

# 시스템프로그래밍 실습 과제 보고서

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

```
"put.s" 14L, 189C written  
root@debian-armhf:/syspro/lab10# as put.o -o put.o  
root@debian-armhf:/syspro/lab10# ar rs libryu.a put.o  
root@debian-armhf:/syspro/lab10# ld lab10-1.o -lryu -L. -o lab10-1  
root@debian-armhf:/syspro/lab10# ./lab10-1  
a  
a  
root@debian-armhf:/syspro/lab10#  
root@debian-armhf:/syspro/lab10#
```

----결과----

```

1  _text
2  .global _my_putchar
3
4  _my_putchar:
5      mov r0, #1
6      sub sp, sp, #4
7      mov r1, sp
8      mov r2, #1
9      mov r7, #4
10     swi 0
11     add sp, sp, #4
12     bx lr
13 .end
14

```

```
---put.s---
```

```

1 .text
2 .global _my_getchar
3
4 _my_getchar:
5     mov r0, #0
6     sub sp, sp, #4
7     mov r1, sp
8     @ldr r1, =buf
9     mov r2, #1
10    mov r7, #3
11    swi 0
12    add sp, sp, #4
13    bx lr
14 .end
~
~
~
~
~
~
~

```

---get.s 사진----

## 2. 결론 (코드 설명 및 결과 분석)

get.s 에서 sub sp, sp, #4 를 통해서 스택의 크기를 4 만큼 늘려준다.

받은 입력이 sp 에있음으로 ldr r2, =buf 대신 mov r1, sp 로 r1 에 입력을 넣어준다. 그후  
syscall 을 통해서 글자를 읽어 들인다

그후 스택의 위치를 원래대로 돌려놓는다

마지막으로 \_my\_putchar 함수를 bx lr 을 통해서 빠져나온다

이어서 Put.s 에서도 sub sp, sp, #4 를 통하여 스택의 크기를 4 만큼 늘리고 r1 에 sp 를  
넣은후 syscall 을 통해 출력을 하고 스택의 위치를 원래대로 돌려놓는다.

마지막으로 \_my\_getchar 함수를 bx lr 을 통해서 빠져나온다.