EDDI
Electronic Design
Development Institute

에디로봇아카데미
임베디드 마스터 Lv1 과정

제 3기

2022. 02. 11

김원석

CONTENTS

EDDI
Electronic Design
Development Institute

구조체

EDDI
Electronic Design Development Institute

- typedef : 실제 C 타입을 우리가 원하는 간소화된 타입으로 변경하는 작업을 지원한다.

구조체를 사용하는 이유?

- 자신만의 새로운 데이터 타입을 만들 수 있다.
- 결국 구조체 = 데이터 타입
- 즉, 변수나 배열을 만들 때 적용했던 규칙들이 그대로 적용될 수 있다.

ex)

```
typedef struct array_list list;
struct array_list
{
        int data;
        struct array_list *link;
};
```

- array_list라는 구조체를 list라는 데이터 타입으로 간소화
- array_list 구조체 정의

주의사항

```c
// 이 부분은 잘못 만들어진 부분입니다!
// 요런 실수를 하면 안되니 예로 남겨두겠습니다.
void set_both_link (list target1, list target2)
{
        target1.link = &target2;
        target2.link = &target1;
}

int main(void)
{
        // 1.
        struct array_list test1 = { 3, NULL };
        // 2.
        list test2 = { 7, NULL };

        set_both_link(test1, test2);

        print_list_struct(test1);
        print_list_struct(test2);

        return 0;
}
```

```
(gdb)
36              set_both_link(test1, test2);
(gdb)
38              print_list_struct(test1);
(gdb) r
The program being debugged has been started already.
Start it from the beginning? (y or n) y
Starting program: /home/oem/proj/eddi/academy/EmbeddedMasterLv1/37| /LSH/c/6/a.out

Breakpoint 1, main () at struct_ex2.c:30
30      {
(gdb) n
32              struct array_list test1 = { 3, NULL };
(gdb)
34              list test2 = { 7, NULL };
(gdb)
36              set_both_link(test1, test2);
(gdb) p test1
$1 = {data = 3, link = 0x0}
(gdb) p test2
$2 = {data = 7, link = 0x0}
(gdb) p &test1
$3 = (struct array_list *) 0x7fffffffda90
(gdb) p &test2
$4 = (list *) 0x7fffffffdaa0
(gdb) s
set_both_link (target1=..., target2=...) at struct_ex2.c:24
24      {
(gdb) n
25              target1.link = &target2;
(gdb) p &target2
$5 = (list *) 0x7fffffffda60
(gdb) p &target1
$6 = (list *) 0x7fffffffda70
(gdb)
```

target1과 target2는 test1과 test2의 값에 의한 호출이므로 set_both_link 함수의 스택 프레임에서 값을 바꿔도 main함수 프레임의 test1과 test2에는 영향을 주지 못한다.

주의사항


```c
void set_both_link (list *target1, list *target2)
{
        // 포인터 변수를 통해 구조체 내부에 접근하는 경우엔 '->' 연산자를 사용합니다.
        target1->link = target2;
        target2->link = target1;
}

int main(void)
{
        // 1.
        struct array_list test1 = { 3, NULL };
        // 2.
        list test2 = { 7, NULL };

        set_both_link(&test1, &test2);

        print_list_struct(test1);
        print_list_struct(test2);

        return 0;
}
```

다음과 같이 주소에 의한 호출을 하면 set_both_link 함수의 스택 프레임에서도 주소에 접근해 값을 바꿔 main 함수의 스택 프레임 속 test1과 test2의 값을 변경할 수 있다.

CONTENTS

EDDI
Electronic Design
Development Institute

CONTENTS

EDDI
Electronic Design
Development Institute