

\*2진수, 16진수 정리

공통점 : 컴퓨터가 씀

? 2진수와 16진수의 목적이 일치하는 것 아닐까?

0101010101      2 진수  
0x155              16 진수

10101010101010101010      2 진수  
0x2aaaaa                          16 진수

컴퓨터 기계어를 사용하면서도 16진수는 인간이 상대적으로 쉽게 볼 수 있다.

목적 : 컴퓨터를 배운 사람과 기계의 혼용어라고 보면 된다.

변환 방법 -

-2진수

	128	64	32	16	8	4	2
1							
	2 <sup>7</sup>	2 <sup>6</sup>	2 <sup>5</sup>	2 <sup>4</sup>	2 <sup>3</sup>	2 <sup>2</sup>	2 <sup>1</sup>
2 <sup>0</sup>							
자릿수	7	6	5	4	3	2	1
0							

$$144 = 128 + 16$$

$$2^7 + 2^4 \rightarrow 1001\ 0000$$

자릿수 7 4

-16진수

2진수를 이용하면 빠르다

2진수 1자리 0, 1 - 2개  $2^1$

2자리 00 01 10 11 - 4개  $2^2$

3자리 000 001 100 010 110 101 011 111 - 8개  $2^3$

4자리 16개  $2^4$

2진수에서 16진수는 4자리씩 끊으면 빠르게 구할 수 있다.

1001 0000

9 0

↑ ↖

16진수 1자리 16진수 0자리

-> 0X90 (  $9 * 16^1 + 0 * 16^0 = 144$  )

문1) 10진수 33을 2, 16진수로

$32 + 1 = 2^5 + 2^0$  (2진수의 5자리와 0자리에 1의 값을)

10 0001

이것을 16진수로 - 10 0001

$$16^1 * 2 + 16^0 * 1 = 0x21 = 32$$

문2) 10진수 2568을 2, 16진수로

2진수에서  $2568 = 2048 + 512 + 8$

$$2048 = 2^{11} \quad 512 = 2^9 \quad 8 = 2^3 \quad 1010 \ 0000 \ 1000$$

이것을 16진수로 1010 0000 1000

$$16^2 * a + 16^1 * 0 + 16^0 * 8$$

$$= 0xa08$$

문3) 0x48932110 -> 2진수로

4	8	9	3	2	1	1	0
0100	1000	1001	0011	0010	0001	0001	0000

\*포인터 정리

포인터의 크기?

8 비트 시스템 : 1 byte

16 비트는 2 byte

32 비트는 4 byte

64 비트는 8 byte

컴퓨터의 산술 연산이 ALU(산술 논리 장치)에 의존적이기 때문

ALU의 연산은 범용 레지스터에 종속적이고

컴퓨터가 64비트라는 의미는 이들이 64비트로 구성되어 있음을 의미한다.

변수의 정의는 메모리에 정보를 저장하는 공간이었다

포인터의 정의는 메모리에 주소를 저장하는 공간이다

그렇다면 64비트로 표현할 수 있는 최대값 또한 저장할 수 있어야한다.

(포인터의 크기가 작으면 이 주소를 표현할 방법이 없기때문에 최대치인 64비트가 포인터의 크기가 된 것)

증명 - 터미널창으로 ..

vi pointer\_size.c

```
#include <stdio.h>
```

```
int main(void) {  
    printf("sizeof(int *) = %lu\n", sizeof(int *));  
    printf("sizeof(double *) = %lu\n", sizeof(double *));  
    printf("sizeof(float *) = %lu\n", sizeof(float *));  
    return 0;  
}
```

전부 8이 나오는 것을 확인할 수 있다.

왜 이런걸 하나? 스택의 동작 가정이 포인터의 베이스이기 때문이다

또한 모든 컴퓨터의 동작과정이 이 포인터 베이스로 동작하게 됨