

# Computer File Systems

소프트웨어융합학부

노기섭 교수

(kafa46@cju.ac.kr)

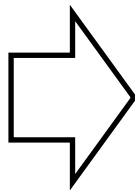
# 컴퓨터에 정보를 저장하는 방법

# 정보의 저장

## ■ 사람이 정보를 저장하는 방법

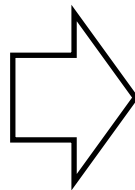


이미지 출처: <https://bit.ly/3Sh8yS4>



이미지 출처: <https://m.dongascience.com/news.php?idx=15092>

## ■ 컴퓨터가 정보를 저장하는 방법?



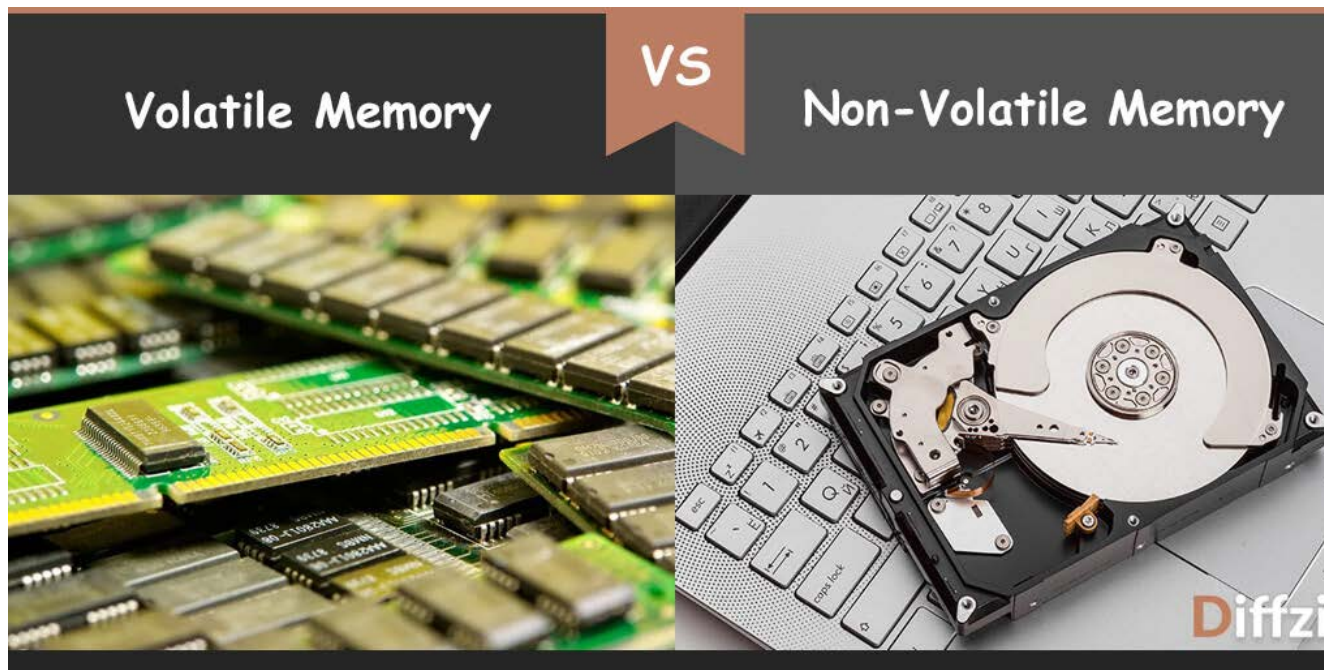
본체 어딘가  
저장됨

이미지 출처: <https://m.blog.naver.com/onluv4u/221305500464>

# 컴퓨터 저장 장치 구분

## ■ 컴퓨터 저장 장치 구분

- 휘발성(volatile) 메모리: 전원이 꺼지면 즉시 사라짐
- 비휘발성(non-volatile) 메모리: 전원이 꺼져도 정보 유지



이미지 출처: <https://diffzi.com/volatile-memory-vs-non-volatile-memory/>

# 파일의 저장 형태

- 연관된 정보를 묶어서 '파일(File)' 형태로 저장
- 실제 컴퓨터에 File로 저장된 내용은 바이너리(Binary) 형태
  - 바이너리: 0과 1로 정보를 표현한 것

```
00000000 00000000 00000000 11101000 10110111 01000010 00000000
10001011 01010100 00100100 00010000 01101000 01101000 00001101
00000000 01010010 10001101 01000100 00100100 01110000 01010000
01110011 01001010 00000000 00000000 11000110 10000100 00100100
00000000 00000000 00000000 00001001 01010111 10001101 01001100
01101100 11101000 10010001 01000010 00000000 00000000 10001011
00100100 00010000 01010001 10001101 01010100 00100100 01001000
11101000 00110010 01000001 00000000 00000000 11000110 10000100
11011000 00000000 00000000 00000000 00001010 10111001 00000000
01000011 00000000 11101000 00110000 01100010 00000000 00000000
10001101 01001100 00100100 01001000 11101000 10000110 00111010
00000000 10001011 01000100 00100100 00010000 01101000 10100100
01000011 00000000 01010000 10001101 01001100 00100100 01011000
11101000 00100010 01001010 00000000 00000000 11000110 10000100
11011000 00000000 00000000 00000000 00001011 01010011 10001101
00100100 01010100 11101000 01000000 01000010 00000000 00000000
```

# 컴퓨터 파일

## ■ 비휘발성 메모리에 저장되는 정보의 형태

- 주로 HDD, USB, SSD, CD 등에 저장

## ■ 컴퓨터 파일의 종류

- 텍스트 파일, 음악 파일, 사진 파일, 동영상 파일, ...



- 컴퓨터 파일은 종류에 따라 각각 파일 포맷(Format)을 가짐
  - 누구라도 파일 포맷을 만들 수 있으나,
  - 대부분은 표준화된 포맷을 사용 (예: jpeg, mp3, mp4 등)

# Txt 파일

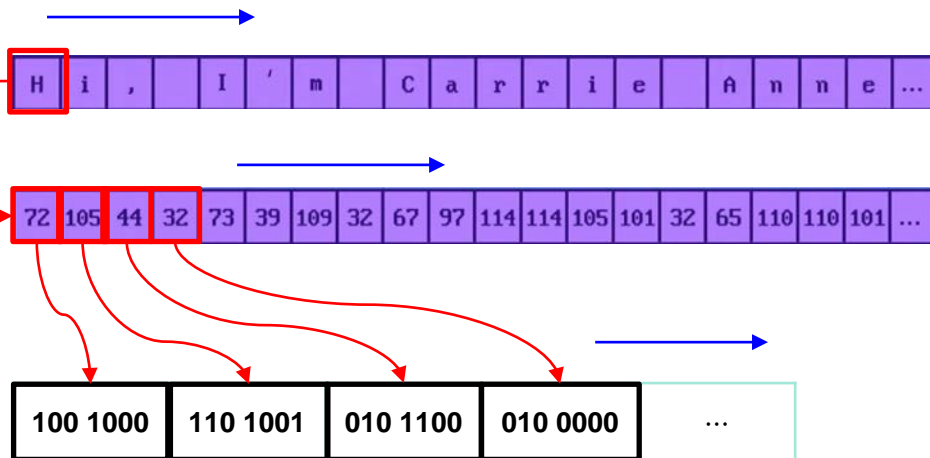
## ■ Txt 파일: 글자(Text)로 표현된 정보를 저장한 파일

- 글자 정보를 바이너리로 표현하는 방법

- 글자 코드표 (ASCII 코드, Unicode 등) 활용

### ASCII TABLE

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	\$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(	72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29	)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[END OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[	123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D	]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

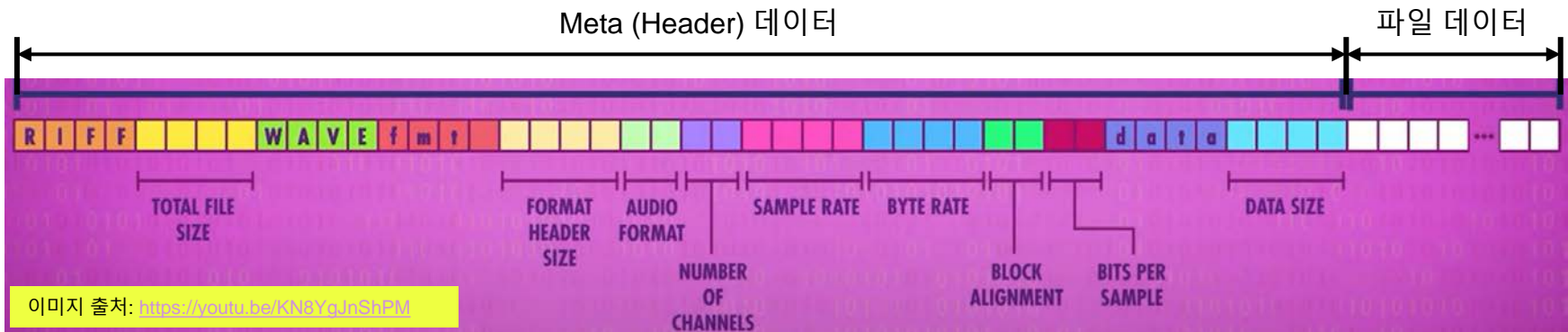




# Wave 파일 (1/2)

## ■ Wave 파일: 소리로 표현된 정보를 저장한 파일

- 소리 정보를 바이너리로 표현하는 방법
  - Txt 파일과 다르게 메타 데이터가 필요
  - 메타(Meta) 데이터 → 데이터를 설명하는 데이터, 파일 헤더라고 부르기도 함



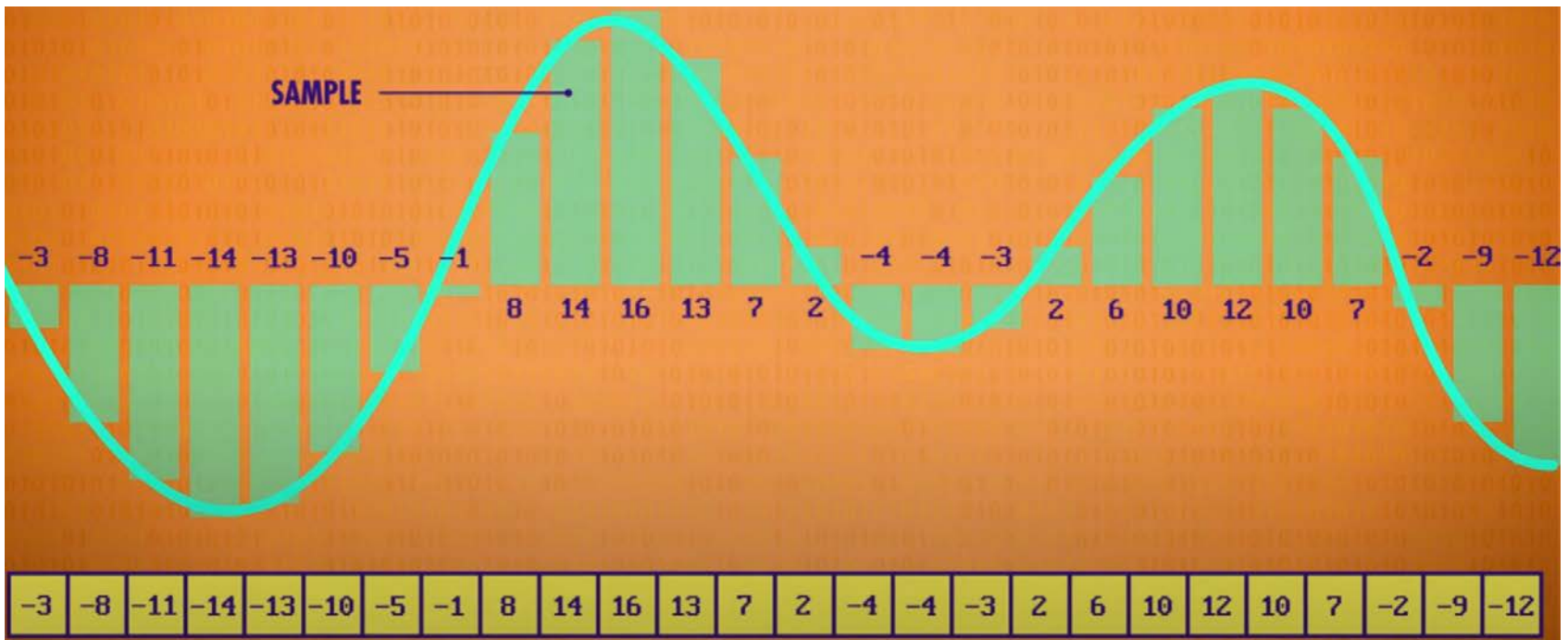
Offset (d)	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	
00000000	52	49	46	46	BC	3E	65	02	57	41	56	45	66	6D	74	20	RIFF4>e.WAVEfmt
00000016	10	00	00	00	01	00	02	00	44	AC	00	00	10	B1	02	00	.....D~...±..
00000032	04	00	10	00	64	61	74	61	98	3E	65	02	00	00	00	00	....data~>e....
00000048	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
00000064	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....



# Wave 파일 (2/2)

## ■ 음성을 숫자로 변환하는 방법

- Sampling Quantization



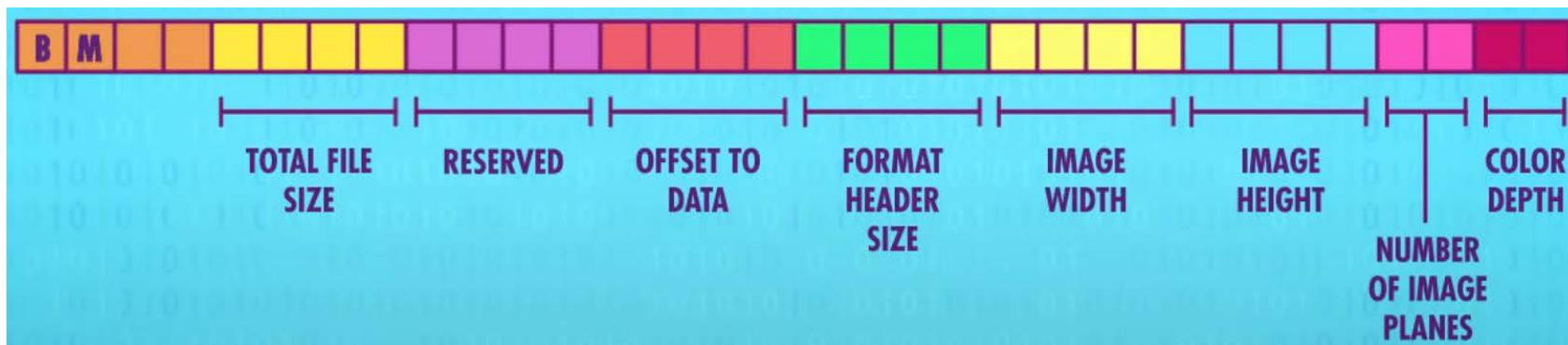
이미지 출처: <https://youtu.be/KN8YgJnShPM>

# 이미지 파일 (1/2)

## ■ Bit Map or BMP

- Image: 'pixel' 조합 (combination)
- Pixel: 3색 (red, green, blue) 정보를 조합하여 최종 color 생성 → 디스플레이 시현

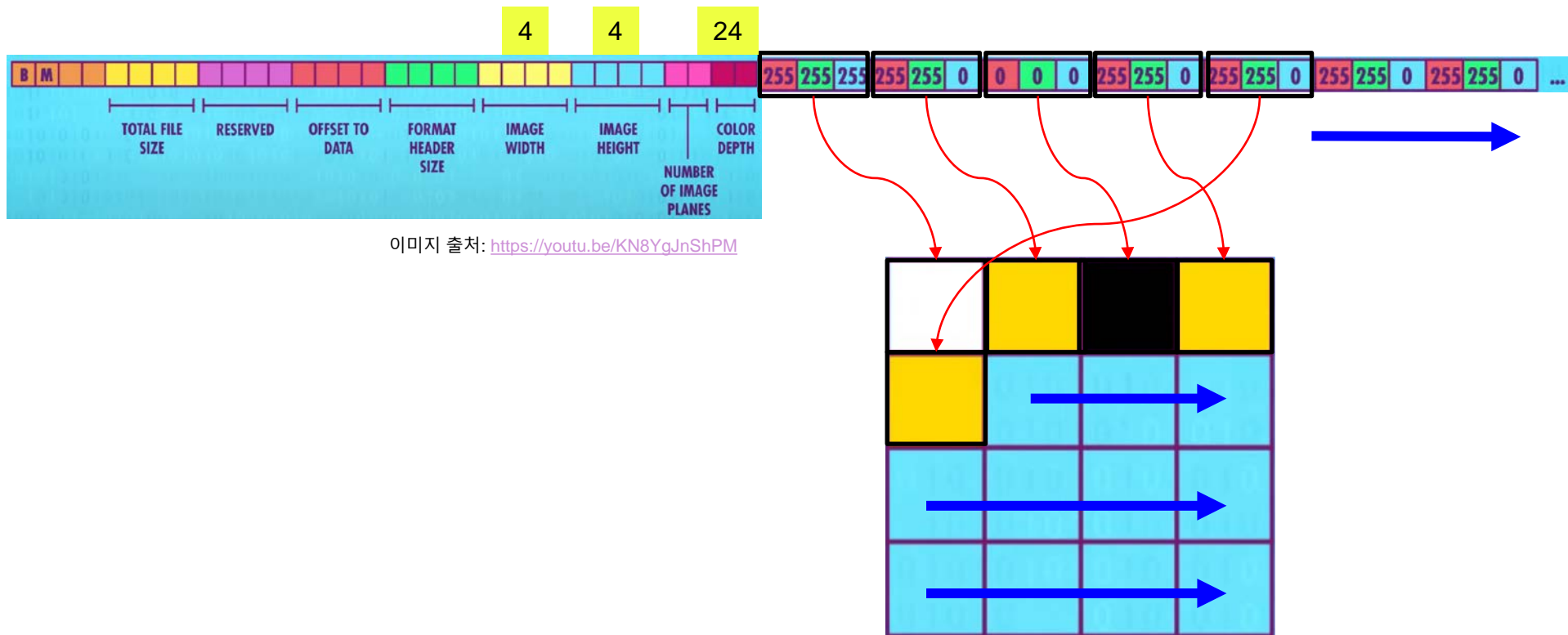
## ■ Bit Map 파일: 메타 데이터 (헤더)



이미지 출처: <https://youtu.be/KN8YgJnShPM>

# 이미지 파일 (2/2)

## ■ Bit Map 파일: 데이터



이미지 출처: <https://youtu.be/KN8YgJnShPM>

# One File System

# File 저장 방법 - One File System

## ■ 초기의 파일 저장

- 하나의 스토리지 → 하나의 파일로 처리

(예) 충격 실험



이미지 출처: <https://tenor.com/view/kamehameha-lee-seung-gee-yuna-kim-gif-11264675>

result.txt



...

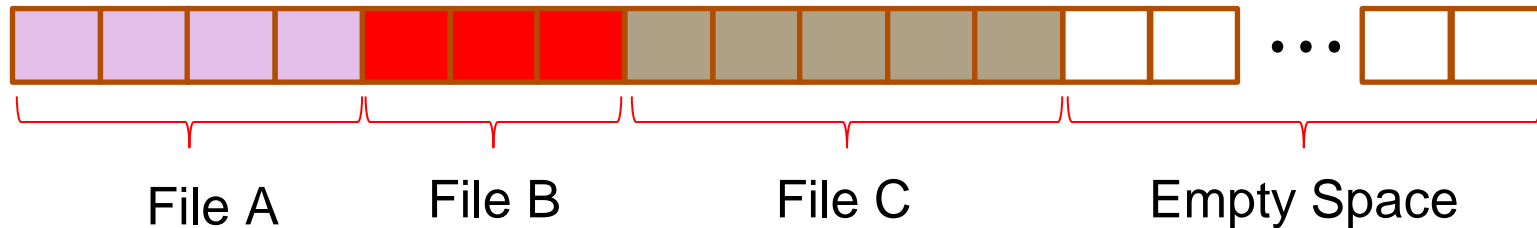
# Flat File System



# File 저장 방법 - Flat File System (1/3)

## ■ 하나의 스토리지에 여러 파일 저장

- 하나의 파일 마지막에서 다른 파일 시작



파일이 어디서 시작해서 어디서 끝나는지 알아야 됩니다.  
→ Solution: Directory File 추가

Directory File



# File 저장 방법 - Flat File System (2/3)

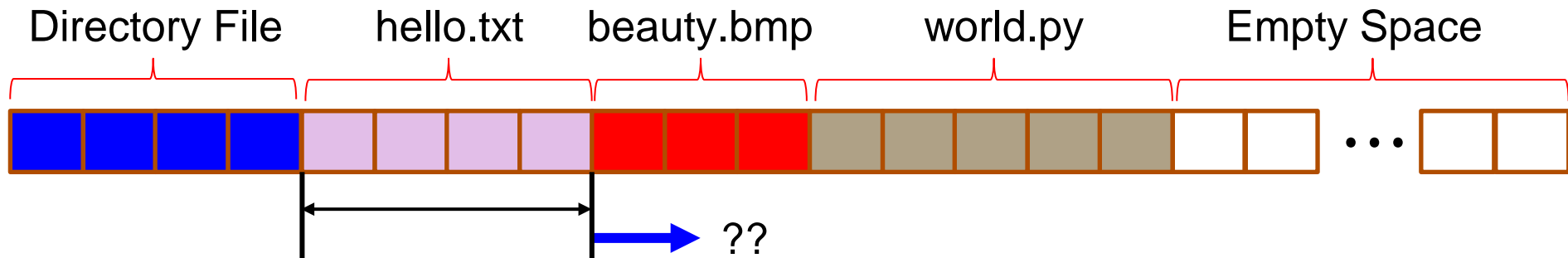
## ■ Directory File 메타 데이터

Name	Created	Last Modified	Owner	Read/Write/Execution	Begin	Length
hello.txt	09:14 2/28/2021	09:14 2/28/2021	홍길동	R W -	10	8
beauty.bmp	21:28 3/9/2022	21:28 3/14/2022	이순신	R - -	18	13
world.py	13:28 7/30/2020	13:53 7/30/2020	홍길동	R W X	31	6
tutorial.docx	15:42 3/9/2022	15:42 3/10/2022	강감찬	R W -	37	8
⋮	⋮	⋮	⋮	⋮	⋮	⋮

# File 저장 방법 - Flat File System (3/3)

## ■ Flat File System 단점

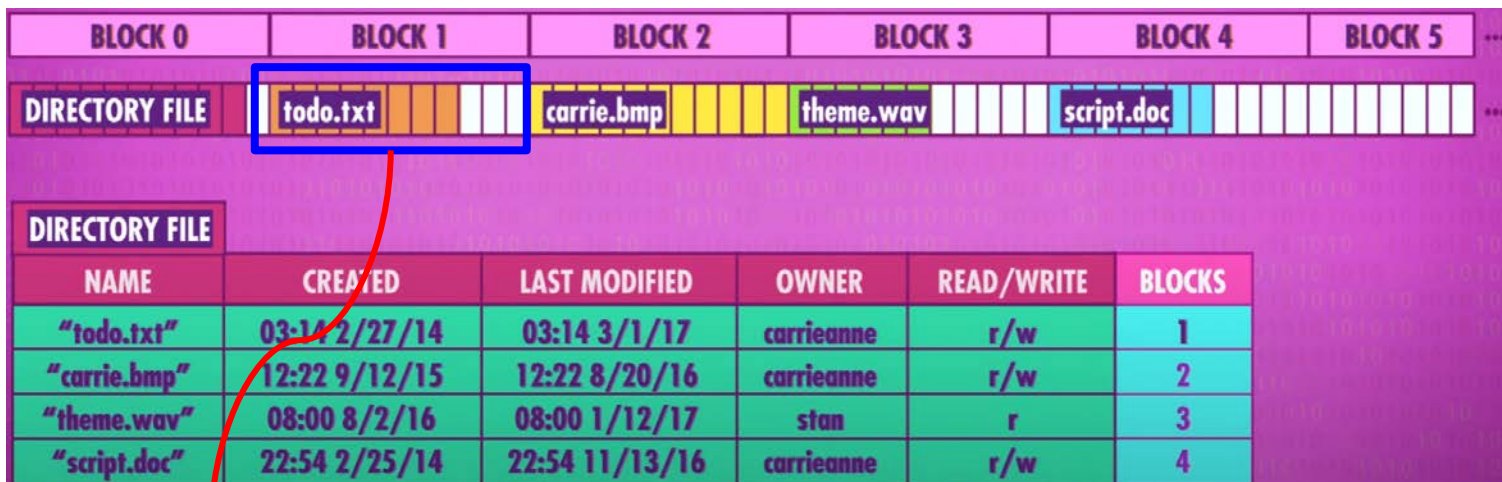
- 기존 파일을 수정하기 어려움



파일에 내용을 추가할 경우 다른 파일을 모두 조정해야 함  
→ solution: 블록을 할당하여 정보 저장

# File 저장 방법 - Flexible File System (1/2)

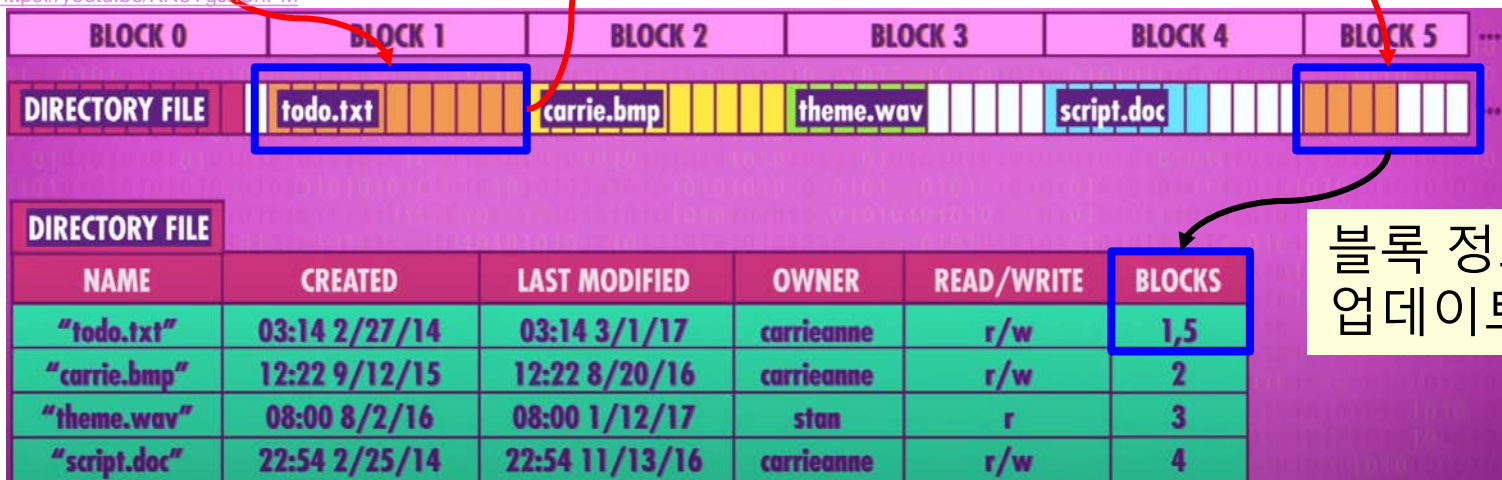
이미지 출처: <https://youtu.be/KN8YgJnShPM>



내용추가

부족하면 다른 블록에 기록

이미지 출처: <https://youtu.be/KN8YgJnShPM>



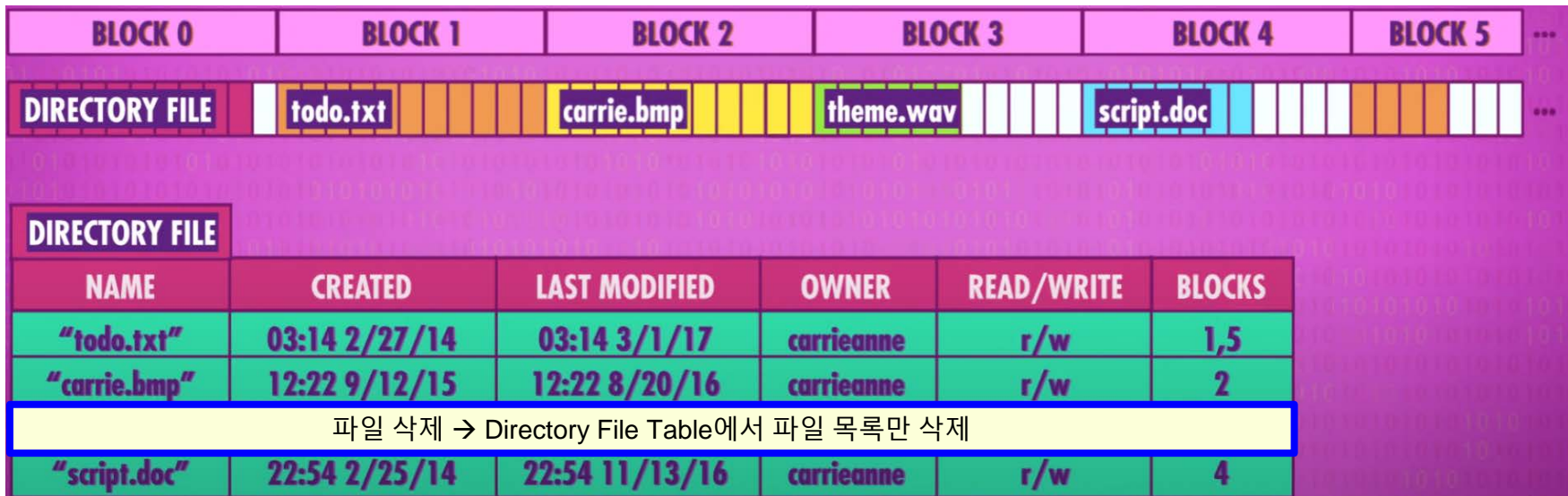
블록 정보  
업데이트

# File 저장 방법 - Flexible File System (2/2)

## ■ 파일 삭제

- Directory File 목록에서 삭제
- 블록 안에 있는 데이터는 그대로 유지
- Free 상태에 있는 블록은 다른 파일의 쓰기 공간이 필요할 경우 덮어씀

→ 완전삭제 하더라도 컴퓨터 포렌식을 통해 살려낼 수 있는 이유이기도 함



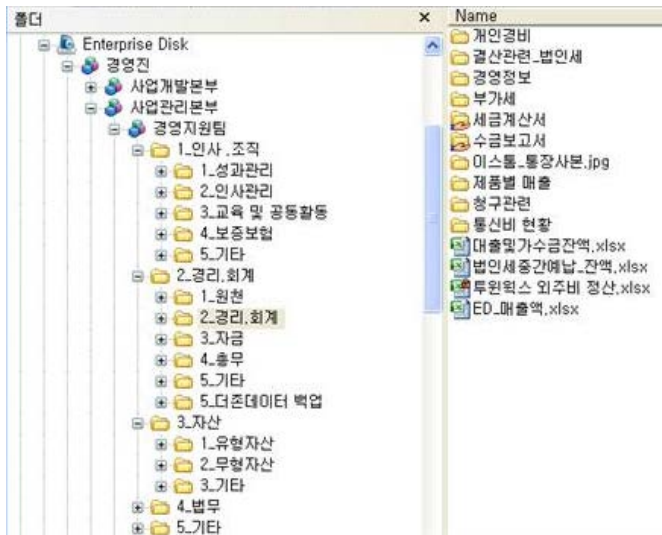
이미지 출처: <https://youtu.be/KN8YgJnShPM>

# Hierarchical File System



# Hierarchical File System – idea

In real world



Root Directory 파일을 만들고,  
모든 Directory, File은 Root Directory 아래에 배치

이미지 출처: <https://youtu.be/KN8YgJnShPM>

DIR:"root"				
NAME	IS DIRECTORY	CREATED	LAST MODIFIED	BLOCKS
"todo.txt"	no	03:14 2/27/14	03:14 3/1/17	1,2,3
"theme.wav"	no	08:00 8/2/16	08:00 1/12/17	5
"script.doc"	no	22:54 2/25/14	22:54 11/13/16	4
"music"	yes	7:01 3/4/13	08:22 5/21/17	6
"photos"	yes	13:55 3/5/14	09:20 4/18/17	8

이미지 출처: <https://youtu.be/KN8YgJnShPM>

DIR:"music"				
NAME	IS DIRECTORY	CREATED	LAST MODIFIED	BLOCKS
"beat.mp3"	no	03:14 2/27/14	03:14 3/1/17	9,11,25
"believe.wav"	no	08:00 8/2/16	08:00 1/12/17	13,37,23
"royals.mp3"	no	22:54 2/25/14	22:54 11/13/16	24,26,27,28
"magic.aiff"	no	7:01 3/4/13	08:22 5/21/17	19
"breathe.mp3"	no	13:55 3/5/14	09:20 4/18/17	20,29,30

복잡한 디렉토리 구조

# Hierarchical File System - Advantage

## ■ Advantages of Hierarchical File System

- Unlimited Depth of Directory Structure
- Ease of Management: 예) “theme.wav” 파일을 “music” 디렉토리로 옮기는 경우

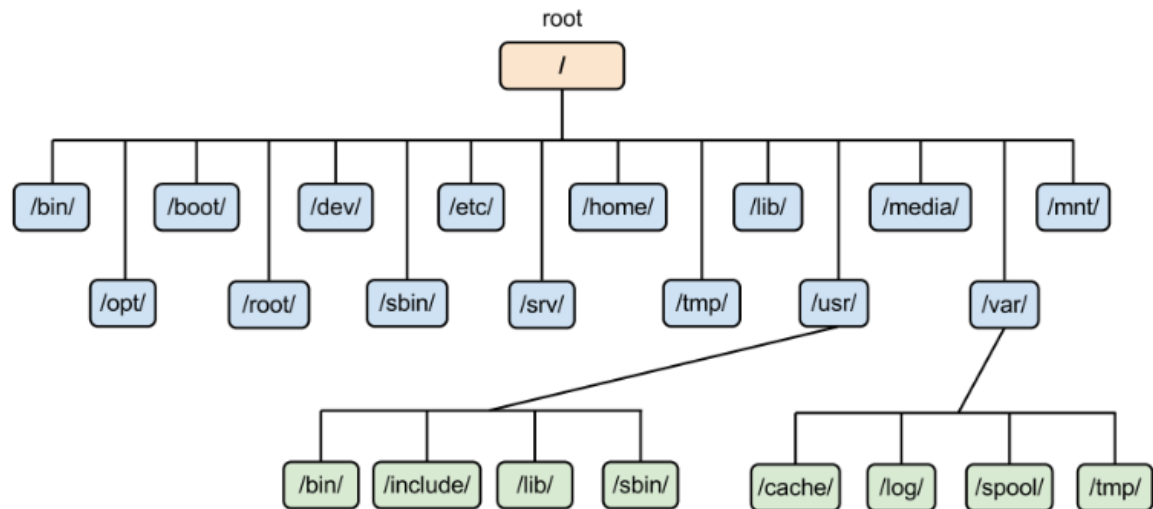
이미지 출처: <https://youtu.be/KN8YqJnShPM>

DIR: "root"				
NAME	IS DIRECTORY	CREATED	LAST MODIFIED	BLOCKS
"todo.txt"	no	03:14 2/27/14	03:14 3/1/17	1,2,3
파일 삭제 → Directory File Table에서 파일 목록만 삭제				
"script.doc"	no	22:54 2/25/14	22:54 11/13/16	4
"music"	yes	7:01 3/4/13	08:22 5/21/17	6
"photos"	yes	13:55 3/5/14	09:20 4/18/17	8

이미지 출처: <https://youtu.be/KN8YqJnShPM>

DIR: "music"				
NAME	IS DIRECTORY	CREATED	LAST MODIFIED	BLOCKS
"beat.mp3"	no	03:14 2/27/14	03:14 3/1/17	9,11,25
"believe.wav"	no	08:00 8/2/16	08:00 1/12/17	13,37,23
"royals.mp3"	no	22:54 2/25/14	22:54 11/13/16	24,26,27,28
"magic.aiff"	no	7:01 3/4/13	08:22 5/21/17	19
"breathe.mp3"	no	13:55 3/5/14	09:20 4/18/17	20,29,30
"music"	yes	7:01 3/4/13	08:22 5/21/17	6

# 리눅스 파일 구조



디렉토리 명	설명
home	사용자 홈 디렉토리가 생성되는 곳입니다.
media	CD_ROM이나 USB같은 외부 장치를 연결하는 디렉토리입니다.
opt	추가 패키지가 설치되는 디렉토리입니다.
dev	장치파일들이 저장되어 있는 디렉토리입니다.
root	root계정의 홈 디렉토리입니다. (/ 디렉토리와는 다릅니다.)
sys	리눅스 커널관련 정보가 있는 디렉토리입니다.
usr	기본 실행파일과 라이브러리 파일, 헤더 파일등의 파일이 저장되어있는 디렉토리입니다.
boot	부팅에 필요한 정보를 가진 파일들이 있는 디렉토리입니다.

디렉토리 명	설명
var	시스템 운영중에 발생한 데이터와 로그가 저장되는 디렉토리입니다.
tmp	시스템 사용중에 발생한 임시데이터가 저장됩니다. (부팅 시 초기화)
srv	FTP나 Web등 시스템에서 제공하는 서비스의 데이터가 저장되는 디렉토리입니다.
run	실행중인 서비스와 관련된 파일이 저장되는 디렉토리입니다.
proc	프로세스 정보 등 커널 관련 정보가 저장되는 디렉토리입니다.
mnt	파일 시스템을 임시로 연결하는 디렉토리입니다.
etc	리눅스 설정을 위한 각종 파일들을 가지고 있는 디렉토리입니다.

자료 출처: <https://coding-factory.tistory.com/499>

# 리눅스 파일 종류

## ■ 리눅스에서는 모든 것이 파일!

## ■ 리눅스 파일 종류

- 일반 파일(Regular File)
  - 텍스트 파일, 실행파일, 이미지 파일 등 리눅스에서 사용하는 대부분의 파일
- 디렉토리 파일(Directory File)
  - 디렉토리도 파일로 취급
  - 해당 디렉토리에 저장된 파일, 하위 디렉토리 정보 저장
- 링크 파일(Link File)
  - 원본 파일을 다른 파일명으로 지정한 것으로 윈도우의 바로가기와 개념
  - 심볼릭 링크(Symbolic Link)와 하드 링크(Hard Link)가 있음
- 장치 파일(Device File)
  - 리눅스는 하드디스크나 마우스 같은 장치들도 파일로 취급
  - 장치파일은 이러한 장치들을 관리하기 위한 파일

소프트웨어  
꼰대 강의!



수고하셨습니다 ..^..