

시스템프로그래밍 실습 과제 보고서

1. 실행 결과 (디렉토리 path 까지 보여주는 console 창을 스크린 캡처)

```
root@debian-armhf: ~/syspro/lab11# ld lab11.o -lgnu -L. -o lab11
root@debian-armhf:~/syspro/lab11# ./lab11
S
83
root@debian-armhf:~/syspro/lab11#
root@debian-armhf:~/syspro/lab11# ./lab11
0
48
root@debian-armhf:~/syspro/lab11#
root@debian-armhf:~/syspro/lab11# _
```

-----lab11 결과-----

```
Machine view
1  .text
2  .global _my_disp_dec
3  .extern _my_divmod
4  .global dd1
5
6  msgSize = 10
7
8  _my_disp_dec:
9      push {r0-r5, r7, lr}
10     @ldr r5, =msg
11     sub sp, sp, #20
12     mov r5, sp
13     add r5, r5, #msgSize
14     mov r2, r5
15     add sp, sp, #20
16     b dd1
17     bx lr
18
19 dd1:
20     bl _my_divmod
21     add r1, r1, #'0'
22     strb r1, [r5, #-1]!
23     cmp r0, #0
24     bne dd1
25
26
27     mov r1, r5
28     mov r0, #1
29     sub r2, r2, r5
30     mov r7, #4
31     swi 0
32     pop {r0-r5, r7, pc}
33     @add sp, sp, #4
34
35
36 .end
~
~
~
```

-----disp_dec.s 코드-----

```

1 .text
2 .global _my_divmod
3
4
5 _my_divmod:
6     push {r2-r4}
7     mov r3, r0
8     movw r2, #0x6667
9     movt r2, #0x6666
10    smull r1, r4, r2, r0
11    mov r1, r0, asr #31
12    rsb r2, r1, r4, asr #2
13    mov r0, r2
14    add r4, r2, r2, asl #2
15    sub r1, r3, r4, asl #1
16    pop {r2-r4}
17    mov pc, lr
18
19 .end
20
21

```

-----divmod.s 코드-----

2. 결론 (코드 설명 및 결과 분석)

각 코드 파일에 .global 과 .extern 으로 라이브러리에 있는 함수를 선언해 주었다.

링커를 이용할 경우 전역변수의 사용을 불가하기에 버퍼를 스택에 저장하여 사용하여야한다.

Disp_dec.s 파일의 10 번째 줄부터 ldr 함수를 없애고 sub sp,sp, #4 를 이용하여 sp 의 위치를 4 만큼 이동시켜준후 sp 에 들어있는 msg 를 r5 로 mov 시켜준다. 이후 함수가 끝난후 add sp, sp, #4 를 통하여 sp 의 위치를 원래대로 돌려준다.

이과정을 통해서 ldr r5, =msg 를 대체하여 라이브러리의 함수를 통하여 disp_dec 를 할수 있게 된다.