

Assembly Code

Assembly code

Pointer related operator

```
int num = 10;

int * ptr = &num;
//lea eax, [num]
//mov dword ptr[ptr], eax

*ptr = 25;
//mov eax, dword ptr[ptr]
//mov dword ptr[eax], 25
```

Assembly code

lea(load effective address)

lea REG, [address]

```
6:      int num = 10;
002517EE  mov     dword ptr [num],0Ah
7:
8:      int * ptr = &num;
002517F5  lea     eax,[num]
002517F8  mov     dword ptr [ptr],eax
```

C++

- &num은 num의 주소값을 반환

assembly

- lea(load effective address)는 eax에 num의 주소값을 저장

Assembly code

mov(move)

mov dword ptr[address], REG

```
6:      int num = 10;
002517EE  mov     dword ptr [num],0Ah
7:
8:      int * ptr = &num;
002517F5  lea     eax,[num]
002517F8  mov     dword ptr [ptr],eax
```

C++

- ptr도 그저 주소 값을 담는 변수일 뿐이다.

assembly

- mov를 통해 eax에 저장된 주소 값을 ptr에 복사

mov는 메인 메모리에서 메인 메모리의 이동은 불가능

Assembly code

mov(move)

mov REG, dword ptr[address]

```
11:
12:      *ptr = 25;
002517FB  mov     eax,dword ptr [ptr]
002517FE  mov     dword ptr [eax],19h
13:      //mov eax, dword ptr[ptr]
```

C++

- * 연산자는 ptr에 저장된 주소의 공간에 접근

assembly

- mov로 ptr에 저장된 값, 즉 주소값을 eax로 가져온다.

Assembly code

lea(load effective address)

mov dword ptr[address], constant

```
11:
12:      *ptr = 25;
002517FB  mov     eax,dword ptr [ptr]
002517FE  mov     dword ptr [eax],19h
13:      //mov eax, dword ptr[ptr]
```

C++

- 접근한 공간(num)에 25를 저장

assembly

- eax에는 num의 주소 값이 저장되어 있다.
- mov로 eax를 주소 값으로 해당 공간에 접근하여 25(hex로 19h)를 저장