

시스템프로그래밍 실습 과제 보고서

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

```
~
~
~
"lab06.s" 33L, 479C written
root@debian-armhf:~/syspro/lab6# as -g lab06.s -o lab06.o
root@debian-armhf:~/syspro/lab6# ld lab06.o -o lab06
root@debian-armhf:~/syspro/lab6# gdb lab06
GNU gdb (GDB) 7.4.1-debian
Copyright (C) 2012 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law. Type "show copying"
and "show warranty" for details.
This GDB was configured as "arm-linux-gnueabi".
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>...
Reading symbols from /root/syspro/lab6/lab06...done.
(gdb) r
Starting program: /root/syspro/lab6/lab06
[Inferior 1 (process 2613) exited normally]
(gdb) b 24
Breakpoint 1 at 0x8098: file lab06.s, line 24.
(gdb) r
Starting program: /root/syspro/lab6/lab06

Breakpoint 1, _start () at lab06.s:24
24      add sp,sp, #12
(gdb) info reg
r0                0x37          55
r1                0x1e          30
r2                0x19          25
r3                0x4           4
r4                0x0           0
r5                0x0           0
r6                0x0           0
r7                0x0           0
r8                0x0           0
r9                0x0           0
r10               0x0           0
r11               0x0           0
r12               0x0           0
sp                0x7efffdd0     0x7efffdd0
lr                0x8098        32920
pc                0x8098        0x8098 <_start+36>
cpsr              0x10         16
(gdb)
```

```
QEMU
Machine View
1  _text
2  .global _start
3
4  func:
5      add r1, r0, r1, LSL #1
6      add r2, r2, r2, LSL #1
7      add r1, r1, r2
8      add r1, r1, r3, LSL #2
9      ldr r2, [sp]
10     add r2, r2, r2, LSL #2
11     add r0, r1, r2
12     bx lr
13
14 _start:
15     push {lr}
16     sub sp, sp, #12
17     mov r3, #5
18     str r3, [sp]
19     mov r0, #1
20     mov r1, #2
21     mov r2, #3
22     mov r3, #4
23     bl func
24     add sp, sp, #12
25     @pop {pc}
26
27 exit:
28     mov r0, #0
29     mov r7, #1
30     swi 0
31
32 .data
33 .end
```

2. 결론 (코드 설명 및 결과 분석)

5: $r1(w)$ 에 $\times 2$ 한 것을 $r0(v)$ 와 더하여 $r1$ 에 넣어준다.

6: $r2(x)$ 에 $x(r2)\times 2$ 와 x 를 더하여서 $r2$ 에 넣어준다. $\Rightarrow 3x$

7: $r1$ 에 $r2(3x)$ 를 더하여 넣어준다.

8: $r3(y)$ 에 2^2 를 곱하고 $r1$ 에 더하여 넣어준다 $\rightarrow r1 = v+2w+3x+4y$

10: $r2(z)$ 에 2^2 를 곱한 값과 z 를 더하여서 $r2$ 에 넣어준다. $\Rightarrow r2 = 5z$

11: $r2$ 를 $r1$ 에 더하여 넣어준다. $\rightarrow r1 = v+2w+3x+4y+5z$

이 과정을 통하여 $v+2w+3x+4y+5z$ 계산을 완료하였고, 디버그 결과를 보면, $r0$ 에 55 가 들어가 있는 것을 확인 할 수 있다.