따라 하면서 배우는 IT

기본적으로 알아야 할 컴퓨터 기초

목차
INDEX

따라 하면서 배우는 IT

HW 구성요소

HW 구성요소
CPU

"

중앙 처리 장치
CPU

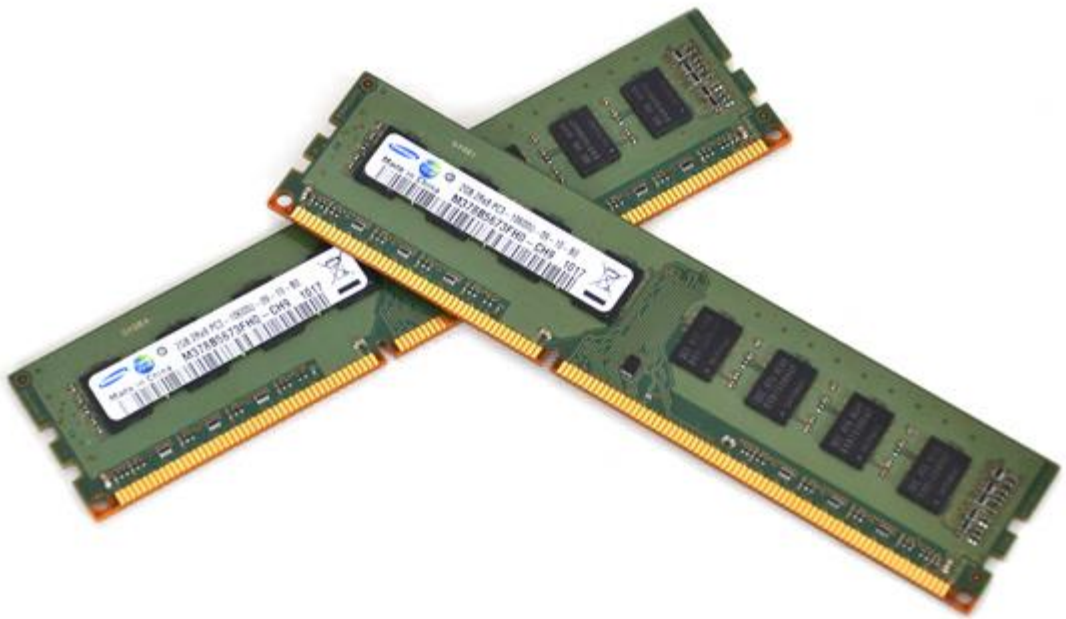
"

HW 구성요소
주 기억 장치

"

주 기억 장치
메모리
RAM

"



HW 구성요소
보조 기억 장치

"

보고 기억 장치
하드 디스크
SSD

"

HW 구성요소
입출력 장치

"

입출력 장치
I/O 장치

"

따라 하면서 배우는 IT

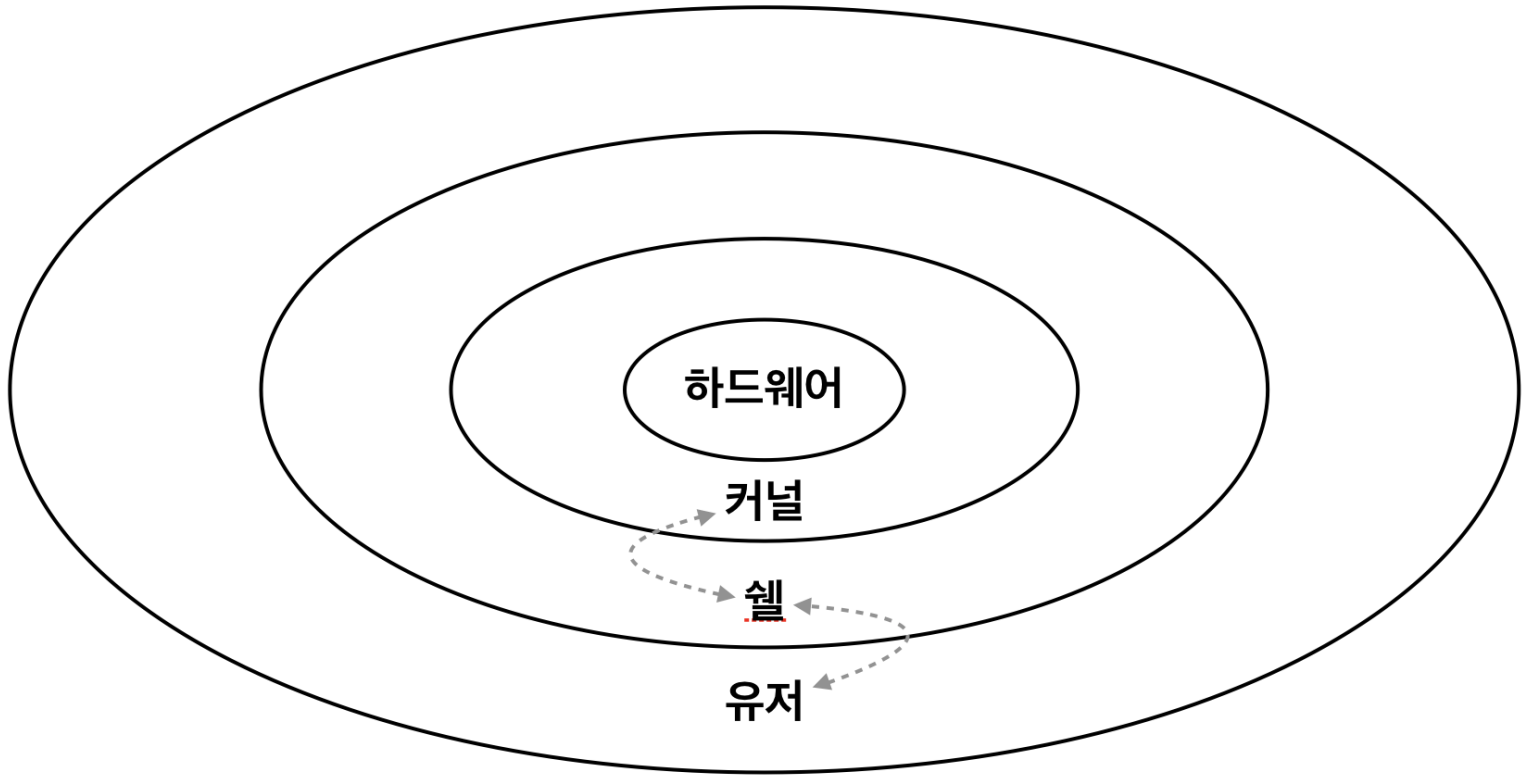
OS 운영 체제

OS 운영 체제
OS란?

"

하드웨어를 관리해주는
프로그램

"

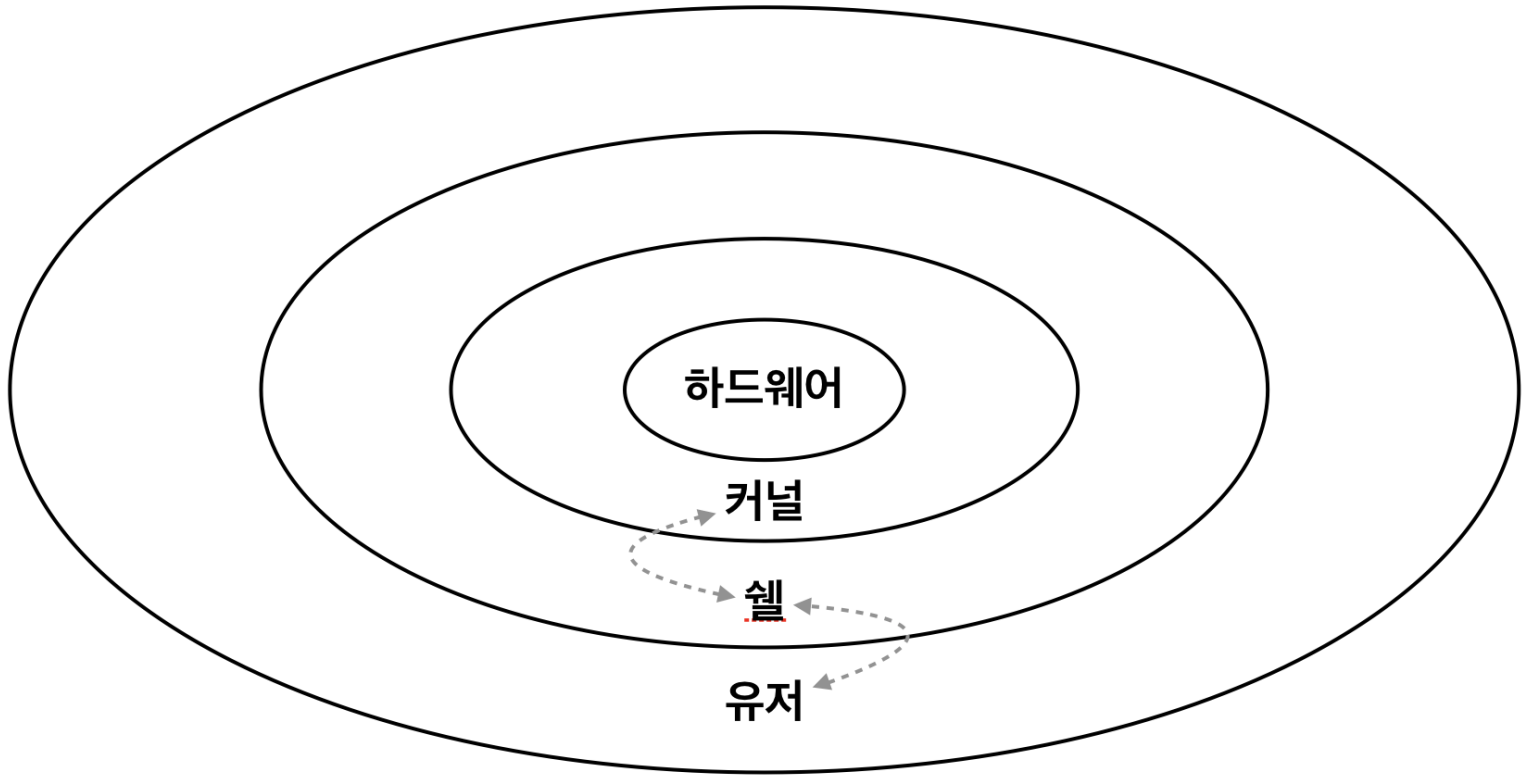


OS 운영 체제
OS의 구성요소

"

커널, 쉘, 응용 프로그램

"

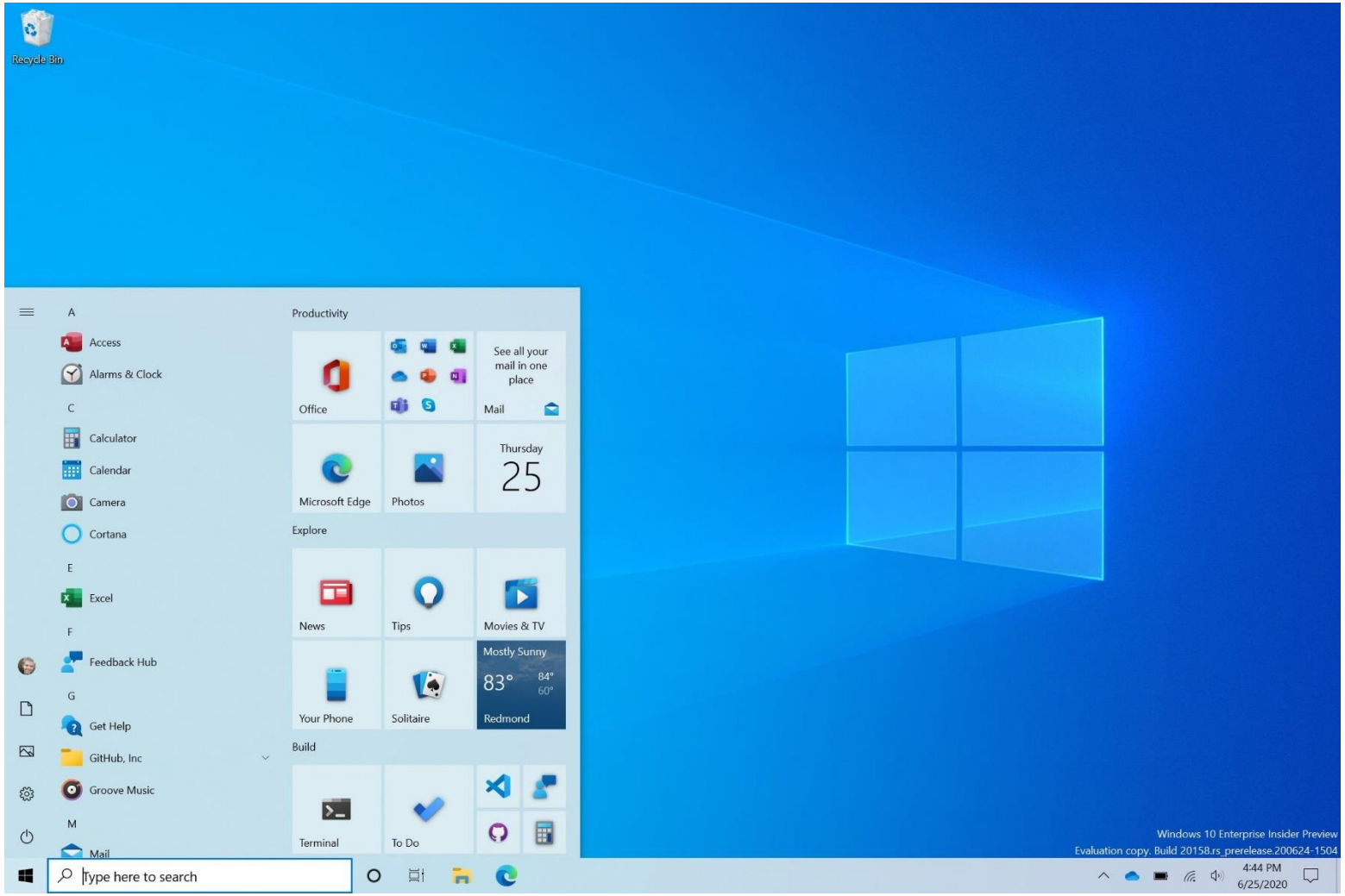


OS 운영 체제
OS의 구성요소

"

윈도우

"

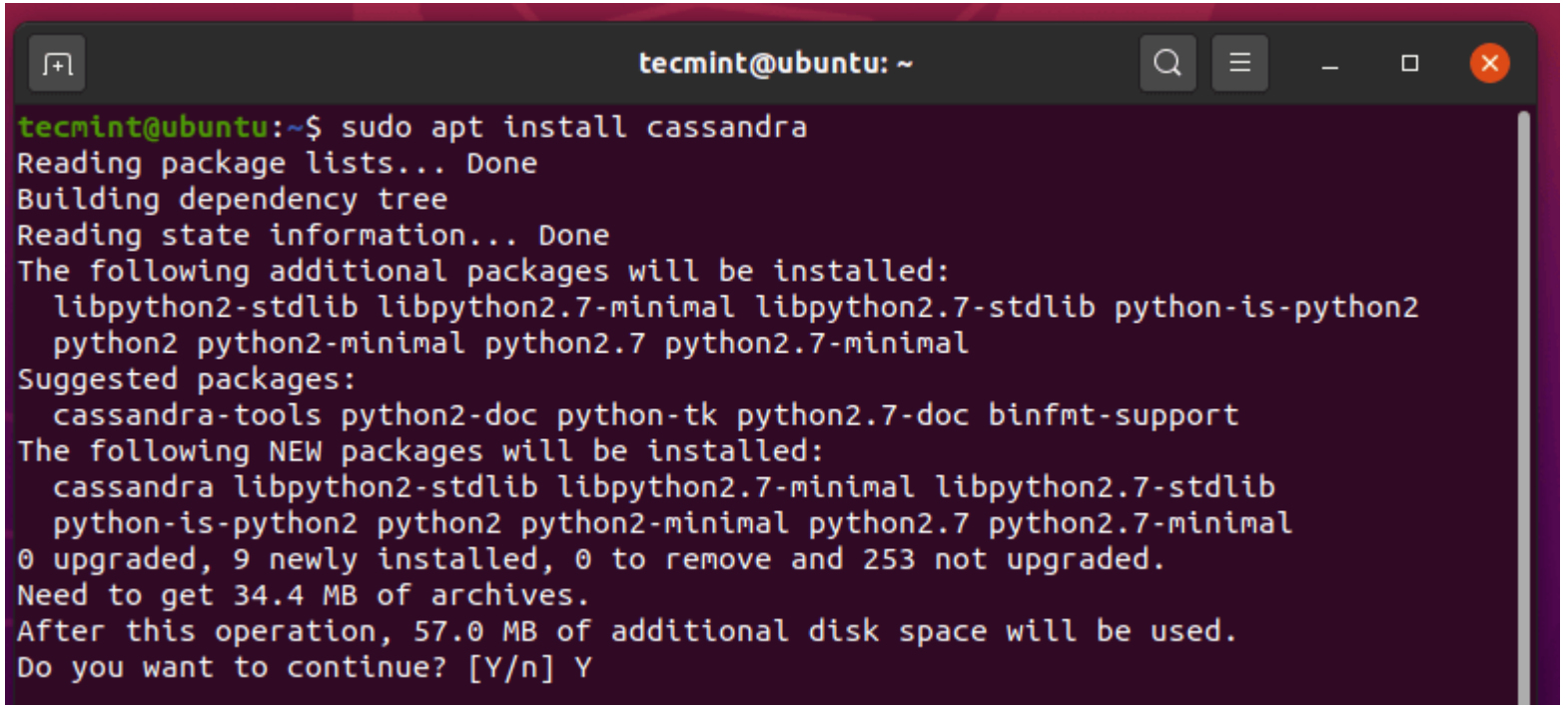


OS 운영 체제
OS의 구성요소

"

리눅스

"



따라 하면서 배우는 IT

파일의 특징

파일의 특징
데이터 표현

"

컴퓨터가 사용하는 2진법
다양한 진법
사람이 사용하는 10진법

"

- 진법 변환
  - 10진 변환

$$\mathbf{972} = 9 \times 10^2 + 7 \times 10^1 + 2 \times 10^0$$

  - 2진수 → 10진수 변환

$$\mathbf{11101}_{(2)} = 1 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$$
$$= 16 + 8 + 4 + 0 + 1$$
$$= 29$$

데이터 표현
문자 표현

"

컴퓨터가 사용하는 2진법
다양한 진법
사람이 사용하는 10진법

"

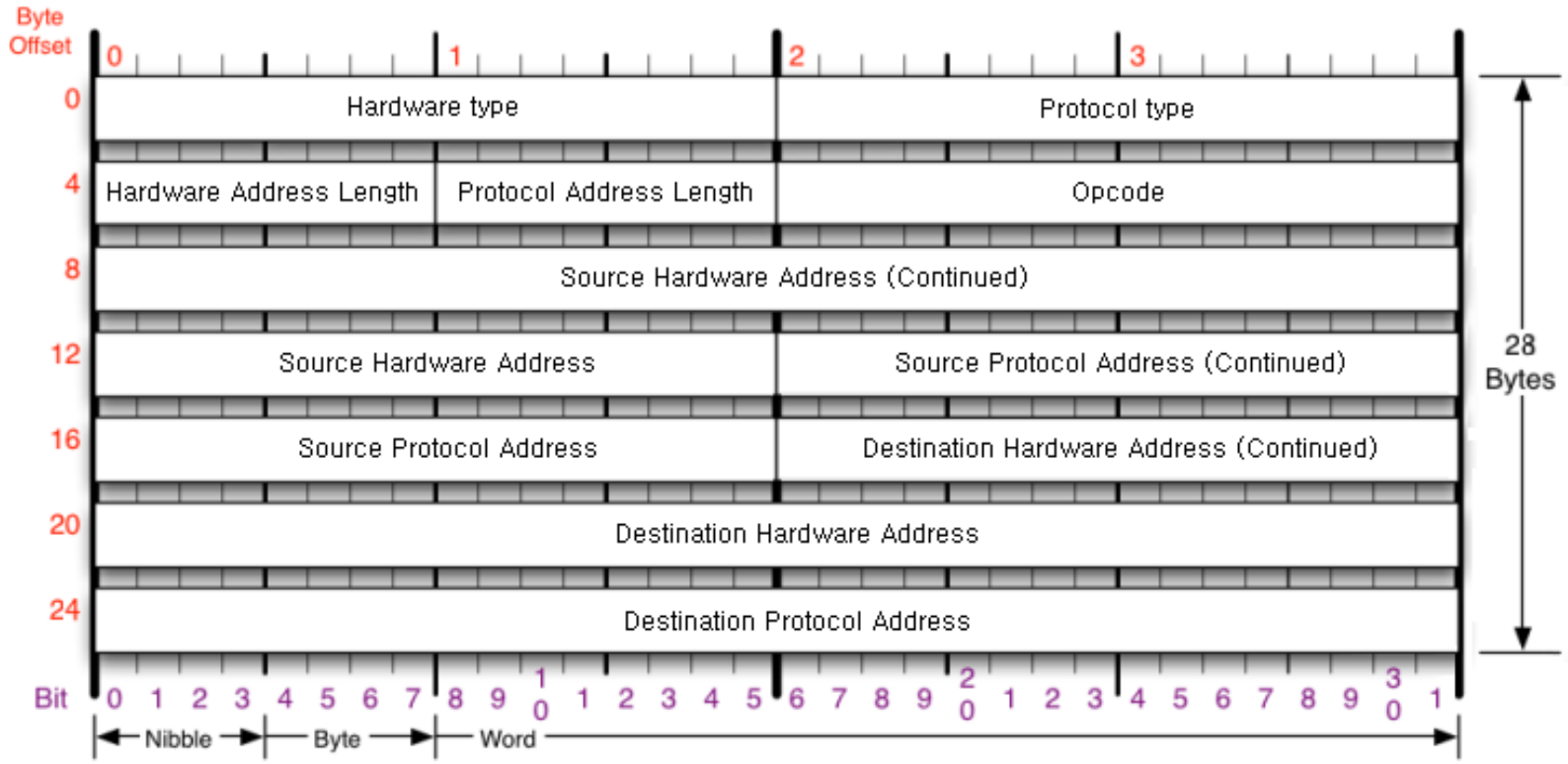
10진수	16진수	8진수	2진수	ASCII
64	0×40	100	1000000	@
65	0×41	101	1000001	A
66	0×42	102	1000010	B
67	0×43	103	1000011	C
68	0×44	104	1000100	D
69	0×45	105	1000101	E
70	0×46	106	1000110	F
71	0×47	107	1000111	G
72	0×48	110	1001000	H
73	0×49	111	1001001	I
74	0×4A	112	1001010	J
75	0×4B	113	1001011	K
76	0×4C	114	1001100	L
77	0×4D	115	1001101	M
78	0×4E	116	1001110	N
79	0×4F	117	1001111	O
80	0×50	120	1010000	P

데이터 표현
네트워크 데이터

"

컴퓨터가 사용하는 2진법
다양한 진법
사람이 사용하는 10진법

"



데이터 표현

네트워크 데이터

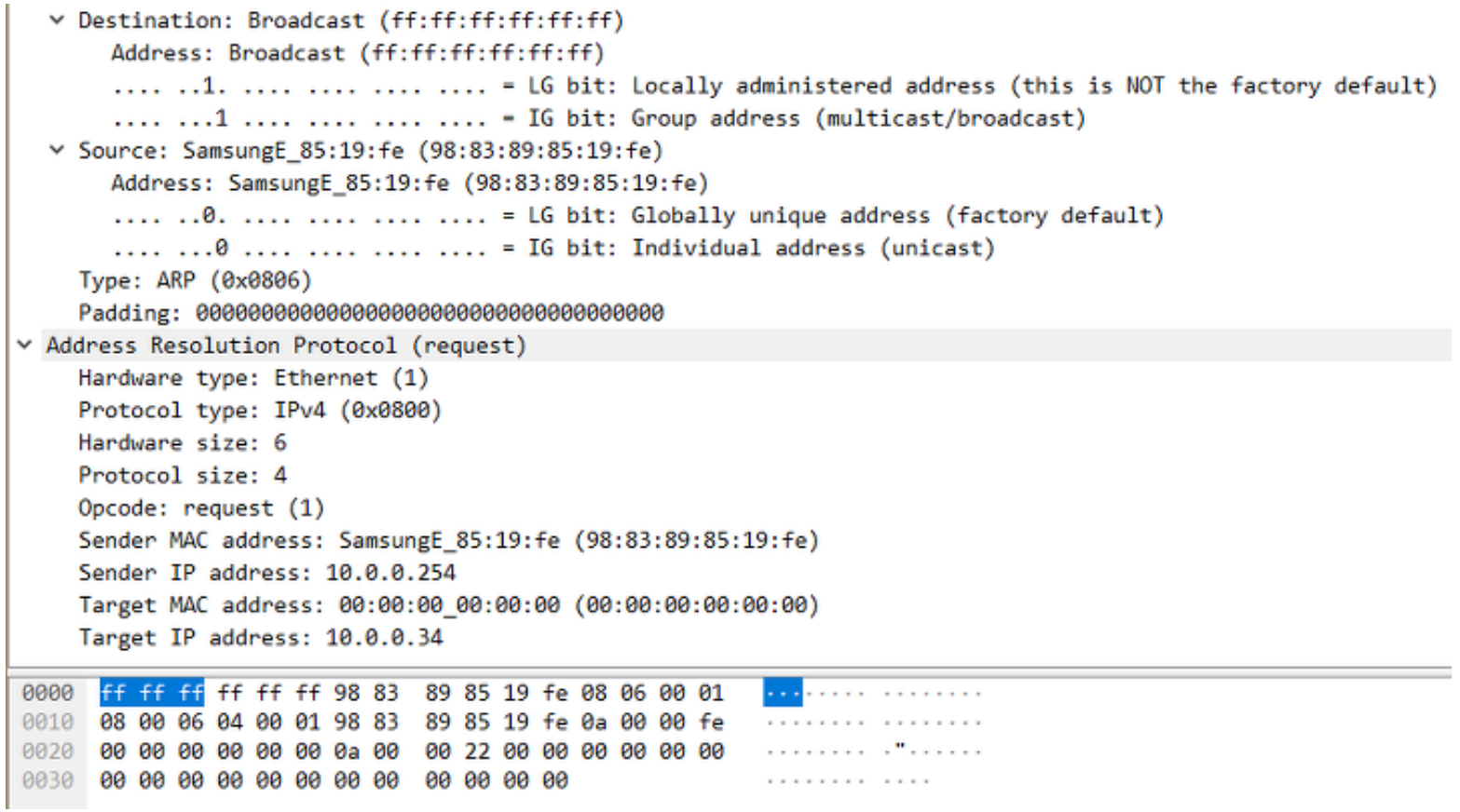
"

컴퓨터가 사용하는 2진법

다양한 진법

사람이 사용하는 10진법

"



파일의 특징
파일의 이름

"

다양한 형식의
파일의 이름

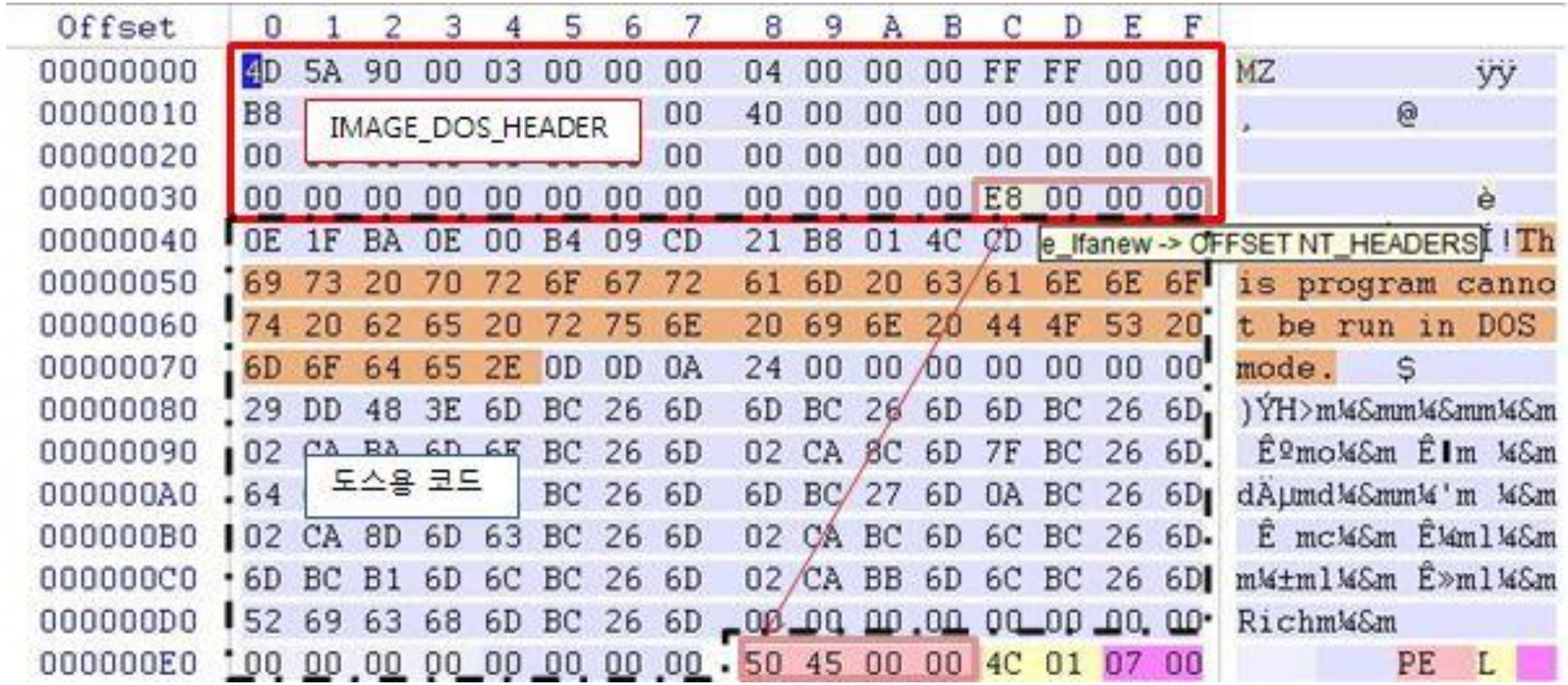
"

파일의 특징
파일의 이름

"

다양한 형식의
파일의 이름

"

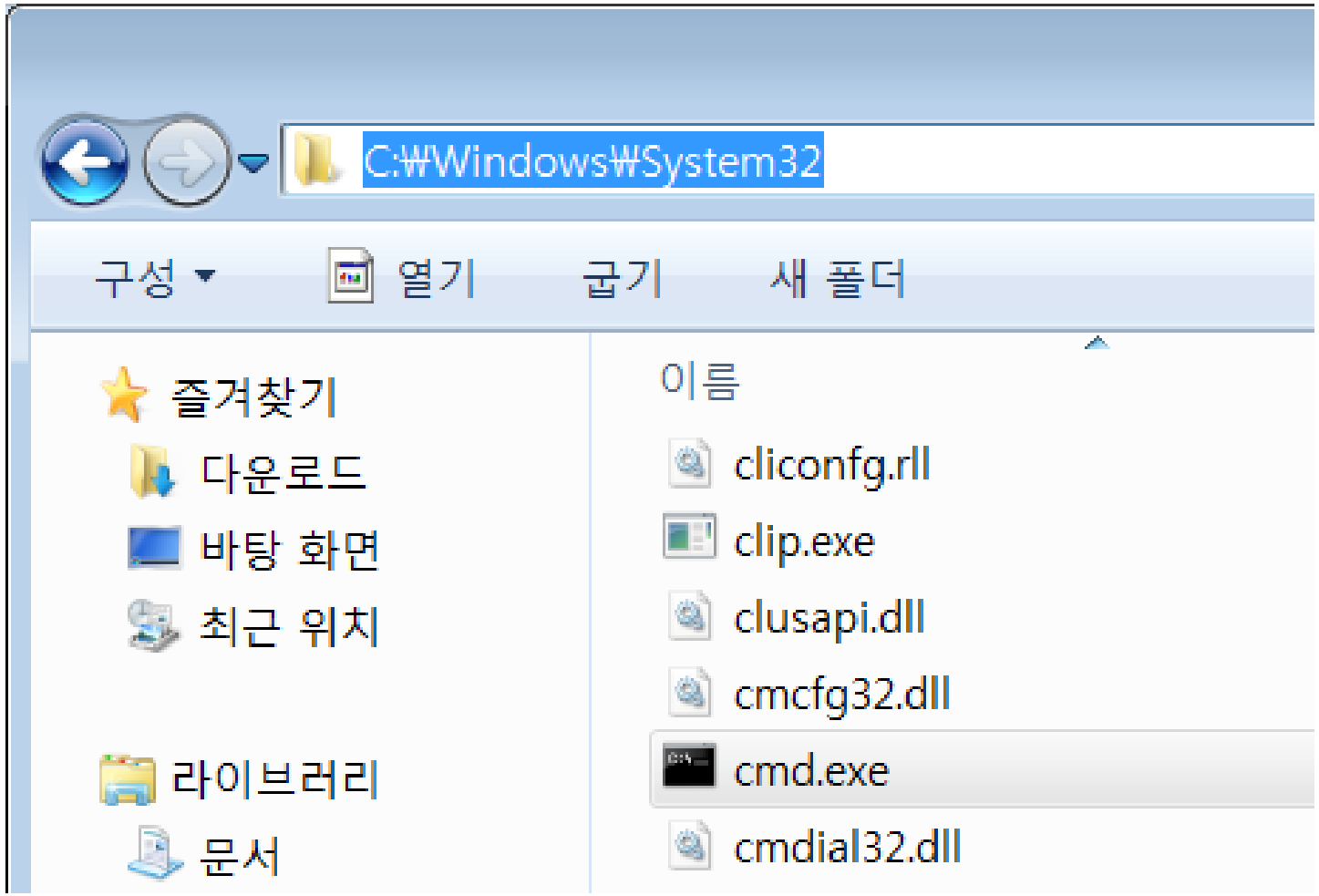


파일의 특징
파일의 경로

"

다양하게 지정할 수 있는
파일의 경로

"

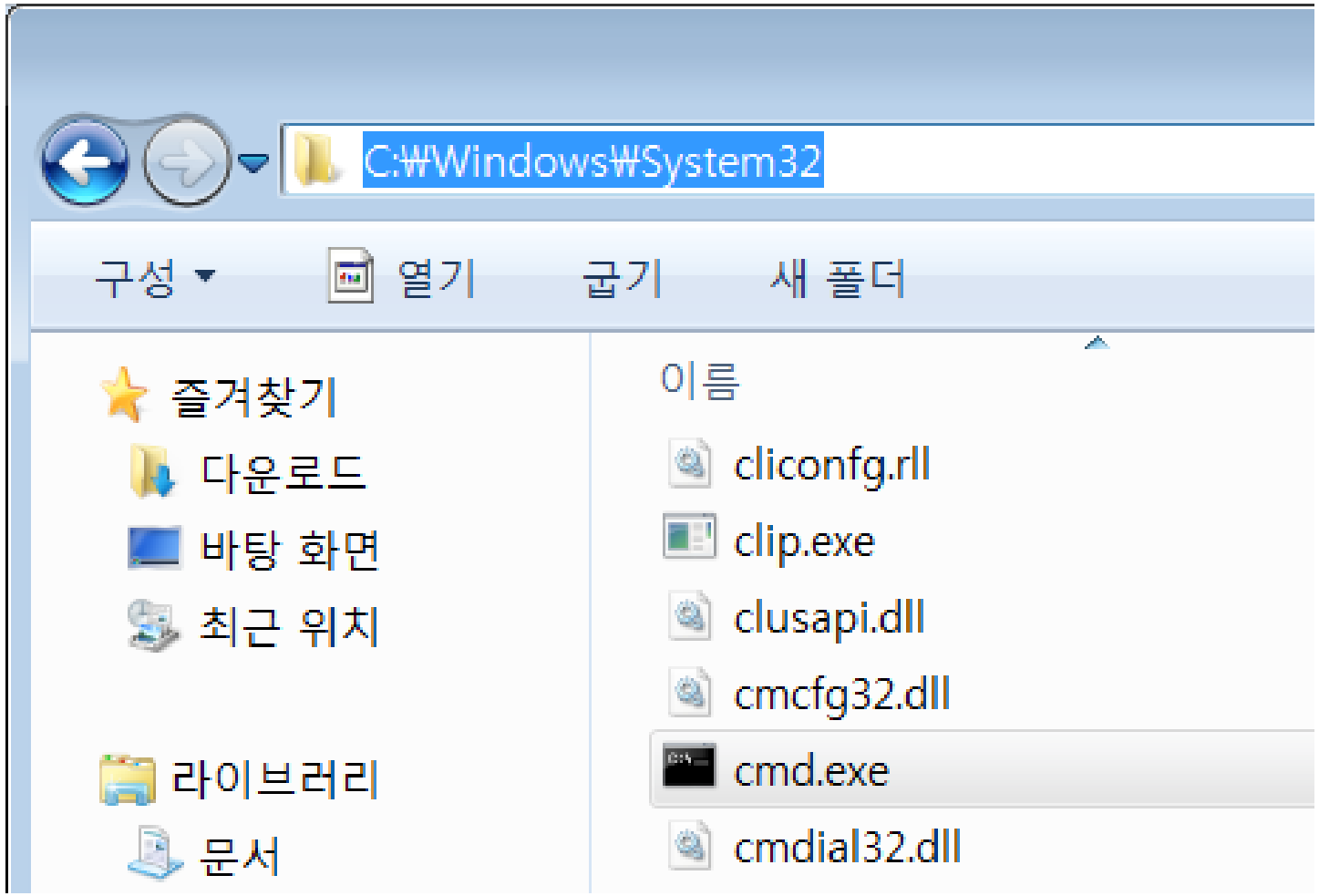


파일의 특징
파일의 경로

"

경로를 모두 입력해야하는
파일의 실행

"



따라 하면서 배우는 IT

프로그램의 특징

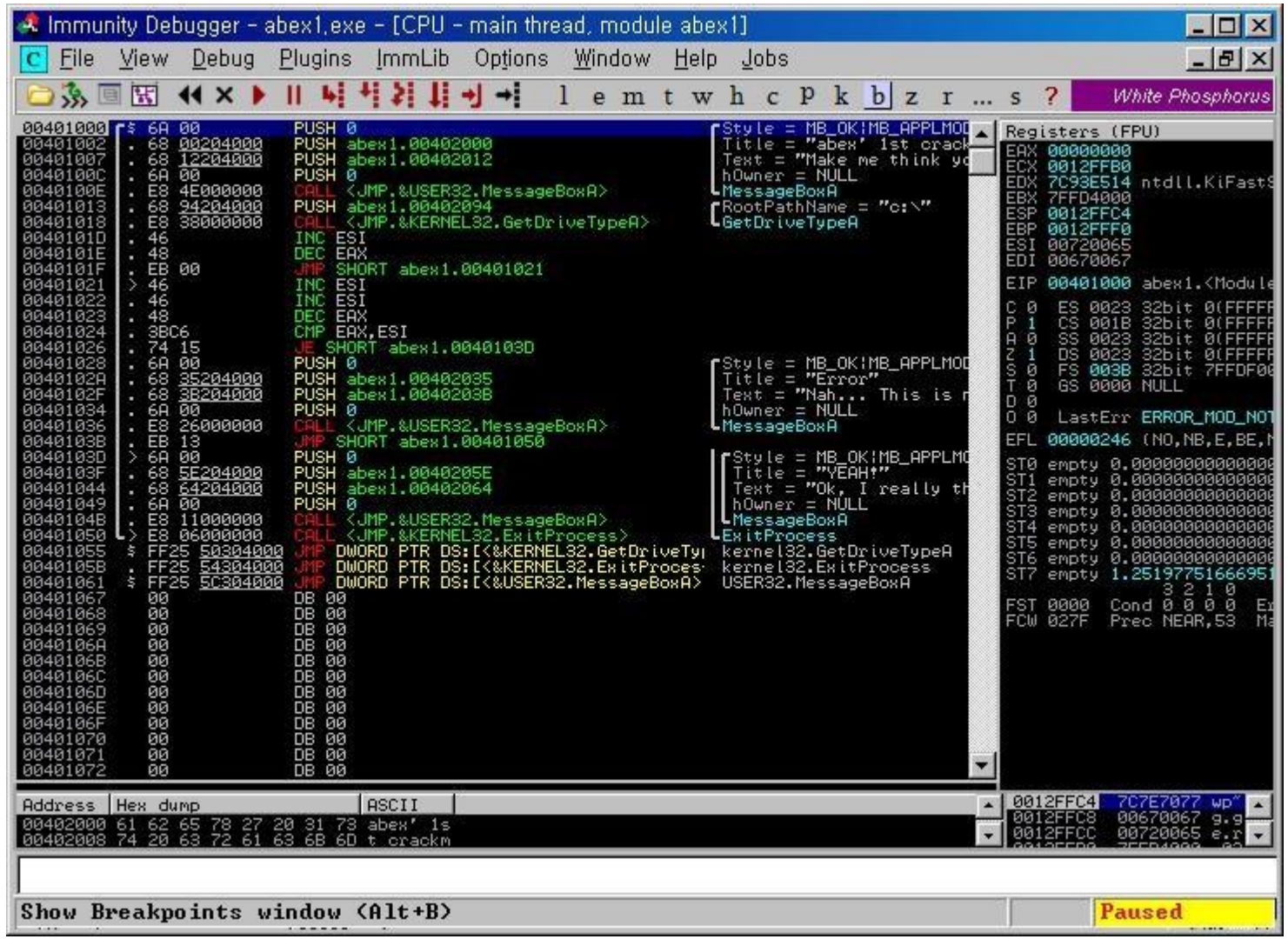
프로그램의 특징

프로그램이란?

"

다양한 작업을 수행하는
명령어의 집합

"



프로그램의 특징

프로그램을 만들려면?

"

특정 프로그래밍 언어를 활용
명령어의 집합
을 만든다

"

```
def add5(x):
    return x+5

def dotwrite(ast):
    nodename = getNodename()
    label=symbol.sym_name.get(int(ast[0]),ast[0])
    print '    %s [label="%s' % (nodename, label),
    if isinstance(ast[1], str):
        if ast[1].strip():
            print '= %s"];' % ast[1]
        else:
            print '"]'
    else:
        print '"];'
        children = []
        for  in n, childenumerate(ast[1:]):
            children.append(dotwrite(child))
        print ,'    %s -> {' % nodename
        for  in :namechildren
            print '%s' % name,
```