

시스템프로그래밍 실습 과제 보고서

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

```
1 .text
2 _start: .global _start
3     @ r3 <= msg
4     @ r4 <= idx
5     adr r3, msg      @r3 = msg
6     mov r4, #0        @r4 = lidx = 0
7     mov r5, #1
8     mov r6, #0
9 loop:
10    cmp r5, #0    @while(idx < 14)
11    beq exit      @{
12    @ r3 <= msg
13    @ r4 <= idx
14    @ sys_write(fd, ptr, len);
15    @ r7 = 4, r0 r1 r2
16    mov r0, #1    @ fd = stdout
17    add r1, r3, r4 @ ptr = @msg[idx]
18    mov r2, #1    @ len = 1
19    mov r7, #4    @ **can move above loop
20    swi 0         @ sys_write(1,&msg[idx],1)
21    ldrb r5, [r1]
22    add r4, r4, #1 @ idx ++;1
23    b loop        @ }
24 exit:
25    @ sys_exit(0)
26    @ r7 = 1 r0
27    mov r0, #0    @ exit code = 0
28    mov r7, #1
29    swi 0         @ sys_exit(0)
30    .align
31 msg:
32 .asciz "Hello, World!\0zero\n"
33 .end
```

"hello.s" 32L, 846C written
root@debian-armhf:~/syspro/lab3# ./hello
Hello, World!_root@debian-armhf:~/syspro/lab3#

----1. Hello.s 코드와 결과

```

1  _text
2  _start: .global _start
3      adr r3, msg
4      mov r4, #0
5      mov r5, #1
6
7
8  loop:
9      cmp r5, #0
10     beq exit
11
12     mov r0, #1
13     add r1, r3, r4
14     mov r2, #1
15     mov r7, #4
16     swi 0
17     LDRB r5, [r1]
18     add r4, r4, #1
19     b loop
20     bl disp_str
21 disp_str:
22     mov r0, #1
23     adr r1, msg
24     mov r2, #1
25     mov r7, #4
26     @swi
27     mov pc, lr
28
29
30 exit:
31     mov r0, #0
32     mov r7, #1
33     swi 0
34     .align
35 msg:
36     .asciz "Helloo, World!\n\nOkk"
37 .end
38

```

```

"dis.s" 38L, 448C written
root@debian-armhf:~/syspro/lab3# ./dis
Helloo, World!
root@debian-armhf:~/syspro/lab3#

```

2. Dis.s 코드와 결과

2. 결론 (코드 설명 및 결과 분석)

1 번의 경우

21 번째 줄에 `ldrb r5, [r1]` 을 추가하여 `r1` 의 내용을 `r5` 를 루프의 마지막에 저장한다. 루프문의 조건에서 10 번째 줄에 `cmp r5, #0` 을 통하여 `r5` 의 내용이 null 된다면 루프에서 빠져 나온다.

이과정을 통해 `msg` 출력값에서 `w0` 을 만나면 그대로 출력 후 종료 할 수 있도록 한다.

2 번의 경우

1 번에서 , 2 번째 줄에 `disp_str:` 을 추가하여 `msg` 를 호출하는 함수를 만든다. 21 번째 줄에 `bl disp_str` 을 추가하여 loop 안에서 `disp_str` 을 추가하여 함수안에서 `disp_str` 을 호출할 수 있도록 한다.