

1) c 파일

```
cse20181662@cspro7: ~  
1 #include <stdio.h>  
2 #include <stdlib.h>  
3  
4 enum season {Spring, Summer, Fall, Winter} s1;  
5 float fl1;  
6 union myuni {  
7     char *a;  
8     int b;  
9     char c;  
10 } u1;  
11 char CC;  
12  
13 struct myst {  
14     union myuni u;  
15     char k;  
16     int i;  
17 } st1[5][5][5];  
18 void main() {  
19     s1 = Summer;  
20     fl1 = -1.125;  
21     CC = 'a';  
22     u1.a = &CC;  
23     st1[3][2][1].u.a = &CC;  
24     st1[3][2][2].i = 20;  
25 }
```

~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~  
~

25:1 [All] ~/hw3.c\

80% Q 4/30 2:43 PM ssh ~

## 2) s 파일

```
cse20181662@cspro7: ~  
1  .file    "hw3.c"  
2  .text  
3  .comm    s1,4,4  
4  .comm    fl1,4,4  
5  .comm    u1,8,8  
6  .comm    CC,1,1  
7  .comm    st1,2000,32  
8  .globl   main  
9  .type    main, @function  
10 main:  
11 .LFB6:  
12 .cfi_startproc  
13 endbr64  
14 pushq    %rbp  
15 .cfi_def_cfa_offset 16  
16 .cfi_offset 6, -16  
17 movq     %rsp, %rbp  
18 .cfi_def_cfa_register 6  
19 movl     $1, s1(%rip)  
20 movss    .LC0(%rip), %xmm0  
21 movss    %xmm0, fl1(%rip)  
22 movb     $97, CC(%rip)  
23 leaq     CC(%rip), %rax  
24 movq     %rax, u1(%rip)  
25 leaq     CC(%rip), %rax  
26 movq     %rax, 1376+st1(%rip)  
27 movl     $20, 1404+st1(%rip)  
28 nop  
29 popq     %rbp  
30 .cfi_def_cfa 7, 8  
31 ret  
32 .cfi_endproc  
33 .LFE6:  
34 .size    main, .-main  
35 .section .rodata  
36 .align 4  
37 .LC0:  
38 .long    3213885440  
39 .ident    "GCC: (Ubuntu 9.4.0-1ubuntu1~20.04.1) 9.4.0"  
40 .section .note.GNU-stack,"",@progbits  
41 .section .note.gnu.property,"a"  
42 .align 8  
43 .long    1f - 0f  
44 .long    4f - 1f  
45 .long    5  
46 0:  
47 .string  "GNU"  
48 1:  
49 .align 8  
50 .long    0xc0000002  
51 .long    3f - 2f  
52 2:  
53 .long    0x3  
54 3:  
55 .align 8  
56 4:  
~  
~  
56:2 [ALL] ~/hw3.s\
```

### 3)해석

s파일의 3번째 줄부터는 전역 변수의 메모리 할당이 일어난다. enum season 타입인 s1의 경우 int와 같은 4바이트, float 타입의 f1은 4바이트가 할당된다. union myuni 타입인 u1의 경우 내용의 char\*, int, char 중 가장 사이즈가 큰 char\*의 크기인 8바이트만큼 할당되며, cc의 경우 char의 크기인 1바이트만큼 할당된다. struct myst 타입의 3차원 array인 st1를 살펴보면, 우선 myst의 크기는 myuni(8바이트), char(1바이트 + 3바이트의 padding), int(4 바이트)에 해당하는 16바이트가 된다. 3차원 배열이 5\*5\*5 이므로 총 크기는  $16 * 125 = 2000$ 바이트가 된다.

s파일의 10번째 줄의 main 부분부터 살펴보면, 19번째 줄의 movl \$1, s1(%rip)라는 명령어를 통해 s1에 1의 값을 할당하는데, 이는 enum 타입에서 2번째 값에 해당하는 Summer를 1로 변환한 것에 해당한다.

다음으로 c파일의 20번째 줄에 floating point 변수에 값을 할당하는 부분이 있는데, s파일에서 해당 부분을 살펴보자. s파일의 20번째 줄에 movss .LC0(%rip), %xmm0 라는 명령어가 있는데, movss는 부동소수점 값 하나를 복사해주는 명령어이다. %xmm0 레지스터는 부동소수점 값을 저장하기 위한 레지스터이다. 복사할 값에 해당하는 .LC0(s파일의 37번째 줄)에서 해당 내용을 보면, long 값으로 3213885440이라는 값이 있다. 이 수가 어떤 방식으로 표시된 수 인지 생각하여 보면, 우선 해당 값을 이진수로 변경하면 10111111100100000000000000000000<sub>(2)</sub> 라는 값이 된다. 이는 IEEE 754 format 방식에 의해  $-1.001 * 2^0$ 으로 변환되며 최종적으로 -1.125라는 값을 의미하게 된다. 해당 값을 %xmm0를 통해 f1에 할당하게 되는 것이다.

char 타입 변수인 cc에 'a'를 할당하는 부분을 s파일의 22번째 줄에서 살펴볼 수 있는데, 이 때 'a'의 ASCII code인 97을 cc에 복사하는 것을 확인할 수 있다. 이후 char\* 타입인 u1.a에 cc의 주소를 할당하는 부분(c파일 22번째 줄)을 어셈블리로 변환한 부분(s파일 23번째 줄~25번째 줄)을 보면 leaq 명령어를 사용하는 것을 볼 수 있다. leaq란 load effective address quad word로 64비트 메모리를 참조하는 명령을 의미한다. 즉 rax 레지스터에 cc의 주소를 옮긴 후 그것을 u1에 옮기는 것으로 pointer에 주소를 할당하는 것을 구현하는 것이다.

이후 c파일의 23번째 줄에 st1[3][2][1].u.a에 마찬가지로 cc의 주소를 할당하는 부분을 보자. myst 타입 변수 하나의 크기가 16바이트였고, 배열이 [5][5][5]의 사이즈를 가지고 있던 것을 생각하면 st1[3][2][1]은  $(5^2 * 3 + 5 * 2 + 1) * 16 = 1376$  번째 바이트부터 시작한다는 것을 알 수 있다. 구조체에서 myuni u가 가장 먼저 선언되었으므로 1376번째 위치에 바로 cc의 주소가 있던 rax 레지스터의 값을 복사해 주는 것을 확인할 수 있다.

비슷하게 c파일의 24번째 줄의 경우 st1[3][2][2]의 위치가  $(5^2 * 3 + 5 * 2 + 2) * 16 = 1392$ 이며, myuni 타입의 u가 8바이트를 차지하고 char타입의 k가 1바이트와 padding인 3바이트만큼 차지하여 i의 시작 위치가  $1392 + 12 = 1404$ 가 되어  $1404 + st1(\%rip)$ 에 20의 값을 할당하는 것을 확인할 수 있다.