

2주차 2차시 데이터 표현 방식의 이해

【학습목표】

1. 데이터 표현 방식에 대해 설명할 수 있다.
2. 정수와 실수의 표현방식을 구분할 수 있다.

학습내용1 : 데이터 표현 방식

1. 컴퓨터가 데이터를 표현하는 방식

- 1) 2진수, 10진수, 16진수란 무엇인가?



√2진수

- 두 개의 기호를 이용해서 값(데이터)를 표현하는 방식

√10진수

- 열 개의 기호를 이용해서 값(데이터)을 표현하는 방식

√N진수

- N개의 기호를 이용해서 값(데이터)을 표현하는 방식

10 진수	16 진수	10 진수	2진수
9	9	0	0
10	A	1	1
11	B	2	10
12	C	3	11
13	D	4	100
14	E	5	101
15	F		
16	10		
17	11		

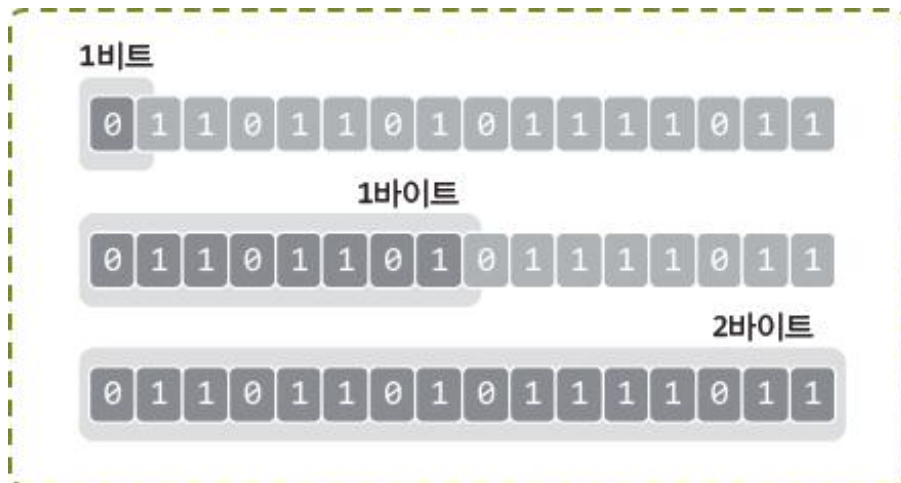
자릿수 증가

자릿수 증가

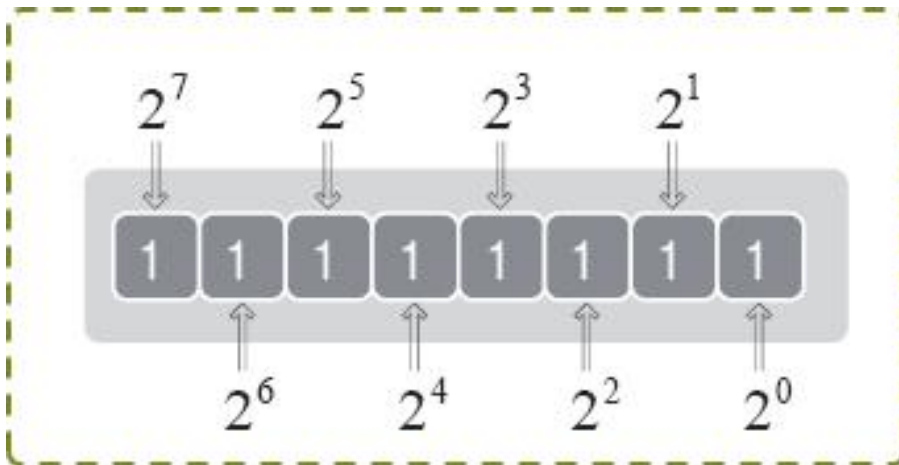
자릿수 증가

자릿수 증가

2) 데이터의 표현단위인 비트(Bit)와 바이트(Byte)



컴퓨터 메모리의 주소 값은 1바이트당 하나의 주소가 할당되어 있다.
따라서 바이트는 컴퓨터 에 있어서 상당히 의미가 있는 단위이다



왼쪽의 의미 있는 정보를 이용하면 2진수를 쉽게 10진수로 변환할 수 있다.

3) 8진수와 16진수를 이용한 데이터 표현

```
int num1 = 10;    // 특별한 선언이 없으면 10진수의 표현
int num2 = 0xA;   // 0x로 시작하면 16진수로 인식
int num3 = 012;   // 0으로 시작하면 8진수로 인식
```

```

int main(void)
{
    int num1=0xA7, num2=0x43;
    int num3=032, num4=024;

    printf("0xA7의 10진수 정수 값: %d \n", num1);
    printf("0x43의 10진수 정수 값: %d \n", num2);
    printf(" 032의 10진수 정수 값: %d \n", num3);
    printf(" 024의 10진수 정수 값: %d \n", num4);

    printf("%d-%d=%d \n", num1, num2, num1-num2);
    printf("%d+%d=%d \n", num3, num4, num3+num4);
    return 0;
}

```

실행결과

```

0xA7의 10진수 정수 값: 167
0x43의 10진수 정수 값: 67
 032의 10진수 정수 값: 26
 024의 10진수 정수 값: 20
167-67=100
26+20=46

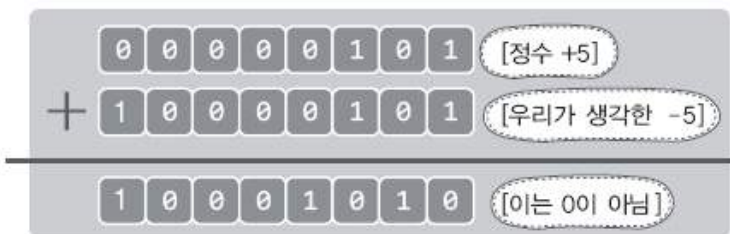
```

학습내용2 : 정수와 실수의 표현방식

1. 정수의 표현 방식

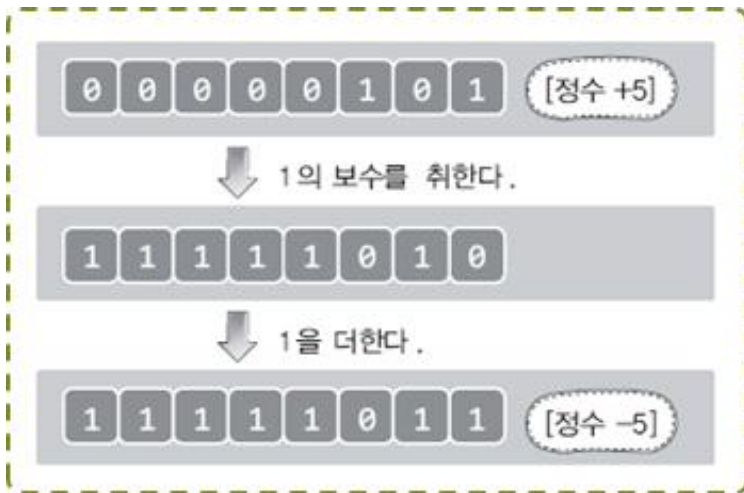


- 가장 왼쪽 비트를 MSB(Most Significant Bit)라 한다.
- MSB는 부호를 나타내는 비트이다.
- MSB를 제외한 나머지 비트는 크기를 나타내는데 사용된다
- 정수의 표현방식은 바이트의 크기와는 상관 없다.
- 바이트의 크기가 크면 그만큼 넓은 범위의 정수를 표현할 수 있을 뿐이다.



- 왼쪽에서는 양의 정수를 표현하는 방식으로 음의 정수를 표현하는 것이 적절치 않은 이유를 설명한다.

2. 음의 정수의 표현 방식



· 음의 정수를 표현하는 방식

- 2의 보수를 취하여 음의 정수를 표현한다

3. 실수의 표현방식

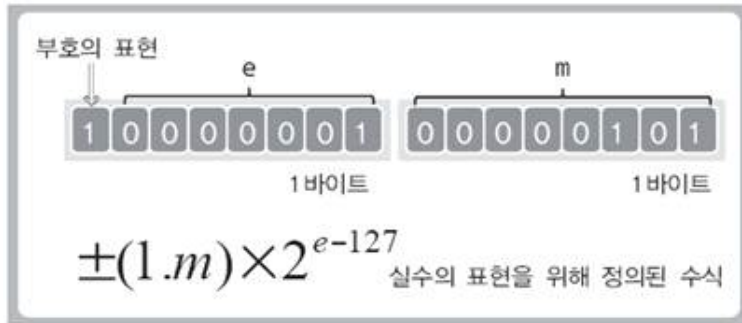
다음의 방식과 같이 정수를 표현하는 방식과 유사하게 실수를 표현하면 다음의 문제가 따른다.

- 표현할 수 있는 실수의 수가 몇 개 되지 않는다.
- 3.12456과 3.12457 사이에 있는 무수히 많은 실수조차 제대로 표현하지 못한다.



따라서 실수의 표현방식은 정수의 표현방식과 다르다.

- 실수의 표현방식에서는 정밀도를 포기하는 대신에 표현할 수 있는 값의 범위를 넓힌다.
- 따라서 컴퓨터는 완벽하게 정밀한 실수를 표현하지 못한다.



오차가 존재하지만 적절한 실수의 표현방법

4. 실수표현의 오차 확인하기

```
int main(void)
{
    int i;
    float num=0.0;

    for(i=0; i<100; i++)
        num+=0.1; // 이 연산을 총 100회 진행하게 됩니다.

    printf("0.1을 100번 더한 결과: %f \n", num);
    return 0;
}
```

0.1을 100번 더한 결과: 10.000002

이론적으로 오차 없이 모든 실수를 완벽하게 표현할 능력이 있는 컴퓨팅 환경은 존재하지 않는다.
즉, 실수 표현에 있어서의 오차 발생은 C언어의 특성이 아닌 컴퓨터의 기본적인 특성이다.

【학습정리】

1. 컴퓨터 메모리의 주소값은 1바이트당 하나의 주소가 할당되어 있다.
2. MSB는 부호를 나타내는 비트이다,
3. 음의 정수를 표현하는 방식은 2의 보수를 취하여 음의 정수를 표현한다.
4. 실수의 표현방식은 정밀도를 포기하는 대신에 표현할 수 있는 값의 범위를 넓힌다.