

# 시스템프로그래밍 실습 과제 보고서

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

```
1  _text
2  _start: .global _start
3      adr r4, dot
4      mov r3, #3
5      ldrb r5, [r4]
6  read:
7
8  @sys_read(stdin, buf, 1);
9
10     mov r0, #0 @ fd = stdin
11     ldr r1, =buf @ buf = buf
12     mov r2, #1 @ len = 1
13     mov r7, #3 @ syscall = sys_read
14
15     @ldrb r3, [r1]
16     @ldrb r5, [r4]
17     @cmp r3, r5
18     @beq exit
19     swi 0
20     ldrb r3, [r1]
21     cmp r3, r5
22     beq exit
23
24 @sys_write(stdout, buf, 1);
25     mov r0, #1 @fd = stdout
26     ldr r1, =buf @buf = buf
27     mov r2, #1 @ len = 1
28     mov r7, #4 @ syscall = sys_write
29     swi 0
30
31     b read
32
33 exit:
34 @sys_exit(0);
35     mov r0, #0
36     mov r7, #1
37     swi 0
38
39 dot: .asciz "."
40
41 .data
42 .align
43 buf: .skip 4
44 .end
~
~
~
"lab05.s" 44L, 633C
```

```
QEMU - Press Ctrl+Alt+G to release grab
Machine View
11 ldr r1, =buf @ buf = buf
12 mov r2, #1 @ len =1
13 mov r7, #3 @ syscall = sys_read
14
15 @ldrb r3, [r1]
16 @ldrb r5, [r4]
17 @cmp r3, r5
18 @beq exit
19 swi 0
20 ldrb r3, [r1]
21 cmp r3, r5
22 beq exit
23
24 @sys_write(stdout, buf, 1);
25 mov r0, #1 @ fd = stdout
26 ldr r1, =buf @buf = buf
27 mov r2, #1 @ len =1
28 mov r7, #4 @ syscall = sys_write
29 swi 0
30
31 b read
32
33 exit:
34 @sys_exit(0);
35 mov r0, #0
36 mov r7, #1
37 swi 0
38
39 dot: .asciz "."
40
41 .data
42 .align
43 buf: .skip 4
44 .end

"lab05.s" 44L, 633C written
root@debian-armhf:~/syspro/lab5# as lab05.s -o lab05.o
root@debian-armhf:~/syspro/lab5# ld lab05.o -o lab05
root@debian-armhf:~/syspro/lab5# ./lab05
asdf
asdf
qwer
qwer
asdf.
asdfroot@debian-armhf:~/syspro/lab5#
root@debian-armhf:~/syspro/lab5#
```

## 2. 결론 (코드 설명 및 결과 분석)

Dot: 부분에 “.” 을 넣어두고 선언부에서 r4 의 주소에 dot 를 넣어둔다.

선언부에서 r5 에 r4(“.”)를 넣도록 하고 sys\_read 의 swi 0 로 종료된 후 , r3 에 r1 의 값을 넣고 cmp r3, r5 를 통해서 입력 받은 값과 r5(“.”)을 비교한다.

Beq exit 를 통해서 r3 의 입력받은 값이 .일 경우 exit 로 빠져나가도록 한다.

결과 코드를 보면 . 부분은 빠지고 출력 후 코드가 종료된 것을 볼 수 있다.