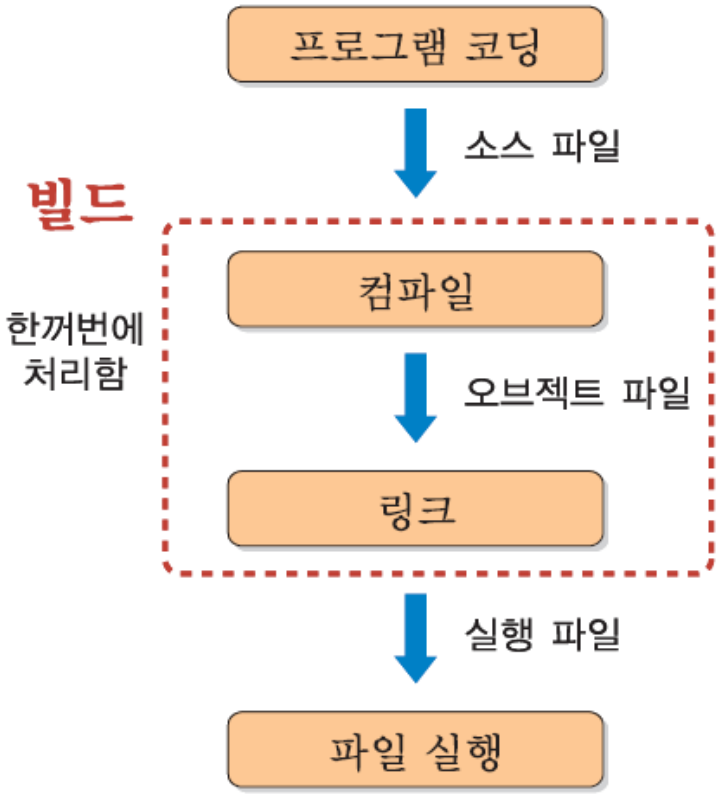
4 장 : 자주 쓰이는 C 연산

```
a=5;
b=3;
c=a+b;
printf("%d",c);
```

JCnet 제이씨넷 신 상 석

목차

C 언어 실행환경



소스 프로그램을 코딩하고, 확장자를 *.c로 한다.

작성된 *.c를 기계어로 변환한다.
그 결과로 오브젝트 파일이 생성된다.

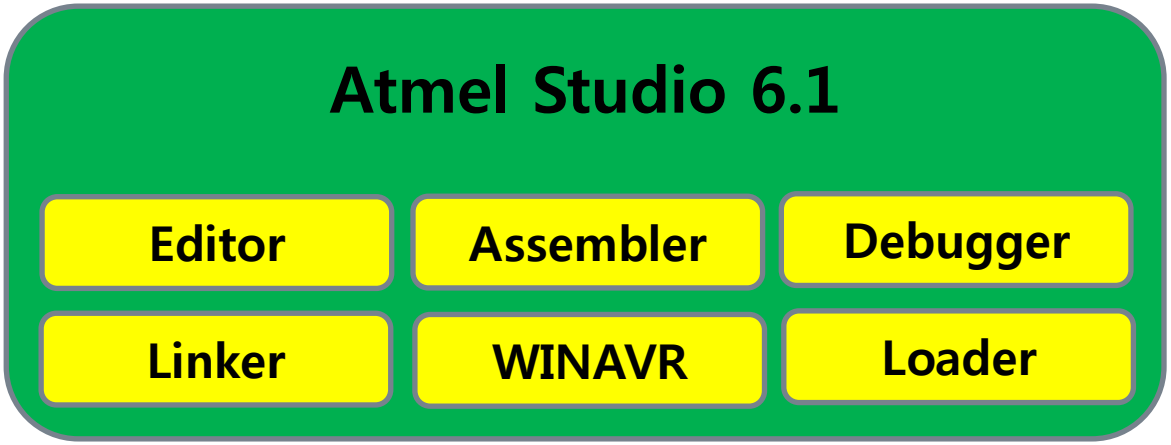
관련 파일을 하나로 결합한다.
그 결과로 실행 파일이 생성된다.

생성된 *.exe 파일을 실행한다. **(or 다운로드)**
(*.bin, *.hex ...)

C 언어 실행환경

- Atmel Studio 6.1 에서의 C 언어 실행 환경

우리가 사용하는
컴파일러는 ? 링커는? IDE는?



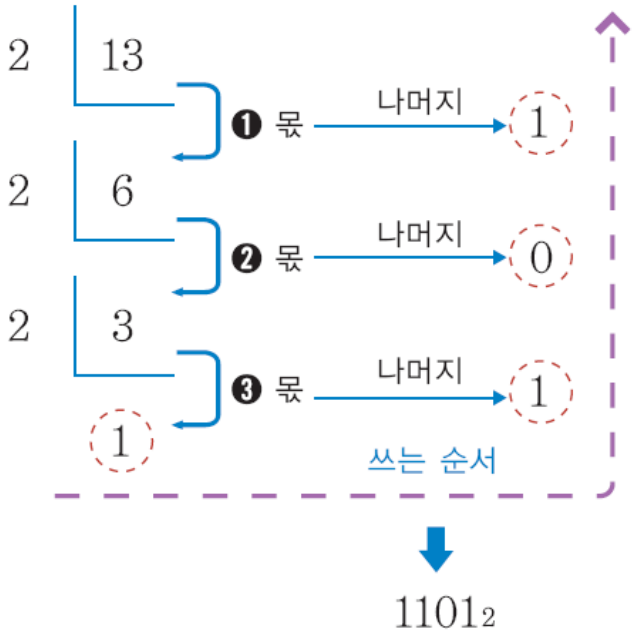
데이터의 표현

- 진수(진법)
  - 10진수 : 0~9의 숫자로 10 가지 숫자를 표현하며, 10 이상의 수는 자릿수를 한자리 올려서 표현
  - 2진수 : 0~1의 숫자로 2 가지 숫자를 표현하며, 2 이상의 수는 자릿수를 한자리 올려서 표현
  - 16진수 : 0~9의 숫자와 A~F의 문자로 16 가지의 숫자를 표현하며, 16 이상의 수는 자릿수를 한자리 올려서 표현

1234 **01001110** **A3**

데이터의 표현

- 10진수 ➔ 2진수 변환
  - 10진수를 계속 2로 나눠가면서 몫과 나머지를 구하고,
  - 더 이상 나눌 수 없을 때의 몫과 나머지를 역순으로 모아 2진수를 형성

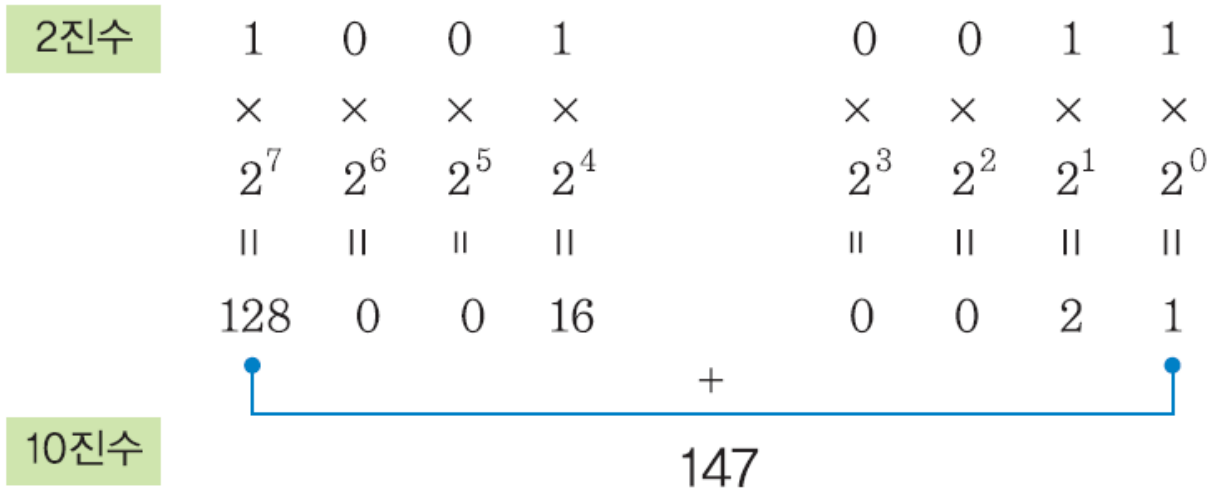


23 → 00010111

129 → 10000001

데이터의 표현

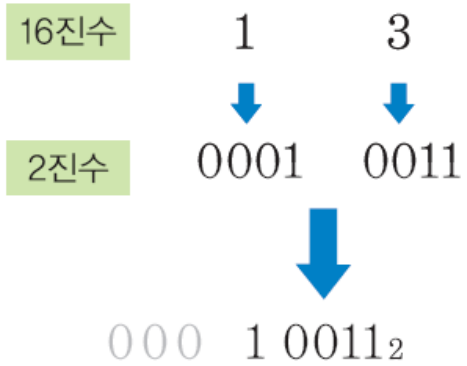
- 2진수 ➔ 10진수 변환
  - 2진수의 맨 뒤에서부터 각 자리에 해당하는 가중치(20, 21, 22,...)를 곱한 후,
  - 각 자리의 결과를 모두 합한다



01111111 → 127

데이터의 표현

- 16진수 ←→2진수 변환
  - 16진수 1자리 = 2진수 4자리 해당



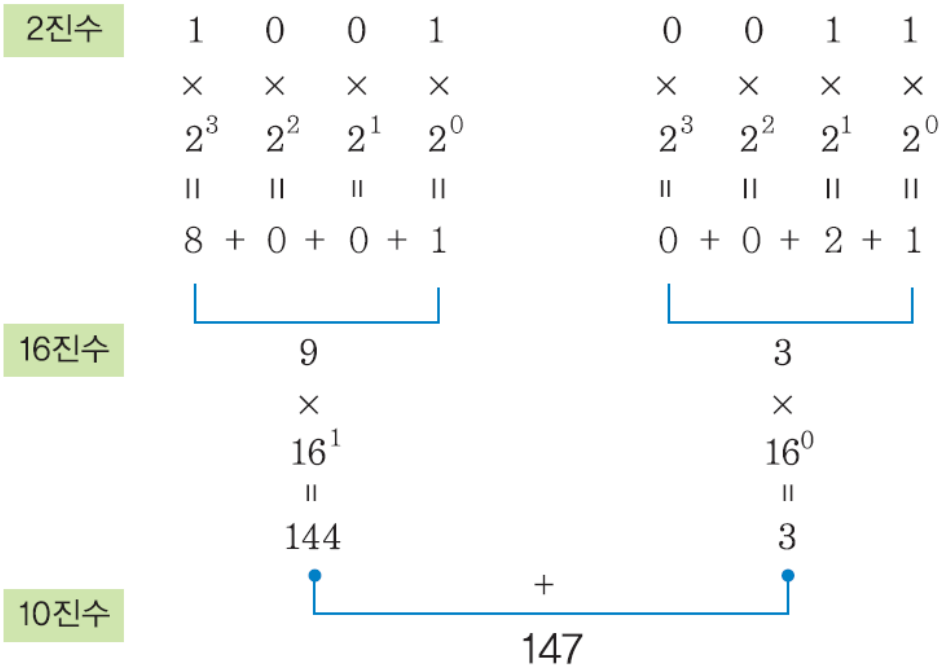
11000101 → 0xc5

0xa9 → 10101001

16진수	2진수
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
A	1010
B	1011
C	1100
D	1101
E	1110
F	1111

데이터의 표현

- 16진수 ➔ 10진수
  - 16진수를 10진수로 바꾸려면 각 자리의 가중치(160, 161,…)를 곱한 후,
  - 각 자리의 결과를 모두 합한다



0xc5 →

0x3a →

데이터의 표현

- 10진수 ➔ 16진수
  - 10진수를 16진수로 바꾸려면 먼저 10진수를 2진수로 바꾼 후, 다시 2진수를 4비트씩 모아 16진수로 바꾼다.

57 → 0b →

253 → 0b →

데이터의 표현

- 진수(진법) 종합
  - 진수(진법) 변환이 자유자재로 바로바로 실행될 수 있도록 원리를 알고 외어야 함

십진수	2진수	16진수
00	0000	0
01	0001	1
02	0010	2
03	0011	3
04	0100	4
05	0101	5
06	0110	6
07	0111	7
08	1000	8
09	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

논리 연산, 비트 연산, 시프트 연산

- &&, ||, !, &, |, ~, >>, <<
- a, b가 unsigned char이고 a = 3, b = 4 일 때
  - a, b 각각을 2진수와 16진수로 표시
  - a && b =
  - a || b =
  - a & b =
  - !a | b =
  - ~a >> 2 =
  - b << a =

선택적 세트 / 선택적 클리어(마스크) 연산

- 선택적 세트(Selective Set) : B 레지스터의 비트들 중에서 1인 비트들과 같은 위치에 있는 A 레지스터의 비트들을 1로 세트 <OR 이용>

  A = 1 0 0 1 0 0 1 0 (연산전)

  B = 0 0 0 0 1 1 1 1

  A = 1 0 0 1 1 1 1 1 (연산결과)

- 특정 비트 세트에 이용

  bit3를 세트 하려면?

- 선택적-클리어(Selective Clear) : B 레지스터의 비트들 중에서 0인 비트들과 같은 위치에 있는 A 레지스터의 비트들을 0으로 클리어 <AND 이용>

  A = 1 0 0 1 0 1 0 1 (연산 전)

  B = 0 0 0 0 1 1 1 1

  A = 1 0 0 1 1 0 1 0 (연산결과)

- 특정 비트 클리어에 이용

  bit4를 클리어 하려면?

삽입 연산

- 삽입(insert) 연산 : 새로운 비트 값들을 데이터 단어내의 특정 위치에 삽입하는 연산
- 삽입할 비트 위치들에 대하여 선택적클리어(AND) 연산 수행 후 새로이 삽입할 비트들과 선택적세트(OR) 연산을 수행

A의 왼쪽 4비트에 1110 삽입

```
A  =  1 0 0 1 0 1 0 1
B  =  [          ]      선택적세트(AND) 연산
------------------
A  =  0 0 0 0 0 1 0 1    첫 단계 결과
B  =  [          ]      선택적클리어(OR) 연산
------------------
A  =  1 1 1 0 0 1 0 1    최종(삽입 연산) 결과
```

포인터 연산

- char *p;
  - 변수 이름은 p이고, p는 어드레스 값을 가지고 있는 포인터임
  - 처음 선언할 때만 사용하는 방식
  - 초기값을 줄 수 있음
    - 예 : char *p = 0x01000000; / p에는 0x01000000 값 들어감
- p = &a; 또는 p = (char *)0x00001fe0;
  - 포인터변수 p에 a 변수가 저장된 어드레스 값을 넣음
  - 또는, p에 0x0001fe0 이라는 어드레스 값을 넣음
- *p = 0x1e; 또는 *p = abc;
  - 포인터 변수 p가 가르키는 어드레스에 0x1e 값을 넣음
  - 또는, p가 가르키는 어드레스에 abc 변수 값을 넣음

포인터 연산

```
int main(void)
{
    unsigned char *p;
    int *q;
    unsigned char abc = 12;
    p = &abc;
    printf("p = %x, *p = %d₩n", (int)p, *p);
    q = (int *)0x01000000;
    *q = 12345678;
    printf("q = %x, *q = %d₩n", (int)q, *q);
}
```

묻고 답하기

Q & A