인덕대학교

# 파일 시스템 I
## - 기본 개념

컴퓨터소프트웨어학과

김병국 교수

# 학습목표



- 파일 시스템의 기능을 이해한다.
- 대표적인 파일시스템들에 대한 각 특징을 안다.
- 파일의 개념을 안다.
- 디렉터리의 개념을 안다.
- 파티션과 디스크 포켓 등을 이해한다.

# 목차



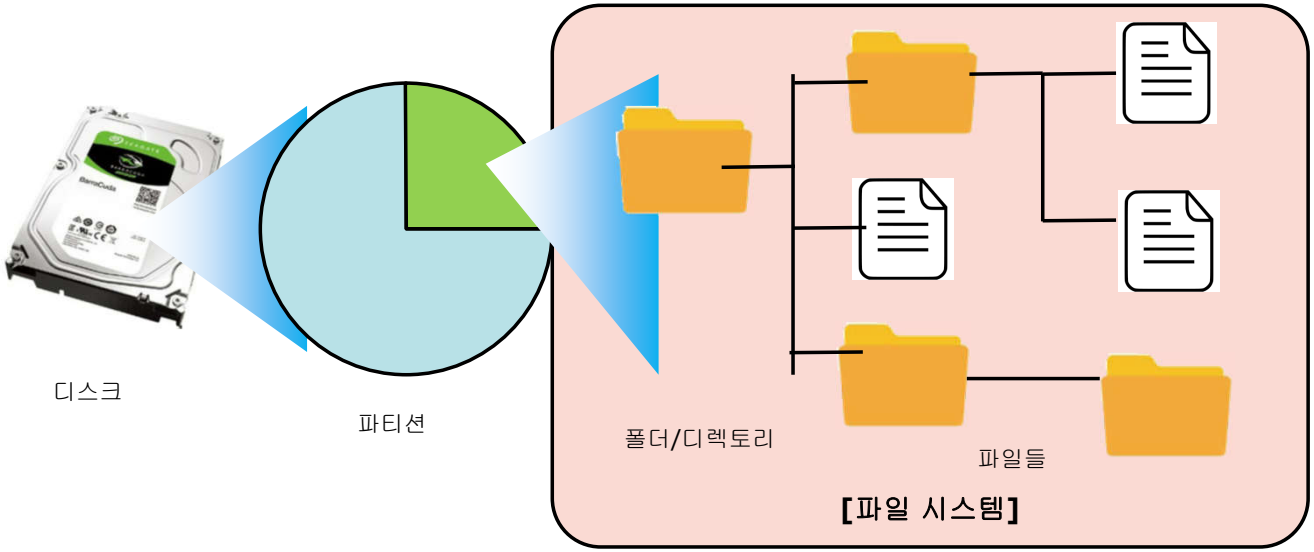


# 1. 파일 시스템 (1/10)



## ❑ 정의

- 저장매체(예: 하드디스크)에 데이터를 효율적으로 관리하기 위한 용도
- 컴퓨팅을 위한 파일들을 관리하는 운영체제의 한 부분
  - 파일에 데이터를 효율적으로 읽고 쓰기 위해 구현
  - 파일들을 효율적으로 관리하기 위해 구현
- 운영체제의 기능 및 특성에 따라 다양한 파일 시스템이 존재



# 1. 파일 시스템 (2/10)



## ❑ 파일 테이블 & 블록(Block) (1/2)

- 블록
  - 운영체제가 저장장치에 데이터의 접근을 위한 가장 작은 단위
  - 블록의 크기는 사용자의 선택에 따라 달리 지정할 수 있음 ← 포맷(Format)과정에서 수행

| 파일 A | 1, 3, 9 |
|---|---|
| 파일 B | 4, 2 |
| 파일 C | 13 |
| 파일 D | 15, 12 |
| 파일 E | 23, 7 |

파일 테이블

| 블록번호 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | | A | B | A | B | | | E | | A |
| 2 | | | D | C | | D | | | | |
| 3 | | | | E | | | | | | |
| 4 | | | | | | | | | | |
| 5 | | | | | | | | | | |

저장장치

  - 작은 블록의 크기
    - 내부 단편화 줄어듦
    - 블록의 개수가 많음 ← 성능 저하
  - 큰 블록의 크기
    - 내부 단편화가 많아짐 ← 공간 낭비
    - 블록의 개수가 적음 ← 성능 향상

# 1. 파일 시스템 (3/10)



## ❑ 파일 테이블 & 블록(Block) (2/2)

- 파일 테이블
  - 블록 테이블이라고도 함
  - 파일의 항목들과 각 파일이 기록된 하드디스크내 블록의 위치정보를 관리

| 파일 A | 1, 3, 9 |
|---|---|
| 파일 B | 4, 2 |
| 파일 C | 13 |
| 파일 D | 15, 12 |
| 파일 E | 23, 7 |

파일 테이블

| 블록번호 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | | A | B | A | B | | | E | | A |
| 2 | | | D | C | | D | | | | |
| 3 | | | | E | | | | | | |
| 4 | | | | | | | | | | |
| 5 | | | | | | | | | | |

저장장치

※ 하드디스크의 최소 저장 단위 : 섹터(sector)

→광대한 용량으로 인해 섹터의 주소는 너무 많음

→섹터들을 관리하기에는 시스템적으로 부담이 큼

→여러 섹터를 하나의 블록으로 묶음 처리

→최종적으로, 블록의 주소값을 활용

# 1. 파일 시스템 (4/10)



## ❑ 대표적 종류

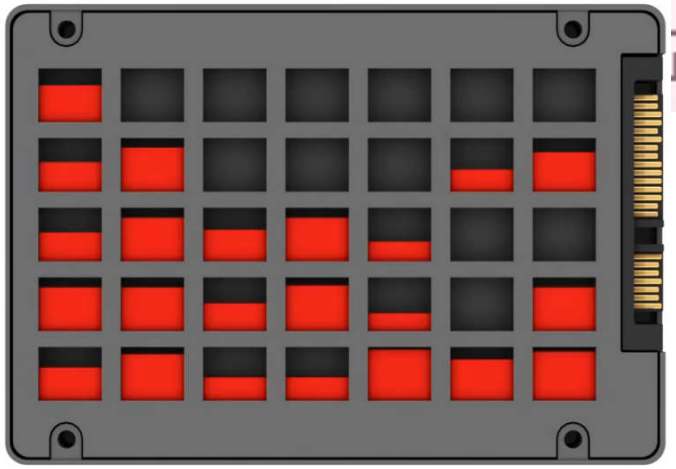
- FAT(16/32) : 윈도우용 파일시스템
- NTFS : 윈도우용 파일시스템
- EXT(2/3/4) : 리눅스에서 주로 사용
- HFS+/APFS : macOS용 파일시스템

# 1. 파일 시스템 (5/10)

❑ FAT 16 파일 시스템
- FAT : File Allocation Table(파일 할당 테이블)
- MS-DOS에서 부터 윈도우95까지 사용
- 최대 2GB까지의 용량을 지원
- 파일명은 최대 8문자가 가능(확장자 3글자 제외)

❑ FAT 32 파일 시스템
- 기존 FAT 16의 결점을 보완
- 윈도우 98 ~ 윈도우ME에서 기본 채택
- 최대 2TB까지의 용량을 지원
- 파일명은 최대 256문자가 가능
- USB 메모리 및 다양한 외부 저장매체에도 활용됨
  ← 일부 변형 파일 시스템(exFAT)이 탑재

❑ FAT 16 & 32의 단점
- 보안 기능이 결여 → 공용 파일 저장용으로 활용
- 저용량 볼륨(Volume)에 최적



[클러스터내 데이터 예]

# 1. 파일 시스템 (6/10)



## ❑ NTFS 파일 시스템

- NTFS : New Technology File System
- Windows NT 파일 시스템
- FAT32를 극복하기 위해 설계
- 지원 운영체제: NT 이후 모든 윈도우 운영체제
- 파일명 길이 : 최대 32,767문자 (운영체제 별 일부 차이는 있음)
- 최대 지원 용량: 256TB
- 보안 및 암호화 지원
- 저널링 파일 시스템(Journaling File System)


## ※ 저널링 파일 시스템

- 기록 데이터의 변화 과정을 로그(log)
- 문제 발생 시 복구가 가능

# 1. 파일 시스템 (7/10)



## ❑EXT(2/3/4) 파일 시스템

- EXT: Extended File System
- 리눅스 기반의 운영체제를 위해 설계(EXT1: 1992)
- EXT1에서 부터 기능이 보완되고 확장되면서 EXT4(2008)까지 개량
- EXT3에서부터 저널링 기능이 탑재
- EXT4기준 최대 파일 크기: 16TB
- EXT4기준 최대 볼륨 크기: 1EB(1TB x $10^6$)

# 1. 파일 시스템 (8/10)



## ❑HFS & APFS 파일 시스템

- HFS: Hierarchical File System
- 애플의 macOS를 위해 1985에 도입
- 1998년 HFS+로 개량(저널링 기능 탑재)
- HFS : 최대 파일 크기: 2GB, 볼륨 크기: 2TB
- HFS+ : 최대 파일 및 볼륨 크기: 8EB
- 2017년 APFS(Apple File System)로 변경

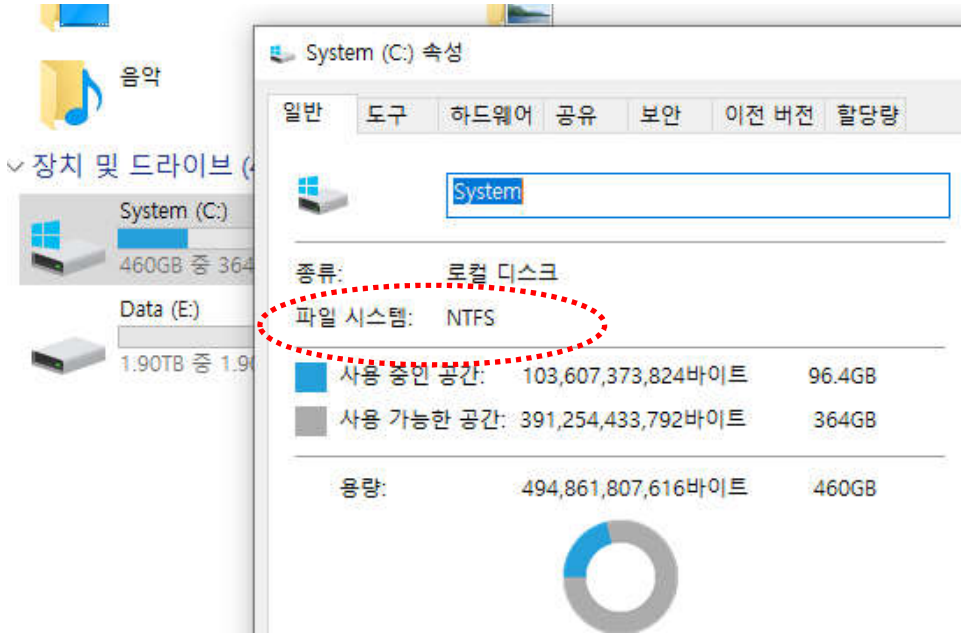
macOS®

Mac OS

# 1. 파일 시스템 (9/10)

인덕대학교

## ❑파일 시스템 확인하기



# 1. 파일 시스템 (10/10)



## ❑ 구성

- 파일(Files)
  - 연관된 정보들의 집합
  - 데이터 본체가 기록된 공간
- 디렉토리(Directories)
  - 파일들의 정보를 구성
  - 파일들의 구성을 계층화
  - 동의어: 폴더(Folders)
  - 디렉토리도 파일의 한 종류
- 파티션(Partitions)
  - 파일시스템을 구성하기 위한 디스크의 논리적인 공간

# 2. 파일 (1/8)



## ❑ 정의

- 보조 기억장치에 저장된 정보들의 집합
- 기록된 정보 집합의 최소 단위
- 바이트 배열이 저장된 공간(Sequence of bytes)

- 기록 형태에 따른 분류
  - 텍스트(아스키) 파일
  - 바이너리(이진) 파일

# 2. 파일 (2/8)



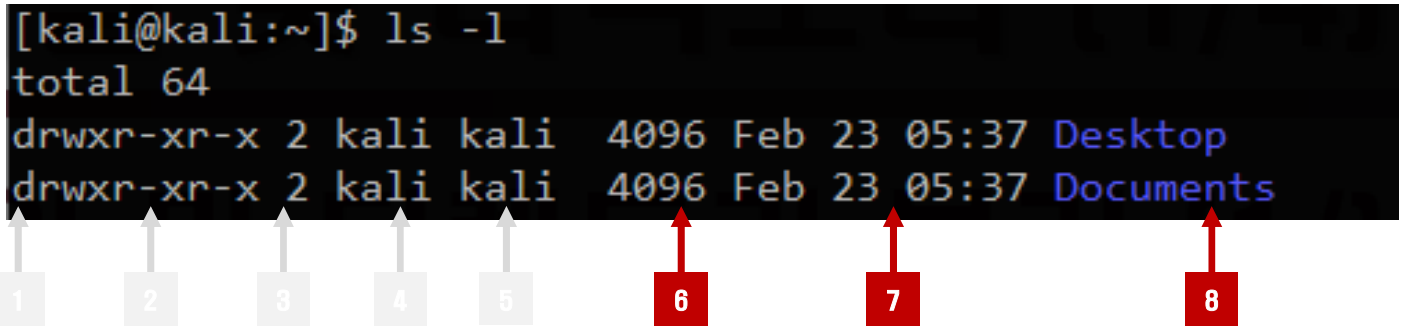
## ❑파일 속성 (1/2)

```
[kali@kali:~]$ ls -l
total 64
drwxr-xr-x 2 kali kali  4096 Feb 23 05:37 Desktop
drwxr-xr-x 2 kali kali  4096 Feb 23 05:37 Documents
```

1 2 3 4 5 6 7 8

| 번호 | 값 | 동작 |
|---|---|---|
| 1 | d | 파일 종류 (- : 일반파일, d: 디렉토리) |
| 2 | rwx-xr--r-x | 파일을 읽고, 쓰고, 실행할 수 있는 권한 표시 |
| 3 | 2 | 물리적 연결 개수 |
| 4 | kali | 파일 소유자의 사용자 명 |
| 5 | kali | 파일 소유자의 그룹명 |

## ❑파일 속성 (2/2)




| 번호 | 값 | 동작 |
|---|---|---|
| 6 | 4096 | 파일 크기(바이트 단위) |
| 7 | Feb 23 05:37 | 파일이 마지막으로 변경된 시간 |
| 8 | Desktop | 파일 또는 디렉토리 명 |

# 2. 파일 (4/8)



## ❑파일의 유형

```
[kali@kali:~]$ ls -l
total 64
drwxr-xr-x 2 kali kali  4096 Feb 23 05:37 Desktop
drwxr-xr-x 2 kali kali  4096 Feb 23 05:37 Documents
drwxr-xr-x 2 kali kali  4096 Feb 23 05:37 Downloads
-rwxr-xr-x 1 kali kali 16608 Apr  2 01:37 hello
-rw-r--r-- 1 kali kali    72 Apr  2 01:37 hello.c
drwxr-xr-x 2 kali kali  4096 Feb 23 05:37 Music
drwxr-xr-x 2 kali kali  4096 Mar 26 01:03 NetworkPro
```

| 번호 | 파일 유형 |
|---|---|
| – | 일반 (정규) 파일 |
| d | 디렉토리 파일 |
| b | 블럭 단위로 읽고 쓰는 블럭 장치 특수 파일 |
| c | 문자 단위로 읽고 쓰는 문자 장치 특수 파일 |
| l | 기호적 링크 |
| p | 파이프 |
| s | 소켓 |

# 2. 파일 (5/8)



## ❑파일 처리 기능

- 생성
- 읽기
- 쓰기
- 이동(파일의 이름 변경 포함)
- 삭제
- 기타

- 운영체제의 시스템 함수 호출에 의해 제공

# 2. 파일 (6/8)



## ❑ 파일 기록 구조 (1/3)

- 순차 파일 구조(Sequential File Structure)
  - 파일의 내용이 하나의 연속된 줄로 기록된 형태
  - 연속적 데이터 접근을 기반한 매체(자기 테이프)에서 가장 효율적 방식

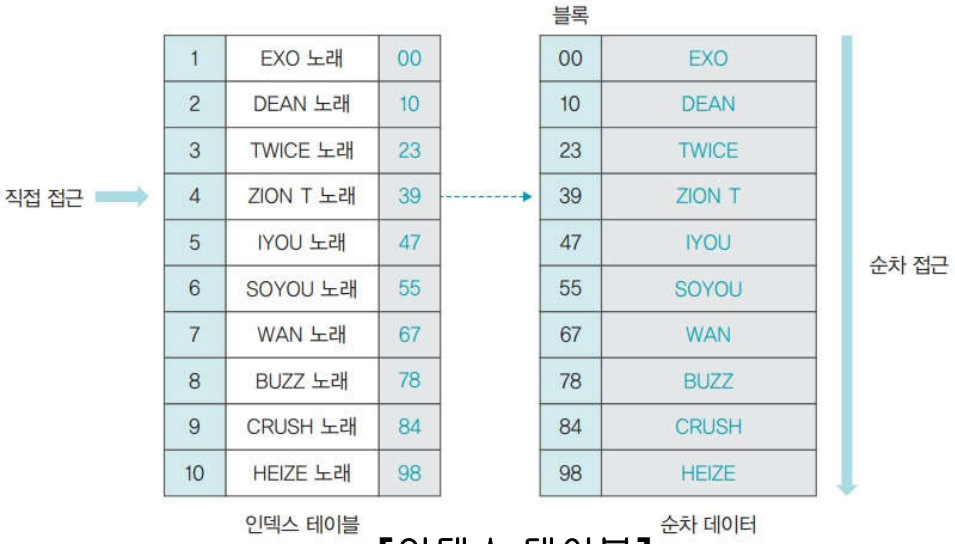
[자기 테이프]

  - 장점
    - 낭비되는 저장 공간이 없음
    - 데이터의 빠른 읽기/쓰기가 가능
  - 단점
    - 내용 갱신 및 중간 삽입이 비효율적임
    - 위치 이동 시 너무 오래 걸림

# 2. 파일 (7/8)

## ❑ 파일 기록 구조 (2/3)

- 인덱스 파일 구조(Index File Structure)
  - 순차 파일 구조에 인덱스를 위한 필드를 추가한 방식
  - 주로 디스크 기반의 저장매체에서 사용
  - 현대의 기록매체에서 주로 사용하는 방식




**[인덱스 테이블]**

  - 장점
    - 빠른 접근 및 위치 이동이 가능
    - 중간에 삽입 및 변경이 쉬움
  - 단점
    - 내부 단편화가 발생
    - 디스크의 낭비가 발생될 수 있음
      (← 가성비 고려 대용량으로 극복)

# 2. 파일 (8/8)

## ❑파일 기록 구조 (3/3)

- 직접 파일 구조(Direct File Structure)
  - 물리적인 저장매체의 주소에 직접 접근하는 방식
  - 해시(hash) 기법을 사용


[직접 파일 구조]

  - 장점
    - 블록의 개수가 너무 많을 때 빠른 접근이 가능
  - 단점
    - 해시 충돌에 대한 해결방안이 필요

# 3. 디렉터리 (1/6)



## ❑ 정의

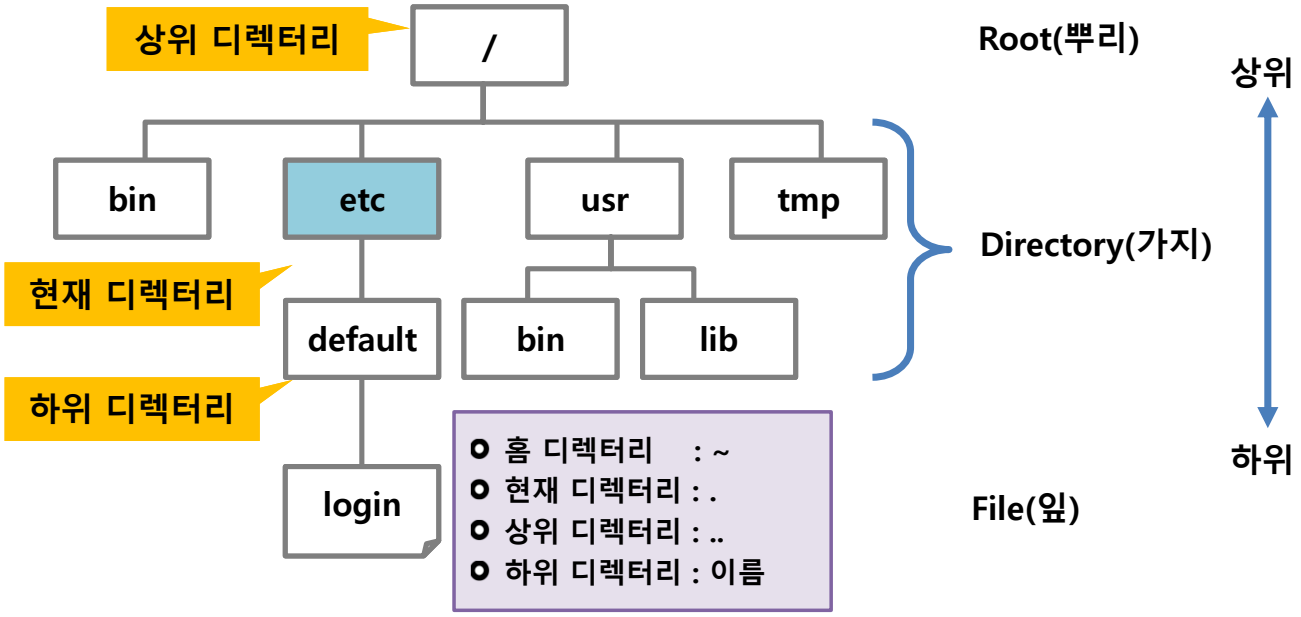
- Directory
- 파일시스템에서 관리되는 파일들을 체계적으로 관리하기 위한 공간
- 공간: 논리적(가상) 영역
- 파일 및 하부 디렉터리들의 리스트를 관리
- 디렉터리 내 별도의 파일시스템 구성이 가능
  - 파일들을 기능 또는 속성에 따라 분류해서 보관이 가능

# 3. 디렉터리 (2/6)



## ❑ 경로(Path) (1/3)

- 파일이 전체 디렉터리 중 어디에 있는지를 나타내는 정보
- 동일 디렉터리에는 동일한 명칭의 파일 또는 디렉터리가 존재할 수 없음
  - 단, 동일 명의 파일이나 디렉터리는 다른 디렉터리에 존재할 수 있음
- 루트(root) 디렉터리: 최상위 디렉터리를 의미



# 3. 디렉터리 (3/6)



## ❑경로(Path) (2/3)

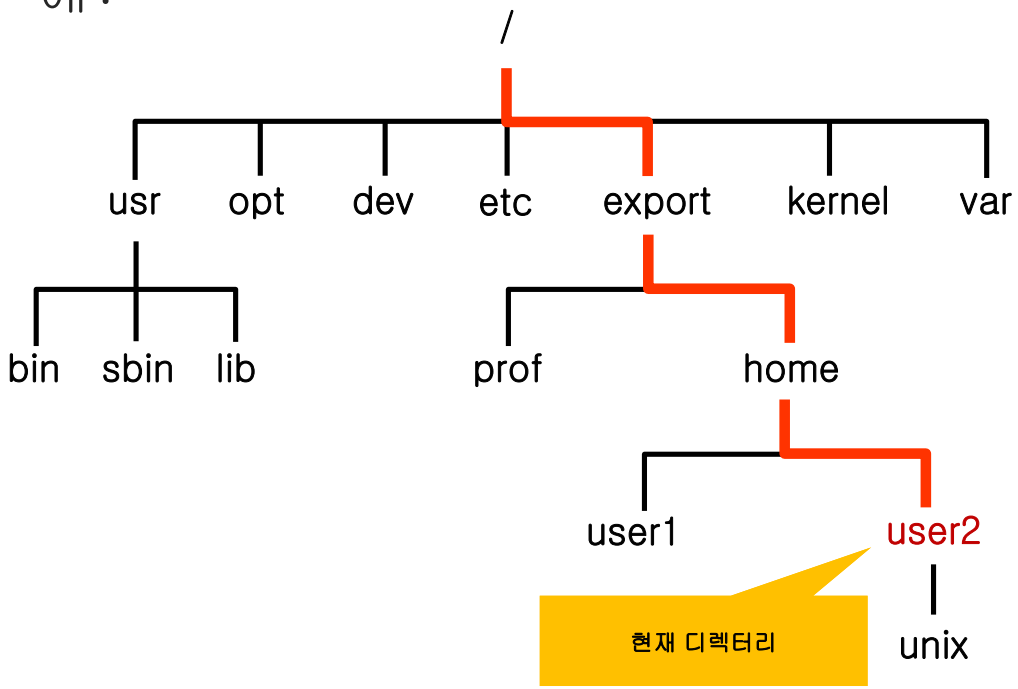
- 절대 경로
  - 루트 디렉터리(/)를 기준으로 파일의 위치를 나타내는 방식
  - 예: /etc/default/
- 상대 경로
  - 현재 있는 위치를 기준으로 파일의 위치를 표시하는 방식
  - 예: ../ (상위 경로를 의미)

# 3. 디렉터리 (4/6)



## ❑경로(Path) (3/3)

- 예 :




unix의 절대경로

/export/home/user2/unix

user1의 상대경로

../user1

# 3. 디렉터리 (5/6)



## ❑파일 및 디렉터리의 명명 규칙

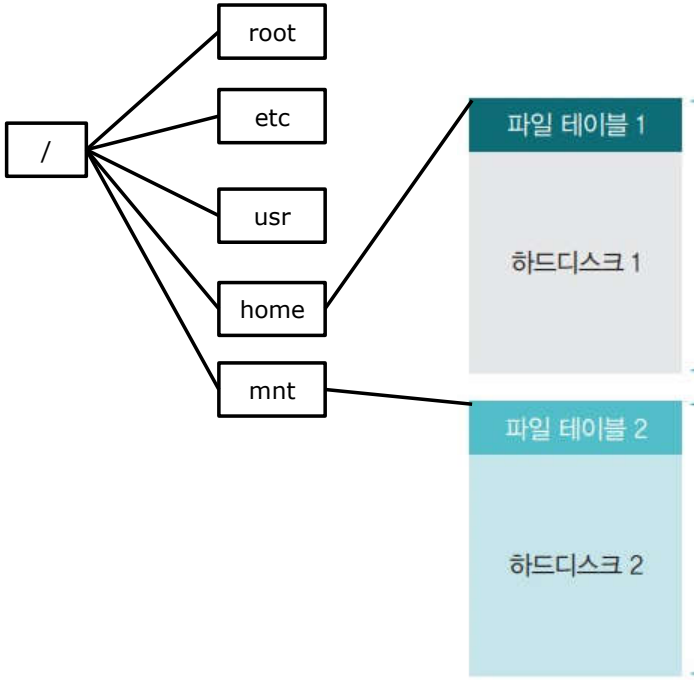
- 사용 가능
  - 알파벳(대소문자 구분), 한글, 숫자, 하이픈(-), 밑줄(_), 점(.)
  - 기타: 공백, *, &, |, 등의 특수 문자
- 사용 불가
  - /

# 3. 디렉터리 (6/6)



## ❑ 마운트(mount)

- 현재 파일 시스템에 다른 파일시스템을 연결하는 것
- 유닉스 운영체제에서 여러 개의 파티션을 하나의 파일시스템의 하위 폴더에 연결



# 4. 파티션



## ❑ 정의

- 디스크를 논리적인 구역으로 분할
- 파티션 하나에 하나의 파일 시스템이 탑재
- 대용량 하드디스크의 경우 여러 개로 나누어 사용하면 관리하기가 편함
- 여러 개의 하드디스크를 하나의 파티션으로 통합하여 사용하기도 함

# 5. 디스크 장치 관리 (1/2)

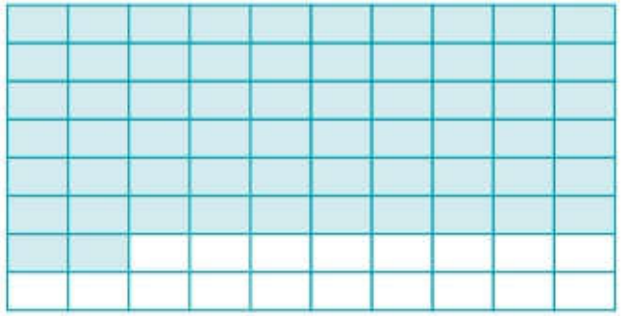


## ❑ 포멧팅(formatting)

- 디스크에 파일 시스템을 탑재
  - 디스크 표면을 초기화
  - 빈 저장장치에 파일 테이블을 탑재

| 파일 A | 1, 3, 9 |
|---|---|
| 파일 B | 4, 2 |
| 파일 C | 13 |
| 파일 D | 15, 4 |
| 파일 E | 23, 7 |

파일 테이블



저장장치

## ❑ 빠른 포매팅과 느린 포매팅

- 빠른 포매팅 :
  - 파일 테이블만 초기화
  - 데이터 복구 가능
- 느린 포매팅 :
  - 파일 시스템을 탑재
  - 디스크 표면을 초기화
  - 데이터 복구 불가

# 5. 디스크 장치 관리 (2/2)

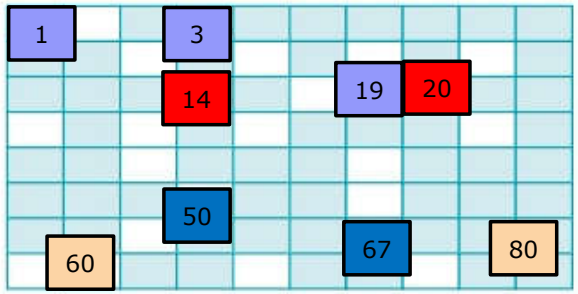
인덕대학교

## ❑ 조각모음

- 하드디스크의 분산된 섹터에 저장된 데이터들에 대하여 재정렬하는 과정
- 동일 파일 블록에 대하여 연속된 섹터에 할당
- 데이터 접근이 빨라짐

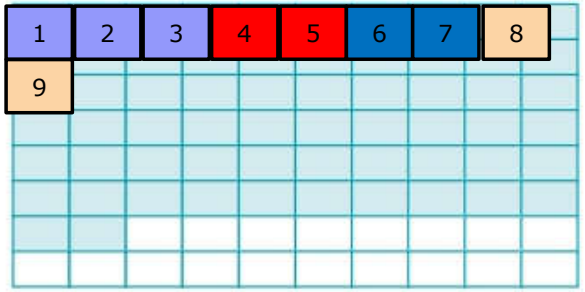
[파일 테이블 A]

| 파일 | 블록 ID |
|---|---|
| 파일A | 1,3,19 |
| 파일B | 14,20 |
| 파일C | 50,67 |
| 파일D | 80,60 |




[파일 테이블 B]

| 파일 | 블록 ID |
|---|---|
| 파일A | 1,2,3 |
| 파일B | 4,5 |
| 파일C | 6,7 |
| 파일D | 8,9 |



수고하셨습니다.