- Sparse File 이란 무엇인가?(10)
- 동일 증거물에서 로지컬 획득 이미지가 피지컬 획득 이미지보다 더 큰 경우가 생겼다면 어떤 이유에서 발생하는지 설명하시오.(20)